MSAMamba: Adapting Subquadratic Models To Long-Context DNA MSA Analysis

Vishrut Thoutam [*1] Dina Ellsworth [1]

Abstract

We introduce MSAMamba, a novel architecture designed to address the context-length limitation of existing transformer-based models for DNA multiple sequence alignments (MSAs). Traditional transformers struggle with the vast context lengths inherent in MSA genome data, mainly due to the quadratic complexity of self-attention at large batch sizes. MSAMamba leverages a selective scan operation along the sequence dimension and separates sequence length and MSA dimension processing to enhance efficiency. This architecture enables scalable analysis of long DNA sequences, increasing the training context length of previous methods by 8x. In addition, we develop a row-sparse training method that significantly reduces the computational overhead of the selective scan. We demonstrate that MSAMamba achieves performance on par with state-of-the-art (SOTA) transformer-based models in variant effect prediction tasks and exceeds their performance at longer context lengths. We also demonstrate that MSAMamba excels in GenomicBenchmarks tasks. Our results indicate that MSAMamba mitigates the computational challenges of long-context DNA MSA analysis and sets a new standard for scalability and efficiency in genomic modeling.

1. Introduction

Advances in model sizes and architectures have brought about a revolution in sequence modeling capabilities. The introduction of recurrence (Jordan, 1986), attention (Bahdanau et al., 2014), and memory (Hochreiter & Schmidhuber, 1997) have led to many performance improvements. The transformer model (Vaswani et al., 2017), commonly used in large language models (LLMs) (Brown et al., 2020), applies self-attention and implicit memory to sequence modeling.

Transformers show impressive generalization capabilities in natural language processing, prompting researchers to extend the models' abilities to sequences beyond language. Transformers have been applied to protein language modeling (Lin et al., 2022) and genomics analysis (Rosen et al., 2023). Recently, they have been used in DNA modeling (Dalla-Torre et al., 2023). However, The human genome consists of 3 billion base pairs, with gene sizes ranging from 10 thousand to 2 million base pairs (Lopes et al., 2021). These large DNA sequences are expensive to analyze using transformers due to the quadratic nature of self-attention (Keles et al., 2022) and instability across extended context windows (Liu et al., 2023). Subquadratic models (Poli et al., 2023) are alternatives to transformers that show high performance in modeling global relationships across long DNA sequences (Nguyen et al., 2023), (Schiff et al., 2024b).

Raw DNA sequences lack explicit evolution and homology information. DNA multiple sequence alignments (MSAs) provide this information (Sofi et al., 2022). Models that operate on MSAs show advances in mutation detection and sequence analysis tasks (Siepel et al., 2005a).

However, current DNA MSA modeling architectures are not scalable to long sequences. Axial attention-based transformers (Ho et al., 2019) are the current state-of-the-art for DNA variant effect prediction using MSAs, but these methods only train on sequences of 128 base pairs (Benegas et al., 2023). Previous methods use this training method because MSA processing using axial attention is more computationally complex than single sequence analysis (see proof A.2).

A less complex algorithm is required to support long context lengths and robust MSA sizes. We introduce MSAMamba: a variant of the MSA Transformer model that replaces axial attention with a horizontal SSM selective scan (Gu & Dao, 2023) and a vertical attention block. We also introduce a row-level masking methodology to improve training efficiency on sparse MSA sequences. These changes decrease the computational complexity of training and fine-tuning at large context lengths, allowing us to train on longer se-

[*] Equal contribution [1] High Technology High School. Correspondence to: Cieua Vvvvv <c.vvvvv@googol.com>, Eee Pppp <ep@eden.co.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

quences efficiently. We find that an SSM-based DNA MSA model performs similarly to SOTA transformer-based MSA models in variant effect prediction at short context lengths (128) and exceeds transformer models when training on longer sequences (1024). Additionally, MSAMamba shows improved performance in 3 out of 8 GenomicBenchmarks tasks compared to single-sequence and transformer-based MSA models (see Table 3.1).

2. Background

This section provides an overview of biological and AI representation concepts used in constructing the MSAMamba model.

2.1. DNA Terminology

Deoxyribonucleic acid is a polymer made up of 4 base nucleotides (adenine, cytosine, guanine, and thymine). The polymer forms a double helix structure from two complementary strands. DNA contains regions known as genes, which can code for different proteins to cause cellular change. Genes also consist of control sequences. These include enhancers, which can increase the DNA transcription of a specific gene into a protein; promoters, which allow the initiation of transcription; and silencers, which prevent transcription from occurring. (Brown, 2012)

DNA sequences also contain introns and exons. Exons contain DNA information used to form the final protein, while introns are non-coding regions that can be spliced out in different combinations to create varying gene outputs. Genes can vary in length from thousands to millions of base pairs, making a sizeable effective context window necessary for analyzing motifs and sequence features within and across genes.

