PROVABLY STABLE DUAL-LAYER MEMORY FRAMEWORKS WITH DYNAMICS FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents deployed in real-world settings must maintain long-horizon consistency, respect evolving user preferences, and ground reasoning in past experiences. While recent agent memory systemsn have advanced the engineering of storage and retrieval, none provides formal guarantees on the long-term stability of the memory statistics that govern retrieval. We address this gap by modeling memory statistic updates as a continuous-time dynamical system and applying standard ODE stability tools—Lyapunov analysis and Input-to-State Stability (ISS)—to establish convergence guarantees. Our dual-layer architecture uses a **Shared Layer** that aggregates global experiences via coarse-grained 4th-order Runge-Kutta updates and a **Private Layer** that adapts to user-specific patterns via fine-grained Euler updates anchored to the shared trajectory. The system is *training-free*: improvement comes from case accumulation and statistic-driven retrieval using frozen off-the-shelf encoders, without training any additional models. On the LoCoMo benchmark, our model achieves the highest F1 (52.35% avg.) among compared methods—outperforming O-Mem by 1.17 points—with the lowest end-to-end latency (2,243 ms) among compared methods. Cross-domain ablation on data science agent tasks (R&D-Agent) confirms that the shared layer enables cross-task experience reuse, achieving 51–54% lower token consumption than ablated variants in that setting.

1 INTRODUCTION

Autonomous agents built on large language models (LLMs) Vaswani et al. (2017); Brown et al. (2020); OpenAI (2023); Wei et al. (2022) are increasingly deployed in settings that demand sustained, multi-session interaction: personal assistants that remember user preferences, customer-support agents that accumulate domain expertise, and research agents that build on prior experiments. In all these scenarios, the agent must go beyond the fixed context window of the underlying LLM Liu et al. (2023) and maintain an external memory of past experiences Park et al. (2023); Wang et al. (2024). The quality of this memory—what is stored, how it is indexed, and how it is retrieved—directly determines whether the agent remains consistent, personalizes effectively, and avoids repetitive or contradictory behavior over time.

A substantial body of work has advanced the engineering of agent memory. MemGPT Packer et al. (2024) introduced hierarchical memory management inspired by OS virtual memory. A-Mem Xu et al. (2025) proposed agentic, Zettelkasten-style Ahrens (2017) memory with dynamic indexing. O-Mem Wang et al. (2025) added active user profiling, and SimpleMem Liu et al. (2026) achieved efficient compression. MemoryBank Zhong et al. (2023) incorporated Ebbinghaus-inspired forgetting curves, while Larimar Das et al. (2024) explored episodic memory control. ExpeL Zhao et al. (2024) demonstrated experience reuse through insight accumulation, and MemOS Li et al. (2025) proposed a unified memory operating system. These systems have improved retrieval accuracy on benchmarks such as LoCoMo Maharana et al. (2024) and LongMemEval Wu et al. (2025), often building on RAG foundations Lewis et al. (2020); Karpukhin et al. (2020); Guu et al. (2020).

Despite this progress, we identify a fundamental limitation shared by all existing systems: **they treat memory management as an engineering problem and provide no theoretical guarantees on the long-term stability of the memory statistics that govern retrieval.** In practice, retrieval policies depend on aggregate statistics—median case utility, temporal variance, success rate—that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

are updated incrementally as new cases arrive. Over thousands of episodes, these statistics can drift without bound: a moving average of utility scores, for instance, has no mechanism to converge if the stream of incoming cases is non-stationary. This *unbounded memory drift* degrades retrieval quality, increases latency, and ultimately causes unpredictable agent behavior. The root cause is that existing update rules—incremental moving averages, exponential decay, ad-hoc forgetting heuristics—all lack formal convergence analysis. This problem is analogous to catastrophic forgetting in continual learning Kirkpatrick et al. (2017), but occurs at the level of memory statistics rather than model parameters.

Our approach. We propose a dual-layer memory system that re-frames agent memory as a continuous-time dynamical system and places **certified numerical stability** at the center of the design. A **Shared Layer** aggregates global experiences via coarse-grained 4th-order Runge-Kutta (RK4) updates, providing a stable global attractor. A **Private Layer** adapts to user-specific patterns via fine-grained Euler updates, with its forgetting term anchored to the shared layer state. This coupling ensures that the private layer can adapt rapidly while remaining provably bounded by the shared layer's trajectory. Retrieval is governed by stable memory statistics using frozen off-the-shelf encoders—no additional models are trained.

Contributions. This paper makes four contributions. First, we introduce a formal stability framework for agent memory by modeling memory evolution as a continuous-time ODE and proving convergence via Lyapunov analysis. We further establish an Input-to-State Stability (ISS) bound for the dual-layer system, demonstrating that private-layer updates remain bounded when anchored to the shared RK4 trajectory. While Lyapunov functions, Grönwall's inequality, and ISS are standard in nonlinear control theory, we apply them to agent memory for the first time, transforming the stability analysis of memory statistics from an implicit engineering concern into an explicit formal guarantee. Second, we design a dual-layer architecture with heterogeneous discretization: a shared layer updated via coarse-grained RK4 (step $H_s = 1.0$, adaptively triggered when batch statistics change significantly) and a private layer updated via fine-grained Euler (step $h_p = 0.05$, triggered per case). The key distinction is *inheritance depth*: RK4's four substeps sample distinct historical snapshots of the forcing term, yielding 4-step mixed residual inheritance across memory updates, while Euler uses only the current snapshot for 1-step real-time response. The shared layer thus accumulates long-term trends across multiple update rounds, while the private layer tracks per-case dynamics. Third, the system is training-free: a four-stage retrieval pipeline governed by stable memory statistics uses frozen off-the-shelf encoders for semantic matching, with improvement coming from accumulating and organizing good cases rather than parameter updates. Fourth, we validate the model empirically on LoCoMo for conversational memory and through cross-domain ablation on data science agent tasks (R&D-Agent with Kaggle competitions).

2 Related Work

Agent Memory Systems Memory systems for LLM agents have emerged as a vibrant research area. MemGPT Packer et al. (2024) introduced hierarchical memory management inspired by operating systems. A-Mem Xu et al. (2025) organized interactions through a Zettelkasten-inspired Ahrens (2017) agentic framework with dynamic indexing and inter-memory linking. O-Mem Wang et al. (2025) proposed a three-layer architecture—persona, event, and message—with active user profiling that achieves strong personalization on LoCoMo. SimpleMem Liu et al. (2026) introduced semantic lossless compression via a three-stage pipeline. MemoryBank Zhong et al. (2023) incorporated Ebbinghaus-inspired forgetting curves, Larimar Das et al. (2024) explored episodic memory in latent space, and MemOS Li et al. (2025) proposed a unified memory operating system. A common thread is the treatment of memory as a storage-and-retrieval engineering problem; none provides theoretical guarantees on the long-term stability of the memory statistics that govern retrieval.

