Just read twice: closing the recall gap for recurrent language models

Reading order, finite memory, and prefix linear attention

Paper by Simran Arora et al.

Independent research-group reading presentation

arXiv:2407.05483v1 (7 July 2024)
Read on 29 September 2026

Let the task inform what a recurrent model stores.

Knowing the question changes what to remember

A fixed state cannot retain every fact in a long document

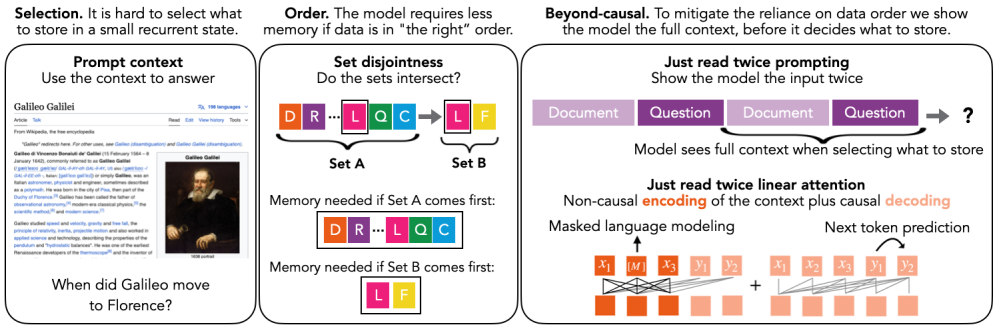
The selection problem is harder when the question arrives *after* the document.

Input order	**What selection can use**
Document → Question	Predict which facts will be useful.
Question → Document	Select facts for a known request.
Document → Question → Document	Revisit facts after seeing the request.

Extra reading can improve the use of the same state capacity.

Teaching interpretation of Secs. 1–3, PDF pp. 1–5. This is not a guarantee of perfect recall.

Two interventions give selection more context

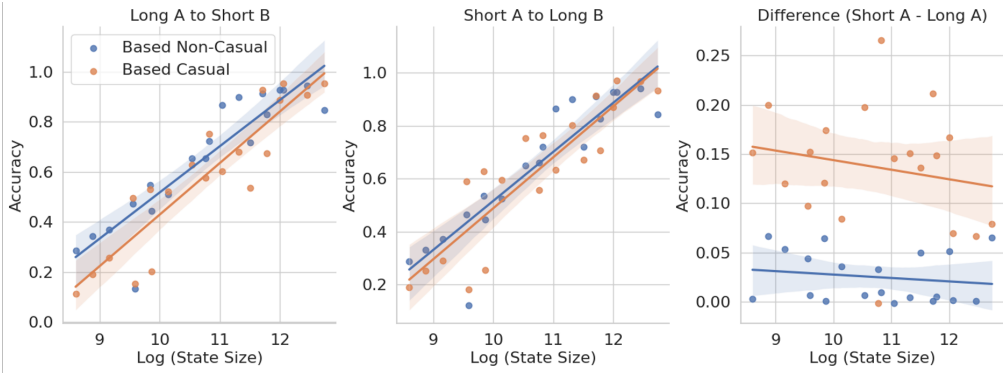


JRT-Prompt
Repeat input using an existing LM.

JRT-RNN
Train with a non-causal prefix and causal decoding.

Original Fig. 1, PDF p. 2. JRT-Prompt changes the input; JRT-RNN changes the model and training.

Causal models are more sensitive to set order



- Input: two sets with one shared token; output that token.
- The right panel measures accuracy(short A) − accuracy(long A).
- Smaller states show a larger order penalty for causal Based.

Original Fig. 2, PDF p. 4; App. B.1, pp. 22–23. Points are best over 3 learning rates and 2 seeds.

Re-reading helps, but a memory lower bound remains

Let $k = \min(|A|, |B|)$, p be the number of passes, and S the state in bits.

Communication argument

Alice holds keys; Bob holds queries. Each pass transfers the state.

$$2pS \;\geq\; \Omega(k) \quad\Longrightarrow\quad S \;\geq\; \Omega(k/p).$$

- Two passes can store the smaller set, using $O(nk)$ space for n-bit elements.
- The Based construction needs input-dependent convolution and kernel assumptions.
- Constant passes cannot solve arbitrarily large recall tasks with a fixed finite state.

Prop. 3.1 and Thm. 3.2, p. 5; Thm. G.15, pp. 44–45; Prop. G.29, pp. 52–53.