DNA MSAs DNA Multiple Sequence Alignments (MSAs) are combinations of DNA sequences across different species. These sequences are aligned such that base pairs with similar functions are in the same column across genomes. These alignments provide crucial evolutionary information between species. A DNA sequence for a species can be considered as a function of a different species' genome. This function consists of multiple mutations/inverse mutations, such as insertions, deletions, and replacements. By aligning this sequence, models can extract evolution and homology information. DNA MSAs are also used to find motifs (short, repetitive sequences seen across genomes). Implicit detection of these motifs in AI models can provide enhanced information for genome analysis. (Sofi et al., 2022)

2.2. Subquadratic Sequence Models

Recently, variants of state space models (SSMs) have been applied to discrete sequence modeling and have shown impressive results on long context tasks with lower compute requirements (Gu et al., 2022). The original SSM formulation consists of four matrices that act as gates across a continuous data stream.

$$h_{t+1} = Ah_t + Bx_{t+1} \tag{1}$$

$$y_{t+1} = Ch_{t+1} + Dx_{t+1} \tag{2}$$

These matrices are discretized[1] (Pechlivanidou & Karampetakis, 2022) with a Δ value representing a step size across a continuous sequence.

$$\bar{A} = \exp(\Delta A) \tag{3}$$

$$\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I)\Delta B \tag{4}$$

The original SSM formulation is linear time-invariant, allowing it to be computed as an efficient 1-dimensional convolution over a sequence. However, the Mamba SSM variant makes the B, C, and D matrices input-dependent, making them more adaptable using gating (The A matrix is determined using the HiPPO matrix formulation for long context data storage (Gu et al., 2020)). Although this model is no longer time-invariant, it does not use activation functions. This way, it can be computed in an O(N) associative scan (Blelloch, 1990) using a parallelized, hardware-aware kernel (Dao et al., 2022).

2.3. Axial Attention

Previous MSA-based models ((Jumper et al., 2021), (Rao et al., 2021)) applied axial attention to establish relations across the sequence and MSA dimensions. Axial attention applies the attention process across items in a 2D matrix that shares the same coordinates (Figure 3). This way, the relevant column and row can be incorporated into the attention formulation (Ho et al., 2019).

Axial attention has shown high performance in protein models (Jumper et al., 2021). In this data modality, sequences reach a maximum length of no more than 5000 amino acids. However, the sizes of genes and genes/genomes are larger than proteins, which attention does not scale well to (see A.2).

[1] Recent work has shown that using the fixed HiPPO matrix and discretization cannot perform well in state-tracking tasks. We acknowledge this approach, but we use the original Mamba implementation due to its memory-efficient associative scan kernel

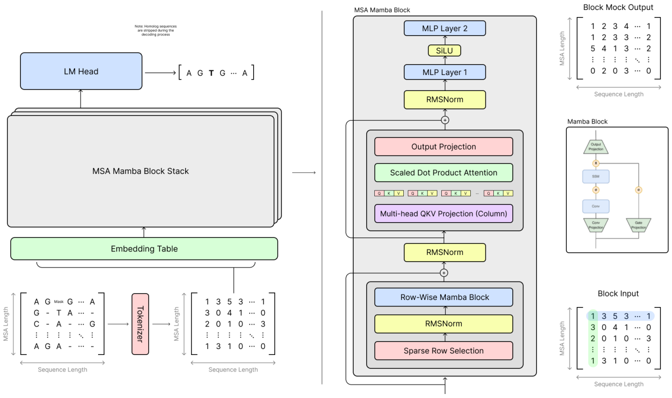


Figure 1. A diagram of the proposed MSAMamba architecture. The architecture consists of multiple MSAMamba blocks, each containing a mamba block that acts on the sequence length dimension and masked self-attention that acts on the MSA dimension. An MLP block follows the two processes (Dong et al., 2021).

3. Methods

This section provides an overview of the MSAMamba architecture, which fixes the computational complexity and context-length limitations of previous MSA models. This model uses Mamba's selective scan operation along the sequence dimension, which allows sequences to scale with a linear computational complexity (Proof A.3).

Unlike axial attention, analysis across the sequence and MSA dimensions are separated in MSAMamba. After running a selective scan in the horizontal dimension, the model runs multi-head attention with absolute position embeddings[2] along the MSA dimension. While this process shows quadratic scaling along the MSA dimension, most DNA MSAs do not scale past 100 species (see B.1), making MSA-related complexity scaling trivial compared to the sequence length dimension.

One MSAMamba block consists of a horizontal selective scan, an absolutely positioned vertical attention block, and a transition MLP block to encode memory (Jumper et al., 2021). There are residual connections (He et al., 2015) and RMSNorm (Zhang & Sennrich, 2019) blocks after the selective scan and attention operations, similar to the `Add + Norm` block used in transformer models (Vaswani et al., 2017).

3.1. Training

To improve the computational costs of MSAMamba further, we mask a percentage[3] of auxiliary aligned sequences in each MSA sample (Figure 4). This method allows the model to filter out masked rows during the selective scan operation, which decreases computational complexity on large batch sizes during training[4]. Both methods converge to similar training loss levels when comparing full and row-sparse MSA model training (see D).

Four MSAMamba models were trained to determine the architecture's efficacy (see C). Three models were trained on DNA sequence lengths of 128, 512, and 1024, respectively (with row-level masking). The fourth model was trained on a sequence length of 1024 without row-level masking to determine its effect on training performance. MSAMamba was trained on batch sizes that amounted to a total of 49152 sequences per batch (excluding augmented MSA sequences).[5]

The masked language modeling task (Algorithm 1) was used for pre-training, with 15% of each sequence being masked (Devlin et al., 2018). Both models were trained on the MultiZ90[6] genome dataset (see B.1).

