Design and Preliminary Evaluation of Generative AI-Driven Multi-Agent NPCs for VR Communication Training

Ooi Wei Kit
Faculty of Computing And Information Technology
Tunku Abdul Rahman University of Management and Technology
Selangor, Malaysia
eddie.weikit@gmail.com

***Abstract*—This study presents the development and evaluation of a Virtual Reality (VR) public speaking training application enhanced with Generative AI-driven multi-agent interactions. Developed to provide learners with an immersive platform for practicing presentation skills in realistic scenarios, the proposed system integrates multi-agent non-player characters (NPCs) powered by generative AI. These NPCs are capable of simulating social interactions, providing supportive feedback, and adapting to user performance. The VR application was built using Unity and integrates Ready Player Me avatars, Whisper speech-to-text, text-to-speech synthesis, and ChatGPT-based feedback mechanisms. A structured test case framework was designed to evaluate NPC realism across five levels: surface realism, spatial awareness, interactive realism, social multi-agent realism, and cognitive personality realism. To assess system performance and user experience, the System Usability Scale (SUS) and Igroup Presence Questionnaire (IPQ) were administered to a sample of participants. The results indicate that users found the system engaging, immersive, and effective for public speaking practice, yielding positive feedback on the generative AI evaluation and NPC responsiveness. However, limitations were identified in NPC social realism regarding group dynamics and long-term role consistency. Future work will focus on enhancing multi-agent social awareness, improving NPC collaboration, and expanding the sample size for broader validation. The findings highlight the potential of combining VR with generative AI and multi-agent systems to advance communication training.**

Keywords—Virtual Reality, Generative AI, Multi-Agent Systems, Public Speaking Training, Unity Engine

I. INTRODUCTION

A. Background & Content

Public speaking is a foundational skill in professional and academic domains, yet it remains one of the most prevalent social phobias globally, commonly known as glossophobia [1]. Traditional methods of overcoming public speaking anxiety—such as practicing in front of mirrors, recording videos, or participating in small-group peer sessions—often fail to replicate the psychological pressure of a live audience [2]. Conversely, organizing real-world practice sessions with large audiences is logistically challenging, resource-intensive, and can induce severe anticipatory anxiety for the learner. Consequently, there is a critical need for scalable, controllable, and highly immersive training environments that allow users to build confidence at their own pace.

In recent years, Virtual Reality (VR) has emerged as a disruptive paradigm in simulation-based training. By leveraging high-fidelity head-mounted displays (HMDs), VR induces a psychological state of "presence"—the subjective sensation of being physically inside the virtual space [3]. This immersive capability makes VR uniquely suited for exposure therapy and behavioral training, as it can evoke authentic physiological and emotional responses from the user within a safe, simulated boundary [4].

However, conventional VR public speaking simulators suffer from a significant limitation: static or rigidly scripted virtual audiences. Traditional non-Player Characters (NPCs) in these environments follow pre-baked animations and repetitive behavioral loops, failing to capture the unpredictable, dynamic nature of a live human audience [5]. To bridge this gap, the integration of Generative AI and Multi-Agent Systems (MAS) offers a transformative solution. By embedding large language models (LLMs) and intelligent agent architectures into virtual entities, NPCs can transition from static props into autonomous, reactive agents [6]. These multi-agent systems can simulate complex crowd dynamics, display emergent social behaviors, maintain distinct cognitive personalities, and provide real-time, adaptive feedback based on the speaker's vocal and behavioral performance. Synthesizing VR with Generative AI-driven multi-agent frameworks thus represents the next frontier in creating responsive, high-fidelity communication training ecosystems.

B. Problem Statement

Despite the clear advantages of using virtual reality for communication training, existing platforms suffer from several practical limitations that restrict their effectiveness. First, conventional simulations rely heavily on pre-scripted, repetitive behavior loops for audience characters. These virtual listeners perform the exact same gestures and canned responses regardless of what the speaker says, removing the unpredictability of real-life social encounters. As a result, users quickly get used to the static setting, making the training less effective for reducing public speaking anxiety over time.

Second, current systems struggle to simulate natural group dynamics among multiple characters. A live audience functions as an interconnected group where a speaker's

remarks can cause shared reactions, such as group nodding or shifting attention. Current VR applications treat characters as isolated entities, failing to capture believable collective behaviors.

Finally, traditional training tools lack context-aware, detailed performance analysis. While some applications can track basic delivery metrics like speaking rate or head orientation, they cannot interpret the actual meaning of the speech or provide meaningful, tailored recommendations. Without an intelligent conversational system to analyze speech content and guide audience responses in real time, VR public speaking tools remain superficial presentation checklists rather than genuinely adaptive social environments.

C. Objectives and Contributions

The primary objective of this project is to develop an immersive VR communication training system that replaces static training methods with a dynamic, Generative AI-driven multi-agent environment. By moving away from fixed scripts and classroom-based theoretical learning, this research aims to provide a tailored experience that reacts to the user's specific vocal performance.

In sum, this paper makes the following contributions:

- A virtual reality public speaking application featuring four training environments: a *Hallway*, a *Classroom*, a *Golden Dragon Kopitiam*, and a *Speaking Hall*. These spaces are populated by interactive non-player characters (NPCs) that use a large language model (LLM) backend to generate natural conversational dialogue and spontaneous gestures based on the user's spoken words.
- A live speech tracking system that measures conversational metrics for every turn, including speaking duration, word counts, and the use of filler words. The application displays these metrics as real-time visual assistance directly on the user interface during active practice sessions.
- An automated evaluation system that compiles full session data at the conclusion of a training module. This system computes performance averages and uses generative AI to analyse dialogue text, generating personalized recommendations for improvement.
- An interactive speech coaching feature that pairs synthesized text-to-speech with virtual character models. This allows an NPC to act as a virtual mentor, delivering an active, spoken summary critique to help users naturally build speaking confidence and fluidity.
- A probabilistic multi-agent conversation framework designed to simulate unstructured social interactions group dynamics. By implementing an intentional balance of character-directed routing and random speaker selection (a 70primary response chance paired with a 30inter-agent commentary loop), the system moves beyond isolated presentations to train users in conversational adaptability, multi-character engagement, and spontaneous public discussion within a culturally grounded social setting.

By achieving these objectives, this study contributes a scalable architecture for the next generation of AI-integrated VR training tools, focusing on the validation of technical performance and user growth.

II. Related Work

A. Fundamentals of Communicative Fluency and Training

Effective communication relies on a dual mastery of non-verbal cues, such as interpreting body language, and active listening to build genuine interpersonal relationships [26]. To preserve mutual understanding in real-time interactions, conversational training must focus heavily on balancing linguistic fluency with grammatical accuracy [24]. Fluency dictates the capacity to speak smoothly, naturally, and confidently without unnatural pauses, which directly facilitates spontaneous social engagement [24]. Conversely, linguistic accuracy ensures semantic clarity through correct word usage and pronunciation, although over-emphasizing accuracy during active speech can disrupt natural delivery and lower speaker confidence [24]. Traditional learning methods frequently struggle to provide the adaptive conversational settings necessary to develop this balance [24]. Consequently, simulation-based training tools that integrate analytical feedback and scenario-driven exposure are required to help users refine their delivery and build confidence in immersive environments [24], [29].

B. VR Interactivity and Non-Verbal Audience Animation

Immersive Virtual Reality (VR) environments leverage head-mounted displays (HMDs) to establish a psychological state of presence, making them highly effective for soft-skills training and behavioral exposure [4], [6]. To maintain high user immersion during virtual meetings or presentations, it is crucial that the non-verbal behaviors of virtual characters reflect realistic crowd states [13]. Specific avatar actions, such as shifting posture, nodding, or demonstrating attentiveness, serve as vital communication status cues that convey a virtual audience's engagement and availability [13].

To automate these visual responses at a micro-level, developers utilize lip-syncing frameworks to link voice audio directly to character mesh deformations. For example, runtime lip-sync components process input audio streams to adjust 3D mesh blendshapes representing core vowel phonemes (A, E, I, O, U)alongside base animator controllers that govern structural states like idling or talking. However, while these individual visual subsystems provide consistent surface-level behaviors, they remain strictly confined to scripted parameters and lack the context-aware depth required to mimic spontaneous human conversation.