The prompt changes both repetition and question order

Abstract idea in the main text

$$\widehat{\mathcal{Y}} = \mathcal{A}(\mathcal{C}, \mathcal{Q}, \mathcal{C}, \mathcal{Q}).$$

Templates in Appendix F

Default: Document $\rightarrow$ Cloze cue

Twice: Question $\rightarrow$ Document $\rightarrow$ Document $\rightarrow$ Cloze cue.

- No weight updates; the same document information appears twice.
- The reported gain is for this complete prompting strategy.
- These comparisons do not isolate repetition from question-first ordering.

Sec. 3.2, p. 5; Table 1, p. 6; actual templates in App. F, pp. 34–39. Last point: reading analysis.

JRT-Prompt improves several off-the-shelf models

Model / parameters / training tokens	Default	JRT	Δ pp
Transformer++ / 1.3B / 50B	47.8	57.8	+10.0
Mamba / 1.3B / 50B	34.7	46.5	+11.8
Based / 1.3B / 50B	40.3	51.8	+11.5
GLA / 1.3B / 100B	36.7	48.9	+12.2
Mamba / 2.8B / 300B	42.8	52.6	+9.8
Mamba-2 / 2.7B / 300B	48.9	58.8	+9.9

- Six-task average accuracy; higher is better.
- Default input: up to 1k tokens; JRT: up to 2k.
- Transformers also benefit; some individual task scores decrease.

Selected rows from Table 1, PDF p. 6. Differences use rounded averages. Model-count caveat in backup.

Linear attention compresses history into two running sums

Use row-vector features $\phi(q_i), \phi(k_i) \in \mathbb{R}^{1\times \tilde{d}}$ and values $v_i \in \mathbb{R}^{1\times d_v}$.

$$s_i = s_{i-1} + \phi(k_i)^\top v_i, \qquad s_i \in \mathbb{R}^{\tilde{d}\times d_v},$$
$$z_i = z_{i-1} + \phi(k_i)^\top, \qquad z_i \in \mathbb{R}^{\tilde{d}\times 1}.$$

$$\boxed{y_i = \frac{\phi(q_i)s_i}{\phi(q_i)z_i}}$$

- Write a key–value outer product, then read with the query.
- The denominator normalizes by accumulated key similarity.
- Per-token state and compute are constant in sequence length, with cost $O(\tilde{d}d_v)$ per head.

Eqs. (1)–(2), PDF p. 7. Row-vector notation made dimensionally consistent for this presentation.

Prefix linear attention adds a non-causal state

$$s_{\mathrm{E}} = \sum_{j=1}^{M} \phi(k_{e_j})^\top v_{e_j}, \qquad z_{\mathrm{E}} = \sum_{j=1}^{M} \phi(k_{e_j})^\top,$$

$$s_{\mathrm{D},i} = \sum_{j=1}^{i} \phi(k_{d_j})^\top v_{d_j}, \quad z_{\mathrm{D},i} = \sum_{j=1}^{i} \phi(k_{d_j})^\top.$$

$$\boxed{y_i = \frac{\phi(q_i)(s_{\mathrm{E}} + s_{\mathrm{D},i})}{\phi(q_i)(z_{\mathrm{E}} + z_{\mathrm{D},i})}}$$

- Separate encoder and decoder K/V projections.
- The M prefix tokens enter both paths; prefix queries see all of them.
- After prefill, merge states and update only with new decoder K/V.

Eqs. (3)–(4), p. 7; implementation, pp. 29–30. The query typo in Eq. (3) is corrected here.

JRT-RNN trains both the prefix and the causal suffix

Prefix: $M = 1024$

Non-causal encoding.
Mask 15% of prefix tokens.
Predict their original identities.

Suffix: $N - M = 1024$

Causal language modeling.
Standard next-token prediction.
No future suffix access.

$$\mathcal{L} = \frac{\mathcal{L}_{\text{NTP}} + 0.25\,\mathcal{L}_{\text{MLM}}}{1.25}.$$

- Replace Based linear-attention layers with PLA; retain a hybrid of local attention and short convolutions.
- Train on the Pile, with matched data order across model groups.
- No training-time mask tokens are inserted at inference.

Sec. 4.3, Eq. (5), p. 8; App. B, pp. 23–24; Table 15, p. 54. Configuration caveats in notes.

