Dynamic Prompting Improves Turn-taking in Embodied Spoken Dialogue Systems

Yifan Shen*, Dingdong Liu*, Xiaoyu Mo, Fugee Tsung, Xiaojuan Ma, Bertram E. Shi

***Abstract*— The ability to coordinate turn taking during spoken dialogue is crucial for an embodied spoken dialogue system (SDS), e.g., in a humanoid robot. The SDS needs to model transitions in the conversational floor, which describes each party's stance (either speaking or listening). Further, the SDS needs to signal its perception of the floor to the human, so that they can coordinate floor transitions and resolve conflicts. Conventional SDS employ standalone modules to control floor transitions but do not produce timely and appropriate responses. Recent end-to-end audio LLMs generate responses quickly, but do not coordinate floor transitions as accurately. In this work, we propose an SDS architecture that dynamically adjusts its prompts to an end-to-end audio LLM based upon its perception of the conversational floor state. The LLM output determines not only the audio output, but also the perceived floor state. This enables the system to signal its stance to the human, both when listening and when speaking. We conducted an experiment where a humanoid robot administered a semi-structured interview with human subjects. Results show that, compared with baseline systems using static prompts, dynamic prompting enables the LLM to model floor transitions more accurately, to generate more appropriate signalling, and to interrupt less, leading to smoother turn-taking in dialogue.**

I. Introduction

Humanoid conversational robots offer significant advantages for social tasks [1], [2]. Their anthropomorphic design evokes expectations of human-like interaction. However, when these are unmet, the perceived interaction quality deteriorates [3], [4], [5]. Natural turn-taking in spoken dialogue is critical aspect in meeting these expectations, yet most spoken dialogue systems (SDS) [6], [7], [8] assume simplified patterns of alternating speech separated by silence, which fail to capture the nuanced, overlapping, and context-sensitive nature of human conversation.

Dialogue turn-taking can be conceptualized as joint management of the conversational floor [4]. The floor state comprises the stance of both interlocutors (speaking or listening). Stance transitions trigger floor state transitions. Human interlocutors coordinate turn-taking by estimating their partner's stance through behavioral cues, deciding their own stance accordingly, and signaling their floor state estimate via behavioral feedback [9], [10]. For natural human-robot interaction, an SDS should model floor transitions and signal its estimates similarly.

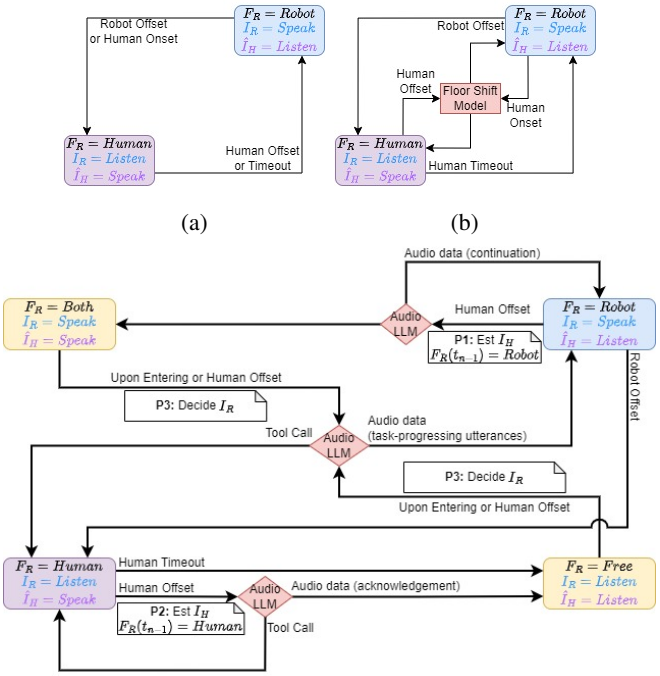

Fig. 1: (a) shows the state transition of VAD-based SDS. (b) shows state transition in more advanced SDS where a model of floor transition is employed. (c) illustrates the state transition diagram of the proposed system.

Conventional SDS architectures employ a pipelined approach. A dedicated module models floor transitions, identifying changes in human stance and triggering/stopping robot speech when needed [11]. While advanced models [12], [13] approximate human-human floor transition patterns effectively, the required audio-text conversion and cascading components introduce response delays that impede timely floor state signaling.

End-to-end audio LLMs like GPT4o-Realtime [14], [15], [16], [17] have emerged as alternatives. These process utterances from the human and generate responses directly without explicit audio-text conversion. By integrating response generation and floor transition modeling into a single model, they achieve substantially lower response latency. However, they typically model floor transitions less accurately than dedicated models in conventional SDS [15], [18].

