Memory for Large Language Models

Storage, writing, and retrieval as architectural choices

Paper authors: Sining Zhoubian, Dan Zhang,
Evgeny Kharlamov, and Jie Tang

Reading presentation • 4 October 2026
arXiv:2607.25380v1 (28 July 2026)

Memory quality depends on what is stored, how it changes, and whether it can still be recovered.

The survey connects mechanisms that share a problem

The same old fact can survive in different places

A token KV entry, a compressed state, writable parameters, or an independently indexed record.

- **Scope:** model architecture and inference dynamics.
- **Contribution:** a taxonomy, update-rule vocabulary, and evaluation agenda.
- **Evidence:** synthesis of prior work; no new controlled leaderboard.

Paper, Secs. I–II, PDF pp. 1–6. Agent orchestration and prompt-level memory are outside the main scope.

Three axes answer three different questions

Axis	Question	Distinction
Representation	Storage and access interface	Implicit / explicit
Update dynamics	Timing of updates	Offline / online
Persistence	Effective time horizon	Short-term / long-term

Two crucial separations

Online updates need not use gradients.
Long-horizon influence need not imply cross-session recall.

Paper, Sec. II-A–C, PDF pp. 2–5.

In this taxonomy, KV tensors are still implicit memory

Mechanism	Form	Update	Horizon
Transformer KV cache	Implicit	Online	Short
Mamba / Gated DeltaNet	Implicit	Online	Long
Titans / TTT-E2E	Explicit	Online	Long
kNN-LM / Engram	Explicit	Offline	Long
Sparse MoE	Explicit	Offline	Long

- KV access is fixed by attention's forward computation.
- MoE changes the selected experts at inference, not their weights.
- Classify a specified memory component, not just a model name.

Selected and reformatted from Table I, PDF p. 4; definition in Sec. II-A, pp. 2–3.

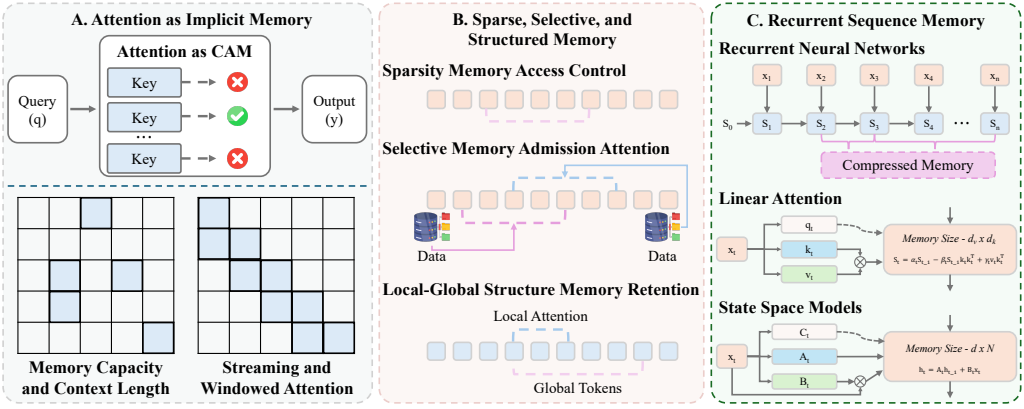
Update timing alone does not explain memory behavior

Rule family	Action	Main failure mode
Optimization-based	Fit writable parameters	Drift; objective mismatch
State transition	Recur, project, or filter	Compression; interference
Signal-gated	Decide whether to write/read	Miscalibration; delayed value
Admission / eviction / consolidation	Retain, delete, or rewrite	Irreversible information loss
Objective-induced / structural	Shape memory through training or design	Rigidity at inference

These families overlap: HAM combines recurrent updates and gated KV admission.

Reorganized from Table II, PDF p. 5; risks elaborated in Sec. VI-C, p. 17.

Implicit memory either retains items or compresses history



Read left to right: token access, selection policies, recurrent states.

Original Fig. 3, PDF p. 7. Conceptual diagram, not an empirical comparison.