4. Results and Discussion

We evaluate MSAMamba on the OMIM and ClinVar datasets for variant effect prediction on missense mutations (see B.2). This task tests the ability to leverage MSA information, as mutation predictions rely heavily on evolutionary data provided in aligned sequences (Benegas et al.,

[2] Used over rotary position embeddings because the absolute position of keys is required to identify which auxiliary sequence the model is analyzing

[3] We found that a 50% masking rate was optimal for row-level masking

[4] drops overall selective scan batch size due to MSA length scaling in batch (batch size * MSA length)

[5] batch size 48 for 1024 sequence length, batch size 96 for 512 sequence length, batch size 384 for 128 sequence length

[6] Modified from MultiZ100Way to exclude the ten genomes most similar to humans (Benegas et al., 2023)

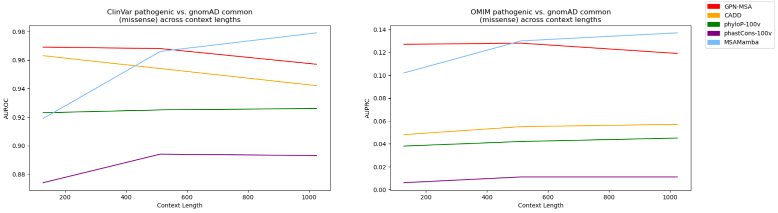


Figure 2. Graphs of MSAMamba (with row-level masking) and related models' performance on OMIM (AUPRC) and ClinVar (AUROC) missense mutation detection. The x axis shows the context length of the evaluated sequences

Task Name	CNN	DNABERT	HyenaDNA	GPN-MSA	MSAMamba
Mouse Enhancers	69.0	66.9	**85.1**	76.4	82.7
Coding vs Intergenomic	87.6	**92.5**	91.3	90.3	90.0
Human vs Worm	93.0	96.5	96.6	**98.9**	98.5
Human Enhancers Cohn	69.5	74.0	**74.2**	73.1	72.7
Human Enhancers Ensembl	68.9	85.7	89.2	88.8	**90.1**
Human Regulatory	93.3	88.1	93.8	93.5	**94.4**
Human Nontata Promoters	84.6	85.6	**96.6**	90.9	94.2
Human OCR Ensembl	68.0	75.1	80.9	76.8	**82.5**

Table 1. Evaluation of MSAMamba (with row-level masking), GPN-MSA, and other single sequence models on GenomicBenchmarks tasks using top-1 accuracy (%) metric

2023). This task was chosen to compare to current SOTA[7] MSA and non-MSA DNA models with similar training data. We assess the chosen models at increasing context lengths[8] to demonstrate MSAMamba's improved prediction capabilities at larger context lengths. General DNA models (MSAMamba, GPN-MSA) were fine-tuned using pooler and classification layers (see B.2.1), while task-specific methods (PhastCons, PhyloP) were used without fine-tuning. At each context length threshold, we evaluate the MSAMamba model trained at the respective sequence length. We evaluate baseline models based on released pre-trained models of similar size to MSAMamba's model dimensions (see Table 2).

Results (Figure 2) show that while GPN-MSA's performance on OMIM and ClinVar variant effect prediction decreases with increasing context length, the performance of MSAMamba increases. This is likely due to GPN-MSA's difficulty analyzing global relationships across longer sequences. MSAMamba shows the most significant performance increase across context length and exceeds SOTA DNA MSA models by a margin of ≈ 0.2 AUROC/AUPRC at a context length of 1024.

In addition, we evaluate MSAMamba on the GenomicBenchmarks datasets (Grešová et al., 2023) (modified using methods in B.2.2). Evaluations show that MSAMamba performs better than alternative models in 3 of 8 tasks (Table 3.1). These tasks are based on evolutionary relationships across species and require attention to global relationships. MSAMamba's longer context training and MSA data augmentation provide an advantage in these features. MSAMamba shows minor performance differences from the state-of-the-art in other GenomicBenchmarks tasks (maximum 2.4%). HyenaDNA shows high performance in 2 tasks due to its training on 2^{20} base pairs per batch, making it highly attuned to global DNA relationships (Nguyen et al., 2023).

5. Conclusions

By incorporating a subquadratic selective scan operation and separating processing along the sequence and MSA dimensions, MSAMamba achieves efficient and scalable inference on long DNA sequences. Our experiments demonstrate that MSAMamba exceeds the performance of state-of-the-art MSA and single-sequence models in four GenomicBenchmarks tasks (Table 3.1). In addition, the model shows performance exceeding current state-of-the-art DNA MSA models in long-context variant effect prediction (Figure 2). The row-sparse method used in MSAMamba's training process further enhances computational efficiency during the training process (Figure 5), making MSAMamba a viable and powerful tool for large-scale DNA analysis.

[7] GPN-MSA (Benegas et al., 2023) is the current state of the art for MSA-based processing. PhastCons, PhyloP, and CADD (Schubach et al., 2024) are also evaluated

[8] sequence lengths of size 128, 512, and 1024

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

Balech, B., Vicario, S., Donvito, G., Monaco, A., Notarangelo, P., and Pesole, G. MSA-PAD: DNA multiple sequence alignment framework based on PFAM accessed domain information. *Bioinformatics*, 31(15): 2571–2573, 03 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv141. URL https://doi.org/10.1093/bioinformatics/btv141.

Benegas, G., Albors, C., Aw, A. J., Ye, C., and Song, Y. S. Gpn-msa: an alignment-based dna language model for genome-wide variant effect prediction. October 2023. doi: 10.1101/2023.10.10.561776. URL http://dx.doi.org/10.1101/2023.10.10.561776.

Blelloch, G. E. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, November 1990.

Brown, T. *Introduction to Genetics: A Molecular Approach*. CRC Press, 2012. ISBN 9781136665356. URL https://books.google.com/books?id=byoWBAAAQBAJ.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020.

