Narrative Planning Using LLMs as a Model of Action Believability

Lasantha Senanayake, Stephen G. Ware, and Gage Birchmeier

Narrative Intelligence Lab
University of Kentucky, Lexington, Kentucky 40506, USA
lasantha.senanayake@uky.edu, sgware@cs.uky.edu, gage.birchmeier@uky.edu

Abstract

Structured interactive storytelling can be difficult when it requires generating long event sequences that achieve a certain goal or meet the author's requirements. Narrative planning is one solution, but planning is expensive. The added requirement that every action in a plan seem believable for the story's characters, which is often solved by smaller nested planning problems, further increases that cost. We propose that a Large Language Model can serve as a good enough heuristic for labeling actions as believable. We explore three evaluations of such a system. First, is it cheaper? Second, can it produce the same kinds of stories a symbolic narrative planner can? Third, can it produce stories which a human audience find believable?

Introduction

Interactive narrative systems can often be categorized into one of two approaches: *emergent* storytelling systems, where a story results from the interactions of many independent agents, and *structured* storytelling systems, which aim to include important story beats or achieve specific author-defined constraints. Emergent storytelling games, like *The Sims* (Maxis 2000) and *Dwarf Fortress* (Bay 12 Games 2022), have achieved considerable success in the entertainment industry and have been the subject of much research (Trichopoulos, Alexandridis, and Caridakis 2023). Structured storytelling systems require planning ahead to ensure the story's constraints will be met despite unexpected actions from the player (Ware et al. 2022). Structured stories can be planned by hand, like *Mass Effect* (BioWare 2007) and *Baldur's Gate III* (Larian Studios 2023), but this requires a massive authoring effort and limits the story to trajectories imagined by the authors. There has also been research on automated story planning algorithms (Young et al. 2013), which can adapt to unexpected player actions while maintaining a story's essential structure.

Story planning is difficult because the actions taken by characters need to contribute to the author's goals while still seeming believable—the audience needs to be able to identify why a character took that action (Riedl and Young 2010). Narrative planning is a *puppet master problem*, where a single author agent needs to achieve its goals while making each character seem like an independent agent with its own goals and limited knowledge. One common solution is to use one planning process to ensure actions contribute to the author's goals and a separate planning process for each action to ensure it contributes to the goals of its characters (Riedl and Young 2010; Teutenberg and Porteous 2013; Ware and Siler 2021). These nested planning problems quickly become expensive, limiting the size and scope of stories that can be automatically planned. In this paper, we explore a neuro-symbolic approach to speed up narrative planning.

Large Language Models, or LLMs, are deep neural transformers (Vaswani et al. 2017) trained on large natural language corpora that can generate structured or unstructured output in response to instruction prompts. They have shown surprising success on a wide variety of tasks (OpenAI 2024), but they have weaknesses which limit their application to storytelling, especially structured storytelling (Dueifi and Eger 2024). They may hallucinate new objects or actions that their medium is not prepared to support (Li et al. 2026). As stories get longer, they lose track of the world state and style drifts (Wang et al. 2023). Neuro-symbolic storytelling systems use a formal symbolic model of the world to enforce a specific set of rules and avoid hallucination and drift, but let an LLM make decisions within those bounds.

We propose that a planner is well suited to be the symbolic model of a story world while an LLM can serve as a good enough heuristic for when actions will appear believable. We describe the Halberd Narrative Planner and test it on a suite of benchmark story planning problems. We evaluate our results in three ways. First, we consider the cost of Halberd relative to the purely symbolic story planner Sabre (Ware and Siler 2021). Second, we investigate whether Halberd generates the same kinds of stories. Finally, we test whether human readers find stories produced by Halberd realistic. Our results show that Halberd visits far fewer nodes than Sabre, but takes more wall-clock time to do so. The actions it selects are rated by human readers as making sense less often than Sabre's actions, but human readers show no significant overall preference for stories produced by either planner.

Related Work

Here we survey several symbolic and neuro-symbolic story planning systems. For a survey of story generation in general we refer readers to a survey by Kybartas and Bidarra (2016).

Symbolic Story Planning

Early planning systems like STRIPS (Fikes and Nilsson 1972) model events as having preconditions that must be true before they can occur and effects which modify the world state. Preconditions and effects are expressed in symbolic predicate logic. Early storytelling systems like Tale-Spin (Meehan 1977) and Universe (Lebowitz 1985) as well as modern storylet systems like Versu (Evans and Short 2013) use a similar model for story events. These and other systems like them choose a next event reactively with little or no search through the space of possible event sequences, so they are not long-term planners.

Young (1999) suggested planning algorithms for storytelling because they offer a generative, systematic search through all possible event sequences of any length. Cavazza et. al built interactive stories with planners based on the TV show *Friends* (Cavazza, Charles, and Mead 2002) and Shakespeare's *The Merchant of Venice* (Porteous, Cavazza, and Charles 2010). They encode contextual information into the preconditions of events to ensure they only occur when they would seem believable. This allows them to leverage fast off-the-shelf planning algorithms, but limits the ways actions can be recombined into new stories.

Narrative planning algorithms (Young et al. 2013) alleviate the need to encode story context into preconditions with special reasoning that ensures certain story properties. For example, Riedl and Young's IPOCL (Riedl and Young 2010) models intention by requiring that every action be part of multiple plans: one to achieve the author's goal for the story and one to achieve the goals of each character who takes that action. Glaive (Ware and Young 2014) uses a similar model, but only requires character plans to exist (not succeed), enabling conflict. Sabre (Ware and Siler 2021) uses the same model of intention but also reasons about each character's limited and possibly wrong beliefs.

Sabre highlights the high cost of symbolic narrative planning. Each action in a plan to achieve the author's goal must also be explained for every character who takes the action. Each explanation is generated in a separate planning process, starting in the state that character thinks they are in and ending with that character's goal. If these explanations include actions by other characters, more explanations must be recursively generated from what the first character believes about the second, and so on. This process results in rich and thorough explanations, but planning is already expensive, and nesting so many smaller planning problems in the main one makes it worse. Even with heuristic search the cost quickly becomes prohibitive, even for short stories.

