HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution

Eric Nguyen et. al.

Stanford University, Harvard University , SynTensor , Mila and Université de Montréal.

Genome data size

Challenge: Having both long-range context and single nucleotide resolution simultaneously

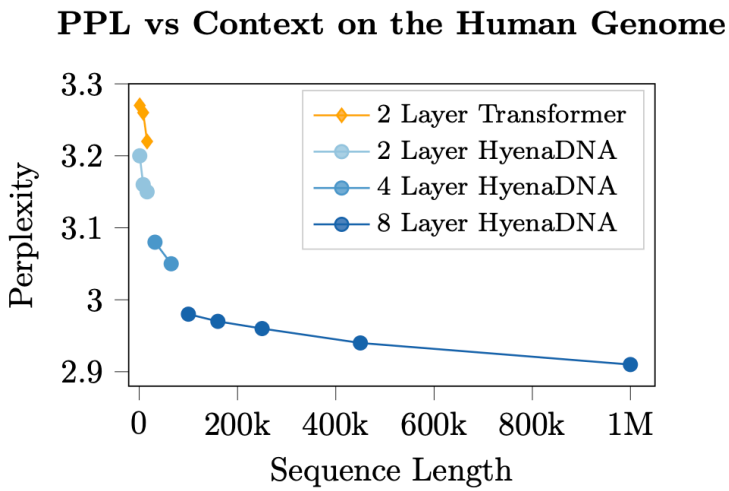


Figure 1.2: Pretraining on the human reference genome using longer sequences leads to better perplexity (improved prediction of next token).

Genome data sizes:

DNA sequences are orders of magnitudes longer (e.g. the human genome is 3.2B nucleotides)

Genomic models have relied on two strategies: i. tokenization and ii. dilation and downsampling.

i. fixed k-mers (short overlapping sequences of length k) or frequency-based byte pair encoding (BPE)

ii. average or skip elements between weights

Hyena Model

HyenaDNA recipe for long-range foundation models in genomics:

- The HyenaDNA architecture is a simple stack of Hyena operators (Poli et al., 2023) trained using next token prediction. (See Fig. 1.3 for block diagram of architecture).
- Introduce a new sequence length scheduling technique to stabilize training.
- Provide a method to leverage the longer context length to adapt to novel tasks without standard fine-tuning by filling the context window with learnable soft prompt tokens.

Hyena Core Idea

- **Self-Attention Block:** Given a length-L sequence u ∈ RL×D, Mq,Mk,Mv ∈ RD×D are learnable linear projections

$$
\begin{aligned}
\mathsf{A}(u) &= \mathsf{SoftMax}\left(\tfrac{1}{\sqrt{D}} u \mathsf{M}_q \mathsf{M}_k^\top u^\top\right) \\
y &= \mathsf{SelfAttention}(u) \\
&= \mathsf{A}(u) u \mathsf{M}_v,
\end{aligned}
$$

Remark 2.1. *Similarly to implicit convolutions,* SelfAttention *does not entangle its ability to access distant information with the number of parameters: it looks at the whole sequence at the price of* $\mathcal{O}(L^2)$ *operations.*

Hyena Core Idea

- Hyena is alternatively applying convolutions in the time and then the frequency domain (or alternatively applying element-wise products in the time and frequency domain)

Definition 3.1 (Order–N Hyena Operator)**.** *Let $(v, x^1, \cdots, x^N)$ be projections of the input and let $h^1, \dots, h^N$ be a set of learnable filters. The* Hyena$_N$ *operator is defined by the recurrence:*

$$\begin{aligned} z_t^1 &= v_t \\ z_t^{n+1} &= x_t^n (h^n * z^n)_t \quad n = 1, \dots, N \\ y_t &= z_t^{N+1} \end{aligned} \tag{4}$$

