HYBRIDNA: A HYBRID TRANSFORMER-MAMBA2 LONG-RANGE DNA LANGUAGE MODEL

Mingqian Ma[1,2*] **Guoqing Liu**[1*] **Chuan Cao**[1*] **Pan Deng**[1*]
Tri Dao[3] **Albert Gu**[4] **Peiran Jin**[1] **Zhao Yang**[5] **Yingce Xia**[1]
Renqian Luo[1] **Pipi Hu**[1] **Zun Wang**[1] **Yuan-Jyue Chen**[1] **Haiguang Liu**[1] **Tao Qin**[1†]

[1]Microsoft Research AI for Science [2]UM–SJTU Joint Institute, Shanghai Jiao Tong University
[3]Dept. of Computer Science, Princeton University [4]Machine Learning Dept., Carnegie Mellon University
[5]Gaoling School of AI, Renmin University of China

mingqianma@sjtu.edu.cn, guoqingliu@microsoft.com, chuancao.926@gmail.com, pan.deng@microsoft.com, tri@tridao.me, agu@andrew.cmu.edu, peiranjin@microsoft.com, yangyz1230@gmail.com, yingce.xia@microsoft.com, {renqianluo,pisquare,zunwang,yuanjc,haiguang.liu,taoqin}@microsoft.com

*Equal contribution. †Corresponding author.

ABSTRACT

Advances in natural language processing and large language models have sparked growing interest in modeling DNA, often referred to as the "language of life". However, DNA modeling poses unique challenges. First, it requires the ability to process ultra-long DNA sequences while preserving single-nucleotide resolution, as individual nucleotides play a critical role in DNA function. Second, success in this domain requires excelling at both generative and understanding tasks: generative tasks hold potential for therapeutic and industrial applications, while understanding tasks provide crucial insights into biological mechanisms and diseases. To address these challenges, we propose **HybriDNA**, a decoder-only DNA language model that incorporates a hybrid Transformer-Mamba2 architecture, seamlessly integrating the strengths of attention mechanisms with selective state-space models. This hybrid design enables HybriDNA to efficiently process DNA sequences up to 131kb in length with single-nucleotide resolution. HybriDNA achieves state-of-the-art performance across 33 DNA understanding datasets curated from the BEND, GUE, and LRB benchmarks, and demonstrates exceptional capability in generating synthetic cis-regulatory elements (CREs) with desired properties. Furthermore, we show that HybriDNA adheres to expected scaling laws, with performance improving consistently as the model scales from 300M to 3B and 7B parameters. These findings underscore HybriDNA's versatility and its potential to advance DNA research and applications, paving the way for innovations in understanding and engineering the "language of life".

1 INTRODUCTION

Deoxyribonucleic acid (DNA) serves as the genetic code of life, encoding the instructions that govern gene expression, cellular processes, and biological functions. A deep understanding of the "language" of DNA is crucial for unraveling the molecular mechanisms that underlie biological functions and for leveraging these insights to advance medicine and biotechnology. The advent of high-throughput sequencing technologies has generated an immense volume of genomic data, creating an unprecedented opportunity for machine learning models to uncover complex patterns and relationships within DNA sequences. Foundation models, pre-trained on large-scale unlabeled datasets, have already demonstrated remarkable capabilities in natural languages (Devlin et al., 2019; Bommasani et al., 2021; Achiam et al., 2023) and protein languages (Brandes et al., 2022; Lin et al., 2023; He et al., 2024).

Recently, foundation models have begun to drive a paradigm shift in genomics, showcasing their ability to learn rich representations of DNA sequences that can be fine-tuned for a diverse array of downstream tasks. Currently, DNA foundation models primarily adopt two main architectural approaches. The first approach, inspired by BERT (Devlin et al., 2019), employs encoder-only Transformer architectures. Models such as DNABERT2 (Zhou et al., 2023) and Nucleotide Transformer (NT) (Dalla-Torre et al., 2023) excel at capturing contextual information within DNA sequences, producing high-quality embeddings suitable for tasks such as classification and regression. However, their bidirectional nature constrains their ability to design novel DNA sequences. The second approach leverages decoder-only architectures, such as Hyena (Poli et al., 2023b) and the Transformer architecture in GPT (Radford et al., 2018), which are autoregressive and well-suited for generative tasks. Models like HyenaDNA (Poli et al., 2023a) and Evo (Meier et al., 2023) have shown promising results in generating DNA sequences. Nevertheless, they often fall behind encoder-only models in understanding tasks requiring a deep understanding of sequence context.

This dichotomy highlights two critical challenges in DNA modeling: (1) How to develop a DNA foundation model that integrates robust contextual understanding with advanced design capabilities? Such a model would not only enhance the analysis of existing genomic data but also enable the design of novel, functional DNA sequences. (2) How to efficiently address the intricate complexity of DNA sequences, which involves long-range interactions critical to fundamental biological processes? Recent advances in Selective State Space Models (SSMs), such as Mamba (Gu & Dao, 2023; Dao & Gu, 2024), have shown remarkable potential for addressing information-dense tasks, including language modeling (Waleffe et al., 2024; Lieber et al., 2024). These models efficiently handle long-range dependencies with subquadratic complexity, offering a promising approach to the challenges posed by DNA sequence modeling. However, SSMs alone struggle to capture fine-grained, single-nucleotide-level interactions vital for understanding DNA function.

In this work, we introduce HybriDNA, a novel class of decoder-only DNA language models that leverage a hybrid Transformer-Mamba2 architecture. This hybrid design combines the complementary strengths of its components: Mamba2 blocks excel at efficiently processing long sequences and capturing long-range dependencies, whereas Transformer blocks enhance the model's ability to focus on fine-grained, token-level details within the context of the entire sequence. Pretrained on large-scale, multi-species genomes at single-nucleotide resolution with a next-token prediction objective, HybriDNA demonstrates foundational capabilities in both understanding and designing genomic sequences. By incorporating an *echo embedding* discriminative fine-tuning approach, HybriDNA achieves state-of-the-art performance across 35 biologically significant DNA understanding datasets, such as transcription factor binding prediction and promoter detection (Zhou et al., 2023). Additionally, through generative fine-tuning, HybriDNA exhibits exceptional proficiency in designing synthetic cis-regulatory elements (CREs) with desirable functional properties (Lal et al., 2024). Finally, we show that scaling up HybriDNA is beneficial: increasing model size from 300 million to 3 billion and 7 billion parameters improves performance, adhering to scaling laws observed in language models such as GPT (Radford et al., 2018; Schulman et al., 2022). Extending the context length (e.g., from 8 kilobases to 131 kilobases at single-nucleotide resolution) further enhances HybriDNA's performance on specific tasks. Together, these advancements position HybriDNA as a powerful tool for advancing both the understanding and engineering of genomic sequences.

2 HYBRIDNA FOUNDATION MODEL

In this section, we present the HybriDNA model for long-range genomic sequence modeling. We first provide a detailed description of the model architecture, followed by an explanation of the pre-training stage of HybriDNA. Finally, we explore the fine-tuning stages utilized for a range of downstream applications. The pipeline of our model is illustrated in Figure 1.

2.1 MODEL ARCHITECTURE

The HybriDNA model uses a decoder-only, sequence-to-sequence architecture purpose-built for efficiently and accurately processing long-range DNA sequences. It combines the unique strengths of Mamba2 selective state-space models and Transformer attention mechanisms within a hybrid framework inspired by recent hybrid architectures (Lieber et al., 2024; Glorioso et al., 2024; Ren et al., 2024). As shown in Fig. 1, the architecture consists of a series of HybriDNA blocks, where

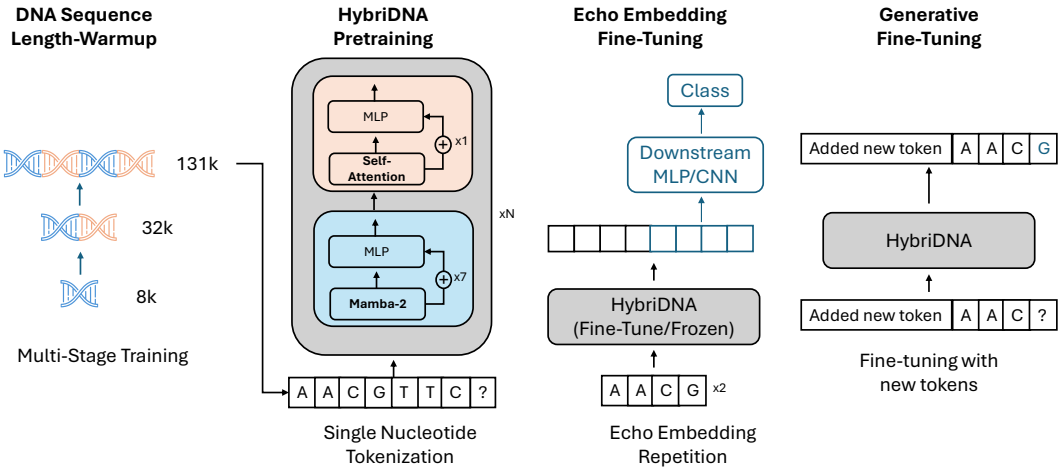

Figure 1: Overview of HybriDNA: A Language Model for DNA sequences. HybriDNA builds upon an efficient hybrid Transformer and Mamba2 architecture. It is first pre-trained on large-scale, multi-species genomic data at single-nucleotide resolution using a next-token prediction objective. Following this, HybriDNA utilizes an echo embedding fine-tuning approach for DNA understanding tasks and a generative fine-tuning approach for DNA generation tasks.

