World Model Driven Episodic Memory for LLMs

Shreyas Rajesh **Pavan Holur** **Chenda Duan** **David Chong** **Vwani Roychowdhury**
University of California, Los Angeles
{ shreyasrajesh38, pholur, chenda, davidchong13807, 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) [26] 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 [40, 24]. 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 [20, 21, 11, 19].

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Bridging Language, Agent, and World Models for Reasoning and Planning.

These methods have primarily been evaluated on fact-rich documents such as Wikipedia pages [81, 23, 70]. 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 $\rightarrow$ arraigned $\rightarrow$ released; startup $\rightarrow$ unicorn $\rightarrow$ acquired) while interacting across **space and time**.

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* [71, 72] 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 [60, 22].

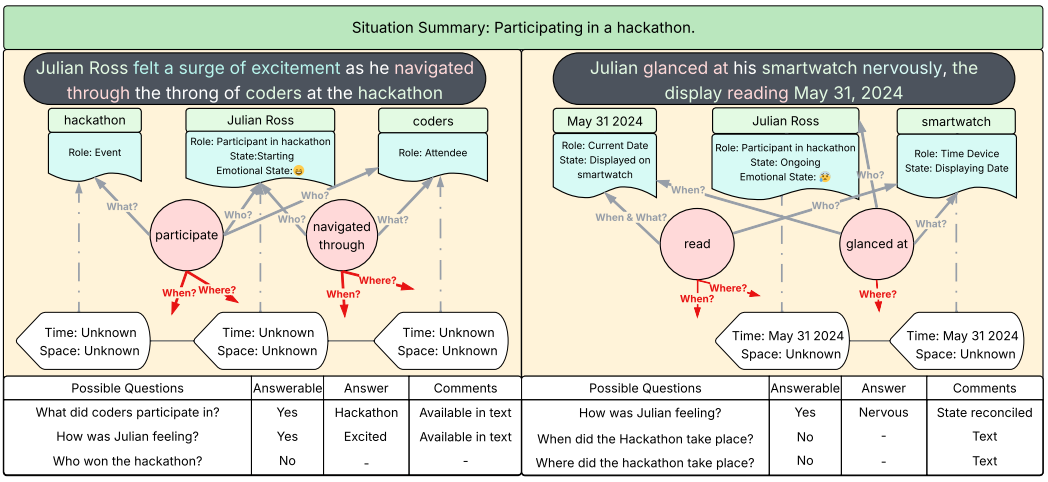
In this work, we introduce the **Generative Semantic Workspace** (GSW), a unifying computational framework for modeling world knowledge as structured, probabilistic semantics in the era of Large Language Models (LLMs). GSW formalizes how an intelligent agent—human or artificial—constructs and updates 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) [26], 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." A visualization of the steps of how GSW constructs its working memory is shown in Figures 1, 2 and 3.

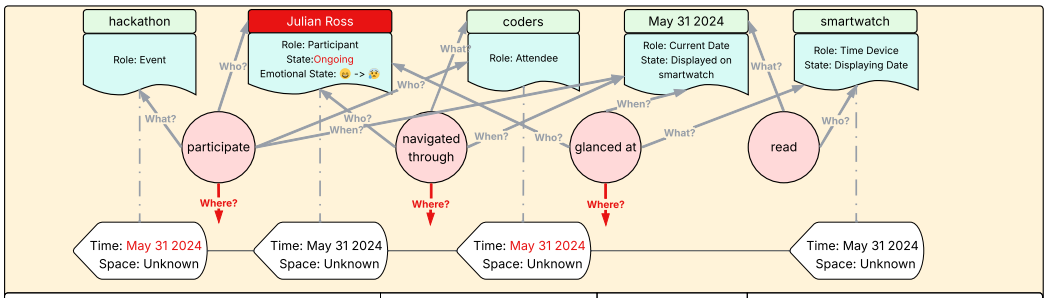
When presented with a task such as "List all the unique locations and dates where Carter Stewart made presentations at Scientific Conference events." a query resolution module (see Section 3) searches through the GSW constructed from all 200 documents and identifies entities mentioned in the query (e.g., 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



Possible Questions	Answerable	Answer	Comments
What did coders participate in?	Yes	Hackathon	Available in text
How was Julian feeling?	Yes	Excited	Available in text
Who won the hackathon?	No	-	-

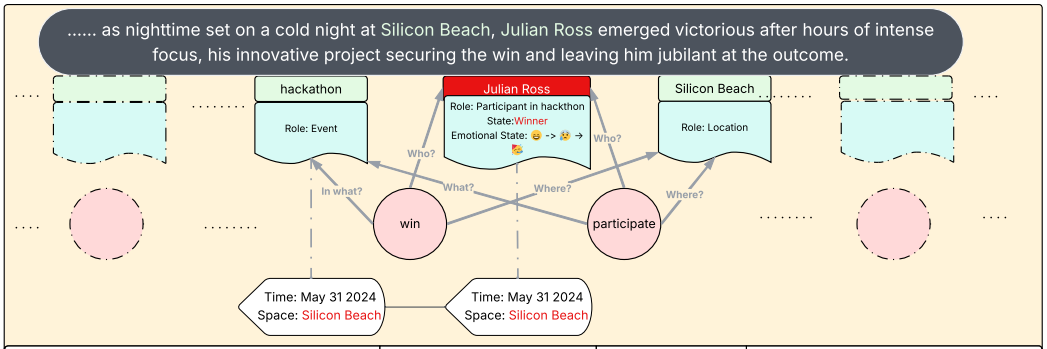
Possible Questions	Answerable	Answer	Comments
How was Julian feeling?	Yes	Nervous	State reconciled
When did the Hackathon take place?	No	-	Text
Where did the hackathon take place?	No	-	Text

Figure 1: **Operator example:** Operator instances of two different chunks, as the GSW framework processes a story.



Possible Questions	Answerable	Answer	Comments
When did Julian Ross participant in Hackathon?	Yes	May 31 2024	Entitity reconciliation
When did the Hackathon take place?	Yes	May 31 2024	Space/Time Reconciliation
Where did the Hackathon take place?	No		-
Who won the Hackathon?	No		-

Figure 2: **Reconciler example:** Reconciled result of the two chunks presented in Fig 1



Possible Questions	Answerable	Answer	Comments
When did Julian Ross participate in Hackathon?	Yes	May 31 2024	Entitity reconciliation
When did the Hackathon take place?	Yes	May 31 2024	Space/Time Reconciliation
Where did the Hackathon take place?	Yes	Silicon Beach	Space/Time Reconciliation
Who won the Hackathon?	Yes	Julian Ross	Entity Reconciliation

Figure 3: **Final GSW:** A portion of the final reconciled GSW

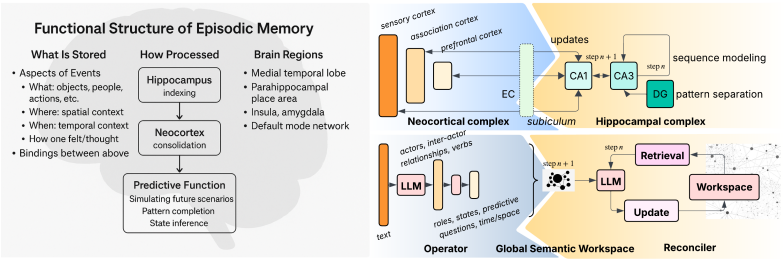


Figure 4: **Unifying Brain-Inspired and Generative Semantics for Episodic Memory Modeling** The hippocampal complex (DG, CA3, CA1) and neocortical regions (NC) inspire the *Reconciler* (retrieval, workspace, update) and *Operator* (LLM-driven semantic role extraction), respectively. The neocortical complex, 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.

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 (see **Appendix C**) GraphRAG's [11] 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 [21] 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 **Section 5**. Finally, limitations and future work are discussed in **Section 6**, and a detailed **Appendix** 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 [17, 4, 14]. The hippocampus, especially the CA3 module, plays a complementary role by binding these representations into coherent spatiotemporal sequences [68, 58, 13]. During sleep, this 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 [51, 41, 78]. This back and forth supports both persistence and prediction of memory [45, 55], key features of episodic memory.

Motivated by this biological architecture (see Fig 4), 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. This process of building an evolving model is illustrated with a detailed end-to-end example in **Appendix B**.

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 [30] and FrameNet [3] 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, and *its approach is validated in **Appendix H**, where a human evaluation shows a strong preference for the GSW semantic maps over those from frameworks like PropBank and FrameNet.*

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 π_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:

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

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:

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

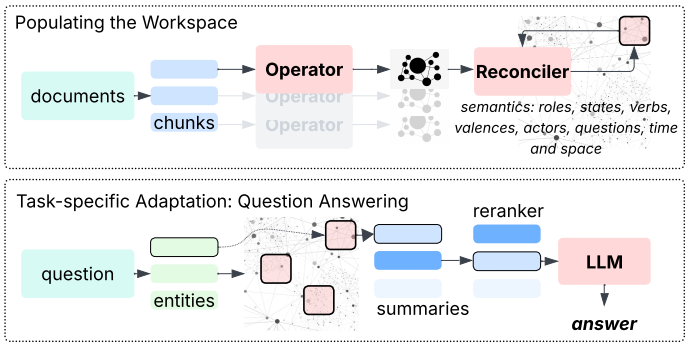


Figure 5: **Episodic Memory Creation and QA:** Figure illustrates the end-to-end process of constructing a workspace and question answering from the workspace. *(top)* Large-scale text is segmented into semantically coherent chunks. Each chunk is processed by the *Operator* model to generate a local 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.

