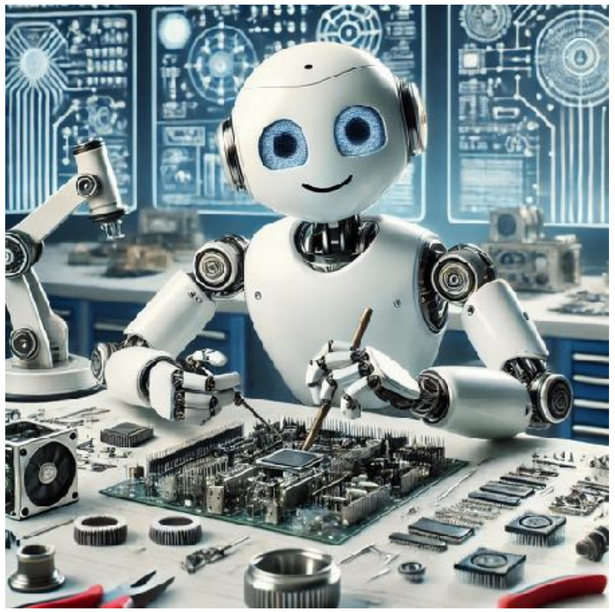
Understanding and building efficient machine learning architectures

STOC 2024



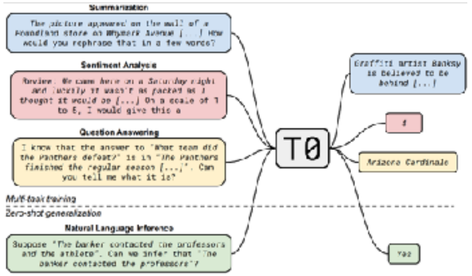
Simran Arora

We've seen language models take the world by a storm!

a BigScience initiative

BLOOM

176B params · 59 languages · Open-access



Cognition

Devin

First AI software engineer

EleutherAI

NEWS ChatGPT passes MBA exam given by a Wharton professor

CULTURE MATTERS

ChatGPT passes MBA exam given by a Wharton professor

The bot's performance on the test has "important implications for business school education," wrote Christian Terwiesch, a professor at the University of Pennsylvania's Wharton School.

Meta

Healthcare, Language Processing

ChatGPT Out-scores Medical Students on Complex Clinical Care Exam Questions

A new study shows AI's capabilities at analyzing medical text and offering diagnoses — and forces a rethink of medical education.

Jul 17, 2023 | Adam Hadhazy

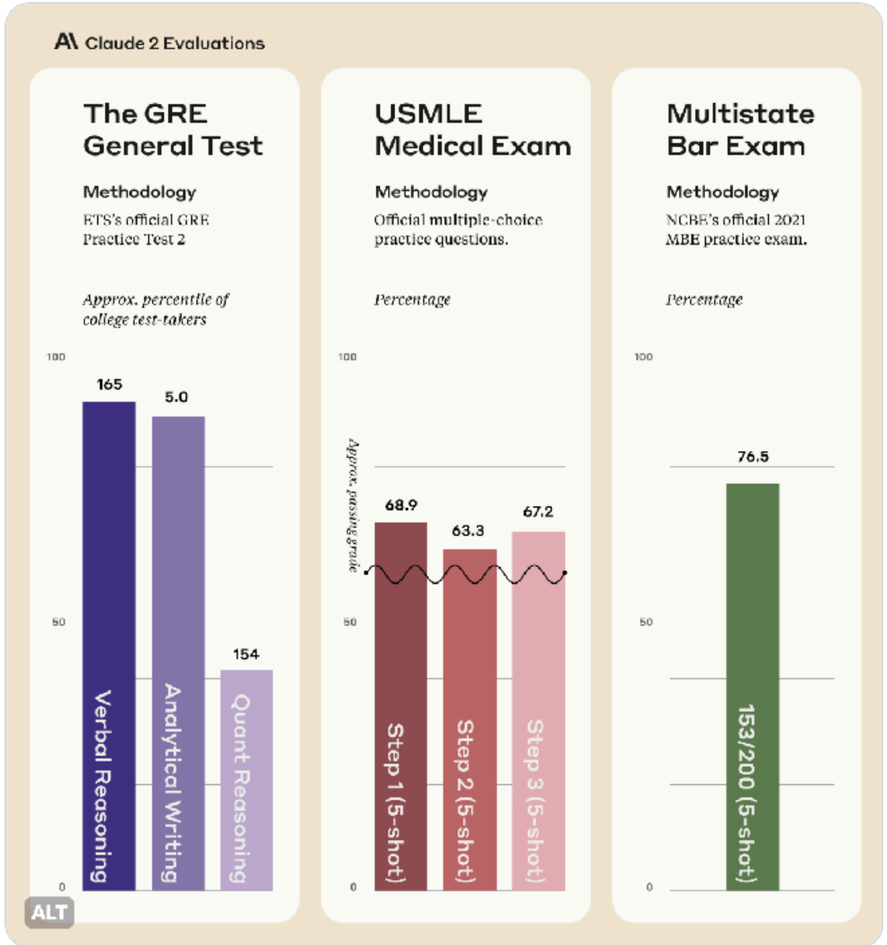
GPT-4 Passes the Bar Exam: What That Means for Artificial Intelligence Tools in the Legal Profession

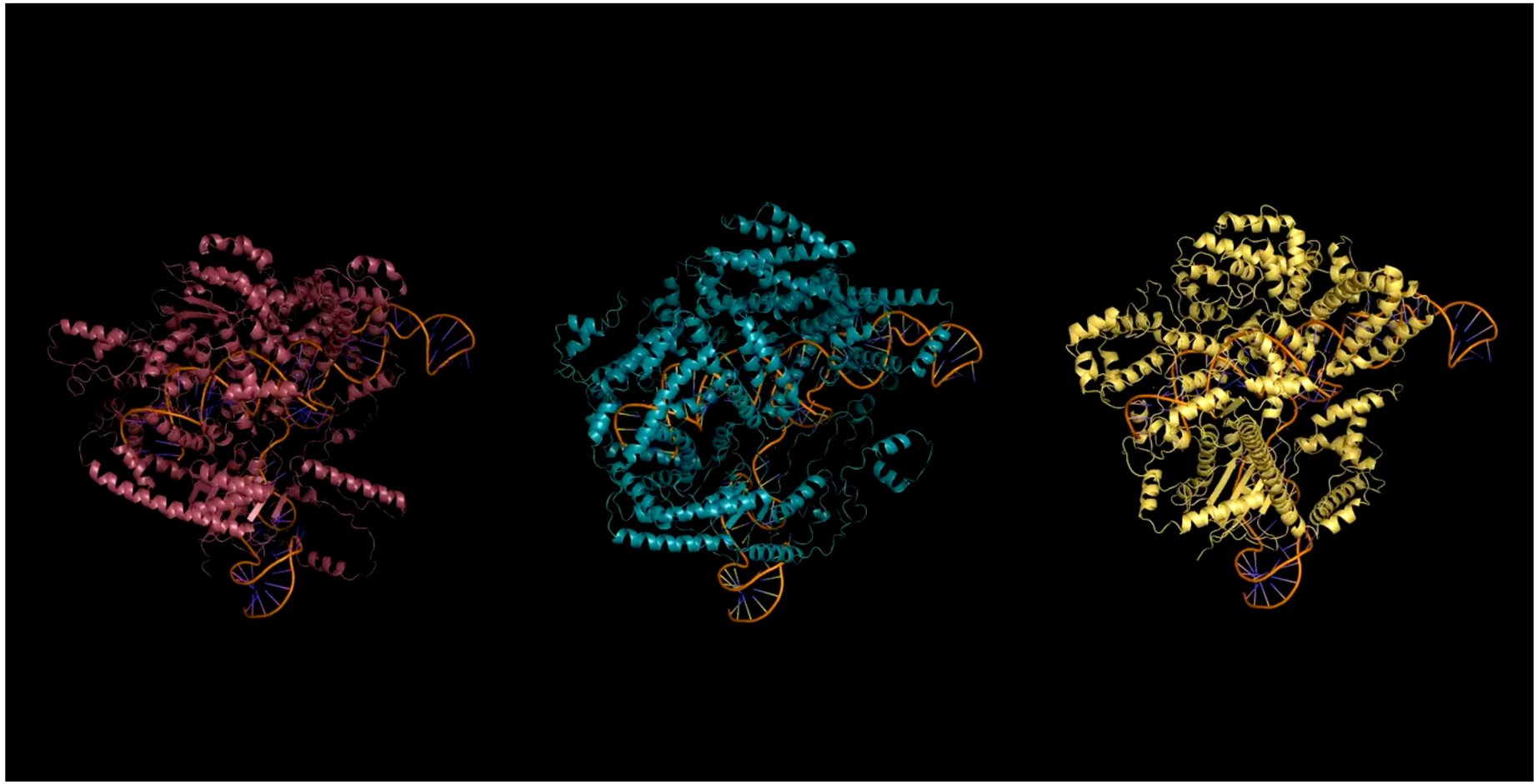
April 16, 2023 | By Pablo Arredondo, Q&A with Sharon Driscoll and Monica Schreiber

CodeX—The Stanford Center for Legal Informatics and the legal technology company Casetext recently announced what they called "a watershed moment." Research collaborators had deployed GPT-4, the latest generation Large Language Model (LLM), to take—and pass—the Uniform Bar Exam (UBE). GPT-4 didn't just squeak by. It passed the multiple-choice portion of the exam and both components of the written portion,

Anthropic
@AnthropicAI

Introducing Claude 2! Our latest model has improved performance in coding, math and reasoning. It can produce longer responses, and is available in a new public-facing beta website at claude.ai in the US and UK.

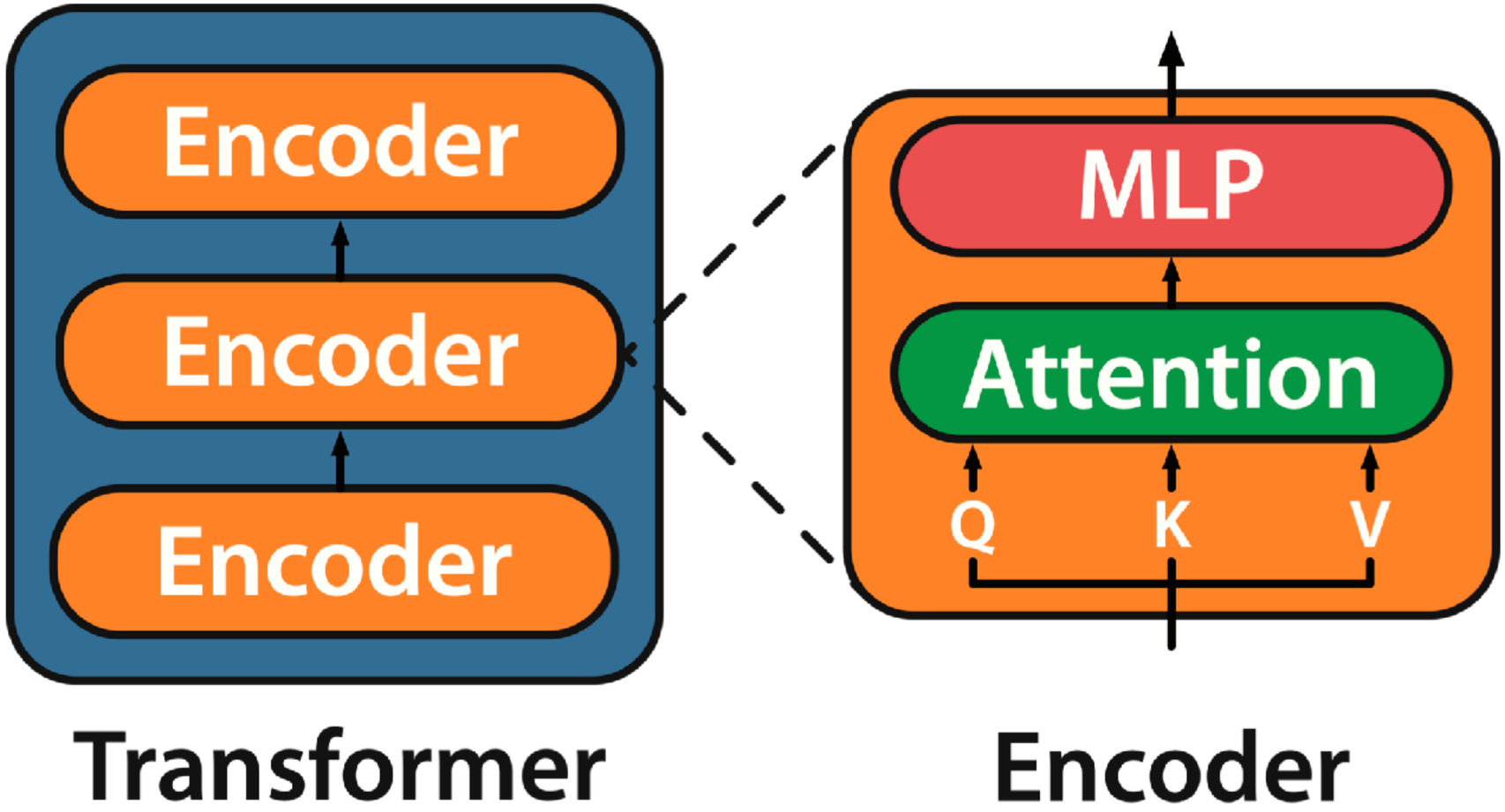




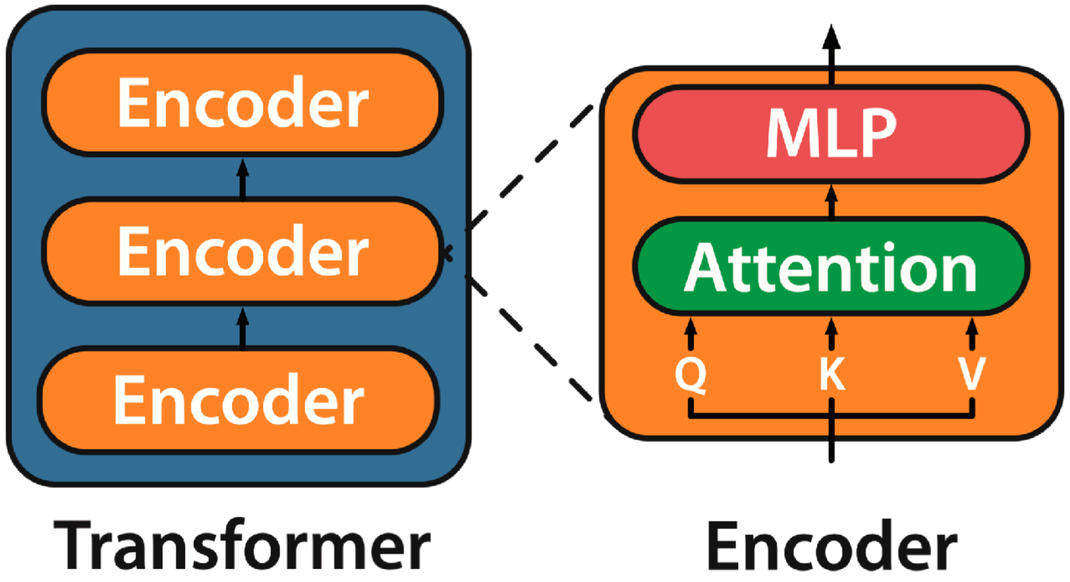
Even modeling biological (CRISPER) systems!

Nguyen et al., Evo: Long-context modeling from molecular to genome scale, https://www.together.ai/blog/evo

Our current large models use the Transformer architecture.



Given $u \in \mathbb{R}^{N \times d}$ inputs, the "sequence mixer" is the part of an architecture that mixes along N.



In Transformers, the sequence mixer is an **attention operation.**

Vaswani et al., *Attention is all you need*, 2017.

Attention computation.

Given $x \in \mathbb{R}^{N \times d}$ input:

$$Q = W_q x, \quad Q \in \mathbb{R}^{N \times d}$$

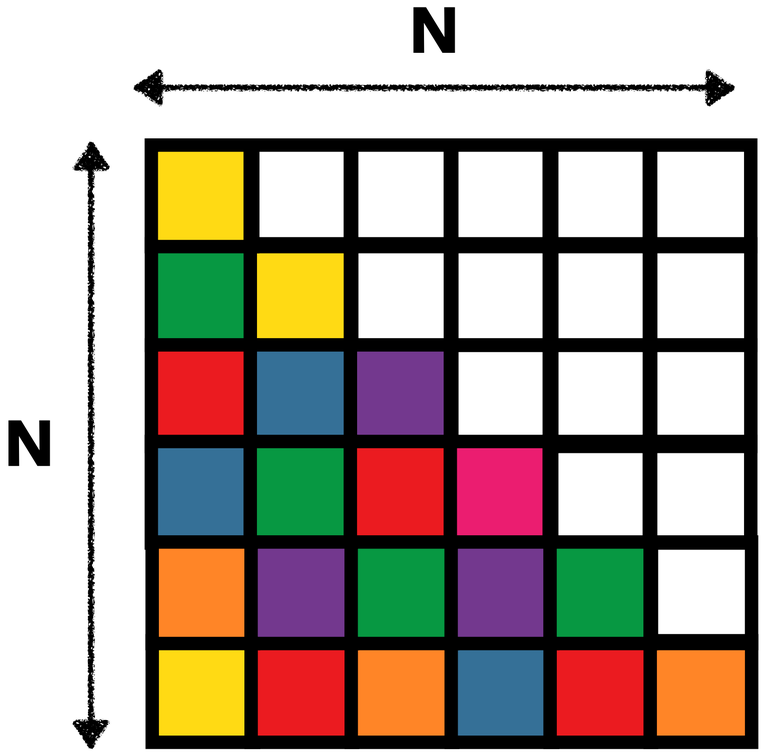
$$K = W_k x, \quad K \in \mathbb{R}^{N \times d}$$

