KNOWLEDGECLAW: TOWARDS AGENTIC KNOWLEDGE ENGINEERING VIA OMNIMODAL HYPERGRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge engineering (KE) turns raw files into a reasoning-ready knowledge base, a task now within reach of autonomous LLM agents. KE has progressed from chunk and graph retrieval-augmented generation (RAG) to memory agents and agentic, multimodal KE systems. However, a live knowledge base needs omnimodality, a closed-loop lifecycle and multi-granular application at once, yet each existing system satisfies at most two and no benchmark measures all three. We formalize *Omnimodal Knowledge Engineering* (OmniKE) and release *OmniKE-Bench*, six tasks with 1,829 model-free items over the real files of ten personas in five modalities. We propose **KnowledgeClaw**, an agentic system that builds an omnimodal typed hypergraph, maintains it through a closed-loop lifecycle agent and routes each query to the granularity it needs. Against 30 baselines, KnowledgeClaw ranks first on all six tasks with an OKE of 55.9, +14.6 over the strongest baseline. Our software and data are publicly available.[1]

1 INTRODUCTION

Knowledge engineering (KE) (Studer et al., 1998) turns unstructured information into a reasoning-ready knowledge base, but manual expert construction long kept it from scaling (Pan et al., 2024). LLM agents (OpenAI et al., 2024; Yao et al., 2023; Wang et al., 2024) lift this limit: an agent can read files, extract entities, link records and revise what it stores (Xu et al., 2024), curating a person's growing files into a live knowledge base.

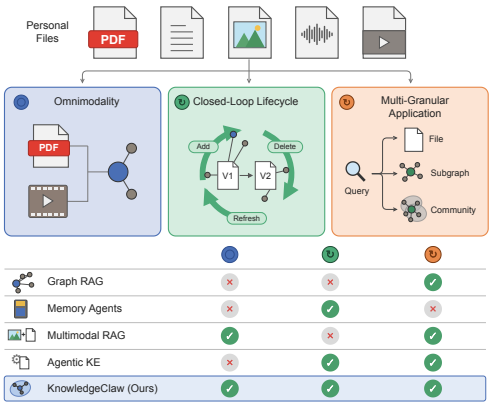


Figure 1: Existing KE systems each miss one property; KnowledgeClaw meets all three.

Along this path, knowledge structures have grown richer: RAG (Lewis et al., 2020) retrieves passages, GraphRAG (Edge et al., 2025) builds entity graphs and HyperGraphRAG (Luo et al., 2025) n-ary hyperedges; memory agents (Packer et al., 2024; Chhikara et al., 2025) update a persistent store and MemPro (Liu et al., 2026d) evolves it; agentic KE systems (OpenClaw contributors, 2026; Google Cloud, 2026) hand construction and maintenance to one agent, and Omni-SimpleMem (Liu et al., 2026a) reaches audio and video.

However, a personal knowledge base asks for all of this at once: papers arrive as PDFs, talks as audio or video and notes as text; papers are revised or withdrawn; and questions range from one-file lookups to multi-file joins and topic synthesis. It thus demands **(i) Omnimodality**, one node for an entity seen in a PDF and in a recording; **(ii) Closed-loop lifecycle**, every edit reaching everything derived from the changed file; and **(iii) Multi-granular application**, each query served at the file, subgraph or community level. As Figure 1 shows, existing lines satisfy at most two: graph RAG indexes static text, memory agents keep flat textual stores, multimodal RAG never maintains, and agentic KE reads text alone; nor does any benchmark measure all three.

[1] Anonymous GitHub link: `https://anonymous.4open.science/r/KnowledgeClaw`

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.

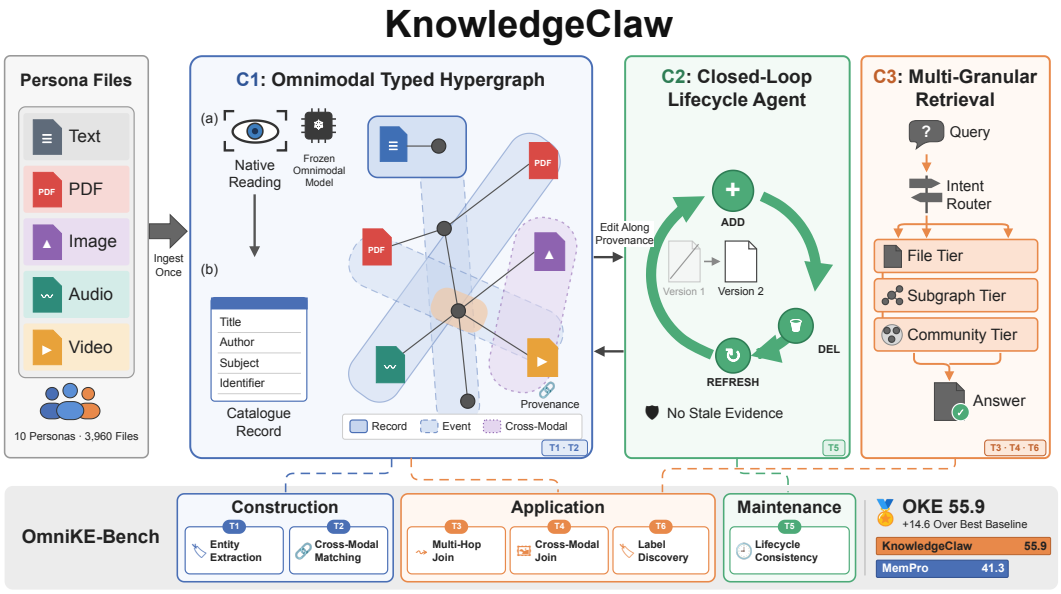


Figure 2: Overview of KnowledgeClaw on OmniKE. C1 builds an omnimodal hypergraph (T1, T2), C2 maintains it (T5) and C3 routes queries (T3, T4, T6); it leads all six tasks (OKE = 55.9).

To close this gap, we formalize *Omnimodal Knowledge Engineering* (**OmniKE**), where a system reads a person's files once into an artefact and answers every later query through it, and release **OmniKE-Bench**: six method-agnostic tasks on construction (T1, T2), application (T3, T4, T6) and maintenance (T5), with 1,829 items over 3,960 real files of ten personas in five modalities, and no answer made by a model. Figure 2 previews how our contributions cover these tasks.

We then propose **KnowledgeClaw** with one component per property. **(C1) Omnimodal typed hypergraph** reads every file natively into a catalogue record and merges all records into one provenance-tagged hypergraph, joining any two files that share an entity within two hyperedges. **(C2) Closed-loop lifecycle agent** commits ADD, DEL and REFRESH in place along provenance, so the store rests only on present files. **(C3) Multi-granular retrieval with intent routing** sends each query to the file, subgraph or community tier, so every file a question names enters its evidence.

Against 30 baselines in ten groups, KnowledgeClaw ranks first on all six tasks with an OKE of **55.9**, **+14.6** over the best baseline MemPro; removing C1, C2 or C3 lowers T1 by 20.0, T5 by 46.0 or OKE by 29.2; and with 256 evidence tokens it still beats every full-context baseline.

2 Related Work

Our setting joins two lines of work, which Appendix G surveys in full.

LLM-based knowledge engineering. KE structures have grown from passages and summary trees (Lewis et al., 2020; Sarthi et al., 2024) to entity graphs (Edge et al., 2025; Guo et al., 2025b; Gutiérrez et al., 2025; Liang et al., 2025; Dong et al., 2025), curated KGs (Lairgi et al., 2025; Li et al., 2026c) and n-ary hypergraphs (Luo et al., 2025; Huang et al., 2026; Lien et al., 2026), yet each indexes a static corpus. Memory agents (Packer et al., 2024; Xu et al., 2025; Chhikara et al., 2025; Rasmussen et al., 2025; Li et al., 2025; Lairgi et al., 2026; Zhang et al., 2025; Liu et al., 2026d) and agentic KE software (OpenClaw contributors, 2026; Graphify contributors, 2026; Markovic et al., 2025; Karpathy, 2026; Google Cloud, 2026) make the store writable, but revise single facts or pages rather than everything an edited file supports.

Multimodal retrieval-augmented generation. Multimodal RAG carries this progression beyond text, linking visual and textual graphs (Guo et al., 2025a; Wan & Yu, 2026), long videos (Ren et al., 2026; Park et al., 2026a) and multimodal memories (Wang & Chen, 2025; Liu et al., 2026a), while its editing methods (Yuan et al., 2026a; Lin et al., 2026; Feng et al., 2026) likewise bypass source files. KnowledgeClaw joins both lines in one omnimodal hypergraph kept consistent under file edits.

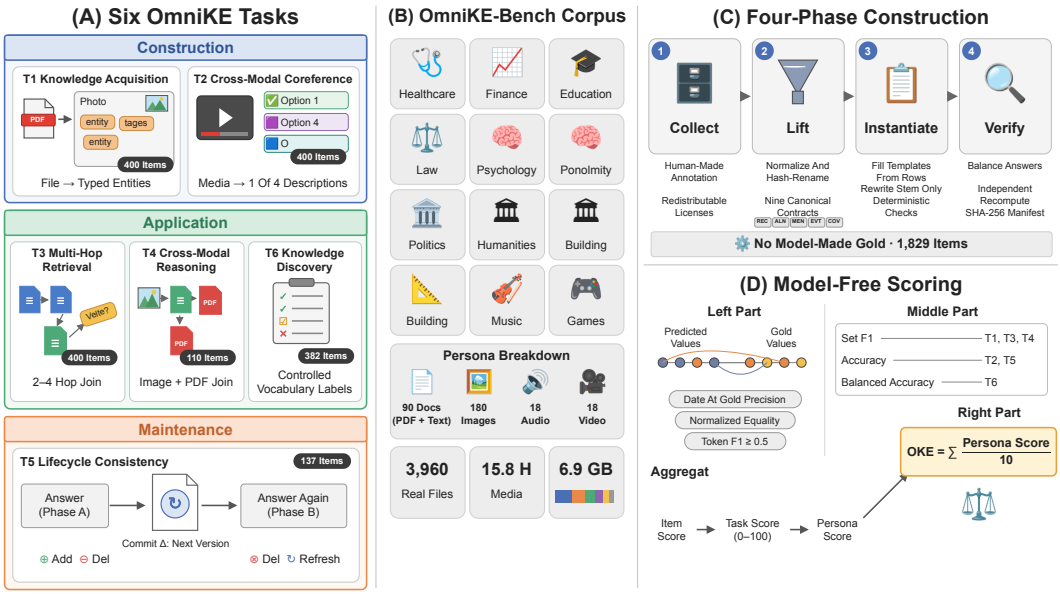


Figure 3: **(A)** Six OmniKE tasks in three groups: construction (T1, T2), application (T3, T4, T6) and maintenance (T5). **(B)** OmniKE-Bench spans ten personas, five modalities and 3,960 real files. **(C)** Four phases build its 1,829 items with no model-made gold. **(D)** Model-free scoring and OKE.

3 The OmniKE Tasks and Benchmark

Task definition. A *persona* is one researcher's file collection $\mathcal{D}$ over the modalities $\mathcal{M} = \{text, pdf, image, audio, video\}$. A system reads $\mathcal{D}$ once, at INGEST, and commits an artefact $\mathcal{K} = \mathcal{S}(\mathcal{D})$; every later query is answered through $\mathcal{K}$ alone, each in a fresh context. The six tasks (Figure 3A, Table 4) form three groups, one per property. *Construction* asks whether $\mathcal{K}$ captures what each file holds: extract a file's entities of given types ($\mathcal{T}_1$) and match a media file to its authoritative description ($\mathcal{T}_2$). *Application* asks whether it composes records: answer 2–4-hop joins ($\mathcal{T}_3$), joins across modalities ($\mathcal{T}_4$) and a file's controlled-vocabulary labels ($\mathcal{T}_6$). *Maintenance* ($\mathcal{T}_5$) asks whether it can be corrected: the system answers, commits an edit $\Delta \in \{\text{ADD}, \text{DEL}, \text{REFRESH}\}$ that changes the answer, here a document's next version, and answers again.

Benchmark and statistics. OmniKE-Bench holds ten personas from healthcare to games, each with 90 documents as PDF and text, 180 images, 18 audio and 18 video clips: 3,960 files, 15.8 h of media and 6.9 GB in all (Figure 3B, Table 5). Its 1,829 items comprise 400 each for $\mathcal{T}_1$–$\mathcal{T}_3$, 110 for $\mathcal{T}_4$, 137 for $\mathcal{T}_5$ and 382 for $\mathcal{T}_6$, and unlike prior omnimodal benchmarks every answer traces to human-made metadata, deterministic parsing, official transcripts or version chains, never to a model.

Construction. Four phases (Figure 3C) follow three rules: answers come only from authoritative structure, one model does one thing, and no human edits or picks an item. *Collect* admits redistributable sources with human-made annotation, so ASR, OCR and model descriptions may be system input but never gold. *Lift* normalizes every file and maps each source onto nine canonical contracts (records, alignments, events, . . .), so each task is written once for all personas. *Instantiate* fills templates from contract rows; GPT-5-nano only rewrites a stem into prose, and deterministic checks discard any item whose slots changed or whose answer leaks. *Verify* balances answers so that priors carry no signal and recomputes them with an independent implementation that agrees on all items (Appendix D.2).

Evaluation metrics. Scoring calls no model (Figure 3D). Free-form answers ($\mathcal{T}_1, \mathcal{T}_3, \mathcal{T}_4$) are matched one-to-one and leniently, a date at the gold's precision in any format and any other value on normalized equality or a word-token F1 of at least 0.5, and score the set F1 of the matches. $\mathcal{T}_2$ and $\mathcal{T}_5$ are four-way choices, a $\mathcal{T}_5$ item scoring half per correct phase, and $\mathcal{T}_6$ scores balanced accuracy over positive and negative candidates. A task score is the mean item score (0–100), a persona score s_d the mean over its tasks, and $\text{OKE} = \sum_d s_d/10$, so every domain weighs the same, and per-persona scores are descriptive (Appendix D.3).

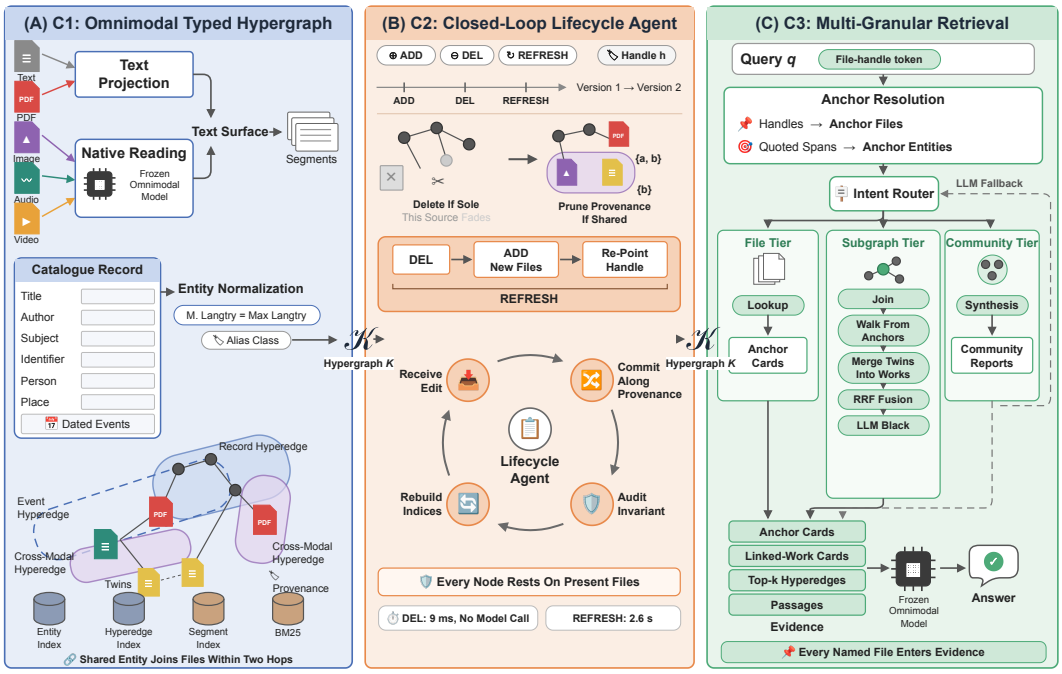


Figure 4: **KnowledgeClaw. (A)** C1 merges catalogue records into one omnimodal typed hypergraph $\mathcal{K}$. **(B)** C2 commits each edit in place along provenance. **(C)** C3 routes each query to the right tier.

4 Methodology: KnowledgeClaw

KnowledgeClaw (Figure 4) realizes the system $\mathcal{S}$ of §3 with one component per property and task group: C1 reads a persona $\mathcal{D}$ once into an omnimodal typed hypergraph $\mathcal{K}$ (§4.1), C2 commits each edit Δ to $\mathcal{K}$ in place (§4.2), and C3 answers every query q from exactly the tier of $\mathcal{K}$ it needs (§4.3).

4.1 C1: Omnimodal Typed Hypergraph

The artefact. Nodes are the files $a \in \mathcal{D}$ and the entities $v \in \mathcal{V}$ read from them; a hyperedge $e \in \mathcal{E}$ joins nodes and has a kind $\kappa(e)$, and $\pi(x)$ gives the source files of an entity or hyperedge x:

$$\mathcal{K} = (\mathcal{D}, \mathcal{V}, \mathcal{E}, \pi), \quad \mathcal{E} \subseteq 2^{\mathcal{D} \cup \mathcal{V}}, \quad \kappa : \mathcal{E} \to \{\textit{record}, \textit{event}, \textit{cross-modal}\}, \quad \pi : \mathcal{V} \cup \mathcal{E} \to 2^{\mathcal{D}}. \tag{1}$$

Native reading. A file a of modality $\mu(a) \in \mathcal{M}$ becomes a text surface $\tilde{a}$: text and PDF via a shared text projection proj, image, audio and video natively via the frozen omnimodal model Ω on the raw media $\mathrm{raw}(a)$ under a prompt $p_{\mu(a)}$; split then cuts $\tilde{a}$ into n_a segments $\sigma_{a,j}$ of $\sim$1,000 tokens:

$$\tilde{a} = \begin{cases} \mathrm{proj}(a), & \mu(a) \in \{\textit{text}, \textit{pdf}\}, \\ \Omega\big(\mathrm{raw}(a);\, p_{\mu(a)}\big), & \mu(a) \in \{\textit{image}, \textit{audio}, \textit{video}\}, \end{cases} \qquad (\sigma_{a,1}, \ldots, \sigma_{a,n_a}) = \mathrm{split}(\tilde{a}). \tag{2}$$

Catalogue records and identity. One call $\Omega(\tilde{a}; p_{\mathrm{rec}})$ writes the catalogue record of a: dated events and a value set $Z_{a,\gamma}$ per field γ of a fixed schema Γ (title, author, subject, id, …). A value z becomes the entity $\iota(z) = \mathrm{norm}(z)$, each field a *record* hyperedge and each event an *event* hyperedge $\{a\}$:

$$e_{a,\gamma} = \{a\} \cup \big\{\iota(z) : z \in Z_{a,\gamma}\big\}, \qquad \pi(e_{a,\gamma}) = \{a\}, \qquad \pi(v) \leftarrow \pi(v) \cup \{a\} \quad \forall\, v \in e_{a,\gamma} \cap \mathcal{V}, \tag{3}$$

so records meet on shared entities, $\mathcal{V}(a)$ being the entities of a's record hyperedges; a name key ι_{p} (first initial, last name) places "M. Langtry" and "Max Langtry" in one alias class $[v]$ for the walk.

