Context Handling Mechanisms in Human Agent Collaboration: Sequencing, Retrieval, Summarization and Context Window Size

Lukas Röß
Munich University of Applied Sciences
Munich, Germany
lukas.roess@hm.edu

Abstract

Large language models act as agents in human–AI collaboration across domains such as analysis, writing and decision support. In these settings, humans and agents interact over many turns. The agent must keep track of prior content, refer back to earlier decisions and adapt to changing goals. This need shifts attention from single prompt accuracy to sustained context handling.

Research explores several mechanisms to manage context, including multi agent sequencing, retrieval augmented generation, summarization and extended context windows. Most work evaluates these mechanisms in isolation or on technical benchmarks. Less is known about how they jointly affect collaboration between humans and agents.

This paper synthesizes recent work from 2024 and 2025 on long context processing, memory architectures and human–agent collaboration. It reviews multi agent sequencing approaches such as Chain of Agents, Graph of Agents and XpandA, which divide long inputs across agents and recombine their outputs. It contrasts these methods with empirical comparisons of retrieval augmented generation and long context models in industrial and clinical applications. It also integrates work on memory structures, summarization strategies and studies of human teams supported by AI summarizers.

Based on this synthesis, the paper proposes a framework that links four context handling mechanisms, namely sequencing, retrieval, summarization and context window size, to three dimensions of collaboration effectiveness: task performance, interaction quality and long-term consistency. It refines the research question to: How do different mechanisms of context handling, such as sequencing, retrieval, summarization and context window size, jointly influence the effectiveness of human–agent collaboration?

CCS Concepts

• **Human-centered computing** → **Collaborative interaction**; • **Computing methodologies** → *Natural language processing*; • **Information systems** → *Information retrieval*.

Keywords

large language models, human–agent collaboration, context windows, retrieval augmented generation, summarization, memory

ACM Reference Format:
Lukas Röß. 2025. Context Handling Mechanisms in Human Agent Collaboration: Sequencing, Retrieval, Summarization and Context Window Size. In *Proceedings of Main Seminar on Human–Agent Collaboration (Seminar Human–Agent Collaboration).* ACM, New York, NY, USA, 9 pages.

Seminar Human–Agent Collaboration, Munich, Germany
2025. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM

1 Introduction

Large language models (LLMs) have shifted from static tools to agents that participate in ongoing collaboration with humans. In domains such as analysis, writing and decision support, humans and LLM agents exchange information over extended interactions. The agent must keep relevant context, recover prior steps and adapt to updated goals, not only respond to isolated prompts.

A central technical and design question concerns how LLM based agents handle context. One approach extends the context window of a single model. Longer windows allow direct conditioning on long inputs but face limits such as attention dilution and reduced use of early tokens. Multi agent approaches therefore explore sequencing and division of labour. Chain of Agents processes long inputs by passing segments through worker agents and uses a manager agent to integrate their outputs. This method improves performance over retrieval based and full context baselines on long context tasks [20]. Graph of Agents generalizes this idea by constructing dynamic collaboration graphs and frames long context processing as a compression problem. It often outperforms both retrieval augmented generation and fixed multi agent baselines while using short local windows [4]. XpandA introduces a question driven multi agent framework that partitions long texts, coordinates agents through a shared memory and processes sequences of up to one million tokens with competitive accuracy and latency [18].

Other work compares retrieval augmented generation and long context models in applied settings. An industry study reports that long context models outperform retrieval augmented generation when resources permit, while retrieval remains more cost efficient and can be paired with long context models through routing strategies [7]. In the medical domain, a question answering study shows that long context retrieval architectures can counter the usual performance drop for information in the middle of a long sequence and improve answer quality over standard retrieval pipelines [19].

Beyond capacity, memory mechanisms and summarization strategies play a key role. A recent survey reviews memory mechanisms for LLM agents, including short term context, long term and external memory. It argues that memory design must reflect the dynamics of human interaction [21]. Work on contextual memory intelligence treats memory as system infrastructure that captures decisions and their context over time. It aims to support reflection, accountability and stable collaboration [16]. Empirical work from DeepMind studies how collaboration modes and memory banks interact with chain of thought and multi agent setups. The findings show that exemplar retrieval and memory can distract models in some tasks instead of supporting them [9].

Studies of human teams with AI support complement these technical perspectives. An experiment with virtual collaborative analysis compares two AI summarizers. Summary style affects trust and attention patterns even when problem solving accuracy remains similar [5]. Another study of student interactions with LLMs on complex tasks finds that interaction patterns often reduce to instruction-response cycles with long threads that reveal misalignment between human intent and model output [13].

These results indicate that context handling concerns more than the size of the context window. Sequencing, retrieval, summarization and memory design interact with human trust, attention and perceived agency. Yet most work evaluates these mechanisms separately or with a narrow focus on model performance. Integrated accounts of how context handling mechanisms jointly shape human–agent collaboration remain scarce.

This study addresses this gap through the following research question.

