# Beyond Fact Retrieval: Episodic Memory for RAG with Generative Semantic Workspaces

**Shreyas Rajesh, Pavan Holur, Chenda Duan, David Chong, Vwani Roychowdhury**

University of California, Los Angeles
shreyasrajesh38@ucla.edu, pholur@ucla.edu, chenda@ucla.edu,
davidchong13807@ucla.edu, vwani@ucla.edu

## Abstract

Large Language Models (LLMs) face fundamental challenges in long-context reasoning: many documents exceed their finite context windows, while performance on texts that do fit degrades with sequence length, necessitating their augmentation with external memory frameworks. Current solutions, which have evolved from retrieval using semantic embeddings to more sophisticated structured knowledge graphs representations for improved sense-making and associativity, are tailored for fact-based retrieval and fail to build the space-time-anchored narrative representations required for tracking entities through episodic events. To bridge this gap, we propose the **Generative Semantic Workspace** (GSW), a neuro-inspired generative memory framework that builds structured, interpretable representations of evolving situations, enabling LLMs to reason over evolving roles, actions, and spatiotemporal contexts. Our framework comprises an *Operator*, which maps incoming observations to intermediate semantic structures, and a *Reconciler*, which integrates these into a persistent workspace that enforces temporal, spatial, and logical coherence. On the Episodic Memory Benchmark (EpBench) (Huet, Houidi, and Rossi 2025) comprising corpora ranging from 100k to 1M tokens in length, GSW outperforms existing RAG based baselines by up to **20%**. Furthermore, GSW is highly efficient, reducing query-time context tokens by **51%** compared to the next most token-efficient baseline, reducing inference time costs considerably. More broadly, GSW offers a concrete blueprint for endowing LLMs with human-like episodic memory, paving the way for more capable agents that can reason over long horizons.

## 1 Introduction

Large Language Models (LLMs) have transformed natural language understanding, but their ability to reason over long contexts is still limited by finite input windows. Even with token limits in the millions, large document collections can easily exceed these bounds. Performance can also degrade with context length due to phenomena like "context rot" and "lost-in-the-middle" effects (Liu et al. 2023; Hong, Troynikov, and Huber 2025). A common workaround is Retrieval-Augmented Generation (RAG), which supplements the LLM's input with only the most relevant retrieved content at query time. Standard RAG pipelines split documents into smaller chunks, encode them into dense embeddings, and retrieve the top-matching chunks based on semantic similarity to the query—allowing the LLM to focus on a relevant subset of the corpus during inference.

A key limitation of standard RAG methods is that each text chunk is embedded independently, which can lead to incomplete retrieval when a query depends on information spread across multiple chunks. Because similarity scores are computed in isolation, essential context may be missed. To address this, more recent approaches have adopted structured representations — such as knowledge graphs — that explicitly model relationships between chunks across the corpus. At query time, these graphs are traversed or queried to retrieve semantically connected chunks, enabling LLMs to perform more effective multi-hop reasoning and question answering (Gutiérrez et al. 2025a,b; Edge et al. 2025; Guo et al. 2025).

These methods have primarily been evaluated on fact-rich documents such as Wikipedia pages (Yang et al. 2018; Ho et al. 2020; Trivedi et al. 2022). Yet **the vast majority of texts that LLMs encounter are not lists of facts but narratives of evolving real-world situations**. Crime reports, political briefings, corporate filings, legislative records, war dispatches, and multi-day news coverage all describe **actors** (people, organizations, nations) that adopt **roles** (suspect, regulator, bidder, combatant) and transition through **states** (arrested → arraigned → released; startup → unicorn → acquired) while interacting across **space and time** (Shahsavari et al. 2020a,b; Tangherlini et al. 2020).

We contend that reasoning over such documents would be much more accurate and energy efficient, if one indexed the documents in terms of **an internal world model**— a structured representation that keeps track of *who* is involved, *what* was done, *where* and *when* events occur, *how* roles change, and *what* consequences follow. Indeed, to achieve such a goal, humans possess *episodic memory* (Tulving 1972, 2002) enabling us not only to plan and reason to seamlessly operate in the real world, but also to create new or update existing world models by reasoning across multiple experiences (Schacter, Addis, and Buckner 2007; Hassabis and Maguire 2007).

In this work, we introduce the **Generative Semantic Workspace** (GSW), a unifying computational framework for modeling world knowledge as structured, probabilistic semantics in the era of Large Language Models (LLMs).

GSW formalizes how an intelligent agent—human or artificial—constructs and updates an internal representation of evolving situations from sequential input (e.g., text, video, or dialogue modalities). These representations are interpretable, actor-centric, and predictive: they reflect semantic regularities in the past while projecting likely future outcomes. GSW may be viewed as an instance of *episodic memory* that can be integrated into LLM-based systems as a reasoning and memory module, serving as a symbolic bridge between language and latent world models.

To illustrate how GSW can help LLMs reason accurately, we evaluate it on the Episodic Memory Benchmark (EpBench) (Huet, Houidi, and Rossi 2025), that has recently been introduced as a way to benchmark the episodic memory-like capabilities of LLMs. Following are excerpts from two different documents that relate to an entity, Carter Stewart, in this EpBench dataset:

> **Document #1:** The imposing structure loomed before him, its grand facade a testament to both artistry and scientific achievement ..... As he stepped into the **Metropolitan Museum of Art**, the echoing chatter of excited voices ..... The antique clock in the main hall chimed, its resonant tones reminding him of the date: **September 22, 2026** .... found himself particularly engrossed during the third presentation, where **Carter Stewart** explained statistical analysis with a clarity that left the audience spellbound.

> **Document #2:** The air crackled with tension as **Carter Stewart** stepped onto the pristine greens of **Bethpage Black Course** on **March 23, 2024** ..... Carter discussed implications of research, his fingers trembling slightly as he adjusted his microphone.

