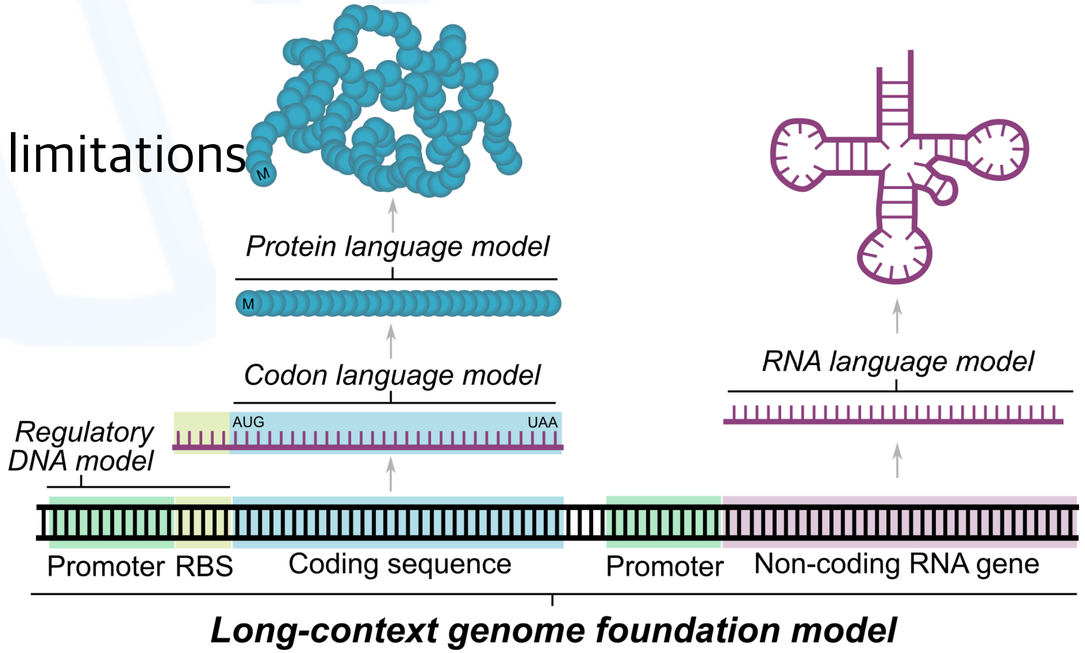
Sequence modeling and design from molecular to genome scale with Evo

Eric Nguyen, Michael Poli, Matthew G. Durrant, Armin W. Thomas, Brian Kang, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aman Patel, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, Brian L. Hi

03.07.24 Teresa Lin
Knowles Lab Journal Club

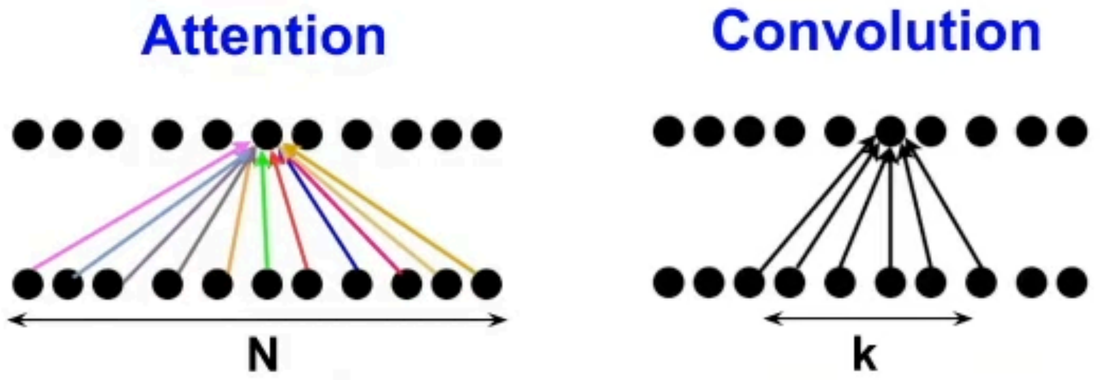
Is DNA all you need?

- Existing breakthroughs in modeling biological sequences with AI have focused on short context, task-specific, and single-modality capabilities
- Can we build a model that do everything all at once?
- A lot of successful language models are using autoregressive Transformers
  - But the computational cost is high
  - Attention-based models have similar resolution limitations



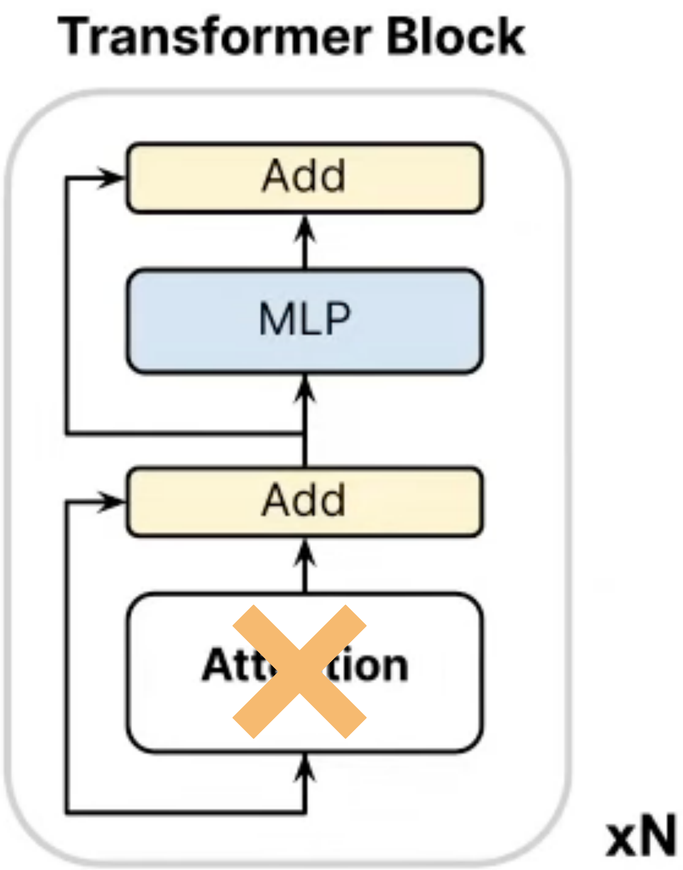
Attention is great, but it's expensive!

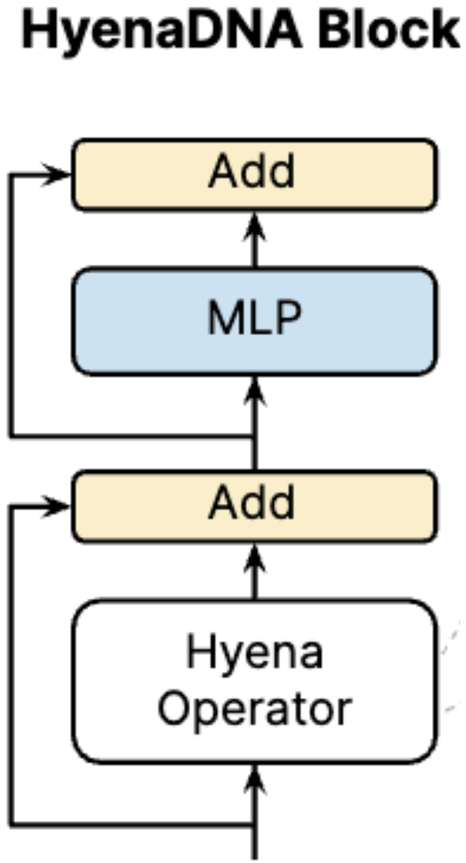
Can we get the best of both worlds?



