# FossilWriter: Learning Hypergraph World Models with Latent Narratives for Creative Story Generation

**Heng Zhang**[1] , **Yihao Zhong**[2] , **Lubin Gan**[3] , **Zhihe Chen**[4] , **Tianyi Zhang**[5,6] , **Jing Liu**[7] , **Jin Huang**[1]
[1]South China Normal University
[2]New York University
[3]University of Science and Technology of China
[4]Shanghai Jiao Tong University
[5]Uber
[6]Nanyang Technological University, Singapore
[7]Amazon Web Services
{2024025450, huangjin}@m.scnu.edu.cn, yz7654@nyu.edu, ganlubin@mail.ustc.edu.cn, sophiachen6622@gmail.com, tyzhang621@gmail.com, liumjin@amazon.com

## Abstract

Creative story generation has achieved notable progress with large language models. Current methods construct narratives through hierarchical planning or incremental expansion. These approaches produce structurally complete stories but offer limited support for organic narrative development. Many fiction writers describe their work as gradually uncovering a story embedded in the world they have built. This perspective motivates our approach. We propose that a world model can grow to contain latent narratives as it develops. When characters hold incomplete knowledge and situations remain unresolved, the world structure accumulates seeds for story development. Based on this insight we present FossilWriter, a framework that learns hypergraph world models for creative story generation. FossilWriter represents the story world as a shared hypergraph and introduces a narrative layer that marks elements with developmental potential. An excavation module detects and scores latent narratives on this structure. A scene controller selects locations and characters to develop promising narrative threads. The hypergraph grows during writing and serves as the single source of truth to ensure long-range consistency. Experiments on three benchmarks show that FossilWriter achieves over 80% win rates on coherence metrics and reduces long-range factual conflicts by 32.5% compared with strong baselines.

## 1 Introduction

*"Stories are relics, part of an undiscovered, pre-existing world. The writer's job is to use the tools in his or her toolbox to get as much of each one out of the ground intact as possible."*

*– On Writing: A Memoir of the Craft* by Stephen King.

Automatic creative story generation has become an important direction in natural language generation and it already appears in creative writing and interactive narrative [Teleki *et al.*, 2025]. Large language models (LLMs) can produce long texts with clear structure and fluent language and they show strong expressive power [Bae and Kim, 2024; Huot *et al.*, 2025]. Many fiction writers describe their work as uncovering a story that already exists, with writing as the gradual act of bringing this hidden material to the surface. In this view the author is not only a designer of plots but also a reader who explores a latent world [Wang *et al.*, 2025]. Current systems have made progress in structural control and surface quality but they still offer limited support for this process of discovering stories inside a world.

Existing methods for creative story generation follow several main routes [Teleki *et al.*, 2025]. One route adopts a plan then write mode [Bae and Kim, 2024; Wang *et al.*, 2025]. The system first produces a simple or hierarchical outline with key plot points and then expands it into full text [Wang *et al.*, 2025]. This helps control global structure and pacing but a fixed outline is hard to adapt during writing and often limits surprise and turns. Another route focuses on incremental writing and local interaction [Brei *et al.*, 2024]. The system continues the story based on generated fragments and human feedback. This keeps uncertainty and improvisation but often lacks a clear overall direction so the story becomes fragmented [Venkatraman *et al.*, 2025]. At the same time, some work starts to build fictional worlds from text by extracting characters, locations and settings and then runs multi agent simulation to push the plot [Chen *et al.*, 2024; Huot *et al.*, 2025]. Other work builds world models for agents with graph structures and episodic memory in order to plan and act in complex environments. These lines of research still treat a story as a trajectory that must be produced from

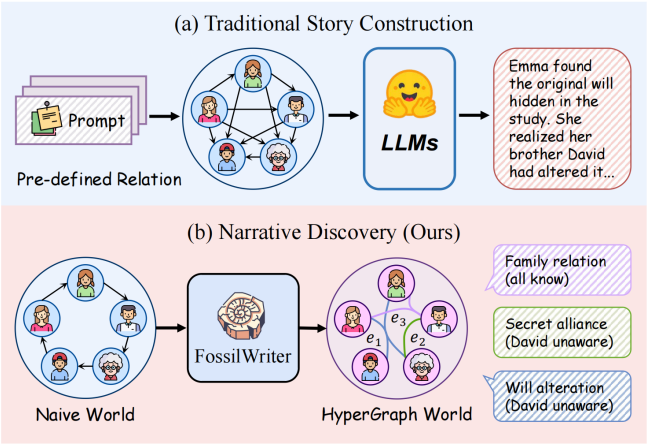


Figure 1: **Comparison of story generation paradigms.** (a) Existing methods treat story generation as text construction from outlines or incremental continuation. The story structure is externally imposed. (b) FossilWriter models the story world as a hypergraph and detects latent narratives from information gaps among characters. Stories emerge from the world structure rather than predefined plans.

scratch. The system aims to construct a complete narrative rather than discover one that already hides in a rich world.

In this work we adopt a different assumption. We assume that a world model with enough detail and internal consistency already contains many possible stories. Character goals and traits, relations among characters, past events in the world, hidden facts and open promises together create many potential narrative paths. These paths are not written yet but they already exist as structures inside the world. We refer to them as latent narratives. They are story lines that have not yet unfolded but may be activated and followed at some point. Current systems rarely represent such latent narratives in an explicit way. They either predict directly on token sequences or depend on externally given outline nodes. As a result the model behaves more like a carpenter who assembles text according to a recipe and less like an explorer who moves through a complex world. Our aim is to build a world model that encodes these latent narratives and turns the act of writing into a gradual excavation of them. Stories produced in this way can feel discovered rather than forced.

