Lightweight AI Voice-Interactive Dialogue Framework for Real-Time NPC Interaction in Games

Mohith R, Mohan Chandra S S, Nithish Gowda H N, Jeethu V Devasia

***Abstract*—Voice-controlled chatbots open up possibilities for developing immersive and interactive game experiences. This work presents a lightweight, offline, Transformer-based NPC chatbot trained on a domain-specific dataset of Skyrim dialogues. The system Integrates offline speech-to-text using Vosk with text-to-speech using Silero, allowing for real-time low-latency interaction without utilizing cloud services. The proposed approach generates coherent and lore-consistent NPC responses. only based on player speech, showcasing a practical pipeline for improving the level of immersion in open-world games. This study lays the foundation for future extensions toward more robust, Context-aware NPC dialogue systems. Index Terms-voice-activated chatbot, real-time speech interaction, Transformer architecture, offline inference, Skyrim dialogue dataset, low-latency NPC responses, texttto-speech, speech-to-text, game immersion.**

***Index Terms*—Voice-activated chatbot, real-time speech interaction, Transformer architecture, offline inference, Skyrim dialogue dataset, low-latency NPC responses, text-to-speech, speech-to-text, game immersion.**

I. Introduction

Gamelike speech interaction can achieve intuitive and immersive control without interrupting game flow. Existing voice Most gaming assistants rely on cloud APIs or are scripted. Dialogue trees, add latency, dependence upon internet poor connectivity, and static responses that reduce immersion.

This research focuses on a fully offline, lightweight Transformer-based chatbot trained on a custom Skyrim dialogue dataset (Player - NPC exchanges). The system integrates:

- **Offline Speech-to-Text (STT):** Converts player speech to text using the Vosk engine.
- **Transformer-based Chatbot:** Generates NPC responses in a coherent, lore-consistent manner.
- **Offline Text-to-Speech (TTS):** Uses Silero to convert responses into natural-sounding audio.

By combining these components, the system achieves **real-time, low-latency voice interaction** suitable for open-world gaming environments. While the current scope focuses on training with a fixed dataset, the architecture provides a foundation for future work in context-aware and personality-driven NPC dialogues.

II. Related Work

Recent advances in speech recognition and spoken language understanding (SLU) have demonstrated significant progress through the integration of convolutional and transformer-based architectures, especially for streaming and real-time applications.

A. Streaming Transformer ASR

Conformer (Gulati et al., 2020, Interspeech) – Combines convolution and self-attention for improved ASR accuracy, tested on LibriSpeech. Strengths: Strong local and global feature modeling. Limitations: Not designed for streaming by default.

Emformer (Shi et al., 2021, ICASSP) – Memory-augmented Transformer enabling streaming ASR with low latency. Strengths: Efficient for real-time streaming applications. Limitations: Complex implementation and training requirements.

Zipformer (Yao et al., 2025, ACL Industry Track) – Memory-efficient Transformer unifying streaming and non-streaming ASR modes. Strengths: Flexible for different use cases with low memory footprint. Limitations: Tooling and community adoption are still emerging.

B. SLU and End-to-End Models

WhiSLU (Bai et al., 2023, Interspeech) – End-to-end spoken language understanding system based on Whisper with multi-task learning; evaluated on SLURP dataset. Robust pretrained representations improve noise tolerance. Model size is heavy for edge or low-resource devices.

CMSST (Wang et al., 2024, EACL) – Cross-modal selective self-training to enable zero-shot SLU, improving entity and intent recognition F1 scores. Strengths: Label-efficient learning reduces annotation needs. Limitations: Pipeline complexity may limit real-time deployment.

GMA-SLU (Li et al., 2024, IJCAI) – Generates synthetic audio from labels to enhance textless SLU performance. Strengths: Reduces dependency on costly labeled data. Limitations: Possible domain mismatch leading to performance degradation.

C. Context-Aware Game Speech Interaction

Context-Aware ASR for Games (Kumar et al., 2024, ACM Multimedia) – Injects game-state vocabulary into ASR for recognition biasing. Strengths: Improves accuracy on game-specific commands. Limitations: Requires integration with game state APIs.

