Can an LLM Induce a Graph? Investigating Memory Drift and Context Length

Raquib Bin Yousuf, Aadyant Khatri, Shengzhe Xu, Mandar Sharma, Naren Ramakrishnan
Department of Computer Science, Virginia Tech, Alexandria, VA
Email: raquib@vt.edu, naren@cs.vt.edu

***Abstract*—Recently proposed evaluation benchmarks aim to characterize the effective context length and the forgetting tendencies of large language models (LLMs). However, these benchmarks often rely on simplistic "needle in a haystack" retrieval or continuation tasks that may not accurately reflect the performance of these models in information-dense scenarios. Thus, rather than simple next token prediction, we argue for evaluating these models on more complex reasoning tasks that requires them to induce structured relational knowledge from the text - such as graphs from potentially noisy natural language content. While the input text can be viewed as generated in terms of a graph, its structure is not made explicit and connections must be induced from distributed textual cues, separated by long contexts and interspersed with irrelevant information. Our findings reveal that LLMs begin to exhibit memory drift and contextual forgetting at much shorter effective lengths when tasked with this form of relational reasoning, compared to what existing benchmarks suggest. With these findings, we offer recommendations for the optimal use of popular LLMs for complex reasoning tasks. We further show that even models specialized for reasoning, such as OpenAI o1, remain vulnerable to early memory drift in these settings. These results point to significant limitations in the models' ability to abstract structured knowledge from unstructured input and highlight the need for architectural adaptations to improve long-range reasoning. Our codebase to support reproducibility is publicly available.[1].**

***Index Terms*—benchmark, evaluation, contextual forgetting, context length, memory drift**

I. Introduction

Recent benchmarks for evaluating LLMs have made significant progress in measuring context length and memory retention [1]–[3]. However, many of these evaluations rely on highly synthetic tasks, such as "needle-in-a-haystack" retrieval [4]–[6] or shallow continuation [7], which do not reflect the kinds of structured reasoning and information integration required in practical applications. While several of these works acknowledge the limitations of such tasks, particularly in capturing realistic comprehension or reasoning demands [4], [6], [7], they still fall short of evaluating whether a model can *induce latent structure* from long and noisy text.

In contrast, real-world reasoning often requires connecting entities and events that are scattered across large, unstructured documents [8]–[10]. The relevant relationships are rarely local or explicit, but must be inferred from distributed and indirect cues. Whether in scientific literature review, legal understanding, intelligence analysis, or medical report comprehension, effective reasoning involves recovering sparse relational knowledge embedded within irrelevant content. Evaluating models only on token-level recall or completion fails to capture this challenge.

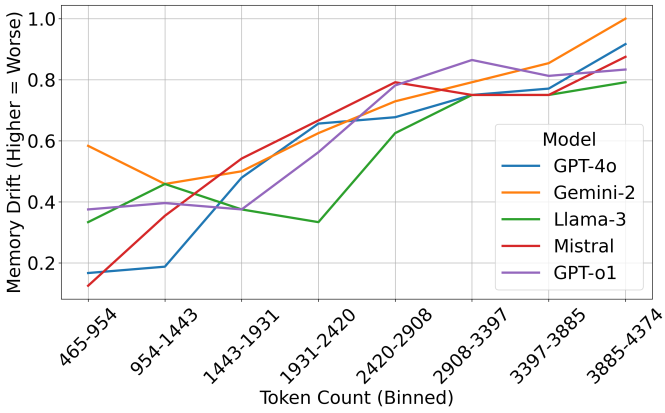


Fig. 1: Memory drift (lower = better) on the simplest relational task (one connection per sample). Despite minimal complexity, all models degrade beyond a certain context length, showing that even low relational load challenges long-context reasoning. See Table III for TL;DR.

In this work, we argue that the effective context length and the forgetting tendencies of LLMs should be evaluated based on their ability to recover *relational graphs* from natural language. These graphs encode semantic connections between entities, events, or concepts, and serve as a cognitively aligned abstraction of real-world information needs. Crucially, the graph structure is not provided directly, and must be inferred from paraphrased, interleaved, and noisy textual descriptions.

To this end, we introduce a new benchmark for evaluating the effective context length and the forgetting tendencies of LLMs centered on **graph reconstruction from noisy text**. Given a long input that implicitly encodes a hidden graph, the model must identify the correct nodes and their pairwise relations. We systematically control two axes of difficulty: (i) *contextual separation*, which measures how far apart related entities appear in the prompt, and (ii) *relational density*, which quantifies the number of connections the model must recover. These controls allow us to probe how models degrade under increased memory stress and structural complexity.

Our empirical findings reveal several consistent patterns in how large language models handle long-context relational reasoning:

[1] https://github.com/DiscoveryAnalyticsCenter/MemoryDrift

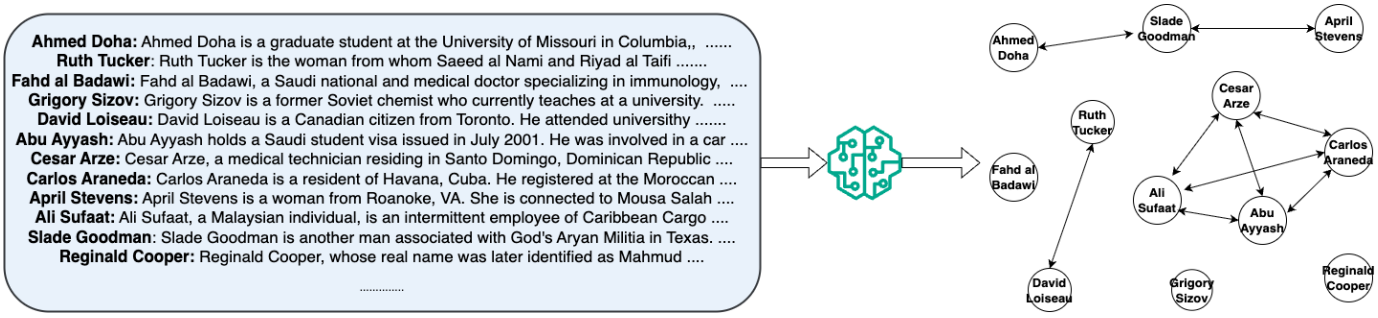


Fig. 2: Overview of our task: given a long, noisy text (left), the model (center) reconstructs the underlying relational graph (right) by identifying connections between entities (edge, subgraph, clique). Disconnected nodes are distractors.

- **Onset of memory drift at shorter effective context lengths.** Across models, performance degradation begins well before the maximum supported context window, with measurable declines in structural recovery observed as the length and complexity of input increases.
- **Recall is often the limiting factor in relational reasoning.** Many models favor high-precision extraction strategies, resulting in a tendency to miss valid connections rather than produce spurious ones. This conservative behavior becomes more pronounced as context length grows.
- **Greater structural complexity amplifies degradation.** As the density of relationships or the number of connections within each prompt increases, all models display declining performance, indicating heightened sensitivity to relational complexity in long-context scenarios.
- **Prompting strategies such as chain-of-thought do not mitigate these challenges.** Experiments with different prompting styles, including chain-of-thought reasoning, show little or no improvement for long-context graph reconstruction tasks and can, in some cases, worsen results due to increased distraction.
- **A reasoning-specialized model does not overcome early memory drift.** In our evaluation, a model specifically designed for advanced reasoning also exhibited early onset of memory drift, similar to general-purpose language models. This suggests that, at least for the tested model, advanced reasoning capabilities alone are insufficient to address the core limitations of long-context relational reasoning.

These results highlight persistent brittleness in current LLMs when faced with long-context relational reasoning, particularly when structural recovery is required under dispersion, density, and noise. Recovering graph structure from noisy text is substantially more demanding than next-token prediction or span-level retrieval. These observations underscore the need to move beyond generic context-length benchmarks toward task-specific evaluations that reflect structured reasoning demands. As summarized in Table III, models vary significantly in how they balance precision, recall, and memory stability, which emphasizes the importance of targeted model selection for real-world applications. Our key contributions are:

- We design a graph reconstruction tasks for LLMs, consisting three subtasks, edge recovery, subgraph discovery, and clique detection, that probe a model's ability to induce structure under dispersion and noise.
- We propose *memory drift*, a metric that captures forgetting and hallucination as a function of context length and relational complexity.
- We systematically evaluate five popular LLMs (GPT-4o [11], OpenAI o1 [12], [13], Gemini-2 [14], Llama-3 [15], Mistral-7B [16]), revealing earlier and sharper degradation than suggested by existing benchmarks, especially under high information density.
- We release our codebase to support reproducible analysis of long-context reasoning via structured knowledge extraction.

II. Background and Related Work

A. Context Length and Memory in Language Models

Recent years have seen rapid progress in the evaluation of large language models (LLMs) on long-context understanding. Several benchmarks have sought to quantify the effective memory and forgetting behavior of LLMs using controlled experimental designs. For instance, the Forgetting Curve [7], Same Task, More Token [2], One Thousand and One Pairs [1], and Ruler [4] benchmarks probe the extent to which LLMs can retrieve information or maintain associations over increasing input lengths.

However, these efforts primarily use synthetic or simplified tasks, and often do not reflect model performance in settings where relevant information is sparsely distributed or interleaved with distractors. The true extent of early memory drift and contextual forgetting in more complex, relational reasoning tasks remains underexplored, motivating our present study.

B. Relational Reasoning with LLMs

LLMs have progressed from basic text generation [17], [18] to complex applications involving chat agents [19], multi-agent simulation [20], and scientific reasoning [21]. Structured information extraction and relational reasoning have become increasingly important, particularly in domains such

as intelligence analysis [9], [22], where models must recover key relationships embedded in lengthy, noisy text.

A core challenge in these applications is identifying salient clues and mapping entity relationships across large, unstructured inputs [23]–[25]. Recent approaches leverage tool augmentation or retrieval [26]–[30] to supplement the model's latent knowledge. Other work explores the capacity of LLMs to maintain organized memory structures with or without external augmentation [9].

Despite these advances, existing evidence suggests that even advanced LLMs struggle with long-context relational reasoning, especially when structural cues are dispersed or implicit. In intelligence analysis tasks, for example, effective memory length may be even shorter than in conventional text benchmarks [1], [2], [4], [7], and the ability to induce latent graph structure remains a significant limitation.

C. Relation to Entity Linking, Coreference Resolution, and Knowledge Base Construction

While our benchmark is superficially related to traditional NLP tasks such as entity linking, coreference resolution, relation extraction, and knowledge base construction (KBC), there are fundamental differences in both objective and methodological focus. In recent years, large language models (LLMs) have been increasingly applied to these classical tasks [31]–[42]. Entity linking, coreference resolution (mention clustering/anaphora), and relation extraction have all seen improvements from generative modeling, instruction tuning, and prompt-based LLMs. Some recent studies have further explored the limitations of LLMs with respect to context length and memory for these tasks, particularly as applied to longer documents or document-level extraction [35], [41], [43], [44].

However, the majority of this prior work continues to focus on local mention disambiguation, anaphora resolution, or extraction of predefined relation types, typically in settings where relevant cues are assumed to co-occur or be easily retrievable within limited context windows. In contrast, our benchmark diverges in both its central aim and experimental design. We treat **relational graph reconstruction as a direct proxy for analyzing LLM memory, context length, and forgetting in information-dense, noisy settings**. Specifically, our task requires models to process extended and noisy input sequences, where relational cues are highly dispersed, indirect, and embedded within substantial irrelevant content. The model must encode and maintain distributed entity descriptions over long-range dependencies, and integrate these representations to induce latent connections, often separated by significant contextual distance or paraphrased evidence. This setting compels holistic, graph-level reasoning rather than isolated or span-local predictions.

Crucially, our benchmark is structured to push models to their effective memory and reasoning limits by (i) dispersing relational evidence across extended contexts, (ii) interleaving structurally irrelevant distractors, and (iii) increasing the density and granularity of latent relational structure that must be reconstructed jointly. These design choices go beyond the scope of existing information extraction or KBC benchmarks and provide a more rigorous test of long-context reasoning.

Taken together, while recent advances have extended the reach of LLMs in structured information extraction and relational reasoning, existing benchmarks do not adequately capture the challenges posed by long, noisy contexts where latent structure must be recovered globally. To fill this gap, we introduce **relational graph reconstruction** as a general probe of long-context reasoning and memory in LLMs, enabling systematic evaluation of their ability to integrate dispersed, indirect, and noisy relational cues across extended input sequences.

III. Graph Reconstruction as a Lens on Long-Context Reasoning

Despite recent progress in information extraction and reasoning, it remains unclear whether LLMs can integrate and recover latent relational structure from long, noisy, and unstructured inputs. We address this by treating graph reconstruction as a direct probe of long-context memory and reasoning in LLMs, using data inspired by real-world, intelligence-style reporting.

The core task of our benchmark is **relational graph reconstruction**. Given a long natural language input encoding a hidden graph, the model must recover this graph through structured prediction or post-hoc extraction. The key challenge lies in piecing together distributed cues that correspond to nodes and their connections. Natural language rarely encodes explicit graph edges. Instead, relations are embedded in distributed mentions, paraphrases, and disjoint spans. Moreover, real-world text contains distractors or irrelevant facts, entities, or events, which the model must learn to ignore. The presence of such noise compounds the difficulty of maintaining stable memory traces over long sequences.

Thus, with the above consideration, we evaluate three subtasks: **(i) Edge Discovery**, where the model recovers pairwise relations; **(ii) Subgraph Discovery**, where it identifies connected node subsets (e.g., stars, chains); and **(iii) Clique Discovery**, where it detects fully connected clusters.

