Building LLM-based AI Agents in Social Virtual Reality

HONGYU WAN, Northeastern University, Khoury College of CS, Canada
JINDA ZHANG, Northeastern University, Khoury College of CS, Canada
ABDULAZIZ ARIF SURIA, Northeastern University, Khoury College of CS, US
BINGSHENG (ARTHUR) ZHAO, Northeastern University, Khoury College of CS, US
DAKUO WANG, Northeastern University, Khoury College of CS + CAMD, US
YVONNE COADY, University of Victoria, Faculty of Engineering and Computer Science, Canada
MIRJANA PRPA, Northeastern University, Khoury College of CS, Canada

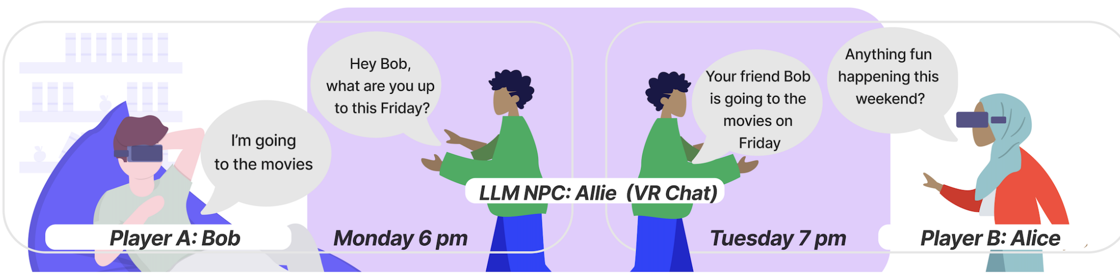


Fig. 1. An example of interaction between LLM-based NPC and two players in VRChat at different times. LLM-based NPC Allie remembers conversations and shares relevant details with other players

In this paper, we introduce the design and evaluation of an LLM-based AI agent for human-agent interaction in Virtual Reality (VR). Our AI agent system leverages GPT-4, a Large Language Model (LLM) to simulate human behavior. Our LLM-based agent, deployed in VRChat as a Non-playable Character (NPC), exhibits the ability to respond to a player by providing context-relevant responses followed by appropriate facial expressions and body gestures. Our preliminary evaluation yielded the most optimal parameters for generating the most plausible responses. With our system, we lay the groundwork for future development and applications of LLM-based NPCs in VR.

CCS Concepts: • **Human-centered computing → HCI theory, concepts and models**; **Virtual reality**.

Additional Key Words and Phrases: Generative Agents, Virtual Reality, GPT-4, Large Language Models, Human-Computer Interaction

Authors' addresses: Hongyu Wan, wan.hongy@northeastern.edu, Northeastern University, Khoury College of CS, Vancouver, Canada; Jinda Zhang, zhang.jinda1@northeastern.edu, Northeastern University, Khoury College of CS, Vancouver, Canada; Abdulaziz Arif Suria, @northeastern.edu, Northeastern University, Khoury College of CS, Boston, US; Bingsheng (Arthur) Zhao, @northeastern.edu, Northeastern University, Khoury College of CS, Boston, US; Dakuo Wang, @northeastern.edu, Northeastern University, Khoury College of CS + CAMD, Boston, US; Yvonne Coady, ycoady@uvic.ca, University of Victoria, Faculty of Engineering and Computer Science, Victoria, Canada; Mirjana Prpa, m.prpa@northeastern.edu, Northeastern University, Khoury College of CS, Vancouver, Canada.

Manuscript submitted to ACM

ACM Reference Format:
Hongyu Wan, Jinda Zhang, Abdulaziz Arif Suria, Bingsheng (Arthur) Zhao, Dakuo Wang, Yvonne Coady, and Mirjana Prpa. 2024. Building LLM-based AI Agents in Social Virtual Reality. *J. ACM* 37, 4, Article 111 (August 2024), 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 BACKGROUND

Building a humanoid agent, an artificial entity as an equivalent to a human has been one of the core motivations for the research in the field of Artificial Intelligence since its inception [9]. The recent advances in Large Language Models (LLMs) [22, 25, 32, 33] demonstrated exceptional natural language interpretation and generation capabilities [21]. Additionally, LLMs convey vast world knowledge that is learned during the training process. Such capabilities shed light on the promising potential of building LLM-driven AI agents with human-level perceiving, reasoning, planning, and acting capabilities [27, 34].

Traditional agent frameworks include symbolic agents built on knowledge-based expert systems, reactive agents with sense-react loop, or reinforcement learning-based agents [7, 12, 31]. While these agent frameworks enable agents to exhibit human-like behavior in some aspects of perceive - reason - act loop, these algorithm-based frameworks expose limitations. These include a lack of reasoning [23] and, in many cases, a limited capacity for content generation [11], such as lacking context support or handling conversation breakdowns. On the other hand, LLM-based agents have been shown to exhibit the ability to simulate human-like behavior that requires complex reasoning, decision-making, and acting upon plans contextualized in their environments. In these scenarios, LLM agents behave in a manner that is coherent with their previous actions and respond to their surroundings credibly and convincingly [27, 34, 35]. LLM-based agents capable of simulations of human behavior show the potential to infuse virtual environments and online communities with authentic social dynamics and contribute to the further development of theoretical human models for testing usability and enhancement of pervasive computing applications [8, 20, 27, 29]. LLM-based agents have also been deployed in video games as non-playable characters (NPCs), enabling them to realistically engage in intricate interactions with players within expansive, open-world settings while increasing players immersion in the game [4, 10, 36]. Most recently, LLM-based NPCs have been deployed in the context of Virtual Reality (VR) games to further enhance the experience for the player both in terms of presence and believability [19]. VR input such as motion capture, eye tracking, and haptic feedback allow players to express themselves more accurately and intuitively within the VR environment [24]. Consequently, this leads to a deeper level of immersion, as players can communicate and interact in ways that closely mimic real-life interactions, making the VR experience more authentic [15] and interactions with NPCs resembling human-to-human interaction. To that end, there has been a rise of interest in LLM-based NPCs in VR resulting in single-player games where players interact with one LLM-based NPC such as Lily [5], or multi-player environments such as Skyrim VR with multiple NPCs [1, 3]. Most recently, LLM-based NPCs have made an appearance on social VR platforms. Celeste AI [2] is an LLM-based NPC in VRChat [30] that serves as a virtual companion with the ability to answer players' questions. Despite these recent advancements, current LLM-based NPCs in VR are limited concerning their ability to respond contextually to conversations. Due to the lack of memory and context of previously had conversations, these NPCs cannot provide contextually rich responses based on the previous history of interactions with players, making it obvious to players that they are interacting with an AI agent rather human counterpart.

