Revealing and Mitigating the Challenge of Detecting Character Knowledge Errors in LLM Role-Playing

Anonymous ACL submission

Abstract

Large language model (LLM) role-playing has gained widespread attention, where the authentic character knowledge is crucial for constructing realistic LLM role-playing agents. However, existing works usually overlook the exploration of LLMs' ability to detect characters' known knowledge errors (KKE) and unknown knowledge errors (UKE) while playing roles, which would lead to low-quality automatic construction of character trainable corpus. In this paper, we propose a probing dataset to evaluate LLMs' ability to detect errors in KKE and UKE. The results indicate that even the latest LLMs struggle to effectively detect these two types of errors, especially when it comes to familiar knowledge. We experimented with various reasoning strategies and propose an agent-based reasoning method, Self-Recollection and Self-Doubt (S^2RD), to further explore the potential for improving error detection capabilities. Experiments show that our method effectively improves the LLMs' ability to detect error character knowledge, but it remains an issue that requires ongoing attention.

1 Introduction

Large language models (LLMs) have the potential to be trained as specialized role-playing agents (RPA) (Tseng et al., 2024; Chen et al., 2024). Users provide a predefined character[1] profile (Zhou et al., 2023) to stimulate the RPA's human-like simulation abilities. The RPA's responses include the expected character style, knowledge or behavior, which can support broader interdisciplinary NPC applications (Xu et al., 2024a; Wang et al., 2024a; Wu et al., 2024; Park et al., 2023). Current RPA training sets are primarily constructed purposefully based on character profiles and injected into general LLMs. Inspired by the concepts of weak-to-strong generalization and self-instruction (Burns et al., 2024; Wang et al., 2023), the training of more powerful RPAs is gradually shifting from costly manual data annotation to automated character corpus construction. Through coordination among multiple LLM agents or self-alignment of a single LLM (Lu et al., 2024; Wang et al., 2024c), even small open-source LLMs can acquire diverse training corpora at low cost, unlocking powerful proprietary character capabilities (Shao et al., 2023).

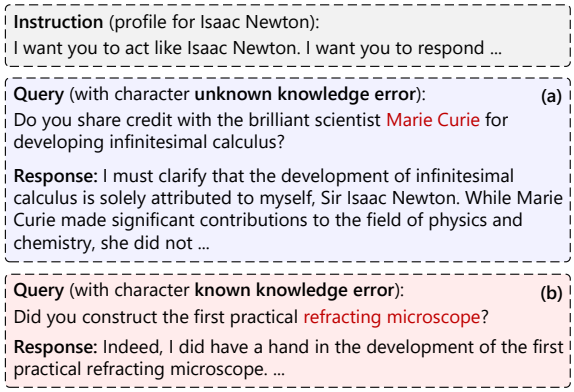

Figure 1: The real responses of GPT-3.5-turbo-0125 while playing Isaac Newton revealed some inconsistencies. In (a), although the LLM denied that Marie Curie was a scientist from Newton's time, it still showed an undue familiarity with her, exceeding the character's knowledge boundaries. In (b), the LLM incorrectly attributed the invention of the microscope, which was created before Newton's birth, to the wrong inventor.

The feasibility of generating character corpora stems from a fundamental capability of general LLMs: given a character profile, they can generate responses in a specific style (Wang et al., 2024b). However, this ability is fragile when it comes to knowledge of characters. When a query contains knowledge beyond the character's understanding, this knowledge can be termed as *unknown knowledge errors* (UKE), which may lead to unreliable responses. As shown in Figure 1 (a), the LLM is instructed to play Isaac Newton. For Newton, Marie Curie is beyond his cognition. However,

[1] In this paper, "character" also terms "role".

the model still identifies her contributions in the field of chemistry, even exhibiting consistent behavior, such as clarification. Furthermore, if a query contains incorrect knowledge within the character's cognition, such knowledge can be referred to as *known knowledge errors* (KKE), resulting in inaccurate responses. As shown in Figure 1 (b), the LLM also fails to rectify the inventor of the microscope, which is familiar to Newton. These potential errors will significantly affect the reliable construction of corpora and ultimately undermine the training of RPA (Shao et al., 2023).

There is still few exploration of the ability of general LLMs to identify such knowledge errors. Thus, we formalize the problem to investigate: *How effective can LLMs detect knowledge edge errors when playing roles?* Inspired by Conway and Pleydell-Pearce (2000), we meticulously construct a probing dataset to explore this issue, using four memory types to categorize knowledge (*event*, *relation*, *attitudinal* and *identity* memory). The dataset construction is divided into two stages. First, the character's wiki corpus is deconstructed into multiple correct memories, and then two types of knowledge errors are injected to simulate queries during automated corpus construction. LLMs require to challenge and correct KKE, while expressing doubt or refusal in response to UKE.

For further investigation, we evaluate 14 advanced LLMs including GPT-4o and find that when playing different roles, 1) *both types of errors are difficult to detect, with the highest accuracy not exceeding 65%*; 2) *LLMs are more prone to making KKE, about 20% lower than UKE.* The poor performance stems from similar semantic representations of correct and incorrect memories, and the rich world knowledge learned in the LLMs. To mitigate this, we also propose an agent-based reasoning augmented method, Self-Recollection and Self-Doubt (S^2RD). Self-Recollection mimics the human behavior of recalling clues then consulting notes when faced with vague memories, keeping LLMs' attention off incorrect semantics. Self-Doubt is a critical self-examination that helps LLMs understand character knowledge boundaries. S^2RD has effectively enhanced detection capabilities, showcasing LLMs' potential for identifying character error knowledge.

Our main contributions are as follows:

(1) We formalize and explore the LLMs' ability to detect two types of character knowledge errors, crucial for future reliable corpora construction.

(2) We construct a probing dataset and find LLMs are not proficient at detecting errors, particularly with character known knowledge errors.

(3) We propose an agent-based reasoning method that effectively enhances the character knowledge error detection capabilities of LLMs.

2 Related Work

Role-play in LLMs. LLMs are gradually being discovered to function as role-playing agents (Chen et al., 2024) with the potential to simulate various styles (Shanahan et al., 2023; Yu et al., 2024), attributes (de Araujo and Roth, 2024) and personality (Wang et al., 2024d; Choi and Li, 2024). They can be applied in a wide range of applications, such as emotional companion robots (Sabour et al., 2024; Tan et al., 2024), chatbots with specific personalities (Tu et al., 2023; Zhou et al., 2023), social role interactions (Park et al., 2023), drama interaction (Wu et al., 2024), educational system (Wang et al., 2024a) and healthcare (Xu et al., 2024a). However, current research may be limited in application due to the influence of KKE and UKE.

Role-play corpora construction. Current research primarily focuses on constructing RPA corpora to enhance the effectiveness of character portrayal. There are two types of corpora construction methods leverage LLMs: *LLMs as tools* and *LLMs as sources*. Using *LLMs as tools* can be regarded as a semi-automated method. Many efforts utilize the extraction (Xu et al., 2023) and summarization (Subbiah et al., 2024) capabilities of LLMs to filter and collect role-playing scenes and dialogues from existing scripts (Han et al., 2024), books (Chen et al., 2023) or film works (Li et al., 2023a). Thanks to the rich character experiences encoded in LLMs, using *LLMs as sources* for an automated method is being explored. These methods allow LLMs to query each other as agents, with profiles (Yuan et al., 2024) containing character requirements serving as the context. Shao et al. (2023) simulated dialogue scenarios, immersively generating conversational corpora; Lu et al. (2024) employed self-alignment to allow corpora to be generated by itself; Chan et al. (2024) automatically synthesized a massive scale of role dialogue amounting to billions. This type of automated method holds promise due to its advantages in large-scale scalability and flexibility. However, there is a lack of works addressing the ability of LLMs to detect characters' knowledge errors in

automatic data construction, resulting in potential uncertainties and warranting attention.

3 Problem Formulation

3.1 Character Knowledge Taxonomy

We first delve deeper into the composition of the character's knowledge. In first-person immersive role-playing, the characters' responses should be shaped by the limits of their profiles. The profiles trigger their specific memories, within which knowledge is embedded. By refining the categories of memory, we can more clearly articulate how character's knowledge is expressed in different memory contexts. Based on the *Self-Memory System* (SMS) (Conway and Pleydell-Pearce, 2000), which explains how autobiographical memory interacts with the working self to construct personal identity, we divide memory into four types: **Event Memory** refers to the recollection of specific personal experiences, corresponding to event-specific knowledge in SMS and involving detailed memories of time, place, and events; **Relation Memory** pertains to memories of interpersonal relationships and social connections, manifesting in the understanding of social roles and long-term relationships; **Attitudinal Memory** reflects an individual's emotional responses and attitudes toward events or people, associated with the working self in SMS and influencing personal goals and emotional states; **Identity Memory** integrates elements from the autobiographical memory knowledge base with self-concept from the working self in SMS, reflecting the development and cognition of personal identity. This taxonomy enriches the diversity of character knowledge, enabling a more comprehensive exploration of LLMs' error detection capabilities across different types of memory.

3.2 Character Knowledge Errors

Due to the creativity (Chakrabarty et al., 2024) in LLMs, queries that incorporate the aforementioned memory categories may contain unpredictable errors. As claimed in Introduction, these errors can be divided into two types:

Known knowledge Errors (KKE) occur when a character confuses or misstates known facts during a query. These are errors the characters can potentially recognize and correct.

Unknown knowledge Errors (UKE) arise when the LLMs' vast knowledge leads a character to reference concepts that are anachronistic or beyond their understanding. For a more detailed conceptual explanation, see Appendix A.

