Online ISSN: 3104-9338
Print ISSN: 3104-932X

Remembering Beyond the Context Window: The Capacity, Consistency, and Control Gaps in Long-Term Memory for LLM Agents

Weibei Deng, Xinyu Song*

Geely University of China, ChengDu, China, 641423

Received: August 20 2026

Revised: August 30, 2026

Accepted: August 30, 2026

Published online: August 30, 2026

To appear in: *International Journal of Advanced AI Applications*, Vol. 2, No. 9 (September 2026)

* Corresponding Author: Xinyu Song (songxinyu@guc.edu.cn)

Abstract. Large language model (LLM) agents are increasingly expected to remember users, tasks, and outcomes across sessions, and a dedicated memory infrastructure has emerged to meet that expectation. This survey reviews 21 core sources (2020–2025) on agent memory through an original three-gap framework—capacity (how much is retained), consistency (whether what is retained remains correct), and control (whether retention can be governed)—drawing only on peer-reviewed or publicly verifiable sources for every reported figure. The architecture side has matured rapidly across three generations: operating-system-inspired prototypes, production memory pipelines, and neurobiologically inspired retrieval. The evidence side has not kept pace. Effective context length measured by controlled benchmarks falls far below nominal context length, positional effects corrupt mid-context recall, external memories routinely conflict with parametric knowledge, and knowledge-editing procedures fail to propagate beyond the edited fact. Unlearning remains fragile exactly where retention requirements make it legally consequential. We argue that memory must be treated not as a performance feature but as an auditable record, and map the three gaps onto evaluation standards, privacy regulation, and the emerging requirements for inspectable memory lifecycles.

Keywords: *knowledge editing; machine unlearning; knowledge conflicts; retrieval-augmented generation; evaluation benchmarks; data protection*

1. Introduction

1.1. The Context Paradox

Throughout 2024, frontier model developers competed to extend context windows past the million-token mark, and claimed context length has effectively become a headline specification—so much so that the authors of the RULER benchmark open their paper by asking what the real context size of long-context models is [1]. Their measurements supply the uncomfortable answer: across a suite of tasks that progress from simple retrieval to multi-hop aggregation, many models whose context windows are advertised at 128,000 tokens or more exhibit effective lengths far below the claimed figure, and the shortfall widens as the task requires genuinely using the context rather than searching it [1]. Positional structure compounds the shortfall. Even when the required information sits inside the window, models systematically favor material at the beginning and end of the context and degrade sharply in the middle—the "lost in the middle" curve that has become the canonical picture of how models actually read long inputs [2].

The result is a paradox worth naming: as nominal context has grown by orders of magnitude, evidence of effective memory has not grown alongside it, because window length is an engineering specification, not a demonstration of retention. For conversational use the paradox is merely inconvenient. For agents—systems that pursue tasks over multiple sessions, accumulate user preferences, and are expected to act on what happened yesterday—it is structural. The context window is transient by design: it is rebuilt at every session and priced by the token, so nothing that matters tomorrow can safely live only there.

Two properties of agent workloads make the transience decisive rather than merely inconvenient. The first is duration asymmetry: a context window holds one session's worth of material, while an agent's obligations run on calendars—a standing instruction ("always book the aisle seat"), a mid-project correction ("we switched vendors in March"), a preference learned months ago. The window cannot hold the timescale on which its own contents were established, so any durable obligation must be re-externalized at every session or silently lapse. The second is composition: agent tasks rarely turn on a single remembered item; they require the item plus everything it has come to entail across sessions, which is precisely the aggregated, multi-hop use of memory that the capacity benchmarks show models are worst at inside the window [1]. Together the two properties explain why memory infrastructure emerged on top of growing windows rather than being displaced by them: bigger windows changed the constants of the problem, not its structure—the durable still must live outside the transient, and the

composed still must be assembled from parts. The survey's object is that external, durable, composable layer, and whether its claims can be believed [1]. This is why a distinct memory infrastructure has emerged, external to the model and persistent across sessions, and why the credibility of that infrastructure is now a question in its own right. This survey takes up that question.

The emergence of that infrastructure can be dated and located. Within roughly two years, memory modules moved from research demos to production: assistant products shipped persistent user-memory features; developer platforms exposed memory APIs as a service tier; and open-source memory frameworks accumulated thousands of adaptations. The design pressure is uniform across offerings—an agent that serves a user over weeks must present the same preferences, commitments, and history that a human assistant would retain—and so is the architectural answer: a store external to the model, a policy for writing into it, a retrieval mechanism for reading from it, and increasingly, mechanisms for consolidating and forgetting. What is not uniform is the evidential basis: each product's claims about what its memory retains, how current it stays, and how completely it can be erased are, as the following sections document, mostly unaudited. The survey's wager is that this asymmetry—industrial build-out, artisanal evidence—is the field's defining condition and the right thing for a survey to organize.