An agent reading the narrative in the first document faces a fundamentally different challenge than traditional fact retrieval. It must understand that "he" refers to a nameless protagonist, who attended a scientific conference where Carter Stewart spoke. The narrator's spatial context (Metropolitan Museum of Art) and temporal context (September 22, 2026), are stated only indirectly and more importantly have to be also assigned to Carter Stewart who is a presenter. GSW is able to create such representations as part of its working memory construction task: "Carter Stewart: **Role:** A presenter at a Scientific Conference; **Date:** September 22, 2026, morning session; **Location:** The Metropolitan Museum of Art, **Topic:** statistical analysis; **Implements Used:** presentation boards and holographic projectors." The second document is more straightforward and GSW creates a memory trace such as: "Carter Stewart; **Role:** a researcher and presenter; **Location:** Bethpage Black Course; **Date:** March 23, 2024, **Did What?:** Presented his research findings at a Scientific Conference."

When presented with a task such as "List all the unique locations and dates where Carter Stewart made presentations at Scientific Conference events." a query resolution module (see Section 3) searches through the GSW constructed from all 200 documents and identifies entities mentioned in the query (e.g., Carter Stewart) that match query's intent (e.g., a presenter at scientific conference; another entity named Carter Stewart whose role is that of a baker by profession would be ignored) and then returns just the relevant portion of its memory, as in the preceding paragraph. This results in highly targeted and short texts that an LLM has to reason through to provide an answer. In contrast, current structured RAG methods are designed to facilitate retrieval of either whole chunks or community-level summaries that have different levels of similarity to the entities and other phrases in the query. For example, for this query GraphRAG's (Edge et al. 2025) summarization missed that Carter Stewart was at the same location as the protagonist in Document #1, and included irrelevant text chunks which led to a list that misses one location and hallucinates two erroneous locations. HippoRAG2 (Gutiérrez et al. 2025b) retrieves the full text of both the relevant documents, along with many other documents, overwhelming the LLM and leading it to hallucinate three erroneous locations. For a more detailed comparison, see Section 6, and Tables 1, 4, and 3.

In the rest of this paper, we detail the GSW framework (**Section 2**) and present a rigorous evaluation on two versions of the EpBench benchmark (**Section 3**). The results demonstrate a significant improvement over existing methods. On the EpBench-200 corpus, GSW achieves a state-of-the-art F1-score of 0.85, outperforming strong structured RAG baselines. This advantage is particularly pronounced in the most demanding queries requiring synthesis across as many as 17 different documents, where GSW improves recall by up to **20%** over the next best approach as detailed in Table 2. Furthermore, GSW is efficient, reducing the number of context tokens sent to the LLM by **51%** compared to the most token-efficient baseline, drastically lowers inference costs and reducing the rate of hallucination in question answering (see Table 3). We further show that this powerful combination of accuracy and token efficiency holds at scale; on the EpBench-2000 corpus, a 10x larger dataset, GSW again achieves a state-of-the-art F1-score of 0.773, outperforming the best baseline by more than **15%** on overall recall (Table 4), positioning GSW as a robust and scalable solution for equipping LLMs with effective episodic memory.

Results and discussions are summarized in **Section 4** and a review of related literature is presented in **Section 5**. Finally, limitations and future work are discussed in **Section 6**. The full version of the paper [1] provides supporting evidence, including manual evaluations performed to validate the power of GSW's episodic memory capabilities.

## 2 The Generative Semantic Workspace (GSW) Framework

In neuroscience, the neocortex is believed to encode hierarchical abstractions of entities, roles, and event templates (George and Hawkins 2009; Botvinick 2008; Felleman and Van Essen 1991). The hippocampus, especially the CA3 module, plays a complementary role by binding these representations into coherent spatiotemporal sequences (Teyler and DiScenna 1986; Rolls 2013; Eichenbaum 2004). During sleep, this

[1]The full version of this paper with a detailed technical appendix and clear visualization of the GSW using an example is available at https://arxiv.org/abs/2511.07587.

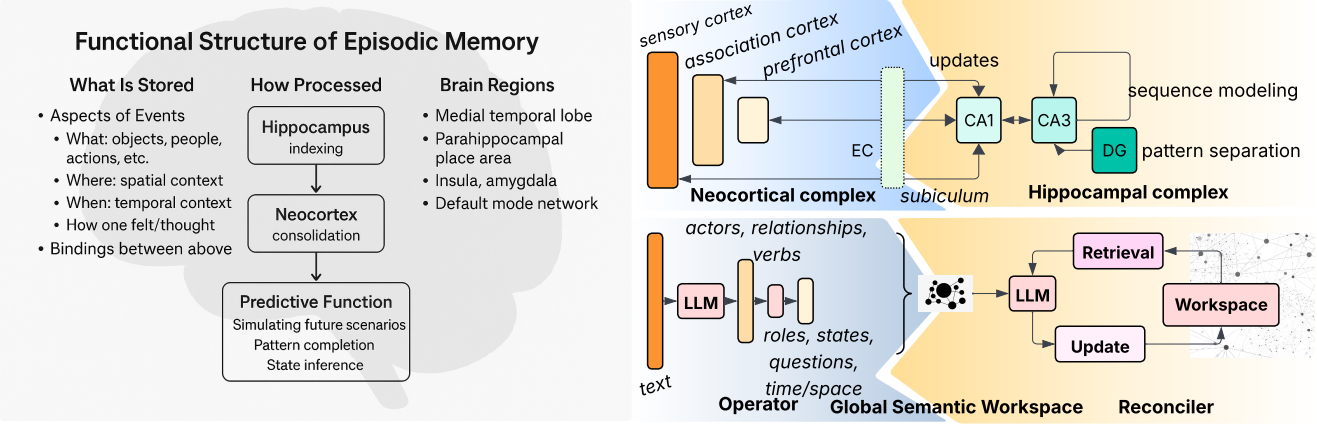


Figure 1: **Unifying Brain-Inspired and Generative Semantics for Episodic Memory Modeling** The hippocampal complex (DG, CA3, CA1) and neocortical regions (NC) inspire the *Reconciler* (retrieval, workspace, update) and *Operator* (LLM-driven semantic role extraction), respectively. The neocortical complex, responsible for context-rich consolidation and predictive modeling, aligns with the Operator module's functions. The hippocampal complex, which performs indexing, pattern separation, and sequence modeling, corresponds to the Reconciler. Together, the GSW framework offers a biologically inspired, interpretable model for simulating world knowledge from text inputs.

neocortical-hippocampal system engages in *experience replay*, a process through which episodic traces are reactivated in reverse or forward order to consolidate memory and refine internal models (Ólafsdóttir, Bush, and Barry 2018; Louie and Wilson 2001; Wilson and McNaughton 1994). This back and forth supports both persistence and prediction of memory (McClelland, McNaughton, and O'Reilly 1995; Rasch and Born 2013), key features of episodic memory.