All authors are with Hong Kong University of Science and Technology in the Department of Electronic and Computer Engineering (YS, BES), Department of Computer Science (DL, XJM), Division of Emerging Interdisciplinary Areas(XYM), Department of Industrial Engineering and Decision Analytics (FT), and the Center for Aging Science (BES). This work was supported by the HKUST Center for Aging Science (projects [TBD]).

Supplementary materials are in `https://drive.google.com/drive/folders/1S9_LlaB9lT6-52fOg70i6myVN7LC1bQ7?usp=sharing`

*These authors contributed equally to this work.

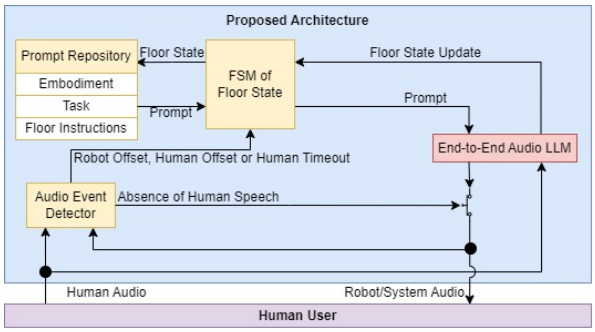

Fig. 2: Our proposed SDS architecture.

We propose an SDS architecture that dynamically prompts an audio LLM to model floor transitions and produce appropriate robot responses (Fig. 2). Our floor coordinator employs a dynamic prompting mechanism. Upon detecting audio events such as human utterance offsets, it uses context-specific prompts based on the current floor state estimate. These prompts instruct the LLM to provide updated human stance estimates,new robot stance decisions, and robot responses that signal the latest floor state estimate.

Evaluation in a humanoid robot interviewer scenario demonstrates that our architecture more accurately models floor transitions and better signals the system's intent than baseline systems using static prompts. Overall, this results in smoother dialogue turn-taking.

II. Background and Related Work

A. Turn-Taking and the Conversational Floor

Natural turn-taking in human-human dialogue can be described as transitions in the conversational floor [4], [19].

The stance of agent k, $I_{\mathrm{k}} \in \{\mathrm{Speak}, \mathrm{Listen}\}$, refers to their intent, where $k \in \{H, R\}$ denotes human or robot. $I_{\mathrm{k}} = \mathrm{Speak}$ means that agent k is speaking or intends to. $I_{\mathrm{k}} = \mathrm{Listen}$ means that agent k is attending for incoming speech. The state of the conversational floor F comprises of the stance of both parties, i.e., $F = (I_{\mathrm{R}}, I_{\mathrm{H}})$. Stance transitions lead to transitions in the conversational floor.

An agent's stance is correlated with, but not equivalent to, the presence or absence of speech. Thus, speech onsets or offsets do not imply stance transitions. For example, agent k could pause while $I_{\mathrm{k}} = \mathrm{Speak}$. They might also speak when $I_{\mathrm{k}} = \mathrm{Listen}$, e.g., when back-channelling (BC) [20], [21], [9].

To coordinate turn-taking, the two parties need to model floor transitions and signal their perception to one another.

First, each party maintains an estimate of the conversational floor by estimating the other agent's stance and deciding one's own stance accordingly. We denote the robot's estimate of floor state by $F_{\mathrm{R}} = (I_{\mathrm{R}}, \hat{I}_{\mathrm{H}})$, where the $\hat{}$ symbol indicates an estimate.

Second, to establish common ground and resolve conflicts, each party can signal its floor state estimate [10]. For example, when party A perceives that party B has finished speaking and intends to speak up, A may signal this with an acknowledging utterance, referred to as a turn-initial cue [22], which precedes A's stance transition. If A is wrong, this cue allows B to recognize A's mis-perception and resolve conflicts efficiently.

B. Turn-Taking in Existing SDS

Existing SDS models floor state F_R with different levels of granularity and flexibility.

The majority of SDS, including both pipelined systems and end-to-end audio LLM's, represent two different floor states, namely $F_R = Robot$ and $F_R = Human$. In the former state the system produces robot utterances, whereas in the latter it remains quiet and observes human utterances.

Transitions between these two states are managed to different level of flexibility.

The most naive solution involves a VAD module with silence thresholds (Fig. 1a). This approach equates onset/offset in human utterances with floor transitions [9]. The GPT4o-realtime API developed by OpenAI adopts this approach [14]. This assumption ignores the differences between behavior changes and stance changes noted above.