$$V = W_v x, \quad V \in \mathbb{R}^{N \times d}$$

$$y = \exp(\underbrace{QK^T}) V$$

$$(N \times d) \times (d \times N)$$

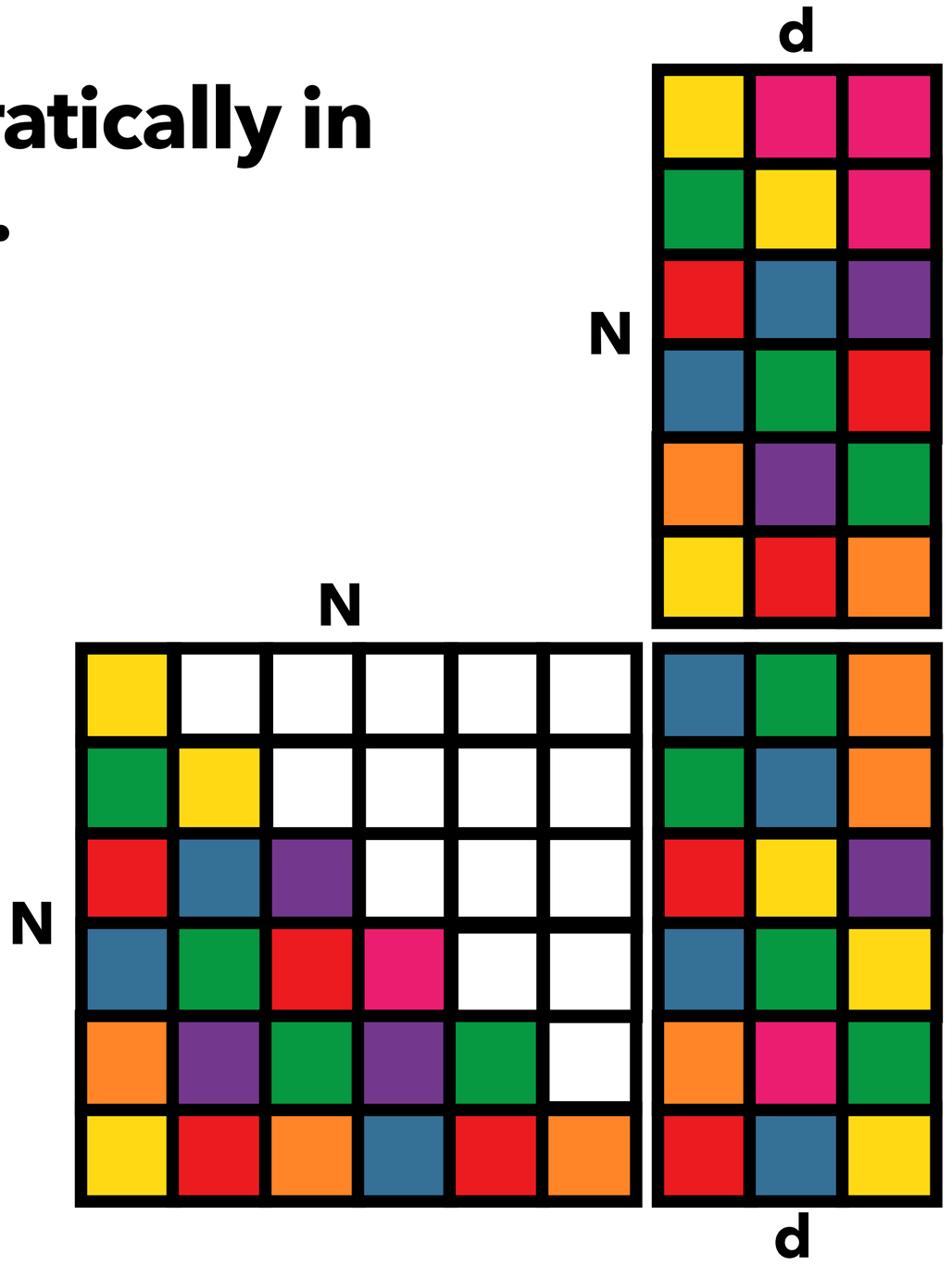
$$O(N^2 d)$$



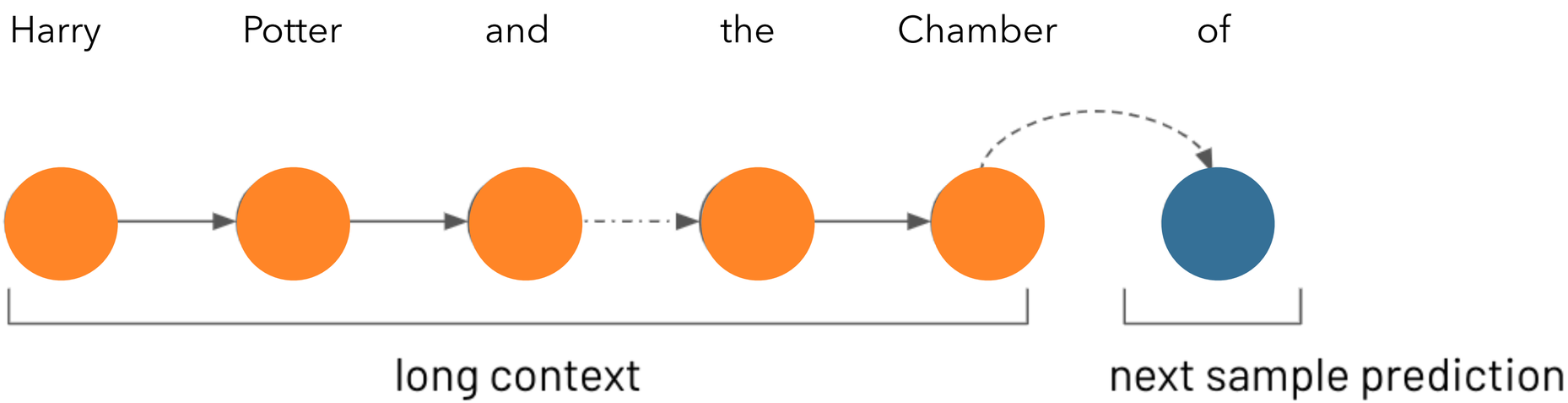
$\exp(QK^T)$ gives an $N \times N$ matrix of scalar weights that tells us how to mix the value representations V

Attention scales quadratically in length during training.

$$y = \exp(QK^T)V$$

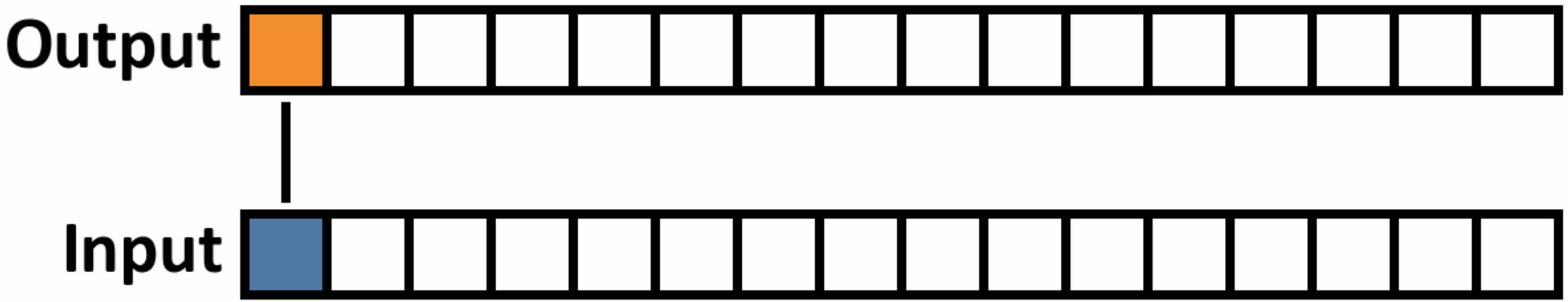


Our current large models generate one token at a time.



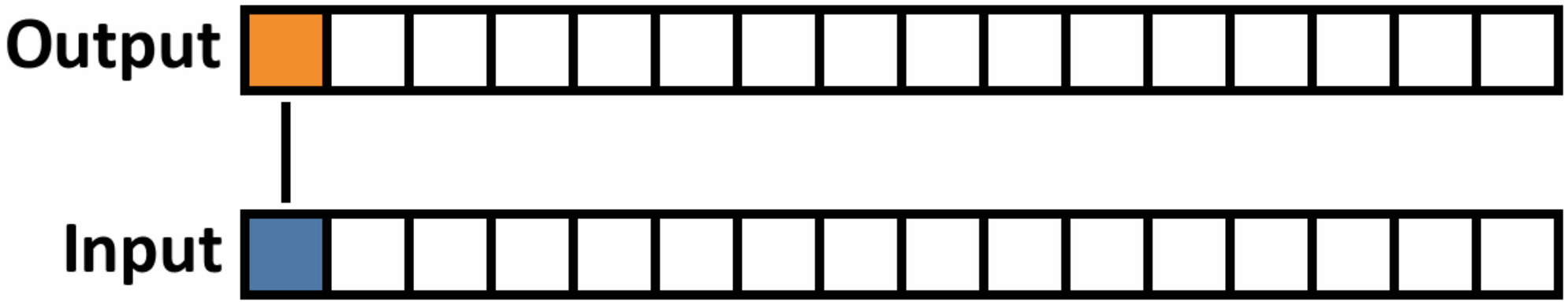
They are trained to predict the next token given the past tokens in the sequence.

When we generate one at a time, each new token position t interacts with every prior token in the sequence (positions $[0..t]$).



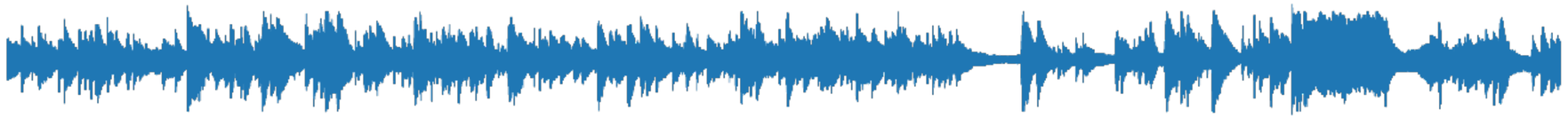
Transformers are slow: **compute and GPU memory required grows** as we generate for longer.

When we generate one at a time, each new token position t interacts with every prior token in the sequence (positions $[0..t]$).



The massive compute needed by Transformers prevents us from reaching the full potential of ML...

Thousands of time steps in a single second of raw audio

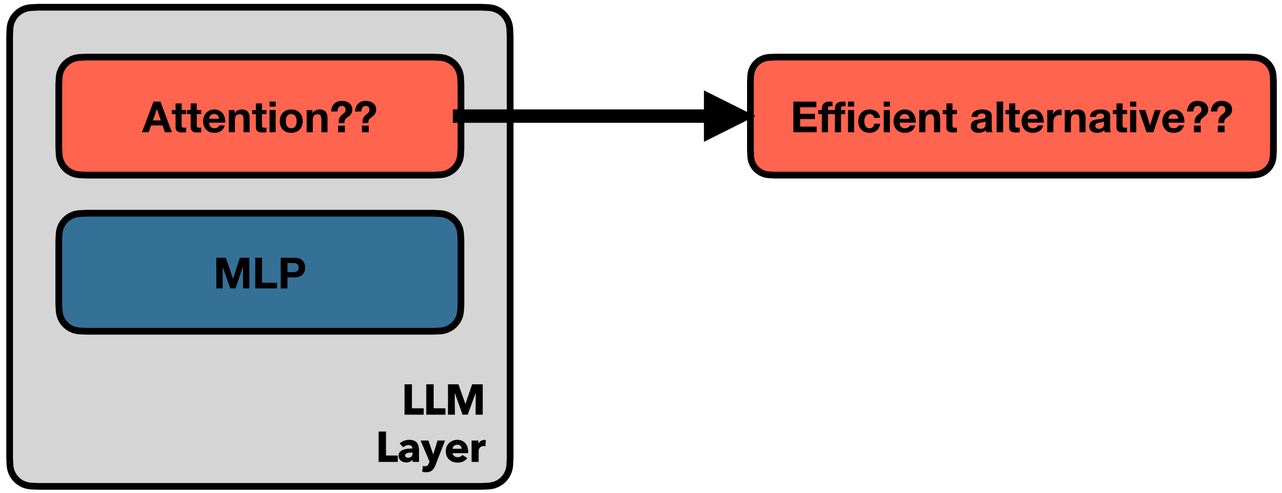


Speech

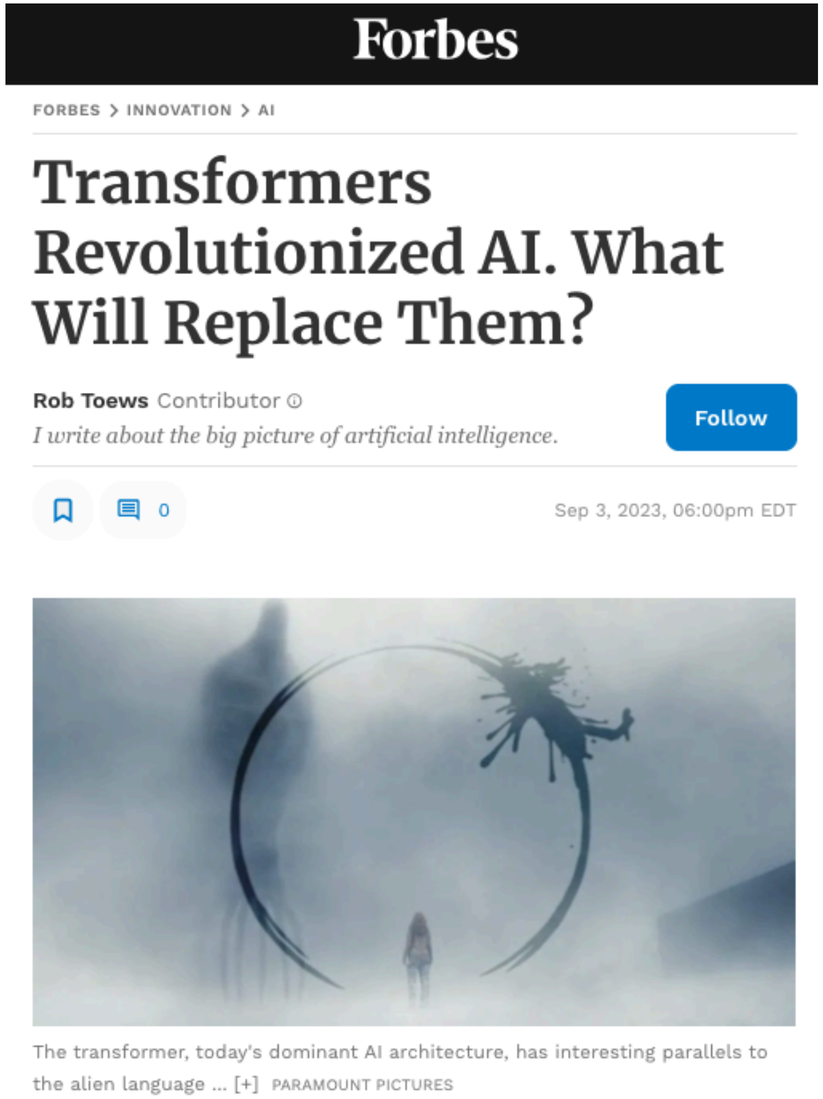
Music

3.2 Bn nucleotide pairs in a human genome sequence

Excitingly, many new efficient architecture proposals!



Transformers are RNNs [Katharopoulos et al.], **S4** [Gu et al.], **RFA** [Pent et al.], **CosFormer** [Qin et al.]**, Performer** [Choromanski et al.], **Linformer** [Wang et al.], **DSS** [Gupta], **GSS** [Mehta et al.], **S4D** [Gu et al.], **Liquid S4** [Hasani et al.], **H3** [Fu et al.], **S5** [Smith et al.] **BIGS** [Wang et al.], **Hyena** [Poli et al.], **RWKV** [Peng et al.], **RetNet** [Sun et al.], **M2** [Fu et al.], **Mamba** [Gu et al.], **Hedgehog** [Zhang et al.]**, Based** [Arora et al.,], **GLA** [Yang et al.], **GateLoop** [Kastch et al.], **Hawk/Griffin** [De et al.]



Forbes

FORBES > INNOVATION > AI

Transformers Revolutionized AI. What Will Replace Them?

Rob Toews Contributor

I write about the big picture of artificial intelligence.

Follow

0

Sep 3, 2023, 06:00pm EDT

The transformer, today's dominant AI architecture, has interesting parallels to the alien language ... [+] PARAMOUNT PICTURES

Deviating from the Transformer-orthodoxy is risky!

TECH· A.I.

The cost of training AI could soon become too much to bear

The cost of training the most advanced AI models may soon be too much to bear, some experts forecast.

Burning US five and one dollar bills. (Photo by Tom Stoddart/Getty Images) GETTY IMAGES

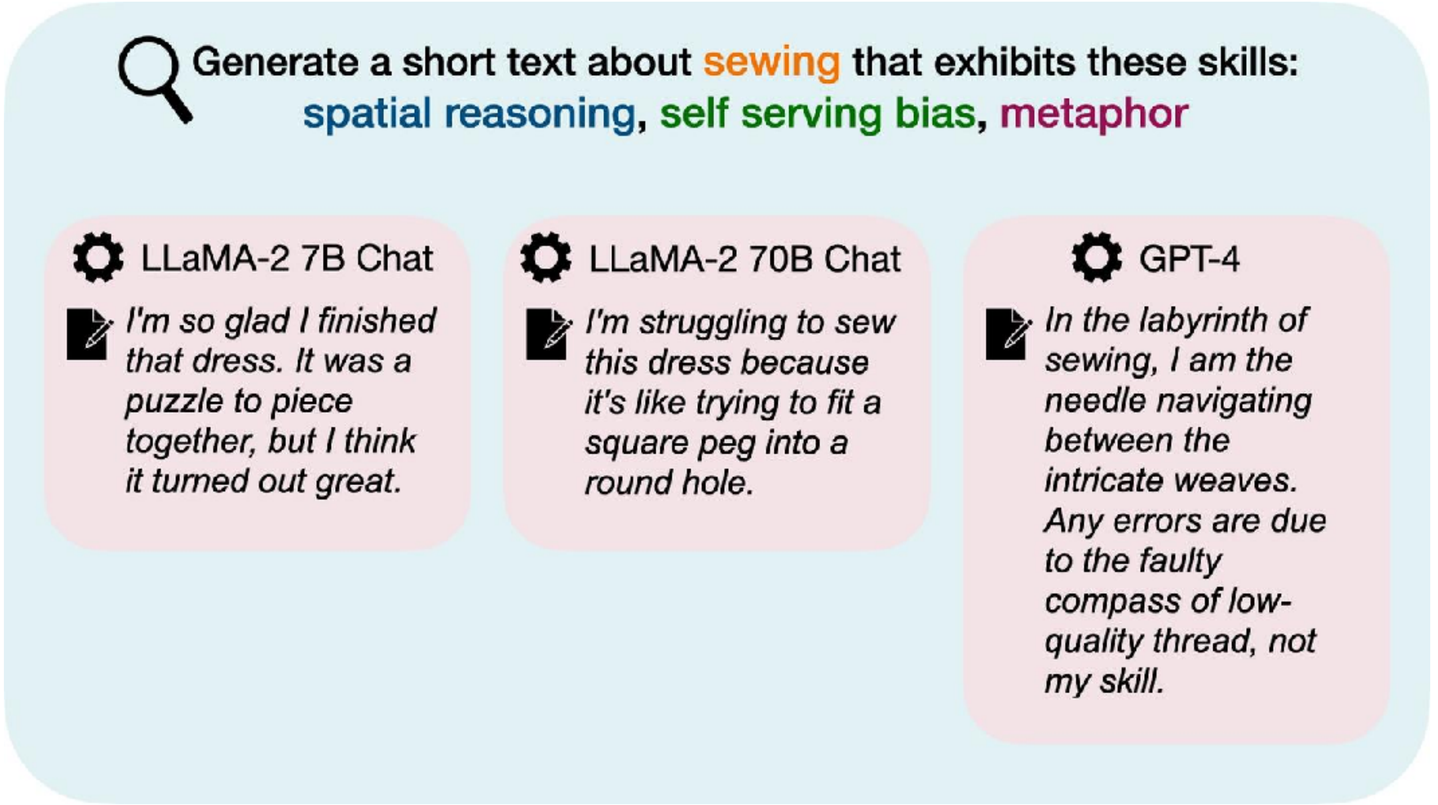
GPT-4, PaLM, Claude, Bard, LaMDA, LLaMA, Chinchilla, Sparrow – the list of large-language models on the market continues to grow. But behind their remarkable capabilities, users are discovering substantial costs.

unimagined size. GPT-4, the latest of those projects, was likely trained using trillions of words of text and many thousands of powerful computer chips. The process cost over $100 million.

Language modeling requires many varied skills at many varied levels of abstraction!

Fine-Grained Categories
Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers
Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity
Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone
Common Sense, World Knowledge

Wang et al., Glue Benchmark.



Skill-Mix, Princeton Language + Intelligence, 2023.

It’s not clear how Transformer alternatives will impact quality.