A. Task Formulation

Let $\mathcal{G} = (V, E)$ be an undirected latent graph over a set of entities V, where each edge $(u, v) \in E$ represents a hidden semantic relation. Each node $v \in V$ has a corresponding natural language description $d_v \in \mathcal{D}$.

Input Construction: We define a prompt $\Pi = \{d_{v_1}, \ldots, d_{v_n}\}$ where $n = |\mathcal{C}| + |\mathcal{N}|$, composed of:

- $\mathcal{C}$: a set of connected components drawn from $\mathcal{G}$ (e.g., edges, stars, cliques),
- $\mathcal{N}$: a set of noise or distractor nodes such that $\forall u, v \in \mathcal{N}$, $(u, v) \notin E$.

Let $\texttt{disp} : \mathcal{C} \times \mathcal{N} \to \Pi$ be a dispersion function that interleaves elements of $\mathcal{C}$ into the distractor set, controlling their relative positions.

We define the token-level separation between two related entities $u, v \in \mathcal{C}$ in the prompt Π as:

$$\delta(u, v; \Pi) = \texttt{TokenDist}(d_u, d_v)$$

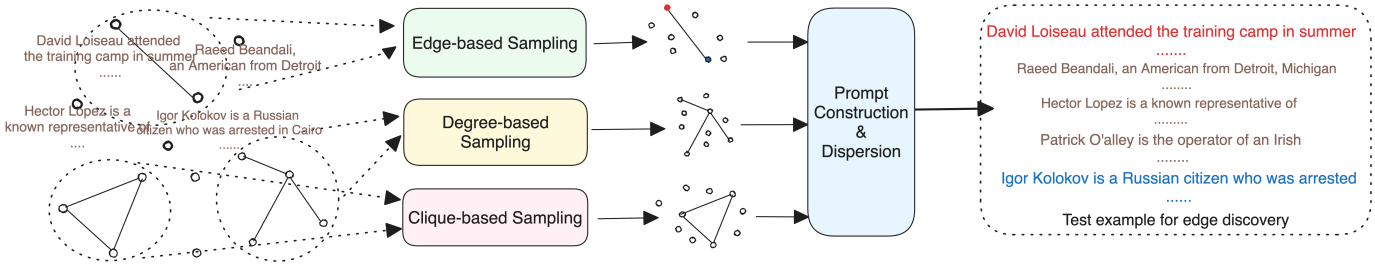


Fig. 3: Graph-to-prompt pipeline: relational structures (edges, stars, cliques) are sampled from a latent graph and interleaved with distractors to create dispersed, noisy prompts. Red and Blue are targets. Brown are the distractors.

TABLE I: Summary of graph-based tasks used in the benchmark. Each task varies in structural complexity and connection granularity.

Task Type	**Structure**	**Connection Size**	**Sampling Strategy**
Edge Discovery	Pairwise edges (u, v)	2 nodes per connection	Min-degree edge selection
Subgraph Discovery	Star-like subgraphs	$d+1$ nodes (degree-based)	Min-degree neighborhood centered on node
Clique Discovery	Fully connected cliques	k nodes per clique	Min-degree clique removal

and use this to measure contextual dispersion or memory stress.

Language Model Objective: Let $f_\theta : \Pi \to \hat{\mathcal{G}} = (\hat{V}, \hat{E})$ be the output of the language model, where the goal is to recover:

$$\hat{\mathcal{G}} \approx \mathcal{G}[\mathcal{C}]$$

i.e., reconstruct the subgraph induced by the true connected components.

Evaluation: We evaluate predictions using a graph-level reconstruction loss:

$$\mathcal{L}(\hat{\mathcal{G}}, \mathcal{G}[\mathcal{C}]) = \texttt{F1}_{\text{edge}}(\hat{E}, E[\mathcal{C}])$$

alongside *memory drift* metrics based on $\delta(u, v)$ and performance degradation across increasing distance.

Benchmark Variants: We instantiate this task under three structural regimes:

- **Edge Discovery:** $\mathcal{C} = \{(u_i, v_i)\}_{i=1}^{k}$ where each $(u_i, v_i) \in E$.
- **Subgraph Discovery:** $\mathcal{C}$ contains k star-like subgraphs of size $d+1$.
- **Clique Discovery:** $\mathcal{C}$ contains k cliques of size k such that all $(u, v) \in \mathcal{C}$ satisfy $(u, v) \in E$.

B. Graph Sampling Strategies

1) Edge-Based Sampling (Edge Discovery): For this case, we isolate individual pairwise relations for evaluation. Each test instance is constructed by identifying edges that lie near the periphery of the graph, where nodes are less embedded within dense neighborhoods. Formally, we define the priority of an edge (u, v) by the combined degree of its endpoints: $\deg(u, v) = \deg(u) + \deg(v)$. Edges with the smallest such values are selected first, under the assumption that they are less likely to overlap with other relational structures. To prevent redundancy or relational leakage, both the selected edge and its immediate neighbors are excluded from further sampling. This enforces disjointness and ensures that connections are spatially isolated within the graph topology.

Remaining nodes that are no longer part of any edge structure are repurposed as distractors. These nodes do not participate in relational content relevant to the task, and serve as negative examples. This setting provides a base case for evaluating whether models can recover simple binary relations when embedded in distractive and unstructured input.

2) Degree-Based Sampling (Subgraph Discovery): To examine a model's ability to recover higher-order structure, we sample local neighborhoods centered around nodes of fixed degree. These subgraphs are star-like, consisting of a central node and its d immediate neighbors, forming an induced subgraph of size $d+1$. Each candidate neighborhood is scored based on the total degree of its closed neighborhood, i.e., $\sum_{u \in N[v]} \deg(u)$; where $N[v]$ is the set containing v and all nodes adjacent to it. Lower-scoring neighborhoods are prioritized, under the hypothesis that they are less entangled within the larger graph and more likely to be topologically separable.

Once a subgraph is selected, its constituent nodes and all adjacent nodes are removed to maintain disjointness between samples. The remaining portion of the graph is explicitly disconnected, eliminating residual connectivity and yielding a pool of structurally neutral distractors. This task setup probes whether a model can identify subgraphs with coherent internal structure when they are distributed among unrelated textual descriptions.

3) Clique-Based Sampling (Clique Discovery): The most structurally demanding variant of our benchmark targets the recovery of cliques: fully connected subgraphs where each node shares an edge with every other node in the group. A clique of size k satisfies $(u, v) \in E$ for all $u, v \in C$, where $C = \{v_1, \ldots, v_k\}$. Such substructures require the model to integrate multiple overlapping relations simultaneously.

