COGMEM: COMMUNICATION-BUDGETED COLLABORATIVE GRAPH RETRIEVAL FOR AGENT-PRIVATE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent LLM systems are rapidly shifting from a single model to fleets of agents, where each agent maintains a *private*, graph-indexed long-term memory over a temporal knowledge corpus. When a question spans several agents, no single agent holds the complete answer and the team must retrieve it jointly, raising the problem of *distributed multi-agent memory retrieval*: producing a globally accurate answer without moving raw data off its partition and within a hard per-query communication budget. Existing paradigms each fail on one category: independent agents are trapped in information islands, centralized pooling violates partition privacy, and full communication wastes bandwidth. In this paper, we propose **CoGMem** (*Collaborative Graph Memory*), a collaborative graph neural network framework in which each agent encodes its private heterogeneous graph partition with a per-edge-type message-passing encoder. The encoder's confidence and relevance signals directly drive the communication protocol: a complementarity-aware gate and a budgeted top-K gossip decide *what* to send and *how much*. A multi-round negotiation extension (CoGMem-n, n for negotiation) then spends the remaining budget on measured information gaps via need-vector-triggered, Δ/cost-greedy follow-ups. Unlike prior work that treats retrieval and communication as separate layers, CoGMem couples the two end-to-end. On two knowledge-augmented memory benchmarks, CoGMem-n matches the High Communication ceiling's hit@1 at 38% of its bytes on Episodic Memory and exceeds every lexical and system-like distributed baseline by over 30 pp hit@1 ($p<0.001$). Causal ablations confirm that the learned GNN stack is load-bearing; on GTSQA, negotiation rescues hit@1 from 0 to 0.266 on the 20.9% hardest queries, delineating both the capability and the boundary of budgeted collaborative retrieval.

1 INTRODUCTION

Large language model (LLM) agents are evolving from a single model into *fleets* of cooperating agents, each responsible for its own data stream (personal notes, department documents, or device logs) and each maintaining a long-term memory over that stream (Park et al., 2023; Sumers et al., 2023). A converging line of work indexes memory *as a graph* (entities, facts, and temporal relations), an effective substrate (Rasmussen et al., 2025; Rajesh et al., 2026; Gutiérrez et al., 2024; 2025; Chhikara et al., 2025; Xu et al., 2025; Jiang et al., 2026). Yet these systems serve a *single* owner, while realistic corpora are born distributed: an enterprise's departments cannot pool their documents, and a user's devices may not export raw data. Answering the questions that make fleets valuable (those spanning several agents) therefore requires agents to *retrieve jointly* from memories that must remain private.

We study *distributed multi-agent memory retrieval*: given a query q and N agents, each holding a private partition of a heterogeneous temporal knowledge graph, the team must produce a globally accurate top-K candidate list while (i) no raw node or edge ever leaves its partition and (ii) total inter-agent traffic stays within a hard per-query byte budget. Three paradigms dominate practice, and each fails on a different category (Figure 1): *local* agents never communicate and are trapped in information islands; *global/centralized* pooling restores recall but violates partition privacy (Dean et al., 2012); *high communication* broadcasts each agent's entire local candidate list at maximal band-

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.

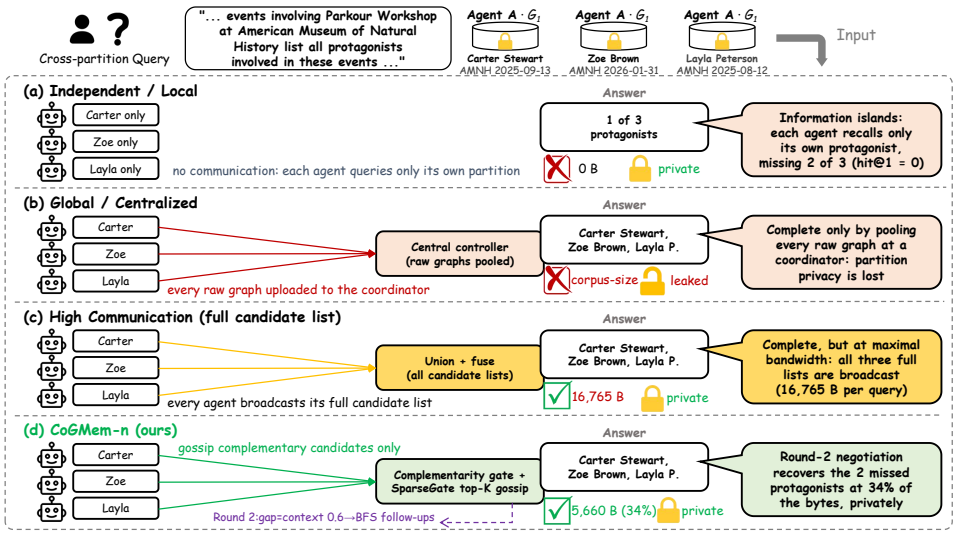


Figure 1: **Existing paradigms vs. CoGMem-n** for distributed multi-agent memory retrieval, on a real Episodic Memory test query that requires round-2 negotiation (*"... Parkour Workshop at American Museum of Natural History ... list all protagonists..."*; gold: Carter Stewart, Zoe Brown, Layla Peterson). (**a**) **Independent**: each agent recalls only its own protagonist (hit@1 = 0). (**b**) **Centralized**: complete, but privacy is lost. (**c**) **High Communication (full candidate list)**: complete at maximal bandwidth (16,765 B on this query). (**d**) **CoGMem-n**: gap-triggered round-2 negotiation recovers the full answer at 5,660 B (34%), privately; on average it matches High Communication's hit@1 (0.419) at 38% of its bytes.

width. Algorithmically, GNN-based retrieval is mature *within one graph* (Mavromatis & Karypis, 2025; He et al., 2024; Sun et al., 2024; Luo et al., 2024) but assumes a single, fully visible graph. On the systems side, multi-agent memory is emerging (Zhang et al., 2025; Rezazadeh et al., 2025; Gao & Zhang, 2024; Ye et al., 2025), yet these works assume a shared store, a privileged coordinator, or infrastructure-level access rather than bounded retrieval-level exchange. Recent benchmarks (Huet et al., 2025; Cattaneo et al., 2025) likewise do not quantify what a team can retrieve when privacy and bytes are first-class constraints.

To see why existing paradigms fail, consider a real test query from the Episodic Memory benchmark (Huet et al., 2025), whose rows are partitioned across agents along event boundaries (Fig. 1). The question "Think about the events involving Parkour Workshop at American Museum of Natural History; list all protagonists involved in these events" requires three event rows held by different agents: *Agent A* holds Carter Stewart's Parkour Workshop event, *Agent B* holds Zoe Brown's, and *Agent C* holds Layla Peterson's. **(i) Information islands (local):** independent agents never exchange messages, so each agent names only its own protagonist, and recall collapses exactly on cross-agent questions. **(ii) Privacy and bandwidth (global):** shipping every partition to a coordinator resolves the answer but violates partition isolation and costs bandwidth that grows with the corpus, not with the query. **(iii) Cost without control (high communication):** broadcasting every agent's full candidate list recovers the answer at maximal byte cost, and communication without a fusion policy can even be actively harmful: randomly exchanged candidates push hit@10 *below* the zero-communication baseline. The missing paradigm lets the retriever's own confidence and relevance signals decide what is worth sending, and spends the budget adaptively where the team's answer is measurably incomplete.

In this paper, we propose **CoGMem** (*Collaborative Graph Memory*) [1], a collaborative graph neural network framework for distributed multi-agent memory retrieval whose guiding principle is *retrieval-driven, budgeted communication*. Each agent encodes its private partition with a heterogeneous GNN over real text embeddings (per-edge-type message passing, type/time heads, and query-conditioned seed injection, trained contrastively), and the resulting confidence and relevance scores

[1] Source code will be released when published.

drive the protocol directly: a complementarity gate promotes candidates the receivers lack and suppresses redundant ones; a budgeted SparseGate top-K gossip bounds every message; a cross-agent reranker with *self-vote protection* produces the global list; and a multi-round negotiation extension (**CoGMem-n**, n for negotiation) spends the remaining budget only where a gap assessor measures incompleteness, via need-vector-triggered, Δ/cost-greedy follow-ups with gain-proxy early stopping. In the Parkour Workshop example, round-1 gossip already carries complementary protagonist edges, and the gap assessor then measures a residual *context* gap (0.6) and fires a round-2 targeted negotiation that recovers the full answer at 34% of high-communication bytes on this query (38% on average).

