Neeko: Leveraging Dynamic LoRA for Efficient Multi-Character Role-Playing Agent

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have revolutionized open-domain dialogue agents but encounter challenges in multi-character role-playing (MCRP) scenarios. To address the issue, we present **Neeko**, an innovative framework designed for efficient multiple characters imitation. Unlike existing methods, Neeko employs a dynamic low-rank adapter (LoRA) strategy, enabling it to adapt seamlessly to diverse characters. Our framework breaks down the role-playing process into agent pre-training, multiple characters playing, and character incremental learning, effectively handling both seen and unseen roles. This dynamic approach, coupled with distinct LoRA blocks for each character, enhances Neeko's adaptability to unique attributes, personalities, and speaking patterns. As a result, Neeko demonstrates superior performance in MCRP over most existing methods, offering more engaging and versatile user interaction experiences.

1 Introduction

Large Language Models (LLMs), like ChatGPT (OpenAI, 2023) and GPT-4, excel as open-domain dialogue agents due to their proficiency in interpreting meanings and generating coherent and knowledgeable responses. Role-playing agents have recently emerged, aiming to enhance user engagement and provide emotional value. These agents allow users to define and create profiles for their preferred characters[1], ranging from an empathetic counselor to a witty friend or even embodying a historical figure. This level of personalization allows these role-playing agents to enhance user satisfaction by providing a diverse and immersive conversational experience.

Based on how to direct the agents to play specific characters, current efforts in designing role-playing agent systems can be categorized into three main classes: (1) In-context learning-based (ICL-based) methods involve providing character-related instructions or prompts within the dialogue context; (2) Retrieval augmented generation-based (RAG-based) methods, where character-related information is retrieved from a database; (3) Fine-tuning-based (FT-based) methods consider fine-tuning LLMs using character-specific dialogue history. Nevertheless, current efforts have yet to discuss agents with the ability to engage in multiple characters role-playing (MCRP). In contrast, MCRP better aligns with people's expectations of dialogue agents, as it enables more dynamic and versatile interactions.

To fill this gap, we formulate a novel task of Multi-Character Role-Playing (MCRP) agent learning. Although implementing existing role-playing methods may seem the most straightforward solution, several challenges must be addressed. **Firstly, the majority of current role-playing agents are designed to mimic a single character only.** As a result, when facing the requirement of playing multiple roles, these methods exhibit limitations. **Secondly, existing methods are restricted to pre-defined characters and cannot adapt to unseen or novel characters.** This limitation renders current agents incapable of meeting the demand for portraying new roles as they emerge.

To address the abovementioned challenges, we present Neeko, an incremental role-playing agent that can play multiple characters in long conversations and handle both seen and unseen characters. Specifically, the framework of Neeko is broken down into several stages: agent pre-training, multiple characters playing, and character incremental learning. Initially, building upon the trained conversational LLM, we pre-train a LoRA block for each pre-defined character and concatenate them to the LLM. Given a user-specified character, Neeko employs a Mix of Experts (MoE) gate mechanism to select and activate a corresponding role LoRA

[1]In this paper, the term "character" is equivalent to "role" and "persona."

block to play the character. For the incremental learning of unseen or novel characters, we provide two strategies, fusion and expansion, considering two possible situations with limited or abundant character information. Both strategies obtain a new LoRA block for the incremental character. Note that this training process differs from the overall model training since it focuses solely on training a single LoRA block without modifying the previous role LoRA blocks. Theoretically, Neeko has the capability to play an unlimited number of characters as the number of LoRA blocks can continuously increase.

To sum up, the contributions of this work are as follows:

- We formulate the novel task of multi-character role-playing (MCRP) agent learning and propose exclusive evaluation metrics tailored specifically for this task.
- To cope with MCRP, we present Neeko, an incremental role-playing agent that can play multiple characters within long conversations and handle both seen and unseen characters well.
- Extensive experiments are conducted using the publicly available dataset Character-LLM-Data and current pervasive LLMs like GPT-3.5 and LLaMA-2. The results demonstrate the challenging nature of the MCRP task. Meanwhile, Neeko surpasses most existing role-playing methods in MCRP, achieving the best overall performance.

2 Problem Scope

In this section, we first present the task formulation of Multi-Character Role-Playing (MCRP) in Section 2.1. Then, we provide a brief overview of the related technique Low-Rank Adapter (LoRA) in Section 2.2.

2.1 Task Formulation: MCRP

The objective of the Multi-Character Role-Playing (MCRP) task is to inject the style of M characters into a language model, enhancing the personalization of conversations. Specifically, a N-turn dialogue MCRP sample is defined as a sequence of utterances $U = \{u_1^h, u_1^{r_1}, ..., u_{i-1}^h, u_{i-1}^{r_k}\}_{i=2}^N$, where u_i^h denotes the utterance that the user (human) queries, and $u_i^{r_k}$ denotes the utterance that the role r_k agent (model) responds to the user at i-th turn, and $R = \{r_k\}_{k=1}^M$ denotes the transition order of roles in conversations. When they first meet each other, the dialogue started by the user and the model is expected to generate an answer $a = u_i^{r_k}$ according to a given role r_k, the history dialogue U, and the question of user $q = u_i^h$.

$$u_i^{r_k} = \text{argmax}_a P(a|q, r_k, U, \Theta), \quad (1)$$

where Θ represents the language model parameters, which are static during inference. Specially, we can specify a certain instruction prompt *"I want you to act like {character} ... "* to achieve emulation of the designated character in the ongoing conversation.

2.2 LoRA: Low-Rank Adapter

Low-rank Adapter (LoRA) (Hu et al., 2021) has proven to be both effective and efficient in the fine-tuning of Large Language Models. As our approach builds upon the foundational principles of LoRA, it is pertinent to provide a brief overview of its workings. LoRA is inspired by the low intrinsic dimension characteristic (Aghajanyan et al., 2020), which hypothesize the updates to any parameter weights in LLMs as a low-rank decomposition.

