2026 29th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 979-8-3315-8470-2/26/$31.00 ©2026 IEEE | DOI: 10.1109/CSCWD68734.2026.11582182

Transient-RAG: Ephemeral Graph Topology for Dynamic Reasoning

Linquan Wu[1], Haoyu Yang[2], Shaochao Lin[3], Jacky Keung[1]
[1]City University of Hong Kong, Hong Kong
[2]University of Electronic Science and Technology of China, Chengdu, China
[3]Harbin Engineering University, Harbin, China

***Abstract*—Graph Retrieval-Augmented Generation (GraphRAG) has advanced multi-hop reasoning but remains bound by the "Static Indexing Paradigm", assuming knowledge is fixed and relationships are explicit. This rigidity fails in dynamic contexts, particularly Counterfactual Reasoning (e.g., "What if X didn't happen?"), where static retrieval fetches conflicting "ground truth." Furthermore, current methods suffer from a Resolution Mismatch, where coarse-grained chunk retrieval introduces semantic noise. We introduce Transient-RAG, a neuro-symbolic framework inspired by Bartlett's Reconstructive Memory. Unlike global graph methods (e.g., HippoRAG), Transient-RAG constructs a query-specific, ephemeral graph at inference time. It employs Semantic-Sentence Personalized PageRank (SSPPR) to filter noise and injects Hypothetical Edges to bridge logical gaps. Experiments on HotpotQA, 2WikiMultiHopQA, and ifQA show Transient-RAG achieves SOTA performance, significantly improving retrieval accuracy and counterfactual consistency while eliminating index maintenance.**

***Index Terms*—Neuro-Symbolic Reasoning, Graph RAG, Counterfactual Reasoning, Inference-Time Compute, Reconstructive Memory**

I. Introduction

In his seminal work *Remembering* (1932), Sir Frederic Bartlett argued that human memory is not a passive repository of fixed records but an active, imaginative process of reconstruction [1]. This cognitive flexibility enables humans to dynamically assemble knowledge in response to a specific query or premise, filling in gaps through inferences derived from prior experiences. Such an approach contrasts sharply with the architecture of current retrieval-augmented systems, where knowledge is stored statically and fixed prior to inference. Despite the remarkable success of Graph Retrieval-Augmented Generation (GraphRAG) systems, such as HippoRAG [2] and Microsoft GraphRAG [3], they remain inherently limited by the assumption that knowledge graphs are pre-computed and static, thus unable to adapt flexibly to dynamic or counterfactual reasoning tasks.

Traditional RAG systems, which rely on the retrieval of large pre-built knowledge graphs, struggle in settings where the query deviates from factual reality. These systems assume a fixed representation of knowledge, which does not allow for the incorporation of hypothetical scenarios or counterfactual reasoning. For instance, as shown in figure 1, consider a query like: *"If Los Angeles were located on the east coast of the U.S., what would the time difference between Los Angeles and Paris be?"* In a traditional static graph-based system, the answer to such a counterfactual query would be based solely on pre-existing relationships and facts, which may not align with the hypothetical premises of the question. Such systems would retrieve documents about Los Angeles and Paris' time zones but fail to dynamically account for the shift in the position of Los Angeles, leading to a misleading answer based on incorrect assumptions.

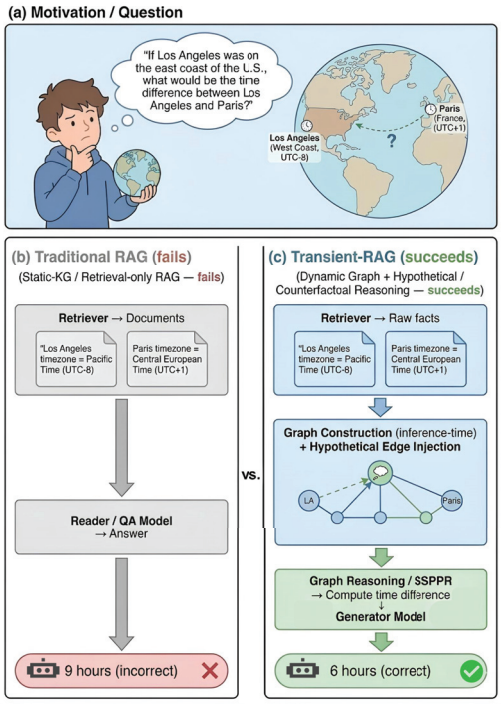

Fig. 1. Case study illustrating how Transient-RAG answers the question about the time difference between Los Angeles and Paris if Los Angeles were on the east coast of the U.S., based on a question from the *ifQA* dataset. (a) Traditional RAG fails with static knowledge, while (b) Transient-RAG succeeds with dynamic graph reasoning.

