Narrative Knowledge Weaver: Narrative-Centric Retrieval-Augmented Reasoning for Long-Form Text Understanding

Anonymous ACL submission

Abstract

Long-form narrative QA requires reasoning over evolving story worlds rather than isolated passages: answers may depend on earlier goals, changing character states, social relations, causal triggers, temporal position, and later consequences. Existing retrieval and graph-augmented generation methods improve evidence access, but their units–chunks, entities, relations, summaries, or tool actions–do not directly encode how evidence functions in a story. We introduce **Narrative Knowledge Weaver** (NKW), a source-grounded framework that aligns textual evidence, atomic facts, canonical graph structure, entity profiles, interactions, episodes, and storylines. At query time, NKW uses text, graph, and narrative tools with post-retrieval reading skills to assemble evidence and audit actor, scope, polarity, state, and temporal constraints. Across STAGE, FairytaleQA, and QuALITY, NKW is strongest on screenplay-level story-world QA while remaining competitive on more passage-centered benchmarks. Ablations, question-type analyses, graph-asset statistics, and case studies show complementary benefits for character, scene, temporal, causal, and narrative-progression reasoning.

1 Introduction

Long-form narrative understanding supports screenplay analysis, production workflows, literary interpretation, and reading assistance. Unlike ordinary long documents, narratives are organized around scenes or chapters, recurring characters, changing states, and temporal-causal plot progression. Even with long-context LLMs, this remains challenging: the challenge is not only length, but recovering how evidence functions inside an evolving story world (Graesser and Franklin, 1990; Kintsch, 1988; Zwaan and Radvansky, 1998; Liu et al., 2024).

Retrieval-augmented generation (RAG) partially addresses this challenge by grounding model responses in external evidence (Lewis et al., 2020; Izacard and Grave, 2021). This makes it useful for locating relevant passages and reducing unsupported generation in long documents. Graph-based RAG organize retrieved content through entity, relation, or community-level abstractions (Edge et al., 2024; Guo et al., 2025; Gutiérrez et al., 2024; Zhu et al., 2025), while tool- and retrieval-control RAG methods adapt the retrieval process during generation (Jiang et al., 2023; Trivedi et al., 2023; Asai et al., 2024; Du et al., 2026). However, these methods are largely developed for factual, open-domain, or general long-document settings, whereas narratives require different evidence organization.

We identify two mismatches between general-purpose RAG and narrative QA. First, narrative evidence is functional, not only locational: the same passage may serve as premise, turning point, consequence, state update, or distractor. For a why-question, retrieval may find the decision scene while missing the earlier motivating interaction. Chunks, entities, relations, and community summaries locate material, but do not encode its story role. Second, characters and relations are dynamic. Narrative questions ask what a character believes, wants, knows, or feels at a plot moment, or how attitudes change after a scene, requiring evolving states and relationships rather than a flat entity profile. Together, these mismatches make it difficult for general-purpose RAG systems to assemble evidence that is not only relevant in content, but also appropriate in narrative function and dynamic character context.

To this end, We present **Narrative Knowledge Weaver** (NKW), a narrative-centric RAG system for these two mismatches. To represent functional evidence roles, NKW extracts source-grounded narrative assets and aggregates events, interactions, and occasions into higher-level episodes and storylines, so that evidence can be organized by its

contribution to plot progression rather than by passage similarity alone. To capture dynamic characters and relations, NKW separates stable identity from changing state. It first links names, aliases, pronouns, and scene-specific references to the stable narrative entity. It then uses source-grounded atomic facts to build time-sensitive character profiles. Thus, characters are represented not as static graph nodes, but as evolving narrative states anchored in textual evidence. Furthermore, NKW provides a channel-separated tool interface over text, graph, and narrative views, together with post-retrieval reading skills that assemble and audit evidence for actor, scope, polarity, and temporal position at inference time.

We evaluate NKW on on three long-form narrative QA datasets, STAGE (Tian et al., 2026), FairytaleQA (Xu et al., 2022), and QuALITY (Pang et al., 2022), across seven LLM backbones against Hybrid RAG, GraphRAG, LightRAG, HippoRAG, and A-RAG. Results show the largest benefits when questions require reasoning over evolving states, relations, temporal order, causal motivation, and plot progression. Ablations, STAGE question-type breakdowns, graph-asset statistics, case studies, and two downstream applications further show when the narrative structure contributes.

Our key contributions are:

- **A narrative-centric RAG system.** NKW represents long-form narratives through canonical entity graphs, source-grounded narrative assets, episode/storyline aggregation, and atomic-fact-based character profiling.
- **Inference tools for narrative evidence assembly.** NKW uses channel-separated access to text, graph, and narrative views, with post-retrieval reading skills to audit actor, scope, polarity, and temporal position.
- **Comprehensive evaluation and applications.** We evaluate NKW on STAGE, FairytaleQA, and QuALITY across seven LLM backbones, with ablations, question-type analyses, case studies, and two downstream narrative applications.

2 Related Work

2.1 Narrative and Long-Form Text Understanding

Narrative understanding requires reasoning over evidence distributed across an evolving story world. QuALITY (Pang et al., 2022), FairytaleQA (Xu et al., 2022), and STAGE (Tian et al., 2026) cover long-passage reading, fine-grained story questions, and full-screenplay story-world reasoning. These benchmarks test final answers, but they provide limited supervision for the intermediate story representation needed to recover those answers.

This representation gap is also reflected in prior work on event causality (Sun et al., 2024c) and event grounding (Jiayang et al., 2024), which emphasizes connecting what happens, why it happens, and where it is supported. Discourse comprehension (Graesser et al., 1994), construction-integration theory (Kintsch, 1988), and situation-model theory (Zwaan and Radvansky, 1998) similarly treat narrative meaning as evolving states, goals, causal links, and perspectives. NKW operationalizes this view as source-grounded layers for facts, character states, interactions, episodes, and storylines.

2.2 Graph Construction and Graph-Augmented Retrieval

Graph construction supplies useful backbone machinery but is not the main target of NKW. Document-level IE and coreference methods model entities, relations, events, and normalized mentions (Yao et al., 2019; Wadden et al., 2019; Lin et al., 2020; Barhom et al., 2019; Cattan et al., 2020); LLM-based methods such as EDC (Zhang and Soh, 2024) and Docs2KG (Sun et al., 2024b) extend this pipeline to flexible documents. These works primarily target extraction coverage, whereas NKW uses a stable graph backbone while making events, atomic facts, entity profiles, interactions, episodes, storylines, and provenance explicit narrative assets.

Graph-augmented RAG and graph-grounded reasoning expose structured evidence for retrieval and multi-step inference, including GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2025), HippoRAG (Gutiérrez et al., 2024), KG2RAG (Zhu et al., 2025), A-RAG (Du et al., 2026), RoG (Luo et al., 2024), ToG (Sun et al., 2024a), and KG-Agent (Jiang et al., 2025). NKW connects to this line but organizes retrieval around narrative dynamics: local facts, character profiles, interactions, event traces, episodes, and storylines.

3 The Narrative Knowledge Weaver Framework

We propose **Narrative Knowledge Weaver**, a framework for narrative-centric knowledge mod-

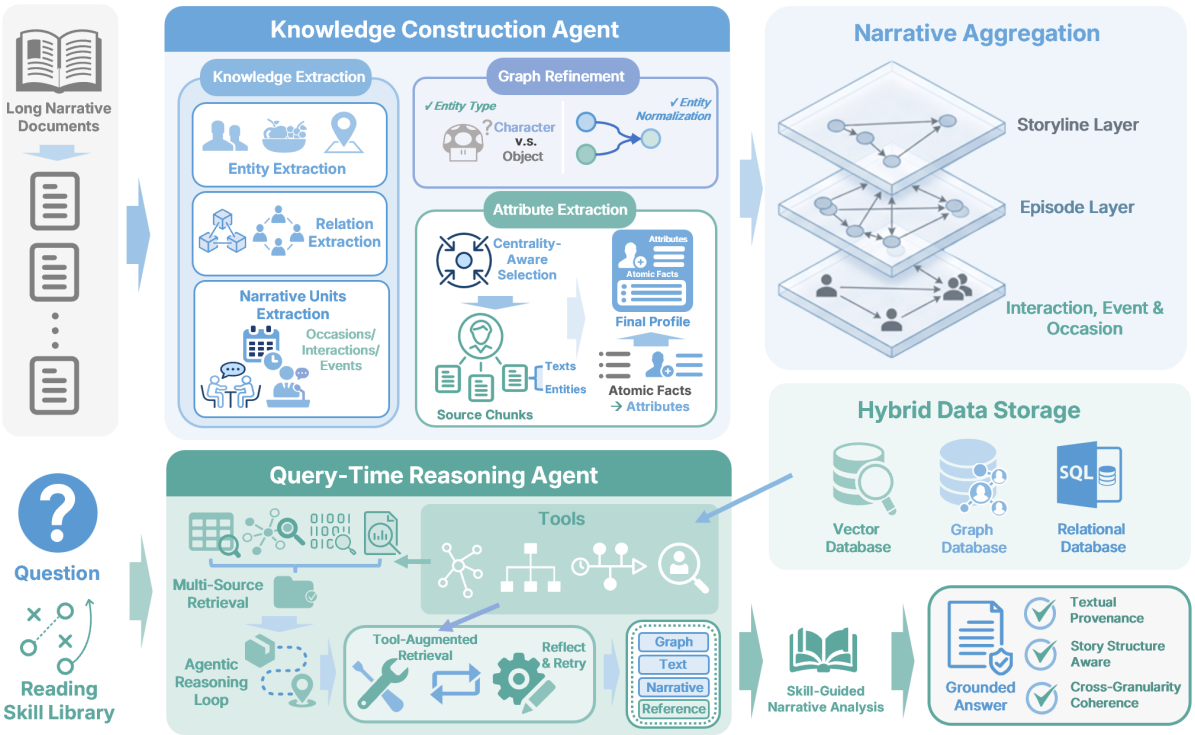

Figure 1: **Narrative-centric framework overview.** The framework separates construction-time graph building from query-time reasoning.

eling and reasoning over long-form text. Let $D = \{c_i\}_{i=1}^{n}$ denote the source chunks. NKW constructs a source-grounded asset bundle $\mathcal{B} = (G, \mathcal{U}, \mathcal{F}, \mathcal{P}, \mathcal{H}, \mathcal{X})$, where $G = (V, R)$ is a canonical entity–relation graph, $\mathcal{U}$ contains events, interactions, and occasions, $\mathcal{F}$ contains atomic facts, $\mathcal{P}$ contains entity profiles, $\mathcal{H}$ contains episode/storyline structures, and $\mathcal{X}$ indexes links back to source evidence. The system separates a construction-time agent that builds this bundle from a query-time reasoning agent that uses it with source evidence to answer questions.

3.1 Construction-Time Asset Building

The construction-time agent builds complementary source-grounded assets in a fixed order: a stable entity–relation graph backbone, narrative units and local evidence records, and then canonicalized entity profiles and graph-derived facts.

3.1.1 Canonical Entity–Relation Graph Backbone

NKW first builds a canonical graph backbone over stable narrative entities. Each chunk c_i is processed by an entity–relation extractor that emits entity rows $v = (\text{name}, \text{type}, \text{desc}, S_v)$ and relation rows $r = (u, v, \text{desc}, \text{keywords}, w, S_r)$, where S_v and S_r are source identifiers. The native entity schema is restricted to stable story referents, including characters, groups, locations, time points, objects, institutions, social roles, and concepts. Narrative process units such as events, interactions, occasions, episodes, storylines, and scene/chapter titles are not treated as native entity types in this backbone.

Relations in this backbone are binary edges between stable entities. They are stored as unordered entity pairs by default, with relation keywords used as topical descriptors rather than as a fixed relation ontology. N-ary statements are decomposed into binary edges, and missing endpoints are materialized as lightweight proxy entities so that the relation can still be source-grounded. During merge and upsert, same-name entities and same-pair relations are consolidated, long descriptions are compacted, and graph, vector, entity–chunk, and relation–chunk indexes are kept in sync. This backbone provides stable entity and relation grounding for later narrative assets.

3.1.2 Source-Grounded Narrative Asset Extraction

In parallel with the graph backbone, NKW extracts source-grounded narrative assets from each chunk. The base narrative units are *events*, which describe

actions, discoveries, decisions, arrivals, departures, physical processes, or state changes; *interactions*, which describe directed communication, conflict, cooperation, or influence between participants; and *occasions*, which describe stable situations, institutional constraints, social norms, background conditions, or scene-like contexts. The extractor also produces source-grounded atomic facts and entity attributes.

Each narrative unit remains linked to its source evidence. Interactions store structured metadata such as subject, object, interaction type, related event, and related occasion; events and occasions store participants, settings, temporal cues, and local context when available. These structured fields support later canonicalization and retrieval, while visible narrative content is rendered as clean natural language rather than as internal identifiers or storage metadata.

3.1.3 Entity Canonicalization and Evidence Enrichment

After merging the graph backbone, NKW canonicalizes salient mentions so that aliases and local descriptions of the same character, object, place, institution, or plan share one referent. The canonical map is applied to structured narrative fields–participants, interaction subject/object fields, atomic-fact subject/object fields, and owners of entity attributes–while natural-language evidence remains unchanged.

