

GraphStory: Collaborative Story Writing through Event-Based Narrative Editing

Xuan-Vu Le*
University of Science
Ho Chi Minh, Vietnam
Vietnam National University
Ho Chi Minh, Vietnam

Minh-Loi Nguyen*
University of Science
Ho Chi Minh, Vietnam
Vietnam National University
Ho Chi Minh, Vietnam

Khanh-Duy Le
University of Science
Ho Chi Minh, Vietnam
Vietnam National University
Ho Chi Minh, Vietnam

Minh-Triet Tran
University of Science
Ho Chi Minh, Vietnam
Vietnam National University
Ho Chi Minh, Vietnam

Trung-Nghia Le[†]
University of Science
Ho Chi Minh, Vietnam
Vietnam National University
Ho Chi Minh, Vietnam

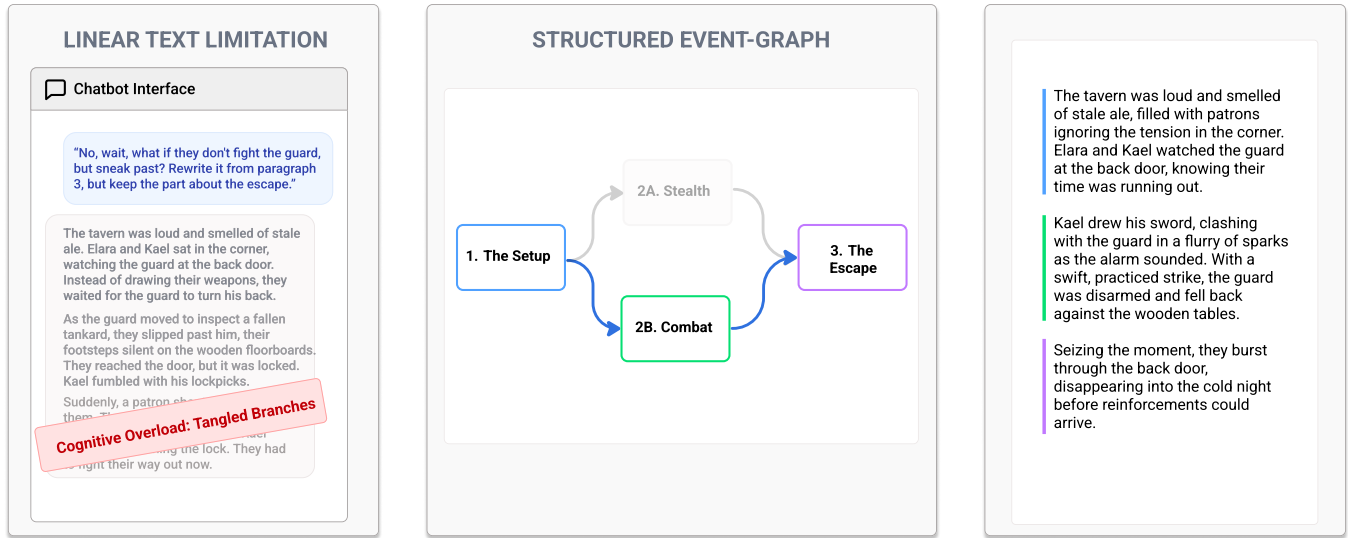


Figure 1: GraphStory concept and core workflow. (Left) Standard linear text interfaces often lead to cognitive overload when writers attempt to manage and rewrite branching ideas. (Middle) GraphStory introduces a structured event-graph to support explicit branching and multi-level editing, allowing users to efficiently structure their intent. (Right) The structured graph translates directly into a coherent narrative with clear visual provenance connecting the text back to the graph nodes.

Abstract

Story writing is a popular yet complex creative activity that requires organization of ideas and iterative exploration, particularly during early-stage ideation. While many AI-based writing assistants have

been developed, existing approaches primarily focus on generating long-form coherent text and improving user controllability during text production, providing limited support for brainstorming, connecting ideas, and validating alternative narrative flows. We present GraphStory, an interactive writing support system that leverages a graph-based representation to provide a comprehensive view of narrative structure and facilitate ideation. The system enables users to organize and connect plot points, explore alternative branches, and validate evolving narratives through an integrated story generation workflow. It further provides a structured interface to support efficient iteration over multiple story paths. Results from a user study with professional and semi-professional writers show that GraphStory reduces the effort of organizing narrative structures and better supports creativity and exploration compared to normal AI-based writing workflows.

*Both authors contributed equally to this research.

[†]Corresponding author. Email: ltngnia@fit.hcmus.edu.vn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, Washington, DC, USA

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

CCS Concepts

• **Computing methodologies** → **Computer vision; Natural language processing**; • **Human-centered computing** → **Information visualization; Interactive systems and tools**.

Keywords

AI-assisted creativity, Co-creative human–AI interaction, Graph-based storytelling, Interactive narrative editing, Event-based story modeling

ACM Reference Format:

Xuan-Vu Le, Minh-Loi Nguyen, Khanh-Duy Le, Minh-Triet Tran, and Trung-Nghia Le. 2026. GraphStory: Collaborative Story Writing through Event-Based Narrative Editing. In . ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 Introduction

Story writing is a common activity across a wide range of domains, including literature, screenwriting, journalism, marketing, and interactive media such as games [11, 49, 60]. In these contexts, writers must construct engaging stories by organizing events, developing characters, and maintaining coherent narrative structures [1, 17]. Given its creative and cognitively demanding nature [15, 29], story writing has attracted significant attention from researchers seeking to design interactive applications to support this process [5, 31, 54]. In particular, recent advances in artificial intelligence, especially Large Language Models (LLMs) [24, 35], have enabled new systems that assist writers by generating text, providing suggestions [7, 8, 27, 32, 57], and supporting different stages of the writing workflow [30, 39, 47, 58, 59].

Recent work has focused on enabling the generation of long-form, coherent narratives using a wide range of inputs [7, 39, 57]. These systems can condition generation on prefixes or suffixes, as well as higher-level specifications such as topic, style, theme, and tone [9, 13, 30]. More advanced approaches further incorporate multimodal or structured inputs, including images, sketches, outlines, world settings, and character descriptions [7, 23, 47, 58, 59]. By supporting diverse forms of input, these methods allow writers to better control the generated content and ensure alignment with their intent while maintaining fluency, coherence, and engagement [54]. However, these approaches primarily support text production and control, and provide limited assistance for developing initial narrative flow. In this stage, writers must connect individual ideas into a coherent structure, reason about relationships between events, and iteratively explore alternative story developments. Existing systems often require writers to rely on sequential prompting or manual rewriting, making it difficult to externalize, organize, and refine evolving narratives. As a result, writers frequently struggle to connect disparate ideas and effectively iterate over different narrative directions.

Graph-based representations have been widely adopted in various domains to model complex and structured information, offering users an intuitive way to organize and explore relationships between elements [21]. Prior work has shown that graphs are effective for supporting sensemaking, planning, and knowledge organization, as they externalize connections and provide a clear overview of interconnected components [36, 42, 45]. Users often report that graph

structures are more intuitive than linear formats when dealing with complex relationships, as they allow flexible navigation and direct controllability of individual elements [3]. These properties make graphs particularly well-suited for representing narrative structures, where events and plot points are inherently interconnected and often evolve in a non-linear manner. By explicitly capturing these relationships and supporting branching, graph representations provide a natural medium for developing and refining narrative flow.

