BOOKMARKS: Checkpoint-Accelerated Active Grounding for Role-playing

Anonymous EMNLP submission

Abstract

Memory systems are critical for role-playing agents (RPAs) to maintain long-horizon consistency. However, existing RPA memory methods (e.g., profiling) mainly rely on incremental summarization, whose compression inevitably discards important details. To address this issue, we propose a search-based memory framework called **BOOKMARKS**, which actively initializes, maintains, and updates task-relevant **bookmarks** for the current task (e.g., character acting). A bookmark is structured as the **answer** to a **question** at a specific **point** in the storyline, serving as knowledge checkpoints. For each current task, BOOKMARKS proposes search questions and goes through storyline for answers. If a proposed question matches an existing bookmark, the search will start from that checkpoint instead of the storyline beginning, which accelerates searching. The synchronized bookmark will also represent a closer checkpoint for faster future search. Compared with incremental summarization, BOOKMARKS offers **(1) active grounding** for capturing task-specific details and **(2) passive updating** to avoid unnecessary computation. Our implementation supports concept, behavior, and state searches, each powered by an efficient synchronization method. BOOKMARKS significantly outperforms RPA memory baselines on 85 characters from 16 artifacts, demonstrating the effectiveness of search-based memory for RPAs.[1]

1 Introduction

Role-playing agents (RPAs) (Chen et al., 2024b,c) are expected to predict actions or utterances that remain faithful to characters throughout storyline by precisely capturing character information and dynamics, such as states and behaviors. Existing methods support such memory systems by either retrieving relevant past behaviors (e.g., retrieval-augmented generation) or iteratively updating character profiles. A common weakness of both approaches is that only part of the preceding storyline can reach the grounding stage, either due to the filtering mechanism in retrieval or the compression of details in profiling.

Figure 1: BOOKMARKS grounds role-playing by actively searching for useful information from the preceding storyline, while passively updating previous search results for efficiency.

In contrast, search-based grounding (Jin et al., 2025) can utilize the full preceding storyline by actively collecting important information to ground character behaviors in the current scene. A naive implementation is to let RPAs write search queries (e.g., *"How does the character respond to danger?"*) based on the current scene, and then search the preceding storyline for answers. These answers provide precise grounding information based on the full history to support character action prediction.

However, search-based grounding incurs high computational cost because every query requires reading the storyline from the beginning to ensure lossless search. Our observation is that many human readers do not revisit the whole book for certain information, but instead leave **bookmarks** as information-checking points, either physically or in memory (e.g., *"the protagonist's location in Chapter 4"*). We are thus inspired to propose an efficient search-based memory framework, **BOOKMARKS**.

[1]The code and data will be released for reproduction.

Specifically, BOOKMARKS imitates human reading habits by maintaining a pool of bookmarks inserted at different positions in the storyline. Each bookmark contains three values: (1) Query q: what is being searched; (2) Answer y: the answer to q; (3) Synchronization position p: the stage of the storyline where y is valid (e.g., *"Chapter 4"*). In summary, a bookmark represents search-style grounding information (q, y) at a specific time point.

Based on this data structure, BOOKMARKS grounds RPAs in three steps: (1) Proposal: observe the current scene and write queries that are potentially beneficial for RPA grounding; (2) Matching: find an identical or relevant bookmark from the existing pool. If matched, synchronize the matched bookmark with its answer to the current time point; otherwise, create an empty bookmark at the beginning of the storyline and synchronize it; (3) Grounding: use information in nearby bookmarks to ground RPA acting. We present a running example of BOOKMARKS in Figure 1, with more details in Figure 2. In implementation, BOOKMARKS supports three search types: (1) Entity: search entity definitions from the preceding storyline by locating their appearances; (2) State: obtaining current character states by incrementally updating answers through the storyline; (3) Behavior: deriving character behaviors from past conditional actions.

From a methodological perspective, BOOKMARKS provides a stronger alternative to incremental profiling. We can view conventional profile updating as a special case of BOOKMARKS, it remains unaware of which information is important for grounding the current scene, and updates all information together, including information that may not be reused in the future. In contrast, BOOKMARKS supports **active grounding** to search for useful information and **passive updating** to update bookmarks only when needed.

We test BOOKMARKS on multiple role-playing benchmarks, evaluated based on the likelihood of reproducing the 15.2K original actions in total from 85 characters across 16 artifacts. We find that BOOKMARKS outperforms incremental profiling and retrieval-based grounding, especially on long-horizon-dependent storylines such as *"Death Note"* and *"A Game of Thrones"*, demonstrating the advantage of active grounding. We further analyze the match rate and saved computational cost, showing a significant acceleration with a hit rate above 90%, saving over 70% search computation cost. Our ablation further validates that the match-and-derive mechanism achieves comparable performance to searching from the beginning of the storyline.

In conclusion, BOOKMARKS contributes to both (1) performance, by introducing active grounding to improve RPA performance through useful information retrieved from the whole storyline, and (2) efficiency, by maintaining a bookmark pool that accelerates synchronization via passive updating.

2 Background and Related Work

With the rapid development of LLMs' capability comes the ever-growing demand for more personalized interactions, to which role-playing agent (RPA) emerges as one of the central paradigms (Chen et al., 2024c; Tseng et al., 2024). These RPAs are expected to consistently produce in-character actions given story scene context, and the effective construction of such context is a *grounding* problem. As storylines lengthen, a static context would fail to provide enough relevant information, and thus dynamic *memory* systems become a key research direction. We accordingly cover **Evaluation**, **Training and Inference**, and **Memory** of role-playing agents.

Evaluation of RPAs splits by granularity into holistic judgment and per-action scoring. *Holistic judgment* aggregates over one of three character facets (identity, behavior, or knowledge) or bundles them. Identity is probed by psychiatric-style personality inventories (Wang et al., 2024b; Cheng et al., 2025), which rely on LLM-as-judge whose human alignment is known to be weak (Zhou et al., 2025). Behavior is tested in game, multi-agent, and social simulation testbeds (Yu et al., 2025; Zhou et al., 2024; Chen et al., 2024a). Knowledge is checked with factual or hallucination tests (Shen et al., 2023; Sadeq et al., 2024; Ahn et al., 2024). Broad-coverage suites bundle several facets into one composite score (Tu et al., 2024; Lu et al., 2025; He et al., 2025; Ding et al., 2025). *Per-action scoring*, in contrast, compares the next in-character action against a structured reference and is therefore facet-agnostic (Peng and Shang, 2025).

Training and Inference are two routes to adapt a model over time. The training-time route bakes character into parameters through supervised fine-tuning on character experiences (Shao et al., 2023) or large-scale synthetic dialogue (Moore Wang et al., 2024; Lu et al., 2024; Wang et al., 2025b; Yang et al., 2025a), multi-character LoRA

hot-swapping (Yu et al., 2024), boundary- and personality-aware data (Tang et al., 2024; Yang et al., 2025b; Ji et al., 2025), and reinforcement-learning recipes (Wang et al., 2025e; Fang et al., 2025; Liu et al., 2025). These methods often suffer from plot scarcity, out-of-distribution hallucination, and an inability to absorb facts the storyline adds after training. The inference-time route leaves the backbone frozen and inserts structure between scene and response, like role-aware reasoning (Tang et al., 2025), strategy-conditioned dialogue (Ye et al., 2025), retrieval-augmented exemplars (Wang et al., 2024a), and activation-level persona steering (Chen et al., 2025a). Memory belongs to this same family but specifically deals with what content to store.

Memory for RPAs focuses on what is stored (a compressed profile vs. an explicit structure) and how it's updated (statically before inference vs dynamically as scenes arrive). *Static-profile methods* compress the storyline into a single profile re-attached every scene, in forms ranging from executable-function profiles (Peng and Shang, 2025) to dialogue-recursive and topic-indexed summaries (Wang et al., 2025a; Zhong et al., 2024; Lu et al., 2023). These methods guess what to keep and what to discard, and could lose critical cues that are useful later. *Static-structure methods* fix a storage scheme ahead of time, like typed memory hierarchies (Yan et al., 2023; Sun et al., 2024), event-and-relation graphs (Ran et al., 2025; Li et al., 2024; Wang et al., 2025d), and mined if-then decision trees with distilled discriminators (Peng et al., 2026). However, only a small, scene-dependent subset is actually needed at a time. *Dynamic-profile retrieval* picks relevant profile entries for each scene (Chen et al., 2025b; Huang et al., 2024; Wang et al., 2026), but memory pool is fixed. Outside RP, state maintenance, world-model grounding, and long-form-story reasoning (Yoneda et al., 2024; Liu et al., 2024; Gurung and Lapata, 2025; Yi et al., 2025; Xia et al., 2025) share one principle: keep just enough state information to answer the next action-dependent question, and pull new information only when needed.