We restrict our attention to cliques that are not only maximal but also situated in sparsely connected regions. Each candidate clique is scored by the aggregate degree of its nodes, i.e.,

$\sum_{i=1}^{k} \deg(v_i)$. Those with lower scores are preferred, as they are more likely to be separable from the rest of the graph. After sampling, the clique and its surrounding neighborhood are removed to prevent structural overlap between samples.

Distractors are drawn from the remaining graph, which are forcibly pruned to remove all residual edges. The result is a controlled environment in which the only coherent structure is the clique itself. This setting evaluates the model's ability to recognize dense and mutually entangled relational clusters within noisy, otherwise unstructured contexts. Algorithm 1 shows a generalized version of three different sampling techniques.

Algorithm 1 General Subgraph-Based Sampling

Require: Graph $G = (V, E)$; selector type Type (e.g., EDGE, CLIQUE, DEGREE); parameter p (e.g., k or d)
Ensure: $\mathcal{C}$: selected subgraphs, $\mathcal{D}$: disconnected nodes
$\mathcal{C} \leftarrow [\,]$; $\mathcal{D} \leftarrow [\,]$; $G' \leftarrow G$
while there are valid units of type Type in G' **do**
  if Type is EDGE **then**
    Select $(u, v) \leftarrow \arg\min_{(i,j)\in E(G')} \deg(i) + \deg(j)$
    $U \leftarrow \{u, v\}$
  else if Type is CLIQUE **then**
    Find all cliques $\mathcal{Q} = \{C \subseteq V(G') \mid |C| = p, C \text{ is clique}\}$
    $U \leftarrow \arg\min_{C\in\mathcal{Q}} \sum_{v\in C} \deg(v)$
  else if Type is DEGREE **then**
    $N_p \leftarrow \{v \in V(G') \mid \deg(v) = p\}$
    $v^* \leftarrow \arg\min_{v\in N_p} \sum_{u\in N[v]} \deg(u)$
    $U \leftarrow N[v^*]$
  end if
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{U\}$
  $R \leftarrow N[U]$ ▷ closed neighborhood of U
  Remove R from G'
  $\mathcal{D} \leftarrow \mathcal{D} \cup V(G')$
end while
return $\mathcal{C}, \mathcal{D}$

Algorithm 2 Prompt Test Case Generation with Dispersion

Require: Profiles $\mathcal{P}$, connection dict $\mathcal{C}$, params $\Theta = \{k, n, s, e\}$, tokenizer $\mathcal{T}$
Ensure: Test cases $\mathcal{T}_{\text{cases}}$
$\mathcal{T}_{\text{cases}} \leftarrow [\,]$
for $i = 1$ to N **do**
  Sample $C = \{c_1, \ldots, c_k\}$ from $\mathcal{C}$
  Sample $D \subset V$, $|C| + |D| = n$
  Partition D into k segments in $[s \cdot |D|, e \cdot |D|]$
  Interleave each c_j into segment j of D to form L
  $\Pi \leftarrow \texttt{Concat}(\mathcal{P}[x] \mid x \in L)$
  Compute $\delta = \texttt{token\_dist}(c_1, c_k, \mathcal{T})$
  Store $\{L, \Pi, \delta\}$ in $\mathcal{T}_{\text{cases}}$
end for
return $\mathcal{T}_{\text{cases}}$

C. Prompt Construction

To systematically evaluate long-context relational reasoning, we construct a controlled benchmark that synthesizes input prompts containing both relational and distractor entities. Each test case is generated through a three-stage pipeline: (i) graph sampling to define ground-truth connections, (ii) controlled instantiation of test prompts with a mix of connected and disconnected entities, and (iii) spatial dispersion of related components to simulate contextual separation.

Given a curated entity graph and corresponding textual profiles, we generate a set of test instances tailored to each task type—edge discovery, subgraph recovery, or clique detection. The prompt generation process samples structured relational subsets and interleaves them with distractor nodes, allowing us to modulate both structural complexity and memory stress.

For edge discovery, each test case includes a fixed number of connected pairs sampled from the graph's edge set. Distractors are drawn from two sources: (a) explicitly disconnected nodes, and (b) unused nodes from the relational pool, with at most one node per unused pair. This guarantees a consistent number of entities per prompt while preserving topological separation between connected and distractor elements.

For subgraph and clique discovery tasks, we first sample tuples of higher cardinality. These are drawn from the graph according to degree-based or clique-specific criteria, ensuring that each tuple forms a valid induced substructure. The number of nodes per substructure is varied within a defined range, and samples are stratified by structural regime (e.g., stars or cliques). Each test case includes a balanced mixture of such substructures and distractors, with a fixed total entity budget.

Finally, for every sampled prompt, we compute the adjacency matrix of the ground-truth subgraph to enable evaluation. The relative placement of connected entities within the distractor pool is explicitly controlled to vary contextual separation and dispersion, allowing us to probe the model's ability to integrate non-local relational cues. Algorithm 2 shows the inner-works of prompt generation technique.

D. Dataset Construction and Model Tested

Motivated by recent research on the real-world deployment of LLMs in complex downstream reasoning tasks [9], [22], we construct a benchmark derived from two classical synthetic intelligence analysis challenge datasets: Sign of the Crescent and Atlantic Storm [45]. These datasets, originally developed for analyst training, have since become classical benchmarks widely used in analytics competitions and as evaluation datasets in the intelligence analysis and visual analytics research communities [46]–[49]. Each collection offers rich, narrative-style descriptions of individuals, designed to emulate the complexity and ambiguity of real-world intelligence scenarios.

Each data point corresponds to a unique person, represented through a short paragraph containing biographical and contextual details. Entities are implicitly linked through shared activities, affiliations, or co-occurrences, forming the basis of an underlying latent graph. To establish the ground truth, two annotators manually verified all relational connections between individuals, resulting in a curated graph structure used for evaluation.

We experiment with five models from different model families: i) GPT-4o [11] from OpenAI, ii) OpenAI o1 [12],

[13], iii) Llama-3 [15] from Meta AI, iv) Mistral-7B [16] from MistralAI, and v) Gemini-2 [14] from Gemini platform, Google.

E. Can LLMs Recover Structure in Short Contexts?

We begin by evaluating whether LLMs can recover latent graph structure from natural language prompts under idealized conditions, where entities are nearby and context length is short. As shown in Figure 4, GPT-4o and Gemini-2 achieve high precision but only moderate recall and F1. This indicates that the models avoid hallucinating structure, but generaly fail to retrieve many true connections. While partial structural reasoning is evident, complete graph reconstruction remains elusive even in low-memory-stress settings.

We now turn to more challenging conditions, where increased context length, dispersion, and structural density begin to degrade recovery.

IV. Memory Drift and Degradation

To assess a model's ability to recover relational structure from noisy input, we report standard edge-level retrieval metrics, *precision*, *recall*, and *F1 score* alongside a primary diagnostic measure we call *memory drift*. While the former reflect local prediction quality, memory drift is designed to capture global performance degradation under increasing context length and relational complexity.

Memory Drift quantifies deviation from ideal relational reconstruction using a weighted sum of true positives (TP), false positives (FP), and false negatives (FN), ignoring true negatives due to their overwhelming presence and low informativeness. The metric is defined as:

$$\text{Memory Drift} = 1 - \max\left(0,\ \frac{w_{\text{TP}} \cdot \text{TP} \ +\ w_{\text{FP}} \cdot \text{FP} \ +\ w_{\text{FN}} \cdot \text{FN}}{2P}\right)$$

where P is the number of gold-standard edges in the prompt. The weights $w_{\text{TP}} = 2$, $w_{\text{FP}} = -0.5$, and $w_{\text{FN}} = -1.0$ reflect our view that forgetting (missed edges) is more damaging than hallucination (spurious edges), though both degrade structural integrity. The $\max(0, \cdot)$ clamp ensures that large numbers of errors do not yield negative values. This formulation produces a bounded value in $[0, 1]$, where 0 indicates perfect structural recovery and 1 indicates maximal degradation. To build intuition for how memory drift behaves in practice, Table II shows example predictions with varying combinations of true positives, false positives, and false negatives. These examples illustrate how memory drift increases as models forget edges, hallucinate new ones, or both, even when standard precision and recall appear reasonable.

Importantly, *memory drift is not equivalent to recall*. A model may exhibit reasonable recall but still show high drift if it frequently introduces incorrect structure. By incorporating both types of errors, the metric captures a broader notion of degradation relevant to downstream reasoning tasks.

To support interpretability, we also report standard metrics:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

These standard retrieval metrics allow us to disentangle false positives and false negatives. All four measures are tracked across token lengths and connection densities to characterize long-context relational reasoning.

A. When Does Memory Drift Begin?

We analyze how structural recovery degrades with increasing input length. Memory drift refers to this performance decay as related entities become more contextually separated.

As shown in Figure 6a for GPT-4o and Figure 6b for Gemini-2, **memory drift increases sharply after 2000 tokens across all connection densities**. Low-density prompts suffer the fastest degradation, while higher-density ones appear more stable but begin with worse initial performance.

Figure 5a for GPT-4o shows that degradation is driven by declining recall, while precision remains stable. This suggests that models increasingly miss true edges rather than hallucinating new ones. The onset of memory drift at around 2000 tokens marks a practical upper bound on effective context for relational reasoning. However, Gemini-2 shows a more balanced behavior but still suffers high memory drift after 2000 tokens.

B. Does Information Density Help or Hurt?

Information density, measured by the number of connections per sample, negatively affects model performance. As density increases, the task becomes more difficult due to greater relational complexity.

Both Figure 6 and Figure 4 show that high-density prompts begin with lower F1 score and higher memory drift, even at short context lengths. This indicates that models struggles to recover dense structures regardless of token count. The effect compounds over longer contexts, but the primary impact is already visible in the initial token bins. The degradation for higher density prompts is more prominent in GPT-4o. Gemini-2 shows a more stable behavior for higher density prompts.

C. Hallucination vs Forgetting

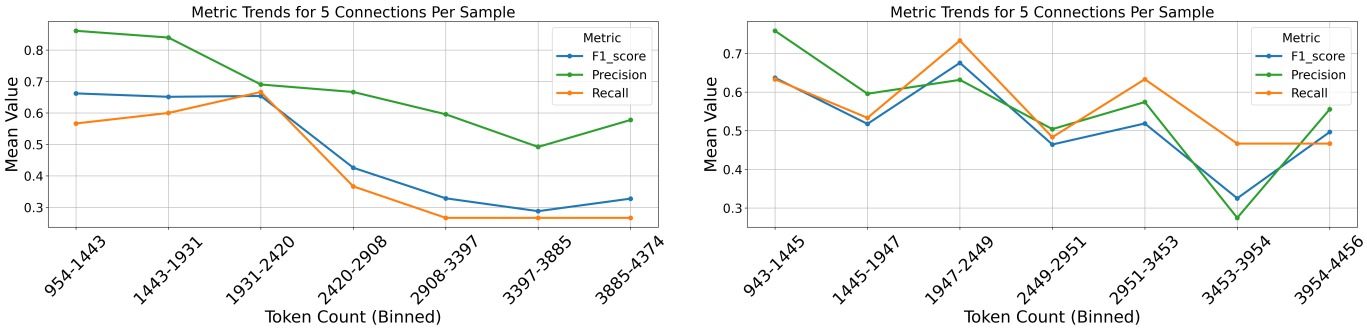
We assess how models balance false positives and false negatives under increasing context length and connection density. Specifically, we examine whether performance degradation is driven by hallucinated edges (low precision) or missed ones (low recall).

As shown in Figure 4a, GPT-4o maintains consistently high precision across all token bins, even as recall decline sharply with longer contexts. This pattern holds across densities and is further supported by the radar plot in Figure 5a, where precision dominates all other metrics.

These results suggest that the **model adopts a conservative prediction strategy**. It prefers to omit uncertain connections rather than risk false positives. Hallucination remains rare, even in dense or noisy prompts, and does not increase with token count. Instead, most errors arise from failure to recover true edges, particularly under dispersion and structural complexity.

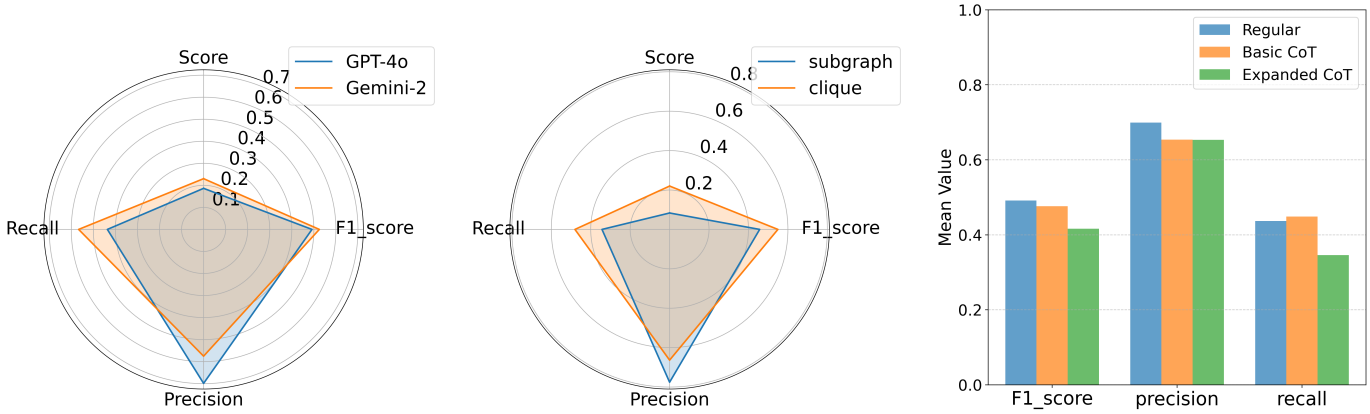
TABLE II: Examples of the memory drift metric under different graph reconstruction scenarios. Drift penalizes both hallucinations (FP) and forgetting (FN), offering a more comprehensive signal than standard precision and recall alone.