In this paper, we present GraphStory, a writing support system that leverages a graph-based representation as an intermediate layer for both user interaction and LLM input. GraphStory enables writers to organize, connect, and iteratively develop their ideas into coherent narrative structures through an interactive graph interface (Fig. 1). The graph serves as structured context for the LLM, allowing the system to generate an instant output that are grounded in the evolving narrative and provide suggestion in improving connections between plot points. To support iterative development, GraphStory allows users to explore and compare alternative story branches, facilitating the validation and refinement of different narrative flows.

We evaluate GraphStory through a user study with professional and semi-professional writers. Our results show that GraphStory improves the writing experience by reducing the effort required to organize narrative structures and by better supporting creativity, exploration, and control compared to conventional LLM-based writing assistants.

In summary, our contributions are as follows:

- We propose a graph-based interaction paradigm for story writing that uses a structured representation to support the development and organization of narrative flow while bridging user interaction and LLM reasoning.
- We design and implement GraphStory, an interactive system that enables writers to connect ideas, explore alternative story branches, and iteratively refine narratives with grounded LLM assistance.
- We present findings from a user study demonstrating that GraphStory reduces the effort of organizing narratives and better supports creativity, exploration, and control compared to conventional LLM-based writing workflows.

2 Related Work

2.1 Narrative Writing with LLMs

Recent advances in large language models (LLMs) have demonstrated strong capabilities in generating fluent, coherent, and contextually relevant long-form text [7, 39, 57]. Both researchers and practitioners have recognized their potential for narrative writing tasks, including story generation, rewriting, and stylistic adaptation [14, 54]. Beyond casual use through chat-based interfaces, prior work has explored more structured and interactive systems that integrate LLMs into writing workflows, aiming to provide greater control and usability.

These systems aim to provide greater control over story generation by allowing users to specify a wide range of inputs that guide the output [48]. Early approaches rely on textual conditioning, such as prefixes or suffixes, where the model continues or completes

a given piece of text. More recent work expands this by incorporating higher-level attributes, including topic, genre, style, theme, and tone, enabling writers to shape not only what is written but how it is expressed [9, 13, 30]. In addition, some systems introduce structured inputs such as outlines, world settings, and character descriptions, which help anchor the generation in a predefined narrative context [47, 58, 59]. Beyond text, multimodal inputs, such as images, sketches, or videos, have also been explored to inspire or constrain storytelling [23]. By combining these diverse forms of input, these systems allow writers to exert finer-grained control over the generated content and produce narratives that better align with their intentions while maintaining coherence, fluency, and engagement.

Moreover, these systems seek to improve the user experience of story authoring by rethinking how writers provide input, moving beyond unrestricted natural-language prompting [2]. Rather than relying on a single, free-form prompt, many interfaces introduce more structured and interactive mechanisms, such as editable forms, sliders, templates, and modular input fields [10, 18, 40, 46]. These approaches allow writers to specify elements like characters, settings, and plot points incrementally, while making the effects of their choices easier to interpret and revise [47, 59]. Some systems further support step-by-step workflows, guiding users through stages such as planning, drafting, and refinement, so that input can be provided in a more organized and iterative manner [39, 58]. By making the input process more explicit and guided, these designs reduce ambiguity, lower the burden of prompt engineering, and provide writers with a clearer sense of control over the generated narrative.

Existing approaches to controllable text generation primarily focus on improving output quality [54], but provide limited support for developing and structuring narrative ideas. Writers often struggle to connect individual ideas into a coherent flow, explore alternative developments, and assess the impact of changes, as most systems rely on linear text and sequential prompting with little support for organizing evolving structures. To address this gap, our work shifts the focus to interaction design, introducing an interface that enables writers to structure, organize, and connect ideas before generation, and to receive outputs grounded in this structured representation.

2.2 Graph-based Interaction

Graph-based representations are widely used to model complex, structured information, providing an intuitive way for users to organize and explore relationships between interconnected elements. Prior research has demonstrated their effectiveness in tasks, such as sensemaking, planning, and knowledge management, where making relationships explicit helps users build and refine their understanding [36, 42]. Compared to linear representations, graphs enable flexible navigation and direct manipulation, allowing users to focus on individual components while maintaining awareness of the overall structure [3, 22, 41].

These characteristics make graphs particularly suitable for representing narrative structures. Stories are inherently composed of interconnected elements, including events, characters, and causal

relationships, which often evolve in non-linear ways [38, 43]. Existing work in narrative modeling has explored various graph-like structures, such as event networks and plot diagrams, to represent temporal sequences, causal dependencies, and character interactions [33, 50, 53, 56]. In domains like interactive storytelling and game design, branching structures are commonly used to model alternative story paths, enabling authors to design multiple possible developments and outcomes [43, 44].

Graph representations enable a level of abstraction that is difficult to achieve with plain text, allowing writers to model high-level story components as nodes and refine them incrementally for both overview and detail-oriented editing. This facilitates restructuring narratives, exploring alternative configurations, and maintaining coherence across complex storylines through branching, reorganization, and iterative refinement. Building on this idea, our system extends prior work by treating the graph not merely as a visualization or planning aid, but as the central interaction medium for narrative writing. In GraphStory, the graph functions both as an interface for organizing and exploring story elements and as structured input to the LLM, tightly integrating user-defined structure with model-generated content to support more interactive and flexible development and validation of narrative flows.

3 Formative Study

To inform the design of our system, we conducted a formative study to better understand writers' needs during narrative development. In particular, we focused on the stage where initial ideas are expanded into a coherent story, a process that involves organizing concepts, establishing connections between events, and iterating over alternative narrative directions. We also sought to explore how graph-based representations could support this process and what interaction mechanisms would be most intuitive and effective for writers. Based on this study, we aim to address the following research questions:

- **RQ1:** What challenges do writers encounter when developing initial ideas into a coherent narrative structure?
- **RQ2:** How do writers expect to interact with graph-based representations to support narrative development and exploration?

3.1 Participants

We recruited five participants (aged 18–22; 2 male, 3 female), all of whom were students majoring in writing-related fields, including screenwriting, literature, and copywriting. We refer to them as FP1–FP5 throughout the paper. All participants were experienced in writing and had prior experience using LLMs to support their work. Participants provided informed consent and received \$5 as compensation for their participation.

3.2 Study Design and Procedure

We conducted a two-part formative study to investigate writers' practices and challenges during narrative development. First, we carried out a semi-structured interview to understand how participants transform initial ideas into complete stories, focusing on their workflows, strategies for organizing ideas, and difficulties in connecting events and iterating over narrative directions.

In the second part, we developed a simple prototype that represents a story as a graph, where nodes correspond to narrative elements and edges represent progression and branching. Participants interacted with this interface to modify and reorganize a familiar story, and they could submit their edits to a large language model (LLM) to generate updated versions reflecting those changes. They followed a think-aloud protocol [12] while performing these tasks, allowing us to observe how they engaged with the graph and used LLM feedback.

Each session was conducted in a private room. Participants were first briefed on the study goals, then completed the initial interview, followed by a 30-minute interaction with the prototype with researcher support. Finally, they participated in a follow-up interview to provide feedback on the system's usability and usefulness.

