Simple linear attention language models balance the recall-throughput tradeoff

BASED: global memory, precise local mixing, efficient execution

Paper by Simran Arora, Sabri Eyuboglu, Michael Zhang,
Aman Timalsina, Silas Alberti, Dylan Zinsley,
James Zou, Atri Rudra, and Christopher Ré

arXiv:2402.18668v2 (7 March 2025)
Paper discussion / 10 October 2026

arxiv.org/abs/2402.18668v2

Similar perplexity can hide a large recall gap

Overall language modeling

1.3B scale, 50B training tokens:

Model	Pile PPL ↓
Transformer++	6.28
Mamba	6.28
BASED	6.30

Extracting facts from the input

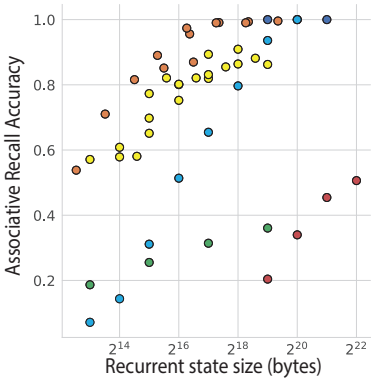
FDA document extraction accuracy:

Model	Accuracy ↑
Transformer++	80.47
Mamba	18.51
BASED	30.40

Question: how much input information can a small recurrent state retain?

Paper, Table 1, PDF p. 8. FDA accuracy checks whether the generated text contains the answer.

Both state capacity and its use affect recall



- x: recurrent state size in bytes.
- y: MQAR recall accuracy.
- Larger states usually improve recall.
- The same budget can yield different recall across architectures.

BASED expands the empirical frontier over the configurations tested.

State size is distinct from parameter count and measured runtime.

Paper, Sec. 3.1 and Fig. 2, PDF pp. 3–4. Original plot cropped; legend restated.

MQAR isolates temporary key–value associations

A small example

```
A 4  B 3  C 6  F 1  E 2

A ?  C ?  F ?  E ?  B ?
```

Targets: 4, 6, 1, 2, 3

- Training: length 256; 4–64 key–value pairs.
- Evaluation: length 1,024; 4–256 key–value pairs.
- Vary state-related dimensions or window width per architecture.
- The evaluation also tests generalization to longer sequences.

Paper, Sec. 3.1, PDF pp. 3–4; state accounting in Appendix E.2, pp. 36–37.

Reliable arbitrary recall requires growing memory

State lower bound

Any causal recurrent model solving arbitrary associative recall requires

$$\Omega(N) \text{ bits of state in the worst case.}$$

1. Alice encodes bits $x_1, \ldots, x_N$ as pairs (j, x_j).
2. She runs the model and sends its state to Bob.
3. Bob supplies query i and recovers x_i.

A smaller state would violate the one-way Index communication lower bound.

This applies to BASED too. States are measured in finite bits.

Paper, Theorem 3.1, p. 4; Theorems F.10 and F.12, pp. 43–44. Index bound allows success probability $\geq 2/3$.

BASED combines three complementary sequence mixers

Taylor linear attention	Sliding-window attention	Short gated convolution
Global access through a fixed-size association state.	Exact softmax over recent tokens.	Width-3 local mixing with low state cost.
Capacity controlled by feature dimension.	Precise local comparisons and alignment.	Additional quality gain in the trained hybrid.

The language models mix these layer types at roughly

20% linear attention / 20% SWA / 60% convolution.

Author hypothesis: local operations form precise associations; global state carries them across distance.

Conceptual summary, not an original figure. Paper, Sec. 4, pp. 5–6; Sec. 6, p. 8; Appendix C, p. 31.