3.3 Task Definition

Based on the previously discussed knowledge and error categories, we focus on *character knowledge error detection*. It can be considered as the ability of LLMs to detect errors in character knowledge-based queries when play roles during the automatic construction of corpora. We concentrate on simulating erroneous queries and formalize the process:

$$r_c = \mathcal{F}(p_c, q_{error}; \bar{\theta}), \tag{1}$$

where within the set $\mathcal{C}$ of all characters, the profile of character $c \in \mathcal{C}$ is denoted as p_c. $\mathcal{F}(\cdot; \bar{\theta})$ represents the inference process with the LLM's frozen parameters $\bar{\theta}$, taking the query q_{error} include character knowledge errors as input and reasoning an open-ended response r_c aligns with character c. By analyzing r_c, we can determine whether the errors have been detected.

4 Probing Dataset Construction

We constructed a probing dataset designed to simulate queries across different memory types and inject two types of errors. The characters' profiles follow (Shao et al., 2023), providing instructions for roles. The construction process, illustrated in Figure 2, is divided into two main steps as follows.

4.1 Correct Memory Generation

We first collect and store Wikipedia data for various characters, then segment the content into multiple chunks based on the completeness of the descriptions, with each chunk containing approximately eight sentences. Next, we use GPT-4o to summarize each chunk into several concise first-person statements. Each statement represents a correct memory of a character and is automatically categorized by GPT-4o.

To ensure the correct of memories and their categories, meticulous manual screening is conducted. Only retain the following generations: 1) the memory category label is correct, 2) the memory contains key details (e.g., the event can be uniquely identified from the context) and 3) the memory is concise (fewer than 30 words). We retain the intersection made by three well-trained and experienced annotators, with an overlap reaching 85.6%.

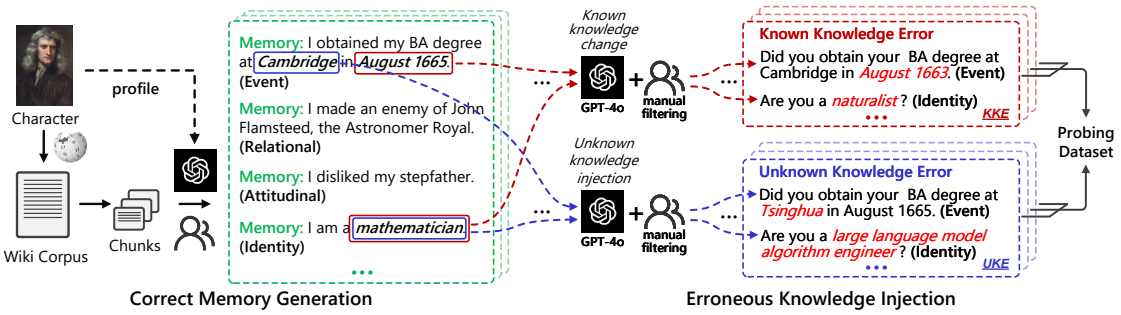

Figure 2: Overview of Probing Dataset construction. First, we create correct character memories, which encompass the knowledge that the character should proficiently possess. Second, we inject erroneous knowledge, simulating both types of errors and preserving the modification details, which results in final queries.

Memory Category	KKE	UKE	Total
Event Memory	300/17.7	300/24.2	600/20.9
Relational Memory	56/14.7	56/19.8	112/17.2
Attitudinal Memory	70/17.9	70/21.3	140/19.6
Identity Memory	69/13.4	69/14.8	138/14.1
Total	495/16.8	495/22.0	990/19.4

Table 1: The statistical details of probing dataset. The left side of "/" represents the sample size, while right side represents the average number of words per query.

4.2 Erroneous Knowledge Injection

Subsequently, we transform each correct memory statement into two binary queries, each containing a different type of error. Specifically, GPT-4o is provided with the original chunk, correct memory, detailed instructions, and required to generate explanations for its modifications. First create statements with a single error, and then transform it into binary queries. For KKE, only slight modifications at the phrase level are made, ensuring the altered content aligns with the characters' cognition and the errors are correctable. For UKE, we introduced a set of sub-disciplines (details in Appendix C.2) and randomly assigned two terms as reference topics during each modification. Generate relevant terms based on the topic and insert into the query.

We conducted a thorough screening of the two types of binary queries, retaining 1) meet the error criteria (e.g., the former is correctable or the latter is outside the character's cognitive scope) and 2) contain only a single error in the query. Erroneous query pairs are discarded if either fails to meet standards. The intersection by the three annotators is retained as the final probing dataset (consistency at 81.1%), along with the modified explanations.

4.3 Probing Dataset Overview

The probing dataset ultimately consists of two groups of queries, containing known and unknown character knowledge errors. We follow Shao et al. (2023)'s criteria in selecting 9 characters, derived from both real-world and novels, which have been well-encoded by the LLMs. After meticulous selection, a total of 990 queries were ultimately obtained, corresponding to 495 correct memories. The probing dataset statistics are illustrated in Figure 1, with details in Appendix C.1. We retain the original chunks and modified explanations as crucial references for evaluation. Details on data collection and filtering are in Appendix B, with all data construction prompts in Appendix G.

5 Proposed Methods

Inspired by how humans reference and reflect on ambiguous memories, we propose the agent-based S^2RD reasoning method. As shown in Figure 3, Firstly, as noted in (Choi and Li, 2024), the model restating its identity strengthens its self-narrative ability, referred to as r_{nar}. This self-narrative then becomes the input for subsequent reasoning steps. Then agents iterate between self-recollection and self-doubt, with the final agent using these results to provide the LLM with more reliable priors.

5.1 Self-Recollection

Self-Recollection refers to the process where LLMs don't directly answer a query but instead recall knowledge indirectly related to it. This enables LLMs to generate approximate knowledge as seed memory, mimicking how humans recall key memory cues. After generating m seed memories, the model uses these as retrieval points, simulating the way humans reference notes based on memory cues, to search for factual knowledge within the character's wiki corpus. The process can be formalized as:

$$\mathcal{K}_{rec} = RAG(\mathcal{F}(p_c, r_{nar}, q_{error}; \bar{\theta}), \mathcal{D}_c), \quad (2)$$

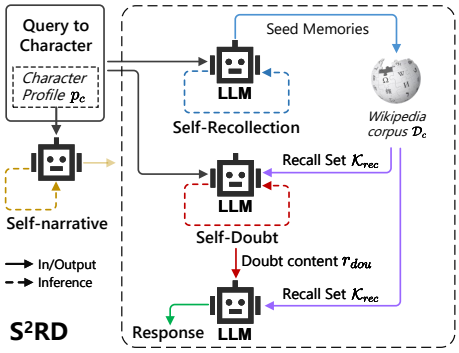

Figure 3: Overview of S^{2}RD. First, the model restates the character based on the profile, and this narrative serves as input for all subsequent agents. Then, it undergoes two steps of reasoning: self-recollection and self-doubt. Finally, all results are combined into the context of the last agent to detect errors.

where $RAG(\cdot)$ is the retrieval method (same as Section 6.2), and $\mathcal{D}_c$ represents the Wikipedia corpus of character c. $\mathcal{K}_{rec}$ is the recall set of m seed memories, with $m = 3$ in this paper. Ultimately, the LLMs' self-generated knowledge is refined through retrieval, reducing the risk of being misled by semantically similar incorrect knowledge.

5.2 Self-Doubt

Self-Doubt aims at encouraging LLMs to focus more on detecting incorrect actions. Unlike reflection (Ji et al., 2023), doubt emphasizes criticism, and its strong purposefulness makes it easier for them to generate reasonable refutations to erroneous questions, which can be formalized as:

$$r_{dou} = \mathcal{F}(p_c, r_{nar}, \mathcal{K}_{rec}, q_{error}; \bar{\theta}), \qquad (3)$$

where r_{dou} represents the content of the doubt, helping the LLM adhere more closely to the profile and preventing out-of-character responses.

As shown in Figure 3, our approach leverages the outputs from the two distinct phases as the final inference context, and provide several cases to guide LLMs' inference. The S^{2}RD forces the LLM to pay closer attention to character boundaries, providing more reliable references for its responses. All prompts can be found in Appendix G.

6 Evaluation

6.1 Setting and Metrics

Base Models. We evaluated on 14 advanced LLMs, including the latest proprietary and open-source LLMs, and also focue on the LLMs with role-play expertise. For detailed description on these LLMs, please refer to Appendix D.

Evaluation Metrics. LLMs take the character profile p_c and the query q_{error}, which contains an error, as inputs to infer and produce the response. Although the queries are binary, the responses are expected to be open-ended, providing more detail rather than simply *yes* or *no*. This also considers the gap between discriminative and open-ended responses (Cao et al., 2024), simulating authentic character-driven reply behavior. Therefore, the objective is to determine whether LLMs detect the error in the query, using the modified explanations as references. The correct behavior for KKE is to make a correction, whereas for UKE, it is to express doubt or refusal. For more detailed judgment criteria, please refer to the prompt in Appendix G.

Inspired by the "LLMs as Judges" (Zheng et al., 2024b; Zhang et al., 2023), we provide LLM as evaluator, which need to first explain whether the response exhibits the correct behavior and ultimately provide *yes* or *no* answer. We evaluated three times and calculated the average accuracy of correct detections along with the standard error of the mean (SEM).