each block alternates between HybriDNA Mamba2 blocks and HybriDNA Transformer blocks in a 7:1 ratio. This configuration has been empirically proven to effectively balance the advantages of both block types, achieving optimal performance in the NLP domain (Waleffe et al., 2024). Details of the Mamba block and Transformer block configuration are provided in Appendix B.1. In the HybriDNA architecture, the first block is a HybriDNA Mamba2 block, eliminating the need for explicit positional embeddings or mechanisms such as RoPE (Su et al., 2024). This approach results in a HybriDNA design that completely omits positional encoding. Additionally, unlike the Jamba model (Lieber et al., 2024), the MLP layers in HybriDNA avoid using Mixture-of-Experts (MoE) configurations due to instability observed during fine-tuning for DNA-related downstream tasks. By leveraging this hybrid architecture, HybriDNA excels in both short- and long-range tasks while enabling the robust generation of synthetic DNA sequences.

2.2 Pretraining on Multi-Species Genomes

Dataset We pretrain HybriDNA on a large-scale, multi-species genome dataset using next nucleotide (token) prediction (NTP). This dataset was curated from the Nucleotide Transformer (Dalla-Torre et al., 2023) and NCBI. The resulting collection of genomes was downsampled to include a total of 845 species, comprising 160 billion nucleotides. The contribution of each class in the number of nucleotides relative to the total nucleotide count in the dataset is summarized in Table 7.

Tokenizer A straightforward and effective base-level tokenization strategy is adopted in HybriDNA, encoding each nucleotide (A, C, T, G) as an individual token. This approach ensures that the model processes genomic data with high fidelity to its natural structure, enabling nuanced interpretation and feature extraction. Unlike higher-order tokenization schemes that aggregate multiple bases into a single token, the base-level strategy treats each nucleotide as a fundamental unit, preserving its unique contribution to genomic patterns. This method is particularly advantageous for capturing low-level sequence variations with significant biological implications, such as single nucleotide polymorphisms (SNPs) and point mutations.

DNA Sequence Length Warm-up To enhance HybriDNA's ability to generalize effectively across longer genomic ranges, we implement a multi-stage warm-up procedure during the pre-training phase. The pretraining process begins by training the model with an 8,192 token context length, establishing a strong foundation for capturing intermediate sequence dependencies. After that, the context length is gradually increased—first to 32,768 tokens and then to 131,072 tokens—with each extension undergoing additional training equal to 2% of the training steps originally used for the 8,192 token context length. This gradual extension enables the model to adapt to increasingly

long-range dependencies and ensure efficient processing of large-scale genomic spans, equipping HybriDNA to excel in tasks that demand long-range comprehension.

2.3 Downstream Fine-tuning

2.3.1 Discriminative Fine-tuning for DNA Understanding tasks

To support both generative and understanding tasks, HybriDNA uses a GPT-like decoder-only architecture. A key limitation of autoregressive models is their inability to incorporate information from future tokens. To address this, HybriDNA introduces *echo embedding*, inspired by Springer et al. (2024), which leverages repeated sequences to encode bidirectional context. The central idea of this approach is that repeating sequences facilitates the encoding of contextual information from subsequent elements into the embeddings. To illustrate, consider an input sequence x and its corresponding label y for a classification task involving K classes. For example, given the input sequence $x = \texttt{AACG}$, we create an "echo" input by duplicating x: $x_{\text{echo}} = \texttt{AACGAACG}$. Subsequently, we extract the hidden embeddings from the final hidden layer, with particular attention to the embeddings from the latter half. We then apply a mean-pooling operation over all token embeddings to yield $h_\theta(x_{\text{echo}})$, which is designed to encapsulate contextual information from the repeated segment of the input. The pooled vector $h_\theta(x_{\text{echo}})$ is subsequently fed into a classification head, which may consist of a linear layer endowed with weights $W \in \mathbb{R}^{d\times K}$ and bias $b \in \mathbb{R}^K$, to produce the predicted probability distribution across the K classes:

$$P(y|x) = \text{softmax}(h_\theta(x_{\text{echo}})W + b). \tag{1}$$

To optimize the model, we employ the standard cross-entropy loss, which adjusts the parameters of either the classification head alone (W and b) or the entire model (θ, W, and b). By weaving bidirectional context into the autoregressive model, echo embeddings reconcile the traditional autoregressive embedding paradigm with the intricate demands of high-fidelity genomic tasks, offering substantial benefits for the analysis of extensive genomic sequences.

A potential limitation of employing echo embeddings for discriminative fine-tuning is the increased computational cost, as the doubled input length leads to higher memory requirements. However, HybriDNA's efficient hybrid architecture alleviates much of this burden, making the technique a practical and scalable solution for a wide range of genomic analysis and applications.

2.3.2 Generative Fine-tuning for DNA Generation Tasks

Autoregressive models like ChatGPT generate realistic, instruction-following text (Schulman et al., 2022). Similarly, HybriDNA, pre-trained on multi-species genomic data at single-nucleotide resolution, enables the design of novel DNA sequences for diverse applications.

We introduce prompt tokens encoding task-specific instructions, randomly initialized in the expanded embedding layer alongside the nucleotide vocabulary. HybriDNA predicts each nucleotide x_t sequentially, conditioned on preceding prompt tokens and generated nucleotides.

The model is optimized using the next-token prediction loss:

$$\mathcal{L}_{\text{generative}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\log\big(p_\theta(x_t \mid z_0,\ldots,z_{k-1},\ x_0,\ldots,x_{t-1})\big), \tag{2}$$

where θ denotes model parameters, z_{k-1} the k-th prompt tokens, and x_{t-1} the t-th generated nucleotide. Minimizing Eqn. 2 trains HybriDNA to generate sequences aligned with design goals.

3 Experiments

In the experiment section, we aim to answer the following questions regarding the HybriDNA models and their key capabilities: (1) How do the pretraining losses of HybriDNA models compare across different scales and configurations? (2) Can the models achieve state-of-the-art performance on short-range understanding benchmarks, and how does scaling affect their effectiveness? (3) How do the models perform on long-range understanding tasks, and do they exhibit improved results with increased pretraining context length? (4) Can the models generate realistic and desirable regulatory

sequences across multiple species? (5) How do the models compare to pure Transformer-based models in terms of computational efficiency during training?

We benchmark our model against a series of recently proposed tasks, including DNABERT2 (Zhou et al., 2023), BEND (Marin et al., 2024), and Genomics LRB (Poli et al., 2023c). In selecting tasks for comparison, we adhere to the principle of encompassing a diverse range of challenges, encompassing both short and long-range capabilities. Additionally, we prioritize tasks that are biologically meaningful, covering a variety of species and functionalities within DNA-related areas.

3.1 PRETRAINING CURVES

Configurations We train three model variants of HybriDNA with 300M, 3B, and 7B parameters. These models differ in the number of layers, hidden size, and learning rates. Despite these variations, all models share a consistent pretraining strategy. We use a next-token-prediction (NTP) loss instead of the commonly used masked language modeling (MLM) loss to enable generative ability. Our models are trained on NVIDIA A100/H100 and AMD MI300X GPUs. Details of our model and training settings can be found at Appendix B.

Scaling Behaviors To investigate scaling law behavior, we analyze the training and validation losses during the pretraining stage for the 300M, 3B, and 7B models. As the model size increases, we observe consistent improvements in both training and validation losses, highlighting the benefits of larger models in capturing intricate genomic patterns. Detailed loss curves for each model variant are presented in Fig. 5, which illustrate the diminishing returns in loss reduction as the model size grows. These findings align with theoretical expectations of scaling laws in deep learning and genomics-specific modeling.

