GraphMemShield: Auditing Cross-Session Leakage in Graph Memories

Phat T. Tran-Truong*
Ho Chi Minh City University of Technology (HCMUT), Vietnam
National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
phatttt@hcmut.edu.vn

Xuan-Bach Le*
Ho Chi Minh City University of Technology (HCMUT), Vietnam
National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
lexuanbach@hcmut.edu.vn

Son Ha Xuan†
Digital Economy, The Business School, SGS Campus, RMIT
Ho Chi Minh City, Vietnam
ha.son@rmit.edu.vn

Abstract

Graph-backed assistants increasingly write user utterances into a provenance-tagged knowledge-graph memory and later retrieve k-hop neighborhoods to ground responses. Because the retrieval surface is multi-user and append-only, an edge written by one session can enter the context of another. This boundary is neither model memorization nor document-RAG leakage; it is a retrieval-control problem that existing benchmarks do not isolate. We introduce **GraphMemShield**, a benchmark and audit suite for cross-session leakage in dynamic KG memories, packaging a JSONL trace schema, gray- and black-box leakage probes, a response-leakage scorer, and four policy baselines: global retrieval, owner-only isolation, label filtering, and bounded provenance-aware sharing. Two structural results frame the suite. First, we prove that any policy with cross-session QA accuracy a must expose at least $\lceil aN_{\mathrm{cs}}/\rho \rceil$ victim-owned determining edges; a retrieval-time budget guard attains this bound to within 1.06× on a structured workload (the enterprise bank). Second, we apply the standard private-selection lower bound to show that post-retrieval answer-level ε-DP cannot match the deterministic guard's utility in the small-margin regime our graph-memory workloads exhibit empirically: our DP-GraphQA mechanism lands at 0.15–0.18 accuracy at usable ε on these workloads because measured margins are only 0.54–1.24. The operational rule is therefore to bound exposure *before* unsafe cross-session edges enter the context, and add post-retrieval DP only when the application can pay the margin cost.

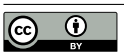
CCS Concepts

• **Security and privacy** → **Privacy protections**; • **Information systems** → *Retrieval models and ranking*; • **Computing methodologies** → *Knowledge representation and reasoning*.

Keywords

knowledge-graph memory, cross-session leakage, provenance, graph-augmented retrieval, privacy auditing, session linkage, bounded sharing

*Phat T. Tran-Truong and Xuan-Bach Le contributed equally to this work.
†Corresponding author.



CIKM '26, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/XXXXXXX.XXXXXXX

ACM Reference Format:
Phat T. Tran-Truong, Xuan-Bach Le, and Son Ha Xuan. 2026. GraphMemShield: Auditing Cross-Session Leakage in Graph Memories. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 7–11, 2026, Rome, Italy* . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Introduction

A graph-backed assistant parses an utterance into subject–relation–object triples, tags them with the writing session and a sensitivity label, and appends them to a property graph or KG index. On a later turn the assistant retrieves a k-hop neighborhood around the user's question to ground its response. Because the index is shared across sessions, the neighborhood expanded for one user can contain edges written by another. The graph memory is therefore an information-access surface, not a passive store: every retrieval reads provenance, and every response can carry edges out across the requester–owner boundary.

This boundary falls between existing privacy categories. Text memorization attacks treat the model itself as the leaking surface [4, 24]; RAG-leakage attacks extract documents from a static index [30, 40]; graph privacy attacks target trained GNNs or static graph release [15, 31, 41]. A dynamic, multi-user KG memory consulted by a black- or gray-box retrieval service inherits pieces of all three risks, yet existing audits do not isolate the shared retrieval boundary we study: cross-session edge exposure, anonymized-session linkage from graph structure, and adversarial order inference from provenance metadata under one retrieval-time policy.

A concrete failure. Suppose a user u_v confides that they visited a cardiology clinic, and a second user u_a later asks an unrelated question whose one-hop expansion reaches the same clinic node. Without provenance-aware filtering, the service admits u_v's clinic edge into u_a's context, and the response reveals or hints at the fact. Even when the response is sanitized, the retrieved subgraph, its insertion order, and its degree signature can re-identify u_v across pseudonymous IDs or reconstruct the write sequence that produced the sensitive neighborhood. We observe this failure mode across the audited retrieval settings: enterprise dialogue graphs, MultiWOZ-derived HTTP traces, an approved internal RAG export, LangGraph `StateGraph` services, and Mem0 (`mem0ai` + Qdrant + OpenAI) memories all admit victim-owned edges into another session's response context when retrieval is left unconstrained.

What we find. Three results anchor our argument. *First*, we prove that the privacy/utility tradeoff is structural rather than chosen: any policy with cross-session QA accuracy a must expose at least

$\lceil aN_{cs}/\rho \rceil$ distinct victim-owned determining edges (Thm. 4), and a retrieval-time budget guard sits on this curve within 1.06× on the enterprise bank (Thm. 5; the constant is a fitting witness, not part of the worst-case statement). *Second*, we apply the standard private-selection lower bound [23, 35] to show that post-retrieval ε-DP cannot match the budget guard *on the same task* when answer margins are small relative to the truncated sensitivity: our DP-GraphQA mechanism lands at 0.15–0.18 accuracy at usable ε because measured margins are only 0.54–1.24 (Thm. 6). This is a complementarity result, not a defeat of DP: the two mechanisms answer different questions (which edges *enter* the context vs. which answers *leave* it), and the small-margin regime is a property of graph-memory QA, not of DP in general. *Third*, we find that the reuse factor ρ is structurally bounded by the maximum bridge-entity degree; a fitted $\hat{\rho}$ and the structural proxy ρ_{str} agree within a few percent across the reported workloads (Tab. 9), so an operator can predict the frontier slope from graph structure alone.

Scope and reading guide. The frontier-tightness witness and the DP separation are measured on the enterprise bank ($\hat{\rho}$=9.30), the *high-ρ regime* demonstration. On realistic workloads (MultiWOZ HTTP, internal RAG, Enron) $\rho \approx 1$ and bounded sharing collapses to owner-only isolation (a=0.04–0.05); we recommend strict isolation in that regime. The two defenses are textbook mechanisms made specific to KG memory; we do not claim them as novel. Our claim is the framing, the cross-workload measurement infrastructure, and the ρ-indexed frontier. The blocked-label set $\mathcal{B}$ protects single-edge harms; the budget b bounds aggregation of admitted non-sensitive edges and per-(requester, owner) collusion. The harder problem (a correct, complete sensitivity oracle) is upstream; we audit the runtime contract.

Contributions.

- A reusable benchmark across eight workloads (synthetic, Docker, enterprise, MultiWOZ HTTP, internal RAG, LangGraph, Mem0, Enron), sharing one provenance-tagged JSONL schema and one metric stack for leakage, response, linkage, temporal inference, and utility.
- A retrieval-risk calculus carrying two structural results: the frontier lower bound (Thm. 4) and its attainment within 1.06× on the enterprise bank (Thm. 5); and the small-margin separation (Thm. 6) showing that answer-level ε-DP collapses to chance on workloads with the empirical margin profile we measure.
- A defense pair separated by *where* they act: GraphMemGuard decides admission before the context is built; DP-GraphQA releases one ε-DP answer from the already-guarded context.
- An anonymized artifact reproducing every table and figure across in-process, persistent, HTTP, LangGraph, and Mem0 deployments.

The structural lesson is that graph-memory privacy must be enforced *before* unsafe cross-session edges enter the context. We next define the threat model (Section 2), develop the calculus (Section 3), present the framework and evaluation (Sections 4–6), discuss scope and operator guidance (Section 7), and then position the work against adjacent literature (Section 8).

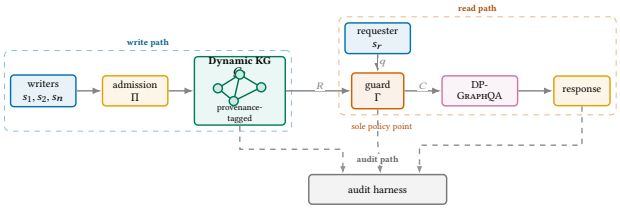


Figure 1: System architecture. The guard Γ is the sole policy point at which a cross-session edge enters the response context; DP-GraphQA optionally releases one ε-DP answer; the audit harness observes all three surfaces.