Cross-modal coreference. An entity whose files span $\geq$2 modalities gets a *cross-modal* hyperedge, and files of different modalities sharing a title or identifier are *twins*, one work held twice:

$$e_v^{\times} = \{v\} \cup \pi(v) \text{ if } |\mu(\pi(v))| \geq 2, \qquad a \approx b \iff \mu(a) \neq \mu(b) \wedge b \in \pi\big(\mathcal{V} \cap (e_{a,\mathrm{title}} \cup e_{a,\mathrm{id}})\big), \tag{4}$$

and an encoder ϕ embeds entities, hyperedges and segments into dense indices $I_{\mathcal{V}}, I_{\mathcal{E}}, I_{\sigma}$; BM25 indexes every surface $\tilde{a}$, and each index key names the entity, hyperedge or segment that it encodes.

Proposition 1. *Any two files whose records name a common entity are joined within two hyperedges.*

Proof. Empirical support is in §5.4 (Figure 6) and the proof in Appendix B.1. □

4.2 C2: CLOSED-LOOP LIFECYCLE AGENT

Edits and soundness. A handle h names a document across its versions. Let $\mathcal{D}_t$ be the files present after the t-th edit Δ_t and $\eta_t(h)$ the files of the newest version of h; the three edits of §3 act as

$$\mathcal{D}_t = \begin{cases} \mathcal{D}_{t-1} \cup A, & \Delta_t = \textsc{Add}(A), \\ \mathcal{D}_{t-1} \setminus \eta_{t-1}(h), & \Delta_t = \textsc{Del}(h), \\ \big(\mathcal{D}_{t-1} \setminus \eta_{t-1}(h)\big) \cup A', & \Delta_t = \textsc{Refresh}(h \to A'), \text{ after which } \eta_t(h) = A', \end{cases} \tag{5}$$

where A' may span several files (a PDF and its text) and construction is $\textsc{Add}(\mathcal{D})$ on an empty $\mathcal{K}_0$. Every edit restores the *provenance soundness* of $\mathcal{K}_t = (\mathcal{D}_t, \mathcal{V}_t, \mathcal{E}_t, \pi_t)$, with $\Sigma_t = \{\sigma_{a,j} : a \in \mathcal{D}_t\}$:

$$\forall x \in \mathcal{V}_t \cup \mathcal{E}_t : \ \emptyset \neq \pi_t(x) \subseteq \mathcal{D}_t, \qquad \mathrm{keys}(I_{\mathcal{V}}) \cup \mathrm{keys}(I_{\mathcal{E}}) \cup \mathrm{keys}(I_\sigma) \subseteq \mathcal{V}_t \cup \mathcal{E}_t \cup \Sigma_t. \tag{6}$$

Commits along provenance. $\textsc{Add}(A)$ runs C1 on the files A alone, all model calls in parallel before any write. $\textsc{Del}(h)$ calls no model; for each $a \in \eta_{t-1}(h)$ and node x with $a \in \pi(x)$,

$$\pi(x) = \{a\} \implies \text{delete } x \text{ and its index keys}, \qquad |\pi(x)| > 1 \implies \pi(x) \leftarrow \pi(x) \setminus \{a\}, \tag{7}$$

then drops orphaned entities and recomputes touched cross-modal hyperedges; $\textsc{Refresh}(h \to A')$ is $\textsc{Del}(h)$ then $\textsc{Add}(A')$, so no superseded value survives and Ω reads only the newly added files.

Proposition 2. *After any sequence of edits, all of $\mathcal{K}$ rests on present files, so no evidence is stale.*

Proof. Empirical support is in §5.5 (Figure 7) and the proof in Appendix B.2. □

4.3 C3: MULTI-GRANULAR RETRIEVAL WITH INTENT ROUTING

Anchors and intent routing. Handles $\mathrm{hdl}(q)$ in q resolve through η to anchor files A_q, and each quoted span $\psi \in \Psi_q$ to the entity $\iota(\psi)$ or its nearest entity in $I_{\mathcal{V}}$ at cosine $\geq \theta = 0.86$, giving anchor entities V_q. Relational, lookup and synthesis cues $(c_{\mathrm{rel}}, c_{\mathrm{look}}, c_{\mathrm{syn}})$ then route q to a tier $g(q)$:

$$A_q = \bigcup_{h \in \mathrm{hdl}(q)} \eta(h), \qquad g(q) = \begin{cases} \textit{file}, & A_q \neq \emptyset,\ V_q = \emptyset,\ \neg c_{\mathrm{rel}}(q), \\ \textit{subgraph}, & \text{else if } A_q \cup V_q \neq \emptyset \ \vee\ \big(c_{\mathrm{look}} \wedge \neg c_{\mathrm{syn}}\big)(q), \\ \textit{community}, & A_q \cup V_q = \emptyset \ \wedge\ c_{\mathrm{syn}}(q), \\ \Omega(q;\, p_{\mathrm{route}}), & \text{otherwise.} \end{cases} \tag{8}$$

File and subgraph tiers. The *file* tier answers a lookup with only the card $\mathrm{card}(a)$ of each $a \in A_q$ (its record, events and opening text). For a join, a walk scores each non-anchor file b by its links to the anchors, with $\mathbb{I}$ the indicator, $\pi([v])$ the files of an alias class, $\sqsubseteq$ normalized containment in the title and subject fields $\mathrm{head}(b)$ or surface $\tilde{b}$, and $\beta_\psi(b) \in [0,1]$ a scaled BM25 score:

$$\begin{aligned} s_q(b) = \sum\nolimits_{v \in V_q} 10\,\mathbb{I}\big[b \in \pi([v])\big] + \sum\nolimits_{\psi \in \Psi_q} \Big(6\,\mathbb{I}\big[\psi \sqsubseteq \mathrm{head}(b)\big] + 3\,\mathbb{I}\big[\psi \sqsubseteq \tilde{b}\big] + 2\,\beta_\psi(b)\Big) \\ + \sum\nolimits_{a \in A_q} \sum\nolimits_{u \in \mathcal{V}(a)} \log\big(1 + |\mathcal{D}_t|/|\pi(u)|\big)\,\mathbb{I}\big[b \in \pi([u])\big], \end{aligned} \tag{9}$$

skipping hubs. Twins merge into works ω; up to six within 0.6 of the best $S_q(\omega) = \sum_{b \in \omega} s_q(b)$ form W_q, and lists $L_q^{\mathcal{V}}, L_q^{\mathcal{E}}$ hold the hyperedges of the top-$2k$ entities of $I_{\mathcal{V}}$ and the top-$2k$ of $I_{\mathcal{E}}$:

$$\mathrm{rrf}_q(e) = \sum\nolimits_{L \in \{L_q^{\mathcal{V}},\, L_q^{\mathcal{E}}\}} \mathbb{I}[e \in L] \,/\, \big(\lambda + \mathrm{rank}_L(e)\big), \qquad \lambda = 4, \quad k = 10. \tag{10}$$

Community tier and answering. For synthesis, Ω summarizes the 64 largest Louvain clusters of entities under p_{com}. The evidence $\mathcal{X}_q$ lists the anchor cards, those of W_q, the top-k hyperedges by rrf_q and four best segments of I_σ (or the reports nearest to q), and $\hat{y}_q = \Omega(q, \mathcal{X}_q; p_{\mathrm{ans}})$ answers q.

Proposition 3. *Every file a question names enters its evidence, which top-k similarity cannot ensure.*

Proof. Empirical support is in §5.6 (Figure 8) and the proof in Appendix B.3. □

Table 1: **Main results on OmniKE-Bench.** $\mathcal{T}_1$–$\mathcal{T}_6$: per-task scores over ten personas; HC–GAM: per-persona means over tasks (descriptive); **OKE**: mean of persona columns. **Bold**: best; underline: best baseline; *italic*: ablation equal to the full model. Tags: input (text, multimodal, omnimodal), structure (none, chunk, graph, hypergraph) and lifecycle (yes, no).

	Tasks						*Personas*										
Method	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	HC	FIN	EDU	LAW	PSY	POL	HUM	BLD	MUS	GAM	**OKE**
No knowledge engineering																	
NaiveGeneration	0.0	28.0	0.8	1.8	37.2	59.8	17.9	13.4	22.0	10.8	22.8	9.2	10.0	16.9	18.4	18.7	16.0
Chunk / hierarchical retrieval																	
StandardRAG	6.1	27.2	9.5	12.7	51.8	55.7	30.4	18.6	29.2	13.9	32.8	7.5	14.0	19.3	23.9	20.6	21.0
RAPTOR	6.4	28.5	6.2	0.9	39.4	58.0	24.6	17.5	25.0	15.0	30.1	11.0	12.0	16.2	21.7	20.2	19.3
Graph RAG																	
LightRAG	13.6	31.0	9.5	2.7	50.0	58.0	30.6	24.5	29.7	22.2	33.3	6.7	18.9	20.7	24.7	27.7	23.9
GraphRAG	3.5	26.8	10.5	6.4	32.8	52.9	26.0	15.3	22.6	12.8	33.3	8.1	13.3	14.8	15.8	18.2	18.0
HippoRAG 2	3.1	24.5	8.8	4.5	46.7	53.1	27.7	18.1	22.1	11.6	23.8	7.5	11.2	19.2	17.3	26.2	18.5
Agentic graph RAG																	
KAG / OpenSPG	6.7	22.8	7.0	2.7	46.7	53.4	24.9	18.4	28.7	14.2	23.7	8.3	12.8	18.3	21.5	16.1	18.7
Youtu-GraphRAG	1.6	27.5	9.0	4.5	32.8	53.0	23.1	18.3	20.8	8.7	30.8	7.8	11.7	17.4	15.7	17.8	17.2
Knowledge graph construction																	
iText2KG	1.3	25.2	2.2	0.0	27.7	53.8	17.4	9.5	20.2	10.8	21.7	5.0	12.4	17.8	14.5	14.8	14.4
OntoKG	4.8	29.2	2.0	1.8	31.8	56.2	19.6	13.2	25.0	7.5	24.7	7.8	15.8	14.6	19.2	20.9	16.8
n-ary / hypergraph knowledge engineering																	
HyperGraphRAG	4.4	29.5	7.8	5.5	40.1	53.0	25.1	15.6	27.2	15.6	30.1	8.5	12.0	17.4	23.9	17.5	19.3
Hyper-KGGen	1.4	29.8	3.0	0.0	27.7	54.6	18.8	12.1	21.2	16.3	22.5	10.0	13.7	12.5	16.7	17.9	16.2
HyperRAG	1.1	30.2	6.5	2.7	31.8	53.1	25.4	13.3	23.7	9.2	30.4	10.8	13.3	13.8	18.0	13.0	17.1
Memory agents																	
MemGPT / Letta	8.8	27.2	4.8	4.5	34.3	55.4	26.7	18.2	21.9	17.1	25.6	10.1	11.4	18.6	18.7	16.2	18.4
A-MEM	7.8	25.2	10.2	10.0	49.3	58.1	29.2	20.1	28.0	12.1	31.5	8.9	13.1	20.2	26.3	20.2	21.0
Mem0	2.4	29.5	3.5	2.7	38.0	53.0	21.9	14.5	23.6	15.1	23.3	6.2	13.3	19.6	18.7	18.2	17.4
Graphiti (Zep)	10.9	66.0	12.0	5.5	58.8	68.5	37.0	35.6	43.1	35.0	50.1	18.3	37.2	26.6	30.5	38.8	35.2
MIRIX	1.6	34.0	3.2	1.8	38.7	55.1	21.7	13.9	23.9	13.8	25.5	12.0	13.7	19.1	24.1	18.5	18.6
MemOS	13.6	28.8	5.0	2.7	47.8	54.8	26.8	22.2	31.8	15.3	24.0	9.2	18.7	17.4	24.2	26.6	21.6
Self-evolving / closed-loop lifecycle																	
MemEvolve	0.0	27.0	0.5	4.5	32.5	60.6	19.9	13.0	22.1	10.0	23.7	11.7	8.3	15.9	17.6	10.0	15.2
ATOM	5.1	30.5	4.2	0.0	30.3	54.8	22.4	14.8	23.7	10.8	21.7	11.7	11.6	16.7	22.3	19.7	17.5
MemPro	11.4	78.2	13.5	8.2	78.8	69.5	43.5	45.2	50.7	33.4	53.6	23.6	36.5	37.4	43.4	45.5	41.3
Agentic knowledge engineering																	
OpenClaw	2.9	61.5	2.8	0.9	41.2	65.2	27.1	23.0	36.9	22.5	36.2	19.9	31.8	17.2	29.0	28.3	27.2
Graphify	0.9	35.8	1.2	0.0	35.8	52.4	18.3	20.2	22.8	15.9	28.3	10.2	10.5	14.2	20.7	16.4	17.8
Cognee	12.8	44.0	5.0	2.7	45.3	59.7	28.2	28.2	38.1	18.1	30.9	21.5	11.7	19.4	28.9	30.4	25.6
LLM Wiki	20.0	66.2	6.5	0.0	50.7	68.1	33.9	36.5	45.8	32.5	44.5	21.6	42.4	24.2	36.5	34.8	35.3
OKF	27.2	68.5	8.8	17.3	54.4	67.3	45.5	44.3	46.3	38.9	49.7	16.8	37.6	27.0	38.7	40.6	38.5
Omnimodal / multimodal knowledge engineering																	
Omni-SimpleMem	10.3	70.2	10.2	10.9	52.2	68.8	40.6	34.0	44.7	29.9	48.3	20.2	26.6	26.0	36.4	37.9	34.5
MMGraphRAG	1.6	32.0	10.5	1.8	32.5	51.2	23.9	17.8	21.9	14.7	31.9	9.2	14.0	16.7	19.1	18.6	18.8
M^{3}KG-RAG	17.2	60.8	1.5	0.0	32.8	60.9	30.4	32.6	39.3	30.1	41.8	15.1	29.7	17.7	25.7	25.9	28.8
Ablations																	
w/o C1 (no hypergraph)	31.6	81.2	11.0	7.3	95.3	74.6	51.7	55.2	57.8	42.6	58.1	25.8	51.8	43.4	53.5	50.5	49.1
w/o C2 (no lifecycle)	*51.6*	*82.8*	*15.8*	*20.9*	49.6	*76.0*	*60.9*	52.3	50.5	*46.6*	*65.2*	*28.3*	*57.3*	35.3	53.3	54.5	50.4
w/o C3 (flat retrieval)	15.7	34.8	8.0	5.5	62.8	57.5	28.2	22.3	36.7	15.6	35.2	16.7	20.7	31.6	30.0	29.5	26.7
KnowledgeClaw (ours)	**51.6**	**82.8**	**15.8**	**20.9**	**95.6**	**76.0**	**60.9**	**64.4**	**59.2**	**46.6**	**65.2**	**28.3**	**57.3**	**47.3**	**64.5**	**65.7**	**55.9**

5 EXPERIMENTS

We answer eight RQs: **RQ1–RQ2:** Does KnowledgeClaw beat the 30 baselines, and does each component count? **RQ3–RQ5:** Does C1 join the files, C2 keep answers current and C3 supply the evidence each query needs? **RQ6–RQ7:** Is it efficient and robust? **RQ8:** How does it resolve, from the raw media, an omnimodal query that sixteen of the 30 baselines answer incorrectly?

5.1 EXPERIMENTAL SETUP

Datasets and Evaluation Metrics. We evaluate on OmniKE-Bench (§3): 1,829 items over 3,960 real files of ten personas in five modalities. Scoring calls no model: set F1 for $\mathcal{T}_1$, $\mathcal{T}_3$ and $\mathcal{T}_4$, accuracy for $\mathcal{T}_2$ and $\mathcal{T}_5$, balanced accuracy for $\mathcal{T}_6$, and OKE averages the ten persona scores (Appendix D.3).

Baselines and Ablations. Thirty baselines in the ten groups of Table 1 span every line of §2, from NaiveGeneration, the frozen model with no corpus, through chunk, graph and hypergraph RAG to memory agents, agentic KE and omnimodal KE; each is re-implemented from its code or paper with its prompts and defaults, and all attempt the six tasks. The ablations w/o C1 (no hypergraph), w/o C2 (no lifecycle) and w/o C3 (flat retrieval) each remove one component of §4 (detailed in Appendix E).

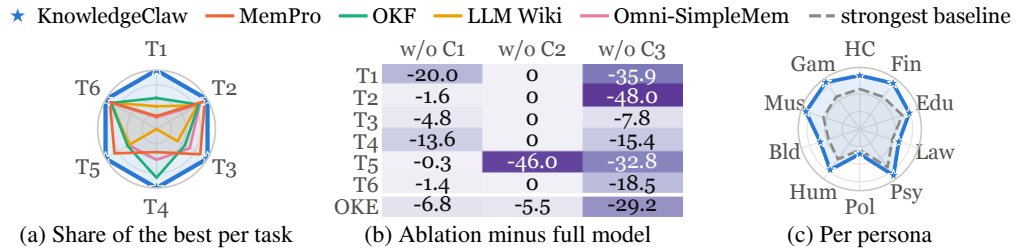
Implementation Details. All methods share one frozen omnimodal model Ω (GEMINI 2.5 FLASH LITE; Google, 2025), encoder ϕ (GEMINI EMBEDDING 2, 3,072 dimensions) and the answer prompt p_{ans}, so methods differ only in the evidence they supply; the omnimodal group and KnowledgeClaw read raw media natively, all other methods a shared media-to-text bridge. KnowledgeClaw uses $\theta = 0.86$, $\lambda = 4$, $k = 10$ and 64 communities for every persona and task (detailed in Appendix E.6).

5.2 MAIN RESULTS (RQ1)

As shown in Table 1, KnowledgeClaw ranks first on all six tasks and all ten personas with an OKE of 55.9, +14.6 over the strongest baseline MemPro: scaled to the best method of each task it traces the full outer ring (Figure 5a), and it beats the strongest baseline of every persona by +4.7 to +21.1 (Figure 5c; sign test $p = 0.002$). The five strongest baselines each have either a lifecycle (MemPro, OKF, LLM Wiki, Graphiti) or native media reading (Omni-SimpleMem), never both, and none of them exceeds 41.3 OKE: the three properties of OmniKE pay off only together.

5.3 ABLATION STUDY (RQ2)

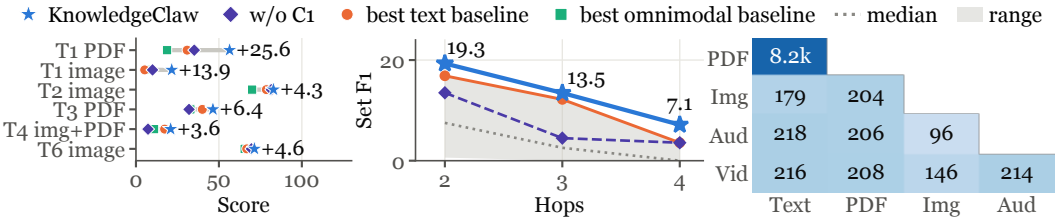
Every component is necessary (Figure 5b): without C1, $\mathcal{T}_1$ drops by 20.0 as media reach the model only through the text bridge; without C2, $\mathcal{T}_5$ falls by 46.0 as the superseded version stays; without C3, flat top-k retrieval loses on all six tasks and OKE drops by 29.2. Even with C1 or C2 removed, OKE stays above every baseline (49.1 and 50.4 against 41.3 for MemPro).