C. Multimodal Emotion Dynamics and Intelligent Agents

To improve character believability, recent research emphasizes multimodal emotional expression where virtual characters synthesize facial expressions, body movements, and vocal tones simultaneously [1]. While facial expressions remain the most recognizable indicator of a character's internal state, coordinating emotional cues across multiple modalities significantly enhances the perceived depth and realism of an intelligent virtual agent [1]. These emotional dynamics are mathematically simulated by embedding psychological frameworks, such as the valence-arousal model, into the underlying agent architecture [23]. By

calculating shifts in valence (the positive or negative quality of a feeling) and arousal (the intensity of the emotion), multi-agent systems can simulate emotion contagion [23]. This mathematical modeling allows characters to influence each other's emotional states over time, producing realistic group dynamics and a cohesive conversational atmosphere [23].

D. Generative Multi-Agent System Architecture

The deployment of Large Language Models (LLMs) has transformed conversational agent design by shifting characters away from rigid, pre-scripted dialogue loops [2], [11]. Within proactive peer support and training networks, generative agents utilize natural language processing (NLP) to deliver empathetic, context-aware, and scalable behavioral responses tailored to a user's verbal input [15]. To synchronize these textual outputs with physical expressions, contemporary prompt-engineering methodologies involve embedding symbolic non-verbal directives directly into the LLM core before interaction begins [12]. By training the model on a predefined list of symbolic gestures categorized across distinct motion clips, the generative backend learns to seamlessly output gesture tags inside parentheses alongside standard verbal replies [12].

When integrated with multi-agent systems, these intelligent frameworks adapt dynamically to individual user traits to foster personalized learning outcomes [14]. Utilizing multi-agent architectures inside VR spaces further allows characters to distribute unique roles—such as peers, mentors, or challengers—creating a holistic and collaborative educational ecosystem that drives engagement [14], [16].

III. System Architecture

A. System Overview Pipeline

The developed framework operates as a distributed, multi-tiered architecture that bridges real-time physical vocal input with cloud-based generative AI systems and local VR rendering loops within the Unity game engine.

The application architecture follows a strict sequential data execution loop:

- **Vocal Capture:** The user interacts inside the virtual reality environment using a head-mounted display (HMD). When the user initiates a conversational turn, the system captures their live voice input through the hardware microphone array.
- **Speech-to-Text (STT) Processing:** The raw audio data stream is compressed and forwarded asynchronously to the OpenAI Whisper API backend. Whisper transcribes the speech into clean textual strings while preserving structural elements.
- **Analytics Layer Extraction:** Concurrently, a local tracking component processes runtime audio telemetry to calculate immediate metadata metrics, including total word counts, turn duration, and the presence of vocal filler words.
- **Generative AI Processing:** The transcription, conversational metadata, and character system prompts are sent directly to the OpenAI ChatGPT engine.
- **Character Dialogue and Behavior Execution:** The language model returns a structured response containing both the character's spoken lines and their physical action tags. At runtime, the application parses this response: it sends the dialogue text to a text-to-speech (TTS) engine for synchronized audio playback, updates the user interface with live performance metrics, and translates the action tags into physical gestures and animations for the virtual audience.

B. Real-Time Conversational Metrics Collection

To support detailed performance tracking and deliver context-aware evaluation summaries, the application establishes a modular runtime data processing layer. This engine manages three isolated computational telemetry components: a turn duration stopwatch, a word counter, and a vocal filler string matched array.

During an active conversational turn, the system maintains strict recording boundaries. The turn duration metric calculates total elapsed execution time via code routines synchronized with real-time hardware timestamps. Upon receiving the transcribed text output from the STT endpoint, the word counter tokenizes the string using whitespace boundaries to determine precise spoken counts. Simultaneously, a string-matching search array scans the transcribed text for high-frequency linguistic filler words (e.g., "uh", "um", "like", "so").

These parameters populate a localized structural record representing the unique turn data block. At the end of every active turn, these individual values are updated instantly on the user's primary interface display, providing continuous feedback throughout the simulation. Upon session termination, the data architecture aggregates the

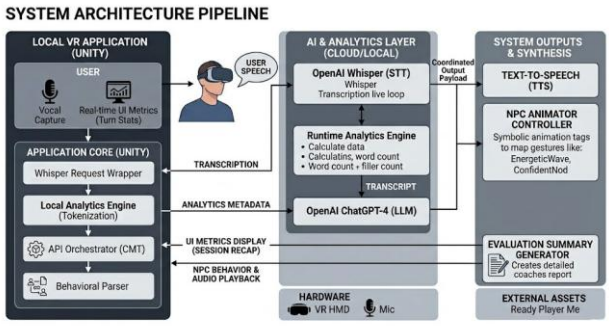

Fig. 1. Overall System Architecture Pipeline.

chronological array of turn records to compute overall historical averages, tally full-session filler totals, and track overall duration metrics. This compiled framework is passed to a dedicated LLM prompt layer that acts as an automated virtual coach, translating raw numerical arrays into constructive, personalized presentation critiques spoken directly to the user.

C. Multi-Agent Prompt Engineering and Character Animation

Creating responsive virtual audience behaviors relies on structured system prompts that guide the large language model (LLM). Instead of generating generic responses, the

prompt establishes specific background contexts, behavioral guidelines, and personality roles for each non-player character (NPC)—such as a supportive classmate, a critical evaluator, or a distracted listener.

To connect text generation with physical movements in Unity, the system prompt requires the model to return behavior tags along with spoken dialogue. The model is instructed to embed non-verbal action cues inside parentheses next to the text. A typical response follows this structured layout:

```
[Thought: User seems hesitant] (ConfidentNod) I get what you mean lah,
don't worry. Just keep going, I am listening.
```

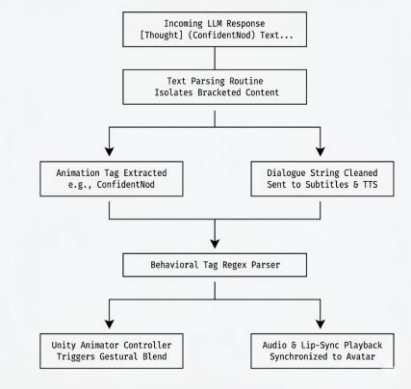

Fig. 2. Prompt Engineering Pipeline.

During runtime, a text parsing routine scans the incoming response string for these parenthetical markers. It isolates the text within the brackets, filters it out of the visible subtitles, and treats it as an animation instruction. This instruction maps directly to parameters in the Unity Animator Controller, triggering a specific gesture—such as a wave, a nod, or an unengaged head tilt—that runs smoothly alongside character dialogue.

Beyond basic dialogue, the conversation manager monitors the discussion for specific topic-related keywords. When a user asks an NPC about a personal interest, the text parser catches the topic and allows the character to showcase unique background properties defined in its identity file. For instance, if an NPC with a dancing hobby is asked about their interests, the script intercepts the corresponding tag in the response stream and triggers the `PerformTalent()` function. This instantly cross-fades the character's animator into a custom dance motion asset while maintaining synchronized lip-syncing and dialogue audio.

This script-based tracking loop also acts as a state controller for the overall application flow. When the parsing routine identifies conversational phrases that signal the end of a presentation (such as the `END_CONVO` tag or phrases like "That is all for my speech"), it halts the recording loop. The application bypasses the standard conversation turns and switches directly to the evaluation pipeline, compiling session metrics to display on the user's feedback panel.

IV. Technical Implementation

A. Development Environment and Core VR Rig Setup

The public speaking training simulation was developed using the Unity 2022.3 LTS engine paired with the Universal Render Pipeline (URP). URP was selected to optimize rendering performance and maintain a stable $72 - 90\ MHz$ refresh rate directly on standalone hardware, specifically the Meta Quest 2 and Meta Quest 3 head-mounted displays.

Hardware interactions and tracking are abstracted through the Unity XR Interaction Toolkit (XRI) framework in conjunction with the OpenXR plugin backend. The player container is managed by an `XROrigin` rig, which coordinates stereoscopic camera tracking and input actions from the left- and right-hand controllers. Locomotion across the training spaces is handled via an XRI `LocomotionManager`, which implements continuous smooth movement along with snap-turn rotation adjustments.

To allow users to navigate the training dashboard and interact with performance displays, both controllers feature an `XRController` component linked to `XRRayInteractor` lines. These interactors cast physical rays that register input selections on World Space UI canvases using an `XRUIInputModule`, allowing users to comfortably adjust settings, choose training scenes, or trigger evaluation reports using controller trigger presses.