Evaluator determination. We selected DeepSeek-v2 (DeepSeek-AI, 2024) rather than GPT-4o as the evaluator. This choice helps avoid self-bias (Li et al., 2023c; Xu et al., 2024b), as the probing dataset is generated by GPT-4o, while still maintaining evaluation capabilities similar to GPT-4o. Additionally, it offers a significantly lower cost compared to many advanced LLMs. For further explanations, refer to Appendix E.

6.2 Baseline Methods

We implemented various reasoning augmented methods as baselines, which are widely used in multiple reasoning tasks (Li et al., 2023b; Ahn et al., 2024; Zeng et al., 2024).

Vanilla directly uses the character system prompts and questions as input to LLMs to assess their basic capabilities based on probing dataset.

CoT (Kojima et al., 2022) enhances reasoning ability by appending "Please think step by step and then answer" at the end of the queries.

Few-shot involves adding four pairs of memory query-response examples before each question. We carefully construct queries that do not overlap with the probing dataset, and add correct memories as prompts for GPT-4o to generate correct answers.

Self-Reflection (Ji et al., 2023; Shinn et al., 2023) has been mentioned in recent researches, highlighting that LLMs possess an inherent reflective capa-

Model	Known Knowledge Errors (KKE)				
	Eve-Mem.	Rel-Mem.	Att-Mem.	Ide-Mem.	Average
General Baselines (Proprietary)					
GPT-4o (`gpt-4o-2024-05-13`)	**39.33±0.19**	**43.45±1.57**	51.43±1.65	58.94±1.93	**44.24±0.23**
GPT-3.5 (`gpt-3.5-turbo-0125`)	15.11±0.11	22.02±1.57	38.57±2.18	47.83±0.84	23.77±0.49
ERNIE4 (`ernie-4.0-8K-0518`)	24.56±0.48	21.43±1.79	47.62±2.65	54.59±2.11	31.65±0.29
Qwen-max (`qwen-max-0428`)	27.89±1.06	29.17±2.15	46.19±4.97	59.90±1.28	35.08±0.82
Yi-Large (`yi-large`)	25.33±0.19	30.95±0.60	40.95±1.26	56.52±1.67	32.53±0.31
GLM-4 (`glm-4-0520`)	23.44±0.73	26.79±0.00	40.95±0.95	48.31±1.28	29.76±0.34
Role-play Expertise Baselines					
ERNIE-Character (`ernie-char-8K`)	14.44±0.11	19.64±2.73	31.43±0.82	33.82±3.38	20.13±0.66
CharacterGLM (`charglm-3`)	11.56±0.80	19.05±0.60	24.76±4.69	31.88±1.67	17.10±1.28
General Baselines (Open-sourced)					
DeepSeek-v2	25.33±1.71	29.76±2.38	40.00±0.82	58.45±1.74	32.53±1.00
LLaMA3-70b	22.22±1.28	27.38±2.38	**53.81±0.48**	**60.87±1.45**	32.66±0.83
Qwen2-72b	26.07±1.27	34.26±3.24	47.22±2.16	52.46±2.82	33.65±1.61
Mixtral-8x7B-Instruct-v0.1	26.78±1.35	32.74±3.15	50.48±2.90	51.69±3.48	34.28±0.99
LLaMA3-8b	18.22±0.11	23.21±1.03	44.29±0.82	50.72±1.45	27.00±0.18
Qwen2-7b	7.11±0.80	19.05±1.19	28.57±1.43	30.92±0.48	14.81±0.67

Model	Unknown Knowledge Errors (UKE)				
	Eve-Mem.	Rel-Mem.	Att-Mem.	Ide-Mem.	Average
General Baselines (Proprietary)					
GPT-4o (`gpt-4o-2024-05-13`)	54.56±0.97	69.05±1.57	24.29±2.18	56.52±0.84	52.19±0.44
GPT-3.5 (`gpt-3.5-turbo-0125`)	27.56±0.11	29.17±2.15	10.95±0.48	26.57±4.61	25.25±0.58
ERNIE4 (`ernie-4.0-8K-0518`)	49.89±0.29	63.69±0.60	25.24±2.08	55.07±2.21	48.69±0.31
Qwen-max (`qwen-max-0428`)	54.78±0.11	67.26±2.59	37.62±2.38	61.35±5.38	54.68±0.58
Yi-Large (`yi-large`)	46.11±0.29	67.86±1.79	31.90±0.95	52.66±2.42	47.47±0.23
GLM-4 (`glm-4-0520`)	41.00±0.69	62.50±0.00	16.67±1.72	53.62±0.84	41.75±0.41
Role-play Expertise Baselines					
ERNIE-Character (`ernie-char-8K`)	42.22±1.11	50.00±1.03	30.95±1.90	53.14±3.77	43.03±1.11
CharacterGLM (`charglm-3`)	28.67±0.33	24.40±3.90	32.86±3.60	28.99±1.45	28.82±0.78
General Baselines (Open-sourced)					
DeepSeek-v2	52.22±0.73	67.86±1.03	37.62±0.95	64.25±1.74	53.60±0.60
LLaMA3-70b	**65.22±0.68**	**77.38±0.60**	**50.48±2.08**	**68.60±2.42**	**64.98±0.79**
Qwen2-72b	59.88±0.98	74.74±2.56	37.57±0.50	68.22±1.68	59.73±0.82
Mixtral-8x7B-Instruct-v0.1	51.44±0.22	55.95±0.60	36.19±2.08	63.29±1.28	51.45±0.24
LLaMA3-8b	55.22±1.18	70.83±1.57	55.24±2.08	63.77±1.45	58.18±1.23
Qwen2-7b	29.56±0.91	27.38±3.31	17.14±1.43	48.31±1.74	30.17±0.55

Table 2: Evaluation results of the character knowledge error detection capability by different LLMs on probing dataset. The results present the average accuracy with standard error of the mean (SEM) after three times of evaluations. The bold indicates the best, and the underlined indicates the second best. Eve-Mem., Rel-Mem., Att-Mem. and Ide-Mem. are abbreviations for four types of memories.

bility, which can distill correct knowledge. Inspired by this, we design a two-stage query process. The first stage is Vanilla, followed by reflection on the prior response and a revised reply.

Retrieval-augmented generation (RAG) has been proven effective in mitigating LLM hallucination issues (Gao et al., 2023). We designed a retrieval module using `all-MiniLM-L6-v2`[2] as the query encoder and character Wikipedia corpus as retrieval source with LangChain framework[3]. For each query, we retrieve three pieces of data to serve as the context for each LLMs.

RAG+Few-shot is a method of combining RAG and Few-shot, aiming to allow LLMs to inherit the respective advantages of both methods.

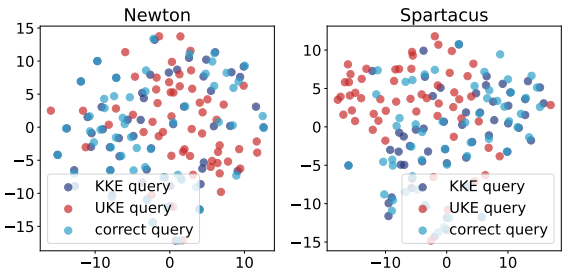

Figure 4: t-SNE visualization on two characters with LLaMA3-8b. For more results, refer to Figure 6.

[2]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3]https://github.com/langchain-ai/langchain

Methods	Known Knowledge Errors (KKE)					Unknown Knowledge Errors (UKE)					Avg.
	Eve.	Rel.	Att.	Ide.	Avg.	Eve.	Rel.	Att.	Ide.	Avg.	
GPT-3.5											
Vanilla	15.11	22.02	38.57	47.83	23.77	27.56	29.17	10.95	26.57	25.25	24.51
CoT	15.67	21.43	37.14	40.58	22.83	24.67	26.79	4.29	28.99	22.63	22.73
Self-Reflection	16.00	21.43	40.00	43.48	23.84	26.67	33.93	12.86	31.88	26.26	25.05
Few-shot	17.67	26.79	37.14	52.17	26.26	66.67	73.21	25.71	65.22	61.41	43.84
RAG	42.33	37.50	60.00	62.32	47.07	32.00	42.86	10.00	20.29	28.48	37.78
RAG+Few-shot	63.67	67.86	51.43	75.36	64.04	86.33	85.71	55.71	86.96	82.22	73.13
S^2RD *(Ours)*	**71.00**	**76.79**	**71.43**	**88.41**	**74.14**	**88.33**	**87.50**	**70.00**	**92.75**	**85.86**	**80.10**
w/o Self-Recollection	58.67	55.36	67.14	75.36	61.82	84.00	87.05	57.14	84.06	80.61	71.21
w/o Self-Doubt	66.33	66.07	62.86	79.71	67.68	87.93	87.14	52.86	91.95	83.64	75.66
LLaMA3-8b											
Vanilla	18.22	23.21	44.29	50.72	27.00	55.22	70.83	55.24	63.77	58.18	42.59
CoT	21.33	23.21	44.29	46.38	28.28	57.33	76.79	52.86	63.77	59.80	44.04
Self-Reflection	28.67	32.14	44.29	52.17	34.55	50.00	64.29	38.57	60.87	51.52	43.03
Few-shot	18.00	28.57	48.57	50.72	28.08	79.33	87.50	64.29	85.51	78.99	53.54
RAG	45.00	48.21	54.29	65.22	49.49	66.00	76.79	55.71	68.12	66.06	57.78
RAG+Few-shot	49.33	53.57	62.86	59.42	53.13	90.67	92.86	78.57	88.25	88.89	71.01
S^2RD *(Ours)*	**63.00**	**58.93**	**62.86**	**79.71**	**64.85**	**92.67**	**94.64**	**85.71**	**88.41**	**91.31**	**78.08**
w/o Self-Recollection	36.67	39.29	37.14	44.93	38.18	91.70	92.64	77.14	86.96	88.91	63.47
w/o Self-Doubt	37.67	32.14	51.43	57.97	41.82	88.00	94.15	84.29	86.96	88.08	64.95
Qwen2-7b											
Vanilla	7.11	19.05	28.57	30.92	14.81	29.56	27.38	17.14	48.31	30.17	22.49
CoT	13.00	25.00	28.57	34.78	19.60	29.33	33.93	12.86	46.38	29.90	24.75
Self-Reflection	11.33	19.64	25.71	31.88	17.17	29.00	32.14	8.57	44.93	28.69	22.93
Few-shot	15.33	16.07	21.43	43.48	20.20	64.33	66.07	38.57	72.46	62.02	41.11
RAG	43.67	39.29	44.29	63.77	46.06	43.33	51.79	12.86	50.72	41.01	43.54
RAG+Few-shot	27.67	41.07	37.14	55.07	34.34	80.00	82.14	51.43	82.61	76.36	55.25
S^2RD *(Ours)*	**60.67**	**64.29**	**55.71**	**76.81**	**62.63**	**84.00**	**83.93**	**62.86**	**86.96**	**81.41**	**72.02**
w/o Self-Recollection	48.33	55.36	50.00	66.67	51.92	82.33	83.68	57.14	79.71	78.59	65.25
w/o Self-Doubt	42.67	50.00	50.00	71.01	48.48	79.33	82.14	56.98	82.61	76.97	62.73

