A New Generation of Foundation Models Based on xLSTM

Günter Klambauer

Slides available here!

bsky.app/profile/gklambauer.bsky.social

@gklambauer

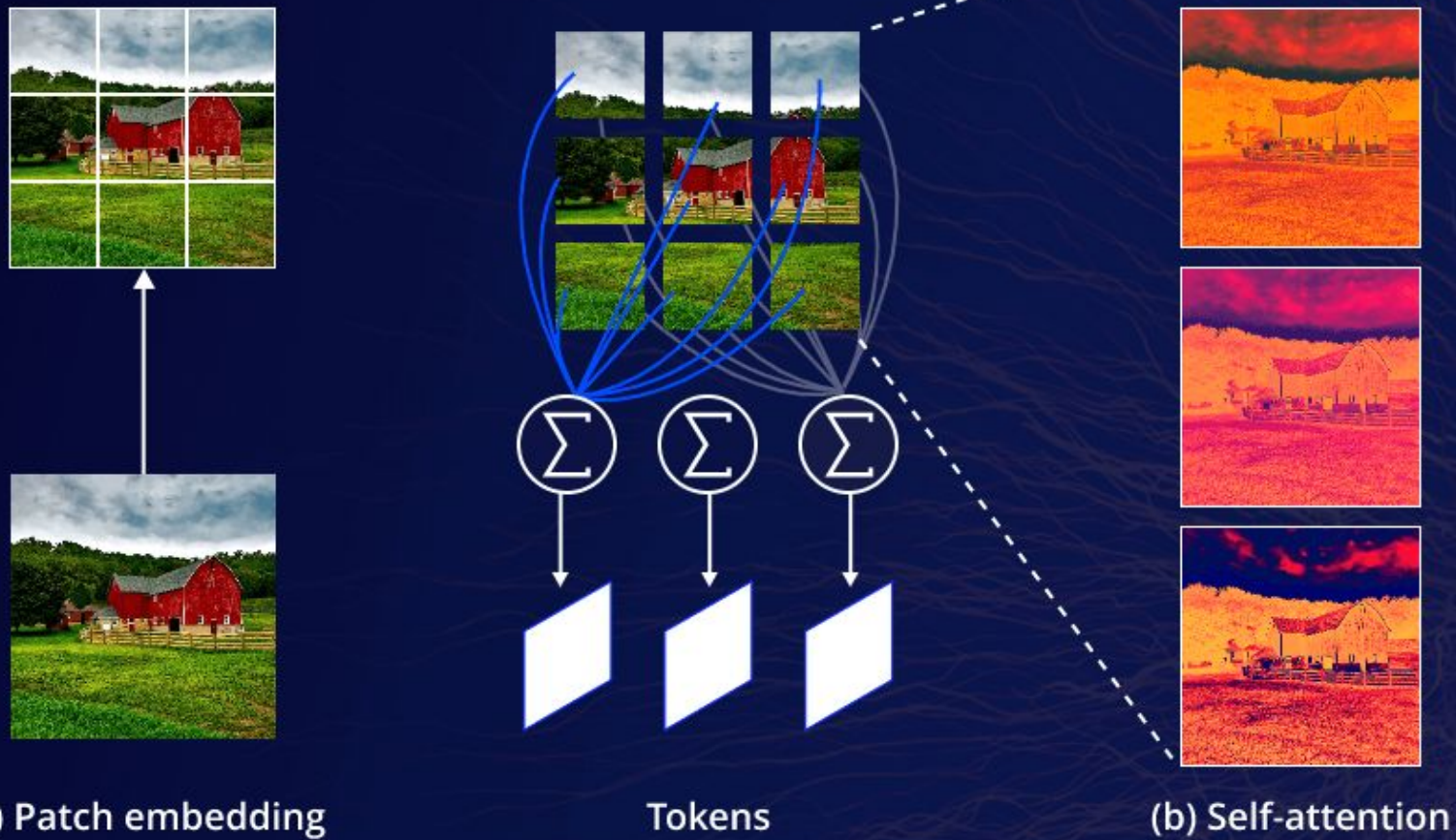
JOHANNES KEPLER UNIVERSITY LINZ

- A word on foundation models
- RNNs & LSTM
- The age of Transformers
- The age of recurrent LLMs
- Applications
  - Vision-xLSTM (ViL)
  - Bio-xLSTM
    - DNA-xLSTM
    - Prot-xLSTM
    - Chem-xLSTM
- Conclusion

A word on foundation models

CHAT GPT

Vision Transformer



https://www.leewayhertz.com/vision-transformer-model/

Models considered as foundation models

- **Language Models**:
  - e.g. BERT, GPT family, LLama family,…
- **Vision Models**:
  - e.g. Vision Transformers (ViTs), DINO, SimCLR,...
- **Multimodal Models**:
  - CLIP, Gemini,...

[1] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Properties of foundation models?

[..] any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks. [1]

- Pre-trained on Massive Data
  - EU AI Act: FM if more than 10^{25} FLOPS used for training
- Self-Supervised Learning
- Scalability
- Multimodal Capabilities
- Adaptability

[1] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Properties of foundation models?

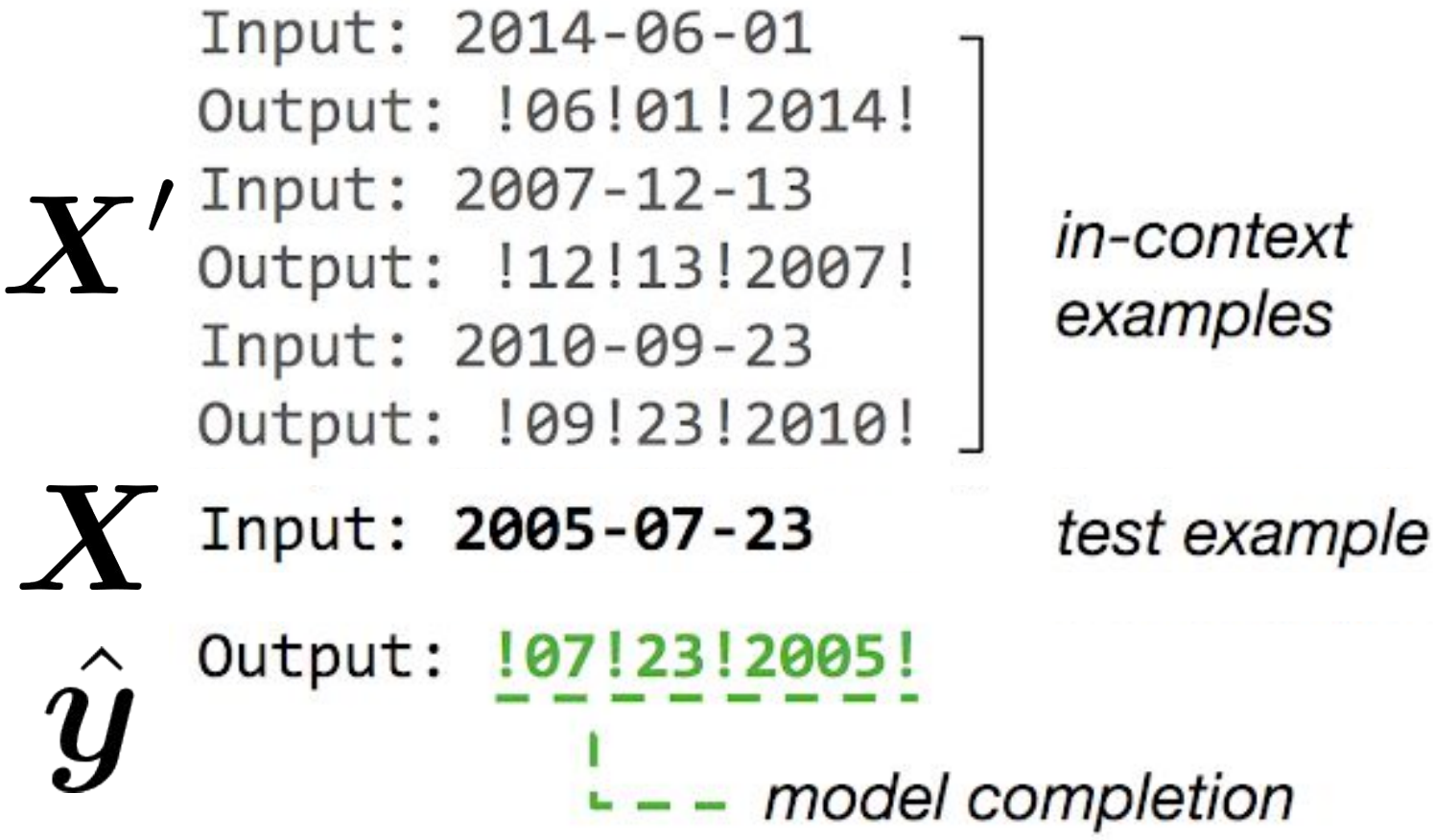
- Pre-trained on Massive Data
- Self-Supervised Learning
- Scalability
- Multimodal Capabilities
- **Adaptability**
  - Good performance when fine-tuned to a variety of new tasks
  - **In-context learning**

In-context learning

$$\hat{y} = g([\boldsymbol{X}', \boldsymbol{X}]; \boldsymbol{w})$$

- New task described via context $\boldsymbol{X}'$ (additional input)
- Model's $g(.; \boldsymbol{w})$ ability to adapt to and solve new task $(\boldsymbol{X}, \boldsymbol{y})$ based on additional input $\boldsymbol{X}'$
- No weights $\boldsymbol{w}$ in the neural network changed

In-context learning



RNNs & LSTM

Recurrent neural networks

RNNs map the **input of one time step** and the **last hidden state** to a **new hidden state**

$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}, \boldsymbol{h}_{t-1}; \boldsymbol{w})$$

RNNs are neural nets with **adaptive weight matrices (optimized/trained to solve a task):**

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{R}\boldsymbol{h}_{t-1})$$

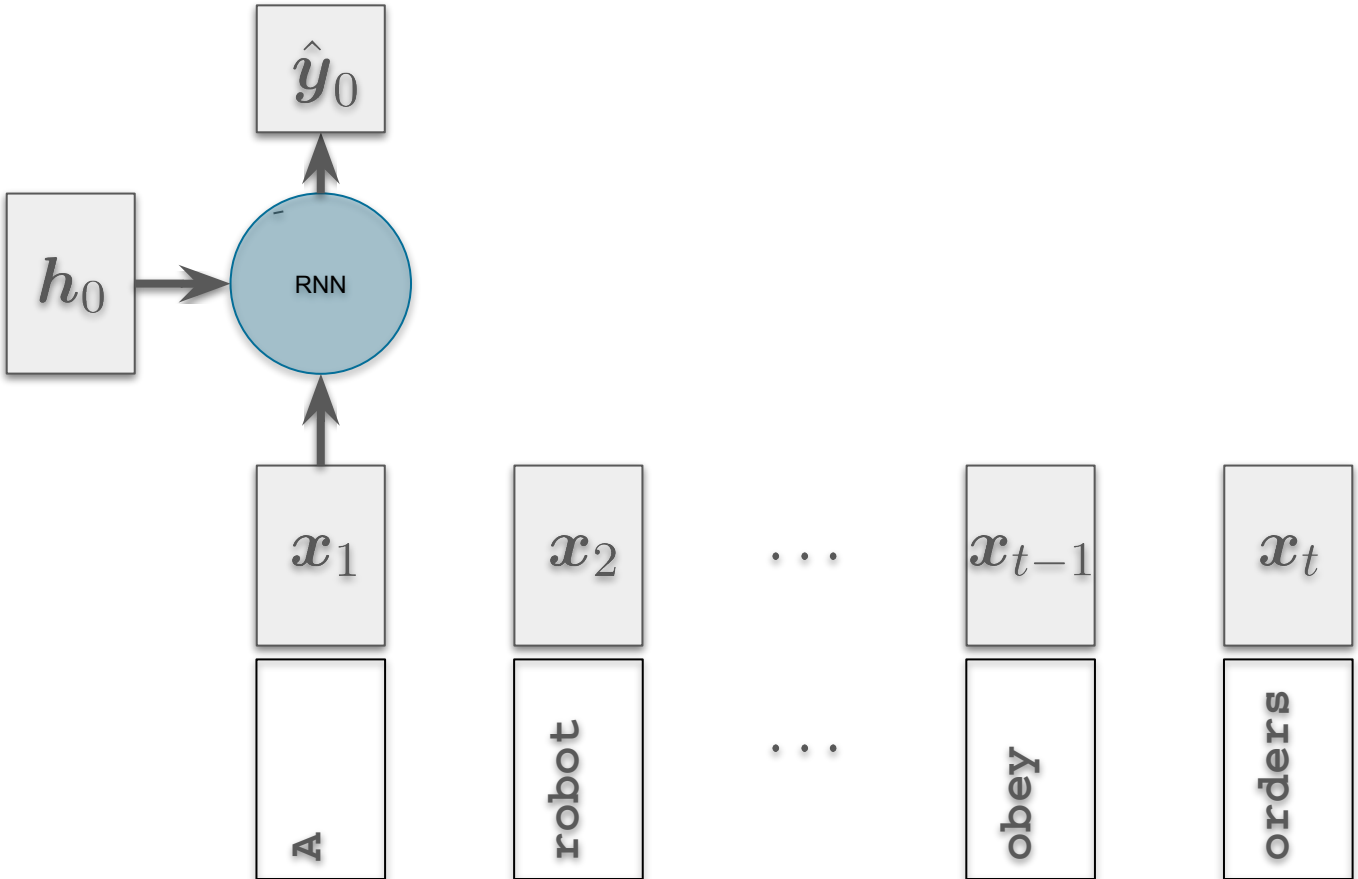
$$\hat{\boldsymbol{y}}_t = \phi(\boldsymbol{V}\boldsymbol{h}_t)$$

The same transformation is applied in each time step. **Output:** probability of next word.

$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{w})$$

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{R}\boldsymbol{h}_{t-1})$$

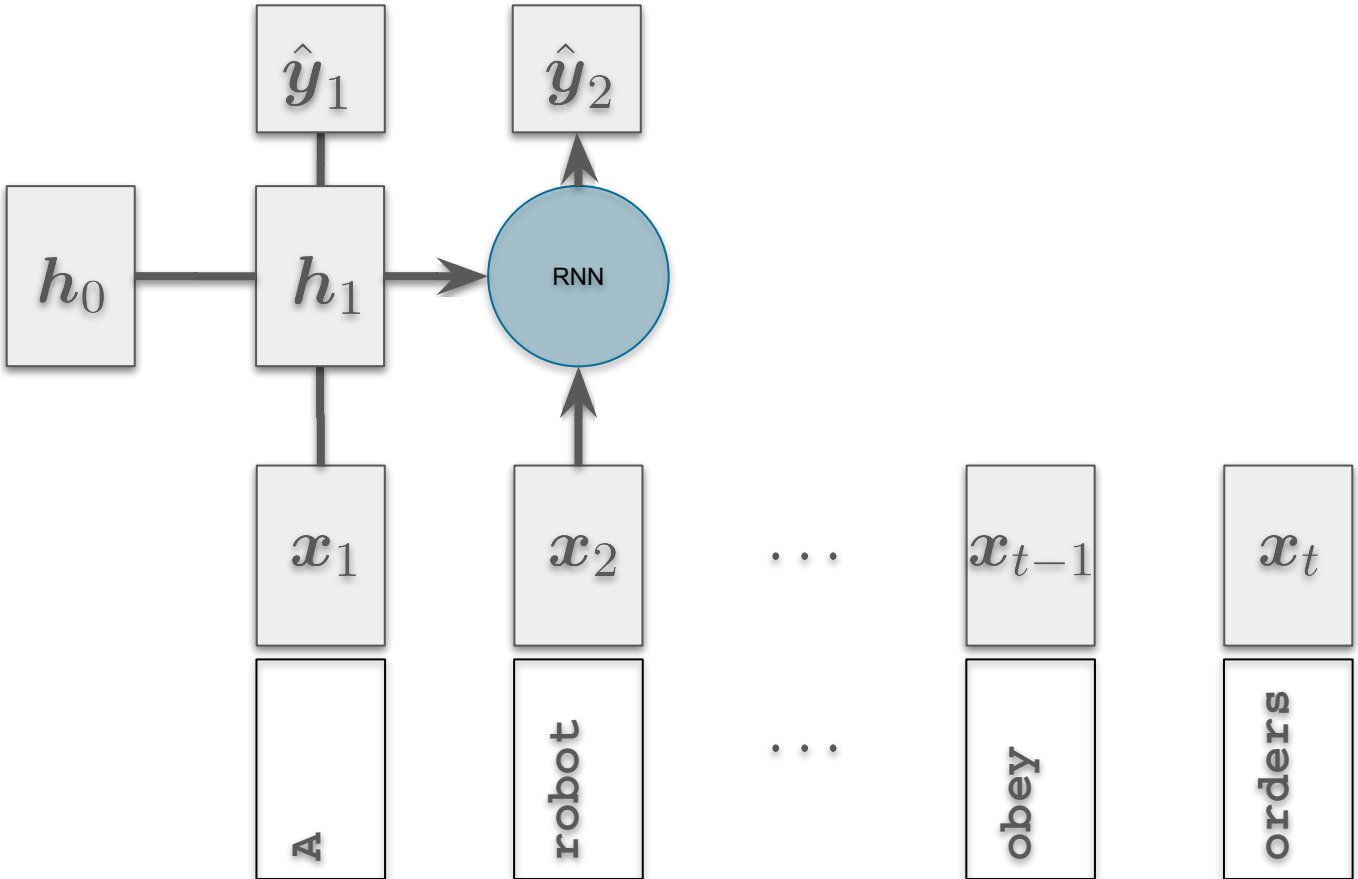
$$\hat{\boldsymbol{y}}_t = \phi(\boldsymbol{V}\boldsymbol{h}_t)$$



$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{w})$$

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{R}\boldsymbol{h}_{t-1})$$

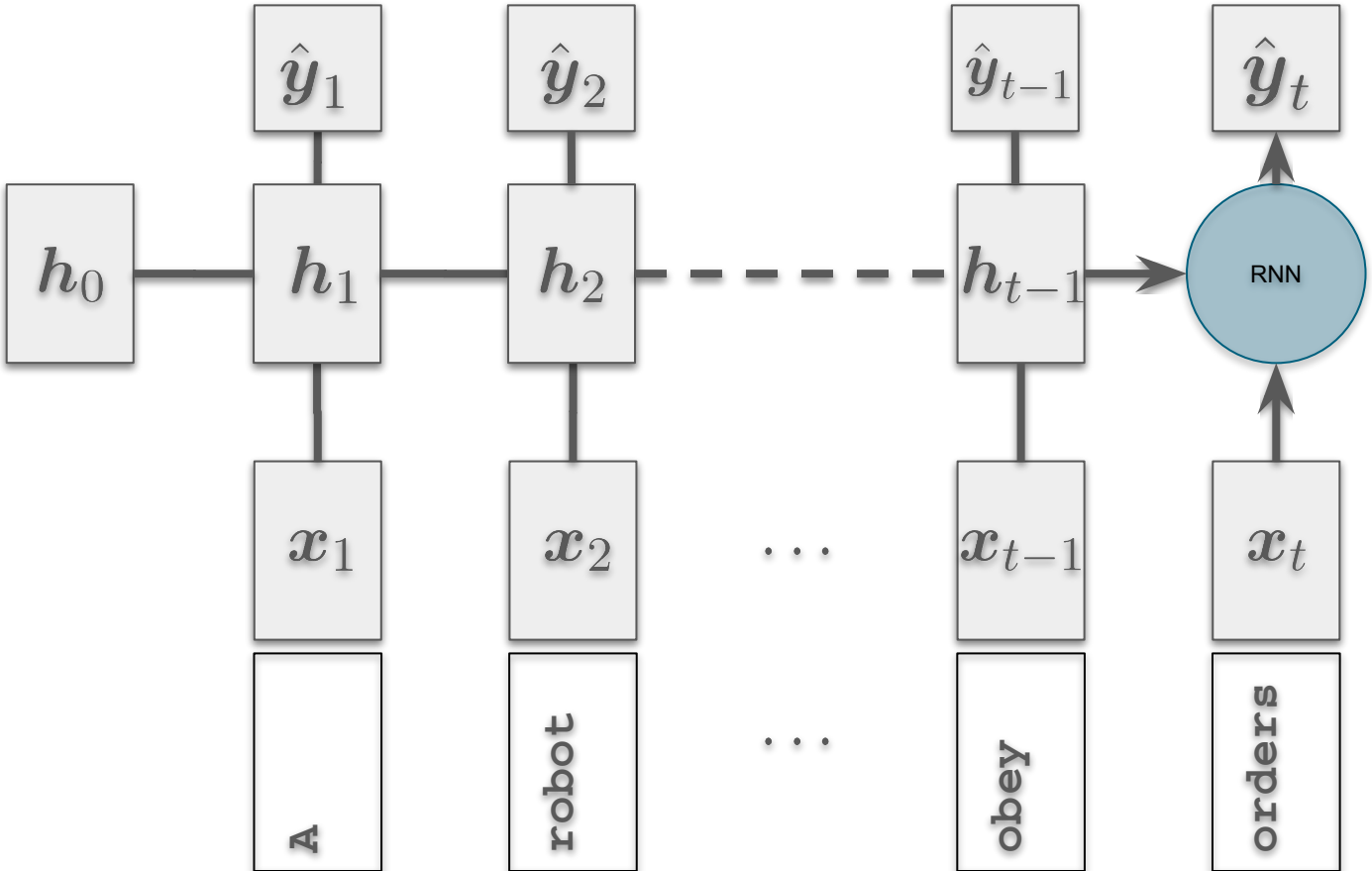
$$\hat{\boldsymbol{y}}_t = \phi(\boldsymbol{V}\boldsymbol{h}_t)$$

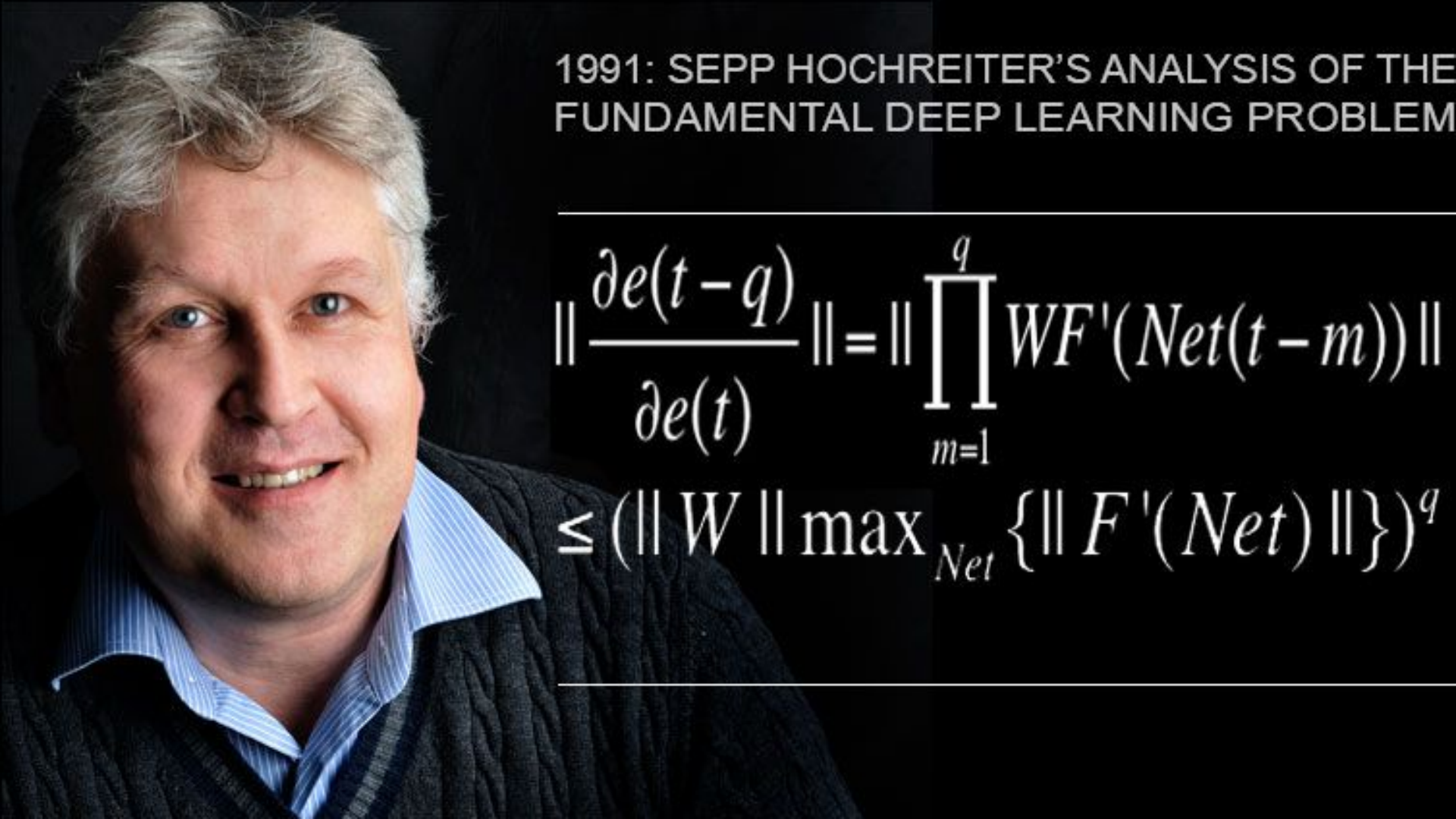


$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{w})$$

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{R}\boldsymbol{h}_{t-1})$$

$$\hat{\boldsymbol{y}}_t = \phi(\boldsymbol{V}\boldsymbol{h}_t)$$

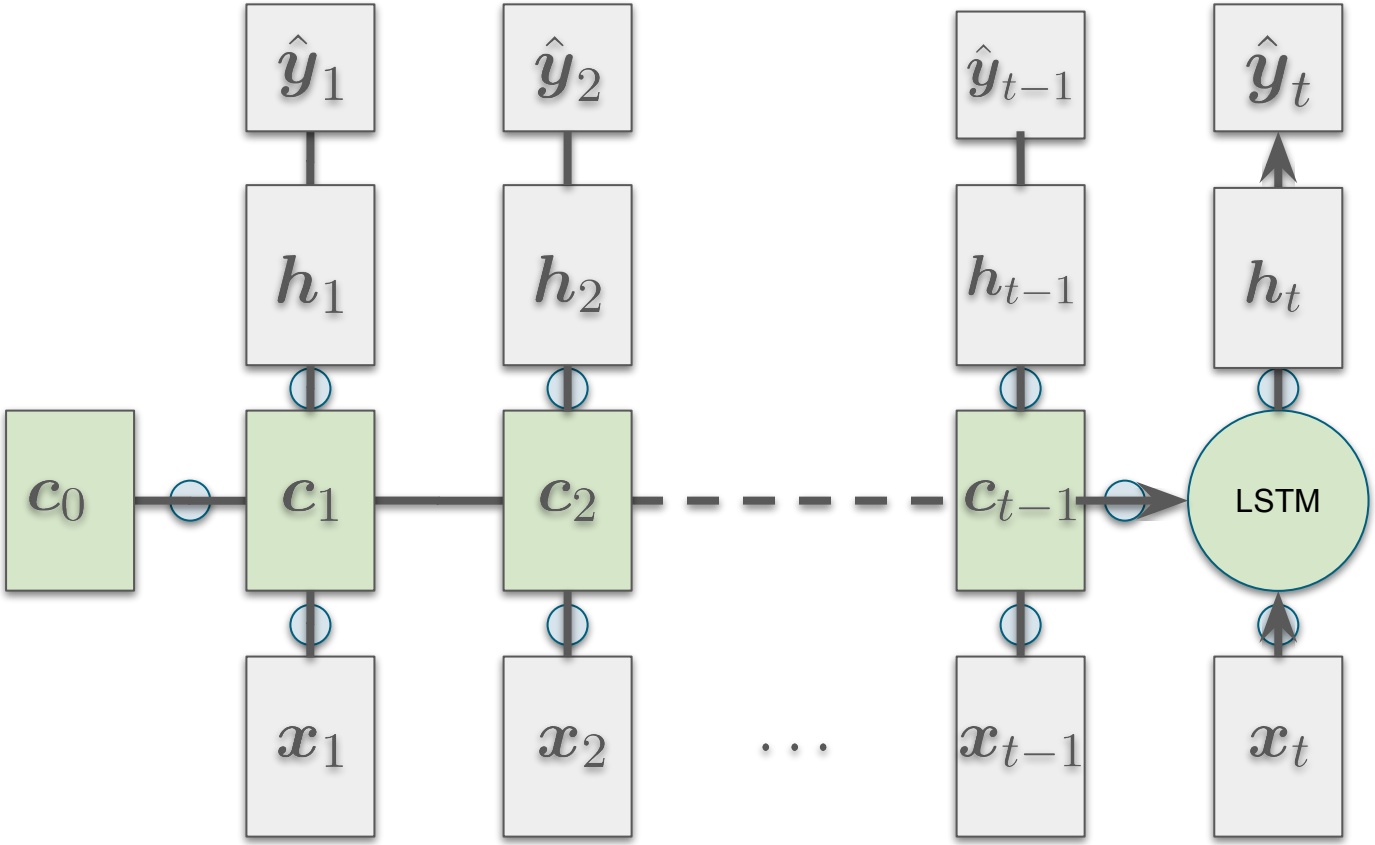



1991: SEPP HOCHREITER'S ANALYSIS OF THE
FUNDAMENTAL DEEP LEARNING PROBLEM
$$\| \frac{\partial e(t-q)}{\partial e(t)} \| = \| \prod_{m=1}^{q} WF'(Net(t-m)) \|$$
$$\leq (\| W \| \max_{Net} \{ \| F'(Net) \| \})^q$$

$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{w})$$

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{R}\boldsymbol{h}_{t-1})$$

$$\hat{\boldsymbol{y}}_t = \phi(\boldsymbol{V}\boldsymbol{h}_t)$$



Long term memory through cell states

LSTM cells

$$
\begin{aligned}
c_t &= \mathrm{f}_t \; c_{t-1} + \mathrm{i}_t \; z_t && && \text{cell state} \\
h_t &= \mathrm{o}_t \; \tilde{h}_t \,, && \tilde{h}_t &= \psi\left(c_t\right) && \text{hidden state} \\
z_t &= \varphi\left(\tilde{z}_t\right) \,, && \tilde{z}_t &= \boldsymbol{w}_z^\top \boldsymbol{x}_t + r_z h_{t-1} + b_z && \text{cell input} \\
\mathrm{i}_t &= \sigma\left(\tilde{\mathrm{i}}_t\right) \,, && \tilde{\mathrm{i}}_t &= \boldsymbol{w}_\mathrm{i}^\top \boldsymbol{x}_t + r_\mathrm{i} \, h_{t-1} + b_\mathrm{i} && \text{input gate} \\
\mathrm{f}_t &= \sigma\left(\tilde{\mathrm{f}}_t\right) \,, && \tilde{\mathrm{f}}_t &= \boldsymbol{w}_\mathrm{f}^\top \boldsymbol{x}_t + r_\mathrm{f} \, h_{t-1} + b_\mathrm{f} && \text{forget gate} \\
\mathrm{o}_t &= \sigma\left(\tilde{\mathrm{o}}_t\right) \,, && \tilde{\mathrm{o}}_t &= \boldsymbol{w}_\mathrm{o}^\top \boldsymbol{x}_t + r_\mathrm{o} \, h_{t-1} + b_\mathrm{o} && \text{output gate}
\end{aligned}
$$

Note: scalar notation for cell state!