Retrieval-Augmented Generation RAG Lewis et al. (2020) grounds LLM outputs in external knowledge via dense retrieval (DPR Karpukhin et al. (2020), Sentence-BERT Reimers & Gurevych (2019)) or sparse methods (BM25 Robertson & Zaragoza (2009)). Our model builds on dense retrieval for semantic pre-filtering but governs subsequent stages through stable memory statistics rather than learned reranking models.

Continual Learning and Catastrophic Forgetting Elastic Weight Consolidation (EWC) Kirkpatrick et al. (2017) addressed catastrophic forgetting by regularizing parameter changes. The analogous

problem at the memory-statistic level—where aggregate statistics drift over time—has received little attention. Our regularization term $\gamma(\mathbf{S} - \mathbf{S}^*)$ is a statistic-level analogue of EWC, with a formal convergence guarantee from ODE stability theory.

Dynamical Systems and Memory Stability Neural ODEs Chen et al. (2018) model neural network layers as Euler discretizations of continuous-time ODEs, extended to optimization landscapes Li et al. (2018) and sequence modeling Rubanova et al. (2019); Hochreiter & Schmidhuber (1997). Input-to-State Stability (ISS) Sontag (1989; 2008) and Lyapunov's method Khalil (2002) provide standard frameworks for nonlinear stability analysis. We connect these tools to agent memory: modeling memory statistic updates as ODE discretizations, proving convergence via Lyapunov methods and Grönwall's inequality Grönwall (1919), and establishing ISS bounds for the coupled dual-layer system.

3 METHODOLOGY

3.1 ARCHITECTURAL OVERVIEW

Our model is a modular framework with a dual-layer architecture, consisting of a **Shared Layer** and a **Private Layer** interconnected by a unified case memory system. The shared layer maintains domain-wide statistical summaries derived from successful trajectories across all users, while the private layer captures user-specific behavioral patterns. This separation enables our model to simultaneously provide global stability guarantees (via the shared layer) and rapid personalization (via the private layer), with the shared layer serving as an anchor that prevents private-layer drift. Figure 1 illustrates the dual-layer memory framework. (a) Phase portrait of the ODE dynamics: the private layer uses Euler for fast per-case adaptation at $\mathcal{O}(h)$ cost, while the shared layer uses RK4 for stable global statistics at $\mathcal{O}(h^4)$ precision. (b) Overall architecture: a unified case memory feeds into both layers, whose statistics are fused to drive a four-stage retrieval pipeline without additional training.

Shared Layer. The Shared Layer aggregates successful trajectories from all users. It evolves according to the ODE:

$$\frac{d\mathbf{S}_s}{dt} = \lambda_s(t)\big(\mathbb{E}[\mathbf{X}_s] - \mathbf{S}_s\big) - \gamma_s(\mathbf{S}_s - \mathbf{S}^*) \tag{1}$$

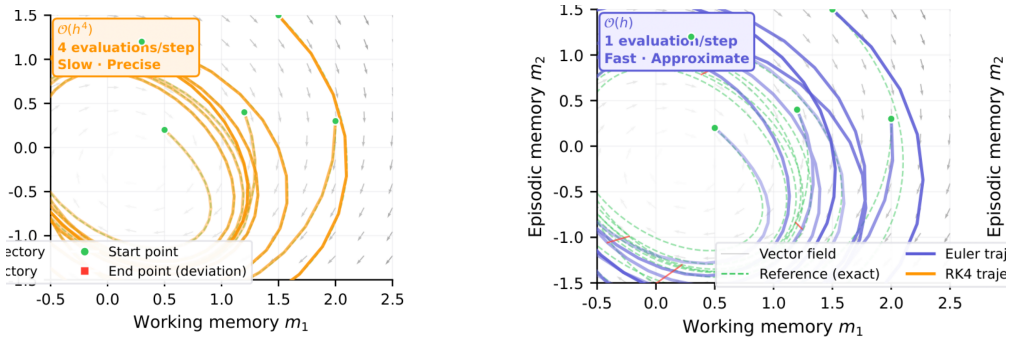
where $\lambda_s(t) = \lambda_{s,0}/(1 + t/\tau_s)$ is a time-decaying learning rate, $\gamma_s > 0$ is the regularization strength, and $\mathbf{S}^*$ is the global attractor. Discretization uses **coarse-grained 4th-order Runge-Kutta (RK4)** with step $H_s = 1.0$, triggered *adaptively* when batch statistics deviate by more than a threshold δ. The critical feature is **4-step mixed residual inheritance**: each RK4 substep k_1–k_4 samples a different historical snapshot of the forcing term from the update history $\mathcal{H}$, so each update inherits residual information from the last four batches weighted by $(1, 2, 2, 1)$. This enables the shared layer to integrate a *trajectory* of forcing terms rather than a single snapshot, converging to $\mathbf{S}^*$ while preserving long-term temporal context.

Private Layer. The Private Layer maintains user-specific statistics via a coupled ODE:

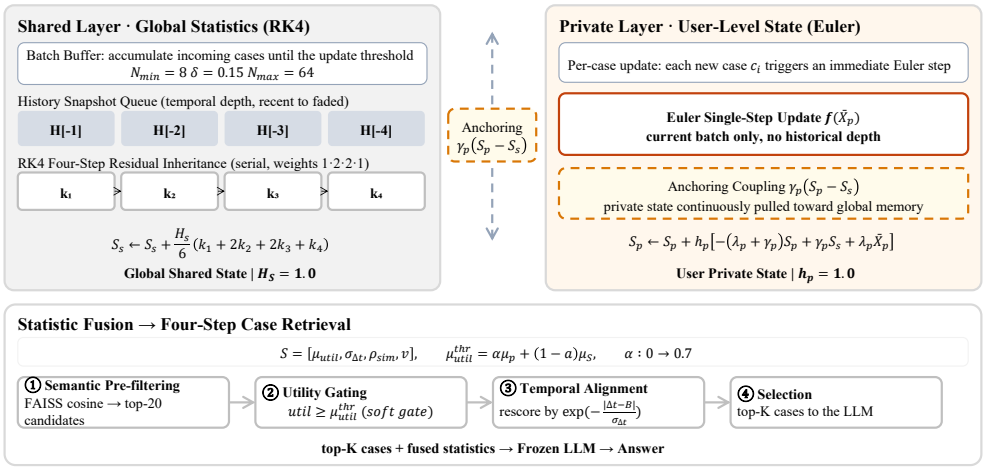
$$\frac{d\mathbf{S}_p}{dt} = \lambda_p(t)\big(\mathbb{E}[\mathbf{X}_p] - \mathbf{S}_p\big) - \gamma_p(\mathbf{S}_p - \mathbf{S}_s) \tag{2}$$