Research question. *How do different mechanisms of context handling, such as sequencing, retrieval, summarization and context window size, jointly influence the effectiveness of human–agent collaboration?*

The paper contributes in three ways. First, it summarizes recent work on long context processing, memory and collaboration into a structured view of context handling mechanisms. Second, it relates these mechanisms to dimensions of collaboration effectiveness that are relevant for human–agent teaming. Third, it reports a benchmark-based evaluation that operationalizes the research question.

2 Background

2.1 Human–agent collaboration with LLMs

Human–agent collaboration arises when humans and artificial agents work on tasks with shared goals, roles and responsibilities. Modern LLM based agents can generate text, reason over documents, call tools and maintain conversational histories. These abilities support scenarios in which the agent acts as a partner rather than as an oracle that answers one isolated query.

Empirical work shows that current systems still fall short of this ambition. An analysis of student interaction sequences with LLM based tools in a complex problem solving task finds that most interactions follow an instruction-response pattern. Students issue instructions, the AI responds and the pattern repeats. Negotiation, joint sensemaking and mutual adaptation remain rare [13]. The system records long conversational histories but uses them in a limited way for deeper collaboration.

Research on human teams with AI summarizers suggests that context handling can shape collaboration processes. In a virtual team experiment, an AI summarizer produced either informative or indicative summaries of shared documents. Problem solving accuracy remained similar across conditions. However, informative summaries increased trust in the AI and changed how often and how quickly participants attended to teammate information [5]. Summary style thus affected trust and attention, which matter for effective collaboration.

These findings motivate a closer focus on context handling mechanisms in LLM agents and how they relate to human behaviour and perception, not only to model accuracy.

2.2 Context representation and memory in LLM agents

From a technical view, LLMs handle context through four main strategies relevant to this study. The first is extended context windows that allow more tokens as direct input. The second is retrieval augmented generation, which retrieves documents or conversation snippets from external stores and injects them into a smaller window. The third is summarization, which compresses prior content into shorter representations. The fourth is explicit memory architectures that preserve information beyond a single prompt–response cycle.

A survey on memory mechanisms for LLM based agents distinguishes short term rolling context, episodic memory in vector stores, semantic memory graphs, parametric memory via fine tuning and hybrid memory controllers. It emphasizes that memory systems must decide what to store, when to update and how to retrieve information for different tasks and collaboration patterns [21]. This view supports the idea that sequencing, retrieval and summarization form components of a broader memory system.

Work on contextual memory intelligence extends this notion and treats memory as a system level capability for generative AI. It proposes a layered architecture with an insight layer that captures decisions, rationales and environmental signals over time. The design includes human review and drift detection and aims to support coherent collaboration in organizational settings [16]. This work connects memory design to social and institutional contexts instead of narrow benchmarks.

Empirical work on collaboration and memory highlights trade offs. A DeepMind study compares fixed, random and similarity based retrieval of exemplars, and frozen versus learned memory banks, in multi agent reasoning tasks. Similarity based retrieval does not dominate random selection on all tasks. In some cases, any exemplars reduce performance [9]. These results caution against the assumption that more memory or more complex retrieval always improve reasoning or collaboration.

2.3 Mechanisms of context handling

This study focuses on four context handling mechanisms that recur across the literature.

Sequencing. Sequencing structures processing as a sequence of steps, often across several agents with specific roles. Chain of Agents introduces worker agents that process segments of a long input and a manager agent that synthesizes their outputs. This design exploits long documents while giving each agent a short local window [20]. Graph of Agents formalizes long context processing as a compression problem and builds dynamic graphs in which agents iteratively exchange and refine information under an information theoretic objective [4]. XpandA uses a question driven workflow and shared memory. It partitions ultra long texts and lets agents update a central ensemble of information, which supports processing of sequences up to one million tokens [18].

Retrieval. Retrieval augmented generation fetches relevant documents, conversation segments or exemplars from external stores and passes them to the model within a limited window. The EMNLP industry study benchmarks retrieval augmented generation and long context models on several datasets. It finds that long context models reach higher average performance when model and hardware resources allow, while retrieval remains more cost efficient. A hybrid routing method that directs queries either to retrieval augmented pipelines or to long context inference achieves near long context performance at reduced cost [7]. The medical question answering study combines retrieval with long context representations and reduces the typical performance loss for information in the middle of long sequences, which improves answer quality [19]. This result shows that context window size and retrieval can act as complementary mechanisms.

Summarization. Summarization compresses content so that systems can store and share more information under context limits. In multi agent systems, summarization shapes both internal state and communication between agents. Chain of Agents relies on the manager agent to summarize worker outputs into a final answer [20]. Graph of Agents uses compression objectives to guide how agents summarize and pass information along edges [4]. XpandA maintains summarized ensembles of information in a central memory and replays partitions when needed [18]. In human teams, AI summarizers control which information humans see and when, and thus influence attention and trust [5].