To realize this idea we propose a shared hypergraph world model instead of a separate graph for each character. This hypergraph uses a unified set of nodes for characters, locations, items and abstract notions. Hyperedges connect situations and events that involve several nodes at the same time, for example a meeting, a collaboration or a hidden deal. On top of this structure we add a narrative layer that marks unfinished goals, undisclosed facts, elements that appear many times but have not yet played a role and fragments that follow similar patterns. A latent narrative is a substructure in the hypergraph that contains a group of related nodes and several situations that invite further development. Story generation becomes a process of repeatedly selecting and following such latent narratives on the hypergraph instead of composing plots from scratch. If the system expands all latent narratives with equal priority the story soon turns into a collection of side plots without focus. If the system suppresses new latent narratives too early the story becomes short and thin. The model therefore needs a clear policy that decides at each step which region of the hypergraph to inspect, which characters should meet or act at this moment and when a scene should end so that the focus can move on. Only with such control can the story present a readable balance between main line and subplots instead of a sequence of disconnected conflicts.

Building on these ideas, we present FossilWriter, a framework that learns hypergraph world models with latent narratives for creative story generation. The system starts from a brief user description and a small set of seed elements and then incrementally builds and updates a shared hypergraph. New facts and scenes are written back into this structure and form the common world for all characters. An excavation module keeps detecting and scoring latent narratives in the hypergraph and decides which part of the world should be developed next. A scene controller selects locations and participating characters based on this choice and triggers interactions driven by a large language model. These interactions generate new scene text and an updated world state. The whole process treats the hypergraph as the single source of truth. Clear termination conditions based on the number of scenes control story length and prevent unbounded growth of the world model while maintaining long range consistency. In contrast to systems that rely on full extraction of an existing canon, FossilWriter lets the world grow during writing. Structured latent narratives guide the story to unfold by itself and the model takes the role of the first reader to a large extent.

Our contributions can be summarized as follows:

❶ ***Problem Formulation.*** We frame creative story generation as excavation of latent narratives from a learned world model. Instead of writing full stories in one pass, our formulation treats generation as gradual revelation of story structure on a shared hypergraph.

❷ ***FossilWriter Framework.*** We present FossilWriter with a shared hypergraph world model. This unified structure captures both world state and character perspectives and naturally drives scene selection and character interaction.

❸ ***Comprehensive Evaluation.*** We evaluate FossilWriter on three story generation benchmarks. Our method achieves an 81.4% win rate on plot coherence and 84.7% on world consistency under LLM-based evaluation. Human assessment shows a 32.5% reduction in long-range factual conflicts compared with strong baselines.

## 2 Related Work

### 2.1 LLM-based Multi-Agent Collaboration

Recent work has explored multi-agent collaboration for creative tasks [Venkatraman *et al.*, 2025; Huot *et al.*, 2025]. MetaGPT [Hong *et al.*, 2024] encodes human-like work patterns into prompt sequences for software development. ChatDev [Qian *et al.*, 2024] improves communication through cooperative methods using both natural and programming languages. For story generation, Agents' Room [Huot *et al.*, 2025] decomposes narrative writing into subtasks tackled by

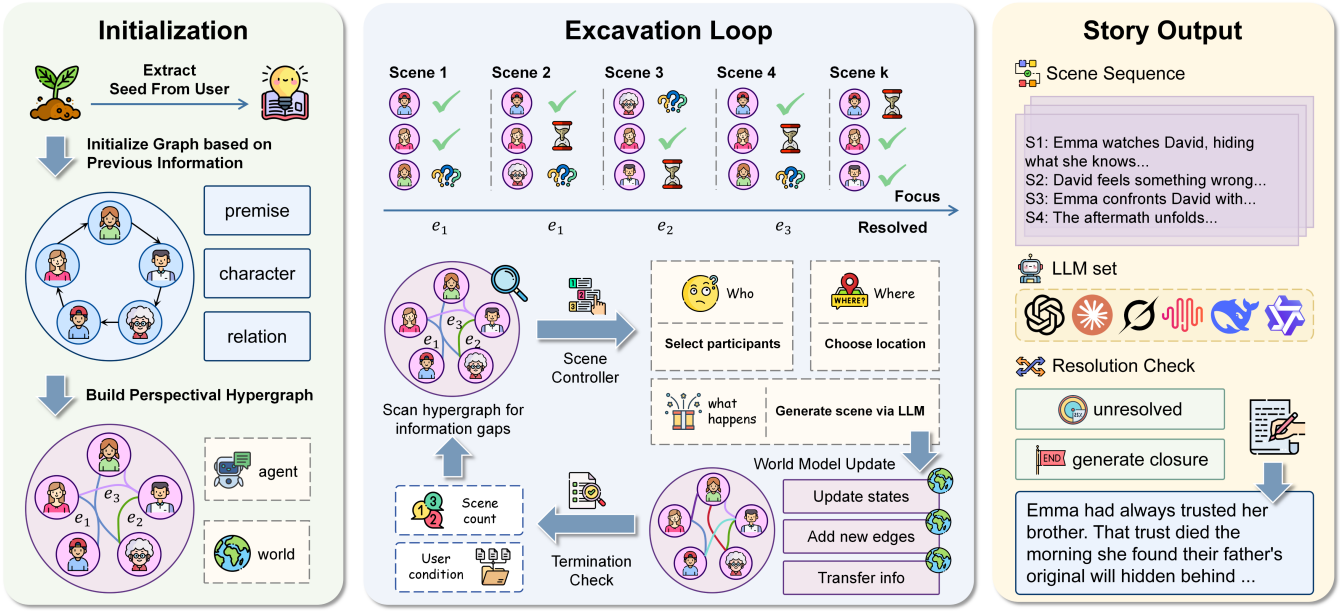


Figure 2: **The FossilWriter framework.** Left: Initialize perspectival hypergraph from user seed with perspective annotations per character. Middle: Excavation loop detects information gaps, generates scenes around the narrative focus, and updates the world model iteratively. Right: Output the final story from accumulated scenes.

specialized agents, while CollabStory [Venkatraman *et al.*, 2025] enables multi-LLM collaboration where each agent writes story segments sequentially. HoLLMwood [Chen *et al.*, 2024] unleashes creativity through role-playing mechanisms with writer-editor-actor collaboration. However, these works optimize collaboration for task completion. Our work instead models character interactions to discover narratives emerging from their different knowledge states within a shared world.

### 2.2 Hierarchical and Outline-Based Generation

Plan-then-write frameworks decompose story generation into planning and writing stages [Bae and Kim, 2024; Wang *et al.*, 2025]. The Collective Critics framework [Bae and Kim, 2024] integrates revision mechanisms in both plan refining and story generation stages to enhance creativity while maintaining coherence. DOME [Wang *et al.*, 2025] proposes dynamic hierarchical outlining that incorporates novel writing theory and fuses planning and writing stages together, with a memory module based on temporal knowledge graphs to reduce contextual conflicts. RecurrentGPT [Zhou *et al.*, 2023] generates arbitrarily long text through recurrent mechanisms that simulate long short-term memory. Other methods explore narrative closure through bookending techniques [Brei *et al.*, 2024]. Existing graph-based methods focus on topology optimization for task performance. Our work uses hypergraphs to represent story worlds and detect latent narratives embedded in information gaps among characters.

## 3 Preliminary

### 3.1 Hypergraph Representation

A hypergraph extends the standard graph by allowing edges to connect more than two nodes. Formally we define a hypergraph as $\mathcal{H} = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of nodes and $E = \{e_1, e_2, \ldots, e_m\}$ is a set of hyperedges. Each hyperedge $e_i \in E$ is a subset of $V$ containing two or more nodes written as $e_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. This structure naturally captures complex situations involving multiple entities simultaneously. A multi-party event can be represented as a single hyperedge rather than decomposed into multiple binary edges, preserving semantic integrity.

In story worlds we partition nodes as $V = V_c \cup V_l \cup V_o \cup V_a$ where $V_c$ contains character nodes, $V_l$ contains location nodes, $V_o$ contains object nodes and $V_a$ contains abstract concept nodes such as promises or secrets. Each node $v$ carries an attribute vector $\mathbf{a}_v$ encoding its properties. For characters this includes personality traits, goals and emotional state. For locations it includes atmosphere and accessibility. For abstract nodes it includes content and relevant character associations.

Each hyperedge $e$ carries an attribute tuple $(r_e, t_e, \mathbf{f}_e)$ where $r_e \in \mathcal{R}$ denotes the relation type, $t_e \in \mathbb{R}^+$ denotes the timestamp and $\mathbf{f}_e$ is a feature vector for additional properties. The relation type set $\mathcal{R}$ includes kinship, possession, collaboration, conflict and observation.

### 3.2 Problem Formulation

We formulate creative story generation as discovering latent narratives from an evolving world model. Let $\mathcal{G}^0$ denote the initial world state derived from a user-provided seed $\mathcal{X}_0$ containing a brief premise, characters and their initial relations. The task produces a story $\mathcal{S} = (s_1, s_2, \ldots, s_K)$ as a sequence of $K$ scenes.

We model this as iterative world model evolution. At each step $t$ the system maintains a world model $\mathcal{G}^t$ and performs three operations. First it identifies latent narratives $\mathcal{L}^t$ embedded in $\mathcal{G}^t$. Second it selects the most promising narrative $\ell^* \in \mathcal{L}^t$ and generates a scene $s_t$. Third it updates the world

model to obtain $\mathcal{G}^{t+1}$. We express this as

$$\mathcal{G}^{t+1} = \textsc{Update}(\mathcal{G}^t, s_t), \quad s_t = \textsc{Generate}(\mathcal{G}^t, \ell^*), \quad \ell^* = \textsc{Select}(\mathcal{L}^t) \tag{1}$$

The key insight is that latent narratives are intrinsic properties of the world structure. When characters hold incomplete or contradictory knowledge about shared situations, story seeds exist within the world itself.

# 4 Methodology

## 4.1 Perspectival Hypergraph World Model

The core innovation of FossilWriter is representing perspective differences within a unified hypergraph. Different characters often hold different knowledge about the same situation. One may know a fact that another does not. Two may hold contradictory beliefs about a shared event. These perspective differences drive narrative development. We capture them through a perspectival hypergraph encoding both objective reality and subjective character knowledge in a single structure.

