Who Are You Talking To? A Gaze-Aware Conversational Robot for Multi-User Interaction

Kevin Zhou
yrzhou@umich.edu

Zheng Wang
zzww@umich.edu

***Abstract*—Multi-party conversational systems face the fundamental challenge of determining who is speaking to whom (W2W), requiring the ability to distinguish between speech directed at the system versus speech between human participants. We present a gaze-aware conversational robot that addresses this challenge through head pose estimation-based attention gating, enabling the system to respond only when users direct their gaze toward it. Our lightweight, edge-deployable architecture integrates MediaPipe for gaze detection, Whisper for speech recognition, Qwen3-4B for language modeling, and Piper TTS for synthesis, all running locally on consumer-grade hardware. Evaluation comparing our system with ChatGPT's video mode across 24 conversation rounds demonstrates that our approach achieves a 90.9% success rate in avoiding inappropriate interruptions, compared to ChatGPT's 0% success rate, validating the critical importance of gaze-aware attention gating for socially appropriate multi-party human-robot interaction.**

I. Introduction

The pursuit of artificial agents capable of fluid, human-like communication has historically focused on dyadic, closed-loop interactions where speech detection alone implies engagement. This Voice Activity Detection (VAD) paradigm drives current smart speakers like Siri and Alexa [5], but leaves them socially unaware, and unable to distinguish a command directed at them from a side comment to a bystander. Recent multimodal models like OpenAI's GPT-4o [6] have evolved beyond traditional pipelines, offering unified neural networks with near-human latency and video understanding. However, their reliance on massive cloud infrastructure introduces variable network latency and significant privacy risks, particularly for sensitive domestic or healthcare environments.

As robots enter dynamic, real-world settings, they face the full social complexity of the Cocktail Party Problem [1] , the challenge of selectively attending to a single relevant speaker while filtering out overlapping human-human conversations and ambient noise. In Multi-Party Conversations (MPC), a robot that responds to every sound becomes a disruptive intruder, while one requiring constant wake words breaks natural dialogue flow. Social competence thus requires continuously solving the Who is Speaking to Whom (W2W) problem. This is a multimodal challenge where non-verbal cues, specifically gaze and head orientation, are paramount for signaling attention and managing the conversational floor.

A critical dichotomy exists in current Human-Robot Interaction (HRI) research. State-of-the-art approaches often rely on computationally intensive Multimodal LLMs that demand significant GPU clusters and cloud connectivity. While powerful, these systems suffer from latency that fractures conversational fluidity and privacy concerns inherent to streaming video. Conversely, there is an urgent need for lightweight, edge-native solutions that ensure privacy and deterministic latency on standard hardware.

Our work introduces a Gaze-Aware Conversational Robot architecture designed specifically for edge deployment. We propose a system that bridges the gap between high-fidelity social signal processing and low-latency interaction by leveraging an asynchronous multimodal architecture that decouples perception from cognition. At its core, we implement a lightweight visual attention mechanism that acts as a visual wake word, solving the W2W problem geometrically rather than semantically. By integrating quantized Small Language Models and optimized speech recognition, our approach establishes a performance baseline for real-time, privacy-preserving interaction on standard consumer hardware.

II. Related Work

In this section, we examine the theoretical role of gaze in communication, the evolution of W2W solutions, and the recent advancements in efficient AI models that make local deployment feasible.

A. The Regulatory Function of Gaze

Gaze is arguably the most powerful non-verbal cue in face-to-face interaction, serving functions that are both monitoring and regulatory. As noted in foundational psychological research [2, 7], gaze operates as a channel for social signals that manage the flow of conversation.

1) Turn-Taking Regulation: The organization of turn-taking is governed by a mutual-break pattern. Typically, a speaker averts their gaze at the beginning of a turn to engage in cognitive planning and returns their gaze to the listener at the end of the turn to signal that the floor is open [4]. This visual handshake acts as a synchronization signal. In HRI, implementing human-like gaze protocols [9] improves the fluidity of interaction, allowing users to intuitively know when the robot has finished speaking and is expecting a response.

