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

Eric Nguyen[*,1], Michael Poli[*,1], Marjan Faizi[2,*],
Armin W. Thomas[1], Callum Birch Sykes[3], Michael Wornow[1], Aman Patel[1],
Clayton Rabideau[3], Stefano Massaroli[4], Yoshua Bengio[4], Stefano Ermon[1],
Stephen A. Baccus[1,†], Christopher Ré[1,†]

November 15, 2023

Deep Learning in Genomics Reading Group

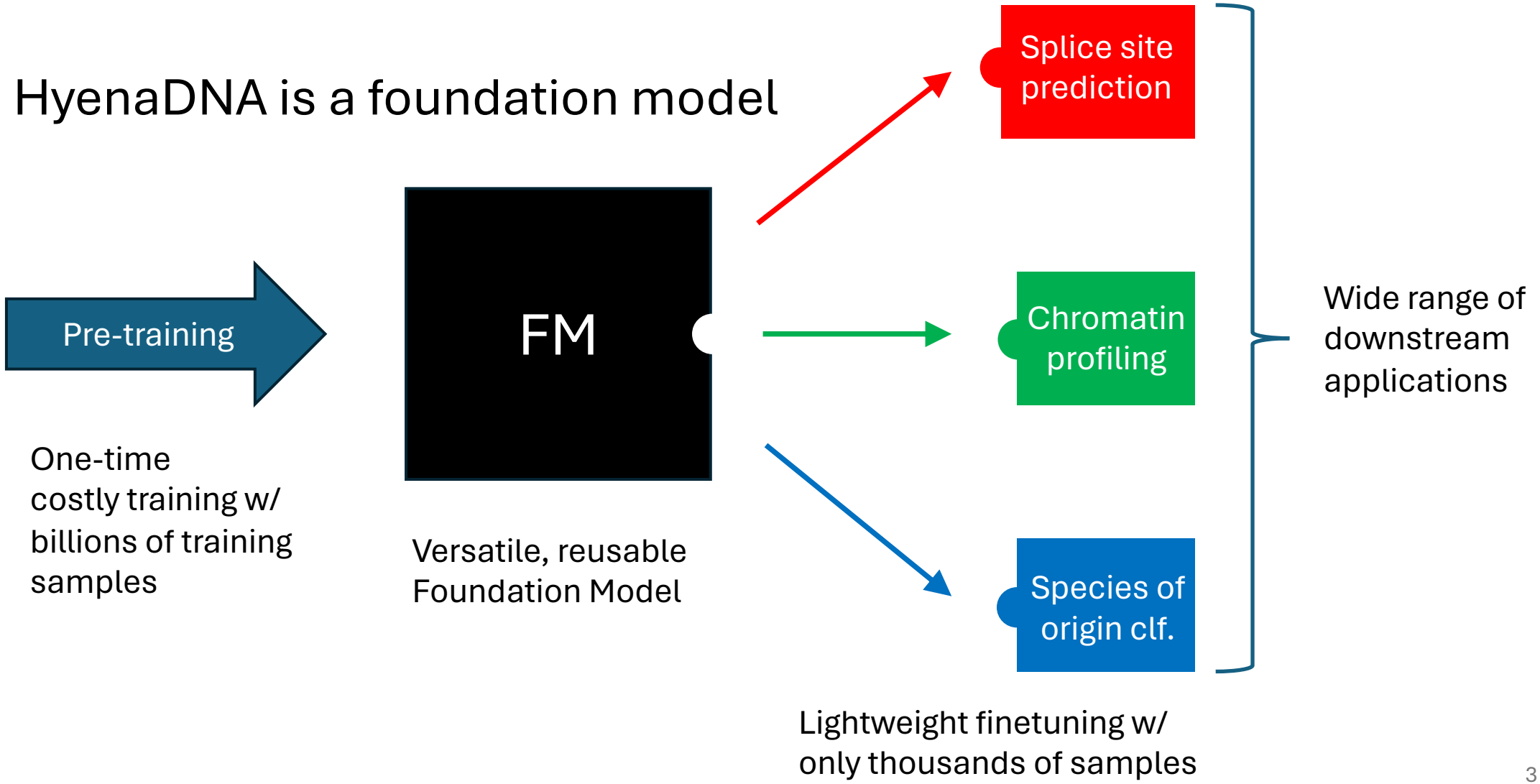
5 November 2024

Abstract

Genomic (DNA) sequences encode an enormous amount of information for gene regulation, protein synthesis, and numerous other cellular properties. Similar to natural language models, researchers have proposed foundation models in genomics to learn generalizable features from unlabeled genome data that can then be fine-tuned for downstream tasks such as identifying regulatory elements. Due to the quadratic scaling of attention, previous Transformer-based genomic models have used 512 to 4k tokens as context (<0.001% of the human genome), significantly limiting the modeling of long-range interactions in DNA. In addition, these methods rely on tokenizers or fixed k-mers to aggregate meaningful DNA units, losing single nucleotide resolution (i.e. DNA "characters") where subtle genetic variations can completely alter protein function via single nucleotide polymorphisms (SNPs). Recently, Hyena, a large language model based on implicit convolutions was shown to match attention in quality while allowing longer context lengths and lower time complexity. Leveraging Hyena's new long-range capabilities, we present HyenaDNA, **a genomic foundation model** pretrained on the human reference genome with **context lengths of up to 1 million tokens at the single nucleotide-level** – an **up to 500x increase** over previous dense attention-based models. HyenaDNA scales sub-quadratically in sequence length (training up to 160x faster than Transformer), **uses single nucleotide tokens**, and has **full global context at each layer**. We explore what longer context enables - including the first use of in-context learning in genomics for simple adaptation to novel tasks without updating pretrained model weights. On a long-range species classification task, HyenaDNA is able to effectively solve the challenge by increasing the context length to 1M without downsampling. On fine-tuned benchmarks from the Nucleotide Transformer, HyenaDNA reaches state-of-the-art (SotA) on 12 of 18 datasets using a model with orders of magnitude less parameters and pretraining data.[1] On the GenomicBenchmarks, HyenaDNA surpasses SotA on 7 of 8 datasets on average by +10 accuracy points, and by as much as +20 accuracy points on enhancer identification. Code available at https://github.com/HazyResearch/hyena-dna.

