Player Voice Mimicry in Lethal Company

Allan Ma
New York University
New York, USA
allan.ma@nyu.edu

Andy Ng
New York University
New York, USA
an3299@nyu.edu

Matthew Ma
New York University
New York, USA
mm11762@nyu.edu

Matthew Maung
New York University
New York, USA
matthewmaung@nyu.edu

Ushan Abeysekera
New York University
New York, USA
ua2047@nyu.edu

***Abstract*—Lethal Company is a multiplayer horror game focused around trying to collect items to sell for money while also avoiding monsters and other obstacles with limited view and vision. Our proposal goes into the possibility of adding an additional feature for mobs, which samples and copies the voices of players and then uses Artificial Intelligence (AI) to alter the way a player interacts with the game (hopefully leading to their death). Based on the samples, the AI would generate new speech based on the sample data given that players might say based on context. The goal is to introduce a new element to the game that challenges players to consider what they can trust, while also focusing on developing a new avenue of how AI can be implemented into video games.**

I. Introduction

Lethal Company is a co-op survival horror game that focuses around players collecting items to sell for money in order to reach a target quota after a certain period of time. The players each represent an employee from the mysterious company, Lethal Company, where they are sent to abandoned moons and settings to collect valuable resources. The main objective is to collect as much value from the resources as possible without dying to the hostile creatures that lurk in the area. Throughout the entire game, communication is vital to ensure the survival of the players while being able to take calculated risks on the search for hidden treasures.

Our goal in this project is to use deep learning speech synthesis models to enhance the gameplay of Lethal Company by giving the monsters the ability to talk, adding emotions of confusion, interest, and scared to the players, making the game more complex, interesting, and deceptive. To do so, we have to gather game-related dialogue from YouTube video transcripts and accurate settings, entities, and details from the Lethal Company wiki data. Meanwhile, we will need to connect to an API that provides us with the Text-to-Speech system we need to generate human-like audio by performing voice cloning based on the audio inputs of players in the game. Using the metadata gathered on Lethal Company's environment, entities, and emotions, we will build prompts to input to our connected LLM, which then will output game-related texts to be generated as audio.

There are three main components we want to investigate as part of the implementation of this mod. The first is how successful was AI at replicating the voices of a human. Were they able to emulate screams, tonal differences, or at least fool a person into thinking someone else spoke? By looking into this aspect of our mod, we are able to further explore the extent of AI and how successful it is in copying someones voice, and additionally, how successful it is at emulate the nuances of speech such as tone and intonation. How easily was someone convinced that that was a real person?

The second thing we wanted to see was if players' strategies would change based on discovering the features of the mod, and how they would use it to their advantage or disadvantage. We're curious to see if players would try and find ways to maneuver around the feature, developing new strategies to identify themselves or differentiate themselves from the AI voices.

Lastly, we want to see how the player experience is enhanced with the mod. Does it make the game more enjoyable, and does interacting with the AI (through avoidance) add a new element to the game that is more interesting? Do players feel like the game is more refreshing by adding this new twist or at least, feel there are new challenges to be explored? What is the re-playability of this mod?

A. Voice Cloning Structure

To choose a valid voice cloning method, we need to ensure that our model consists of a speaker encoder, encoder, synthesizer, and vocoder (Fig. 1). The speaker encoder [1] is used to take input from audio samples and extract a speaker embedding that represents the speaker's voice, in our case, a player in the game. The speaker encoder consists of a few LSTM layers to efficiently obtain speaker characteristics in a time-dependent sequential audio file. Meanwhile, the encoder, synthesizer, and vocoder are all neural network components of the conditioned TTS model. In the conditioned TTS model, it first receives text in its encoder and extracts the text embeddings (feature transformation). Next, its synthesizer uses the Tacotron2 model [1] to generate a mel spectrogram that is conditioned on the text and speaker embeddings, which is then converted into a waveform in the vocoder, utilizing the WaveRNN [1]. Utilizing the voice cloning model to create a more entertaining mod of the game, we need to direct our focus to the emotional aspect of the generated audio. To produce audio that satisfies our three emotional states: interested, confused, and scared, we will have to adjust the intonation, speed, and volume of the generated audio clip conditioned to a specific emotion, text embedding, and speaker embedding.