Voice-Controlled NPC Dialogue (Lee et al., 2025, IWSDS) – Maps speech input to dialogue nodes via LLMs for NPC interactions. Strengths: Enables natural, context-aware interaction. Limitations: Dependent on ASR accuracy and latency.

Exploratory NPC Speech Studies (Brown et al., 2024, Taylor & Francis) – User studies analyzing player immersion and latency thresholds. Strengths: Provides actionable UX benchmarks. Limitations: Does not propose or evaluate technical models.

III. Identified Gaps and Proposed Research Direction

An examination of the recent research on automatic speech recognition. The vocal-based game interaction, (ASR), spoken language understanding (SLU) demonstrates a number of challenges that are yet to be solved essential to live action, virtual gameplay.

A. Identified Gaps

1) Limited Streaming-Optimized Architectures:

- While state-of-the-art low-latency speech recognition widely adopts architectures such as the Conformer, it is not naturally streaming capable and requires modifications to be used in real time.
- Most existing streaming-capable Transformers involve Complex memory management or training pipelines, which are difficult to integrate into resource-constrained gaming environments.

2) Model Size and Deployment Constraints:

- Large pre-trained SLU models, such as WhiSLU, are powerful robust, while entailing high computational costs, making on-device inference infeasible for consoles, VR headsets, or mobile devices.
- Lightweight alternatives remain underexplored, especially for edge deployment without notable accuracy

3) Domain-Specific Adaptation Limitations:

- Context-aware approaches for gaming, for instance, Kumar et et al., 2024) improve recognition of particular commands but require tight coupling with game state APIs.
- Few architectures generalize across different kinds of games. genres, resilient to noisy, jargon-heavy in-game Speech environments.

4) Latency and Immersion Trade-Offs:

- Brown et al. (2024) stress stringent latency thresholds (¡ 200 ms) for maintaining immersion, yet many existing Models are designed to favor accuracy over latency.
- Few empirical studies combine architecture optimization with latency-focused benchmarks in gaming contexts.

B. Proposed Research Direction

In order to address the limitations of existing in-game NPC dialogue systems, we present a lightweight transformer-based architecture that is specifically optimized for offline, real-time, game-specific speech interaction.

System Overview

1) **Speech-to-Text (STT)** – The player's spoken input is converted into text using the Vosk engine. This module operates completely on-device and is optimized for low-latency performance, ensuring fluid interaction without requiring an internet connection.
2) **Chatbot NLP Engine** – A Transformer-based language model trained on a domain-specific Skyrim dialogue dataset is used to generate NPC responses. The model produces outputs that are contextually coherent and stylistically consistent with the game's lore and character tone.
3) **Text-to-Speech (TTS)** – Generated responses are synthesized into natural-sounding audio through the Silero TTS engine. This allows NPCs to speak dynamically generated dialogue in real time, enhancing the sense of presence.
4) **Audio Playback in Game** – The synthesized audio is delivered directly to the game environment, providing continuous real-time feedback and creating an immersive conversational experience.

Core Model Features

- **Transformer-based Dialogue Generation:** Captures sequential input patterns effectively, enabling the model to produce responses that align with conversational context and in-game narrative style.
- **Offline Inference Pipeline:** Operates as a fully local system, removing reliance on external cloud services and providing stable performance in both offline and low-resource scenarios.
- **Domain-Specific Training:** Uses a curated Skyrim dialogue dataset to ensure character behavior, mannerisms, and speech patterns remain faithful to the game's role-playing world.
- **Lightweight and Low-Latency:** Engineered for rapid computation and optimized to run efficiently on standard hardware, enabling seamless real-time dialogue.

C. Expected Impact

This system demonstrates a practical method for enabling dynamic, voice-driven NPC interactions within the limitations of real-time game environments. By combining offline speech processing, efficient dialogue generation, and integrated audio response, it supports immersive conversational gameplay without cloud dependencies or extensive preprocessing pipelines. This work also establishes a platform for future advancements in areas such as long-term context retention, emotional tone modeling, and adaptive NPC personality behavior.

IV. Problem Statement

Despite advancements in speech recognition and natural language generation, non-playable characters (NPCs) in games continue to rely primarily on static, pre-scripted dialogue interactions. Such limitations reduce player immersion and restrict the NPCs' ability to respond dynamically within the game environment. Many existing voice-based game interactions depend on cloud services (e.g., Alexa Skills, Google Speech APIs, or other cloud-based NLP endpoints), which introduce latency, require stable internet connectivity, and do not effectively capture game-specific lore or role-play nuances.