Pipelined SDS often employ more advanced models capable of recognizing floor transition events from acoustic and textual features [23], [24], [25]. State transitions in these systems are more flexible, since they do not necessarily start/stop robot utterances upon offset/onset of human utterances (Fig. 1b). For example, the Duplex Conversation system [23] achieves an F1 score of 0.89 in differentiating listeners' BC from actual barge-ins. Models like voice activity projection (VAP) [12] and Turn-GPT [13] utilize features from *both* parties' utterances to estimate floor transitions. Skantze et al. report the user felt interrupted by an SDS that combined these two models only 6.9% of the time [26], indicating that these models can accurately differentiate human pauses from floor transitions. However, due to their pipelined architecture, these SDS can be slow. For example, the system in [26] has a mean response latency of 1.5s. This prevents the system from producing timely signals of its floor perception.

Some end-to-end audio LLMs generate system utterances and model floor transitions simultaneously. For example, the systems in [18] and [27] are trained to produce special tokens representing floor transitions, which trigger starts/stops in audio streaming. However, the performance of these LLMs currently lags that of pipelined systems [28]. They also cannot estimate floor transitions accurately. For example, the RTTL-DG model in [18] achieved an F1 scores of 0.52 and 0.62 in starting and stopping speech.

Some end-to-end models like Moshi [17] and omni-flatten [15] eliminate explicit representation of floor states altogether. These LLMs constantly output audio chunks, which can contain silence if the system decides to listen. Making floor state implicit promises infinite granularity and flexibility. Unfortunately, these LLMs still do not model floor transitions accurately. For instance, the Moshi model and the omni-flatten model have accuracies of 0.55 and 0.71 in speaking up at the right time [15].

C. Our Contribution

In this work, we propose an SDS architecture to improve an end-to-end audio LLM's turn-taking performance using dynamic prompting (Fig. 2).

The dynamic prompt mechanism is based on a 4-state model of floor transitions (Fig. 1c), where we adjust the prompt based on F_R. When $F_R = Human$, at human offset, the audio LLM is prompted to evaluate whether the human has finished speaking. If so, F_R transitions to $Free$, where the audio LLM is triggered to estimate the system's next stance according to the dialogue task. Otherwise, F_R is unchanged. On the other hand, when $F_R = Robot$, if the human speaks up the audio LLM is prompted to evaluate the person's intent. If the human does not appear to want to interrupt, e.g. is backchanneling, F_R is unchanged. The system picks up from where it left off. Otherwise, F_R transitions to $Both$, where the LLM is prompted to decide its next stance.

The advantage of the proposed system is three-fold. First, the 4-state model affords more nuanced turn-taking patterns. In previous systems, F_R will transition into $Human$ when the person interrupts the robot to take the floor. Our 4-state model represents this situation as a different state $Both$, potentially enabling the robot to compete for the floor with the human in a debate. A similar formulation was proposed in [4], [19] for text-based systems.

Second, the dynamic prompting mechanism decouples low- and high-level aspects of turn-taking. When $F_R = Human$ or $Robot$, the prompts used are agnostic to the task. They only instruct the LLM to evaluate whether a complete message can or has been obtained/delivered. When $F_R = Free$ or $Both$, task-specific prompts are used to decide the robot's next step in achieving its conversational goal. Narrowing down the scope of floor transition modeling in each state simplifies the estimation task and allows more targeted prompts. Recently, Wang et al. [27] modified a chat-based LLM to directly interface with ASR and TTS modules, but used only a static prompt to estimate floor transitions. Our experimental results show that dynamic prompting produces fewer interruptions compared static prompting.

Finally, our system asks the LLM to simultaneously output floor state estimates and robot utterances, enabling the robot to signal its perception of the floor. Although previous studies can accurately estimate floor transitions [13][29], they have yet to demonstrate timely signalling.

III. Proposed Architecture

The proposed SDS architecture, illustrated in Fig. 2, consists of an audio event detector, a finite-state machine, a prompt repository and an end-to-end audio LLM. Our experiments use GPT4o-Realtime [14] for the audio LLM, but other end-to-end audio LLMs, such as Gemini 2 [30], could be used with no to minimal modification to our architecture. The LLM that takes in audio streams is prompted to return tool calls, audio streams, and transcripts.

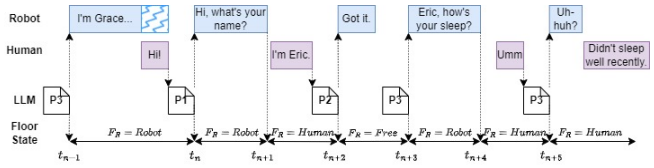

Fig. 3: Example work flow of the floor state FSM, annotated with audio events, prompts and sample transcripts. Solid arrows represent time periods when floor state estimate remains constant. Dashed arrows show how an audio event causes the FSM to prompt the LLM and update F_R. Zigzag shade represents robot utterances discarded due to human onset. Latency in event detection is not shown, and the length of solid arrows is not proportionate to actual duration.