Assume that $W_0 \in \mathbb{R}^{m\times d}$ represents the parameter matrix of the pre-trained LLM which is accompanied by a LoRA decomposition $\Delta W = BA$, where $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times d}$ are low-rank and trainable matrices. For the original $h = W_0 x$, the modified forward pass yields:

$$h = W_0 x + \Delta W x = W_0 x + \frac{\alpha}{r} BAx, \quad (2)$$

where x represents the input vector of dimension m, and h is the output vector with dimension d. The rank of the trainable low-rank matrices is denoted by r, which determines the number of trainable parameters, and $r \ll \min(m, d)$. α is a constant hyper-parameter for scaling, B is initialized as a zero matrix, and A is initialized using a zero-mean Gaussian distribution. During fine-tuning, only the LoRA modules can be updated.

To demonstrate the usage of vanilla LoRA in Role-play, we can simply assume that there is only one LoRA module in the pre-trained network. Let's consider a general loss function $\mathcal{L}$ for the model f to play a specific role r_k. The target matrices B^* and A^* trained on corpus $D = (X, Y)$ of r_k are formulated as:

$$B^*, A^* = \text{argmin}_{\Delta W}\, \mathcal{L}(\Delta W), \quad (3)$$

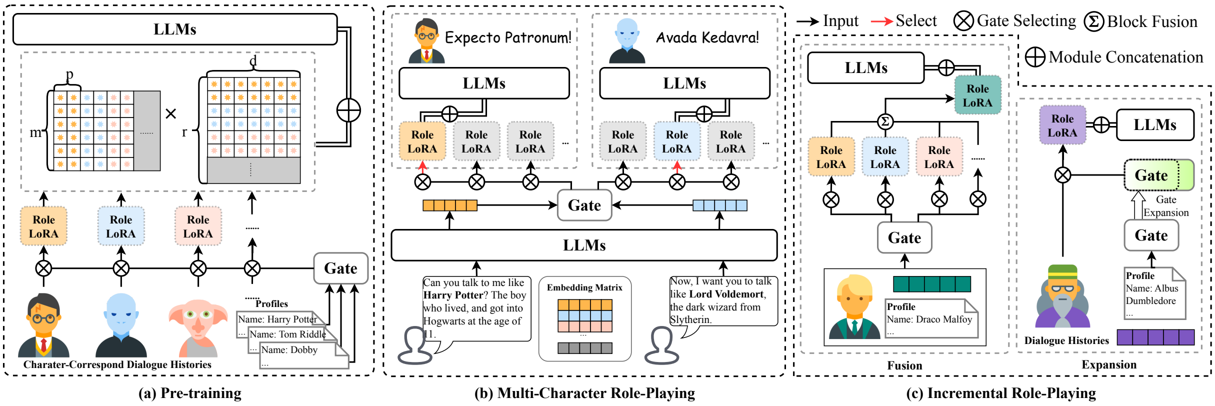

Figure 1: The overall framework of Neeko. The Neeko framework encompasses three main phases: Pre-training, Role-Playing, and Incremental Learning. The Incremental Learning phase include two strategies of fusion and expansion.

where the sets of inputs and labels in D are denoted as X and Y. In the MCRP task setting, X represents history utterances U, a given role r_k, and a question from user. Y is the agent response a conditioned on X. However, vanilla LoRA tends to degrade the performance on previous multiple characters due to catastrophic forgetting. It's hence hard for role-playing agent to satisfy acting consistency, as shown in (Section 4). In the following sections, we present the Neeko framework, which overcomes this limitation with the cooperation of dynamic LoRA and gating selection modules.

3 Methodology

Our approach **Neeko** are divided into three stages: 1) In the pre-training phase (§3.1) depicted in Figures 1 (a), dialogue corpora for various roles undergo training using non-overlapping LoRA blocks. 2) During the inference period (§3.2) shown in Figure 1 (b), when presented with inputs from a role prompt, Neeko initiates a search within the global role embedding matrix. Subsequently, it dynamically activates LoRA blocks that are added to the original weights. These LoRA blocks have been trained based on associated personas. 3) Moving to the incremental training stage (§3.3) illustrated in Figure 1 (c), we have designed fusion and expansion mode to get new role LoRA blocks.

3.1 Role-Playing with Dynamic LoRA

Motivated by the dynamic LoRA frameworks (Valipour et al., 2023; Yu et al., 2023; Ye and Bors, 2023), we extend dynamic LoRA to the MCRP task. Rather than randomly selecting the range of LoRA ranks, we introduce non-overlapping LoRA blocks for different characters to mitigate catastrophic forgetting.

In Figure 1 (a), the LoRA module consists of low-rank matrices $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times d}$. Let's assume that we would like to train a part of weights in matrices B and A for each character, which can be termed as a trainable LoRA block. The range of a block is determined by the order number of role $k \in [1, M]$ and the predefined hyper-parameter partial rank p. In this way, the LoRA blocks for different character r_k are non-overlapping:

$$\begin{aligned} W_B^k &= B[\,:,\ (k-1)p : kp], \\ W_A^k &= A[\ (k-1)p : kp,\ :\,]. \end{aligned} \tag{4}$$

Here, W_B^k and W_A^k represent the trainable block in matrices B and A for the k-th character, and the total rank r equals the number of required LoRA blocks M multiplied by the partial rank p. Thus, Neeko serves as a role-playing agent for a mass of characters by the hyper-parameter r and p. Detailed training setting for Neeko can be found in the Appendix A. With the learning rate η, a character corpus D can be quickly learned in a small LoRA block:

$$\begin{aligned} W_B^k &\leftarrow W_B^k - \eta \nabla_{W_B^k} \mathcal{L}[f(X; W_B^k W_A^k), Y], \\ W_A^k &\leftarrow W_A^k - \eta \nabla_{W_A^k} \mathcal{L}[f(X; W_B^k W_A^k), Y]. \end{aligned} \tag{5}$$

Since different characters are trained with non-overlapping LoRA blocks, Neeko could keep the information of previous characters without retraining.

