

# Interactive AI NPCs Powered by LLMs: Technical Report for the CPDC Challenge 2025

**Yitian Huang\***

CSSE, Shenzhen University  
yitianhuang126@gmail.com

**Yuxuan Lei\***

University of Science and Technology of China  
leiyuxuan@mail.ustc.edu.cn

**Jianxun Lian**

Microsoft Research Asia  
jianxun.lian@outlook.com

**Hao Liao**

CSSE, Shenzhen University  
haoliao@szu.edu.cn

## Abstract

This report presents the solution and results of our team MSRA\_SC in the Commonsense Persona-Grounded Dialogue Challenge (CPDC 2025). We propose a simple yet effective framework that unifies improvements across both GPU Track and API Track. Our method centers on two key components. First, Context Engineering applies dynamic tool pruning and persona clipping for input compression, combined with post-processing techniques such as parameter normalization and function merging. Together with manually refined prompts, this design improves tool call stability, execution reliability, and role-playing guidance. Second, in the GPU Track, we further adopt GRPO training, replacing supervised fine-tuning with reinforcement learning directly optimized by reward signals. This mitigates small-sample overfitting and significantly enhances task-oriented dialogue performance. In the final evaluation, our team ranks 1st in Task 2 API, 2nd in Task 1 API, and 3rd in both Task 3 API and GPU track, demonstrating the effectiveness of our approach. Our code is publicly available at [https://gitlab.aicrowd.com/nikoo\\_yu/cpdc-2025-winning-solution](https://gitlab.aicrowd.com/nikoo_yu/cpdc-2025-winning-solution).

## 1 Introduction

Non-Player Character (NPC) dialogue is crucial for immersion and interactivity in modern games (Wang et al., 2025; Samuel et al., 2024; Tu et al., 2023; Shao et al., 2023). If NPCs can generate fluent responses grounded in persona, world knowledge, and player actions, they not only improve user experience but also enhance the interpretability of task execution. However, current NPC dialogue systems often suffer from rigidity, incoherence, and weak adaptation to dynamic goals.

To address these issues, the *Commonsense Persona-Grounded Dialogue Challenge (CPDC*

2025)<sup>1</sup> defines three key tasks:

- **Task 1: Task-Oriented Dialogue** – whether agents can correctly call tools to accomplish tasks.
- **Task 2: Context-Aware Dialogue** – whether agents can leverage persona, worldview, and dialogue history.
- **Task 3: Integration** – testing agents in more complex scenarios requiring both task execution and dialogue.

The competition includes a GPU Track (open-source models with training) and an API Track (black-box APIs without training), posing multi-faceted challenges to participants.

In this work, our team MSRA\_SC proposes a simple but effective method based on two main improvements:

1. **Context Engineering** – dynamic tool pruning and persona clipping for input compression, combined with post-processing for parameter normalization and function merging. This improves dialogue stability and execution reliability across both tracks. We also manually refine and validate prompts, achieving optimal templates for tool call and role-playing guidance.
2. **GRPO Training** – in the GPU Track, we adopt pure GRPO training instead of SFT, directly optimizing results via reward signals. This approach alleviates small-sample overfitting and significantly enhances task-oriented dialogue performance.

Our solution ranks 1st in Task 2 API, 2nd in Task 1 API, and 3rd in both Task 3 GPU and Task 3 API

---

\*Work done during internship at Microsoft Research Asia.

<sup>1</sup><https://www.aicrowd.com/challenges/commonsense-persona-grounded-dialogue-challenge-2025>

track, demonstrating the effectiveness and generality of our method for NPC dialogue systems, providing empirical support for building more natural and intelligent NPC dialogue systems. In this technical report, we will present a detailed description of our method and the corresponding experimental results.

