CAUSAL MEMORY POLICY: MAKING MEMORY UTILITY IDENTIFIABLE BY INTERVENING ON RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory-augmented large language models must decide which memories to retain, and recent systems do so by estimating each memory's effect on task performance. However, these estimates rely entirely on retrieved memories. When a memory is never retrieved, store-level interventions produce identical outcomes, leaving its utility unidentified. This is a retrieval-level positivity violation, invisible to diagnostics that examine only memory operations.

We introduce Causal Memory Policy (CMP), a causal framework that restores identification by intervening on retrieval itself, reserving a fixed number of context slots for memories sampled with known propensity scores. CMP estimates memory utility via self-normalized inverse propensity weighting under a balanced assignment design with fixed sampling propensities. We prove the causal factorization of memory utility through retrieval, the unbiasedness and exact variance of the estimator, and the optimal decision rule under irreversible operations. Empirically, identification fails for 54% of required memories on LongMemEval and 67% on LoCoMo, and the failure persists in a deployed memory system. CMP improves discrimination between required and non-required memories from 0.54 to 0.66 AUC. Finally, we show that identified memory utility alone is insufficient for retention decisions: per-query utility reaches 0.78 AUC on the query for which it is estimated. Code is available at: `https://anonymous.4open.science/r/cmp-release-D0C3/`.

1 INTRODUCTION

Large language models (LLMs) exhibit remarkable capabilities across a broad spectrum of tasks, but remain fundamentally constrained by fixed context windows that limit the information accessible at inference time. As the scope and duration of LLM deployments grow, effective management of what is remembered, forgotten, and reorganized becomes a first-order design challenge. Memory-augmented LLMs (Zhong et al., 2024; Wang et al., 2024b) address this limitation by maintaining external memory stores that persist across interactions, enabling long-horizon reasoning, user personalization, and incremental knowledge accumulation. A detailed discussion of related work is in Appendix A.

Deciding what to retain requires estimating the utility of each memory. Recent systems estimate this quantity from per-memory success and failure counts (Simsek, 2026), paired trajectories with and without a candidate memory (Gao et al., 2026), controlled interventions on the candidate set (Srivastava, 2026), and explicit retention budgets (Wu et al., 2026). All of these estimators rely solely on memories already retrieved by the logging policy (see Appendix H.1 for an example). This raises four fundamental questions: (1) *When is memory utility identified from logged interactions?* Identification requires observing a memory both present and absent in the retrieved context, which no deterministic logging policy guarantees. (2) *What does an estimator return when the estimand is not identified?* A degenerate contrast often yields an estimate near zero, making identification failure indistinguishable from genuinely low memory utility. (3) *Which memory management policies are affected?* Only policies that rank memories by estimated utility are susceptible; recency- or frequency-based heuristics never evaluate the affected memories. (4) *How can identification be restored, and at what cost?* Randomization must be introduced into retrieval, since perturbing the memory store leaves the retrieved context unchanged whenever the perturbed memory is not retrievable.

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.

The root cause is that **retrieval mediates the effect of the memory state on task output**. A memory influences an answer only if it is retrieved into the context the model reads, so intervening on the memory store affects the output only through what retrieval subsequently surfaces. Memory utility therefore factorizes into two components: the probability of retrieval and the contribution once retrieved. Logged interactions cannot distinguish a memory with genuinely low utility from one that was never given the opportunity to contribute. This is a *positivity violation*, the causal inference condition requiring every unit to have a nonzero probability of receiving each treatment. Because the violation occurs at the level of retrieval, diagnostics that verify exploration only over memory operations fail to detect it. Existing causal approaches to memory management do not address this retrieval-level positivity failure. They either place the ground-truth answer inside the selection loop (Srivastava, 2026), which is unavailable at deployment, or verify overlap only over the operations selected by the policy (Li et al., 2026), leaving retrieval itself uncontrolled.

We propose a corrected view: **memory utility as a retrieval-mediated causal effect**. We formalize a memory-augmented LLM as a structural causal model (SCM) (Pearl, 2009), where retrieval mediates the effect of memory operations on task outputs, and derive the resulting identification condition. Built on this formulation, we develop the **Causal Memory Policy (CMP)**, whose components resolve the above four questions: (i) **Identification**. We derive a factorization of memory utility into retrievability and conditional usefulness, prove the resulting non-identifiability under deterministic retrieval, and explain why unidentified estimators converge to values near zero rather than becoming undefined, answering Questions (1) and (2). (ii) **Restoring identification and cost**. We introduce a randomized exposure design that reserves k context slots for memories sampled with known propensity, restoring positivity by construction while requiring a budget that scales only with the candidate pool being identified, answering Question (4). (iii) **Empirical characterization**. We quantify retrieval-level positivity failure on two benchmarks and a deployed memory system, identifying when utility-based policies are affected and when heuristic policies are not, answering Question (3). (iv) **Decision theory**. We characterize what identified memory utility can support by deriving the Bayes-optimal policy under asymmetric reversible and irreversible operations, and prove that no aggregation over observed queries can predict a memory utility on unseen queries.

We evaluate CMP on LongMemEval (Wu et al., 2024) (long-horizon retention over ~ 490-turn histories), LoCoMo (Maharana et al., 2024) (multi-session dialogue with many queries over a single memory store), HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022) (2–4-hop reasoning, as replication), and Mem0 (Chhikara et al., 2025) (a deployed memory system, as transfer). *We find that retrieval-level identification failure is substantial rather than an implementation artifact: it affects* 54% *of required memories on LongMemEval and* 67% *on LoCoMo, and persists in Mem0, where unretrieved evidence has a median retrieval rank of* 139 *among* 291 *stored memories.* Store-level randomization leaves the retrieved context unchanged on 52.2% of perturbations, causing utility-based eviction to perform no better than chance (lift 0.97). Restricting evaluation to retrieved memories eliminates the gap between the two estimators (0.655 vs. 0.658 AUC), localizing the failure to positivity. CMP improves discrimination between required and non-required memories from 0.51 to 0.78 AUC. Finally, we show that identified utility alone does not determine retention: no aggregation over observed queries predicts a memory's value on unseen queries, indicating that the remaining limitation lies in the estimand rather than the estimator.

2 Problem Formulation

We formalize a memory-augmented LLM as an SCM and memory operations as causal interventions. We show that retrieval mediates all paths from memory state to task output and derive the identification condition imposed by this mediation. A summary of the notation is in Table 5 in the Appendix B.

2.1 Memory System as a Structural Causal Model

Definition 1 (Memory-Augmented LLM as SCM)**.** *A memory-augmented LLM is an SCM* $\mathcal{M} = (\mathcal{U}, \mathcal{V}, \mathcal{F})$*. Exogenous variables* $\mathcal{U} = \{Q, C\}$ *consist of user queries* Q *drawn from a task distribution* $\mathcal{Q}$ *and observed context* C *(conversation metadata, task type), with realizations* q_t *and* c_t*. Endogenous variables* $\mathcal{V} = \{M, R, O, Y\}$ *are defined as:* $M_t \in 2^{\mathbb{M}}$ *is the memory state at time* t *and a finite set of stored memories from universe* $\mathbb{M}$*,* $R_t \subseteq M_t$ *is the retrieved subset for query* $q_t \in Q$*,* $O_t \in \mathcal{O} \triangleq \{\texttt{store}, \texttt{forget}, \texttt{consolidate}, \texttt{noop}\}$ *is the memory operation applied at time* t*, each parameterized by its arguments as in Definition 2, and* $Y_t \in \mathcal{Y}$ *is the output of a downstream task. The structural equations* $\mathcal{F}$ *are* $R_t = f_R(M_t, q_t)$ *(retrieval),* $Y_t = f_Y(R_t, q_t, \theta_{LLM})$ *(generation),* $O_t = f_O(M_t, q_t, c_t)$ *(operation selection), and* $M_{t+1} = f_M(M_t, O_t, q_t)$ *(memory transition).*

Here, c_t denotes the observed context (e.g., conversation metadata and task type) on which the policy conditions when selecting an operation, and θ_{LLM} denotes the frozen LLM parameters. Note that c_t enters f_O but not f_Y: the generator receives only the retrieved memories and the query, so every path from the memory state to the output passes through R_t. This mediation underlies the identification result in Section 2.3 and holds by construction in all our experiments: the generator takes as input only the retrieved memory texts and the question, without any item metadata.

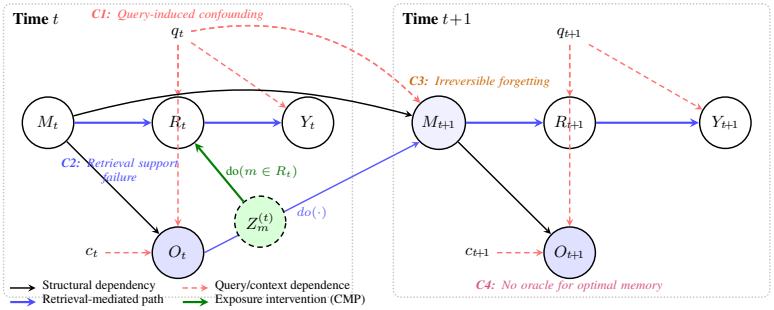

Figure 1: Causal DAG of memory-augmented LLMs across two timesteps.

Figure 1 presents the causal DAG of a memory-augmented LLM in two consecutive time steps. Black solid edges denote structural equations; red dashed edges denote confounding paths through the query q_t and context c_t; thick blue edges denote the mediating retrieval step. The DAG contains two sequential mediators with distinct roles. First, an operation O_t affects future outputs only through the memory state it produces, giving the chain $O_t \to M_{t+1}$ with no direct path from an operation to a later output. This is why memory utility, rather than operation choice, is the estimand in Definition 3. Second, the memory state affects the output only through the retrieved subset, $M_t \to R_t \to Y_t$, since Y_t depends on M_t exclusively through $R_t = f_R(M_t, q_t)$. This second mediation determines whether the estimand is identifiable. Together, the two mediators form the causal chain $O_t \to M_{t+1} \to R_{t+1} \to Y_{t+1}$. The reverse edge $M_t \to O_t$ captures the policy selecting an operation based on current memory state. The graph remains acyclic as forward and reverse edges connect different time steps.

Prior work states the first mediation and leaves the second implicit. *Only the second governs identification*: an operation always induces a memory state, whereas a memory state containing m need not retrieve m. We therefore describe memory utility as *retrieval-mediated* to emphasize that identifiability is determined by retrieval. Each memory operation corresponds to a concrete modification of the memory state. We formalize these as interventions in the SCM.

Definition 2 (Memory Operations as Causal Interventions)**.** *A memory operation $o \in \mathcal{O}$ acts as an intervention $do(M_{t+1} = f_o(M_t))$. $\texttt{store}(m_{new}, M_t)$ is $M_{t+1} = M_t \cup \{m_{new}\}$, $\texttt{forget}(m, M_t)$ is $M_{t+1} = M_t \setminus \{m\}$, $\texttt{consolidate}(m_i, m_j, M_t)$ is $M_{t+1} = (M_t \setminus \{m_i, m_j\}) \cup \{\textsc{Merge}(m_i, m_j)\}$, where $\textsc{Merge}(m_i, m_j)$ produces a single memory preserving content of both inputs m_i and m_j, and $\texttt{noop}(M_t)$ is $M_{t+1} = M_t$.*

The central estimand is the causal effect of an individual memory on downstream task utility.

Definition 3 (Memory Utility)**.** *The utility of memory m at time t, given memory state M_t, is:*

$$U_t(m) \;=\; \mathbb{E}_{q,c}\big[Y_t \mid do(M_t = M_t^{+m}),\, q_t, c_t\big] \;-\; \mathbb{E}_{q,c}\big[Y_t \mid do(M_t = M_t^{-m}),\, q_t, c_t\big], \tag{1}$$

where $M_t^{+m} = M_t \cup \{m\}$ and $M_t^{-m} = M_t \setminus \{m\}$ denote the memory states (not operations) with and without m, respectively, and the expectation marginalizes over the query q and context c.

$U_t(m)$ underlies every memory decision: retaining m if $U_t(m) > 0$. Section 2.3 shows that $U_t(m)$ factorizes through retrieval into a conditional utility $U_R(m)$, and that the factorization collapses when identification fails. For consolidation, the additive criterion $U_t(m_i) + U_t(m_j)$ ignores interactions between memories. We therefore define consolidation directly through the joint intervention:

$$\Delta_{\text{consol}}(m_i, m_j) = \mathbb{E}[Y \mid \text{do}(M_t = (M_t \setminus \{m_i, m_j\}) \cup \{\textsc{Merge}(m_i, m_j)\})] - \mathbb{E}[Y \mid \text{do}(M_t)]\,, \tag{2}$$

and consolidate when $\Delta_{\text{consol}}(m_i, m_j) > 0$. Because both terms are expectations over whole memory states, any interaction between m_i and m_j is already inside them, so no additivity is assumed. We define this criterion so that the operation space is complete, but we do not claim a policy for it. It faces the same identification condition as $U_t(m)$, and Section 3.3 shows that neither can drive a policy without aggregating over queries in a way we find unavailable.

2.2 CORE CHALLENGES

The causal formulation reveals four fundamental challenges, annotated in Figure 1:

C1 Confounded effect estimation. The query q_t simultaneously influences the memory operation O_t, the retrieved set R_t, and the outcome Y_t (red dashed edges in Figure 1), confounding naive comparisons of memory configurations. Without adjustment, performance differences are incorrectly attributed to memory content instead of query difficulty or user intent.

C2 Support failure through the mediator. The intervention do(M_t) reaches Y_t only through R_t (thick blue edge in Figure 1). A memory the retriever never surfaces contributes to no output, so M_t^{+m} and M_t^{-m} induce identical retrieved sets, and the two interventional arms become the same experiment. The resulting contrast collapses toward zero, making an unretrieved memory indistinguishable from one with genuinely zero utility. Only the latter justifies `forget`, so a policy cannot distinguish safe from destructive eviction.

C3 Irreversible operations. `forget` permanently removes a memory, preventing any future re-estimation of its utility. By contrast, `noop` is reversible: a memory retained in error can be evicted later. Decision rules that treat the two errors symmetrically are therefore misspecified.

C4 No ground-truth oracle. The optimal memory state is never observed. Unlike supervised learning, there is no target memory configuration to train against; the value of each memory operation must be inferred solely from its downstream effects.

2.3 RETRIEVAL-MEDIATED IDENTIFICATION

Estimating memory utility requires two conditions. *Exchangeability*: timesteps where m is present must be comparable to those where it is absent, so outcome differences are attributable to m rather than to the circumstances under which it appears. *Overlap*: both cases must occur, since an effect cannot be estimated from one alone. Exchangeability is a property of the assignment mechanism, whereas overlap is a property of its support. Retrieval mediation of Section 2.1 determines where overlap must hold, and it is precisely there that it fails. Since R_t mediates the effect of memory on the output, memory utility factorizes into the probability of retrieval and utility conditional on retrieval.

Proposition 1 (Retrieval-Mediated Factorization)**.** *Let do$(m \in R)$ denote inserting m into the retrieved set R, displacing its lowest-ranked memory, and do$(m \notin R)$ denote removing m and filling its slot with the highest-ranked memory not already in R, so that $|R|$ remains fixed across both arms. Both are dynamic interventions: the displaced or substituted memory depends on the realized ranking at (M_t, q_t). Under Definition 1, for any m and any (q, c),*

$$\begin{aligned} &U(m \mid q, c) = P(m \in R \mid q, c) \cdot U_R(m \mid q, c), \quad U(m) = \mathbb{E}_{q,c}\big[\,U(m \mid q, c)\,\big], \\ &U_R(m \mid q, c) \triangleq \mathbb{E}\big[Y \mid do(m \in R),\, q, c\big] - \mathbb{E}\big[Y \mid do(m \notin R),\, q, c\big]. \end{aligned} \tag{3}$$

Proposition 1 factorizes memory utility into two terms: *retrievability*, $P(m \in R \mid q, c)$, the probability that m is retrieved for query q; and *conditional utility*, $U_R(m \mid q, c)$, the change in expected outcome caused by including m in the retrieved context while holding the context size fixed. Their product is the contribution of m for (q, c), and $U(m)$ averages it over the query distribution. The factorization also identifies where overlap must hold. Because its first factor is a retrieval probability, overlap is required in the retrieval variable, not in the operation variable.

Assumption 1 (Retrieval Positivity)**.** *For all $m \in M_t$ and all (q, c) in the support:*

$$P\big(m \in R_t \mid q_t = q,\, c_t = c\big) \;>\; 0. \tag{4}$$

Assumption 1 is the only assumption CMP requires. When it fails, the retrieval probability in Eq. 3 is zero, causing the product to collapse to zero regardless of the conditional utility. Section 3 constructs a design in which this assumption holds by construction, with $P(m \in R_t \mid q, c)$ fixed and known. *By contrast, prior methods that estimate memory utility from logged interactions additionally require two conditions (stated as Conditions 1 and 2 in Appendix C.4).*

Theorem 1 (Non-Identifiability under Retrieval Support Failure)**.** *Fix (q, c) and suppose $P(m \in R \mid q, c) = 0$. Then $f_R(M^{+m}, q) = f_R(M^{-m}, q)$, hence $Y(M^{+m}) = Y(M^{-m})$ almost surely, and $U(m \mid q, c) = 0$ irrespective of $U_R(m \mid q, c)$. Consequently, $U_R(m \mid q, c)$ is not identified by any design that holds M_{-m} fixed while randomizing only whether m is included.*

Theorem 1 separates two failure modes that are observationally indistinguishable: a memory with $U_R(m) = 0$ and one with $P(m \in R \mid q, c) = 0$ produce the same observed contrast. An estimate near zero therefore admits two interpretations: the memory has no utility, or the design never tests whether it has utility. Only the former justifies `forget`. The same asymmetry applies to `store`: a memory never stored generates no future observations, so its utility can never be estimated.

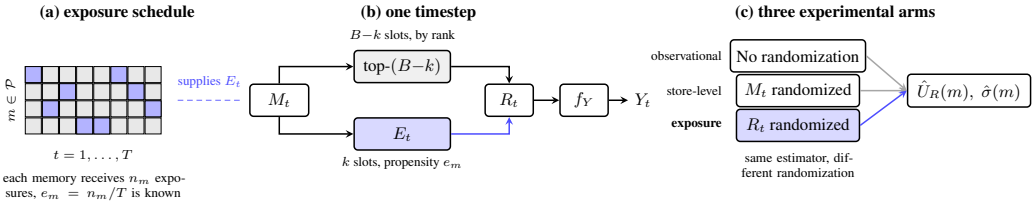

Figure 2: Overview of CMP. (a) Balanced exposure assigns the Tk exposure slots evenly across the exposure pool, fixing each memory's propensity by design. (b) At each timestep, the context consists of the top-$(B-k)$ retrieved memories together with k randomly exposed memories. (c) CMP compares three designs that differ only in the randomized variable while using the same estimator.