3 Our BOOKMARKS Framework

3.1 Preliminary

Storyline can be viewed as a sequence of actions from different characters, including special ones such as *"narration"* or *"environment"*. We denote the storyline as $A = [a_1, a_2, \cdots, a_N]$, where $N = |A|$. A character sequence $C = [c_1, c_2, \cdots, c_N]$ assigns each a_i to the character c_i who takes it.

Role-playing Agents RPAs aim to reproduce character behaviors in different situations by predicting the next action a_i from the current scene s_i and the target character c_i. In practice, s_i is a preprocessed local context before a_i, such as several preceding actions. Thus, an RPA can be viewed as a sampling policy:

$$a_i \sim \mathrm{RPA}(\cdot \mid s_i, c_i).$$

Grounding Stage The grounding stage augments the local scene with additional information before predicting a_i. This information may come from character profiles, retrieved examples, or storyline memory. This paper focuses on the data-driven setting, where useful grounding information is derived from the preceding storyline $[a_1, \cdots, a_{i-1}]$.

3.2 BOOKMARKS Framework

Figure 2 illustrates the overall workflow of BOOKMARKS. Given a target character c_i and scene s_i, BOOKMARKS constructs a grounded memory view from the preceding storyline, then feeds it to the RPA. Instead of compressing the whole history into a single profile, BOOKMARKS maintains a memory bank $\mathcal{B}$ of reusable *bookmarks*. Each bookmark tracks a task-relevant question over the storyline.

At each prediction step, BOOKMARKS first proposes useful grounding questions, then matches them with existing bookmarks or initializes new ones, and finally synchronizes only the selected bookmarks to the current story point. The synchronized answers are composed into a grounding context g_i, which is used for final prediction,

$$a_i \sim \mathrm{RPA}(\cdot \mid s_i, c_i, g_i).$$

Bookmark Data Structure A bookmark is represented as

$$b = (q, y, \tau, p, m),$$

where q is a natural-language question, y is the maintained answer, τ is the bookmark type, p is the synchronization point in the storyline, and m is optional auxiliary memory for type-specific updates. In this way, a bookmark stores the current answer y to q at story point p.

We use three types of bookmarks, which means $\tau \in$`{concept, state, behavioral}`. **Concept**

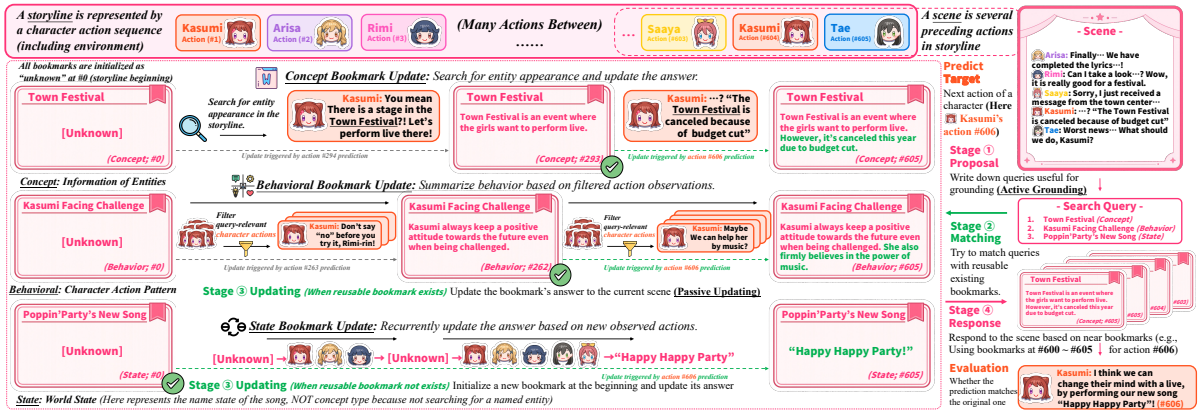

Figure 2: The role-playing grounding workflow of BOOKMARKS.

bookmarks track named entities or concepts introduced in the story. **State** bookmarks track evolving story states, such as locations, relationships, goals, or knowledge. **Behavioral** bookmarks track how a character tends to behave under certain conditions. This structure separates memory into explicit search targets, allowing different information to be maintained and reused independently.

3.3 Active Grounding

The first stage of BOOKMARKS proposes a small set of useful questions for grounding the current prediction. Given (s_i, c_i), the proposal module generates

$$Q_i = \{(q_i^{(1)}, \tau_i^{(1)}), \cdots, (q_i^{(K)}, \tau_i^{(K)})\},$$

where each question is paired with a bookmark type. In our implementation, these questions are generated by an LLM.

The proposal stage is active because it is conditioned on the current task. Rather than maintaining a fixed memory template for all scenes, BOOKMARKS asks what information is worth grounding for the current action prediction. This allows the framework to focus on task-relevant details while still creating bookmarks that may be reused later.

3.4 Passive Updating

After proposal, BOOKMARKS resolves each query by reusing an existing bookmark, deriving a new bookmark from a related one, or creating a fresh bookmark. The resulting active bookmarks are then synchronized to the current story point. This stage is passive because bookmarks are updated only when they are selected for the current grounding task, rather than continuously throughout the storyline.

Matching For each proposed query, BOOKMARKS searches the memory bank for bookmarks of the same type and determines whether an existing bookmark can support it. In implementation, we first apply a lightweight candidate filter, such as lexical overlap after stop-word removal, and then use an LLM to classify the relation between the proposed query and each candidate.

There are three possible outcomes. **Reuse** means the proposed query and the existing bookmark refers to the same maintained memory target, so the existing bookmark is activated directly. **Derive** means the existing bookmark is related but not identical, so a new bookmark is initialized from the parent bookmark's synchronized answer and inherits its story point. **New** means no suitable bookmark is found, so a fresh bookmark is created from the beginning of the storyline. This matching mechanism lets BOOKMARKS reuse stable memory slots while still branching into more specific memory targets once required.

Synchronization Once a bookmark is activated, it is synchronized from its stored point p to the current story point $i-1$ by processing only the unseen suffix $[a_{p+1}, \cdots, a_{i-1}]$. We denote the type-specific update operator U_τ as

$$(y', m') = U_\tau(q, y, m, [a_{p+1}, \cdots, a_{i-1}], C),$$

which updates the bookmark to $(q, y', \tau, i-1, m')$.

Different bookmark types define different ways of reading the unseen suffix $[a_{p+1}, \cdots, a_{i-1}]$, depending on what kind of answer the bookmark maintains. For a **state** bookmark, the suffix is used to move the answer from its previous value to the current story point, written as $y \leftarrow U_{\texttt{state}}(q, y, [a_{p+1}, \cdots, a_{i-1}])$. This is suitable for information that changes over time, such as locations, goals, relationships, or knowledge states.

Scene	**Tae:** I didn't expect all of you to sing.
	Rimi: I didn't either. I just heard Kasumi-chan singing, and...
	Arisa: She swept me up in her deceptive sparkly wave, too.
	Saaya: It just shows that we really are all feeling the same feelings right now.
	Rimi: It was a lot of fun.
Question	What'll be Kasumi's next action in response to the current scene?
Action	**Kasumi:** ... That's it! Guys, I've got it! This feeling... You know what this is, right? It's the real spirit of the festival!