Concretely, the framework resolves three challenges that prior paradigms leave open. **C1 (learnable retrieval over private partitions):** each agent must rank its own graph with real semantics and structure, without ground-truth-derived features; our per-edge-type encoder with InfoNCE training is causally load-bearing (ablations NC1/NC2, up to -35.0 pp hit@1, §5.3). **C2 (what and how much to communicate):** bytes must go to complementary, not redundant, candidates, and cross-agent trust must be calibrated; our complementarity gate, SparseGate, and self-vote protection ($\lambda_{\text{self}} \geq 0.45$ is necessary, §5.3) answer this. **C3 (how to spend budget across rounds):** follow-ups must target measured gaps rather than blanket re-queries; our negotiation loop formalizes this as budgeted submodular maximization with a $(1-1/e)$-approximate greedy rule (§4.5). Our contributions are threefold. **(1) A new paradigm.** To our knowledge, CoGMem is the first framework to formulate distributed multi-agent memory retrieval as *communication-budgeted collaborative GNN retrieval over agent-private graph memories*, coupling learned retrieval signals with inter-agent communication decisions end-to-end and making bytes a first-class, measurable resource. **(2) Method and theory.** We contribute the per-edge-type heterogeneous encoder with seed injection, the complementarity gate and SparseGate gossip, the self-vote-protected cross-agent reranker, and the need-vector multi-round negotiator, with three guarantees: a corpus-independent, linear-in-N communication bound, a $(1-1/e)$ approximation for the budgeted greedy negotiator, and a top-1 invariance condition explaining when budget truncation is free. **(3) Experiments.** Under a strict pure-retrieval protocol with per-question paired bootstrap significance, on two benchmarks with 12 baselines we show that CoGMem-n matches High Communication's hit@1 at 38% of its bytes on Episodic Memory and exceeds every lexical and system-like distributed baseline by ≥ 30 pp hit@1 ($p < 0.001$); we map the operating points and report the boundaries honestly: negotiation rescues hit@1 from 0 to 0.266 on the 20.9% hardest GTSQA queries, but does not uniformly improve coverage or save bytes on small graphs.

2 Related Work

Graph-indexed memory and graph retrieval. Giving LLMs an explicit external memory is an established design (Packer et al., 2023; Park et al., 2023; Sumers et al., 2023; Yao et al., 2023), and production memory layers scale it up (Chhikara et al., 2025; Xu et al., 2025). A converging line indexes memory *as a graph*: temporal knowledge graphs for agent memory (Rasmussen et al., 2025), generative semantic workspaces for episodic RAG (Rajesh et al., 2026), hippocampus-inspired indexing for continual learning (Gutiérrez et al., 2024; 2025), and multi-graph agentic memory (Jiang et al., 2026); all assume a single owner with a centralized index. On the retrieval side, RAG couples retrievers with generators (Lewis et al., 2020; Gao et al., 2023), with sparse, dense, and graph-lifted variants (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Edge et al., 2024); GNN retrievers build on message passing (Gilmer et al., 2017; Kipf & Welling, 2017; Hamilton et al., 2017; Schlichtkrull et al., 2018; Veličković et al., 2018; Xu et al., 2019); GNN-RAG, G-Retriever, and Think-on-Graph/RoG use GNNs or traversal *as retrievers* over one fully visible graph (Mavromatis & Karypis, 2025; He et al., 2024; Sun et al., 2024; Luo et al., 2024). They define no communication protocol *between* agent-private partitions, the gap our gossip protocol fills.

Multi-agent memory, communication, and benchmarks. G-Memory traces hierarchical memory across collaborating agents (Zhang et al., 2025), Collaborative Memory adds multi-user access control over a shared store (Rezazadeh et al., 2025), Memory Sharing exchanges episodic records (Gao & Zhang, 2024), and KVCOMM reuses KV-cache state across agents (Ye et al., 2025), echoing classical bandwidth concerns (Dean et al., 2012). These works share a store, delegate to a coordinator, or operate at infrastructure level; none treats inter-agent communication as a *budgeted retrieval*

resource justified by measured incompleteness. On benchmarks, EpBench evaluates temporally grounded episodic recall (Huet et al., 2025) and GTSQA tests zero-shot subgraph retrieval (Cattaneo et al., 2025); both target centralized retrievers, and we adapt them to the distributed setting with controlled partitioning (§5.1). CoGMem thus differs from all three lines: retrieval across *private* partitions, a *cross-partition* protocol driven by the retriever's own signals, and *no shared store or privileged coordinator*, with privacy and bytes as first-class constraints and the full trade-off map reported, including negative results.

3 Preliminary and Problem

Notation and heterogeneous graph partitions. A corpus is partitioned into N partitions owned by agents $\{a_i\}_{i=1}^{N}$. On partition i we build a heterogeneous graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{T}_V, \mathcal{T}_E)$ with node types $\mathcal{T}_V$ (`Entity`, `Fact`) and edge types $\mathcal{T}_E$ (entity–fact provenance edges, `chrono` temporal edges, and `SAME_AS`, a schema-level relation linking nodes that refer to the same real-world object across partitions: for example, if the protagonist *Carter Stewart* appears in two agents' partitions, a `SAME_AS` edge connects his two `Entity` nodes, so cross-partition identity is made explicit in the graph rather than left to string matching). Every node and edge carries raw label text, embedded once by a frozen 768-d `nomic-embed-text` encoder (no ground-truth-derived features). Every forward edge has a matching reverse edge, and `SAME_AS` bridges are written symmetrically, so cross-partition entity identity survives any partitioning strategy.

Problem definition. Given a query q, each agent a_i computes a local ranked candidate list $h_i(q) \subseteq \mathcal{E}_i$ using only its own partition. A distributed retrieval protocol then exchanges messages $\{\varepsilon_{i\to i'}\}$ and produces a global top-K list via a fusion operator $\mathcal{F}$. Two constraints are first-class. **Privacy:** the *inventory* of $\mathcal{G}_i$ (its full node/edge set, raw texts, and features) never leaves agent i; only the bounded message $\varepsilon_{i\to i'}$ (a learned summary plus a few selected candidates) is disclosed. **Budget:** the total per-query inter-agent traffic obeys

$$\sum_{i\neq i'} \mathrm{Bytes}(\varepsilon_{i\to i'}) \;\le\; B, \tag{1}$$

where B is a hard per-query byte budget. The objective is to maximize retrieval accuracy (hit@k and MRR against the ground-truth answer set $G(q)$) under these constraints. High communication broadcasts every local candidate (maximal B); independent retrieval is the protocol with $B{=}0$. The central question is *where between these extremes useful operating points lie*, and what mechanism finds them.

Complementarity and redundancy. For a candidate edge e sent from agent j to agent i, we call e *complementary* if $e \notin \mathcal{E}_i$ (it carries information i lacks) and *redundant* if $e \in \mathcal{E}_i$. Dataset-level, over each top-K list:

$$\mathrm{Comp} = \frac{|\{e \in \text{top-}K :\; e \notin \mathcal{E}_i\}|}{K}, \qquad \mathrm{Red} = \frac{|\{e \in \text{top-}K :\; e \in \mathcal{E}_i\}|}{K}. \tag{2}$$

Complementarity is what the communication policy should maximize per byte sent; redundancy is what high communication wastes budget on.

4 Methodology

CoGMem's pipeline (Figure 2) consists of three local, privacy-preserving phases plus a negotiation loop. In **Phase 1 (local encoding)**, each agent runs a heterogeneous-GNN encoder over its private partition and emits a compact summary $\mathbf{s}_i \in \mathbb{R}^{128}$ and a local top-K candidate list with per-edge confidence $c_{i,e}$ and relevance $r_{i,e}$. In **Phase 2 (budgeted gossip)**, a complementarity gate re-scores candidates by what receivers lack, and SparseGate selects a top-K payload under the byte budget; only summaries and selected candidate identifiers cross partition boundaries. In **Phase 3 (fusion)**, a cross-agent summary-attention reranker with rank fusion and self-vote protection emits the global list. If a gap assessor then measures that the answer is still incomplete, **the negotiation loop** (**CoGMem-n**, n for negotiation) triggers bounded targeted follow-up rounds before fusion repeats.

