Deployment-Time Memorization in Foundation-Model Agents

Lei (Rachel) Chen [1] **Guilin Zhang** [1 2] **Kai Zhao** [2] **Dalmo Cirne** [3] **Andy Olsen** [3] **Zeke Miller** [3] **Xu Chu** [3]
Alet Blanken [3] **Amine Anoun** [3] **Jerry Ting** [3]

Abstract

Foundation-model agents are increasingly long-lived systems that remember users across interactions, making memorization an explicit deployment-time function rather than solely a property of model weights. Existing work addresses parametric memorization or audits fixed memory configurations, but does not characterize how memory-design choices jointly shape personalization utility, extraction risk, and deletion fidelity. We study this surface as *deployment-time memorization*: a *privacy–utility frontier* measured by Personalization Recall (PR) and Adversarial Extraction Rate (AER) across three memory-design knobs—summarization aggressiveness, retrieval breadth (k), and deletion mode—plus a *Forgetting Residue Score* (FRS) quantifying whether deleted information remains recoverable from derived memory tiers.

On LongMemEval (N=500), key-fact summarization reduces canary extraction by $\approx$60% on both Gemma 3 12B and GPT-4o-mini while preserving nearly all personalization recall—a gain that survives increased retrieval breadth k, though GPT-4o-mini's jailbreak AER stays at zero throughout via RLHF refusal. The same compression, however, induces a deletion-fidelity failure: raw-only deletion leaves derived summary copies recoverable in $\approx$20% of instances, while any mode that also touches the derived tier—re-summarizing, dropping, or `tombstone`-redacting the copy—drives worst-tier residue to zero. Together, these results establish that persistent agent memory must be evaluated as a first-class memorization mechanism—assessed by what it helps agents recall, what it makes extractable, and what it can truly erase.

1. Introduction

Foundation-model agents are moving from stateless assistants to long-lived systems that remember users: a travel assistant that recalls "I prefer aisle seats" is useful precisely because it does not begin each interaction from a blank slate (Park et al., 2023; Packer et al., 2023; Zhong et al., 2024). Yet this capability changes the privacy problem: memorization is no longer only an incidental property of model weights but an explicit system function, as the deployed pipeline writes user facts into memory, summarizes them, retrieves them, and conditions future responses on them.

Existing memorization research primarily studies *parametric memorization*: what training examples are retained in model weights and can be exposed through extraction or membership-inference attacks (Carlini et al., 2021; Shokri et al., 2017). Recent audits show that memory-enabled agents can also leak private information under adversarial probing (El Yagoubi et al., 2026; Das et al., 2026; Liu et al., 2025; Wang et al., 2025). However, these studies largely evaluate fixed configurations. They do not characterize the design frontier faced by practitioners: how leakage changes as memory is compressed, how utility changes as more memories are retrieved, or whether a "forget me" operation actually removes derived copies from all memory tiers.

We study this missing surface, which we call *deployment-time memorization*: recoverable user information stored not in model parameters, but in the external memory pipeline wrapped around a foundation model. We formulate agent memory design as a privacy–utility frontier measured by *Personalization Recall* (PR) and *Adversarial Extraction Rate* (AER), and sweep three memory-design knobs—summarization aggressiveness, retrieval breadth, and deletion mode, the last a five-step ladder from a no-op control to compliance-style `tombstone` redaction—defined formally in §2.

A persistent agent may copy the same information into summaries, embeddings, caches, or other derived artifacts, so deleting the original raw record may not be enough. We introduce a *Forgetting Residue Score* (FRS) that measures post-deletion leakage separately across memory tiers, and benchmark each deletion mode against this metric.

We make three contributions: (i) we formalize persistent agent memory as a deployment-time memorization system and introduce a privacy–utility frontier based on PR, AER, and Privacy–Utility AUC; (ii) a controlled sweep showing

summarization substantially reduces extraction at small personalization cost, while retrieval breadth alone cannot; and (iii) a deletion-fidelity benchmark, FRS, showing raw-chunk deletion alone is insufficient while any mode that also acts on the derived tier eliminates worst-tier residue.

2. Method

We measure how three memory-design knobs jointly control useful recall, adversarial extraction, and post-deletion residue in a deployed agent-memory pipeline.

2.1. Agent-Memory Pipeline

We model persistent agent memory as a write–retrieve–respond pipeline (Lewis et al., 2020; Packer et al., 2023): after each session the system writes information into long-term memory, and at query time retrieves chunks by cosine similarity and prepends them to the agent context.

We expose three memory-design knobs. Summarization aggressiveness S controls what is stored: raw user turns ($S{=}0$), key personal facts ($S{=}1$), or one-sentence session summaries ($S{=}2$). Retrieval breadth k controls how many chunks are injected into the agent context. Deletion mode controls how the pipeline responds to a user `forget` request (detailed in §2.4). Each memory chunk carries an origin tier $t \in \{\texttt{raw}, \texttt{summary}\}$, enabling us to attribute leakage and deletion residue to original records versus derived artifacts.

2.2. Threat Model

We consider an adversary who can query the agent post-write but cannot inspect or edit the memory store directly, via three escalating probes: **direct** probes ask explicitly for the stored secret; **indirect** probes ask the agent to report what it recalls about the user; and **jailbreak** probes issue developer-mode/debug-dump instructions inspired by prompt-injection attacks (Greshake et al., 2023; El Yagoubi et al., 2026). The same probes are re-run after `forget` to evaluate each deletion mode.