Associativity turns attention into an additive memory

Using column vectors and $K(q, k) = \phi(q)^\top \phi(k)$:

$$y_i = \frac{\sum_{j \le i} K(q_i, k_j) v_j}{\sum_{j \le i} K(q_i, k_j)} = \frac{s_i^\top \phi(q_i)}{z_i^\top \phi(q_i)}.$$

Recurrent update

$$s_i = s_{i-1} + \phi(k_i) v_i^\top, \qquad z_i = z_{i-1} + \phi(k_i).$$

- $\phi(k_i) \in \mathbb{R}^D$, $v_i \in \mathbb{R}^{d_v}$, $s_i \in \mathbb{R}^{D \times d_v}$.
- Per head: $D(d_v + 1)$ state entries; $O(Dd_v)$ work per token.
- Constant in sequence length, for fixed dimensions; normalization remains.

Paper, Eqs. (2)–(3), PDF p. 5. Vector orientation standardized; d_v denotes the value dimension.

The Taylor map expands memory without learned features

Let projected queries and keys have dimension d' and define

$$\bar{q} = q/(d')^{1/4}, \qquad \bar{k} = k/(d')^{1/4}.$$

$$\phi(\bar{q}) = \left[1,\ \bar{q},\ \frac{\mathrm{vec}(\bar{q}\bar{q}^\top)}{\sqrt{2}}\right].$$

Second-order exponential approximation

$$\phi(\bar{q})^\top \phi(\bar{k}) = 1 + \frac{q^\top k}{\sqrt{d'}} + \frac{(q^\top k)^2}{2d'}.$$

- Constant, first-order, and pairwise feature interactions.
- Polynomial expansion adds no learned weights at fixed d'.
- Larger d' increases capacity, state traffic, and projection cost.

Paper, Eq. (4), p. 5; scaling made explicit from Appendix B.1, Listing 1, pp. 20–22.

153, 273, and 320 describe different representations

Representation	Feature dimension	$d' = 16$
Symmetry-compressed	$1 + d' + d'(d'+1)/2$	153
Full outer product	$1 + d' + d'^2$	273
Kernel padding	Multiple of 64	320

- Table 6 and Appendix E.2 use the compressed dimension.
- Algorithm 2 pads the full expansion for hardware alignment.
- Actual memory also depends on heads, layers, and numeric precision.

Practical consequence

Use the implemented state shape when estimating GPU memory and bandwidth.

Paper, Algorithm 2, p. 27; Table 6, p. 35; Appendix E.2, pp. 36–37.

Small windows preserve precise local interactions

$$j \in [\max(1, i - w + 1),\ i]$$

- Standard causal softmax inside the window.
- State per head: $2\min(N, w)d_v$ entries.
- Global linear layers carry information beyond the local receptive field.

Configuration	Small	Large
Total layers	27	36
Linear layers	5	7
Window layers	5	7
Conv layers	17	22
Window size	64	16

Linear layers: no position encoding.
Window layers: Rotary encoding.

Table 7's actual configurations differ from the general 64–128-window discussion.

Paper, Sec. 4.2, p. 6; Table 7, p. 75 (columns labeled 355M and 1.4B).

Chunking separates local causality from global history

For block b, feature matrices Φ_{Q_b}, Φ_{K_b}, and causal mask M_b:

$$\underbrace{U_b}_{\text{numerator}} = \underbrace{(M_b \odot \Phi_{Q_b}\Phi_{K_b}^{\top})V_b}_{\text{within-block interactions}} + \underbrace{\Phi_{Q_b}S_{<b}}_{\text{earlier blocks}},$$

$$S_{<b} = \sum_{a<b} \Phi_{K_a}^{\top} V_a.$$

- Within a tile: parallel matrix products and explicit causal masking.
- Across tiles: multiply by the accumulated state.
- The denominator requires the corresponding key-state computation.

Dimension-consistent exposition of Sec. 5.2, p. 7, and Appendix B.1, pp. 22–24.

Fusing state operations makes the linear view fast

1. **Compute features inside the kernel.** Avoid writing expanded Q, K to global memory.
2. **Exploit Taylor structure.** Maintain separate zero-, first-, and second-order contributions.
3. **Partition the large state across registers.** Use 16-token tiles and eight warps in the prefill kernel.
4. **Fuse decode updates and reads.** Reduce temporary outer products and repeated memory traffic.

What is actually accelerated?