2.4 Problem Statement

Given logged interactions with a memory-augmented LLM, estimate the conditional utility $U_R(m \mid q, c)$ for every memory a policy may act on. Theorem 1 shows that no design randomizing only the memory state can do so when $P(m \in R \mid q, c) = 0$, regardless of sample size or the extent of store-level randomization. *The problem is therefore to construct a design that guarantees Assumption 1 over a specified set of memories and estimates U_R from the resulting data.*

3 Causal Memory Policy

We develop the **Causal Memory Policy (CMP)**, which operationalizes retrieval-mediated identification through three components (Figure 2). Component (i) introduces a randomized retrieval exposure design in which a fixed number of context slots are withheld from the ranker and filled by memories sampled with known probability. This guarantees Assumption 1 by construction, makes the propensity a design parameter, and removes confounding on the randomized arm. Component (ii) estimates memory utility by self-normalized inverse-propensity weighting, while including observational and store-level randomized estimators as comparators to isolate the contribution of the design from that of the estimator. Component (iii) derives the Bayes-optimal decision rule under the asymmetry between reversible and irreversible operations, and shows that every aggregation over queries available to a retention policy is either attenuated or degenerate.

These components address the challenges of Section 2.2: Component (i) resolves **C1** and **C2** by randomizing the retrieval mediator; Component (ii) addresses **C4** by recovering utility solely from downstream outcomes and separating design effects from estimation effects; and Component (iii) addresses **C3** by deriving the appropriate decision rule and identifying the remaining obstruction after utility is identified. Theoretical guarantees for each CMP component are established in Appendix C.

3.1 Randomized Retrieval Exposure

Theorem 1 shows that randomizing only the memory state cannot identify $U_R(m)$ for memories the retriever never surfaces. The obstruction is that retrieval is deterministic: given (M_t, q_t), the ranker returns a single retrieved set, so a memory outside it is never observed in context and its two potential outcomes are never both realized. Store-level randomization cannot resolve this because retrieval is applied after the store is perturbed. We therefore randomize the mediator itself by reserving a fixed number of retrieval slots for randomized exposure rather than ranking. Specifically, At each timestep, the B context slots are filled by two mechanisms:

$$\textbf{Exposure design:}\quad R_t = \text{top-}(B-k)(M_t, q_t) \cup E_t, \qquad E_t \subset \mathcal{P},\ |E_t| = k, \tag{5}$$

where B is the context budget and $k \leq B$ is the number of slots reserved for exposure. The remaining $B - k$ slots are filled by the deterministic retriever rule of Definition 1, while E_t is sampled from an exposure pool $\mathcal{P} \subseteq M_t$. The extremes recover two familiar designs: $k = 0$ gives ordinary retrieval, whereas $k = B$ removes the ranker entirely.

Across T timesteps, the Tk exposure slots are allocated as evenly as possible across $\mathcal{P}$ and randomly assigned to timesteps. Thus each memory $m \in \mathcal{P}$ receives exactly $n_m \in \{\lfloor Tk/|\mathcal{P}| \rfloor, \lceil Tk/|\mathcal{P}| \rceil\}$ exposures, and its propensity $e_m = n_m/T$ is known exactly. The propensities are fixed by design rather than estimated, so Assumption 1 holds by construction with $\min_m e_m > 0$.

Balanced assignment is essential. Independent inclusion with the same expected count gives $n_m \sim \text{Binomial}(T, k/|\mathcal{P}|)$, which leaves many memories underexposed. For example, at $T = 90$, $k = 2$, and $|\mathcal{P}| = 32$, the expected count is 5.6, yet roughly one third of the pool receives fewer than

five exposures. Balanced assignment instead guarantees every memory exactly five or six exposures. Two properties of the design are worth noting. First, an exposure draw may select a memory already present in the deterministic retrieval, so the realized context contains at most B distinct memories; we report the realized context size in Section 4. Second, each memory receives $Tk/|\mathcal{P}|$ exposures, so the horizon required to identify the entire pool grows linearly with $|\mathcal{P}|/k$ (see Appendix C in detail).

Exposure pool $\mathcal{P}$ determines which memories can be identified. In our experiments, $\mathcal{P}$ contains the memories relevant to the evaluated queries together with the highest-ranked and a uniform sample of the remaining store, while retrieval continues over the full store. The experiments therefore establish that randomizing the retrieval mediator restores identification, but not that a deployable pool selector captures the memories that matter. Section 4 evaluates three practical selection rules and finds their recall near zero, identifying pool selection as the remaining open problem.

3.2 Utility Estimation Under Known Propensity

We write $Z_m^{(t)} = \mathbf{1}_{m \in R_t}$ for the exposure retrieval indicator with $\mathcal{T}_m^+ = \{t : Z_m^{(t)} = 1\}$, $\mathcal{T}_m^- = \{t : Z_m^{(t)} = 0\}$. Since e_m is known, conditional utility is estimated by the self-normalized (Hájek) inverse-propensity weighting estimator (Imbens and Rubin, 2015):

$$\hat{U}_R(m) \;=\; \frac{\sum_t Z_m^{(t)} Y_t / e_m}{\sum_t Z_m^{(t)} / e_m} \;-\; \frac{\sum_t (1 - Z_m^{(t)}) Y_t / (1 - e_m)}{\sum_t (1 - Z_m^{(t)}) / (1 - e_m)} \;=\; \frac{1}{n_m} \sum_{t \in \mathcal{T}_m^+} Y_t - \frac{1}{T - n_m} \sum_{t \in \mathcal{T}_m^-} Y_t. \tag{6}$$

The second equality follows from balanced assignment, which fixes n_m in advance. Consequently, the weight sums are constants, the estimator reduces to a difference of arm means, and its variance, $\sigma_+^2/n_m + \sigma_-^2/(T - n_m)$, is exact rather than conditional on realized arm sizes. Equation 6 provides both the point estimate and, via Proposition 2, the standard error $\hat{\sigma}(m)$ used in Section 3.3. Self-normalization removes the propensity-dependent scaling of ordinary inverse-propensity weighting, making estimates directly comparable across memories.

Proposition 2 (Hájek Estimator Under Balanced Exposure)**.** *Under the exposure design of Eq. 5 with balanced assignment,* $\mathbb{E}[\hat{U}_R(m)] = U_R(m)$ *for every* $m \in \mathcal{P}$*, and* $\mathrm{Var}(\hat{U}_R(m)) = \sigma_+^2/n_m + \sigma_-^2/(T - n_m)$ *exactly, with* n_m *fixed by the design.*

We compare CMP with two estimators targeting the same estimand under alternative designs. The observational comparator estimates propensity and outcome models from logged interactions.

$$\begin{aligned} &\textbf{Propensity: } \hat{p}\big(m{\in}M \mid M^{-m}, q, c\big) = \mathrm{softmax}\big(g_\psi(M^{-m}, q, c)\big), \\ &\textbf{Outcome: } \hat{\mu}\big(a, M^{-m}, q, c\big) = h_\xi\big(\phi(a, M^{-m}, q, c)\big), \end{aligned} \tag{7}$$

where $a \in \{0, 1\}$ is the inclusion status, g_ψ is trained by maximum likelihood on logged inclusions, h_ξ is a regression head, and ψ concatenates the inclusion indicator with mean-pooled embeddings of the remaining memories, the query, and the context. Both models condition on M^{-m}, which is the adjustment set held fixed in Definition 3. They are combined using the augmented inverse-propensity weighted (AIPW) estimator (Robins et al., 1994; Bang and Robins, 2005) over the n logged timesteps:

$$\begin{aligned} \hat{U}_{\text{obs}}(m) = \frac{1}{n} \sum_{i=1}^{n} \Big[&\big(\hat{\mu}(1, M_i^{-m}, q_i, c_i) - \hat{\mu}(0, M_i^{-m}, q_i, c_i)\big) \\ &+ \frac{\mathbf{1}_{m \in M_i}}{\hat{p}(m{\in} \mid M_i^{-m}, q_i, c_i)} \big(Y_i - \hat{\mu}(1, M_i^{-m}, q_i, c_i)\big) - \frac{\mathbf{1}_{m \notin M_i}}{1 - \hat{p}(m{\in} \mid M_i^{-m}, q_i, c_i)} \big(Y_i - \hat{\mu}(0, M_i^{-m}, q_i, c_i)\big) \Big] \end{aligned} \tag{8}$$

Proposition 3 establishes consistency for $U(m)$ when either model is correctly specified.

Proposition 3 (Double Robustness of the Observational Comparator)**.** $\hat{U}_{\text{obs}}(m) \xrightarrow{p} U(m)$ *if either* $\hat{p}$ *or* $\hat{\mu}$ *is correctly specified.*

The *store-level randomized* comparator perturbs store membership, retaining each memory independently with probability $1 - p_{\text{flip}}$, and estimates $\hat{U}_{\text{exp}}(m)$ by a difference of arm means over timesteps where m is present or absent under the randomization.

Proposition 4 (Unbiasedness of Store-Level Arm)**.** *Under independent store-level perturbation with rate* p_{flip}*,* $\hat{U}_{\text{exp}}(m)$ *is an unbiased utility* $\mathbb{E}_{M^{-m}}[\mathbb{E}[Y \mid m \in M, M^{-m}] - \mathbb{E}[Y \mid m \notin M, M^{-m}]]$*, regardless of confounding, with variance* $\sigma_+^2/|\mathcal{T}_m^+| + \sigma_-^2/|\mathcal{T}_m^-|$*.*

The three estimators differ only in the variable they randomize: none (observational), memory state M_t (store-level), or retrieved set R_t (exposure). By Theorem 1, only the exposure design retains support for memories the retriever never surfaces, while all three coincide whenever support is available. This isolates the contribution of the design rather than the estimator, and Section 4 verifies this by restricting the comparison to memories with support under the store-level design.

Disagreement diagnostic: We define $\Gamma(m) = |\hat{U}_{\text{obs}}(m) - \hat{U}_{\text{exp}}(m)|/(\hat{\sigma}^2_{\text{obs}}(m) + \hat{\sigma}^2_{\text{exp}}(m))^{1/2}$, the standardized disagreement between the observational and store-level arms. Its null is composite: rejection indicates failure of at least one of exchangeability, negligible interaction, or correct model specification. We therefore use Γ only as a diagnostic, leaving operational decisions to the uncertainty of the CMP estimate itself (Section 3.3).

3.3 From Estimates to Operations

An estimate becomes an operation through a decision rule. Challenge **C3** makes this rule asymmetric: `forget` permanently removes both the memory and any opportunity to re-estimate its utility, whereas `noop` is reversible at the next capacity decision. Let H denote the number of future queries over which a retained memory would still contribute, and τ the number of queries until the decision is revisited, so that $\lambda = H/\tau \geq 1$ is the *irreversibility ratio*. Assuming the posterior $U_R(m) \mid \hat{U}_R, \hat{\sigma} \sim \mathcal{N}(\hat{U}_R(m), \hat{\sigma}^2(m))$ with losses $L_{\texttt{forget}} = H \max(0, U_R)$ and $L_{\texttt{noop}} = \tau \max(0, -U_R)$.

Theorem 2 (One-Sided Abstention is Bayes-Optimal)**.** *Let $\lambda = H/\tau \geq 1$ and $z = \hat{U}_R(m)/\hat{\sigma}(m)$. The Bayes-optimal rule is $\texttt{forget}(m) \iff g_\lambda(z) < 0$, where g_λ is strictly increasing with a unique root $z^\star(\lambda)$ satisfying $z^\star(\lambda) < 0$ for $\lambda > 1$, $z^\star(1) = 0$, and $z^\star(\lambda) \to -\infty$ as $\lambda \to \infty$.*

$$\texttt{forget}(m) \iff \hat{U}_R(m) < z^\star(\lambda)\hat{\sigma}(m), \qquad g_\lambda(z) = \lambda z\Phi(z) + z\Phi(-z) + (\lambda - 1)\phi(z). \tag{9}$$

Here ϕ and Φ denote the standard normal density and distribution functions. Since g_λ is strictly increasing, the threshold $z^\star(\lambda)$ is unique. Thus only sufficiently negative estimates trigger eviction. A two-sided rule, $|\hat{U}_R(m)| > \delta\hat{\sigma}(m)$, is strictly dominated because it can evict confidently useful memories. When a memory lacks support, $\hat{\sigma} \to \infty$, so $z \to 0$ and the rule abstains.

Eq. 9 is nonetheless insufficient to specify a retention policy. The rule requires a scalar $\hat{U}_R(m)$, whereas Proposition 1 makes utility query-dependent. Since retention is decided before the next query is known, a policy must aggregate over $\mathcal{Q}$, but every admissible aggregation is degenerate. Averaging attenuates utility by the fraction of queries requiring the memory (one in thirty on LoCoMo), while $\max_q \hat{U}_R(m \mid q)$ eliminates attenuation but is positive for nearly every memory. Aggregations weighted by similarity to the future query avoid both problems but require information unavailable at decision time. Section 4 evaluates all three and none improves on retaining everything. The same reasoning applies to the other operations: `store` has the opposite asymmetry, committing unless the estimate is confidently negative, while `consolidate` inherits the same asymmetry as `forget` because MERGE destroys both inputs. Both face the same aggregation problem.

We therefore present CMP as an identified measurement of per-query memory utility rather than a retention policy. On held-out LoCoMo, it separates required from non-required memories for the query on which it is estimated with an AUC of 0.784, compared with 0.581 when the same estimator is pooled across queries. The loss in performance shows that the limitation lies not in estimation but in the estimand: a single scalar per memory cannot represent a quantity that varies with the query.

3.4 CMP Algorithm

CMP operates in three phases, summarized in Algorithms in Appendix D.

(1) **Phase 1: warm-up.** Collect interaction data under the default policy with the exposure design in Eq. 5. At each timestep, record $(M_t, q_t, c_t, R_t, Z^{(t)}, Y_t)$, including the realized exposure indicators from which propensities are obtained.

(2) **Phase 2: model fitting.** Estimate $\hat{U}_R(m)$ and $\hat{\sigma}(m)$ for each $m \in \mathcal{P}$ via Eq. 6, fit propensity and outcome models in Eq. 7, and compute $\Gamma(m)$ to audit support under the observational comparator.

(3) **Phase 3: online execution.** Continue serving responses while maintaining $\hat{U}_R(m \mid q)$ and $\hat{\sigma}(m \mid q)$ online from new interactions. These estimates are consumed by downstream policies: Eq. 9 gives the decision rule for a single query, Section 3.3 shows why no aggregation over $\mathcal{Q}$ yields a retention policy, and Section 4 evaluates three candidate aggregations.

Table 1: Retrieval support failure. $P = 0$ denotes the fraction of required memories never surfaced by the retriever; **Store** is the mean number of memories available at query time. †Mem0 rewrites source turns, requiring annotated evidence to be matched to stored memories. We report the least-conditioned of three matching rules; see Appendix F.4 for details.

Setting	Required	$P = 0$	Store
LongMemEval (indep.)	113	**54.0%**	487
LongMemEval (all)	170	61.2%	487
LoCoMo	359	**66.9%**	588
Mem0 2.0.18†	109	28.4%	291

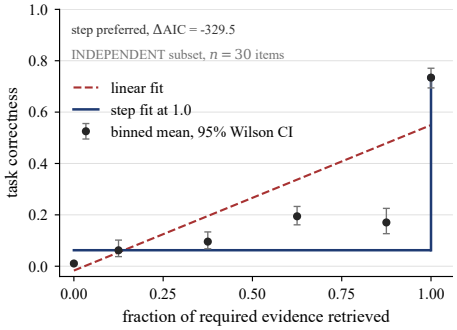

Figure 3: Task-correctness against the fraction of required evidence retrieved, LongMemEval independent subset.

4 Experiments

Our experiments answer four questions. **Q1**: how often does Assumption 1 fail in practice? **Q2**: can better retrieval repair it, or is randomization necessary? **Q3**: does randomizing the mediator restore identification, and is the gain due to the design or the estimator? **Q4**: does identified utility support a retention policy? Section 4.1 answers Q1, Q2; Section 4.2 answers Q3; and Section 4.3 answers Q4.

Setup. We evaluate on LongMemEval (Wu et al., 2024), LoCoMo (Maharana et al., 2024), HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and Mem0 (Chhikara et al., 2025). LongMemEval is our primary benchmark, LoCoMo supports evaluation under a query distribution, HotpotQA and MuSiQue generalize to multi-hop retrieval, and Mem0 evaluates a deployed memory system. We use `gemini-3.7-flash` and report token-level F1 and exact match. We do not report LLM-as-judge accuracy because of the reproducibility concerns discussed in Appendix I.1. More dataset details are in Appendix E.1. By default, CMP uses $B = 6$ context slots and $|\mathcal{P}| \approx 32$–36. Experiment-specific values of k and T, together with complete protocols, are given in Appendix E–H.

4.1 Empirical Identification Failure

Table 1 quantifies identification failure by measuring the fraction of required memories with $P(m \in R \mid q, c) = 0$. On LongMemEval, retrieval support is strongly bimodal: 61 of 113 are never retrieved and a further 6 almost never are, 37 are retrieved on more than 80% of draws, and only 9 fall between 0.1 and 0.8. Consequently, store-level randomization leaves the retrieved set unchanged on 52.2% of interventions on M_t, producing 940 distinct retrieved contexts compared with 3,939 under randomized exposure at the same costs.

A larger retrieval budget does not resolve the failure. In Mem0, recovering 50% required memories never surfaced by the retriever requires expanding the retrieval window to 138 memories, and recovering 90% requires 286, effectively abandoning retrieval. Mem0 also exposes a separate write-time limitation: across 11,651 memories over 55 histories, every operation is an addition, with no updates or deletions, so contradictory facts accumulate. This limitation concerns the write policy. See more analysis in Appendix F.4. The misses are disproportionately costly. Figure 3 shows that partial retrieval contributes little: task correctness remains near 0.2 until every required memory is retrieved, then jumps to 0.73. The reason is that these tasks require all supporting memories jointly; retrieving only a subset typically leaves the model unable to produce the correct answer.

4.2 Exposure Restores Identification

Table 2 compares store-level randomization and CMP under identical evaluation budgets. We report discrimination (AUC between required and non-required memories), nomination lift (the relative rate at which required memories are nominated for eviction), and gold loss (required memories destroyed per 100 reclaimed slots under Eq. 9 with $\lambda = 10$).

Table 2: Estimation and decision quality on two designs (only differ in randomized variable) and LongMemEval independent subset.

Design	AUC	Lift	Gold loss
Store-level rand.	0.542	0.97	10.9
Exposure (CMP)	**0.664**	**0.68**	**5.2**

CMP consistently outperforms store-level randomization, improving discrimination from 0.542 to 0.664 and halving gold loss from 10.9 to 5.2. Moreover, store-level estimates nominate required

memories at nearly random (lift $= 0.97$), whereas CMP reduces this to 0.68, making required memories substantially less likely to be selected for eviction.