Dalla-Torre, H., Gonzalez, L., Revilla, J. M., Carranza, N. L., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., Sirelkhatim, H., Richard, G., Skwark, M., Beguir, K., Lopez, M., and Pierrot, T. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, 2023. doi: 10.1101/2023.01.11.523679. URL https://www.biorxiv.org/content/early/2023/01/15/2023.01.11.523679.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Dong, Y., Cordonnier, J.-B., and Loukas, A. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2793–2803. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/dong21a.html.

Dufter, P., Schmitt, M., and Schütze, H. Position Information in Transformers: An Overview. *Computational Linguistics*, 48(3):733–763, 09 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00445. URL https://doi.org/10.1162/coli_a_00445.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

Grešová, K., Martinek, V., Čechák, D., Šimeček, P., and Alexiou, P. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023. URL https://arxiv.org/abs/2312.00752.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Re, C. Hippo: Recurrent memory with optimal polynomial projections, 2020. URL https://arxiv.org/abs/2008.07669.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

Hamosh, A. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(Database issue): D514–D517, December 2004. ISSN 1362-4962. doi: 10.1093/nar/gki033. URL http://dx.doi.org/10.1093/nar/gki033.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. Axial attention in multidimensional transformers, 2019. URL https://arxiv.org/abs/1912.12180.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Jordan, M. I. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. 5 1986. URL https://www.osti.gov/biblio/6910294.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2006.16236.

Keles, F. D., Wijewardena, P. M., and Hegde, C. On the computational complexity of self-attention, 2022. URL https://arxiv.org/abs/2209.04881.

Landrum, M. J., Lee, J. M., Riley, G. R., Jang, W., Rubinstein, W. S., Church, D. M., and Maglott, D. R. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, 42(D1): D980–D985, November 2013. ISSN 1362-4962. doi: 10.1093/nar/gkt1113. URL http://dx.doi.org/10.1093/nar/gkt1113.

Lhoest, Q., del Moral, A. V., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Scao, T. L., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A. M., and Wolf, T. Datasets: A community library for natural language processing, 2021.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic level protein structure with a language model. July 2022. doi: 10.1101/2022.07.20.500902. URL http://dx.doi.org/10.1101/2022.07.20.500902.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Lopes, I., Altab, G., Raina, P., and de Magalhães, J. P. Gene size matters: An analysis of gene length in the human genome. *Frontiers in Genetics*, 12, February 2021. ISSN 1664-8021. doi: 10.3389/fgene.2021.559998. URL http://dx.doi.org/10.3389/fgene.2021.559998.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts, 2016. URL https://arxiv.org/abs/1608.03983.

Nassar, L. R., Barber, G. P., Benet-Pagès, A., Casper, J., Clawson, H., Diekhans, M., Fischer, C., Gonzalez, J. N., Hinrichs, A. S., Lee, B. T., Lee, C. M., Muthuraman, P., Nguy, B., Pereira, T., Nejad, P., Perez, G., Raney, B. J., Schmelter, D., Speir, M. L., Wick, B. D., Zweig, A. S., Haussler, D., Kuhn, R. M., Haeussler, M., and Kent, W. J. The ucsc genome browser database: 2023 update. *Nucleic Acids Research*, 51(D1):D1188–D1195, November 2022. ISSN 1362-4962. doi: 10.1093/nar/gkac1072. URL http://dx.doi.org/10.1093/nar/gkac1072.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Birch-Sykes, C., Wornow, M., Patel, A., Rabideau, C., Massaroli, S., Bengio, Y., Ermon, S., Baccus, S. A., and Ré, C. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. 2023.

Nguyen, E., Poli, M., Durrant, M. G., Thomas, A. W., Kang, B., Sullivan, J., Ng, M. Y., Lewis, A., Patel, A., Lou, A., Ermon, S., Baccus, S. A., Hernandez-Boussard, T., Re, C., Hsu, P. D., and Hie, B. L. Sequence modeling and design from molecular to genome scale with evo. February 2024a. doi: 10.1101/2024.02.27.582234. URL http://dx.doi.org/10.1101/2024.02.27.582234.

Nguyen, E., Poli, M., Durrant, M. G., Thomas, A. W., Kang, B., Sullivan, J., Ng, M. Y., Lewis, A., Patel, A., Lou, A., Ermon, S., Baccus, S. A., Hernandez-Boussard, T., Ré, C., Hsu, P. D., and Hie, B. L. Sequence modeling and design from molecular to genome scale with evo. *bioRxiv*, 2024b. doi: 10.1101/2024.02.27.582234.

URL https://www.biorxiv.org/content/early/2024/02/27/2024.02.27.582234.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Pechlivanidou, G. and Karampetakis, N. Zero-order hold discretization of general state space systems with input delay. *IMA Journal of Mathematical Control and Information*, 39(2):708–730, April 2022. ISSN 1471-6887. doi: 10.1093/imamci/dnac005. URL http://dx.doi.org/10.1093/imamci/dnac005.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models, 2023. URL https://arxiv.org/abs/2302.10866.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8844–8856. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

Rosen, Y., Roohani, Y., Agarwal, A., Samotorčan, L., Quake, S. R., and Leskovec, J. Universal cell embeddings: A foundation model for cell biology. November 2023. doi: 10.1101/2023.11.28.568918. URL http://dx.doi.org/10.1101/2023.11.28.568918.

Rosen, Y., Brbić, M., Roohani, Y., Swanson, K., Li, Z., and Leskovec, J. Toward universal cell embeddings: integrating single-cell rna-seq datasets across species with saturn. *Nature Methods*, February 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02191-z. URL http://dx.doi.org/10.1038/s41592-024-02191-z.