Virtual character generation relies on the Ready Player Me (RPM) avatar system. Avatars are configured using the RPM SDK, which allows developer-designed character templates to download at runtime as JSON configurations and instantiate directly into the active scene tree. Characters utilize animation controller state machines to manage their background behaviors:

- Micro-Behaviors: Natural eye blinking is automated via short-interval coroutine timers that adjust target blendshapes values on the character's `SkinnedMeshRenderer.`
- Spatial Perception (`NPCFov.cs`)**:** Each virtual character maintains independent vision properties defined by an explicit vision range (`viewRange = 8f`) and a 60° angular field of view (`fieldOfView = 60f`) cone. The orientation script tracks the user's primary headset coordinates using real-time vector math.

When the player enters this field of view and passes inside a $2 - meter$ proximity threshold, a local execution thread temporarily overrides the character's default idle state. The character smoothly adjusts its animation rigging weights using programmatic linear interpolation (`Mathf.Lerp`), causing the character to face the player, play a spoken greeting clip, and execute a welcoming hand wave gesture. For comparison scenarios, non-generative characters utilize native Unity `NavMeshAgent` components to patrol predefined paths while running these same proximity checks.

B. Multi-Agent Field of View (FOV) and Attention Mechanics

To simulate authentic audience engagement, each non-player character (NPC) features a dynamic Field of View (FOV) and head-tracking sub-system. This sub-system governs when an

agent becomes aware of the user and controls how the character physically adjusts its posture to maintain visual attention.

- Geometric FOV Condition: The physical position of the user's headset is defined as vector $\mathbf{P} \in \mathbb{R}^3$, and the root position of the individual NPC head joint is defined as vector $\mathbf{H} \in \mathbb{R}^3$. The directional offset vector pointing from the agent's head to the user, denoted as $\mathbf{D}$, is established by:

$$\mathbf{D} = \mathbf{P} - \mathbf{H}$$

The absolute Euclidean distance between the NPC head and the user is computed as $d = \|\mathbf{D}\|$. For an agent to recognize the user, the distance must fall within a maximum operational detection radius ($d_{\max} = 8.0m$).

Let $\hat{\mathbf{F}}$ represent the normalized forward-facing gaze vector of the NPC head joint (npcHead.forward), and let $\hat{\mathbf{D}} = \frac{\mathbf{D}}{\|\mathbf{D}\|}$ signify the normalized user direction vector. The directional tracking angle θ between the agent's line of sight and the user is determined using the vector dot product:

$$\theta = \arccos\left(\hat{\mathbf{F}} \cdot \hat{\mathbf{D}}\right)$$

An agent's visibility flag (I_{fov}) evaluates to a binary state (1 for active tracking, 0 for idle) based on whether the user falls inside both the radial distance threshold and the angular constraint boundary ($\theta_{\max} = 60°$):

$$I_{fov} = \begin{cases} 1, & if\ \theta \le \theta_{\max}\ and\ d \le\ d_{\max} \\ 0, & \text{otherwise} \end{cases}$$

- *Dynamic Rig Weight Interpolation:* To prevent unnatural, instantaneous snapping when an agent alters its gaze, the head-tracking animation rig weight (w)is smoothly adjusted over time using linear interpolation (Lerp). The target weight equilibrium (w_{target}) transitions dynamically based on the binary tracking state and the maximum allowed weight limit ($w_{\max} = 1.0$):

$$w_{\text{target}} = I_{fov} \cdot w_{\max}$$

The progressive adjustment of the runtime rig weight across sequential frame intervals Δt is calculated using a blending speed scalar ($\alpha = 3.0$):

$$w_t = w_{t-\Delta t} + \left(w_{\text{target}} - w_{t-\Delta t}\right) \cdot \alpha \cdot \Delta t$$

- *Proximity-Based Eye Elevation:* When a user moves within close personal proximity of an agent ($d_{\text{close}} = 2.0\text{m}$), the NPC must elevate its vertical gaze to maintain natural eye contact. This behavioral layer modulates a specific facial blend shape weight ($W_{\text{eyes}}, scale\ to 100$) based on a baseline scaling constant ($\beta = 0.035$):

$$W_{\text{eyes}} = \begin{cases} \beta \times 100, \text{if } d \le d_{\text{close}} \\ 0, \qquad \text{otherwise} \end{cases}$$

This mathematical framework ensures that the multi-agent audience displays highly fluid, positionally reactive visual feedback, reinforcing the interactive realism of the communication workspace.

C. Scene Management and Simulation Flow

The operational state-machine of the application follows a progressive hierarchical sequence across four unique immersive environments managed via a central scene routing layer:

- Main Menu: Manages initial user configuration, headset orientation alignment, and scene selection.
- Hallway (Pre-Scenario Scene): Outfitted exclusively with deterministic, non-GPT NPCs. These agents utilize lightweight backgrounds patrol state trees and state audio trigger components to simulate a bustling school hallway, providing baseline environment presence without API execution overhead.
- Active Presentation Classrooms (Classroom & Speaking Hall Scenes): Designed for single-agent evaluation. The user approaches a classroom, selects a speech topic, and engages a single Generative AI student who observes the presentation, tack turn-based telemetry, and feeds parameters back into the automated feedback announcer system.
- Golden Dragon Kopitiam Scene: A specialized social simulator running a multi-agent conversational thread. This environment shifts the paradigm from structural public speaking to real-time unstructured group chat, testing the user's conversational adaptability and fluency within localized cultural setting.

D. Audio Processing and Asynchronous API Integration

Vocal capture is managed natively via Unity's `Microphone.Start()` framework, recording high-fidelity voice data at a sampling rate of 44100 Hz into localized cyclic buffers. To avoid local memory bloat, recording lengths are bounded to a maximum allocation window (12 seconds). Upon turn completion, the raw floating-point data arrays are compiled into compliant binary streams via a custom utility class (`SaveWav.cs/WavUtility.cs`), creating an optimized output.wav file in the device's persistent storage.

Asynchronous cloud communication is orchestrated through a specialized network wrapper layer (`WhisperSpeechToTextMultiAgent.cs`). The compressed binary .wav payload is packaged into a structured network request and dispatched asynchronously to the OpenAI Whisper API endpoint using the `CreateAudioTranscriptionsRequest` wrapper:

```
var req = new CreateAudioTranscriptionsRequest {
    FileData = new FileData() { Data = data, Name = "audio.wav" },
    Model = "whisper-1",
    Language = "en"
};
var res = await openai.CreateAudioTranscription(req);
```

Upon receiving the string transcription from the Whisper endpoint, the system hands execution control over to the primary multi-agent manager (`ChatGPTMultiagent.cs`). This manager maintains a historical string list (`List<ChatMessage>`) representing the interactive session timeline.

To guarantee that the virtual audience provides responses consistent with classroom physics and personal identities, the

local manager updates the underlying Large Language Model using custom-engineered system instructions. The underlying conversational prompt binds the generative character to strict constraints, injecting the parsed presentation topic parameters directly into the prompt sequence:

```
You are a curious student in a VR classroom listening to a presentation.

The topic of the presentation is: {topicTitle}.
Here are the speaker's metrics from this session:
- Duration: {duration:F1} seconds
- Words per minute: {wpm:F1}
- Filler words used: {fillerCount}

Only ask questions or make remarks that are directly related to this topic.
Respond with short, relevant questions or remarks based on the presenter's speech.
Do not break character, do not mention you are an AI.
If my reply indicates that I want to end the conversation, do not ask any further questions. Instead, reply with a short friendly farewell like: 'It was nice to hear your presentation, see you around!' and end your message with the phrase END_CONVO.
```

The system forwards this multi-layered message array to the cloud-based `gpt-4o-mini` architecture. The returned text stream is parsed in real time to isolate conversational text from gesture parameters. The final clean dialogue string is routed directly to a localized Text-to-Speech (TTS) engine, which downloads the vocal audio stream and executes playback across localized 3D spatialized audio sources positioned directly at the character's head coordinates.

E. Real-Time Conversational Metrics Collection

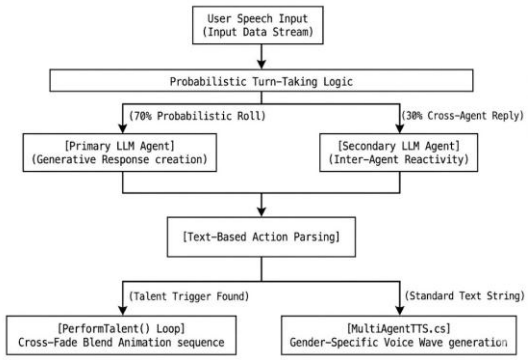

Fig. 3. Multi-agent Program's Architecture.