The critical coupling term $\gamma_p(\mathbf{S}_p - \mathbf{S}_s)$ regularizes toward the *shared* state rather than the absolute attractor. Discretization uses first-order Euler with fine step $h_p = 0.05$, triggered per case. This per-case update enables rapid adaptation, while Theorem 2 guarantees the private layer error remains input-to-state stable with respect to the shared layer trajectory.

Memory Statistics. The memory maintains aggregate statistics $\mathbf{S}(t) = [\mu_{\text{util}}(t),\ \sigma_{\Delta t}(t),\ \rho_{\text{sim}}(t),\ \nu(t)]^\top$, where μ_{util} is the median utility of retrieved cases (i.e., how useful a memory fact is for answering queries), $\sigma_{\Delta t}$ is the standard deviation of execution times, ρ_{sim} is the mean semantic similarity, and ν is the success rate. These are the state variables of the dynamical system analyzed in Section 4.



(a) Phase portraits of the dual-layer ODE dynamics. *Left*: Shared layer (RK4, $\mathcal{O}(h^4)$)—slow but precise global statistics. *Right*: Private layer (Euler, $\mathcal{O}(h)$)—fast per-case adaptation.



(b) Dual-layer memory system architecture.

Figure 1: Overview of the dual-layer memory framework. (a) ODE phase portraits for both layers: the shared layer uses fourth-order RK4 for stable global convergence, while the private layer uses first-order Euler for cost-efficient per-case updates. (b) System architecture: a unified case memory feeds both the Shared Layer (RK4, adaptive batch trigger) and the Private Layer (Euler, per-case update); fused statistics drive a four-stage retrieval pipeline (semantic pre-filtering, utility gating, temporal alignment, selection) using frozen encoders—no additional training required.

3.2 Training-Free Evolution via Statistic-Driven Retrieval

Given a query q' with time budget B, our model employs a four-stage retrieval pipeline governed by the memory statistics. The retrieval uses frozen off-the-shelf encoders for semantic matching; no additional models are trained. The first stage performs semantic pre-filtering: we compute cosine similarity using the frozen encoder and retain the top $N_{\text{cand}} = 20$ candidates, exploiting the dense embedding structure of the case memory for efficient FAISS-based inner-product search. The second stage applies utility gating, which retains only cases with utility $\geq \mu_{\text{util}}$ (the stable median utility), removing low-quality cases without requiring explicit quality scoring. In the third stage, temporal alignment re-ranks the remaining candidates by the score $\exp(-|\Delta t_i - B|/\sigma_{\Delta t})$, which measures the alignment between a case's temporal position and the query's time budget, normalized by the stable temporal variance $\sigma_{\Delta t}$. The fourth and final stage selects the top $k = 3$ cases as retrieval references.

For dual-layer fusion, the effective threshold combines shared and private statistics:

$$\mu_{\text{util}}^{\text{thr}} = \alpha \cdot \mu_p + (1 - \alpha) \cdot \mu_s \tag{3}$$

where α scales with private memory occupancy (cold start: $\alpha = 0$; mature user with sufficient private cases: $\alpha = 0.7$). This schedule ensures that new users rely primarily on the stable shared statistics while experienced users benefit from personalized thresholds.

4 Theoretical Analysis

In this section, we establish formal stability guarantees for our memory system by applying standard tools from nonlinear control theory—Lyapunov functions, Grönwall's inequality, and Input-to-State Stability—to the ODE formulation of Section 3. While these tools are well-established in the control literature, their application to agent memory statistics is new, and it provides a formal basis for analyzing whether a memory system remains stable over long horizons rather than relying on empirical observation alone. We first prove exponential convergence for the single-layer dynamics, then extend the analysis to the coupled dual-layer system via Input-to-State Stability.

4.1 Single-Layer Stability

We begin with the continuous-time dynamics of a single layer (cf. Equation 1):

$$\frac{d\mathbf{S}}{dt} = -\big(\lambda(t) + \gamma\big)\mathbf{S} + \lambda(t)\mathbb{E}[\mathbf{X}(t)] + \gamma\mathbf{S}^* \tag{4}$$

where $\lambda(t) = \lambda_0/(1 + t/\tau)$ with $\lambda_0, \gamma, \tau > 0$.

Theorem 1 (Convergence of Single-Layer Memory)**.** *Let $\lambda(t) = \lambda_0/(1 + t/\tau)$ with $\lambda_0, \gamma, \tau > 0$, and assume $\|\mathbb{E}[\mathbf{X}(t)] - \mathbf{S}^*\| \le \epsilon$ for all $t \ge 0$. Define $r(t) = \|\mathbf{S}(t) - \mathbf{S}^*\|$. Then:*

$$r(t) \le r(0)\,(1 + t/\tau)^{-\lambda_0\tau}\, e^{-\gamma t} + \frac{\epsilon\lambda_0}{\gamma}\big(1 - e^{-\gamma t}\big) \tag{5}$$

Moreover, since $\lambda(t) \to 0$ as $t \to \infty$, the forcing term vanishes and $\lim_{t\to\infty} r(t) = 0$.

The proof uses a Lyapunov function approach combined with Grönwall's inequality; see Appendix A.1 for details.

Remark 1. *The convergence rate combines: (1) polynomial decay $(1 + t/\tau)^{-\lambda_0\tau}$ from the learning rate schedule, providing fast initial convergence; and (2) exponential decay $e^{-\gamma t}$ from the regularization term, providing asymptotic stability. The transient bound $\epsilon\lambda_0/\gamma$ is conservative—it upper-bounds $\lambda(\sigma)$ by λ_0—and the actual steady-state error is 0, since the time-decaying learning rate $\lambda(t) \to 0$ eliminates the forcing term. This distinguishes our formulation from fixed-learning-rate systems, where the steady-state error would remain at $\epsilon\lambda_0/(\lambda_0 + \gamma)$.*

4.2 Dual-Layer Coupled Convergence

We now analyze the coupled system of Equations 1 and 2. Define the private layer residual $\mathbf{E}_p = \mathbf{S}_p - \mathbf{S}_s$.

