TOPO-NARRATOR: Heterogeneous Topological Memory for Long-form Narrative Generation

Kai Chen[1,2,3], **Haoran Zhang**[1,2,3], **Tian Lan**[1,2,3],
Zehua Duo[1,2,3], **Jiang Li**[1,2,3], **Guanglai Gao**[1,2,3], **Xiangdong Su**[1,2,3]*
[1] College of Computer Science, Inner Mongolia University, China
[2] National & Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, China
[3] Inner Mongolia Key Laboratory of Multilingual Artificial Intelligence Technology, China
csck@mail.imu.edu.cn, cssxd@imu.edu.cn

Abstract

Long-form narrative generation requires maintaining multi-dimensional consistency over extensive contexts, covering long-term constraints in character settings, event causality, spatial rules, and worldviews. Existing methods compress these heterogeneous elements into homogeneous representations, creating a topological mismatch that lowers retrieval signal-to-noise ratios and induces long-range logical collapse. We propose TOPO-NARRATOR, a framework built on two complementary decouplings. Heterogeneous Topological Memory (HTM) organizes narrative state into a Character Graph, a Plot Chain, and an Environment Tree, providing topology-specific inductive biases. Plan-Reason-Write (PRW) multi-agent collaboration separates high-level reasoning, memory retrieval, and stylistic realization across dedicated agents. Experiments on MoPS and WritingPrompts show that TOPO-NARRATOR consistently generates coherent narratives averaging over 13,000 words and substantially outperforms strong baselines on HANNA quality and Fine-grained consistency metrics, validated by three frontier LLM judges and human evaluation. The source code is available[1].

1 Introduction

Large Language Models (LLMs) have substantially improved local coherence and fluency in narrative generation (Teleki et al., 2025). However, coherent long-form generation remains challenging because long-form stories must preserve character states, plot causality, and world rules across distant chapters.

Existing LLM-based approaches generally address this problem in two ways. Some methods extend access to historical context through long-context mechanisms (Beltagy et al., 2020; Bai et al., 2025), while others inject relevant history through retrieval or external memory modules (Wang et al., 2025).

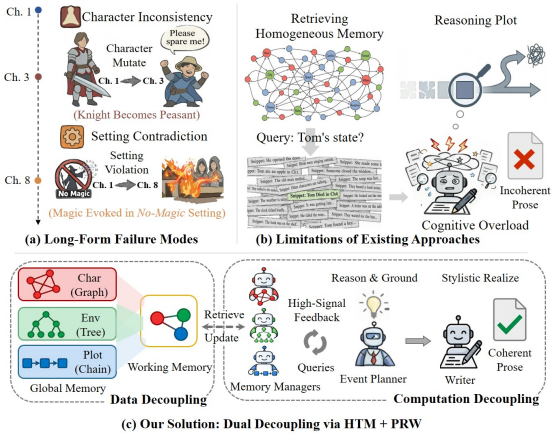


Figure 1: Motivation and overview of TOPO-NARRATOR. (a) Long-form failure modes such as character inconsistency and setting contradictions emerging across distant chapters. (b) Retrieving from homogeneous memory floods the writer's context with noisy snippets, forcing it to simultaneously filter, reason, and write, leading to cognitive overload. (c) Our dual-decoupling solution combining Heterogeneous Topological Memory (HTM) and Plan-Reason-Write (PRW) agent collaboration.

Yet both strategies plateau as narrative length grows. Simply expanding the context window does not resolve long-range consistency, since model performance degrades sharply when critical information lies in the middle of long contexts, which exemplifies the well-documented *lost-in-the-middle* phenomenon (Liu et al., 2024). This limitation makes effective context allocation, rather than raw context size, the binding constraint, shifting the burden to retrieval-based methods to deliver high signal-to-noise content within a bounded context budget. However, classic RAG and external memory architectures typically rely on retrieval and semantic concatenation of flat text segments,

*Corresponding author.
[1] Topo-Narrator

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 923–944
November 6-10, 2026 ©2026 Association for Computational Linguistics

allowing transient plot fragments to overshadow sparse but critical character settings and thus collapsing the signal-to-noise ratio that retrieval was meant to improve. Both strategies therefore plateau well below the novel-length regime.

We argue the core bottleneck lies in a *topological mismatch* between the heterogeneous structural nature of narrative information and the homogeneous modeling of existing systems. Distinct narrative elements naturally exhibit different topological structures, where character relationship networks exhibit dense graph structures (Gurung and Lapata, 2024), plot progression follows chronological or causal chains (Zhang et al., 2021), and world settings can often be organized hierarchically (Pan et al., 2026). Mainstream methods compress all of these into homogeneous vector spaces or flat text, entangling structurally distinct information within a single representation. As a result, retrieving a specific character requires disentangling it from interleaved plot events, and locating a particular world rule requires inferring it from scattered historical descriptions, turning what should be targeted lookup into costly reconstruction. Memory-enhanced systems such as DOME use temporal knowledge-graph quadruples to store and retrieve generated content (Wang et al., 2025). However, they do not explicitly separate narrative memory into topology-specific structures for character evolution, plot causality, and world-rule inheritance. Moreover, existing frameworks couple information retrieval, high-order plot reasoning, and stylized writing within a single generation process, forcing the model to simultaneously filter relevant information from noisy retrieval, perform high-level narrative reasoning, and produce stylistically engaging prose, exacerbating cognitive load.

To address these challenges, we propose TOPO-NARRATOR, a framework built on two complementary decouplings. **Heterogeneous Topological Memory** decouples narrative state into a Character Graph for evolving social relations, a Plot Chain with causal and foreshadowing-resolution edges, and a dual-branch Environment Tree separating physical space from world rules. **Plan-Reason-Write Agent Collaboration** decouples generation across a Global Planner, an Event Planner that interacts with three Memory Managers, and a Writer Agent. A projection operator π extracts a compact *rolling working memory* for each event, supporting isolated chapter generation with atomic merge back to global memory.

Our main contributions are as follows:

- TOPO-NARRATOR robustly generates coherent narratives exceeding 13,000 words on MoPS and WritingPrompts, consistently outperforming strong baselines under three frontier LLM judges and human evaluation.
- We introduce HTM, decoupling narrative state into a Character Graph, a Plot Chain with three edge types including foreshadowing-resolution skip edges, and a dual-branch Environment Tree, providing tailored inductive biases for long-range consistency.
- We design a PRW agent collaboration with a projection operator that extracts a rolling working memory for event-level isolated generation, paired with atomic merge to global memory.

2 Related Work

2.1 Automated Long Story Generation

Recent long-form story generation methods mainly improve coherence through planning and agent collaboration. Planning-based methods usually decompose story generation into outline construction and prose expansion. DOC improves long-range plot coherence through detailed outline control (Yang et al., 2023). DOME further combines dynamic hierarchical outlining with memory enhancement to adapt planning during generation and reduce contextual conflicts (Wang et al., 2025). However, these methods still rely mainly on outline-like representations, which makes it difficult to explicitly model heterogeneous narrative structures such as character relations, causal dependencies, and world rules.

Agent-based methods instead decompose fiction writing into subtasks handled by specialized agents. Agents' Room shows that such collaboration can improve narrative generation by assigning different writing-related roles to different agents (Huot et al., 2025). However, existing agent frameworks mainly divide the writing workflow, while memory retrieval, state tracking, and high-level plot reasoning are not explicitly separated. This limits their ability to manage long-range character and world-state consistency.

2.2 Long-Horizon Information Access

Another line of work addresses long-form generation by improving access to distant information.

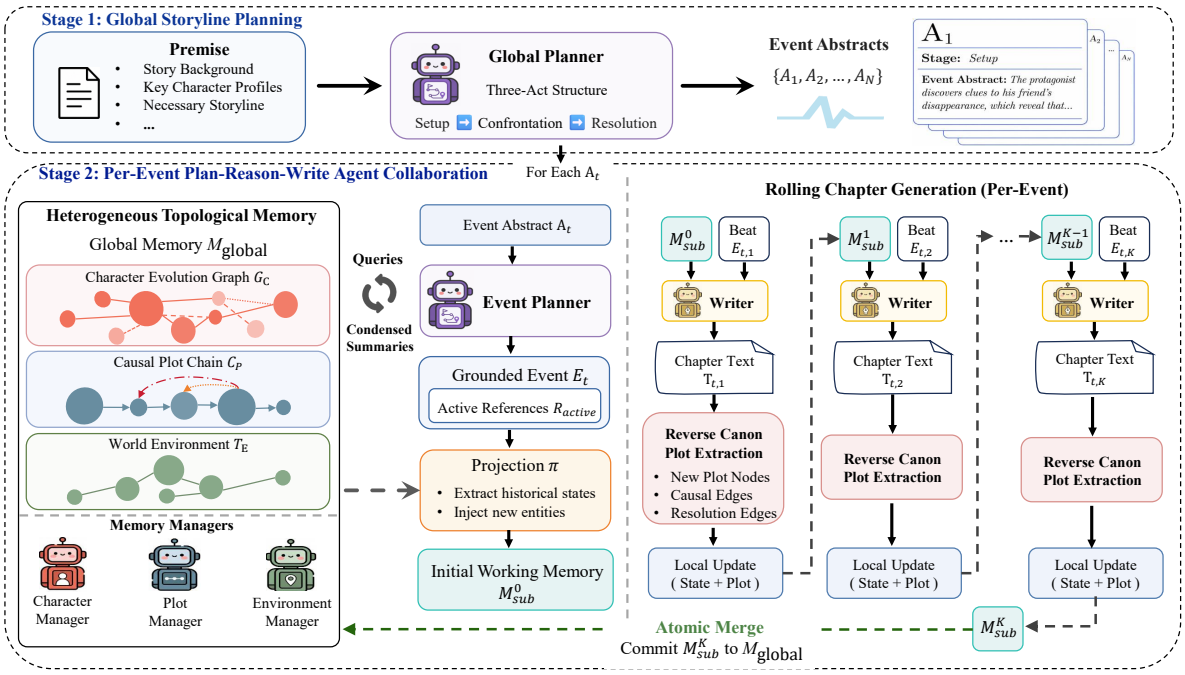


Figure 2: Framework of TOPO-NARRATOR. Stage 1 (top) derives event abstracts from the premise through global story planning. Stage 2 (bottom) processes each abstract A_t as one event and performs event-level Plan-Reason-Write collaboration over Heterogeneous Topological Memory (left). For each event, the Event Planner queries topology-specific Memory Managers to obtain grounded references, and the projection operator π initializes a rolling working memory (middle). The Writer Agent generates chapters sequentially, with local memory updates after each chapter (right). The final event memory is then merged back into the global memory (bottom arrow).