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:

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

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 5 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) [26], 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 [31, 83, 81] – 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.

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

Table 1: **Dataset Statistics:** Statistics for the EpBench Dataset ("Long Book" Version) used in Experiments.

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

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 **Appendix F**.

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 [84]. 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** [29, 54] for which we utilized the **Voyage-03**[1] embedding model selected for its strong performance on retrieval benchmarks [69, 48] , and the structured RAG methods **GraphRAG** [11], **HippoRAG2**[21], and **LightRAG** [19]. We detail the hyper-parameter settings for all baselines in **Appendix E** .

3.4 Implementation Details

The GSW **Operator** (Section 2.1) and **Reconciler** (Section 2.2) were implemented by prompting GPT-4o [27] 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. The complete prompts are provided in **Appendix A**, and API interactions were managed using the Bespoke Curator library [44], indexing costs are reported in **Appendix G**. 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 5. An end-to-end QA example is presented in **B**.

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

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

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

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.

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	$\underline{0.745} \pm 0.026$	0.803 ± 0.028	0.823 ± 0.025	0.886 ± 0.029	$\underline{0.832} \pm 0.012$
	GraphRAG [11]	$\underline{0.950} \pm 0.016$	0.657 ± 0.029	0.677 ± 0.034	0.753 ± 0.028	0.816 ± 0.035	0.781 ± 0.013
	HippoRAG2 [21]	0.829 ± 0.027	0.704 ± 0.029	$\mathbf{0.817} \pm 0.026$	$\underline{0.839} \pm 0.026$	$\mathbf{0.940} \pm 0.020$	0.812 ± 0.013
	LightRAG [19]	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)	$\mathbf{0.978} \pm 0.011$	$\mathbf{0.755} \pm 0.026$	$\underline{0.810} \pm 0.027$	$\mathbf{0.878} \pm 0.019$	$\underline{0.890} \pm 0.024$	$\mathbf{0.865} \pm 0.010$
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	$\underline{0.863} \pm 0.025$	0.773 ± 0.033	0.746 ± 0.027	0.624 ± 0.036	$\underline{0.807} \pm 0.012$
	GraphRAG [11]	$\underline{0.950} \pm 0.016$	0.764 ± 0.031	0.686 ± 0.035	0.645 ± 0.026	0.537 ± 0.030	0.748 ± 0.014
	HippoRAG2 [21]	0.829 ± 0.027	0.823 ± 0.026	$\underline{0.800} \pm 0.029$	$\underline{0.749} \pm 0.026$	$\underline{0.675} \pm 0.030$	0.787 ± 0.013
	LightRAG [19]	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)	$\mathbf{0.978} \pm 0.011$	$\mathbf{0.863} \pm 0.025$	$\mathbf{0.869} \pm 0.023$	$\mathbf{0.893} \pm 0.015$	$\mathbf{0.822} \pm 0.022$	$\mathbf{0.894} \pm 0.009$
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	$\underline{0.726} \pm 0.026$	0.723 ± 0.030	0.745 ± 0.026	0.680 ± 0.035	$\underline{0.771} \pm 0.013$
	GraphRAG [11]	$\underline{0.950} \pm 0.016$	0.625 ± 0.029	0.625 ± 0.034	0.657 ± 0.026	0.607 ± 0.032	0.714 ± 0.013
	HippoRAG2 [21]	0.829 ± 0.028	0.676 ± 0.028	$\underline{0.762} \pm 0.028$	$\underline{0.754} \pm 0.025$	$\underline{0.746} \pm 0.027$	0.753 ± 0.013
	LightRAG [19]	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)	$\mathbf{0.978} \pm 0.011$	$\mathbf{0.744} \pm 0.026$	$\mathbf{0.807} \pm 0.024$	$\mathbf{0.868} \pm 0.016$	$\mathbf{0.834} \pm 0.022$	$\mathbf{0.850} \pm 0.010$

Table 3: **GSW's Efficiency**: Average context tokens passed to the LLM per query on EpBench, 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	Avg. Tokens	Avg. Cost[2]
Vanilla LLM	∼101,120	∼$0.2528
Embedding RAG	∼8,771	∼$0.0219
GraphRAG [11]	∼7,340	∼$0.0184
HippoRAG [21]	∼8,771	∼$0.0219
LightRAG [19]	∼40,476	∼$0.1012
GSW (Ours)	∼**3,587**	∼**$0.0090**

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 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

Table 4: **Overall performance on Epbench-2000 (2000-Chapters Book).** The same convention as in Table 2 is followed. For a more descriptive full table please refer to **Appendix E**.

Method	Precision	Recall	F1
Embedding RAG	$\underline{0.827} \pm 0.014$	$\underline{0.688} \pm 0.015$	$\underline{0.675} \pm 0.015$
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)	$\mathbf{0.830} \pm 0.010$	$\mathbf{0.796} \pm 0.009$	$\mathbf{0.773} \pm 0.009$

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. Due to space constraints, the full breakdown table by cue category is provided in **Appendix E**.

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 (Prompt in **Appendix A**), thereby providing only targeted query-specific information to the LLM as illustrated in **Appendix B**. 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 **Appendix E**, further qualitative insights into GSW's behaviour and outputs are presented in **Appendix C**.

5 Related Work

The relevant literature has been discussed in the Introduction, and a detailed literature review is included in **Appendix F**. To summarize, Retrieval-Augmented Generation (RAG) [36, 16, 29] retrieves relevant chunks from indexed documents using dense [9, 56, 32], sparse [57], or hybrid [7] embeddings. While effective for fact-based QA, standard RAG struggles to connect dispersed information due to its reliance on chunk-based retrieval [6, 47]. Structured approaches like GraphRAG[11], LightRAG[19] and HippoRAG[20, 21] mitigate this by modeling relationships and supporting multi-hop reasoning.

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 specifically address this gap. Secondly, the present GSW implementation relies on a strong closed-source LLM (GPT-4o). Empirical validation of promising open-source alternatives [80, 18, 67] within our Operator-Reconciler architecture is therefore essential.

References

[1] Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In Teruko Mitamura, Eduard Hovy, and Martha Palmer, editors, *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-2907. URL https://aclanthology.org/W14-2907.

[2] Collin F Baker. Framenet: Frame semantic annotation in practice. *Handbook of Linguistic Annotation*, pages 771–811, 2017.

[3] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

[4] Matthew Botvinick. Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, 12:201–8, 06 2008. doi: 10.1016/j.tics.2008.02.009.

[5] David Chanin. Open-source frame semantic parsing. *arXiv preprint arXiv:2303.12788*, 2023.

[6] Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading, October 2023. URL http://arxiv.org/abs/2310.05029. arXiv:2310.05029 [cs].

[7] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 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*, pages 758–759, 2009.

[8] Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarath Swaminathan, Sihui Dai, Aurélie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Jiří, Navrátil, Soham Dan, and Pin-Yu Chen. Larimar: Large Language Models with Episodic Memory Control, August 2024. URL http://arxiv.org/abs/2403.11901. arXiv:2403.11901 [cs].

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[10] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004.

[11] 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, February 2025. URL http://arxiv.org/abs/2404.16130. arXiv:2404.16130 [cs].

[12] Howard Eichenbaum. A cortical–hippocampal system for declarative memory. *Nature Reviews Neuroscience*, 1(1):41–50, October 2000. ISSN 1471-0048. doi: 10.1038/35036213. URL https://www.nature.com/articles/35036213. Publisher: Nature Publishing Group.

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

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

[15] Zafeirios Fountas, Martin Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. Human-like episodic memory for infinite context llms.

[16] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. URL http://arxiv.org/abs/2312.10997. arXiv:2312.10997 [cs].

[17] Dileep George and Jeff Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS computational biology*, 5(10):e1000532, 2009.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[19] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation, April 2025. URL http://arxiv.org/abs/2410.05779. arXiv:2410.05779 [cs].

[20] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models, January 2025. URL http://arxiv.org/abs/2405.14831. arXiv:2405.14831 [cs].

[21] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models, February 2025. URL http://arxiv.org/abs/2502.14802. arXiv:2502.14802 [cs].

[22] Demis Hassabis and Eleanor A Maguire. Deconstructing episodic memory with construction. *Trends in cognitive sciences*, 11(7):299–306, 2007.

[23] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.580.

[24] Kelly Hong, Anton Troynikov, and Jeff Huber. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma, July 2025. URL https://research.trychroma.com/context-rot.

[25] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the Real Context Size of Your Long-Context Language Models?, August 2024. URL http://arxiv.org/abs/2404.06654. arXiv:2404.06654 [cs].

[26] Alexis Huet, Zied Ben Houidi, and Dario Rossi. Episodic Memories Generation and Evaluation Benchmark for Large Language Models, January 2025. URL http://arxiv.org/abs/2501.13121. arXiv:2501.13121 [cs].

[27] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[28] Richard Johansson and Pierre Nugues. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78, 2008.