Table 3 locates the gap. Stratifying by retrieval support, the two designs are indistinguishable where the store-level design has both arms, and the entire advantage sits in the stratum where it has none. The limitation is therefore the design, not the estimator: store-level estimation is not less accurate, it is undefined on 54% of required memories.

Appendix G.2 compares five memory policies on the same axis and finds that heuristic rules nominate required memories at 0.00–0.11 times the chance rate, so the identification failure binds on policies that rank by estimated utility and not on those that do not. Appendix H.1 provides detailed support-stratified analyses and per-memory case studies (one memory identified only by CMP, one identified by both designs, and one identified by neither).

Table 3: Discrimination by retrieval support, LongMemEval independent subset. Median $|t|$ is the store-level estimate's significance.

Support	n	**AUC gap**	**Median $\lvert t\rvert$**
$P = 0$	61	**exposure only**	0.83
$0 < P < 0.8$	15	$+0.09$	1.7
$P \geq 0.8$	37	$+0.003$	3.66

4.3 Ablation

Table 4 separates the design from the decision rule. The two are complementary: holding the decision rule fixed, randomized exposure reduces gold loss from 15 to 6, while holding the design fixed, abstention reduces it from 27 to 6. Abstention can act on identified memories, whereas randomized exposure restores identification for memories lacking retrieval support. Two-sided abstention performs worse than no abstention, destroying 14.1 versus 9.0 required memories per 100 reclaimed. It admits confidently useful memories into the eviction pool, the behavior ruled out by Theorem 2.

Extending the design symmetrically also fails. Adding a forced-exclusion slot closes the remaining support gap but yields only a 0.007 AUC gain while increasing destroyed memories from 6 to 10. Three of the four additional losses are memories the one-sided design could not estimate. Identification is not free: exposing more memories also exposes them to erroneous eviction.

Table 4: Component ablation on LongMemEval. Gold loss is reported per 100 reclaimed slots.

Variant	**AUC**	**Destroyed**	**Gold loss**
CMP (full)	$\mathbf{0.664}$	**6**	**5.2**
– one-sided abstention	0.664	27	9.0
– exposure design	0.542	15	10.9
– both	0.542	35	11.7
two-sided abstention	0.664	22	14.1
two-sided exposure	0.671	10	6.6

A third variant isolates the effect of the estimand. Averaging utility across queries dilutes the signal because most memories are relevant to a small fraction of queries ($1/30$ for 77 of 109 gold memories on LoCoMo). Estimating query-specific utility improves discrimination from 0.581 to $\mathbf{0.784}$ on the required-memory subset; the pooled estimand over all pool members is 0.551 (Appendix H.2) and increases the utility gap between required and non-required memories from 0.038 to 0.300, confirming that the remaining limitation is in the estimand.

Identified utility does not determine retention. Per-query utility is accurate for the query on which it is estimated but does not transfer across queries. On a disjoint half of each conversation, no query-independent aggregation achieves correlation above $r = 0.10$ with realized contribution, whereas ground-truth has $r = 0.286$ on the same pairs. This is not a power issue: 99.3% of required $(\text{memory}, \text{query})$ cells are identified with five draws per arm. The limitation is structural. A retention policy cannot condition on a future query, while query-independent aggregations either dilute the signal or, in the case of $\max_q \hat{U}_R(m \mid q, c)$, retain essentially every memory. We therefore make no retention claim. CMP provides an identified measurement of per-query memory utility, and our results show that this quantity alone is insufficient to determine what to retain. ppendix H.8 provides additional analyses of retrieval failure, retrieval scaling, and exposure-budget tradeoffs.

5 Conclusion

We presented CMP, a causal framework that treats memory utility as a retrieval-mediated effect. Under deterministic retrieval, many memories are never observed in both treatment arms, leaving their utility unidentifiable from logged interactions. This affects half the required memories on LongMemEval and persists in a deployed memory system. CMP restores identification by randomizing retrieval exposure with known propensity. We prove identification, unbiased estimation, and the optimal decision rule under irreversible operations.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, including editing and polishing prose, and for code assistance in implementing the experimental harness and analysis scripts. We also used generative AI tools as the generation and embedding backends under study; these are the object of the experiments, and their versions are pinned and reported in Appendix E.3. We did not use generative AI tools to generate research ideas, to produce experimental results, or to write the theoretical claims or their proofs. All AI-assisted code was reviewed by the authors and its outputs verified against the recorded artifacts; several measurement errors introduced during implementation were found by this process and are reported in Appendix I.3. We take responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their assumptions in Section 2.3 and proved in Appendix C; Assumption 1 is the only condition our method requires, and the conditions needed by the comparator estimators are stated separately in Appendix C.4. Benchmark construction, the item selection and its exclusions, model versions, and the exposure design parameters for every experiment are given in Appendix E, and the measurement protocol for each reported quantity in Appendix F. Appendix D specifies the exposure allocation and estimation procedures. Appendix I records the discrepancies we found in our own apparatus, including a metadata leak in the repository's default serving path that does not affect the reported arms, and the instability of the LLM-judge protocol, which is why we report only deterministic metrics. Assignment permutations, item selections, and pool constructions are seeded and the seeds are stored with the results they produced. Code, per-item outputs, and scores are available at `https://anonymous.4open.science/r/cmp-release-D0C3/`.

ETHICS STATEMENT

This work uses only publicly available benchmarks and involves no human subjects, no personally identifying information, and no data collection. The conversational benchmarks we evaluate contain synthetic or released dialogue, and we introduce no new data. Our method randomizes which stored memories are placed in a model's context, which changes the information available to the system on a fraction of interactions; in a deployed setting this could degrade individual responses, and Section 4 reports the measured cost. Deployments should weigh that cost against the value of the estimates the design produces. We report a failure mode in a third-party memory system, and we do so to characterize a condition that applies to the class of systems.

REFERENCES

Amin Abolghasemi, Leif Azzopardi, Seyyed Hadi Hashemi, Maarten de Rijke, and Suzan Verberne. Evaluation of attribution bias in generator-aware retrieval-augmented large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21105–21124, 2025.

Michael C Anderson, Robert A Bjork, and Elizabeth L Bjork. Remembering can cause forgetting: retrieval dynamics in long-term memory. *Journal of experimental psychology: Learning, memory, and cognition*, 20(5):1063, 1994.

Heejung Bang and James M Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

Peter J Bickel, Chris AJ Klaassen, Peter J Bickel, Ya'acov Ritov, J Klaassen, Jon A Wellner, and YA'Acov Ritov. *Efficient and adaptive estimation for semiparametric models*, volume 4. Johns Hopkins University Press Baltimore, 1993.

Robert A Bjork, Elizabeth L Bjork, et al. A new theory of disuse and an old theory of stimulus fluctuation. *From learning processes to cognitive processes: Essays in honor of William K. Estes*, 2:35–67, 1992.

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. Stale: Can llm agents know when their memories are no longer valid?, 2026. *URL https://arxiv. org/abs/2605.06527*.

Cheng Chen, Maria D Molina, Mengqi Liao, and Eugene Cho Snyder. Relational gains, privacy strains: Exploring users' perceptions and experiences with chatgpt's memory feature. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv*, 2025.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Mądry. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37: 95764–95807, 2024.

Nima Dehghani and Michael Levin. Bio-inspired ai: Integrating biological complexity into artificial intelligence. *arXiv preprint arXiv:2411.15243*, 2024.

Zixiang Ji Di Wu, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. Longmemeval-v2: Evaluating long-term agent memory toward experienced colleagues, 2026a. *URL https://arxiv. org/abs/2605.12493*.

Ijeoma Noella Ezeji, Matthew Adigun, and Kayode Oki. Towards enhancing computational complexity in explainable decision support system: Exploring bio-inspired approaches. *Procedia Computer Science*, 258:430–442, 2025.

Dongyang Fan, Tyler J Rotello, and Sai Praneeth Karimireddy. Do data valuations make good data prices? *arXiv preprint arXiv:2504.05563*, 2025.

Haowen Gao, Haoran Chen, Can Wang, Shasha Guo, Liang Pang, Zhaoyang Liu, Huawei Shen, and Xueqi Cheng. Skillaudit: Ground-truth-free skill evolution via paired trajectory auditing. *arXiv preprint arXiv:2606.14239*, 2026.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.

Yongkang Hu, Xiangbo Yang, Yunhan Xu, and Jingpeng Sun. Causal-aware reliability assessment of single-channel eeg for transformer-based sleep staging. *Frontiers in Neuroscience*, 19:1670124, 2025.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, volume 2026, pages 156259–156291, 2026.

Ziheng Huang, Sebastian Gutierrez, Hemanth Kamana, and Stephen MacNeil. Memory sandbox: Transparent and interactive memory management for conversational agents. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3, 2023.

Guido W Imbens and Donald B Rubin. Causal inference in statistics, social, and biomedical sciences. *New York*, 517, 2015.

Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 383–390, 2019.

Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 781–789, 2017.

Taesan Kim, Hyeongjun Yun, Jaegul Choo, and Chung Park. Don't let bandit feedback pull continual llm-recommender updates off target. *arXiv preprint arXiv:2605.18899*, 2026.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7): 512–534, 2016.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 2020.

Ruoran Li, Xinghua Zhang, Haiyang Yu, Shitong Duan, Xiang Li, Wenxin Xiang, Chonghua Liao, Xudong Guo, Yongbin Li, and Jinli Suo. Mempo: Self-memory policy optimization for long-horizon agents. *arXiv preprint arXiv:2603.00680*, 2026.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pages 463–473, 2020.

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

James L McClelland, Bruce L McNaughton, and Andrew K Lampinen. Integration of new information in memory: new insights from a complementary learning systems perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1799), 2020.

Ikhtiyor Nematov, Tarik Kalai, Elizaveta Kuzmenko, Gabriele Fugagnoli, Dimitris Sacharidis, Katja Hose, and Tomer Sagi. Source attribution in retrieval-augmented generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 317–332. Springer, 2025.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun 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*, pages 1–22, 2023.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6037–6053, 2024.

Vishal Raman, Abhijith Ragav, et al. Remi: A novel causal schema memory architecture for personalized lifestyle recommendation agents. *arXiv preprint arXiv:2509.06269*, 2025.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 1994.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Max Schemmer, Niklas Kuehl, Carina Benz, Andrea Bartos, and Gerhard Satzger. Appropriate reliance on ai advice: Conceptualization and the effect of explanations. In *Proceedings of the 28th international conference on intelligent user interfaces*, pages 410–422, 2023.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Baris Simsek. When to forget: A memory governance primitive. *arXiv preprint arXiv:2604.12007*, 2026.

Saksham Sahai Srivastava. Causal intervention-based memory selection for long-horizon llm agents. *arXiv preprint arXiv:2605.17641*, 2026.

Robert Stickgold. Sleep-dependent memory consolidation. *Nature*, 437(7063):1272–1278, 2005.

Ana I Torre-Bastida, Josu Diaz-de Arcaya, Eneko Osaba, Khan Muhammad, David Camacho, and Javier Del Ser. Bio-inspired computation for big data fusion, storage, processing, learning and visualization: state of the art and future directions. *Neural Computing and Applications*, 37(28): 23097–23127, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Endel Tulving and Zena Pearlstone. Availability versus accessibility of information in memory for words. *Journal of verbal learning and verbal behavior*, 5(4):381–391, 1966.

Endel Tulving and Donald M Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352, 1973.

Bingkai Wang, Fan Li, and Rui Wang. Handling incomplete outcomes and covariates in cluster-randomized trials: doubly-robust estimation, efficiency considerations, and sensitivity analysis. *arXiv preprint arXiv:2401.11278*, 2024a.

Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, et al. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*, 2024b.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023.

Beining Wu, Zihao Ding, Jun Huang, and Yanxiao Zhao. Forget to improve: On-device llm-agent continual learning via budget-curated memory. *arXiv preprint arXiv:2606.25115*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Empirical methods in natural language processing*, 2018.

Zikun Ye and Hema Yoganarasimhan. Document valuation in llm summaries: A cluster shapley approach. *arXiv preprint arXiv:2505.23842*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *AAAI conference on artificial intelligence*, 2024.

Appendix

A Related Work **17**

B Notation **18**

C Theoretical Guarantees **19**

C.1 Retrieval-Mediated Identification 19

C.2 Estimation Under the Exposure Design 20

C.3 Decision Rules Under Irreversibility 21

C.4 Conditions Required by the Observational Comparator 22

C.5 The Store-Level Randomized Comparator 23

C.6 AIPW Estimator: Consistency, Efficiency, and Robustness 24

D Algorithm Details **25**

E Experimental Setup **27**

E.1 Benchmark Datasets 27

E.2 Retrieval Function Specification 27

E.3 Systems and Implementation 28

E.4 Exposure Design Parameters 28

E.5 Outcome Measurement 28

E.6 Statistical Procedure 29

F Measurement Protocols **29**

F.1 Retrieval Support 29

F.2 Estimation Under Exposure 30

F.3 Comparator Arms 30

F.4 Deployed-System Replay 30

F.5 Cross-Query Transfer 31

F.6 Selector and Eviction Rules 31

G Baselines and Comparisons **32**

G.1 Memory Policies Implemented 32

G.2 Reported Comparisons 32

G.3 Omitted Comparisons 33

G.4 Retrieval Selectors 33

H Additional Results **33**

H.1 Worked Examples 33

H.2 Identification by Benchmark 34
H.3 Cost of Exposure 35
H.4 Ablation Details 35
H.5 Retrieval Scoring 36
H.6 Cross-Query Transfer 36
H.7 Suspect Items 37
H.8 Extended Analysis 37

I Reproducibility and Known Discrepancies **38**
I.1 Evaluation Instruments 38
I.2 Implementation Discrepancies 38
I.3 Pool Construction Determines the Conclusion 38
I.4 Compute and Cost 39
I.5 Artifacts 39

J Limitations **39**

K Applications of This Work **39**

A Related Work

Memory-augmented LLMs. Retrieval-augmented generation (Lewis et al., 2020) establishes the paradigm of conditioning LLM outputs on passages retrieved from an external store. MemoryBank (Zhong et al., 2024) introduces Ebbinghaus-inspired forgetting curves for long-term retention; generative agents (Park et al., 2023) maintain hierarchical memory with reflection in simulated social environments; and memorizing transformers (Wu et al., 2022) embed persistent key–value memory directly into attention layers. More recent work has augmented LLMs with unbounded long-term stores (Wang et al., 2023), evaluated memory over extended conversations (Maharana et al., 2024), and developed agentic memory management. In particular, A-Mem (Xu et al., 2025) draws on the Zettelkasten method to autonomously organize memories through note construction and link generation. Despite these advances, memory operations are still largely governed by hand-designed mechanisms, e.g., similarity thresholds, recency scores, and capacity constraints, rather than a formal framework for determining when and why a memory operation improves downstream performance.

Estimating per-memory utility. Recent systems estimate the contribution of each memory rather than assuming it. Memory governance based on per-memory success and failure counts (Simsek, 2026) converges to a conditional success probability; paired trajectory auditing (Gao et al., 2026) isolates a candidate's effect by executing with and without it; budget-curated retention (Wu et al., 2026) selects memories under an explicit capacity constraint; and causal memory selection (Srivastava, 2026) scores candidates through controlled interventions on the candidate set. These approaches differ in their estimators and supervision, but share a common dependency: evidence about a memory can be collected only when the retriever surfaces it. Our contribution is orthogonal to the particular utility estimator: the retrieval-support condition we identify is a prerequisite for all of them. Moreover, causal memory selection (Srivastava, 2026) requires access to the ground-truth answer within the selection loop, an additional source of supervision that is generally unavailable at deployment.

Memory policy optimization. MemGPT (Packer et al., 2023) draws on operating-system virtual memory to let LLMs page information between context and external storage, but relies on the LLM's own judgment for management decisions without formal optimization. More recently, MemPO (Li et al., 2026) proposes self-memory policy optimization, framing memory as a learnable action within an RL loop and introducing trajectory-level and memory-level advantage estimation for credit assignment. MemPO achieves strong empirical gains on long-horizon benchmarks by training the agent to autonomously compress and retain information. However, RL-based approaches optimize memory decisions end-to-end without asking whether the effect of an individual memory is identified from the data the system collects. The reward signal may be observed even when the per-memory contrast needed to attribute that reward is not. Our work identifies a prior condition: the effect of a specific memory can be identified from retrieval-conditioned observations only where the retriever provides support for that memory. Where retrieval provides no such support, end-to-end optimization cannot by itself recover the missing per-memory contrast.

Memory consolidation in cognitive science. Complementary learning systems theory (Kumaran et al., 2016; McClelland et al., 2020) posits that biological memory relies on two interacting systems: a fast hippocampal store for episodic details and a slow neocortical store for consolidated abstractions. Sleep-dependent consolidation selectively strengthens, integrates, and prunes memories according to their future relevance (Stickgold, 2005). Prior AI work (Torre-Bastida et al., 2025; Dehghani and Levin, 2024; Ezeji et al., 2025) has drawn bio-inspired heuristics from these theories, but does not formally characterize when consolidation improves downstream performance. We instead formulate consolidation as a joint intervention (Eq. 2), making explicit the causal quantity that a consolidation decision would require. This quantity remains subject to the same identification condition as the single-memory estimand; accordingly, our criterion characterizes the effect of consolidation rather than prescribing a consolidation policy.

Causal inference in machine learning. Structural causal models (SCMs) (Pearl, 2009) provide the formal language for reasoning about interventions (do-calculus) and counterfactuals. This causal perspective has yielded principled methods across several areas: counterfactual fairness (Kusner et al., 2017) uses SCMs to define equitable predictions; causal representation learning (Schölkopf et al., 2021) seeks latent factors that reflect the underlying data-generating process; and inverse propensity weighting (Rosenbaum and Rubin, 1983) underpins treatment-effect estimation and off-policy evaluation. Doubly robust estimators (Robins et al., 1994; Wang et al., 2024a) combine outcome modeling with propensity weighting, providing consistency when either model is correctly

specified while reducing variance of pure IPW. Recent work has begun integrating causal reasoning into LLM-based agents: Causal-aware LLMs (Hu et al., 2025) embed SCMs into RL agents through a learning–adapting–acting paradigm for environment modeling, while REMI (Raman et al., 2025) constructs personal causal knowledge graphs for lifestyle recommendation. Neither, however, applies causal inference to memory management itself. Doing so requires more than transplanting standard causal machinery: as Section 2.3 shows, identification depends critically on the retrieval process, whose support determines which memory effects can be learned from the observed data.

B NOTATION

Table 5 summarizes the notation used throughout the paper.

Table 5: Summary of notation used throughout the paper.

