# Proteus: when should memory become available?

## Incremental Memory Activation for Long-Context Sequence Modeling

Reza Bayat, Ali Behrouz, Vahab Mirrokni, Aaron Courville

arXiv:2608.16844v1 — 17 August 2026

Paper discussion prepared on 16 September 2026

### One-sentence takeaway

A fixed memory budget can work better when its read/write capacity is unlocked progressively.

# A fixed memory budget still leaves a scheduling choice

- Attention retains historical keys and values; full attention has quadratic sequence-processing cost.
- Recurrent memory compresses history into a fixed-size state, but long-context retrieval remains difficult.
- Common design axes: memory architecture, initialization, objective, and update rule.
- Proteus asks a different question: how much capacity should be accessible at each position?

Secs. 1–2, PDF pp. 2–4.

# The proposed mechanism targets two failure modes

## Early context

Full capacity may fit early inputs too freely.

**Intervention:** impose a small active subspace.

## Later context

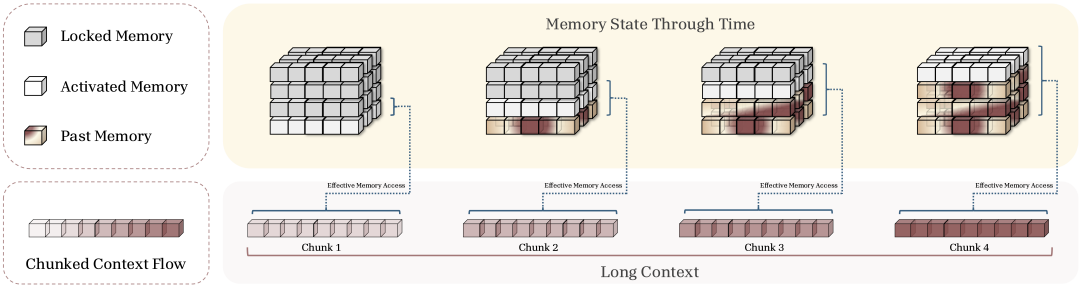
New writes may interfere with an already-written state.

**Intervention:** unlock unused blocks over time.

These are the authors' explanations. The paper does not directly measure memory pollution or information compression.

Secs. 2.4–3, PDF pp. 4–5; evidence boundary: reader analysis.

# New blocks open while old blocks remain writable




Total state size is constant. Previously activated blocks remain part of the read/write set; old content is not frozen.

Original Fig. 1, PDF p. 1.

# Online memory learning differs from model training

$$\mathcal{M}_t = \mathcal{M}_{t-1} - \theta_t \nabla \widetilde{\mathcal{L}}(\mathcal{M}_{t-1}; k_t, v_t), \qquad y_t = \mathrm{Read}(\mathcal{M}_{t-1}, q_t).$$

- $k_t, v_t, q_t$: key, value, query; $\theta_t$: online memory step size.
- $\widetilde{\mathcal{L}}$ fits key–value associations; the outer loss $L$ trains next-token prediction.
- Hebbian, delta-rule, and Titans updates instantiate different memory learners.
- Proteus schedules accessible coordinates, rather than replacing the inner objective.

Sec. 2, Eqs. (1)–(5), PDF pp. 3–4.

# Gate the state used for both reading and learning

$$\mathcal{M}_{t-1}^{(g)} = g_t \odot \mathcal{M}_{t-1}, \qquad y_t = \mathrm{Read}(\mathcal{M}_{t-1}^{(g)}, q_t)$$

$$\mathcal{M}_t = \mathcal{M}_{t-1} - \theta_t \left[ g_t \odot \nabla \widetilde{\mathcal{L}}(\mathcal{M}_{t-1}^{(g)}; k_t, v_t) \right].$$

- $g_t$ is a binary mask over memory coordinates.
- Compute with the active state, then apply only the active update to the full state.
- Locked coordinates must satisfy $(\mathbf{1} - g_t) \odot \mathcal{M}_t = (\mathbf{1} - g_t) \odot \mathcal{M}_{t-1}$.

Definition 2, Eq. (7), p. 5; Eqs. (8)–(11), p. 6.