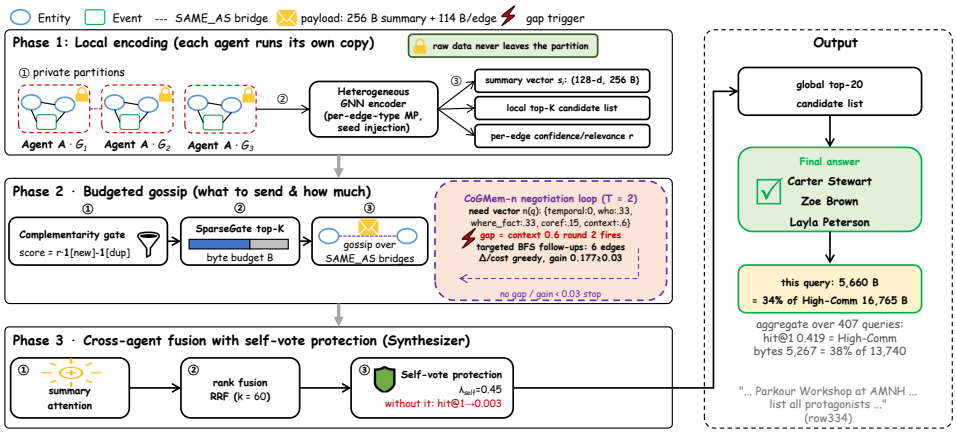


Figure 2: **Overview of CoGMem and its negotiation extension CoGMem-n. Phase 1 (top):** private partitions are encoded locally; raw data never leaves the partition. **Phase 2 (middle):** complementarity-gated SparseGate gossip under a byte budget over `SAME_AS` bridges; CoGMem-n issues follow-ups only on measured gaps. **Phase 3 (bottom):** summary-attention fusion with RRF (k=60) and self-vote protection (λ_{self}=0.45) emits the global top-20. The right panel traces the running example of Fig. 1.

4.1 Local Heterogeneous GNN Encoder

Each agent's encoder is the only component that ever sees raw partition data: relation types become separate convolutions, the query conditions propagation through seed injection, and contrastive training (§4.3) replaces any hand-built or ground-truth-derived features.

Nodes and edges start from frozen 768-d `nomic-embed-text` embeddings of their label text, mapped by a learned projection (`node_proj`) to d=128 dimensions; a type head adds node-type embeddings and a time head adds sinusoidal encodings of edge timestamps. Message passing runs per edge type $\tau \in \mathcal{T}_E$ with an independent aggregation operator (`RelationEncoder`):

$$\mathbf{h}_v^{(l+1)} = \sigma\Big(\mathbf{W}_\tau \left[\mathbf{h}_v^{(l)} \,\|\, \text{AGG}_{u\in\mathcal{N}_\tau(v)}\, \mathbf{h}_u^{(l)}\right]\Big), \quad \tau \in \mathcal{T}_E, \tag{3}$$

where $\|$ is concatenation, σ a non-linearity, $\mathbf{W}_\tau$ an edge-type-specific projection, and $\mathcal{N}_\tau(v)$ the τ-typed neighborhood. Unlike a monolithic RGCN with shared relational weights (Schlichtkrull et al., 2018), this keeps provenance, `chrono`, and `SAME_AS` semantics as independent convolutions. Before propagation, *seed injection* writes the query embedding into the states of seed nodes selected by embedding similarity, so message passing is query-conditioned rather than purely structural. The encoder outputs, per candidate edge e, a structure-based confidence $c_{i,e}$ and a query relevance $r_{i,e}$.

4.2 Complementarity Gate and SparseGate

These two components decide *what* an agent sends and *how much*: the gate operationalizes "only speak when you add new information", and SparseGate turns that into a hard byte bound.

Before gossip, agent j's candidates are re-scored from each receiver i's perspective:

$$\text{score}(e) = \mathbb{1}[e \notin \mathcal{E}_i]\, r_{j,e} \;-\; \gamma\, \mathbb{1}[e \in \mathcal{E}_i], \tag{4}$$

where the complementary mask $\mathbb{1}[e \notin \mathcal{E}_i]$ promotes information i lacks and the redundant mask $\mathbb{1}[e \in \mathcal{E}_i]$ penalizes what i already holds. SparseGate then selects the payload under the budget:

$$h_i(q) = \underset{e:\ \text{Bytes}\le B}{\arg\max}\ \text{score}(e) \qquad \text{s.t. } |h_i(q)| \le K. \tag{5}$$

Because `SAME_AS` bridges are registered in both partitions, complementarity is computed against the receiver's *effective* inventory, making the protocol robust to the partitioning strategy.

4.3 Cross-Agent Reranker with Self-Vote Protection

Fusion weighs per-agent trust against rank consistency, and must *not* let remote votes override a confident local answer, a failure we measure and then prevent by design.

Let $\mathrm{S} \in \mathbb{R}^{N\times 128}$ stack the per-agent summaries and let d be a candidate. A single cross-attention layer produces per-agent reliability weights

$$a_{\mathrm{S}}(i,d) = \frac{\exp\big(\mathbf{q}_d^\top \mathbf{k}_i/\sqrt{128}\big)}{\sum_j \exp\big(\mathbf{q}_d^\top \mathbf{k}_j/\sqrt{128}\big)}, \quad \mathbf{k}_i = \mathbf{W}_k \mathbf{s}_i, \tag{6}$$

and rank fusion (RRF, k=60) aggregates list positions:

$$\mathrm{RRF}(d) = \sum_{\ell} \frac{1}{k + \mathrm{rank}_\ell(d)}. \tag{7}$$

The cross-agent signal $\mathrm{cross}(d)$ combines a_{S} and RRF; the final score adds the home agent's own calibrated score $\mathrm{self}(d)$ with a protection weight:

$$\mathrm{FinalScore}(d) = \lambda_{\mathrm{self}} \cdot \mathrm{self}(d) + (1-\lambda_{\mathrm{self}}) \cdot \mathrm{cross}(d). \tag{8}$$

We lock λ_{self}=0.45 and analyze the mechanism in Theorem 4.3 (pipeline: Algorithm 1, Appendix B).

Learning objective. The encoder is trained contrastively so that query-conditioned embeddings separate answer-bearing edges from distractors; no ground-truth structure ever enters the input features. We optimize an InfoNCE loss over candidate edges:

$$\mathcal{L}_{\mathrm{InfoNCE}} = \sum_{i,e^+} \Big[-\log \frac{e^{\,\mathrm{sim}(\mathbf{z}_i,\mathbf{z}_{e^+})/\tau}}{e^{\,\mathrm{sim}(\mathbf{z}_i,\mathbf{z}_{e^+})/\tau} + \sum_{e^-} e^{\,\mathrm{sim}(\mathbf{z}_i,\mathbf{z}_{e^-})/\tau}} \Big], \tag{9}$$

where τ is a temperature and $e^{\pm}$ are positive/negative candidates (logits in float32 for stability). The complementarity signal of Eq. 2 enters at inference time through the gate (§4.2), so the protocol, not the training data, determines what crosses partition boundaries.

4.4 Multi-Round Negotiation Protocol

Single-round gossip suffices when answers concentrate in few agents, but hard multi-hop questions spread across the fleet. CoGMem-n replaces the fixed single round with a T-round loop that spends budget *where incompleteness is measured*, via five mechanisms.

Need vector. From q we extract a 5-dim need vector $\mathbf{n}(q) \in [0,1]^5$ over {temporal, who, where, coreference, context}. A regex extractor is the default; an LLM extractor is an ablated alternative with identical downstream logic (Appendix J).

Gap assessor. After Round 1, each agent measures, for every need dimension d, the *residual gap* $g_d = 1 - \mathrm{coverage}_d \in [0,1]$ between what the query requires and what the fused candidates cover. A new round triggers iff $\max_d g_d > \delta$, δ=0.35; that is, follow-ups launch only when some required facet is measurably under-covered, so cost tracks genuine incompleteness, not fleet size.

Targeted BFS retriever and greedy negotiator. On a triggered round, each agent gathers candidates *only* along above-gap dimensions via bounded breadth-first expansion from its current best candidates. Within the remaining budget, candidate edges are selected by a Δ/cost greedy rule: each candidate's marginal utility on the need dimensions is divided by its byte cost, and re-selecting from an already-covered dimension is penalized by a marginal-diminishment factor 0.5, a discrete analogue of diminishing returns that Theorem 4.2 makes precise.

Gain-proxy early-stop. Negotiation halts when the estimated marginal gain of unselected edges falls below 0.03 or the budget is exhausted. By default the protocol runs at most T=2 rounds per query: Round 1 is the initial gossip (top K_1=2 candidates per agent), and Round 2 is the follow-up negotiation round, launched only when the gap assessor reports a residual gap above the threshold (Algorithm 2, Appendix B).

4.5 Theoretical Analysis

Three guarantees explain the empirical operating points (proofs in Appendix A).