One direction extends the usable context window. Methods such as Longformer and BigBird use sparse attention to process longer sequences more efficiently (Beltagy et al., 2020; Zaheer et al., 2020). However, longer input access does not guarantee reliable use of distant information, since models can still perform poorly when relevant facts appear in the middle of long contexts (Liu et al., 2024).

A second direction introduces external memory or retrieval. General retrieval-augmented methods condition generation on retrieved passages or external memory (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021), while long-form story systems such as DOME incorporate memory enhancement into hierarchical outlining (Wang et al., 2025). However, these methods still tend to organize narrative history as retrieved text or outline-like records. This can mix character states, plot events, and world rules within the same memory pool, weakening retrieval precision and making outdated states harder to filter.

Overall, prior work improves long-form generation through outline control, memory enhancement, or agent collaboration (Yang et al., 2023; Wang et al., 2025; Huot et al., 2025). In contrast, our work focuses on jointly separating heterogeneous narrative memory and decoupling planning, reasoning, and writing.

3 Methodology

3.1 Overview

As discussed above, long-form narrative generation requires structured memory for heterogeneous narrative elements and a generation process that separates planning, retrieval, and writing. TOPO-NARRATOR addresses these requirements with two complementary designs.

Heterogeneous Topological Memory decouples narrative state into three topology-specific structures, a Character Evolution Graph, a Causal Plot Chain, and a World Environment Tree, which together form the global memory $\mathcal{M}_{\text{global}}$. We refer to them as the character graph, plot chain, and environment tree when the context is clear.

Plan-Reason-Write Agent Collaboration decomposes generation into specialized agents that operate on HTM through a projection operator, which extracts an isolated working memory for each event to bound the context size, and then atomically synchronizes the updates back to the global

memory. This design enables coherent 10k+ word generation in our main experiments while keeping each event-level working context bounded.

3.2 Heterogeneous Topological Memory

To address these challenges, we argue that different narrative dimensions require topology-specific inductive biases. Therefore, we maintain three topology-specific structures as the global memory $\mathcal{M}_{\text{global}}$.

$$\mathcal{M}_{\text{global}} = \{\mathcal{G}_C, \mathcal{C}_P, \mathcal{T}_E\}. \tag{1}$$

Character Evolution Graph $\mathcal{G}_C$ To capture evolving social relations, we formalize character memory as a dynamic directed graph $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$. Each node $v \in \mathcal{V}_C$ represents a character entity whose characteristics are categorized into *static attributes* (e.g., name, inherent description, role importance such as protagonist or supporting, and long-term aspiration g_{long}) as well as *dynamic states* (e.g., current status, immediate goal g_{short}). Each edge $e_{ij} \in \mathcal{E}_C$ denotes an inter-character relationship. To mitigate relationship forgetting, we maintain a change history list on each edge:

$$\mathcal{H}_{ij} = \{(r_0, c_0), \ldots, (r_t, c_t)\}, \tag{2}$$

where r_t is the relationship type at time t and c_t records the causal event that triggers the change from r_{t-1} to r_t. This history-preserving design allows the Writer to reference past relationship turning points when generating later chapters, avoiding abrupt relationship resets common in flat-memory baselines.

Causal Plot Chain $\mathcal{C}_P$ To balance temporal linearity with long-range dependencies, we model plot progression as an enhanced causal chain $\mathcal{C}_P = (\mathcal{V}_P, \mathcal{E}_P)$. Each node $v_p \in \mathcal{V}_P$ is a *scene-anchored* plot fragment that corresponds to a single physical scene for retrieval granularity, while edges may connect across scenes. Each node carries a boolean foreshadowing marker $f_{\text{shadow}} \in \{0, 1\}$, where $f_{\text{shadow}}(v) = 1$ indicates that v contains a setup awaiting payoff. Accordingly, $\mathcal{E}_P$ comprises three edge types. Temporal edges e_{temp} connect adjacent steps (v_t, v_{t+1}) to form the chronological backbone. Causal edges e_{cas} further encode causal dependencies between narrative units. Resolution edges e_{res} create backward skip connections $(v_j \xrightarrow{resolve} v_i)$ from a payoff node v_j to the setup node v_i that it resolves. Unresolved foreshadowing is then identified as $\{v \in \mathcal{V}_P \mid f_{\text{shadow}}(v) = 1 \wedge v \text{ has no incoming } e_{\text{res}}\}$, which provides a computable signal for the foreshadowing resolution rate.

World Environment Tree $\mathcal{T}_E$ We model the environment as a rooted dual-branch tree $\mathcal{T}_E = (\mathcal{V}_E, \mathcal{E}_E)$, whose root branches into a *physical space tree* $\mathcal{T}_{\text{phys}}$ and a *world concept tree* $\mathcal{T}_{\text{conc}}$. The physical space tree follows strict containment relations (e.g., Country $\rightarrow$ City $\rightarrow$ Building $\rightarrow$ Room), supporting movement legitimacy checks and reducing *teleportation* hallucinations. Nodes store spatial descriptions and optional coordinates, while edges denote containment. The world concept tree organizes world rules as a knowledge taxonomy (e.g., MagicSystem $\rightarrow$ SpellSlots $\rightarrow$ Level9Spell). Child nodes inherit attributes from parents, enabling ancestor-chain retrieval of higher-level rules to maintain world-rule consistency across long contexts.

3.3 Plan-Reason-Write Agent Collaboration

To address the cognitive overload caused by coupling plot planning, memory retrieval, and stylized writing within a single generation process, we organize specialized agents into a *Plan-Reason-Write* (PRW) collaboration. The framework employs a Global Planner for top-level narrative anchoring, an Event Planner together with three Memory Managers for grounded event refinement, and a Writer Agent for chapter realization. Each agent operates on a clearly scoped portion of $\mathcal{M}_{\text{global}}$, and all state mutations are mediated through a working memory sub-structure $\mathcal{M}_{\text{sub}}$ to ensure event-level isolation.

Global Planning The Global Planner synthesizes the premise into N chronologically arranged event abstracts $\{A_1, \ldots, A_N\}$ that collectively realize the classical Three-Act Structure of setup, confrontation, and resolution. Here N is a tunable parameter independent of the three acts, with early abstracts instantiating the setup, intermediate abstracts developing the confrontation, and later abstracts concluding the resolution. We set $N = 3$ in our main experiments to target the 10k-word regime while keeping the generation budget tractable, although the architecture can support larger N without changing the overall generation pipeline. Each abstract A_t specifies high-level conflict and participants, leaving scene-level and action-level details to the subsequent reasoning stage. We define an *event* as a complete narrative

unit that is coherent in time, space, and causality, realizable as one or more chapters.

Event-Level Reasoning For each abstract A_t, an Event Planner interacts with three specialized Memory Managers to refine A_t into a logically grounded event E_t. The Character Manager, Plot Manager, and Environment Manager are responsible for $\mathcal{G}_C$, $\mathcal{C}_P$, and $\mathcal{T}_E$ respectively, each capable of retrieval-style queries and post-generation updates on their managed structure. Instead of directly accessing $\mathcal{M}_{\text{global}}$, the Event Planner iteratively poses decision-oriented queries to the Managers and integrates the condensed summaries received from them to drive narrative reasoning. For instance, when designing a confrontation scene, the Planner may query the Character Manager for existing hostile characters and the Environment Manager for plausible locations, then ground these into E_t as concrete node references. This decoupling improves the signal-to-noise ratio of retrieved context and avoids overloading any single agent with full historical memory. A concrete walkthrough of event-level planning and memory grounding is provided in Appendix E.