# A deterministic prefix mask controls effective capacity

$$\Delta = \max\left(1, \left\lfloor \frac{N}{E} \right\rfloor\right), \qquad k(t) = \min\left(E, 1 + \left\lfloor \frac{t-1}{\Delta} \right\rfloor\right).$$

$$g_t[j] = \begin{cases} 1, & 1 \le j \le k(t)d, \\ 0, & \text{otherwise}, \end{cases} \qquad d = \dim(\mathcal{M})/E.$$

- $N$: training context length; $E$: block count; $d$: coordinates per equal block.
- $k(t)$ is a block count, distinct from the key vector $k_t$.
- Default $E = 16$; $E = 1$ recovers the base model.

Sec. 4.1, Eqs. (12)–(13) and intervening definition, PDF p. 6.

# The schedule finishes inside the training window

| Token positions | Active blocks | Active fraction |
|---|---|---|
| 1–512 | 1 | 1/16 |
| 513–1024 | 2 | 2/16 |
| 3585–4096 | 8 | 8/16 |
| 7681 onward | 16 | 1 |

Example computed from $N = 8192$, $E = 16$, $\Delta = 512$.

At 16K or 32K, no new blocks are being unlocked. Evaluation keeps the original schedule.

Teaching example derived from Sec. 4.1, PDF p. 6; not an extra experiment.

# Hope extends the idea to a different time axis

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta_{i+1}(g_{i+1} \odot \boldsymbol{e}_i).$$

- Here $i$ indexes training samples/updates, not positions within one context.
- Hope-Attention passes attention output through a chain of MLP blocks updated at different frequencies.
- The extension gates parameter updates over training; it is demonstrated on one architecture.
- Forward masking and the detailed mapping to training steps are not fully specified.

Sec. 4.2, Eqs. (14)–(18), PDF pp. 6–8; implementation boundary: reader analysis.

# Evaluation pairs each backbone with its Proteus variant

| | |
|---|---|
| Training data | FineWeb |
| Scale / training tokens | 760M / 50B; 1.3B / 100B |
| Context / partition count | 8K / $E = 16$ |
| Optimizer | AdamW, $4 \times 10^{-4}$, cosine schedule |
| Batch / weight decay | 0.5M tokens / 0.1 |

Paired families: Hope-Attention, SWLA, Comba, Titans.

Reference models include Transformer++, RetNet, and DeltaNet.

Long-context tables do not explicitly identify the model scale per result.

Sec. 5 setup, PDF p. 8; Tables 1–3, pp. 7, 9–10.

# Mean downstream accuracy improves in all eight pairs

| Backbone | 760M: base → Proteus | 1.3B: base → Proteus |
|---|---|---|
| Hope-Attention | 53.15 → 53.99 | 56.93 → 57.30 |
| SWLA | 50.12 → 50.90 | 53.67 → 54.20 |
| Comba | 51.43 → 52.15 | 55.42 → 56.00 |
| Titans | 52.65 → 53.36 | 56.95 → 58.00 |

Avg. is the mean of eight accuracy columns: LAMBADA plus seven commonsense benchmarks. HellaSwag and ARC-c use normalized accuracy.

Improvements span 0.37–1.05 percentage points. No variance estimates are reported.

Selected values from Table 1, PDF p. 7; differences computed from reported means.

# Perplexity improves often, but not universally

| Scale / backbone | Metric | Base → Proteus |
|---|---|---|
| 760M Hope | Wiki. ppl | 20.67 → 19.87 |
| 760M Hope | LAMBADA ppl | 21.36 → 19.72 |
| 1.3B Titans | Wiki. ppl | 15.36 → 14.94 |
| 1.3B Hope | Wiki. ppl | 15.66 → 15.70 |
| 1.3B SWLA | LAMBADA ppl | 16.95 → 17.18 |

Lower is better. At 1.3B, the best LAMBADA ppl is **13.01 (Hope+Proteus)**, not 13.03 (Titans+Proteus), despite the prose claim.

Table 1, PDF p. 7; conflicting best-ppl statement in Sec. 5.1, p. 9.

