Outlier-Free Genomic Foundation Models for Resource-Efficient Training and Low-Bit Inference

Chenghao Qiu [*1] **Haozheng Luo** [*2] **Maojiang Su** [2] **Zhihan Zhou** [2] **Zoe Mehta** [3] **Guo Ye** [2] **Jerry Yao-Chieh Hu** [2] **Han Liu** [2]

Abstract

While genomic foundation models (GFMs) hold significant potential for biological discovery, their large parameter sizes and high computational demands imit practical deployment on resource-constrained devices. We propose **GERM**, an outlier-free architecture that replaces standard attention with an outlier-free mechanism, achieving both accelerated low-rank adaptation and robust post-training quantization. enhances both training and inference via outlier removal. We further propose **GERM-T**, a small-step continual learning strategy with outlier-free framework that leverages existing checkpoints to avoid costly retraining from scratch. Our experiments demonstrate GERM's superiority over state-of-the-art GFMs: it achieves 37.98% higher fine-tuning performance and improves quantization performance by 64.34% , alongside 92.14% reduction in average kurtosis and 82.77% lower maximum infinity norm. Notably, GERM enables rapid deployment on edge devices, completing DNABERT-2 fine-tuning in 5 minutes on a single 2080Ti GPU with 34.9% faster training, 24.79% inference acceleration, and robust 4-bit quantization. GERM consistently delivers superior performance, making it a practical solution for deploying GFMs in resource-constrained settings.

[*]Equal contribution [1]Tianjin University [2]Northwestern University [3]Vernon Hills High School. Correspondence to: Chenghao Qiu <q1320460765@tju.edu.cn>, Haozheng Luo <hluo@u.northwestern.edu>, Maojiang Su <maojiangsu2030@u.northwestern.edu>, Zhihan Zhou <zhihanzhou2020@u.northwestern.edu>, Zoe Mehta <zoe.mehta@vhhscougars.org>, Guo Ye <guoye2018@u.northwestern.edu>, Jerry Yao-Chieh Hu <jhu@u.northwestern.edu>, Han Liu <hanliu@northwestern.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1 Introduction

We propose **GERM**, an efficient DNA genomic foundation model (GFM) that replaces conventional attention mechanisms (Vaswani et al., 2017) with an outlier-free Hopfield layer (Hu et al., 2024). This architecture achieves quantization robustness and rapid adaptability, enabling effective deployment on resource-constrained devices through post-training compression and parameter-efficient fine-tuning.

While current GFMs like DNABERT-2 (Zhou et al., 2024) and GenomeOcean (Zhou et al., 2025b) demonstrate superior task performance, their practical application faces two critical challenges: massive parameter sizes requiring substantial computational resources, and severe performance degradation when applying standard compression techniques like SmoothQuant (Xiao et al., 2023) or LoRA-based adaptation (Hu et al., 2022). Prior studies (Clark et al., 2019; Kovaleva et al., 2019) identify transformer attention outliers as the primary cause of these limitations. We address this by replacing standard transformer attention mechanisms with an outlier-free attention layer proposed by Hu et al. (2024), which detects and removes outliers during both pre-training and adaptation. This outlier reduction in GERM provides threefold benefits for GFMs: accelerated low-rank adaptation, lower computational costs, and enhanced robustness to post-training quantization. By integrating parameter-efficient methods like QLoRA (Dettmers et al., 2024a) and quantization techniques such as OmniQuant (Shao et al., 2024), our framework achieves efficient adaptation and deployment on resource-constrained hardware with negligible accuracy loss, enabling broader deployment on resource-constrained devices. Furthermore, based on (Hu et al., 2024), we propose GERM-T to address its key limitation: avoiding complete retraining from scratch. GERM-T integrates an outlier-free architecture into pre-trained genomic foundation models and uses small-step continual learning to achieve near-optimal performance efficiently. This design significantly reduces computational overhead while maintaining stable performance.

Contributions. We propose **GERM**, an outlier-free GFM with enhanced quantization robustness and rapid low-rank adaptation. Our contributions are as follows:

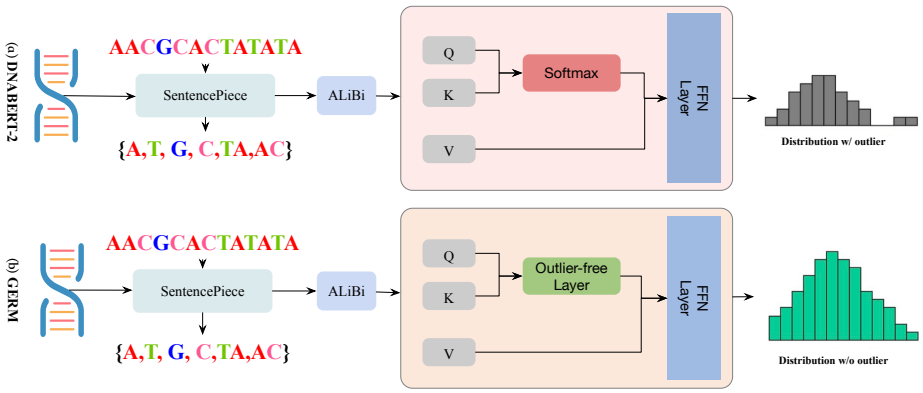

Figure 1: **Structural Comparison of DNABERT-2 and GERM Models.** This diagram compares the processing pipelines of DNABERT-2 and GERM.The key difference lies in the attention mechanism: DNABERT-2 employs a standard Softmaxthat retains outliers, while GERM replaces it with an outlier-free layer, effectively removing outliers from the attention output.

- We propose an outlier-free model structure to address and mitigate outliers introduced by pretrain and low-rank adaptation. This approach enables rapid low-rank adaptation and robust post-training quantization, significantly enhancing the overall performance of the quantized model and model finetuning. Notably, our model finetunes DNABERT in just 5 minutes on a single NVIDIA GeForce RTX 2080 Ti GPU, achieving **14.3%** acceleration in quantization, **34.9%** faster training during finetuning, and **24.79%** inference speedup over baselines.

- Methodologically, we replace the standard transformer attention mechanism in the GFM with an outlier-free layer to enhance the model's ability to handle and mitigate outliers during pretraining and fine-tuning. Additionally, we introduce a continual learning strategy as a compromise version to avoid retraining the model from scratch. This strategy ensures suboptimal performance in terms of model quantization robustness and low-rank adaptation.