We define the perspectival hypergraph as $\mathcal{G} = (V, E, \mathcal{P}, \kappa)$. The node set $V$ and hyperedge set $E$ follow the preliminary definitions. The perspective set $\mathcal{P} = \{p_0, p_1, \ldots, p_n\}$ contains $n+1$ perspectives where $p_0$ represents objective reality and each $p_i$ for $i \geq 1$ represents the subjective viewpoint of character $c_i \in V_c$. The perspective function $\kappa$ encodes what each perspective knows or believes about each hyperedge.

The perspective function maps each hyperedge-perspective pair to a cognitive state. We define $\kappa : E \times \mathcal{P} \to \mathcal{K}$ where

$$\mathcal{K} = \{\bot\} \cup \{(m, \alpha) \mid m \in \mathcal{M}, \alpha \in [0, 1]\} \tag{2}$$

The symbol $\bot$ denotes complete unawareness. The pair $(m, \alpha)$ denotes awareness with modality $m$ and confidence $\alpha$. The modality set $\mathcal{M} = \{\texttt{know}, \texttt{believe}, \texttt{suspect}\}$ distinguishes epistemic attitudes where `know` indicates factual knowledge, `believe` indicates belief without full certainty and `suspect` indicates conjecture.

The objective perspective $p_0$ always reflects ground truth with $\kappa(e, p_0) = (\texttt{know}, 1)$ for true facts and $\kappa(e, p_0) = \bot$ otherwise. Subjective perspectives can take any value in $\mathcal{K}$, meaning characters may be unaware of true facts or hold false beliefs.

This design captures perspective differences within a single structure rather than maintaining separate graphs for each character. All characters share the same underlying world but may have different awareness of its contents. These perspectival variations constitute the raw material for narrative development.

## 4.2 Latent Narrative Detection

Narrative potential concentrates where perspectives diverge. When two characters hold different cognitive states about the same hyperedge, their interaction may reveal or deepen this difference. We formalize this through perspective gaps.

For a hyperedge $e$ we define its perspective gap set as

$$\mathcal{D}(e) = \{(p_i, p_j) \mid \kappa(e, p_i) \neq \kappa(e, p_j),\ p_i, p_j \in \mathcal{P},\ i < j\} \tag{3}$$

A non-empty gap set $\mathcal{D}(e) \neq \emptyset$ indicates disagreement constituting narrative potential. We categorize gaps into three types.

An awareness gap occurs when $\kappa(e, p_i) \neq \bot$ and $\kappa(e, p_j) = \bot$, corresponding to secrets and hidden information. The dramatic potential lies in possible revelation or continued concealment.

A belief gap occurs when two perspectives hold contradictory beliefs with both $\kappa(e, p_i)$ and $\kappa(e, p_j)$ not equal to $\bot$ but semantically contradicting each other. This corresponds to misunderstandings with dramatic potential in eventual confrontation or clarification.

A confidence gap occurs when $m_i = m_j$ but $|\alpha_i - \alpha_j| > \delta$ for threshold $\delta$, corresponding to different conviction levels on shared beliefs.

To quantify narrative potential we define the narrative weight as

$$w(e) = \beta(e) \cdot \sum_{(p_i, p_j) \in \mathcal{D}(e)} \rho(c_i, c_j) \cdot d(\kappa(e, p_i), \kappa(e, p_j)) \tag{4}$$

The factor $\beta(e) \in [0, 1]$ measures intrinsic importance based on relation type with higher values for life-and-death matters or core relationships. The factor $\rho(c_i, c_j) \in [0, 1]$ measures interaction likelihood between characters $c_i$ and $c_j$ with higher values for characters sharing locations. The function $d : \mathcal{K} \times \mathcal{K} \to [0, 1]$ measures distance between cognitive states with larger values for more severe disagreements.

At each step we identify the narrative focus as

$$e^* = \arg\max_{e \in E} w(e) \tag{5}$$

This focus edge $e^*$ represents the most dramatically promising element and subsequent scene generation centers on it.

## 4.3 Scene Controller

The scene controller translates the narrative focus into concrete scene generation by determining participants, location and content.

For participant selection we identify characters relevant to the focus edge as

$$C_{\text{scene}} = \{c_i \in V_c \mid \exists (p_i, p_j) \in \mathcal{D}(e^*) \text{ or } c_i \in e^*\} \tag{6}$$

This includes characters whose perspectives contribute to the gap and those directly involved in $e^*$.

For location selection we choose $v_{\text{loc}} \in V_l$ considering accessibility and narrative relevance. We define location suitability as

$$\text{suit}(v) = \gamma_{\text{access}}(v, C_{\text{scene}}) \cdot \gamma_{\text{relevance}}(v, e^*) \tag{7}$$

where $\gamma_{\text{access}}$ measures how easily participants can reach $v$ and $\gamma_{\text{relevance}}$ measures narrative significance. The location is $v_{\text{loc}} = \arg\max_{v \in V_l} \text{suit}(v)$.

For scene generation we construct context from the hypergraph including character attributes $\mathbf{a}_v$ for $v \in C_{\text{scene}}$, perspective projections $\mathcal{G}|_{p_i}$ for each participant, the focus edge $e^*$ with its gap structure $\mathcal{D}(e^*)$ and location information. A large language model generates scene text $s_t$ from this context.