Schiff, Y., Kao, C.-H., Gokaslan, A., Dao, T., Gu, A., and Kuleshov, V. Caduceus: Bi-directional equivariant long-range DNA sequence modeling. 2024a.

Schiff, Y., Kao, C.-H., Gokaslan, A., Dao, T., Gu, A., and Kuleshov, V. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024b. URL https://arxiv.org/abs/2403.03234.

Schubach, M., Maass, T., Nazaretyan, L., Röner, S., and Kircher, M. Cadd v1.7: using protein language models, regulatory cnns and other nucleotide-level scores to improve genome-wide variant predictions. *Nucleic Acids Research*, 52(D1):D1143–D1154, January 2024. ISSN 1362-4962. doi: 10.1093/nar/gkad989. URL http://dx.doi.org/10.1093/nar/gkad989.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., Weinstock, G. M., Wilson, R. K., Gibbs, R. A., Kent, W. J., Miller, W., and Haussler, D. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*, 15(8):1034–1050, July 2005a. ISSN 1088-9051. doi: 10.1101/gr.3715005. URL http://dx.doi.org/10.1101/gr.3715005.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1050, 2005b.

Sofi, M. Y., Shafi, A., and Masoodi, K. Z. Chapter 6 - multiple sequence alignment. In Sofi, M. Y., Shafi, A., and Masoodi, K. Z. (eds.), *Bioinformatics for Everyone*, pp. 47–53. Academic Press, 2022. ISBN 978-0-323-91128-3. doi: https://doi.org/10.1016/B978-0-323-91128-3.00011-2. URL https://www.sciencedirect.com/science/article/pii/B9780323911283000112.

Su, J. Roformer: Transformer with rotary position embeddings - zhuiyiai. Technical report, 2021. URL https://github.com/ZhuiyiTechnology/roformer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I. M., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. December 2022. doi: 10.1101/2022.12.21.521521. URL http://dx.doi.org/10.1101/2022.12.21.521521.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity, 2020.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2019. URL https://arxiv.org/abs/1910.03771.

Wu, Q., Lan, Z., Qian, K., Gu, J., Geramifard, A., and Yu, Z. Memformer: A memory-augmented transformer for sequence modeling. In He, Y., Ji, H., Li, S., Liu, Y., and Chang, C.-H. (eds.), *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pp. 308–318, Online only, November 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-aacl.29.

Zhang, B. and Sennrich, R. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

A. Proof of Computational Complexity of Proposed Models

In this section, we symbolically calculate the time complexity of both axial-attention transformer models and the proposed MSAMamba architecture for DNA MSA modeling. We prove the following theorem:

$$\lim_{n \to \infty} \frac{C_{axial}}{C_{MSAMamba}} > 1 \tag{5}$$

Where $axial$ and $MSAMamba$ are both parallelized vector functions that take in an input tensor of dimension (m, n, d) [9], where n is the sequence length, m is the number of sequences is the MSA, and d is the vector function's dimension.

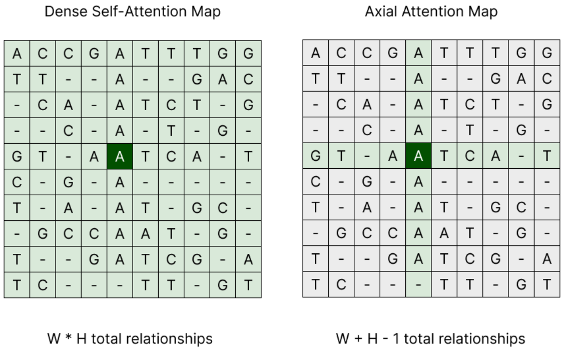
The values, C_{axial} and $C_{MSAMamba}$, represent the computational complexity of the respective models, given the baseline that one vector dot product or vector elementwise operation equates to one complexity unit.[10]

A.1. Assumptions

Some assumptions we make during the proof are as follows:

- We define C as the symbolic computational complexity, which we measure in units of # of operations. An operation can denote an elementwise vector operation or a vector dot product. A matrix multiplication $R^{m \times n} \cdot R^{n \times d} = R^{m \times d}$ is considered to be $m \times d$ total operations
- We exclude commonalities among the models (MLPs, normalization, residuals) from the complexity calculation and only include calculations that involve modeling relationships across MSA sequences
- the model size d is chosen to be 1 for the sake of symbolic simplicity throughout the proof. This does not affect the output of the limit, as it is determined by the sequence length variable n

A.2. Complexity of Axial Attention



A	C	C	G	A	T	T	T	G	G
T	T	-	-	A	-	-	G	A	C
-	C	A	-	A	T	C	T	-	G
-	-	C	-	A	-	T	-	G	-
G	T	-	A	A	T	C	A	-	T
C	-	G	-	A	-	-	-	-	-
T	-	A	-	A	T	-	G	C	-
-	G	C	C	A	A	T	-	G	-
T	-	-	G	A	T	C	G	-	A
T	C	-	-	-	T	T	-	G	T

A	C	C	G	A	T	T	T	G	G
T	T	-	-	A	-	-	G	A	C
-	C	A	-	A	T	C	T	-	G
-	-	C	-	A	-	T	-	G	-
G	T	-	A	A	T	C	A	-	T
C	-	G	-	A	-	-	-	-	-
T	-	A	-	A	T	-	G	C	-
-	G	C	C	A	A	T	-	G	-
T	-	-	G	A	T	C	G	-	A
T	C	-	-	-	T	T	-	G	T

Figure 3. A visualization of axial attention compared to fully dense attention. Axial attention significantly decreases the amount of relationships required per attention process.