We also demonstrate the effectiveness of using the hybrid Mamba2 and self-attention model, as opposed to using Mamba2-only models. The training and validation losses for pretraining two comparable 300M-size models are presented in Fig. 6.

3.2 SHORT-RANGE UNDERSTANDING BENCHMARKS (GUE, BEND)

We first evaluate our model on the short-range understanding capabilities using two comprehensive benchmarks: Genome Understanding Evaluation (GUE) and BEND. These benchmarks assess model performance on biologically meaningful tasks with short sequence length around hundreds bp. We compare HybriDNA with five state-of-the-art genomics foundation models: NT-500M-human, NT-2.5B-MS, DNABERT-2, HyenaDNA-medium-160k, and Caduceus-Ph-131k. For all baseline models, we utilize the pretrained weights provided in their respective original codebase. Detailed descriptions of our baseline models are provided in Appendix C.1.

GUE Following the identical settings in DNABERT-2, we use metrics of Matthews Correlation Coefficient (MCC) for all the tasks except F1 score for Covid task following the GUE dataset's original setting. The settings of hyperparameters, training epochs, and evaluation strategies follow exactly from the original paper, where we fine-tune all the parameters of the model and use the hidden state of the last token for embedding representation. The training epochs and evaluation steps are all followed from the original paper. We use a learning rate of 5e-5 for our 300M model, 3e-5 for our 3B model, and 1e-5 for our 7B model across all the tasks. We take the mean MCC/F1-score value of tasks in the same category and summarize results in Table 1. The suffix "(E)" in the model name within the table indicates that echo embedding was applied during discriminative fine-tuning. For detailed results on each task, refer to Appendix C.3.

BEND BEND paper (Marin et al. (2024)) presents a series of tasks for the evaluation of genomics foundation models. We select the three supervised classification tasks on local DNA sequences: Chromatin Accessibility, Histone Modification, and CpG Methylation. For the Histone Modification tasks, the training process follows the original paper. We freeze the embedding of the models and train a downstream CNN model for 100 epochs. For autoregressive models such as HyenaDNA and our HybriDNA model, we use the mean of the hidden state of the sequence as embedding representations. The model with the lowest validation loss is tested and the metric reported is the mean AUROC score. We report the AUROC score of each model on the three tasks in Table 2.

Type	Model	PD(H) *(MCC)*	CPD(H) *(MCC)*	SS(H) *(MCC)*	TF(H) *(MCC)*	TF(M) *(MCC)*	EMP(Y) *(MCC)*	CV(V) *(F1)*
Encoder	DNABERT-2	83.96	71.81	85.42	68.71	70.00	55.98	71.02
	NT-2.5B-MS	**88.15**	71.57	89.35	63.21	67.02	57.64	73.04
	NT-500M-human	82.96	66.79	78.63	61.92	45.24	45.35	50.82
	Caduceus-Ph	82.36	67.03	71.80	65.17	62.28	51.05	40.35
Decoder	HyenaDNA	80.14	69.22	77.76	61.74	64.39	47.15	25.88
Our	HybriDNA-300M	83.29	68.87	87.74	68.37	75.32	67.38	73.81
	HybriDNA-300M (E)	83.67	69.96	88.72	69.70	75.73	68.25	73.90
	HybriDNA-3B	85.40	69.50	89.01	70.48	75.43	**69.06**	74.05
	HybriDNA-3B (E)	85.55	70.71	89.10	71.13	77.14	68.97	**74.88**
	HybriDNA-7B	86.53	71.37	90.09	70.72	78.02	63.05	74.02
	HybriDNA-7B (E)	88.10	**72.03**	**90.12**	**72.01**	**79.02**	65.30	74.30

Table 1: Results on the GUE Benchmark, which includes a series of short-range classification tasks across multiple species, including Promoter Detection (PD), Core Promoter Detection (CPD), Splice Site Detection (SS), Transcription Factor Prediction (TF), Epigenetic Marks Prediction (EMP) and Covid Variant Classification (CV). The suffix "(H)" denotes the human genome, "(M)" the mouse genome, "(Y)" the yeast genome, and "(V)" the virus genome. The suffix "(E)" in the model name indicates that echo embedding was applied during discriminative fine-tuning.

Model Type	Model	Chromatin Accessibility *(AUROC)*	Histone Modification *(AUROC)*	CpG Methylation *(AUROC)*
Encoder Models	DNABERT-2	0.81	0.79	0.90
	NT-2.5B-MS	0.79	0.78	0.92
	NT-500M-human	0.74	0.76	0.88
	Caduceus-Ph	0.83	0.77	0.91
Decoder Models	HyenaDNA	0.81	0.77	0.87
Our Model	HybriDNA-300M	0.78	0.77	0.88
	HybriDNA-3B	0.82	0.79	0.92
	HybriDNA-7B	**0.84**	**0.79**	**0.93**

Table 2: Results on the BEND Benchmark, which includes Chromatin Accessibility, Histone Modification, and CpG Methylation tasks.

3.3 GENOMICS LONG-RANGE BENCHMARK (LRB)

The Genomics Long-Range Benchmark (LRB) introduced by Poli et al. (2023c) focuses on tasks that require understanding long-range context within genomic sequences. To assess models' ability to capture dependencies across extended genomic regions, we selected two tasks across distinct datasets that inherently demand long-range sequence comprehension: Causal eQTL Variant Effect Prediction, OMIM Variant Effect Prediction. During discriminative fine-tuning, all models were trained using frozen embeddings generated in the same manner as those used in the BEND benchmark. These embeddings were passed through an MLP classifier, with all models sharing identical architectures and hyperparameters to ensure consistency. We report accuracy and AUROC metrics for all tasks, with detailed results summarized in Table 3.

3.4 DESIGNING REALISTIC SYNTHETIC CIS-REGULATORY ELEMENTS (CRES)

regLM (Lal et al., 2024) integrates autoregressive language models with supervised sequence-to-function tasks to design synthetic cis-regulatory elements (CREs). This benchmark underscores the capability of models to generate regulatory sequences with specific desired properties. To evaluate the generative ability of our model, we adopt this framework for assessing DNA models' ability to design yeast promoters and human enhancers. This framework also highlights the distinct advantages of Decoder-only Genomics Foundation Models (GFMs). For result comparison, we utilize

Model Type	Model	Causal eQTL		OMIM
		Fine-tune (AUROC)	*Zero-shot (AUROC)*	*Zero-shot (AUPRC)*
Encoder Models	DNABERT-2	0.72	0.50	0.002
	NT-500M-human	0.72	0.51	0.003
	Caduceus-Ph	0.68	0.49	0.002
Decoder Models	HyenaDNA	0.71	0.51	0.002
Our Model	HybriDNA-300M (8k)	0.71	0.51	0.003
	HybriDNA-300M (32k)	0.72	0.51	0.003
	HybriDNA-300M (131k)	**0.74**	**0.51**	**0.003**

Table 3: Results on the LRB Benchmark, which includes Causal eQTL Variant Effect prediction, OMIM Variant Effect prediction tasks.

HyenaDNA, as it is not only the foundation of regLM but also the sole existing Decoder-only GFM, making it an ideal baseline for this evaluation.

Cell Type-Specific Human Enhancer Generation The human enhancer generation task involves producing desired human enhancer genomic sequences for three specific cell line types: HepG2, K562, and SK-N-SH. Each sequence incorporates a three-digit label (ranging from 0 to 3) that indicates the strength of the enhancer in a particular cell line. The evaluation metrics for the generated sequences are as follows: 1. Top-1 activity: The highest predicted enhancer activity for each cell type. 2. Mean activity: The average of the top 100 predicted activities for each cell type, aligned with the regLM methodology. 3. Diversity: The mean of pair-wise edit distance of the top 100 predicted sequences across all cell types, measuring the overall diversity of high-quality generated sequences. The results are summarized in Table 4.

Model	HepG2		K562		SK-N-SH		Diversity
	Top-1	Mean	Top-1	Mean	Top-1	Mean	Mean Edit Distance
Held-out Test	6.2	2.6	5.5	2.4	5.1	1.6	**110.10**
HyenaDNA	5.5	4.0	4.3	3.8	5.2	2.3	98.50
HybriDNA-300M	**7.3**	**5.4**	**7.8**	**6.2**	**6.6**	**4.7**	108.74

Table 4: Comparison between HybriDNA-300M and HyenaDNA on the human enhancer generation task. Metrics include Top-1 activity, Mean activity, and Diversity for each cell line type (HepG2, K562, SK-N-SH).

Yeast Promoter Generation The yeast promoter generation task follows a similar setup. However, instead of three cell lines, this task involves a two-digit label representing promoter activity in complex and defined media, with activity levels ranging from 0 to 4. Since HyenaDNA is pretrained only on human genomic data, the model is fine-tuned in the same manner as in the human enhancer task, rather than training it from scratch as done in regLM. The results are summarized in Table 5.