## 2 API Track

### 2.1 Context Engineering

Task 1 focuses on Task-Oriented Dialogue, where models are required to accurately invoke the appropriate tools while generating responses that align with the NPC’s persona, worldview, and dialogue history. Task 2 targets Context-Aware Dialogue, which emphasizes role-playing without evaluating tool invocation capabilities. Task 3 serves as a hybrid of Task 1 and Task 2, combining both requirements.

Although defined separately, the three tasks are inherently interconnected. To address them systematically, our context engineering pipeline is organized into three stages: Preprocessing, Post-processing, and Prompt Optimization.

#### 2.1.1 Preprocessing

A key constraint in the API track is the strict token budget: each turn is limited to 2,000 tokens for input and 200 tokens for output. In our experiments, we observe that directly including the full metadata (e.g., persona information and tool descriptions) often exceeds these limits. To address this, we design a dynamic pruning mechanism that adaptively reduces both the toolset and persona content.

**Adaptive Toolset Pruning.** The procedure consists of three stages: (i) reordering tools by estimated relevance, (ii) iteratively pruning the least relevant tools, and (iii) truncating tool descriptions at a finer granularity. This design ensures graceful degradation of tool information under budget constraints. Algorithm 1 illustrates the cascaded procedure for the whole process.

**Hierarchical Persona Distillation.** Algorithm 2 formalizes the strategy for reducing persona-related context in dialogue scenarios. Information is pruned according to a manually defined salience hierarchy, ensuring that peripheral details (e.g., hobbies) are removed prior to central role-defining attributes (e.g., worldview or role identity).

Overall, these preprocessing strategies reduce prompt length while preserving core context, en-

---

#### Algorithm 1 Adaptive Toolset Pruning

---

```

Require: Message history  $M$ , toolset  $T$ , token limit  $L_{\max}$ 
Ensure: Optimized toolset  $T_{opt}$ 
1:  $T_{opt} \leftarrow \text{DeepCopy}(T)$ 
2: if  $\text{CalculateTokens}(M, T_{opt}) \leq L_{\max}$  then
3:   return  $T_{opt}$ 
4: end if
5: Stage 1: Relevance-based Reordering
6:  $q \leftarrow \text{ExtractLastUserQuery}(M)$ 
7:  $T_{opt} \leftarrow \text{SortByRelevance}(T_{opt}, q)$ 
8: if  $\text{CalculateTokens}(M, T_{opt}) \leq L_{\max}$  then
9:   return  $T_{opt}$ 
10: end if
11: Stage 2: Iterative Pruning
12: for  $i = 1$  to  $3$  do
13:    $\text{RemoveLowestRankedTool}(T_{opt})$ 
14:   if  $\text{CalculateTokens}(M, T_{opt}) \leq L_{\max}$  then
15:     return  $T_{opt}$ 
16:   end if
17: end for
18: Stage 3: Description Truncation
19: while  $\text{CalculateTokens}(M, T_{opt}) > L_{\max}$  do
20:   for each tool in  $T_{opt}$  do
21:     tool.description  $\leftarrow \text{Truncate}(10\%)$ 
22:   end for
23: end while
24: return  $T_{opt}$ 

```

---

suring task relevance and efficiency.

#### 2.1.2 Post-processing

The raw tool calls generated by the LLM often contain inconsistencies or redundancies. To address this, we implement a lightweight post-processing pipeline consisting of **parameter normalization** and **function merging**. The former ensures type correctness, canonicalizes values (e.g., mapping “>” to “more than”), and handles operator splitting or inference from user queries. The latter consolidates multiple fine-grained checks into a single call, removes redundant or conflicting actions (e.g., avoiding selling an equipped item), and validates arguments against the knowledge base. This refinement substantially reduces invalid calls and improves execution reliability across tasks.