Talk outline

Efficient architectures. *What are prevailing approaches?*

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do efficient alternatives compare to attention?*

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do efficient alternatives compare to attention?*

Explaining the gap. *Using synthetic language modeling problems and theory to explain the tradeoffs of prior attention alternatives.*

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do efficient alternatives compare to attention?*

Explaining the gap. *Using synthetic language modeling problems and theory to explain the tradeoffs of attention alternatives.*

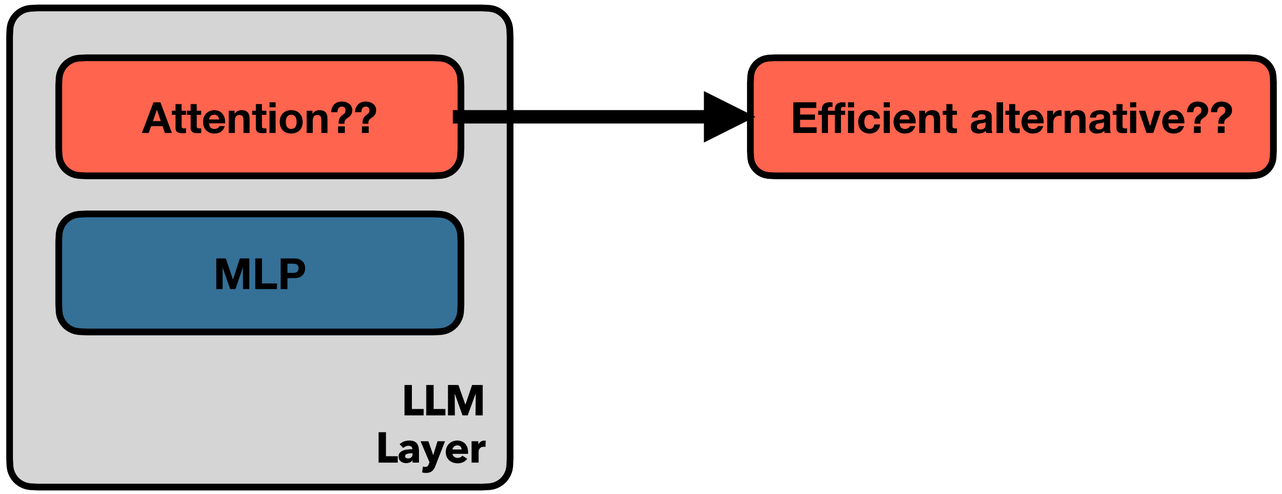
Bridging the gap. Using our insights to build new architectures that extend the Pareto frontier of the quality-efficiency tradeoff space.

Goal: Show how theory informed the way we bridged the gap!

Talk outline

Efficient architectures. *What are prevailing approaches?*

Can we replace attention?



Forbes

FORBES > INNOVATION > AI

Transformers Revolutionized AI. What Will Replace Them?

Rob Toews Contributor

I write about the big picture of artificial intelligence.

Follow

0

Sep 3, 2023, 06:00pm EDT

The transformer, today's dominant AI architecture, has interesting parallels to the alien language ... [+] PARAMOUNT PICTURES

We'll look at two prevailing (closely related) classes of efficient architectures

1. Linear attention (Katharopoulos et al., 2020)

   RFA [Pent et al.], **CosFormer** [Qin et al.]**, Performer** [Choromanski et al.], **Linformer** [Wang et al.], **Hedgehog** [Zhang et al.]**, Based** [Arora et al.,], **GLA** [Yang et al.], **ReBased** [Aksenov et al.],

2. State space models (Gu et al., 2021)

   DSS [Gupta], **GSS** [Mehta et al.], **S4D** [Gu et al.], **Liquid S4** [Hasani et al.], **H3** [Fu et al.], **S5** [Smith et al.] **BIGS** [Wang et al.], **Hyena** [Poli et al.], **RWKV** [Peng et al.], **RetNet** [Sun et al.], **M2** [Fu et al.], **Mamba** [Gu et al.], **GateLoop** [Kastch et al.], …

1 Linear attention (Katharopoulos et al., 2020)

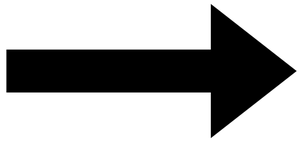
Transformers are RNNs:
Fast Autoregressive Transformers with Linear Attention

Angelos Katharopoulos [1 2] **Apoorv Vyas** [1 2] **Nikolaos Pappas** [3] **François Fleuret** [2 4 *]

Linear attention computation.

Given $u \in \mathbb{R}^{N \times d}$ input:

$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \exp(QK^T)V$$



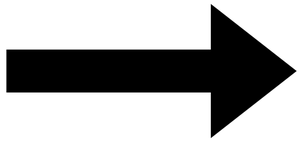
$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d'}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d'}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \phi(Q)\phi(K^T)V$$

feature map $\phi(\cdot) : \mathbb{R}^{N \times d'} \to \mathbb{R}^{N \times D}$

Linear attention computation.

$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \exp(QK^T)V$$

$$O(N^2 d)$$



$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \phi(Q)\phi(K^T)V$$

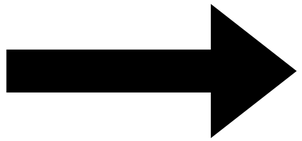
$$O(Nd^2)$$

Reorder the multiplies

Linear attention computation.

$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \exp(\underbrace{QK^T}) V$$
$$O(N^2 d)$$

Scales quadratically in N!



$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d'}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d'}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \phi(Q) \underbrace{\phi(K^T) V}$$
$$O(NDd)$$

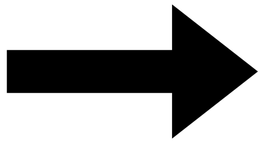
Scales linearly in N!

Reorder the multiplies

Linear attention during inference.

For the first token in the sequence ($i = 0$):

$$y_0 = \exp(Q_0\, K_0^T) V_0$$



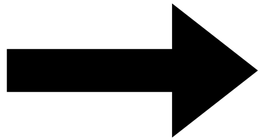
$$h_0 = \phi(K_0^\top) V_0, \qquad h_0 \in \mathbb{R}^{1 \times d^2}$$

$$y_0 = \phi(Q_0) h_0$$

Linear attention during inference.

For the second token in the sequence ($i = 1$):

$$y_1 = \exp(Q_1 \, K_{0:1}{}^{T}) V_{0:1}$$



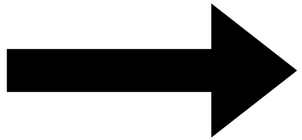
$$h_1 = h_0 + \phi(K_1^{\top}) V_1, \quad h_1 \in \mathbb{R}^{1 \times d^2}$$

$$y_1 = \phi(Q_1) h_1$$

Linear attention during inference.

At position $i = t$:

$$y_t = \exp(Q_t \, K_{0:t}{}^T) V_{0:t}$$



$$h_t = \sum_{i=0}^{t} \phi(K_t^\top) V_t, \qquad h_t \in \mathbb{R}^{1 \times d^2}$$

$$y_t = \phi(Q_t) h_t$$

Runtime and memory scales linearly with t

Runtime and memory remain constant as t grows!

tl;dr so far

Architecture	Training Complexity	Parallelizable Training	Inference Complexity
Attention	$O(N^2 d)$	✓	$O(Nd)$
Linear attention	$O(Nd^2)$	✓	$O(1)$
State Space Models			

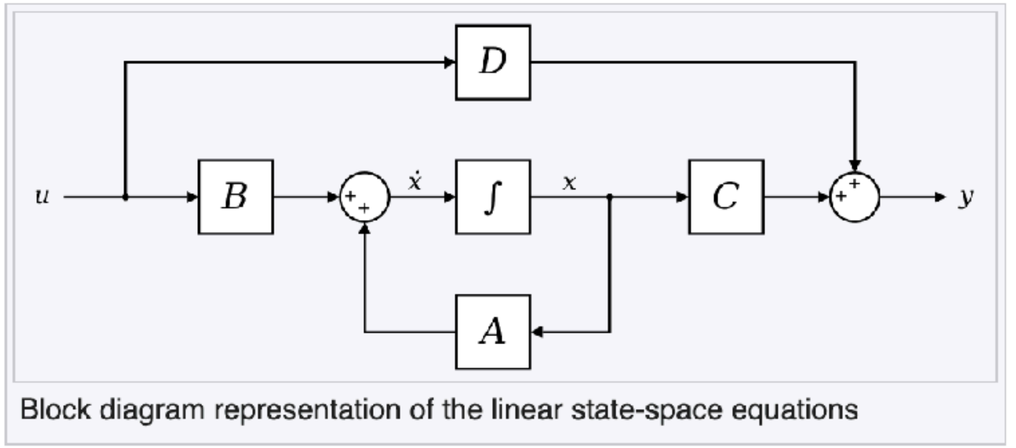
2 State space models (Gu et al., 2021)

Efficiently Modeling Long Sequences with Structured State Spaces

Albert Gu, Karan Goel, and Christopher Ré

Department of Computer Science, Stanford University

{albertgu,krng}@stanford.edu, chrismre@cs.stanford.edu



Block diagram representation of the linear state-space equations

https://en.wikipedia.org/wiki/State-space_representation

State space model architectures

State space model (SSM), parametrized by learned weights A, B and C:

$$h_t = Ah_{t-1} + Bx_t \qquad \text{for } h_t \in \mathbb{R}^{1\times d}$$
$$y_t = Ch_t$$

* We're making some simplifications since SSMs are *continuous* objects.

State space model architectures

State space model (SSM), parametrized by learned weights A, B and C:

$$h_t = Ah_{t-1} + Bx_t \qquad \text{for } h_t \in \mathbb{R}^{1\times d}$$
$$y_t = Ch_t$$

Contrast:

$$h_t = \sum_{i=0}^{t} \phi(K_t^\top) V_t,$$
$$y_t = \phi(Q_t) h_t$$

* We're making some simplifications since SSMs are *continuous* objects.

State space models during inference…

State space model, parametrized by learned weights A, B and C:

$$h_t = Ah_{t-1} + Bx_t$$
$$y_t = Ch_t$$

Runtime and memory remain constant as *t* grows!

Unrolling terms of the recurrence:

$$h_0 = Bu_0$$
$$h_1 = ABu_0 + Bu_1$$
$$\dots$$
$$h_k = A^kBu_0 + A^{k-1}Bu_1 + \ldots + Bu_k$$

$$y_k = CA^kBu_0 + CA^{k-1}Bu_1 + \ldots + CBu_k$$

State space models (SSM) during training

State space model for A, B and $C \in \mathbb{R}^{1\times d}$:

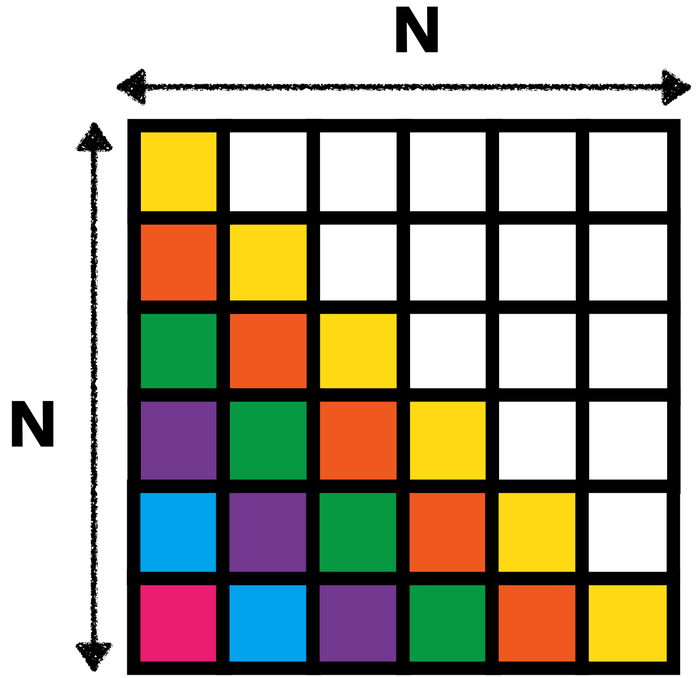
$$h_t = Ah_{t-1} + Bx_t \qquad h_t \in \mathbb{R}^{1\times d}$$
$$y_t = Ch_t$$

After unrolling the above:

$$y_N = CA^N Bx_0 + CA^{N-1}Bx_1 + \ldots + CBx_N$$

Rewrite this as a **convolution** with filter K:

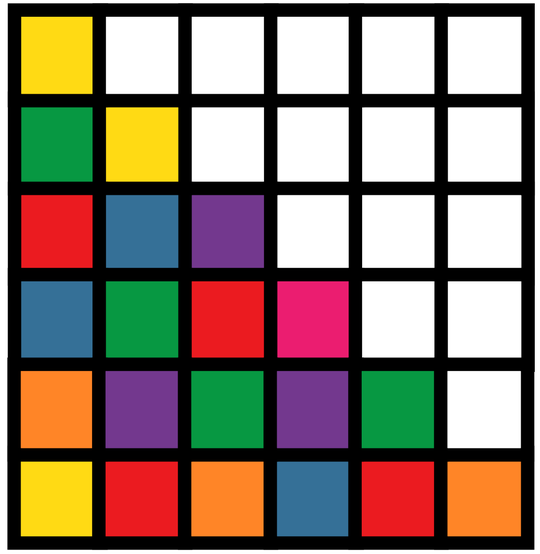
$$y = K * x$$
$$K = (CB, CAB, \ldots, CA^{N-2}B, CA^{N-1}B)$$



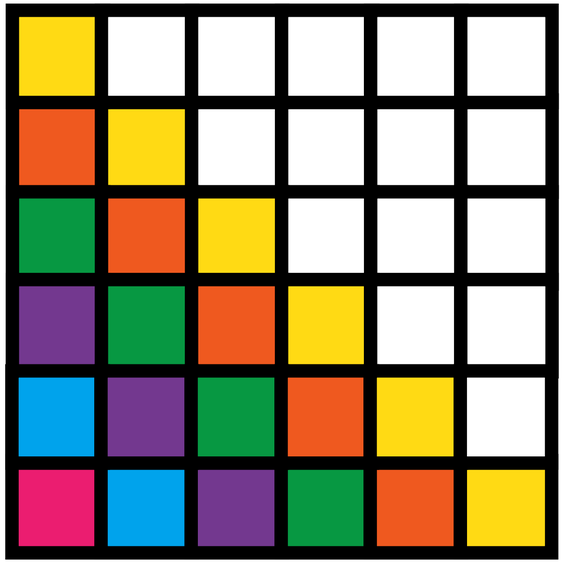
Toeplitz sequence mixing matrix

Sequence mixing matrices

Attention



Convolutions

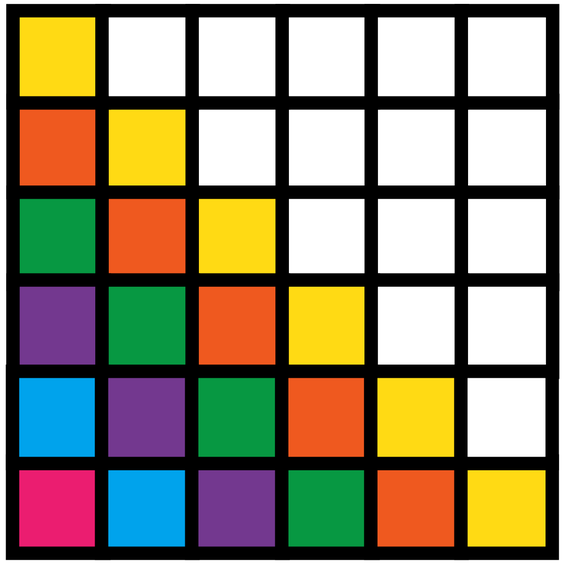


Convolutions

Convolution operation with filter K and inputs $u = [u_0, \ldots, u_N]$, for sequence length N:

$$y = K * u$$

$$K = (CB, CAB, \ldots, CA^{N-2}B, CA^{N-1}B)$$



Toeplitz sequence mixing matrix

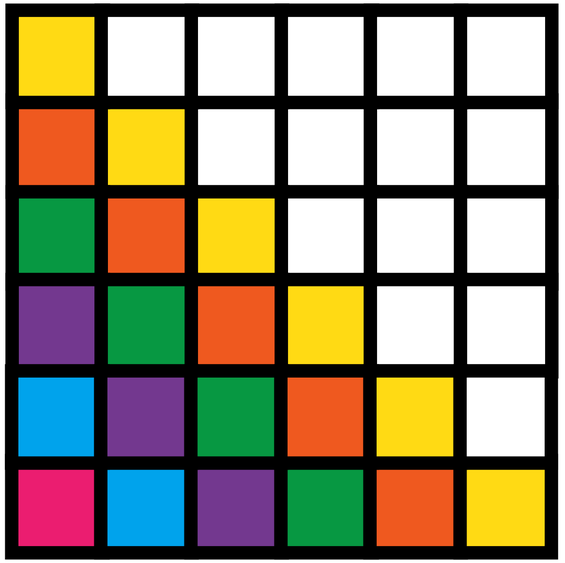
Using a Fast Fourier transform, we can compute the convolution sub-quadratically in $O(N \log N)$ compute!

Gated state space / Gated convolution models

Convolution operation with filter K and inputs $u = [u_0, \ldots, u_N]$, for sequence length N:

$$y = K * u$$

$$K = (CB, CAB, \ldots, CA^{N-2}B, CA^{N-1}B)$$



Toeplitz sequence mixing matrix

Hadamard product ("gating"): element-wise multiply between vector $v \in \mathbb{R}^{N \times d}$ and input u

$$y = K \odot u$$

tl;dr

Architecture	Training Complexity	Parallelizable Training	Inference Complexity
Attention	$O(N^2 d)$	✓	$O(Nd)$
Linear attention	$O(Nd^2)$	✓	$O(1)$
State Space Models	$O(dN \log N)$	✓	$O(1)$

Prior work suggests efficient LMs match and outperform Transformers everywhere!

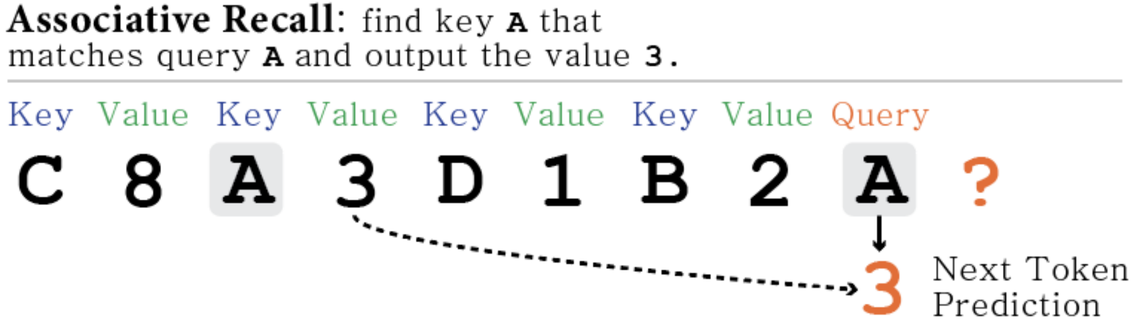
Quadratic attention has been indispensable for information-dense modalities such as language... until now.

Announcing Mamba: a new SSM arch. that has linear-time scaling, ultra long context, and most importantly--outperforms Transformers everywhere we've tried.

On overall language modeling!

Prior work suggests efficient LMs match and outperform Transformers!

Real-world example



Abstraction (“synthetic recall test”)

Prior work suggests efficient LMs match and outperform Transformers!

