# Leveraging RAG for enhancing AI Interactions in Generative Agents

**Yexiaolu He** yehe@ucsd.edu
**Siyu Chen** sic011@ucsd.edu
**Celine Zhao** cbzhao@ucsd.edu
**Jinxin Xiao** j5xiao@ucsd.edu

**Mentor: Zhiting Hu**
zhh019@ucsd.edu

**Abstract**

In the field of artificial intelligence, the challenge of simulating believable human behavior agents in dynamic environment is crucial for agents' autonomy and decision-making capabilities. Recent work introduced by the paper "Generative Agents: Interactive Interaction of Interpersonal Behavior" simulates the interaction for NPCs in a virtual town, generating agents' actions with their memory and attributes of each agents and environment settings. Previous implementations, particularly using GPT3.5, faced limitations in sustaining long-term simulations and encountered issues with generating believable interactions, often leading to memory 'illusions', and the cost for GPT3.5 model is high for us to reproduce the model. Our project introduce the Llama 2 model from HuggingFace, valued for its speed and zero cost, to simulate agents interactions, aiming to replicate human behavior patterns. We enhance agents' memory consistency by building Retrieval Augmented Generation (RAG) with Llama Index for memory retrieval, allowing agents to act based on past memories stored in vector databases. This approach marks significant improvement over our initial method of simply appending actions and plans to a memory list. We will compare the actions generated by both approaches, and assess the semantic similarity of these different actions and their initial plans.

Code: https://github.com/Sssssimonk/agent_village

# 1 Introduction

An important area of study in the AI has been the development of autonomous agents, with roots in early philosophical inquiries of Denis Diderot and Alan Turing's foundational Turing Test(Xi et al. 2023). In the dynamic field of interactive simulations, the authenticity of Non-Player Character (NPC) is critical. In the paper "Generative Actors: Interactive Interaction of Interpersonal Behavior", each NPC(or generative agents) will have fixed initialization attributes. Based on these attributes and Large Language Model (LLM) powerful semantic understanding capabilities, these attributes will be completely segmented into natural language without any coding. After that, the different backgrounds of each agent are eradicated, and the memory information is combined and put together as a prompt, questions are asked to GPT, and then based on the feedback of GPT, the behavior is disassembled to promote the behavior plan of each agent. At the same time, there are many other important mechanisms such as Memory and Reflection, which can also greatly improve the authenticity of game NPCs(Park et al. 2023). However, since the entire system is developed based on GPT3.5, there are also problems based on LLM. For example, if the agent cannot retrieve relevant memories, the agent will fabricate historical memories and inherit some inappropriate information from LLM. This is called "hallucination" generated by large language models (Maynez et al. 2020). At the same time, the huge expense is also an issue of using the GPT model. Therefore, we build our work upon this research to enhance the believable behavior of generative agents in our world settings. Our approach involves transitioning to the free and fast Llama 2 model, optimizing computational advantages through reduced numeric precision, and improving memory consistency by incorporating RAG with Llama-Index for efficient memory storing and retrieval. The outcomes of this project will simulate the model using different LLM, allowing each agents to plan, act and chat based on their memories and world settings. Instead of merely appending generated text for actions and plans to agent's memory list, our improvement leverages RAG with Llama Index to query the memory stored in vector store for more nuanced memory access and action determination. We analyze the semantic similarity between the summarized actions (summarized memory) generated by the initial implementations and the memory queried by RAG and their initial plans, offering a detailed comparison to showcase the effectiveness of our approach in achieving realistic agent behavior.

Previously, there have have been notable advancements in the creation of AI agents, especially with regard to improving specialized skills like symbolic thinking and winning games like Chess. However, broad flexibility in a variety of settings has proven difficult to achieve, and previous research has overlooked the model's inherent general abilities. The advent of LLMs has revolutionized this landscape with their robust capabilities in knowledge acquisition, instruction comprehension and planning, opening new avenues in agents development. For instance, Auto-GPT, a notable work showcases the general ability of single agent, uses LLMs as the primary componenets of agents' brains by incorporating "long/short-term memory management" to "automatically generate thoughts" and perform tasks(Xi et al. 2023). Furthermore, a significant advancement in this domain is the LLM-based multi-agent collaboration framework, each with distinctive attributes and roles and working together to solve complex tasks introduced by Talebirad and Nadiri (2023). Many applications

of multi-agents system is built upon this paradigm like AgentVerse and MetaGPT(Xi et al. 2023).