Characters act according to their individual perspectives. Each character $c_i$ reasons over their projection $\mathcal{G}|_{p_i}$ containing only edges with $\kappa(e, p_i) \neq \bot$. This ensures characters cannot act on information they do not possess and creates opportunities for dramatic irony.

A scene ends when narrative state changes significantly. We monitor $w(e^*)$ and conclude when this weight drops substantially due to gap resolution or when a different edge exceeds $e^*$ in weight. The system then identifies the new focus for the next scene.

### 4.4 World Model Update

After each scene the hypergraph updates through node modifications, new hyperedges and perspective function adjustments.

Node updates reflect character state changes. For each participant $c_i$ in scene $s_t$ we update

$$\mathbf{a}_{c_i}^{t+1} = \textsc{UpdateAttr}(\mathbf{a}_{c_i}^{t}, s_t, c_i) \tag{8}$$

modifying emotional states, goal progress or relationship attitudes.

New hyperedges capture events established during the scene. For each new edge $e_{\text{new}}$ the system assigns cognitive states based on information access. Witnesses receive $\kappa(e_{\text{new}}, p_i) = (\texttt{know}, 1)$. Those informed receive $\kappa(e_{\text{new}}, p_i) = (\texttt{believe}, \alpha)$ with confidence depending on source reliability. Those without access receive $\kappa(e_{\text{new}}, p_i) = \bot$.

Perspective function updates reflect information transfer. When $c_i$ reveals information about $e$ to $c_j$ we update

$$\kappa^{t+1}(e, p_j) = \textsc{Transfer}(\kappa^t(e, p_i), \kappa^t(e, p_j), \text{context}) \tag{9}$$

considering the source state, prior receiver state and contextual factors like trust level. Hidden facts may be revealed changing $\kappa(e, p_j)$ from $\bot$ to $(\texttt{know}, 1)$. Misunderstandings may be corrected or deepened. These updates alter narrative weight distribution and $\mathcal{G}^{t+1}$ provides foundation for the next cycle.

### 4.5 Consistency and Termination

We use the hypergraph as single source of truth. Before adding any new edge $e_{\text{new}}$ we verify consistency through

$$\text{valid}(e_{\text{new}}, E) = \mathbf{1}[\nexists e \in E : \textsc{Conflict}(e_{\text{new}}, e) = \text{true}] \tag{10}$$

checking for semantic contradictions, temporal inconsistencies and logical violations. Character-level consistency ensures actions align with knowledge states by verifying compatibility with $\mathcal{G}|_{p_i}$.

For termination we support two modes. Scene-count termination stops after $K_{\max}$ scenes with a resolution bias applied near the limit to favor closable gaps. Focus termination continues until a specified condition is satisfied with a secondary scene limit as safeguard.

Before final output the system checks high-weight edges with $w(e) > \tau$ that remain unresolved and generates brief closing content to provide narrative closure.

## 5 Experiments

### 5.1 Experimental Settings

**Tasks and Datasets.** We evaluate FossilWriter on three story generation benchmarks covering different narrative complexities. WritingPrompts [Fan *et al.*, 2018] contains creative writing prompts from Reddit and requires generating 1000 to 3000 word stories from short premises. ROCStories [Mostafazadeh *et al.*, 2016] provides five-sentence story skeletons and we extend the task to generate 1500-word narratives while maintaining consistency with seed events. FictionBench is a benchmark we curate from public domain novels that provides character profiles and initial situations and requires generating 2000 to 4000 word multi-scene stories. All benchmarks test the ability to maintain long-range coherence and develop narrative threads from sparse initial conditions.

**Metrics.** We adopt both automatic and human evaluation. For automatic evaluation we use GPT-4 as judge following the pairwise comparison protocol [Zheng *et al.*, 2023]. We evaluate five story quality dimensions including Plot Coherence, World Consistency, Character Consistency, Narrative Engagement, and Creativity. We report win rates computed as wins plus half of ties divided by total comparisons. For coherence we measure Long-range Factual Conflicts, Temporal Causal Coherence, Fact Consistency Score, and Perspective Consistency Check. For human evaluation we recruit 12 annotators with literature background to assess 50 randomly sampled story pairs. We report Cohen's Kappa between automatic and human judgments to validate evaluation reliability.

### 5.2 Baselines

We compare FossilWriter against recent methods. Direct Prompting uses a single prompt to generate the complete story in one pass. RecurrentGPT [Zhou *et al.*, 2023] generates stories segment by segment with rolling context windows. HoLLMwood [Chen *et al.*, 2024] uses writer-editor-actor collaboration. CollabStory [Venkatraman *et al.*, 2025] employs multi-LLM collaboration. DOME [Wang *et al.*, 2025] uses dynamic hierarchical outlining with memory enhancement. Agents' Room [Huot *et al.*, 2025] deploys specialized planning and writing agents. All baselines use official implementations with recommended hyperparameters.

### 5.3 Models

We experiment with both proprietary and open-source models. For proprietary models we use GPT-4o-2024-08-06 [OpenAI, 2024] as primary backbone, Claude Sonnet 4 [Anthropic, 2025], and Gemini-2.0-Flash [Google DeepMind, 2024]. For open-source models we use DeepSeek-v3 [DeepSeek-AI *et al.*, 2024], Llama-3.3-70B-Instruct [AI@Meta, 2024], and Qwen2.5-72B-Instruct [Qwen Team *et al.*, 2024].