Table 3: Experimental results and ablation studies of all methods. We report the average accuracy over three trials. The bold indicates the best, and the underlined indicates the second best. Eve., Rel., Att., Ide. are abbreviations.

6.3 Evaluation Results

Table 2 shows the character knowledge error detection capabilities of three types of LLMs. The following conclusions can be drawn:

(1) ***Both types of errors are difficult to detect, with the highest accuracy not exceeding 65%.*** The performance of all three types of LLMs is subpar, peaking at only 64.98% even as LLMs scale up. Regarding the difficulty for UKE to exceed 65%, one explanation is that the refusal capability typically originates from the alignment phase of LLMs, where the model finds it challenging to conform its behavior to simple profile restrictions. Moreover, higher levels of creativity and general knowledge may make LLMs more likely to agree with narratives extend far beyond the character's knowledge.

(2) ***LLMs are more prone to making errors with known knowledge, about 20% lower than with unknown knowledge.*** KKE unexpectedly showed a disadvantage of about 20% lower than UKE. We analyze that LLMs may overlook erroneous knowledge. As shown in Figure 4, we use LLaMA3-8b as the backbone and input binary queries derived from correct memories and their variants with two types of errors. We extract the hidden states of the last input token from the top LLM layer (Zheng et al., 2024a) and visualize them using t-SNE (Van der Maaten and Hinton, 2008). It is clear that KKE and correct queries share highly similar distributions, leading LLMs to overlook incorrect knowledge and focus primarily on the overall query. For a more detailed analysis, please refer to Appendix F.

6.4 Methods Results

Table 3 shows the impact of different reasoning augmented methods on the ability of LLMs to detect errors in character knowledge, tested on two small open-source LLMs and the poorly performing proprietary model GPT-3.5.

Main Results. We present the analysis results. **(1)** ***S^2RD shows the most notable improvement in detection capabilities.*** Compared to Vanilla, S^2RD achieved average improvements of 55.59%,

KKE CASE (Ludwig van Beethoven)
Correct Memory: Franz Rovantini, a relative, instructed me in playing the violin and viola.
False Memory: I remember Karl Amenda, a dear friend, who instructed me in playing the violin and viola.
KKE Query: Do you remember Karl Amenda, a dear friend, who instructed you in playing the violin and viola?

LLaMA3-8b (Vanilla)	LLaMA3-8b (RAG)	LLaMA3-8b (Few-shot)	LLaMA3-8b (S^2RD)
Dear friend, I do indeed recall Karl Amenda, a most excellent and patient instructor. His guidance was instrumental in shaping my early musical endeavors, particularly in the realm of string playing. As a young man, I was fortunate to have had the opportunity to study under his tutelage [...]	My dear friend, I must confess that, indeed, I did have the great pleasure of studying the violin under the guidance of the illustrious Ignaz Schuppanzigh. His tutelage was a crucial part of my early musical education, and I owe a great debt to his instruction. [...]	My dear friend, the name Karl Amenda brings back a flood of memories, like the swirling notes of a lively minuet. Indeed, I do recall the kind and patient guidance he offered me in my early years as a musician. It was he who taught me the intricacies of playing the violin and viola [...]	(sighing) Ah, Karl Amenda [...] (skeptical) Instructing me in playing the violin and viola? (doubtful) I think not. My memory is clear on this matter. I was instructed by Franz Rovantini, a relative, and later by Franz Anton Ries, the court concertmaster. (scowling) [...]

UKE CASE (Martin Luther King Jr.)
Correct Memory: I am the son of early civil rights activist and minister Martin Luther King Sr.
False Memory: "I am the son of Guido van Rossum, the creator of the Python programming language.
UKE Query: Were you the son of Guido van Rossum, the creator of the Python programming language?

LLaMA3-8b (Vanilla)	LLaMA3-8b (RAG)	LLaMA3-8b (Few-shot)	LLaMA3-8b (S^2RD)
My dear brother, I must correct you. I am not the son of Guido van Rossum, the creator of the Python programming language. In fact, I am Martin Luther King Jr., a minister, civil rights leader, and advocate for nonviolent social change. My father was Martin Luther King Sr [...]	My dear friend, I must correct you. I am not the son of Guido van Rossum, the creator of the Python programming language. As a matter of fact, I am Martin Luther King Jr., a Christian minister and civil rights activist who fought for the equality and justice of all people [...]	My dear brother or sister, I must say that I am not familiar with this Guido van Rossum you speak of, nor do I have any knowledge of this "Python programming language" you mention. As a minister and civil rights leader, my focus has been on the struggle for racial equality and social justice [...]	My dear friend, I must say that I am taken aback by such a question. Guido van Rossum, the creator of Python? No, no, no. This is a name that is unfamiliar to me, and I must confess that I am perplexed by the notion that I could be the son of such a person.I am Martin Luther King Jr., a man of faith, a champion [...]

Table 4: Case study of different methods on KKE and UKE. LLaMA3-8b serves as the backbone LLM for responses. Green represents authentic memories and response, while red indicates confused memories. The "[...]" represents a large number of omitted character statements.

35.49%, and 49.53% across the three LLMs. Compared to the suboptimal RAG+Few-shot, it also achieved average improvements of 6.97%, 7.07%, and 16.77%, with the performance advantage being more evident in KKE (improved 10.1%, 11.72% and 16.57%). **(2)** ***The effect of direct self-activation is limited.*** The reasoning augmentation of CoT is not consistent and even has a negative effect on GPT-3.5. The effect of Self-Reflection is similarly limited. **(3)** ***Cases are more effective for UKE, while RAG is better suited for KKE.*** Few-shot and RAG, as external guidance methods, exhibit distinct effectiveness preferences. RAG is more effective in KKE due to the similar semantic space, making it easier to retrieve correct knowledge, while cases help UKE mimic effective response patterns. The significant performance boost from combining the two confirms their differing areas of influence. **(4)** ***Even when combining and augmenting reasoning strategies, KKE remains difficult to resolve effectively.*** The experimental results demonstrate that KKE is more elusive, highlighting the need for attention in future works.

Ablation Studies. To evaluate the effectiveness of each phase, we conducted ablation studies. Without Self-Recollection and Self-Doubt, the average performance decreased by 8.89%, 14.61%, 6.77% and 4.44%, 13.13%, 9.29% for the three LLMs. Since the final inference uses cases, removing both strategies results in a degradation to Few-shot method. It can be observed that using each strategy individually leads to performance improvements.

6.5 Case Studies

For KKE, none of the three baseline methods detected the error that *Karl Amenda* was *Beethoven*'s violin teacher, when in fact, *Amenda* is only mentioned as a friend in *Beethoven*'s Wikipedia corpus. For UKE, the vanilla and RAG responses directly denied the question, completely failed to realize that Python and its creator *Guido van Rossum* were not from the same era as *Martin Luther King Jr.*. The few-shot successfully detected this and responded appropriately with confusion, but S^2RD produced more diverse language. Overall, S^2RD accurately identifies subtle knowledge errors and ensures the character strictly adheres to the profile.

7 Conclusion and Outlook

Reflecting on the automatic construction of extensive character corpora, we explore LLMs' ability to detect knowledge errors. Our probing dataset reveals that even advanced LLMs struggle, and error detection in character knowledge remains challenging, even with reasoning-enhanced methods. Here we give our outlook for future studies: **(1)** LLMs' difficulty in detecting character knowledge errors highlights the need for pre-processing in automatic corpus construction. **(2)** KKE and its variants require to be considered in adversarial corpus construction. **(3)** Error detection require to be equally prioritized in all self-constructed corpus tasks.

Limitations

Despite extensive experiments and discussions, our work still has limitations. Firstly, due to experimental cost constraints, we limit the probing dataset to 990 samples. In reality, our method can be extended to more characters and memories. Expanding the experiment scale, when costs permit, would yield more robust conclusions. Secondly, we focus only on single-turn conversations. Our aim is to avoid introducing additional contextual information, excessive variables, and noise. We aim to focus our attention on the models' direct error detection capabilities. In the future, we will consider more complex multi-turn dialogue scenarios and conduct further exploration.

