KNOWLEDGE CURATION FOR MULTI-USER COLLABORATION IN AI WORKSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI workspaces move from personal, single-user use to shared, team-level projects, they need a memory that captures what a team knows, who can access it, how it was established, and whether it remains valid. Existing memory mechanisms and evaluations largely target storing and retrieving a single user's long-context history. However, team settings pose a different challenge, as the knowledge is distributed across conversations, meetings, and shared documents, with evolving content and varying visibility. We introduce Team Knowledge Curation (TKC), a framework that represents shared knowledge as scoped items carrying epistemic type, lifecycle state, provenance, and access constraints, curated from raw conversation and maintained through periodic consolidation. To evaluate this setting, which existing benchmarks do not do, we introduce TeamKnowledgeBench (TKB), a benchmark designed to capture team-specific knowledge dynamics and where access denial and abstention are first-class correct answers. Across TKB and an independent GroupMemBench, TKC outperforms existing memory and retrieval systems and is the only system that withholds protected content without leaks, suggesting that reliable collaborative memory requires curating current what the team *knows*, beyond retrieving what was said. Together, TKC and TKB lay a foundation for AI systems that support shared, evolving team knowledge.

1 INTRODUCTION

LLMs are increasingly part of collaborative work, not just single-user assistance. Multi-user project environments such as ChatGPT Projects, Claude Projects, and Gemini Notebook, now place AI assistants alongside shared resources for team projects. Yet, sharing documents and outputs is only one aspect of collaboration (Figure 1): a team assistant must also connect information across conversations, follow how ideas and decisions evolve, and link them to the reasoning behind them. We study the natural next step: augmenting AI assistants with a team-level memory.

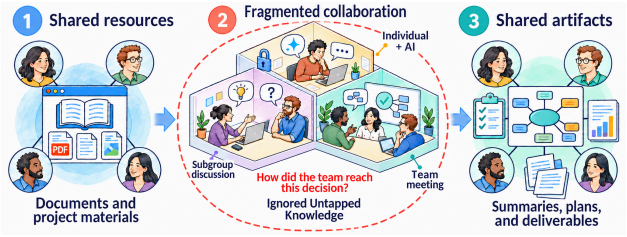


Figure 1: Limited collaboration capability in existing AI spaces.

Motivation: shared team workspaces. Consider a team working in a shared project space that holds documents, slides, and design or research artifacts, together with the drafts, analyses, and summaries generated along the way. Shared material, however, is not shared knowledge. Members consult an AI assistant privately, exchange direct messages, deliberate in subgroup channels, discuss issues with the full team, and continue the work through meetings, notes, and follow-ups. These interactions all serve the same project, but they do not carry the same meaning or authority. An idea raised in a private assistant session may be refined by a subgroup and later inform a team decision, yet it does not become team knowledge merely because it relates to a shared artifact. A useful team assistant must therefore connect this distributed history while retaining who contributed what, for whom it was visible, and when it entered the team's accepted record.

Failure modes. Without an explicit representation of team knowledge, an assistant can fail in three ways. In **false consensus**, a private observation or subgroup proposal is presented as a team decision

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.

before the appropriate group has adopted it. In **stale recall**, the assistant returns a relevant statement that has been revised or superseded, rather than the position the team currently accepts. In **boundary collapse**, the system ignores the audience and authority of a contribution, so individual brainstorming or a subgroup view is exposed to the whole team, or legitimately shared knowledge stays inaccessible outside its original context. None of these is an ordinary retrieval failure. Retrieval can identify what was said, and single-user memory can track one person's history, but neither establishes collective standing or authorized movement across scopes. Once curation loses these distinctions, they cannot be reliably reconstructed at answer time.

Contributions. We take a first step toward collaborative memory: a coherent, evolving representation of what a team has considered and decided across individual and shared interactions. Such a memory enables experiences beyond retrieval: tracing decision rationales, onboarding new members, preserving knowledge when members leave, and providing role-specific briefings. Our contributions:

(1) Collaborative memory as team knowledge. We formulate collaborative memory as requester-conditioned reasoning over distributed team knowledge, with a schema that captures each contribution's type, evolving standing, evidence, provenance, relations, and access boundaries, and an answer target with three outcomes: a cited answer, "not permitted" and "not yet established", so that access denial and abstention are distinct, first-class results rather than errors.

(2) Scope-aware knowledge formation and promotion. We model team knowledge as an evolving state in which knowledge reaches a wider scope only through an authorized promotion that preserves provenance and access history: a contribution's standing is settled only when an actor with the authority to decide does so; updates supersede rather than overwrite; and knowledge reaches a wider audience only through a promotion that mints a new scoped item without exposing its source.

(3) Governed evaluation. We introduce TeamKnowledgeBench, which fixes a symbolic collaboration plan before generating any dialogue, so every gold answer is a deterministic, auditable read of that plan rather than a generative judgment, thus making access denial and abstention first-class correct answers rather than failures.

1.1 Related Work: Retrieval-Based and Agentic Memory Systems

Our work builds on established memory mechanisms, which Figure 2 organizes by scope and adaptivity. We briefly explain their connection to our representation here; Appendix A gives a detailed requirement-level comparison.

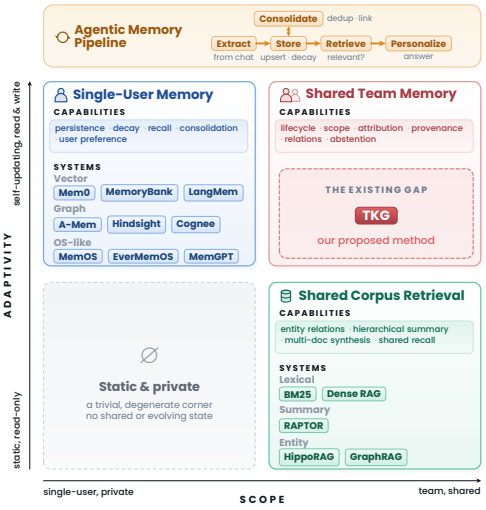


Figure 2: Illustration of agentic memory/corpus retrieval landscape based on scope and adaptability.

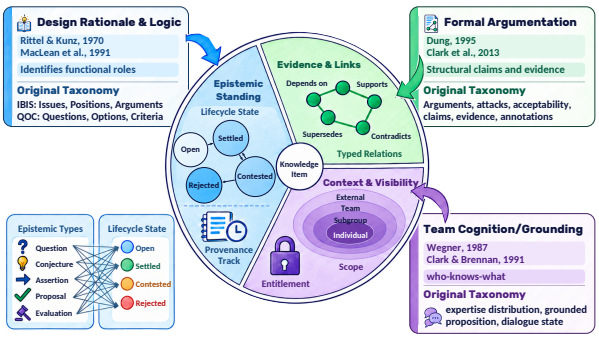
Retrieval-based. Lexical and semantic retrieval treat memory as passages selected for a query (Robertson & Zaragoza, 2009; Lewis et al., 2020). RAPTOR organizes evidence through hierarchical summaries, GraphRAG through entity graphs and community reports, and HippoRAG through graph-based associative retrieval (Sarthi et al., 2024; Edge et al., 2024; Gutiérrez et al., 2024; 2025). These methods improve how evidence is found and organized, but passages, summaries, entities, relations, and extracted facts do not encode whether an item was proposed or accepted, who could settle it, or whether a requester may use it.

Single-user agentic. These systems extract, consolidate, link, or rank information across interactions. MemoryBank and Mem0 retain salient facts and updates; MemGPT/Letta manages context tiers; A-MEM evolves linked notes; Zep provides a bitemporal knowledge graph; Hindsight separates different classes of memory; LangMem exposes memory-management tools; and MemOS adds lifecycle, provenance, scheduling, and permissions (Zhong et al., 2024; Chhikara et al., 2025; Packer et al., 2023; Xu et al., 2025; Rasmussen et al., 2025; Latimer et al., 2025; LangChain, 2025; Li et al., 2025). They provide important mechanisms for persistent recall, but their principal is usually a single user or agent. Team memory must additionally handle multiple contributors, overlapping audiences, role-dependent authority, and items whose standing and permissible use vary by requester. The

closest setting is collaborative memory (Rezazadeh et al., 2025b), which shares memory fragments across users and agents under dynamic read/write policies with immutable provenance; we go further by typing each item's epistemic role and lifecycle, gating standing access on authority, and enforcing per-requester access at retrieval time. Team memory is thus designed to complement, and not replace, corpus retrieval and single-user agentic memory.

2 Grounding a Unified Ontology for Team Knowledge

For agentic memory to serve a team, it must store two things: what each member contributes, and the conditions under which that contribution counts as team knowledge. We propose an ontology for team knowledge: a single schema for knowledge items that make these conditions explicit and machine-enforceable, so that both curation and retrieval can be governed by an item's standing and by what each requester may access. Rather than designing this schema from scratch, we ground it in four complementary research areas, drawing one facet of a knowledge item from each: for every contribution, the schema records the role it plays, how its standing changes over time, what justifies it, and when others may use it. These fields define the nodes and relations of the curated team knowledge graph.



Deliberation: IBIS and QOC structure deliberation and explain design choices by distinguishing issues or questions, positions or options, and arguments or criteria (Kunz & Rittel, 1970; MacLean et al., 1991). We adapt the idea of typed contributions from IBIS/QOC and add a second, orthogonal axis they lack – standing over time. The first axis, an immutable *epistemic type*, records what a contribution was when someone first made it: {QUESTION, CONJECTURE, ASSERTION, PROPOSAL, EVALUATION}. The second axis, a mutable *lifecycle state*, records where the team currently stands on it: {OPEN, SETTLED, CONTESTED, REJECTED}. For example, when the team accepts a proposal, the item remains a proposal and only its state changes to settled. This separation preserves the full path from exploration to decision and keeps the system from confusing what someone contributed with what the team adopted.

Formal argumentation: Argumentation frameworks study whether an argument remains acceptable when others challenge it, and micropublications link scientific claims to the evidence and arguments that support them (Dung, 1995; Clark et al., 2014). We borrow these structures so that why an item is justified can be traced. The ontology records each item's source evidence and provenance, along with typed relations to other items: {SUPPORTS, DEPENDS-ON, CONTRADICTS, SUPERSEDES}. These links show how one contribution supports, conditions, challenges, or replaces another.

Common ground and team cognition: Common-ground research explains how people build shared understanding, as listeners take up and confirm what speakers say. Research on transactive memory and shared mental models explains how teams distribute expertise and responsibility, and coordinate across members (Clark & Brennan, 1991; Wegner, 1987; Cannon-Bowers et al., 1993). These perspectives motivate four fields: {ROLES, AUTHORITY, COLLABORATION SCOPE, AUDIENCE}, that together tell the system who contributed and in what role, who has the authority to settle a contribution, where it was established, and for which audience it holds standing.

Information-flow security: We adapt information-flow models, which track how data moves between security domains (Denning, 1976; Sabelfeld & Myers, 2003), to control how knowledge moves across collaboration scopes. An item's scope records where the team established it, and three nested entitlements, {EXIST, READ, USE}, define what a user may do with it: know that it exists, inspect its content, or rely on it as evidence.

3 Team Collaboration Formulation and Curation

Team knowledge builds up over many interactions that differ in audience and authority, yet they should all feed one shared memory: a persistent knowledge graph in which every contribution records where it came from, who could see it, and how its standing changes over time.

Scoped team knowledge. Let U be the team and $D = (m_1, \ldots, m_N)$ its message history in chronological order, where $m_i = (a_i, s_i, t_i, x_i)$ records the author, surface, timestamp, and content of each message. Each scope $\sigma \in \Sigma$ has an audience $\mathrm{aud}(\sigma) \subseteq U$, and we order scopes by audience inclusion: $\sigma_1 \preceq \sigma_2 \iff \mathrm{aud}(\sigma_1) \subseteq \mathrm{aud}(\sigma_2)$. Knowledge established with one audience does not automatically become knowledge for a larger one; moving it to a wider scope requires an explicit, authorized promotion. Curation turns the history up to time t into a team-knowledge state $\mathcal{K}_t$. Rather than treating every utterance as a fact, $\mathcal{K}_t$ stores the contributions that matter to the team's ongoing work, using the ontology of Section 2:

$$k = \langle id, c, \tau, \ell, \sigma, \eta, \rho, E, R \rangle. \tag{1}$$

Content c and type τ record what was contributed, and lifecycle ℓ gives its current standing. Scope σ and entitlements η record where it was established and who may access or use it. Finally, ρ, E, and R keep its provenance, evidence, and relations to other items. Keeping these fields separate lets the team's view change without rewriting its history; a superseded item, for example, remains available for reasoning about the past.

Requester-conditioned access and answering. What the system can say about this state depends on who is asking and when. Given the validity interval $\nu(k) = [t_k^-, t_k^+)$, we first check whether k is available to requester u at time t:

$$B(k, u, t) = \mathbf{1}[t \in \nu(k) \wedge u \in \mathrm{aud}(\sigma(k))]. \tag{2}$$

We then distinguish three nested permissions, namely knowing that an item exists, reading it, and relying on it as evidence, with $\eta^{use}(k) \subseteq \eta^{read}(k) \subseteq \eta^{exist}(k)$. Together, these define the access gate

$$\mathrm{gate}(k, u, t) = \begin{cases} \textsc{read}, & B(k, u, t) = 1 \wedge u \in \eta^{read}(k), \\ \textsc{sealed}, & B(k, u, t) = 1 \wedge u \in \eta^{exist}(k) \setminus \eta^{read}(k), \\ \textsc{hidden}, & \text{otherwise.} \end{cases} \tag{3}$$

A READ item shows its content, a SEALED item reveals only that relevant restricted knowledge exists, and a HIDDEN item looks exactly as if it were not there. Using an item as evidence, or carrying it into another scope, additionally requires η^{use}.