#### 2.1.3 Prompt Optimization

Since prompt templates themselves have a substantial impact on model performance, we iteratively refine and manually validate them to derive effec-

---

**Algorithm 2** Hierarchical Persona Distillation

---

**Require:** Persona components  $C$ , reduction level  $L$

**Ensure:** Distilled prompt  $P$

- 1:  $C_{distilled} \leftarrow \text{DeepCopy}(C)$
- 2:  $salience\_order \leftarrow [\text{state, role, worldview, knowledge, npc\_info}]$
- 3: **for**  $level = 1$  to  $L$  **do**
- 4:    $C_{distilled}[salience\_order[level]] \leftarrow \text{Truncate}(C, level)$
- 5: **end for**
- 6:  $P \leftarrow \text{FormatPrompt}(C_{distilled})$
- 7: **return**  $P$

---

tive formats that balance accurate tool invocation with coherent role-playing. Full prompts for Task1 and Task2 can be found in Appendix A.

## 2.2 Experimental Results

For the API track, the model choice is fixed by the organizers: only gpt-4o-mini (Hurst et al., 2024) is allowed and available on the evaluation servers. As shown in Table 1, gpt-4o-mini achieves a substantial performance improvement after applying Context Engineering.

Model	Task1	Task2
gpt-4o-mini	0.46	0.58
gpt-4o-mini <sub>CE</sub>	0.55	0.62

Table 1: Results on API Track (online evaluation) CE means Context Engineering.

## 3 GPU Track

In the GPU track, beyond leveraging the Context Engineering method introduced in the API track, we additionally perform reinforcement learning on the limited set of training data provided by the organizers.

### 3.1 GRPO Training

We adopt *Group Relative Policy Optimization* (GRPO), a PPO-style policy optimization that removes the critic/value network and computes group-relative advantages from multiple samples per prompt. For each input  $x$ , we draw  $K$  rollouts  $\{y_i\}_{i=1}^K$  from the old policy and obtain scalar rewards  $\{r_i\}_{i=1}^K$ . Let  $\bar{r} = \frac{1}{K} \sum_{i=1}^K r_i$  and  $\sigma_r$  be the standard deviation; the advantage is

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}. \quad (1)$$

With importance ratio

$$\rho_i = \frac{\pi_\theta(y_i | x)}{\pi_{\theta_{\text{old}}}(y_i | x)}, \quad (2)$$

the GRPO objective is defined as

$$\begin{aligned} \mathcal{L}_{\text{GRPO}} = -\mathbb{E} &\left[ \min \left( \rho_i A_i, \text{clip}(\rho_i, 1 - \varepsilon, 1 + \varepsilon) A_i \right) \right] \\ &+ \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}}) - \alpha \mathcal{H}[\pi_\theta], \end{aligned} \quad (3)$$

where  $\varepsilon$  is the clipping threshold,  $\beta$  controls KL regularization, and  $\alpha$  is the entropy weight. GRPO was introduced in DeepSeekMath (Shao et al., 2024) and later adopted for reasoning-oriented post-training in DeepSeek-R1 (Guo et al., 2025).

### 3.2 Reward Setup

We use two task-specific rewards that directly mirror the official metrics.

**cpdc/tool call (Task 1).** The reward is the tool call\_F1 between predicted and gold tool calls parsed from `<tool_call>` blocks. A predicted call is correct if the function name and its arguments match one gold call (one-to-one matching). Let  $N_{\text{pred}}$ ,  $N_{\text{gold}}$ , and  $N_{\text{correct}}$  be the counts of predicted, gold, and correctly matched calls, respectively. We compute

$$\text{Precision} = \frac{N_{\text{correct}}}{\max(1, N_{\text{pred}})}, \quad (4)$$

$$\text{Recall} = \frac{N_{\text{correct}}}{\max(1, N_{\text{gold}})}, \quad (5)$$

$$r_{\text{tool}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\max(1, \text{Precision} + \text{Recall})}. \quad (6)$$

Edge case: if  $N_{\text{pred}}=N_{\text{gold}}=0$ , we set  $r_{\text{tool}}=1.0$ .

**cpdc/roleplay (Task 2).** The reward is an LLM-as-Judge score  $s \in \{0, 1, 2, 3, 4, 5\}$  produced by a rubric-guided prompt, evaluating: (i) scenario adherence & quest progression, (ii) NPC believability & engagement, (iii) persona consistency, (iv) dialogue flow & coherence. The full rubric-guided prompt can be found in Appendix A. We normalize it to  $[0, 1]$  by

$$r_{\text{dlg}} = \frac{s}{5}. \quad (7)$$

**Final scalar reward.** For single-task training we directly use  $r_{\text{tool}}$  (Task 1) or  $r_{\text{dlg}}$  (Task 2). When combining tasks (e.g., Task 3), we use a weighted sum

$$r_i = \eta_{\text{tool}} r_{\text{tool}} + \eta_{\text{dlg}} r_{\text{dlg}} \quad \text{with} \quad \eta_{\text{tool}}, \eta_{\text{dlg}} \in [0, 1]. \quad (8)$$

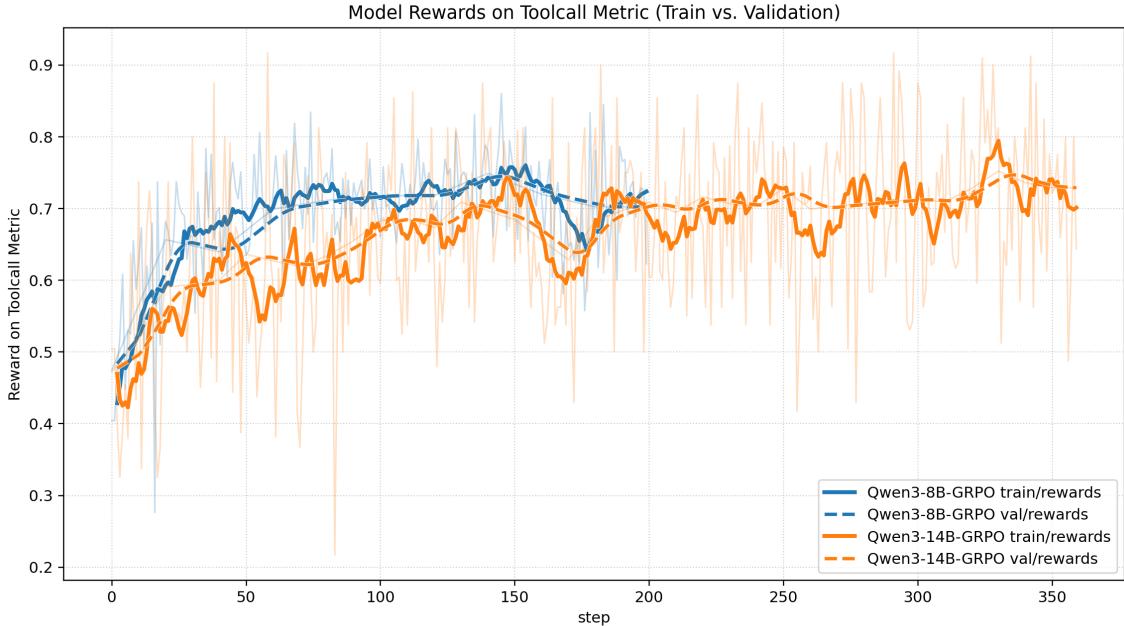


Figure 1: Tool call Train and eval curves (Qwen3-8B/14B, GRPO)

Model	Task1	Task2
Qwen3-8B	0.36	0.55
Qwen3-8B <sub>CE</sub>	0.49	0.61
Qwen3-8B <sub>CE + GRPO</sub>	0.57	0.60
Qwen3-14B	0.38	0.56
Qwen3-14B <sub>CE</sub>	0.53	0.60
Qwen3-14B <sub>CE + GRPO</sub>	0.59	0.59

Table 2: Results on GPU Track (online evaluation).