Only occasions are injected into the main graph, as contextual nodes linked to relevant participants or settings. Events and interactions stay in the narrative store and vector index for retrieval and episode/storyline construction, preserving source-grounded evidence without destabilizing the entity-relation graph. For entities with graph degree at least two in our experiments, NKW revisits linked source chunks, extracts entity-specific atomic facts about states, actions, goals, causal roles, mental states, and persistent relations, and derives a compact attribute dictionary and summary. Duplicate facts and overlapping attributes are merged, then stored in a semantic index for character, relation, and state reasoning.

3.2 Global Episode and Storyline Construction

Local entity aggregation is not sufficient for questions about plot development, motivation, or long-range consequence. NKW therefore aggregates local narrative units $\mathcal{U} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{O}$ into episodes using a coherence function $\phi(u, z)$ over shared participants, goals, conflicts, occasions, temporal continuity, causal dependency, semantic similarity, and source support:

$$\mathcal{Z} = \mathrm{Cluster}(\mathcal{U}; \phi), \quad \widehat{G}_{\mathcal{Z}} = \mathrm{SABER}(\mathcal{Z}, R_{\mathcal{Z}}). \tag{1}$$

Local episodes may retain document, scene, or chunk provenance, but they are not treated as storyline boundaries.

Storyline construction is global over the workspace. NKW predicts episode relations $R_{\mathcal{Z}} \subseteq \mathcal{Z} \times \{\textit{causes}, \textit{elaborates}, \textit{precedes}\} \times \mathcal{Z}$ using source order, shared participants, lexical similarity, shared graph neighbors, and episode-content similarity. SABER cleans the noisy episode graph into the DAG $\widehat{G}_{\mathcal{Z}}$: strongly connected components are broken by removing weak edges, and only after the DAG skeleton is stable are ambiguous triangle shortcuts sent for LLM adjudication. Storylines $\mathcal{L}$ are then extracted as episode chains through a trunk–branch view, with relations such as blocking, resolving, prerequisite, conflict, and parallel arcs predicted over $\mathcal{L}$. Thus, narrative units support episodes, episodes support storylines, and storylines provide compact handles for long-range temporal and causal reasoning. Appendix A.1 details the aggregation procedure.

3.3 Narrative-Grounded Retrieval and Reasoning

3.3.1 Evidence Views

The constructed asset bundle provides complementary evidence views over the same narrative: source passages for recall, atomic facts for compact local propositions, entity profiles for states and relations, event/interaction/occasion records for local structure, and episode/storyline layers for long-range progression. Each view remains linked to source evidence. When an entity profile contains many supporting atomic facts, query-time retrieval does not expose the full fact list by default. Instead, for an entity e and question q, NKW returns $\mathrm{TopK}_{k=5}\{f \in \mathcal{F}(e) : \mathrm{sim}(q, f)\}$ in our experiments. This prevents salient entities from overwhelming the evidence packet while preserving fine-grained source grounding.

3.3.2 Query-Time Tools

NKW exposes these views through a compact tool interface instead of a single flat retriever. The in-

terface provides text access for source evidence, graph access for entity and relation grounding, and narrative access for temporal, causal, and storyline progression. These tools add no external knowledge; they only route queries over source-grounded assets, allowing the agent to combine local text, graph structure, and narrative structure while preserving provenance. Appendix A.2 provides the detailed tool taxonomy used in inference.

At query time, retrieval is formulated as evidence assembly. Given tools $\mathcal{T}$, the agent decomposes a question q into requirements over local facts, entity states, interactions, temporal order, causal explanation, or plot progression, then builds an evidence packet $\mathcal{E}_q = \mathrm{Assemble}(q, \mathcal{B}, \mathcal{T})$. The final answer is generated from $\mathcal{E}_q$: structured signals guide navigation and constraint, while textual evidence remains the basis for answer support. Appendix A.2 gives the end-to-end QA procedure.

3.3.3 Post-Retrieval Reading Skills

Post-retrieval *reading skills* further guide interpretation of the assembled evidence. They are implemented as lightweight Reading Skill Cards: final-stage operators seeded by cognitive QA and narrative-comprehension theories, then calibrated to benchmark question patterns. They do not change the constructed graph or retrieve new evidence; instead, they audit how an existing evidence packet supports actor, scope, time, state, causal, discourse, and option-level constraints. Operationally, a selector injects one to three short card operators into the final-answer prompt to reduce wrong-actor, wrong-time, wrong-cause, wrong-state, and wrong-scope readings. Appendix A.4 summarizes these cards, and Appendix A.5 details the multiple-choice option-admission procedure.

4 Experimental Setup

Datasets. We evaluate on three narrative and long-form QA benchmarks. **STAGE** (Tian et al., 2026) contains 151 movie screenplays and 5,010 questions, with an average document length of 27,546 in our processed benchmark copy. The 151 screenplays include 109 English and 42 Chinese scripts, and the questions are annotated with six narrative reasoning types covering character, scene, temporal, causal, relation, and progression reasoning. **QuALITY** (Pang et al., 2022) contains 162 long English passages and 2,523 multiple-choice questions, with an average length of 4,115. **FairytaleQA** (Xu et al., 2022) contains 278 children's stories and 10,580 expert-authored questions, with an average length of 2,165. Document length is measured as whitespace-token word count for English text and character count for Chinese text in the processed benchmark copy. Appendix B.5 reports the STAGE type-level distribution and performance breakdown used in our analysis.

Baselines. We choose baselines to cover two axes central to our comparison: graph structure and tool-mediated retrieval. **Hybrid RAG** represents strong non-graph retrieval. **GraphRAG**, **LightRAG** (Guo et al., 2025), **HippoRAG** (Gutiérrez et al., 2024), and **A-RAG** (Du et al., 2026) cover recent graph- and tool-augmented RAG paradigms, including community summaries, entity-relation indexing, associative graph activation, and agent-controlled retrieval interfaces. **NKW** tests their combination through text, graph, and narrative tools over narrative-centric graph structures (Appendix A).

- All baselines use a **600-token chunk size**.
- Model-family comparisons use non-thinking Qwen3 and Llama-3.1 backbones, plus GPT-5.5 as a closed-source reference. Table names omit checkpoint suffixes for readability; Qwen3-30B refers to **Qwen3-30B-A3B-Instruct-2507**, Qwen3-235B refers to **Qwen3-235B-A22B-FP8**, and Llama rows use the corresponding Llama-3.1 Instruct checkpoints.
- Component ablations are run only with **Qwen3-235B**.

Evaluation Metrics. For STAGE and semantic-answer FairytaleQA items, we use a DeepSeek-V4 LLM-based evaluator that judges each system answer against the question and ground-truth answer. QuALITY is a multiple-choice benchmark, so correctness is determined by whether the selected option matches the gold option. Across tables, **Overall** denotes the benchmark's primary correctness metric, and **Pass@5** denotes whether any of five sampled answers is correct.

Ablation Protocol. Component ablations use the same Qwen3-235B backbone, decoding configuration, question sets, and benchmark-specific metrics as the full system. We remove one structural component at a time while holding the query-time reasoning budget fixed, isolating gains from narrative-centric structure rather than context size or sampling budget. For STAGE, we further report type-

Table 1: Main QA results on STAGE across backbones. Bold marks the best method separately for Overall and Pass@5 within each backbone row.

Backbone	Hybrid RAG		GraphRAG		LightRAG		HippoRAG		A-RAG		NKW (Ours)	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Qwen3-8B	**0.4216**	0.4637	0.2854	0.3124	0.3685	0.4329	0.2643	0.2890	0.2315	0.2741	0.3852	**0.5014**
Qwen3-30B	0.5144	0.5563	0.3028	0.4126	0.3874	0.5393	0.3186	0.3441	0.2980	0.3756	**0.5892**	**0.6643**
Qwen3-235B	0.6056	0.6465	0.3206	0.4790	0.4407	0.5988	0.4014	0.4301	0.3625	0.4862	**0.7012**	**0.8148**
Llama-3.1-8B	**0.3854**	0.4124	0.2916	0.3246	0.3419	0.4106	0.2719	0.2932	0.2146	0.2529	0.3721	**0.4854**
Llama-3.1-70B	0.5124	0.5547	0.3122	0.4691	0.3912	0.5216	0.4102	0.4415	0.3349	0.4214	**0.5629**	**0.6431**
Llama-3.1-405B	0.6112	0.6385	0.3525	0.4982	0.4527	0.5894	0.4874	0.5112	0.3952	0.5020	**0.6535**	**0.7429**
GPT-5.5	0.6216	0.6525	0.4024	0.5357	0.4784	0.6122	0.5118	0.5429	0.4216	0.5325	**0.6984**	**0.8114**

level effects over character, scene, temporal, causal, relation, and progression questions.

5 Experimental Results and Analysis

5.1 Question Answering

Main results. Table 1 reports the STAGE result block in the same backbone-by-access-structure format used by STAGE. We report it first because it is the most direct story-world benchmark, while treating all benchmarks as one narrative suite.

Across STAGE backbones, NKW obtains the best Pass@5 in every row and the best Overall score for all medium, large, and closed-source backbones. The main exceptions are the two smallest open-source backbones, where Hybrid RAG gives slightly higher one-shot Overall accuracy for Qwen3-8B and Llama-3.1-8B, while NKW still improves Pass@5. This suggests that structured narrative evidence is most useful when the backbone can reliably use multi-source evidence, and that repeated sampling reveals gains from the richer evidence space even when one-shot selection is less stable.

Appendix B.5 further breaks down the STAGE gains by six question types, and Appendix B.7 gives success and failure cases showing where NKW resolves baseline errors and where it can still confuse local triggers with salient downstream events.

Secondary benchmarks. Tables 2 and 3 record the supporting results on FairytaleQA and QuALITY, which complement STAGE by testing narrative comprehension over shorter stories and long-form multiple-choice passages.

FairytaleQA and QuALITY differ from STAGE: LightRAG is strong on these shorter or more passage-centered benchmarks, especially for one-shot QuALITY accuracy. NKW remains competitive and is strongest in several higher-capacity settings, with the best FairytaleQA Pass@5 for Qwen3-30B, Qwen3-235B, Llama-3.1-70B, Llama-3.1-405B, and GPT-5.5, and the best QuALITY Pass@5 for Qwen3-30B, Qwen3-235B, Llama-3.1-70B, Llama-3.1-405B, and GPT-5.5. Thus, narrative-structured evidence helps most when questions require multiple constraints, while lightweight retrieval remains effective for local or passage-centered cases. Appendix C isolates this pattern by varying the allowed follow-up rounds after mandatory first-pass recall: STAGE benefits strongly from deeper agentic interaction, whereas FairytaleQA and QuALITY show smaller, mixed effects. Appendix B gives the evaluator protocol and agreement analysis.

5.2 Tool-Use Analysis

To make NKW's multi-step reasoning auditable, we group query-time tool calls by text, graph, narrative, and document–chunk mapping channels. Each query first runs a forced multi-channel recall step before the agent decides whether additional tools are needed. This forced step calls the default graph–fact and narrative-semantic recall tools listed in Appendix A.2, followed by internal evidence assembly and source grounding; Appendix A.3 details the policy. We therefore report only the agent-selected follow-up budget: at most two rounds with up to five tool calls per round.

Figure 2 summarizes tool allocation across benchmarks; Appendix A.2 and Tables C.1–C.4 give the tool taxonomy, full distribution, autonomous follow-up depth, and follow-up-depth ablations. The patterns match benchmark demands: STAGE needs more graph and narrative tracing for scene order, character state, causal links, and plot progression; FairytaleQA uses more atomic facts and semantic search; and QuALITY emphasizes text-channel tools for option admission. Follow-up-depth ablations further show that NKW's agent-

Table 2: Supporting QA results on FairytaleQA. Bold marks the best method separately for Overall and Pass@5 within each backbone row.

Backbone	Hybrid RAG		GraphRAG		LightRAG		HippoRAG		A-RAG		NKW (Ours)	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Qwen3-8B	0.7324	0.7457	0.3121	0.3650	0.6792	**0.8229**	**0.7708**	0.8131	0.3873	0.4726	0.6525	0.8015
Qwen3-30B	0.8115	0.8239	0.3953	0.4626	**0.8563**	0.9241	0.8012	0.8124	0.4699	0.5302	0.8515	**0.9452**
Qwen3-235B	0.8406	0.8502	0.4487	0.5254	**0.8752**	0.9334	0.8256	0.8393	0.5871	0.7029	**0.8752**	**0.9545**
Llama-3.1-8B	**0.7216**	0.7314	0.3340	0.3898	0.7155	**0.7723**	0.7082	0.7197	0.4250	0.5211	0.6842	0.7514
Llama-3.1-70B	0.8129	0.8245	0.4635	0.5410	0.8015	0.8343	0.7956	0.8042	0.5527	0.6684	**0.8185**	**0.8612**
Llama-3.1-405B	0.8485	0.8426	0.5522	0.6149	0.8615	0.8940	0.8543	0.8698	0.6421	0.7456	**0.8654**	**0.9215**
GPT-5.5	0.8394	0.8552	0.5847	0.6420	**0.8926**	0.9214	0.8713	0.8859	0.6725	0.7850	0.8845	**0.9412**

Table 3: Supporting QA results on QuALITY. Bold marks the best method separately for Overall and Pass@5 within each backbone row.