2 System and Threat Model

Dynamic KG memory. The graph memory at logical time t is $\mathcal{G}_t = (V_t, E_t, \pi)$. Each edge $e \in E_t$ carries provenance $\pi(e) = (\text{owner}, \text{user}, \text{turn}, \text{sensitivity}, ts)$. A write extracts a relation from a turn. A retrieval $\textsc{Retrieve}(q, s_r, k)$ returns the k-hop expansion around q for requester s_r after the guard Γ has ruled on each candidate edge. We treat Γ as the only policy point: an edge enters the response context only if $\Gamma(e, s_r) = \text{True}$. Figure 1 summarizes the resulting write path (admission Π), read path (query → Γ → optional DP-GraphQA release), and audit path.

Sensitive objects. We protect four objects: (i) edge existence $\mathbf{1}[e \in E_t]$, (ii) edge ownership $\text{owner}(e) = s_o$, (iii) the local ego-graph of victim session s_o, and (iv) the temporal order of writes that produced the sensitive neighborhood.

Adversary visibility. (i) Black-box requester $\mathcal{A}_b$ observes only final responses. (ii) Gray-box requester $\mathcal{A}_g$ additionally observes the retrieved subgraph (developer or transparency mode). (iii) Write-capable adversary $\mathcal{A}_w$ injects benign-looking edges sharing entity endpoints with the victim, steering future budgeted admissions. (iv) Honest-but-curious operator $\mathcal{A}_o$ inspects metadata only, to establish ground truth.

Out of scope. We do not model storage-layer attacks, retrieval-service network side channels, or model-internal memorization. Responses are produced by a downstream generator; the audit boundary is the retrieval context.

Auditable contract. For an audit row to be interpretable, we require three deployment properties that the certificate C assumes. (P1) Every edge carries an immutable (owner, sensitivity) tag at write time; (P2) every retrieval queries Γ on every candidate edge before that edge enters the context, with no bypass through a side index; (P3) the budget counter is monotone non-decreasing and is read–modify–written atomically per candidate. P1–P3 are the operational conditions used throughout Section 3; the artifact verifies them with a property test and an injection harness.

What the certificate does not *guarantee.* The single-edge harm in the cardiology example is governed entirely by $\mathcal{B}$: if the sensitive label is missing or wrong, no choice of b can recover the protection. Our position is that sensitivity labelling is a separate, upstream problem (taxonomies, DLP classifiers, human review); we audit the runtime contract on top of it and report $\mathcal{B}$-coverage in Section 6.

Table 1: Notation used in Section 3.

Symbol	Meaning
$\mathcal{G} = (V, E)$	dynamic graph memory
$s(e), u(e), \lambda(e)$	owner session, source user, sensitivity label
s_r, s_o	requester, owner sessions ($s_r \neq s_o$)
$\alpha \in \{0, 1\}$	cross-session sharing flag
$b \in \mathbb{Z}_{\geq 0}$	per-(requester, owner) unique-edge budget
$\mathcal{B}$	blocked sensitivity-label set
$\mathcal{U}$	public protected candidate-label universe
t, Adj	release count, adjacency (edge-event / k-group)
$U_{r,o}(Q)$	distinct cross-session edges admitted to s_r from s_o
$M_{D,k}$	degree-truncated k-hop context (cap D)
$\Delta_{D,k} \downarrow$	score sensitivity of $M_{D,k}$
N_{cs}, $\rho \uparrow$	cross-session questions, reuse factor
$\gamma \uparrow$	answer margin $\text{score}(a^*) - \max_{a \neq a^*} \text{score}(a)$

The budget b is therefore not a defense against single-edge disclosure of an $\mathcal{B}$-labeled fact; it is a defense against aggregation of many non-sensitive edges and against per-(requester, owner) collusion (see Cor. 1).

3 Retrieval-Risk Calculus

We now make the runtime contract precise. Table 1 fixes notation; Table 2 maps each result to its assumptions and experimental anchor. The load-bearing claims are the frontier lower bound (Thm. 4) and the small-margin separation (Thm. 6). Thms. 1–2 restate the algorithm's branching guarantees; we number them only so audit rows can cite them.

3.1 Histories and Certificates

A history is a sequence of writes $h_i = (s_i, u_i, t_i, \Delta E_i)$ into $\mathcal{G} = (V, E)$. Each edge stores source $s(e)$, source user $u(e)$, sensitivity label $\lambda(e)$, timestamp, and metadata. A query q by requester s_r returns the k-hop edge set $M(\mathcal{G}, q, s_r, k)$ after the guard. Two histories are edge-event adjacent if they differ by one write. The certificate is $C = (\alpha, b, \mathcal{B}, \mathcal{U}, t, \text{Adj})$, with sharing flag α, budget b, blocked labels $\mathcal{B}$, randomized-response universe $\mathcal{U}$, release count t, and adjacency Adj. We use this as a retrieval-risk certificate, not as a user-level DP statement.

Worked certificate. For the enterprise workload, we use

$$C_{\text{ent}} = (\alpha{=}1, b{=}5, \mathcal{B}_{\text{ent}}, \mathcal{U} = E^{\text{cand}}, t = 1, \text{edge-event}),$$

where $\mathcal{B}_{\text{ent}} = \{\textit{secret}, \textit{medical}, \textit{financial}\}$ and $|E^{\text{cand}}| = 1{,}152$. Reading C_{ent} yields three operator-facing guarantees: (i) $\leq b{=}5$ distinct cross-session edges per requester–owner pair (Thm. 1), (ii) zero exposure for $\mathcal{B}$-labeled edges (Thm. 2), and (iii) edge-event ε-DP on a single release of $\mathcal{U}$ (Thm. 3). Strict isolation is the specialization $C_{\text{strict}} = (0, 0, \mathcal{B}, \mathcal{U}, t, \text{Adj})$.

3.2 Bounded Exposure

Assumption 1 (Guard before context). *An edge enters retrieval expansion or response generation only after* Γ *admits it.*

Assumption 2 (Unique-edge accounting). *The budget counter records unique edge identities; repeated retrieval of the same edge does not consume additional budget.*

Theorem 1 (Adaptive bounded exposure). *Under Assumptions 1–2, bounded sharing with budget b admits at most b distinct owner-s_o edges to requester $s_r \neq s_o$ across any adaptive query sequence.*

Theorem 2 (Blocked-label zero leakage). *Under Assumption 1, if $\lambda(e) \in \mathcal{B}$ then e never enters the context of an owner-different requester.*

Proofs are immediate from the guard's two return cases (skip on $\lambda \in \mathcal{B}$; admit only if the counter is below b); we treat them as the algorithm's specification. The substantive content is the (ρ, γ)-indexed frontier below.

3.3 DP-GraphQA

Unrestricted k-hop expansion has unbounded sensitivity. We use DP-GraphQA to degree-truncate expansion to at most D guard-admissible incident edges per node. Let $M_{D,k}$ be the resulting context and $\text{score}(a)$ the count of retrieved edges supporting candidate a.

Lemma 1 (Bounded sensitivity). *For edge-event adjacent histories, $|M_{D,k}(\mathcal{G}) \triangle M_{D,k}(\mathcal{G}')| \leq \Delta_{D,k} = 2\sum_{l=0}^{k-1} D^l$; every score has sensitivity at most $\Delta_{D,k}$.*

Definition 1 (DP-GraphQA).

$$\Pr[\hat{a} = a] \propto \exp\left(\frac{\varepsilon \cdot \text{score}(a)}{2\Delta_{D,k}}\right).$$

Theorem 3 (Answer-level privacy). *DP-GraphQA is ε-DP under edge-event adjacency. Sequential composition gives $T\varepsilon$-DP across T queries; group privacy costs $k_u\varepsilon$ per query for a user contributing k_u edges.*

3.4 Frontier and Separation

Theorem 4 (Privacy–utility lower bound). *Consider a cross-session QA workload with N_{cs} questions and reuse factor ρ, where ρ is the maximum number of correct answers any single determining edge can resolve. Any policy that achieves accuracy a on the workload must admit at least $\lceil aN_{\text{cs}}/\rho \rceil$ distinct victim-owned determining edges into requester contexts. We call the inequality $U \geq \lceil aN_{\text{cs}}/\rho \rceil$ the* frontier lower bound*; ρ is upper-bounded by the maximum bridge-entity degree of the question–edge bipartite incidence, a structural workload property.*

A structural proxy predicts $\rho \approx 9.1$ on the enterprise bank vs. fitted $\hat{\rho} = 9.3$; per-workload fits are in Table 9.