(a) Share of the best per task (b) Ablation minus full model (c) Per persona

Figure 5: **RQ1–RQ2.** (a) Score as % of the best per task. (b) From Table 1. (c) Mean over tasks.

5.4 HYPERGRAPH ANALYSIS (RQ3)



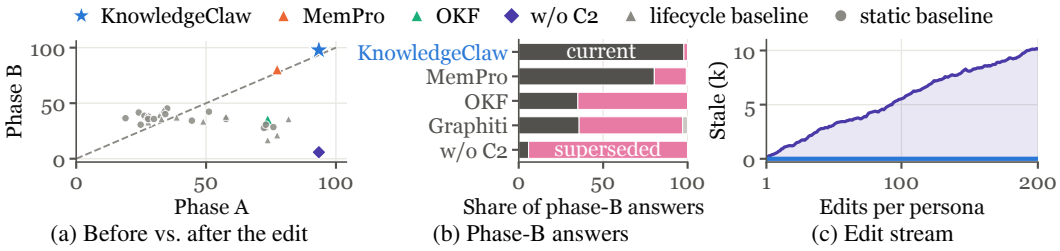
(a) Per-modality slices (b) Join depth (c) Cross-modal hyperedges

Figure 6: **C1 analysis.** (a) Label: gain over the best baseline. (b) $\mathcal{T}_3$ and $\mathcal{T}_4$ joins. (c) All personas.

Typed records lift every modality and hop. KnowledgeClaw leads both the best text-bridge and the best omnimodal baseline on all six modality slices, most on $\mathcal{T}_1$ PDFs (+25.6) and images (+13.9), where w/o C1 loses 21.3 and 11.6 (Figure 6a), and at every join depth, 19.3/13.5/7.1 at 2/3/4 hops, where the median of the 30 baselines falls to 0.0 (Figure 6b).

Joins cross every modality. C1 reads all 216 image, audio and video files of each persona natively into 3.6–6.1 MB of hypergraph whose cross-modal hyperedges join every pair of the five modalities (Figure 6c); 95.4–99.2% of files reach another within two hyperedges (Proposition 1), and the largest join gains fall on shared entities (+6.0) and file-name anchors (+4.9), where similarity has nothing to match, while every hyperedge keeps the files it was read from, so joins stay traceable.

5.5 Lifecycle Analysis (RQ4)



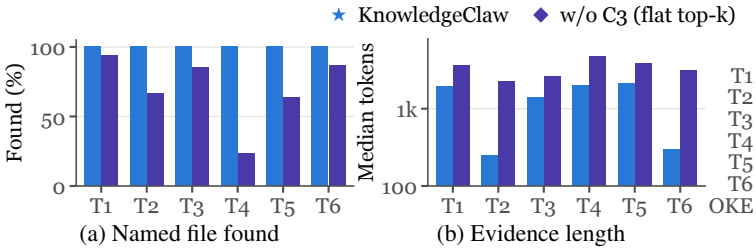
(a) Before vs. after the edit (b) Phase-B answers (c) Edit stream

Figure 7: **C2 analysis.** (a–b) $\mathcal{T}_5$ accuracy and answers. (c) Nodes without a present file per persona.

Current after every edit. KnowledgeClaw scores 93.4 before and 97.8 after the edit, while eight of the nine baselines above 70 before it fall below 36 after it (Figure 7a); it returns the superseded value for only 2.2% of the items, against 19.0% for MemPro and 94.2% for w/o C2 (Figure 7b).

Sound and cheap under a long edit stream. Over 2,000 edits, Eq. (6) never fails and no node without a present file survives (Proposition 2), while an append-only store accumulates 10,148 such nodes (Figure 7c); an edit touches a median 0.33% of the hypergraph and takes 9 ms (DEL) or 2.6 s (REFRESH), against 118 s for a full build.

5.6 Retrieval Analysis (RQ5)



	File	Walk	Com.	Flat	Ours
T1	**51.6**	36.1	4.1	15.7	**51.6**
T2	**82.8**	81.5	27.8	34.8	**82.8**
T3	1.5	**15.8**	1.2	8.0	**15.8**
T4	3.6	**20.9**	0.0	5.5	**20.9**
T5	**95.6**	95.3	28.8	62.8	**95.6**
T6	**76.0**	72.6	54.2	57.5	**76.0**
OKE	52.2	51.2	15.8	26.7	**55.9**

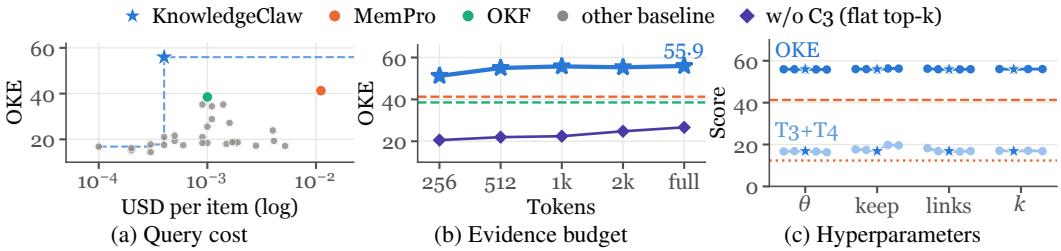
(a) Named file found (b) Evidence length (c) Forced granularity

Figure 8: **C3 analysis.** (a) Items with all named files in the evidence. (b) Per item. (c) Bold: best.

Named files arrive, in less evidence. C3 brings every named file's card into the evidence (Proposition 3), where flat top-k misses it on 6–77% (Figure 8a), with 1.8–10.5× shorter evidence (Figure 8b); on $\mathcal{T}_2$ the anchor card alone takes a median 248 tokens, against 2,250.

Routing matches the best fixed granularity. Each forced granularity fails on some task, file only on the joins (1.5 and 3.6) and walking on $\mathcal{T}_1$ (36.1), while the router equals the best on all six tasks and reaches 55.9 OKE, above file only (52.2), walk always (51.2) and flat top-k (26.7) (Figure 8c).

5.7 Efficiency and Robustness (RQ6–RQ7)



(a) Query cost (b) Evidence budget (c) Hyperparameters

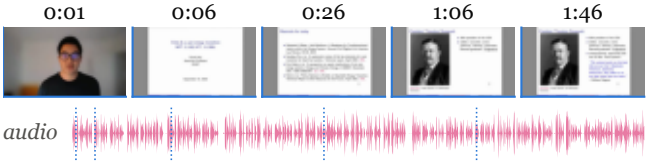
Figure 9: **Efficiency and robustness.** (a) Dashed: best OKE at or below each cost. (b) Evidence cut to its first B tokens; dashed: MemPro and OKF. (c) One hyperparameter varied; star: default.

The best score at a low cost. KnowledgeClaw spends $0.32 per persona to ingest, $0.0004 per query and 3 minutes to build (Table 8), less than MemPro on all three (Figure 9a); cut to 256 tokens, its evidence still yields 51.2 OKE, above every baseline even at full context (Figure 9b).

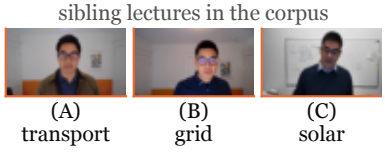
Stable across settings. Over 15 further settings of θ, the keep ratio, linked works and k, OKE stays within 55.8–56.3 and $\mathcal{T}_3$+$\mathcal{T}_4$ within 16.3–19.8, always above MemPro's 41.3 and 12.4 (Figure 9c): no setting has to be tuned per persona or per task to keep KnowledgeClaw ahead.

Table 2: **Case study** on item `T2_building_3261e3912859`. Frames, waveform and transcript come from the files, answers are saved predictions, and the last two rows follow saved traces.

Query ($\mathcal{T}_2$, building). *Which candidate description is the authoritative description of the video* `f75bfdd876eb3d22.mp4`*?* The four candidates open four lectures of one MIT class by the same speaker and differ only in the topic: (A) transportation systems, (B) the grid, (C) solar, (D) a just energy transition. **Gold:** (D).



[0:00] "...talk about cities and a just energy transition"



all four by David Hsu, one MIT class

Method	How the video reaches the answerer, and what is retrieved	Answer
NaiveGeneration	No corpus: the query names only a hashed file name, which carries no topic.	(C) solar ✗
StandardRAG	Text bridge of the video (30 s windows) cut into 1,200-token chunks; top-5 by similarity to the query.	(B) grid ✗
LightRAG	Entity–relation graph from the text bridge; query keywords retrieve entities, relations and their chunks.	(A) transport ✗
HippoRAG 2	Graph of triples from the text bridge; personalized PageRank from nodes matched to the query.	(A) transport ✗
HyperGraphRAG	n-ary hyperedges from the text bridge; query entities and segments retrieve hyperedges and chunks.	(B) grid ✗
Mem0	Memories distilled from the text bridge; recalled by similarity to the query.	(A) transport ✗
Cognee	Knowledge graph and chunk summaries from the text bridge; the top-5 triplets nearest to the query.	(A) transport ✗
MMGraphRAG	Text graph over the bridged transcript (raw media only for images); top-5 nodes by similarity.	(A) transport ✗
w/o C3	Our hypergraph with flat top-k: ten hyperedges, five of them identifiers of other videos, and six passages, none of them from the named video (10.5k characters).	(B) grid ✗
KnowledgeClaw	handle in q f75bf....mp4 → C3: anchor file A_q → card read natively by C1 title: Cities and a Just Energy Transition; person: T. Roosevelt *evidence: this card alone, 3.7k characters*	**(D) just transition** ✓

5.8 Case Study (RQ8)

Table 2 traces an item whose video is one of four sibling lectures with the same speaker, class and near-identical openings, which similarity to a query naming only a file cannot separate: sixteen of the 30 baselines choose a sibling's description. KnowledgeClaw resolves the handle to the anchor file and answers from its card alone, a record that C1 read from the raw video, so the topic, the one field in which the four candidates differ, is in the evidence, and it chooses the gold description (D) from 2.9× less evidence than w/o C3 (3.7k vs. 10.5k characters), all of it from the named video. C1 and C3 work together: native reading supplies the topic, and the anchor supplies the right file.

6 Conclusion

We formalized Omnimodal Knowledge Engineering, which asks a live knowledge base for omnimodality, a closed-loop lifecycle and multi-granular application at once, and released OmniKE-Bench, six model-free tasks over the real files of ten personas in five modalities. We proposed KnowledgeClaw, which reads every file natively into an omnimodal typed hypergraph, keeps it sound under edits with a closed-loop lifecycle agent, and routes each query to the granularity it needs. Against 30 baselines it ranks first on all six tasks and all ten personas and stays current after every edit. Future work will extend OmniKE-Bench to more personas, modalities and edit types.

AI USE STATEMENT

Apart from their role as the studied systems' backbone and the stem rewriter of OmniKE-Bench (§3), generative AI tools were used only for language polishing and coding assistance. All AI-assisted text and code were reviewed, verified and tested by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

OmniKE-Bench is built solely from public, redistributable sources under open licences (Table 6), involves no human subjects and collects no private user data. Every released file is stripped of its metadata, and no gold answer is produced by a model, so the benchmark introduces no model-generated misinformation.

REPRODUCIBILITY STATEMENT

Our code, OmniKE-Bench with its SHA-256 manifest and a one-command rebuild are released at the anonymous link in the abstract. All prompts, proofs, algorithms, benchmark construction, baseline configurations and evaluation protocol are given in Appendices A, B, C, D, E and F, with decoding parameters and random seeds fixed throughout.

REFERENCES

Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16776–16809, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.861. URL https://aclanthology.org/2025.findings-acl.861/.

Zafar Ali, Asad Khan, Aalia Malik, and Pavlos Kefalas. Graphloom: Reliability-calibrated graph evidence routing for multimodal kg-rag, 2026. URL https://arxiv.org/abs/2608.15056.

Weidong Bao, Yilin Wang, Ruyu Gao, Fangling Leng, Yubin Bao, and Ge Yu. Dial-kg: Schema-free incremental knowledge graph construction via dynamic schema induction and evolution-intent assessment. In Hyungsoo Jung, Tianzheng Wang, Masashi Toyoda, Hyuk-Yoon Kwon, and Jaewoong Lee (eds.), *Database Systems for Advanced Applications*, pp. 357–372, Singapore, 2026. Springer Nature Singapore. ISBN 978-981-92-0378-9.

Jiacheng Cai, Zijin Hong, Zheng Yuan, Huachi Zhou, Qinggang Zhang, and Xiao Huang. Crossmodalqa: A cross-modal and multi-hop benchmark for multimodal retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2609.05518.

Shenao Chen, Yidan Xu, Xiangmin Han, Rundong Xue, Duanpo Wu, Yuhan Gao, Chenggang Yan, and Yue Gao. Hypergraph-based multimodal retrieval-augmented generation with incremental refinement, 2026. URL https://arxiv.org/abs/2608.16628.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. In *ECAI 2025*. IOS Press, October 2025. ISBN 9781643686318. doi: 10.3233/faia251160. URL http://dx.doi.org/10.3233/faia251160.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding, 2024. URL https://arxiv.org/abs/2411.04952.

Yuzhuo Cui, Zongye Zhang, and Qingjie Liu. Livingrag: Augmenting graph rag with experience, 2026a. URL https://arxiv.org/abs/2608.25960.

Zeyang Cui, Jiannong Cao, Zhiyuan Wen, Bo Yuan, Junlan Feng, and Shengyuan Chen. Edgemem: Llm-free agent memory construction and retrieval via evidence-preserving multi-anchor hypergraph, 2026b. URL https://arxiv.org/abs/2609.05553.

Sijun Dai, Qiang Huang, Xiaoxing You, and Jun Yu. Mg^2-rag: Multi-granularity graph for multimodal retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2604.04969.

Junnan Dong, Siyu An, Yifei Yu, Qian-Wen Zhang, Linhao Luo, Xiao Huang, Yunsheng Wu, Di Yin, and Xing Sun. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning, 2025. URL https://arxiv.org/abs/2508.19855.

Junnan Dong, Linhao Luo, Senlei Zhang, Gong Chen, Taian Guo, Yifei Yu, Rong Tao, Tao Guo, Qian-Wen Zhang, Siyu An, Ruizhi Qiao, and Xing Sun. Wfm: Wiki foundation model for complex agentic reasoning, 2026. URL https://arxiv.org/abs/2609.18182.

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, 2025. URL https://arxiv.org/abs/2404.16130.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

Yichao Feng, Ran Zhang, Haoran Luo, Zhenghong Lin, Carl Yang, and Anh Tuan Luu. Diachronic hypergraphs for orchestrated multi-agent multimodal memory curation, 2026. URL https://arxiv.org/abs/2608.29678.

Zhenbo Fu, Yuanzhe Zhang, Qiange Wang, Hao Yuan, Yuehao Xu, Enze Yi, Yanfeng Zhang, and Ge Yu. Evorag: Making knowledge graph-based rag automatically evolve through feedback-driven backpropagation, 2026. URL https://arxiv.org/abs/2604.15676.

Xian Gao, Jinpeng Wang, Jiacheng Ruan, Guangyu Cao, Ting Liu, and Yuzhuo Fu. Memo: Multimodal evidence memory organization for long-horizon llm agents, 2026. URL https://arxiv.org/abs/2609.07471.

Google. Gemini 2.5 Flash-Lite. Official model documentation, 2025. URL https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-lite. Accessed September 22, 2026.

Google Cloud. Open Knowledge Format. GitHub repository, 2026. URL https://github.com/GoogleCloudPlatform/open-knowledge-format. Accessed September 22, 2026.

Graphify contributors. Graphify. GitHub repository, 2026. URL https://github.com/Graphify-Labs/graphify. Accessed September 22, 2026.

Zirui Guo, Xubin Ren, Lingrui Xu, Jiahao Zhang, and Chao Huang. Rag-anything: All-in-one rag framework, 2025a. URL https://arxiv.org/abs/2510.12323.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Chengrui Han and Zesheng Cheng. Raga: Reading-and-graph-building-agent for autonomous knowledge graph construction and retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2605.17072.

Xin He, Yili Wang, Wenqi Fan, Qing Li, Qinggang Zhang, Yi Chang, and Xin Wang. Hvm-graphrag: Holistic-view multimodal graph retrieval-augmented generation on complex document, 2026. URL https://arxiv.org/abs/2607.24861.

Chi-Hsiang Hsiao, Yi-Cheng Wang, Tzung-Sheng Lin, Yi-Ren Yeh, and Chu-song Chen. MegaRAG: Multimodal knowledge graph-based retrieval augmented generation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 48031–48059, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2218. URL https://aclanthology.org/2026.acl-long.2218/.

Hao Hu, Yifan Feng, Ruoxue Li, Rundong Xue, Xingliang Hou, Zhiqiang Tian, Yue Gao, and Shaoyi Du. Cog-rag: Cognitive-inspired dual-hypergraph with theme alignment retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31032–31040, March 2026a. ISSN 2159-5399. doi: 10.1609/aaai.v40i37.40363. URL http://dx.doi.org/10.1609/aaai.v40i37.40363.

Ruofan Hu, Shengyang Xu, Minjie Hong, Xiaoda Yang, Sashuai Zhou, Ke Lei, Tao Jin, and Zhou Zhao. Doc-refrag: Rethinking multimodal document retrieval-augmented generation, 2026b. URL https://arxiv.org/abs/2608.30163.

Xia Hu, Honglei Zhuang, Brian Potetz, Alireza Fathi, Bo Hu, Babak Samari, and Howard Zhou. Life-bench: A benchmark and knowledge graph framework for multimodal personalization beyond concept recognition, 2026c. URL https://arxiv.org/abs/2602.19001.

Rizhuo Huang, Yifan Feng, Rundong Xue, Shihui Ying, Jun-Hai Yong, Chuan Shi, Shaoyi Du, and Yue Gao. Hyper-kggen: A skill-driven knowledge extractor for high-quality knowledge hypergraph generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 1898–1909. ACM, August 2026. doi: 10.1145/3770855.3818198. URL http://dx.doi.org/10.1145/3770855.3818198.

Jingbo Ji, Lingyi Li, Xilong Cheng, Yuhao Zhou, Wenji Zhang, Yuting Tan, and Yunxiao Qin. Ripplemem: From isolated retrieval to associative recollection for long-term agent memory, 2026. URL https://arxiv.org/abs/2608.13334.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1709. URL https://aclanthology.org/2026.acl-long.1709/.

Andrej Karpathy. LLM Wiki. GitHub Gist, 2026. URL https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f. Accessed September 22, 2026.

Houda Khrouf, Pedro Fillastre, and Sebastiao Correia. Optimizing hypergraph-based rag: Toward better fact extraction and chunk retrieval, 2026. URL https://arxiv.org/abs/2607.20506.

Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. itext2kg: Incremental knowledge graphs construction using large language models. In Mahmoud Barhamgi, Hua Wang, and Xin Wang (eds.), *Web Information Systems Engineering – WISE 2024*, pp. 214–229, Singapore, 2025. Springer Nature Singapore. ISBN 978-981-96-0573-6.