A. Audio Event Detector

The audio event detector identifies onsets/offsets/timeouts in human/robot utterances. Events in human audio are detected via a VAD with a silence threshold [31].

At the onset of human utterances, ongoing processing in the audio LLM is stopped and robot utterances are cut-off (the switch in Fig. 2). At the offset of human/robot utterances or the timeout in human audio (absence of human utterances within a window), the finite-state machine is triggered.

B. Finite-State Machine for Floor State

We implement floor transitions via a finite-state machine (FSM) whose state transition diagram is shown in Fig. 1. An example workflow of the FSM is given in Fig. 3.

Transitions occur at times t_n. If $F_R = Robot$ or $Human$, transitions are triggered by human offsets detected by the audio event detector. If $F_R = Both$ or $Free$, transitions are triggered upon entry to the state or by human offsets detected by the audio event detector.

The FSM decides the next floor state by calling the audio LLM using a state-dependent prompt. If $F_R = Robot$ or $Human$, the prompt instructs the LLM to estimate the human's stance and change the floor state accordingly. If $F_R = Both$ or $Free$, the prompt instructs the LLM to determine the robot's stance and change the floor state to either $F_R = Robot$ or $Human$.

The prompts also instruct the LLM to generate audio outputs, whose delivery is initiated at state transitions. These outputs either progress the conversation to achieve the goals of the task or signal the robot's understanding of the floor state, e.g., backchannelling. Audio outputs that progress the conversation are generated to directly as audio chunks. Backchanneling outputs are chosen from among a set of pre-recorded utterances.

Below we give more details about the state transitions.

1) If $F_R(t_{n-1}) = Robot$: State transitions are triggered by offsets of human or robot utterances.

Upon a robot offset, the state directly transitions to $F_R(t_n) = Human$, i.e., the robot assumes that the human will respond to its completed utterance.

Upon a human offset, the FSM prompts the LLM to evaluate whether the human wishes to take over the floor, e.g.,

a pre-emptive answer to a question, or is backchanneling. If the former, the LLM returns a tool call, which contains a parameter indicating which from among a set of pre-recorded backchanneling utterances the robot should speak. The FSM sets the next state $F_R(t_n) = Both$. If the latter, the LLM returns audio data, which is biased to continue the robot's previous utterance. The state is unchanged.

2) If $F_R(t_{n-1}) = Human$: State transitions are triggered either by a human offset or after a timeout period during which the human is silent.

Upon a human timeout, the FSM directly sets directly transitions to $F_R(t_n) = Free$.

Upon a human offset, the FSM prompts LLM to determine whether the offset is due to a temporary pause in or to the end of the human's utterance. If the former, the LLM returns a tool call containing a parameter indicating which backchannel utterance the robot should deliver. The state remains unchanged ($F_R(t_n) = Human$). If the latter, the LLM returns audio data, which is biased by the prompt to be a non-committal acknowledgement,such as 'Got it' (Fig. 3). The FSM sets the next state to be $F_R(t_n) = Free$.

3) If $F_R(t_{n-1}) = Both$ or $Free$: Upon entry, the FSM prompts the LLM to determine the new robot stance I_R based on the past dialogue history and the demands of the task. If there is a human utterance during the LLM processing time, the request is canceled and re-issued after the human offset with an updated dialogue history that includes the most recent utterance.

If the LLM decides the robot should listen for more information, it returns a tool call. The FSM sets the next state $F_R(t_n) = Human$.

If the LLM decides the robot should speak, it returns audio data, which is biased by the prompt to both repeat/paraphrase information provided by the user and progress the dialogue task. The FSM sets the next state to be $F_R(t_n) = Robot$. The transcript of robot utterances provided by the LLM is recorded for composing prompt P2 in the $Robot$ state.

Our current implementation uses the same prompt in both states $F_R(t_{n-1}) = Both$ and $Free$. However, in future implementations, this should not be the case. For example, if $F_R(t_{n-1}) = Both$ the robot and human should be able to compete for the floor. However, due to current limitations in GPT4o-realtime, we always interrupt robot output utterances upon human onset. [1]

C. Prompt Repository

Prompts are assembled by combining up to four parts: an embodiment description, a task description, the floor instructions and a transcript.

The first two parts remain constant during the conversation. The embodiment description characterizes the form and capability of the robot. The task description outline the goal of the dialogue task and strategies to progress it.

[1]To estimate floor state upon overlap, robot utterances in the conversation history should contain only what has been said. GPT4o-realtime currently does not allow client programs to add robot utterances into the conversation history. Hence, robot utterance playback is stopped when truncating server-side history for consistency.