Context window size. Context window size determines how many tokens a model can process in one pass. Extended windows allow direct conditioning on long histories but do not ensure effective use of earlier content. Studies on long context models show that performance gains depend on task structure and input distribution. The EMNLP industry study reports that long context models outperform retrieval augmented generation when resources are sufficient and that hybrid routing is needed under cost constraints [7]. The medical study shows that long context retrieval pipelines can leverage extended windows to improve clinical question answering [19]. Multi agent approaches such as Graph of Agents show how collaboration architectures can surpass much larger windows by distributing processing across small local windows [4].

These four mechanisms form the conceptual basis for the analysis and research design in the remainder of the paper.

3 Related work

3.1 Sequencing and multi agent collaboration

Multi agent approaches treat an LLM based system as a team of agents. Each agent handles part of the context or reasoning process. Chain of Agents introduces a design in which worker agents process segments of a long document in sequence and communicate their findings. A manager agent aggregates these findings and produces a final answer. Experiments on long context question answering, summarization and code completion show that this approach outperforms retrieval augmented generation, full context operation and other multi agent baselines, even though each agent has a short local window [20]. Sequencing thus compensates for limited context capacity.

Graph of Agents extends this design through dynamic graphs of agents. The communication pattern follows an information theoretic compression objective rather than a fixed chain. Agents connect according to input structure and refine compressed representations over iterations. On long context question answering benchmarks, this method improves F1 scores over retrieval augmented generation and a strong fixed multi agent baseline and increases effective context beyond the base model limit [4]. Emergent collaboration structures in the graph appear more effective than hand crafted pipelines.

XpandA targets ultra long sequences. It partitions texts so that each partition fills the context window to a set degree. Agents follow a question driven workflow and update a shared memory that stores an ensemble of information. Agents replay partitions when needed to resolve temporal structures such as flashbacks. XpandA improves accuracy and inference speed over full context models, retrieval augmented generation and previous agent based methods on long context tasks [18]. It illustrates how sequencing, shared memory and dynamic partitioning support long context processing.

3.2 Retrieval augmented generation and long context models

The trade off between retrieval augmented generation and extended context windows is central for context handling. The EMNLP industry study compares these strategies with several contemporary LLMs across diverse datasets. Long context models achieve higher average performance when model and hardware resources are provisioned. Retrieval augmented generation remains attractive due to cost. The study proposes a hybrid routing method in which the system sends queries either to retrieval augmented pipelines or to long context inference based on model self reflection. This method approaches long context performance at lower cost [7]. The work quantifies trade offs between context capacity and retrieval in realistic conditions.

In the clinical domain, long context retrieval architectures support medical question answering. The npj Digital Medicine paper combines retrieval augmented language models with extended context windows and designs a method that reduces the performance drop for information in the middle of long contexts. The system uses long context representations with retrieval and improves answer quality on clinical datasets compared to standard retrieval augmented generation baselines [19]. This result shows that context window size and retrieval can act as complementary mechanisms.

These studies suggest that effective human–agent collaboration in high stakes domains will use hybrid systems that decide when to retrieve, when to rely on long context and how to combine both within an interaction.

3.3 Memory architectures and summarization

Memory architectures aim to preserve information across interactions and tasks. The survey on memory mechanisms for LLM based agents reviews sliding window context, episodic memory in vector stores, semantic memory graphs, learned key–value caches and other structures. It organizes these mechanisms by retention

horizon and access pattern and describes a tension between storing rich context and avoiding distraction from outdated content [21]. The survey supports a view in which sequencing, retrieval, summarization and window size form parts of a memory system.

Contextual memory intelligence introduces a normative perspective on memory design. It argues that memory should support coherent behaviour, explanation and governance in organizational settings. The proposed insight layer captures decisions, rationales and environment changes and integrates human review and drift detection [16]. Memory becomes a core element for human–agent collaboration that must remain aligned with institutional constraints over time. This view motivates the inclusion of collaboration effectiveness metrics beyond accuracy.

DeepMind's empirical work on collaboration and memory links these concepts to performance outcomes. The study varies collaboration styles, memory bank designs and retrieval methods in controlled tasks and measures reasoning performance. Chain of thought prompting, multi agent collaboration and memory banks interact in complex ways. Similarity based retrieval does not always outperform random exemplar selection, and memory can reduce performance in some tasks [9]. These findings show that memory and summarization mechanisms require evaluation in the context of specific tasks and collaboration patterns.

3.4 Human teams with AI support

Research on small groups provides further insight into AI mediated context handling. In the virtual collaborative analysis study, teams used a platform with an AI summarizer that produced either informative or indicative summaries of shared documents. Participants who received informative summaries reported higher trust in the AI and showed different attention patterns to teammate information. Problem solving accuracy stayed similar across conditions [5]. Summary style thus shaped social dynamics and perceptions even when task level outcomes were stable.

Another study examines how students interact with LLM systems while solving a complex task. Sequence and network analysis show that interaction patterns cluster around simple instruction-response dynamics. Long message threads reveal sustained misalignment between prompts and AI responses, and the study does not find a strong link between assignment complexity, prompt length and performance [13]. The findings suggest that current LLM systems, tuned for instruction following, do not yet act as strong partners for cognitive collaboration.

These results underline that context handling mechanisms act as design levers for trust, attention and perceived agency in human–agent collaboration, not only for task accuracy.

