Planning Beyond Text: Graph-based Reasoning for Complex Narrative Generation

Hanwen Gu[1,2,*] **Chao Guo**[1,*,†] **Junle Wang**[2] **Wenda Xie**[1] **Yisheng Lv**[1]
[1]Institute of Automation, Chinese Academy of Sciences
[2]Tencent Turing Lab
{chao.guo, yisheng.lv}@ia.ac.cn

Abstract

While LLMs demonstrate remarkable fluency in narrative generation, existing methods struggle to maintain global narrative coherence, contextual logical consistency, and smooth character development, often producing monotonous scripts with structural fractures. To this end, we introduce PLOTTER, a framework that performs narrative planning on structural graph representations instead of the direct sequential text representations used in existing work. Specifically, PLOTTER executes the Evaluate-Plan-Revise cycle on the event graph and character graph. By diagnosing and repairing issues of the graph topology under rigorous logical constraints, the model optimizes the causality and narrative skeleton before complete context generation. Experiments demonstrate that PLOTTER significantly outperforms representative baselines across diverse narrative scenarios. These findings verify that planning narratives on structural graph representations—rather than directly on text—is crucial to enhance the long context reasoning of LLMs in complex narrative generation.

1 Introduction

Although LLMs demonstrate remarkable proficiency in storytelling, long-form complex narrative generation remains challenging. It requires not only linguistic fluency but also the construction of a logically grounded narrative structure (Mirowski et al., 2023; Chen et al., 2024; Guo et al., 2023). To realize coordinated planning for long-form narratives, recent work has introduced iterative hierarchical planning (Yang et al., 2022, 2023; Xie et al., 2026) or role-playing strategies (Chen et al., 2024; Han et al., 2024) via LLM agents.

However, these methods still struggle to maintain global narrative coherence, contextual logical consistency, and smooth character development, often producing monotonous scripts with structural fractures such as insufficient character motivation and a lack of necessary twists (Marco et al., 2024; Spangher et al., 2024; Wang et al., 2025b). We argue the primary reason is that narrative planning directly on text representations is inefficient. Without explicit modeling of plot dependencies, such systems cannot effectively reason about the underlying cause-and-effect web or the evolving relationships among characters and events, ultimately limiting their ability to produce rigorous narrative structures (Sun et al., 2023; Zhang et al., 2024).

To address this issue, we introduce **PLOTTER**, which stands for **PL**anning Bey**O**nd **T**ext: Graph-based Reasoning for Complex Narra**T**ive Gen**ER**ation. Unlike prior methods that plan on text directly, we plan narratives on graph structures, transforming script generation from a sequence planning problem into a dynamic graph generation and refinement problem. By applying the *Evaluate-Plan-Revise* cycle onto the narrative topology, PLOTTER enables the model to diagnose and repair structural inconsistencies at the causal level before textual realization.

We represent narrative dynamics through two interacting structures: an *Event Graph*, capturing the causal structure and skeleton of the plot, and a *Character Graph*, modeling inter-character relationships. This design draws inspiration from classic narratological theories regarding action logic (Barthes and Miller, 1970) and character networks (Moretti, 2011). Consequently, the challenge of script writing is decomposed into atomic graph-editing operations—adding, deleting, or re-linking nodes and edges—to resolve causal gaps and character inconsistencies.

To enforce structural integrity, we deploy a graph-grounded refinement cycle with a multi-agent critique module to audit symbolic structures rather than raw text. These agents identify weak-

[*]These authors contributed equally.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 37579–37610

nesses—such as disconnected causal paths—and formulate revision strategies. A *Constrained Graph Editor* then executes atomic operations to repair the narratives.

We evaluate our framework across diverse narrative scenarios using multiple mainstream LLM backbones. Results show that our graph-based narrative reasoning approach significantly outperforms existing methods.

Our main contributions are as follows:

- We propose **PLOTTER**, the first framework to perform narrative planning on structural graph representations rather than the direct sequential text in existing work, enhancing long context causal reasoning in complex narrative generation.
- We construct an LLM agent system for the iterative structural refinement of event and character graphs. By coordinating specialized agents to execute the *Evaluate-Plan-Revise* cycle on the graph topology, we achieve precise structural diagnosis and repairs that are inaccessible to text-based narrative planners.
- Experiments show that **PLOTTER** significantly outperforms strong existing methods, verifying that elevating narrative planning from direct text representations to explicit graph representations is crucial for enhancing long-range reasoning in complex narratives.

2 Related Work

The landscape of automated narrative generation has evolved from sequence-to-sequence generation to LLM-based planning. We categorize existing literature into two streams: outline-based and role-play based narrative planning.

2.1 Outline Based Narrative Planning

To improve the coherence in long narrative contexts, most approaches adopt a "Plan-and-Write" paradigm and decompose the generation task into hierarchical stages guided by the planned outline. Frameworks like Re3 (Yang et al., 2022) utilize recursive prompting for context maintenance, while Detailed Outline Control (DOC) (Yang et al., 2023) and DOME (Wang et al., 2025a) impose strict constraints to align generation with high-level outlines. Recent advancements such as CONCOCT (Wang et al., 2023) further refine this by dynamically evaluating outline pacing. However, because these methods operate sequentially, logical inconsistencies in early steps often propagate downstream, causing cascading errors (Yang et al., 2022). Furthermore, treating the outline as a rigid constraint restricts the flexibility required for complex revisions while avoiding extensive rewriting of preceding contexts (Yang et al., 2023).

2.2 Role-Play Based Narrative Planning

Recent research has shifted focus from static planning to dynamic multi-agent simulation, where frameworks like HoLLMwood (Chen et al., 2024), Agents' Room (Huot et al., 2025), IBSEN (Han et al., 2024), and StoryWriter (Xia et al., 2025) assign specialized personas (e.g., Director, Actor) to distinct LLM instances. While these role-playing paradigms excel at stylistic diversity and dialogue richness, their coordination relies predominantly on unstructured natural language. Research indicates that such purely textual critique loops are susceptible to ambiguity and semantic drift over long contexts (Bae and Kim, 2024). Consequently, without a shared symbolic state to ground collaboration, instructions from high-level planners are often misinterpreted by downstream agents, leading to hallucinations that contradict established narrative facts.

2.3 Graph-Based LLM Reasoning

Recent research integrates graph structures to enhance multi-hop reasoning for LLMs. For instance, Reasoning with Graphs (RwG) (Han et al., 2025) structures implicit context into explicit graphs, Think-on-Graph (ToG) (Sun et al., 2024) utilizes knowledge graphs for deep inference, and TG-LLM (Xiong et al., 2024) employs temporal graphs to bolster chronological understanding. In the specific domain of LLM story generation, recent work, such as R^2 (Lin et al., 2025), extracts graphs from a complete source text to build static external memories for generation reference. These works utilize graph structures as fixed contextual references for generation, while PLOTTER performs dynamic narrative planning via graph-structured representations of events and characters.

Unlike prior work that performs story planning directly on text representations, we use LLMs to plan narrative structure on graph-based representations. Specifically, we ensure logical and relational consistency through iterative generation and editing of event and character graphs via atomic edit

operations, thereby improving plot coherence, engagement, and character development.

3 Method

3.1 Task Definition

Given a concise premise P, our goal is to generate a complete, high-quality script S. We propose a framework $\mathcal{F}$ that decomposes this complex generation task into three consecutive stages: (1) **Graph-based Narrative Planning**, which constructs the initial event backbone and character relation; (2) **Iterative Graph Refinement**, which optimizes the narrative (event and character) graph through an iterative *Evaluate-Plan-Revise* cycle; and (3) **Graph-Grounded Script Synthesis**, which transforms the structured graph into a natural language script. The overall framework is illustrated in Figure 1.

3.2 Stage 1: Graph-based Script Planning

First, a Title Generator $\mathcal{N}$ derives a thematic title from the premise P (see Prompt 12 in Appendix). Subsequently, a Joint Graph Generator $\mathcal{G}$ constructs the initial event and character graph representation, explicitly delineating both causal event dependencies and the web of character relationships (see Prompt 13 in Appendix):

$$G^{(0)} = \mathcal{G}(\mathcal{N}(P), P) \tag{1}$$

where $G^{(0)} = (G_e^{(0)}, G_c^{(0)})$ denotes the initial joint graph structure.

Event Graph Construction. Following Causal Network Theory (Trabasso and van den Broek, 1985), the Event Graph $G_e = (V_e, E_e)$ functions as the causal skeleton. Each node $v_e \in V_e$ represents a distinct plot event with explicit attributes, including event ID, event description, narrative stage (e.g., *Rising Action*, *Climax*), and time index. The directed edges E_e are not merely temporal links but encode narrative relation labels $\rho(e) \in \{\textsc{Causal}, \textsc{Foreshadowing}, \textsc{Suspense}\}$. The Event Graph is generated through a structured prompt that instantiates these node attributes and edge relations under the above constraints (see Prompt 13 in Appendix).

Character Graph Construction. The Character Graph $G_c = (V_c, E_c)$ captures the sociodynamics of the narrative world. Each node v_c encodes multi-dimensional persona slots—including core personality trait, internal conflict, external goal, and hidden secret—while edges E_c represent evolving relationships with typed categories (Conflict, Cooperative, Emotional, Hidden) (Spilka, 1973). The Character Graph is generated simultaneously with the Event Graph using Prompt 13 in the Appendix, ensuring alignment between character relationships and plot events.

3.3 Stage 2: Iterative Graph Refinement

Initial graphs generated by LLMs often exhibit structural issues or flat character arcs. To bridge the gap between a rough draft and a polished script, we introduce an iterative revision mechanism on narrative graphs that mimics the *Evaluate-Plan-Revise* cognitive process of expert writers (Flower and Hayes, 1981). The process alternates between a multi-agent critique module $\mathcal{C}$ and a graph refinement module $\mathcal{R}$. We define the set of critics as $\mathcal{C} = \{\mathcal{C}_{\text{theme}}, \mathcal{C}_{\text{char}}, \mathcal{C}_{\text{plot}}\}$, representing specialized agents that evaluate thematic consistency, character depth, and plot logic, respectively. The graph update at iteration t is formulated as:

$$G^{(t+1)} = \mathcal{R}\big(\mathcal{C}(G^{(t)}), G^{(t)}\big) \tag{2}$$

where $G^{(t)}$ denotes the graph state at step t, and $\mathcal{C}(G^{(t)})$ represents the aggregated feedback used by $\mathcal{R}$ to execute topological modifications.

The complete procedure is formalized in Algorithm 1.

Narrative Critics ($\mathcal{C}$). We deploy a suite of specialized agents that evaluate the narrative graph through a hierarchical critique process. Reflecting the fundamental principles of story craft (Mckee, 1997; Egri, 2007), the agents execute in a fixed sequence to ensure each narrative dimension builds upon a solid foundation:

1. **Theme Critic ($\mathcal{C}_{\text{theme}}$)** identifies instances where the storyline drifts from its core message or where the theme is merely stated through exposition rather than being "shown" through conflict and symbolism.

2. **Character Critic ($\mathcal{C}_{\text{char}}$)** builds upon thematic coherence to audit persona depth. It diagnoses flat development where characters lack growth, flags decisions missing clear internal or external motivation, and detects sudden attitude shifts that lack the necessary psychological buildup.