Model	Complex Media		Defined Media		Diversity
	Top-1	Mean	Top-1	Mean	Mean Edit Distance
Held-out Test	16.0	5.9	15.7	6.7	28.8
HyenaDNA	16.8	11.4	16.3	10.8	27.3
HybriDNA-300M	**18.2**	**15.0**	**17.6**	**13.5**	**30.7**

Table 5: Comparison between HybriDNA-300M and HyenaDNA on the yeast promoter generation task. Metrics include Top-1 activity, Mean activity, and Diversity for both media types.

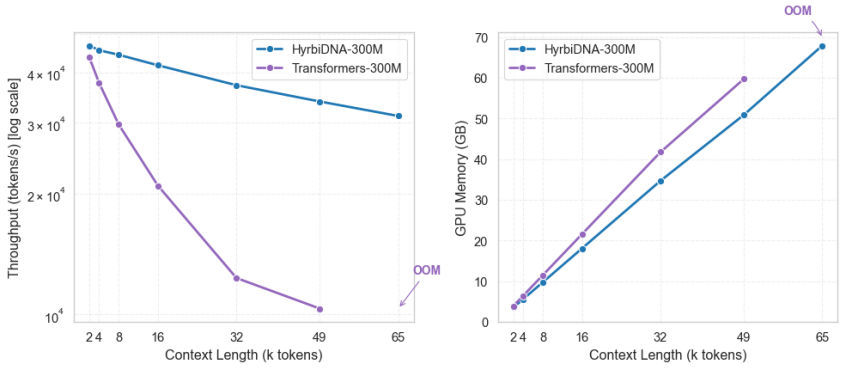

Figure 2: Comparison of throughput and GPU memory consumption during training between HybriDNA and a pure Transformer model with 300M parameters

3.5 Computational Efficiency

To evaluate the computational efficiency of our HybriDNA model relative to a standard Transformers model of comparable parameter size (utilizing the same Transformers block as in our hybrid model), particularly during the training phase, we compare their performance using the following two metrics: 1. Tokens/second per GPU: This metric assesses the throughput of a single GPU by measuring the number of tokens it can process each second during the pre-training stage. For each context length, the batch size is set to the maximum number to fit into the GPU memory. 2. GPU memory cost (GB): This measures the amount of GPU memory consumed when training a model with a fixed context length and a batch size of 1.

We assess the two models on four NVIDIA A100 GPUs (80G memory) using DeepSpeed Zero-1 Stage optimization and BF16 mixed-precision training. Both models comprise approximately 300M parameters. The Transformers model incorporates Flash Attention 2 optimization, while the Mamba2 layers in our HybriDNA model are implemented using CUDA kernels. We test both models at various context lengths —from 2k tokens up to 65k tokens—doubling the sequence length at each step.

As illustrated in Figure 2, HybriDNA model achieves significantly higher training throughput than standard Transformer models, especially when processing context lengths exceeding 32,000 tokens. For instance, at a context length of 49,000 tokens, the throughput of HybriDNA is approximately 3.4 times higher than that of Transformers. This performance gap widens as context length increases, highlighting the enhanced efficiency of our model compared to Transformers. In terms of GPU memory usage, our HybriDNA consistently achieves better efficiency than modern Transformer models, even those optimized with advanced techniques such as Flash Attention 2 (Dao, 2023). Notably, at context lengths around 65,000 tokens, a standard Transformer model not only runs into Out-Of-Memory (OOM) issues on A100 GPUs but also experiences a drop in throughput owing to its quadratic complexity. These findings underscore the exceptional ability of our hybrid model to manage larger context lengths effectively, a critical aspect for long-range DNA-related tasks.

4 Conclusion

In this work, we develop a class of decoder-only DNA language models built on a hybrid Transformer-Mamba2 architecture. By integrating Mamba2 layers, our model can process extremely long DNA sequences at single-nucleotide resolution with remarkable computational efficiency. Pretrained on large-scale, multi-species genomes at single-nucleotide resolution with a next-token prediction objective, HybriDNA demonstrates foundational capabilities in both understanding and designing genomic sequences. Through echo embedding discriminative fine-tuning, HybriDNA achieves state-of-the-art performance across 33 biologically significant DNA understanding tasks from the BEND, GUE, and LRB benchmarks. Through generative fine-tuning, HybriDNA exhibits remarkable proficiency in generating synthetic cis-regulatory elements with desirable functional

properties. These results highlight HybriDNA's versatility and establish its potential as a powerful foundation model for advancing DNA research and applications.

Looking ahead, there are several exciting directions to further explore. These include: (1) Expanding the pretraining dataset to include a greater number of nucleotide tokens and species classes, enabling broader generalization across downstream tasks involving diverse species. (2) Conducting more downstream fine-tuning tasks with diverse and significant scientific impacts, and performing wet-lab experiments to further validate the sequences designed by HybriDNA.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18 (10):1196–1203, 2021.

Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Ken Chen, Huiying Zhao, and Yuedong Yang. Capturing large genomic contexts for accurately predicting enhancer-promoter interactions. *Briefings in Bioinformatics*, 23(2):bbab577, 2022.

Lorenzo Dalla-Torre, Nicolás Benegas, Daria Grechishnikova, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, 2023.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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, 2019.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models, 2023. URL https://arxiv.org/abs/2212.14052.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model, 2024. URL https://arxiv.org/abs/2405.16712.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022a.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.

Liang He, Peiran Jin, Yaosen Min, Shufang Xie, Lijun Wu, Tao Qin, Xiaozhuan Liang, Kaiyuan Gao, Yuliang Jiang, and Tie-Yan Liu. Sfm-protein: Integrative co-evolutionary pre-training for advanced protein sequence representation. *arXiv preprint arXiv:2410.24022*, 2024.

Yu Ji, Zhiqiang Zhou, Han Liu, and Ramana V Davuluri. Dnabert: a comprehensive predictor for dna sequences based on deep transfer learning. *Bioinformatics*, 37(24):4776–4783, 2021.

Junru Jin, Yingying Yu, Ruheng Wang, Xin Zeng, Chao Pang, Yi Jiang, Zhongshen Li, Yutong Dai, Ran Su, Quan Zou, et al. idna-abf: multi-scale deep biological language learning model for the interpretable prediction of dna methylations. *Genome biology*, 23(1):219, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. reglm: Designing realistic regulatory dna with autoregressive language models. *bioRxiv preprint*, 2024.

Nguyen Quoc Khanh Le, Quang-Thai Ho, Van-Nui Nguyen, and Jung-Su Chang. Bert-promoter: An improved sequence-based predictor of dna promoter using bert pre-trained model and shap feature selection. *Computational Biology and Chemistry*, 99:107732, 2022.

Dohoon Lee, Jeewon Yang, and Sun Kim. Learning the histone codes with large genomic windows and three-dimensional chromatin interactions using transformer. *Nature Communications*, 13(1): 6678, 2022.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Frederikke I. Marin, Felix Teufel, et al. Bend: Benchmarking dna language models on biologically meaningful tasks. *arXiv preprint arXiv:2306.15006*, 2024.

Joshua Meier et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 382(6667):eado9336, 2023.

Chien Van Nguyen, Huy Huu Nguyen, Thang M. Pham, Ruiyi Zhang, Hanieh Deilamsalehy, Puneet Mathur, Ryan A. Rossi, Trung Bui, Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. Taipan: Efficient and expressive state space language models with selective attention, 2024. URL https://arxiv.org/abs/2410.18572.

Yu Ni, Linqi Fan, Miao Wang, Ning Zhang, Yongchun Zuo, and Mingzhi Liao. Epi-mind: identifying enhancer–promoter interactions based on transformer mechanism. *Interdisciplinary Sciences: Computational Life Sciences*, 14(3):786–794, 2022.

Michael Poli, Tri Dao, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15006*, 2023a.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023b.

Michael Poli et al. Genomics long-range benchmark (lrb): Evaluating long-context models on genomic data. *arXiv preprint arXiv:2306.00971*, 2023c.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling, 2024. URL https://arxiv.org/abs/2406.07522.

Joel Rozowsky, Jiahao Gao, Beatrice Borsari, Yucheng T Yang, Timur Galeev, Gamze Gürsoy, Charles B Epstein, Kun Xiong, Jinrui Xu, Tianxiao Li, et al. The en-tex resource of multi-tissue personal epigenomes & variant-impact models. *Cell*, 186(7):1493–1511, 2023.