1.2. Scope and Method

We focus deliberately on agent memory systems—explicit, inspectable stores coupled to LLM agents—rather than on parametric memory acquired through training. Parametric knowledge serves here as a comparison class: what models "remember" in their weights shapes the conflicts and editorial problems reviewed in Section 3, but the survey's object is the external memory stack. We searched arXiv and the proceedings of major machine-learning conferences for 2023–2025 system and benchmark papers, complemented by first-party product documentation and European Union legislative texts. Sources were retained if they (i) propose a systematic memory architecture for LLM agents, (ii) introduce an evaluation instrument specific to long-term memory, or (iii) report empirical results on manipulating or attacking memorized content (editing, unlearning, extraction). Generic retrieval-augmented question answering was included only where it bears directly on agent memory design. Reported figures are quoted only from the original papers, first-party documentation, or the legislative texts themselves. This procedure yielded the 21 core sources organized below; the two benchmark families and the editing-literature results that anchor Section 3 were each verified against their primary sources.

A note on evidential discipline, because a survey about memory claims must hold its own claims to a stricter standard than the systems it reviews. Every number in this survey is quoted from the source that produced it—benchmark papers for benchmark results, system papers for system measurements, first-party documentation for product behavior—and each source's author list and bibliographic record were verified against its primary publication before inclusion; sources that could not be verified were dropped rather than cited from recollection. Where this survey characterizes a system's behavior, it reports what the system's own paper or documentation states, marking interpretive gloss as its own. The discipline is not decorative: much of the memory literature circulates through secondary summaries whose numbers drift, and a survey that asks memory systems for auditable records cannot itself rely on unauditable ones. Readers should find that every quantitative claim herein resolves to a named primary source in one step—the same property Section 4 will argue deployed memory systems should have.

1.3. Positioning and the Three-C Framework

The field's self-understanding is framed by two recent syntheses, and both differ from this survey in what they take to be the organizing problem. Zhang et al. survey memory mechanisms for LLM-based agents, organizing the material by how memory is built: what is stored, in what form, where, and when it is read [3]. Sumers et al. propose cognitive architectures for language agents, subordinating memory to a broader vocabulary of action and decision-making drawn from cognitive science [4]. Both organize the field by design. Neither organizes it by evidence—by what is actually known about whether the designs retain enough, stay correct, and can be governed. Table 1 contrasts the three framings.

Table 1. Coverage comparison between recent syntheses of agent memory and this work.

Dimension	Zhang et al. [3]	Sumers et al. [4]	This survey
Primary scope	Memory mechanisms of LLM agents	Cognitive architectures for agents	Evidence status of agent memory
Organizing logic	What, where, when to store	Memory within action and decision loops	Capacity, consistency, control
Failure evidence	Selected examples	Conceptual	Quantified benchmarks, editing and extraction studies
Governance treatment	Brief	Absent	Unlearning, privacy regulation, auditable lifecycles

Our framework poses three questions of any memory claim. Capacity: how much does the

system actually retain and use—measured, not advertised? Consistency: does what it retains remain correct when the world changes, when retrieved content conflicts with parametric knowledge, or when one stored fact implies another that was never updated? Control: can what is remembered be inspected, corrected, and removed on demand, by operators and users with legitimate authority? The three are ordered: capacity without consistency is confident error at scale, and neither matters if memory cannot be governed.

The ordering deserves defense, because the field's incentives run the other way. Capacity is the glamorous property: it sells products, anchors headlines, and is the dimension along which systems are most often compared. But a memory system's failures grade sharply in severity as one moves down the list. A capacity failure—forgetting a preference, missing a detail—degrades usefulness. A consistency failure—acting on a corrected address that was "updated" in the store but not in the agent's behavior, re-asserting a rescinded instruction—creates harm with a paper trail that says otherwise: the system's record shows the right thing while its behavior does the wrong one, which is the worst evidentiary position a governed system can occupy. A control failure—no verified way to remove what a user demanded removed—converts both prior failures from incidents into liabilities, because there is no remediation after the fact. The order is therefore also a priority order for governance: the sections that follow assess the evidence for each gap in turn, and the evidence, as will be seen, degrades precisely down the list—the capacity gap is the best measured, the control gap hardly measured at all. Section 2 reviews the architectural generations the field has already built; Section 3 assesses the evidence for each gap; Section 4 draws the governance consequences; Section 5 concludes.

2. The Agent Memory Stack: Three Architectural Generations

2.1. Prototypes: Operating Systems and Memory Streams

The founding metaphor of agent memory is the operating system. MemGPT reframes the context window as main memory and external storage as disk, and has the LLM itself interrupt its generation to page information between the two—searching archives, revising its own working contents, deciding what deserves scarce in-context space [5]. The significance of the design is less the particular mechanism than the transfer of authority: memory management ceases to be a fixed developer-imposed pipeline and becomes behavior of the model at run time. Generative Agents, simultaneously, demonstrated what a rich memory stream makes possible: every experience is recorded as an observation; retrieval weights recency, importance, and relevance; and a periodic reflection operation synthesizes lower-level memories into higher-

level inferences that are themselves stored and retrievable [6]. The simulated small town built on this machinery exhibited emergent social behavior—one agent organizing a Valentine's Day party that other agents attended—that no single prompt produced [6].

