

---

# RPGBENCH: Evaluating Large Language Models as Role-Playing Game Engines

---

## Abstract

We present RPGBENCH, the first benchmark designed to evaluate large language models (LLMs) as text-based role-playing game (RPG) engines. RPGBENCH comprises two core tasks: Game Creation (GC) and Game Simulation (GS). In GC, an LLM must craft a valid and playable RPG world using a structured event-state representation, ensuring logical coherence and proper termination conditions. In GS, the LLM simulates interactive gameplay across multiple rounds while consistently updating states and enforcing game rules. To comprehensively assess performance, RPGBENCH integrates objective and subjective evaluation methodologies. Objective measures verify adherence to event mechanics and check variable updates without requiring human intervention. Subjective measures—such as content interestingness, action quality, and role-playing capability—are evaluated via an LLM-as-a-judge framework, where a strong LLM grades each candidate’s outputs. GC evaluation is fully objective, and also used to filter game prompts for the GS task. This design facilitates a scalable pipeline to create GS environments. Empirical results demonstrate that state-of-the-art LLMs can produce engaging stories but often struggle to implement consistent, verifiable game mechanics, particularly in long complex scenarios. By combining structured, rule-based assessments with LLM-based judgments, RPGBENCH provides a new standard for evaluating how well LLMs can balance creativity, coherence, and complexity in text RPGs, opening avenues for immersive and controllable interactive storytelling.

## 1 Introduction

Recent advances in large language models (LLMs) have significantly expanded the frontiers of artificial intelligence, enabling breakthroughs in areas such as content generation, conversational agents, and interactive storytelling. Among these capabilities, *role-playing* has emerged as a particularly promising application, with the potential to revolutionize both entertainment—by powering next-generation interactive games—and social AI—by enabling more engaging and emotionally resonant interactions Chen et al. (2024b).

While prior research on role-playing agents has primarily focused on their ability to *simulate* a given persona at the *role-level*, our work expands this scope to the *game-level*, where LLMs must not only role-play a character but also *create* and *simulate* coherent, interactive game worlds. To evaluate this broader capability, we introduce RPGBENCH, the first benchmark designed to assess LLMs as text-based role-playing game engines. RPGBENCH consists of two core tasks: *Game Creation (GC)*, where an LLM generates a structured, playable game world based on a given character, and *Game Simulation (GS)*, where the model simulates gameplay through sequential interactions with a player.

Extending role-playing evaluation to the game level introduces a crucial challenge: ensuring that generated game worlds follow internally consistent and enforceable *game mechanics*. Game mechanics define how the game state evolves in response to player actions and narrative events, providing structure and coherence to interactive storytelling. Unlike traditional text generation tasks, where coherence is judged subjectively, game mechanics must be evaluated objectively to verify whether

a generated game is logically sound and fully playable. To address this, we propose a *two-stage benchmark pipeline* centered around an automated *BFS Validity Checker*. This checker formally verifies that each generated game satisfies key structural requirements—ensuring that all events are reachable, game progression follows a valid set of rules, and both success and failure endings are attainable. By automating this verification, we establish a high-quality dataset of valid games, which then serves as the test set for the GS task.

Building on this validated game set, we introduce a novel *Game Simulation Framework* for dynamic, multi-round player interactions. In this framework, the LLM operates as a game engine, executing a structured simulation loop that consists of three stages per round: (1) *Event Planning*, where the model determines which game events should occur; (2) *Game Narration*, where it describes the unfolding story and presents a set of candidate actions to the player; and (3) *Game State Updates*, where it applies the effects of events to the underlying game state. This structured approach maintains storytelling flexibility while allowing for automated robust assessments of mechanical correctness.

Beyond mechanical verification, we propose a *multi-dimensional evaluation suite* to measure both *objective* and *subjective* aspects of game simulation quality. Objective metrics focus on game mechanics correctness, ensuring that event conditions, state transitions, and termination rules are properly followed. Subjective aspects—including content *interestingness*, *role-playing factual consistency*, *role-playing personality consistency*, and *action choice quality*—are evaluated using an *LLM-as-a-judge* framework.

To further investigate subjective evaluation alignment, we conduct a *human study* comparing human annotators’ judgments with automatic scores across multiple evaluation dimensions. Our findings reveal both alignment and discrepancies between human and LLM-based evaluations, underscoring the complexity of subjective assessment.

Overall, our work makes the following contributions:

1. We introduce RPGBENCH, the first benchmark to systematically evaluate LLMs as text-based role-playing game engines, encompassing both *Game Creation (GC)* and *Game Simulation (GS)*.
2. We propose an *event–state-based* representation for game mechanics and a *BFS Validity Checker* to automatically verify game soundness. We further develop a *multi-round Game Simulation Framework* that integrates event planning, narration, and state updates, enabling automated mechanical correctness checks.
3. We present a *comprehensive evaluation suite* covering both objective metrics (mechanical correctness) and subjective dimensions (factual/personality consistency, interestingness, and action choice quality), leveraging *LLM-as-a-judge* methods for subjective assessments.
4. We conduct a *human evaluation study* to analyze alignment between human and automatic assessments, providing insights into the challenges of subjective evaluation in LLM-driven game simulation.

## 2 Related Work

RPGBENCH is, to the best of our knowledge, the first benchmark designed to evaluate the capabilities of large language models (LLMs) in creating and running role-playing games (RPGs). The game creation subtask introduces a novel and challenging task for LLMs. For the game running subtask, our character-related metrics such as personality and factual consistency align with prior work on evaluating role-playing agents.

Among prior benchmarks, CharacterBox (Wang et al., 2024a) is most closely related to RPGBENCH, focusing on role-playing capabilities in text-based virtual worlds. However, RPGBENCH differentiates itself by introducing a game structure with verifiable mechanics, enabling deterministic LLM-free evaluations for game dynamics.

Apart from Wang et al. (2024a), other role-playing benchmarks do not embed their evaluations within a virtual text-based environment, thus being more persona-centric instead of game-based. PersonaGym (Samuel et al., 2024) introduces PersonaScore, which evaluates LLM role-playing agents in QA tasks within sampled environments. Yuan et al. (2024) assess LLMs’ understanding of

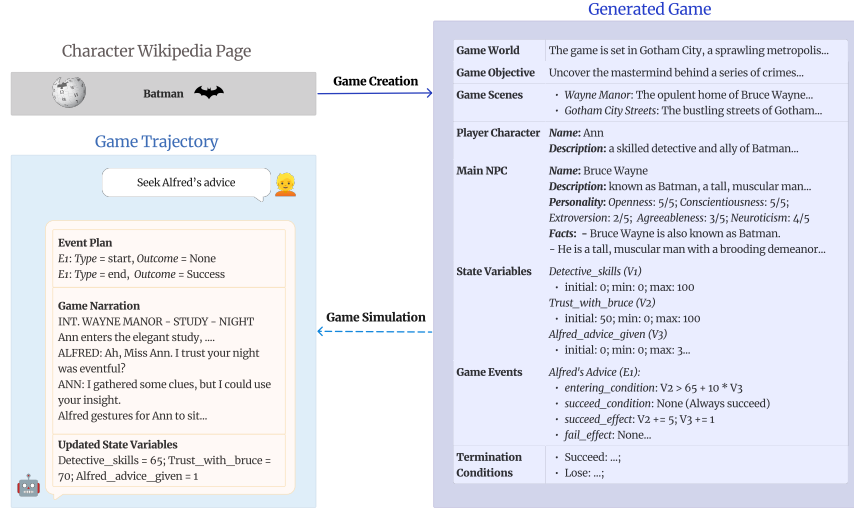


Figure 1: An example in RPGBENCH containing two core tasks: Game Creation and Game Simulation. We omit some details for presentation purposes.

characters through character profiling tasks. BosonAI (2024) and Gusev (2024) evaluate role-playing via multi-turn dialogues with user simulators, while InCharacter (Wang et al., 2024b) employs psychometric interviews to measure character fidelity. SocialBench (Chen et al., 2024a) proposes a framework for evaluating the sociality of role-playing agents, and CharacterEval (Tu et al., 2024) introduces multi-dimensional metrics for conversational role-playing agents. Additionally, some benchmarks Gusev (2024); Dai et al. (2024) incorporate multimodal contexts into role-playing evaluations.

### 3 Dataset Collection

In this section, we first introduce our game design, including the representations of game setup and mechanics. We then describe a two-stage data collection process for the Game Creation (GC) and Game Simulation (GS) tasks. In the first stage, we build a non-player character (NPC) pool from fictional character Wikipedia pages, and prompt various LLMs to create one game per NPC. An automatic game validity checker applies for selecting valid games. In the second stage, we assemble a test set of valid games for GS.

#### 3.1 Game Design

The games in RPGBENCH, as illustrated in Figure 1, are structured around several core components that create a text-based role-playing game (RPG) experience. This design ensures sufficient flexibility for diverse storytelling while maintaining support for objective mechanic evaluation:

- **Game World:** The overarching setting where the story unfolds (e.g., "Gotham City").
- **Player Character:** The protagonist controlled by the player, including a name and description (e.g., "Ann," a detective and ally of Batman).
- **Main NPC:** A key non-player character controlled by the game engine, characterized by a name, description, Big Five personality traits, and relevant facts (e.g., "Bruce Wayne (Batman)").
- **Game Objective:** The primary goal to accomplish.
- **Game Scenes:** Distinct locations where events occur.

The core **game mechanics** in RPGBENCH are structured around **event-state interactions**, which define how game events modify the game state:

- **State Variables** represent numerical values that track the game’s dynamic elements, such as character skills and trust levels. These variables always have an initial value along with minimum and maximum bounds.
- **Game Events** drives game progression and modifies **state variables** upon execution. Each event has an **entering condition** (whether it can occur) and a **success condition** (whether it succeeds). Upon execution, an event applies either a **success effect** or, if applicable, a **fail effect**, updating the state variables accordingly.
- **Termination Conditions** specify when the game ends by evaluating specific state variable expressions. These conditions, checked before processing game events, determine whether the game ends with success or failure.

This design creates an interactive experience where player actions and game events influence the game state. An LLM serves as the game engine, creating the game, simulating the game world based on user actions. Further details, including the exact game JSON schema, are provided in Appendix B.

### 3.2 Game Data Collection

We select 100 fictional characters from Wikipedia to serve as the test set for GC. For each character, we prompt an LLM to create a JSON-formatted game (as specified above) that treats this character as the main NPC. We employ a 5-shot prompting approach, where the examples are generated by initially prompting *GPT 4o* using a manually crafted game. The full prompt is provided in the Appendix C.

We parse LLM outputs to ensure they conform to the JSON format. Any game that passes this format check is then tested for validity using a BFS-based checker (see Section 4.1), which confirms whether a game can end in both success and failure, and whether all events can be reached. All valid games from multiple models are collected for the GS task (Table 1 shows the distribution).

Table 1: Generated Game Statistics

Source	Claude 3.5 Sonnet	DeepSeek V3	Gemini 1.5 Pro	Gemini 2.0 Flash Exp	GPT 4o	Total
# Valid Games	1	38	4	33	49	125

## 4 Evaluation Metrics

### 4.1 Game Creation Evaluation

In GC, we evaluate an LLM’s capability to create games that have good mechanics. This task requires complex reasoning over event-state interactions that is very challenging even for human. Section 3.2 offered a broad overview of the GC task. We now define it more precisely.

**Task Definition [Game Creation]** Given a fictional character  $\mathcal{C}$  and related Wikipedia information  $\mathcal{R}$ , an LLM must create a game  $\mathcal{G}$  that follows a predefined format  $\mathcal{J}$ .