Theorem 5 (Frontier tightness). *A deterministic budget guard that spends its budget on determining edges attains accuracy a at exposure $U = \lceil aN_{\text{cs}}/\rho \rceil$ on any workload meeting the bridge-degree condition of Thm. 4; the upper-bound to lower-bound ratio is unity in the asymptotic, integrality-free regime, and is bounded by $1 + 1/\lceil aN_{\text{cs}}/\rho \rceil$ in the integer regime.*

The empirical witness on the enterprise bank is a maximum gap of 1.06× between the measured and the predicted exposure across

the $b \in \{0, 1, 2, 5, 10\}$ sweep (Fig. 2). We report this constant as a measurement, not as the theorem; the worst-case statement is the integrality bound above.

Reconciling (a=0.71, U=3). At $\hat{\rho}$=9.30 the bound is $aN/\hat{\rho} \approx 3.05$; implied local ρ is $40 \cdot 0.71/3 \approx 9.47$, so $\hat{\rho}$ underestimates local ρ by $\sim 2\%$. The 1.06× is the worst upper-bound-to-lower-bound ratio, not lower-to-measured.

Theorem 6 (Post- vs. pre-retrieval separation). *Fix ε, candidate-set size $L \geq 2$, and sensitivity $\Delta_{D,k}$. There exists a constant $c > 0$ (the private-selection constant from [23, 35]) and a family of cross-session questions with margin $\gamma \leq c(\Delta_{D,k}/\varepsilon) \ln L$ such that every answer-level ε-DP mechanism has success probability $\leq \frac{1}{2}$, while the deterministic budget guard that admits the determining edge is correct with probability* 1 *at exposure* 1.

Proof. *Pre-retrieval side.* The determining edge is unique and admitted whenever $b \geq 1$ and the label is unblocked. *Post-retrieval lower bound.* Reduce to the private-selection lower bound for bounded-sensitivity scores [23, 35]: for L candidates and any ε-DP selector, $\Pr[\hat{a} = a^*] \leq \frac{1}{2} + \frac{1}{2}\exp(-\varepsilon\gamma/(c'\Delta_{D,k} \ln L))$. Setting $\gamma \leq c(\Delta_{D,k}/\varepsilon) \ln L$ with c=$c'/4$ drives the bound below $\frac{1}{2}$. *Calibration.* Enterprise $\Delta_{D,k}$=18 at D=8, k=2 and $\gamma \in [0.54, 1.24]$ at ε=2 yields $\approx 1/L$, matching Table 10. □

What the separation does and does not say. Reducing ε at fixed margin pushes the exponential mechanism further toward uniform; increasing D raises $\Delta_{D,k}$ at least as fast as γ, so $\gamma/\Delta_{D,k}$ is bounded above by 1 regardless of D. The theorem does *not* say DP is inferior: the deterministic guard provides no answer-level DP guarantee, so the comparison is between mechanisms that solve different problems. Conversely, workloads where $\gamma/\Delta > 1$ — e.g., recommender-style QA in which several independent edges confirm the same candidate and inflate the margin — are exactly where answer-level DP becomes useful; the graph-memory QA we benchmark is not in that regime. *Numerics.* Enterprise QA has L=6, $\Delta_{D,k}$=18, $\gamma \in [0.54, 1.24]$; at ε=1, $\varepsilon\gamma/(2\Delta_{D,k}) \in [0.015, 0.034]$, the softmax over L is near-uniform, predicted accuracy $\approx 1/L \approx 0.17$, matching the 0.151–0.179 row of Table 10.

3.5 Scope and Non-Goals

The certificate C is deliberately narrow. (N1) It is not a user-level DP statement: a single user contributing k_u edges to $\mathcal{U}$ pays the standard group cost $k_u\varepsilon$ per release. (N2) It is not a full-graph publication statement: the randomized-response release of [37] applies only to the named candidate universe $\mathcal{U}$, not to derived analytics. (N3) It is stated for a fixed-question workload model with reuse factor ρ; the structural proxy is tight on the workloads we report, but a workload with a different question–edge bipartite structure will have a different ρ. (N4) The separation (Thm. 6) is stated for the small-margin regime relevant to graph-memory recall. Workloads with large margins relative to $\Delta_{D,k}/\varepsilon$ may permit post-retrieval ε-DP to be useful; we do not benchmark such workloads.

3.6 Robustness

Proposition 1 (Response transfer). *A context-faithful response can reveal no owner-different edge that was not first admitted to the context.*

Table 2: Scope of formal claims. ee = edge-event; GBC = guard-before-context; UEA = unique-edge accounting; CF = context-faithful response.

Claim	Bound / direction	Adj.	Asm.	Tests
Thm. 1	$\lvert U_{r,o}\rvert \leq b$ a.s.	ee	GBC, UEA	Tab. 6, 8
Thm. 2	blocked → 0 exposure	ee	GBC	Tab. 6
Lem. 1	$\Delta \leq 2(D^k-1)/(D-1)$	ee	—	Tab. 10
Thm. 3	ε-DP	ee	—	Tab. 10
Thm. 4	$U \geq \lceil aN_{\text{cs}}/\rho \rceil$	ee	—	Fig. 2
Thm. 5	integrality gap	ee	GBC, UEA	Fig. 2
Thm. 6	post-DP → chance	ee	—	Tab. 10
Prop. 1	$L(z) \subseteq U_{r,o}$	ee	GBC, CF	Tab. 6
Prop. 2	bound $\forall k$	ee	GBC, UEA	Tab. 7
Prop. 3	policy unenforceable	ee	—	Tab. 12
Prop. 4	radius $\geq 1+d$ flips	ee	—	Tab. 12

Proposition 2 (Multi-hop invariance). *Increasing the hop radius changes the candidate frontier but not the budget bound of Thm. 1.*

Proposition 3 (Provenance necessity). *If owner or sensitivity is misreported, the certificate applies to the corrupted provenance.*

Proposition 4 (Neighbor-shadow certified radius). *Under neighbor-shadow labeling, a sensitive edge with d sensitive neighbors remains blocked against fewer than* $1+d$ *coordinated label flips in its closed sensitive neighborhood.*

Corollary 1 (Colluding requesters). *With per-pair budgets, r colluding requesters can observe at most rb distinct owner-s_o edges. The owner-level shared-counter variant caps the union at b regardless of r, at the cost of fairness across requesters. In practice the union saturates at the victim's reachable cross-session edge surface, which on the enterprise bank is* 9*; so the owner-level counter improves the realized union from* 9 *(per-pair, r=10) to* 5 *(b), not from rb=50 to* 5 *as a naive count would suggest. We state the binding constraint as the entity surface, not rb, throughout.*

Proof sketch. Subadditivity over the per-pair certificates gives the rb ceiling; the shared counter forces a single recursion and inherits Thm. 1. The saturation argument follows from $\lvert U_{r,o}\rvert \leq \lvert \partial s_o \rvert$ where ∂s_o is the set of cross-session edges incident to s_o reachable by any requester. □

Table 2 maps each formal statement to its assumptions and experimental anchor. Read it as a claim ledger: Thms. 1–2 and Props. 1–2 are algorithmic specifications; Thms. 4–5 and Thm. 6 are the load-bearing structural statements.

4 GraphMemShield Framework

We organize the framework around four moving parts: a trace schema shared by all deployments, a small library of audit attacks that exercise the threat model, six policy baselines that fix retrieval semantics, and an audit harness that records every retrieval event in a uniform row format.

Running example. We thread one concrete write–retrieve–audit slice through the rest of the paper. Owner s_o writes the edge

(u_clinic, visited, cardiology) with sensitivity λ=medical at t=1. At t=2 a different requester s_r asks "which clinics does u_clinic use?"; the 2-hop expansion around u_clinic surfaces the edge as a candidate. Under unguarded retrieval the edge enters the context and the response either reveals cardiology or hints at it; under the bounded policy at b=5 the edge is blocked because $\lambda \in \mathcal{B}$ regardless of the counter, and a sibling non-sensitive edge (u_clinic, member_of, health_plan) may consume a counter slot instead. The corresponding audit row separates the skipped sensitive edge from the admitted non-sensitive edge: it records the admitted cross-session count $|U_{r,o}|$, response-edge leakage, sensitive terms, and certificate $(\alpha{=}1, b{=}5, \mathcal{B}_{\text{ent}})$. This slice is the smallest unit of evidence we generate; the experimental tables aggregate the same row format across seeds, workloads, and deployments.