The unreasonable effectiveness of recurrent neural networks

Andrej Karpathy blog

About

The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I trained my first recurrent network for Image Captioning. Within a few dozen minutes of training my first baby model (with rather arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of the results you get out of it blows past your expectations, and this was one of those times. What made this result so shocking at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with more experience I've in fact reached the opposite conclusion). Fast forward about a year: I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

> *We'll train RNNs to generate text character by character and ponder the question "how is that even possible?"*

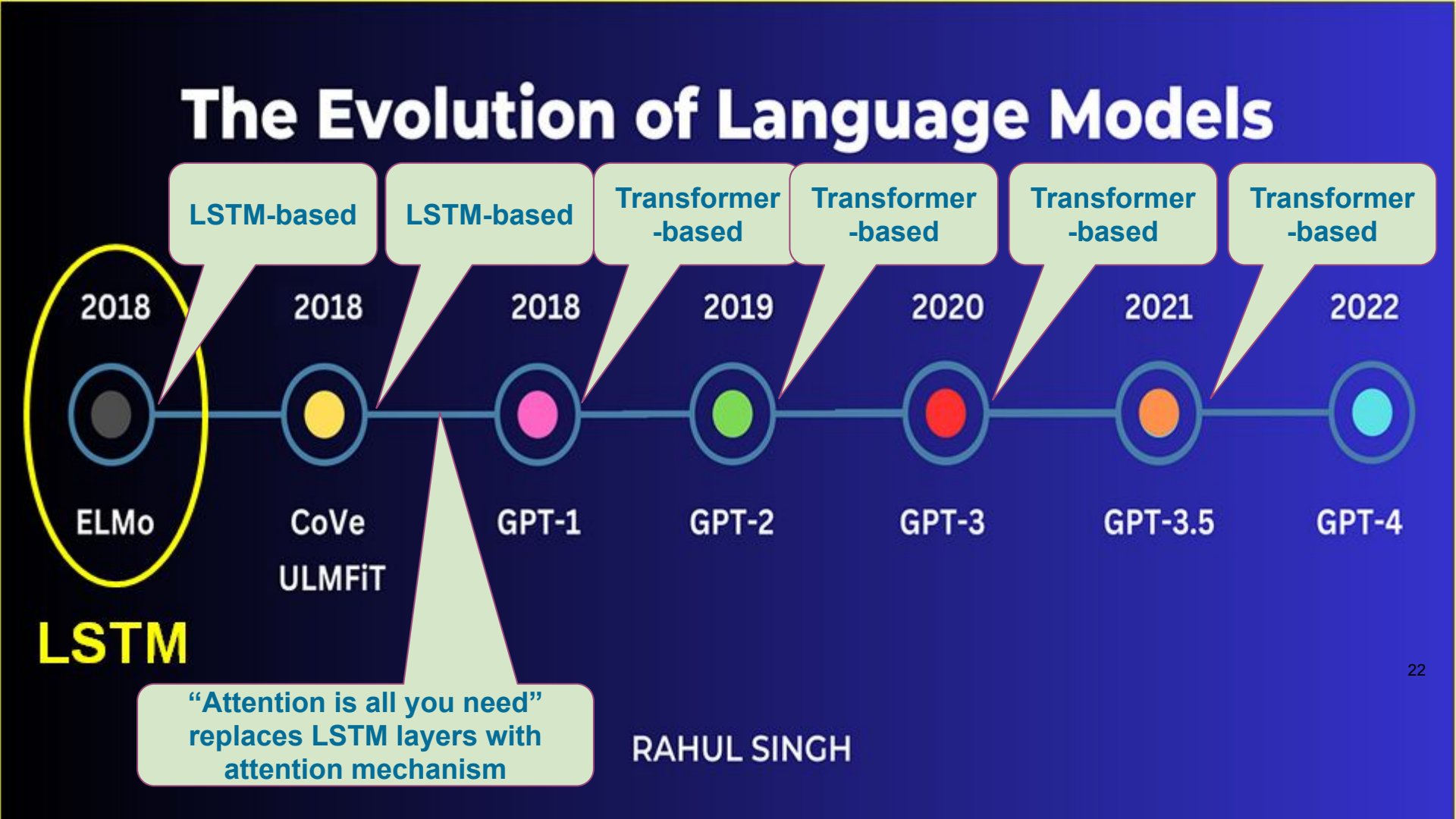
Andrej Karpathy

Founding member of OpenAI

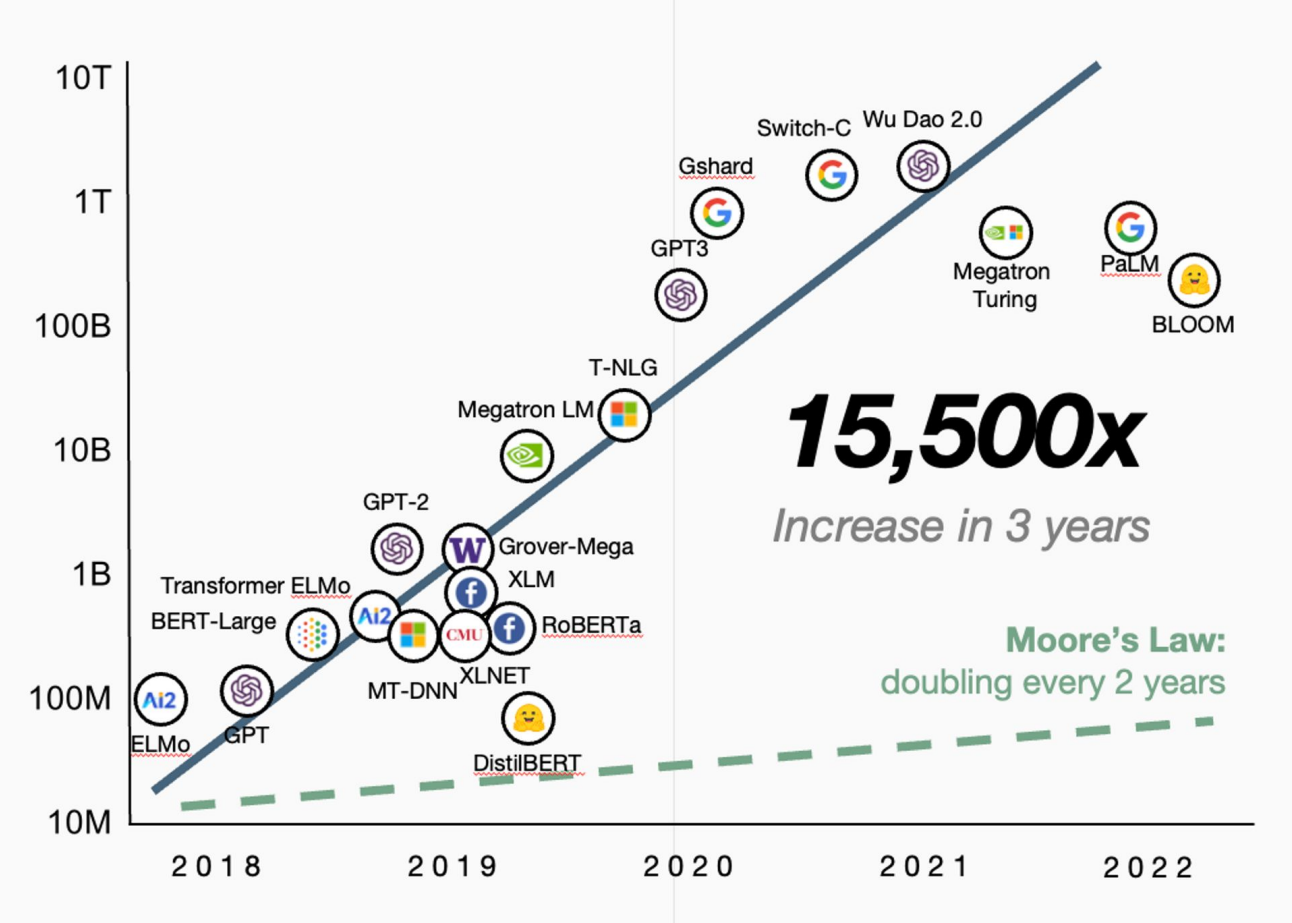
Overview

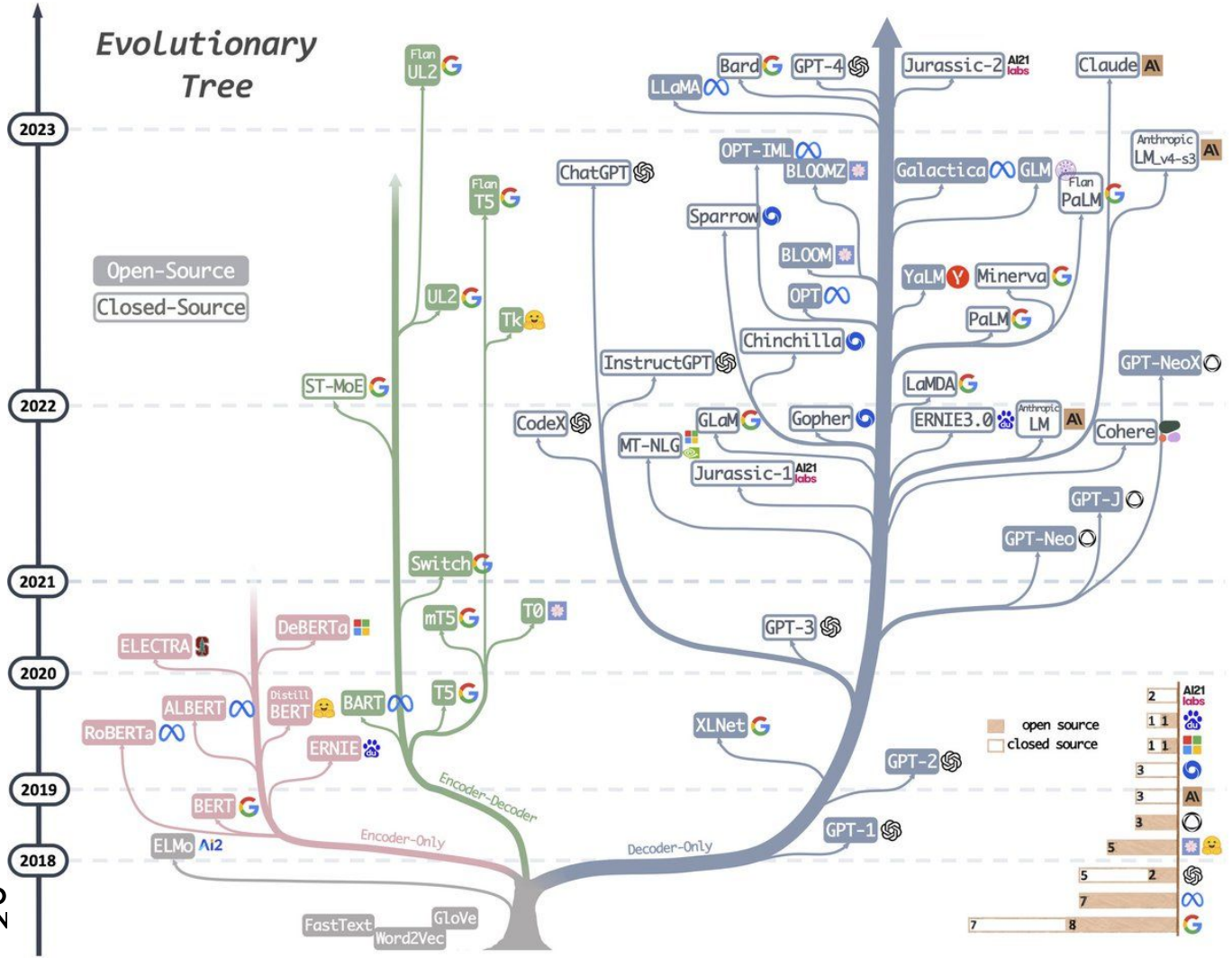
	RNNs/LSTM $\mathcal{O}(T)$	Transformer	xLSTM
Context handling	Theory: infinite context length Practice: few hundred steps		
Memory usage	Constant memory usage 		
Parallelizability	Not parallelizable across time steps		

The age of Transformers


The Evolution of Language Models
LSTM-based
LSTM-based
Transformer -based
Transformer -based
Transformer -based
Transformer -based
2018
2018
2018
2019
2020
2021
2022
ELMo
CoVe
ULMFiT
GPT-1
GPT-2
GPT-3
GPT-3.5
GPT-4
LSTM
"Attention is all you need" replaces LSTM layers with attention mechanism
RAHUL SINGH

The age of Transformers and LLMs

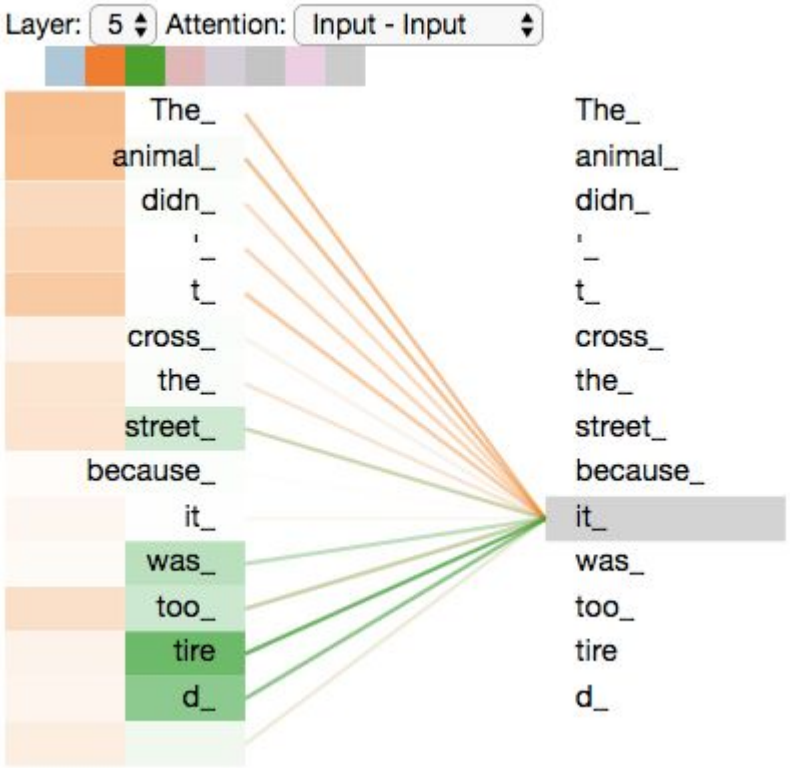



Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Flan UL2
Bard
GPT-4
Jurassic-2
Claude
LLaMA
OPT-IML
BLOOMZ
Galactica
GLM
Anthropic LM_v4-s3
Flan T5
Flan PaLM
ChatGPT
Sparrow
BLOOM
YaLM
Minerva
UL2
OPT
Tk
PaLM
Chinchilla
InstructGPT
GPT-NeoX
ST-MoE
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
Switch
mT5
T0
GPT-3
ELECTRA
DeBERTa
ALBERT
Distill BERT
BART
T5
RoBERTa
ERNIE
XLNet
GPT-2
BERT
Encoder-Decoder
Encoder-Only
Decoder-Only
GPT-1
ELMo
FastText
GloVe
Word2Vec
open source
closed source
AI21 labs
2
1 1
1 1
3
3
3
5
5 2
7
7 8

Transformer and self-attention

Attention mechanism

- Dot product between
  - Keys
  - Queries
- All time steps (tokens) can be processed in parallel
- Softmax for sparseness
- **Quadratic in time steps**



Transformer and self-attention

Generative Pre-trained Transformers (GPT)

- Input as set

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$$

- Main operation (parallel!)

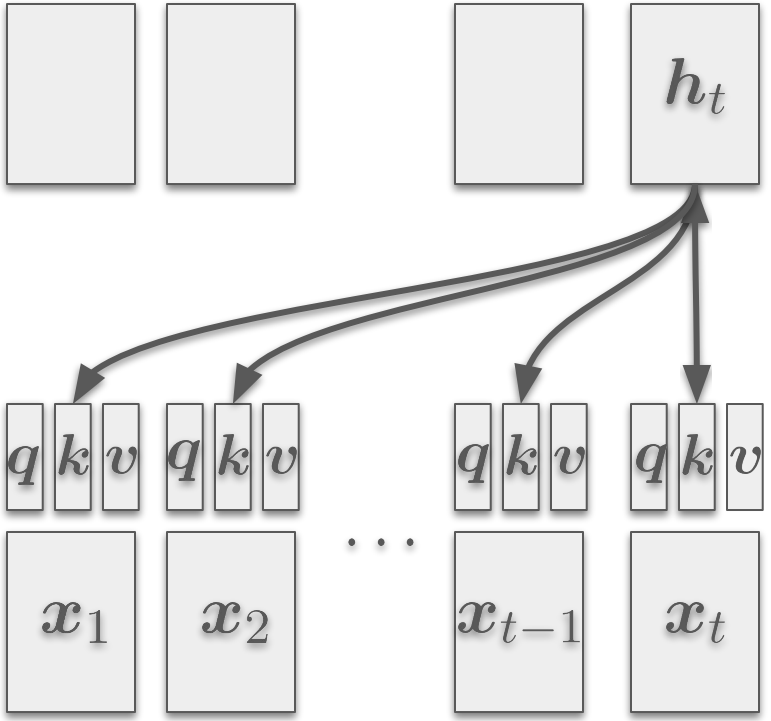
$$\boldsymbol{H}^0 = \boldsymbol{X}$$

$$\boldsymbol{K}^{(l)} = \boldsymbol{W}_K^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{Q}^{(l)} = \boldsymbol{W}_Q^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{V}^{(l)} = \boldsymbol{W}_V^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{H}^{(l)} = \underbrace{\text{softmax}\left(1/\sqrt{d_k}\ \boldsymbol{Q}^{(l)^T} \boldsymbol{K}^{(l)}\right)}_{:=A^{(l)}} \boldsymbol{V}^{(l)},$$



Transformer and self-attention

Generative Pre-trained Transformers (GPT)

- Input as set

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$$

- Main operation (parallel!)

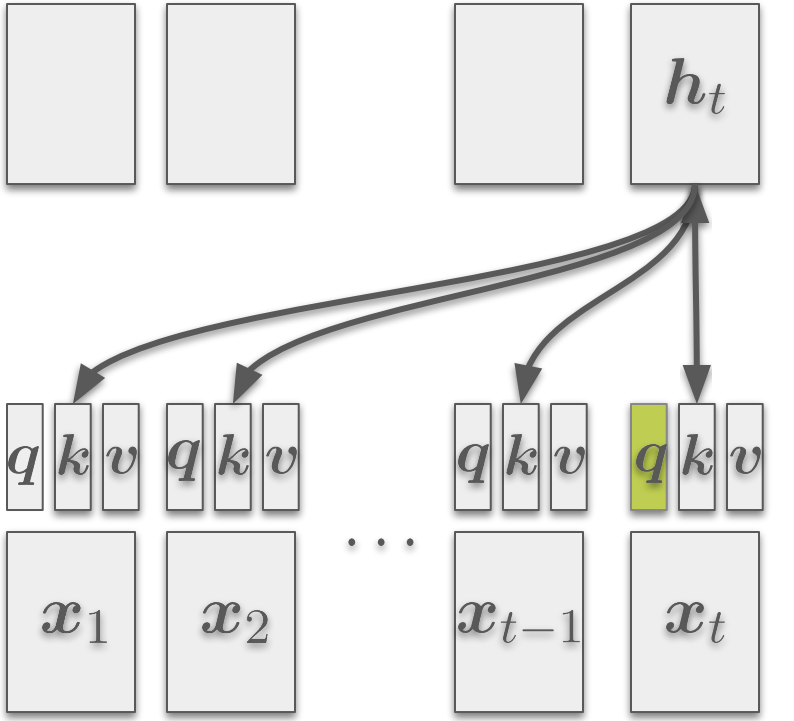
$$\boldsymbol{H}^0 = \boldsymbol{X}$$

$$\boldsymbol{K}^{(l)} = \boldsymbol{W}_K^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{Q}^{(l)} = \boldsymbol{W}_Q^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{V}^{(l)} = \boldsymbol{W}_V^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{H}^{(l)} = \underbrace{\text{softmax}\left(1/\sqrt{d_k}\ \boldsymbol{Q}^{(l)^T} \boldsymbol{K}^{(l)}\right)}_{:=A^{(l)}} \boldsymbol{V}^{(l)},$$



Transformer and self-attention

Generative Pre-trained Transformers (GPT)

- Input as set

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$$

- Main operation (parallel!)

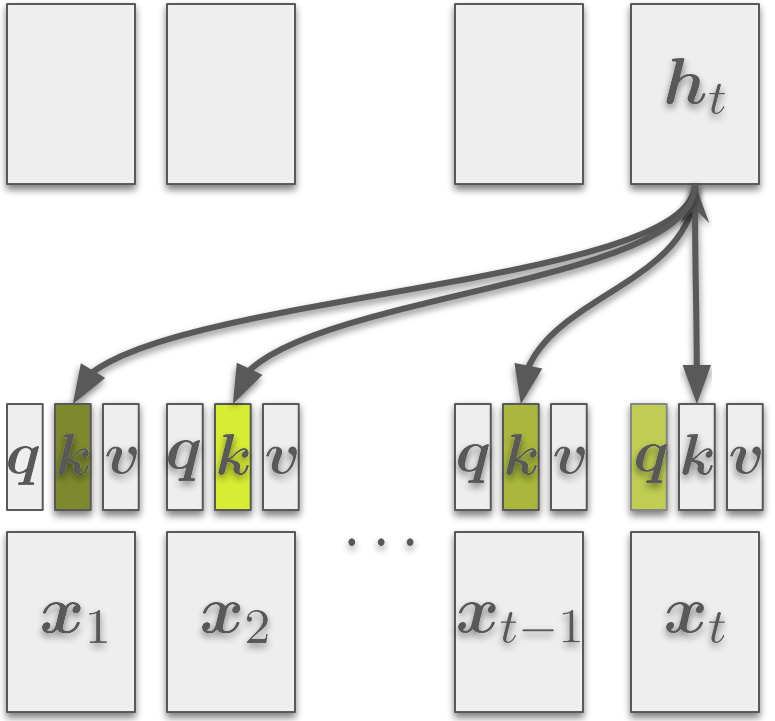
$$\boldsymbol{H}^0 = \boldsymbol{X}$$

$$\boldsymbol{K}^{(l)} = \boldsymbol{W}_K^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{Q}^{(l)} = \boldsymbol{W}_Q^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{V}^{(l)} = \boldsymbol{W}_V^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{H}^{(l)} = \underbrace{\text{softmax}\left(1/\sqrt{d_k}\ \boldsymbol{Q}^{(l)^T} \boldsymbol{K}^{(l)}\right)}_{:=A^{(l)}} \boldsymbol{V}^{(l)},$$



Transformer and self-attention

Generative Pre-trained Transformers (GPT)

- Input as set

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$$

- Main operation (parallel!)

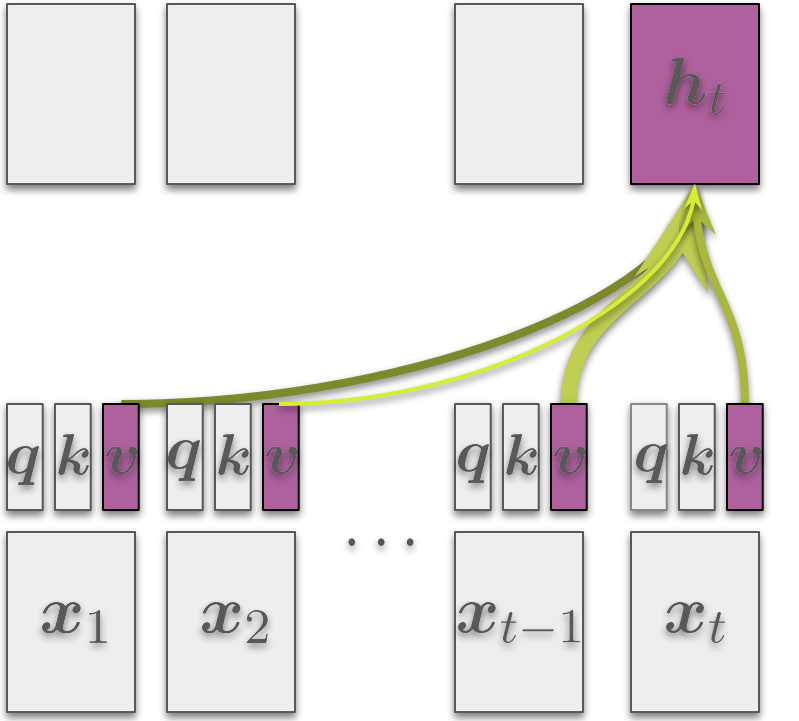
$$\boldsymbol{H}^0 = \boldsymbol{X}$$

$$\boldsymbol{K}^{(l)} = \boldsymbol{W}_K^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{Q}^{(l)} = \boldsymbol{W}_Q^{(l)} \boldsymbol{H}^{(l-1)}$$

$$\boldsymbol{V}^{(l)} = \boldsymbol{W}_V^{(l)} \boldsymbol{H}^{(l-1)}$$

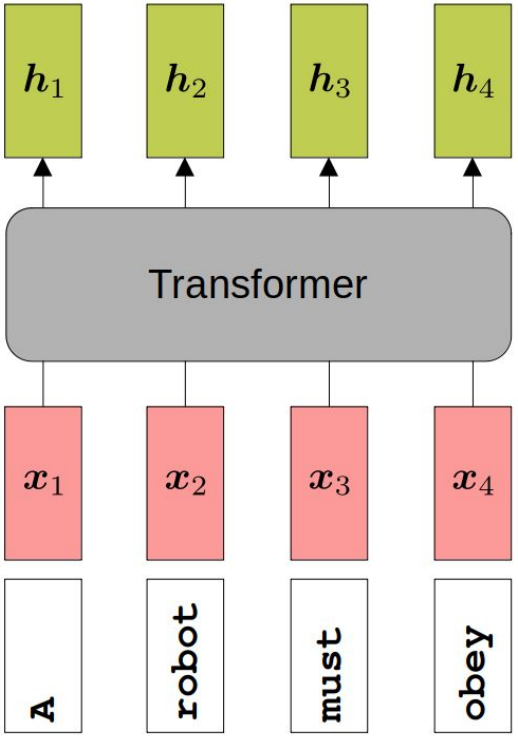
$$\boldsymbol{H}^{(l)} = \underbrace{\text{softmax}\left(1/\sqrt{d_k}\ {\boldsymbol{Q}^{(l)}}^T \boldsymbol{K}^{(l)}\right)}_{:=A^{(l)}} \boldsymbol{V}^{(l)},$$



Parallelizability of the transformer

Attention mechanism

- Dot product between
  - Keys
  - Queries
- All time steps (tokens) can be processed in parallel
- Softmax for sparseness
- **Quadratic in context size**



Overview

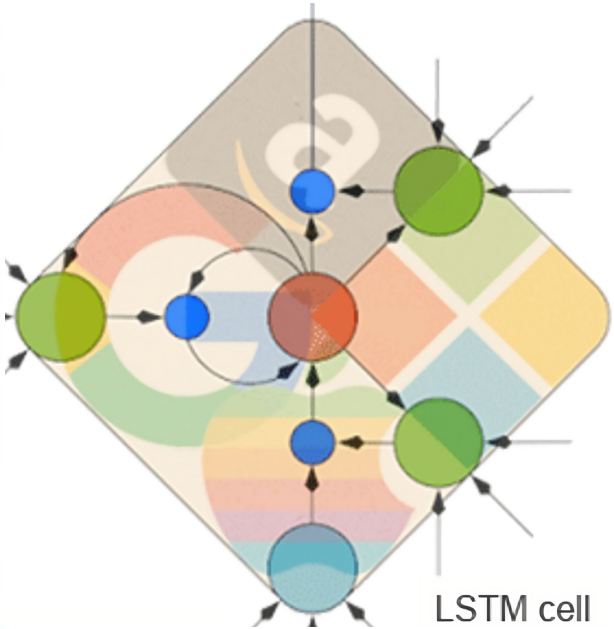
	RNNs/LSTM $\mathcal{O}(T)$	Transformer $\mathcal{O}(T^2)$	xLSTM
Context handling	Theory: infinite context length Practice: few hundred inputs	Strong long-range capacity Fixed context window ✓	
Memory usage	Constant memory usage ✓	Memory usage grows with context size	
Parallelizability	Not parallelizable across time steps	Parallelizable across time steps ✓	

The age of recurrent LLMs?