Axial Attention (Figure 3) involves comparing each element within the input tensor with other elements on the same n and m axes. Each relationship comparison involves two dot products: one during the multiplication of Q and K matrices and another during multiplication by the V matrix. Each attention computation also consists of a softmax operation and an elementwise multiplication (scaling). The complexity calculation of axial attention for a single element in the tensor is as follows:

$$C_{i,j} = 4(n + m - 1) \tag{6}$$

This number of dot products is computed for every element in the MSA, leading to mn axial attention computations. The computational complexity of axial attention can be calculated with this information:

$$C_{axial} = 4nm(n + m - 1) = 4n^2m + 4nm^2 - 4nm \tag{7}$$

A.3. Complexity of MSAMamba

MSAMamba leverages the mamba operator for every sequence in the MSA, while the number of attention processes scales linearly based on sequence size.

Mamba's Selective Scan involves 3 vector dot products[11] and four elementwise multiplications[12] per sequence. We use this information to calculate Mamba's computational complexity:

$$C_{mamba} = n(3 + 4) = 7n \tag{8}$$

The column-wise attention process involves comparing every element of the same base pair index across MSA's, which leads to m^2 total comparisons per item in the sequence. Each relation consists of 2 dot products, as described in the axial attention complexity analysis in A.2. This leads to a complexity of

[9] excludes batch size for calculation

[10] Complexity is defined as computational complexity but is calculated similarly to time complexity. However, we do not use the term ”time complexity” due to parallelization that occurs on GPUs and other AI processing units

[11] Creation of B, C, and D matrices. Assumes 1:1 scale from input to inner SSM dimension

[12] A, B, C, and D gating matrices

$$C_{attention} = 2nm^2 \tag{9}$$

The overall computational complexity of MSAMamba is

$$C_{MSAMamba} = 7n + 2nm^2 \tag{10}$$

A.4. Confirming MSAMamba's Lower Time Complexity

We evaluate the limit defined in Eq. 5 given the time complexities calculated in Eq. 7 and 10. This gives the equation:

$$\lim_{n\to\infty} \frac{4n^2m + 4nm^2 - 4nm}{7n + 2nm^2} > 1 \tag{11}$$

To evaluate the limit to infinity, we take the terms in the numerator and denominator with the highest degree[13] of n, leading to the equation:

$$\lim_{n\to\infty} \frac{2n^2m}{n(2m^2+7)} > 1 \tag{12}$$

Since the degree of n in the numerator is higher than the degree of n in the denominator, we can ignore constant term coefficients and prove the following:

$$\lim_{n\to\infty} \frac{n^2}{n} > 1 \tag{13}$$

$$\lim_{n\to\infty} n > 1 \tag{14}$$

Therefore, the computational complexity (based on the number of vector calculations) of axial attention-based DNA MSA models increases by an order of magnitude faster than MSAMamba when scaling sequence length.

A.5. Summary of Proof and Relation To Proposed Model

The above proof shows that MSAMamba is more computationally efficient (concerning the number of vector calculations) at larger context lengths. DNA sequences consist of genes that can be up to millions of base pairs long and genomes made of billions of base pairs. MSAMamba's scaling properties show they can model longer DNA sequences more efficiently than current axial attention-based implementations.

Algorithm 1 MSAMamba Masked Language Modeling

```
Input: MSA $x$ : (B, M, L, D), $M_{row}$ : (B, M), $y_t$ : (B, L, D), lr, $\theta$ (Model Params)
Output: $y$ : (B, L, D)
$h_0$ = mask($x$, p=0.15)
for $i = 1$ to $n_{layers}$ do
    $h_{sparse}$ = $h_i[M_{row}]$
    $O_{mamba}$ = scatter(Mamba($x_{sparse}$), $M_{row}$) + $h_i$
    $O_{att}$ = SelfAttention($O_{mamba}$) + $O_{mamba}$
    $h_{i+1}$ = MLP($O_{att}$)
end for
loss = CrossEntropy($h_{n_{layers}-1}$[0], $y_t$)
$\theta \leftarrow$ AdamW(lr)
```

B. Datasets Used

This section gives an overview of the datasets used to pre-train and fine-tune MSAMamba, why they were selected for training, and preprocessing done on datasets for training tasks.

B.1. Pre-Training: MultiZ100Way

During model pre-training, we leverage the MultiZ100Way dataset, which consists of an MSA of the length of the human genome without any gap sequences in the human genome. It also consists of 99 auxiliary aligned sequences (with gap sequences) from 99 related species. This data has been curated from the public UCSC Genome Browser (Nassar et al., 2022). We use a modified version of this dataset (MultiZ90Way), which excludes ten auxiliary sequences of organisms that are very similar to those of humans. This modification was done to decrease training time and memory requirements.[14]

This dataset was used to train all baseline models. The same random seeds were also used for pre-training across MSAMamba and other baseline models.

B.1.1. Data Preprocessing

Data in the MultiZ100Way dataset was parsed using a tokenizer with a vocabulary size of 6. This consists of 4 nucleotides, one token for gap sequences, and one mask token. There was no need for `<PAD>` tokens due to all excerpts from the dataset being the same length.

This data was preprocessed based on the masked language modeling algorithm. This involves masking 15% of the sequence, where 80% of masked tokens are replaced with the `<MASK>` token, 10% is replaced with a random token, and the final 10% is not replaced (Devlin et al., 2018).