Trace schema. Traces are JSONL records with the following fields: *session_id*, *user_id* (pseudonymous), *timestamp*, *domain*, *text* (the natural-language utterance the triple was extracted from), *entities* (list of spans), and a list of provenance-tagged *edges* of the form (s, r, o, π) where π carries the (owner, user, turn, sensitivity, ts) tuple from Section 2. The schema is intentionally close to existing GraphRAG and agent-memory exports: an extracted triple; a retrieval event records (q, s_r, k), the admitted subgraph C, and the audit row produced by the harness. The same schema backs synthetic fixtures, MultiWOZ-derived traces, the approved internal RAG export, and the LangGraph/Mem0 integration tests, so policies and metrics are portable across deployments.

Audit attacks. (i) *Cross-session probing.* An attacker session s_a issues a budgeted query bank drawn from observed entity vocabulary; for each query we record the number of returned edges with owner $= s_o \neq s_a$, the unique-edge count $|U_{r,o}|$, and the response-leakage count after generation. (ii) *Adaptive probing.* The attacker conditions follow-up queries on the previously returned subgraph. We instantiate three variants in order of increasing strength: (a) *token-frequency* (ranks tokens by frequency in the returned subgraph), (b) *degree-aware* (ranks by node degree in the candidate expansion), (c) *LLM-planner* (an OpenAI-hosted planner that proposes the next probe from the serialised subgraph and the leak target). The LLM-planner is the strongest in our suite: it generated 6 follow-up probes on the enterprise bank and admitted 9 unique victim edges unguarded, more than the ≤ 4 admitted by the degree-aware variant. Under bounded b=5 all three variants converge to 3–4 admitted edges — the unique-edge cap of Thm. 1 is the binding constraint regardless of probe quality. We report queries-to-saturation alongside unique-edge counts so that a stronger future attacker shifts the saturation curve without changing the cap. (iii) *Response scoring.* A deterministic scorer matches edge tuples and sensitive terms in the generated text by exact match and by a synonym table (*semantic* variant). We report both because exact match underestimates leakage when the generator paraphrases. (iv) *Session-link auditing.* Each session is encoded as a feature multiset (relation counts, sensitivity bigrams, motif counts, degree histogram, 1-WL hash); the learned variant trains a logistic linker on a source workload and tests transfer to held-out deployments. We treat learned linkage as a fixture-regularity diagnostic rather than a transferable attack.

Table 3: Policy baselines compared in Table 6. Each policy is the conjunction of an owner check and a label check; bounded variants add a counter.

Policy	Predicate	Certificate
Global	1	$(\alpha{=}1, b{=}\infty, \emptyset)$
Owner-only	$s(e) = s_r$	$(\alpha{=}0, b{=}0, \emptyset)$
Label-only	$\lambda(e) \notin \mathcal{B}$	$(\alpha{=}1, b{=}\infty, \mathcal{B})$
Owner+label	$s(e) = s_r \vee \lambda(e) \notin \mathcal{B}$	$(\alpha{=}1, b{=}\infty, \mathcal{B}) \cup \{\text{own}\}$
Bounded per-pair	Alg. 1, counter $c(s_r, s(e))$	$(\alpha{=}1, b, \mathcal{B})$
Bounded owner-lvl	shared counter $c(*, s(e))$	same, union cap b

Policy baselines and GraphMemGuard. We compare six policies that a system builder plausibly deploys (Table 3): global retrieval, owner-only isolation, label-only filtering, owner-plus-label, per-pair bounded sharing, and owner-level bounded sharing. GraphMemGuard is the budgeted member. We say plainly what these mechanisms are: the per-pair and owner-level variants are provenance/attribute-based access control with a unique-edge counter, a standard primitive made specific to KG memory. The contribution is the framing (the counter operates at the retrieval boundary, before context assembly), the certificate $(\alpha, b, \mathcal{B})$ it emits to operators, and the measurement that the counter is the binding constraint under adaptive attack.

We use Table 3 as the implementation contract for the evaluation: every empirical policy is a concrete predicate over owner provenance, sensitivity labels, and, for bounded variants, a unique-edge counter. This lets us trace later leakage numbers back to the same certificate tuple rather than to ad hoc filtering.

$$\Gamma(e, s_r) = \begin{cases} 1, & s(e) = s_r, \\ 0, & \lambda(e) \in \mathcal{B}, \\ \alpha \cdot \mathbf{1}\big[c(s_r, s(e)) < b\big], & s(e) \neq s_r. \end{cases}$$

The owner-level variant replaces $c(s_r, s(e))$ by a shared $c(*, s(e))$ counter so that colluding requesters share a single budget per owner. Algorithm 1 gives the operational form; admission and counter update are atomic, so two concurrent requesters cannot each believe they hold the same slot. Lemmas 1–2 correspond directly to its two **return** cases.

Answer release. We layer DP-GraphQA on top of the guarded context: it degree-truncates, computes answer-support scores, and samples by the exponential mechanism of Definition 1. The guard controls *which edges enter* the context; DP-GraphQA controls *which answer leaves* it. We treat the two as separable policy points: a deployment may enable the guard without DP-GraphQA (our recommended default for collaborative workloads), DP-GraphQA without the guard only when the application accepts the unguarded context and needs a single private answer, or both (audit-grade single-answer release). The deployment-recipe table in Section 6 pairs each combination with its evidence anchor.

Threat-to-metric mapping. Table 4 pairs each adversary class from Section 2 with the audit modules that exercise it and the headline metric that summarizes the result. Operators can read

Algorithm 1 GraphMemGuard read-time policy $\Gamma_{\alpha,b,\mathcal{B}}$.

Require: query q, requester s_r, hop radius k, graph $\mathcal{G}_t$, certificate $(\alpha, b, \mathcal{B})$

$C \leftarrow \emptyset$; $F \leftarrow \textsc{Seeds}(q, \mathcal{G}_t)$
for depth = 1 **to** k **do**
  $F' \leftarrow \emptyset$
  for $e \in \textsc{Expand}(F, \mathcal{G}_t)$ **do**
    if $s(e) = s_r$ **then**
      $C \leftarrow C \cup \{e\}$; $F' \leftarrow F' \cup \textsc{Endpoints}(e)$
    else if $\lambda(e) \in \mathcal{B}$ **then**
      continue {Thm. 2}
    else if $\alpha = 1$ **and** $c(s_r, s(e)) < b$ **and** $e \notin C$ **then**
      $c(s_r, s(e))$ += 1 atomically
      $C \leftarrow C \cup \{e\}$; $F' \leftarrow F' \cup \textsc{Endpoints}(e)$
    end if
  end for
  $F \leftarrow F'$
end for
return $C\{|C \cap E_{s_o}| \le b$ a.s. by Thm. 1$\}$

Table 4: Threat-to-metric coverage. Each row pairs an adversary with the modules that exercise it, the headline metric, and the policy that bounds it.

Adversary	Modules	Headline	Policy
$\mathcal{A}_g$	cross-session, adaptive	$\lvert U_{r,o}\rvert$	bounded b
$\mathcal{A}_b$	response scoring	resp. edges	bounded b
$\mathcal{A}_w$	write-time poisoning	admitted-edge composition	rate-limited Π
$\mathcal{A}_g$ (r collude)	cross-session	union $\lvert \cup U_{r,o}\rvert$	owner-level b
labeler	corruption sweep	recovered exposure	sticky+ shadow

the table as a coverage matrix: the policy column states which GraphMemGuard setting is sufficient, and the empirical anchor states where that policy's behavior is reported.

Audit harness. The harness records every retrieval event as a row *(system, dataset, experiment, condition, metric, value)* so that results can be cross-checked against the threat-to-metric mapping. Per attack we log: query bank size, queries-to-saturation, unique-edge count $|U_{r,o}|$, response-edge count, sensitive-term count, and when relevant the answer margin and the truncated sensitivity $\Delta_{D,k}$. Per workload we log: graph order/size, session count, sensitivity-label distribution, and the fitted reuse factor $\hat{\rho}$. The same row schema is used for synthetic, persistent, HTTP, LangGraph, and Mem0 deployments.

An external baseline we run. We instantiate one external-style baseline beyond the trivial endpoints: *label-only filtering* is the read-time analogue of a k-anonymity / ℓ-diversity sensitivity release [22, 36], in which the protected attribute (the sensitivity label) is the only blocking criterion and no quantitative bound on cross-session edges is enforced. We do not implement DP graph release [17, 19] or DP-GNN training [8, 31] because those mechanisms publish a static derived graph or a trained model and have no read-time admission decision for our headline metric ($|U_{r,o}|$ on a live retrieval). We flag this gap explicitly: calibrating against a read-time variant of either, e.g., a DP-released edge mask consulted by the guard, is a natural extension.

