GPT-VR Nexus: ChatGPT-Powered Immersive Virtual Reality Experience

Jiangong Chen† Tian Lan‡ Bin Li†
†Department of Electrical Engineering, The Pennsylvania State University, Pennsylvania, USA
‡Department of Electrical and Computer Engineering, George Washington University, Washington, USA
{jiangong, binli}@psu.edu, tlan@gwu.edu

***Abstract*—The fusion of generative Artificial Intelligence (AI) like ChatGPT and Virtual Reality (VR) can unlock new capabilities to interact with VR environments through natural language, e.g., automatically generating and animating 3D scenes using only audio input. However, significant gaps exist in supporting this vision: 1) limited AI data processing ability for accurate VR context comprehension; 2) AI "hallucinations" that leads to misaligned responses in VR; and 3) the absence of tools for directly translating AI's responses into VR scene creation and animated interactions. To address these challenges, we introduce GPT-VR Nexus, a novel framework creating a truly immersive VR experience driven by an underlying generative AI engine. It employs a two-step prompt strategy and robust post-processing procedures, without the need of fine-tuning the complex AI model. Our experimental results show quick responses of the VR environment to a diverse range of user audio requests/inputs in merely a few seconds.**

***Index Terms*—Virtual Reality, Generative AI, ChatGPT**

I. Introduction

The advent of generative Artificial Intelligence (AI) such as ChatGPT has ignited a wave of exploration into its diverse application scenarios, such as content creation and software development. The fusion of Virtual Reality (VR) with generative AI has the potential to extend user interaction beyond the conventional realms of text, vision, and voice, thus creating a truly immersive experience. However, enabling this vision of the GPT-VR nexus must address several key challenges: 1) the need to improve generative AI's contextual understanding of user requests/inputs in VR settings, by effectively utilizing the vast amounts of VR data to generate relevant and accurate responses; 2) the hallucination problem (see [1]) due to possibly inapplicable responses from generative AI, violating physical constraints and significantly degrading user experience; and 3) the lack of tools mapping generative AI's responses directly to drive VR scene creation and animated interactions.

In this demo, we present a novel GPT-VR nexus to bridge the gap between generative AI and the VR environment. It enables a ChatGPT-powered VR experience – e.g., automated generation of 3D scenes and interaction with VR objects – from VR user audio inputs/commands. In particular, we propose a two-step strategy to process VR contextual data. It first categorizes user requests/inputs and then queries relevant data for precise prompts. To achieve the most relevant response, we develop an additional processing layer for response validation and adjustment. It creates VR environment/scene and animated interaction directly from ChatGPT responses. By showcasing the novel capabilities by integrating advanced generative AIs like ChatGPT into VR, this demo underscores the immense potential of building a GPT-VR nexus in creating novel interactive experiences, as well as significantly reducing content/scene development costs.

II. System Architecture

Fig. 1 depicts the system architecture of GPT-VR Nexus. The user-initiated interaction begins with an audio recording, activated via the VR controllers. This audio input is first transcribed into text using OpenAI's Whisper model, which is then combined with a custom-designed prompt and relayed to the GPT-4 Turbo model. Upon receiving the processed response from the server, the Nexus undertakes different strategies based on the categorization of the response. For responses that are in plain text, the Nexus employs OpenAI's text-to-speech (TTS) model to convert the text responses into voice outputs, which are then emitted from the virtual avatar's audio source within the VR environment, thereby simulating a natural human conversation. For responses that involve more complex commands, the Nexus parses these commands and distributes them to various specialized modules within the system. Note that generating an entire response at once can be time-consuming; hence, we opt for delivering the response in smaller, manageable chunks. These chunks are subsequently merged into coherent sentences for further processing. In the next few sections, we will introduce the core design of GPT-VR Nexus to deal with the unique challenges.

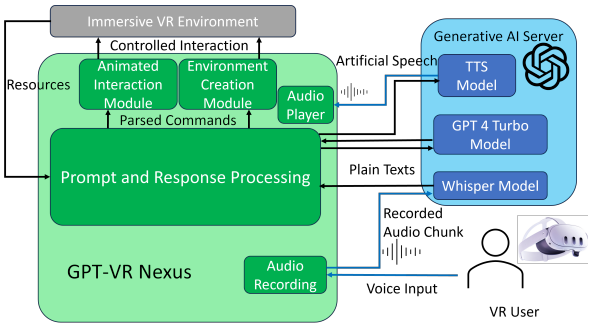

Fig. 1: System architecture.