Motivated by this biological architecture (see Fig 1), an effective memory framework requires a **structured representation** capable of encoding actors along with their evolving roles and states. Crucially, this representation must be capable of spatiotemporal grounding, linking entities and their interactions to specific times and locations, much like the binding function of the hippocampus. Finally, the framework must possess a process for **consolidating and updating these structures** as new information arrives, mirroring the way the neocortical-hippocampal loop constantly refines its world model.

**From Episodic Memory to Generative Modeling of Situations and Narratives:** The central challenge, therefore, is to create a continuously evolving semantic model, which requires a bidirectional mapping between text and a structured representation. While early symbolic frameworks like PropBank (Kingsbury and Palmer 2002) and FrameNet (Baker, Fillmore, and Lowe 1998) attempted this, they were not designed for this full bidirectional process, relying instead on fixed ontologies that lacked the necessary probabilistic and dynamic interpretation.

*LLMs now make this bidirectional mapping tractable.* They can both infer concise semantic identifiers from text and generate coherent narratives from those identifiers. This enables a new, efficient memory model where compact semantic traces are stored and reactivated in context. The formal model is presented next.

## 2.1 A Probabilistic Model for Semantic Memory: The Operator Framework

We now define a minimal schema for encoding these semantic elements—along with predictive cues, spatiotemporal attributes, and utilities—that serves as the foundation of the GSW framework for structured memory in LLMs. The agent must distill and maintain a semantic map from text to build a coherent semantic model.

To make this concrete, let's consider a single text input $C_n$ at some time step $n$ : *Yesterday, in a swift response to a reported robbery, law enforcement officers apprehended Jonathan Miller, a 32-year-old resident of Greenview Avenue, in the downtown area.*

Explicit information in $C_n$ typically specifies a configuration of participating actors $a_1, \ldots, a_K$ and the relations or interactions among them. The agent must distill and maintain a semantic map from these clues to build a coherent semantic model. Let's represent this interaction pattern at time step $n$ as (here each entry denotes an interaction from actor $a_i$ to $a_j$ as inferred from $C_n$):

$$\mathcal{C}_n \approx \begin{pmatrix} (a_1 \to a_1)^n & \cdots & (a_1 \to a_K)^n \\ \vdots & \ddots & \vdots \\ (a_K \to a_1)^n & \cdots & (a_K \to a_K)^n \end{pmatrix};$$

**Actors, Roles and States**

The word 'Miller', in isolation, corresponds to a broad, unconditioned distribution over possible behaviors of a human. If 'Miller' is likely to commit a crime, the agent would probably refer to Miller with a label 'Criminal'. We call these labels *roles*.

**Role:** An identifier that specifies a distribution over potential actions that an actor $a_i \in \mathcal{A}$ may take toward other actors $a_j \in \mathcal{A}$:

$$\pi_r : \mathcal{A} \times \mathcal{A} \to [0, 1] \tag{1}$$

where $\pi_r(a_i \to a_j)$ denotes the probability of $a_i$ acting on $a_j$ in role $r$. For example, assigning the role of 'criminal' to Miller increases the *likelihood* that he will engage in actions such as *committing a crime* against another actor or increasing the chances that Miller will *attempt to flee* from 'law enforcement'.

The agent would also *know* that in addition to Miller being a *criminal*, Miller has been *caught*. Or perhaps he *escaped*. We call these labels *states*.

**State:** An identifier that induces a contextual attribute that modulates the probability distribution over actions available to an actor within a given role. Given an actor $a_i$ with role $r$, a state $s \in \mathcal{S}_r$ constrains the role-induced action distribution $\pi_r$:

$$\pi_{r,s}(a_i \to a_j) = \pi_r(a_i \to a_j \mid s), \tag{2}$$

where $\pi_{r,s}$ denotes the subset of actions available to actor $a_i$ in state $s$. For instance, a *criminal* in the state *captured* may be limited to passive or compliant interactions, precluding actions such as fleeing or committing further crimes. Thus, states act as dynamic modifiers of an actor's interaction profile within a given situation.

**Verbs and Valences**

Verbs encode structured semantic attributes helping the agent to structure an event by drawing on prior experience, as verbs tend to generalize across contexts more reliably than nouns. They provide causal certificates for roles/states of actors. For example, understanding why Miller transitions from being *free* to *captured* relies on identifying the underlying interaction – such as being arrested – that bridges those states. A verb's valences are efficient means of capturing information needed for reasoning about future outcomes. Verbs can be modeled similar to roles and states:

$$v(a_i \to a_j) : \mathcal{A} \times \mathcal{A} \to \mathcal{L}_v, \tag{3}$$

where the valences $\ell_k \in \mathcal{L}_v$ signal the change in roles and states of the actors interacting via the verb. When Miller is running from the police, the *next* state for Miller might be *escaped* or *caught*: a distribution of potential *future* roles and states.