Attention has separate storage and access costs

$$y_t = V_t^\top \operatorname{softmax}\left(\frac{K_t q_t}{\sqrt{d_k}}\right)$$

$$K_t \in \mathbb{R}^{t \times d_k}, \quad V_t \in \mathbb{R}^{t \times d_v}, \quad q_t \in \mathbb{R}^{d_k}.$$

- **KV storage:** $\Theta(t(d_k + d_v))$ per layer/head.
- **Full-sequence attention work:** quadratic in sequence length.
- Compressing entries, reading fewer entries, and evicting entries change different costs.

MLA: representation MoBA/NSA: access StreamingLLM: retention

Teaching formula; paper, Sec. III-A–B, pp. 6–8; KV scaling in Sec. V-B(a), p. 16. Fixed dimensions for scaling.

A delta write corrects the current association

$$M_t = M_{t-1} + \eta_t \underbrace{\left(v_t - M_{t-1} k_t\right)}_{\text{prediction residual}} k_t^\top$$

$$M_t \in \mathbb{R}^{d_v \times d_k}, \quad k_t \in \mathbb{R}^{d_k}, \quad v_t \in \mathbb{R}^{d_v}.$$

For a unit-norm key

$M_t k_t = (1 - \eta_t) M_{t-1} k_t + \eta_t v_t$.
With $\eta_t = 1$, the new value replaces the old prediction along this key.

Other, non-orthogonal keys can still be affected: state interference.

Teaching derivation, not a full Gated DeltaNet variant. Motivation: Sec. III-C, PDF pp. 8–9.

Recurrent memory improves through editing and allocation

Change the write

- Input-dependent decay: GLA, RWKV, Mamba.
- Erase/write control: GDN-2, RWKV-7, KDA.
- Calibrated updates: projection and Kalman-style filtering.

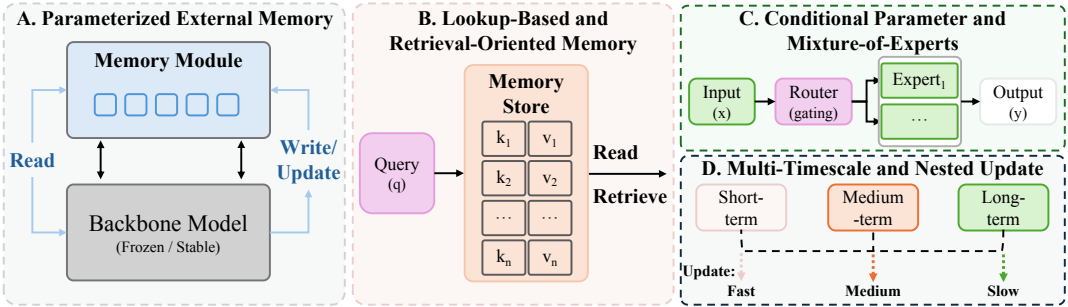
Change the capacity

- Multiple routed states: MoM.
- Hierarchical states: Log-Linear Attention.
- Cached state snapshots: Memory Caching.

Long streams do not imply unlimited exact recall.

Paper, Sec. III-C–D, PDF pp. 8–9; Memory Caching in Sec. V-B(a), p. 16.

Explicit memory assigns a storage role to a component



- A writable module can be parametric or slot-based.
- A lookup store can remain fixed throughout inference.

Original Fig. 4, PDF p. 10; Sec. IV. The taxonomy concerns interfaces, not whether storage contains tensors.

Online writing separates a memory state from the backbone

$$\phi_{t+1} = \phi_t - \eta_t \nabla_{\phi_t} \mathcal{L}_{\text{write}}(\phi_t; x_t)$$

- ϕ_t: designated memory parameters; $\mathcal{L}_{\text{write}}$: write objective.
- Titans / TTT-E2E: optimization-based writing. GDWM adds write control.
- In-Place TTT reuses an FFN matrix; MEMORYLLM instead rewrites latent slots through forward updates.

The key design question

Which input is worth writing, and how much should it change memory?

Generic teaching rule; Sec. IV-A, PDF pp. 10–11. Momentum, decay, and causal read/write order are model-specific.

One gradient step can be exactly a recurrent update

$$\mathcal{L}(M) = \tfrac{1}{2}\|Mk_t - v_t\|_2^2$$

$$\nabla_M \mathcal{L} = (Mk_t - v_t)k_t^\top$$

$$M_t = M_{t-1} + \eta_t(v_t - M_{t-1}k_t)k_t^\top$$

Implication for the taxonomy

"Optimization" and "state transition" can describe the same operation.
Inspect the write interface and lifecycle, not the method's name alone.

Teaching derivation for linear memory and a squared loss. Paper discusses the TTT–linear-attention boundary in Sec. IV-A(c), p. 11.

Lookup selects a record; MoE selects a transformation

Lookup memory

kNN-LM: context–target records.

Engram: structured hash lookup.

PlugLM / ExplicitLM: editable or interpretable banks.

Conditional parameter memory

$$y = \sum_{i \in \mathcal{S}(h)} g_i(h)\,\mathrm{Expert}_i(h)$$

$\mathcal{S}(h)$: selected experts.
$g_i(h)$: routing weights.

Mixtral example: select 2 of 8 experts per layer.

Sparse activation reduces active work; total storage and data movement still matter.

Paper, Sec. IV-B–C, pp. 11–13; original Eq. (1), p. 12. MoE-as-memory is the survey's interpretation.

Update frequency and information lifetime are different



- Nested schedules distribute plasticity across parameter subsets.
- Control **frequency, magnitude, and scope** of each update.
- Saving, resetting, and consolidating state determine its lifecycle.

Teaching schematic, not the paper's schedule. Sec. IV-D–E, PDF pp. 13–14.

Adaptive hybrids gate different resources

AMOR: when to compute

Recurrent backbone produces a prediction.

High normalized output entropy triggers attention refinement.

HAM: what to retain

Recurrent memory processes every token.

Hard-to-predict tokens enter a sparse KV cache.

- Fixed hybrids allocate layers, heads, or channels in advance.
- Adaptive controllers allocate work or storage by input.
- A low-surprise event today may be essential to a future query.

Paper, Sec. V-A(b), PDF p. 15; delayed relevance in Sec. VI-C, p. 17.

System optimizations act at different memory layers

Technique	Changes	Interpretation
KV quantization / low rank	Entry representation	Fidelity vs. bytes
PagedAttention	Physical allocation	Fragmentation and reuse
Windows / sparse global links	Accessible positions	Recency and global anchors
Memory Caching	Recurrent snapshots	Extra levels of state capacity
Bottlenecked Transformers	Periodic cache processing	Working-memory consolidation

Better allocation is not itself a new retention capability.

Paper, Sec. V-B, PDF p. 16. Reformatted comparison.

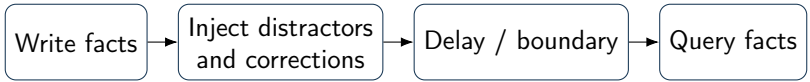
Memory evaluation needs more than a context length

Dimension	Diagnostic question
Recall fidelity	Can the target fact be recovered among distractors?
Integration	Can distant evidence support a structured answer?
Persistence	Does the fact survive delay, updates, and boundaries?
Interference	Do later writes corrupt older associations?
Efficiency	What storage, latency, and write cost buy that accuracy?

NIAH / RULER probe retrieval; broader long-context tasks also involve reasoning.

Paper, Sec. V-C, PDF pp. 16–17; Sec. VI-F, p. 18. No unified empirical metric is established here.

A fair protocol controls both interference and budget



- Vary fact count, key similarity, delay, and reset policy.
- Compare token KV, fixed states, hybrids, and disabled-write controls.
- Include TTT updates, retrieval storage, optimizer state, and routing costs.

Report curves, not only a best score

Recall $A(\Delta, I, B)$ over delay Δ, interference I, and budget B.

Presenter-proposed protocol, motivated by Sec. V-C and Sec. VI-F, pp. 16–18. Not an experiment in the paper.

The taxonomy is useful, but its boundaries need care

1. **Interfaces form a spectrum.** KV retrieval, activation banks, writable matrices, and experts differ in autonomy.
2. **Classification levels can differ.** Next-Latent Prediction is an offline training intervention; recurrent state updates are online operations.
3. **Long-term is a coarse label.** State retention, useful influence, and exact cross-session recall need separate tests.