Table 1: Instance examples in benchmarks. (Scenes include 10 preceding actions in real benchmarks)

For a **behavioral** bookmark, we use a verifier $V(q, s_t, a_t)$ to check whether action a_t under local scene s_t directly grounds the queried behavior q. The bookmark then keeps only verified evidence, written as $m \leftarrow m \cup \{a_t : V(q, s_t, a_t) = 1\}$, and updates the answer based on the accumulated evidence m. For a **concept** bookmark, we use a matcher $M(q, a_t)$ to locate actions that mention or instantiate the queried concept, written as $\{a_t : M(q, a_t) = 1\}$, and update the answer from their local contexts. In our implementation, V can be an LLM or a distilled binary classifier, while M can be approximated by keyword matching. These update rules make synchronization incremental, since each activated bookmark only reads the new suffix after its previous synchronization point. Combined with active proposal, BOOKMARKS therefore updates only the memory needed for the current prediction.

Grounding Finally, BOOKMARKS constructs the grounding context g_i from the synchronized answers of active bookmarks, together with nearby bookmarks whose synchronization points are close to the current story point. Active bookmarks provide information selected for the current prediction, while nearby bookmarks supply recently maintained context that may still support local continuity. In this way, BOOKMARKS augments the local scene with reusable, task-specific memory grounded in the preceding storyline.

4 Benchmark

Datasets To validate the advantage of BOOKMARKS, we use existing sequentialized storylines as the resource for benchmarking RPAs. Specifically, we use the *Fine-grained Fandom Benchmark* and *Bandori Conversational Benchmark* (Peng et al., 2026), which process well-known artifacts into action sequences. The Fine-grained Fandom Benchmark includes 45 characters from 8 artifacts with 20,778 actions from benchmarked characters, while the Bandori Conversational Benchmark includes 40 characters from 8 band stories of the *"BanG Dream! Project"* with 7,866 actions from benchmarked characters. Given a target character, RPAs are evaluated by reproducing the character's next action from the preceding storyline, where the local scene is represented by the preceding 10 actions in the benchmark. Table 1 shows an example instance with only 5 preceding actions displayed for space.

Criterion For each character, we follow the established benchmarking process by using the first half of the character's actions as observed storyline history and evaluating whether RPAs can reproduce actions in the second half. At each prediction point, the model can observe all preceding actions in the storyline, while the target action itself is held out for evaluation, resulting in 15.2K evaluated instances in total. Next-action prediction provides an objective benchmark for role-playing because each prediction can be compared against the original storyline action, avoiding the ambiguity of subjective scoring (e.g., scoring from 1 to 10). After an RPA predicts an action, we compare it with the original reference using key action match (AM), which judges whether the key move of the predicted action matches the reference. We use `gpt-4.1` as the judge because of its reliability for this simple discrimination task. We also manually validate its precision, finding that 483 of 500 cases (96.6%) match human judgments, which indicates the reliability of the experimental results.

5 Experiment

5.1 Baselines and Implementation Details

We compare BOOKMARKS with representative grounding methods that differ in how they access and maintain storyline information.

- **Vanilla** predicts character actions without additional grounding context, relying only on the parametric knowledge of the RPA.
- **Retrieval-based In-Context Learning** (Wang et al., 2024a) **(RICL)** retrieves top-k similar past scene-action pairs as grounding examples for the current scene.
- **Extract-and-Aggregate (ETA)** (Wang et al., 2025c) incrementally maintains a single character profile by summarizing new observations and aggregating them into the previous profile.

Fandom	Haruhi	K-On!	FMA	JOJO	AGOT	ATLA	DN	SxF	Avg.
Vanilla	20.36	24.29	29.18	25.66	33.87	31.91	20.89	26.35	26.56
RICL	25.42	23.37	30.33	27.32	37.64	31.02	19.91	25.20	27.53
ETA	22.96	26.71	31.48	26.93	35.78	32.54	21.34	26.71	28.06
3-way ETA	24.01	27.81	34.73	21.91	38.62	33.62	19.77	28.80	28.66
BOOKMARKS	23.56	27.58	33.65	28.53	39.35	34.58	24.76	28.26	**30.03**
Bandori	**PoPiPa**	**AG**	**PasuPare**	**Roselia**	**HHW**	**Monica**	**RAS**	**MyGO**	**Avg.**
Vanilla	41.22	41.26	41.29	37.38	38.54	44.62	38.90	30.79	39.25
RICL	41.19	42.29	41.37	40.19	41.61	47.35	39.86	32.62	40.81
ETA	41.83	42.46	46.45	41.01	42.58	46.47	41.08	33.94	41.98
3-way ETA	44.75	42.81	44.35	40.76	44.77	49.15	40.33	33.56	42.56
BOOKMARKS	46.84	44.93	46.72	42.62	47.58	51.05	42.01	34.46	**44.53**

Table 2: RP performance (Key Action Match Rate) comparison on Fandom and Bandori benchmarks.

- **3-way ETA** is an intermediate profiling baseline between ETA and BOOKMARKS. It splits the character profile into three fixed sections, *experience*, *state*, and *behavior*. We engineer the prompt to control each sections about one-third length of ETA profiles during update. For each prediction, the RPA actively selects which sections are useful for grounding, then only the selected sections are synchronized through ETA-style recurrent updating. This baseline keeps the profiling formulation of ETA, but introduces the active proposal and passive update philosophy of BOOKMARKS.

Methodological Comparison RICL retrieves only a small subset of past scene-action pairs, so it observes only partial storyline evidence and treats retrieved examples as isolated instances rather than a synchronized memory over narrative development. ETA maintains persistent character memory, but compresses all information into a single profile and updates it through generic aggregation. 3-way ETA provides a more direct methodological comparison with BOOKMARKS because it tests whether simply decomposing ETA into fixed profile sections and updating them selectively can already improve grounding and efficiency. The comparison between 3-way ETA and BOOKMARKS therefore isolates the value of flexible bookmark queries, type-specific synchronization, and reusable search-style memory beyond conventional profiling.

Implementation For all methods, we use `gpt-5.1` as the final response model because of its outstanding role-playing performance (Shi et al., 2026). The same model is also used for ETA and 3-way ETA profile updating, while `gpt-5.4-mini` is used for auxiliary calls (except for behavior bookmark updates) in BOOKMARKS and other methods. For RICL, we retrieve 8 scene-action examples for grounding based on scene embedding similarity encoded by `qwen3-embedding-8b`. For BOOKMARKS, we propose 5 bookmarks for each prediction and include nearby bookmarks within 5 action distances. For behavioral bookmark updating, we use a distilled `deberta-v3-base` classifier (He et al., 2021) as the verifier V to filter relevant action, trained from `gpt-5.4-mini` labels on an 80% split of 20K instances generated by running BOOKMARKS on the observed storyline portion. Validating the discriminator on the rest 20% set shows around 90% consistency with the teacher model. The prompts can be found in Appendix C.

5.2 Main Results

We present the main results in Table 2. Overall, BOOKMARKS achieves the best average performance across both Fandom and Bandori benchmarks, demonstrating the effectiveness of search-based storyline memory for role-playing. The improvement over Vanilla confirms the need for explicit grounding beyond parametric character knowledge, while the improvement over RICL shows that synchronized memory is more reliable than retrieving isolated examples that cover only partial storyline evidence and weaken narrative sequentiality. ETA improves over Vanilla by maintaining persistent character profiles, and 3-way ETA further strengthens this profiling baseline by decomposing the profile into experience, state, and behavior sections with selective updating. This shows that even a fixed-profile variant can benefit from the active proposal and passive update philosophy of BOOKMARKS.

Moreover, BOOKMARKS still outperforms 3-way ETA, indicating that flexible bookmark queries, type-specific synchronization, and reusable search-style memory provide advantages beyond simply splitting a profile into fixed sections.

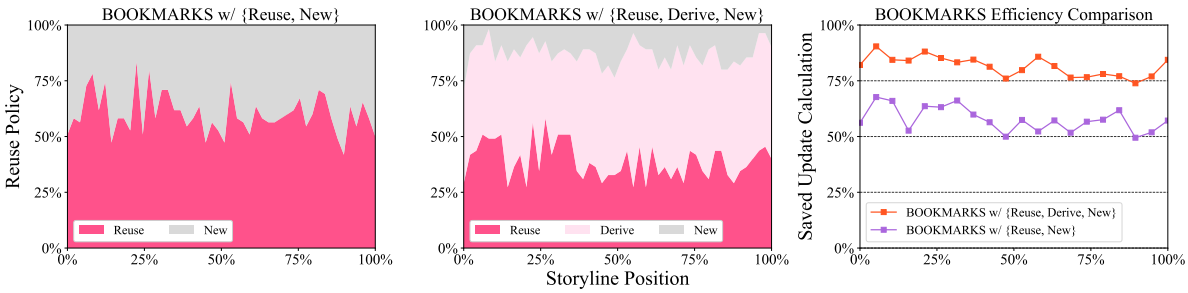

Figure 3: Reuse proportion of different policy (computed by hit rate) and efficiency analysis of BOOKMARKS.