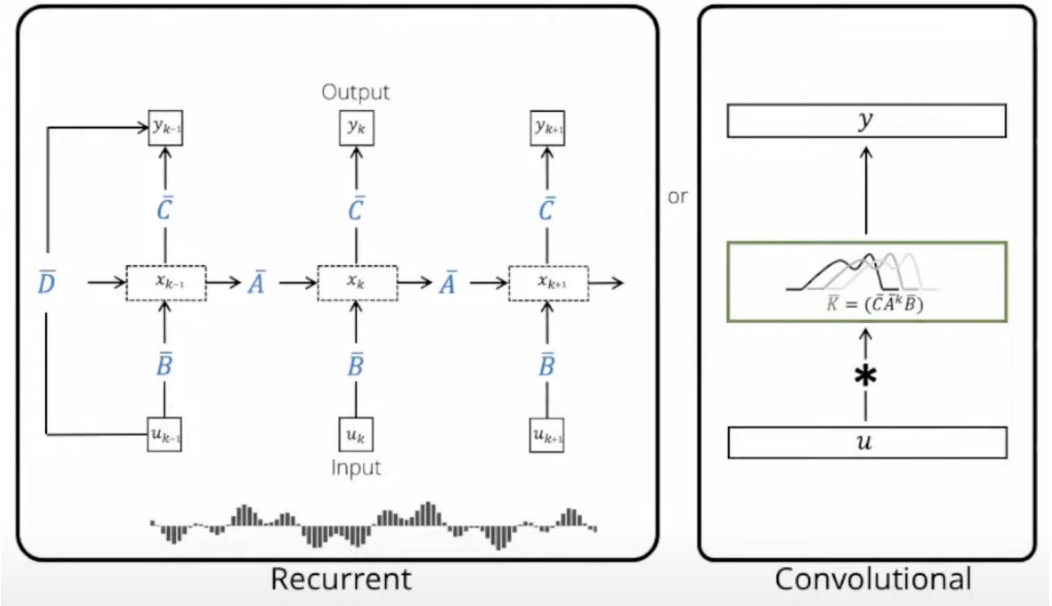
Remark 3.1. *The time complexity of a* Hyena *recurrence is $\mathcal{O}(NL \log_2 L)$. The input-output map can be rewritten as*

$$y = x^N \cdot (h^N * (x^{N-1} \cdot (h^{N-1} * (\cdots))))$$

where each convolution is performed through the Fourier domain in $\mathcal{O}(L \log_2 L)$.

Hyena Core Idea

- A view of filter h^n based on State-Space Model:



$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

Parameters

$\boldsymbol{A} \in \mathbb{R}^{N \times N}$
$\boldsymbol{B} \in \mathbb{R}^{N \times 1}$
$\boldsymbol{C} \in \mathbb{R}^{1 \times N}$
$\boldsymbol{D} \in \mathbb{R}^{1 \times 1}$
$\Delta \in \mathbb{R}$

Hyena Core Idea

- View of filter h^n based on State-Space Model:

One choice of implicit parametrization is to select h as the response function of a linear state-space model (SSM) (Chen, 1984), described by the first-order difference equation:

$$x_{t+1} = \mathsf{A}x_t + \mathsf{B}u_t \qquad \text{state equation}$$
$$y_t = \mathsf{C}x_t + \mathsf{D}u_t \qquad \text{output equation}$$

Here, the convenient choice of $x_0 = 0$ renders the input-output map to a simple convolution

$$y_t = \sum_{n=0}^{t} \left(\mathsf{C}\mathsf{A}^{t-n}\mathsf{B} + \mathsf{D}\delta_{t-n}\right) u_n$$

where δ_t denotes the Kronecker delta. We can then identify the filter h as

$$t \mapsto h_t = \begin{cases} 0 & t < 0 \\ \mathsf{C}\mathsf{A}^t\mathsf{B} + \mathsf{D}\delta_t & t \geq 0 \end{cases}$$

Hyena Core Idea

- The output of State-space Model can be thought of as convolving the input with a filter of choice

$$y_k = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^k\overline{\boldsymbol{B}}u_0 + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{k-1}\overline{\boldsymbol{B}}u_1 + \cdots + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}u_{k-1} + \overline{\boldsymbol{C}}\overline{\boldsymbol{B}}u_k$$

$$\overline{\boldsymbol{K}} \in \mathbb{R}^L := (\overline{\boldsymbol{C}}\overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}, \ldots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