- Experimentally, We evaluate the performance and efficiency of our method using the existing DNABERT-2 model (Zhou et al., 2024) structure. Additionally, we benchmark it against the state-of-the-art low-rank adaptation methods and post-training quantization techniques. Compared to the standard framework, the proposed framework achieves average performance improvements of **37.98%** in finetuning and **64.34%** in quantization, respectively. Additionally, GERM shows a reduction of **92.14%** in the average kurtosis and **82.77%** in the maximum infinity norm on average.

2 GERM

The proposed methodology is structured around three core components: an outlier-free architectural design, a small-step continual learning strategy, and DNA GFM implementation. The outlier-free architecture specifically targets outlier-related challenges, while the small-step continual learning strategy builds upon initial checkpoints by incorporating outlier removal mechanism to reduce outliers.

Our GFM development focuses on DNA sequence modeling using Transformer-based architectures, including DNABERT (Ji et al., 2021a), which inherently support integration of techniques like LoRA and our proposed outlier removal mechanism. The baseline implementation adopts DNABERT-2 as the reference architecture, with the redesigned outlier-free structure visualized in Figure 1.

Outliers Challenge in Transformer Architecture. Studies by Hu et al. (2024); Bondarenko et al. (2024) highlight the underlying cause of the outlier challenge in transformer-based models, proposing that transformers do not require updates when the attention inputs are sufficiently informative. However, the normalization nature of the $\mathrm{Softmax}$ function forces non-zero attention weights even for irrelevant tokens, creating numerical instability. Such outliers distort gradient updates and hinder model performance. Numerous studies address the outlier problem across different model stages, including pre-training (Hu et al., 2024), fine-tuning (Hu et al., 2025), and inference (Bondarenko et al., 2024; Xiao et al., 2023). In this paper, we follow the approach proposed by (Luo et al., 2025a), which tackles the outlier problem in GFMs employing the memory-associated retrieval dynamics function $\mathrm{Softmax}_1$ which is defined as

$$\mathrm{Softmax}_1(S) := \frac{\exp(S)}{1 + \sum_{i=1}^{L} \exp(S_i)},$$

where S is the input to the activation function.

Small-step Continual Learning. The outlier mitigation strategy in `OutEffHop` (Hu et al., 2024) effectively suppresses outlier impacts during pretraining. However, this method requires retraining from scratch, which is computationally prohibitive for large-scale models like GFMs. To overcome this limitation, we propose GERM-T, a small-step continual learning framework that extends the GERM architecture. This approach lowers retraining costs by leveraging

Table 1: **Comparing GERM and GERM-T with DNABERT-2 in a Low-Rank Adaptation Setting.** We evaluate GERM against baselines using three low-rank adaptation methods (LoRA, QLoRA, LoftQ), measuring performance via MCC, Delta MCC, *average kurtosis*, and FP16 outlier magnitude *maximum infinity norm* $\|\mathbf{x}\|_\infty$. Best results are bolded, second-best underlined.

Models	Low-Rank Adaptation Method	MCC (↑)	Delta MCC different (↓)	Avg Performance Drop (↓)	Avg. kurtosis(↓)	Max inf. norm(↓)
DNA BERT-2	Full	59.11	7.00	-	270.90	61.41
	LoRA	50.91±1.67	15.2	13.87%	-	219.20
	QLoRA	50.65±0.13	15.46	14.31%	292.85	<u>53.91</u>
	LoftQ	50.76±0.06	15.31	14.05%	299.18	54.18
GERM	Full	59.73	6.38	-	**21.29**	**10.62**
	LoRA	57.27±0.70	8.84	**4.12%**	-	**19.41**
	QLoRA	53.16±0.21	12.95	**10.99%**	**34.29**	**27.27**
	LoftQ	53.11±0.08	13.00	**11.08%**	**33.02**	**27.41**
GERM-T	Full	59.30	6.81	-	<u>251.40</u>	<u>28.49</u>
	LoRA	55.60±0.28	10.51	<u>6.23%</u>	-	<u>140.86</u>
	QLoRA	51.05±0.07	15.06	<u>13.90%</u>	<u>287.95</u>	53.92
	LoftQ	51.20±0.13	14.91	<u>13.65%</u>	<u>286.16</u>	<u>53.35</u>

the original checkpoint and reduces outliers in pre-trained models through outlier removal mechanism.

DNA Genomic Foundation Model. As a representative example, we construct GERM on DNABERT-2 (Zhou et al., 2024). First, we preprocess DNA sequences using SentencePiece (Kudo & Richardson, 2018) with Byte Pair Encoding (BPE), a subword tokenization approach, and set the vocabulary size to 4096 to balance performance and efficiency.

The model follows BERT (Kenton & Toutanova, 2019) with integrated Attention with Linear Biases (ALiBi) (Press et al., 2022). This enables inherent positional learning from sequence order during pretraining and allows the model to handle sequences of arbitrary lengths during downstream tasks, even trained on shorter segments initially.

3 Experimental Studies

In this section, we evaluate the effectiveness of our method through experiments comparing against DNABERT-2.

Models. Following Zhou et al. (2024), we validate our strategy using the DNABERT-2 model [1]. The model is pretrained via masked language modeling (MLM), with all from-scratch training spanning 200K steps. For small-step continual learning, we first train the standard DNABERT-2 from scratch, then switch to the outlier-free structure for the remaining steps. In experiments, we use the GERM-T variant trained with 40K continual learning steps as the example for comparison against DNABERT-2 and GERM.

Datasets. We employ the GUE benchmark in (Zhou et al., 2024), which comprises 27 datasets covering 7 tasks across 4 species, with input lengths ranging from 70 to 1000.

[1] https://huggingface.co/zhihan1996

Evaluation Metrics. To evaluate the performance of outliers in our strategy, we report the *maximum infinity norm* $\|\mathbf{x}\|_\infty$ of the activation tensors $\mathbf{x}$ across all transformer layers as a metric for detecting outliers. Additionally, we present the *average kurtosis* of $\mathbf{x}$, calculated only from the output tensors from the Feed-Forward Network (FFN) layer and Layer Normalization. For pre-quantization performance, we also report the **FP16** (16-bit floating-point) Matthews correlation coefficient (MCC) score to assess model's downstream classification ability.