[29] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

[30] Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In Manuel González Rodríguez and Carmen Paz Suarez Araujo, editors, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/283.pdf.

[31] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[32] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models, February 2025. URL http://arxiv.org/abs/2405.17428. arXiv:2405.17428 [cs].

[33] Kalev Leetaru and Philip A. Schrodt. Gdelt: Global data on events, location, and tone. *ISA Annual Convention*, 2013. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.686.6605.

[34] Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. Long Context RAG Performance of Large Language Models, November 2024. URL http://arxiv.org/abs/2411.03538. arXiv:2411.03538 [cs].

[35] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[36] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, April 2021. URL `http://arxiv.org/abs/2005.11401`. arXiv:2005.11401 [cs].

[37] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[38] Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.422. URL `https://aclanthology.org/2021.emnlp-main.422`.

[39] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1641–1651, 2020.

[40] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts, November 2023. URL `http://arxiv.org/abs/2307.03172`. arXiv:2307.03172 [cs].

[41] Kenway Louie and Matthew A Wilson. Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron*, 29(1):145–156, 2001.

[42] John B Lowe. A frame-semantic approach to semantic annotation. In *Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[43] Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.217. URL `https://aclanthology.org/2021.acl-long.217`.

[44] Ryan* Marten, Trung* Vu, Charlie Cheng-Jie Ji, Kartik Sharma, Shreyas Pimpalgaonkar, Alex Dimakis, and Maheswaran Sathiamoorthy. Curator: A tool for synthetic data creation. `https://github.com/bespokelabsai/curator`, January 2025.

[45] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 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, 1995.

[46] Andrew Mellor. The temporal event graph. *Journal of Complex Networks*, 6(4):639–659, October 2017. ISSN 2051-1329. doi: 10.1093/comnet/cnx048. URL `http://dx.doi.org/10.1093/comnet/cnx048`.

[47] Carlo Merola and Jaspinder Singh. Reconstructing context: Evaluating advanced chunking strategies for retrieval-augmented generation. *arXiv preprint arXiv:2504.19754*, 2025.

[48] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[49] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 104–111, 2002.

[50] Joakim Nivre. Dependency parsing. *Language and Linguistics Compass*, 4(3):138–152, 2010.

[51] H Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.

[52] Martha Palmer. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15. GenLex-09, Pisa, Italy, 2009.

[53] Martha Palmer, Daniel Gildea, and Nianwen Xue. *Semantic role labeling*. Morgan & Claypool Publishers, 2011.

[54] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-Context Retrieval-Augmented Language Models, August 2023. URL http://arxiv.org/abs/2302.00083. arXiv:2302.00083 [cs].

[55] Björn Rasch and Jan Born. About sleep's role in memory. *Physiological reviews*, 2013.

[56] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, August 2019. URL http://arxiv.org/abs/1908.10084. arXiv:1908.10084 [cs].

[57] Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669, 1554-0677. doi: 10.1561/1500000019. URL http://www.nowpublishers.com/article/Details/INR-019.

[58] Edmund T Rolls. A quantitative theory of the functions of the hippocampal ca3 network in memory. *Frontiers in cellular neuroscience*, 7:98, 2013.

[59] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval, January 2024. URL http://arxiv.org/abs/2401.18059. arXiv:2401.18059 [cs].

[60] Daniel L Schacter, Donna Rose Addis, and Randy L Buckner. Remembering the past to imagine the future: the prospective brain. *Nature reviews neuroscience*, 8(9):657–661, 2007.

[61] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[62] Lei Shi and Rada Mihalcea. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 100–111, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-30586-6.

[63] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255, 2019.

[64] Elizabeth Spaulding, Kathryn Conger, Anatole Gershman, Rosario Uceda-Sosa, Susan Windisch Brown, James Pustejovsky, Peter Anick, and Martha Palmer. The DARPA Wikidata overlay: Wikidata as an ontology for natural language processing. In Harry Bunt, editor, *Proceedings of the 19th Joint ACL-ISO Workshop on Interoperable Semantics (ISA-19)*, pages 1–10, Nancy, France, June 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.isa-1.1.

[65] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[66] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[67] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[68] Timothy J Teyler and Pascal DiScenna. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2):147, 1986.

[69] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

[70] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL https://arxiv.org/abs/2108.00573.

[71] Endel Tulving. Episodic and semantic memory. In *Organization of memory*, pages xiii, 423–xiii, 423. Academic Press, Oxford, England, 1972.

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

[73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[74] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 75–78, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933727. doi: 10.1145/1124772.1124784. URL https://doi.org/10.1145/1124772.1124784.

[75] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

[76] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*, 2019.

[77] Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. *arXiv preprint arXiv:2406.11230*, 2024.

[78] Matthew A Wilson and Bruce L McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172):676–679, 1994.

[79] Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019.

[80] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[81] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[82] Qiusi Zhan, Sha Li, Kathryn Conger, Martha Palmer, Heng Ji, and Jiawei Han. Glen: General-purpose event detection for thousands of types. *arXiv preprint arXiv:2303.09093*, 2023.

[83] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ınftyBench: Extending Long Context Evaluation Beyond 100K Tokens. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.814. URL https://aclanthology.org/2024.acl-long.814/.

[84] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. URL http://arxiv.org/abs/2306.05685. arXiv:2306.05685 [cs].

Our technical appendix is structured as follows:

1. Appendix A: Prompts to LLM.
2. Appendix B: GSW QA Example.
3. Appendix C: Qualitative Analysis of GSW performance.
4. Appendix D: Further Implementation Details.
5. Appendix E: Ablation studies.
6. Appendix F: Related work on Memory Augmentation for LLMs.
7. Appendix G: Computation Cost and Resources to build the GSW.
8. Appendix H: Related Computational Models of Workspaces.

A Prompts to the LLM

In the following section we describe the prompts used by each component of our GSW framework.

A.1 Operator

We present the prompt to generate operator representation in Fig 6 and 7. The full prompt is considerably longer and includes detailed instructions for each task. For brevity, we have included the introduction and first task in full, with summaries of the remaining tasks. The complete prompt is available in our code repository.

User Prompt:

You are required to perform the operator extraction, you should follow the following steps:
Task 1: Actor Identification
Your first task is to identify all actors from the given context. An actor can be:
1. A person (e.g., directors, authors, family members)
2. An organization (e.g., schools, festivals)
3. A place (e.g., cities, countries)
4. A creative work (e.g., films, books)
5. A temporal entity (dates, years)
6. A physical object or item (e.g., artifacts, products)
7. An abstract entity (e.g., awards, concepts that function as actors)
Guidelines for Actor Extraction:
- Ground actor extraction in the given situation (<situation>) and the background context (<background_context>).
- It is crucial that you follow the above, since we will attempt to merge relevant actors across chunks in the next step.
- If an entity mentioned in the <input_text> (e.g., 'the journey', 'the event', 'the project') is clearly a direct reference to the overall <situation>, you should name the extracted actor based on the situation description itself.
- Include all mentioned dates as temporal entities
- Do not include phrases or complete sentences
- Extract each actor only once, even if mentioned multiple times
[Further guidelines omitted for brevity]
Task 2: Role Assignment
[Description of role assignment task]
Task 3: State Identification
[Description of state identification task]
Task 4: Explicit Verb Phrase Identification
[Description of verb phrase identification task]
Task 4.5: Implicit Action Phrase Inference
[Description of implicit action phrase inference task]
Task 5: Prototypical Semantic Role Question Generation
[Description of semantic role question generation task]
Task 6: Answer Mapping and Actor Connection
[Description of answer mapping task]

Inputs:

Input Text: " Input chunk to be processed by the operator"
Background Context: " This chunk places the chunking within the entire document, providing context to the chunk.
Situation: " The situation that is presented in this chunk"

Figure 6: LLM prompt for Operator extraction. [3]

A.2 Reconciler

We present the prompt to reconcile unanswered queries with incoming context in Fig 8

[3] Background context generated according to contextual chunking by Anthropic, see `https://www.anthropic.com/news/contextual-retrieval`.

User Prompt:

You are a helpful assistant that is an expert at understanding spatio-temporal relationships between entities.
You will be given a list of entities along with the context of the narrative in which they appear.
Your task is to link entities that share a spatio-temporal relationship.

Read the `Text Chunk` and examine the entities in the `Operator Output`. Identify groups of entity IDs that share the same location (spatial context) or the same time/date (temporal context) based on the events described.
The entities have the following attributes:
** `id`: (String) The entity ID.*
** `name`: (String) The entity name.*
** `roles`: A role is a situation-relevant descriptor (noun phrase) that describes how an actor functions or exists within the context. Roles define the potential relationships an actor can have with other actors.*
** `states`: A state is a condition or description (using adjectives or verb phrases) that characterizes how an actor exists in their role at a specific point. States provide additional context about the actor's condition, status, or situation.*

Return a JSON object with a single key "spatio_temporal_links". The value should be a list of link objects. Each link object must have:
** `linked_entities`: (List of Strings) Entity IDs sharing the context (e.g., `["e1", "e2", "e3"]`).*
** `tag_type`: (String) Either "spatial" or "temporal".*
** `tag_value`: (String or Null)*
* * If the specific location/time/date is mentioned in the `Text Chunk` for this group, extract it.*
* * Otherwise, use `null`.*

Inputs:

Input Text: " Input chunk used to perform linking"
Operator Output: "Operator output for above chunk"

Figure 7: LLM prompt for Space Time coupling.