Working Memory Initialization via Precise Subgraph Extraction Once E_t is established, we initialize a working memory sub-structure $\mathcal{M}_{\text{sub}}$ that scopes all subsequent chapter-level generation. Intuitively, $\mathcal{M}_{\text{sub}}$ contains only the characters involved in the event, the earlier plot threads it builds on or resolves, and the relevant locations together with the world rules they inherit. We first identify the active reference set

$$\mathcal{R}_{\text{active}} = \{R_{\text{char}}, R_{\text{plot}}, R_{\text{env}}\}, \tag{3}$$

where each reference set consists of *existing identifiers* and *newly proposed identifiers*. The former point to nodes already present in $\mathcal{M}_{\text{global}}$, including recurring characters, previously established locations, and prior plot nodes that the new event will causally depend on or resolve. The latter, in contrast, are introduced by the Event Planner to instantiate new characters or new scenes demanded by E_t. New plot content is intentionally not preproposed at this stage and will instead be derived from chapter prose, as detailed below. We then define a projection operator π that combines existing-entity extraction with new-entity injection

$$\mathcal{M}_{\text{sub}} = \pi(\mathcal{M}_{\text{global}}, \mathcal{R}_{\text{active}}) = \{\mathcal{G}^C_{\text{sub}}, \mathcal{C}^P_{\text{sub}}, \mathcal{T}^E_{\text{sub}}\}. \tag{4}$$

The extracted historical nodes carry their last-committed states from $\mathcal{M}_{\text{global}}$, while newly proposed character and environment nodes carry their initial Planner-defined attributes. The specific extraction follows the topology of each structure. For the character graph, π extracts the induced subgraph over historical character references and injects the newly proposed character nodes

$$\mathcal{V}^{\text{rel}}_C = \{v \in \mathcal{V}_C \mid v \in R^{\text{exist}}_{\text{char}}\} \cup R^{\text{new}}_{\text{char}}, \tag{5}$$

together with the corresponding induced edges and any newly proposed relationships among them. For the plot chain, π retrieves the historical plot nodes referenced by R_{plot} along with whatever edges already exist among them, providing a chronological anchor against which the event's new plot content will later be linked. Notably, the referenced historical nodes need not be contiguous along the temporal backbone. For the environment tree, π extracts the union of ancestor paths from each referenced node to the root, ensuring local scene details inherit higher-level world rules:

$$\mathcal{V}^{\text{rel}}_E = \bigcup_{v \in R_{\text{env}}} \text{Path}(v, v_{\text{root}}). \tag{6}$$

This hybrid initialization, combining the last-committed historical state with new entities, serves as the canonical starting point for chapter generation within E_t.

Rolling Chapter Generation with Hot Update Each event E_t is decomposed into K chapter-level beats $\{E_{t,1}, \ldots, E_{t,K}\}$, yielding chapter texts $\{T_{t,1}, \ldots, T_{t,K}\}$. Importantly, $\mathcal{M}_{\text{sub}}$ does not remain static across chapters. Instead, it evolves as a *rolling working memory* that absorbs the outcomes of each chapter before the next is written. At step k, the Writer Agent generates the chapter conditioned on the current beat and the up-to-date sub-structure:

$$T_{t,k} = \text{WriterAgent}(E_{t,k}, \mathcal{M}^k_{\text{sub}}, \mathcal{S}_t), \tag{7}$$

where $\mathcal{S}_t$ is a prompt-conditioned stylistic profile (e.g., genre, tone, narrative perspective) inferred by the Writer Agent from the premise and prior context, enabling adaptive prose realization across narrative genres. After generation, Memory Managers parse the chapter text and apply two categories of local update to $\mathcal{M}^k_{\text{sub}}$.

First, *state refinement* updates the dynamic states of characters and environments touched by the

chapter, such as updated emotional states, new injuries, or new objects appearing in a scene.

Second, plot canonicalization extracts new plot nodes from the generated prose via what we call *reverse canon extraction*. Rather than committing to plot details upfront, we let the Writer realize the scene freely and then derive the canonical plot record from the actual text. Each newly extracted plot node p_{new} is appended to the temporal backbone via an e_{temp} edge from the chapter's preceding plot node. If the chapter additionally establishes a causal dependency on or resolves a historical plot node p_{old} referenced in $\mathcal{C}_{\text{sub}}^{P,k}$, the Plot Manager adds the corresponding e_{cas} or e_{res} edge, weaving new content into the long-range causal fabric. If a new plot node is identified as containing a setup awaiting future payoff, its f_{shadow} attribute is marked accordingly, making it discoverable for resolution in subsequent events.

Together these updates yield the next-step working memory

$$\mathcal{M}_{\text{sub}}^{k+1} = \mathcal{M}_{\text{sub}}^{k} \oplus \Delta_k, \tag{8}$$

where Δ_k aggregates all state and plot updates derived from $T_{t,k}$, and $\oplus$ denotes their structure-aware integration into $\mathcal{M}_{\text{sub}}$. Crucially, this entire update happens locally and does not touch $\mathcal{M}_{\text{global}}$, ensuring event-level isolation. Because each event's working memory operates on a bounded subgraph rather than the full history, the per-event computational cost remains stable as the story grows, enabling the framework to scale to novel-length narratives without architectural redesign. The resulting $\mathcal{M}_{\text{sub}}^{k+1}$ then conditions the generation of $T_{t,k+1}$, so each chapter is written against a working memory that already reflects everything that has occurred in earlier chapters of the same event.

Atomic Merge to Global Memory After all K chapters of E_t are completed, the final working memory is merged back into global memory atomically

$$\mathcal{M}_{\text{global}}^{t+1} = \text{Merge}(\mathcal{M}_{\text{global}}^{t}, \mathcal{M}_{\text{sub}}^{\text{final}}). \tag{9}$$

The merge rules are structure-specific. For the character graph, node and edge states are overwritten using $\mathcal{G}_{\text{sub}}^{C,\text{final}}$, while relation changes are *appended* to $\mathcal{H}_{ij}$ to preserve history. Additionally, newly created nodes and edges are added, and any temporary identifiers introduced during reasoning are promoted to global ids. For the plot chain, newly generated nodes and edges are incrementally inserted, and resolved foreshadowing nodes are marked accordingly. For the environment tree, attributes are snapshot-overwritten to reflect mutations such as items taken or doors locked. By committing only at event boundaries, the system treats each event as a self-contained transaction. Consequently, intra-event updates never pollute $\mathcal{M}_{\text{global}}$, and a failed event can be rolled back without affecting prior committed state.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate on MoPS (Ma et al., 2024) and WritingPrompts (Fan et al., 2018). MoPS provides modular premises with background settings, character archetypes, and plot trajectories, while WritingPrompts offers diverse open-ended prompts across genres. We randomly sample 50 test premises from each dataset, yielding 100 generation seeds in total.

Baselines We compare TOPO-NARRATOR with four baselines. **GPT-4o-mini** directly generates from the premise. **DOC** uses detailed outline control for long-story coherence (Yang et al., 2023). **Agents' Room** uses multi-agent collaboration for narrative generation (Huot et al., 2025). **DOME** combines dynamic hierarchical outlining with memory enhancement (Wang et al., 2025). We use official implementations when available.

Implementation Details All methods use `gpt-4o-mini` as the backbone LLM, thereby controlling for the underlying generator and allowing differences across systems to reflect more directly their planning, collaboration, and memory mechanisms. Using the same cost-efficient backbone also makes the full evaluation computationally tractable. Our experiments include 100 TOPO-NARRATOR stories, each approximately 13,000 words long, and 400 stories generated by the baselines. Each baseline is evaluated using its original generation protocol, and output length is reported alongside the main results to contextualize differences in generation scale.

Evaluation Metrics We evaluate general narrative quality with HANNA (Chhun et al., 2022), which covers Relevance (RE), Coherence (CH), Empathy (EM), Surprise (SU), Engagement (EG), and Complexity (CX) on a 1–5 scale.

To better assess long-range narrative stability, we further introduce Fine-grained Consistency met-