The quality benchmark tests document-grounded completion

Tasks	FDA, SWDE, NQ, SQuAD, TriviaQA, DROP
Prompt format	Cloze completion; questions rewritten using Llama-3-70B, with the answer removed
Accuracy	Generated text contains the specified exact answer span
Generation	Stop at a newline or 48 generated tokens
Main average	Nine columns: two lengths each for FDA/SWDE/NQ, plus three Full-document tasks

This is a processed evaluation suite, with its own metric and weighting.

Table 2, p. 9; App. B.5 and Table 7, pp. 24–26. Truncated documents retain the answer span.

JRT-RNN closes much of the average recall gap

Params / tokens	Transformer	Mamba	Based	JRT-RNN
∼360M / 30B	43.4	24.0	29.2	**42.9**
∼1.3B / 10B	43.9	25.2	33.7	**42.0**
∼1.3B / 50B	51.4	36.8	42.6	**49.5**

The reported 99% and 96% are score ratios

$$\frac{42.9}{43.4} \approx 98.85\%, \qquad \frac{49.5}{51.4} \approx 96.30\%.$$

- Improvement over Based: **13.7 pp** and **6.9 pp**.
- The three 512-token JRT conditions use read-twice padding.

Table 2 and Sec. 5.1, PDF p. 9. These are nine-column averages, not universal capability ratios.

Read-twice padding contributes to the main quality result

JRT-RNN, 1.3B / 50B; documents of 512 tokens

Inference strategy	FDA	SWDE	NQ
Left-pad	75.8	49.3	30.9
Read-2×	**86.7**	**49.4**	**38.3**
Iterative encoding	80.2	43.3	34.2

- Fill the fixed $M = 1024$ prefix with another copy of the context.
- At 360M, FDA rises from 61.9 to 82.0: a 20.1 pp change.
- Iterative encoding reprocesses generated text and is more expensive; it does not win here.

Table 4 and Sec. 5.1, PDF pp. 9–10. The main table uses the Read-2x rows for these tasks.

The 2k setting still leaves a substantial Transformer gap

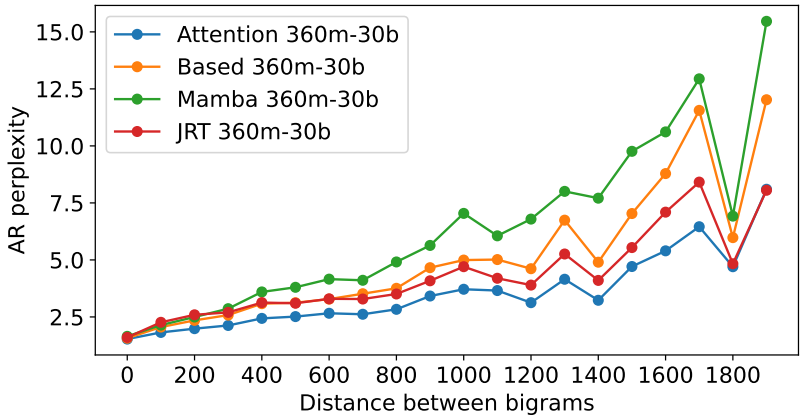
1.3B / 50B; 2k document inputs

Model	FDA	SWDE	NQ
Transformer	**79.7**	**55.5**	**30.2**
Mamba	21.0	29.9	23.1
Based	36.1	37.7	23.4
JRT-RNN	55.2	41.4	26.2

- JRT improves over recurrent baselines, but FDA remains 24.5 pp behind.
- 2k exceeds the *encoder region* ($M = 1024$), not the total pretraining length ($N = 2048$).
- The small-model rows in Table 3 are labeled 360M/10B, not 30B.

Table 3, PDF p. 9; training context in Sec. 5.2, p. 10.

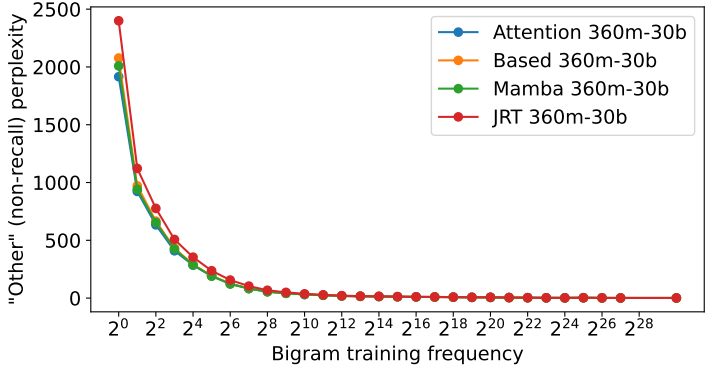
JRT helps recall across larger gaps between repeated bigrams



Lower perplexity is better. JRT improves on Based and Mamba; attention often remains stronger.