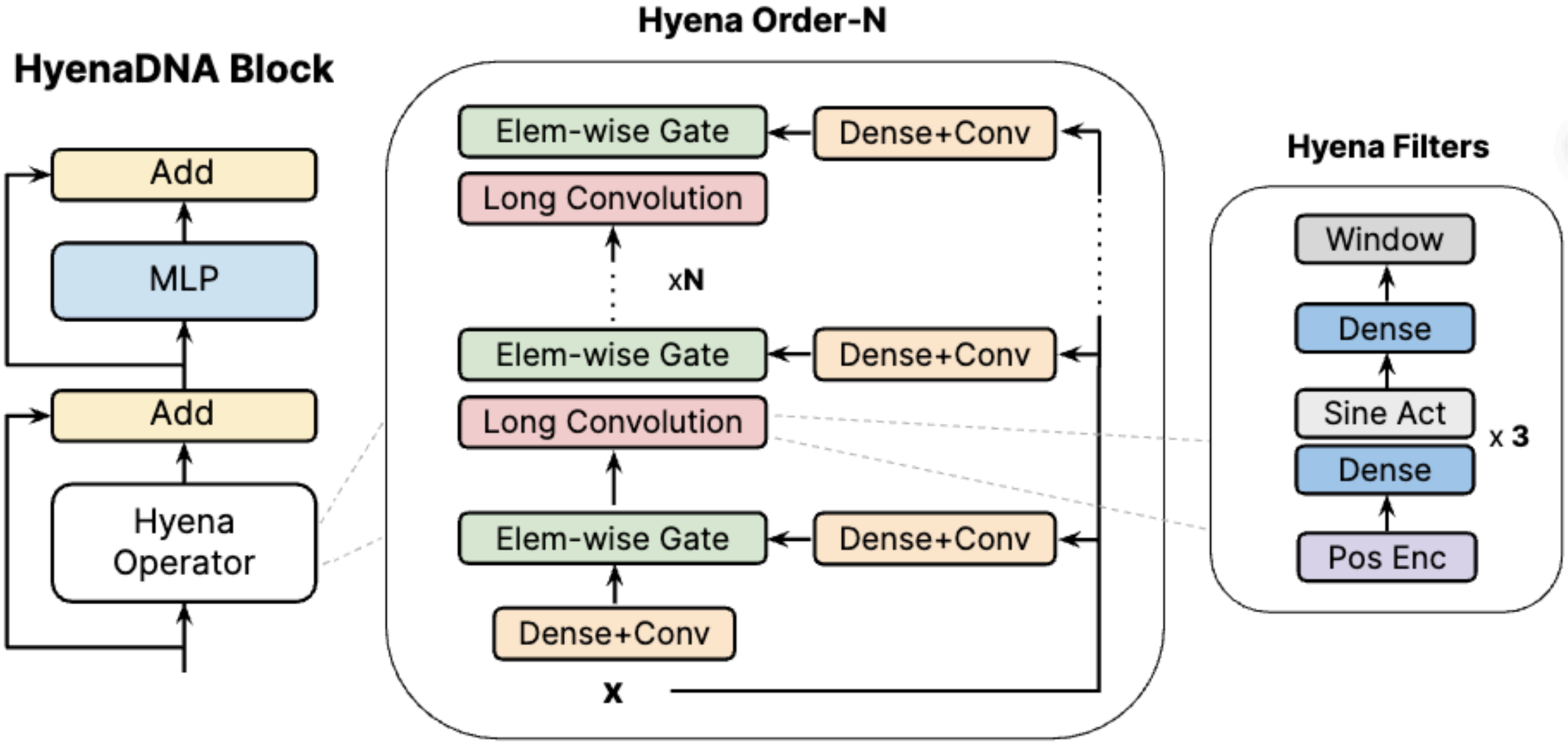
Quality (expressivity)	High ↑	Low ↓
Time Complexity	O(N^2) ↑	O(N*k) ↓

Hyena


Transformer Block
Add
MLP
Add
Attention
xN


HyenaDNA Block
Add
MLP
Add
Hyena
Operator

HyenaDNA


HyenaDNA Block
Add
MLP
Add
Hyena Operator
Hyena Order-N
Elem-wise Gate
Dense+Conv
Long Convolution
xN
Elem-wise Gate
Dense+Conv
Long Convolution
Elem-wise Gate
Dense+Conv
Dense+Conv
x
Hyena Filters
Window
Dense
Sine Act
x 3
Dense
Pos Enc

Hyena Recurrences

(1) Compute a set of *N+1* linear projections of the input.

- Similar to self-attention except that here the projections are not restricted to 3.
- The *n* projections are called x^n , where n runs from *1* to *N*
- The *+1* projection serves a similar role to the *value* vector in self-attention, and is denoted as *v*

(2) Perform a set of interleaved implicit long convolutions and element wise multiplication with one projection at a time.

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

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

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

Hyena

Discrete convolution: input *u* signal of length *L* and a learnable filter *h*

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

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

Toeplitz matrix *S*

Hyena Matrix

Using a matrix form surrogate for attention

- Utilize H3 mechanism
- convolution between the filter *h* and the input *u* can be computed as a matrix multiplication between the Toeplitz matrix form of the filter *h,* S_h, and the input *u.*

$$x^n \cdot (h^n * z^n) \equiv D_x^n S_h z^n$$

Applying the diag operator on vector x^n to obtain D^n_x

$$x^N \cdot \left(h^N * \left(\cdots * x^2 \cdot \left(h^2 * x^1 \cdot \left(h^1 * v\right)\right)\right)\right) \equiv \boxed{\mathrm{D}_x^N \ \mathrm{S}_h^N \cdots \mathrm{D}_x^2 \ \mathrm{S}_h^2 \mathrm{D}_x^1 \ \mathrm{S}_h^1 v}$$

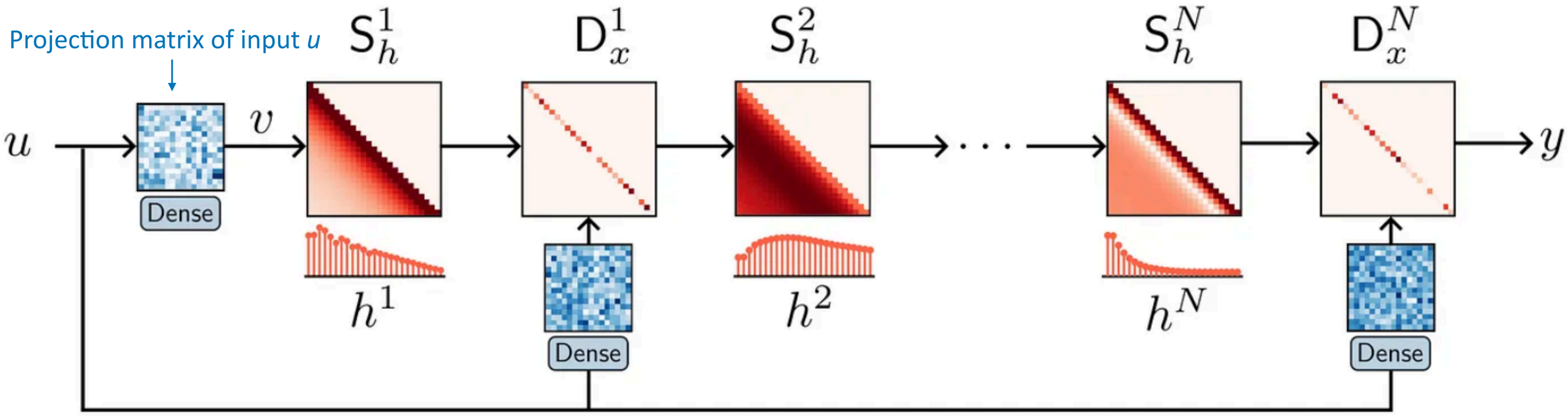
Can be further compressed as $H(u)v$

Hyena Recurrence

S_h^n represents the Toeplitz form of a filter h^n

represents the diagonal form of a projection x^n

Projection matrix of input u



Algorithm 1 Projection

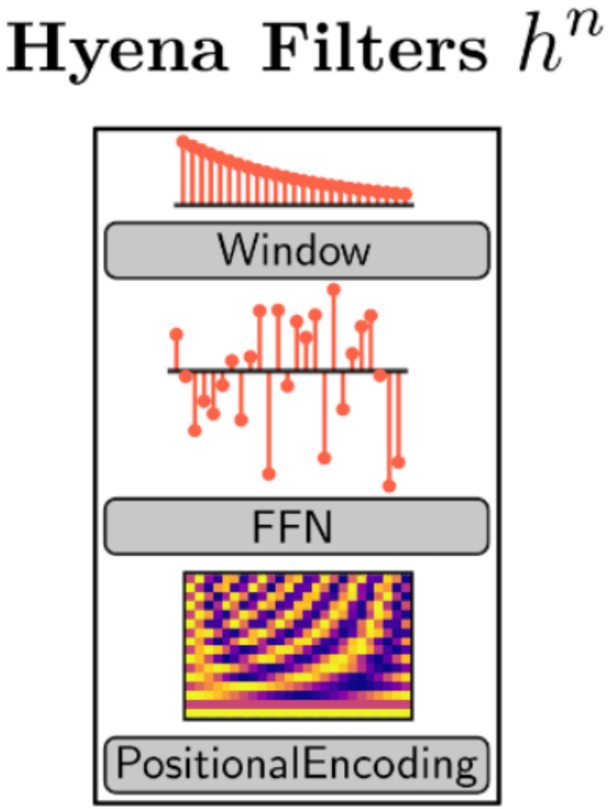
Require: Input sequence $u \in \mathbb{R}^{L \times D}$