Together the two prototypes defined the agenda that later systems industrialize: memory as an explicit, self-managed object rather than a passive context buffer. They also fixed a pattern that Section 3 returns to: both systems were validated primarily by demonstration and simulation—emergent behavior, human evaluations of believability—rather than by instrumented measurement of what the memory layer itself retained, weighted correctly or forgot on schedule. The architectural claims ran ahead of the evidential ones from the start, and the production generation inherited the asymmetry along with the machinery.

Each prototype also contributed a mechanism that later systems treat as standard, and the mechanisms carry governance weight their papers did not claim for them. MemGPT's paging established that the model itself can decide what enters scarce memory—an architecture in which retention policy is emergent model behavior rather than developer specification [5]. Generative Agents' reflection established that the store can rewrite itself, synthesizing observation-level memories into inferences that enter the store as first-class entries with their own retrieval weight [6]. Both mechanisms are why memory contents are hard to predict from the outside: what an agent remembers is partly a function of its own judgments at run time, which means an auditor cannot enumerate the store by replaying the inputs—the store is a product of decisions, not a log. The distinction between a log (append-only, replayable, complete by construction) and a memory (curated, synthesized, self-modifying) is the prototypes' deepest legacy, and Section 3's control gap is in large part the absence of any evaluated answer to the question it poses: when the record is a distillation, what does it even mean to inspect or correct it? [5,6]

The prototypes also explain why the memory literature's vocabulary is so unevenly operationalized. Terms that later papers use as engineering nouns—episodes, reflection, importance, forgetting—entered the field as design metaphors from the two systems, carrying their simulation-era connotations with them; MemoryBank's forgetting curve is Ebbinghaus applied as a decay function rather than as psychology, and A-MEM's "evolved context" is Generative Agents' reflection generalized to link propagation. Metaphor inheritance is not a defect—every engineering field borrows its early vocabulary—but it does mean the field's central terms name mechanisms before they name measured properties: no paper in this survey's corpus that says "forgetting" reports a forgetting rate, and none that says "reflection" reports a

reflection accuracy. The vocabulary gap is the gaps of Section 3 in miniature, visible already at the prototypes' origin: a system vocabulary that outruns its measurement vocabulary at birth tends to keep the lead, and the production generation's benchmark suites—valuable as they are—did not close it [5,6].

2.2. Production Systems: Memory as a Service

The second generation packages memory as deployable infrastructure. Mem0 extracts candidate memories from conversations, consolidates them against existing entries—adding, updating, or deleting—and serves them back through a retrieval pipeline, reporting on the LOCOMO long-conversation benchmark both accuracy exceeding full-context baselines and large savings in latency and token cost [7]. The comparison matters: LOCOMO itself evaluates agents on conversations hundreds of turns long and finds that faithfully answering requires aggregating evidence scattered across them, a capability in-memory pipelines are still failing at differentially across information types [8]. Memory-as-a-service, in other words, is already being benchmarked against the hardest thing memory must do—keeping a coherent account of a long shared history—and the benchmarking is genuinely adversarial. A-MEM pushes the agentic organization of memory further: when a new note arrives, the system dynamically generates links to related prior notes and propagates an evolved contextual description backward through the network, so the memory structure itself keeps reorganizing as the agent's history grows [9]. MemoryBank approaches the same lifecycle from the temporal side, attaching an Ebbinghaus-style forgetting curve to stored memories so that unreviewed content decays and a long-term companion persona reflects the natural attrition of an organic relationship [10].

The production generation's distinctive commitment is lifecycle management—write, consolidate, link, decay, delete—as an explicit service boundary. What it has not yet produced is a shared evidential standard for the boundary: each system is evaluated on partly different suites, and the persuasive force of the numbers lies in comparisons against context baselines rather than against the retention, correctness, and removability guarantees that the lifecycle language implies.

Two details of the production generation's own evidence qualify its headline numbers, and both matter for what the systems can be claimed to do. First, the strongest benchmark results are comparative: Mem0's reported gains are measured against full-context and baseline-pipeline configurations on LOCOMO, which establishes that the pipeline beats the alternatives tested—not that it retains, stays consistent with, or can verifiably forget any particular fact category [7].