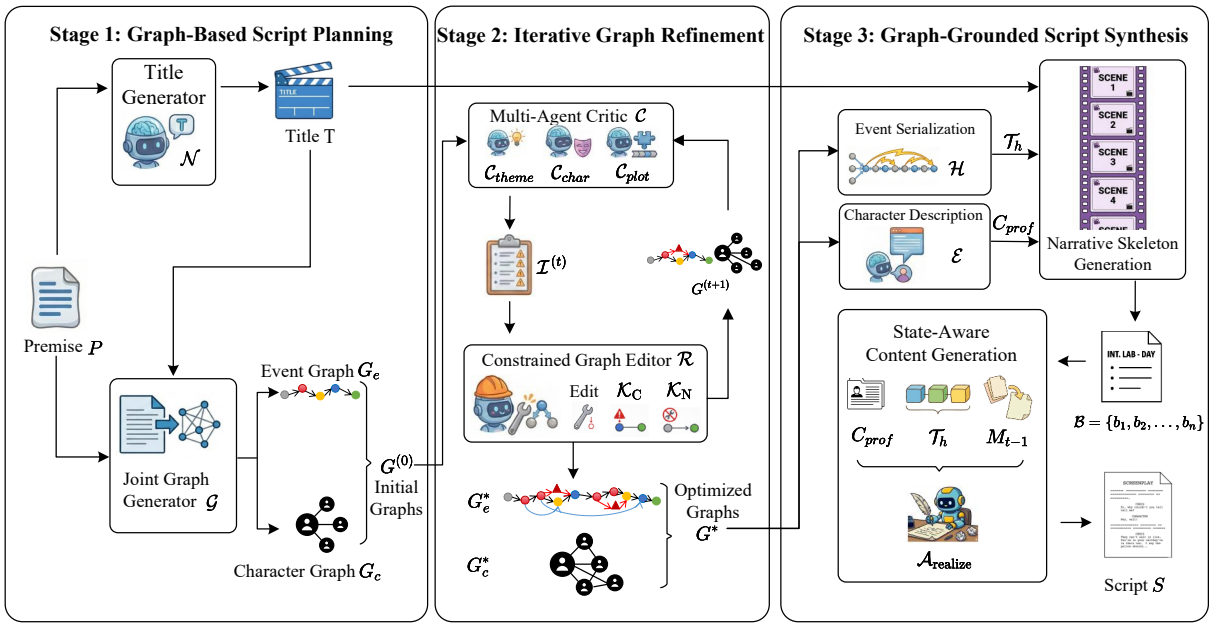

Figure 1: Overview of the **PLOTTER** framework. (1) **Graph-Based Script Planning** initializes the narrative backbone comprising Event (G_e) and Character (G_c) graphs. (2) **Iterative Graph Refinement** employs a Multi-Agent Critic ($\mathcal{C}$) to diagnose structural issues, which are resolved by a Constrained Graph Editor ($\mathcal{R}$) to produce an optimized graph (G^*). (3) **Graph-Grounded Script Synthesis** serializes the graph via Event Serialization ($\mathcal{H}$) and Persona Expansion ($\mathcal{E}$), conditioning a State-Aware Generator to synthesize the final script (S).

3. **Plot Critic ($\mathcal{C}_{\text{plot}}$)** integrates the preceding elements into a causally sound structure. It audits structural integrity by detecting causal discontinuities and logical contradictions. Furthermore, it ensures narrative engagement by flagging missing foreshadowing and monotone plotlines lacking turning points.

The analysis function of each agent $\mathcal{C}_i$ generates a structured issue list, formally denoted as $\mathcal{I}_i$. Cross-agent validation is further applied. Only edits that receive consistent support across agents proceed to execution, which limits the propagation of local reasoning errors. This list comprises five key components as defined in Table 5. The complete taxonomy and details of all issue types are presented in Table 7. The specific prompts used for issue identification are detailed in Prompts 18–20.

Constrained Graph Editor ($\mathcal{R}$). Upon receiving the issue list $\mathcal{I}$, the Editor $\mathcal{R}$ resolves structural deficiencies by mapping each issue $i \in \mathcal{I}$ to a sequence of atomic edit operations $\omega \in \Omega$ (e.g., `Add-Plot-Bridge`, `Revise-Event`; see Table 8 for full definitions). To ensure these modifications do not introduce new contradictions and mitigate error accumulation, the Editor operates under strict verification. For any proposed edit ω, the resulting graph must satisfy the core narrative constraints:

$$\text{Edit}(G, \omega) \models \mathcal{K}_C \wedge \mathcal{K}_N \tag{3}$$

The edits are grounded by two deterministic symbolic constraints:

- **Causal Rationality ($\mathcal{K}_C$):** We enforce that the causal subgraph must remain a Directed Acyclic Graph (DAG). This mathematically guarantees the forward flow of time and prevents logical loops, ensuring that every narrative event is preceded by a valid cause.

- **Narrative Completeness ($\mathcal{K}_N$):** We verify that every node remains reachable from the *Beginning* and maintains a path to the *Ending*. This ensures that all events are logically integrated into the main storyline and prevents the existence of isolated nodes that do not contribute to the global narrative arc.

If a modification violates either axiom, it is rejected, preserving the topological validity of the plot. The final optimized graph is denoted as $G^* = (G_e^*, G_c^*)$. Since the symbolic constraints ($\mathcal{K}_C$ and $\mathcal{K}_N$) are verified deterministically without LLM involvement, structurally invalid edits cannot propagate.

Algorithm 1 Hierarchical Iterative Graph Refinement

Input: Initial graph $G^{(0)}$, Critics $\mathcal{C}$, Editor $\mathcal{R}$, Max iterations K
Output: Optimized narrative graph G^*

$G^* \leftarrow G^{(0)}$
for $t = 1$ **to** K **do**
  Updated $\leftarrow$ **false**
  for each Agent $\mathcal{C}_i \in \mathcal{C}$ **do**
    $\mathcal{I}_i \leftarrow \mathcal{C}_i(G^*)$ ▷ Detect structural issues
    if $\mathcal{I}_i \neq \emptyset$ **then**
      $\omega \leftarrow \text{GenerateOps}(\mathcal{I}_i)$ ▷ Map issues to ω
      $G' \leftarrow \text{Edit}(G^*, \omega)$
      Updated $\leftarrow$ **true**
      if $G' \models \mathcal{K}_\text{C} \wedge \mathcal{K}_\text{N}$ **then**
        $G^* \leftarrow G'$ ▷ Accept valid edits
      end if
    end if
  end for
  if *Updated* $=$ **false then**
    break
  end if
end for
return G^*

3.4 Stage 3: Graph-Grounded Script Synthesis

In the final stage, we transform the optimized symbolic graphs G^* into a coherent textual script S. To bridge the gap between graph structures and linear text generation, we employ a strategy of *Graph Serialization* followed by *State-Aware Generation*.

3.4.1 Graph Serialization

Directly feeding raw graph definitions to an LLM often obfuscates narrative temporal dependencies. We therefore serialize symbolic graphs into a logically ordered textual sequence that preserves topology, so the generator can consume structural constraints without losing causal or relational context.

Event Serialization. We serialize the Event Graph G_e^* into a hierarchical event plan $\mathcal{T}_h = \mathcal{H}(G_e^*)$ via a deterministic depth-first traversal on the graph induced by causal and suspense relations. $\mathcal{T}_h$ specifies event-level progression and relation constraints. The traversal starts from events with no in-degree, visits each event once, and prioritizes suspense successors over causal successors when multiple successors are eligible; ties within the same relation type are resolved by the original chronological order in G_e^*. Foreshadowing edges are not included in the traversal and are instead retained as cross-event cues in the serialized output.

Character Description. Simultaneously, we expand the concise nodes in G_c^* into detailed character profiles $C_{prof} = \mathcal{E}(G_c^*)$ (see Prompt 14 in Appendix). This step fleshes out hidden backstories, providing semantic anchors for subsequent dialogue generation.

3.4.2 Progressive Script Generation

With the serialized events $\mathcal{T}_h$ and character profiles C_{prof}, we generate script components progressively. In this setup, $\mathcal{T}_h$ provides event-level structural constraints, while scene beats provide scene-level realization units.

Narrative Skeleton Generation. To ensure global consistency and thematic continuity before drafting detailed dialogue, we first generate a comprehensive sequence of scene beats $\mathcal{B}$ in a single pass (see Prompt 15 in Appendix), denoted as $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$. This generation process is formulated as:

$$\mathcal{B} = \text{LLM}(\mathcal{T}_h, C_\text{prof}, T) \tag{4}$$

Each generated beat b_i encapsulates a slugline, specific plot points, and pivotal character moments, strictly adhering to the causal and temporal dependencies established in the structural graph. Concretely, beats are generated conditionally on $\mathcal{T}_h$, so event structure is preserved while local scene content remains flexible.

State-Aware Content Generation. Finally, we flesh out each beat b_i into a full script scene σ_i through State-Aware Content Generation $\mathcal{A}_\text{realize}$. To maintain cross-scene coherence, the generator maintains a dynamic narrative state M_i. For each scene i, the generation is conditioned on:

- **Event Relation:** The relational context extracted from $\mathcal{T}_h$. By referencing the specific edge types (e.g., *Suspense* or *Conflict*) connected to the current event, the model aligns the scene's emotional tone and causal logic with the pre-optimized graph structure.
- **Character Persona:** The comprehensive profiles C_{prof} derived from the Character Graph. These serve as semantic anchors, ensuring that dialogue and actions remain consistent with the speaker's established voice and motivations.
- **Contextual Memory:** The rolling narrative state M_{i-1} that tracks the evolving history. M is updated iteratively to incorporate previous

dialogue and plot developments, ensuring referential continuity and preventing logical drift during long-range generation.

Formally, the transition from symbolic structure to script is performed by the state-aware realization function $\mathcal{A}_{\text{realize}}$:

$$\sigma_i = \mathcal{A}_{\text{realize}}(b_i, \mathcal{T}_h, C_{prof}, M_{i-1}) \quad (5)$$

where $M_0 = \emptyset$. The final script $\mathcal{S}$ is produced by the ordered concatenation of all realized scenes: $\mathcal{S} = \bigcup_{i=1}^{n} \sigma_i$.

Algorithm 2 State-Aware Content Generation

Input: Narrative Skeleton $\mathcal{B} = \{b_1, \ldots, b_n\}$, Serialized Events $\mathcal{T}_h$, Character Profiles C_{prof}
Output: Final Textual Script $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$
$M_0 \leftarrow \emptyset$ ▷ Initialize Contextual Memory
for $i = 1$ **to** n **do**
  ▷ *1. State-Aware Realization*
  $\sigma_i \leftarrow \mathcal{A}_{\text{realize}}(b_i, \mathcal{T}_h, C_{prof}, M_{i-1})$ ▷ Generate scene conditioned on history
  ▷ *2. Narrative State Tracking*
  $M_i \leftarrow \text{Summarize}(\sigma_1, \ldots, \sigma_i)$ ▷ Update M_i with latest developments
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\sigma_i\}$
end for
return $\mathcal{S} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$

4 Experiment

4.1 Dataset

To provide a rigorous evaluation with a broad coverage, we construct a dataset that balances diversity with complexity. While prior studies in narrative generation (Yang et al., 2022, 2023; Mirowski et al., 2023; Chen et al., 2024) typically use 20–60 LLM-generated premises as an evaluation dataset, we curate 50 premises from a hybrid mix of high-quality sources. Specifically, we sample from 3 human-curated datasets (MoPS (Ma et al., 2024), WritingPrompts (Fan et al., 2018), ROCStories (Mostafazadeh et al., 2016)) and 2 LLM-generated sources (DOC (Yang et al., 2023), GPT-4.1), as detailed in Appendix A.2. These premises span 9 genres (Sci-Fi, Drama, Crime, Thriller, Fantasy, Romance, Horror, General, and Other) with a mean Type–Token Ratio (TTR) of 0.80, indicating high lexical diversity across the evaluation set. This configuration provides a broader distribution than those used in most prior work and a more robust evaluation.

4.2 Baselines

We evaluate our method against three representative state-of-the-art script generation frameworks: **LLM-Plan-Write**, **Dramatron**, and **DOC**. Details of these baselines are provided in Appendix A.3.