1. In parallel across L: $\hat{z} = \mathsf{Linear}(u)$, $\mathsf{Linear} : \mathbb{R}^D \to \mathbb{R}^{(N+1)D}$
2. In parallel across D: $z = \mathsf{DepthwiseConv1d}(h, \hat{z})$, h is a short convolution filter
3. Reshape and split z into $x^1, \ldots, x^N, v$. Dimensions of one element are $x^n \in \mathbb{R}^{D \times L}$

Return $x^1, \ldots, x^N, v$, x^n

Hyena filter

- running a positional encoding vector, t, into an FFN (shallow feed forward network)
- modulating it by a windowing function, W



Hyena filter

Parametrization of convolutions

- Here they did it *implicitly*. (because the explicit way (FIR) can be computationally prohibitive)
- Can think of it as a way of kernel compression

$$\gamma_\theta : t \mapsto h_t = \gamma_\theta(t)$$

T: index of an entry in the filter. (Time step)

Fast Method for Convolutions: cooley-Tukey fast Fourier transform (FFT) algorithm

- Convolution theorem: Space convolution = frequency multiplication

$$y = (h * u) = \mathrm{iFFT}(\mathrm{FFT}(h) \cdot \mathrm{FFT}(u))$$

Inverse Fourier Transform

this allows you to perform the convolution operation with a complexity of $O(n \log_2(n))$ instead of $O(n^2)$
Because the Fourier Transform operation can be performed as a matrix multiplication with the DST (discrete Fourier Transform) matrix

Hyena

Algorithm 1 Projection

Require: Input sequence $u \in \mathbb{R}^{L \times D}$

1. In parallel across L: $\hat{z} = \mathsf{Linear}(u)$, $\mathsf{Linear} : \mathbb{R}^{D} \to \mathbb{R}^{(N+1)D}$
2. In parallel across D: $z = \mathsf{DepthwiseConv1d}(h, \hat{z})$, h is a short convolution filter
3. Reshape and split z into $x^1, \ldots, x^N, v$. Dimensions of one element are $x^n \in \mathbb{R}^{D \times L}$

Return $x^1, \ldots, x^N, v$, x^n

Algorithm 2 Hyena Filter

Require: Sequence length L, positional embedding dimension D_e

1. $t = \mathsf{PositionalEncoding}(L)$, $t \in \mathbb{R}^{L \times D_e}$
2. In parallel across N, L: $\hat{h} = \mathsf{FFN}(t)$, $\mathsf{FFN} : \mathbb{R}^{D_e} \to \mathbb{R}^{ND}$, $\hat{h} \in \mathbb{R}^{L \times ND}$
3. Reshape to $\hat{h} \in \mathbb{R}^{N \times D \times L}$
4. $h = \hat{h} \cdot \mathsf{Window}(t)$, $h \in \mathbb{R}^{N \times D \times L}$
5. Split h into $h^1, \ldots, h^N$

Return $h^1, \ldots, h^N$

Algorithm 3 Forward pass of Hyena

Require: Input sequence $u \in \mathbb{R}^{L \times D}$, order N, model width D, sequence length L, positional embedding dimension D_e

1. $x^1, \ldots, x^N, v = \mathsf{Projection}(u)$
2. $h^1, \ldots, h^N = \mathsf{HyenaFilter}(L, D_e)$

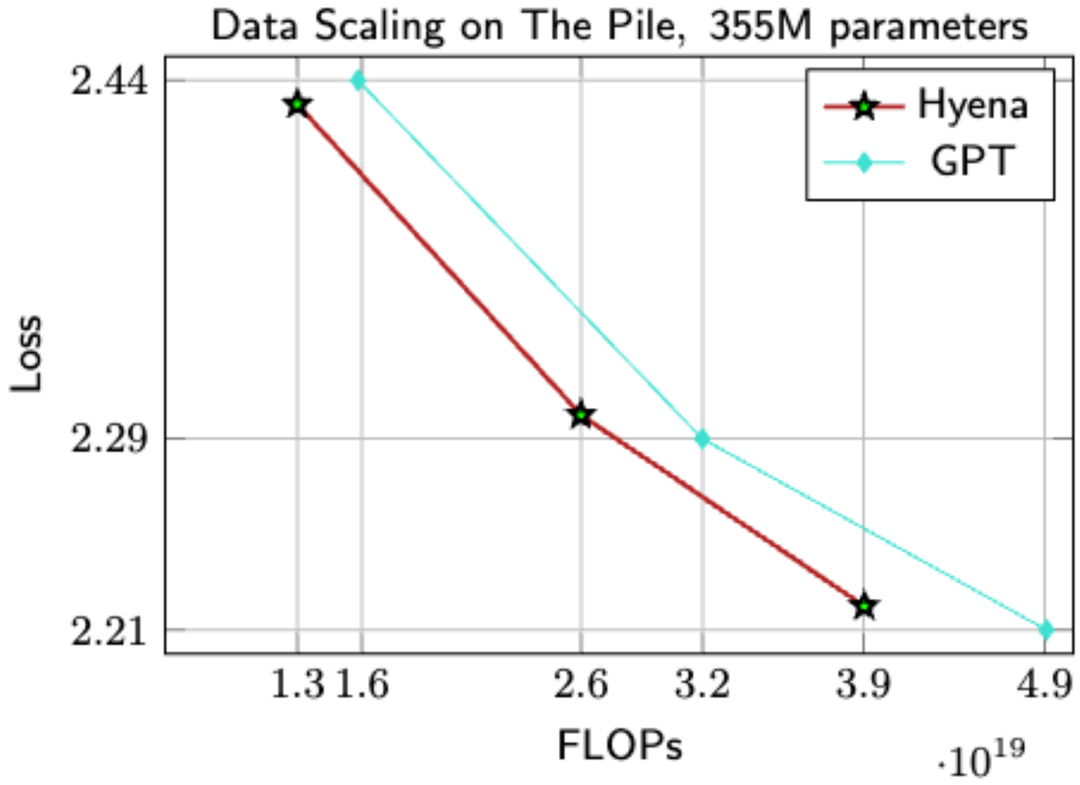
for $n = 1, \ldots, N$ **do**

3. In parallel across D: $v_t \leftarrow x_t^n \cdot \mathsf{FFTConv}(h^n, v)_t$

end for

Return $y = v$

Result



reaching Transformer quality with a 20% reduction in training compute required at sequence length 2K.

4 Architectures tested in Evo

Transformer ++

- Decoder-only with rotary position embeddings
- pre-norm with root mean square layer normalization
- SwiGLU as channel mixer.

Mamba

State-space model

Hyena

- Has the same architecture applied to Transformer ++, but replace all multi-headed self-attention layers with hyena layers

StripedHyena

Attention + gated convolutions in Hyena

Mamba

- Also based on Structured State Space sequence (S4) model.