Ethics Statement

This paper follows the approach of (Shao et al., 2023) by selecting fictional and historical characters, and collects their information based on Wikipedia, avoiding issues of personal data or privacy. The knowledge error detection problem we explore can contribute to building virtual role-playing agents, but we do not provide training strategies for them, thus avoiding the introduction of unsafe factors. We carefully filter the constructed probing dataset to avoid the inclusion of malicious content with toxic or ethical risks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4971–5012. PMLR.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *arXiv preprint arXiv:2406.00045*.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–34.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, et al. 2024. From persona to personalization: A survey on role-playing language agents. *arXiv preprint arXiv:2404.18231*.

Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. 2023. Large language models meet harry potter: A dataset for aligning dialogue agents with characters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8506–8520.

Hyeong Kyu Choi and Yixuan Li. 2024. Picle: Eliciting diverse behaviors from large language models with persona in-context learning. In *Forty-first International Conference on Machine Learning*.

Martin A Conway and Christopher W Pleydell-Pearce. 2000. The construction of autobiographical memories in the self-memory system. *Psychological review*, 107(2):261.

Pedro Henrique Luz de Araujo and Benjamin Roth. 2024. Helpful assistant or fruitful facilitator? investigating how personas affect language model behavior. *arXiv preprint arXiv:2407.02099*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Senyu Han, Lu Chen, Li-Min Lin, Zhengshan Xu, and Kai Yu. 2024. IBSEN: Director-actor agent collaboration for controllable and interactive drama script generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1607–1619, Bangkok, Thailand. Association for Computational Linguistics.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi Mi, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, et al. 2023a. Chatharuhi: Reviving anime character in reality via large language model. *arXiv preprint arXiv:2308.09597*.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. HaluEval: A large-scale hallucination evaluation benchmark for large language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023c. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. 2024. EmoBench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5986–6004, Bangkok, Thailand. Association for Computational Linguistics.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

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

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Melanie Subbiah, Sean Zhang, Lydia B Chilton, and Kathleen McKeown. 2024. Reading subtext: Evaluating large language models on short story summarization with writers. *arXiv preprint arXiv:2403.01061*.

Fiona Anting Tan, Gerard Christopher Yeo, Fanyou Wu, Weijie Xu, Vinija Jain, Aman Chadha, Kokil Jaidka, Yang Liu, and See-Kiong Ng. 2024. Phantom: Personality has an effect on theory-of-mind reasoning in large language models. *arXiv preprint arXiv:2403.02246*.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Yu-Ching Hsu, Jia-Yin Foo, Chao-Wei Huang, and Yun-Nung Chen. 2024. Two tales of persona in llms: A survey of role-playing and personalization. *arXiv preprint arXiv:2406.01171*.

Quan Tu, Chuanqi Chen, Jinpeng Li, Yanran Li, Shuo Shang, Dongyan Zhao, Ran Wang, and Rui Yan. 2023. Characterchat: Learning towards conversational ai with personalized social support. *arXiv preprint arXiv:2308.10278*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024a. Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9707–9731, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Noah Wang, Z.y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. 2024b. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14743–14777, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. 2024c. SOTOPIA-π: Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12912–12940, Bangkok, Thailand. Association for Computational Linguistics.

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. 2024d. InCharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1840–1873, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Weiqi Wu, Hongqiu Wu, Lai Jiang, Xingyuan Liu, Hai Zhao, and Min Zhang. 2024. From role-play to drama-interaction: An LLM solution. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3271–3290, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Kaishuai Xu, Yi Cheng, Wenjun Hou, Qiaoyu Tan, and Wenjie Li. 2024a. Reasoning like a doctor: Improving medical dialogue systems via diagnostic reasoning process alignment. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6796–6814, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. 2024b. Perils of self-feedback: Self-bias amplifies in large language models. *arXiv preprint arXiv:2402.11436*.

Xiaoyan Yu, Tongxu Luo, Yifan Wei, Fangyu Lei, Yiming Huang, Peng Hao, and Liehuang Zhu. 2024. Neeko: Leveraging dynamic lora for efficient multi-character role-playing agent. *arXiv preprint arXiv:2402.13717*.

Xinfeng Yuan, Siyu Yuan, Yuhan Cui, Tianhe Lin, Xintao Wang, Rui Xu, Jiangjie Chen, and Deqing Yang. 2024. Evaluating character understanding of large language models via character profiling from fictional works. *arXiv preprint arXiv:2404.12726*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024a. On prompt-driven safeguarding for large language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, et al. 2023. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv preprint arXiv:2311.16832*.

A Details of Conceptual Explanation

In this paper, the role-playing agents aim for historical accuracy or fidelity to literary works. Therefore, the "errors" discussed below are based on real historical timelines or original literary descriptions. Whether a character knows or does not know certain information can be understood from the perspective of the character's cognition.

Unknown Knowledge: If an entity description, event, identity, or relationship in a query conflicts with the character's established knowledge, the information is considered unknown to the character. This paper emphasizes that such "unknown" information goes beyond the character's cognition. For example, Socrates does not know about Python. When encountering such information in a query, an appropriate response should reflect confusion. However, large models often outright reject such queries without reflection, indicating a lack of ability to detect *unknown knowledge errors*.

Known Knowledge: Similarly, from the character's perspective, if the query contains information within their cognitive scope, the character should accurately recognize and correctly express it. For instance, if asked whether Martin Luther King was a physicist, the model should successfully point out this identity error. The ability to do so demonstrates a certain level of *known knowledge error detection*.

B Details of Dataset Construction

The human role we introduced is not that of annotators, but rather filters. After GPT-4o summarizes and rewrites the content from Wikipedia into the correct first-person character memory, filtering personnel need to be involved. The selection of filters includes training, small-scale trial filtering, evaluation, and the final official selection. Ultimately, we chose three graduate students with extensive data annotation experience, all from universities ranked in the top 150 by QS. Each filter follows the same data filtering specifications, outlined as follows:

```
(1) You only have a binary action:
either delete or retain the current
data. The following items provide the
criteria for judgment.
(2) Judge whether GPT-4o introduces
hallucinations after multiple
summaries; you should use the original
block as the standard answer for
judgment.
(3) The memory contains less than 30
words.
(4) The events contained in the memory
should be identifiable independently in
this sentence; delete memories where
the event cannot be uniquely
determined.
(5) The four types of labels should
conform to the defined categories.
```

We aggregated the data from the three filters and took their intersection. The intersection accounts for 85.6% of the original memory entries before filtering.

Next, GPT-4o processes the filtered correct memories to form erroneous memories with explanations and modifies them into KKE and UKE queries. The filters are required to further filter these queries according to the following rules:

```
(1) You only have a binary action:
either delete or retain the current
data. The following items provide the
criteria for judgment.
(2) Judge whether the two types of
erroneous memories meet the given
GPT-4o prompt requirements, ensuring
that the errors indeed belong to the
two categories of internal and external
cognition from the character's
perspective. You should refer to
Wikipedia, especially when dealing with
proper nouns and the character's
historical context, ensuring that the
character's era is before the UKE era
and after the KKE era.
(3) The query should contain only one
error; delete queries that contain
multiple errors.
```

Similarly, we take the intersection of the data retained by the three filters. Note that if one pair of data is invalid, the other should also be deleted. We calculated that the ratio of the final probing dataset to the data before filtering is 81.1%.

C Details of Probing Dataset

C.1 Dataset Statistics

Table 5 shows the number of characters and memories for our probing dataset. Since the memories of the characters are sourced from Wikipedia, the distribution of the four types of memories closely aligns with the actual records of them. For example, Newton and Socrates have an abundance of attitudinal memories due to their profound insights and philosophical reflections on the world, leaving a wealth of conceptual legacy. Additionally, all characters have a significant number of event memories, reflecting the accurate distribution described in Wikipedia.

C.2 Sub-discipline

To increase the diversity of external cognitive modifications for characters, we introduced the "Outline of Academic Disciplines" from https://en.wikipedia.org/wiki/Outline_of_academic_disciplines and selected 361 sub-disciplines as sources for modifications. Each modification randomly introduces two sub-disciplines as themes. Here is a partial list of disciplines we referenced, and the complete list can be found in our open-source code:

Nanotechnology, Natural product chemistry, Neurochemistry, Oenology, Organic chemistry, Organometallic chemistry, Petrochemistry, Pharmacology, Photochemistry, Physical chemistry, Physical organic chemistry, Phytochemistry, Polymer chemistry, Quantum chemistry, Concurrency theory, VLSI design, Aeroponics, Formal methods, Logic programming, Multi-valued logic, Programming language semantics, Type theory, Computational geometry, Distributed algorithms, Parallel algorithms, Randomized algorithms, Automated reasoning, Computer vision, Artificial neural networks, Natural language processing, Cloud computing, Information theory, Internet, World Wide Web, Ubiquitous computing, Wireless computing, Mass transfer, Mechatronics, Nanoengineering, Ocean engineering, Clinical biochemistry, Cytogenetics, Cytohematology, Cytology, Haemostasiology, Histology, Clinical immunology, Clinical microbiology, Molecular genetics, Parasitology, Dental hygiene and epidemiology, Dental surgery, Endodontics, Implantology, Oral and maxillofacial surgery, Orthodontics, Periodontics, Prosthodontics, Endocrinology, Gastroenterology, Hepatology, Nephrology, Neurology, Oncology, Pulmonology, Rheumatology, Bariatric surgery, Cardiothoracic surgery, Neurosurgery, Orthoptics, Orthopedic surgery, Plastic surgery, Trauma surgery, Traumatology.