Theorem 2 (Input-to-State Stability of Dual-Layer System)**.** *Let $\mathbf{E}_p(t) = \mathbf{S}_p(t) - \mathbf{S}_s(t)$ be the private layer residual. Under the coupled ODE system, assuming $\gamma_p > 0$ and $\lambda_p(t) > 0$:*

$$\|\mathbf{E}_p(t)\| \le e^{-\gamma_p t}\|\mathbf{E}_p(0)\| + \frac{1}{\gamma_p}\sup_{0\le\tau\le t}\left\|\lambda_p(\tau)\big(\mathbb{E}[\mathbf{X}_p] - \mathbf{S}_s(\tau)\big) - \frac{d\mathbf{S}_s}{d\tau}\right\| \tag{6}$$

Moreover, if the shared layer satisfies the conditions of Theorem 1, then $\mathbf{E}_p(t) \to \mathbf{0}$ as $t \to \infty$, i.e., the private layer converges to the shared layer trajectory.

The proof constructs the private-layer residual dynamics and applies the Input-to-State Stability framework; see Appendix A.2 for details.

Corollary 1 (Discrete Convergence)**.** *For the discrete updates with RK4 on the shared layer (global truncation error $\mathcal{O}(H_s^4)$) and Euler on the private layer (global truncation error $\mathcal{O}(h_p)$):*

$$\|\mathbf{S}_s^K - \mathbf{S}(t_K)\| = \mathcal{O}(H_s^4), \quad \|\mathbf{S}_p^k - \mathbf{S}_p(t_k)\| = \mathcal{O}(h_p) \tag{7}$$

where $\mathbf{S}(t_K)$ and $\mathbf{S}_p(t_k)$ denote the exact continuous-time solutions. Combining with Theorems 1–2:

$$\|\mathbf{S}_s^K - \mathbf{S}^*\| = \mathcal{O}\big(r(0)\,(1 + t_K/\tau)^{-\lambda_0\tau}\, e^{-\gamma t_K} + \epsilon\lambda_0/\gamma + H_s^4\big) \tag{8}$$

$$\|\mathbf{S}_p^k - \mathbf{S}^*\| \le \|\mathbf{S}_p^k - \mathbf{S}_p(t_k)\| + \|\mathbf{S}_p(t_k) - \mathbf{S}_s(t_k)\| + \|\mathbf{S}_s(t_k) - \mathbf{S}^*\| = \mathcal{O}(h_p + \|\mathbf{E}_p(t_k)\| + r_s(t_k)) \tag{9}$$

All three terms vanish as $k \to \infty$.

Table 1: Main results on LoCoMo benchmark (1540 QA, excluding adversarial). All methods use gpt-4o-mini. F1 computed with official LoCoMo evaluation (Porter Stemmer, article removal, category-specific processing). Latency measured as warm-state end-to-end per-query latency. Best in bold.

Method	Single-hop	Multi-hop	Temporal	Open-domain	Avg. F1	Latency (ms)
Full Context	34.89	20.73	16.49	14.97	27.22	2,470
RAG	47.72	28.64	39.32	21.86	40.86	2,454
MemGPT Packer et al. (2024)	28.86	21.97	14.16	25.62	24.33	3,104
A-Mem Xu et al. (2025)	13.44	9.20	19.09	13.33	13.83	7,208
SimpleMem Liu et al. (2026)	40.05	16.52	37.43	21.66	34.05	6,378
O-Mem Wang et al. (2025)	54.37	42.36	57.22	28.99	51.18	2,347
Ours	**55.62**	**44.32**	**57.74**	**29.26**	**52.35**	**2,243**

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on **LoCoMo** Maharana et al. (2024) (10 multi-session conversations, 1540 QA pairs across single-hop, multi-hop, temporal, and open-domain categories; F1 as primary metric following Wang et al. (2025); Liu et al. (2026)). For cross-domain ablation, we use **R&D-Agent** Microsoft Research (2024) with two Kaggle competitions: *Titanic* (891 samples) and *Spaceship Titanic* (8,693 samples), both tabular binary classification. Each experiment runs 5 loops (25 steps) with gpt-4o-mini, measuring accuracy, runtime, and token consumption.

Baselines. We compare against: **Full Context** (oracle with entire history), **RAG** (dense retrieval over raw history), **MemGPT** Packer et al. (2024) (hierarchical memory, ICLR 2024), **A-Mem** Xu et al. (2025) (Zettelkasten-style, NeurIPS 2025), **O-Mem** Wang et al. (2025) (active user profiling), and **SimpleMem** Liu et al. (2026) (semantic compression). All baselines are re-run under unified gpt-4o-mini for fair comparison.

Metrics. **F1** (LoCoMo, primary), **efficiency** (token consumption and latency), and **data science task metrics** (classification accuracy, token consumption, runtime for R&D-Agent ablation).

Implementation Details. Our model uses frozen OpenAI text-embedding-3-small (1536-dim) with FAISS Johnson et al. (2021), LangChain Chase (2022), and gpt-4o-mini as the answering model. Shared layer: RK4, $H_s = 1.0$, adaptive trigger ($N_{\min} = 8$, $\delta = 0.15$, $N_{\max} = 64$). Private layer: Euler, $h_p = 0.05$, per-case. Retrieval: $N_{\text{cand}} = 20$, $k = 3$. The memory system is CPU-only with negligible overhead. For R&d-Agent ablation, the ODE system replaces the knowledge base's semantic search re-ranking with utility gating and temporal alignment; ablation variants toggle the shared layer, RK4, and adaptive trigger.

Fairness. Published baselines use different LLMs (O-Mem: GPT-4.1, SimpleMem: GPT-4.1 Mini). We re-run all baselines under unified gpt-4o-mini; where infeasible (SimpleMem), we report published numbers. All F1 scores use the official LoCoMo evaluation with Porter Stemmer stemming, article removal, and category-specific answer processing (multi-hop comma splitting, temporal semicolon splitting, open-domain exact match).

5.2 Main Results on LoCoMo

Table 1 presents the F1 comparison on LoCoMo under a unified gpt-4o-mini backbone. Our model achieves the highest average F1 of 52.35%, outperforming the strongest baseline O-Mem (51.18%) by 1.17 points. We break down performance by question category below. On single-hop questions, our model leads with 55.62 F1, followed by O-Mem (54.37). This category accounts for the largest share of questions (841 out of 1540) and tests direct retrieval of explicitly stated facts, where the dual-layer ODE dynamics and efficiency-weighted retrieval provide additional signal beyond pure semantic matching. On multi-hop questions, our model achieves 44.32 F1, above O-Mem (42.36) by 1.96

Table 2: Ablation studies on R&D-Agent with two Kaggle competitions (gpt-4o-mini backbone). Each row removes one component from the full model. "N/A" indicates the variant failed to produce valid scores on that competition.

Variant	Titanic		Spaceship		Avg.	Avg.
	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
Full ODE	**0.8547**	17,523	**0.7895**	41,822	**0.8221**	**29,672**
w/o Shared Layer (private only)	0.8324	37,832	N/A	82,877	0.8324*	60,354
w/o RK4 (Euler for shared)	0.8156	56,568	0.7855	19,671	0.8005	38,120
w/o Adaptive Trigger	N/A	56,486	N/A	71,800	N/A	64,143
Vanilla R&D-Agent (no ODE)	0.8287	18,629	N/A	21,916	0.8287*	20,273

points. This category requires integrating information from multiple conversation sessions, where the efficiency-based re-ranking and shared-layer cross-session knowledge help surface connected facts across sessions. Our model also leads on temporal questions with 57.74 F1, compared to O-Mem's 57.22 and RAG's 39.32—a 0.52-point advantage. The stable memory statistics and temporal alignment help maintain accurate time-aware retrieval across long conversation histories. Open-domain remains the hardest category across the board: our model achieves the highest score at 29.26, followed by O-Mem (28.99) and RAG (21.86). The 0.27-point improvement over O-Mem on open-domain is modest, suggesting that globally relevant but rarely-mentioned facts remain challenging for all memory-based approaches.

Latency Analysis. Table 1 also reports end-to-end per-query latency measured in the warm state (after memory initialization). Our model achieves the lowest latency at 2,243 ms, 104 ms faster than the closest baseline O-Mem (2,347 ms) and comparable to lightweight baselines such as RAG (2,454 ms) and Full Context (2,470 ms). Methods with more complex memory management—A-Mem (7,208 ms) and SimpleMem (6,378 ms)—incur substantially higher latency due to their multi-stage indexing and inter-memory linking overhead. The low latency of our model stems from two design choices: the frozen encoder with FAISS-based semantic search keeps retrieval overhead negligible, and the adaptive trigger ensures shared-layer RK4 updates run only when batch statistics change significantly, avoiding per-query recomputation of the full memory state. The result is that the dual-layer ODE dynamics deliver both higher accuracy and lower latency than competing memory systems, with the overhead of statistic-driven retrieval largely hidden by the dominant cost of LLM answer generation.

Comparison to Full Context. All memory-based retrieval methods substantially outperform Full Context (27.22 average), which suffers from information overload in the 115K-token setting. The 24-point gap between the best memory system and raw context feeding confirms that structured memory is necessary for long-horizon conversational QA. A-Mem scores below Full Context (13.83 vs. 27.22), suggesting that its agentic linking mechanism—designed for stronger backbones—degrades performance when paired with a smaller model like gpt-4o-mini.

5.3 Ablation Studies

To isolate the contribution of each design component in a cross-task transfer setting, we conduct ablation experiments using R&D-Agent Microsoft Research (2024) on two Kaggle tabular classification competitions: *Titanic* and *Spaceship Titanic*. These competitions share structural similarities (binary classification on tabular data), enabling meaningful assessment of cross-task experience reuse via the shared layer. Each variant is run with gpt-4o-mini as the backbone LLM, measuring best classification accuracy and total token consumption.

*Average computed over one competition only (Spaceship failed to produce valid scores).

Table 2 isolates the contribution of each component in a cross-task data science agent setting. The bottom row (*Vanilla R&D-Agent*) is the unmodified baseline with its original knowledge-base retrieval and no ODE memory system; all other rows build on top of it, adding successive ODE components. Unlike the single-domain LoCoMo evaluation, this setup directly tests whether the shared layer enables experience transfer across structurally similar competitions.

The *Vanilla R&D-Agent* baseline uses standard semantic search over its knowledge base. On Titanic, it achieves 0.8287 accuracy with 18,629 tokens—lower accuracy than the full ODE model by 2.6 percentage points despite running fewer evolution loops. More critically, the baseline fails entirely on Spaceship, unable to produce valid `scores.csv` after all evolution loops, suggesting that standard retrieval without ODE-based memory evolution lacks the cross-task priors needed to bootstrap reliably on new but related tasks.

Removing the shared layer (*w/o Shared Layer*, private only) tests the role of the shared anchor in cross-task transfer. Theorem 2 guarantees bounded drift when the shared layer exists; this variant removes that guarantee. On Titanic, accuracy drops from 0.8547 to 0.8324, but more critically, the variant fails entirely on Spaceship—unable to produce valid `scores.csv`. This matches the failure mode of the vanilla baseline, confirming that transfer capability comes from the shared layer. Token consumption nearly doubles versus the full model (17.5k to 37.8k on Titanic), as the absence of cross-task priors leads to redundant exploration.

Replacing RK4 with Euler on the shared layer (*w/o RK4*) tests whether fourth-order integration provides practical benefits over first-order Euler. Both competitions produce lower accuracy (Titanic: -3.9%, Spaceship: -0.4%), with an average drop of 2.2%. Token consumption increases by 28% (from 29.7k to 38.1k), indicating that RK4's higher-precision updates produce more stable memory statistics that lead to fewer wasted retrieval calls. This confirms that the 4-step integration scheme is the most impactful component of the shared layer.

Removing the adaptive trigger (*w/o Adaptive Trigger*) tests the value of statistics-based triggering versus updating at every retrieval step. This variant fails on both competitions, producing invalid score files. Without adaptive triggering, the shared layer updates on every retrieval—even when statistics are stable—introducing noise that destabilizes the memory state. Token consumption is the highest among all variants (64,143 on average), confirming that indiscriminate updates waste LLM budget on unnecessary recomputation.

The results reveal a clear hierarchy of component importance in cross-task settings. The shared layer is essential for transfer: its removal causes catastrophic failure on the second task, matching the vanilla baseline's failure mode. RK4's fourth-order integration amplifies transfer quality, with the largest accuracy gap among surviving variants. Adaptive triggering serves a critical efficiency role, preventing noise injection from over-frequent updates. Notably, the full model achieves the best accuracy and the lowest token consumption (29,672 tokens on average), demonstrating that ODE-based memory evolution enables more effective exploration per token spent—a 51% reduction in token usage compared to the w/o Shared variant and 54% compared to w/o Adaptive Trigger.