Original Fig. 3, top right, p. 10. 360M/30B; bigrams seen fewer than 1000 times in training.

Better recall comes with a non-recall perplexity tradeoff



- Rare non-recall tokens are harder for JRT in this comparison.
- Evaluate only the final 1024 causal positions of 2048-token sequences.
- Reduced NTP supervision is the authors' explanation, not an isolated cause.

Original Fig. 3, bottom, p. 10; protocol: App. B.4, p. 24. 3200 test sequences.

General language understanding shows mixed changes

SuperGLUE reported averages, ∼360M / 30B

Model	Zero-shot	Five-shot
JRT-RNN	39.6	45.3
Based	40.9	48.1
Transformer	42.0	45.3
Mamba	43.7	45.5

- Strong recall gains do not imply gains on every downstream task.
- The paper does not establish statistical equivalence.
- The 1.3B table omits ReCoRD; do not compare its average across scales.

Tables 8–9, PDF p. 27. Official LM-Eval Harness settings; retain the paper's reported averages.

Custom CUDA yields large gains for long prefill kernels

Implementation	2048 tokens	32768 tokens
FlashAttention-2	0.5 ms	107.8 ms
Based Triton (FLA)	1.0 ms	123.7 ms
Based Custom CUDA	0.3 ms	4.5 ms
JRT-Prompt Custom CUDA	0.6 ms	9.0 ms
JRT-RNN Custom CUDA	0.4 ms	5.6 ms

Reported speedups at 32k, batch 16, NVIDIA H100

JRT-Prompt: **11.9**×; JRT-RNN: **19.2**×, relative to FA2.

- Average of 20 iterations. Prompt repetition is charged at 2× Based time.
- Kernel prefill timing; no end-to-end generation speedup is established.

Table 5, PDF p. 11; Sec. 5.3, pp. 11–12. Short-input JRT-Prompt is slower than FA2 here.

The strongest claims depend on four boundaries

1. **Memory.** Re-reading does not remove the finite-state lower bound. Benefits depend on how much relevant information is needed.
2. **Attribution.** Prompt order, repetition, training objectives, and padding all change. The main tables evaluate complete recipes.
3. **Length.** Quality is measured on short inputs and around 2k; 32k is a separate efficiency test.
4. **Scope.** JRT is a small-to-mid-scale pretrained prototype. Generalization to large instruction models is not demonstrated.

Reading synthesis of Secs. 3–5 and Apps. B–G. Evidence boundaries, not additional experiments.

The design lesson is to condition compression on the task

A useful question for efficient language models

When the model compresses a document, does it already know what information will matter?

- **JRT-Prompt:** revisit the input after acquiring more context.
- **JRT-RNN:** make the prefix visible non-causally, then retain efficient causal state updates.
- **Next experiment:** isolate question placement from repetition, and compare complete systems at matched quality and cost.

Paper conclusion, PDF p. 12. Final experiment suggestion is this presentation's analysis.

Source and reading scope

Just read twice: closing the recall gap for recurrent language models

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, Christopher Ré.

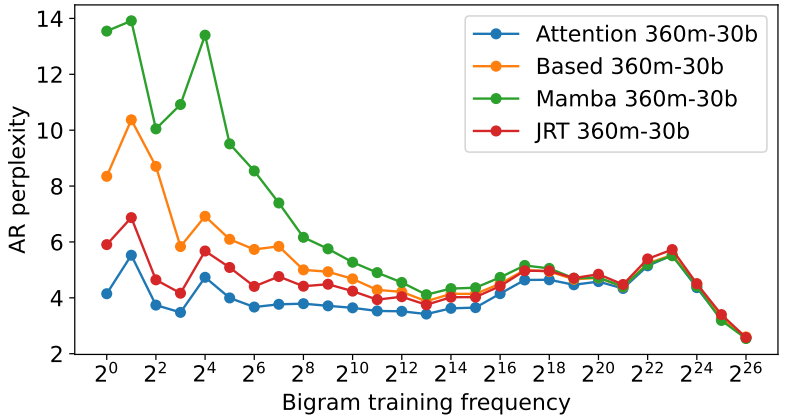
arXiv:2407.05483v1, submitted 7 July 2024.

Official repository linked by the paper. The repository was not executed for this reading.