3.3 Analysis

We conducted a thematic analysis [4] of the collected data, including the initial interviews, interaction sessions, and follow-up interviews. We adopted an inductive approach to identify recurring patterns related to participants' writing practices, challenges, and interactions with the graph-based prototype. The analysis process involved iterative coding and refinement, with multiple rounds of review and discussion among the authors to consolidate themes. Subjectivity was mitigated through this iterative process and multi-stage review.

3.4 Narrative Flow Development Challenge

Iterative Process. Participants consistently described narrative development as a highly iterative process. Rather than following a fixed sequence, they often explored multiple possible directions, revised earlier ideas, and refined story elements over time, involving frequent backtracking and comparison between alternatives when evaluating how different plot developments might influence the overall story. However, participants noted that existing tools provide limited support for such iteration, often requiring them to manually track or rewrite different versions. For example, FP1 shared that she “often struggles to trace back old ideas when using tools like ChatGPT,” making it difficult to revisit prior thoughts, while FP3, a screenwriting student, explained that he “usually prepares multiple story flows to compare and present to teacher,” highlighting the need to manage parallel narrative paths. This makes it difficult to efficiently explore and manage multiple narrative trajectories.

Coherent Connections Between Narrative Elements. Maintaining coherence across narrative elements emerged as a key challenge. Participants reported difficulty in establishing clear relationships between events, particularly when integrating new ideas into an existing structure, often needing to reason about causal, temporal, and thematic connections. For instance, FP2 explained that they “often think of some key events first and then try to connect them,” highlighting the effort required to link story components meaningfully, while FP5 noted that “initial ideas are often not coherent,” requiring further refinement to achieve consistency. As a result, ensuring that the story flows logically and cohesively across different parts demanded significant cognitive effort.

Content Length Guarantee. Participants highlighted the need to ensure that a story contains an appropriate amount of content depending on its intended form, such as a short story, script, or longer narrative. This often requires adding, expanding, or condensing events to meet expectations for pacing and completeness, as writers adjust the number of narrative elements to ensure the story feels sufficiently developed without being overly sparse or unnecessarily detailed. However, managing this balance can be challenging, as it requires maintaining awareness of both the overall structure and the contribution of individual events. For example, FP5 noted that they “may need to add more detail to ensure the content scope even when the current plot feels complete,” which can lead to introducing new events without disrupting the existing structure or, in some cases, requiring changes to the overall flow.

3.5 Graph-based Interaction

Comprehensive Overview with Multi-level Detail. Participants reported that the graph representation provides a more comprehensive view of the narrative compared to linear text. By organizing story elements as nodes and connections, the graph allows writers to simultaneously perceive the overall structure while inspecting individual components, supporting both high-level understanding and localized editing. For example, FP2 explained that “the graph view of the full story helps me know where to add more events and trace what it will affect,” highlighting how the representation supports reasoning about structural impact. This dual perspective enables participants to move more effectively between abstract story flow and detailed narrative elements than with paragraph-based formats.

Flexible Exploration of Narrative Flow. Participants found it easy to explore and test different narrative flows using the graph structure. In particular, modifying edges to change progression or branching allowed them to quickly reorganize the story and evaluate alternative directions. This interaction was perceived as more efficient and intuitive than rewriting text, as it enabled users to focus on structural changes without altering the entire narrative. For example, FP1 highlighted that adjusting connections between nodes made it easier to try out different story paths, while FP4 noted that “adjusting the graph and seeing the output from AI inspires how to write his own story,” emphasizing how rapid feedback supports creative exploration. As a result, the graph representation supports rapid experimentation and helps writers systematically explore and compare different story development.

4 GraphStory

4.1 System Overview

Based on insights from our formative study, we propose GraphStory, a grounded system designed to support writers in developing and refining their ideas through structured yet flexible exploration. The system comprises three core components: an Event-Graph Constructor, a Story Generator, and a Multi-level Flow Management module. The Event-Graph Constructor transforms user inputs into a sequence of nodes, providing a comprehensive representation that facilitates the organization and progressive development of narrative ideas. Building on this structure, the Story Generator functions as a human-in-the-loop module that produces provisional narrative

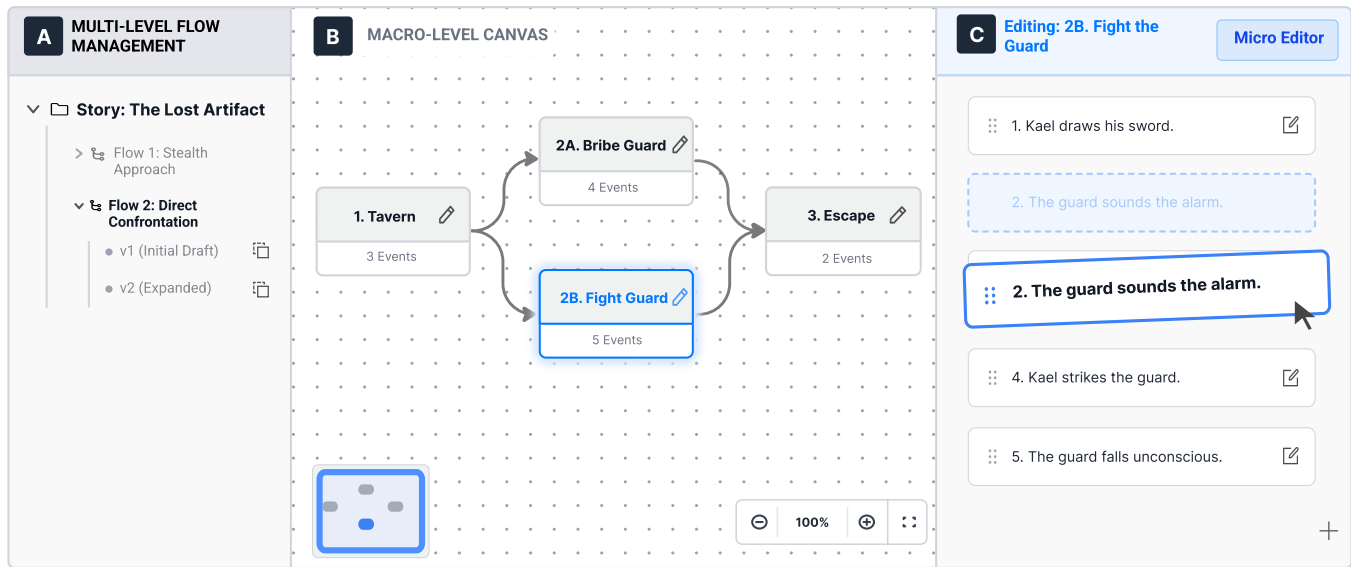


Figure 2: The main interface of GraphStory. (A) The Multi-level Flow Management panel organizes variations by Story, Flow, and Version. (B) The Macro-level Canvas provides a comprehensive overview of the narrative structure, allowing users to build and branch storylines using interconnected nodes (chunks). (C) The Micro-Editor enables fine-grained, drag-and-drop manipulation of individual events within a selected chunk.

outputs, offering writers reference points to validate emerging storylines, compare alternative flows, improve coherence, and manage content scope. Complementing these components, the Multi-level Flow Management module enables iteration across different levels of organization, allowing users to explore, compare, and refine narrative flows to identify those that best align with their intent.