Furthermore, most conventional chatbot models are not optimized for on-device execution, where low latency and lightweight inference are critical for maintaining real-time interaction. Traditional chatbots also lack domain-specific grounding, making them unsuitable for generating responses that reflect character personality, lore consistency, and immersive behavior expected in role-playing contexts.

Research Focus: This work addresses the problem of designing a lightweight, fully offline NPC dialogue system capable of generating coherent, context-aware responses in real time. The model is trained on a custom Skyrim dialogue dataset consisting of 7,565 NPC–player exchanges, enabling the system to emulate the linguistic tone and conversational patterns present in the game's role-play environment.

Objectives:

- **Offline Response Generation:** Produce valid NPC responses solely based on player input without reliance on cloud services.
- **Low-Latency Interaction:** Ensure rapid processing to preserve real-time gameplay responsiveness.
- **Extendable Framework:** Establish a foundation for future development of more advanced, context-aware, and personality-driven NPC dialogue systems.

By addressing these objectives, this work contributes to improving immersive interaction in open-world role-playing games. The system provides a pathway for more adaptive NPC behavior and represents a step toward richer, real-time conversational gameplay experiences.

V. Methodology

The proposed system is designed as an **offline, speech-driven NPC dialogue agent** consisting of three sequential modules: a Speech-to-Text (STT) processor, a Transformer-based dialogue generation model, and a Text-to-Speech (TTS) synthesizer. These modules operate locally, ensuring **low-latency real-time interaction** and eliminating dependence on external cloud APIs, which enhances both responsiveness and privacy during gameplay.

A. Dataset Collection and Preprocessing

The model was trained using a **custom synthetic Skyrim-style dialogue dataset** containing **7,565 conversational pairs**, where each pair represents a short exchange between a player and a non-playable character (NPC).

1) Dataset Creation:

- **Source Material:** The dataset was constructed using *official Skyrim lore*, including quest scripts, dialogue archives, and character backstories. This ensured that generated samples maintained the tone, linguistic style, and thematic consistency of the game.
- **LLM-Based Dialogue Generation:** Dialogue pairs were created by prompting a large language model (LLM) with structured instructions. Each prompt specified:
  - The assumed player scenario or intent.
  - A natural player utterance.
  - A corresponding NPC response that adhered to the character's personality and the game's lore.
- **Quality Review:** Every generated dialogue was manually inspected. Samples that contained incomplete sentences, incoherent phrasing, or that deviated from canonical Skyrim content were removed to ensure narrative authenticity.
- **Variation Expansion:** Selected dialogues were paraphrased and reformulated to introduce linguistic diversity while preserving semantic and stylistic fidelity to the source material.

2) Data Preprocessing:

- **Text Normalization:** All text was converted to lowercase, and punctuation, annotations, and non-verbal cues were removed for uniformity.
- **Tokenization:** Each word in the dataset was assigned a unique integer ID through a custom tokenizer, ensuring consistent text-to-token mappings.
- **Sequence Markers:** Special tokens `startseq` and `endseq` were added to delineate the beginning and end of each model-generated response.
- **Padding and Length Benchmark:** Dialogue sequences were padded or truncated to a fixed maximum length corresponding to the **90th percentile** of all sequence lengths. This approach balanced computational efficiency with context preservation.

- **Vocabulary Construction:** Separate vocabularies were built for **player inputs** and **NPC outputs** to allow the model to better capture dialogue-specific terms, colloquial expressions, and recurring narrative patterns.

B. Custom Transformer Model Architecture

The dialogue generation model employs a **sequence-to-sequence Transformer** architecture, chosen for its strong ability to model long-range dependencies and contextual coherence in dialogue systems.

1) Embedding: Input tokens are converted into dense vector representations using a shared embedding layer, allowing the model to learn semantic similarities and relationships among words.

2) Positional Encoding: Since Transformers process all tokens simultaneously, positional encodings are added to embeddings to preserve the order of words within a sentence, ensuring sequential awareness during training and inference.