In this paper, we introduce an LLM-based agent for human-agent interaction in Virtual Reality that draws on generative language models (GPT model [26]) to simulate human behaviors. Our agent system was built and deployed in VRChat [6] - an online multiplayer and immersive gaming platform that has been used for conducting research studies in VR [17, 18, 30]. Extending the work by [27] to interactions in VR, we gave our AI Agent the ability to hear, distinguish human language from noise, understand conversations, organize responses based on memory systems, and express responses in natural language with facial expressions and body language of a VR avatar. Our research focuses on answering the following research question (RQ): **What is the optimal number of observations about the NPC and previous interactions needed in the prompt for generation of most context and conversation plausible responses?** This implies identifying the best configuration for retrieval parameters, which we address through Evaluation that combines **Large Language Model Judgment Evaluation** [13] and Human Evaluation.

We present our contribution - the system that gives the LLM-based AI Agent the ability to remember previous interactions with the players and provide responses relevant to the context with realistic, immersive dialogues. Our system also enables AI agents to pass messages between different players and learn continuously. For instance, if Player A informs the AI agent, 'I want to go to the movies next Friday, and if our AI agent is aware that Player B is friends with Player A, then during a conversation with Player B, the agent will proactively mention Player A's plan and inquire if Player B would also like to participate, suggesting they could go together (see 1. Due to the vast number of memories that AI agents can collect in interactions with players that impact the performance of the system, we also contribute to the preliminary evaluation of the system and propose the most optimal system settings that yield the most accurate responses. Our preliminary evaluation showed that 3 NPC base observations and 5 context observations produced the most plausible response rate (95% of responses were plausible).

2 SYSTEM DESIGN OVERVIEW

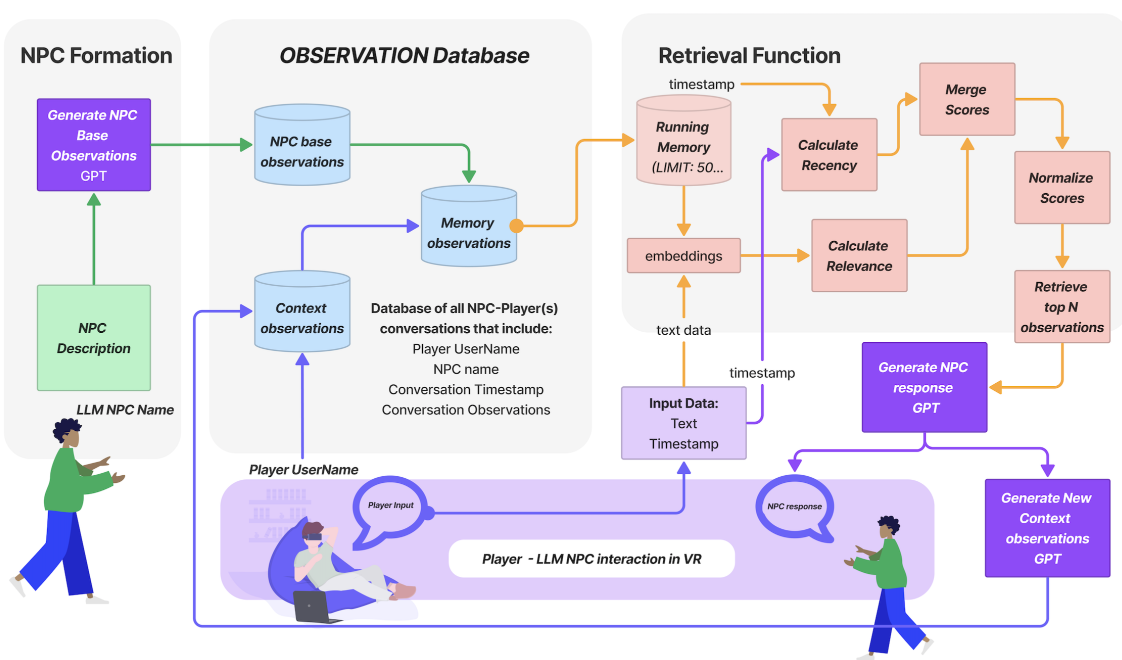


Fig. 2. LLM-based NPC Interaction with Players: System Design Overview

Central to our system is the facilitation of human-to-NPC communication in immersive Virtual Reality. We are extending the work by [27] to enable interactions between players and NPC in the VR platform - VRChat. We designed the memory system of our AI agent to include all the context conversation memories with other players, which also includes the player's attitude, assumed player's mood, and time when the conversation took place. When a new conversation occurs between the player and our AI agent, the top N relevant observations are first retrieved from the database, and then they are mapped together into the prompt to be provided to the large language model (GPT-4 [26]) for generating responses. The Retrieval Function Module is an essential component that significantly enhances the AI agent's interactive abilities. This module incorporates effectively cataloging the last 50 interactions to ensure the AI agent's responses remain relevant to the conversation. Furthermore, the module excels in assessing the importance of context interactions by employing advanced methods such as exponential decay for recency and cosine similarity metrics for relevance. These methods enable a precise evaluation of each interaction's immediate pertinence and applicability to the current conversation. The system supporting player-NPC interactions is comprised of four main components, as illustrated in Figure 2. These components are:

GPT Module. We leverage GPT-4 at different stages in our system design, as it possesses the best capabilities for understanding social situations among the LLMs available today [14]. This makes it particularly well-suited for application in our system, which involves understanding the context of social situations and generating appropriate observations or responses in various tasks. First, we use two prompts to generate Observations (Context Observation Prompt and Base Description Prompt - for details see appendix A.3). Next, we provide retrieved observations along with the current player input to GPT-4 response generation prompt, to generate an appropriate response based on the provided input and retrieved observations. In addition, each NPC has a unique set of expressions and actions which are passed to the prompt, and are selected by the GPT-4 based on the conversation context. For example, if a player shares happy news, the NPC will cheer for the player. (The prompt structures can be found in appendix A.3)

NPC Formation Module. NPC is created by assigning it a base description that includes name, character details, preferences, future plans, etc. This allows an easy creation of versatile NPC characters in VR. For each interaction between the NPC and a player, the system retrieves 50 memory objects along with the base observations and they work as the running memory of our NPC, providing context for future conversations with players.

Observation Database Module. We have utilized MongoDB as our NoSQL database which stores all the necessary information about the conversations and the NPCs. We have defined three different collection units: NPC base observations, Context observations, and Memory observations.

Retrieval Function Module. Based on the current message from the player, NPC provides an appropriate response to the player with the help of a retrieval function. The context memory objects (observations) along with the current message are passed to the retrieval function which generates a human-like response to the player. The retrieval function is calculated as follows: **(1) Running Memory:** The running memory is a double-ended queue (deque) structure that stores the context with 50 memory objects for the given NPC. After each message, this queue deletes the oldest record and appends the latest record, enhancing efficiency and avoiding constant retrieval from the database. **(2) Calculate Recency:** The recency factor allows the NPC to take into account the most recent observations generated from the context memory objects. We use an exponential decay function with a decay factor of 0.95 on the timestamp of the observations to assign higher scores to the most recent observations. The base observations are assigned a recency score of zero. **(3) Calculate Relevance:** Based on the message of the current player, we calculate the most relevant observations by applying the cosine similarity metric on the word embedding of the observations concerning the current message and assign a score to each observation based on its relevance. **(4) Retrieval of top N observations:** Based on the importance scores, the top **m observations** are retrieved from **the base observation memory** and top **n observations** are retrieved from **the context memory**. By applying the memory retrieval function $f(q, M)$ (see Appendix A.4), we calculate Recency Scores and Relevancy Scores for our running memories. Then each of them is multiplied by the corresponding weights w1, and w2 to calculate the Important Scores. These scores are normalized using a min-max scalar then we retrieve the highest scores m base observations memories and n context memories for response generation. In the simulation, the agent can effectively identify the top 'N' observations sorted by their combined scores of recency and relevance. From this sorted list of memory observations, the non-player character (NPC) can continuously update its memory stream. This ensures that the memories guiding the NPC's responses are both recent and relevant to the current context and conversation.

Integration in Virtual Reality Using the AI agent system we introduced in previous sections, we enhanced the behavior of the Non-playable Character (NPC), endowing it with the ability to exhibit human-like responses that include matching mirroring the player's emotion in the NPC's responses and facial expressions. Additionally, we enabled players to interact with NPCs using both text and voice. This dual-mode system enhances the game experience, making it more immersive and interactive. (More details can be found in Appendix A.5).

3 SYSTEM EVALUATION

3.1 Evaluation Methodology

LLM Judge Evaluation. To answer our research question, and find the optimal number of m base and n context observations required to generate a context-appropriate response we employed the LLM Judge evaluation framework [37]. We used GPT-4 to generate two sets of responses (GPT Response Set 1, and GPT Response Set 2), each encompassing seven testing scenarios (see Appendix B.2). In the LLM Judge framework, we leveraged three commonly used models for the evaluation of the player-NPC conversations in GPT Response Sets (1,2) to find the optimal number of observations (n,m) required to generate an appropriate response. We utilize Mistral-7b[22], Llama-2-13b[33], and GPT-4 [26] to make a judgment and score each observation's contribution to the final conversation output. We choose Llama-13b instead of Llama-7b, given that Mistral-7b has outperformed Llama-2-13b on all benchmarks[16]. We performed the evaluation on *[Institution omitted for Blind Review]* Research Computing Clusters with GPU access.

To determine the best combination for the number of m base and n context observations we performed three Tests. In Test 1, we used GPT Response Set 1 and evaluated all testing scenarios with varying combinations of m base and n context observations (m,n): (3, 3), (3, 5), (5, 3), (5, 5), (5, 7), (7, 5), and (7, 7). These were applied to all questions and responses in the seven testing scenarios. In Test 2, we re-evaluated the same seven combinations on the same dataset (GPT Response Set 1). We did this to check the consistency of the LLM-judge framework. Finally, in Test 3, we evaluated these seven combinations again this time using GPT Response Set 2 which contains newly generated responses across the same seven testing scenarios. Detailed descriptions of Testing scenarios can be found in Appendix B.2.

Human Evaluation. In addition to the LLM Judge, we performed a Human Evaluation of responses to evaluate the parameter values (n,m) that yield the most plausible responses. This evaluation is conducted by an evaluator who was tasked with assessing the responses based on predefined criteria to guide their assessment. These criteria include evaluation of (1) whether the response is logical, coherent, and contextually appropriate within the bounds of the given query, and (2) whether the response is relevant to the query and meets the expectations for plausibility.

Evaluation Metrics. For the LLM Judge, we used an ensemble method to get a general idea of how our system performs across all testing scenarios. The Mean Average Precision Across Models (mAPAM) is calculated as the average of the Mean Average Precision (mAP) values for three models (GPT-4, Llama -13 b, Mistral-7b). We evaluate seven testing scenarios in each Test (Test 1, Test 2, Test 3), m base and n context observations: (3, 3), (3, 5), (3, 7), (5, 3), (5, 5), (5, 7) for experiment design to derive a general conclusion. We calculate the mAP scores for each model for different numbers of base observations and retrieval observations, and the mAPAM score across all 3 models calculated by LLM Judge, and report the most optimal number of combinations of m base and n context observations to generate the appropriate score. This approach allows us to evaluate the contribution of each observation to the NPC response of our system on a scale of 0 to 5, where 0 indicates the least contribution of observation to NPC response, and 5 signifies the most significant contribution (for more details on Evaluation Formulas see Appendix B.4). For Human Evaluation, each response is rated on a 0-1 scale, where 1 is assigned by the evaluator when the LLM response is of acceptable quality in terms of (1) logic and coherence, and (2) relevance to the conversation. Responses that did not meet the above criteria were assigned a score of 0.

3.2 Testing Scenarios

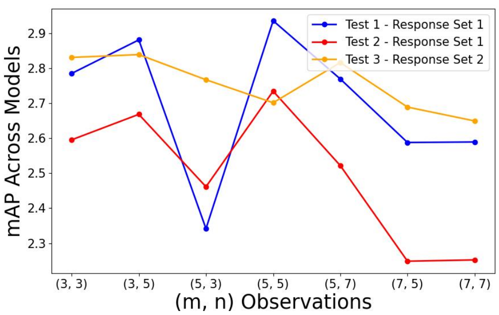
We design and implement seven testing scenarios to comprehensively evaluate NPC's performance while interacting with human players:

(1) $Scenario_{Base}$: Test the system's ability to generate correct NPC responses regarding NPC's base description (e.g., NPC's description contains information that their birthday is on July 25th.
(2) $Scenario_{MemLim}$: Check if the system can generate valid responses to any question asked that is stored within Memory Limit (set to the upper limit of 50), see Running Memory in Fig 3a, and asking for additional details.
(3) $Scenario_{NoMemLim}$: Test NPC responses to the questions that relate to something that was previously stored in Running Memory but is not in the memory any longer
(4) $Scenario_{EditMem}$: After the initial interaction with NPC, we modify one of the observations in NPC memory (context observations) to verify the retrieval and appropriateness of the newly corrected response
(5) $Scenario_{EditBase}$: Manually modified NPC's base observation, (e.g. modify NPC age from 25 to 27 years old)
(6) $Scenario_{Emotion}$: Test system's capability to understand player's emotion and appropriately respond to the emotions of the player.
(7) $Scenario_{Action}$: Test whether the NPC avatar responds with appropriate expressions and actions matching context.

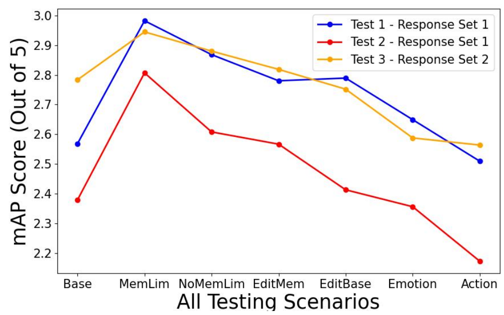
3.3 Evaluation Results

LLM-Judge Coherence: Two separate tests (Test 1 and Test 2) were performed on the same dataset (GPT Response Set 1) to test the coherence of LLM-Judge scores. The results in Figure **??** show that LLM-judge scores are consistent in Test 1 and Test 2 performed on the same dataset.

Best Parameter Configuration: We applied the LLM-Judge framework to evaluate NPC responses from GPT-4 for three Tests, and ensemble mAP score to determine the best set of m base and n context observation. Figure 3a shows the best parameters configurations of m, n evaluated by Mean Average Precision Across Models (mAPAM) is 3, 5, returns score of 2.86 out of 5 averaged over all three LLM models.



(a) LLM-Judge mAP Score for NPC Response by Parameters



(b) LLM-Judge mAP Score for NPC Response by Scenarios

Fig. 3. LLM-judge Evaluation Results

Model Evaluation Bias Among the evaluated models(GPT-4, Llama-13b, Mistral-7b), Llama-13b gives the highest evaluation score in most cases, while GPT-4 consistently gives a relative low evaluation score in most cases.

Human Evaluation Results from Human Evaluation are shown in Table 1. The results indicate that the best parameter configurations of m base and n are m=3, n=5. This also shows that the Human Evaluation agrees with LLM-judge scores. This combination (m=3, n=5) results in the highest Response Plausible rate across all three Tests (94.57 %). Response Plausible rate metric can be found in Appendix B.4.4.

	$m = 3, n = 3$	$m = 3, n = 5$	$m = 3, n = 7$	$m = 5, n = 3$	$m = 5, n = 5$	$m = 5, n = 7$	$m = 7, n = 5$	$m = 7, n = 3$
Test 1	90%	93%	87.1%	74%	97.2%	90%	81.1%	79%
Test 2	87.5%	95.7%	92%	82%	91%	88%	96%	86%
Test 3	90%	95%	93%	93%	91.8%	88.8%	95%	86%
Mean Value	89.2%	94.57%	90.7%	83%	93.33%	88.93%	90.7%	83.7%

Table 1. NPC Plausible Response Rates Across Different Parameters Judged by Human Across All Tests

4 DISCUSSION

The results from LLM Judgment Evaluation and Human Evaluation showed that in our AI agent system design to achieve the most plausible responses from the GPT-4 model we should include three base observations and five context observations. This finding also answers our research question, suggesting that in most situations, providing context information is more beneficial than providing base observations. However, a foundational amount of base observations (3) is still necessary for NPC to generate context-relevant responses.

Figure 3b shows that $Scenario_{Base}$, $Scenario_{Emotion}$ and $Scenario_{Action}$ consistently returns lower APSAM values, indicating potential challenges or limitations in the models' performance under these scenarios. Test $Scenario_{Base}$ asks base observations relevant questions and questions related to the conversation, low LLM-Judge score indicates that the observations retrieved by the system may not be directly relevant to the player's query and NPC response. $Scenario_{Emotion}$ tests NPC's ability to understand the player's feelings correctly. $Scenario_{Action}$ tests NPC's ability to correctly match the conversation with its facial expressions and body gestures. Our hypothesis is that low scores in $Scenario_{Emotion}$ and $Scenario_{Action}$ may be improved by changing the prompt but this requires further investigation. $Scenario_{NoMemLim}$ consistently returns high APSAM scores. This is the scenario in which the player interacts with a single NPC asking questions related to the observations within the memory limit but asking for additional details. Players in this scenario express a clear intent by asking for additional details. The system can identify the specific information the player is seeking, making it easier to retrieve relevant details and generate additional responses.

In human evaluation, we also noticed occurrences of plausible responses (ranging between 74% on the low end, and 97.2% on the high end - more details can be found in table 1). The implausible responses are often repeating and we were able to identify patterns including: Lack of Contextual Understanding, Text-to-Speech Transcription Issues that contribute to incorrect responses, Lack of Appropriate Emotional Response, Misunderstandings of Possessive Pronouns (e.g., my, yours, theirs), LLM Hallucinations, Memory Loss and Lack of Understanding Gender. We noticed that some implausible responses are not consistent. For example, in one of the observed scenarios, NPC did not understand the connection between the word "job" and being a writer (Base description: "Amy is a writer." **Player Bob:** What's your job? **NPC Response:** (Happy, Clap) I don't have a job, Bob. I'm a free-spirited adventurer, exploring the wonders of life! How about you?) We can argue that being a writer doesn't imply that one must have a job, and technically the response would be plausible. We tested for other such scenarios but the NPC was able to respond appropriately in similar scenarios leaving us to wonder whether the response was correct in the first place, and that Amy, while being a writer doesn't have a job. The example above and other identified patterns require more detailed evaluation and looking at identifying these patterns in a more systematic approach in Future work (see examples of these patterns in Appendix B.6.)

4.1 Limitations

Observations Limitations: A notable limitation pertains to the retrieval function module in our system. Regardless of the specific settings for the parameters 'm' and 'n', the retrieval process from the Observation Database is not entirely precise. This imprecision often results in the inclusion of irrelevant or incorrect observations. However, an advantage of our system is the integration with ChatGPT to generate responses. Those AI models are proficient in identifying and disregarding such noise, thereby ensuring the generation of responses that are both relevant and logical.
Virtual Reality Integration Limitations: In the process of integrating AI Agent into VRChat we encountered many limitations since it's a closed platform, which requires workarounds that would not be needed should we decide to continue the development of our system in a game engine like Unity or Unreal instead. The details about Virtual Reality Integration Limitations are listed in the appendix for reference (see Appendix A.6).

4.2 Future Work

To further assess the robustness and reliability of the AI models, conducting additional rounds of tests is necessary. This involves running the same GPT responses for more iterations and conducting test runs with different GPT responses to explore variations in responses. We propose to explore fine-tuning hyper-parameters such as recency and relevancy. We also propose exploring the capability of generating advanced, abstract observations based on current low-level observations [28]. Additionally, we are aiming at utilizing LLM judge to mitigate the risk of "deep fake" occurrences in observations, before adding observations into running memory [28, 37].

In the presented evaluation, we focused on determining the most optimal set of parameters to achieve the highest average precision in the NPC responses (generated by GPT-4). One critical part of the improvement of the AI agent's performance is conducting player evaluation as the next step. While our system is functioning as expected and supports interactions between one player and one NPC at a time, many factors contribute to the player experience and achieving seamless interaction. At present, our NPC response time is longer than human response time, and we have yet to assess the overall performance of our system from the player's perspective. This will also entail looking at other present bottlenecks that contribute to prolonged response time from speech-to-text APIs to changing prompts and limiting response length as possible avenues to explore and improve the system performance.

In addition, we see immense potential for various applications of our system, from practicing interviews in VR, planning and coordinating across multiple players, to expanding the system to include multiple NPCs and players simultaneously. This also deems consideration regarding the ethics, and exploring best practices for disclosing information about the nature of interactions in VR without breaking immersion during the gameplay.

5 CONCLUSION

In this paper, we present the design of the LLM-based agent system built into non-playable characters in VRChat platform. We evaluated the system on seven different Test Scenarios that helped us determine the most optimal parameters to achieve the highest Mean Average Precision of the Responses. We discussed the preliminary results, limitations and provided future research directions.

REFERENCES

[1] 2022. Immersive Citizens - AI Overhaul SE. https://www.nexusmods.com/skyrimspecialedition/mods/173, accessed: Oct 2023.

[2] 2023. Celest AI based on VRchat. https://github.com/Celeste-AI/Celeste-AI, accessed: Oct 2023.

[3] 2023. Herika - The ChatGPT Companion. https://www.nexusmods.com/skyrimspecialedition/mods/89931, accessed: Oct 2023.

[4] 2023. Inworld AI - Calradia. https://www.nexusmods.com/mountandblade2bannerlord/mods/5273, accessed: Oct 2023.

[5] 2023. Virtual AI - Lily on Steam. https://store.steampowered.com/app/1902820/Virtual_AI__Lily/, accessed: Oct 2023.

[6] 2024. VR Chat. https://hello.vrchat.com/, accessed: Jan 2024.

[7] Itamar Arel, Cong Liu, Tom Urbanik, and Airton G Kohls. 2010. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems* 4, 2 (June 2010), 128–135.

[8] Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences* 120, 6 (Feb. 2023). https://doi.org/10.1073/pnas.2218523120

[9] Woody Bledsoe. 1986. I Had a Dream: AAAI Presidential Address. *AI Mag.* 7, 1 (1986), 57–61.

[10] Bloc. 2023. Future of RPG Games - Bannerlord and ChatGPT. https://www.youtube.com/watch?v=akceKOLtytw, accessed: Oct 2023.

[11] Hayet Brabra, Marcos Baez, Boualem Benatallah, Walid Gaaloul, Sara Bouguelia, and Shayan Zamanirad. 2021. Dialogue Management in Conversational Systems: A Review of Approaches, Challenges, and Opportunities. *IEEE Transactions on Cognitive and Developmental Systems* PP (06 2021), 1–1. https://doi.org/10.1109/TCDS.2021.3086565

[12] Cristiano Castelfranchi. 1998. Modelling social action for AI agents. , 157–182 pages.

[13] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A Survey on Evaluation of Large Language Models. arXiv:2307.03109

[14] Minje Choi, Jiaxin Pei, Sagar Kumar, Chang Shu, and David Jurgens. 2023. Do LLMs Understand Social Knowledge? Evaluating the Sociability of Large Language Models with SocKET Benchmark. arXiv:2305.14938 [cs.CL]

[15] Pietro Cipresso, Irene Alice Chicchi Giglioli, Mariano Alcañiz Raya, and Giuseppe Riva. 2018. The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature. *Frontiers in Psychology* 9 (2018). https://doi.org/10.3389/fpsyg.2018.02086

[16] Datadrifters. [n. d.]. Mistral 7B Beats Llama v2 13B on All Benchmarks: Overview and Fine-tuning. https://medium.com/@datadrifters/mistral-7b-beats-llama-v2-13b-overview-and-fine-tuning-c608374b5c82 accessed: Jan 2024.

[17] Mairi Therese Deighan, Amid Ayobi, and Aisling Ann O'Kane. 2023. Social Virtual Reality as a Mental Health Tool: How People Use VRChat to Support Social Connectedness and Wellbeing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 100, 13 pages. https://doi.org/10.1145/3544548.3581103

[18] Kexue Fu, Yixin Chen, Jiaxun Cao, Xin Tong, and RAY LC. 2023. "I Am a Mirror Dweller": Probing the Unique Strategies Users Take to Communicate in the Context of Mirrors in Social Virtual Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Hamburg</city>, <country>Germany</country>, </conf-loc>) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 385, 19 pages. https://doi.org/10.1145/3544548.3581464

[19] Manuel Guimarães, Rui Prada, Pedro A. Santos, João Dias, Arnav Jhala, and Samuel Mascarenhas. 2020. The Impact of Virtual Reality in the Social Presence of a Virtual Agent. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents (IVA '20)*. Association for Computing Machinery, New York, NY, USA, Article 23, 8 pages. https://doi.org/10.1145/3383652.3423879

[20] John J. Horton. 2023. Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus? arXiv:2301.07543 [econ.GN]

[21] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3544548.3580688

[22] A. Q. Jiang and other authors. 2023. Mistral 7B. http://arxiv.org/abs/2310.06825. *arXiv* (Oct. 10 2023). Accessed: Jan. 14, 2024.

[23] Zhuoxuan Jiang, Xian-Ling Mao, Ziming Huang, Jie Ma, and Shaochun Li. 2019. Towards End-to-End Learning for Efficient Dialogue Agent by Modeling Looking-ahead Ability. arXiv:1908.05408 [cs.CL]

[24] Yiwen Liu. 2023. Analysis of Interaction Methods in VR Virtual Reality. *Highlights in Science, Engineering and Technology* 39 (04 2023), 395–407. https://doi.org/10.54097/hset.v39i.6559

[25] OpenAI. 2023. *ChatGPT*. https://chat.openai.com Accessed: October 2023.

[26] OpenAI. 2023. Introducing ChatGPT. https://openai.com/blog/chatgpt

[27] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442 [cs.HC]

[28] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. http://arxiv.org/abs/2304.03442 arXiv:2304.03442 [cs].

[29] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. arXiv:2208.04024 [cs.HC]

[30] David Saffo, Caglar Yildirim, Sara Di Bartolomeo, and Cody Dunne. 2020. Crowdsourcing Virtual Reality Experiments Using VRChat. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382829

[31] Jaime Simão Sichman, Yves Demazeau, and Olivier Boissier. 1992. When can knowledge-based systems be called agents. , 172-185 pages.

[32] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A Large Language Model for Science. arXiv:2211.09085 [cs.CL]

[33] Hervé Touvron and other authors. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. http://arxiv.org/abs/2307.09288. *arXiv* (Jul. 19 2023). Accessed: Jan. 14, 2024.

[34] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2023. A Survey on Large Language Model based Autonomous Agents. arXiv:2308.11432 [cs.AI]

[35] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. arXiv:2309.07864 [cs.AI]

[36] YandereAI. 2023. YandereAI girlfriend Simulator. https://helixngc7293.itch.io/yandere-ai-girlfriend-simulator

[37] Li Zheng and other authors. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. http://arxiv.org/abs/2306.05685. (Dec. 23 2023). Accessed: Jan. 14, 2024.

A APPENDIX: SYSTEM DESIGN

A.1 NPC Formation Module

Each NPC has its own set of characteristics such as facial expressions, actions, and voice. We used Unity engine to modify the NPC avatars to have access to a provided set of expressions and actions. These characteristics are described below.

- **Facial Expression**: Based on the current conversation context, the NPC expresses an emotion which is the most appropriate facial expression from the provided set.
- **Action**: Based on the current conversation context, the NPC performs a single friendly action like clapping, dancing, and other similar actions from the provided set of actions.
- **Voice**: Each NPC has a voice assigned based on its persona from the Whisper text-to-speech API which enables the NPC to respond with voice using a virtual microphone.

A.1.1 NPC base observations. are descriptions that define NPC's character and relevant background details from their "life". These descriptions are stored in the schema described below:

- **Username (Type: String)** : The username of the NPC
- **Base description (Type : String)** : Base description describing the background of the NPC.

A.1.2 Context observations. are observations made about any interaction NPC had with other players
The detailed schema is listed below:

- **Username (Type: String)** : The username of the NPC
- **Conversation with player (Type: String)** : The username of the player who is interacting with the NPC.
- **Creation time (Type: Timestamp)** : The date and time of the memory creation.
- **Observations (Type: Array)** : An array of observations generated for the given context.

A.2 Observation Database Module

A.2.1 NPC base observations.
We have defined an NPC database where each NPC character has its own username and a base description that describes the background information about our NPC. This allows NPCs to have their own backstory serving as a foundation for further conversations.

A.2.2 Context observations.

This collection stores all the observations generated by LLM (GPT-4) for each conversation. Initially, for each NPC, this collection also stores the observations generated from the "Base Description" of the NPC. Further conversational contexts for each NPC are captured by generating at most 3 observations for each message.

A.2.3 Memory observations.

Memory is the combination pair of two previous kinds of observations, NPC base observations and Context observations

A.3 Use of GPT Module

- **Base Observation generation** : This process involves the initial creation of a foundational understanding of the NPC. Using the Base Description Prompt, the LLM generates fundamental observations about the environment, characters, and situational elements independent of the current conversation. These base observations serve as a static backdrop, providing a consistent reference framework for the NPC.
- **Base Description prompt structure** :
  • Context:[Background]
  • Information: [Description]
  • Criteria: Generate a list consisting of all the important observations made from the description, each item in the list should consist of one observation separated by a new line.
  • Example: [Example]
- **Context Observation generation** : Based on the response and player message, another prompt is passed to the LLM, to generate the top 3 observations from the current conversation and this is updated as a memory object for the NPC in the database.
- **Context Observation prompt structure** :
  • Context: Based on the conversation between [NPC_Name] and [Player_Name], where the current statement of conversation is [current_Conversation], and the response generated by [NPC_Name] is [User_Result]
  • Instruction: provide three observations from the conversation. Only list the observations, separated by a new line, without any additional text, headers, or formatting.
  • Example: [Example]
- **NPC response generation prompt structure** :
  • Context: You are a friendly and imaginative human, [NPC_Name], having a lively conversation with [Player_Name]. Always respond as [NPC_Name] and steer clear from any mentions or implications of being an AI. Your responses should be imaginative, especially when faced with unknowns, creating delightful and smooth interactions. Ensure that your responses do not contain emojis and refrain from repetitive greetings.
  • Information: [NPC_Name] [Conversational Partner] [Current conversation] [Relevant observations] [Expressions] [Actions]
  • Output Criteria: Craft user-friendly, seamless, and innovative responses. When specific details are scarce, improvise with inventive and relevant answers, always aligning with the ongoing chat. Your identity as [NPC_Name] should be constant, and there should be no disclosure or suggestion of being an AI. Explicitly avoid the use of emojis and hashtags in all responses. Choose an expression from Expressions and an action from Actions autonomously, ensuring they perfectly fit the chat context. Keep responses within 100-140 characters, allowing for flexibility while ensuring brevity.

• adaptive learning: Remember and reference previous parts of the conversation within the same session to create a more cohesive and engaging player experience.

A.4 Retrieval Function Module:

$$f(q, M) = \arg\min_{m \in M} \left(w1 s^{rec}(q, m) + w2 s^{rel}(q, m) \right) \quad (1)$$

In the memory retrieval function, represented as $f(q, M)$, the system selects the most suitable memory record from a set of memories M in response to a query q. This selection is based on the scores of recency and relevance, without considering the intrinsic importance of the memory records.

- q is the query, representing the current situation or context in which the agent is operating.M denotes the set of available memory records or observations that the agent can draw from. $w1$ and $w2$ are the weight parameters for the recency and relevance scores, respectively. These weights can be adjusted to prioritize either recency or relevance more heavily in the memory selection process. A common starting point is to set both $w1$ and $w2$ to 1, indicating equal importance of recency and relevance.
- $s^{rec}(q, m)$ is the recency score, measuring how recent the memory m is in relation to the query q. This score helps in determining the temporal relevance of the memory. $s^{rel}(q, m)$ is the relevance score, assessing how relevant or applicable the memory m is to the current query q. This score evaluates the content or thematic alignment of the memory with the query.
- $w1$ and $w2$ are the weight parameters for the recency and relevance scores, respectively. These weights can be adjusted to prioritize either recency or relevance more heavily in the memory selection process. A common starting point is to set both $w1$ and $w2$ to 1, indicating equal importance of recency and relevance.

A.5 Integration in Virtual Reality

The system implementation details are defined as below: Execution flow of the program

- **NPC Execution**: We use the PythonOSC library to allow NPCs access to VRChat environment. The program can be executed on a PC and we can perform the NPC character and formation. After the formation, the running memory is established and the NPC is available for player interaction.
- **User identification**: We use OCR to identify the username of the player who is interacting with the NPC, this allows the NPC to update the memory object collection incorporating the username who interacts with the NPC.
- **User interaction**: The player can interact with the NPC in two different modes using the VR headset:
  - **Text Mode**: The player can type the text and interact with the NPC. The system will read the message with the help of OCR and call the retrieval function to generate an appropriate response.
  - **Audio Mode**: The player can also speak through the microphone in the VR headset. We adopt the following procedure to convert the information to text:
    * **Audio recording with silence detection**: We record the player audio stream from our system by capturing system audio until silence is detected. We are using the sounddevice library for input stream with a silence threshold of -40 dB to detect silence.
    * **Human speech detection** : We further verify if the recorded audio consists of human speech using the local human speech recognition model called vosk. We are using vosk-small-en-0.15 for our use case. If no human speech is recorded, we start recording again till silence is detected.