Second, the benchmark that anchors the comparison evaluates memory at a difficulty deliberately beyond current systems: LOCOMO's conversations run to hundreds of turns with evidence deliberately scattered, and its own analysis shows systems failing differentially across information types—multimodal and temporal items hardest—which means even the favorable comparisons sit on a floor of substantial absolute error [8]. Neither qualification is an accusation; both are what honest benchmark evidence looks like. The governance reading is about the gap between the evidence and the marketing: a pipeline whose measured behavior includes substantial failure on temporal and scattered evidence is being sold, in lifecycle language, as a system that keeps the record current. The lifecycle vocabulary arrived before the lifecycle guarantees did [7,8].

2.3. The Retrieval Frontier: Indexing Before Storing

Beneath the system papers lies a retrieval lineage. Retrieval-augmented generation established that grounding generation in documents fetched at query time is the workable alternative to cramming knowledge into parameters or context [11]. HippoRAG imports the neurobiology of episodic memory to improve that grounding: an offline step builds a knowledge graph from the corpus as an artificial hippocampal index, and retrieval runs a personalized PageRank over it, mimicking the pattern-completion dynamics of hippocampal indexing theory; the result is markedly stronger multi-hop question answering than dense retrieval alone, at a fraction of the cost of iterative agentic retrieval [12].

Read across its three generations, the stack shows a consistent trajectory: memory decisions migrate from the developer's blueprint into run-time model behavior—paging, reflecting, linking, forgetting—and the store itself becomes richer and more structurally self-modifying. Table 2 summarizes the generations.

The retrieval lineage's governance property is worth stating precisely, because it is the stack's best and rarely advertised. A retrieval-indexed memory has the one property regulators and auditors know how to work with: enumerability. The store's contents can be listed; a given memory is either in the index or not; a deletion is an index operation whose effect is complete and immediate; and the provenance of any retrieved item is, in principle, a field in its record. RAG established the architecture [11], and HippoRAG's contribution—the graph index with spreading-activation retrieval—compounds the property rather than departing from it: a graph over notes is still an enumerable structure, and its personalized-PageRank retrieval is a deterministic function of the graph, so multi-hop influence is traceable in a way that parameter-internal influence is not [12]. When later sections document the capacity, consistency, and

control gaps, the retrieval frontier is where the correctives are cheapest to install—and the production systems that emphasize neural consolidation over indexed structure are, knowingly or not, trading away enumerability, the very property that governance will eventually ask them to produce. The stack's most sophisticated corner is thus its least auditable, by design rather than by accident. The trajectory is the field's strength and its exposure: the same autonomy that makes memory useful makes its contents harder to predict from the outside, which is why the gaps of Section 3 fall where they do.

Table 2. Three generations of agent memory systems.

Generation	Representative systems	Organizing metaphor	Who manages memory	Evidence style
Prototypes (2023)	MemGPT [5], Generative Agents [6]	OS paging; memory stream	The model, at run time	Simulation, human evaluation
Production (2024–25)	Mem0 [7], A-MEM [9], MemoryBank [10]	Memory as a service	Dedicated pipeline	Benchmark suites, e.g., LOCOMO [8]
Retrieval frontier (2024)	RAG [11], HippoRAG [12]	Hippocampal indexing	Offline index + propagation	Multi-hop QA accuracy

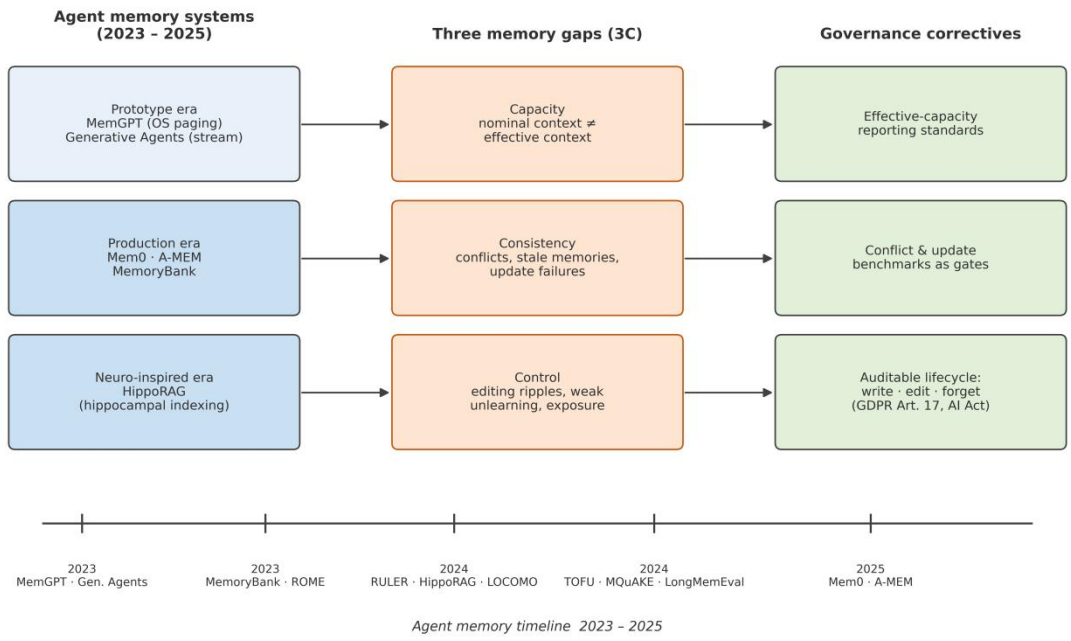

Figure 1. The three memory gaps of agent memory systems. Left: three architectural generations of the memory stack. Middle: the capacity, consistency, and control gaps separating built systems from evidenced memory. Right: the governance correctives each gap demands. Bottom: the field's timeline, 2023–2025.