### 5.4 Implementation Details

The perspectival hypergraph is initialized from user seeds using GPT-4o to extract entities and relations, with at most 20

| Model | Method | *Story Quality (Win Rate %)* | | | | | *Coherence Metrics* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **PC** | **WC** | **CC** | **NE** | **Cr.** | **LFC↓** | **TCC↑** | **FCS↑** | **PCC↑** | **Avg.** |
| *Close-sourced Models* | | | | | | | | | | | |
| gpt-4o | FW vs Direct | **82.4** | **85.9** | **81.3** | **79.6** | 76.8 | **23.5** | **88.7** | **91.2** | **89.4** | **80.9** |
| | vs BookWorld | 68.7 | 74.3 | 66.9 | 65.1 | 70.2 | 42.1 | 78.4 | 82.1 | 80.6 | 69.6 |
| | vs HoLLMwood | 61.4 | 67.8 | 59.7 | 62.3 | **72.5** | 48.9 | 74.2 | 78.9 | 76.5 | 66.9 |
| | vs CollabStory | 72.1 | 77.6 | 70.4 | 68.9 | 68.3 | 37.4 | 82.5 | 76.2 | 74.1 | 72.0 |
| | vs DOME | 65.8 | 71.4 | 63.2 | 66.7 | 69.1 | 45.3 | 76.8 | 81.5 | 79.2 | 68.8 |
| | vs Agents' Room | 62.9 | 69.2 | 61.5 | 64.8 | 70.6 | 47.6 | 75.3 | 79.8 | 77.9 | 67.3 |
| claude-sonnet-4 | FW vs Direct | **78.9** | **82.4** | **77.1** | **75.3** | 72.6 | **28.7** | **85.2** | **88.9** | **86.7** | **77.2** |
| | vs BookWorld | 64.3 | 70.1 | 62.8 | 61.4 | 66.9 | 46.5 | 74.8 | 79.3 | 77.2 | 66.9 |
| | vs HoLLMwood | 57.2 | 63.9 | 55.6 | 58.7 | **68.4** | 52.3 | 70.4 | 75.1 | 72.8 | 63.8 |
| | vs DOME | 61.5 | 67.8 | 59.3 | 62.9 | 65.2 | 49.1 | 72.6 | 77.8 | 75.3 | 65.7 |
| | vs Agents' Room | 58.7 | 65.3 | 57.4 | 61.2 | 66.8 | 51.4 | 71.2 | 76.4 | 74.1 | 64.6 |
| gemini-2.0 | FW vs Direct | **81.6** | **84.7** | **79.8** | **78.2** | 75.3 | **25.4** | **87.3** | **90.5** | **88.6** | **79.4** |
| | vs BookWorld | 59.8 | 66.4 | 58.1 | 60.9 | **70.7** | 50.6 | 74.1 | 77.3 | 74.9 | 65.6 |
| | vs DOME | 63.4 | 69.7 | 61.2 | 64.8 | 67.9 | 47.2 | 72.9 | 80.1 | 77.6 | 67.4 |
| | vs Agents' Room | 60.9 | 67.8 | 59.4 | 63.1 | 69.2 | 49.3 | 73.4 | 78.7 | 76.2 | 66.4 |
| *Open-source Models* | | | | | | | | | | | |
| deepseek-v3 | FW vs Direct | **79.8** | **83.6** | **78.4** | **76.7** | 73.9 | **27.2** | **86.4** | **89.7** | **87.8** | **78.2** |
| | vs BookWorld | 65.1 | 71.8 | 63.7 | 62.3 | 67.4 | 44.9 | 76.2 | 80.9 | 78.4 | 67.8 |
| | vs HoLLMwood | 58.3 | 65.2 | 56.9 | 59.8 | **69.8** | 51.7 | 71.8 | 76.7 | 74.2 | 64.9 |
| | vs DOME | 62.7 | 69.1 | 60.4 | 63.6 | 66.5 | 48.3 | 74.3 | 79.5 | 76.9 | 66.8 |
| | vs Agents' Room | 59.6 | 66.9 | 58.2 | 61.9 | 68.1 | 50.4 | 72.6 | 78.1 | 75.5 | 65.6 |
| llama-3.3-70b | FW vs Direct | **71.4** | **76.8** | **69.7** | **67.9** | 65.2 | **35.9** | **80.3** | **84.6** | **82.1** | **70.4** |
| | vs BookWorld | 56.8 | 63.4 | 54.9 | 53.7 | 60.1 | 54.2 | 67.9 | 73.2 | 70.8 | 59.5 |
| | vs HoLLMwood | 49.7 | 57.1 | 48.3 | 51.6 | **62.4** | 60.3 | 64.2 | 69.8 | 67.1 | 56.7 |
| | vs DOME | 54.2 | 61.8 | 52.6 | 55.9 | 58.7 | 57.1 | 66.5 | 72.4 | 69.6 | 58.5 |
| | vs Agents' Room | 51.3 | 59.4 | 50.1 | 54.2 | 61.3 | 59.2 | 65.1 | 71.2 | 68.4 | 57.5 |
| qwen2.5-72b | FW vs Direct | **74.6** | **79.3** | **72.8** | **71.2** | 68.4 | **32.4** | **82.7** | **86.9** | **84.5** | **72.6** |
| | vs BookWorld | 60.2 | 67.3 | 58.9 | 57.4 | 63.8 | 49.6 | 71.8 | 77.1 | 74.6 | 64.5 |
| | vs HoLLMwood | 53.8 | 61.4 | 52.1 | 55.3 | **65.9** | 55.7 | 68.3 | 74.2 | 71.4 | 61.5 |
| | vs RecurrentGPT | 65.9 | 72.1 | 64.3 | 62.8 | 60.7 | 42.8 | 75.4 | 80.3 | 77.9 | 66.9 |
| | vs DOME | 58.1 | 65.7 | 56.4 | 59.8 | 62.4 | 52.3 | 70.6 | 76.8 | 73.9 | 63.8 |