Critical reading of Sec. II, pp. 2–6; Sec. III-C(g), p. 9; Sec. IV-A–C, pp. 10–13.

The open problem is reliable allocation over time

Mechanisms

- Measure capacity, fidelity, and interference together.
- Accumulate knowledge without destructive drift.
- Diagnose and correct unreliable write signals.

Deployment and evaluation

- Adapt storage and compute to the input.
- Match access patterns to hardware memory tiers.
- Standardize controlled retention and cost tests.

Paper, Sec. VI-A–F, PDF pp. 17–18. Research agenda, not demonstrated solutions.

Read a memory architecture through four concrete objects

Storage · Read path · Write path · Reset boundary

1. The three axes locate the mechanism.
2. The update rule explains its behavior and failure modes.
3. Recall–interference–cost curves test whether it is useful.

Discussion question

If a compressed state predicts an item well now, should we still cache it for a future exact query?

Synthesis of the survey. Zhoubian, Zhang, Kharlamov, and Tang, *Memory for Large Language Models*, arXiv:2607.25380v1. Main talk ends here; remaining slides are backup.

Backup: state editing has several independent directions

Direction	Examples reviewed in the paper
Structured dynamics	Mamba-3: richer discretization, complex states, MIMO
Separate erase and write	Gated DeltaNet-2; RWKV-7 uses richer vector controls
Channel-wise transition	KDA: diagonal decay with low-rank state updates
Compose transformations	DeltaProduct: multiple Householder updates
Closed-loop correction	Comba: scalar-plus-low-rank transitions and feedback
Calibrate the write	Kaczmarz projection; Kalman-style state estimation

Paper, Sec. III-C, PDF pp. 8–9. Mechanism descriptions, not a ranking of these methods.

Backup: a write can disturb other keys

Let $r_t = v_t - M_{t-1}k_t$. For another query q,

$$(M_t - M_{t-1})q = \eta_t r_t (k_t^\top q).$$

- Orthogonal keys do not interfere in this one-step linear model.
- Correlated keys couple the corrected association to other reads.
- Larger state dimension helps only if representations and writes use it effectively.

For scalar decay, $s_t = \lambda s_{t-1} + u_t$, a past input contributes $\lambda^r u_{t-r}$ after r steps.

Teaching derivations; linear delta update, and scalar $0 < \lambda < 1$. Motivated by Sec. III-C–D, pp. 8–9.

Backup: similar names can hide different memory substrates

Method	Distinction that matters
MEMORYLLM (2024)	Online, forward-updated latent memory pool
MemoryLLM, FFN (2026)	Offline-trained, context-free token-level FFN lookup
LM2	Dedicated slots; classified offline in Table I
In-Place TTT	A standard FFN matrix receives special online write semantics
LongMem / Memorizing Transformers	Indexed activation stores: a boundary class in this survey

Always identify which object changes when calling a mechanism online.

Paper, Table I, p. 4; Sec. IV-A–B, PDF pp. 10–12.

Backup: hybrids also differ in how they are obtained

- **LightTransfer / Priming:** replace selected attention layers and adapt or align the replacements.
- **Hydra:** combine state-space computation, sparse attention, expert routing, and distinct memory roles.
- **Expansion Span:** allocate attention span to retrieved tokens beyond the ordinary finite window.
- **Nested updates:** combine token-level states with parameter components updated at different frequencies.

Paper, Sec. V-A(a,d,e), PDF pp. 14–15. These are complementary design choices.

Backup: map a benchmark to the capability it probes

Evaluation family	Interpretation
NIAH / RULER	Retrieval and degradation with position or context length
LongBench / L-Eval / ∞Bench	Broad long-context use; separate task categories
SCROLLS / NarrativeQA	Integrating distributed evidence and reasoning
Continual-update tests	Retention after new writes; update consistency
Resource sweeps	Quality at fixed storage, compute, or latency

Perplexity measures average prediction quality; precise recall asks a different question.

Paper, Sec. V-C, pp. 16–17; ∞Bench also in Sec. III-A, p. 7 and Fig. 2. No common scoring protocol is provided.