3) Encoder Structure: Each encoder layer applies:

- **Multi-head self-attention** to relate tokens across the input sequence.
- A **two-layer feed-forward network** for feature transformation.
- **Residual connections**, **layer normalization**, and **dropout** to enhance gradient flow and training stability.

4) Decoder Structure: The decoder generates responses autoregressively, using:

- **Masked self-attention** to ensure each token prediction depends only on previously generated tokens.
- **Cross-attention** to incorporate contextual meaning from the encoder outputs.
- A **feed-forward sublayer** with residual normalization for stability and consistency in output generation.

5) Output Projection: The decoder output is projected through a fully connected layer, converting hidden states into a probability distribution over the output vocabulary, enabling next-token prediction during inference.

6) Training Procedure:

- The model was trained using **masked sparse categorical cross-entropy**, which excludes padding tokens from loss computation.
- Optimization utilized the **AdamW optimizer** with **learning rate warm-up** and **cosine decay** scheduling for stable convergence.
- Training was performed for **50 epochs** on a **Google Colab T4 GPU**, completing in approximately **five minutes**.
- **Mixed-precision computation** was enabled to reduce memory usage and accelerate training.
- Model performance was monitored via validation loss and manual inspection of generated dialogue samples to assess coherence and fidelity.

CUSTOM TRANSFORMER MODEL ARCHITECTURE: SKYRIM_TRANSFORMER

Layer (type)	Output Shape	Param #	Connected to
decoder_input (InputLayer)	(None, None)	0	-
encoder_input (InputLayer)	(None, None)	0	-
shared_embedding (Embedding)	(None, None, 128)	655,360	encoder_input, decoder_input
pos_enc (PositionalEncoding)	(None, None, 128)	8,192	shared_embedding
dropout (Dropout)	(None, None, 128)	0	pos_enc
layer_norm1 (LayerNormalization)	(None, None, 128)	256	dropout
multi_head_attention1 (MultiHeadAttention)	(None, None, 128)	66,048	layer_norm1
add1 (Add)	(None, None, 128)	0	dropout, multi_head_attention1
layer_norm2 (LayerNormalization)	(None, None, 128)	256	add1
dense1 (Dense)	(None, None, 256)	33,024	layer_norm2
dropout2 (Dropout)	(None, None, 256)	0	dense1
dense2 (Dense)	(None, None, 128)	32,896	dropout2
add2 (Add)	(None, None, 128)	0	add1, dense2
layer_norm3 (LayerNormalization)	(None, None, 128)	256	add2
decoder_block_0 (DecoderBlock)	(None, None, 128)	198,784	layer_norm3, add2
decoder_block_1 (DecoderBlock)	(None, None, 128)	198,784	decoder_block_0
decoder_block_2 (DecoderBlock)	(None, None, 128)	198,784	decoder_block_1
decoder_block_3 (DecoderBlock)	(None, None, 128)	198,784	decoder_block_2
logits_dense (Dense)	(None, None, 5120)	660,480	decoder_block_3

Total params: 2,657,280 (10.14 MB)
Trainable params: 2,657,280 (10.14 MB)
Non-trainable params: 0 (0.00 B)

Fig. 1. Overview of the Transformer-based dialogue generation model

C. Comparative Fine-Tuning with GPT-Neo

To objectively evaluate the performance of the custom Transformer model, a comparative fine-tuning study was conducted using the EleutherAI GPT-Neo 125M model. The same Skyrim dialogue dataset, preprocessing steps, and evaluation procedure were used to ensure a controlled and consistent comparison. The objective of this study was to examine the generative fluency and contextual coherence of a larger pre-trained language model (LLM) when adapted to the same conversational domain.

1) Model and Data Preparation: Table I summarizes the dataset preparation and training configuration used during GPT-Neo fine-tuning, which closely mirrors that of the custom Transformer model.

TABLE I
GPT-NEO FINE-TUNING DATA PREPARATION SUMMARY

Aspect	Specification	Description
Base Model	GPT-Neo (125M)	Pre-trained causal LM for benchmarking.
Dataset	`skyrim.csv`	Same dataset as custom Transformer.
Preprocessing	Tokenization, lowercasing, padding	Identical normalization pipeline.
Split Ratio	90% / 10%	Consistent evaluation framework.
Sequence Format	Input–Response pairs	Maintains turn-based alignment.