4.2 Event-Graph Constructor

The Event-Graph Constructor is a core module designed to transform user-provided content into a structured event graph that represents the narrative flow (Fig. 3). It receives input in three forms: abstract ideas, structured outlines, and complete stories, each of which can be provided via text prompts or uploaded documents (e.g., PDF or Word files). Users explicitly choose the type of input, allowing the module to adapt its processing strategy to best fit the provided material. The output is a sequence of nodes that captures the key narrative events and organizes them in a hierarchical format, enabling writers to visualize and refine the story structure effectively.

To process abstract ideas, the system constructs a fine-grained outline from user input through a two-stage procedure. It first generates a preliminary outline that organizes the ideas into high-level sections, referred to as chunks, which represent major narrative units. It then elaborates each chunk by generating a corresponding list of events that capture key plot points and instantiate the narrative content, transforming initial concepts into a structured sequence that supports early-stage exploration and development.

For structured outlines, the system follows a similar process by converting the input into a more detailed representation. It divides the outline into high-level sections, referred to as chunks, based on its inherent structure and then generates a corresponding list

of events for each chunk grounded in its content. This expands the outline into a coherent sequence of key narrative points while preserving the original organizational intent.

For complete stories, the system first segments the text into paragraphs and determines an appropriate grouping size to combine consecutive paragraphs into chunks, forming higher-level narrative units that maintain coherence. Each chunk is then parsed to identify its main events, resulting in a structured and detailed representation of the story’s narrative flow.

Across all input types, the resulting sequence of nodes serves as the initialization of the event graph on the current flow. The interface provides a multi-resolution editing paradigm to manage both overarching plot and detailed narrative elements (Fig. 2). At the macro-level, the graph offers an overview of the structure (Fig. 2B): nodes (representing chunks) display a concise summary title and are connected by structural arrows depicting story progression. Writers can organize the plot by connecting nodes or extending paths to illustrate branching storylines during brainstorming. At the micro-level, writers can utilize semantic zooming to expand a node and inspect its numbered list of events. By interacting with a specific node, users open a dedicated event editor (Fig. 2C). Here, they can utilize direct manipulation to drag-and-drop, reorder, edit, delete, or manually add individual events strictly within the boundary of that specific chunk. Together, these interactions enable flexible, iterative manipulation of the narrative, supporting both high-level restructuring and fine-grained editing.

4.3 Story Generator

The story generator module is designed to produce a refined narrative from user-selected chunks, supporting validation of a chosen

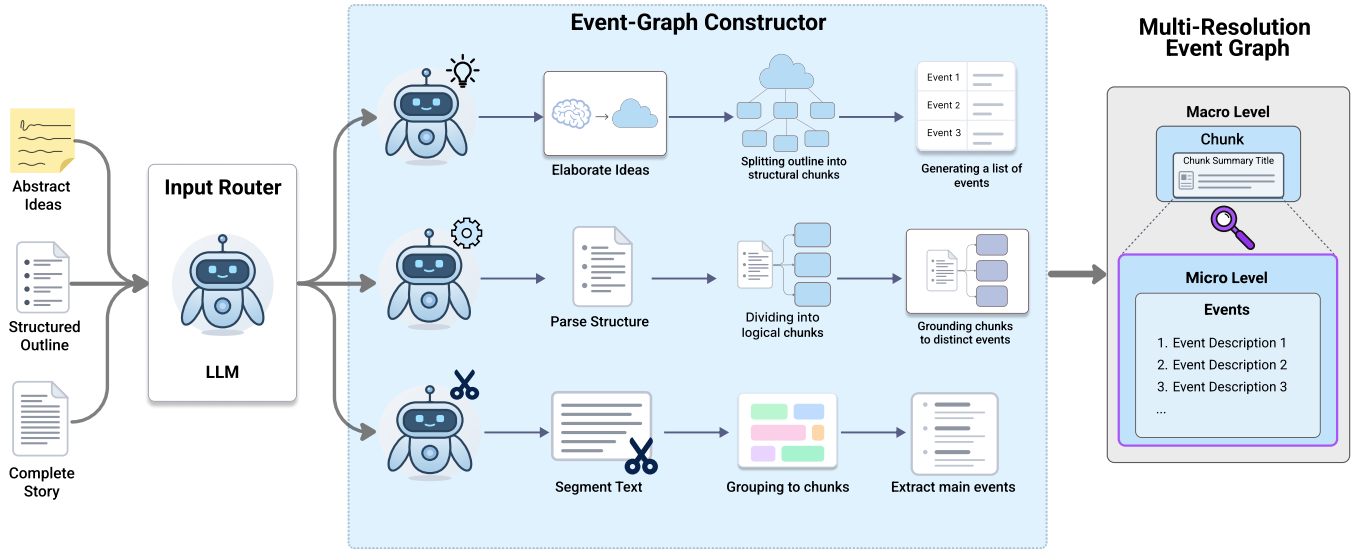


Figure 3: The Event-Graph Constructor pipeline. The Input Router processes three distinct forms of input (Abstract Ideas, Structured Outline, and Complete Story) via specialized workflows. Each pathway translates the input into structural chunks at the Macro Level and distinct narrative events at the Micro Level to build the event-graph.

storyline while offering guidance on improving coherence and managing content scope. To initiate generation for a specific narrative branch, users enter a selection mode by right-clicking an origin node to anchor the path, followed by sequentially left-clicking subsequent nodes to define the exact narrative chain (Fig. 4, Step 1). As nodes are selected, they illuminate and are connected by blue arrows, visually distinguishing the active generation queue from the underlying structural graph. If a user wishes to alter the sequence or cancel the selection, they simply click anywhere on the empty graph space to reset the path. In addition, users specify a target content length and may optionally configure attributes such as writing style, tone, or theme, which guide subsequent generation without constraining manual edits.

The generation procedure constructs a new sequence of events in two stages with a human-in-the-loop confirmation. In the first stage, the system operates at the intra-chunk level by adding new events to each selected chunk to improve local coherence and, when necessary, better align with the desired content scope, without modifying existing events so as to preserve the user’s original intent. In the second stage, the system considers the sequence of chunks at a global level, modifies added events, and inserts new events to improve transitions and overall flow across chunks. These additions are designed to provide supportive suggestions rather than enforce correctness, leaving users in control of refining or modifying the narrative.

Following this two-stage generation, the system presents the newly mixed sequence of original and AI-added events as a refined graph, utilizing the exact same visual interface as the baseline flow (Fig. 4, Step 2). To ensure clear visual provenance and maintain user agency, all AI-generated events are highlighted in a distinct color, instantly differentiating them from the writer’s original ideas. This visual and interactive consistency drastically

reduces the user’s cognitive load; users see the AI’s inter-chunk and intra-chunk suggestions integrated directly into the familiar node-and-event-list structure. Importantly, interaction is not restricted to the AI-generated content. Using the standard event editor, users can easily examine, modify, or reject the color-coded suggestions, while also retaining the ability to seamlessly edit any original events across all chunks. This comprehensive control allows authors to iteratively adjust the entire structure through direct manipulation, smoothing the narrative flow exactly to their preferences. Once users are satisfied with this refined graph and provide confirmation, the system inputs the finalized structure into the underlying LLM (specifically, GPT-4o [26]), incorporating user-specified configurations such as style and theme to generate a complete narrative text (Fig. 4, Step 3). To support user evaluation and explainability, the interface features a mapping mechanism that links chunks to their corresponding generated text segments. The resulting output provides a smooth and natural reading experience, enabling writers to intuitively sense the story flow while easily tracing how the LLM translated their events into prose.