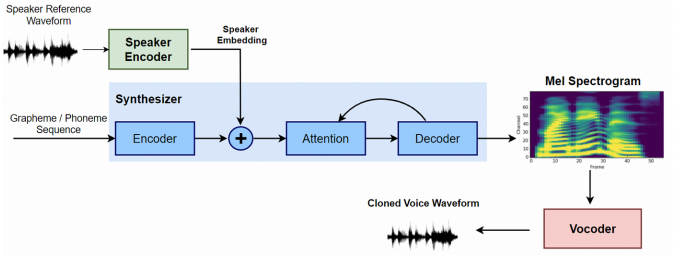

Fig. 1: *Voice Cloning Process*

II. Related Work

A. Existing Lethal Company Mods

The Lethal Company modding community has developed a wide range of player-created modifications, some of which explore ideas adjacent to mimicry and deception. Two of the most applicable are Skinwalkers by RugbugRedfern and Wendigos by TimShaw1. These mods provided inspiration and design benchmarks for our work, though none fully achieve the context-aware, near-real-time voice generation system we implemented.

Skinwalkers [2] replays short, verbatim recordings of players' voice chat through enemy entities. The mod captures live audio and triggers it randomly during gameplay, creating eerie coincidences. However, it does not use AI or synthesis, and the mimicry is purely reactive and thus is lacking emotional nuance, context awareness, and adaptive timing.

Wendigos [3] introduces AI-based voice cloning and uses locally trained models or ElevenLabs TTS API to generate speech in the player's voice. Phrases are predefined and categorized (eg, idle, chasing) and triggered by enemy behavior. While the mod includes experimental real-time features using Azure and ChatGPT, these require complex setup and are not deeply integrated with in-game events. Moreover, requiring players to pre-record voice samples removes the element of surprise.

In contrast, our system generates personalized, emotionally expressive lines in real time, grounded in the current game context and synthesized on demand—delivering a far more dynamic and immersive mimicry experience.

B. Emotional Speech Synthesis

Emotional Speech Synthesis (ESS) is a subfield of Text-to-Speech (TTS) that focuses on expressing the emotional aspect of speech. With the field of speech synthesis from human audio and text input growing, more emphasis is concentrated on the naturalness of the synthesized speech. One main factor that contributes to the realism and non-robot-like speech is its expressed emotion [4]. It uses a combination of natural language processing, deep learning, and signal processing to produce distinguishable emotions in generated speech. ESS can be used to alter some features of the audio to create different emotional states [4], such as:

- **Volume**: Loudness of the speech
- **Pitch**: Spectrum of tone or voice
- **Speed/Tempo**: How fast the speech is being said
- **Intonation Pattern**: how much the pitch varies

With prosody rules employed, global prosodic parameters are treated as near-universal cues for emotion [4]. In formant and time-dependent speech synthesis, prosody parameters are the heart of controlling emotional expressivity. There has been studies shown that successfully support the employment of prosody rules to express different emotions (Fig. 2). However, another step in improving the emotional expressiveness of the synthetic speech can be to consider the tempo or vowels and consonants, stressed and unstressed syllables, and the placement of pauses in dialogue [4].

Emotion Study Language Rec. Rate	**Parameter settings**
Joy [9] German 81% (1/9)	**F0 mean**: +50% **F0 range**: +100% **Tempo**: +30% **Voice Qu.**: modal or tense; "lip-spreading feature": F1 / F2 +10% **Other**: "wave pitch contour model": main stressed syllables are raised (+100%), syllables in between are lowered (-20%)
Sadness [4] American English 91% (1/6)	**F0 mean**: "0", reference line "-1", less final lowering "-5" **F0 range**: "-5", steeper accent shape "+6" **Tempo**: "-10", more fluent pauses "+5", hesitation pauses "+10" **Loudness**: "-5" **Voice Qu.**: breathiness "+10", brilliance "-9" **Other**: stress frequency "+1", precision of articulation "-5"
Anger [6] British English	**F0 mean**: +10 Hz **F0 range**: +9 s.t. **Tempo**: +30 wpm **Loudness**: +6 dB **Voice Qu.**: laryngealisation +78%; F4 frequency -175 Hz **Other**: increase pitch of stressed vowels (2ary: +10% of pitch range; 1ary: +20%; emphatic: +40%)

Fig. 2: *Alteration of Prosodic Parameters*

III. Methods

Using voice cloning with prosodic parameter alterations to generate human-like voice models in order to improve Lethal Company's game content and player experience, we've distributed our approach into several subparts.