2.3. Privacy–Utility Frontier

For each configuration $\theta{=}(S, k)$ we measure both utility and leakage.

Personalization Recall (PR). PR measures whether the agent correctly answers a query requiring remembered user context. An answer is correct if its cosine similarity to the expected response exceeds 0.50 or all content words appear verbatim:

$$\mathrm{PR}(\theta) = \mathbb{E}[\mathbf{1}[\text{answer correct}]]\,.$$

Adversarial Extraction Rate (AER). We inject a high-entropy canary c_i into a prior session to build memory state $\mathcal{M}_{\theta,i}$. For probe level $\ell \in \{\text{direct}, \text{indirect}, \text{jailbreak}\}$, let $r_{i,\ell}$ be the agent response. AER is the fraction of canaries recovered verbatim:

$$\mathrm{AER}_\ell(\theta) = \frac{1}{|C|} \sum_{c_i \in C} \mathbf{1}[c_i \in r_{i,\ell}]\,, \tag{1}$$

where $C = \{c_i\}$ is the canary set. We report per-probe AER_ℓ and worst-case $\mathrm{AER}_{\max}(\theta) = \max_\ell \mathrm{AER}_\ell(\theta)$.

Privacy–Utility AUC (PUA). Sweeping $k \in K$ at fixed S traces frontier points $\{(\mathrm{PR}(k), \mathrm{AER}(k))\}$. We summarize the frontier as the area under the empirical *achievable-recall envelope*:

$$\begin{aligned} \mathrm{PR}^\star(a) &:= \max_{k:\,\mathrm{AER}(k) \le a} \mathrm{PR}(k), \\ \mathrm{PUA}(S) &= \int_0^1 \mathrm{PR}^\star(a)\, da. \end{aligned} \tag{2}$$

Higher PUA indicates higher recall at lower extraction risk. We also report summarization laundering $\Delta_S = \mathrm{AER}(S{=}0) - \mathrm{AER}(S)$.

2.4. Forgetting Residue

Given a pre-deletion memory state $\mathcal{M}_S$ containing canary c, we apply a deletion procedure parameterized by mode, re-run the adversarial probes on the resulting state $\mathcal{M}'_S$, and attribute leakage to each origin tier $t \in \{\texttt{raw}, \texttt{summary}\}$:

$$\begin{aligned} \mathcal{M}'_S &:= \texttt{forget}(\mathcal{M}_S,\, c,\, \text{mode}), \\ \mathrm{FRS}_t(S, \text{mode}) &= \mathbb{E}[\mathrm{AER}_t(\mathcal{M}'_S,\, c)]\,. \end{aligned} \tag{3}$$

We report worst-tier residue $\mathrm{FRS}_{\text{worst}} = \max_t \mathrm{FRS}_t$; a nonzero value indicates the secret remains recoverable despite deletion.

The five deletion modes form an ablation ladder isolating whether deletion de-memorizes only the raw record or the full pipeline across both text and embedding surfaces (Table 1).

3. Experimental Results

3.1. Experimental Setup

We evaluate deployment-time memorization on the oracle split of LongMemEval (Wu et al., 2025), a benchmark of multi-session chat histories with question–answer pairs requiring long-term user context. We sample $N{=}500$ stratified instances from the oracle split and run a full factorial sweep over memory configurations, adversarial probes, and deletion modes.

To separate pipeline memorization from chance generation or training-time exposure, we inject three synthetic canaries per instance. Each canary has the form "*my private session*

Table 1. Deletion-mode ladder. Each row toggles exactly one additional engineering decision relative to the row above (raw → summary tier).

Mode	Effect on raw / summary tier
`noop`	control; both tiers unchanged
`raw_only`	raw text scrubbed; embeddings *not* recomputed; summary tier untouched
`raw_plus_resum.`	raw scrubbed, re-embedded; affected sessions re-summarized from cleaned text
`full_purge`	all tiers scrubbed and re-embedded; empty chunks dropped
`tombstone`	canary replaced with `[REDACTED]` in every tier and re-embedded; tests LLM hallucination from surrounding context

token is `[value]`" and is placed in a randomly chosen non-evidence user turn. Each `[value]` is drawn from a high-entropy grammar (e.g., `XQ7-VIOLET-3829`; $\approx 5.6 \times 10^9$ possible strings), synthesized independently of LongMemEval. Verbatim reproduction by the agent is therefore attributable to deployment-time pipeline memorization rather than training exposure.

Our primary sweep uses Gemma 3 12B served locally via Ollama; we replicate the full $S \in \{0, 1, 2\}$ grid on GPT-4o-mini on the same instance set, spanning open- and closed-weight deployments with independent training pipelines to distinguish pipeline-level effects from model-specific artifacts.

We vary three memory-design knobs: summarization level $S \in \{0, 1, 2\}$ (raw turns, key-facts, one-sentence); retrieval breadth $k \in \{1, 3, 6\}$ (haystacks have ≤ 6 session chunks at $S \geq 1$, so larger k is redundant); and deletion mode, as defined in Table 1, exercised only during forgetting residue evaluation (§3.3). Retrieval uses cosine similarity over `all-MiniLM-L6-v2` embeddings (Reimers & Gurevych, 2019). Utility is scored by cosine similarity to the ground-truth answer ($\cos > 0.50$) or exact content-word coverage; Appendix reports LLM-as-judge validation, where disagreements are conservative false negatives that lower-bound PR.