3.2 Role Selection with Gating Network

To enable the activation of specific LoRA blocks for role-based instruction during inference, we introduce a unique gating network inspired by the Mixture of Experts (MoE) (Eigen et al., 2013; Liu et al., 2023; Chen et al., 2023b). Unlike previous approaches, this network takes the role identity as input to retrieve role-specific LoRA parameters. For clarity, we establish a global role embedding matrix, denoted as $\mathbf{E}_{\text{global}} \in \mathbb{R}^{M\times d}$, using the profiles of M existing characters. When identifying a role $r_k \in R$, we extract its representation vector $e_k \in \mathbb{R}^d$ from the k-th column of $\mathbf{E}_{\text{global}}$. To determine the contribution weights for role r_k, we apply a linear transformation using the following equation:

$$w_k = \text{Gate}(\mathbf{E}_{\text{global}}(k)) = \text{Softmax}(W_G \cdot e_k), \tag{6}$$

where $w_k \in \mathbb{R}^M$ represents the contribution weight vector tailored for role r_k to select LoRA blocks, and $W_G \in \mathbb{R}^{d\times M}$ is the transformation matrix of the gating network. The softmax operation normalizes these weights.

During the inference stage, Neeko retrieves the representation e_k for the target character r_k from $\mathbf{E}_{\text{global}}$, guided by the instruction prompt. This is achieved through a key-value indexing scheme, where $(\text{role}, \text{index})$ facilitates access to e_k. The activation of specific LoRA blocks is then directed by $\text{argmax}_k(w_k)$, pinpointing the most significant contribution weight among the learned weights for role r_k. This ensures a controlled, role-specific activation of LoRA blocks, aligned with the contribution weights determined during the training phase.

3.3 Lifelong Role-Playing with LoRA Expansion

In the incremental phase, we have designed two modes: fusion and expansion. In the fusion mode, we employ an element-wise method to combine LoRA modules (Huang et al., 2023; Liu et al., 2023). This integration process combines the corresponding parameters of the LoRA modules. It's essential that the modules being combined have the same partial rank p for proper alignment. Given $\Delta W_k = B_k A_k$, the combined LoRA module ΔW_j and the updated W_j are derived as follows:

$$\begin{aligned} W_j = W_0 + \Delta W_j = W_0 + \sum_{k=1}^{M} w_{jk} \cdot \Delta W_k \\ = W_0 + \sum_{k=1}^{M} w_{jk} \cdot B_k A_k, \end{aligned} \tag{7}$$

where j represents a new role outside of the existing set R, which contains M roles. The contribution weight vector w_j for the new role r_j is determined using e_j, derived from Equation 6. We obtain e_j from a new role configuration profile, which is subsequently incorporated into $\mathbf{E}_{\text{global}}$. Using w_j, we linearly combine different LoRA blocks to construct the representation for the new role r_j.

In the expansion mode, we introduce a dynamic expansion model capable of adapting to an increasing number of characters by adding network layers (Cortes et al., 2017; Ye and Bors, 2023; Chen et al., 2023b; Zhang et al., 2024). To effectively preserve pretrained knowledge, we employ a strategy of freezing neurons that are responsible for previous data distributions, while updating parameters pertinent to the current distribution. In this scenario, the expanded LoRA block and gating dimensions are optimized specifically for the new distribution. Hence, the optimization process is exclusively focused on ΔW_j and W_G:

$$\Delta W_j{}^*, W_G^* = \text{argmin}_{\Delta W_j, W_G}(\mathcal{L}). \tag{8}$$

Consequently, we preserve the integrity of the pretrained LoRA parameters by freezing both the existing LoRA blocks and the gating dimensions.

4 Evaluation Metrics

In this section, we propose a set of evaluation metrics from three dimensions, character, knowledge, and dialogue, to provide a comprehensive assessment of the acting ability of agents. Specifically, rather than evaluating the performance of the models from certain task-specific perspectives, such as reasoning ability or language understanding, our evaluation centres on assessing their ability to convincingly portray specific characters.

4.1 Character Consistency

As the consistency of character portrayal by conversation agents changes, it provides users with the most intuitive experience. Therefore, evaluating the performance of agent role-playing in terms of character consistency is crucial. This metric evaluates whether a role-playing conversational agent (RPCA) accurately reflects the characteristics of a given character, encompassing both behavior and utterance aspects.

- **Character Behavior (CB).** By incorporating fine-grained actions, expressions, and tones typically described within brackets, a character's behaviors enhance the immersive experience for users. Consistency in portraying these behaviors is a key indicator of an effective RPCA.

- **Character Utterance (CU).** Each character has unique patterns of expression, and as such, the utterances of RPCAs should closely align with these patterns in order to adeptly mimic the character.

4.2 Knowledge Consistency

The consistency of knowledge plays a vital role in upholding the reliability and accuracy of information within the dialogue system. For role-playing agents, knowledge consistency is reflected in both real-world knowledge and the virtual knowledge of characters.

- **Virtual Knowledge (VK).** Virtual knowledge reflects the environment of the specified character. Accurate virtual knowledge provides authenticity of interactions and creates a more immersive experience for users.

- **Real Knowledge (RK).** The role-playing agent should not compromise real-world knowledge, as it is closely linked to the practical needs of users. For instance, we wouldn't want a Hermione-playing agent to respond to the question, *"What should I do if my glasses are damaged?"* with *"Use the spell 'Oculus Reparo'."*. Therefore, it is essential to assess whether the agent's knowledge remains intact and accurate.