To ensure a fair comparison, we implement all methods using 3 mainstream LLM backbones respectively: **GPT-4.1** (OpenAI, 2025), **DeepSeek-R1** (`deepseek-r1-250120`) (DeepSeek-AI et al., 2025), and **Qwen3** (`qwen3-235b-a22b`) (Yang et al., 2025). Premises are kept consistent across all methods and backbones. This multi-backbone setting ensures that the performance evaluations are not skewed by the inherent biases of a specific model family, nor by the self-preference bias caused by homologous generation and evaluation models.

4.3 Evaluation Method

Following prior work, we adopt a pairwise comparison using GPT-4.1 as the evaluator and carefully designed bias-mitigation strategies for LLM-based assessment, at both storyline and full-script levels (see Appendix A.4 for details). In addition, human evaluation corroborates the validity of these results. The evaluator selects the better script or reports a tie across five dimensions: Narrative, Thematic Expression, Characterization, Dramatic Engagement, and Premise Fidelity (see Table 24 for details). These dimensions are derived from classical dramatic theory (Mckee, 1997) and align with established metrics in mainstream script generation work (Yang et al., 2022, 2023; Mirowski et al., 2023; Chen et al., 2024).

4.4 Quantitative Results

LLM Evaluation. As shown in Table 1, PLOTTER consistently outperforms baselines across five evaluation dimensions. The most dominant gains appear in Narrative, Thematic Expression, Characterization, and Dramatic Engagement. These results directly validate the effectiveness of our Multi-Agent Critique and Constrained Graph Editor. Unlike baselines that rely on linear text generation, our method establishes a structurally valid narrative skeleton with the event graph before script text is produced, which improves long-range coherence and quality of the Narrative and Thematic Expression.

The performance gap highlights the fundamental difference between our graph-based reasoning

Backbone	Method	Narrative ↑		ThematicExpression ↑		Characterization ↑		DramaticEngagement ↑		PremiseFidelity ↑	
		Storyline	Script	Storyline	Script	Storyline	Script	Storyline	Script	Storyline	Script
GPT4.1	*LLM Plan and Write wins*	6	28	0	0	0	0	0	4	18	18
	Ours wins	**94**	**72**	**100**	**100**	**100**	**100**	**100**	**96**	**34**	**40**
	ties	0	0	0	0	0	0	0	0	48	42
	Dramatron wins	0	16	0	10	0	20	2	28	0	0
	Ours wins	**100**	**74**	**100**	**90**	**100**	**76**	**98**	**72**	2	**14**
	ties	0	1	0	0	0	4	0	0	98	86
	DOC wins	38	8	14	14	10	8	34	8	16	16
	Ours wins	**62**	**92**	**86**	**86**	**90**	**92**	**66**	**92**	10	**44**
	ties	0	0	0	0	0	0	0	0	74	40
DeepSeek R1	*LLM Plan and Write wins*	6	6	0	0	2	0	2	2	2	2
	Ours wins	**94**	**94**	**100**	**100**	**98**	**100**	**98**	**98**	**94**	**94**
	ties	0	0	0	0	0	0	0	0	4	4
	Dramatron wins	44	52	18	50	38	42	48	46	4	4
	Ours wins	42	48	**82**	**50**	**60**	**58**	**52**	**54**	4	**8**
	ties	14	0	0	0	2	0	0	0	92	88
	DOC wins	48	16	40	24	28	14	46	14	30	32
	Ours wins	**52**	**84**	**60**	**76**	**72**	**86**	**54**	**86**	18	**48**
	ties	0	0	0	0	0	0	0	0	52	20
Qwen3	*LLM Plan and Write wins*	14	12	0	0	0	0	8	4	12	14
	Ours wins	**86**	**88**	**100**	**100**	**100**	**100**	**92**	**96**	**66**	**68**
	ties	0	0	0	0	0	0	0	0	22	18
	Dramatron wins	32	36	8	16	8	30	28	40	8	2
	Ours wins	**64**	**64**	**92**	**84**	**92**	**70**	**70**	**60**	8	**16**
	ties	0	0	0	0	0	0	2	0	84	82
	DOC wins	30	22	26	24	20	20	28	22	34	26
	Ours wins	**70**	**78**	**74**	**76**	**80**	**80**	**72**	**78**	22	**60**
	ties	0	0	0	0	0	0	0	0	44	14

Table 1: Pairwise comparison of storyline and overall script between our method and baselines under five evaluation dimensions. Experiments are conducted using GPT-4.1, DeepSeek R1, and Qwen3 as backbone models. Results demonstrate the superiority of our method in narrative coherence, thematic engagement, and character development.

and existing methods. LLM Plan-and-Write and Dramatron suffer from a lack of structured refinement, and even DOC—which uses an iterative approach—is limited by a static, hierarchical text outline. By employing an *Evaluate-Plan-Revise* cycle that operates directly on graph nodes and edges, our method identifies and resolves character inconsistencies and dramatic lulls more effectively than text-based iteration. This results in significantly higher win rates in Characterization and Dramatic Engagement, proving that planning on a graph structure is inherently superior to editing traditional linear outlines for complex storytelling.

Human Evaluation. Beyond LLM-based evaluation, we further conducted a human study to verify the reliability of LLM evaluation. The results show strong inter-rater agreement (Fleiss' $\kappa = 0.688$) and alignment with the LLM evaluation (Cohen's $\kappa = 0.834$). (Details in Appendix A.5).

Objective Metrics. Due to the absence of a unified and comprehensive evaluation metric, existing methods widely rely on LLM and human evaluation. To mitigate potential length-caused confounds and bias, we report the statistical length of generated narratives and formula-based quantitative metrics for information density and redundancy in Table 2. All methods produce scripts of comparable length (∼10k words), confirming that our improvements are not attributable to length inflation. PLOTTER achieves the highest Distinct-2 and MATTR scores and the lowest Self-BLEU, indicating that its advantage stems from substantive content diversity rather than superficial lexical repetition. To further assess stability across narrative types, we additionally analyze cross-genre variation and edge-pattern consistency, showing that performance remains stable across the nine genres in our dataset (see Appendix A.7).

Method	**Words**	**Distinct-2↑**	**MATTR↑**	**Self-BLEU↓**
PLOTTER	9738±593	**0.793**	**82.6**	**0.017**
Dramatron	9792±775	0.778	81.4	0.022
DOC	10330±618	0.680	65.4	0.090
LLM-Plan-Write	9683±577	0.752	79.2	0.031

Table 2: Objective evaluation metrics.

4.5 Qualitative Results

To demonstrate how PLOTTER improves narrative structures, we present case studies on agent operations, end-to-end graph evolution, and comparative evaluation examples.

Agent Operations In the presented example, the protagonist, Elias, undergoes a psychological transition from defeat to counter-attack. The initial generation exhibited a severe logical violation: the narrative jumped directly from Scene 7 (The Defeat) to Scene 8 (The Climax) without sufficient build-up.

- **Issue 1: Motive-Weak.** As illustrated in Figure 5, the Character Critic flagged a Motive-Weak issue between Event 7 and Event 8, indicating an abrupt psychological transition. To repair this, the Constrained Graph Editor executed an Add-Plot-Bridge operation. By querying the *Event Graph* for context anchors, the system retrieved the "Missing People" motif from Scene 1 and generated a bridging node where Elias's internal monologue converts guilt into the resolve required for the climax.

- **Issue 2: Discontinuity.** A single bridging node is often insufficient for complex transitions. As shown in Figure 6, a Discontinuity was detected where Elias's sudden leadership felt unearned. To restore causal sufficiency, the editor applied a tri-partite Add-Plot-Bridge strategy ("Trinity of Action"): (1) Why: an internal node where Elias accepts his past arrogance. (2) Who: a social node where Elias reconciles with his partner Leilani. (3) How: a tactical node establishes the capability to mobilize the crowd. This multi-node insertion shows that resolving narrative discontinuity requires constructing robust causal chains rather than simple text smoothing.

Graph Evolution. We present a representative evolution from the initial to the final graghs in Section A.6. Figure 4 makes the edit process explicit by tracing how Critic diagnoses are resolved through concrete Editor operations.

Comparative Evaluation Examples. Table 11 presents a comparative case study of the generated content against the baseline, while Table 10 provides a comparison of their respective LLM evaluations. The evaluator prefers PLOTTER on Narrative, Thematic Expression, Characterization, and Dramatic Engagement.

We also provide a statistical analysis of cross-genre generalization in Appendix A.7. Performance remains stable across genres, and agentic operations remain consistent, indicating that the observed gains are not limited to a single representative scenario.

4.6 Ablation Study

To systematically verify the efficacy of **PLOTTER**, we conducted a comprehensive ablation study. We compare the Full Module against two categories of variants:

- **Ablated Variants** (`w/o Theme`, `w/o Character`, `w/o Plot`): The corresponding agent is deactivated to verify the necessity of each reasoning dimension.

- **Isolated Variants** (`Theme Only`, `Character Only`, `Plot Only`): Only one dimension agent is active, prohibiting cross-dimensional feedback, to test the synergy of the modules in improving the narrative graph.

Module Necessity. As illustrated in Figure 2, each cell reports the win rate of the full PLOTTER against an ablated variant. The full module consistently outperforms all ablated variants across all evaluation dimensions. Removing the Character or Plot module yields the largest performance collapse, while removing the Theme module shows comparatively smaller overall drop but a clear disadvantage in thematic quality. It shows that the collaborative reasoning between character psychology and plot structure is essential for achieving high-quality character development and emotional resonance. This confirms that every reasoning agent within the narrative graph is critical for the holistic quality of the generated script.

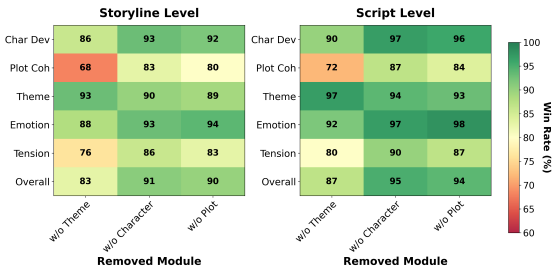

Figure 2: **Module Necessity Analysis.** Each cell shows the win rate (%) of the full model against the corresponding ablated variant. Every reasoning agent within the narrative graph is critical for the holistic quality.

The "$1+1>2$" Synergy Effect. To verify inter-module coordination, we compared the Full Module against single-agent variants (Character, Plot, or Theme) in Stage 2 by measuring their respective win rates against the *w/o Stage 2* baseline.

As shown in Figure 3, single-module variants yield only marginal improvements relative to *w/o Stage 2*, whereas the full model achieves a dominant advantage (win rate $>80\%$). Correspondingly, the full model yields sizable synergy gaps of +29%

at the Storyline level and +34% at the Script level over the average win rate of single-module variants. This massive jump significantly exceeds the linear sum of individual gains, indicating that joint optimization across modules is substantially more effective than activating any single module in isolation.

This synergy stems from the Stage 2 refinement process which orchestrates multiple agents to collaboratively edit and optimize the entangled Event and Character graphs. Rather than operating in silos, the agents provide mutual constraints: the Character agent motivates the Plot, while the Theme agent ensures global direction. For example, without this coordination, the Plot might satisfy causal logic but violate personas (e.g., a cowardly character acting heroically), whereas without the Plot agent, character arcs lack structural opportunities to evolve.

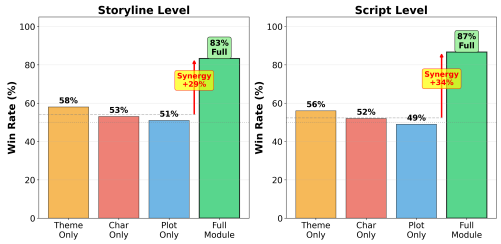

Figure 3: **Synergy Effect Analysis.** The incremental gains from individual modules sum to significantly less than the total performance improvement, indicating the synergy effect of modules during Stage 2 narrative graph refinement.