The floor instructions change depending upon the floor state, as described earlier. These provide guidance on how to estimate human stance, decide robot stance and signal robot perception of the floor. These instructions are task-independent.

The FSM queries the prompt repository with $F_R(n-1)$ to obtain prompts. If $F_R(n-1) = Robot$ or $Human$, the prompts (P1 or P2) contain the embodiment description and floor instructions, but not the task description. P1 also contains the transcript of the robot's current utterances.

If $F_R(n) = Both$ or $Free$, the returned prompt P3 contains the embodiment description, the task description, and the floor instructions.

Exact assembly procedures are exemplified in the "prompt_details" file in the supplemental materials.

IV. Experimental Methods

A. System Implementation

We implemented three SDS: a vanilla (baseline) system, a static floor-instructed system, and the proposed system with dynamic (state dependent) floor instructions.

For all systems the LLM is triggered at human offset or timeout. All processing/playback is stopped upon human onset, and the system starts a timeout timer on robot offset.

The vanilla system follows an "out-of-box" approach to using GPT4o-realtime as an SDS [32] following the state transition diagram in Fig. 1(a). Calls to the LLM are generated after a human offset or a human timeout using a static prompt consisting of embodiment and task descriptions only. The LLM generates audio data only, but no tool calls. Thus, the LLM *always* starts audio stream after any human offset or timeout. In other words, floor transitions are controlled reflexively by the VAD component.

The floor-instructed system uses a static prompt, which contains not only the embodiment and task descriptions, but also instructions on how to model floor transitions. These instructions combine the floor instructions used by the three different prompts (P1-P3) from the proposed system. It's operation follows the state transition diagram in Fig. 1(b). Calls to the LLM are generated after either human offset a human timeout. The LLM can either output an audio stream, which causes the state to transition to $F_R = Robot$ or make a tool call. which cause the state to transition to $F_R = Human$. The floor-instructed system prompts the LLM to model floor transitions, but does not alter the prompt based on floor state and has a simpler elaboration of floor state than the proposed system.

The same GPT4o-realtime model and VAD model were used by all systems. Timeout and VAD parameters were kept the same. The prompts share the same set of task descriptions/embodiment description and floor-instructions. Further, the floor-instructed system can produce pre-recorded BC in the same way as the proposed system.

All three systems were deployed on the same humanoid robot [33]. Human utterances were acquired through a collar microphone that subjects wore during the experiment. Robot utterances were played via its integrated speaker. More

System	# of Floor States	LLM-Based Floor Transitions	Dynamic Prompting
Vanilla	2	✗	✗
Floor-Instructed	2	✓	✗
Proposed	4	✓	✓

TABLE I: Key differences in the three systems.

details are provided in the "implementation_details" file in the supplementary materials.

B. Interaction Task

The robot administered semi-structured interviews (SSI) [34] about the human subjects' opinions and experiences on sleep-related behaviors: "revenge bedtime procrastination", "social jetlag" and taking naps during the day. The task descriptions used in the prompt are provided in the "prompt_details" file in the supplemental materials.

C. Procedure

Each experiment session consisted of three phases.

In the introduction phase, the experimenter explained the flow of the experiment to the subject, obtained written consent and collected demographic information.

In the interaction phase, the subject sat in front of the robot and had three conversations with it. In each conversation the robot was controlled by one of the three systems. The order of topics was fixed across experiment sessions. The order of system versions was randomized and counterbalanced. After each conversation, the subject filled out a questionnaire. The two parties' speech were segmented into inter-pausal units (IPUs) [9] by running a VAD offline. Details of the questionnaire and the segmentation are provided in the "questionnaire" file and the "robot_delay_calculation" file in the supplemental materials.

In the annotation phase, the subject listened through the three conversations in order. Similar to [26], the subject was instructed to inform the experimenter whenever they felt interrupted by the robot. The experimenter recorded the interrupting robot IPU together with the subject's explanation of why they felt interrupted.

D. Participant Demographics

We recruited 18 participants (12 male, 6 female), ranging in age from 21 to 48 years ($Mean = 26.3$, $SD = 6.24$). Their educational backgrounds were balanced between technical (10 participants; Mechanical Engineering, Electronic Engineering, Computer Science, etc) and non-technical disciplines (8 participants; Finance, Education, etc).

V. Results

A. Response Time and Interruption Rate

Following [26], we evaluated the turn-taking performance of an SDS by its response time and interruption rate. Response time is defined as time interval between the offset of a human IPU and the onset of the subsequent robot IPU. Interruption rate is defined as the portion of robot IPUs that interrupted the human subject. Often, systems must trade-off between the two metrics. For example, using a VAD with a short silence threshold for turn-taking leads to short response time but high interruption rate.