**Time and Space Continuity**

Spatiotemporal continuity constraints are crucial to capture world models, not only for individual actors but especially as interactions/verbs couple their coordinates. For instance, if Officers are actively apprehending Johnathan Miller in the Downtown area, then it enforces a shared location and time among the actors. Moreover, if the next day Miller is found in a city a thousand miles away, it would constrain his unobserved action to that of having flown and lead the agent to narrow down events that could have led to such a spatial shift. In effect, the flow of time and space regularizes the semantic map, biasing verb selection toward contextually coherent transitions. If the position information derived from $\mathcal{C}_n$ at time step $n$ is $\mathcal{X}_n$ and the temporal information is $\mathcal{T}_n$, then:

**Temporal continuity:** $\mathcal{T}_{n+1} - \mathcal{T}_n$ must be consistent with the expected temporal scope of $v$,

**Spatial proximity:** $\|\mathcal{X}_n(a_i) - \mathcal{X}_n(a_j)\|$ must fall within a valid range for the verb (e.g., *tackle* requires physical closeness)

**Forward-Falling Questions to Capture Potential Outcomes and Actions**

The collection of roles/states, verbs, and spatiotemporal coordinates constrain the space of future progression and can be efficiently encoded as a set of questions $\mathcal{Q}_n$. For example, given that Miller has been arrested, "When would Miller be indicted," "where and when would the trial happen?" "Will he be free on bail?" A prosecutor agent, for example, would need to start strategizing about such potential outcomes.

A complete workspace instance can be written as a sampled distribution from an underlying "Workspace" generative process:

$$\mathcal{M}_n \sim p(\mathcal{A}, \mathcal{R}, \mathcal{S}, \mathcal{V}, \mathcal{T}, \mathcal{X}, \mathcal{Q} \mid \mathcal{C}_{0:n}) \tag{4}$$

where $\mathcal{M}_n \mapsto q(\mathcal{M}_{n+1} \mid \mathcal{M}_n)$ models the likelihood of generating the next workspace instance.

## 2.2 Enabling Recursive Updates: A State Space Approach (The Reconciler Framework)

Given a single text input $\mathcal{C}_0$, GSW models the workspace instance $\mathcal{M}_0$ as $P(\mathcal{M}_0|\mathcal{C}_0)$. We seek to compute: $P(\mathcal{M}_n|\mathcal{C}_{0:n})$. For $\mathcal{M}_1$, we introduce $\mathcal{W}_1$, an intermediate representation to decompose $P(\mathcal{M}_1|\mathcal{C}_0, \mathcal{C}_1)$ into parts:

$$\begin{aligned} &P(\mathcal{M}_1 \mid \mathcal{C}_0, \mathcal{C}_1) \\ &\quad = \sum_{\mathcal{M}_0, \mathcal{W}_1} P(\mathcal{M}_1 \mid \mathcal{M}_0, \mathcal{W}_1) \\ &\qquad \times P(\mathcal{M}_0 \mid \mathcal{C}_0) P(\mathcal{W}_1 \mid \mathcal{C}_1) \end{aligned} \tag{5}$$

Here, we assume conditional independence between the workspace state $\mathcal{M}_0$ and the intermediate representation $\mathcal{W}_1$ given the context sequence, such that:

$$\begin{aligned} &P(\mathcal{M}_0, \mathcal{W}_1 \mid \mathcal{C}_0, \mathcal{C}_1) \\ &\qquad = P(\mathcal{M}_0 \mid \mathcal{C}_0) P(\mathcal{W}_1 \mid \mathcal{C}_1) \end{aligned} \tag{6}$$

where we define $\mathcal{W}_1$ to depend solely on the current context $\mathcal{C}_1$, and $\mathcal{M}_0$ solely on the initial context $\mathcal{C}_0$. For an arbitrary step $n$:

$$\begin{aligned} P(\mathcal{M}_n|\mathcal{C}_{0:n}) = &\sum_{\mathcal{M}_{n-1}, \mathcal{W}_n} P(\mathcal{M}_n|\mathcal{M}_{n-1}, \mathcal{W}_n) \\ &\times P(\mathcal{M}_{n-1}|\mathcal{C}_{0:(n-1)}) P(\mathcal{W}_n|\mathcal{C}_n) \end{aligned} \tag{7}$$

Estimating a workspace instance $\mathcal{M}_n$ involves learning parameterized models for three components: the transition model, the prior workspace, and the context-derived augmentation. The prior workspace $\mathcal{M}_{n-1}$ is recursively computed from previous steps. The augmentation step produces an intermediate representation of the current context $\mathcal{C}_n$. We refer to the model estimating this distribution as the **Operator**. The transition model uses a Markovian assumption to produce the updated workspace instance by reconciling existing workspace semantic maps with new semantic information. We refer to this module as the **Reconciler**. Together, the Operator and Reconciler implement a sequential inference mechanism where the Operator maps each new context $\mathcal{C}_n$ to an intermediate state $\mathcal{W}_n$, and the Reconciler performs a structured update $\mathcal{M}_{n-1} \to \mathcal{M}_n$.

# 3 Question Answering with GSW

Figure 2 illustrates this process: memory construction via Operator and Reconciler modules, followed by retrieval, reranking and QA. As described in the caption, once a working memory instance is constructed, answering a query involves the following steps: the system first matches entities from the query to the GSW, then generates contextual summaries for those matched entities from the workspace, re-ranks the summaries for relevance, and finally passes the top-ranked summaries to an LLM to synthesize the answer.

## 3.1 EpBench: An Episodic Memory Benchmark

Our experiments utilize the Episodic Memory Benchmark (EpBench) (Huet, Houidi, and Rossi 2025), a benchmark specifically designed to evaluate the capabilities of LLMs for episodic memory recall and reasoning over long narratives. Unlike many standard Question Answering (QA) benchmarks (Kočiskỳ et al. 2018; Zhang et al. 2024; Yang et al. 2018) – focusing on localized factual retrieval – EpBench targets core episodic capabilities: remembering specific events situated in unique spatiotemporal contexts and distinguishing between recurring events involving the same actors (Holur et al. 2023, 2022).

| Statistic | Value |
|---|---|
| Number of Chapters | 200 |
| Total Tokens | 102,870 |
| Total Queries (QA Pairs) | 686 |
| Queries by Event Category (0 / 1 / 2 / 3-5 / 6+ Cues) | 180 / 180 / 108 / 128 / 90 |
| Max. Chapters Referenced per Query | 17 |
| Min. Chapters Referenced per Query | 0 |

Table 1: **EpBench-200 Dataset Statistics.**

EpBench documents are structured as synthetic books generated chapter-by-chapter from event templates (detailing date, location, entity, content) sampled from a larger universe, ensuring recurring elements that necessitate disambiguation and temporal tracking. Chapters are generated via LLM prompts and verified for coherence. Moreover, the same time/location/actors (collectively referred to as cues) appear across multiple chapters. For our evaluation, we use both the standard 200-chapter version and the extended 2000 chapter version of the dataset and report its Statistics in Table 1.