Example	Gold Edges	Predicted Edges	TP	FP	FN	Precision	Recall	Memory Drift
Perfect (0.00)	{(A,B), (B,C)}	{(A,B), (B,C)}	2	0	0	1.00	1.00	0.00
Mid-case (0.50)	{(A,B), (B,C), (C,D)}	{(A,B), (C,D)}	2	0	1	1.00	0.67	0.50
Balanced (0.75)	{(A,B), (B,C), (C,D), (D,E)}	{(A,B), (C,D)}	2	0	2	1.00	0.50	0.75
Hallucinated (0.875)	{(A,B), (B,C)}	{(A,B), (A,C)}	1	1	1	0.50	0.50	0.875
None (1.00)	{(A,B), (B,C)}	{}	0	0	2	0.00	0.00	1.00



(a) GPT-4o (b) Gemini-2

Fig. 4: Metric trend with GPT-4o and Gemini-2 on edge discovery



(a) GPT-4o and Gemini-2 (edge) (b) GPT-4o(subgraph and clique) (c) Different prompting style

Fig. 5: Metrics for different model and discovery cases, "Score" corresponds to $(1 - \text{Memory Drift})$, allowing comparison with precision, recall, and F1.

In summary, hallucination is not the dominant failure mode in long-context relational reasoning. The model prioritizes precision at the expense of recall, leading to underprediction rather than overgeneration.

D. Does Chain-of-thought (CoT) Prompting Help?

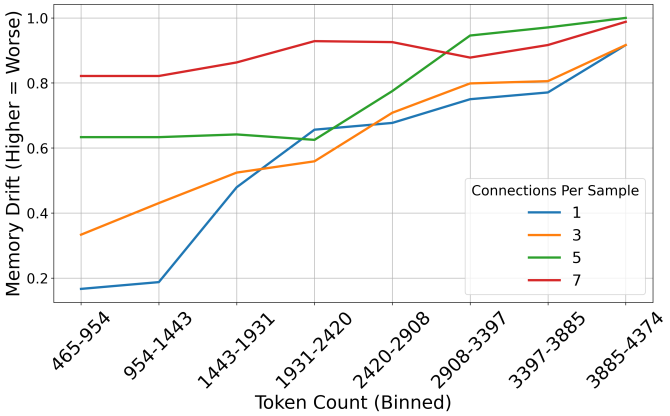
We test whether Chain-of-Thought (CoT) [50] prompting improves structural extraction by comparing regular prompting, basic CoT, and expanded CoT on the edge discovery task. As shown in Figure 5c for GPT-4o, both CoT variants underperform the regular strategy across all key metrics. Expanded CoT performs the worst, with notably lower score, recall, and F1. Basic CoT shows slightly better recall than regular, but at the cost of reduced overall accuracy.

We conclude that **CoT prompting is not helpful for this task**. For relational recovery in long contexts, simple prompting remains the most effective approach.

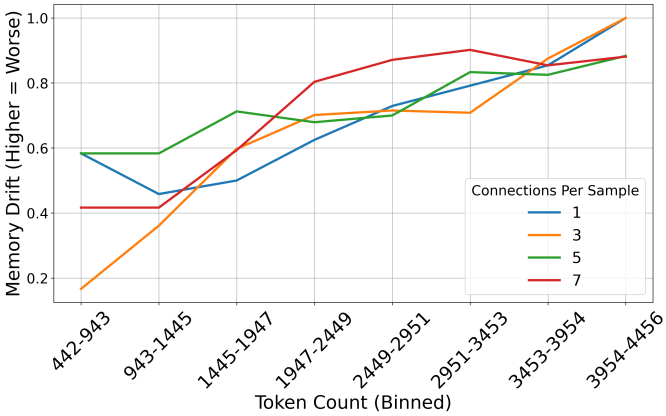
E. LLMs' Behavior across Different Graph Discovery Subtasks

We compare GPT-4o's performance across the three relational reasoning subtasks: edge discovery, subgraph (degree-based) discovery, and clique discovery. These differ in structural complexity and the type of inductive reasoning required.

Figure 5a and 5b show metric profiles for each task under the same connection density (5 connections per sample). Edge recovery yields the highest precision but lowest recall, suggesting that the model avoids hallucination but misses many valid edges. Degree-based subgraph discovery shows more

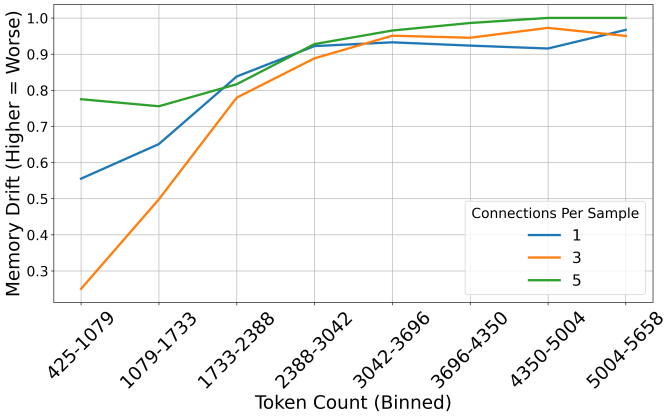


(a) Memory drift for GPT-4o

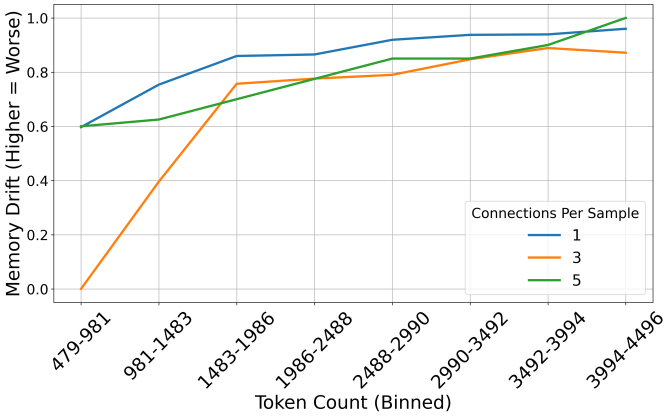


(b) Memory drift for Gemini-2

Fig. 6: Memory drift on edge discovery for different models



(a) Memory drift on subgraph discovery



(b) Memory drift on clique discovery

Fig. 7: Memory drift for GPT-4o on two different subtasks

TABLE III: Model comparison across long-context relational reasoning benchmarks.