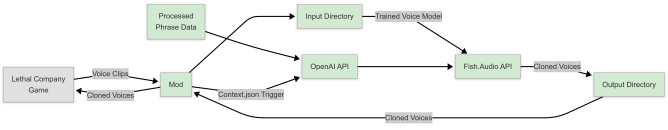

Fig. 3: *Voice Cloning Process*

A. Mod System Overview

As shown in Figure 3, the way the mod works is by first taking audio data from the game itself and storing it in a folder.

This is then turned into training data for the voice AI to create clones of for later. From here, the mod takes training data from YouTube let's plays as well as data regarding important aspects of the game such as the moons, monsters, and loot items, and uses it to generate realistic voice lines. These voice lines are then fed to the Fish Audio API to be cloned and spoken by one of the players who are currently in the game, and then fed to an output folder for the mod to intake and feed back to the game itself.

B. Voice Model Training

The voice model training and running is delegated entirely to the Fish Audio service. This solution was selected due to its relatively low cost, low technical learning curve, and response latency.

Once a multiplayer session starts up, the game mod will create a Dissonance Diagnostics folder [2], which is monitored by the voice model generation service. The mod will load in many segmented clips, usually between 1 and 5 seconds in length, containing short phrases spoken by the player. Once the combined length of the voice clips reaches 100 seconds, the the audio snippets are stitched together into a single file, sent via the API to Fish Audio where the voice model is created and hosted.

This process is performed on each player game instance. By ensuring that each player in the party uses the same Fish Audio key, each player hitting the 100 second threshold means that the voice models are created under the same parent account, and each player's mod can access the other players' voice models. Best practices indicated by Fish Audio specifies that creating a voice model should feature a single speaker. [5] Since the audio snippets are captured at the microphone level and not at the streaming audio level, each player's Dissonance Diagnostics folder contains just their audio.

C. Voice Cloning + Speech Synthesis

Since the Fish Audio connection is the same via an identical key for the player lobby, any text can be passed to any available model. The result is a game ready audio file, returned within a few seconds. After successful model creation and voice cloning expressing the anticipated emotion, we made sure to incorporate alteration of prosodic parameters to allow us to achieve the expression of emotions in our synthetic speech. Below is an example of the adjustments we made to the parameters, pitch, speed, and volume, using the Prosody schema from Fish Audio. Testing identified 3 useful parameter sets that could be used to simulate the intonations outlined later in III-D. Both the text phrases and the prosodic parameters were directed by the LLM as described in III-E.

```
{
  "panic": Prosody(pitch=1.7, speed=1.5,
    volume=1.3),
  "confusion": Prosody(pitch=1.3,
    speed=0.95, volume=1.0),
  "interest": Prosody(pitch=1.5,
    speed=1.1, volume=1.2)
}
```

D. Text Data Collection and Preprocessing

To ensure that our language model-generated voice lines reflected realistic and emotionally expressive in-game communication, we curated a custom dataset from two primary sources: the Lethal Company Wiki and YouTube transcripts from Lethal Company gameplay. The goal was to gather voice lines that felt natural within the game's environment, while aligning with our specific emotions: panic, confusion, and interest. We extracted gameplay transcripts from publicly available YouTube videos and passed them through a custom LLM-based classifier pipeline. The classifier was instructed to identify short, emotionally charged phrases that mimic how real players speak over voice chat. To help the model focus on relevant in-game situations, we provided grounding examples in the prompt—such as monster names (eg, Bracken, BaboonHawk), loot items (eg, gold bar, airhorn), and common scenarios (eg, being lost, hearing strange noises, discovering bodies). Each extracted phrase was required to be 10 words or fewer, emotionally distinct, and structurally varied to avoid repetitive or robotic-sounding output. The classification process categorized each extracted phrase into one of three emotional groups:

- Panic: Urgent or fear-based lines that might cause teammates to scatter or flee.
- Confusion: Disoriented or misleading lines that could result in miscommunication or hesitation.
- Interest: Curious or tempting lines designed to lure players toward loot or danger.