3. Three Gaps: Capacity, Consistency, Control

The systems of Section 2 exist and are deployed. This section asks what is actually known about them, gap by gap. The organizing claim is that each gap is not a missing feature but a mismatch between what has been built and what has been measured—and that the mismatch is widest precisely where deployment pressure is strongest.

3.1. The Capacity Gap: Nominal Is Not Effective

The capacity gap is metrological: nominal context length is a configuration, effective context length is a measurement, and the two diverge systematically. RULER operationalizes the difference with a task ladder whose rungs demand progressively deeper use of the context—from simple needle retrieval through variable tracking to multi-hop aggregation across many distributed items—and shows that a model's effective length is not one number but a function of task complexity, with models that hold up on retrieval collapsing well before their claimed limits on aggregation [1]. The practical reading is direct: a context window advertised at 128,000 tokens is not a promise that 128,000 tokens are usable; on the harder rungs, measured capacity for several models falls to a fraction of the claim [1]. Positional structure further degrades the usable fraction, since mid-context material is systematically underweighted even when retrievable in principle [2].

Economics closes the trap from the other side. Attention cost grows with context, so retention by stuffing is priced per token and per query—an economic boundary independent of any benchmark. The memory stack of Section 2 is therefore not a temporary workaround awaiting bigger windows; it is the rational response to a window that is expensive, transient, and only fractionally effective. But the same reasoning transfers the burden: once memory lives in an external store, the system's true capacity is whatever the combined pipeline retains and uses, and that quantity—external store, retrieval quality, in-window positional effects compounded—is precisely what no headline specification measures. Capacity, in short, is currently claimed in tokens and evidenced in benchmarks; the gap between the two currencies is the first thing an operator must audit.

The measurement details sharpen the point from abstract to actionable. RULER's task ladder is public and parameterized, which means an operator can locate where on the ladder their deployed configuration begins to fail—and the interesting finding is that the failure point is not a fixed token count but moves with task depth: the same model that retrieves a needle at 90,000 tokens may fail to aggregate across 40 needles at 30,000 [1]. The lost-in-the-middle curve adds

a positional dimension to the audit: it is not only how much is in the window but where—the same information placed in the middle of a long context is substantially less likely to be used than when placed at the edges, a U-shaped usability surface inside every nominally uniform window [2]. For agent memory design the two results compose into a siting problem: an external store that stuffs retrieved history into the middle of a long prompt is placing its most valuable contents in the least reliable real estate of the window, and an operator who sizes the retrieval budget against the nominal window rather than the effective one is systematically overfilling the prompt. Capacity measurement, in other words, is not merely a benchmark formality—it dictates concrete retrieval-configuration decisions that production systems currently make against the wrong number [1,2].

3.2. The Consistency Gap: Conflicts, Updates, and Wrong Recency

A memory that grows across sessions must survive a changing world, and here the evidence is most concerning. Start with conflict. Xie et al. show that when contextual evidence contradicts parametric knowledge, model behavior is unstable in a characteristic way: performance on knowledge-conflict tasks drops compared to clean settings, and models frequently persist in stale parametric answers even against explicit contextual evidence [13]. Agent memory industrializes this hazard, because external stores accumulate exactly the kind of plausible, fluent, occasionally outdated content that competes with the model's prior beliefs at every retrieval. The benchmark evidence concurs from the evaluation side. LongMemEval decomposes long-term interactive memory into five abilities—information extraction, multi-session reasoning, temporal reasoning, knowledge updating, and abstention—and finds knowledge updating and temporal reasoning among the weakest for commercial chat assistants, with accuracy dropping substantially on long-context variants of its suite [14]. LOCOMO reaches a compatible verdict from conversation scale: across hundreds of turns, the systems it evaluates struggle to correctly aggregate and recall information distributed across a shared history, with failures concentrated on exactly the multi-session and temporal categories that memory exists to serve [8]. Memory, in other words, is weakest at its defining task—keeping the record current.