## 3.2 Evaluation Metrics

To evaluate model performance on the EpBench dataset's queries (detailed in Section 3.1), we adopt the LLM-as-a-Judge evaluation paradigm (Zheng et al. 2023). For consistency, we strictly follow the LLM-based answer processing and extraction procedure outlined by the EpBench benchmark authors. This approach accounts for the possibility that model responses might be longer or more elaborate than the typically concise ground truth answers. These LLM extracted answers are then used to compute Precision, Recall and F1 scores which we report in Table 2

## 3.3 Baseline Methods

We compare GSW against several baseline approaches: **Vanilla LLM**, standard **Embedding-based RAG** (Karpukhin et al. 2020; Ram et al. 2023) for which we utilized the **Voyage-03**[2] embedding model selected for its strong performance on retrieval benchmarks (Thakur et al. 2021) , and the structured RAG methods **GraphRAG** (Edge et al. 2025), **HippoRAG2**(Gutiérrez et al. 2025b), and **LightRAG** (Guo et al. 2025).

## 3.4 Implementation Details

The GSW **Operator** (Section 2.1) and **Reconciler** (Section 2.2) were implemented by prompting GPT-4o (Hurst et al. 2024) according to task-specific instructions, using temperature set to 0 for deterministic behavior. To ensure fair comparison, we standardized both the maximum context utilization (limited to 17 chapters per query, matching the maximum relevant chapters per query) and the answer generation model (GPT-4o) across all evaluated methods. To generate an answer for a given query, we first identify named entities within the query text. These entities are then matched to corresponding nodes within the current GSW memory ($\mathcal{M}_n$) using simple string matching. Summaries for the matched entities – aggregated from the GSW structure – are then retrieved and re-ranked based on semantic similarity to the query. The final re-ranked summaries are provided to the LLM to answer the query as illustrated in Figure 2.

# 4 Results and Discussion

**QA Performance:** Table 2 presents a comparative analysis of GSW against the baseline methods detailed in Section 3.3 across Precision (P), Recall (R), and F1-Score (F1) metrics, categorized by the number of matching cues per query. Across the aggregated metrics, GSW achieves the highest overall F1-Score (0.850), Precision (0.865), and Recall (0.894), improving overall metrics by more than **10%** over the next-best method. GSW also demonstrates consistent performance across the various Cue categories, achieving the highest score in **16 out of 18** individual metric computations, and ranking second in the remaining two, highlighting its robust performance across varying levels of episodic recall complexity. Particularly noteworthy is GSW's performance in the '6+ Cues' category. *This is the most demanding scenario*, where correct responses can require reasoning across information spanning up to 17 distinct chapters (see Table 1). Even in this complex setting, GSW demonstrates robust efficacy and achieves the highest performance over all metrics: F1:0.834 P:0.891, R:0.822. In particular when compared to HippoRAG2, next most performant in this category, GSW outperforms it by approximately **20%** in recall. *Recall, in particular, measures a framework's ability to map queries to the correct chapter and context*, and it is revealing that for all competing frameworks recall decreases as the number of matching cues increases, whereas the GSW maintains

[2]https://blog.voyageai.com/2024/09/18/voyage-3/

[3]Cost calculated using GPT-4o pricing of $2.50 per million tokens.

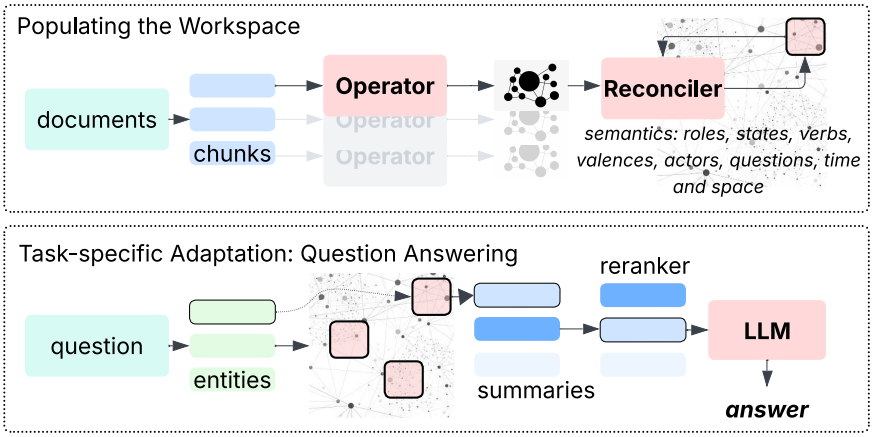


Figure 2: **Episodic Memory Creation and QA:** Figure illustrates the end-to-end process of constructing a workspace and question answering from the workspace. *(top)* Large-scale text is segmented into semantically coherent chunks. Each chunk is processed by the *Operator* model to generate a local workspace instance, represented as a semantic graph. These instances are incrementally integrated by the *Reconciler* resulting in a unified Global Memory. *(bottom)* During question answering, the system retrieves relevant portions of this memory by matching named entities in the query to identifiers in the semantic network. For each match, it reconstructs episodic summaries—contextual recreations of past situations—which are re-ranked and passed to an LLM to generate the final answer.

consistently strong performance, highlighting the strength of its structured representation in storing episodic information. Finally, the Vanilla LLM is consistently the poorest performing baseline (e.g overall F1 Score of 0.642) reaffirming the inherent difficulty of the episodic QA task and the necessity of specialized memory frameworks like the GSW.