# Hard 16K retrieval shows larger absolute gains

| Task | Backbone | Base $\rightarrow$ Proteus | Gain (pp) |
|---|---|---|---|
| S-NIAH-2 | Comba | 13.4 $\rightarrow$ 21.2 | +7.8 |
| S-NIAH-3 | Titans | 21.4 $\rightarrow$ 29.8 | +8.4 |
| MK-NIAH-1 | Titans | 11.8 $\rightarrow$ 16.8 | +5.0 |
| MQ-NIAH | Hope | 30.6 $\rightarrow$ 34.2 | +3.6 |
| MV-NIAH | Hope | 23.0 $\rightarrow$ 25.8 | +2.8 |

S-NIAH-2/3: numeric/UUID needles; MK/MQ/MV: multiple keys/queries/values.

The +8.4 pp UUID gain still leaves Titans at 29.8: difficult retrieval remains unsolved.

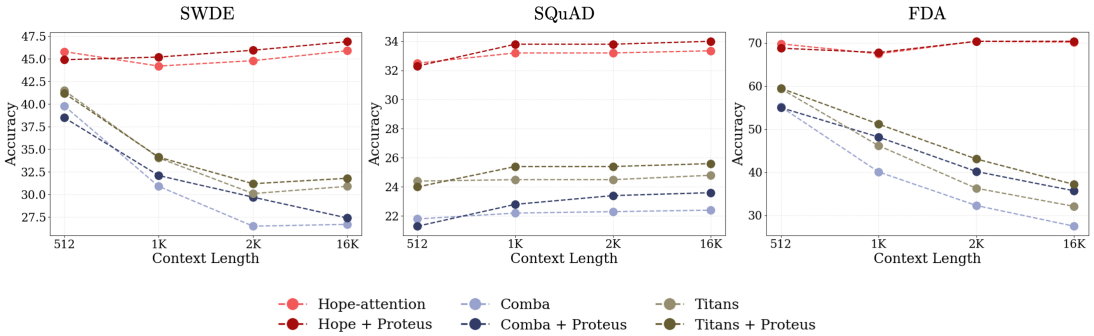
Selected 16K results from Table 2, PDF p. 9.

# Short-context and saturated tasks limit the claim

- S-NIAH-1 is already near 100 for most paired models: little room to improve.
- Comba, S-NIAH-3 at 4K: 62.4 $\rightarrow$ 60.8.
- Hope, MK-NIAH-1 at 4K: 81.4 $\rightarrow$ 80.8.
- At 16K, Comba/Titans remain well below Transformer/Hope on multi-needle tasks.
- Long-context benefits are compatible with better early compression; they do not isolate that causal mechanism.

Table 2, PDF p. 9; interpretation boundary: reader analysis.

# Retrieval benefits depend on the dataset and backbone




Read within each panel and backbone. SWDE/FDA show stronger length degradation; SQuAD does not show the same downward trend. No error bars are given.

Original Fig. 2, PDF p. 8; discussion in Sec. 5.3, p. 10.

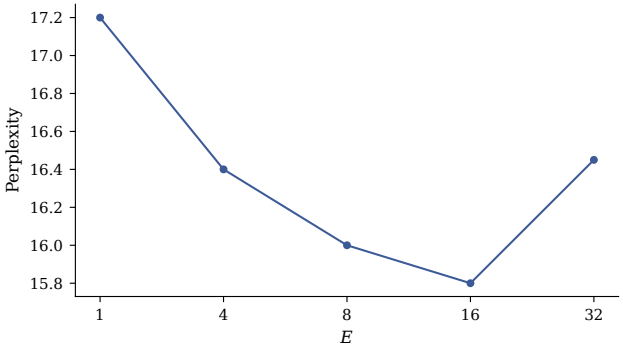
# Six-task LongBench means improve modestly

| Backbone | Base | Proteus | Difference |
|---|---|---|---|
| Hope-Attention | 15.72 | 16.65 | +0.93 |
| Comba | 13.05 | 13.23 | +0.18 |
| Titans | 13.80 | 14.15 | +0.35 |

- Six tasks: Narrative, Qasper, MultiField, Hotpot, 2WikiMulti, Musique.
- Exceptions: Comba Hotpot 14.6 $\rightarrow$ 14.5; Titans Musique 6.7 $\rightarrow$ 6.7.
- These are mean task scores for the reported subset, not a full LongBench aggregate.

Table 3, PDF p. 10.