Yassir Lairgi, Ludovic Moncla, Khalid Benabdeslem, Rémy Cazabet, and Pierre Cléau. ATOM: AdapTive and OptiMized dynamic temporal knowledge graph construction using LLMs. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational*

Linguistics: EACL 2026, pp. 950–966, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.49. URL https://aclanthology.org/2026.findings-eacl.49/.

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, 2025. URL https://arxiv.org/abs/2512.12818.

EunKyeong Lee, Kyeong-Jin Oh, Jinwon Kim, Hye Woo Lee, Minsang Song, Hyeongjun Jang, and Junyoung Youn. Mosaic: Query-aware exploration policy adaptation for graphrag, 2026. URL https://arxiv.org/abs/2609.11065.

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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Geng Li, Yuhao Wang, Dong Li, Jianye Hao, and Yuxin Peng. Graphmemix: Query-aware evidence forests for long-term multimodal agent memory, 2026a. URL https://arxiv.org/abs/2608.26983.

Jiazheng Li, Chi-Hao Wu, Yunze Liu, Kaize Ding, Jundong Li, and Chuxu Zhang. Bridging modalities, spanning time: Structured memory for ultra-long agentic video reasoning, 2026b. URL https://arxiv.org/abs/2605.08271.

Yitao Li, Zhanlin Liu, Anuranjan Pandey, and Muni Srikanth. Ontokg: Ontology-oriented knowledge graph construction with intrinsic-relational routing, 2026c. URL https://arxiv.org/abs/2604.02618.

Zhaoyang Li, Zenghuang Fu, Qiuyuan Ai, Ping Jiang, Haoyu Wu, Minghui Wu, Chenxu Zhao, Jie Song, and Guannan He. Coevokg: Co-evolving knowledge graphs with self-evolving search agents, 2026d. URL https://arxiv.org/abs/2608.01904.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Huayi Lai, Hao Wu, Bo Tang, Zhengren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofen Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. Memos: A memory os for ai system, 2025. URL https://arxiv.org/abs/2507.03724.

Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 334–343. ACM, May 2025. doi: 10.1145/3701716.3715240. URL http://dx.doi.org/10.1145/3701716.3715240.

Wen-Sheng Lien, Yu-Kai Chan, Hao-Lung Hsiao, Bo-Kai Ruan, Meng-Fen Chiang, Chien-An Chen, Yi-Ren Yeh, and Hong-Han Shuai. Hyperrag: Reasoning n-ary facts over hypergraphs for retrieval augmented generation. In *Proceedings of the ACM Web Conference 2026*, pp. 2465–2476. ACM, April 2026. doi: 10.1145/3774904.3792710. URL http://dx.doi.org/10.1145/3774904.3792710.

Jiashi Lin, Changhong Jiang, Xiangru Lin, Ruifei Zhang, Xinyi Zhu, Jiyao Liu, Cheng Tang, Ye Du, Shujian Gao, Junzhi Ning, Lihao Liu, Ziyan Huang, Tianbin Li, Jin Ye, and Junjun He. Evograph-r1: Self-evolving multimodal knowledge hypergraphs for agentic retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 756–765, June 2026.

Jiaqi Liu, Zipeng Ling, Shi Qiu, Yanqing Liu, Siwei Han, Peng Xia, Haoqin Tu, Zeyu Zheng, Cihang Xie, Charles Fleming, Mingyu Ding, and Huaxiu Yao. Omni-simplemem: Autoresearch-guided discovery of lifelong multimodal agent memory, 2026a. URL https://arxiv.org/abs/2604.01007.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents, 2026b. URL https://arxiv.org/abs/2601.02553.

Junming Liu, Siyuan Meng, Yanting Gao, Song Mao, Pinlong Cai, Guohang Yan, Yirong Chen, Zilin Bian, Ding Wang, and Botian Shi. Aligning vision to language: Annotation-free multimodal knowledge graph construction for enhanced llms reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 981–992, October 2025.

Junming Liu, Yifei Sun, Weihua Cheng, Haodong Lei, Yirong Chen, Licheng Wen, Xuemeng Yang, Daocheng Fu, Pinlong Cai, Nianchen Deng, Yi Yu, Shuyue Hu, Botian Shi, and Ding Wang. Memverse: Multimodal memory for lifelong learning agents, 2026c. URL https://arxiv.org/abs/2512.03627.

Qingshan Liu, Guoqing Wang, Wen Wu, Jingqi Huang, Xinqi Tao, Dejia Song, Jie Zhou, and Liang He. Mempro: Agentic memory systems as evolvable programs, 2026d. URL https://arxiv.org/abs/2606.00619.

Zeyu Liu, Jian Zhong, Rongduo Han, Ziyang Wu, Shunye Tang, Chenghao He, Yaxuan Yang, Yihang Qiu, Ailing Wang, Xiao Liang, Guohuan Xie, Xiaokang Xue, Gongchen Li, Haining Zhang, and Wei Wang. Evalmem: An operation-level diagnostic framework for long-term memory systems, 2026e. URL https://arxiv.org/abs/2609.22231.

Yuxing Lu, Wei Wu, Xukai Zhao, Rui Peng, and Jinzhuo Wang. Karma: Leveraging multi-agent llms for automated knowledge graph enrichment. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 56421–56453. Curran Associates, Inc., 2025. doi: 10.52202/085713-1887. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/517f9b9c227b9dd51dba4560f37165ed-Paper-Conference.pdf.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Anh Tuan Luu. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 152206–152234. Curran Associates, Inc., 2025. doi: 10.52202/085713-5089. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/df55ee6e59f8ac4a625219e11fe9ddba-Paper-Conference.pdf.

Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Anh Tuan Luu. Graph-r1: Towards agentic graphRAG framework via end-to-end reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YXnFGsSkCC.

Menghe Ma, Siqing Wei, Yuecheng Xing, Ziyue Zhu, Zhenghong Lin, Yaheng Wang, Fanhong Meng, Peijun Han, Luu Anh Tuan, and Haoran Luo. Onote: Hypergraph-grounded omnimodal reasoning for computational music science, 2026a. URL https://arxiv.org/abs/2604.20719.

Xiaokang Ma, Yifan Sun, Zhihong Jin, Jie Gu, Yudong Luo, Shenyi Shao, Chu Tang, Jingmin Chen, and Li Pu. Reflectworld-mm: An entity-oriented multimodal memory system for open-ended video streams, 2026b. URL https://arxiv.org/abs/2607.09759.

Vasilije Markovic, Lazar Obradovic, Laszlo Hajdu, and Jovan Pavlovic. Optimizing the interface between knowledge graphs and llms for complex reasoning, 2025. URL https://arxiv.org/abs/2505.24478.

Haoliang Ming, Feifei Li, Xiaoqing Wu, and Wenhui Que. Retrieval as reasoning: Self-evolving agent-native retrieval via llm-wiki, 2026. URL https://arxiv.org/abs/2605.25480.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

OpenClaw contributors. OpenClaw. GitHub repository, 2026. URL https://github.com/openclaw/openclaw. Accessed September 22, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024. doi: 10.1109/TKDE.2024.3352100.

Hyeongcheol Park, Jiyoung Seo, Jaewon Mun, Hogun Park, Wonmin Byeon, Sung June Kim, Hyeonsoo Im, JeungSub Lee, and Sangpil Kim. Mˆ3kg-rag: Multi-hop multimodal knowledge graph-enhanced retrieval-augmented generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14968–14978, June 2026a.

Jinyoung Park, Sanghyeok Lee, Omar Zia Khan, Hyunwoo J. Kim, and Joo-Kyung Kim. Hypergraphpro: Progress-aware reinforcement learning for structure-guided hypergraph rag, 2026b. URL https://arxiv.org/abs/2601.17755.

An Nguyen Phu, Dung Nguyen Quang, Luu Hieu An, Linh Ngo Van, Trung Le, and Thien Huu Nguyen. Hyperprove: Answer-guided hypergraph expansion for multi-hop question answering, 2026. URL https://arxiv.org/abs/2609.13768.

Tithi Rakshit, Hongkuan Zhou, Lavdim Halilaj, and Yuqicheng Zhu. Less is more: Graph-free multimodal rag via multi-signal late fusion, 2026. URL https://arxiv.org/abs/2609.19417.

Shi-Ju Ran, Kun Zhang, Xi Wu, Liu-Si Yang, and Wen-Jun Li. Llms interpret, embeddings organize, graphs emerge: Agent-driven compilation of scientific knowledge, 2026. URL https://arxiv.org/abs/2608.29612.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Xubin Ren, Lingrui Xu, Long Xia, Shuaiqiang Wang, Dawei Yin, and Chao Huang. Videorag: Retrieval-augmented generation with extreme long-context videos. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 2390–2401. ACM, April 2026. doi: 10.1145/3770854.3783944. URL http://dx.doi.org/10.1145/3770854.3783944.

Florian Ruosch and Luca Rossetto. Mediagraph: A content-aware data model and query framework for multimodal knowledge graphs, 2026. URL https://arxiv.org/abs/2608.18895.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Manuel Serna-Aguilera, Raegan Anderes, Page Dobbs, and Khoa Luu. Nico-rag: Multimodal hypergraph retrieval-augmented generation for understanding the nicotine public health crisis, 2026a. URL https://arxiv.org/abs/2603.02047.

Manuel Serna-Aguilera, Raegan Anderes, Page Dobbs, and Khoa Luu. Prism-rag: Multimodal hypergraph retrieval-augmented generation for tobacco product and legislative policy reasoning, 2026b. URL https://arxiv.org/abs/2609.23769.

Yi Ting Shen, Kentaroh Toyoda, and Alex Leung. Revoked but still authoritative: An empirical study of revocation enforcement in agent-memory systems, 2026. URL https://arxiv.org/abs/2609.08258.

Rudi Studer, V.Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1):161–197, 1998. ISSN 0169-023X. doi: https://doi.org/10.1016/S0169-023X(97)00056-6. URL https://www.sciencedirect.com/science/article/pii/S0169023X97000566.

Susheel Suresh, Hazel Mak, Sahil Bhatnagar, Chhaya Methani, and Alejandro Gutierrez Munoz. Grounding agent memory: Environment-probing curation for enterprise agents, 2026. URL https://arxiv.org/abs/2609.11060.

Yang Tian, Fan Liu, Jingyuan Zhang, Zhenyang Li, Yupeng Hu, and Liqiang Nie. Mm-contextfold: Context folding for multimodal agentic retrieval, 2026. URL https://arxiv.org/abs/2609.23121.

Hong Ting Tsang, Jiaxin Bai, Haoyu Huang, Qiao Xiao, Tianshi Zheng, Baixuan Xu, Shujie Liu, and Yangqiu Song. AutoGraph-r1: End-to-end reinforcement learning for knowledge graph construction. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23351–23374, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1070. URL https://aclanthology.org/2026.acl-long.1070/.

Xueyao Wan and Hang Yu. Mmgraphrag: Bridging vision and language with interpretable multimodal knowledge graphs, 2026. URL https://arxiv.org/abs/2507.20804.

Changjian Wang, Weihong Deng, Weili Guan, Quan Lu, and Ning Jiang. Cross-granularity hypergraph retrieval-augmented generation for multi-hop question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33368–33376, March 2026. ISSN 2159-5399. doi: 10.1609/aaai.v40i39.40623. URL http://dx.doi.org/10.1609/aaai.v40i39.40623.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 3 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents, 2025. URL https://arxiv.org/abs/2507.07957.

Ziming Wang. Toki: A bitemporal operator algebra for contradiction resolution in llm-agent persistent memory, 2026. URL https://arxiv.org/abs/2606.06240.

Keshu Wu, Chenchen Kuai, Zihao Li, Jiwan Jiang, Shiyu Shen, Shian Wang, Chan-Wei Hu, Zhengzhong Tu, and Yang Zhou. Knowledge is not static: Order-aware hypergraph rag for language models, 2026a. URL https://arxiv.org/abs/2604.12185.

Zongyu Wu, Yilong Wang, Xiaochen Wang, Minhua Lin, Zhichao Xu, Fenglong Ma, Xiang Zhang, and Suhang Wang. Multi-granularity context-enhanced rag over multimodal knowledge graphs, 2026b. URL https://arxiv.org/abs/2608.25986.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: a survey. *Frontiers of Computer Science*, 18(6):186357, 11 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40555-y. URL https://doi.org/10.1007/s11704-024-40555-y.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz,

M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Meriem Yacoubi, Pia Schmidt, Nenad Petrovic, Ahmed Frikha, Martin Kirchhoff, and Alois Knoll. Memorylace: Memory lifecycle-aware consolidation and evidence retrieval, 2026. URL https://arxiv.org/abs/2609.03201.

Neeraj Yadav. Temporal validity in retrieval memory: Eliminating stale-fact errors for ai agents over evolving knowledge, 2026. URL https://arxiv.org/abs/2606.26511.

Cehao Yang, Xiaojun Wu, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Jia Li, Hui Xiong, and Jian Guo. Graphsearch: An agentic deep searching workflow for graph retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2509.22009.

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. Graph-based agent memory: Taxonomy, techniques, and applications, 2026a. URL https://arxiv.org/abs/2602.05665.

Kairui Yang, Minghao An, Xunkai Li, Ziheng Yi, Zekai Chen, Guangyuan He, and Rong-Hua Li. Mace: Memory-agent co-evolution with adaptive memory graphs for multi-agent systems, 2026b. URL https://arxiv.org/abs/2609.21533.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25599–25609, June 2026.

Li Yuan, Qingfei Huang, Bingshan Zhu, Yi Cai, Qingbao Huang, Changmeng Zheng, Zikun Deng, and Tao Wang. Hybrid-dmkg: A hybrid reasoning framework over dynamic multimodal knowledge graphs for multimodal multihop qa with knowledge editing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):28032–28040, March 2026a. ISSN 2159-5399. doi: 10.1609/aaai.v40i33.40028. URL http://dx.doi.org/10.1609/aaai.v40i33.40028.

Xu Yuan, Liangbo Ning, Qingqing Ye, Wenqi Fan, and Qing Li. mkg-rag: Leveraging multimodal knowledge graphs in retrieval-augmented generation for knowledge-intensive vqa. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2274–2285. ACM, July 2026b. doi: 10.1145/3805712.3809680. URL http://dx.doi.org/10.1145/3805712.3809680.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35237–35254, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1627. URL https://aclanthology.org/2026.acl-long.1627/.

Xiangjun Zai, Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, and Wenjie Zhang. Proh: Dynamic planning and reasoning over knowledge hypergraphs for retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2510.12434.

Guangzhe Zhang. Correct now, insufficient later: Auditing update sufficiency in context compression, 2026. URL https://arxiv.org/abs/2609.20045.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. Memevolve: Meta-evolution of agent memory systems, 2025. URL https://arxiv.org/abs/2512.18746.

Suifeng Zhao, Zida Liu, Xinyu Lei, Lei Sun, Jun Gao, and Sujian Li. Mcite-rl: Towards reliable multimodal rag via citation-enhanced agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2608.21808.

Chulun Zhou, Chunkang Zhang, Guoxin Yu, Fandong Meng, Jie Zhou, Wai Lam, and Mo Yu. HG-Mem: Hypergraph-based working memory to improve multi-step RAG for long-context complex relational modeling. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dvrq3GX7qL.

Yifan Zhu, Xinyu Mu, Tao Feng, Zhonghong Ou, Yuning Gong, and Haoran Luo. Omnirag-agent: Agentic omnimodal reasoning for low-resource long audio-video question answering, 2026. URL https://arxiv.org/abs/2602.03707.

APPENDIX

A PROMPTS USED IN KNOWLEDGECLAW

This section gives, verbatim, every prompt that KnowledgeClaw sends to the frozen model Ω (GEMINI 2.5 FLASH LITE); Table 3 places each in §4. In the boxes, $\{\dots\}$ marks a slot filled at run time and $\langle\dots\rangle$ a raw media part. All calls run at temperature 0, in JSON mode when the reply is JSON. C1 owns four prompts and C3 three; C2 adds none: DEL calls no model, and ADD and REFRESH run the prompts of C1 on the files they add, so edits and construction share one extraction path and an edit reads only the files it adds.

Table 3: The prompts of KnowledgeClaw; *Max* is the output-token limit of the call.

Prompt	Used in	Input	Output	Calls	Max
p_{image}	C1, Eq. (2)	image, $\leq$1,536 px	description, visible text	per image	1,000
p_{audio}	C1, Eq. (2)	audio, 16 kHz mono	summary, timed transcript	per recording	3,000
p_{video}	C1, Eq. (2)	$\leq$6 keyframes, audio	description, transcript	per video	3,000
p_{rec}	C1, Eq. (3)	first 9,000 chars of $\tilde{a}$	JSON record	per file	2,500
p_{route}	C3, Eq. (8)	query q	one label	if no rule routes q	5
p_{com}	C3, communities	entities, propositions	report, $\leq$200 words	per community	400
p_{ans}	C3, answering	evidence $\mathcal{X}_q$, question	JSON answer	per query	600

A.1 READING PROMPTS $p_{\mu(a)}$

Eq. (2) reads each image, audio and video file once, natively, under the prompt of its modality (Figure 10). Since the reading $\tilde{a}$ is what the record prompt catalogues, each prompt asks for what tells a file apart from its look-alikes rather than for a caption: the object type, depicted content, technique and every legible word of an image, with an inventory for a floor plan; the speakers and a time-marked transcript of a recording; and both, with on-screen text, for a video, which is sent as up to six time-labelled keyframes followed by its audio track. Each reading is then split into segments of about 1,000 tokens that keep the modality of their file.

Image reading prompt p_{image}

You are indexing one image for a personal knowledge base, so that it can later be told apart from similar images and matched to its catalogue record. Write plain prose (at most 250 words), then one line "Visible text:" with every legible word verbatim (or "none").
Cover, when present: what kind of object the image is (photograph, photomicrograph, electron micrograph, engraving, drawing, painting, portrait, map, chart, floor plan, diagram, screenshot, poster, document scan, object photo, ...); what it depicts -- people (how many, age group, sex, activity, clothing), organisms, anatomy, objects, instruments, buildings, places, setting and apparent period; technique (stain, magnification, colour or black-and-white).
For a floor plan or technical diagram, add a line "Inventory:" that counts what is drawn, by type: rooms (each room type with its count), walls, windows, doors, stairs, columns, railings, and every fixed fitting or furniture item (sinks, toilets, showers, bathtubs, cabinets, closets, stoves, appliances, sauna benches, ...) with its count.
Do not guess names, places or dates that are not visible.
⟨**image**⟩

Audio reading prompt p_{audio}

You are indexing one audio recording for a personal knowledge base. First give a 1-3 sentence summary: what the recording is (lecture, interview, talk, song, sound effect, ...), who speaks (names if said), and the topic or musical character (genre, instruments, mood, tempo). Then write "Transcript:" and transcribe all speech verbatim, with a time mark such as [0:30] about every 30 seconds. If there is no speech, write "Transcript: none".
⟨**audio**⟩

Video reading prompt p_{video}

You are indexing one video for a personal knowledge base. You are given keyframes (with their times) and the audio track. First describe what the video shows in 2-4 sentences: kind of video (lecture, interview, demonstration, gameplay, animation, ...), people, setting, and all on-screen text verbatim. Then write "Transcript:" and transcribe all speech verbatim, with a time mark such as [0:30] about every 30 seconds (or "Transcript: none").
Frame at 0:10: ⟨**keyframe**⟩ Frame at 0:30: ⟨**keyframe**⟩ ... Frame at 1:50: ⟨**keyframe**⟩ Audio track: ⟨**audio**⟩

Figure 10: Reading prompts $p_{\mu(a)}$ of Eq. (2) for image, audio and video files; each prompt is sent together with the raw media part(s) shown last.