4.4 Multi-level Flow Management

The Multi-level Flow Management module (Fig. 2A) provides structured organization and version control for story development, supporting multiple levels of narrative exploration. It is designed around three levels: story, flow, and version. When a user begins their work by providing input to the Event-Graph Constructor, a new story-level unit is created. This unit serves as the overarching container, encompassing all variations of event graphs and story iterations that the writer generates.

Within a story unit, each flow represents a distinct progression of the narrative. Flows are particularly useful when writers wish to explore entirely different story directions or when an existing

graph becomes too complex due to an excessive number of chunks and events. Each flow captures the structure and sequence of a particular variation, allowing writers to experiment without losing prior work.

A version is created whenever a story is generated from a selected set of chunks, capturing the refined sequence of events produced during the story generation phase along with the resulting narrative text. Rather than storing the original chunks alone, each version preserves this enhanced event structure as a concrete representation of the developed storyline. Writers can reuse a version by initializing a new flow with its associated graph, using the refined sequence of events as the starting point for further exploration and modification. This organization, spanning story, flow, and version, provides flexible control over the creative process, supporting iterative development, exploration of alternatives, and efficient reuse of previously generated structures.

5 User Study

We conducted a study to evaluate our system in terms of functionality, usability, and its impact on reducing cognitive load for writers. The study aimed to understand how effectively the system supports writers in managing complex narratives, generating coherent stories, and iteratively refining their ideas. By examining these aspects, we sought to assess the overall effectiveness of the system in facilitating both creative and cognitive aspects of story writing.

5.1 Participants

We recruited 16 students aged 18 to 22 (8 male, 8 female) to participate in the study. We refer to them as UP1–UP16 throughout the paper. All participants were majoring in fields related to writing and possessed advanced writing skills. None of the participants had taken part in any prior study involving our system. All participants reported being familiar with AI-based tools for writing. Informed consent was obtained from each participant, and they were compensated at a rate of \$5 per hour for their participation.

5.2 Study Design and Procedure

Each participant took part in the study individually in a private room, with one of the authors present to provide guidance and support as needed. Each session lasted approximately one hour. Participants were asked to complete a single story-writing task using a set of provided ideas.

Participants first wrote a story using ChatGPT and then completed the same task using our system. They were instructed to follow a think-out-loud protocol throughout the process, verbalizing their thoughts, strategies, and reasoning as they interacted with the tools. The entire session was recorded to capture both screen activity and verbalizations, allowing detailed analysis of participants' interactions and cognitive processes during story creation.

During the session, additional feedback was collected as participants worked on the tasks, providing immediate insights into their experience. After completing both tasks, participants filled out evaluation forms using a Likert-scale format based on predefined metrics. In addition, they completed the NASA Task Load Index (NASA-TLX) questionnaire [20] to assess perceived workload across

multiple dimensions. Finally, semi-structured interviews were conducted to gather more detailed qualitative feedback about their experience, preferences, and perceptions of the system compared to ChatGPT. This procedure enabled a comprehensive assessment of usability, functionality, and cognitive support in a controlled and consistent setting.

5.3 Quantitative Results

For the statistical analysis, we used Wilcoxon signed-rank tests [55] to analyze the data. Statistical significance was established at $p < 0.05$ for all analyses.

5.3.1 Interaction experience and generated content evaluation. A questionnaire using a 7-point Likert scale [28] (as shown Fig. 5) is used to assess the usability of our system compared to ChatGPT for this task. While chat-based interaction provides slightly greater ease of use than our graph interface (Q1, $p = 0.26$), our system better fulfills its purpose as a tool for writers to organize, brainstorm, and iteratively develop narrative ideas. It outperforms ChatGPT in terms of ease of iteration (Q2, $p < 0.001$), task efficiency (Q4, $p < 0.01$), and user comfort (Q5, $p < 0.001$). The slight increase in user ratings (Q3, $p < 0.05$) for the plot points generated by both systems suggests that the story generation module performs better than relying on ChatGPT alone.

5.3.2 Task load. NASA-TLX results (Fig. 6) indicate that GraphStory reduces perceived workload across multiple dimensions. Participants reported lower mental, physical, and temporal demands, while also rating their performance more favorably compared to the baseline condition. In addition, frustration levels were consistently lower, suggesting a more comfortable and manageable interaction experience.

These findings imply that the graph-based representation, together with the multi-level flow management mechanism, helps users better organize and navigate narrative elements. By providing a clearer structural overview and supporting incremental exploration, the system reduces cognitive effort and time pressure during story development. As a result, users can focus more on creative decision-making rather than on managing complexity, leading to a more efficient and less stressful writing process.

5.3.3 Functionality evaluation. We used a 7-point Likert-scale [28] questionnaire (Fig. 7) to evaluate key functionalities of GraphStory, including node sequence generation (Q1), the graph representation and interaction engine (Q2–Q3), the generated story (Q4), and the multi-level management system (Q5). The results show consistently high ratings across most questions, indicating that participants found the system effective and easy to use. Overall, these findings suggest that GraphStory provides strong support for structured and flexible narrative development.

5.4 Qualitative Results

Comprehensive Graph Representation. Participants consistently emphasized that the event graph offered a clear and well-structured view of their narratives. By organizing story elements from high-level concepts down to detailed events, the representation made relationships and progression more transparent across multiple levels of abstraction. This hierarchical structure enabled