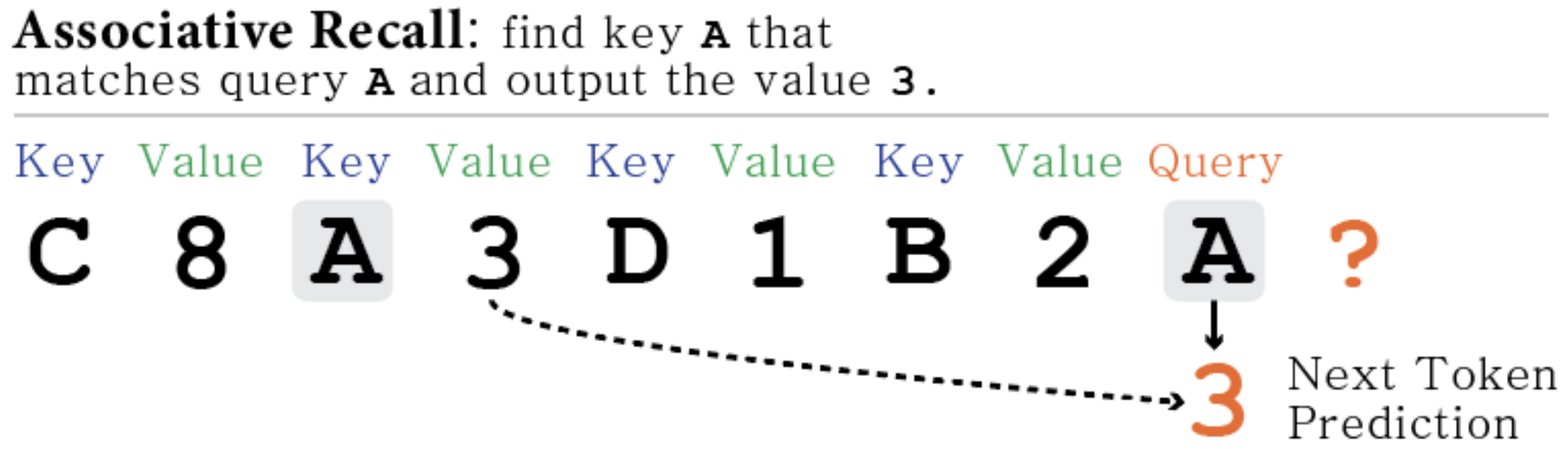


Table 2: Evaluation of 2-layer models on synthetic language tasks.

Task	Random	S4D	Gated State Spaces	H3	Attention
Induction Head	5.0	35.6	6.8	**100.0**	**100.0**
Associative Recall	25.0	86.0	78.0	99.8	**100.0**

Fu, Dao et al., Hungry Hungry Hippos, 2023. https://arxiv.org/abs/2212.14052

tl;dr

Architecture	Training Complexity	Parallelizable Training	Inference Complexity	Quality
Attention	$O(N^2 d)$	✓	$O(Nd)$	✓
Linear attention	$O(Nd^2)$	✓	$O(1)$	✓
State Space Models	$O(dN \log N)$	✓	$O(1)$	✓

Deviating from the Transformer-orthodoxy is risky!

TECH· A.I.

The cost of training AI could soon become too much to bear

The cost of training the most advanced AI models may soon be too much to bear, some experts forecast.

Burning US five and one dollar bills. (Photo by Tom Stoddart/Getty Images) GETTY IMAGES

GPT-4, PaLM, Claude, Bard, LaMDA, LLaMA, Chinchilla, Sparrow – the list of large-language models on the market continues to grow. But behind their remarkable capabilities, users are discovering substantial costs.

unimagined size. GPT-4, the latest of those projects, was likely trained using trillions of words of text and many thousands of powerful computer chips. The process cost over $100 million.

🤔 No free lunch!

How does deviating from the Transformer impact quality?

Architecture	Training Complexity	Parallelizable Training	Inference Complexity	Quality
Attention	$O(N^2 d)$	✓	$O(Nd)$	✓
Linear attention	$O(Nd^2)$	✓	$O(1)$	?
State Space Models	$O(dN \log N)$	✓	$O(1)$	?

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do sub-quadratic alternatives compare to attention?*

We trained language models across popular efficient architecture proposals and found quality gaps…

Model (360M)	Perplexity
Attention	8.39
S4 (SSM)	13.13
H3 (Gated SSM)	10.60
Hyena (Gated SSM)	10.11
RWKV-V5 (Gated SSM)	9.79
Linear attention	9.49

Let's perform an **error analysis** of the models' next token predictions!

We performed a manual **error analysis** of next token predictions, *color coding* tokens in the test set:

Both correct

Only Attention correct

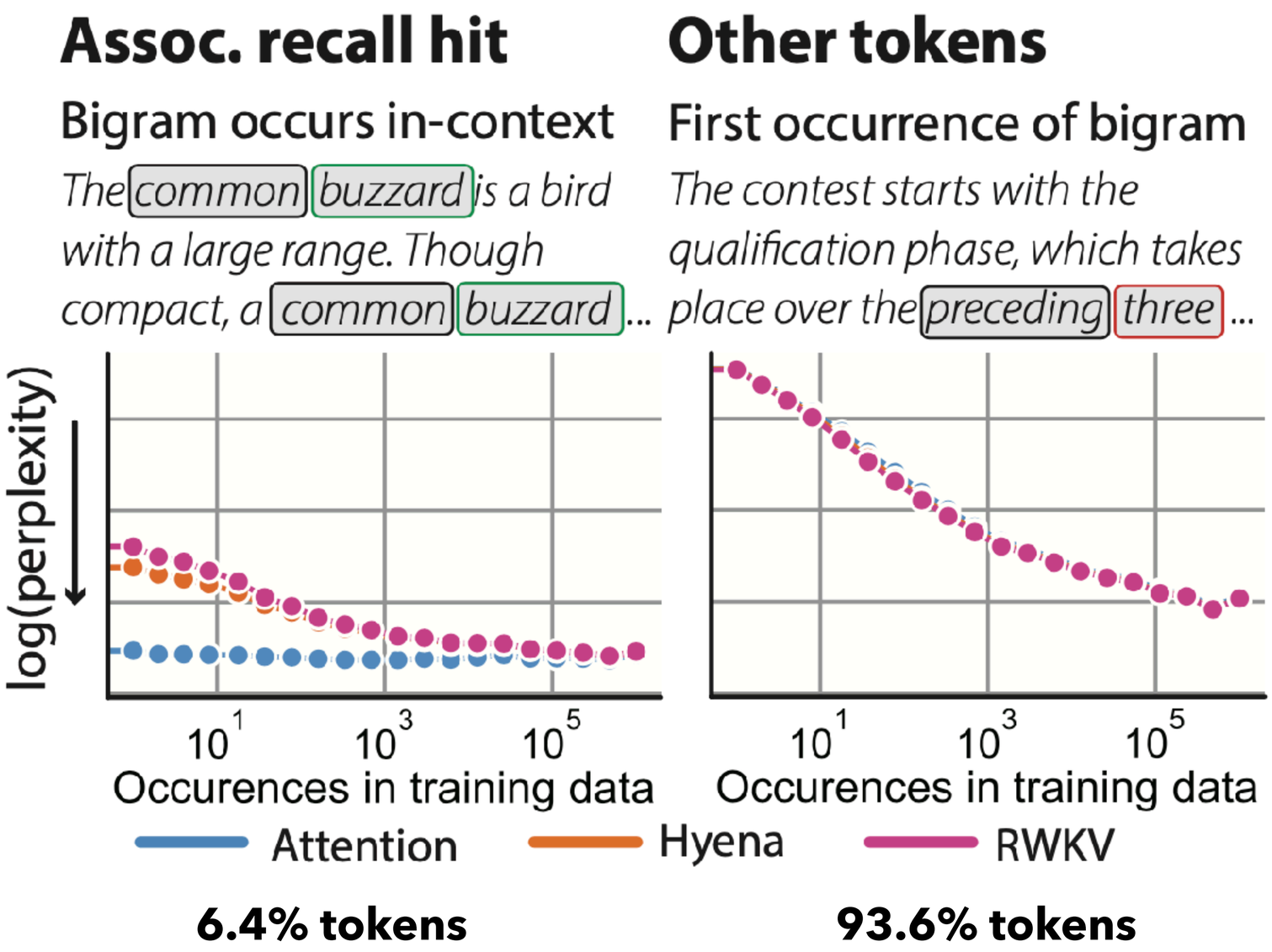
Both incorrect

Only efficient-LM correct

The second(rwkv= first, attn= second) section is all about Pixar Fest , and the (rwkv= third, attn= the) final section is all(rwkv= about, attn= all) about Pixar Pier . ...(480 tokens)... -If(rwkv=-Disney, attn=-If) there wasn -âĢ -t enough Pixar at Disneyland , Pixar Fest(rwkv= would, attn= Fest) is(rwkv= at, attn= is) coming to the Disneyland Resort on April 13 , 2018 .

A *single skill, associative recall,* was a glaring failure mode for efficient LMs.

Scaling up the error analysis.



🤔 **Associative recall accounted 80%+ of the gap** between Transformers and the efficient LMs on average, despite representing **just 6.4% of tokens**!

The efficient LMs struggled on in context learning tasks that need recall (e.g., answering questions from documents)…

Model	Averaged across 3 tasks that need recall (Accuracy metric)
Attention	**47.7**
H3 (Gated SSM)	5.1
Linear attention	17.2
Mamba	18.1

Even though they were similar to attention on non-recall language tasks.

Model	Averaged across 3 tasks that need recall (Accuracy metric)	Non recall tasks (Accuracy)
Attention	**47.7**	**44.1**
H3 (Gated SSM)	5.1	39.4
Linear attention	17.2	43.2
Mamba	18.1	43.5

Prior efficient LMs were specifically designed with AR in mind!

🤔 This was ***surprising***.

What gives?

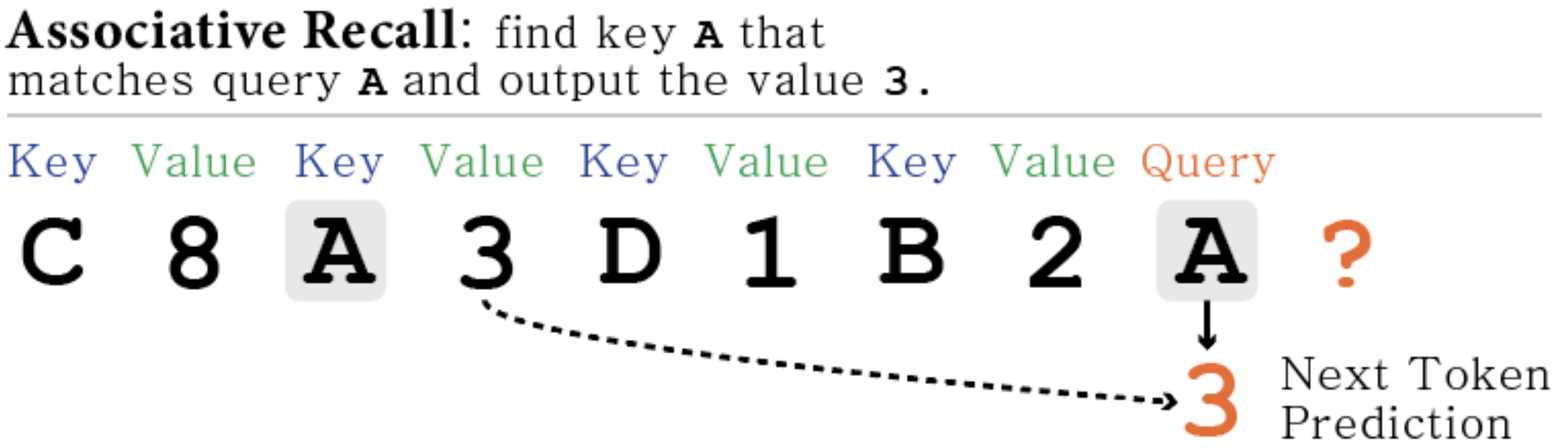


Table 2: Evaluation of 2-layer models on synthetic language tasks.

Task	Random	S4D	Gated State Spaces	H3	Attention
Induction Head	5.0	35.6	6.8	**100.0**	**100.0**
Associative Recall	25.0	86.0	78.0	99.8	**100.0**

https://arxiv.org/abs/2212.14052

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do sub-quadratic alternatives compare to attention?*

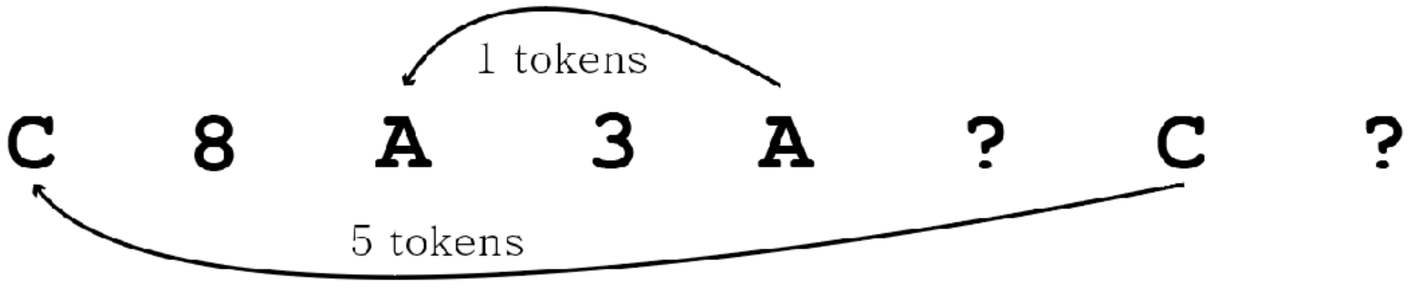
Explaining the gap. *Using synthetics and theory to explain the tradeoffs of prior attention alternatives.*

Let’s start with an intuitive explanation…

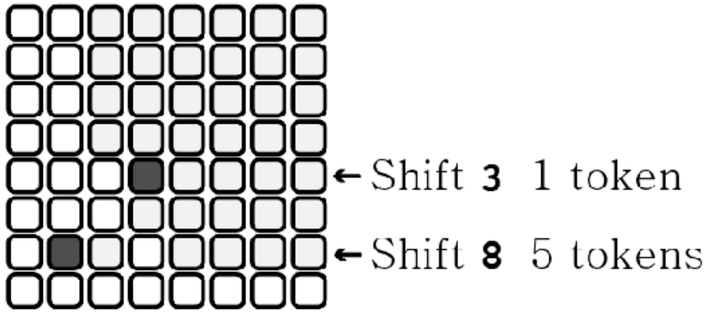
What does it take for a model to solve AR?

1. **Compare**: Which pairs of tokens in the sequence match?

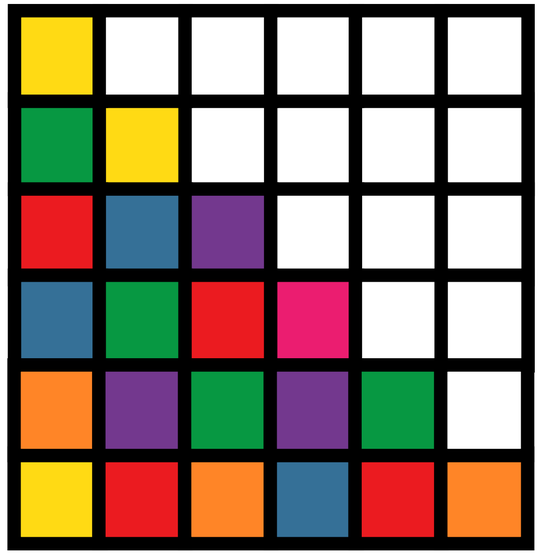
2. **Shift**: Bring forwards the values (e.g., 3) corresponding to the matches (e.g. A) to generate the next token (fill the "?")



Ideal sequence mixing matrix

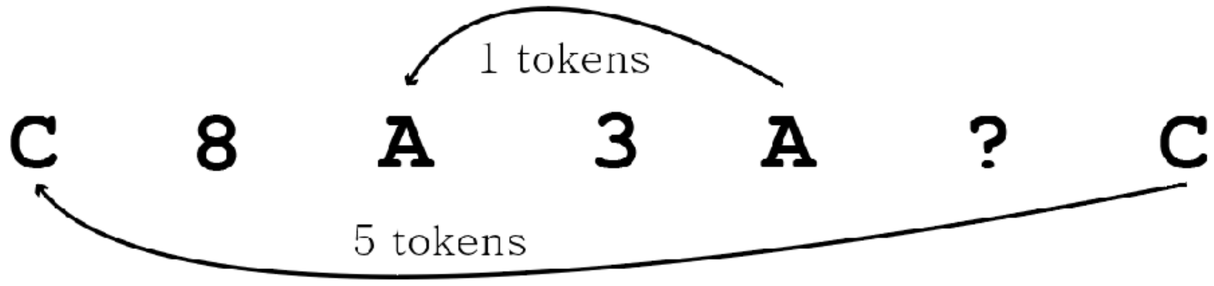


This is easy with attention inner products!

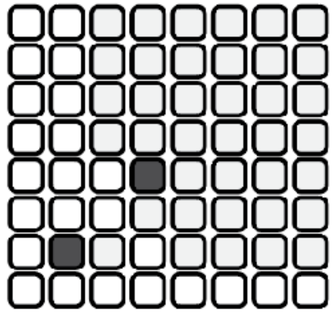


Attention computes an "input-dependent" mixing matrix:

$$A = \exp((uW_Q)(uW_k)^T)$$

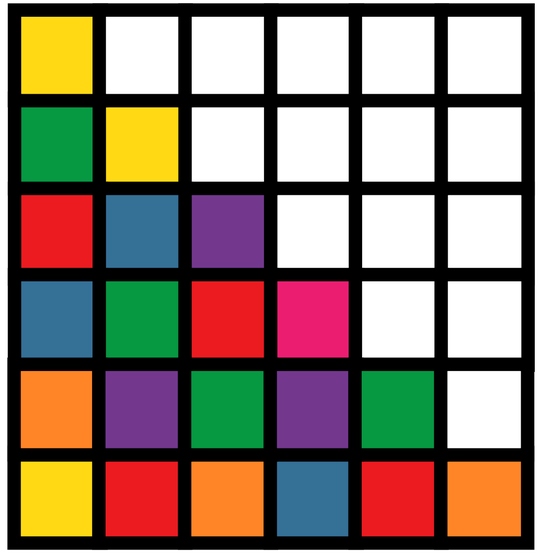


Ideal sequence mixing matrix



But our gated convolution models mix in more restricted ways.

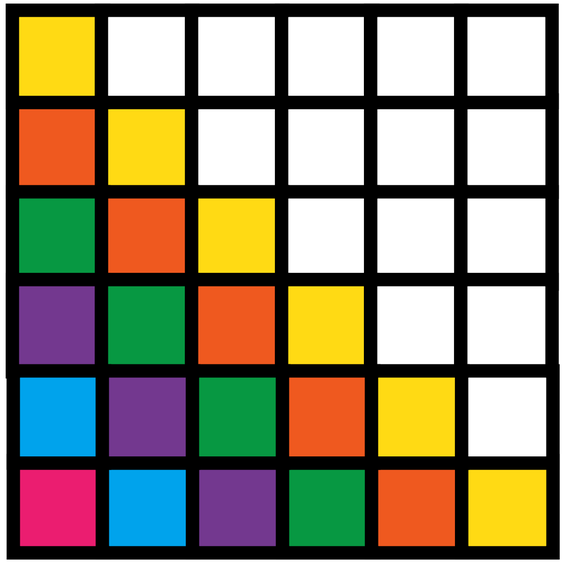
Attention



Attention performs input-dependent mixing

$$A = \exp((uW_Q)(uW_k)^T)$$

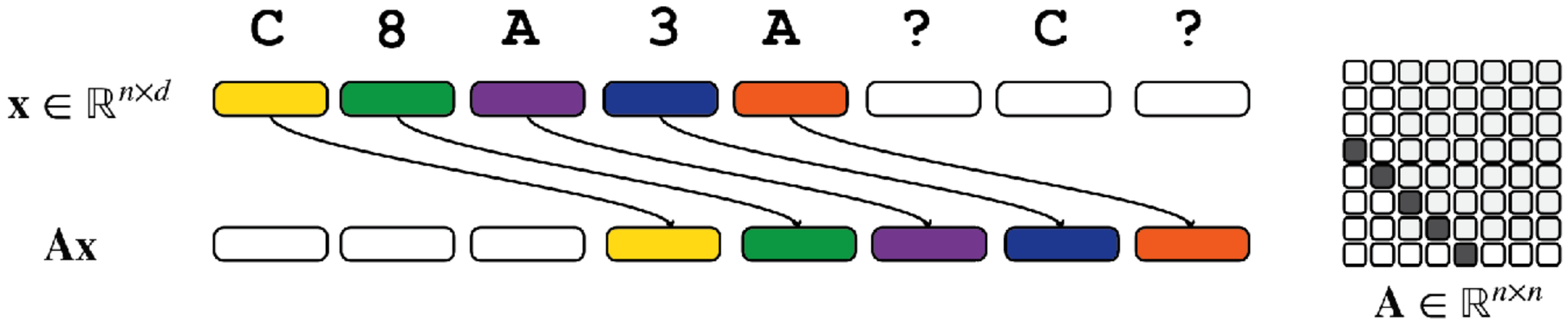
Convolutions



Convolutions do not perform input-dependent mixing. The filters are fixed.

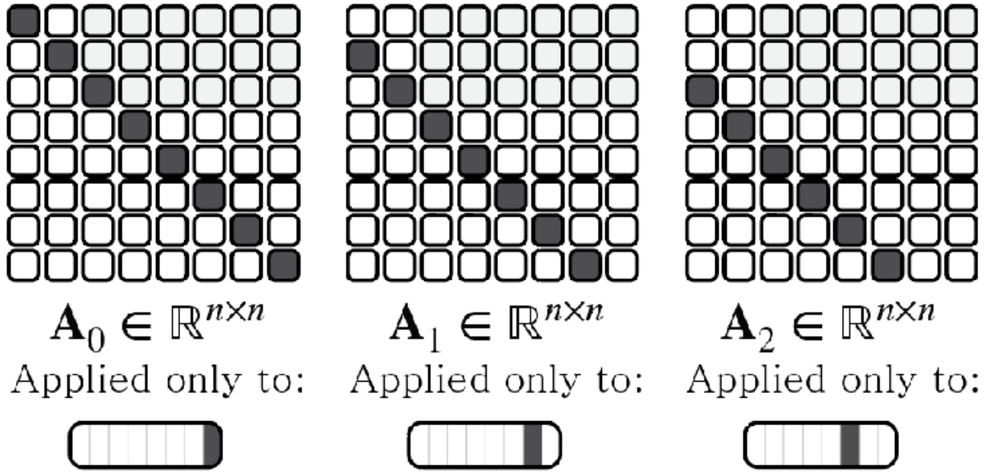
Toeplitz sequence mixing matrix

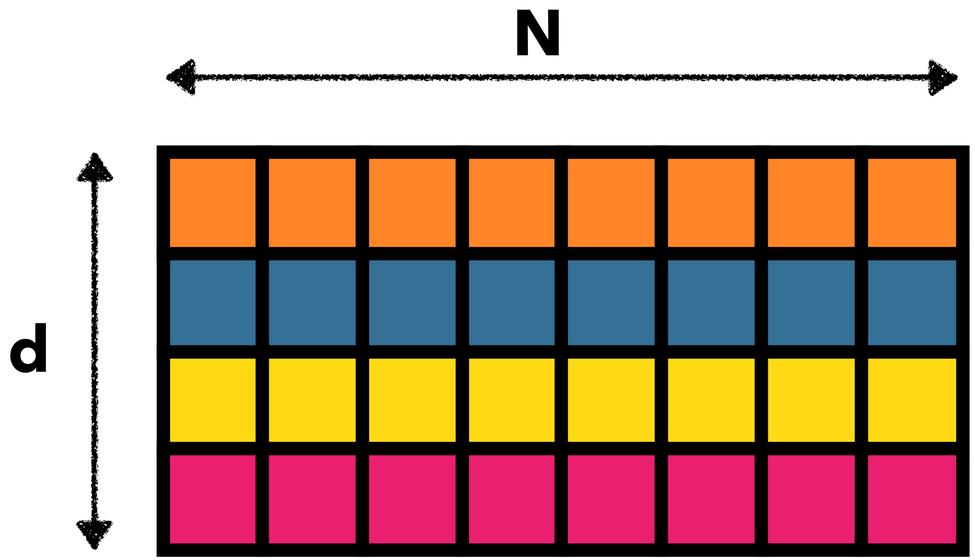
A token can look back *m* tokens only if all tokens look back *m* tokens.
With *m* = *four* below:



Gated convolutions and recall

Luckily, the convolutional LMs apply a unique convolution to each dimension of the d-dimensional input!

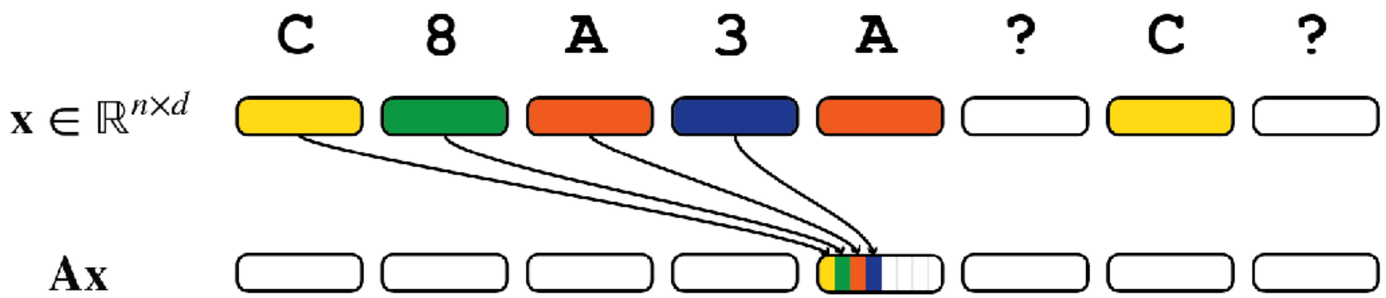




Thus, they can support **multiple token-to-token comparisons!**

Gated convolutions and recall

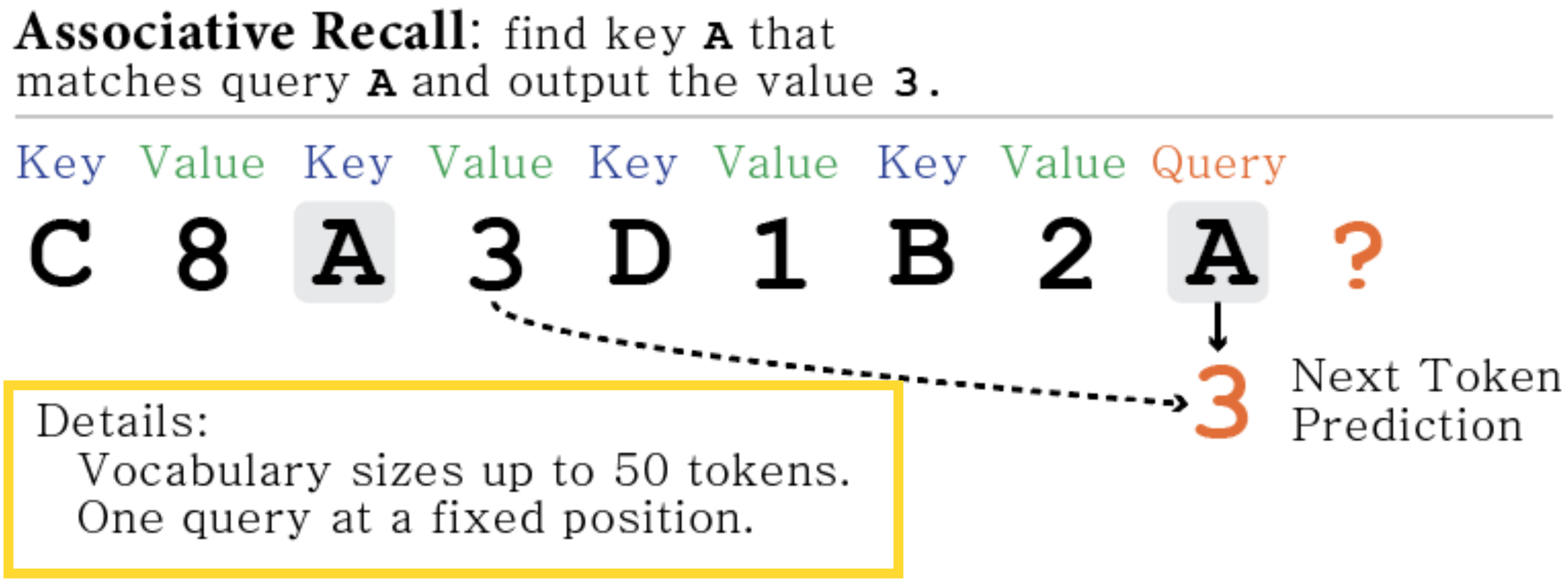
To support all $\mathcal{O}(n^2)$ token interactions, we need to perform all shifts.



Dimensionality would need to grow linearly in sequence length to store a fully copy of the input in a single embedding!

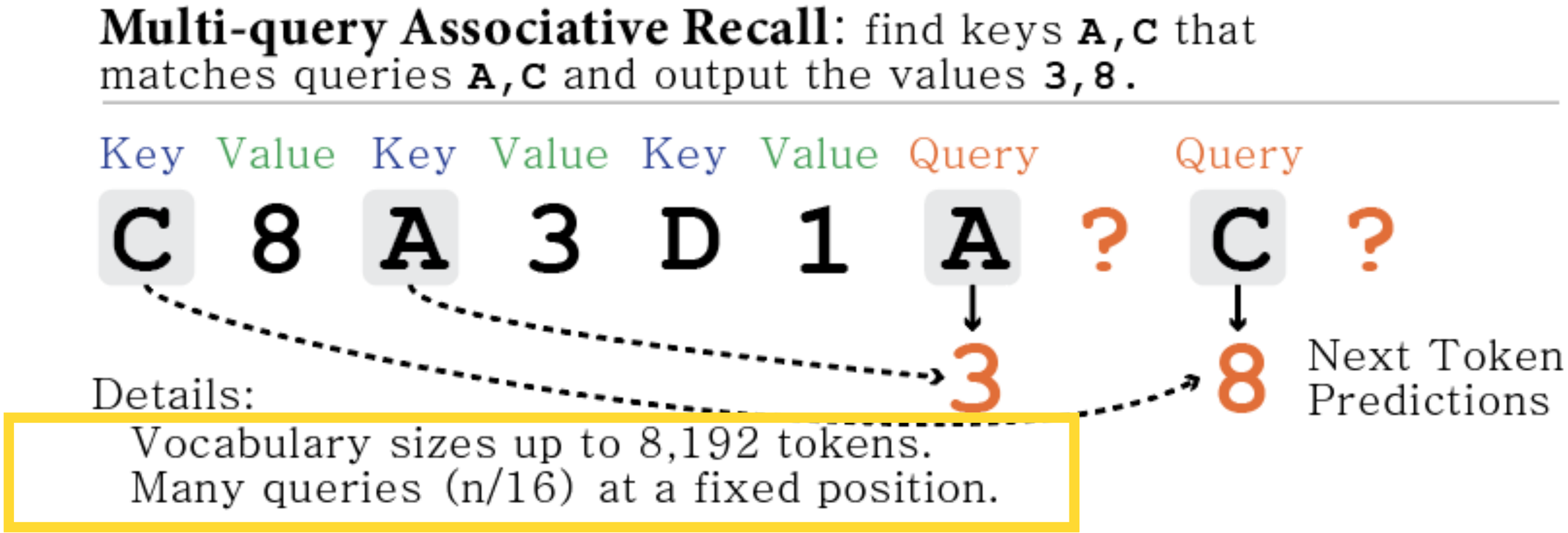
We devise an improved formalization of the recall problem.

Explaining the gap.



A few convolution shifts (small d) are sufficient for this formalization!

Recall could be needed at arbitrary positions. The key-value mappings occur one-to-few times in a sequence.



Does MQAR better capture what's going on in language modeling?

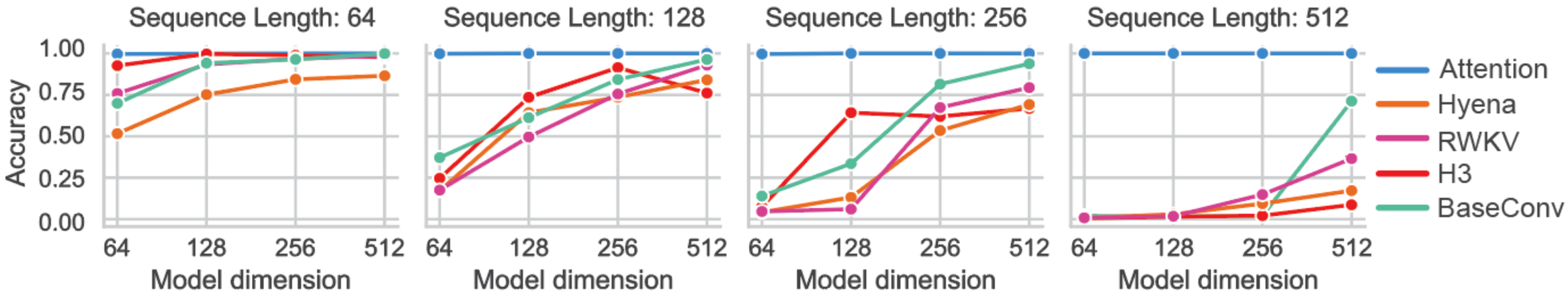
We measured MQAR quality a function of *model dimension* and *sequence length*.

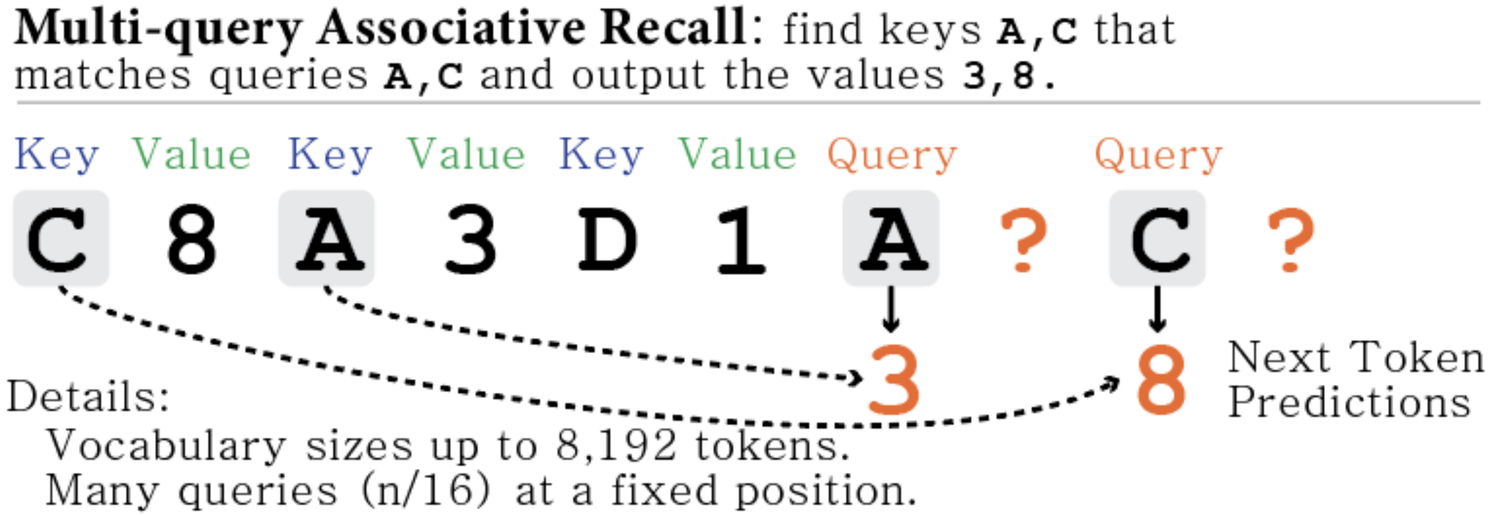
Yes! The trends on the MQAR synthetic correlate with the results we saw earlier.

We measured MQAR quality a function of *model dimension* and *sequence length*.

Attention requires only a constant model dimension to solve MQAR.

Gated-convolutions require scaling the model parameters with sequence length.

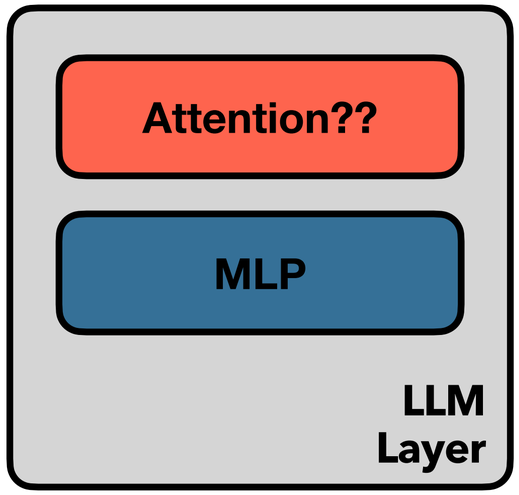


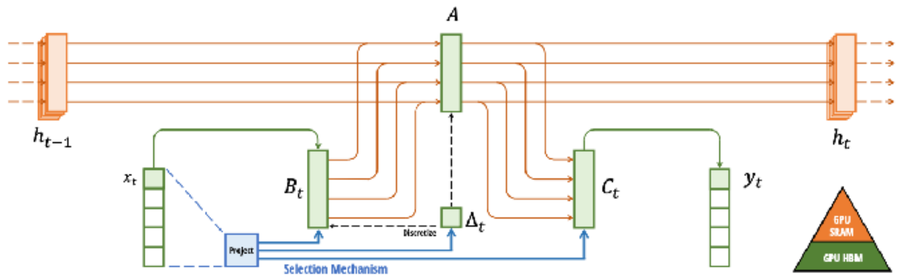


MQAR has already seen wide adoption in designing the next wave of efficient models! Try it out: https://github.com/HazyResearch/zoology

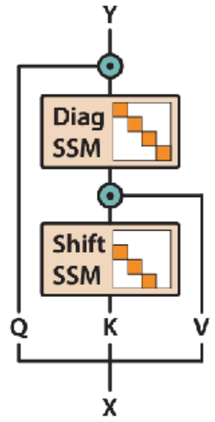
We need theory to reason about the massive landscape in a systematic way.

The architecture landscape is massive!

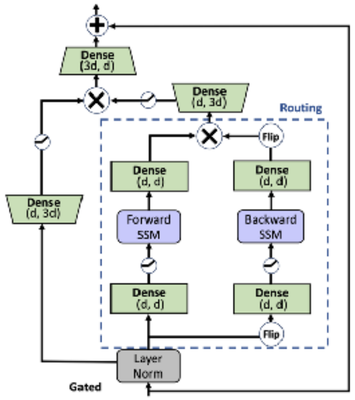




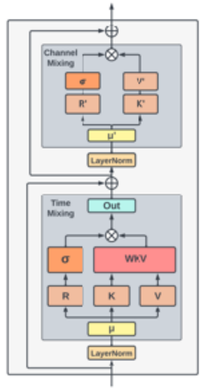
Mamba Layer



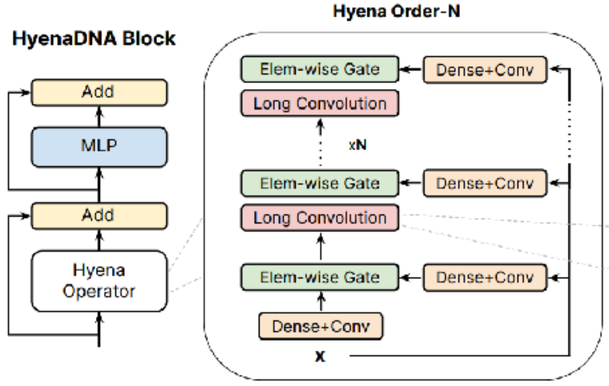
H3 Layer



BiGS Layer



RWKV Layer



Hyena Layer

S4 [Gu et al.]**, DSS** [Gupta], **GSS** [Mehta et al.], **S4D** [Gu et al.], **Liquid S4** [Hasani et al.], **H3** [Fu et al.], **S5** [Smith et al.] **BIGS** [Wang et al.], **Hyena** [Poli et al.], **RWKV** [Peng et al.], **RetNet** [Sun et al.], **M2** [Fu et al.], **Mamba** [Gu et al.], **Based** [Arora et al.,], **GLA** [Yang et al.], **GateLoop** [Kastch et al.], **Hawk/Griffin** [De et al.], **Transformers are RNNs** [Katharopoulos et al.]