We supplemented this with static context from the Lethal Company Wiki, including monster behavior summaries and moon-specific loot tables, which later helped us enhance prompt realism and generate contextually grounded dialogue.

Finally, all extracted phrases were aggregated, deduplicated, and stored in a structured JSON format, which we used as the base pool from which we randomly sampled during gameplay.

E. Contextual Phrase Generation

To enhance the realism and effectiveness of the mimicry, our mod includes a real-time pipeline that generates short, emotionally expressive voice lines tailored to the player's current in-game context. The goal of this component is to enable the enemy monsters to dynamically "speak" in a way that resembles natural player communication, via reacting to environmental stimuli, referencing relevant items, and mimicking emotional tone, all through AI-generated voice lines. This system leverages pre-created emotional phrases, contextual cues from the environment, and an LLM to generate these believable bait dialogues. The final output is then synthesized using a voice cloning model and played back through the game's audio system. This section will explain each step in the process, from receiving the gameplay context to generating and selecting the emotionally expressive voice lines that are optimized for in-game delivery.

1) Context Input File: The contextual phrase generation system is triggered whenever a new .json file is detected in a designated directory during gameplay. These context files act as the primary input, and they encode the relevant game states needed to produce appropriate memory lines for the monsters.

Each context file includes several key fields that help guide the LLM prompt and the emotion in the voice line. An example of the typical context file is shown below:

```
{
  "moonName": "20 Adamance",
  "enemyName": "FlowerSnake",
  "key": "0adda51a-8c09-4cc4-b068-1f5ee",
  "distanceToPlayer": "far",
  "emotion": "interest"
}
```

As can be seen, the fields represent information such as the current map (moonName), the name of the enemy (enemyName) nearby that will be mimicking the voice line, and how close the enemy is to a target player (distanceToPlayer). The key (key) is a unique identifier for the mods to distinguish between the output files. The emotion (emotion) for the audio is chosen depending on multiple factors such as the enemy type and the distance from player. This structured input format helps allow each generated voice line to be tailored to the physical game and the psychological pacing of the situation.

2) Emotion-Based Phrase Sampling: Once a context file is parsed, the next step involves selecting a subset of voice lines that match the emotional tone of the gameplay situation. These lines come from the previously pre-processed file "`emotion_phrases.json`", which as previously mentioned, categorizes the phrases into panic, confusion, and interest. Based on the emotion that is randomly chosen, the script samples 15 phrases from the correspondent category. These sampled lines aren't spoken directly in-game, instead they act as candidate phrases for the LLM to personalize and rewrite based on the game context. By sampling and not using the whole category, we ensure that there is a manageable number of phrases past to the LLM, the prompt time, length, and token usage. Stylistically, these generated lines still remain realistic with voice chat communication. Furthermore, this helps ensure diversity in statements across different script runs.

3) Prompt Construction: Once a subsets of emotionally tagged phrases are selected, the script will construct a dynamic prompt which is then passed to the LLM. We carefully designed this prompts to guide the model in rewriting the raw phrases into short, expressive, and contextually relevant voice lines that can realistically be delivered using a low-quality (ie. cheaper) text-to-speech engine. Shown below is a compactly formatted version of our prompt.