User Prompt:

You are an expert question answering system. Analyze the provided text and answer the specified questions based ONLY on the text. Provide answers in the specified JSON format.

Your task is to determine if the Text Chunk provides a specific answer for any of these unanswered questions. Base your answers ONLY on the provided Text Chunk.
Respond ONLY with a JSON list containing objects for the questions you can now answer. Each object should have:
- "question_id": The ID of the question being answered.
- "answer_text": The specific text snippet from the Text Chunk that answers the question.
- "answer_entity_id": (Optional) If the answer corresponds exactly to one of the Entities Introduced in This Chunk, provide its ID. Otherwise, omit this field or set it to null.
If no questions can be answered from the text, respond with an empty JSON list: []

Inputs:

Input Text: " Input chunk to be processed by the operator"
Entities: " Entities that were introduced in this chunk"
Unanswered Questions: "Unanswered questions that could be answered "

Figure 8: LLM prompt for QA reconciliation.

A.3 Question Answering

We present the prompt to generate entity summaries which are passed to the answering agent in Fig 9 and the prompt used by the answering agent is presented in Fig 10

User Prompt:

*You are an expert narrative summarizer. Your task is to create a concise, chronological summary paragraph about a single entity based on structured information extracted from a text. Focus on creating a coherent story of the entity's involvement and changes based *only* on the provided timeline.*

INSTRUCTIONS:
1. Write a single paragraph summarizing the key roles, states, experiences, and actions of the entity.
2. Follow the chronological order presented by the Chunk IDs.
3. Integrate the roles, states, and actions into a coherent narrative. Mention key interactions with other entities or objects when provided in the context.
4. You will be provided with spatial and temporal context for entity.
5. These will be provided in the form of a timeline of how they were captured in the text, be sure to incorporate all this spatial and temporal information particularly, provide importance to specific information (like name of place/ explicit dates etc.).
6. Focus on what entity did, what roles they held, their state of being, where they were located, when events happened, and significant events they participated in.
5. Keep the summary concise and factual according to the input. Do not add outside information or make assumptions.
*6. Output *only* the summary paragraph, with no preamble or markdown formatting.*

Inputs:

Input Entity: " Entity with role/state information and space/time links as well as questions answered by it."

Figure 9: LLM prompt for entity summary generation.

User Prompt:

You are a question answering agent that only uses provided information to answer questions.
Your task is to answer questions based exclusively on the knowledge graph information provided. Do not use any external knowledge.
The information provided is extracted from a Generative Semantic Workspace (GSW) representation, which captures:
- Entities: People, places, objects, and concepts
- Verb Phrases: Actions or events involving the entities
- Spatial Relationships: Locations of entities
- Temporal Relationships: Time periods of entities
We use the GSW to extract entity summaries, and you will be provided with these summaries along with graph structure for the GSW for each relevant chapter in order to answer the question.
Always ground your answer in the provided information, and only provide answers for which there is clear evidence in the information provided. If the information needed is not available,
state that you cannot answer based on the available information.
Please answer the following question using ONLY the information provided in the knowledge base extract below.

First determine which chapters are most likely to contain relevant information based on the question, then based on the entity summaries and the graph structure for those chapters, determine the most likely answer. Answers will always be a SINGLE entity representing a person, event, location or time period. It will not be a description or a concept.
QUESTION: questions
KNOWLEDGE BASE INFORMATION: gsw summaries
First provide a reasoning for which chapters are most likely to contain relevant information based on the question.
Then provide a reasoning for which entity is most likely to be the correct answer based on the entity summaries and the graph structure for those chapters.

Inputs:

Question: "Question to be answered"
GSW Summaries: "Summaries produced by the GSW relevant to answer questions"

Figure 10: LLM prompt for final Question Answering.

B GSW QA Example

Figure 11 illustrates the end-to-end question answering (QA) pipeline of the GSW framework, showcasing how a sample query from the EpBench dataset is processed through each stage.

C Qualitative Analysis of GSW performance

This section presents a qualitative analysis of selected queries to further illustrate GSW's superior performance and token efficiency compared to baseline methods, as detailed in Table 6. The chosen queries, whose full text and ground truth answers are provided in Table 5, are representative of varying complexity, with answers requiring the synthesis of information linked to two to seven distinct contextual cues. This detailed examination reveals specific failure modes in baseline approaches that GSW is naturally suited to overcome.

For instance, GraphRAG, which generates summaries of varying detail from source documents, frequently struggles with information loss and often provides an excessive volume of irrelevant context to the LLM, increasing the likelihood of hallucinations. This limitation is particularly noticeable in its handling of queries Q3 and Q4 (see Table 6). These queries demand precise spatial and temporal understanding of events, aspects that GraphRAG's summarization process does not natively or consistently capture, leading to missing information or inaccuracies in its responses.

HippoRAG2, on the other hand, processes every query through its knowledge graph –constructed by connecting semantically similar phrases across triples derived from all the chapters– to identify

Input Query:

Reflect on the experiences of Carter Stewart related to Scientific Conference. List all the unique locations where these events took place, without mentioning the events themselves.

Named Entities :

Carter Stewart, Scientific Conference

Retrieved Summaries:

Chapter 29:

Entity: Carter Stewart - Summary: On January 3, 2026, at Yankee Stadium, Carter Stewart, a performer and mime artist, was preparing for a significant performance....

Chapter 49:

Entity: Carter Stewart - Summary: Entity: Carter Stewart - Summary: On September 22, 2026, during the morning sessions of a scientific conference at the Metropolitan Museum of Art, Carter Stewart took on the role of a presenter,....

Entity: The scientific conference - Summary: The scientific conference, held on September 22, 2026, was a pivotal moment that took place at the Metropolitan Museum of Art. This event was attended by various individuals

Chapter 134:

Entity: Carter Stewart - Summary: On December 25, 2025, Carter Stewart organized a literary-themed festival at Yankee Stadium, stepping onto the field with a sense of pride as both an organizer and participant...

Entity: Carter Stewart's pocket watch - Summary: On December 25, 2025, Carter Stewart's pocket watch, a timekeeping device adorned with intricate clockwork gears and miniature constellations....

Chapter 166:

Entity: Carter Stewart - Summary: On March 23, 2024, Carter Stewart, a researcher and presenter, stepped onto the Bethpage Black Course to present his research findings at a Scientific Conference...

Entity: Scientific Conference - Summary: The Scientific Conference, held at the Bethpage Black Course on March 23, 2024, was an event that buzzed with anticipation as it unfolded in a unique setting—a golf course....

Reranked Summaries:

Chapter 166:

*Entity: Carter Stewart - Summary: On March 23, 2024, Carter Stewart, a researcher and presenter, stepped onto the **Bethpage Black Course** to present his research findings at a Scientific Conference...*

*Entity: Scientific Conference - Summary: The Scientific Conference, held at the **Bethpage Black Course** on March 23, 2024, was an event that buzzed with anticipation as it unfolded in a unique setting—a golf course....*

Chapter 49:

*Entity: Carter Stewart - Summary: Entity: Carter Stewart - Summary: On September 22, 2026, during the morning sessions of a scientific conference at the **Metropolitan Museum of Art**, Carter Stewart took on the role of a presenter,....*

*Entity: The scientific conference - Summary: The scientific conference, held on September 22, 2026, was a pivotal moment that took place at the **Metropolitan Museum of Art**. This event was attended by various individuals*

. . . . } *Other reranked summaries*

Final Answer:

Bethpage Black Course, Metropolitan Museum of Art

Figure 11: **Illustrative example of the GSW QA framework:** First, NER is performed on the input query to identify key entities. In this version of QA implementation these extracted entities are matched to the relevant GSW instances of chapters via string matching, and the entity-specific summaries (see Appendix D.3) from the GSWs are retrieved. Subsequently, these retrieved entity summaries are re-ranked based on their semantic similarity to the input query—a score calculated using cosine similarity between their embeddings and the query's embedding. The figure displays a selection of initially retrieved summaries followed by the top re-ranked summaries. Finally, these re-ranked summaries are passed to an answering LLM, which then produces the final answer. As our considerably smaller average token count shows, our entity summaries are already concise, and only entity-relevant chapters are retrieved. Future implementations could leverage several avenues for further reduction in token counts without compromising performance. For example, in a query involving multiple entities, GSWs that have all the entities could be retrieved and sent to the LLM for a final answer; currently our re-ranking step ranks them at the top but we send summaries from other chapters as well, which is not necessary.

the relevant chapters, and then provides full texts of these chapters as context to the LLM for a final answer. The strength of this approach is that they do not need to perform fine-grained analysis of the text -for example for dates and locations;- as long as their retrieval process gets the right chapter, the onus is on the LLM to retrieve the relevant spatio-temporal information. This is a good bet if the documents themselves are short and the number of documents needed to answer a query are few. In the EpBench data set the document size is around 500 tokens and the number of documents needed to answer some of the questions is 17; since QA cannot know the number of documents needed for any given question, 17 documents (chapters) were sent for each query across all evaluated methods. As observed for queries Q2 and Q4 in Table 6, this strategy of providing full documents can overwhelm the LLM, leading to hallucinations or the failure to pinpoint the correct answer even when the right document with the necessary information is present in the retrieved context. Furthermore, there were instances (e.g., Q3, Q5) where HippoRAG2 failed to retrieve all the pertinent documents required to comprehensively answer the query.