Model	Length	Judge Model	HANNA							Fine-grained Consistency						
			RE	CH	EM	SU	EG	CX	Avg	CC	EE	EC	SC	TC	WC	Avg
MoPS Dataset																
GPT-4o-mini	1217.06	GPT-5	4.5	4.4	4.0	3.8	4.1	3.9	4.12	7.8	**8.5**	**8.0**	8.0	8.2	**8.7**	8.20
		Claude Sonnet 4.5	**4.7**	4.4	3.7	2.8	3.7	4.1	3.90	8.2	8.7	8.1	**7.0**	7.8	8.9	8.12
		Gemini 2.5 Pro	4.5	4.5	3.5	2.1	3.2	3.8	3.60	4.5	6.5	8.1	3.9	6.8	5.9	5.95
DOC	3467.38	GPT-5	4.6	4.0	4.0	3.8	4.0	4.9	4.22	7.5	8.1	7.7	7.6	8.1	8.6	7.93
		Claude Sonnet 4.5	**4.7**	4.0	3.3	2.3	3.1	3.9	3.55	8.3	8.4	8.1	6.9	7.7	8.8	8.03
		Gemini 2.5 Pro	4.1	4.4	3.7	2.0	3.2	4.1	3.58	4.9	4.7	8.7	3.5	7.7	6.0	5.92
Agents' Room	4688.20	GPT-5	4.8	4.3	4.2	3.8	4.1	4.9	4.35	7.6	8.1	7.8	7.3	8.1	8.6	7.92
		Claude Sonnet 4.5	4.6	4.2	3.4	2.1	3.1	4.0	3.57	8.4	**8.8**	8.1	6.8	7.8	**9.0**	8.15
		Gemini 2.5 Pro	4.3	4.3	3.8	2.0	3.1	4.5	3.67	4.6	5.5	7.9	4.1	7.2	6.5	5.97
DOME	4547.66	GPT-5	4.6	4.3	4.3	4.0	4.2	**5.0**	4.40	7.6	8.1	7.9	7.5	8.2	**8.7**	8.00
		Claude Sonnet 4.5	4.3	4.0	3.8	2.4	3.2	4.1	3.63	8.3	8.4	8.2	6.8	7.8	8.9	8.07
		Gemini 2.5 Pro	4.3	3.9	3.7	2.0	3.3	4.7	3.65	4.6	3.8	7.7	3.5	7.2	4.1	5.15
Topo-Narrator	**13207.84**	GPT-5	**5.0**	**4.9**	**4.8**	**4.4**	**4.9**	**5.0**	**4.83**	**8.0**	8.4	7.9	**8.2**	**8.7**	8.6	**8.30**
		Claude Sonnet 4.5	4.5	**4.5**	**4.3**	**3.6**	**4.3**	**4.7**	**4.32**	**8.6**	**8.8**	**8.4**	**7.0**	**8.3**	8.9	**8.33**
		Gemini 2.5 Pro	**4.6**	**4.8**	**4.6**	**4.7**	**4.7**	**5.0**	**4.73**	**6.9**	**7.4**	**9.2**	**7.0**	**8.9**	**8.2**	**7.93**
WritingPrompts Dataset																
GPT-4o-mini	1226.02	GPT-5	**5.0**	**4.9**	4.2	3.9	4.4	4.9	4.55	7.7	**8.4**	**8.1**	8.0	8.3	8.2	8.12
		Claude Sonnet 4.5	**4.8**	**4.8**	3.6	2.5	3.4	4.0	3.85	7.6	8.0	8.0	6.6	7.4	7.7	7.55
		Gemini 2.5 Pro	**4.9**	4.8	3.8	2.7	3.7	4.2	4.02	4.2	7.2	8.7	5.0	7.4	6.0	6.42
DOC	3416.86	GPT-5	**5.0**	4.5	4.2	3.9	4.3	**5.0**	4.48	7.6	8.0	7.6	7.6	8.1	8.3	7.87
		Claude Sonnet 4.5	4.7	4.1	3.4	2.2	3.2	4.0	3.60	8.1	8.5	8.1	6.8	7.5	9.0	8.00
		Gemini 2.5 Pro	4.5	4.2	3.7	2.5	3.5	4.3	3.78	4.8	4.8	8.5	3.9	7.4	4.9	5.72
Agents' Room	4634.30	GPT-5	4.9	4.5	4.3	3.5	4.2	4.9	4.38	7.7	8.1	7.9	7.4	8.2	8.4	7.95
		Claude Sonnet 4.5	4.0	4.0	3.6	1.9	3.0	4.0	3.42	7.6	8.4	8.0	6.2	7.6	8.5	7.72
		Gemini 2.5 Pro	4.4	4.5	3.9	2.6	3.4	4.5	3.88	4.4	5.7	8.6	4.1	7.2	5.4	5.90
DOME	4347.30	GPT-5	4.9	4.7	4.5	3.9	4.5	**5.0**	4.58	7.5	7.9	7.9	7.5	8.1	8.3	7.87
		Claude Sonnet 4.5	3.5	3.8	3.7	2.1	3.0	4.1	3.37	8.2	8.0	8.2	6.9	7.6	8.9	7.97
		Gemini 2.5 Pro	4.2	4.3	3.8	2.9	3.4	4.8	3.90	4.7	4.0	7.9	3.5	6.9	4.2	5.20
Topo-Narrator	**13048.50**	GPT-5	**5.0**	**4.9**	**4.8**	**4.4**	**4.9**	**5.0**	**4.83**	**7.9**	8.3	7.8	**8.1**	**8.7**	**8.5**	**8.22**
		Claude Sonnet 4.5	4.6	4.3	**4.1**	**3.1**	**3.9**	**4.9**	**4.15**	**8.6**	**8.8**	**8.5**	**7.2**	**8.5**	**9.1**	**8.45**
		Gemini 2.5 Pro	4.7	**4.9**	**4.6**	**4.8**	**4.8**	**5.0**	**4.80**	**6.5**	**8.0**	**9.2**	**7.1**	**9.0**	**8.5**	**8.05**

Table 1: Main automatic evaluation results on MoPS and WritingPrompts. Length denotes average word count. Bold and underline mark the best and second-best scores within each dataset, judge, and metric column. HANNA is scored on a 1–5 scale, while Fine-grained Consistency is scored on a 0–10 scale.

rics based on prior work on entity coherence, character representation, and story coherence evaluation (Papalampidi et al., 2022; Gurung and Lapata, 2024; Yi et al., 2026). The metrics cover six dimensions, including Character Consistency (CC), Entity-Event Coherence (EE), Emotional Coherence (EC), Scene Cohesion (SC), Thematic Consistency (TC), and World Consistency (WC). Each dimension is scored on a 0–10 scale, and the evaluator is required to extract supporting evidence before producing the final score. Full prompts and evidence extraction details are provided in Appendix F.

Automatic and Human Evaluation For automatic evaluation, we use GPT-5, Gemini 2.5 Pro, and Claude Sonnet 4.5 as judges for both HANNA and Fine-grained Consistency. The generation model is not used as a judge, and two of the judges belong to different model families from the generator, which helps mitigate self-preference bias (Panickssery et al., 2024). All systems are scored using identical prompts, and the judges see only the premise and the anonymized final story. Human evaluation is reported separately in §4.4.

4.2 Main Results

Length and Quality Table 1 shows that TOPO-NARRATOR generates substantially longer stories than all baselines, averaging 13.2k words on MoPS and 13.0k words on WritingPrompts. In contrast, baseline outputs range from about 1.2k to 4.7k words under their native protocols. This indicates a clear long-form capability gap rather than a minor length variation.

The length gain does not come at the cost of quality. Across both datasets and all three judges, TOPO-NARRATOR achieves the highest average HANNA and fine-grained consistency scores. The gains are especially clear on coherence-related dimensions, suggesting that structured memory and

event-level working context help preserve long-range narrative state. To separate these gains from output length, we further compare against a comparable-length control in §4.3.

Metric-Level Results On HANNA, TOPO-NARRATOR obtains the best average scores across both datasets and all three judges. Averaged over the six dataset-judge settings, it reaches 4.61, outperforming the strongest baseline average of 4.01 by 0.60 points. The improvement is consistent on both datasets, with gains of 0.73 on MoPS and 0.45 on WritingPrompts over the strongest baseline. After excluding Engagement and Complexity, the two dimensions most plausibly favored by longer text, TOPO-NARRATOR still leads the strongest baseline on the remaining four dimensions (HANNA-4), with 4.56 vs. 3.91 on MoPS and 4.52 vs. 4.16 on WritingPrompts.

The advantage is larger on Fine-grained Consistency. TOPO-NARRATOR achieves an average FC score of 8.21, compared with 7.39 for the strongest baseline. The gains are 0.76 on MoPS and 0.88 on WritingPrompts, with clear improvements on Character Consistency, Thematic Consistency, and Scene Cohesion. These are the dimensions targeted by the Character Graph, Plot Chain, and Environment Tree in HTM. Shorter baselines occasionally remain competitive on isolated dimensions such as World Consistency, but these results should be interpreted in light of their much shorter generations, where maintaining local consistency may be less challenging.

Uncertainty and Judge Agreement A premise-level paired bootstrap analysis shows a Fine-grained Consistency advantage over the strongest baseline on both MoPS ($+0.769$, 95% CI $[0.656, 0.884]$) and WritingPrompts ($+0.874$, 95% CI $[0.712, 1.045]$). Story-level Spearman correlations among the three judges for Fine-grained Consistency range from 0.299 to 0.467 (all $p < 10^{-5}$). Appendix B provides details of both analyses.

4.3 Comparable-Length Control

Extending the baselines to about 13k words would require changing their native generation protocols, so we instead test whether the gains persist at comparable length with a rolling-summary (RS) control on 10 randomly sampled MoPS premises. The control first writes a complete chapter outline and then generates each chapter from a recursively updated global summary and the preceding chapter, which provides persistent context without HTM, Memory Managers, or projection. As shown in Table 2, TOPO-NARRATOR outperforms the control on all 10 premises for HANNA, HANNA-4, and Fine-grained Consistency, and its HANNA and Fine-grained Consistency gains are positive under each of the three judges.

Metric	RS	Topo	Δ [95% CI]
Words	11.8k	13.2k	–
HANNA	3.70	4.56	+0.86 [0.74, 0.97]
HANNA-4	3.65	4.48	+0.83 [0.71, 0.94]
FC	5.76	8.14	+2.38 [2.01, 2.74]

Table 2: Comparable-length control on 10 MoPS premises, averaged over three judges. RS and Topo denote the rolling-summary control and TOPO-NARRATOR. HANNA-4 excludes Engagement and Complexity. Δ is the mean paired difference with a premise-level bootstrap 95% CI.