PoPiPa *(S1)*	Kasumi	Arisa	Rimi	Tae	Saaya	Avg.
BOOKMARKS	51.50	42.24	44.44	48.31	47.73	46.84
w/o Derive	52.69	47.41	45.68	43.82	47.73	47.47
+w/o Reuse	55.69	44.83	50.62	46.06	40.91	47.62
w/o Near Notes	49.10	43.97	45.66	42.70	45.45	45.38
→ IBU	50.30	41.84	41.98	43.82	46.59	44.91

Table 3: Ablation Study on BOOKMARKS framework. Incremental Behavior Update (IBU): Apply incremental updating for behavior.

This design is especially suitable for long-horizon storylines discussed in the introduction, such as *"Death Note"* and *"A Game of Thrones"*, where subtle earlier details and evolving states can become important much later, supporting our claim that RPAs require a solid and efficient memory system for long-horizon consistency.

5.3 Reusing Hit Rate

Reusing and deriving bookmarks are central to the efficiency advantage of BOOKMARKS, since they allow the framework to avoid synchronizing every proposed query from the beginning of the storyline. We therefore report the hit rate in Figure 3, where a hit refers to either directly reusing an existing bookmark or deriving a new bookmark from a previously synchronized one. The reuse-only variant already achieves a substantial hit rate, suggesting that many grounding demands recur across nearby story positions and can be served by maintained memory slots. This confirms the value of reuse itself: once a bookmark has been synchronized, it can repeatedly provide useful grounding without paying the full cost of rereading the preceding storyline.

Introducing derivation further improves the effective hit rate. Unlike strict reuse, derivation does not require the new query to be equivalent to an existing bookmark. Instead, it allows BOOKMARKS to initialize a related but distinct bookmark from a synchronized answer that already captures relevant storyline information. This makes the memory bank more flexible, since related grounding needs can branch from existing bookmarks rather than being recomputed from scratch. The hit rate also fluctuates across storyline positions, which reflects changes in scenes, characters, and narrative focus. When the story shifts to a new situation, BOOKMARKS naturally proposes more new bookmarks; when the narrative remains within related states, concepts, or behaviors, reuse and derivation become more frequent. Therefore, the fluctuation represents a sign that the bookmark pool adapts to storyline dynamics. Overall, the observed hit rate and saved update computation show that BOOKMARKS can maintain task-relevant memory while substantially reducing synchronization cost.

5.4 Ablation Study

We further conduct ablation experiments to examine the effect of key design choices in BOOKMARKS. For efficiency, the ablation is performed on the PoPiPa dataset, *BanG Dream! Poppin'Party Band Story 1*, which contains five benchmarked characters, as shown in Table 3. Removing derivation or further removing reuse maintains comparable performance, but these variants reduce the efficiency advantage of BOOKMARKS, since more bookmarks must be initialized or synchronized from earlier storyline positions, consistent with the efficiency analysis in Figure 3.

Removing near notes lowers performance, suggesting that recently synchronized bookmarks provide useful local continuity beyond the bookmarks actively proposed for the current prediction. Replacing evidence-based behavioral updating with incremental behavior update also hurts performance, indicating that behavioral memory is better maintained by collecting verified action evidence rather than treating behavior as a continuously transitioned state. Overall, these results support the main design of BOOKMARKS, where active proposal identifies useful memory targets, passive updating accelerates synchronization, and type-specific update preserve grounding quality.

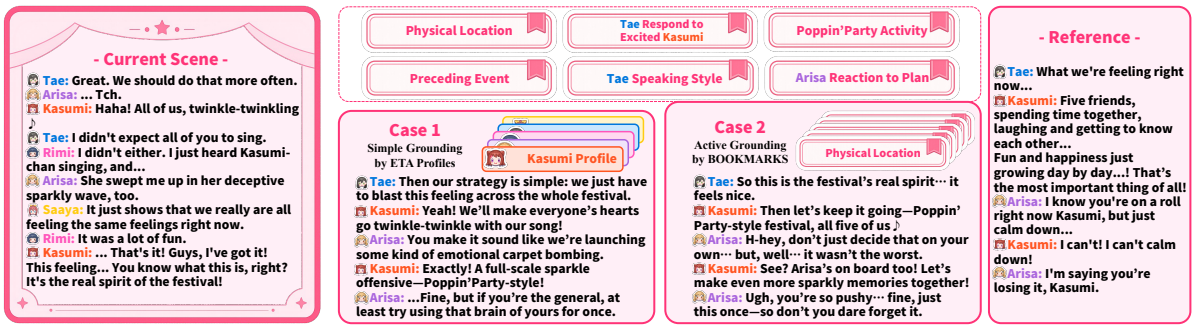

Figure 4: Case study on comparing BOOKMARKS and conventional profiling, based on multiple action prediction.

PoPiPa *(E321)*	Kasumi	Arisa	Rimi	Tae	Saaya	Avg.
Vanilla	36.52	45.54	33.33	46.67	33.33	39.79
RICL	42.61	51.79	42.86	43.33	36.67	43.45
ETA	40.87	49.11	42.86	46.67	36.67	43.95
BOOKMARKS	44.35	54.46	47.62	53.33	40.00	**48.70**

Table 4: Live storyline *(BanG Dream! Girls Band Party! Event 321, released on Feb 8th, 2026)* results.

5.5 Live Evaluation

We further evaluate the performance of BOOKMARKS on storylines after the knowledge cutoff of the experimented models. Specifically, we select the 321st event storyline of *"BanG Dream! Girls Band Party!"*, with its Japanese version released on Feb. 8th, 2026 (which is after the knowledge cutoff of `gpt-5.1` and `gpt-5.4-mini`). As shown in Table 4, BOOKMARKS achieves the best overall performance in this live updating setting, outperforming Vanilla, RICL, and ETA across most characters. This result indicates that BOOKMARKS can more effectively organize and synchronize information from the observed storyline when the model cannot rely on memorized familiarity with the target narrative. This result supports the robustness of bookmark-based grounding on newly released storylines, where RPAs must track evolving states and behaviors from the provided context.

5.6 Case Study: Multi-action Generation

Beyond next single-action prediction, we further conduct a case study with multi-action generation to examine whether BOOKMARKS can support longer continuations. This setting is harder to evaluate automatically, since valid continuations may differ from the reference in wording, speaker allocation, or local ordering while still preserving the same character dynamics and storyline direction. As shown in Figure 4, the current scene centers on Poppin'Party collectively realizing the *"real spirit of the festival."* With simple ETA profile grounding, the prediction captures some surface-level character traits, such as Kasumi's enthusiasm and Arisa's reluctant response, but shifts the continuation toward a generic strategic discussion. This weakens the emotional flow of the scene and moves away from the shared feeling among the five members.

In contrast, BOOKMARKS grounds the generation with multiple reusable memory anchors, such as the physical location, preceding event, Poppin'Party activity, Tae's speaking style, and Arisa's reaction to Kasumi's plans. These bookmarks help preserve the local narrative focus and produce a continuation closer to the reference: Tae recognizes the festival's spirit, Kasumi turns it into a forward-looking group moment, and Arisa responds with affectionate resistance. This example illustrates both what our metrics capture and what goes beyond them: BOOKMARKS better matches character behavior, group interaction, and storyline development, while also maintaining emotional continuity and ensemble structure across several actions, which is difficult to fully measure through single-action prediction alone.

6 Conclusion and Future Work

We propose BOOKMARKS, an efficient search-based memory framework for role-playing agents that maintains task-relevant bookmarks as synchronized question-answer pairs along the storyline. BOOKMARKS provides a practical alternative to retrieval-only grounding and incremental profile compression, improving both long-horizon consistency and memory efficiency. Future work includes extending BOOKMARKS to recognition management for tracking which character knows what, combining bookmark memory with self-refinement, and designing more customized update policies for different query types and narrative structures.