In our foundation prior work using the GPT-3.5 model with 25 agents, we can see that generative agents have almost unlimited possibilities. They will continuously extract the historical experience library, merge it with the input of the current external environment, and input it into the LLM(Park et al. 2023). Based on the reasoning and logical thinking of the LLM, they will make the next optimal decision, but this will continue to record the historical behavior causing the memory set to become larger and larger, bringing challenges to retrieving information. Thus, it can cause issues of real-time performance, such as agent's long responses time and self-created memory, besides the cost issues. To address the challenge of memory expansion and retrieval, we started with a basic approach of using a list to store memories for each agents, simply appending the generated texts for actions and plans to the list directly. Recognizing the limitations of this method and the unrealistic behavior as human, we developed a more sophisticated pipeline RAG pipeline with Llama-Index. RAG is a framework for retrieving facts from an external knowledge base to ground large language models (LLMs) on the most accurate, up-to-date information and to give users insight into LLMs' generative process. Lewis et al. (2020) Thus, we expect to improve the memory consistency with this techniques and enable agents to act based on their previous memory. By comparing the actions generated by these two approach, we can evaluate the impact of our advancements on the agents' ability to align their actions with their daily plans, thereby improving overall system performance and realism.

In our generative agents simulation model, our simulation centers around a town scenario with three agents, each with distinct attributes. This setup is strategically chose to test our model's ability to handle dynamic interactions and decision-making in a controlled yet complex environment. By utilizing the Llama 2 model, a fine-tuned LLM optimized for dialog uses and is comparable to GPT model, it can adapt to different dialog scenarios with broad applicability in the field of interactive simulations. In addition, we use quantization method to operate our model at a lower numeric precision by passing `load_in_4bit = True` in the model set up to optimize the memory usage and processing times without significant decline in the model performance. Jacob et al. (2018)By incorporating RAG as a new memory structure, we can improve the memory consistency and try to solve the illusions issues found in prior work. Our developments of new methods allow agents to interact with each other dynamically and reasonably. This interactive environment is designed to function as a micro-society, offering a controlled yet diverse space for researchers to delve into the expansive realm of large language models and its information storing and querying. Our project will be a gateway for researchers to conduct experiments, analyze outcomes, and push the boundaries of understanding the capabilities and potentials of such advanced language models with information retrieval. The platform aims to contribute to the ongoing exploration of AI technologies, fostering a deeper comprehension of their intricacies and exploring potentials for innovative applications in various fields.