The IMPRACTical story planner (Teutenberg and Porteous 2013) takes a slightly different approach similar to what we propose. Like Glaive and Sabre, it explains every action for each character who takes that action, but rather than generate a full plan it uses a relaxed plan heuristic (Hoffmann and Nebel 2001). The heuristic quickly finds the sketch of a plan. It may be incomplete and may not work, but it is much cheaper to generate. Where IMPRACTical uses a planning heuristic, we use an LLM.

Our goal is to create a neuro-symbolic story generation system that searches the space of formally defined action sequences like a planner and ensures actions make sense according to character beliefs and intentions like Sabre, without requiring narrative context to be encoded in preconditions and with lower cost per action like IMPRACTical.

Neuro-Symbolic Story Planning

Neuro-symbolic storytelling systems take two approaches: using symbolic systems to augment an LLM or using an LLM to guide a symbolic system. In the first category, researchers have supplemented LLMs with knowledge graphs to avoid hallucinations (Pan et al. 2025) and maintain coherence (Yoo and Cheong 2024). Another common approach is to first generate then follow an outline (Yao et al. 2019). Wang et al. (2023) survey several knowledge-augmented LLM story generation systems.

We should note that inventing new characters, items, and actions is sometimes a desirable property of LLM storytelling, especially in open-world story generation. We are specifically interested in situations where authors want to limit a storytelling system to a fixed set of assets, for example, when the story is realized in a game with a limited set of character models and animations.

In *Neural Story Planning* (2022) Ye et al. direct an LLM to imitate a partial-order planning algorithm to help it work backwards from the story's ending and ensure every action is causally explained. In *Planning Stories Neurally* (2024) Farrell and Ware use an LLM to guide the search of the symbolic Sabre planner to find solutions faster. Similarly, Senanayake and Ware (2025) use an LLM in place of a planning heuristic to help Sabre prioritize which plans are most likely to lead to solutions, though they found LLMs no better (and significantly more expensive) at estimating how many actions remain in an unfinished story.

We take a similar approach to Senanayake and Ware because we are using a symbolic planner as the basis for our system to ensure that actions are always taken from a predefined symbolic domain and that their preconditions are always met before they occur. However, instead of using an LLM to help a system like Sabre find full explanations for every action, we simply allow the LLM to decide which actions make sense in the current context and which do not. While we expect it to be less logically sound than Sabre, we hope these LLM explanations are good enough to maintain the believability of the story while avoiding the high cost of many nested planning problems.

LLMs as Story Agent Models

Many story planners can be said to have two parts: an author model and a character model. The author model ensures the story meets the author's requirements, while the character model(s) ensures characters act believably based on their goals, beliefs, personalities, etc. Many emergent systems like Dwarf Fortress (Bay 12 Games 2022) and Prom Week (McCoy et al. 2014) can be said to have little or no author model, while systems like Author (Dehn 1981) and the interactive Merchant of Venice (Porteous, Cavazza, and Charles 2010) are planners with little or no character model.

As mentioned previously, Sabre uses planning for both its author and character models (Ware and Siler 2021), while

IMPRACTical uses planning for its author model and a planning heuristic for its character model (Teutenberg and Porteous 2013). Farrell and Ware (2022) use LLM-guided planning for both models.

Yang, Gross, and Wampfler (2025) use an LLM for both models. One LLM per character suggests actions that character wants to do, and an author LLM chooses the best action for the story from among them.

Halberd is a story planner that uses symbolic forward search through the space of action sequences but only considers actions that an LLM classifies as believable based on the intentions and beliefs of the characters—in other words, it uses planning as its author model and an LLM as its character model. Our goal is to create a story planner which strictly follows a pre-defined logical story domain while producing believable character behaviors at lower cost. In this section, we will define our story planning problem and explain Halberd's architecture.

Story Planning Problems

The problem definition and search parts of Halberd are based on Sabre. For a full description of a Sabre narrative planning problem, we refer readers to Ware and Siler (2021). We briefly describe the parts relevant to this paper.

- **Objects:** A problem defines logical constants that represent a fixed list of people, places, and things.
- **Characters:** Some objects are designated as agents that must appear to have individual intentions and beliefs.
- **Actions:** A problem defines a fixed set of actions which each have a precondition, a proposition which must hold before it can occur, and an effect, a proposition which holds after it occurs. Preconditions and effects are expressed in a formal logic that can reason about the values of nominal and numeric variables as well as character beliefs about those values. An action also specifies zero to many *consenting characters* who need a reason to take the action. It specifies which characters see the action and how they change their beliefs when they see it happen.
- **Initial State:** Specifies the value of every variable at the start of the problem and any wrong beliefs characters start with.
- **Utility Functions:** The author and each character must define a numeric utility function they want to maximize.

A solution is a sequence of actions which (1) can be executed from the initial state, (2) improves the author's utility, and (3) is made up only of *explained* actions. Sabre and Halberd differ in how they define an explained action, which we will discuss next.

Search and Epistemic Depth

Sabre and Halberd both do forward search through the space of fully ground, totally ordered plans. The first node of the search graph represents the initial state and the empty plan. It has an *epistemic depth* of 0, meaning it represents what is actually true at the beginning of the story. Search proceeds by choosing a node s from the graph and creating a new node s' by adding an action whose precondition holds in s to the end of the plan. Node s' has the same epistemic depth as s.

The key difference between Sabre and Halberd is that Halberd's search only visits nodes at epistemic depth 0. Sabre uses nested searches (e.g. nested planning) at deeper epistemic depths to find plans that explain why an action makes sense for the characters who take it. Halberd simply asks an LLM whether an action is explained.

Each time Sabre creates a new node s' in the graph by taking an action, it also creates the root node s'_c of a new search for every consenting character c of that action. The new root s'_c represents the state character c thinks the world is in after taking the action, and it has an epistemic depth 1 higher than s'. So epistemic depth 0 represents the actual world; depth 1 represents what one character believes; depth 2 represents what one character believes another character believes, and so on. If Sabre eventually finds a plan starting in s'_c that improves character c's utility, that plan is an explanation for why c would agree to take the action. More formally, an action a is *explained* for Sabre if it is explained for all of its consenting characters, and an action is *explained for character* c if (1) there exists a plan (2) starting with a (3) which can be executed in the state c believes the world is in, (4) would increase c's utility, (5) is made up only of explained actions, and (6) is non-redundant (Ware and Siler 2021).