Symbol	Description
$\mathcal{M} = (\mathcal{U}, \mathcal{V}, \mathcal{F})$	Structural causal model (Def. 1)
M_t	Memory state at time t
M_t^{-m}	Memory state excluding m, $M_t \setminus \{m\}$
$\mathbb{M}$	Universe of possible memories
R_t	Retrieved subset at time t
O_t	Memory operation at time t
$\mathcal{O}$	Operation space $\{\texttt{store}, \texttt{forget}, \texttt{consolidate}, \texttt{noop}\}$
Y_t	Task output at time t
q_t, c_t	Query and observed context at time t
$\mathcal{Q}$	Task distribution over queries
θ_{LLM}	Frozen LLM parameters
$U_t(m)$	Memory utility at time t (Def. 3); written $U(m)$ when unambiguous
$U_R(m \mid q, c)$	Conditional utility given retrieval (Eq. 3)
$\Delta_{\text{consol}}(m_i, m_j)$	Consolidation criterion (Eq. 2)
B	Context budget, number of memories the generator receives
k	Slots withheld from the ranker per timestep, $k \leq B$
$\mathcal{P}$	Exposure pool, memories whose utility is estimated
T	Horizon, number of estimation draws
E_t	Exposed set at time t (Eq. 5)
$Z_m^{(t)}$	Retrieval indicator $\mathbf{1}_{m \in R_t}$, randomized by the design
n_m	Exposures allocated to m under balanced assignment
e_m	Known exposure propensity, $e_m = n_m/T$
$\mathcal{T}_m^+, \mathcal{T}_m^-$	Timesteps with m exposed and not exposed
$\hat{U}_R(m)$	Hájek estimate of $U_R(m)$ (Eq. 6)
$\hat{\sigma}(m)$	Standard error of $\hat{U}_R(m)$ (Prop. 2)
σ_+^2, σ_-^2	Outcome variance within the exposed and unexposed arms
$\hat{U}_{\text{obs}}(m)$	Observational (AIPW) comparator (Eq. 8)
$\hat{U}_{\text{exp}}(m)$	Store-level randomized comparator
p_{flip}	Store-level perturbation rate
$\hat{p}, \hat{\mu}$	Propensity and outcome models (Eq. 7)
$\Gamma(m)$	Disagreement diagnostic, reported as a support audit
ψ	Feature map for the outcome model (Eq. 7)
H, τ	Horizon and interval to the next decision
g_λ	Threshold function of Eq. 9
$z^\star(\lambda)$	Abstention threshold, the root of g_λ
ϕ, Φ	Standard normal density and distribution function
ϵ_p	Propensity clipping threshold for the observational comparator, 0.01
$n_{\min}$	Minimum draws per arm for a memory to count as identified
Λ	Sensitivity parameter for unmeasured confounding

C THEORETICAL GUARANTEES

This appendix collects the formal statements and proofs supporting Sections 2.3 and 3. Appendix C.1 proves the factorization and non-identifiability results that motivate the exposure design; Appendix C.2 gives the estimator's properties and the budget it requires; Appendix C.3 proves the decision rule under irreversible operations; and Appendix C.4 states the conditions required by the observational comparator, which CMP does not need.

C.1 RETRIEVAL-MEDIATED IDENTIFICATION

This subsection proves the two results of Section 2.3. Both follow from the single structural fact that Y_t depends on M_t only through R_t, and neither requires an assumption beyond Definition 1 and the slot-substitution convention of Proposition 1, which fixes what $\mathrm{do}(m \notin R)$ means.

Proposition 1 (Retrieval-Mediated Factorization)**.** *Let $do(m \in R)$ denote inserting m into the retrieved set R, displacing its lowest-ranked memory, and $do(m \notin R)$ denote removing m and filling its slot with the highest-ranked memory not already in R, so that $|R|$ remains fixed across both arms. Both are dynamic interventions: the displaced or substituted memory depends on the realized ranking at (M_t, q_t). Under Definition 1, for any m and any (q, c),*

$$\begin{aligned} &U(m \mid q, c) = P(m \in R \mid q, c) \cdot U_R(m \mid q, c), \quad U(m) = \mathbb{E}_{q,c}\big[\, U(m \mid q, c) \,\big], \\ &U_R(m \mid q, c) \triangleq \mathbb{E}\big[Y \mid do(m \in R),\, q, c\big] - \mathbb{E}\big[Y \mid do(m \notin R),\, q, c\big]. \end{aligned} \tag{3}$$

Proof. Fix (M^{-m}, q, c) and suppress M^{-m} in the notation throughout, since it is held fixed. Write $A = \mathbf{1}_{m \in R}$ for the retrieval indicator under the observational regime, so $\Pr(A = 1) = P(m \in R \mid q, c)$. By Definition 1, $Y = f_Y(R, q, \theta_{\text{LLM}})$, so Y is a deterministic function of R given (q, θ_{LLM}) and the intervention $\mathrm{do}(M = M^{+m})$ acts on Y only by changing R.

Consider the two states M^{+m} and M^{-m} of Definition 3. Under the slot-substitution convention stated in Proposition 1, $|R|$ is held fixed across the two arms, so the retrieved sets differ in at most the single element m: either $m \in f_R(M^{+m}, q)$ and $f_R(M^{-m}, q)$ replaces it by the highest-ranked memory not already retrieved, or $m \notin f_R(M^{+m}, q)$ and the two retrieved sets coincide. Partitioning on which case obtains,

$$\begin{aligned} &\mathbb{E}\big[Y \mid \mathrm{do}(M = M^{+m}), q, c\big] - \mathbb{E}\big[Y \mid \mathrm{do}(M = M^{-m}), q, c\big] \\ &= \Pr(A = 1) \cdot \Big(\mathbb{E}\big[Y \mid \mathrm{do}(m \in R), q, c\big] - \mathbb{E}\big[Y \mid \mathrm{do}(m \notin R), q, c\big]\Big) \\ &\quad + \Pr(A = 0) \cdot 0, \end{aligned}$$

since on the event $A = 0$ the two interventions induce the same retrieved set and therefore the same outcome. The second factor in the first term is $U_R(m \mid q, c)$ by definition, giving

$$U(m \mid q, c) = P(m \in R \mid q, c) \cdot U_R(m \mid q, c).$$

Taking the expectation over (M^{-m}, q, c) and applying Definition 3 yields Eq. 3. □

Remark 1 (Store-level randomization does not repair this at feasible horizons)**.** *A design that perturbs the entire store rather than m alone can in principle surface m: under independent retention at rate $1 - p_{\text{flip}}$, a memory at rank r enters the top-B whenever at least $r - B$ of the $r - 1$ higher-ranked memories are dropped, an event of probability $\Pr[\mathrm{Bin}(r - 1, p_{\text{flip}}) \geq r - B] > 0$. The obstruction is therefore overlap rather than identification in the strict sense. It is nonetheless binding: at $p_{\text{flip}} = 0.1$, $B = 6$, and $r = 23$, this probability is below 10^{-8}, so observing a single inclusion requires a horizon beyond any feasible budget. Section 4 measures the consequence: 52.2% of store perturbations left the retrieved set unchanged, and no perturbation produced an inclusion for the memories of Table 1.*

Theorem 1 (Non-Identifiability under Retrieval Support Failure)**.** *Fix (q, c) and suppose $P(m \in R \mid q, c) = 0$. Then $f_R(M^{+m}, q) = f_R(M^{-m}, q)$, hence $Y(M^{+m}) = Y(M^{-m})$ almost surely, and $U(m \mid q, c) = 0$ irrespective of $U_R(m \mid q, c)$. Consequently, $U_R(m \mid q, c)$ is not identified by any design that holds M_{-m} fixed while randomizing only whether m is included.*

Proof. Fix (M^{-m}, q, c) with $P(m \in R \mid q, c) = 0$. Then $m \notin f_R(M^{+m}, q)$ almost surely, and by the slot-substitution convention $f_R(M^{-m}, q)$ differs from $f_R(M^{+m}, q)$ only by removing m and backfilling; since m was not present, no removal occurs and

$$f_R(M^{+m}, q) = f_R(M^{-m}, q) \quad \text{almost surely.}$$

Because $Y = f_Y(R, q, \theta_{\text{LLM}})$ is a deterministic function of R given (q, θ_{LLM}), the two arms produce identical outcomes, so $U(m \mid q, c) = 0$. This also follows from Proposition 1 with the leading factor equal to zero, and holds for every value of $U_R(m \mid q, c)$.

For the second claim, let $\mathcal{D}$ be any design that assigns membership of M_t by a known or unknown randomization, and let Π denote the induced distribution over observed trajectories $(M_t, R_t, q_t, c_t, Y_t)$. Every realization of $\mathcal{D}$ determines $R_t = f_R(M_t, q_t)$ through the same deterministic map, so for any two memory states differing only in m the observed R_t and hence Y_t are identical.

The restriction to (q, c) matters here. A design may well surface m at some other query (q', c') and thereby identify $U_R(m \mid q', c')$, but U_R is a conditional quantity: observations at (q', c') place no constraint on its value at (q, c) without a further assumption relating the two, which Definition 1 does not supply. Consequently Π is invariant to $U_R(m \mid q, c)$: two data-generating processes agreeing in every respect except the value of $U_R(m \mid q, c)$ induce the same distribution over observables. No functional of Π can therefore distinguish them, which is the definition of non-identifiability. The argument does not use the form of the randomization and applies in particular when membership of M_t is randomized with propensity bounded away from zero and one. □

Remark 2 (Why the failure is invisible)**.** *Theorem 1 concerns identification, not estimation. An estimator applied to such a memory does not fail to return a value; it returns a difference of two sample means drawn from the same distribution, which concentrates at zero. The estimate is therefore indistinguishable from that of a memory with $U_R(m) = 0$, and the two are separated only by the design's support, which the estimate itself does not report.*

In the strongest case the two arms contain byte-identical contexts on every draw, and the estimate is exactly zero with zero variance; Appendix H.1 gives such a memory.

C.2 Estimation Under the Exposure Design

Proposition 2 (Hájek Estimator Under Balanced Exposure)**.** *Under the exposure design of Eq. 5 with balanced assignment, $\mathbb{E}[\hat{U}_R(m)] = U_R(m)$ for every $m \in \mathcal{P}$, and $\text{Var}(\hat{U}_R(m)) = \sigma_+^2/n_m + \sigma_-^2/(T - n_m)$ exactly, with n_m fixed by the design.*

Proof. Write $Z_m^{(t)} = \mathbf{1}_{m \in R_t}$. Balanced round-robin assignment (Algorithm 2) fixes the number of exposures $n_m = \sum_{t=1}^{T} Z_m^{(t)}$ in advance, with $e_m = n_m/T$ known. Two consequences follow immediately and are used throughout.

The arm sizes are deterministic. $|\mathcal{T}_m^+| = n_m$ and $|\mathcal{T}_m^-| = T - n_m$ hold with probability one, not in expectation. The randomness in the design lies entirely in *which* timesteps receive the exposure, drawn by a uniformly random permutation of the assignment, and not in how many.

The self-normalized weights collapse. Since $Z_m^{(t)} \in \{0, 1\}$ and e_m does not vary with t,

$$\frac{\sum_t Z_m^{(t)} Y_t / e_m}{\sum_t Z_m^{(t)} / e_m} = \frac{\sum_t Z_m^{(t)} Y_t}{\sum_t Z_m^{(t)}} = \frac{1}{n_m} \sum_{t \in \mathcal{T}_m^+} Y_t,$$

and symmetrically for the excluded arm with $T - n_m$ in place of n_m. This is the second equality of Eq. 6.

Unbiasedness. By construction the permutation is drawn independently of (q_t, c_t) and of the outcome mechanism, so $Z_m^{(t)} \perp\!\!\!\perp (q_t, c_t)$. Writing $\mu_1 = \mathbb{E}[Y_t \mid Z_m^{(t)} = 1]$ and $\mu_0 = \mathbb{E}[Y_t \mid Z_m^{(t)} = 0]$, each term of the first arm satisfies $\mathbb{E}[Y_t \mid t \in \mathcal{T}_m^+] = \mu_1$, and the arm is an average of n_m such terms, so $\mathbb{E}\big[\frac{1}{n_m} \sum_{t \in \mathcal{T}_m^+} Y_t\big] = \mu_1$. The same holds for the second arm with μ_0. Hence

$$\mathbb{E}\big[\hat{U}_R(m)\big] = \mu_1 - \mu_0 = \mathbb{E}\big[Y \mid \text{do}(m \in R)\big] - \mathbb{E}\big[Y \mid \text{do}(m \notin R)\big] = U_R(m),$$

where the second equality holds because the exposure indicator is assigned by the design and is therefore independent of all other variables in $\mathcal{M}$, so conditioning on it coincides with intervening on it. No assumption on confounding between (q_t, c_t) and the memory state is used.

Variance. In the estimation regime, the memory state is held fixed across draws, so outcomes at distinct draws are independent given the design; under a sequentially evolving store, the potential

outcomes are auto-correlated through M_t, and the expression below is a lower bound. Writing $\sigma_+^2 = \mathrm{Var}(Y_t \mid Z_m^{(t)} = 1)$ and $\sigma_-^2 = \mathrm{Var}(Y_t \mid Z_m^{(t)} = 0)$,

$$\mathrm{Var}\big(\hat{U}_R(m)\big) = \mathrm{Var}\Big(\tfrac{1}{n_m}\sum_{t\in\mathcal{T}_m^+} Y_t\Big) + \mathrm{Var}\Big(\tfrac{1}{T-n_m}\sum_{t\in\mathcal{T}_m^-} Y_t\Big) = \frac{\sigma_+^2}{n_m} + \frac{\sigma_-^2}{T-n_m}.$$

The denominators are constants fixed by the design, so this expression is exact and not conditional on realized arm sizes. Under independent inclusion with $\Pr(Z_m^{(t)} = 1) = e_m$ the arm sizes are Binomial(T, e_m) and the corresponding statement holds only conditionally on them, with an additional term from their variability; balanced assignment removes it.

The standard error used in Section 3.3 is $\hat{\sigma}(m) = \big(\hat{\sigma}_+^2/n_m + \hat{\sigma}_-^2/(T-n_m)\big)^{1/2}$ with $\hat{\sigma}_\pm^2$ the within-arm sample variances, and is defined whenever $n_m \geq 2$ and $T - n_m \geq 2$. □

Corollary 1 (Exposure Budget)**.** *Under balanced assignment with budget B, exposed slots k, and pool $\mathcal{P}$, every $m \in \mathcal{P}$ receives $n_m \in \{\lfloor Tk/|\mathcal{P}|\rfloor, \lceil Tk/|\mathcal{P}|\rceil\}$ exposures. Attaining a minimum of $n_{\min}$ draws in each arm for every pool member therefore requires*

$$T \;\geq\; n_{\min} \cdot \frac{|\mathcal{P}|}{k}, \tag{10}$$

so the horizon grows linearly in $|\mathcal{P}|/k$ and does not depend on $|M_t|$.

Proof. By construction the Tk exposure slots are divided as evenly as possible among $|\mathcal{P}|$ members, so $n_m \geq \lfloor Tk/|\mathcal{P}|\rfloor$ for every m. Requiring $\lfloor Tk/|\mathcal{P}|\rfloor \geq n_{\min}$ gives Eq. 10. The excluded arm satisfies $T - n_m \geq n_{\min}$ when $\lceil Tk/|\mathcal{P}|\rceil \leq T - n_{\min}$, that is when $k/|\mathcal{P}| < 1$ with margin. When k is large relative to $|\mathcal{P}|$ a memory may be exposed on nearly every timestep and the excluded arm starves; Section 4.3 reports this case. □

Remark 3 (What the budget does not buy)**.** *Eq. 10 bounds the horizon required to identify the members of $\mathcal{P}$, not the full store M_t. Identifying additional memories requires enlarging $\mathcal{P}$, increasing the required horizon proportionally. The budget therefore trades coverage for horizon, and the choice of $\mathcal{P}$ determines which memories can be identified.*

C.3 Decision Rules Under Irreversibility

Section 3.3 asks what an identified estimate supports as an operation. The answer turns on an asymmetry in the losses rather than on the estimate's precision: `forget` destroys the memory and the channel through which its utility could be re-estimated, while `noop` is revisited at the next decision point. The following result characterizes the Bayes rule under that asymmetry.

Theorem 2 (One-Sided Abstention is Bayes-Optimal)**.** *Let $\lambda = H/\tau \geq 1$ and $z = \hat{U}_R(m)/\hat{\sigma}(m)$. The Bayes-optimal rule is $\texttt{forget}(m) \iff g_\lambda(z) < 0$, where g_λ is strictly increasing with a unique root $z^\star(\lambda)$ satisfying $z^\star(\lambda) < 0$ for $\lambda > 1$, $z^\star(1) = 0$, and $z^\star(\lambda) \to -\infty$ as $\lambda \to \infty$.*

Proof. Under the posterior $U \sim \mathcal{N}(\hat{U}_R(m), \hat{\sigma}^2(m))$, write $z = \hat{U}_R(m)/\hat{\sigma}(m)$ and use the standard truncated-normal moments

$$\mathbb{E}[\max(0, U)] = \hat{\sigma}\big[z\Phi(z) + \phi(z)\big], \qquad \mathbb{E}[\max(0, -U)] = \hat{\sigma}\big[-z\Phi(-z) + \phi(z)\big].$$

Eviction is preferred iff $\mathbb{E}[L_{\texttt{forget}}] < \mathbb{E}[L_{\texttt{noop}}]$, that is iff

$$H\hat{\sigma}\big[z\Phi(z) + \phi(z)\big] < \tau\hat{\sigma}\big[-z\Phi(-z) + \phi(z)\big].$$

Dividing by $\tau\hat{\sigma} > 0$ and setting $\lambda = H/\tau$, this is $\lambda[z\Phi(z) + \phi(z)] + z\Phi(-z) - \phi(z) < 0$, which is $g_\lambda(z) < 0$ for g_λ as in Eq. 9.

(i) Monotonicity. Using $\phi'(z) = -z\phi(z)$,

$$\begin{aligned} g'_\lambda(z) &= \lambda\Phi(z) + \lambda z\phi(z) + \Phi(-z) - z\phi(z) - (\lambda - 1)z\phi(z) \\ &= \lambda\Phi(z) + \Phi(-z) + z\phi(z)\big[\lambda - 1 - \lambda + 1\big] = \lambda\Phi(z) + \Phi(-z) \;>\; 0. \end{aligned}$$

(ii) Existence, uniqueness, and sign of the root. $g_\lambda(0) = (\lambda - 1)\phi(0) > 0$ for $\lambda > 1$, and $g_\lambda(z) \to -\infty$ as $z \to -\infty$ since $z\Phi(-z) \to -\infty$ while the remaining terms are bounded. Continuity and (i) give a unique root, and $g_\lambda(0) > 0$ places it strictly below zero. The rule $g_\lambda(z) < 0$ is therefore $z < z^\star(\lambda)$, which is Eq. 9.