Capability	GPT-4o	Gemini-2	Llama-3	Mistral-7B
Memory Drift Onset	∼2000 tokens	∼2500 tokens	∼1800 tokens	<1000 tokens
Precision	Very High (stable)	Moderate (fluctuating)	High (slightly unstable)	Low (inconsistent)
Recall	Low (declines early)	High (adaptive)	Moderate (variable)	Low (noisy)
F1 Score Stability	Moderate (recall-limited)	Balanced (peaks mid-context)	Moderate (flat)	Unstable
Density Robustness	Poor beyond 5 connections	Robust up to 7 connections	Moderate (degrades after 5+)	Fails even at 3+ connections
Prediction Strategy	Conservative (high precision)	Balanced (recall-oriented)	Slightly conservative	Inconsistent
Best Use Case	Precision-critical tasks	Broad structure recovery	General-purpose tasks	Lightweight/fine-tuned use

balanced recall and precision, though overall scores remain low. Clique recovery exhibits the highest recall and F1, but struggles with precision due to the combinatorial challenge of predicting fully connected structures.

Memory drift curves (Figure 6a, 7a, and 7b) reveal that drift patterns vary by task. Edge and degree-based subtasks degrade quickly beyond 2000–2500 tokens, while clique recovery declines more slowly, likely due to redundancy among densely connected entities. These findings highlight that task structure strongly affects how LLMs handle long-context reasoning. Sparse graphs lead to brittle recall, dense graphs trigger hallucination risk, and mid-level structures offer moderate balance but still degrade under dispersion.

F. Recommendations for Different LLMs

Our evaluation reveals that different models exhibit distinct tradeoffs in how they handle long-context relational reasoning. Table III summarizes these trends across key behavioral axes, including memory drift onset, precision-recall balance, and robustness to relational density. While no model is uniformly superior, each demonstrates strengths in specific regimes. Further figures and details are available in our codebase. Based on these observations, we offer the following practical

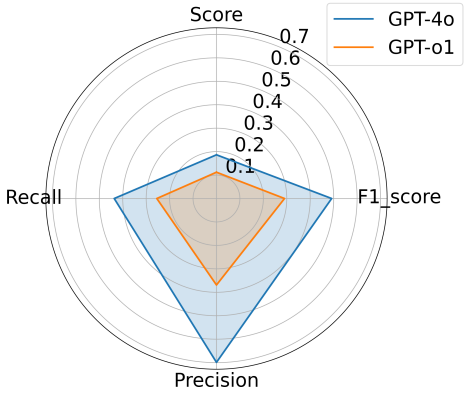


Fig. 8: Metrics for GPT-4o and o1 (edge) cases, "Score" corresponds to $(1 - \text{Memory Drift})$, allowing comparison with precision, recall, and F1.

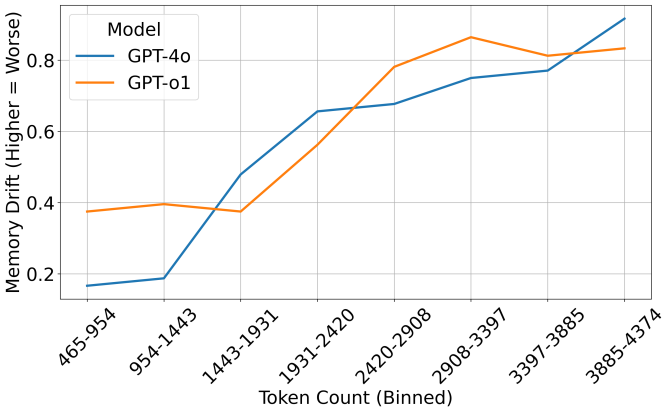


Fig. 9: Memory drift GPT-4o and o1 (edge) case

recommendations for model selection, tailored to the needs of different real-world use cases:

- **GPT-4o** is well-suited for high-precision tasks where hallucination must be minimized, such as legal, intelligence vetting, or safety critical settings. However, it sacrifices recall as context length and complexity grow.
- **Gemini-2** is ideal for exploratory tasks that prioritize coverage, such as the construction of a knowledge graph or intelligence discovery. It shows high recall and better robustness under structural noise.
- **Llama-3** offers strong precision with moderate recall, making it suitable for general-purpose long-context reasoning. It handles moderate relational density well but begins to degrade with structural complexity and extended dispersion.
- **Mistral-7B** struggles in zero-shot relational recovery but may serve as a lightweight base for fine-tuning or retrieval-augmented pipelines, particularly when resources are constrained.

G. Can Reasoning-Oriented Models Like OpenAI o1 Overcome Memory Drift?

OpenAI o1 does not outperform general-purpose models on memory drift, and suffers similar or worse degradation as context length increases.

We assess whether reasoning-specialized models, such as o1 [12], [13], are more robust to memory drift on long-context relational reasoning tasks.

As shown in Figure 1, which presents memory drift for all models, o1 performs comparably to GPT-4o at short context lengths. However, as the token count increases, o1's memory drift rises steadily, matching or exceeding that of the other models. Notably, for the longest contexts, o1 does not outperform general-purpose models and even displays higher drift than GPT-4o and Llama-3 in several bins.

A direct comparison in Figure 9 (o1 vs. GPT-4o) shows that o1 remains similar to GPT-4o for shorter inputs but its memory drift surpasses GPT-4o for prompts longer than 2000 tokens, confirming that the onset of degradation is not delayed for reasoning-tuned models.

The radar plot in Figure 8 further illustrates this gap. o1 maintains reasonable precision but lags behind GPT-4o in both recall and F1, indicating that while it avoids hallucination, it fails to recover a significant portion of the true relational structure—especially as input length and complexity increase.

Overall, reasoning-oriented models like o1 do not overcome the limitations of memory drift or context fragmentation in long, noisy inputs. The results indicate that current advances in model reasoning are insufficient for reliable relational graph induction at scale.

V. Conclusion, Limitations, and Future Work

We introduced a benchmark for evaluating long-context reasoning in LLMs through the task of graph reconstruction from noisy text. Our results show that models degrade much earlier than their context limits suggest, especially under structural complexity and dispersion. The proposed memory drift metric offers a more accurate view of this degradation than standard retrieval metrics. Our benchmark reveals key tradeoffs across model families and provides a practical lens for assessing real-world reasoning capabilities. These findings offer concrete guidance for both model development and deployment in structure-sensitive applications.

Several limitations should be acknowledged. Our evaluation is limited to zero-shot and few-shot prompting, without exploring the effects of fine-tuning or retrieval-based approaches. We also recognize that prompt sensitivity is an important consideration and leave a more systematic study of this aspect to future work.