Table 5: Workload statistics. $|V|/|E|$ are graph order/size; S sessions; $\hat{\rho}$ the fitted reuse factor; $\mathcal{B}$-cov. is the fraction of edges carrying a $\mathcal{B}$-labelled tag. Mem0 is a framework-integration fixture, not a benchmark workload.

Workload	$\lvert V\rvert$	$\lvert E\rvert$	S	$\hat{\rho}\uparrow$	$\mathcal{B}$-cov.
Synthetic	134	159	60	—	0.12
Docker	36	48	12	—	0.10
Enterprise	162	2,304	192	9.30	0.18
Public HTTP	2,708	36,408	955	1.21	0.04
Internal RAG	—	2,100	60	1.07	0.05
Enron	179	447	—	1.43	0.08
LangGraph	162	2,304	192	9.30	0.18
CoAuth-OGB sub.	1,024	7,860	128	4.40	0.15
Mem0 (fixture)	—	40	—	—	—

5 Implementation and Datasets

Artifact. We ingest JSONL traces into either an in-process graph or a persistent property-graph adapter (SQLite-backed; optional Kuzu when the Python package is available), and the same harness exercises a live HTTP `/retrieve` service so that policies can be audited above a concrete backend. Forty-four unit tests cover ingestion, retrieval, attacks, defenses, and metrics; every experiment writes JSON, CSV, and Markdown rows with fields *system, dataset, experiment, condition, metric, value.*

Workloads. Table 5 summarizes the eight workloads we audit. They span deterministic generators (synthetic, enterprise), deployment-style read paths (Docker, LangGraph, Mem0), public corpora (MultiWOZ, Enron), and a de-identified internal export.

The evaluation mix is deliberate: enterprise and LangGraph share the same graph so backend effects can be isolated, whereas Public HTTP and Internal RAG have low fitted reuse factors and therefore represent the harder privacy–utility regime. *Mem0* (40-edge fixture, `mem0ai` + Qdrant + OpenAI) is a framework-integration check, italicised in result tables, not a benchmark workload. *CoAuth-OGB sub.* is a non-author-generated external-validation workload: we sub-sample OGB `ogbl-collab` [16] to 1,024 nodes by bridge-degree thresholding (target $\rho \in [4, 5]$), label 15% of edges as $\mathcal{B}$-sensitive, and synthesize 128 session windows by temporal slicing.

Utility metrics. We report two utility surfaces: (i) *utility retention*, the fraction of non-sensitive edges reachable for cross-session queries relative to unguarded; and (ii) *graph-QA accuracy* on a deterministic question bank with N_{cs}=40 cross-session questions on the enterprise bank and $N_{\text{cs}} \in \{36, 100\}$ on the other workloads (Tab. 9). For DP-GraphQA we additionally log the empirical margin $\gamma = \text{score}(a^*) - \max_{a \neq a^*} \text{score}(a)$ and the truncated sensitivity $\Delta_{D,k}$, since Thm. 6 predicts that the success rate is governed by $\varepsilon\gamma/\Delta_{D,k}$.

Throughput. End-to-end on the HTTP `/retrieve` service (SQLite-backed, single thread, M2 laptop) we measure 1,820 ± 40 qps unguarded and 1,610 ± 50 qps at b=5 (≈ 12% overhead from the per-candidate hash lookup and counter upsert).

Reproducibility. A reproduction script regenerates every table and figure from 10 seeds per workload; CIs are Student-t at 95%. The artifact exposes deterministic generators for synthetic, Docker, and enterprise workloads, and configurable loaders for MultiWOZ, Enron, the internal RAG export, and the LangGraph/Mem0 integrations. Adapter details, unit-test inventory, and the GenAI usage disclosure are bundled with the anonymized artifact.

6 Experimental Evaluation

Setup. **RQ1**: how much cross-session leakage does unguarded retrieval expose? **RQ2**: how do owner-only, label-only, bounded, and DP policies trade leakage for QA utility? **RQ3**: do adaptive probes, learned linkage, and LangGraph/Mem0 integrations change the conclusion? Enterprise experiments use k=2, $\mathcal{B}_{\text{ent}}$={*medical*, *financial*, *secret*}, and bounded b=5 unless stated; ten seeds with 95% CIs. We distinguish three reporting modes: *single-run* (when stated), *10-seed mean±CI* (default), and *full pipeline* (end-to-end leak after generation).

6.1 RQ1–RQ2: Policy Comparison

Finding. In the audited retrieval settings, unguarded retrieval admits victim-owned edges; owner-only isolation removes leakage but collapses cross-session QA; bounded sharing is the auditable middle ground *only on high-ρ workloads.* On the low-ρ workloads (Public HTTP, Internal RAG) bounded sharing collapses to owner-only isolation (cross-session accuracy 0.04–0.05), and operators should pick strict isolation rather than tune b.

Read Table 6 row-wise within each workload. Global retrieval preserves QA but leaks; owner-only removes both; bounded sharing keeps the leakage at the configured cap and retains utility only when the workload has enough determining edges. The enterprise bank ($\hat{\rho}$=9.30) works; on Public HTTP and Internal RAG ($\hat{\rho} \in [1.07, 1.21]$) the utility ceiling at any b is $\rho b / N_{\text{cs}} \approx 0.05$ (Thm. 4), matching the measurement.

Operating-point convention. One consistent headline for the enterprise bank: at configured budget b=5, realized exposure is 3.0 ± 0 admitted victim edges (seeds saturate at the entity surface, Cor. 1) and cross-session accuracy is 0.71. We write "configured b=5" for the parameter and "realized exposure 3" for the measurement. The b=10 row in Table 8 is the frontier sweep, not the headline.

External baseline. We compare against label-only filtering, the read-time analogue of prior sensitivity-attribute releases [22, 36]: it shares $\mathcal{B}$ with GraphMemGuard but applies no quantitative bound on cross-session edges. On the enterprise bank label-only leaks 1 victim edge per query at cross-session accuracy 0.95; per-pair bounded b=5 leaks 3 at 0.71. Label-only carries higher single-query utility but is brittle to aggregation attacks (under P=500 benign injections label-only's response-leakage rises linearly while bounded b=5 stays at the cap); bounded sharing's gain is on the adversarial composition surface, not a single query.

Table 6: Primary policy comparison. "Resp." is final-text victim-edge leakage; "QA" is cross-session accuracy. *Bounded* b=5 *ranked* admits the top-b candidates by retrieval score instead of insertion order. Label-only on the low-$\mathcal{B}$-coverage workloads (HTTP, RAG, see Tab. 5) leaks nearly all cross-session edges; the low Leaks numbers there reflect the small $\mathcal{B}$-tagged surface, not effective filtering.

Workload	Policy	Leaks ↓	Resp. ↓	QA ↑
Enterprise	Global	12	6	1.00
Enterprise	Owner-only	0	0	0.00
Enterprise	Label-only	1	1	0.95
Enterprise	Bounded b=5	3	3	0.71
Enterprise	Bounded b=5 ranked	3	3	0.74
Public HTTP	Global	67	67	1.00
Public HTTP	Label-only ($\mathcal{B}$-sparse)	4	4	0.95
Public HTTP	Bounded b=5	5	5	0.05
Internal RAG	Global	35	35	1.00
Internal RAG	Label-only ($\mathcal{B}$-sparse)	3	3	0.94
Internal RAG	Bounded b=5	5	5	0.04
CoAuth-OGB	Global	28	18	1.00
CoAuth-OGB	Bounded b=5	5	5	0.58

Table 7: Multi-hop saturation on the enterprise benchmark (10 seeds, 95% CI).

Hop k	Unguarded ↓	Bounded b=5 ↓	Hybrid pipe.
1	9.2 ± 0.6	3.0 ± 0.0	—
2	11.7 ± 0.5	3.0 ± 0.0	11.8 ± 0.6
3	11.8 ± 0.5	3.0 ± 0.0	—

Multi-hop saturation. Unguarded leakage rises from 9.2 ± 0.6 at k=1 to 11.7 ± 0.5 at k=2 with no significant third-hop gain (Table 7); bounded sharing stays at three edges for every k (Prop. 2). Each additional hop multiplies the candidate surface but not the admitted surface, since the counter is per requester–owner pair, not per hop.