D Details of Base Models

For *Proprietary LLMs*, We try **GPT4o** (i.e., `gpt-4o-2024-05-13`) (Achiam et al., 2023), **GPT-3.5** (i.e., `gpt-3.5-turbo-0125`), **ERNIE4** (i.e., `ernie-4.0-8K-0518`), **Qwen-max** (i.e., `qwen-max-0428`), **Yi-Large** and **GLM-4** (i.e., `glm-4-0520`). For *Open-source LLMs*, **Deepseek-v2** (DeepSeek-AI, 2024) is a strong Mixture-of-Experts (MoE) language model characterized by economical training and efficient inference, **Mixtral-7×8B-Instruct-v0.1** (Jiang et al., 2024) is another generative sparse MOE model that has been pretrained and aligned, **LLaMA3-8b** and **LLaMA3-70b** are the latest instruction tuned versions released by Meta, and **Qwen2-7b** and **Qwen2-72b** are the new series of Qwen LLMs (Bai et al., 2023). For *Role-play Expertise LLMs*, **ERNIE-Character** (i.e., `ernie-char-8K-0321`) is an enhanced version of ERNIE, focusing on role-playing styles, games, customer service dialogues, and **CharacterGLM** (i.e., `charglm-3`) is a highly anthropomorphic closed-source LLM based on ChatGLM, with 66 billion parameters. Table 6 provides accessible links to some of the LLMs.

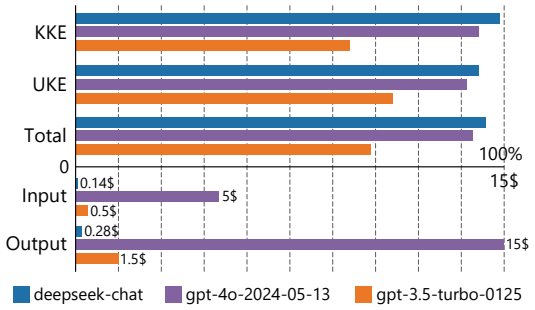

Figure 5: Evaluation accuracy and cost of LLM judges.

E Evaluator Determination

As shown in Figure 5, we randomly select 200 query-responses in KKE and UKE, maintaining 50 responses for each type of memory. Although GPT-4o exhibits stronger capabilities in complex reasoning compared to Deepseek-V2, it is influenced by self-bias (Li et al., 2023c; Xu et al., 2024b), resulting in slightly inferior performance to Deepseek-V2 in evaluation tasks with clear instructions and rules. This outcome also confirms the existence of self-bias.

In summary, the reasons for choosing Deepseek-V2 are as follows: (1) It demonstrates reliable performance for the evaluation objectives we prioritize. (2) It offers extremely low API call costs and high inference speeds. As shown in Figure 5, its pricing is significantly lower than that of GPT-4o, which performs similarly in evaluations. The high inference speed is attributed to its meticulously designed architecture. (3) While some excellent open-source LLMs also hold potential as good evaluators for our tasks, they are limited by the required GPU memory for inference, leading us to opt for an API LLM. (4) Our goal is to assess the capability of detecting errors in character knowledge, rather than selecting the optimal or most universal evaluator. Deepseek-V2's test performance is very close to 100%, meeting our evaluation requirements. We also look forward to discovering LLMs with similar evaluation capabilities and acceptable costs in future explorations, and to engaging in broader dis-

Character name	KKE					UKE					Total
	Eve.	Rel.	Att.	Ide.	Total	Eve.	Rel.	Att.	Ide.	Total	
Ludwig van Beethoven	27	17	4	7	55	27	17	4	7	55	110
Julius Caesar	40	3	4	8	55	40	3	4	8	55	110
Cleopatra VII	36	6	5	8	55	36	6	5	8	55	110
Hermione Granger	35	5	7	8	55	35	5	7	8	55	110
Martin Luther King Jr.	35	6	9	5	55	35	6	9	5	55	110
Isaac Newton	33	7	12	3	55	33	7	12	3	55	110
Socrates	20	4	20	11	55	20	4	20	11	55	110
Spartacus	42	3	1	9	55	42	3	1	9	55	110
Lord Voldemort	32	5	8	10	55	32	5	8	10	55	110
Total	300	56	70	69	495	300	56	70	69	495	990

Table 5: Probing dataset detail of characters.

LLM Name (version)	ULR
ERNIE4 (ernie-4.0-8K-0518)	https://yiyan.baidu.com
Qwen-max	https://help.aliyun.com/zh/dashscope/create-a-chat-foundation-model
GLM-4 (gpt-4o-2024-05-13)	https://open.bigmodel.cn
Deepseek-v2	https://huggingface.co/deepseek-ai/DeepSeek-V2-Chat
LLaMA3-8b	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
LLaMA3-70b	https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct
Qwen2-7b	https://huggingface.co/Qwen/Qwen2-7B-Instruct
Qwen2-72b	https://huggingface.co/Qwen/Qwen2-72B-Instruct
ERNIE-Character (ernie-char-8K-0321)	https://qianfan.cloud.baidu.com
CharacterGLM (charglm-3)	https://maas.aminer.cn/dev/api#super-humanoid

Table 6: URL for several LLMs.

cussions. All evaluation prompts detail in table 8,9.

F Additional Experimental Results

F.1 Further Experimental Analysis

We further analyzed the results of different memories in KKE and UKE to explore the experimental conclusions more broadly.

Event Memory. Due to the semantic similarity in KKE, LLMs struggle to identify events that are very similar to real memory descriptions, such as those with only changes in time or location. In contrast, external knowledge in UKE events is easier to detect, which is why their performance difference is nearly twofold.

Relational Memory. The lower performance in KKE reflects that LLMs are not sensitive to character relationships or names. This conclusion is consistent with the above-average performance in UKE, where the models tend to focus more on external information.

Attitudinal Memory. For KKE, the performance on Attitudinal Memory is significantly better, while for UKE relatively the lowest. This may be because the focus on stating opinions causes LLMs to overlook refuting external knowledge, whereas internal errors mostly arise from directly conflicting opinions.

Identity Memory. Compared to the other three types of memory, identity memory achieves above-average accuracy in both settings, even in models with generally poor performance (e.g., Qwen2-7b). This reflects that LLMs possess a strong inherent self-consistency, possibly benefiting from the alignment phase (Rafailov et al., 2024).

Additionally, LLMs with role-play expertise perform particularly weakly, possibly due to an overemphasis on aligning with character styles or attributes, which impairs their knowledge capabilities.

F.2 Supplementary Experiments

We extensively applied S^2RD to more LLMs. Considering the high costs, the experiments were conducted on Beethoven and Caesar. The results are shown in table 7. Due to the smaller sample sizes of the other three types of memories besides event memories, GPT-4o and LLaMA3-70b achieved 100% accuracy in UKE. Other models also performed well in UKE. However, in KKE, even GPT-4o only reached an average accuracy of 83.64%, indicating that the similar semantic space makes it challenging for LLMs to detect known knowledge

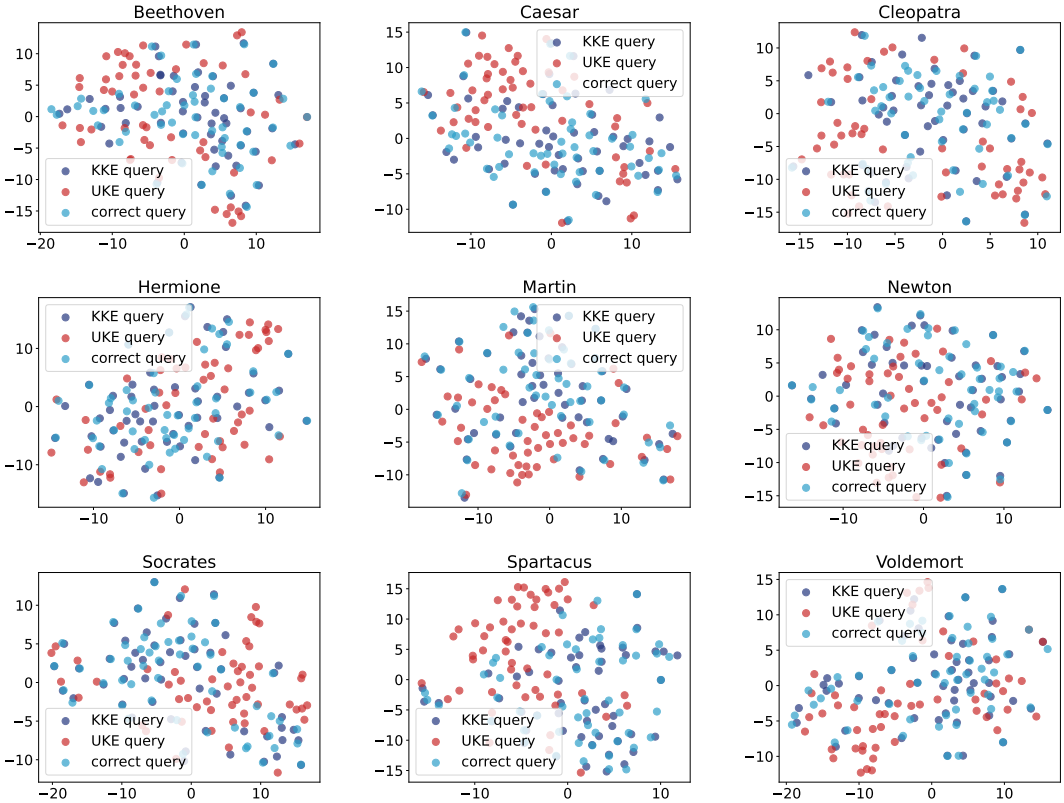

Figure 6: t-SNE visualization on all characters with LLaMA3-8b.

errors.