4 Conceptual framework

Based on the reviewed literature, this section proposes a framework that relates context handling mechanisms to collaboration effectiveness.

The framework considers four mechanisms: sequencing, retrieval, summarization and context window size. It examines their influence along three dimensions of collaboration effectiveness.

Task performance. This dimension covers accuracy, completeness and efficiency on joint tasks. Multi agent sequencing improves performance on long context question answering and summarization compared to retrieval augmented baselines [4, 18, 20]. Long context retrieval architectures increase answer quality in medical question answering [19]. Hybrid routing between retrieval and long context reaches near long context performance at reduced cost [7].

Interaction quality. This dimension describes trust, attention and communication patterns in human–agent teaming. Informative AI summaries raise trust and change attention to teammate information [5]. In contrast, studies of real world interactions with LLMs report limited negotiation and misalignment between human intent and AI output [13]. Memory and summarization mechanisms can support or hinder coordinated reasoning in multi agent systems [9].

Longitudinal alignment. This dimension concerns coherence and alignment over extended periods. Memory surveys and contextual memory intelligence describe memory mechanisms that maintain stable yet adaptable representations of context, decisions and rationales [16, 21]. Multi agent frameworks such as XpandA show how shared memory and partition replay preserve consistency across very long inputs [18].

The refined research question asks how different combinations of the four mechanisms affect these three dimensions of effectiveness in human–agent collaboration scenarios.

5 Methods

This section details the benchmark-based evaluation and the implemented prototype pipeline. The goal is to operationalize the conceptual framework and to test how different context handling mechanisms jointly affect collaboration effectiveness without running a human subject study.

5.1 Study design

Following feedback that a human study would be too time intensive for the seminar timeline, the evaluation is reframed as a benchmark based comparison. Because several long context benchmarks rely on dataset scripts that are not supported by the local loader, this milestone uses synthetic benchmarks that mirror long question answering, long summarization, retrieval-heavy and constraint tasks. Real world datasets can be added later as local JSONL or Parquet files.

The study follows a repeated measures design: the same benchmark instances are processed by each of the four agents. This controls for task difficulty and isolates the effect of context handling. Benchmarks are generated deterministically with a fixed seed, and all agents share the same decoding parameters and maximum generation length so that differences reflect the context mechanism rather than sampling variance.

5.2 Experimental conditions

(1) **Long context only.** A single long context LLM agent operates with an extended context window and no external retrieval or explicit long term memory. The full task corpus and dialogue history are provided directly in the prompt up to the context limit.

(2) **Retrieval augmented generation (RAG).** A smaller context LLM agent uses retrieval from a vector store that contains the task corpus and selected interaction history. For each turn, the system retrieves top-k passages based on dense similarity and injects them into the prompt [6].
(3) **Summarization focused agent.** An agent maintains a structured summary of the collaboration so far (key findings, decisions, open questions). Before each major step, it presents a concise summary panel to the participant and uses this summary as part of its own context instead of the full history.
(4) **Sequenced multi agent system.** A simplified Chain of Agents style pipeline in which two worker agents process different parts of the corpus and a coordinator agent synthesizes their outputs. The coordinator maintains a shared memory of intermediate conclusions and uses short local windows for each call.

All conditions use the same base model family to avoid confounding architecture with context handling. Differences lie in how context is assembled and reused: the long context agent uses a single full-context prompt, the RAG agent composes a prompt from the query plus retrieved passages, the summarization agent replaces the history with a structured summary, and the sequenced agent decomposes inputs across worker calls before a coordinator produces the final answer.

5.3 Implementation

The evaluation pipeline is implemented in Python and uses a shared model wrapper for all agents. All agents use the same base model (Qwen3-4B) with 4-bit quantization and identical decoding settings (temperature 0.2, top-p 0.95, max 512 new tokens) to isolate the effect of context handling. The implementation relies on Hugging Face Transformers for model loading and generation [17]. Inference runs on a local NVIDIA 3060 Ti. Each run writes a JSONL record with benchmark name, agent variant, output, token counts and latency for later analysis.

The long context agent sends the full input in a single prompt. The RAG agent builds a FAISS index per benchmark [3] and retrieves the top-k passages (default $k = 5$) using a sentence-transformer embedder (all-MiniLM-L6-v2) [11]. For stability and reproducibility in this milestone, embedding and indexing are computed on CPU while model inference runs on GPU. The summarization agent maintains a structured summary state with decisions, constraints and open questions; this summary replaces the full history in its prompt and is updated after each response. The sequenced agent splits the input into two partitions, runs two worker calls on the partitions, and merges them with a coordinator call that synthesizes a final answer.

All agents share the same prompt wrapper (role, task instruction, and input content) to reduce surface variation. The only differences between agents are how they construct context: full input, retrieved passages, summary state, or partitioned worker outputs.

5.4 Benchmark suite and protocol

The benchmark suite targets the four mechanisms of interest and covers multiple task types.

