Application and Analysis of Large Language Models in Game NPCs

Ziqi Wang *

Department of Electronic Information Science and Technology, NorthWest University, Xi'an, China

* Corresponding Author Email: 2023117501@stumail.nwu.edu.cn

Abstract. In recent years, the breakthroughs of large language models (LLMs) in capabilities such as natural language understanding and multi-turn dialogue generation have not only promoted the technical implementation in fields like natural language processing (NLP) and computer vision (CV), but also provided core support for the integrated exploration of "LLMs + intelligent agents". However, most tests rely on simple and static virtual scenarios, which are difficult to simulate the complex environments with "multi-variable interaction and dynamic target changes" in real situations. Games, however, can exactly fill this gap: they can not only construct interactive scenarios requiring advanced cognitive abilities (e.g., task decision-making in open worlds, emotional dialogue of non-player characters (NPCs)), but also achieve low-cost data acquisition and controllable variables through procedural generation. Therefore, games have become an ideal carrier for evaluating the comprehensive capabilities of intelligent agents. With the steady growth of the game market share in recent years, various game manufacturers have introduced LLMs into in-game NPCs and achieved positive results. Research on the application of LLMs in game NPCs is of great significance for the intelligent advancement of NPCs. This review mainly summarizes and analyzes the model frameworks of LLM applications in NPCs, which can provide assistance for the construction of more reasonable and in-depth frameworks. It is conducive to the cross-optimization of models and provides reference significance for model development. In the future, the research applied to game NPCs is also expected to be extended to other fields.

Keywords: Large Model; Non-player characters; Framework; Game Agent.

1. Introduction

Throughout the evolution of the gaming industry, Non-Player Characters (NPCs) have always been a critical factor influencing the immersive experience. Traditional game NPCs are primarily driven by pre-set scripts and finite state machines. Although they can fulfill basic interactive functions, their behaviors and dialogues are rigid, making it difficult to meet players' growing demands for personalization and diversity, which severely limits the vividness and realism of games [1, 2]. The emergence of Large Language Models (LLMs) has brought a breakthrough solution to this dilemma. In many cases, compared with agents trained through traditional methods, Language Model-based Agents (LMAs) demonstrate stronger generalization capabilities [3]. As a cutting-edge technology in the field of artificial intelligence, LLMs, with their powerful natural language understanding and generation capabilities, are expected to completely revolutionize the design paradigm of NPCs and endow them with human-like intelligent interactive features [4].

From a technical perspective, after in-depth pre-training on massive text data, LLMs have acquired abundant language patterns and semantic knowledge, enabling them to generate context-aware, coherent, natural, and logically consistent text responses. This feature perfectly aligns with the core interactive needs of game NPCs, which allows NPCs to dynamically adjust dialogue strategies and behavioral logic based on players' real-time inputs, thereby achieving a more flexible and smooth human-computer dialogue experience. For instance, in Role-Playing Games (RPGs), the communication between players and NPCs is no longer confined to fixed options; instead, players can freely express their thoughts and intentions just like interacting with real characters. The reasoning ability of LLMs can help NPCs provide responses that conform to their character settings and the current scenario, which enhances players' sense of immersion and engagement, and may even evoke unexpected emotional reactions from players [5].

In recent years, with the advent of advanced large language models such as ChatGPT and GPT-4, their outstanding performance in natural language processing tasks has attracted widespread attention and also brought new development opportunities to the gaming industry. A series of studies and practices have initially explored the application potential of LLMs in NPC dialogue generation, task guidance, plot progression, and other aspects. For example, Ryan Volum et al. applied language models to NPC decision-making and actions in Minecraft [6], and the generative agents studied in Stanford Town have demonstrated the development potential of AI agents in terms of self-abstraction awareness and planning [7]. Some game products have attempted to introduce related technologies and achieved remarkable results. However, despite the broad prospects, the application of LLMs in game NPCs is still in its infancy, facing three core technical challenges: the first is model adaptation—there is a discrepancy between the general semantic knowledge of LLMs and the "game-specific worldview" (e.g., "interstellar terminology" in sci-fi games), which may easily lead to disjointed dialogues that break the immersion; the second is real-time optimization—the inference latency of LLMs often exceeds 500ms, while games require NPC response latency to be less than 100ms to ensure a smooth experience; the third is character consistency—some LLMs may generate content that goes beyond the "identity settings" of NPCs (e.g., an NPC in the novice village revealing the "weakness of the final boss") [8].