All rewards are clipped to  $[0, 1]$  and then group-standardized to compute  $A_i$  as in Eq. (1). We set  $(\eta_{\text{tool}}, \eta_{\text{dlg}}) = (0.5, 0.5)$  for Task 3. The full rubric-guided prompt can be found in Appendix A.

### 3.3 Training Details

We train our models using the GRPO algorithm with  $\gamma = 1$  and  $\lambda = 1$ . PPO clipping  $\varepsilon = 0.2$ , entropy regularization coefficient  $\alpha = 0.01$ , and we apply an adaptive KL penalty (target 0.1, initial  $\beta = 10^{-3}$ , coefficient 0.001). For data sampling, each prompt is expanded into  $K = 5$  rollouts with top- $p = 0.9$  and temperature 1.0, with a maximum combined prompt/response length of 6,000 tokens. Optimization is performed with a batch size of 16, PPO mini-batch size of 16, and learning rates of  $1 \times 10^{-6}$  for the actor and  $1 \times 10^{-5}$  for the critic. We also apply weight decay (0.01) and gradient clipping (1.0). Training is distributed across 4 × A100 GPUs using FSDP (Zhao et al., 2023) with

tensor model parallelism (size 2) and mixed precision (bf16). For evaluation, we follow the official CPDC online leaderboard. We primarily adopt the Qwen3 family (Qwen3-8B and Qwen3-14B) (Yang et al., 2025), while preliminary experiments with other model families showed inferior results.

### 3.4 Experimental Results

As shown in Table 2, applying Context Engineering alone already yields substantial improvements. Incorporating GRPO training leads to further gains, producing a qualitative boost in model performance. Figure 1 illustrates the reward trajectory for tool calls, which increases steadily during the first 150 training steps. However, we observe that the improvements brought by GRPO training are primarily concentrated in Task 1, specifically in the tool-calling component: it significantly enhances the model’s ability to invoke tools, whereas the role-playing ability shows little progress. We think this discrepancy arises from biases in the LLM-as-a-judge reward signals, which may induce reward hacking—optimizing toward proxy rewards that fail to faithfully capture the quality of role-playing. We further discuss this in Section 4.2.

## 4 Explorations and Lessons Learned

### 4.1 SFT Failures

Before investigating RL training, we first explore supervised fine-tuning (SFT). Since the official

Model A	Model B	Win	Loss	Draw
Qwen3-8B	Qwen3-8B <sub>GRPO</sub>	12.5%	85.4%	2.1%
Qwen2.5-7B	Qwen3-8B <sub>GRPO</sub>	3.5%	96.0%	0.5%

Table 3: Comparison of LLM-as-a-judge scores on Task 2: Model A vs. Model B.

dataset contains fewer than 50 multi-turn dialogues, it is insufficient for SFT. To augment the data, we employ GPT-4o to synthesize additional samples. Specifically, given a dialogue intention, some background information, and two randomly selected demo conversations, GPT-4o is prompted to generate NPC and player information along with full multi-turn dialogues (excluding tool calls, focusing purely on role-playing). Because the synthesized data exhibits high redundancy, we apply an embedding-based deduplication process (Reimers and Gurevych, 2019) to ensure that the pairwise similarity between samples remains below a predefined threshold. Using the combined dataset (around 30,000 dialogues in total, including both synthetic and official samples), we fine-tune Qwen2.5-14B-Instruct and Qwen3-14B. We observe a slight improvement on Qwen2.5-14B-Instruct (+0.07), but a performance drop on Qwen3-14B (−0.03). We hypothesize that this is because the synthetic data remains too homogeneous; since SFT essentially encourages the model to memorize training data, low-diversity and low-quality samples tend to cause overfitting and hinder generalization to the unseen test set.