Limitations

This work focuses on establishing BOOKMARKS as a general search-based memory framework for long-horizon role-playing, while leaving several extensions for future exploration. First, the current framework mainly maintains storyline-level states, behaviors, and concepts, and can be further extended to finer-grained recognition management, such as explicitly tracking which character knows which information at each story point. Second, BOOKMARKS is currently used as a grounding module before generation, while future work may integrate it with self-refinement so that agents can revise generated actions when bookmark evidence reveals inconsistencies. Finally, although we design type-specific update procedures for concept, state, and behavioral bookmarks, more customized update policies could be developed for different query types, characters, and narrative structures.

References

Jaewoo Ahn, Taehyun Lee, Junyoung Lim, Jin-Hwa Kim, Sangdoo Yun, Hwaran Lee, and Gunhee Kim. 2024. Timechara: Evaluating point-in-time character hallucination of role-playing large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3291–3325.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Gao Xing, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, and Fei Huang. 2024a. Socialbench: Sociality evaluation of role-playing conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2108–2126.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. 2024b. From persona to personalization: A survey on role-playing language agents. *Preprint*, arXiv:2404.18231.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, et al. 2024c. From persona to personalization: A survey on role-playing language agents. *Transactions on Machine Learning Research*.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. 2025a. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*.

Weishu Chen, Jinyi Tang, Zhouhui Hou, Shihao Han, Mingjie Zhan, Zhiyuan Huang, Delong Liu, Jiawei Guo, Zhicheng Zhao, and Fei Su. 2025b. Moom: Maintenance, organization and optimization of memory in ultra-long role-playing dialogues. *arXiv preprint arXiv:2509.11860*.

Xilong Cheng, Yunxiao Qin, Yuting Tan, Zhengnan Li, Ye Wang, Hongjiang Xiao, and Yuan Zhang. 2025. Psymem: Fine-grained psychological alignment and explicit memory control for advanced role-playing llms. *arXiv preprint arXiv:2505.12814*.

Hang Ding, Qiming Feng, Dongqi Liu, Qi Zhao, Tao Yao, Shuo Wang, Dongsheng Chen, Jian Li, Zhenye Gan, Jiangning Zhang, et al. 2025. Rolermbench & rolerm: Towards reward modeling for profile-based role play in dialogue systems. *arXiv preprint arXiv:2512.10575*.

Feiteng Fang, Ting-En Lin, Yuchuan Wu, Xiong Liu, Xiang Huang, Dingwei Chen, Jing Ye, Haonan Zhang, Liang Zhu, Hamid Alinejad-Rokny, et al. 2025. Charm: Character-based act-adaptive reward modeling for advanced role-playing language agents. *arXiv e-prints*, pages arXiv–2505.

Alexander Gurung and Mirella Lapata. 2025. Learning to reason for long-form story generation. *CoRR*, abs/2503.22828.

Kai He, Yucheng Huang, Wenqing Wang, Delong Ran, Dongming Sheng, Junxuan Huang, Qika Lin, Jiaxing Xu, Wenqiang Liu, and Mengling Feng. 2025. Crab: A novel configurable role-playing llm with assessing benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15030–15052.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Le Huang, Hengzhi Lan, Zijun Sun, Chuan Shi, and Ting Bai. 2024. Emotional rag: Enhancing role-playing agents through emotional retrieval. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*, pages 120–127. IEEE.

Ke Ji, Yixin Lian, Linxu Li, Jingsheng Gao, Weiyuan Li, and Bin Dai. 2025. Enhancing persona consistency for llms' role-playing using persona-aware contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26221–26238.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, et al. 2024. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786.

Cheng Liu, Yifei Lu, Fanghua Ye, Jian Li, Xingyu Chen, Feiliang Ren, Zhaopeng Tu, and Xiaolong Li. 2025. Cogdual: Enhancing dual cognition of llms via reinforcement learning with implicit rule-based rewards. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27295–27324.

Zeyang Liu, Xinrui Yang, Shiguang Sun, Long Qian, Lipeng Wan, Xingyu Chen, and Xuguang Lan. 2024. Grounded answers for multi-agent decision-making problem through generative world model. *Advances in Neural Information Processing Systems*, 37:46622–46652.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Junru Lu, Jiazheng Li, Guodong Shen, Lin Gui, Siyu An, Yulan He, Di Yin, and Xing Sun. 2025. Rolemrc: A fine-grained composite benchmark for role-playing and instruction-following. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21008–21030.

Keming Lu, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7828–7840.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, et al. 2024. Rolellm: benchmarking, eliciting, and enhancing role-playing abilities of large language models. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777.

Letian Peng and Jingbo Shang. 2025. Codifying character logic in role-playing. In *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025) Poster Session*. Poster presentation, NeurIPS 2025.

Letian Peng, Kun Zhou, Longfei Yun, Yupeng Hou, and Jingbo Shang. 2026. Deriving character logic from storyline as codified decision trees. *arXiv preprint arXiv:2601.10080*.

Yiting Ran, Xintao Wang, Tian Qiu, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2025. BOOKWORLD: from novels to interactive agent societies for story creation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 15898–15912. Association for Computational Linguistics.

Nafis Sadeq, Zhouhang Xie, Byungkyu Kang, Prarit Lamba, Xiang Gao, and Julian McAuley. 2024. Mitigating hallucination in fictional character role-play. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14467–14479.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187.

Tianhao Shen, Sun Li, Quan Tu, and Deyi Xiong. 2023. Roleeval: A bilingual role evaluation benchmark for large language models. *arXiv preprint arXiv:2312.16132*.

Wenlong Shi, Jianxun Lian, Mingqi Wu, Haiming Qin, Mingyang Zhou, Xing Xie, Naipeng Chao, and Hao Liao. 2026. Personaarena: Dynamic simulation for evaluating and enhancing persona-level role-playing in large language models. *arXiv preprint arXiv:2605.17044*.

Libo Sun, Siyuan Wang, Xuanjing Huang, and Zhongyu Wei. 2024. Identity-driven hierarchical role-playing agents. *arXiv preprint arXiv:2407.19412*.

Yihong Tang, Kehai Chen, Muyun Yang, Zhengyu Niu, Jing Li, Tiejun Zhao, and Min Zhang. 2025. Thinking in character: Advancing role-playing agents with role-aware reasoning. *arXiv preprint arXiv:2506.01748*.

Yihong Tang, Jiao Ou, Che Liu, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Erabal: Enhancing role-playing agents through boundary-aware learning. *arXiv preprint arXiv:2409.14710*.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. 2024. Two tales of persona in llms: A survey of role-playing and personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16612–16631.

Quan Tu, Shilong Fan, Zihang Tian, Tianhao Shen, Shuo Shang, Xin Gao, and Rui Yan. 2024. Charactereval: A chinese benchmark for role-playing conversational agent evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11836–11850.

Kai Wang, Haoyang You, Yang Zhang, and Zhongjie Wang. 2026. Memory-driven role-playing: Evaluation and enhancement of persona knowledge utilization in llms. *arXiv preprint arXiv:2603.19313*.

Liang Wang, Nan Yang, and Furu Wei. 2024a. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025a. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193.

Xiaoyang Wang, Hongming Zhang, Tao Ge, Wenhao Yu, Dian Yu, and Dong Yu. 2025b. Opencharacter: Training customizable role-playing llms with large-scale synthetic personas. *arXiv preprint arXiv:2501.15427*.

Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Wei Wang, Yanghua Xiao, and Shuchang Zhou. 2025c. Coser: Coordinating llm-based persona simulation of established roles. *CoRR*, abs/2502.09082.

Xintao Wang, Yunze Xiao, Jen-tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, Jiangjie Chen, Cheng Li, and Yanghua Xiao. 2024b. Incharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1840–1873. Association for Computational Linguistics.

Yongjie Wang, Jonathan Leung, and Zhiqi Shen. 2025d. Rolerag: Enhancing llm role-playing via graph guided retrieval. *arXiv preprint arXiv:2505.18541*.