LLM Prompt Template:

```
You are helping develop a horror mod for the game *Lethal
    Company*. In this game, an enemy NPC mimics real
    players by speaking voice lines over voice chat. These
    lines are played using a low-quality text-to-speech
    model**, so its absolutely critical that the phrasing
    sounds:
- **Short**
- **Natural**
- **Emotionally expressive**
- And most importantly: **like something a real player would
    say in voice chat.**

You are given:
- A list of candidate voice lines from the current emotional
    category (`{context["preferred_emotion"]}`)
- The current game context (player names, monster, moon loot
    , etc.)
---
### GAME CONTEXT
- Players in this session: {", ".join(context["player_names
    "])}
- Current moon: {context["current_moon"]}
- Enemy nearby: {context["enemy_name"]}
- Enemy behavior: {monster_description}
- Distance to target player: {context["distance_to_player"]}
- Loot found on this moon: {", ".join(moon_loot)}
---
### FORMAT RULES
- Keep each phrase **under 8 words**
- Use **casual spoken English**  contractions, slang, filler
    words
- Use **ALL CAPS** for yelling and **extra vowels** for
    drama (e.g., ruuuuun!, nooo way!)
---
### YOUR TASK
1. Select and rewrite voice lines from the list provided.
2. Make the lines **feel like real player speech**  panicked
    , curious, mocking, or urgent  depending on the
    situation.
3. Inject contextual hints using:
  - **Player names** from the session  like calling someone
      out or pretending to help.
  - **Monster behavior**  fake warnings like DONT RUN! It
     tracks sound! or Its the spider, stay above ground!
  - **Loot items**  suggest lures like Gold bar over here!,
      Who left a clown horn?, Air horn, grab it!
4. The NPC is trying to **trick, bait, or mislead** real
    players. Keep this in mind.
5. Voice lines must be **short, casual, emotionally
    expressive**, and optimized for **low-quality TTS** (
    like voice chat).
---
### FORMAT
Return your output as a flat list of voice lines, like this:
["line 1", "line 2", "line 3", ...]
---
### CANDIDATE PHRASES TO PERSONALIZE
{json.dumps(phrases, indent=2)}
```

We carefully pieced together each aspect of the prompt to achieve the best results out of our LLM. The main aspects of the prompt are as follows:

- **Role and Objective Definition:** The LLM is told it is assisting in developing a horror game mod. The NPC's role is to speak believable, short, emotionally expressive voice lines that mimic human players.
- **Stylistic Constraints:** Voice lines must be short (under 8 words), casual, and expressive. Instructions include formatting like ALL CAPS for yelling and elongated vowels for emotion (e.g., "noooo!", "ruuuun!"). These were added to compensate for low-quality TTS playback, which tends to flatten tone.
- **Dynamic Game Context:** Injects player names, moon name, monster identity, monster behavior, loot on the map, and distance to player. This helps the LLM personalize responses (e.g., referencing the correct moon or mimicking fear of a known monster). Monster behavior is pulled from a curated dictionary to add semantic grounding (e.g., "DON'T RUN, it hears footsteps!").

- **Task Instructions:** The model is explicitly told to: rewrite voice lines to match emotional tone and context, use deception, mimicry, or baiting strategies, and make it feel like casual voice chat, not scripted dialogue.
- **Output Formatting:** Specifies output should be returned as a flat list of strings for the script to parse. The candidate phrases (sampled earlier) are included inline.

By engineering our prompt like previously described, we can minimize the occurrences of hallucination and avoid overly verbose responses that can expose the memory. In addition to this, we help convey emotion via stylistic constraints.

To demonstrate the effect of personalization, the following figure shows select phrases before and after being passed to the LLM.

Non-Personalized Phrases	Personalized Phrases
Is anybody here?	Where are you guys? It's so quiet!
What is that?!	What's that noise?!
I was in a tizzy.	Guys, I saw something weird!
There's so much nothing here.	There's nothing in here, I swear!
What just happened? Did anyone see that?	What just happened?! Do you see that?!
What? I thought it was right here!	What? I thought the loot was here!
Wait, where are we going?	Matt, don't stick together! Spread out!
Why are you all clumped together?	Why are we all huddled up?
There was nothing downstairs, I think it's a dead end.	I'm telling you, it's a dead end!
How did he see that?	Hey, can you grab the wedding ring?
I thought this was a Minecraft world?	Allan, stop playing with that air horn!
How do I take the flashlight out?	Hey, Ushan, drop the gold bar!
What happens if we don't make quota?	It's creeping me out, stay close!
What is that?	Is that the BaboonHawk? RUN!
Wait, what's going on?	Guys, did you hear that?!
	Why do I hear wings flapping?!