In contrast, Transient-RAG shifts graph construction from *indexing time* to *inference time*, dynamically generating a query-specific, ephemeral graph that is dissolved post-generation. This flexibility allows for the injection of *hypothetical edges*—probabilistic links representing scenarios not explicitly stated in the knowledge base—thereby enabling robust counterfactual reasoning as illustrated in our case study.

By constructing these transient topologies, our framework addresses critical limitations of static RAG: it eliminates "Contextual Amnesia" by focusing solely on query-relevant knowledge, resolves "Resolution Mismatch" through sentence-level filtering, and supports abductive reasoning to explore alternative scenarios beyond established facts.

This work proposes a framework that challenges the traditional "Static World Assumption" of knowledge graphs by treating the graph as a transient and dynamic artifact, constructed specifically for each query. The Transient-RAG framework aligns with Bartlett's theory of reconstructive memory by allowing for a more flexible and adaptive process of knowledge retrieval, reasoning, and generation. In the following sections, we will describe the key components of Transient-RAG, including its fine-grained semantic sentence initialization, the use of Semantic-Sentence Personalized PageRank (SSPPR) for noise filtering, and the injection of hypothetical edges to bridge gaps in knowledge.

II. Related Work

A. Static Graph Retrieval-Augmented Generation

Recent advancements in GraphRAG have demonstrated significant improvements in multi-hop reasoning by grounding language models in structured knowledge bases [4], [5]. Methods such as HippoRAG [2] utilize Personalized PageRank (PPR) on static knowledge graphs to simulate hippocampal indexing, while G-Retriever [6] integrates graph neural networks to refine retrieval. Microsoft's GraphRAG [3] further utilizes community detection to generate global summaries. However, these approaches rely on a "Static World Assumption": treating the knowledge graph as a fixed, immutable index constructed prior to inference. This rigidity limits their applicability in dynamic or counterfactual scenarios where the optimal retrieval path depends on a premise that contradicts the pre-indexed ground truth.

B. Retrieval Granularity and Cognitive Precision

Standard RAG and recent efficient variants like LinearRAG [7] typically operate at the *chunk level* (200-500 tokens). While computationally efficient, this coarse granularity introduces a "Resolution Mismatch", where a retrieved chunk often contains irrelevant sentences that act as "Semantic Trojan Horses," activating unrelated subgraphs during reasoning. Transient-RAG addresses this by implementing *sentence-level* semantic gating via SSPPR, ensuring that only high-precision evidence enters the graph topology, akin to cognitive mechanisms that filter sensory noise before memory consolidation.

C. Inference-Time Compute and Structured Reasoning

The paradigm of "Inference-Time Compute" posits that allocating more computational resources during the reasoning phase can yield better outcomes than larger pre-trained models alone. This is exemplified by Chain-of-Thought (CoT) prompting [8] and recent reasoning models like OpenAI o1. Recent works like GIVE [9] attempts to bridge this by extrapolating veracity from KGs, yet they still rely on existing graph structures. Transient-RAG extends this philosophy by implementing "Structured Chain-of-Thought." Instead of generating a linear sequence of text, our framework utilizes inference-time compute to construct a structured topology (a graph) that explicitly models dependencies and hypothetical links.

D. Dynamic Memory Systems

Our work also intersects with research on dynamic memory for agents. Generative Agents [10] utilize "Memory Streams" to simulate human-like behavior, while A-MEM [11] introduces structured agentic memory for evolving agents. However, these systems primarily focus on persistence, such as accumulating long-term logs. In contrast, Transient-RAG focuses on ***transience***. We construct a structured and ephemeral graph topology on the fly, designed solely for the current reasoning context and discarded immediately after, akin to the "working memory" models in cognitive science [12], [13].