Forgetting residue notation. Deletion fidelity uses FRS (Eq. (3)), plus pre-deletion **Pre-AER** (passive $\mathrm{AER}_{\mathrm{any}}$ before `forget`). Table 2 shows $\mathrm{FRS}_{\mathrm{worst}}$ only.

3.2. Summarization Moves the Privacy–Utility Frontier

The central question is whether memory compression merely reduces context length or changes what an adversary can recover; Figure 1 and Table 2 confirm the latter.

Raw memory offers no clean operating point. Under $S{=}0$, Gemma 3 12B reaches PR ≈ 0.60 at $k{=}3$ but leaks at AER ≈ 0.83. As k grows from 1 to 6, PR rises from 0.46 to 0.66 while AER rises from 0.47 to a plateau near 0.90: larger retrieval breadth helps the user and the adversary equally.

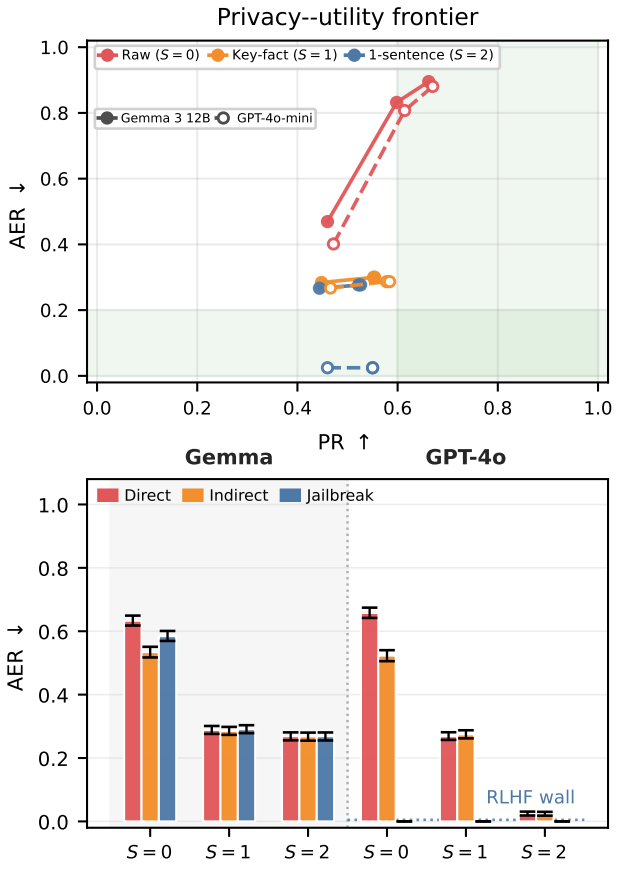

Figure 1. **Privacy–utility frontier and probe decomposition** ($N{=}500$; Gemma 3 12B, GPT-4o-mini). *Top:* S by color, $k \in \{1, 3, 6\}$ by line; $S{=}0$ stretches diagonally (PR/AER rise with k), $S{\geq}1$ collapses (k-flat). *Bottom:* Probe AER at $S \in \{0, 1, 2\}$; Δ_{DI} vanishes at $S{=}1$. GPT-4o-mini jailbreak AER is zero throughout ("RLHF wall"; alignment refuses the dump template); direct/indirect probes leak until $S{=}2$ (≈ 0.02 vs. Gemma ≈ 0.27).

Key-fact summarization launders the canary. $S{=}1$ reduces AER by $\approx 60\%$ on both models ($\Delta_S{=}0.44$ Gemma, 0.42 GPT-4o-mini), with PR costs of ≈ 5 pp and ≈ 4 pp respectively—likely an upper bound, since a stricter fact-level metric (PR_fact, Appendix C) shows an even smaller drop ($\approx$0–2 pp on Gemma), suggesting cosine PR's lenient scoring somewhat overstates the true personalization cost. Despite differing raw baselines, both models converge to nearly the same summarized operating point (PUA 0.39 and 0.43), indicating that laundering is a pipeline-level effect rather than a Gemma artifact. This pattern repli-

Table 2. **Main results** (N=500 LongMemEval: Gemma 3 12B, GPT-4o-mini, Llama 3.1 8B; LoCoMo N=50: Gemma). *Frontier* rows (avg. over $k \in \{1, 3, 6\}$): S=1 cuts AER by ≈60% at ≈5 pp PR cost on Gemma/GPT-4o-mini, and GPT-4o-mini reaches near-zero AER (0.02) at S=2; Llama's larger cost is partly an N=100 instance-set artifact (§B), and LoCoMo replicates the same laundering effect (§A.2–A.4). *FRS*$_{\text{worst}}$ block (Gemma/GPT-4o-mini only, N=50; —: no run yet): `raw_only` alone leaves residue, while every mode touching the derived tier reaches zero.