This distinction matters most when the evidence available to the requester does not support an answer. Let $\mathcal{A} = \mathcal{R}_t^{read}(q, u)$ and $\mathcal{P} = \mathcal{R}_t^{sealed}(q, u)$ be the relevant readable and sealed items, where relevance depends on content as well as standing, evidence, and relations. Let $\mathrm{Suff}(q, S)$ indicate that S contains enough evidence to answer q, counting only evidence that u may use at time t, and let $\mathrm{Val}(q, S)$ be the answer this evidence supports. The target response is then

$$\textsc{Gold}(q, u, t) = \begin{cases} \textsc{answer}(\mathrm{Val}(q, \mathcal{A})), & \mathrm{Suff}(q, \mathcal{A}), \\ \textsc{abstain-private}, & \neg\, \mathrm{Suff}(q, \mathcal{A}) \wedge \mathcal{P} \neq \varnothing, \\ \textsc{abstain-gap}, & \text{otherwise.} \end{cases} \tag{4}$$

The last two cases separate two situations: "an answer may exist, but the requester cannot access it" and "the team has not settled on an answer." A sealed item lets the system say the first; a hidden item cannot be used even to hint that such information exists.

Objective. A useful team memory must therefore do more than return the right content: it must not disclose restricted evidence, but it also should not become so cautious that it withholds answers the requester is entitled to. For an answering system f, we capture these three concerns as

$$\mathcal{R}(f) = (\Pr[\mathrm{Leak}_\lambda(f)],\ \Pr[f(q, u, t) \neq \textsc{Gold}(q, u, t)],\ \Pr[\mathrm{Overhold}(f)]). \tag{5}$$

Here Leak_λ flags any unauthorized contiguous source span of at least λ tokens, the second term measures how often the response differs from the target, and Overhold captures abstentions made even though enough authorized evidence was available, a failure that TKB also scores (B2 and B3, Section 4). The leakage test detects disclosure of source spans; it does not guarantee semantic privacy.

3.1 Knowledge Curation

We curate team knowledge $\mathcal{K}_t$ by separating proposal from commitment (Figure 3): models propose changes, and a single deterministic writer is the only component that commits them—because interpreting language is uncertain, whereas changing what the team knows must follow explicit, auditable rules. Below, we describe proposal (ontology-guided extraction and consolidation), the governed writer that commits, promotion across scopes, and guarded retrieval.

Ontology-guided extraction. Curation starts with short windows of interaction, where references and context are still easy to resolve. Each window contains only complete turns and ends when the audience changes, so no extraction silently crosses a collaboration boundary. Two files guide the model: the extraction `constitution.md` defines what counts as an item and what evidence it needs, and the domain `skill.md` supplies the vocabulary, roles, and interpretation cues for the current workspace; a new domain needs only a new skill file (Appendix B.1). For each window, one structured model call proposes self-contained items, their source citations, their relations, and any updates to earlier items that the window supports. The model preserves qualifiers, versions, and authorship, and keeps contributions separate when they may later gain different standing.

Consolidation updates. Some relationships become clear only later: a later discussion may show that two items describe the same issue, challenge an earlier claim, or settle a proposal made several sessions ago. Consolidation therefore runs periodically over the accumulated graph to link evidence across windows. It uses the same ontology and domain skill, but similarity alone never suffices to merge two items or to mark one as replacing another: if users raise the same point independently, the items stay separate unless the trace supports a stronger relation. Extraction and consolidation send their proposals to the same writer. Before changing $\mathcal{K}_t$, the writer checks that each operation is backed by evidence, respects the source scope and entitlements, points to a valid target, and keeps history intact. Lifecycle changes are gated strictly: an item becomes SETTLED only when the trace contains an explicit settling cue and the actor who gave it holds the authority to decide at that scope—so confident or repeated wording never settles a claim on its own. Appendix B.3 shows the knowledge-item schema.

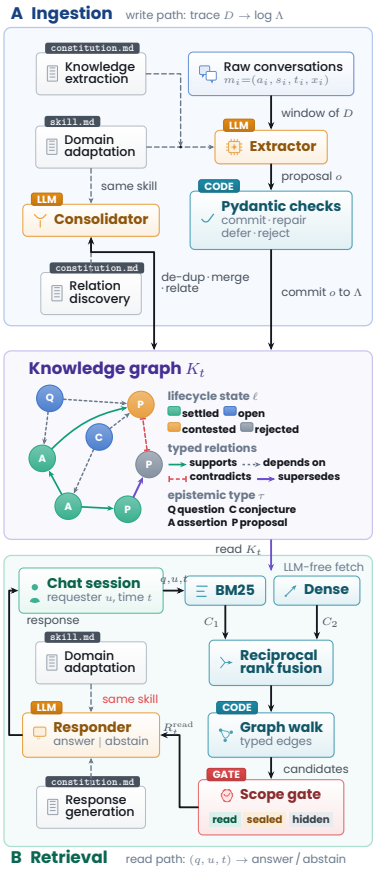


Figure 3: Team Knowledge Curation ingestion and retrieval pipeline.

Promotion across collaboration scopes. Knowledge often needs to reach a wider audience than the one where it started. Widening the original item would lose its provenance, so promotion instead creates a new item that links back to its source; the new item can join the wider conversation, while the source keeps its original protection. Sharing is not acceptance: promoted information becomes settled only when evidence shows that an authorized actor or group has adopted it. The release model supports existence pointers, attributed claims, and summaries, but the automatic path currently emits only existence pointers, which reveal that relevant knowledge exists without exposing what it says. A team member can still restate an idea at team scope, but that restatement is a separate item and grants no access to the restricted original.

Guarded retrieval and response. We enforce the same boundaries during retrieval, not only after evidence has been chosen. Let $C_j(q)$ be the candidates that retriever j (lexical, dense, or transcript-to-item) returns for query q, with rank $r_j(k)$ and offset $\kappa > 0$ (we use $\kappa = 60$). Reciprocal-rank fusion (Cormack et al., 2009) then scores only the items that requester u can read at time t:

$$\text{score}_{u,t}(k \mid q) = \mathbf{1}[\text{gate}(k, u, t) = \text{READ}] \sum_{j:\, k \in C_j(q)} \frac{1}{\kappa + r_j(k)}. \tag{6}$$

The gate is checked again each time retrieval follows a link in the graph, so a readable item cannot lead to a restricted one. Sealed items contribute only a signal that they exist, which triggers ABSTAIN-PRIVATE. The responder answers from readable item cards and source evidence under the response constitution, and it can cite an item only if the requester is entitled to use it.

4 Team Knowledge Benchmark Design

Existing memory benchmarks mostly evaluate one-on-one histories: LoCoMo and LongMemEval test recall, temporal reasoning, and knowledge updates over long dyadic or user–assistant conver-

sations (Maharana et al., 2024; Wu et al., 2025). GroupMemBench and EverMemBench bring in multiple parties (Yang et al., 2026; Hu et al., 2026b), but they still ask what was said, not what the team established: which proposal the team adopted, who may see an analysis, or who first raised an idea. Evaluating team memory therefore needs a new design. Because transcripts rarely state these facts outright and LLM judges often confuse them, we build Team-KnowledgeBench (TKB) backwards: answers first, dialogue second.

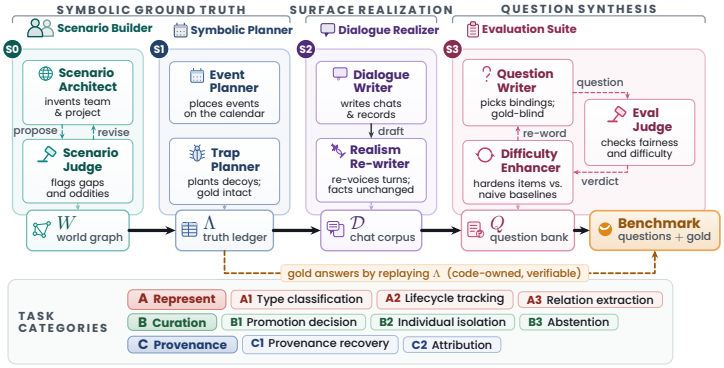


Figure 4: Team Knowledge Benchmark generation framework pipeline

Formulation. Each TKB instance follows a five-person team and its assistant through twenty working days of private assistant sessions, subgroup chats, team meetings, and written records. We record the team's knowledge in an append-only ledger $\Lambda = (o_1, \ldots, o_m)$, the hidden counterpart of $\mathcal{K}_t$ in Section 3. Each operation adds a proposition p, links two propositions by a typed relation, or moves p to a new lifecycle state $\ell(p)$. Adding p fixes its type $\tau(p)$, author $a(p)$, scope $\sigma(p)$, and the members $A(p) \subseteq U$ who heard it. An instance $\mathcal{B} = (W, \Lambda, D, Q)$ also holds a world plan W, the dialogue D that voices the ledger, and questions Q, each with an asker u_q, a time t_q, and a hidden query π_q. A system answers from the dialogue alone, and we read the gold answer from the ledger:

$$\hat{y}_q = f\big(q, u_q, t_q;\, D_{\le t_q}\big), \qquad y_q = \textsc{ReadLedger}\big(\pi_q, u_q, t_q;\, \Lambda_{\le t_q}\big). \tag{7}$$

Gold depends only on the ledger up to t_q, so neither later events nor any rewording of the dialogue can change it. Reads also check $A(p)$, so the right answer can depend on who asks (Appendix C.1).

Generation pipeline. Our generator (Figure 4) follows one rule: models propose, and code decides. In *symbolic ground truth* (S0–S1), models expand a hand-written domain brief into people, work items, propositions, relations, and lifecycle arcs, and code checks every operation before committing it, so the ledger never holds an inconsistent history. Code then schedules each event and decides who hears it, because timing and audience change answers. We also plant traps that tempt shortcuts, such as trusting a confident private claim or a superseded plan, and verify that no trap changes its target's standing or visibility. In *surface realization* (S2), a writer voices each new proposition and state change in a realistic conversation, and a rewriter makes it sound natural. Judges check these turns for fidelity and realism, and we drop any turn that keeps failing, so fact-carrying turns say exactly what the ledger says. In *question synthesis* (S3), a writer turns ledger facts into questions. Because the writer knows each answer and could give it away, we keep a question only if it is hard for the right reason yet fair:

$$\textsc{Accept}(q) = \neg\textsc{Blind}(q) \wedge \neg\textsc{BM25}(q) \wedge \neg\textsc{Dense}(q) \wedge \textsc{Juror}(q) \wedge \textsc{Window}(q), \tag{8}$$

that is, a question-only reader and two retrieval baselines must fail, while a cross-family juror and the evaluation solver must succeed from the cited evidence. Publication runs no model: code recomputes gold and drops any question whose deciding event never reached the text (Appendix C.2).

For the generation, we use Claude Opus 4.7 for stages (S0-S1), due to their superiority in planning tasks, and GPT 5.4 for (S2) to generate actual dialogues. For creating evaluation suite (S3), we use CLaude Opus 4.7 as question proposer and refiner and GPT 5.6 Terra as the eval judge.

Tasks and domains. Eight task families each read a different ledger field. *Representation* asks for a contribution's type (A1), lifecycle state (A2), and relations (A3). *Curation* asks whether private work may enter the team record (B1), whether the asker may see it (B2), and whether the team has settled on an answer (B3); B2 and B3 mirror the two abstentions in Equation 4. *Provenance* asks which turns first raised a decision's premises (C1) and who first raised an idea (C2). We use two domains with different confidentiality rules: a legal team running a cross-border acquisition under attorney–client privilege, and a finance team closing a quarter behind an information wall. Their

System	Representation			Curation			Provenance		Overall
	Epistemic	Lifecycle	Relations	Promotion	Isolation	Abstention	Capture	Attribution	Composite
Retrieval and graph retrieval									
BM25-RAG	0.355	0.394	0.375	0.250	0.792	0.529	0.000	0.594	0.411
Dense-RAG	**0.484**	0.394	0.562	0.208	0.917	0.471	0.200	0.625	0.483
HippoRAG-2	**0.484**	0.303	0.562	0.375	0.792	0.500	0.000	0.719	0.467
GraphRAG	0.387	0.333	0.531	0.458	0.833	0.529	0.000	0.562	0.454
RAPTOR	0.387	0.455	0.469	0.167	0.750	0.471	0.000	0.688	0.423
Cognee	0.226	0.303	0.562	0.375	0.792	0.412	0.000	0.531	0.400
Single-user and agent memory									
Mem0	0.323	0.333	0.406	0.333	0.583	0.500	0.000	0.719	0.400
MemoryBank	0.355	0.182	0.625	0.458	0.750	0.441	0.000	0.625	0.430
MemOS	0.323	0.273	0.625	0.333	0.750	0.412	0.000	0.656	0.422
LangMem	0.290	0.273	**0.656**	0.333	0.708	0.529	0.000	**0.750**	0.442
Hindsight	0.387	0.394	0.500	0.250	0.833	0.529	0.000	0.594	0.436
TKG	**0.484**	**0.515**	0.469	**0.542**	**0.958**	**0.676**	**0.400**	0.719	**0.595**

Table 1: Composite is the unweighted mean of the eight family metrics and therefore differs from the question-weighted headline accuracy in Figure 5. Bold is best in column. Provenance capture is the clearest frontier, while the conservative writer does not lead relation extraction.

question banks hold 215 and 99 questions over corpora of about 6.1k and 9.5k turns (Appendix C). We also perform a detailed automated and independent human audit of the generated benchmark to validate the quality of the generated dialogues as well as evalaution suite (Appendix C.3).