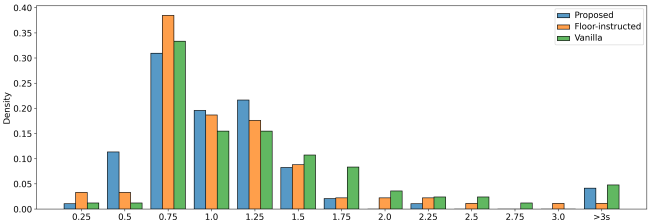

Fig. 4: Histogram of response time (s).

System Type	Response Time (s)		Interruption Rate	
	Mean	Median	Mean	Median
Vanilla	1.232	0.970	19.81%	12.13%
Floor-Instructed	**0.962**	**0.810**	12.77%	10.56%
Proposed	0.986	0.840	**7.82%**	**5.56%**

TABLE II: Mean and median of response time and interruption rate. The mean response time is computed by averaging over all human-robot IPU pairs from conversations controlled by a particular system. The mean interruption rate is computed for each conversation and averaged over conversations. The medians are computed accordingly.

Response time is computed from the IPUs of each conversation. See "robot_delay_calculation" in the supplementary material. Interruption rate is calculated by dividing the number of reported robot interruptions by the number of robot IPUs in a conversation.

Fig. 4 shows response time histograms for the different systems. The statistics are given in the first columns of Table II. Since the response time data are not normally distributed, we used the Mann-Whitney U test for between-system comparisons. Despite the increased complexity of the state space in the proposed system, there was no significant difference between the proposed system and the floor-instructed system ($U = 4416, p = 0.503$). The response time of the vanilla system was longer than both the floor-instructed system ($U = 3156, p = 0.023$) and the proposed one ($U = 3322, p = 0.016$).

The perceived response speed was evaluated in the post-interaction questionnaire using a visual analog scale (VAS). As shown in Fig. 5, there was no significant difference between the systems in terms of their deviation from the origin.

Fig. 6 shows box plots of the interruption rate. The second column of Table II shows the statistics. The vanilla system has the largest interruption rate, followed by the floor-instructed system and the proposed one.

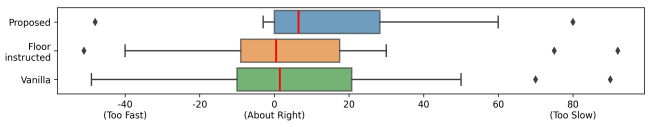

Fig. 5: Perceived latency of different systems.

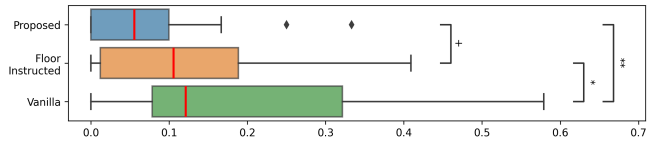

Fig. 6: Box plot of robot interruption rate. The red line shows the median, asterisks indicate significance ($+$: $p < 0.1$, $*$: $p < 0.05$, $**$: $p < 0.01$), and diamonds mark outliers.

The mean interruption rate of the vanilla system is 19.81%, slightly worse than the baseline system (16.6%) evaluated in [26]. Since both systems modeled floor transitions via VAD and silence thresholds, it's not surprising that they had similar interruption rates. Our vanilla system responded faster than the baseline system in [26] (median 2.7s), making it more likely to interrupt the human.

The interruption rate data violate normality assumptions. We employed the Wilcoxon signed-rank test to assess between-system differences. The floor-instructed system had significantly fewer interruptions than the vanilla system ($W = 28, p = 0.011$). Both systems utilized static prompts. The floor-instructed system's prompt is longer, containing instructions to identify human pauses and either remain silent or produce BC during these pauses. Hence, the reduction in interruption rate quantifies the improvement achievable through prompt engineering alone.

The median interruption rate of the proposed system was lower than the floor-instructed system. Although the p-value was slightly above 0.05 ($W = 38, p = 0.06$), the direction suggests that there is advantage to dynamic prompting. More data is needed to establish the statistical reliability of this effect.

Although the two results are not directly comparable as the tasks were different, the mean interruption rate of our proposed system (7.82%) is close to that reported in [26] (6.9%). However, the response time of our proposed system is much lower (mean $0.986s$ versus mean 1.5s).

In summary, the proposed system exhibits the best turn-taking performance among the three, achieving the lowest interruption rate without increasing response time.

B. Subjective Evaluations

Fig. 7 show box plots of the questionnaire results.