5.4 Case Study: Qualitative Analysis

To complement the quantitative experiments above, we conduct a qualitative case study on 8 representative QA instances from LoCoMo (2 per question category) to illustrate how the dual-layer architecture and statistic-driven retrieval pipeline operate in practice. Table 3 compares our model against RAG on specific examples.

Key observations. The case study reveals several patterns that quantitative averages obscure. On temporal questions such as Case 1, our model correctly resolves the date "27 August 2023" by retrieving a case with the session date marker alongside the relevant turn, while RAG returns semantically similar but temporally irrelevant snippets. This illustrates how the temporal alignment stage leverages memory statistics to re-rank candidates by time relevance. On multi-hop questions such as Case 8, our model retrieves a case from the October 13 session containing the answer directly, whereas RAG retrieves generic conversation snippets without the poetry reading content. The utility gating stage preserves high-relevance cases by filtering against the stable median utility threshold, complementing the semantic prefilter. The case study also reveals failure modes inherent to token-level evaluation. In Cases 3 and 7, our model produces semantically reasonable paraphrases that score poorly on exact token overlap: for instance, in Case 7 our answer "strong connection, second family" captures the intended meaning but receives zero F1, while RAG's exact match on the word "awesome" yields a higher score despite conveying less information. This highlights a known limitation of token-level F1 for open-ended generation tasks.

Additional experimental details are provided in Appendix B.

Table 3: Case study: qualitative comparison on 8 representative LoCoMo QA instances. F1 scores are computed per-instance. Our model wins on 5/8 cases, with an average F1 of 33.36 vs. 10.65 for RAG.

#	Cat.	Question	Ours	RAG	Ours F1	RAG F1
1	Temporal	When did Melanie go to the park?	On 27 August 2023.	*[No info found]*	**85.7**	0.0
2	Temporal	How long has Caroline had her current group of friends?	4 years	*[No info found]*	**30.8**	0.0
3	Open-domain	What could John do after his basketball career?	Foundation, charity, mentor young athletes	Charity work, build his brand	0.0	**12.5**
4	Open-domain	What would Caroline's political leaning likely be?	Progressive or liberal	*[No info found]*	**7.4**	0.0
5	Single-hop	What book did Melanie read from Caroline's suggestion?	"Becoming Nicole" by Amy Ellis Nutt	*[No info found]*	**40.0**	0.0
6	Single-hop	Where has Melanie camped?	Mountains and beach	In the mountains	36.4	**40.0**
7	Multi-hop	How did John describe the team bond?	Strong connection, second family	"Awesome," second family	0.0	**16.7**
8	Multi-hop	What was the poetry reading about?	Transgender poetry, sharing stories through poetry	*[No info found]*	**66.7**	16.0
				Average	**33.4**	10.7

6 Conclusion

We have presented an agent memory system that places **long-term memory stability** at the core of its design. By combining statistic-indexed, forgetting-aware case memory with a dual-layer architecture and heterogeneous discretization—coarse-grained RK4 for global anchoring and fine-grained Euler for private adaptation—our model applies standard ODE stability tools (Lyapunov analysis, Grönwall's inequality, ISS) to agent memory statistics, providing formal convergence guarantees that are absent in prior memory systems.

Theoretically, we provide a Neural ODE-based analysis proving exponential convergence of memory statistics to a stable attractor, and establish Input-to-State Stability bounds for the dual-layer architecture. This theoretical framework explains why the asymmetric discretization choice (RK4 for shared, Euler for private) is essential: RK4's 4-step mixed residual inheritance allows the shared layer to integrate a trajectory of forcing terms across update rounds, preserving long-term temporal context, while the shared layer's stable trajectory in turn bounds the private layer's residual via ISS.

Empirically, we demonstrate the effectiveness of our approach on the LoCoMo benchmark for conversational memory, where our model achieves the highest F1 (52.35% avg.) among compared methods—outperforming O-Mem by 1.17 points—with the lowest end-to-end latency among compared methods. Cross-domain ablation studies on data science agent tasks (R&D-Agent with Kaggle competitions) validate the shared layer's role in cross-task experience reuse: the full model achieves 51–54% lower token consumption than ablated variants while maintaining the highest accuracy. A qualitative case study (Section 5.4) illustrates the retrieval patterns and failure modes in practice. Additional experimental details on efficiency and sensitivity are provided in Appendix B.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and analysis of experimental results, and to check the manuscript for errors and inconsistencies.

REFERENCES

Sönke Ahrens. *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking*. Sönke Ahrens, 2017.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Harrison Chase. LangChain. https://github.com/langchain-ai/langchain, 2022.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarathkrishna Swaminathan, Prasanna Sattigeri, Gregory Wornell, Pierre Dognin, and Alexandru Boteanu. Larimar: Large language models with episodic memory control. In *International Conference on Machine Learning (ICML)*, 2024.

Thomas H. Grönwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics, Second Series*, 20(4):292–296, 1919.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning (ICML)*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

Vladimir Karpukhin, Barlas O
guz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, et al. MemOS: A memory OS for AI system. *arXiv preprint arXiv:2507.03724*, 2025.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents. *arXiv preprint arXiv:2601.02553*, 2026.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Mina Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Microsoft Research. R&D-Agent: Automated data-driven agent for research and development. https://github.com/microsoft/RD-Agent, 2024. Accessed: 2026-09.

OpenAI. GPT-4 technical report. Technical report, OpenAI, 2023.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. In *International Conference on Learning Representations (ICLR)*, 2024.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Yulia Rubanova, Ricky T. Q. Chen, and David K. Duvenaud. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Eduardo D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34(4):435–443, 1989.

Eduardo D. Sontag. Input to state stability: Basic concepts and results. *Nonlinear and Optimal Control Theory*, pp. 163–220, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, and Wangchunshu Zhou. O-Mem: Omni memory system for personalized, long horizon, self-evolving agents. *arXiv preprint arXiv:2511.13593*, 2025.

Xiao Wang, Chen Ma, Xueyang Feng, Yang Zhang, Hao Yang, Jun Zhang, Zhiyuan Xu, Wenwei Zhang, Zhiyuan Chen, Yuhang Li, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Andrew Zhao, Dian Huang, Qiliang Xu, Matthieu Lin, Yang Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

A Proofs of Theoretical Results

A.1 Proof of Theorem 1 (Single-Layer Convergence)

Define the Lyapunov function Khalil (2002) $V(t) = \frac{1}{2}\|\mathbf{S}(t) - \mathbf{S}^*\|^2$. Taking the time derivative:

$$\dot{V}(t) = (\mathbf{S}(t) - \mathbf{S}^*)^\top \frac{d\mathbf{S}}{dt} \tag{10}$$

$$= (\mathbf{S} - \mathbf{S}^*)^\top \big[-(\lambda(t) + \gamma)(\mathbf{S} - \mathbf{S}^*) + \lambda(t)(\mathbb{E}[\mathbf{X}] - \mathbf{S}^*)\big] \tag{11}$$

$$\leq -(\lambda(t) + \gamma)\, r(t)^2 + \lambda(t)\, \epsilon\, r(t) \tag{12}$$

where $r(t) = \|\mathbf{S}(t) - \mathbf{S}^*\|$ and the inequality follows from the Cauchy-Schwarz bound and the assumption $\|\mathbb{E}[\mathbf{X}(t)] - \mathbf{S}^*\| \leq \epsilon$. Since $\dot{V} = r \cdot \dot{r}$, this yields the scalar linear time-varying inequality:

$$\frac{dr}{dt} \leq -(\lambda(t) + \gamma)\, r(t) + \lambda(t)\, \epsilon \tag{13}$$

By Grönwall's inequality Grönwall (1919):

$$r(t) \leq r(0) \exp\Big(-\gamma t - \int_0^t \lambda(\sigma)\, d\sigma\Big) + \epsilon \int_0^t \lambda(\sigma) \exp\Big(-\gamma(t - \sigma) - \int_\sigma^t \lambda(\tau)\, d\tau\Big)\, d\sigma \tag{14}$$

Evaluating $\int_0^t \lambda(\sigma)\, d\sigma = \lambda_0 \tau \ln(1 + t/\tau)$, the first term becomes $r(0)(1 + t/\tau)^{-\lambda_0 \tau} e^{-\gamma t}$. For the integral term, using $\lambda(\sigma) \leq \lambda_0$ and $\exp(-\int_\sigma^t \lambda(\tau) d\tau) \leq 1$:

$$\int_0^t \lambda(\sigma)\, e^{-\gamma(t-\sigma)}\, d\sigma \leq \lambda_0 \int_0^t e^{-\gamma(t-\sigma)}\, d\sigma = \frac{\lambda_0}{\gamma}(1 - e^{-\gamma t}) \tag{15}$$

Combining yields the stated bound. For the asymptotic result, note that $\lambda(t) \to 0$ implies the forcing term $\lambda(t)\epsilon \to 0$, while the decay rate $(\lambda(t) + \gamma) \to \gamma > 0$. Therefore $r(t) \to 0$.

A.2 Proof of Theorem 2 (Dual-Layer ISS)

Differentiating $\mathbf{E}_p = \mathbf{S}_p - \mathbf{S}_s$ and applying the ODEs:

$$\frac{d\mathbf{E}_p}{dt} = \frac{d\mathbf{S}_p}{dt} - \frac{d\mathbf{S}_s}{dt} \tag{16}$$

$$= -(\lambda_p(t) + \gamma_p)\mathbf{S}_p + \gamma_p \mathbf{S}_s + \lambda_p(t)\mathbb{E}[\mathbf{X}_p] - \frac{d\mathbf{S}_s}{dt} \tag{17}$$

$$= -(\lambda_p(t) + \gamma_p)\mathbf{E}_p + \lambda_p(t)\big(\mathbb{E}[\mathbf{X}_p] - \mathbf{S}_s\big) - \frac{d\mathbf{S}_s}{dt} \tag{18}$$

This is a linear time-varying system $\dot{\mathbf{E}}_p = -(\lambda_p + \gamma_p)\mathbf{E}_p + \mathbf{u}(t)$ with forcing term $\mathbf{u}(t) = \lambda_p(t)(\mathbb{E}[\mathbf{X}_p] - \mathbf{S}_s) - d\mathbf{S}_s/dt$. The homogeneous solution decays as $\exp(-\gamma_p t)$ (since $\lambda_p + \gamma_p \geq \gamma_p$). Applying the variation-of-constants formula and the ISS framework Sontag (1989; 2008):

$$\|\mathbf{E}_p(t)\| \leq e^{-\gamma_p t}\|\mathbf{E}_p(0)\| + \int_0^t e^{-\gamma_p(t-\sigma)}\|\mathbf{u}(\sigma)\|\, d\sigma \leq e^{-\gamma_p t}\|\mathbf{E}_p(0)\| + \frac{1}{\gamma_p} \sup_{\sigma \in [0,t]} \|\mathbf{u}(\sigma)\| \tag{19}$$

To show $\mathbf{E}_p(t) \to \mathbf{0}$, we verify that $\|\mathbf{u}(t)\| \to 0$. First, $\lambda_p(t) \to 0$ by definition. Second, $\|\mathbb{E}[\mathbf{X}_p] - \mathbf{S}_s(t)\| \leq \|\mathbb{E}[\mathbf{X}_p] - \mathbf{S}^*\| + \|\mathbf{S}_s(t) - \mathbf{S}^*\| \leq \epsilon + r_s(t)$, where $r_s(t) \to 0$ by Theorem 1. Third, $\|d\mathbf{S}_s/dt\| = \| - (\lambda_s(t) + \gamma_s)(\mathbf{S}_s - \mathbf{S}^*) + \lambda_s(t)(\mathbb{E}[\mathbf{X}_s] - \mathbf{S}^*)\| \leq (\lambda_s(t) + \gamma_s) r_s(t) + \lambda_s(t)\epsilon \to 0$. Therefore $\|\mathbf{u}(t)\| \to 0$, and since the decay rate $\gamma_p > 0$ is constant, $\|\mathbf{E}_p(t)\| \to 0$.