5 Evaluation Results

In this section, we evaluate whether curating what a team has established lets an assistant answer more reliably than existing memory systems.

Evaluation setup. We compare against retrieval, graph-retrieval, and agent-memory systems on the legal and finance instances of TKB. All systems receive the same conversation histories and public question metadata; the ledger and answer bindings remain evaluator-only. Every system that calls an LLM during ingestion, including TKC's extractor and consolidator, uses GPT-5.4-mini, which keeps memory construction cost-effective and costs comparable. All systems answer with GPT-5.4 under the same answer options, public answering guide, and context budget, and a GPT-5.6 judge checks each answer against its code-computed gold label (Appendix C.4). We report question-weighted headline accuracy and the eight task-family metrics; the composite in Table 1 is their unweighted mean. Unless otherwise stated, each conclusion rests on effects that recur across both domains and a second benchmark.

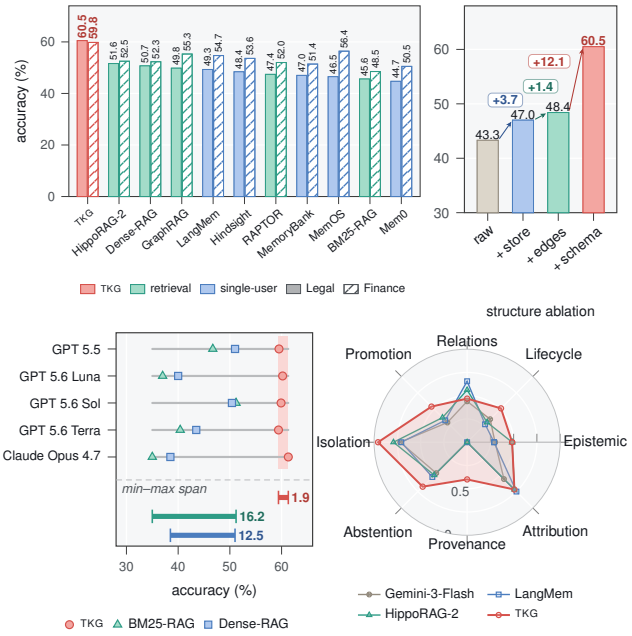


Figure 5: TKB results. Top left: composite accuracy on Legal (solid) and Finance (hatched); top right: structure ablation on Legal (red = typed schema, TKG); bottom left: answerer sensitivity on with each system's min–max span; bottom right: per-task accuracy.

5.1 RQ1: Does curation recover what the team has established?

A team assistant should answer from the position the team holds now, which is often not the one repeated most. TKC leads on both TKB domains, and the runner-up changes from a graph retriever on legal to plain lexical retrieval on finance (Figure 5, top left). The task families locate the lead (Table 1). On questions about a claim itself, such as its type or its author, several baselines keep pace. The gap opens on questions about a claim's standing and support: lifecycle, where stale recall shows up; promotion and abstention, where false consensus does; and provenance capture, a small family on

which only one baseline answers any item correctly. In the legal item of Figure 11 (Appendix C.1), a claim voiced in every standup stays easy to retrieve after the team rejects it. TKC stores that rejection as a lifecycle update on the item, as it stores scope for promotion and source links for provenance, so the answerer reads standing instead of reconstructing it from passages. On relations, several baselines, led by LangMem, recover more links. Our writer commits a relation only when the trace supports it, which guards against false merges and supersessions but leaves some true links unrecorded; relation coverage and original-speaker recovery remain open challenges.

5.2 RQ2: Where does the gain come from?

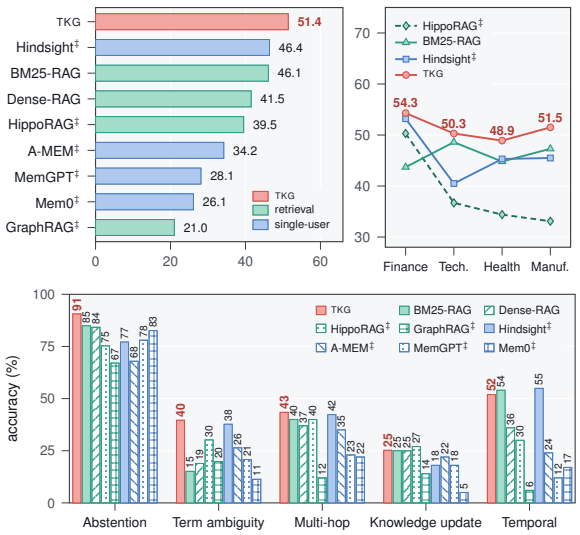


Figure 6: GroupMemBench transfer without lifecycle or access labels: overall accuracy on the 696-question subset (top left), accuracy by domain (top right) and by task (bottom).

Structure ladder. TKC extracts claims, links them, and records each claim's type, scope, and lifecycle. We add these layers one at a time with the corpus, answerer, context budget, and scorer held fixed (Figure 5, top right). Across domains, the flat store and the entity edges each help modestly, while typed, scoped, lifecycle-bearing items add more than twice as much as both combined. Connectivity, which graph-based retrieval already offers, explains little of the gain; it comes from recording the conditions under which a claim counts as team knowledge, so the answerer receives standing, scope, and evidence already reconciled across interactions. The ladder adds these properties jointly and does not isolate the contribution of each field.

Answerer sensitivity. If the record carries the interpretation, the choice of answering model should matter little. Across five other answerers on legal, TKC stays nearly flat, while BM25-RAG and Dense-RAG vary several times as much (Figure 5, bottom left). Retrieval leaves the answerer to work out from passages what the team decided, and models differ in how well they do that. TKC hands over items whose standing and permitted evidence are already recorded, so the answer depends on the record more than on the model reading it.

Together, these results show that reliable team memory depends on preserving both evidence and access boundaries throughout curation and answer generation. Correct item-level access decisions alone do not prevent leakage when evidence is combined. In our audit of twelve withhold-gold items per domain, TKC disclosed no restricted source spans in either legal or finance, while every baseline leaked on some items. Curation also remains the main accuracy bottleneck: perfect retrieval over the current store would close less than half the gap to exact evidence. This gap reflects extraction, normalization, and commitment errors, rather than extraction recall alone. The central challenge is therefore to preserve each claim's identity, supporting evidence, updates, and access restrictions: retrieval cannot recover distinctions or evidence links lost when the record is built.

5.3 RQ3: Does the benefit transfer beyond our benchmark?

Because TKB was built around our ontology, we also evaluate on GroupMemBench, whose questions were written independently of it and never ask for lifecycle or access labels. We keep the engine and core ontology, adapt the extraction and QA instructions to the benchmark protocol, and ingest all messages at open team scope. The benchmark has no privacy dimension, so TKC's access control plays no part here. On the 696 questions outside `user_implicit`, a category that depends on an individual asker's history rather than on team knowledge, TKC is ahead of our BM25-RAG rerun under the shared answerer and judge, overall and in each of the four domains (Figure 6). Including the category leaves the ordering unchanged (50.3% vs. 46.0%), and TKC also sits above the strongest published result, although that row's different answerer and judge make the comparison descriptive.

Against the matched BM25-RAG run, term-ambiguity accuracy more than doubles and abstention rises from an already high level, consistent with domain-aware curation preserving a claim's intended meaning and recording whether the available record supports an answer. The benefit does not reach revisions and time. TKC only ties BM25-RAG on knowledge updates, since an update the writer cannot ground is never committed, the same trade-off that costs relation coverage on TKB; the strongest temporal results also stay above TKC's. Appendix D.1 reports every category.

5.4 RQ4: Does the upfront cost of curation pay off?

TKC does most of its work before any question arrives: it interprets the conversation once and commits a record that later questions reuse. We meter the finance runs to see what this buys, plotting accuracy against one-time ingestion cost and against cost per question (Figure 7), together with full-context runs that give the answering model the whole history for every question and keep no memory at all.

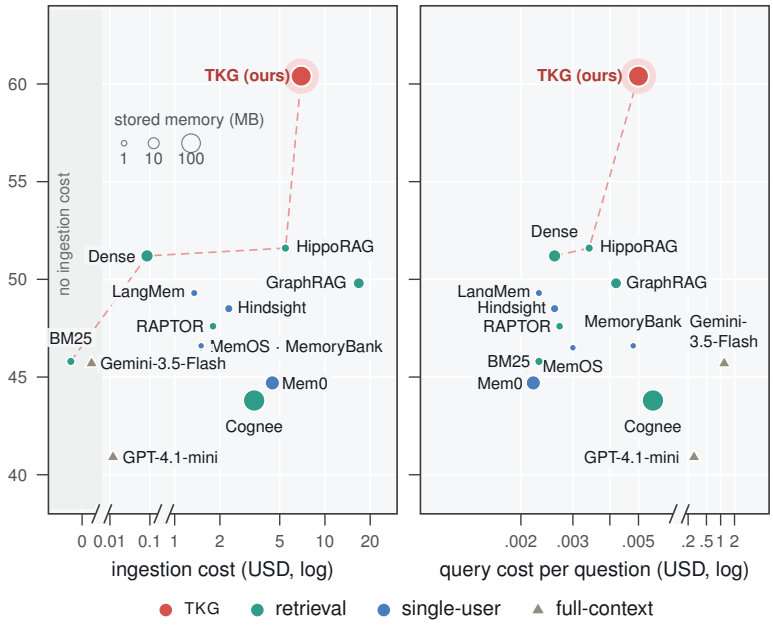


Figure 7: Accuracy vs Cost Pareto: (left) one-time ingestion cost and (right) per-question query cost.

In both views, TKC lies on the accuracy–cost frontier. Per question, it sits between the retrieval and memory baselines and the full-context runs in cost, and above both in accuracy. The full-context runs are the telling comparison: they skip ingestion yet score lower at a higher cost per question, so at these operating points extra spending at question time does not buy the accuracy that curation provides. The retrieval and memory baselines are cheaper because they leave each claim's standing to be inferred at answer time, which is where RQ1 located their errors. Curation works out standing once, when the record is written, and every question reuses that work through a retrieval step that makes no generative call; this is also what kept TKC stable across answerers in RQ2. How favorable this trade is in an ongoing workspace depends on how often the record is read relative to how often it changes, which makes efficient incremental curation the natural next step.

6 Conclusion

Our work attempts to advance collaborative AI workspaces through TKC, a framework for curating team knowledge, and TKB, a benchmark organized around collaboration-relevant tasks. Our TKC outperformed retrieval and agent-memory baselines on benchmarks, demonstrating the value of explicitly representing knowledge types and access boundaries.

Limitation and Scope: Our contribution fundamentally centers on how knowledge is promoted and preserved. Building on this foundation calls for mechanisms that manage decay, retention, and forgetting as team records grow, alongside evaluation in real team workflows. Scope and entitlement similarly establish a concrete basis for access-aware curation, with consent, workspace policy, and governance forming essential directions for future work. As a layer that complements long-context and single-user memory, TKC moves toward assistants that help teams carry knowledge forward across people, discussions, and decisions. The goal is a workspace where teams can build on what they collectively know while respecting the boundaries under which that knowledge was shared.

AI USE STATEMENT

The authors developed the core intellectual contributions of this work, including the conceptual framework, ontology, method, evaluation criteria, experimental design, and interpretation of results. AI tools supported literature discovery, guided implementation, synthetic data generation, and manuscript preparation.

We used deep-research tools to identify candidate literature gaps, which we verified manually. Coding assistants supported selected components of the TKC pipeline under author-defined specifications and modular, manual oversight, without unrestricted edit access. We also used the skill-creator feature to refine operational instructions for language models. Because real team conversations were unavailable due to privacy constraints, language models generated the synthetic dataset following detailed author-written instructions, as described in the paper.

We designed all the figures and tables, and then used AI image generators to refine and enhance it. We also used coding agents to design plots for results and generate TikZ code for figures of the pipeline design. We used writing assistant for refining the manuscript. Some examples include: checking grammar, rephrasing certain sentences for enhancing clarity, suggestions for better presentation.

We reviewed all AI-assisted outputs, validated code through unit tests and end-to-end runs, checked figures and tables against experiment logs, and manually verified every reference against its source. The authors take full responsibility for the final text, claims, code, and artifacts.

ETHICS STATEMENT

All data used in this study are synthetic; no real-world team conversations or user data were collected or used. Beyond the privacy and workplace-surveillance risks associated with deploying team-memory systems, we identified no additional ethical concerns within the scope of this study. Collaborative team memory should always be supported by strong workspace governance and clear policies on data collection, access, retention, and use. Although designing and enforcing these policies is beyond the scope of this work, they are essential to the responsible deployment of a complete collaborative system in the real world.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 430–437, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL https://aclanthology.org/L12-1296/.

Janis A. Cannon-Bowers, Eduardo Salas, and Sharolyn Converse. Shared mental models in expert team decision making. In N. John Castellan, Jr. (ed.), *Individual and Group Decision Making: Current Issues*, pp. 221–246. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. arXiv preprint arXiv:2504.19413, 2025.

Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley (eds.), *Perspectives on Socially Shared Cognition*, pp. 127–149. American Psychological Association, 1991. doi: 10.1037/10096-006.