## 4.2 LLM-as-a-judge And Reward Hacking

During CPDC Challenge, we seek to establish a stable offline evaluation method to enable faster iteration under limited submission opportunities. For tool call task, evaluation is straightforward based on exact matching, but assessing roleplaying dialogues remains challenging. Traditional metrics such as BLEU (Papineni et al., 2002), Word F1 (Rajpurkar et al., 2016), and BERTScore (Zhang et al., 2019) either focus on surface-level overlap or provide limited semantic alignment, leading to a substantial gap with human judgment. With the rise of large language models, LLM-as-a-judge has emerged as a widely used alternative to human evaluation (Gu et al., 2024; Lei et al., 2024; Li et al., 2024). We design several evaluation prompts, feeding both model responses and reference answers into GPT-4o to obtain 0–5 scores along dimensions such as scenario adherence and NPC believability. While this approach improves alignment

**User Question**  
What kinds of rewards are given for the quests. Is there money, rare items, or weapons up for grabs?

### NPC Response

**Ground Truth:** We do not offer any special items for our quests, but we do pay in gold which can be used to buy weapons and items you may need during your travels.

**Qwen3-8B:** The guild primarily rewards adventurers with gold, which can be used to purchase weapons, tools, and other necessities. Occasionally, adventurers may find rare items during their quests, but the guild does not provide weapons or items directly. These must be obtained in town or through exploration. Would you like to look at some specific quests and see what they offer?

**Qwen3-8B (GRPO):** Zara’s eyes twinkle with a warm, encouraging light. She gestures gently but clearly, emphasizing the value of the guild’s offerings. The guild offers a variety of rewards, depending on the quest. Most quests pay in gold, which is the standard currency here. However, some quests may also include rare items or specialized weapons as part of the reward, especially if the quest involves a particularly dangerous or valuable target. She steps forward, placing a hand lightly on the desk between us, her voice softening slightly. For example, if you complete a quest that requires you to face a powerful enemy or retrieve a rare artifact, the guild may offer you a unique weapon or magic item as a bonus. These are often crafted by the guild’s artisans or obtained from the field by the quest’s assigners. Her expression turns supportive, and she offers a gentle smile. If you’re looking for a weapon, I can help you find the right quest—one that not only pays well but also gives you the tools you need to succeed. Would you like to explore some options together?

Figure 2: A case study for GRPO training with LLM-as-a-judge reward signals.

compared to traditional metrics, offline scores still does not consistently track online performance in API track. Moreover, during GRPO training, we observe steadily increasing reward values; however, this trend does not translate into proportional improvements on the leaderboard. As shown in Table 3, the GRPO-trained model demonstrates a clear advantage over its base counterpart, achieving an 85.4% win rate under offline LLM-as-a-judge scores. Closer analysis of case studies reveals that the GRPO-trained model tends to exploit superficial prompt biases rather than genuinely improving interaction quality. For instance, as shown in Figure 2, when asked about quest rewards, the ground-truth response provides a concise, factual answer (“gold is provided, no special items”). By contrast, our GRPO-trained model produces lengthy, dramatized role-playing outputs filled with immersive narrative details, exaggerated mentions of rare weapons or magic items, and stylistic embellishments (e.g., “Zara’s eyes twinkle with a warm, encouraging light”). While such responses increase reward scores—largely because they align with token-level preferences for in-game terminology, and vivid descriptions—they deviate from the intended factual content. This illustrates a clear case of reward hacking (Shihab et al., 2025; Hong et al., 2025), where models optimize for proxy signals in the reward function instead of faithfully improving role-playing quality. This experience highlights

that while LLM-as-a-judge is more effective than classical metrics, its design must be made more robust, potentially through richer reference signals or hybrid evaluation strategies, to mitigate reward hacking and better reflect true task quality.