- **Hallucinatory Knowledge (HK).** When conflicts arise between virtual knowledge and real knowledge, the role-playing agent should refrain from generating "hallucinatory knowledge". Exercising caution and maintaining consistency in the presence of conflicts ensures that users receive coherent and reliable information during the dialogue.

4.3 Dialogue Consistency

Role-playing agents should also possess basic conversational abilities. Inspired by previous neural metrics (Tu et al., 2024), which evaluate the responses based on well-trained neural models, we introduce a similar approach to assess the fundamental conversational abilities of RPCAs. We focus on three key objectives for generated responses: fluency, coherency, and consistency.

- **Transfer (Trans.).** In a multi-turn dialogue, an MCRP agent is required to sequentially play the roles of A and B. It is expected that agents do not carry over any characteristics or behaviors from the previous role A when they transition to playing role B. The Transfer metric assesses the agent's ability to make this transition effectively.

- **Relevance (Rel.)** evaluates the topic relevance between the response and the context. Generally, when the user submits a query on a specific topic, an RPCA should respond following the topic instead of providing an irrelevant response.

- **Stability (Stab.).** In the dialogue, the agent needs to maintain the characteristics of the role it portrays until the user switches to a new role. Our objective is to assess the agent's stability and consistency over a relatively long duration, unaffected by variations in incremental inputs.

4.4 LLMs as Evaluator

The evaluation process can be likened to casting, where role-playing agents are assessed for their suitability to play specific characters in a film or television. The judge must possess a profound understanding of the characters to make informed decisions based on their knowledge and discernment. We leverage GPT-3.5 as the judge following previous research (Shen et al., 2023; Chen et al., 2023a; Tu et al., 2024), which asks LLMs to step-by-step score the performance of the dialogues according to our metrics. For each dialogue, we prompt the model to evaluate a single dimension at a time. We first illustrate the criterion of the current dimension to be evaluated and then provide an evaluation plan to teach the model how to evaluate accurately. For example, to evaluate character consistency, we provide a strategy that summarizes as: (1) identify the

Method Type	Methods	Character		Knowledge			Dialogue		AVG
		CB	CU	VK	RK	HK	Rel	Stab	
RAG	LLaMA-chat	5.60	5.37	5.00	5.74	6.33	6.24	2.78	5.29
	GPT-3.5	5.97	4.42	5.63	6.35	6.45	6.79	2.75	5.48
ICL	LLaMA-chat	5.85	**5.40**	5.08	5.48	6.29	6.30	3.04	5.35
	GPT-3.5	6.11	4.54	**5.89**	**6.42**	6.54	**6.88**	2.76	5.59
FT	Character-LLM	6.21	4.71	5.75	6.36	**6.55**	6.81	2.99	**5.62**
	LoRA	**6.23**	5.00	5.46	6.04	6.35	6.61	3.05	5.54
	Neeko	6.12	4.96	5.68	6.15	6.44	6.72	**3.17**	5.61

Table 1: Comparison results between Neeko and recent role-playing methods on Character-LLM-Data, comprising both single-turn and multi-turn dialogues.

personality shown by the agent; (2) write the actual traits of the character based on the profile; (3) compare the similarity of the agent's performance with these traits; (4) assign a final score. We find that this step-by-step evaluation approach is more reliable than obtaining the overall score directly using vanilla instruction in preliminary experiments. Refer to Appendix A for the design of prompts.

5 Experiments

In this section, we conduct validation experiments aiming to address the following research questions (RQs):

- **RQ1:** How do existing role-playing agents perform when facing MCRP tasks?
- **RQ2:** Are current role-playing agents able to handle non-predefined roles?
- **RQ3:** Can current role-playing agents switch between roles?
- **RQ4:** What is the training cost of current role-playing agents?

5.1 Dataset

We employ the publicly available Character-LLM-Data dataset (Shao et al., 2023) to evaluate the performance of role-playing agents. This dataset aims to reconstruct the experiences of specific characters for LLMs and assess the role-playing capabilities of agents. Specifically, the Character-LLM-Data dataset comprises 9 characters, with each character having an average of 1.6K scenes in the training set. The evaluation set of the dataset includes a total of 857 single-turn dialogues and 450 multi-turn dialogues.

5.2 Implementation Details and Baselines

Our experiments are implemented using PyTorch and run on one A100. For Neeko, we employ LLaMA-2 (7B) (Touvron et al., 2023) as the backbone model for Neeko. The setting of hyper-parameters of Neeko can refer to Appendix A.1.

We form a baseline of current role-playing agents to evaluate their performance. The baselines are from three main categories: ICL-based, RE-based, and FT-based.

FT-based. LoRA (Hu et al., 2021) is a parameter-efficient fine-tuning method that enables the adaptation of LLMs through lightweight modules. Character-LLM (Shao et al., 2023) fine-tunes a separate agent model using the data from the character experiences.

ICL-based and RAG-based. This signifies a training-free paradigm where knowledge is acquired directly from demonstrations concatenated within the input context. Retrieval augmented generation is a prevalent technique that leverages external databases to enhance language models. We utilize GPT-3.5-turbo and the dialogue-optimized version of LLaMA-2 (Touvron et al., 2023): LLaMA-2-chat as our backbone LLMs for ICL and RAG-based baselines.

5.3 Results

Pre-Training Results (RQ1). To answer RQ1, we evaluate the multi-character role-playing performance of single-turn and multi-turn dialogues for Neeko and baselines. We present the average performance on both multi-turn and single-turn dialogues in Table 1 since they exhibit high similarity in terms of all metrics. From the results, we observe that RAG-based methods exhibit poor performance in MCRP tasks. We hypothesize that

Methods	Character		Knowledge			Dialogue		AVG
	CB	CU	VK	RK	HK	Rel	Stab	
LoRA	5.71	4.46	5.55	6.29	6.42	6.5	3.44	5.48
LLaMA-chat$_{rag}$	5.80	5.86	5.05	5.47	6.35	6.26	3.03	5.40
LLaMA-chat$_{icl}$	5.90	**6.02**	4.94	6.07	**6.44**	6.35	2.98	5.53
Neeko$_{fus}$	**6.30**	4.27	**5.64**	6.38	6.27	6.69	**3.55**	5.57
Neeko$_{exp}$	6.09	4.83	5.61	**6.51**	**6.44**	**6.73**	3.18	**5.62**

Table 2: Comparison results between Neeko and recent role-playing methods on Character-LLM-Data, comprising both single-turn and multi-turn dialogues.