2) Training Configuration and Procedure: Fine-tuning was carried out using the HuggingFace `Trainer` API under a Causal Language Modeling (CLM) objective. Hyperparameter settings were matched to the custom Transformer to isolate differences arising purely from model architecture.

- **Epochs:** 3 (approx. training time: 20 minutes)
- **Batch Size:** 2 per device
- **Optimizer:** AdamW, learning rate 5×10^{-5}, 500 warm-up steps
- **Precision:** FP16 mixed-precision for improved computational efficiency
- **Checkpointing:** Lowest validation loss checkpoint retained

3) Inference and Observations: Inference followed the same pipeline used for the custom Transformer to allow direct comparison of generated responses.

Prompt Template:

```
Input: {user query}
Response:
```

Generation Parameters:

- Sampling enabled (`do_sample=True`)
- Temperature = 0.7, Top-k = 50, Top-p = 0.95

The fine-tuned GPT-Neo produced fluent and coherent responses; however, inference latency was noticeably higher when compared with the lighter custom Transformer model. This illustrates that while large pre-trained models offer stronger expressive capabilities, the custom Transformer provides superior efficiency and responsiveness—qualities that are critical for real-time NPC dialogue systems.

D. Integration with STT and TTS

To enable fully voice-based interactions:

- **STT (Vosk):**This STT model is able to transcript the player's speech to text in real-time and it is lightweight and optimized for low-latency inference.
- **TTS (Silero):** This TTS mode generates responses into readily understandable speech to provide immersive NPC voice output.

E. End-to-End Inference Pipeline

The inference pipeline is designed to showcase an implementation of the custom Transformer model for real-time NPC dialogue generation. The process flows as follows:

- **Voice Input Capture:**The player's speech is captured via microphone input.
- **Speech-to-Text (STT) Conversion:** The input audio is converted to text via a pretrained offline STT system (e.g. Vosk or Whisper).
- **NPC Response Generation:** The custom Transformer model takes the transcribed audio input text, and it predicts an on-topic NPC response.
- **Text-to-Speech (TTS) Conversion:** The output text is converted to audio via a pre-trained TTS system (e.g. Silero TTS) for immediate audio playback.

The modular nature allows for quick experimentation of various STT and TTS backends without needing to modify the Transformer model which maintains a focused direction of the dialogue generator. The pipeline is optimized for low-latency inference and can be adapted for full automaton voice interaction in a game context.

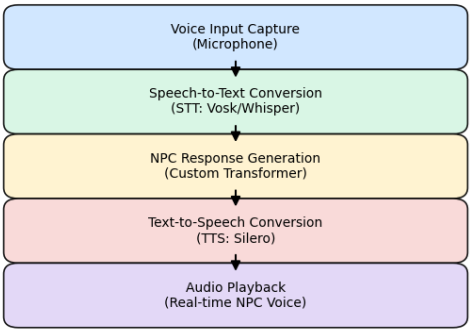

Fig. 2. End-to-end voice-based Transformer chatbot pipeline

This approach guarantees that the system provides responses that are not only syntactically correct but that also maintain semantic relevance and awareness of context, which are important for interactive dialogue in gaming environments.

VI. Results and Insights

A. Quantitative Evaluation

To check the model performance, the metrics were calculated for both the custom Transformer which is of 30mb in size and fine-tuned GPT-Neo model of 125M. The metrics capture both lexical similarity and semantic-level understanding, giving a better view of dialogue quality.(Table II).

TABLE II
Comparison of Model Evaluation Metrics

Metric	Custom (2.67M)	GPT-Neo (125M)	Interpretation
BLEU	0.1786	0.0589	Phrase-level n-gram overlap; higher indicates better lexical match.
ROUGE-L	0.5396	0.2010	Measures longest common subsequence; reflects structural coherence.
METEOR	0.4249	0.2179	Captures semantic and synonym-based similarity between responses.
BERTScore-F1	0.9040	–	Evaluates meaning-level correspondence using contextual embeddings.

Despite being smaller in size of 2.67 M, the custom Transformer have higher scores across all metrics specified through the code. The BLEU and ROUGE-L values indicates a very good lexical and structural alignment with reference to responses, while the METEOR score reflects a stronger semantic understanding and contextual relevance

In according to the fine-tuned GPT-Neo (125M) this is pre-trained on vast general-domain text showed that there is lower adaptation to the domain-specific to Skyrim dialogue dataset. The reduced BLEU and ROUGE-L scores tells that phrase-level and structural retention, while the drop in METEOR underscores weaker context modeling.

These highlights that a compact, domain-trained model can outperform a larger pretrained language model when the

dataset is specialized and constrained. The high BERTScore-F1 further validates the custom model's ability to preserve narrative tone and aligned with Skyrim's in-game conversational context.

B. Qualitative Results

The dialogues generated were evaluated based on coherence, tone, and relevance. The chatbot successfully in adapting its way according to the Skyrim specific character and produced its responses in a behavioral patterns.

TABLE III
SAMPLE INPUT–OUTPUT DIALOGUE PAIRS

Player Input	NPC Response (Model Output)
How are you	i'm well thank you and i assist
Home	wish i had one
Best place to train	find a master learn from them
Cure for the plague?	the cure is in footprints near you need
Dragons	keepers of time or its destroyers

C. Explainable AI (XAI) Analysis

To understand the model's behavior,a XAI analysis was done:

- **Token-level Probability Inspection:** Examined the decoder's token-level output probabilities to identify which tokens the model favors at each step, giving insight into its decision-making process.
- **Attention Visualization:** The token-level attention weights which highlighted the model's contextual focus on the input tokens.

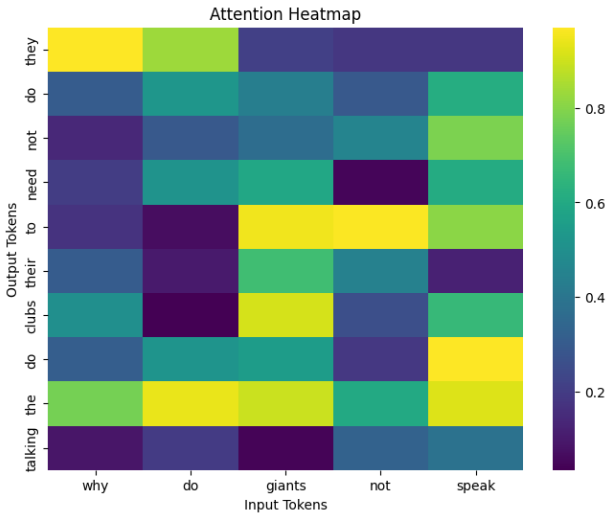

Fig. 3. Attention heatmap showcases the key token influence. In this image the darker colors indicates higher attention weights which highlights the model's focus on important input tokens for generating responses.

D. Real-Time Evaluation

The prototype provided real-time performance and computational efficiency. Even though three deep learning components Speech-to-Text (STT), the Transformer-based chatbot, and Text-to-Speech (TTS) were integrated the overall system had a highly lightweight and a faster responsive

TABLE IV
MODULE SIZE AND REAL-TIME LATENCY OVERVIEW

Module	Model Size	Average Latency (ms)
Speech-to-Text (Vosk)	≈ 67 MB	85.3
Chatbot (Custom Transformer)	30.9 MB	245.1
Text-to-Speech (Silero)	≈ 30 MB	1560.8
End-to-End Total	**<130 MB (All Models)**	**≈1891.2**

The results tells that the full voice-based dialogue system operates perfectly even on modest hardware resources:

- **Low memory footprint:** the total model is of 130 MB which is significantly smaller compared to most commercial dialogue systems based on other LLMs.
- **Real-time response:** STT and chatbot modules maintain, collectively Sub-300 ms latency to ensure fluent conversations.
- **TTS bottleneck:** Most of the latency comes from Silero TTS.playback duration rather than synthesis computation, which still allows natural, human-like response pacing.

The observations confirms that the architecture is not only contextually coherent but also highly efficient for real-time, with ondevice deployment, a light-weight, and a responsive NPC dialogue Framework.