TABLE I: *Example of raw and personalized voice lines. Each pre-LLM phrase was tagged with the emotion confusion and rewritten by a large language model (LLM) using the context: moon = Experimentation, enemy = BaboonHawk, player distance = far. Personalized lines reflect grounded, emotionally expressive voice chat suitable for deceptive NPC playback.*

As we can see, the original 15 sampled phrases reflect generic confusion, and they were designed in mind to be ambiguous enough for various situations. The language model successfully integrated key elements of the game state, referencing player names ("Allan," "Ushan" - these are manually entered by the players before the game), loot items found on the map (e.g., "gold bar," "wedding ring"), and even behavioral cues from the Baboon Hawk's description (e.g., "Why do I hear wings flapping?!"). This grounding enhances realism and also reinforced the mimicry mechanic, as lines sounded like believable teammate chatter while subtly misleading players. The rewritten phrases preserved the core emotion of confusion, while adding social pressure ("Matt, don't stick together!") and environmental misdirection ("It's a dead end!"). Overall, the personalized outputs demonstrate strong alignment with our design goals: emotionally expressive, context-aware, and optimized for deceptive, in-character delivery.

4) Voice Model Selection and TTS Generation: After a personalized voice line is selected from the list generated by the language model, the system processes it through the Fish Audio API, which allows for real-time text-to-speech generation using fine-tuned speaker voices. The script dynamically queries the authenticated Fish Audio account to retrieve a list of available voice models using the API, and this list represents the list of players in the game. From this, one of the available model names is randomly selected and this randomness it helps ensure that the memory feels less deterministic and also emphasizes the idea that the monster could potentially imitate anyone within the session. We hope that this further reinforces the core theme of mistrust and deception. The resultant .wav is generated by calling on the voice cloning scripts discussed in the previous section, and once saved, this audio file is picked up by the mod and played through the games existing voice channel system to simulate the live player speech.

5) Real-Time Execution: To support seamless integration with gameplay, the entire contextual phrase generation system is designed to operate continuously and respond to game events in real time. This is achieved through a single fib monitoring loop that watches for new .json files in a directory. Once a new context files is detected, the system initiates a full .wav file generation cycle. The monitoring loop runs every second and keeps track of previously seen files to prevent duplicate processing. Overall, this modular architecture allows for near real-time phrase generation, currently at around two seconds per phrase generation cycle.

IV. Experimental Setup and Evaluation Results

A. User Study Design

As part of the experiment, participants were invited to fill out a survey outlining their experience with Lethal Company. The survey started with questions before the player started playing asking questions such as how many hours they've put into Lethal Company as well as their proficiency in the game. The goal of asking these questions is to further investigate where their skill level is, and seeing if potentially there was a difference in how higher skilled players would navigate the situation.

Next, the participants would play Lethal Company up to the 1st quota, which equates to about three rounds of exploring moons, gathering loot and avoiding monsters. This was an opportunity for our test subjects to experience the mod in an organic manner, and see if there were any common deviations from how the game was played.

Lastly, participants would finish by completing an exit survey, which focused on ranking their experience on a scale of 1-10 with questions such as "The mod made the playing the game a creepier experience" and "I could easily discern which voice was the mod and which voice was from a human".

B. Evaluation Metrics

The evaluation metrics were focused around major objectives that we wanted to see reached as well as qualitative experiences by our play-testers. Major objectives that were focused upon were things such as "Would an AI voice be able to trick humans into believing it was real?", seeing how well an AI could convince humans that another human was

speaking. Additionally, we wanted to see how much the mod enhanced the game, asking questions regarding the level of "creepiness" that the addition of the mod included, as well as whether or not people enjoyed playing the mod as well.

V. Discussion

A. Summary and Interpretation of Feedback

To evaluate the Lethal Mimicry mod, we collected feedback from five participants with varying levels of experience in Lethal Company, ranging from casual to experienced players. Most participants had played between 10 to 50 hours of the base game, with one reporting under 10 hours.

Overall Experience Ratings varied from 2 to 10, with an average score of 6.6. One player gave a notably low score (2), citing audio distortion issues. On the other end, others praised the mod's creativity and effectiveness, calling it "creepy but super cool."

When asked if the mod made the game better/scarier, responses ranged from 3 to 10, averaging around 6.4. This indicates a generally positive impact on the horror atmosphere. Comments supported this, with one player stating the mod "wasn't a game changer... but definitely added tension."

The detectability of the voice cloning was more mixed. Scores ranged from 2 to 6, averaging about 4, suggesting some failure in mimicking player voices. Participants often recognized the voice as synthetic or "not really a player," reducing the effectiveness of the illusion for some.