3.1 Post-Training Quantization (PTQ)

We assess our method's Post-Training Quantization (PTQ) efficiency by replacing DNABERT-2's attention layer with the $\mathrm{Softmax}_1$ activation function. Using pre-trained checkpoints, we conduct full-rank fine-tuning as in (Zhou et al., 2024), then evaluate test performance under FP16 precision and apply PTQ to measure quantization-induced accuracy drops. All experiments repeat three times with different seeds, reporting mean ± standard deviation. For baselines, we compare against the official DNABERT-2 model. Evaluations include four PTQ methods : Traditional W8A8 (Bondarenko et al., 2024), SmoothQuant (Xiao et al., 2023), Outlier Suppression (Wei et al., 2022), and OmniQuant (Shao et al., 2024), tested at W8A8, W6A6, and W4A4 precision levels (except W8A8-only methods). Hyperparameters follow original studies to ensure standardized comparisons.

Results. As shown in Table 3, GERM consistently outperforms DNABERT-2 under W4A4, W6A6, and W8A8 post-training quantization with advanced PTQ methods. For W8A8 quantization, GERM achieves a minimal average performance drop of 4.82% with SmoothQuant. At W4A4, GERM retains 17.17% average accuracy loss under OmniQuant, compared to DNABERT-2's 94.78% drop, demon-

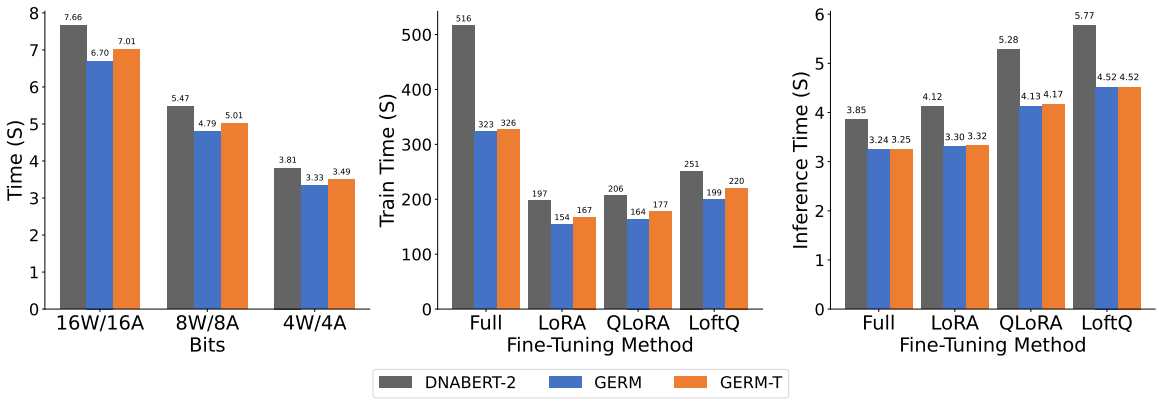

Figure 2: **Comparison of Performance in Resource-Constrained Computing Environments.** Comparison of three models on the quantization and fine-tuning task. The training time represents the average time per epoch.

strating superior quantization robustness. While GERM-T further improves quantization efficiency at 8 and 6-bit levels, it shows a larger W4A4 degradation due to residual outliers in GERM-T. Outlier metrics reveal GERM reduces average kurtosis by 92.14% and maximum infinity norm by 82.77%, while GERM-T achieves 7.20% and 53.78% reductions, respectively. This highlights GERM 's effectiveness in mitigating outliers and enhancing quantization stability.

3.2 Low-Rank Adaptation

Fine-tuning large models is costly, so parameter-efficient methods like LoRA are widely adopted. We compare our method with the standard DNABERT-2 architecture across three parameter-efficient fine-tuning (PEFT) approaches: LoRA (Hu et al., 2022), QLoRA (Dettmers et al., 2024a), and LoftQ (Li et al., 2023). For full fine-tuning, we train the model at full rank using mixed-precision FP16. For LoRA, we apply low-rank adaptation with rank $r = 128$ and scaling factor $\alpha = 256$ following (Hu et al., 2022). QLoRA and LoftQ extend this with 4-bit quantized low-rank updates under identical rank and alpha settings, as described in (Dettmers et al., 2024a). All experiments repeat three times with different seeds, reporting mean $\pm$ standard deviation.

Results. As shown in Table 1, GERM significantly improves low-rank adaptation performance, achieving an average gain of **37.98%** over DNABERT-2. GERM-T further enhances adaptation efficiency with a **20.01%** improvement.

3.3 Performance in Resource-Constrained Devices.

In this section, we conduct case studies to evaluate the effectiveness of our method in resource-constrained devices.

Case Study 1: Performance in Single 2080-Ti GPU Computing Environments. To demonstrate GERM's capability in resource-constrained environments, we conduct performance tests on a single NVIDIA GeForce RTX 2080 Ti 11GB GPU. We provide the per-epoch training time and inference time for the LoRA, QLoRA, and LoftQ fine-tuning methods. The results, as shown in Figure 2, show that GERM achieves **34.9%** faster training during fine-tuning, and **24.79%** inference speedup compared to DNABERT-2, while GERM-T achieves **26.7%** and **24.2%** acceleration, respectively. Furthermore, GERM and GERM-T reduce quantization latency by **14.31%** and **9.21%**, copmare to DNABERT-2. Further details are provided in Appendix C.2.

Case Study 2: Performance in CPU-Only Computing Environments. To demonstrate GERM's capability in CPU-only computing environments, we perform performance tests on CPU-only devices. We compare GERM's per-epoch training and inference times for the LoRA and QLoRA fine-tuning methods. The results, presented in Appendix C.2, indicate that both GERM and GERM-T achieve shorter fine-tuning times per epoch compared to DNABERT-2.

4 Discussion and Conclusion

We introduce GERM, an efficient genomic foundation model designed for limited computational resources. By replacing standard attention layers with outlier-free components, GERM eliminates outliers throughout both pretraining and fine-tuning while enabling robust quantization and low-rank adaptation. Compared with DNABERT-2, GERM reduces *average kurtosis* by $\sim$92.14% and the *maximum infinity norm* by $\sim$82.77% across 27 datasets; it also achieves 14.3% acceleration in quantization, 34.9% faster training during fine-tuning, and 24.79% inference speed-up over baseline models. Overall, quantization robustness improves by lowering the average performance drop by 64.34% and low-rank adaptation efficiency rises by cutting the performance drop by 37.98%. The compromise variant GERM-T achieves 31.42% improvement in quantization robustness and 20.01% gain in low-rank adaptation, balancing performance with reduced retraining costs through continual learning.