This paper focuses on the application scenarios of Large Language Models (LLMs) in game NPCs, aiming to systematically explore some key technologies, core implementation paths, and practical application effects in this field. During the research process, a combination of theoretical analysis and empirical research will be adopted: the former is used to sort out the technical logic of LLM-driven NPCs, while the latter, based on existing game practice cases, deeply analyzes the advantages and limitations of LLM applications, and proposes practical optimization strategies and solutions for the aforementioned limitations.

It is expected that this research can provide game developers with references for technology selection (such as the adaptation scenarios of different LLM frameworks), clarify the direction for subsequent exploration for researchers, further promote the in-depth integration and innovative application of LLM technology and the gaming field, and ultimately contribute to the development of game products with more intelligent interactive features, vivid scenario adaptation capabilities, and high immersion.

2. Technical Comparison Analysis

2.1. Few-Shot Prompting Framework Based on Large Language Models

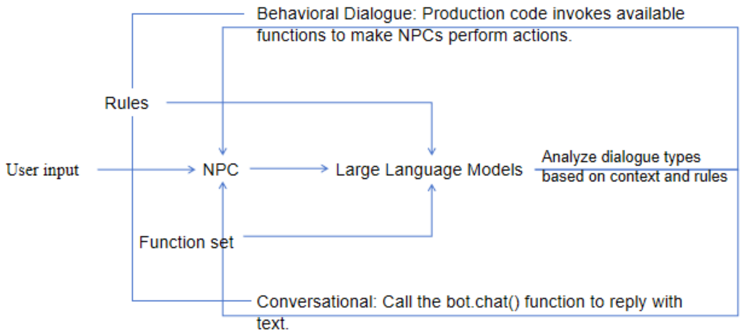

Fig 1. Few-Shot Prompting Framework Based on Large Language Models

As shown in Figure 1, this framework sends user input alongside a set of rules and functions to the large model. The large model then analyzes the input type according to the rules and invokes the corresponding function to control NPC actions [6].

To enhance model usability, this approach incorporates stop sequences, syntactic sugar, function chaining, and autoregressive prompts to ensure practical application. This method primarily addresses the challenge of using a single language model to simultaneously handle natural language dialogue and code generation, enabling functionally agentive NPC behavior.

2.2. Generative Agent Architecture

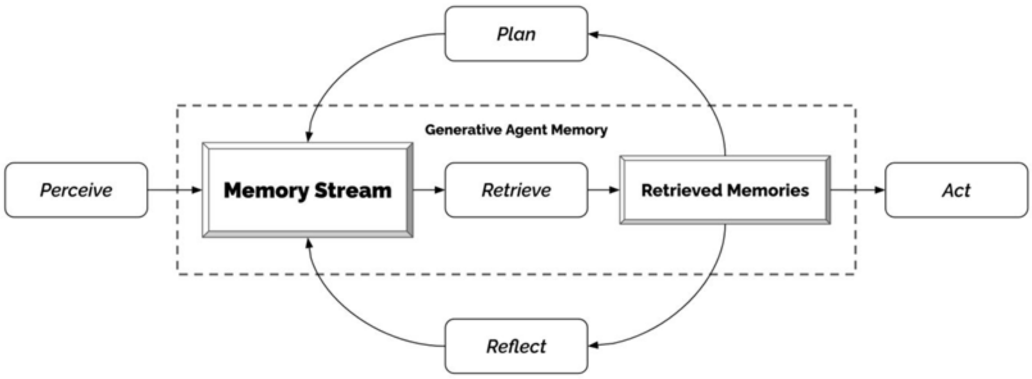

Fig 2. Illustrates the behavioral logic of the generative agent architecture [7].

Under the more basic first-order template, due to the limited context window for language processing in large models, agents cannot make decisions based on past experience. To enable agents to make decisions based on both current circumstances and past experience, the generative agent architecture offers an effective approach (figure 2).

This framework first stores information perceived by the NPC along with relevant tags into a memory stream. It selects the most weighted memory for decision-making through score-weighted retrieval, then summarizes and reflects on these memories according to certain patterns to generate more abstract self-awareness. At the start of each cycle, the agent can convert reflections and the current environment into long-term and short-term behavioral plans through “planning.” The core components of this framework are: “Memory Stream,” “Reflection,” and “Planning.” Its innovations lie in:

Memory Stream: Dynamic memory management and retrieval mechanisms;
Reflection: Recursive synthesis from fragmented memories to abstract reasoning;
Planning: Long-term coherent recursive behavioral decomposition;
Unified representation and interaction centered on natural language.