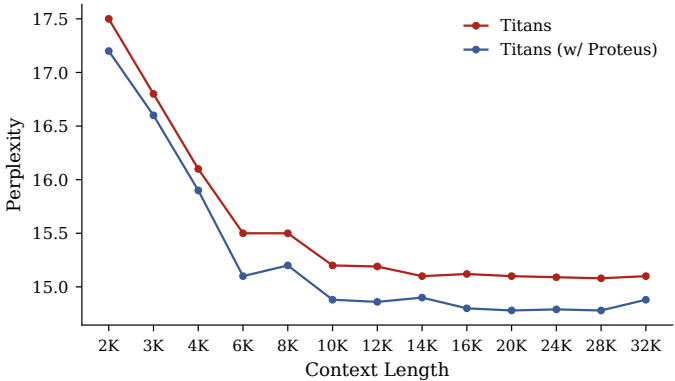
# More partitions help only up to a point




- $E = 1$: no scheduling.
- $E = 16$: best tested setting.
- $E = 32$: perplexity rises again.
- Block count changes both initial capacity and unlock frequency.

Original Fig. 3 and Appendix A, PDF p. 18. No error bars; exact values not tabulated.

# An advantage persists after all blocks are open




- Blue: Titans+Proteus; red: Titans.
- Lower ppl at all plotted points.
- Large gap near 8K; positive gap at 32K.
- No new blocks beyond the training window.

Original Fig. 4 and Appendix A, PDF p. 18. Discrete plotted points, not every token.

# The evidence supports scheduling, with clear limits

- Cross-backbone mean gains make capacity scheduling a useful design axis.
- No new learned parameters or larger main state; runtime overhead is not benchmarked.
- No random-seed variation or significance estimates are reported.
- No causal separation of early bottleneck versus fresh later capacity.
- Fixed hand-designed schedule; one-architecture MLP proof of concept.

Secs. 5 and 7; Appendix A. Cost/statistical boundaries: reader analysis.

# Three takeaways for a research discussion

1. **Capacity has a time dimension.** A fixed budget need not be accessible uniformly.
2. **Longer-context gains do not require continued expansion.** Here the schedule has already saturated by 8K.
3. **Mechanism and implementation deserve further tests.** Separate early regularization, later free capacity, and optimizer effects.

Learned schedules, genuinely growing memory, and stretched inference schedules are future directions, not evaluated solutions.

Sec. 7, PDF pp. 11–12; discussion synthesis.

# Reference and reading materials

**Primary source**

Reza Bayat, Ali Behrouz, Vahab Mirrokni, and Aaron Courville.
*Proteus: Incremental Memory Activation for Long-Context Sequence Modeling.*
arXiv:2608.16844v1, 17 August 2026.

This presentation uses the version-matched author TeX, three source tables, four original figures, and the final 18-page PDF.

Figures are reproduced from the paper. Selected tables are re-typeset from reported values. Teaching calculations and reader critiques are labeled.

# Backup: dimension conventions need care

For column vectors, $\mathcal{M}k_t \approx v_t$ implies

$$\mathcal{M} \in \mathbb{R}^{d_v \times d_k}, \qquad \nabla_{\mathcal{M}} \tfrac{1}{2}\|\mathcal{M}k_t - v_t\|^2 = (\mathcal{M}k_t - v_t)k_t^\top.$$

Thus a consistent delta update is

$$\mathcal{M}_t = \mathcal{M}_{t-1}(I - \theta_t k_t k_t^\top) + \theta_t v_t k_t^\top.$$

The paper's Eq. (4) instead left-multiplies the erase factor, while p. 5 states the transposed matrix shape. This is a notation concern, not evidence that the experiments used incorrect code.

Reader derivation; compare Eqs. (3)–(4), PDF p. 3, and matrix shape on p. 5.

# Backup: masked gradients do not mask weight decay

The paper's AdamW expression includes

$$\boldsymbol{\theta}_{i+1} = \alpha_{i+1}\boldsymbol{\theta}_i - \eta_{i+1}\frac{\hat{\boldsymbol{m}}_{i+1}}{\sqrt{\hat{\boldsymbol{v}}_{i+1}} + \varepsilon},$$

while moments use masked gradients.

- Even with zero moments, an inactive nonzero parameter changes if decay is unmasked.
- A monotone mask avoids previous nonzero momentum in never-active coordinates, but does not remove this decay issue.
- Exact freezing, forward masking, and the schedule-to-training-step mapping require clarification.

Eq. (15), pp. 7–8; discrepancy with the freezing description: reader analysis.