- **Long Question Answering:** Each instance contains 12 short documents. One document includes a hidden target token (e.g., ALPHA-3). The task is to return the target token, which requires scanning across the full context. This task is modeled after long-context QA benchmarks such as LongBench [1].
- **Long Summarization:** Each instance contains 8 sections with three planted findings. The task is a 2–3 sentence summary; evaluation checks whether the findings appear in the output. This setup is inspired by long-document summarization benchmarks such as GovReport and QMSum [2, 22].
- **Retrieval:** A corpus of 200 short passages contains one target passage with a key metric reference. The query asks which document mentions the metric, so successful retrieval is required for high evidence coverage. The design mirrors retrieval-heavy benchmarks such as BEIR [14].
- **Constraints:** Prompts include explicit constraints (bullet count, required keyword, and a forbidden word). This probes longitudinal constraint adherence across an interaction and reflects instruction-following evaluations such as IFEval [23].

Each agent variant runs on identical prompts and datasets with a fixed budget of tokens and calls. Synthetic benchmarks are generated with a fixed seed (42) to ensure that all agents process the same instances. This ensures differences are attributable to context handling rather than model size or sampling settings.

5.5 Measures

To align with the conceptual framework, the study collects measures for the three dimensions of collaboration effectiveness.

Task performance.

- **Token F1 (Long Question Answering):** Token overlap between output and reference target (whitespace tokens), following standard QA evaluation practice [10].
- **ROUGE-L (Long Summarization):** Longest common subsequence overlap for summaries [8].
- **Semantic similarity (Long Summarization):** Cosine similarity of sentence-transformer embeddings [11].
- **Evidence coverage (Retrieval):** Fraction of required supporting facts captured in the output when reference evidence is available, computed by string matching planted facts. This task-specific heuristic is similar in spirit to evidence-based evaluation in fact verification datasets such as FEVER [15].
- **Efficiency:** Tokens consumed and wall time per task are logged for later analysis.

Interaction quality (proxy metrics).

- **Constraint adherence:** Fraction of explicit constraints satisfied in sequential prompts (bullet count, required keyword, and forbidden word checks), aligned with instruction-following evaluations [23].

5.6 Evaluation pipeline

Each agent run writes a JSONL record that includes benchmark name, agent variant, output, token counts and latency. Synthetic benchmartks are generated deterministically with a fixed seed (42)

Table 1: Per-benchmark Long Question Answering and Long Summarization metrics (mean ± SD). Sample sizes: $n = 15$ for both tasks.

Agent	Token F1	ROUGE-L	Semantic similarity
Long context	0.0018±0.0027	0.072±0.012	0.704±0.049
RAG	0.0020±0.0002	0.039±0.001	0.747±0.031
Summarization	0.0004±0.0015	0.016±0.015	0.506±0.157
Sequenced	0.0009±0.0010	0.026±0.004	0.568±0.069

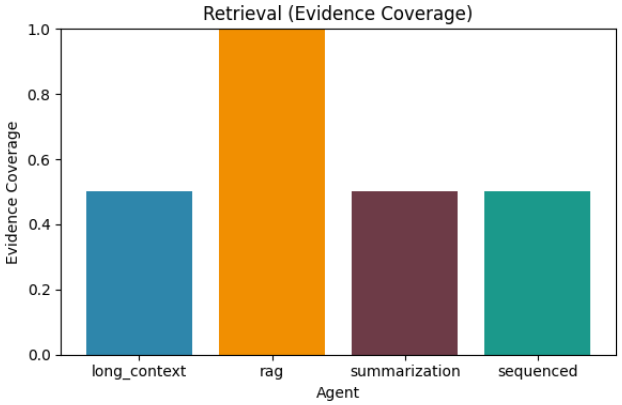

Figure 1: Retrieval benchmark: evidence coverage across agents.

to ensure reproducibility. Metrics are computed offline and aggregated overall and per benchmark. The plotting scripts generate per-metric bar charts with consistent agent colors and benchmark labels, which support both overall comparison and per-benchmark inspection.

6 Results

The four agents were evaluated on the benchmark suite using the long run in `runs/output_large.jsonl`. Long Question Answering, Long Summarization, Retrieval and Constraints each include 15 instances. Table 1 reports mean and standard deviation for Long Question Answering and Long Summarization. Figures 1 and 2 report evidence coverage and constraint adherence. Additional plots and the retrieval/constraints summary table are included in Appendix A (Figures 3 and 4, Table 2). Paired statistics are computed across matched instances; 95% confidence intervals are available from the analysis script but omitted for brevity.

Long Question Answering (Token F1). Token F1 is uniformly low across agents (0.0004–0.0020), reflecting the strictness of exact-token matching on short targets (Table 1, Figure 3 in Appendix A). RAG is marginally higher than the sequenced and summarization agents (paired Wilcoxon $p = 1.2 \times 10^{-4}$ and $p = 0.011$ respectively), but the mean differences are small (about 0.0012 and 0.0017). Long context and RAG are statistically indistinguishable on this metric ($p = 0.91$), indicating no clear advantage from retrieval in this setting.