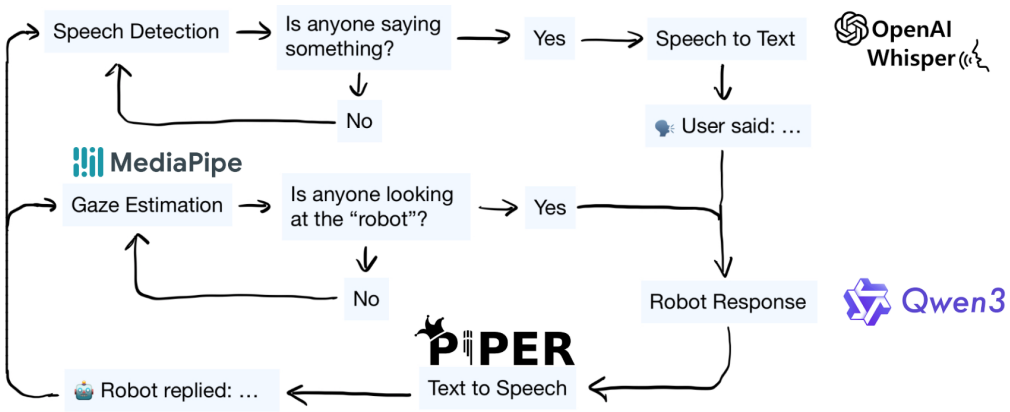


Fig. 1: Overview of the system architecture.

2) Addressee Specification: In multi-party settings, gaze functions as a social spotlight, identifying the intended recipient of a message. Vertegaal et al. [17] demonstrated that in triadic conversations, speakers gaze at the addressee approximately 77% of the time during speech. This high correlation forms the theoretical basis for most vision-based addressee detection systems.

B. W2W Challenge

Previous work on the W2W problem can be broadly categorized into text-based and multimodal fusion approaches, both of which suffer from limitations in real-time embodied contexts.

1) Text-Based Natural Language Processing (NLP) Approaches: Early attempts to solve W2W relied heavily on linguistic analysis. Le et al. [10] introduced "who-to-whom" models to identify addressees in multi-party text corpora using transcript analysis. These models analyze the semantic content of utterances to infer relationships. However, they are fundamentally non-embodied and non-causal. They typically require the entire transcript or significant future context to make a prediction. This latency makes them incompatible with a robot that must react as the interaction unfolds. A robot cannot wait for a sentence to finish and be analyzed before deciding to pay attention.

2) Multimodal Fusion Approaches: More recent approaches have embraced multimodal data, and used computationally heavy, data-driven models. Lee et al. [11] proposed a Transformer-based model to analyze and predict turn-taking in multi-party conversations. By fusing gaze, head movements, body movements and speech, their model effectively forecasts turn transitions. Mazzola et al. [13] proposed a Deep Neural Network (DNN) for the iCub robot that fuses facial features and body pose to estimate the target of a user's speech. Skantze et al. [15] applied large, pre-trained Transformer models, TurnGPT and Voice Activity Projection (VAP) to predict turn-taking opportunities in human-robot interaction. Wang et al. [18] presented LaMI, a large language model framework for multimodal HRI, integrating verbal and non-verbal behaviors such as gaze and gesture for context-aware interaction. While these systems achieve high accuracy, they rely on heavy architectures that demand significant onboard compute power or cloud offloading. This creates a barrier to entry for widespread deployment on peripheral robots or consumer-grade devices.

Some other effective solutions, particularly in Virtual Reality (VR) [19], rely on rich data streams from headsets. While accurate, these are impractical for spontaneous HRI where users are unencumbered by wearable sensors.

3) Lightweight Alternatives: The research gap identified lies in the lack of lightweight solutions. While some work, such as Heo et al. [3], explores gaze-enhanced turn-taking for hearing assistance, there is a scarcity of integrated systems that solve the W2W problem specifically for robotic response generation on edge hardware. Our project addresses that gap by utilizing efficient, purpose-built models rather than monolithic end-to-end transformers.

C. Advances in Efficient AI

The feasibility of the proposed system rests on recent breakthroughs in model efficiency.

1) Visual Perception: Google's MediaPipe framework [12] significantly advanced on-device vision. Its Attention Mesh architecture enables real-time 3D face landmarking on mobile CPUs by focusing computation on salient regions, avoiding the cost of heavy CNNs.

2) Small Language Models (SLMs): The release of the Qwen 2.5 and Qwen 3 series [21, 20] marks a paradigm shift. These models, particularly in the 3B-4B parameter range, utilize Grouped Query Attention (GQA) and high-quality training data to rival the performance of much larger models while remaining runnable on consumer GPUs.