VII. CONCLUSION AND FUTURE SCOPE

This research presents the development and evaluation of a real-time, offline Transformer-based voice-enabled NPC chatbot trained on a Skyrim-specific dialogue dataset. The system integrates offline Speech-to-Text (Vosk) and Text-to-Speech (Silero) modules to enable natural, interactive communication without relying on cloud-based services. The prototype demonstrates strong semantic response quality (BERTScore-F1: 0.9040) and maintains low-latency performance suitable for real-time gameplay environments (STT + Chatbot 330 ms).

The model successfully generates lore-consistent NPC responses directly from player speech input, illustrating the feasibility of deployable, lightweight, and fully offline conversational NPCs in role-playing games. While this work provides a functional demonstration, it serves as an initial foundation for further advancements in dynamic and context-aware game dialogue systems.

A. Key Contributions

- Developed a lightweight Transformer-based dialogue model trained on 7,565 Skyrim NPC–player dialogue exchanges.

- Designed a fully offline inference pipeline using Vosk for speech recognition and Silero for speech synthesis.
- Implemented and evaluated a real-time prototype using a Tkinter-based GUI to demonstrate practical interaction flow and measure component latency.
- Conducted preliminary model interpretability analysis through attention visualization to understand dialogue relevance and token dependency.

B. Future Scope

- **Contextual Memory:** Incorporating conversation history to enable ongoing context rather than single-turn responses.
- **Personality and Emotion Modeling:** Embedding NPC personality traits, tone variation, and emotional modulation in dialogue.
- **Game Engine Integration:** Embedding the system natively into engines like Unity or Unreal for direct in-world NPC interaction.
- **Reinforcement-Based Dialogue Adaptation:** Allowing NPC responses to adapt based on player behavior and quest progression.
- **Performance Optimization:** Further reducing inference latency and improving on-device efficiency for low-resource hardware.

By advancing toward these areas, this work lays the groundwork for immersive, adaptive conversational NPCs that strengthen role-play depth and player engagement in future game environments.

C. Future Work

Future work will focus on advancing the current prototype into a system robust enough for deployment within real game environments. The primary directions include:

- **Game Engine Integration:** Developing interfaces (e.g., through sockets, APIs, or embedded plugins) to connect the chatbot system directly to common game engines such as Unity and Unreal Engine. This will enable seamless real-time NPC speech and dialogue as an in-game mechanic rather than a standalone prototype.
- **Context Awareness:** Enhancing the chatbot's ability to maintain and utilize conversational context. Future improvements include storing dialogue history and integrating real-time game state information (such as player position, active quests, character relationships, and surrounding events). This will allow the system to produce adaptive, situationally relevant, and multi-layered dialogue rather than isolated question-response interactions.
- **System Robustness:** Improving system performance under real-world gaming conditions. This includes handling background noise, variations in player accents, and acoustic characteristics of different gaming environments. Possible approaches involve fine-tuning the STT model, incorporating noise reduction methods, or applying domain-specific audio preprocessing.
- **Model Optimization:** Applying model compression techniques to reduce computational demands. This will support efficient deployment on resource-constrained platforms, including gaming consoles, VR devices, and mobile hardware, without compromising response quality.
- **Expressive and Personalized TTS:** Enhancing the Text-to-Speech system to support emotional prosody control and multiple voice profiles. This would allow NPCs to convey different emotions, tones, and personality traits, resulting in more natural and expressive character interactions.
- **Multi-Character and Multi-Language Support:** Extending the architecture to support dialogues involving multiple NPCs and adapting the model to additional languages. This will broaden accessibility and enable diverse narrative and cultural representations within game environments.

These enhancements aim to bridge the gap between the current prototype and a fully integrated, contextually intelligent, and robust voice interaction system for immersive gaming experiences.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[2] K. Ramineni, K. H. Reddy, L. S. T. Chary, L. Nikhil, and P. Akanksha, "Designing an intelligent chatbot with deep learning: Leveraging FNN algorithm for conversational agents to improve the chatbot performance," in *World Conference on Artificial Intelligence: Advances and Applications*, Singapore: Springer Nature, Feb. 2024, pp. 143–151.

[3] A. Sawant, S. Phadol, S. Mehere, P. Divekar, S. Khedekar, and R. Dound, "ChatWhiz: Chatbot built using transformers and Gradio framework," in *Proc. 5th Int. Conf. on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2024, pp. 454–461, IEEE.