- Same-version author TeX and 56-page PDF were used.
- Main text, substantive Appendices A–G, and training tables were read.
- Original figures are copied from author assets; selected tables are re-typeset.
- Suggested talk: about 25 minutes through slide 20; backups follow.

Companion Chinese notes document detailed source checks and version-specific inconsistencies.

Backup: rare repeated bigrams benefit from prefix processing



Recall-frequency analysis restricts repeated bigrams to gaps of at least 1024 tokens.

Original Fig. 3, top left, p. 10; App. B.4, p. 24. 360M/30B; lower perplexity is better.

Backup: two passes expose the smaller set before a useful scan

$A_1 \quad B_1 \quad A_2 \quad B_2$

Case	Store	Compare against
$\lvert A\rvert \le \lvert B\rvert$	A_2	B_2
$\lvert B\rvert < \lvert A\rvert$	B_1	A_2

- The construction uses the separator and known input length to identify the smaller set.
- Store $k = \min(\lvert A\rvert, \lvert B\rvert)$ elements: $O(nk)$ space for n-bit items.
- This is an existence construction, not an optimization guarantee for any pretrained RNN.

Teaching reconstruction of Thm. G.15 and Algorithm 3, PDF pp. 44–45.

Backup: a matching kernel separates hits from misses

For distinct tokens, assume near-orthogonal features:

$$\langle \phi(a), \phi(b) \rangle = \begin{cases} 1, & a = b, \\ \text{a value in } [-\epsilon, \epsilon], & a \neq b. \end{cases}$$

Summing over a stored set A with $\epsilon \leq 1/(3|A|)$:

$$\text{no match: } \rho \leq \tfrac{1}{3}, \qquad \text{match: } \rho \geq \tfrac{2}{3}.$$

- A thresholding MLP separates the two cases.
- The $O(nk)$ Based construction uses input-dependent reordering and IP features.
- It does not prove that a fixed Taylor map learns exact recall.

Prop. G.18, pp. 46–47; Thm. G.22 and Remark G.23, p. 51. Full assumptions and caveats in notes.

Backup: prefix processing helps beyond the default hybrid

360M / 30B; selected Table 12 columns

Mixer	SWDE-512	FDA-1024	FDA-2048
Linear attention	29.6	16.0	4.6
Prefix linear attention	36.8	48.7	8.2
Linear + sliding attention	25.4	8.1	3.0
Prefix linear + sliding	35.5	46.5	30.0

- Prefix variants improve these reported comparisons.
- Pure prefix linear attention still struggles on FDA-2048.
- The no-MLM Table 11 lacks a matched full-JRT row: it does not isolate the magnitude of the MLM effect.

App. C.2, Tables 11–12, PDF p. 28. Table 11 uses 10B training tokens; Table 12 uses 30B.

Backup: repetition reduces some errors and creates others

Mamba 2.8B, SQuAD; percentage of all test examples

Error category	Default	JRT-Prompt
Irrelevant output	51.3	13.8
Repetition	5.5	32.9

- A base LM may continue a repetition pattern instead of answering.
- Few-shot is not uniformly inferior: on SWDE, Mamba 2.8B scores 51.9 with two examples versus 48.9 with JRT.
- Table 13 trends vary by task; no single error pattern explains all gains.

App. E, Tables 13–14, PDF pp. 31–33.

Backup: the CUDA path accumulates the encoder state first

1. Apply the Based kernel to encoder K/V, using a sum instead of a causal cumulative sum and skipping query multiplication.
2. Keep the encoder KV state in registers.
3. Run the causal decoder kernel from this initial state, then write the outputs.

Taylor features for 16 inputs: $1 + 16 + 16^2 = 273$.

Managing this expanded state is essential to realizing the asymptotic benefit.

Algorithm 2 and App. D, PDF pp. 29–30; Taylor features in App. B, p. 23.

Backup: several v1 claims need careful bookkeeping

- **Model count.** Table 1 has 16 configurations: 14 recurrent and 2 Transformer rows. Recomputing six-task gains gives about 11.28 pp for the 14 recurrent rows.
- **Supervised positions.** The stated mask policy implies $0.5 + 0.15(0.5) = 57.5\%$, while the prose says 65%.
- **Theory.** Prop. G.26's reverse reduction uses intersection membership, more information than a Boolean disjointness oracle.
- **Configuration.** Window sizes and layer totals in Tables 15–16 do not fully agree with the prose.

Reading checks of Table 1, p. 6; Sec. 5.2, p. 10; Prop. G.26, p. 52; Tables 15–16, pp. 54–55.