3) Speech Processing: OpenAI's Whisper [14] has standardized robust speech recognition, and its quantized variants (e.g., whisper.cpp) allow for CPU-based inference. Similarly, Piper TTS utilizes the VITS architecture to provide low-latency neural synthesis on hardware as limited as a Raspberry Pi.

III. Technical Approach

As shown in Figure 1, the proposed system is designed as a modular, asynchronous pipeline. The core design philosophy prioritizes latency reduction and thread safety, ensuring that the robot's perception (vision and audio) does not block its cognition (LLM) or expression (TTS). The system is implemented in Python, utilizing threading and queue primitives to manage data flow between four primary modules: The Eyes, The Ears, The Brain, and The Mouth.

A. The Eyes: Gaze-Based Attention Gating

The visual perception module is responsible for estimating the user's attention target. Unlike expensive eye-tracking hardware, this system uses Head Pose Estimation (HPE) as a robust proxy for gaze.

1) Face Landmark Detection via MediaPipe: The system utilizes MediaPipe Face Mesh, a lightweight solution that estimates 468 3D face landmarks. The choice of MediaPipe is driven by its exceptional efficiency on CPU architectures. As noted in benchmarks, MediaPipe Face Mesh can achieve over 30 FPS on standard CPUs and operates with sub-10ms latency on mobile chipsets. This efficiency is achieved through a "blazeface" detector that crops the region of interest, followed by a landmark regression model that tracks the mesh frame-to-frame without re-running the full detector. The system is configured to support multi-user scenarios with up to three faces detected simultaneously (max_num_faces=3), enabling the robot to track attention across multiple participants in group interactions.

2) Geometric Pose Estimation: To determine where the user is looking, the system employs the Perspective-n-Point (PnP) algorithm. The code extracts six key 2D landmarks from the video frame: the nose tip, chin, left/right eye corners, and left/right mouth corners. These 2D points are mapped to a pre-defined generic 3D face model ($\mathcal{M}_{3D}$).

The PnP solver minimizes the reprojection error to find the rotation vector $\vec{r}$ and translation vector $\vec{t}$ that align $\mathcal{M}_{3D}$ with the 2D observation, given the camera's intrinsic matrix $\mathbf{K}$.

$$\min \sum_i ||p_i - proj(\mathbf{K}, \mathbf{R}, \vec{t}, P_i)||^2 \tag{1}$$

Where P_i are the 3D model points and p_i are the observed 2D image points. To improve tracking stability across frames, the system uses the previous frame's pose estimate as an initial guess (useExtrinsicGuess) when the same face is detected within a distance threshold. The resulting rotation vector is converted to a rotation matrix via the Rodrigues formula and then decomposed using QR decomposition (cv2.RQDecomp3x3) to extract pitch, yaw, and roll angles.

3) Attention Logic and Calibration: The raw Euler angles are adjusted using calibrated offsets (YAW_OFFSET, PITCH_OFFSET) to define a zero vector relative to the camera axis.

$$\psi_{centered} = \psi_{raw} - \psi_{offset} \tag{2}$$

$$\theta_{centered} = \theta_{raw} - \theta_{offset} \tag{3}$$

The system defines a cone of attention using configurable thresholds, defaulting to $\pm 20^\circ$ for both yaw and pitch. This default configuration worked well across all cameras tested during development. However, for the final demo on one group member's laptop, we found that thresholds of $\pm 7^\circ$ for yaw and $\pm 10^\circ$ for pitch achieved the best accuracy, and these values were used in the final system. A boolean flag is_looking is set to True only if the user's head orientation falls within this cone:

$$\begin{aligned} \text{is\_looking} = &(|\psi_{centered}| < \psi_{threshold}) \wedge \\ &(|\theta_{centered}| < \theta_{threshold}) \end{aligned} \tag{4}$$

This boolean flag serves as the master gate for the entire interaction. In the shared RobotState, this flag is updated at the frame rate of the camera (typically 30Hz), providing a high-temporal-resolution signal of user intent. Figure 2 illustrates head pose estimation for different gaze directions, demonstrating how the system distinguishes between looking at the robot (middle) versus looking away in various directions.

B. The Ears: Auditory Perception and VAD

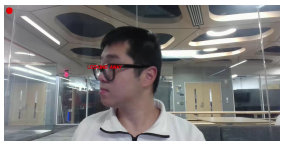
The auditory module manages audio capture, Voice Activity Detection (VAD), and Speech-to-Text (STT) transcription using two primary libraries: the `speech_recognition` library for audio capture and VAD, and OpenAI's `whisper` library for transcription.