- Mamba replaces the complex attention and MLP blocks of Transformers with a single, unified SSM block. This aims to reduce computational complexity and improve inference speed

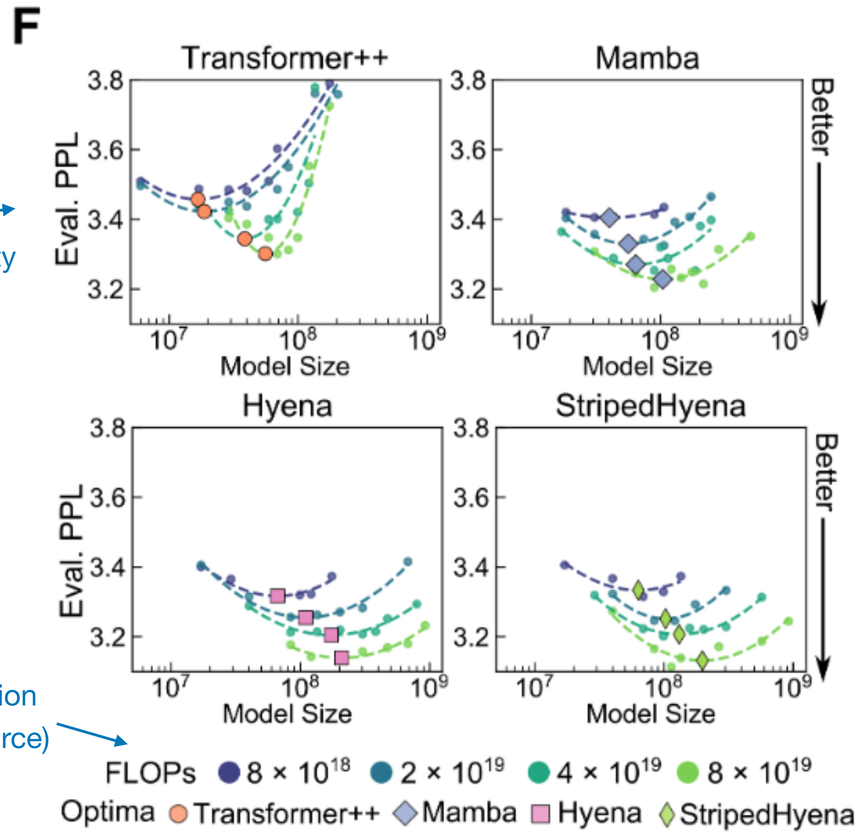
- By integrating **a selection mechanism** into its state space models, Mamba can effectively **decide whether to propagate or discard** information based on the relevance of each token in the sequence.

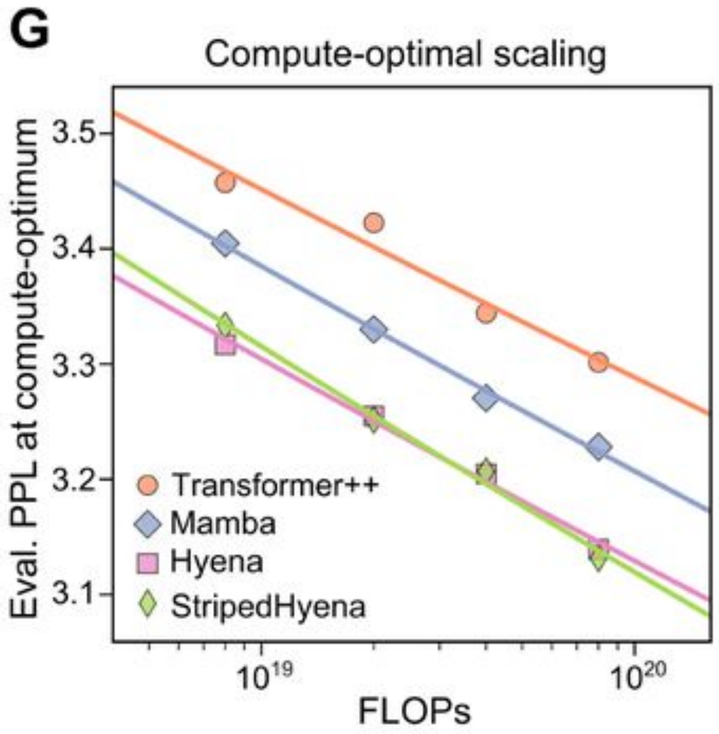
Aspect	Transformers	Mamba
Efficiency & Scalability	Struggles with long sequences	Linear-time efficiency
Selective SSMs	One size fits all	Smart selection, adaptive memory
Performance	Decent in standard scenarios	Outshines similar-sized, rivals larger ones
Throughput	Average speed	5x faster, lightning-fast
Handling Long Sequences	Challenges with length	Effortlessly handles long data

4 Architectures tested in Evo

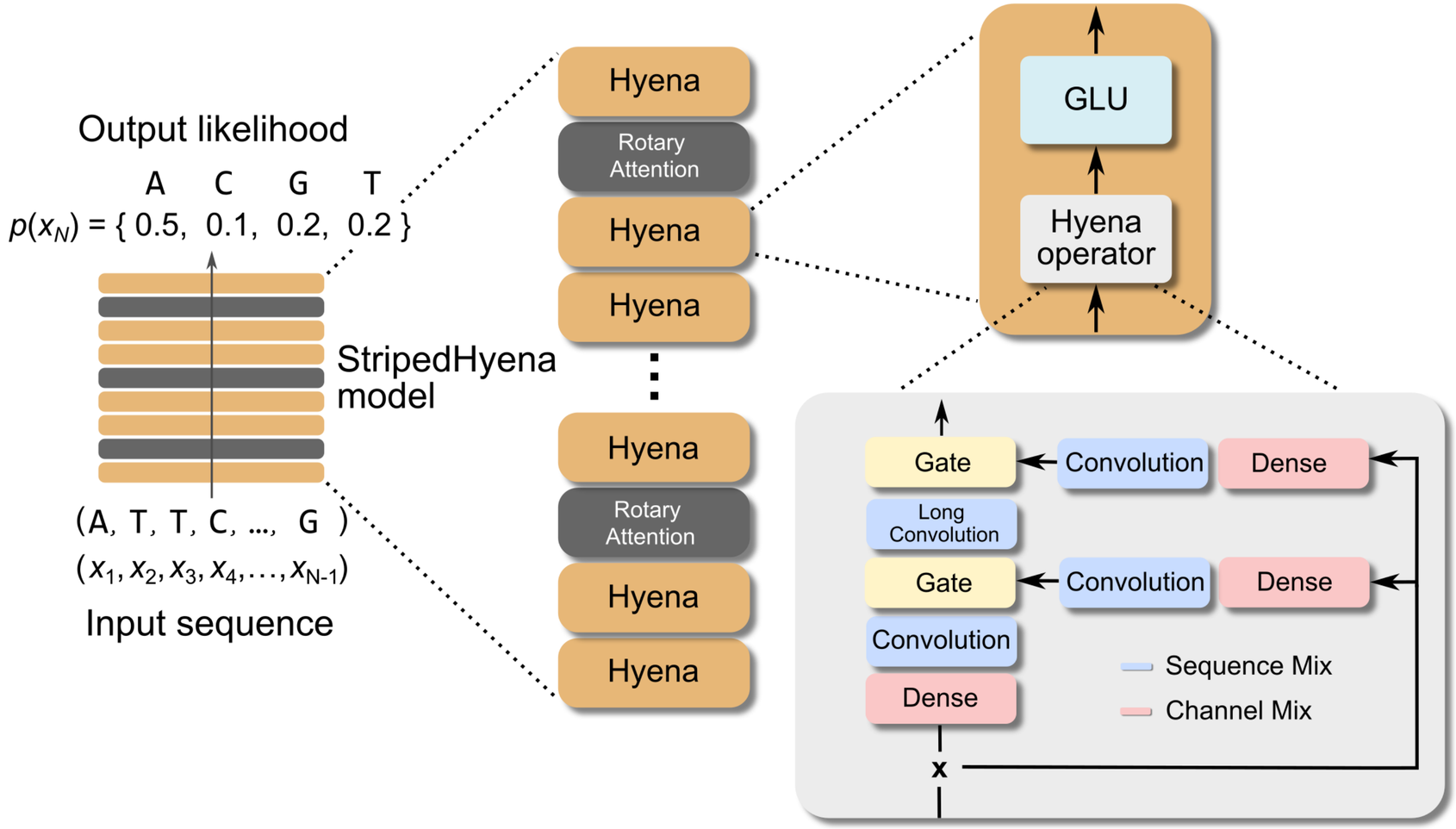
Evolution perplexity

Floating point operation (computational resource)





Evo



29 layers of hyena layers interleaved with 3 layers of multi-head attention equipped with rotary position embedding

Rotary Position Embedding (RoPE)

- Transformer models, by their inherent design, do not consider the order of input tokens.
- To maintain the sequence information and thus the meaning, positional embeddings are integrated into the model.

The pig chased the dog

Ways of Position Embedding:

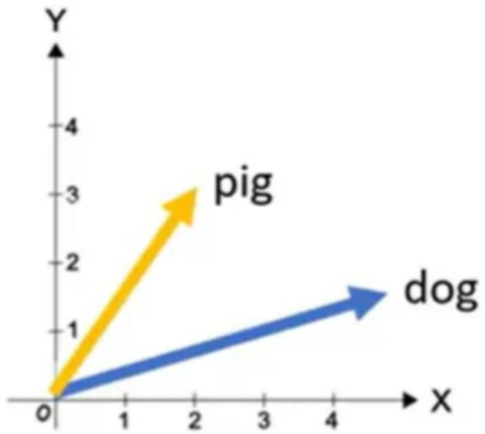
- Absolute Positional Embeddings
- Relative Positional Embeddings

How to combine the strength of both?

Rotary Position Embedding (RoPE)

- Each position in the sequence is represented by a rotation in the embedding space
- Instead of adding a positional vector, it applies a rotation to the word vector
- Have great stability of vectors, and can preserve of relative positions

The pig chased the dog



StripedHyena

Ways to improve scaling:

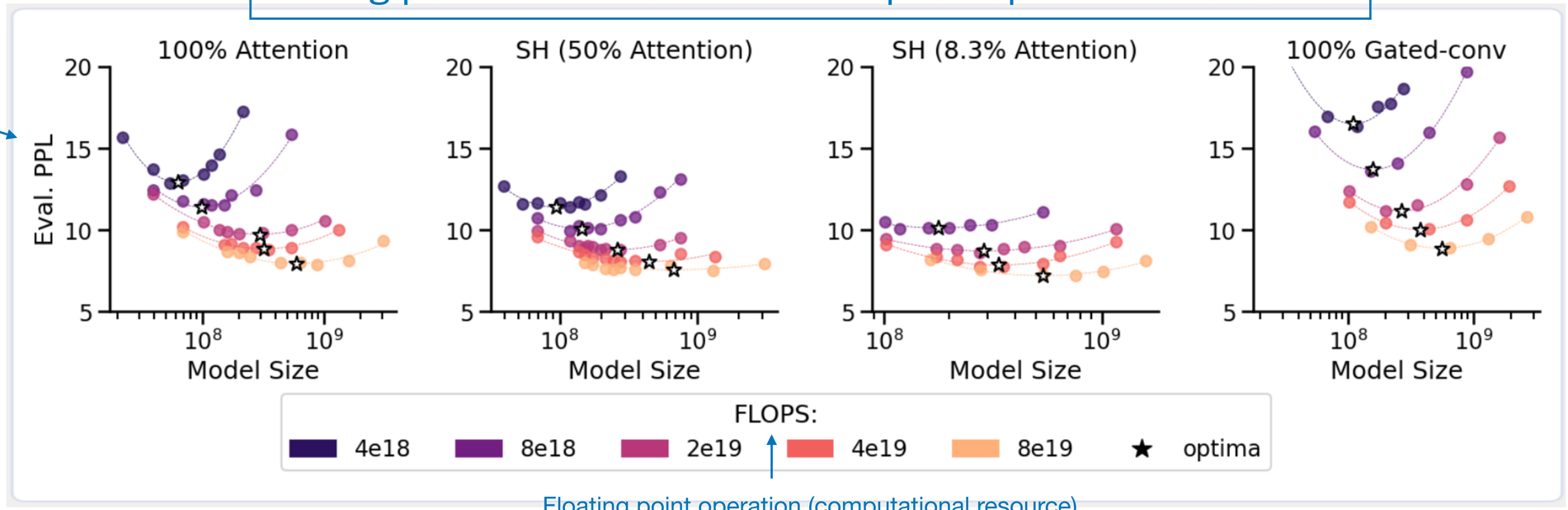
Hybridization

- given a compute budget, architectures built out of mixtures of different key layers always outperform homogenous architectures

Multi-head gated convolution

Hybridizing improves on both scaling and robustness to training performed outside the compute-optimal frontier.

Perplexity



Floating point operation (computational resource)

StripedHyena

- a hybrid architecture composed of multi-head, grouped-query attention and gated convolutions arranged in Hyena blocks
- The hyena layers: implementing the bulk of the computation required for sequence processing
- Attention layers: supplementing the ability to recall information from the context of an input

Evo

Pre-training

Stage 1	Stage 2
8k tokens	131k tokens

Choose a base model from one of the two stages, depending on the downstream task,
Then fine-tune on smaller datasets of interest

- 340B tokens (300 B nucleotides and uses a byte-level, single nucleotide tokenizer)
- No explicit supervision or annotations were used
- 2 x 10^{22} FLOPS

Multi-stage sequence length pretraining has been shown to reduce the overall number of compute hours required to train long context models

Scaling laws analysis

Determine the relationship between training, architectural details, and performance metrics.

Once done, can be used to optimally scale training to larger models and datasets

- First scaling laws analysis for DNA sequence modeling

What does Evo do?

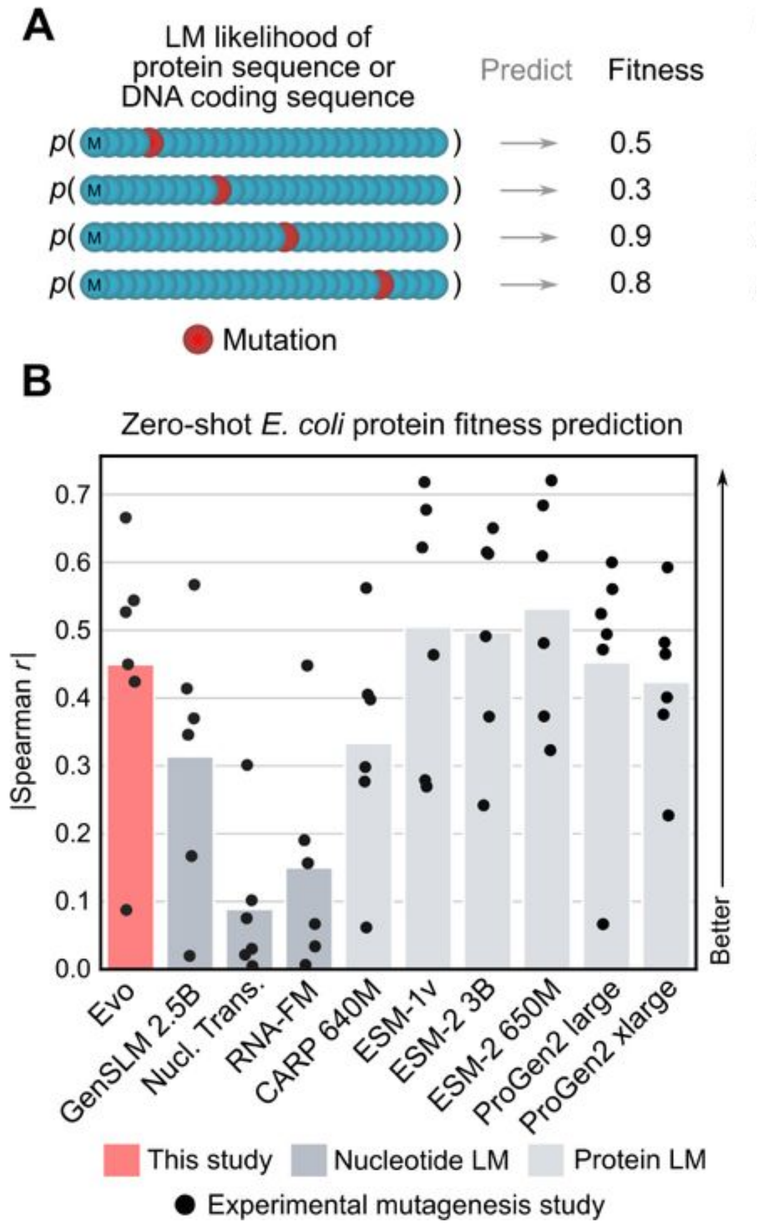
Molecular level:

- Zero-shot function prediction for proteins, non-coding RNAs, and regulatory RNA
- Generative design of CRISPR-Cas molecular complexes and transposable biological systems