[4] N. Esfandiari, K. Kiani, and R. Rastgoo, "Transformer-based generative chatbot using reinforcement learning," *Journal of AI and Data Mining*, vol. 12, no. 3, pp. 349–358, 2024.

[5] H. A. Ahmad and T. A. Rashid, "Planning the development of text-to-speech synthesis models and datasets with dynamic deep learning," *Journal of King Saud University–Computer and Information Sciences*, vol. 36, no. 7, p. 102131, 2024.

[6] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart, *et al.*, "XTTS: A massively multilingual zero-shot text-to-speech model," *arXiv preprint arXiv:2406.04904*, 2024.

[7] D. H. Cho, H. S. Oh, S. B. Kim, S. H. Lee, and S. W. Lee, "Emosphere-TTS: Emotional style and intensity modeling via spherical emotion vector for controllable emotional text-to-speech," *arXiv preprint arXiv:2406.07803*, 2024.

[8] X. Zhu, Y. Lei, T. Li, Y. Zhang, H. Zhou, H. Lu, and L. Xie, "MetTS: Multilingual emotional text-to-speech by cross-speaker and cross-lingual emotion transfer," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 32, pp. 1506–1518, 2024.

[9] X. Xiong and M. Zheng, "GPT-Neo-CRV: Elevating information accuracy in GPT-Neo with cross-referential validation," *Authorea Preprints*, 2024.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[11] J. Shi, M. Likhomanenko, Q. Zhang, R. Schlüter, and Y. Saraf, “Emformer: Efficient memory transformer for streaming speech recognition,” in *Proc. ICASSP*, 2021, pp. 6783–6787.
[12] K. Yao, H. Hu, X. Chen, S. Watanabe, and D. S. Park, “Zipformer: Unified streaming and non-streaming ASR,” in *Proc. ACL Industry Track*, 2025.
[13] Z. Bai, Q. Zhang, Y. Wu, and S. Watanabe, “WhiSLU: End-to-end SLU with Whisper,” in *Proc. Interspeech*, 2023.
[14] L. Wang, X. Chen, H. Hu, and S. Watanabe, “Cross-modal selective self-training for zero-shot SLU,” in *Proc. EACL*, 2024.
[15] Y. Li, Z. Bai, and S. Watanabe, “GMA-SLU: Label-conditioned audio generation for SLU,” in *Proc. IJCAI*, 2024.
[16] P. Kumar, R. Singh, and M. Verma, “Context-aware ASR for video games,” in *Proc. ACM Multimedia*, 2024.
[17] M. Lee, H. Park, and S. Kim, “Voice-controlled NPC dialogue in Unity,” in *Proc. IWSDS*, 2025.
[18] T. Brown, L. Johnson, and M. Williams, “Player perception of NPC speech systems,” *Games Studies*, Taylor & Francis, 2024.

TABLE V
COMPARISON OF KEY WORKS IN STREAMING ASR, SLU, AND GAME-SPECIFIC SPEECH INTERACTION

Author & Year	**Method**	**Dataset**	**Metric**	**Strengths**	**Limitations**
Gulati et al., 2020	Conformer	LibriSpeech	WER 2.1%	High accuracy, local + global modeling	Not streaming capable
Shi et al., 2021	Emformer	LibriSpeech	Low WER @ 200 ms	Streaming friendly	Complex implementation
Yao et al., 2025	Zipformer	LibriSpeech	Low WER, memory efficient	Flexible streaming/non-streaming	Emerging tooling
Bai et al., 2023	WhiSLU	SLURP	Improved F1	Robust pretrained model	Large model size
Wang et al., 2024	CMSST	SLURP	Zero-shot F1 increase	Label-efficient learning	Complex pipeline
Li et al., 2024	GMA-SLU	SLURP	E2E SLU accuracy	Reduces annotation cost	Domain mismatch risk
Kumar et al., 2024	Context-aware ASR	Game prototype	Command recognition accuracy	Game vocabulary biasing	Needs game state integration
Lee et al., 2025	NPC voice dialogue	Unity prototype	Usability metrics	Immersive interaction	Dependent on ASR quality
Brown et al., 2024	NPC UX study	Gameplay	Latency tolerance	UX insights	No technical model