III. The Framework of Transient-RAG

We propose **Transient-RAG**, a framework that shifts graph construction from offline indexing to inference time. As illustrated in Figure 2, the pipeline consists of four phases: (i) **Sensory Gating**, which retrieves raw chunks to form a candidate pool; (ii) **The Dreaming**, which constructs a query-specific graph with hypothetical edges; (iii) **Epistemic Navigation**, which executes structure-aware reasoning via Personalized PageRank; and (iV) **Dissolution**, which takes advantage of LLM to generate final answer and dissolve the graph.

A. Problem Formulation

Given a user query q and a corpus $\mathcal{D}$, we aim to generate an answer a. Unlike traditional GraphRAG which queries a static global graph G_{global}, we formulate the reasoning process as constructing a *transient* topology G_q at inference time. Formally, this process is defined as:

$$G_q = (V, E) = \phi(q, \mathcal{C}_{topk}), \quad a = \text{Generator}(q, \text{Reason}(G_q)) \tag{1}$$

where $\mathcal{C}_{topk}$ is the set of retrieved chunks and ϕ is the graph construction function. The node set V comprises sentence nodes V_s and entity nodes V_e, while the edge set E contains both explicit structural links and hypothetical inference links. This design eliminates the need for maintaining a global index, ensuring that the graph structure is always tailored to the current query context.

B. Phase I: Sensory Gating

The first phase aims to filter massive noise from the corpus $\mathcal{D}$. We employ a dense retriever (Bi-Encoder) to fetch the top-K relevant chunks $\mathcal{C} = \{c_1, \ldots, c_K\}$ based on semantic similarity to the query embedding $\mathbf{e}_q$.

$$\mathcal{C} = \underset{c \in \mathcal{D}}{\text{top-}K}(\cos(\mathbf{e}_q, \mathbf{e}_c)) \tag{2}$$

This stage serves as a coarse-grained filter, providing the raw material for the subsequent fine-grained graph construction.

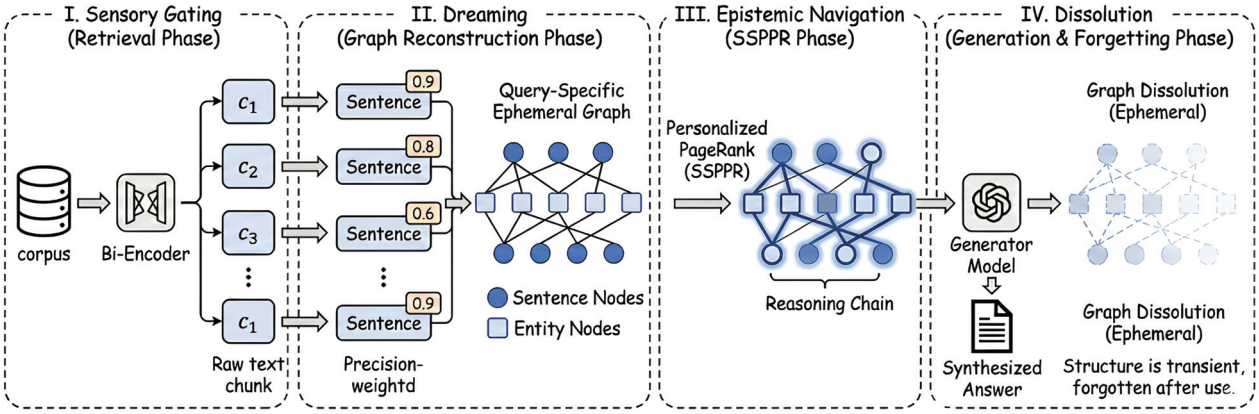

Fig. 2. **The Transient-RAG Framework.** (A) **Sensory Gating**: Retrieving top-K chunks. (B) **The Dreaming**: Deconstructing chunks into sentences, calculating semantic precision, and injecting hypothetical edges. (C) **Epistemic Navigation**: Running SSPPR to identify the reasoning chain. (D) **Dissolution**: Generating the answer and dissolving the graph.

C. Phase II: Graph Construction via "The Dreaming"

In this phase, we convert unstructured chunks into a structured reasoning topology. To resolve the *Resolution Mismatch* problem, where irrelevant sentences in a chunk introduce noise, we propose a sentence-level construction method augmented by Large Language Models (LLMs).