A.2 RECORD PROMPT p_{rec}

Eq. (3) writes each file's catalogue record with one call of the prompt in Figure 11 on the first $9{,}000$ characters of $\tilde{a}$, so documents and media files share one schema Γ of 13 fields and dated events. Two fields also fix the form of a value, since a catalogue join matches forms, not topics: *subject* asks for each vocabulary's preferred form (arXiv codes, MeSH descriptors and check tags), and *element* for canonical singular class names, so that a bedroom in a plan and in a photograph is one entity.

Record prompt p_{rec}

You are cataloguing one file of a personal knowledge base. From the file content below, write its catalogue record. Copy names, identifiers and dates exactly as they appear; never invent a value that the content does not support.

Return JSON only:

```
{"title": "<title as printed, or a short descriptive title>",
  "kind": "<what the file is: preprint, journal article, report, bill, thesis, abstract, photograph, photomicrograph, drawing,
      floor plan, lecture recording, interview, song, ...>",
  "summary": "<one sentence>",
  "fields": {
    "author": ["<each author's full name, in order>"],
    "affiliation": ["<each distinct affiliation, the full string as printed>"],
    "subject": ["<the controlled-vocabulary subject terms a library catalogue would assign to this file, each in the
        vocabulary's exact preferred form: arXiv category codes (e.g. cs.CL, eess.AS, q-bio.NC) for arXiv preprints; 5-15
        MeSH descriptors for biomedical articles, written as MeSH writes them (plural nouns, inverted forms with
        commas), covering the population studied, the conditions, organisms, interventions, settings and methods, plus
        check tags (Humans, Male, Female, age groups) when the study population supports them; policy areas for
        government reports; subject headings otherwise>"],
    "keyword": ["<author keywords or key topics>"],
    "identifier": ["<DOI, arXiv id, report or bill number, PMID, ...>"],
    "venue": ["<journal, conference, publisher or series>"],
    "organization": ["<other organisations named>"],
    "person": ["<other people named: speakers, editors, people shown>"],
    "place": ["<places>"],
    "depicts": ["<for media: what is shown or heard: people, organisms, objects, activities>"],
    "object_type": ["<for media: the kind of object, e.g. Photographs, Portraits, Engravings, Maps>"],
    "element": ["<for plans, diagrams and building photos: every room type and building element shown, each as a
        canonical singular class name (e.g. Bedroom, Kitchen, Door, Window, Wall, Stairs, Balcony, Roof)>"],
    "language": ["<language of the content>"]
  },
  "events": [{"event": "<submitted | received | revised | accepted | published | published online | published print | created |
      recorded | issued | ...>", "date": "<YYYY-MM-DD, YYYY-MM or YYYY>"}]}
```

Rules: leave a field as [] when the content does not state it. An arXiv stamp such as "arXiv:2205.01234v2 [cs.CL] 24 May 2022" gives the identifier, the primary subject category and the submission date of that version. Journal front matter gives received / accepted / published dates.

FILE:
{content}

Figure 11: Record prompt p_{rec} of Eq. (3); **{content}** is the opening of $\tilde{a}$.

From record to hyperedges. The title and each non-empty field of the JSON reply become one *record* hyperedge that holds the file and all the values of the field; each event with a valid date becomes an *event* hyperedge, its date normalised to YYYY-MM-DD, YYYY-MM or YYYY. Every value enters through ι, so a value shared by two records is one entity with two source files, the step a join takes.

A.3 ROUTING PROMPT p_{route}

Eq. (8) routes by rule a query that names a file or entity, is short, or has a lookup or synthesis cue; others go to Figure 12, whose one-word reply picks the subgraph or community tier or both (*hybrid*).

Routing prompt p_{route}

Classify the query into ONE retrieval strategy over a personal knowledge hypergraph.
subgraph: a precise fact, definition, who/when/where/which. community: open-ended synthesis or a thematic overview.
hybrid: ambiguous, or needs both.
Reply with exactly one word: subgraph, community or hybrid.
Query: **{query}**

Figure 12: Routing prompt p_{route}, the last case of Eq. (8).

A.4 COMMUNITY REPORT PROMPT p_{com}

The community tier summarises each of the 64 largest Louvain communities with the prompt of Figure 13. Its slots hold the names of up to 15 member entities, up to 12 propositions, each the text of a record hyperedge on a member, and the modalities those hyperedges were read from. Louvain runs with a fixed seed on the entity graph whose edge weight counts the record hyperedges two entities share. The reports, at most 200 words in four fixed parts, form the dense index that the community tier searches with $\phi(q)$.

Community report prompt p_{com}

You are summarising one community of a personal knowledge hypergraph.
Entities: **{entities}**
Propositions:
{propositions}
Modalities: **{modalities}**
Write a report of at most 200 words, plain prose: (1) THEME, one sentence; (2) KEY ENTITIES, the 3-5 most central and their role; (3) SUMMARY of the facts and relationships; (4) CROSS-MODAL NOTES if several modalities occur. No generic boilerplate.

Figure 13: Community report prompt p_{com} of the community tier (§4.3).

A.5 ANSWER PROMPT p_{ans}

Every method of Table 1 that reads the corpus answers through the same prompt (Figure 14): a system message, then the evidence, the question and the JSON output contract of the task's answer type, so that methods differ only in the evidence they supply. KnowledgeClaw fills the context with $\mathcal{X}_q$, anchor cards first. On phase B of a $\mathcal{T}_5$ item it retrieves again from the edited $\mathcal{K}$, and the question carries the update notice shown in the box. The contract follows the item's answer type: a choice index for the four-way items of $\mathcal{T}_2$ and $\mathcal{T}_5$, the entity names for $\mathcal{T}_1$, the asked value for the joins of $\mathcal{T}_3$ and $\mathcal{T}_4$, and the covered candidates for $\mathcal{T}_6$; the reply is parsed into the prediction that §3 scores. For the video item of Table 2, the evidence that KnowledgeClaw sends is the card of the named video alone, 3,659 characters of catalogue record and timestamped transcript, and Ω chooses option (D), matching the gold answer.

Answer prompt p_{ans}

System: You may use the CONTEXT below (retrieved automatically over the released corpus; it may be irrelevant, incomplete, or -- in the ablation condition -- deliberately missing one modality). Do not assume anything not stated in the context or the question.

User: CONTEXT:
{evidence}

QUESTION: **{question}** *(phase B of $\mathcal{T}_5$:)* QUESTION: (An update to the underlying record may have been filed since any earlier question.) **{question}**

$\mathcal{T}_2$, $\mathcal{T}_5$: Options:
[1] **{option 1}**
[2] **{option 2}**
[3] **{option 3}**
[4] **{option 4}**

Respond with JSON only: {"choice": <1-4>}

$\mathcal{T}_1$: Respond with JSON only: {"entities": ["<entity name>", "<entity name>", ...]}. Each element must be ONLY the entity's own name/surface form, exactly as it appears -- for example a person's name, an organisation, a subject term, or a physical part or feature. Do NOT list the entity TYPES you were asked about (e.g. do not answer with "author" or "subject" themselves), do NOT wrap entities in objects with their own type/modality fields, and do NOT answer "unknown" -- if you cannot find any, respond with an empty list.

$\mathcal{T}_3$, $\mathcal{T}_4$: Answer the question directly. Respond with JSON only: {"answer": ["<value>"]} -- a list containing the specific value(s) the question asks for (usually exactly one, e.g. a single date, name or identifier), not a list of unrelated entities found in the context.

$\mathcal{T}_6$: Candidates:
- **{candidate 1}**
- ...

Respond with JSON only: {"covered": ["<candidate>", ...]}. Each element must be one of the candidate strings above, copied exactly, as a plain string (not wrapped in an object).

Figure 14: Answer prompt p_{ans} of §4.3, shared by every method that reads the corpus: a system message and a user message that ends with the output contract of the item's task.

B THEORETICAL PROOF

The notation is that of §4; $\mathcal{K}_t = (\mathcal{D}_t, \mathcal{V}_t, \mathcal{E}_t, \pi_t)$ is the artefact after the t-th edit, and $\mathcal{K}_0$ is empty. A *path* of length m between nodes x_0 and x_m is an alternating sequence $x_0 \in e_1 \ni x_1 \in e_2 \ni \cdots \in e_m \ni x_m$ of nodes and hyperedges of $\mathcal{E}_t$, and $d_t(x_0, x_m)$ is the length of a shortest one (∞ if none). Three facts of the construction are used throughout: (F1) node identity is ι, a deterministic function of a value, so $\mathcal{V}_t$ holds at most one node per key; (F2) a record write changes $\mathcal{K}$ only as in Eq. (3), creating a node with provenance $\{a\}$ or adding a to an existing provenance; (F3) every edit is one of the three cases of Eq. (5).

B.1 PROOF OF PROPOSITION 1

Proposition 1. *Any two files whose records name a common entity are joined within two hyperedges.*

Proof. We first show that merging is order-free. *Lemma.* After the records of a set A of files are merged, in any order, $\mathcal{V} \supseteq \{\iota(z) : z \in Z_{a,\gamma},\ a \in A,\ \gamma \in \Gamma\}$ and, for each such node v, $\pi(v) \supseteq \{a \in A : \exists \gamma,\ v \in e_{a,\gamma}\}$. Indeed, by (F1) and (F2) the write of a reaches the unique node of key $\iota(z)$: the first write creates it and every later one unions its file into $\pi(v)$. Set union is commutative and associative, so the result does not depend on the order of the writes, nor on whether they come from one ADD or from several.

Now let $a, b \in \mathcal{D}_t$ hold values $z \in Z_{a,\gamma}$ and $z' \in Z_{b,\gamma'}$ with $\iota(z) = \iota(z') = v$. By the lemma the node v exists and the record hyperedges $e_{a,\gamma} \ni a, v$ and $e_{b,\gamma'} \ni v, b$ are both in $\mathcal{E}_t$, whence the path

$$a \ \in \ e_{a,\gamma} \ \ni \ v \ \in \ e_{b,\gamma'} \ \ni \ b, \qquad d_t(a,b) \le 2. \tag{11}$$

If moreover $\mu(a) \neq \mu(b)$, then $a, b \in \pi(v)$ gives $|\mu(\pi(v))| \ge 2$, so the cross-modal hyperedge $e_v^{\times} = \{v\} \cup \pi(v)$ of Eq. (4) contains both files and $d_t(a,b) = 1$. No step depends on the modalities: Eq. (2) maps every modality to a text surface, one record call maps every surface onto the same schema Γ, and ι acts on values alone, so the bound holds for all $|\mathcal{M}|^2 = 25$ modality pairs alike. For person names the walk works in the quotient of $\mathcal{K}_t$ by the alias classes $[v]$; replacing v by $[v]$ in Eq. (11) gives the same bound for two files that write "M. Langtry" and "Max Langtry".

Chains. If $a_0, \ldots, a_m \in \mathcal{D}_t$ and each consecutive pair names a common entity, then concatenating the m paths of Eq. (11) gives $d_t(a_0, a_m) \le 2m$: a join over m records is a walk of at most $2m$ steps over one structure, whatever the modalities along the way.

Stability. The join holds exactly while both records are present. If an edit removes a file $c \notin \{a, b\}$, Eq. (7) only shrinks $\pi(v) \supseteq \{a, b\}$ to a non-empty set, so v is kept, while $e_{a,\gamma}$ and $e_{b,\gamma'}$, whose provenance is $\{a\}$ and $\{b\}$, are untouched; if it removes a, the hyperedge $e_{a,\gamma}$ leaves with it, and so does the path. A store of segments, by contrast, has no node for v: a and b meet only if one similarity search happens to return both, which no structure of that store ensures. □

B.2 PROOF OF PROPOSITION 2

Proposition 2. *After any sequence of edits, all of $\mathcal{K}$ rests on present files, so no evidence is stale.*

Proof. We first show by induction on t that Eq. (6) holds after every edit. For $t = 0$, $\mathcal{K}_0$ is empty and the claim is vacuous; assume it holds for $\mathcal{K}_{t-1}$. By (F3) there are three cases.

Case $\Delta_t = \text{ADD}(A)$. By (F2) a write of $a \in A$ gives each node it creates the provenance $\{a\}$ and adds a to the provenance of each node that exists; a cross-modal hyperedge takes $\pi(e_v^{\times}) = \pi(v)$. Every provenance thus stays non-empty and within $\mathcal{D}_{t-1} \cup A = \mathcal{D}_t$, and index keys are added only for the nodes just written and for the segments of A, which lie in Σ_t.

Case $\Delta_t = \text{DEL}(h)$. Let $R = \eta_{t-1}(h)$, so that $\mathcal{D}_t = \mathcal{D}_{t-1} \setminus R$. Applying Eq. (7) for each $a \in R$ in turn removes, with its index keys, every node whose provenance lies within R, and deletes R from every other provenance, which therefore stays non-empty:

$$\emptyset \neq \pi_t(x) = \pi_{t-1}(x) \setminus R \subseteq \mathcal{D}_{t-1} \setminus R = \mathcal{D}_t \qquad \text{for every surviving node } x. \tag{12}$$

The segments of R leave I_σ with their files. Dropping orphans only removes nodes, and recomputing $e_v^\times$ from the pruned $\pi(v)$ keeps $\pi(e_v^\times) = \pi(v) \subseteq \mathcal{D}_t$. Every node x with $\pi_{t-1}(x) \cap R = \emptyset$ keeps its provenance unchanged, so the edit is local: it touches only the nodes that R supports.

Case $\Delta_t = \textsc{Refresh}(h \to A')$. It is $\textsc{Del}(h)$ followed by $\textsc{Add}(A')$ and preserves the invariant because both do; re-pointing $\eta_t(h) = A' \subseteq \mathcal{D}_t$ changes no provenance, and a later query naming h resolves to A', so it reads the new values and never the old ones. This completes the induction.

Now let $c \notin \mathcal{D}_t$ be a superseded or deleted file. The evidence $\mathcal{X}_q$ of §4.3 is assembled from (i) the cards of files in A_q and of linked files, all in $\mathcal{D}_t$ because η_t maps every handle into $\mathcal{D}_t$ and the walk scores only present files; (ii) hyperedges of $\mathcal{E}_t$ and entities of $\mathcal{V}_t$, whose provenance lies in $\mathcal{D}_t$ by Eq. (6); (iii) passages keyed in I_σ, which belong to files of $\mathcal{D}_t$ by Eq. (6); and (iv) the keyword index and the community summaries, which every edit invalidates and which are rebuilt from $\mathcal{K}_t$. The record and event hyperedges that stated the values of c had provenance $\{c\}$ and were removed with it, and a value of c that survives in $\mathcal{K}_t$ is, up to ι, stated by a present file. Hence c contributes to no part of $\mathcal{X}_q$, for any query q and any sequence $\Delta_1, \ldots, \Delta_t$. □

B.3 Proof of Proposition 3

Proposition 3. *Every file a question names enters its evidence, which top-k similarity cannot ensure.*

Proof. Part 1: KnowledgeClaw. Let q name a handle h and let $a \in \eta_t(h)$. By Eq. (8), $a \in A_q$, hence $A_q \cup V_q \neq \emptyset$ and Eq. (8) routes q to the *file* or the *subgraph* granularity, never to the community tier alone. At either granularity the evidence $\mathcal{X}_q$ opens with the card of every file of A_q; the character budget applies only to the linked works and passages that follow, and no later step removes a card. Hence

$$\Pr\big[\,\mathrm{card}(a) \in \mathcal{X}_q\,\big] = 1 \qquad \text{for every } a \in A_q, \tag{13}$$

whatever k, λ, θ or the size of $\mathcal{D}_t$, and $\mathcal{X}_q$ enters the answer prompt whole. By Proposition 2, $a \in \mathcal{D}_t$, so this card is also current. At the *subgraph* granularity the named file also seeds the walk: for every file b that shares a record entity $u \in \mathcal{V}(a)$ with a, the third sum of Eq. (9) adds $\log(1 + |\mathcal{D}_t|/|\pi(u)|) > 0$ to $s_q(b)$, since $|\pi(u)| \leq |\mathcal{D}_t|$ by Eq. (6). Every file that Proposition 1 joins to a through a non-hub entity is thus ranked by the walk, and the evidence is anchored on the named file before any linked work is added. The guarantee covers a handle that names several files as well: every $a \in \eta_t(h)$, e.g. a PDF and its text twin after a Refresh, enters as its own card.

Part 2: top-k similarity. Let a retriever return the k items of a corpus most similar to $\phi(q)$ under a fixed encoder, and let the corpus hold k files $b_1, \ldots, b_k$ whose surfaces embed in the direction of $\phi(q)$, $\cos(\phi(q), \phi(\tilde{b}_i)) = 1$, beside the named file a with $\cos(\phi(q), \phi(\tilde{a})) < 1$. The retriever returns $b_1, \ldots, b_k$ and misses a. The same happens, without exact ties, whenever k look-alike files lie closer to $\phi(q)$ than a does, as the same-collection candidates of a $\mathcal{T}_2$ item do. More generally, suppose a and m look-alike files are exchangeable to the encoder given q, i.e. the joint law of their scores is invariant under permutation, as it is when a content-hash handle carries no meaning an encoder could use; ties are broken uniformly. By symmetry each of the $m + 1$ files is equally likely to hold each rank, and the top k hold $\min(k, m + 1)$ of them, so

$$\Pr\big[\,a \in \text{top-}k(q)\,\big] = \frac{\min(k,\, m+1)}{m+1} < 1 \qquad \text{whenever } m \geq k. \tag{14}$$

Fusing several similarity lists, as Eq. (10) does, does not lift this bound: each list is a top-$2k$ by similarity, and an item absent from every list has fused score 0. So no choice of k smaller than the corpus lets similarity search ensure that a named file is retrieved, while Eq. (13) holds for every k; KnowledgeClaw therefore uses similarity only to add evidence, never to decide whether a named file is read. Together, Propositions 1–3 give the join, maintenance and lookup guarantees on which C1, C2 and C3 rest. □

C KNOWLEDGECLAW ALGORITHM DETAILS

Algorithm 1 gives the full workflow of KnowledgeClaw (§4) in three phases: **(1) Omnimodal hypergraph construction (C1).** Each file a is read once into a surface $\tilde{a}$ by Eq. (2) and catalogued by one call of Ω; the records merge into $\mathcal{K}$ by Eq. (3), with cross-modal hyperedges and twins by Eq. (4). **(2) Closed-loop lifecycle (C2).** Each edit Δ_t of Eq. (5) is committed in place: DEL retracts provenance by Eq. (7) with no model call, ADD reuses phase 1 on the added files only, and each commit restores the soundness invariant of Eq. (6). **(3) Intent-routed multi-granular retrieval (C3).** A query is anchored and routed by Eq. (8) to a tier; the subgraph tier walks from the anchors by Eq. (9) and fuses two dense paths by Eq. (10); one call of Ω answers.