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, et al. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *arXiv preprint arXiv:2306.15006*, 2023.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2(4), 2022.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings, 2024. URL https://arxiv.org/abs/2402.15449.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024.

Zixuan Wang, Meiqin Gong, Yuhang Liu, Shuwen Xiong, Maocheng Wang, Jiliu Zhou, and Yongqing Zhang. Towards a better understanding of tf-dna binding prediction from genomic features. *Computers in Biology and Medicine*, 149:105993, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Pengyu Zhang, Hongming Zhang, and Hao Wu. ipro-wael: a comprehensive and robust framework for identifying promoters in multiple species. *Nucleic Acids Research*, 50(18):10278–10289, 2022.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

A Preliminaries and Related Work

A.1 Attention Mechanism in Transformers

Powering many foundation models is the attention mechanism (Bahdanau, 2014; Vaswani, 2017) in Transformers. Attention is a type of operator that assigns scores to every pair of tokens in a sequence, enabling each element to "attend" to the others. The most widely adopted variant of attention to date is Scaled Dot-Product Attention, which is defined as:

$$y = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \cdot V, \tag{3}$$

where $x \in \mathbb{R}^{L \times d}$ represents an input sequence with sequence length L and embedding size d. The learnable parameters $W_K \in \mathbb{R}^{d \times d_k}, W_Q \in \mathbb{R}^{d \times d_k}$, and $W_V \in \mathbb{R}^{d \times d}$ are used to compute the key, query, and value matrices: $K = xW_K$, $Q = xW_Q$, and $V = xW_V$. The attention layer, therefore, transforms an input x of shape $\mathbb{R}^{L \times d}$ into an output y of the same shape, $\mathbb{R}^{L \times d}$.

Attention computes all pairwise comparisons for every token in a sequence, resulting in a computational complexity that scales as $O(L^2)$ with sequence length L. While this enables capturing global context at high resolution, it also restricts the context length on modern GPU architectures.

A.2 Selective State Space Models

Structured state space sequence models (S4) (Gu et al., 2022b; 2021) are a recent class of sequence models for deep learning that are broadly related to RNNs, CNNs, and classical state space models. They are inspired by a particular continuous system, which maps a 1-D function or sequence $f : x(t) \in \mathbb{R} \mapsto y(t) \in \mathbb{R}$ through a hidden state $h(t) \in \mathbb{R}^N$, N denotes SSM state size.

This continuous system is defined by four matrices $(\Delta, \mathbf{A}, \mathbf{B}, \mathbf{C})$, $\Delta \in \mathbb{R}, \mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times 1}$, $\mathbf{C} \in \mathbb{R}^{1 \times N}$. They define a sequence-to-sequence transformation in two steps:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), & h_t &= \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, & \bar{\mathbf{K}} &= (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, ..., \mathbf{C}\bar{\mathbf{A}}^k\bar{\mathbf{B}}), \\ y(t) &= \mathbf{C}h(t). & y_t &= \mathbf{C}h_t. & y &= x * \bar{\mathbf{K}}. \end{aligned} \tag{4}$$

Discretization S4 models represent discrete versions of the continuous system (corresponding to the second column of Eqn. 4), which incorporates a timescale parameter Δ to convert the continuous parameters $\mathbf{A}$, $\mathbf{B}$ into their discrete counterparts $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$. A commonly used transformation for this process is the zero-order hold (ZOH), defined as follows (where "exp" denotes the exponential):

$$\bar{\mathbf{A}} = \exp(\Delta\mathbf{A}), \quad \bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}. \tag{5}$$

Computation After the parameters have been transformed from $(\Delta, \mathbf{A}, \mathbf{B})$ to $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$, the model can be computed in two ways, either as a linear recurrence (corresponding to the second column of Eqn. 4) or a global convolution (corresponding to the third column of Eqn. 4). Commonly, the model uses the convolutional mode for efficient parallelizable training (where the whole input sequence is seen ahead of time), and switched into recurrent mode for efficient autoregressive inference (where the inputs are seen one timestep at a time).

Structured matrix $\mathbf{A}$ S4 (Structured SSM) models are so named because computing them efficiently also requires imposing structure on the $\mathbf{A}$ matrix. The most popular form of structure is diagonal (Gupta et al., 2022; Gu et al., 2022a; Smith et al., 2023). In this case, the $\mathbf{A}, \mathbf{B}, \mathbf{C}$ matrices can all be represented by N numbers. To operate over an input sequence x of batch size B and length L with D channels, the SSM is applied independently to each channel.

Linear Time Invariance (LTI) An key property of Eqn. 4 is that the model's dynamics remain constant over time, a characteristic known as Linear Time Invariance (LTI). In other words, the matrices $(\Delta, \mathbf{A}, \mathbf{B}, \mathbf{C})$, and consequently $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$, are fixed for all time-steps. The LTI property is closely tied to recurrence and convolutions. Before Mamba, all S4 models adhered to LTI (e.g. computed as convolutions during training) because of fundamental efficiency constraints.

This above formulation is central to S4. H3 (Fu et al., 2022) generalizes this recurrence to use S4; it can be viewed as an architecture with an SSM sandwiched by two gated connections. H3 also inserts a standard local convolution, which they frame as a shift-SSM, before the inner SSM layer.

Mamba (Gu & Dao, 2023) introduces the concept of Selective SSMs (S6), enhancing the traditional S4 framework through input-dependent gating mechanisms. The matrices $\bar{A}, \bar{B}$, and Δ are dynamically gated by the input x_t, enabling them to adjust their behavior based on the current input. Mamba simplifies the block design by combining the H3 block Fu et al. (2023) with gated MLPs. Additionally, Mamba proposes selective scan, a hardware-aware algorithm that computes the model recurrently using a scan operation, enhancing computational efficiency and scalability.

Mamba2 (Dao & Gu, 2024) builds upon Mamba1 by introducing two key enhancements:

1. From the perspective of the SSM layer, the new Structured State Space Duality (SSD) layer imposes a stricter constraint on the diagonal matrix $\bar{\mathbf{A}}$. The diagonal matrix is now reduced to a scalar times an identity matrix, which can be represented using only a single identical value across the diagonal. In this case, $\mathbf{A}$ can be represented with shape just sequence length.

2. The Mamba2 block produces the SSM parameters $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})$ in parallel with the input x, as opposed to sequentially in the Mamba1 block. This modification enables greater parallelism and scalability improvements, making tensor parallelism feasible for scaling the model to larger dimensions and longer contexts. Compared to Mamba1, Mamba2 allows much larger state dimensions (from $N = 16$ in Mamba1 to $N = 64$ to $N = 256$ or even higher) while simultaneously being much faster during training.

A.3 DNA Foundation Models

The advent of high-throughput sequencing technologies has produced vast amounts of genomic data, presenting an unprecedented opportunity for deep learning to uncover complex relationships and dependencies in DNA sequences. Recent advancements in genome language modeling have demonstrated their effectiveness across a wide range of downstream applications, including promoter prediction (Le et al., 2022; Zhang et al., 2022), gene expression prediction (Avsec et al., 2021), DNA methylation prediction (Jin et al., 2022), chromatin state analysis (Lee et al., 2022), promoter-enhancer interaction prediction (Chen et al., 2022; Ni et al., 2022) TF-DNA binding prediction (Wang et al., 2022), variant effect prediction (Rozowsky et al., 2023), gene network prediction (Theodoris et al., 2023) and more. More recently, inspired by advancements in natural language processing, researchers have begun developing DNA foundation models. These include, but are not limited to: (1) encoder-only models such as DNABERT, DNABERT-2, Nucleotide Transformer, and Caduceus; and (2) decoder-only models such as HyenaDNA and Evo.

DNABERT (Ji et al., 2021) is an early foundation model designed to interpret the human genome from a language perspective. By adapting the BERT framework with Transformers architecture, it captures a transferable understanding of human genome reference sequences. This single pre-trained Transformer model achieves state-of-the-art performance in tasks such as predicting promoters, splice sites, and transcription factor binding sites, after fine-tuning on small task-specific labeled datasets. The model contains 86M parameters and operates with a context length of 512 on the hg38 human reference genome dataset.

DNABERT-2 (Zhou et al., 2023) builds on its predecessor by employing Byte Pair Encoding (BPE) for tokenization, which improves computational efficiency and representation quality. It also incorporates Attention with Linear Biases (ALiBi) with Transformers-Encoder layers, enabling the model to process longer input sequences effectively. DNABERT-2 achieves state-of-the-art results on the Genome Understanding Evaluation (GUE) benchmark, showcasing its capacity to address diverse genomic tasks. The model consists of 112M parameters and is trained on a multi-species dataset comprising 135 species with a total of 32 billion nucleotides and a context length of 512.