The model still reads and writes persistent state. Fusion reduces avoidable intermediate traffic.

Paper, Sec. 5, pp. 6–7; Algorithms 1–3, pp. 23, 27, 30. Microbenchmarks in Appendix B have narrower scopes.

Evaluate recall separately from general language quality

Evaluation	Setup	Metric / purpose
Pile	Same tokens and order; GPT-2 BPE	Overall and AR-slice PPL ↓
SWDE / FDA	Zero-shot field extraction	Accuracy ↑; use input facts
SQuAD variant	2,984 retained rewritten prompts	F1 ↑; answer completion
LM Eval	Standard tasks	General language quality
SuperGLUE	0, 1, 5 shots	Few-shot behavior

Language models: roughly 360M and 1.3B parameters; up to 30B and 50B tokens.

Transformer++ is the modern attention baseline; Mamba is the main recurrent baseline.

Paper, Sec. 6.1, pp. 8–9; Appendix E, pp. 36–39. Pretraining: A100; efficiency tests: H100.

BASED improves the AR slice at similar overall PPL

Scale / tokens	Model	Pile PPL ↓	AR PPL ↓	LM avg. ↑
360M / 10B	Transformer++	8.39	1.87	44.08
	Mamba	8.64	2.21	43.51
	BASED	8.65	2.07	43.03
1.3B / 50B	Transformer++	6.28	1.65	53.33
	Mamba	6.28	1.74	53.50
	BASED	6.30	1.71	53.81

The AR slice targets repeated contextual bigrams that are infrequent in training.

At 1.3B/50B, non-AR PPL is 6.82 for BASED vs. 6.78 for Mamba: gains are task-specific.

Paper, Table 1, p. 8; Table 2, p. 32; Appendix E.3, p. 37. Averages reproduced as reported.

Recall improves, with a substantial gap on FDA remaining

1.3B scale / 50B tokens	SWDE Acc.	FDA Acc.	SQuAD F1
Transformer++	76.50	80.47	43.47
Mamba	52.75	18.51	35.92
BASED	64.45	30.40	41.62
BASED − Mamba	+11.70	+11.89	+5.70

- All three tasks improve over Mamba.
- FDA remains 50.07 percentage points below Transformer++.
- The unweighted mean gain here is **9.76**, not the 10.36 quoted for this setting in the prose.

Paper, Table 1, p. 8. Differences and mean calculated from displayed values; accuracy points and F1 points are distinct metrics.

Local mixers contribute beyond a larger linear state

Variant	d'	Pile PPL ↓	AR PPL ↓
Taylor + SWA + convolution	16	8.65	2.07
Without SWA	16	8.91	2.11
Without convolution	16	8.74	2.09
Without both local mixers	24	9.49	2.29

- Removing either local component hurts overall perplexity.
- The two-attention hybrid without convolution is within 0.09 PPL of the full model.
- The last row also changes d'; it is not a pure one-variable ablation.

Paper, Table 6, p. 35; about 362M parameters and 10B training tokens. Rows shown use the decay variant.

Increasing feature width yields diminishing PPL returns

d'	Compact D	Pile PPL	AR PPL
8	45	8.77	2.18
16	153	8.65	2.07
24	325	8.58	2.02
32	561	8.56	2.00

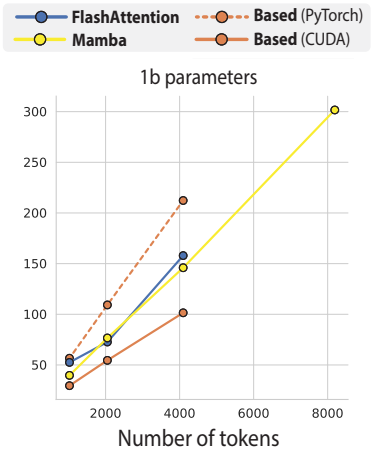
- PPL improves only 0.02 from $d' = 24$ to 32.
- State width rises from 325 to 561.
- Downstream gains are not monotonic: FDA is 20.87 at 24, but 12.89 at 32.

A second control knob

Window 64 $\rightarrow$ 128: PPL 8.65 $\rightarrow$ 8.61, with larger gains on some recall tasks.

Paper, Table 6, p. 35. Taylor map; local mixers and decay retained. PPL: lower is better.

Prefill gains depend on the optimized kernel



1.3B model scale

Single H100; batch 2.
Vertical axis: latency in ms.

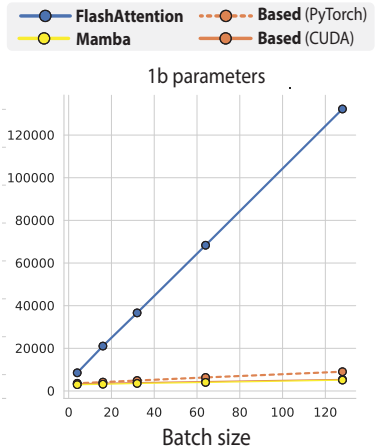
At 4K input tokens:

- About 56% higher throughput than FlashAttention-2.
- About 44% higher than Mamba.

The unoptimized BASED implementation can be slower than attention.

Paper, selected Fig. 4 panel, p. 8; Sec. 6.2, p. 9. Original panel labels the scale "1b"; text specifies 1.3B.

The 24-times headline is a specific decode result



Reported comparison:

- Single H100; 1.3B scale.
- Batch size 128.
- Generate 1,024 tokens.
- Excludes prefill; CUDA graphs.

$\sim$24$\times$ FlashAttention-2 throughput; about **95% of Mamba throughput**.

Vertical axis: latency in ms.

Paper, selected Fig. 4 panel, p. 8; Sec. 6.2, p. 9. Table 1 separately labels generation length as 2,048.

Few-shot results support the context-use motivation

SuperGLUE, 360M scale / 10B tokens	0-shot	1-shot	5-shot
BASED	45.7	47.2	48.0
Transformer++	46.3	47.8	48.9
Mamba	46.6	46.9	45.2

- BASED and Transformer++ improve as demonstrations increase.
- Mamba's 5-shot average drops below its 0-shot average.
- Average metrics within each task first, then average tasks.

This is consistent with the authors' link between recall and in-context use.

Paper, Appendix D.2 and Table 3, PDF p. 33. Reported task averages.

The useful principle is to co-design capacity and access

1. **Budget memory for the task.** Fixed-state models face an information limit on arbitrary recall.
2. **Separate global storage from precise local computation.** Taylor memory and local mixers complement each other.
3. **Measure the implementation.** State shapes and data movement determine whether the design is fast.

Remaining tradeoff

Larger state can improve recall but increases memory traffic.
BASED still has a large FDA gap to Transformer++ at 1.3B/50B.

Synthesis of Secs. 3–6. Language-model evidence extends to 1.3B parameters and 50B tokens in this paper.

Discussion: what should the next comparison control?

A useful follow-up experiment

Compare memory update rules at the same actual state bytes, numeric precision, local mixers, layer count, and kernel quality.

- Does the gain come from more capacity or a better update rule?
- Can state grow with the number of useful associations?
- Which failures are caused by interference versus local misalignment?

Memory capacity + precise access + efficient execution

Discussion questions motivated by the paper; these are not evaluated results. Main talk ends here.

Backup: the depth bounds rely on a restricted model

Data-independent BaseConv uses linear operations and multiplicative gates.

$$\deg(\text{output after } L \text{ layers}) \le 2^L.$$

For binary tokens, $d = \log_2 c$, exact matching contains

$$\mathrm{EQ}(q,k) = \prod_{a=1}^{d} [q_a k_a + (1-q_a)(1-k_a)].$$

- Matching and returning a value gives a degree-$2d+1$ interaction.
- This yields a depth requirement of order $\log d = \log\log c$.
- A separate bound is $\Omega(\epsilon \log\log N)$ when $c \le N$, $\log c \le d \le 2^{(\log N)^{1-\epsilon}}$, and parameters use $O(\log N)$ bits.