Zongsheng Wang, Kaili Sun, Bowen Wu, Qun Yu, Ying Li, and Baoxun Wang. 2025e. Raiden-r1: Improving role-awareness of llms via grpo with verifiable reward. *arXiv preprint arXiv:2505.10218*.

Haotian Xia, Hao Peng, Yunjia Qi, Bin Xu, Juanzi Li, Hou Lei, and Xiaozhi Wang. 2025. Storywriter: A multi-agent framework for long story generation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6559–6563.

Ming Yan, Ruihao Li, Hao Zhang, Hao Wang, Zhilan Yang, and Ji Yan. 2023. Larp: Language-agent role play for open-world games. *arXiv preprint arXiv:2312.17653*.

Bohao Yang, Dong Liu, Chenghao Xiao, Kun Zhao, Chen Tang, Chao Li, Lin Yuan, Yang Guang, and Chenghua Lin. 2025a. Crafting customisable characters with LLMs: A persona-driven role-playing agent framework. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20216–20240, Suzhou, China. Association for Computational Linguistics.

Tao Yang, Yuhua Zhu, Xiaojun Quan, Cong Liu, and Qifan Wang. 2025b. Psyplay: Personality-infused role-playing conversational agents. *arXiv preprint arXiv:2502.03821*.

Jing Ye, Lu Xiang, Yaping Zhang, and Chengqing Zong. 2025. Sweetiechat: A strategy-enhanced role-playing framework for diverse scenarios handling emotional support agent. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 4646–4669. Association for Computational Linguistics.

Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Shiyao Qian, Xinhang Yuan, Yi Xin, Yijin Wang, Jingqun Tang, Yuchen Li, et al. 2025. Score: Story coherence and retrieval enhancement for ai narratives. *arXiv preprint arXiv:2503.23512*.

Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R Walter. 2024. Statler: State-maintaining language models for embodied reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15083–15091. IEEE.

Pengfei Yu, Dongming Shen, Silin Meng, Jaewon Lee, Weisu Yin, Andrea Yaoyun Cui, Zhenlin Xu, Yi Zhu, Xingjian Shi, Mu Li, and Alex Smola. 2025. RPG-BENCH: evaluating large language models as role-playing game engines. *CoRR*, abs/2502.00595.

Xiaoyan Yu, Tongxu Luo, Yifan Wei, Fangyu Lei, Yiming Huang, Hao Peng, and Liehuang Zhu. 2024. Neeko: Leveraging dynamic lora for efficient multi-character role-playing agent. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12540–12557.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19724–19731.

Lingfeng Zhou, Jialing Zhang, Jin Gao, Mohan Jiang, and Dequan Wang. 2025. Personaeval: Are llm evaluators human enough to judge role-play? In *Second Conference on Language Modeling*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. 2024. Sotopia: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

A LLM Usage Statement

We used large language models (LLMs) only for copy-editing: improving grammar, clarity, and overall writing style to enhance readability. The LLMs did not contribute to or modify the scientific content, core ideas, methodology, or experimental results. The authors take full responsibility for the manuscript's final content and accuracy.

B Statistics

Table 5 presents the statistics of both benchmarks used in our experiments.

C Prompts

We place the prompts used in the main experiment and analyses in Figures 5, 6, and 7 for result reproduction. For AM scoring, we only reward exact match (*"A" selection*) with 100 scores and the other 0, even for partial matching.

D Character & Artifact Background Information

We provide brief descriptions of the artifacts and characters used in our experiments from Table 6 to Table 9.

Prompt #1: Proposal Stage

```
# Scene
{scene}

# Profile of {character}
{profile}

Now, it's {character}'s turn to make the next {predict_target}.
To precisely ground {character}'s {predict_target},
we need more information about the state of the current story.
The system supports three kinds of search queries:

1. concept search:
   - Used to look up the meaning or details of a concrete concept
     mentioned earlier in the story.
   - Must query a named concept or entity. (e.g., What is "..." ...?)
   - Example: What is "Star Beat" as mentioned in this conversation?

2. state search:
   - Used to iteratively update and recover the *current* state of the world
     up to this point in the story.
   - Never use chronical words like **before/after the scene** or **recent** in the query
     because the search is done recursively, always ask about **now**.
   - Example: Where are the band members right now?

3. behavioral search:
   - Used to recall or infer how a {character} tends to behave
     in a particular situation or under certain conditions, based on the story so far.
   - Behavioral queries should be phrased in a **general** way so that many past
     scenes can potentially match the condition, rather than referring to one
     extremely specific single moment. (simply speaking, make it fewer than 15 words.)
   - Example: How does Arisa usually react when someone makes an impulsive decision?

Search history:
{history}

Your task:
Propose {k} most crucial search queries as questions to check
before deciding {character}'s next {predict_target}.
You may freely use any of the three query types above, as long as
the query is useful for grounding {character}'s next {predict_target}.
**However, you should make sure each query is meaningful, targeting at
information not given in the scene.**

Important (for each query):
- Keep the question simple and concise, "unknown" in the history
  means a failed trial, don't try that direction again and
  **avoid even relevant queries** unless no other important information
  to gather, the storyline might just does not mention it.
- Don't focus only on the content, after the content to express
  is clear, search for how the character generally express it,
  especially given that the personality of the character will
  affect the content they will explicitly express.
- For the question, make sure there is no coreference involved:
  the question must explicitly mention all subjects and objects
  instead of using pronouns like "she", "they", "there", or "that".
- The question should be about the story world, characters, and events,
  not about the model or the querying system itself.
- When query tagged as "behavioral", keep the condition general enough that
  it could match multiple past scenes, not a single unique instance.
- For the query, you MUST assign a tag field with one of:
  "concept", "state", or "behavioral",
  indicating which search type this query is intended for.
  only tag as "concept" when searching a named concept or entity.
- Keep each query concise and high potential for information gathering.

Output as Python code in the following form:

```python
queries = [
    {{"query": "...", "tag": "..."}},
    {{"query": "...", "tag": "..."}},
    # ...
]
```
```

Prompt #2: Match Bookmarks

```
Tag: {new_bookmark["tag"]}

Existing bookmark query:
{old_bookmark["query"]}

Existing bookmark answer:
{old_bookmark["answer"]}

New bookmark query:
{new_bookmark["query"]}

Decide the relation from the existing bookmark to the new bookmark.

Return exactly one word:

reuse
- The two queries should share the same maintained bookmark slot.
- They ask for essentially the same information over time.

derive
- The existing bookmark should not be reused as the same slot.
- But its current synchronized answer can be used to initialize the new bookmark.

none
- The existing bookmark is not sufficiently relevant.

Be conservative:
- Use "reuse" only for near-equivalent maintained memory targets.
- Use "derive" for closely related but non-identical queries.
- Use "none" otherwise.
```

Figure 5: Prompts/Templates used in the main experiments for BOOKMARKS (1/3).

Prompt #3: Derive Answer

```
{scene_block}You are initializing a new bookmark from an
existing bookmark.

# Existing bookmark query
{old_bookmark["query"]}

# Existing bookmark answer
{old_bookmark["answer"]}

# New bookmark query
{new_bookmark["query"]}

Write an initial answer for the new bookmark using the existing bookmark answer as a basis.

Requirements:
- Be concise.
- Be conservative.
- Do not invent unsupported details.
- If the existing bookmark is not enough, output exactly: unknown

Return only the answer.
```

Prompt #4: State Update Check

```
Before the following scene,
Current answer to "{query}": "{answer}"

After the following scene:
{chunk}

Considering this scene, how should we treat the current answer to "{query}"?

Please answer with exactly one of:
- "reset"  -> the current answer is now outdated or contradicted and should be set to "unknown"
- "update" -> the scene triggers state transition or provides more accurate information so the answer should be updated
- "none"   -> the scene does not contain relevant update or information to affect the current answer
```

Prompt #5: State Update

```
Current maintained answer to "{query}":
"{answer}"

After the following scene:
{chunk}

The answer to "{query}" is detected to be updated.

Please directly output the new maintained answer concisely.
Important:
- Preserve the trace of important transitions instead of only giving the latest state.
- Record how the answer changed when the transition matters for future understanding.
- Keep it compact; do not summarize unrelated events.
- If the previous answer is still relevant, incorporate it as transition history.
- Do not explain your reasoning.

Output only the updated answer.
```