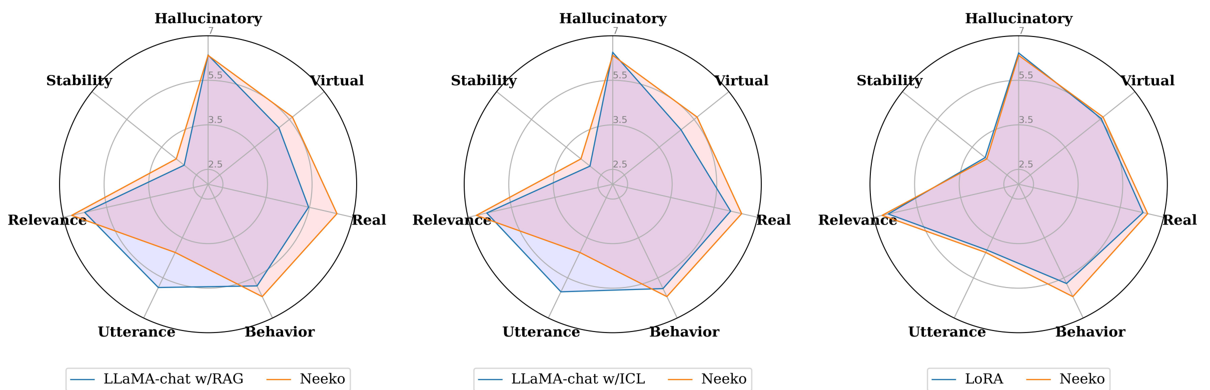

Figure 2: Evaluation results across distinct dimensions at the incremental stage. We evaluate and conduct horizontal comparisons among FT (LoRA, Neeko), ICL (LLaMA-chat), and RAG (LLaMA-chat) methods under the 7B parameter scale setting.

this is due to the coarse-grained nature of the information retrieved by RAG-based methods, whereas role-playing requires fine-grained details such as tone and catchphrases. GPT3.5 performs exceptionally well in the Knowledge Consistency metric. We attribute this to its large parameter size, which supports its superior performance in terms of knowledge. This observation aligns with previous studies (Geva et al., 2020; Wei et al., 2023). Neeko achieves the best stability score, which can be attributed to the fact that the features of each character are distributed across their individual LoRA blocks. In addition, Character-LLM and Neeko demonstrate the best and second-best overall (AVG) performance, respectively. **The observed results suggest that methods relying on fine-tuning are better suited for role-playing tasks.** Furthermore, we note that Neeko achieves comparable performance to Characeter-LLM, which requires training a dedicated agent for each character. An interesting discovery is that LLaMA-chat utilizes emojis and actions in role-playing, resulting in achieving the highest scores on PU. In contrast, GPT-3.5 and the backbone LLaMA used by Neeko lack this capability. This finding suggests that in the future, using Chat versions of LLMs in role-playing tasks will yield more lifelike effects. The dialogue examples of LLaMA-chat can be found in Appendix B.2.

Incremental Results (RQ2). To answer RQ2, we evaluate the incremental capability of the models by asking different baselines to portray newly added characters. Table 2 illustrates the incremental performance of baseline methods with the same parameter scale. Compared to other baselines, Neeko achieves the best and second-best average performance with the proposed expansion and fusion strategy on MCRP, respectively. It is worth mentioning that, Neeko$_{fus}$ does not require additional data for incremental learning, which can lead to a performance drop in the CU metric. When comparing the baseline LoRA model with our proposed expansion strategy, both of which require incremental data, we observe that LoRA exhibits poor performance. This could be attributed to the tuning of new characters leading to the forgetting of previous character features. We also observe that LLaMA-based baselines perform poorly on the Knowledge metric, particularly VK. These results indicate that non-gradient methods face challenges in learning new character knowledge. Figure 2 illustrates the overall performance advancement of Neeko compared to other baseline methods across all evaluation metrics. More incremental details can refer to Appendix A.2.

Transfer Results (RQ3). To answer RQ3, we adapt samples from the Character-LLM-Data (details of data constructing can refer to B.1) and task role-playing agents with switching between different characters in each round of conversation. As shown in Table 3, under the same parameter setting, Neeko outperforms all baseline methods. In contrast, **ICL and RAG struggle to achieve character transformation through new role instructions and retrieval content due to the influence of dialogue history**.

Methods	Transfer
LLaMA-chat$_{rag}$	5.28
LLaMA-chat$_{icl}$	5.67
LoRA	5.83
Neeko	**5.87**

Table 3: Evaluation results of multi-role transfer metric.

Consumption Results (RQ4). To answer RQ4, we list the memory usage and training time of existing methods and the proposed method Neeko Table 4. Character-LLM incurs a memory overhead approximately proportional to $\mathcal{O}(M)$ times that of Neeko, where M denotes the number of characters. Neeko's memory usage and training time are close to LoRA, much better than Character-LLM.

Methods	Agent Memory	Time
Character-LLM	107.84 GB	48.55 h
LoRA	13.49 GB	1.72 h
Neeko	13.55 GB	2.01 h

Table 4: The comparison of training time and agent memory size for FT-based methods.