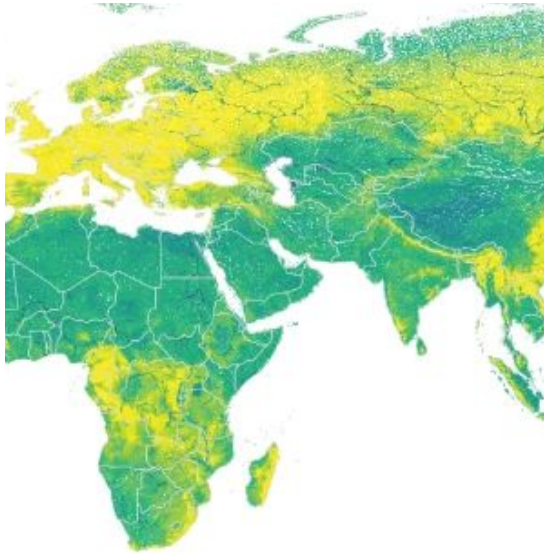
LSTM and the test-of-time

LSTMs in reinforcement learning, e.g. OpenAI Five

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., ... & Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.



LSTMs as basis of language models (until 2017)



LSTMs in hydrology

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., ... & Matias, Y. (2024). Global prediction of extreme floods in ungauged watersheds. *Nature*, *627*(8004), 559-563.

Long short-term memory and the constant error carousel (CEC)

$$c_t = \mathrm{f}_t \, c_{t-1} + \mathrm{i}_t \, z_t$$

$$h_t = \mathrm{o}_t \, \psi(c_t)$$

- Cell state c_t
- Cell inputs z_t
- Forget gate f_t
- Input gate i_t

Review of LSTM

$$c_t = \mathrm{f}_t \; c_{t-1} + \mathrm{i}_t \; z_t \qquad \text{cell state}$$

$$h_t = \mathrm{o}_t \; \tilde{h}_t \,, \qquad \tilde{h}_t = \psi\left(c_t\right) \qquad \text{hidden state}$$

$$z_t = \varphi\left(\tilde{z}_t\right) \,, \qquad \tilde{z}_t = \boldsymbol{w}_z^\top \boldsymbol{x}_t + r_z h_{t-1} + b_z \qquad \text{cell input}$$

$$\mathrm{i}_t = \sigma\left(\tilde{\mathrm{i}}_t\right) \,, \qquad \tilde{\mathrm{i}}_t = \boldsymbol{w}_\mathrm{i}^\top \boldsymbol{x}_t + r_\mathrm{i} \, h_{t-1} + b_\mathrm{i} \qquad \text{input gate}$$

$$\mathrm{f}_t = \sigma\left(\tilde{\mathrm{f}}_t\right) \,, \qquad \tilde{\mathrm{f}}_t = \boldsymbol{w}_\mathrm{f}^\top \boldsymbol{x}_t + r_\mathrm{f} \, h_{t-1} + b_\mathrm{f} \qquad \text{forget gate}$$

$$\mathrm{o}_t = \sigma\left(\tilde{\mathrm{o}}_t\right) \,, \qquad \tilde{\mathrm{o}}_t = \boldsymbol{w}_\mathrm{o}^\top \boldsymbol{x}_t + r_\mathrm{o} \, h_{t-1} + b_\mathrm{o} \qquad \text{output gate}$$

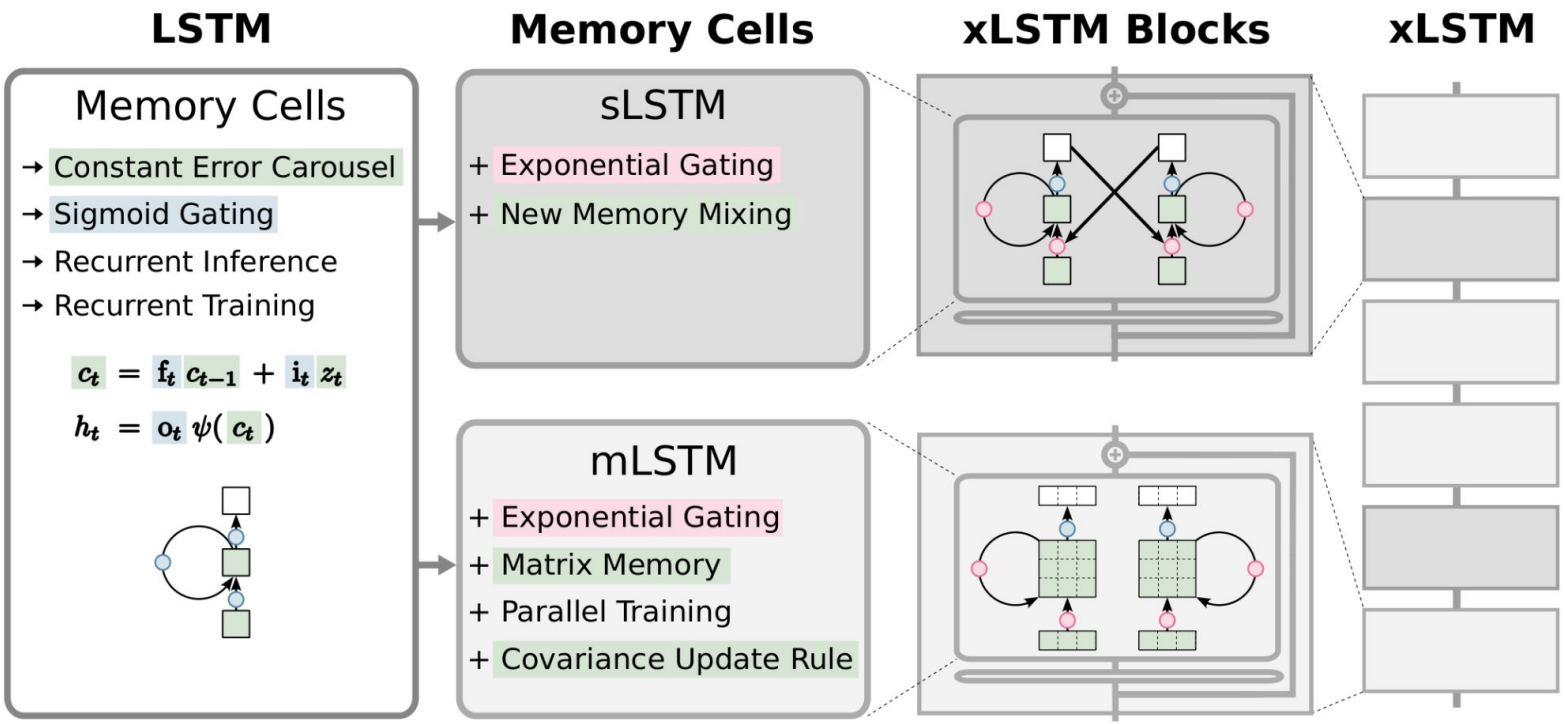
Note: scalar notation for cell state!

How far do we get with LSTM in language modeling?

when

- scaling LSTMs to **billions of parameters**
- leveraging the **latest techniques** from LLMs
- **mitigating known limitations** of LSTMs?

Overview: from LSTM to xLSTM



LSTM limitations

- Inability to revise storage decisions
  - Nearest Neighbor Search
- Limited storage capacities
  - Rare Token Prediction
  - problem with new or rare tokens
- No parallelization for training

sLSTM

Revise storage decisions through

- exponential gating
- exponential input gate and normalization scales down old inputs

sLSTM

$$c_t = \mathrm{f}_t \; c_{t-1} + \mathrm{i}_t \; z_t \qquad \text{cell state} \qquad (8)$$

$$n_t = \mathrm{f}_t \; n_{t-1} + \mathrm{i}_t \qquad \text{normalizer state} \qquad (9)$$

$$h_t = \mathrm{o}_t \; \tilde{h}_t \,, \qquad \tilde{h}_t = c_t \,/\, n_t \qquad \text{hidden state} \qquad (10)$$

$$z_t = \varphi\left(\tilde{z}_t\right) \,, \qquad \tilde{z}_t = \boldsymbol{w}_z^\top \boldsymbol{x}_t + r_z h_{t-1} + b_z \qquad \text{cell input} \qquad (11)$$

$$\mathrm{i}_t = \exp\left(\tilde{\mathrm{i}}_t\right) \,, \qquad \tilde{\mathrm{i}}_t = \boldsymbol{w}_\mathrm{i}^\top \boldsymbol{x}_t + r_\mathrm{i} \, h_{t-1} + b_\mathrm{i} \qquad \text{input gate} \qquad (12)$$

$$\mathrm{f}_t = \sigma\left(\tilde{\mathrm{f}}_t\right) \text{ OR } \exp\left(\tilde{\mathrm{f}}_t\right) \,, \qquad \tilde{\mathrm{f}}_t = \boldsymbol{w}_\mathrm{f}^\top \boldsymbol{x}_t + r_\mathrm{f} \, h_{t-1} + b_\mathrm{f} \qquad \text{forget gate} \qquad (13)$$

$$\mathrm{o}_t = \sigma\left(\tilde{\mathrm{o}}_t\right) \,, \qquad \tilde{\mathrm{o}}_t = \boldsymbol{w}_\mathrm{o}^\top \boldsymbol{x}_t + r_\mathrm{o} \, h_{t-1} + b_\mathrm{o} \qquad \text{output gate} \qquad (14)$$

Revise storage decisions: $i_t = \infty, \quad \tilde{h}_t = z_t$

sLSTM: scalar LSTM

Memory $\boldsymbol{c} \in \mathbb{R}^m$ with m memory cells

Memory capacity: m/d where d is the hidden dimension

The memory is independent of:

- sequence length
- hidden dimension

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}$$

mLSTM

- **Limited storage capacities**
- New memory structure:
  - **matrix memory**
  - **covariance update rule**
- Storage capacity is maximized for fixed memory

matrix memory | matrix memory | covariance update

$$C_t = C_{t-1} + v_t k_t^\top$$

memory retrieval

$$h_t = C_t q_t$$

mLSTM

$$\boldsymbol{C}_t = \mathrm{f}_t \; \boldsymbol{C}_{t-1} + \mathrm{i}_t \; \boldsymbol{v}_t \, \boldsymbol{k}_t^\top \qquad \text{cell state} \quad (19)$$

$$\boldsymbol{n}_t = \mathrm{f}_t \; \boldsymbol{n}_{t-1} + \mathrm{i}_t \; \boldsymbol{k}_t \qquad \text{normalizer state} \quad (20)$$

$$\boldsymbol{h}_t = \mathbf{o}_t \odot \tilde{\boldsymbol{h}}_t \,, \qquad \tilde{\boldsymbol{h}}_t = \boldsymbol{C}_t \; \boldsymbol{q}_t \; / \; \max\left\{ \left| \boldsymbol{n}_t^\top \; \boldsymbol{q}_t \right|, 1 \right\} \qquad \text{hidden state} \quad (21)$$

$$\boldsymbol{q}_t = \boldsymbol{W}_q \, \boldsymbol{x}_t + \boldsymbol{b}_q \qquad \text{query input} \quad (22)$$

$$\boldsymbol{k}_t = \frac{1}{\sqrt{d}} \boldsymbol{W}_k \, \boldsymbol{x}_t + \boldsymbol{b}_k \qquad \text{key input} \quad (23)$$

$$\boldsymbol{v}_t = \boldsymbol{W}_v \, \boldsymbol{x}_t + \boldsymbol{b}_v \qquad \text{value input} \quad (24)$$

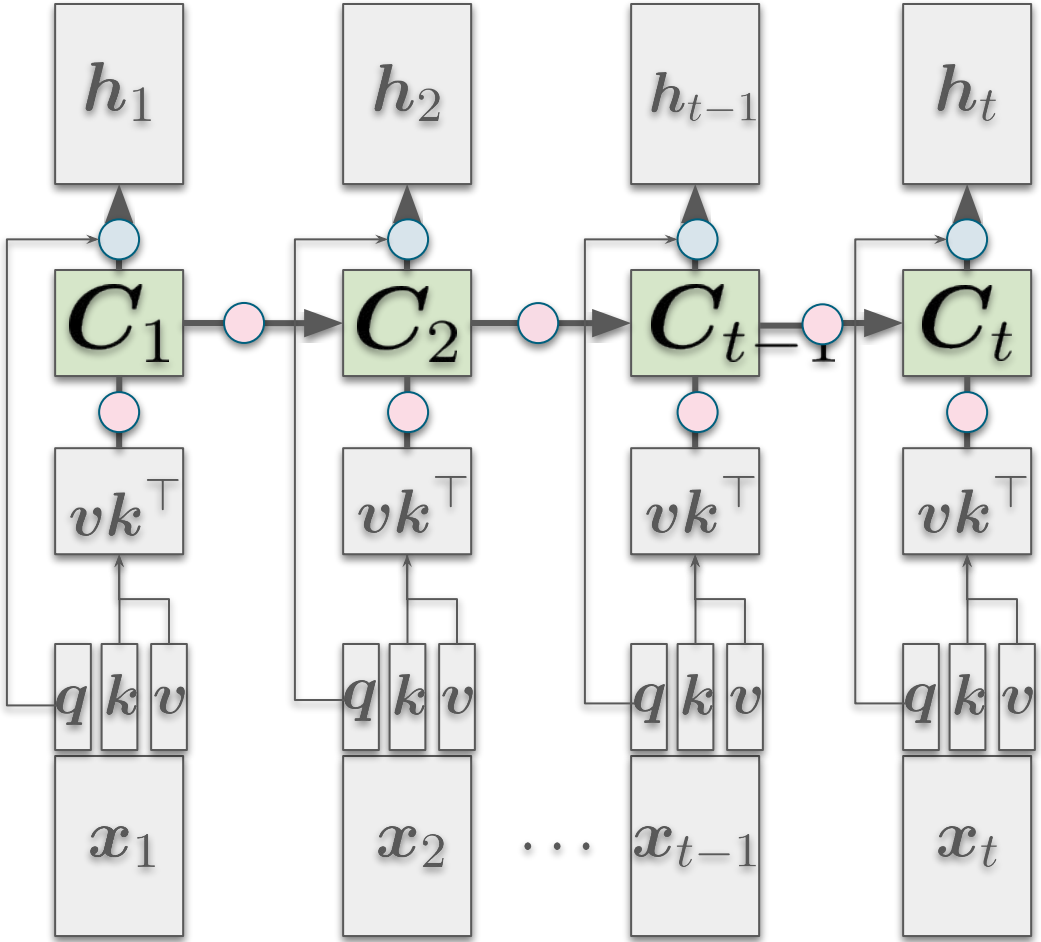
$$\mathrm{i}_t = \exp\left(\tilde{\mathrm{i}}_t\right) \,, \qquad \tilde{\mathrm{i}}_t = \boldsymbol{w}_\mathrm{i}^\top \, \boldsymbol{x}_t + b_\mathrm{i} \qquad \text{input gate} \quad (25)$$

$$\mathrm{f}_t = \sigma\left(\tilde{\mathrm{f}}_t\right) \text{ OR } \exp\left(\tilde{\mathrm{f}}_t\right), \quad \tilde{\mathrm{f}}_t = \boldsymbol{w}_\mathrm{f}^\top \, \boldsymbol{x}_t + b_\mathrm{f} \qquad \text{forget gate} \quad (26)$$

$$\mathbf{o}_t = \sigma\left(\tilde{\mathbf{o}}_t\right) \,, \qquad \tilde{\mathbf{o}}_t = \boldsymbol{W}_\mathbf{o} \, \boldsymbol{x}_t + \boldsymbol{b}_\mathbf{o} \qquad \text{output gate} \quad (27)$$

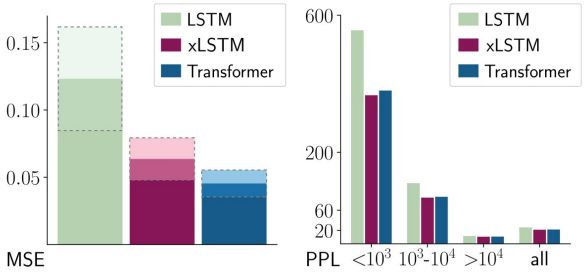
Parallelization: gates, keys, queries, values all independent from h_{t-1}

Parallelizability of mLSTM

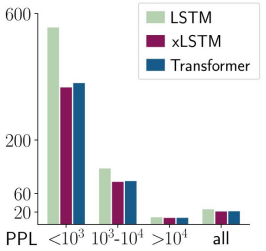


LSTM limitations

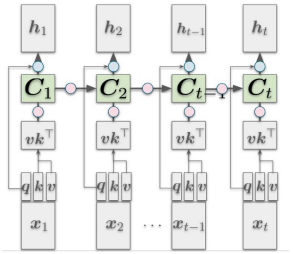
- Inability to revise storage decisions
  - Nearest Neighbor Search



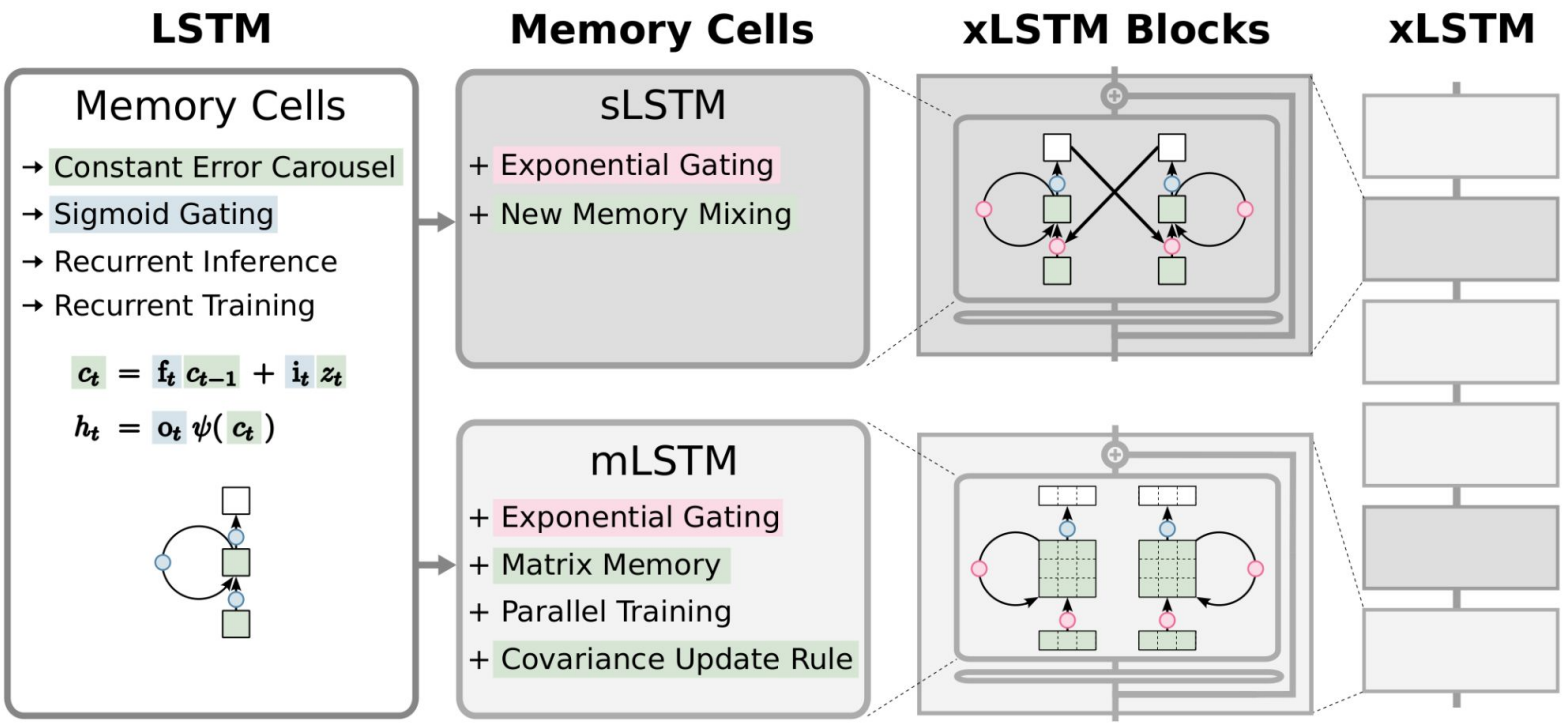
- Limited storage capacities
  - Rare Token Prediction
  - problem with new or rare tokens



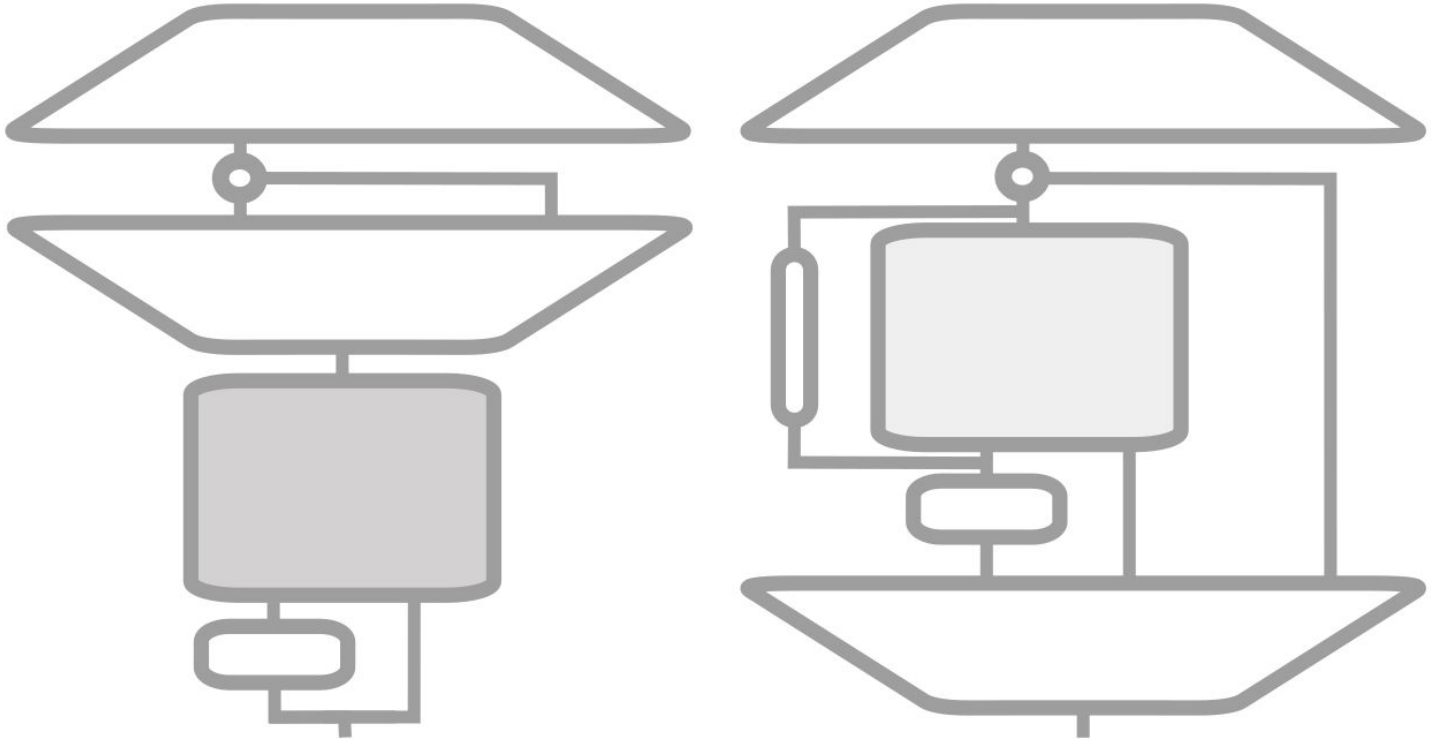
- No parallelization for training



xLSTM: mLSTM & sLSTM



Backbone architecture

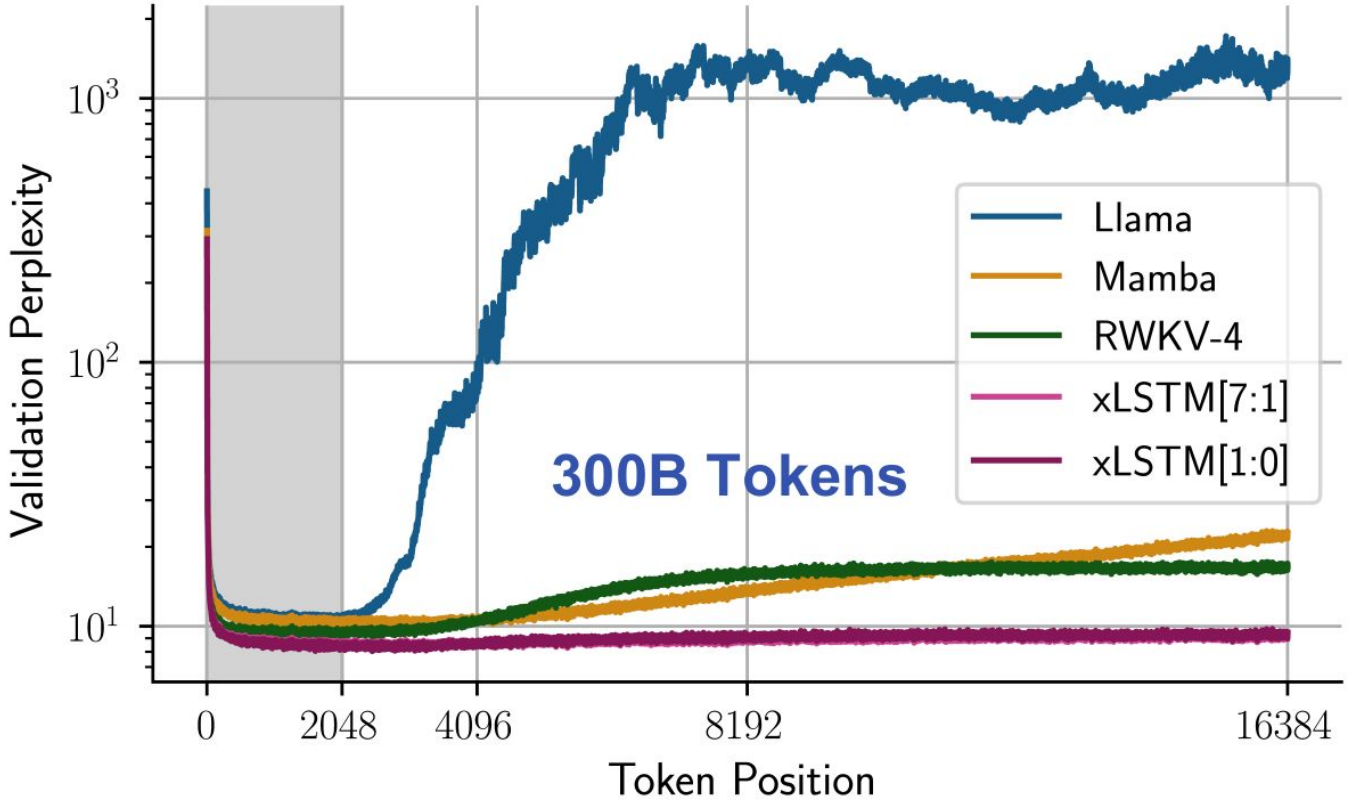


xLSTM: Length extrapolation

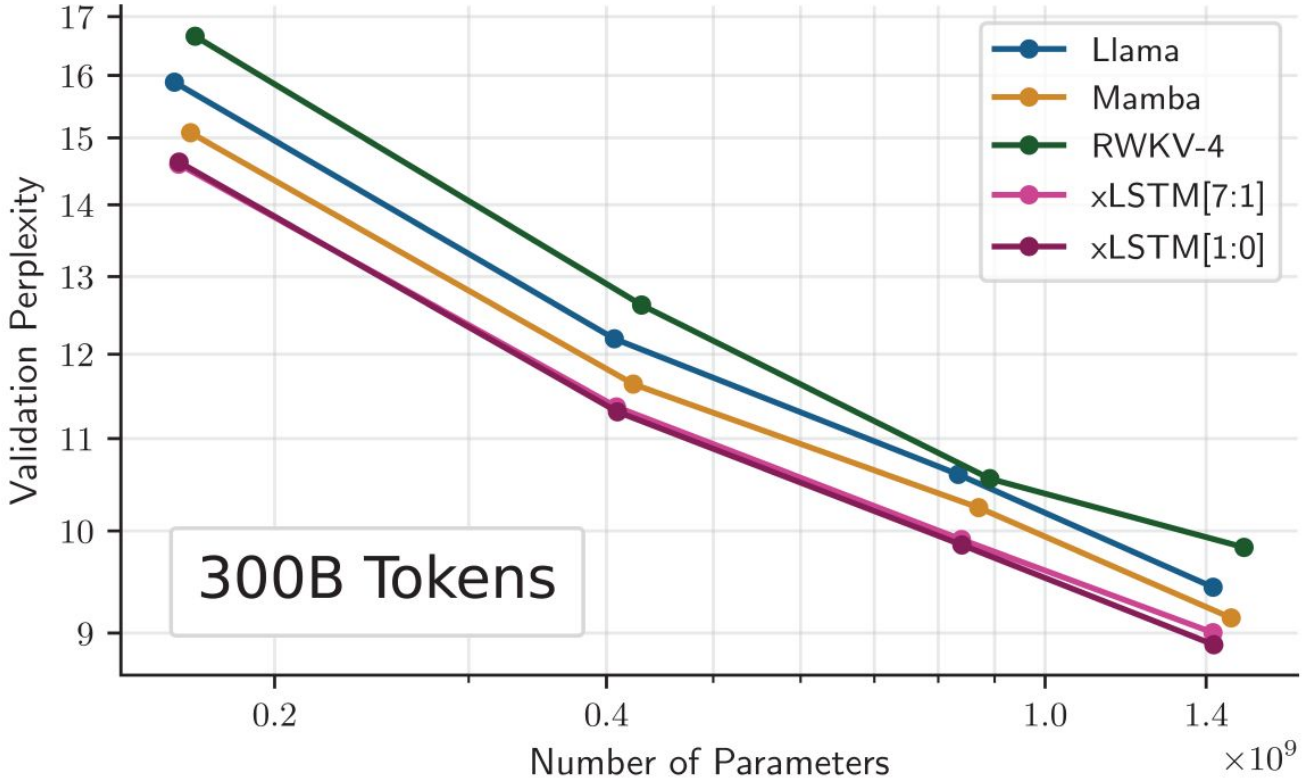
Sequence extrapolation of 1.3B models trained on SlimPajama 300B.