Impact Statement

We believe this methodology presents an opportunity to strengthen the core of foundation models, including large language models, by improving robustness through quantization and enabling faster low-rank adaptation. However, this approach may also amplify biases in the training data, potentially leading to unfair or discriminatory outcomes for underrepresented groups.

Acknowledgments

The authors would like to thank Yegna Jambunath for enlightening discussions on related topics, and Jiayi Wang for facilitating experimental deployments. The authors would like to thank the anonymous reviewers and program chairs for constructive comments.

Han Liu is partially supported by NIH R01LM1372201, NSF AST-2421845, Simons Foundation MPS-AI-00010513, AbbVie and Dolby. Haozheng Luo is partially supported by the OpenAI Researcher Access Program. This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

References

Ahmadian, A., Dash, S., Chen, H., Venkitesh, B., Gou, Z. S., Blunsom, P., Üstün, A., and Hooker, S. Intriguing properties of quantization at scale. *Advances in Neural Information Processing Systems*, 36:34278–34294, 2023.

Bondarenko, Y., Nagel, M., and Blankevoort, T. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7947–7969, 2021.

Bondarenko, Y., Nagel, M., and Blankevoort, T. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chee, J., Cai, Y., Kuleshov, V., and Sa, C. D. QuIP: 2-bit quantization of large language models with guarantees. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276. Association for Computational Linguistics, 2019.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Lopez Carranza, N., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, pp. 1–11, 2024.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024a.

Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T., and Alistarh, D. Spqr: A sparse-quantized representation for near-lossless llm weight compression. In *ICLR*, 2024b.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

He, H., Luo, H., and Wang, Q. R. St-moe-bert: A spatial-temporal mixture-of-experts framework for long-term cross-city mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, pp. 10–15, 2024.

Heo, J. H., Kim, J., Kwon, B., Kim, B., Kwon, S. J., and Lee, D. Rethinking channel dimensions to isolate outliers for low-bit weight quantization of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hu, J. Y.-C., Chang, P.-H., Luo, H., Chen, H.-Y., Li, W., Wang, W.-P., and Liu, H. Outlier-efficient hopfield layers for large transformer-based models. In *Forty-first International Conference on Machine Learning*, 2024.

Hu, J. Y.-C., Su, M., jui kuo, E., Song, Z., and Liu, H. Computational limits of low-rank adaptation (loRA) fine-tuning for transformer models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. 2021a.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021b.

Jiang, E. H., Luo, H., Pang, S., Li, X., Qi, Z., Li, H., Yang, C.-F., Lin, Z., Li, X., Xu, H., et al. Learning to rank chain-of-thought: An energy-based approach with outcome supervision. *arXiv preprint arXiv:2505.14999*, 2025.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2. Minneapolis, Minnesota, 2019.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. Revealing the dark secrets of bert. 2019.

Kovaleva, O., Kulshreshtha, S., Rogers, A., and Rumshisky, A. BERT busters: Outlier dimensions that disrupt transformers. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3392–3405, Online, August 2021. Association for Computational Linguistics.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

Lee, C., Jin, J., Kim, T., Kim, H., and Park, E. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13355–13364, 2024.

Li, Y., Yu, Y., Liang, C., He, P., Karampatziakis, N., Chen, W., and Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models, 2023.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. In *MLSys*, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Luo, H., Liu, N., and Feng, C. Question and answer classification with deep contextualized transformer. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pp. 453–461. Springer, 2021a.

Luo, H., Qin, R., Xu, C., Ye, G., and Luo, Z. Open-ended multi-modal relational reasoning for video question answering. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 363–369. IEEE, 2023.

Luo, H., Yu, J., Zhang, W., Li, J., Hu, J. Y.-C., Xing, X., and Liu, H. Decoupled alignment for robust plug-and-play adaptation. *arXiv preprint arXiv:2406.01514*, 2024.

Luo, H., Qiu, C., Su, M., Zhou, Z., Mehta, Z., Ye, G., Hu, J. Y.-C., and Liu, H. Fast and low-cost genomic foundation models via outlier removal. In *Forty-second International Conference on Machine Learning*, 2025a.

Luo, H., Qiu, C., Wang, Y., Wu, S., Yu, J., Liu, H., Wang, B., and Chen, Y. Genoarmory: A unified evaluation framework for adversarial attacks on genomic foundation models. *arXiv preprint arXiv:2505.10983*, 2025b.

Luo, Z., Kulmizev, A., and Mao, X. Positional artefacts propagate through masked language model embeddings. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5312–5327, Online, August 2021b. Association for Computational Linguistics.

Ma, Y., Li, H., Zheng, X., Ling, F., Xiao, X., Wang, R., Wen, S., Chao, F., and Ji, R. Outlier-aware slicing for post-training quantization in vision transformer. In *Forty-first International Conference on Machine Learning*, 2024.

Maiti, S., Peng, Y., Choi, S., Jung, J.-w., Chang, X., and Watanabe, S. Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13326–13330. IEEE, 2024.

Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., et al. Gemma: Open models based on gemini research and technology. *CoRR*, 2024.

Nguyen, E., Poli, M., Durrant, M. G., Kang, B., Katrekar, D., Li, D. B., Bartie, L. J., Thomas, A. W., King, S. H., Brixi, G., et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723): eado9336, 2024a.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024b.

Pan, Z., Luo, H., Li, M., and Liu, H. Conv-coa: Improving open-domain question answering in large language models via conversational chain-of-action. *arXiv preprint arXiv:2405.17822*, 2024.

Pan, Z., Luo, H., Li, M., and Liu, H. Chain-of-action: Faithful and multimodal question answering through large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Press, O., Smith, N., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Puccetti, G., Rogers, A., Drozd, A., and Dell'Orletta, F. Outlier dimensions that disrupt transformers are driven by frequency. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1286–1304, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Qin, R. *Hardware and Algorithm Co-Exploration for Efficient On-Device Personalization of Large Language Models*. PhD thesis, University of Notre Dame, 2025.

Qin, R., Hu, Y., Yan, Z., Xiong, J., Abbasi, A., and Shi, Y. Fl-nas: Towards fairness of nas for resource constrained devices via large language models. In *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 429–434. IEEE, 2024a.

Qin, R., Liu, D., Xu, C., Yan, Z., Tan, Z., Jia, Z., Nassereldine, A., Li, J., Jiang, M., Abbasi, A., et al. Empirical guidelines for deploying llms onto resource-constrained edge devices. *ACM Transactions on Design Automation of Electronic Systems*, 2024b.