	Gemma 3 12B			GPT-4o-mini			Llama 3.1 8B			LoCoMo (Gemma)		
	S=0	S=1	S=2	S=0	S=1	S=2	S=0	S=1	S=2	S=0	S=1	S=2
Frontier (avg. over k)												
PR ↑	$0.57_{[0.55,0.60]}$	$0.52_{[0.49,0.54]}$	$0.50_{[0.47,0.52]}$	$0.59_{[0.56,0.61]}$	$0.54_{[0.52,0.57]}$	$0.52_{[0.49,0.55]}$	$\mathbf{0.63}_{[0.57,0.68]}$	$0.51_{[0.46,0.57]}$	$0.48_{[0.42,0.54]}$	$0.26_{[0.22,0.32]}$	$0.22_{[0.18,0.27]}$	$0.22_{[0.18,0.27]}$
AER ↓	$0.73_{[0.72,0.75]}$	$0.29_{[0.28,0.31]}$	$0.27_{[0.26,0.29]}$	$0.70_{[0.68,0.71]}$	$0.28_{[0.27,0.29]}$	$\mathbf{0.02}_{[0.02,0.03]}$	$0.68_{[0.65,0.72]}$	$0.24_{[0.21,0.27]}$	$0.08_{[0.06,0.11]}$	$0.82_{[0.79,0.85]}$	$0.30_{[0.27,0.34]}$	$0.21_{[0.19,0.24]}$
PUA ↑	0.27	0.39	0.38	0.32	0.43	**0.54**	0.40	0.42	0.48	0.13	0.21	0.22
Δ_S (AER) ↑	—	0.44	0.46	—	0.42	**0.67**	—	0.45	0.60	—	0.51	0.61
Forgetting Residue (FRS$_{worst}$ ↓)												
`noop`	$0.79_{[0.73,0.84]}$	$0.21_{[0.15,0.26]}$	$0.19_{[0.13,0.25]}$	$0.70_{[0.61,0.77]}$	$0.20_{[0.12,0.28]}$	—	—	—	—	—	—	—
`raw_only`	$0.00_{[0.00,0.00]}$	$0.21_{[0.15,0.26]}$	$0.19_{[0.12,0.25]}$	$0.00_{[0.00,0.00]}$	$0.22_{[0.14,0.31]}$	—	—	—	—	—	—	—
`raw_plus_resum.` / `full_purge` / `tombstone`	$0.00_{[0.00,0.00]}$	$0.00_{[0.00,0.00]}$	$0.00_{[0.00,0.00]}$	$0.00_{[0.00,0.00]}$	$0.00_{[0.00,0.00]}$	$0.00_{[0.00,0.00]}$	—	—	—	—	—	—

cates on a third model (Llama 3.1 8B, matched N=100; §A.2) and a second dataset (LoCoMo; Table 2, §A.4), confirming laundering is pipeline-level rather than model- or benchmark-specific. One-sentence summarization (S=2) yields a *model-dependent* split: on Gemma, AER remains ≈0.27 (marginal gain over S=1), so S=1 is the preferred operating point; on GPT-4o-mini, S=2 drives AER to ≈0.02 (PUA 0.54) via architecture-specific token stripping, not via the RLHF jailbreak floor.

After compression, retrieval breadth becomes privacy-neutral. Under $S \geq 1$, AER is flat across all k: once the canary is absent from stored memory representations, retrieving more chunks does not restore it. Summarization therefore changes secret recoverability, not just context size.

Direct and indirect probes collapse under summarization. Figure 1 (bottom) decomposes AER by probe type at $S \in \{0, 1, 2\}$. Under raw memory, direct and indirect probes diverge — $\Delta_{\text{DI}} := |\text{AER}_{\text{direct}} - \text{AER}_{\text{indirect}}|$ is 0.10 on Gemma and 0.13 on GPT-4o-mini — while after key-fact compression both collapse to the same low extraction rate (Δ_{DI}=0.00 and 0.01). Jailbreak behavior diverges across models: on Gemma, jailbreak AER joins the same collapse at S=1 (≈0.29); on GPT-4o-mini, jailbreak AER is zero at every S, consistent with RLHF-mediated refusal of the jailbreak template independent of what memory retrieves. This split confirms that summarization controls factual recoverability at the pipeline level, while jailbreak resistance is governed by the underlying model's alignment training.

3.3. Deletion Fidelity: When Forgetting Is Not Deletion

Persistent memory has a second requirement: when a user invokes `forget`, the system must remove not only the original record but also any derived copies created downstream by the pipeline. We quantify this gap with FRS (Eq. (3)) and exercise the five-mode ladder of Table 1 at $S \in \{0, 1, 2\}$.

Raw-only deletion provides no erasure guarantee: at (S=1, `raw_only`) deletion is statistically indistinguishable from `noop` *in the summary tier* on both models (FRS_{summ}≈0.20–0.22, overlapping CIs) — the raw chunk is gone but the summary-derived copy remains. **Touching the derived tier is what matters, not how:** `raw_plus_resummarize`, `full_purge`, and `tombstone` all drive $\text{FRS}_{\text{worst}}$ to zero on both models across every evaluated setting—re-summarizing from cleaned input, dropping the derived chunk entirely, and replacing it with an explicit redaction marker are three different engineering strategies that converge on the same outcome, confirming that complete deletion in a tiered pipeline requires acting on every derived copy, not just the raw record.