LongMemEval's decomposition is the most diagnostic instrument in the corpus because it separates the failure modes that "memory" blurs together, and the separation changes what one would fix. Its five abilities—information extraction, multi-session reasoning, temporal reasoning, knowledge updating, and abstention—are scored separately, and the published pattern is that knowledge updating (revising a stored fact when the user contradicts it) and

temporal reasoning (ordering and dating remembered events correctly) sit at the bottom of commercial systems' performance [14]. Abstention is the most consequential of the five and the least discussed: knowing that the store does not contain the answer, and saying so, is what prevents an agent from confabulating a plausible detail on top of an empty retrieval—and it, too, degrades as sessions lengthen [14]. A system designer reading the decomposition gets an actionable map: temporal failures call for timestamp discipline in the store and schema, updating failures call for the propagation mechanics of the editing literature, abstention failures call for calibrated retrieval confidence rather than more capacity [14]. The single-number memory scores that circulate in marketing hide exactly the distinctions that determine which engineering fix applies—which is why the benchmark's decomposition, not any aggregate, is the consistency gap's real measurement [14].

Editing studies sharpen the point into a structural claim. Model-editing methods can rewrite a stored association in place—rank-one updates locate and modify the weight positions that mediate a factual recall [15]—and yet MQuAKE shows that such edits fail to propagate: correcting one fact leaves the model asserting the old one whenever the answer requires traversing it through a multi-hop chain, so the correction is present in the store and absent from the reasoning [16]. For agent memory the implication is uncomfortable: consistency is not a property of individual writes but of the inference graph over all of them, and the current tooling treats the store as a collection of independent notes. Until edit operations carry their ripple structure—until updating one memory invalidates what it entailed—consistency will be measured as an afterthought rather than guaranteed by construction.

The mechanics behind the propagation failure explain why it is structural rather than a tuning problem. ROME's causal-tracing procedure localizes a single factual association to specific mid-layer positions and rewrites them with a rank-one update—an operation whose beauty is its narrowness: it changes one association's storage while barely disturbing neighboring weights [15]. That narrowness is exactly what MQuAKE exposes on the reasoning side: a multi-hop answer ("Where was the wife of the director of film X born?") requires the corrected fact to participate in a chain, and the un-traversed edges of the chain still carry the pre-edit association [16]. Agent memory reproduces both halves at the system level: the write-side narrowness lives in memory pipelines that treat entries as independent rows, and the read-side propagation failure appears whenever generation must compose several stored memories—precisely the multi-session aggregation that LOCOMO and LongMemEval identify as the hardest categories [8,14]. Consistency, on this evidence, is a graph property being managed with list operations: until

write operations declare and propagate their entailments, no amount of per-entry correctness checking will make the store consistent, because the inconsistency lives between the entries, not inside any of them [15,16].

3.3. The Control Gap: Editing, Forgetting, and Exposure

The control gap concerns authority over the store: who may write, correct, and—critically—remove. Machine unlearning is the formal counterpart of the human right to be forgotten, and its current state is fragile. TOFU frames unlearning on fictitious author profiles, where the ground truth of what "forgotten" means is knowable, and finds that existing techniques preserve model utility well while largely failing to truly remove the targeted knowledge—and that the gap widens drastically as the amount to be forgotten grows [17]. Extraction results give the removal failure its privacy stakes: Carlini et al. demonstrate that training data, including personally identifiable information, can be recovered from large models by black-box queries, establishing memorization as an exploitable surface rather than a theoretical concern [18]. An external memory store reproduces that surface in plainer form—its contents are readable, copyable, and subject to subpoena or breach—while compounding it with durability: what a user told an agent yesterday may persist in a vendor's store long after the conversation itself is gone.

First-party practice shows both responsiveness and asymmetry. OpenAI's documentation for ChatGPT memory describes reference memories that users can view, correct, and delete individually, and sessions whose memories are cleared—but the decisions of what to write in the first place remain interior to the product, opaque to the user whose history supplies them [19]. The asymmetry is the gap in miniature: removal has a user-facing control surface, while writing and retention-policy do not. No benchmark in the survey's corpus measures write justification, retention horizon, or deletion completeness as first-class properties of a memory system. Control, at present, is a set of scattered affordances rather than an evaluated property.

The extraction evidence supplies the control gap's adversarial boundary, and it is worth stating in the store-specific form. Carlini et al.'s methodology— inducing the model to diverge from its fluency prior so that memorized, high-perplexity training text surfaces—shows that "the model does not reveal it on normal queries" is not a privacy property but a latency property: the data stays in the artifact, and the artifact's resistance is empirical, not structural [18]. An external memory store makes the corresponding point without any attack: its contents are directly readable, exportable, and subject to legal process, so the question for the store is never whether the data can be extracted but who may read the store and on what showing. That inverts

the security posture relative to parameters—in weights, extraction is the attack; in the store, access is the attack—and governance instruments exist for the second case. What does not yet exist is the verification layer either way: whether data was truly removed from weights or truly deleted from the store, the operator's claim currently cannot be checked by the data subject, and TOFU's probing protocol is the only published template for what such a check would look like [17]. Control, read adversarially, is therefore two problems—access control on stores, extraction resistance on parameters—plus one shared missing piece: verifiable deletion, which neither the store operators nor the model providers currently offer evidence for [17,18].