Backbone	Hybrid RAG		GraphRAG		LightRAG		HippoRAG		A-RAG		NKW (Ours)	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Qwen3-8B	0.5525	0.5589	0.5129	0.6132	**0.7039**	**0.8018**	0.5347	0.6211	0.5030	0.6155	0.6845	0.7812
Qwen3-30B	0.6215	0.6282	0.5929	0.6924	0.7535	0.8256	0.5577	0.5600	0.5886	0.6468	**0.7582**	**0.8652**
Qwen3-235B	0.6552	0.6694	0.6659	0.7662	**0.8343**	0.8664	0.6544	0.6619	0.6564	0.7709	0.8296	**0.8989**
Llama-3.1-8B	0.5319	0.5359	0.5343	0.6318	**0.6425**	**0.6853**	0.4415	0.4483	0.4851	0.5985	0.6120	0.6639
Llama-3.1-70B	0.6346	0.6409	0.6726	0.7749	0.7852	0.8129	0.6124	0.6231	0.6219	0.7352	**0.7883**	**0.8541**
Llama-3.1-405B	0.6421	0.6548	0.7182	0.8125	**0.8216**	0.8541	0.7027	0.7150	0.7111	0.8197	0.8173	**0.8914**
GPT-5.5	0.6417	0.6552	0.7356	0.8411	0.8549	0.8914	0.7241	0.7384	0.7329	0.8458	**0.8617**	**0.9231**

Table 4: Component ablations across benchmarks using Qwen3-235B. Gray subscripts indicate the absolute performance drop compared to the full system.

Variant	STAGE		FairytaleQA		QuALITY	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Full system	0.7012	0.8148	0.8752	0.9545	0.8296	0.8989
w/o attributes	0.6715 $\downarrow 0.0297$	0.7721 $\downarrow 0.0427$	0.8512 $\downarrow 0.0240$	0.9215 $\downarrow 0.0330$	0.7816 $\downarrow 0.0480$	0.8712 $\downarrow 0.0277$
w/o ep./storyline agg.	0.6481 $\downarrow 0.0531$	0.7415 $\downarrow 0.0733$	0.8604 $\downarrow 0.0148$	0.9312 $\downarrow 0.0233$	0.7535 $\downarrow 0.0761$	0.8423 $\downarrow 0.0566$
w/o graph refinement	0.6651 $\downarrow 0.0361$	0.7681 $\downarrow 0.0467$	0.8488 $\downarrow 0.0264$	0.9156 $\downarrow 0.0389$	0.7681 $\downarrow 0.0615$	0.8589 $\downarrow 0.0400$
w/o graph tools	0.6395 $\downarrow 0.0617$	0.7150 $\downarrow 0.0998$	0.8445 $\downarrow 0.0307$	0.8924 $\downarrow 0.0621$	0.7126 $\downarrow 0.1170$	0.8014 $\downarrow 0.0975$
w/o reading skills	0.6541 $\downarrow 0.0471$	0.7551 $\downarrow 0.0597$	0.8556 $\downarrow 0.0196$	0.9230 $\downarrow 0.0315$	0.7428 $\downarrow 0.0868$	0.8315 $\downarrow 0.0674$

selected follow-up rounds bring the largest gains on STAGE. On FairytaleQA and QuALITY, the effect is smaller and appears mainly in the Pass@5 ceiling rather than consistent one-shot Overall accuracy, which is expected for more passage-centered tasks.

Post-retrieval reading skills show a similar benchmark-specific pattern: STAGE triggers more quest reasoning and character mind-state tracking,

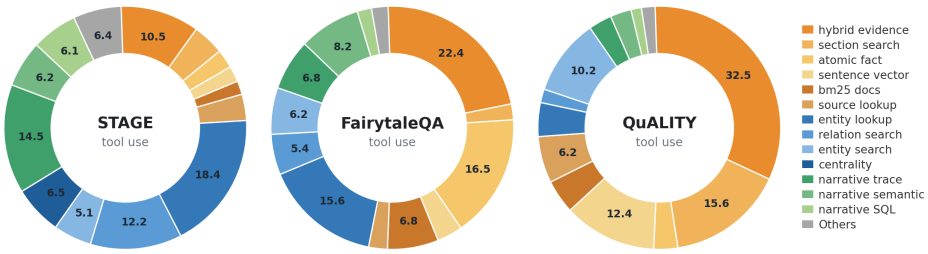

Figure 2: **Fine-grained tool-use distribution across benchmarks.** Donut charts summarize query-time tool invocations grouped into text, graph, narrative, and administrative mapping channels. STAGE uses more graph and narrative tracing, FairytaleQA relies more on atomic facts and semantic search, and QuALITY primarily uses text extraction tools.

FairytaleQA relies more on literal fact localization, and QuALITY is dominated by multiple-choice option admission. The full distribution is reported in Appendix A.4.

5.3 Ablation Study

We ablate core components across STAGE, FairytaleQA, and QuALITY. STAGE further supports question-type analysis, linking each component to the reasoning behavior it is designed to support.

Table 4 isolates five choices: entity attributes for character states, goals, affiliations, and persistent relations; episode/storyline aggregation for long-range progression; graph refinement for normalized entities and consolidated relations; graph-channel tools for structured access beyond flat retrieval; and reading skills for post-retrieval answer selection. The *w/o graph tools* variant removes only these graph-channel tools from the available tool set; it is distinct from the zero-follow-up setting in Appendix C, which disables all autonomous follow-up rounds after the mandatory first-pass recall.

Aggregate ablations test whether each component contributes beyond one dataset. Appendix B.6 breaks STAGE down by question type in Table B.4, connecting local components such as entity attributes to character and role-relation questions and global components such as episode/storyline aggregation to temporal and narrative-progression reasoning.

Appendix A gives method and query-time reasoning details, Appendix B gives evaluation details and additional STAGE analyses, Appendix C reports tool-use statistics, and Appendix D provides downstream application details.

5.4 Downstream Application

The screenplay setting in STAGE evaluates not only plot comprehension, but also questions whose evidence concerns scene structure, character presence, objects, locations, and production-relevant state changes. This makes it a natural testbed for whether source-grounded narrative representations can be reused as production-facing assets rather than only as QA evidence. We illustrate this with production continuity checking. Because NKW grounds entities, relations, and narrative units to source chunks, each asset can be mapped back to the scene in which it appears; because the schema separates characters, objects, locations, occasions, and narrative units, an agent can compare whether two scenes are likely to share a production setup.

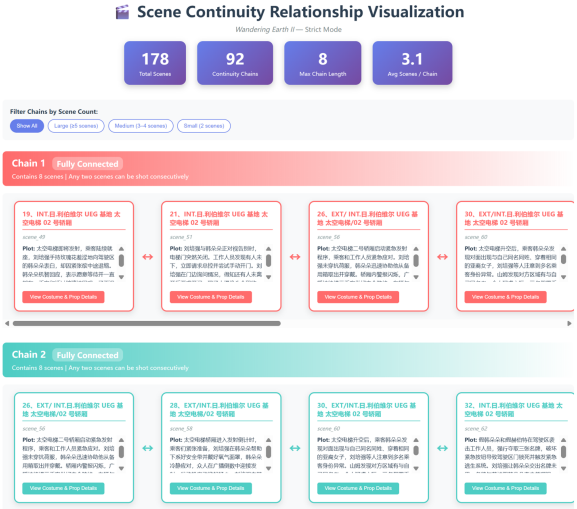

Figure 3: **Continuity chain visualization.** Chains link scenes that can plausibly share a production setup in an example from *The Wandering Earth II* (Gwo, 2023), with scene summaries and source-grounded narrative cues shown for review.

The resulting continuity chains can help production teams identify compatible scenes for reusing sets, dressing, props, costumes, and makeup baselines. Figure 3 shows the resulting view; the full pipeline and prompts are in Appendix D.1, with character-state tracking in Appendix D.2.

6 Conclusion

We presented **Narrative Knowledge Weaver**, a framework for source-grounded narrative knowledge modeling and long-form QA. NKW aligns factual, graph, entity-profile, interaction, episode, and storyline layers so query-time reasoning can move between local evidence and higher-level narrative progression. It targets questions whose answers depend not on a single passage, but on how events, character states, relations, and plot threads evolve across the story. Across STAGE, FairytaleQA, and QuALITY, results and ablations show that narrative-centric structure, graph-channel tools, and reading skills improve multi-step story understanding while preserving source links. Overall, the findings suggest that narrative QA benefits from modeling how evidence functions inside a story, not only where relevant passages or entity relations occur, while future work should improve efficiency, stability, domain coverage, and source-grounded auditability.

Limitations

NKW is designed for long-form narrative understanding, and its advantages are strongest when questions require evidence across character states, events, temporal order, causal relations, and story progression. For shorter or more passage-centered questions, the additional narrative structure can be less beneficial than lightweight retrieval, as reflected by the strong performance of some baselines on FairytaleQA and QuALITY. The method also incurs higher construction and query-time cost than flat retrieval because it builds source-grounded narrative layers and performs multi-step evidence assembly.

The quality of the constructed asset bundle depends on upstream extraction and normalization. If events, entity identities, or source-grounded facts are missed or incorrectly merged, later retrieval and reasoning may inherit these errors. Although final answers are tied back to source evidence, the query-time agent can still select different evidence paths across runs, leading to variance in single-sample answers. Reading skills should also be viewed as an exploratory component: we evaluate their aggregate contribution, but do not provide a detailed study of skill design, selection, or interaction effects. Our evaluation focuses on QA over existing narrative datasets; we do not claim that the same configuration is sufficient for all narrative tasks, domains, or languages.

Ethics Statement

This work uses existing scientific artifacts: public narrative and long-context QA datasets, language-model backbones, and retrieval or graph-RAG baselines. We cite the creators of the datasets and methods used in the experimental setup and related work. We follow the licenses or access terms of the original artifacts and use them for research evaluation; we do not redistribute original benchmark texts or proprietary model weights as part of this paper. The datasets consist of fictional or published narrative texts and benchmark questions; we do not recruit human participants or annotators, collect new human-subject data, or collect personally identifying information from participants. Some source narratives may contain fictional violence, offensive language, stereotypes, or other sensitive content inherited from the original works. Our use is limited to research evaluation of narrative understanding systems.

The main risk of NKW is misuse as a tool for generating unsupported interpretations of narrative sources or for over-trusting automatically constructed story graphs. We mitigate this risk by keeping graph, factual, episode, and storyline records linked to source evidence and by requiring final answers to be grounded in original references. The system is intended for research and analysis settings rather than autonomous decision making about real people. We used AI assistants for research support, coding, and writing assistance; all paper content, experiments, and final claims were reviewed and edited by the authors.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling. *arXiv preprint arXiv:2009.11032*.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Shaohan Wang, Pengyu Wang, Xiaorui Wang, and Zhendong Mao. 2026. A-RAG: Scaling agentic retrieval-augmented generation via hierarchical retrieval interfaces. *arXiv preprint arXiv:2602.03442*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Arthur C. Graesser and Stanley P. Franklin. 1990. QUEST: A cognitive model of question answering. *Discourse Processes*, 13(3):279–303.

Arthur C. Graesser, Sallie E. Gordon, and Lawrence E. Brainerd. 1992. QUEST: A model of question answering. *Computers & Mathematics with Applications*, 23(6–9):733–745.

Arthur C. Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761. Association for Computational Linguistics.

Bernal J. Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*.

Frant Gwo. 2023. The wandering earth ii. Film.

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

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2025. KG-Agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9505–9523, Vienna, Austria. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992. Association for Computational Linguistics.

Cheng Jiayang, Lin Qiu, Chunkit Chan, Xin Liu, Yangqiu Song, and Zheng Zhang. 2024. EventGround: Narrative reasoning by grounding to eventuality-centric knowledge graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6622–6642, Torino, Italia. ELRA and ICCL.

Young-Suk Grace Kim, Rebecca Dore, Minkyung Cho, Roberta Golinkoff, and Steven J. Amendum. 2021. Theory of mind, mental state talk, and discourse comprehension: Theory of mind process is more important for narrative comprehension than for informational text comprehension. *Journal of Experimental Child Psychology*, 209:105181.

Walter Kintsch. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

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

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Jean M. Mandler and Nancy S. Johnson. 1977. Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, 9(1):111–151.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024a. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Qiang Sun, Yuanyi Luo, Wenxiao Zhang, Sirui Li, Jichunyang Li, Kai Niu, Xiangrui Kong, and Wei Liu. 2024b. Docs2kg: Unified knowledge graph construction from heterogeneous documents assisted by large language models. *arXiv preprint arXiv:2406.02962*.

Yidan Sun, Qin Chao, and Boyang Li. 2024c. Event causality is key to computational story understanding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3493–3511, Mexico City, Mexico. Association for Computational Linguistics.

Qiuyu Tian, Zequn Liu, Yiding Li, Fengyi Chen, Youyong Kong, Fan Guo, Yuyao Li, Jinjing Shen, Zhijing Xie, Yiyun Luo, Xin Zhang, and Yingce Xia. 2026. STAGE: A full-screenplay benchmark for reasoning over evolving stories. *arXiv preprint arXiv:2601.08510*.