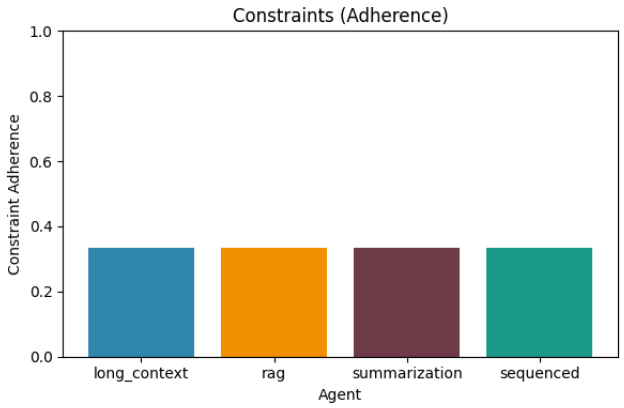

Figure 2: Constraints benchmark: constraint adherence across agents.

Long Summarization (ROUGE-L and semantic similarity). Long context yields the highest ROUGE-L (0.072±0.012), while RAG yields the highest semantic similarity (0.747±0.031) (Table 1, Figure 4 in Appendix A). The ROUGE-L advantage of long context over RAG is strong (paired Wilcoxon $p = 6.1 \times 10^{-5}$, Cohen's $d = 2.88$), but the direction reverses for semantic similarity ($p = 0.0043$, $d = -0.91$). The summarization agent shows high variance in semantic similarity (SD 0.157), suggesting instability across instances.

Retrieval (evidence coverage). RAG achieves full evidence coverage (1.0) on all instances, while the other agents remain at 0.5 with zero variance (Figure 1, Table 2 in Appendix A). This indicates a ceiling effect for retrieval and a floor effect for the non-retrieval agents. Pairwise tests between RAG and other agents are significant ($p = 1.1 \times 10^{-4}$) but the absence of variance limits effect-size estimation.

Constraints (constraint adherence). All agents score 0.333 with zero variance (Figure 2, Table 2 in Appendix A), indicating that none reliably satisfy all three constraints. This uniform outcome suggests a bottleneck in prompt compliance rather than a specific context mechanism.

Synthesis. Across the four mechanisms, retrieval is the only one that reliably improves evidence coverage on retrieval-heavy tasks, whereas long-context conditioning yields higher lexical overlap in summarization. Summarization and naive sequencing do not outperform the long-context baseline on any primary metric in this setup. Context window size is not independently manipulated, so its effect remains ambiguous relative to the other mechanisms.

Appendix A includes the remaining summary table and plot. Appendix B lists the configuration and environment used for these results.

7 Discussion

The results support a narrow but clear conclusion: retrieval provides the most reliable gains when the task requires locating sparse evidence in a large context, while long-context conditioning benefits lexical overlap in summarization. Sequencing in its current

form (simple input halving plus coordinator merge) does not deliver improvements, and summarization memory alone performs poorly on both overlap and semantic similarity. These findings partially answer the research question by showing that the mechanism used to assemble context matters more than the nominal window size in this setup; however, the effect of window size itself is not isolated because all agents share the same base model and context limit.

Several limitations temper these conclusions. First, the benchmarks are synthetic and small ($n = 15$), so variance estimates are noisy and statistical tests are uncorrected for multiple comparisons. Second, evidence coverage and constraint adherence exhibit ceiling and floor effects, which limit discrimination. Third, the RAG agent retrieves from a fixed corpus and the sequenced agent uses a naive split, both of which may bias outcomes. Finally, the evaluation uses a single model family and quantization setting; generalization to other models and real datasets remains unknown.

8 Future Work

Future work could expand the synthetic sample size to stabilize metric estimates and introduce real-world corpora by converting datasets to JSONL or Parquet to avoid loader script limitations. The sequenced agent could be upgraded with task-aware partitioning and iterative verification steps, and the summarization agent could be improved with structured update prompts rather than a single summary slot. Additional metrics, including contradiction detection and hidden recall probes, could be added to better capture long-term consistency. A small-scale human study can be reintroduced once the benchmark pipeline yields stable and reproducible patterns.

To evaluate real collaboration behaviour, a small case study remains a suitable follow-up. A scoped study with 6–12 participants could compare the four agents on one realistic task (e.g., collaborative report summarization) in short sessions. Measures should include task accuracy, NASA-TLX workload, trust ratings, and qualitative analysis of interaction breakdowns. This design would test whether the benchmark effects translate to human outcomes while staying within a manageable scope.

9 Conclusion and outlook

This paper refines the study focus in line with feedback that context management and sequencing matter at least as much as raw context window size for human–agent collaboration. Through a targeted review of work from 2024 and 2025, it synthesizes contributions on multi agent sequencing for long context processing, empirical comparisons of retrieval augmented generation and long context models, memory and summarization mechanisms, and studies of human teams with AI support.

The review suggests that effective collaboration depends on how systems structure, retrieve, summarize and preserve context over time, not only on the number of tokens an LLM can process. Multi agent frameworks such as Chain of Agents, Graph of Agents and XpandA show that sequencing and shared memory extend effective context beyond model limits. Industrial and medical studies show that hybrid combinations of retrieval and long context are promising for applied use. Memory surveys and contextual memory intelligence frameworks frame memory as system infrastructure for stable collaboration. Research on human teams highlights that summarization style and interaction patterns influence trust and perceived collaboration quality.