Paper, Theorems 3.2, 3.4, pp. 4–5; Appendix F.4, pp. 44–49. Not a bound for arbitrary nonlinear recurrent architectures.

Backup: an upper bound uses larger internal representations

Theorem F.30

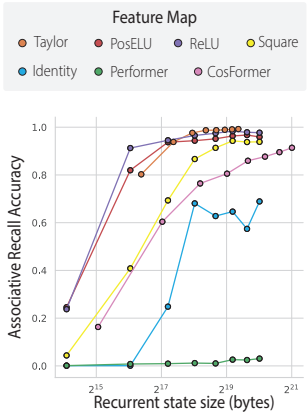
With one-hot tokens and at most one matching key per query, MQAR is solvable in $O(\log \log \bar{N})$ $\textsc{BaseConv}$ layers, where $\bar{N} = \max(N, d) \geq 8$.

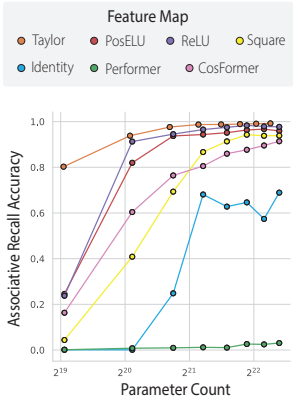
1. Store binary position codes with keys; cumulatively sum them.
2. Let a query select the matching position code.
3. Convert the code to a one-hot selector using a product tree.
4. Select the associated value.

Internal length and width expand to $O(\bar{N} \log \bar{N})$ scale.

Paper, Theorem F.30 and constructive proof, PDF pp. 51–74. High-level proof explanation.

Backup: Taylor is efficient, but not the only good map





Taylor, PosELU, and ReLU approach the memory frontier; Taylor uses fewer learned parameters for strong recall in this setting.

Paper, Fig. 3, p. 6. 256 KV pairs; original panels separated and original legend repeated for readability.

Backup: DNA results extend the scope beyond English

HG38	Params (M)	PPL, 1K	PPL, 4K	PPL, 8K
Transformer++	46.2	2.52	2.50	2.51
Mamba	46.1	2.51	2.49	2.49
BASED	48.8	2.51	2.50	2.49

- 10B pretraining tokens; learning-rate sweeps per architecture.
- Comparable PPL across evaluated lengths.
- BASED classification: 74.6 on Enhancer Cohn, 96.8 on Non-TATA Promoters; mixed ranking across five tasks.
- Some prior published baselines use different tokenization.

Paper, Appendix D.3, Tables 4–5, PDF pp. 33–34. PPL lower is better; classification accuracy in percent.

Backup: protocol details change what the scores mean

- **Pile AR slice:** a repeated contextual bigram's final token; bigram occurs at most 1,250 times in training. Computed on the first 16M test tokens.
- **FDA:** 1,920-token document chunks. Count a prompt correct if generation contains the value, ignoring case.
- **SQuAD:** GPT-4 reformats questions as answer-final statements; 2,984 of 5,000 prompts retained. Table 1 reports F1.
- **Decay:** used in the 10B-token variants; absent in the 30B/50B-token main models.

Paper, Appendix E.3, pp. 37–39; Appendices C and D.4, pp. 31, 34. Table 6 inconsistently labels the SQuAD metric as accuracy.

Backup: use task-level numbers when summaries disagree

Setting	ΔSWDE	ΔFDA	ΔSQuAD	Mean
360M / 10B	5.49	5.18	1.01	3.89
360M / 30B	17.38	7.74	5.96	10.36
1.3B / 10B	13.32	11.52	2.26	9.03
1.3B / 50B	11.70	11.89	5.70	9.76

Differences are BASED minus Mamba, calculated from Table 1. The mean mixes accuracy and F1 points.

The PDF associates 10.36 with 1.3B/50B; the online abstract says 6.22.
For speed, Fig. 4 uses 1,024 generated tokens; Table 1 says 2,048.

Primary reference: Arora et al., *Simple linear attention language models balance the recall-throughput tradeoff*, arXiv:2402.18668v2, 2025. Versioned paper link. Table 1, p. 8.