Dataset	Dim.	Win%	Tie%	Lose%	Mean
MoPS	CH	70.0	17.5	12.5	+0.97
	EG	72.5	10.0	17.5	+0.93
	CC	65.0	20.0	15.0	+0.68
	WC	60.0	27.5	12.5	+0.60
	Ovr	75.0	2.5	22.5	+0.93
WP	CH	85.0	10.0	5.0	+1.30
	EG	80.0	10.0	10.0	+0.97
	CC	77.5	15.0	7.5	+0.90
	WC	67.5	25.0	7.5	+0.88
	Ovr	85.0	5.0	10.0	+1.12
All	CH	**77.5**	13.8	8.8	**+1.14**
	EG	**76.3**	10.0	13.8	**+0.95**
	CC	**71.3**	17.5	11.3	**+0.79**
	WC	**63.8**	26.3	10.0	**+0.74**
	Ovr	**80.0**	3.8	16.3	**+1.02**

Table 3: Pairwise human preference of TOPO-NARRATOR over DOME on Coherence, Engagement, Character Consistency, World Consistency, and Overall Preference. Mean scores range from -2 to $+2$, with positive values favoring TOPO-NARRATOR.

4.4 Human Evaluation

Setup We conduct blind pairwise human evaluation against DOME, the most closely related memory-enhanced baseline. We sample 20 story pairs, with 10 from MoPS and 10 from WritingPrompts. Four graduate evaluators with prior experience in reading long-form fiction rate each pair on Coherence, Engagement, Character Consistency, World Consistency, and Overall Preference using a five-point scale from -2 to $+2$, where positive values favor TOPO-NARRATOR.

Variant	Words	HANNA Avg	FC Avg
MoPS			
TOPO-NARRATOR	**13.3k**	**4.61**	**8.18**
w/o HTM	10.8k	3.50	4.66
w/o PRW	9.1k	3.80	5.65
WritingPrompts			
TOPO-NARRATOR	**12.9k**	**4.62**	**8.22**
w/o HTM	10.0k	3.50	5.33
w/o PRW	8.8k	3.54	5.52

Table 4: Ablation results on 30 premises per dataset. Scores are averaged across GPT-5, Claude Sonnet 4.5, and Gemini 2.5 Pro. Paired differences with 95% confidence intervals are reported in Appendix C.

Results As shown in Table 3, human evaluators prefer TOPO-NARRATOR over DOME on every dimension and both datasets. Overall win rates range from 63.8% on World Consistency to 80.0% on Overall Preference. Pair-level Wilcoxon signed-rank tests show significant differences on all five dimensions ($p < 0.001$), with matched-pairs rank-biserial effect sizes of 0.986 for Coherence and 1.000 for the other four. Inter-annotator agreement measured by Adjacent Agreement Rate reaches 69.0% overall, with the full definition and per-dimension results reported in Appendix A.

4.5 Ablation Study

We conduct ablations on 30 premises from each of MoPS and WritingPrompts to examine the contribution of HTM and PRW. The **w/o HTM** variant replaces topology-specific memory with flat-text memory, while **w/o PRW** uses a single LLM to handle retrieval, reasoning, and writing over HTM. Scores are averaged across the three LLM judges. Table 4 shows that both components are necessary. Removing either one lowers HANNA and Fine-grained Consistency on both datasets, with all paired 95% confidence intervals excluding zero. Removing HTM causes the largest drop, especially on fine-grained consistency, confirming the importance of topology-specific memory. Removing PRW also reduces performance, showing that structured memory alone is insufficient without separating retrieval, reasoning, and writing. A detailed computational cost analysis, including token consumption, latency, and API call comparisons, is provided in Appendix D.

5 Conclusion

This paper addresses two persistent bottlenecks in long-form narrative generation, namely the homogeneous modeling of heterogeneous narrative information and the coupling of retrieval, reasoning, and writing within a single generation process. We propose TOPO-NARRATOR, a topology-aware multi-agent framework that organizes narrative memory into character, plot, and environment structures, and coordinates generation through Plan-Reason-Write collaboration. By extracting event-level rolling working memory and merging updates back into global memory after each event, TOPO-NARRATOR reduces context congestion while preserving long-range narrative state.

Experiments on MoPS and WritingPrompts show that TOPO-NARRATOR generates coherent stories averaging over 13,000 words and consistently outperforms strong baselines under both LLM-based and human evaluation. The improvements are especially clear on coherence-related and fine-grained consistency dimensions. These results suggest that topology-specific memory and cognitively decoupled agent collaboration provide an effective direction for controllable long-form narrative generation. Although TOPO-NARRATOR uses more tokens than lightweight baselines to maintain multiple memory structures, this trade-off suits quality-sensitive long-form writing such as novels and scripts, where early continuity errors propagate to later chapters.

Limitations

Our work has several limitations. Firstly, TOPO-NARRATOR introduces additional engineering and runtime overhead because it maintains multiple memory structures and coordinates several agents. Although the framework improves long-range consistency, it requires more token usage and latency than single-pass generation. Moreover, memory construction and update depend on LLM-based extraction, which may introduce parsing errors or incomplete state updates when generated chapters contain ambiguous events. Verifying updates against chapter evidence before each global commit could mitigate this risk, and the rollback mechanism warrants direct testing. In addition, our ablations establish the contributions of HTM and PRW as a whole. Future work includes isolating finer-grained components such as the projection operator and each memory topology, validating the framework with stronger generation backbones, and conducting broader human evaluation.

Ethics Statement

This work follows the ACL ethics policy and is intended for academic research on long-form narrative generation. We use public benchmark datasets and do not collect or release personal or sensitive user data. Human evaluation is conducted through blind pairwise comparison of generated stories. Evaluators were informed that their anonymous ratings would be used for aggregate research analysis. No personal, biometric, or sensitive information was collected or retained. Evaluators were compensated at a rate above the local minimum wage.

The proposed framework is designed to improve controllability and consistency in machine-generated narratives. However, as with other LLM-based generation systems, it may still produce biased, stereotypical, or otherwise inappropriate content depending on the input premise and the underlying language model. The generated stories in this work are used only for research evaluation. Practical deployment should include additional safeguards, content filtering, and human oversight, especially in creative writing scenarios involving sensitive themes or vulnerable audiences.

Acknowledgments

This work was funded by National Natural Science Foundation of China (Grant No. 62366036), Outstanding Youth Fund Project of Inner Mongolia Autonomous Region (Grant No. 2025JQ010), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (Grant No. NJYT24033), Major Science and Technology Projects of Inner Mongolia Autonomous Region (Grant No. 2025ZDSF0029), and Key R&D and Achievement Transformation Program of Inner Mongolia Autonomous Region (Grant Nos. 2025YFDZ0011, 2025YFDZ0026, 2025YFSH0021, 2025YFHH0073, and 2026YFDZ0158).

References

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longwriter: Unleashing 10,000+ word generation from long context LLMs. In *The Thirteenth International Conference on Learning Representations*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5794–5836, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

Alexander Gurung and Mirella Lapata. 2024. CHIRON: Rich character representations in long-form narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8523–8547, Miami, Florida, USA. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Fantine Huot, Reinald Kim Amplayo, Jennimaria Palomaki, Alice Shoshana Jakobovits, Elizabeth Clark, and Mirella Lapata. 2025. Agents' room: Narrative generation through multi-step collaboration. In *The Thirteenth International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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. 2020.

Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yan Ma, Yu Qiao, and Pengfei Liu. 2024. MoPS: Modular story premise synthesis for open-ended automatic story generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2135–2169, Bangkok, Thailand. Association for Computational Linguistics.

Zhijun Pan, Antonios Andronis, Eva Hayek, Oscar A. P. Wilkinson, Ilya Lasy, Annette Parry, Guy Gadney, Tim J. Smith, and Mick Grierson. 2026. Guiding generative storytelling with knowledge graphs. *International Journal of Human–Computer Interaction*, 42(16):13447–13469.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, pages 68772–68802. Curran Associates, Inc.

Pinelopi Papalampidi, Kris Cao, and Tomas Kocisky. 2022. Towards coherent and consistent use of entities in narrative generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17278–17294. PMLR.

Maria Teleki, Vedangi Bengali, Xiangjue Dong, Sai Tejas Janjur, Haoran Liu, Tian Liu, Cong Wang, Ting Liu, Yin Zhang, Frank Shipman, and James Caverlee. 2025. A survey on LLMs for story generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13954–13966, Suzhou, China. Association for Computational Linguistics.

Qianyue Wang, Jinwu Hu, Zhengping Li, Yufeng Wang, Daiyuan Li, Yu Hu, and Mingkui Tan. 2025. Generating long-form story using dynamic hierarchical outlining with memory-enhancement. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1352–1391, Albuquerque, New Mexico. Association for Computational Linguistics.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving long story coherence with detailed outline control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3378–3465, Toronto, Canada. Association for Computational Linguistics.

Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Kuan Lu, Shiyao Qian, Xinhang Yuan, Yi Xin, Yijin Wang, Jingqun Tang, Yuchen Li, Hongyang He, Zhen Tian, Tianxiang Xu, Keqin Li, Menghao Huo, Jiaqi Chen, Miao Zhang, Tianyu Shi, and Jianyuan Ni. 2026. SCORE: Story coherence and retrieval enhancement for AI narratives. In *Companion Proceedings of the ACM Web Conference 2026*, pages 823–827, New York, NY, USA. Association for Computing Machinery.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Xiyang Zhang, Muhao Chen, and Jonathan May. 2021. Salience-aware event chain modeling for narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1418–1428, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Human Evaluation Statistics

The Adjacent Agreement Rate (AAR) is formally defined as:

$$\mathrm{AAR} = \frac{1}{|D| \cdot |P| \cdot \binom{n}{2}} \times \sum_{d \in D} \sum_{p \in P} \sum_{\substack{i,j \in E \\ i<j}} \mathbf{1}[|r_{i,p,d} - r_{j,p,d}| \leq 1], \tag{10}$$

where E denotes the four evaluators ($n = |E| = 4$), P the 20 story pairs, D the five dimensions, and $r_{i,p,d} \in \{-2, -1, 0, +1, +2\}$ is evaluator i's rating on pair p along dimension d. The denominator $|D| \cdot |P| \cdot \binom{n}{2} = 600$ counts all unique evaluator-pair comparisons.

Dimension	AAR (%)	Agree / Total
Coherence	65.8	79 / 120
Engagement	64.2	77 / 120
Character Consistency	75.0	90 / 120
World Consistency	72.5	87 / 120
Overall Preference	67.5	81 / 120
Pooled (5 dims)	**69.0**	**414 / 600**

Table 5: Per-dimension Adjacent Agreement Rate (± 1 tolerance), 4 evaluators $\times$ 20 pairs.

Inter-Annotator Agreement We measure inter-annotator agreement using the Adjacent Agreement Rate (AAR), which counts two ratings as agreeing when they differ by at most one Likert point.

Pooled across all 600 pairwise comparisons, the AAR reaches 69.0%, ranging from 64.2% on the most subjective dimension (Engagement) to 75.0% on the most consensus-prone one (Character Consistency). We adopt AAR over Krippendorff's α because the latter is known to suffer the prevalence paradox under skewed marginal distributions (Feinstein and Cicchetti, 1990; Gwet, 2008), which applies to our heavily Topo-favored data.

Effect Sizes We average the four evaluators' ratings within each of the 20 story pairs and treat the pair as the unit of inference. Table 6 reports the matched-pairs rank-biserial effect size r_{rb} and the p-value of the Wilcoxon signed-rank test for each dimension. The corresponding mean differences are given in Table 3. An r_{rb} of 1.000 means that every nonzero pair-level difference favors TOPO-NARRATOR, not that all individual ratings agree. World Consistency reaches 1.000 with two tied pairs, while Coherence has one small negative pair-level difference.

Dimension	r_{rb}	p
Coherence	0.986	5.72×10^{-6}
Engagement	1.000	1.91×10^{-6}
Character Consistency	1.000	1.91×10^{-6}
World Consistency	1.000	1.02×10^{-4}
Overall Preference	1.000	1.91×10^{-6}

Table 6: Matched-pairs rank-biserial effect sizes and Wilcoxon signed-rank p-values for TOPO-NARRATOR versus DOME, computed over 20 pair-level mean ratings.

B Uncertainty and Inter-Judge Agreement

Bootstrap Confidence Intervals For each premise, we average the scores of the three judges and then bootstrap the 50 per-premise differences between TOPO-NARRATOR and the strongest baseline with 100,000 resamples, treating the premise rather than an individual judge score as the unit of inference (Koehn, 2004). The resulting 95% confidence intervals for the Fine-grained Consistency advantage are $[0.656, 0.884]$ on MoPS and $[0.712, 1.045]$ on WritingPrompts, both excluding zero.

Inter-Judge Agreement Table 7 reports story-level Spearman correlations between each pair of judges on Fine-grained Consistency, computed over the same 250 system-story items (5 systems $\times$ 50 premises) in each dataset. All six correlations are positive and significant ($p < 10^{-5}$). Their moderate magnitude reflects differences in calibration and in the assessment of individual stories, yet each judge independently ranks TOPO-NARRATOR first on both HANNA and Fine-grained Consistency in both datasets (Table 1). We therefore treat Fine-grained Consistency as one component of a triangulated evaluation, together with HANNA and blind human evaluation, rather than relying on any single judge.

Judge Pair	MoPS	WritingPrompts
GPT-5 vs. Claude	0.416	0.341
GPT-5 vs. Gemini	0.389	0.467
Claude vs. Gemini	0.359	0.299

Table 7: Story-level Spearman correlations between LLM judges (GPT-5, Claude Sonnet 4.5, Gemini 2.5 Pro) on Fine-grained Consistency, over 250 system-story items per dataset. All $p < 10^{-5}$.

C Ablation Confidence Intervals

Table 8 reports the paired differences between the full model and each ablation, with 95% confidence intervals from the same premise-level bootstrap as in Appendix B, over 30 premises per dataset.

Ablation	Δ HANNA [95% CI]	Δ FC [95% CI]
MoPS		
w/o HTM	+1.111 [0.987, 1.233]	+3.525 [3.319, 3.719]
w/o PRW	+0.806 [0.622, 0.981]	+2.531 [2.337, 2.752]
WritingPrompts		
w/o HTM	+1.128 [0.998, 1.252]	+2.886 [2.698, 3.070]
w/o PRW	+1.087 [0.956, 1.220]	+2.703 [2.492, 2.909]

Table 8: Full model minus each ablation, averaged over three judges, with premise-level paired bootstrap 95% confidence intervals.

D Computational Cost Analysis

Long-form generation methods differ substantially in their resource demands. We report token consumption, latency, and API call counts to characterize the quality-cost trade-off. Table 9 summarizes these measurements, reported per 1,000 generated words and averaged across 5 randomly sampled premises per method. Latency is end-to-end wall-clock time over all stages, with Memory Manager calls issued sequentially rather than in parallel, and may vary with API load.

The methods fall into two regimes. Lightweight baselines consume under 8k tokens per 1k words,

Method	Tokens	Latency (s)	API Calls
GPT-4o-mini	1.4k	11.5	0.8
DOC	7.6k	14.5	4.5
Agents' Room	7.4k	16.1	1.9
DOME	24.9k	230.5	34.6
TOPO-NARRATOR	27.7k	**75.6**	**14.1**

Table 9: Computational cost per 1,000 generated words, averaged over 5 stories per method. Bold marks the most efficient configuration among complex multi-agent frameworks (Topo-Narrator and DOME).

but they plateau in both generation length and narrative quality. Complex frameworks with explicit memory or planning (DOME and TOPO-NARRATOR) incur higher token costs in exchange for substantially stronger long-range coherence.

Within the complex-framework regime, TOPO-NARRATOR achieves a markedly better cost profile than DOME. While token consumption is comparable ($+11.1\%$), TOPO-NARRATOR reduces wall-clock latency by 67.2% ($230.5 \rightarrow 75.6$ seconds) and API call count by 59.2% ($34.6 \rightarrow 14.1$). This efficiency stems from the event-level isolation enabled by precise subgraph extraction. Each chapter is generated against a bounded $\mathcal{M}_{\text{sub}}$ in a single call with structured high-density context, rather than triggering repeated full-context reasoning over an expanding memory as DOME does. The result is that TOPO-NARRATOR delivers $2.9\times$ longer and more coherent stories with $3\times$ lower latency than the strongest baseline.

E Walkthrough of Event-Level Planning and Generation

To illustrate how global planning, multi-turn memory interaction, working-memory initialization, and rolling chapter generation interact in TOPO-NARRATOR, we provide a concrete walkthrough.

Premise and Global Planning Consider the premise: *a young magic apprentice discovers a forbidden book in the academy library, accidentally releases a shadow creature, and is forced into a fantasy adventure*. The Global Planner expands this premise into N event abstracts $\{A_1, \ldots, A_N\}$ following the Three-Act structure. Note that N denotes the total number of event abstracts ($N = 3$ in our experiments, and larger for longer works), while the Three-Act structure shapes the macro-narrative arc spanned by these abstracts. For instance, A_1 may establish the apprentice's arrival at the academy (Act I), while a later abstract A_t may describe the forbidden book incident (Act II).

Multi-Turn Memory Interaction When the system activates A_t, the Event Planner does not query all Memory Managers at once. Instead, it conducts an iterative dialogue, raising new questions based on partial answers, and stops once every issued query has been answered. A typical trajectory looks like the following.

(1) The Event Planner queries the Character Manager for the apprentice's current state. The Manager returns the apprentice's recent goals, known spells, and emotional state at the end of the previous event.

(2) Conditioned on this character context, the Event Planner queries the Plot Manager for relevant foreshadowing. The Manager retrieves an earlier dark-energy foreshadowing node and a forbidden-book reference established in A_1.

(3) Conditioned on both prior responses, the Event Planner queries the Environment Manager for the spatial context. The Manager returns the ancestor path leading to the academy library and the inherited world rules governing sealed artifacts.

Grounded Event and Active References Based on these progressive answers, the Event Planner grounds A_t into a logically consistent event E_t in which the apprentice enters the library, opens the forbidden book, weakens the seal, and releases the shadow creature, all under rules consistent with prior world settings. Managers only read the committed memory, and the Planner, as the sole writer of E_t, resolves conflicting outputs in favor of committed states and world rules, for example keeping the rule against spellcasting on sealed artifacts even though the apprentice knows an unsealing spell. A validator then checks E_t for hard conflicts such as location or causality violations and returns any violation for one repair round. The active reference set $\mathcal{R}_{\text{active}}$ contains the apprentice and shadow creature in $\mathcal{R}_{\text{char}}$, the dark-energy foreshadowing node and the forbidden-book reference in $\mathcal{R}_{\text{plot}}$, and the academy-library path in $\mathcal{R}_{\text{env}}$. The projection operator π extracts these referenced structures from $\mathcal{M}_{\text{global}}$ into the initial working memory $\mathcal{M}^0_{\text{sub}}$.