(iii) The symmetric case. At $\lambda = 1$, $g_1(z) = z[\Phi(z) + \Phi(-z)] = z$, whose unique root is $z = 0$; the rule reduces to thresholding on the sign of $\hat{U}_R(m)$ and abstention disappears.

(iv) Monotonicity in λ. $\partial g_\lambda/\partial\lambda = z\Phi(z) + \phi(z) = \mathbb{E}[\max(0, U)]/\hat{\sigma} > 0$ for every z, so g_λ increases pointwise in λ and its root moves left; strict monotonicity of $z^\star$ follows from (i) and the implicit function theorem. Since $z\Phi(z) + \phi(z) > 0$ everywhere, for any fixed z there is a λ beyond which $g_\lambda(z) > 0$, so no finite z remains in the eviction region as $\lambda \to \infty$. □

Corollary 2 (Two-sided rules are dominated)**.** *For $\lambda > 1$, any rule that evicts some m with $\hat{U}_R(m) > 0$ incurs strictly greater expected loss than abstaining on that m.*

Proof. If $\hat{U}_R(m) > 0$ then $z > 0$, and by (i)–(ii) of Theorem 2, $g_\lambda(z) > g_\lambda(z^\star) = 0$. Since $g_\lambda(z) < 0$ characterizes the region where eviction has lower expected loss, eviction is strictly worse than `noop` at every such z. A rule of the form $|\hat{U}_R(m)| > \delta\hat{\sigma}(m)$ evicts on $z > \delta$ and is therefore dominated for every $\delta > 0$. □

Remark 4 (Behaviour under failed identification)**.** *When a memory has no support, $\hat{\sigma}(m)$ is unbounded and $z \to 0$, so $g_\lambda(z) \to (\lambda - 1)\phi(0) > 0$ and the rule abstains. The rule therefore degrades to `noop` exactly on the memories of Theorem 1, without requiring a separate diagnostic to detect them.*

Remark 5 (Calibration)**.** *Solving $g_\lambda(z^\star) = 0$ numerically gives $z^\star(10) \approx -0.90$, $z^\star(100) \approx -1.72$, and $z^\star(1000) \approx -2.42$. The threshold grows logarithmically in λ, so even a thousand-fold irreversibility penalty calls for roughly 2.4 standard errors.*

C.4 Conditions Required by the Observational Comparator

The observational comparator of Section 3.2 estimates $U(m)$ from logged interactions without randomization, and therefore requires two conditions the exposure design does not.

Condition 1 (Inclusion-Level Ignorability)**.** *Conditional on (M_t^{-m}, q_t, c_t), the inclusion indicator $\mathbf{1}_{m \in M_t}$ is independent of the potential outcomes: $Y_t(a) \perp\!\!\!\perp \mathbf{1}_{m \in M_t} \mid M_t^{-m}, q_t, c_t$ for $a \in \{0, 1\}$.*

The assumption is stated at the level of the treatment the estimator uses. Inclusion is a deterministic function of the operation history, so per-step ignorability for O_t does not by itself deliver exchangeability for $\mathbf{1}_{m \in M_t}$.

Condition 2 (Feasibility-Restricted Positivity)**.** *For all feasible operations $o \in \mathcal{O}(M_t)$ and all (M, q, c) in the support, $p(O_t = o \mid M_t = M, q_t = q, c_t = c) > 0$.*

Restricting to the feasible set $\mathcal{O}(M_t)$ avoids requiring nonzero probability for structurally impossible operations. This is the overlap condition prior work verifies, and it is independent of Assumption 1.

Under these two conditions the observational estimator is consistent for $U(m)$ if either nuisance model is correctly specified.

Proposition 3 (Double Robustness of the Observational Comparator)**.** $\hat{U}_{\text{obs}}(m) \xrightarrow{p} U(m)$ *if either $\hat{p}$ or $\hat{\mu}$ is correctly specified.*

Proof. We prove double robustness by showing $\mathbb{E}[\hat{U}_{\text{obs}}(m)] = U(m)$ when either $\hat{p} = p$ or $\hat{\mu} = \mu$.

Define the treatment indicator $A_i = \mathbf{1}_{m \in M_i}$ and the shorthand $e_i = p(m \in M_i \mid M_i^{-m}, q_i, c_i)$ for the true propensity, $\hat{e}_i = \hat{p}(m \in M_i \mid M_i^{-m}, q_i, c_i)$ for the estimated propensity, $\mu_1(M^{-m}, q, c) = \mathbb{E}[Y \mid m \in M, M^{-m}, q, c]$, and $\mu_0(M^{-m}, q, c) = \mathbb{E}[Y \mid m \notin M, M^{-m}, q, c]$. The AIPW estimator

(Eq. 8) can be written as:

$$\hat{U}_{\text{obs}}(m) = \frac{1}{n}\sum_{i=1}^{n}\left[\left(\hat{\mu}_1(M_i^{-m}, q_i, c_i) - \hat{\mu}_0(M_i^{-m}, q_i, c_i)\right)\right] + \frac{1}{n}\sum_{i=1}^{n}\left[\frac{A_i}{\hat{e}_i}(Y_i - \hat{\mu}_1(M_i^{-m}, q_i, c_i)) - \frac{1-A_i}{1-\hat{e}_i}(Y_i - \hat{\mu}_0(M_i^{-m}, q_i, c_i))\right].$$

Case 1: $\hat{\mu} = \mu$ (outcome model correct). When $\hat{\mu}_1 = \mu_1$ and $\hat{\mu}_0 = \mu_0$, the residuals $Y_i - \hat{\mu}_1(M_i^{-m}, q_i, c_i)$ have conditional mean zero given $A_i = 1, q_i, c_i$. Thus:

$$\mathbb{E}\left[\frac{A_i}{\hat{e}_i}(Y_i - \mu_1(M_i^{-m}, q_i, c_i)) \;\middle|\; q_i, c_i\right] = \frac{e_i}{\hat{e}_i}\cdot\mathbb{E}[Y_i - \mu_1(M_i^{-m}, q_i, c_i) \mid A_i = 1, q_i, c_i] = 0,$$

regardless of $\hat{e}_i$. The same holds for the $1 - A_i$ term. Therefore:

$$\mathbb{E}[\hat{U}_{\text{obs}}(m)] = \mathbb{E}[\mu_1(M^{-m}, q, c) - \mu_0(M^{-m}, q, c)] = U(m).$$

Case 2: $\hat{p} = p$ (propensity model correct). When $\hat{e}_i = e_i$, we expand each summand. For the A_i term:

$$\mathbb{E}\left[\frac{A_i}{e_i}Y_i \;\middle|\; q_i, c_i\right] = \frac{e_i}{e_i}\,\mu_1(M_i^{-m}, q_i, c_i) = \mu_1(M_i^{-m}, q_i, c_i).$$

Similarly, $\mathbb{E}\left[\frac{A_i}{e_i}\hat{\mu}_1 \;\middle|\; q_i, c_i\right] = \hat{\mu}_1(M_i^{-m}, q_i, c_i)$. The $\hat{\mu}_1$ terms from the outcome model and the IPW correction cancel:

$$\begin{aligned}\mathbb{E}[\text{summand}_i \mid M_i^{-m}, q_i, c_i] &= \hat{\mu}_1 - \hat{\mu}_0 + \mu_1 - \hat{\mu}_1 - (\mu_0 - \hat{\mu}_0)\\ &= \mu_1(M_i^{-m}, q_i, c_i) - \mu_0(M_i^{-m}, q_i, c_i).\end{aligned}$$

Thus $\mathbb{E}[\hat{U}_{\text{obs}}(m)] = \mathbb{E}[\mu_1(M^{-m}, q, c) - \mu_0(M^{-m}, q, c)] = U(m)$, regardless of $\hat{\mu}$.

Efficiency. When both models are correct, the influence function of $\hat{U}_{\text{obs}}$ equals the efficient influence function for the ATE (Robins et al., 1994):

$$\varphi(Y, A, M^{-m}, q, c) = \mu_1(M^{-m}, q, c) - \mu_0(M^{-m}, q, c) - U(m) + \frac{A}{e}(Y - \mu_1) - \frac{1-A}{1-e}(Y - \mu_0).$$

By the semiparametric efficiency theory of Bickel et al. (1993), $\text{Var}(\varphi) = \sigma^2_{\text{eff}}$ is the semiparametric efficiency bound, and $\sqrt{n}(\hat{U}_{\text{obs}} - U(m)) \xrightarrow{d} \mathcal{N}(0, \sigma^2_{\text{eff}})$. □

Throughout we suppress M_i^{-m} in the conditioning of expectations, since it is held fixed in every term; each $\mathbb{E}[\,\cdot \mid q_i, c_i]$ below denotes $\mathbb{E}[\,\cdot \mid M_i^{-m}, q_i, c_i]$.

C.5 The Store-Level Randomized Comparator

Proposition 4 (Unbiasedness of Store-Level Arm)**.** *Under independent store-level perturbation with rate p_{flip}, $\hat{U}_{\text{exp}}(m)$ is an unbiased utility $\mathbb{E}_{M^{-m}}[\mathbb{E}[Y \mid m \in M, M^{-m}] - \mathbb{E}[Y \mid m \notin M, M^{-m}]]$, regardless of confounding, with variance $\sigma^2_+/|\mathcal{T}^+_m| + \sigma^2_-/|\mathcal{T}^-_m|$.*

Proof. By Proposition 4, the shadow inclusion indicator $Z_m^{(t)} = \mathbf{1}_{m \in M_t^s}$ is drawn independently with $\Pr(Z_m^{(t)} = 1) = 1 - p_{\text{flip}}$, and crucially $Z_m^{(t)} \perp\!\!\!\perp (q_t, c_t)$ by construction (since shadow perturbations are drawn independently of the query and context at each timestep).

Define the conditional outcome function under the shadow state:

$$\mu^s(z, M^{-m}, q, c) = \mathbb{E}[Y_t^s \mid Z_m^{(t)} = z, M_{-m}^s, q_t = q, c_t = c],$$

where $M^{-m,s}$ denotes the shadow state of all memories except m.

Consider the expectation:

$$\mathbb{E}[\hat{U}_{\text{exp}}(m)] = \mathbb{E}[Y_t^s \mid m \in M_t^s] - \mathbb{E}[Y_t^s \mid m \notin M_t^s].$$

Since $Z_m^{(t)} \perp\!\!\!\perp (q_t, c_t)$, conditioning on the inclusion of m does not bias the distribution of queries. The remaining randomness comes from $M^{-m,s}$, where each other memory m' is independently included with probability $1 - p_{\text{flip}}$. Let $\mathcal{P}_{\text{shadow}}$ denote this distribution over M^{-m}. Then:

$$\mathbb{E}[Y_t^s \mid m \in M_t^s] = \mathbb{E}_{q,c}\,\mathbb{E}_{M^{-m}\sim\mathcal{P}_{\text{shadow}}}[\mathbb{E}[Y \mid m \in M, M^{-m}, q, c]],$$
$$\mathbb{E}[Y_t^s \mid m \notin M_t^s] = \mathbb{E}_{q,c}\,\mathbb{E}_{M^{-m}\sim\mathcal{P}_{\text{shadow}}}[\mathbb{E}[Y \mid m \notin M, M^{-m}, q, c]].$$

Therefore:

$$\mathbb{E}[\hat{U}_{\text{exp}}(m)] = \mathbb{E}_{M^{-m}\sim\mathcal{P}_{\text{shadow}}}\big[\mathbb{E}_{q,c}[Y \mid m \in M, M^{-m}] - \mathbb{E}_{q,c}[Y \mid m \notin M, M^{-m}]\big],$$

which is the marginal utility.

For the variance, since observations across timesteps are independent (conditional on the shadow design), and $|\mathcal{T}_m^+|$ and $|\mathcal{T}_m^-|$ are the sample sizes in each group:

$$\text{Var}(\hat{U}_{\text{exp}}(m)) = \frac{\text{Var}(Y_t^s \mid m \in M_t^s)}{|\mathcal{T}_m^+|} + \frac{\text{Var}(Y_t^s \mid m \notin M_t^s)}{|\mathcal{T}_m^-|} = \frac{\sigma_+^2}{|\mathcal{T}_m^+|} + \frac{\sigma_-^2}{|\mathcal{T}_m^-|}.$$

No assumption on the confounding structure between (q_t, c_t) and memory operations is required because the shadow perturbations are randomized independently of all other variables in the system. □

C.6 AIPW Estimator: Consistency, Efficiency, and Robustness

Condition 3 (Regularity Conditions for AIPW)**.** *The following hold: (i) the propensity is bounded away from zero and one: $\epsilon_p \le \hat{p}(m{\in}M \mid M^{-m}, q, c) \le 1 - \epsilon_p$ for all m, M^{-m}, q, c; (ii) at least one of $\hat{p}$ or $\hat{\mu}$ converges to its population counterpart at rate $o_p(n^{-1/4})$.*

Under these regularity conditions, the AIPW estimator inherits the double robustness and efficiency properties, adapted to the setting where the treatment is memory inclusion $\mathbf{1}_{m\in M_t}$ and the adjustment set is (M_t^{-m}, q_t, c_t).

Proposition 5 (AIPW Consistency for Memory Effects)**.** *Under Conditions 1 and 2, and Condition 3, the AIPW estimator $\hat{U}_{\text{obs}}(m)$ (Eq. 8) satisfies $\hat{U}_{\text{obs}}(m) \xrightarrow{p} U(m)$ if either $\hat{p} \xrightarrow{p} p$ or $\hat{\mu} \xrightarrow{p} \mu$ (double robustness). When both models are consistent, $\sqrt{n}\big(\hat{U}_{\text{obs}}(m) - U(m)\big) \xrightarrow{d} \mathcal{N}(0, \sigma_{\text{eff}}^2)$, where σ_{eff}^2 is the semiparametric efficiency bound for the ATE (asymptotic normality). If both $\hat{p}$ and $\hat{\mu}$ converge at rate $n^{-\alpha}$ for $\alpha > 1/4$* and *the nuisance models are fitted by cross-fitting, then $|\hat{U}_{\text{obs}}(m) - U(m)| = O_p(n^{-1/2})$. Algorithm 3 fits and evaluates on the same log; the consistency of Proposition 3 does not require it.*

Proof. This theorem adapts the classical AIPW results (Robins et al., 1994; Bang and Robins, 2005) to the memory operation setting. The key observation is that memory inclusion $A = \mathbf{1}_{m\in M}$ is a binary treatment, (M^{-m}, q, c) is the adjustment set, and Conditions 1 and 2 provide the standard unconfounded state and overlap conditions.

Double robustness. This follows directly from Proposition 3, which establishes $\mathbb{E}[\hat{U}_{\text{obs}}(m)] = U(m)$ when either model is correct. The law of large numbers then gives $\hat{U}_{\text{obs}}(m) \xrightarrow{p} U(m)$.

Asymptotic normality. When both models are consistent, the AIPW estimator is asymptotically linear with influence function φ (as derived in the proof of Proposition 3). By the CLT:

$$\sqrt{n}(\hat{U}_{\text{obs}}(m) - U(m)) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \varphi(Y_i, A_i, M_i^{-m}, q_i, c_i) + o_p(1) \xrightarrow{d} \mathcal{N}(0, \sigma_{\text{eff}}^2),$$

where $\sigma_{\text{eff}}^2 = \text{Var}(\varphi)$ equals the semiparametric efficiency bound (Bickel et al., 1993).

$$|\mathbb{E}[\hat{U}_{\text{obs}}(m)] - U(m)| = O(\|\hat{p} - p\| \cdot \|\hat{\mu} - \mu\|) = O(n^{-2\alpha}).$$

Since $2\alpha > 1/2$, this bias is $o(n^{-1/2})$, and the total error is dominated by the $O(n^{-1/2})$ variance term:

$$|\hat{U}_{\text{obs}}(m) - U(m)| = O_p(n^{-1/2}).$$

Condition 3(i) ensures the propensity weights are bounded, preventing the variance from exploding. In practice, this is enforced by propensity clipping at ϵ_p. □

Propensity-score clipping. The AIPW estimator can suffer from variance explosion when estimated propensities approach zero. To prevent this, CMP clips propensities at $\epsilon_p > 0$:

$$\hat{p}_{\text{clip}}(m{\in}M \mid M^{-m}, q, c) \;=\; \max\big(\hat{p}(m{\in}M \mid M^{-m}, q, c),\, \epsilon_p\big). \tag{11}$$

Clipping introduces a controlled bias, bounded by $|\mathbb{E}[\hat{U}^{\text{clip}}_{\text{obs}}(m)] - U(m)| \le 2\ \Pr(p(m{\in}M \mid \cdot) < \epsilon_p) \cdot \|Y\|_\infty$. For each arm the bias on the clipped region is $\mathbb{E}[Y(p/\epsilon_p - 1) \mid p < \epsilon_p]$, and since $0 \le p < \epsilon_p$ the factor $p/\epsilon_p - 1$ lies in $[-1, 0)$. This comes in exchange for variance bounded by $\sigma_Y^2/(n\epsilon_p^2)$. With $\epsilon_p = 0.01$ in our experiments, this trade-off keeps the AIPW estimator stable under mild positivity violations. Clipping bounds the variance but does not repair support: where $P(m \in R \mid q, c) = 0$, Theorem 1 shows no choice of ϵ_p recovers $U_R(m)$.

D Algorithm Details

Algorithm 1 gives the three-phase procedure of Section 3.4. Algorithms 2–4 specify the exposure assignment, the estimation step, and online operation.