Genome-scale

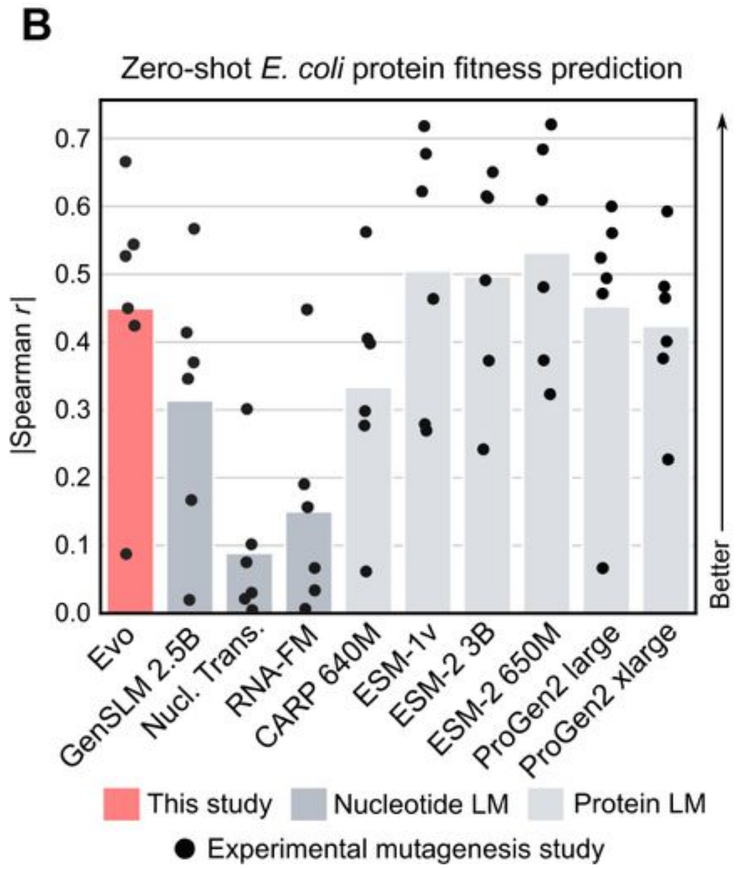
- Gene essentiality prediction across diverse bacterial and phase genomes

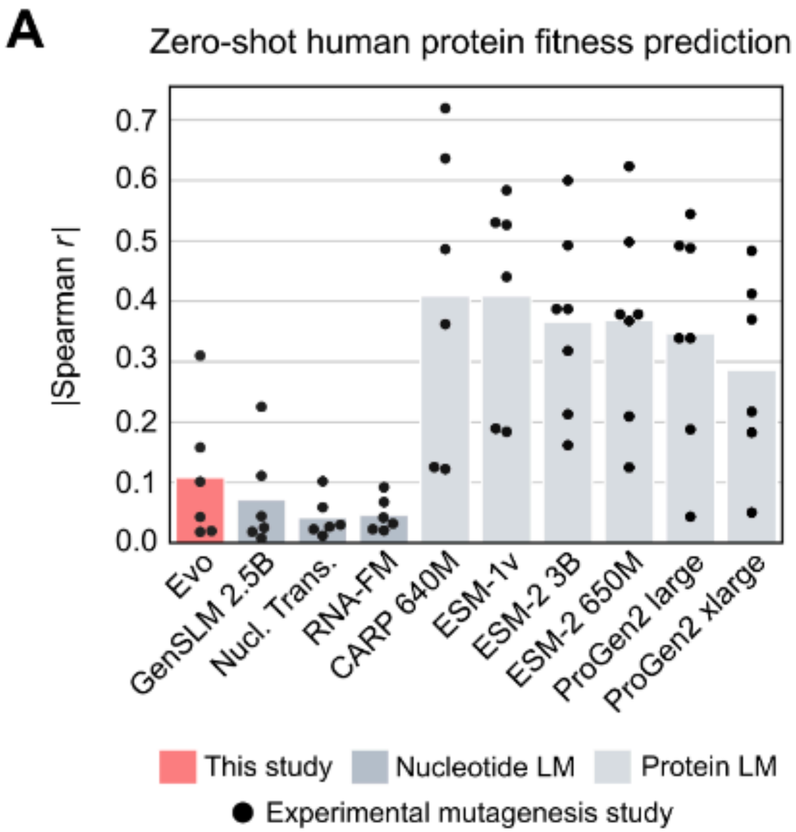
Protein Function prediction

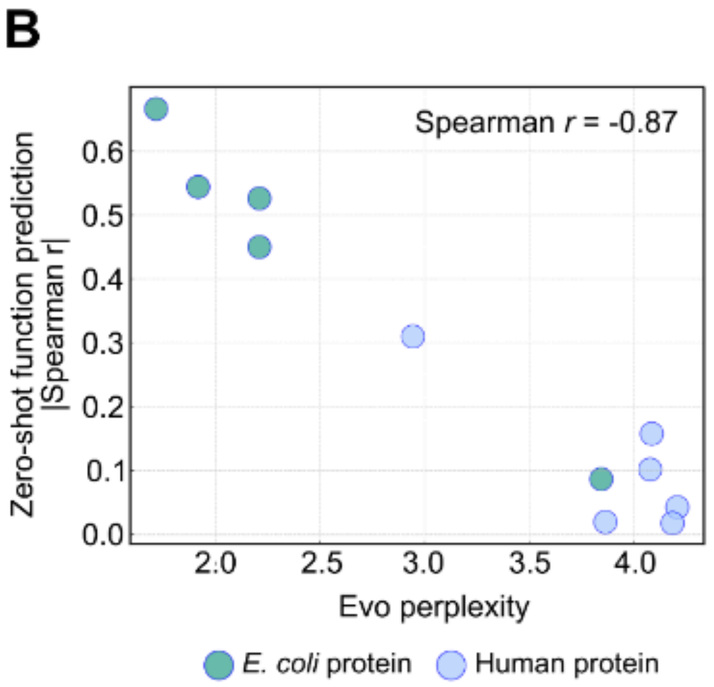


- They performed the analysis to both *E. coli* and human protein
- The compilation of *E. coli* and human both contains 6 studies.
- Evo here is pertained with 8k context
- If the predicted values are different for the nucleotide sequence and the protein sequence, use the one from nucleotide sequence.
- Spearman correlation here is calculated between the experimental fitness values and the sequence likelihood' sequence pseudolikelihood.