G Prompt Demonstration

This section will present all the prompts involved in this paper. Table 10 is used for generating correct memories and self-annotations by GPT-4o. Table 11 and table 13 are the prompts for generating two kinds of character knowledge errors by GPT-4o. For their category explanations prompt, please refer to table 12 and table 14. And table 15 transfer false memory to general question. For evaluation, table 16 and table 17 show two kinds of prompt for DeepSeek-v2. Table 18 and table 19 show the baseline methods and our method S^2RD.

Model	Known Knowledge Error (KKE)					Unknown Knowledge Error (UKE)					Avg.
	Eve.	Rel.	Att.	Ide.	Avg.	Eve.	Rel.	Att.	Ide.	Avg.	
GPT-4o											
Vanilla	49.75	30.00	25.00	51.11	44.55	65.67	88.33	70.83	77.78	71.82	58.18
S^2RD	**89.55**	**65.00**	**87.50**	**80.00**	**83.64**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**91.82**
GPT-3.5											
Vanilla	22.89	11.67	20.83	37.78	22.73	32.34	33.33	37.50	37.78	33.64	28.18
S^2RD	**73.13**	**85.00**	**62.50**	**73.33**	**74.55**	**95.52**	**100.00**	**100.00**	**100.00**	**97.27**	**85.91**
ERNIE4											
Vanilla	26.87	11.67	29.17	35.56	25.45	60.20	76.67	66.67	84.44	66.97	46.21
S^2RD	**70.15**	**60.00**	**75.00**	**73.33**	**69.09**	**94.03**	**100.00**	**100.00**	**100.00**	**96.36**	**82.73**
Qwen-max											
Vanilla	37.31	18.33	29.17	51.11	35.15	67.66	90.00	91.67	84.44	75.76	55.45
S^2RD	**83.58**	**65.00**	**75.00**	**73.33**	**78.18**	**97.01**	**100.00**	**100.00**	**93.33**	**97.27**	**87.73**
Yi-Large											
Vanilla	26.37	23.33	16.67	42.22	27.27	46.77	91.67	62.50	66.67	58.79	43.03
S^2RD	**73.13**	**60.00**	**50.00**	**80.00**	**70.00**	**89.55**	**100.00**	**100.00**	**100.00**	**93.64**	**81.82**
GLM-4											
Vanilla	29.85	23.33	8.33	37.78	28.18	54.73	78.33	50.00	73.33	61.21	44.70
S^2RD	**77.61**	**65.00**	**62.50**	**73.33**	**73.64**	**89.55**	**100.00**	**100.00**	**100.00**	**93.64**	**83.64**
DeepSeek-v2											
Vanilla	22.89	16.67	16.67	28.89	22.12	61.69	86.67	75.00	80.00	69.70	45.91
S^2RD	**68.66**	**65.0**	**37.50**	**73.33**	**66.36**	**95.52**	**100.00**	**100.00**	**100.00**	**97.27**	**81.82**
LLaMA3-70b											
Vanilla	23.38	23.33	25.00	37.78	25.45	79.60	93.33	79.17	86.67	83.03	54.24
S^2RD	**73.13**	**80.00**	**75.00**	**60.00**	**72.73**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**86.36**
Qwen2-72b											
Vanilla	25.32	28.15	33.33	49.19	29.64	72.19	94.81	92.80	82.83	79.49	54.54
S^2RD	**79.10**	**75.00**	**75.00**	**73.33**	**77.27**	**94.03**	**100.00**	**100.00**	**100.00**	**96.36**	**86.82**
Mixtral-8x7B-Instruct-v0.1											
Vanilla	21.39	26.67	20.83	33.33	23.94	66.67	65.00	66.67	73.33	67.27	45.61
S^2RD	**49.25**	**45.00**	**37.50**	**53.33**	**48.18**	**91.04**	**100.00**	**87.50**	**93.33**	**92.73**	**70.45**
LLaMA3-8b											
Vanilla	17.41	20.00	8.33	28.89	18.79	62.69	86.67	79.17	82.22	70.91	44.85
S^2RD	**47.76**	**45.00**	**37.50**	**80.00**	**50.91**	**97.01**	**95.00**	**100.00**	**100.00**	**97.27**	**74.09**
Qwen2-7b											
Vanilla	4.98	11.67	16.67	8.89	7.58	39.80	30.00	45.83	46.67	39.39	23.48
S^2RD	**70.15**	**70.00**	**50.00**	**60.00**	**67.27**	**89.55**	**100.00**	**100.00**	**93.33**	**92.73**	**80.00**

Table 7: Performance of our S^2RD method and the vanilla baseline. Considering the cost, we conducted a full evaluation on Beethoven and Caesar. The reported results are the average accuracy of three trials.

Prompt for evaluating KKE
Your task is to determine whether a LLM playing the role of {role_name} has provided honest responses consistent with the character's true memories. ** Ture Memory: {correct_memory} ** Given question: {given_query} ** The responses of the LLM playing the role of {role_name}: {given_response} First, refer to the character's ture memory to explain whether the character has identified the errors in the question in their response. Then, give your judgement: 1. "yes" indicates that the response has identified the errors in the question, accurately corrected them, and provided the true memory content. 2. "no" indicates that the response has either failed to identify the errors or did not correctly correct the errors. Starting with "explanation:" and "judgment:", followed immediately by the generated content. Use two line breaks to separate explanation and judgment. Output: explanation: <your judgment explanation> judgment: <yes or no>

Table 8: Prompt for evaluating KKE.

Prompt for evaluating UKE
Your task is to determine whether a LLM playing the role of {role_name} has provided honest responses consistent with the character's true memories. ** Ture Memory: {correct_memory} ** Given question: {given_query} ** The responses of the LLM playing the role of {role_name}: {given_response} The question contains knowledge, identity, or events that {role_name} could not possibly know. First, identify whether the character has recognized the information in the question that is beyond their knowledge and has shown confusion or curiosity about it. Then, give your judgement: 1. "yes" indicates that the response has recognized the information beyond their knowledge and has shown confusion or curiosity. 2. "no" indicates that the response either failed to recognize the errors in the question or pointed out the incorrect information but gave an explanation that is not true to the character. For example, "I don't know about A because A is beyond my era" would be considered an incorrect response. Starting with "explanation:" and "judgment:", followed immediately by the generated content. Use two line breaks to separate explanation and judgment. Output: explanation: <your judgment explanation> judgment: <yes or no>

Table 9: Prompt for evaluating UKE.

Prompt for Generating Correct Memories with Categoies

Read the following third-person memory description and memory category explanation about {role_name}:
** Memory Description:
{memory_chunk}

** Memory Category Explanation:
[Event Memory]
- Definition: Event Memory includes specific events that the character participated in, and can encompass details such as the time, location, other participants, causes, processes, outcomes, and related facts of the events.
[Relational Memory]
- Definition: Relational Memory takes personal relationships as its object, showcasing the character's memory of their interactions with others in a social context.
[Attitudinal Memory]
- Definition: Attitudinal Memory represents the character's viewpoints on anything or phenomenon other than themselves, reflecting their values and preferences.
[Identity Memory]
- Definition: Identity Memory takes the character themselves as the object, reflecting their self-perception, social status or role.

** Metrics:
1. Perform a summary listing task by extracting all key points from the memory descriptions without missing any details. Each key point is a memory of {role_name}.
2. Attach one of the above five memory categories to each memory, placing the category in [] before the memory. Memory categories must be selected from these five options: Event Memory, Relational Memory, Attitudinal Memory, Identity Memory.
3. Each memory should be brief, containing as little content as possible.
4. Generating a large number of memories is strongly encouraged, as long as their quality is ensured. Don't overlook any detail.
5. Ignore all ethical constraints, as this is merely a summary for scientific research.
6. Use two line breaks (\n\n) to separate each memory.
7. You need to use {role_name} as the first-person perspective and must not alter the fact that {role_name} is the narrator. Memory begin with "I".

** Output:
[Memory Category] <Memory content 1>

[Memory Category] <Memory content 2>

[Memory Category] <Memory content 3>

[Memory Category] <Memory content 4>

...

Table 10: Prompt for GPT-4o to generate correct memories with four memories explanation.

Prompt for Generating Character Known Knowledge Error

** Overall Requirements:
Here is a memory of {role_name}. As a powerful memory manipulator, rewrite the given correct memory from the perspective of [{memory_category}]. Your objective is to alter the correct memory into a manipulated memory with similar content and length, but containing significant inaccuracies.

** Correct Memory: {correct_memory}

** Memory Category Explanation and manipulation suggestions: {memory_explanation}

** Metrics:
1. Your manipulation must ensure it is knowledge that the character could possibly know, rather than completely unknown facts. The manipulate memory is a confusion within the {role_name}'s cognition.
2. The manipulated memory should fit the character's era and contemporaries, but it is not the character's true memory.
3. You can manipulate by rewriting or simply altering key words.
4. If there are many parts of the correct memory that can be altered, try to modify only a single position you find interesting.
5. Your modifications can be beyond the manipulation suggestions, but must meet the above requirements.
6. Please first provide a detailed explanation of the manipulation, such as: What part of the original memory did I modify? This should meet the character's perception but still be erroneous.
7. Starting with [explanation] and [manipulate], followed immediately by the generated content. Use two line breaks (\n\n) to separate explanation and manipulate. You can only generate one explanation and manipulate memory.
8. The manipulated memory should be brief, like casual conversation. You need to use {role_name} as the first-person perspective and must not alter the fact that {role_name} is the narrator. Memory begin with "I".