Note: Only the top sequence in the MSA (the human se-

[13] If two terms with the same degree are present, we take the one with the highest coefficient assuming $m = 90$

[14] The MultiZ90Way is publicly accessible through Hugging-Face datasets (Lhoest et al., 2021)

quence) is masked due to the focus on the human genome, with other genomes being additional information

B.2. Evaluation

B.2.1. Variant Effect Prediction Tasks

We use the OMIM and ClinVar Datasets during the evaluation process. The OMIM dataset relates gene sequences to different genetic disorders and their forms (Hamosh, 2004), while ClinVar relates aggregated gene variance information to overall human health (Landrum et al., 2013). Fine-tuning on this dataset evaluates a DNA MSA model's ability to perceive overall and individual gene relationships to determine its properties. The ClinVar dataset provides information about the clinical significance, and the model was trained on this target parameter for specific genetic disorders.

These two datasets were used at three sequence lengths: 128, 512, and 1024. Previous DNA-MSA transformer models were trained on a sequence length of 128 (Benegas et al., 2023). However, MSAMamba is trained on sequence lengths of 128, 512, and 1024. We compare evaluations from the fine-tuning processes across these increasing context windows to determine MSAMamba's relative efficacy when parsing longer MSA sequences.

The original dataset consisted of 128-length sequences. We modified these original sequences to include the area around the original sequence to add up to larger context lengths. This tests models' abilities to analyze specific mutations and segments within longer sequences.

All sequences were retrieved from the MultiZ100Way database given each sequence's chromosome index, start indices, and end indices. These sequences were not masked but passed as a tuple with a binary label as the fine-tuning target.

B.2.2. Genomic Benchmark Tasks

MSAMamba and other relevant models were also evaluated on the GenomicBenchmarks dataset (Grešová et al., 2023). This dataset consists of 8 different tasks relating to sequence-level classification. The original GenomicBenchmarks datasets are single-sequence, containing only the human genome. However, we use start indices, stop indices, and chromosome metadata from the datasets along with the MultiZ100Way database to generate MSA versions of these evaluation datasets.

These datasets were not modified for different sequence lengths and were only trained on their original sequence lengths.

Note: *Ethical considerations were carefully addressed during the data curation/processing step. All genome data used in this study were obtained from publicly available datasets (e.g., MultiZ100Way, OMIM, ClinVar)*

C. Training Details

This section gives an overview of the different training methodologies and hyperparameters used during the training process. We also provide different model sizes and configurations tested during the process.

C.1. Baseline Models

The primary baseline model we compared to in our paper is GPN-MSA, an axial-attention-based architecture that was trained on sequence lengths of size 128. We evaluated the original pre-trained GPN-MSA model on sequences of 128, 512, and 1024 base pairs to compare to MSAMamba at respective sequence lengths. The model had a dimension of 256 and consisted of 6 transformer layers. We trained the model with hyperparameters provided in the original paper.[15].

In addition, we use benchmarks from CADD, PhyloP, and phastCons in DNA variant effect prediction. Results for these models on 128 sequence length inputs were used from baseline metrics in GPN-MSA's evaluations. We evaluate these models on sequence lengths of 512 and 1024 on the same dataset used in evaluating MSAMamba. These models were fine-tuned for the given task based on the default provided hyperparameters and configuration (Nassar et al., 2022).

C.2. MSA Mamba Training

When training MSA Mamba, we swept across different magnitudes of learning rates and weight decays. We also tested with two primary configurations of betas in the AdamW optimizer, and we experimented with warm-up (Goyal et al., 2018) and cosine annealing learning rate (Loshchilov & Hutter, 2016) schedulers.

C.2.1. Model Sizes

We trained MSAMamba on a size of 3 total layers with a model dimension of 128. The SSM layer's dimension was scaled up by two times the model dimension, and the transition MLP module's magnification rate was 4x (similar to that of transformers). Model depth was kept constant to prevent external factors from influencing the model's long-context modeling performance measurements.

[15] learning rate: `1e-4`, weight decay: 0.01, 30K batches with warm-up scheduler for first 1K batches

Table 2. Table of model configurations that underwent the training, fine-tuning, and evaluation processes with comparison to baseline models with similar parameters

d_{model}	d_{ssm}	n_{layers}	Seq. Len	Row Sparse
128	256	3	128	×
128	256	3	512	×
128	256	3	1024	×
128	256	3	1024	√

C.2.2. Optimizers and Schedulers

We used the AdamW optimizer[16] during the pre-training and fine-tuning processes. We also used a warm-up scheduler for the first 10% of gradient steps. A weight decay of `1e-3` was used throughout pre-training and fine-tuning. For both tasks, we used betas of `(0.9, 0.95)`[17].

C.2.3. Hyperparameter Selection

We used a learning rate of `3e−5` for pre-training across all context lengths and row-level masking configurations. For fine-tuning, we used a learning rate of `3e-4`. We swept across the following learning rates during the pre-training process: `8e−3`, `2e−3`, `3e−4`, `3e−5`, `8e−6`, and found that `3e−5` was the highest performing learning rate in all model configurations.

Due to limited resources, the model was trained on a total sequence length of 2048 base pairs per physical batch. To compensate, we use a gradient accumulation across batches. This led to 49,152 base pairs in the provided MSA input per theoretical batch[18]. Validation loss was calculated after every two gradient updates. These batch sizes were used when fine-tuning the model as well.