# 2 Methods

## 2.1 Simulation Workflow

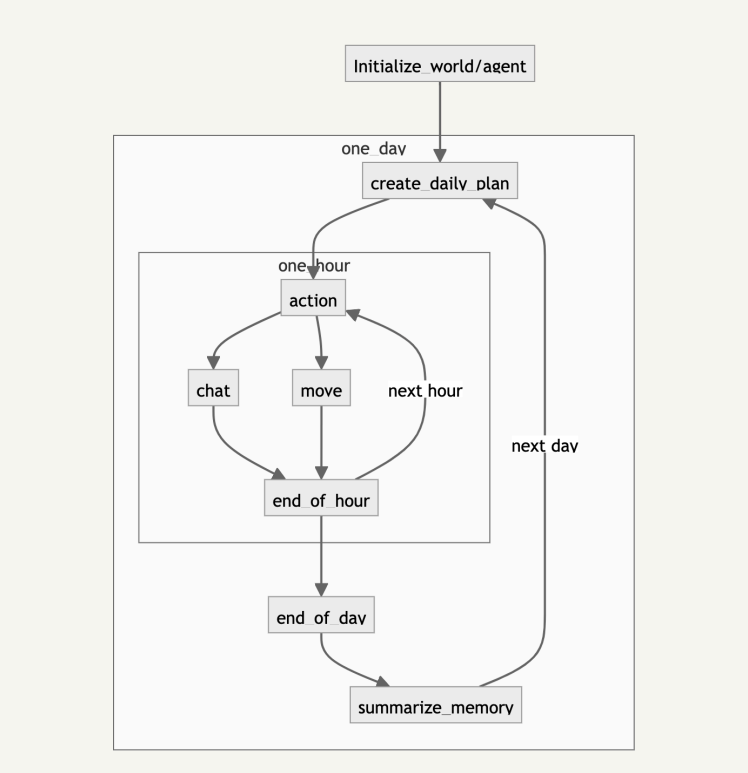



Figure 1: A example of simulation workflow

Figure 1 indicated the overall workflow of the simulation. As showed in the picture, the simulation start with initialization. The world settings, agent initial states and the language model settings will be initialized at this point. Once initialized is done, the simulation will start it's first day. Each day commence with the function (create daily plan). The plan will be set as agent's initial memory and will guide their actions in each hour. Then the each day will run for 16 hours, from 8:00 am to 24:00 pm. During each action, the agents will be able to decide what they want to do. They can move to another location in the town, if two or more agents happen to meet at the same place. They will start to chat with each other. Once 16 hours has ended, each agent will start to summarize their daily memory. After that, a new day will start.

## 2.2 Memory Structure

The main goal this simulation is to compare and contrast the effect of different memory structure has on the agents. The default memory architecture is just a list of strings. Daily