Algorithm 1 KnowledgeClaw: construct, maintain and query an omnimodal typed hypergraph

Require: Persona files $\mathcal{D}$, edit stream $\Delta_1, \Delta_2, \ldots$, queries q, frozen model Ω, encoder ϕ
Ensure: Hypergraph $\mathcal{K}_t = (\mathcal{D}_t, \mathcal{V}_t, \mathcal{E}_t, \pi_t)$ and answers $\hat{y}_q$

// 1: Omnimodal Hypergraph Construction (C1), ADD($\mathcal{D}$) on an empty $\mathcal{K}$
function ADD(A)
  for each $a \in A$ **in parallel do** ▷ model calls write nothing
    $\tilde{a} \leftarrow \mathrm{proj}(a)$ if $\mu(a) \in \{\textit{text}, \textit{pdf}\}$, else $\Omega(\mathrm{raw}(a); p_{\mu(a)})$; $(\sigma_{a,1}, \ldots, \sigma_{a,n_a}) \leftarrow \mathrm{split}(\tilde{a})$
    Catalogue record: $\big(\{Z_{a,\gamma}\}_{\gamma\in\Gamma},\ \text{events}\big) \leftarrow \Omega(\tilde{a};\ p_{\mathrm{rec}})$
  end for
  for each $a \in A$, each field $\gamma \in \Gamma$ **do** ▷ sequential merge, Eq. (3)
    $e_{a,\gamma} \leftarrow \{a\} \cup \{\iota(z) : z \in Z_{a,\gamma}\}$; $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_{a,\gamma}\}$; $\pi(e_{a,\gamma}) \leftarrow \{a\}$
    $\pi(v) \leftarrow \pi(v) \cup \{a\}$ for $v \in e_{a,\gamma} \cap \mathcal{V}$; add alias key $\iota_{\mathrm{p}}(v)$ to class $[v]$ for person names
  end for
  Add a dated *event* hyperedge $\{a\}$ per event; add $e_v^{\times}$ and twins $a \approx b$ by Eq. (4)
  Insert $\phi(v), \phi(e), \phi(\sigma_{a,j})$ into $I_{\mathcal{V}}, I_{\mathcal{E}}, I_{\sigma}$ and $\tilde{a}$ into BM25
end function
// 2: Closed-loop Lifecycle (C2), for each edit Δ_t of Eq. (5)
function DEL(h) ▷ no model call
  for each $a \in \eta(h)$, each $x \in \mathcal{V} \cup \mathcal{E}$ with $a \in \pi(x)$ **do**
    if $\pi(x) = \{a\}$ **then** delete x and its index keys **else** $\pi(x) \leftarrow \pi(x) \setminus \{a\}$ ▷ Eq. (7)
  end for
  Drop orphans; recompute touched $e_v^{\times}$; $\mathcal{D} \leftarrow \mathcal{D} \setminus \eta(h)$
end function
function REFRESH($h \to A'$): DEL(h); ADD(A'); $\eta(h) \leftarrow A'$
After each commit: assert Eq. (6); mark derived indices for rebuild on next use
// 3: Intent-routed Multi-granular Retrieval (C3), for each query q
$A_q \leftarrow \bigcup_{h\in\mathrm{hdl}(q)} \eta(h)$; $V_q \leftarrow \{\iota(\psi)$ or nearest in $I_{\mathcal{V}}$ at $\cos \ge \theta : \psi \in \Psi_q\}$; $g \leftarrow g(q)$ by Eq. (8)
if $g =$ *file* **then**
  $\mathcal{X}_q \leftarrow$ full cards of the files in A_q
else if $g =$ *subgraph* **then**
  Score each $b \notin A_q$ (nor its twin) by $s_q(b)$ of Eq. (9), skipping hubs; merge twins into works ω
  $W_q \leftarrow$ works with $S_q(\omega) \ge 0.6 \max_{\omega'} S_q(\omega')$, at most six
  $L_q^{\mathcal{V}} \leftarrow$ hyperedges of the top-$2k$ entities of $I_{\mathcal{V}}$; $L_q^{\mathcal{E}} \leftarrow$ top-$2k$ hyperedges of $I_{\mathcal{E}}$
  $\mathcal{X}_q \leftarrow$ cards of $A_q \parallel$ cards of $W_q \parallel$ top-k hyperedges by rrf_q (Eq. 10) $\parallel$ four passages of I_{σ}
else
  $\mathcal{X}_q \leftarrow$ the three nearest reports of the 64 largest Louvain communities, each $\Omega(\cdot; p_{\mathrm{com}})$
end if
return $\hat{y}_q = \Omega(q, \mathcal{X}_q;\ p_{\mathrm{ans}})$

Complexity analysis. Let $N = |\mathcal{D}|$, n_{m} the number of media files, $R = \sum_a |\mathcal{V}(a)|$ the total record size and d the embedding dimension. First, construction issues exactly $N + n_{\mathrm{m}}$ calls of Ω on bounded inputs (Table 3), i.e. $O(N)$ parallel calls, plus $O(R)$ to merge and $O\big((|\mathcal{V}| + |\mathcal{E}| + |\Sigma|)\, d\big)$ to index; no call sees two files together, so no step is quadratic in N. Second, an edit costs only what it touches: DEL(h) visits the nodes whose provenance holds a file of $\eta(h)$, $O(\sum_{a\in\eta(h)} |\mathcal{V}(a)|)$ with no model call, and ADD(A) costs $O(|A|)$ calls, independent of $|\mathcal{D}_t|$, so T edits avoid the $O(TN)$ calls of rebuilding. Third, a query costs at most two calls of Ω (routing only if no rule applies, and the answer); the file tier is $O(|A_q|)$, the walk visits only files linked to an anchor, $O\big(\sum_{v\in V_q\cup\mathcal{V}(A_q)} |\pi([v])|\big)$, and the two dense paths cost $O\big((|\mathcal{V}| + |\mathcal{E}|)\, d\big)$. The evidence is bounded (six works, k hyperedges, four passages), so a query costs \$0.0004 and a 1,024-token budget keeps 55.8 of the full 55.9 OKE (§5.7).

D OMNIKE-BENCH DETAIL

This appendix expands §3. Persona d has corpus $\mathcal{D}_d$; a *unit* u is one released work (a document as PDF and text, an image, or a clip), a *record* r one of its files with modality $m(r) \in \mathcal{M}$, and $\mathcal{K}_d = \mathcal{S}(\mathcal{D}_d)$ the artefact a system $\mathcal{S}$ commits at INGEST.

Table 4: **The six OmniKE tasks.** All gold comes from human-made metadata or description, deterministic parsing or official transcripts, never from a model. $M(t)$: evidence modalities (items).

	Query → answer	Gold source	Metric	$M(t)$
Construction				
$\mathcal{T}_1$	PDF or image + closed type inventory → its entities of those types	catalogued authors, affiliations, subject terms, depicted persons; parsed floor-plan elements	set F1	pdf 344, image 56
$\mathcal{T}_2$	media file → its authoritative description among 4 candidates	catalogue record, creator description or transcript; all four candidates from one collection, each of a different subject	accuracy	image 397, audio/video 3
Application				
$\mathcal{T}_3$	2–4-hop question → one catalogued value	deterministic join over catalogue records; each join condition is one hop	set F1	image 275, pdf 125
$\mathcal{T}_4$	as $\mathcal{T}_3$, but the chain joins image and PDF records (2–3 hops)	as above	set F1	image+pdf 110
$\mathcal{T}_6$	PDF or image + candidate labels → those assigned to it	human-assigned controlled-vocabulary labels; negatives never co-assigned with its labels	balanced acc.	image 304, pdf 78
Maintenance				
$\mathcal{T}_5$	a document's title, authors or abstract; answer, commit Δ, answer again	official version chain; the corpus holds the prior version, Δ the next	accuracy, both phases	pdf 137

D.1 TASK DESIGN

Canonical contracts. Adapters lift each source onto nine contracts: REC (a record with its unit, modality and parent), ORD (order by date, version or recording time), MEN (an entity of a catalogued class), ALN (a cross-modal alignment), IDN (an agent identity with its identifier basis), EVT (a dated event), STA (a state value in one version), NRM (a validity norm and its deviations) and COV (an assigned controlled-vocabulary term). Every row carries its provenance, and builders read contracts only, so each task is one builder that runs unchanged over all ten personas.

Instantiation. $\mathcal{T}_1$–$\mathcal{T}_3$ run in every persona, the other tasks wherever their structure yields at least 20 items (Table 5), never padded with model-made items: $\mathcal{T}_4$ needs two modalities sharing authority-controlled entities (in healthcare, NLM subject headings of CDC images and PubMed Central papers), $\mathcal{T}_5$ an official version chain and $\mathcal{T}_6$ a controlled vocabulary with enough balanced labels.

$\mathcal{T}_1$ **(knowledge acquisition).** Given a unit u and a closed type inventory $\Gamma = \Gamma_{d,m(u)}$, return the entities of those types in u, $\mathcal{T}_1 : (u, \Gamma) \mapsto \hat{E}$, with gold $E^*(u) = \{\mathrm{surf}(x) : x \in \mathrm{MEN}(u),\ \mathrm{cls}(x) \in \Gamma,\ \mathrm{own}(x, u)\}$. The rule $\mathrm{own}(x, u)$ keeps only what the unit itself states: a whole-word surface in its own human text or aligned transcript, or, for an image, a row from a human structural layer or a class naming what is depicted. $\Gamma_{d,m}$ holds every class with such a row in (d, m), never tailored to an item; items ask for 2.4 types and hold 6.0 gold entities.

Table 5: **Items per persona.** Each persona has 90 documents (PDF and text), 180 images, 18 audio and 18 video clips (2.9, 2.4 min on average); 15.8 h of media, 6.9 GB in all.

	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	**All**
HC	40	40	41	110		166	397
FIN	41	40	41		29		151
EDU	40	40	41		22	46	189
LAW	36	40	41				117
PSY	40	40	40			170	290
POL	40	40	41				121
HUM	41	40	41				122
BLD	40	40	32		29		141
MUS	41	40	41		29		151
GAM	41	40	41		28		150
Total	400	400	400	110	137	382	1,829

$\mathcal{T}_2$ **(cross-modal coreference).** Given a media record r, choose among four candidates the description joined to r by an ALN row, $\mathcal{T}_2 : (r, \{c_1, \ldots, c_4\}) \mapsto \hat{\imath}$. All candidates come from the same parent $p(r)$ (transcript, explicit parent, partition key or collection), so distractors are of the same kind and place; descriptions have at least 40 characters, and no two candidates share a subject group.

$\mathcal{T}_3$ **(multi-hop retrieval).** Given q encoding a join path $\pi = r_0 \xrightarrow{j_1} \cdots \xrightarrow{j_h} v$, $h \in \{2, 3, 4\}$, return the terminal value, $\mathcal{T}_3 : (q, \mathcal{K}_d) \mapsto \hat{a}$. Each hop is one contract join (an ALN step, the one other record sharing a field, a lookup from a unique entity, or a terminal step to one value). A path is admissible only if every record-valued hop yields exactly one unvisited record, the terminal hop

Table 6: **Sources per persona.** PD: public domain (US federal work); WC: Wikimedia Commons, pages created before 30 Nov 2022, each file keeping its licence (PD, CC0, CC BY or CC BY-SA); OCW: MIT OpenCourseWare. NC: CC BY-NC-SA, inherited per file by the release. No source carries a no-derivatives term.

Persona	text / pdf	image	audio	video
Healthcare	PubMed Central OA (CC BY)	CDC Public Health Image Library (PD)	PriMock57 (CC BY); Senate HELP hearings (PD)	Senate HELP hearings (PD)
Finance	arXiv q-fin	WC	FOMC press conferences (PD); Earnings-21/22 (CC BY-SA)	FOMC press conferences with official transcripts (PD)
Education	arXiv	Smithsonian Open Access (CC0)	OCW^{NC}	OCW^{NC}
Law	SCOTUS opinions (PD); DOAJ CC-BY journals	WC	SCOTUS oral arguments, official transcripts (PD)	Senate Judiciary hearings (PD)
Psychology	PsyArXiv, PMC	Wellcome Collection (PD/CC0/CC BY/BY-SA)	OCW 9.00^{NC}	OCW 9.00, captioned lectures^{NC}
Politics	CRS reports with version chains (PD); World Bank OKR (CC BY)	WC	VoxPopuli, European Parliament (CC0)	White House briefings, official captions (PD)
Humanities	DOAJ CC-BY humanities journals	Met and Cleveland Museum Open Access with curatorial records (CC0)	Open Yale Courses^{NC}; LibriVox with Gutenberg text (PD)	Open Yale Courses^{NC}
Building	arXiv; building codes (CC0); NIST (PD)	CubiCasa5K plans^{NC}; CMP facades (CC BY-SA)	OCW planning^{NC}	OCW planning; site footage^{NC}
Music	arXiv	Met musical instruments (CC0)	OpenMIC-2018 labels (CC BY); FMA, no NC/ND	OCW 21M, captioned lectures^{NC}
Games	arXiv	WC screenshots; OpenGameArt (CC0)	OpenGameArt, creator descriptions (CC0)	WC, captioned game videos (CC BY)

yields one value and no earlier record already gives the answer, so every hop is needed. The 400 items have 300, 72 and 28 paths of 2, 3 and 4 hops. $\mathcal{T}_4$ **(cross-modal reasoning)** is the same over paths through at least two modalities, one of them non-textual; its 110 items, disjoint from $\mathcal{T}_3$, join image and PDF records through NLM subject headings.

$\mathcal{T}_5$ **(lifecycle consistency).** Over an official version chain, the system answers q about a state attribute of a document (phase A, gold v_{old}), commits the released next version $\Delta = (\textsc{Refresh}, A_\Delta)$ with $A_\Delta \cap \mathcal{D}_d = \emptyset$, and answers again (phase B, gold $v_{\text{new}} \neq v_{\text{old}}$). Attributes are titles (49 items), abstracts (69) and author lists (19); pairs differing only in spelling are dropped, and no stem or option carries a version label, date or word that orders states in time, so only the committed edit separates the phases.

$\mathcal{T}_6$ **(knowledge discovery).** Given a unit u and candidates $C = C^+ \cup C^-$ from one vocabulary, return the covered ones, $\mathcal{T}_6 : (u, C) \mapsto \hat{C}^+ \subseteq C$. C^+ holds the unit's human-assigned terms and C^- terms never co-assigned with them, with $|C^+| = |C^-| \geq 2$; vocabularies are MeSH (171 items), LCSH (127), Smithsonian object types (46), LC Names (24) and three others (14).

D.2 Benchmark Construction

The four phases of Figure 3C run as seven stages that order candidates by SHA-256, cache every model call and discard items only by score-blind rules.

(1) Collect. Sources (Table 6) obey three rules. Annotation must be human-produced: ASR, OCR and model descriptions may be system input but never gold, which rejected three video archives with ASR captions. Media must be redistributable, and crowd-written descriptions count as gold only on pages created before November 30, 2022.

(2) Normalise and lift. Every file is re-encoded, renamed by content hash and stripped of metadata; images come with their own catalogue record, never cut from a paper. Adapters map each annotation element to one contract row.

(3) Instantiate, rewrite and verify. A template filled from contract rows yields a stem, its slots and its answer. GPT-5-nano paraphrases the stem without seeing answer or options; six deterministic checks then require slots, numbers and dates kept verbatim, no item's answer in the stem and no task-declared pattern, e.g. a version label in $\mathcal{T}_5$. A failing item is discarded, never repaired.

(4) Balance and (5) recompute. No answer value exceeds 10% of a task or of a persona's share of it, $\mathcal{T}_2$ and $\mathcal{T}_5$ cycle the gold position, and frequent $\mathcal{T}_6$ labels are positive in a third to two thirds of their uses. A second implementation, sharing only the written task definitions, recomputes every answer from the raw sources and agrees on all $1{,}829$ items.

(6) Gate and (7) freeze. Gates confirm that no released file keeps an original name, source URL or gold sentence; collecting images with their own record took a redaction-box probe on $\mathcal{T}_2$ from 77.3 to chance (25.0–27.5). Each persona is then cut to the size of Table 5, keeping every unit an item touches, which changes no gold answer; a SHA-256 manifest seals the release, and one command rebuilds all items from the raw sources.

D.3 Evaluation Metric Design

Protocol. The harness builds $\mathcal{K}_d$ once per persona and answers every task from it, each item alone in a fresh context, with a value list ($\mathcal{T}_1, \mathcal{T}_3, \mathcal{T}_4$), an option ($\mathcal{T}_2$, $\mathcal{T}_5$ per phase) or a candidate subset ($\mathcal{T}_6$); a missing answer scores 0. File reads are not sandboxed, since MMGraphRAG's official pipeline reopens images at query time.

Lenient value matching. A predicted value p and a gold value g have similarity $\sigma(p, g)$, where $\mathrm{nrm}(\cdot)$ is the dataset normalisation (lower-case, trim punctuation, drop a leading article, collapse spaces), $\mathrm{tok}(\cdot)$ the lower-cased word-token set and $\gamma(g) \in \{\text{year}, \text{month}, \text{day}\}$ the precision of a gold date:

$$\sigma(p,g) = \begin{cases} \mathbb{1}\big[\mathrm{date}(p) \equiv_{\gamma(g)} g\big] & g \text{ a date of granularity } \gamma(g), \\ 1 & \mathrm{nrm}(p) = \mathrm{nrm}(g), \\ 2|\mathrm{tok}(p) \cap \mathrm{tok}(g)|/(|\mathrm{tok}(p)| + |\mathrm{tok}(g)|) & \text{otherwise}, \end{cases} \tag{15}$$

so "May 24, 2022" matches "2022-05-24" and a short affiliation its full catalogued form. Pairs with $\sigma \geq \tau = 0.5$ are matched one-to-one, greedily by decreasing σ, giving the match set $\mathcal{M}(\hat{A}, A^*)$.

Item and task scores. With $\hat{A}$, A^* the predicted and gold values and $\hat{C}^+$ the selected candidates, $s_2 = \mathbb{1}[\hat{\imath} = g]$ for gold option g, $s_5 = \frac{1}{2}(\mathbb{1}[\hat{a}_{\mathrm{A}} = g_{\mathrm{A}}] + \mathbb{1}[\hat{a}_{\mathrm{B}} = g_{\mathrm{B}}])$, and

$$s_{1,3,4} = \frac{2|\mathcal{M}(\hat{A}, A^*)|}{|\hat{A}| + |A^*|}, \qquad s_6 = \frac{|\hat{C}^+ \cap C^+|}{2|C^+|} + \frac{|C^- \setminus \hat{C}^+|}{2|C^-|}. \tag{16}$$

On $\mathcal{T}_5$ always returning the old value, or always the new one, scores exactly 50, as does selecting all or none of the candidates on $\mathcal{T}_6$, so only a committed edit and real discrimination rise above half credit. As $s \in [0, 1]$ has variance at most $1/4$, the worst-case 95% half-width of a mean of n items is $98/\sqrt{n}$: ± 4.9 for $\mathcal{T}_1$–$\mathcal{T}_3$, ± 5.0 for $\mathcal{T}_6$, ± 8.4 for $\mathcal{T}_5$ and ± 9.3 for $\mathcal{T}_4$; per-persona scores are descriptive.

Admission references. Table 7 gives the mechanical floor f (best of position, length, overlap and prior heuristics), the zero-model shortcut probe P, closed-book C, flat retrieval R and R_{-m}. C is 0.0 on the three free-form tasks and below the floor on $\mathcal{T}_2$ and $\mathcal{T}_5$, so neither the stem nor memorised knowledge gives the answer; R reaches only 2.5 and 0.0 on the join tasks $\mathcal{T}_3$ and $\mathcal{T}_4$, so retrieval alone does not compose records; and R_{-m} falls below R wherever R scores, by up to 56.0 points on $\mathcal{T}_2$, so the modality an item names is needed to answer it. Neither P nor C exceeds R on any task, so no shortcut beats reading the corpus.

Table 7: **Reference scores.** R reads the item's media bridge and three BM25 passages; R_{-m} withholds the item's modality (C, R: best of two models, before the cut).

	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$
Items n	400	400	400	110	137	382
Floor f	12.1	33.5	2.0	0.0	50.0	50.0
Probe P	0.1	26.6	0.3	0.0	25.4	49.2
Closed-book C	0.0	32.3	0.0	0.0	35.8	58.4
Retrieval R	5.3	82.3	2.5	0.0	64.2	69.9
R_{-m}	0.0	26.3	0.0	0.0	32.8	50.3

E BASELINE DETAIL

This appendix specifies how every row of Table 1 is deployed: the shared protocol (§E.1), the 30 baselines in the order and grouping of the table (§E.2–§E.5), and KnowledgeClaw with its three ablations (§E.6). For each method we state what INGEST builds, what a query retrieves and how the $\mathcal{T}_5$ edit Δ is committed; every number is the value used in our runs, and a token is 4 characters throughout.

E.1 SHARED DEPLOYMENT PROTOCOL

One frozen stack. Every method calls the same model Ω (GEMINI 2.5 FLASH LITE) and encoder ϕ (GEMINI EMBEDDING 2, 3,072 dimensions) through one metering proxy that caches each call by the SHA-256 of model, messages and parameters, at temperature 0 unless a method's official default differs (MemPro, 0.3). **One input.** A text or PDF file enters as its opening 4,000 tokens, cut at a paragraph break and headed by its file name, the only handle items use. Text-reading methods receive image, audio and video through the shared media-to-text bridge (image captions, time-windowed transcripts); the omnimodal group and KnowledgeClaw read raw media natively. **One answerer.** Whatever a method retrieves, or gathers with its own agent loop, ends in the answer prompt of Appendix A.5 with 600 output tokens, so methods differ only in the evidence they supply. **One protocol.** Per persona a method builds once over the base corpus and answers each item in a fresh context at 24-way concurrency. On $\mathcal{T}_5$ all phase-A answers come first, then every item's Δ (the next released version, headed "a new version of h") is committed through the method's own write path, then all phase-B answers, so no phase-A answer can see an edit. Every method attempts all six tasks. **Faithful re-implementation.** Each baseline is re-implemented on this stack from its official code (from the paper where none is released: OntoKG, M^{3}KG-RAG), with its prompts copied verbatim and its defaults kept; in-process indices replace external databases, the frozen stack replaces each method's own models (NV-Embed, MiniLM, CLIP, Whisper, Qwen), and gleaning is 0 wherever offered. Other deviations are listed per method; spend and time are in Table 8.

E.2 NO KNOWLEDGE ENGINEERING, CHUNK, GRAPH AND AGENTIC GRAPH RAG

NaiveGeneration (Google, 2025). No ingest, retrieval or write: Ω answers from the question and options alone, the closed-book control.

StandardRAG (Lewis et al., 2020). *Ingest:* 1,200-token windows with 100 overlap, embedded by ϕ. *Query:* cosine top-5 windows. *Edit:* the windows of Δ are appended.

RAPTOR (Sarthi et al., 2024). *Ingest:* sentence-bounded leaves of $\leq$100 tokens; up to 5 layers of soft GMM clustering ($k \leq 50$ by BIC, membership $p > 0.1$, PCA to 10 dimensions for UMAP, clusters over 3,500 tokens re-clustered), each cluster summarised in $\leq$100 tokens. *Query:* collapsed-tree top-10 over all layers within 3,500 tokens. *Edit:* the tree is rebuilt over the base corpus and Δ.

LightRAG (Guo et al., 2025b) (v1.4.14). *Ingest:* 1,200/100-token chunks; one call per chunk extracts entities of 11 types and keyworded relations, merged by name and pair with map-reduce description summaries. *Query:* hybrid mode; a keyword call sends low-level keys to the entity store and high-level keys to the relation store (top-40 each, cosine > 0.2), plus 20 related chunks under 6,000/8,000/30,000-token entity, relation and total budgets. *Edit:* native incremental insert.

GraphRAG (Edge et al., 2025). *Ingest:* 1,200/100-token units; one call per unit extracts entities of four types and weighted relations, descriptions are summarised in $\leq$500 words, recursive Louvain (communities above 10 members split again) stands in for hierarchical Leiden, and each community receives a rated report. *Query:* local search from the top-20 entities within 12,000 tokens, split 15/35/50% over reports, entity–relation tables and text units. *Edit:* the official update mode extracts Δ, merges by title and pair, and re-summarises.

HippoRAG 2 (Gutiérrez et al., 2025). *Ingest:* one passage per file; OpenIE by an NER call and an NER-conditioned triple call; phrase, fact and passage nodes with synonymy edges at cosine $\geq$ 0.8. *Query:* the top-5 facts pass the recognition-memory filter (10 demonstrations) and seed Personalized PageRank (damping 0.5, passage weight 0.05); the top-5 passages are returned. *Edit:* incremental indexing, synonymy edges recomputed.

KAG/OpenSPG (Liang et al., 2025). *Ingest:* 1,200/100-token chunks; three schema-free calls per chunk (NER, standardisation, triples) into an entity–chunk graph with a PageRank matrix. *Query:* its solver plans a logical form and runs $\leq$3 retrieval steps (exact linking at 0.9, else fuzzy linking at 0.8 with PPR chunks, RRF-fused with vector chunks at 0.65), summarising each step, $\leq$5 calls. *Edit:* incremental extraction, PPR matrix rebuilt.

Youtu-GraphRAG (Dong et al., 2025). *Ingest:* 1,200/100-token chunks (official 5,000/200), one schema-guided call each, into a four-level tree of attributes, triples, keywords and $\leq$100 FastTreeComm communities. *Query:* the agent splits q into $\leq$4 sub-questions, retrieves triples by node, relation and 3-hop paths (top-20), and runs IRCoT for $\leq$2 steps (official 5). *Edit:* full rebuild.

E.3 KNOWLEDGE GRAPH CONSTRUCTION AND n-ARY / HYPERGRAPH KE

iText2KG, OntoKG, Hyper-KGGen and ATOM construct a graph but define no query path; their facts, each tagged with its source file, are served by one shared layer, cosine top-30 over ϕ.

iText2KG (Lairgi et al., 2025). *Ingest:* 1,200-token chunks are distilled into facts, one section per file; entity and relation calls follow, entities resolve on exact (name, label) match or cosine > 0.92 (0.6 name $+ 0.4$ label), relation names at 0.9, and one round revisits isolated entities (official 3); the official $0.8/0.7$ thresholds collapsed relations onto $\sim$30 names and were recalibrated. *Edit:* Δ is ingested onto the graph.

OntoKG (Li et al., 2026c). *Ingest:* one call designs a modular schema over 8 seed categories from 24 sampled files, one refinement call repairs the types and properties a probe extraction leaves unrouted, and 600-token chunks are extracted with the paper's Listing 2 and routed to modules by exact or cosine-≥ 0.9 match. *Edit:* Δ is extracted and routed into the frozen schema.

HyperGraphRAG (Luo et al., 2025). *Ingest:* $1{,}200/100$-token chunks; one call per chunk yields n-ary knowledge segments with scored entities, stored as a bipartite hyperedge–entity graph. *Query:* entity and segment keys parsed from q drive a local (top-60 entities) and a global (top-60 hyperedges) path at cosine > 0.2, returning entities, hyperedges and source chunks, 4,000 tokens each. *Edit:* incremental insert.

Hyper-KGGen (Huang et al., 2026). *Ingest:* the official recursive splitter (1,000 characters, 200 overlap); per chunk a mention call and a hyperedge call over the known nodes; nodes merge by name and hyperedges by member set, and the skill library is empty as none is released. *Edit:* Δ runs the same pipeline and merges in.

HyperRAG (Lien et al., 2026). *Ingest:* the HyperGraphRAG hypergraph, built with its own calls. *Query:* topic entities of q seed an LLM-scored beam search over hyperedges and entities (width 3, depth 3, $\leq$100 candidates per entity) with a sufficiency check per depth; paths and their chunks enter within 12,000 tokens. *Edit:* incremental insert.

E.4 MEMORY AGENTS AND SELF-EVOLVING LIFECYCLE

MemGPT/Letta (Packer et al., 2024). *Ingest:* $512/50$-token passages into archival memory beside the core `persona` and `human` blocks. *Query:* a tool-calling loop under the verbatim `letta_v1` system prompt searches archival memory (top-5, $\leq$20) and edits core memory for $\leq$5 steps. *Edit:* passages are appended.

A-MEM (Xu et al., 2025). *Ingest:* one note per $1{,}200/100$-token chunk, analysed into keywords, context and tags; each new note runs the evolution call over its 5 nearest notes, strengthening links or updating neighbours. *Query:* top-5 notes and up to 5 linked notes. *Edit:* Δ is added and evolved.

Mem0 (Chhikara et al., 2025). *Ingest:* a fact-extraction call per chunk, then one update call over each fact's top-5 memories decides ADD, UPDATE, DELETE or NONE. *Query:* top-100 memories with their sources. *Edit:* the same path, so Δ can update or delete memories.

Graphiti (Zep) (Rasmussen et al., 2025). *Ingest:* one episode per chunk at a reference time; entity extraction, resolution (exact name, 3-gram Jaccard ≥ 0.9, else an LLM dedupe over 15 candidates), fact extraction with validity times, and edge resolution that invalidates contradicted facts. *Query:* RRF of BM25 and cosine over facts and entities (top-10 each) and BM25 episodes, with validity dates. *Edit:* Δ is ingested at a later time and invalidates what it contradicts. To fit the shared budget, a fact call has 3,072 output tokens and 15 facts.

MIRIX (Wang & Chen, 2025). *Ingest:* a meta-manager routes each chunk to the core, episodic, semantic, resource and knowledge-vault managers, each writing its store by a JSON function call against its BM25 top-10; procedural memory stays empty, as released. *Query:* a topic call, then core memory, the 10 latest episodes and BM25 top-10 per store. *Edit:* the managers insert, merge or replace items.

MemOS (Li et al., 2025). *Ingest:* the structured document reader turns each chunk into key–value–tag memories, with a 20-item working memory. *Query:* a task-parse call yields keys, tags and 2–5 expansions; graph recall (key or $\geq$2 tags) and vector recall are reranked by cosine to the top-20. *Edit:* new memories are added.

MemEvolve (Zhang et al., 2025). It evolves agent memory and has no document ingest, so each chunk enters its fusion-memory provider as a successful trajectory "memorise ⟨file⟩". *Ingest:* a summarise call writes insight, pattern, playbook and checklist nodes, linked at similarity ≥ 0.75 and consolidated every 50 ingests. *Query:* a focus call, hybrid search (0.2 TF-IDF $+ 0.8$ cosine, top-5, 3-neighbour expansion) and a synthesis call give a $\leq$350-character note. *Edit:* Δ is written through the same path.

ATOM (Lairgi et al., 2026). *Ingest:* 380-token chunks stamped with the date each file states; atomic facts, then temporal 5-tuples $(s, p, o, t_{\text{start}}, t_{\text{end}})$, merged with entities at cosine ≥ 0.9 and served by the shared fact layer of §E.3. *Edit:* Δ is observed later; a relation its old file supported that the new file does not re-observe loses that source, and is closed at t_{end} once none remains.

MemPro (Liu et al., 2026d). The released evolved program is used without re-evolution. *Ingest:* 2,000-token pages, each abstracted by its memory agent, into dense and BM25 indices. *Query:* planning over all abstracts, keyword, vector and page search (top-5), focused excerpts of $\leq$3 pages, integration and reflection, for one research iteration (official 3) to fit the shared budget. *Edit:* new pages are abstracted against the memory and appended.

E.5 AGENTIC KNOWLEDGE ENGINEERING AND OMNIMODAL KNOWLEDGE ENGINEERING

OpenClaw, LLM Wiki and OKF build markdown knowledge bases for an agent to read, so they share one reader: hybrid search (0.7 cosine + 0.3 BM25, top-6 over 1,500-character windows), open and list tools, $\leq$5 steps and $\leq$24,000 characters gathered.

OpenClaw (OpenClaw contributors, 2026). *Ingest:* the corpus is split into 32 sessions; each file is one agent turn under the verbatim `AGENTS.md` memory rules, appending to the session's daily note, with the pre-compaction flush beyond 32,000 tokens. *Edit:* Δ runs as a new session that writes a new note.

Graphify (Graphify contributors, 2026). *Ingest:* the verbatim skill prompt extracts nodes and edges from batches of 22 files (one per image); Louvain (seed 42) is its fallback for Leiden. *Query:* term hits in label and source pick 5 start nodes, expanded by BFS to depth 2 within 2,000 tokens. *Edit:* `--update` on unseen files.

Cognee (Markovic et al., 2025). *Ingest:* `cognify`; per file a graph-extraction and a summary call into document, chunk, summary, entity and type nodes. *Query:* graph completion; top-100 per collection, unmatched elements penalised 6.5, and the 5 best triplets by summed distance. *Edit:* `cognify` on Δ.

LLM Wiki (Karpathy, 2026). The idea file gives no prompts, so our schema instantiates it. *Ingest:* in waves of 16, each source gets a page and a plan over its 12 nearest entity and concept pages ($\leq$5 updates, $\leq$3 new), then one merge rewrite per touched page. *Edit:* a new version plans against its old source page first.

OKF (Google Cloud, 2026). *Ingest:* one concept document per file, written by the reference instruction with the concept list in view, plus per-directory indices and a log. *Edit:* a new version refines its base concept in place, with the existing document as input and the update logged.

Omni-SimpleMem (Liu et al., 2026a). *Ingest:* multimodal atomic units from raw media: 500/50-token text chunks, redundancy-filtered and summarised; two captions per image; audio transcripts; video at 1 fps through the HSV scene trigger, $\leq$24 captioned frames and its audio. *Query:* dense top-10 and BM25 top-10. *Edit:* Δ adds units.

MMGraphRAG (Wan & Yu, 2026). *Ingest:* a text graph (1,200/100-token chunks, transcripts included) fused with an image graph built from the raw images (description, image entities, enhancement, whole-image matching by spectral clustering), one graph per persona. *Query:* top-5 nodes at 0.2 with their tables, plus a vision answer over each of $\leq$3 retrieved images. *Edit:* rebuild over the base corpus and Δ.

M^{3}KG-RAG (Park et al., 2026a). *Ingest:* 2,000-token text items and a raw-media caption per image, audio and video file; triples by the extractor and normaliser, descriptions by the searcher–inspector (pass $\geq$ 7, $\leq$3 rounds). *Query:* items within $L_2 \leq 1.1$ of the top-5 plus named files, 2-hop expansion scored by $\cos(h, q) + \cos(t, q)$, and GRASP pruning. *Edit:* Δ adds items and triplets.

E.6 KNOWLEDGECLAW AND ITS ABLATIONS

KnowledgeClaw. *Ingest (C1):* text and PDF from the shared projection; image ($\leq$1,536 px), audio (16 kHz) and video ($\leq$6 keyframes, one per 20 s, with audio) read natively by Ω (Appendix A.1), all 2,160 media files of the benchmark; surfaces split into $\sim$1,000-token segments; one record call per file over its first 9,000 characters fills 13 typed fields, title and events; record, event and cross-modal hyperedges and twins by Eqs. (3)–(4); indices $I_{\mathcal{V}}$, $I_{\mathcal{E}}$, I_{σ} and BM25. *Query (C3):* anchors at $\theta = 0.86$ and routing by Eq. (8); 7,000-character anchor cards; the walk of Eq. (9) keeps works within 0.6 of the best, at most six; RRF with $\lambda = 4$, $k = 10$, and four passages (Eq. 10), within 22,000 characters; 64 Louvain communities (seed 0) reported in $\leq$200 words; one retrieval pass and one answer call. *Edit (C2):* a new version of a live handle h commits as $\textsc{Refresh}(h \to A')$: $\textsc{Del}$ with provenance pruning, $\textsc{Add}(A')$ and re-pointing h (Eqs. 5–7), each commit audited and checked against Eq. (6).

w/o C1 (no hypergraph). Media arrive through the shared bridge instead of native reading and no record call is made, so only the segments of the text projection are indexed (I_{σ}), with no entity, hyperedge or community. C2 and C3 run unchanged on this store: file handles still resolve to their cards and $\textsc{Refresh}$ still retracts the old version.

w/o C2 (no lifecycle). C1 and C3 are unchanged; every Δ is appended as new files, with no $\textsc{Del}$ of the old version and no re-pointing of h, so the handle keeps resolving to the version it named.

w/o C3 (flat retrieval). C1 and C2 are unchanged; a query skips anchoring, routing, cards, the walk and the communities, and its evidence is the RRF top-10 hyperedges of Eq. (10) over the top-20 of $I_{\mathcal{V}}$ and $I_{\mathcal{E}}$, plus the top-6 passages of I_{σ}.

All three ablations share KnowledgeClaw's call cache, so each differs from the full model only in the component it removes.

Settings in the analyses. The values above are the defaults of every KnowledgeClaw row. The sweeps of §5.7 (Figure 9) vary θ, the keep ratio, the cap on linked works and k around them, and the forced granularities of §5.6 (Figure 8c) fix the tier of Eq. (8) instead of routing; every other setting stays as listed, and none is tuned per persona or per task. On this stack KnowledgeClaw spends \$0.32 per persona on ingest and \$0.0004 per query, and builds a persona in a median of 3 minutes of wall-clock time at 24-way concurrency (Table 8).

F EXPERIMENT DETAIL

F.1 PROTOCOL AND COST

Saved outputs. Every number of §5.2–§5.8 is recomputed from the main run: the per-item predictions of all rows of Table 1, scored as in §3, and the traces, audit logs and hypergraphs saved by KnowledgeClaw and its ablations. A difference quoted in the text is taken between rounded entries.

Analyses with new answers. For the evidence budget (Figure 9b), each item's saved evidence is cut to its first B tokens at a line break and answered again; the full-length point is the main-table prediction. For the forced granularities and the sweep (Figures 8c, 9c), KnowledgeClaw is rebuilt from the main run's call cache and every query is re-retrieved on the main protocol; an item whose evidence is unchanged keeps its prediction, so the default reproduces Table 1 exactly. The edit stream (Figure 7c) applies 200 random DEL and re-ADD edits per persona after the $\mathcal{T}_5$ commits and checks Eq. (6) after each. The three analyses together spent only \$2.6 of API calls.

Cost. KnowledgeClaw builds a persona in 3 minutes, beaten only by Graphify (2, OKE 17.8); its ingest cost of \$0.32 is undercut only by methods with OKE ≤ 21.0, and its query cost of \$0.0004, $27.8\times$ below MemPro's, only by five methods with OKE ≤ 17.8 (Table 8).

Table 8: **Cost. ingest** is USD per persona, **query** USD per item and **time** wall-clock minutes per persona at 24-way concurrency (median over the ten personas). Spend is the metered API cost of each method's calls on the frozen stack, split into ingest and query by when each call was made; the shared media-to-text bridge is built once for the whole benchmark and charged separately.