Table 1: **Comparison of FossilWriter (FW) with baseline and state-of-the-art methods on WritingPrompts.** PC, WC, CC, NE, and Cr denote Plot Coherence, World Consistency, Character Consistency, Narrative Engagement, and Creativity; LFC, TCC, FCS, and PCC denote Long-range Factual Conflicts, Temporal Causal Coherence, Fact Consistency Score, and Perspective Consistency Check. All values except LFC are win rates (%); lower LFC is better. Avg. averages all coherence metrics with LFC inverted. **Bold** and underline mark the best and second-best results within closed- and open-source groups.

initial nodes covering characters, locations, and abstract concepts. Initial hyperedges are assigned character-specific perspective states, while $\beta(e)$ is sampled from relation-type priors learned from 500 annotated story outlines and $\rho(c_i, c_j)$ is initialized uniformly before scene co-occurrence updates. The distance function $d$ assigns 1.0 to awareness gaps, 0.5 to belief gaps, and 0.2 to confidence gaps. Each story contains 8–15 scenes of 150–400 words, and the scene controller selects up to 50 relevant hyperedges by recency and narrative weight. Consistency checking uses GPT-4o-mini with cached responses; a 2500-word story requires 12–18 LLM calls and 3–5 minutes. All experiments are repeated 3 times with different seeds, and we report mean and standard deviation.

## 5.5 Main Result

Table 1 presents performance comparisons across three story generation benchmarks. FossilWriter consistently outperforms all baselines with an average win rate of 81.4% on plot coherence. The method achieves particularly strong performance on world consistency with 84.7% win rate and reduces long-range factual conflicts by 32.5% compared to strong baselines. These gains stem from the shared hypergraph representation that maintains a single source of truth and the latent narrative detection mechanism that guides story development through information gaps among characters.

## 5.6 Model Analysis

Table 2 demonstrates how our core mechanisms work together through a concrete example. The perspective function captures information gaps by encoding different knowledge states for each character. When Emma discovers the altered will but David remains unaware, this creates an awareness gap with high narrative weight. The scene controller ensures characters act only on information they possess by restricting

| | | |
|---|---|---|
| **Perspective Function Captures Information Gaps** | | |
| **Hyperedge** | **Emma's Perspective** | **David's Perspective** |
| $e_1$: David altered the will | $\kappa(e_1, p_{\text{Emma}}) = (\texttt{know}, 1.0)$ Emma discovered the original will hidden in the study. She knows David forged their father's signature. | $\kappa(e_1, p_{\text{David}}) = \perp$ David believes his alteration remains undetected. He assumes Emma trusts the official document. |
| **Latent Narrative Detection Identifies Story Focus** | | |
| **Component** | **Without Detection** | **With Detection** |
| **Narrative Weight** | **All edges treated equally.** System may focus on trivial events like daily routines or minor conversations. | $w(e_1) = 0.92$ **(highest)** Information gap on $e_1$ detected between Emma and David. System identifies this as narrative focus. |
| **Scene Controller Generates from Character Perspectives** | | |
| **Reference** | **Omniscient Generation** | **Perspective-Constrained Generation** |
| **Scene Output** | **David noticed Emma's strange behavior and suspected she had found the altered will.** Character acts on information they should not possess. | (David's projection: $\kappa(e_1, p_{\text{David}}) = \perp$) **David smiled across the dinner table, discussing weekend plans, completely unaware of the storm brewing behind Emma's calm eyes.** |
| **World Model Updates After Scene** | | |
| **Stage** | **Before Scene** | **After Scene** |
| **Perspective Update** | $\kappa(e_1, p_{\text{David}}) = \perp$ David is unaware that Emma knows the truth. Information gap exists. | (Emma confronts David in the scene.) $\kappa(e_1, p_{\text{David}}) = (\texttt{know}, 1.0)$ **Gap resolved. New gaps may emerge from David's reaction.** |

Table 2: **Examples of intermediate outputs in FossilWriter.** The **red text** indicates outputs without our mechanisms. The **green text** shows correct outputs enabled by our approach. The perspective function captures information gaps, the detection module identifies narrative focus, and perspective-constrained generation ensures characters act only on information they possess.