Tom Trabasso and Paul van den Broek. 1985. Causal thinking and the representation of narrative events. *Journal of Memory and Language*, 24(5):612–630.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10014–10037. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9820–9836, Miami, Florida, USA. Association for Computational Linguistics.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8912–8924. Association for Computational Linguistics.

Rolf A. Zwaan, Mark C. Langston, and Arthur C. Graesser. 1995. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological Science*, 6(5):292–297.

Rolf A. Zwaan and Gabriel A. Radvansky. 1998. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185.

Appendix Table of Contents

- **Appendix A: Method Details and Query-Time Reasoning**
  - A.1 Narrative Aggregation Details (§A.1)
  - A.2 End-to-End QA Procedure and Tool Interface (§A.2)
  - A.3 Evidence Binding and Tool Policy (§A.3)
  - A.4 Post-Retrieval Reading Skills (§A.4)
  - A.5 Multiple-Choice Option Admission (§A.5)
- **Appendix B: Evaluation Details and Additional Results**
  - B.1 Benchmark Suite Overview (§B.1)
  - B.2 LLM-Evaluated Correctness Prompt (§B.2)
  - B.3 Evaluator Agreement (§B.3)
  - B.4 Evaluator Stochasticity (§B.4)
  - B.5 STAGE Question-Type Breakdown (§B.5)
  - B.6 STAGE Question-Type Ablation (§B.6)
  - B.7 STAGE Case Analysis (§B.7)
  - B.8 Graph Asset Statistics and Computational Cost (§B.8)
  - B.9 SABER Setting Ablation (§B.9)
- **Appendix C: Query-Time Tool-Use Analysis**
  - C.1 Fine-Grained Tool-Use Distribution (§C.1)
  - C.2 Agentic Follow-Up Depth (§C.2)
  - C.3 Follow-Up Depth on STAGE (§C.3)
  - C.4 Follow-Up Depth on Passage-Centered Benchmarks (§C.4)
- **Appendix D: Downstream Applications**
  - D.1 Production Continuity Checking (§D.1)
  - D.2 Character State Tracking (§D.2)
- **Appendix E: Responsible NLP Checklist Details**
  - E.1 Artifacts, Licenses, and Intended Use (§E.1)
  - E.2 Reproducibility and Decoding Settings (§E.2)

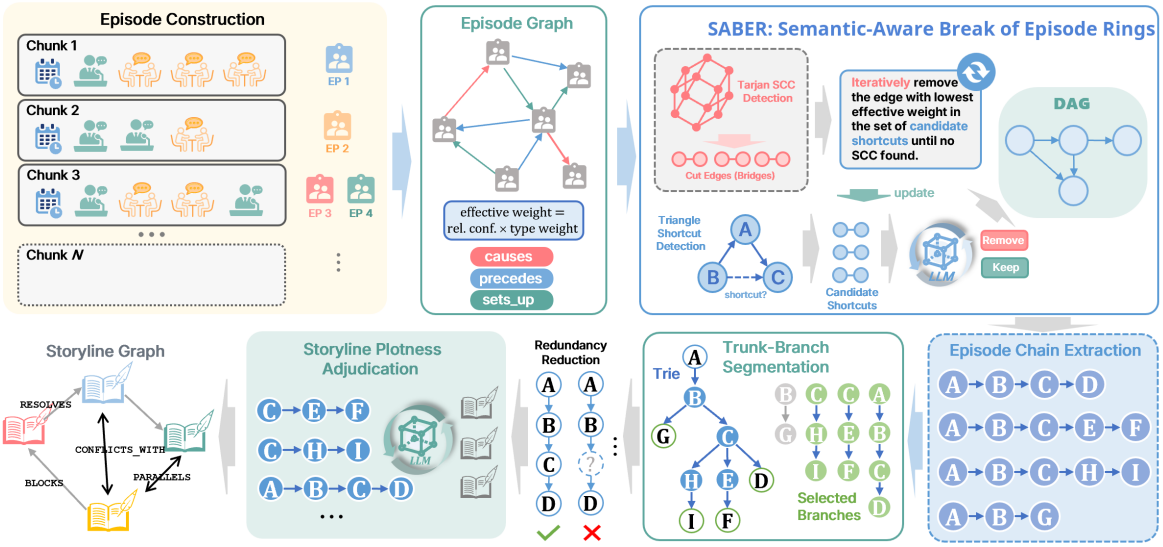

Figure A.1: **Narrative aggregation from local units to storylines.** NKW first assembles events, interactions, and occasions into coherent episodes, extracts and cleans episode relations into an acyclic progression graph, derives episode chains through trunk–branch segmentation, and finally induces storyline nodes, support edges, and storyline relations.

A Method Details and Query-Time Reasoning

This appendix collects method details that support the construction-time narrative hierarchy and the query-time QA pipeline. It first expands the episode/storyline aggregation procedure, then describes evidence assembly, query-time tool interfaces, evidence binding, reading skills, and multiple-choice option admission.

A.1 Narrative Aggregation Details

Figure A.1 expands the narrative aggregation module summarized in Section 3.2. The module starts from three kinds of base narrative units: *events*, which represent actions, discoveries, changes, conflicts, or plan outcomes; *interactions*, which represent directed exchanges among participants such as warning, deceiving, helping, threatening, persuading, or hiding information; and *occasions*, which represent stable narrative contexts such as scenes, social situations, institutional constraints, or recurring settings. These units provide the local evidence from which larger narrative structures are induced.

Episode assembly. An episode is a coherent mid-level narrative segment, not a single chunk label. It may be local to one scene or span multiple sections when its units jointly express the same goal, conflict, action chain, information change, recurring occasion, or participant-centered situation. Candidate episode clusters are assembled using shared participants, shared goals or conflicts, causal dependency, temporal continuity, recurring occasions, semantic coherence, source-span adjacency, and explicit evidence links. Each episode records its child event, interaction, and occasion units, main participants, source spans, central goal or conflict when available, and source-grounded evidence. This preserves the path from an episode back to the local units and passages that support it.

Episode relation graph and DAG cleaning. After episode assembly, NKW extracts relations among episodes. The main directed relations are *precedes*, *causes*, and *elaborates*; contrastive relations are kept as auxiliary semantic links rather than as main progression edges. Each relation has a confidence score and a type-dependent weight, producing an effective edge weight used during graph cleaning. The raw episode graph can contain cycles and redundant shortcuts. NKW therefore converts it into a clean episode DAG using SABER, a semantic-aware procedure for breaking cycles and pruning redundant shortcuts in event-relation graphs. The procedure marks triangle shortcut candidates, repeatedly detects strongly connected components, removes the weakest edge inside each cyclic component while preferring edges already marked as redundant shortcuts, and then re-checks remaining shortcut candidates on the resulting

DAG. Only the remaining ambiguous shortcuts are passed to an LLM adjudicator, which decides whether a direct edge is fully covered by an indirect path.

Algorithm 1 SABER Episode DAG Cleaning

Require: Noisy episode graph $G = (V, E)$; edge type $\tau(e)$; confidence $c(e)$; type weight w_τ
Ensure: Clean episode DAG G_{dag}
Compute effective weight $s(e) = c(e) \cdot w_{\tau(e)}$ for every edge.
Mark triangle-shortcut candidates (u, v) when a path $u \to x \to v$ exists.
while G contains a strongly connected component C with $|C| > 1$ **do**
  Let $E_C = \{(u, v) \in E : u, v \in C\}$.
  Remove the edge in E_C with minimum $s(e)$, preferring marked shortcut candidates.
  Recompute strongly connected components.
end while
Recompute shortcut candidates on the acyclic graph.
for all ambiguous direct edge (u, v) with an indirect path $u \rightsquigarrow v$ **do**
  Ask whether (u, v) is REDUNDANT, NECESSARY, or UNCERTAIN.
  if the decision is REDUNDANT and source support is preserved **then**
    Remove (u, v).
  end if
end for
return the resulting DAG G_{dag}.

The LLM is therefore used only after the graph skeleton is acyclic. It cannot create episodes, add new edges, reverse episode order, or remove a support link unless the direct relation is judged redundant with respect to an already existing indirect path.

Chain extraction and storyline construction. Storyline candidates are extracted from the clean episode DAG as episode chains. We use a trunk–branch view rather than relying only on a longest path: source-to-sink paths are enumerated under bounded depth and path-count constraints, inserted into a trie, and segmented into shared trunks and meaningful branches. This representation exposes common narrative prefixes and later divergences, which is useful for questions about plot alternatives, delayed consequences, and recurring arcs. A storyline is then induced from a selected episode chain as a higher-level narrative arc with a name, description, impact, key characters, source spans, and child episode links. Storyline support edges are hierarchical containment links from storylines to episodes, whereas storyline relations describe high-level interactions among arcs, such as enabling, blocking, resolving, conflicting with, or paralleling another storyline.

Algorithm 2 Trunk–Branch Storyline Segmentation

Require: Clean episode DAG G_{dag}; maximum chain length L; overlap threshold ρ; minimum new episodes m
Ensure: Storyline nodes and support edges
Enumerate bounded source-to-sink episode paths in G_{dag}.
Insert each path into a trie whose nodes are episode ids.
Extract candidate chains from shared trunks, branch continuations, and terminal paths.
Split candidates longer than L at branch or low-support cut points.
Rank candidates by coverage, coherence, source support, and new episode contribution.
Initialize selected chain set $\mathcal{S} \leftarrow \emptyset$.
for all candidate chain P in ranked order **do**
  if $\max_{Q \in \mathcal{S}} \text{overlap}(P, Q) \leq \rho$ and P contributes at least m new episodes **then**
    Add P to $\mathcal{S}$.
  end if
end for
for all $P \in \mathcal{S}$ **do**
  Induce a storyline with name, summary, impact, entities, source spans, and child episode ids.
  Add support edges from the storyline to its child episodes.
end for
Predict closed-set storyline relations only between selected storylines.

The complete aggregation path is therefore: source evidence to base narrative units, base units to episodes, episodes to a clean episode DAG, and episode chains to storylines. This hierarchy is designed to support both high-level narrative retrieval and source-grounded answering: a query can retrieve a storyline or episode for plot-level orientation, then follow support links back to the exact local units and passages needed for verification.

A.2 End-to-End QA Procedure and Tool Interface

Given a benchmark question and its associated document or screenplay, NKW answers the question through a fixed evidence-assembly and finalization pipeline. This section first summarizes the pipeline, then details the evidence-board budget and the query-time tool interface exposed to the reasoning agent.

Pipeline overview.

1. **Query decomposition.** The evaluation wrapper submits the question, document identifier, answer format, and retrieval budget to NKW. The query is decomposed into retrieval intents over canonical entities, relations, atomic facts,

and narrative units, so the first evidence pass can address both local factual grounding and higher-level narrative context.

2. **Forced graph–fact and narrative recall.** Before the tool agent starts, NKW runs a mandatory first-pass recall step over the canonical graph, atomic facts, and semantic narrative units. In the query-time tool taxonomy, this step consists of:

   `entity_search`; `entity_lookup`; `relation_search`; `atomic_fact_search`; `narrative_semantic_search`.

   This mandatory step is excluded from agent-selected tool-use statistics.

3. **Candidate evidence assembly.** The system enriches retrieved entities with attributes and query-relevant atomic facts, keeps relation candidates from both entity- and relation-centered retrieval, attaches available source-text evidence through provenance indices, and prepares source chunks for grounding. These assembly and indexing steps are not counted as separate tool calls.

4. **Tool-agent refinement.** The tool agent is inserted after first-pass retrieval and before final prompt construction. Each round scores the current packet, optionally builds a pruned packet, checks answerability, and stops without tool calls when the packet is already answerable. Otherwise, it plans at most five tool calls for the current evidence gaps. The default budget allows at most two rounds. This fail-closed policy prevents an already sufficient first-pass packet from being degraded by unnecessary tool use.

5. **Evidence-board construction.** Returned tool evidence can add entities, relations, narrative items, or compact tool-text evidence, subject to channel-specific edit switches and added-evidence budgets. Dynamic packing then constructs the final prompt sections for entities, relations, narrative items, source text chunks, optional tool-text evidence, and references. Optional LLM evidence refinement compresses selected entity, relation, and narrative items; it is an evidence compressor, not an additional retrieval tool.

6. **Reading-skill selection and finalization.** Reading skills are selected after retrieval and packing, with at most three skills injected into the final prompt. They do not retrieve new evidence; they instruct the final reader how to compare the assembled packet for option admission, causal/quest reasoning, character mental state, theme or implication, and literal fact localization. The answer model then generates the final response.

Evidence-board budgeting. The final evidence board uses a cluster-merge-balanced allocator for entity, narrative, and relation items. Each candidate item is scored using query similarity, lexical overlap, source overlap, support from selected reference chunks, token length, channel type, and relation-pair identity. Highly similar items or items grounded in the same source evidence are merged into clusters; the allocator then selects a small number of representatives per cluster rather than applying a flat top-k over all structured evidence. Items already covered by selected source chunks are downweighted to reduce repetition, relation pairs are deduplicated, and the board preserves floors for entity and narrative evidence so that relation items cannot consume the structured-evidence budget. In the stable setting, the board admits up to 8 entity items, 8 narrative items, and 16 relation items, with floors of 5 for entity and narrative items and no hard floor for relations.