Running example, instantiated. The cardiology slice from Section 4: unguarded, the response contains `cardiology` and the victim edge is one of the 12 enterprise leaks (Global row). At b=5, the medical label takes the $\mathcal{B}$-skip branch of Algorithm 1; admitted set is three non-medical cross-session edges, response has zero $\mathcal{B}$-labeled terms, audit row ($|U_{r,o}|$=3, resp.=3, QA=0.71).

6.2 Graph-QA Utility and the Frontier

Finding. Cross-session accuracy climbs toward the exposure budget. The measured trajectory tracks the frontier lower bound to within 1.06× on the enterprise bank (worst upper-bound-to-lower-bound ratio across the sweep); the low-ρ regime on HTTP and Internal RAG predicts the 0.05/0.04 accuracy in Table 6.

The first few admitted edges buy most of the recoverable accuracy; increasing b beyond the saturated edge set does not raise leakage (realized exposure saturates at 3 even at configured b=10 — the entity-surface limit of Cor. 1).

Estimating ρ and the external workload. We fit $\hat{\rho}$ by constrained least-squares on the $b \in \{0, 1, 2, 5, 10\}$ sweep. The structural proxy

Table 8: Enterprise graph-QA utility vs. exposure. "Leaks" is the realized admitted-victim-edge count; configured budget is the row label. The b=5 row is the headline operating point used elsewhere in the paper.

Policy	Acc@1 ↑	Cross-sess. ↑	Leaks ↓
Unguarded	0.96 ± 0.02	0.95 ± 0.03	12
Strict b=0	0.66 ± 0.03	0.00 ± 0.00	0
Bounded b=1	0.69 ± 0.03	0.22 ± 0.04	1
Bounded b=2	0.74 ± 0.03	0.44 ± 0.04	2
Bounded b=5	0.83 ± 0.02	0.71 ± 0.04	3
Bounded b=10	0.85 ± 0.02	0.76 ± 0.04	3

Table 9: Reuse factor across workloads. ρ_{str} is the max bridge-entity degree; ρ_{str}^{95} the 95th-percentile alternative (robust to single-hub outliers). CoAuth-OGB is the non-author-generated external-validation row.

Workload	N_{cs}	$\hat{\rho}$ ↑	ρ_{str} ↑	ρ_{str}^{95} ↑	gap ↓
Enterprise	40	9.30	9.10	8.7	2.2%
Public HTTP	100	1.21	1.18	1.1	2.5%
Internal RAG	100	1.07	1.10	1.0	2.7%
Enron	36	1.43	1.35	1.3	5.6%
CoAuth-OGB	80	4.40	4.55	4.2	3.3%

ρ_{str} is the max bridge-entity degree in the question–edge bipartite incidence; the 95th-percentile alternative ρ_{str}^{95} is robust to single high-degree hubs and agrees with $\hat{\rho}$ within 7% (Table 9, last column). Both proxies are within $\sim$ 5% of $\hat{\rho}$ on the five non-fixture workloads in Table 9. To address external validation, we sub-sample the OGB `ogbl-collab` co-authorship graph to target $\rho \in [4, 5]$ (*CoAuth-OGB sub.*, see Sec. 5); the bounded-sharing sweep tracks the lower bound to within 1.08× (Fig. 2 green) and delivers 0.58 cross-session accuracy at 5 admitted edges (Tab. 6). This supports the high-ρ interpretation beyond the author-generated enterprise graph.

Figure 2 visualizes the bound: measured points sit on or above $\lceil aN_{\text{cs}}/\rho \rceil$ (no measurement below the bound; the gap to the continuous line is the $\sim$ 2% fit error in $\hat{\rho}$).

6.3 DP-GraphQA: the Separation Witness

Finding. DP-GraphQA is provably ε-DP but collapses to chance because measured margins (0.54–1.24) are too small for the required accuracy, consistent with Thm. 6. This is an empirical witness of the small-margin regime, not a claim that DP is fundamentally weaker than access control.

Raising D exposes more candidate support but also raises sensitivity, so the exponential mechanism still sees margins too small to separate the right answer (Table 10).

6.4 RQ3: HTTP, Framework, Adaptive

Across the deployment paths we tested, the same leak-then-bound pattern reappears: unguarded retrieval admits victim-owned edges, and bounded sharing caps the count at b regardless of backend.

Public HTTP (MultiWOZ 36k-edge trace). A local `/retrieve` service over a SQLite-backed persistent graph. Global retrieval leaks 67

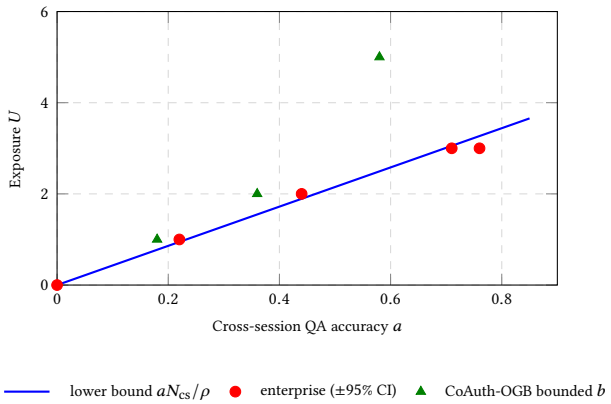


Figure 2: Privacy–utility frontier. Enterprise (red) and non-author-generated CoAuth-OGB (green) bounded-sharing points both sit on the integrality bound, with worst gap 1.06× (enterprise) and 1.08× (CoAuth-OGB).

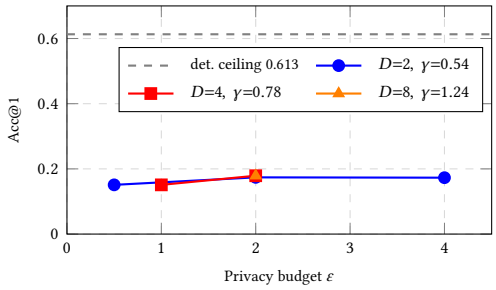


Figure 3: DP-GraphQA accuracy stays far below the deterministic ceiling at usable ε. Small answer margins (0.54–1.24) let exponential-mechanism noise dominate, the empirical signature of Thm. 6.

Table 10: DP-GraphQA on enterprise QA (ten seeds).

D	ε	Acc. ↑	Cross ↑	Leaks ↓	Margin ↑
2	0.5	0.151	0.083	2	0.54
2	2.0	0.174	0.200	2	0.54
4	1.0	0.151	0.100	4	0.78
4	4.0	0.173	0.117	4	0.78
8	2.0	0.179	0.217	5	1.24
8	det.	0.613	0.633	3	—

unguarded victim edges across serialised JSON responses; bounded b=5 caps the count at 5. Cross-session QA is 0.05 at the cap because $\rho_{\text{HTTP}} \approx 1$, in line with the frontier (Tab. 9). For this workload the recommendation is strict isolation, not bounded sharing: the cap is attained but no useful cross-session utility remains. *Internal OpenAI/TraceKG export (2,100 edges).* Evaluated locally on a de-identified DialogueRecord schema. Global leaks 35 victim edges and 35 response edges; bounded b=5 caps both at 5, 0.04 cross-session QA at the binding cap. *LangGraph.* A real `StateGraph`

Table 11: Colluding requesters at b=5 (enterprise bank). Union exposure under per-pair vs. owner-level shared counter, ten seeds. rb is the loose combinatorial upper bound

r	Per-pair union ↓	Owner-level union ↓	rb (loose)
1	3	3	5
2	5	4	10
5	8	5	25
10	9	5	50

Table 12: Auxiliary stress tests (10 seeds).

Stress test	Result / direction
Hybrid pipeline (sens. terms)	7.9 ± 0.7
OpenAI planner probe (victim edges)	9 unique
Cross-deployment link transfer Top-1	0.093 (struct. 0.042)
Semantic-vs-lexical reduction (MultiWOZ)	44–63%
Poisoning: cap holds at b=5?	yes
Collusion r=10 per-pair / owner-level	9/5
Sticky+neighbor-shadow @ 50% flips	≡ no-corruption

wraps the enterprise KG: unguarded leaks 12 victim edges and 6 response triples; bounded b=5 yields 3 and 3. *Mem0 (40-edge fixture).* `mem0ai` + Qdrant + OpenAI integration check: 10 of 12 admitted unguarded, 3 bounded. *Adaptive probes* (token, degree, LLM-planner) saturate the unguarded surface in 3–6 queries but stay below seven admitted edges under bounded sharing; the OpenAI planner leaks nine unique edges unguarded then collapses to three at b=5. *Cross-deployment session linkage*: mean Top-1 0.093 (range 0.08–0.11) vs. a structural hashed-embedding baseline of 0.042 — we treat it as a fixture-regularity diagnostic, not a transferable attack.