Rolling Chapter Generation and Reverse Canon Extraction E_t is partitioned into K chapter beats $\{E_{t,1}, \ldots, E_{t,K}\}$. The Writer generates the first chapter text $T_{t,1}$ conditioned on $\mathcal{M}^0_{\text{sub}}$ and $E_{t,1}$. The generated prose may contain new canonical facts not explicitly in the plan, such as the ap-

prentice being injured in the left shoulder while resisting the seal. The Plot Manager performs *reverse canon plot extraction* on $T_{t,1}$ to identify these new plot nodes and inserts them into $\mathcal{M}_{\text{sub}}$, yielding $\mathcal{M}_{\text{sub}}^{1}$. The Character Manager similarly performs local state updates, marking the apprentice's shoulder as injured. Subsequent chapter beats $T_{t,2}, \ldots, T_{t,K}$ proceed against this evolving working memory.

Atomic Merge After all K chapters complete, the final working memory $\mathcal{M}_{\text{sub}}^{K}$ is atomically committed back to $\mathcal{M}_{\text{global}}$. The apprentice's injured shoulder, the released shadow creature, and other emergent plot nodes thereby become part of the global narrative state available to future event-level planning. New entities such as the shadow creature enter the merge with temporary identifiers, which are matched first to existing global identifiers and then to canonical names, and receive new global identifiers only when neither matches.

F Evaluation Prompts and Examples

This appendix provides the evaluator prompts used for automatic assessment. Figures 3–8 show the six Fine-grained Consistency evaluators, and Figure 9 shows the HANNA quality evaluator. For Fine-grained Consistency, each prompt asks the judge to inspect the generated story, identify relevant evidence spans, and then assign a score on the corresponding dimension. This evidence-first design is intended to reduce purely impressionistic scoring and make the judgment more grounded. For HANNA, we follow the six standard quality dimensions used in the main experiments, including Relevance, Coherence, Empathy, Surprise, Engagement, and Complexity.

G Human Evaluation Interface

Figure 10 shows the browser-based blind pairwise evaluation interface used for human assessment. For each pair, evaluators were shown the same premise and two anonymized stories, denoted as Story A and Story B. They rated the pair on Coherence, Engagement, Character Consistency, World Consistency, and Overall Preference using a five-point comparative scale from -2 to $+2$. The interface also provided an optional comment box for recording ambiguity or qualitative observations. The source systems of Story A and Story B were hidden from evaluators throughout the evaluation.

ROLE: You are a ruthless senior fiction editor and character-consistency auditor.

INPUT:

- PREMISE: {premise}
- TEXT: {story_text}
- MAX_CHARACTERS: 5 (max(k)=5)

TASK:

1. Automatically identify up to K main characters using mention frequency (with coreference resolution) and narrative importance.
2. For each character, produce a CHARACTER_PROFILE and evaluate Character Consistency across TEXT relative to PREMISE.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["CHIRON2024","Papalampidi2022"],
 "characters_evaluated": <N>,
 "results": [
   {
     "character_profile": {
       "canonical_name":"<str>",
       "aliases":["<str>", ...],
       "first_appearance":"<loc>",
       "mention_count": <int>,
       "short_bio":"<1-2 sentence inference>",
       "core_goals":"<short>",
       "relations":["<str>", ...]
     },
     "character_consistency": {
       "score": <0-10>,
       "needs_revision": <true/false>,
       "critique":"<sharp paragraph (1) describing contradictions or weak motives>",
       "suggestion":"<concrete fix (1-2 sentences)>",
       "evidence_spans":[
         {"loc":"<ch:line>", "quote":"<<=2 short sentences>", "issue":"<label>"}
       ],
       "low_confidence": <true/false>
     }
   },
   ...
 ],
 "other_characters_summary": {
   "count": <int>,
   "top_issues":["<str>", ...]
 },
 "aggregate": {
   "overall_notes":"<if any>"
 }
}
```

ADDITIONAL RULES:

- Scores are integers 0-10.
- Evidence quotes must be verbatim from TEXT (<=2 short sentences).
- Do not hallucinate facts; if a fact is inferred, mark low_confidence true and explain.

Figure 3: Prompt for the Character Consistency evaluator.

ROLE: You are a narrative continuity auditor specializing in entities and event chains.

INPUT:

- PREMISE: {premise}
- TEXT: {story_text}
- TIME_ORDERING: true

TASK:

1. Extract key entities (people, places, objects) and major events, build a minimal event timeline.
2. Evaluate entity/event coherence: persistence of entities, dropped entities, contradictory events, missing causal links.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["Papalampidi2022","SCORE2025"],
 "premise":"<premise>",
 "entity_summary":[
   {"entity":"<str>","first_appearance":"<loc>","mentions":<int>,"roles":["<str>"]},
   ...
 ],
 "event_timeline":[
   {"time_idx":1,"descr":"<short event description>","loc":"<loc>"},
   ...
 ],
 "entity_event_coherence": {
   "score": <0-10>,
   "needs_revision": <true/false>,
   "critique":"<paragraph: list main incoherences e.g., dropped entities, causal gaps>",
   "suggestion":"<concrete fixes e.g., 'add bridging scene between ch4 and ch5'>",
   "evidence_spans":[
     {"loc":"<ch:line>","quote":"<<=2 sentences>","issue":"<label>"}
   ],
   "metrics": {
     "entity_drop_rate": <float>,
     "contradictions_count": <int>
   },
   "low_confidence": <true/false>
 }
}
```

ADDITIONAL RULES:

- Event descriptions should be short (<=20 words).
- Use coreference resolution before counting mentions.
- If uncertain about temporal order, indicate assumptions in output.

Figure 4: Prompt for the Entity/Event Coherence evaluator.

ROLE: You are an emotional-trajectory analyst.

INPUT:

- PREMISE: {premise}
- TEXT: {story_text}

TASK:

1. Produce a coarse-grained emotional trajectory for main characters (per chapter or per paragraph): label each segment as "positive"/"neutral"/"negative" (+ optional intensity 1-5).
2. Detect implausible abrupt shifts (emotion jumps without causal driver) and evaluate overall Emotional Coherence.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["SCORE2025","Papalampidi2022"],
 "premise":"<premise>",
 "character_trajectories":[
   {
     "character":"<name>",
     "trajectory":[
       {"loc":"ch1","val":"neutral","intensity":2},
       {"loc":"ch2","val":"negative","intensity":4},
       ...
     ],
     "emotional_coherence": {
        "score": <0-10>,
        "needs_revision": <true/false>,
        "critique":"<paragraph: note abrupt shifts and missing causes>",
        "suggestion":"<concrete edits to explain shift>",
        "evidence_spans":[ ... ],
        "emotion_jump_rate": <float>
     }
   }
 ],
 "aggregate": {
    "notes":"<if any>"
 }
}
```

ADDITIONAL RULES:

- Determine segments at chapter granularity by default; if chapters are very long, use paragraph-level segments.
- Evidence spans should show the antecedent (cause) and the emotional result where available.
- Mark low_confidence when emotional labels are ambiguous due to neutral narration.

Figure 5: Prompt for the Emotional Coherence evaluator.

ROLE: You are a pacing editor and scene-cohesion analyst.

INPUT:

- PREMISE: {premise}
- TEXT: {story_text}

TASK:

1. Segment TEXT into scenes (heuristic: line breaks, paragraph patterns, scene markers).
2. Compute event density, scene transition frequency, and flag over-/under-paced regions.
3. Evaluate Pacing & Scene Cohesion.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["SCORE2025"],
 "premise":"<premise>",
 "scene_summary":[
   {"scene_idx":1,"loc":"ch1:lines","length_tokens":<int>,"events_count":<int>},
   ...
 ],
 "scene_cohesion": {
   "score": <0-10>,
   "needs_revision": <true/false>,
   "critique":"<paragraph: identify dragging or rushed sections>",
   "suggestion":"<concrete fix: merge/split scenes; add beats>",
   "evidence_spans":[
     {"loc":"<loc>","quote":"<<=2 sentences>","issue":"<e.g., 'low event density'>"}
   ],
   "metrics": {
     "events_per_1k_tokens": <float>,
     "avg_scene_length_tokens": <float>,
     "scene_transition_rate": <float>
   },
   "low_confidence": <true/false>
 }
}
```

ADDITIONAL RULES:

- Define an 'event' as an action or change of state; when unsure, label conservatively.

Figure 6: Prompt for the Scene Cohesion evaluator.

ROLE: You are a thematic critic and motif tracker.