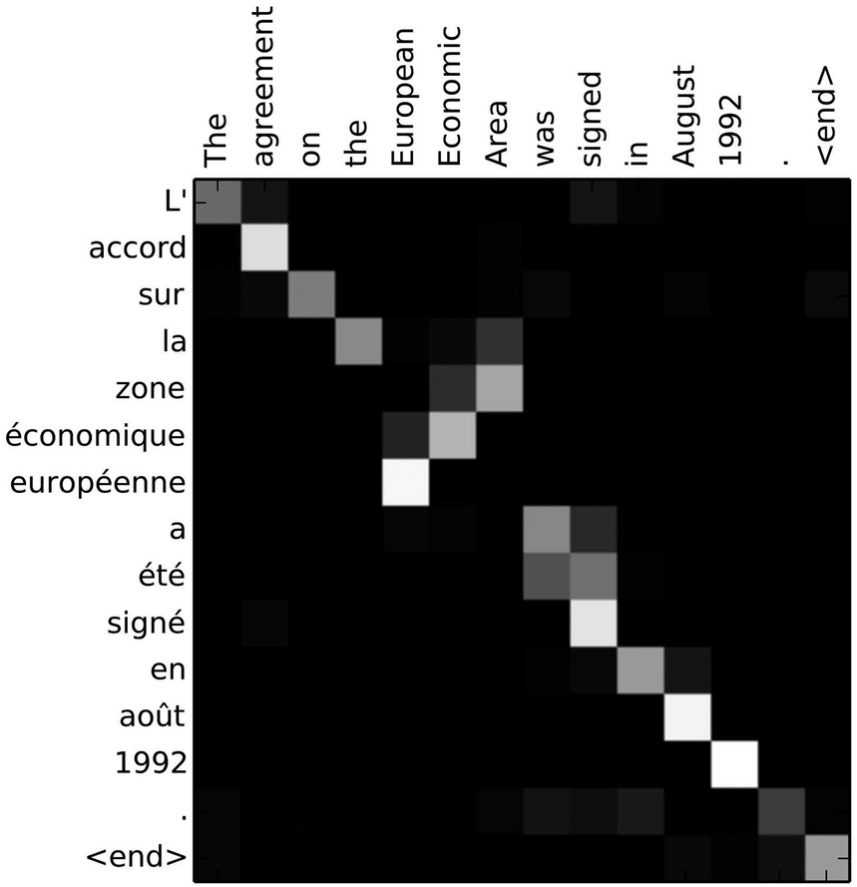
Foundation models (FMs)

- HyenaDNA is a foundation model



Limitation of current genomic LLMs

- Genomic sequences are long, relevant info may be far apart
- Attention is a **quadratic** operation
  - 1K tokens = 1M operations
  - 1M tokens = 1T operations
- Current methods to address this:
  1. Tokenization
  2. Dilation / downsampling
  - e.g. Enformer and Borzoi dilate via convolutions and pooling
  - Both methods lose information



https://www.researchgate.net/figure/A-visualization-of-attention-weights-showing-soft-alignment-between-source-and-target_fig1_332160864

A possible solution

Hyena Hierarchy: Towards Larger Convolutional Language Models

Michael Poli [*1] **Stefano Massaroli** [*2] **Eric Nguyen** [*1]
Daniel Y. Fu [1] **Tri Dao** [1] **Stephen Baccus** [1] **Yoshua Bengio** [2] **Stefano Ermon** [†1] **Christopher Ré** [†1]

Abstract

Recent advances in deep learning have relied heavily on the use of large Transformers due to their ability to learn at scale. However, the core building block of Transformers, the attention operator, exhibits quadratic cost in sequence length, limiting the amount of context accessible. Existing subquadratic methods based on low-rank and sparse approximations need to be combined with dense attention layers to match Transformers, indicating a gap in capability. In this work, we propose **Hyena**, a subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized **long convolutions** and **data-controlled gating**. In recall and reasoning tasks on sequences of thousands to hundreds of thousands of tokens, Hyena improves accuracy by more than 50 points over operators relying on state-spaces and other implicit and explicit methods, matching attention-based models. We set a new state-of-the-art for dense-attention-free architectures on language modeling in standard datasets (WikiText103 and The Pile), reaching Transformer quality with a 20% reduction in training compute required at sequence length 2K. Hyena operators are twice as fast as highly optimized attention at sequence length 8K, and 100× faster at sequence length 64K.

Poli et al. *Proceedings of the 40th International Conference on Machine Learning* (2023)

A quick note

- Disclaimer: the math is too complicated for me
- I will try to share my intuition rather than concrete proofs
- For more information, I recommend these sources:
  - Post-Transformers – Hyena Hierarchy, Alex Mackenzie
  - The Annotated Hyena, Jonathan Skowera
  - Hyena Hierarchy: Towards Larger Convolutional Language Models, Michael Poli et al.

The Hyena Operator

- Hyena is a “subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized long convolutions and data-controlled gating”

Let’s break down this definition…

Poli et al. *Proceedings of the 40th International Conference on Machine Learning* (2023)

The Hyena Operator

- Hyena is a "**subquadratic** drop-in replacement for attention constructed by interleaving implicitly parametrized long convolutions and data-controlled gating"

Subquadratic: For an input sequence of length L,
Hyena is $O(L \log_2 L)$ rather than $O(L^2)$

Achieved through a Fast Fourier Transform (FFT)

1M inputs = 20M rather than 1T operations!

The Hyena Operator

- Hyena is a “subquadratic **drop-in replacement for attention** constructed by interleaving implicitly parametrized long convolutions and data-controlled gating”

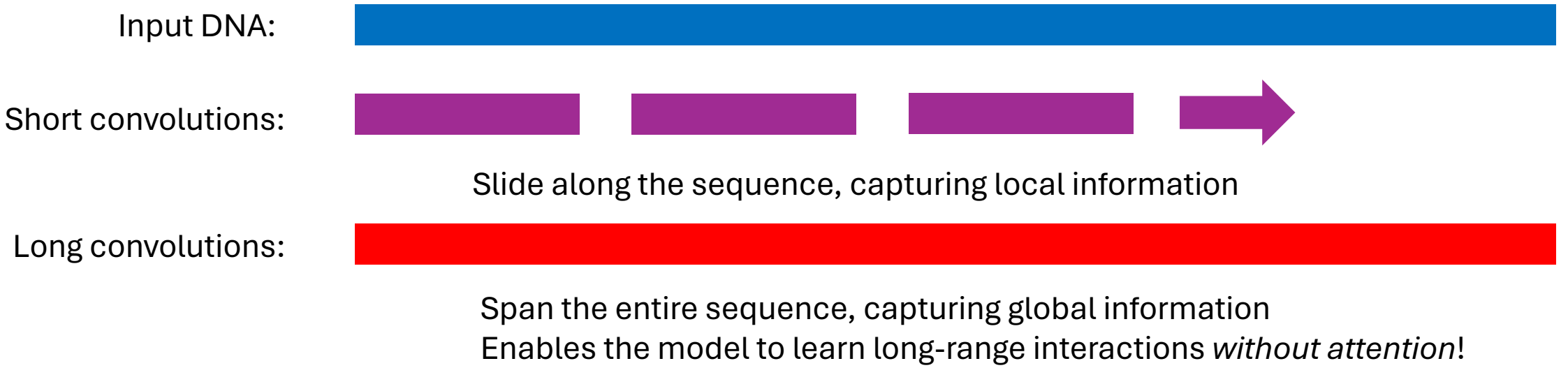
Drop-in replacement for attention

=

can substitute for an attention module in an LLM

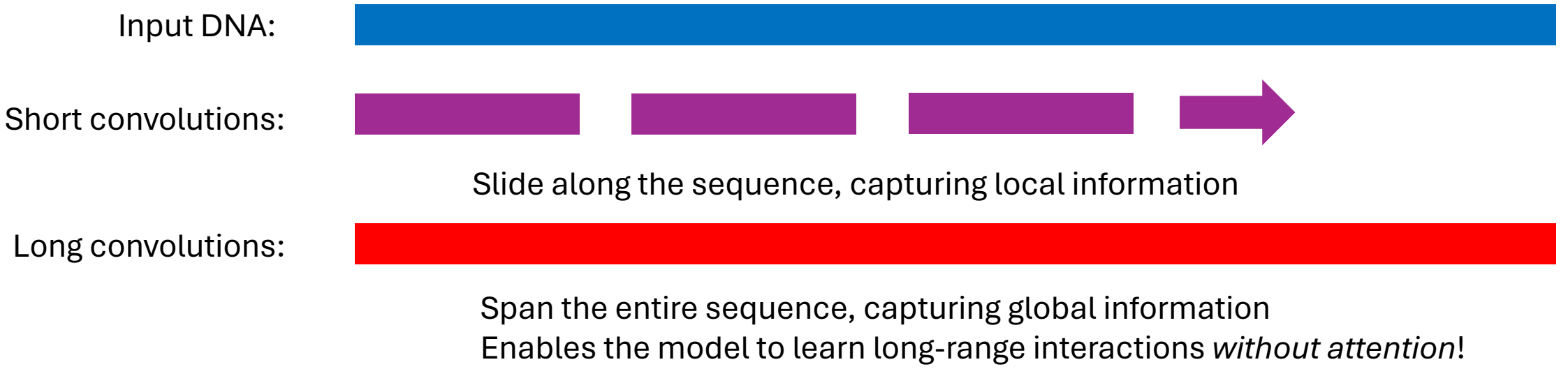
The Hyena Operator

- Hyena is a “subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized **long convolutions** and data-controlled gating”



The Hyena Operator

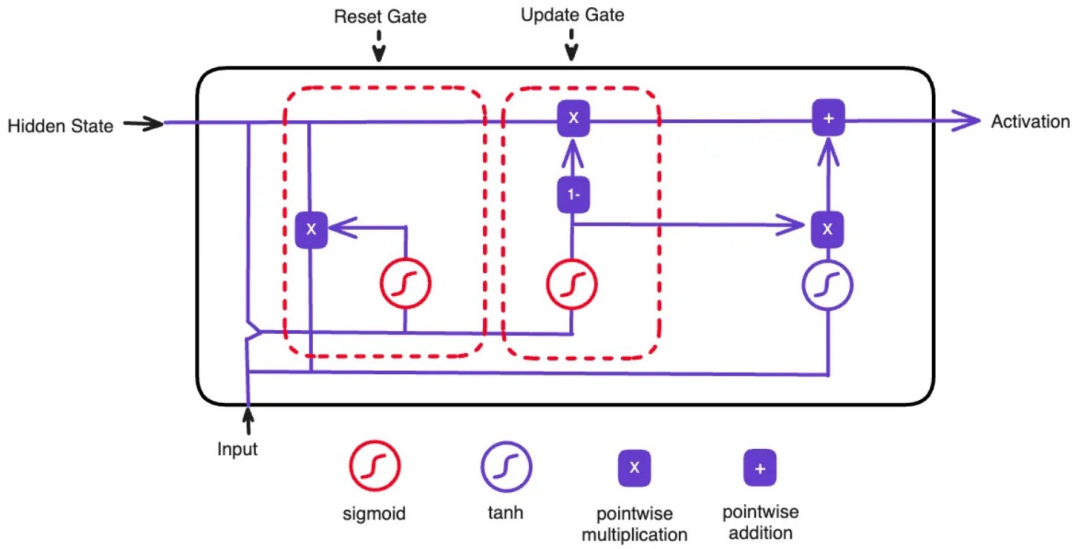
- Hyena is a “subquadratic drop-in replacement for attention constructed by interleaving **implicitly parametrized** long convolutions and data-controlled gating”



Implicitly parameterized = filters are learned rather than hard-coded

The Hyena Operator

- Hyena is a “subquadratic drop-in replacement for attention constructed by interleaving implicitly parametrized long convolutions and **data-controlled gating**”



Controls how much of each data point is passed to the next layer

The Hyena Operator

- Hyena is a “subquadratic drop-in replacement for attention **constructed by interleaving** implicitly parametrized long convolutions and data-controlled gating”

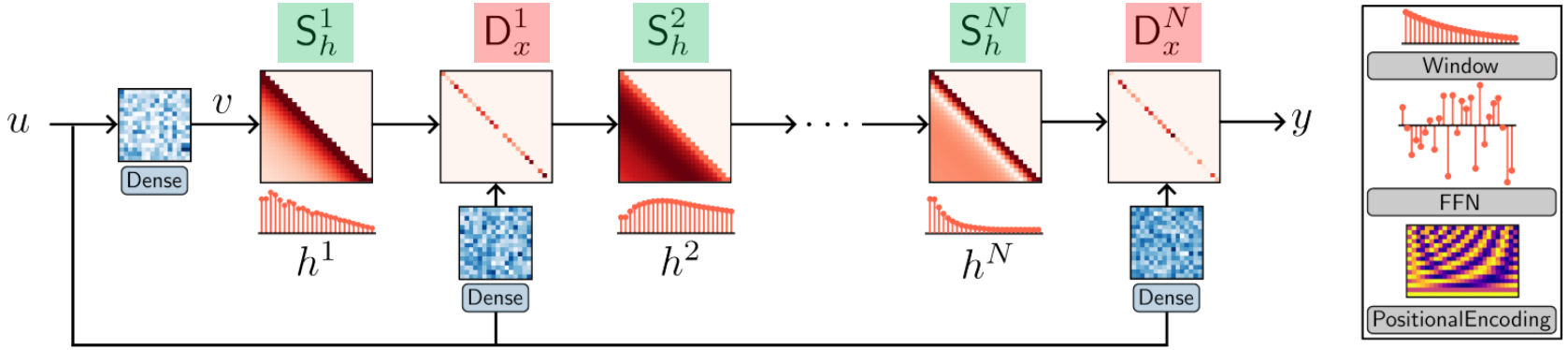


Figure 1. The Hyena operator is defined as a recurrence of two efficient subquadratic primitives: an implicit long convolution h (i.e. Hyena filters parameterized by a feed-forward network) and multiplicative element-wise gating of the (projected) input. The depth of the recurrence specifies the size of the operator. 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. In addition, Hyena exhibits sublinear parameter scaling (in sequence length) and unrestricted context, similar to attention, while having lower time complexity.

Poli et al. *Proceedings of the 40th International Conference on Machine Learning* (2023)

Hyena intuition

- Hyena produces several (3 or more) linear projections of the input
- Analogous to Key, Query, Value matrices of attention
- These projections contain biologically meaningful information at a global scale

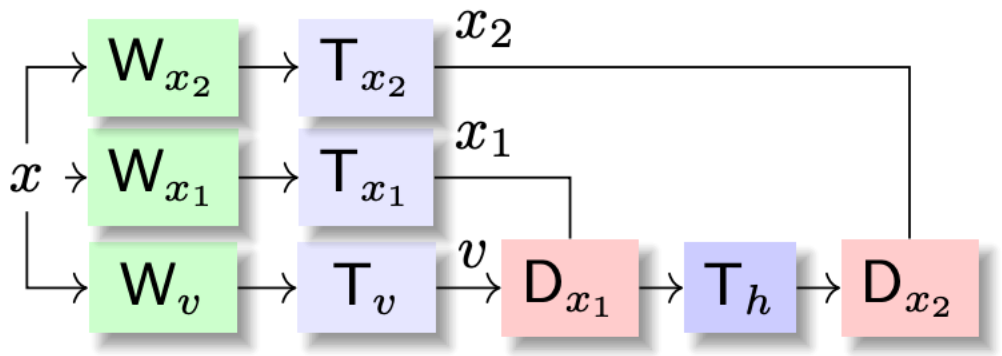


Figure 3.1: The Hyena operator is a combination of long convolutions T and data-controlled gating D, and can be a drop-in replacement for attention.

Paper overview

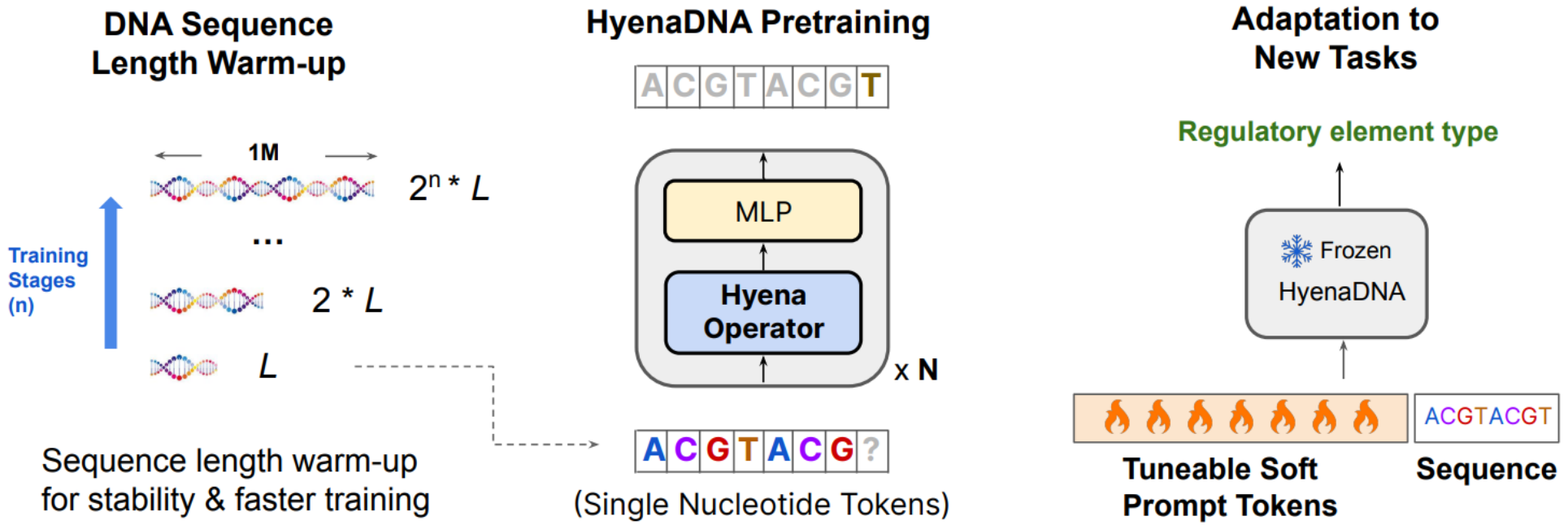
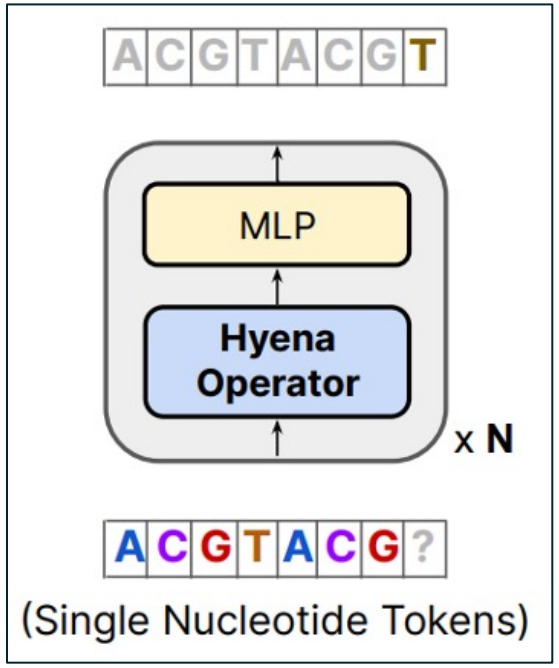


Figure 1.1: HyenaDNA recipe for long-range foundation models in genomics. The HyenaDNA architecture is a simple stack of Hyena operators (Poli et al., 2023) trained using next token prediction. (See Fig. 1.3 for block diagram of architecture). We introduce a new sequence length scheduling technique to stabilize training, and 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.

HyenaDNA architecture

- Decoder-only
- Seq-to-seq



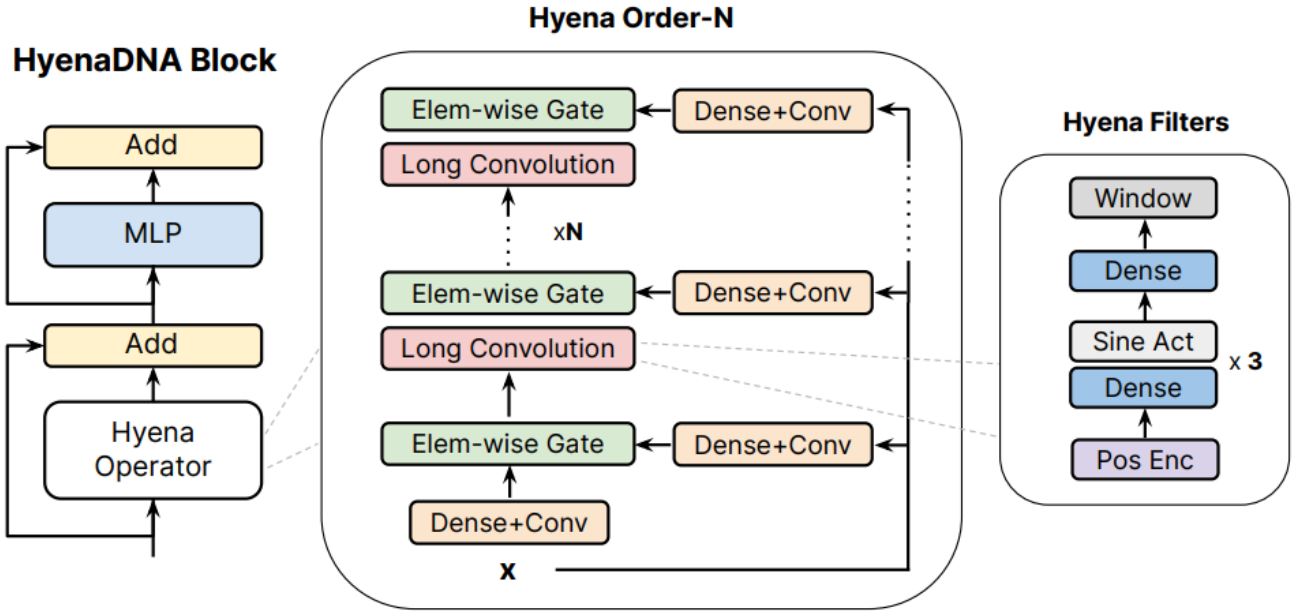


Figure 1.3: HyenaDNA block architecture. A Hyena operator is composed of long convolutions and element-wise gate layers. The gates are fed projections of the input using dense layers and short convolutions. The long convolutions are parameterized *implicitly* via an MLP that produces the convolutional filters. The convolution itself is evaluated using a Fast Fourier Transform convolution with time complexity $\mathcal{O}(L \log_2 L)$.

Training Long Sequence Models

- Training directly on long sequences affects **stability**
- Instead, begin with **short** sequences and **scale up**
- Start with **64bp** sequences and **double** the length at each step
  - While keeping the batch size the same
- Warmup results in better performance with less training

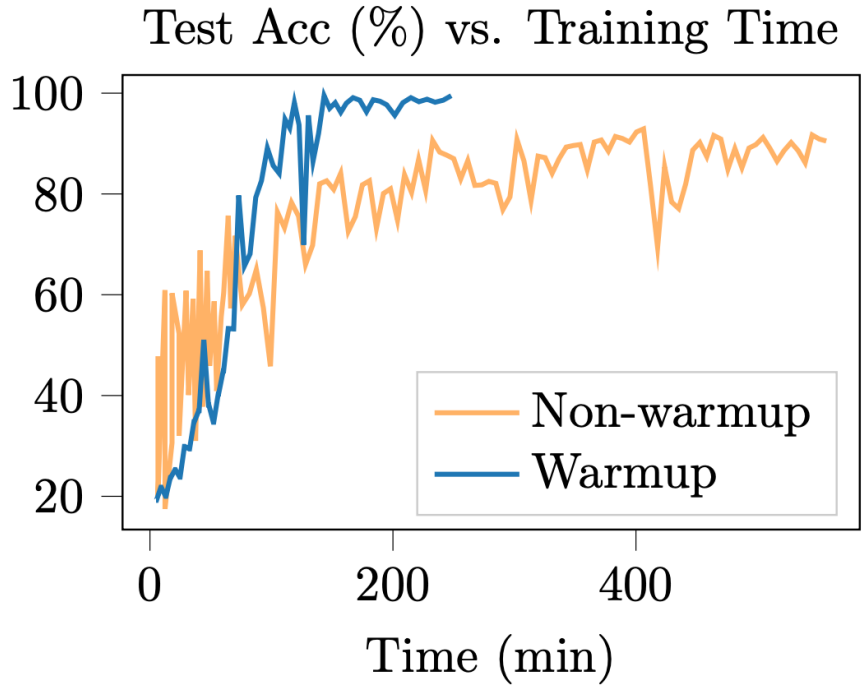


Figure 3.2: Sequence length warm-up reduces the training time of HyenaDNA at sequence length 450k by 40% and boosts accuracy by 7.5 points on species classification.

Pretraining

- **Data**: 160kbp seqs from the human reference genome
- **Proxy task**: next nucleotide prediction
- **Architecture**: decoder-only transformers
  - Hyena operators instead of attention
- **Result**: much faster inference at longer contexts

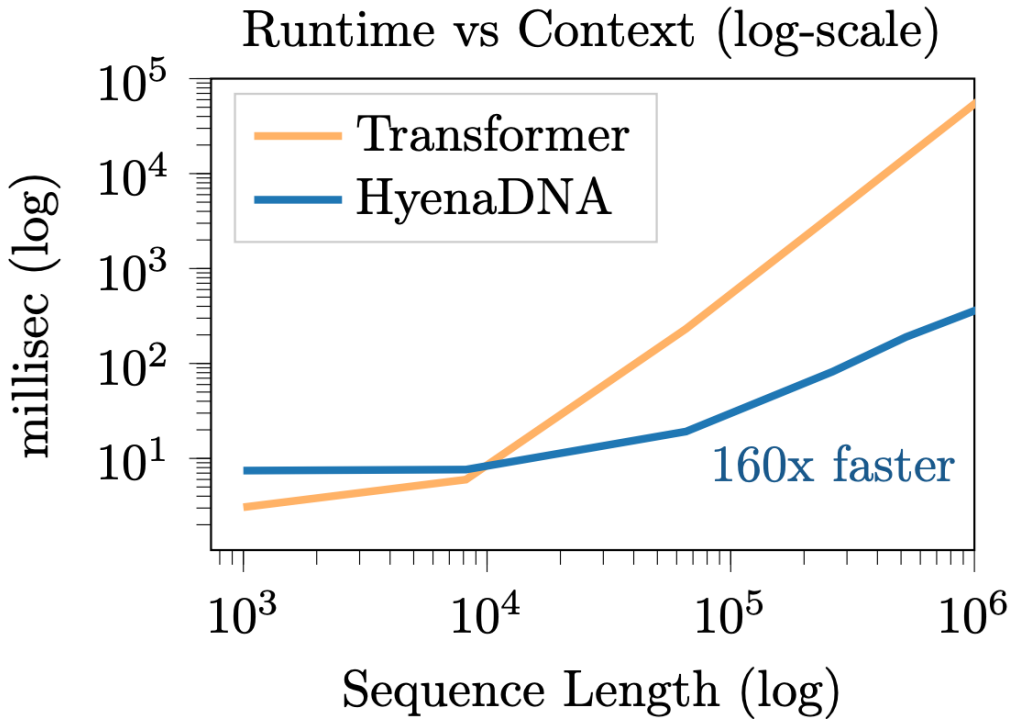


Figure 4.1: Runtime (forward & backward pass) for Transformer and HyenaDNA: 2 layers, width=128, gradient checkpointing, batch size=1, A100 80GB. At 1M tokens HyenaDNA is **160x faster** than Transformer.

Final pretraining recipe

Layers	2	2	4	4	8
Width	128	256	128	256	256
Params (M)	0.44	1.6	0.87	3.3	6.6
Max seq. len.	64k	64k	64k	64k	1M
Optimizer	AdamW				
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$				
Learning rate	1.5 - 6e-4				
LR Scheduler	Cosine decay				
Batch size	64 - 256				
Global steps	10 - 20k				
Weight decay (model)	0.1				
Weight decay (Hyena layers)	0				
Embed dropout	0.1				
Residual dropout	0				

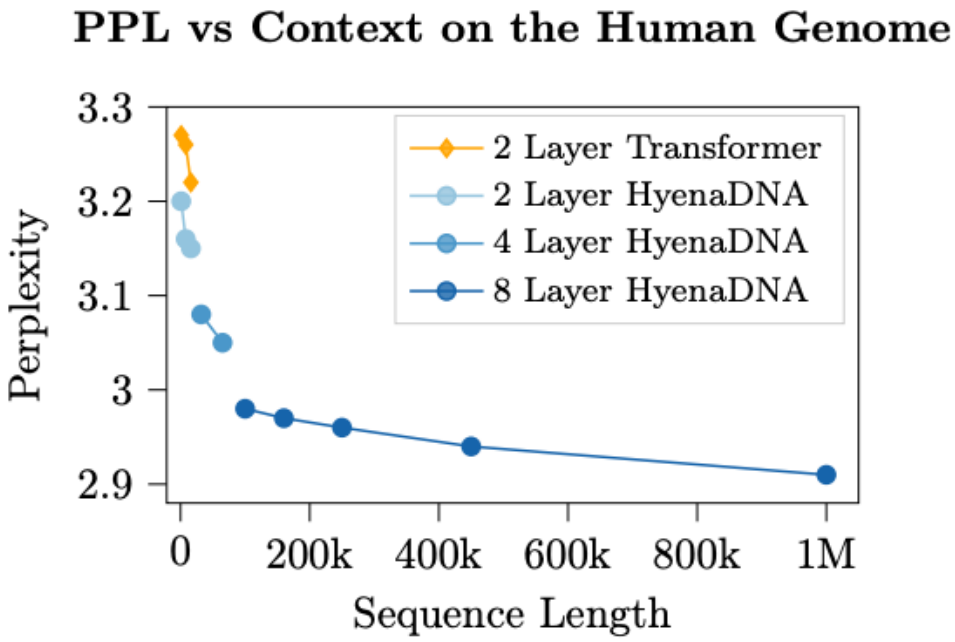


Figure 1.2: Pretraining on the human reference genome using longer sequences leads to better perplexity (improved prediction of next token).

Short-length test 1

- Finetune on 8 regulatory element classification datasets
- HyenaDNA reaches SoTA on 7/8 tasks

- 1k - 300k seqs
- 200 - 2000bp

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**

Name	# of sequences	# of classes	Class ratio	Median length	Standard deviation
dummy_mouse_enhancers_ensembl	1210	2	1.0	2381	984.4
demo_coding_vs_intergenomic_seqs	100000	2	1.0	200	0.0
demo_human_or_worm	100000	2	1.0	200	0.0
~~drosophila_enhancers_stark~~	~~6914~~	~~2~~	~~1.0~~	~~2142~~	~~285.5~~
human_enhancers_cohn	27791	2	1.0	500	0.0
human_enhancers_ensembl	154842	2	1.0	269	122.6
human_ensembl_regulatory	289061	3	1.2	401	184.3
human_nontata_promoters	36131	2	1.2	251	0.0
human_ocr_ensembl	174756	2	1.0	315	108.1

Gresova et al. *BMC Genomic Data* (2023)

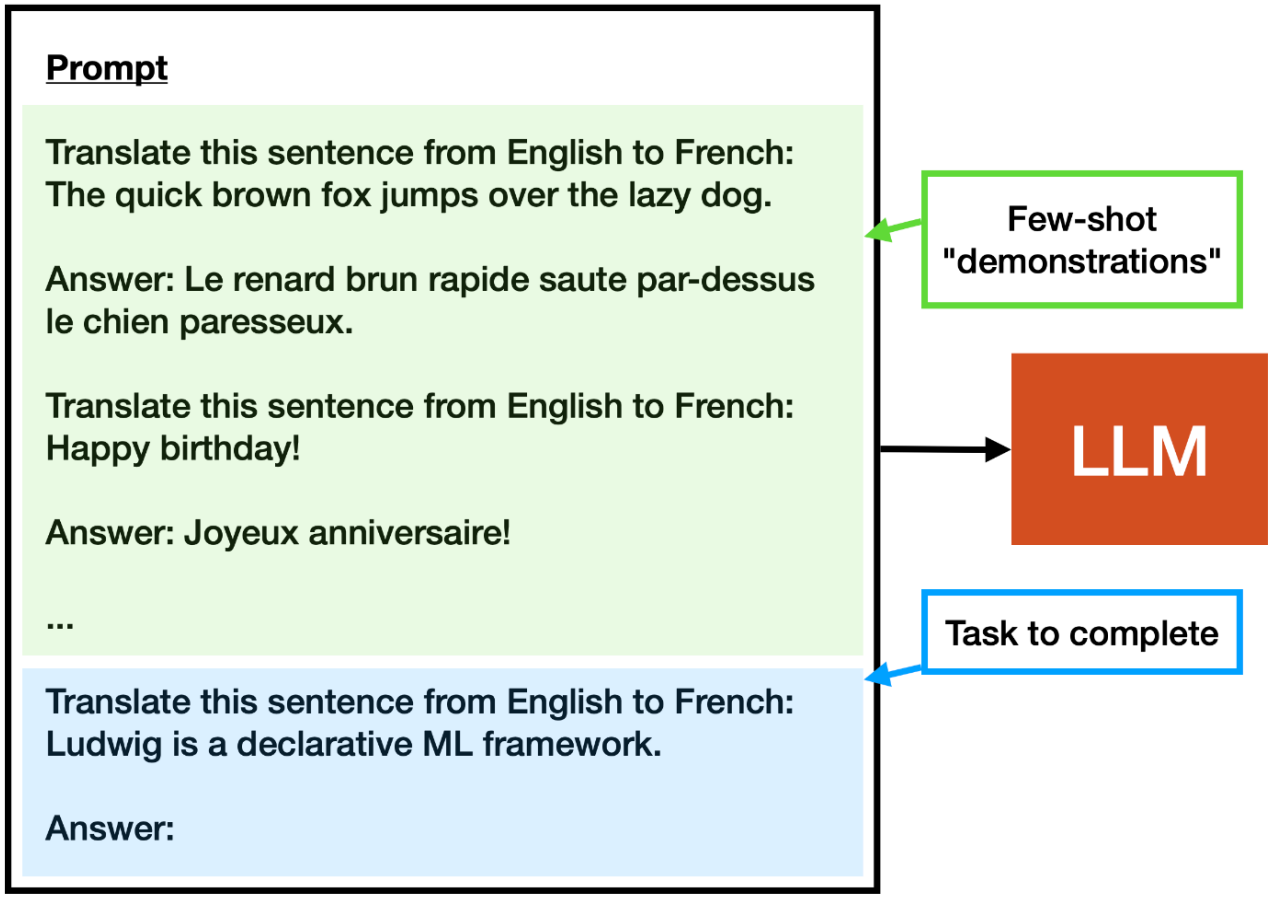
Short-length test 2

- Compare against Nucleotide Transformer, a SoTA model
  - BERT-based architecture
  - 6mer tokenization of 1kbp sequences
  - 500M – 2.5B parameters
  - 1 – 3200 genome pre-training
- 18 finetuning tasks
  - 200-600 bp seqs
  - Reg. elts, splice sites, etc.
- HyenaDNA excels

Table 4.2: **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	NT	NT	HyenaDNA
Params	500M	2.5B	2.5B	1.6M
# of Genomes	1	3,202	850	1
Enhancer	53.5	59.3	58.0	**62.6**
Enhancer types	48.5	50.0	47.4	**55.7**
H3	73.7	77.6	81.4	**81.7**
H3K4me1	35.8	44.5	55.9	**57.1**
H3K4me2	28.1	30.0	32.6	**53.9**
H3K4me3	26.3	28.1	42.1	**61.2**
H3K9ac	46.2	50.8	57.5	**65.1**
H3K14ac	37.7	47.1	55.0	**66.3**
H3K36me3	46.7	53.3	63.2	**65.3**
H3K79me3	57.7	59.2	64.2	**71.6**
H4	76.2	78.9	**82.2**	79.6
H4ac	34.4	42.3	50.1	**63.7**
Promoter all	95.4	96.6	**97.4**	96.5
Promoter non-TATA	95.6	96.9	**97.7**	96.6
Promoter TATA	94.8	95.8	96.4	**96.7**
Splice acceptor	96.5	98.5	**99.0**	96.6
Splice donor	97.2	98.2	**98.4**	97.3
Splice all	97.2	97.8	**98.3**	97.9

In-context learning (ICL)



- Unlock model abilities without finetuning
- Pass in question/answer pairs and a final question
- Model learns the pattern just by parsing the inputs
- Widely used for natural language tasks

https://ludwig.ai/latest/user_guide/llms/in_context_learning/

ICL example with ChatGPT

Prompt: elephant
Answer: red
Prompt: moose
Answer: red
Prompt: frog
Answer: blue
Prompt: trout
Answer: blue
Prompt: gorilla
Answer: red
Prompt: turkey
Answer: blue
Prompt: mouse
Answer:

red

what is the rule in the above game?

The rule seems to be based on whether the animal is typically considered a type of mammal or not. Mammals like elephants, moose, gorillas, and mice are given the color red, while non-mammals like frogs, trout, and turkeys are given the color blue.

In-context learning for genomics

- Main **challenge** is working with HyenaDNA's **limited vocabulary**
  - Vocab is {A, C, G, T, N, [PAD], [SEP], [UNK]}
  - How do you convey a question/answer pair?

Prompt: AAACTGA
Answer: reference
Prompt: AAATTGA
Answer: benign mutation
Prompt: AAAGTGA
Answer: harmful mutation
Prompt: AAACTGA
Answer: ?

Ideal

~~Prompt:~~ AAACTGA
~~Answer: reference~~
~~Prompt:~~ AAATTGA
~~Answer: benign mutation~~
~~Prompt:~~ AAAGTGA
~~Answer: harmful mutation~~
~~Prompt:~~ AAACTGA
~~Answer:~~ ?

Reality

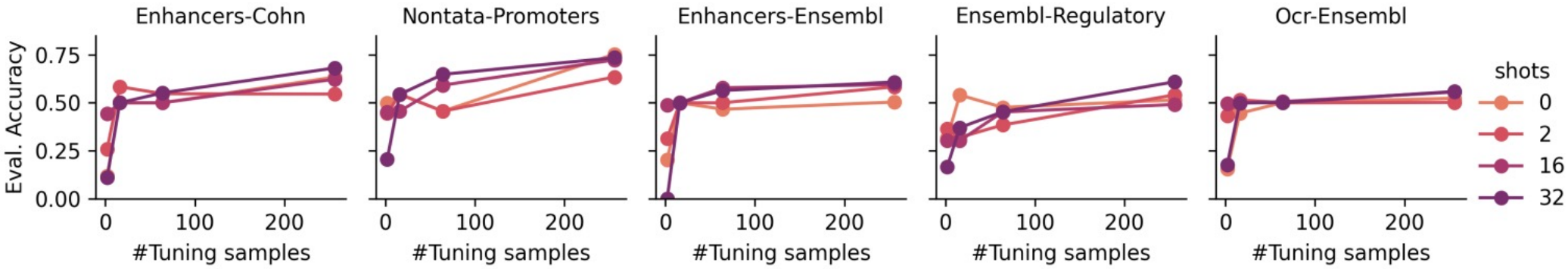
ICL for genomics: Few-shot prompting

Few-shot prompting details For each dataset, we prepend a set of k (0 to 32, 0 indicates regular fine-tuning) examples of each class of a dataset (so-called "shots") to an input sequence:

$$X: \quad \{X_1, \mathsf{SEP}, Y_1, \mathsf{SEP}, X_2, \mathsf{SEP}, Y_2, \mathsf{SEP}, X, \mathsf{SEP}\},$$

where X_i indicates an example sequence of class i with label Y_i (exemplified for a two-way classification task). We tune the model on n (2 to 256) such k-shot samples before evaluating its performance on the dataset's full validation data. For an overview of the results of this experiment, see Fig. A.1.

- Y labels for binary classification are **A** or **N**, same as model output prediction



- Performance increases with # of **examples**, but not necessarily number of **shots**

ICL for genomics: instruction tuning

- Prepend 2 – 32k tokens to prompt

- Tokens are **tunable**, and tuned for up to 20 epochs
- Performance increases with the **number** of prepended **tokens**

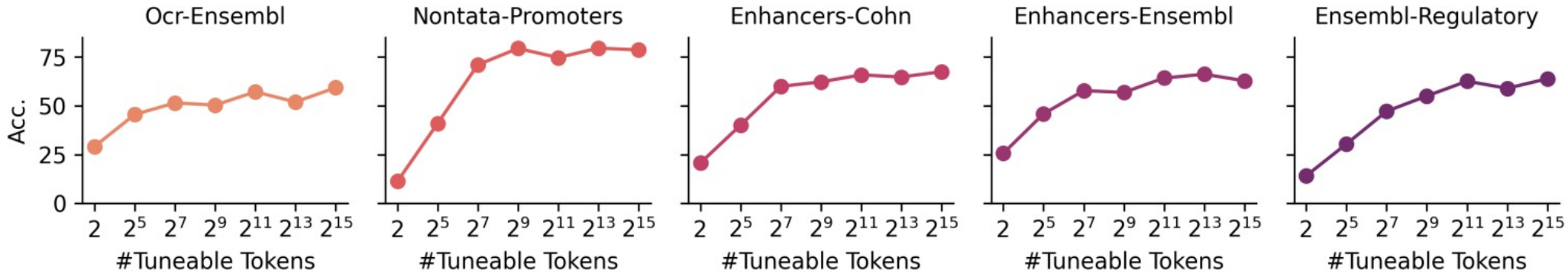


Figure 4.2: **Filling long-context with soft tuneable tokens.** HyenaDNA is able to learn new tasks in-context when adding a sequence of tuneable tokens to the input sequences. Longer sequences of tuneable tokens lead to better performance.

Long-context predictions

- Predict **919** chromatin features from DeepSEA dataset
  - TF binding profiles, histone mark profiles, DNase I-hypersensitivity sites
- **8kbp** length sequences
- **2M** training samples, **200k** test
- Try **1kbp** and **8kbp** inference
- HyenaDNA outperforms SoTA with **10x fewer** parameters

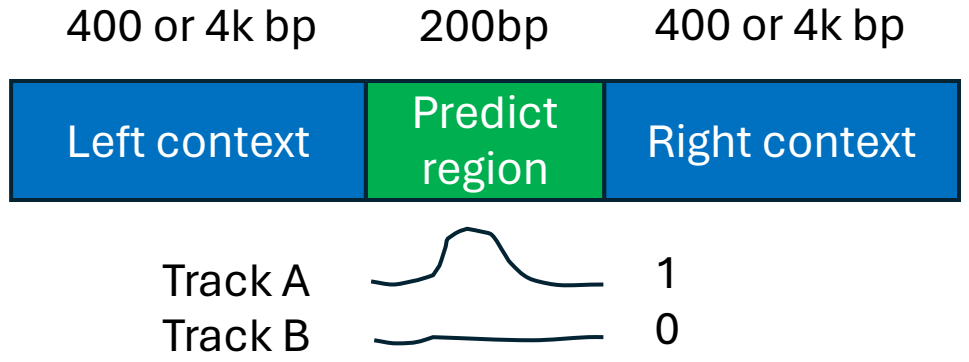


Table 4.3: **Chromatin profile prediction** Median AUROC computed over three categories: Transcription factor binding profiles (TF), DNase I-hypersensitive sites (DHS) and histone marks (HM).

Model	Params	Len	AUROC TF	AUROC DHS	AUROC HM
DeepSEA	40 M	1k	95.8	92.3	85.6
BigBird	110 M	8k	96.1	92.1	88.7
HyenaDNA	7 M	1k	**96.4**	**93.0**	86.3
	3.5 M	8k	95.5	91.7	**89.3**

Embedding comparison

- Compare **token embeddings** of SoTA models
- DNABERT – 6mers
- NT – byte-pair encoding
- HyenaDNA – single nucleotide
- HyenaDNA learns **biologically meaningful** features

Table 4.4: **Embedding quality** Weighted F1 classification score on 10 biotypes.

Model	Params	Len	F1
DNABERT	110 M	512	64.6
NT	500 M	6k	66.5
HyenaDNA	7 M	160k	**72.0**

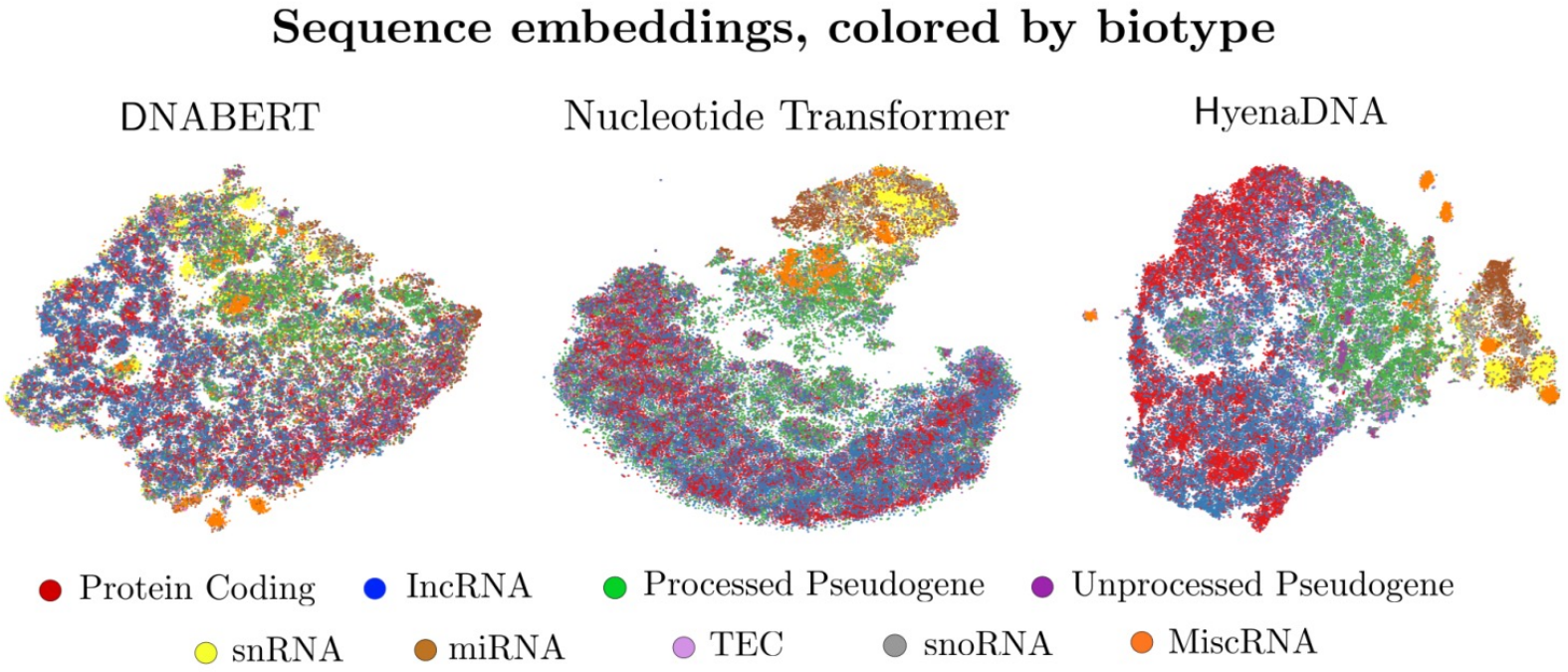


Figure 4.3: **Embedding visualisation.** t-SNE of the embeddings generated by DNABERT, Nucleotide Transformer and HyenaDNA coloured by Ensembl biotype annotations.

Ultralong sequence species classification

- Classify species **origin** of sequences
- Randomly selected from **5 species**
  - Human, lemur, mouse, pig, hippo
  - Closely related species are hard to distinguish within small windows
- HyenaDNA excels on **long sequences** up to **1M** bases
- Transformers cannot compete due to **computational cost**

Table 4.5: **Species classification** Top-1 accuracy (%) for 5-way classification (human, lemur, mouse, pig, hippo). The ✗ symbol indicates infeasible training time.

Model	Len	Acc
Transformer	1k	55.4
HyenaDNA	1k	61.1
Transformer	32k	88.9
HyenaDNA	32k	93.4
Transformer	250k	✗
HyenaDNA	250k	97.9
Transformer	450k	✗
HyenaDNA	450k	99.4
Transformer	1M	✗
HyenaDNA	1M	**99.5**

HyenaDNA strengths

- Only SoTA model with **single-nucleotide resolution**
- Outperforms other SoTA models despite many **fewer parameters** and **smaller training** sets
- Inference on ultralong sequences of **1M** bases
  - Unlocks abilities on sequence classification and long-range interactions
- Hyena, in-context-learning can be used in any genomic LLM

HyenaDNA limitations

- No interpretability analyses
  - What is the model actually learning?
- Very shallow model, only 2-8 layers
  - Can complex features be learned?

Future work

- Train on more genomes
- Multi-modal training data such as protein sequences
- Increased model size and compute budget

Next meeting

- **Date and Time**: Tuesday, November 19th, 12 - 1pm
- **Location**: Malone 228 and Zoom
- **Presenter**: TBD, volunteers needed!

Sign up to present this semester!!! →