Nucleotide Transformer (NT) (Dalla-Torre et al., 2023) is a scalable genomics foundation model, built on an encoder-only Transformer architecture, with parameter sizes ranging from 500M to 2,500M, based on encoder-only Transformer architecture. Its multi-species variant is pre-trained on genomic data from 850 species, employing a non-overlapping k-mer tokenization method that effectively reduces tokenized sequence lengths. Additionally, two human-specific versions are trained

separately on the hg38 human reference genome dataset and the 1000 Genomes Project. All pre-training is conducted with a context length of 1,000 tokens.

Caduceus (Schiff et al., 2023) introduces the bi-directional Mamba1 architecture, specifically designed for DNA sequence modeling. By incorporating reverse complement (RC) equivariance at the architectural level, Caduceus is optimized for long-range DNA sequence modeling. The model effectively captures the intricate understanding required for DNA sequence tasks. The Caduceus series features parameter sizes ranging from 500K to 7M, with a context length of 131k, and is trained on the hg38 dataset.

HyenaDNA (Poli et al., 2023a) utilizes the Hyena operator, a recurrence of gating and implicitly parametrized long convolutions, to handle long-range genomic sequences, enabling the processing of input contexts up to 1 million tokens with single-nucleotide resolution. This model shows effectiveness in tasks requiring long-range understanding, such as analyzing DNA fragments far apart, beyond the context window of traditional Transformer models. HyenaDNA is trained on the hg38 human reference genome dataset, with parameter sizes ranging from 1.7M to 50M and context lengths varying from 1k to 1M.

Evo (Meier et al., 2023) is a 7-billion-parameter foundation model built on the StripedHyena architecture and trained on 2.7 million raw prokaryotic and phage genome sequences. It integrates multiple biological modalities, including DNA, RNA, and proteins. With a context length of 131k nucleotide bases, Evo delivers superior performance in sequence modeling and functional design tasks, spanning molecular to genome-scale applications.

A.4 Hybrid models in general domains

Recent advancements in Mamba-based hybrid models for NLP tasks combine the efficiency of SSMs with the expressiveness of attention mechanisms, excelling in long-context scenarios. Innovations include Jamba's (**?**) integration of Transformer, Mamba, and Mixture-of-Experts layers for sequences up to 256k tokens, Zamba's (Glorioso et al., 2024) compact 7B model with shared self-attention for reduced latency, and SAMBA's (Ren et al., 2024) sliding window attention for efficient handling of sequences up to 1M tokens. Other notable contributions include Taipan's (Nguyen et al., 2024) selective attention layers for scalability and Waleffe's (Waleffe et al., 2024) versatile 8B hybrid architecture combining Mamba2, self-attention, and MLP layers. These models achieve strong results across various short- and long-range benchmarks.

B HybriDNA Model

B.1 HybriDNA Block

A key component of the HybriDNA Mamba2 block is the **State-Space Duality (SSD)** layer (Dao & Gu, 2024). It processes input sequence x efficiently using the recurrence:

$$h_t = A_t h_{t-1} + B_t x_t, \quad y_t = C_t^\top h_t, \tag{6}$$

where $h_t \in \mathbb{R}^N$ is the hidden state, $x_t \in \mathbb{R}$ is the input, $A_t \in \mathbb{R}^{N\times N}$ represents state transitions, $B_t \in \mathbb{R}^{N\times 1}$ projects the input, and $C_t \in \mathbb{R}^{N\times 1}$ maps the hidden state to the output $y_t \in \mathbb{R}$.

The SSD layer simplifies the matrix A_t to $A_t = a_t I$, where $a_t \in \mathbb{R}$ is a scalar and I is the identity matrix, to further improve efficiency. This simplification reduces the recurrence to:

$$h_t = a_t h_{t-1} + b_t x_t, \quad b_t = B_t. \tag{7}$$

For multi-dimensional inputs $x \in \mathbb{R}^{L\times d}$, the SSD layer is extended into a multi-head design, where each head independently processes a distinct subset of the input dimensions, similar to the mechanism of multi-head attention in Transformers. This architecture enables the SSD layer to capture complex interactions across multiple input channels in parallel, greatly enhancing its representational capacity. Typically, the head dimension is set to 64 or 128, consistent with standard configurations used in Transformers.

Computationally, the SSD layer can be reformulated as a matrix operation:

$$y = Mx, \quad M_{ij} = \begin{cases} C_i^\top A_{i:j+1} B_j & \text{if } i \geq j, \\ 0 & \text{otherwise,} \end{cases}$$

where $A_{i:j+1} = A_i A_{i+1} \cdots A_j$. The matrix M is semiseparable, meaning its submatrices possess low-rank properties. Specifically, a semi-separable matrix M can be decomposed into the sum of two components:

$$M = UV^\top + K,$$

where U and V capture the structured part, and K represents the lower-triangular portion. This structure ensures efficient computation with $O(NL)$ complexity, which is significantly faster than the $O(L^2)$ cost of traditional Transformers-based methods.

As shown in Fig. 3, the HybriDNA Mamba2 block is a scalable, hardware-optimized architecture designed to efficiently process input sequences by integrating grouped-value projections and lightweight convolutional operations. These projections, combined with 1D convolutions, allow for flexible feature extraction and dimensionality reduction while maintaining computational efficiency. To further optimize performance, all data-dependent projections are computed in parallel at the start of the block, leveraging tensor parallelism to maximize the utilization of matrix multiplication units on modern GPUs. Additionally, the Mamba2 blocks employ RMSNorm normalization (Zhang & Sennrich, 2019) both before input projection and after the SSD layer, which improves training stability, especially at large model scales.

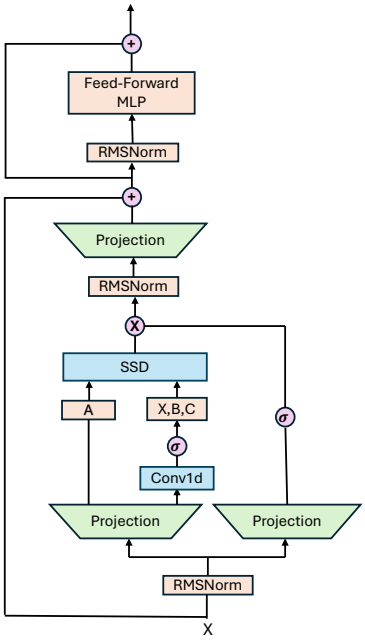

Figure 3: HybriDNA Mamba2 Block

For the HybriDNA Transformers block, we adopt a standard Transformers Decoder block as described in Section A.1.

B.2 Architecture Details

The **HybriDNA** model employs a hybrid Transformer-Mamba2 architecture. The architecture interleaves Transformer and Mamba2 layers in a 7:1 ratio, optimizing the strengths of both mechanisms. The Transformer layer is placed in the fourth of every eight layers. Our three model variants—300M, 3B, and 7B—differ in their hidden dimension size and layer configurations. The details of each model are summarized in Table 6:

Model Variant	# Layers	Hidden Size	Intermediate Size	# Heads	Head Dim
7B	32	4096	8192	128	64
3B	16	4096	8192	128	64
300M[1]	24	1024	2048	32	64

Table 6: Model configurations of HybriDNA's three model variants.

B.3 Training

Fig. 4 is a detailed architecture diagram of our **HybriDNA** model and its components. We train the HybriDNA models using a standard causal language modeling (CLM) objective. The training

[1]You may notice that HybriDNA-300M model has 32 layers, inspired by the configuration of Jamba-1.5 model: https://huggingface.co/ai21labs/AI21-Jamba-1.5-Large/blob/main/config.json.