Protein Function prediction

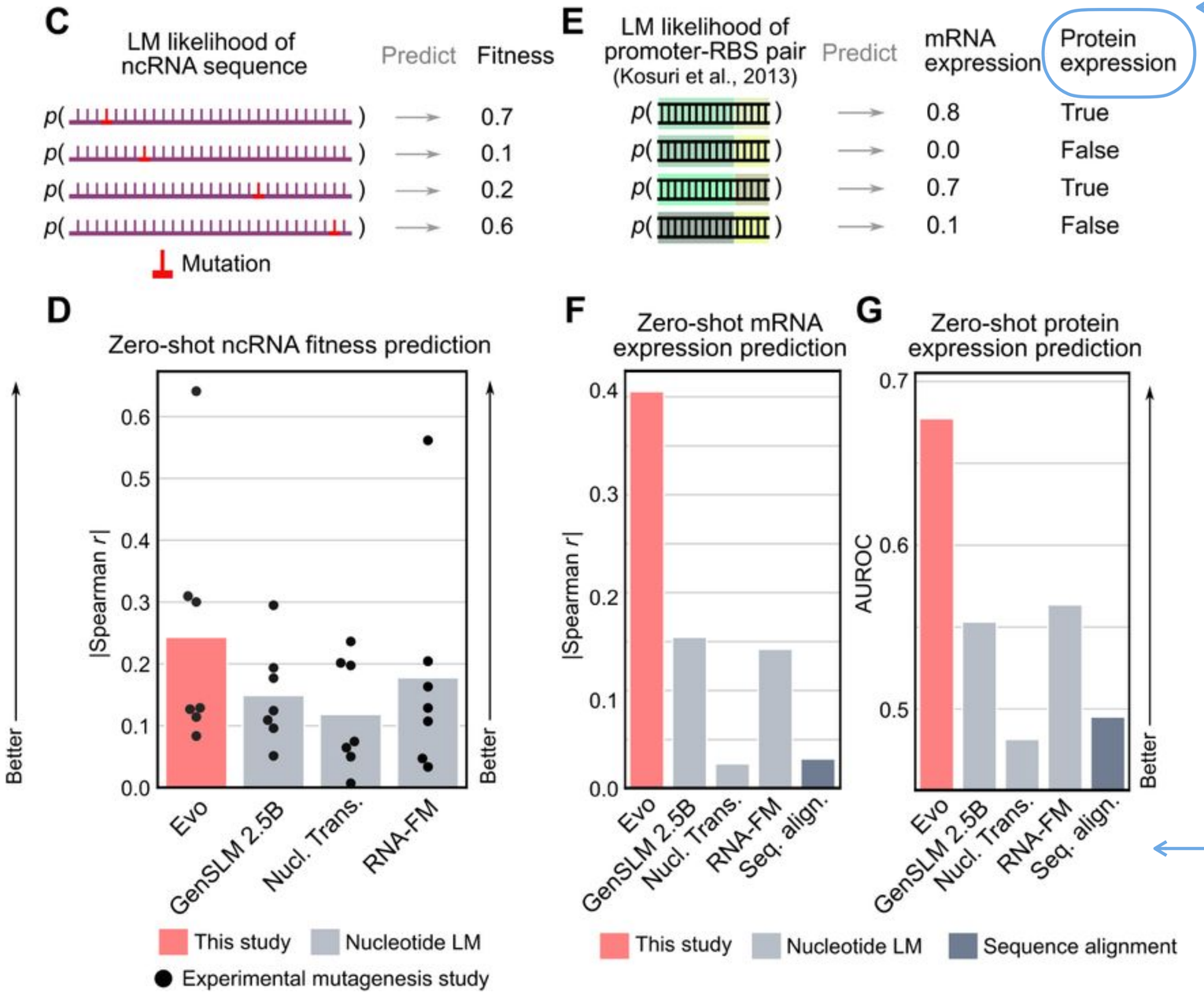






X axis: Evo perplexity of the wild type nucleotide sequence

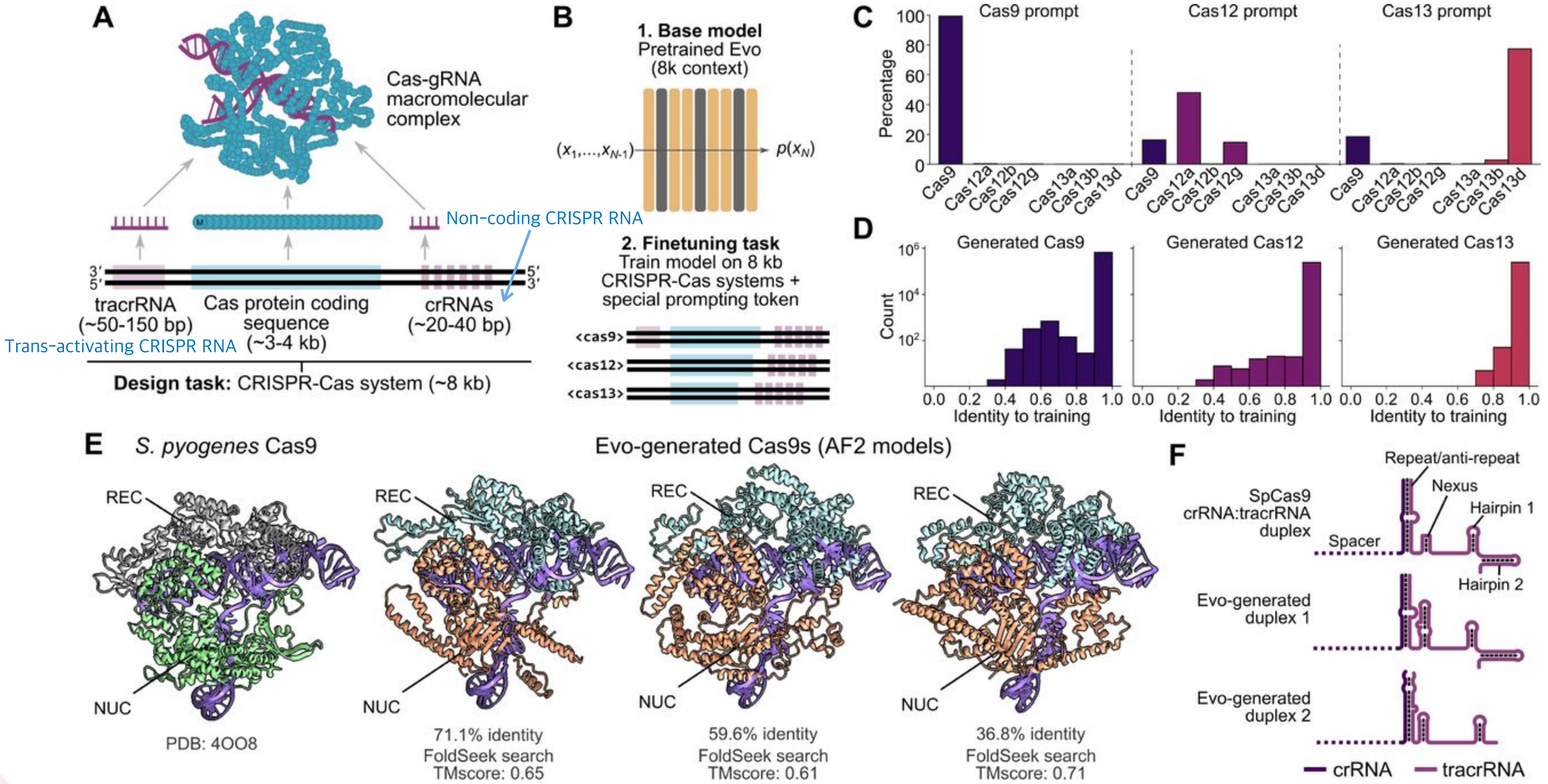
Doesn't perform good in human
because Evo was only trained on procaryotes



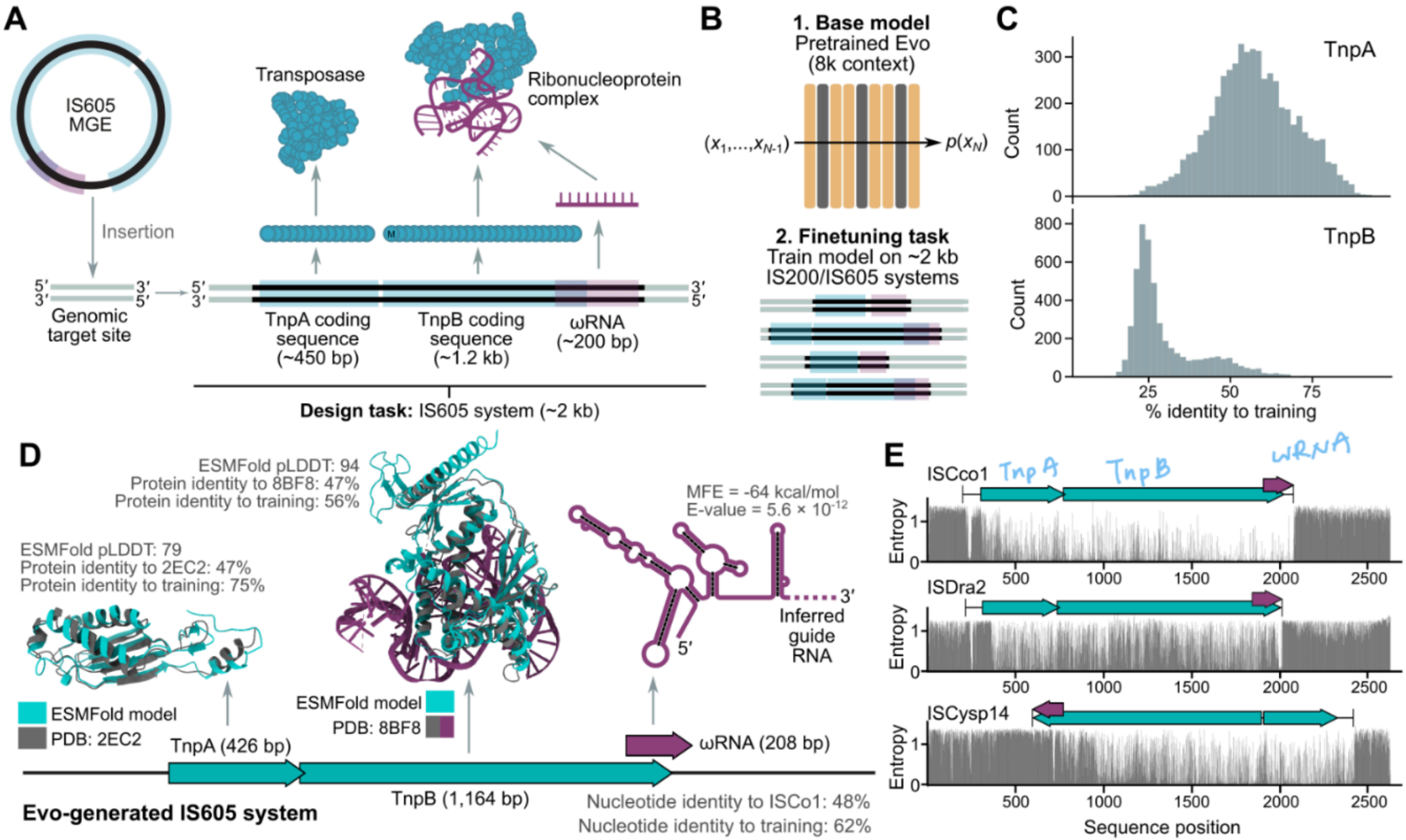
Binarized with a cutoff where the expression values is above or below 100,000