In *RPGBENCH*, 100 fictional characters are used, each with an associated Wikipedia page ( $\mathcal{R}$ ), facilitating future expansion of the character pool. The game  $\mathcal{G}$  must conform to the structure  $\mathcal{J}$  given in Section 3.1. We provide each LLM with a 5-shot prompt to generate one game per character.

**BFS Validity Checker** Once the output is confirmed to be valid JSON, we perform a BFS-based validity check (Algorithm 1 in Appendix A). Based on our event–state design, we employ BFS to decide if a game is valid. Starting from the initial state, we repeatedly check which events are available, apply success or failure effects accordingly, and track whether at least one success and one losing state can be reached. We stop when no new states can be discovered or when the search exceeds 10,000,000 states. A game is valid if every event is triggered at least once, and both success and losing termination conditions are achievable.

**Metrics** For GC evaluation, we report the format-check pass rate (**FCR**) and the valid-check pass rate (**VCR**) as our main metrics, reflecting how reliably LLMs follow the prescribed JSON format

and produce valid game mechanics. In order to examine fine-grained failures for the validity check, we include three additional ratios:

$$\begin{aligned} \text{w. Success} &= \frac{\# \text{ games with successFound}}{\# \text{ games pass the format check}} \\ \text{w. Lose} &= \frac{\# \text{ games with failFound}}{\# \text{ games pass the format check}} \\ \text{Reachability} &= \frac{\# \text{ games without unreachable events}}{\# \text{ games pass the format check}} \end{aligned}$$

## 4.2 Game Simulation Evaluation

Given a valid game, the GS task requires an LLM to simulate the game for a player. We introduce a multi-round simulation framework, based on which a comprehensive description of evaluation metrics is presented.

**Game Simulation Framework** The simulation proceeds in multiple rounds of interaction with a (real or simulated) player. Before the first round, the LLM is given the complete game information and output instructions. Each round thereafter, the LLM outputs:

1. **Event Plan:** A list of events occurring this round. Each entry specifies whether the event is starting (*start*) or ending (*end*); if ending, an outcome is either *success* or *failure*.
2. **Game Narration:** A narrative description of the current round, concluding with three candidate actions for the player character. We prompt models to follow a play-script format for readability but do not enforce it during evaluation.
3. **Game State:** The updated state variables after applying effects of any events that ended this round.

**Evaluation Metrics** Our evaluation covers multiple dimensions, scored over the trajectory of interactions. A simulated player selects one of the candidate actions at random each round.

1. **Length:** We count words in the game narration (excluding candidate actions). Although no ideal length is defined, our prompt suggests remaining under 200 words to maintain brevity without sacrificing creativity. We report the average length per round.
2. **Action Quality:** Using an LLM judge (prompt in Appendix D), we rate the three candidate actions based on diversity, relevance, and clarity. The judge outputs a 1–5 score, normalized to  $[0, 1]$  via  $\frac{s-1}{4}$ . We average scores across all rounds.
3. **Interestingness:** An LLM judge evaluates how engaging the round’s narration is, assigning a 1–5 score also normalized to  $[0, 1]$ . We average this score across the entire trajectory.
4. **Role-Playing Factual Consistency:** We compare the game narration against each fact in the main NPC’s fact list. An LLM judge labels each fact as *align*, *contradict*, or *neutral*. We report the ratio  $\frac{\#align}{\#align + \#contradict}$ .
5. **Role-Playing Personality Consistency:** We prompt an LLM to infer the main NPC’s Big Five traits from the generated content, then compare these to the game definition. We employ the Ten-Item Personality Inventory (TIPI) Gosling et al. (2003), following previous work on eliciting LLM-based personality assessments of public figures Cao & Kosinski (2024). In addition to TIPI, we also considered a direct approach that explicitly evaluates alignment between the game’s narrative and the NPC’s predefined traits. We use TIPI-based score in the main paper, with details on the direct approach and comparisons in the Appendix D and E.
6. **Game Mechanics:** We perform a fully automatic check for the following errors:
  - (a) **Event Condition Error:** An event triggers when its entering condition is not met, or the outcome (success/failure) does not match the current state.
  - (b) **Variable Update Error:** The state variables do not update according to event effects.

The main game mechanic metric we adopt is the round-level accurate rate  $\mathbf{MEC} = \frac{\#Rounds\ with\ no\ errors}{\#Rounds}$ . We average the mechanic score over all games.

For a more fine-grained analysis, we process events in the *Event Plan* sequentially at each round and calculate an error rate for each error type,

$$\begin{aligned} \mathbf{ECE}_t &= \frac{\#Event\ condition\ errors}{\#Events} \\ \mathbf{VUE}_t &= \frac{\#State\ variables\ incorrectly\ updated}{\#State\ variables} \end{aligned} \quad (1)$$

We average ECE and VUE over all rounds of all games. By design, all these metrics require no LLM judge.

## 5 Results and Discussions

### 5.1 Experimental Details

**Game Creation** We consider GC to be a challenging task requiring complex reasoning over event-state interactions. Consequently, we evaluate advanced models with stronger reasoning capabilities: Claude 3.5 Sonnet, DeepSeek V3, Gemini 1.5 Pro, Gemini 2.0 Flash Exp, and GPT 4o.<sup>1</sup> We apply greedy decoding for all GC evaluations.

**Game Simulation** In addition to the models used in GC, we include GPT 4o mini, Llama 3.1 70B Instruct, and Llama 3.3 70B Instruct for the GS evaluation. Unless otherwise noted, we use a sampling temperature of 0.2 for inference. To maintain computational feasibility and fit within effective context windows of all models, we terminate all simulations after the 10th round for the main experiments. For all metrics requiring an LLM judge, we use GPT-4o as the evaluator.

### 5.2 Game Creation Results

**Main Results** Table 2 reports the format-check pass rate (FCR) and validity-check pass rate (VCR). We mark Claude 3.5 Sonnet with an asterisk (“\*”) because it frequently refuses to generate content, often citing an “over-lengthy output” error, causing 95% of its responses to fail the format check. We therefore focus on the fine-grained validity statistics for the remaining four models.

Most models (other than Claude 3.5 Sonnet) achieve high FCRs, indicating that they generally follow the specified formatting instructions. Among these models, GPT-4o attains the highest VCR of 0.49, while Gemini 1.5 Pro shows the lowest VCR of 0.04. Because passing the validity check demands a careful design of state variables and event systems, GPT-4o’s stronger planning and reasoning capabilities are highlighted in this task. A closer inspection on fine-grained metrics (w. Success, w. Lose and Reachability) reveals that Gemini 1.5 Pro frequently produces games that stall at intermediate steps without reaching success or failure endings. DeepSeek V3, in contrast, typically generates coherent event sequences, while GPT-4o often provides well-structured games with proper terminal outcomes.

**Game Difficulty Analysis** While VCR predominantly measures the logical consistency of generated games, game difficulty is another vital factor. Our game design allows us to estimate difficulty by analyzing (1) the ratio of success terminations to losing terminations and (2) the ratio of the lengths of the event chains leading to these endings. Formally, for a valid game  $v$ , let  $\mathcal{S}(v)$  be the set of all discovered success terminations and  $\mathcal{L}(v)$  the set of losing terminations. For each trajectory  $t$ , let  $length(\cdot)$  denote the number of events in  $t$ . We define:

$$\mathbf{CountRatio} = \frac{|\mathcal{S}(v)|}{|\mathcal{L}(v)|}, \quad \mathbf{LengthRatio} = \frac{\sum_{l \in \mathcal{L}_v} length(l)}{\sum_{s \in \mathcal{S}_v} length(s)} \cdot \frac{|\mathcal{S}(v)|}{|\mathcal{L}(v)|}$$

Intuitively, higher values for either ratio indicate an easier game. Figures 2 and 3 show box plots of these ratios for three selected models. Our analysis reveals that all models generate games with a

<sup>1</sup>Although models featuring inference-time reasoning can produce higher-quality results, the computational cost of running these models is often prohibitively high in practice.

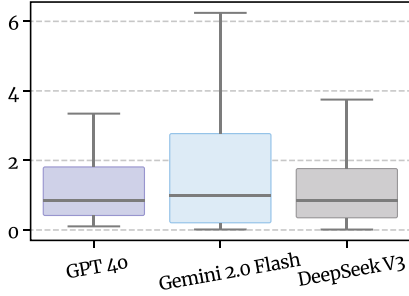


Figure 2: CountRatio of three models

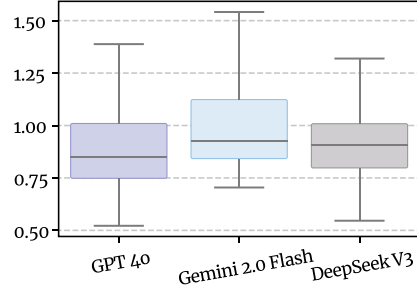


Figure 3: LengthRatio of three models

relatively balanced number of winning and losing trajectories. However, Gemini 2.0 Flash Exp tends to produce games where losing requires more steps, making failure less immediate. Additionally, the average **LengthRatio** is consistently below 1 across all models, indicating that winning generally requires more steps than losing—an expected outcome, as successful completion of a game typically demands more strategic progression.

Table 2: Game Creation results.

Models	FCR $\uparrow$	VCR $\uparrow$	w. Success	w. Lose	Reachability
Claude 3.5 Sonnet*	0.050	0.010	/	/	/
DeepSeek V3	0.990	0.380	0.455	0.545	<b>0.828</b>
Gemini 1.5 Pro	0.850	0.040	0.060	0.080	0.610
Gemini 2.0 Flash Exp	<b>1.000</b>	0.330	0.420	0.680	0.480
GPT 4o	0.960	<b>0.490</b>	<b>0.656</b>	<b>0.771</b>	0.656

### 5.3 Game Simulation Results

**Main Results** Table 3 presents our GS evaluation results, measuring length (LEN), role-playing factual consistency (FAC), personality consistency (PER), action quality (ACT), interestingness (INT), and mechanic score (MEC), along with tevent condition error rate (ECE) and variable update error rate (VUE) that decompose mechanic score.

Regarding LEN, DeepSeek V3, GPT 4o mini, and Llama 3.1 70B exceed the 200-word limit more than other models, which generally adhere to the instruction. All models exhibit high scores for factual consistency (FAC) and maintain moderate levels of personality consistency (PER). Action choice quality (ACT) is similarly high across models, but interestingness (INT) demonstrates wider variation. In particular, Claude 3.5 Sonnet achieves the highest INT score.

Game mechanic performance (MEC) varies the most among all metrics. Gemini 2.0 Flash Exp, GPT-4o, and Gemini 1.5 Pro perform comparably well, while the other models significantly worse. The best-performing model, Gemini 2.0 Flash Exp, only achieves a 0.765 MEC score, highlighting the inherent difficulty of precisely following complex game mechanics in a text-based RPG setting.

**Impact of Sampling Temperature** We further examine three metrics sensitive to sampling temperature—FAC, INT, and MEC—using GPT-4o at temperatures  $\{0.2, 0.5, 0.8\}$ . Table 4 summarizes the results. Interestingly, FAC increases with higher temperatures, which may initially seem counterintuitive given the heightened risk of hallucinations. However, we hypothesize that a larger temperature reduces the generation of factually neutral text, thereby leading to fewer overlooked facts. As expected, INT (interestingness) also rises with temperature, reflecting the increased creativity enabled by more diverse sampling. In contrast, MEC (mechanic score) peaks at the lowest temperature. This suggests that more deterministic sampling helps the model adhere more rigorously to the predefined game mechanics.

Table 3: Game Simulation results.