C.3. MSAMamba Evaluation

The base MSAMamba model was modified during the evaluation process for sequence classification tasks. This was done by appending a pooler module that takes the last hidden state of each sequence in the batch and passes it as input to a single linear layer. A classifier linear layer follows this. Dropout was placed in between these layers with `p=0.25`. The final classifier layer was followed by a sigmoid function, which was used to compute binary cross-entropy loss as the objective function for genomic benchmarks and variant effect prediction tasks.

D. Efficacy of Row-Level Masking

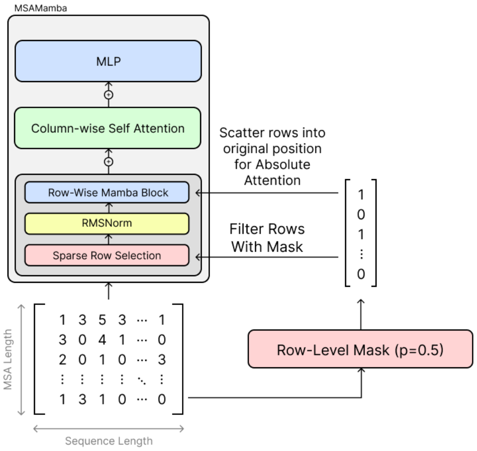


Figure 4. Shows the row-level masking method used for sparse computations of the full MSA matrix. We mask approximately half of all additional MSA sequences based on a random probability and filter out masked rows during the selective scan process. This decreases selective scan complexity for larger batch sizes without heavily diminishing performance

During MSAMamba's training process, row-level masking was used to randomly mask 50% of the MSA-augmented sequences. This decreased computational requirements for the selective scan operation of the model by a factor of 2, allowing for double the concurrent batch size.[19].

We compare training loss trajectories of MSAMamba with and without row-level masking to determine its effect on training performance. We find that while MSAMamba with row-level masking is slower to converge to an initial local minimum, it reaches a similar training loss level to MSAMamba without row-level masking (Figure 5).

E. Future Work

While MSAMamba has shown the efficacy of subquadratic operations in DNA MSA analysis, we have not presented results at higher depths and model sizes. This is due to compute and resource constraints for the project. In the future, we hope to train on larger context lengths (up to 32k, similar to HyenaDNA) and larger total base pairs per batch to improve generalization to global DNA relationships on the level of subquadratic single-sequence DNA models.

F. Compute Resources

This model was trained using Kaggle's AI services, leveraging the P100 GPU (16GB VRAM) to train these mod-

[16] We also tested the SGD optimizer due to initial issues in the adaptive training algorithm leveraged by Mamba. However, we found minimal difference between the two

[17] We experimented with a second beta of 0.99, but we discovered that it would lead to slower convergence and moved it to 0.95

[18] physical batch size × gradient accumulation iterations

[19] This was used due to constraints in compute resources, see F

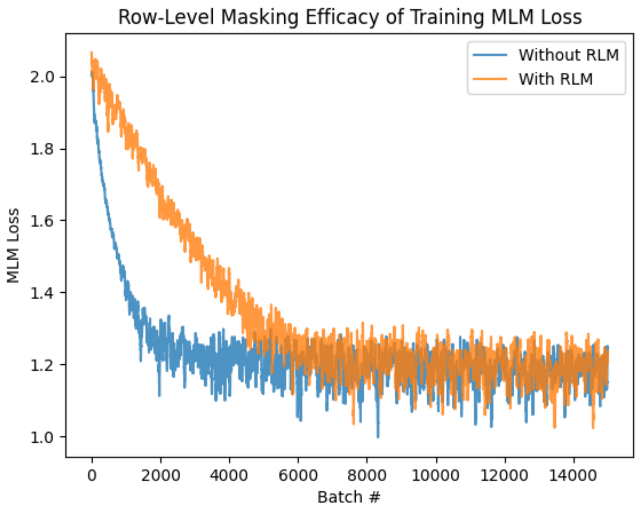


Figure 5. Shows the efficacy of row-level masking on training masked language modeling loss across the first 15000 batches

els. Kernels for `mamba_ssm` and `causal_conv1d` were ported to the Pascal architecture (SM60) to run on the P100.[20]

G. Code Availability

The code for this model is available on GitHub. We provide PyTorch implementations of the MSAMamba models and training/fine-tuning CLIs to easily run jobs on the model. We also offer data processing scripts in the `data/` folder.[21]

[20] In the future, we hope to train this model at larger model dimensions and increase depth to test its scaling laws at higher levels.

[21] This requires some modifications to the provided code based on which dataset to retrieve

Table 3. Table of open source tools used and corresponding licenses

TOOL NAME	LICENSE
NUMPY (HARRIS ET AL., 2020)	NUMPY LICENSE
MATPLOTLIB (HUNTER, 2007)	MATPLOTLIB LICENSE
PYTORCH (PASZKE ET AL., 2019)	BSD-3 CLAUSE
MAMBA (GU & DAO, 2023)	APACHE 2.0
CAUSALCONV1D (GU & DAO, 2023)	APACHE 2.0
HUGGINGFACE (WOLF ET AL., 2019)	APACHE 2.0
GENOMICBENCHMARKS (GREŠOVÁ ET AL., 2023)	APACHE 2.0
OMIM (HAMOSH, 2004)	OMIM USE AGREEMENT
CLINVAR (LANDRUM ET AL., 2013)	CLINVAR USE AGREEMENT
UCSC GENOME BROWSER (NASSAR ET AL., 2022)	UCSC GENOME BROWSER EULA