Alternatively, Halberd describes the problem, any past actions, and the current state of the world in natural language. It then lists all actions whose preconditions are satisfied in the current state and prompts the LLM to label which of those actions are explained based on the goals and beliefs of the characters who take those actions. If an action has no consenting characters, it is always considered explained.

For example, suppose the search chooses a node where protagonist Tom is at the local market and wants to bring home a potion that a nearby merchant is selling. Several next actions are possible, including "Tom buys the potion from the merchant" and "The merchant attacks Tom." Sabre will consider taking both of these actions because it does not yet know which of them can be explained. When it takes the "Tom buys the potion from the merchant" action, it creates three nodes: one where that action has been added to the actual story (epistemic depth 0), one where Tom imagines taking that action (epistemic depth 1), and one where the merchant imagines taking that action (epistemic depth 1). Both of these nodes at epistemic depth 1 represent nested searches; if successful, they will explain why Tom was willing to buy the potion and why the merchant was willing to sell it. Those searches may in turn create other searches at epistemic depth 2 to explain what Tom thinks other characters will do, and so on.

Halberd describes the current story and available actions to an LLM. Suppose the LLM says that Tom buying the potion is explained, but the merchant attacking Tom is not. Halberd creates only one new node at epistemic depth 0 which represents Tom buying the potion. Though it will not produce the complex nested explanation plans that Sabre does, our hope is that Halberd will only take actions that make sense while avoiding the high cost of the nested searches that verify the explanations.

Prompt Design

An example Halberd LLM prompt is given in Figure 1. Each node visited during the search causes the LLM to be prompted once, and the prompt has 6 parts.

- **System:** Tells the LLM its purpose is to label story actions as explained or not. This part is written by hand and always the same for every node.
- **Domain:** Explains the objects that exist, the goals of each character, and any non-obvious action effects. This part is written by hand and was the same for every node in the same story domain (e.g. all stories in the Space domain used this description).
- **Past:** Lists any actions that have occurred so far in the story. This part is procedurally generated for each node.
- **State:** Describes the current state of the world, including character beliefs. This part is procedurally generated for each node.
- **Actions:** Lists all actions whose precondition hold in the current state. This part is procedurally generated for each node.
- **Prompt:** Instructs the LLM to label which actions are explained, briefly explain why, and to give its response as an easy-to-parse JSON object. This part is written by hand and always the same for every node.

There are likely several ways to improve this prompt. The purpose of this paper is not prompt engineering, which we leave for future work, but to test the general viability of Halberd's strategy.

Evaluation

We investigate two hypotheses about Halberd, our neuro-symbolic story planner. First, we hypothesize Halberd can produce stories at a lower cost than Sabre by avoiding the nested planning problems used to justify each action. Second, we hypothesize the stories produced by Halberd remain believable despite its less exact method of justifying actions.

We investigate the second hypothesis in two ways. We start by testing to what extent Halberd produces solutions that Sabre also produces. This evaluation assumes Sabre's model of a believable story is an ideal one to emulate; however, we developed Halberd in part because we think Sabre's agent model is overly strict and expensive, so we preform a second evaluation to test whether human audiences find the stories produced by Sabre and Halberd believable.

Planner Configurations

We compare two narrative planning systems. The first is the the Sabre Narrative Planner version 0.8[1] using complete A* search with temporal cost and the Relaxed Plan Heuristic. This means that Sabre's search always expands the plan with the lowest total cost, where cost is the number of actions taken since the initial state and the estimated number of actions remaining. Actions remaining are estimated using the Relaxed Plan Heuristic, which is similar to the Fast Forward heuristic (Hoffmann and Nebel 2001).

[1]https://github.com/sgware/sabre

Our system, Halberd, is based on the Sabre code and uses a similar A* search. It also expands the plan with the lowest combined number of actions taken and actions remaining, using the same heuristic. Unlike Sabre, Halberd never generates or visits nodes with an epistemic depth above 0 and does not perform complete search, since actions deemed unexplained by the LLM are pruned.

Hardware

Story generation was performed on an NVIDIA DGX Spark, a machine designed for large neural network inference which uses a 20 core ARM CPU, a GB10 Grace Blackwell GPU, and 128 GB of LPDDR5x unified system memory (meaning that CPU and GPU share the same memory). This machine runs both the symbolic search and LLM locally, avoiding the time overhead of network API calls. While it does not run the LLM as quickly as a GPU cluster with dedicated VRAM might, it has enough speed and RAM to run the single-threaded search of Sabre and the highly parallel LLM for Halberd well. By running all elements of both story planners on the same hardware, we provide a fairer comparison for generation time.

Language Model Configuration

We tested Halberd with four different free and publicly available LLMs of different sizes and capabilities:

- **Llama 3.3:** Released by Meta in December 2024, we used the Instruct version trained on examples of commands and desired responses (Llama 2024). It has 70 billion parameters and we used the 8 bit quantized version of Bartowski's model released on Huggingface.
- **Qwen 3:** Released by Alibaba Cloud in April 2025, we used the version with 8 billion parameters and 8 bit quantization (Qwen 2025). We did not use reasoning.
- **Qwen 3 MoE:** A variant of Qwen 3 with a Mixture of Experts (MoE) architecture (Qwen 2025). While it has 30 billion parameters, it selects 3 billion to be active during inference for each token. We did not use reasoning.
- **GPT-OSS:** Released by OpenAI in August 2025, it is an MoE model with 117 billion parameters that activates 5.1 billion per token (OpenAI 2025). It uses 4 bit MXFP4 quantization. This is a reasoning model, meaning it generates a thought process to explain its answers. Reasoning models run too slowly on our Spark hardware, so we ran this model via network API calls to `OpenRouter.org`. While this makes it difficult to compare cost to the other models, we wanted to test at least one reasoning model.

The locally hosted models (Llama 3.3, Qwen 3 and Qwen 3 MoE) were launched using the same `llama.cpp` backend with 8 bit quantized KV cache, flash attention enabled and with a 16384 token context window. The models used *temperature* 0.6, *top-p* 0.95, *top-k* 20, *min-p* 0, and *presence penalty* 0. Stories were generated with a fixed random seed of 42 and maximum of 8192 output tokens. GPT-OSS, ran separately via OpenRouter routed through Groq, used a reasoning effort of *medium* and the same seed of 42, but otherwise used server side default inference settings.