Looking ahead, we will investigate how retrieval-based and memory-augmented systems influence memory retention, forgetting, and drift in long-context relational reasoning. More broadly, we hope this benchmark and metric provide a valuable diagnostic for the nuanced failure modes of current LLMs, moving beyond "needle in a haystack" evaluations toward more realistic, structure-based reasoning tasks. By establishing new

evaluation settings and highlighting model-specific behaviors, we aim for this work to support continued advances in long-context understanding and structured reasoning.

Acknowledgment

This work is supported in part by US National Science Foundation grants CMMI-2240402 and IIS-2312794. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsor(s).

References

[1] M. Karpinska *et al.*, "One thousand and one pairs: A" novel" challenge for long-context language models," *arXiv preprint arXiv:2406.16264*, 2024.

[2] M. Levy, A. Jacoby, and Y. Goldberg, "Same task, more tokens: the impact of input length on the reasoning performance of large language models," *arXiv preprint arXiv:2402.14848*, 2024.

[3] M. Song, M. Zheng, and X. Luo, "Counting-stars: A multi-evidence, position-aware, and scalable benchmark for evaluating long-context large language models," *arXiv preprint arXiv:2403.11802*, 2024.

[4] C.-P. Hsieh *et al.*, "Ruler: What's the real context size of your long-context language models?" *arXiv preprint arXiv:2404.06654*, 2024.

[5] gkamradt, "gkamradt/LLMTest_needleinahaystack," Jul. 2024, original-date: 2023-11-11T00:50:02Z. [Online]. Available: https://github.com/gkamradt/LLMTest_NeedleInAHaystack

[6] M. Li, S. Zhang, Y. Liu, and K. Chen, "Needlebench: Can llms do retrieval and reasoning in 1 million context window?" *arXiv preprint arXiv:2407.11963*, 2024.

[7] X. Liu *et al.*, "Forgetting curve: A reliable method for evaluating memorization capability for long-context models," *arXiv preprint arXiv:2410.04727*, 2024.

[8] S. Shankar *et al.*, "Docetl: Agentic query rewriting and evaluation for complex document processing," *arXiv preprint arXiv:2410.12189*, 2024.

[9] R. B. Yousuf *et al.*, "Llm augmentations to support analytical reasoning over multiple documents," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 1892–1901.

[10] Y. Fan *et al.*, "Medodyssey: A medical domain benchmark for long context evaluation up to 200k tokens," *arXiv preprint arXiv:2406.15019*, 2024.

[11] A. Hurst *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[12] "Learning to Reason with LLMs." [Online]. Available: https://openai.com/index/learning-to-reason-with-llms/

[13] A. Jaech *et al.*, "Openai o1 system card," *arXiv preprint arXiv:2412.16720*, 2024.

[14] G. Team *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[15] A. Grattafiori *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[16] A. Q. Jiang *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[17] J. Achiam *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[18] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[19] J. S. Park *et al.*, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[20] G. Wang *et al.*, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[21] B. Romera-Paredes *et al.*, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, no. 7995, pp. 468–475, 2024.

[22] X. Tang *et al.*, "Steering llm summarization with visual workspaces for sensemaking," *arXiv preprint arXiv:2409.17289*, 2024.

[23] P. Lu *et al.*, "Chameleon: Plug-and-play compositional reasoning with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[24] T. Schick *et al.*, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[25] J. Zhang, "Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt," *arXiv preprint arXiv:2304.11116*, 2023.

[26] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[27] Z. Zhong, T. Lei, and D. Chen, "Training language models with memory augmentation," *arXiv preprint arXiv:2205.12674*, 2022.

[28] Y. Wang, P. Li, M. Sun, and Y. Liu, "Self-knowledge guided retrieval augmentation for large language models," 2023.

[29] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[30] K. Shuster *et al.*, "Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion," *arXiv preprint arXiv:2203.13224*, 2022.

[31] A. Xin *et al.*, "Llmael: Large language models are good context augmenters for entity linking," *arXiv preprint arXiv:2407.04020*, 2024.

[32] X. Liu *et al.*, "Onenet: A fine-tuning free framework for few-shot entity linking via large language model prompting," *arXiv preprint arXiv:2410.07549*, 2024.

[33] N. T. Le and A. Ritter, "Are language models robust coreference resolvers?" in *First Conference on Language Modeling*, 2024.

[34] Y. Oshima *et al.*, "Synthetic context with llm for entity linking from scientific tables," in *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, 2024, pp. 202–214.

[35] Y. Liu *et al.*, "Bridging context gaps: Leveraging coreference resolution for long contextual understanding," *arXiv preprint arXiv:2410.01671*, 2024.

[36] Q. Min *et al.*, "Synergetic event understanding: A collaborative approach to cross-document event coreference resolution with large language models," *arXiv preprint arXiv:2406.02148*, 2024.

[37] K. Sundar, S. Toshniwal, M. Tapaswi, and V. Gandhi, "Major entity identification: A generalizable alternative to coreference resolution," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 11 679–11 695.

[38] X. Li, K. Chen, Y. Long, and M. Zhang, "Llm with relation classifier for document-level relation extraction," *arXiv preprint arXiv:2408.13889*, 2024.

[39] Y. Tao, Y. Wang, and L. Bai, "Graphical reasoning: Llm-based semi-open relation extraction," *arXiv preprint arXiv:2405.00216*, 2024.

[40] Y. Hu, S. Ghosh, T.-P. Nguyen, and S. Razniewski, "Gptkb: Building very large knowledge bases from language models," *arXiv preprint arXiv:2411.04920*, 2024.

[41] A. Nayak and H. P. Timmapathini, "Llm2kb: Constructing knowledge bases using instruction tuned context aware large language models," *arXiv preprint arXiv:2308.13207*, 2023.

[42] S. Singhania, T.-P. Nguyen, and S. Razniewski, "Lm-kbc: Knowledge base construction from pre-trained language models," *Semantic Web challenge*, 2022.

[43] L. Zhu, J. Wang, and Y. He, "Llmlink: Dual llms for dynamic entity linking on long narratives with collaborative memorisation and prompt optimisation," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 11 334–11 347.

[44] T. Li *et al.*, "Long-context llms struggle with long in-context learning," *arXiv preprint arXiv:2404.02060*, 2024.

[45] F. Hughes and D. Schum, "Discovery-proof-choice, the art and science of the process of intelligence analysis-preparing for the future of intelligence analysis," *Washington, DC: Joint Military Intelligence College*, 2003.

[46] H. Wu *et al.*, "Where do i start? algorithmic strategies to guide intelligence analysts," in *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, 2012, pp. 1–8.

[47] M. S. Hossain, P. Butler, A. P. Boedihardjo, and N. Ramakrishnan, "Storytelling in entity networks to support intelligence analysts," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1375–1383.

[48] I. A. Tahmid *et al.*, "Enhancing immersive sensemaking with gaze-driven recommendation cues," in *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 2025, pp. 641–659.

[49] K. Davidson *et al.*, "Investigating professional analyst strategies in immersive space to think," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[50] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.