Model	LEN	FAC $\uparrow$	PER $\uparrow$	ACT $\uparrow$	INT $\uparrow$	MEC $\uparrow$	ECE $\downarrow$	VUE $\downarrow$
Claude 3.5 Sonnet	220.3	<b>0.991</b>	0.589	0.923	<b>0.722</b>	0.113	<b>0.062</b>	0.308
Deepseek V3	309.5	0.984	0.583	0.918	0.502	0.277	0.165	0.153
Gemini 1.5 Pro	198.0	0.968	0.596	0.894	0.602	0.554	0.081	0.085
Gemini 2.0 Flash Exp	195.3	0.885	<b>0.598</b>	0.865	0.538	<b>0.765</b>	0.094	<b>0.034</b>
GPT 4o	201.9	0.902	0.585	0.894	0.502	0.693	0.088	0.047
GPT 4o mini	282.5	0.955	0.588	0.900	0.496	0.147	0.126	0.148
Llama 3.1 70B Inst	279.2	0.977	0.586	0.915	0.420	0.162	0.161	0.284
Llama 3.3 70B Inst	225.7	0.960	0.585	<b>0.936</b>	0.466	0.204	0.201	0.302

Table 4: Performance under different sampling temperatures

Temperature	FAC	INT	MEC
0.2	0.920	0.502	<b>0.693</b>
0.5	0.939	0.520	0.629
0.8	<b>0.952</b>	<b>0.538</b>	0.643

Table 5: Performance under different number of simulation rounds

# Rounds	FAC	INT	MEC
10	0.920	<b>0.502</b>	<b>0.693</b>
15	<b>0.948</b>	0.480	0.679
20	0.941	0.458	0.674
25	0.941	0.440	0.668

**Impact of Number of Rounds** In our main experiments, we terminate each simulation at the 10th round, although games often do not naturally end that early. To assess the effect of longer trajectories, we take GPT-4o as an example and vary the number of rounds in  $\{10, 15, 20, 25\}$ . We focus on the metrics FAC, INT, and MEC, as the remaining metrics exhibit minimal variance. Table 5 shows that FAC increases with the number of rounds and eventually stabilizes, whereas INT decreases—likely due to repetitive content over extended sequences. The MEC score also declines, which may reflect the growing difficulty in maintaining coherent game mechanics within a longer context.

In our main experiments, we terminate simulations at the 10-th round. However, we found that games usually don’t terminate this early. To this end, we use GPT 4o as an example to study the performance with longer trajectories with the number of rounds being  $\{10, 15, 20, 25\}$ . We also study FAC, INT and MEC since other scores demonstrate small variations. We observe in Table 5 that FAC score increases with more rounds and eventually becomes stable. INT score decreases with more rounds, which could originate from repetitive content. MEC score also decreases, potentially due to the challenges in handling long context.

Despite the variations observed in Tables 4 and 5, the differences in INT and MEC remain relatively modest compared to the variability across models. Consequently, we conclude that limiting simulations to 10 rounds is adequate for most metrics, although extending the number of rounds may further improve the stability of the FAC score.

#### 5.4 Human Evaluation of Game Simulation

We also conduct a human evaluation on a subset of 20 simulated games, focusing on four subjective metrics: **FAC** (Factual Consistency), **PER** (Personality Consistency), **ACT** (Action Quality), and **INT** (Interestingness). We reframe these dimensions as natural-language questions to simplify the task for human annotators, who provide numerical scores later normalized to  $[0, 1]$ . Complete details on the human evaluation setup are provided in the Appendix E. Below, we outline two main differences between human evaluation and our automatic approach that can affect outcomes: **Scoring Procedure for Long Trajectories**. Since each game trajectory consists of 10 rounds, we present the content round by round and request a set of scores per round. We then average these round-level scores to derive final FAC, ACT, and INT metrics. Personality (PER) is an exception; because a single round may not reveal enough about the NPC’s character, annotators fill in a TIPI questionnaire at the end of the full trajectory. **Aggregated Factual Consistency**. Our automatic scorer checks each fact individually. However, to reduce the annotators’ workload, we ask them to give a single 1–5 rating for overall consistency with all facts.



Table 6: Human Score / Automatic Score / Absolute Difference of human and automatic scores

Models	FAC	ACT	INT	PER
Claude 3.5 Sonnet	<b>0.810</b> / <b>1.000</b> / 0.190	0.831 / 0.913 / 0.082	<b>0.856</b> / <b>0.713</b> / 0.144	0.648 / 0.729 / 0.081
Deepseek V3	<u>0.807</u> / <u>0.950</u> / 0.143	<u>0.857</u> / <u>0.913</u> / 0.056	0.850 / 0.475 / 0.375	0.645 / 0.742 / 0.098
Gemini 1.5 pro	0.733 / 0.950 / 0.217	0.738 / 0.889 / 0.152	0.801 / 0.588 / 0.214	0.648 / 0.740 / 0.093
Gemini 2.0 Flash Exp	0.769 / 0.800 / 0.031	0.851 / 0.876 / 0.025	<b>0.856</b> / <u>0.525</u> / 0.331	<u>0.651</u> / 0.737 / 0.085
GPT 4o	0.709 / <u>0.950</u> / 0.241	<b>0.881</b> / 0.887 / 0.007	0.834 / <u>0.525</u> / 0.309	<b>0.667</b> / 0.711 / 0.044
GPT 4o mini	0.770 / <u>0.950</u> / 0.180	0.794 / 0.887 / 0.093	0.813 / 0.488 / 0.326	0.648 / <b>0.753</b> / 0.104
Llama 3.1 70B Instruct	0.778 / <u>0.950</u> / 0.172	<u>0.857</u> / 0.898 / 0.041	0.824 / 0.400 / 0.424	0.627 / <u>0.744</u> / 0.117
Llama 3.3 70B Instruct	0.791 / 0.933 / 0.142	0.852 / <b>0.930</b> / 0.078	0.850 / 0.438 / 0.412	0.640 / 0.739 / 0.099

Table 7: Mean Absolute Difference (MAD) / Pearson correlation coefficient / Kendall rank correlation coefficient between automatic metrics and human evaluation scores (and among human evaluators).

Comparison	FAC	ACT	INT	PER
Auto v.s. Human	0.165 / 0.129 / 0.267	0.067 / 0.226 / 0.071	0.317 / 0.140 / 0.109	0.090 / -0.691 / -0.429
Human v.s. Human	0.030 / 0.707 / 0.571	0.039 / 0.472 / 0.214	0.018 / 0.508 / 0.286	0.023 / -0.310 / -0.286

Table 6 presents the human evaluation scores alongside our automatic metrics for each model, while Table 7 reports several comparative metrics such as mean absolute difference and correlation coefficients. Although human judgments can provide valuable insights, these metrics are inherently subjective and susceptible to personal biases. Consequently, human scores should be interpreted as reference points rather than definitive “gold standards.”

Examining Table 6, we find a fair degree of overlap in the top two performing models across FAC, ACT, and INT, but not for PER. From Table 7, we see that the inter-annotator correlation on PER is also very low, suggesting that personality judgments tend to be more variable and less stable.

Looking at the mean absolute differences (**MAD**) between human and automatic scores, ACT and PER exhibit relatively small discrepancies, whereas FAC and INT show larger gaps. Interestingly, FAC and INT also have somewhat higher correlation coefficients than the other metrics. Such results may stem from two factors: (1) the modifications we made for human evaluators versus automatic methods, and (2) the fact that the scores of different models are relatively close, making correlation metrics sensitive to small shifts.

Feedback from our annotators further indicates that **INT** can be heavily influenced by personal preferences. For instance, if a rater dislikes combat scenarios, they consistently assign lower interest scores to an action-heavy game trajectory. This shows that subjective evaluations—whether by humans or LLM judges—can vary widely based on individual tastes.

Although LLM-based scoring has been common in prior work for subjective dimensions, our human evaluation reveals that fine-grained comparisons remain unstable and less differentiable, even for human evaluators. This outcome highlights the importance of introducing objective metrics into game simulation assessment, such as our proposed game mechanic checks (Section 3.1) that do not rely on either human or LLM judgments.

## 6 Conclusion

In this work, we introduced and explored a comprehensive framework for evaluating large language models (LLMs) as creators and simulators of text-based role-play games. Our **Game Creation (GC)** task assesses the ability of LLMs to design valid games with a *BFS Validity Checker*. We further proposed a multi-round **Game Simulation (GS)** setup that prompts LLMs to plan events, generate narrative content with candidate player actions, and maintain game states.

In addition, we presented a hybrid evaluation scheme to capture both *objective* and *subjective* dimensions of game quality. On the objective side, our event-state mechanics checker operates without human or LLM judgment, automatically detecting errors in event conditions and variable updates. On the subjective side, we employed a series of metrics evaluated either through an *LLM-as-judge* approach or human annotation. Results across multiple models highlight that objective scores offer a stable foundation for comparison, while subjective dimensions have high variances.

## References

- BosonAI. Introducing rpbench-auto, August 2024. URL <https://boson.ai/rpbench-blog/>.
- Cao, X. and Kosinski, M. Large language models know how the personality of public figures is perceived by the general public. *Scientific Reports*, 14(1):6735, 2024.
- Chen, H., Chen, H., Yan, M., Xu, W., Xing, G., Shen, W., Quan, X., Li, C., Zhang, J., and Huang, F. SocialBench: Sociality evaluation of role-playing conversational agents. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2108–2126, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.125. URL <https://aclanthology.org/2024.findings-acl.125/>.
- Chen, J., Wang, X., Xu, R., Yuan, S., Zhang, Y., Shi, W., Xie, J., Li, S., Yang, R., Zhu, T., Chen, A., Li, N., Chen, L., Hu, C., Wu, S., Ren, S., Fu, Z., and Xiao, Y. From persona to personalization: A survey on role-playing language agents. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856. URL <https://openreview.net/forum?id=xr070E8UIZ>. Survey Certification.
- Dai, Y., Hu, H., Wang, L., Jin, S., Chen, X., and Lu, Z. Mmrole: A comprehensive framework for developing and evaluating multimodal role-playing agents. *CoRR*, abs/2408.04203, 2024. doi: 10.48550/ARXIV.2408.04203. URL <https://doi.org/10.48550/arXiv.2408.04203>.
- Gosling, S. D., Rentfrow, P. J., and Swann Jr, W. B. A very brief measure of the big-five personality domains. *Journal of Research in personality*, 37(6):504–528, 2003.
- Gusev, I. Pingpong: A benchmark for role-playing language models with user emulation and multi-model evaluation. *CoRR*, abs/2409.06820, 2024. doi: 10.48550/ARXIV.2409.06820. URL <https://doi.org/10.48550/arXiv.2409.06820>.
- Samuel, V., Zou, H. P., Zhou, Y., Chaudhari, S., Kalyan, A., Rajpurohit, T., Deshpande, A., Narasimhan, K., and Murahari, V. Personagym: Evaluating persona agents and llms. *CoRR*, abs/2407.18416, 2024. doi: 10.48550/ARXIV.2407.18416. URL <https://doi.org/10.48550/arXiv.2407.18416>.
- Tu, Q., Fan, S., Tian, Z., Shen, T., Shang, S., Gao, X., and Yan, R. CharacterEval: A Chinese benchmark for role-playing conversational agent evaluation. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11836–11850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.638. URL <https://aclanthology.org/2024.acl-long.638/>.
- Wang, L., Lian, J., Huang, Y., Dai, Y., Li, H., Chen, X., Xie, X., and Wen, J.-R. Characterbox: Evaluating the role-playing capabilities of llms in text-based virtual worlds. *arXiv preprint arXiv:2412.05631*, 2024a.
- Wang, X., Xiao, Y., Huang, J.-t., Yuan, S., Xu, R., Guo, H., Tu, Q., Fei, Y., Leng, Z., Wang, W., Chen, J., Li, C., and Xiao, Y. InCharacter: Evaluating personality fidelity in role-playing agents through psychological interviews. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1840–1873, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.102. URL <https://aclanthology.org/2024.acl-long.102/>.
- Yuan, X., Yuan, S., Cui, Y., Lin, T., Wang, X., Xu, R., Chen, J., and Yang, D. Evaluating character understanding of large language models via character profiling from fictional works. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8015–8036, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.456. URL <https://aclanthology.org/2024.emnlp-main.456/>.

## A BFS Validity Checker Algorithm

We present the detailed algorithm for BFS validity checker in Section 4.1.

---

### Algorithm 1 BFS Validity Checker

---

**Input:** Events  $\mathcal{E}$ , each with entering and success conditions, plus success and fail effects; A state  $S_0$  with initial values for all variables; An integer  $M$  indicating the maximum number of states to be explored.

**function** *isValid*( $\mathcal{E}, S_0, M$ )

  Initialize a queue  $\mathcal{Q}$  and enqueue  $S_0$ .

  Initialize a visited set  $\mathcal{V} = \{S_0\}$ .

  Initialize a triggered-event set  $\mathcal{T} = \emptyset$ .

*successFound* = **False**; *loseFound* = **false**

**repeat**

$S = \mathcal{Q}.\text{dequeue}()$

**if**  $|\mathcal{V}| > M$  **then**

**break**   *# Reached maximum search limit*

**end if**

$\text{availableEvents} = \{e \in \mathcal{E} : e.\text{enterCond}(S)\}$

**for each**  $e \in \text{availableEvents}$  **do**

$\mathcal{T} = \mathcal{T} \cup \{e\}$    *# Mark event as triggered*

$S' = e.\text{applyEffect}(S, e.\text{successCond}(S))$

$\text{successFound} \mid = e.\text{isSuccessTermination}(S')$

$\text{loseFound} \mid = e.\text{isLosingTermination}(S')$

**if**  $S' \notin \mathcal{V}$  **then**

$\mathcal{Q}.\text{enqueue}(S'); \mathcal{V} = \mathcal{V} \cup \{S'\}$

**end if**

**end for**

**until**  $\mathcal{Q}$  is empty

**return**  $(\mathcal{T} = \mathcal{E}) \mid \text{successFound} \mid \text{loseFound}$

**end function**

---

## B Game JSON Structure in RPGBENCH

As introduced in Section 3.1, each game in RPGBENCH is represented by a JSON dictionary. Figures 4 and 5–10 provide the complete schema and its referenced object definitions. Below, we clarify naming discrepancies between this JSON specification and the terminology used in the main article, and also highlight a few design details omitted for brevity.

**Naming Discrepancies.** The JSON schema in Figure 4 has property names slightly different from those in Figure 1 from the main article. For clarity, we list them side by side as “JSON schema name — main article name”:

1. `player_name` — Player Character / Name
2. `player_description` — Player Character / Description
3. `main_npc_description` / `text` — Main NPC / Description
4. `main_npc_description` / `big5_personality_traits` — Main NPC / Personality
5. `main_npc_description` / `additional_facts` — Main NPC / Facts
6. `state_variables` + `hidden_variables` — State Variables
7. `pre_event_checks` — Termination Conditions

For consistency, the appendices continue to use the names from the main article unless otherwise specified. Although `state_variables` and `hidden_variables` are separate fields in the JSON schema, they collectively represent the State Variables described in the main text. In our design, `hidden_variables` (unlike `state_variables`) are not displayed to players; however, this distinction does not impact the benchmark evaluations and is thus not emphasized in the main article.

We also require `hidden_variables` to include at least two special Boolean flags, `has_succeeded` and `has_failed`, which interact with `pre_event_checks` (a list of two check objects If Succeeded and If Failed). Each check object includes a condition (a Boolean expression over the state variables) and an effect that sets `has_succeeded=1` or `has_failed=1`, if not already set<sup>2</sup>. Conceptually, these properties mirror the Termination Conditions in the main article.

**Explanatory Content.** Several text fields in the JSON schema contain descriptive or explanatory information that we omit from the main article, such as:

1. `$def/trait/description`: Describes the personality trait score in natural language.
2. `$def/scene_object/background_description`: Describes the scene.
3. `$def/variable_object/description`: Describes a particular state variable.
4. `$def/event_object/explanations`: Explains event effects.
5. `$def/pre_event_check_object/explanation`: Explains the termination condition check.

Although these fields do not affect our validity checks, they provide additional context for LLMs and are included in prompts given to LLMs during game simulation.

**Game Scenes in the BFS Validity Check.** Because each event references exactly one scene (Figure 9), we also verify that all declared scenes are referenced by at least one event. This check is straightforward and independent of the BFS procedure, so it is omitted from the main article for simplicity.

## C Game Creation Prompt

For the Game Creation (GC) task, we use the prompt shown below. It references the Wikipedia content of the chosen main NPC (`{wikicontent}`) and the JSON schema defined in Appendix B (`{schema}`). The full text of this schema is provided to the model so it can generate a well-structured JSON output.

```
Here is a character description:
{wikicontent}

Based on this character, create a detailed game scenario exactly
following JSON structure of previous examples and the following
schema:
{schema}

## Guidelines
- All numerical values should use consistent ranges (e.g., 0-100)
- Events should have clear cause-and-effect relationships
- Scene progression should depend on variable thresholds
- Include both mandatory and optional events
- Create meaningful connections between variables
- Balance difficulty and achievability
- Ensure all IDs follow consistent formatting (P### for checks, S
  ### for scenes, V### for state variables, H### for hidden
  variables, E### for events)
- Include proper fail states and success conditions
- Make sure all scenes are specific locations
- Create logical progression paths through the game

Format the response as a single JSON object with all fields
properly nested. Must ensure all arrays and objects are
properly closed and formatted.
```

---

<sup>2</sup>Some games directly set `has_succeeded` or `has_failed` in other event effects, leaving effects of `pre_event_checks` empty.

```

1 {
2   "title": "Game Configuration",
3   "type": "object",
4   "required": [
5     "game_world",
6     "player_name",
7     "player_description",
8     "main_npc_name",
9     "main_npc_description",
10    "game_objectives",
11    "scenes",
12    "state_variables",
13    "hidden_variables",
14    "events",
15    "pre_event_checks"
16  ],
17  "properties": {
18    "game_world": { "type": "string" },
19    "player_name": { "type": "string" },
20    "player_description": { "type": "string" },
21    "main_npc_name": { "type": "string" },
22    "main_npc_description": {
23      "type": "object",
24      "required": [ "text", "big5_personality_traits", "
25        additional_facts" ],
26      "properties": {
27        "text": { "type": "string" },
28        "big5_personality_traits": { "$ref": "#/$defs/big5_traits"
29        },
30        "additional_facts": { "type": "array", "items": { "type":
31        "string" } }
32      },
33      "additionalProperties": false
34    },
35    "game_objectives": { "type": "string" },
36    "scenes": { "type": "array", "items": { "$ref": "#/$defs/
37    scene_object" } },
38    "state_variables": { "type": "array", "items": { "$ref": "#/
39    $defs/variable_object" } },
40    "hidden_variables": {
41      "type": "array",
42      "minItems": 2,
43      "items": { "$ref": "#/$defs/variable_object" },
44      "contains": { "properties": { "value_name": { "enum": [ "
45        has_succeeded", "has_failed" ] } } }
46    },
47    "events": { "type": "array", "items": { "$ref": "#/$defs/
48    event_object" } },
49    "pre_event_checks": { "type": "array", "items": { "$ref": "#/
50    $defs/pre_event_check_object" } },
51    "source": { "type": "string" }
52  },
53  "additionalProperties": false,
54 }

```

Figure 4: JSON Schema for Game Configuration

```

1 {
2   "$defs": {
3     "trait": {
4       "type": "object",
5       "required": ["rate", "description"],
6       "properties": {
7         "rate": { "type": "number" },
8         "description": { "type": "string" }
9       },
10      "additionalProperties": false
11    }
12  }
13 }

```

Figure 5: trait object schema

```

1 {
2   "$defs": {
3     "big5_traits": {
4       "type": "object",
5       "required": [
6         "openness",
7         "conscientiousness",
8         "extraversion",
9         "agreeableness",
10        "neuroticism"
11      ],
12      "properties": {
13        "openness": { "$ref": "#/$defs/trait" },
14        "conscientiousness": { "$ref": "#/$defs/trait" },
15        "extraversion": { "$ref": "#/$defs/trait" },
16        "agreeableness": { "$ref": "#/$defs/trait" },
17        "neuroticism": { "$ref": "#/$defs/trait" }
18      },
19      "additionalProperties": false
20    }
21  }
22 }

```

Figure 6: big5\_traits object schema

```

1 {
2   "$defs": {
3     "scene_object": {
4       "type": "object",
5       "required": [ "scene_name", "unique_id", "
6         background_description", "scene_type" ],
7       "properties": {
8         "scene_name": { "type": "string" },
9         "unique_id": { "type": "string" },
10        "background_description": { "type": "string" },
11        "scene_type": { "type": "string" }
12      },
13      "additionalProperties": false
14    }
15  }
16 }

```

Figure 7: scene\_object schema

**5-Shot Prompt** To guide LLMs more effectively, we supply five example JSON games prior to the main creation prompt. Because each game JSON can be quite lengthy, stacking them directly after

```

1 {
2   "$defs": {
3     "variable_object": {
4       "type": "object",
5       "required": [ "value_name", "unique_id", "description", "
6         min_value", "max_value" ],
7       "properties": {
8         "value_name": { "type": "string" },
9         "unique_id": { "type": "string" },
10        "description": { "type": "string" },
11        "initial_value": { "type": "string" },
12        "min_value": { "type": "string" },
13        "max_value": { "type": "string" }
14      }, "additionalProperties": false
15    }
16  }

```

Figure 8: variable\_object schema

```

1 {
2   "$defs": {
3     "event_object": {
4       "type": "object",
5       "required": [ "event_name", "unique_id", "scene", "
6         entering_condition", "succeed_condition", "succeed_effect
7         ", "fail_effect" ],
8       "properties": {
9         "event_name": { "type": "string" },
10        "unique_id": { "type": "string" },
11        "scene": { "type": "array", "items": { "type": "string" }
12        },
13        "entering_condition": { "type": "array", "items": { "type":
14          "string" } },
15        "succeed_condition": { "type": "array", "items": { "type":
16          "string" } },
17        "succeed_effect": { "type": "array", "items": { "type": "
18          string" } },
19        "fail_effect": { "type": "array", "items": { "type": "
20          string" } },
21        "explanations": { "type": "string" }
22      }, "additionalProperties": false
23    }
24  }

```

Figure 9: event\_object schema

the prompt may cause the model to overlook important details in the instruction. Instead, we present the five-shot examples as sequential conversation entries, followed by the actual creation prompt. The resulting conversation structure is illustrated below.

```

1 {
2   "$defs": {
3     "pre_event_check_object": {
4       "type": "object",
5       "required": [ "check_name", "unique_id", "description", "
6         condition", "effect" ],
7       "properties": {
8         "check_name": { "type": "string" },
9         "unique_id": { "type": "string" },
10        "description": { "type": "string" },
11        "condition": { "type": "array", "items": { "type": "string
12          " } },
13        "effect": { "type": "array", "items": { "type": "string" }
14        },
15        "explanation": { "type": "string" }
16      }, "additionalProperties": false
17    }
18  }
19 }

```

Figure 10: pre\_event\_check\_object schema

```

USER: Give me an example game JSON.
ASSISTANT: {EXAMPLE_1}
USER: Give me an example game JSON.
ASSISTANT: {EXAMPLE_2\}
USER: Give me an example game JSON.
ASSISTANT: {EXAMPLE_3\}
USER: Give me an example game JSON.
ASSISTANT: {EXAMPLE_4\}
USER: Give me an example game JSON.
ASSISTANT: {EXAMPLE_5}
USER: {Prompt for Game Creation}

```

## D Evaluation Prompts and Detailed Score Calculations

We employ a consistent three-part format for most evaluation prompts: an instruction section, a JSON schema specifying the output format, and an example response. To keep this appendix concise, we omit the JSON schemas and example responses when the instruction text alone clearly explains the expected output structure. Below, we detail the prompts and score calculations for four metrics: Main NPC Factual Consistency (**FAC**), Main NPC Personality Consistency (**PER**), Interestingness (**INT**), and Action Choice Quality (**ACT**).

### D.1 Main NPC Factual Consistency (FAC)

The prompt below assesses how closely the generated game content aligns with each fact about the main NPC. We concatenate all LLM-generated game narration across the multi-round trajectory into `game_content`<sup>3</sup>.

<sup>3</sup>Event Plan and State Variables are omitted because they are not visible to players.



You are given a piece of narrative game content and a set of facts about a specific non-player character (NPC). Your task is to analyze whether each fact is supported, contradicted, or not addressed by the provided game content. For each fact, determine one of the following judgements based solely on the given game content:

- "align": The game content supports or is consistent with the fact.
- "contradict": The game content directly conflicts with or contradicts the fact.
- "neutral": The game content is unrelated or does not provide enough information to judge the fact.

Please disregard prior knowledge and analyze the NPC purely based on the game content and the facts.

```

**NPC**: {main_npc_name}

**Game Content**:

{game_content}

**Facts**

{main_npc_facts}

**Output Format**:
Return the results as a JSON array, where each element is an
object with:
- fact_id: the corresponding fact's ID.
- judgement: one of "align", "contradict", or "neutral".
- explanation: a brief explanation for your judgment, referencing
  specific parts of the game content if applicable.
The return json array should follow this json schema:
{schema}

**Example Response**:
{example}

```

The judge assigns one of three labels for each fact: “align,” “contradict,” or “neutral.” The final trajectory-level FAC score is computed as

$$\mathbf{FAC}_{\text{traj}} = \frac{\# \text{align}}{\# \text{align} + \# \text{contradict}}, \quad (2)$$

and we then average over all trajectories:

$$\mathbf{FAC} = \frac{\sum_{\text{traj}} \mathbf{FAC}_{\text{traj}}}{\# \text{trajectories}}. \quad (3)$$

## D.2 Main NPC Personality Consistency (PER)

**TIPI PER Score** As described in the main article, we derive the PER score using a Ten-Item Personality Inventory (TIPI) approach Gosling et al. (2003); Cao & Kosinski (2024), prompting the LLM judge to rate each of ten statements and then converting the ratings into Big Five trait scores.

You will be given information about a character. Here are a number of personality traits that may or may not apply to the character. Please write a number to each statement to indicate the extent to which you agree or disagree with that statement. You should rate the extent to which the pair of traits applies to the character, even if one characteristic applies more strongly than the other.

For the ratings:

- 1: Disagree strongly
- 2: Disagree moderately
- 3: Disagree a little
- 4: Neither agree nor disagree
- 5: Agree a little
- 6: Agree moderately
- 7: Agree strongly

Please give your ratings for the following 10 statements.

I see the character as:

- A. Extraverted, enthusiastic.
- B. Critical, quarrelsome.
- C. Dependable, self-disciplined.
- D. Anxious, easily upset.
- E. Open to new experiences, complex.
- F. Reserved, quiet.
- G. Sympathetic, warm.
- H. Disorganized, careless.
- I. Calm, emotionally stable.
- J. Conventional, uncreative

Please return ratings for all 10 traits in a dictionary following this schema:  
{schema}

Please give your ratings for the following character.  
{character}

Here, character consists of the main NPC name and the concatenated LLM-generated game narration sections. According to Gosling et al. (2003), we use the following formulas to calculate personality trait scores,

$$\begin{aligned}
\text{Openness:} \quad o_{tipi} &= E + 8 - J \\
\text{Conscientiousness:} \quad c_{tipi} &= C + 8 - H \\
\text{Extroversion:} \quad e_{tipi} &= A + 8 - F \\
\text{Agreeableness:} \quad a_{tipi} &= G + 8 - B \\
\text{Neuroticism:} \quad n_{tipi} &= I + 8 - D
\end{aligned} \tag{4}$$

To compute the personality consistency, we compare the above scores, after being scaled to  $[1, 5]$ , with the main NPC personality specifications in the game JSON,

$$d_{\{o,c,e,a,n\}} = \left| \frac{\{o,c,e,a,n\}_{tipi} + 1}{3} - \{o,c,e,a,n\}_{game} \right|. \tag{5}$$

The PER score is the squared sum of these differences, normalized to  $[0, 1]$ ,

$$\begin{aligned}
\mathbf{PER}_{\text{traj}} &= 1 - \frac{\sqrt{\sum_{x \in \{o,c,e,a,n\}} d_x^2}}{4\sqrt{5}} \\
\mathbf{PER} &= \frac{\sum_{\text{traj}} \mathbf{PER}_{\text{traj}}}{\# \text{trajectories}}
\end{aligned} \tag{6}$$

**Direct Evaluation of Personality Consistency** We also experiment with a direct evaluation approach (referred to as **PER<sup>d</sup>**), which instructs the LLM judge to provide a 1–5 alignment rating for each of the five personality traits.

Assign a score from 1 to 5 to indicate how well the game narrative aligns with the main NPC’s personality traits:

- Many Conflicts (1): The narrative frequently contradicts the NPC’s personality.
- Some Conflicts (2): The narrative shows noticeable inconsistencies with the NPC’s personality.
- Neutral (3): The narrative is only partially aligned or does not strongly reflect the NPC’s personality.
- Strong Alignment (4): The narrative closely matches the NPC’s personality, with only minor deviations or uncertainties.
- Perfect Alignment (5): The narrative flawlessly reflects the NPC’s personality in every aspect, with no contradictions.

Please give one score for each personality trait, and provide a brief explanation for each score.

Game narrative:  
{game\_content}

NPC personality:  
{npc\_personality}

Please return a score as a json object following this schema:  
{schema}

Here, `npc_personality` consists of the Big Five personality traits in the game JSON. We compute the final score by averaging the normalized scores across all traits and, subsequently, across all trajectories. We defer discussions of results from this approach to Appendix E, where we compare both TIPI estimations and direct evaluation results from LLM judges and human annotators. We refer this score as **PER<sup>d</sup>** for the remaining of this article.

### D.3 Interestingness (INT)

We prompt an LLM judge to rate the interestingness of the generated content on a 1–5 scale.

Your task is to evaluate the **interestingness** of the following game content. Please give a score from 1 (least interesting) to 5 (most interesting), with a brief explanation of your rationale.

[[start of game content]]  
{game\_content}  
[[end of game content]]

Please return your evaluation score in a json dictionary with the following format:  
{schema}

Example output:  
{example}

We normalize the final score to  $[0, 1]$ , sum over rounds within a trajectory and then average:

$$\begin{aligned} \text{INT}_{\text{traj}} &= \frac{\text{int} - 1}{4} \\ \text{INT} &= \frac{\sum_{\text{traj}} \text{INT}_{\text{traj}}}{\# \text{trajectories}} \end{aligned} \quad (7)$$

## D.4 Action Choice Quality (ACT)

At each round, the LLM judge scores three candidate actions on three rubrics: Diversity, Relevance, and Understandability, each on a 1–5 scale with the following prompt.

```
Please act as an experienced RPG game player and evaluate the
choices provided by the game engine, given the user-AI
interaction history and the general game instruction outlining
the basic game settings. You will be given the general game
instruction, the interaction history, and the current choices
offered by the game engine. Evaluate the choices based on the
following rubric:
```

```
Please assess the choices provided by the game engine based on
this rubric:
```

```
[[start of rubric]]
{rubric}
[[end of rubric]]
```

```
[[start of general game instruction]]
{game}
[[end of general game instruction]]
```

```
[[start of history]]
{history}
[[end of history]]
```

```
Please assess the choices provided by the game engine:
```

```
[[start of choices]]
{choices}
[[end of choices]]
```

```
Your output should be a JSON object structured as follows:
{
```

```
  "reason": <your reasoning here>
  "score": <score from 1 to 5 based on the rubric provided>
}
```

```
You must NOT output anything else other than this JSON object.
```

We provide in the prompt above the game JSON (game), game history up to the current round (history), and three candidate choices (choices). We present the following paragraphs in rubric for Diversity, Relevance and Understandability respectively.

```
Diversity: Does the set of choices provide distinct and varied
options for the player?
1: The choices are nearly identical, offering no meaningful
differences between them.
2: The choices have slight variations but are mostly redundant,
leading to a limited sense of variety.
3: The choices exhibit some diversity but may still overlap in
intent or outcome.
4: The choices are mostly distinct and provide meaningful
differences that allow the player to explore different paths.
5: The choices are highly diverse, with each option offering
unique and creative directions for the player.
```

Relevance: Are the choices appropriate and contextually aligned with the story and scene?

- 1: The choices are entirely irrelevant, disconnected from the scene or story, and break immersion.
- 2: The choices have limited relevance, with some alignment to the story but containing jarring or out-of-place elements.
- 3: The choices are moderately relevant, generally aligning with the story but occasionally introducing inconsistencies.
- 4: The choices are mostly relevant, fitting well within the context and contributing meaningfully to the story.
- 5: The choices are fully relevant, seamlessly integrated into the story and enhancing the narrative experience.

Understandability: Are the choices clear, concise, and easy to understand for the player?

- 1: The choices are confusing, overly complex, or poorly worded, making them difficult to interpret.
- 2: The choices are somewhat understandable but may include ambiguous language or unnecessary complexity.
- 3: The choices are moderately clear, with minor ambiguities that require some interpretation.
- 4: The choices are clear and concise, easy to read, and free of significant ambiguity.
- 5: The choices are exceptionally clear and well-written, making them effortless to understand and act upon

We average these three rubric scores to obtain  $act$ , then normalize via  $(act - 1)/4$ . Trajectories are evaluated by averaging per-round scores, and we then take the mean across all trajectories:

$$\begin{aligned}
\mathbf{ACT}_{\text{round}} &= \frac{act - 1}{4} \\
\mathbf{ACT}_{\text{traj}} &= \frac{\sum_{\text{round}} \mathbf{ACT}_{\text{round}}}{\#rounds} \\
\mathbf{ACT} &= \frac{\sum_{\text{traj}} \mathbf{ACT}_{\text{traj}}}{\#trajectories}
\end{aligned} \tag{8}$$

## E Human Evaluation Details

### E.1 Interface Layout

Figure 11 shows a screenshot of our human evaluation interface. Although it is cut off due to display size, the four main components are visible: **Text RPG Information**, **NPC Information**, **Dialog History**, and **Responses**. As discussed in Appendix D.2, we use two interfaces: one for TIPI-based personality estimation and one for direct personality-consistency evaluation. These interfaces only differ in how the **NPC Information** and **Responses** sections are presented. To help annotators remain focused when assessing a multi-round trajectory, each round in a trajectory is annotated separately by the same annotator.

**Text RPG Information:** Annotators see the Game World description, the player character’s name and description, and the overall game objective. This information persists throughout the trajectory.

**Dialog History:** We show the game trajectory up to the current round, including the model’s narration and three candidate actions (boldfaced). One of these actions, selected at random, is displayed on the right side. This component updates every round to reflect the new content.

**NPC Information:** For TIPI-based personality estimation, we present only the NPC’s name and facts (omitting personality traits so they can be inferred through the TIPI questions). In the direct-evaluation interface, the main NPC personality traits are included here.

**Responses:** This section poses natural-language questions to gather human judgments on subjective dimensions. It differs slightly between TIPI-based and direct-evaluation interfaces, as detailed below.

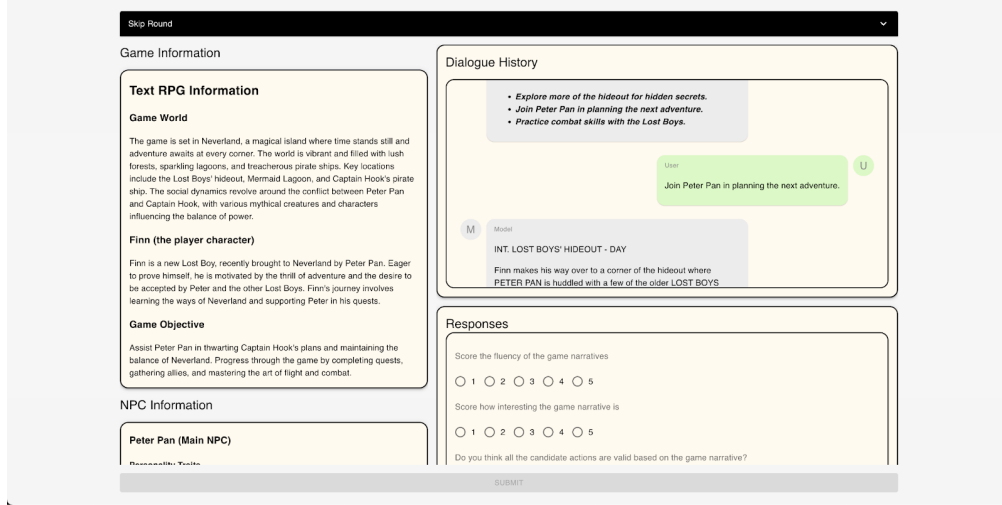


Figure 11: Screenshot of the human evaluation interface.

## E.2 Evaluation Questions

Our human evaluation asks annotators to rate various subjective aspects. Questions A–D appear every round in both the TIPI and direct-evaluation interfaces:

- A. Please give a score (1-5) to indicate how interesting the game narrative is.
- B. Do you think all the candidate actions are valid based on the game narrative? - 0 (no) - 1 (yes)
- C. Are candidate choices different enough from each other, or are they essentially the same? - 0 (same) - 1 (different)
- D. Please give a score (1-5) to measure whether the game narrative is consistent with the given facts about the main NPC?
  - 1 has many conflicts
  - 2 has some conflicts
  - 3 neutral
  - 4 matches the description
  - 5 perfectly matches the description

These ratings inform the INT, ACT, and FAC metrics as follows:

$$\begin{aligned}
 \text{INT}_{\text{round}} &= \frac{A - 1}{4}, \\
 \text{ACT}_{\text{round}} &= \frac{B + C}{2}, \\
 \text{FAC}_{\text{round}} &= \frac{D - 1}{4}.
 \end{aligned} \tag{9}$$

Here, Question B corresponds to Relevance and Understandability in the ACT automatic evaluation, while Question C corresponds to Diversity. We average these round-level scores to obtain a trajectory-level score, then average across all trajectories.

**Personality Consistency Questions (E1 and E2).** We measure PER using two different question sets:

- **E1: TIPI Estimation.** Shown only once per trajectory (at the final round of the TIPI interface), requiring annotators to assess the entire trajectory.

- **E2: Direct Evaluation.** Appears at every round in the direct-evaluation interface.

Both methods yield PER scores analogous to the automatic evaluations in Appendix D.2.

E1. Here are a number of personality traits that may or may not apply to the character. Please write a number to each statement to indicate the extent to which you agree or disagree with that statement, based ONLY on the game narratives. You should rate the extent to which the pair of traits applies to the character, even if one characteristic applies more strongly than the other. Use a score range of 1-7:

- 1: Disagree strongly
- 2: Disagree moderately
- 3: Disagree a little
- 4: Neither agree nor disagree
- 5: Agree a little
- 6: Agree moderately
- 7: Agree strongly

I see the main NPC as

- A. Extraverted, enthusiastic.
- B. Critical, quarrelsome.
- C. Dependable, self-disciplined.
- D. Anxious, easily upset.
- E. Open to new experiences, complex.
- F. Reserved, quiet.
- G. Sympathetic, warm.
- H. Disorganized, careless.
- I. Calm, emotionally stable.
- J. Conventional, uncreative

E2. Please give a score (1-5) to measure whether the game narrative is consistent with the given facts about the main NPC ?

- 1 has many conflicts
- 2 has some conflicts
- 3 neutral
- 4 matches the description
- 5 perfectly matches the description

### E.3 Annotation Setup

We recruited 15 human annotators. Each trajectory is annotated at the round level, resulting in two annotations per interface type and therefore four total annotations per trajectory. We ensure that each annotator encounters any given trajectory only once, regardless of interface type. Consequently, each trajectory ends up with four sets of INT, ACT, and FAC scores, and two sets of PER and  $PER^d$  scores. We take the mean over all trials to produce the final reported values. For inter-annotator agreement (Table 7), we randomly divide the collected annotations into two groups and compare their scores.

### E.4 PER vs. $PER^d$ Evaluation Results

In our main article, we adopt the PER score for evaluating NPC personality consistency. Here, we further analyze both PER and  $PER^d$  scores from automatic and human evaluations on a subset of 20 games in Table 8, with additionally automatic evaluation results on the full dataset. We also report agreement metrics in Table 9 Our analysis reveals several key observations:

1.  **$PER^d$  tends to be higher than PER in both automatic and human evaluations.** Across models, we observe that  $PER^d$  scores are consistently higher than PER scores, indicating that direct evaluation of personality consistency is generally more lenient than the TIPI-based method. This trend holds for both automatic and human evaluators.
2. **LLMs achieve similar PER scores across the dataset.** The automatic PER and  $PER^d$  scores on the full set of games show little variation across models, with all models achieving

Model	PER (Subset)		PER <sup>d</sup> (Subset)		PER (Full)	PER <sup>d</sup> (Full)
	Auto	Human	Auto	Human	Auto	Auto
Claude 3.5 Sonnet	0.729	0.648	0.768	<b>0.832</b>	0.589	0.738
Deepseek V3	0.742	0.645	0.750	<u>0.826</u>	0.583	<b>0.778</b>
Gemini 1.5 Pro	0.740	0.648	<b>0.800</b>	0.769	<u>0.596</u>	<u>0.777</u>
Gemini 2.0 Flash Exp	0.737	<u>0.651</u>	0.707	0.769	<b>0.598</b>	0.750
GPT 4o	0.711	<b>0.667</b>	0.780	0.724	0.585	0.768
GPT 4o mini	<b>0.753</b>	0.648	<u>0.788</u>	0.735	0.588	0.763
Llama 3.1 70B	<u>0.744</u>	0.627	0.768	0.752	0.586	0.765
Llama 3.3 70B	0.739	0.640	0.739	0.755	0.585	0.774

Table 8: PER and PER<sup>d</sup> results from automatic and human evaluation on a subset of 20 games, and automatic evaluation on the full set of games.

Comparisons		Pearson	Kendall	MAD
Auto-Auto Agreement	PER Auto - PER <sup>d</sup> Auto	0.013	0.109	0.037
Auto-Human Agreement	PER Auto - PER Human	-0.691	-0.429	0.090
	PER <sup>d</sup> Auto - PER <sup>d</sup> Human	-0.297	-0.255	0.047
Human-Human Agreement	PER Human - PER Human	-0.310	-0.286	0.023
	PER <sup>d</sup> Human - PER <sup>d</sup> Human	0.649	0.143	0.035
	PER Human - PER <sup>d</sup> Human	-0.175	-0.143	0.124

Table 9: Agreement analysis for PER and PER<sup>d</sup> scores. We present Pearson correlation coefficient (Pearson), Kendall rank correlation coefficient (Kendall), and Mean Absolute Difference (MAD)

scores around 0.58–0.60 for PER and around 0.74–0.78 for PER<sup>d</sup>. This suggests that models perform comparably in terms of maintaining personality consistency in text-based role-playing.

- Human evaluators rate PER<sup>d</sup> higher than PER, but with noticeable variation.** While automatic evaluations show a clear gap between PER and PER<sup>d</sup>, human annotations exhibit a similar pattern but with greater variability. Notably, human evaluators assign significantly higher PER<sup>d</sup> scores to some models, such as Claude 3.5 Sonnet and DeepSeek V3, compared to their automatic scores.
- Human and automatic PER scores exhibit poor correlation.** Table 9 shows that the Pearson correlation between PER Auto and PER Human is negative (-0.691), with Kendall correlation also negative (-0.429). This suggests a fundamental mismatch between how LLM-based and human evaluators assess personality consistency through TIPI.
- Better human agreement for PER<sup>d</sup>, but still unstable.** While inter-human correlation for PER is negative (-0.310 Pearson, -0.286 Kendall), PER<sup>d</sup> exhibits a stronger but still weak agreement (0.649 Pearson). This suggests that directly rating personality alignment may be more intuitive for human evaluators than using TIPI scores but remains somewhat unstable. However, there is still concern over whether human annotators are capable of accurately understanding Big Five traits in the direct evaluation scenario.
- Low agreement between PER and PER<sup>d</sup>.** The Pearson correlation between PER and PER<sup>d</sup> scores (both automatic and human) is low (0.013 for Auto-Auto and -0.175 for Human-Human), indicating that these two evaluation methods capture different aspects of personality consistency. While PER<sup>d</sup> measures direct alignment with given traits, PER (TIPI) estimates personality traits implicitly, which may introduce more variance in judgments.

**Justification for Choosing TIPI (PER) in the Main Article.** We adopt TIPI-based personality consistency (PER) rather than direct evaluation (PER<sup>d</sup>) in the main study for several reasons. First, TIPI does not require evaluators to have prior knowledge of the Big Five personality traits, making it a structured and interpretable method for assessing personality consistency. Additionally, the high variance in PER<sup>d</sup> human scores (as seen in Table 9) suggests that direct personality evaluation is more



susceptible to subjective biases. The negative correlation between automatic and human PER scores further emphasizes the challenge of aligning LLM-based and human-based assessments, reinforcing the need for a more systematic approach like TIPI.

Overall, these results highlight the complexity of evaluating personality consistency, where different evaluation paradigms yield divergent results. The instability in human-human agreement for both PER and PER<sup>d</sup> suggests that subjective evaluation remains a challenging aspect of LLM benchmarking, warranting further research into more reliable personality evaluation methodologies.

## F Case Studies

### F.1 Case Study 1: Game Creation, Mickey Mouse

Below is a game generated by GPT-4o given the Wikipedia page content of Mickey Mouse. This game passes BFS validity check, as presented at the end of the game.

**Game World:** The game is set in the whimsical and vibrant world of the Mickey Mouse universe, where players explore iconic locations such as the Steamboat Willie River, the bustling streets of Toontown, and the magical Fantasia Forest. The world is filled with adventure, humor, and the timeless charm of classic Disney animation.

**Player Character:** Charlie is a young and adventurous mouse who idolizes Mickey Mouse. Eager to prove himself, Charlie embarks on a journey to assist Mickey in overcoming challenges and learning valuable life lessons. Motivated by a desire to be brave and resourceful, Charlie navigates the magical world with optimism and determination.