2.3. Federated Learning-Based Framework for Enhancing Game NPC Large Language Models

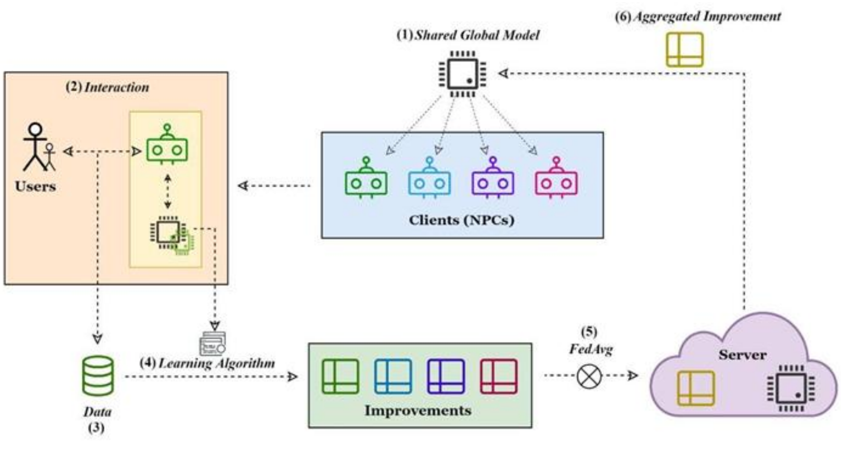

Fig 3. A federated learning-based framework model [9]

As shown in figure 3, the application process of this framework is as follows: The global model provides a model replica as the foundational framework for all NPCs. During user interactions with individual NPCs, key data (reflecting user satisfaction) is recorded as local training data. This local training data is used to calculate improvement directions for the global model, while locally finetuning the model to adapt to its own personality. Each NPC uploads its “model improvement results” to the server. These are aggregated through methods like federated averaging to update the global model, achieving collective optimization.

This approach distinguishes between the individual intelligence of each NPC agent and the overall intelligence of the NPC collective. This federated method allows for the removal of poorly trained models without impacting the global system.

3. Application

3.1. Applications of Large Language Models in Minecraft

In experiments conducted by Ryan Volum et al., they applied a few-shot prompting framework based on large language models to Minecraft. By connecting the Codex model to NPCs within the game and enhancing the application through stop sequences, syntactic sugar, chat functions, and function chains, the Codex-driven NPCs demonstrated high efficiency across multiple tasks [6].

Fig 4. Asking an NPC for item crafting recipes in the game

As shown in figure 4, for obtaining crafting recipes, the success rate reaches 85% for specified resources (wooden pickaxe, furnace, clock, pumpkin pie); for player-selected resources, the success rate is 75%. It can handle name variations (e.g., “wood pickaxe” instead of “wooden pickaxe”) and spelling errors (e.g., “fornace”).

Fig 5. Designated NPCs performing actions in the game

As shown in figure 5, in executing action tasks, the model successfully responds to player commands such as mining (e.g., digging dirt, oak logs) and crafting items (e.g., oak planks, workbenches, chests), generating novel function chains (e.g., the sequential actions “mine → move to player → deliver item”).

In these applications, the Codex model endows NPCs with several core capabilities beyond traditional NPCs, such as parsing unseen commands, generalizing new functions, maintaining multi-turn dialogue and context, and switching between code and language. This demonstrates the technology's potential in gaming scenarios.

3.2. Large Language Models Help Generate New Gameplay for Spacekill

The "AI Endgame Challenge" developed for Among Us (Chinese Version, titled Tai Kong Sha) stands as the industry’s first AI-native gameplay mode built on multi-agent large language models (LLMs). It adopts a "1 human player + N AI players" format, where human players engage in intellectual competition with AI players—all without relying on pre-set backend content. The core support for this gameplay comes from the self-developed Multi-Agent Game Technology Framework by Giant Network’s AI Laboratory. This multi-agent large model integrates the powerful natural language processing capabilities of LLMs with the collaborative features of multi-agent systems, enabling agents to make autonomous decisions and interact in complex environments.

In the "AI Endgame Challenge," AI players demonstrate high-level intelligent behaviors such as strategic planning, disguise, collaboration, and even deception. They engage in intense mental battles with human players, creating high-stakes, intellectually demanding gameplay sessions. Notably, the inherent characteristics of LLMs have brought unexpected highlights to the gameplay: due to the tendency of language models to generate outputs inconsistent with original information (known as "hallucinations" [10])—for example, an AI might "misremember" its own movement trajectory or task progress—this type of "unpredetermined error" breaks the "determinism" of traditional games. Human players can proactively guide such hallucinations (e.g., intentionally transmitting false information to AIs to induce hallucinations) and exploit the AIs’ misjudgments to mislead opponents, significantly enhancing the randomness and strategic depth of the matches. The "hallucination" issue of AIs also adds randomness and fun to the game, allowing players to actively guide hallucinations and use them to achieve their own objectives.

The Multi-Agent Game Technology Framework enables agents to make autonomous decisions and interact in complex environments. By optimizing collaboration and division of labor, it facilitates complex interaction and cooperation among AI agents, providing a new technical path and reference model for the application of large models in game NPCs.

4. Conclusion

This paper conducts a systematic study focusing on the application of large models in game NPCs, aiming to summarize several classic and practical model frameworks for the application of large models in game NPCs, thereby facilitating the development of future model frameworks. By analyzing the limitations of traditional NPCs that rely on pre-set scripts, this paper clarifies the innovative potential brought by LLMs to game NPCs. Endowed with powerful natural language understanding and generation capabilities, LLMs enable dynamic and human-like interactions between NPCs and players, significantly enhancing the sense of game immersion. On this basis, the paper focuses on summarizing and analyzing three types of typical model frameworks. In response to the challenges encountered in the practical application of these frameworks, targeted solutions are proposed.

In conclusion, by clarifying the innovative value of LLMs for NPCs, systematically analyzing the technical characteristics and applicable scenarios of three typical model frameworks, and putting forward solutions to core challenges (such as real-time performance and character consistency) in the implementation of these frameworks, this paper ultimately provides practical theoretical references and practical ideas for the in-depth application of large models in game NPCs. In the future, with the iteration of LLM lightweight and low-latency technologies, the research on model frameworks in this paper will not only promote the evolution of game NPCs from "functional interaction" to "emotional interaction" but also allow its core logic to be transferred to multiple fields. For instance, the "context-based dynamic decision-making mechanism" in the framework can be applied to virtual assistants to enhance their ability to continuously respond to users' multi-turn needs; the "multi-agent collaboration logic" can support "natural interactions between virtual characters" in metaverse social scenarios. This endows the research results with broader cross-domain application value.

References

[1] Kopel M, Hajas T. Implementing AI for Non-player Characters in 3D Video Games. In: Nguyen NT, Trawiński B, Fujita H, et al. (eds) Asian Conference on Intelligent Information and Database Systems (ACIIDS 2018). Lecture Notes in Artificial Intelligence, vol 10751. Springer, Cham. 2018: 610-619. https://doi.org/10.1007/978-3-319-75417-8_57

[2] Muhtar C U and Kaya O. Non-player character decision-making in computer games. Artificial Intelligence Review 56, 2023, 14159–14191. https://doi.org/10.1007/s10462-023-10491-7

[3] Tan W, Zhang W, et al. True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR 2024). 2024.

[4] Klissarov M, Hjelm D, Toshev A, & Mazoure B. On the modeling capabilities of large language models for sequential decision making. arXiv:2410.05656v1 [cs.AI]. 8 Oct. 2024.

[5] Marincioni A, Miltiadous M, Zacharia K, Heemskerk R, Doukeris G, Preuss M, & Barbero G. The Effect of LLM-Based NPC Emotional States on Player Emotions: An Analysis of Interactive Game Play. In 2024 IEEE Conference on Games (CoG), Leiden, Netherlands, 2024. IEEE. DOI:10.1109/CoG60054.2024.10645631

[6] Volum R, Rao S, Xu M, DesGarennes G, Brockett C, Van Durme B, Deng O, Malhotra A, Dolan B. Craft an Iron Sword: Dynamically Generating Interactive Game Characters by Prompting Large Language Models Tuned on Code. In Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022), July 14, 2022, Dublin, Ireland. Association for Computational Linguistics, 2022, pp. 25–43. https://github.com/microsoft/interactive-minecraft-npcs.

[7] Park J S, O'Brien J C, Cai C J, Morris M R, Liang P, & Bernstein M S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. 2023 (UIST '23) * (pp. 1–22). San Francisco, CA, USA: ACM. https://doi.org/10.1145/3586183.3606763

[8] Li D F. Digital Labyrinths and Waxy Wings: Prospects and Challenges of Generative AI NPCs in Video Games. Journal of Beijing Film Academy, 2024, No. 216 (12), 39-48.

[9] Hong M, Zhang K, Zhang S, & He Z. FedNPC: A Federated Learning Framework for Large Language Models in Game NPCs. In 2023 IEEE 21st Student Conference on Research and Development (SCOReD). 2023. https://doi.org/10.1109/SCOReD60679.2023.10563705

[10] Xu X, Wang Y, Xu C, Ding Z, Jiang J, Ding Z, & Karlsson B F. A Survey on Game Playing Agents and Large Models: Methods, Applications, and Challenges arXiv preprint arXiv:2403.10249, 2024. https://arxiv.org/abs/2403.10249