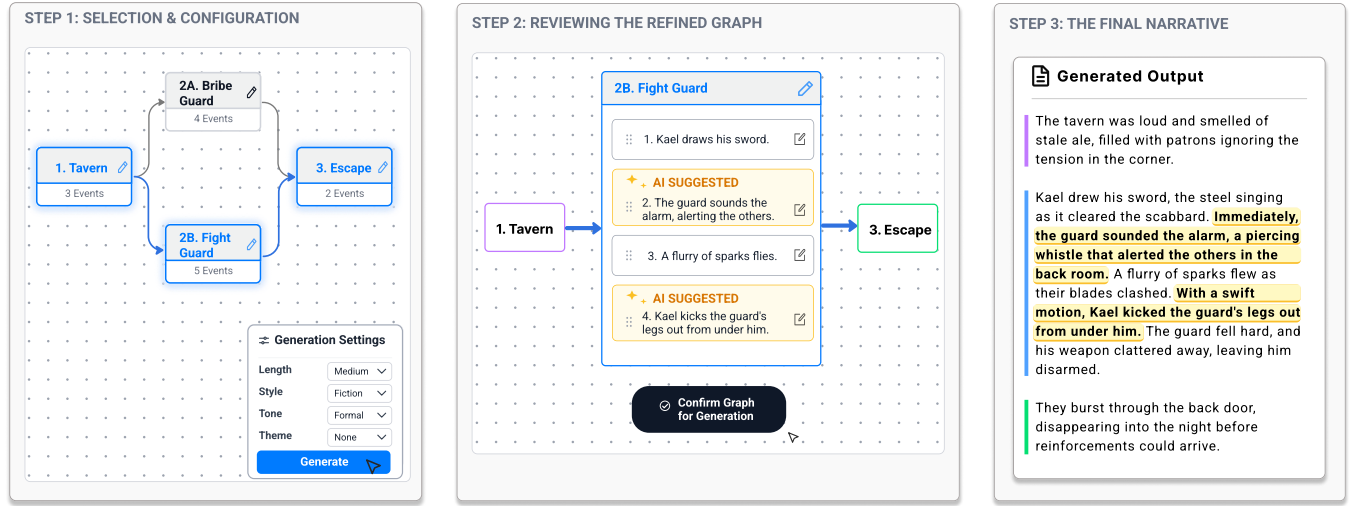


Figure 4: Interactive story generation workflow. (Step 1) Users select an origin node and path to configure the generation queue. (Step 2) The system operates a 2-stage generation, resulting in a refined graph where original events are preserved and AI-suggested intra-chunk events are highlighted in yellow for human-in-the-loop review. (Step 3) After user confirmation, the system generates the final coherent narrative, with visual provenance matching the text to the AI-suggested events. (Note: The yellow text highlighting in the Generated Output panel is for visualization purposes in this paper only, demonstrating how specific AI-suggested events translate into the final prose.)

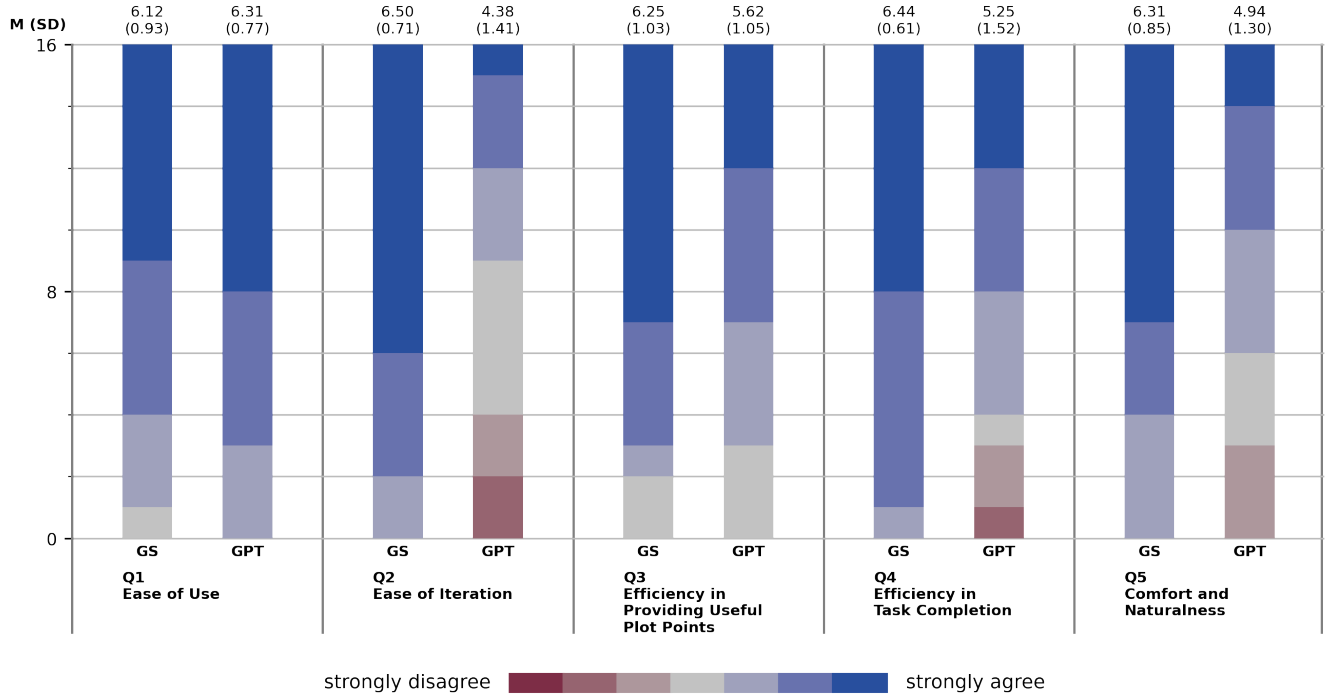


Figure 5: Questionnaire results on the 7-point Likert scale from "Strongly Disagree" to "Strongly Agree" in User Study to compare between GraphStory (GS) and ChatGPT (GPT). The x -axis represents different questions, covering various aspects such as system usability and interaction experience. The y -axis shows the Likert scale ratings provided by participants for two systems.

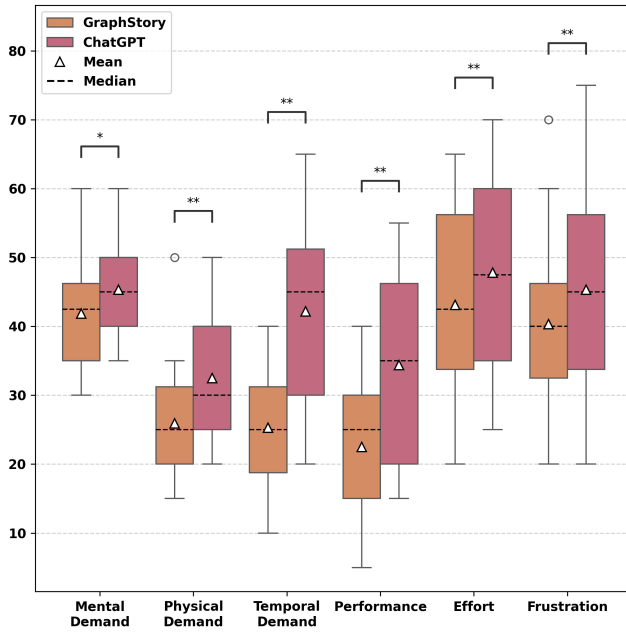


Figure 6: Task load results in NASA-TLX from User Study (the lower the better). The y -axis displays the six NASA-TLX subscales. The x -axis shows the corresponding scores. (* indicates $p < .05$, ** indicates $p < .01$)

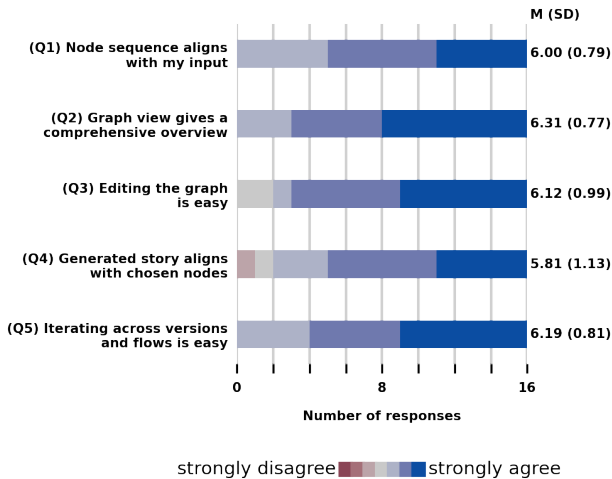


Figure 7: Questionnaire results on the 7-point Likert Scale from User Study to evaluate functionality proposed in GraphStory. Ratings are collected on a scale from “strongly disagree” to “strongly agree”.

writers to maintain an overview of complex storylines while also identifying gaps or inconsistencies. Several participants highlighted the ease of navigation, noting that they could quickly locate and edit specific parts of the story without rereading large portions of text. Others pointed out that the graph supported early-stage

ideation by allowing them to capture and organize emerging ideas without committing to fully written prose.