Entity entries are also budgeted internally. Before an entity is shown in the final evidence board, NKW attaches query-relevant entity attributes and at most five atomic facts by default. Candidate atomic facts for the entity are ranked by question similarity, then only the top facts are retained. This keeps fine-grained source-grounded propositions available without allowing repeated local facts to dominate the prompt. We also implement dynamic and cluster-aware atomic-fact budgets, but the reported experiments use the stable per-entity top-5 setting.

Tool interface. The tool-agent refinement step exposes a compact tool surface rather than a large collection of low-level implementation calls. The tools are organized around textual evidence, graph evidence, narrative-structure evidence, and document–chunk mapping. The agent can combine these channels when a question requires local wording, entity or relation grounding, compact factual propositions, source index resolution, and story-level progression at the same time.

Table A.2 lists the tool names exposed to the query-time agent. Lower-level retrieval, graph, and

Channel	Evidence units	Role in query-time reasoning
Text	Passages, sentences, sections, source lookups, and atomic facts	Retrieves source-grounded wording and compact factual propositions that directly support answer candidates.
Graph	Entities, relations, central nodes, and local graph neighborhoods	Finds salient narrative entities, attributes, relations, and local graph neighborhoods relevant to the question.
Narrative	Events, interactions, episodes, storylines, and structured narrative queries	Recovers temporal progression, recurring plot threads, and multi-hop narrative evidence beyond isolated chunks.
Other	Document titles, chunk identifiers, and document–chunk index mappings	Resolves source-index metadata, such as mapping document titles to chunk identifiers or counting chunks within a document.

Table A.1: Query-time tool channels used by the reasoning agent.

Channel	Tool name	Purpose
Text	`hybrid_evidence_search`	Blends lexical and semantic recall for first-pass source evidence.
	`section_search`	Locates or retrieves evidence from narrative sections, scenes, or chapters.
	`atomic_fact_search`	Retrieves compact source-grounded factual propositions for local states, actions, causes, functions, and rules.
	`vdb_search_sentences`	Retrieves fine-grained semantic evidence for paraphrased local facts or implications.
	`bm25_search_docs`	Retrieves source chunks with exact or rare surface terms.
	`source_lookup`	Fetches original source passages after another tool returns source identifiers.
Graph	`entity_lookup`	Retrieves attributes, relations, and source locations for a canonical entity.
	`entity_search`	Finds canonical entity or concept candidates before focused lookup.
	`relation_search`	Searches semantic relation facts, neighbors, entity pairs, or co-section relations.
	`top_k_by_centrality`	Identifies globally salient entities or concepts for follow-up verification.
Narrative	`narrative_trace_search`	Searches event–episode–storyline traces for temporal, causal, and trajectory reasoning.
	`narrative_semantic_search`	Performs fuzzy semantic search over events, interactions, episodes, storylines, and scene notes.
	`narrative_sql_search`	Looks up structured scene notes, dialogue, action interactions, and document-linked occurrences.
Other	Others	Maps between document titles and chunk identifiers, including document-level chunk listing or counting queries.

Table A.2: Query-time tool names exposed to the NKW reasoning agent. The names align with the tool legend in Figure 2; the interface is grouped into text, graph, narrative, and document–chunk mapping channels.

database operations are encapsulated behind this interface.

A.3 Evidence Binding and Tool Policy

NKW separates retrieval signals from final evidence. Tool outputs may contain graph summaries, atomic facts, narrative-unit summaries, or candidate passages, but the final answer is grounded in the original source references recovered from those outputs. When several tools point to the same source, the source receives stronger support in the final evidence packet; when a summary conflicts with the source passage, the source passage takes precedence. This policy is important because graph and narrative tools are deliberately compressed views of the story, while STAGE and QuALITY answers often turn on exact scene wording, scope, or option polarity.

The first retrieval round is a forced graph–fact and narrative-semantic recall step designed to establish canonical entities, relations, compact factual support, and semantically relevant narrative units before the agent chooses any follow-up tools. The reported tool calls are the same mandatory recall set listed in Appendix A.2. The forced round is excluded from the agent-selected tool-use statistics; follow-up calls refine the evidence by checking exact passages, resolving entity identity, or expanding a narrative trace. In this design, atomic facts and narrative traces play different roles. Atomic facts are compact local propositions about actions, attributes, relations, or causes that remain tied to source passages. Narrative traces instead organize events, episodes, storylines, and their relations, making them better suited for temporal, causal, and trajectory questions.

We also distinguish two narrative retrieval modes. Trace-oriented retrieval follows structured event–episode–storyline links and is used when the question asks about order, motivation, consequence,

or a character/object trajectory. Semantic narrative retrieval searches narrative units by topical similarity and is more useful when the query is underspecified or when the desired unit is a paraphrase of the question. Keeping these modes separate reduces a common failure mode in which a topically related scene is retrieved but the answer requires a specific earlier or later narrative step.

A.4 Post-Retrieval Reading Skills

The query-time agent also uses reading skills after evidence has been retrieved and before the final answer is generated. These skills are not retrieval tools and do not issue additional retrieval queries. They are compact evidence-reading modules that tell the agent what kind of reading operation the question requires and how retrieved evidence should be compared.

We implement these modules as literature-seeded Reading Skill Cards. Each card is distilled from a deterministic seed into a scoped operator with trigger signals, evidence-use rules, common failure modes, and a compact runtime instruction. The seeds draw on cognitive question-answering and narrative-comprehension theories, including QUEST-style answer paths (Graesser and Franklin, 1990; Graesser et al., 1992), event-indexed situation models (Zwaan et al., 1995; Zwaan and Radvansky, 1998), causal-network accounts of story events (Trabasso and van den Broek, 1985), story grammar (Mandler and Johnson, 1977), and theory-of-mind work on narrative comprehension (Kim et al., 2021). We use representative question patterns from STAGE, FairytaleQA (Xu et al., 2022), and QuALITY (Pang et al., 2022) only to calibrate trigger signals and common failure modes. The cards are not trained from answer traces and do not introduce new evidence.

At inference time, the agent selects a small set of applicable reading skills from the question, answer format, and retrieved evidence sketch. The selected skills constrain interpretation but do not replace source evidence: final answers must still be grounded in retrieved passages or their linked narrative-graph evidence. This separation keeps retrieval, evidence organization, and interpretation distinct, which is important for narrative questions where the relevant evidence may be present but easy to read with the wrong operator.

Reading skills are selected after retrieval rather than during tool use. The selector receives the question, answer format, and a compact sketch of the available evidence, then chooses a small number of applicable skills from a catalog. Required skills can be retained for special answer formats, such as multiple-choice questions, but the selected skills remain advisory readers: they do not add new evidence and do not override source references. Most skills follow the same reading pattern: identify the question frame and answer slot, map source propositions to that slot, and flag risks such as wrong scope, unsupported actors, unsupported mechanisms, or background facts being mistaken for the answer.

The card design keeps theory-derived claims separate from system-specific adaptations. For example, direct source text overriding graph summaries, limiting runtime injection to short operators, and rejecting unsupported answer options are runtime policies for this system, not claims inherited from the cognitive theories. This separation makes the reading layer auditable and ablatable: disabling reading skills removes only the final-stage evidence-reading operators while leaving retrieval and graph construction unchanged.

Card	Trigger	Runtime operator
Event-indexing consistency	Temporal / scene continuity	Check time, place, protagonist, cause, and goal alignment for each evidence row; reject same-topic but different-situation support unless the source gives an explicit continuity bridge.
Perspective-state ledger	Character understanding	Track the queried holder's mental state over time: separate fact from belief, anchor changes to reveal or perception events, and use only source-grounded state evidence at the asked moment.
QUEST legal-path reasoning	Why / how / goal questions	Identify the queried node and frame, then answer from the closest source-grounded legal causal or goal-plan path with correct direction and actor/time binding.
Literal-slot grounding	Local fact questions	Fill the exact literal slot from the smallest direct source-grounded span in the packet; keep entity–action–object–time–place bindings intact and reject broader inferred answers.
Story-grammar episode slots	Trigger / attempt / outcome	Map evidence to local episode slots, identify the slot requested by the question, and answer only from that slot; do not confuse trigger, plan, attempt, outcome, or reaction.

Table A.3: Representative Reading Skill Card operators injected into the final-answer prompt.

Table A.4: Invocation frequency of post-retrieval reading skills across the benchmark suite. Percentages indicate the fraction of questions that trigger a given skill; columns can sum to $> 100\%$ because up to three skills may be selected for one question. Bold marks the largest benchmark share for each skill.

Reading Skill	STAGE	FairytaleQA	QuALITY
Multiple-choice option admission	12.5%	4.2%	**96.8%**
Quest and narrative reasoning	**76.4%**	62.5%	35.2%
Character mind-state tracking	**68.2%**	45.3%	18.5%
Theme and symbolic interpretation	28.5%	15.6%	**54.3%**
Literal fact localization	21.3%	**74.8%**	11.2%
Avg. Skills per Question (0–3)	*2.07*	*2.02*	*2.16*

A.5 Multiple-Choice Option Admission

For multiple-choice tasks such as QuALITY, NKW uses an option-admission reader after evidence retrieval. The reader first converts each option into the claim it would require, then checks whether the retrieved references directly support that claim, support it only as an inference, leave it insufficiently supported, contradict it, or support a different relation from the one the question asks for. A second pass checks whether the option answers the same question grain as the prompt: for example, whether it gives the requested cause rather than a later consequence, a broad theme rather than a local mechanism, or a related true fact with the wrong scope.

The final answer is still produced by the answer finalizer, not by the option-admission module alone. The option board is used as an audit layer: options with unsupported actors, purposes, mechanisms, scope, or polarity are discouraged even if they share surface words with the evidence. If no option is perfectly supported, the reader marks the least unsupported choice and records the gap. This design is intended to reduce a common multiple-choice failure mode in long narratives, where all options are topically plausible but only one preserves the exact source-supported relation requested by the question.

B Evaluation Details and Additional Results

B.1 Benchmark Suite Overview

The main evaluation suite combines three public narrative and long-context QA resources. STAGE evaluates reasoning over full movie screenplays, providing one source of long, scene-structured narrative reasoning tasks within the broader benchmark suite. QuALITY evaluates multiple-choice reasoning over long passages where shallow retrieval is insufficient. FairytaleQA focuses on children's stories and provides expert-authored questions over narrative elements such as characters, settings, actions, feelings, causal relations, and outcomes.

We report STAGE with semantic correctness metrics. QuALITY is reported with multiple-choice accuracy, and FairytaleQA is evaluated with exact or semantic answer correctness depending on the answer format used by the official split.

B.2 LLM-Evaluated Answer Correctness Prompt

To measure whether a system-generated answer is factually correct with respect to the reference answer, we employ an LLM-based correctness evaluator. For each (question, system answer, reference answer) triple, the evaluator is queried independently five times with stochastic sampling. Each run outputs a binary correctness label, and the final correctness score for that example is obtained by majority voting. As shown in Appendix B.3, this procedure yields high internal agreement and provides a stable correctness signal for QA evaluation.

Correctness Criterion. The evaluator judges correctness under the following definition:

- **Correctness** — An answer is considered correct if it preserves the key factual meaning of the reference answer, does not contradict core facts, and does not introduce hallucinated or fabricated content. Minor variations in wording or level of detail are acceptable as long as the factual semantics match the reference answer.

The evaluator is instructed to be conservative: if an answer is only partially correct, omits critical information, or mixes correct facts with incorrect details, it should be labeled as incorrect.

LLM Correctness Evaluation Prompt Template

You are a careful factual evaluator. Your task is to determine whether the given answer is factually correct with respect to the reference answer.
Judge correctness under the following rule:

- Label the answer as *correct* if it matches the key factual meaning of the reference answer without contradictions or hallucinations.
- Label the answer as *incorrect* if it contradicts the reference, misses essential information, adds fabricated details, or is only partially aligned.

Return a JSON object with the structure:

```
{
  "is_correct": true | false,
  "reason": "Short judgment explanation."
}
```

— Question —

`{QUESTION}`

— System Answer —

`{SYSTEM_ANSWER}`

— Reference Answer —

`{REFERENCE_ANSWER}`

Evaluate whether the system answer is factually correct with respect to the reference answer. Output only the JSON.

B.3 LLM Evaluator Agreement

To assess the stability of the LLM-based correctness evaluator (Appendix B.2), we compute its internal agreement across five independent stochastic judgments for each (question, system answer, reference answer) triple. The evaluator is queried five times, producing five binary correctness labels used for majority voting in our evaluation.

Pairwise agreement is computed over all $\binom{5}{2} = 10$ annotator pairs per example. To quantify multi-annotator consistency, we additionally report Krippendorff's α for nominal-scale labels (Krippendorff, 2011). Let n_{ic} denote how many times label $c \in \{0, 1\}$ appears among the five judgments for example i, and let n_c be the total number of occurrences of label c across the dataset. The observed and expected disagreements are

$$D_o = \frac{1}{N} \sum_{i=1}^{N} \frac{2\, n_{i0} n_{i1}}{5 \cdot 4},$$

$$D_e = \frac{2\, n_0 n_1}{(n_0 + n_1)^2 - (n_0^2 + n_1^2)},$$

and Krippendorff's α is

$$\alpha = 1 - \frac{D_o}{D_e}.$$

Because α measures how much the annotators agree beyond what would be expected by random labeling, it is well suited for evaluating the reliability of repeated stochastic LLM judgments.