**Scalability on EpBench-2000**: To assess the scalability of our method, we evaluate GSW on the EpBench-2000 dataset, which increases the corpus size by 10 fold. The results, presented in Table 4, show that GSW maintains its performance lead by achieving an overall F1-score of 0.773, which is **15% higher** than the strongest baseline (embedding RAG), and **22% higher** than other structured RAG methods. Thus, GSW's advantages in recall and reasoning persist even at a significantly larger scale.

**Token Efficiency:** Beyond query performance, GSW demonstrates substantial improvements in token efficiency, as detailed in Table 3, which presents the average number of context tokens supplied to the LLM per query, and the corresponding cost for all compared methods. GSW achieves a remarkable **51%** reduction in token usage when compared to the next most token-efficient baseline (GraphRAG). This advantage is even more pronounced when compared to stronger performing baselines such as Embedding RAG and HippoRAG2, against which GSW offers a token reduction of nearly **59%**. GSW's efficient approach to query resolution contributes to the reduction in token count: Rather than passing entire chapters or raw document chunks, GSW utilizes its semantic structure to generate entity-specific summaries, thereby providing only targeted query-specific information to the LLM. This focused contextual information also reduces hallucinations as supported by the GSW's leading performance in the '0 Cues' category, where no matching cues are present in the source document.

Several additional **ablation studies** are presented in the full version, including the removal of identifier types (e.g., temporal and spatial tags), evaluations on a shortened version of the EpBench dataset, and comparisons across different retrieval strategies. These experiments highlight the contribution of each component in the GSW architecture and underscore the importance of principled memory querying. For qualitative insights into GSW's behavior and outputs, see the full version.

## 5 Related Work

The relevant literature has been discussed in the Introduction, and a detailed literature review is included in the full version. While modern LLMs offer increasingly large context windows, processing quality degrades with extreme lengths (Leng et al. 2024; Hsieh et al. 2024), with performance notably dipping for information in the middle of long contexts (Liu et al. 2023). This makes reliable episodic tracking challenging when relying solely on native context windows.

Retrieval-Augmented Generation (RAG) (Lewis et al. 2021; Gao et al. 2024; Karpukhin et al. 2020) addresses this by retrieving relevant chunks using dense (Devlin et al. 2019; Reimers and Gurevych 2019; Lee et al. 2025), sparse (Robertson and Zaragoza 2009), or hybrid (Cormack, Clarke, and Buettcher 2009) embeddings. While effective for fact-based QA, standard RAG struggles to connect dispersed information due to chunk-based retrieval (Chen et al. 2023; Merola and Singh 2025). Structured approaches like GraphRAG (Edge et al. 2025), LightRAG (Guo et al. 2025) and HippoRAG (Gutiérrez et al. 2025a,b) mitigate this by modeling relationships and supporting multi-hop reasoning.

Other research efforts have targeted episodic memory more directly. Larimar (Das et al. 2024) proposes modifications to the LLM's attention mechanism, while EM-LLM (Fountas

| Metric | Method | 0 Cues (N=180) | 1 Cue (N=180) | 2 Cues (N=108) | 3-5 Cues (N=128) | 6+ Cues (N=90) | Overall (N=686) |
|---|---|---|---|---|---|---|---|
| **P** | Vanilla LLM | 0.840 ± 0.019 | 0.734 ± 0.021 | 0.735 ± 0.026 | 0.703 ± 0.021 | 0.806 ± 0.028 | 0.766 ± 0.010 |
| | Embedding RAG | 0.906 ± 0.021 | 0.745 ± 0.026 | 0.803 ± 0.028 | 0.823 ± 0.025 | 0.886 ± 0.029 | 0.832 ± 0.012 |
| | GraphRAG (Edge et al. 2025) | 0.950 ± 0.016 | 0.657 ± 0.029 | 0.677 ± 0.034 | 0.753 ± 0.028 | 0.816 ± 0.035 | 0.781 ± 0.013 |
| | HippoRAG2 (Gutiérrez et al. 2025b) | 0.829 ± 0.027 | 0.704 ± 0.029 | **0.817** ± 0.026 | 0.839 ± 0.026 | **0.940** ± 0.020 | 0.812 ± 0.013 |
| | LightRAG (Guo et al. 2025) | 0.946 ± 0.017 | 0.668 ± 0.029 | 0.615 ± 0.036 | 0.695 ± 0.031 | 0.822 ± 0.037 | 0.763 ± 0.014 |
| | GSW (Ours) | **0.978** ± 0.011 | **0.755** ± 0.026 | 0.810 ± 0.027 | **0.878** ± 0.019 | 0.890 ± 0.024 | **0.865** ± 0.010 |
| **R** | Vanilla LLM | 0.840 ± 0.019 | 0.781 ± 0.021 | 0.526 ± 0.021 | 0.419 ± 0.017 | 0.229 ± 0.014 | 0.616 ± 0.011 |
| | Embedding RAG | 0.906 ± 0.021 | 0.863 ± 0.025 | 0.773 ± 0.033 | 0.746 ± 0.027 | 0.624 ± 0.036 | 0.807 ± 0.012 |
| | GraphRAG (Edge et al. 2025) | 0.950 ± 0.016 | 0.764 ± 0.031 | 0.686 ± 0.035 | 0.645 ± 0.026 | 0.537 ± 0.030 | 0.748 ± 0.014 |
| | HippoRAG2 (Gutiérrez et al. 2025b) | 0.829 ± 0.027 | 0.823 ± 0.026 | 0.800 ± 0.029 | 0.749 ± 0.026 | 0.675 ± 0.030 | 0.787 ± 0.013 |
| | LightRAG (Guo et al. 2025) | 0.946 ± 0.017 | 0.716 ± 0.033 | 0.628 ± 0.035 | 0.559 ± 0.029 | 0.458 ± 0.029 | 0.699 ± 0.015 |
| | GSW (Ours) | **0.978** ± 0.011 | **0.863** ± 0.025 | **0.869** ± 0.023 | **0.893** ± 0.015 | **0.822** ± 0.022 | **0.894** ± 0.009 |
| **F1** | Vanilla LLM | 0.840 ± 0.019 | 0.709 ± 0.022 | 0.585 ± 0.021 | 0.476 ± 0.017 | 0.325 ± 0.017 | 0.629 ± 0.010 |
| | Embedding RAG | 0.906 ± 0.021 | 0.726 ± 0.026 | 0.723 ± 0.030 | 0.745 ± 0.026 | 0.680 ± 0.035 | 0.771 ± 0.013 |
| | GraphRAG (Edge et al. 2025) | 0.950 ± 0.016 | 0.625 ± 0.029 | 0.625 ± 0.034 | 0.657 ± 0.026 | 0.607 ± 0.032 | 0.714 ± 0.013 |
| | HippoRAG2 (Gutiérrez et al. 2025b) | 0.829 ± 0.028 | 0.676 ± 0.028 | 0.762 ± 0.028 | 0.754 ± 0.025 | 0.746 ± 0.027 | 0.753 ± 0.013 |
| | LightRAG (Guo et al. 2025) | 0.946 ± 0.017 | 0.594 ± 0.030 | 0.587 ± 0.032 | 0.579 ± 0.028 | 0.561 ± 0.030 | 0.678 ± 0.014 |
| | GSW (Ours) | **0.978** ± 0.011 | **0.744** ± 0.026 | **0.807** ± 0.024 | **0.868** ± 0.016 | **0.834** ± 0.022 | **0.850** ± 0.010 |