We can write our efficient models as polynomials

Consider two vectors $u, v \in \mathbb{R}^{N \times d}$

$$u = [u_0, u_1, \ldots, u_{N-1}]$$
$$v = [v_0, v_1, \ldots, v_{N-1}]$$

Hadamard product computes:

$$u \odot v = [u_0 v_0, u_1 v_1, \ldots, u_{N-1} v_{N-1}]$$

Each output is a degree 2 polynomial.

💡 We can write our efficient models as polynomials

Consider two vectors $u, v \in \mathbb{R}^{N \times d}$

$$u = [u_0, u_1, \ldots, u_{N-1}]$$
$$v = [v_0, v_1, \ldots, v_{N-1}]$$

Hadamard product computes:

$$u \odot v = [u_0 v_0, u_1 v_1, \ldots, u_{N-1} v_{N-1}]$$

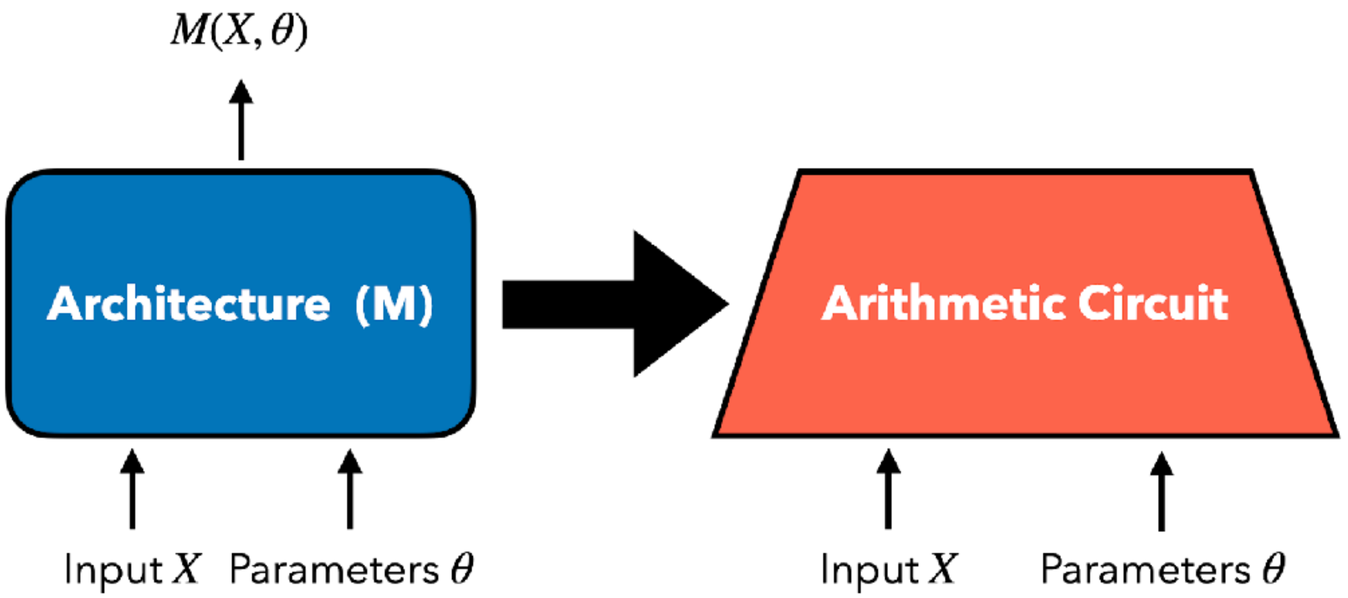
Each output is a degree 2 polynomial.

Convolution computes:

$$(u * v)[i] = \sum_{j=0}^{i} u_{i-j} v_j$$

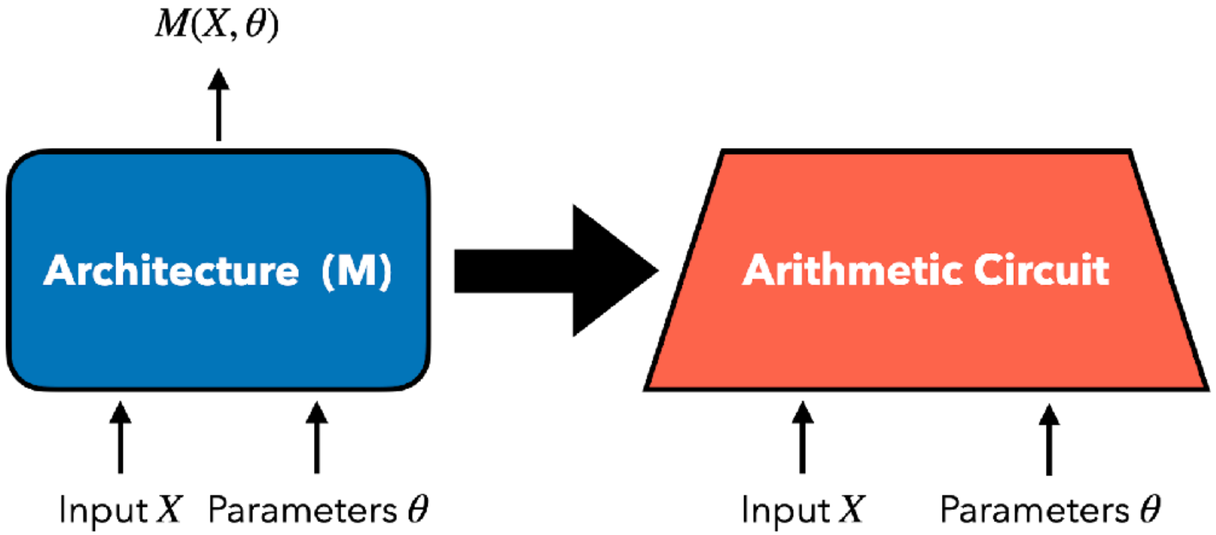
Each output is a degree 1 polynomial if v is “fixed”.

We *unify* the architectures using arithmetic circuit complexity.



Reminders

Arithmetic circuits are directed acyclic graphs, where each node is a linear (+) or multiplication (x) operation between two of the input variables.



Arithmetic circuits compute Nd polynomial outputs from the Nd input variables.

Bürgisser et al., *Algebraic Complexity Theory*, 1997.

Theorem (Arithmetic circuit equivalency): *For every low-depth arithmetic circuit of size s, depth Δ, that takes $u \in \mathbb{R}^{N \times d}$ as input, there is an equivalent BaseConv operator that uses $\tilde{O}(s\Delta)$ parameters and $\tilde{O}(\Delta)$ layers.*

$$Y = \underbrace{(uW + b_1)}_{\text{Linear map}} \odot \underbrace{(u * h + b_2)}_{\text{Convolution}}$$

BaseConv Layer

BaseConv takes input $u \in \mathbb{R}^{N \times d}$ and is defined with $h \in \mathbb{R}^{N \times d}$ learnable filters, $W \in \mathbb{R}^{d \times d}$ linear projection, $b_1, b_2 \in \mathbb{R}^{N \times d}$ bias terms.

We distill the zoo of architectures into a *canonical representation*, BaseConv.

S4 [Gu et al.], DSS [Gupta], GSS [Mehta et al.], S4D [Gu et al.], Liquid S4 [Hasani et al.], H3 [Fu et al.], S5 [Smith et al.] BIGS [Wang et al.], Hyena [Poli et al.], RWKV [Peng et al.], RetNet [Sun et al.], M2 [Fu et al.], Based [Arora et al.,], GLA [Yang et al.], GateLoop [Kastch et al.], Hawk/ Griffin [De et al.], Transformers are RNNs [Katharopoulos et al.], …

$$Y = \underbrace{(uW + b_1)}_{\textit{Linear map}} \odot \underbrace{(u * h + b_2)}_{\textit{Convolution}}$$

BaseConv Layer

Theorem (Arithmetic circuit equivalency): *For every low-depth arithmetic circuit of size s, depth Δ, that takes $u \in \mathbb{R}^{N \times d}$ as input, there is an equivalent BaseConv operator that uses $\tilde{O}(s\Delta)$ parameters and $\tilde{O}(\Delta)$ layers.*

Invoking parallel binary search, we build a parallel arithmetic circuit for MQAR, which has $O(\log N)$ depth. So $\tilde{O}(Nd)$ parameter and $\tilde{O}(1)$ layer BaseConv can solve MQAR.

Parallel Binary Search

SELIM G. AKL AND HENK MEIJER

Abstract—Two arrays of numbers sorted in nondecreasing order are given: an array A of size n and an array B of size m, where $n < m$. It is required to determine, for every element of A, the smallest element of B (if one exists) that is larger than or equal to it. We show how to $\left(\log m \log n\right)$

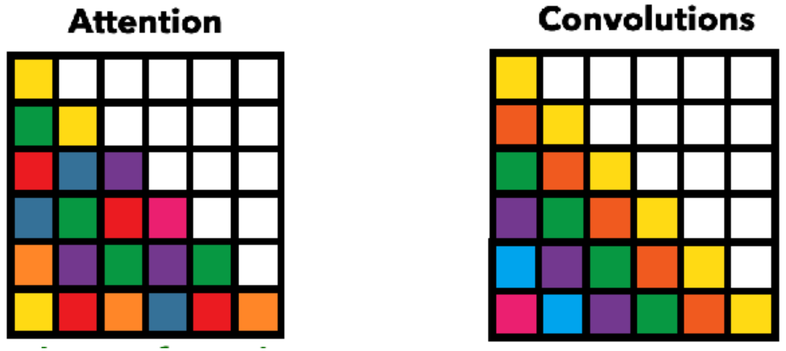
Akl and Meijer, IEEE Transactions on Parallel and Distributed Systems,1990.

Theorem (BaseConv lower bound): Regardless of how x is encoded, with $d \leq 2^{(\log N)^{1-\epsilon}}$, a BaseConv model (where each parameter takes $O(\log N)$ bits) *requires* $\Omega(\epsilon \log \log N)$ layers to solve MQAR.

💡 **Theorem (BaseConv lower bound):** Regardless of how x is encoded, with $d \leq 2^{(\log N)^{1-\epsilon}}$, a BaseConv model (where each parameter takes $O(\log N)$ bits) *requires* $\Omega(\epsilon \log \log N)$ layers to solve MQAR.

We need something else...

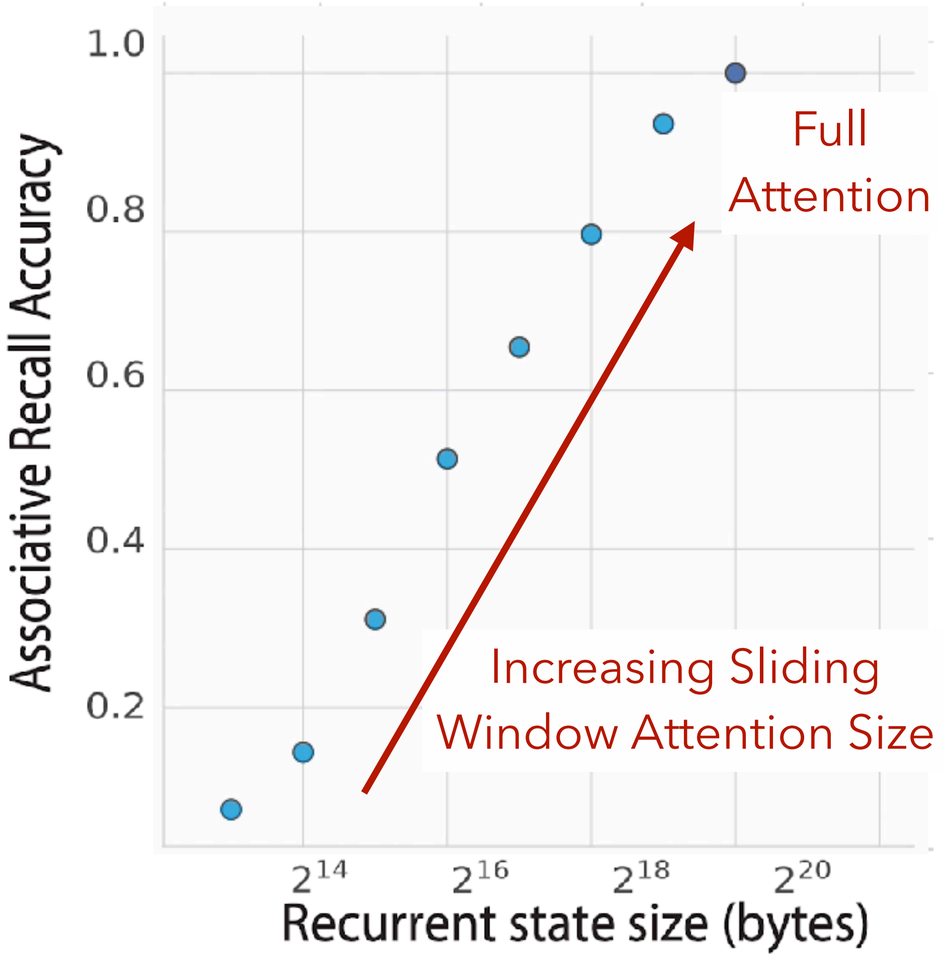
💡 Theory shows that **input-dependent** sequence mixing (like attention) is important for recall.



Consider the ‘simplest’ input-dependent and sub-quadratic mixer: ‘sliding window attention’

✗ Limited memory for long range recall

✔ Precise local token shifts and comparison



Consider the ‘simplest’ input-dependent and sub-quadratic mixer: ‘sliding window attention’

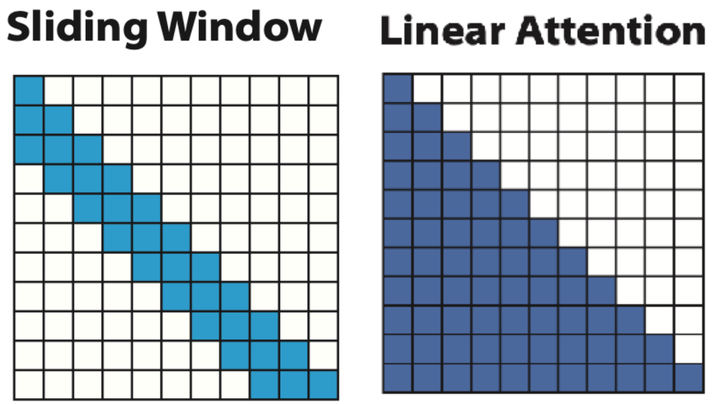
🤔

Can we expand the Pareto-frontier of this tradeoff space?

Maybe linear attention can help us?

💡 Maybe linear attention can help us?

✔ “Globally” approximates standard attention



Maybe linear attention can help us?

“Globally” approximates standard attention

Uses input-dependent mixing like attention

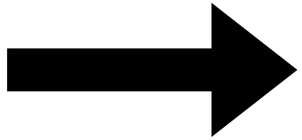
💡 Maybe linear attention can help us?

- ✓ “Globally” approximates standard attention
- ✓ Uses input-dependent mixing like attention
- ✓ Still sub-quadratic training and O(1) inference

💡 Let's mimic attention with linear attention

Attention

$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times d}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times d}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \text{Softmax}(QK^T)V$$



Linear attention

$$Q = W_q u, \quad Q \in \mathbb{R}^{N \times f}$$
$$K = W_k u, \quad K \in \mathbb{R}^{N \times f}$$
$$V = W_v u, \quad V \in \mathbb{R}^{N \times d}$$
$$y = \phi(Q)\phi(K^T)V$$

- Feature map $\phi(\cdot)$
- Feature dim. f

💡 Let's use $\phi(\cdot)$ to approximate the attention $\exp(\cdot)$

Taylor approximation for the exponential function:

$$\exp(x) = 1 + x + \frac{x}{2!} + \ldots$$

Keles et al., *On The Computational Complexity of Self-Attention*, 2022.

Zhang et al., *The Hedgehog & the Porcupine: Expressive Linear Attentions with Softmax Mimicry*, ICLR 2024.

💡 Let's use $\phi(\cdot)$ to approximate the attention $\exp(\cdot)$

Taylor approximation for the exponential function:

$$\exp(x) = 1 + x + \frac{x}{2!} + \dots$$

Let our linear attention feature map $\phi(\cdot)$ be, for outer product $\otimes$:

$$\phi(q) = [1, q, q \otimes q/\sqrt{2}, \dots]$$
$$\phi(k) = [1, k, k \otimes k/\sqrt{2}, \dots]$$

$$\Rightarrow \exp(qk) = \phi(q)\phi(k) = [1, qk, (q \otimes q)(k \otimes k)/2, \dots]$$

Need infinite terms to exactly represent $\exp(\cdot)$

Zhang et al., *The Hedgehog & the Porcupine: Expressive Linear Attentions with Softmax Mimicry*, ICLR 2024.

💡 Let's use $\phi(\cdot)$ to approximate the attention $\exp(\cdot)$

Let our linear attention feature map $\phi(\cdot)$ be, for outer product $\otimes$:

$$\phi(q) = [1, q, q \otimes q/\sqrt{2}]$$

$$\phi(k) = [1, k, k \otimes k/\sqrt{2}]$$

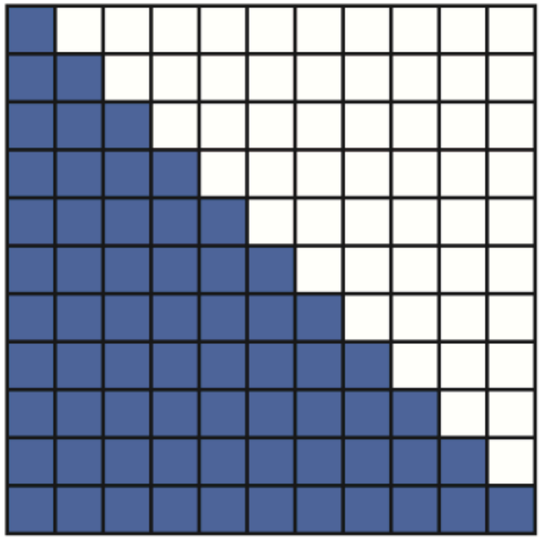
$$\Rightarrow \quad \exp(qk) \approx \phi(q)\phi(k) = [1, qk, (q \otimes q)(k \otimes k)/2]$$

2nd order Taylor polynomial approximation is empirically effective

Zhang et al., *The Hedgehog & the Porcupine: Expressive Linear Attentions with Softmax Mimicry*, ICLR 2024.

Combine local and global approximations!