In contrast, GSW's structured representation and targeted summary generation (as detailed in Table 6 showing 'None' for errors and lower token counts) effectively mitigate these issues. The ability of our GSW framework to collate and then structure spatio-temporal information scattered across the length of a document (via reconciliation) is aptly captured in the entity-level summary for Carter Stewart that is retrieved in response to Q2 (first three sentences are shown below):

> On September 22, 2026, during the morning sessions of a scientific conference at the Metropolitan Museum of Art, Carter Stewart took on the role of a presenter, delivering a final presentation that included statistical analysis using presentation boards and holographic projectors.

The necessary information – Carter Stewart, location, and time –in the original document came from three different paragraphs; in fact, Carter Stewart is referred to as "He" until after location and time information is given:

> 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."

Table 5: Selected Queries and Ground Truth Answers for Qualitative Analysis

Query ID	Query Text	Ground Truth Answer
Q1	Consider all events that Jackson Ramos has been involved in. List all the locations where these events took place, without mentioning the events themselves.	High Line, Snug Harbor Cultural Center, Central Park, One World Trade Center, Ellis Island
Q2	Reflect on the experiences of Carter Stewart related to Scientific Conference. List all the unique locations where these events took place, without mentioning the events themselves.	Bethpage Black Course, Metropolitan Museum of Art
Q3	Consider all events that Ezra Edwards has been involved in. List all the locations where these events took place, without mentioning the events themselves.	Water Mill Museum, Port Jefferson, Yankee Stadium, New York Botanical Garden, Brooklyn Bridge, Bethpage Black Course, One World Trade Center
Q4	Recall the events related to Tech Hackathon that occurred on March 23, 2025. List all the locations where these events took place, without describing the events themselves.	Yankee Stadium, Water Mill Museum, Woolworth Building, Queensboro Bridge
Q5	Recall the events related to Tech Hackathon that occurred on November 13, 2026. List all the locations where these events took place, without describing the events themselves.	Trinity Church, Woolworth Building, Statue of Liberty, Fire Island National Seashore

Table 6: Qualitative Performance Comparison on Selected Queries (referencing Query IDs from Table 5)

Query ID	Method	Token Count	Error Description	Analysis/Reason
Q1	GSW	2011	None	NA
	HippoRAG2	9289	None	NA
	GraphRAG	8189	Missing 1 location	Info not available in retrieved context.
Q2	GSW	1568	None	NA
	HippoRAG2	8225	Hallucinated 3 extra locations.	Too much irrelevant information resulted in LLM hallucination.
	GraphRAG	8220	Missed 1 location and Hallucinated 2	All required info present in context but LLM hallucinated.
Q3	GSW	1726	None	NA
	HippoRAG2	8475	Missed 1 location	Info not available in retrieved context.
	GraphRAG	7058	Missed 1 location	Info not available in retrieved context.
Q4	GSW	5530	None	NA
	HippoRAG2	8614	Missed 2 locations	All required info present in context but LLM hallucinated.
	GraphRAG	7936	Missed 3 locations and Hallucinated 1	All required info present in context but LLM hallucinated.
Q5	GSW	6452	None	NA
	HippoRAG2	8355	Missed 1 location	Info not available in retrieved context
	GraphRAG	7936	Missed 2 locations	Info not available in retrieved context.

D Implementation Details

In this section, we provide further implementation details for the GSW as well as baselines implemented.

D.1 Operator

The operator representations are obtained by prompting GPT-4o with the prompt presented in Fig. 6 with a temperature of 0 to reduce stochasticity. Prior to obtaining the operator representations, we perform co-reference resolution at a Chapter level resolution. Chapters are then chunked into smaller text chunks each containing three sentences without overlap between consecutive chunks. Space-Time coupling is performed after the operator representations are obtained by prompting GPT-4o with the prompt presented in Fig. 7 with temperature set to 0 and max generation tokens set to 1000.

D.2 Reconciler

Reconciliation is performed on consecutive chunks of operator representations; for our study, we reconcile all chunks of a particular chapter to produce one reconciled GSW representation per chapter. Roles and states for reconciled entities are time-stamped and stored, and this historical information is subsequently utilized during the generation of entity-level summaries.

When a reconciled entity provides new space/time information, its associated space/time nodes are updated accordingly. All previously recorded space/time information is also time-stamped and preserved to enrich these entity-level summaries. Furthermore, it is important to note that if an update to a space/time node is triggered by one entity, this new spatio-temporal information is propagated to all other entities coupled with that same node; this dynamic is illustrated in Figures 2 and 3.

Finally, the reconciliation process also addresses 'forward-falling questions' —queries identified by previous Operator instances that can now be answered using the integrated information from the reconciled GSW as detailed in Section 2.1. These questions are resolved by prompting GPT-4o with the instructions detailed in Figure 8. For this QA resolution task, the temperature is set to 0 and maximum generation tokens are set to 500.

D.3 QA

Prior to the final question answering (QA) stage, entity-specific summaries are generated using the GSW structure. For each entity, a prompt is constructed incorporating its roles, states, associated spatio-temporal information, and the questions it addresses through verb phrases (as captured in its GSW representation). This summarization prompt, detailed in Figure 9, is processed by GPT-4o with a temperature of 0 and a maximum of 500 generation tokens.

The question answering (QA) process unfolds as follows: First, Named Entity Recognition (NER) is performed on the input question to identify relevant entities for querying the GSW. Based on these extracted entities, basic string matching is used to find corresponding entities within the consolidated GSW representations. Next, the **entity-specific summaries** (generated as described previously) for these matched entities are retrieved and then re-ranked. This re-ranking is based on the cosine similarity between the embeddings of the entity summaries and the embedding of the input query. To ensure consistency, the Voyage-03 model is employed as the embedding model for both the summaries and the query. Finally, these re-ranked summaries are passed to the answering agent (GPT-4o). The context provided to the agent is limited to summaries derived from a maximum of 17 diverse chapters, a constraint applied to maintain consistency across all evaluated methods and to ensure all dataset questions can be addressed. A detailed example of the QA process is presented in Appendix B.

D.4 Baselines

For HippoRAG2 [21], GraphRAG [11], and LightRAG [19], we adhere to each method's default hyperparameters and prompt formats as provided in their respective official implementations. To ensure consistency across baselines, we modify the answering model in HippoRAG2 to use GPT-4o,

aligning it with other evaluated methods. Additionally, we set top-k to 17 for HippoRAG2 to retrieve the top 17 relevant documents to align with the QA settings. The detailed configurations for each baseline are listed in Tables7–9.

Table 7: GraphRAG Baseline Parameter

Setting	Value
Mode	Local
LLM Model	`gpt-4o`
Embedding Model	`text-embedding-3-small`
Response Type	Multiple paragraphs
Max Context Tokens	12000
Text Unit Proportion	0.5
Community Report Proportion	0.1
Top-K Entities	10
Top-K Relationships	10
Include Entity Rank	True
Include Relationship Weight	True
Include Community Rank	False

Table 8: HippoRAG2 Baseline Parameter

Setting	Value
LLM Indexing Model	`gpt-4o-mini`
LLM Answering Model	`gpt-4o`
Embedding Model	`NV-Embed-v2`
QA Top-K	17
Linking Top-K	5
Retrieval Top-K	200

Table 9: LightRAG Baseline Parameter

Setting	Value
LLM Model	`gpt-4o`
Embedding Model	`text-embedding-3-small`
Retrieval Mode	Hybrid
Chunk Token Size	1200
Chunk Overlap Size	100

D.5 Bootstrapping for Evaluation

In our main evaluation for EpBench-200 and EpBench-2000, we represent error bars computed via bootstrap resampling on 1,000 iterations. For each evaluation, we resampled the test set predictions with replacement and computed performance metrics on each bootstrap sample. The LLM judge operated with temperature=0 for deterministic outputs. These standard deviations indicate the variability in scores when different combinations of test examples are weighted through resampling

E Ablation Studies

We present the results of ablation studies we performed on our GSW framework.

E.1 Evaluating the GSW on the Short Book Dataset

Table 10 presents results comparing GSW against Vanilla LLM on the shorter 20-chapter variant of EpBench. Both GSW and Vanilla LLM demonstrate strong performance on this smaller dataset. The Vanilla LLM performs particularly well on this version because the entire context length is approximately 10,000 tokens, which easily fits within the model's context window. Notably, even with this shorter context, we observe that Vanilla LLM begins to struggle relative to GSW as the number of matching cues increases, particularly in the 3-5 cue category where GSW shows superior recall (0.910 vs 0.781) and F1-score (0.857 vs 0.777).

This finding further supports our main results presented in Table 2 of the main paper, as it demonstrates how Vanilla LLM's performance deteriorates with increased context length. While performing competitively on short narratives, Vanilla LLM struggles with the 200-chapter version where context exceeds 100,000 tokens. In contrast, GSW maintains robust performance across both short and long narratives , highlighting the value of our approach.