Qualitative responses emphasized that the first encounter with voice cloning was eerie and novel, though repeated exposure may lessen its impact. Some players attempted to "ignore" cloned voices or used logical deduction to identify fakes, finding patterns in the AI.

In summary, the mod was well-received for its creep factor, though its technical execution (especially audio quality and voice believability) has room for refinement.

B. Strengths

Some of the goals that we achieved in our mod development were the successful implementation of the entire mod, and getting every point connected. Having a working version was something that seemed like a far stretch for us at the beginning so being able to complete it at the end was very rewarding. Additionally, we were able to get the LLM to extract the names of players active in the game while collecting voice data, meaning that there are opportunities for live-simulations to be done.

In regards to the game play itself, the mod was able to create in increased level of creepiness by adding newer sounds that players are unfamiliar with. However, it should be noted that there is a sense of novelty, and testing repeated exposure to these kinds of audio clips has not been tested so results could vary as the player gets adjusted to the voice mod. Players would also take these new audio snippets as something to be of interest, and continue to investigate the audio (sometimes leading to their death).

C. Design Limitations

Despite successes in developing a base-version of the mod, there are still a couple of limitations to what the mod can do. For example, one of the features we wanted to add was live conversation, where the AI would be able to extract context of the conversation that happens while the game was being played, and then be able to chime in based on this context. However, due to delays in sending files and the monsters querying for the audio, this was not possible.

Additionally, the distortion of the audio by Lethal Company makes it incredibly difficult to create audio that is able to convince others that it is a real human. Ideally, given more time, the mod would be able to reduce the amount of distortion that is happening or find alternative inputs, and then pass it through Lethal Company's filter later on.

In regards to the experiment, there are a few tweaks that would help improve the qualitative analysis of the experiment, starting with introducing a control group. Although it is already assumed that players who are play-testing our mod have already played the vanilla version of Lethal Company, seeing their performance would be an additional data point that could be further investigated. After adding the control group, additional questions to the survey would be added asking participants to measure the difficulty of Lethal Company, as well as the difficulty after play testing the mod itself.

Lastly, something we would want to continue to investigate is the prosody of the voice AI that we were trying to extract. One of the biggest challenges was how much we were able to incorporate tone or other elements of the human speech, and we also were not able to further investigate things such as accents. Having more emotions exhibited would be ideal for the further examining the limits of what voice AI can do.

VI. Conclusion

This project explored the integration of real-time, emotionally expressive voice mimicry into the gameplay of Lethal Company, using AI-driven voice cloning, contextual language modeling, and dynamic voice audio cloning. By enabling monsters to speak in the voices of players with deceptive intent, we introduced a novel mechanic that heightens immersion, tension, and psychological misdirection.

Our mod successfully demonstrated the feasibility of dynamic voice-based mimicry using cloned player voices and context-aware LLM-generated dialogue. User feedback confirmed that the feature increased the game's creepiness and engagement, though technical limitations such as voice believability and audio distortion remain key areas for refinement.

Looking ahead, improving audio realism, considering more prosodic parameters, reducing latency, and incorporating real-time conversational mimicry are promising directions. Expanding the range of emotions, accents, and situational awareness could further push the boundaries of AI-driven NPC behavior and establish new standards for reactive, emotionally intelligent game AI.

REFERENCES

[1] S. Kadam, A. Jikamade, P. Mattoo, and V. Hole, "Revoice: A neural network based voice cloning system," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, April 2024, pp. 1–6.

[2] R. Redfern, "Skinwalkers," https://thunderstore.io/c/lethal-company/p/RugbugRedfern/Skinwalkers/, accessed: 2025-05-12.

[3] Tim Shaw, "Wendigos Voice Cloning v1.0.9," https://thunderstore.io/c/lethal-company/p/Tim$_{S}haw/Wendigos_{V}oice_{C}loning/, thunderstore, 2024. Accessed : 2025 - 05 - 13.

[4] M. Schroder, "Emotional speech synthesis: A review," Master's thesis, Institue of Phonetics, University of the Saarland, Saarbrucken, Germany.

[5] F. Audio, "Voice clone best practices," https://docs.fish.audio/text-to-speech/voice-clone-best-practices, accessed: 2023-10-27.