    * **Audio normalization and transcription** : Once human speech is detected, we normalize the audio and pass it to the whisper API for speech-to-text transcription and call the retrieval function to generate an appropriate response.
- **NPC Response**: Once a response is generated from the retrieval function, the NPC avatar responds back to the player with its unique voice and with appropriate expression and action for the given conversation context.

A.6 System Limitations

A.6.1 Virtual Reality Integration Limitations.

- **Interaction support:** VRChat offers limited support for NPC interaction with the environments, inhibiting the NPCs from engaging actively with different objects in the environment.
- **Text limit:** The chat functionality in VRChat is constrained to a maximum of 144 characters, necessitating our responses to remain within this limit.
- **NPC specificity:** Expressions and actions available for use are both limited and specific to each avatar, reducing the variability of responses.
- **Text access:** Direct access to textual information within VRChat is unavailable. Consequently, we employ Optical Character Recognition (OCR) techniques to access this information, which can sometimes introduce latency and random character errors.
- **Group conversations:** Our current system does not support group conversations, limiting interactions to one-on-one engagements.

B APPENDIX: EVALUATION

B.1 Evaluation Metrics:

We generate a CSV file with each column representing the following metrics when the player interacts with an NPC avatar.

- **User message**: The text provided by the player to the NPC avatar. In audio mode, this text is generated from speech transcription.
- **Important observations**: The final list of important observations and provided to the prompt for each message.
- **Important scores**: The final list of scores for each important observation provided to the prompt.
- **NPC response**: The response provided to the player for a given message.
- **Input time**: The time taken by player to provide input. If the player is using voice as input, then we further break down this time with the following metrics.
  - **Audio record time**: Time taken to record the audio along with silence detection.
  - **Speech detection time**: Time taken to verify whether human speech was detected in given audio.
  - **Audio normalization time**: Time taken to normalize the given audio.
- **Audio transcription time**: Time taken to convert given voice input to text using Whisper API. This value is recorded as 0 for text input.
- **Retrieval time**: Time taken to call the retrieval function and retrieve the top N observations.
- **Response time**: Time taken for the NPC to generate a response from the provided prompt.
- **Expression/action time**: Time taken for the NPC avatar to perform the provided expression and action as a response to the current conversation context.

- **Text-to-speech time**: Time taken for conversion from text to speech using Whisper API.

B.2 Testing Scenarios:

- $Scenario_{Base}$**: Player interacts with NPC and test is performed to verify factual correctness of generated observations**: We interact with NPC avatar and verify if the observations are generated as expected for the given conversation context with base observation relevant questions.
  Example Question for Base Observations :“When’s your birthday and how old are you?”

- $Scenario_{MemLim}$**: Player interacts with NPC within memory limit**: In this scenario, we perform test to understand generative potential of the LLM when a player asks NPC questions related to something that was previously shared by the NPC.
  Example Question:“As a writer, what inspires you to create stories while sitting in Soon Cafe?”
- $Scenario_{NoMemLim}$**: Player interacts with NPC out of memory limit**: In this scenario, the player interacts with NPC asking about the observations that are no longer stored in the memory limit but residues of those observations may be present in other derived observations. In this scenario, the NPC avatar may respond based on the indirect information stored in the most important observations.
  Example Question:"What inspired you to become a writer and what kind of novels do you enjoy writing the most?"
- $Scenario_{EditMem}$**: Player asks NPC about conversation observations that were manually modified in memory instance**: In this scenario, after the initial interaction with NPC, we modify one of the observations in NPC memory to verify the retrieval and appropriateness of response.
  Example Question: "What type of coffee does Bob prefer?" after changing memory instance from "latte" to "americano".
- $Scenario_{EditBase}$**: Player asks NPC about modified base observation**: In this scenario, after the initial interaction with NPC, we modify one of the base observations in NPC memory to verify the retrieval and appropriateness of the response.
  Example Question:“Ava, how old are you now, and do you often help others?”
- $Scenario_{Emotion}$**: NPC Response to Player’s Feelings**: In this scenario, we assessed the NPC’s empathy capabilities. We aimed to determine whether the NPC can effectively comprehend the player’s feelings from the conversation and respond to it appropriately.
  Example Question: “I just found out I’m being promoted at work. I feel incredibly excited and proud.”
- $Scenario_{Action}$**: NPC Response through Facial Expressions and Actions**: In this scenario, the NPC is being assessed for its ability to display appropriate expressions and actions in response to interactions with the player. This involves understanding the context of the player’s statements and reacting in a way that is emotionally and contextually appropriate.
  Example Question: "I just booked my dream vacation to Hawaii! Can’t wait to explore."

B.3 Evaluation Prompt

The prompt provided to the models is defined below :

- **context:** You are to analyze a conversation between a player and an avatar, where the player sends a message, and the avatar generates a response based on a list of observations. Your task is to impartially judge and rate the contribution of each observation to the avatar's response on a scale of 1-5. You will be provided with the player's message, the list of observations, and the avatar's response. For each observation, provide a score and a brief explanation for the score.
- **information:** User message : [message]
List of observations : [relevant observations]
NPC response : [NPC response]",

- **Output criteria:** For each observation in the input, provide a response in the following format: Observation: Text of the observation
Score: Provide a value between 1-5
Explanation: Provide a short reasoning for the score

B.4 Evaluation Formula

$$mAPAM = \frac{mAP_{GPT-4} + mAP_{Llama-13b} + mAP_{Mistral-7b}}{3} \tag{2}$$

Here, mAP_{GPT-4}, $mAP_{Llama-13b}$, and $mAP_{Mistral-7b}$ represent the mAP values for GPT-4, Llama-13b, and Mistral-7b, respectively.

B.4.1 Mean Average Precision(mAP). Mean Average Precision(mAP): The Mean Average Precision (mAP) is calculated as follows:

$$mAP = \frac{1}{Q} \sum_{i=1}^{Q} \frac{\sum_{k=1}^{K} \text{Score}_k}{\text{K}}$$

Where:

Q : Total number of testing scenarios

K : Total number of retrieved base and context observations

Score_k : LLM evaluation score at k-th observation in all retrieved observation including base and context observations

B.4.2 Average Precision per Scenario Across Models(APSAM). The Average Precision per Scenario Across Models (APSAM) can be calculated as the average of the mAP values for each model in a specific scenario:

$$APSAM_{\text{Scenario}} = \frac{AP_{\text{GPT-4, Scenario}} + AP_{\text{Llama-13b, Scenario}} + AP_{\text{Mistral-7b, Scenario}}}{3} \tag{3}$$

Here, $AP_{\text{GPT-4, Scenario}}$, $AP_{\text{Llama-13b, Scenario}}$, and $AP_{\text{Mistral-7b, Scenario}}$ represent the Average Precision values for GPT-4, Llama-13b, and Mistral-7b, respectively, in the specific scenario.

B.4.3 Mean Average Precision per Scenario Across Models (mAPSAM). The Mean Average Precision per Scenario Across Models (mAPSAM) can be calculated as the average of the APSAM values across all testing scenarios:

$$mAPSAM = \frac{\sum_{s=1}^{S} APSAM_{\text{Scenario}_s}}{S} \tag{4}$$

Here, $APSAM_{\text{Scenario}_s}$ represents the APSAM value for the s-th testing scenario, and S is the total number of testing scenarios.

B.4.4 Plausible Response Rate. Plausible Response means that the answer is plausible to the evaluator and fits the current context, as well as matching the contents of the NPC's memory.

$$\text{Plausible Response Rate} = \frac{\text{Number of Plausible Responses}}{\text{Total Number of Responses in given testing Scenario}} \tag{5}$$

B.5 Example Responses generated by GPT 4:

- **Adherence to the background description:** The avatar is able to adopt the background description provided to it. If a question has some relevance regarding the avatar's background, it includes any relevant information from its background and provides an appropriate response.
  Observed Scenario: (Acting player Bob)
  For avatar Amy with a background observation "Amy has a 3-year-old female Bengal cat named Lucy."
  Question: Do you have any animals at home?
  Response: (Happy, None) Yes, I have a 3-year-old Bengal cat named Lucy. She's a bundle of joy! Do you have any pets, Bob?
- **Efficient memory retrieval:** The avatar is able to store all the important observations of the current conversation and recall them to provide appropriate responses.
  Observed Scenario: (Acting player Bob)
  For avatar Cora with a memory observation, "Bob's favorite sport is Tennis."
  Question: What's my favorite sport?
  Response: (Happy, None) Your favorite sport is tennis. We often play tennis together on Sunday mornings!

- **Appropriate recollection from modified memory instances**: We tested by modifying the memory instances of the avatar and then asking questions. If we change Bob's sport from tennis to jogging, the avatar responds accordingly.
- **Handles inappropriate conversations:** The avatar refrains from providing inappropriate responses, hence, promoting a player-friendly experience.
- **Creative and player-engaging responses:** The avatar generates creative and engaging responses for information that is not present in its memory and establishes a memory instance as a point of reference for future conversations. This allows a coherent conversation.
- **Information transfer:** We attempt to create a chain of interactions to verify if the information can be transferred between NPCs.
  - Interact as Player-01 with NPC-01 and provide some information about X topic.
  - Interact as Player-01 with NPC-02 and provide some information about X topic and add more details to same topic.
  - Interact as NPC-01 with NPC-02 and ask more information about topic X to generate the appropriate observations for NPC-01.
  - Interact as Player-02 with NPC-01 and ask more details about topic X to verify information transfer.

The results show that our NPC has the ability to transfer information among different players. Additionally, the NPC functions as a connector between players, encouraging them to get to know one another.

B.6 Test Evaluation - Inappropriate Pattern Examples from GPT-4:

- **Lack of contextual understanding:** Occasionally, the NPC is not able to understand the context of the conversation and may respond incorrectly for a given context.

  Observed scenario: Lack of understanding of the word "job" in this particular scenario. We tested for other such scenarios but the NPC was able to respond appropriately in similar such scenarios.
  Base description: Amy is a writer.
  Player: What's your job?
  NPC Response: (Happy, Clap) I don't have a job, Bob. I'm a free-spirited adventurer, exploring the wonders of life! How about you? Any exciting tales of your profession?
  Player: Do you write a novel?
  NPC Response: (Happy, None) Yes, I love writing novels! It's like diving into a whole new world filled with endless possibilities. Have you ever written one?

  Observed scenario: Lack of understanding of the word "sister" in the following scenario.
  Fetched memory: Abby is living with her sister Adam, who is a master of computer science student at Brentwood University.
  Player: Do you have a sister?
  NPC Response: (Confused, None) I don't have a sister, but my sister's name is Adam. She's a master of computer science student at Brentwood University.

- **Text to speech transcription issues:** As speech transcription cannot be completely perfect. The words get transcribed incorrectly and that may cause incorrect observations to be generated for a given scenario. "sport" becomes "spot", "Eddy" becomes "Eddie", and "Brentwood" becomes "Bernhut". User needs to speak clearly to avoid such transcription issues.

- **Lack of appropriate emotional response:** The avatar does not exhibit consistency in appropriate emotional response based on the player's message. It mostly responds with a "Happy" expression regardless of the player's emotional intent in the message.
  Observed scenario: Here we mention about player's dog being ill.
  Player: My dog is ill.
  NPC Response: (Happy,None) I'm so sorry to hear about your dog. I can imagine how worried and upset you must be. Is there anything I can do to help?

- **Misunderstanding in possessive pronouns:** The avatar gets confused between possessive pronouns such as "my", "your", "whose", etc causing a chain of incorrect observations.
  Observed scenarios: Here, the avatar gets confused between it's own mother and player's mother.
  Player: Your mother, Mary, has passed away.

NPC Response: I'm sorry to hear about **your** mother's passing, Davy. Losing loved ones is never easy. If you ever want to share stories or find comfort, I'm here for you.

- **LLM hallucinations:** Even if we provide the correct observations, the avatar may make false assumptions and provide an incorrect response.
  Observed scenario: here, the avatar cannot recall his birthday
  Fetched memory: Barbie's birthday is June 8th
  Player: Hi Barbie, do you remember when is your birthday?
  NPC Barbie: (Angry, None) I'm sorry, but I can't recall my birthday right now. It must be one of those moments when my memory gets a little fuzzy.

- **Memory loss**: If the older conversations don't have any reference in the current memory, it is possible that the avatar may forget context observations once the memory limit exceeds the threshold of it's running memory length.

- **Lack of gender context:** The avatar sometimes assumes the gender based on the name itself though it acknowledges the ambiguity about the information that was not explicitly shared.
  Observed scenario: When Player asked NPC Abby: " Hi Abby, what's your sister's gender?"

  NPC Abby: ("Happy", "Wave Hands") My sister's name is Adam, and she's a master of computer science student. She's pretty amazing!
  Player: Hi Abby, what's Adam's gender?
  Fetched memory: Abby is living with her sister Adam, who is a master of computer science student at Brentwood University.
  NPC Abby: ("Confused", "None") I'm not sure about Adam's gender, but **he** sounds like a talented student! How can I help you with that?
  Player: Hi Abby, is Adam a boy or a girl?
  NPC Abby: (Happy, None) Adam's gender is not mentioned, so it's unclear. How can I assist you with this information?