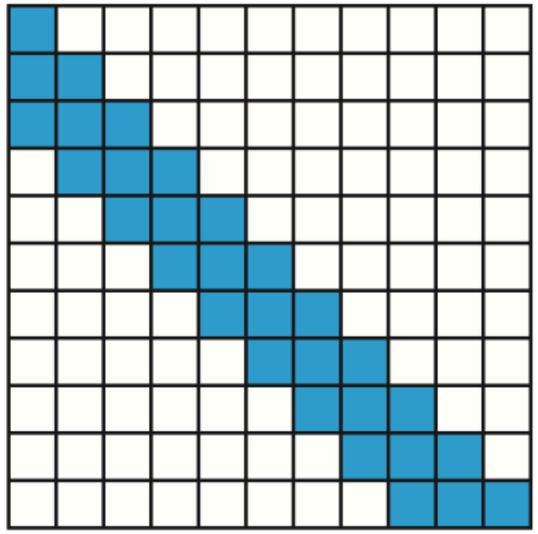
Linear Attention



✔ Taylor approximation provides large memory for recall

✗ Precise local token shifts and comparison

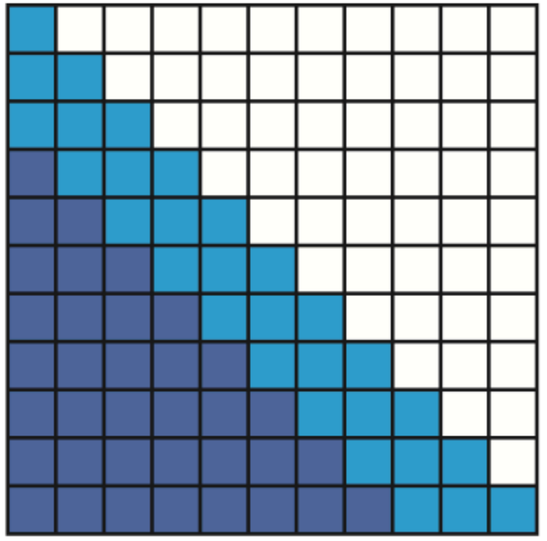
Sliding Window



✗ Limited memory for long range recall

✔ Precise local token shifts and comparison

Based

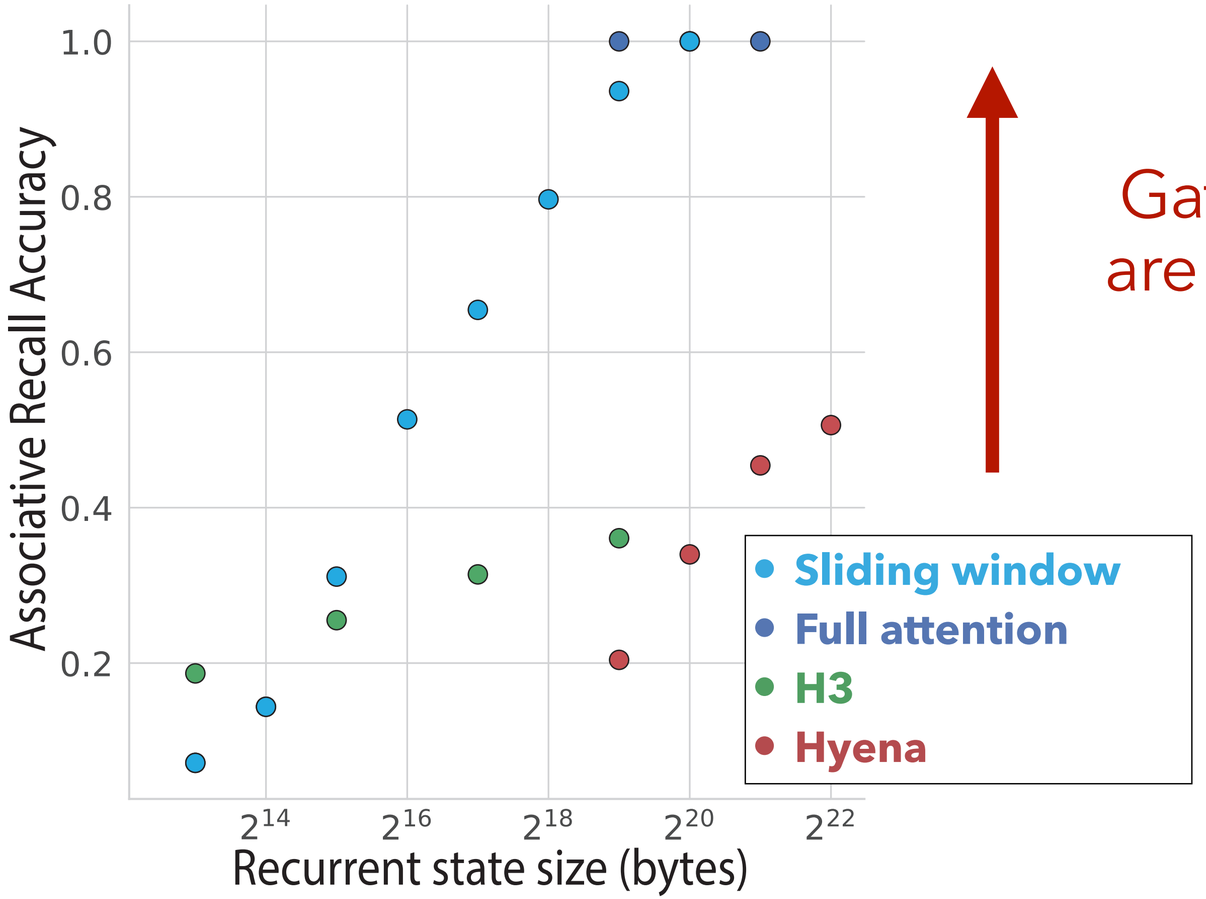


✔ Taylor approximation provides large memory for recall

✔ Precise local token shifts and comparison

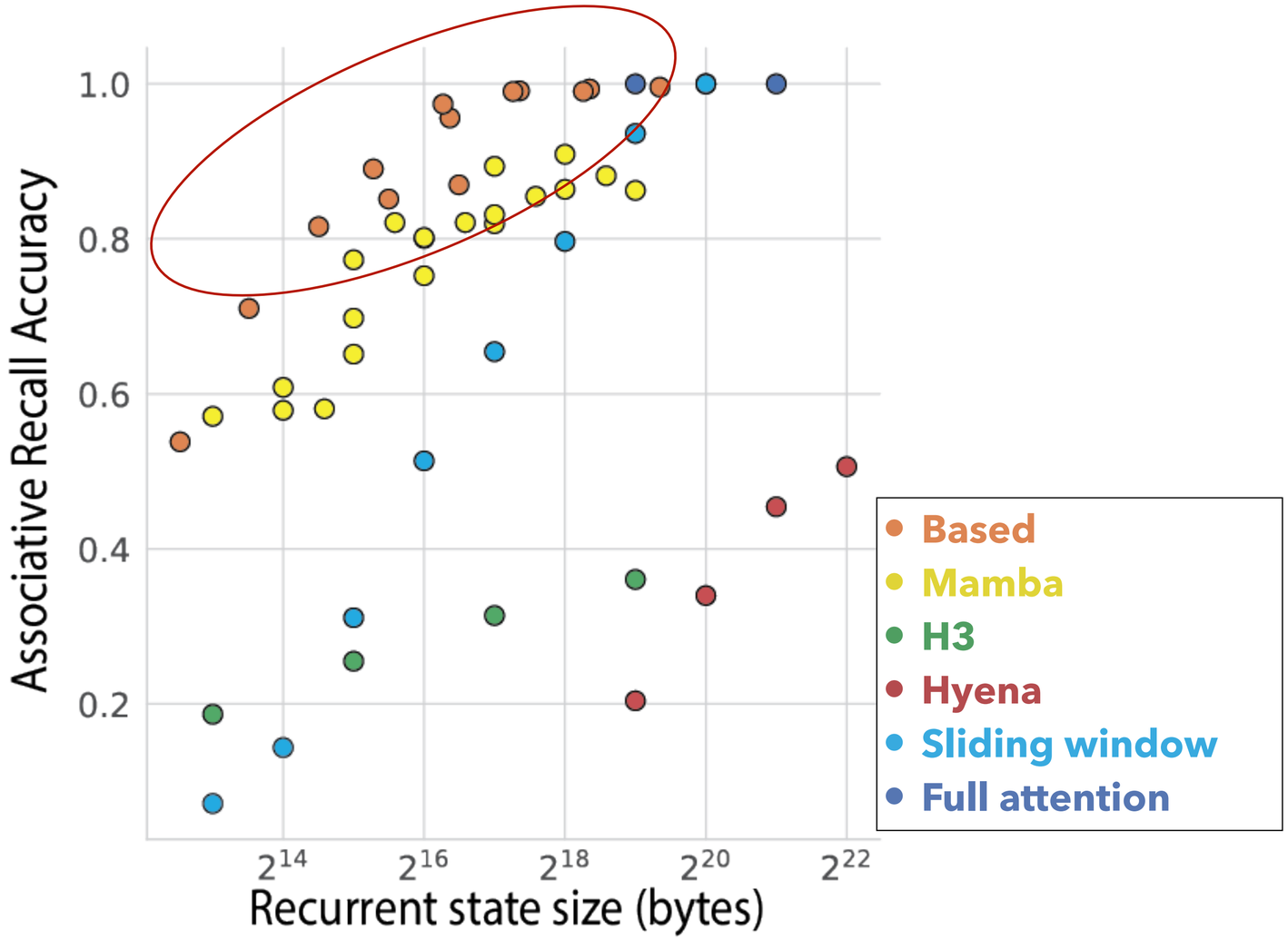
Explaining the gap

We measured MQAR quality a function of the amount of the recurrent memory/state



Gated convolutions are below the Pareto frontier

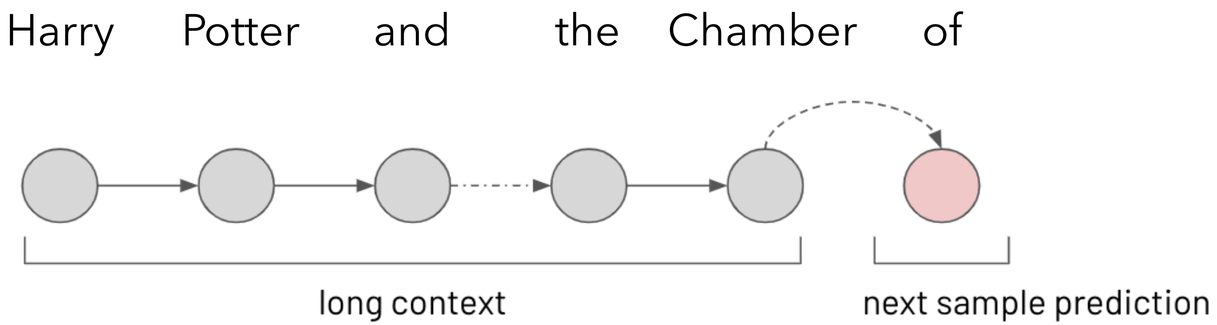
BASED expands the Pareto frontier of the tradeoff space!!!



🤔 No free lunch!

Lower bound for recurrent state memory and MQAR.

Theorem (Recurrent): Any recurrent model depending causally on the input $u \in \{0,1\}^{N \times d}$ requires $\Omega(N)$-bits in state size to solve AR/MQAR.



Next token prediction.
One pass streaming setting.

Lower bound for recurrent architectures and MQAR.

Theorem (Recurrent): Any recurrent model depending causally on the input $u \in \{0,1\}^{N \times d}$ requires $\Omega(N)$-bits in state size to solve AR/MQAR.

We reduce MQAR to the index problem and use a known lower bound for the index problem.

1. Two parties, Alice and Bob.
2. Alice has a length n string $x \in \{0,1\}^n$ and Bob has an index $i \in [n]$.
3. Alice passes Bob a single message and Bob needs to output the i-th entry x_i.

Jayram et al., 2008 proves the $\Omega(n)$ bits communication are required for the index problem for a length n string.

Thathachar S Jayram, Ravi Kumar, and Dandapani Sivakumar. The one-way communication complexity of hamming distance. *Theory of Computing*, 2008.

💡 Let's use $\phi(\cdot)$ to approximate the attention $\exp(\cdot)$

Let our linear attention feature map $\phi(\cdot)$ be, for outer product $\otimes$:

$$\phi(q) = [1, q, q \otimes q/\sqrt{2}]$$

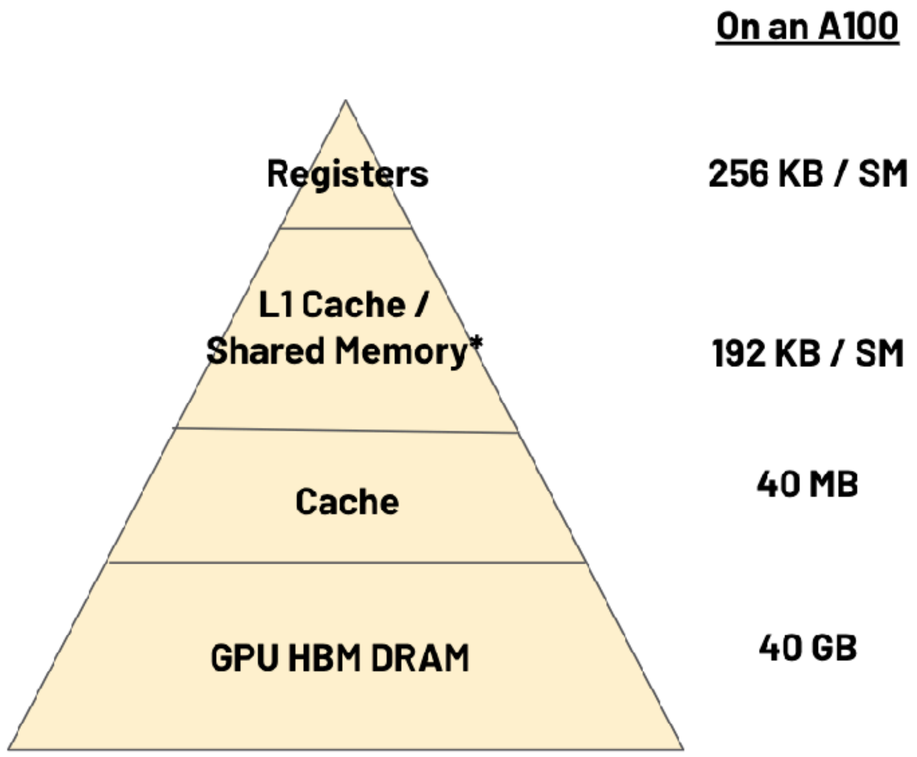
$$\phi(k) = [1, k, k \otimes k/\sqrt{2}]$$

$$\Rightarrow \quad \exp(qk) \approx \phi(q)\phi(k) \approx [1, qk, (q \otimes q)(k \otimes k)/2]$$

Recall $\phi(\cdot) : \mathbb{R}^{N \times f} \rightarrow \mathbb{R}^{N \times (1+f+f^2)}$

We use relatively large state sizes (in a hardware efficient way)

We develop hardware IO-efficient algorithms to retain efficiency.



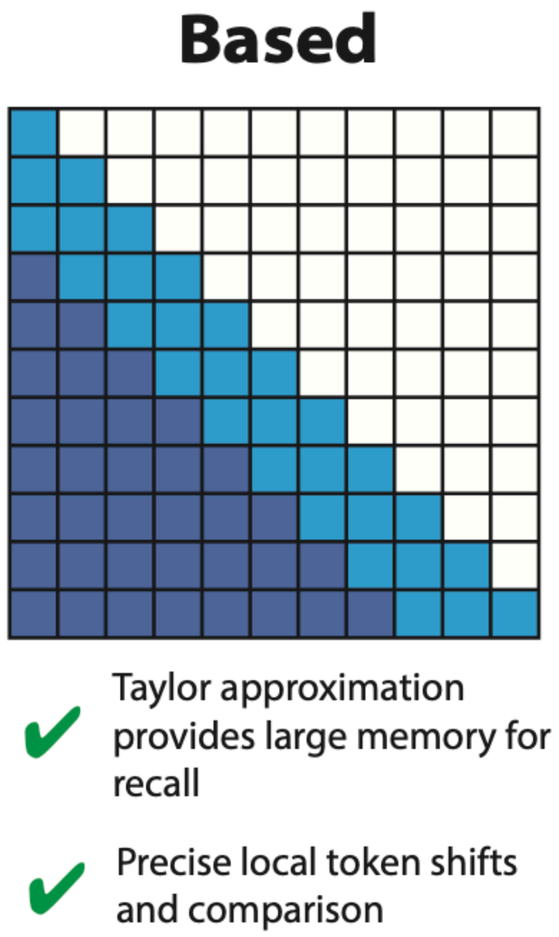
GPU Memory Hierarchy

Downstream results and efficiency

Model	Generation efficiency Tokens/ms	6 Recall-Intensive Tasks (Accuracy)	7 General Language Tasks (Accuracy)
Transformer	0.99	51.4	52.9
Mamba	25.69	36.8	56.6
Based	24.28	42.6	53.8

All LMs are trained on the same 50Bn tokens of the Pile at the 1.3Bn parameter scale.

We’re super excited about BASED! Try it out: https://github.com/HazyResearch/based



- **Large in-context learning and associative recall improvements over prior strong efficient architectures (e.g., Mamba)**
- **Spotlight (top 3.5% of 10K papers) at ICML 2024**
- **Oral (top 5 papers) at ICML ES-FoMo**

Talk outline

Efficient architectures. *What are prevailing approaches?*

Quality gaps. *How do sub-quadratic alternatives compare to attention?*

Explaining the gap. *Using synthetics and theory to explain the tradeoffs of prior attention alternatives.*

Bridging the gap. Using our insights to build new architectures that extend the Pareto frontier of the quality-efficiency tradeoff space.

How theory informs our design of “good” efficient LMs.

Input-dependent sequence mixing (like attention) is important for recall.

There are fundamental memory and recall-quality tradeoffs.

Lower bound for recurrent state memory and MQAR.

Theorem (Recurrent): Any recurrent model depending causally on the input $u \in \{0,1\}^{N \times d}$ requires $\Omega(N)$-bits in state size to solve AR/MQAR.

🤔 **Our work begs the question: Can we rely on O(1) memory recurrent LMs for in-context learning at all?**

💡 This makes recurrent models brittle with respect to data ordering.

Order 1

Galileo Galilei

198 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

"Galileo" redirects here. For other uses, see Galileo (disambiguation) and Galileo Galilei (disambiguation).

Galileo di Vincenzo Bonaiuti de' Galilei (15 February 1564 – 8 January 1642), commonly referred to as **Galileo Galilei** (/ˌɡælɪˈleɪoʊ ˌɡælɪˈleɪ/ *GAL-il-AY-oh GAL-il-AY*, US also /ˌɡælɪˈliːoʊ -/ *GAL-il-EE-oh* -, Italian: [ɡaliˈlɛːo ɡaliˈlɛːi]) or simply **Galileo**, was an Italian astronomer, physicist and engineer, sometimes described as a polymath. He was born in the city of Pisa, then part of the Duchy of Florence.[3] Galileo has been called the father of observational astronomy,[4] modern-era classical physics,[5] the scientific method,[6] and modern science.[7]

Galileo studied speed and velocity, gravity and free fall, the principle of relativity, inertia, projectile motion and also worked in applied science and technology, describing the properties of the pendulum and "hydrostatic balances". He was one of the earliest Renaissance developers of the thermoscope[8] and the inventor of

Galileo Galilei

1636 portrait

When did Galileo move to Florence?

Order 2

When did Galileo move to Florence?

Galileo Galilei

198 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

"Galileo" redirects here. For other uses, see Galileo (disambiguation) and Galileo Galilei (disambiguation).