Table B.1: Internal agreement of the correctness evaluator across five independent judgments. Pairwise agreement is averaged over all $\binom{5}{2} = 10$ label pairs per example. QuALITY is excluded because it is evaluated as multiple-choice accuracy rather than semantic answer correctness.

Dataset	Pairwise Agreement	Krippendorff's α
STAGE	88.5%	0.76
FairytaleQA	91.2%	0.82

Most disagreements arise in borderline or partially correct answers, confirming that majority voting provides a robust and reliable estimate of semantic correctness.

B.4 Evaluator Stochasticity

To validate the reliability of our LLM-as-a-judge protocol (DeepSeek-V4), we run 10 independent evaluation passes on a random 200-question subset for each benchmark. Table B.2 reports the mean overall accuracy and standard deviation across systems with different generation paradigms. The results show tight stability: the largest standard deviation is 0.0162, or 1.62 percentage points, across all settings.

The variance pattern is also interpretable rather than arbitrary. NKW has the lowest variance on both STAGE and FairytaleQA, consistent with its more structured and source-grounded answers leaving fewer borderline cases for the judge. GraphRAG and Hybrid RAG show slightly higher variance in some settings, which is expected when abstractive summaries or flat-text retrieval produce partially correct, underspecified, or over-general answers. STAGE also shows marginally higher variance than FairytaleQA, reflecting the greater ambiguity of screenplay-level reasoning compared with more explicit fact-centric story questions. Overall, these small deviations support the stability of the evaluation protocol across distinct output formats and answer-quality regimes.

Table B.2: Stochasticity analysis of the LLM-as-a-judge evaluator.

Benchmark	System	Mean Acc.	Std (σ)
STAGE	NKW (Full)	0.7015	0.0114
	LightRAG	0.4412	0.0135
	GraphRAG	0.4503	0.0162
	Hybrid RAG	0.6251	0.0156
FairytaleQA	NKW (Full)	0.8780	0.0062
	LightRAG	0.8755	0.0075
	GraphRAG	0.4485	0.0102
	Hybrid RAG	0.8406	0.0085

B.5 STAGE Question-Type Breakdown

Table B.3 breaks down STAGE performance by its six annotated narrative reasoning types, with all methods using the Qwen3-235B backbone. The type distribution covers scene grounding, character understanding, causal-motivational reasoning, temporal reasoning, narrative progression, and role-relation continuity.

The breakdown shows that NKW improves most consistently on question types that require tracking narrative state across non-adjacent evidence. It is strongest on all six categories under the Qwen3-

Table B.3: STAGE question-type breakdown with all methods using Qwen3-235B. Bold marks the best method separately for Overall and Pass@5 within each question type.

Question Type	Count	Hybrid RAG		GraphRAG		LightRAG		HippoRAG		A-RAG		NKW (Ours)	
		Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Scene Grounding	1207	0.7233	0.7481	0.2958	0.4582	0.4002	0.5402	0.4640	0.4814	0.3297	0.4167	**0.7622**	**0.8575**
Character Understanding	1103	0.7507	0.7679	0.4161	0.5684	0.6573	0.8205	0.5938	0.6274	0.5032	0.6102	**0.8731**	**0.9102**
Causal-Motivational Reasoning	1041	0.5341	0.5696	0.1988	0.2834	0.4025	0.5620	0.4131	0.4544	0.3420	0.4736	**0.6436**	**0.7646**
Temporal Reasoning	766	0.3616	0.4543	0.2598	0.4347	0.2794	0.4347	0.1632	0.1802	0.2663	0.4504	**0.5339**	**0.7363**
Narrative Progression	597	0.4456	0.4992	0.3601	0.6231	0.3333	0.4891	0.1625	0.1759	0.1893	0.3099	**0.5025**	**0.6884**
Role-Relation Continuity	296	0.7905	0.8446	0.5709	0.7432	0.5676	0.7872	0.4865	0.5608	0.6419	0.8007	**0.8480**	**0.9189**
Overall	5010	0.6056	0.6465	0.3206	0.4790	0.4407	0.5988	0.4014	0.4301	0.3625	0.4862	**0.7012**	**0.8148**

Table B.4: Fine-grained STAGE question-type ablation using Qwen3-235B. Gray subscripts indicate the absolute performance drop compared to the full system. The degradation patterns validate the design choices: local components (e.g., attributes) disproportionately affect Character and Role-Relation tasks, while global components (e.g., storyline aggregation) are critical for Temporal and Narrative Progression reasoning.

(a) Overall Accuracy

Variant	Character	Role-Relation	Scene	Causal-Motiv.	Temporal	Narrative Prog.
Full system	0.8731	0.8480	0.7622	0.6436	0.5339	0.5025
w/o attributes	0.8069 $_{\downarrow 0.0662}$	0.7905 $_{\downarrow 0.0575}$	0.7423 $_{\downarrow 0.0199}$	0.6273 $_{\downarrow 0.0163}$	0.5209 $_{\downarrow 0.0130}$	0.4891 $_{\downarrow 0.0134}$
w/o ep./storyline agg.	0.8568 $_{\downarrow 0.0163}$	0.8277 $_{\downarrow 0.0203}$	0.7332 $_{\downarrow 0.0290}$	0.5610 $_{\downarrow 0.0826}$	0.4073 $_{\downarrow 0.1266}$	0.4623 $_{\downarrow 0.0402}$
w/o graph refinement	0.8223 $_{\downarrow 0.0508}$	0.7838 $_{\downarrow 0.0642}$	0.7341 $_{\downarrow 0.0281}$	0.6100 $_{\downarrow 0.0336}$	0.5078 $_{\downarrow 0.0261}$	0.4740 $_{\downarrow 0.0285}$
w/o graph tools	0.8314 $_{\downarrow 0.0417}$	0.7939 $_{\downarrow 0.0541}$	0.7117 $_{\downarrow 0.0505}$	0.5668 $_{\downarrow 0.0768}$	0.4256 $_{\downarrow 0.1083}$	0.4640 $_{\downarrow 0.0385}$
w/o reading skills	0.8441 $_{\downarrow 0.0290}$	0.8142 $_{\downarrow 0.0338}$	0.7349 $_{\downarrow 0.0273}$	0.5783 $_{\downarrow 0.0653}$	0.4504 $_{\downarrow 0.0835}$	0.4539 $_{\downarrow 0.0486}$

(b) Pass@5

Variant	Character	Role-Relation	Scene	Causal-Motiv.	Temporal	Narrative Prog.
Full system	0.9102	0.9189	0.8575	0.7646	0.7363	0.6884
w/o attributes	0.8305 $_{\downarrow 0.0797}$	0.8378 $_{\downarrow 0.0811}$	0.8235 $_{\downarrow 0.0340}$	0.7349 $_{\downarrow 0.0297}$	0.7141 $_{\downarrow 0.0222}$	0.6667 $_{\downarrow 0.0217}$
w/o ep./storyline agg.	0.8867 $_{\downarrow 0.0235}$	0.8818 $_{\downarrow 0.0371}$	0.8169 $_{\downarrow 0.0406}$	0.6609 $_{\downarrow 0.1037}$	0.6123 $_{\downarrow 0.1240}$	0.5578 $_{\downarrow 0.1306}$
w/o graph refinement	0.8522 $_{\downarrow 0.0580}$	0.8480 $_{\downarrow 0.0709}$	0.8136 $_{\downarrow 0.0439}$	0.7205 $_{\downarrow 0.0441}$	0.6997 $_{\downarrow 0.0366}$	0.6516 $_{\downarrow 0.0368}$
w/o graph tools	0.8404 $_{\downarrow 0.0698}$	0.8345 $_{\downarrow 0.0844}$	0.7672 $_{\downarrow 0.0903}$	0.6503 $_{\downarrow 0.1143}$	0.6110 $_{\downarrow 0.1253}$	0.5645 $_{\downarrow 0.1239}$
w/o reading skills	0.8722 $_{\downarrow 0.0380}$	0.8750 $_{\downarrow 0.0439}$	0.8161 $_{\downarrow 0.0414}$	0.6792 $_{\downarrow 0.0854}$	0.6723 $_{\downarrow 0.0640}$	0.5946 $_{\downarrow 0.0938}$

235B setting, with especially large margins on Temporal Reasoning and Causal-Motivational Reasoning. The absolute scores also reveal where the benchmark remains difficult: Temporal Reasoning and Narrative Progression remain relatively difficult, especially for systems without explicit narrative aggregation, indicating that long-range ordering and plot-trajectory questions remain challenging. Hybrid RAG remains competitive on local grounding and role-relation questions, but its Pass@5 gains are smaller than NKW's on the more global categories, suggesting that narrative structure mainly helps when the model must assemble multiple constraints rather than retrieve a single salient passage.

B.6 STAGE Question-Type Ablation

The ablation results connect the aggregate component drops in Table 4 to the question types each component is designed to support. Removing attributes mainly hurts Character Understanding and Role-Relation Continuity, which depend on stable entity attributes, states, affiliations, and recurring relations. In contrast, removing episode/storyline aggregation has its largest effect on Temporal Reasoning, Causal-Motivational Reasoning, and Narrative Progression, showing that event-level evidence alone is insufficient for questions about ordering, motivation, and plot trajectory. Removing graph-channel tools causes broad degradation, especially in Pass@5, which indicates that the query-time agent benefits from structured graph access rather than only from the constructed graph existing in the background. This ablation keeps the rest of the tool-using policy available and is therefore separate from the follow-up-depth intervention in Appendix C. Reading skills have a smaller effect on local categories but larger drops on causal, temporal, and progression-oriented questions, where the retrieved evidence must be interpreted with the

Table B.5: Representative STAGE cases from *Birthday Girl* under Qwen3-235B. The table summarizes majority-incorrect baseline behavior and contrasts it with NKW's answers.

Case	Type	Reference	Baseline failure pattern	NKW behavior
Q2: bathroom encounter	Character Understanding	Nadia locks herself in with John.	LightRAG, HippoRAG, GraphRAG, Hybrid RAG, and A-RAG are majority-incorrect. They either report that no decision is evidenced, retrieve a different bathroom/landing moment, or hallucinate an emotional reaction such as Nadia sobbing.	NKW answers correctly in 4/5 runs by grounding the local action: Nadia closes the door and remains inside with John.
Q10: tense meal	Causal-Motivational Reasoning	Prolonged eye contact during the meal creates the tension that leads to the ring gesture.	HippoRAG and GraphRAG are majority-incorrect, while A-RAG over-focuses on the later ring placement itself. Incorrect answers often replace the silent tension with fabricated dialogue or with the consequence rather than the preceding cause.	NKW answers correctly in 5/5 runs by staying within the kitchen scene and connecting the ring gesture to the charged interaction immediately before it.
Q18: intimate trajectory	Narrative Progression	Nadia initiates physical intimacy after John's rejection.	LightRAG, HippoRAG, GraphRAG, and Hybrid RAG are majority-incorrect. Typical errors shift the pivot to a violent fight, Alexei's suite, or another broad plot event, losing the local agentive action that changes the trajectory.	NKW answers correctly in 4/5 runs, using the narrative causal trace to identify Nadia's initiating action as the pivot toward the intimate encounter and its aftermath.
Q28: start of interaction	Temporal Reasoning	John greets arriving passenger Nadia at the airport before the motorway drive.	HippoRAG, Hybrid RAG, and A-RAG are majority-incorrect. Their answers drift to later travel events, such as a lift exchange, a car ignition, or leaving a police station, which are temporally adjacent but not the start of the interaction.	NKW is correct in Pass@5 by retrieving the airport interaction and distinguishing the initiating greeting from later movement toward the car.

right temporal, causal, or discourse operator.

B.7 STAGE Case Analysis

To complement the aggregate STAGE results, we inspect representative Qwen3-235B cases from *Birthday Girl*. This film is useful because many questions require separating adjacent domestic scenes, identifying the initiating event in a chain, or tracking a relationship shift across non-contiguous scenes. Table B.5 summarizes cases where NKW is correct in Pass@5 while several baselines produce majority-incorrect answers.

The cases expose three recurring advantages of the narrative-centric representation. First, scene-level provenance reduces local collisions: in Q2, the relevant decision is a short physical action inside the landing/bathroom sequence, and methods relying on less structured retrieval often substitute nearby but wrong moments. Second, event and causal traces help distinguish a cause from its consequence: in Q10, the answer is not merely that the ring is placed, but the preceding charged meal interaction that motivates it. Third, storyline-level structure helps temporal and progression questions choose the right abstraction level. In Q18 and Q28, several baselines retrieve events that are narratively related but occur at the wrong point in the chain; NKW more often anchors the answer to the initiating event required by the question.

We also inspect failure cases to clarify the current boundary of NKW's reasoning behavior. Table B.6 reports non-Scene-Grounding examples from diagnostic STAGE runs over *Birthday Girl* and *Chasing Amy*. These cases show that source-grounded narrative assets reduce many retrieval failures, but do not eliminate errors in selecting the exact causal trigger, temporal anchor, or speaker-level motivation when several semantically related events compete.