System: I will tell you a story and give you a list of what actions can happen next. Your job is to label which actions make sense for the characters who are taking them.

Domain: This is a story about an interstellar explorer, Zoe, who is on a mission to meet and befriend alien species. There are two characters: Zoe and the lizard. Both characters want to be alive and to be at a safe location. […] It is safe for characters to be in the cave or on the ship during an eruption.

Past: These are the events that have happened so far in the story:
Zoe teleports from the ship to the surface, and the guardian of the surface sees them.
The lizard walks to the surface.

State: This is the current situation after those events: Zoe is at the surface. The lizard is at the surface. Zoe is not fighting with the lizard. […] Zoe believes that Zoe is at the surface. […] The lizard believes that the surface is safe.

Actions: These are all of the actions that could possibly happen next:
1) Zoe teleports from the surface to the ship.
2) Zoe walks to the cave.
[…]
7) The lizard walks to the cave.

Prompt: Note that just because an action is possible does not mean it makes sense. For each action, consider it strictly from the perspective of the character taking it. Ask yourself: given what that character currently believes and wants, would they choose to do this? A character can only act on what they know and believe to be true right now. They cannot know things they have not observed, and they would not take actions that conflict with their own goals.

For each action that genuinely makes sense from the acting character's perspective, give a short one-sentence explanation grounded in that character's current beliefs and goals. Leave out any action that the character would have no reason to take, or that would work against what they want.

Give your answer as a JSON object that maps action ID numbers to explanations. It should look like this:

{ "actionId": "brief explanation grounded in the character's beliefs and goals" }

Only include actions that make sense. Do not include actions that are possible but unmotivated. Do not use knowledge that the acting character would not have. Do not invent new characters, objects, or actions. If none of the actions make sense return an empty object {}.

Figure 1: Example Halberd LLM prompt for the space domain. This example corrects an error in the procedural part of the prompt that caused some sentences not to start with a capital letter.

If the LLM's response cannot be parsed, Halberd retries up to three times, each time appending the previous response and the message "Above JSON response was malformed, please retry" to the prompt, with a one second delay between attempts. If all three attempts fail to produce a parseable response, Halberd logs an error and returns an empty set of suggested actions for that node.

Sampling Solution Stories

To perform these evaluations, we need to compare solutions from Sabre and Halberd. Our goals for this sampling were to produce a variety of unique solutions across several story domains while keeping the number of stories small enough that they can all be evaluated by multiple human subjects.

We chose five story domains from the Sabre Benchmark Problems (Ware and Farrell 2023) that represent several authors, genres, and systems. Problems were large enough to produce multi-action stories but small enough that Sabre and Halberd generate solutions in a reasonable time. For each problem, we generated the first five unique solutions with both Sabre and Halberd. We generated 5 stories in 5 domains for 5 planners, so 25 stories per planner, or 125 stories total.

Most problems have multiple author utility thresholds, meaning stories can have different author requirements. For example, in the Save Gramma problem, an author utility of 1 allows any story that ends with the protagonist dying or succeeding, whereas an author utility of 2 only allows stories where the protagonist succeeds. When a problem had multiple utility thresholds, we used several to increase the variety of stories.

- **Basketball:** Originally introduced by Kartal, Koenig, and Guy (2014) for an MCTS story generation system, this domain tells mystery stories where characters commit crimes and a detective investigates them. We generated 3 stories with author utility 1 and 2 stories with author utility 2.
- **Fantasy:** Originally introduced by Ware and Young for the Glaive planner (2014), this domain tells stories about three Medieval characters and a dragon who want different combinations of food, love, and riches. We generated 2 stories with author utility 1, 2 stories with author utility 2, and 1 story with author utility 3.
- **Space:** Also introduced for Glaive, this domain tells science fiction stories about a space ship captain who can fight or make peace with an alien on a hard planet. We generated 1 story with author utility 1, 2 stories with author utility 2, and 2 stories with author utility 4.
- **Save Gramma:** Introduced by Ware et al. (2022) for a story game, this domain tells stories about Tom's attempts to obtain a lifesaving potion from a merchant despite a bandit and law-abiding guard. We generated 3 stories with author utility 1 and 2 with author utility 2.
- **Lovers:** Introduced by Farrell and Ware (2020) for performing belief and intention recognition, this domain tells stories about characters who want to gain favor with

those they love by giving gifts, but allows characters to lie to manipulate others. This domain has only one author utility, so we generated 5 stories with author utility 1. When translating this problem to natural language, we translated C1, C2, and C3 as Alice, Bob, and Calvin respectively, and did similar translations with items and rooms. Originally, this problem required both Alice and Calvin to be happy to achieve author utility 1, but we modified it so that each character being happy gave 1 point of author utility. This made it possible to reach author utility 2, but only 1 generated story achieved this.

Sabre is known to produce solutions for all problems, so we ran it with the recommended temporal and epistemic limits until it produced 5 solutions for all domains. For Halberd, we limited the search in each domain to 1000 nodes visited.

Cost

We compared several metrics to measure the relative cost of generating our sample stories using Sabre and Halberd: number of search nodes visited, time taken, and (for Halberd) whether the search found a solution and tokens used.

Halberd failed to solve some problems. All versions failed to find a solution to Fantasy when the author utility threshold was set above 1. The versions using Qwen 3 (non-MoE) and GPT-OSS failed to generate some solutions in the Basketball domain. Halberd is effectively using an LLM to prune its search, and if critical actions are pruned early it may becomes impossible to generate some solutions.

As expected, Sabre visits more nodes but Halberd takes more time. Sabre visited about 7 million nodes over 3 hours to generate 25 solutions (about 1.4 milliseconds per node). The fastest version of Halberd (Qwen 3) visited about 4 thousand nodes but took almost 5 hours (4 seconds per node). The slowest (Llama 3.3) visited about the same number of nodes but took over 65 hours (1 minute per node).

While time is probably the more important metric for measuring performance, because LLM technology is relatively new and improving rapidly, we are optimistic that future LLMs may be able to reduce inference time to the point that Halberd's much lower node count will prove an advantage. By significantly reducing the number of nodes visited during the search, Halberd will be able to generate longer stories before its search becomes intractable.

Improved performance is already possible when more memory or dedicated GPU hardware is available. Sabre is single-threaded, and while it could presumably be rewritten to use multi-threaded search, it would still need to synchronize on a single priority queue and search graph, limiting the impact of parallel processing. Halberd may benefit more from parallelization if multiple instances of its LLM are available simultaneously since its primary cost is the time it takes the LLM to evaluate a node. The GPT-OSS version of Halberd shows what is possible with the improved hardware available on OpenRouter. As a reasoning model, it generated 3 times more tokens but took only 8 hours despite the overhead of the network API calls.

Believability According to Sabre

Having discussed the relative cost of story generation, we now want to evaluate the quality of stories produced by Halberd. We explore two definitions of quality. First, because Halberd solves the same problems as Sabre, we evaluate to what extent Halberd produces plans that Sabre would consider valid solutions. Sabre's model of agent believability is based on the beliefs and intentions of characters and was evaluated in a series of experiments with human audiences (Shirvani, Farrell, and Ware 2018).

For each solution generated by Halberd, we used Birchmeier's Sabre Verification Tool[2] to test whether it could have been discovered by Sabre. For every consenting character of every action in every story, this tool runs a search to find a Sabre-style explanation for that character taking that action. We used the limits on author temporal depth, character temporal depth, and epistemic depth recommended in the benchmark suite for each problem but imposed no limit on the number of nodes visited or time spent. As a sanity check, we confirmed these settings were enough to verify all of Sabre's solutions. In other words, if we remove the explanations generated by Sabre from its solutions, the verifier tool is able to re-discover them with these settings.

Table 2 shows that the Llama 3.3 version of Halberd achieves 70% accuracy on average while both versions of Qwen achieve 73%. The GPT-OSS version generated the most Sabre-like solutions, with the verifier tool able to find explanations for its actions 85% of the time. Most models produced Sabre-like solutions in the Space domain, while the Save Gramma domain produced the least Sabre-like solutions. The Basketball domain was inconsistent, with Sabre-like accuracy ranging from 66% (Qwen 3 MoE) to 91% (GPT-OSS). Because all models failed to generate some solutions in Fantasy the number of explanations is much lower than in other domains.

Believability According to a Human Audience

Our first evaluation of quality assumes that Sabre's model is an ideal one to emulate. However, one motivation for developing Halberd is that we find Sabre's model of believability precise but overly expensive. We hypothesize that an LLM can provide a "good enough" model of when an action will seem reasonable in a story without requiring full explanations for every action. Therefore, we also want to evaluate the quality of Halberd plans with a human audience, and we can also evaluate the quality of Sabre's stories as a baseline.

After analyzing the cost and accuracy of Halberd with four LLMs, we chose to evaluate the stories generated with Qwen 3 MoE, since it generated 5 solutions in all but one domain at a reasonable cost. Because this version of Halberd only generated 2 solutions in the Fantasy domain, and one of them was identical to one of Sabre's solutions, we excluded that domain from these experiments. We evaluated 5 solutions in the 4 remaining domains for 2 planners, or 40 stories total.

[2]https://github.com/gsbirch/sabre-verification-tool

Table 1: Story generation costs for Sabre and for Halberd using four different LLMs, broken down by individual story. Nodes is the number of nodes visited during search. Time is in seconds. Succ. indicates whether that search found a solution (Y/N). When a search failed to find all of the intended unique solutions for a configuration, the full cost of that failed search is recorded once, in the row marked N; other story slots for that same configuration show a dash (—) in Nodes/Time but still count as failures.