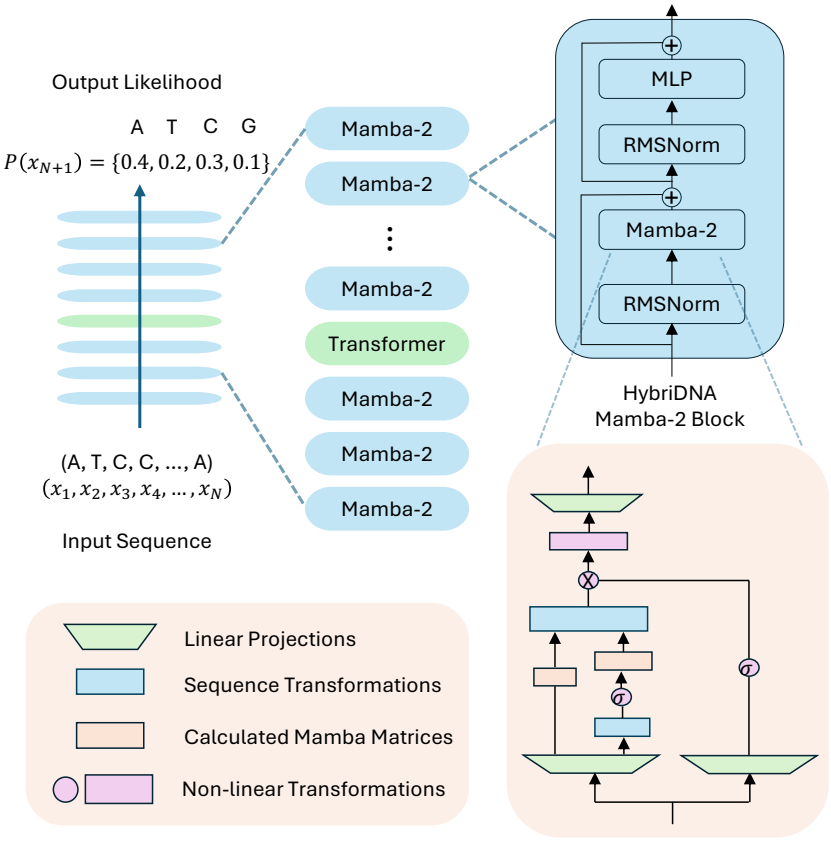

Figure 4: Model Architecture of HybriDNA

employs the Adam Kingma & Ba (2015) optimizer with a learning rate schedule and standard exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 1\text{e}{-8}$. All models are trained with a warmup phase of 2,000 steps and a total of 500,000 steps. The learning rate for the 300M model is $1\text{e}{-3}$, for the 3B model is $6\text{e}{-4}$, and for the 7B model is $1\text{e}{-4}$. Mamba-based models demonstrate higher tolerance for learning rates compared to standard Transformer architectures, showcasing their stability during optimization.

Our 300M, 3B and 7B models are trained on 0.5M tokens per batch, optimized for efficient utilization of computational resources and consistent training dynamics. Initially, the models are pretrained on sequences with 8192 context length for 500k steps, resulting a total of 250B tokens ($\sim$1.5 epoch) in the first pretraining stage. Following this, the models undergo further pretraining to extend their capabilities to handle larger context lengths. This two-stage pretraining strategy allows the models to gradually adapt to more complex and computationally demanding settings, ensuring robust performance across varying sequence lengths.

Training was conducted on the following hardware configurations: the 300M model on 8 AMD MI300X GPUs, the 3B model on 8 NVIDIA H100 GPUs, and the 7B model on 64 AMD MI300X GPUs. Models are trained for approximately 300 hours for the 300M and 7B variants, and 500 hours for the 3B model. These configurations ensure efficient utilization of computational resources and stable training for large-scale models.

B.4 Dataset

We utilized a comprehensive dataset comprising approximately 200 billion tokens, following the Nucleotide Transformer's multi-species dataset (Dalla-Torre et al., 2023). It comprises of a subset of the NCBI dataset with 850 species and the details are in Tab. 7.

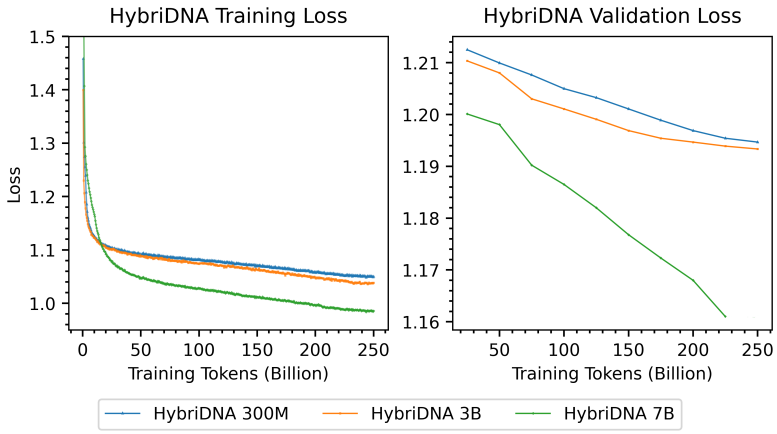

Figure 5: Pretraining loss curves for HybriDNA-300M, 3B, and 7B models

Class	# Species (train)	# Nucleotides (train)	# Species (valid)	# Nucleotides (valid)
Bacteria	647	16.5B	20	0.5B
Fungi	44	2.0B	3	0.2B
Invertebrate	37	19.9B	2	1.9B
Protozoa	9	0.45B	1	0.05B
Mammalian Vertebrate	28	65.2B	3	4.6B
Other Vertebrate	51	57.4B	6	6.0B
Total	**845**	**160.75B**	**35**	**13.25B**

Table 7: Statistics of multi-species pretraining data for our HybriDNA model.

B.5 Hybrid-Model Effectiveness

To evaluate the effectiveness of incorporating Transformers layer in our HybriDNA model, we pretrain a variant of 300M-size model without Transformers layer. Both our Hybrid and pure Mamba2 model are pretrained using 8k context length. We show the training and validation loss in pretraining stage in Fig 6. We can see that for models with similar parameter size, hybird model demonstrates lower training and validation loss comparing to a model comprised with pure Mamba2 blocks.

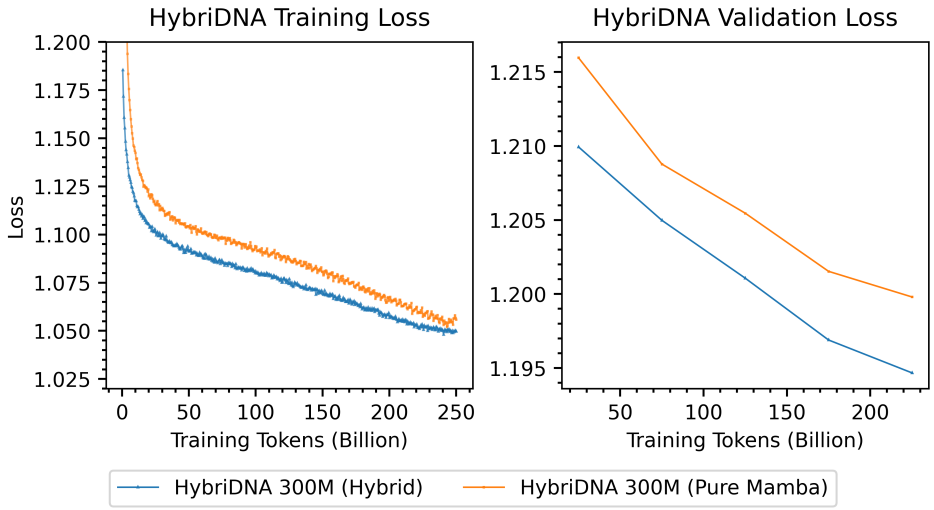

Figure 6: Effectivenss of Hybridization in HybriDNA

It can be observed that the incorporation of some Transformers blocks into the HybriDNA architecture brings improvements in both training and validation loss in the pretraining stage.

C Downstream Experiments Setup and Results

C.1 Baseline Model Descriptions

In this section, we describe the details of the baseline models we benchmark against.

NT-500M-human is a Transformer-encoder based model with 500M parameters pretrained on GRCh38/hg38 human reference genome with about 3.2B nucleotide bases. It utilizes k-mer tokenization method with $k = 6$ and is pretrained using standard Masked Language Modeling (MLM) objective. The pretraining context length is 1,000.

NT-2.5B-MS is another variant of Nucleotide Transformer series with larger size of 2.5B parameters and was pretraiend on a multi-species dataset with 174B nucleotides from a total of 850 species. Other pretraining details remain the same with NT-500M-human.

DNABERT-2 is also a BERT-like model with 112M parameter size. It is pretrained on a multi-species dataset with 135 species of roughly 32.5B total nucleotide bases. It improves the tokenizer with the Byte Pair Encoding (BPE) method and is also trained with standard MLM objective with a context length of 512.

Caduceus-Ph-131k builds upon the Mamba architecture, which exmploys selective state space models for long-range sequence processing. It enables Bi-directional sequence modeling using Bi-Mamba block. The model is trained on GRCh37/hg37 human reference genome with about 3.2B nucleotide bases using MLM objective. This variant has 7.73M parameter size and uses nucleotide-level tokenization with pretraining context length of 131,072.

HyenaDNA-Medium-160k utilizes the Hyena operator, derived from state space models, for computationally efficient long-range sequence modeling. It is pretrained on GRCh38/hg38 human reference genome using next-token prediction objective. This specific variant has 14.2M parameters and uses nucleotide-level tokenization with pretraining context length of 160,000.