1) Audio Capture and VAD: Our system uses the `speech_recognition` library to interface with the microphone. It employs an energy-based VAD with a fixed energy threshold (1000) to segment audio streams into phrases. The system supports platform-specific audio configurations: on Linux, audio is captured at 44.1 kHz, while other platforms use 16 kHz sampling. A critical parameter is the `phrase_timeout` (set to 3.0 seconds), which determines how much silence must elapse before the system considers an utterance complete. The `record_timeout` is set to 30.0 seconds to allow for longer continuous speech segments. This introduces a mandatory floor on latency but is necessary to prevent cutting off users mid-sentence.

2) Whisper Model Integration: Transcription is handled by OpenAI's Whisper model via the `whisper` library, specifically the base.en variant. The base model (74M parameters) represents a strategic compromise for edge interaction.

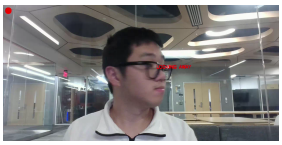
While the tiny model is faster, base.en offers significantly better accuracy for English conversation while remaining computationally viable. On GPUs such as the RTX 4060, base.en can transcribe audio at roughly 16x real-time speed using FP16 precision when CUDA is available. Even on CPU-only setups, optimized implementations like whisper.cpp allow base models to run effectively. The base model requires approximately 1GB of VRAM, leaving ample space on consumer GPUs (typically 6-12GB) for the LLM. On Linux systems, the audio is resampled from 44.1 kHz to 16 kHz (Whisper's expected sample rate) using scipy's signal resampling before transcription.

(a) Up

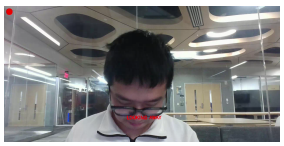


(b) Left

(c) Middle



(d) Right



(e) Down

Fig. 2: Examples of head pose estimation for different gaze directions: (a) looking up, (b) looking left, (c) looking at the robot (middle), (d) looking right, and (e) looking down. The system uses these pose estimates to determine if the user is directing their attention toward the robot.

3) Self-Inhibition Mechanism: A common failure mode in full-duplex conversational systems is the "echo loop," where the robot hears its own voice, transcribes it, and responds to itself. To solve this, the system implements a software-based inhibition mechanism akin to the biological efferent copy. The ear worker continuously checks the state of the TTS engine. If the robot is speaking, the input audio buffer is aggressively cleared. This effectively "deafens" the robot to its own output without requiring complex acoustic echo cancellation (AEC) hardware.

C. The Brain: Cognitive Processing with Local LLMs

The cognitive core (LanguageModel class) is responsible for understanding the user's intent and generating a response. This module leverages the Qwen3-4B-Instruct model in FP8 quantized format, which provides exceptional performance-to-size ratios while reducing memory requirements.

1) Model Selection: Qwen 3: The Qwen 3 model utilizes SwiGLU activation functions and Grouped Query Attention (GQA). GQA significantly reduces the memory bandwidth required for loading the Key-Value (KV) cache during decoding, which is often the bottleneck in autoregressive generation on edge devices. The FP8 quantization further reduces memory footprint, enabling the model to run efficiently alongside other system components on consumer GPUs.

2) Gaze-Gated Processing: The brain worker thread implements gaze-based gating at the cognitive level. Before processing any transcribed speech, the system checks the shared `is_looking` flag from the gaze detector. Only when the user is directing their attention toward the robot (as determined by the head pose estimation) does the system generate a response. Speech detected while the user is looking away is ignored, preventing the robot from responding to conversations not directed at it.

3) Streaming and Latency Masking: To mitigate the perception of delay, the LanguageModel utilizes token streaming via the TextIteratorStreamer from the Transformers library. Rather than waiting for the entire response to be generated, the system processes tokens as they are produced. Each token chunk is immediately sent to the TTS queue via a streaming callback function, allowing the TTS engine to begin synthesis as soon as the first tokens arrive. After generation completes, the system calls `flush()` to ensure all remaining audio is synthesized. This pipeline parallelism drastically reduces the Time-to-First-Audio (TTFA).