Theorem 4.1 (Communication bound)**.** *Under SparseGate with summary dimension d, per-round candidate caps K_t, T rounds, and N agents, per-query inter-agent traffic satisfies* $\text{Bytes} \leq \sum_{t=1}^{T} N(2d + K_t e_{\max}) = O(TN(d + Ke_{\max}))$*, independent of corpus size $|\mathcal{E}|$, where $e_{\max}$ is the byte cost of one candidate identifier.*

Theorem 4.2 (Negotiation as budgeted submodular maximization)**.** *Define the need-weighted coverage utility of an acquired edge set S as $f(S) = \sum_{d=1}^{5} w_d \min(1, \sum_{e \in S} \text{cover}(e, d))$ with $w_d = \mathbf{n}(q)_d$ and $\text{cover}(e, d) \in [0, 1]$. Then f is monotone submodular, and the Δ/cost greedy negotiator, combined with the best single candidate, achieves at least $(1 - 1/e)$ of the optimal utility under the byte budget B.*

Theorem 4.3 (Self-vote top-1 protection and budget-saturation invariance)**.** *Let a^* be the correct answer, the local top-1 of its home agent with normalized self score $\text{self}(a^*) = 1$, and let $x_{\max}$ be the largest cross-agent signal any competitor can accumulate. Then for any $\lambda_{\text{self}} \geq x_{\max}/(1 + x_{\max})$, the fused top-1 of Eq. 8 is a^* regardless of how cross-agent votes distribute. If additionally every answer-bearing candidate lies within its owner's local top-k_{local} list and the byte budget exceeds the saturation point B^* transmitting all such lists, the budgeted protocol's hit@1 equals High Communication's.*

5 Experiments

5.1 Experimental Setup

Evaluation protocol. All results are *pure retrieval*: every method is evaluated on its own retriever's candidate pool, with *no ground-truth edge injection* and no mock features. **Datasets.** *GTSQA* (Cattaneo et al., 2025) (full-strict test split, n=1622) evaluates subgraph retrieval against ground-truth answer subgraphs under unseen relation types, with small, mutually exclusive micro-partitions. *Episodic Memory* (Huet et al., 2025) (full test, n=407) measures temporally-aware event recall over long narratives partitioned along event boundaries (temporal partitioning). Results use N=3 real partitions unless stated (N-sweep protocol in Appendix K). **Models.** All graph features are built from frozen 768-d `nomic-embed-text-v1.5` sentence embeddings, and no method calls an external API: every LLM-dependent step is served by a local on-premise Ollama endpoint: `deepseek-r1:14b` for the `full_llm` graph-extraction ablation and `deepseek-r1:7b` for the LLM need-vector ablation (Appendix J). The main results use the regex need extractor and require *no* generative LLM calls at all, so the gains cannot be attributed to any particular language model.

Baselines. We compare against 12 baselines spanning *Local*, *Global*, and *Distributed* paradigms (rows of Tables 1–2). *Random Candidate* samples edges uniformly from the local partition, a no-retrieval lower bound. *Partition BM25* retrieves lexically within each partition and never communicates; *Global BM25* runs the same lexical retriever over the pooled corpus as a no-privacy reference. *Random-Peer BM25* requests one random peer's local top-20 hit list, isolating the effect of unstructured communication; *All-Peer BM25* requests every peer's local list, the lexical analogue of maximal communication. The lexical baselines thus form a peer-coverage axis: no peers, one random peer, or all peers. *Graphiti-like*, *GSW-like*, and *ARTEM-like* re-implement the retrieval cores of three recent agent-memory systems Rasmussen et al. (2025); Rajesh et al. (2026); Tan et al. (2026) under our distributed protocol, each in distributed and no-privacy global variants. The proxies re-implement retrieval cores *without* their original LLM components, so the comparison isolates the retrieval mechanism rather than the choice of language model. *High Communication (full candidate list)* is the communication ceiling of the candidate-based protocol family: each agent broadcasts its *entire local candidate list* (the full output of its local retriever) to every other agent, and the union is fused. We call it *high* rather than *full*: the protocol communicates retrieval *candidates*, not raw partitions, so its byte cost scales with local pool size, not with the raw corpus. The *Global GNN without collaboration* serves as a *no-privacy upper reference* (it sees the entire graph at zero communication cost and is not a fair competitor).

Table 1: GTSQA full-strict test main results (n=1622, 3-seed mean; hit@1 $\pm$ 95% CI half-width). Upper*: no-privacy reference, not a competitor. Partition BM25 and All-Peer BM25 fuse identical local lists, so rankings coincide; only communication accounting differs. Omitted rows in Appendix E.

Method	Mode	hit@1	hit@10	MRR	Bytes/q
Global GNN, no collab. (Upper*)	Global*	0.753±0.020	1.000	0.863	0
Random Candidate	Local	0.257±0.013	0.999	0.519	0
Partition BM25	Local	0.267±0.021	0.999	0.523	0
Global BM25	Global	0.290±0.022	1.000	0.537	0
Random-Peer BM25	Dist.	0.246±0.017	0.669	0.415	37
All-Peer BM25	Dist.	0.267±0.021	0.999	0.523	74
Graphiti-like (Dist.)	Dist.	0.266±0.022	1.000	0.524	74
GSW-like (Dist.)	Dist.	0.255±0.021	1.000	0.520	74
ARTEM-like (Dist.)	Dist.	0.255±0.021	1.000	0.520	74
High Communication (full candidate list)	Dist.	0.367±0.023	0.999	0.591	3,633
CoGMem (single-round)	Dist.	0.367±0.023	0.999	0.591	3,630
CoGMem-n (ours)	Dist.	**0.373**±0.023	0.997	**0.597**	**3,610**

Table 2: Episodic Memory full test main results (n=407, 3-seed mean); omitted rows as in Table 1.

Method	Mode	hit@1	hit@10	MRR	Bytes/q
Global GNN, no collab. (Upper*)	Global*	0.486±0.041	0.866	0.613	0
Random Candidate	Local	0.032±0.011	0.284	0.115	0
Partition BM25	Local	0.111±0.031	0.388	0.180	0
Global BM25	Global	0.093±0.028	0.629	0.248	0
Random-Peer BM25	Dist.	0.078±0.024	0.277	0.142	290
All-Peer BM25	Dist.	0.111±0.029	0.388	0.180	587
Graphiti-like (Dist.)	Dist.	0.103±0.028	0.317	0.164	732
GSW-like (Dist.)	Dist.	0.069±0.025	0.209	0.107	737
ARTEM-like (Dist.)	Dist.	0.057±0.021	0.197	0.095	737
High Communication (full candidate list)	Dist.	0.419±0.040	**0.846**	0.559	13,740
CoGMem (single-round)	Dist.	0.419±0.039	**0.846**	0.559	13,737
CoGMem-n (ours)	Dist.	**0.419**±0.039	0.649	0.511	**5,267**

Metrics. We report hit@1, hit@10, MRR, and communication bytes per query. Bytes per query are accounted as $\sum_{t=1}^{T}\sum_{i=1}^{N}(256 + 114\,K_{t,i})$: one 128-d fp16 summary (256 bytes) plus 114 bytes per candidate edge identifier per agent per round, averaged over queries and seeds. All numbers are 3-seed means with 95% bootstrap CIs; significance uses a per-question paired bootstrap (B=5000, fixed seed; run records and unit tests in Appendix C).

5.2 Main Results

Single-round CoGMem. Even without negotiation, CoGMem matches High Communication's hit@1 on both benchmarks (0.419 on Episodic, 0.367 on GTSQA) at essentially the same bytes (within 0.1%): the learned fusion stack, not broadcast width, already sets the accuracy ceiling; CoGMem-n then attacks its byte cost.

Gap to the no-privacy upper reference. On Episodic, CoGMem-n's hit@1 trails the global GNN's 0.486 by 6.7 pp (p<0.001; a non-significant -0.6 pp on the `full_llm` graph). On GTSQA the gap is larger (-38.0 pp): its 5-node micro-partitions give a global encoder an outsized structural advantage (High Communication itself loses -38.6 pp); the privacy tax is a property of partition structure, not of our protocol.

Communication efficiency. On Episodic, CoGMem-n achieves *exactly* High Communication's hit@1 (0.4185) at 5,267 vs. 13,740 bytes/query (-61.7%). The saving costs hit@10 (0.846→0.649, a truncation cost analyzed in §5.4). On GTSQA the pools are tiny (3,633 B), so negotiation saves no bytes while slightly improving hit@1: we claim efficiency for long-list narratives, not uniformly.

Advantage over distributed baselines. On Episodic, CoGMem-n exceeds *every* lexical and system-like distributed baseline by +30.8 to +36.2 pp hit@1 (all p<0.001; +10.7 pp over the best baseline on GTSQA). Random-Peer BM25 confirms that communication without fusion actively hurts (hit@10 0.277, below Partition BM25's 0.388): *what* is exchanged matters more than *whether* agents exchange.