Q3 and Q6 directly evaluate robot interruptions. Compared with the vanilla (green, $W = 14.5, p = 0.025$) and the floor instructed systems (orange, $W = 12, p = 0.009$), the proposed system (blue) is significantly better on Q6. Users made less effort to adapt their speaking style, suggesting that they were able to interact with the system more naturally. The trend on Q3 also favors our proposed system. These results are consistent with the interruption rate results above.

Q4 and Q5 evaluates human interruption of the robot. Our proposed system is significantly better on Q4 than the floor-instructed system ($W = 30, p = 0.041$). The trends favor the proposed system over the other two on Q5.

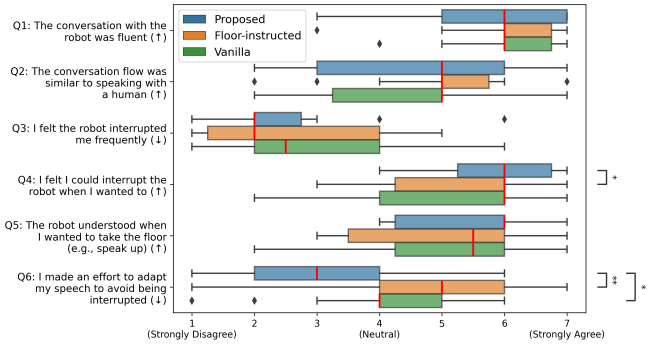

Fig. 7: Box plot of subjective evaluation results. Arrows indicate the preferred direction. Up: higher is better. Down: lower is better. The red line marks the median. Asterisks denote significance ($+$: $p < 0.1$, $*$: $p < 0.05$, $**$: $p < 0.01$). Diamonds represent outliers.

System Type	Vanilla	Floor Instructed	Proposed			
			Total	P1	P2	P3
# of Calls	15.22	15.33	21.05	0.11	8.22	12.72

TABLE III: The average number of calls sent to the audio LLM per conversation. For the proposed system, the calls are further partitioned based on the prompt used.

Q1 and Q2 evaluate the system's overall performance. No trends or statistically significance differences were observed. Intuitively, the fluency and human-likeness of the semi-structured interview may depend more on high-level behavior, rather than low-level turn taking. If this is true, it is perhaps unsurprising that they were not significantly different, since all systems used the same LLM and task descriptions for response generation.

C. Modeling and Signaling of Floor transitions

The average number of LLM calls in each conversation is listed in Table III. The vanilla system and the floor-instructed one had roughly the same number of calls. The proposed system made more calls due to the elaborated state space. The table also shows the average number of times each prompt was used. $P1$ is used only when the human speaks up while the robot is speaking. This happened only twice during the experiment.

Coordinating turn-taking requires estimating the other agent's stance and signaling its perception thereof.

Ideally, to evaluate the accuracy of stance estimation, we need to ask subjects report their stance at the offset of all of their IPUs. Because this would be overly time-consuming, we used human interruptions as a proxy for error, which we identified by a specific IPU pattern following previous studies [12]. As illustrated in Fig. 8, we define a quick human interruption (e_H) as the onset of human IPU shortly after ($t \in W$) the robot assumes the speaking stance. Intuitively, when e_H happens, the robot has *mis-perceived* human stance ($\hat{I}_H = Listening$). The person is trying to re-gain the floor because s/he hasn't finished speaking ($I_H = Speaking$). After W, we assume that the robot holds the floor, so the

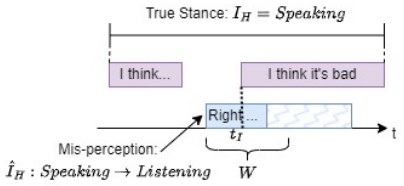

Fig. 8: Illustration of a quick human interruption e_H, which occurs when the onset of human IPU (t_I) falls within W. Human speech shaded purple. Robot speech shaded blue. Zigzag shade indicated canceled speech by the robot.

System Type	Quick Human Interruption	Robot Interruption
Vanilla	31.50%	26.80%
Floor-Instructed	24.47%	26.04%
Proposed	**18.57**%	**10.51**%

TABLE IV: The percentages in the first column is the mean quick interruption rate, computed for each conversation and averaged over conversations. The percentages in the second column is the mean ratio of human quick interruption events (e_H) that were reported by the subject as a robot interruption, again averaged over conversations.

human onset is either backchanneling or an interruption, rather than a signal of mis-perception by the robot.

Both the floor-instructed and the proposed system prompt the LLM to output audio stream only when the human is no longer speaking ($\hat{I}_H : Speaking \rightarrow Listening$). Let e_R denote the event where the model decides to stream audio. The quick human interruption rate, i.e., the number of e_H divided by that of e_R, can then be understood as the error rate of estimating human stance.