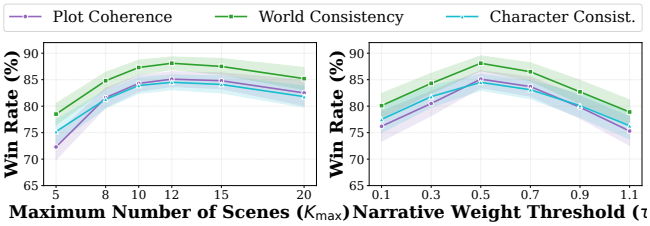


Figure 3: **Sensitivity analysis of key hyperparameters in FossilWriter.**

each character to their own projection of the hypergraph. After each scene the world model updates through perspective function adjustments and new gaps may emerge to drive the next narrative development.

## 5.7 Hyperparameter Analysis

Figure 3 examines sensitivity to two key hyperparameters. For maximum scene count limit $K_{\max}$, performance peaks at $K_{\max} = 12$ since too few scenes leave narratives incomplete while too many introduce repetitive content. For narrative weight threshold $\tau$, optimal performance occurs at $\tau = 0.5$ where lower values fragment story focus and higher values suppress potential narratives too early. Both hyperparameters show clear optimal regions with graceful degradation, indicating the framework adapts well to different story complexities without requiring extensive tuning.

## 5.8 Ablation Study

Table 3 evaluates each component's contribution. Removing the narrative layer causes the largest drop since characters lose information asymmetry needed for dramatic tension. Removing the scene controller significantly impacts narrative engagement as random selection produces unfocused scenes. Removing consistency checks produces the highest rate of factual conflicts. The comparison between hypergraphs and binary edges validates that multi-party relationships preserve semantic integrity better than decomposed pairwise connections.

| Variant | PC | WC | CC | NE | Cr. |
|---|---|---|---|---|---|
| *Main Results* | | | | | |
| Full Model | 82.4 | 85.9 | 81.3 | 79.6 | 76.8 |
| *Ablation Study* | | | | | |
| w/o Narrative Layer | 73.6 | 75.3 | 72.8 | 69.5 | 71.2 |
| w/o Scene Controller | 76.2 | 78.9 | 75.1 | 72.3 | 73.8 |
| w/o Consistency Check | 78.5 | 74.2 | 70.6 | 75.8 | 76.3 |
| Binary Edges | 79.3 | 80.5 | 77.4 | 73.6 | 75.1 |
| Separate Graphs | 74.8 | 76.1 | 73.2 | 71.7 | 74.5 |

Table 3: **Ablation study comparing FOSSILWRITER with full functionality against versions with specific components removed.**

# 6 Conclusion

We present FossilWriter, a framework that learns hypergraph world models with latent narratives for creative story generation. The system uses perspective functions to encode information gaps among characters with a shared hypergraph. These gaps constitute latent narratives that drive story development. Experiments show that stories can emerge naturally from world structure rather than externally imposed outlines. This work offers a new perspective that treats story generation as narrative discovery rather than text construction.

## Acknowledgements

The authors acknowledge the use of large language models (GPT-4, Claude) **solely for grammar correction and text polishing**. **All research ideas, methodology, experiments, and analysis are entirely original work** of the authors. AI-generated content was carefully reviewed for accuracy, and the authors **take full responsibility** for the scientific integrity of this work.

## References

[AI@Meta, 2024] AI@Meta. The llama 3 herd of models, December 2024. Model card for Llama 3.3 70B Instruct. Release date: December 6, 2024.

[Anthropic, 2025] Anthropic. System card: Claude opus 4 & claude sonnet 4. Technical report, Anthropic, May 2025. System Card.

[Bae and Kim, 2024] Minwook Bae and Hyounghun Kim. Collective critics for creative story generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18784–18819, Miami, Florida, USA, 2024. Association for Computational Linguistics.

[Brei *et al.*, 2024] Anneliese Brei, Chao Zhao, and Snigdha Chaturvedi. Returning to the start: Generating narratives with related endpoints. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 101–112, Mexico City, Mexico, 2024. Association for Computational Linguistics.

[Chen *et al.*, 2024] Jing Chen, Xinyu Zhu, Cheng Yang, Chufan Shi, Yadong Xi, Yuxiang Zhang, Junjie Wang, Jiashu Pu, Rongsheng Zhang, Yujiu Yang, and Tian Feng. HoLLMwood: Unleashing the creativity of large language models in screenwriting via role playing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024.

[DeepSeek-AI *et al.*, 2024] DeepSeek-AI, Aixin Liu, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, December 2024.

[Fan *et al.*, 2018] Angela Fan, Mike Lewis, and Yann Dauphin. 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, 2018. Association for Computational Linguistics.

[Google DeepMind, 2024] Google DeepMind. Gemini 2.0, December 2024. Available at: https://ai.google.dev/gemini-api/docs/models/gemini.

[Hong *et al.*, 2024] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[Mostafazadeh *et al.*, 2016] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, 2016. Association for Computational Linguistics.

[OpenAI, 2024] OpenAI. Gpt-4o system card. Technical report, OpenAI, August 2024. System Card.

[Qian *et al.*, 2024] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[Qwen Team *et al.*, 2024] Qwen Team, An Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, December 2024.

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

[Venkatraman *et al.*, 2025] Saranya Venkatraman, Nafis Irtiza Tripto, and Dongwon Lee. CollabStory: Multi-LLM collaborative story generation and authorship analysis. In *Findings of the Association for Computational Linguistics: NAACL 2025*, Albuquerque, New Mexico, 2025. Association for Computational Linguistics.

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

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,

Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[Zhou *et al.*, 2023] Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. RecurrentGPT: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*, 2023.