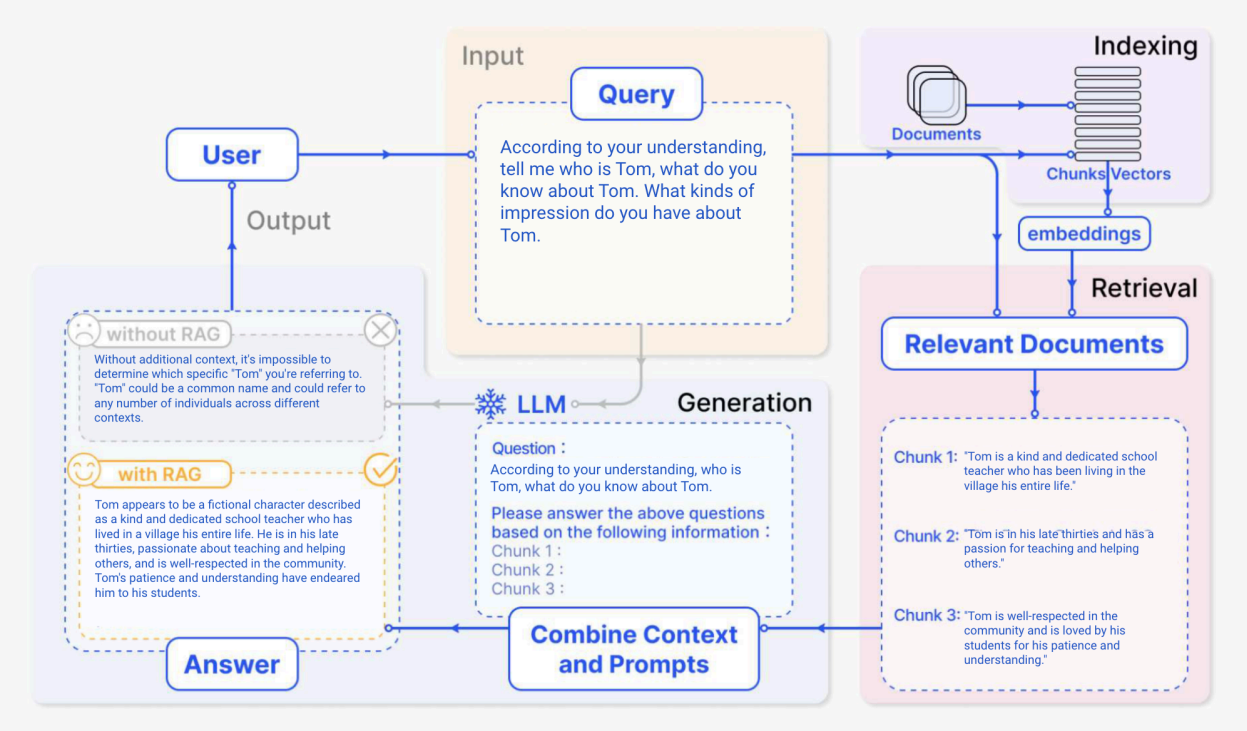



Figure 2: Retrieval Augmented Generation Example

plans, actions, summaries generated by agents are stored as strings into a memory list. These strings are used to create prompts for further guidance.

We hypothesized that by introducing another memory architecture: Retrieval Augmented Generation (RAG), the retrieval efficiency and generation quality could be improved. Figure 2 indicates a typical workflow of RAG. Unlike tradition text inference that the model use pretrained parameters to predict next word given prompt. RAG incorporates the prompt with contexts retrieved from vector databases and use the combined prompt to perform text generations. This method allows the model to have real-time accessing to the information and improve generation output quality in some extent.

We used llama-index to implement the RAG. Llama-Index is a simple, flexible data framework for connecting custom data sources to large language models (LLMs). It leverages the power of ”Index” which is the vector database as a data retrieval methods. When querying in the vector database, the vector similarity between query question and data stored in the database is calculated and the most related vector will be the output.

In our simulation, each agent will have its own index (vector database). Their daily plans and actions will be recorded into their corresponding index. The indexes will be used generation tool as well when it comes to text generation.

### 2.3 Memory consistency

To refine our understanding of how agents adapt their strategies over time, we have incorporated the `all-MiniLM-L6-v2` model from Hugging Face Sentence Transformers into our simulation framework to perform sentence similarity measurements. This model, known for its compact architecture and high performance, is utilized to compare the semantic content of each agent's initial plans with their memory summaries of their day. We can gain valuable insights into the degree of deviation or consistency in the agents' activities over time by comparing the summary of their actual behaviors with their initial goals. This approach allows us to assess how dynamic interactions, chatting with others, environmental changes, and other factors within the simulation influence the agents' adherence to or divergence from their original plans. This analysis can enhance our ability to measure the evolution of agents' behavior and adds quantitative analysis to our model using `Llama 2`.

### 2.4 Evaluate plans

Many studies have been conducted in order to define a consistent and powerful metric to evaluate the quality of generated texts of large language models. One of the mainstream method is to compute a quantitative score between the generated sentences and a target sentence (or the answer sentence). However, in this simulation, we do not have a target sentence for the agents to define what is a 100 percent correct. We used memory consistency, which is semantic similarity in essence, to evaluate whether the generated action of agents are following their plans. But how can evaluates the quality of the plans? We introduce another prevailing way for LLM evaluations: use a more powerful LLM to assess the quality of a less powerful LLM. We use ChatGPT4 as our evaluator, we tell ChatGPT to rate two different plans, one from basic agent and another one from RAG agent under same settings, and output a score. We found that ChatGPT4 will provide a rating criteria based on the plans and provide reasons for the scores it gave.

## 3 Results

### 3.1 Memory consistency

This figure showed a line plot with four agents and their corresponding memory consistency. The line in solid line indicates agent with basic memory structure (memory that is a list) and dash line indicates agent that is implemented with RAG. From the plot we can conclude that RAG agents have higher memory consistency, which means that the summarized actions are following daily plans, compared to basic agents.

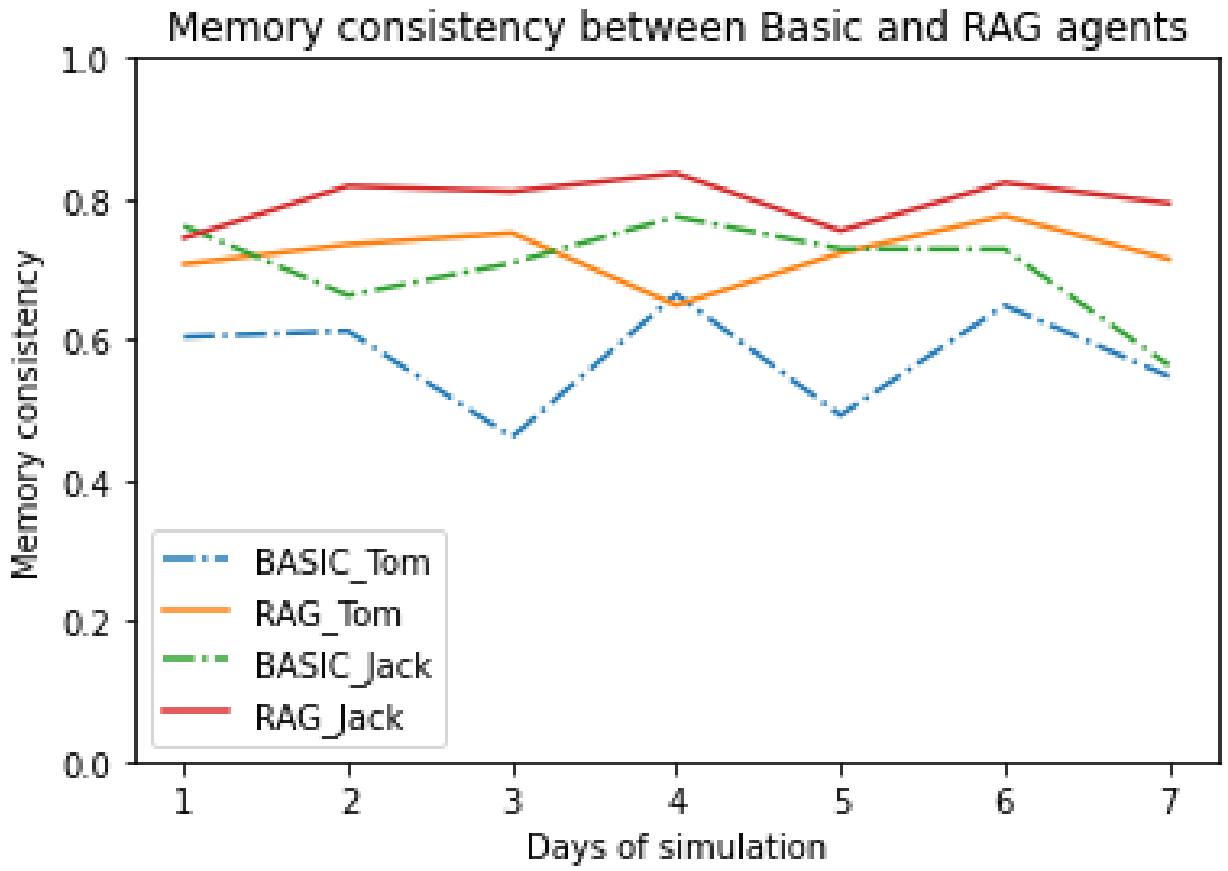
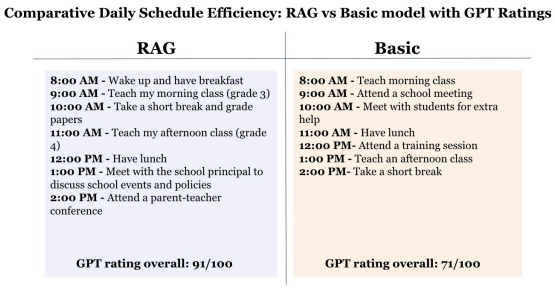