Prompt #6: Behavioral Filter

```
# Scene
{scene}
# Action by {character}
{action}

We are using the following behavioral query as a filter over past scenes:

# Query
{query}

Question:
Does THIS specific action by {character} in THIS scene provide direct evidence
for the behavior that the query is asking about?

Answer "yes" ONLY if BOTH of the following are true:
- The situation described in the query (characters, conditions, emotional state,
  context, etc.) actually holds in this scene.
- {character}'s action shown above is a concrete example of that behavior in
  this situation (not just vaguely related or partially matching).

If any required condition in the query is missing, unclear, or only weakly
related, answer "no".

Directly answer with a single word: yes or no.
```

Prompt #7: Behavioral Summary

```
You are given a list of filtered character actions for {character}

# Character action samples (chronologically ordered):
{action_list}

# Behavioral query:
{query}

Important:
- The action list is sorted chronologically from earliest to latest.
- If earlier and later actions appear to conflict, prioritize the pattern implied by the later actions.
- Focus ONLY on behavioral patterns grounded in the provided samples.

Your task:
Summarize how the character, {character}, behaves in a concise, general sentence or two.
Do NOT add new traits, motivations, or interpretations that are not supported by the actions.

Format:
Return ONLY the final summarized answer, no explanation.
```

Figure 6: Prompts/Templates used in the main experiments for BOOKMARKS (2/3).

Prompt #8: Get Concept Keyword

```
# Query\n{query}
Extract the keyword of the searched concept from the query for string matching search, directly output it.
```

Prompt #9: Concept Update

```
{keyword_appearance}

Based on the appearance of "{keyword}" above, update the answer to "{query}" below and directly output the new one.
Current Answer: {answer}
```

Prompt #10: Next Action Pred

```
# Background Summary
{grounding}
---
# Current Scene
{scene}
---
Infer the next action ({uttr_mark}) of {character} in the following scene:

Output in the following format:
```python
{{
    "reasoning": "...",
    "action": "...",
}}
```
```

Prompt #11: Next Actions Pred

```
{grounding}

Based on the features, infer {n_candidate} diverse potential next actions ({uttr_mark}) of {character} in the following scene:

{scene}

Output in the following format (same language as the given scene):
# Reasoning
...
# Actions
1. ...
2. ...
...
```

Prompt #12: AM Scoring

```
# Scene
{scene}

Your Response: {prediction}
Ground Truth: {action}

Score the action of {character} in the response based on the ground truth.
A: The ground truth entails the action of {character} in the response. (Following the same character logic.)
B: The ground truth is neutral to the action of {character} in the response. (Reflecting a different facet.)
C: The ground truth contradicts the action of {character} in the response. (Following a contradicted character logic.)

Output in json:
```json
{{
"reasoning": "...",
"score": "A/B/C"
}}
```
```

Figure 7: Prompts/Templates used in the main experiments for BOOKMARKS (3/3).

Fandom	Haruhi	K-On!	S×F	DN	FMA	JOJO	AGOT	ATLA
#Main Character	5	5	3	5	5	7	11	4
#Episode	28	57	116	108	108	152	73	61
#Action	991	2555	7688	5006	3349	2958	12073	8619
$\#\text{Action}_{\text{Main Character}}$	781	1882	3341	2738	1351	1578	4859	4248
#Avg. Action Length	12.15	10.51	11.73	13.13	12.58	11.81	12.42	11.58
Bandori	**PoPiPa**	**AG**	**PasuPare**	**Roselia**	**HHW**	**Monica**	**RAS**	**MyGO**
#Main Character	5	5	5	5	5	5	5	5
#Episode	20	20	20	20	20	20	25	41
#Action	1226	1053	968	1079	1122	1040	1183	2050
$\#\text{Action}_{\text{Main Character}}$	1080	914	791	873	827	966	795	1620
#Avg. Action Length	13.18	16.61	21.01	21.95	21.60	19.66	15.64	16.48

Table 5: Statistics of benchmarks (*Fine-grained Fandom Benchmark* and *Bandori Conversational Benchmark*) used in the experiments.

Artifact	**Concise Abstract**
The Melancholy of Haruhi Suzumiya	A lively school comedy and mystery centered on Haruhi, whose eccentric impulses unknowingly reshape reality. Ordinary club activities are mixed with supernatural incidents, while Kyon's pragmatic narration keeps the chaos grounded.
K-On!	A gentle slice-of-life story about the Light Music Club and its everyday musical activities. Rather than relying on major conflict, the narrative emphasizes friendship, routine, and gradual growth through practice and shared leisure.
Fullmetal Alchemist	A fantasy adventure about two brothers who seek to recover what they lost after a forbidden alchemical ritual. The story combines action and moral conflict, exploring sacrifice, human value, and the cost of power within an unstable political world.
JoJo's Bizarre Adventure (Part 3)	A supernatural journey across the world in which Jotaro and his companions confront enemies through Stand battles. The story is shaped by inventive abilities, tactical fights, and flamboyant interactions that mix tension with humor.
Spy × Family	A family comedy that follows a spy who forms a fake household to complete a covert mission. Espionage, action, and comedy are balanced with sincere emotional moments, as hidden identities gradually form real bonds.
Death Note	A psychological thriller in which a gifted student obtains a notebook that can kill anyone whose name is written in it. The story examines justice, morality, and ego through an escalating battle of intellect between brilliant rivals.
A Game of Thrones	A political fantasy about noble houses competing through alliances, betrayal, and war. Personal ambition and moral uncertainty unfold alongside a looming threat beyond the conflicts of human power.
Avatar: The Last Airbender	A coming-of-age adventure following Aang as he learns to master the elements and restore balance. The series blends humor, action, and emotional growth, with recurring themes of responsibility, forgiveness, and identity.

Table 6: Concise descriptions of artifacts used in our experiments for the Fine-grained Fandom Benchmark.

Band	Concise Description
Poppin'Party	A bright pop-rock band formed by high school friends, marked by energetic melodies and an earnest drive toward their dreams. Their story centers on friendship, first ambitions, and growth through practice, performance, and shared effort.
Afterglow	A straightforward rock band built on long-standing childhood friendships and a raw livehouse sound. Their narrative emphasizes loyalty, everyday honesty, and the tension between preserving old bonds and growing into new selves.
Pastel*Palettes	An idol-style unit whose cheerful image is supported by discipline, effort, and constant professional learning. Their arc highlights teamwork under pressure, backstage struggle, and the gap between polished performance and personal difficulty.
Roselia	A gothic, high-intensity band pursuing an ideal sound through ambition, discipline, and strict standards. Their story explores artistic pride, conflict under high expectations, and the trust needed to perform at the highest level.
Hello, Happy World!	A showy, joy-centered band that treats performance as a mission to make people smile. Playful chaos and comedic situations coexist with sincere warmth, as each member contributes to a shared belief in happiness.
Morfonica	A melodic rock band distinguished by violin-centered arrangements, refined textures, and sensitive emotional expression. Their narrative follows insecurity, talent gaps, and self-acceptance as the members build a shared musical identity.
RAISE A SUILEN	A powerful rock and electronic hybrid band built through deliberate recruitment, precision, and stage ambition. The story foregrounds professionalism, creative control, and the difficult cohesion of strong personalities pursuing the same peak.
MyGO!!!!!	An emotionally intense rock band shaped by conflict, vulnerability, and difficulty with honest communication. Their arc focuses on miscommunication, fragile belonging, and the catharsis of turning pain into music and connection.

Table 7: Concise band descriptions used in our experiments for the Bandori Conversational Benchmark.