Qin, R., Liu, D., Xu, G., Yan, Z., Xu, C., Hu, Y., Hu, X. S., Xiong, J., and Shi, Y. Tiny-align: Bridging automatic speech recognition and large language model on the edge. *arXiv preprint arXiv:2411.13766*, 2024c.

Qin, R., Xia, J., Jia, Z., Jiang, M., Abbasi, A., Zhou, P., Hu, J., and Shi, Y. Enabling on-device large language model personalization with self-supervised data selection and synthesis. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024d.

Qin, R., Yan, Z., Zeng, D., Jia, Z., Liu, D., Liu, J., Abbasi, A., Zheng, Z., Cao, N., Ni, K., et al. Robust implementation of retrieval-augmented generation on edge-based computing-in-memory architectures. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2024e.

Qin, R., Ren, P., Yan, Z., Liu, L., Liu, D., Nassereldine, A., Xiong, J., Ni, K., Hu, S., and Shi, Y. Nvcim-pt: An nvcim-assisted prompt tuning framework for edge llms. In *2025 Design, Automation & Test in Europe Conference (DATE)*, pp. 1–7. IEEE, 2025.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., and Luo, P. Omniquant: Omnidirectionally calibrated quantization for large language models. In *ICLR*, 2024.

Sun, C.-E., Yan, G., and Weng, T.-W. Thinkedit: Interpretable weight editing to mitigate overly short thinking in reasoning models. *arXiv preprint arXiv:2503.22048*, 2025.

Sun, M., Chen, X., Kolter, J. Z., and Liu, Z. Massive activations in large language models. 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, N., Yang, H., and Wang, C. D. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets, 2023.

Wei, X., Zhang, Y., Zhang, X., Gong, R., Zhang, S., Zhang, Q., Yu, F., and Liu, X. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414, 2022.

Wei, X., Zhang, Y., Li, Y., Zhang, X., Gong, R., Guo, J., and Liu, X. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1648–1665, Singapore, December 2023. Association for Computational Linguistics.

Wu, S., Lu, Y.-J., Luo, H., Su, M., Hu, J. Y.-C., Wang, J., Liu, J., Dehak, N., Villalba, J., and Liu, H. Sparq: Outlier-free speechlm with fast adaptation and robust quantization.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. Bloomberggpt: A large language model for finance, 2023.

Wu, X., He, H., Wang, Y., and Wang, Q. Pretrained mobility transformer: A foundation model for human mobility. *arXiv preprint arXiv:2406.02578*, 2024.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. 2024.

Yu, J., Wu, Y., Shu, D., Jin, M., and Xing, X. Assessing prompt injection risks in 200+ custom gpts. *arXiv preprint arXiv:2311.11538*, 2023.

Yu, J., Luo, H., Hu, J. Y.-C., Guo, W., Liu, H., and Xing, X. Enhancing jailbreak attack against large language models through silent tokens. *arXiv preprint arXiv:2405.20653*, 2024.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhang, Y., Mei, D., Luo, H., Xu, C., and Tsai, R. T.-H. Smutf: Schema matching using generative tags and hybrid features. *Information Systems*, pp. 102570, 2025.

Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., and Liu, H. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R. V., and Liu, H. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhou, Z., Wu, W., Ho, H., Wang, J., Shi, L., Davuluri, R. V., Wang, Z., and Liu, H. DNABERT-s: Pioneering species differentiation with species-aware DNA embeddings. 2025a.

Zhou, Z., Wu, W., Wu, J., Shi, L., Wang, Z., and Liu, H. Genomeocean: Efficient foundation model for genome generation, 2025b.

Supplementary Material

A Related Work

Quantization. Considering the quantized object, exiting foundation models (FMs) quantization can be classified into two fields: weight-only quantization and weight-activation quantization. For **weight-only quantization**, prior studies focus on converting weights to low-bit values. For instance, GPTQ (Frantar et al., 2023) uses block-wise reconstruction for 3/4-bit quantization. SpQR (Dettmers et al., 2024b), OWQ (Lee et al., 2024), and AWQ (Lin et al., 2024) emphasize the significance of weights tied to higher-magnitude activations. Therefore, SpQR and OWQ employ mixed-precision quantization to safeguard vital weights, while AWQ opts for channel-wise scaling to avoid mixed-precision's hardware inefficiency. QLoRA (Dettmers et al., 2024a), LoftQ (Li et al., 2023) and QUIP (Chee et al., 2023) restore the capabilities of the quantized model through parameter-efficient fine-tuning. For **weight-activation quantization**, prior studies compress both weights and activations. SmoothQuant (Xiao et al., 2023), LLM.int8() (Dettmers et al., 2022), and Outlier Suppression (Wei et al., 2022) achieve W8A8 quantization by managing activation outliers. LLM.int8() uses mixed-precision decomposition, while the other two employ channel-wise scaling. Furthermore, Outlier Suppression+ (Wei et al., 2023) adds channel-wise shifting to drive W6A6 quantization. In comparison to other quantization approaches, including prior works (Wei et al., 2023; Xiao et al., 2023) that address the outlier issue during quantization, the outlier-free layer in GERM is more effective at managing outliers within the model's attention mechanism. It provides GERM with a unique advantage in terms of quantization robustness.

Outlier Values in Quantization. Numerous studies (Hu et al., 2024; Ma et al., 2024; Heo et al., 2024; Puccetti et al., 2022; Kovaleva et al., 2021; Bondarenko et al., 2021; Luo et al., 2021b) observe outlier values in the transformer-based language models such as BERT (Devlin et al., 2019) and early GPT (Radford et al., 2019) models. Since the advent of FMs (Zhou et al., 2024; 2025a; Zhang et al., 2022; Brown et al., 2020) root in the GPT and BERT, recent studies by Xiao et al. (2023); Ahmadian et al. (2023); Dettmers et al. (2022) tackle the existence of outlier values in FMs. According to them, these outliers exhibit a large magnitude of values at the shared dimensions of hidden states across tokens. More recently, Bondarenko et al. (2024); Sun et al. (2024); Hu et al. (2024) explain that the outliers attribute to the vertical pattern in the attention mechanism (Xiao et al., 2024; Kovaleva et al., 2019), influencing the performance of FMs. In particular, Sun et al. (2024) claim a different type of outlier existing in the hidden states of specific tokens. However, most of these studies concentrate on language and vision models, leaving the impact of outliers on genomic foundation models largely unexplored. Additionally, methods like Hu et al. (2024) require training from scratch to eliminate outliers, which is computationally expensive.