Galileo di Vincenzo Bonaiuti de' Galilei (15 February 1564 – 8 January 1642), commonly referred to as **Galileo Galilei** (/ˌɡælɪˈleɪoʊ ˌɡælɪˈleɪ/ *GAL-il-AY-oh GAL-il-AY*, US also /ˌɡælɪˈliːoʊ -/ *GAL-il-EE-oh* -, Italian: [ɡaliˈlɛːo ɡaliˈlɛːi]) or simply **Galileo**, was an Italian astronomer, physicist and engineer, sometimes described as a polymath. He was born in the city of Pisa, then part of the Duchy of Florence.[3] Galileo has been called the father of observational astronomy,[4] modern-era classical physics,[5] the scientific method,[6] and modern science.[7]

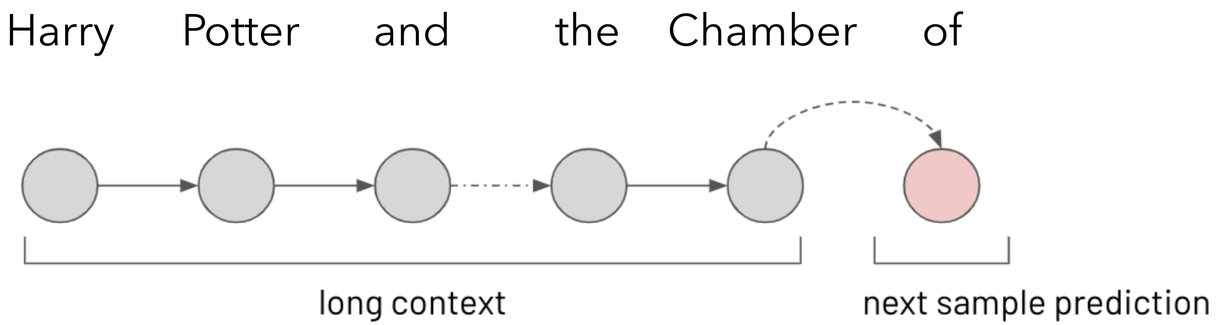
Galileo studied speed and velocity, gravity and free fall, the principle of relativity, inertia, projectile motion and also worked in applied science and technology, describing the properties of the pendulum and "hydrostatic balances". He was one of the earliest Renaissance developers of the thermoscope[8] and the inventor of

Galileo Galilei

1636 portrait

Lower bound for recurrent state memory and MQAR.

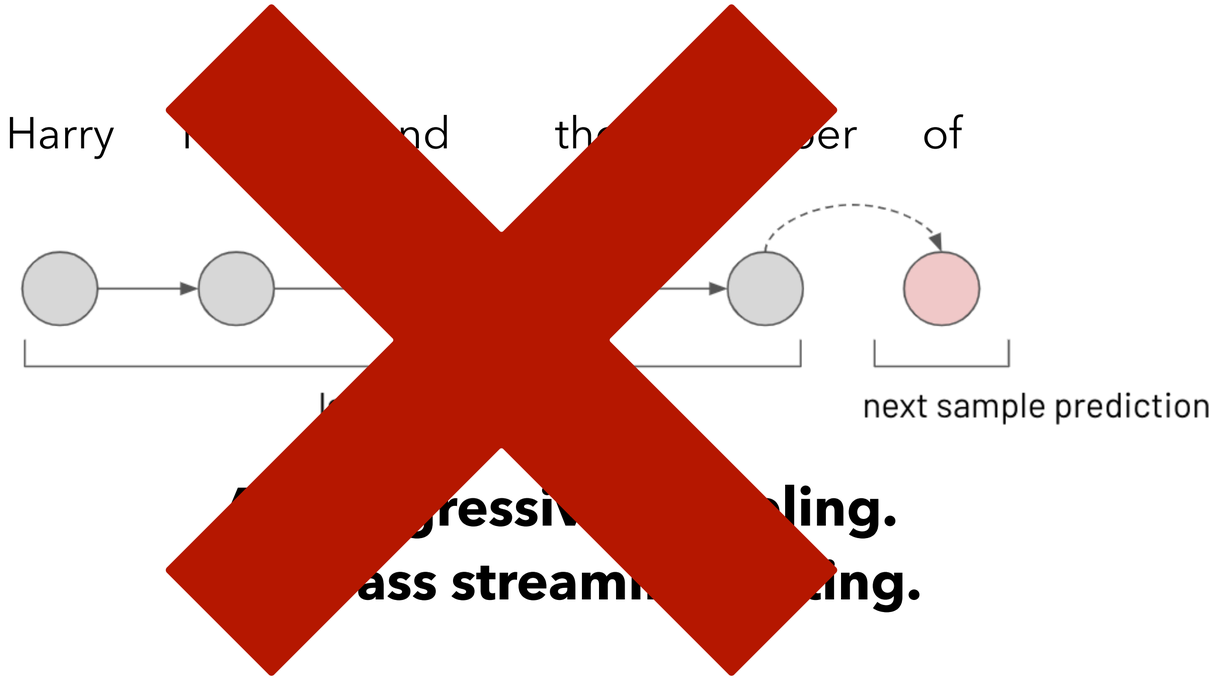
Theorem (Recurrent): Any recurrent model depending causally on the input $u \in \{0,1\}^{N \times d}$ requires $\Omega(N)$-bits in state size to solve AR/MQAR.



Autoregressive modeling.
One pass streaming setting.

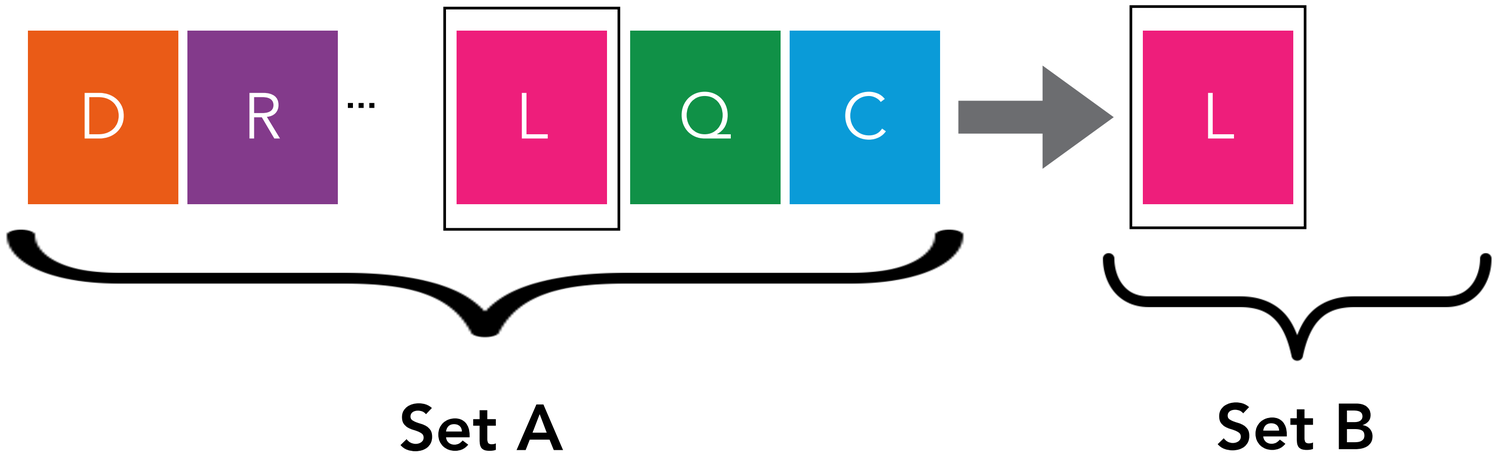
Lower bound for recurrent state memory and MQAR.

Theorem (Recurrent): Any recurrent model depending causally on the input $u \in \{0,1\}^{N \times d}$ requires $\Omega(N)$-bits in state size to solve AR/MQAR.



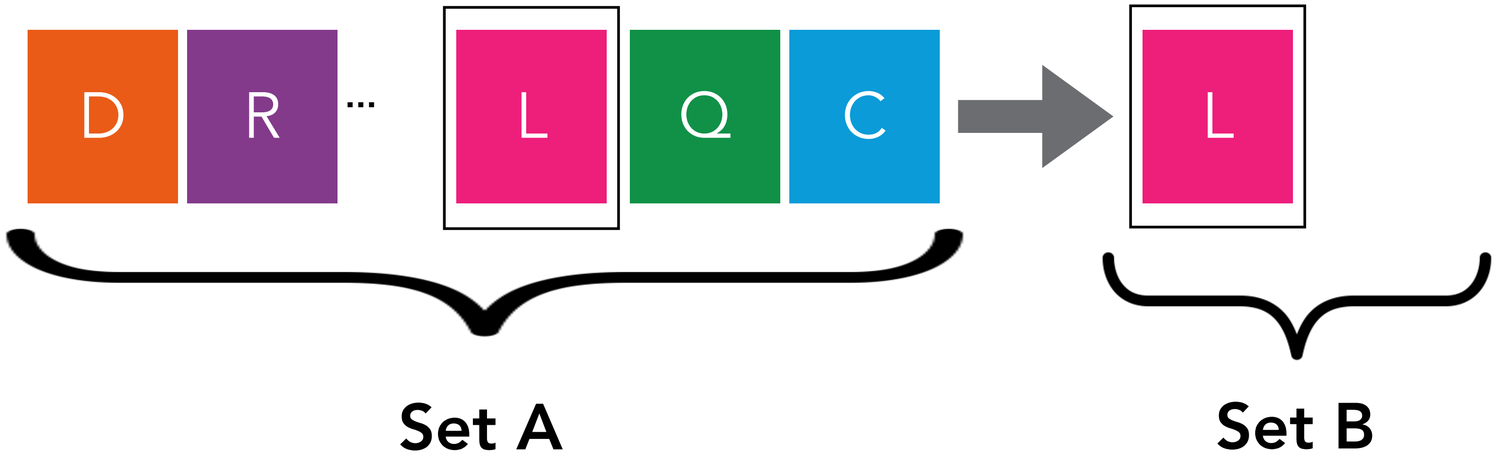
💡 Let's abstract this idea!

I give you two sets of elements and you need to tell me if they intersect?

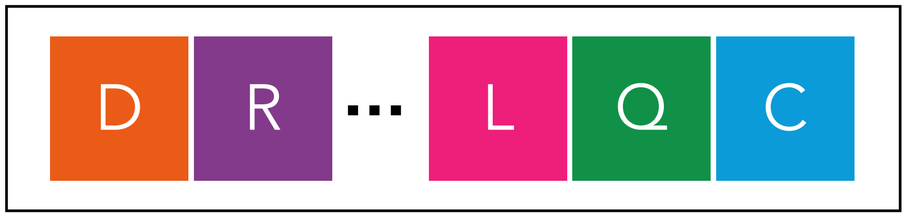


Hemaspaandra, *SIGACT News Complexity Theory Column 67*.

For **autoregressive models**, the amount of memory we need depends on the **size of the first set** in the sequence.

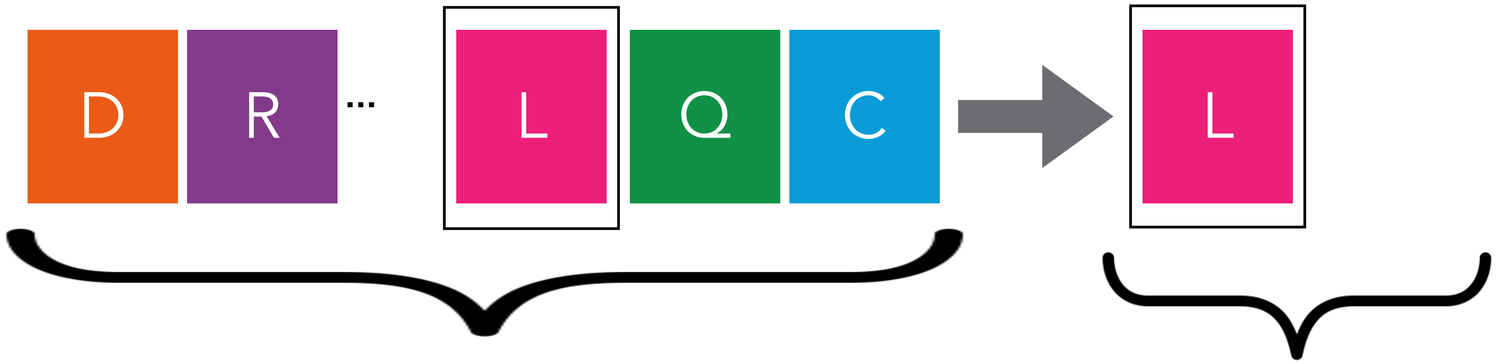


Memory needed if Set A comes first

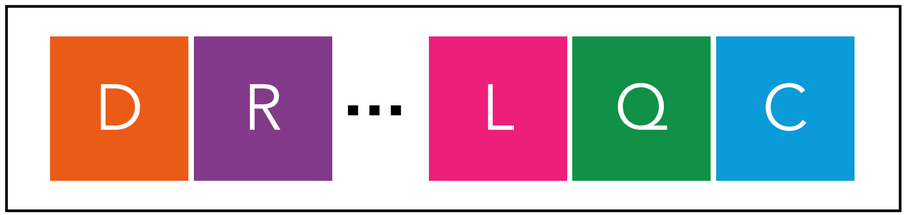


Memory needed if Set B comes first

For **non-causal models**, the amount of memory we need depends on the **min(|A|, |B|)!**

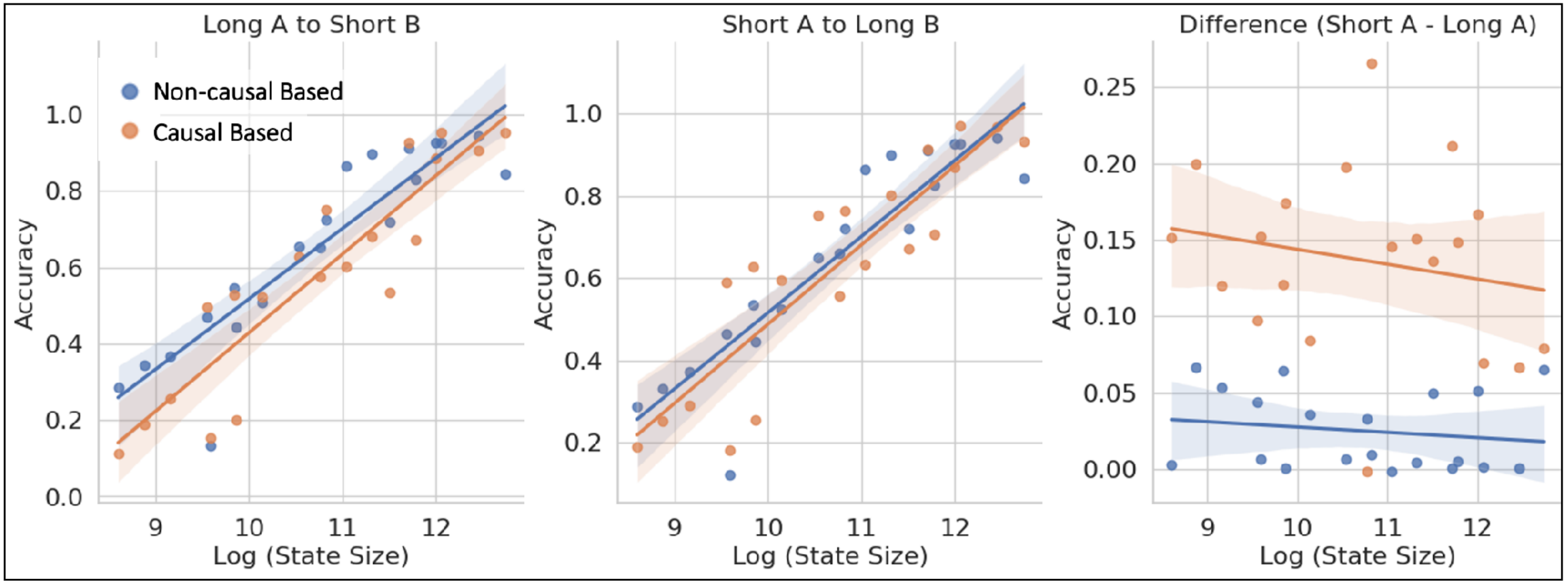


Memory needed if Set A comes first



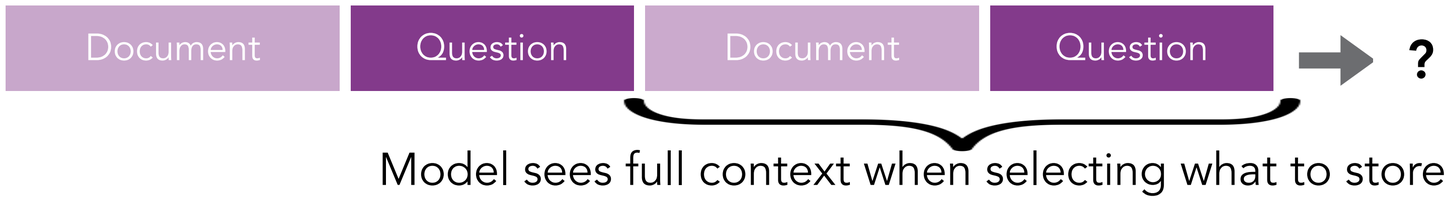
Memory needed if Set B comes first

We find non-causal models perform better than causal models on a synthetic version of this task when the first set (A) is large

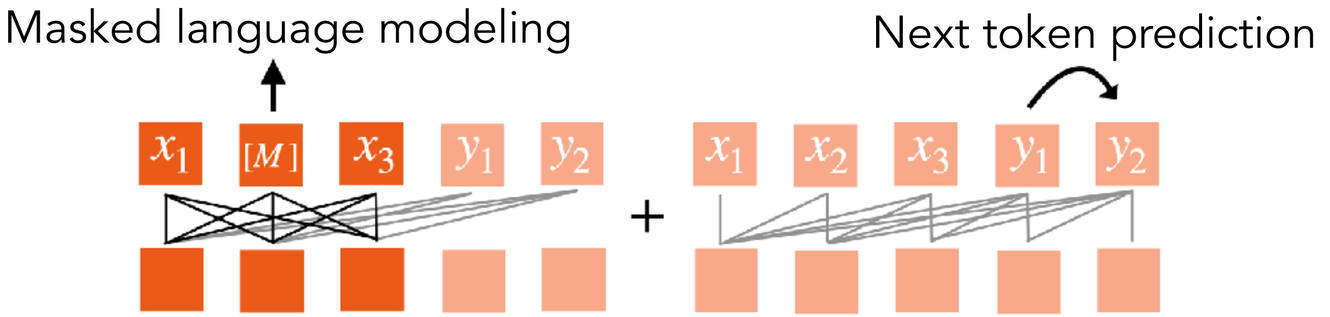


We can make **multiple passes** over the data to better select the information to store in the fixed recurrent state.

Just read twice prompting
Show the model the input twice



Just read twice linear attention
Non-causal **encoding** of the context plus causal **decoding**



Our new architecture **JRT-RNN** achieves **96% the quality of Transformers**, while being **19.2x faster** by using our hardware-efficient algorithm!

Model	Generation efficiency Tokens/ms	6 Recall-Intensive Tasks (Accuracy)	7 General Language Tasks (Accuracy)
Transformer	0.99	51.4	52.9
Mamba	25.69	36.8	56.6
Based	24.28	42.6	53.8
JRT-RNN	24.28	49.5	54.1

Collaborators

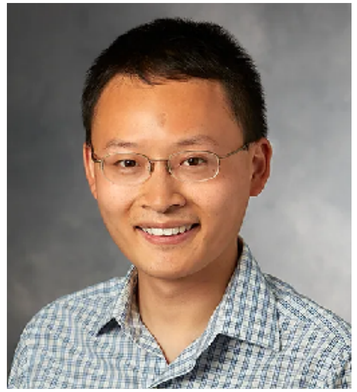
Sabri Eyuboglu

Michael Zhang

Michael Poli

Silas Alberti

Atri Rudra



James Zou

Aman Timalsina

Isys Johnson

Dylan Zinsley

Chris Ré

Thank you!

Blogpost on the theoretical results: https://hazyresearch.stanford.edu/blog/2024-06-22-ac

Paper references:

- Arora*, Eyuboglu* et al., Zoology: Measuring Recall in Input Dependent Models. ICLR 2023.
- Arora*, Eyuboglu*, Zhang* et al., Simple linear attention language models balance the recall-throughput tradeoff. ICML 2024. (Spotlight)
- Arora et al., Just read twice: Closing the recall gap for recurrent language models. 2024.