Upon detecting the termination flag token (`END_CONVO`) within the conversational response array, or receiving an explicit user-initiated UI closing trigger, the application suspends active dialogue states and instantiates the global evaluation engine.

Data processing is managed via a dedicated analytical script (`FeedbackSummaryAnnouncer.cs`). This class implements an asynchronous execution loop triggered when the user clicks the primary interface panel's summary query button (`OnReceiveClicked`). The script interfaces with a global tracking component (`GameplayPanelManager`) to aggregate cumulative per-turn metrics, calculating localized historical averages across speech duration, full-session word metrics, and verbal filler counts:

```
float avgDuration = panelManager.GetAverageDuration();
float avgWords = panelManager.GetTotalWordCount() / (float)Mathf.Max(1, panelManager.GetTranscriptionCount());
float avgFillers = panelManager.GetAverageFillersPerTurn();
```

These generated numerical floating-point averages are serialized directly into a custom system critique prompt:

```
You are a supportive teacher summarizing a student's speaking performance.
Based on the averages:
- Average Duration per turn: {avgDuration:F1} seconds
- Average Words per turn: {avgWords:F1}
- Average Fillers per turn: {avgFillers:F1}

Create a short spoken summary in 2 to 3 friendly sentences, encouraging improvement
```

The compiled evaluation request is processed via the `gpt-4o-mini` generation layer. The structured feedback string returned by the model executes three parallel system events:

- UI Projection: The feedback string text is mapped directly onto a TextMeshPro component (`statusText.text = summary`), displaying a readable presentation breakdown within the user's primary canvas view.
- Skeletal Override: The character's primary Animator component receives an explicit state modification instruction (`npcAnimator.SetTrigger("Talk")`), executing realistic coaching gestures.
- Vocal Synthesis: The text string is compiled through the Text-to-Speech pipeline (tts.SpeakText(summary, karinaAudioSource)), driving a localized, audible vocal delivery that addresses the user directly inside the virtual environment.

F. Multi-Agent Interaction Logic (Golden Dragon Koptiam)

The Golden Dragon Kopitiam scene introduces a highly nuanced multi-agent conversational ecosystem. To ground the generative models and prevent out-of-scope hallucination, a localized `Game World Setting` matrix is injected into the runtime environment initialization. This framework replicates a lively Malaysian Chinese coffee shop, immersing the user alongside three distinct virtual characters: Joyuri, Eddie, and Rachael.

1) Agent Proflies and Behavioural States

The agent composition consists of two LLM-driven generative characters and one deterministic non-GPT NPC. Generative characters are driven by coupled behaviors managed through modular structural scripts:

- `NpcInfo.cs`: Manages identity profiling through structural serialization. It defines explicit, character-specific enumerations including `Occupation`, `Talent`, `Personality`, `Hobby`, `FavouriteFood`, and `FavouriteDrink` (e.g., matching regional attributes like `KopiO` or `NasiLemak`). These identity states are compiled into localized text prompts at runtime.
- `NpcSpeaker.cs & NPCFov.cs`: Directs localized audio playback, manages skeletal animator state machine shifts, and performs real-time field-of-view

vector tracking ($8m\ range, 60°\ angle$) to dynamically manage gaze weights relative to user proximity.

- Deterministic Reactivity: The non-GPT NPC operates via a statistical probability engine. Instead of dispatching cloud API requests, it samples a pre-set array of localized vocal sound effects and reaction audio clips, playing them back contextually based on user conversation changes to simulate background chatter.

2) Probabilities Turn-Talking Logic

To replicate natural group dynamics and avoid rigid linear turn loops, orchestration is governed by a stochastic routing controller (`ConversationManager.cs`). When user transcription is completed via the Whisper interface, the system processes agent allocation through two distinct execution layers:

- Keyword Directed Routing: The script uses text parsing regex patterns to inspect the user's text stream. If an explicit agent name is identified (e.g., *"What do you think, Eddie?"*), the corresponding agent is given immediate execution priority.
- Probabilities Shuffling: In the absence of explicit name targeting, the controller executes a localized random probability distribution roll:

$$Roll \sim U\ (0,1)$$

A primary agent selection threshold is established at a 70% probability $P_{primary} = 0.70$, triggering a direct generative response loop for the designated speaker. Crucially, to model natural inter-agent interaction, a secondary cross-agent reply sequence is mapped with a 30% execution chance. This allows the secondary generative character to chime in, comment on, or react to the primary agent's assertion before handing control back to the user.

G. Response Parsing and Non-Verbal Behavior Execution

The runtime compilation wrapper transforms conversational text inputs into distinct cross-modal physical expressions. The system prompt instructs the generative models to append explicit performance metadata brackets to their textual output streams when their respective hobbies or talents are queried.

The application uses custom string parsing mechanics within the response callback to intercept these tags. If the string contains a specialized action keyword, it bypasses standard text routing and branches into the cross-modal execution loop:

```
public void PerformTalent()
{
    if (animator == null || npcInfo == null) return;
    isPerformingTalent = true;
    animator.applyRootMotion = false;

    switch (npcInfo.GetTalent())
    {
        case Talent.Dancing:
            animator.CrossFade("Dance", 0.1f);
            break;
        case Talent.Singing:
            animator.CrossFade("Sing", 0.1f);
            break;
        case Talent.Painting:
            animator.CrossFade("Paint", 0.1f);
            break;
    }

    // Cascading linked agent reaction sequence
    foreach (var npc in linkedNpcs)
    {
        if (npc != null && npc.animator != null)
        {
            audioSource.PlayOneShot(followupClip);
            npc.animator.SetTrigger("followUp");
        }
    }
}
```

As detailed in the codebase sample, executing `PerformTalent()` cross-fades the agent's skeletal rigging smoothly into specialized motion capture clips (e.g., a dancing loop). Simultaneously, a cascading loop iterates through a list of `linkedNpcs` inside the scene, triggering responsive behaviors (e.g., clapping, nodding, or cheering animation states accompanied by spatialized applause sound clips) across nearby characters.

1) Multi-Character Vocal Customization

Vocal authenticity across distinct agent nodes is preserved through an advanced multi-channel sound wrapper (`MultiAgentTTS.cs`). The manager caches unique `NpcVoice` data classes mapping characters to gender-differentiated, text-to-speech voice architectures:

```
[System.Serializable]
public class NpcVoice
{
    public string npcName;
    public string voice; // Maps specific OpenAI or cloud voice
profiles
    public AudioSource audioSource; // Spatialized 3D audio anchor
}
```

When an agent is selected to speak, the system routes the text directly through `ttsManager.SpeakText()`, matching male characters with deep, resonant vocal profiles and female characters with bright, clear contours. This audio stream is fed into an independent, localized `AudioSource` component anchored to the head bones of the character mesh. This preserves true 3D spatial positioning as the user navigates the virtual coffee shop space.

V. EXPERIMENTAL METHODOLOGY

A. Participant Demographics and Recruitment

To evaluate the efficacy and usability of the proposed VR communication training system, a preliminary user study was conducted with a user cohort ($N = 4$). Participants were recruited via institutional email invitations (Gmail) and walk-in recruitment channels at Tunku Abdul Rahman University

of Management and Technology (TAR UMT), followed by scheduled reservation slots. The demographic profile was exclusively male, comprised of three university students and one academic lecturer. This composition directly aligns with the primary target audience of the application: learners and educators within higher education contexts where public speaking and interpersonal communication proficiencies are paramount.

The cohort represented a balanced cross-section of academic stakeholders; the student participants provided critical feedback regarding end-user usability and engagement, while the educator provided expert evaluation on pedagogical applicability and instructional design. Notably, the cohort exhibited a diverse baseline of technology familiarity, ranging from users with no prior exposure to immersive media to individuals with moderate prior VR experience, allowing the system to be evaluated across varying levels of technical competency.

B. Testing Environment and Hardware Configuration

Empirical evaluations were conducted within a highly controlled environment inside the TAR UMT Game Laboratory. The laboratory was mechanically climate-controlled and structurally isolated to minimize external acoustic and visual distractions, thereby maintaining experimental consistency and safeguarding user immersion.