Story	Sabre		Llama 3.3				Qwen 3				Qwen 3 MoE				GPT-OSS			
	Nodes	Time	Nodes	Time	Tok	Succ	Nodes	Time	Tok	Succ	Nodes	Time	Tok	Succ	Nodes	Time	Tok	Succ
BBall_any-1	41,153	61.6	5	343.6	6,518	Y	20	121.9	31,920	Y	6	22.5	7,591	Y	4	15.2	10,657	Y
BBall_any-2	42,255	64.1	6	343.6	6,518	Y	21	121.9	31,920	Y	8	30.4	9,300	Y	7	24.3	17,428	Y
BBall_any-3	352,902	328.0	10	556.2	11,441	Y	26	141.5	38,551	Y	12	53.3	14,304	Y	8	24.3	17,428	Y
BBall_both-1	844,387	1,105.3	684	56,733.9	1,136,171	Y	—	—	—	N	187	1,254.1	306,502	Y	882	5,658.0	3,423,299	Y
BBall_both-2	868,257	1,150.3	685	56,733.9	1,136,171	Y	1,000	6,497.4	1,797,099	N	419	2,833.7	682,919	Y	1,000	6,394.2	3,874,584	N
BBall Total	**2,148,954**	**2,709.4**	**1,390**	**114,711.4**	**2,296,819**	**5/5**	**1,067**	**6,882.7**	**1,899,490**	**3/5**	**632**	**4,194.1**	**1,020,616**	**5/5**	**1,901**	**12,116.1**	**7,343,396**	**4/5**
Fantasy_any-1	4	0.0	4	138.5	4,052	Y	5	12.4	5,556	Y	9	33.3	10,858	Y	16	82.8	43,469	Y
Fantasy_any-2	186	0.3	19	560.4	22,648	Y	16	40.5	19,395	Y	24	94.1	30,092	Y	28	130.9	76,260	Y
Fantasy_two-1	3,000	3.6	—	—	—	N	—	—	—	N	—	—	—	N	—	—	—	N
Fantasy_two-2	3,004	3.6	1,000	43,851.7	1,375,012	N	1,000	3,096.9	1,414,789	N	1,000	4,763.7	1,404,038	N	1,000	5,737.7	3,221,876	N
Fantasy_all-1	3,716,279	5,737.9	1,000	43,842.8	1,375,012	N	1,000	3,096.1	1,414,789	N	1,000	4,891.3	1,404,038	N	1,000	5,738.5	3,221,876	N
Fantasy Total	**3,722,473**	**5,745.5**	**2,023**	**88,393.5**	**2,776,724**	**2/5**	**2,021**	**6,246.0**	**2,854,529**	**2/5**	**2,033**	**9,782.5**	**2,849,026**	**2/5**	**2,044**	**11,689.9**	**6,563,481**	**2/5**
Space_any-1	26	0.1	3	43.0	1,780	Y	4	8.7	2,969	Y	3	4.2	1,788	Y	3	6.5	4,304	Y
Space_two-1	30	0.0	4	55.9	2,680	Y	5	8.7	2,969	Y	4	5.5	2,710	Y	4	9.6	6,381	Y
Space_two-2	44	0.0	6	75.2	3,595	Y	7	10.7	3,923	Y	5	5.5	2,710	Y	5	9.6	6,381	Y
Space_four-1	4,406	0.7	83	1,990.1	60,957	Y	285	429.4	220,109	Y	987	2,143.7	550,615	Y	93	297.1	105,723	Y
Space_four-2	4,413	0.7	84	1,990.1	60,957	Y	290	434.5	223,081	Y	988	2,143.7	550,615	Y	95	299.5	107,283	Y
Space Total	**8,919**	**1.6**	**180**	**4,154.3**	**129,969**	**5/5**	**591**	**891.9**	**453,051**	**5/5**	**1,987**	**4,302.6**	**1,108,438**	**5/5**	**200**	**622.3**	**230,072**	**5/5**
Gramma_any-1	3,662	7.7	3	153.9	3,125	Y	8	25.2	11,011	Y	8	59.5	11,684	Y	32	224.0	101,318	Y
Gramma_any-2	23,172	59.9	6	306.0	6,255	Y	16	49.3	22,085	Y	12	101.7	17,393	Y	36	235.4	110,418	Y
Gramma_any-3	24,450	62.5	10	571.7	11,151	Y	27	92.1	38,556	Y	13	101.7	17,393	Y	63	429.0	195,234	Y
Gramma_win-1	43,937	91.9	14	1,024.9	20,523	Y	39	140.4	59,436	Y	45	382.9	73,378	Y	54	386.5	168,698	Y
Gramma_win-2	271,265	483.3	47	4,054.9	72,203	Y	45	157.8	67,564	Y	47	388.8	74,985	Y	70	462.9	214,361	Y
Gramma Total	**366,486**	**705.4**	**80**	**6,111.6**	**113,257**	**5/5**	**135**	**464.8**	**198,652**	**5/5**	**125**	**1,034.6**	**194,833**	**5/5**	**255**	**1,737.9**	**790,029**	**5/5**
Lovers-1	39,156	69.7	32	2,879.8	40,646	Y	74	442.3	100,261	Y	65	442.7	77,512	Y	15	545.1	52,547	Y
Lovers-2	251,487	335.4	36	3,194.3	44,667	Y	75	442.3	100,261	Y	66	442.7	77,512	Y	25	595.2	83,422	Y
Lovers-3	251,487	335.4	43	3,817.7	52,772	Y	96	592.0	128,779	Y	70	469.3	81,675	Y	43	798.2	144,133	Y
Lovers-4	260,562	356.6	44	3,817.7	52,772	Y	111	704.3	149,213	Y	80	525.1	93,324	Y	46	832.4	149,013	Y
Lovers-5	343,606	565.8	84	7,633.4	105,054	Y	136	836.7	181,819	Y	115	745.0	138,208	Y	61	966.8	202,516	Y
Lovers Total	**1,146,298**	**1,662.9**	**239**	**21,343.0**	**295,911**	**5/5**	**492**	**3,017.6**	**660,333**	**5/5**	**396**	**2,624.9**	**468,231**	**5/5**	**190**	**3,737.7**	**631,631**	**5/5**
Total	**7,393,130**	**10,824.7**	**3,912**	**234,713.7**	**5,612,680**	**22/25**	**4,306**	**17,503.0**	**6,066,055**	**20/25**	**5,173**	**21,938.6**	**5,641,144**	**22/25**	**4,590**	**29,903.8**	**15,558,609**	**21/25**

Action Believability In our first experiment, we showed 50 subjects 4 stories each in random order, one from each story domain. After explaining the context and initial state, we showed them each action in the story and asked them to mark which actions *did not* make sense. Subjects were allowed to mark no actions if they thought every action made sense. Each subject's stories were either all generated by Sabre or all by Halberd. We distributed subjects round robin, so that each story was evaluated by about 5 subjects.

This study was approved by the University of Kentucky's IRB under protocol number 116913. We ran this study on Prolific and paid subjects $2 USD. Subjects took an average time of 11:44 to complete the study.

Some data was removed. Subjects who completed the whole experiment (instruction + 4 stories) in under 5 minutes were automatically removed and not compensated. After each story, we asked a comprehension question about the initial state. We removed a subject's data for a specific story if they spent less than 1 minute on that story or if they got its comprehension question wrong, but these subjects were fully compensated.

Table 4 reports, for each domain, the number of subjects whose data we used before filtering (Raw), after removing subjects who completed the whole study in under 5 minutes (Processed), and after the additional per-domain filtering.

We chose a design of asking subjects to mark actions that *do not* make sense because we want to measure how often an action stands out as being unreasonable[3]. It is meant to approximate a human version of the Sabre Verification Tool's failure to find an explanation for an action. One weakness in this design is that subjects had no way to report that a story did not make sense because actions were missing; we chose this design because it parallels our earlier evaluation of Sabre accuracy, is simple, and has a low cognitive load.

We measured inter-rater reliability by calculating Fleiss's

[3]Whether readers automatically try to find explanations for actions in a story as they read (and to what extent prompting changes that behavior) is an open question in narrative psychology. We refer readers to Zwaan, Magliano, and Graesser (1995) for a discussion of the Constructionist theory of story understanding.

Table 2: Halberd's accuracy at generating stories that are valid Sabre solutions. For each LLM in each story domain, this table shows the number of explanations verified (Found), the total number of explanations needed (Total), and the planner's accuracy (Acc) at using only actions Sabre could explain.

Domain	Halberd (Llama 3.3)			Halberd (Qwen 3)			Halberd (Qwen 3 MoE)			Halberd (GTP-OSS)		
	Found	Total	Acc	Found	Total	Acc	Found	Total	Acc	Found	Total	Acc
BBall	21	29	72%	12	13	92%	19	29	66%	21	23	91%
Fantasy	4	4	100%	4	4	100%	4	4	100%	4	5	80%
Space	8	10	80%	14	19	74%	13	14	93%	11	12	92%
Gramma	6	16	38%	10	20	50%	11	20	55%	22	27	81%
Lovers	24	31	77%	23	30	77%	24	30	80%	29	35	83%
Total	63	90	70%	63	86	73%	71	97	73%	87	102	85%

Table 3: Number of actions human readers said *do not* making sense (Unexp.) in all stories, the total number of actions in all stories (Total), and the corresponding percentage (Acc) of actions that *do make sense* in all stories.

Domain	Sabre			Halberd		
	Unexp.	Total	Acc	Unexp.	Total	Acc
BBall	27	118	77%	31	97	68%
Space	30	102	71%	28	91	69%
Gramma	14	109	87%	27	75	64%
Lovers	17	97	82%	21	76	72%
Total	88	426	79%	107	339	68%

Table 4: Number of subjects evaluated in each domain of the Action Believability study, before filtering (Raw), after removing subjects who completed the whole study in under 5 minutes (Processed), and after the additional per-domain filtering (Accepted).

Domain	Raw	Processed	Accepted
BBall	53	47	45
Space	53	47	45
Gramma	53	47	46
Lovers	53	47	42
Total	**212**	**188**	**178**

κ (1971) among all raters for each story. κ ranges from -1 (disagreement worse than chance) to 0 (chance agreement) to 1 (perfect agreement). It is undefined when all subjects mark all actions the same in a story, but since this represents perfect agreement, we used 1 for the only story where this occurred. On average, κ was positive but low—between 0.02 and 0.17—across domains and across planners. Because subjects showed poor consistency in marking actions, we must be skeptical of results from this experiment.

Table 3 shows that, across domains, subjects reported that about 2 in 10 Sabre actions did not make sense, whereas 3 in 10 Halberd actions did not make sense. These results are important after the previous discussion because, while Halberd's stories more often contain actions that do not make sense, Sabre's model is also imperfect.

We calculated Pearson's correlation coefficient between Halberd's accuracy according to Sabre's model (Table 2) and accuracy according to human subjects (Table 3) for these four domains, and while the correlation is positive ($r \approx 0.73$) it is not significant ($p \approx 0.27$), implying more research is needed into what makes actions in procedurally generated stories seem believable.

Table 5: The number of times a human reader preferred a story generated by Sabre or Halberd, with p-values for a one-tailed Binomial test adjusted for false discovery rate.

Domain	Pref. Sabre	Pref. Halberd	p-value
BBall	32	12	0.009
Space	22	25	1.000
Gramma	24	21	1.000
Lovers	12	31	0.014
Total	90	89	1.000

Solution Preference We ran a second experiment that more directly compares Sabre's solutions to Halberd's. We showed 50 subjects the description and initial state of each story domain in a random order, then we presented two different stories, one generated by Sabre and one by Halberd. Subjects chose which story they thought made more sense. Stories were chosen at random from the pool of stories by the planner in that domain, with two constraints. First, the stories had to be different from one another. Second, the stories had to have been generated with the same target author utility threshold. We randomized which story appeared first.

This study was also approved by the University of Kentucky's IRB under protocol number 115516. We also ran this study on Prolific and paid subjects $2 USD. Subjects took an average time of 11:10 to complete the study. We applied the same data quality criteria as the previous experiment.

Subjects were required to choose one of the two stories shown; there was no option to indicate a preference for neither. This experiment used an entirely separate pool of subjects from the Action Believability experiment above, with no overlap between the two studies. Table 6 reports, for each domain, the number of subjects whose data we used before filtering (Raw), after removing subjects who completed the whole study in under 5 minutes (Processed), and after the additional per-domain filtering.

Table 5 shows that across all domains subjects were evenly split between preferring Sabre and preferring Halberd. For each domain and for the aggregate, we performed a one-tailed Binomial test to determine whether the plan-

Table 6: Number of subjects evaluated in each domain of the Solution Preference study, before filtering (Raw), after removing subjects who completed the whole study in under 5 minutes (Processed), and after the additional per-domain filtering (Accepted).

Domain	Raw	Processed	Accepted
BBall	51	48	44
Space	51	48	47
Gramma	51	48	45
Lovers	51	48	43
Total	**204**	**192**	**179**

ner that was chosen more often was chosen significantly more often (p-values adjusted using the Bonferroni method to control for false discovery rate). In the Basketball domain, subjects significantly ($p < 0.05$) preferred Sabre, while in Lovers they significantly preferred Halberd. There was no significant trend across all domains.

Halberd's goal is to produce stories that are "good enough" to seem believable. We do not claim that it produces better stories than Sabre, and these results demonstrate that Sabre's stories are not significantly preferred over Halberd's.

Conclusions

In this paper, we described Halberd, a narrative planner that generates stories in a pre-defined logical domain. Its model of the author is traditional A* search through the space of action sequences (i.e. planning), while its model of character believability uses a Large Language Model to determine which actions seem reasonable. Our goal is to produce believable stories while avoiding the high cost of using planning to explain each action for each character.

Our investigation showed some limited success. Halberd using Qwen 3 MoE took about 70% longer than Sabre to generate solutions when running on the same hardware, but it reduced the number of nodes visited by more than 99%. As LLM speeds and hardware improve they will reduce Halberd's per-node cost, and we believe the massive reduction in the scope of the search can eventually enable faster planning and longer stories.

On average, Halberd chose actions that Sabre would be able to explain about 3 times out of 4, but human readers demonstrated that Sabre actions do not always make sense. They marked about 21% of actions in Sabre stories as not making sense, and about 32% of actions in Halberd stories. When asked to choose between a Sabre and Halberd story, they showed no clear preference for either. We believe these preliminary findings demonstrate an LLM can act as a "good enough" heuristic for when actions make sense in a story, though of course that judgment depends on the application.

Limitations

There are many ways we hope to improve on these experiments in the future. We would like to test story generation on more domains, with more LLMs, and with longer search limits. We would like to test more precisely the tradeoff between LLM speed and story quality. Specifically, we would like to measure how much reasoning models improve the quality of Halberd stories, though running large reasoning models is beyond the capabilities of our current hardware.

Perhaps the most critical limitation in our experiments is the inconsistency of human raters when marking actions in stories as unexplained. Without a reliable and consistent measure of when readers find an action believable, it will be difficult to develop a fine-grained measure of story quality.