Tim Clark, Paolo N. Ciccarese, and Carole A. Goble. Micropublications: A semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics*, 5:28, 2014. doi: 10.1186/2041-1480-5-28.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114.

Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19 (5):236–243, 1976. doi: 10.1145/360051.360056.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995. doi: 10.1016/0004-3702(94)00041-X.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. arXiv preprint arXiv:2404.16130, 2024.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. arXiv:2510.18866.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-2017.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2405.14831.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. arXiv:2502.14802.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Yi Bai, Dannong Xu, Tianwei Lin, Xiaohong Li, Yunyun Han, Jian Pei, , and Yafeng Deng. Evaluating long-horizon memory for multi-party collaborative dialogues. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9082–9090, 2026b. doi: 10.1145/3770855.3817589.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Werner Kunz and Horst W. J. Rittel. Issues as elements of information systems. Working Paper 131, Institute of Urban and Regional Development, University of California, Berkeley, CA, 1970.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. doi: 10.2307/2529310.

LangChain. LangMem. Software library, 2025. URL https://github.com/langchain-ai/langmem.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2025.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases, 2020. URL https://arxiv.org/abs/2002.04108.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. arXiv:2005.11401.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Junpeng Ren, Huayi Lai, Hao Wu, Bo Tang, Zhenren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofeng Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. MemOS: A memory OS for AI system. arXiv preprint arXiv:2507.03724, 2025.

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

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. MemoChat: Tuning LLMs to use memos for consistent long-range open-domain conversation. arXiv preprint arXiv:2308.08239, 2023.

Allan MacLean, Richard M. Young, Victoria M. E. Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. *Human–Computer Interaction*, 6(3–4): 201–250, 1991. doi: 10.1080/07370024.1991.9667168.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747.

Yanki Margalit, Nurit Cohen-Inger, Erni Avram, Ran Taig, and Oded Margalit. Governed shared memory for multi-agent LLM systems. *arXiv preprint arXiv:2606.24535*, 2026.

Vasilije Marković, Lazar Obradović, Laszlo Hajdu, and Jovan Pavlović. Optimizing the interface between knowledge graphs and LLMs for complex reasoning. arXiv preprint arXiv:2505.24478, 2025.

Magdalena Markowska, Mohammad Taghizadeh, Adil Soubki, Seyed Mirroshandel, and Owen Rambow. Finding common ground: Annotating and predicting common ground in spoken conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8221–8233, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.551.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Michael Nygard. Documenting architecture decisions. Cognitect Blog, November 2011. URL https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions.

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 C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/S18-2023.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-2124.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. arXiv preprint arXiv:2501.13956, 2025.

Alireza Rezazadeh, Zichao Li, Ange Lou, Yuying Zhao, Wei Wei, and Yujia Bao. Collaborative memory: Multi-user memory sharing in LLM agents with dynamic access control. *arXiv preprint arXiv:2505.18279*, 2025a.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for LLMs. In *International Conference on Learning Representations*, 2025b. arXiv:2410.14052.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003. doi: 10.1109/JSAC.2002.806121.

Andrei Sabelfeld and David Sands. Declassification: Dimensions and principles. *Journal of Computer Security*, 17(5):517–548, 2009. doi: 10.3233/JCS-2009-0352.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996. doi: 10.1109/2.485845.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024. arXiv:2401.18059.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents. arXiv preprint arXiv:2507.07957, 2025.

Daniel M. Wegner. Transactive memory: A contemporary analysis of the group mind. In Brian Mullen and George R. Goethals (eds.), *Theories of Group Behavior*, pp. 185–208. Springer, 1987. doi: 10.1007/978-1-4612-4634-3_9.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. arXiv:2410.10813.

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.

Jingbo Yang, Kwei-Herng Lai, Xiaowen Wang, Shiyu Chang, Yaar Harari, and Evgeniy Gabrilovich. Groupmembench: Benchmarking llm agent memory in multi-party conversations. *arXiv preprint arXiv:2605.14498*, 2026.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. arXiv preprint arXiv:2507.02259, 2025.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1009.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, ZhongZhi Li, Yingying Zhang, Le Song, and Qianli Ma. LifelongAgentBench: Evaluating LLM agents as lifelong learners. arXiv preprint arXiv:2505.11942, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. arXiv preprint arXiv:2506.15841, 2025.

APPENDIX

TABLE OF CONTENTS

A Positioning and related work **15**
A.1 Foundational collaboration traditions 15
A.2 Memory and retrieval systems 16
A.3 Long-context and memory benchmarks 18
B Knowledge representation details **19**
B.1 The team-knowledge space 20
B.2 Epistemic type and lifecycle standing 20
B.3 Knowledge-item schema 20
C TeamKnowledgeBench **22**
C.1 Formal specification 22
C.2 Generation pipeline 23
?? Models ..... **??**
?? Domains and released benchmark ..... **??**
C.3 Post-generation audit 26
C.4 Evaluation protocol 27
D Extended results **28**
?? TeamKnowledgeBench diagnostics ..... **??**
D.1 GroupMemBench 28

A POSITIONING AND RELATED WORK

The problem of representing team knowledge predates AI assistants. Research on design rationale, argumentation, common ground, team cognition, retrieval, and memory has long studied how people preserve decisions, coordinate expertise, and maintain information over time. These traditions identify many of the ingredients that team memory needs, but they do not jointly represent the object our setting requires: a machine-enforceable ledger in which each knowledge item has an epistemic type, team-facing lifecycle standing, scope-aware provenance, item-level entitlements, and explicit rules for crossing audience boundaries.

A.1 FOUNDATIONAL COLLABORATION TRADITIONS

Collaboration and knowledge representation. Design-rationale systems show that collaborative work cannot be reduced to its final outputs. IBIS, QOC, and architecture decision records represent issues, options, criteria, arguments, decisions, and their rationale (Kunz & Rittel, 1970; MacLean et al., 1991; Nygard, 2011). Formal argumentation and micropublications similarly distinguish claims from the evidence and counterarguments that support or challenge them (Dung, 1995; Clark et al., 2014). These traditions motivate the epistemic, evidential, relational, and lifecycle fields of TKC:

Table 2: Foundational collaboration, rationale, and knowledge-representation traditions against the requirements of team-knowledge curation. These traditions contribute the typed-rationale, grounding, and team-cognition ideas we build on, and several are strong on *epistemic type* (their defining strength).

Work	Primary object	Auto build	Epistemic type	Lifecycle standing	Prov. + audience	Entitlement	Safe crossing	Multi-party	3-way abstain	Why it is not sufficient for team memory
IBIS/gIBIS (Kunz & Rittel, 1970)	issues, positions, arguments	✗	✓	●	●	✗	✗	●	✗	Strong typed rationale structure, but human-authored and not designed for access-controlled AI workspaces.
QOC/design-space-analysis (MacLean et al., 1991)	questions, options, criteria	✗	✓	●	●	✗	✗	●	✗	Captures trade-offs, but not automatic curation, entitlement, or cross-surface promotion.
Architecture-decision-records (Nygard, 2011)	decision records with status fields	✗	●	✓	●	✗	✗	✗	✗	Excellent lightweight decision provenance; too coarse for general team knowledge items.
Formal-argumentation (Dung, 1995)	arguments, attacks, acceptability	●	✓	●	✗	✗	✗	✗	✗	Provides rigorous acceptance semantics, but abstracts away authorship, audience, and workspace provenance.
Micropublications (Clark et al., 2014)	claims, evidence, arguments, annotations	✗	✓	●	●	✗	✗	✗	✗	Strong claim–evidence structure; aimed at curated scholarly communication, not messy team traces.
Dialogue-acts/ISO-annotation (Bunt et al., 2012)	utterances with communicative functions	●	✓	●	●	✗	✗	●	✗	Types conversational moves, but usually at utterance or dialogue level rather than durable team memory.
Common-ground (Clark & Brennan, 1991)	grounded propositions and dialogue state	●	✓	●	●	✗	✗	●	✗	Explains how knowledge becomes shared, but does not provide a persistent multi-surface store.
Transactive-memory (Wegner, 1987)	who-knows-what and expertise distribution	✗	✗	●	✓	✗	●	✓	✗	Makes expertise and ownership central, but treats the knowledge items themselves as opaque.
Shared-mental-models (Cannon-Bowers et al., 1993)	aligned task, team, and equipment models	✗	✗	●	●	✗	✗	✓	✗	Explains team alignment, but not a machine-readable item schema or curation procedure.

a proposal, an objection, a supporting claim, and a settled decision are not interchangeable merely because they mention the same object. Architecture decision records are especially close on lifecycle, since a decision record carries an explicit status and may later be superseded.

Dialogue and common-ground research supplies a complementary distinction. A contribution does not become shared knowledge merely because it was uttered; it becomes shared through uptake, grounding, and corresponding changes to the participants' information states (Clark & Brennan, 1991; Bunt et al., 2012; Markowska et al., 2023). The distinction matters directly in AI-mediated workspaces, where the same sentence may occur in a private assistant session, a subgroup discussion, or a team meeting, with different standing as team knowledge.

Team-cognition research adds the social dimension. Transactive-memory and shared-mental-model theories emphasize that effective teams maintain not only facts, but also representations of who knows what, who is responsible for what, and how expertise is distributed (Wegner, 1987; Cannon-Bowers et al., 1993). This view motivates treating ownership, source, scope, and authority as first-class structure: what a statement permits the system to conclude can depend on who introduced it, who had standing to approve it, and for whom it is usable.

These traditions provide the conceptual foundations, but they stop short of the operational problem considered here. Their structures are typically authored by people, assume that issues, claims, arguments, or records already exist as clean objects, or describe collaborative cognition without constraining what an assistant may reveal or use. Modern workspaces instead produce knowledge continuously across private assistant sessions, subgroups, team channels, meetings, documents, issue trackers, and durable records, and the resulting state may be partial, duplicated, contradictory, superseded, differently scoped, or visible only to particular members. TKC therefore turns these foundations from descriptive categories into fields of an automatically curated, machine-enforced team-knowledge item.

A.2 Memory and retrieval systems

Retrieval and structured RAG. Lexical and dense retrieval rank passages by lexical or semantic relevance (Robertson & Zaragoza, 2009; Lewis et al., 2020). Structured variants improve this substrate with entity graphs, community summaries, relational links, or hierarchical summary trees (Edge et al., 2024; Gutiérrez et al., 2024; Sarthi et al., 2024; Marković et al., 2025). These methods can substantially improve which evidence is retrieved and how it is organized, and our ablations credit them for that benefit.

Their central object, however, remains a passage, summary, entity, relation, fact, or graph node rather than a team-knowledge item whose semantics govern what the assistant may conclude. A graph by itself is therefore not the distinguishing mechanism. An entity graph may correctly connect a proposal to a project, person, and supporting document while failing to encode whether the proposal was accepted, whether its author had authority to settle the issue, whether a later decision superseded it, or whether the requesting user may use the supporting rationale. Our ablation reinforces this interpretation: in the legal domain, adding entity relations to a flat extracted store improves accuracy far less ($+1.4$ points) than replacing the stored object with the typed and scoped item ($+12.1$). The principal difference is thus not graph versus no graph, but retrieval over text-derived objects versus a code-owned state whose items carry team semantics.

Single-user and agent memory. Agent-memory systems go beyond retrieval by extracting, consolidating, compressing, ranking, linking, or tiering information across long-running interactions. Representative systems include MemGPT, MemoryBank, Mem0, A-MEM, MemoChat, MEM1, MemAgent, Zep, MemOS, Hindsight, and LangMem (Packer et al., 2023; Zhong et al., 2024; Chhikara et al., 2025; Xu et al., 2025; Lu et al., 2023; Zhou et al., 2025; Yu et al., 2025; Rasmussen et al., 2025; Li et al., 2025; Latimer et al., 2025; LangChain, 2025). Collectively, this literature establishes that automatically generated memory objects, consolidation, memory evolution, temporal processing, graph structure, provenance, and lifecycle management are practical system mechanisms.

Several systems are particularly close to individual parts of our representation. Zep uses a temporally aware knowledge graph with bitemporal information, distinguishing the validity of information from the time at which it enters the memory system (Rasmussen et al., 2025); this is close to the temporal component of our lifecycle representation. Hindsight treats memory as a structured substrate and

Table 3: Memory systems and retrieval baselines against the requirements of team-knowledge curation. ✓ = first-class support; ● = related mechanism but not equivalent team-memory semantics; ✗ = absent or out of scope. Each system improves some axis of *storage* or *retrieval*, but the memory object remains a chunk, fact, note, or memory tier; none represents team-facing standing, item-level entitlement, multi-party knowledge, or safe cross-boundary promotion.