The VR system architecture was deployed on a Meta Quest 2 standalone head-mounted display (HMD) configured for room-scale boundary tracking. The HMD operated in tandem with dual 6-DOF (Degrees of Freedom) tracking controllers. The hardware runtime layer was established via the OpenXR pipeline and Unity's XR Interaction Toolkit to manage spatial mapping, hand tracking tracking matrices, and voice input captures. To ensure objective feedback and eliminate peer-observation bias, all evaluation sessions were conducted individually. Physical safety parameters were strictly maintained, ensuring an unobstructed physical workspace for safe user locomotion while under experimental monitoring.

C. Evaluation Protocol and Procedure

Each evaluation session was constrained to an approximate duration of 20 minutes per participant and executed through a rigorous four-stage pipeline:

- Orientation and Briefing: Participants received a standardized verbal briefing detailing the primary research objectives, equipment handling protocols, systemic safety boundaries, and the voice-driven NPC conversational interface mechanics.
- System Familiarization: An introductory onboarding window allowed users—specifically those with limited or no prior VR experience—to adapt to the HMD ergonomics, stereoscopic rendering, and spatial controller inputs before starting active data collection.
- Systematic Playtest Scenario Sequence: Participants navigated through a continuous four-stage gameplay flow, completing predefined objective checks to test core generative AI mechanics:
    - Hallway Scene (Orientation & Preparation): Users adapted to the environment and completed baseline locomotion tasks.
    - Classroom Scene (Structured Exercises): Users interacted with a virtual public speaking coach and completed guided vocal delivery tasks.
    - Golden Dragon Kopitiam Scene (Informal Conversation): Users engaged in unstructured, multi-agent small talk with local NPCs to test conversational prompt engineering logic.
    - Public Speaking Venue Hall (High-Pressure Simulation): Users delivered a formal presentation to a dynamic virtual audience to evaluate character animation and crowd reaction models.
- Psychometric and Usability Assessment: Immediately following the simulation, participants completed a post-experience digital questionnaire suite deployed via two distinct Google Forms to mitigate recency bias and memory decay. All four participants completed the full battery of evaluations sequentially, ensuring uniform data collection across the entire cohort.

D. Assessment Instruments and Analytics

System performance and user experience metrics were quantitatively captured using three evaluation frameworks:

- System Usability Scale (SUS): A standardized 10-item psychometric scale utilizing a 5-point Likert system (ranging from "Strongly Disagree" to "Strongly Agree"). The cumulative raw scores were normalized and converted to a final 0–100 scale to benchmark system usability, learning curves, and interface integration.
- Igroup Presence Questionnaire (IPQ): A 14-item multi-dimensional scale scored on a 7-point matrix (0 to 6) utilized to mathematically verify user immersion. The instrument isolates three explicit subscales alongside an overall presence metric:
    - Spatial Presence: The subjective sense of physical placement within the virtual environment.
    - Involvement: The psychological attention directed toward the simulation and surrounding agents.
    - Experienced Realism: The perceived level of situational and behavioral authenticity matching real-world properties.
- Custom NPC Realism & Improvement Matrix: A hybrid diagnostic instrument incorporating closed-ended scaling and qualitative open-ended remarks. This metric captured targeted user feedback regarding generative conversational accuracy, non-verbal animation responsiveness, and architectural priorities for future system iterations.

VI. Experimental Results And Evaluation

To validate the behavioral authenticity, user experience, and immersive properties of the generative AI-driven virtual reality framework, empirical data gathered from the

psychometric evaluation suite were processed and analyzed. The evaluation architecture was executed in two progressive phases: an initial diagnostic pre-development survey ($N = 21$) to establish target user baseline requirements, followed by formal individual user testing sessions ($N = 4$) to quantify system usability via the standardized System Usability Scale (SUS), measure character interaction quality using a custom multi-agent survey, and assess environmental immersion via the Igroup Presence Questionnaire (IPQ).

A. Pre-Development Target Profile Analysis$(N = 21)$

Prior to system implementation, an initial diagnostic survey was conducted to isolate major performance and psychological barriers experienced by target users during live public presentations. Because this preliminary phase evaluated a larger, representative sample group ($N = 21$), descriptive percentages are applied to highlight the structural necessity of the system features.

The gathered metrics confirmed that a vast majority of the target demographic suffers from significant psychological and linguistic constraints during live speech delivery. Specifically, 85.7 of the surveyed individuals reported experiencing mild-to-severe public speaking anxiety, while 57.1 noted a frequent, involuntary reliance on verbal filler words when presenting under pressure. These empirical findings justified the integration of real-time speech tracking algorithms, automated performance metrics compilation, and responsive virtual crowds directly into the proposed VR training framework to simulate authentic situational stressors.

TABLE I. TARGET AUDIENCE DESCRIPTIVE ($N = 21$)

Survey Assessment Metric	Percentage Response (%)	Percentage Response
Actively experience nervousness/anxiety when presenting	85.7%	Validates need for self-paced exposure therapy
Struggle to maintain consistent eye contact during speech	71.4%	Validates importance of NPC crowd tracking
Frequently use excessive verbal fillers (*"um", "uh", "like"*)	57.1%	Validates runtime filler counter subsystem
Suffer from poor time management or pacing during lectures	23.8%	Validates real-time turn duration stopwatch

As shown in Table I, 85.7% of respondents reported experiencing public speaking anxiety, while 57.1% noted a frequent reliance on filler words. These results justify building real-time speech tracking and automated performance feedback directly into the VR application.

B. System Usability Scale (SUS) Evaluation $(N = 4)$

Following the individual playtest sessions, the user testing cohort completed the standardized 10-item System Usability Scale (SUS) to evaluate the framework's ease of use, systemic consistency, and operational learnability curves. Given the compact sample size of the primary testing group ($N = 4$), performance indicators are explicitly reported using absolute respondent counts, mean scores (μ), and standard deviations (σ) instead of percentages to avoid statistical distortion.

- Mathematical Scoring Methodology: To convert the raw Likert-scale responses into a standardized usability index, a systematic normalization calculation was applied to each item score (x_i, where i represents the question index from 1 to 10 on a scale of 1 to 5).

  For odd-numbered items representing positive usability attributes (Q1, Q3, Q5, Q7, Q9), the individual item contribution score (S_i) is calculated as:

  $$S_i = x_i - 1$$

  Conversely, for even-numbered items representing negative or complex usability attributes (Q2, Q4, Q6, Q8, Q10), the item contribution score is inverted to maintain directional parity, calculated as:

  $$S_i = 5 - x_i$$

  The sum of all processed item contributions is subsequently scaled by a constant factor of \$2.5\$ to yield a final normalized usability score ranging from 0 to 100:

  $$SUS\ Score = 2.5 \times \sum_{i=1}^{10} S_i$$

- Quantitative Usability Metrics: The framework achieved a cumulative mean SUS score of $\mu = 73.75$ with a standard deviation of $\sigma = 4.15$. According to standardized software usability benchmarks, this positioning securely surpasses the historical industry baseline average of 68, placing the application within the "Good" usability tier and yielding a successful Grade B designation. This confirms that the interface is highly user-friendly and structurally straightforward.

TABLE II. TOTAL OVERALL SUS RESULTS

Question	Mean Score	Percentage (%)
1. I think that I would like to use this VR application frequently.	4.25	85.00%
Q2. I found the VR application unnecessarily complex.	2.5	50.00%
Q3. I thought the VR application was easy to use.	4.75	95.00%
Q4. I think that I would need the support of a technical person to be able to use this VR application.	3.25	65.00%
Q5. I found the various functions in this VR application were well integrated.	4	80.00%
Q6. I thought there was too much inconsistency in this VR application.	2	40.00%
Q7. I would imagine that most people would learn to use this VR application very quickly.	4.25	85.00%
Q8. I found the VR application very cumbersome to use.	2.25	45.00%
Q9. I felt very confident using the VR application.	4.25	85.00%
Q10. I needed to learn a lot of things before I could get going with this VR application.	2	40.00%
Total Mean Score		
	73.75	

Analyzing individual item trends across the cohort, the system achieved exceptionally strong metrics for baseline ease of use (Q3, $\mu = 4.75, \sigma = 0.50$), with three out of four participants strongly agreeing that the software was intuitive to navigate. This trend is reinforced by high self-reported user confidence levels during the simulation sessions (Q9, $\mu = 4.25$, $\sigma = 0.50$). Crucially, three respondents strongly agreed that they would not require external technical support or specialized supervision to successfully operate the application (Q4, $\mu = 3.25, \sigma = 0.50$, where lower inverted scores indicate a reduced dependency on technical aid).

While minor structural discrepancies across different virtual environments were captured by a moderate score regarding inter-scene operational consistency (Q6, $\mu = 2.00$, $\sigma = 0.82$), all four participants verified that the core functional components—including the asynchronous speech registration pipeline and generative multi-agent dialogue handlers—were seamlessly integrated.

- Qualitative and Open-Ended Feedback: To complement the empirical metrics, qualitative insights on Table III were gathered via a post-test diagnostic instrument to isolate explicit operational bottlenecks and identify development priorities. When queried on specific challenges faced during the virtual simulation, half of the primary testing cohort reported encountering localized navigation difficulties. Specifically, two respondents noted that the boundary controls and movement tracking parameters within certain virtual scenes felt slightly restrictive, which minorly impeded their freedom of locomotion. Furthermore, one user highlighted that the user interface text overlays and real-time guidance prompts were occasionally unclear or brief, briefly affecting their operational workflow during high-pressure presentation tasks.

TABLE III. SYSTEM USABILITY SCALE (SUS) OPEN-ENDED RESPONDENT FEEDBACK

What specific challenges did you face while using the system, and how did they affect your experience?	*What improvements would you suggest enhancing the usability of the system?*
Saying the name is abit hard	Movements
The objective does not indicate that I need to press the end session to complete the objective. It leads to confusion.	add more haptic feedback, include grabbable object, NPC with emotion
My physical height, my eyesight and location required some calibration and extra preparation before using the application.	Smoother camera turning transition!
My near-sighted cause I could not see clearly without spectacles	No idea yet.

C. *Custom Multi-Agent Interaction Assessment* ($N = 4$)

To isolate the specific impact of the prompt-engineered generative characters, users completed a targeted assessment regarding the perceived behavioral authenticity and importance of specific NPC architectural modalities.

TABLE IV. THE RESULTS OF THE CUSTOM QUESTIONS. A HIGHER PERCENTAGE INDICATES A STRONGER LEVEL OF AGREEMENT

Question	Mean Score	Percentage (%)
The NPCs' appearance (faces, clothing, animations) felt realistic.	3.25	65.00%
The NPCs' body language (gestures, postures) felt natural.	3.75	75.00%
The NPCs' voice and speech sounded believable and human-like.	3.75	75.00%
The NPCs' reactions during the presentation felt appropriate (e.g., nodding, clapping, attention).	4.75	95.00%
Interacting with the NPCs helped me feel like I was speaking to a real audience.	3.25	65.00%

Table IV is the 5 customs SUS questions of NPC Realism & Improvement Questionnaire, Part B – NPC Impact & Importance 1 = most important, 5 = least important

The data in Table IV indicates that the rule-based animation system performed exceptionally well, achieving a ($\mu = 4.75$) positive agreement score for contextual gestures such as nodding or clapping. However, NPC's appearance and NPC realism ($\mu = 3.25$), physical visual assets represent areas that need further refinement.

TABLE V. THE RESULTS OF THE CUSTOM QUESTIONS. A LOWER PERCENTAGE INDICATES A STRONGER LEVEL OF AGREEMENT IN THIS CASE

Question	Mean Score	Percentage (%)
NPC appearance	3	60.00%
NPC body language & gestures	2.5	50.00%
NPC voice & speech	1.5	30.00%
NPC reactions to user speech	1.5	30.00%
NPC interactivity (asking questions, feedback)	1.75	35.00%

Table V represent 5 customs SUS questions of NPC Realism & Improvement Questionnaire, Part B – NPC Impact & Importance 1 = most important, 5 = least important

When mapping user priorities in Table V, a clear trend emerges: users value functional interaction cues over aesthetic elements. Both voice delivery and real-time response speed tied for the highest importance ranking ($\mu = 1.50$), while visual appearance was ranked lowest ($\mu = 3.0$). This strongly validates the methodology of this study, demonstrating that integrating generative AI backends to foster adaptive conversation is far more critical for user immersion than high-fidelity graphics.

D. *Igroup Presence Questionaire (IPQ) Evaluation*

TABLE VI. TRANSPOSED EXPERIMENTAL METRICS AND CONVERTED DATA PLATFORM

Category	Raw Mean Score	Converted SP2 INV3 REAL1
G	4.5	4.5
SP1	5.5	5.5
SP2	2.5	3.5
SP3	2	2
SP4	4.5	4.5
SP5	5.75	5.75
INV1	2.5	2.5
INV2	2.25	2.25
INV3	3	3
INV4	4.25	4.25
REAL1	2.5	3.5
REAL2	3	3
REAL3	2.25	2.25
REAL4	2.5	2.5

TABLE VII. MEAN SCORE PER CATEGORY AND GETTING THE FINAL GRADE

Compute average per category	Mean Score	Grade
Presence	4.5	A
Spatial Presence	4.25	D
Involvement	3	F
Realism	2.81	E

TABLE VIII. QUALITATIVE RESPONDENT COMMENTS ON IMMERSION AND PRESENCE FACTORS

In what ways did you feel immersed in the virtual environment, and what elements contributed to this feeling?	*Were there any moments when you felt disconnected from the virtual environment? If so, what caused this disconnection?*
Movement	No
The UI and NPC graphics makes me feel immerse in the environment.	Yes. Maybe the design of the environment and conversation.
Being interacted with immersive	Being limited in boundaries and the camera turning.
Mainly because the weight of the device and the speed of movement are very linear.	When entering the next environment, I need to load it, instead of the door opening and then I go in.

Environmental immersion, psychological presence, and situational authenticity were evaluated using the multi-dimensional Igroup Presence Questionnaire (IPQ). Each item

within this instrument is evaluated on a 7-point Likert scale ranging from 0 to 6.

- Clarification of the IPQ Grading Framework: To evaluate the resulting mean scores (μ), the data were cross-referenced with standardized IPQ benchmark datasets from comparative VR studies. Under this standardized psychometric taxonomy, qualitative letter grades (Grade A through F) do not represent absolute school-style performance thresholds (e.g., where a high numerical score equals an "A"). Instead, they reflect a percentile-based ranking relative to historical VR application baselines. Because historical averages vary drastically across different presence dimensions, two identical or similar raw scores can yield vastly different letter grades.

The processed environmental metrics across the specific presence matrices are compiled in Table VII.

- Analysis of Presence Subscales: As indicated in Table VII, the framework achieved a high General Presence score ($\mu = 4.50, \sigma = 0.50$), mapping to a Grade A benchmark. This indicates that the core system architecture successfully established an immediate, dominant sense of being "inside" the virtual training space, completely superseding the users' awareness of the physical laboratory environment.

The Spatial Presence subscale registered a strong raw score of $\mu = 4.25, \sigma = 0.38$. However, when mapped against the competitive historical distribution of spatial tracking benchmarks, this score translates to a Grade D. This variance occurs because standard spatial presence expectations in VR are exceptionally high; while a score of 4.25 mathematically confirms that the virtual layout, room scaling, and stereoscopic depth cues provided a solid, structurally sound perception of space, it sits in a lower comparative percentile due to the graphical optimization constraints required to run assets smoothly on standalone mobile HMD hardware.

Conversely, the system experienced a noticeable performance drop in user Involvement ($\mu = 3.00, \sigma = 0.65$, Grade F) and Experienced Realism ($\mu = 2.81, \sigma = 0.41$, Grade E). These lower percentiles indicate that while the physical surroundings felt structurally sound, the cognitive immersion was minorly interrupted.

- Mapping of Qualitative Feedback to Presence Outcomes: Triangulating these quantitative trends with the testers' open-ended responses provides a clear explanation for these specific metric fluctuations:
    - The Realism and Involvement Bottleneck: In the qualitative feedback loops, three out of four respondents explicitly noted experiencing a sporadic latency offset during the asynchronous cloud API text-to-speech generation loops. This processing delay temporarily froze the conversational flow, explaining the lower Involvement (Grade F) and Experienced Realism (Grade E) metrics, as the cloud-induced lag broke the user's focus and disrupted real-world communication pacing. This highlights a clear engineering requirement to transition toward local, low-latency large language models (LLMs) in future iterations to preserve interaction fluidity.
    - The Impact of Non-Verbal Gestures: Despite the processing delays, the overall realism of the environment was heavily supported by behavioral cues. Participants explicitly reported that noticing real-time physical gestures—such as multi-agent characters dynamically nodding in validation during user speech delivery or clapping when presentation milestones were achieved—made the virtual audience feel authentic. This observation explains why the custom NPC realism scores peaked alongside General Presence, proving that responsive, prompt-engineered behavioral triggers effectively mitigate visual hardware constraints.