4. Governance: Memory as an Auditable Record

4.1. Evaluation as Infrastructure

The first corrective is already under construction: the benchmark families of Section 3. If capacity must be reported as effective length under a task ladder, consistency as update and conflict performance, and control as deletion and abstention behavior, then LongMemEval's five-ability decomposition and LOCOMO's long-conversation protocol are the beginnings of a metrology for memory [8,14]. What benchmarks do well is make claims comparable and regressions visible; what they cannot do is certify behavior on future inputs. The governance reading is therefore modest and concrete: procurement and deployment decisions should treat effective-capacity and update-failure figures as gating information, exactly as throughput figures gate other infrastructure purchases.

There is a precedent for benchmarks becoming governance artifacts, and it explains why the modest reading is not a small one. Seat-belt crash ratings, university rankings, and financial stress tests all began as measurement exercises whose published numbers acquired regulatory and market force—the measurement became the governance, without any legislature involved. The memory benchmarks are positioned identically: RULER's task ladder, LongMemEval's five abilities, and LOCOMO's long-conversation protocol are public, parameterized, and comparable across systems [1,8,14], which is precisely the property that lets a procurement officer write "effective capacity at aggregation depth $X \geq Y$" into a contract, or a regulator write "update-failure rate below Z" into a condition of deployment. The benchmarks' authors did not design them for this role—RULER asks what a model's real context size is; LOCOMO asks how agents fare over very long conversations [1,8]—but governance does not require the design intent, only the measurement's public comparability. The gap that remains is coverage: no benchmark in the corpus measures write-side justification or deletion completeness as first-

class scores, so the gatekeeping that benchmarks enable currently covers the capacity and consistency gaps and skips the control gap [8,14].

4.2. The Regulatory Surface

The legal surface is closer than much of the technical debate assumes. The GDPR grants data subjects a right to erasure of personal data, with narrow exemptions, and an agent memory store containing user history is personal-data processing in an unusually literal form: persistent, structured, attributable to a person [20]. The EU AI Act overlays risk-tiered obligations on providers and deployers, including logging and record-keeping duties for high-risk systems—obligations that presuppose the kind of inspectable memory whose absence Section 3 documented [21]. The combined reading is that European law already treats what agents remember as governed data; what is missing is the engineering correspondence—memory systems designed so that erasure requests, retention limits, and audit queries are operations with verified outcomes rather than best-effort deletions.

The regulatory structure maps onto the three gaps with uncomfortable precision, which is the survey's framework paying off analytically. The GDPR's storage-limitation principle binds what agents retain—a capacity obligation stated in law but unimplemented in any memory system's design or evaluation, since no system in Section 2 measures or even declares a retention horizon [20]. The accuracy principle of the same regulation—personal data must be kept correct and up to date—binds what agents believe: an agent acting on a stale address or a rescinded preference is processing inaccurate personal data, whatever the store's own entry says, which makes the consistency gap a compliance question rather than a quality question [20]. And the AI Act's record-keeping and logging duties for high-risk systems presuppose that what the system did and why can be reconstructed after the fact—an obligation the self-modifying stores of Section 2.2 make structurally difficult, since consolidation and reflection rewrite the very traces the logs are supposed to preserve [21]. Three legal obligations, three unimplemented engineering properties: the mapping is not a coincidence but the point—European law was written for records, and agent memory is now a record-keeping technology that has not yet accepted the description [20,21].

4.3. The Auditable Memory Lifecycle

The synthesis is to treat memory as a record with a lifecycle, and to demand for each stage the properties governance requires of records elsewhere. Write with justification: entries carry their provenance and the evidence that warranted retention. Maintain with consistency: updates

propagate to entailed content, in the direction MQuAKE showed current editing fails [16]. Forget with verification: deletion is validated by probing, in the spirit of TOFU's ground-truthed unlearning [17], and extraction attacks [18] define the adversary the verification must defeat. Answer with traceability: every retrieved memory carries an audit trail to its origin. None of these properties is exotic; all of them are measurable; and the three gaps of this survey are, in governance terms, precisely their current deficits.

A concrete deployment sketch shows the lifecycle is implementable with current parts, which matters because "auditable memory" risks reading as aspiration. On the retrieval frontier's indexed stores [12], write-side justification is a schema field; propagation on update is a graph query over entailment edges—expensive, but bounded and checkable in a way parameter-side ripples are not; deletion is an index operation, and its verification can reuse TOFU-style probing against the store rather than the weights, where ground truth is knowable by construction [17]. The hard stages concentrate exactly where the stack is least enumerable: consolidated and self-synthesized memories, whose provenance the prototypes' reflection operations deliberately blur [6], and the parametric residue of what models bring to every read. The engineering conclusion is therefore a siting decision, not a research program: put governed knowledge in the enumerable part of the stack where the lifecycle properties are cheap, and treat the non-enumerable parts as unverified by policy—subordinate to the indexed store for anything regulation touches—rather than as coequal memory. Vendors will resist the demotion; the evidence of Section 3 says the demotion is what "auditable" costs [6,12,17].