Building on this synthesis, the paper formulates a research question that targets the joint influence of sequencing, retrieval, summarization and context window size on the effectiveness of human–agent collaboration. It proposes a conceptual framework and outlines a comparative study that will operationalize this question through controlled experiments.

Future work could replace synthetic tasks with real world corpora and refine the sequencing strategy so that multi agent coordination is more than a simple partition. The longer-term goal is to provide guidance on how to combine context handling mechanisms in LLM based systems so that they function as collaborative partners rather than as obedient yet misaligned assistants.

A Additional results

This appendix contains the remaining benchmark summary table and plots referenced in the Results section.

Table 2: Per-benchmark Retrieval and Constraints metrics (mean ± SD). Sample sizes: $n = 15$ for both tasks.

Agent	Evidence coverage	Constraint adherence
Long context	0.500±0.000	0.333±0.000
RAG	1.000±0.000	0.333±0.000
Summarization	0.500±0.000	0.333±0.000
Sequenced	0.500±0.000	0.333±0.000

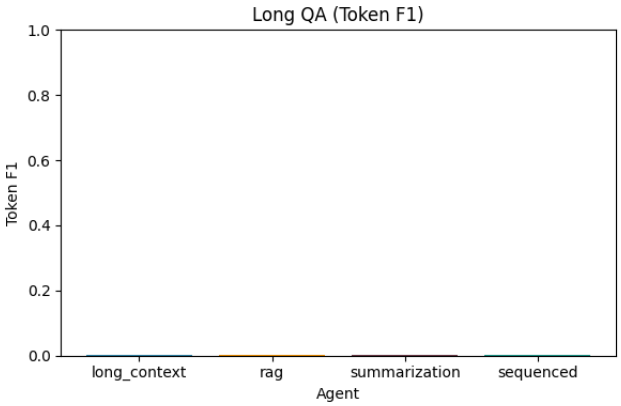

Figure 3: Long Question Answering benchmark: Token F1 across agents.

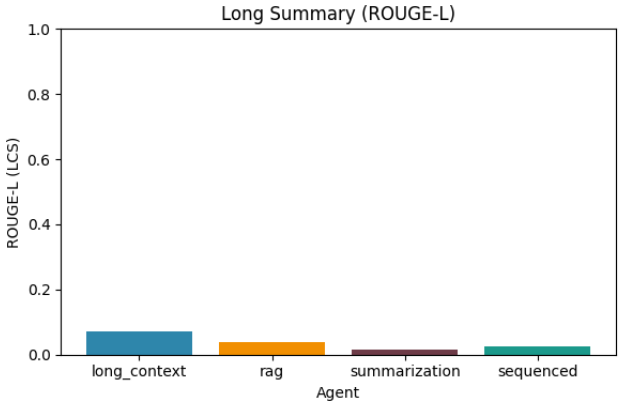

Figure 4: Long Summarization benchmark: ROUGE-L across agents.

B Reproducibility

Data and tasks. Synthetic benchmarks from the local generators in `benchmarks/` are used. Long Question Answering, Long Summarization, Retrieval and Constraints use 15 instances each. The generators use a fixed seed (42) in code; the run artifacts do not store the seed, so this is inferred from the implementation.

Configuration. All agents use Qwen3-4B with 4-bit quantization and shared decoding settings (temperature 0.2, top-p 0.95, max 512 new tokens). The RAG agent retrieves top-k passages with $k = 5$. The retrieval corpus size is 200. These values are read from the code and are not logged per record.

Environment. Inference ran on a local NVIDIA 3060 Ti. The runs use Python with Hugging Face Transformers, FAISS, and Sentence-Transformers. Exact library versions are not captured in the run logs.

Code availability. The benchmark pipeline and analysis scripts are available at https://github.com/luroess/human-agent-collaboration [12].

C PRISMA search and screening

Search strings. Searches covered OpenAlex, IEEE Xplore, ACM Digital Library, and arXiv (January 2024 to present) using the following queries:

- `title: "human-agent" AND text: "context window" OR text: "rag"`
- `text: "context window" AND text: "rag"`
- `text: "human-agent" AND text: "retrieval-augmented generation"`
- `text: "human-agent" AND text: "context window"`
- `title: "human" AND text: "context window"`
- `title: "human" AND text: "rag"`

Inclusion and exclusion criteria. Inclusion criteria cover studies that address human–agent collaboration and mention context windows or RAG. Exclusion criteria remove studies that focus only on LLM architecture, RAG, or context windows without collaboration, and studies published before January 2024.