Haruhi	Haruhi	An impulsive and hyperactive high school girl whose unusual worldview sets off many of the story's strange events.
	Kyon	A sarcastic and level-headed student who narrates the story while serving as Haruhi Suzumiya's reluctant stabilizing partner.
	Nagato	A quiet and unreadable SOS Brigade member with extraordinary intelligence and mysterious otherworldly origins.
	Koizumi	A smiling transfer student and esper who supports the Brigade while carefully concealing important secrets.
	Asahina	A timid and gentle upperclassman drawn into the SOS Brigade as its cute and enigmatic mascot, often swept into its chaos.
K-On!	Yui	A cheerful and scatterbrained lead guitarist whose energy and love of sweets keep the light music club lively.
	Ritsu	A loud, playful drummer whose pranks and casual leadership help keep the group upbeat and close.
	Mio	A shy but highly capable bassist with a gentle nature and strong musical sensitivity.
	Mugi	Tsumugi Kotobuki, a kind and wealthy keyboardist who enjoys treating her friends and making club life feel special.
	Azusa	A diligent and skilled junior guitarist who becomes central to the club's sound and practice habits.
FMA	Edward	A talented and stubborn young alchemist traveling to restore the bodies he and his brother lost.
	Alphonse	A kind boy whose soul is bound to a suit of armor, journeying with his brother to recover what was taken.
	Winry	A gifted automail mechanic and childhood friend of the Elrics, known for technical skill and steady compassion.
	Roy	A charismatic State Alchemist and flame user who seeks to reform the military from within.
	Ling	A determined prince from Xing who pursues immortality while carrying the burden of national duty.
JOJO	Jotaro	A stoic high schooler and leader of the Stardust Crusaders, known for Star Platinum and unshakable resolve.
	Polnareff	A bold French swordsman whose flamboyant personality and Silver Chariot make him a vivid ally.
	Joseph	A quick-thinking Joestar whose tricks, exaggeration, and improvisation often turn battles in his favor.
	DIO	A charismatic and ruthless vampire whose ambition and overwhelming presence define him as a legendary enemy.
	Kakyoin	A calm and analytical Crusader who fights with Hierophant Green and its precise emerald attacks.
	Avdol	A wise and loyal Egyptian Stand user whose Magician's Red controls powerful flames.
	Iggy	A stubborn Boston Terrier Stand user who dislikes cooperation but proves unexpectedly important.
DN	Light	A brilliant student who gains the Death Note and seeks to reshape the world through lethal justice.
	L	An eccentric and secretive detective whose unusual methods make him Light's most dangerous opponent.
	Near	A calm analytical successor to L who pursues the truth through detached logic and careful planning.
	Misa	A devoted idol and second Kira whose impulsive loyalty complicates the battle around the Death Note.
	Mello	A volatile and fiercely competitive successor to L who uses risk and criminal alliances to chase victory.
S×F	Loid	An elite spy who creates a fake family for a mission while learning the difficulties of improvised parenthood.
	Yor	A gentle civil servant secretly working as an assassin, trying to balance her hidden life with domestic normalcy.
	Anya	A cheerful telepathic child who knows everyone's secrets and helps hold the family together through innocence and insight.
AGOT	Tyrion	The sharp-witted youngest Lannister, navigating politics with intelligence, courage, and dark humor despite lifelong scorn.
	Daenerys	An exiled Targaryen princess who grows from a hesitant political pawn into a determined claimant to power.
	Cersei	An ambitious queen whose beauty masks ruthless devotion to family, status, and control.
	Jaime	The infamous Kingslayer, charming and deadly yet deeply conflicted by duty, loyalty, and reputation.
	Robb	The dutiful heir of Winterfell, forced into leadership and war before he is ready.
	Eddard	The honorable Lord of Winterfell, stern in duty and deeply committed to justice and family.
	Arya	An independent Stark girl who rejects courtly expectations in favor of freedom, survival, and the blade.
	Catelyn	The practical Lady of Winterfell, driven by fierce maternal loyalty and political urgency.
	Sansa	The elder Stark daughter, raised with courtly dreams that are tested by harsh political reality.
	Jon	Eddard's brooding illegitimate son, shaped by questions of identity, duty, and belonging.
	Bran	A curious young Stark whose fall places him on an unexpected and consequential path.
ATLA	Aang	The last Airbender and reluctant Avatar, playful by nature but burdened with restoring balance to a world at war.
	Katara	A compassionate and determined waterbender who grounds the group and strongly resists injustice.
	Sokka	A witty and inventive warrior whose humor, planning, and boomerang often prove unexpectedly useful.
	Zuko	An exiled Fire Nation prince whose search for honor gradually becomes a search for identity and moral direction.

Table 8: Brief background information of characters in our experiments for the Fandom Benchmark.

PoPiPa	Kasumi	An upbeat vocalist and guitarist whose starry enthusiasm pulls others forward and begins the band's journey.
	Tae	A free-spirited lead guitarist with strong technique and quirky instincts, often moving at her own pace while enriching the band's sound.
	Rimi	A shy and gentle bassist who gains confidence through performance and supports the group with warmth.
	Saaya	A caring and dependable drummer whose family-minded steadiness anchors the band in practice and daily life.
	Arisa	A sharp-tongued but reliable keyboardist whose practicality and quick judgment keep the band organized and grounded.
AG	Ran	A blunt and prideful vocalist-guitarist who values authenticity and carries the band's direct rock spirit.
	Moca	A relaxed lead guitarist with a mischievous edge, hiding sharp observation and confidence beneath casual teasing.
	Himari	A bright bassist and nominal leader who encourages the group while trying to keep everyone together.
	Tomoe	A steady big-sister drummer who supports others with calm strength when the band needs stability.
	Tsugumi	A kind keyboardist with practical warmth, often mediating tension and keeping the group's everyday rhythm steady.
PasuPare	Aya	An earnest vocalist who pursues her idol dream through persistence, effort, and hard-won confidence.
	Hina	A cheerful genius guitarist who follows whatever seems fun, acting on bright ideas with speed and momentum.
	Chisato	A cool and realistic bassist whose professionalism often reins in chaos and protects the group's direction.
	Maya	A technically skilled drummer with deep audio-gear knowledge, becoming animated around music equipment and stage work.
	Eve	A sincere keytarist devoted to bushido, whose intensity and kindness can inspire others while causing surprises.
Roselia	Yukina	A driven vocalist pursuing an ideal sound with uncompromising focus and strict standards.
	Sayo	A serious and disciplined guitarist who trusts hard work over talent and shows care through responsibility.
	Lisa	A warm bassist who serves as emotional support, balancing ambition with empathy and reassurance.
	Ako	An energetic drummer with dramatic chuuni flair, bringing confidence while still seeking recognition and growth.
	Rinko	A shy and soft-spoken keyboardist with exceptional skill, gradually gaining courage through supportive bonds.
HHW	Kokoro	A fearless optimist who treats making people smile as a mission and turns wild ideas into sincere action.
	Kaoru	A theatrical guitarist who performs a prince-like role, using charm and melodrama to brighten the mood.
	Hagumi	A sunny and athletic bassist who moves with straightforward excitement and honest smiles.
	Kanon	A timid but kind drummer who slowly finds courage through friends who support her.
	Misaki	A pragmatic and overworked coordinator who keeps the group functional as the realist within cheerful chaos.
Monica	Mashiro	A sensitive vocalist and lyricist who struggles with insecurity and learns to express herself through song.
	Touko	A flashy and extroverted lead guitarist who loves attention and momentum, sometimes causing trouble through impulse.
	Nanami	A versatile bassist preoccupied with being normal, hiding inner conflict behind humor and adaptability.
	Tsukushi	A hardworking drummer and leader who tries to be dependable, persisting through clumsiness with care.
	Rui	A cool violinist and composer who values results while gradually confronting emotion and trust in music.
RAS	CHU^2	A demanding genius producer and DJ who builds the band through strict control and professional ambition.
	LAYER	A charismatic bassist and vocalist whose powerful presence and steady musicianship anchor the band's sound.
	LOCK	An earnest young guitarist who grows through pressure, admiration, and the need to prove herself.
	MASKING	A fearless drummer who thrives on volume and adrenaline, powering performances with physical intensity.
	PAREO	A devoted keyboardist with a shy core and idol-like polish, supporting the band's vision through loyalty and effort.
MyGO	Tomori	A withdrawn and sensitive vocalist-lyricist who uses words and songs to seek connection.
	Anon	A social rhythm guitarist who wants visibility and belonging, gradually learning sincerity as her confident front is tested.
	Raana	A freewheeling lead guitarist with mysterious calm, following sound and curiosity over social convention.
	Soyo	A composed bassist who tries to preserve harmony while carrying the pressure of unresolved history.
	Taki	A blunt drummer and composer whose strict standards conceal protectiveness and concern.

Table 9: Brief background information of characters in our experiments for the Bandori Benchmark.