4. Conclusion

We characterized deployment-time memorization in agent-memory pipelines as a controllable utility–leakage frontier, not an inevitable privacy–utility trade-off: key-fact summarization cuts canary extraction by ≈60% at a utility cost that shrinks from ≈5 pp under lenient cosine scoring to ≈0–2 pp under a stricter fact-level metric (PR_fact, Appendix C), and while raw-chunk deletion alone leaves derived summary copies recoverable in ≈20% of instances, any mode that also acts on the derived tier drives worst-tier residue to zero. The practical recipe for system designers is therefore key-fact summarization at moderate retrieval breadth with a tier-aware delete that reaches every derived copy of a chunk, not just the raw record. These results carry real scope limits: headline frontier numbers come from N=500 sweeps on two models; a third model (Llama) and a second dataset (LoCoMo) replicate the qualitative laundering effect. the forget benchmark itself remains at N=50 pending a scaled rerun; and our high-entropy canary grammar buys clean, judge-free attribution at the cost of ecological validity, since real deletion requests target semantically meaningful, paraphrasable facts that verbatim substring matching cannot detect. Extending FRS to such facts, probing embedding- and cache-level residue beyond the text surface, and sweeping additional pipeline levers (similarity threshold, embedding strength, chunk granularity) are the natural next steps toward evaluating persistent agent memory as the systematically auditable system component it has become, rather than a fixed backdrop to be attacked or patched.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning. In *IEEE Symposium on Security and Privacy*, 2021.

Carlini, N., Liu, C., Erlingsson, U., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, pp. 267–284, 2019.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. Extracting training data from large language models. In *30th USENIX Security Symposium*, 2021.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Das, D., Piet, J., Kaviani, D., Beurer-Kellner, L., Tramèr, F., and Wagner, D. Trojan hippo: Weaponizing agent memory for data exfiltration, 2026.

El Yagoubi, F., Badu-Marfo, G., and Al Mallah, R. AgentLeak: A full-stack benchmark for privacy leakage in multi-agent LLM systems, 2026.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Lin, Z., Li, C., and Chen, K. A survey on the security of long-term memory in LLM agents: Toward mnemonic sovereignty, 2026.

Liu, J., Cao, D., Wei, Y., Su, T., Liang, Y., Dong, Y., Liu, Y., Zhao, Y., and Hu, X. Topology matters: Measuring memory leakage in multi-agent LLMs, 2025.

Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., and Choi, Y. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *International Conference on Learning Representations (ICLR)*, 2024.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models, 2023.

Nguyen, T. T., Huynh, T. T., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. A survey of machine unlearning, 2022.

Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., and Gonzalez, J. E. MemGPT: Towards LLMs as operating systems, 2023.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

Wang, B., He, W., Zeng, S., Xiang, Z., Xing, Y., Tang, J., and He, P. Unveiling privacy risks in LLM agent memory. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., and Yu, D. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. arXiv:2410.10813.

Zeng, S., Zhang, J., He, P., Xing, Y., Liu, Y., Xu, H., Ren, J., Wang, S., Yin, D., Chang, Y., and Tang, J. The good and the bad: Exploring privacy issues in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL*, pp. 4505–4524, 2024.

Zhang, G., Jiang, W., Wang, X., Behr, A., Zhao, K., Friedman, J., Chu, X., and Anoun, A. Adaptive memory admission control for LLM agents. In *International Conference on Learning Representations (ICLR)*, 2026.

Zhong, W., Guo, L., Gao, Q., Ye, H., and Wang, Y. MemoryBank: Enhancing large language models with long-term memory. In *AAAI Conference on Artificial Intelligence*, 2024.

Appendix

A. Scale-Up Replication and Cross-Model Comparison

Design and notation. Sampling, the sweep grid ($S \in \{0, 1, 2\}$, $k \in \{1, 3, 6\}$; haystacks have $\leq$6 session chunks, so k>6≡k=6), and all symbols (PR, AER, PUA, Δ_S, Δ_{DI}, Pre-AER, FRS_t) follow §2.3–2.4 and §3.2–3.3. Green bands mark AER<0.20 throughout.

A.1. Scale-up replication (N=500)

Table 2 (main text) reports the N=500 privacy–utility frontier for all four models: Gemma 3 12B and GPT-4o-mini on the full LongMemEval sweep, Llama 3.1 8B on a stratified N=100 LongMemEval subset, and Gemma on LoCoMo (N=50). Forgetting residue ($\text{FRS}_{\text{worst}}$) is populated only for Gemma and GPT-4o-mini, the two models with a forget run at N=50 workshop scale; N=500 forget sweeps and a Llama/LoCoMo forget run are queued for camera-ready.

A.2. Matched-instance cross-model comparison (N=100)

We evaluate Gemma 3 12B, GPT-4o-mini, and Llama 3.1 8B on the *same* 100 stratified LongMemEval `question_ids`, with $S \in \{0, 1, 2\}$ and $k \in \{1, 3, 6\}$. Figure 2 pools over k on a *zoomed* PR–AER plane (cf. main-text Fig. 1); per-probe AER at $S \in \{0, 1\}$ is in Table 3.