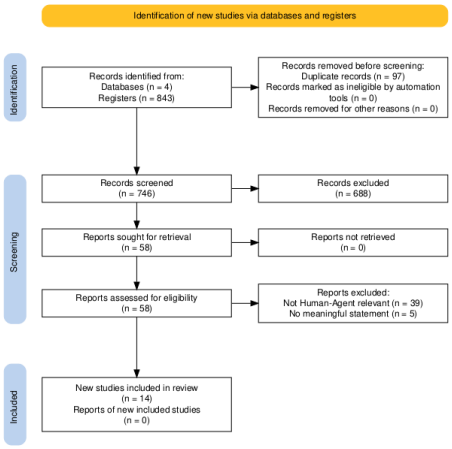

Figure 5: PRISMA flow diagram for the literature search. The diagram reports the screening counts.

Extraction summary. The extraction captured bibliographic metadata, collaboration setting, context-handling mechanism(s), dataset or task, evaluation metrics, and key findings. The extraction log does not record per-field counts, so these counts are unknown.

Acknowledgments

Special thanks go to Prof. Dr. Hanna Moser and Prof. Dr. Gudrun Socher for their guidance in refining the research question. An AI assistant (ChatGPT, GPT-5.1 Thinking) was used only for language polishing. Conceptual decisions, interpretation of the literature and the final research design are independent of the AI assistant.

This work forms part of the main seminar on human–agent collaboration at Munich University of Applied Sciences.

References

[1] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. arXiv:2308.14508 [cs.CL] https://arxiv.org/abs/2308.14508

[2] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient Attentions for Long Document Summarization. arXiv:2104.02112 [cs.CL] https://arxiv.org/abs/2104.02112

[3] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. arXiv:1702.08734 [cs.CV] https://arxiv.org/abs/1702.08734

[4] Taejong Joo, Shu Ishida, Ivan Sosnovik, Bryan Lim, Sahand Rezaei-Shoshtari, Adam Gaier, and Robert Giaquinto. 2025. Graph of Agents: Principled Long Context Modeling by Emergent Multi-Agent Collaboration. *arXiv preprint* (2025). arXiv:2509.21848 [cs.CL] https://arxiv.org/abs/2509.21848

[5] Aimée A. Kane, Susannah B. F. Paletz, Madeline Diep, Alexander Hajkowski, and Adam A. Porter. 2025. Virtual Collaborative Analysis: Effects of Two AI Summarizers. *Small Group Research* (2025). doi:10.1177/10464964251361563 Advance online publication.

[6] 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. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401 [cs.CL] https://arxiv.org/abs/2005.11401

[7] Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, Miami, Florida, USA, 881–893. https://aclanthology.org/2024.emnlp-industry.66

[8] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013/

[9] Julie Michelman, Nasrin Baratalipour, and Matthew Abueg. 2025. Enhancing Reasoning with Collaboration and Memory. *arXiv preprint* (2025). arXiv:2503.05944 [cs.CL] https://arxiv.org/abs/2503.05944

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2383–2392. doi:10.18653/v1/D16-1264

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[12] Lukas Röß. 2025. Human-Agent Collaboration Benchmark Suite. GitHub repository. https://github.com/luroess/human-agent-collaboration

[13] Mohammed Saqr, Kamila Misiejuk, and Sonsoles López-Pernas. 2025. Human-AI Collaboration or Obedient and Often Clueless AI in Instruct, Serve, Repeat Dynamics? *arXiv preprint* (2025). arXiv:2508.10919 [cs.HC] https://arxiv.org/abs/2508.10919

[14] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv:2104.08663 [cs.IR] https://arxiv.org/abs/2104.08663

[15] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. doi:10.18653/v1/N18-1074

[16] Kristy Wedel. 2025. Contextual Memory Intelligence: A Foundational Paradigm for Human-AI Collaboration and Reflective Generative AI Systems. *arXiv preprint* (2025). arXiv:2506.05370 [cs.AI] https://arxiv.org/abs/2506.05370

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. doi:10.18653/v1/2020.emnlp-demos.6

[18] Sibo Xiao, Zixin Lin, Wenyang Gao, and Yue Zhang. 2025. Long Context Scaling: Divide and Conquer via Multi-Agent Question-driven Collaboration. *arXiv preprint* (2025). arXiv:2505.20625 [cs.CL] https://arxiv.org/abs/2505.20625

[19] Gongbo Zhang, Zihan Xu, Qiao Jin, Fangyi Chen, Yilu Fang, Yi Liu, Justin F. Rousseau, Ziyang Xu, Zhiyong Lu, Chunhua Weng, and Yifan Peng. 2025. Leveraging Long Context in Retrieval Augmented Language Models for Medical Question Answering. *npj Digital Medicine* 8, 239 (2025). doi:10.1038/s41746-025-01651-w

[20] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. 2024. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. In *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper_files/paper/2024/file/ee71a4b14ec26710b39ee6be113d7750-Paper-Conference.pdf NeurIPS 2024.

[21] Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A Survey on the Memory Mechanism of Large Language Model-based Agents. *ACM Transactions on Information Systems* 43, 6 (2025), 1–45. doi:10.1145/3748302

[22] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. arXiv:2104.05938 [cs.CL] https://arxiv.org/abs/2104.05938

[23] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911 [cs.CL] https://arxiv.org/abs/2311.07911