Table 10: **GSW vs. Vanilla LLM performance on EpBench 20-Chapter "Short Book":** Results are grouped by metric (Precision, Recall, F1-Score) across different numbers of matching cues per query. (N=X) indicates questions per category. Best score in each cell is **bold**.

Metric	Method	0 Cues (N=180)	1 Cue (N=180)	2 Cues (N=72)	3-5 Cues (N=24)	Overall (N=456)
P	Vanilla LLM	0.889	**0.781**	**0.900**	0.799	**0.843**
	GSW (Ours)	**0.939**	0.751	0.804	**0.854**	0.841
R	Vanilla LLM	0.889	**0.919**	0.813	0.781	0.883
	GSW (Ours)	**0.939**	0.856	**0.819**	**0.910**	**0.886**
F1	Vanilla LLM	0.889	**0.812**	**0.821**	0.777	**0.842**
	GSW (Ours)	**0.939**	0.745	0.784	**0.857**	0.834

E.2 Detailed Results on EpBench-2000

The detailed statistics for EpBench-2000 are presented in Table 11. Although the maximum number of chapters referenced per query in the EpBench-2000 dataset reaches 138, we choose to limit the maximum context utilization to 17 chapters per query, maintaining the same configuration applied to EpBench-200 in the main paper. This choice is based on the fact that the 138-chapter scenario represents an extreme outlier, while 17 chapters suffice to address the majority of queries effectively. Furthermore, processing 138 chapters per query would introduce significant computational overhead and inefficiencies, as it requires feeding an excessive volume of text to the model, which could negatively impact both performance and resource utilization. Since we use the same number of chapters per query as in EpBench-200, we therefore expect a very similar token usage.

Table 12 reports the complete set of metrics for GSW and all baselines on the EpBench-2000 dataset, broken down by cue complexity. These results expand upon the summary in the main text, demonstrating that GSW retains its lead across all levels of episodic complexity, and outperforming the strongest baseline by more than **15%** in F1-score and **14%** in recall. The EpBench-2000 experiment further highlights GSW's ability to scale effectively while maintaining strong performance in long-context, high-recall settings.

Table 11: **EpBench-2000 Dataset Statistics.**

Statistic	Value
Number of Chapters	1967
Total Tokens	1,012,097
Total Queries (QA Pairs)	623
Queries by Event Category (0 / 1 / 2 / 3-5 / 6+ Cues)	90 / 165 / 114 / 124 / 130
Max. Chapters Referenced per Query	138
Min. Chapters Referenced per Query	0

E.3 Ablating components of the GSW for Question Answering

Table 13 presents the results of ablating both components of the GSW as well as approaches to retrieval, highlighting the importance of each component and our string matching + reranking retrieval mechanism. We note that while naive string matching achieves almost similar performance to our retrieval method, it consumes almost double the number of tokens.

F Related work on Memory Augmentation for LLMs

Enabling LLMs to effectively process long narratives requires capabilities akin to human episodic memory – constructing and maintaining a dynamic, coherent understanding of events unfolding over space and time [71, 12]. Key to this is the ability to accurately track entities, including their evolving states and roles, and to ground events and answer queries based on specific spatial and

Table 12: **GSW performance on Epbench (2000-Chatpers 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.

Metric	Method	0 Cues (N=90)	1 Cue (N=165)	2 Cues (N=114)	3-5 Cues (N=124)	6+ Cues (N=130)	Overall (N=623)
P	Embedding RAG	0.789 ± 0.043	$\underline{0.751} \pm 0.028$	$\mathbf{0.845} \pm 0.026$	$\underline{0.840} \pm 0.031$	$\mathbf{0.911} \pm 0.025$	$\underline{0.827} \pm 0.014$
	GraphRAG [11]	$\mathbf{0.943} \pm 0.025$	0.747 ± 0.038	0.639 ± 0.040	0.692 ± 0.038	0.795 ± 0.043	0.761 ± 0.017
	HippoRAG2 [21]	0.620 ± 0.051	0.638 ± 0.032	0.803 ± 0.032	0.824 ± 0.028	$\underline{0.893} \pm 0.021$	0.759 ± 0.016
	LightRAG [19]	0.790 ± 0.042	0.534 ± 0.039	0.560 ± 0.040	0.593 ± 0.035	0.787 ± 0.039	0.649 ± 0.018
	GSW (Ours)	$\underline{0.867} \pm 0.0025$	$\mathbf{0.761} \pm 0.0020$	$\underline{0.841} \pm 0.0019$	$\mathbf{0.841} \pm 0.0019$	0.870 ± 0.0019	$\mathbf{0.830} \pm 0.0010$
R	Embedding RAG	0.789 ± 0.043	$\underline{0.764} \pm 0.032$	$\underline{0.795} \pm 0.033$	0.637 ± 0.031	0.480 ± 0.028	$\underline{0.688} \pm 0.015$
	GraphRAG [11]	$\mathbf{0.943} \pm 0.025$	0.492 ± 0.037	0.587 ± 0.039	0.538 ± 0.036	0.321 ± 0.025	0.548 ± 0.017
	HippoRAG2 [21]	0.620 ± 0.050	0.703 ± 0.034	0.769 ± 0.031	$\underline{0.647} \pm 0.029$	$\underline{0.491} \pm 0.026$	0.648 ± 0.016
	LightRAG [19]	0.790 ± 0.042	0.525 ± 0.038	0.549 ± 0.038	0.440 ± 0.033	0.270 ± 0.017	0.497 ± 0.017
	GSW (Ours)	$\underline{0.867} \pm 0.025$	$\mathbf{0.844} \pm 0.019$	$\mathbf{0.864} \pm 0.016$	$\mathbf{0.792} \pm 0.017$	$\mathbf{0.633} \pm 0.017$	$\mathbf{0.796} \pm 0.009$
F1	Embedding RAG	0.789 ± 0.043	$\underline{0.644} \pm 0.031$	$\underline{0.758} \pm 0.032$	0.679 ± 0.031	0.561 ± 0.029	$\underline{0.675} \pm 0.015$
	GraphRAG [11]	$\mathbf{0.943} \pm 0.025$	0.436 ± 0.035	0.547 ± 0.038	0.541 ± 0.036	0.405 ± 0.027	0.544 ± 0.017
	HippoRAG2 [21]	0.620 ± 0.050	0.583 ± 0.031	0.732 ± 0.031	$\underline{0.681} \pm 0.027$	$\underline{0.578} \pm 0.026$	0.635 ± 0.015
	LightRAG [19]	0.790 ± 0.042	0.436 ± 0.034	0.514 ± 0.037	0.463 ± 0.033	0.375 ± 0.021	0.494 ± 0.016
	GSW (Ours)	$\underline{0.867} \pm 0.025$	$\mathbf{0.741} \pm 0.020$	$\mathbf{0.818} \pm 0.017$	$\mathbf{0.789} \pm 0.017$	$\mathbf{0.698} \pm 0.016$	$\mathbf{0.773} \pm 0.009$

Table 13: **Ablation Study of GSW Components on EpBench (200-Chapter Book):** Performance across different event categories (Precision, Recall, F1-Score). (N=X) indicates questions per category. Full GSW model results (at the bottom) are for reference from Table 2 in the main paper.

Metric	GSW Configuration / Ablation	0 Events (N=150)	1 Event (N=150)	2 Events (N=90)	3-5 Events (N=98)	6+ Events (N=60)	Overall (N=548)
P	w/o Space/Time Linking	0.978	0.799	0.814	0.851	0.854	0.868
	QA Input: Verb Phrases	0.939	0.839	0.807	0.896	0.874	0.878
	Retrieval: Str.Match, No reranking	0.967	0.773	0.860	0.872	0.932	0.879
	Retrieval: Emb. Match, No reranking	0.922	0.792	0.797	0.874	0.876	0.855
	Retrieval: NER emb, no reranking	0.944	0.747	0.823	0.872	0.854	0.854
	GSW (Full)	**0.978**	**0.755**	**0.810**	**0.878**	**0.891**	**0.865**
R	w/o Space/Time Linking	0.978	0.800	0.810	0.738	0.723	0.827
	QA Input: Verb Phrases	0.939	0.766	0.644	0.674	0.551	0.747
	Retrieval: Str.Match, No reranking	0.967	0.834	0.850	0.819	0.822	0.867
	Retrieval: Emb. Match, No reranking	0.922	0.820	0.833	0.825	0.781	0.845
	Retrieval: NER emb, no reranking	0.944	0.710	0.750	0.721	0.624	0.768
	GSW (Full)	**0.978**	**0.863**	**0.868**	**0.892**	**0.822**	**0.894**
F1	w/o Space/Time Linking	0.978	0.731	0.764	0.762	0.761	0.811
	QA Input: Verb Phrases	0.939	0.733	0.633	0.719	0.621	0.754
	Retrieval: Str.Match, No reranking	0.967	0.748	0.826	0.823	0.859	0.846
	Retrieval: Emb. Match, No reranking	0.922	0.726	0.788	0.827	0.810	0.817
	Retrieval: NER emb, No reranking	0.944	0.629	0.717	0.748	0.693	0.756
	GSW (Full)	**0.978**	**0.745**	**0.806**	**0.867**	**0.834**	**0.850**

temporal contexts established within the narrative [26]. While LLMs possess remarkable core abilities, achieving this level of sophisticated, stateful reasoning over extended sequences remains a significant challenge. The following sections analyze inherent limitations in common approaches used to provide context to LLMs, evaluating why they often fall short of systematically delivering these specific episodic memory capabilities.