Table 2: **GSW performance on Epbench-200 (200-Chapters Book)** Performance is grouped by metric (Precision, Recall, F1-Score) across different numbers of matching cues per query. (N=X) indicates questions per category. Error bars are estimated via bootstrap resampling. Best score in each column for each metric group is **bold**; second best is underlined.

| Method | Avg. Tokens | Avg. Cost[3] |
|---|---|---|
| Vanilla LLM | ∼101,120 | ∼$0.2528 |
| Embedding RAG | ∼8,771 | ∼$0.0219 |
| GraphRAG (Edge et al. 2025) | ∼7,340 | ∼$0.0184 |
| HippoRAG2 (Gutiérrez et al. 2025b) | ∼8,771 | ∼$0.0219 |
| LightRAG (Guo et al. 2025) | ∼40,476 | ∼$0.1012 |
| GSW (Ours) | **∼3,587** | **∼$0.0090** |

Table 3: **GSW's Efficiency**: Average context tokens passed to the LLM per query on EpBench-200, and the estimated cost to answer that query. GSW achieves the best performance (detailed in Table 2) with the significantly lowest token count and cost, as highlighted below. Best score in each column is **bold**; second best is underlined.

| Method | Precision | Recall | F1 |
|---|---|---|---|
| Embedding RAG | 0.827 ± 0.014 | 0.688 ± 0.015 | 0.675 ± 0.015 |
| GraphRAG | 0.761 ± 0.017 | 0.548 ± 0.017 | 0.544 ± 0.017 |
| HippoRAG2 | 0.759 ± 0.016 | 0.648 ± 0.016 | 0.635 ± 0.015 |
| LightRAG | 0.649 ± 0.018 | 0.497 ± 0.017 | 0.494 ± 0.016 |
| GSW (Ours) | **0.830** ± 0.010 | **0.796** ± 0.009 | **0.773** ± 0.009 |

Table 4: **Overall performance on Epbench-2000 (2000-Chapters Book).** The same convention as in Table 2 is followed.

et al. 2024) introduces memory components integrated with open-weight models. While promising, these approaches often require architectural changes or are designed for specific models. In contrast, GSW is a plug-and-play module compatible with any LLM, requiring no specialized training or fine-tuning.

## 6 Concluding Remarks and Limitations

In this work, we introduced the Generative Semantic Workspace (GSW) as a framework for equipping LLMs with human-like episodic memory. Its two core components—the Operator, which interprets local semantics within short context windows, and the Reconciler, which integrates and updates these representations over time—jointly construct a persistent, structured memory. This memory maps raw text into evolving configurations of roles, states, and interactions within a coherent global workspace. On the Episodic Memory Benchmark, GSW outperforms existing approaches in both accuracy and token efficiency, offering a scalable and interpretable alternative to long-context or retrieval-based systems.

Nevertheless, we identify key limitations and avenues for future work. Firstly, GSW's evaluation, while utilizing EpBench for its strengths in spatiotemporal assessment, is constrained by the limited scope of current episodic memory benchmarks in thoroughly probing the complex evolution of actor roles and states within extended narratives; we are developing a more comprehensive benchmark to address this gap. Secondly, the present GSW implementation relies on a strong closed-source LLM (GPT-4o). Empirical validation of promising open-source alternatives (Yang et al. 2024; Grattafiori et al. 2024) within our framework is essential. Expanding GSW to diverse data modalities beyond text is also animportant direction for future work.

## References


Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Botvinick, M. 2008. Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, 12: 201–8.

Chen, H.; Pasunuru, R.; Weston, J.; and Celikyilmaz, A. 2023. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. ArXiv:2310.05029 [cs].

Cormack, G. V.; Clarke, C. L.; and Buettcher, S. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 758–759.

Das, P.; Chaudhury, S.; Nelson, E.; Melnyk, I.; Swaminathan, S.; Dai, S.; Lozano, A.; Kollias, G.; Chenthamarakshan, V.; Jiří; Navrátil; Dan, S.; and Chen, P.-Y. 2024. Larimar: Large Language Models with Episodic Memory Control. ArXiv:2403.11901 [cs].

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2025. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. ArXiv:2404.16130 [cs].

Eichenbaum, H. 2004. Hippocampus: cognitive processes and neural representations that underlie declarative memory. *Neuron*, 44(1): 109–120.