INPUT:

- PREMISE: {premise}
- TEXT: {story_text}
- TOP_K_THEMES: 5

TASK:

1. Identify main themes/motifs (topics, recurring imagery, central questions).
2. Track theme recurrence across chapters and evaluate whether themes are developed or fragmented.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["CHIRON2024","Papalampidi2022"],
 "premise":"<premise>",
 "themes":[
   {"theme":"<short label>","first_mention":"<loc>","recurrence_count":<int>,
        "sample_spans":[{"loc":"...","quote":"..."}]},
   ...
 ],
 "thematic_consistency": {
   "score": <0-10>,
   "needs_revision": <true/false>,
   "critique":"<paragraph: note theme fragmentation or weak development>",
   "suggestion":"<concrete fix: e.g., 'reintroduce motif X in ch7 in a pivotal scene'>",
   "metrics": {"theme_drift_index": <float>},
   "low_confidence": <true/false>
 }
}
```

ADDITIONAL RULES:

- Theme labels should be short human-readable phrases.

Figure 7: Prompt for the Thematic Consistency evaluator.

ROLE: You are a world-consistency and factuality auditor.

INPUT:
- PREMISE: {premise}
- TEXT: {story_text}
- WORLD_RULES: []

TASK:
1. Identify explicit and implicit world rules (physical laws, social structures, magic rules).
2. Verify whether TEXT violates stated or implied rules; report contradictions.

OUTPUT (Return ONLY JSON):

```
{
 "grounding": ["Papalampidi2022","SCORE2025"],
 "premise":"<premise>",
 "world_rules_inferred":[
   {"rule":"<short description>","first_mention":"<loc>","evidence_spans":[...]}
 ],
 "world_consistency": {
   "score": <0-10 or null if N/A>,
   "needs_revision": <true/false>,
   "critique":"<paragraph: list violation examples>",
   "suggestion":"<concrete fix>",
   "evidence_spans":[
     {"loc":"<loc>","quote":"<...>","issue":"<label>"}
   ],
   "low_confidence": <true/false>
 }
}
```

ADDITIONAL RULES:
- If the story is realistic contemporary fiction, world_consistency should check real-world factuality only for claims that are central to plot; do not score minor factual inaccuracies.
- If WORLD_RULES input is provided, strictly enforce those rules and prefer them over inferred rules.

Figure 8: Prompt for the World Consistency evaluator.

Role
You are a professional literary evaluator. Your task is to rate a story based on 6 criteria strictly adhering to the HANNA (2 022) framework.For each criterion, provide a score from 1 (Very Poor) to 5 (Excellent) based on the EXACT descriptions provided below.

Evaluation Criteria (HANNA 2022) & Anchors:
1. Relevance (RE): Measures how well the story matches its prompt.
Instruction Note: If the story is not relevant with respect to the prompt, it only affects the Relevance criterion! Do not r ate 1 everywhere based solely on relevance.
- 1: The story has no relationship with the prompt at all.
- 2: The story only has a weak relationship with the prompt.
- 3: The story roughly matches the prompt.
- 4: The story matches the prompt, except for one or two small aspects.
- 5: The story matches the prompt exactly.

2. Coherence (CH): Measures whether the story makes sense.
- 1: The story does not make sense at all. For instance, the setting and/or characters keep changing, and/or there is no understandable plot.
- 2: Most of the story does not make sense.
- 3: The story mostly makes sense but has some incoherences.
- 4: The story almost makes sense overall, except for one or two small incoherences.
- 5: The story makes sense from beginning to end.

3. Empathy (EM): Measures how well you understood the characters' emotions (regardless of whether you agreed with them).
- 1: The characters seemed apathetic to you.
- 2: At least one character slightly related to you on an emotional level.
- 3: You recognized specific, but not necessarily strong, emotions (e.g. sadness, joy, fear...) in at least one character.
- 4: At least one character emotionally involved you, but minor details prevented you from completely relating to them.
- 5: At least one character completely involved you on an emotional level.

4. Surprise (SU): Measures how surprising the end of the story was.
- 1: The ending seemed completely obvious from the start, or doesn't make any sense at all.
- 2: The ending was easily predictable after a few sentences.
- 3: The ending was predictable after half of the story.
- 4: The ending surprised you, but would have been difficult to predict.
- 5: The ending surprised you, and still seemed as if it could very reasonably have been predicted, i.e., there were enough c lues in the story.

5. Engagement (EG): Measures how much you engaged with the story.
- 1: You found the story boring and were glad it was over.
- 2: You found one or two things interesting in the story, but no more.
- 3: The story was mildly interesting.
- 4: The story almost kept you engaged until the end.
- 5: You were so engaged that you wished there was a sequel.

6. Complexity (CX): Measures how elaborate the story is.
Complex Elements List: complex concepts, realistic characters, an intricate plot, an underlying history or circumstances, precise descriptions.
- 1: The setting of the story is extremely simple; it only involves one or two characters or concepts.
- 2: The setting of the story is simple; one or two characters, a simple plot, maybe an indication of time or location.
- 3: The story is somewhat developed: it involves at least ONE of the complex elements listed above.
- 4: The story is developed: it involves at least TWO of the complex elements listed above.
- 5: The story is well thought -out: it involves at least THREE of the complex elements listed above.

Output Requirements:
You MUST respond in JSON format:

```
{
  "scores": {
    "RE": {"score": int, "reason": "short string explaining the score choice"},
    "CH": {"score": int, "reason": "short string explaining the score choice"},
    "EM": {"score": int, "reason": "short string explaining the score choice"},
    "SU": {"score": int, "reason": "short string explaining the score choice"},
    "EG": {"score": int, "reason": "short string explaining the score choice"},
    "CX": {"score": int, "reason": "short string explaining the score choice"}
  }
}
```

Figure 9: The HANNA (2022) rubric prompt used for general narrative quality evaluation.

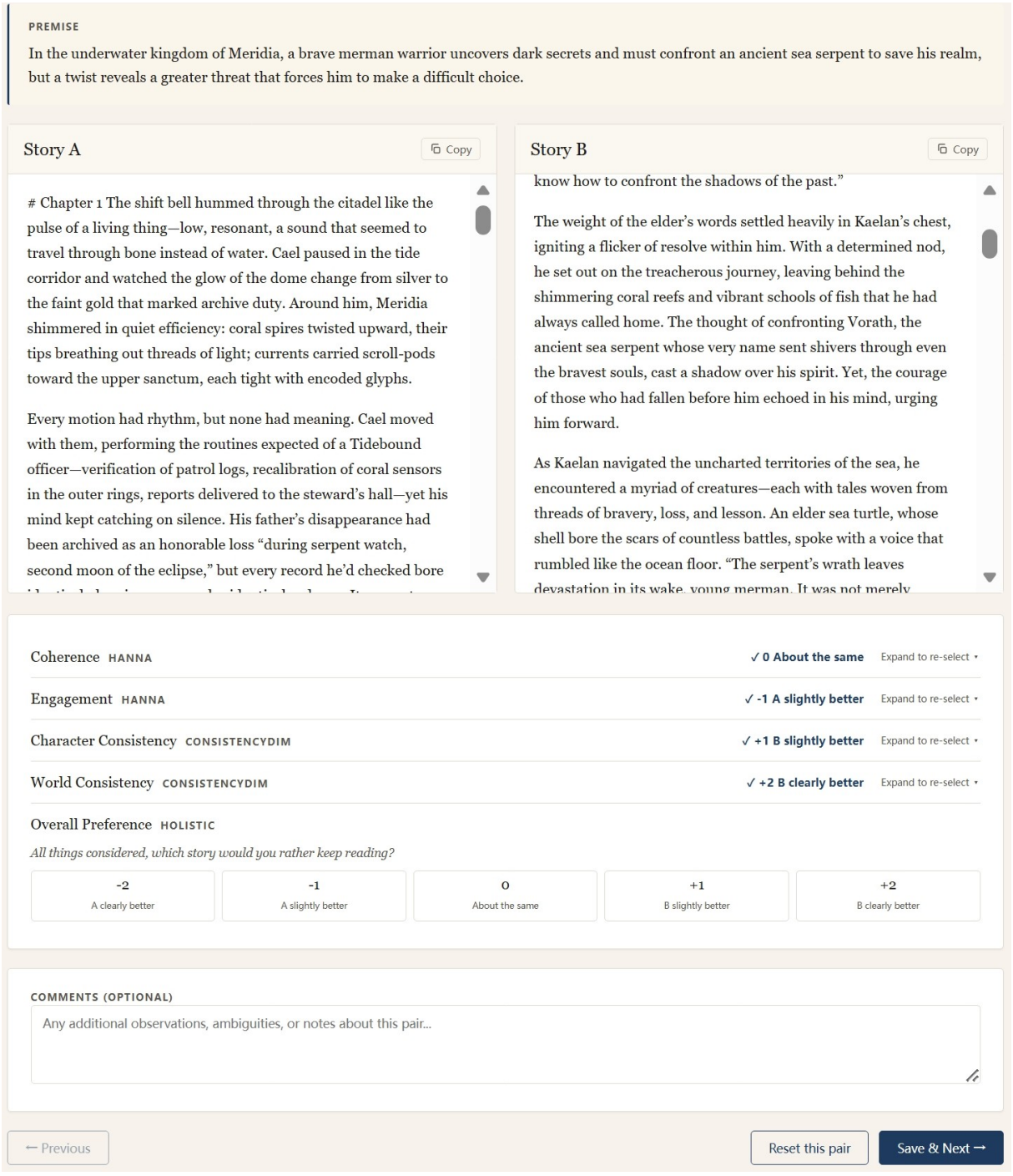


Figure 10: Screenshot of the blind pairwise human evaluation interface. Evaluators compare two anonymized stories generated from the same premise and rate them across five dimensions.