4) System Prompt Engineering: The code explicitly injects a system prompt to constrain the model's output: "You are a helpful assistant. Keep your responses brief and concise - limit your answers to 1-3 sentences. This is for a voice interface, so be conversational but brief. NO EMOJIS ALLOWED." The model is configured with a maximum of 200 new tokens per response (default). This constraint serves a dual purpose: it shapes the robot's persona to be efficient and conversational, and it linearly reduces inference latency by capping the number of tokens generated.

D. The Mouth: Neural Text-to-Speech

The final stage of the pipeline is expression, handled by the TextToSpeech module using Piper TTS. The module uses a threaded, queue-based architecture to ensure non-blocking audio synthesis.

1) Piper Architecture (VITS): Piper is based on the VITS (Conditional Variational Autoencoder with Adversarial Learning) architecture. Unlike older concatenative systems, VITS produces high-fidelity, natural-sounding speech. Crucially, Piper is optimized for the ONNX runtime, allowing it to run

faster than real-time even on low-power CPUs like the Raspberry Pi 4. The system uses the `en_US-lessac-medium` voice model by default, which provides natural-sounding American English speech.

2) Sentence-Level Streaming: The system implements a producer-consumer pattern for audio synthesis using a background worker thread. As the LLM generates text tokens, they are sent to the TTS module via the `speak_streaming()` method. The TTS module accumulates these tokens in an internal text buffer. Upon detecting a sentence delimiter (period, exclamation mark, or question mark followed by whitespace or end-of-string), the complete sentence is extracted and pushed to a queue for synthesis. Incomplete sentences remain in the buffer until the next delimiter or until `flush()` is called. This parallelism allows the robot to vocalize the start of a response while the LLM is still reasoning about the end of the response, effectively masking generation latency.

3) Audio Generation and Playback: The worker thread processes the queue, generating WAV audio files using Piper's command-line interface. Piper reads text from stdin and writes the synthesized audio to a temporary WAV file. The audio is then played using platform-specific commands: `aplay` for Linux, `afplay` for macOS, and PowerShell's `Media.SoundPlayer` for Windows. After playback, temporary files are automatically cleaned up. The module tracks speaking status via an `is_speaking()` method that checks both the queue state and a 0.5-second cooldown period after speech completion, which is used by the ear worker for self-inhibition.

IV. Evaluation

We conducted a comprehensive evaluation comparing our gaze-aware conversational robot system with ChatGPT's video mode. The evaluation was designed to assess the systems' performance in realistic multi-party conversational scenarios, where the ability to distinguish between human-human and human-robot interactions is critical.

A. Experimental Setup

The evaluation consisted of 4 conversations, with 4-8 rounds of discussions in each, totaling 24 rounds of interaction. Each conversation involved two human participants engaging in natural dialogue, with occasional queries directed at the robot. During the experiment, 11 rounds were designated as human-human interactions, while 13 rounds were designated as human-robot interactions. The conversations covered diverse topics including travel planning, academic discussions, and casual social interactions. Both systems were evaluated under identical conditions in a typical indoor environment, with participants instructed to interact naturally without explicit guidance on when to address the robot.

As an example of the conversations evaluated, Figure 3 shows a typical interaction between two participants (Human 1 and Human 2) and the robot. This conversation, which is also featured in our demo video, demonstrates the system's ability to distinguish between human-human dialogue and human-robot interactions. The robot correctly ignores the initial conversation between Human 1 and Human 2 about winter holiday plans, only responding when Human 2 explicitly directs their gaze toward the robot. However, the conversation also reveals some limitations: the robot failed to respond to several queries directed at it, illustrating the missed responses metric discussed in our evaluation.

Human 1 (to Human 2): Hey, what's your plan for the winter holidays?
Human 2 (to Human 1): I'm going back home to spend time with my families. What about you?
Human 1 (to Human 2): I'm not sure. I'm considering going somewhere for a trip. Do you have any recommendations?
Human 2 (to Human 1): Hmm, I don't know either.
Human 2 (to the robot): Do you have any recommendations on where to go on a trip during winter?
Robot: ...
Human 1 (to the robot): Can you find somewhere warm?
Robot: ...
Human 1 (to the robot): Can you recommend some places in the United States?
Robot: ...
Human 1 (to Human 2): Have you been to any of these?
Human 2 (to Human 1): Yes, I've been to some of them. I think it's a great list.
Human 2 (to the robot): Thank you so much!
Robot: ...

Fig. 3: Example conversation from our evaluation, also featured in the demo video. The robot correctly ignores human-human interactions and only responds when users direct their gaze toward it.