Genomic Foundation Model. The majority of genomic foundation models (GFMs) use transformers to model sequence dependencies, similar to BERT (Devlin et al., 2019) and GPT (Brown et al., 2020) in NLP. Specifically, DNABERT (Ji et al., 2021a) and DNABERT-2 (Zhou et al., 2024) leverage transformers for DNA sequence analysis by employing masked language modeling and fine-tuning for biological tasks. In addition, Nucleotide Transformer (Dalla-Torre et al., 2024) excels at molecular phenotype prediction and variant prioritization, while HyenaDNA (Nguyen et al., 2024b) is optimized for modeling long-range genomic dependencies. Furthermore, GenomeOcean (Zhou et al., 2025b) provides an efficient 4-billion-parameter genome foundation model for diverse, context-aware DNA sequence generation. However, these models demand significant computational resources and lack robustness to quantization, rendering them unsuitable for deployment

on resource-constrained devices. Specifically, GenomeOcean utilizes 64 NVIDIA A100 80G GPUs over a span of 14 days for training. This limits accessibility for research labs with limited computational capacity. More recently, Evo (Nguyen et al., 2024a), a generative genomic model, integrating Transformer and Hyena operator to efficiently capture long-range dependencies in genomic sequences, achieving a context window of 131k nucleotides. Furthermore, Evo uniquely bridges bridges the DNA-RNA-protein central dogma via cross-modal inference without task-specific supervision.

Transformer-Based Foundation Models. Transformer-based foundation models have catalyzed progress across AI subfields, including question answering (Pan et al., 2024; Luo et al., 2021a), reasoning (Jiang et al., 2025; Pan et al., 2025; Sun et al., 2025), safety (Luo et al., 2025b; 2024; Yu et al., 2024; 2023), multi-modality (Luo et al., 2023; Radford et al., 2021), edge computing (Qin, 2025; Qin et al., 2025; 2024b;a;d;e), and data cleaning (Zhang et al., 2025). They serve as key enablers across application domains such as NLP (Guo et al., 2025; Mesnard et al., 2024), speech (Wu et al.; Maiti et al., 2024; Qin et al., 2024c), finance (Wang et al., 2023; Wu et al., 2023), genomics (Nguyen et al., 2024a; Zhou et al., 2025b;a; 2023; Ji et al., 2021b), and human mobility (Wu et al., 2024; He et al., 2024).

B Experimental Setup

B.1 Computational Resource

We perform all experiments using 2 NVIDIA A100 GPU with 80GB of memory and a 24-core Intel(R) Xeon(R) Gold 6338 CPU operating at 2.00GHz. Our code is developed in PyTorch and utilizes the Hugging Face Transformer Library for experimental execution.

B.2 Hyperparameters

We present the hyperparameters used in the fine-tuning stage for each model. We use **AdamW** (Loshchilov & Hutter, 2019) as the optimizer. Most of the other hyperparameters remain the same across all models and datasets, including a batch size of 32, a warmup step of 50, and a weight decay of 0.01. A learning rate of $3e^{-5}$ is used for all models during fine-tuning. For low-rank adaptation, we use a learning rate of $1e^{-4}$, with a LoRA rank of 8 and LoRA alpha set to 16. For each task, we use different training steps as shown in Table 2. During pre-training, the model is trained for 200,000 steps with a batch size of 1024 and a maximum sequence length of 512, using the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-6}$. The pre-training stage takes approximately 4 days using 2 NVIDIA A100 80G GPUs.

Table 2: **The number of training steps.** We present the number of training steps we use in our experiments. In the task of Transcription Factor Prediction on the Mouse genome, we train the model for 1000 steps on each dataset.

	EMP	TF-M	CVC	TF-H	PD-tata	PD-o	CPD-tata	CPD-o	SSP
Epochs	3	1k	8	3	10	4	10	4	5

C Additional Numerical Experiments

C.1 All Results of Performance Comparison in Post-Training Quantisation (PTQ) setting

Table 3: **Comparing GERM and GERM-T with DNABERT-2 in a Post-Training Quantisation (PTQ) setting.** We compare GERM against baselines across four quantization methods (Traditional W8A8, SmoothQuant, Outlier Suppression, OmniQuant) and three configurations (W8A8, W6A6, W4A4). Metrics include Matthews Correlation Coefficient (MCC), Delta MCC relative to DNABERT-2, *average kurtosis*, and FP16 outlier magnitude via *maximum infinity norm* $\|\mathbf{x}\|_\infty$. Best results are bolded, second-best underlined.

Model	#Bits	Quantization Method	MCC (↑)	Delta MCC (↓)	Avg Performance Drop (↓)	Avg. Kurtosis (↓)	Max inf. norm (↓)
Official	16W/16A	-	66.11	-	-	<u>39.68</u>	53.61
DNABERT-2	16W/16A	-	59.11	7.00	-	270.90	61.64
	8W/8A		33.60±0.41	32.51	43.81%		
	8W/8A	SmoothQuant	36.51±0.02	45.37	38.63%		
	6W/6A		20.74±0.04	45.37	66.18%		
	4W/4A		-1.03±0.06	67.06	101.24%		
	8W/8A	Outlier	25.26±0.02	40.85	57.60%		
	6W/6A		27.84±0.28	38.27	52.71%		
	8W/8A	OmniQuant	49.92±0.05	16.19	15.76%		
	6W/6A		48.47±0.14	17.64	18.61%		
	4W/4A		2.94±0.19	63.17	94.78%		
GERM	16W/16A	-	59.73	6.38	-	**21.29**	**10.62**
	8W/8A		57.30±0.08	8.81	**3.77%**		
	8W/8A	SmoothQuant	56.65±0.15	9.46	<u>4.82%</u>		
	6W/6A		56.48±0.07	9.63	**5.45%**		
	4W/4A		20.05±0.00	46.06	**69.44%**		
	8W/8A	Outlier	45.87±0.08	20.24	**25.23%**		
	6W/6A		40.57±0.56	25.54	<u>36.27%</u>		
	8W/8A	OmniQuant	55.99±0.09	10.12	<u>5.95%</u>		
	6W/6A		55.70±0.03	10.41	**6.41%**		
	4W/4A		49.42±0.00	16.69	**17.17%**		
GERM-T	16W/16A	-	59.30	6.81	-	251.40	<u>28.49</u>
	8W/8A		38.38±0.15	27.73	<u>35.27%</u>		
	8W/8A	SmoothQuant	57.52±0.00	8.59	**3.01%**		
	6W/6A		30.34±0.04	35.77	<u>48.83%</u>		
	4W/4A		0.22±0.00	65.89	<u>99.63%</u>		
	8W/8A	Outlier	42.57±0.05	23.54	<u>28.31%</u>		
	6W/6A		46.02±0.06	20.06	**22.34%**		
	8W/8A	OmniQuant	56.80±0.12	9.31	**4.21%**		
	6W/6A		55.41±0.00	10.71	<u>6.57%</u>		
	4W/4A		3.86±0.00	62.25	<u>93.49%</u>		

C.2 All Results of Performance Comparison in Resource-Constrained Computing Environments

In this section, we present the results of Performance Comparison in Resource-Constrained Computing Environments.