The first column of Table IV lists the mean quick human interruption rate averaged over conversations. The vanilla system had the highest rate followed by the floor-instructed system. The proposed system had the lowest rate. These results are consistent with the hypothesis that the proposed system is more accurate at modeling floor transitions.

Stance estimation in human conversation isn't perfect. People signal their perception of the floor via behavioral cues such as backchanneling and acknowledgement. These cues allow the other party to recognize and resolve mis-perceptions smoothly without interrupting the dialogue flow.

The proposed system adopts similar signaling. Typically the robot starts with an acknowledgment (prompt $P2$), followed by paraphrasing/repeating before moving on to the next item. We expect these signals to facilitate the resolution of misperceptions. If so, the proposed system should make the human feel interrupted less often, even if it mis-perceives the human stance.

We calculated the ratio of e_H instances containing a robot IPU later reported to be an interruption by the subject. The mean ratio of all three systems are shown in the second column of Table IV. As expected, compared to the baseline and the floor-instructed system, a smaller portion of e_H from the proposed system evoked a feeling of interruption.

In summary, we found interruption related statistics consistent with the hypothesis that the proposed system estimates human stance more accurately and signals robot perception of the floor more appropriately. However, there are limitations to the use of e_H to indicate robot mis-perception. The human might decide to yield the floor even though the robot's decision to take it was inappropriate. This would cause e_H to under-estimate the mis-perception rate. Some human interruptions could be backchannelling, which would cause over-estimation.

VI. Conclusion

We propose a spoken dialogue system architecture for humanoid robots based on an end-to-end audio LLM. A novel dynamic prompting mechanism based on a 4-state floor transition model coordinates turn-taking. An FSM tracks the state of the conversational floor and composes LLM prompts accordingly. Upon offset or timeout in human utterances, prompts simultaneously update the floor state estimate and generate robot responses. This elaborated modeling of floor state and the signaling thereof based on human knowledge improves the audio LLM's ability to coordinate turn-taking.

There are several limitations of this study. In terms of the experiments, our subject population is small. We only considered one task. Also, we only evaluated the proposed architecture with one audio LLM, GPT4o-realtime. Finally, instead of manual annotation, an IPU-based proxy was used to evaluate the accuracy of stance estimation and the appropriateness of signaling.

Moving forward, the modeling of floor transitions can be further improved. Currently, we use the same P3 prompt for $F_R = Free$ and $F_R = Both$ and always stop robot utterances upon human onset. Ideally, the robot's behavior should differ in these two states. For example, to navigate conversation efficiently in the presence of human interjections, the robot may seek to hold the floor in $F_R = Both$ to push the conversation forward by not stopping its speech.

Future studies can explore how to manage context siz. We observe that GPT4o is more likely to hallucinate as session history grows, especially with repeated prompt change.

The system's signaling of floor state estimate can also be enriched. Currently the LLM selects backchannel cues from a finite set of pre-recordings. Ideally the prosody of the BC should also be context-dependent. Also, BC from a humanoid robot should be multi-modal, including not only speech but also facial expressions and gestures.

References

[1] R. Y. Kim, "Anthropomorphism and Human-Robot Interaction," *Communications of the ACM*, vol. 67, pp. 80–85, Feb. 2024.

[2] "Effects of Anthropomorphism and Accountability on Trust in Human Robot Interaction | Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction."

[3] T. Komatsu *et al.*, "How Does the Difference Between Users' Expectations and Perceptions About a Robotic Agent Affect Their Behavior?" *International Journal of Social Robotics*, vol. 4, pp. 109–116, Apr. 2012.

[4] Y. Shen *et al.*, "A Humanoid Robot Dialogue System Architecture Targeting Patient Interview Tasks," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2024, pp. 1394–1401, iSSN: 1944-9437.

[5] A. Sciutti and G. Sandini, "Interacting With Robots to Investigate the Bases of Social Interaction," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, pp. 2295–2304, Dec. 2017, conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[6] R. Amini *et al.*, "On-Demand Virtual Health Counselor for Delivering Behavior-Change Health Interventions," in *2013 IEEE International Conference on Healthcare Informatics*. Philadelphia, PA, USA: IEEE, Sep. 2013, pp. 46–55.

[7] "ERICA: An Empathetic Android Companion for Covid-19 Quarantine - ACL Anthology."

[8] H. S. Ahn *et al.*, "Hospital Receptionist Robot v2: Design for Enhancing Verbal Interaction with Social Skills," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. New Delhi, India: IEEE Press, Oct. 2019, pp. 1–6.

[9] G. Skantze, "Turn-taking in Conversational Systems and Human-Robot Interaction: A Review," *Computer Speech & Language*, vol. 67, p. 101178, May 2021.

[10] H. H. Clark, *Using language*. Cambridge university press, 1996.

[11] K. