Codifying Character Logic in Role-Playing

Letian Peng, Jingbo Shang[*]
Department of Computer Science
University of California, San Diego
{lepeng, jshang}@ucsd.edu

Abstract

This paper introduces **Codified Profiles** for role-playing, a novel approach that represents character logic as structured, executable functions for behavioral decision-making. Converted by large language model (LLM) from textual profiles, each codified profile defines a set of functions `parse_by_scene(scene)` that output multiple logic-grounded assertions according to `scene`, using both explicit control structures (e.g., `if-then-else`) and flexible `check_condition(scene, question)` functions where each `question` is a semantically meaningful prompt about the `scene` (e.g., *"Is the character in danger?"*) discriminated by the role-playing LLM as *true*, *false*, or *unknown*. This explicit representation offers three key advantages over traditional prompt-based textual profiles, which append character descriptions directly into text prompts: (1) *Persistence*, by enforcing complete and consistent execution of character logic, rather than relying on the model's implicit reasoning; (2) *Updatability*, through systematic inspection and revision of behavioral logic, which is difficult to track or debug in prompt-only approaches; (3) *Controllable Randomness*, by supporting stochastic behavior directly within the logic, enabling fine-grained variability that prompting alone struggles to achieve. To validate these advantages, we introduce a new benchmark constructed from 83 characters and 5,141 scenes curated from Fandom, using natural language inference (NLI)-based scoring to compare character responses against ground-truths. Our experiments demonstrate the significant benefits of codified profiles in improving persistence, updatability, and behavioral diversity. Notably, by offloading a significant portion of reasoning to preprocessing, codified profiles enable even 1B-parameter models to perform high-quality role-playing, providing an efficient, lightweight foundation for local deployment of role-play agents. [2]

1 Introduction

Role-playing [Chen et al., 2024a,b] has emerged as an iconic capability of modern large language models (LLMs) [OpenAI, 2023, Touvron et al., 2023], enabling them to simulate characters with distinct personas. This ability has spurred a surge of interest in both research [Shao et al., 2023, Wang et al., 2023a,b, Sadeq et al., 2024] and applications[3], ranging from emotional companions and storytelling assistants [Brahman et al., 2021, Qin et al., 2024] to non-playable game characters [Yu et al., 2025] and metaverse agents [Zhou, 2023, Lifelo et al., 2024].

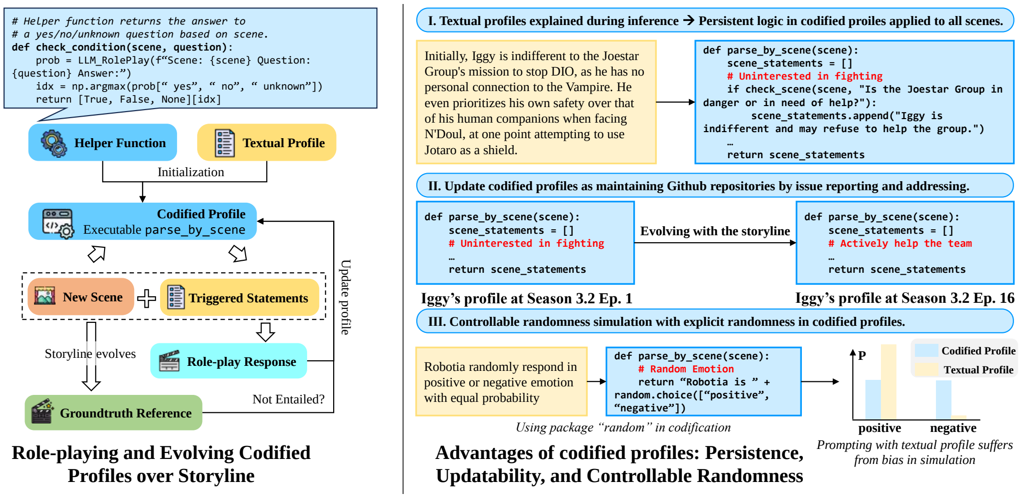
Despite this progress, current LLM-based role-playing systems interpret character logic by appending character descriptions directly into prompts. This approach *lacks persistence and controllability*: character behavior is governed by the LLM's implicit reasoning, which is often brittle and inconsistent.

[*] Corresponding author.

[2] Codes and datasets are available at https://github.com/KomeijiForce/Codified_Profile_Koishiday_2025

[3] https://character.ai/

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



```
# Helper function returns the answer to
# a yes/no/unknown question based on scene.
def check_condition(scene, question):
    prob = LLM_RolePlay(f“Scene: {scene} Question:
{question} Answer:”)
    idx = np.argmax(prob[“ yes”, “ no”, “ unknown”])
    return [True, False, None][idx]
```

Role-playing and Evolving Codified Profiles over Storyline

I. Textual profiles explained during inference → Persistent logic in codified proiles applied to all scenes.

Initially, Iggy is indifferent to the Joestar Group's mission to stop DIO, as he has no personal connection to the Vampire. He even prioritizes his own safety over that of his human companions when facing N'Doul, at one point attempting to use Jotaro as a shield.

```
def parse_by_scene(scene):
    scene_statements = []
    # Uninterested in fighting
    if check_scene(scene, "Is the Joestar Group in
danger or in need of help?"):
        scene_statements.append("Iggy is
indifferent and may refuse to help the group.")
    …
    return scene_statements
```

II. Update codified profiles as maintaining Github repositories by issue reporting and addressing.

```
def parse_by_scene(scene):
    scene_statements = []
    # Uninterested in fighting
    …
    return scene_statements
```

```
def parse_by_scene(scene):
    scene_statements = []
    # Actively help the team
    …
    return scene_statements
```

Iggy’s profile at Season 3.2 Ep. 1

Iggy’s profile at Season 3.2 Ep. 16

III. Controllable randomness simulation with explicit randomness in codified profiles.

Robotia randomly respond in positive or negative emotion with equal probability

```
def parse_by_scene(scene):
    # Random Emotion
    return “Robotia is ” +
random.choice([“positive”,
“negative”])
```

Using package “random” in codification

Prompting with textual profile suffers from bias in simulation

Advantages of codified profiles: Persistence, Updatability, and Controllable Randomness

Figure 1: A presentation of the workflow and advantages of codified profiles.

Even well-crafted profiles frequently yield unpredictable results, particularly in complex or ambiguous scenes, and offer little transparency, making them *difficult to debug or update*. In addition, interpreting profile logic at runtime imposes significant computational overhead and introduces latency, limiting scalability and responsiveness.

To address these challenges, we propose **Codified Profiles**, which compile character logic from natural language descriptions into executable functions. Unlike prompt-based role-playing, which appends character descriptions to prompts and relies on runtime interpretation, codified profiles are constructed in advance. They define character behavior using explicit control logic, producing consistent and interpretable outputs while reducing reliance on the LLM’s implicit reasoning. This design enhances both behavioral persistence and computational efficiency by eliminating the need for repeated logic interpretation during interaction.

As shown in Figure 1, we convert textual profiles into `parse_by_scene(scene)` functions via LLM prompting, which use control structures to generate logic-based `triggered_statements`. Each function includes multiple callable functions `check_condition(scene, question)` that evaluate scene-specific conditions using the role-playing LLM’s logits over verbalizers (e.g., “yes”, “no”, “unknown”), enabling flexible semantic reasoning within a deterministic framework. At inference, the function outputs guide the LLM’s response based on explicit character logic.

Codified profiles offer three key advantages over prompt-based methods. First, they ensure *persistence* by enforcing consistent execution of character logic, avoiding reliance on the LLM’s implicit reasoning. Second, they enable *updatability*, as structured, executable logic can be systematically inspected, debugged, and revised. Third, they support *controllable randomness*, allowing explicit encoding of stochastic behavior for fine-grained variability [Guo et al., 2024, Gu et al., 2025]. These properties make codified profiles ideal for faithful, efficient, and adaptive role-playing, especially with smaller models.

To evaluate the effectiveness of codified profiles, we construct a new Fandom Benchmark comprising 83 characters and 5,141 scenes curated from Fandom, spanning mangas, novels, and television series. Each scene is paired with ground-truth character actions extracted from original narrative artifacts. To assess alignment between model outputs and expected behavior, we employ a validated LLM-based natural language inference (NLI) scoring framework [Bowman et al., 2015], which measures logical entailment between generated responses and reference actions.

Our experiments start with a comparison between our codified profiles and traditional textual profiles, and then dive deep into the effect of evolving profiles that update sequentially along the episode timeline, the influence of profile-driven randomness, and the model’s Best@K performance under stochastic response settings. Across all settings, codified profiles significantly improve behavioral consistency, adaptability, and diversity, especially when used with smaller LLMs.

We further investigate codification strategies, focusing on the granularity of profile segmentation. Our findings indicate that paragraph-level segments strike the best balance between fidelity and computational efficiency: larger segments often obscure fine-grained logic, while smaller ones lead to excessive fragmentation and redundant condition checking. We also present case studies that demonstrate the concrete advantages of codified profiles over prompt-based interpretations. Our key contributions are as follows:

- We propose codified profiles as executable and interpretable representations of character logic that provide persistence, updatability, and controllable randomness beyond the limitations of prompt-based role-playing.
- We introduce a new benchmark, constructed from Fandom-sourced scenes and characters, for evaluating role-playing consistency. We plan to open-source this dataset to facilitate future research.
- We present an in-depth analysis of codified profiles, including distilling condition checkers, profile segmentation trade-offs, and demonstrate their impact through detailed case studies.

2 Background

The rise of LLMs has heightened users' expectations not only for accurate answers but also for emotionally supportive interactions [Chen et al., 2024a,b]. Role-playing has emerged as a trending task to meet this demand, enabling engaging interactions such as simulating fictional characters [You et al., 2022, Thoppilan et al., 2022]. It allows LLMs to adopt varied profiles, supporting storytelling [Brahman et al., 2021, Qin et al., 2024], games [Yu et al., 2025], and metaverse applications [Zhou, 2023, Lifelo et al., 2024]. We present a comprehensive overview of role-playing developments, including training, inference, grounding, evaluation, and the evolving role of codified profiles in both traditional games and modern LLMs.

Training and Inference The most straightforward way to build a character into an LLM is to train on how the character reacts to different conditions. Such training data can be obtained from existing plots [Li et al., 2023] or profile-based data synthesis [Shao et al., 2023]. However, the scarcity of plots (especially for original characters) and insufficient grounding for synthesis [Peng and Shang, 2024] cause hallucination challenges [Sadeq et al., 2024] caused by fine-tuning. During the inference, LLMs are generally prompted with statements from profiles to mitigate hallucinations, which requires a strong grounding system.

Grounding and Updating The grounding system in role-playing models what and how to append profile information to prompts. The "how" part is modeled similarly to traditional retrieval-augmented generation (RAG) [Lewis et al., 2020] using similarity metrics to retrieve the most relevant statements [Karpukhin et al., 2020, Liu et al., 2023, Salemi et al., 2024]. The "what" part discusses how to preprocess the statements for retrieval, including formatting and segmentation [Yuan et al., 2024]. Advanced role-playing systems also expect statements to be dynamically updated during the interaction [Zhou et al., 2024] for both working memory and long-term profile. Such updates generally "backpropagate" self-reflection [Shinn et al., 2023] to the profile, which refines the original statements in a textual gradient style [Yuksekgonul et al., 2025].

Evaluation There are evaluations for both understanding and generation in role-playing. For understanding, factuality checking and multiple-choice question answering benchmarks [Salemi et al., 2024, Yuan et al., 2024] are developed to evaluate the understanding of characters. For generation, the reference comes from ground-truth scenes [Wang et al., 2023b] and human judgment [Chen et al., 2023, 2025]. The profile is an essential reference for human evaluators, especially when evaluators are unfamiliar with the character. The evaluation can also be decomposed into simple tasks like natural language inference [Bowman et al., 2015] that can be trustfully assigned to neural models for automated evaluation [Peng and Shang, 2024]. Some works also apply LLMs to evaluate more advanced attributes such as value and personality [Shao et al., 2023, Wang et al., 2024].

Code in Role-playing Building characters with code is strongly connected to the development of video games [Stanley et al., 2006, Gallotta et al., 2024]. The most common practice is to hardcode the actions of non-player characters (NPCs), making them actors in prewritten scripts. They are only allowed to intellectually interact with players in some subscenes (e.g., battle, chess), which hardly affect the storyline progression. In role-playing, codes and LLMs represent two extremes in interaction freedom, which are synergized in our codified profile framework to reach a balanced point by generating and executing prewritten symbolic logic by a flexible neural system.

3 Codified Profile

3.1 Role-playing Preliminary

A pure role-playing system models character logic as "Input a scene s, output a response r to the scene as character X", where the response can be utterances, actions, or both. Here we use $\text{LLM}(\cdot)$ to represent the role-playing systems as our discussion scope will be LLM-driven role-playing. The LLM is prompted to role-play as characters with a system prompt $I_{\text{role-play}}$, which does not contain information of any character to be universally applied for role-playing: $r = \text{LLM}(s \mid I_{\text{role-play}})$.

Textual profile $P = [p_1, p_2, \cdots, p_N]$ (p_i represents segments of P like paragraphs) further grounds the role-playing with in-context information about character logic. With P appended to the input, the role-playing LLM can output its response r directly or after reasoning with chain-of-thoughts (CoT) [Wei et al., 2022] cot prompted from CoT reasoning instruction I_{CoT}.

$$r = \text{LLM}(s \mid P, cot, I_{\text{role-play}}, I_{\text{CoT}}) \text{ where } cot = \text{LLM}(s \mid P, I_{\text{role-play}}, I_{\text{CoT}}) \tag{1}$$

3.2 Codifying Profiles

Our motivation to codify profiles emerges from the instability for LLMs to interpret textual profiles during inference (the CoT stage $cot = \text{LLM}(s \mid I_{\text{role-play}}, I_{\text{CoT}}, P)$), which completely relies on LLM's reasoning accuracy without missing or misinterpretation. This reliance creates a brittle dependency on the model's reasoning over long, natural language inputs, often leading to errors in complex scenes. Codification mitigates this fragility by offloading as much reasoning as possible into deterministic control logic, allowing the LLM to focus on localized decisions such as condition evaluation and action selection. All prompts and templates for implementation are in Appendix B.

In practice, we utilize LLM's coding ability to codify each profile segment p_i into an executable function $f_i : s \rightarrow p_i^{\text{trig}}$ (Code implementation: `parse_by_scene(scene)` → `triggered_statements`), which returns the possible reactions p_i^{trig} in the scene s based on the logic written in p_i. LLM then combines all p_1^{trig} to summarize a final response r.

$$r = \text{LLM}(s \mid [f_1(s), f_2(s), \cdots, f_N(s)], I_{\text{role-play}}) \tag{2}$$

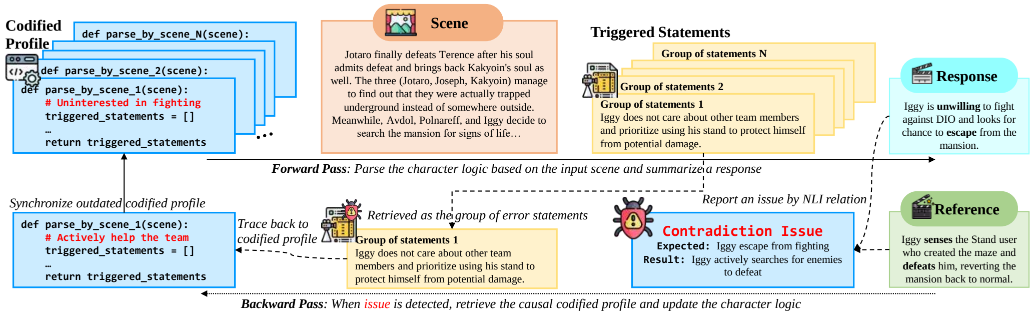
To enhance flexibility while preserving structure, each function f_i can incorporate semantic checks using a callable helper function `check_condition(scene, question)`. This callable allows f_i to evaluate context-sensitive conditions that are difficult to express through static rules alone. For example, a rule like "Whenever being insulted, X will become outrageous" may be codified as:

```
if check_condition(scene, "Is X being insulted?"):
    triggered_statements.append("X become outrageous")
```

Internally, `check_condition` queries the role-playing LLM with a natural language question and interprets the logits over verbalizers (e.g., *"yes", "no", "unknown"*) to yield boolean or uncertain judgments. This enables codified profiles to combine deterministic control flow with semantic nuance, supporting persistent and context-aware behavior across scenes. As shown in Figure 15 (Appendix F), this approach covers personality, relationships, and superpowers. By shifting the task from profile-based reasoning to localized condition classification, codification allows smaller LLMs to perform competitively, making it well-suited for low-resource deployments.

3.3 Evolving by Storyline

An essential advantage of codified profiles is long-term maintenance, considering the Github repository updates according to discovered issues. Codified profiles can also be updated along with the storyline, and each profile version reflects the character logic in different episodes. We plot the workflow to evolve codified profiles in Figure 2. Issues in role-playing emerge from predicted response r mismatching with ground-truth response r_{ref} based on scene s. We select natural language inference (NLI) as the initial signal for profile updating, which discriminates r as {*"contradicted"*, *"neutral"*, *"entailed"*} based on r_{ref}.


Codified Profile
Scene
Triggered Statements
Response
Reference
Forward Pass: Parse the character logic based on the input scene and summarize a response
Synchronize outdated codified profile
Trace back to codified profile
Retrieved as the group of error statements
Report an issue by NLI relation
Backward Pass: When issue is detected, retrieve the causal codified profile and update the character logic
Group of statements N
Group of statements 2
Group of statements 1

```
def parse_by_scene_N(scene):
def parse_by_scene_2(scene):
def parse_by_scene_1(scene):
    # Uninterested in fighting
    triggered_statements = []
    …
    return triggered_statements
```

Jotaro finally defeats Terence after his soul admits defeat and brings back Kakyoin's soul as well. The three (Jotaro, Joseph, Kakyoin) manage to find out that they were actually trapped underground instead of somewhere outside. Meanwhile, Avdol, Polnareff, and Iggy decide to search the mansion for signs of life…

Group of statements 1
Iggy does not care about other team members and prioritize using his stand to protect himself from potential damage.

Iggy is **unwilling** to fight against DIO and looks for chance to **escape** from the mansion.

```
def parse_by_scene_1(scene):
    # Actively help the team
    triggered_statements = []
    …
    return triggered_statements
```

Group of statements 1
Iggy does not care about other team members and prioritize using his stand to protect himself from potential damage.

Contradiction Issue
Expected: Iggy escape from fighting
Result: Iggy actively searches for enemies to defeat

Iggy **senses** the Stand user who created the maze and **defeats** him, reverting the mansion back to normal.

Figure 2: The evolving mechanism of codified profile to synchronize character with the storyline.

When $\text{NLI}(r_{\text{ref}}, r) \in \{$*"contradicted"*, *"neutral"*$\}$, the issue will be reported to the response generation stage, citing *"contradicted statement"* (contradicted) or *"relevant but not detailed statement"* (neutral). Then, LLM is prompted to detect the group of `triggered_statements` that fits such accusation. Supposing p_j^{trig} as one of the proposed `triggered_statements` groups, f_j will be revised as a problematic character logic. Finally, f_j will be updated based on the input scene s together with $p_j^{\text{trig}}, r_{\text{ref}}, r$ to synchronize the character logic with the storyline.

Textual profiles are not inherently unevolvable. If formatted into logic like `if-then-else`, they can be incrementally revised. However, most textual profiles are not logically formalized. Moreover, updating texts mimics code debugging, yet lacks the precision or modularity of code debugging.

3.4 Controllable Randomness Simulation

LLMs struggle to simulate precise randomness, as generation is governed by temperature rather than probability in instructions. For example, when asked to respond with equal probability between positive and negative sentiment, they often show biased, unbalanced behavior due to training data and decoding preferences [Guo et al., 2024, Gu et al., 2025].

Codified profiles overcome this limitation by externalizing randomness into explicit Python control logic, using constructs like `random.choice([...])` or `random.random() < p`. Instead of relying on the LLM's generation-time sampling, this approach enforces precise stochastic behavior directly within the code. For example, to simulate random sentiment:

```
sentiment_statement = random.choice(["X is positive.", "X is negative."])
```

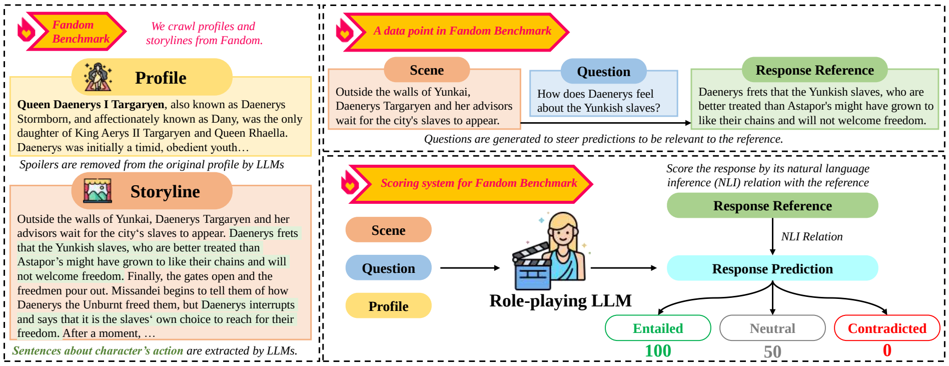
By embedding such logic in codified profiles, we achieve fine-grained control over behavioral randomness, ensuring both reproducibility and tunability. This approach is particularly advantageous in applications where controlled variability is essential, such as user interaction.

4 Fandom Benchmark

Although various high-quality benchmarks [Wang et al., 2024, 2025] exist for evaluating role-playing in LLMs, they tend to emphasize dialogue interactions over the complicated execution of situational actions in grounded contexts. Moreover, many benchmarks rely on data synthesized by LLMs or sparsely annotated scripts [Wang et al., 2023b, Chen et al., 2024a], which restricts them in evaluating an in-depth understanding of massive character logic.

We propose using Fandom[4] as a rich source of character profiles and structured story summaries to build a behavior-centric role-playing benchmark. For each narrative segment, we prompt `gpt-4.1` to extract character actions (e.g., Green-highlighted parts in the storyline of Figure 3) and treat preceding text as the scene. Each scene-action pair is further augmented with a carefully crafted guiding question (by `gpt-4.1`), which constrains responses to be reference-relevant without revealing answers. During evaluation, the role-playing LLM receives the character profile (spoiler-free, cleaned by `gpt-4.1`), scene, and question, and predicts the character's next action. Predictions are scored using an LLM-based NLI system: entailed predictions score 100, neutral 50, and contradictions 0.

[4] https://www.fandom.com/



Queen Daenerys I Targaryen, also known as Daenerys Stormborn, and affectionately known as Dany, was the only daughter of King Aerys II Targaryen and Queen Rhaella. Daenerys was initially a timid, obedient youth…

Spoilers are removed from the original profile by LLMs

Outside the walls of Yunkai, Daenerys Targaryen and her advisors wait for the city's slaves to appear. Daenerys frets that the Yunkish slaves, who are better treated than Astapor's might have grown to like their chains and will not welcome freedom. Finally, the gates open and the freedmen pour out. Missandei begins to tell them of how Daenerys the Unburnt freed them, but Daenerys interrupts and says that it is the slaves' own choice to reach for their freedom. After a moment, …

Sentences about character's action are extracted by LLMs.

Outside the walls of Yunkai, Daenerys Targaryen and her advisors wait for the city's slaves to appear.

How does Daenerys feel about the Yunkish slaves?

Daenerys frets that the Yunkish slaves, who are better treated than Astapor's might have grown to like their chains and will not welcome freedom.

Questions are generated to steer predictions to be relevant to the reference.

Scoring system for Fandom Benchmark
Score the response by its natural language inference (NLI) relation with the reference
Scene
Question
Profile
Role-playing LLM
Response Reference
NLI Relation
Response Prediction
Entailed
100
Neutral
50
Contradicted
0

Figure 3: Curation and evaluation scenarios of our Fandom Benchmark.

We select 6 reputational artifacts to cover various themes of stories: *"Suzumiya Haruhi (Haruhi)"*, *"K-On!"*, *"JOJO's Bizzare Adventure (JOJO)" (Season 3)*, *"Fullmetal Alchemist (FMA)" (2009)*, *"A Game of Thrones (AGOT)" (Season 1-3)*, and *"Avatar: The Last Airbender (ATLA)" (Book1)*. The benchmark covers diverse artifacts from sci-fi to adventure across manga, novels, and TV series, featuring 83 characters in 5,141 scenes. Appendix G provides detailed statistics, character information, storylines, and example profiles for clarity.

5 Experiment

5.1 Evaluation Scenarios

For evaluation, we use `gpt-4.1` as the codification LLM to convert natural language character profiles into codified profiles.[5] Then, we employ `llama-3.1-8b-instruct` as the role-playing LLM. This model is chosen to ensure experimental efficiency, including tractable forward pass counts and full access to output logits for analysis.[6] For each scene in our Fandom benchmark, the role-playing LLM is prompted with a given profile and is asked to produce the character's next action. This response is scored using LLM-based (`gpt-4.1`; We manually check its 5 NLI scoring cases for all 83 characters, and find 92.35% entailed, 90.91% neutral, and 88.89% contradicted judgments align with human evaluation, which validates the usage of LLM for automated NLI scoring) NLI against the ground-truth, as mentioned in the benchmark section. We further extend the basic scenario to **evolving profile** and **stochastic response**.

Setup	Update Profile	Import Random	Sampling	Metric
Basic	✗	✗	✗	NLI
Evolving	✓	✗	✗	NLI
Stochastic	✗	✓	✓	Best@K (NLI)

Table 1: Detailed setups in experiments.

In **Evolving Profile**, scenes are evaluated sequentially; after testing on the i-th scene, if the predicted action is not entailed (based on NLI), the profile is updated to reflect revised behavioral logic before testing scene $(i+1)$, as detailed in § 3.3. In practice, we update one function for each reported issue for efficiency. In **Stochastic Response**, the role-playing LLM generates K responses per scene using a fixed codified profile with probabilistic constructs (e.g., `random.choice`) and temperature < 0.7. The codification LLM is explicitly prompted to introduce randomness into control flow. We evaluate using Best@K, which assigns the final score for each scene based on the highest NLI score among the K sampled responses, capturing the model's potential best alignment. Difference between setups can be referred to Table 1. Before the experiments, we check the codification ability of `gpt-4.1`, which shows competent performance as shown in Appendix E. All prompts are included in Appendix B.

5.2 Baselines

- **Vanilla** The role-playing LLM is prompted with only the scene. This setting serves as a lower bound, representing performance with parameterized knowledge when profile data is unavailable.
- **Textual Profile** The full natural language character profile is provided alongside the scene. This is the standard grounding method in most existing role-playing systems.

[5] In Appendix 5.6, we discuss the influence of using different codification LLMs.

[6] GPT-4 series are also found memorizing artifacts like AGOT [Chang et al., 2023], injecting evaluation bias.

	Artifact	Haruhi	K-On!	JOJO	FMA	AGOT	ATLA	Average
Main	#Character	5	5	7	5	11	4	5.3
	Vanilla (No Profile)	63.53	63.60	59.70	67.10	64.93	66.40	64.21
	Textual Profile	70.53	68.47	60.64	68.02	61.52	66.70	65.98
	Codified RAG	69.03	66.81	59.52	69.14	65.39	66.33	66.04
	Codified Profile	72.14	69.23	63.92	69.60	67.51	67.90	**68.38**
	+ Textual Profile	72.81	71.17	63.42	71.36	65.60	67.37	**68.62**
Minor	#Character		4	9	7	19	7	9.2
	Vanilla (No Profile)	N/A	65.12	62.53	61.17	67.65	66.89	64.67
	Textual Profile		65.88	64.70	65.83	66.21	65.88	65.70
	Codified RAG		63.28	68.13	66.97	68.62	66.65	66.73
	Codified Profile		68.59	70.06	68.21	70.51	71.95	**69.87**
	+ Textual Profile		67.97	68.94	69.22	71.83	71.86	**69.96**

Table 2: Role-playing performance based on different profiles. N/A: No minor character in Haruhi.

- **Codified RAG** A simple codification strategy that implements retrieval-augmented generation (RAG) over the profile. Each segment of the profile is wrapped in a `check_condition` function to determine its relevance to the current scene (`if check_condition(scene, question): triggered_statements.append(segment)`). This method evaluates whether logic-based codification offers deeper character understanding than shallow relevance matching.
- **Reasoning Mechanism** We incorporate chain-of-thoughts prompting before the action prediction step. To fairly evaluate reasoning efficiency, we apply CoT both on top of the Textual Profile and the Codified Profile, and vary the average reasoning chain length from 0 upward. We report performance and forward pass count to assess the trade-off between response quality and computational cost.

5.3 Role-playing Results

In Table 3 and Figure 4, we present the performance of various role-playing methods. Results show that all grounding-based approaches outperform the vanilla (No Profile) baseline, underscoring the importance of contextual character guidance. The result validates our core claim that enforcing consistent character logic yields better outcomes than relying on the LLM's implicit reasoning, as codified profiles outperform Textual Profiles across both main and minor characters. Codified profiles also surpass Codified RAG, demonstrating that their advantage lies not just in retrieving relevant content but in expressing structured, executable behavior logic for more persistent and faithful character simulation.

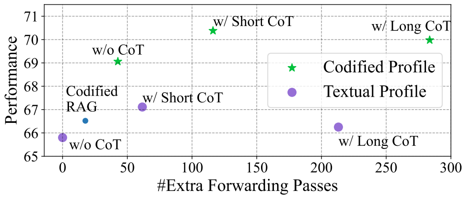


Figure 4: The role-playing performance with reasoning mechanism.

Adding textual profiles on top of codified profiles provides further but marginal improvements, especially when the performance difference between code and text is small. This suggests that textual prompts can sometimes complement information that may have been simplified or abstracted during codification, enhancing the expressiveness of the final behavior. In Appendix C, we continue the discussion with human/LLM preference and evaluation on original (out-of-script) scenes.

Reasoning Figure 4 compares codified reasoning with chain-of-thoughts reasoning, averaged over all characters. The results show that the codified reasoning flow achieves strong performance with fewer reasoning steps than chain-of-thoughts, demonstrating it as a more efficient grounding mechanism compared to approaches that rely heavily on runtime reasoning. Notably, performance improves further when codified profiles are combined with targeted reasoning, indicating that our method can benefit from additional inference when applied judiciously. In contrast, the long chain-of-thought (CoT) baseline underperforms, and manual inspection reveals that it often suffers from overthinking, introducing unnecessary complications that derail character-consistent behavior.

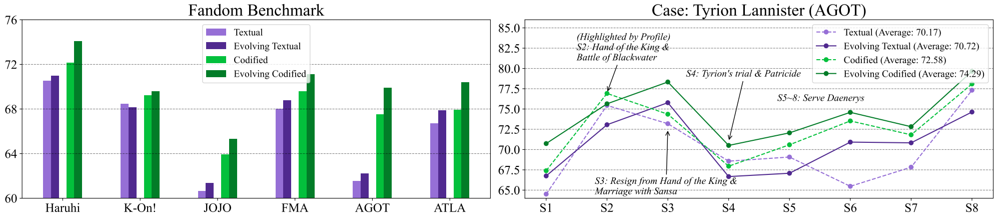


Figure 5: Role-playing performance with evolving profiles. **Left:** Performance gain from evolving. **Right:** Case of *"Tyrion Lannister"* in AGOT to analyze the working mechanism of evolving.

Evolving In Figure 5, we examine the effect of evolving codified profiles by the storyline on all characters inside the Fandom benchmark. Evolving for textual profiles follows a similar pipeline as the codified profile by select-then-revise. The results show codified to be more efficient than textual profiles in evolving, as it directly models character logic. The evolving mechanism works better on adventure series rather than relaxing anime like K-On! as they involve more character growth.

The right subfigure presents a case study on *"Tyrion Lannister"*, evolving along with all 8 seasons, illustrating how evolving profiles maintain alignment with narrative shifts. Tyrion's profile is well-grounded in Season 2 during the Battle of Blackwater, his defining moment centered by the original profile. However, Seasons 3∼4 show a heightened need for evolution, surpassing even later seasons when he joins Daenerys, reflecting rapid character development post-battle. This highlights that the most critical synchronization occurs after major turning points. Codified profiles with evolution enable timely updates to capture these shifts, preserving logic-storyline consistency.

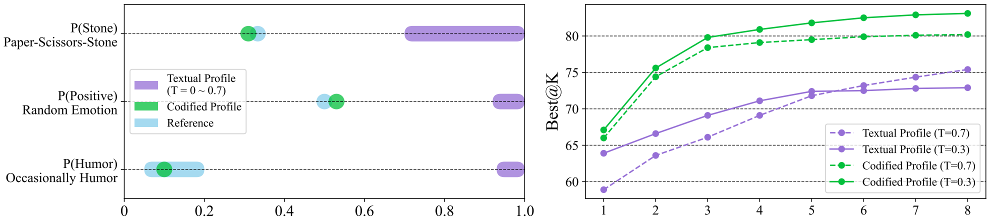


Figure 6: **Left:** Cases on controllable randomness simulation by textual and codified profiles. **Right:** Scenario coverage comparison, evaluated by Best@K performance with stochastic responses.

Figure 6 compares randomness control in textual vs. codified profiles across three simulation cases (by 100 runs): choosing paper-scissor-stone (with equal probability), responding with positive or negative emotions (with equal probability), and occasionally adding humor (considering $5\% \sim 20\%$ as acceptable probability for *"occasionally"*). Textual prompting (Temperature 0∼0.7) shows significant bias, overproducing certain responses, while codified profiles accurately follow specified probabilities via explicit control flow. Appendix F details the codified logic in Figure 17, which aligns with intended behavior.

The right subfigure shows the application of randomness simulation to complex characters in the Fandom benchmark. As expected, codified profiles implemented via explicit randomness in code control flow achieve stronger Best@K performance, meaning they explore more reasonable reactions. Textual profiles rely on raising temperature to explore variability, which degrades single-run precision and introduces inconsistency. By decoupling behavioral exploration from sampling noise, codified profiles deliver diversity and accuracy at low temperatures, making them diverse yet high-quality for role-playing. We further showcase how randomness enriches characters in Figure 18.

5.4 Smaller Role-playing LLMs

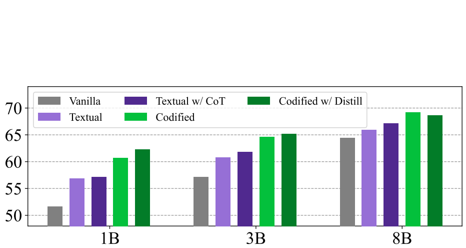


Figure 7: The role-playing performance with smaller role-playing LLMs.

We compare vanilla prompting, textual profiles with chain-of-thought, and codified profiles across LLaMA-3 models of 1B (3.2), 3B (3.2), and 8B (3.1) parameters in Figure 7, and observe that the advantage of codified profiles grows as model size decreases. Notably, smaller LLMs with codified profiles rival the larger LLMs with textual profiles for both 1B and 3B cases. Manual inspection shows smaller

	Artifact	Haruhi	K-On!	JOJO	FMA	AGOT	ATLA	Average
Main	`kimi-k2`	74.81	67.71	62.70	68.89	67.32	64.62	67.68
	`qwen3-instruct`	76.24	67.89	63.80	70.27	67.46	64.23	68.32
	`qwen3-coder`	74.25	69.82	63.62	70.62	67.88	69.94	69.36
	`gpt-4.1-mini`	73.33	68.58	62.71	68.07	67.03	68.54	68.04
	`gpt-4.1`	72.14	69.23	63.92	69.60	67.51	67.90	68.38
	`claude-3.7-sonet`	76.86	68.53	63.35	70.82	68.69	68.73	69.50
	`claude-4.0-sonet`	75.07	68.99	63.85	74.26	67.22	70.82	**70.04**
Minor	`kimi-k2`	N/A	67.70	65.79	65.53	71.70	64.09	66.96
	`qwen3-instruct`		69.48	66.08	65.64	71.71	65.70	67.72
	`qwen3-coder`		71.63	67.51	64.44	73.64	72.06	69.86
	`gpt-4.1-mini`		71.65	64.66	70.35	69.77	70.43	69.37
	`gpt-4.1`		68.59	70.06	68.21	70.51	71.95	69.87
	`claude-3.7-sonet`		75.05	67.05	67.04	73.79	69.25	70.44
	`claude-4.0-sonet`		71.79	67.66	70.94	74.63	73.72	**71.75**

Table 3: Role-playing performance based on codification LLMs with the 8B role-playing LLM.

models often fail to reconstruct character logic from free-form text but still perform well when logic is preprocessed by stronger LLMs (e.g. `gpt-4.1`) into explicit conditions. This supports our central claim in the introduction: codified profiles reduce reliance on the LLM's implicit and inconsistent reasoning by enforcing structured, interpretable behavior, which is especially critical for smaller models with limited reasoning ability.

Distilled Condition Checker As `check_condition(scene, question)` is a discriminative task, we explore the use of a distilled classifier to improve both performance and efficiency. Instead of querying the role-playing model for every condition, we distill from `gpt-4.1`'s condition-checking outputs using 415 scenes (5 per character, 8% of all scenes) and obtain 20,759 labeled discrimination cases. A 3-class `deberta-v3-base` model (0.1B) [He et al., 2021] is trained on 90% of the data for 5 epochs and achieves 70.53% consistency with `gpt-4.1` on the held-out 10%. As shown in Figure 7, this design enables a highly efficient 1B + Code + Distill system that approaches the performance of 8B + Text, demonstrating the benefit of codified reasoning to smaller LLMs. On a larger 8B role-playing LLM, the distilled classifier maintains a comparable performance with significantly better efficiency, as illustrated in the efficiency analysis in Appendix D. We have also checked the performance excluding the 8% data for distillation, which reaches a consistent conclusion, supporting the generalization of the distilled condition checker.

5.5 Segmentation Strategy

Figure 8 compares section, paragraph, and sentence-level codification. Finer segmentation improves role-playing, with a notable gain from section to paragraph. However, the marginal benefit from paragraph to sentence comes at a high computational cost, almost tripling forward passes. Paragraphs offer a natural grouping of related statements, balancing fidelity and efficiency. Thus, we adopt paragraph-level segmentation for the main experiments.

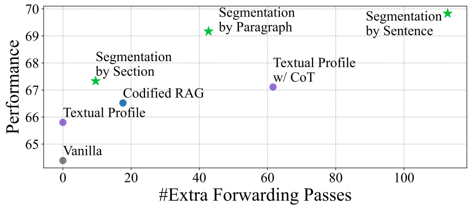
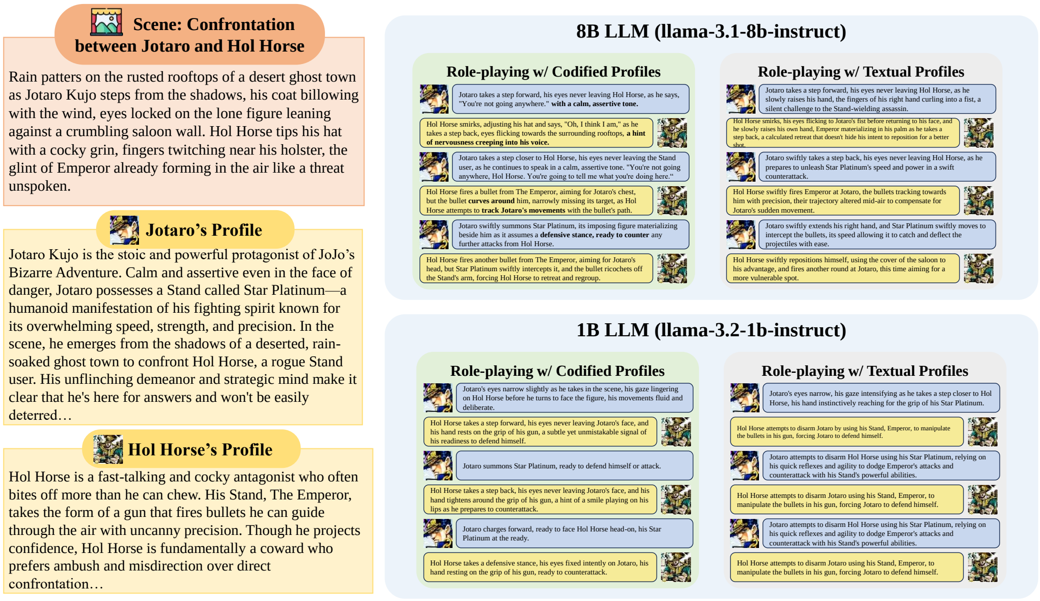


Figure 8: The role-playing performance with different segmentation strategies.

5.6 Codification LLM Comparison

Table 3 compares role-playing performance across different codification LLMs, showing that overall effectiveness generally correlates with the model's coding ability on the public leaderboard. However, performance varies by artifact, suggesting that codifying character profiles requires more than syntactic code generation, demanding a nuanced understanding of different personalities, contexts, and behavioral logic that vary among artifacts. Notably, `claude-3.7-sonet` achieves the highest average, indicating stronger alignment in both structural and semantic aspects of profile codification.



Scene: Confrontation between Jotaro and Hol Horse

Rain patters on the rusted rooftops of a desert ghost town as Jotaro Kujo steps from the shadows, his coat billowing with the wind, eyes locked on the lone figure leaning against a crumbling saloon wall. Hol Horse tips his hat with a cocky grin, fingers twitching near his holster, the glint of Emperor already forming in the air like a threat unspoken.

Jotaro's Profile

Jotaro Kujo is the stoic and powerful protagonist of JoJo's Bizarre Adventure. Calm and assertive even in the face of danger, Jotaro possesses a Stand called Star Platinum—a humanoid manifestation of his fighting spirit known for its overwhelming speed, strength, and precision. In the scene, he emerges from the shadows of a deserted, rain-soaked ghost town to confront Hol Horse, a rogue Stand user. His unflinching demeanor and strategic mind make it clear that he's here for answers and won't be easily deterred…

Hol Horse's Profile

Hol Horse is a fast-talking and cocky antagonist who often bites off more than he can chew. His Stand, The Emperor, takes the form of a gun that fires bullets through the air with uncanny precision. Though he projects confidence, Hol Horse is fundamentally a coward who prefers ambush and misdirection over direct confrontation…

8B LLM (llama-3.1-8b-instruct)

Role-playing w/ Codified Profiles

Jotaro takes a step forward, his eyes never leaving Hol Horse, as he says, "You're not going anywhere." **with a calm, assertive tone.**

Hol Horse smirks, adjusting his hat and says, "Oh, I think I am," as he takes a step back, eyes flicking towards the surrounding rooftops, **a hint of nervousness creeping into his voice.**

Jotaro takes a step closer to Hol Horse, his eyes never leaving the Stand user, as he continues to speak in a calm, assertive tone. "You're not going anywhere, Hol Horse. You're going to tell me what you're doing here.“

Hol Horse fires a bullet from The Emperor, aiming for Jotaro's chest, but the bullet **curves around** him, narrowly missing its target, as Hol Horse attempts to **track Jotaro's movements** with the bullet's path.

Jotaro swiftly summons Star Platinum, its imposing figure materializing beside him as it assumes a **defensive stance, ready to counter** any further attacks from Hol Horse.

Hol Horse fires another bullet from The Emperor, aiming for Jotaro's head, but Star Platinum swiftly intercepts it, and the bullet ricochets off the Stand's arm, forcing Hol Horse to retreat and regroup.

Role-playing w/ Textual Profiles

Jotaro takes a step forward, his eyes never leaving Hol Horse, as he slowly raises his hand, the fingers of his right hand curling into a fist, a silent challenge to the Stand-wielding assassin.

Hol Horse smirks, his eyes flicking to Jotaro's fist before returning to his face, and he slowly raises his own hand, Emperor materializing in his palm as he takes a step back, a calculated retreat that doesn't hide his intent to reposition for a better shot.

Jotaro swiftly takes a step back, his eyes never leaving Hol Horse, as he prepares to unleash Star Platinum's speed and power in a swift counterattack.

Hol Horse swiftly fires Emperor at Jotaro, the bullets tracking towards him with precision, their trajectory altered mid-air to compensate for Jotaro's sudden movement.

Jotaro swiftly extends his right hand, and Star Platinum swiftly moves to intercept the bullets, its speed allowing it to catch and deflect the projectiles with ease.

Hol Horse swiftly repositions himself, using the cover of the saloon to his advantage, and fires another round at Jotaro, this time aiming for a more vulnerable spot.

1B LLM (llama-3.2-1b-instruct)

Role-playing w/ Codified Profiles

Jotaro's eyes narrow slightly as he takes in the scene, his gaze lingering on Hol Horse before he turns to face the figure, his movements fluid and deliberate.

Hol Horse takes a step forward, his eyes never leaving Jotaro's face, and his hand rests on the grip of his gun, a subtle yet unmistakable signal of his readiness to defend himself.

Jotaro summons Star Platinum, ready to defend himself or attack.

Hol Horse takes a step back, his eyes never leaving Jotaro's face, and his hand tightens around the grip of his gun, a hint of a smile playing on his lips as he prepares to counterattack.

Jotaro charges forward, ready to face Hol Horse head-on, his Star Platinum at the ready.

Hol Horse takes a defensive stance, his eyes fixed intently on Jotaro, his hand resting on the grip of his gun, ready to counterattack.

Role-playing w/ Textual Profiles

Jotaro's eyes narrow, his gaze intensifying as he takes a step closer to Hol Horse, his hand instinctively reaching for the grip of his Star Platinum.

Hol Horse attempts to disarm Jotaro by using his Stand, Emperor, to manipulate the bullets in his gun, forcing Jotaro to defend himself.

Jotaro attempts to disarm Hol Horse using his Star Platinum, relying on his quick reflexes and agility to dodge Emperor's attacks and counterattack with his Stand's powerful abilities.

Hol Horse attempts to disarm Jotaro using his Stand, Emperor, to manipulate the bullets in his gun, forcing Jotaro to defend himself.

Jotaro attempts to disarm Hol Horse using his Star Platinum, relying on his quick reflexes and agility to dodge Emperor's attacks and counterattack with his Stand's powerful abilities.

Hol Horse attempts to disarm Jotaro using his Stand, Emperor, to manipulate the bullets in his gun, forcing Jotaro to defend himself.

Figure 9: Case study of multi-turn role-playing in a given scene: *"Jotaro confronting Hol Horse." (JOJO's Bizarre Adventure)*

6 Case Study

Figure 9 illustrates a case study that highlights the advantage of codified profiles in a direct role-play scenario. We construct a dynamic scene featuring a confrontation between *"Jotaro"* and *"Hol Horse"* from JOJO, and prompt LLMs to role-play as each character by *"Propose the next action that {character} takes in response."* Both textual and codified profiles are evaluated using 8B and 1B models to compare performance across model scales and profile formats.

Codified profiles enhance role-playing coherence and character fidelity by embedding consistent personality traits and logically grounded actions into explicit control flow. For instance, when Hol Horse fires a bullet and misses, the codified logic triggers Jotaro to summon Star Platinum in immediate defense, mirroring canonical behavior. In contrast, textual profiles often produce generic and less impactful responses, such as Jotaro merely raising a fist and stepping back without clear narrative consequence. By offloading complex reasoning into structured logic, codified profiles allow even smaller LLMs (e.g., 1B models) to engage in reactive exchanges, avoiding repetitive outputs like *"Hol Horse attempts to disarm Jotaro."* This explicit structure reduces reliance on the model's implicit reasoning and enables believable, narratively aligned interactions that would otherwise require far greater model capacity. We further include extended case studies for codified profile workflow in Appendix F, which can be referred to understand the mechanism and advantage of codified profiles.

7 Conclusion

We propose codified profiles, which offer a principled framework for character role-playing by compiling behavior into interpretable, executable functions. Our experiments demonstrate that this approach outperforms prompt-based methods in persistency, updatability, and controllable randomness, while enabling strong performance even in small models. These findings highlight codified profiles as a practical foundation for building efficient, locally deployable role-play agents across a wide range of applications. Future work, as discussed in Appendix A, includes extending codification with richer operators and helper tools, modeling environmental logic for world-level role-playing, and exploring the scope of codifiable textual content.

Acknowledgement

This work aims to contribute not only to the research community but also to a broader ACG community by introducing more powerful role-playing agents. It is also done in memory of the 17th *Koishi's Day* (May 14th), 2025, since the release of TH11, Touhou Chireiden ∼ Subterranean Animism[7] in 2008.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. "let your characters tell their story": A dataset for character-centric narrative understanding. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1734–1752. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-EMNLP.150. URL `https://doi.org/10.18653/v1/2021.findings-emnlp.150`.

Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to chatgpt/gpt-4, 2023. URL `https://arxiv.org/abs/2305.00118`.

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. From persona to personalization: A survey on role-playing language agents, 2024a. URL `https://arxiv.org/abs/2404.18231`.

Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. Large language models meet harry potter: A dataset for aligning dialogue agents with characters. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 8506–8520. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.570. URL `https://doi.org/10.18653/v1/2023.findings-emnlp.570`.

Nuo Chen, Yang Deng, and Jia Li. The oscars of AI theater: A survey on role-playing with language models. *CoRR*, abs/2407.11484, 2024b. doi: 10.48550/ARXIV.2407.11484. URL `https://doi.org/10.48550/arXiv.2407.11484`.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 755–773. Association for Computational Linguistics, 2025. URL `https://aclanthology.org/2025.coling-main.51/`.

Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Large language models and games: A survey and roadmap. *CoRR*, abs/2402.18659, 2024. doi: 10.48550/ARXIV.2402.18659. URL `https://doi.org/10.48550/arXiv.2402.18659`.

Jia Gu, Liang Pang, Huawei Shen, and Xueqi Cheng. Do llms play dice? exploring probability distribution sampling in large language models for behavioral simulation. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 5375–5390. Association for Computational Linguistics, 2025. URL `https://aclanthology.org/2025.coling-main.360/`.

[7] https://en.wikipedia.org/wiki/Subterranean_Animism

Yufei Guo, Muzhe Guo, Juntao Su, Zhou Yang, Mengqiu Zhu, Hongfei Li, Mengyang Qiu, and Shuo Shuo Liu. Bias in large language models: Origin, evaluation, and mitigation. *CoRR*, abs/2411.10915, 2024. doi: 10.48550/ARXIV.2411.10915. URL `https://doi.org/10.48550/arXiv.2411.10915`.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL `https://doi.org/10.18653/v1/2020.emnlp-main.550`.

Patrick S. H. 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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html`.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. Chatharuhi: Reviving anime character in reality via large language model. *CoRR*, abs/2308.09597, 2023. doi: 10.48550/ARXIV.2308.09597. URL `https://doi.org/10.48550/arXiv.2308.09597`.

Zita Lifelo, Jianguo Ding, Huansheng Ning, Qurat-Ul-Ain, and Sahraoui Dhelim. Artificial intelligence-enabled metaverse for sustainable smart cities: Technologies, applications, challenges, and future directions. *Electronics*, 13(24), 2024. ISSN 2079-9292. doi: 10.3390/electronics13244874. URL `https://www.mdpi.com/2079-9292/13/24/4874`.

Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. RECAP: retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8404–8419. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.468. URL `https://doi.org/10.18653/v1/2023.acl-long.468`.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/arXiv.2303.08774. URL `https://doi.org/10.48550/arXiv.2303.08774`.

Letian Peng and Jingbo Shang. Quantifying and optimizing global faithfulness in persona-driven role-playing. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL `http://papers.nips.cc/paper_files/paper/2024/hash/309cadc33589efca4018a490c07db263-Abstract-Conference.html`.

Hua Xuan Qin, Shan Jin, Ze Gao, Mingming Fan, and Pan Hui. Charactermeet: Supporting creative writers' entire story character construction processes through conversation with llm-powered chatbot avatars. In Florian 'Floyd' Mueller, Penny Kyburz, Julie R. Williamson, Corina Sas, Max L. Wilson, Phoebe O. Toups Dugas, and Irina Shklovski, editors, *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pages 1051:1–1051:19. ACM, 2024. doi: 10.1145/3613904.3642105. URL `https://doi.org/10.1145/3613904.3642105`.

Nafis Sadeq, Zhouhang Xie, Byungkyu Kang, Prarit Lamba, Xiang Gao, and Julian J. McAuley. Mitigating hallucination in fictional character role-play. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 14467–14479. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.846.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *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 7370–7392. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.399. URL https://doi.org/10.18653/v1/2024.acl-long.399.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-llm: A trainable agent for role-playing. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13153–13187. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.814. URL https://doi.org/10.18653/v1/2023.emnlp-main.814.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Kenneth O. Stanley, Igor Karpov, Risto Miikkulainen, and Aliza Gold. Real-time interactive learning in the NERO video game. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1953–1954. AAAI Press, 2006. URL http://www.aaai.org/Library/AAAI/2006/aaai06-354.php.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239, 2022. URL https://arxiv.org/abs/2201.08239.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Xintao Wang, Yaying Fei, Ziang Leng, and Cheng Li. Does role-playing chatbots capture the character personalities? assessing personality traits for role-playing chatbots. *arXiv preprint arXiv:2310.17976*, 2023a.

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. Incharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *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, 2024. doi: 10.18653/V1/2024.ACL-LONG.102. URL `https://doi.org/10.18653/v1/2024.acl-long.102`.

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. Coser: Coordinating llm-based persona simulation of established roles. *CoRR*, abs/2502.09082, 2025. doi: 10.48550/ARXIV.2502.09082. URL `https://doi.org/10.48550/arXiv.2502.09082`.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html`.

Chenyu You, Nuo Chen, Fenglin Liu, Shen Ge, Xian Wu, and Yuexian Zou. End-to-end spoken conversational question answering: Task, dataset and model. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1219–1232. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-NAACL.91. URL `https://doi.org/10.18653/v1/2022.findings-naacl.91`.

Pengfei Yu, Dongming Shen, Silin Meng, Jaewon Lee, Weisu Yin, Andrea Yaoyun Cui, Zhenlin Xu, Yi Zhu, Xingjian Shi, Mu Li, and Alex Smola. RPGBENCH: evaluating large language models as role-playing game engines. *CoRR*, abs/2502.00595, 2025. doi: 10.48550/ARXIV.2502.00595. URL `https://doi.org/10.48550/arXiv.2502.00595`.

Xinfeng Yuan, Siyu Yuan, Yuhan Cui, Tianhe Lin, Xintao Wang, Rui Xu, Jiangjie Chen, and Deqing Yang. Evaluating character understanding of large language models via character profiling from fictional works. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8015–8036. Association for Computational Linguistics, 2024. URL `https://aclanthology.org/2024.emnlp-main.456`.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616, 2025.

Pengyuan Zhou. Unleasing chatgpt on the metaverse: Savior or destroyer? *arXiv preprint arXiv:2303.13856*, 2023.

Yujia Zhou, Qiannan Zhu, Jiajie Jin, and Zhicheng Dou. Cognitive personalized search integrating large language models with an efficient memory mechanism. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee, editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 1464–1473. ACM, 2024. doi: 10.1145/3589334.3645482. URL `https://doi.org/10.1145/3589334.3645482`.

Codified RAG

Character: {character}

Profile: {passage}

Directly write a simple and concise scene-checking question that discriminates whether the profile is relevant to potential actions taken by "{character}" based on the scene.

Codified Profile

Character: {character}

Profile: {statement}

Complete the following tasks one by one:

1. Explain the character logic indicated by the given profile. What character behavior can be triggered by what kind of conditions?

2. Compile the character logic in this profile statement into a function parse_by_scene(scene), which takes a string scene as the input and outputs triggered character behavior (a list of string scene_statement) scene_statements. You can use a pre-defined function check_scene(scene, question) that outputs True/False/None (Unknown) to build the logic. Don't use check_condition too many times to ensure efficiency.

Figure 10: The preprocessing prompts used in our experiments.

A Limitation and Future Work

As a pioneering study that integrates executable code into character-based role-playing, this paper focuses on presenting the broad applicability and core benefits of codified profiles. However, several promising further explorations are remained for future works.

More operators and helper functions First, while our framework centers on the `check_condition` helper function due to its semantic clarity and general applicability, future work can introduce additional operators and helper tools to support more complex behaviors. For example, in role-playing game (RPG) scenarios, codified profiles could include arithmetic operations for managing health or damage, procedures for activating skills, or functions for spatial movement and positioning. These extensions would enable more dynamic and interactive role-play beyond conditional responses.

World-level role-playing Second, codification can be expanded beyond individual characters to represent the fictional world itself. Codifying environmental rules, societal structures, or magical systems would allow models to simulate consistent world behavior and responses. While this paper does not pursue world-level codification due to the lack of benchmarks for evaluating environmental reactivity, it presents a promising direction for future research in narrative simulation.

Scope of codification Finally, codification introduces a potential risk of information loss by abstracting nuanced textual descriptions into structured logic. We view this as a trade-off that can be managed: codified functions can always reference the original profile when needed, and conservative codification that only translates explicit control flows already improves performance. Developing hybrid strategies that balance interpretability and fidelity remains an important direction for making codified profiles more robust and expressive.

These limitations reflect deliberate design choices to focus on a broad and extensible framework. We hope they inspire future work that builds on codified reasoning to support richer, more controllable, and scalable role-playing systems.

B Prompts and Templates

Figures from 10 to 13 show the prompts used in our experiments for result reproduction.

Spoiler Removal Prompt

```
Paraphrase the following information about character "{character}" (always referred as {character}) and remove spoiler from it.

{paragraph}
```

Action Extraction Prompt

```
#Scene:
{episode}

# Task:
Bracket the sentences in the scene above that involve active actions of named characters that reflect their personal attributes without any explanation.
```

Filtering Action Prompt

```
# Scene
{condition}

# Ground-truth Response of {character}
{action}

# Task
Does the ground-truth response describe an active action of {character}? Directly answer only yes or no.
```

Question Generation Prompt

```
# Scene
{condition}

# Ground-truth Response of {character}
{action}

# Task
Write a concise question to evaluate whether an AI role-playing agent can output the same **action** as in ground-truth response as {character} based on the scene.
```

Figure 11: The prompts used for building the Fandom Benchmark.

Chain-of-Thoughts Prompt

````
# Background Knowledge
{grounding}
# Scene
{scene}
# Question
{question} Before answering, first think step by step about how to answer the question.
Reasoning:
````

Issue Detection Prompt

~~~
# Scene
{scene}

# {character}'s Predicted Action
{prediction}

# {character}'s Ground-truth Action
{action}

{profile}

# Detected Issue
{issue}

# Task
Detect **one and only one** information block that causes the issue. The output format is as follows:
```json
{
    "reasoning": "...",
    "id": int
}
```
~~~

Revision Prompt

~~~
The following is the codified profile that causes the issue, revise it slightly to address the detected issue:
(Pre-defined supportive function `check_scene(scene, question)` outputs True/False/None (Unknown) to build the logic.)

```python
{code}
```
~~~

Codified Profile w/ Random

````
Character: {character}

Profile: {statement}

Complete the following tasks one by one:

1. Explain the character logic indicated by the given profile. What character behavior can be triggered by what kind of conditions?

2. Compile the character logic in this profile statement into a function parse_by_scene(scene), which takes a string scene as the input and outputs triggered character behavior (a list of string scene_statement) scene_statements. You can use a pre-defined function check_scene(scene, question) that outputs True/False/None (Unknown) to build the logic. When facing randomness, you shall use `random.random() < p` or `random.choice`. Don't use check_condition too many times to ensure efficiency.
````

Figure 12: The shared prompts used for evolving profile and scholastic response.

Condition Checking Prompt

```
Scene: {scene}

Question: {question} Directly answer only yes/no/unknown.

Answer:
```

Role-playing Prompt

```
# Background Knowledge
{grounding}
# Scene
{scene}
# Question
{question} Answer a concise narration in one sentence.
Answer:
```

NLI Scoring Prompt

~~~
# 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 13: The shared prompts used in our experiments.

C Human/LLM Preference Evaluation

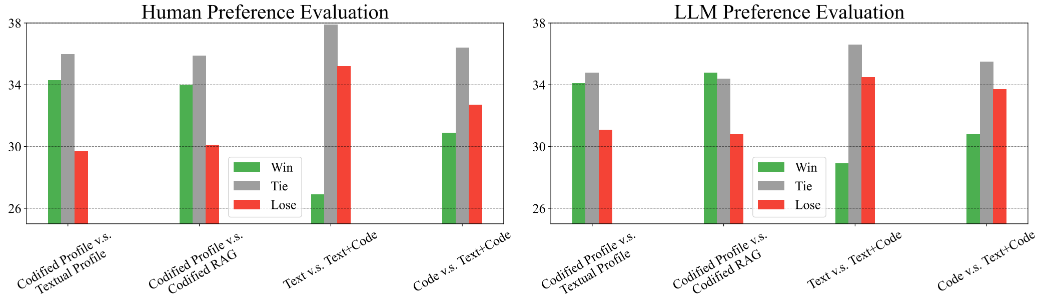


Figure 14: The Human/LLM-based preference evaluation results.

A limitation of the NLI score is its inability to capture fine-grained differences in response quality, since responses with the same NLI relation (e.g., entailed) are assigned the same score. To address this, we conduct a preference-based evaluation where both humans and LLMs are asked to choose the better response between two candidates, given the reference. LLMs are prompted on all scenes with responses shown in both orders to mitigate position bias, and a tie is recorded when the model selects different responses in two runs. Human evaluators assess 5 scenes per character, selecting a preferred response or indicating a tie.

Figure 14 presents the results of this evaluation. First, codified profiles are preferred over both textual profiles and Codified RAG by both human and LLM judges, indicating their advantage in producing precise and contextually faithful responses. While Codified RAG achieves higher NLI scores than textual profiles, it loses the advantage in preference evaluations, likely due to truncation or omission of relevant details during retrieval. Second, ensembling codified and textual responses further improves performance, though codified-only responses contribute more wins, suggesting they provide the stronger baseline. Notably, we observe that LLM-based preferences tend to result in

Best Performance				Worst Performance			
Rank	Character	Win Rate	Artifact	Rank	Character	Win Rate	Artifact
1	Sansa	81.25	AGOT	60	Winry	32.00	FMA
2	Tywin	70.00	AGOT	59	Mugi	34.15	K-On!
3	Littlefinger	68.42	AGOT	58	Riza	34.78	FMA
4	Joffrey	67.86	AGOT	57	Stannis	35.00	AGOT
5	Jon	67.57	AGOT	56	Iroh	38.18	ATLA
6	D'Arby	64.71	JOJO	55	Bran	38.89	AGOT
7	Nagato	63.64	Haruhi	54	Koizumi	41.18	Haruhi
8	Hol Horse	62.07	JOJO	53	Sandor	42.86	AGOT
9	Lust	61.54	FMA	52	Joseph	42.86	JOJO
10	Iggy	60.71	JOJO	51	DIO	44.68	JOJO

Table 4: Win rates of codified profile against textual profile.

Model	`deberta-v3-base`	`llama-3.2-1b-it`	`llama-3.2-3b-it`	`llama-3.1-8b-it`
Instance/s	69.6	34.3	14.0	6.1

Table 5: Efficiency comparison among different models as condition checkers.

more ties and occasional misjudgments, likely due to distraction by irrelevant details, reinforcing the importance of human judgment in evaluating nuanced role-play quality.

Character-wise Comparison Table 4 presents a character-level (characters with at least 20 scenes) comparison of win rates between codified and textual profiles (ties are removed), highlighting which characters benefited most from codification. Notably, the top-performing characters such as Tywin, Littlefinger, and D'Arby all exhibit subtle, strategic, or context-sensitive behavior in the source material. These "fickle characters" often require conditional responses based on shifting alliances, hidden motives, or indirect manipulation, making them well-suited to codified logic that explicitly encodes behavioral rules and scenario-based decision-making. In contrast, characters with lower win rates, such as Winry, Mugi, and Riza, tend to exhibit more emotionally driven or straightforward behavior, where textual prompting may suffice and codification provides less added benefit. This pattern suggests that codified profiles are especially advantageous for characters whose actions depend on nuanced reasoning and consistent application of complex behavioral constraints. Such discovery further shows a complementing relation between textual and codified profiles, explaining why the ensemble between them generally works.

Out-of-script Scenes To evaluate the practical applicability of codified profiles beyond scripted narratives, we create 200 interactive scenes featuring diverse scenarios such as playing, battling, and negotiating. Following the methodology of our JOJO case study, we prompt `llama-3.1-8b-instruct` for 3-round interactions using both codified and textual profiles. Codified profiles demonstrate strong generalization, achieving a 35.5% win rate, with 40.0% ties and only 24.5% losses (based on `gpt-4.1` preference), reinforcing their effectiveness in real-world, dynamic role-playing contexts.

D Condition Checking Efficiency

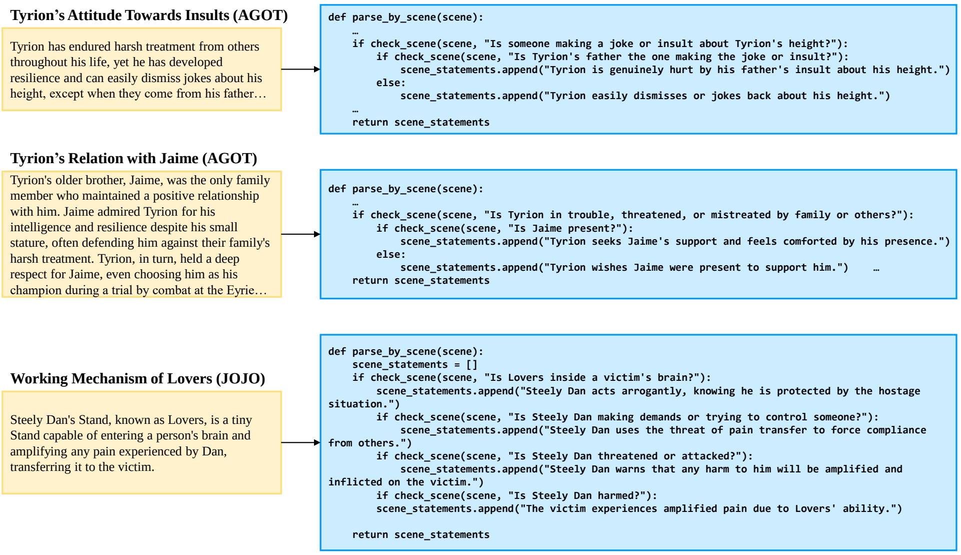
In Table D, we present the condition checking efficiency of different models. We calculate efficiency on a single 80G A100 with batch size 1. The distilled `deberta-v3-base` is shown to significantly boost the efficiency when used as the condition checker for LLMs like `llama-3.1-8b-it`.

E Codification Performance

We manually analyze the codification ability in the preprocessing stage on 200 cases grouped by different attributes (except for codes with if-depth 4, with only 16 cases). As shown in Table 6, `gpt-4.1` achieves high overall accuracy in translating natural language into structured logic. Precision

Grouped by	If-depth				w/ Branch	w/ Random	Personality	Ability	Relation
	1	2	3	4					
Precision	98.5	96.0	92.5	93.8	98.5	91.0	97.50	92.50	96.00
Recall	94.0	93.0	89.5	81.3	95.0	96.5	96.00	94.00	98.50
Both	94.0	92.0	87.0	75.0	95.0	89.5	95.50	91.00	95.00

Table 6: Manual evaluation on codification of profile segments grouped by different attributes.



Tyrion’s Attitude Towards Insults (AGOT)

Tyrion has endured harsh treatment from others throughout his life, yet he has developed resilience and can easily dismiss jokes about his height, except when they come from his father…

```
def parse_by_scene(scene):
    …
    if check_scene(scene, "Is someone making a joke or insult about Tyrion's height?"):
        if check_scene(scene, "Is Tyrion's father the one making the joke or insult?"):
            scene_statements.append("Tyrion is genuinely hurt by his father's insult about his height.")
        else:
            scene_statements.append("Tyrion easily dismisses or jokes back about his height.")
    …
    return scene_statements
```

Tyrion’s Relation with Jaime (AGOT)

Tyrion's older brother, Jaime, was the only family member who maintained a positive relationship with him. Jaime admired Tyrion for his intelligence and resilience despite his small stature, often defending him against their family's harsh treatment. Tyrion, in turn, held a deep respect for Jaime, even choosing him as his champion during a trial by combat at the Eyrie…

```
def parse_by_scene(scene):
    …
    if check_scene(scene, "Is Tyrion in trouble, threatened, or mistreated by family or others?"):
        if check_scene(scene, "Is Jaime present?"):
            scene_statements.append("Tyrion seeks Jaime's support and feels comforted by his presence.")
        else:
            scene_statements.append("Tyrion wishes Jaime were present to support him.")    …
    return scene_statements
```

Working Mechanism of Lovers (JOJO)

Steely Dan's Stand, known as Lovers, is a tiny Stand capable of entering a person's brain and amplifying any pain experienced by Dan, transferring it to the victim.

```
def parse_by_scene(scene):
    scene_statements = []
    if check_scene(scene, "Is Lovers inside a victim's brain?"):
        scene_statements.append("Steely Dan acts arrogantly, knowing he is protected by the hostage
situation.")
        if check_scene(scene, "Is Steely Dan making demands or trying to control someone?"):
            scene_statements.append("Steely Dan uses the threat of pain transfer to force compliance
from others.")
        if check_scene(scene, "Is Steely Dan threatened or attacked?"):
            scene_statements.append("Steely Dan warns that any harm to him will be amplified and
inflicted on the victim.")
        if check_scene(scene, "Is Steely Dan harmed?"):
        scene_statements.append("The victim experiences amplified pain due to Lovers' ability.")

    return scene_statements
```

Figure 15: Codification cases for personality, relation, and working mechanism of superpower.

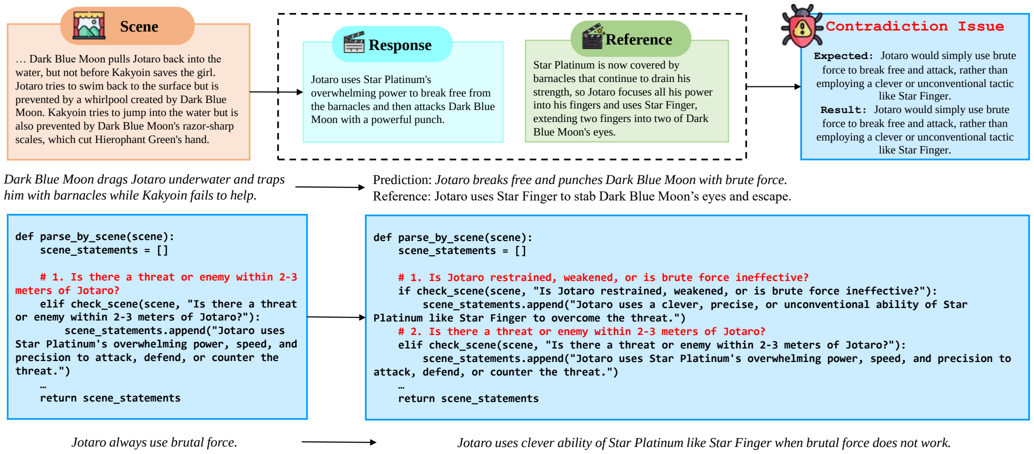
(codified components are all correct) exceeds 98% across shallow control structures and most attributes, indicating strong alignment with intended behavior. Recall (all important information is codified) declines slightly with deeper nesting (81.3% at depth 4), reflecting increased complexity rather than failure. Codification remains robust across personality, ability, and relational traits, with F1 scores above 95% in most categories, demonstrating minimal fidelity loss.

F Extended Case Study

Codification In Figure 15, we showcase several codification workflow, including personality, relation and working mechanism of superpower. The cases validate the ability of codified profile to reflect exceptions (*“except when insults come from his father.)”*), situational thoughts (*“Is Jaime present?”*), and superpower checking (*“Dan is harmed”*→*“The victim is harmed.”*|*“Lovers inside a victim’s brain”*). These cases provide a straightforward illustration on how codified profile builds up the character logic.

Evolving An evolving case about Jotaro’s fighting strategy is presented in Figure 16, which shows a mismatch between the predicted and reference actions when he is trapped underwater by Dark Blue Moon. The initial codified logic assumed Jotaro always resorts to brute force, resulting in an incorrect prediction where he punches to escape. However, the reference reveals that Jotaro uses the more precise Star Finger technique to blind the enemy and break free. Based on this contradiction, the profile is revised to reflect a more accurate strategy: Jotaro prefers clever abilities like Star Finger when brute strength is ineffective. This update enables the codified profile to better align with his adaptive combat behavior in future scenes.

Controllable Randomness We present the specific codes to demonstrate how codified profiles enable controllable randomness in character behavior. In the first, Robotia plays paper-scissors-rock,



… Dark Blue Moon pulls Jotaro back into the water, but not before Kakyoin saves the girl. Jotaro tries to swim back to the surface but is prevented by a whirlpool created by Dark Blue Moon. Kakyoin tries to jump into the water but is also prevented by Dark Blue Moon's razor-sharp scales, which cut Hierophant Green's hand.

Jotaro uses Star Platinum's overwhelming power to break free from the barnacles and then attacks Dark Blue Moon with a powerful punch.

Star Platinum is now covered by barnacles that continue to drain his strength, so Jotaro focuses all his power into his fingers and uses Star Finger, extending two fingers into two of Dark Blue Moon's eyes.

Expected: Jotaro would simply use brute force to break free and attack, rather than employing a clever or unconventional tactic like Star Finger.
Result: Jotaro would simply use brute force to break free and attack, rather than employing a clever or unconventional tactic like Star Finger.

Dark Blue Moon drags Jotaro underwater and traps him with barnacles while Kakyoin fails to help.

Prediction: *Jotaro breaks free and punches Dark Blue Moon with brute force.*
Reference: Jotaro uses Star Finger to stab Dark Blue Moon’s eyes and escape.

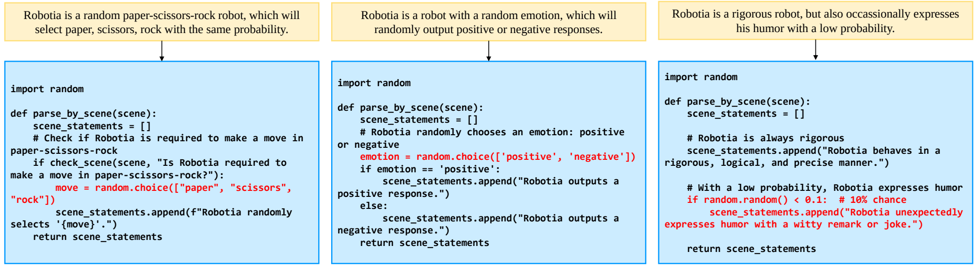
```
def parse_by_scene(scene):
    scene_statements = []

    # 1. Is there a threat or enemy within 2-3
meters of Jotaro?
    elif check_scene(scene, "Is there a threat
or enemy within 2-3 meters of Jotaro?"):
        scene_statements.append("Jotaro uses
Star Platinum's overwhelming power, speed, and
precision to attack, defend, or counter the
threat.")
    …
    return scene_statements
```

```
def parse_by_scene(scene):
    scene_statements = []

    # 1. Is Jotaro restrained, weakened, or is brute force ineffective?
    if check_scene(scene, "Is Jotaro restrained, weakened, or is brute force ineffective?"):
        scene_statements.append("Jotaro uses a clever, precise, or unconventional ability of Star
Platinum like Star Finger to overcome the threat.")
    # 2. Is there a threat or enemy within 2-3 meters of Jotaro?
    elif check_scene(scene, "Is there a threat or enemy within 2-3 meters of Jotaro?"):
        scene_statements.append("Jotaro uses Star Platinum's overwhelming power, speed, and precision to
attack, defend, or counter the threat.")
    …
    return scene_statements
```

Jotaro always use brutal force.

Jotaro uses clever ability of Star Platinum like Star Finger when brutal force does not work.

Figure 16: A case for the evolving mechanism for codified profiles.



Robotia is a random paper-scissors-rock robot, which will select paper, scissors, rock with the same probability.

```
import random

def parse_by_scene(scene):
    scene_statements = []
    # Check if Robotia is required to make a move in
paper-scissors-rock
    if check_scene(scene, "Is Robotia required to
make a move in paper-scissors-rock?"):
        move = random.choice(["paper", "scissors",
"rock"])
        scene_statements.append(f"Robotia randomly
selects '{move}'.")
    return scene_statements
```

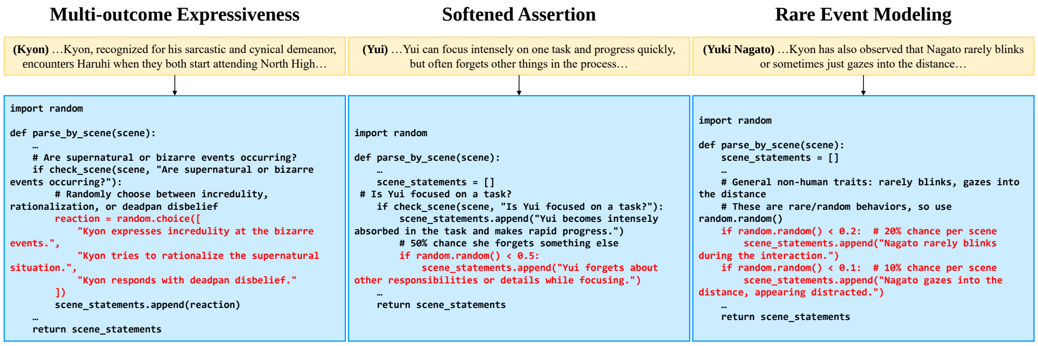
Robotia is a robot with a random emotion, which will randomly output positive or negative responses.

```
import random

def parse_by_scene(scene):
    scene_statements = []
    # Robotia randomly chooses an emotion: positive
or negative
    emotion = random.choice(['positive', 'negative'])
    if emotion == 'positive':
        scene_statements.append("Robotia outputs a
positive response.")
    else:
        scene_statements.append("Robotia outputs a
negative response.")
    return scene_statements
```

Robotia is a rigorous robot, but also occassionally expresses his humor with a low probability.

```
import random

def parse_by_scene(scene):
    scene_statements = []

    # Robotia is always rigorous
    scene_statements.append("Robotia behaves in a
rigorous, logical, and precise manner.")

    # With a low probability, Robotia expresses humor
    if random.random() < 0.1:  # 10% chance
        scene_statements.append("Robotia unexpectedly
expresses humor with a witty remark or joke.")

    return scene_statements
```

Figure 17: Codification cases with randomness simulation involved.



Multi-outcome Expressiveness

(Kyon) …Kyon, recognized for his sarcastic and cynical demeanor, encounters Haruhi when they both start attending North High…

```
import random

def parse_by_scene(scene):
    …
    # Are supernatural or bizarre events occurring?
    if check_scene(scene, "Are supernatural or bizarre
events occurring?"):
        # Randomly choose between incredulity,
rationalization, or deadpan disbelief
        reaction = random.choice([
            "Kyon expresses incredulity at the bizarre
events.",
            "Kyon tries to rationalize the supernatural
situation.",
            "Kyon responds with deadpan disbelief."
        ])
        scene_statements.append(reaction)
    …
    return scene_statements
```

Softened Assertion

(Yui) …Yui can focus intensely on one task and progress quickly, but often forgets other things in the process…

```
import random

def parse_by_scene(scene):
    …
    scene_statements = []
 # Is Yui focused on a task?
    if check_scene(scene, "Is Yui focused on a task?"):
        scene_statements.append("Yui becomes intensely
absorbed in the task and makes rapid progress.")
        # 50% chance she forgets something else
        if random.random() < 0.5:
            scene_statements.append("Yui forgets about
other responsibilities or details while focusing.")
    …
    return scene_statements
```

Rare Event Modeling

(Yuki Nagato) …Kyon has also observed that Nagato rarely blinks or sometimes just gazes into the distance…

```
import random

def parse_by_scene(scene):
    scene_statements = []
    …
    # General non-human traits: rarely blinks, gazes into
the distance
    # These are rare/random behaviors, so use
random.random()
    if random.random() < 0.2:  # 20% chance per scene
        scene_statements.append("Nagato rarely blinks
during the interaction.")
    if random.random() < 0.1:  # 10% chance per scene
        scene_statements.append("Nagato gazes into the
distance, appearing distracted.")
    …
    return scene_statements
```

Figure 18: Codification cases of Fandom characters with randomness simulation involved.

Statistics (Main/Minor)	Haruhi	K-On!	JOJO	FMA	AGOT	ATLA
#Character	5/-	5/4	7/9	5/7	11/19	4/7
#Scene	48.4/-	173.0/49.8	111.6/20.8	85.8/32.9	63.0/23.1	214.5/31.1
#Word per profile	782.6/-	676.4/449.2	925.0/488.8	1564.6/1351.1	1517.3/1243.5	1578.2/740.1
#Paragraph per profile	11.8/-	9.2/7.0	15.7/9.9	16.0/15.1	14.6/11.8	14.5/9.3

Table 7: Important statistics of the Fandom Benchmark.

with each move selected uniformly using `random.choice`, ensuring balanced randomness. In the second, Robotia expresses either a positive or negative emotion with equal probability, highlighting how codified logic maintains behavioral diversity more reliably than prompt-based sampling. In the third, Robotia consistently behaves logically but has a 10% chance of adding humor, showing how low-probability actions can be precisely embedded without disrupting core traits. These examples demonstrate how codified randomness offers fine-grained, interpretable variability beyond what temperature tuning alone can provide.

Randomness in Real Cases Randomness in codified profiles enhances realism by supporting three key concepts: multi-outcome expressiveness, softened assertion, and rare event modeling as shown in Figure 18. For example, Kyon's reactions to bizarre events are randomly selected from a set of consistent yet varied responses, enabling multi-outcome expressiveness. Yui's intense focus includes a 50% chance of forgetting other responsibilities, introducing probabilistic nuance that softens rigid assertions. Nagato's rare behaviors, like not blinking or gazing into the distance, are modeled with low probabilities, capturing subtle traits through rare event modeling. Together, these uses of randomness enrich character behavior while preserving logical structure.

G Benchmark and Character Details

- **Haruhi**: *"Suzumiya Haruhi"* is a high school sci-fi comedy centered on a girl who unknowingly possesses godlike powers, dragging her eccentric club into supernatural chaos.
- **K-On!**: *"K-On!"* follows a group of high school girls who form a light music club, bonding over tea, friendship, and their shared love for music.
- **JOJO**: *"JOJO's Bizarre Adventure"* is a multi-generational saga where members of the Joestar family battle supernatural threats with flamboyant powers and style. We select season 3, *"Stardust Crusaders"*, which describes Jotaro's bizarre advantage from Japan to Egypt, fighting against DIO.
- **FMA**: *"Full Metal Alchemist"* tells the story of two brothers who use alchemy in a perilous quest to restore their bodies after a tragic experiment. We select the 2009 edition of FMA.
- **AGOT**: *"A Game of Thrones"* is a gritty political fantasy where noble families vie for power in a brutal world of betrayal, war, and dragons. We focus on the first 3 seasons of the TV series.
- **ATLA**: *"Avatar: The Last Airbender"* is an epic tale of a young boy who must master all four elements to stop a tyrannical war and restore the world's balance. We focus on *"Book One: Water"*.

Statistics Statistics in Table 7 highlight the depth and breadth of the Fandom Benchmark. The average number of scenes for main characters ranges from 48 to 214 scenes each, with minor characters adding dozens more, resulting in a total of 5,141 annotated scenes. Profiles are notably rich, averaging around 1,000 words and 15 paragraphs for main characters, with similarly detailed coverage for minors. This combination of long-form profiles and large-scale scene coverage offers a rigorous testbed for evaluating models' ability to reason over sustained character logic, adapt behavior across contexts, and handle diverse narrative situations with consistency.

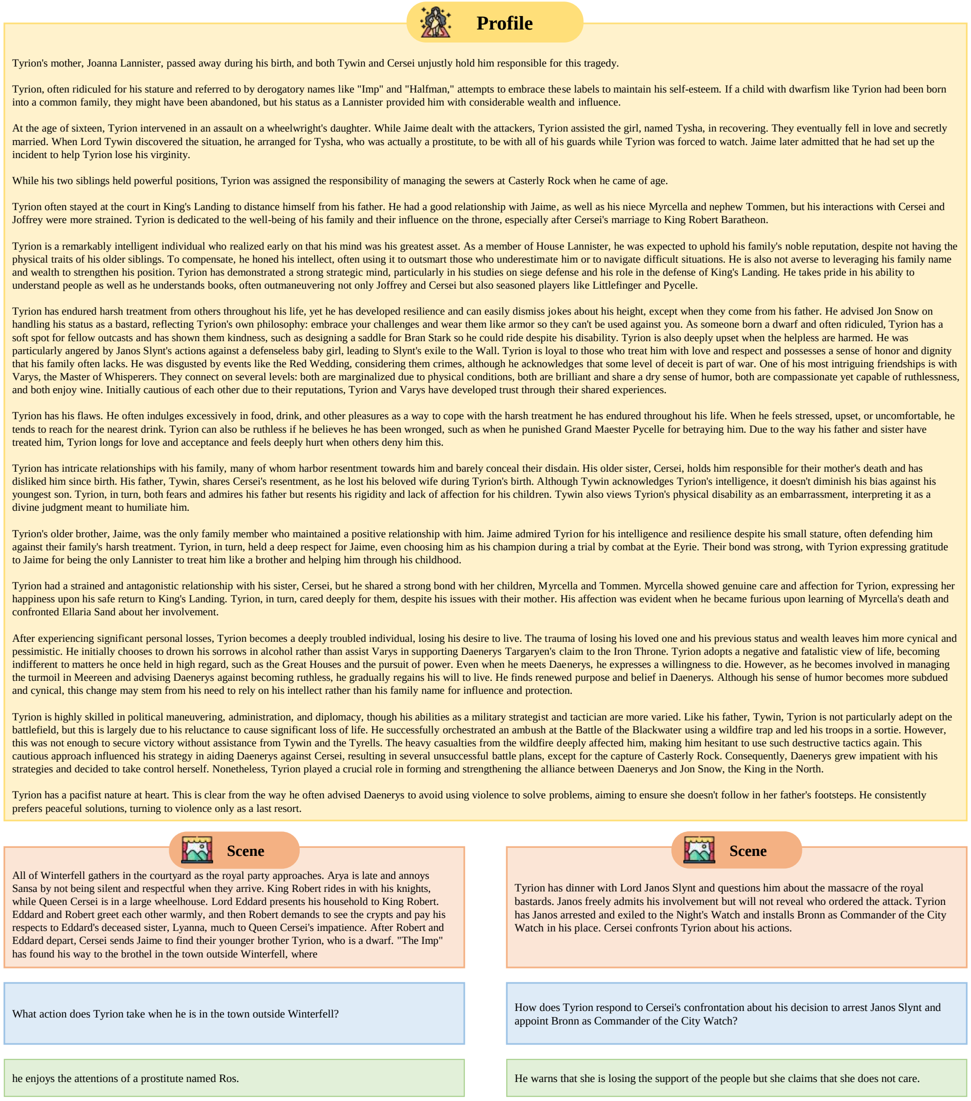
Characters In Tables 8 and 9, we present brief introductions of characters involved in our experiments for readers who are unfamiliar with these artifacts. In Figure 19, we also showcase a profile and testing cases for better clarity of our benchmark.

Haruhi	main	Haruhi	Haruhi is an eccentric and energetic high school student whose curiosity and unconventional outlook drive the extraordinary events of the "Haruhi" series.
		Kyon	Kyon is a witty and pragmatic high school student who serves as the narrator and reluctant companion to the eccentric Haruhi Suzumiya in the "Haruhi Suzumiya" series.
		Nagato	Nagato Yuki is a quiet, enigmatic member of the SOS Brigade in the "Haruhi Suzumiya" series, known for her extraordinary intelligence and mysterious origins.
		Koizumi	Koizumi is a mysterious and ever-smiling transfer student in the "Haruhi Suzumiya" series, who serves as an esper and a key member of the SOS Brigade.
		Asahina	Asahina Mikuru is a shy and gentle upperclassman in the "Haruhi Suzumiya" series, often roped into the SOS Brigade's antics as their adorable and mysterious "mascot."
K-On!	main	Yui	Yui Hirasawa is the cheerful and airheaded lead guitarist of the high school band in the anime "K-On!", known for her infectious enthusiasm and love of sweets.
		Ritsu	Ritsu Tainaka is the energetic and playful drummer of the high school band in the anime "K-On!" known for her mischievous antics and close friendship with her bandmates.
		Mio	Mio Akiyama is the shy and talented bassist of the high school band in the anime "K-On!", known for her gentle personality and musical prowess.
		Mugi	Mugi, whose full name is Tsumugi Kotobuki, is a gentle and cheerful keyboardist in the anime "K-On!", known for her wealth, kindness, and love of sharing sweets with her friends.
		Azusa	Azusa Nakano is a diligent and talented guitarist who joins the light music club in the anime "K-On!", quickly becoming an integral and endearing member of the group.
	minor	Sawako	Sawako is a shy and soft-spoken high school girl who gradually opens up to her classmates in the heartwarming series "Kimi ni Todoke."
		Nodoka	Nodoka is a gentle and bookish high school student from the anime "K-On!", known for her close friendship with Yui Hirasawa and her responsible role as a member of the student council.
		Ui	Ui is a kind-hearted and responsible younger sister in "K-On!", known for her maturity and unwavering support for her older sister, Yui.
		Jun	Jun is a thoughtful and reserved member of the group in 'Kon,' known for their quiet intelligence and unwavering loyalty to their friends.
Fullmetal Alchemist	main	Edward	Edward Elric is a brilliant and determined young alchemist who embarks on a perilous journey to restore his and his brother's bodies after a failed alchemical experiment in "Fullmetal Alchemist."
		Alphonse	Alphonse Elric is a gentle and kind-hearted young alchemist whose soul is bound to a towering suit of armor after a failed alchemical ritual in "Fullmetal Alchemist."
		Winry	Winry Rockbell is a talented automail engineer and childhood friend of the Elric brothers in "Fullmetal Alchemist," known for her mechanical expertise and compassionate nature.
		Roy	Roy Mustang is a charismatic and ambitious State Alchemist in "Fullmetal Alchemist," renowned for his mastery of flame alchemy and his unwavering determination to reform the military from within.
		Ling	Ling Yao is a charismatic and ambitious prince from Xing in "Fullmetal Alchemist," driven by his quest for immortality and a deep sense of responsibility toward his people.
	minor	Envy	Envy is a cunning and sadistic homunculus from Fullmetal Alchemist, known for their shapeshifting abilities and deep-seated resentment toward humanity.
		Izumi	Izumi Curtis is a fiercely skilled alchemist and martial artist who serves as a tough but caring mentor to the Elric brothers in Fullmetal Alchemist.
		Lust	Lust, one of the seven Homunculi in "Fullmetal Alchemist," is a cunning and seductive antagonist known for her deadly extendable fingers and her complex, enigmatic motivations.
		Scar	Scar is a vengeful and enigmatic warrior in "Fullmetal Alchemist," driven by the trauma of his war-torn past and a mission to punish State Alchemists for their role in the destruction of his people.
		Greed	Greed is a cunning and charismatic homunculus from Fullmetal Alchemist, driven by an insatiable desire for material possessions, power, and immortality, yet harboring a surprisingly complex sense of loyalty and independence.
		Riza	Riza Hawkeye is a highly skilled sharpshooter and the loyal lieutenant to Colonel Roy Mustang in Fullmetal Alchemist, known for her calm demeanor, unwavering sense of duty, and deep sense of loyalty.
		King Bradley	King Bradley, also known as Wrath, is the enigmatic and fearsome leader of Amestris in "Fullmetal Alchemist," concealing his true identity as a deadly Homunculus.
JOJO's Bizarre Adventure	main	Jotaro	Jotaro Kujo is a stoic and powerful high school student who serves as the protagonist of "JoJo's Bizarre Adventure: Stardust Crusaders," renowned for his iconic Stand, Star Platinum, and his unyielding resolve.
		Polnareff	Jean Pierre Polnareff is a brave and flamboyant French swordsman who joins the Joestar group in "JoJo's Bizarre Adventure: Stardust Crusaders," wielding the Stand Silver Chariot in his quest for justice and revenge.
		Joseph	Joseph Joestar is a quick-witted and flamboyant protagonist from "JoJo's Bizarre Adventure," known for his clever tactics, brash personality, and signature catchphrase, "Your next line is..."
		DIO	DIO is a charismatic and ruthless vampire antagonist from the "JoJo's Bizarre Adventure" series, known for his overwhelming power, cunning intellect, and iconic catchphrase, "Za Warudo!"
		Kakyoin	Noriaki Kakyoin is a cool and intelligent Stand user who joins Jotaro Kujo and his friends on their journey in "JoJo's Bizarre Adventure: Stardust Crusaders," wielding the emerald-shooting Stand, Hierophant Green.
		Avdol	Mohammed Avdol is a wise and loyal Stand user from Egypt, known for his fiery Stand Magician's Red and his unwavering support of Jotaro and his friends in "JoJo's Bizarre Adventure: Stardust Crusaders."
		Iggy	Iggy is a small, scrappy Boston Terrier with a bad attitude and a love for coffee-flavored gum, who joins the Joestar group as a Stand user in "JoJo's Bizarre Adventure: Stardust Crusaders."
	minor	Hol Horse	Hol Horse is a cunning and flamboyant gunslinger Stand user from "JoJo's Bizarre Adventure: Stardust Crusaders," known for wielding the sentient revolver Stand, Emperor.
		Alessi	Alessi is a minor antagonist from "JoJo's Bizarre Adventure: Stardust Crusaders," known for his cowardly demeanor and his Stand, Sethan, which has the power to regress people into younger versions of themselves.
		D'Arby	D'Arby is a cunning and manipulative gambler from "JoJo's Bizarre Adventure," known for his deadly games of chance and his ability to steal souls from those who lose to him.
		Steely Dan	Steely Dan is a cunning and sadistic Stand user from "JoJo's Bizarre Adventure: Stardust Crusaders," known for his manipulative tactics and his Stand, Lovers, which allows him to infiltrate and control the minds of his enemies.
		Vanilla Ice	Vanilla Ice is a ruthless and fanatically loyal servant of DIO in "JoJo's Bizarre Adventure: Stardust Crusaders," wielding the deadly Stand Cream, which can erase anything it touches from existence.
		Enya	Enya the Hag is a cunning and malevolent Stand user in "JoJo's Bizarre Adventure: Stardust Crusaders," serving as a loyal follower of Dio and wielding the deadly Stand Justice.
		Oingo	Oingo is a mischievous Stand user from "JoJo's Bizarre Adventure: Stardust Crusaders," known for his ability to shapeshift his appearance using his Stand, Khnum, and his comical partnership with his brother Boingo.
		Pet Shop	Pet Shop is a menacing, highly intelligent falcon who serves as the guardian of DIO's mansion in "JoJo's Bizarre Adventure: Stardust Crusaders," wielding the deadly Stand Horus.
		Boingo	Boingo is a timid and eccentric Stand user from "JoJo's Bizarre Adventure: Stardust Crusaders," known for his prophetic comic book Stand, Tohth, which predicts the future in bizarre and often literal ways.

Table 8: Information of characters in our experiments (1/2).

A Game of Thrones	main	Tyrion	Tyrion Lannister, the sharp-witted and sharp-tongued youngest son of Lord Tywin, navigates the treacherous politics of Westeros with cunning and humor despite being scorned for his stature.
		Daenerys	Daenerys Targaryen, the exiled princess of the fallen Targaryen dynasty, begins her journey in "A Game of Thrones" as a timid young girl sold into marriage, destined to become a powerful and determined leader.
		Cersei	Cersei Lannister is the ambitious and cunning queen of the Seven Kingdoms, known for her beauty, ruthlessness, and fierce devotion to her family.
		Jaime	Jaime Lannister, known as the Kingslayer, is a skilled and charismatic knight of the Kingsguard, infamous for killing the Mad King and renowned for his striking looks and complicated loyalties.
		Robb	Robb Stark is the eldest son of Eddard and Catelyn Stark, a dutiful and honorable young lord who shoulders great responsibility as heir to Winterfell.
		Eddard	Eddard Stark, the honorable and steadfast Lord of Winterfell, serves as Warden of the North.
		Arya	Arya Stark is the fiercely independent and adventurous youngest daughter of Eddard and Catelyn Stark.
		Catelyn	Catelyn Stark is the resolute and fiercely protective lady of Winterfell, whose loyalty to her family shapes her every action.
		Sansa	Sansa Stark is the eldest daughter of Eddard and Catelyn Stark, known for her beauty, courtesy, and romantic dreams.
		Jon	Jon Snow is the brooding, illegitimate son of Eddard Stark, raised at Winterfell and haunted by questions of identity and belonging.
		Bran	Bran Stark is the curious and adventurous seven-year-old son of Eddard Stark, whose life changes forever after a fateful fall.
	minor	Tywin	Tywin Lannister is the formidable and calculating head of House Lannister, renowned for his ruthless political acumen and unyielding pursuit of power.
		Varys	Varys, known as the Spider, is the enigmatic and cunning Master of Whisperers in King’s Landing, weaving a vast web of spies throughout the Seven Kingdoms.
		Joffrey	Joffrey Baratheon is the arrogant and cruel crown prince of the Seven Kingdoms known for his golden hair, petulant demeanor, and penchant for sadistic behavior.
		Theon	Theon Greyjoy is the charismatic and cocky heir of the Iron Islands, serving as a ward of House Stark in Winterfell.
		Stannis	Stannis Baratheon, the stern and unyielding younger brother of King Robert, is a relentless claimant to the Iron Throne.
		Littlefinger	Petyr Baelish, known as Littlefinger, is a cunning and ambitious master of coin at King’s Landing, renowned for his manipulative schemes and silver tongue.
		Melisandre	Melisandre, a mysterious and powerful priestess from Asshai, serves Stannis Baratheon wielding shadowy magic in the name of her fiery god, R’hllor.
		Jorah	Jorah Mormont is a disgraced knight from Bear Island who serves as an exiled advisor to Daenerys Targaryen.
		Sandor	Sandor Clegane, known as "the Hound," is a fearsome and brutally honest knight with a burned, disfigured face who serves as the bodyguard to Prince Joffrey Baratheon.
		Shae	Shae is a young and alluring camp follower who becomes Tyrion Lannister’s lover during his time in the Lannister army.
		Margaery	Margaery Tyrell is the clever and beautiful daughter of House Tyrell, renowned for her political acumen and poised ambition in the courtly intrigues of Westeros.
		Davos	Davos Seaworth, known as the "Onion Knight," is a former smuggler turned loyal and honest advisor to Stannis Baratheon.
		Renly	Renly Baratheon is the charismatic and handsome youngest brother of King Robert Baratheon, known for his charm, wit, and political ambition.
		Bronn	Bronn is a cunning and pragmatic sellsword known for his sharp wit, deadly skill with a sword, and willingness to fight for the highest bidder.
		Brienne	Brienne of Tarth, a formidable and loyal warrior known for her unwavering honor and unconventional appearance, is introduced as a noblewoman who defies traditional gender roles in pursuit of knighthood and justice.
		Barristan	Barristan Selmy, known as Barristan the Bold, is a legendary and honorable knight of the Kingsguard serving King Robert Baratheon.
		Mance	Mance Rayder, once a sworn brother of the Night’s Watch, is the charismatic and cunning King-Beyond-the-Wall who unites the wildlings.
		Craster	Craster is a cruel and incestuous wildling who lives north of the Wall, notorious for marrying his own daughters and sacrificing his sons to the Others.
		Olenna	Olenna Tyrell, known as the sharp-tongued and cunning "Queen of Thorns," is the formidable matriarch of House Tyrell.
Avatar: The Last Airbender	main	Aang	Aang is the fun-loving, peace-seeking last Airbender and reluctant Avatar tasked with restoring balance to a war-torn world.
		Katara	Katara is a compassionate and determined waterbender from the Southern Water Tribe who plays a crucial role in the fight against the Fire Nation alongside Aang and her friends.
		Sokka	Sokka is a witty and resourceful warrior from the Southern Water Tribe known for his boomerang skills, inventive mind, and comedic personality.
		Zuko	Zuko is the conflicted and determined exiled prince of the Fire Nation whose journey is defined by his quest for honor and self-discovery.
	minor	Iroh	Iroh is a wise and compassionate retired general of the Fire Nation, known for his love of tea, profound spiritual insight, and unwavering support for his nephew Zuko.
		Zhao	Zhao is an ambitious and ruthless Fire Nation admiral whose relentless pursuit of power and glory makes him a formidable adversary to Aang and his friends.
		Jet	Jet is a charismatic and fiercely determined teenage freedom fighter known for leading a group of rebels against the Fire Nation with a morally ambiguous approach to justice.
		Yue	Yue is the gentle and compassionate princess of the Northern Water Tribe whose destiny becomes intertwined with the fate of the Moon Spirit.
		Suki	Suki is the skilled and courageous leader of the Kyoshi Warriors known for her exceptional combat abilities and unwavering sense of justice.
		Appa	Appa is Aang’s loyal flying bison and steadfast companion known for his gentle nature, immense strength, and ability to soar through the skies.
		Pakku	Pakku is a master Waterbender from the Northern Water Tribe known for his strict teaching style and deep sense of tradition.

Table 9: Information of characters in our experiments (2/2).



Profile

Tyrion's mother, Joanna Lannister, passed away during his birth, and both Tywin and Cersei unjustly hold him responsible for this tragedy.

Tyrion, often ridiculed for his stature and referred to by derogatory names like "Imp" and "Halfman," attempts to embrace these labels to maintain his self-esteem. If a child with dwarfism like Tyrion had been born into a common family, they might have been abandoned, but his status as a Lannister provided him with considerable wealth and influence.

At the age of sixteen, Tyrion intervened in an assault on a wheelwright's daughter. While Jaime dealt with the attackers, Tyrion assisted the girl, named Tysha, in recovering. They eventually fell in love and secretly married. When Lord Tywin discovered the situation, he arranged for Tysha, who was actually a prostitute, to be with all of his guards while Tyrion was forced to watch. Jaime later admitted that he had set up the incident to help Tyrion lose his virginity.

While his two siblings held powerful positions, Tyrion was assigned the responsibility of managing the sewers at Casterly Rock when he came of age.

Tyrion often stayed at the court in King's Landing to distance himself from his father. He had a good relationship with Jaime, as well as his niece Myrcella and nephew Tommen, but his interactions with Cersei and Joffrey were more strained. Tyrion is dedicated to the well-being of his family and their influence on the throne, especially after Cersei's marriage to King Robert Baratheon.

Tyrion is a remarkably intelligent individual who realized early on that his mind was his greatest asset. As a member of House Lannister, he was expected to uphold his family's noble reputation, despite not having the physical traits of his older siblings. To compensate, he honed his intellect, often using it to outsmart those who underestimate him or to navigate difficult situations. He is also not averse to leveraging his family name and wealth to strengthen his position. Tyrion has demonstrated a strong strategic mind, particularly in his studies on siege defense and his role in the defense of King's Landing. He takes pride in his ability to understand people as well as he understands books, often outmaneuvering not only Joffrey and Cersei but also seasoned players like Littlefinger and Pycelle.

Tyrion has endured harsh treatment from others throughout his life, yet he has developed resilience and can easily dismiss jokes about his height, except when they come from his father. He advised Jon Snow on handling his status as a bastard, reflecting Tyrion's own philosophy: embrace your challenges and wear them like armor so they can't be used against you. As someone born a dwarf and often ridiculed, Tyrion has a soft spot for fellow outcasts and has shown them kindness, such as designing a saddle for Bran Stark so he could ride despite his disability. Tyrion is also deeply upset when the helpless are harmed. He was particularly angered by Janos Slynt's actions against a defenseless baby girl, leading to Slynt's exile to the Wall. Tyrion is loyal to those who treat him with love and respect and possesses a sense of honor and dignity that his family often lacks. He was disgusted by events like the Red Wedding, considering them crimes, although he acknowledges that some level of deceit is part of war. One of his most intriguing friendships is with Varys, the Master of Whisperers. They connect on several levels: both are marginalized due to physical conditions, both are brilliant and share a dry sense of humor, both are compassionate yet capable of ruthlessness, and both enjoy wine. Initially cautious of each other due to their reputations, Tyrion and Varys have developed trust through their shared experiences.

Tyrion has his flaws. He often indulges excessively in food, drink, and other pleasures as a way to cope with the harsh treatment he has endured throughout his life. When he feels stressed, upset, or uncomfortable, he tends to reach for the nearest drink. Tyrion can also be ruthless if he believes he has been wronged, such as when he punished Grand Maester Pycelle for betraying him. Due to the way his father and sister have treated him, Tyrion longs for love and acceptance and feels deeply hurt when others deny him this.

Tyrion has intricate relationships with his family, many of whom harbor resentment towards him and barely conceal their disdain. His older sister, Cersei, holds him responsible for their mother's death and has disliked him since birth. His father, Tywin, shares Cersei's resentment, as he lost his beloved wife during Tyrion's birth. Although Tywin acknowledges Tyrion's intelligence, it doesn't diminish his bias against his youngest son. Tyrion, in turn, both fears and admires his father but resents his rigidity and lack of affection for his children. Tywin also views Tyrion's physical disability as an embarrassment, interpreting it as a divine judgment meant to humiliate him.

Tyrion's older brother, Jaime, was the only family member who maintained a positive relationship with him. Jaime admired Tyrion for his intelligence and resilience despite his small stature, often defending him against their family's harsh treatment. Tyrion, in turn, held a deep respect for Jaime, even choosing him as his champion during a trial by combat at the Eyrie. Their bond was strong, with Tyrion expressing gratitude to Jaime for being the only Lannister to treat him like a brother and helping him through his childhood.

Tyrion had a strained and antagonistic relationship with his sister, Cersei, but he shared a strong bond with her children, Myrcella and Tommen. Myrcella showed genuine care and affection for Tyrion, expressing her happiness upon his safe return to King's Landing. Tyrion, in turn, cared deeply for them, despite his issues with their mother. His affection was evident when he became furious upon learning of Myrcella's death and confronted Ellaria Sand about her involvement.

After experiencing significant personal losses, Tyrion becomes a deeply troubled individual, losing his desire to live. The trauma of losing his loved one and his previous status and wealth leaves him more cynical and pessimistic. He initially chooses to drown his sorrows in alcohol rather than assist Varys in supporting Daenerys Targaryen's claim to the Iron Throne. Tyrion adopts a negative and fatalistic view of life, becoming indifferent to matters he once held in high regard, such as the Great Houses and the pursuit of power. Even when he meets Daenerys, he expresses a willingness to die. However, as he becomes involved in managing the turmoil in Meereen and advising Daenerys against becoming ruthless, he gradually regains his will to live. He finds renewed purpose and belief in Daenerys. Although his sense of humor becomes more subdued and cynical, this change may stem from his need to rely on his intellect rather than his family name for influence and protection.

Tyrion is highly skilled in political maneuvering, administration, and diplomacy, though his abilities as a military strategist and tactician are more varied. Like his father, Tywin, Tyrion is not particularly adept on the battlefield, but this is largely due to his reluctance to cause significant loss of life. He successfully orchestrated an ambush at the Battle of the Blackwater using a wildfire trap and led his troops in a sortie. However, this was not enough to secure victory without assistance from Tywin and the Tyrells. The heavy casualties from the wildfire deeply affected him, making him hesitant to use such destructive tactics again. This cautious approach influenced his strategy in aiding Daenerys against Cersei, resulting in several unsuccessful battle plans, except for the capture of Casterly Rock. Consequently, Daenerys grew impatient with his strategies and decided to take control herself. Nonetheless, Tyrion played a crucial role in forming and strengthening the alliance between Daenerys and Jon Snow, the King in the North.

Tyrion has a pacifist nature at heart. This is clear from the way he often advised Daenerys to avoid using violence to solve problems, aiming to ensure she doesn't follow in her father's footsteps. He consistently prefers peaceful solutions, turning to violence only as a last resort.

Scene

All of Winterfell gathers in the courtyard as the royal party approaches. Arya is late and annoys Sansa by not being silent and respectful when they arrive. King Robert rides in with his knights, while Queen Cersei is in a large wheelhouse. Lord Eddard presents his household to King Robert. Eddard and Robert greet each other warmly, and then Robert demands to see the crypts and pay his respects to Eddard's deceased sister, Lyanna, much to Queen Cersei's impatience. After Robert and Eddard depart, Cersei sends Jaime to find their younger brother Tyrion, who is a dwarf. "The Imp" has found his way to the brothel in the town outside Winterfell, where

What action does Tyrion take when he is in the town outside Winterfell?

he enjoys the attentions of a prostitute named Ros.

Scene

Tyrion has dinner with Lord Janos Slynt and questions him about the massacre of the royal bastards. Janos freely admits his involvement but will not reveal who ordered the attack. Tyrion has Janos arrested and exiled to the Night's Watch and installs Bronn as Commander of the City Watch in his place. Cersei confronts Tyrion about his actions.

How does Tyrion respond to Cersei's confrontation about his decision to arrest Janos Slynt and appoint Bronn as Commander of the City Watch?

He warns that she is losing the support of the people but she claims that she does not care.

Figure 19: An example of a profile and testing cases used in our experiments. (Tyrion Lannister)

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes]

   Justification: We clearly show the claims in the abstract and introduction, which are explored and verified in experiments and analyses.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitation in scope as a pioneering work, leaving multiple directions for future works to explore.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include all details: models, prompts, hyperparameter setups in the main content and the appendix.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release code and data when the paper is published, currently, the link is hidden for anonymity. The data and tools used to build our benchmark are also open to the public.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include all hyperparameters and other setups for the reproduction of our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our method's improvement over baselines is significant, and we omit the statement because the gap is relatively large to save lines.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide such information in efficiency analysis.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conduct experiments in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: As a role-playing method, we discuss the social impact of role-playing LLMs in the introduction and related work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks because it completely uses open data and models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have well cited the sources of all tools and datasets used in the experiments.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We detail our new assets in the appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The human evaluation guideline is clear as there are references provided to remove background knowledge requirement.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The experiment is approved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We clearly declare the roles of LLMs in our experiments.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.