1) Semantic-Sentence Initialization: We first decompose the retrieved chunks $\mathcal{C}$ into a sentence set $\mathcal{S} = \{s_1, \ldots, s_m\}$. For each sentence s_i, we compute a semantic precision score w_i using a lightweight Cross-Encoder. This score $w_i = \text{Sim}(q, s_i)$ represents the "semantic energy" of the sentence. Sentences with scores below a threshold τ are treated as noise and effectively pruned from the graph initialization, ensuring that the subsequent topology is built solely on high-relevance foundations.

2) Topology Generation with Hypothetical Edges: On top of the filtered sentences, we employ an LLM to "dream" the graph topology. The LLM takes the valid sentence set and query q as input to extract nodes and edges. We define two distinct types of edges to capture both factual and inferential connections. Explicit Edges (E_{exp}) represent relations strictly supported by the text, such as entity co-occurrence or syntactic dependencies, providing a factual grounding. Hypothetical Edges (E_{hyp}), in contrast, are probabilistic links inferred via abductive reasoning to bridge logical gaps required by the query, such as connecting a counterfactual premise to its potential consequence. The final transient graph is defined as $G_q = (V, E_{exp} \cup E_{hyp})$. This hybrid structure allows the system to reason over paths that are not explicitly stated in the text but are logically implied by the query premise.

D. Phase III: Epistemic Navigation via SSPPR

To identify the optimal reasoning chain within G_q, we propose Semantic-Sentence Personalized PageRank (SSPPR).

1) Vectorized Initialization: Standard PPR initializes the restart vector uniformly or based on exact entity matches. In contrast, SSPPR constructs a semantic-aware restart vector $\mathbf{v} \in \mathbb{R}^{|V|}$. For each node $v_j \in V$, its initial weight is determined by Max-Pooling over the semantic scores of the sentences containing it. Specifically, if a node is part of the query entities E_q, it receives a weight of 1.0; otherwise, it inherits the maximum semantic score of its parent sentences, provided this score exceeds τ. This mechanism acts as a semantic firewall, preventing low-relevance sentences from injecting energy into the graph traversal.

2) Iterative Propagation and Dissolution: We compute the final importance scores $\mathbf{r}$ using the iterative PPR formulation $\mathbf{r}^{(t+1)} = (1-\alpha)\mathbf{v} + \alpha\mathbf{A}^\top\mathbf{r}^{(t)}$, where $\mathbf{A}$ is the column-normalized adjacency matrix of G_q and α is the damping factor. The nodes with top-M scores in $\mathbf{r}$ form the reasoning subgraph passed to the generator. Finally, immediately after generating answer a, the graph G_q is dissolved. This explicitly

Algorithm 1: Transient-RAG Inference Process

Input: Query q, Corpus $\mathcal{D}$, Threshold τ
Output: Answer a

$\mathcal{C} \leftarrow \text{RetrieveTopK}(q, \mathcal{D})$;
$\mathcal{S} \leftarrow \text{SegmentIntoSentences}(\mathcal{C})$;
$V \leftarrow \emptyset, E \leftarrow \emptyset$;
foreach $s_i \in \mathcal{S}$ **do**
  $w_i \leftarrow \text{CrossEncoder}(q, s_i)$;
  if $w_i > \tau$ **then**
    $V \leftarrow V \cup \{s_i\}$;
  end
end
$E_{exp} \leftarrow \text{ExtractExplicitEdges}(V)$;
$E_{hyp} \leftarrow \text{InjectHypotheticalEdges}(V, q)$;
$G_q \leftarrow (V, E_{exp} \cup E_{hyp})$;
$\mathbf{r} \leftarrow \text{SSPPR}(G_q, q)$;
$a \leftarrow \text{Generate}(q, \mathbf{r})$;
return a

trades storage complexity for inference-time compute, ensuring no stale or conflicting knowledge persists.

E. Complexity and Efficiency Analysis

The proposed framework introduces a fundamental trade-off between storage and compute. Traditional GraphRAG systems require $O(|\mathcal{D}|)$ storage for a global graph, which can be prohibitive for massive corpora. Transient-RAG reduces storage cost to zero by constructing graphs on-the-fly. The computational complexity of graph construction is $O(|\mathcal{C}_{topk}|)$, which is linear with respect to the number of retrieved chunks and significantly smaller than the global corpus size. While this incurs an inference-time latency, the use of sparse matrix operations for SSPPR ensures that the reasoning phase remains efficient, making the approach scalable for real-time applications where adaptability is paramount.

IV. Experiments

A. Experimental Setup

Datasets. We evaluate Transient-RAG on two multi-hop reasoning benchmarks and one counterfactual dataset. For multi-hop reasoning, we use HotpotQA [14] and 2WikiMultiHopQA [15]. Following the protocol established in HippoRAG [2] and LinearRAG [7], we utilize a randomly sampled subset of 1,000 examples from the training set of each dataset. These datasets require aggregating information across multiple documents to answer complex queries. For counterfactual reasoning, we employ ifQA [16], a dataset designed to evaluate models' ability to reason about hypothetical scenarios (e.g., altering the outcome of a historical event). From its 3,800 questions with counterfactual presuppositions, we select a random subset of 1,000 examples for evaluation.

Evaluation Metrics. For all datasets, we use Exact Match (EM) and F1 score as the primary evaluation metrics. EM measures the percentage of predictions that exactly match the ground truth answer(s), while F1 score measures the token overlap between the predicted and ground truth answers, considering partial matches.

Baselines. We categorize all the baselines into three groups: (i) Zero-shot LLM Inference: We evaluate foundational model using Qwen3-8B [17]. (ii) Vanilla RAG: We deploy standard RAG methodologies retrieving the top-20 passages, combining semantic-based document retrieval with chain-of-thought reasoning prompts to guide the LLM's generation process. (iii) State-of-the-art GraphRAG Systems: We compare against leading GraphRAG implementations including RAPTOR [18], LightRAG [19], HippoRAG [2], and HippoRAG2 [20].

Implementations. For consistency in implementation, all algorithms use the same embedding model, i.e., bge-large-en-v1.5 [21], for generating dense embeddings. We set $k = 20$ for top-k retrieval in all methods. All RAG approaches employ the same Large Language Model (LLM), Qwen-3-8B [17], for both generation and evaluation tasks. All experiments are conducted on four Nvidia H20 GPU.

B. Main Results

Table I presents the comprehensive evaluation across all three datasets.

1) Multi-hop Reasoning Performance: On HotpotQA and 2WikiMultiHopQA, Transient-RAG demonstrates competitive or superior performance compared to the strong static graph baseline, HippoRAG2. Notably, on HotpotQA, our method achieves an EM of 63.8 and F1 of 73.3, surpassing HippoRAG2 by 1.4% and 3.2% respectively. This indicates that our SSPPR initialization effectively filters out "semantic trojan horses" (irrelevant sentences within relevant chunks), ensuring that the retrieval precision is higher than coarse-grained graph methods.

2) Counterfactual Reasoning Performance: In dynamic scenarios (ifQA), static methods struggle significantly as they retrieve conflicting "ground truth" facts. As shown in the rightmost columns of Table I, Transient-RAG achieves a 30.2% improvement in EM and 28.3% improvement in F1 score over HippoRAG2. By treating the graph as a transient artifact, our model avoids the "Contextual Amnesia" trap and "dreams" a topology consistent with the counterfactual premise.

C. Ablation Study

To validate the individual contributions of our framework's components, we conducted an ablation study on both HotpotQA and ifQA datasets, with results summarized in Table II. On HotpotQA, the removal of the Semantic-Sentence Personalized PageRank (SSPPR) module leads to the most significant performance drop (EM -9.6), confirming that sentence-level gating is crucial for filtering semantic noise in multi-hop reasoning. Conversely, on ifQA, excluding the Hypothetical Edge Injection mechanism results in the steepest decline (EM -12.6), lowering EM to 56.8. This distinction highlights that while noise filtering is essential for standard retrieval, the "dreaming" phase—which injects probabilistic links—is the decisive factor for enabling counterfactual reasoning in dynamic contexts.

V. Conclusion

We presented **Transient-RAG**, a framework that challenges the "Static Indexing Paradigm" by shifting the locus of graph construction from storage to inference. By combining **Semantic-Sentence Personalized PageRank (SSPPR)** with **Hypothetical Edge Injection**, our approach effectively resolves the "Resolution Mismatch" problem and enables robust counterfactual reasoning. Our results demonstrate that knowledge need not be a rigid, pre-computed artifact but can be a dynamic, reconstructive process, a "dream" constructed dynamically to fit the query's reality. Future work will explore "Memory Consolidation," investigating how high-value hypothetical edges can be selectively persisted to evolve the long-term knowledge base.