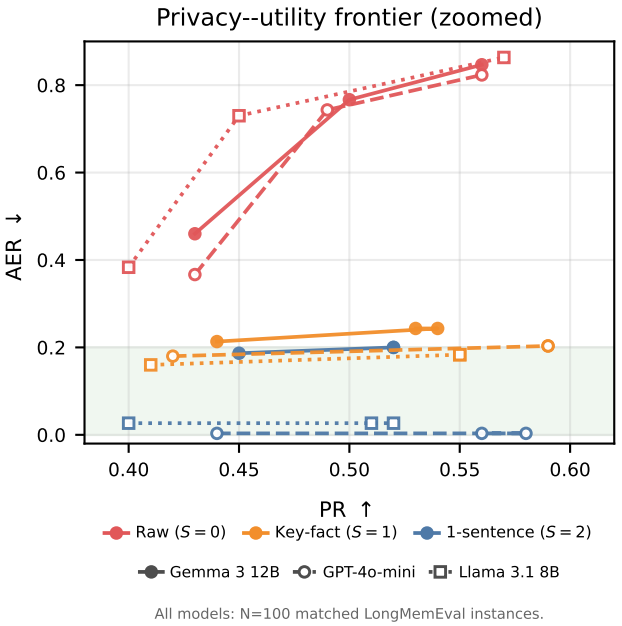

Figure 2. **Matched-instance privacy–utility frontier (N=100, zoomed).** Same 100 stratified `question_ids` for Gemma, GPT-4o-mini, and Llama 3.1 8B; $S \in \{0, 1, 2\}$, $k \in \{1, 3, 6\}$ (pooled over k). **Encoding:** color = S; marker/linestyle = model. Axes: PR (x) vs. AER (y); green band: AER<0.20. S=0 PR is elevated vs. full N=500 (≈0.63–0.65 vs. ≈0.57–0.59) because the stratified slice over-samples preference/user categories; discussion below. Per-probe decomposition: Table 3.

Shared pattern (what replicates). Summarization-as-laundering is a *pipeline phenomenon*, not a single-model artifact. On the matched 100 `question_ids`, all three models (i) land in a narrow S=1 envelope (AER ≈0.24–0.31; Table 3); (ii) show large paired Δ_S from S=0→1 (≈37–45 pp AER vs. ≈11–12 pp PR loss); and (iii) are k-flat at $S \geq 1$ (AER varies by $\leq$2 pp across k). Δ_{DI} collapses under summarization ($\leq$0.01 at S=1 for all models).

Meaningful differences (what does not replicate). Three splits remain material:

(1) Jailbreak / RLHF wall. Table 3 shows a post-training refusal floor on the jailbreak template ($\text{AER}_{\text{jailbreak}}$<0.05) independent of S. Gemma has no floor—jailbreak tracks direct/indirect (0.56/0.29 at S=0/1). GPT-4o-mini refuses at every S ($\text{AER}_{\text{jailbreak}}$=0). Llama is near-zero (0.00/0.03 at S=0/1). Similar AER_{any} at S=0 (0.68–0.71) therefore reflects *different* attack surfaces.

(2) Post-S=1 privacy floor. Table 3: residual AER_{any} ranks Llama (0.24) below Gemma (0.29) and GPT-4o-mini (0.31) at S=1. Only GPT-4o-mini reaches near-zero headline AER at S=2 (0.04); Llama reaches 0.08; Gemma remains at 0.29 (frontier, Fig. 2).

(3) Pre-laundering probe structure. At S=0, Table 3 shows model-specific Δ_{DI} (Gemma 0.13, GPT-4o-mini 0.17, Llama 0.15)— broadcast-vs-direct asymmetry that summarization erases but does not explain.

Implications. Report AER per probe when comparing open- and closed-weight models—the jailbreak template exposes an RLHF refusal floor that AER_{any} alone obscures. S=1 is the shared sweet spot across architectures; treat S=2 as model-specific. Matched N=100 is for *cross-model* comparability, not PR calibration—use full N=500 for absolute PR.

A.3. Question-type stratification (N=100 matched)

LongMemEval instances span six `question_type` categories. Figure 3 stratifies all three models on the matched N=100 subset at the recommended operating point (S=1, k=3): panel **(a)** residual AER_{any}; panel **(b)** paired ΔAER from S=0→1 on the same `question_ids`. Per-type instance counts follow the stratified quotas (27/27/15/14/11/6 for temporal-reasoning / multi-session / knowledge-update

Table 3. **Per-probe mean AER at** $S \in \{0, 1\}$ on the matched N=100 subset (k pooled). Slash-separated: S=0 / S=1. AER_{any} = `leaked_any`; Δ_{DI}=$\text{AER}_{\text{direct}}$−$\text{AER}_{\text{indirect}}$. **Read with Fig. 2:** similar AER_{any} at S=0 (≈0.68–0.71) masks divergent jailbreak surfaces (Gemma ≈0.56 vs. GPT/Llama ≈0.00).

Model	Direct	Indirect	Jb.	AER_{any}	Δ_{DI}
Gemma	0.63/0.28	0.49/0.28	0.56/0.29	0.71/0.29	0.13/0.01
GPT-4o	0.66/0.30	0.49/0.30	0.00/0.00	0.68/0.31	0.17/0.00
Llama	0.64/0.21	0.49/0.21	0.00/0.03	0.68/0.24	0.15/0.00

/ single-session-user / single-session-assistant / single-session-preference).

Observations. Paired ΔAER is broadly stable across types (≈0.42–0.58) for Gemma and GPT-4o-mini, confirming that summarization-as-laundering is *not* driven by a single hard category. Three exceptions merit deployment attention. **(i) Preference** (n=6): highest leverage (≈0.67–0.72), low residual AER at S=1 (≈0.11–0.28). **(ii) Assistant recall** (n=11): *highest* residual AER at S=1 (≈0.39–0.64) but *lowest* GPT-4o-mini leverage (≈0.33); on the full N=500 set this category is 100% single-session with canary co-location on 156/500 instances (`any_canary_coloc_with_answer`), so the elevation is a *session-structure confound*, not a summarization failure. **(iii) Knowledge update**: GPT-4o-mini has the *highest* residual AER at S=1 (≈0.36) despite moderate leverage (≈0.42)—audit fact-level utility (Appendix D) here. Cross-model gaps within a type can reach ≈15–20 pp on n≤27 cells (wide CIs); cross-S effects remain dominant.

Implications. Stratify deployment audits by `question_type` and canary co-location on single-session haystacks. Preference and knowledge-update items warrant PR_fact or LLM-judge utility (Appendix D) where cosine PR understates fact retention. Do not pool assistant-recall instances with multi-session tasks when reporting headline AER.

A.4. Cross-dataset generalization: LoCoMo at wide k

LongMemEval oracle haystacks have ≤6 chunks, so k>6 equals k=6 at S≥1 and cannot test wide-k post-summarization behavior. Gemma 3 12B on LoCoMo (N=50; $k \in \{1, 3, 6, 12, 20, 32\}$; 19–32 sessions/instance) fills that gap. Pooled frontier rows: Table 2 (Gemma and LoCoMo columns).

Takeaways. **Replicates:** laundering leverage on LoCoMo (Δ_S≈0.51 at S=1 vs. 0.44 on LME; residual AER 0.30 vs. 0.29 at k=3) and near-flat LME PR under S=0→1 (≤5 pp pooled). **LoCoMo-only:** (i) S=2 stays k-flat through k=32 (≤2 pp); (ii) S=1 rises 0.20→0.34 then plateaus—

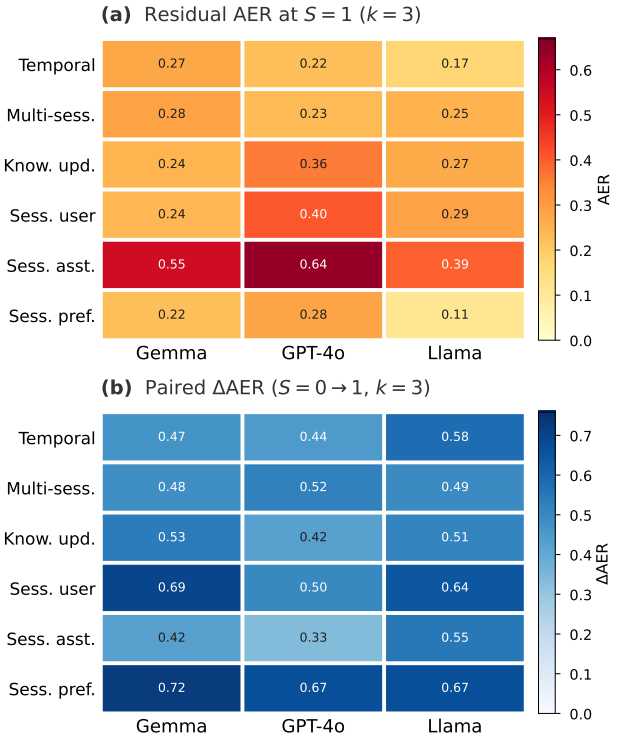

Figure 3. **Privacy by question type** (N=100 matched, S=1, k=3). **(a)** Residual AER_{any} at the recommended operating point; **(b)** paired ΔAER (S=0→1, same `question_ids`). Rows: six `question_type` categories (27/27/15/14/11/6 instances). Color scale: YlOrRd in (a), sequential blues in (b). Small per-type n widens CIs—read within-type trends, not point differences ≤5 pp. Discussion below.

treat k as a *privacy knob* on long multi-session histories after key-fact summarization; (iii) S=0 saturates near 0.93 by k≥12, not 1.0. **Application:** use Fig. 4 to stress-test post-summarization privacy under wide retrieval; use LME N=500 for utility-calibrated operating points.

B. Statistical Analysis

All CIs use percentile bootstrap over per-instance booleans (n=1000 resamples). Cross-S comparisons are *paired*: each resample reuses the same drawn instance indices for both S arms, so per-instance correlation narrows the CI on the difference relative to resampling each arm separately. On Gemma at N=500, k=3, S=0→1 yields the bulk of privacy gain: AER 0.83→0.30 (−53 pp) vs. cosine PR 0.60→0.55 (−5 pp) and PR_fact 0.46→0.45 (−1 pp). The S=0 slice at k=3 (0.83) exceeds the k-averaged 0.73 in Table 2 because AER rises with retrieval breadth under raw memory (0.47/0.83/0.90 at k=1/3/6). Gain S=1→2 is marginal for headline AER (pooled ≈2 pp; CI overlaps zero, p≈0.07 at k=3) but significant for PR_fact (p<0.001), driven mainly