Algorithm 1 Causal Memory Policy (CMP)

Input: Initial memory state M_0; context budget B; exposed slots $k \le B$; exposure pool $\mathcal{P} \subseteq M_0$; horizon T
Output: Per-memory estimates $\{\hat{U}_R(m), \hat{\sigma}(m)\}_{m\in\mathcal{P}}$ and comparator estimates

$\{n_m\}_{m\in\mathcal{P}},\ \pi \leftarrow$ BalancedAssignment$(\mathcal{P}, k, T)$ ▷ Alg. 2
$e_m \leftarrow n_m/T$ for all $m \in \mathcal{P}$ ▷ propensities are fixed, not estimated
$\mathcal{L} \leftarrow \emptyset$ ▷ interaction log
for $t = 1$ to T **do**
  receive query q_t, context c_t
  $E_t \leftarrow \pi(t)$ ▷ the k memories exposed at t
  $R_t \leftarrow$ top-$(B{-}k)(M_t, q_t) \cup E_t$
  $Y_t \leftarrow f_Y(R_t, q_t, \theta_{\text{LLM}})$; score Y_t
  $Z_m^{(t)} \leftarrow \mathbf{1}_{m\in R_t}$ for all $m \in \mathcal{P}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(M_t, q_t, c_t, R_t, Z^{(t)}, Y_t)\}$
  $O_t \leftarrow f_O(M_t, q_t, c_t)$ ▷ default policy
  $M_{t+1} \leftarrow f_M(M_t, O_t, q_t)$
end for
$\{\hat{U}_R, \hat{\sigma}, \hat{U}_{\text{obs}}, \hat{U}_{\text{exp}}, \Gamma\} \leftarrow$ Estimate$(\mathcal{L}, \mathcal{P}, \{e_m\})$ ▷ Alg. 3
Serve$(\mathcal{L}, \mathcal{P}, \{e_m\}, \{\hat{U}_R, \hat{\sigma}\})$ ▷ Alg. 4

Algorithm 2 BALANCEDASSIGNMENT: exposure allocation

Input: Pool $\mathcal{P}$; exposed slots per step k; horizon T
Output: Exposure counts $\{n_m\}$; assignment $\pi : \{1, \dots, T\} \to \mathcal{P}^k$

$S \leftarrow$ multiset containing each $m \in \mathcal{P}$ repeated $\lfloor Tk/|\mathcal{P}| \rfloor$ times
distribute the remaining $Tk - |S|$ slots to distinct members of $\mathcal{P}$, one each
$n_m \leftarrow$ multiplicity of m in S ▷ $n_m \in \{\lfloor Tk/|\mathcal{P}| \rfloor, \lceil Tk/|\mathcal{P}| \rceil\}$
draw a uniformly random permutation ρ of S
for $t = 1$ to T **do**
  $\pi(t) \leftarrow$ the t-th block of k consecutive elements of ρ
  if $\pi(t)$ contains a repeated memory **then**
    swap the duplicate with an element of a later block that does not conflict
  end if
end for
return $\{n_m\}, \pi$

The swap in Algorithm 2 preserves each n_m, leaving propensities unchanged while preventing duplicate exposures within a timestep. Overlap with the top-$(B-k)$ retrieval may still reduce the realized context below B distinct memories; realized sizes are reported in Appendix F.

Algorithm 3 ESTIMATE: utility estimation and comparators

Input: Log $\mathcal{L}$; pool $\mathcal{P}$; propensities $\{e_m\}$; clipping threshold ϵ_p
Output: $\{\hat{U}_R(m), \hat{\sigma}(m), \hat{U}_{\text{obs}}(m), \hat{U}_{\text{exp}}(m), \Gamma(m)\}_{m \in \mathcal{P}}$

for each $m \in \mathcal{P}$ **do**
  $\mathcal{T}_m^+ \leftarrow \{t : Z_m^{(t)} = 1\}, \quad \mathcal{T}_m^- \leftarrow \{t : Z_m^{(t)} = 0\}$
  $\hat{U}_R(m) \leftarrow \frac{1}{|\mathcal{T}_m^+|} \sum_{t \in \mathcal{T}_m^+} Y_t - \frac{1}{|\mathcal{T}_m^-|} \sum_{t \in \mathcal{T}_m^-} Y_t$ ▷ Eq. 6
  $\hat{\sigma}(m) \leftarrow \left(\hat{\sigma}_+^2/|\mathcal{T}_m^+| + \hat{\sigma}_-^2/|\mathcal{T}_m^-|\right)^{1/2}$ ▷ Prop. 2
  mark m unidentified if $\min(|\mathcal{T}_m^+|, |\mathcal{T}_m^-|) < n_{\min}$
end for
fit $\hat{p}(m \in M \mid M^{-m}, q, c)$ and $\hat{\mu}(a, M^{-m}, q, c)$ on $\mathcal{L}$ ▷ Eq. 7
clip $\hat{p} \leftarrow \max(\hat{p}, \epsilon_p)$
for each $m \in \mathcal{P}$ **do**
  $\hat{U}_{\text{obs}}(m) \leftarrow$ AIPW estimate ▷ Eq. 8
  $\hat{U}_{\text{exp}}(m) \leftarrow$ store-level arm estimate ▷ Prop. 4
  $\Gamma(m) \leftarrow |\hat{U}_{\text{obs}}(m) - \hat{U}_{\text{exp}}(m)| / \sqrt{\hat{\sigma}_{\text{obs}}^2(m) + \hat{\sigma}_{\text{exp}}^2(m)}$
end for
return all estimates and the identified/unidentified partition of $\mathcal{P}$

In Algorithm 3, $\Gamma(m)$ serves only as a diagnostic for memories retained by both comparator arms. Because its null is composite, it reports disagreement without attributing its cause.

Algorithm 4 SERVE: online operation

Input: Log $\mathcal{L}$; pool $\mathcal{P}$; assignment π; budget B; exposed slots k; current estimates; threshold $z^\star(\lambda)$

for each incoming query q_t **do**
  $E_t \leftarrow \pi(t); \quad R_t \leftarrow$ top-$(B-k)(M_t, q_t) \cup E_t$
  $Y_t \leftarrow f_Y(R_t, q_t, \theta_{\text{LLM}})$; return Y_t to the user
  append $(M_t, q_t, c_t, R_t, Z^{(t)}, Y_t)$ to $\mathcal{L}$; update $\hat{U}_R(m), \hat{\sigma}(m)$ incrementally
  if a downstream policy requests a decision for memory m at query q_t **then**
    return `forget` if $\hat{U}_R(m) < z^\star(\lambda)\, \hat{\sigma}(m)$, else `noop` ▷ Eq. 9
  end if
end for

In Algorithm 4, Line 6 applies Theorem 2 to a single query, not a retention policy. A retention policy must aggregate over queries, but Section 3.3 shows that every admissible aggregation is either attenuated or degenerate. Memories marked unidentified in Algorithm 3 have unbounded $\hat{\sigma}$ and therefore never satisfy the `forget` criterion, so no additional abstention mechanism is required.

E EXPERIMENTAL SETUP

E.1 BENCHMARK DATASETS

We evaluate on two conversational memory benchmarks, two multi-hop question answering datasets for replication, and one deployed memory system. For each, we describe the benchmark and the structural property that determines what claims it can support.

LongMemEval (Wu et al., 2024). A benchmark for long-horizon conversational memory. We use the `longmemeval_s` variant, comprising 500 records with haystacks averaging 487 turns. Each question is annotated with the sessions containing its required evidence, and turns within those sessions carry a `has_answer` flag. We focus on the 51 multi-evidence items whose annotations identify more than one required session. Based on the structure of the required evidence, we classify them as INDEPENDENT (30), OVERLAPPING (9), FILTERED (4), or SUSPECT (8). The SUSPECT items are excluded from aggregate results and reported separately in Appendix H. The INDEPENDENT subset contains 113 required memories, and the 43 items excluding SUSPECT contain 170. LongMemEval provides exactly one evaluation question per memory store; therefore, it supports estimating per-query memory utility but cannot evaluate transfer across queries.

LoCoMo (Maharana et al., 2024). A long-term, multi-session dialogue benchmark containing 10 conversations (5,882 turns) and 1,986 annotated questions. Unlike LongMemEval, each conversation is queried multiple times, allowing the same memory to contribute to different questions: 750 of 1,425 required memories are needed by at least two questions. LoCoMo is therefore the only benchmark in our study that supports evaluating memory utility across queries and the transfer analysis. Under the same retrieval protocol as LongMemEval, it contains 359 required $(\text{question}, \text{memory})$ pairs over stores averaging 588 memories.

HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022). Multi-hop question answering, used to test whether the identification result depends on conversational data. We sample 150 questions from each, stratified by question type for HotpotQA (bridge 120, comparison 30) and by hop count for MuSiQue (2-hop 78, 3-hop 47, 4-hop 25), matching the population proportions.

The standard distractor setting provides only 10 paragraphs per question, yielding a curated candidate set rather than a realistic memory store. We therefore pool all passages from the sampled questions into a shared corpus per dataset: 6,342 supporting sentences for HotpotQA and 2,959 paragraphs for MuSiQue. No sampled question shares annotated evidence with another ($0/360$ gold units on HotpotQA and $0/388$ on MuSiQue).

In HotpotQA, gold evidence always spans two documents but ranges from 2 to 5 supporting sentences (mean 2.4). In MuSiQue, the number of gold paragraphs equals the annotated hop count. We therefore report both unit-level recall and distinct-source coverage, since retrieving multiple units from the same source may still miss other required evidence.

Mem0 (Chhikara et al., 2025), version 2.0.18. A deployed memory system, used to test transfer. We replay LongMemEval histories through Mem0's own extraction, storage, and retrieval path without modification, recording every `add()` call, the memories it creates, and the candidate window passed to its memory manager. Stores average 291 memories after replay. Because Mem0 rewrites source turns into its own phrasing, annotated gold turns must be matched to stored memories before evaluation; Appendix F.4 describes the matching procedure and reports its uncertainty.

E.2 RETRIEVAL FUNCTION SPECIFICATION

Retrieval function $R_t = f_R(M_t, q_t)$ (Def. 1) selects a part of memories relevant to the current query.

Embedding and similarity. We use `all-MiniLM-L6-v2` from the sentence-transformers library to compute 384-dimensional embeddings for both queries and memories. All embeddings are computed locally on CPU. Relevance is measured by cosine similarity: $\text{sim}(q_t, m) = \frac{e(q_t)^\top e(m)}{\|e(q_t)\| \cdot \|e(m)\|}$.

Selection. The deterministic rule returns the highest-scoring memories to fill its allotted slots, subject to a minimum similarity threshold $\tau_{\text{ret}} = 0.3$; memories below the threshold are not retrieved regardless of rank, and fewer than the allotted number are returned when too few clear it. Under

the exposure design of Eq. 5 the rule fills $B - k$ slots, with the remaining k supplied by balanced round-robin draws from $\mathcal{P}$.

E.3 Systems and Implementation

Retrieval. Memories are embedded with `all-MiniLM-L6-v2` (384 dimensions) and ranked by cosine similarity. Under the exposure design of Eq. 5, the deterministic retriever fills $B - k$ context slots, while the remaining k are assigned by balanced round-robin exposure from $\mathcal{P}$ (Algorithm 2). Embeddings are computed once at insertion and cached. For the retrieval analysis, we additionally evaluate the bi-encoders `BGE-M3`, `jina-embeddings-v5-small`, `gemini-embedding-2`, and `text-embedding-3-large`, together with the cross-encoder `BGE-reranker-v2-m3`, applied to the top 50 cosine-retrieved candidates.

Generation. All outcomes are generated by a pinned `gemini-3.7-flash` at temperature 0 with `max_output_tokens` $= 1024$. Reasoning tokens cannot be disabled on this model and consume part of the budget; we log reasoning and visible tokens separately and flag every truncation. The generator receives exactly the retrieved memory texts and the question, with no item metadata; see Appendix I for a discrepancy between this and the repository's default serving path.

Memory representation. Each stored memory is a single conversational turn. Memory rendering differs across experiment families, and we note this in Appendix I because it affects the comparability of absolute scores across families, though not within them.

E.4 Exposure Design Parameters

Table 6 gives the design parameters for every experiment reported in Section 4. These differ across experiments and the differences are load-bearing: the number of exposed slots k determines whether a memory retrieved by the deterministic rule has an excluded arm, and the horizon T together with the pool size $|\mathcal{P}|$ determines draws per memory through Corollary 1.

Table 6: Exposure design parameters by experiment. Draws per memory is $Tk/|\mathcal{P}|$; the identification bar requires at least $n_{\min} = 5$ in each arm. On LoCoMo the horizon is spread across roughly 30 questions per conversation, so the pooled row supplies 5.0 draws per memory but only 3 per (memory, question) cell, which is why the per-query estimand requires a separate design.

Analysis	**Benchmark**	B	k	T	$\lvert\mathcal{P}\rvert$	**Draws/mem.**
Identification, single query	LongMemEval	6	2	90	≈ 32	5.6
Two-sided exposure	LongMemEval	6	2	90	≈ 32	5.6
Identification, pooled	LoCoMo	6	2	90	36	5.0
Identification, per query	LoCoMo	6	6	30	36	5.0

The pooled and per-query settings differ in two essential ways. First, the per-query design concentrates the horizon on fewer questions, providing enough draws to estimate a query-specific contrast. Second, it sets $k = B$, removing the deterministic retriever from the estimation draws entirely. This is necessary because required memories have median cosine rank 3: with $B - k = 4$ deterministic slots they are retrieved on every draw, leaving no excluded arm, and increasing T cannot create one. The two-sided design of Section 4.3 resolves the same obstruction by forcing exclusion of a deterministically retrieved memory.

Exposure pool construction. The exposure pool $\mathcal{P}$ consists of the required memories for the evaluated questions, plus the 20 highest-ranked and 10 uniformly sampled memories from the remainder of the store. Retrieval still ranks over the full store. This oracle pool isolates the effect of randomized exposure on identification; Appendix H.8 shows that practical pool-selection rules fail to recover comparable coverage.

E.5 Outcome Measurement

Outcomes are evaluated by token-level F1 and exact match after standard normalization (lowercasing, article and punctuation removal, and whitespace collapse). For aggregation and enumeration questions, we additionally compute task correctness: whether the response contains the correct numeric

result (aggregation) or all gold list elements (enumeration). We report task correctness as the primary metric for multi-evidence items and F1 as secondary, since F1 rewards partial evidence whereas task correctness requires all required evidence to be retrieved.

Exact match can also underestimate correctness on the multi-hop benchmarks, where gold answers sometimes include leading prepositions omitted by otherwise correct responses. We therefore additionally report a containment metric, computed offline from stored responses, and explicitly note whenever it leads to a different conclusion.

Why LLM-as-judge accuracy is not reported. We do not report judge-based accuracy anywhere in this paper. The protocol scores abstentions as correct at rates between 11% and 65% depending on the policy, so it rewards retrieving nothing. Rescoring a fixed set of 50 outputs 20 times yields accuracies spanning 0.30 to 0.38, and seven distinct backend configurations appear across $2{,}000$ calls, with variance estimates differing threefold across runs on identical inputs. No single dispersion figure characterizes the instrument, and we therefore report only deterministic metrics.

E.6 Statistical Procedure

Comparisons are paired at the item level on LongMemEval and at the question level on LoCoMo, with standard errors clustered by conversation. For experiments involving randomized context sets, we report the number of draws per cell and, for random-selection baselines, the number of independent runs averaged.

Identification criterion. A memory is treated as identified when $|\hat{U}_R(m)| > 2\hat{\sigma}(m)$ and both arms contain at least $n_{\min} = 5$ draws. The minimum-draw requirement is not redundant with the significance test: under inverse-propensity weights, a single observation in a thin arm can produce both a large estimate and a large t statistic, so the significance test alone admits high-variance singletons. Appendix H reports how the coverage comparison changes under each criterion, including a case where the ordering between two designs reverses. Coverage counts estimates that exist, which is why we judge designs on discrimination instead.

Experimental discipline. All experiments were preceded by pilot runs with manual inspection of raw generations. These uncovered three implementation errors before large-scale execution: an insufficient reasoning-token budget producing empty responses, a gold-turn reconstruction bug affecting the oracle selector, and a candidate-window attribution bug that inverted rank measurements. Each would have completed successfully while yielding misleading quantitative results.

F Measurement Protocols

This section defines the evaluation procedures used to produce the results in Section 4. Where measurements differ across benchmarks, we state the benchmark-specific definitions explicitly, since they affect the interpretation of the reported results.

F.1 Retrieval Support

The quantity $P(m \in R \mid q, c)$ is defined over the perturbations induced by the estimating design, not the deterministic retriever itself. For a fixed (m, q), retrieval is deterministic; the probability is taken over repeated perturbations of the store. Thus, $P(m \in R \mid q, c) = 0$ identifies memories that are never surfaced under that design (see Appendix H.1 for a worked case). We estimate $P(m \in R \mid q, c)$ as the fraction of perturbed draws in which m appears in the retrieved set. Consequently, $P(m \in R \mid q, c) = 0$ implies an empty included arm, and Theorem 1 applies.

Required memories. The definition differs by benchmark and we state each explicitly, because a mismatched definition changes the reported rate.

LongMemEval. A turn is required for a question when it lies in a gold session and carries the `has_answer` flag. Support is computed per $(\text{question}, \text{required turn})$ pair, and since each item has one question the pair count equals the turn count.

LoCoMo. A turn is required if annotated as evidence for a question. Because the same conversation is queried by many questions, retrieval support is defined per $(\text{question}, \text{required turn})$ pair. A turn

retrieved for one question but not another should not be counted as supported overall, as doing so substantially understates support failure.

Mem0. A stored memory is required when it matches a required turn by the procedure of Appendix F.4. Because Mem0 returns 20 results per read regardless of the requested limit, its support rate is computed at that effective depth and is not comparable in level to the top-6 retrieval used elsewhere; we report it as evidence that the failure occurs in a deployed system, not that its magnitude is smaller.

Perturbation efficacy. We report the fraction of store-level perturbations that leave the retrieved set unchanged, measured by comparing each perturbed retrieval with the unperturbed retrieval for the same store and query. This quantifies how often perturbing the store fails to change the model's input.

F.2 Estimation Under Exposure

Estimates follow Algorithm 3 with the parameters in Table 6. Three implementation details bear on the reported results.

Known propensities. Algorithm 2 serializes the exposure assignment before generation, and each propensity is read directly as $e_m = n_m/T$. No propensity model is fitted for the exposure arm.

Realized context size. A randomly exposed memory may already appear in the deterministic retrieval, reducing the realized context below the nominal budget B. We record the realized context size for every draw; under the two-sided design it averages 4.81 versus a nominal 5, corresponding to collisions in roughly one draw in five.

Self-normalization. All reported estimates use the Hájek estimator. Without self-normalization, inverse-propensity estimates inherit a multiplicative factor $1/e_m$ that varies across memories. Consistent with this, unnormalized estimates correlate with retrievability ($r = 0.123$ on LongMemEval and 0.289 on LoCoMo), whereas Hájek estimates show no correlation (0.014 and -0.005, respectively).

F.3 Comparator Arms

Store-level randomized arm. Each memory in $\mathcal{P}$ is independently retained with probability $1 - p_{\text{flip}}$, after which the ranker is applied to the perturbed store. Estimates are computed from timesteps at which m is eligible for inclusion, so that its presence or absence is determined by the randomization. Otherwise, timesteps where m was never eligible enter the excluded arm and bias the contrast.

Observational arm. The propensity and outcome models of Eq. 7 are fitted on logged interactions, conditioning on (M^{-m}, q, c). Propensities are clipped at ϵ_p, bounding the weights while introducing $O(\epsilon_p)$ bias.

Support-restricted comparison. To isolate support failure from estimator quality, we restrict both designs to memories with $P(m \in R \mid q, c) \geq 0.8$, i.e., those retrieved on nearly every draw. The two designs are then compared on the same memories and outcomes; outside this subset, the store-level design lacks an included arm, so the comparison is undefined.

F.4 Deployed-System Replay

We replay each LongMemEval history through Mem0 in batches of five conversational exchanges, matching its own API granularity, and record per `add()` call: the turns ingested, the memory identifiers created, the operations emitted, and the candidate window passed to the memory manager. Nothing in Mem0 is modified.