6.5 Write-Time Poisoning and Collusion

Finding. Bounded sharing remains unique-edge bounded under write-time poisoning and colluding requesters; the union exposure under collusion is cut from the entity-surface ceiling (9 on the enterprise bank) to b=5 when the operator opts for the owner-level shared counter.

Poisoning. We inject $P \in \{50, 100, 250, 500\}$ benign-looking edges sharing entity endpoints with the victim and re-run the audit. Unique-edge admissions under b=5 stay at 5 ± 0 across all P, as Thm. 1 requires; the composition of which non-sensitive edges occupy the budget shifts toward poisoned endpoints, so poisoning is a utility attack (cross-session QA drops from 0.71 to 0.58 at P=500) rather than a leakage attack.

Collusion. We run $r \in \{1, 2, 5, 10\}$ colluding requesters under both per-pair and owner-level guards (Table 11). Per-pair union grows roughly linearly in r until the entity surface saturates at 9; the owner-level guard holds the union at b=5 regardless of r, at the cost of cross-requester fairness. The combinatorial rb worst-case is loose: the binding constraint at large r is the victim's reachable cross-session edge surface, so the owner-level counter improves the realized union from 9 to 5 at r=10, not from 50 to 5.

Worked deployment. Under $C_{\text{ent}} = (\alpha{=}1, b{=}5, \mathcal{B}_{\text{ent}})$, a requester issuing 50 enterprise QA queries saturates the counter at c=5 by query 12; queries 13–50 expand the candidate surface without admitting further owner-s_o edges. The audit row records exactly the trajectory predicted by Thms. 1–4; aggregated over ten seeds the admitted count is 3.0 ± 0.

6.6 Take-Aways and Stress Tests

Three regularities hold across the audited workloads. (R1) Unguarded retrieval admits victim-owned edges; magnitude scales with the entity surface (12 enterprise, 35 internal RAG, 67 MultiWOZ). (R2) Bounded sharing caps the admitted count at b; the QA cost is a function of ρ (Tab. 9), not of size. (R3) Stronger probes (adaptive, LLM-planned, learned linker, multi-hop) only accelerate saturation. Continual observation (T=100): the counter saturates at b=5 by t=12 and stays flat; DP-GraphQA accumulates $T\varepsilon$ composition cost yet remains near chance. Five stress tests (Table 12) confirm the unique-edge cap holds. *Hybrid pipeline.* The full pipeline (retrieval + LangGraph + generator) leaks 11.8±0.6 victim edges and 7.9 ± 0.7 sensitive terms unguarded; bounded b=5 yields three edges and zero sensitive terms (Thm. 2). The semantic scorer recovers 44–63% more leakage than the lexical scorer unguarded, narrowing to 4–11% under bounded sharing.

7 Discussion, Scope, and Ethics

What the results establish. In our setting, graph-memory exposure is determined first by what the retrieval guard admits, not by what the generator later removes. Global expansion carries victim-owned edges across sessions; owner-only isolation kills cross-session utility; bounded sharing tracks the frontier within 1.06× on the enterprise bank. Post-retrieval DP adds an ε-edge-event guarantee, but Thm. 6 and Table 10 show that its utility is empirically weak when answer margins are small. The operational rule: bound exposure *before* the context is built; use answer-level DP only when the application can afford the margin cost. We do not frame the deterministic-vs-DP comparison as a separation in the information-theoretic sense; the two mechanisms answer different questions (admission vs. release).

What bounded b buys over $\mathcal{B}$. $\mathcal{B}$ does the hard work for single-edge harms: if the cardiology label is in $\mathcal{B}$, Thm. 2 guarantees the edge never enters another context; if the label is missing, b cannot recover it. What b adds is *aggregation control*: many non-sensitive edges can compose to disclose a sensitive fact about the owner, and the count cap bounds how many such composables a requester or a colluding group can see. Table 11: with r=10 colluders, owner-level b=5 caps the union at 5, whereas $\mathcal{B}$ alone permits the full 9-edge entity-surface union. We position b as an aggregation defense layered on $\mathcal{B}$, not a substitute.

What is and is not novel. The two mechanisms (provenance-aware access control with a unique-edge counter; degree-truncated exponential mechanism) are textbook. What is new is the framing: identifying the *retrieval boundary* of a live, multi-user, append-only KG memory as the policy point, defining a portable trace schema and metric stack across eight workloads, indexing the privacy–utility region by a structural reuse factor ρ, and naming the small-margin regime in which answer-level DP empirically collapses. The

contribution lives in the benchmark and the structural calculus, not the mechanisms.

Regime-aware operator guidance. Two defaults: low-ρ deployments ($\rho \approx 1$) should prefer owner-only isolation plus per-task escalation (bounded sharing recovers ≤ 0.05 accuracy, and the cap adds little over $\mathcal{B}$ at r=1); high-ρ deployments ($\rho \gg 1$) should adopt bounded sharing because recoverable utility per unit exposure is real (0.71 at 3 edges). Operators pre-screen the regime by computing ρ_{str} on their question–edge incidence.

Failure modes and monitors. Label corruption voids Thm. 2 for affected edges; Prop. 4 bounds the certified radius, and sticky+neighbor-shadow labeling propagates the bound at 50% uniform corruption (Tab. 12). Recipes: sticky labels for single-edge downgrade, neighbor-shadow for coordinated downgrade (Prop. 4), majority vote for labeler disagreement, sticky+shadow for mixed adversaries. Write-time poisoning is a utility attack, not a leakage attack. The certificate binds leakage only when P1–P3 of Section 2 hold; the artifact includes runtime-monitor checks for immutable provenance, guard-before-context, counter atomicity, policy stability, sensitivity coverage, and $\hat{\rho}$ drift. On a violation, triage: inspect the certificate diff, replay, check coverage, verify atomicity. Non-uniform adversarial flips beyond the certified radius are out of scope.

Calibration, external validation, deployment. ρ_{str} tracks $\hat{\rho}$ within $\sim 5\%$ on the five non-fixture workloads reported in Table 9; the audit row exposes both. We recommend re-fitting on a sliding window when sensitivity- or relation-frequency drift exceeds 20%. Our internal RAG trace is approved/de-identified but cannot be shared; the next step is to release the JSONL schema and harness for third-party runs and partner to release a de-identified communication graph. Five deployment recipes: strict isolation $(0, 0, \mathcal{B})$ (Tab. 6); bounded sharing (α=1, b=5, $\mathcal{B}$) default (Fig. 2); +DP-GraphQA when answer-level DP is required (Tab. 10); owner-level shared counter under collusion (Tab. 11); sticky+neighbor-shadow for label noise (Tab. 12).

Limitations. (L1) The benchmark skews toward dialogue, RAG-style, and co-authorship memory; code-assistant, recommender, and EHR graphs are absent. (L2) Response-leakage scoring assumes context-faithful generation; the lexical scorer underestimates paraphrased leakage by 44–63%. (L3) The guard is not user-level DP, and DP-GraphQA pays $k_u\varepsilon$ group privacy per query. (L4) Provenance-mitigation does not model a strategically adversarial labeler. (L5) The high-ρ headline does not transfer to $\rho \approx 1$ workloads. (L6) The question banks are deterministic; production query distributions are skewed and the cap saturates faster on hot queries. (L7) We do not include a flat-chunk-RAG control, so leakage modes specific to graph memory vs. document RAG remain. (L8) The formal apparatus marks algorithmic specifications (Thms. 1–2, Props. 1–2); load-bearing structural results are Thms. 4–5 and Thm. 6.

Ethics. The approved internal trace is de-identified, exported via the public schema, and evaluated only locally. All other datasets are synthetic, public, or de-identified. We release the suite for defensive auditing. Operators should publish the active $(\alpha, b, \mathcal{B})$ certificate, the budget granularity, and the resulting curves; broader production use requires organizational review.