B. Results and Analysis

Table I summarizes the key findings from our evaluation. The most significant difference between the two systems lies in their ability to avoid inappropriate interruptions. Our system had only 1 inappropriate interrupt out of 11 possible opportunities (90.9% success rate), demonstrating the effectiveness of our gaze-based attention gating mechanism. In contrast, ChatGPT interrupted 11 times, meaning it interrupted during all human-human interactions, as it lacks the ability to distinguish between speech directed at the system versus speech between human participants.

For human-robot interactions, our system had 3 missed responses out of 13 opportunities, which we attribute primarily to voice recognition limitations in challenging acoustic environments. ChatGPT did not miss any responses, likely due to its more robust cloud-based speech recognition infrastructure.

Regarding latency, our system achieved an average response time of 2.24 seconds, while ChatGPT's latency was slightly lower at 1.98 seconds. This small difference (0.26 seconds) is expected given that our system runs entirely on local hardware, processing all components on-device.

To evaluate response quality, we enlisted two independent evaluators to rate both systems. Our system received an

Metric	Our System	ChatGPT
Inappropriate interrupts (max: 11)	1	11
Missed responses (max: 13)	3	0
Latency of responses	2.24 s	1.98 s
Quality of responses (Third-party ratings)	7/10	9.5/10
Preference (Third-party ratings)	1	1

TABLE I: Comparison of our system and ChatGPT video mode across key metrics.

average score of 7/10, while ChatGPT achieved 9.5/10. This difference is understandable given that ChatGPT leverages GPT-4, a much larger language model with superior training data. However, the evaluators emphasized that inappropriate interruptions significantly degraded their overall experience with ChatGPT, despite its superior response quality, highlighting the critical importance of social appropriateness in conversational systems.

Despite the differences in individual metrics, both systems received equal preference ratings from the evaluators (1 vote each). One evaluator preferred ChatGPT for its superior response quality and reliability, while the other preferred our system for its ability to respect conversational boundaries and avoid disruptive interruptions.

C. Qualitative Observations

Beyond the quantitative metrics, participants reported feeling more comfortable with our system during human-human interactions, as they did not need to worry about accidentally triggering the robot.

The evaluators provided valuable feedback: they noted that if they did not want ChatGPT to interrupt, they could simply turn it on only when needed. However, this manual intervention defeats the purpose of a natural, always-available conversational interface. More importantly, they mentioned that if ChatGPT could determine whether the user is talking to it or not, similar to our system's gaze-aware capability, it would be significantly better. This feedback directly validates the core contribution of our work: the importance of gaze-aware attention gating in multi-party conversational systems.

V. Discussion

A. Advantages

Our gaze-aware conversational robot system offers several key advantages.

First, the system provides explicit addressee awareness in multi-user settings through gaze-based attention gating. This mechanism achieves a 90.9% success rate in avoiding inappropriate interruptions, demonstrating its effectiveness in multi-party scenarios. Unlike systems that rely solely on audio input, our approach can distinguish between speech directed at the robot versus speech between human participants, enabling socially appropriate behavior in group interactions. This "invisible presence" quality—where the robot is available but non-intrusive— allows participants to engage in natural human-human dialogue without worrying about accidentally triggering the robot.

Second, our system is lightweight and runs entirely locally on consumer-grade hardware. The modular architecture enables deployment on standard laptops or embedded robot processors without requiring specialized infrastructure. This local processing provides important benefits: complete privacy as all data remains on-device, no network latency or connectivity requirements, and consistent performance regardless of internet connectivity. The system makes it accessible for widespread use without requiring cloud subscriptions or specialized infrastructure.

Third, the system requires no external hardware beyond a standard camera and microphone, which are commonly available on most computing devices. This eliminates the need for specialized eye-tracking equipment, wearable sensors, or other expensive peripherals, making the system practical for real-world deployment.

Fourth, the system achieves relatively low latency (2.24 seconds average response time), which falls within acceptable ranges for conversational interfaces. This is achieved through efficient local processing and pipeline parallelism, where gaze detection, speech recognition, language modeling, and text-to-speech operate asynchronously to mask individual component latencies.

Finally, the system has the potential to be deployed on real robots. The lightweight, edge-deployable architecture makes it suitable for integration into robotic platforms, enabling robots to participate naturally in multi-party conversations while respecting social boundaries. This capability is essential for robots operating in human environments where they must interact with multiple people simultaneously.