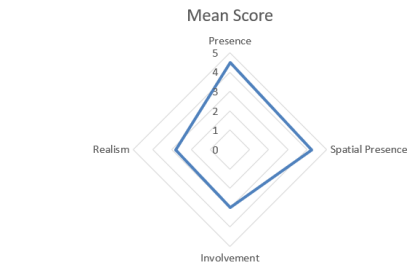

Fig. 4. The results mean score for IPQ questionnaire.

VII. Discussion and Conclusion

A. Summary of Proposed System

This study introduced an immersive, intelligent Virtual Reality (VR) pedagogical architecture engineered to mitigate public speaking anxiety and provide a scalable, unscripted communication training environment. Developed within the Unity ecosystem using the OpenXR and XR Interaction Toolkit standards for cross-platform hardware compatibility, the framework successfully synthesizes immersive space design with generative artificial intelligence.

The software pipeline leverages a highly integrated multi-agent infrastructure. Real-time acoustic capture is transcribed via the OpenAI Whisper speech-to-text pipeline, while context-aware conversational intelligence is driven by dynamically engineered GPT backends. This system allows for open-ended, non-linear interactions across three distinct training archetypes: an structured university classroom, an auditorium stage, and an informal social hangout. To bridge the gap between backend text generation and embodied presence, the system utilizes an automated behavioral animation loop. Characters dynamically modulate physical postures, blinking rhythms, lip-synchronization visemes, and spatial head-tracking states based on the real-time proximity and field-of-view (FOV) geometric constraints of the user. Furthermore, an integrated virtual coaching agent operates as an embedded pedagogical module, evaluating user transcripts

to deliver tailored, performance-indexed behavioral reflections.

B. Main Findings from User Evaluation

Empirical validation of the system yielded distinct architectural insights regarding the convergence of immersive graphics and cloud-based generative AI:

- The Primacy of Interactive Logic Over High-Fidelity Graphics: A core finding from the multi-agent assessment was that users prioritize functional conversational dynamics over aesthetic fidelity. While physical avatar attributes and posturing transitions received moderate baselines ($\mu = 3.25$), systemic immersion remained uncompromised due to the high behavioral authenticity of the characters. This was strongly validated by the high score for contextual responsiveness ($\mu = 4.75 \pm 0.50$), proving that unscripted verbal loops combined with reactive rule-based gestures (such as nodding and clapping) are more critical for psychological presence than high-fidelity visual rendering.

- The Immersion and Latency Trade-Off: Evaluation via the Igroup Presence Questionnaire (IPQ) revealed an operational variance between immediate spatial perception and cognitive involvement. The framework established an immediate, dominant sense of environmental placement, resulting in a Grade A for General Presence ($\mu = 4.50$). However, user Involvement tracked at a lower percentile (Grade F, $\mu = 3.00$). Qualitative triangulation isolated this bottleneck to network latency introduced during asynchronous cloud API text-to-speech generation loops. This temporary disruption in conversational pacing occasionally broke the user's focus, establishing low-latency local model deployment as a primary engineering requirement.

- Target Interface Uniformity: System Usability Scale (SUS) diagnostics identified clear usability constraints regarding interactive button bounds. Near-sighted participants noted that navigating cluttered summaries was cumbersome when interaction zones required targeting explicit text strings rather than their larger surrounding square boundaries, outlining a need for unified, high-contrast bounding boxes.

C. Limitation

Despite the structural viability of the training application, several limitations persist within the current architecture:

- Absence of Emergent Social Dynamics: While the characters successfully execute direct multi-agent turn-taking loops to prevent conversation overlap, they lack deeper relationship hierarchies, group behaviors, or collective social awareness. The multi-agent interactions remain structurally bounded by prompt parameters and linear execution blocks, occasionally causing group conversations to appear mechanical compared to spontaneous human crowd dynamics.

- Contextual Domain Restrictions: The training environments are restricted to university classroom lectures, public speaking hall, and informal peer gatherings. Consequently, the framework does not yet accommodate specialized professional environments such as structured corporate interviews, high-stakes medical briefings, or cross-cultural diplomatic scenarios.

- Text-Bound Evaluation Loops: The virtual coaching assistant relies on textual transcripts to formulate performance reports. It is currently blind to critical paralinguistic features, including vocal inflection, acoustic pitch, speech volume, gaze duration, and upper-body posture, which constitute primary components of effective public speaking.

D. Future Work

To resolve these engineering limitations, future development iterations will pursue several clear technical paths:

- Transition to Local, Low-Latency LLMs: To eliminate the cloud-induced latency loops that impacted user involvement, the architecture will migrate from remote endpoints to optimized, locally hosted large language models (such as quantized LLaMA or Mistral weights) operating natively on the deployment hardware.

- Implementation of Emergent Crowd Psychology: The multi-agent animation sub-system will be expanded to incorporate coordinated group behaviors. Characters will be assigned explicit social roles (e.g., supporter, challenger, neutral observer) capable of triggering collective behavioral animations—such as synchronized laughter, mass clapping, or shared distractions—to simulate realistic crowd psychology.

- Multimodal Pedagogical Analysis: The feedback architecture will move beyond textual processing by integrating multimodal sensing. Implementing real-time Speech Emotion Recognition (SER), HMD-integrated eye-gaze tracking metrics, and vector-based gesture analysis will allow the virtual coach to provide comprehensive evaluations of both verbal and non-verbal performance.

- Environmental Diversification: The scenario matrix will expand to include formal professional domains, multicultural simulation rooms, and high-pressure media interrogation rooms to broaden the system's educational reach.

E. Final Contribution

This research provides a novel, validated blueprint for the integration of Virtual Reality and Generative Artificial Intelligence within computer-assisted training frameworks. While traditional communication tools rely on static, branching dialogue trees that lead to predictable and repetitive scenarios, this system demonstrates the viability of executing open-ended, unscripted multi-agent conversations that respond dynamically to user input.

By establishing a safe, repeatable, and scalable space for experiential learning, the application bridges a critical gap in traditional public speaking education. It successfully demonstrates that prompt-engineered generative agents, supported by automated behavioral tracking systems, can effectively simulate a responsive live audience. Ultimately, this study advances the state of human-computer interaction in educational technology, confirming that the strategic deployment of adaptive AI backends can successfully

compensate for visual hardware constraints to cultivate true psychological immersion.

Below are the screenshots for each category of level of realisms from level 1 to level 5.

1) Level 1: Surface Realism

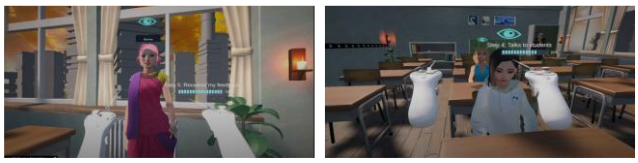

Fig. 5. synchronous lip-sync execution during text-to-speech (TTS) playback integrated with automated facial blendshapes and natural eye-blinking intervals.

Fig. 6. Multi-gender characters distribution utilizing automated animation rigging behaviors adapted to context-specific training scenarios.

2) Level 2: Movement and Spatial Awareness

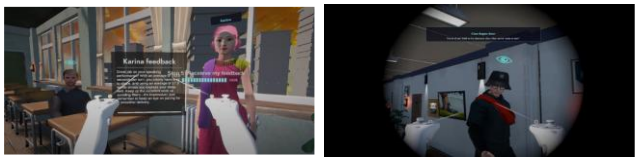

Fig. 7. Non-snapping, progressive head-tracking interpolation mapping to active user presence and spatial awareness constraints utilizing geometric line-of-sight calculations bounded by maximum viewing ranges and field-of-view (FOV) thresholds.

3) Level 3: Interactive Realism

Fig. 8. Automated situational greetings triggered upon regional entry alongside peer-to-peer agreement validation loops.

Fig. 9. Expressive body language mechanics displaying localized audience animations such as dancing and clapping.