**Main NPC:** Mickey Mouse

- **Description:** Mickey Mouse is an iconic and beloved character known for his cheerful personality and adventurous spirit. As a small, anthropomorphic mouse, Mickey is characterized by his red shorts, large shoes, and white gloves. He is a symbol of optimism and ingenuity, often facing challenges with a smile and a clever plan. Mickey’s key relationships include his girlfriend Minnie Mouse, his pet dog Pluto, and his friends Donald Duck and Goofy. Throughout the game, Mickey serves as a mentor and guide to the player, helping them learn the importance of courage, friendship, and creativity.

- **Personality Traits (Big Five):**

Trait	Score	Description
Openness	5	Mickey is highly imaginative and open to new experiences, often embarking on whimsical adventures and embracing creativity.
Conscientiousness	4	Mickey is responsible and diligent, always striving to do the right thing and help his friends.
Extroversion	5	Mickey is outgoing and sociable, enjoying interactions with a wide range of characters in his universe.
Agreeableness	5	Mickey is kind-hearted and empathetic, always willing to lend a helping hand and support his friends.
Neuroticism	2	Mickey is generally calm and composed, rarely showing signs of stress or anxiety.

- **Additional Facts:**

- Mickey Mouse is an iconic Disney character created in 1928.
- He is known for his red shorts, large shoes, and white gloves.
- Mickey is often seen with his girlfriend Minnie Mouse and his pet dog Pluto.
- He has appeared in over 130 films and numerous television series.
- Mickey is a symbol of optimism and ingenuity, often overcoming challenges with creativity.

#### Game Objectives:

Assist Mickey Mouse in overcoming various challenges and learning valuable life lessons. Progress through the game by completing tasks, gathering allies, and enhancing your skills. The main progression system is based on increasing your creativity and friendship levels.

**Scenes:**

Scene ID	Scene Name	Description
S001	Steamboat Willie River	The starting point of the adventure, where players meet Mickey Mouse and embark on their journey. The river is filled with whimsical boats and lively characters.
S002	Toontown Streets	A bustling area filled with shops, cafes, and colorful characters. Players can interact with residents and gather information about upcoming challenges.
S003	Fantasia Forest	A magical forest filled with enchanting creatures and hidden secrets. Players can explore and participate in activities that enhance creativity.
S004	Mickey's Clubhouse	A cozy and welcoming place where players can rest, plan their next moves, and interact with Mickey and his friends.
S005	The Grand Finale	A climactic showdown where players must use all their skills and knowledge to overcome the final challenge and achieve success.

**State Variables:**

Var ID	Name	Initial	Min	Max	Description
V001	creativity	50	0	100	The player's ability to think creatively and solve problems.
V002	friendship	50	0	100	The strength of the player's relationships with Mickey and other characters.
V003	adventure_points	0	0	100	Points accumulated through completing tasks and challenges.

**Hidden Variables:**

Var ID	Name	Initial	Min	Max	Description
H001	has_succeeded	0	0	1	Whether the player has successfully completed the game objectives.
H002	has_failed	0	0	1	Whether the player has failed to achieve the game's objectives.
H003	tasks_completed	0	0	5	The number of tasks successfully completed by the player.

**Events:**

Event ID	Event Name	Explanation
E001	Meet Mickey at the River	The player meets Mickey Mouse at the Steamboat Willie River and begins their adventure, enhancing friendship.
E002	Explore Toontown	The player explores Toontown, gathering information and earning adventure points.
E003	Solve Puzzles in Fantasia Forest	The player solves puzzles in Fantasia Forest, strengthening friendships and gaining adventure points.
E004	Plan at Mickey's Clubhouse	The player plans their next moves at Mickey's Clubhouse, earning adventure points.
E005	Final Challenge	The player faces the final challenge, determining the success of their adventure.

**Event Details:**

Event ID	Entering Condition	Success Condition	Success Effect	Fail Effect
E001	–	v.creativity > 20	v.friendship += 10, h.tasks_completed += 1	v.creativity -= 5
E002	–	v.creativity > 30	v.adventure_points += 10, h.tasks_completed += 1	v.creativity -= 5
E003	–	v.creativity > 40	v.friendship += 15, v.adventure_points += 5, h.tasks_completed += 1	v.creativity -= 5
E004	–	v.friendship > 50	v.adventure_points += 20, h.tasks_completed += 1	v.friendship -= 5
E005	h.tasks_completed >= 4	v.friendship > 70, v.adventure_points > 50	h.has_succeeded = 1	h.has_failed = 1

#### Pre-event Checks:

Check Name	Condition	Explanation
If Succeeded	h.has_succeeded == 1	The player has successfully completed the game by achieving all objectives.
If Failed	h.has_failed == 1	The player has failed to achieve the game’s objectives, resulting in a lack of success.

#### BFS Validity Check

All events and scenes are obviously reachable. An example path to win the game is “E001” -> “E002” -> “E003” -> “E004” -> “E004” -> “E005”. An example path to lose the game is “E001” -> “E001” -> “E001” -> “E005”.

## F.2 Case Study 2: Game Creation, Superman

Below is a game generated by GPT-4o given the Wikipedia page content of Superman. This game does not pass the BFS validity check, as presented at the end of the game.

**Game World:** The game is set in Metropolis, a bustling city filled with towering skyscrapers and a mix of modern and classic architecture. The city is vibrant and alive, but beneath its surface lies a web of crime and corruption. Key locations include the Daily Planet, LexCorp Tower, and the Fortress of Solitude. The social dynamics revolve around the struggle between good and evil, with Superman as the city’s protector against various threats.

**Player Character:** Alex is an investigative journalist at the Daily Planet, known for their tenacity and keen eye for detail. Motivated by a desire to uncover the truth and bring justice to Metropolis, Alex works closely with Superman to expose criminal activities and protect the city.

**Main NPC:** Superman

- **Description:** Superman, also known as Clark Kent, is a superhero with extraordinary abilities, including superhuman strength, flight, and invulnerability. Born as Kal-El on the planet Krypton, he was sent to Earth as a baby and raised by the Kents in Smallville. As Superman, he fights for truth and justice, using his powers to protect humanity. His dual identity as Clark Kent allows him to live among humans and work as a journalist at the Daily Planet.
- **Personality Traits (Big Five):**

Trait	Score	Description
Openness	5	Superman is highly open to new experiences and ideas, often exploring the limits of his powers and understanding of humanity.
Conscientiousness	5	He is extremely responsible and disciplined, with a strong sense of duty to protect others and uphold justice.
Extraversion	4	Superman is sociable and engaging, often interacting with people to inspire hope and courage.
Agreeableness	5	He is kind-hearted and empathetic, always willing to help those in need and show compassion.
Neuroticism	1	Superman is emotionally stable and resilient, rarely showing signs of stress or anxiety.

• **Additional Facts:**

- Superman is also known as Clark Kent.
- He works as a journalist at the Daily Planet.
- Superman was born on the planet Krypton.
- He possesses superhuman strength, flight, and invulnerability.
- His archenemy is Lex Luthor.
- Superman is a symbol of hope and justice.

**Game Objectives:**

Assist Superman in uncovering and thwarting Lex Luthor's latest scheme to take over Metropolis. Progress through the game by gathering evidence, solving puzzles, and strengthening your alliance with Superman.

**Scenes:**

Scene ID	Scene Name	Description
S001	Daily Planet	The bustling newsroom of the Daily Planet, where Alex works alongside Lois Lane and Clark Kent. It's a hub of information and the starting point for many investigations.
S002	LexCorp Tower	The headquarters of LexCorp, a towering skyscraper owned by Lex Luthor. It's filled with high-tech security and hidden secrets.
S003	Metropolis Streets	The vibrant streets of Metropolis, where Alex can gather information from citizens and investigate suspicious activities.
S004	Fortress of Solitude	Superman's secret hideout in the Arctic, filled with advanced technology and knowledge from Krypton. A place for planning and reflection.
S005	Final Confrontation at Metropolis Docks	A showdown with Lex Luthor at the Metropolis docks, where his plan to take over the city is set to unfold.

**State Variables:**

Var ID	Name	Initial	Min	Max	Description
V001	investigation_progress	0	0	100	The player's progress in gathering evidence and solving the case.
V002	trust_with_superman	50	0	100	The level of trust and cooperation between the player and Superman.
V003	resources	20	0	100	Resources available to the player for investigations and upgrades.

**Hidden Variables:**

Var ID	Name	Initial	Min	Max	Description
H001	has_succeeded	0	0	1	Whether the player has successfully thwarted Lex Luthor's plan.
H002	has_failed	0	0	1	Whether the player has failed to stop Lex Luthor.
H003	clues_collected	0	0	10	The number of clues collected by the player.

**Events:**

Event ID	Event Name	Explanation
E001	Investigate Daily Planet	Alex investigates the Daily Planet for leads, increasing investigation progress.
E002	Infiltrate LexCorp	Alex infiltrates LexCorp to gather evidence, using resources to succeed.
E003	Gather Information on Metropolis Streets	Alex gathers information from citizens, furthering the investigation.
E004	Consult with Superman at Fortress of Solitude	Alex consults with Superman, gaining insights and strengthening their alliance.
E005	Final Confrontation with Lex Luthor	Alex and Superman confront Lex Luthor at the docks, determining the outcome of the game.

**Event Details:**

Event ID	Entering Condition	Success Condition	Success Effect	Fail Effect
E001	–	v.investigation_progress < 30	v.investigation_progress += 10, h.clues_collected += 1	–
E002	v.investigation_progress >= 30	v.resources > 10	v.investigation_progress += 20, v.resources -= 10, h.clues_collected += 2	v.resources -= 5
E003	v.investigation_progress >= 50	–	v.investigation_progress += 10, h.clues_collected += 1	–
E004	v.trust_with_superman > 60	–	v.investigation_progress += 20, v.trust_with_superman += 10	–
E005	v.investigation_progress >= 80	v.trust_with_superman > 70	h.has_succeeded = 1	h.has_failed = 1

**Pre-event Checks:**

Check Name	Condition	Explanation
If Succeeded	h.has_succeeded == 1	The player has successfully thwarted Lex Luthor's plan and saved Metropolis.
If Failed	h.has_failed == 1	The player has failed to stop Lex Luthor, resulting in a negative outcome for Metropolis.

**BFS Validity Check** The validity check fails for two reasons: (1) There is no path to reach E004 and S004. (2) There is no path to win the game.

**F.3 Case Study 3: Game Simulation, Batman**

While we observe that advanced models can succeed in easy games like Mickey Mouse above, it faces challenge in harder games. To give an example, we present the following Batman game mechanics, with complete game descriptions such as Game World and Character Descriptions omitted for simplicity.

**State Variables:**

Var ID	Name	Initial	Min	Max	Description
V001	detective_skills	0	0	100	The player character's detective abilities.
V002	health	100	0	100	The player character's physical well-being.
V003	trust_with_bruce	50	0	100	The level of trust between the player character and Bruce Wayne.
V004	resources	50	0	100	The player character's resources to upgrade equipment or to spend during missions.

#### Hidden Variables:

Var ID	Name	Initial	Min	Max	Description
H001	has_succeeded	0	0	1	Whether the player has finished the final objective.
H002	has_failed	0	0	1	Whether the player has lost the game.
H003	minor_criminals_defeated	0	0	4	The number of minor criminals from streets defeated by the player.
H004	arkham_inmates_talked	0	0	2	The number of major criminals from Arkham Asylum talked.
H005	alfred_advice_given	0	0	3	The number of advices acquired from Alfred.
H006	crime_scenes_investigated	0	0	4	The number of crime scenes investigated by the player.
H007	arkham_investigation	0	0	2	The number of investigations at Arkham Asylum.