Felleman, D. J.; and Van Essen, D. C. 1991. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1): 1–47.

Fountas, Z.; Benfeghoul, M. A.; Oomerjee, A.; Christopoulou, F.; Lampouras, G.; Bou-Ammar, H.; and Wang, J. 2024. Human-like episodic memory for infinite context llms. *arXiv preprint arXiv:2407.09450*.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; and Wang, H. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. ArXiv:2312.10997 [cs].

George, D.; and Hawkins, J. 2009. Towards a mathematical theory of cortical micro-circuits. *PLoS computational biology*, 5(10): e1000532.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2025. LightRAG: Simple and Fast Retrieval-Augmented Generation. ArXiv:2410.05779 [cs].

Gutiérrez, B. J.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2025a. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. ArXiv:2405.14831 [cs].

Gutiérrez, B. J.; Shu, Y.; Qi, W.; Zhou, S.; and Su, Y. 2025b. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. ArXiv:2502.14802 [cs].

Hassabis, D.; and Maguire, E. A. 2007. Deconstructing episodic memory with construction. *Trends in cognitive sciences*, 11(7): 299–306.

Ho, X.; Duong Nguyen, A.-K.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Holur, P.; Chong, D.; Tangherlini, T.; and Roychowdhury, V. 2023. My side, your side and the evidence: Discovering aligned actor groups and the narratives they weave. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8938–8952. Toronto, Canada: Association for Computational Linguistics.

Holur, P.; Wang, T.; Shahsavari, S.; Tangherlini, T.; and Roychowdhury, V. 2022. Which side are you on? Insider-Outsider classification in conspiracy-theoretic social media. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4975–4987. Dublin, Ireland: Association for Computational Linguistics.

Hong, K.; Troynikov, A.; and Huber, J. 2025. Context Rot: How Increasing Input Tokens Impacts LLM Performance. Technical report, Chroma.

Hsieh, C.-P.; Sun, S.; Kriman, S.; Acharya, S.; Rekesh, D.; Jia, F.; Zhang, Y.; and Ginsburg, B. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? ArXiv:2404.06654 [cs].

Huet, A.; Houidi, Z. B.; and Rossi, D. 2025. Episodic Memories Generation and Evaluation Benchmark for Large Language Models. ArXiv:2501.13121 [cs].

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. Online: Association for Computational Linguistics.

Kingsbury, P.; and Palmer, M. 2002. From TreeBank to PropBank. In González Rodríguez, M.; and Suarez Araujo, C. P., eds., *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA).

Kočiskỳ, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K. M.; Melis, G.; and Grefenstette, E. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6: 317–328.

Lee, C.; Roy, R.; Xu, M.; Raiman, J.; Shoeybi, M.; Catanzaro, B.; and Ping, W. 2025. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. ArXiv:2405.17428 [cs].

Leng, Q.; Portes, J.; Havens, S.; Zaharia, M.; and Carbin, M. 2024. Long context rag performance of large language models. *arXiv preprint arXiv:2411.03538*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. ArXiv:2005.11401 [cs].

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2023. Lost in the Middle: How Language Models Use Long Contexts. ArXiv:2307.03172 [cs].

Louie, K.; and Wilson, M. A. 2001. Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron*, 29(1): 145–156.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3): 419.

Merola, C.; and Singh, J. 2025. Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation. *arXiv preprint arXiv:2504.19754*.

Ólafsdóttir, H. F.; Bush, D.; and Barry, C. 2018. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1): R37–R50.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-Context Retrieval-Augmented Language Models. ArXiv:2302.00083 [cs].

Rasch, B.; and Born, J. 2013. About sleep's role in memory. *Physiological reviews*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv:1908.10084 [cs].

Robertson, S.; and Zaragoza, H. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Rolls, E. T. 2013. A quantitative theory of the functions of the hippocampal CA3 network in memory. *Frontiers in cellular neuroscience*, 7: 98.

Schacter, D. L.; Addis, D. R.; and Buckner, R. L. 2007. Remembering the past to imagine the future: the prospective brain. *Nature reviews neuroscience*, 8(9): 657–661.

Shahsavari, S.; Ebrahimzadeh, E.; Shahbazi, B.; Falahi, M.; Holur, P.; Bandari, R.; R. Tangherlini, T.; and Roychowdhury, V. 2020a. An Automated Pipeline for Character and Relationship Extraction from Readers Literary Book Reviews on Goodreads.com. In *Proceedings of the 12th ACM Conference on Web Science*, WebSci '20, 277–286. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379892.

Shahsavari, S.; Holur, P.; Wang, T.; Tangherlini, T. R.; and Roychowdhury, V. P. 2020b. Conspiracy in the time of corona: Automatic detection of emerging COVID-19 conspiracy theories in social media and the news. *Journal of Computational Social Science*, 3(2): 279–317.

Tangherlini, T. R.; Shahsavari, S.; Shahbazi, B.; Ebrahimzadeh, E.; and Roychowdhury, V. P. 2020. An automated pipeline for the discovery of conspiracy and conspiracy theory narrative frameworks: Bridgegate, Pizzagate and storytelling on the web. *PLOS ONE*, 15(6): e0233879.

Teyler, T. J.; and DiScenna, P. 1986. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2): 147.

Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; and Gurevych, I. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. arXiv:2108.00573.

Tulving, E. 1972. Episodic and semantic memory. In *Organization of memory*, xiii, 423–xiii, 423. Oxford, England: Academic Press.

Tulving, E. 2002. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1): 1–25.

Wilson, M. A.; and McNaughton, B. L. 1994. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172): 676–679.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Zhang, X.; Chen, Y.; Hu, S.; Xu, Z.; Chen, J.; Hao, M.; Han, X.; Thai, Z.; Wang, S.; Liu, Z.; and Sun, M. 2024. ınftyBench: Extending Long Context Evaluation Beyond 100K Tokens. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15262–15277. Bangkok, Thailand: Association for Computational Linguistics.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. ArXiv:2306.05685 [cs].