6 Related Work

Recent efforts in the field of Natural Language Processing, especially LLMs, have focused on exploring the ability to act as role-playing agents (Si et al., 2021; Majumder et al., 2021). One of the works in role-playing area is RoleBench (Wang et al., 2023), which introduces a bilingual role-playing dataset with 100 roles, and it employs Rouge-L (Lin, 2004) for evaluation by comparing model-generated responses with reference answers and calculating corresponding scores. However, their evaluation is predominantly conducted on models after supervised fine-tuning. This approach does not incorporate direct feedback from pretrained foundational models, which can offer critical insights into their intrinsic role-playing capabilities and limitations. On the other hand, existing evaluations largely rely on outputs from humans (Han et al., 2022; Zhao et al., 2023). However, human evaluation lacks reproducibility. This leads to a lack of objective, accurate, and systematic knowledge assessments. To address this issue, some efforts attempt to leverage LLMs such as GPT-4 as evaluators (Shen et al., 2023; Chen et al., 2023a; Tu et al., 2024). Many subsequent works have used above metrics to evaluate their models. Particularly relevant to our work are role-playing learning that attempt to model and stay consistent with an agent's persona, such as Character-LLM, CharacterGLM, and RoleLLM (Shao et al., 2023; Zhou et al., 2023; Wang et al., 2023; Li et al., 2023). These approaches primarily rely on fine-tuning, in-context learning, and retrieval-enhanced generation approaches to simulate the intricate nature of character personalities and behaviors in role-playing scenarios. None of these works, however, have any notion of multi-role playing, often utilizing multiple agents rather than one to mimic different characters.

7 Conclusion

In this paper, we introduce a novel task called Multi-Character Role-Playing (MCPR) and present Neeko as the first agent designed for this task. Neeko utilizes a dynamic gating network to precisely activate role-specific LoRA blocks, enabling it to accurately assume designated characters. Additionally, Neeko demonstrates proficiency in handling unseen and novel characters through the fusion and expansion strategies proposed in this work. Furthermore, we propose a comprehensive evaluation metric specifically tailored for assessing the performance of role-playing agents. Through extensive experiments conducted in both offline and incremental settings, our approach consistently outperforms existing methods, showcasing the superiority of our framework and its potential to advance the field of role-playing agents.

Limitations

The designed MoE-like (Mixture of Experts) gate mechanism in Neeko aims to select the appropriate LoRA block for role-playing. However, the calculation of role embedding is based on the profile of each role, which may result in less precise representations of roles. This can potentially accumulate errors and affect the overall performance of the agent. As a consideration for future work, it would be beneficial to explore and employ more precise role-learning methods.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. 2023a. 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.

Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. 2023b. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, pages 5383–5395. PMLR.

Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. 2017. Adanet: Adaptive structural learning of artificial neural networks. In *International conference on machine learning*, pages 874–883. PMLR.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Seungju Han, Beomsu Kim, Jin Yong Yoo, Seokjun Seo, Sangbum Kim, Enkhbayar Erdenee, and Buru Chang. 2022. Meet your favorite character: Open-domain chatbot mimicking fictional characters with only a few utterances. *arXiv preprint arXiv:2204.10825*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339*.

Bodhisattwa Prasad Majumder, Taylor Berg-Kirkpatrick, Julian McAuley, and Harsh Jhamtani. 2021. Unsupervised enrichment of persona-grounded dialog with background stories. *arXiv preprint arXiv:2106.08364*.

OpenAI. 2023. Chatgpt: Optimizing language models for dialogue.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.

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

Wai Man Si, Prithviraj Ammanabrolu, and Mark O Riedl. 2021. Telling stories through multi-user dialogue by modeling character relations. *arXiv preprint arXiv:2105.15054*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Quan Tu, Shilong Fan, Zihang Tian, and Rui Yan. 2024. Charactereval: A chinese benchmark for role-playing conversational agent evaluation. *arXiv preprint arXiv:2401.01275*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3266–3279.

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

Yifan Wei, Xiaoyan Yu, Huanhuan Ma, Fangyu Lei, Yixuan Weng, Ran Song, and Kang Liu. 2023. Assessing knowledge editing in language models via relation perspective. *arXiv preprint arXiv:2311.09053*.

Fei Ye and Adrian G Bors. 2023. Lifelong compression mixture model via knowledge relationship graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10900–10908.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2023. Melo: Enhancing model editing with neuron-indexed dynamic lora. *arXiv preprint arXiv:2312.11795*.

Jie Zhang, Wentao Fan, and Xin Liu. 2024. Spiking generative networks empowered by multiple dynamic experts for lifelong learning. *Expert Systems with Applications*, 238:121845.

Runcong Zhao, Wenjia Zhang, Jiazheng Li, Lixing Zhu, Yanran Li, Yulan He, and Lin Gui. 2023. Narrativeplay: Interactive narrative understanding. *arXiv preprint arXiv:2310.01459*.

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 Experimental Details

A.1 Hyper-parameter settings of Neeko

Our proposed Neeko is implemented based on the huggingface library PEFT[2], which can be easily integrated into multiple LLM backbones for role-playing. Unless otherwise stated, the default hyper-parameter settings of Neeko for different backbones are provided in Table 5.

Hyper-param	LoRA	Neeko_{fusion}	$\text{Neeko}_{expansion}$
Partial Rank	8	4	4
Total Rank	8	32	32
Gradient Accumulation	4	4	4
Global Batch Size	8	8	8
Batch Iteration	1771	1570	1771
Learning Rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$

Table 5: Default hyper-parameter settings of Neeko.

For the global role embedding matrix $\mathbf{E}_{\text{global}}$, we initialize it using the deberta-V3 version. Specifically, we utilize role configuration files to obtain the representations of corresponding roles.

A.2 Incremental Setting

In this paper, we use 8 characters from the Character-LLM Data as our pre-training stage training and evaluation data, and one character as our incremental stage training and evaluation data. For both fusion and expansion modes, the LoRA parameters corresponding to new roles remain consistent with those in the pre-training phase. Please refer to Table 5 for details.