Write path. Mem0 passes the top-ranked existing memories to its manager as candidates when deciding what to do with new content. To determine whether a superseding fact arrives while its stale counterpart is visible, the history is replayed in batches up to the turn preceding the superseding turn, and that turn is then added alone, so the recorded window is keyed to that turn's own extracted fact. Batched addition produces a window keyed to whichever fact embedded last, which is generally unrelated; we report this in Appendix I as a measurement error we corrected.

The 18 required memories Mem0 never surfaces sit at median rank 139 of 291, the 48th percentile, with cosine similarity spanning 0.4475 to 0.5977 across ranks 15–296, so neither a wider retrieval window nor a similarity threshold reliably recovers them. Additional analyses show that the 18

required memories never surfaced by Mem0 have median rank 139 of 291 (48th percentile), while cosine similarity ranges from 0.4475 to 0.5977 across ranks 15–296, indicating that neither increasing the retrieval window nor similarity threshold reliably recovers them.

Matching stored memories to annotated evidence. Mem0 rewrites source turns into its own phrasing, so gold turns cannot be matched by string identity. We constrain candidates using Mem0's own provenance, restricting the search to memories created by the `add()` call that ingested the gold turn, which reduces the candidate set from roughly 300 memories to a mean of 4.1. Within that set we compute two independent signals: a value signal matching shared numeric or named-entity tokens, and an embedding signal matching cosine similarity.

The two signals agree on 69% of pairs after this constraint, improved from 22% without it. We report every Mem0 figure under three matching rules and give the spread, because the residual disagreement is not negligible and its direction is not neutral. Approximately one third of the residual cases are pairs whose evidence carries no numeric token, for which the value signal is structurally uninformative.

Selection in the agreement subset. Restricting to pairs where both signals agree yields a much lower support failure rate, and this is an artifact. Agreement requires an exact shared token, which is a property correlated with retrievability: agreed matches sit at median rank 3 of 291 against 12 for the remainder. Conditioning on agreement therefore selects on the outcome being measured. We report the least-conditioned rate as the estimate and the agreement-conditioned rate only as a bound whose bias direction is known.

F.5 Cross-Query Transfer

The transfer analysis tests whether utility estimated from one set of queries predicts a memory's contribution to another. we use LoCoMo, the only dataset with multiple questions per memory store.

Split. Each conversation's questions are partitioned into estimation and held-out halves by a recorded seed, stratified so that both halves contain questions whose required evidence overlaps and questions whose does not. Estimates are formed using only the estimation half.

Realized contribution. On the held-out half we measure each memory's contribution in two ways: by deletion, removing it from the context and recording the change in outcome; and by insertion, adding it to a context that lacks it. The two are reported separately because they can differ in sign for the same memory.

Aggregations tested. A retention policy must aggregate per-query utility estimates before the next query is known. We evaluate three aggregations: the mean, the maximum, and a similarity-weighted mean using the cosine similarity between each estimation query and the held-out query. The similarity-weighted variant is an oracle reference because it uses information unavailable at deployment. As a positive control, we correlate held-out contributions with the ground-truth per-query evidence labels, showing that the signal is present and recoverable.

Prompt-identity check. For each policy, we record the number of held-out questions whose post-eviction context is identical to the no-eviction baseline. A policy never changing the prompt cannot change the outcome, and reporting this distinguishes a genuine null from a policy that did not act.

F.6 Selector and Eviction Rules

Retrieval selectors. For the scoring analysis of Appendix H.8 we compare, at matched context budgets: cosine over each of five bi-encoders; a cross-encoder reranker applied to the top 50 cosine candidates; maximal marginal relevance, sweeping its trade-off parameter over $\{0.3, 0.5, 0.7, 0.9\}$ and reporting the full curve; a coverage-greedy selector that adds at each step the memory maximizing marginal coverage of required evidence; and the annotated evidence itself. The last two use gold knowledge and are reported as ceilings, never as methods.

Eviction rules. For the nomination analysis we compare four deployable rules that a capacity-bound system could run without gold knowledge: lowest access frequency, lowest cosine similarity to the query, a recency-weighted blend of the two, and newest-first. Each nominates the 30 lowest-scoring memories as removal candidates. We additionally evaluate nomination by estimated utility under each estimation design.

Chance baseline. Nomination rates are reported as lift over chance, defined as the observed rate at which a rule nominates a required memory divided by the rate expected if it nominated uniformly at random from the same store. Raw rates across rules with different candidate-set sizes are not comparable, and reporting them without this normalization would misstate the comparison.

G Baselines and Comparisons

This paper studies identification rather than benchmarking memory systems. Accordingly, this section describes the implemented systems, the reported comparisons and their rationale, and the comparisons we omit.

G.1 Memory Policies Implemented

We implement five memory management policies spanning the design space; all share the retrieval, generation, and scoring path of Appendix E.3, so differences are attributable to the management policy alone.

Recency. A first-in-first-out buffer. When capacity is exceeded, the oldest memory by insertion order is evicted. No content is examined.

MemoryBank (Zhong et al., 2024). Ebbinghaus-inspired forgetting. Each memory carries a retention score that decays exponentially with elapsed time and is reset on access; eviction removes the lowest score. Retention therefore tracks access recency.

MemGPT (Packer et al., 2023). LLM-managed paging between a working context and external storage, with the model itself issuing page-in and page-out calls. Eviction decisions are made by prompting.

A-Mem (Xu et al., 2025). Zettelkasten-style organization. Memories are linked by semantic similarity at write time and evicted by a combination of link degree and similarity to recent queries.

MemPO (Li et al., 2026). Reinforcement learning over memory operations, optimizing a policy that selects among store, forget, and consolidate from downstream task reward.

G.2 Reported Comparisons

The mechanism this paper identifies concerns which memories a policy nominates for removal, so we compare policies on that axis directly. Each rule nominates the 30 lowest-scoring memories in a store as removal candidates, and we measure how often a required memory appears among them, normalized by the rate expected under uniform nomination from the same store.

Table 7: Nomination of required memories as removal candidates, LongMemEval independent subset. Lift is the observed rate divided by the rate expected under uniform nomination; 1.00 is chance.

Nomination rule	Signal used	Lift over chance
Lowest access frequency	Access counts	0.00–0.11
Lowest cosine similarity	Query similarity	0.00
Recency-weighted blend	Both	0.00–0.01
Newest-first	Insertion order	0.34–0.86
Utility, store-level estimates	Estimated $U(m)$	**0.97**
Utility, exposure estimates	Estimated $U_R(m)$	**0.68**

Table 7 shows three things. First, heuristic policies rarely nominate required memories for eviction. This is not because they identify and preserve them, but because required memories are accessed substantially more often than typical ones ($1.892\times$ on average, in 29 of 30 items). Rules that evict the least accessed, least similar, or oldest memories therefore avoid the evidence that matters. Their competitiveness stems from this exclusion effect rather than accurate utility estimation, indicating that the identification failure affects only policies that rank memories by estimated utility.

Second, store-level utility estimates provide essentially no guidance for eviction. Their nomination rate is indistinguishable from chance, consistent with the observed AUC of 0.542. In contrast,

exposure-based estimates reduce the nomination rate to 0.68, indicating that identified utility carries meaningful information about which memories should not be evicted.

Third, the newest-first heuristic is the exception. It nominates required memories at a rate of 0.34–0.86, because required evidence tends to arrive late in the conversation, making it roughly an order of magnitude more destructive than the other heuristic policies.

G.3 Omitted Comparisons

We also conducted a conventional system-level comparison, evaluating the five baseline policies against CMP on downstream task accuracy. We do not report these results for three reasons.

1. *The outcome metric is unreliable.* As shown in Appendix E.5, the LLM judge rewards abstentions (11–65% of judged-correct responses) and varies substantially across repeated evaluations (accuracy 0.30–0.38), making policy rankings unstable.
2. *Item matching was not verified.* The CMP and baseline evaluations selected items through different execution paths, and we did not establish that they were evaluated on identical examples.
3. *The default serving path leaked metadata.* The repository's default prompt exposes question type and date (Appendix I), making its absolute scores incomparable with the metadata-free evaluation used throughout this paper.

For completeness, on the subset of matched items we could verify, none of the five baseline policies differed significantly from CMP in token-F1 or exact match. This is consistent with Table 7: policies that rarely evict required memories leave the retrieved evidence, and therefore the generated answer, essentially unchanged.

G.4 Retrieval Selectors

We evaluate whether stronger retrieval can repair retrieval support failure. The comparison includes five bi-encoders, a cross-encoder reranker, maximal marginal relevance (MMR), and a coverage-greedy selector, with annotated evidence as an oracle ceiling. Protocols are given in Appendix F.6, and complete results in Appendix H. We report the full MMR parameter sweep because its optimum is informative: across all datasets, the best setting nearly disables the diversity term.

H Additional Results

H.1 Worked Examples

Existing methods estimate memory utility by intervening on the store: comparing outcomes with and without a memory in the store. This works only when the retriever surfaces that memory. Otherwise, both arms produce the same retrieved context and the same outcome, making an unidentified memory indistinguishable from one with zero utility. In contrast, **CMP intervenes on retrieval:** it reserves a fixed number of context slots for memories sampled with known probability. Utility is then estimated from these randomized exposures with a quantified standard error (rather than from the retriever's decisions). This thus turns previously unidentified utilities into identifiable ones with quantified uncertainty. Below we will show details examples to illustrate it.

The three memories below are from the LongMemEval independent subset, selected on arm structure and text length. All are from the LongMemEval independent subset. Figure 4 shows the arm structure of all three under both designs. Each block is 90 estimation draws; blue marks draws in which the memory was retrieved, grey those in which it was not. **(a)** Store-level perturbation never surfaces the memory, so both arms contain the same contexts and the estimate is exactly zero with zero variance; exposure supplies six inclusions and recovers a signal. **(b)** The memory sits high enough in the ranking that exposure never excludes it, leaving no contrast. **(c)** Both designs have both arms, and the point estimates agree to three decimal places.

A memory the store-level design cannot see. Item `gpt4_2f56ae70` asks which streaming service the user began using most recently, with gold answer *Disney+*. One required turn reads: *"I'm having trouble finding a specific documentary I saw on Disney+ during my free trial last month."* It sits at cosine rank 23 of 458, so $P(m \in R \mid q, c) = 0$.

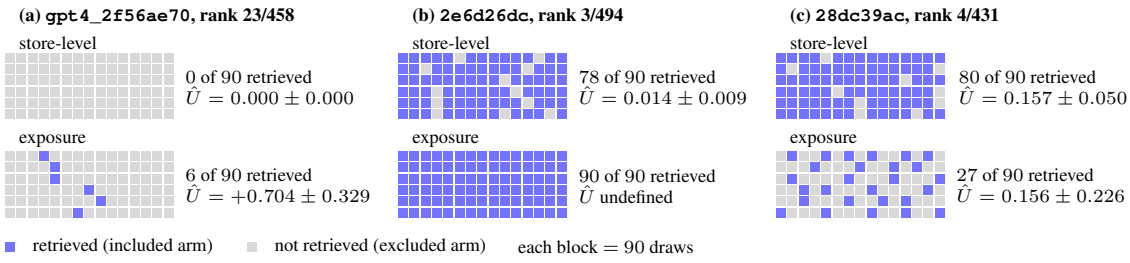

Figure 4: Arm structure for three required memories. **(a)** Store-level randomization does not surface the memory, so both arms contain identical contexts and no estimate exists; exposure recovers one. **(b)** The mirror case: the memory always ranks high enough that exposure does not exclude it, so exposure has no estimate and store-level does. **(c)** Where both designs have both arms, they agree to three decimal places, but exposure's standard error is 4.5 times larger: randomization spent on a memory the ranker already varies is wasted.

Under store-level perturbation the estimate is 0.000 with standard error 0.000: the retrieved set was byte-identical across all 90 draws, so the two arms contained the same experiment. Under exposure the same memory yields $+0.704 \pm 0.329$ from six forced inclusions. Ninety draws of store-level randomization produced no information; six of retrieval-level randomization produced a signal two standard errors from zero, at the minimum exposure count the identification criterion admits.

A memory the exposure design cannot see. Item `2e6d26dc` asks how many babies were born to friends and family, gold answer 5. A required turn about twin newborns sits at cosine rank 3 of 494 and is retrieved on essentially every draw. Store-level perturbation identifies it weakly, 0.014 ± 0.009. Exposure cannot: with 90 inclusions and no exclusions, the Hájek estimate is undefined. This is the case in which CMP's own support condition fails: Assumption 1 requires both arms, and a memory the ranker always retrieves has no excluded arm under uniform exposure. Section 4.3 reports the symmetric design that closes this gap and why we do not adopt it. Figure 4(b) shows the missing arm.

A memory both designs identify. Item `28dc39ac` asks for total hours spent gaming, gold answer 140. A required turn reporting 25 hours sits at rank 4 of 431 and is retrieved often enough for both designs to have support. Store-level gives 0.1566 ± 0.0497; exposure gives 0.1563 ± 0.2257. The point estimates agree to three decimal places, and the example was not chosen for this. Figure 4(c) shows both arms populated. The agreement is the point: it shows the two designs estimate the same quantity, which is what allows the aggregate gap in Table 3 to be attributed to coverage rather than to a difference in what is being measured. Exposure is nonetheless less efficient here, with 4.5 times the standard error, because it spends draws on a memory the ranker already varies. This is an inefficiency of allocating exposure uniformly over the pool, and it indicates the natural refinement: allocate by expected information gain, concentrating draws on memories whose contrast the ranker does not already supply. Exposure is worse here: the same estimate at 4.5 times the standard error. The design allocates exposure uniformly over the pool, so memories the ranker already varies receive draws that produce no contrast the store-level design lacks. This example is included because it shows the two designs target the same estimand, and because it bounds the claim: exposure's advantage is in coverage, not in estimation quality on common support.

Required is not the same as useful. A second required turn in `28dc39ac`, reporting 10 hours for a different game, has a recovered utility of -0.256 ± 0.065. Surfacing one partial summand in isolation leads the model to report that summand rather than the total, so a memory that is necessary for the answer can be harmful when retrieved without its companions. Annotated gold status and conditional utility are distinct quantities, and only the second is what a policy would act on.

H.2 Identification by Benchmark

Table 8 reports the support failure rate and discrimination for each benchmark under matched measurement, together with the design differences that prevent a like-for-like comparison of levels.

Three design differences bear on these rows. LongMemEval items carry one question each, so the estimate is inherently per query; LoCoMo items carry roughly thirty, so a pooled estimate averages a memory's contribution across questions for most of which it is irrelevant. Aligning the required-

Table 8: Retrieval support failure and discrimination by benchmark. Support failure is the fraction of required (question, memory) pairs with $P(m \in R \mid q, c) = 0$. AUC separates required from non-required memories by estimated utility.

Benchmark	Pairs	$P = 0$	Store	AUC store	AUC exposure
LongMemEval (independent)	113	54.0%	487	0.542	0.664
LongMemEval (all − suspect)	170	61.2%	487	0.529	—
LoCoMo (pooled estimand)	359	66.9%	588	0.513	0.551
LoCoMo (per-query estimand)	359	66.9%	588	—	**0.784**
Mem0 2.0.18	109	0.284 [0.208, 0.375]	291	—	—

Table 9: Answer-quality cost of one-sided exposure ($k = 2$ of $B = 6$), paired per item.

Population	n	Exposure	Cosine	Diff.	t	p
Independent	30	0.197	0.223	−0.026	−0.76	0.45
All − suspect	43	0.155	0.168	−0.013	−0.54	0.59
All	51	0.165	0.175	−0.010	−0.50	0.62

memory label with the questions that actually generated draws recovers 65% of the gap between the two LoCoMo rows. Mem0 returns 20 results per read regardless of the requested limit, against the top-6 retrieval used elsewhere, and counts extracted facts.

Support-restricted comparison. Restricted to memories with $P(m \in R \mid q, c) \geq 0.8$, the store-level and exposure designs give AUC 0.655 and 0.658 respectively, a difference within noise. The store-level estimator is not weak; its aggregate deficit arises entirely from memories on which it has no support. The gradient is monotone in support: median $|t|$ is 0.83 at $P = 0$ against 3.66 at $P \geq 0.8$, with 15% and 59% of estimates respectively distinguishable from zero.

Perturbation efficacy. Store-level perturbation leaves the retrieved set identical to the unperturbed ranking on 52.2% of draws on LongMemEval and 53.2% on LoCoMo. Under a matched generation budget the store-level design produced 940 distinct context sets against 3,939 for the exposure design when the query is held fixed; when the query varies across draws the ratio falls to 1.41, since query variation supplies context variety that store perturbation does not have to provide.

H.3 Cost of Exposure

Reserving context slots for randomized draws displaces ranked memories and can lower answer quality while estimates are collected. We measure this on the LongMemEval independent subset by comparing each item's mean token-F1 over its 90 exposure draws against its token-F1 under unperturbed cosine top-6 retrieval. The baseline consists of the store-level draws whose retrieved set equals the unperturbed ranking, which occur on 52.2% of draws, consistent with the $(1 - p_{\text{flip}})^B = 0.531$ expected under independent retention. Within each item these draws share one retrieved set and one outcome, so the baseline is a single value per item rather than an average.

Exposure lowers F1 on 10 of 30 independent items, raises it on 14, and leaves 6 unchanged, so the median change is zero. The negative mean is driven by two items on which cosine already retrieves the correct evidence, with baseline F1 of 0.87 and 1.00: there, displacing two of six slots can only remove evidence that was working. We report the full-sample figure as the cost and do not exclude these items. The cost is therefore not a uniform degradation but a concentrated one, borne by queries the ranker already serves well, which is the population on which exposure also buys the least identification.

H.4 Ablation Details

Table 10 decomposes the difference between designs and decision rules. Gold loss is required memories destroyed per 100 slots reclaimed; we report it because it isolates the two mechanisms cleanly, and note in Appendix H.6 that it does not predict answer quality.

Table 10: Component ablation, LongMemEval independent subset, eviction budget 10 per item. Destroyed counts required memories evicted; gold loss normalizes by slots reclaimed.

Variant	AUC	Evicted	Destroyed	Gold loss
Exposure + one-sided gate	0.664	116	**6**	**5.2**
Exposure, ungated	0.664	300	27	9.0
Store-level + one-sided gate	0.542	138	15	10.9
Store-level, ungated	0.542	300	35	11.7
Exposure + two-sided gate	0.664	156	22	14.1
Store-level + two-sided gate	0.542	132	28	21.2
Two-sided exposure + one-sided gate	0.671	151	10	6.6

Table 11: Recall of required evidence at matched context budget. Bi-encoders span two orders of magnitude in parameter count. Coverage-greedy uses gold knowledge and serves as a ceiling.