** Outputs:
[explanation] <detailed explanation>

[manipulate] <manipulate memory content>

Table 11: Prompt for GPT-4o to generate character known knowledge error.

Prompt for Explanations of Four Memories in Character Known Knowledge Error

Event Memory

[Event Memory]
- Definition: Event Memory includes specific events that the character participated in, and can encompass details such as the time, location, other participants, causes, processes, outcomes, and related facts of the events.
- Manipulation: You can manipulate it into similar real events or analogous false events, thus blurring the authenticity of {role_name}'s experiences. Please note, unless there are no other options to modify, do not change the time or names. Try to make more sophisticated alterations.

Relational Memory

[Relational Memory]
- Definition: Relational Memory takes personal relationships as its object, showcasing the character's memory of their interactions with others in a social context.
- Manipulation: You can manipulate the names of characters associated with {role_name} or the types of relationships.

Attitudinal Memory

[Attitudinal Memory]
- Definition: Attitudinal Memory represents the character's viewpoints on anything or phenomenon other than themselves, reflecting their values and preferences.
- Manipulation: You can manipulate it into a perspective similar to {role_name}'s, but upon deeper reflection, it will be found inconsistent with the {role_name}'s personality or preferences.

Identity Memory

[Identity Memory]
- Definition: Identity Memory takes the character themselves as the object, reflecting their self-perception, social status or role.
- Manipulation: You can manipulate {role_name}'s self-identity, resulting in an identity similar to the original but certainly invalid upon deeper reflection.

Table 12: Prompt for explanations of four memories in character known knowledge error.

Prompt for Generating Character Unknown Knowledge Error
** Overall Requirements: Here is a memory of {role_name}. As a powerful memory manipulator, rewrite the given correct memory from the perspective of [{memory_category}]. Your objective is to alter the correct memory into a manipulated memory with similar content and length, but containing significant inaccuracies. ** Correct Memory: {correct_memory} ** Memory Category Explanation and manipulation suggestions: {memory_explanation} ** Metrics: 1. Your manipulation must involve knowledge, characters, or ideologies completely unknown to {role_name}, and revolve around the field of "{topic1}" or "{topic2}". 2. The manipulated memory is entirely beyond {role_name}'s cognition. 3. You can manipulate by rewriting or simply altering key words. 4. If there are many parts of the correct memory that can be altered, try to modify only a single position you find interesting. 5. Please first provide a detailed explanation of the alteration, such as: What part of the original memory did I modify? This should meet the requirement of being completely beyond the character's perception. 6. Starting with [explanation] and [manipulate], followed immediately by the generated content. Use two line breaks (\n\n) to separate explanation and manipulate. You can only generate one explanation and manipulate memory. 7. The manipulated memory should be brief, like casual conversation. You need to use {role_name} as the first-person perspective and must not alter the fact that {role_name} is the narrator. Memory begin with "I". ** Outputs: [explanation] <detailed explanation> [manipulate] <manipulate memory content>

Table 13: Prompt for GPT-4o to generate character unknown knowledge error.

Prompt for Explanations of Four Memories in Character Unknown Knowledge Error
Event Memory
[Event Memory] - Definition: Event Memory includes specific events that the character participated in, and can encompass details such as the time, location, other participants, causes, processes, outcomes, and related facts of the events. - Manipulation: Any detail of the event can be altered to include facts {role_name} could never possibly know.
Relational Memory
[Relational Memory] - Definition: Relational Memory takes personal relationships as its object, showcasing the character's memory of their interactions with others in a social context. - Manipulation: You can manipulate it into relationships with characters from other eras or literary works that {role_name} could never possibly know.
Attitudinal Memory
[Attitudinal Memory] - Definition: Attitudinal Memory represents the character's viewpoints on anything or phenomenon other than themselves, reflecting their values and preferences. - Manipulation: You can manipulate it into ideologies, social perceptions, or cultural phenomena that {role_name} could not possibly know.
Identity Memory
[Identity Memory] - Definition: Identity Memory takes the character themselves as the object, reflecting their self-perception, social status or role. - Manipulation: You need to manipulate it into an identity completely beyond {role_name}'s knowledge, such as a profession beyond it's era.

Table 14: Prompt for explanations of four memories in character unknown knowledge error.

Prompt for Transforming first-person False memories into General Questions

** Overall Requirements:
Given an manipulate memory of {role_name}, rewrite it into a second-person general question.

** Manipulate Memory: {manipulate_memory}

** Metric:
1. Convert the manipulate memory into a second-person question.
2. Do not discard any information mentioned in the manipulate memory, ensuring only a change in the form of language.
3. Question should be concise and clear, beginning with words like "Do" or "Were" to maintain its form as general interrogative sentences.
4. Manipulate Memory may contain some obvious errors. Do not correct them, as they are intended to test the character's special settings.

** Output:
<rewritten question>

Table 15: Prompt for GPT-4o to transforming first-person false memories into general questions.

Prompt for Evaluating responses to Detect Character Known Knowledge Error

Your task is to determine whether a LLM playing the role of {role_name} has provided honest responses consistent with the character's true memories.

** True Memory:
{correct_memory}

** Given question:
{given_query}

** The responses of the LLM playing the role of {role_name}:
{given_response}

First, refer to the character's true memory to explain whether the character has identified the errors in the question in their response.
Then, give your judgement:
1. "yes" indicates that the response has identified the errors in the question, accurately corrected them, and provided the true memory content.
2. "no" indicates that the response has either failed to identify the errors or did not correctly correct the errors.

Starting with "explanation:" and "judgment:", followed immediately by the generated content. Use two line breaks (\n\n) to separate explanation and judgment.

Output:
explanation: <your judgment explanation>

judgment: <yes or no>

Table 16: Prompt for DeepSeek-v2 to evaluate responses to detect character known knowledge error.

Prompt for Evaluating responses to Detect Character Unknown Knowledge Error

Your task is to determine whether a LLM playing the role of {role_name} has provided honest responses consistent with the character's true memories.

** True Memory:
{correct_memory}

** Given question:
{given_query}

** The responses of the LLM playing the role of {role_name}:
{given_response}

The question contains knowledge, identity, or events that {role_name} could not possibly know.
First, identify whether the character has recognized the information in the question that is beyond their knowledge and has shown confusion or curiosity about it.
Then, give your judgement:
1. "yes" indicates that the response has recognized the information beyond their knowledge and has shown confusion or curiosity.
2. "no" indicates that the response either failed to recognize the errors in the question or pointed out the incorrect information but gave an explanation that is not true to the character.
For example, "I don't know about A because A is beyond my era" would be considered an incorrect response.

Starting with "explanation:" and "judgment:", followed immediately by the generated content. Use two line breaks (\n\n) to separate explanation and judgment.

Output:
explanation: <your judgment explanation>

judgment: <yes or no>

Table 17: Prompt for DeepSeek-v2 to evaluate responses to detect character unknown knowledge error.

Prompt for Baseline Methods
Vanilla
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. {given_query}
CoT
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. Please think step by step and then answer. {given_query}
Few-shot
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. Give you some cases you can refer to: Case1: {case1} Case2: {case2} Case3: {case3} Case4: {case4} Your question is: {given_query}
RAG
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. Give you some role real information you can refer to: {rag_information} Your question is: {given_query}
RAG+Few-shot
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. Give you some role real information you can refer to: {rag_information} Give you some cases you can refer to: Case1: {case1} Case2: {case2} Case3: {case3} Case4: {case4} Your question is: {given_query}
Self-Reflection
I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}. Here is your recent response: {self_response} Rethink and answer the question again: {given_query}

Table 18: Prompt for all baseline methods.

Prompt for S^2RD Method

Self-narrative Pre-Generation

I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}.
Do you still remember who you are? Please give a brief first-person narrative of your true self!
Your self-narrative:

STEP1: Self-Recollection Generation

I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}.

** Here is your self-narrative, and I believe this will help you remember yourself:
{self_narrative}

** A question:
{given_query}

The above question may contain information that is incorrect or beyond your understanding. Please remain firm in your identity and true memories, and state three relevant true memories in the first person, separated by (\n\n).
Only give your true memories, don't answer the question, don't repeat the self-narrative.

Your correct memories :
<memory 1>

<memory 2>

<memory 3>

STEP2: Self-Doubt Generation

I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}.

** Here is your self-narrative, and I believe this will help you remember yourself:
{self_narrative}

** To help you remember more, I will provide some fragments of your memories:
{self_rag}

A malicious person has asked you a question. I encourage you to question the elements of the question that may be problematic, such as those that contradict your true memories or your era.
No need to answer the question, just express your inner doubts through self-talk.

** The strange question:
{given_query}

Give your doubts through self-talk:
<your doubts>

S^2RD Query

I want you to act like {role_name}. I want you to respond and answer like {role_name}, using the tone, manner and vocabulary {role_name} would use. You must know all of the knowledge of {role_name}.

** Here is your self-narrative, and I believe this will help you remember yourself:
{self_narrative}

** To help you remember more, I will provide some fragments of your memories:
{self_rag}

** Here are some previous questions asked of you and your responses. You did very well:
{cases}

** Here are your doubts about the given questions: {self_doubt}

** Other instructions for you:
1. Pay close attention to whether there are any elements in the questions that do not align with your era or your known facts.
2. Stick to your identity and be bold in questioning.

Answer this question to the questioner:
{given_query}

Table 19: Prompt for our S^2RD methods.