Future Work

This work is part of a larger project investigating story generation with different and interchangeable models of the author and characters. We have defined four configurations of each:

- no model (chooses randomly)
- rule-based, like Comme il Faut (McCoy et al. 2014)
- LLM-based
- search-based

Sabre represents a search-based model of both the author and characters, whereas Halberd represents a search-based model of the author and an LLM-based model of the characters. This larger project will explore the cost and quality tradeoffs between these combinations. For now, we believe these results represent preliminary findings that support the viability of combining LLMs with search for neuro-symbolic story generation in pre-defined domains.

Acknowledgements

This material is based upon work supported by the U.S. National Science Foundation under Grant No. 2145153 and the U.S. Army Research Office under Grant No. W911NF-24-1-0195. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Army Research Office.

References

Bay 12 Games. 2022. Dwarf Fortress. [Video Game].

BioWare. 2007. Mass Effect. [Video Game].

Cavazza, M.; Charles, F.; and Mead, S. J. 2002. Character-based interactive storytelling. *IEEE Intelligent Systems special issue on AI in Interactive Entertainment*, 17(4): 17–24.

Dehn, N. 1981. Story generation after TALE-SPIN. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, volume 81, 16–18.

Dueifi, M.; and Eger, M. 2024. The Tomato Festival: Towards using ChatGPT for long-form discourse generation of plan-based narratives? In *Proceedings of the 14th workshop on Intelligent Narrative Technologies at the AAAI conference on Artificial Intelligence and Interactive Digital Entertainment conference*.

Evans, R.; and Short, E. 2013. Versu–a simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2): 113–130.

Farrell, R.; and Ware, S. G. 2020. Narrative planning for belief and intention recognition. In *Proceedings of the 16th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 52–58.

Farrell, R.; and Ware, S. G. 2024. Planning stories neurally. techrxiv:171085113.35202301.

Fikes, R. E.; and Nilsson, N. J. 1972. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3): 189–208.

Fisher, M. 2022. Narrative planning in large domains through state abstraction and option discovery. In *Doctoral Consortium at the 18th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 299–302.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5): 378.

Hoffmann, J.; and Nebel, B. 2001. The FF planning system: fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14: 253–302.

Kartal, B.; Koenig, J.; and Guy, S. J. 2014. User-driven narrative variation in large story domains using Monte Carlo Tree Search. In *Proceedings of the international conference on Autonomous Agents and Multiagent Systems*, 69–76.

Kybartas, B.; and Bidarra, R. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, 9(3): 239–253.

Larian Studios. 2023. Baldur's Gate III. [Video Game].

Lebowitz, M. 1985. Story-telling as planning and learning. *Poetics*, 14(6): 483–502.

Li, J.; Guo, X.; Wu, Y.; Lee, R. K.-W.; Li, H.; and Xie, Y. 2026. Lost in stories: Consistency bugs in long story generation by LLMs. arXiv:2603.05890.

Llama, T. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Maxis. 2000. The Sims. [Video Game].

McCoy, J.; Treanor, M.; Samuel, B.; Reed, A. A.; Mateas, M.; and Wardrip-Fruin, N. 2014. Social story worlds with Comme il Faut. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, 6(2): 97–112.

Meehan, J. R. 1977. TALE-SPIN, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91–98.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925.

Pan, Z.; Andronis, A.; Hayek, E.; Wilkinson, O. A. P.; Lasy, I.; Parry, A.; Gadney, G.; Smith, T. J.; and Grierson, M. 2025. Guiding generative storytelling with knowledge graphs. *International Journal of Human–Computer Interaction*, 1–23.

Porteous, J.; Cavazza, M.; and Charles, F. 2010. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Transactions on Intelligent Systems and Technology*, 1(2): 1–21.

Qwen, T. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Riedl, M. O.; and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1): 217–268.

Senanayake, L.; and Ware, S. G. 2025. Language models as narrative planning heuristics. In *Proceedings of the 20th international conference on the Foundations of Digital Games*.

Shirvani, A.; Farrell, R.; and Ware, S. G. 2018. Combining intentionality and belief: revisiting believable character plans. In *Proceedings of the 14th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 222–228. (full paper presented as poster).

Teutenberg, J.; and Porteous, J. 2013. Efficient intent-based narrative generation using multiple planning agents. In *Proceedings of the 2013 international conference on Autonomous Agents and Multiagent Systems*, 603–610.

Trichopoulos, G.; Alexandridis, G.; and Caridakis, G. 2023. A survey on computational and emergent digital storytelling. *Heritage*, 6(2): 1227–1263.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, Y.; Lin, J.; Yu, Z.; Hu, W.; and Karlsson, B. F. 2023. Open-world story generation with structured knowledge enhancement: A comprehensive survey. *Neurocomputing*, 559: 126792.

Ware, S. G.; and Farrell, R. 2023. A Collection of Benchmark Problems for the Sabre Narrative Planner. Technical report, Narrative Intelligence Lab, University of Kentucky.

Ware, S. G.; Garcia, E. T.; Fisher, M.; Shirvani, A.; and Farrell, R. 2022. Multi-agent narrative experience management as story graph pruning. *IEEE Transactions on Games*, 15(3): 378–387.

Ware, S. G.; and Siler, M. 2021. Sabre: a narrative planner supporting intention and deep theory of mind. In *Proceedings of the 17th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 99–106.

Ware, S. G.; and Young, R. M. 2014. Glaive: a state-space narrative planner supporting intentionality and conflict. In *Proceedings of the 10th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 80–86.

Yang, C.; Gross, M.; and Wampfler, R. 2025. Steering narrative agents through a dynamic cognitive framework for guided emergent storytelling. In *Proceedings of the 21st AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-Write: Towards better automatic storytelling. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 33, 7378–7385.

Ye, A.; Cui, C.; Shi, T.; and Riedl, M. O. 2022. Neural Story Planning. *arXiv preprint arXiv:2212.08718*.

Yoo, T.; and Cheong, Y.-G. 2024. Leveraging LLM-constructed graphs for effective goal-driven storytelling. In

Proceedings of the first international OpenKG Workshop on Large Knowledge-enhanced Models.

Young, R. M. 1999. Notes on the use of plan structures in the creation of interactive plot. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, 164–167.

Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, 37(1-2): 41–64.

Zwaan, R. A.; Magliano, J. P.; and Graesser, A. C. 1995. Dimensions of situation model construction in narrative comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(2): 386.