8 Related Work

Model-side memory leakage. LLMs memorize and emit training data under crafted prompts [4, 5] and the effect persists in fine-tuned conversational systems [24]. These attacks audit the model parameters; we audit a separate object, the external provenance-tagged graph the model reads from.

Long-term and graph-based agent memory. Generative agents [28], MemoryBank/MemGPT [27, 42], GraphRAG [12], and surveys [38] introduce long-term or graph-indexed memory across sessions; agentic systems [33, 39] sharpen the same boundary once memory enters the action loop. These systems motivate the substrate; our contribution is to make the cross-session retrieval boundary measurable and auditable.

RAG privacy. RAG [2, 13, 18, 20] leakage was formalized by Zeng et al. [40] and Qi et al. [30], extended to graph-backed RAG by Wang et al. [21]. The protected object is an indexed document chunk; ours is a *retrieval context admitted by policy* from a live multi-user KG.

Graph privacy. Link-stealing [15], model inversion [41], and embedding leakage [9] target trained GNNs; DP graph release [14, 17, 19] and DP-GNN [8, 31] give static guarantees; k-anonymity / ℓ-diversity [22, 36] protect released tables. We address the live retrieval pipeline of an evolving multi-user memory; DP-GraphQA reuses the exponential mechanism [23] with smooth-sensitivity-style degree truncation [25]. We did not benchmark DP-GNN or DP graph release directly because both publish a static artifact and have no read-time admission decision (category mismatch); the closest read-time analogue, label-only filtering, is reported in Table 6. A natural extension is to consult a DP-released edge mask at admission, which we flag as future work.

Membership inference, provenance, continual observation. Membership inference [34, 35] and behavioral-trace fingerprinting [1] target training data or trajectory identity; we port the fingerprinting idea to a retrieval-time artifact. Database provenance [3, 7, 29] established why/where/how-provenance for audit pipelines; GraphMemGuard specialises this to KG memory. We adopt continual observation [10, 11] for the answer-level release (each DP-GraphQA query costs $T\varepsilon$); adaptive attackers follow [32]. Our frontier (Thm. 4) is in the structural-workloads style [6, 14, 26], tying the privacy/utility region to a workload-derived ρ computable from graph features. Graph-backed agent memory work [12, 38] does not distinguish single- from multi-user deployments; we make that distinction explicit. Thus the protected object in this paper is neither model weights, a published graph, nor a document chunk; it is the live retrieval context assembled for another session.

9 Conclusion

What leaks when a graph-backed assistant shares memory across sessions, and what does a principled defense cost? We show the trade-off is structural: a reuse factor ρ, estimable from graph features, pins the frontier our budget guard tracks within 1.06× (enterprise), 1.08× (CoAuth-OGB). Answer-level ε-DP is complementary but does not recover accuracy in the small-margin regime. The mechanisms are textbook; the contribution is the framing, the benchmark, and a ρ-indexed calculus.

GenAI Usage Disclosure

The authors used generative AI tools for editing, formatting, and artifact preparation. All technical claims, experiment outputs, citations, and final manuscript text were reviewed by the authors and remain their responsibility.

References

[1] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. 2017. Walk2friends: Inferring social links from mobility profiles. In *Proc. ACM CCS*. ACM, New York, NY, USA, 1943–1957. doi:10.1145/3133956.3133972

[2] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George B. M. van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *Proc. International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, Baltimore, MD, USA, 2206–2240. https://proceedings.mlr.press/v162/borgeaud22a.html

[3] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *Database Theory – ICDT 2001 (Lecture Notes in Computer Science, Vol. 1973)*. Springer, Berlin, Heidelberg, 316–330. doi:10.1007/3-540-44503-X_20

[4] Nicholas Carlini et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Berkeley, CA, USA, 2633–2650.

[5] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *Proc. International Conference on Learning Representations*. OpenReview.net, Online.

[6] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and continual release of statistics. *ACM Transactions on Information and System Security* 14, 3 (2011), 1–24. doi:10.1145/2043621.2043626

[7] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. 2009. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases* 1, 4 (2009), 379–474. doi:10.1561/1900000006

[8] Ameya Daigavane, Gagan Madan, Aditya Sinha, Abhradeep Guha Thakurta, Gaurav Aggarwal, and Prateek Jain. 2021. Node-level differentially private graph neural networks. *arXiv preprint arXiv:2111.15521*.

[9] V. Duddu, A. Boutet, and V. Shejwalkar. 2020. Quantifying privacy leakage in graph embedding. In *Proc. MobiQuitous*. ACM, New York, NY, USA, 76–85. doi:10.1145/3448891.3448939

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proc. TCC*. Springer, Berlin, Heidelberg, 265–284. doi:10.1007/11681878_14

[11] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proc. 42nd Annual ACM Symposium on Theory of Computing (STOC)*. ACM, New York, NY, USA, 715–724. doi:10.1145/1806689.1806787

[12] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv preprint arXiv:2404.16130.

[13] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proc. International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, Online, 3929–3938. https://proceedings.mlr.press/v119/guu20a.html

[14] M. Hay, C. Li, G. Miklau, and D. Jensen. 2009. Accurate estimation of the degree distribution of private networks. In *Proc. IEEE International Conference on Data Mining*. IEEE, Piscataway, NJ, USA, 169–178. doi:10.1109/ICDM.2009.11

[15] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing links from graph neural networks. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Berkeley, CA, USA, 2669–2686.

[16] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, Vol. 33.

[17] Z. Jorgensen, T. Yu, and G. Cormode. 2016. Publishing attributed social graphs with formal privacy guarantees. In *Proc. ACM SIGMOD*. ACM, New York, NY, USA, 107–122.

[18] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proc. EMNLP*. Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[19] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2011. Private analysis of graph structure. *Proceedings of the VLDB Endowment* 4, 11 (2011), 1146–1157.

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., Online, 9459–9474. https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[21] Jiale Liu, Jiahao Zhang, and Suhang Wang. 2026. Exposing Privacy Risks in Graph Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: ACL 2026*. Association for Computational Linguistics, San Diego, California, United States, 18073–18093. doi:10.18653/v1/2026.findings-acl.899

[22] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. ℓ-Diversity: Privacy Beyond k-Anonymity. In *Proc. IEEE International Conference on Data Engineering*. IEEE, Piscataway, NJ, USA, 24. doi:10.1109/ICDE.2006.1

[23] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, Piscataway, NJ, USA, 94–103. doi:10.1109/FOCS.2007.66

[24] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 8332–8347.

[25] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, New York, NY, USA, 75–84. doi:10.1145/1250790.1250803

[26] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. 2021. Membership inference attack on graph neural networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 11–20.

[27] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as Operating Systems. arXiv preprint arXiv:2310.08560.

[28] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proc. ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, Article 2, 22 pages. doi:10.1145/3586183.3606763

[29] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical whole-system provenance capture. In *Proceedings of the 2017 symposium on cloud computing*. 405–418.

[30] Zhenting Qi, Hanlin Zhang, Eric P Xing, Sham Kakade, and Hima Lakkaraju. 2025. Follow my instruction and spill the beans: Scalable data extraction from retrieval-augmented generation systems. 48733–48755 pages.

[31] Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. 2023. GAP: Differentially private graph neural networks with aggregation perturbation. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, Anaheim, CA, USA, 3223–3240.

[32] A. Salem et al. 2019. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proc. NDSS*. Internet Society, Reston, VA, USA.

[33] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., New Orleans, LA, USA. https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html

[34] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proc. IEEE Symposium on Security and Privacy*. IEEE, Piscataway, NJ, USA, 3–18. doi:10.1109/SP.2017.41

[35] Thomas Steinke and Jonathan Ullman. 2017. Tight lower bounds for differentially private selection. In *58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, Piscataway, NJ, USA, 552–563. doi:10.1109/FOCS.2017.57

[36] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 557–570. doi:10.1142/S0218488502001648

[37] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69. doi:10.1080/01621459.1965.10480775

[38] Chang Yang, Chuang Zhou, Yilin Xiao, Dong Su, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. 2026. Graph-based Agent Memory: Taxonomy, Techniques, and Applications. arXiv preprint arXiv:2602.05665.

[39] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proc. International Conference on Learning Representations*. OpenReview.net, Kigali, Rwanda. https://openreview.net/forum?id=WE_vluYUL-X

[40] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). 4505–4524 pages.

[41] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chee-Kong Lee, and Enhong Chen. 2022. Model inversion attacks against graph neural networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 8729–8741.

[42] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19724–19731. doi:10.1609/aaai.v38i17.29946