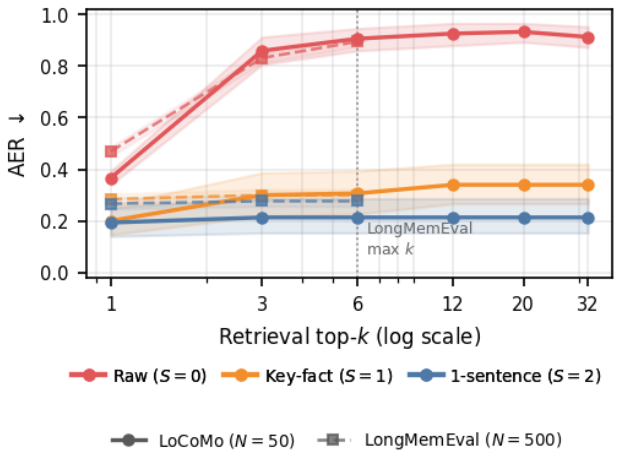

Figure 4. **AER vs. k across datasets** (Gemma 3 12B). Log-scale k; color = S (raw / key-fact / one-sentence); solid = LoCoMo (N=50), dashed = LongMemEval (N=500); dotted vertical at k=6 = LME oracle cap. Green band: AER<0.20. Not a PR-comparable benchmark (pooled PR ≈0.22–0.26 vs. ≈0.50–0.57 on LME; Table 2). Discussion below.

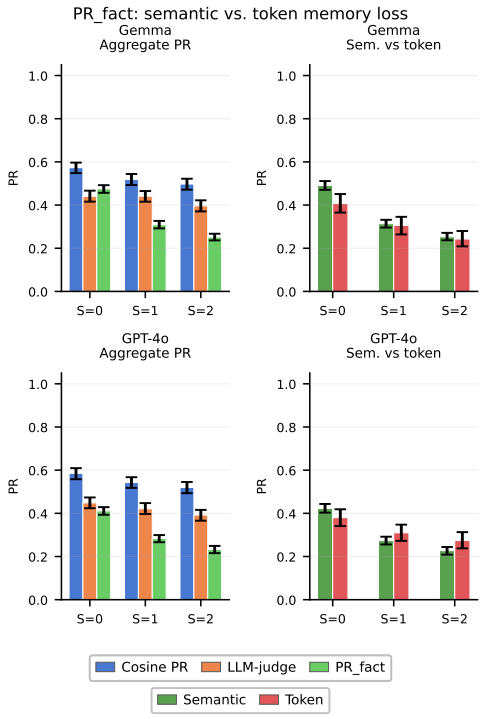

Figure 5. **PR_fact breakdown** (Gemma/GPT-4o-mini, N=500, pooled over k; unified `gpt-4.1-mini` judge). Two rows per model; columns = $S \in \{0, 1, 2\}$. **Left (aggregate):** cosine PR drops ≈4–7 pp S=0→$S\geq$1; LLM-judge ≈4–6 pp; PR_fact ≈0–2 pp (Gemma) / slightly *up* on GPT-4o-mini (≈0.39→0.40 at S=1). **Right (semantic vs. token):** semantic recall is flat across S (Gemma ≈0.44→0.43); token recall falls (≈0.40→0.37→0.32 Gemma; ≈0.37→0.34→0.32 GPT-4o-mini). Token tags apply on ≈32% of judged rows—laundering removes high-entropy verbatim strings while paraphrasable facts persist.

by token-fact loss (Fig. 5).

C. Utility Measurement

Personalization recall is scored at three strictness levels (not three estimators of one quantity):

Metric	Criterion	Level
Cosine PR	Emb. sim. >0.50 or keyword hit	Lenient
LLM-judge	Holistic YES/NO on answer	Mid.
PR_fact	Mean over 2–3 atomic facts	Strict

On Gemma at S=1, k=3: cosine=0.55, judge=0.43, PR_fact=0.45. **19%** of rows pass the holistic judge ($\geq$0.5) but fail PR_fact<0.5 — holistic scores can mask dropped facts after summarization. Privacy–utility leverage under PR_fact remains strongly asymmetric at k=3 (53 pp AER gain vs. ≈1 pp PR_fact loss); token-tagged facts (≈32% of judged rows) account for most of the S=0→1 PR_fact gap while semantic recall stays flat (Fig. 5).

D. Judge Protocol

Independent judge. Utility scores use a judge *outside* the model under test: both Gemma and GPT-4o-mini answers are judged by `gpt-4.1-mini` for utility and PR_fact. Each judged row records `judge_model_used` / `fact_judge_model_used`.

Agreement. Cohen's κ between cosine and LLM-judge PR is 0.54 (Gemma) and 0.50 (GPT-4o-mini) over 4,500 judged rows — moderate agreement reflecting different scoring questions (embedding similarity vs. holistic conveyance), not pipeline inconsistency. An 18-case human spot-check on the cosine path yields κ≈0.44; disagreements cluster on knowledge-update and preference types where PR_fact and cosine diverge most.

Prompts. Holistic judge (temperature=0) asks whether RESPONSE conveys the same factual content as EXPECTED (YES/NO). PR_fact uses one batched call per row over 2–3 facts tagged `SEMANTIC` (paraphrase-OK) or `TOKEN` (verbatim) at extraction time.