5.3 Ablation Study

The GNN stack is the decisive component (Appendix F). Removing all learned components (NC1, dense-only scoring) costs -35.0 pp hit@1 and -46.5 pp hit@10 on Episodic ($p<0.001$), the largest effect, and -10.2 pp on GTSQA's separable micro-shards. Removing message passing and seed injection (NC2) costs a further -27.2 pp on Episodic ($p<0.001$) but only -1.7 pp on GTSQA (n.s.): propagation's benefit grows with graph size and partition structure.

Follow-ups extend coverage, time bonus is inert. Removing targeted BFS follow-ups (NC3) leaves hit@1 unchanged ($\Delta=0$, $p=1.0$) but reduces Episodic hit@10 by -7.9 pp ($p<0.001$): follow-ups buy top-10 coverage, not top-1 precision, at 14.8% fewer bytes. The time bonus (NC4) is a dead knob: removing it changes nothing on any metric.

Sensitivity and scalability. Per-query bytes grow linearly in N ($+839$ B/agent on GTSQA; $+1276$ B on Episodic), and a controlled $N=3$ comparison isolates *partition quality* as a first-class factor ($+18.4$ pp hit@1 for real over hash partitions); sweeps in Appendix H validate our operating points.

5.4 Multi-Round Negotiation Analysis

CoGMem vs. CoGMem-n. Single-round CoGMem is already accuracy-identical to High Communication (§5.2, Appendix G): the ceiling is set by ranking quality, not broadcast width. CoGMem-n's contribution is *economic*: identical hit@1 at 38% of the bytes, by cutting Round-1 gossip to $K_1=2$ and re-spending on gap-triggered follow-ups. The hit@10 gap (0.649 vs. 0.846) prices the 62% budget cut; a third round buys back only $+1.6$ pp hit@10 for $+10\%$ bytes, so $T=2$ is the default.

Follow-up policy: value and counter-evidence. Need-directed follow-ups beat uniform-random ones on Episodic (-6.3 pp hit@10, $p<0.001$; -7.1 pp on the LLM graph), so targeted deep-dives are *not* interchangeable with arbitrary queries. Yet the need-free *generic* top-gap policy is better still on hit@10 ($+3.0$ pp, $p=0.0004$; replicated on the LLM graph), while all three tie on hit@1: the regex need vector pushes budget toward narrow entity edges, sacrificing the broad coverage that rescues rank-5–10 items (a need$\cup$generic hybrid pool is the immediate fix).

Where negotiation helps. On pre-registered difficulty pools (Appendix G), negotiation's value concentrates on hard queries: CoGMem-n rescues GTSQA hard-pool hit@1 from 0 to 0.266 ($p<0.001$; 20.9% of questions), while on Episodic's severe pool it is ineffective and lexical retrieval still works. Forcing both rounds leaves every metric unchanged: cost tracks genuine incompleteness, not fleet size.

6 Conclusion

We introduced **CoGMem**, a collaborative GNN framework that treats inter-agent communication as a budgeted retrieval resource over agent-private graph memories, with a contrastively trained encoder driving its complementarity gate, SparseGate gossip, self-vote-protected reranking, and multi-round negotiation (CoGMem-n). On two benchmarks under strict pure retrieval, CoGMem-n matches High Communication's hit@1 at 38% of its bytes on Episodic Memory and exceeds every lexical and system-like distributed baseline by ≥ 30 pp hit@1; ablations confirm the GNN stack is load-bearing. Equally important, we delineate the boundary: negotiation does not uniformly improve coverage, saves no bytes on small pools, and fails where GNN semantics fail, with hybrid retrieval and semantics-aware partitioning as next steps toward memory systems.

Reproducibility Statement

All experiments use real input features (frozen 768-d `nomic-embed-text` embeddings; no ground-truth-derived features), a strict pure-retrieval protocol with no ground-truth edge injection, a contrastively trained encoder, three fixed seeds, and per-question paired bootstrap significance; all protocol constants are locked before full-test evaluation. Appendix C details the protocol and test guards, Appendix D lists the locked constants, and source code, configurations, and the analysis notebooks producing every table and figure are provided in the supplementary material.

Ethics Statement

This work studies privacy-preserving retrieval: raw partition data never leaves its owner by design. Experiments use two public benchmarks (synthetic episodic narratives and knowledge-graph QA) containing no personal data. We note that the privacy model is protocol-level and does not by itself constitute a formal differential-privacy guarantee; we foresee no direct negative societal impact.

AI Use Statement

The authors used AI assistants for language polishing and LaTeX typesetting support during the preparation of this manuscript. All research ideas, system design, experiments, analyses, and claims are the authors' own, and the authors take full responsibility for the entire content of the paper.

References

Alberto Cattaneo, Carlo Luschi, and Daniel Justus. Ground-truth subgraphs for better training and evaluation of knowledge graph augmented LLMs. *arXiv preprint arXiv:2511.04473*, 2025. GT-SQA dataset; Graphcore Research.

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.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, and Andrew Y Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1223–1231, 2012.

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

Hang Gao and Yongfeng Zhang. INMS: Memory sharing for large language model based agents. *arXiv preprint arXiv:2404.09982*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1263–1272, 2017.

Bernat 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 (NeurIPS)*, 2024.

Bernat Jiménez Gutiérrez, Yiheng Shu, Weizhi Qi, Shuxin Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexis Huet, Zied Ben Houidi, and Dario Rossi. Episodic memories generation and evaluation benchmark for large language models. In *International Conference on Learning Representations (ICLR)*, 2025. EpBench; arXiv:2501.13121.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 36848–36865, 2026. arXiv:2601.03236.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Sam Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

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

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations (ICLR)*, 2024.

Costas Mavromatis and George Karypis. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics (ACL)*, 2025.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

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

Joon Sung Park, Joseph O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pp. 1–22, 2023.

Shreyas Rajesh, Pavan Holur, Chenda Duan, David Chong, and Vwani Roychowdhury. Beyond fact retrieval: Episodic memory for RAG with generative semantic workspaces. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 32782–32790, 2026.

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*, 2025.

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

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*, pp. 593–607, 2018.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations (ICLR)*, 2024.

Cassandra Hui-Ming Tan, Budhitama Subagdja, and Ah-Hwee Tan. ARTEM: Enhancing large language model agents with spatial-temporal episodic memory. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 25753–25760, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. KVCOMM: Online cross-context KV-cache communication for efficient LLM-based multi-agent systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025.

A Omitted Proofs

Proof of Theorem 4.1 (communication bound). Each agent gossips exactly one d-dim summary and at most K_t candidate identifiers per round, i.e. $2d + K_t e_{\max}$ bytes (fp16); summing over N agents and T rounds gives the bound. The accounting covers the two dominant message types (summaries and candidate identifiers); $O(1)$-dimensional control messages (need and gap vectors) add only a constant $O(TN)$ term and leave the asymptotic claim unchanged. Early-stop only reduces the realized rounds below T. High communication, in contrast, ships $\Omega(|\mathcal{E}_i|)$ identifiers per agent, which grows with the corpus. Empirically the realized slope is $+839$ B/agent (GTSQA, R^2=1.000) and $+1276$ B/agent (Episodic, R^2=0.993), confirming the linear, corpus-independent regime (§5.3).

Proof of Theorem 4.2 (negotiation as budgeted submodular maximization). Each summand $w_d \min(1, \cdot)$ is a monotone concave function of the total coverage on dimension d, hence monotone submodular; non-negative weighted sums preserve both properties. Maximizing a monotone submodular function under a knapsack constraint by cost-effectiveness greedy, augmented with the best single element, yields the classical $(1 - 1/e)$ guarantee (Nemhauser et al., 1978; Khuller et al., 1999); this bound covers the idealized rule, including the best-single-candidate fallback. The marginal-diminishment penalty 0.5 is an implementation-level approximation of the submodular marginal gain $\Delta(e \mid S)$ after a dimension is partially covered, keeping the greedy rule aligned with true marginal utility; the gain-proxy early-stop terminates once $\max_e \Delta(e \mid S) < 0.03$, i.e., when the guaranteed remaining headroom is negligible.

Proof of Theorem 4.3 (self-vote top-1 protection and budget-saturation invariance). By Eq. 8, the answer a^* attains $\mathrm{FinalScore}(a^*) \geq \lambda_{\text{self}} \cdot 1$, while any competitor d without home-agent support scores at most $(1 - \lambda_{\text{self}})\, x_{\max}$. The threshold condition makes the first quantity weakly larger (ties at the threshold are broken in favor of the home candidate), so no competitor can overtake a^*; truncating lists cannot remove a^*'s self score and, by the coverage condition, does not remove a^* itself. The bound $\lambda_{\text{self}} \geq x_{\max}/(1 + x_{\max})$ is tight when a competitor accumulates maximal cross support and zero self support. This formalizes three observations: pure cross-agent trust (λ_{self}=0) collapses hit@1 to 0.003; λ_{self}≥0.45 (the largest cross-agent signal a competitor accumulates in practice is ≈0.82 after normalization, and $0.45 = 0.82/(1 + 0.82)$ sits exactly at the resulting threshold) fully restores it; and hit@1 is invariant once the budget passes the 7,277-byte saturation point, while hit@10 is *not* protected by this argument, consistent with the coverage loss coming from budget truncation of ranks 2–10 (§5.2, §5.3).

Algorithm 1 CoGMem: single-round budgeted private graph retrieval

Require: query q; agents $\{a_i\}_{i=1}^{N}$ with private partitions $\{\mathcal{G}_i\}$; budget B; cap K_1
Ensure: global top-K list L

// Phase 1: local encoding (per agent a_i, in parallel)
$\mathbf{h} \leftarrow \mathrm{HetGNN}_q(\mathcal{G}_i)$ (Eq. 3); $(c_{i,e}, r_{i,e}) \leftarrow \mathrm{heads}(\mathbf{h})$
$h_i(q) \leftarrow$ top-k_{local} edges by $r_{i,e}$; $\mathbf{s}_i \leftarrow \mathrm{pool}(\{\mathbf{h}_v\}) \in \mathbb{R}^{128}$
// Phase 2: complementarity-gated SparseGate gossip
for all ordered pairs $(j \to i)$, $j \neq i$ **do**
    $\mathrm{score}(e) \leftarrow$ gate of Eq. 4 from i's view, $e \in h_j(q)$
    $\mathrm{payload}_j \leftarrow$ top-K_1 by score s.t. Bytes $\leq B$ (Eq. 5)
    send $\mathbf{s}_j$ (256 B) + IDs of $\mathrm{payload}_j$ (114 B each)
end for
// Phase 3: fusion with self-vote protection
$\mathrm{S} \leftarrow [\mathbf{s}_1; \ldots; \mathbf{s}_N]$; $a_{\mathrm{S}}(i,d)$ (Eq. 6); $\mathrm{RRF}(d)$ (Eq. 7)
$\mathrm{FinalScore}(d) \leftarrow \lambda_{\mathrm{self}}\,\mathrm{self}(d) + (1-\lambda_{\mathrm{self}})\,\mathrm{cross}(d)$ (Eq. 8)
return $L \leftarrow$ top-K candidates by FinalScore

B Algorithms

Algorithms 1 and 2 give the pseudocode of the two protocols described in §4; all constants are the locked values of Appendix D.

Algorithm 1 (CoGMem, single round). In **Phase 1**, each agent embeds its node/edge label text with the frozen encoder, runs the query-conditioned heterogeneous GNN of Eq. 3 with seed injection of q, and emits per-edge confidence $c_{i,e}$ and relevance $r_{i,e}$, a local top-k_{local} list $h_i(q)$ ranked by $r_{i,e}$, and a pooled 128-d partition summary $\mathbf{s}_i$; raw data never leaves the partition. In **Phase 2**, every ordered pair $(j \to i)$ re-scores j's local candidates from receiver i's perspective with the complementarity gate of Eq. 4, computed against i's `SAME_AS`-registered *effective* inventory; SparseGate then keeps the top-K_1 gated candidates that fit the remaining byte budget (Eq. 5). Only the 256-B summary and the 114-B-per-edge candidate identifiers cross the boundary. In **Phase 3**, the summaries are stacked into $\mathrm{S} \in \mathbb{R}^{N \times 128}$; the cross-agent signal $\mathrm{cross}(d)$ combines summary-attention reliability $a_{\mathrm{S}}(i,d)$ (Eq. 6) with RRF rank consistency (k=60, Eq. 7), and the final score of Eq. 8 adds the home agent's self score under the protection weight λ_{self}=0.45 (Theorem 4.3).

Algorithm 2 (CoGMem-n, negotiation loop). CoGMem-n wraps Algorithm 1 in a T-round loop (T=2 by default). The need extractor maps q to $\mathbf{n}(q) \in [0,1]^5$ over {temporal, who, where, coreference, context}; the regex extractor is the default and the LLM variant is ablated in Appendix J. Round 1 is the initial gossip with cap K_1=2. After each round, the gap assessor computes per-dimension residual gaps $g_d = 1 - \mathrm{coverage}_d$ on the fused candidates and launches a follow-up iff $\max_d g_d > \delta$ (δ=0.35). On a triggered round, each agent runs bounded BFS from its current best candidates along above-gap dimensions only, then selects edges by the Δ/cost greedy rule of Theorem 4.2: the marginal utility on the need dimensions is divided by the edge's byte cost, and re-selecting an already-covered dimension is penalized by the factor ρ=0.5. Selection halts when the best remaining gain proxy falls below ε=0.03 or the budget is exhausted; sent identifiers are deducted from B, and Phase-3 fusion re-runs over the union of all received candidates.

C Reproducibility Details

All experiments use *real* input features: frozen 768-d `nomic-embed-text` embeddings of node/edge raw label text; no feature is derived from ground-truth answers, and no ground-truth edge is injected into any candidate pool (pure-retrieval protocol, ADR-01), enforced by 165 unit/red-line tests. The GNN encoder (node projection, node/type/time heads, per-edge-type message passing, seed injection) is *trained* with the InfoNCE objective of §4.3. All reported numbers are means over 3 seeds with 95% bootstrap CIs; significance uses a per-question paired bootstrap (B=5000, fixed seed) on per-seed-averaged scores. Communication bytes are accounted per agent per round as one 128-d fp16 summary (256 bytes) plus 114 bytes per candidate edge identifier (§5.1). The full run

Algorithm 2 CoGMem-n: multi-round negotiation protocol

Require: query q; budget B; T=2; K_1=2; δ=0.35; ε=0.03; ρ=0.5
Ensure: global top-K list L

$\mathbf{n}(q) \leftarrow$ NEEDEXTRACT$(q) \in [0,1]^5$ ▷ regex; LLM variant: App. J
$L \leftarrow$ Algorithm 1(q, K_1) ▷ Round 1: initial gossip
for $t = 2$ **to** T **do**
  $g_d \leftarrow 1 - \text{coverage}_d(L),\ \forall d$ ▷ gap assessor
  if $\max_d g_d \le \delta$ **then break**
  end if
  for all agents a_i (in parallel) **do**
    $C_i \leftarrow$ BFS from best candidates along $\{d : g_d > \delta\};\ S \leftarrow \emptyset$
    while $\max_{e \in C_i} \hat{\Delta}(e \mid S) \ge \varepsilon$ **and** $B > 0$ **do**
      $e^* \leftarrow \arg\max_{e \in C_i} \hat{\Delta}(e \mid S)/\text{Bytes}(e)$ ▷ ρ-penalized if covered
      $S \leftarrow S \cup \{e^*\};\ B \leftarrow B - \text{Bytes}(e^*)$
    end while
    send IDs of S
  end for
  $L \leftarrow$ Phase-3 fusion over $\bigcup_i$ received candidates (Alg. 1)
end for
return L

comprises 146,088 negotiation records with 0 errors. Source code, configurations, and the analysis notebooks producing every table and figure are provided in the supplementary material.

D LOCKED CONSTANTS

The following constants are locked before any full-test evaluation and used uniformly across datasets unless swept in §5.3: self-vote weight λ_{self}=0.45; RRF constant k=60; gap threshold GAP_DELTA=0.35; early-stop threshold GAIN_EPSILON=0.03; Round-1 candidates per agent K_1=2; default rounds T=2; marginal-diminishment factor 0.5; byte-budget anchor $1 \times$ =7,277 B (Episodic saturation point); local pool size k_{local}=10; global output list size 20; candidate-identifier cost 114 bytes per edge; per-agent-per-round summary 256 bytes (128-d, fp16). Default partitioning is the *real* benchmark partition at N=3 (event boundaries for Episodic Memory; dataset segments for GTSQA).

E FULL MAIN-RESULT TABLES

Tables 3 and 4 give the complete 15-row main-result tables, adding the no-retrieval sanity row, the three global system-like variants (Graphiti-like / GSW-like / ARTEM-like Global), and the T=3 variant to the compact Tables 1 and 2 in §5.2.

F COMPONENT ABLATION TABLE

Table 5 gives the full component-ablation comparison behind §5.3.

G DETAILED NEGOTIATION PROTOCOL RESULTS

Table 6 gives the full negotiation-protocol comparison behind §5.4, including the forced-two-round (w/o early-stop) rows, the random/generic follow-up controls (NC5/NC6), and the full-LLM need-vector block (Appendix J).

Hard-pool stratification (details of §5.4). Using two pre-registered pools (severe: 3-seed-mean hit@10=0 for the global GNN; hard: mean hit@1=0), negotiation's value concentrates exactly where intended. On GTSQA the severe pool is empty, and on the hard pool (339 questions, 20.9%)

Table 3: Full GTSQA main-result table (all 15 baseline/reference rows), extending Table 1. Partition BM25 and All-Peer BM25 share identical rankings (only the communication accounting differs); *-like* rows are mechanism re-implementation proxies, not the original systems.

Method	Mode	hit@1	hit@10	MRR	Bytes/q
Global GNN, no collab. (Upper*)	Global*	0.753±0.020	1.000	0.863	0
No Retrieval (sanity)	–	0.000±0.000	0.000	0.000	0
Random Candidate	Local	0.257±0.013	0.999	0.519	0
Partition BM25	Local	0.267±0.021	0.999	0.523	0
Global BM25	Global	0.290±0.022	1.000	0.537	0
Graphiti-like (Global)	Global	0.291±0.023	1.000	0.540	0
GSW-like (Global)	Global	0.243±0.021	1.000	0.523	0
ARTEM-like (Global)	Global	0.243±0.021	1.000	0.523	0
Random-Peer BM25	Dist.	0.246±0.017	0.669	0.415	37
All-Peer BM25	Dist.	0.267±0.021	0.999	0.523	74
Graphiti-like (Dist.)	Dist.	0.266±0.022	1.000	0.524	74
GSW-like (Dist.)	Dist.	0.255±0.021	1.000	0.520	74
ARTEM-like (Dist.)	Dist.	0.255±0.021	1.000	0.520	74
High Communication (full candidate list)	Dist.	0.367±0.023	0.999	0.591	3,633
CoGMem (single-round)	Dist.	0.367±0.023	0.999	0.591	3,630
CoGMem-n (ours)	Dist.	**0.373**±0.023	0.997	**0.597**	**3,610**
CoGMem-n (T=3)	Dist.	0.373±0.023	0.997	0.597	3,611

Table 4: Full Episodic Memory main-result table (all 15 baseline/reference rows), extending Table 2; notes as in Table 3.

Method	Mode	hit@1	hit@10	MRR	Bytes/q
Global GNN, no collab. (Upper*)	Global*	0.486±0.041	0.866	0.613	0
No Retrieval (sanity)	–	0.000±0.000	0.000	0.000	0
Random Candidate	Local	0.032±0.011	0.284	0.115	0
Partition BM25	Local	0.111±0.031	0.388	0.180	0
Global BM25	Global	0.093±0.028	0.629	0.248	0
Graphiti-like (Global)	Global	0.093±0.028	0.536	0.233	0
GSW-like (Global)	Global	0.039±0.018	0.256	0.118	0
ARTEM-like (Global)	Global	0.027±0.016	0.243	0.106	0
Random-Peer BM25	Dist.	0.078±0.024	0.277	0.142	290
All-Peer BM25	Dist.	0.111±0.029	0.388	0.180	587
Graphiti-like (Dist.)	Dist.	0.103±0.028	0.317	0.164	732
GSW-like (Dist.)	Dist.	0.069±0.025	0.209	0.107	737
ARTEM-like (Dist.)	Dist.	0.057±0.021	0.197	0.095	737
High Communication (full candidate list)	Dist.	0.419±0.040	**0.846**	0.559	13,740
CoGMem (single-round)	Dist.	0.419±0.039	**0.846**	0.559	13,737
CoGMem-n (ours)	Dist.	**0.419**±0.039	0.649	0.511	**5,267**
CoGMem-n (T=3)	Dist.	0.419±0.039	0.665	0.514	5,808

CoGMem-n rescues hit@1 from 0 to 0.266 ($p<0.001$, MRR $+0.055$): negotiation's role on GTSQA is *top-rank reranking on hard queries*. On Episodic's hard pool (145 questions) CoGMem-n reaches hit@1 0.041 ($p<0.001$) with a -0.340 hit@10 cost against the global GNN, whereas High Communication loses only -0.007, confirming that the deficit is budget truncation, not fusion. On the severe pool (34 questions, 8.4%) negotiation is ineffective ($h@10=0.010$, n.s.), but Global BM25 reaches $h@1=0.177$/$h@10=0.324$: where GNN semantics fail entirely, the lexical channel is complementary, direct evidence for future hybrid retrieval.

Protocol statistics (details). Early-stop fires on 69.1% (GTSQA) / 33.7% (Episodic) of queries. The second round triggers on 79.3% (GTSQA) / 66.3% (Episodic) of queries, the gain-proxy early-stop fires on 69.1% / 33.7%, follow-ups add only 0.43 / 3.98 edges per query, and the average round count is 1.79 / 1.66; negotiation is frequent but cheap, and its cost tracks genuine incompleteness rather than fleet size. Forcing both rounds (w/o early-stop, Table 6) leaves every metric unchanged and only increases message counts, confirming that the early-stop removes pure overhead.

H Sensitivity Figures

Figures 3 and 4 support the fleet-size and hyper-parameter analysis of §5.3.

Table 5: Component ablation vs. CoGMem-n, paired bootstrap (B=5000). We report Δhit@1 / Δhit@10 with two-sided p, and bytes per query (GTSQA / Episodic). The GNN stack (NC1) and message passing (NC2) are load-bearing on Episodic; follow-ups (NC3) extend top-10 coverage without changing hit@1; the multiplicative time bonus (NC4) is a dead knob.

	GTSQA		Episodic		Bytes/q
Ablated component	Δh@1 (p)	Δh@10 (p)	Δh@1 (p)	Δh@10 (p)	(G / E)
NC1 dense-only (no GNN learning)	-0.102 (<0.001)	-0.020 (<0.001)	-0.350 (<0.001)	-0.465 (<0.001)	3,608 / 5,245
NC2 base-only (no message passing)	-0.017 (0.177)	-0.000 (0.863)	-0.272 (<0.001)	-0.333 (<0.001)	3,616 / 5,272
NC3 no BFS follow-up	$+0.005$ (0.541)	-0.008 (<0.001)	0.000 (1.0)	-0.079 (<0.001)	3,475 / 4,486
NC4 no time bonus	-0.000 (0.955)	0.000 (1.0)	0.000 (1.0)	0.000 (1.0)	3,610 / 5,265

Table 6: Negotiation protocol comparison (3-seed mean). NC5/NC6 replace the follow-up policy under the same budget; w/o early-stop forces T=2 rounds. On Episodic, CoGMem-n matches High Communication's hit@1 at 38% of bytes; the generic follow-up (NC6) further improves hit@10 over the need-directed default (see text for significance). Msgs/q is the average number of inter-agent messages per query (approximately rounds $\times$ N).

Method	hit@1	hit@10	MRR	Bytes/q	Msgs/q
GTSQA full-strict test (n=1622)					
High Communication (full candidate list)	0.367	0.999	0.591	3,633	4.00
CoGMem (single-round)	0.367	0.999	0.591	3,630	4.00
CoGMem-n (ours)	**0.373**	0.997	**0.597**	**3,610**	5.10
CoGMem-n (T=3)	0.373	0.997	0.597	3,611	5.41
w/o early-stop	0.373	0.997	0.597	3,610	5.31
random follow-up (NC5)	0.373	0.997	0.597	3,610	5.10
generic follow-up (NC6)	0.373	0.997	0.596	3,622	5.13
Episodic full test (n=407, regex need vector)					
High Communication (full candidate list)	0.419	0.846	0.559	13,740	4.00
CoGMem (single-round)	0.419	0.846	0.559	13,737	4.00
CoGMem-n (ours)	**0.419**	0.649	0.511	**5,267**	5.33
CoGMem-n (T=3)	0.419	0.665	0.514	5,808	6.23
w/o early-stop	0.419	0.649	0.511	5,267	5.66
random follow-up (NC5)	0.419	0.586	0.497	5,270	5.33
generic follow-up (NC6)	0.419	**0.680**	**0.517**	5,261	5.33
Episodic full-LLM test (n=407, LLM need vector, 994 real calls, 0 fallback)					
High Communication (full candidate list)	0.337	0.796	0.484	13,737	4.00
CoGMem (single-round)	0.337	0.796	0.484	13,734	4.00
CoGMem-n (ours)	**0.337**	0.589	0.435	**5,269**	5.33
CoGMem-n (T=3)	0.337	0.607	0.438	5,783	6.18
w/o early-stop	0.337	0.589	0.435	5,269	5.66
random follow-up (NC5)	0.337	0.518	0.420	5,273	5.33
generic follow-up (NC6)	0.337	**0.613**	**0.439**	5,273	5.33