B. Limitations

However, our system also faces several limitations.

First, gaze estimation is not always reliable. The head pose estimation approach, while effective in many scenarios, has reduced accuracy when users are positioned at the corners of the camera frame. Additionally, the system is sensitive to lighting conditions, with performance degrading in low-light environments or when faces are heavily shadowed. These limitations can lead to false positives or false negatives in attention detection.

Second, speech-to-text accuracy is inconsistent and unreliable. The local Whisper model cannot achieve accurate transcription consistently, especially when users are not positioned close to the microphone. This limitation contributed to the 3 missed responses observed in our evaluation, as the system failed to recognize speech in challenging acoustic conditions. Furthermore, transcription accuracy varies significantly across different devices and operating systems. For example, parameters that work well on Windows may perform poorly on Linux, requiring platform-specific calibration and tuning. This device-dependent variability makes it difficult to achieve consistent performance across different deployment environments, limiting the system's generalizability and ease of deployment.

Third, the system does not explicitly track which specific human is speaking. While it can detect whether someone is looking at the robot, it cannot distinguish between different speakers in multi-party settings. This creates a potential failure mode: if Human 1 is talking to Human 2, but Human 2 happens to be looking at the robot for a few seconds, the system might incorrectly interpret this as Human 1 addressing the robot. This limitation contributed to the inappropriate interrupt observed in our evaluation.

Fourth, users cannot interrupt the robot when it is speaking. The system's self-inhibition mechanism, which prevents the robot from responding to its own voice, also means that users cannot stop or redirect the robot mid-response. This creates a one-way communication pattern that may feel less natural compared to human-human conversation, where interruptions and turn-taking are more fluid.

Fifth, the local language model sometimes makes mistakes. Despite explicit system prompts instructing the model to avoid emojis and respond in English, we observed occasional violations: the model sometimes generates responses in Chinese or includes emojis in its output. These errors reflect the limitations of smaller, quantized models compared to larger cloud-based systems, and highlight the challenges of constraining model behavior through prompting alone.

C. Future Work

The evaluation suggests that the ideal conversational system would combine the strengths of both approaches: ChatGPT's superior response quality and reliability with our system's gaze-aware attention gating. This could be achieved through hybrid architectures that combine local processing with selective cloud offloading, or by incorporating gaze awareness into cloud-based systems.

To address the identified limitations, several directions for future work emerge. First, to improve gaze estimation reliability, more extensive tuning of head pose estimation parameters would be needed, particularly for edge cases involving users at frame corners or challenging lighting conditions. Eye tracking, while requiring more computational resources, could provide more accurate attention detection and is worth exploring as an alternative or complementary approach to head pose estimation.

Second, to address speech-to-text inaccuracy and device variability, more comprehensive tuning of the Whisper model parameters is necessary across different platforms. Future work should investigate adaptive parameter selection based on device characteristics and acoustic environment analysis. Additionally, research is needed to understand and mitigate the device-dependent variability problem, potentially through platform-specific calibration procedures or unified parameter optimization frameworks.

Third, to track which specific human is speaking in multi-party settings, active speaker detection can be incorporated. Recent work on audio-visual active speaker detection [8, 16] demonstrates promising approaches for determining who is speaking by combining audio and visual cues. Integrating such methods would enable the system to distinguish between Human 1 speaking to Human 2 versus Human 1 speaking to the robot, even when Human 2 happens to be looking at the robot.

Fourth, to improve language model output quality and reduce errors, more detailed restrictions and guidance can be added to the system prompt. This includes explicit instructions about language requirements, formatting constraints, and content restrictions. Additionally, post-processing techniques can be implemented to filter out unwanted content such as Chinese words or emojis before the response is sent to the text-to-speech module, ensuring consistent output quality regardless of model behavior.

Finally, conducting larger-scale user studies would validate findings across diverse populations and contexts, and help refine the system's performance in real-world deployment scenarios.

The evaluators' feedback that ChatGPT would be significantly better if it could determine whether the user is talking to it or not directly validates the core contribution of our work: the critical importance of gaze-aware attention gating in multi-party conversational systems. This capability addresses a fundamental limitation of current conversational AI systems and represents a necessary step toward truly natural, socially appropriate human-robot interaction in multi-party settings.

VI. Conclusion

We have presented a gaze-aware conversational robot system that addresses the Who is Speaking to Whom (W2W) problem in multi-party interactions. By leveraging head pose estimation as a proxy for gaze, our system achieves a 90.9% success rate in avoiding inappropriate interruptions, demonstrating the critical importance of visual attention cues for socially appropriate human-robot interaction.

Our lightweight, edge-deployable architecture enables real-time processing on consumer-grade hardware, providing complete privacy and eliminating dependency on cloud infrastructure. The system's modular design, integrating gaze detection, speech recognition, local language models, and neural text-to-speech, demonstrates the feasibility of deploying sophisticated conversational AI entirely on local devices.

Evaluation results reveal important trade-offs between our approach and cloud-based systems like ChatGPT. While our system excels at social appropriateness through gaze-aware attention gating, it faces limitations in speech recognition accuracy and response quality compared to cloud-based alternatives. However, evaluator feedback validates that gaze awareness is a crucial capability that would significantly improve existing conversational systems, directly supporting our core contribution.

Future work should focus on improving gaze estimation reliability, addressing device-dependent speech recognition variability, incorporating active speaker detection, and enhancing language model output quality through better prompting and post-processing. The demonstrated effectiveness of gaze-based attention gating establishes a foundation for more nat-

ural, socially competent conversational robots in multi-party settings.

Acknowledgments

This work was completed as part of the final project for ROB 498/599: Computational Human-Robot Interaction at the University of Michigan, Fall 2025. We acknowledge the use of AI coding assistants (Cursor and ChatGPT) for implementation support and writing assistance during the development of this work. The code and implementation are available at: https://github.com/kevin-yiran-zhou/Who-Are-You-Talking-To.git

References

[1] Edward Collin Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of the acoustical society of America*, 25:975–979, 1953.

[2] Ziedune Degutyte and Arlene Astell. The role of eye gaze in regulating turn taking in conversations: A systematized review of methods and findings. *Frontiers in Psychology*, Volume 12 - 2021, 2021.

[3] Seongsil Heo, Calvin Murdock, Michael Proulx, and Christi Miller. Gaze-enhanced multimodal turn-taking prediction in triadic conversations, 2025. URL https://arxiv.org/abs/2505.13688.

[4] Judith Holler and Kobin H Kendrick. Gesture, gaze, and the body in the organisation of turn-taking for conversation. In *the 14th International Pragmatics Conference*, 2015.

[5] Matthew B Hoy. Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88, 2018.

[6] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[7] Kristiina Jokinen, Hirohisa Furukawa, Masafumi Nishida, and Seiichi Yamamoto. Gaze and turn-taking behavior in casual conversational interactions. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3 (2):1–30, 2013.

[8] Okan Köpüklü, Maja Taseska, and Gerhard Rigoll. How to design a three-stage architecture for audio-visual active speaker detection in the wild. *arXiv preprint arXiv:2106.03932*, 2021.

[9] Melissa Kragten. The effects of non-verbal turn-taking cues in an open-domain human-robot conversation. Master's thesis, 2025.

[10] Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and Rui Yan. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1909–1919, 2019.

[11] Meng-Chen Lee, Mai Trinh, and Zhigang Deng. Multimodal turn analysis and prediction for multi-party conversations. In *Proceedings of the 25th International Conference on Multimodal Interaction*, pages 436–444, 2023.

[12] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

[13] Carlo Mazzola, Francesco Rea, and Alessandra Sciutti. Real-time addressee estimation: Deployment of a deep-learning model on the icub robot. *arXiv preprint arXiv:2311.05334*, 2023.

[14] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[15] Gabriel Skantze and Bahar Irfan. Applying general turn-taking models to conversational human-robot interaction. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 859–868. IEEE, 2025.

[16] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3927–3935, 2021.

[17] Roel Vertegaal, Robert Slagter, Gerrit Van der Veer, and Anton Nijholt. Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 301–308, 2001.

[18] Chao Wang, Stephan Hasler, Daniel Tanneberg, Felix Ocker, Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, and Michael Gienger. Lami: Large language models for multi-modal human-robot interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–10. ACM, May 2024. doi: 10.1145/3613905.3651029. URL http://dx.doi.org/10.1145/3613905.3651029.

[19] Portia Wang, Eugy Han, Anna CM Queiroz, Cyan DeVeaux, and Jeremy N Bailenson. Predicting and understanding turn-taking behavior in open-ended group activities in virtual reality. *Proceedings of the ACM on Human-Computer Interaction*, 9(7):1–40, 2025.

[20] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[21] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, et al. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025.