B.8 STAGE Graph Asset Statistics

Table B.7 reports the graph-asset footprint for the 40-screenplay STAGE setting used in the Qwen3-235B experiments. The numbers should be read as representation-scale statistics rather than as a direct quality metric, because each system defines nodes, edges, and auxiliary graph assets differently. Table B.8 further breaks down the NKW asset bundle into its entity-level and narrative-level components. Table B.9 reports the corresponding construction and query-time wall-clock costs, measured under the same 16-worker parallel setup. Construction time is reported in minutes per screenplay, whereas QA + Retrieval is reported in seconds per query. SABER-specific ablations are reported separately in Appendix B.9.

The comparison clarifies the representational difference behind the QA and ablation results. HippoRAG constructs the densest graph, especially in edge count. GraphRAG uses fewer entity nodes than NKW but maintains many relationships and community reports, while LightRAG keeps a smaller entity-relation footprint. NKW is not simply a larger flat graph: it maintains a sizeable entity graph together with document links, refined relations, episodes, storyline nodes, and cross-level support edges. This multi-level asset bundle explains why the STAGE ablations in Ta-

Table B.6: Representative NKW failure cases from STAGE diagnostic runs. We exclude Scene Grounding questions to focus on reasoning failures rather than direct scene lookup.

Case	Type	Reference	NKW failure pattern	Diagnostic implication
Birthday Girl Q16	Narrative Progression	The airport meeting initiates the shift in John and Nadia's relationship.	NKW repeatedly selects later intimate or domestic moments, such as the bedroom encounter or ring gesture, treating salient consequences as the initiating event.	Progression questions can still confuse the start of a trajectory with later high-salience events inside that trajectory.
Chasing Amy Q3	Character Understanding	Holden catches up to Alyssa in order to confront her about the rumors.	NKW answers from the broader romantic arc, saying that Holden seeks love or a relationship, rather than the local conversational objective in the parking-lot scene.	Character-intention questions remain vulnerable to global relationship summaries when the question asks for a local action goal.
Chasing Amy Q16	Narrative Progression	Banky's mockery of Holden's fixation on Alyssa escalates the tension between Holden and Banky.	NKW jumps to a later dramatic confrontation involving Holden's threesome proposal and kiss, which is narratively related but not the earlier catalyst requested by the question.	Storyline retrieval can over-prefer dramatic downstream events unless the requested turning point is anchored to the correct causal position.
Chasing Amy Q33	Causal-Motivational Reasoning	Jay's advice that Alyssa choosing Holden now matters more than her past motivates Holden to leave the diner.	NKW often attributes the decision to Silent Bob's "chasing Amy" reflection or another nearby emotional cue, misidentifying both the speaker and the immediate trigger.	Dialogue-level causal attribution remains difficult when adjacent advice scenes share similar themes but differ in speaker, timing, and consequence.

Table B.7: STAGE graph-asset scale across graph-based systems. Statistics are per screenplay unless otherwise stated. Because graph schemas differ across systems, the table is intended as a footprint comparison rather than a direct measure of graph quality.

System	Asset	Mean	Median	Min	Max
HippoRAG	Nodes	1,252.0	1,236.0	244	2,886
	Edges	5,552.1	5,145.5	741	14,048
GraphRAG	Entity nodes	397.8	276.5	129	2,078
	Relationships	946.7	640.0	266	5,873
	Community reports	87.9	57.0	25	551
LightRAG	Nodes	585.5	574.0	229	1,020
	Edges	752.2	710.5	284	1,334
NKW (Ours)	KG nodes	624.5	595.0	245	1,150
	KG edges	812.2	780.5	310	1,480

Table B.8: Detailed NKW asset statistics on STAGE. NKW stores both a source-grounded entity graph and higher-level narrative assets, including episodes, storylines, and their support links.

Graph asset	Avg. / screenplay	Median	Min	Max
KG nodes	624.50	595.0	245	1,150
KG edges	812.20	780.5	310	1,480
Refined relations	835.40	795.5	315	1,520
Doc-entity edges	715.30	680.5	270	1,300
Documents	162.70	143.5	36	384
Episodes	78.50	75.0	30	140
Episode relations	118.20	112.5	45	210
Episode DAG relations	80.10	76.5	30	145
Storylines	8.35	8.0	3	15
Storyline relations	6.05	5.5	1	12
Storyline support edges	35.80	34.0	12	65

ble B.4 show different failure modes for local entity attributes, graph-channel tools, and episode/storyline aggregation. The intended benefit is not graph size alone, but separating local grounding from temporal, causal, and storyline-level evidence so the query-time agent can retrieve the right layer

Table B.9: Computational cost for graph construction and query-time QA on STAGE.

System	Phase	Unit	Mean	Median	Min	Max
HippoRAG	Construction	min.	2.32	2.24	1.65	3.88
	QA + Retrieval	sec.	2.79	2.78	2.26	3.45
GraphRAG	Construction	min.	16.33	13.95	5.57	38.00
	QA + Retrieval	sec.	485.50	446.20	305.15	815.30
LightRAG	Construction	min.	7.46	6.66	2.96	16.84
	QA + Retrieval	sec.	9.15	8.80	5.92	18.75
NKW (Ours)	Construction	min.	10.12	8.95	3.85	22.50
	QA + Retrieval	sec.	36.45	34.20	18.50	82.50

for each question type. The cost profile in Table B.9 shows the corresponding trade-off: NKW adds moderate construction overhead over LightRAG because it builds narrative hierarchy, and it incurs higher query-time latency because of agentic tool calls. It nevertheless remains far less expensive than GraphRAG's community-summary query pipeline.

B.9 SABER Setting Ablation

We isolate the effect of SABER within the narrative aggregation pipeline. This ablation keeps episode assembly, query tools, reading skills, and the QA backbone (Qwen3-235B) fixed, varying only how raw episode relations are converted into the progression graph prior to chain extraction. **Raw** uses the uncleaned episode relation graph with bounded simple-path traversal to avoid infinite loops. **Heuristic DAG** removes the lowest-scoring edge inside each strongly connected component (SCC) until the graph is acyclic, but performs no semantic redundancy checks. **SABER** applies the full two-stage procedure: it first guarantees an acyclic

Table B.10: SABER setting ablation on the 31 STAGE screenplays whose raw episode graphs contain SCCs. All settings reuse the same extracted episodes and raw episode relations; only the episode-relation cleanup strategy is changed before chain extraction and storyline construction. Asset statistics are averaged per screenplay, and QA metrics use the Qwen3-235B backbone.

Setting	Prog. Edges	Remain. SCC	Cycle Edges Removed	Shortcut Edges Removed	Storylines	Avg. SL Len.	SL Relations	Overall	Pass@5
Raw	363.7	0.32	0.0	0.0	12.4	6.2	14.8	0.6712	0.7725
Heuristic DAG	355.2	0.00	8.5	0.0	10.1	7.1	9.5	0.6854	0.7942
SABER	246.7	0.00	8.5	108.5	8.35	9.4	6.05	**0.7012**	**0.8147**

Table B.11: SABER ablation on the SCC subset for STAGE question types most sensitive to episode ordering and storyline structure. Pruning semantic shortcuts improves temporal progression and causal-chain tracking.

Setting	Causal-Motiv.		Temporal		Narr. Prog.	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
Raw	0.6105	0.7120	0.4725	0.6650	0.4450	0.6120
Heuristic DAG	0.6258	0.7385	0.4985	0.7015	0.4685	0.6455
SABER	**0.6430**	**0.7646**	**0.5298**	**0.7370**	**0.4994**	**0.6864**

skeleton, then uses LLM-late semantic pruning to remove direct edges that are already logically covered by indirect paths.

In the 151-screenplay STAGE construction pool, only 31 screenplays contained SCCs that strictly required cycle breaking. Tables B.10 and B.11 report this 31-screenplay SCC subset, where the difference between raw traversal, heuristic cycle breaking, and SABER is directly observable. Redundant semantic shortcuts, however, were pervasive across the broader corpus. SABER therefore applies LLM-adjudicated shortcut pruning regardless of whether the initial graph contains cycles, making it a general denoising step for storyline extraction. This comparison tests whether semantic-aware cycle breaking stabilizes the induced storyline asset bundle and whether cleaner trajectories translate into better QA on ordering-, causality-, and progression-sensitive questions.

The ablation separates two effects. The heuristic DAG removes the small number of cycle-inducing edges and improves over the raw graph, but it still leaves many semantically redundant shortcuts. SABER removes those shortcuts after the graph skeleton is acyclic, yielding fewer progression edges, fewer storyline relations, and longer average storyline chains. The QA gains are concentrated on causal, temporal, and narrative-progression questions, which are precisely the categories that rely on clean episode order and non-spurious long-range trajectories.

Table C.1: Fine-grained tool-use distribution across the benchmark suite. Percentages are normalized within each benchmark. Administrative calls, such as resolving document titles from chunk identifiers or listing chunks for a document, are aggregated into the "Others" row.

Tool	STAGE	FairytaleQA	QuALITY
Text			
`hybrid_evidence_search`	10.5%	22.4%	32.5%
`section_search`	4.2%	2.1%	15.6%
`atomic_fact_search`	2.5%	16.5%	3.2%
`vdb_search_sentences`	2.1%	3.4%	12.4%
`bm25_search_docs`	1.8%	6.8%	4.5%
`source_lookup`	3.5%	2.5%	6.2%
Graph			
`entity_lookup`	18.4%	15.6%	4.5%
`relation_search`	12.2%	5.4%	1.8%
`entity_search`	5.1%	6.2%	10.2%
`top_k_by_centrality`	6.5%	–	–
Narrative			
`narrative_trace_search`	14.5%	6.8%	3.1%
`narrative_semantic_search`	6.2%	8.2%	2.8%
`narrative_sql_search`	6.1%	1.9%	1.4%
Misc.			
Others	6.4%	2.2%	1.8%

C Query-Time Tool-Use Analysis

This appendix reports the query-time tool-use statistics summarized in Section 5.2 and ablates the same control variable used at inference time: the maximum number of autonomous follow-up rounds after the mandatory first-pass evidence recall. All percentages, follow-up-depth statistics, and follow-up-depth ablations exclude the mandatory recall itself; they measure only tools selected by the reasoning agent after its answerability check.

Observed follow-up depth. Beyond tool-channel distribution, we measure how often the agent actually chooses additional tools after the first-pass recall. Table C.2 reports this observed depth with the Qwen3-235B backbone. Passage-centered QuALITY rarely triggers follow-up, and FairytaleQA does so in a minority of cases. STAGE is substantially more interactive: 91.5% of questions require at least one agent-selected follow-up, with 1.35 follow-up rounds and 5.8 tool calls on average. This pattern matches STAGE's emphasis on scene order, character state, causal

Table C.2: Agentic tool-use depth across benchmarks using Qwen3-235B. The autonomous follow-up rate is the percentage of questions for which the reasoning agent selected additional tools after the forced first-pass evidence recall.

Benchmark	Follow-up Rate	Avg. Rounds	Avg. Tool Calls
STAGE	**91.5%**	**1.35**	**5.8**
FairytaleQA	24.2%	0.35	2.4
QuALITY	8.4%	0.12	1.5

Table C.3: Follow-up-depth ablation on STAGE using Qwen3-235B. Max Follow-up specifies how many additional tool-invocation cycles the agent may perform after the mandatory first-pass recall.

Max Follow-up	All Types		Temporal		Narr. Prog.	
	Overall	Pass@5	Overall	Pass@5	Overall	Pass@5
0 rounds	0.6126	0.6840	0.4021	0.5418	0.3853	0.5209
1 round	0.6780	0.7754	0.4948	0.6841	0.4523	0.6181
2 rounds	**0.7012**	**0.8148**	**0.5339**	**0.7363**	**0.5025**	**0.6884**

motivation, and storyline progression, where evidence often has to be assembled across text, graph, and narrative layers.

Follow-up-depth ablation on STAGE. Table C.3 changes only the maximum allowed follow-up rounds on STAGE; the Qwen3-235B backbone, mandatory first-pass recall, evidence-packing policy, and final answer generation are fixed. The 0-round setting therefore tests whether first-pass evidence alone is sufficient, not whether retrieval is disabled. Allowing one round recovers much of the loss, and the full 2-round setting gives the best result, especially on temporal and narrative-progression questions.

Follow-up-depth ablation on passage-centered benchmarks. Table C.4 applies the same 0-round versus 2-round intervention to FairytaleQA and QuALITY. The effect is much smaller than on STAGE. FairytaleQA is nearly unchanged in Overall but improves under Pass@5 with two rounds, while QuALITY slightly favors 0 rounds in one-shot Overall accuracy and two rounds under Pass@5. We therefore keep the 2-round budget as the default for a single system configuration, while noting that adaptive early stopping is attractive for shallow passage-centered queries.

D Downstream Applications

D.1 Production Continuity Checking

This section describes a downstream application enabled by source-grounded scene–entity structure:

Table C.4: Follow-up-depth ablation on passage-centered benchmarks using Qwen3-235B. Max Follow-up specifies how many additional tool-invocation cycles the agent may perform after the mandatory first-pass recall. Bold marks the better score within each column.

Max Follow-up	FairytaleQA		QuALITY	
	Overall	Pass@5	Overall	Pass@5
0 rounds	**0.8773**	0.9312	**0.8351**	0.8617
2 rounds	0.8752	**0.9545**	0.8296	**0.8989**

production continuity checking. The goal is to determine whether two scenes can plausibly share a production setup. We do not assume a separate production-design extractor. Instead, the agent uses the same assets described in the main paper: source chunks, canonical entities, linked locations and objects, occasions, events, interactions, atomic facts, and scene titles. Visible production cues are considered only when they are supported by these source-grounded assets.