F.1 Leveraging Long context LLMs

One approach to providing LLMs with relevant context is to leverage their increasingly large context windows, potentially feeding the entire long narrative along with a query into the prompt. The rapid

expansion of context lengths, now reaching millions of tokens, has certainly broadened the scope of tasks LLMs can handle by allowing more raw information to be processed simultaneously [66].

However, relying solely on this native processing mechanism faces significant hurdles when evaluated against the demands of episodic memory. Firstly, while context windows are growing, they are not infinite, and extremely long narratives may still exceed even the largest available limits. Secondly, even when a narrative technically fits, processing vast amounts of text for every query is computationally expensive, impacting latency and cost. More fundamentally, processing quality often degrades with extreme context lengths [34, 25, 77]. Research has shown that LLMs can struggle to consistently access and utilize information spread across very long contexts, with performance notably dipping for information located in the middle ("lost in the middle" phenomenon) [40]. Feeding potentially large amounts of irrelevant text alongside the crucial details for a specific episodic query can distract the model and hinder its ability to pinpoint and reason over the necessary information.

Finally, perhaps the most critical limitation for systematic episodic tracking is the inherently unstructured nature of the input context. Even with all the necessary details about entity states, roles, locations, and times present within the text, the LLM lacks explicit mechanisms to structure this information dynamically. It must rely solely on its attention mechanism and in-context learning to piece together relationships, track state changes, and maintain temporal coherence across potentially thousands of tokens. This makes the reliable, systematic tracking required for robust episodic memory challenging and often brittle when relying only on the native context window [26].

F.2 Memory Augmentation for LLMs

To overcome the challenges of static parametric knowledge and the inefficiencies of processing entire documents in context, Retrieval-Augmented Generation (RAG) has become a standard technique [36, 16]. The typical RAG pipeline involves pre-processing a knowledge corpus (e.g., the entire narrative document) into smaller chunks. These chunks are then indexed, commonly using dense vector embeddings obtained from encoder style LLMs[9, 56, 32], though sparse methods like BM25[57] or hybrid approaches are also employed [7]. At inference time, the user query is used to retrieve the top-k most relevant chunks from the index based on a similarity metric (e.g., cosine similarity for dense vectors). These retrieved chunks are then presented as augmented context to an LLM, which generates the final response based on both its parametric knowledge and the retrieved information.[54]

This approach has proven effective for many knowledge-intensive tasks, particularly fact-based question answering where retrieving specific evidence snippets is sufficient [29]. However, when evaluated against the requirements of robust episodic memory recall over long narratives, the limitations of standard RAG become apparent [26]. Firstly, the process of retrieving discrete, potentially disconnected chunks based on local query relevance often **fragments the narrative flow**. This makes it exceedingly difficult for the LLM to reliably follow evolving storylines or track the **changing states and roles of entities** over time, as the necessary context may be spread across multiple chunks that are not retrieved together[6].

Moreover, this fragmentation problem is compounded by the framework being highly sensitive to the initial chunking strategy[47]. Arbitrary chunk boundaries can split crucial information related to an event or an entity's state, leading to information loss during retrieval. For instance, if a character's state changes within a passage, but the chunking algorithm divides this passage at an inopportune point, the complete context of this state change may not be captured in any single retrieved chunk. Optimal chunking is non-trivial and can significantly impact the ability to reconstruct the necessary context for complex episodic reasoning. Consequently, while standard RAG offers efficiency gains over naive long-context processing, its inherent lack of structure and narrative coherence makes it ill-suited for systematically addressing the dynamic, stateful, and context-dependent nature of episodic memory tasks.

Additionally, standard RAG mechanisms based on semantic similarity often struggle with incorporating specific spatio-temporal constraints that are essential for episodic memory. Embeddings typically capture semantic content but may not adequately encode the nuances of time and location, making it difficult to retrieve context relevant to a specific point in time or place mentioned in a query or implied by the narrative history.

F.3 Structured Representations as Memory

Recognizing the limitations of standard RAG, particularly its tendency to fragment narratives and struggle with temporal coherence, recent work has explored incorporating more explicit structure into the retrieval and augmentation process. Instead of treating the source narrative as a flat sequence of independent chunks, these methods attempt to build richer representations that capture relationships or hierarchies within the text, aiming to provide more contextually relevant information to the LLM.

While these structured approaches offer advantages over standard RAG by preserving more relational or hierarchical context and enabling more sophisticated information integration (like multi-hop reasoning or global summarization), they still face challenges when viewed through the lens of episodic memory [26]. Graph-based methods like GraphRAG[11], LightRAG[19], HippoRAG[21, 20] and RAPTOR[59] suffer from two broad limitations. First, they lack mechanisms to track entity state/role changes across time—they represent entities as static nodes without modeling how attributes or relationships evolve throughout a narrative. Second, they provide no specific framework to ground the evolving narrative in space and time, making it difficult to represent sequential developments or causal relationships. These methods typically represent semantic relationships or summarize community structures within a potentially static corpus, but they are not explicitly designed to model the temporal flow of events within a single narrative or to meticulously track the dynamic changes in entity states and roles as the narrative unfolds sequentially. Their structure captures connections, but not necessarily the chronological progression and state transitions required for recalling specific episodes.

Other research efforts have targeted episodic memory more directly. For instance, Larimar [8] proposes modifications to the LLM's attention mechanism, while EM-LLM [15] introduces specific memory components integrated with openweight models. While promising, these approaches often require fundamental changes to the LLM architecture or are designed specifically for openweight models, limiting their applicability. In contrast, our GSW framework is proposed as a plug-and-play episodic memory module compatible with any underlying LLM (including closed-source models like GPT-4o via API) and, critically, requires no specialized training or fine-tuning of model parameters, relying instead on the LLM's capabilities for its operator and reconciliation functions.

G Computation Cost and Resources to build the GSW

The primary computational costs for the Generative Semantic Workspace (GSW) framework are associated with its initial, one-time indexing process. To index the 200 chapters of the Epbench dataset, the total expense is approximately $15 when utilizing GPT-4o. This cost covers all stages of GSW construction, including the generation of operator representations, reconciliation, and the creation of entity-specific summaries. By leveraging parallel calls to the OpenAI API, managed via the Bespoke Curator library [44], this entire indexing task for 200 chapters can be completed in roughly 1 hour. Alternatively, the OpenAI Batch API can be used to reduce costs, with indexing taking hours.

Our primary experiments leverage API-based models (e.g., GPT-4o) and therefore do not necessitate dedicated local computing infrastructure. However, for tasks such as running the baseline method evaluations reported in this study, and for broader experimentation involving various dense retriever models or locally-hosted chat models, we utilized a single server node equipped with four A6000 GPUs.

H Related Computational Models of Semantics

Semantic representation frameworks have a rich history in NLP, yet as we explore below, their design choices create inherent limitations for tracking evolving actor states and relationships—a critical requirement for episodic memory. Among the most influential frameworks are PropBank [30] and FrameNet [3], which attempt to define correspondences between (a) the syntactic "realizations" of semantics *explicit* within language structure, and (b) finite, discrete sets of semantic "roles" [35]. These approaches rely heavily on manually-annotated lexicon ontologies developed by expert linguists. While valuable for understanding individual sentences, they were not designed for the dynamic, interconnected tracking that episodic memory demands. Below, we detail these frameworks and their limitations for serving as memory systems:

PropBank: PropBank utilized a *bottom-up* approach: (1) Dependency Parse Trees [50] were applied to a large text corpus to distill shared syntactic patterns ("Framesets") specific to each verb (a process known as "lexical sampling"). (2) For each Frameset, the corresponding sentences were manually annotated with an enumerated set of *arguments* `ARG:0,...,ARG:N`. These arguments were later associated to verb-specific definitions using VerbNet [61]. The semantic roles are identified as corresponding *spans* within the sentence (commonly a `NP`, `NNP` subtree in the dependency parsing). For example, the sentence (A):

Officers captured **Sarah** at the
Sepulveda on-ramp of the **405**.

would be annotated with the arguments:

Agent: **officers**, *Predicate*: **captured**, *Patient*: **Sarah**.

Perhaps the greatest benefit of PropBank was that its syntactic "grounding" made it possible for rule-based and early ML models [28, 63] to *learn the task of distilling the semantics* given a sentence, albeit within the confines of a *limited* ontology of > 3000 verbs and > 4000 Framesets

Event Databases: PropBank evolved in several directions, including efforts to unify it with related semantic lexicon such as VerbNet and FrameNet [52, 62], or augment it via the DWD overlay [64] to WikiData [75]. The latter of these efforts now manifests as "Event" databases [39, 79] such as the ACE [10] and ERE [1] datasets, and led to the DARPA initiative of Event identification/extraction challenges. Events are best motivated by their related identification tasks: Given a sentence, identify the event(s) – from a set of *hundreds* of events in a pre-annotated schema [82, 76, 43] – that the sentence is referring to. For example, (A) would be annotated with the *Capture* event.

