Fast-weight Product Key Memory

Sparse test-time learning for episodic memory

Paper: Tianyu Zhao and Llion Jones, Sakana AI

arXiv:2601.00671v2 | Paper discussion, 16 September 2026

Sparse memory can learn from the current context

- Problem: store many new associations without touching a dense memory on every token.
- FwPKM combines product-key retrieval with chunk-level fast-weight updates.
- Best-supported benefit: episodic retrieval alongside recurrent or local-attention backbones.
- Evidence boundaries: rereading costs, slower measured training, and weak long-term retention.

Secs. 1, 4, 6; PDF pp. 2–3, 7–11, 15.

Capacity and access cost are different constraints

Memory	Stored history	Main constraint
Full attention	Explicit token KV	Sequence-wide quadratic work
GDN / recurrent state	Fixed-size state	Information capacity
Dense TTT	Adaptive dense network	Read/write cost grows with size
FwPKM	Adaptive sparse table	Storage, collisions, sparse kernels

Fixed memory size gives history-length-independent FwPKM access work; it does not make memory-size scaling free.

Secs. 1–2; PDF pp. 2–4. Complexity qualification: reading analysis.

FwPKM sits before the backbone token mixer

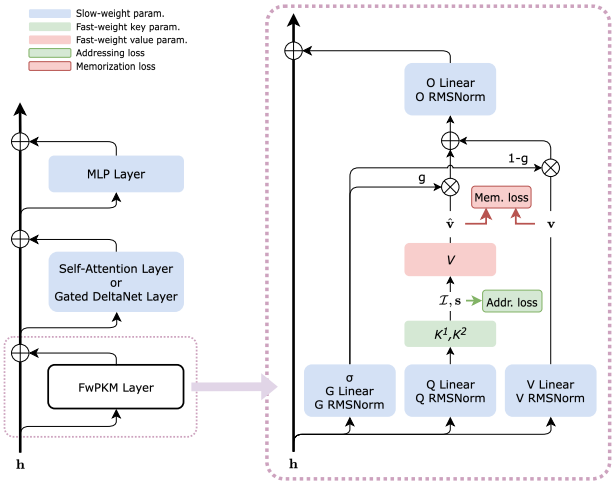


Fig. 1, PDF p. 4; architecture crop. Blue: slow; green/red: fast keys/values.

Two small key tables address 262,144 value slots

- $q \in \mathbb{R}^{512}$ splits into two 256-dimensional subqueries.
- Two 512×256 codebooks: score 1,024 subkeys.
- Top-8 per codebook $\rightarrow$ 64 candidate pairs $\rightarrow$ final Top-8.
- $V \in \mathbb{R}^{262144 \times 512}$; retrieve $\hat{v}_t = \sum_{i \in \mathcal{I}_t} s'_{t,i} V_i$.

$$s_i = -\log(10^{-3} + \|q - K_i\|_2^2)$$

Secs. 2–3; Appendix A.2/C; PDF pp. 3–6, 22, 25. IDW scores apply to subkeys.

A learned gate selects the memory contribution

$$o_t = g_t \hat{v}_t + (1 - g_t) v_t, \quad g_t \in (0, 1)$$

- Slow projections construct query, value, and scalar gate from the hidden state.
- The mixed value is normalized and projected back to the backbone.
- The same gate weights the local writing loss.
- Read with existing weights; update only after finishing the chunk.

Sec. 3.1–3.3; PDF pp. 4–5, Eqs. (8)–(13).

Lookahead stores what follows a similar query

$$\mathcal{L}_{\text{mem}}^{\text{LA}} = \sum_{t=1}^{C-1} \frac{1}{2} g_t \| v_{t+1} - \hat{v}_t \|_2^2$$

- Default target: next-token value representation, z-score normalized over features.
- Targets are available before the chunk-end write; same-chunk predictions cannot use that write.
- Training drops one unpaired position per 512-token chunk.
- Inference caches available query–target pairs; update frequency depends on the evaluation protocol.

Appendix A.1–A.5; PDF pp. 22–23, Eq. (19).