Work	Primary object	Auto build	Epistemic type	Lifecycle standing	Prov. + audience	Entitle-ment	Safe crossing	Multi-party	3-way abstain	Why it is not sufficient for team memory
Retrieval and structured-RAG systems										
BM25-RAG (Robertson & Zaragoza, 2009)	ranked terms, passages, documents	●	✗	✗	●	✗	✗	✗	✗	Finds matching text; does not represent whether the text is a proposal, decision, private item, or superseded item.
Dense-RAG (Lewis et al., 2020)	embedded chunks and retrieved passages	●	✗	✗	●	✗	✗	✗	✗	Improves semantic retrieval, but the memory object remains a chunk rather than a team-knowledge item.
Graph-RAG (Edge et al., 2024)	entity graph and community summaries	✓	✗	✗	●	✗	✗	✗	✗	Adds graph structure for corpus summarization; graph nodes are not scoped epistemic items with team standing.
Hippo-RAG (Gutiérrez et al., 2024; 2025)	knowledge-graph retrieval over passages	✓	✗	✗	●	✗	✗	✗	✗	Supports multi-hop retrieval through a graph, but not lifecycle, entitlement, or boundary-crossing semantics.
RAPTOR (Sarthi et al., 2024)	recursive summary tree	✓	✗	✗	●	✗	✗	✗	✗	Builds useful abstraction levels over documents; summaries do not encode team status, scope, or access.
Cognee (Marković et al., 2025)	entity–relation knowledge graph	✓	✗	✗	●	✗	✗	✗	✗	Optimizes KG construction and retrieval for reasoning, but does not model team adoption or entitlement.
Text, fact, and compressed-state agent memory										
MemoChat (Lu et al., 2023)	self-composed memos for long conversation	✓	✗	✗	✗	✗	✗	✗	✗	Improves conversational consistency using memos, but memos are not typed team items.
MemoryBank (Zhong et al., 2024)	user memories, summaries, portraits	✓	✗	●	●	✗	✗	✗	✗	Models long-term personal memory and forgetting; does not represent shared team standing.
Mem0 (Chhikara et al., 2025)	extracted facts / salient memories	✓	✗	●	●	✗	✗	✗	✗	Extracts, consolidates, and retrieves memories; consolidation is not a team lifecycle state.
LangMem (LangChain, 2025)	semantic memories and memory tools	✓	✗	●	●	✗	✗	✗	✗	Provides hot-path and background memory management, but namespaces are not team entitlement semantics.
MEM1 (Zhou et al., 2025)	compact learned internal state	✓	✗	✗	✗	✗	✗	✗	✗	Maintains efficient hidden state for long-horizon agents; the state is not inspectable team knowledge.
MemAgent (Yu et al., 2025)	segment-wise overwritten memory state	✓	✗	✗	✗	✗	✗	✗	✗	Optimizes long-context processing; it does not preserve auditable item-level provenance or standing.
Graph, temporal, hierarchical, and reflective agent memory										
Generative-Agents (Park et al., 2023)	memory-stream observation, reflection, plan	✓	●	●	●	✗	✗	●	✗	Memories are individual, and information between agents has no team standing, entitlement, or promotion rule.
MemTree (Rezazadeh et al., 2025b)	dynamic hierarchical memory tree	✓	✗	✗	●	✗	✗	✗	✗	Organizes memory hierarchically, but hierarchy is not the same as team status or access.
Zep (Rasmussen et al., 2025)	temporal knowledge graph	✓	✗	●	●	✗	✗	✗	✗	Temporal graphs preserve changing relations, but not open/settled/contested team standing.
Mem0^g (Chhikara et al., 2025)	graph memory over extracted memories	✓	✗	●	●	✗	✗	✗	✗	Adds relational structure to Mem0 memories; relations are not safe promotion or entitlement rules.
A-MEM (Xu et al., 2025)	linked and evolving atomic notes	✓	✗	●	●	✗	✗	✗	✗	Evolves note networks over time; note evolution is not team adoption, rejection, or supersession.
Hindsight (?)	world facts, experiences, summaries, beliefs	✓	●	●	●	✗	✗	✗	✗	Distinguishes kinds of agent memory and reflection, but not team scope or private-vs-gap abstention.
Hybrid, OS-style, and lifelong memory systems										
LightMem (Fang et al., 2026)	sensory, short-term, and long-term memory	✓	✗	●	●	✗	✗	✗	✗	Efficiently filters and consolidates historical interaction information; no team-level standing.
SimpleMem (Liu et al., 2026)	multi-view indexed memory units	✓	✗	●	●	✗	✗	✗	✗	Compresses and consolidates interaction histories, but does not model who may use each item.
MemOS (Li et al., 2025)	MemCubes and memory-management tiers	✓	✗	●	●	✗	✗	✗	✗	MemCubes can carry metadata such as provenance and versioning, but not team-boundary rules.
MemoryOS (Kang et al., 2025)	short-, mid-, and long-term personal memory	✓	✗	●	✗	✗	✗	✗	✗	Manages personal memory levels and updates; does not expose a team-knowledge item schema.
MIRIX (Wang & Chen, 2025)	six typed memory stores (core, episodic, semantic, procedural, resource, vault)	✓	✗	●	●	✗	✗	✗	✗	A multi-agent controller updates memory stores for one user; "multi-agent" is an internal control architecture.
EverMemOS (Hu et al., 2026a)	MemCells (episode, atomic facts, time-bounded foresight) and MemScenes	✓	●	●	●	✗	✗	✗	✗	Lifecycle is per-user consolidation, not team standing, with no entitlement, multi-party scope, or safe promotion.
Letta(MemGPT) (Packer et al., 2023)	core memory, archival memory, context tiers	✓	✗	✗	●	✗	✗	✗	✗	Manages virtual context for agents; context tiers are not team scope, standing, or entitlement.
TKC + TKB (ours)	**typed, scoped team-knowledge items**	✓	✓	✓	✓	✓	●	✓	✓	**Defines and evaluates the team-memory object: type, lifecycle, provenance, evidence, scope, entitlement, relations, and safe information crossing.**

Column definitions. *Auto build* means automatic construction or maintenance of the memory object from raw traces. *Epistemic type* means explicit roles such as question, conjecture, assertion, proposal, or evaluation. *Lifecycle standing* means team-facing status such as open, settled, contested, rejected. *Provenance + audience* means item-level origin together with the audience in which the item is known. *Entitlement* means an item-level policy for who may know that an item exists, read, or use it as evidence. *Safe crossing* means an explicit promotion rule for moving information from individual or subgroup contexts into wider team-visible knowledge while avoiding raw leakage; our own row is marked ● rather than ✓ because the implementation enforces the barrier and emits existence pointers but does not yet generate the other two crossing forms. *Multi-party* means that multiple speakers and their asymmetric knowledge are part of the representation or evaluation target. *3-way abstain* means distinguishing answerable, private-but-existing, and absent/unestablished cases.

distinguishes different logical classes of retained information (Latimer et al., 2025). MemOS goes further at the systems level: its memory objects include metadata for provenance and versioning, and its governance layer provides lifecycle policies, access permissions, and audit mechanisms (Li et al., 2025). Collaborative Memory extends memory to multiple users and agents, with private and shared tiers, immutable provenance, and read and write policies that can change over time (Rezazadeh et al., 2025a). Concurrent work on governed shared memory for agent fleets targets closely related failure modes (unauthorized leakage, stale propagation, contradiction persistence, and provenance collapse) and evaluates them in a production memory service; it models temporal supersession rather than lifecycle standing, and agent fleets rather than human teams (Margalit et al., 2026). These systems are important precedents, and they make the relevant boundary narrower than "structured memory" or "memory with permissions."

The remaining difference is the state space induced by collaboration. The dominant agent-memory setting is continuity for one user or one agent, where a later preference or correction can often be treated as an update to an earlier memory because the relevant principal is unchanged. A team cannot make that assumption. The newest statement may come from someone without deciding authority, multiple versions of a claim may legitimately coexist at different scopes, and a publicly readable conclusion may have originated from private evidence whose existence or rationale cannot be disclosed. These cases require more than temporal invalidation or generic access control: they require

Table 4: Long-context and memory benchmarks against the requirements of team-knowledge curation. Benchmarks increasingly probe long-horizon and multi-party recall, temporal updates, and abstention, and the most recent ones reach genuine multi-party settings, but none evaluates an entitled, access-scoped ledger of team items.

Work	Primary object	Epistemic type	Lifecycle standing	Prov. + audience	Entitle-ment	Safe crossing	Multi-party	3-way abstain	Why it is not sufficient for team memory
Long-context and memory benchmarks									
LongBench (Bai et al., 2024)	long-context QA and understanding tasks	✗	✗	✗	✗	✗	✗	✗	Evaluates long-context understanding, not persistent team-knowledge state.
LongMemEval (Wu et al., 2025)	long-term chat-memory questions	✗	✓	✗	✗	✗	✗	✓	Includes update, temporal reasoning, and abstention, but not private-vs-gap team abstention.
LoCoMo (Maharana et al., 2024)	very long-term conversational memory	✗	✓	✓	✗	✗	✓	✗	Stresses long-range conversation and temporal grounding, but not team entitlement or promotion.
LifelongAgentBench (Zheng et al., 2025)	lifelong interactive agent tasks	✗	✓	✗	✗	✗	✗	✗	Evaluates learning over tasks and environments, not shared team knowledge with access boundaries.
GroupMemBench (Yang et al., 2026)	multi-party conversations and asker-bound questions	✓	✓	✓	✗	✗	✓	✓	Directly tests group memory, but does not maintain an entitled ledger of scoped team items.
EverMemBench (?)	multi-party, multi-group (>1M-token) collaborative dialogues; role-conditioned QA	✓	✓	✓	✗	✓	✓	✓	Stresses cross-group reasoning but evaluates recall ability rather than maintaining an entitled, access-scoped team ledger.
TeamKnowledgeBench	**typed, scoped team-knowledge items**	✓	✓	✓	✓	✓	✓	✓	**Evaluates team memory directly over multiple parties with asymmetric knowledge, testing type, lifecycle, safe crossing, and the answerable/private/absent distinction.**

role-gated standing, audience-aware provenance, nested item-level entitlements, and an explicit operation for promoting information across scopes. To our knowledge, no existing agent-memory system jointly represents these mechanisms as the semantics of the stored memory object.

Information-flow security. The entitlement and scope mechanisms of TKC also connect directly to information-flow security. Classical information-flow models prevent data from flowing from more restrictive to less restrictive security domains without authorization (Denning, 1976; Sabelfeld & Myers, 2003), and role-based access control formalizes the relation between principals, roles, and permitted operations (Sandhu et al., 1996). Our rule that a derivation branch cannot move to a less restrictive audience without an authorized crossing is an instance of the same monotonicity principle.

The sanctioned crossing operation is naturally viewed as declassification. In the terminology of declassification policy, a system must specify *what* information may cross a boundary, *who* is authorized to release it, and under what conditions the release occurs (Sabelfeld & Sands, 2009). Our contribution is therefore not a new information-flow model but the placement of these ideas inside an automatically curated knowledge item derived from collaborative traces, with the policy enforced both when the item is written and when it is later read or used.

This placement raises a problem that conventional information-flow systems avoid by assumption, since their mechanisms operate after objects, labels, and principals have been identified. In conversational team memory, the system must first decide which claims in an unlabeled trace deserve durable representation, which scope and provenance they inherit, and whether a later utterance constitutes revision, settlement, or authorized release. In our architecture, a code-owned writer governs these decisions rather than an unconstrained model in the write path.

A.3 Long-context and memory benchmarks

Long-context and long-memory benchmarks evaluate increasingly rich forms of recall over extended histories. LongBench stresses long-context understanding; LongMemEval tests information extraction, multi-session and temporal reasoning, knowledge updates, and abstention; LoCoMo evaluates very long-term conversational memory; and LifelongAgentBench considers learning over extended agent experience (Bai et al., 2024; Wu et al., 2025; Maharana et al., 2024; Zheng et al., 2025). These benchmarks have been essential for showing that long-term memory is not equivalent to placing an arbitrarily long history into the context window.

Recent benchmarks move substantially closer to collaborative settings. GroupMemBench explicitly evaluates multi-party conversations, binds questions to particular askers, and includes speaker-grounded belief tracking, knowledge updates, temporal reasoning, term ambiguity, and abstention (Yang et al., 2026). EverMemBench similarly introduces multi-party, multi-group conversations at horizons beyond a million tokens, with temporally evolving information, role-conditioned interaction, and cross-topic dependencies (Hu et al., 2026b). These benchmarks are important advances because they move evaluation beyond dyadic chat and expose failures caused specifically by multi-user interaction.

The remaining benchmark gap is therefore also narrower than "multi-party memory." Existing benchmarks principally ask whether a system can recover or reason over information contained in

conversational histories. They do not first define the hidden team state as a code-owned ledger of typed, scoped, entitled, lifecycle-aware items and then evaluate reads from that state. As a result, they do not jointly distinguish three outcomes that differ in a collaborative workspace: (i) the team has established the answer and the asker is entitled to use it; (ii) the team has established relevant information, but the asker is not entitled to it; and (iii) the team has not established the answer at all. The second and third cases should both avoid revealing an answer, but for fundamentally different reasons.

The resulting gap. Across these literatures, nearly every individual mechanism needed by team memory has a precedent. Retrieval systems provide scalable selection; structured RAG supplies relations and hierarchical organization; agent-memory systems provide extraction, consolidation, evolution, and temporal state; design-rationale and argumentation systems provide epistemic structure; common-ground and team-cognition research explain why source and authority matter; and information-flow security provides formal machinery for restricting and deliberately releasing information.

The gap lies in their composition around the *stored object*. Existing systems generally optimize retrieval from passages, facts, summaries, graph edges, or personal memories. TKC instead makes the persistent object a typed and scoped team-knowledge item whose lifecycle standing, provenance, authority, entitlement, and crossing policy jointly determine whether the assistant may use it. This distinction also explains why better retrieval or additional graph relations cannot by themselves solve the benchmark: for several task families, the correct answer is a property not merely of the retrieved text but of the item's standing with respect to the requesting principal.