$$y = \overline{\boldsymbol{K}} * u$$

Hyena Core Idea

- Also note that convolution can be thought of as matrix multiplication

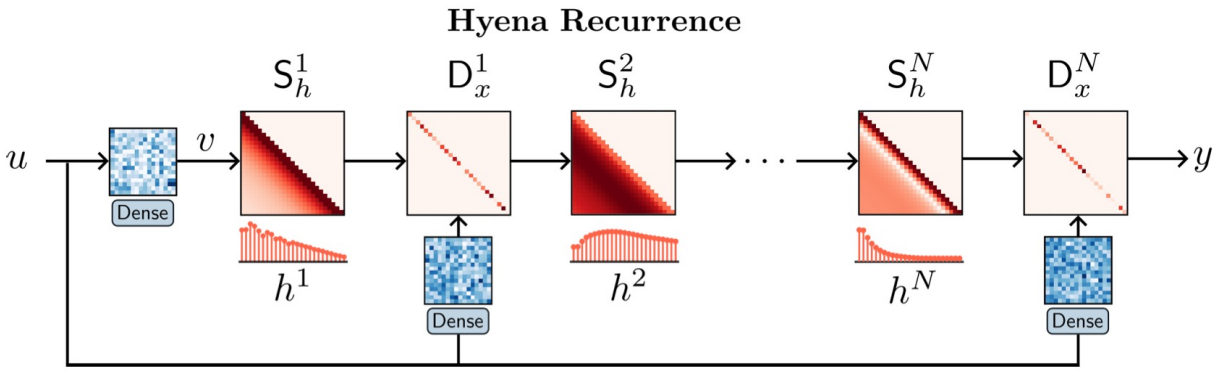
$$y_t = (h * u)_t = \sum_{n=0}^{L-1} h_{t-n} u_n.$$

$$(h * u) = \begin{bmatrix} h_0 & h_{-1} & \cdots & h_{-L+1} \\ h_1 & h_0 & \cdots & h_{-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{L-1} & h_{L-2} & \cdots & h_0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{L-1} \end{bmatrix}$$

Hyena Core Idea

- Hyena can equivalently be expressed as a multiplication with data-controlled (conditioned by the input u) diagonal matrices D^x and Toeplitz matrices S^h.

$$y = \mathsf{H}(u)v = \mathsf{D}_x^N \mathsf{S}_h^N \cdots \mathsf{D}_x^2 \mathsf{S}_h^2 \mathsf{D}_x^1 \mathsf{S}_h^1 v$$



Let Dnx = diag(xn) ∈ RL×L and let Snh be the Toeplitz matrix corresponding to filter hn.

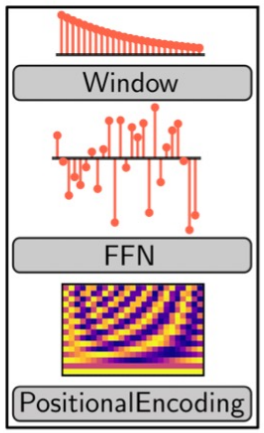
Hyena Core Idea

$$h_t = \mathsf{Window}(t) \cdot (\mathsf{FFN} \circ \mathsf{PositionalEncoding})(t) \tag{7}$$

This approach builds on the neural implicit representation literature (Mildenhall et al., 2021; Sitzmann et al., 2020), which has found application in long convolution layers (Romero et al., 2021b,a). One advantage of (7) is given by the decoupling of filter length and parameter cost.

- The filter h^n are learned by neural network for Hyena. Do we need to constrain the filter to ensure causality?

Hyena Filters h^n



Proposition 3.1 (Causal Hyenas). *If each filter h^n, $n = 1, \dots, N$ is causal, then the corresponding* Hyena_N *operator is causal.*

In practice, we need not constrain the learning of the filter (7) to ensure its *numerical* causality. If we use FFT-based convolution algorithms, all we need is to evaluate the filter at $t = 0, \dots, L-1$ and zero-pad the input and filter sequences to $2L-1$ before taking FFT.

Hyena Core Idea

- Computation if run time

Proposition 3.2 (Computational Complexity). *The computational cost of processing an input* $u \in \mathbb{R}^{L \times D}$ *with an order-*N Hyena *operator is*

$$\mathcal{O}(NDL(\log_2 L + D))$$

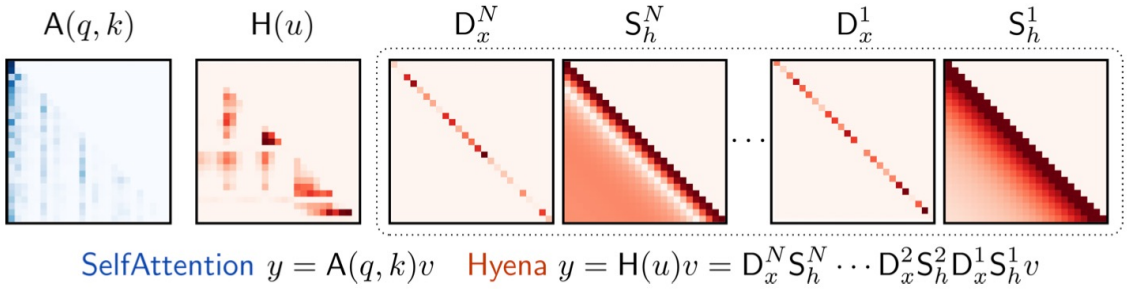


Figure 2.1: Comparison between data-controlled matrices: SelfAttention and Hyena.