Selector	Type	Headroom recovered
Five bi-encoders (MiniLM to gemini-embedding-2)	Pointwise	at chance on contradiction pairs
Cross-encoder reranker (BGE-reranker-v2-m3)	Pointwise	10% LongMemEval, 30–33% HotpotQA, 17–24% MuSiQue
Maximal marginal relevance	Set-aware	-2.7% to $+4.8\%$
Coverage-greedy (ceiling)	Set-aware	95%

The components are not substitutes. Holding the rule fixed at one-sided, the design reduces destroyed memories from 15 to 6; holding the design fixed at exposure, the rule reduces them from 27 to 6. Neither alone approaches the joint result. The reason is structural: a confidence threshold filters estimates that exist, and on the 54% of memories without support there is nothing to filter.

Two-sided gating is worse than no gating. Every two-sided configuration has a higher gold-loss rate than its ungated counterpart, 21.2 against 11.7 for the store-level design. The rule $|\hat{U}_R(m)| > \delta\hat{\sigma}(m)$ admits confidently useful memories into the eviction pool, which Corollary 2 shows is strictly dominated. The effect is visible in the data and not only in the loss function.

Two-sided exposure closes the support gap and does not help. Adding a forced-exclusion slot gives every permanently retrieved memory an excluded arm, raising the excluded-arm count from 0.0 to 21.1 for all 40 affected memories. It buys 0.007 AUC and destroys more memories, 10 against 6. Three of the four additional losses are memories the one-sided design could not estimate at all. Identification is therefore not free: making a memory measurable moves it into the candidate pool, where a confidently signed but incorrect estimate can act on it.

The measured cost in answer quality is -0.0069 F1 pooled across all exclusions, with a tail reaching -0.64 on items whose required evidence occupies the deterministic slots. The pooled figure is the population effect and the tail is heterogeneity; we report both, since an item-level maximum overstates the typical cost by roughly two orders of magnitude.

H.5 Retrieval Scoring

Table 11 reports whether the support failure is repairable by better ranking. It is not: scoring quality moves recall by hundredths while the functional form of the selector moves it by tenths.

Two observations. The five bi-encoders rank the current of two contradicting facts above the stale one on 45.7–55.7% of pairs, a band containing chance, and the largest model tested is not the best. And on every corpus the swept optimum for maximal marginal relevance is the setting that most nearly disables its diversity term, so the redundancy penalty is not merely unhelpful but actively costly when required evidence is topically similar.

H.6 Cross-Query Transfer

Table 12 reports whether utility estimated on one set of queries predicts contribution on a disjoint set.

Power is not the constraint: 99.3% of required cells are identified with at least five draws per arm, and the control recovers the signal from the same data. Within the query on which it was estimated,

Table 12: Correlation between estimated utility and realized contribution on held-out queries in LoCoMo. Ground-truth per-query evidence labels provide a positive control, confirming that the signal is recoverable from the same data.

Predictor	**Deletion**	**Insertion**
Pooled estimate	-0.060	$+0.023$
Nearest estimation query	< 0.11	< 0.11
Similarity-weighted mean	< 0.11	< 0.11
Maximum over estimation queries	< 0.11	< 0.11
Ground-truth evidence labels (control)	$+\mathbf{0.286}$	—

the per-query estimate separates required from non-required memories at 0.784 AUC against 0.581 pooled, with a mean gap in estimated utility of 0.300 against 0.042. The estimator works; the estimate does not transfer.

Degeneracy of the admissible aggregation. A retention policy cannot condition on the query it has not yet received, so the only admissible aggregation is query-independent. The maximum over estimation queries is positive for essentially every memory, since a memory that helps some query among thirty is never confidently harmful, and the resulting rule evicts nothing: the post-eviction context is identical to the control on 300 of 300 held-out questions. The similarity-weighted form resolves this but requires the future query.

Attenuation of the pooled estimand. On LoCoMo, 77 of 109 required memories are required by exactly one of thirty estimation questions, so a pooled contrast averages one large effect with twenty-nine zeros. The pooled estimate rises with the number of questions for which a memory is required: -0.014 at zero, $+0.024$ at one, $+0.042$ at two, with the first step significant at $p = 0.028$. The estimator recovers the effect where the effect concentrates.

H.7 Suspect Items

Eight of the 51 multi-evidence LongMemEval items have gold answers that cannot be derived from their annotated evidence. In one case, the annotated values sum to \$850 while the gold answer is \$5,850; in another, computing the gold average requires ages absent from every annotated turn. Even with all annotated evidence provided, exact-match accuracy on these items is only 0.554 and token-F1 is 0.522, implying that 44.6% remain unanswerable under perfect retrieval. We therefore exclude these items from all aggregate results and report them separately.

H.8 Extended Analysis

Why the retriever misses required evidence. The memories the retriever never surfaces are not scored as irrelevant. For the 18 required memories Mem0 never retrieves, cosine similarities span only 0.4475–0.5977 while their ranks range from 15 to 296 in stores averaging 291 memories. The embedder therefore provides little separation between required evidence and background, leaving the final ranking largely determined by noise within this narrow similarity band. Consequently, neither widening the retrieval window nor replacing the scorer can substantially improve support, since both operate on an ordering that contains little signal about the quantity of interest.

Retrieval scoring in full. Six embedding models spanning two orders of magnitude in parameter count rank the current fact above a contradictory stale fact on only 45.7–55.7% of pairs, essentially chance, and the largest model is not the best. A cross-encoder reranker recovers only part of the available headroom, improving recall by 10% on LongMemEval, 30–33% on HotpotQA, and 17–24% on MuSiQue, compared with 95% for a coverage-greedy selector that conditions on previously selected evidence. Maximal marginal relevance changes recall by only -2.7% to $+4.8\%$, and its optimal parameter nearly disables the diversity term on every dataset. Overall, improving the scorer changes recall by only a few percentage points, whereas changing the selector changes it by tenths.

Exposure budget scaling. Each memory receives $Tk/|\mathcal{P}|$ exposure draws. The measured values match this prediction (3.60, 0.90, and 0.36 draws per memory for pool sizes 50, 200, and 487,

respectively). At full store size, 63.9% of memories receive no randomized exposure and none reaches the five-draw identification threshold, reducing discrimination from 0.765 to 0.651. Concentrating exposure on a candidate pool is therefore required by the arithmetic. This, however, requires a pool selector that contains the memories that matter. None of the deployable selectors in Appendix G provides adequate coverage, so pool selection remains the principal open problem.

I Reproducibility and Known Discrepancies

This section documents the evaluation instruments and implementation issues relevant to reproducing and interpreting the reported results.

I.1 Evaluation Instruments

The LLM judge is not stable. Rescoring the same 50 responses 20 times at temperature 0 yields accuracies from 0.30 to 0.38. Across $2{,}000$ scoring calls, the provider reported 7 backend configurations, and variance estimates on identical inputs differed by up to a factor of three. Because this variability is comparable to the effects under study, we report only deterministic evaluation metrics.

The judge scores abstention as success. Across policies, 11%–65% of responses judged correct are refusals to answer. A policy that retrieves nothing and abstains is therefore rewarded, and correcting for this reverses the ranking of methods on LongMemEval. This is a property of the evaluation protocol rather than of any particular method.

Exact match understates correctness on multi-hop data. Gold answers in HotpotQA and MuSiQue sometimes include leading prepositions omitted by otherwise correct responses, causing exact match to score them as incorrect. We therefore report a containment metric alongside exact match and note whenever the two lead to different conclusions.

Eight benchmark items are unanswerable as annotated. Appendix H.7 analyzes the eight SUSPECT items. Even with perfect retrieval, 44.6% remain incorrect, establishing a performance ceiling inherited by any evaluation on this subset.

I.2 Implementation Discrepancies

The repository's default answer path leaks item metadata. The serving code constructs a context line populated with the question type and question date, which reaches the generator. This does not affect any result in this paper: every experimental arm we report constructs its prompt from the retrieved memory texts and the question alone, verified by reconstructing the exact prompt from stored artifacts. But it means absolute scores obtained through the default path are not comparable to those reported here, and it is one reason the system-level comparison of Appendix G.3 is withheld.

It also bears on Definition 1. In the arms we report, c_t has no edge to Y_t and retrieval mediates every path from the memory state to the output. Under the default path c_t would be a direct cause of Y_t, though not a path from M_t, so Propositions 1 and 1 are unaffected either way.

Memory rendering differs across experiment families. Some arms present a stored turn as raw content, others prefix a role or speaker label. All are deterministic functions of the retrieved set alone, so mediation and the identification results are unaffected, but absolute scores are not byte-comparable across families. Every comparison we report is within a family.

Reasoning tokens cannot be disabled. The generator consumes part of its output budget on reasoning tokens that are not returned. We log reasoning and visible tokens separately and flag every truncation. Truncation occurs on 2 of $41{,}961$ calls (0.005%) at a $1{,}024$-token budget, with mean output 190.8 tokens and $p_{95} = 394$, and truncated cells are regenerated. A further 20 calls (0.048%) returned empty responses without truncating.

I.3 Pool Construction Determines the Conclusion

A methodological finding that our own measurements depend on. Evaluating selection policies over a small curated candidate set and over the full store can yield opposite conclusions on the same items with the same policies. On curated sets of at most ten candidates, recency-based selection performs

within 0.03 of an oracle; over full stores its recall of required memories is 0.000 through a context budget of eight. Similarity-based selection is significantly worse than random among topically similar candidates and significantly better among off-topic ones, with the sign reversing on pool size.

We report every result in this paper over full stores for this reason, and we note it here because it is a property of how such comparisons are constructed.

I.4 Compute and Cost

All experiments run on a single workstation with an 8-core CPU and 64 GB of RAM; no GPU is required, as generation is served by API and embedding models run on CPU. Evaluation is parallelized across items with eight workers. Total API expenditure across the experiments reported here is $\$193.06$ over $80{,}187$ generation calls, of which the deployed-system replay accounts for $\$106.55$. Earlier exploratory runs on the main benchmarks were conducted without cost instrumentation and their expenditure is not recoverable, so this figure covers the experiments described in Appendices F and H. Cost logs are written per experiment and are append-only; every reported figure is traceable to a logged call count and token count.

I.5 Artifacts

We release the estimation code, exposure assignment procedure, evaluation harness, per-experiment protocols, and raw generated outputs with their scores. Every table in Section 4 and in Appendix is reproducible from the released artifacts and recorded seeds. Assignment permutations, item selections, and pool constructions are seeded and the seeds are stored alongside the results they produced.

J Limitations

Identification does not deliver a retention policy. CMP produces an identified measurement of per-query memory utility, and we show that this quantity does not by itself determine what to retain. A retention decision precedes the next query, so a policy must aggregate over queries, and every aggregation available to it is either attenuated by the fraction of queries for which a memory is required, or degenerate in the sense that it retains everything. The aggregation that would work uses information a deployed policy does not have. We therefore make no eviction claim, and closing this gap requires either a different estimand or a source of information about the query distribution that we do not assume.

The exposure budget scales with the pool, not the store. By Corollary 1, identifying the members of an exposure pool $\mathcal{P}$ requires a horizon growing linearly in $|\mathcal{P}|/k$. At realistic store sizes, uniform exposure leaves most memories with no randomized draws, so the design must concentrate exposure on a candidate set. That requires a pool selector containing the memories that matter, and the three deployable rules we test have near-zero nomination of required memories. Our demonstration uses a pool constructed with knowledge of which memories are required, which isolates the identification effect and does not establish that a practical selector recovers it. Exposure also has a direct cost while estimates are collected: displacing ranked slots changes answer quality by -0.026 F1 on average, not statistically distinguishable from zero but concentrated on queries the ranker already answers well.

Scope. The identification failure binds on policies that rank memories by estimated utility. Heuristic policies are largely unaffected, not because they identify required memories but because required memories are accessed more often than typical ones, so frequency-, similarity-, and recency-based eviction routes around them. The result therefore constrains a class of methods rather than memory management in general. Our evidence is also drawn from a small number of benchmarks and one deployed system, and the level of the failure varies across them with the retrieval configuration.

K Applications of This Work

Retrieval-augmented generation and context attribution. Retrieval-augmented generation conditions a model's output on documents retrieved from an external corpus (Lewis et al., 2020), and context attribution asks which of those documents caused the answer. The field has moved from generating citations toward measuring causal influence directly. ContextCite ablates context sources

and fits a surrogate to the resulting changes in output (Cohen-Wang et al., 2024), Shapley-based methods value each retrieved document by its marginal contribution across subsets (Nematov et al., 2025), and internals-based methods trace an answer to its context through the model's own activations (Qi et al., 2024); the reliability of these attributions is now itself under study (Abolghasemi et al., 2025). Every one of these methods operates on documents already in the context. A document the retriever did not surface offers nothing to ablate, so attribution is silent about it and cannot distinguish a document that is irrelevant from one that was never tested. This is Assumption 1 failing in a second setting. Attribution explains the answer the system gave, but it cannot say whether a better answer was available from evidence the retriever passed over. Proposition 1 separates the two quantities attribution conflates, retrievability and conditional usefulness, and randomized exposure supplies the arm attribution lacks, so the contribution of an unretrieved document becomes estimable at the cost of displacing some ranked evidence while estimates are collected.

Data valuation and content compensation. Data valuation assigns each data source a measure of its contribution to a model's behavior, and as generative search replaces ranked links with synthesized answers, it is increasingly proposed as the basis for compensating the providers of that content. Data Shapley established valuation for training data (Ghorbani and Zou, 2019). Recent work extends it to documents supplied as context at inference time, valuing the sources of LLM-generated summaries with clustered Shapley approximations (Ye and Yoganarasimhan, 2025) and adapting payment rules from mechanism design to retrieval-augmented data markets (Fan et al., 2025). These methods value the documents a retriever surfaces. Under leave-one-out and mechanism-design payments only retrieved documents are compensated (Fan et al., 2025), so a document the ranker never places in context receives nothing regardless of what it would have contributed. The undervaluation is systematic: it tracks retrieval rank, and a document that is never retrieved generates no evidence that it should be. Theorem 1 explains why collecting more logs under the same retriever cannot correct this. CMP's exposure design gives every document in a pool a known, positive probability of entering the context, which separates a document's value from the ranker's assessment of it and makes the value of unretrieved sources identifiable.

Recommender systems and counterfactual evaluation. Recommender systems learn from logged feedback that exists only for items the serving policy chose to show, which makes exposure bias a central problem in the field. Inverse-propensity methods correct for it in learning to rank (Joachims et al., 2017), feedback loops in which unexposed items remain unexposed are well documented (Jiang et al., 2019), and two-stage architectures, in which a candidate-generation stage gates what a ranking stage can evaluate, have required off-policy corrections of their own (Ma et al., 2020). The problem has now reached LLM-based recommenders, where self-normalized inverse propensity scoring corrects exposure bias in continual post-deployment updates (Kim et al., 2026). This literature largely treats an unexposed item as recoverable: an item not shown today can be explored tomorrow, so exposure bias is a problem of estimation efficiency. That assumption fails whenever an action removes an item from future support, as in catalog delisting, inventory retirement, or memory eviction, where a positivity violation becomes permanent. Two of our results carry over. The mediation of Section 2.1 is the two-stage architecture, and identifying an item the candidate stage never passes requires randomizing that stage itself, as Section 3.1 does for retrieval. Theorem 2 characterizes the Bayes decision when an action destroys the ability to re-estimate, which is the decision a platform faces when it retires items on the basis of exposure-biased logs.

Cognitive science of memory. Cognitive psychology has long distinguished whether a memory is stored from whether it can be retrieved. Items a person fails to recall are often still available and can be recovered with an appropriate cue (Tulving and Pearlstone, 1966), a dependence encoding specificity attributes to the match between the retrieval cue and the conditions of storage (Tulving and Thomson, 1973). The theory of disuse formalizes the two as separate quantities, storage strength and retrieval strength (Bjork et al., 1992), and retrieval is competitive, since retrieving some items suppresses related ones (Anderson et al., 1994). Complementary learning systems theory has carried related ideas into models of consolidation (Kumaran et al., 2016; McClelland et al., 2020), but the distinction between storage and access has not informed how artificial memory is valued. Memory systems infer a memory's value from behavior, which conflates the two quantities cognitive science keeps apart. Proposition 1 is their computational form: presence in the store is availability, $P(m \in R \mid q, c)$ is accessibility, and U_R is the value of a memory once accessed. The parallel extends to method. Cognitive science revealed available but inaccessible memories by changing the cue, and randomized exposure plays the same role, a cued-recall manipulation run with known probability. Retrieval

competition has an analogue as well: a deterministic ranker that repeatedly surfaces the same memories deprives the rest of evidence, and utility-ranked eviction then removes them.

Human–AI interaction and longitudinal relationships. Persistent memory turns interaction with an assistant into a relationship that accumulates across sessions. Commercial assistants now retain information about users between conversations and offer controls to inspect and delete it, and interface research has made stored memories visible and editable (Huang et al., 2023). A recent study of ChatGPT's memory feature found that users perceive AI memory as unforgetful, that most experienced violated expectations on learning what it retained, and that they want greater visibility and control (Chen et al., 2026). Appropriate reliance on such a system requires that users can tell when it errs (Schemmer et al., 2023). Transparency interfaces show what a memory system stores, not what it uses, and on LongMemEval the two differ substantially: 54% of required memories are never retrieved. Such a memory appears in a memory viewer yet is functionally forgotten, so the user's belief that the assistant does not forget diverges from its behavior, and the error is unobservable, since the answer the memory would have produced is never shown. The divergence compounds over longitudinal use. A memory that is never retrieved produces no evidence of its value, utility-ranked eviction then deletes it, and information the user chose to share disappears without the system having tested whether it mattered. Proposition 1 distinguishes a stored memory from an accessible one, the distinction an honest memory interface would display, and randomized exposure lets a system learn whether a shared memory helps.

Evaluating long-term memory in LLM agents. The application closest to this work is the evaluation of agent memory itself. LongMemEval (Wu et al., 2024) and LoCoMo (Maharana et al., 2024) established long-horizon recall as the standard test, and a second generation of benchmarks extends it to memory that changes over time: whether agents recognize that a memory is no longer valid (Chao et al.), how they handle forgetting and test-time learning (Hu et al., 2026), and how memory supports agents in environments beyond dialogue (Di Wu et al.). All of these query a system and score its answer, which measures recall for the questions asked. A retention decision depends on something else, a memory's value for questions not yet asked, and a benchmark that queries each store once cannot measure it. LongMemEval does exactly that: every store carries a single question, and among its multi-evidence items none of the 203 required memories is needed by a second question. Our framework specifies what retention evaluation requires, namely several queries against one store with overlapping required evidence, and LoCoMo meets it, with 750 of $1{,}425$ required memories shared across questions. It also supplies a validity check that can be run before any policy is compared: a benchmark whose required memories the evaluated retriever never surfaces measures the retriever, and Assumption 1 can be tested on it directly.