Acceleration of the Iterative Process. The system was widely perceived as speeding up the iterative writing process. Participants described how immediate outputs, such as updated graphs or draft text, allowed them to rapidly test narrative variations and refine story structures. This reduced the effort typically required to track and revise interconnected story elements, enabling greater focus on creative decisions. The ability to explore multiple narrative paths before committing to one direction was particularly valued. In addition, compared to traditional text-based tools, the system’s structured approach to managing revisions made it easier for participants to revisit and modify earlier ideas without losing track of changes.

Generation of Story Ideas. The story generation module played a key role in supporting creativity by introducing new plot points and connections. Participants found that the generated content often surfaced possibilities they had not initially considered, helping them expand and enrich their narratives. By illustrating how different story components could be linked, the system encouraged exploration of alternative directions while preserving a sense of authorial control. Some participants noted that longer generated outputs were especially useful for uncovering additional narrative opportunities, as they provided more detailed and varied suggestions.

Challenges with AI-Generated Content. Despite these benefits, participants also reported difficulties when interacting with AI-generated suggestions. In particular, an overabundance of newly introduced events sometimes led to confusion and disrupted the intended narrative flow. Deciding which suggestions to incorporate could become challenging, especially when outputs diverged significantly from the original storyline. Participants also observed that different generated versions, while sharing similar high-level plots, could vary substantially in detail, making it harder to maintain coherence across iterations. These issues suggest that while generative support enhances creativity, it can also increase cognitive load when not carefully balanced.

6 Discussion

Structured Graph Representations Bridge User Intent and AI Generation. A core aspect of GraphStory’s design is the use of a structured event graph as a transparent, editable intermediate layer between user intent and LLM generation. Rather than treating prompts as opaque, one-shot inputs as in chat-based interfaces [39], this representation aligns with human-centered AI principles [2] by making the mapping from intent to output inspectable and revisable at multiple levels of granularity. Supporting editing at both the macro (chunk) and micro (event) levels mirrors the divergent-convergent creative process [7, 16, 51]; systems that constrain writers to a single level of abstraction disrupt this natural oscillation, whereas GraphStory’s two-layer model explicitly accommodates both open-ended structuring and fine-grained refinement. Our quantitative results support this: GraphStory clearly outperformed ChatGPT in ease of iteration, task efficiency, and user comfort. Furthermore, functionality evaluations showed that participants highly rated the graph representation for providing a

comprehensive overview and for its ease of editing. GraphStory's three-level organization (story, flow, version) further addresses the challenge of tracking diverging narrative directions simultaneously—something linear chat interfaces fundamentally cannot support [52]. Participants confirmed this in qualitative feedback, noting that the system's structured approach to managing revisions made it easier to revisit and modify earlier ideas without losing track of changes, and that iterating across versions and flows was straightforward.

Visual Provenance Builds Trust in AI-Generated Content. Highlighting AI-suggested events in a distinct color and mapping them directly to generated prose gave participants greater confidence and editorial control. In qualitative feedback, participants consistently emphasized that the visual distinction between original and AI-generated content allowed them to quickly assess, accept, or reject suggestions without losing track of their own contributions. These observations are consistent with findings in explainable AI research [2, 34, 37], which show that provenance mechanisms rendering AI contributions legible are essential for sustaining authorial agency in co-creative workflows [32]. Our functionality evaluation further supports this, as participants favorably rated the alignment between generated stories and their selected nodes, suggesting that the two-stage generation process with visual provenance provides a meaningful degree of controllability over AI outputs.

Balancing AI Agency and Writer Control. A recurring tension in GraphStory is calibrating the boundary between AI contribution and writer autonomy [6]. When the story generator proposes inter-chunk events, it inevitably introduces narrative directions the writer did not intend. Our qualitative findings confirm this duality: participants valued creative suggestions and reported that generated content surfaced possibilities they had not initially considered, but they also noted that an overabundance of AI-added events was sometimes disruptive and could increase cognitive load. This tension is consistent with broader concerns about over-reliance on generative systems [6, 39, 59] and reflects a fundamental co-creative design challenge—systems contributing too little provide insufficient scaffolding, while those contributing too much erode authorial identity [32]. Notably, while survey responses indicated that chat-based interaction was perceived as slightly easier to use initially, this did not translate into better creative outcomes. This suggests that the modest learning curve of graph-based interaction is successfully offset by its structural advantages. Future systems should explore adaptive mechanisms that modulate AI contribution based on writer preferences and the current creative phase.

Broader Applicability and Ethical Considerations. While GraphStory targets short-to-medium narrative writing, the underlying paradigm of using a structured graph as an editable intermediate layer extends to screenwriting [39], game dialogue trees [49], and interactive fiction [31]. Recent LLM-powered authoring tools confirm the broader value of structured representations for bridging user intent and system execution [19, 25]. Ethical considerations also arise when applying this paradigm more broadly: authorship attribution becomes contested when AI substantially shapes narrative content [39], and models trained on large fiction corpora may encode cultural biases in story archetypes or character demographics [54] that writers absorb without awareness.

7 Limitations & Future Work

Despite GraphStory's demonstrated benefits, several limitations point to concrete future directions.

Non-determinism in generation. AI-generated content is stochastic: the same input can produce substantially different event sequences across runs. While variability can introduce novel ideas [7, 59], it makes it difficult for writers to isolate the precise effects of small edits, disrupting iterative workflows [39]. Future work should explore temperature-aware generation controls that let writers explicitly trade variability for consistency depending on their creative phase.

Graph-narrative alignment. Generated prose does not always faithfully reflect the event graph-text may conflate chunks, omit events, or introduce content outside the specified structure. This is partly inherent to current LLMs [54, 57], which struggle to maintain structural fidelity over long outputs. While GraphStory's two-stage generation mitigates this, future approaches should explore constrained decoding strategies that ground narrative outputs more reliably in the specified event sequence.

Participant population. Our study recruited 16 students aged 18–22 in writing-related fields, limiting generalizability to professional writers such as novelists or screenwriters, who may have different workflows and evaluation criteria [39]. Our current metrics may also not fully capture priorities such as stylistic voice or narrative coherence. Future work should conduct longitudinal studies with professionals in authentic production contexts.

Baseline and evaluation scope. Using ChatGPT as the sole baseline, while representative of common practice, does not isolate the contribution of graph-based structuring against dedicated outline-based tools [47, 58] or structured co-authoring systems [32, 59]. Furthermore, GraphStory was tested on moderately complex stories; at scale, with dozens of nodes or many parallel flows, the canvas may become visually cluttered and the cognitive benefit of graph overview may diminish [3]. Future work should include ablation comparisons against graph-free baselines and investigate adaptive layout algorithms with progressive disclosure mechanisms to support large-scale narrative graphs.

8 Conclusion

We presented GraphStory, an interactive writing support system designed to facilitate creative story writing through a graph-based representation of narrative elements. By providing a clear overview of story structure and supporting the organization and connection of events, GraphStory enables writers to explore alternative narrative paths, validate evolving storylines, and iterate efficiently over multiple story flows. Evaluation results indicate that the system reduces the cognitive effort required to manage complex narratives, accelerates the iterative writing process, and supports creativity by offering new story ideas, while maintaining user control over narrative development.