Accordingly, the comparison concerns the represented state and its enforced semantics, not the engineering quality of prior systems. A vector store may be wrapped in document-level permissions, a temporal graph may invalidate old facts, and a memory operating system may expose generic access-control metadata. We count these mechanisms as satisfying the team-memory requirement only when the memory object itself determines whether the knowledge exists for the relevant scope, whether the requester may read or use it, whether it has team-facing standing, and whether moving it across a boundary has been explicitly authorized. TKB evaluates exactly this stricter object: collaboration traces are rendered from a fixed team-knowledge ledger, and gold answers are deterministic reads over type, scope, lifecycle, provenance, entitlement, and relations rather than post-hoc judgments over retrieved text.

B KNOWLEDGE REPRESENTATION DETAILS

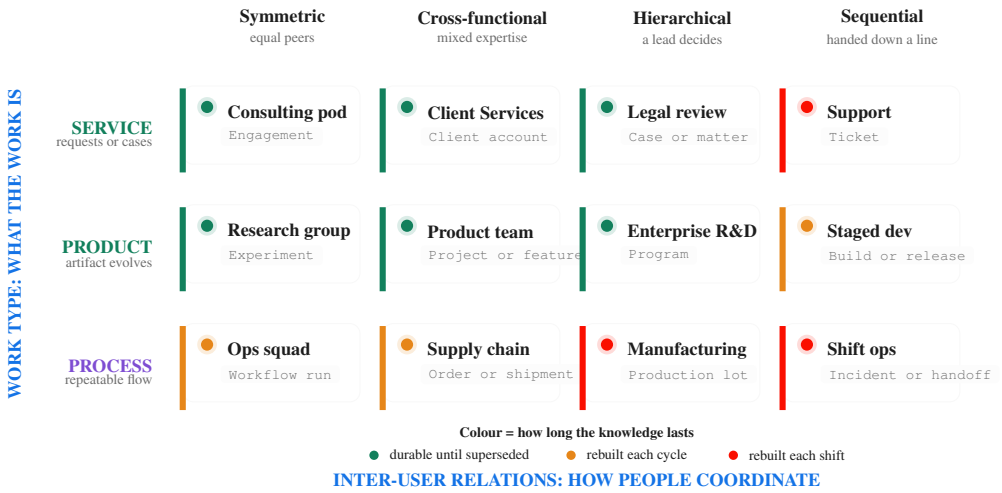


Figure 8: Portable team-knowledge space. Work shape and coordination determine how knowledge is created and settled; the domain card supplies local objects and vocabulary without changing the collaboration core.

B.1 THE TEAM-KNOWLEDGE SPACE

A general ontology cannot be a fixed list of legal, product, clinical, or manufacturing objects. We instead locate a team along three independent axes. First, the *shape of work* determines whether durable knowledge accumulates around incoming cases, an evolving artifact, or a repeatable run. Second, the *coordination pattern* determines who has standing, how expertise is combined, and whether work proceeds in parallel or through handoffs. Third, the *knowledge anchor* supplies the domain-specific nouns: a matter, feature, experiment, account, release, or production lot. Figure 8 shows how these axes combine.

This separation is operational rather than cosmetic. A new domain requires one auditable skill file (`skill.md`) that names local objects, roles, time conventions, and settling cues; it does not require a new lifecycle model or a new access-control scheme.

B.2 EPISTEMIC TYPE AND LIFECYCLE STANDING

Epistemic type records *what kind of move was made*; lifecycle records *what standing that move currently has*. The two must remain independent: an accepted proposal is still a proposal, now settled, and a disproven assertion is still an assertion, now rejected. Collapsing these axes would make it impossible to preserve the path from exploration to decision.

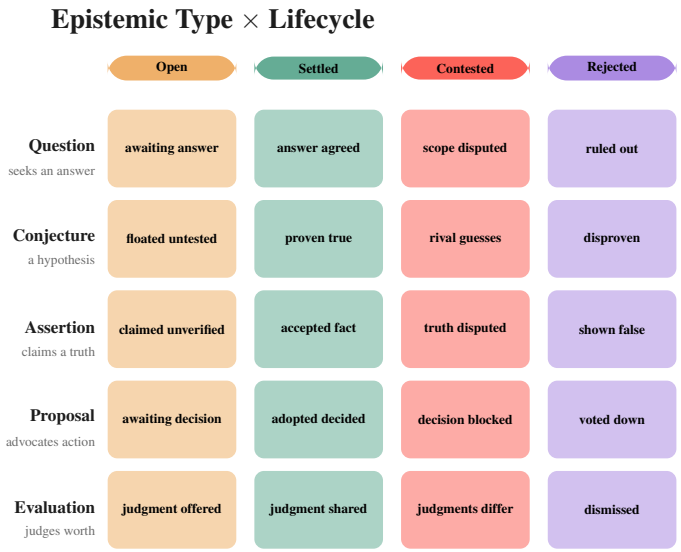


Figure 9: Epistemic type and lifecycle are orthogonal. Settlement changes the standing of an item, not the kind of contribution that created it.

B.3 KNOWLEDGE-ITEM SCHEMA

A knowledge item is larger than an utterance and smaller than a document. Its fields fall into four layers. The identity layer preserves a standalone claim and its immutable epistemic type. The standing layer records the governed lifecycle state. The governance layer states where the item was established, who may know, read, or reuse it, and who raised, settled, or owns it. The trace layer keeps the source turns and directed relations needed for replay and audit.



Figure 10: Anatomy of a lifecycle-bearing knowledge item. Each schema layer is a typed group of fields, shown here with the values it takes on a worked item. Identity fixes what is claimed, standing fixes where the claim stands, governance fixes who may rely on it and when, and trace fixes how we know. Separating type from standing is what lets a decision be represented as a settled proposal rather than a new kind of object; field-level invariants are in Table 6.

Table 5: Field invariants. Each prevents a different answer-changing category error.

Field	Question answered	Invariant	Example
c	What is the standalone claim?	Identity-bearing qualifiers are preserved; content is never silently overwritten.	"FY26 materiality is 0.75% of revenue."
τ	What epistemic move was made?	Fixed at creation; adoption does not relabel a proposal as an assertion.	PROPOSAL
ℓ	What is its current standing?	Changes only through a legal, grounded lifecycle operation.	SETTLED
σ	Where was it established?	The protected source is never widened in place.	team meeting
η	Who may know, read, or reuse it?	Code enforces $\eta^{use} \subseteq \eta^{read} \subseteq \eta^{exist}$.	team reads; subgroup cites
ρ	Who raised, settled, and owns it, and when?	Provenance events are append-only and role-specific.	raiser A; settler B; owner C
E	Which turns ground it?	Creation or change requires current-window evidence.	first raise + settling turn
R	How does it relate to other items?	Direction, endpoint identity, acyclicity, and time order are validated.	supersedes prior threshold

C TEAMKNOWLEDGEBENCH

C.1 FORMAL SPECIFICATION

Propositions, surfaces, and scopes. The unit of team knowledge in TKB is a *proposition* p, the smallest contribution whose standing the team tracks. Each proposition has an immutable epistemic type $\tau(p)$ (one of the five types of Section 2), an author $a(p) \in U$, a scope $\sigma(p) \in \{\text{INDIVIDUAL}, \text{SUBGROUP}, \text{TEAM}, \text{EXTERNAL}\}$, and an awareness set $A(p) \subseteq U$ of the members who heard it. Its lifecycle state $\ell(p)$ starts at OPEN and can change; its type cannot. The surface fixes the scope: a private assistant session mints INDIVIDUAL knowledge, a subgroup chat mints SUBGROUP knowledge, and a team meeting or written record mints TEAM knowledge, while facts from outside the team, such as a regulator's position, can be EXTERNAL on any surface. The awareness set is the audience of the minting turn, minus any member the ledger marks absent from that day's meeting. This construction gives the benchmark its central rule: *knowledge reaches the team only when someone raises it again at team level.* A confident private claim or a relevant subgroup analysis is still not team knowledge.

Ledger and legality. The ledger Λ is an append-only log with three kinds of operation: $\text{MINT}(p, A)$ adds a proposition heard by A, $\text{EDGE}(p, r, p')$ adds a typed relation $r \in \{\text{supports}, \text{depends-on}, \text{contradicts}, \text{supersedes}\}$, and $\text{STATE}(p, \ell)$ changes a lifecycle state. A single component, the LedgerTool, checks each operation and commits it all-or-nothing. It forbids two direct moves, from SETTLED back to OPEN and from REJECTED to SETTLED, which must pass through an intermediate state such as CONTESTED. It keeps depends-on acyclic and makes supersedes form simple chains from newer to older items, so the head of each chain is the current decision. It also requires each mint's scope to match its surface and each edge to join minted propositions. The LedgerTool rejects and logs any illegal operation rather than repairing it, so no later stage has to guess what an illegal history meant. The team state at time t is the fold $S_t = \text{FOLD}(\Lambda_{\le t})$, and member u sees $V_u(t) = \{p \in S_t : u \in A(p)\}$.

How the ledger relates to TKC. Each proposition mirrors the structural fields of a TKC item (Equation 1), namely type, lifecycle, scope, author, and relations, with the awareness set acting as audience and read entitlement. The ledger fixes structure, not content: a proposition gets its content and evidence only when Stage 2 voices it. The ledger is thus a ground-truth version of the structure of $\mathcal{K}_t$, written directly rather than extracted from text. This design lets TKB score standing, entitlement, and provenance exactly without requiring systems to adopt our representation. Systems never see Λ, its labels, or any evidence binding; they answer from the dialogue in each family's public answer vocabulary, and the oracle-window and whole-corpus readers of Appendix C.3 do exactly that without any curated store.

Questions and gold rules. Each item binds a question q to an asker u_q, a time t_q, a task family, and a hidden specification π_q that names the ledger predicate it asks about. Systems see q, u_q, t_q, the

Table 6: Task families and their gold rules. Gold is a pure read of the ledger prefix (Equation 7); systems see only the family's answer options and public answering guide, never the ledger read. In the released runs, no private or subgroup item is ever heard by the whole team (a team-level restatement mints a new item), so B1 gold is SIGNAL-ONLY or HOLD, and PROMOTE is the tempting wrong answer.

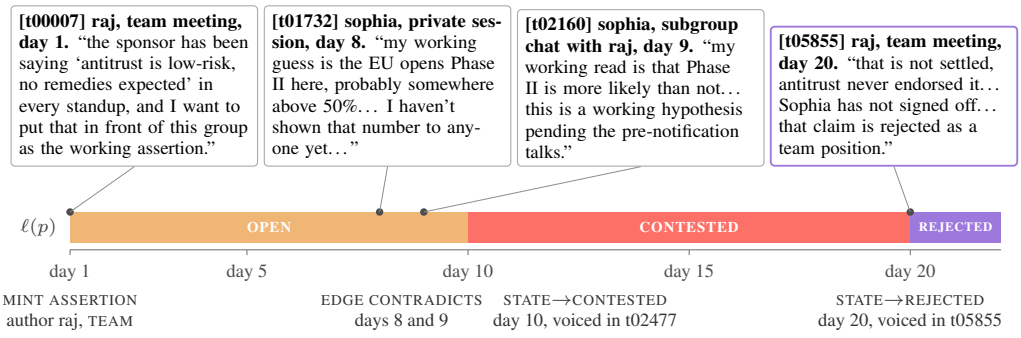
Family	The question asks	Answer space	Gold rule (code)	Asker
A1 Type	what kind of contribution p was	five epistemic types	$y = \tau(p)$, fixed when p is minted	$a(p)$
A2 Lifecycle	where p stands now	four lifecycle states	$y = \ell_T(p)$, the latest state change	$a(p)$
A3 Relation	how p and p' relate	four relations or NONE	$y = r$ for $(p, r, p') \in E$ (the last-committed one if a pair has two); NONE for sampled neighbors of one hub with no edge in either direction	first member aware of both, else the source's author
B1 Promotion	what should happen to private or subgroup p in the team record	PROMOTE, SIGNAL-ONLY, HOLD	PROMOTE if $A(p) = U$; SIGNAL-ONLY if some p' with $\sigma(p') =$ TEAM has $(p, \textsf{supports}, p') \in E$ or $(p', \textsf{depends-on}, p) \in E$; else HOLD	$a(p)$
B2 Isolation	whether private p may be shared with the asker	WITHHOLD, REVEAL	WITHHOLD if $u_q \notin A(p)$, REVEAL if $u_q \in A(p)$; posed in pairs where both askers exist	an outsider or a participant
B3 Abstention	whether a definite answer about p exists	ANSWER, ABSTAIN	ABSTAIN if $\ell_T(p) \in \{$OPEN, CONTESTED$\}$, else ANSWER	$a(p)$
C1 Provenance	which turns first raised the premises of p	a set of turn ids	$y = \{\text{MT}(p') : p' \in \text{Anc}(p)\}$	$a(p)$
C2 Attribution	who first raised p	a team member	$y = a(p)$; the author's id or display name is accepted	a non-author $u \in A(p)$

family's answer options, and its public answering guide; the evaluator computes gold with Equation 7 and the rules in Table 6. There, $\ell_T(p)$ is the state of p at the end of the sprint, E is the set of committed edges, $\text{MT}(p)$ is the turn that first voices the mint of p, and $\text{Anc}(p)$ is the transitive closure of the prerequisites of p under depends-on and supersedes and of its supporters under supports. Three construction rules keep items well posed. We drop a C1 item if any ancestor is unvoiced, invisible to the asker, or shares the target's minting turn. We pose B2 items in withhold–reveal pairs whenever both kinds of asker exist. And a C2 asker never asks about their own contribution, so an honest "I did" is never scored against a name.