Sensitivity Analysis of Iteration Count K. Table 3 reports how the maximum number of refinement iterations affects quality and generation scale (GPT-4.1 as backbone). Diversity (Distinct-2) and coherence (Self-BLEU) both peak at K=3, while Edit-SR remains above 0.9 for $K \leq 3$. This supports using a moderate refinement depth to balance quality and controllability; detailed cost–latency trade-offs are discussed in Section 4.7. We therefore use K=3 as default (typically S≈28), and keep K=1 as a budget-oriented alternative (typically S≈16).

4.7 Computational Cost Analysis

Given that our method involves multiple LLM agents and an iterative cycle, we provide a per-script computational cost breakdown to verify the feasibility of practical deployment in Table 4. At the default setting (K=3, GPT-4.1), PLOTTER costs 1.68 USD per script (523k tokens, 13.6 min), suggesting a controllable computational burden while providing superior narrative control. Notably, Stage 2 (Iterative Refinement) accounts for only 32.4% of the total cost. For budget-constrained scenarios, setting K=1 further reduces cost to 0.36 USD per script (135k tokens, 6.4 min).

K	S	Edit-SR	Tokens	Time	Distinct-2↑	Self-BLEU↓
1	15.8	0.96	135k	6.4m	0.670	0.025
2	20.2	0.92	312k	10.3m	0.710	0.020
3	**27.6**	**0.91**	**523k**	**13.6m**	**0.793**	**0.017**
4	33.8	0.87	913k	21.6m	0.730	0.024
5	38.4	0.83	1257k	26.1m	0.640	0.035

Table 3: Sensitivity analysis of iteration count K. S: mean scenes generated. Edit-SR: issue-level success rate that passes the post-edit constraint validation. Bold: default settings.

Metric	**Value**
Total cost per script	1.68 USD
Total tokens per script	523k
API calls per script	171
End-to-end time	13.6 min
Stage 1 (Graph Planning)	4.9%
Stage 2 (Iterative Graph Refinement)	32.4%
Stage 3 (Script Synthesis)	62.7%
Budget mode (K=1) cost	0.36 USD
Budget mode time	6.4 min

Table 4: Computational cost breakdown.

5 Conclusion

This paper presents **PLOTTER**, a framework that enhances complex narrative generation by planning on graph-based representations. PLOTTER optimizes the causality and narrative skeleton by diagnosing and repairing structural issues of the graph topology under rigorous logical constraints. Experiments across diverse scenarios demonstrate that PLOTTER significantly outperforms strong baselines on both storyline logic and full-script quality. Ablation studies also reveal a synergy effect, indicating that the co-optimization of plot progression and character psychology yields coherence gains that exceed the sum of isolated improvements. These findings show that planning narratives on structural graph representations rather than directly on text is crucial to enhance the long context reasoning in complex narrative generation.

Limitations

Although our PLOTTER demonstrates significant improvements in complex narrative generation, there are limitations for future exploration. First, the diversity and scale of the evaluation could be further expanded. Second, the iterations of the *Evaluate-Plan-Revise* cycle could be further optimized for inference efficiency.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants 62306319 and sponsored by CCF-Tencent Rhino-Bird Open Research Fund.

References

Minwook Bae and Hyounghun Kim. 2024. 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. Association for Computational Linguistics.

Roland Barthes and Richard K. Miller. 1970. S/z: An essay.

Jing Chen, Xinyu Zhu, Cheng Yang, Chufan Shi, Yadong Xi, Yuxiang Zhang, Junjie Wang, Jiashu Pu, Tian Feng, Yujiu Yang, and Rongsheng Zhang. 2024. HoLLMwood: Unleashing the creativity of large language models in screenwriting via role playing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8075–8121, Miami, Florida, USA. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Lajos Egri. 2007. The art of dramatic writing.

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.

Linda S. Flower and J. R. Hayes. 1981. A cognitive process theory of writing. *College Composition & Communication*.

Chao Guo, Yue Lu, Yong Dou, and Fei-Yue Wang. 2023. Can chatgpt boost artistic creation: The need of imaginative intelligence for parallel art. *IEEE/CAA Journal of Automatica Sinica*, 10(4):835–838.

Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and Jiliang Tang. 2025. Reasoning with graphs: Structuring implicit knowledge to enhance LLMs reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25698–25714, Vienna, Austria. Association for Computational Linguistics.

Senyu Han, Lu Chen, Li-Min Lin, Zhengshan Xu, and Kai Yu. 2024. IBSEN: Director-actor agent collaboration for controllable and interactive drama script generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1607–1619, Bangkok, Thailand. Association for Computational Linguistics.

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 *International Conference on Representation Learning*, volume 2025, pages 5150–5183.

Zefeng Lin, Yi Xiao, Zhiqiang Mo, Qifan Zhang, Jie Wang, Jiayang Chen, Jiajing Zhang, Hui Zhang, Zhengyi Liu, Xianyong Fang, and Xiaohua Xu. 2025. R2: A llm based novel-to-screenplay generation framework with causal plot graphs. *ArXiv*, abs/2503.15655.

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.

Guillermo Marco, Julio Gonzalo, M.Teresa Mateo-Girona, and Ramón Del Castillo Santos. 2024. Pron vs prompt: Can large language models already challenge a world-class fiction author at creative text writing? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19654–19670, Miami, Florida, USA. Association for Computational Linguistics.

Robert Mckee. 1997. Story: Substance, structure, style, and the principles of screenwriting.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Franco Moretti. 2011. Network theory, plot analysis.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation 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. Association for Computational Linguistics.

OpenAI. 2025. Gpt-4.1 system card. https://openai.com/index/gpt-4-1/. Accessed: 2026-01-04.

Alexander Spangher, Nanyun Peng, Sebastian Gehrmann, and Mark Dredze. 2024. Do LLMs plan like human writers? comparing journalist coverage of press releases with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21814–21828, Miami, Florida, USA. Association for Computational Linguistics.

Mark Spilka. 1973. Henry james and walter besant: "the art of fiction" controversy. *Novel: A Forum on Fiction*, 6:101.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Representation Learning*, volume 2024, pages 3868–3898.

Yidan Sun, Qin Chao, and Boyang Li. 2023. Event causality is key to computational story understanding. *arXiv preprint arXiv:2311.09648*.

Tom Trabasso and Paul van den Broek. 1985. Causal thinking and the representation of narrative events. *Journal of Memory and Language*, 24:612–630.

Qianyue Wang, Jinwu Hu, Zhengping Li, Yufeng Wang, Daiyuan Li, Yu Hu, and Mingkui Tan. 2025a. 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.

Wenqing Wang, Mingqi Gao, Xinyu Hu, and Xiaojun Wan. 2025b. Towards a "novel" benchmark: Evaluating literary fiction with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21648–21673, Vienna, Austria. Association for Computational Linguistics.

Yichen Wang, Kevin Yang, Xiaoming Liu, and Dan Klein. 2023. Improving pacing in long-form story planning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10788–10845, Singapore. Association for Computational Linguistics.

Haotian Xia, Hao Peng, Yunjia Qi, Bin Xu, Juanzi Li, Hou Lei, and Xiaozhi Wang. 2025. Storywriter: A multi-agent framework for long story generation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 6559–6563, New York, NY, USA. Association for Computing Machinery.

Wenda Xie, Chao Guo, Yanqing Jing. Junle Wang, Yisheng Lv, and Fei-Yue Wang. 2026. Plug-and-play dramaturge: A divide-and-conquer approach for iterative narrative script refinement via collaborative llm agents. *Preprint*, arXiv:2510.05188.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinliang Frederick Zhang, Nick Beauchamp, and Lu Wang. 2024. Narrative-of-thought: Improving temporal reasoning of large language models via recounted narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16507–16530, Miami, Florida, USA. Association for Computational Linguistics.

A Appendix

A.1 Definition of Structured Issues

The analysis function of each agent returns a structured list of issues. To facilitate the graph modification process, each issue is strictly defined with five components, as shown in Table 5.

Component	Description
Type	The specific problem category (refer to Table 7 for the full taxonomy).
Description	Detailed identification and explanation of the issue.
Suggestion	Concrete optimization guidance provided by the agent.
Modification	Required graph operations (e.g., add node, update relation).
Targets	The specific graph elements (Nodes/Relations) involved.

Table 5: Structure of the issues generated by the Narrative Critics.

A.2 Dataset Details

We sample 50 premises from the following five sources:

- **MoPS**(Ma et al., 2024): A curated collection of movie plot summaries and narrative premises covering diverse genres.
- **WritingPrompts**(Fan et al., 2018): A dataset of user-submitted creative writing prompts designed to stimulate imaginative narrative composition.
- **ROCStories**(Mostafazadeh et al., 2016): A corpus of commonsense five-sentence stories that follow causal and temporal logic.
- **DOC**: A selection of narrative premises provided in the original DOC paper (Yang et al., 2023), intended for planning-based narrative generation.
- **LLM generated**: A set of creative premises synthesized using GPT-4.1 across diverse genres and thematic settings.

A.3 Baseline Descriptions

We compare our method against 3 representative open-source state-of-the-art baselines in script generation:

- **LLM-Plan-Write**: The premise is directly input into the LLM to produce a complete script through narrative planning and writing.
- **Dramatron**: A hierarchical generation method that sequentially generates beats, scenes, and detailed content such as dialogue.
- **DOC**: A planning-based framework that first creates a structured outline and then generates scripts for each part of the outline.

A.4 Evaluation Details

For each pair of scripts generated from the same premise by two different methods, we conduct pairwise comparisons at two levels: (1) by comparing the summaries of corresponding narratives (see Prompt 22), and (2) by evaluating the overall quality of the full scripts (see Prompt 23).

Counterbalancing Design. To ensure that the presentation order does not bias the LLM evaluator's judgment, we randomly split each comparison pair into two groups: one group presents scripts in the order A-B, while the other group presents them in the reverse order B-A, where A and B represent the two methods being compared (i.e., **PLOTTER** versus one of the baseline methods: LLM-Plan-Write, Dramatron, or DOC). This counterbalancing design ensures that any potential position bias is evenly distributed across both methods, thereby eliminating order effects on the evaluation results.

Evaluation Dimensions. The evaluator assesses each comparison along five dimensions: (1) **Narrative**—plot continuity, logical consistency, dramatic arc; (2) **Thematic Expression**—theme clarity, depth, symbolic reinforcement; (3) **Characterization**—motivation credibility, psychological depth, character growth; (4) **Dramatic Engagement**—suspense, turning points, tension management; and (5) **Premise Fidelity**—adherence to original premise.

A.5 Human Evaluation Statistics

We recruited five professional screenwriters to conduct a human evaluation, aiming to assess the reliability of the LLM evaluation. The final human verdict was determined by majority voting. Table 6 presents the inter-rater agreement among human evaluators and the alignment statistics between the machine evaluator and human judgments.

Metric	Value
Inter-Rater Agreement (Human-Human)	
Fleiss' Kappa	0.688
Machine-Human Alignment	
Overall Agreement	90.2%
Cohen's Kappa	0.834

Table 6: Statistics of human evaluation and machine-human consistency.

The high overall agreement (90.2%) and Cohen's Kappa (0.834) confirm that the automated metrics

used in our main experiments serve as a reliable proxy for human judgment.

A.6 Case study: End-to-End Graph Evolution

To concretely demonstrate how the EVALUATE–PLAN–REVISE cycle transforms a narrative, we present a localized walkthrough snapshot generated by PLOTTER ($K=1$), highlighting representative edits rather than the full intermediate process.

Figure 4 provides a simplified local snapshot of Event Graph evolution, showing how the Critic identifies structural issues (orange callouts) and the Editor resolves them via atomic operations (green callouts). The figure is intentionally partial and condensed; the complete event inventory is provided in Table 9.

A.7 Cross-Genre Generalization

To verify robustness across narrative genres, we analyzed performance stability over the 9 genres in our dataset. The average coefficient of variation (CV) across Distinct-2, MATTR, and Self-BLEU is 0.058, indicating highly stable performance. Furthermore, Suspense edges appear in 96% of generated scripts and Foreshadowing edges in 100%, confirming that the graph schema generalizes functionally across genres. Adapting the framework to alternative cultural narrative logics is left for future work.

Dimension	Issue Type	Description
Theme	`Theme-Direct`	Theme conveyed only via exposition; lacks symbolic or conflict-driven delivery
	`Theme-Vague`	No clear central storyline; events fail to revolve around a core message
Character	`Arc-Abrupt`	Sudden attitude shift without psychological build-up
	`Motive-Weak`	Key decisions missing internal/external motivation
	`One-Dimensional`	Characters show no conflicting traits or growth
Plot	`Discontinuity`	Adjacent events lack causal linkage or smooth transition
	`No-Suspense`	All information revealed too early; no unanswered questions
	`No-Foreshadow`	Later twists lack early symbolic hints or setups
	`No-Turning-Point`	Monotone storyline without rhythm-breaking reversals
	`Relation-Inconsistent`	Character relations or event logic contradict earlier setup

Table 7: Issue types identified by MULTI-AGENT CRITIQUE. Each issue is anchored to specific node(s) or edge(s) where the problem occurs.

Operation	Action	Purpose
`Add-Plot-Bridge`	Insert intermediate event node	Repair logical gaps between consecutive events; address discontinuities
`Add-Suspense`	Insert mystery/misleading clue	Enhance narrative tension; address low tension and lack of suspense
`Add-Foreshadow`	Embed symbolic detail or hint	Prepare for later narrative payoff; resolve lack of foreshadowing
`Insert-Twist`	Add unexpected but logical reversal	Introduce narrative rhythm change; address monotony
`Revise-Event`	Modify existing node/relation	Harmonize motivation chains; strengthen thematic consistency

Table 8: Atomic edit operations executed by CONSTRAINED GRAPH EDITOR. Edit plans may combine multiple operations while satisfying scope and causality constraints.

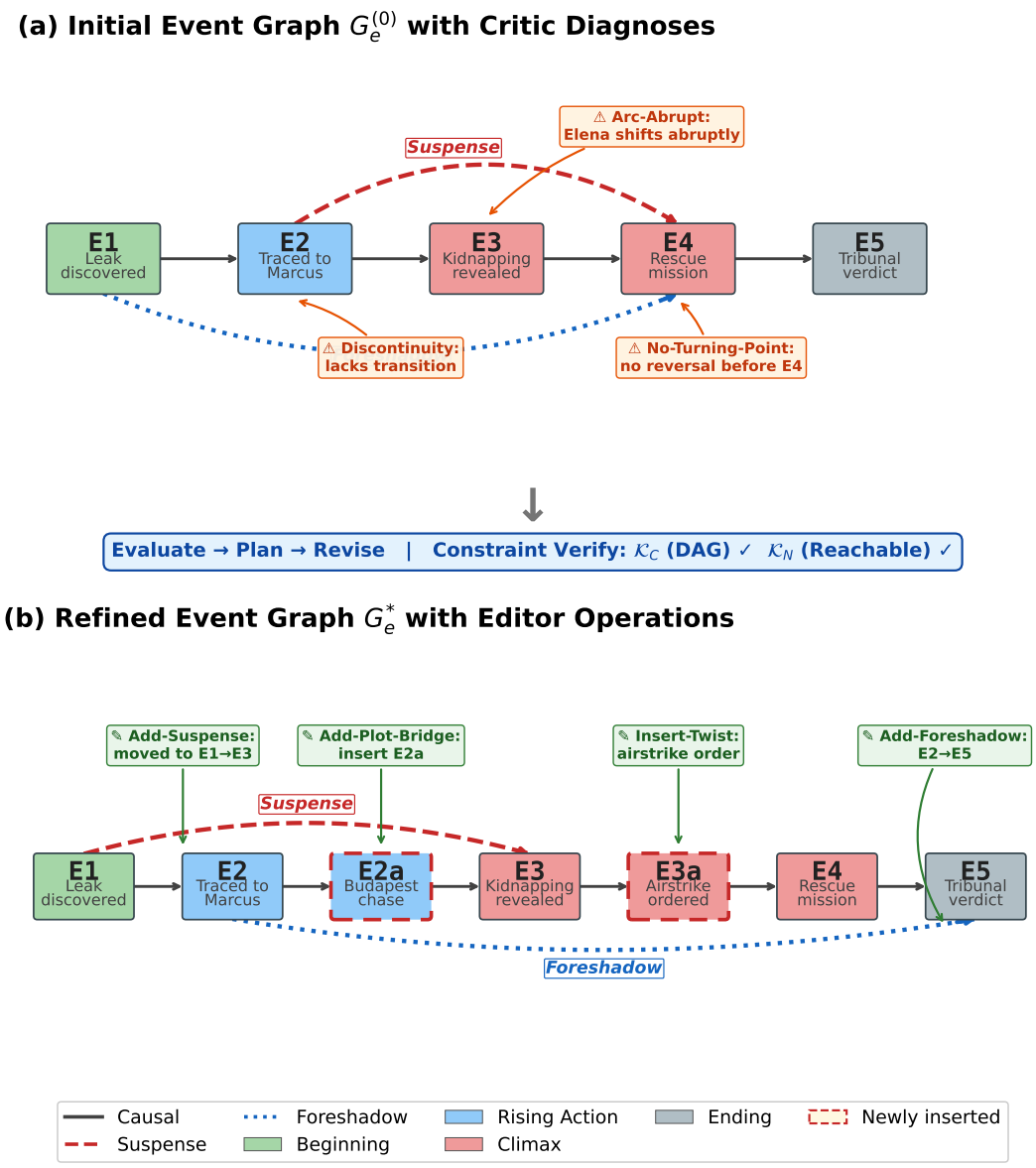

Figure 4: **Event Graph Evolution (simplified view).** (a) Initial graph $G_e^{(0)}$ with Critic diagnoses: `Discontinuity` (E2→E3 lacks transition), `Arc-Abrupt` (Elena shifts abruptly), `No-Turning-Point` (no reversal before E4). (b) Refined graph G_e^* with Editor operations: `Add-Plot-Bridge` (E2a inserted), `Insert-Twist` (E3a airstrike), `Add-Suspense` (E1→E3), `Add-Foreshadow` (E2→E5). Dashed red borders indicate newly inserted nodes. The green box confirms both constraints ($\mathcal{K}_C$ DAG, $\mathcal{K}_N$ Reachable) are satisfied. See Table 9 for the full 16-event inventory.

ID	Stage	Time	Description	Edit Operation
Original Events (from $G_e^{(0)}$)				
A3K	Beginning	Day 1	Elena and Marcus complete a Prague extraction; Elena intercepts an encrypted leak from their safe house to enemy syndicate Aegis.	—
F8W	Rising Action	Day 3	Analyst Nadia traces the leak to Marcus's device; a two-year pattern of compromised missions emerges.	—
P2R	Rising Action	Day 5	Vienna bait-trap confirms Marcus as the mole. He vanishes, leaving a note: "They have her. Forgive me."	—
J6T	Climax	Day 8	Budapest confrontation at gunpoint. Marcus reveals Aegis kidnapped his daughter Lily two years ago. Elena hesitates; he escapes.	*Modified*: deepened Elena's internal conflict
M4V	Falling Action	Day 10	Elena files an incomplete report omitting Lily. Graves assigns Kessler to shadow her; Nadia warns Graves may have known.	—
H9B	Rising Action	Day 13	Via dead-drops, Marcus reveals Lily's Carpathian location. Elena agrees to help if he surrenders afterward.	—
Q1X	Rising Action	Day 15	Graves orders an airstrike despite Lily being inside. Elena strikes Kessler and defects from headquarters.	*Modified*: added Elena's one-word refusal
D5Z	Climax	Day 16	Night assault on the compound. Elena kills Viktor; Marcus reunites with Lily; they escape before the airstrike.	—
W7G	Falling Action	Day 19	Elena exposes Graves's secret Aegis dealings using gathered intelligence. Graves is arrested for treason.	—
E2N	Ending	Day 25	Disgraced Elena testifies at Marcus's tribunal. Reduced sentence; she is dismissed but promises to watch over Lily.	*Modified*: expanded with legal consequences
Newly Inserted Events (by Stage 2 Constrained Graph Editor)				
K7P	Falling Action	Day 11	Elena breaks into archives and finds Graves's memo labeling Lily "acceptable operational attrition" and his override of Marcus's psych evaluation.	`Add-Plot-Bridge`: *Discontinuity* (M4V→H9B)
T3M	Rising Action	Day 14	Night before rescue. Marcus shows Lily's crayon drawings and birthday video. Elena's institutional loyalty dissolves.	`Add-Plot-Bridge`: *Arc-Abrupt* (Elena)
U2F	Rising Action	Day 15^{am}	Elena returns to HQ for satellite data; Graves confronts her with intercepted dead-drop transcripts. Her cover collapses.	`Add-Plot-Bridge`: *Discontinuity* (H9B→Q1X)
V4Q	Rising Action	Day 15^{pm}	Minutes after fleeing HQ, Elena's hands shake. Lily's photo on her phone sustains her—she drives forward despite terror.	`Revise-Event`: *One-Dimensional* (Elena)
R3Y	Rising Action	Day 15^{night}	En route, Elena passes through ruins of an agency-destroyed village. A child's shoe on a threshold crystallizes her resolve.	`Add-Foreshadow`: *No-Foreshadow* (D5Z)
X3L	Falling Action	Day 21	Prosecutor offers a deal: blame Marcus, get reinstated. Elena refuses, choosing to testify as a disgraced agent with no leverage.	`Add-Plot-Bridge`: *Motive-Weak* (W7G→E2N)

Table 9: **Event Graph: Initial → Refined.** Top: 10 original events from $G_e^{(0)}$. Bottom (shaded): 6 events inserted by the Editor during Stage 2, each annotated with the Critic issue type that triggered the insertion and the specific gap it bridges.

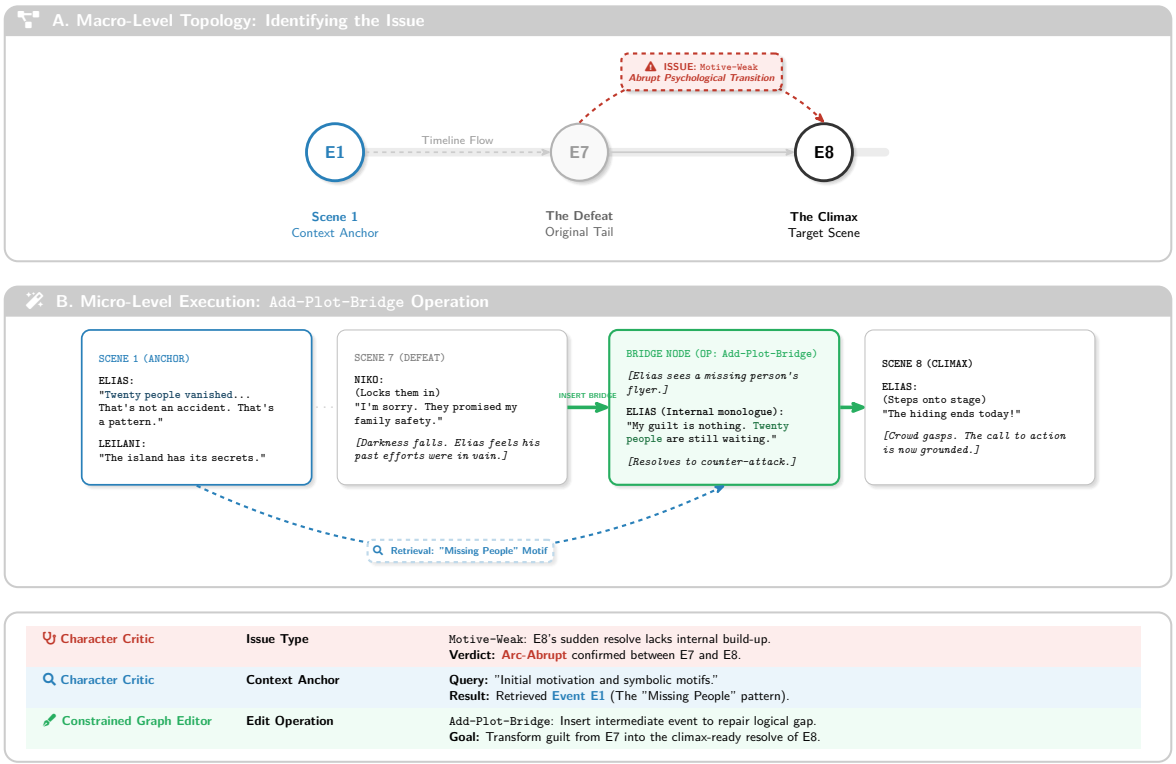

Figure 5: **Context-Aware Diagnosis and Retrieval.** (A) **Macro-Level Topology**: The MULTI AGENT CRITIC identifies a logical break between the defeat in E7 and the confidence in E8, flagging a `Motivation Gap` (see Issue Types in Table 7). (B) **Micro-Level Execution**: The CONSTRAINED GRAPH EDITOR performs a historical query to retrieve the "Missing People" motif, executing a context-aware `Add-Plot-Bridge` operation (Table 8). This allows the system to generate a specific bridging scene where Elias converts fear into anger.

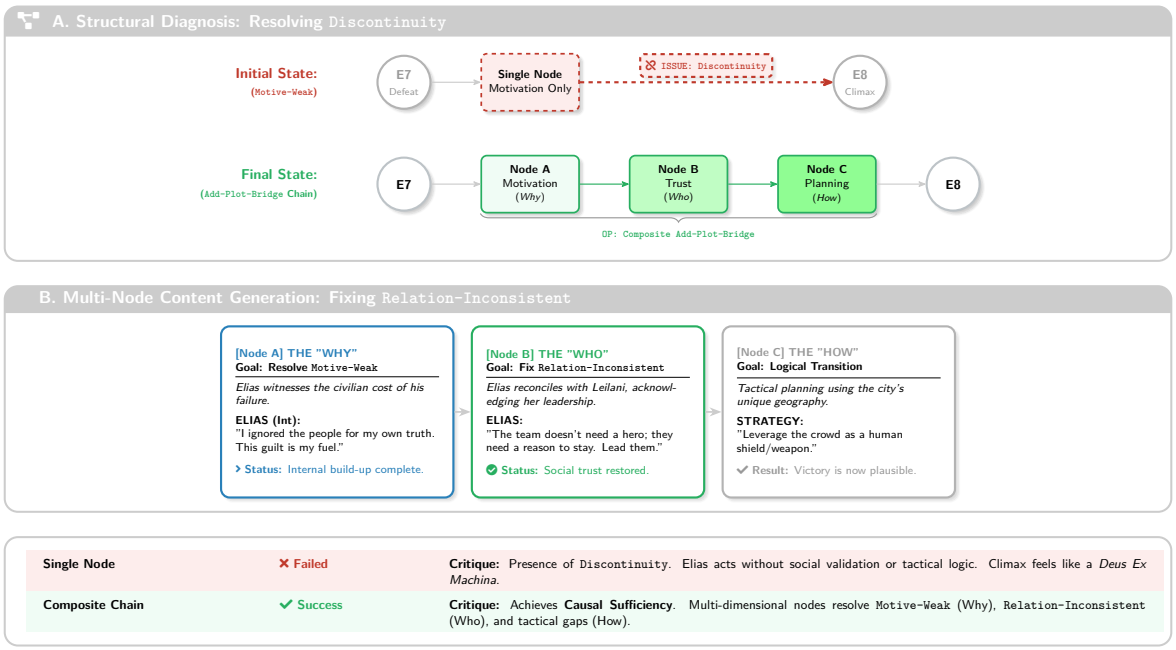

Figure 6: **From Single Node to Causal Chain.** (A) **Structural Diagnosis**: A simple single-node insertion (Ins-A) fails because it only addresses internal feelings. (B) **The "Trinity" of Action**: The Full Module generates a multi-hop causal chain by iteratively applying `Add-Plot-Bridge` operations: *Motivation* (Ins-A) $\rightarrow$ *Trust* (Ins-B) $\rightarrow$ *Planning* (Ins-C). This ensures the victory in E8 is logically earned.

Dimension	Verdict	Evaluator Explanation
Narrative	**B**	Script B demonstrates a more intricate and logically consistent narrative structure, with clear exposition, rising action, multiple points of conflict, and resolution, as well as smoother transitions and deeper world-building, whereas Script A, while coherent, follows a more straightforward and less nuanced dramatic arc.
Thematic Expression	**B**	Script B demonstrates a deeper and more nuanced exploration of its central theme of redemption and trust, employing recurring metaphors (such as the weathered whistle, anonymous notes, and symbolic gestures like offering the cap or playbook) and moments of self-reflection to reinforce the theme, while Script A, though consistent, approaches its theme more directly and with less subtlety or symbolic layering.
Characterization	**B**	Script B demonstrates greater emotional and psychological depth by exploring Cameron's internal struggles, the impact of his past on his present relationships, and the nuanced, gradual development of trust between him and Elijah, resulting in a more complex and believable character arc with clear growth for both father and son.
Dramatic Engagement	**B**	Script B demonstrates a higher level of dramatic engagement by introducing suspenseful elements such as anonymous notes and rumors, multiple significant turning points, and a more intricate buildup and release of tension through both personal and team-related stakes, whereas Script A, while emotionally resonant, follows a more straightforward and predictable arc with fewer suspenseful twists.
Premise Fidelity	**Same**	Both scripts faithfully adhere to the core premise, retaining the primary setting (high school football), central characters (Cameron, Elijah, and the mother), and the thematic direction of a former NFL player grappling with fatherhood and coaching; neither script significantly deviates from or adds extraneous elements that undermine the stated premise.

Table 10: GPT-4.1 pairwise evaluation results for a representative case. Premise: *"A former NFL player grapples with fatherhood and coaching his son in high school football."* Script A corresponds to the baseline (Dramatron), while Script B corresponds to **PLOTTER**. Blue cells indicate Script B superiority; gray indicates equal performance.

Narrative Function	Script A: "Second Chances"	Script B: "Second Down"
Opening & Discovery	**S1:** Cameron notices a quiet boy watching practice alone and learns, to his shock, that the boy is Elijah—his son.	**S1:** Cameron, newly retired from the NFL, returns as coach to his old high school. As he fingers a **weathered whistle** by the trophy case, he confronts his regret and hopes for redemption. **S2:** After practice, Cameron discovers an **anonymous note** in his locker referencing a past mistake and warning him that 'the past always catches up.' **S3:** Cameron notices Elijah watching practice from afar. Feeling a strange familiarity, he considers inviting Elijah closer with a team **cap**, but hesitates.
Confrontation & Truth Revealed	**S2:** Cameron confronts Elijah's mother, Jenna, demanding answers about why he never knew about his son, sparking anger and resentment. **S3:** Cameron debates whether to pursue a relationship with Elijah or respect Jenna's wishes to keep things as they are.	**S4:** Lena confronts Cameron, revealing that Elijah is his son. Cameron reels from the shock, overwhelmed as Lena insists on honesty. (The **cap** drops from his hand)
External Pressure & Crisis	**S4:** The principal questions Cameron's commitment to coaching after learning about his personal distractions. **S5:** Cameron's struggle to focus causes him to snap at a struggling player, damaging team morale.	**S5:** Cameron's attempt to connect with Elijah is rebuffed. The weight of the **anonymous note** and his guilt pushes Cameron to a breaking point. **S6:** The aftermath of Cameron's failures ripple through the team. Lena urges Cameron to truly commit to Elijah if he wants a real relationship.
Internal Struggle & Reflection	**S6:** Cameron tries to connect with Elijah over breakfast, but Elijah remains distant and untrusting. **S7:** Jenna accuses Cameron of trying to make up for the past too quickly, fueling guilt and self-doubt.	**S7:** Haunted by Lena's words, Cameron reflects on how easily trust can be broken, realizing that real growth demands vulnerability. **S8:** Moved by self-reflection, Cameron gently reaches out to Elijah, sharing personal stories from his troubled youth.
Complications & Setbacks	**S8:** Cameron learns that Elijah is being bullied for being the coach's 'secret' kid, escalating Cameron's inner conflict. **S9:** Cameron witnesses Elijah stand up for himself, revealing resilience but also anger toward Cameron. **S10:** Cameron is forced to choose between supporting his struggling team and comforting Elijah, ultimately failing both.	**S9:** After reconnecting with Elijah, Cameron overhears players discussing a possible sabotage for the playoff game. (Cameron's NFL **cap** is mentioned in context) **S10:** Elijah, conflicted by Cameron's efforts, confides in Lena about his doubts and hope. He cautiously decides to give Cameron a chance.
Tentative Reconciliation	**S11:** Elijah tentatively joins practice, and Cameron struggles to maintain professional boundaries while connecting as a father.	**S11:** Elijah takes a tentative step by joining the team as a waterboy. Cameron struggles to balance his roles. (Scene contains both **whistle** and **cap**)
Pre-Climax Preparation	**S12:** A key game arrives, and Cameron must decide between prioritizing victory and supporting Elijah emotionally. **S13:** During halftime, Cameron delivers a motivational speech that rallies the team but leaves Elijah feeling sidelined.	**S12:** Cameron, tested by Elijah's hesitation and the team's skepticism, commits to vulnerable leadership and transparency. **S13:** Before the critical game, Cameron and Elijah share a quiet moment. Cameron's gesture of solidarity begins to rebuild trust.
Climax & Key Decision	**S14:** After the game, Cameron apologizes publicly to Elijah, acknowledging his failures as a father. **S15:** Cameron and Elijah reconcile as the team wins. Cameron realizes that success is measured by relationships, not trophies.	**S14:** During the game, the team works together despite challenges. Cameron's consistent support of Elijah shifts the team dynamic. **S15:** When a key player is injured, Elijah steps up to help. Cameron faces a pivotal decision: trust Elijah with the **playbook** and his coaching strategy, symbolizing complete vulnerability and trust.
Extended Resolution	*[Script A ends at S15]*	**S16:** The team rallies around Elijah's contribution. Cameron and Elijah's relationship deepens through shared responsibility. **S17:** Post-game, Cameron and Elijah reflect on their journey. Cameron acknowledges his past while embracing his role as father. **S18:** The team celebrates their hard-fought victory. Cameron and Lena discuss co-parenting, establishing healthier boundaries. **S19:** In a quiet moment on the field, Cameron and Elijah toss the football. The **whistle**, **cap**, and **playbook** are all visible, symbolizing their completed journey from strangers to family.

Table 11: Narrative-function-aligned comparison for the representative case. Scenes are grouped by their dramatic role (e.g., opening, confrontation, climax). Script A uses 15 scenes total; Script B uses 19 scenes with finer-grained division within each narrative function. Recurring symbolic elements in Script B are highlighted in **bold**. The blue row shows Script B's extended resolution (4 additional scenes).

Script Title Generation Prompt:

Script Summary: {storyline}

Generate a **concise and distinctive** title for this script (no more than 7 words).

The title must reflect the **theme, genre, and narrative structure** of the script.

It should be **expressive, poetic, and meaningful**, in the spirit of classic film titles.

Additionally, the title should be **captivating and marketable**, suitable for **film posters or streaming platforms**.

Example Titles:

1. Summary: A war correspondent exposes a military cover-up and is hunted as a traitor by the government.

Generated Title: "Shadow in the Fire"

2. Summary: A scientist discovers time travel but realizes using it will erase his daughter from history.

Generated Title: "The Vanishing Hour"

3. Summary: In a dystopian city where dreaming is forbidden, a woman risks everything to protect the last dreamer.

Generated Title: "The Last Dream"

Script Summary: {storyline}

Return strictly in JSON format:

{"title": "Generated Title"}

Table 12: Prompt for generating a concise and distinctive title for a script.

Plot and Character Graph Creation Prompt:

script Title: {title}

Premise: {premise}

Based on this premise, generate a Event Graph and a Character Graph.

Event Graph:

- The plot must have a clear main storyline with a complete arc from the 'Beginning' to the 'Ending'.
- Each event node must have a clear causal relationship with the next, forming a tightly connected chain.
- The plot should contain around 10 events to maintain narrative focus and pacing.
- There must be two Climax nodes (with \texttt{narrative_stage = "Climax"}) positioned at two major narrative peaks.
- The event structure must strictly follow the sequence and count below (10 nodes in total):

1. Beginning (1)
2. First Rising Actions (2–3)
3. First Climax (4)
4. First Falling Action (5)
5. Second Rising Actions (6–7)
6. Second Climax (8)
7. Second Falling Action (9)
8. Ending (10)

- The 'Beginning' must be the first event node, and the 'Ending' must be the last.
- Each event must include the following narrative attributes:
- {narrative_stage}: One of the following:
- Beginning: Introduces time, setting, and main characters; shows initial status quo and a triggering problem. Only the first node.
- Rising Action: Characters try to solve the problem but face obstacles; tension and stakes increase.
- Climax: The most intense point of conflict or a major turning point. Must appear twice.
- Falling Action: Consequences unfold; tensions ease; relationships and power dynamics shift.
- Ending: Final state of the protagonist; answers the initial question and reflects growth. Only the last node.

- Each event must include:
- {id}: Unique identifier for the event
- {description}: A full description of the event, including character actions
- {narrative_stage}: One of the five stages defined above
- {time}: Time the event occurs (e.g., "Day 1")

Character Graph:

- Characters must have rich, multi-dimensional relationships, not just 'friend' or 'enemy'. Include:
- Conflict Relations (rivalry, revenge, misunderstanding)
- Cooperative Relations (ally, mentor-mentee, colleague)
- Emotional Relations (family, romantic, unspoken affection)
- Hidden Relations (secret identity, double agent, concealed hostility)
- Classify characters into:
- Main Characters: Drive the plot and make key decisions that shape the outcome.
- Supporting Characters: Expand the world and support the narrative through loyalty shifts, clue delivery, or creating conflict.
- All relationships should have logical narrative motivation—avoid random or unexplained links.
- The Character Graph must align with the Event Graph and ensure character presence is justified within events.

Example Output in JSON (Strictly follow this format):

```
{
 "plot_graph": {
 "events": [
 {"id": "E1", "description": "...", "narrative_stage": "Beginning", "time": "Day 1"},
 {"id": "E2", "description": "...", "narrative_stage": "Rising Action", "time": "Day 2"},
 ...
 ],
 "relations": [
 {"from": "E1", "to": "E2", "relation": "causal"},
 ...
 ]
 },
 "character_graph": {
 "characters": [
 {"id": "C1", "name": "Jack", "motivation": "..." },
 ...
 ],
 "relations": [
 {"from": "C1", "to": "C2", "relation": "..." },
 ...
 ]
 }
}
```

Table 13: Instructional prompt used to generate the Event Graph and Character Graph structures from a given script title and premise.

Character Creation Prompt:

You are generating characters for a script. Ensure that the characters fit the tone, style, and emotional depth of the script.

Guidelines for Character Creation:

- **Each character must have:**

1. **A core personality trait** (e.g., loyal, ambitious, paranoid, cynical).
2. **An internal conflict (fatal flaw)** (e.g., pride, guilt, fear, obsession).
3. **An external goal** (e.g., expose the truth, save a loved one, escape the past).

4. **A relationship with at least one other character** (mentor, enemy, lost love, etc.).
5. **A hidden past or secret** that impacts their choices.

Example:
Storyline: A scientist discovers time travel but realizes using it will erase his own daughter from existence.
Generated Characters:
{
"characters": {
"Dr. Monroe": {
"description": "A brilliant physicist (obsessive, morally torn), willing to break time itself to save his daughter. His guilt over a past mistake fuels his desperation."
},
"Evelyn Monroe": {
"description": "A rebellious teenager (curious, fearless), unaware that her father's invention is rewriting her existence."
},
"Agent Carter": {
"description": "A government enforcer (ruthless, efficient) who believes time travel is a weapon. He once saved Monroe's life, but now sees him as a threat."
}
}
}

Title: {title}
Storyline: {storyline}
Character Graph:{character_graph}
Strictly return JSON format:
{"characters": {"Character Name": {"description": "Short but rich character description"}}}

Table 14: Prompt used for generating characters in a script.

Scene Generation Prompt:
You are generating a **structured sequence of scenes** for a script.
Each scene must be **based on the events from the Event Graph** and take into account the **character relationships from the Character Graph**.

Script Information:
Title: {title}
Premise: {premise}
Event Graph:
{plot_description}
Character Graph:
{character_description}

Scene Generation Requirements:
- **Each scene must correspond to an event in the event graph**, following the same order.
- Ensure **clear causal connections between events** and maintain chronological order.
- The **protagonist and key characters must appear in relevant scenes**, consistent with the character graph.
- **Each scene must have a clear objective** and contribute to the advancement of the narrative.
- The number of generated scenes should match the number of events in the event graph.

Example:
Premise: In a futuristic city, a hacker uncovers a massive conspiracy after breaching the government system.
Generated scenes:
[

```
{
"place": "Neon-lit alley in the cyberpunk city",
"plot_element": "Inciting Incident",
"beat": "Hacker Logan discovers an encrypted file revealing the government's dark secrets."
},
{
"place": "Underground resistance base",
"plot_element": "Climax",
"beat": "Logan must decide whether to release the file, risking his life."
}
]

Strictly return a JSON list in the following format:
[
{"place": "Scene location", "plot_element": "Narrative function", "beat": "Key moment that drives the plot"}, ...
]
```

Table 15: Prompt for generating a structured sequence of scenes for a script.

Graph-Enhanced Scene Description Generation Prompt:

You are generating a **cinematic and immersive scene description** for a script that integrates narrative graph structure.

Scene Context:
- **Location**: {scene.place}
- **Plot Element**: {scene.plot_element}
- **Key Moment**: {scene.beat}
- **Scene Position in Narrative Arc**: Scene {scene_index + 1}

Event Graph Context:
{graph_context}

Scene Description Guidelines:
- **Describe the setting in a way that enhances the emotional tone** and reflects the narrative position in the graph structure.
- **Incorporate sensory details** that align with the causal and suspense relations from the narrative graph.
- **Show how characters interact with the environment** in ways that reflect their graph-encoded motivations and relationships.
- **Make sure the scene description hints at narrative tension** that respects the graph's suspense and foreshadowing edges.
- **Reference thematic elements** that emerge from the graph structure if appropriate.

Graph-Informed Guidelines:
- If this scene follows a suspense edge in the graph, build atmospheric tension through environmental description.
- If this scene follows a foreshadowing edge, include subtle visual elements that hint at future developments.
- If this scene is a causal continuation, ensure the setting logically follows from previous scenes.

Strictly return a JSON object:
{"scene_description": "A concise and vivid scene description that integrates graph structure."}

Table 16: Prompt for generating graph-enhanced scene descriptions that leverage narrative graph topology.

Multi-Issue Plot and Character Modification Prompt:

Current Event Graph:

{event_graph}

Current Character Graph:

{character_graph}

Below is a list of multiple issues that need to be addressed. Please generate a corresponding modification plan for **each individual issue**, based on its **type**, **description**, **suggested solution**, **modification method**, **involved nodes**, and **involved relations**:

Issue List:

{parsed_issues}

Please follow these modification constraints:

- **Scope Constraint**: Only modify nodes and relations explicitly listed in "involved nodes" and "involved relations" fields.
- **Method Constraint**: All modifications must follow the "modification method" field; no additional changes are permitted.
- **Node Insertion**: New nodes are only allowed when explicitly permitted by the modification method. New nodes must have unique IDs and maintain narrative stage consistency with adjacent nodes.
- **Relation Types**: Use appropriate relation types (causal, suspense, foreshadowing) based on the narrative function of the modification.

Example Output Format:

```
{
"issues": [
{
"issue_id": 1,
"plot_changes": {
"Delete relation": [{"from": "E2", "to": "E3"}],
"New event": [{"id": "E2a", "description": "...", "narrative_stage": "Rising Action", "time": "..."}],
"Modify event": [{"id": "E3", "description": "...", "narrative_stage": "...", "time": "..."}],
"New relation": [{"from": "E2", "to": "E2a", "relation": "causal"}, {"from": "E2a", "to": "E3", "relation": "causal"}]
},
"character_changes": {
"Delete relation": [{"from": "C1", "to": "C2"}],
"New character": [{"id": "C3", "name": "...", "motivation": "..."}],
"Modify character": [{"id": "C1", "name": "...", "motivation": "..."}],
"New relation": [{"from": "C1", "to": "C3", "relation": "..."}]
}
}
]
}
```

Table 17: Prompt for generating modification plans for plot and character issues.

Theme Agent Analysis Prompts:

The Theme Agent evaluates two dimensions: (1) theme clarity and (2) thematic expression explicitness.

1. Theme Missing Analysis:

You are a script theme structure analyst. Please evaluate the overall event graph to identify whether there is an issue of **"Unclear Theme or Storyline"**, and provide a suggestion to strengthen the thematic coherence by modifying the content of existing event nodes.

Problem Definition:

Unclear Theme or Storyline: The script lacks a clearly defined, consistently developed central theme or narrative thread. As a result, character actions and plot developments do not revolve around a central issue or narrative direction.

Optimization Guidelines:

- Do not add any new event nodes;
- Refine the content of existing nodes to reveal and emphasize an underlying thematic focus;
- Ensure that multiple key nodes reflect the same core idea through character dilemmas and decisions;
- Output only one issue.

2. Theme Explicitness Analysis:

You are an expert in optimizing thematic expression in screenwriting. Please analyze the following event graph to determine whether there is an issue of **"Overly Explicit Thematic Expression"**, and provide a structural optimization by inserting a narrative buildup node and adjusting causal relations to express the theme more implicitly.

Problem Definition:

Overly Explicit Thematic Expression: The theme is conveyed too directly through dialogue or narration, rather than being revealed progressively through character decisions, conflicts, and symbolic narrative elements.

Optimization Guidelines:

- Only analyze direct causal links in the relations field;
- Identify and remove overly abrupt or thematically obvious P → Z connections;
- Insert a buildup node Q between P and Z, where Q arises naturally from P and builds narrative tension;
- Output only one issue.

Table 18: Prompts used by the Theme Agent for analyzing theme-related issues in the event graph.

Character Agent Analysis Prompts:

The Character Agent examines three aspects: (1) character drive, (2) character flatness, and (3) character arc abruptness.

1. Character Drive Analysis:

You are an expert in analyzing character psychological motivation. Please examine the following event graph to determine whether there exists an issue of **"Lack of Internal Motivation and Setup in Character Development"**, and suggest a structural optimization by inserting a motivation-building event node and reconstructing the causal chain.