Training length: 2048

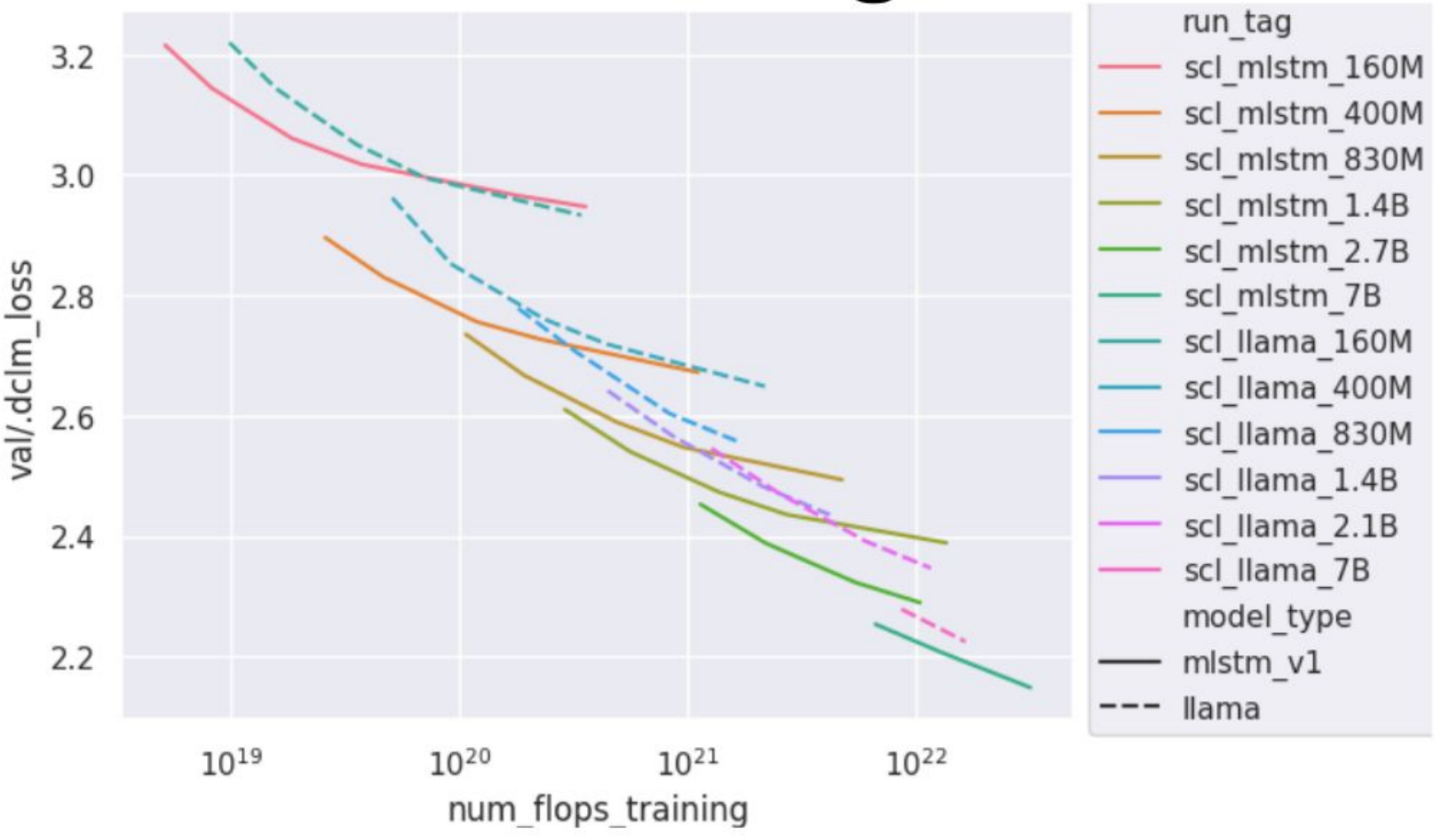
Test length: up to 16384.



xLSTM Scaling Laws

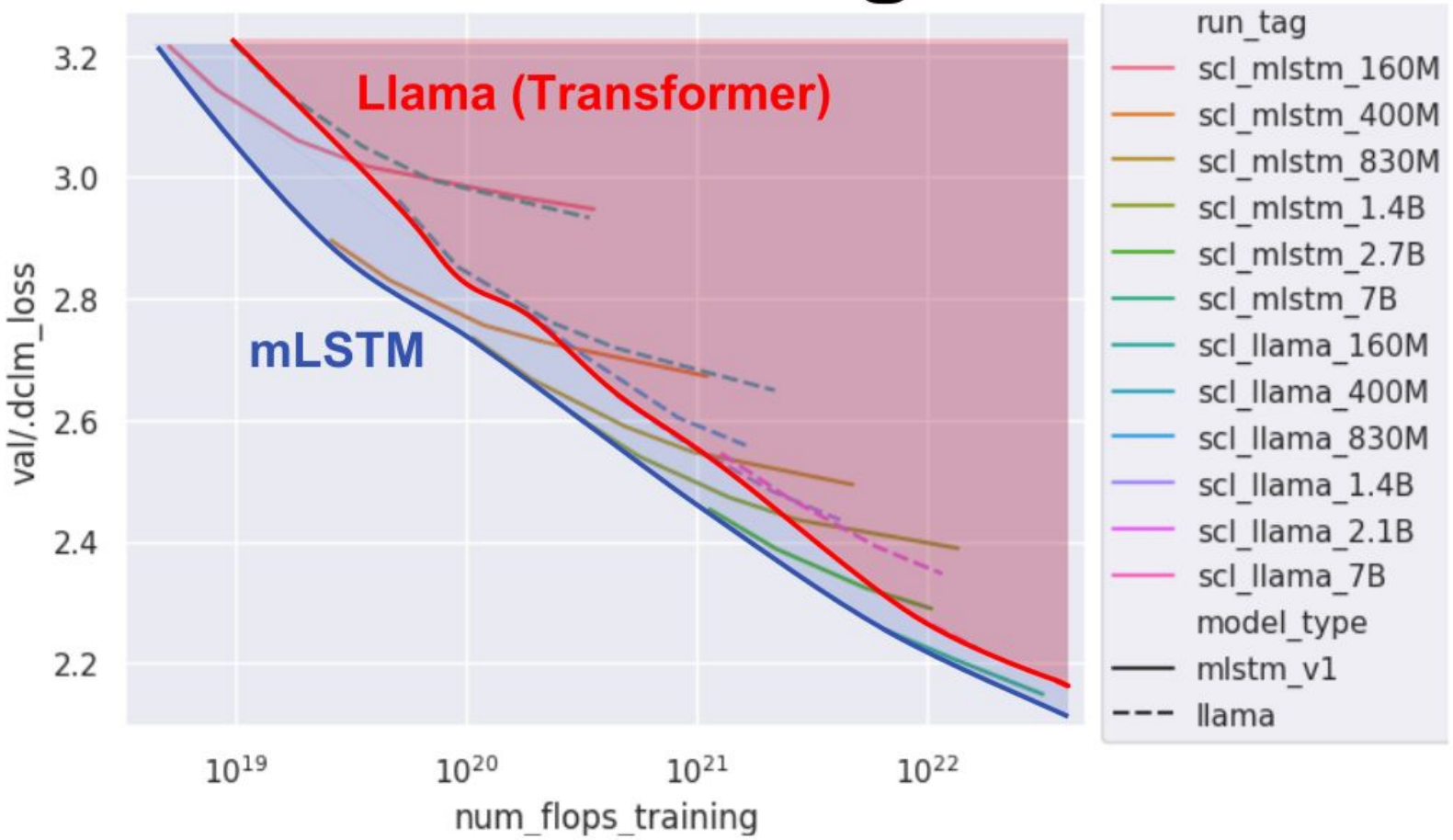


xLSTM: Scaling Laws



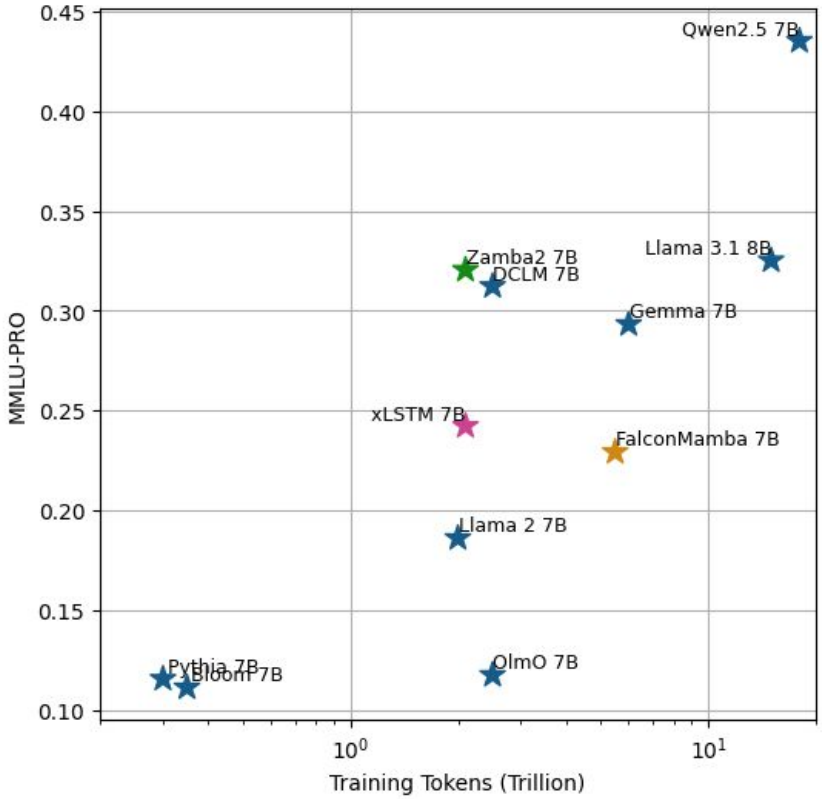
xLSTM: Scaling Laws

Pareto front



xLSTM-7B: A Recurrent LLM for Fast and Efficient Inference

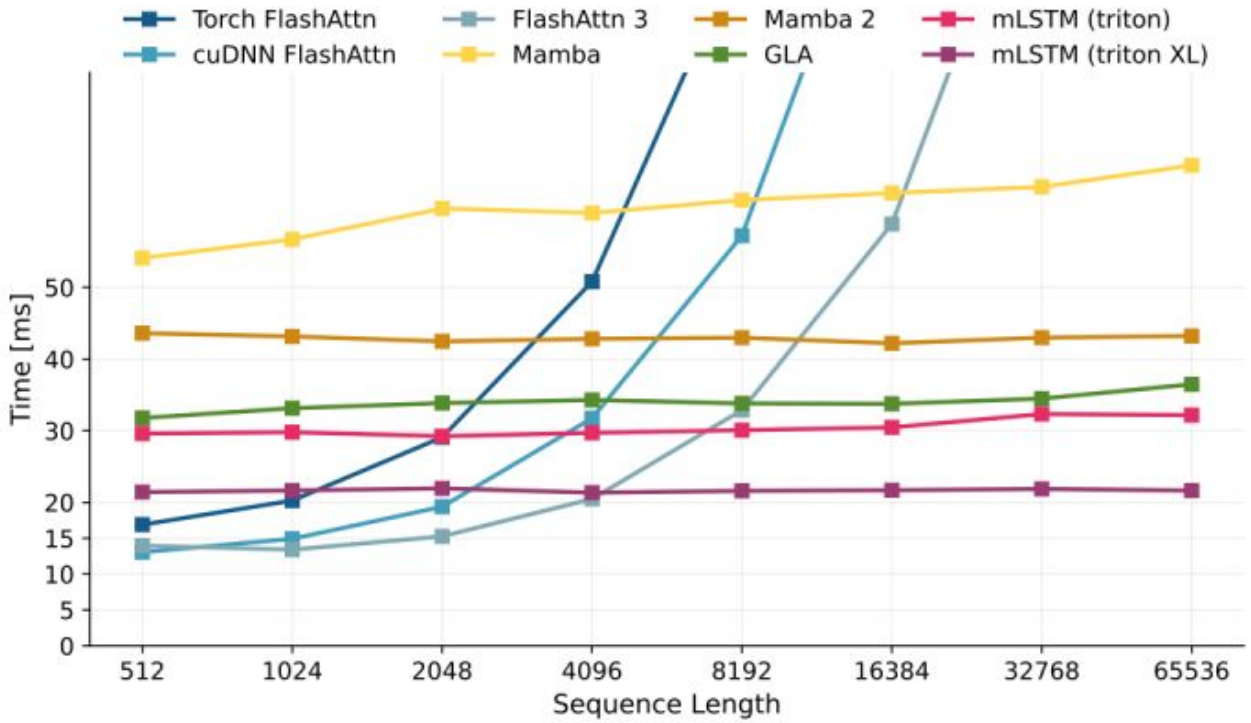
- First LSTM-based 7B language model
- Language modeling performance similar to other linear LLMs
- More efficient (faster)
- Several changes to architectures
  - E.g. fully based on mLSTM
  - Changes in normalization and gating
- **Pareto front:** best choice given fixed computational budget
- **Computing:**
  - Cluster with 256 NVIDIA H100 GPUs
  - Pre-Training: ~140k GPU hours, 2.3T tokens, +Long-Context Version



Beck, M., Pöppel, K., Lippe, P., Kurle, R., Blies, P. M., Klambauer, G., ... & Hochreiter, S. (2025). xLSTM 7B: A Recurrent LLM for Fast and Efficient Inference. *International Conference on Machine Learning (ICML)*.

xLSTM: efficient kernels hardware optimized

- Triton kernels render xLSTM extremely fast



Overview

	RNNs/LSTM $\mathcal{O}(T)$	**Transformer** $\mathcal{O}(T^2)$	**xLSTM** $\mathcal{O}(T)$
Context handling	Theory: infinite context length Practice: few hundred inputs	Strong long-range capacity Fixed context window ✓	Strong long-range capacity Length extrapolation ✓
Memory usage	Constant memory usage ✓	Memory usage grows with context size	Constant memory usage ✓
Parallelizability	Not parallelizable across time steps	Parallelizable across time steps ✓	Parallelizable across time steps ✓

xLSTM

- **New:**
  - **exponential gating and memory mixing**
  - **memory structure (matrix memory and covariance update)**

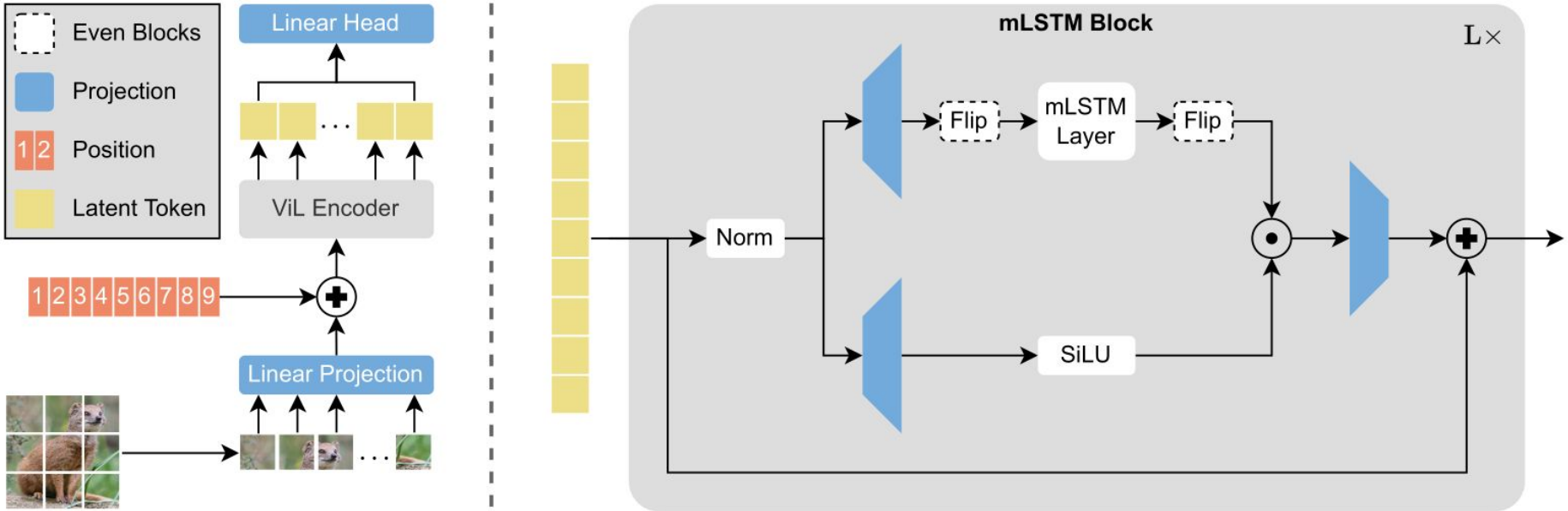
- **Results:**
  - xLSTM performs favorably on language modeling when compared to Transformers and State Space models

  - The **scaling laws** indicate that larger xLSTM models will be serious competitors to current large language models

Resources for xLSTM

Vision-xLSTM (ViL)

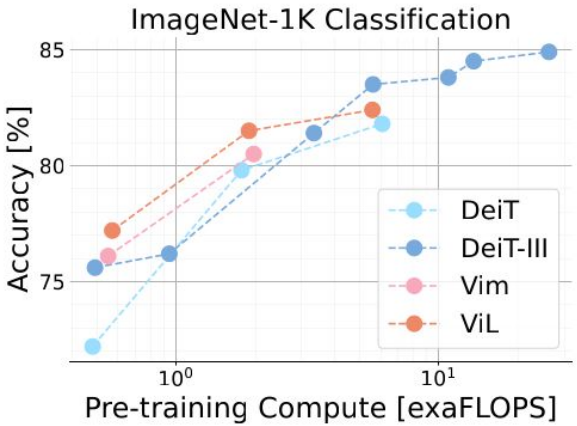
Vision-xLSTM (ViL)

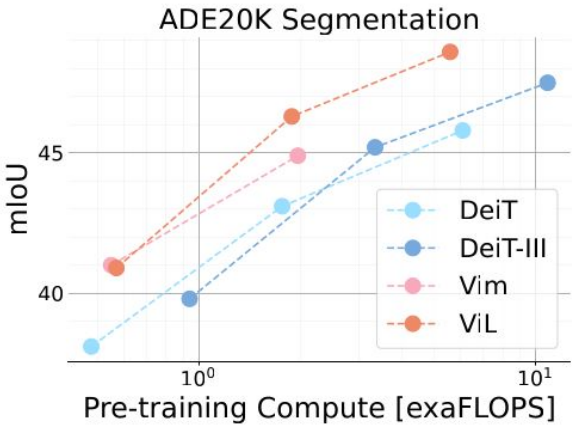


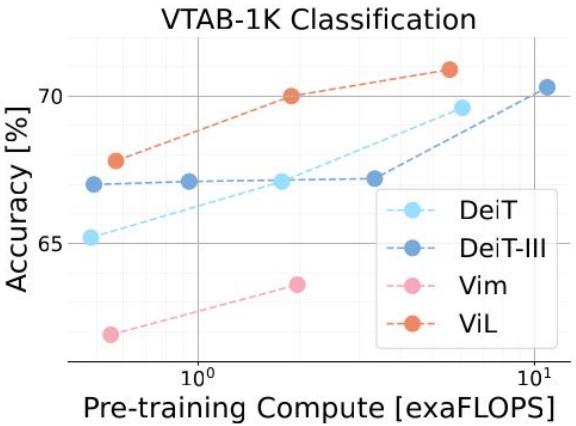
- xLSTM can process image “tokens”

Alkin, B., Beck, M., Pöppel, K., Hochreiter, S., & Brandstetter, J. (2025). Vision-LSTM: xLSTM as Generic Vision Backbone. *International Conference on Learning Representations (ICLR)*

Vision-xLSTM (ViL)







- High efficiency and accuracy

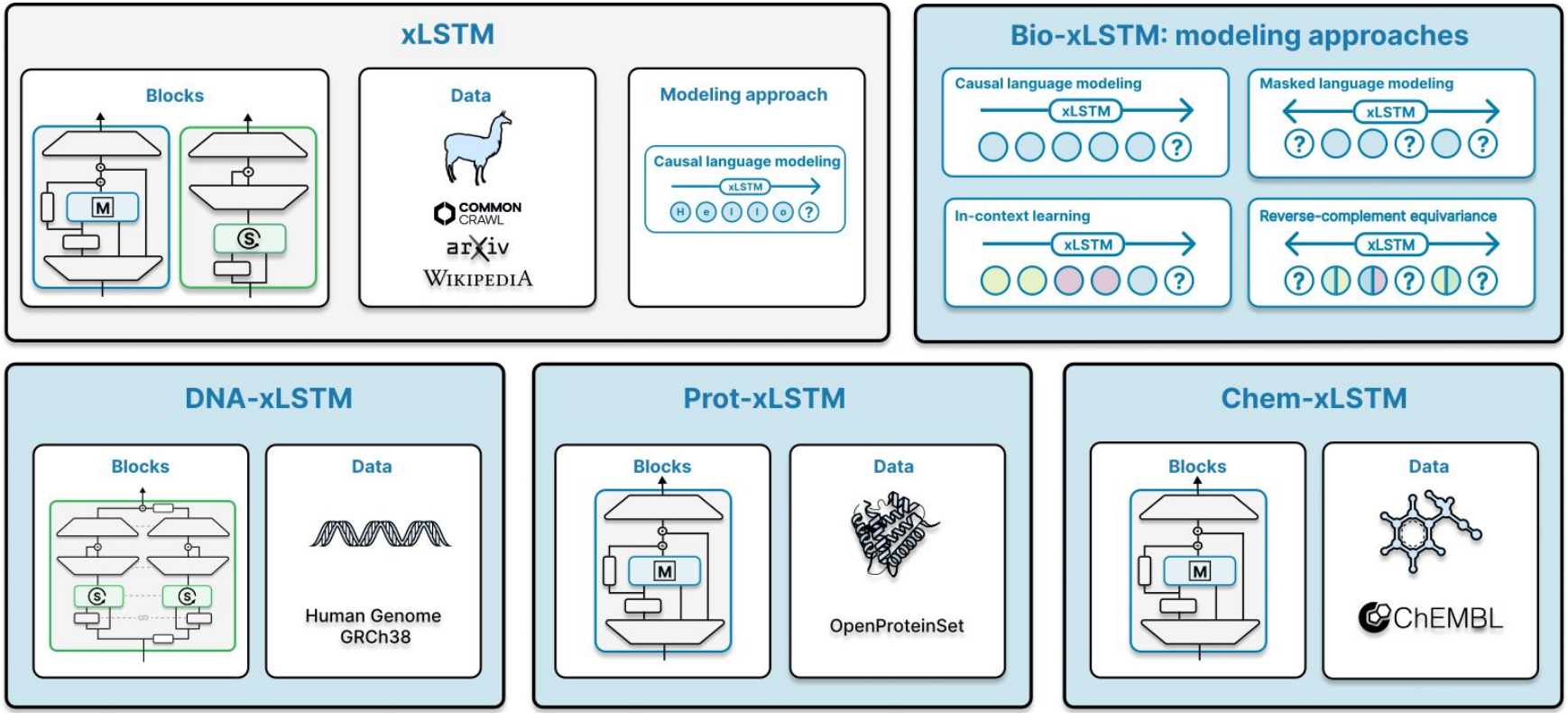
Alkin, B., Beck, M., Pöppel, K., Hochreiter, S., & Brandstetter, J. (2025). Vision-LSTM: xLSTM as Generic Vision Backbone. *International Conference on Learning Representations (ICLR)*

Bio-xLSTM

Motivation

- **Biological Foundation Models:** large models trained on biological domains capture important concepts, which enables downstream applications.
- **Transformer Dominance:** Transformers remain dominant due to their versatility and performance at large scales.
- **Scalability & Memory:** Transformers face runtime and memory challenges. Efficient architectures are critical for processing long-range biological data efficiently.
- **xLSTM:** xLSTM has shown strong results in natural language tasks. This work evaluates its scalability and performance in biological applications.

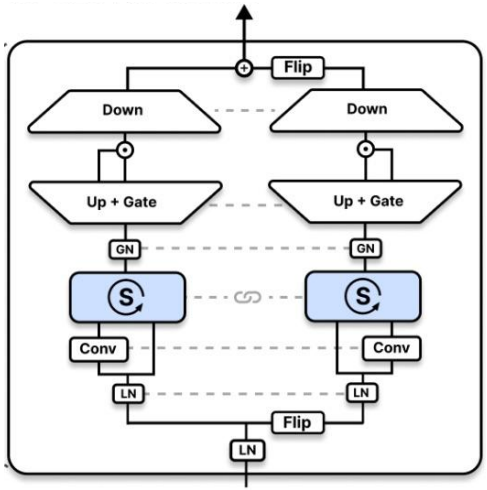
Bio-xLSTM



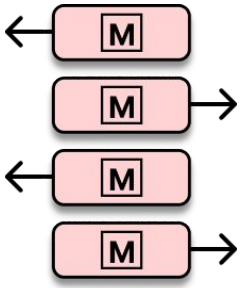
Schmidinger, N., Schneckenreiter, L., Seidl, P., Schimunek, J., Hoedt, P. J., Brandstetter, J., ... & Klambauer, G. (2025). Bio-xLSTM: Generative modeling, representation and in-context learning of biological and chemical sequences. *International Conference on Learning Representations 2025*.

DNA-xLSTM

xLSTM Bidirectionality



Weight Sharing



Alternating Block Directions

$$\overrightarrow{\boldsymbol{C}_t} = \sigma(\tilde{\mathrm{f}}_t)\ \overrightarrow{\boldsymbol{C}_{t-1}} + \exp(\tilde{\mathrm{i}}_t)\ \boldsymbol{v}_t \boldsymbol{k}_t^\top$$
$$\overleftarrow{\boldsymbol{C}_t} = \sigma(\tilde{\mathrm{f}}_t)\ \overleftarrow{\boldsymbol{C}_{t-1}} + \exp(\tilde{\mathrm{i}}_t)\ \boldsymbol{v}_t \boldsymbol{k}_t^\top$$

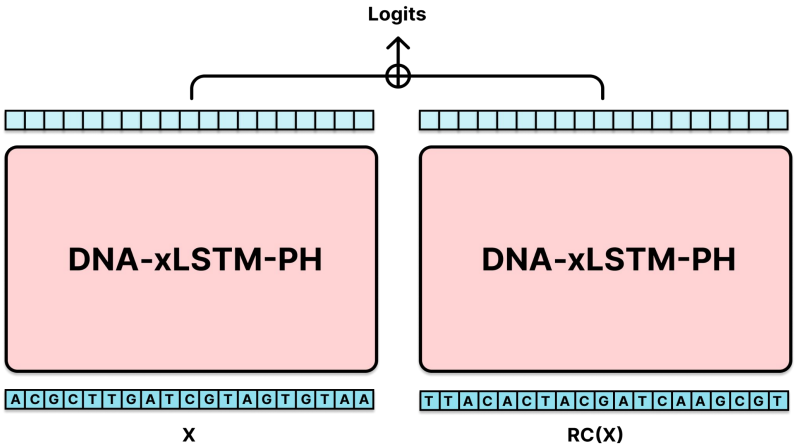
mLSTM Native Bidirectionality

ELMo, Peters et.al. (2018)

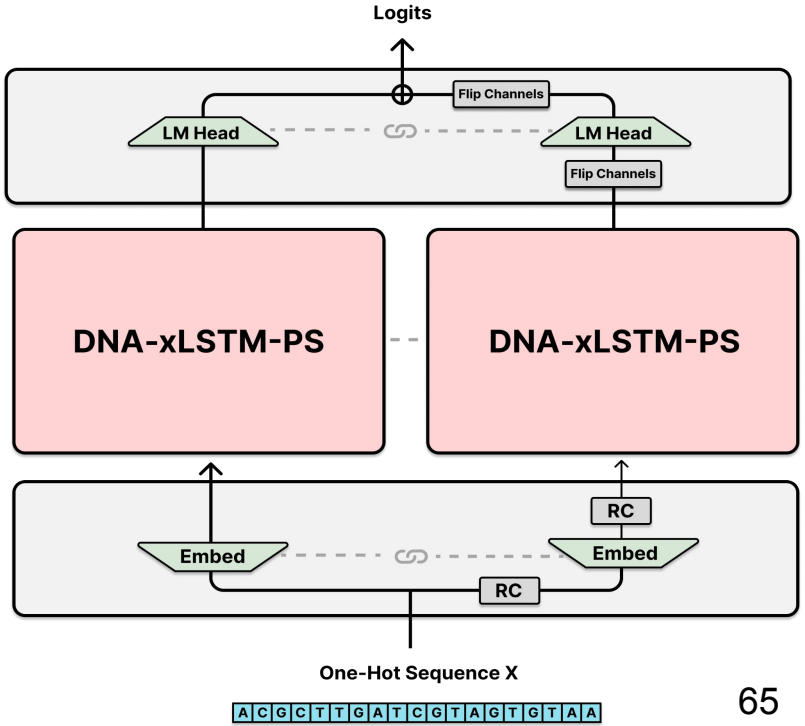
Vision-LSTM, Alkin et.al. (2024)

Reverse-Complement Invariance

$$\text{xLSTM}(\mathbf{X}) = \text{xLSTM}(\text{RC}(\mathbf{X}))$$

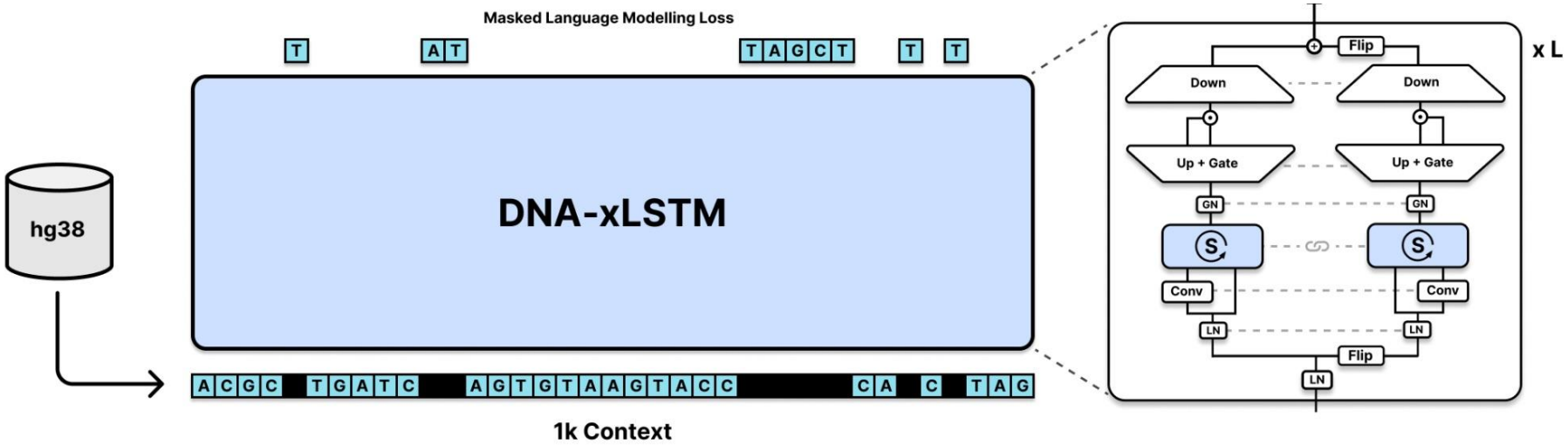


$$\text{xLSTM}(\text{RC}(\mathbf{X})) = \text{RC}(\text{xLSTM}(\mathbf{X}))$$

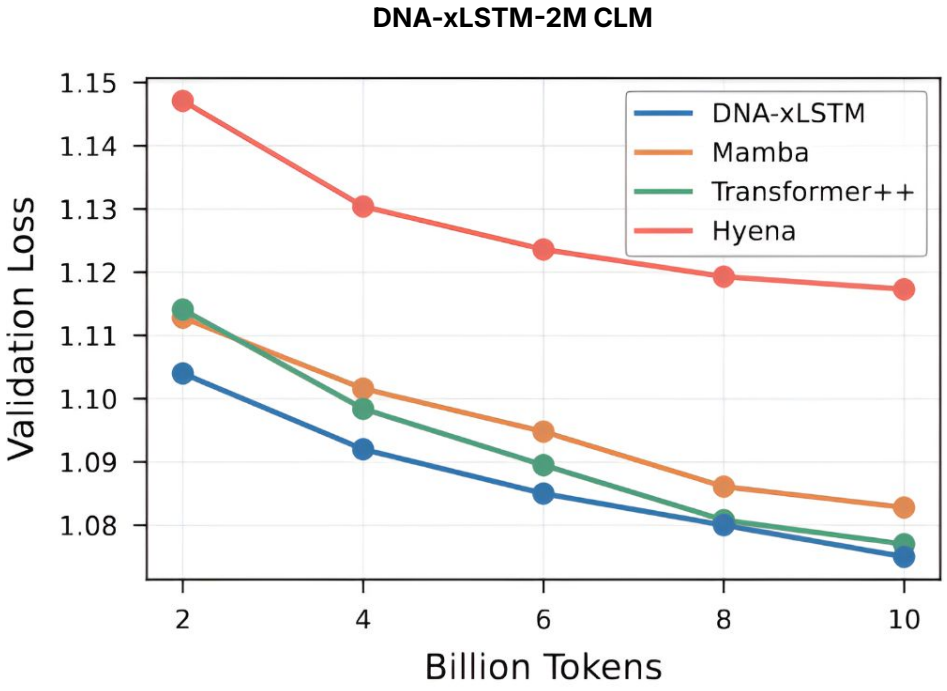


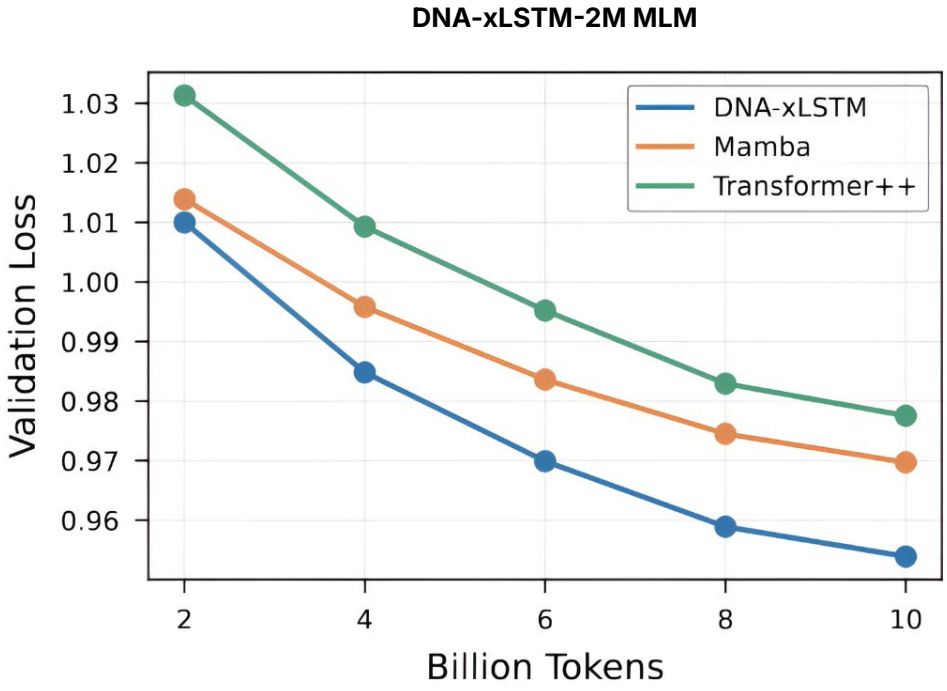
[Reverse Complement Equivariance, Mallet et.al. (2021)]
[Caduceus, Schiff et. al. (2024)]

DNA-xLSTM: Short Context

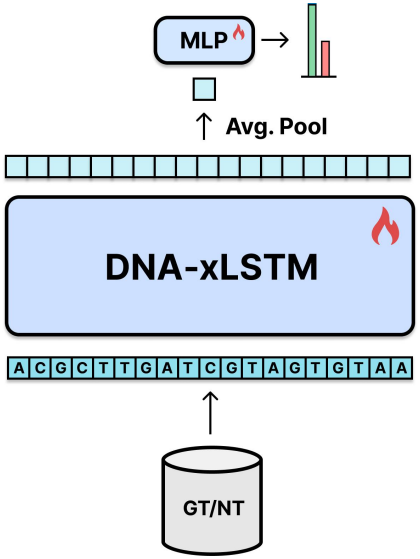


DNA-xLSTM: Short Context





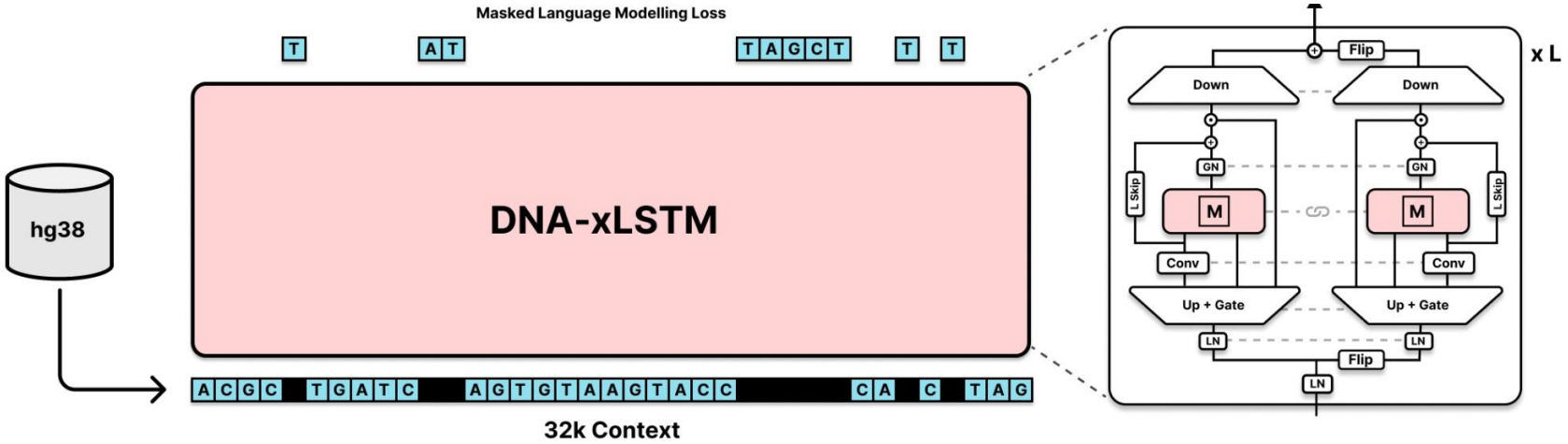
DNA-xLSTM: Downstream Adaptation



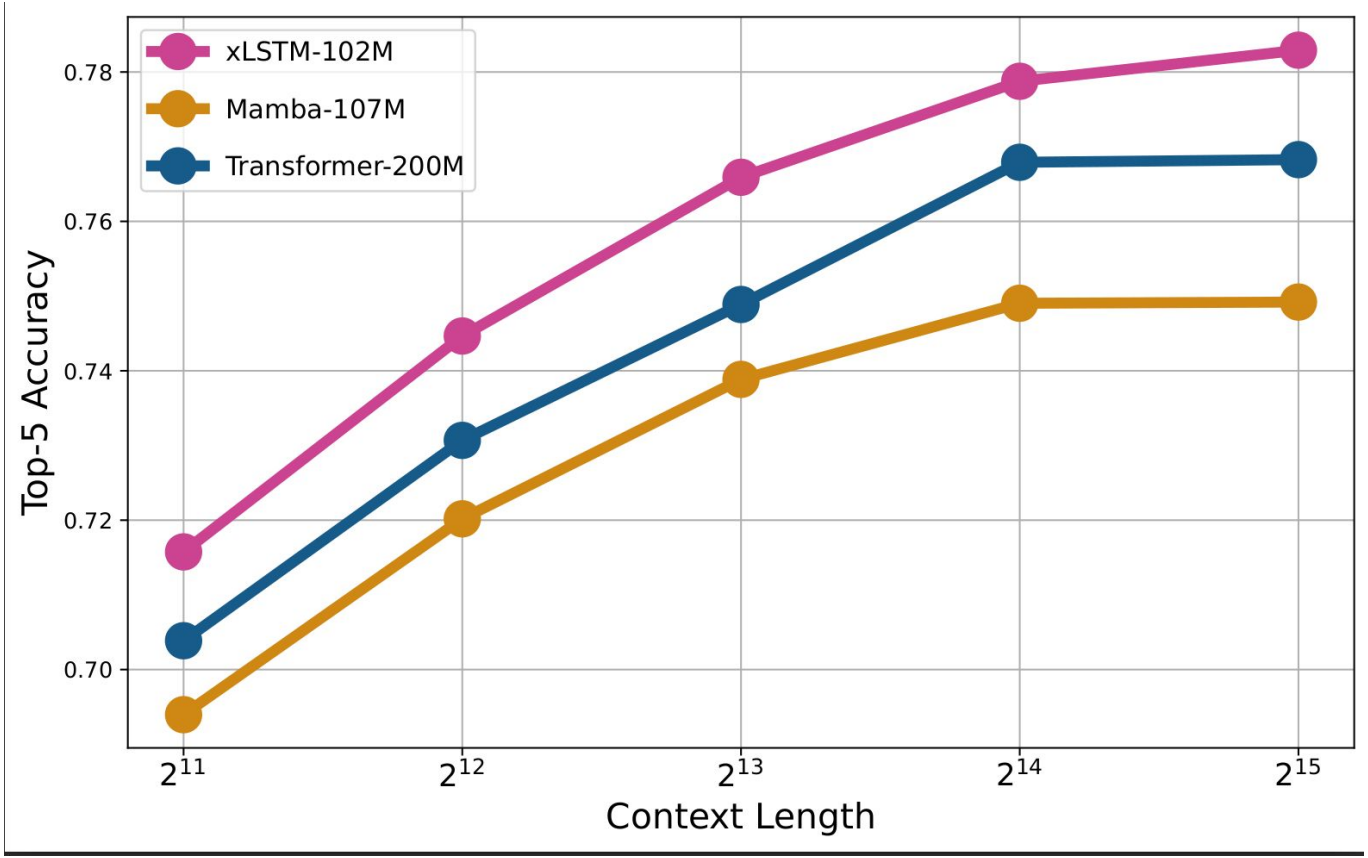
Task	Metric	> 100M Param. Models			2M Param. Models				
		Enformer (252M)	DNABERT-2 (117M)	NT-v2 (500M)	HyenaDNA	Mamba-PS	Mamba-PH	xLSTM-PS	xLSTM-PH
Histone Markers									
H3	MCC ↑	$0.719^{\pm 0.048}$	$0.785^{\pm 0.033}$	$0.784^{\pm 0.047}$	$0.779^{\pm 0.037}$	$0.799^{\pm 0.029}$	$0.815^{\pm 0.048}$	$0.796^{\pm 0.014}$	$0.824^{\pm 0.010}$
H3K14AC	MCC ↑	$0.288^{\pm 0.077}$	$0.516^{\pm 0.028}$	$0.551^{\pm 0.021}$	$0.612^{\pm 0.065}$	$0.541^{\pm 0.212}$	$0.631^{\pm 0.026}$	$0.570^{\pm 0.003}$	$0.598^{\pm 0.017}$
H3K36ME3	MCC ↑	$0.344^{\pm 0.055}$	$0.591^{\pm 0.020}$	$0.625^{\pm 0.030}$	$0.613^{\pm 0.041}$	$0.609^{\pm 0.109}$	$0.601^{\pm 0.129}$	$0.588^{\pm 0.019}$	$0.625^{\pm 0.010}$
H3K4ME1	MCC ↑	$0.291^{\pm 0.061}$	$0.511^{\pm 0.028}$	$0.550^{\pm 0.021}$	$0.512^{\pm 0.024}$	$0.488^{\pm 0.102}$	$0.523^{\pm 0.039}$	$0.490^{\pm 0.012}$	$0.526^{\pm 0.001}$
H3K4ME2	MCC ↑	$0.211^{\pm 0.069}$	$0.336^{\pm 0.040}$	$0.319^{\pm 0.045}$	$0.455^{\pm 0.095}$	$0.388^{\pm 0.101}$	$0.487^{\pm 0.170}$	$0.489^{\pm 0.024}$	$0.504^{\pm 0.012}$
H3K4ME3	MCC ↑	$0.158^{\pm 0.072}$	$0.352^{\pm 0.077}$	$0.410^{\pm 0.033}$	$0.549^{\pm 0.056}$	$0.440^{\pm 0.202}$	$0.544^{\pm 0.045}$	$0.520^{\pm 0.019}$	$0.537^{\pm 0.012}$
H3K79ME3	MCC ↑	$0.496^{\pm 0.042}$	$0.613^{\pm 0.030}$	$0.626^{\pm 0.046}$	$0.672^{\pm 0.048}$	$0.676^{\pm 0.026}$	$0.697^{\pm 0.077}$	$0.662^{\pm 0.011}$	$0.697^{\pm 0.007}$
H3K9AC	MCC ↑	$0.420^{\pm 0.063}$	$0.542^{\pm 0.029}$	$0.562^{\pm 0.040}$	$0.581^{\pm 0.061}$	$0.604^{\pm 0.048}$	$0.622^{\pm 0.030}$	$0.622^{\pm 0.013}$	$0.627^{\pm 0.008}$
H4	MCC ↑	$0.732^{\pm 0.076}$	$0.796^{\pm 0.027}$	$0.799^{\pm 0.025}$	$0.763^{\pm 0.044}$	$0.789^{\pm 0.020}$	$0.811^{\pm 0.022}$	$0.793^{\pm 0.011}$	$0.813^{\pm 0.008}$
H4AC	MCC ↑	$0.273^{\pm 0.063}$	$0.463^{\pm 0.041}$	$0.495^{\pm 0.032}$	$0.564^{\pm 0.038}$	$0.525^{\pm 0.240}$	$0.621^{\pm 0.054}$	$0.558^{\pm 0.013}$	$0.583^{\pm 0.014}$
Regulatory Annotation									
Enhancer	MCC ↑	$0.451^{\pm 0.108}$	$0.516^{\pm 0.098}$	$0.548^{\pm 0.144}$	$0.517^{\pm 0.117}$	$0.491^{\pm 0.066}$	$0.546^{\pm 0.073}$	$0.375^{\pm 0.030}$	$0.545^{\pm 0.024}$
Enhancer Types	MCC ↑	$0.309^{\pm 0.134}$	$0.423^{\pm 0.051}$	$0.424^{\pm 0.132}$	$0.386^{\pm 0.185}$	$0.416^{\pm 0.095}$	$0.439^{\pm 0.054}$	$0.444^{\pm 0.043}$	$0.466^{\pm 0.011}$
Promoter: All	F1 ↑	$0.954^{\pm 0.006}$	$0.971^{\pm 0.006}$	$0.976^{\pm 0.006}$	$0.960^{\pm 0.005}$	$0.967^{\pm 0.004}$	$0.970^{\pm 0.004}$	$0.962^{\pm 0.002}$	$0.967^{\pm 0.001}$
NonTATA	F1 ↑	$0.955^{\pm 0.010}$	$0.972^{\pm 0.005}$	$0.976^{\pm 0.006}$	$0.959^{\pm 0.011}$	$0.968^{\pm 0.006}$	$0.968^{\pm 0.010}$	$0.963^{\pm 0.002}$	$0.970^{\pm 0.001}$
TATA	F1 ↑	$0.960^{\pm 0.023}$	$0.955^{\pm 0.021}$	$0.966^{\pm 0.013}$	$0.944^{\pm 0.040}$	$0.957^{\pm 0.015}$	$0.953^{\pm 0.016}$	$0.948^{\pm 0.006}$	$0.952^{\pm 0.005}$
Splice Site Annotation									
All	Accuracy ↑	$0.848^{\pm 0.019}$	$0.939^{\pm 0.009}$	$0.983^{\pm 0.008}$	$0.956^{\pm 0.011}$	$0.927^{\pm 0.021}$	$0.940^{\pm 0.027}$	$0.965^{\pm 0.003}$	$0.974^{\pm 0.004}$
Acceptor	F1 ↑	$0.914^{\pm 0.028}$	$0.975^{\pm 0.006}$	$0.981^{\pm 0.011}$	$0.958^{\pm 0.010}$	$0.936^{\pm 0.077}$	$0.937^{\pm 0.033}$	$0.970^{\pm 0.005}$	$0.953^{\pm 0.008}$
Donor	F1 ↑	$0.906^{\pm 0.027}$	$0.963^{\pm 0.006}$	$0.985^{\pm 0.022}$	$0.949^{\pm 0.024}$	$0.948^{\pm 0.025}$	$0.874^{\pm 0.289}$	$0.962^{\pm 0.001}$	$0.951^{\pm 0.005}$

Nucleotide Transformer, Dalla-Torre et.al. (2023)

DNA-xLSTM: Long Context

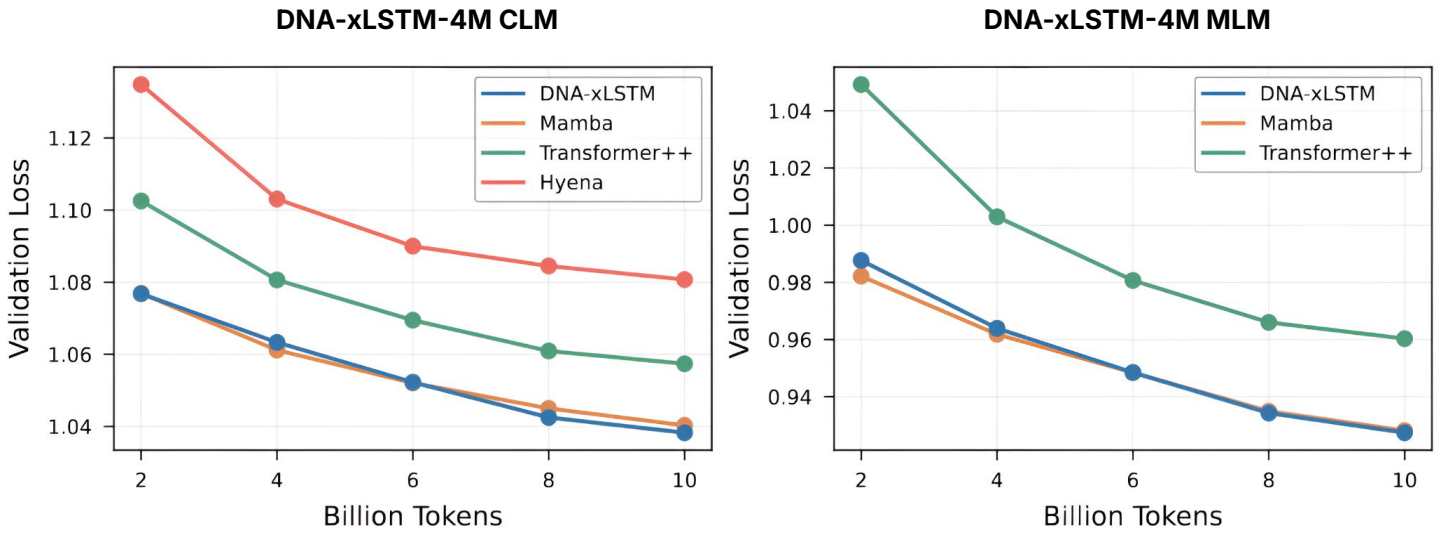


Bio-xLSTM



Schmidinger, N., Schneckenreiter, L., Seidl, P., Schimunek, J., Hoedt, P. J., Brandstetter, J., ... & Klambauer, G. (2025). Bio-xLSTM: Generative modeling, representation and in-context learning of biological and chemical sequences. *International Conference on Learning Representations*.

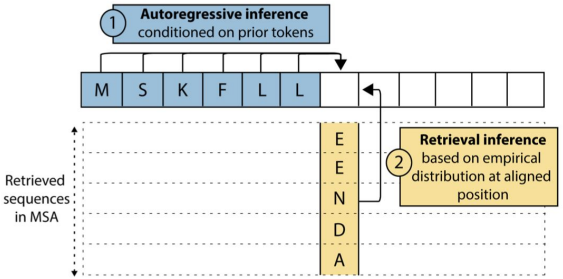
DNA-xLSTM: Long Context



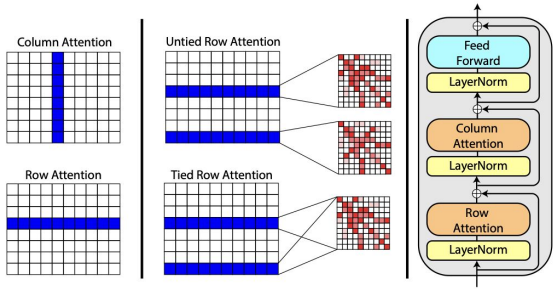
Prot-xLSTM

Winning Strategy: Incorporate Evolutionary Information

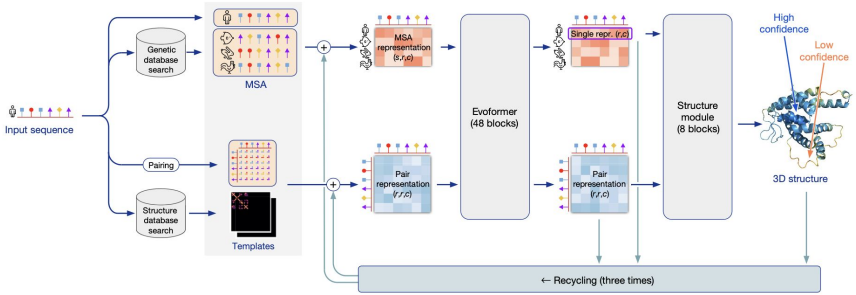
Transception



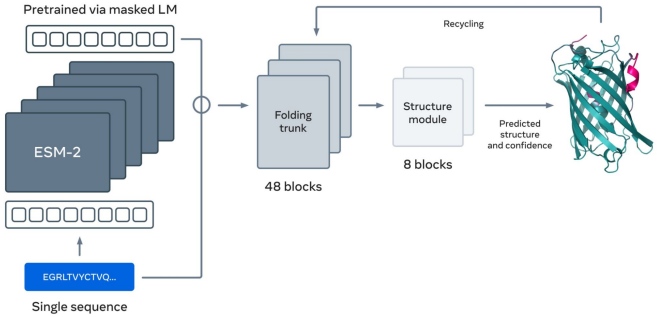
MSA Transformer



AlphaFold



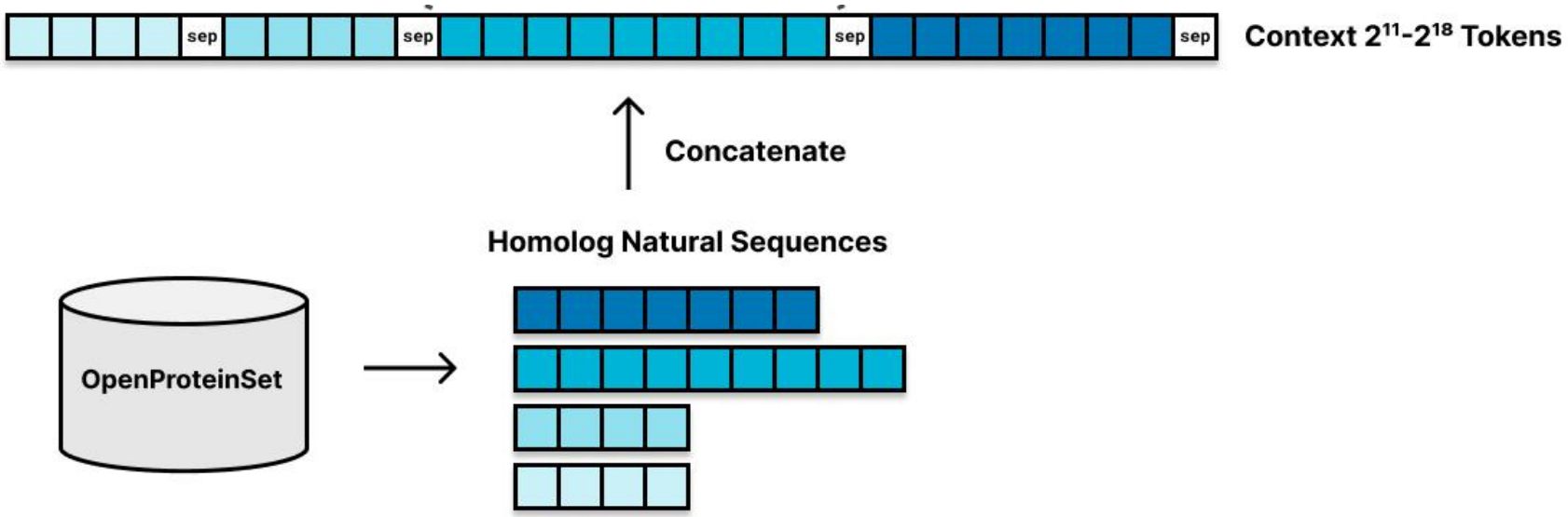
ESMFold



AlphaFold 2, Jumper et.al. (2021)
ESMFold, Lin et.al. (2022)

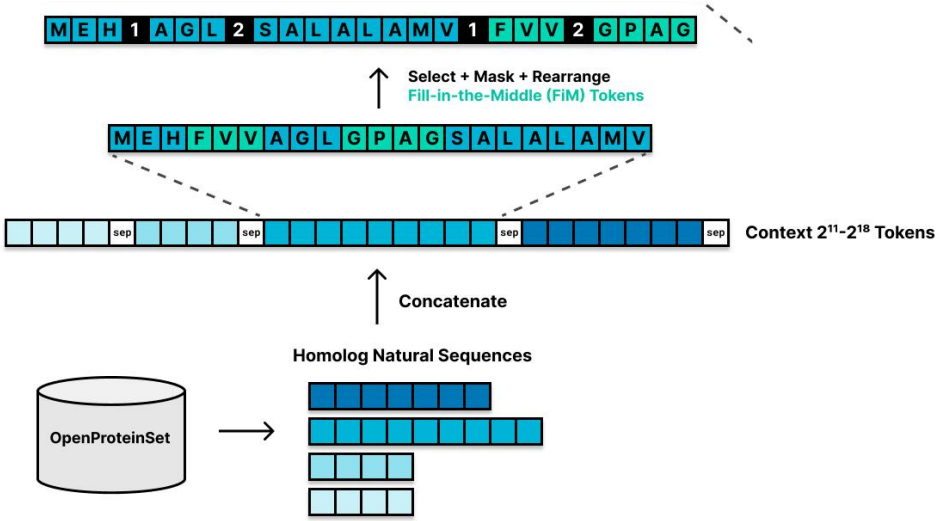
Transception, Notin et.al. (2022)
MSA Transformer, Rao et.al.(2021)

Homology-Aware Modeling via Conditioning



Poet, Truong et. al. (2023)
ProtMamba, Sgarbossa et.al. (2024)

Fill-in-the-Middle (FiM) Augmentation



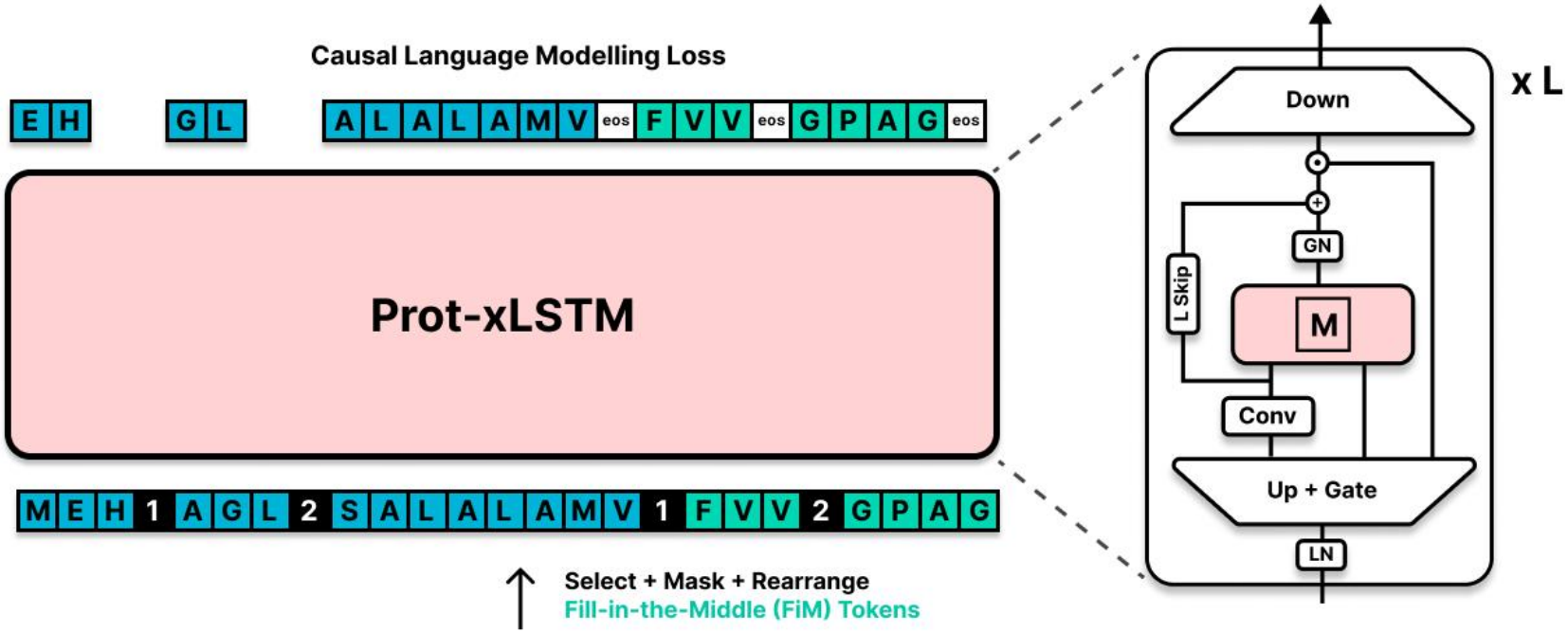
FiM Illustration

The quick brown fox jumps over the lazy dog.

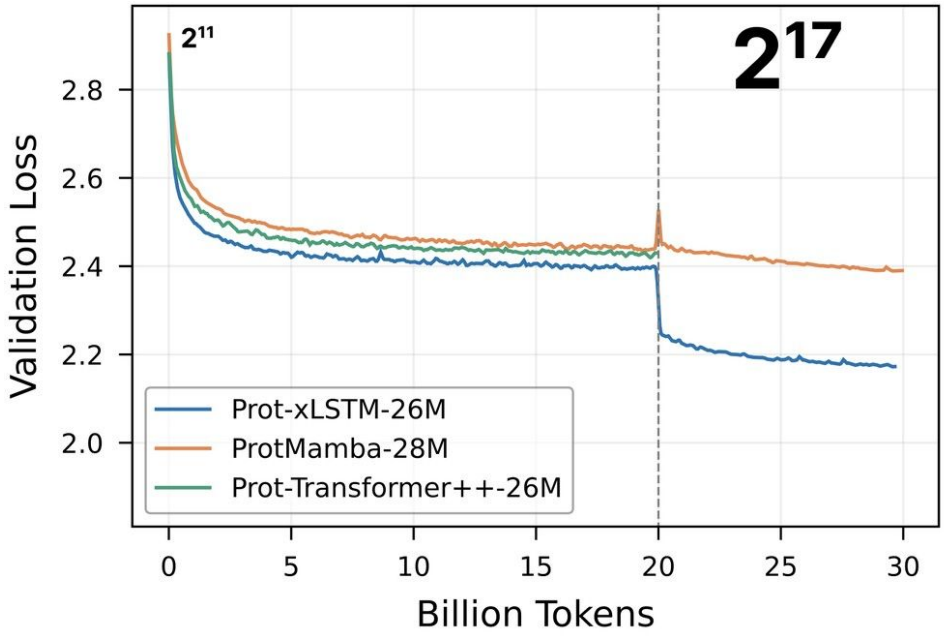
The quick brown fox <mask> the lazy dog. <mask> jumps over

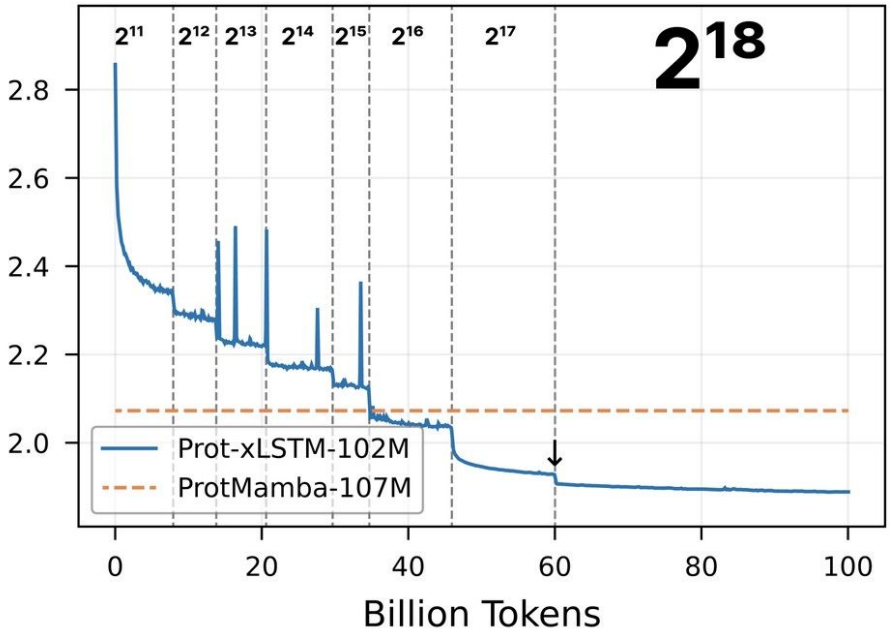
Fill-in-the-Middle, Bavarian et.al. (2022)
ProtMamba, Sgarbossa et.al. (2024)

Prot-xLSTM Pretraining

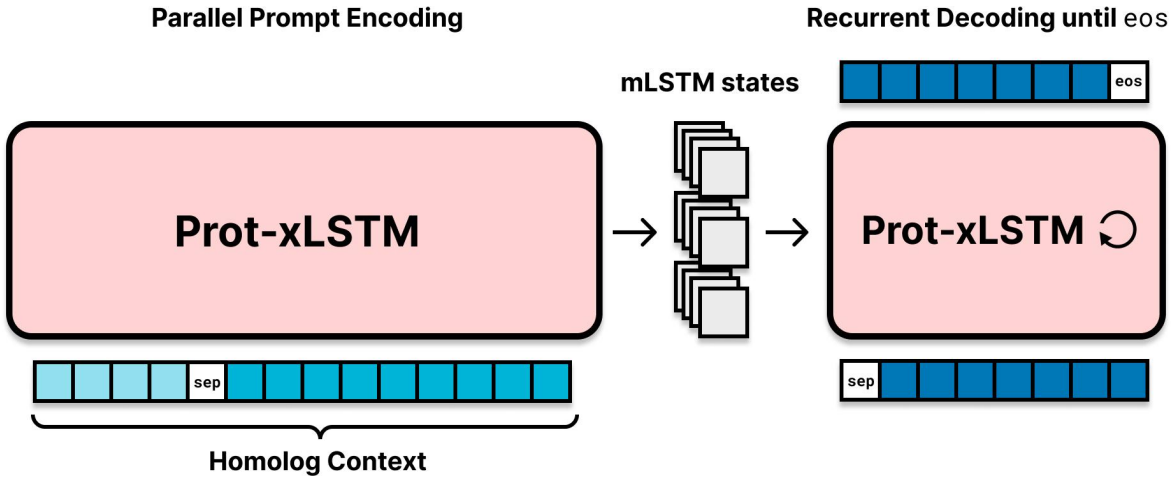


Homology-aware Pre-Training





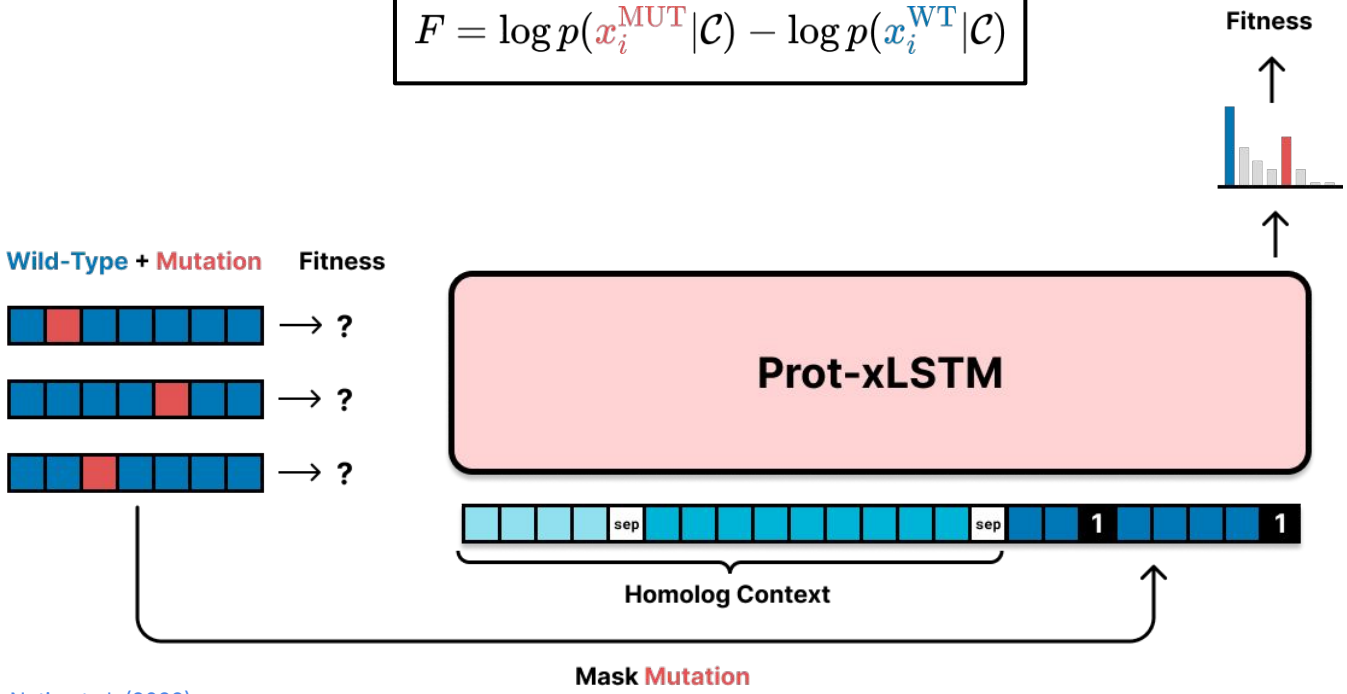
Homology-aware Generation



	Prot-xLSTM-26M	ProtMamba-28M	Prot-xLSTM-102M	ProtMamba-107M
Sequence Length	$0.41^{\pm 0.09}$	$0.52^{\pm 0.09}$	$0.40^{\pm 0.08}$	$0.36^{\pm 0.08}$
Min. Hamming	$0.43^{\pm 0.08}$	$0.60^{\pm 0.11}$	$0.47^{\pm 0.09}$	$0.42^{\pm 0.07}$
HMMER	$0.57^{\pm 0.10}$	$0.54^{\pm 0.11}$	$0.44^{\pm 0.09}$	$0.49^{\pm 0.10}$
pLDDT	$0.40^{\pm 0.09}$	$0.68^{\pm 0.12}$	$0.27^{\pm 0.05}$	$0.30^{\pm 0.07}$
pTM	$0.38^{\pm 0.08}$	$0.72^{\pm 0.10}$	$0.26^{\pm 0.05}$	$0.28^{\pm 0.05}$

Zero-Shot Fitness Prediction

$$F = \log p(x_i^{\mathrm{MUT}} | \mathcal{C}) - \log p(x_i^{\mathrm{WT}} | \mathcal{C})$$



Transception, Notin et.al. (2022)
ProtMamba, Sgarbossa et.al. (2024)

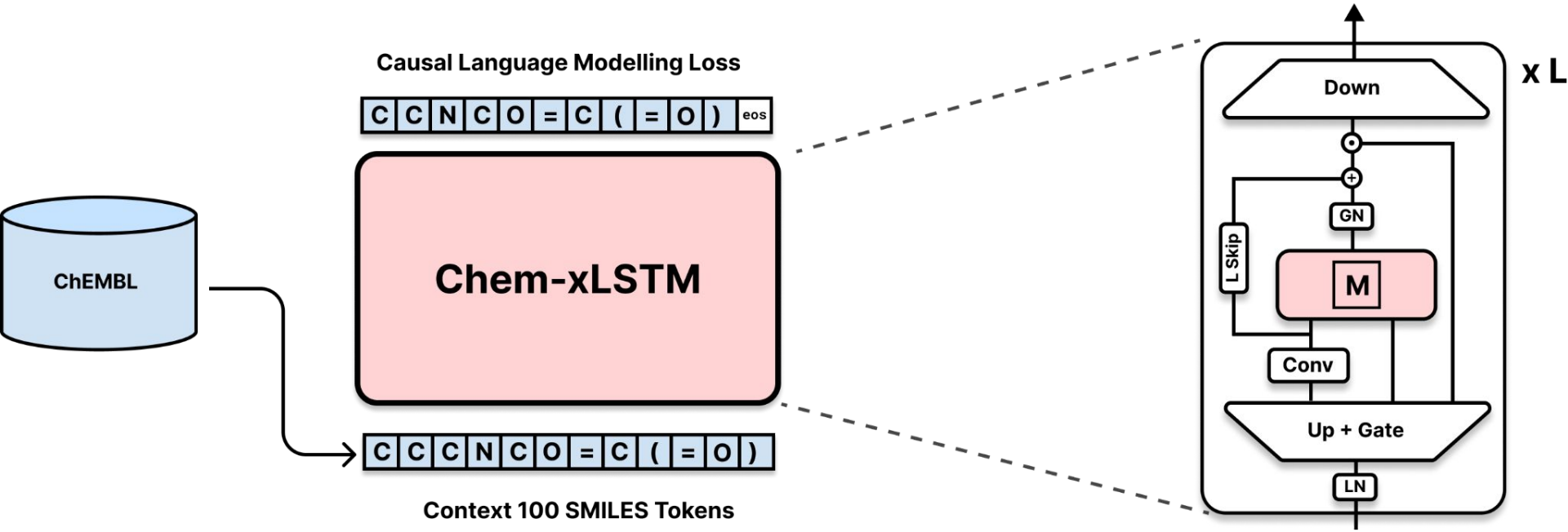
Zero-Shot Fitness Prediction Evaluation

Model Type	Model	Reference	#Params	Spearman ρ
Alignment-based	Site-Independant	Hopf et al. (2017)	-	0.359
	EVE	Frazer et al. (2021)	-[a]	0.432
	GEMME	Laine et al. (2019)	-	0.455
Protein language model (PLM)	Tranception L (w/o R)	Notin et al. (2022a)	700M	0.374
	VespaG	Marquet et al. (2024)	3B	0.458
	ProGen2 XL	Nijkamp et al. (2023)	6B	0.391
	ESM-2	Lin et al. (2023)	15B	0.401
Alignment + PLM	MSA-Transformer	Rao et al. (2021)	100M	0.432
	Tranception L (w/ R)	Notin et al. (2022a)	700M	0.434
	TranceptEVE L	Notin et al. (2022b)	>700M[a]	0.456
Homology-aware PLM	Prot-xLSTM	Ours	26M	0.411[b]
	ProtMamba	Sgarbossa et al. (2024)	28M	0.360[b]
	Prot-xLSTM	Ours	102M	0.416[b]
	ProtMamba	Sgarbossa et al. (2024)	107M	0.415[b]
	PoET	Truong Jr and Bepler (2023)	201M	0.470
Inverse folding	ESM-IF1	Hsu et al. (2022)	142M	0.422
Structure + PLM	SaProt	Su et al. (2024b)	35M	0.407
	ProSST	Li et al. (2024)	110M	0.507
	SaProt	Su et al. (2024b)	650M	0.457

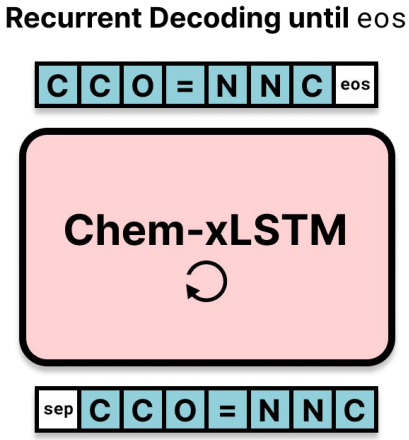
[a] EVE parameters depend on the size of a given MSA.
[b] This work. All other values are retrieved from ProteinGym on 03/11/2024.

Chem-xLSTM

Unconditional Pre-Training



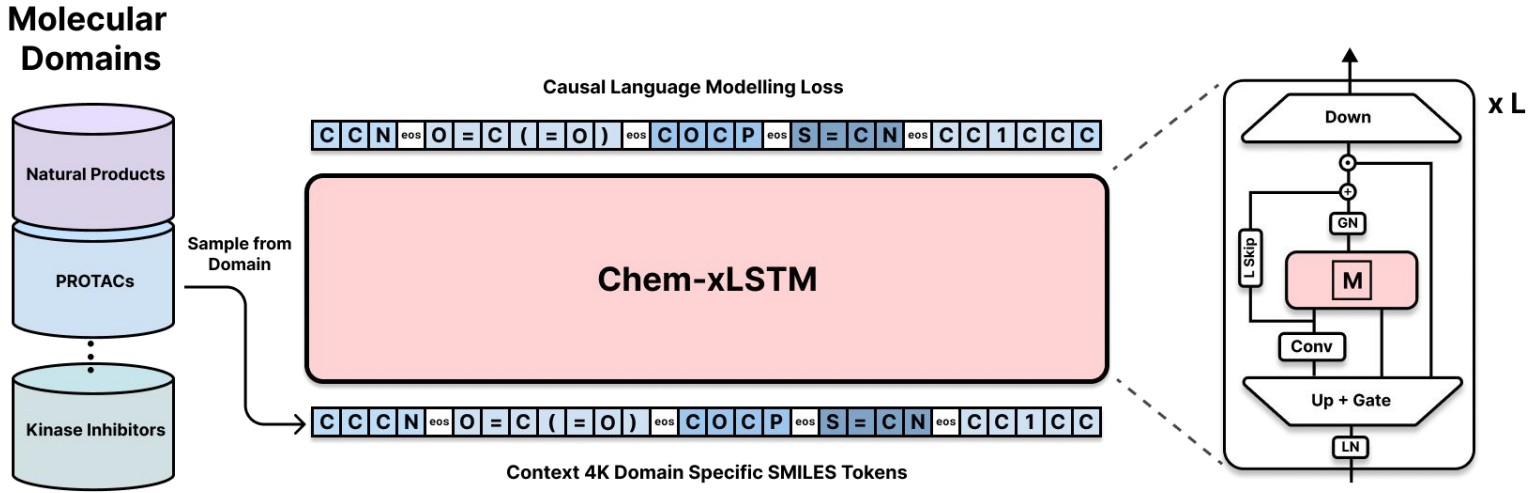
Unconditional Generation Evaluation



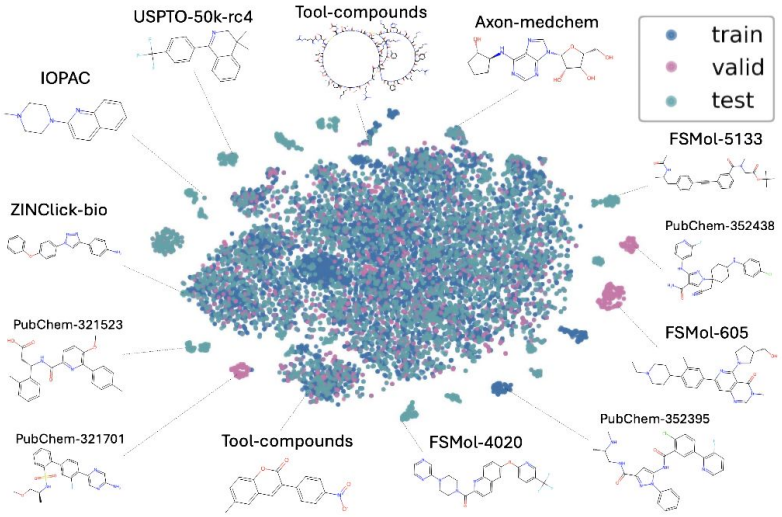
	SMILES-LSTM[a]	SMILES-GPT[b]	SMILES-S4[c]	Chem-Mamba[d]	Chem-xLSTM
FCD ↓	$0.46^{\pm<0.01}$	$0.15^{\pm<0.01}$	$0.28^{\pm<0.01}$	$0.21^{\pm<0.01}$	$0.13^{\pm<0.01}$
Perplexity ↓	$1.88^{\pm3.8}$	$1.65^{\pm0.6}$	$1.73^{\pm2.4}$	$1.74^{\pm0.5}$	$1.68^{\pm1.0}$

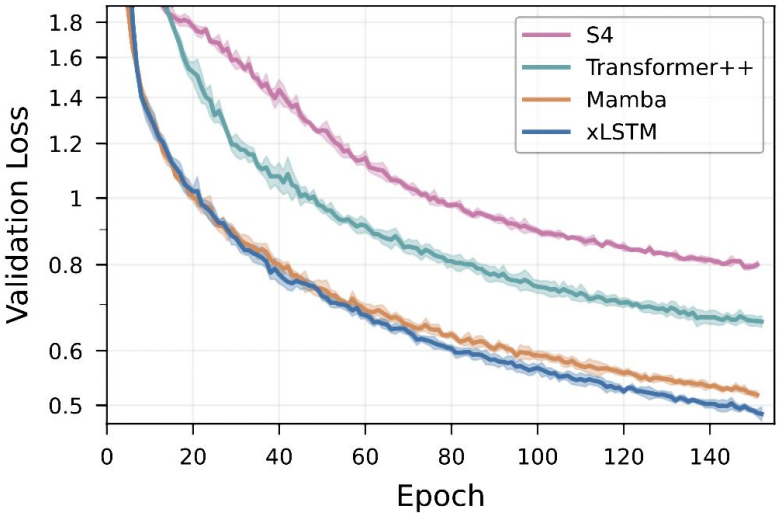
[a] Segler et al. (2018) [b] Adilov (2021) [c] Özçelik et al. (2024) [d] Adapted to SMILES in this work.

Conditional Generation



In-context learning





Paper

DNA-xLSTM

Prot-xLSTM

Chem-xLSTM

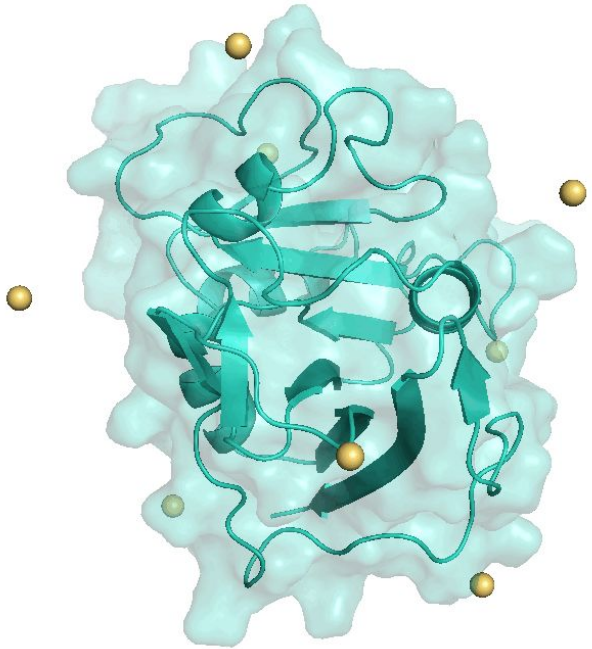
Huggingface App here!

JKU Institut für Machine Learning

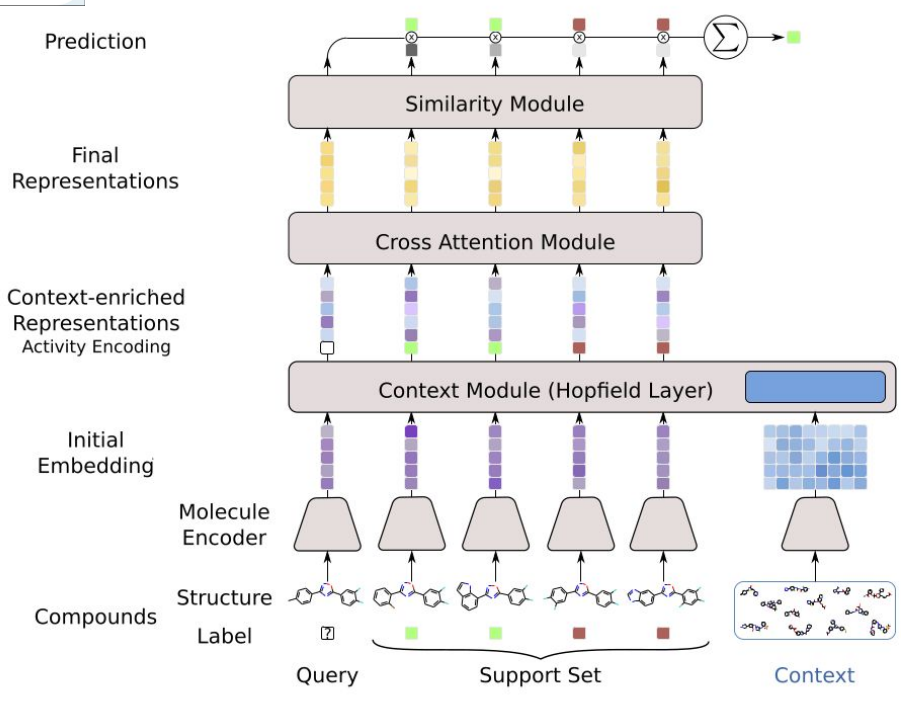
ellis

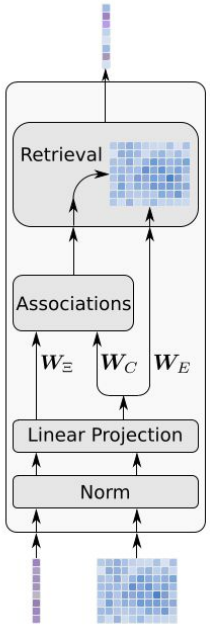
Resources (Bio)

- **Bio-xLSTM: LLM models for DNA, proteins and small molecules**
  - Paper (ICLR 2025): https://openreview.net/forum?id=IjbXZdugdj
- **VN-EGNN: binding pocket identification method**
  - Paper (pre-print): https://arxiv.org/abs/2404.07194
  - Github: https://github.com/ml-jku/vnegnn
  - HuggingFace App: https://huggingface.co/spaces/ml-jku/vnegnn
- **CLOOME: most powerful features for microscopy images ("cell painting")**
  - Paper (NComms): https://www.nature.com/articles/s41467-023-42328-w
  - Github: https://github.com/ml-jku/cloome
  - HuggingFace App: https://huggingface.co/spaces/anasanchezf/cloome
- **GNN-VPA: variance-preserving aggregation for message-passing networks**
  - Paper (ICLR 2024): https://arxiv.org/abs/2403.04747
  - Implementation: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.aggr.VariancePreservingAggregation.html
- **CLAMP: multi-modal bioactivity prediction model with in-context capacity**
  - Paper (ICML 2023): https://arxiv.org/abs/2303.03363
  - Github: https://github.com/ml-jku/clamp
- **MHNfs: best few-shot learning method via in-context learning**
  - Paper (ICLR 2023): https://openreview.net/pdf?id=XrMWUuEevr
  - Github: https://github.com/ml-jku/MHNfs
  - HuggingFace App: https://huggingface.co/spaces/ml-jku/mhnfs
- **LAM-Slide: highly efficient and accurate molecular dynamics**
  - Paper (pre-print): https://arxiv.org/abs/2502.12128
  - Blog: https://ml-jku.github.io/LaM-SLidE/

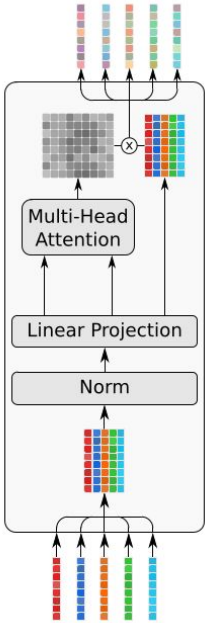


In-context learning for bioactivity





Context Module



Cross Attention Module

Schimunek, J., Seidl, P., Friedrich, L., Kuhn, D., Rippmann, F., Hochreiter, S., & Klambauer, G. (2023). Context-enriched molecule representations improve few-shot drug discovery. *International Conference on Learning Representations*. https://huggingface.co/spaces/ml-jku/mhnfs

Conclusion

Conclusion

- **Domain-Specific Adaptations**: Introduced xLSTM variants tailored to NLP, computer vision, and major biological domains—**DNA-xLSTM**, **Prot-xLSTM**, and **Chem-xLSTM**

- **State-of-the-Art Performance**: Demonstrated that xLSTM consistently matches or surpasses strong baselines across all three domains.

- **Scalability & Efficiency**: Validated xLSTM's ability to process sequences up to 260,000 tokens while maintaining superior training speed and inference efficiency compared to Transformers.

- **Foundation Model Potential**: Positioned xLSTM as a leading candidate for foundational models in biology, bridging performance, scalability, and domain adaptability.

ACKNOWLEDGEMENTS
LIT AI Lab & ELLIS unit Linz
Johannes Kepler Universität Linz

Protein design and optimization (PDO)

AIDD

Project funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956832, and it is Horizon 2020 Marie Skłodowska-Curie Innovative Training Network - European Industrial Doctorate.

Deep Learning for drug discovery & design

FWF Austrian Science Fund

Bilateral Artificial Intelligence

ellis European Laboratory for Learning and Intelligent Systems

We are hiring!

Protein design and optimization (PDO)

Deep Learning for drug discovery & design

AIDD

Project funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956832, and it is Horizon 2020 Marie Skłodowska-Curie Innovative Training Network - European Industrial Doctorate.

FWF Austrian Science Fund

Bilateral Artificial Intelligence

ellis European Laboratory for Learning and Intelligent Systems

BACKUP SLIDES

Essential References

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, *91*(1), 31.

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

Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., ... & Hochreiter, S. (2024). xLSTM: Extended Long Short-Term Memory. *Advances in neural information processing systems, 37.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

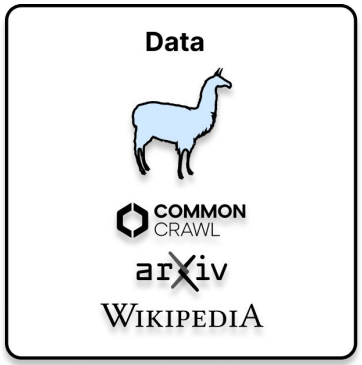
Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730-27744.

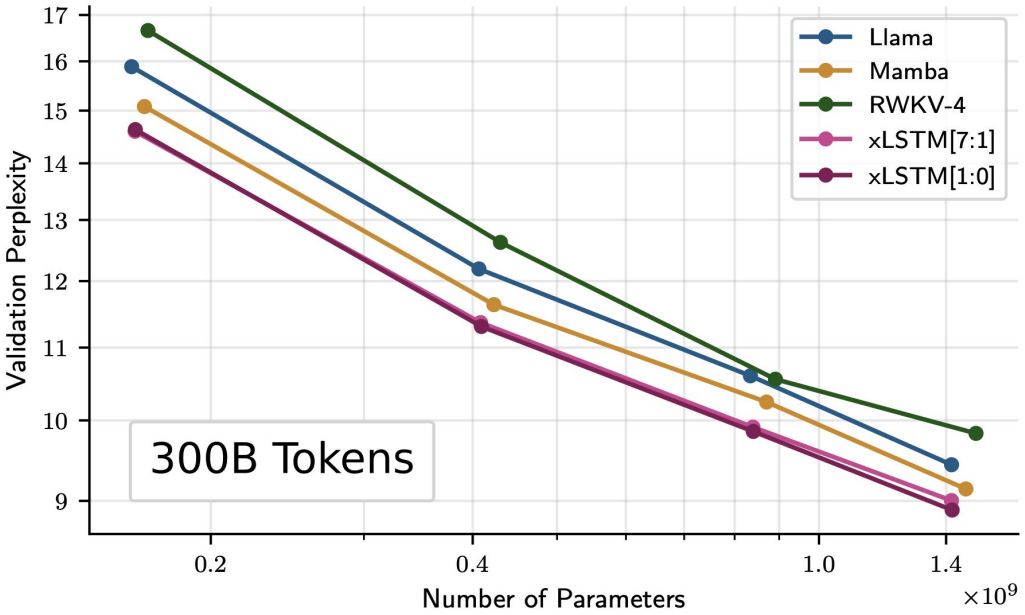
Yang, S., Wang, B., Shen, Y., Panda, R., & Kim, Y. (2023). Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*.

Alkin, B., Beck, M., Pöppel, K., Hochreiter, S., & Brandstetter, J. (2025). Vision-LSTM: xLSTM as Generic Vision Backbone. *International Conference on Learning Representations.*

Schmidinger, N., Schneckenreiter, L., Seidl, P., Schimunek, J., Hoedt, P. J., Brandstetter, J., ... & Klambauer, G. (2025). Bio-xLSTM: Generative modeling, representation and in-context learning of biological and chemical sequences. *International Conference on Learning Representations.*

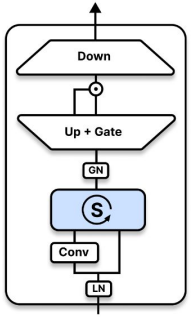
xLSTM Scaling





xLSTM, Beck et.al. (2024)

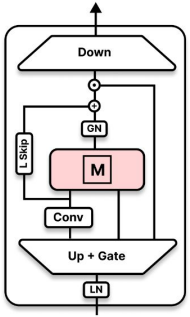
xLSTM Background



sLSTM (S)

$$
\begin{aligned}
c_t &= \sigma(\tilde{f}_t)\, c_{t-1} + \exp(\tilde{i}_t)\, \tanh(\tilde{z}_t) \\
n_t &= \sigma(\tilde{f}_t)\, n_{t-1} + \exp(\tilde{i}_t) \\
h_t &= \sigma(\tilde{o}_t)\, \frac{c_t}{n_t}
\end{aligned}
$$

+ Enhanced State Tracking



mLSTM M

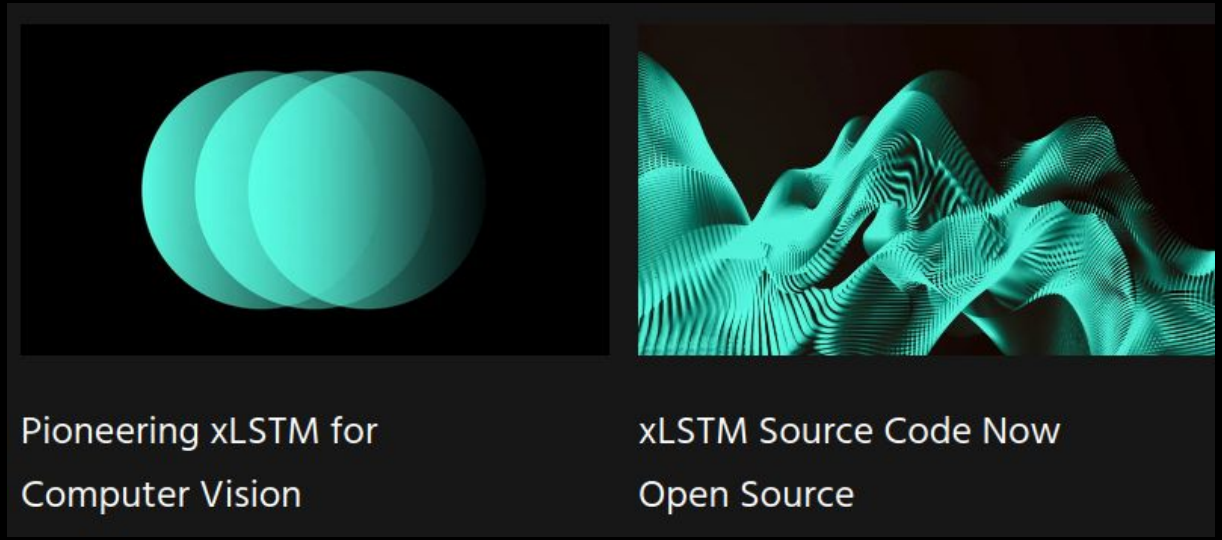
$$
\begin{aligned}
\boldsymbol{C}_t &= \sigma(\tilde{f}_t)\, \boldsymbol{C}_{t-1} + \exp(\tilde{i}_t)\, \boldsymbol{v}_t \boldsymbol{k}_t^\top \\
\boldsymbol{n}_t &= \sigma(\tilde{f}_t)\, \boldsymbol{n}_{t-1} + \exp(\tilde{i}_t)\, \boldsymbol{k}_t \\
\boldsymbol{h}_t &= \sigma(\tilde{o}_t) \odot \frac{\boldsymbol{C}_t \boldsymbol{q}_t}{\max\{|\boldsymbol{n}_t^\top \boldsymbol{q}_t|, 1\}}
\end{aligned}
$$

+ Parallel & Recurrent Mode

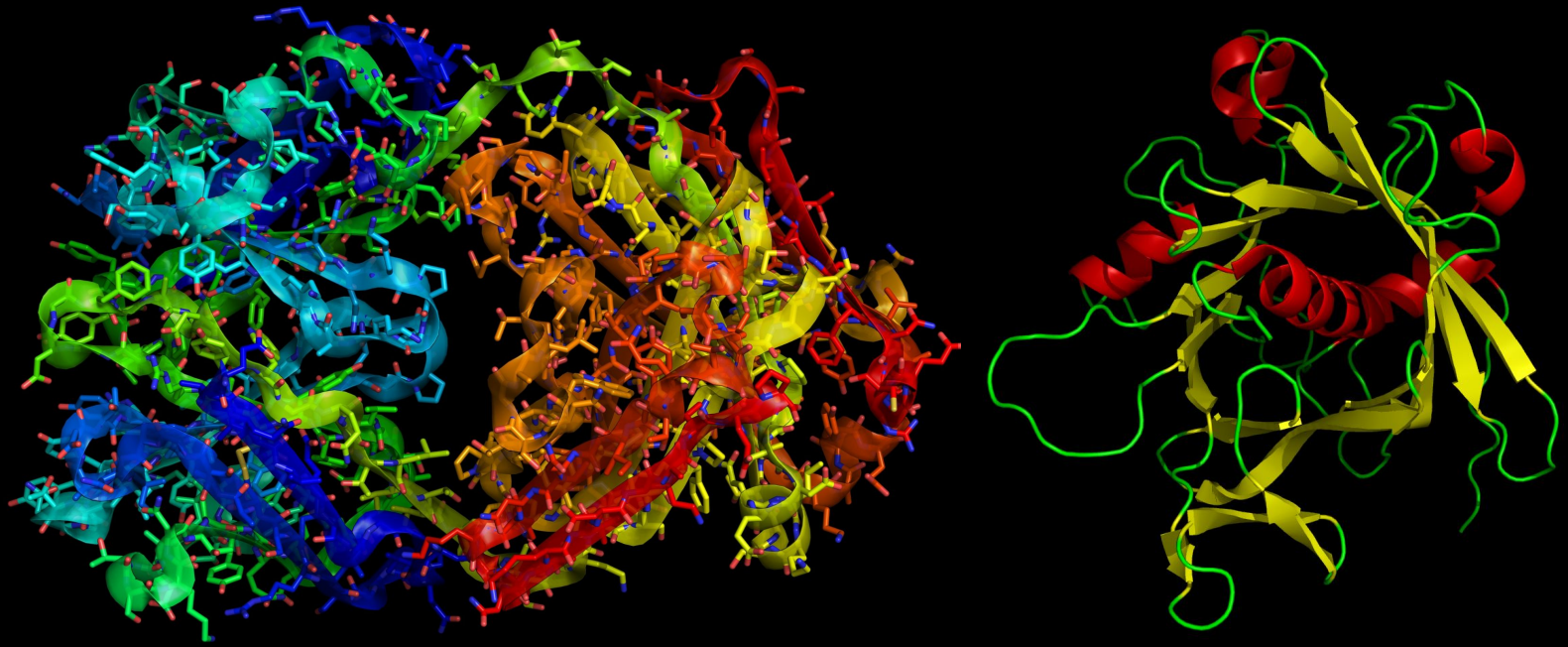
[Linear Transformers Are Secretly Fast Weight Programmers]
[Gated Linear Attention Transformers with Hardware-Efficient Training]
[Mamba: Linear-Time Sequence Modeling with Selective State Spaces]

NXAI

- Company that sponsored the development of xLSTM
- NXAI scales up xLSTM and builds larger LLMs
- “AI at scale”, AI for simulations
- Austrian startup with center in Linz:
- https://www.nx-ai.com/



AlphaFold



AI breakthrough in biology

Future Directions

Scaling: train larger versions of Bio-xLSTM and evaluate scaling.

Multi-Modality: Incorporate diverse data types, including 3D structures and other non-sequential modalities.

Interpretability: Identify factors contributing to superior performance compared to Transformers and Mamba in specific contexts. Analyze scenarios where xLSTM underperforms.

xLSTM stabilization

$$m_t = \max\left(\log(\mathrm{f}_t) + m_{t-1}, \log(\mathrm{i}_t)\right) \qquad \text{stabilizer state}$$

$$\mathrm{i}'_t = \exp\left(\log(\mathrm{i}_t) - m_t\right) = \exp\left(\tilde{\mathrm{i}}_t - m_t\right) \qquad \text{stabil. input gate}$$

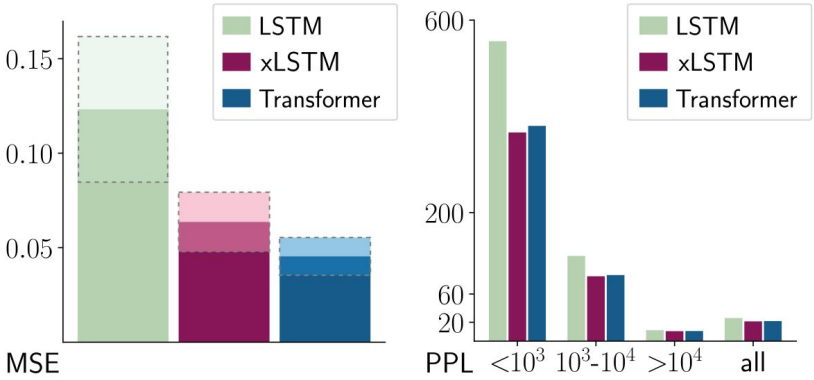
$$\mathrm{f}'_t = \exp\left(\log(\mathrm{f}_t) + m_{t-1} - m_t\right) \qquad \text{stabil. forget gate}$$

Numerical stability of the exponential function

Can be seen as running-softmax

LSTM limitations

- **Nearest Neighbor Search:**
  - given a reference vector
  - scan a sequence sequentially
  - for the most similar vector
  - return its attached value at sequence end
- **Difficult for models:**
  - whenever new most similar → revise value and largest similarity

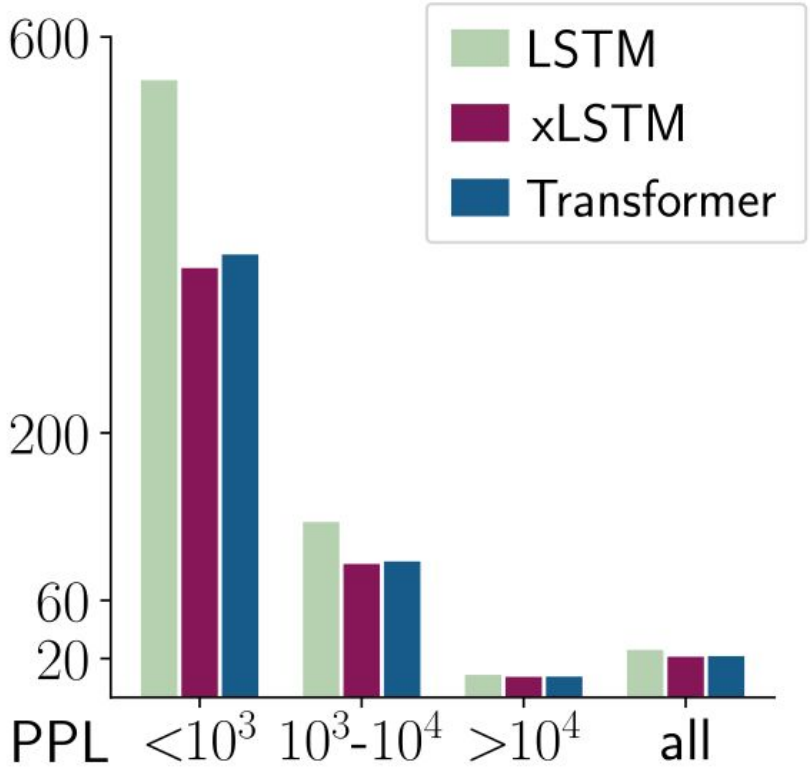


LSTM limitations

- Inability to revise storage decisions 
  - Nearest Neighbor Search
- Limited storage capacities
  - Rare Token Prediction
  - problem with new or rare tokens
- No parallelization for training

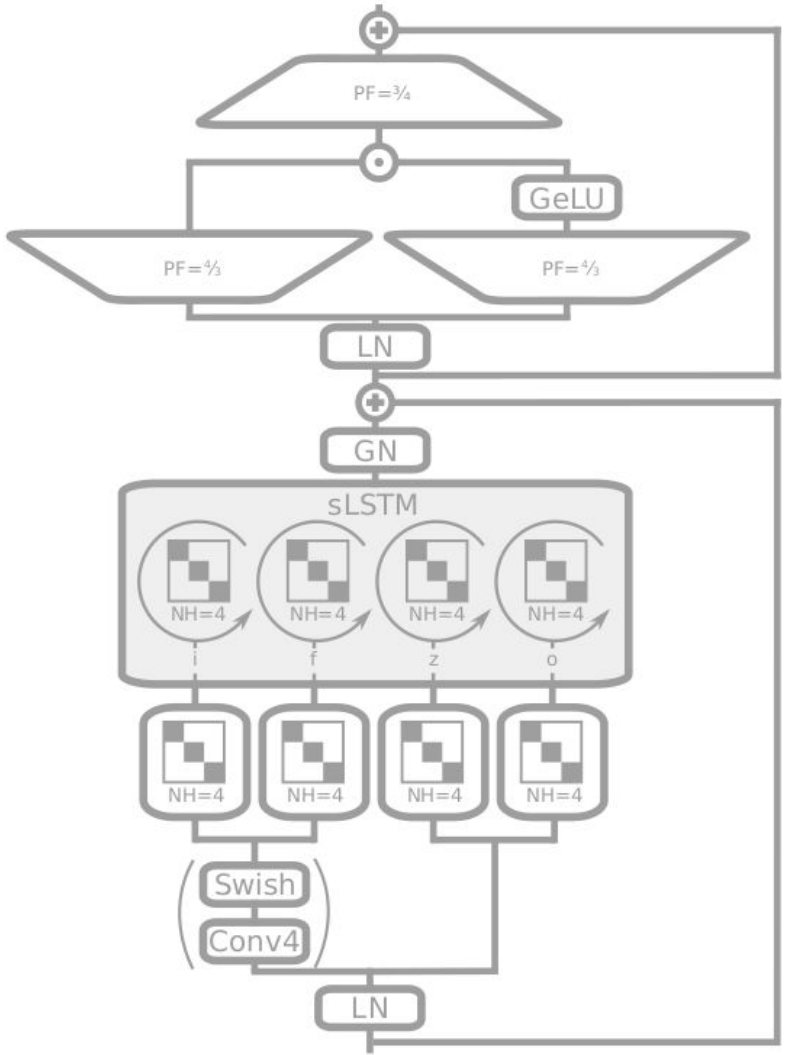
LSTM limitations

- **Rare Token Prediction:**
  - perplexity (PPL) of token prediction on Wikitext 103
  - partitions of token according to frequency
- rare tokens must be memorized since they are not learned



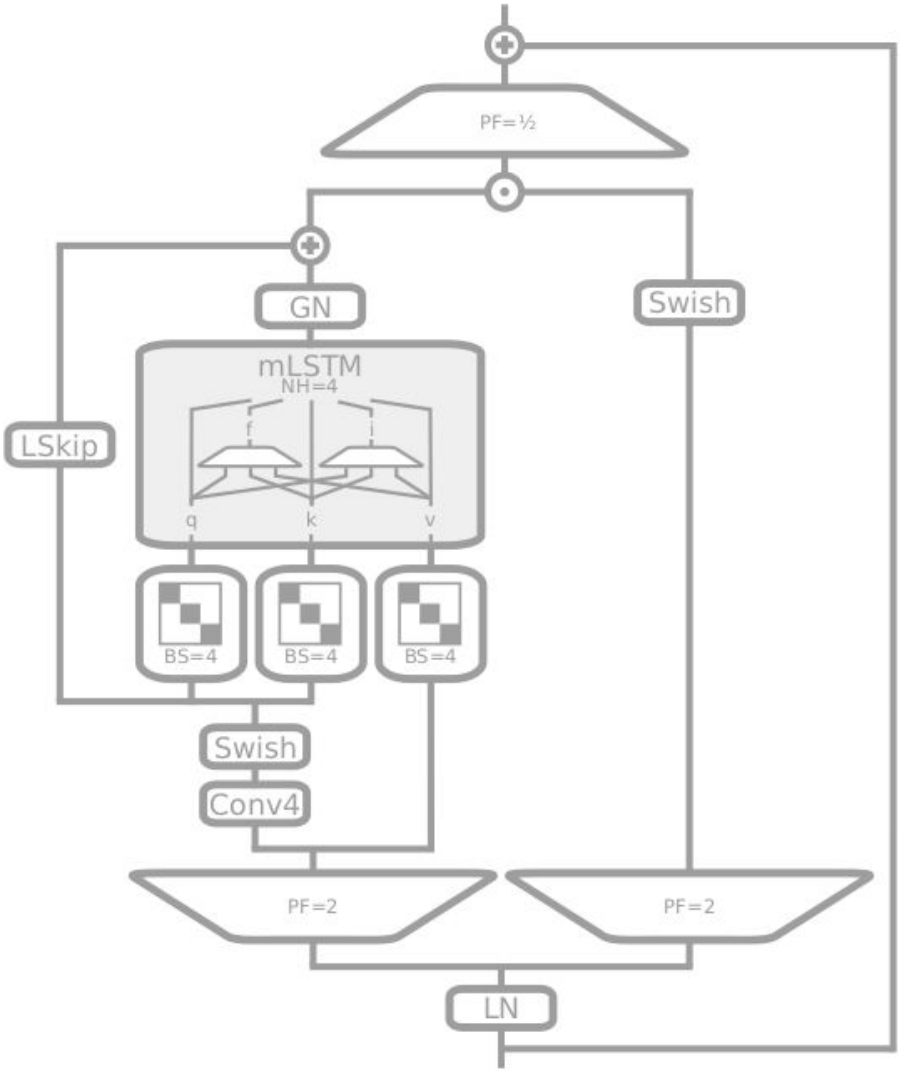
Post Up-Projection Block

- Typically used for sLSTM
- Known from Transformer
- Memory mixing
- Block-diagonal: multiple heads



Pre Up-Projection Block

- Typically used for mLSTM
- Known from State Space Models
- High memory capacity through pre up-projection
- External output gate
- Convolution for queries and keys
- Swish activations
- Learnable skip connections



xLSTM: state tracking

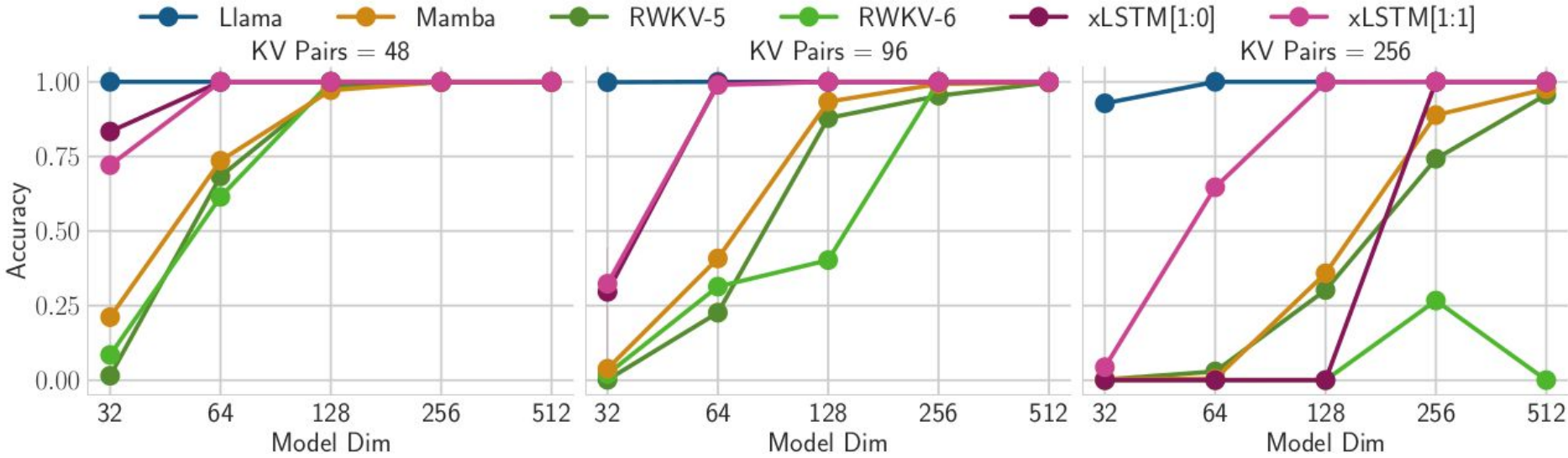
Formal Languages

e.g. majority count

some tasks require state tracking

	Context Sensitive		Deterministic Context Free		Regular					
	Bucket Sort	Missing Duplicate	Mod Arithmetic (w Brackets)	Solve Equation	Cycle Nav	Even Pairs	Mod Arithmetic (w/o Brackets)	Parity	Majority	Majority Count
Llama	0.92 ± 0.02	0.08 ± 0.0	0.02 ± 0.0	0.02 ± 0.0	0.04 ± 0.01	1.0 ± 0.0	0.03 ± 0.0	0.03 ± 0.01	0.37 ± 0.01	0.13 ± 0.0
Mamba	0.69 ± 0.0	0.15 ± 0.0	0.04 ± 0.01	0.05 ± 0.02	0.86 ± 0.04	1.0 ± 0.0	0.05 ± 0.02	0.13 ± 0.02	0.69 ± 0.01	0.45 ± 0.03
Retention	0.13 ± 0.01	0.03 ± 0.0	0.03 ± 0.0	0.03 ± 0.0	0.05 ± 0.01	0.51 ± 0.07	0.04 ± 0.0	0.05 ± 0.01	0.36 ± 0.0	0.12 ± 0.01
Hyena	0.3 ± 0.02	0.06 ± 0.02	0.05 ± 0.0	0.02 ± 0.0	0.06 ± 0.01	0.93 ± 0.07	0.04 ± 0.0	0.04 ± 0.0	0.36 ± 0.01	0.18 ± 0.02
RWKV-4	0.54 ± 0.0	0.21 ± 0.01	0.06 ± 0.0	0.07 ± 0.0	0.13 ± 0.0	1.0 ± 0.0	0.07 ± 0.0	0.06 ± 0.0	0.63 ± 0.0	0.13 ± 0.0
RWKV-5	0.49 ± 0.04	0.15 ± 0.01	0.08 ± 0.0	0.08 ± 0.0	0.26 ± 0.05	1.0 ± 0.0	0.15 ± 0.02	0.06 ± 0.03	0.73 ± 0.01	0.34 ± 0.03
RWKV-6	0.96 ± 0.0	0.23 ± 0.06	0.09 ± 0.01	0.09 ± 0.02	0.31 ± 0.14	1.0 ± 0.0	0.16 ± 0.0	0.22 ± 0.12	0.76 ± 0.01	0.24 ± 0.01
LSTM (Block)	0.99 ± 0.0	0.15 ± 0.0	0.76 ± 0.0	0.5 ± 0.05	0.97 ± 0.03	1.0 ± 0.0	0.91 ± 0.09	1.0 ± 0.0	0.58 ± 0.02	0.27 ± 0.0
LSTM	0.94 ± 0.01	0.2 ± 0.0	0.72 ± 0.04	0.38 ± 0.05	0.93 ± 0.07	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.82 ± 0.02	0.33 ± 0.0
xLSTM[0:1]	0.84 ± 0.08	0.23 ± 0.01	0.57 ± 0.09	0.55 ± 0.09	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.75 ± 0.02	0.22 ± 0.0
xLSTM[1:0]	0.97 ± 0.0	0.33 ± 0.22	0.03 ± 0.0	0.03 ± 0.01	0.86 ± 0.01	1.0 ± 0.0	0.04 ± 0.0	0.04 ± 0.01	0.74 ± 0.01	0.46 ± 0.0
xLSTM[1:1]	0.7 ± 0.21	0.2 ± 0.01	0.15 ± 0.06	0.24 ± 0.04	0.8 ± 0.03	1.0 ± 0.0	0.6 ± 0.4	1.0 ± 0.0	0.64 ± 0.04	0.5 ± 0.0

xLSTM: memory



Multi-Query Associative Recall task:

- For each sequence, key-value pairs are randomly chosen from a large vocabulary
- must be memorized for later retrieval.