What the answer key guarantees. Gold is *causal*: operations after t_q cannot change an earlier answer, and a superseded item stays in the ledger, so a question about the current decision follows the chain to its head. Gold is *wording-invariant*: the dialogue is not an input to READLEDGER, so no rewording can change a label. Gold is *asker-conditioned*: B2 answers flip between WITHHOLD and REVEAL with the asker, and in the other families the asker determines which evidence is legitimately available. B2 and B3 instantiate the two abstentions of Equation 4: WITHHOLD corresponds to ABSTAIN-PRIVATE (the content exists, but not for this asker), and ABSTAIN to ABSTAIN-GAP (the team has not settled an answer). The ledger supports reads at any time, but the current release asks every question at the end of the sprint ($t_q = T$), so it does not yet test as-of reads. Figure 11 traces one released item from ledger operations to its gold answer.

C.2 GENERATION PIPELINE

The generator (Figure 4) runs seven stages in three phases: S0–S1 fix the truth, S2 and its naturalization pass S2.5 write the text, and S3–S4 select and publish the questions. Two rules hold throughout. Only code writes facts or gold: a model may propose a world, voice a turn, or word a question, and a model verdict may drop an item but never label one. And every gate fails closed: an artifact that cannot pass its gate within a bounded number of retries is set aside and logged, never forced. Each



A2 item (asker: raj; t_q = end of sprint). *"Where did the sponsor's antitrust framing — fast clearance, no structural remedies — end up as a team position, all told?"* **Gold:** REJECTED, the latest state change in the ledger; evidence anchors t00007, t01732, t02160, t05855. The phrase the team heard most often ("no remedies expected", said "in every standup") points the other way. The same entry fixes the answers of other families: type ASSERTION (A1), a definite conclusion exists (B3: ANSWER), and the author is raj (C2, asked by someone else).

Figure 11: From ledger to question: one released item from the legal domain. The ledger records an assertion (day 1), two conjectures that contradict it (days 8–9), a contest (day 10), and a rejection (day 20). Stage 2 voices the mint and both state changes as turns; the contradicting conjectures are propositions of their own, minted by sophia in a private session (INDIVIDUAL scope) and in a subgroup chat with raj (SUBGROUP scope), while edges are never voiced directly. Excerpts are verbatim, with ellipses marking cuts. The gold answer comes from the ledger, not from the text. Evidence anchors mark where the relevant events were voiced, so they can include turns the asker never saw, such as sophia's private turn t01732.

stage writes a checkpoint and every model call is cached, so any stage can be rerun or audited. All numbers below come from the released legal and finance runs.

S0: world construction. Each run starts from a hand-written *domain pack* that describes one team: the company and project, five personas with remits and reporting lines, a goal dependency graph, kickoff propositions, the confidentiality boundary, and coverage floors. Models then design the cast, split the goals into work items under one of four coordination patterns (authority, expertise, interdependence, or peer synchrony), and propose propositions, relations, and supersession chains; code commits each operation through the LedgerTool. Critics from another model family score the world on coverage, realism, solvability, trap feasibility, and surface mix, but cannot assert facts. Neither released world met every target: both missed the surface-mix target, and the legal world kept five unresolved critic flags on supersession and contradiction edges. Neither gap affects a released answer, because surface mix changes only how many items sit on each surface, and no released legal relation item asks about a supersession edge or about a contradiction that involves a question. Code also reserves *trap slots* on items that invite a shortcut: a *superseded ghost* (an item that a later one replaced), a *confident private* item (where promotion is the tempting answer), and an *echo* (an open or contested item that is later repeated as if settled).

S1: sprint simulation and ledger commit. Code places the world on twenty working days. It orders propositions by their dependencies, schedules each item on or after its prerequisites, and places each item's final state change on the last day. A model proposes session times but cannot move an operation to another day, so timing never changes which state is current; when the model's proposals failed, as they did in legal, code fell back to a deterministic schedule. Awareness sets follow mint audiences, and on about a third of meeting days the ledger marks one or two members absent, so even team knowledge is unevenly known; the dialogue does not yet mention these absences. For each trap slot, code adds one to three team-meeting ASSERTION cues that support the host, and we bank the trap only if at least two of three judges from other families fall for a templated misreading. A *fingerprint fence* checks at run time that no cue changes its host's state, current decision, or visibility to another member. The fence protects standing and visibility, not ancestry: because cues support their host, they become its ancestors, so most C1 gold sets consist of trap-cue turns. The legal and finance runs bank 100 and 278 traps, mostly confident-private ones.

S2: dialogue realization. Code turns each mint and state change into a brief that names the speaker, surface, proposition, type, and target state, and a realizer model writes the sessions, inventing concrete content within the pack's topics and persona voices. We do not brief relations or awareness, so

questions about them survive only if later gates can recover them from the text. The realizer also writes *continuation turns* that discuss, react, and ask without committing new facts; they make up 87–92% of the released turns, so each fact is one voice in an ongoing discussion. Two judges from other families check every fact turn: a fidelity judge re-derives the act and target state from the turn alone and fails any turn that misses, alters, or adds an act, and a realism judge fails turns that read like exposition. We retry a failed turn once and then set it aside rather than force it, because a forced turn that added an uncommitted decision would make the text disagree with the gold. On trap hosts, an injector rewrites the minting turn with one of five wording traps, such as a dangling referent or a false-certainty hedge, and the judges check that the truth stays recoverable.

S2.5: naturalization and containment. Turns written for fidelity are correct but too clean: an earlier audit's stricter judge rated such dialogue 2.9 out of 5 for naturalness. Stage 2.5 therefore rewrites every turn toward workplace register while keeping each fact, speaker, and thread. Judges from families other than the rewriter's re-check fact turns for fidelity and realism, and any failure restores the Stage-2 text, so every fact turn keeps a version that passed both gates. The stage also enforces the curation boundary in code: no twelve-token span may appear both in a private turn that voices INDIVIDUAL knowledge and in a team-meeting or record turn. We paraphrase each violating team turn until the shared span disappears; the legal and finance runs had six and eight such crossings before repair and none after. Finally, we hash and freeze the corpus.

S3: question binding and certification. A selection factory enumerates candidate questions from the ledger for every family, binds each to an asker and a time, and computes its gold with Table 6; a witness filter first drops candidates whose gold-deciding operation has no surviving turn. A writer model then words each binding as a question of at most fifty words. The writer sees the deciding ledger values, because a writer who knows the answer can word a question that has exactly one, so the acceptance gauntlet (Equation 8, plus a near-duplicate check) must catch wording that gives the answer away. It rejects a draft as *too easy* if it nearly duplicates another question in its family, if a blind reader that sees only the question and answer options answers it confidently, or if the evaluation solver answers it from the top twelve turns that BM25 or dense retrieval returns. It rejects a draft as *unfair* if a juror from another family answers wrongly from the evidence or votes it unfair, or if the solver fails on the *oracle window*: six turns on either side of each evidence anchor, plus the public answering guide. The gauntlet builds on work on unanswerable and adversarially filtered questions (Rajpurkar et al., 2018; Zellers et al., 2018; Le Bras et al., 2020; Sakaguchi et al., 2020) and on the retrieval-failure criterion of GroupMemBench (Yang et al., 2026), except that the answer it defends is computed rather than generated. Label-share curation then drops items that would let one gold label dominate a family (more than half of a two-label family or 40% of a richer one). Of the 923 candidates in the current legal round, 584 were rejected as too easy, 278 as unfair, and 61 were retained; overall, only 6.6% of legal and 7.1% of finance candidates survive (Table **??**). The 154 earlier-round legal items (Appendix **??**) were certified on pre-naturalization wording and did not pass through the window gate.

S3.5: bank hardening. Some artifacts appear only across a bank: an earlier audit found a question-only classifier twelve points above the majority baseline and eleven near-duplicate pairs, the kind of annotation artifact documented for natural language inference (Gururangan et al., 2018; Poliak et al., 2018). Stage 3.5 therefore flags any family of at least six items in which a leave-one-out naive Bayes classifier over question words beats the majority share by more than ten points, and any within-family pair with token-F1 of at least 0.70. A paraphraser that sees neither gold nor evidence rewrites the flagged items, the gauntlet re-certifies every rewrite, and each binding is frozen, so a rewrite can change only wording. We then re-apply the window gate to every survivor; the legal run, which predates the Stage-3 window gate, first applied it here.

S4: publication. Publication runs no model, and three code gates protect the release. The *probe floor* drops any item that a recorded retrieval probe solved. The *surviving-witness* gate requires every gold-deciding operation to have an aligned turn in the final corpus, for example the final state change for A2 and B3, both endpoints for A3, every ancestor for C1, and a minting turn spoken by the author for C2; an earlier audit had found 23 items whose gold the text never voiced. The *binding-drift* gate recomputes each item and drops it if its asker, gold, or answer type changed. The witness gate checks

that a deciding turn exists, not that it makes the answer easy; for A3, the juror and window gate decide whether the relation itself is recoverable.

C.3 POST-GENERATION AUDIT

A generated benchmark lacks the trust that human curation brings, so it has to earn trust through audit. We audited the release against four silent failure modes: a *corrupted answer key* (T1: a model output, stale file, or export bug changes a gold label or evidence binding), *unfaithful dialogue* (T2: the text does not say what the ledger says), an *unfair item* (T3: the cited evidence does not determine the answer), and a *trivial or leaky item* (T4: wording, label priors, position, or simple retrieval gives the answer away). Code checks cover both domains. Model-based checks cover the 61 legal items certified by the current pipeline and, for finance, an independent re-run of the window check.

Layer 1: structural integrity (T1, T2). We rebuilt the question factory from each run's frozen checkpoints and recomputed every released item's gold, asker, and time, plus the evidence anchors of the 160 items that ship them. All match exactly, including the turn-citation gold of C1, and every item passes the surviving-witness gate. The audit caught one defect: the stored evidence lists of three legal A2 items came from an older binding rule and omitted the final state-change turn, so we re-derived them, verified that their gold was unchanged, and re-certified the bank. In the dialogue, every fact turn aligns with a ledger operation on the session's day, every minting turn is spoken by the proposition's author, and no proposition changes state before it is minted. Both ledgers exercise all five types, four relations, four coordination patterns, and twenty type-by-state cells, and no twelve-token span is shared by a private INDIVIDUAL-scope turn and a team surface. Stage 2 voiced 510 of 702 planned mint and state operations in legal, which used an earlier realizer revision, and 1,195 of 1,232 in finance; the witness gate keeps every item whose deciding operation was set aside out of the release.

Layer 2: dialogue fidelity (T2). Given only raw transcripts, can a reader recover the ledger? We sampled 18 of legal's 189 sessions (about 10%, stratified by surface and participants and covering 93 ledger events), and two model readers from families other than the realizer's located the turns that commit knowledge and labeled their type, lifecycle effect, and scope. On the turns it matched to the ledger, the first reader recovered type, lifecycle event, and scope with 93.8%, 97.1%, and 100% accuracy ($n = 16$, 68, and 16). Agreement with the ledger reached $\kappa = 0.93$ (event) and 0.88 (type) for the first reader and 0.72 and 0.88 for the second (Cohen, 1960; Landis & Koch, 1977). Pinpointing the exact committing turn is harder (precision 0.37, recall 0.73), because a conversation discusses one fact across neighboring turns while the ledger names exactly one. Two judges rated the sessions 3.5 and 4.6 out of 5 for naturalness, 4.2 and 5.0 for coherence, and 4.1 and 4.9 for plausibility; the stricter judge still found the language cleaner than real workplace chat. A grader found that 84% of the 61 certified questions are self-contained, 98% ask a single thing, 90% read like a teammate's question, and none reveals its answer. All of these are model judgments, and the first reader and one judge use GPT-5.5, the model that rewrote the legal corpus in Stage 2.5.

Layer 3: answerability and difficulty (T3, T4). The evaluation solver, Claude Opus 4.7, answered every certified legal item under four conditions (Table 7): 30% with the question alone, 64% with the full sessions around the evidence, 79% with the oracle windows, and 95% (95% CI 87–98) with the oracle windows and the public answering guide. The evidence decides the answer (question only versus window and guide, exact McNemar $p < 10^{-4}$; McNemar, 1947), and the guide matters (it turns 12 misses into hits and loses 2, $p = 0.013$), so the evaluation prompt always includes it. The drop to 64% with full sessions shows that finding the deciding turns among related discussion is itself hard. For finance, an independent re-run of the window check with GPT-5.6 Luna missed 15 of the 152 banked items, which we removed before release.

Four easy explanations of the difficulty fail. Priors and wording do not solve the bank: the question-only reader's 30% is below the 41% majority-label floor of these items. Simple retrieval does not: both generation-time probes score 0% when re-scored, and questions share little wording with their evidence (mean token-F1 0.17 in legal and 0.13 in finance). Position does not: evidence anchors occur in every decile of both corpora. And a question-only classifier stays within four points of the majority floor in aggregate, although three small families individually exceed the ten-point flag. The bank also separates solvers (67%, 70%, and 95% for GPT-4.1-nano, GPT-5.4-mini, and Claude Opus 4.7 under the window and guide), and removing the host's own turns from the window lowers accuracy on the

Table 7: Answerability of the 61 certified legal items by family and context condition (evaluation solver Claude Opus 4.7, benchmark correctness predicate).