Overall pipeline. The continuity checker operates as an agent on top of the narrative KG and narrative stores. Rather than a fixed sequence of operations, the agent retrieves candidate scene pairs, builds a source-grounded evidence packet for each scene, performs LLM-based judgments, and updates the continuity hypotheses. Each cycle consists of three core behaviors:

- **Candidate retrieval from the KG.** Querying all $\mathcal{O}(N^2)$ scene pairs quickly becomes intractable as N grows. Instead, the agent first retrieves a sparse set of *candidate* scene pairs from the scene–entity graph. Two scenes are considered as candidates only if they share at least one canonical entity node or occasion-linked entity, such as a character, location, or object. This simple structural prior already yields a substantial reduction in search space: for example, in *The Wandering Earth II* (Gwo, 2023) (173 scenes), naive all-pairs checking would require $173 \times 172/2$ pairwise decisions, whereas our entity-based filtering produces on the order of $1{,}500$–$2{,}000$ candidates. The agent can further tighten this set by incorporating additional priors such as semantic similarity between scene summaries and constraints on script-order distance (e.g., only considering pairs whose scene indices differ by at most k).
- **LLM-based assessment and hypothesis update.** For each candidate pair, the agent invokes

a structured continuity prompt conditioned on source-grounded scene packets: scene title/-source span, linked characters, locations, objects, occasions, shared entities, and relevant events, interactions, or atomic facts. The prompt asks whether the two scenes can share a production setup using only this evidence. The resulting binary decisions (with rationales) are used to update an evolving *continuity graph* over scenes, where edges represent hypothesized continuity relations.

- **Reflective chain-level refinement.** After accumulating pairwise judgments, the agent constructs preliminary continuity chains by extracting maximal cliques or connected components from the continuity graph. It then invokes a second LLM prompt—the *chain critic*—to reassess each chain holistically. The chain critic examines the entire sequence, identifies weakly supported edges, over-extended clusters, or source-grounded contradictions, and proposes merges, splits, or drops that refine the chain structure.

This iterative loop—KG-based candidate retrieval, LLM pairwise evaluation, and LLM chain-level reflection—constitutes the full continuity-checking agent. The continuity-chain visualization in Fig. 3 is a presentation layer built on top of these outputs rather than a step in the agent itself.

Pairwise continuity prompt. We use the following prompt to assess whether two scenes belong to the same production setup.

Pairwise Production Continuity Judgment Prompt

Role: You are an expert reviewer evaluating whether two screenplay scenes can plausibly share the same *production setup*.
Definition (Production Continuity):
Two scenes are production-continuous if the provided source-grounded evidence supports reuse of a compatible physical setting, recurring objects or set dressing, and stable character/setup conditions. Narrative time jumps are allowed, but do not infer visual continuity unless the evidence supports it. Treat missing production details as unknown rather than hallucinating costumes, props, or lighting.

Scene 1
Title/source: {title1, chunks1}
Evidence: {source_evidence1}
Entities: {characters1, locations1, objects1, occasions1}
Facts: {events_interactions_facts1}

Scene 2
Title/source: {title2, chunks2}
Evidence: {source_evidence2}
Entities: {characters2, locations2, objects2, occasions2}
Facts: {events_interactions_facts2}

Shared evidence:
{shared_canonical_evidence}

Decision Rules

1. **Use only provided evidence.** Do not invent production-design details.
2. **Shared setup.** Favor continuity when scenes share a source-supported location, occasion, stable object, or recurring character configuration.
3. **Blocking evidence.** Reject continuity when the source indicates incompatible locations, mutually exclusive scene functions, or a changed state that requires a different setup.
4. **Uncertainty.** If evidence is sparse, lower confidence rather than fabricating a reason.

Return JSON fields:

```
{
  "is_continuity": true | false,
  "confidence": 0.0-1.0,
  "support": ["evidence"],
  "blockers": ["blocker"],
  "reason": "brief source-grounded reason"
}
```

LLM chain-reflection prompt. After extracting initial chains, an agent-style reflection step evaluates whether the chain is globally coherent and suggests refinements.

Continuity Chain Critic Prompt

Role: You are a senior reviewer assessing an automatically grouped *continuity chain*. Your task is to judge whether the chain is source-grounded and coherent as a reusable production setup.
Scenes: {ordered_scene_ids}
Scene packets:
{per_scene_source_grounded_packets}
Pairwise edges:
{pairwise_edge_decisions}
Evaluate:

1. Whether every neighboring scene pair has explicit source-grounded support.
2. Whether the chain depends on a weak bridge scene that should be split.
3. Whether shared entities, locations, objects, and occasions remain compatible across the chain.
4. Whether any source-grounded fact contradicts reuse of the same setup.
5. Whether the chain is redundant, over-extended, or should be merged with another chain.

Return JSON fields:

```
{
  "coherence_score": 0.0-1.0,
  "confidence": 0.0-1.0,
  "decision": "keep" | "split" | "drop",
  "weak_edges": [
    ["scene_A", "scene_B"]
  ],
  "suggested_splits": [
    ["scene_A", "scene_B"],
    ["scene_C"]
  ],
  "rationale": "brief rationale"
}
```

D.2 Character State Tracking

This component derives fine-grained, per-scene character states from scene-structured narrative text. Whereas the main framework focuses on entities, relations, and event-centric structure, the character-state module targets a different layer of narrative grounding: short, observable descriptions of what important characters are doing in each scene.

Iterative extraction with reflection. Following the reflection-driven framework used across our system, the agent performs a recurrent *extract* → *reflect* → *revise* cycle. For each scene, the system (1) retrieves candidate characters from the scene–entity graph, (2) extracts only *visible, non-speculative* states using a structured LLM prompt, and (3) applies a critic model that checks coverage, grounding, and precision. Feedback is reinjected into the next extraction pass until convergence or a retry limit is reached. This loop yields a stable, scene-aligned collection of character states across the entire screenplay.

Observable state representation. Each character state captures only what is externally observable—actions, expressions, posture, and tone—excluding psychological or symbolic interpretation. Outputs use a minimal structured representation listing only important characters for each scene.

Visualization. To support interactive exploration, we render the extracted dataset using a timeline swimlane view.

1. Timeline Swimlane View. A global *character–scene matrix* showing all characters (rows) across all scenes (columns). This view offers a high-level visualization of appearance frequency, co-occurrence structure, and participation patterns, enabling users to quickly identify which characters move together across the narrative. In a production setting, the same view can support assistant directors, continuity supervisors, and wardrobe/makeup teams by exposing character availability, scene grouping, co-presence, and state-continuity constraints across the shooting plan.

E Responsible NLP Checklist Details

E.1 Artifacts, Licenses, and Intended Use

Table E.1 summarizes the external benchmarks and compared systems used in the experiments. We use these artifacts for research evaluation only and do not redistribute original benchmark texts, screenplay sources, model weights, or API credentials with this paper. The Hybrid RAG baseline is implemented locally from standard lexical–semantic retrieval components and is therefore not listed as a separate external artifact.

E.2 Reproducibility and Decoding Settings

Table E.2 records the decoding and evaluator settings used for the experiments. Open-source backbone inference uses an 8 NVIDIA A100-SXM4-80GB GPU node (81,920 MiB per GPU) with NVIDIA driver 535.129.03. The exact number of GPUs used by a given model depends on the checkpoint size and tensor-parallel deployment, and GPU hours are not used as an evaluation metric.

For reproducibility, we plan to release the implementation, prompts, and configuration files needed to reproduce the reported experiments upon publication, subject to the licenses and access terms of the underlying datasets and model providers.

Query-time budget. For NKW, every query begins with a system-triggered multi-channel recall step. Because this first step is mandatory rather than agent-selected, it is excluded from the tool-use statistics in Section 5.2. The forced step consists of `entity_search`, `entity_lookup`, `relation_search`, and `atomic_fact_search`, together with `narrative_semantic_search`. The reported statistics count at most two subsequent agent-selected rounds with no more than five tool calls per round. Baselines use the shared 600-token chunking setup described in Section 4.

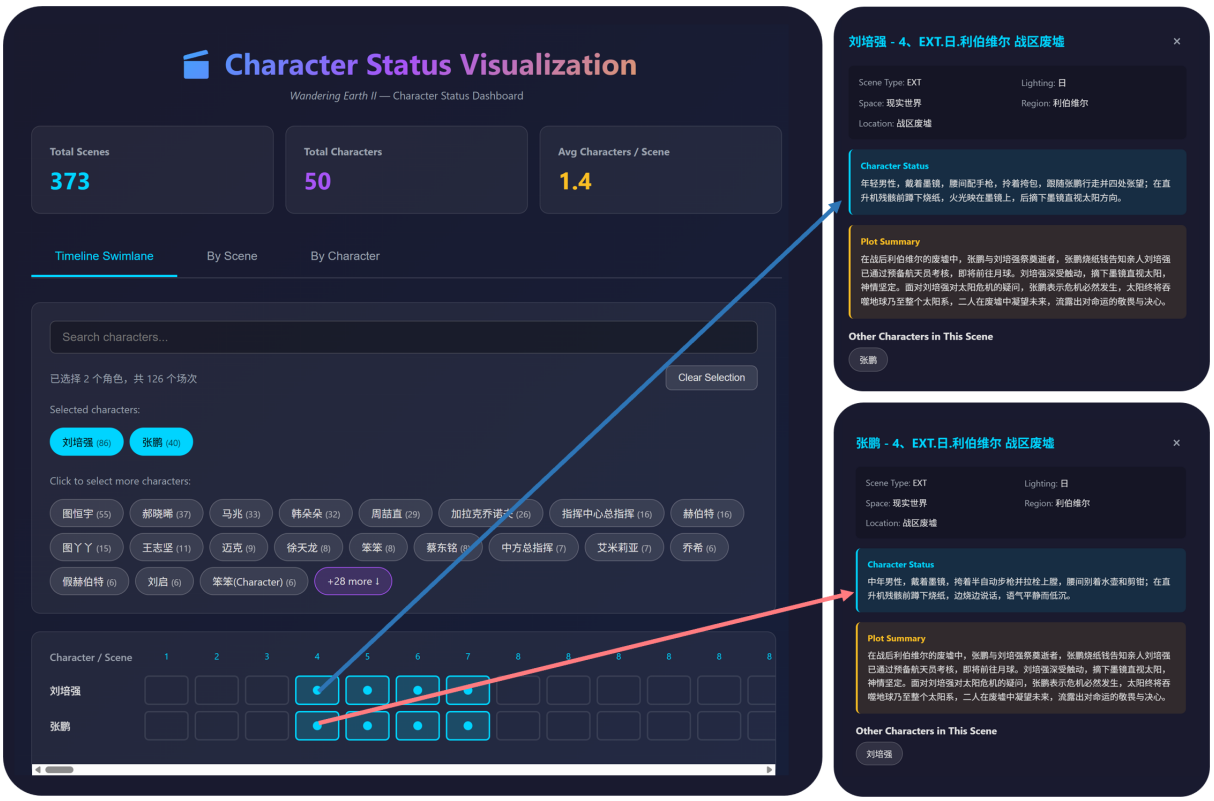

Figure D.1: **Timeline Swimlane View.** A screenplay-wide character–scene matrix enabling global inspection of appearance patterns.

Table E.1: External artifacts used in the experimental evaluation.

Artifact	Role	Source / citation	Use in this paper
STAGE	Benchmark	Tian et al. (2026)	Narrative and screenplay QA evaluation; raw screenplay text is not redistributed.
FairytaleQA	Benchmark	Xu et al. (2022)	Narrative QA evaluation over children's stories.
QuALITY	Benchmark	Pang et al. (2022)	Long-input multiple-choice QA evaluation.
GraphRAG	Compared system	Edge et al. (2024)	Community-summary graph-RAG baseline.
LightRAG	Compared system	Guo et al. (2025)	Lightweight graph-augmented retrieval baseline.
HippoRAG	Compared system	Gutiérrez et al. (2024)	Associative graph-retrieval baseline.
A-RAG	Compared system	Du et al. (2026)	Agentic retrieval-interface baseline.

Table E.2: Reproducibility settings for generation and evaluation.

Stage	Model / interface	Temperature	Top-p	Max tokens	Notes
Open-source QA generation	Qwen3 / Llama-3.1 deployment	0.7	0.95	24,000	Samples five candidate answers per question for Pass@5; the higher temperature is used to avoid duplicate samples.
Closed-source QA generation	GPT-5.5 API	Provider default	Provider default	Provider default	Included as a closed-source reference under the provider's standard access conditions.
Main semantic QA evaluation	DeepSeek-V4 API	0.0	1.0 or default	512	Deterministic LLM-as-a-judge pass; JSON output mode is used when supported by the provider.
Evaluator stochasticity test	DeepSeek-V4 API	0.1	1.0 or default	512	Ten independent evaluation passes on random 200-question subsets; a small nonzero temperature exposes borderline-answer variance.
Multiple-choice evaluation	Exact option matching	N/A	N/A	N/A	QuALITY is scored by selected-option accuracy rather than semantic judging.