These findings highlight the benefits of integrating visual, structured representations with AI-assisted generation in creative writing workflows. While limitations remain, such as variability in AI-generated content and occasional misalignment with longer narratives, GraphStory demonstrates the potential of graph-based tools to enhance both the creative and cognitive aspects of storytelling.

In future work, we aim to improve the consistency of generated content and further strengthen the connection between the graph and story output, continuing to support writers in exploring, refining, and realizing complex narratives.

References

- [1] H Porter Abbott. 2021. *The Cambridge introduction to narrative*. Cambridge University Press.
- [2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.
- [3] Christopher Andrews, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 55–64.
- [4] Virginia Braun and Victoria Clarke. 2021. Thematic analysis: A practical guide. (2021).
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [6] Zana Bućinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction* 5, CSCW1 (2021), 1–21.
- [7] John Joon Young Chung, Woosook Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–19.
- [8] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*. 329–340.
- [9] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164* (2019).
- [10] Paramveer S Dhillon, Somayeh Molaei, Jiaqi Li, Maximilian Golub, Shaochun Zheng, and Lionel Peter Robert. 2024. Shaping human-AI collaboration: Varied scaffolding levels in co-writing with language models. In *Proceedings of the 2024 CHI conference on human factors in computing systems*. 1–18.
- [11] Tom Dowd. 2015. *Storytelling across worlds: Transmedia for creatives and producers*. Routledge.
- [12] K Anders Ericsson. 2017. Protocol analysis. *A companion to cognitive science* (2017), 425–432.
- [13] 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)*. 889–898.
- [14] Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2650–2660.
- [15] Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College Composition & Communication* 32, 4 (1981), 365–387.
- [16] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–18.
- [17] Gérard Genette. 1980. *Narrative discourse: An essay in method*. Vol. 3. Cornell University Press.
- [18] Parsa Ghaffari and Chris Hokamp. 2025. Narrative studio: Visual narrative exploration using llms and monte carlo tree search. In *Proceedings of the The 7th Workshop on Narrative Understanding*. 83–96.
- [19] Aditya Gunturu, Ben Pearman, Keiichi Ihara, Morteza Faraji, Bryan Wang, Rubaiat Habib Kazi, and Ryo Suzuki. 2025. MapStory: Prototyping Editable Map Animations with LLM Agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–20.
- [20] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
- [21] Ivan Herman, Guy Melançon, and M Scott Marshall. 2002. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on visualization and computer graphics* 6, 1 (2002), 24–43.
- [22] Antonio Hernandez, Jesús Bobadilla, Fernando Ortega, and Abraham Gutiérrez. 2018. Method to interactively visualize and navigate related information. *Expert Systems with Applications* 111 (2018), 61–75.
- [23] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. 1233–1239.
- [24] Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, et al. 2023. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351* (2023).
- [25] Zeyuan Huang, Cangjun Gao, Yaxian Shan, Haoxiang Hu, Qingkun Li, Xiaoming Deng, Cuixia Ma, Yu-Kun Lai, Yong-Jin Liu, Feng Tian, et al. 2025. SketchGPT: A sketch-based multimodal interface for application-agnostic LLM interaction. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–18.
- [26] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).
- [27] Daphne Ippolito, Ann Yuan, Andy Coenen, and Sehmon Burnam. 2022. Creative writing with an ai-powered writing assistant: Perspectives from professional writers. *arXiv preprint arXiv:2211.05030* (2022).
- [28] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British journal of applied science & technology* 7, 4 (2015), 396–403.
- [29] Ronald T Kellogg. 2008. Training writing skills: A cognitive developmental perspective. *Journal of writing research* 1, 1 (2008), 1–26.
- [30] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).
- [31] Ben Kybartas and Rafael Bidarra. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 3 (2016), 239–253.
- [32] Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–19.
- [33] Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081* (2018).
- [34] Q Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: informing design practices for explainable AI user experiences. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–15.
- [35] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics* 12 (2024), 157–173.
- [36] Zhicheng Liu, Nancy Nersessian, and John Stasko. 2008. Distributed cognition as a theoretical framework for information visualization. *IEEE Transactions on Visualization & Computer Graphics* 14, 06 (2008), 1173–1180.
- [37] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.
- [38] Semi Min and Juyong Park. 2019. Modeling narrative structure and dynamics with networks, sentiment analysis, and topic modeling. *PloS one* 14, 12 (2019), e0226025.
- [39] 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*. 1–34.
- [40] Nick Montfort. 2006. Natural language generation and narrative variation in interactive fiction. In *Proceedings of the AAAI Workshop on Computational Aesthetics*.
- [41] Tamara Munzner. 2025. Visualization analysis and design. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses*. 1–2.
- [42] Joseph D Novak and Alberto J Cañas. 2008. The theory underlying concept maps and how to construct and use them. (2008).
- [43] Kalpesh Padia, Kaveen Herath Bandara, and Christopher G Healey. 2019. A system for generating storyline visualizations using hierarchical task network planning. *Computers & Graphics* 78 (2019), 64–75.
- [44] Barbara Page. 1999. Hamlet on the Holodeck: The future of narrative in cyberspace. *MFS Modern Fiction Studies* 45, 2 (1999), 553–556.
- [45] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. McLean, VA, USA, 2–4.
- [46] Ahmed Y Radwan, Khaled M Alasmari, Omar A Abdulbagi, and Emad A Alghamdi. 2024. SARD: A human-AI collaborative story generation. In *International Conference on Human-Computer Interaction*. Springer, 94–105.
- [47] Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In

- Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4274–4295.
- [48] Melissa Roemmele and Andrew S Gordon. 2015. Creative help: A story writing assistant. In *International Conference on Interactive Digital Storytelling*. Springer, 81–92.
 - [49] Marie-Laure Ryan, James Ruppert, and John W Bernet. 2004. *Narrative across media: The languages of storytelling*. U of Nebraska Press.
 - [50] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics* 16, 6 (2010), 1139–1148.
 - [51] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.
 - [52] Sarah Stermann, Molly Jane Nicholas, and Eric Paulos. 2022. Towards Creative Version Control. *Proceedings of the ACM on human-computer interaction* 6, CSCW2 (2022), 1–25.
 - [53] Yuzuru Tanahashi and Kwan-Liu Ma. 2012. Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2679–2688.
 - [54] Maria Teleki, Vedangi Bengali, Xiangjue Dong, Sai Tejas Janjur, Haoran Liu, Tian Liu, Cong Wang, Ting Liu, Yin Zhang, Frank Shipman, et al. 2025. A Survey on LLMs for Story Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. 13954–13966.
 - [55] Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials* (2007), 1–3.
 - [56] Zhihua Yan and Xijin Tang. 2023. Narrative graph: Telling evolving stories based on event-centric temporal knowledge graph. *Journal of Systems Science and Systems Engineering* 32, 2 (2023), 206–221.
 - [57] 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*. 4393–4479.
 - [58] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7378–7385.
 - [59] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 841–852.
 - [60] Xi Zhang and T Ramayah. 2024. Solving the mystery of storytelling in destination marketing: A systematic review. *Journal of Hospitality and Tourism Management* 59 (2024), 222–237.