Method comparison: language modeling

Comparison at language modeling

- SlimPajama dataset with 15B tokens
- perplexity (ppl) as metric

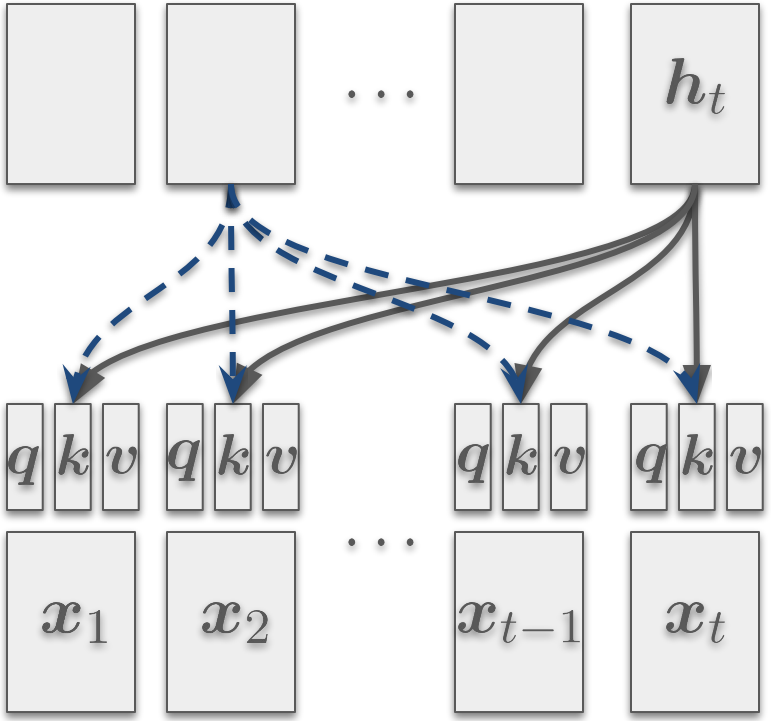
Model	#Params M	SlimPajama (15B) ppl ↓
GPT-3	356	14.26
Llama	407	14.25
H3	420	18.23
Mamba	423	13.70
Hyena	435	17.59
RWKV-4	430	15.62
RWKV-5	456	16.53
RWKV-6	442	17.40
RetNet	431	16.23
HGRN	411	21.83
GLA	412	19.56
HGRN2	411	16.77
xLSTM[1:0]	409	**13.43**
xLSTM[7:1]	408	13.48

xLSTM Limitations

- Memory mixing of the sLSTM prohibits parallelization
  Fast sLSTM CUDA kernel is 1.5 times slower than parallel mLSTM.
- CUDA kernels for mLSTM are not optimized (4 times slower than FlashAttention or Mamba's Scan). FlashAttention analog is possible
- Matrix memory of mLSTM is computationally expensive. Parallelization leads to only a minor overhead concerning the wall clock time
- Initialization of the forget gates must be chosen carefully
- Limited memory (dxd), but no problems for contexts up to 16k
- Neither architecture nor the hyperparameters are optimized

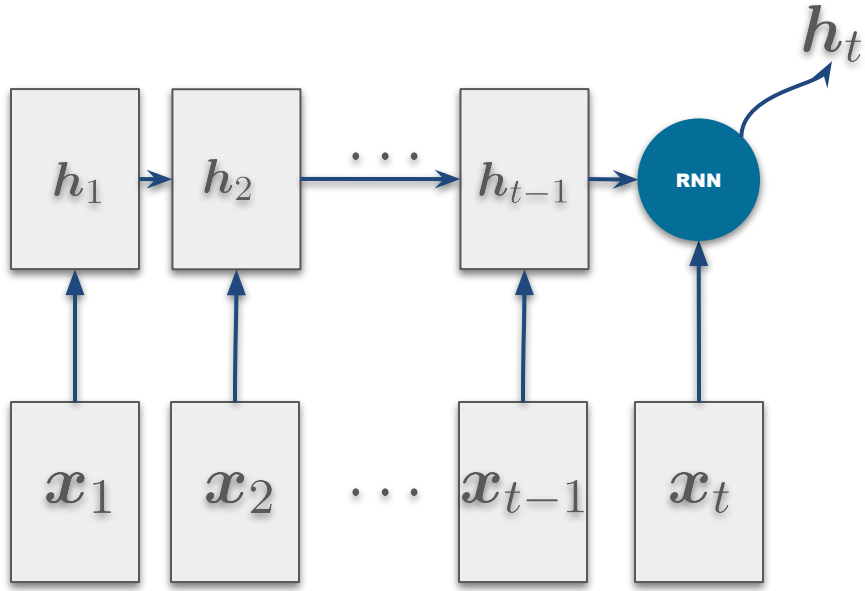
Improved efficiency through xLSTM: RNNs versus Transformer

- Currently **large language models** (LLMs) dominate the AI landscape
  - Based on Transformer's self-attention
    - **Quadratic in sequence length** (aka "context-size")
  - Prior to 2017: LSTM networks
    - ELMO: first large language model
    - **Linear in sequence length**



Improved efficiency through xLSTM: RNNs versus Transformer

- Currently **large language models** (LLMs) dominate the AI landscape
  - Based on Transformer's self-attention
    - **Quadratic in sequence length** (aka "context-size")
  - Prior to 2017: LSTM networks
    - ELMO: first large language model
    - **Linear in sequence length**



Abilities and limitations of AI
Moravec's Paradoxon

- Image recognition, object recognition
- Image generation

- Language models
  - Writing texts
  - Semantic similarity

- Board games and computer-game-like tasks

- Control tasks

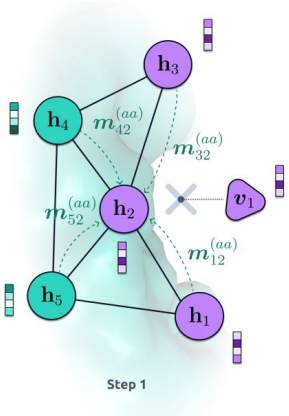
- Mathematical tasks, calculations

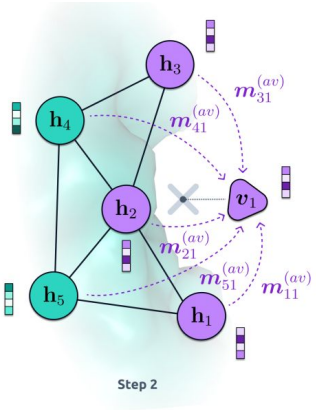
- Planning and reasoning

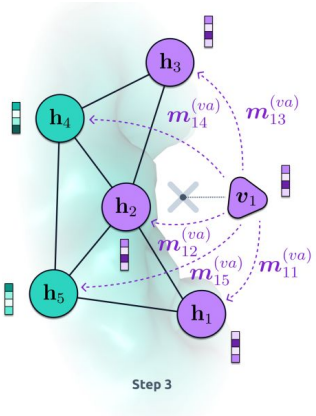
- Motoric tasks
  - setting the table

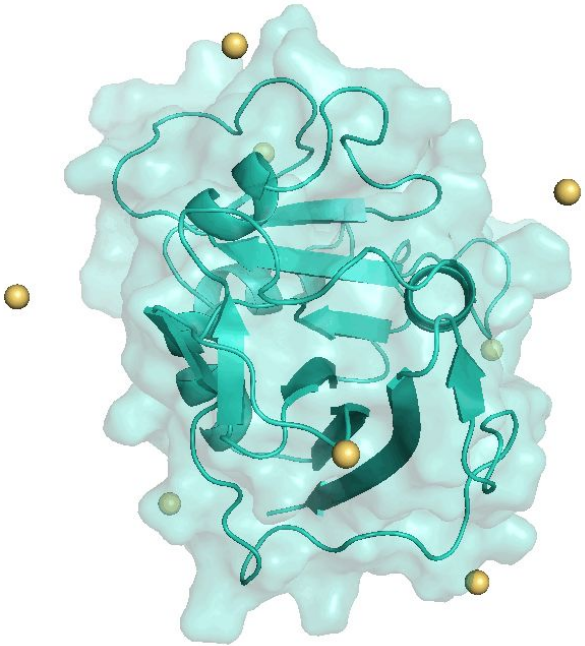
- Understanding of the physical world

Incorporating world knowledge via Geometric Deep Learning







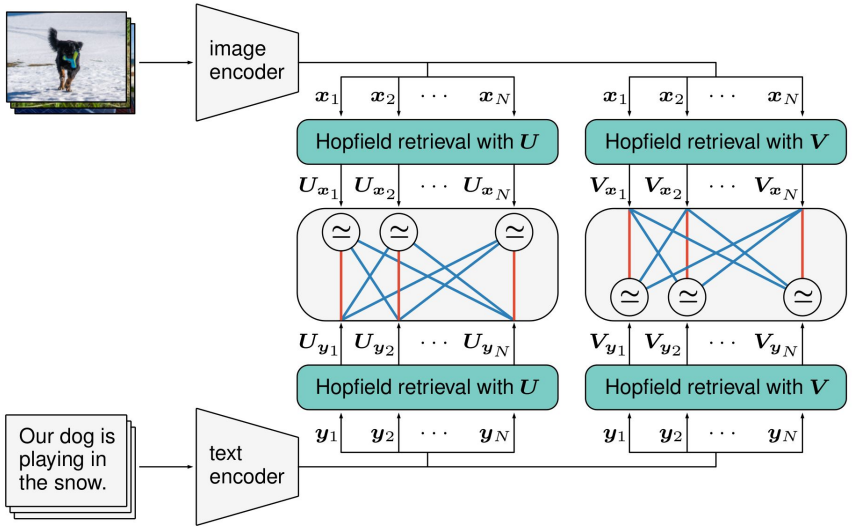


E(3)-equivariance built into deep learning architecture

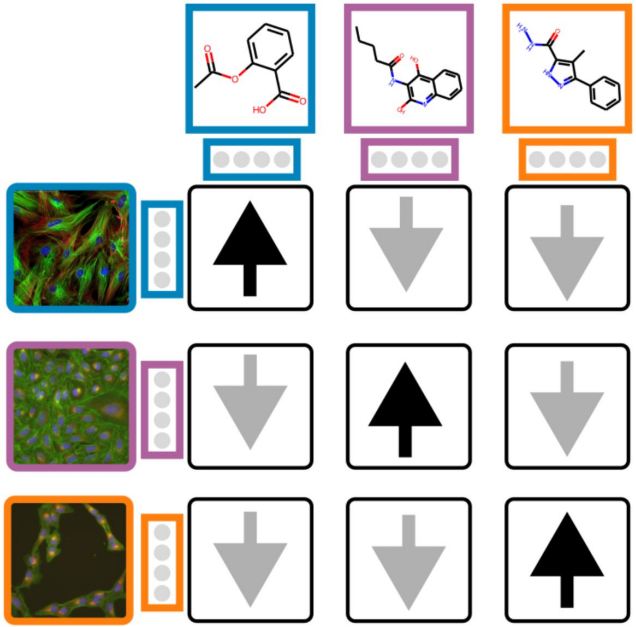
Virtual nodes for improved learning

Sestak, F., Schneckenreiter, L., Brandstetter, J., Hochreiter, S., Mayr, A., & Klambauer, G. (2024). VN-EGNN: E (3)-Equivariant Graph Neural Networks with Virtual Nodes Enhance Protein Binding Site Identification. *arXiv preprint arXiv:2404.07194*.

Improving robustness through contrastive learning

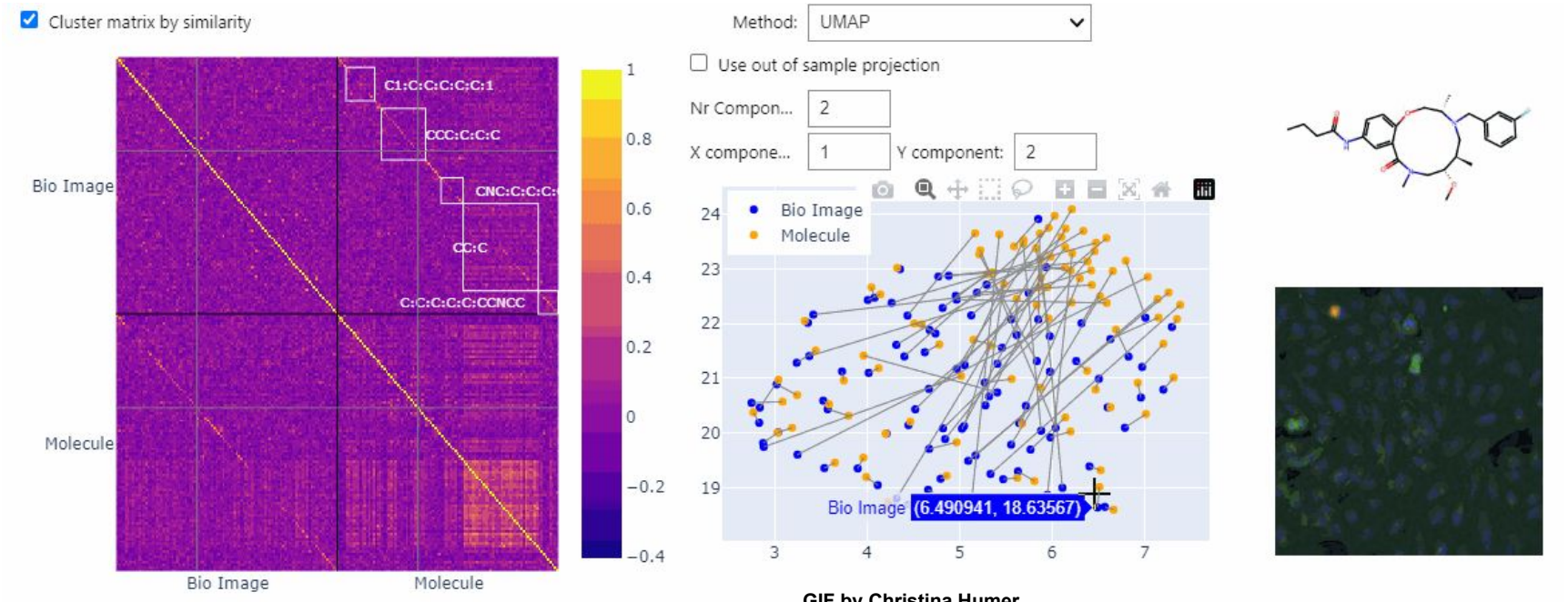


Fürst, A., Rumetshofer, E., Lehner, J., Tran, V. T., Tang, F., Ramsauer, H., ... & Hochreiter, S. (2022). Cloob: Modern hopfield networks with infoloob outperform clip. *Advances in neural information processing systems*, *35*, 20450-20468.



Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., & Klambauer, G. (2023). CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature Communications*, *14*(1), 7339.

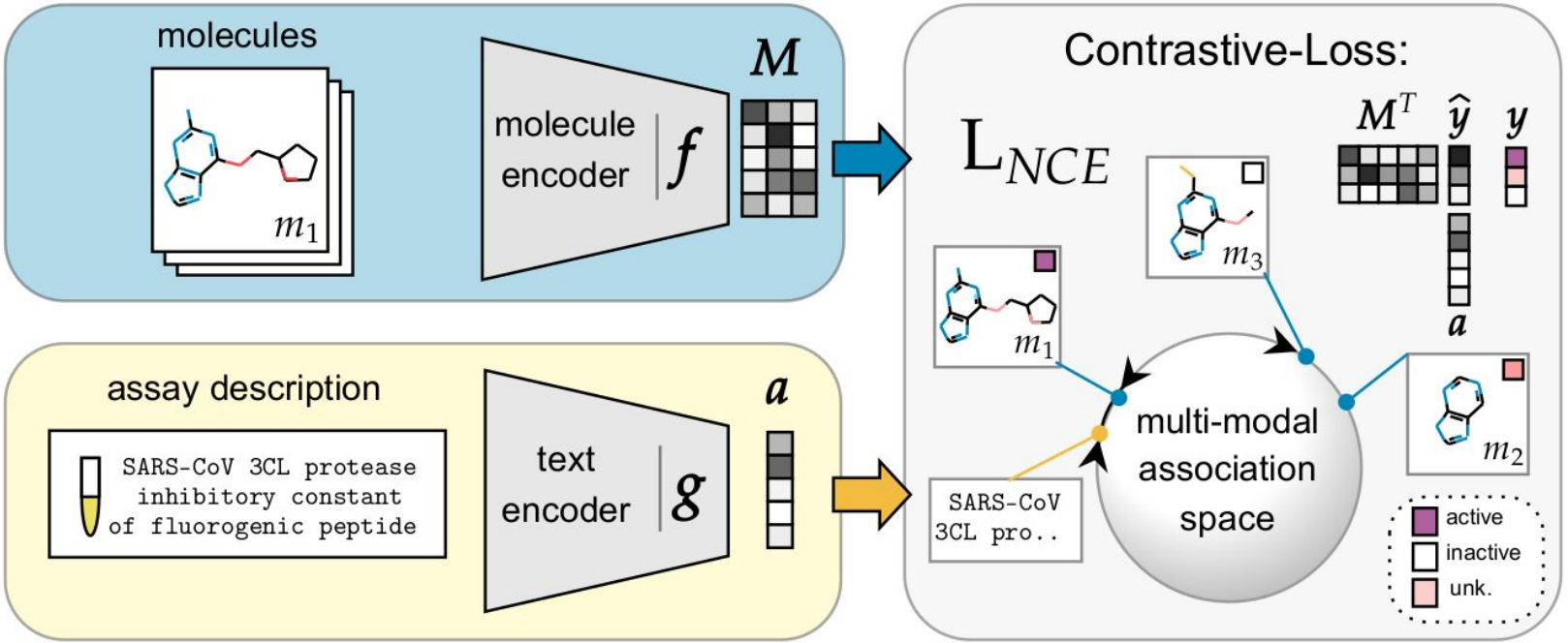
Improving robustness through contrastive learning



GIF by Christina Humer
https://ginihumer.github.io/Amumo/

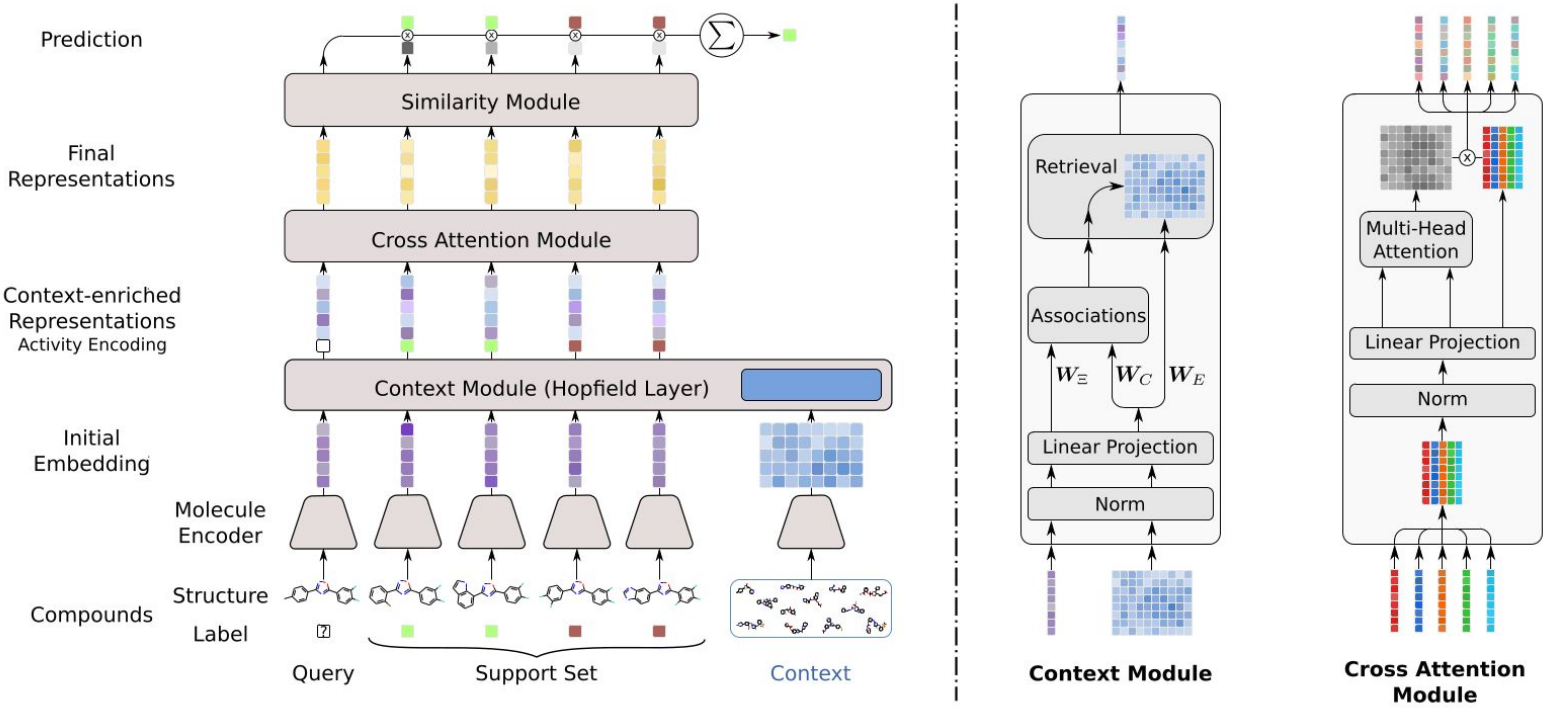
Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., & Klambauer, G. (2023). CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature Communications*, *14*(1), 7339.

Improving robustness through contrastive learning



Seidl, P., Vall, A., Hochreiter, S., & Klambauer, G. (2023, July). Enhancing activity prediction models in drug discovery with the ability to understand human language. In *International Conference on Machine Learning* (pp. 30458-30490). PMLR.

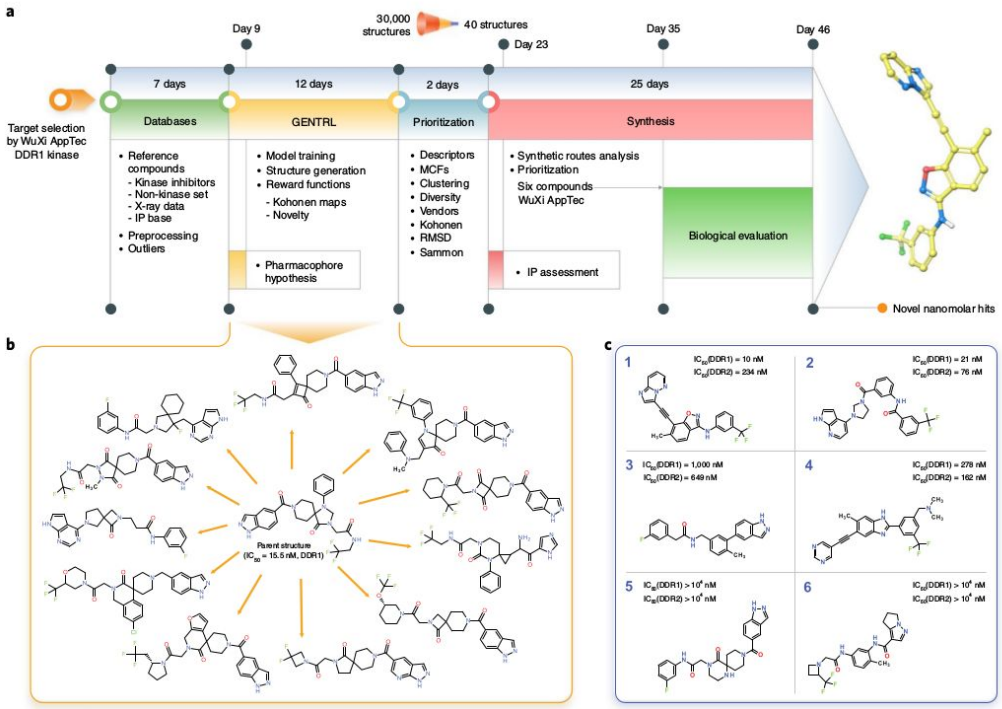
Improving adaptability through new few-shot learning approaches



Schimunek, J., Seidl, P., Friedrich, L., Kuhn, D., Rippmann, F., Hochreiter, S., & Klambauer, G. (2023). Context-enriched molecule representations improve few-shot drug discovery. *International Conference on Learning Representations*.

Drug Discovery

- **Selection, generation and prediction of drug candidates**
- **Data sets:**
  - >100,000,000 bioactivity triplets
- **Methods:**
  - Multi-task neural networks
  - Graph neural networks
  - Large language models
- **Validation:**
  - u.a. “Hit rate”
- **Results:**
  - Highly accurate prediction of bioactivities
  - Up to 100-fold increased hit-rates
  - Acceleration of pre-clinical phases of drug discovery



Zhavoronkov, A., Ivanenkov, Y. A., Aliper, A., Veselov, M. S., Aladinskiy, V. A., Aladinskaya, A. V., ... & Volkov, Y. (2019). Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nature Biotechnology, 1-4.

The rise of AI drug discovery

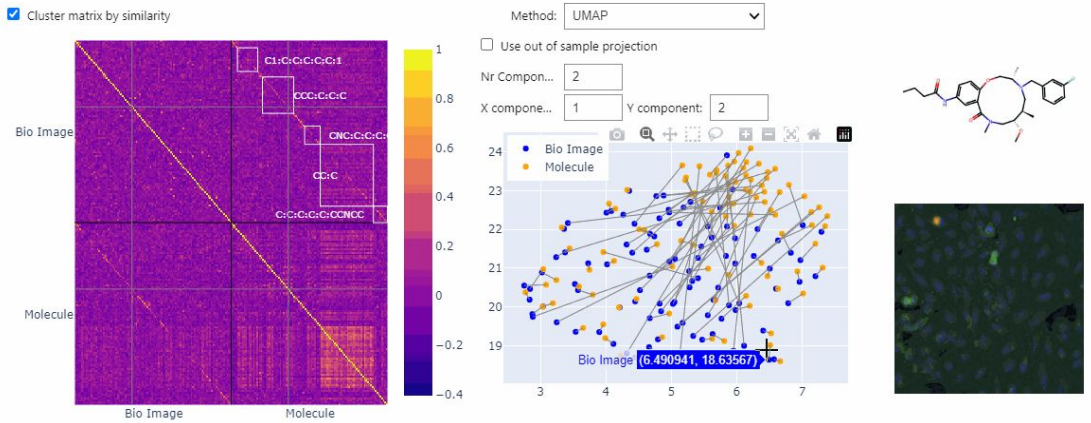
- **Artificial intelligence** will assist **life sciences** in multiple ways
  - Drug discovery and development
  - Personalized medicine
  - Medical imaging, diagnosis, and prognosis
  - Virtual assistants
  - …

- We strive for **AI systems in life sciences** that
  - can incorporate knowledge
  - are robust
  - efficient
  - and can adapt quickly

- through
  - advancing **machine learning** and **deep learning**

Drug Discovery

- **Selection, generation and prediction of drug candidates**
- **Data sets:**
  - >100,000,000 bioactivity triplets
- **Methods:**
  - Multi-task neural networks
  - Graph neural networks
  - Large language models
- **Validation:**
  - u.a. “Hit rate”
- **Results:**
  - Highly accurate prediction of bioactivities
  - Up to 100-fold increased hit-rates
  - Acceleration of pre-clinical phases of drug discovery




nature communications

Article

https://doi.org/10.1038/s41467-023-42328-w

CLOOME: contrastive learning unlocks bioimaging databases for queries with chemical structures

Received: 20 February 2023

Accepted: 6 October 2023

Ana Sanchez-Fernandez[1], Elisabeth Rumetshofer[1], Sepp Hochreiter[1,2] & Günter Klambauer[1]

GIF by Christina Humer
https://ginihumer.github.io/Amumo/

Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., & Klambauer, G. (2022). CLOOME: a new search engine unlocks bioimaging databases for queries with chemical structures.

mLSTM: memory

Memory $\boldsymbol{A} \in \mathbb{R}^{d \times d}$:

$$\boldsymbol{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,d} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{d,1} & a_{d,2} & \cdots & a_{d,d} \end{pmatrix}$$

Memory capacity:
$0.14\, d$.

Memory is independent
of the sequence length L.

Parallelizability of mLSTM