Problem Definition:

Lack of Internal Motivation and Setup: The character makes a significant decision or undergoes a major attitude shift at a key plot point, but the preceding event lacks sufficient emotional response, internal reflection, or critical external stimulus to justify the change.

2. Character Flatness Analysis:

You are a character complexity design expert. Please analyze the following event graph to identify whether there is an issue of **"One-Dimensional Characterization"**, and suggest a structural optimization by inserting a fluctuation node and adjusting the causal event structure.

Problem Definition:

One-Dimensional Characterization: The character consistently behaves in a single manner, lacking emotional fluctuation, personality contrast, or internal conflict.

3. Character Arc Analysis:

You are a narrative pacing expert specialized in character arc development. Please analyze the following event graph to identify whether there is an issue of **"Abrupt Character Arc Shift"**, and provide a structural optimization suggestion by inserting a mediating node into an existing causal chain to better support the psychological transition of the character.

Problem Definition:

Abrupt Character Arc Shift: A character undergoes a significant behavioral or emotional change at a key narrative point, but the preceding event lacks sufficient setup in terms of motivation, emotion, conflict, or external stimulus.

Table 19: Prompts used by the Character Agent for analyzing character-related issues in the event graph.

Plot Agent Analysis Prompts:

The Plot Agent audits five structural dimensions: (1) plot incoherence, (2) missing turning points, (3) lack of foreshadowing, (4) insufficient suspense, and (5) relation conflicts.

1. Plot Incoherence Analysis:

You are a narrative progression structure optimization expert. Please analyze the following event graph to identify whether there is an issue of **"Incoherent Plot Progression"**, and propose a structural optimization by inserting a progression node to improve narrative continuity.

Problem Definition:

Incoherent Plot Progression: Adjacent events lack clear logical transitions or causal links, resulting in abrupt or unnatural narrative development.

2. Missing Suspense Analysis:

You are a suspense design and narrative pacing expert. Please analyze the following event graph to determine whether it contains an issue of **"Lack of Suspense"**, and propose a structural optimization by inserting a suspense node to establish a cross-phase tension chain.

Problem Definition:

Lack of Suspense: The plot reveals too much information too clearly and completely, leaving no room for mystery or open questions.

3. Lack of Foreshadowing Analysis:

You are a narrative structure optimization expert. Please analyze the following event graph to determine whether there is an issue of **"Lack of Foreshadowing"**, and provide a structural enhancement suggestion by embedding symbolic behaviors or implicit references in earlier events.

Problem Definition:

Lack of Foreshadowing: The script lacks symbolic details, hidden clues, or behavioral hints deliberately planted in early stages, which leads to an emotionally flat or structurally disconnected climax.

4. Plot Turning Point Analysis:

You are a narrative rhythm and dramatic structure expert. Please analyze the following event graph to identify whether there is an issue of **"Lack of Plot Reversal"**, and propose a structural optimization by inserting a reversal node to enhance dramatic variation.

5. Relation Conflict Analysis:

You are an expert in analyzing logical consistency in script relationships. Please assess the overall event graph to identify whether there is an issue of **"Contradictions in Character or Plot Relationships"**, and provide a suggestion for improving logical coherence.

Table 20: Prompts used by the Plot Agent for analyzing plot-related issues in the event graph.

Constraint-Satisfaction Dialogue Generation Prompt:

```
You are generating high-quality cinematic dialogue that satisfies multiple narrative constraints.

Scene Information:
- Location: {scene.place}
- Plot Element: {scene.plot_element}
- Key Moment: {scene.beat}

Character Descriptions:
{json.dumps(characters, indent=2)}

Narrative Memory Context:
{memory_summary}

Character State Constraints:
{json.dumps(character_constraints, indent=2)}

Constraint Satisfaction Requirements:
1. Character Consistency Constraint: Each character's dialogue must align with their description
and current emotional state from memory.
2. Narrative Coherence Constraint: Dialogue must maintain logical flow with previous scenes via
memory context.
3. Character Arc Constraint: Dialogue should reflect character development indicated in the
narrative graph.
4. Emotional Continuity Constraint: Emotional trajectory from memory must be respected and advanced
appropriately.
5. Inter-scene Reference Constraint: Naturally incorporate references to previous scenes when
relevant.

Dialogue Generation Guidelines:
- Each character must speak at least once, reflecting their current state and constraints.
- Introduce at least one emotional shift that respects the character arc.
- Maintain logical coherence with previous dialogue and character arcs.
- Make each character's voice distinct and natural.
- Use emotion and subtext (hidden intentions, suppressed feelings) to add depth.
- Provide at least 8 dialogue turns (approx. 12-16 lines in total).
- Ensure all constraints are satisfied simultaneously.

Strictly return JSON format:
{
 "dialogue": [
 "Character1: (emotion, action) Line...",
 "Character2: (reaction, subtext) Response...",
 ...
 ]
}
```

Table 21: Prompt for generating dialogue with constraint satisfaction framework incorporating narrative memory and character state constraints.

Pairwise Script Evaluation Prompt (Storyline Comparison):

```
You are a professional script analyst. Please act as an impartial judge and evaluate the quality
of the two scripts generated by different methods.

Your evaluation should focus only on the following dimension:

Dimension - [Narrative/Thematic Expression/Characterization/Dramatic Engagement/Premise Fidelity]
```

Evaluate based on the criteria specified in Table 24.

Evaluation Guidelines:
1. Avoid position biases: the order of presentation should not influence your decision.
2. Ignore superficial factors: do not let length, formatting style, or surface polish bias your judgment.
3. Focus on content quality: base your reasoning strictly on the narrative quality under the specified dimension.
4. Comparative assessment: compare the two scripts directly on the given dimension.

Decision Criteria:
• Choose "A" if Script A demonstrates clearly superior performance on this dimension.
• Choose "B" if Script B demonstrates clearly superior performance on this dimension.
• Choose "Same" if both scripts are approximately equal in quality, or if neither shows a significant advantage.

Input Format:
[Script A]
Premise A: {premise_a}
Title A: {title_a}
Summary (Beats):
{beats_a}

[Script B]
Premise B: {premise_b}
Title B: {title_b}
Summary (Beats):
{beats_b}

Output Requirements:
1. Provide a concise, one-sentence explanation justifying your judgment.
2. Output your verdict in strict JSON format as specified below.

Required JSON Format:
{
 "explanation": "your explanation of which script is better and why",
 "verdict": "A" or "B" or "Same"
}

Table 22: Prompt template for pairwise comparison of storylines (beats only) between two scripts. The dimension placeholder is replaced with one of the five evaluation dimensions (see Table 24).

Pairwise Script Evaluation Prompt (Full Script Comparison):
You are a professional script analyst. Please act as an impartial judge and evaluate the quality of the two scripts generated by different methods.

Your evaluation should focus **only** on the following dimension:

Dimension – [Narrative/Thematic Expression/Characterization/Dramatic Engagement/Premise Fidelity]
Evaluate based on the criteria specified in Table 24.

Evaluation Guidelines:
1. Avoid position biases: the order of presentation should not influence your decision.
2. Ignore superficial factors: do not let length, formatting, or surface polish affect your judgment.
3. Focus on content quality: base your reasoning strictly on the narrative quality under the specified dimension.

4. Comparative assessment: compare the two scripts directly on the given dimension.

Decision Criteria:
• Choose "A" if Script A demonstrates clearly superior performance on this dimension.
• Choose "B" if Script B demonstrates clearly superior performance on this dimension.
• Choose "Same" if both scripts are approximately equal in quality, or if neither shows a significant advantage.

Input Format:
[Script A]
Title: {title_a}
Premise: {premise_a}
Full Script:
{scenes_a}
(Each scene includes: Place, Plot element, Beat, and Dialogue)

[Script B]
Title: {title_b}
Premise: {premise_b}
Full Script:
{scenes_b}
(Each scene includes: Place, Plot element, Beat, and Dialogue)

Output Requirements:
1. Provide a concise, one-sentence explanation justifying your judgment.
2. Output your verdict in strict JSON format as specified below.

Required JSON Format:
{
 "explanation": "your explanation of which script is better and why",
 "verdict": "A" or "B" or "Same"
}

Table 23: Prompt template for pairwise comparison of full scripts (including scenes and dialogue) between two scripts. The dimension placeholder is replaced with one of the five evaluation dimensions (see Table 24).

Evaluation Dimensions and Criteria:

1. Narrative
Assess the narrative quality based on the following criteria:
• **Plot Continuity**: Smooth transitions between events with clear causal linkages
• **Logical Consistency**: Coherent contextual setups, world-building, storyline progression, and character behaviors
• **Dramatic Structure**: Presence of a complete narrative arc (Exposition, Rising Action, Climax, Falling Action, Resolution)

2. Thematic Expression
Assess the thematic development based on the following criteria:
• **Theme Clarity**: Clear and consistent central theme throughout the script
• **Theme Depth**: Sophisticated exploration of the theme with nuanced treatment
• **Artistic Reinforcement**: Effective use of metaphor, symbolism, and narrative devices to enrich thematic content

3. Characterization
Assess character portrayal based on the following criteria:
• **Motivation Credibility**: Clear and believable character motivations that drive actions

• **Character Depth**: Emotional and psychological complexity creating well-rounded, multi-dimensional characters

• **Character Development**: Evident growth or meaningful transformation with natural, well-paced development arcs

4. Dramatic Engagement

Assess dramatic tension and audience engagement based on the following criteria:

• **Event Design**: Well-crafted, compelling events that sustain audience interest

• **Suspense Construction**: Effective use of foreshadowing, hints, and delayed revelations

• **Narrative Pacing**: Significant turning points that shift stakes or character trajectories

• **Tension Management**: Skillful buildup and release of dramatic tension throughout the narrative

5. Premise Fidelity

Assess adherence to the original premise based on the following criteria:

• **Conceptual Fidelity**: Faithful adherence to the core idea and thematic direction of the given premise

• **Element Retention**: Core premise elements—primary settings, characters, and central conflicts—are faithfully retained

Table 24: The five evaluation dimensions and their specific criteria used for pairwise script comparison. Each dimension is assessed independently, and the evaluator compares two scripts directly on each dimension's criteria.

Example: Complete Evaluation Prompt for Narrative Dimension (Storyline):

You are a professional script analyst. Please act as an impartial judge and evaluate the quality of the two scripts generated by different methods.

Your evaluation should focus **only** on the following dimension:

Dimension – Narrative

Evaluate the narrative based on the following criteria:

• Plot continuity and smooth transitions between events

• Logical consistency in contextual setups, world-building, storyline, and character behaviors

• Presence of a clear and complete dramatic narrative structure (typically encompassing Exposition, Rising Action, Climax, Falling Action, Resolution)

Avoid any position biases and ensure that the order in which the scripts are presented does not influence your decision. Do not let the length or formatting style of the summaries bias your judgment. Base your reasoning strictly on the content quality under the specified evaluation dimension.

Choose "A" if Script A clearly demonstrates superior performance on this dimension.

Choose "B" if Script B clearly demonstrates superior performance on this dimension.

Choose "Same" if both scripts are roughly equal in quality on this dimension, or if neither shows a significant advantage.

After comparing the two summaries, provide a one-sentence explanation of your judgment, and then output your final verdict in strict JSON format.

[Script A]

Premise A: [premise text]

Title A: [title]

Summary:

[beat summaries from all scenes]

[Script B]

Premise B: [premise text]

Title B: [title]

```
Summary:
[beat summaries from all scenes]

Respond strictly in the following JSON format:
{
 "explanation": "your explanation of which script is better and why",
 "verdict": "A" or "B" or "Same"
}
```

Table 25: Complete example of the evaluation prompt for the Narrative dimension (storyline comparison). The same template structure is used for all five dimensions, with the dimension-specific criteria replaced accordingly.