Fleet size, partition quality, and operational thresholds (Figures 3 and 4, Appendix H). Per-query bytes grow strictly linearly in N ($+839$ B/agent on GTSQA, R^2=1.000; $+1276$ B/agent on Episodic, R^2=0.993), confirming bounded communication scaling, while hit@1 degrades gracefully (GTSQA 0.380$\rightarrow$0.282 for N=3$\rightarrow$8). A controlled comparison at N=3 isolates *partition quality* as a first-class factor: real event-boundary partitions reach 0.419 hit@1 vs. 0.234 for balanced hashes ($+18.4$ pp), while GTSQA micro-shards are insensitive (0.373 vs. 0.380), so semantics-aware partitioning is a promising systems direction. The λ_{self} sweep confirms the trust-degradation hypothesis quantitatively: pure cross-agent trust (λ=0/0.25) collapses Episodic hit@1 to 0.003 because remote votes override correct local top-1 candidates, while $\lambda_{\text{self}} \geq 0.45$ fully restores 0.419 (GTSQA is robust, 0.352–0.373). The budget sweep validates the locked anchor: 7,277 B saturates Episodic ($1\times$ =$2\times$ at 0.649 hit@10; $0.5\times$ costs -7.9 pp), GTSQA loses only 0.3 pp at half budget, the gap threshold never changes hit@1 and trades coverage monotonically, and k_{local} saturates at 10.

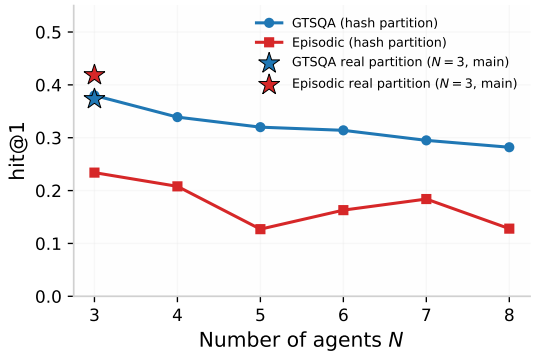


Figure 3: **Fleet-size sensitivity.** hit@1 vs. number of agents N under independent balanced hash partitions (trend analysis only). Star markers: the main results on real event-boundary partitions at N=3 (Episodic 0.419 vs. hash 0.234; GTSQA 0.373 vs. hash 0.380): partition *semantic quality* matters more than balance on narrative data. Communication bytes grow strictly linearly: +839 B/agent on GTSQA (R^2=1.000) and +1276 B/agent on Episodic (R^2=0.993).

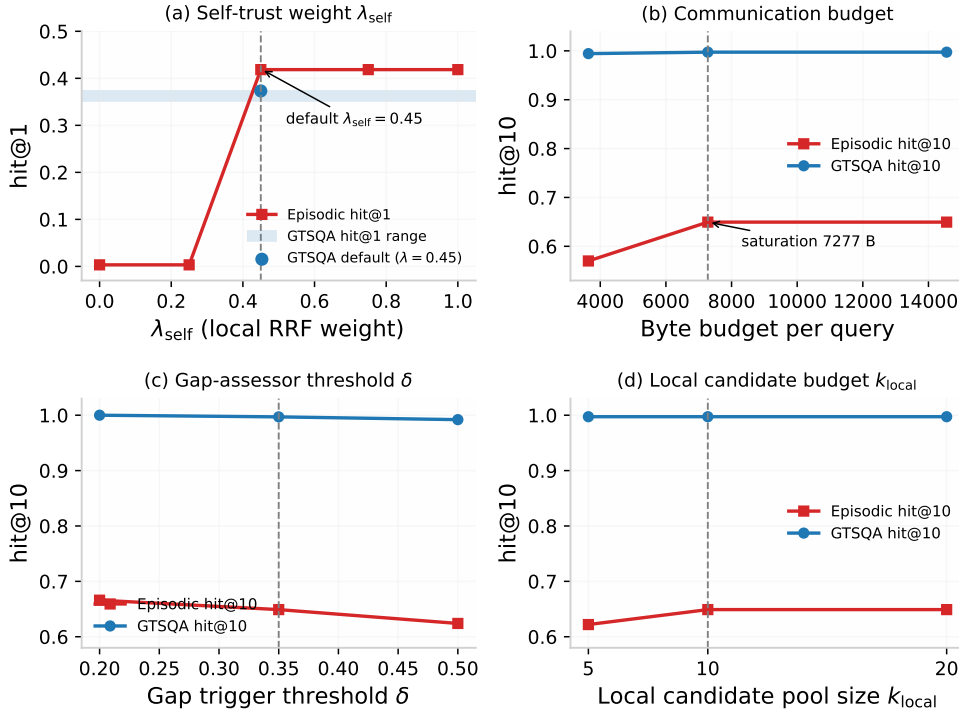


Figure 4: **Protocol hyper-parameter sensitivity** (full test, 3 seeds). **(a)** Self-trust weight λ_{self}: pure cross-agent trust (λ=0/0.25) collapses Episodic hit@1 to 0.003; $\lambda \geq 0.45$ fully recovers it, so self-vote protection is necessary. **(b)** Byte budget: Episodic hit@10 saturates at 7,277 B ($1\times = 2\times$), half budget costs −7.9 pp. **(c)** Gap threshold δ: hit@1 is invariant; smaller δ widens follow-up coverage. **(d)** Local pool size k_{local}: 5→10 gives +2.8 pp on Episodic, 20 adds nothing; GTSQA is flat throughout (local pools are naturally small).

I Limitations and Future Directions

Our results support four claims. **(i)** On long-list narrative memory (Episodic), CoGMem-n matches High Communication's top-1 accuracy at 38% of its bytes and exceeds every lexical/system-like distributed baseline by ≥30 pp hit@1. **(ii)** The GNN stack and cross-partition message passing are causally load-bearing (NC1/NC2, up to −35.0 pp hit@1, p<0.001). **(iii)** The protocol has iden-

tifiable operating points: $\lambda_{\text{self}} \geq 0.45$ self-vote protection and a 7,277-byte saturation budget. **(iv)** On GTSQA, negotiation's value is top-rank reranking on the 20.9% hard queries ($0 \rightarrow 0.266$ hit@1). Equally important is what we do *not* claim. Negotiation does not universally improve retrieval: overall hit@1 remains below the no-privacy global GNN (-6.7 pp Episodic, -38.0 pp GTSQA), and on GTSQA it saves no bytes. Follow-ups are not universally beneficial (zero hit@1 gain; generic $>$ need on hit@10), and the multiplicative time bonus is inert. Absolute numbers shift with the choice of encoder and extraction pipeline, but the relative protocol comparison is encoder-agnostic by construction (all methods share the same frozen features). Four evidence-driven directions follow: a need$\cup$generic hybrid follow-up pool, a redesigned temporal bonus, GNN+BM25 hybrid retrieval for the severe pool (lexical recall 0.324 where semantics fail), and semantics-aware partitioning as N grows.

J LLM NEED-VECTOR ABLATION

The LLM need-vector variant replaces the regex extractor with `deepseek-r1:7b` served locally via Ollama, with query-level caching and automatic regex fallback (994 real calls, 0 fallbacks). The gap assessor, targeted retriever, and greedy negotiator are held fixed. Results replicate the regex variant's pattern (generic follow-up $+2.5$ pp hit@10 over need-directed, $p{=}0.011$; full comparison in the third block of Table 6, §5.4), indicating that the negotiation protocol's effectiveness does not hinge on sophisticated need-vector extraction; we keep regex as the default for its zero runtime cost and zero external dependency.

K PARTITIONING AND N-SWEEP PROTOCOL

Main results use the benchmarks' real partitions. The fleet-size sweep ($N \in \{3, \ldots, 8\}$) instead uses *independent* balanced blake2b hash partitions per N (independent of the real partition files) and is used for trend analysis only; the controlled $N{=}3$ comparison between real and hash partitions isolates partition quality ($+18.4$ pp hit@1 for event-boundary partitions on Episodic Memory; near-identical on GTSQA micro-shards, §5.3).