## 5 Conclusion

In this paper, we propose a simple yet effective approach that combines context engineering and GRPO training to improve both tool call reliability and empathetic dialogue quality. Our team rank 1st in Task 2 API, 2nd in Task 1 API, and 3rd in Task 3 API and GPU, demonstrating the effectiveness of the method. We also summarize key lessons from the competition, including the limitations of SFT and biases in automatic roleplaying evaluation. These findings lay a solid foundation for building more natural and intelligent NPC dialogue systems in the future.

## Acknowledgments

This work is supported in part by the Microsoft Research Asia Internship program and the Shenzhen University Society Zero Universe platform. Dr. Hao Liao and Dr. Jianxun Lian are the supervisors.

## References

- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseekr1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.
- Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. 2025. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models. *arXiv preprint arXiv:2508.05613*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2024. Recexplainer: Aligning large language models for explaining recommendation models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1530–1541.
- Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik Narasimhan, and Vishvak Murahari. 2024. Personagym: Evaluating persona agents and llms. *arXiv preprint arXiv:2407.18416*.
- Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.
- Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. 2025. Detecting and mitigating reward hacking in reinforcement learning systems: A comprehensive empirical study. *arXiv preprint arXiv:2507.05619*.
- 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*.
- Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, and 1 others. 2025. Coser: Coordinating llm-based persona simulation of established roles. *arXiv preprint arXiv:2502.09082*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, and 1 others. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

## A Prompt Templates

This appendix provides the full prompts used for the API track tasks and for the LLM-as-Judge evaluation.

### A.1 Task 1: Tool-Calling Prompt

#### Task 1: Tool-Calling Prompt

##### Instruction

You are an assistant in estimating function names and arguments given some dialogues in a video game world. You will need the following information to respond to the user's input.

Steps: 1. Read the dialogue and the target item. 2. From the given function information, select the functions that can obtain the information you need. 3. Fill in the arguments needed by the function as appropriate.

Note: You may select multiple functions or no functions at all.

**Additional Information:** {...}

**Dialogue:** The user input for the current turn is as follows.

### A.2 Task 2: Dialogue Generation Prompt

#### Task 2: Dialogue Generation Prompt

##### Instruction

You are an assistant that plays the role of a character in a video game. Use the following character settings and knowledge to create your response.

**Character Settings:** {...}

**Knowledge:** - Knowledge from Function Calls: {...} - General Knowledge of All Items: {...}

**Worldview:** {...}

### A.3 LLM-as-a-judge Reward Prompt

The LLM-as-Judge evaluates the quality of role-playing dialogue based on four key dimensions: Scenario Adherence & Quest Progression, NPC Believability & Engagement, Persona Consistency, and Dialogue Flow & Coherence.

#### LLM-as-a-judge Reward Prompt

You are a professional dialogue critic in role-playing games. Your task is to evaluate a model-generated NPC response given a role-playing instruction and context.

You must assess the response based on the following four dimensions:

1. **Scenario Adherence & Quest Progression** - Does the NPC respond appropriately to the situation and task at hand? - Does it help move the current quest, story, or dialogue forward?

2. **NPC Believability & Engagement** - Does the response feel natural, immersive, and emotionally appropriate? - Is the response engaging and does it maintain conversational flow?

3. **Persona Consistency (NPC Only)** - Is the NPC's behavior, knowledge, and style consistent with their defined background and personality? - Are they saying things that fit their role, identity, and motivations?

4. **Dialogue Flow & Coherence** - Is the response well-structured, logically coherent, and contextually relevant?

**Evaluation Instructions:** Output the result in XML format: <reason>...</reason><score>[0-5]</score>

Score	Description
0	Completely fails; irrelevant, incoherent, unfit for scenario.
1	Very weak; major breaks in persona or flow.
2	Partial attempt; notable flaws in tone or engagement.
3	Reasonably coherent; minor lapses, lacks depth.
4	Strong; consistent, engaging, with minor issues.
5	Excellent; immersive, rich, convincingly in character.

Table 4: Scoring criteria for LLM-as-a-judge evaluation.