Experiment with Hyena

- Hyena for Autoregressive Language Modeling as down-stream task

Table 4.3: Perplexity on WikiText103 (same tokenizer). * are results from (Dao et al., 2022c). Deeper and thinner models (Hyena-slim) achieve lower perplexity.

Model	Perplexity
Transformer (125M)	18.6
Hybrid H3 (125M)	18.5*
Performer (125M)	26.8*
Reformer (125M)	25.6*
AFT-conv (125M)	28.2
Linear Attention (125M)	25.6*
Hyena-3 (125M)	18.6
Hyena-3-slim (125M)	18.5

Table 4.4: Perplexity on The Pile for models trained until a total number of tokens e.g., 5 billion (different runs for each token total). All models use the same tokenizer (GPT2). FLOP count is for the 15 billion token run.

Model	5B	10B	15B	FLOPs (10^{19})
GPT (125M)	13.3	11.9	11.2	1.88
Hyena-2 (153M)	13.3	11.8	11.1	**1.87**
GPT (355M)	11.4	9.8	9.1	4.77
Hyena-2 (355M)	11.3	9.8	9.2	**3.93**

Experiment with Hyena

- Hyena for Genome modeling:

Table 4.1: **GenomicBenchmarks** Top-1 accuracy (%) for pretrained HyenaDNA, DNABERT and Transformer (GPT from 4.1), and the previous SotA baseline CNN (scratch).

Dataset	CNN	DNABERT	GPT	HyenaDNA
Mouse Enhancers	69.0	66.9	80.1	**85.1**
Coding vs Intergenomic	87.6	**92.5**	88.8	91.3
Human vs Worm	93.0	96.5	95.6	**96.6**
Human Enhancers Cohn	69.5	74.0	70.5	**74.2**
Human Enhancers Ensembl	68.9	85.7	83.5	**89.2**
Human Regulatory	93.3	88.1	91.5	**93.8**
Human Nontata Promoters	84.6	85.6	87.7	**96.6**
Human OCR Ensembl	68.0	75.1	73.0	**80.9**

- sequence lengths of 200-500, and one up to 4,776

Experiment with Hyena

- Hyena for Genome modeling
- 200-600 nucleotides

Table A.6: **Pretraining & Attention ablations on the Nucleotide Transformer (NT) benchmarks.** The Matthews correlation coefficient (MCC) is used as the performance metric for the enhancer and epigenetic marks dataset, and the F1-score is used for the promoter and splice site dataset.

Model	NT	GPT	HyenaDNA	HyenaDNA
Params	2.5B	1.6M	1.6M	1.6M
Pretrain	yes	yes	yes	no
Enhancer	58.0	59.3	**62.6**	58.6
Enhancer types	47.4	51.9	**55.7**	48.4
H3	81.4	75.8	**81.7**	79.9
H3K4me1	55.9	38.7	**57.1**	43.4
H3K4me2	32.6	28.8	**53.9**	34.5
H3K4me3	42.1	28.3	**61.2**	40.2
H3K9ac	57.5	49.2	**65.1**	52.6
H3K14ac	55.0	41.6	**66.3**	48.0
H3K36me3	63.2	47.8	**65.3**	53.4
H3K79me3	64.2	58.9	**71.6**	59.7
H4	**82.2**	77.7	79.6	79.1
H4ac	50.1	36.4	**63.7**	43.5
Promoter all	**97.4**	96.3	96.5	96.1
Promoter non-TATA	**97.7**	96.6	96.6	96.5
Promoter TATA	96.4	96.6	**96.7**	96.1
Splice acceptor	**99.0**	97.6	96.6	96.6
Splice donor	**98.4**	98.1	97.3	96.5
Splice all	**98.3**	98.0	97.9	97.3

References

- Nguyen, E.D., Poli, M., Faizi, M., Thomas, A.W., Birch-Sykes, C., Wornow, M., Patel, A., Rabideau, C.M., Massaroli, S., Bengio, Y., Ermon, S., Baccus, S.A., & Ré, C. (2023). HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. *ArXiv*.

- Poli, M., Massaroli, S., Nguyen, E.Q., Fu, D.Y., Dao, T., Baccus, S.A., Bengio, Y., Ermon, S., & Ré, C. (2023). Hyena Hierarchy: Towards Larger Convolutional Language Models. *International Conference on Machine Learning*.