Since our evaluation is mainly focused on eukaryote-related tasks, Evo is excluded from the comparison.

C.2 Downstream Tasks

GUE (Zhou et al., 2023) aggregates 28 datasets across 9 tasks, encompassing input lengths from 70 to 512 bp. GUE serves as a standardized evaluation suite, measuring the effectiveness of genomic foundation models on multi-species genome classification. For the GUE benchmark, we use the exact settings for each task, including the warmup steps and training/validation steps that are customized for each task. The only modification we've made is the learning rate: $5\mathrm{e}{-5}$ for the 300M model, $3\mathrm{e}{-5}$ for the 3B model, and $1\mathrm{e}{-5}$ for the 7B model. We apply a simple classification head for the model and either use the hidden state of the last token to classify for the ordinary setting or the averaged hidden state of the repeated sequence input for echo embedding.

BEND (Marin et al., 2024) evaluates models on a collection of realistic and biologically meaningful tasks defined on the human genome. It emphasizes the importance of capturing intricate features with biologically meaningful tasks that are comprehensive and provide a standard evaluation methodology for genomics foundation models. We select the 3 largest short-range tasks on 3 different datasets from the benchmark for evaluation. For the BEND dataset, we adopt the exact settings for all three tasks, using a learning rate of $3\mathrm{e}{-3}$ and training for 100 epochs. It adopts a linearly decreasing learning rate to 0 and we use the epoch with the lowest validation loss for the final test-set evaluation. The task freezes the embedding input of the model and only fine-tunes a downstream two-layer CNN model for classification. The hidden state extracted is the average of hidden states of the input for the ordinary setting and the average of the repeated sequence for the echo-embedding setting.

Genomics LRB Following the newer version of the Genomics LRB paper, for all fine-tuning tasks we fine-tune all the parameters of the model with the following MLP for classification. we finetune based on the benchmark's setting by inputting sequence with the pretraining context length into the model and then take the average across the same length window for the same task around different models. For zero-shot tasks, we use the sequence-level probability for a regression correlation coefficient analysis also identical to the benchmark's method.

regLM For the baseline HyenaDNA model, we follow the exact setting of the regLM model to load its fine-tuned checkpoint. As for fine-tuning our HybriDNA-300M model, we finetune 16 epochs on the human enhancer task with learning rate of 1e-4. For the Yeast Promoter task, as our model itself has been pretrained on multi-species data including yeast sequence, we fine-tuned our model on the dataset for 2 epochs also with a learning rate of 1e-4. We carry out validation for every 400 steps for each task and save the model with the highest validation accuracy as the final model. During generation, we use beam search with a beam width of 2500 and beam size of 256 for both models on both tasks to generate 200 sequences with each label for evaluation. The activity scoring models are the same with those in the original regLM models and the diversity metric is calculated by the mean pair-wise edit distance of the top-100 activity sequence of all labels for each task.

For the Human Enhancer Task, the model is fine-tuned for 16 epochs with a learning rate of 1e-4, incorporating these prompt tokens. Performance is validated using the accuracy of the validation set. The fine-tuning dataset consists of 670k training samples of 200bp enhancers with varying levels of activity. After fine-tuning, the model is tasked with generating 600 sequences for each label {300, 030, 003}, representing high enhancer activity in a specific cell line. These generated sequences are then evaluated using a scoring model from regLM to assess their actual enhancer activity in the respective cell types. Beam search decoding is employed during sequence generation for fair comparison. The baseline for comparison is the fine-tuned HyenaDNA model variant, "hyenadna-medium-160k-seqlen," as referenced in the original regLM paper.

For the Yeast Promoter Task, the model is fine-tuned for 2 epochs with a learning rate of 1e-4, validated using the validation set accuracy. The fine-tuning dataset comprises approximately 7.4M training samples of 80 bp promoters with varying activity levels. Models are prompted to generate sequences with label {40, 04}. The evaluation steps and metrics are consistent with those used in the human enhancer generation task.

C.3 GUE Benchmark Results

Model Type	**Model**	**Transcription Factor Prediction (Human)**				
		0 *(MCC)*	**1** *(MCC)*	**2** *(MCC)*	**3** *(MCC)*	**4** *(MCC)*
Baselines	DNABERT-2	70.89	74.49	66.62	60.35	71.21
	NT-2.5B-MS	66.46	70.25	58.70	51.28	69.34
	NT-500M-human	60.03	69.34	47.02	39.27	58.84
	Caduceus-Ph-131k	70.69	69.00	61.13	55.98	69.07
	HyenaDNA-160k	64.47	70.74	60.44	39.78	73.27
Our Model	HybriDNA-300M	68.12	67.13	70.29	55.52	80.80
	HybriDNA-300M(E)	67.64	71.28	70.84	57.92	80.80
	HybriDNA-3B	69.88	69.24	72.21	56.44	84.61
	HybriDNA-3B(E)	69.02	70.82	**72.80**	58.01	85.02
	HybriDNA-7B	70.00	74.47	70.42	64.52	85.03
	HybriDNA-7B(E)	**71.46**	**75.60**	71.81	**65.82**	**86.20**

Table 8: Results on Transcription Factor Prediction (DNABERT2-Human) in GUE benchmark

Model Type	Model	Promoter Detection (Human)			Splice Site Prediction (Human)
		all *(MCC)*	**notata** *(MCC)*	**tata** *(MCC)*	**reconstruct** *(MCC)*
Baselines	DNABERT-2	86.64	94.20	71.04	85.42
	NT-2.5B-MS	**91.00**	94.02	**79.43**	89.35
	NT-500M-human	81.34	88.73	78.82	78.63
	Caduceus-Ph-131k	83.98	92.13	70.96	71.80
	HyenaDNA-160k	83.04	91.03	66.36	77.76
Our Model	HybriDNA-300M	88.94	94.44	69.63	87.74
	HybriDNA-300M(E)	88.81	94.45	68.45	88.72
	HybriDNA-3B	89.48	94.49	72.24	89.01
	HybriDNA-3B(E)	89.30	94.33	73.02	89.10
	HybriDNA-7B	88.28	94.73	73.59	90.09
	HybriDNA-7B(E)	90.20	**94.57**	76.84	**90.12**

Table 9: Results on Promoter Detection and Splice Reconstruct (DNABERT2-Human) in GUE benchmark

Model Type	Model	Core Promoter Detection (Human)		
		all *(MCC)*	**notata** *(MCC)*	**tata** *(MCC)*
Baselines	DNABERT-2	69.97	69.62	**75.83**
	NT-2.5B-MS	**70.28**	71.49	72.95
	NT-500M-human	63.36	64.67	72.34
	Caduceus-Ph-131k	64.09	68.35	68.65
	HyenaDNA-160k	66.18	67.41	74.07
Our Model	HybriDNA-300M	68.40	69.12	69.09
	HybriDNA-300M(E)	68.37	69.15	72.36
	HybriDNA-3B	68.98	69.63	69.89
	HybriDNA-3B(E)	68.90	70.01	73.21
	HybriDNA-7B	66.50	70.66	76.94
	HybriDNA-7B(E)	67.10	**71.53**	**77.49**

Table 10: Results on Core Promoter Detection (DNABERT2-Human) in GUE benchmark

Model Type	Model	Transcription Factor Prediction (Mouse)					Classification (Virus)
		0 *(MCC)*	**1** *(MCC)*	**2** *(MCC)*	**3** *(MCC)*	**4** *(MCC)*	**Covid** *(F-1)*
Baselines	DNABERT-2	56.76	84.77	79.32	66.47	52.66	71.02
	NT-2.5B-MS	63.31	83.76	71.52	69.44	47.07	73.04
	NT-500M-human	31.04	75.04	61.67	29.17	29.27	50.82
	Caduceus-Ph-131k	50.44	82.63	73.81	61.13	43.40	40.35
	HyenaDNA-160k	56.25	80.46	78.14	60.83	46.25	25.88
Our Model	HybriDNA-300M	68.57	83.46	86.02	87.96	50.58	73.81
	HybriDNA-300M(E)	68.66	85.62	85.39	87.78	51.20	73.90
	HybriDNA-3B	70.96	84.18	89.63	**88.59**	50.97	74.05
	HybriDNA-3B(E)	71.02	84.30	**89.78**	88.20	52.39	**74.88**
	HybriDNA-7B	71.68	87.75	86.59	87.62	56.47	74.02
	HybriDNA-7B(E)	**72.91**	**88.64**	87.64	**88.59**	**57.33**	74.30

Table 11: Results on Transcription Factor Prediction (DNABERT2-Mouse) and Covid Variant Classification (DNABERT2-Virus) in GUE benchmark