

# GPT-VR Nexus: ChatGPT-Powered Immersive Virtual Reality Experience\*

Jiangong Chen<sup>†</sup>

The Pennsylvania State University

Tian Lan<sup>‡</sup>

George Washington University

Bin Li<sup>§</sup>

The Pennsylvania State University

## ABSTRACT

The fusion of generative Artificial Intelligence (AI) like ChatGPT and Virtual Reality (VR) can unlock new interaction capabilities through natural language. 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 fine-tuning the complex AI model. Our experimental results show quick responses to various user audio requests/inputs.

**Index Terms:** Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Computing methodologies—Artificial intelligence—Distributed artificial intelligence—Intelligent agents; Computing methodologies—Artificial intelligence—Natural language processing—Natural language generation; Computing methodologies—Artificial intelligence—Natural language processing—Speech recognition

## 1 INTRODUCTION

The advent of generative Artificial Intelligence (AI) such as ChatGPT [3] has ignited a wave of exploration into its diverse application scenarios (see [2, 7]), such as content creation and software development. The fusion of Virtual Reality (VR) with generative AI (see [1, 6]) 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 [5]) 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 environments/scenes and animated interaction directly from ChatGPT responses. By showcasing the novel capabilities of integrating advanced generative AIs like ChatGPT into VR, this demo underscores the immense potential of building a GPT-VR nexus to create im-

mersive and interactive experiences and significantly reduce content/scene development costs.

## 2 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 [4], 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.

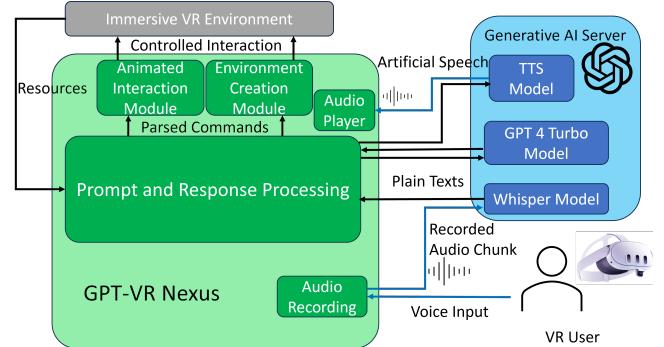


Figure 1: System architecture.

**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 following module design subsections.

**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

\*This work has been supported in part by NSF under the grants CNS-2152610 and CNS-2152658, and ONR under the grant N00014-23-1-2850.

<sup>†</sup>e-mail: jiangong@psu.edu

<sup>‡</sup>e-mail: tlan@gwu.edu

<sup>§</sup>e-mail: binli@psu.edu

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.

### 3 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<sup>1</sup>. 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 online<sup>2</sup>.

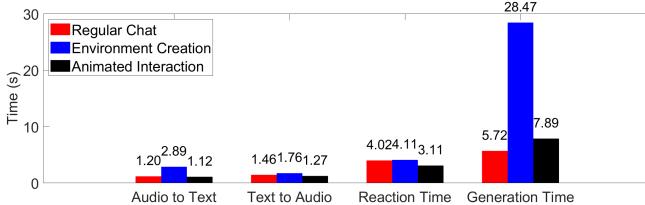


Figure 2: Time consumption for GPT-VR Nexus's pipeline.

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 collect the audio-to-text time and text-to-audio time for each audio recording and plain text sentence uploaded to the AI server, respectively. Moreover, 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. On the other hand, 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 observe that without post-processing, the initial placement may result in collisions with objects on the table, and misplacement of office supplies even fall to the ground. However, these issues are effectively rectified in the refined placement following post-processing.



(a) Initial placement.

(b) Refined placement.

Figure 3: Post-processing effect.

### 4 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

- C. Bussell, A. Ehab, D. Hartle-Ryan, and T. Kapsalis. Generative ai for immersive experiences: Integrating text-to-image models in vr-mediated co-design workflows. In *International Conference on Human-Computer Interaction*, pp. 380–388. Springer, 2023.
- F. Fui-Hoon Nah, R. Zheng, J. Cai, K. Siau, and L. Chen. Generative ai and chatgpt: Applications, challenges, and ai-human collaboration, 2023.
- OpenAI. Chatgpt-4. <https://chat.openai.com/chat>, 2024. Large language model.
- A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pp. 28492–28518. PMLR, 2023.
- V. Rawte, A. Sheth, and A. Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.
- M. Taylor, Z. Muwaffak, M. Penny, B. Szulc, S. Brown, A. Merritt, and S. Hilton. The rise of the ai scientist: Unleashing the potential of chat-gpt powered avatars in virtual reality digital-twin laboratories. *ChemRxiv*, 2023.
- R. Wahid, J. Mero, and P. Ritala. Written by chatgpt, illustrated by midjourney: generative ai for content marketing. *Asia Pacific Journal of Marketing and Logistics*, 35(8):1813–1822, 2023.

<sup>1</sup><https://github.com/OkGoDoIt/OpenAI-API-dotnet>

<sup>2</sup><https://sites.psu.edu/binli/vrchatgpt/>