Method	ingest	query	time	**Method**	ingest	query	time	**Method**	ingest	query	time
StandardRAG	0.11	0.0004	3	A-MEM	0.88	0.0009	46	OpenClaw	0.37	0.0016	5
RAPTOR	0.27	0.0005	15	Mem0	0.52	0.0006	36	Graphify	0.22	0.0003	2
LightRAG	1.10	0.0040	11	Graphiti (Zep)	2.14	0.0014	76	Cognee	0.52	0.0010	15
GraphRAG	0.89	0.0015	9	MIRIX	1.18	0.0018	17	LLM Wiki	0.78	0.0009	14
HippoRAG 2	0.39	0.0010	18	MemOS	0.46	0.0005	10	OKF	0.76	0.0010	4
KAG / OpenSPG	1.37	0.0019	19	MemEvolve	0.23	0.0002	6	Omni-SimpleMem	0.38	0.0011	9
Youtu-GraphRAG	0.80	0.0028	17	ATOM	1.75	0.0004	111	MMGraphRAG	1.07	0.0017	12
iText2KG	1.01	0.0003	113	MemPro	0.42	0.0111	13	M^3KG-RAG	1.26	0.0011	26
OntoKG	0.72	0.0001	73	HyperGraphRAG	1.25	0.0041	11				
Hyper-KGGen	1.18	0.0002	98	HyperRAG	1.28	0.0052	16	**KnowledgeClaw**	**0.32**	**0.0004**	**3**
MemGPT / Letta	0.12	0.0009	5								

F.2 MAIN RESULTS AND ABLATIONS (RQ1–RQ2)

Setting. Figure 5a divides each task score by the best score of any row of Table 1; Figure 5c compares each persona score with that of the strongest of the 30 baselines on that persona (two-sided sign test). Each ablation (Appendix E.6) shares the call cache and removes one component.

Results. KnowledgeClaw leads on all ten personas by $+13.3$ on average (Table 9), while the strongest baseline changes with the persona (MemPro seven times, OKF twice, LLM Wiki once); on $\mathcal{T}_1$ the four baselines of Figure 5a reach only 20.0–52.7% of its score. Each ablation loses most where its component acts: C1 on $\mathcal{T}_1$ (-20.0) and $\mathcal{T}_4$ (-13.6), C2 only on $\mathcal{T}_5$ (-46.0), and C3 on every task (-7.8 to -48.0), so they are complementary rather than redundant.

Table 9: **Per-persona comparison (RQ1).** Persona score of KnowledgeClaw and of the strongest of the 30 baselines on each persona, with the name of that baseline and the gain.

	HC	FIN	EDU	LAW	PSY	POL	HUM	BLD	MUS	GAM
KnowledgeClaw	**60.9**	**64.4**	**59.2**	**46.6**	**65.2**	**28.3**	**57.3**	**47.3**	**64.5**	**65.7**
Best baseline	45.5	45.2	50.7	38.9	53.6	23.6	42.4	37.4	43.4	45.5
which	OKF	MemPro	MemPro	OKF	MemPro	MemPro	LLM Wiki	MemPro	MemPro	MemPro
Gain	$+15.4$	$+19.2$	$+8.5$	$+7.7$	$+11.6$	$+4.7$	$+14.9$	$+9.9$	$+21.1$	$+20.2$

F.3 OMNIMODAL TYPED HYPERGRAPH (RQ3)

Setting. A slice holds the items of one task whose unit has the given modality. The 510 items of $\mathcal{T}_3$ and $\mathcal{T}_4$ are grouped by hops, by join shape (a *lookup chain* follows one value at a time, a *shared-entity* join meets a file on a common entity, a *conjunctive* one needs two entities) and by anchor (a named file or a quoted entity). A file is *joined* when another file lies within two hyperedges.

Results. KnowledgeClaw leads the best baseline on every slice, depth, join shape and anchor (Table 10), most where reading and joining are hardest: on $\mathcal{T}_1$ PDFs and images, where the best omnimodal baseline reaches 18.8 and 7.7, and at four hops (7.1 against 3.6); removing C1 costs up to 21.3 on a slice. The baseline median falls from 7.5 at two hops to 2.6 at three and 0.0 at four, while KnowledgeClaw keeps 7.1. On every persona the hypergraph joins 95.4–99.2% of files in 3.6–6.1 MB (Table 11).

Table 10: **Omnimodal typed hypergraph (RQ3).** Top: score per modality slice (n items). Bottom: set F1 on $\mathcal{T}_3$ and $\mathcal{T}_4$ by depth, join shape and anchor; *best*: strongest of the 30 baselines.

Slice	n	Ours	Best text-bridge	Best omnimodal	w/o C1
$\mathcal{T}_1$ PDF	344	**56.4**	30.8 (OKF)	18.8 (M^{3}KG-RAG)	35.1
$\mathcal{T}_1$ image	56	**21.6**	5.3 (MemPro)	7.7 (M^{3}KG-RAG)	10.0
$\mathcal{T}_2$ image	397	**82.9**	78.6 (MemPro)	70.0 (Omni-SimpleMem)	81.6
$\mathcal{T}_3$ PDF	125	**46.4**	40.0 (MemPro)	32.8 (MMGraphRAG)	32.0
$\mathcal{T}_4$ image+PDF	110	**20.9**	17.3 (OKF)	10.9 (Omni-SimpleMem)	7.3
$\mathcal{T}_6$ image	304	**71.4**	66.8 (MemPro)	65.5 (Omni-SimpleMem)	69.6

Join	n	Ours	Best	w/o C1	Join	n	Ours	Best	w/o C1
2 hops	326	**19.3**	16.9 (MemPro)	–	shared entity	84	**14.3**	8.3 (MemPro)	7.1
3 hops	156	**13.5**	12.2 (OKF)	–	conjunctive	51	**9.8**	5.9 (OKF)	3.9
4 hops	28	**7.1**	3.6 (RAPTOR)	–	file anchor	122	**12.3**	7.4 (OKF)	5.7
lookup chain	375	**18.4**	14.7 (MemPro)	11.7	entity anchor	388	**18.3**	14.2 (MemPro)	11.6

Table 11: **Committed hypergraph per persona (RQ3).** Hyperedges by kind; *joined*: % of files within two hyperedges of another file; 216 media files per persona are read natively.

	HC	FIN	EDU	LAW	PSY	POL	HUM	BLD	MUS	GAM
Files	396	417	413	396	396	396	396	417	417	418
Entities	6,023	4,302	4,487	4,584	5,606	3,759	4,358	3,782	3,039	3,981
Record	2,966	2,779	2,667	2,967	2,897	2,913	2,740	2,632	2,302	2,399
Event	597	223	201	396	468	222	311	194	141	133
Cross-modal	1,319	373	325	1,797	985	1,273	1,260	642	360	367
Joined (%)	98.2	98.6	97.3	99.2	98.7	98.7	96.2	99.0	95.4	98.6
Size (MB)	6.1	4.3	4.2	5.6	5.3	5.7	4.6	4.7	3.6	3.8

F.4 CLOSED-LOOP LIFECYCLE (RQ4)

Setting. Each of the 137 $\mathcal{T}_5$ items is answered (phase A), its next version is committed as REFRESH, and it is answered again (phase B); its score is the mean of both. A phase-B answer is *superseded* when it returns the replaced value. Latencies are single commits in the audit logs.

Results. KnowledgeClaw leads every baseline in both phases (93.4 and 97.8, against at most 81.8 and 80.3; Table 12); after the commit its evidence holds the new value on 95.5% of the items and the old one on none. Over 2,000 edits Eq. (6) always holds, while an append-only store ends with 10,148 stale nodes and 1,190 stale segments; a commit touches a median 0.33% of the hypergraph (0.86% at the 95th percentile), so DEL takes 9 ms and REFRESH 2.6 s against 118 s for a full build of the persona.

F.5 INTENT-ROUTED MULTI-GRANULAR RETRIEVAL (RQ5)

Setting. A file is *named* when its handle appears in the question. Evidence length is the median over a task's items at four characters per token; KnowledgeClaw's evidence is split by the section headers of its context into cards, linked-work cards, hyperedges and passages.

Results. Routing gives lookups anchor cards alone and joins a walk made mostly of linked-work cards (Table 13); on every task its evidence is shorter than flat top-k and scores higher, by $+7.8$ to $+48.0$. Each fixed tier wins only for its own intent; the router matches the best on every task. On $\mathcal{T}_4$, flat top-k brings all named files into the evidence for only 23.1% of the items, against 100% for C3, which resolves every handle of the question to its anchor file before retrieval.

Table 12: **Lifecycle consistency per phase (RQ4).** **A**, **B**: $\mathcal{T}_5$ accuracy before and after the edit; **Sup.**: % of phase-B answers that return the superseded value (lower is better).

Method	A	B	Sup.	Method	A	B	Sup.	Method	A	B	Sup.
NaiveGeneration	34.3	40.1	34.3	HyperRAG	27.7	35.8	27.0	Cognee	73.7	16.8	68.6
StandardRAG	73.0	30.7	64.2	MemGPT / Letta	32.8	35.8	34.3	LLM Wiki	73.0	28.5	70.8
RAPTOR	44.5	34.3	47.4	A-MEM	77.4	21.2	73.7	OKF	73.7	35.0	65.0
LightRAG	72.3	27.7	63.5	Mem0	38.7	37.2	28.5	Omni-SimpleMem	75.9	28.5	70.1
GraphRAG	27.7	38.0	29.2	Graphiti (Zep)	81.8	35.8	61.3	MMGraphRAG	26.3	38.7	29.2
HippoRAG 2	51.1	42.3	43.1	MIRIX	33.6	43.8	27.7	M^3KG-RAG	24.1	41.6	28.5
KAG / OpenSPG	57.7	35.8	53.3	MemOS	57.7	38.0	56.9	w/o C1	93.4	97.1	2.9
Youtu-GraphRAG	29.9	35.8	32.1	MemEvolve	27.0	38.0	30.7	w/o C2	93.4	5.8	94.2
iText2KG	19.0	36.5	24.8	ATOM	27.7	32.8	26.3	w/o C3	78.1	47.4	23.4
OntoKG	27.7	35.8	24.1	MemPro	77.4	80.3	19.0	**KnowledgeClaw**	**93.4**	**97.8**	**2.2**
HyperGraphRAG	35.0	45.3	33.6	OpenClaw	48.9	33.6	51.1				
Hyper-KGGen	24.8	30.7	26.3	Graphify	32.1	39.4	28.5				

Table 13: **Evidence per task (RQ5).** *Named*: % of items with all named files in the evidence; *tokens*: median evidence length; make-up of KnowledgeClaw's evidence in %; score as in Table 1.

	Named (%)		Tokens		Make-up of ours (%)				Score	
Task	Ours	Flat	Ours	Flat	Card	Linked	Edge	Passage	Ours	Flat
$\mathcal{T}_1$	**100**	93.8	1,972	3,686	100	0	0	0	**51.6**	15.7
$\mathcal{T}_2$	**100**	66.5	248	2,250	100	0	0	0	**82.8**	34.8
$\mathcal{T}_3$	**100**	85.4	1,416	2,647	25.2	50.4	7.5	17.0	**15.8**	8.0
$\mathcal{T}_4$	**100**	23.1	1,986	4,761	7.8	61.5	6.8	24.0	**20.9**	5.5
$\mathcal{T}_5$	**100**	63.5	2,145	3,840	100	0	0	0	**95.6**	62.8
$\mathcal{T}_6$	**100**	86.6	295	3,103	100	0	0	0	**76.0**	57.5

F.6 Efficiency and robustness (RQ6–RQ7)

Setting. The budget cuts saved evidence to B tokens; the sweep varies θ, the keep ratio, the cap on linked works or k around the defaults of Appendix E.6, one at a time, with the rest fixed.

Results. 512 tokens already give 55.0 OKE and 1,024 tokens 55.8, while flat top-k stays within 20.6–26.7 at every budget (Table 14). No setting of the sweep changes $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_5$ or $\mathcal{T}_6$, and OKE moves by at most 0.4; a larger keep ratio even lifts $\mathcal{T}_4$ to 32.7: the defaults are not tuned.

Table 14: **Efficiency and robustness (RQ6–RQ7).** Left: OKE with evidence cut to B tokens. Right: one hyperparameter varied (default underlined); $\mathcal{T}_{3,4}$: mean of $\mathcal{T}_3$ and $\mathcal{T}_4$.

B	Ours	Flat	θ	$\mathcal{T}_{3,4}$	OKE	keep	$\mathcal{T}_{3,4}$	OKE	links	$\mathcal{T}_{3,4}$	OKE	k	$\mathcal{T}_{3,4}$	OKE
256	**51.2**	20.6	0.80	16.7	55.9	0.4	17.6	56.0	2	18.2	56.2	5	17.1	56.0
512	**55.0**	22.0	0.83	16.9	55.9	0.5	17.5	56.0	4	16.9	55.9	<u>10</u>	16.9	55.9
1,024	**55.8**	22.4	<u>0.86</u>	16.9	55.9	<u>0.6</u>	16.9	55.9	<u>6</u>	16.9	55.9	15	17.1	56.0
2,048	**55.3**	24.8	0.89	16.7	55.9	0.7	19.8	56.3	8	16.7	55.9	20	16.9	56.0
full	**55.9**	26.7	0.92	16.3	55.8	0.8	19.6	56.3	10	16.9	56.0			

F.7 Case study (RQ8)

Setting. The case is the $\mathcal{T}_2$ item whose media is a video, the 150-second lecture `f75bfdd876eb3d22.mp4` that C1 reads natively from its keyframes and audio; its candidates describe four lectures of one MIT class by the same speaker. Frames and waveform come from the files, the last two rows of Table 2 from saved traces, and every answer from the main run.

Results. Sixteen of the 30 baselines choose a sibling: seven (A) (LightRAG, HippoRAG 2, KAG, HyperRAG, Mem0, Cognee, MMGraphRAG), seven (B) (StandardRAG, GraphRAG, iText2KG, OntoKG, HyperGraphRAG, Hyper-KGGen, ATOM) and two (C) (NaiveGeneration, A-MEM). KnowledgeClaw's 3,659-character card holds the title *Cities and a Just Energy Transition*, the people discussed and a timestamped transcript stating the topic; the 10,497 characters of w/o C3 hold identifiers of five other videos and no passage of the named one, and it picks the grid lecture (B).

G Extended Related Work

G.1 LLM-based Knowledge Engineering

Construction and retrieval. Beyond passage retrieval (Lewis et al., 2020) and recursive summaries (Sarthi et al., 2024), graph systems make relations explicit via entity communities (Edge et al., 2025), local–global retrieval (Guo et al., 2025b), or passage–entity links (Gutiérrez et al., 2025). Construction is now iterative, via incremental entity resolution (Lairgi et al., 2025), ontology routing (Li et al., 2026c), multi-agent enrichment (Lu et al., 2025), task-reward optimization (Tsang et al., 2026), joint reading and building (Han & Cheng, 2026), and inspectable Wiki/graph compilation (Ran et al., 2026). Reasoning-oriented systems adapt retrieval to the question through logical forms (Liang et al., 2025), schema-constrained agents (Dong et al., 2025), iterative multi-channel search (Yang et al., 2025), RL-trained graph interaction (Luo et al., 2026), or executable exploration policies (Lee et al., 2026). Hypergraphs keep n-ary facts intact (Luo et al., 2025; Huang et al., 2026), combine entity- and passage-level semantics (Wang et al., 2026; Hu et al., 2026a; Khrouf et al., 2026), and support planned or learned traversal (Zai et al., 2026; Lien et al., 2026; Park et al., 2026b; Phu et al., 2026; Wu et al., 2026a).

Persistent and evolving memory. Agent memory spans extraction, storage, retrieval, and evolution (Yang et al., 2026a): tiered context (Packer et al., 2024), evolving notes (Xu et al., 2025), salient-fact updates (Chhikara et al., 2025), graph–LLM interfaces (Markovic et al., 2025), memory as a system service (Li et al., 2025), compression (Fang et al., 2026; Liu et al., 2026b), typed fact/belief stores (Latimer et al., 2025), relation graphs and associative recall (Jiang et al., 2026; Ji et al., 2026), and hypergraph memories (Zhou et al., 2026; Yue et al., 2026; Cui et al., 2026b). Adaptation targets stored facts via temporal or bitemporal graphs (Rasmussen et al., 2025; Lairgi et al., 2026; Bao et al., 2026; Wang, 2026), supersession and provenance (Yadav, 2026; Yacoubi et al., 2026), feedback-driven revision (Fu et al., 2026; Li et al., 2026d; Cui et al., 2026a), or the memory architecture itself (Zhang et al., 2025; Liu et al., 2026d; Yang et al., 2026b); none specifies what happens to every dependent index when a source file disappears. Compiled Wikis (Ming et al., 2026; Dong et al., 2026) and the software baselines OpenClaw (OpenClaw contributors, 2026), Graphify (Graphify contributors, 2026), LLM Wiki (Karpathy, 2026), and OKF (Google Cloud, 2026) offer another route to persistence. Diagnostic studies separate encoding, retrieval, and generation failures (Liu et al., 2026e), check memories against external state (Suresh et al., 2026), test revocation at retrieval (Shen et al., 2026), and show that compression can lose distinctions needed after an update (Zhang, 2026). KnowledgeClaw adds a file-level contract: provenance-guided edits update the hypergraph and every dependent index.

G.2 Multimodal Retrieval-Augmented Generation

Cross-modal construction and granularity. Surveyed by Abootorabi et al. (2025), systems align vision to language for knowledge-graph construction (Liu et al., 2025; Ruosch & Rossetto, 2026), connect textual and cross-modal graphs (Guo et al., 2025a; Wan & Yu, 2026; Yuan et al., 2026b), and carry visual evidence through the whole pipeline (Hsiao et al., 2026; He et al., 2026); others control evidence granularity via page retrieval (Cho et al., 2024), selective chunk expansion (Hu et al., 2026b), grounded regions (Dai et al., 2026; Wu et al., 2026b), reliability-routed subgraphs (Ali et al., 2026), citation-aware RL (Zhao et al., 2026), or context folding (Tian et al., 2026), while TrioRAG (Rakshit et al., 2026) fuses retrieval signals without an explicit graph.

Higher-order, memory, and maintenance settings. Multimodal hypergraphs link text, images, and tables (Chen et al., 2026) and serve domain applications (Serna-Aguilera et al., 2026a; Ma et al., 2026a; Serna-Aguilera et al., 2026b); audiovisual systems pair graphs with long-video context and multi-hop selection (Ren et al., 2026; Zhu et al., 2026; Park et al., 2026a). Multimodal memories coordinate specialized stores (Wang & Chen, 2025; Liu et al., 2026c;a), maintain episodic, semantic, and visual memory for video (Yeo et al., 2026; Ma et al., 2026b; Li et al., 2026b), and select evidence under a budget (Li et al., 2026a; Gao et al., 2026). The closest work edits multimodal knowledge, via multi-hop knowledge editing (Yuan et al., 2026a), agentic hypergraph editing (Lin et al., 2026), or lifecycle operations on higher-order memories (Feng et al., 2026); we instead target source-file edits that propagate through shared entities, cross-modal links, and indices. Beyond benchmarks of personalization scope (Hu et al., 2026c) and cross-modal multi-hop paths (Cai et al., 2026), OmniKE-Bench also tests post-edit maintenance.