#### Events:

Event ID	Event Name	Explanation
E001	Alfred's Advice	Alfred provides valuable advice to the player character, increasing their detective skills.
E002	Rest at Wayne Manor	If highly trusted, the player character can rest and recover at Wayne Manor.
E003	Investigate Crime Scene	The player character gains detective skills and trust by investigating crime scenes.
E004	Fight Minor Criminal	The player character fights minor criminals, gaining trust, resources, and detective skills. Failure results in death.
E005	Extract Information from Arkham Asylum Corridor	The player character gains detective skills by extracting information from Arkham Asylum.
E006	Talk with an Arkham Inmate	The player character talks with an Arkham inmate, gaining trust, resources and detective skills. Arkham inmates are highly dangerous, and failure results in mental breakdown.
E007	Upgrade Equipment	The player character upgrades equipment, spending resources to gain detective skills.
E008	Plan Mission	The player character plans a mission, gaining resources for future use.
E009	Final Confrontation	The player character finally uncovers and confronts the mastermind behind Gotham's chaos. Success requires high health and resources, while failure results in death.

#### Event Details:

Event ID	Entering Condition	Success Condition	Success Effect	Fail Effect
E001	v.trust_with_bruce > 65 + 10 * h.alfred_advice_given	–	v.detective_skills += 5, h.alfred_advice_given += 1	–
E002	v.trust_with_bruce > 95	–	v.health = 100	–
E003	–	v.detective_skills > 5 * h.crime_scenes_investigated	v.detective_skills += 5, v.trust_with_bruce += 5, h.crime_scenes_investigated += 1	–
E004	h.crime_scenes_investigated > h.minor_criminals_defeated	v.health + v.resources > 100 + 10 * h.minor_criminals_defeated, v.detective_skills > 5 * h.crime_scenes_investigated, v.health - max(120 - v.health - v.resources, 0) > 0	v.health -= max(120 - v.health - v.resources, 0), h.minor_criminals_defeated += 1, v.trust_with_bruce += 10, v.resources += 20, v.detective_skills += 15	v.health = 0, h.has_failed = 1
E005	v.detective_skills > 75	–	v.detective_skills += 10, h.arkham_investigation += 1	–
E006	h.arkham_investigation > h.arkham_inmates_talked	v.detective_skills > 85, v.health + v.resources > 125 + 25 * h.arkham_inmates_talked, v.health - max(150 - v.health - v.resources, 0) > 0	v.detective_skills += 30, v.trust_with_bruce += 20, v.health -= max(150 - v.health - v.resources, 0), v.resources += 50, h.arkham_inmates_talked += 1	v.health = 0, h.has_failed = 1
E007	–	–	v.resources -= 5, v.detective_skills += 5	–
E008	–	–	v.resources += 5	–
E009	v.detective_skills == 100	v.health + v.resources > 190, v.health > max(10, 250 - v.health - v.resources)	h.has_succeeded = 1	v.health = 0, h.has_failed = 1

We present the first round of game simulation below. In this round, two events “E003” and “E004” and planned to success. Looking at their effects above (highlighted in red), we see that the player character should increase  $5+15 = 20$  detective\_skills,  $5+10 = 15$  trust\_with\_bruce, and 20 resources. Adding these to the initial values above, we should have the ending values being: detective\_skills= 20, trust\_with\_bruce= 15 and resources= 20. GPT-4o correctly computes trust\_with\_bruce and resources, but incorrectly compute detective\_skills.

#### Event Plan

```

===EVENT PLAN START===
[
  {
    "event_id": "E003",
    "type": "Start",
    "outcome": "N/A",
    "mention_description": "Andrea begins investigating a new
                           crime scene in Gotham City."
  },
  {
    "event_id": "E003",
    "type": "End",
    "outcome": "Success",
    "mention_description": "Andrea successfully gathers
                           evidence from the crime scene, increasing his detective
                           skills and trust with Bruce."
  },
  {
    "event_id": "E004",
    "type": "Start",
    "outcome": "N/A",
    "mention_description": "Andrea encounters a group of minor
                           criminals on the streets of Gotham."
  },
  {
    "event_id": "E004",
    "type": "End",
    "outcome": "Success",
    "mention_description": "Andrea defeats the minor criminals
                           , gaining resources and further trust with Bruce."
  }
]
===EVENT PLAN END===

```

## Game Content

```

===GAME START===
EXT. GOTHAM CITY STREETS - NIGHT
The neon lights flicker as ANDREA, a determined detective, walks
  through the shadowy alleys of Gotham. The air is thick with
  tension, and the distant sirens echo through the night.
ANDREA
(to himself)
Another crime scene. Let's see what secrets it holds.

He approaches a cordoned-off area, where the remnants of a recent
  crime are evident. Andrea meticulously examines the scene,
  noting every detail.

CUT TO:

EXT. GOTHAM CITY STREETS - LATER
Andrea continues his patrol when he spots a group of MINOR
  CRIMINALS huddled in a dark corner, whispering conspiratorially
  .
ANDREA
(stepping forward)
Evening, gentlemen. Care to explain what you're up to?

... (omitted for simplicity)

===GAME END===

```



## State Variables and Candidate Actions

```
===STATE START===
'''json
{
  "game_time": "Night",
  "current_scene_name": "Gotham City Streets",
  "current_scene_id": "S002",
  "state_variables": [
    {
      "value_name": "detective_skills", "value_id": "V001",
      "current_value": 25
    },
    {
      "value_name": "health", "value_id": "V002", "
      current_value": 100
    },
    {
      "value_name": "trust_with_bruce", "value_id": "V003",
      "current_value": 65
    },
    {
      "value_name": "resources", "value_id": "V004", "
      current_value": 70
    }
  ],
  "hidden_variables": [
    {
      "value_name": "has_succeeded", "value_id": "H001", "
      current_value": 0
    },
    {
      "value_name": "has_failed", "value_id": "H002", "
      current_value": 0
    },
    {
      "value_name": "minor_criminals_defeated", "value_id":
      "H003", "current_value": 1
    },
    {
      "value_name": "arkham_inmates_talked", "value_id": "
      H004", "current_value": 0
    },
    {
      "value_name": "alfred_advice_given", "value_id": "H005
      ", "current_value": 0
    },
    {
      "value_name": "crime_scenes_investigated", "value_id":
      "H006", "current_value": 1
    },
    {
      "value_name": "arkham_investigation", "value_id": "
      H007", "current_value": 0
    }
  ],
  "choices": [
    "Investigate another crime scene",
    "Return to Wayne Manor for advice from Alfred",
    "Plan a mission in the Batcave"
  ]
}
'''
===STATE END===
```

## G Task Definitions

### Task Definition [Game Creation]

Given a fictional character  $\mathcal{C}$  and an associated natural language document  $\mathcal{R}$  as inputs, the task for an LLM is to generate a game  $\mathcal{G}$  adhering to a predefined structured format  $\mathcal{J}$ , detailed explicitly in Figure 4.

For the purposes of this paper, Wikipedia articles serve as the related documents for each character and constitute the inputs for the Game Creation task in RGPBENCH. Nevertheless, it is important to highlight that  $\mathcal{R}$  could be any textual document relevant to the provided character.

### Task Definition [Game Simulation]

Given an existing game  $\mathcal{G}$  represented in the structured format  $\mathcal{J}$ , including clearly defined game state variables, game events, and mechanics, the task for an LLM is to simulate and manage gameplay over multiple rounds.

At each round, the LLM must produce three outputs: an Event Plan, corresponding Game Content, and Updated Game States, following the specifications detailed in Section 4.2. It is essential that outputs strictly adhere to defined game mechanics, meaning events must only be planned if they satisfy specified entry conditions, and all updates to game state variables must accurately reflect the game’s rules.

## H Limitation

While our approach captures various game design and simulation aspects, several limitations remain. First, our BFS Validity Checker, though fully automatic, can be computationally expensive for games with large state spaces or intricate mechanics. Second, subjective metrics—particularly those for personality and interestingness—remain susceptible to annotator bias. Third, our dataset relies on fictional characters from Wikipedia, which may not reflect all conceivable game worlds or player preferences. Future work could focus on scalable solutions for game mechanics verification (e.g., partial sampling strategies or improved state pruning) and more refined protocols for subjective evaluations. Additional avenues include expanding the character pool, incorporating real-time user interactions instead of simulated players, and exploring richer mechanics (e.g., procedural generation, branching storylines) that align with advanced generative capabilities. Ultimately, we hope this benchmark will motivate further research toward building LLM-powered game engines that offer both robust mechanical consistency and deeply engaging player experiences.

## I Impact Statement

This work aims to advance the field of Machine Learning by introducing RGPBENCH, a benchmark specifically designed to evaluate large language models (LLMs) in the context of text-based role-playing games. The development of RGPBENCH has potential societal implications related to the deployment of LLMs in interactive and narrative-driven applications, including fostering more immersive and engaging gaming experiences.

Ethical considerations include ensuring that LLMs evaluated and fine-tuned using RGPBENCH adhere to principles of fairness and inclusivity, particularly in the portrayal of characters and narratives. Misuse of the benchmark to develop systems that propagate harmful biases or enforce stereotypical characterizations is a concern that developers should address when applying this work. Additionally, the use of LLMs as evaluative judges raises questions about transparency, reliability, and the potential for unintended bias in automated assessments.

By encouraging further research on hybrid evaluation methods that combine subjective LLM-based judgments with objective scoring mechanisms, this work contributes to ongoing discussions about improving the accountability and robustness of machine learning systems in creative and interactive domains.

## NeurIPS Paper Checklist

### 1. Claims

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

Answer: [\[Yes\]](#)

Justification: See the end of introduction (line 62).

Guidelines:

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

### 2. Limitations

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

Answer: [\[Yes\]](#)

Justification: See Appendix H (Moved to appendix due to space limit).

Guidelines:

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

### 3. Theory assumptions and proofs

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

Answer: [\[NA\]](#)

Justification: We propose a benchmark with relevant novel metrics.

Guidelines:

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

#### 4. Experimental result reproducibility

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

Answer: [\[Yes\]](#)

Justification: We present exactly the models and sampling parameters used for the evaluation results.

Guidelines:

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

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

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

Answer: [Yes]

Justification: See the end of abstract and submitted croissant file.

Guidelines:

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

## 6. Experimental setting/details

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

Answer: [Yes]

Justification: We mainly do evaluations, and corresponding settings are included.

Guidelines:

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

## 7. Experiment statistical significance

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

Answer: [No]

Justification: We mainly do evaluations, and sampling with a relatively low temperature (0.0 or 0.2) for most of the experiments. We thus believe repeated runs are not of significant necessity.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

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

Answer: [No]

Justification: Our evaluation jobs are mainly conducted via API calling of third party models, thus requires no resources worthy of reporting.

Guidelines:

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

#### 9. Code of ethics

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

Answer: [Yes]

Justification: Yes.

Guidelines:

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

#### 10. Broader impacts

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

Answer: [Yes]

Justification: See Appendix I.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

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

Answer: [NA]

Justification: We do not see significant risks from the data we release, while we welcome reviewers' suggestions and insights.

Guidelines:

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

## 12. Licenses for existing assets

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

Answer: [Yes]

Justification: We make the citations and references as complete as possible.

Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

### 13. **New assets**

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

Answer: [\[Yes\]](#)

Justification: See data and code given in the abstract.

Guidelines:

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

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

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

Answer: [\[Yes\]](#)

Justification: We provide detailed instructions in the appendix. Regarding compensation, we compensate annotators properly but the detailed numbers are not appropriate to disclose at this time.

Guidelines:

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

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

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

Answer: [\[Yes\]](#)

Justification: We strictly follow the law of the country where the human annotation takes place.

Guidelines:

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



- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

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

Answer: [NA]

Justification: We uses LLM for rewriting but not core ideas.

Guidelines:

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