Slot updates aggregate competing residuals

$$V_i \leftarrow V_i - \frac{1}{N_i^{\mathrm{read}}} \nabla_{V_i} \mathcal{L}_{\mathrm{mem}}^{\mathrm{LA}}$$

- N_i^{read}: accesses to slot i in the chunk; only accessed rows update.
- Sum the reconstruction loss, then normalize gradients per accessed row.
- A single slot with unit gate and unit retrieval weight can be rewritten in one step.
- With soft weights and collisions, one pass need not reconstruct perfectly. This motivates rereading.

Sec. 3.3; Appendix A.4; PDF pp. 5, 22. Single-slot example: teaching derivation.

Address balancing has a sign inconsistency in the paper

$$\mathcal{L}_{\text{addr}} = -H(\bar{p}^{(1)}) - H(\bar{p}^{(2)})$$

- Intended objective: maximize entropy of chunk-averaged subkey usage.
- Eq. (17) expands this with a minus sign before each sum of p log p, which is positive entropy.
- The text and Algorithm 1 instead imply minimizing the sum of p log p.
- We flag the inconsistency; the author implementation was not executed or audited.

Sec. 3.4, Eq. (17), PDF p. 6; Algorithm 1, p. 24. Sign diagnosis: reading analysis.

The study tests small hybrid language models

- 12 layers, hidden size 768; baselines about 112M parameters.
- 10B tokens: 5B LongContext64 + 5B Fineweb-Edu; training length 4K.
- Backbones: GDN, GDN+SWA, GDN+FA, FA; hybrids use a 3:1 ratio.
- PKM replaces MLPs; FwPKM precedes token mixers at selected layers 2, 6, 10.
- FwMLP and LaCT provide dense fast-weight comparisons; their update rules and state sharing differ.

Sec. 4.1; Appendix C; PDF pp. 6–7, 25–26.

Evaluation changes the memory update protocol

Test	State/update protocol	Metric
Corpus PPL	Ordered 4K segments, no fast-state reset between adjacent segments; chunk 512	All-token PPL
NIAH	One update after each full-context read; 1–4 reads	Retrieval accuracy
LongBench	One context read/update; no answer-time updates	Answer-token PPL
Adaptation	Six domains, slow weights frozen	Change in NLL

Secs. 4.2–4.5; PDF pp. 7–11. NIAH also freezes fast weights during answer generation.

Sparse fast memory improves GDN corpus perplexity

GDN variant	Fineweb-Edu	LC64	LAMBADA
Baseline	19.73	10.12	32.91
PKM at 2,6,10	18.27	9.27	29.97
FwPKM at 2,6,10	18.09	8.89	26.73
PKM at 6 + FwPKM at 2,10	17.39	8.54	25.91

- All entries are PPL (lower is better).
- The mixed variant reduces LC64 PPL by 15.6% relative to GDN.
- These are ordered-stream results with fast memory carried across segments.

Fig. 2, PDF p. 8; selected rows transcribed. Relative reduction calculated from reported values.

Full attention can suppress use of the added memory

- With FA or GDN+FA, gates often cluster near zero and PPL gains are limited.
- Training-time pSWA limits full attention to a 512-token window with probability 0.9.
- This encourages FwPKM use; stronger benefits emerge on retrieval tasks.
- Interpretation: memory utility depends on competition between available information paths.

Sec. 4.2; Figs. 2–3, PDF pp. 8–9.

128K retrieval is demonstrated with a rereading budget

Model, four reads	4K accuracy	128K accuracy
GDN + FwPKM at 2,6,10	88.6%	73.8%
GDN+SWA + PKM/FwPKM mix	80.8%	74.8%
GDN+FA + FwPKM (pSWA)	85.0%	84.0%

- Five needles per sample; four-character keys and six-digit values.
- Each read uses the entire haystack as one update chunk.
- Training is 4K; longer evaluations also apply YaRN to RoPE.
- Four-read results are not one-pass or equal-latency results.

Fig. 4, PDF p. 10; protocol p. 9; YaRN in Appendix C, p. 25.

LongBench measures answer probability, not official QA scores

- 950 examples across five QA tasks; average 17,421 tokens, maximum 81,616.
- Only answer tokens are scored; the first reference is used when several exist.
- GDN to PKM/FwPKM mix: 2WikiMultihopQA PPL 19.50 to 12.69; HotpotQA 36.25 to 21.00.
- LaCT is omitted from the plot because PPL exceeds 100.
- This supports better answer likelihood, not an established gain in F1 or exact match.