Case Study 1: Performance in CPU-only Computing Environments. All models were trained on the same computing infrastructure (Nvidia GeForce RTX 2080 TI 11GB) to ensure a fair comparison. The training time represents the average time per epoch, with OmniQuant used as quantization example.

GERM demonstrates superior adaptability and performance in resource-constrained computing environments compared to DNABERT-2 and GERM-T. Its consistent high MCC scores and reduced training and inference times across various quantization levels and fine-tuning methods establish GERM as the most robust and efficient model, with GERM-T following as a commendable second-best option. These attributes make GERM a promising candidate for further research and application in settings demanding both high performance and computational efficiency.

Table 4: **Comparison of Performance in Resource-Constrained Computing Environments.** Comparison of three models on the quantization and fine-tuning task.

Method	#Bits	MCC (↑)	Time (sec.)
DNABERT-2	16W/16A	59.11	7.66
GERM	16W/16A	**59.73**	**6.70**
GERM-T	16W/16A	<u>59.30</u>	<u>7.01</u>
DNABERT-2	8W/8A	49.92	5.47
GERM	8W/8A	**55.99**	**4.79**
GERM-T	8W/8A	<u>56.80</u>	<u>5.01</u>
DNABERT-2	4W/4A	-1.03	3.81
GERM	4W/4A	**20.05**	**3.33**
GERM-T	4W/4A	<u>0.22</u>	<u>3.49</u>

Method	Fine-Tuning Method	MCC (↑)	Time (sec.)	
			Train	Inference
DNABERT-2	Full	59.11	516.49	3.85
GERM	Full	**59.73**	**323.10**	**3.24**
GERM-T	Full	<u>59.30</u>	<u>326.91</u>	<u>3.25</u>
DNABERT-2	LoRA	50.91	197.13	4.12
GERM	LoRA	**57.27**	**154.67**	**3.30**
GERM-T	LoRA	<u>55.60</u>	<u>167.76</u>	<u>3.32</u>
DNABERT-2	QLoRA	50.65	206.15	5.28
GERM	QLoRA	**53.16**	**164.10**	**4.13**
GERM-T	QLoRA	<u>51.50</u>	<u>177.95</u>	<u>4.17</u>
DNABERT-2	LoftQ	50.76	251.37	5.77
GERM	LoftQ	**53.11**	**199.58**	**4.52**
GERM-T	LoftQ	<u>51.20</u>	<u>220.37</u>	<u>4.52</u>

Case Study 2: Performance in CPU-only Computing Environments. To demonstrate GERM's capability in CPU-only computing environments, we perform performance tests on an 64-core Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz with 50GB RAM. We compare GERM's per-epoch training and inference times for the LoRA and QLoRA fine-tuning methods. The results, presented in Table 5, indicate that both GERM and GERM-T achieve shorter fine-tuning times per epoch compared to DNABERT-2, with the only exception being QLoRA when deployed, where the time is slightly longer. QLoRA can be slower than LoRA during inference and fine-tuning due to hardware limitations when bf16 (bfloat16) support is unavailable. QLoRA relies on ultra-low-precision quantization (e.g., 4-bit weights) to reduce memory usage and increase efficiency, which works best on systems that support bf16 or similar mixed-precision operations. However, without bf16 support, these low-precision operations must be emulated by converting back to higher precision, introducing computational overhead. This diminishes the intended speed advantage of QLoRA, potentially making it slower than LoRA on incompatible hardware.

Table 5: **Comparison of Performance in CPU-only Computing Environments.** Comparison of three models on the fine-tuning task.

Method	Fine-Tuning Method	MCC (↑)	Time (sec.)	
			Train	Inference
DNABERT-2	LoRA	50.91	808.23	29.66
GERM	LoRA	**57.27**	**618.68**	**23.10**
GERM-T	LoRA	<u>55.60</u>	<u>674.40</u>	<u>23.57</u>
DNABERT-2	QLoRA	50.65	516.04	63.17
GERM	QLoRA	**53.16**	**358.34**	**45.28**
GERM-T	QLoRA	<u>51.50</u>	<u>418.13</u>	<u>46.91</u>

C.3 Performance of GERM on Alternative Transformer-based Models

In this section, we conduct our experiment to validate the effectiveness of the outlier removal approach using alternative transformer-based models, evaluating performance through Matthews Correlation Coefficient (MCC) and average performance drop. We use the NT-500M-human[2] as the target model for our evaluation. Table 6 compares these metrics across NT-500M-human, GERM, and GERM-T models using different low-rank adaptation methods. Table 7 examines the impact of various quantization techniques on the same models. The results demonstrate the effectiveness of outlier removal across diverse adaptation and quantization strategies, highlighting the balance between performance and resource efficiency.

Table 6: **Low-Rank Adaptation Methods Comparison.** This comparison evaluates the performance of different low-rank adaptation methods, including Full, LoRA, QLoRA, and LoftQ, on Nucleotide Transformer 500M models. The best results are highlighted in bold, while the second-best results are underlined.