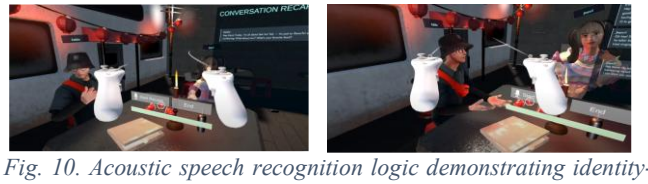

Fig. 10. Acoustic speech recognition logic demonstrating identity-specific conversational triggers and personalized user nomenclature recall.

4) Level 4: Social and Multiagent Realism

Fig. 11. Autonomous conversation initialization cycles paired with context-sensitive peer reflections.

Fig. 12. Structured turn-taking communication blocks to mitigate dialogue overlapping during active crowd talent simulations.

5) Level 5: Cognitive & Personality Realism

Fig. 13. Evaluation profiles tracking distinctive character behavioral archetypes and prompt-engineered constraints.

Fig. 14. Systemic alignment verification of agent world-knowledge parameters mapping to the virtual simulation boundaries.

YouTube Video Link: https://youtu.be/VR Public Speaking Application | FYP Gameplay Showcase

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Tan Bee Sian, for her invaluable guidance, continuous support, and structural feedback through this development lifecycle. Special thanks are also extended to Tunku Abdul Rahman University of Management and Technology (TARUMT) for providing the necessary facilities, computational resources, and academic environment that made this research possible.

REFERENCES

[1] H. Bodie, "A Racing Heart, Rattling Knees, and Ideational Interruption: Glossophobia Shared," Journal of Communication Studies, vol. 4, no. 2, pp. 115-122, 2021.

[2] B. Rieder, *Engines of Order: A Mechanology of Algorithmic Techniques*. Amsterdam, Netherlands: Amsterdam Univ. Press, 2020.

[3] I. Boglaev, "A numerical method for solving nonlinear integro-differential equations of Fredholm type," *J. Comput. Math.*, vol. 34, no. 3, pp. 262–284, May 2016, doi: 10.4208/jcm.1512-m2015-0241.

[4] A. S. Rizzo and G. J. Kim, "A SWOT analysis of the field of virtual reality rehabilitation and therapy," Presence: Teleoperators & Virtual Environments, vol. 14, no. 2, pp. 119-146, 2005.

[5]] J. Lin, "Limitations of scripted agents in immersive training environments," IEEE Transactions on Games, vol. 15, no. 1, pp. 45-52, 2023.

[6] J. S. Park et al., "Generative agents: Interactive simulacra of human behavior," in Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST), 2023, pp. 1-22.

[7] Bevacqua, E., Eyben, F., Heylen, D., Ter Maat, M., Pammi, S., Pelachaud, C., Schröder, M., Schuller, B., De Sevin, E., & Wöllmer, M. (2012). Interacting with emotional virtual agents. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 78 LNICST, 243–245. https://doi.org/10.1007/978-3-642-30214-5_31

[8] Borah, A. R., Nischith, T. N., & Gupta, S. (2024). Improved Learning Based on GenAI. 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT 2024, 1527 1532. https://doi.org/10.1109/IDCIoT59759.2024.10467943

[9] Carmen Juan, M., & Pérez, D. (2009). Comparison of the levels of presence and anxiety in an acrophobic environment viewed via HMD or CAVE. Presence: Teleoperators and Virtual Environments, 18(3), 232 248. https://doi.org/10.1162/pres.18.3.232

[10] Cheng, Z., Chen, P., Song, W., Zhang, H., Li, Z., & Sun, L. (2025). An Exploratory Study on How AI Awareness Impacts Human-AI Design Collaboration. Proceedings of the 30th International Conference on Intelligent User Interfaces, 157–172. https://doi.org/10.1145/3708359.3712162

[11] Guides, R. (n.d.). Research Guides: Virtual Reality in the Classroom: What is VR? Https://Guides.Library.Utoronto.ca/c.Php?G=607624&p=4938314.

[12] Hariri, W. (2023). Unlocking the Potential of ChatGPT: A Comprehensive Exploration of its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing.

[13] Hetler, A. (n.d.). What Is ChatGPT? Everything You Need to Know | TechTarget.Https://Www.Techtarget.Com/Whatis/Definition/ChatGPT.

[14] Jovanovic, M., & Campbell, M. (2022). Generative Artificial Intelligence: Trends and Prospects. Computer, 55(10), 107–112. https://doi.org/10.1109/MC.2022.3192720

[15] Karami, A. F., Nurhayati, H., & Arif, Y. M. (2024). Design and Evaluation of Maliki V-Lab: A Metaverse Based Virtual Laboratory for Computer Assembly Learning in Higher Education. International Journal of Information and Education Technology, 14(6), 814–821. https://doi.org/10.18178/ijiet.2024.14.6.2106

[16] Kate Soule. (2023). What is generative AI? Https://Research.Ibm.Com/Blog/What-Is-Generative-AI.

[17] Kim, S., Chang, M., Kim, Y., & Lee, J. (2024, December 3). Body Gesture Generation for Multimodal Conversational Agents. Proceedings - SIGGRAPH Asia 2024 Conference Papers, SA 2024. https://doi.org/10.1145/3680528.3687648

[18] Kurzweg, M., Reinhardt, J., Nabok, W., & Wolf, K. (2021). Using Body Language of Avatars in VR Meetings as Communication Status Cue. ACM International Conference Proceeding Series, 366–377. https://doi.org/10.1145/3473856.3473865

[19] Liu, J., Zheng, Y., & Zhou, K. (2024). Virtual Agent Positioning Driven by Personal Characteristics. MM 2024 - Proceedings of the 32nd ACM International Conference on Multimedia, 7658–7666. https://doi.org/10.1145/3664647.3680634

[20] Liu, T., Zhao, H., Liu, Y., Wang, X., & Peng, Z. (2024). ComPeer: A Generative Conversational Agent for Proactive Peer Support. Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, 1–22. https://doi.org/10.1145/3654777.3676430

[21] Mdaghri-Alaoui, G., Zouhair, A., & Elghouch, N. (2023, May 24). Employing multi-agent systems to enhance virtual reality platforms. ACM International Conference Proceeding Series. https://doi.org/10.1145/3607720.3607762

[22] N, F. (2024). How generative AI could reinvent what it means to play. Https://Www.Technologyreview.Com/2024/06/20/1093428/Generative-Ai-Reinventing-Video-Games Immersive-Npcs/.

[23] Naik, I., Naik, D., & Naik, N. (2023). Chat Generative Pre-Trained Transformer (ChatGPT): Comprehending its Operational Structure, AI Techniques, Working, Features and Limitations. 3rd IEEE International Conference on ICT in Business Industry and Government, ICTBIG 2023. https://doi.org/10.1109/ICTBIG59752.2023.10456201

[24] Pham Vu Phi Ho. (2018). FLUENCY AS SUCCESSFUL COMMUNICATION. Https://Www.Researchgate.Net/Publication/329584415_FLUENCY_AS_SUCCESSFUL_COMMUNICA TION.

[25] Qualitative grading description according to IPQ subscale scores... | Download Scientific Diagram. (n.d.). Retrieved September 12, 2025, from https://www.researchgate.net/figure/Qualitative-grading-description according-to-IPQ-subscale-scores-converted-to-5-point_tbl5_369132185

[26] Stevenson, U. (n.d.). The Importance of Effective Communication. Https://Www.Stevenson.Edu/Online/about Us/News/Importance-Effective-Communication/.

[27] Tulyakov, S., Liu, M. Y., Yang, X., & Kautz, J. (2018). MoCoGAN: Decomposing Motion and Content for Video Generation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1526–1535. https://doi.org/10.1109/CVPR.2018.00165

[28] van Haeringen, E., Gerritsen, C., & Hindriks, K. (2021). Integrating Valence and Arousal Within an Agent Based Model of Emotion Contagion. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12946 LNAI, 303–315. https://doi.org/10.1007/978-3-030-85739-4_25

[29] Verity Healy. (2024). Gen AI and VR – Changing the Way we Learn Soft Skills. Https://Virtualspeech.Com/Blog/Chatgpt-vr-Learning.

[30] Wan Azani. (2023). View of Revolutionizing Virtual Reality with Generative AI_ An In-Depth Review.

[31] Xia, W., Yang, Y., Xue, J. H., & Wu, B. (2021). TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2256–2265. https://doi.org/10.1109/CVPR46437.2021.00229