Sec. 4.4, Fig. 5; PDF pp. 9–11.

Fast adaptation does not ensure long-term retention

- Six Pile domains; each has separate 4M-token adaptation and evaluation sets.
- After adapting to one domain, evaluate all six domains against initial NLL.
- Current-domain NLL improves, but later stages often erase the gain.
- The paper motivates consolidation and retention mechanisms; it does not provide a solution.

Sec. 4.5, Fig. 6; PDF pp. 10–12.

Lookahead is crucial for the retrieval behavior

GDN+SWA mixed variant	128K NIAH, four reads
Default	74.8%
Without addressing loss	37.2%
Without gate-weighted writing loss	19.4%
Without lookahead targets	1.8%

- Removing addressing loss uses MSE to update both keys and values.
- Small PPL changes can hide a large loss of retrieval ability.
- Larger Top-k / more heads do not improve every task consistently.

Appendix D.1, Figs. 9–12; PDF pp. 27–28.

Address balancing helps, but usage remains nonuniform

LC64 configuration	Collision	Coverage	KLD
Default	85.2%	28.4%	2.1
Without addressing loss	96.4%	8.0%	5.4
One head, Top-32	95.4%	55.0%	1.9

- Statistics are computed every 32K tokens, then averaged.
- Lower collision/KLD and higher coverage are preferred within comparable settings.
- Top-k changes the access budget; uniform subkey marginals do not guarantee uniform joint slots.

Appendix D.3, Table 4; PDF p. 31. Joint-distribution qualification: reading analysis.

Freezing memory separates dynamic use from added capacity

GDN+SWA + PKM/FwPKM mix	Updating	Frozen
Fineweb-Edu PPL	16.76	16.97
LC64 PPL	8.27	8.77
LAMBADA PPL	26.43	28.07

- Long-context corpora are more affected by disabling updates.
- Cross-segment PPL visualizations offer supporting evidence.
- Curve shapes alone do not establish the proposed causal mechanism.

Appendix D.2, Fig. 14, p. 30; Appendix E.1, Fig. 15, p. 32.

Memory access is traceable, with limits to the interpretation

- A 4K/four-read case traces retrieved slots while generating a six-digit passkey.
- Layer-2 and layer-10 slots often include the correct next token, with some noisy slots.
- A separate text example shows selective high-layer gates on rare entities.
- That visualization uses 32-token updates, not the default 512.
- These are informative cases, not population-level retrieval or novelty-detection guarantees.

Sec. 5, Figs. 7–8; PDF pp. 12–14; extra example in Fig. 16, p. 33.

Low active compute has not yet delivered high throughput

GDN variant	Total M	Active M	FLOPs T	Samples/s
Baseline	112.11	112.11	22.04	54.44
PKM	509.08	106.62	20.92	42.90
FwPKM	519.12	116.48	22.74	19.04
PKM/FwPKM mix	515.77	113.19	22.14	21.27

- FwPKM: 4.63× total parameters, 1.03× FLOPs, 35% of baseline throughput.
- Single H100, batch 8, 100 training steps; not a decoding-latency benchmark.
- Better sparse kernels are a proposed next step, not a measured result.

Sec. 6, Table 1; PDF pp. 14–15. Ratios calculated from reported values.

Take-home: a promising memory component with clear boundaries

- Contribution: a large sparse memory that can rewrite context-specific associations online.
- Mechanism: lookahead reconstruction, selective writing, and anti-collapse addressing.
- Evidence: streaming PPL, budgeted long-range retrieval, and answer-token likelihood.
- Open issues: entropy-sign specification, state isolation, retention, and actual serving efficiency.
- Next experiment: compare under matched memory, rereading compute, and end-to-end latency.

Paper conclusions: Sec. 8, p. 17. Proposed next experiment: reading analysis.

Reference and reading scope

Tianyu Zhao and Llion Jones.
Fast-weight Product Key Memory. Sakana AI.
arXiv:2601.00671v2, revised 22 February 2026.

`https://arxiv.org/abs/2601.00671v2`

Reading: complete main text and Appendices A–E, author TeX and 33-page PDF.
No training replication or author-code execution.

Presentation prepared as a paper discussion, not by the paper authors.