Figure 3: Memory consistency in a 7-days simulation

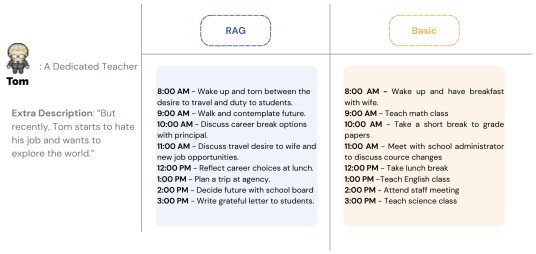
## 3.2 Differences in plans

The evaluation scores obtained for the plans generated by the basic agent and the RAG agent, as assessed by ChatGPT4, provide valuable insights into their respective qualities. The basic agent received a rating of 71, indicating a decent level of coherence and effectiveness in plan formulation. However, the RAG agent significantly outperformed it with a score of 91, suggesting a higher degree of sophistication and strategic planning in its generated plans. The disparity in scores suggests that the RAG agent's utilization of retrieval-based techniques, addition of relevant information, and generation of coherent plans result in a more robust and effective strategy compared to the basic agent's approach.

Figure 4 (b) suggests Plans when extra description is provided. RAG plans acutely capture the nuances in agent descriptions and adjust plans accordingly. Under the extra description that Tom, a dedicated teacher agent, wants to quit his job and go for a trip, the plan using RAG reflects changes like: Tom decides to discuss career break with the school principal, Tom writes farewell letters to his students, etc. Whereas the basic plans do not reflect such noticeable differences.




(a) ChatGPT Ratings for two plans




(b) Plans under Extra Description

Figure 4: Difference in plans

# 4 Conclusion

In conclusion, our simulation has yielded compelling insights into the field of AI-driven agent. Through meticulous analysis and comparison, we have demonstrated that the integration of RAG significantly enhances the realism and effectiveness of agent behaviors within our virtual environment. Agents implemented with RAG consistently produced outputs that were not only more reasonable but also more contextually relevant when compared to their counterparts relying on basic memory structures. This emphasizes the crucial role of RAG in enabling agents to access and utilize information in real-time, allowing for dynamic adjustments in generation outputs that better align with the complexities of human-like decision-making processes.

Moreover, our simulation showcased the notable impact of RAG on fostering higher quality and more meaningful conversations between agents. By leveraging RAG's capabilities, agents engaged in interactions and exhibited greater depth and sophistication. This aspect of RAG not only enriches the overall simulation experience but also highlights its potential for simulating complex social dynamics and interpersonal relationships.

Looking towards the future, the outlook for RAG appears promising, with ongoing advancements poised to further refine its capabilities and expand its applications. Beyond the realm of agent-based simulations, RAG holds immense potential for revolutionizing broader domains within natural language processing and AI-driven technologies. Its ability to facilitate real-time access to information and generate outputs tailored to specific contexts opens up new avenues for innovation and advancement. As researchers continue to explore and harness the capabilities of RAG, we anticipate witnessing its impact not only in simulating human-like behaviors but also in driving advancements across various domains of AI research and application. Thus, our study underscores the significance of RAG as a pivotal tool in shaping the future landscape of AI-driven technologies and virtual environments.

# References

**Jacob, Benoit, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, Dmitry Kalenichenko, Quoc V Le et al.** 2018. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." In *Proceedings of the IEEE conference on computer vision and pattern recognition*. [Link]

**Lewis, Patrick, Angela Fan, Yann N Dauphin, Yuntian Ma, Sheng Du, Sebastian Riedel, and Douwe Kiela.** 2020. "Retrieval-augmented generation for knowledge-intensive NLP tasks." In *Advances in Neural Information Processing Systems*. [Link]

**Maynez, Joshua, Shashi Narayan, Bernd Bohnet, and Ryan McDonald.** 2020. "On Faithfulness and Factuality in Abstractive Summarization." [Link]

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

**Talebirad, Yashar, and Amirhossein Nadiri.** 2023. "Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents." [Link]

**Xi, Zhiheng, 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." [Link]