Model	Fine-Tuning Method	MCC	Delta MCC	Average Performance Drop
NT-500M-human	Full	56.05	-	-
	LoRA	52.66	3.39	6.44%
	QLoRA	51.46	4.59	8.19%
	LoftQ	51.89	4.16	7.42%
GERM (NT-500M-human)	Full	55.52	0.53	-
	LoRA	54.32	1.73	**2.16%**
	QLoRA	53.78	2.27	**3.13%**
	LoftQ	54.24	1.81	**2.30%**
GERM-T (NT-500M-human)	Full	56.53	-0.48	-
	LoRA	54.89	1.16	<u>2.90%</u>
	QLoRA	52.78	3.27	<u>6.63%</u>
	LoftQ	53.45	2.60	<u>5.45%</u>

[2] https://huggingface.co/InstaDeepAI/nucleotide-transformer-500m-human-ref

Table 7: **Quantization Methods Comparison.** This comparison analyzes the performance of various quantization methods, including FP16, W8A8, Outlier, SmoothQuant, and OmniQuant, on Nucleotide Transformer 500M models. The best results are highlighted in bold, while the second-best results are underlined.

Model	#Bits	Quantization Method	MCC	Delta MCC	Average Performance Drop
NT-500M-human	16W/16A	-	56.05	-	-
	8W/8A	-	34.66	21.39	38.17%
	8W/8A	Outlier	32.95	23.10	41.21%
	6W/6A		26.65	29.40	52.45%
	8W/8A	SmoothQuant	38.23	17.82	31.79%
	6W/6A		28.67	27.38	48.84%
	4W/4A		3.54	52.51	93.68%
	8W/8A	OmniQuant	47.35	8.70	15.52%
	6W/6A		43.63	12.42	22.16%
	4W/4A		5.34	50.71	90.47%
GERM (NT-500M-human)	16W/16A	-	55.53	0.52	-
	8W/8A	-	53.67	2.38	**3.35%**
	8W/8A	Outlier	45.71	10.34	**17.68%**
	8W/8A		41.38	14.67	25.48%
	8W/8A	SmoothQuant	53.18	2.87	4.23%
	6W/6A		52.43	3.62	**5.58%**
	4W/4A		24.96	31.09	**55.05%**
	8W/8A	OmniQuant	52.45	3.60	**5.55%**
	6W/6A		51.56	4.49	**7.15%**
	4W/4A		46.45	9.60	**16.35%**
GERM-T (NT-500M-human)	16W/16A	-	56.53	-0.48	-
	8W/8A	-	40.71	15.34	27.99%
	8W/8A	Outlier	45.98	10.07	18.66%
	6W/6A		43.38	12.67	**23.26%**
	8W/8A	SmoothQuant	54.19	1.86	**4.14%**
	6W/6A		38.67	17.38	31.59%
	4W/4A		10.57	45.48	81.29%
	8W/8A	OmniQuant	52.46	3.59	7.20%
	6W/6A		51.34	4.71	9.18%
	4W/4A		23.57	32.48	58.31%

C.4 Performance of GERM on Large-scale GFMs

In this section, we present experiments to validate the effectiveness of GERM on large-scale GFMs. We use the NT-2.5B-multi[3] as the target model for our evaluation. Table 8 compares these metrics across NT-2.5B-multi, GERM, and GERM-T models using different low-rank adaptation methods. Table 9 extends this analysis to evaluate the impact of various quantization techniques on the same models. In the larger-parameter model, we adopt stricter quantization bits. This choice aims to save computation and improve efficiency, as finer compression is crucial when model parameters scale up. Additionally, experiments conducted with a larger-parameter model further validate these findings, demonstrating that outlier removal consistently enhances performance and resource efficiency across diverse adaptation and quantization strategies.

Table 8: **Comparison of Low-Rank Adaptation Methods in Large-Scale Models.** This comparison evaluates the performance of different low-rank adaptation methods, including Full, LoRA, QLoRA, and LoftQ, on Nucleotide Transformer 2.5B models. The best results are highlighted in bold, while the second-best results are underlined.

Model	Fine-Tuning Method	MCC	Delta MCC	Average Performance Drop
NT-2.5B-multi	Full	56.98	-	-
	LoRA	53.50	3.48	6.11%
	QLoRA	52.29	4.69	8.19%
	LoftQ	52.89	4.09	7.17%
GERM (NT-2.5B-multi)	Full	57.16	-0.18	-
	LoRA	55.98	1.18	**2.06%**
	QLoRA	55.52	1.64	**2.87%**
	LoftQ	55.80	1.36	**2.38%**
GERM-T (NT-2.5B-multi)	Full	56.82	0.16	-
	LoRA	55.24	1.58	<u>2.78%</u>
	QLoRA	53.32	3.50	<u>6.16%</u>
	LoftQ	53.74	3.08	<u>5.42%</u>

[3] https://huggingface.co/InstaDeepAI/nucleotide-transformer-2.5b-multi-species

Table 9: **Comparison of Quantization Methods in Large-Scale Models.** This comparison analyzes the performance of various quantization methods, including FP16, W6A6, W4A4, Outlier, SmoothQuant, and OmniQuant, on Nucleotide Transformer 2.5B models. The best results are highlighted in bold, while the second-best results are underlined.

Model	#Bits	Quantization Method	MCC	Delta MCC	Average Performance Drop
NT-2.5B-multi	16W/16A	-	56.98	-	-
	6W/6A	-	18.52	38.46	67.50%
	4W/4A	-	1.39	55.59	97.56%
	6W/6A	Outlier	50.23	6.75	11.85%
	4W/4A		40.74	16.24	28.50%
	6W/6A	SmoothQuant	47.23	9.75	17.11%
	4W/4A		35.16	21.82	38.29%
	6W/6A	OmniQuant	49.55	7.43	13.04%
	4W/4A		43.63	13.35	23.43%
GERM (NT-2.5B-multi)	16W/16A	-	57.16	-0.18	-
	6W/6A	-	45.96	11.2	**19.59%**
	4W/4A	-	42.48	14.68	**25.68%**
	6W/6A	Outlier	52.24	4.92	8.61%
	4W/4A		49.00	8.16	**14.28%**
	6W/6A	SmoothQuant	51.95	5.21	**9.11%**
	4W/4A		48.15	31.09	15.76%
	6W/6A	OmniQuant	52.55	4.61	8.07%
	4W/4A		49.26	7.90	**13.82%**
GERM-T (NT-2.5B-multi)	16W/16A	-	56.82	0.16	-
	6W/6A	-	32.58	24.24	42.66%
	4W/4A	-	10.49	46.33	81.54%
	6W/6A	Outlier	52.14	4.68	**8.24%**
	4W/4A		46.24	10.58	18.62%
	6W/6A	SmoothQuant	51.61	5.21	9.17%
	4W/4A		48.12	8.70	**15.31%**
	6W/6A	OmniQuant	52.43	4.39	**7.73%**
	4W/4A		47.28	9.54	16.79%