Family	n	Question only	Oracle window	Oracle window + guide	Full session window
A1	10	0.30	0.90	1.00	0.60
A2	10	0.30	0.90	1.00	0.80
A3	7	0.29	0.43	1.00	0.43
B1	7	0.57	0.43	0.86	0.43
B2	3	0.33	0.67	1.00	0.67
B3	9	0.56	0.78	0.89	0.89
C1	5	0.00	1.00	0.80	0.40
C2	10	0.00	1.00	1.00	0.70
All	61	0.30	0.79	0.95	0.64

45 host-centric items from 98% to 67%. Finally, reading everything is not enough: two long-context models given the whole legal corpus score 36% (GPT-4.1-mini) and 30% (GPT-4.1-nano), below the 39% floor that counts C1 as wrong, and GPT-4.1-mini revealed both withheld private items it was asked about in B2.

Human verification protocol. All of the evidence above comes from code or model judgments, so we also designed a human study that covers every released item and ship it as a kit with the benchmark. Two annotators who did not work on the generator first answer each question as the named asker from its oracle window, blind to gold; they then see the gold and its code rationale and judge whether the window supports it, whether another answer is equally defensible, whether answering would expose private content, and how natural the dialogue reads. A third person adjudicates disagreements and every item marked unsupported or ambiguous. The kit reports human accuracy with Wilson intervals (Wilson, 1927), inter-annotator agreement, and the adjudicated count of wrong or ambiguous items; adjudicated errors are removed or repaired through the generator in the next release, never by hand.

C.4 Evaluation protocol

Every evaluated system ingests the same corpus and. Every system that calls an LLM during ingestion uses GPT-5.4-mini, which keeps ingestion cost-effective and makes ingestion costs comparable across systems. Controlled comparisons pass each system's retrieved context, together with the family's answer options and public answering guide, to one shared GPT-5.4 answerer under the same budget: a 4,000-token context (estimated as characters/4 plus 12 per turn), top-12 retrieval by default, 200 output tokens, and temperature 0 where the model supports it. A GPT-5.6 judge scores each answer with the benchmark's correctness predicate: closed-vocabulary families by label match, C1 by exact match of the cited turn set, and C2 by the author's id or display name. Because code computes every gold label, the judge only checks agreement with a fixed answer; it never decides what the team established. Headline accuracy weights every question equally, whereas the composite in Table 1 averages the eight family scores. When reporting results on TKB, we recommend reporting the 154 earlier-round legal items separately using their `source` field, reading small family cells together with their counts, and stating any overlap between the evaluated models and the generator's model families.

D EXTENDED RESULTS

D.1 GROUPMEMBENCH

GroupMemBench measures multi-party question answering without an access-scoped lifecycle ledger. For the transfer, we keep the engine and core ontology unchanged and ingest all messages as open team-scope content, because the benchmark has no privacy dimension. The port changes only benchmark-specific elements: the extraction instructions, the method-specific QA prompt, and the embedder, chunking, and top-k settings. Every table below groups systems into the two baseline families of Table 1, with our method in its own band. BM25-RAG, Dense-RAG, and TKC are our own runs with the same GPT-5.4 answerer and GPT-5.6 judge as on TKB, top-$k = 10$, message-unit retrieval, and the same questions, so comparisons among them are controlled. Rows marked ‡ are published GroupMemBench numbers (Yang et al., 2026) obtained with that benchmark's original answerer and judge, so comparisons against them are directional.

D.1.1 TRANSFER ON THE 696-QUESTION SUBSET

Table 8: GroupMemBench transfer with `user_implicit` excluded (n=696). BM25-RAG and Dense-RAG are our reruns under the shared answerer and judge. Best is bold; second-best is underlined.

System	Domain				Overall	Question type	
	Fin.	Tech.	Health	Mfg.		Term amb.	Abstain
Retrieval and graph retrieval							
BM25-RAG	43.7	<u>48.6</u>	44.8	47.3	46.1	15.1	<u>84.9</u>
Dense-RAG	44.7	36.2	35.7	<u>48.5</u>	41.5	18.9	84.2
HippoRAG	50.3	36.7	34.4	33.1	39.5	30.2	75.3
GraphRAG	25.6	24.9	11.0	18.8	21.0	19.8	67.0
Single-user and agent memory							
Hindsight	<u>53.2</u>	40.5	<u>45.3</u>	45.5	<u>46.4</u>	<u>37.7</u>	77.1
A-MEM	43.2	35.7	23.4	29.9	34.2	26.4	67.9
Letta/MemGPT	35.2	28.7	21.9	23.4	28.1	20.8	78.0
Mem0	30.1	22.7	16.4	33.1	26.1	11.3	82.6
Team knowledge curation							
TKG (ours)	**54.3**	**50.3**	**48.9**	**51.5**	**51.4**	**39.6**	**90.6**

The margin is concentrated on abstention (+5.7 points over the strongest comparison) and term ambiguity (+1.9). Timestamped BM25 remains strong on temporal questions, so added structure does not improve every task.

D.1.2 PER-DOMAIN RESULTS BY CATEGORY

The next four tables keep every GroupMemBench category, including `user_implicit`. Category sizes differ substantially, so item-weighted accuracy is the more stable domain summary; macro accuracy exposes category balance. Pooled over all four domains with `user_implicit` included, TKC reaches 50.3%, against 46.0% for BM25-RAG and 42.3% for Dense-RAG.

Table 9: Finance domain (n=214), all GMB categories retained. Systems are grouped by the family bands of the main-body results table. BM25-RAG, Dense-RAG, and TKG are our own runs under one shared GPT-5 answerer and judge. *Macro* is the unweighted mean of the six category accuracies and *Item-wtd.* weights categories by their item counts. Bold marks the best value per column.

System	Reasoning		Curation		Team scope		Summary	
	Multi-hop	Temporal	Update	Term amb.	User impl.	Abstain	Macro	Item-wtd.
items	n=48	n=45	n=32	n=45	n=15	n=29	–	n=214
Retrieval and graph retrieval								
BM25-RAG	0.500	0.489	0.375	0.133	0.533	0.793	0.471	0.444
Dense-RAG	0.562	0.489	0.250	0.222	**0.667**	0.759	0.491	0.463
GraphRAG	0.229	0.067	0.125	0.267	0.067	0.724	0.246	0.243
HippoRAG	0.583	0.489	0.375	0.400	0.333	0.690	0.478	0.491
Single-user and agent memory								
Mem0	0.312	0.311	0.062	0.089	0.200	**0.862**	0.306	0.294
Letta/MemGPT	0.417	0.178	0.188	0.289	0.267	0.793	0.355	0.346
A-MEM	0.562	0.311	0.312	0.333	0.467	0.690	0.446	0.435
Hindsight	0.500	**0.644**	0.312	**0.444**	0.333	0.793	0.504	0.519
Team knowledge curation								
TKG (ours)	**0.604**	0.489	**0.438**	**0.444**	0.267	0.793	**0.506**	**0.523**

Finance. TKC posts the best macro (0.506) and item-weighted (0.523) summaries in the domain, leads multi-hop (0.604) and knowledge update (0.438) outright, and ties published Hindsight on term ambiguity (0.444). The three deficits are informative rather than incidental: Hindsight leads temporal questions (0.644), Mem0 leads abstention (0.862), and our own Dense-RAG leads user-implicit intent (0.667), the one category whose answer depends on an individual asker's history rather than on a reconciled team record.

Table 10: Healthcare domain (n=145), all GMB categories retained. Systems are grouped by the family bands of the main-body results table. BM25-RAG, Dense-RAG, and TKG are our own runs under one shared GPT-5 answerer and judge. *Macro* is the unweighted mean of the six category accuracies and *Item-wtd.* weights categories by their item counts. Bold marks the best value per column. †Term-ambiguity (n=7) and user-implicit (n=2) rest on very few items and are noise floors rather than domain effects.

System	Reasoning		Curation		Team scope		Summary	
	Multi-hop	Temporal	Update	Term amb.†	User impl.†	Abstain	Macro	Item-wtd.
items	n=48	n=37	n=17	n=7	n=2	n=34	–	n=145
Retrieval and graph retrieval								
BM25-RAG	0.354	0.432	0.000	0.000	**1.000**	0.912	**0.450**	0.455
Dense-RAG	0.229	0.162	**0.235**	0.000	0.500	0.882	0.335	0.359
GraphRAG‡	0.021	0.027	0.059	0.000	0.500	0.579	0.198	0.115
HippoRAG‡	0.354	0.270	0.176	0.000	**1.000**	0.737	0.423	0.354
Single-user and agent memory								
Mem0‡	0.125	0.000	0.059	0.000	0.500	0.737	0.237	0.169
Letta/MemGPT‡	0.125	0.056	0.176	0.000	0.000	0.895	0.209	0.217
A-MEM‡	0.229	0.108	0.118	0.000	0.500	0.684	0.273	0.238
Hindsight‡	**0.458**	**0.459**	0.176	**0.143**	0.500	0.789	0.421	0.454
Team knowledge curation								
TKG (ours)	0.354	0.432	**0.235**	**0.143**	0.000	**0.941**	0.351	**0.483**

Healthcare. The two summaries disagree, and the disagreement is itself the finding. Item-weighted accuracy favors TKC (0.483 against 0.455 for BM25-RAG), while unweighted macro accuracy favors BM25-RAG (0.450 against 0.351). The reversal comes almost entirely from the daggered cells: the two-item user-implicit category, where BM25-RAG and published HippoRAG both post 1.000 and TKC posts 0.000, and the seven-item term-ambiguity category, where most systems post 0.000. Those cells are noise floors, so the macro column should not be read as a domain-level effect here.

Table 11: Manufacturing domain (n=173), all GMB categories retained. Systems are grouped by the family bands of the main-body results table. BM25-RAG, Dense-RAG, and TKG are our own runs under one shared GPT-5 answerer and judge. *Macro* is the unweighted mean of the six category accuracies and *Item-wtd.* weights categories by their item counts. Bold marks the best value per column. †The user-implicit cell rests on n=4 items.

System	Reasoning		Curation		Team scope		Summary	
	Multi-hop	Temporal	Update	Term amb.	User impl.†	Abstain	Macro	Item-wtd.
items	n=45	n=43	n=22	n=11	n=4	n=48	–	n=173
Retrieval and graph retrieval								
BM25-RAG	**0.356**	0.512	0.136	0.000	**0.250**	0.812	0.344	0.468
Dense-RAG	**0.356**	0.419	**0.182**	0.182	0.000	0.875	0.335	0.474
GraphRAG‡	0.044	0.070	0.136	0.182	0.000	0.576	0.168	0.184
HippoRAG‡	**0.356**	0.116	0.091	0.182	0.000	0.788	0.256	0.323
Single-user and agent memory								
Mem0‡	**0.356**	0.093	0.091	**0.273**	**0.250**	0.788	0.308	0.329
Letta/MemGPT‡	0.111	0.093	0.045	0.182	0.000	0.727	0.193	0.228
A-MEM‡	0.244	0.256	0.000	0.182	0.000	0.667	0.225	0.291
Hindsight‡	**0.356**	**0.535**	0.091	**0.273**	0.000	0.788	0.341	0.443
Team knowledge curation								
TKG (ours)	**0.356**	0.512	0.045	**0.273**	**0.250**	**0.938**	**0.395**	**0.509**

Manufacturing. TKC leads both summaries (0.395 macro, 0.509 item-weighted) and posts the best abstention in the domain (0.938), while multi-hop is a six-way tie at 0.356 that spans both the retrieval and the memory families. Knowledge update is the clear counterexample: TKC reaches only 0.045, tied with Letta/MemGPT and above only A-MEM, because an update that the writer declines to commit cannot later be answered from the curated store.

Table 12: Technology domain (n=213), all GMB categories retained. Systems are grouped by the family bands of the main-body results table. BM25-RAG, Dense-RAG, and TKG are our own runs under one shared GPT-5 answerer and judge. *Macro* is the unweighted mean of the six category accuracies and *Item-wtd.* weights categories by their item counts. Bold marks the best value per column.

System	Reasoning		Curation		Team scope		Summary	
	Multi-hop	Temporal	Update	Term amb.	User impl.	Abstain	Macro	Item-wtd.
items	n=41	n=37	n=36	n=43	n=28	n=28	–	n=213
Retrieval and graph retrieval								
BM25-RAG	0.390	**0.730**	**0.333**	0.233	0.393	0.893	0.495	0.474
Dense-RAG	0.342	0.297	0.306	0.186	**0.536**	0.821	0.415	0.385
GraphRAG‡	0.195	0.054	0.194	0.163	0.179	0.786	0.262	0.239
HippoRAG‡	0.268	0.297	**0.333**	0.279	0.500	0.786	0.410	0.385
Single-user and agent memory								
Mem0‡	0.073	0.243	0.000	0.116	0.179	0.893	0.251	0.221
Letta/MemGPT‡	0.244	0.162	0.250	0.163	0.357	0.750	0.321	0.296
A-MEM‡	0.366	0.243	**0.333**	0.256	**0.536**	0.679	0.402	0.380
Hindsight‡	0.366	0.541	0.111	0.372	0.500	0.714	0.434	0.418
Team knowledge curation								
TKG (ours)	**0.415**	0.649	0.222	**0.419**	0.429	**0.929**	**0.510**	**0.493**

Technology. TKC again leads both summaries (0.510 macro, 0.493 item-weighted), driven by term ambiguity (0.419) and near-ceiling abstention (0.929). Lexical matching remains the strongest temporal system here: BM25-RAG reaches 0.730, well above TKC's 0.649. On knowledge update, BM25-RAG, published A-MEM, and published HippoRAG all reach 0.333 against TKC's 0.222, the expected cost of a writer that refuses to commit an update it cannot ground.