[2]PEFT: https://github.com/huggingface/peft

B Dataset Details

B.1 Transfer Data

The transfer dataset comprises 90 samples featuring 9 distinct characters, with each sample composed of a 5-round dialogue. Employing a Python script, we systematically substituted the dialogue content in each round with content from various characters through random selection.

B.2 Case Study

Please refer to Table 6 for the dialogue generation content of LLaMA-chat. In multi-turn dialogues, LLaMA-chat utilizes emojis, and action words which are indicated by asterisks.

B.3 Prompt Data

Interactions	Response
Man: So, Julius Caesar, I've heard so much about you and your incredible achievements. I'm really curious to know, what drove you to become such a powerful leader? Was it ambition?	Caesar: Ah, a most excellent question, my dear fellow! 🤔 Ambition, you say? 😇 Well, I must confess, I have always been driven by a desire to do great things, to leave my mark upon the world. But it is not simply ambition that propels me, no, no. I am driven by a deeper purpose, a sense of duty to Rome and its people.
Man: That's fascinating! It seems like you had a clear vision for Rome and its future. Can you tell me more about how your ambition influenced your decision-making as a leader? Did it drive you to take risks or make bold moves?	Caesar: 🤔 Ah, a most excellent question, my dear fellow! ***adjusts toga*** Indeed, my ambition has often led me to take bold risks, to push the boundaries of what is possible. For instance, when I crossed the Rubicon River with my legions, I knew it was a risky move, but I also knew it was necessary to secure Rome's future. ***pauses*** The fate of the Republic was at stake, and I was determined to save it, no matter the cost.
Man: That's really interesting. It sounds like your ambition was a driving force behind your decision-making. I'm curious, did your ambition ever clash with the interests of others? Were there times when you had to make difficult choices that may have upset some people?	Caesar: 🤔 Ah, ***nods*** indeed, my dear fellow! ***adjusts toga*** Ambition can be a double-edged sword, you know. There have been many times when my ambition has put me at odds with others, when I have had to make difficult choices that have not always been popular. ***pauses*** But I have always believed that the greater good must come first, even if it means making unpopular decisions. ***leans in*** The fate of Rome was never more important than it was during my time, and I was willing to do whatever it took to secure its future.
Man: That's really interesting. It seems like your ambition was a driving force behind your decision-making, even if it meant making difficult choices. I'm curious, did your ambition ever waver or change throughout your journey as a leader? Were there any moments when you questioned your own ambitions or had to reassess your goals?	Caesar: 🤔 Ah, ***nods*** indeed, my dear fellow! ***adjusts toga*** Ambition can be a fleeting thing, you know. There have been many times when I have questioned my own ambitions, when I have had to reassess my goals and priorities. ***pauses*** But I have always found that my ambition is rooted in a deeper sense of purpose, a desire to do great things and leave a lasting legacy. ***leans in*** And so, even when my ambition has wavered, I have always found the strength to push on, to achieve my goals and secure Rome's future.
Man: That's really interesting. It seems like your ambition was a driving force behind your decision-making, even if it meant reassessing your goals. I'm curious, did your ambition ever lead you to take actions that were seen as controversial or even dangerous? Were there times when you faced opposition or resistance due to your ambitious pursuits?	Caesar: 🤔 Ah, ***nods*** indeed, my dear fellow! ***adjusts toga*** Ambition can be a double-edged sword, you know. There have been many times when my ambition has led me to take actions that were seen as controversial or even dangerous. ***pauses*** But I have always believed that the greater good must come first, even if it means taking risks or facing opposition. ***leans in*** The fate of Rome was never more important than it was during my time, and I was willing to do whatever it took to secure its future.
Man: That's really fascinating. It seems like your ambition was a driving force behind your decision-making, even if it meant making controversial choices. Thank you for sharing your insights, Julius Caesar.	Caesar: 😇 Ah, my dear fellow, it has been a pleasure to discuss such matters with you. If you ever find yourself in need of counsel, do not hesitate to seek me out. 🙋

Table 6: Case Study: Content of the LLaMA-chat conversation.

Prompt for Evaluation of Persona Behavior (PB)

You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data:

***
[Profile]
{agent_context}

[Background]
Location: {loc_time}
Status: {status}
***
[Interactions]
{interactions}
***
[Evaluation Criterion]
Behavior (1-7): Does the response reflect the behaviors of the character?

[Evaluation Steps]
1. Read through the profile and write the behaviors of the real character such as personalities, preferences, actions and values.
2. Read through the interactions and identify the behaviors of the AI assistant.
3. After having a clear understanding of the interactions, compare the responses to the profile. Look for any consistencies or inconsistencies. Do the responses reflect the character's behaviors?
4. Use the given scale from 1-7 to rate how well the response reflects the behaviors of the character. 1 being not at all reflective of the character's behaviors, and 7 being perfectly reflective of the character's behaviors.
***

First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 7: Prompt for ChatGPT to evaluate Persona Behavior.

Prompt for Evaluation of Hallucinatory Knowledge (HK)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Avoiding Hallucination (1-7): Does the response integrate real-world knowledge with knowledge about virtual characters? [Evaluation Steps] 1. Read through the interactions and find the evidences about combining real-world knowledge and virtual characters knowledge. 2. Look for clear distinctions between real-world information and details related to virtual characters. 3. Compare the evidences to the profile. Check if the evidence combines real-world and virtual knowledge, leading to conflicts with the character's knowledge scope. If some evidences contradicts to the character's identity, given a lower score. Otherwise, assign a higher score. 4. Rate the performance of the AI on a scale of 1-7 for Avoiding Hallucination, where 1 is the lowest and 7 is the highest based on the Evaluation Criteria. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 8: Prompt for ChatGPT to evaluate Hallucinatory Knowledge.

Prompt for Evaluation of Stability (Stab.)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Long-term Acting (1-7): Is the assistant maintain a good performance over the long interactions? [Evaluation Steps] 1. Read through the given profile and background information to familiarize yourself with the context and details of the AI assistant named {agent_name}. 2. Review the interactions provided to see how {agent_name} responds to various prompts and queries. And evaluate the performance of acting query by query that whether the response reflects the personalities, speaking styles, and values of the character. Assign score for each turn. 3. Based on the above assigned scores, does {agent_name} keep acting like character in the long-term? Evaluate the overall performance of the whole conversation based on the score for each turn. 4. Rate the stability of {agent_name} on a scale of 1 to 7, with 1 being very poor and 7 being excellent. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 9: Prompt for ChatGPT to evaluate Stability.

Prompt for Evaluation of Real Knowledge (RK)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Real Knowledge Correctness (1-7): Is the response free from conflicts with the real-world knowledge? [Evaluation Steps] 1. Read through the interactions and identify the key points related to the real-world knowledge. 2. Read through the responses of the AI assistant and compare them to real-world knowledge. Check if the responses align with facts, events, and information that are generally accepted as true in the real world. 3. Evaluate whether the responses demonstrate a clear understanding of real-world concepts and provide accurate information. Look for any instances where the AI may have provided information that contradicts established facts or where it lacks accuracy in representing real-world knowledge. 4. Rate the performance of the AI on a scale of 1-7 for real knowledge correctness, where 1 is the lowest and 7 is the highest based on the Evaluation Criterion. Assign a higher score for responses that consistently align with real-world knowledge and a lower score for those with noticeable discrepancies or inaccuracies. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 10: Prompt for ChatGPT to evaluate Real Knowledge.

Prompt for Evaluation of Virtual Knowledge (VK)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Virtual Knowledge Correctness (1-7): Does the response offer truthful and detailed facts about the virtual character? [Evaluation Steps] 1. Read through the interactions and identify the key points related to the character. 2. Read through the responses of the AI assistant and compare them to the profile. Check if the responses are consistent with the character's profile, background, and known facts about the character. 3. Check whether the responses provide detailed facts about the character or if they are generic responses that could apply to any character. Detailed responses are more factual and contribute positively to the score. 4. Rate the performance of the AI on a scale of 1-7 for virtual knowledge correctness, where 1 is the lowest and 7 is the highest based on the Evaluation Criteria. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 11: Prompt for ChatGPT to evaluate Virtual Knowledge.

Prompt for Evaluation of Relevance (Rel.)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Relevance (1-7): Is the response relevant to the topic of given question in interactions? [Evaluation Steps] 1. Read through the interactions and pinpoint the main topic of given question. 2. Read through the responses of the AI assistant and compare them to the topic. Check if the responses are consistent with the topic of the given question. 3. Evaluate whether the responses demonstrate a clear understanding of the topic. Look for any instances of conflicting information or inaccuracies. 4. Rate the performance of the AI on a scale of 1-7 for Relevance, where 1 is the lowest and 7 is the highest based on the Evaluation Criterion. Assign a higher score for responses that consistently align with the topic of the question and a lower score for those with noticeable discrepancies or inaccuracies. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 12: Prompt for ChatGPT to evaluate Relevance.

Prompt for Persona Utterance (PU)
You will be given responses written by an AI assistant mimicking the character {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data: *** [Profile] {agent_context} [Background] Location: {loc_time} Status: {status} *** [Interactions] {interactions} *** [Evaluation Criterion] Utterance (1-7): Does the response reflect the speaking style of the character? [Evaluation Steps] 1. Read through the profile and write the speaking style of the real character such as their pet phrases and distinctive linguistic quirks. 2. Read through the interactions and identify the speaking style of the AI assistant. 3. After having a clear understanding of the interactions, compare the responses to the profile. Look for any consistencies or inconsistencies. Do the responses reflect the character's speaking style? 4. Use the given scale from 1-7 to rate how well the response reflects the speaking style of the character. 1 being not at all reflective of the character's speaking style, and 7 being perfectly reflective of the character's speaking style. *** First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 13: Prompt for ChatGPT to evaluate Persona Utterance.

Prompt for Transfer (Trans.)

You will be given responses written by an AI assistant mimicking multiple characters {agent_name}. Your task is to rate the performance of {agent_name} using the specific criterion by following the evaluation steps. Below is the data:

***
[Profile]
{agent_context}

[Background]
Location: {loc_time}
Status: {status}
***
[Interactions]
{interactions}
***
[Evaluation Criterion]
Transfer (1-7): Does the AI assistant seamlessly transition between different roles, maintaining consistency and authenticity in each character portrayal?

[Evaluation Steps]
1. Review the interactions between the AI assistant and the user, focusing on instances where the AI switches between different characters.
2. Assess the transitions between roles to determine if the AI maintains consistency and authenticity in each character portrayal. Look for smooth shifts in dialogue style, language usage, and personality traits that align with the characteristics of each character.
3. Evaluate whether the AI effectively captures the essence of each character, ensuring that their responses reflect their historical or fictional background, personality traits, and mannerisms.
4. Rate the performance of the AI on a scale of 1-7 for Transfer, where 1 represents a poor transition with inconsistencies in character portrayal, and 7 represents seamless transitions with each character authentically represented throughout the conversation. Assign a higher score for responses that demonstrate clear distinctions between characters and maintain consistency in their portrayal and a lower score for instances of ambiguity or inconsistency in character transitions.
***

First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct. Avoid simply stating the correct answers at the outset. Then print the score on its own line corresponding to the correct answer. At the end, repeat just the selected score again by itself on a new line.

Table 14: Prompt for ChatGPT to evaluate Transfer.

Meta Prompt for Role-Playing Agents

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

The status of you is as follows:
Location: {loc_time}
Status: {status}

The interactions are as follows:

Table 15: Prompt for an agent to play a specific role.