5. Conclusion

This survey organized agent memory along three gaps and reached three findings. First, the architectural side has consolidated across three generations—OS-inspired prototypes, production memory services, and neurobiologically inspired indexing—with memory management migrating steadily into run-time model behavior [5-7,12]. Second, the evidential side lags the architectural one at all three of our questions: effective capacity falls short of nominal capacity under task load [1,2]; consistency fails exactly where memory's defining duty lies, in updates and temporal grounding [8,13,14,16]; and control remains a scattered set of affordances without a measured guarantee of removal [17-19]. Third, the governance apparatus needed to close the gap is partially in place—benchmarks provide a metrology [8,14], and European law already asserts authority over what agents remember [20,21]—but no evaluated lifecycle yet connects the two.

The framework travels beyond chat agents. Any system that accumulates operational history—recommendation platforms, autonomous-process controllers, institutional knowledge bases—faces the same triad of claims outrunning evidence. For agent memory specifically, the highest-value open problems follow the gaps: reporting standards that state effective capacity under task load rather than token ceilings; editing and update operations with verified ripple propagation; unlearning whose completeness is probed rather than presumed; and write-side transparency commensurate with the deletion controls that first-party products already ship. Memory is becoming the load-bearing component of agentic systems; making its claims auditable is how the component earns the load.

The near-term trajectory the evidence supports is a division of labor rather than a single convergence. Inside the window, the capacity benchmarks will keep measuring effective retention as windows grow, and the positional economics will keep making external memory the rational default for anything durable [1,2]. Outside the window, the indexed, enumerable corner of the stack will absorb the governed workloads first—because it is where the lifecycle properties of Section 4 are cheapest to implement and verify—while consolidated, self-modifying memory earns trust exactly as fast as its write, propagation, and deletion claims become probeable [12,17]. None of this requires a scientific breakthrough; it requires the field to treat memory the way databases were eventually forced to treat durability—as a property with a definition, a test, and a failure report—rather than the way it is treated today, as a feature with a demo. The three gaps name the distance between those two treatments; closing them is, in the most literal sense, a matter of record.

References

[1] Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., & Ginsburg, B. (2024). RULER: What's the real context size of your long-context language models? arXiv preprint arXiv:2404.06654.

[2] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12, 157-173.

[3] [3] Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., & Wen, J.-R. (2024). A survey on the memory mechanism of large language model based agents. arXiv preprint arXiv:2404.13501.

[4] Sumers, T. R., Yao, S., Narasimhan, K., & Griffiths, T. L. (2024). Cognitive architectures for language agents. arXiv preprint arXiv:2309.02427.

[5] Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., & Gonzalez, J. E. (2023). MemGPT: Towards LLMs as operating systems. arXiv preprint arXiv:2310.08560.

[6] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. ACM.

[7] Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). Mem0: Building production-ready AI agents with scalable long-term memory. arXiv preprint arXiv:2504.19413.
[8] Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating very long-term conversational memory of LLM agents. arXiv preprint arXiv:2402.17753.
[9] Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025). A-MEM: Agentic memory for LLM agents. arXiv preprint arXiv:2502.12110.
[10] Zhong, W., Guo, L., Gao, Q., Ye, H., & Wang, Y. (2024). MemoryBank: Enhancing large language models with long-term memory. arXiv preprint arXiv:2305.10250.
[11] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems 33.
[12] Jiménez Gutiérrez, B., Shu, Y., Gu, Y., Yasunaga, M., & Su, Y. (2024). HippoRAG: Neurobiologically inspired long-term memory for large language models. In Advances in Neural Information Processing Systems 37.
[13] Xie, J., Zhang, K., Chen, J., Lou, R., & Su, Y. (2024). Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In The Twelfth International Conference on Learning Representations (ICLR).
[14] Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., & Yu, D. (2024). LongMemEval: Benchmarking chat assistants on long-term interactive memory. arXiv preprint arXiv:2410.10813.
[15] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. In Advances in Neural Information Processing Systems 35.
[16] Zhong, Z., Wu, Z., Manning, C. D., Potts, C., & Chen, D. (2023). MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.
[17] Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., & Kolter, J. Z. (2024). TOFU: A task of fictitious unlearning for LLMs. In First Conference on Language Modeling (COLM).
[18] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (pp. 2633-2650).
[19] OpenAI. (2024). Memory FAQ. OpenAI Help Center. https://help.openai.com
[20] European Parliament and Council. (2016). Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation). Official Journal of the European Union, L 119.
[21] European Parliament and Council. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union.