Prompt and Response Processing. To effectively utilize VR contextual data for user requests, GPT-VR Nexus employs

a two-step approach. Initially, upon receiving a user request, the Nexus issues a *primary prompt* to the AI server, which categorizes the request into specific patterns. Following this categorization, the Nexus sends a *refined prompt* tailored to the identified patterns, which requests a more detailed response and includes specific queries for VR context knowledge. By doing so, the AI focuses only on relevant data, thereby minimizing the influence of extraneous, potentially misleading information. To counteract the unreliable AI responses, the Nexus employs an additional processing layer to parse and adjust the parameters for responses that involve complex commands. The principles of such processing will be elaborated in the module design.

Environment Creation Module. When a user requests an environment setup, the Nexus queries our project prefab resources and incorporates their names and properties, like the model size, into the refined prompt. The server's response contains both descriptive text and specific properties for object placement, including prefab names, positions, and orientations. Additionally, objects are assigned layer properties, with lower-layer objects prioritized for placement at the base level, establishing the foundational layout of the environment. The server also identifies any necessary base objects for others (e.g., a table as a base for a pen). To create a coherent environment, the Nexus leverages the renderers of the base and other same-layer objects in the processing procedure to ensure the positions fit within the bounds and prevent collision.

Animated Interaction Module. GPT-VR Nexus's ability to understand its virtual environment equips it with the capacity for intuitive user engagement. When a user request entails knowledge about the VR environment, the Nexus efficiently collects relevant data about nearby objects, such as their positions and names, and integrates this information into the refined prompt. Armed with contextual data, the Nexus autonomously executes actions to fulfill user requests, ranging from guiding users to designated objects to bringing the items within the virtual realm. The response for these interactions mirrors the format used in the environment creation module. However, in this context, the server's response directs the movements of the Nexus's avatar instead of creating objects. To achieve vividly lifelike animation, the Nexus strategically enqueues the received commands as sequential keyframes and interpolates actions between these keyframes.

III. Demonstration

The demo system is developed using Unity and Meta XR All-in-One SDK. It's designed for commercial VR headsets, such as Oculus Quest 2/3, allowing users to interact seamlessly with GPT-VR Nexus at a low cost. Communication with the OpenAI API is facilitated through an unofficial OpenAI wrapper library[1]. However, we encounter some challenges with text response streaming from the GPT-4 Turbo model using this package. To address this, we design a Python-based forward server, running on an edge server and utilizing the official OpenAI Python API. The communication delay between this edge server and the Nexus is minimal, only tens of milliseconds, which is negligible compared to the response generation time. A demo video is available at [2].

[1]https://github.com/OkGoDoIt/OpenAI-API-dotnet

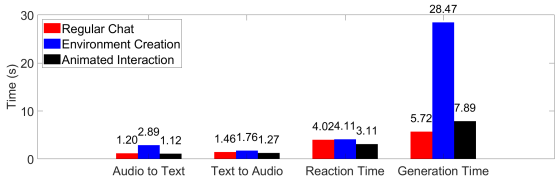

Fig. 2: Time consumption for GPT-VR Nexus's pipeline.

For performance evaluation, we conduct tests using ten pre-recorded audio files representing typical user requests. These tests are repeated ten times to collect average time consumption results, as shown in Fig 2. We define *Reaction Time* as the time from audio recording completion to the first audio response, while *Generation Time* refers to receiving the complete AI response. The latter is longer for complex tasks like environment creation due to the need for detailed AI guidance. However, our system's streaming design minimizes impact on user experience, with overall reaction times remaining low. Our post-processing effectiveness is visually demonstrated through figures comparing initial and refined object placements in the virtual environment, as shown in Fig 3. We can observe that the collision issue has been addressed and all office supplies are within the table's boundary.

(a) Initial placement. (b) Refined placement.

Fig. 3: Post-processing effect.

IV. Conclusion

In this demo, we presented GPT-VR Nexus, a solution designed to seamlessly integrate generative AI with VR setups. We introduced a two-step prompt strategy for efficiently utilizing VR contextual data for user requests. Additionally, we proposed tailored post-processing procedures to address the challenge of unreliable AI responses. Following those design principles, we showcased two distinct modules to handle various user requests, ranging from scene creation to animated interaction within the VR environment. Our evaluations reveal that GPT-VR Nexus consistently reacts to users in just a few seconds, demonstrating its effectiveness and responsiveness.

References

[1] V. Rawte, A. Sheth, and A. Das, "A survey of hallucination in large foundation models," *arXiv preprint arXiv:2309.05922*, 2023.
[2] "Demo Video." [Online]. Available: https://sites.psu.edu/binli/vrchatgpt/