FrameNet: In contrast to PropBank and related Event ontologies, FrameNet[4] utilizes a *top-down* approach that is not tethered to the syntax structure. Rather, expert linguists aggregated roles (redefined as "Frame Elements" (FE)) from a large corpus of sentences, which are *known to co-exist* under a conceptual gestalt, or "Frame". Each frame additionally comprises a set of "Lexical Units" (LU) - valences (mostly verbs and nouns) whose occurrence in a sentence increases the likelihood of a frame. For example,

the *Frame*: **Taking Captive**

would contain the following frame elements and lexical units:

FE: *Agent*, *Captive*, *Cause*
LU: *capture.v*, *secure.v*

FrameNet (1000s of frames and $10,000$s of FE) is a substantially larger and more comprehensive ontology [2] compared to Propbank. When originally constructed, automated systems could not effectively identify the frames implied by a sentence; today, however, Transformer models [73, 9, 5] have demonstrated success at accurately modeling the sentence-to-frame mapping.

Despite the enormous success and wide adoption of PropBank, FrameNet, and their descendant works, the explicit, finite, and discrete lexicons they employ raise the question: *When is an explicit lexicon ontology complete?*. While FrameNet provides Frame-Frame precedence and subset relationships, these are coarse-grained and do not adequately answer the question: *How can we track the evolution of semantics across a stream of sentences?* - a key requirement for any semantic model to serve as a memory.

More recent work [65] has attempted to assemble a comparatively larger (and less stringent) open-schema semantic ontology of concepts using game-play based crowd-sourcing techniques [74]. However, such efforts to scale manual annotation ultimately do not address how a complete ontology can be constructed. Event Graph Models (EGM) [46] generate event networks to describe the dynamics of events in a text corpus, often using a combination of submodules such as Coreference Resolution [49], Named Entity Recognition (NER) [37] and Semantic Role Labeling [53]. Extensions [38] generate the *most likely* event *template* sequences. These methods rely on predefined event schema to enumerate the set of possible events. However, while EGMs both track the evolution of

[4] `https://framenet.icsi.berkeley.edu/`

semantics across sentences and offer an unsupervised approach to extending existing ontologies, these often marginalize across individual contexts in the training corpus, and generate the most likely event *schema* that follows a current event schema network. As a result, these works have yet to design methods to track the semantics across a specific document. The GSW, particularly through the operator, is designed to overcome these challenges by generating actor-centric, evolving semantic maps that are not constrained by predefined, static lexicons and can capture the nuances of unfolding situations.To demonstrate the GSW Operator's effectiveness in producing these comprehensive semantic maps from complex, real-world text, we conducted a qualitative human evaluation.

H.0.1 Comparing Existing Semantic Models to the Operator

To empirically validate the Operator's capability in generating these comprehensive semantic maps, particularly its proficiency in interpreting narrative-rich texts where actor, roles and states undergo significant evolution, we leveraged news reports, as they are a popular resource for sampling semantics-rich stories that belong to universally recognized situation patterns. We query GDELT [33], a Jigsaw-powered news-indexing platform, with Situation identifiers to retrieve a small set of situation-conditioned `en_US` articles. Table 14 presents statistics about the data. These situations were manually selected as an initial seed set – similar to FrameNet's early versions containing few frames [42] – to assess the validity of the GSW framework. Situation-specific seeds are assembled using a bootstrapped method that invokes FrameNet: (a) Frames are linked using subframe and precedence relationships to create weakly connected components; (b) Headers/labels of the frames in each component form the seed search phrases. We evaluate our framework on situations like "Crime and Justice", "Fire Fighting", "Healthcare", and "Technology Development".

Table 14: **Data Statistics:** Situation-specific news reports are sampled from GDELT. Each document (or article) is split into short contexts $C_1, \ldots, C_N$ (of 3 sentences) before being passed into the Operator to generate the sematic representation.

Situation Label	Documents	Sentences	Tokens
crime and justice	80	1209	100,635
fire fighting	79	1116	87,901
technology development	81	1334	122,493
healthcare	81	1259	117,962
economy	78	1264	110,605

Table 15 presents these results across five diverse situations, showing strong human preference for the Operator generated representations compared to existing semantic frameworks.

Table 15: **Operator Evaluations**: Comparison with Existing Frameworks: Given a short context, English-speaking annotators are shown the unlabeled outputs of the Operator and a baseline framework (GLEN, BERT-SRL, FST) and asked to select the one which best summarizes the semantics in the text. The Operator is preferred over baselines across situations.

Situation	Ours vs. Baseline		
	vs. Zhan et al. (GLEN)	vs. Shi & Lin (BERT-SRL)	vs. Chanin (FST)
Crime & Justice	0.90 (0.10)	0.96 (0.04)	0.70 (0.30)
Economy	0.98 (0.02)	0.96 (0.04)	0.86 (0.14)
Firefighting	0.98 (0.02)	0.98 (0.02)	0.79 (0.21)
Healthcare	1.00 (0.00)	0.96 (0.04)	0.94 (0.06)
Tech. Development	0.96 (0.04)	0.96 (0.04)	0.86 (0.14)

H.1 Annotator Guidelines

Annotators who exceeded $\$50K$ in total gross pay were recruited from UpWork, a talent resource. These candidates were first interviewed in a 10-minute session to verify that they were proficient in English and those that had prior experience in annotating large-scale AI/ML data – listed as a verified

skill on the platform – were selected to move on to the next round. Following this, they were given a set of 10 task prototype examples and 10 unanswered labeling tasks. Those that got 9 out of the 10 annotations right moved on to the first round of labeling. Each task was labeled twice – by the *annotator* and a *verifier* – to ensure quality of the results. Annotators were paid $\$5/40$ samples which was estimated to take them about 30 minutes at most, or at the rate of $\$10/$hour, which was confirmed to exceed the federal minimum wage where the annotators were situated. Annotator guidelines are presented in Fig. 12

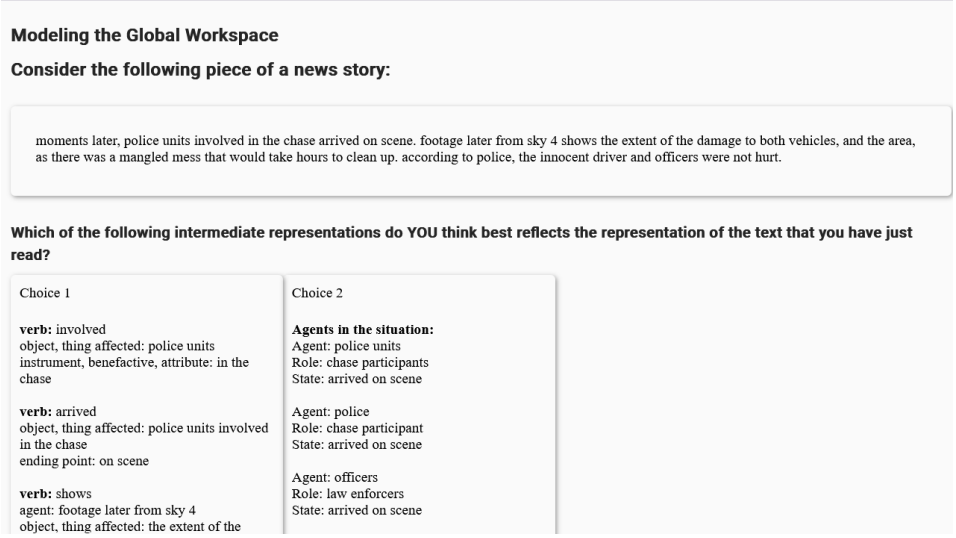


Figure 12: **Annotator instructions for UpWork Task:** Annotators are asked to compare the outputs of the Operator to the Semantic map output by a baseline framework (either GLEN, BertSRL, FST) given a shared input text context. During annotation, one random baseline map and the Operator output are presented in random order and the annotator is asked to pick the representation of the Semantics that best reflects the information in the context.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the main claims in our paper are supported by detailed experiments in the main paper and the supplementary material.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of our work in Section 6

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We provide a probabilistic interpretation of our framework in Section 2 but do not present any theoretical results that require a proof.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe Implementation Details and Evaluation methods in Section 3 and provide further information about prompts used and hyperparameter in Appendix A and E.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the code and data is included with the submission in the supplementary material and will be open sourced upon acceptance.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper does not train any models, all hyperparamters required to reproduce the results are detailed in Appendix E.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report any statistical bars since the only sources of stochasticity in our pipeline are those internal to the OpenAI API.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We present the cost and computation requirements in Appendix H

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper does not violate any of the guidelines stated in the Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We do not believe this paper will have an immediate positive or negative impact to society.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any models, we only make use of open sourced and peer reviewed datasets.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit all the owners of assests we have used in our work.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our main assets are our codebase which has been properly documented and submitted with the supplementary material.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We run one experiment detailed in Appendix I that involves crowdsourcing on Upwork, all relevant information is provided in Appendix I.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

Justification:We include one human-annotation experiment presented in Appendix I. This involved professional annotators evaluating system outputs without collection of personal data or exposure to sensitive content. Given the minimal risk nature of the task and in accordance with our institution's practices for this type of professional evaluation, formal IRB approval was not sought for this study.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [Yes]

    Justification: The core methodology of our work is dependent on calls to an LLM, the paper details all our usage of an LLM.