2G: noted that none of the baseline language models have been trained on datasets containing regulatory elements

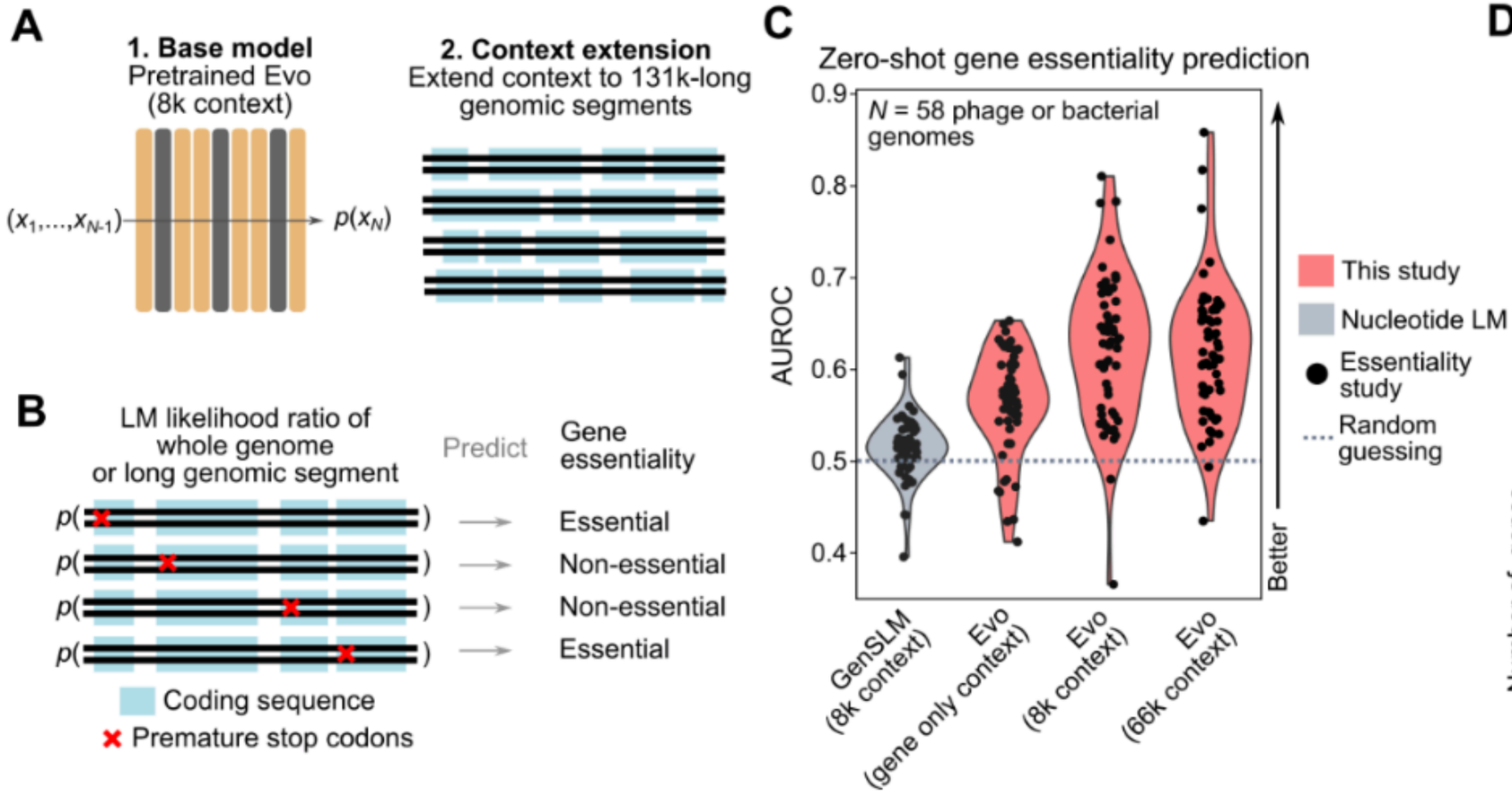
Generative design of protein-RNA complexes



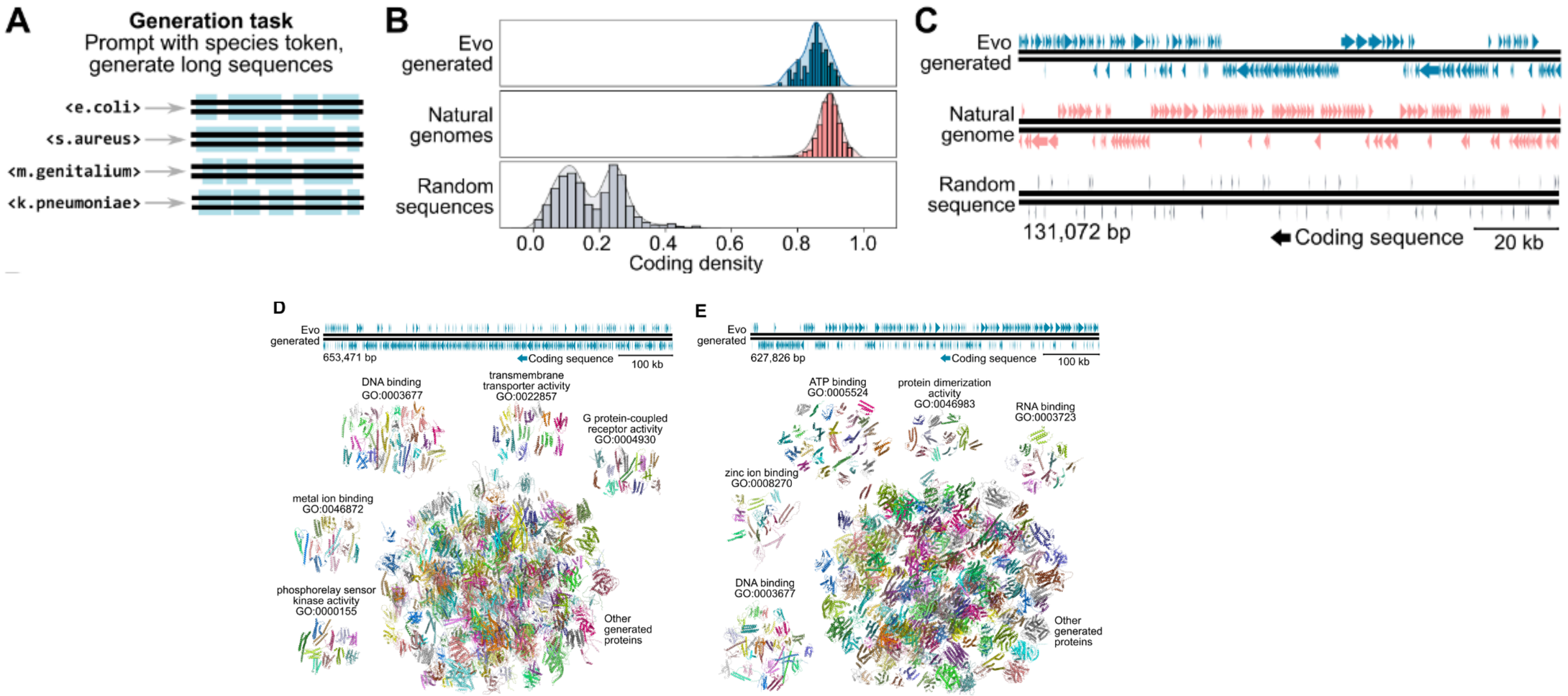
Generative design of transposable biological system



Predict gene essentiality



Predict gene essentiality



Challenges

- The ability oto predict functional effects of mutations on human protein is limited
- Maintaining coherent and diverse generation over long sequences is tough
  - hundreds of kilobases that demonstrate a high-level understanding of genome organization, but struggles to include key marker genes such as full tRNA-encoding repertoire

Reference

https://medium.com/@ghadi.alhajj/the-hyena-operator-say-goodbye-to-self-attention-f23e7e578c15
https://www.youtube.com/watch?v=haSkAC1fPX0&t=503s&ab_channel=OpenBioML
https://medium.com/ai-insights-cobet/rotary-positional-embeddings-a-detailed-look-and-comprehensive-understanding-4ff66a874d83
https://medium.com/@jelkhoury880/what-is-mamba-845987734ffc

Thanks for listening