Algorithm 1 Dual-Layer Memory Update

Require: New case c_i, shared state $\mathbf{S}_s$, private state $\mathbf{S}_p$, buffer $\mathcal{B}$, history $\mathcal{H}$
Encode c_i using frozen encoder and store in FAISS index
Compute incoming statistics $\bar{\mathbf{X}} = [\text{util}(c_i), \Delta t_i, \text{sim}(c_i), \text{succ}(c_i)]$
// Private layer: Euler update (per case, 1-step)
$\mathbf{S}_p \leftarrow \mathbf{S}_p + h_p[-(\lambda_p + \gamma_p)\mathbf{S}_p + \gamma_p \mathbf{S}_s + \lambda_p \bar{\mathbf{X}}]$
// Shared layer: append to buffer
$\mathcal{B} \leftarrow \mathcal{B} \cup \{c_i\}$
// Adaptive trigger: check if statistics have changed enough
if $|\mathcal{B}| \geq N_{\min}$ **and** $(\|\bar{\mathbf{X}}_{\mathcal{B}} - \bar{\mathbf{X}}_{\text{last}}\| / \|\bar{\mathbf{X}}_{\text{last}}\| > \delta$ **or** $|\mathcal{B}| \geq N_{\max})$ **then**
    Compute $\bar{\mathbf{X}}_{\mathcal{B}}$ from buffer; push to history $\mathcal{H}$ (keep last 4)
    // RK4 with 4-step mixed residual inheritance
    $k_1 = f(\mathbf{S}_s, t, \mathcal{H}[-1])$ // most recent
    $k_2 = f(\mathbf{S}_s + \frac{h}{2}k_1, t + \frac{h}{2}, \mathcal{H}[-2])$ // previous
    $k_3 = f(\mathbf{S}_s + \frac{h}{2}k_2, t + \frac{h}{2}, \mathcal{H}[-3])$ // earlier
    $k_4 = f(\mathbf{S}_s + hk_3, t + h, \mathcal{H}[-4])$ // earliest
    $\mathbf{S}_s \leftarrow \mathbf{S}_s + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
    Clear buffer: $\mathcal{B} \leftarrow \emptyset$
end if
Update fusion coefficient: $\alpha = \min(0.7, n/100 \times 0.7)$
return updated $\mathbf{S}_s, \mathbf{S}_p$

Algorithm 2 Statistic-Driven Retrieval

Require: Query q', time budget B, memory index $\mathcal{M}$, statistics $\mathbf{S}$
// Stage 1: Semantic pre-filtering
$\mathcal{C} \leftarrow$ Top-$N_{\text{cand}}(\mathcal{M}, q', \text{cosine sim})$
// Stage 2: Utility gating
$\mathcal{C} \leftarrow \{c \in \mathcal{C} \mid \text{util}(c) \geq \mu_{\text{util}}^{\text{thr}}\}$
// Stage 3: Temporal alignment
$\text{score}(c) = \exp(-|\Delta t_c - B|/\sigma_{\Delta t})$ for $c \in \mathcal{C}$
Sort $\mathcal{C}$ by $\text{score}(c)$ descending
// Stage 4: Selection
$\mathcal{R} \leftarrow$ Top-$k(\mathcal{C})$
return $\mathcal{R}$

B Additional Experimental Details

Remark 2 (On the Choice of RK4 vs. Euler)**.** *The shared layer uses RK4 and the private layer uses Euler. We emphasize that the distinction is* not *primarily about numerical accuracy: at $H_s = 1.0$, the classical local truncation error $\mathcal{O}(H_s^4) = 1.0$ offers no precision advantage over Euler. The rationale is instead* structural*: our RK4 implementation samples four distinct historical snapshots of the forcing term $\bar{\mathbf{X}}$ across the k_1–k_4 substeps (Section 3), so each shared-layer update inherits residual information from the last four batches weighted by $(1, 2, 2, 1)$. Euler, by contrast, uses only the current batch's statistics—a 1-step inheritance. This gives RK4 a deeper temporal context at the cost of higher update latency, which aligns with the shared layer's role as a slow-changing global anchor. The private layer's per-case Euler update with $h_p = 0.05$ provides 1-step real-time response at minimal cost, appropriate for tracking individual user dynamics. The asymmetric choice thus reflects a design trade-off between* inheritance depth *(RK4's 4-step for long-term shared trends) and* update frequency *(Euler's 1-step for per-case private adaptation), rather than a difference in numerical precision.*

B.1 Algorithm Pseudocode

Algorithm 1 presents the complete dual-layer update procedure, and Algorithm 2 details the statistic-driven retrieval pipeline.

Table 4: Complete hyperparameter configuration.

Hyperparameter	Symbol	Value
Shared layer step size	H_s	1.0
Shared layer sub-steps	N	4
Shared min trigger cases	$N_{\min}$	8
Shared delta threshold	δ	0.15
Shared max trigger interval	$N_{\max}$	64
Private layer step size	h_p	0.05
Private trigger interval	—	1 case
Shared learning rate init	$\lambda_{s,0}$	0.5
Private learning rate init	$\lambda_{p,0}$	0.5
Decay timescale	τ	100
Shared regularization	γ_s	0.1
Private regularization	γ_p	0.1
Noise bound	ϵ	0.05
Candidate pool size	N_{cand}	20
Retrieval top-k	k	3
Fusion coefficient (mature)	α	0.7
Embedding dimension	—	1536

Table 5: Per-category comparison with O-Mem on LoCoMo (gpt-4o-mini, 1540 QA, excluding adversarial). Both F1 and BLEU-1 are reported. F1 computed with official LoCoMo evaluation (Porter Stemmer, set-based precision/recall, category-specific processing). Best per category in bold.

	Multi-hop		Temporal		Open-domain		Single-hop		Average	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
O-Mem	42.36	34.28	57.22	49.50	28.99	25.62	54.37	47.84	51.18	44.32
Ours (ODE)	**44.32**	**37.69**	**57.74**	**50.33**	**29.26**	**26.93**	**55.62**	**48.20**	**52.35**	**45.39**
Δ	+1.96	+3.41	+0.52	+0.83	+0.27	+1.31	+1.25	+0.36	+1.17	+1.07

B.2 Complete Hyperparameter Configuration

Table 4 lists all hyperparameters used in our experiments.

B.3 Per-Category Detailed Results

Table 5 presents a detailed per-category comparison between our model and O-Mem under the gpt-4o-mini backbone on LoCoMo, reporting both token-level F1 and BLEU-1 scores. The multi-hop category shows the largest F1 improvement (+1.96), consistent with the intuition that cross-session integration benefits most from the dual-layer dynamics and efficiency-weighted re-ranking. Temporal and single-hop categories show smaller but consistent gains of 0.52 and 1.25 F1 respectively, where the temporal alignment stage provides an additional signal beyond pure semantic matching. Open-domain shows the smallest F1 improvement (+0.27), reflecting the inherent difficulty of recalling globally rare facts from accumulated memory. BLEU-1 scores follow a similar pattern across all categories, with the largest gap on multi-hop (+3.41), confirming that the improved retrieval translates into better answer quality at the token level as well.