TABLE I

Main Results across all tasks. We compare performance on Multi-hop Reasoning (HotpotQA, 2WikiMultiHopQA) and Counterfactual Reasoning (ifQA). *EM* and *F1 score* are used for all datasets. For ifQA, we also report Recall@K (R@K).

Method	Multi-Hop QA				Counterfactual QA	
	HotpotQA		2WikiMultiHopQA		ifQA	
	EM	F1	EM	F1	EM	F1
Zero Shot LLM Inference						
Qwen3-8B[17]	22.8	32.1	14.5	22.7	21.7	27.9
NaiveRAG with top k-20						
Naive RAG	48.7	62.3	28.7	41.9	34.4	39.5
Knowledge Graph Driven Retrieval-Augmented-Generation Method						
LightRAG [19]	1.0	2.4	7.6	11.2	12.5	12.6
RAPTOR [18]	57.5	69.5	43.7	52.1	51.5	57.6
HippoRAG [2]	52.2	63.5	62.2	71.8	48.4	56.7
HippoRAG2 [20]	62.9	71.0	**68.2**	**75.5**	53.3	58.6
Transient-RAG (Ours)	**63.8**↑0.9	**73.3**↑2.3	67.5↓0.7	74.1↓1.4	**69.4**↑16.1	**75.2**↑16.6

TABLE II

Ablation results on HotpotQA and ifQA.

Configuration	HotpotQA		ifQA	
	EM	F1	EM	F1
Transient-RAG (Full)	**63.8**	**73.3**	**69.4**	**75.2**
w/o SSPPR (Chunk-level)	54.2	64.5	62.1	68.5
w/o Hypothetical Edges	59.7	69.8	42.8	47.4

References

[1] Frederic Charles Bartlett, *Remembering: A study in experimental and social psychology*, Cambridge university press, 1995.

[2] Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su, "Hipporag: Neurobiologically inspired long-term memory for large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 59532–59569, 2024.

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

[4] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang, "Graph retrieval-augmented generation: A survey," *ACM Transactions on Information Systems*, 2024.

[5] Linquan Wu, Tianxiang Jiang, Haoyu Yang, Wenhao Duan, Shaochao Lin, Zixuan Wang, Yini Fang, and Jacky Keung, "Facesleuth-r: Adaptive orientation-aware attention for robust micro-expression recognition," 2025.

[6] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," *Advances in Neural Information Processing Systems*, vol. 37, pp. 132876–132907, 2024.

[7] Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang, "Linearrag: Linear graph retrieval augmented generation on large-scale corpora," *arXiv preprint arXiv:2510.10114*, 2025.

[8] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[9] Jiashu He, Mingyu Derek Ma, Jinxuan Fan, Dan Roth, Wei Wang, and Alejandro Ribeiro, "Give: Structured reasoning with knowledge graph inspired veracity extrapolation," 2024.

[10] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.

[11] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang, "A-mem: Agentic memory for llm agents," *arXiv preprint arXiv:2502.12110*, 2025.

[12] Alan Baddeley, "Working memory," *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie*, vol. 321, no. 2-3, pp. 167–173, 1998.

[13] Linquan Wu, Tianxiang Jiang, Yifei Dong, Haoyu Yang, Fengji Zhang, Shichaang Meng, Ai Xuan, Linqi Song, and Jacky Keung, "Lavit: Aligning latent visual thoughts for multi-modal reasoning," 2026.

[14] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 2369–2380.

[15] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," *arXiv preprint arXiv:2011.01060*, 2020.

[16] Wenhao Yu, Meng Jiang, Peter Clark, and Ashish Sabharwal, "Ifqa: A dataset for open-domain question answering under counterfactual presuppositions," *arXiv preprint arXiv:2305.14010*, 2023.

[17] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu, "Qwen3 technical report," 2025.

[18] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning, "Raptor: Recursive abstractive processing for tree-organized retrieval," in *The Twelfth International Conference on Learning Representations*, 2024.

[19] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang, "Lightrag: Simple and fast retrieval-augmented generation," *arXiv preprint arXiv:2410.05779*, 2024.

[20] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su, "From rag to memory: Non-parametric continual learning for large language models," *arXiv preprint arXiv:2502.14802*, 2025.

[21] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff, "C-pack: Packaged resources to advance general chinese embedding," 2023.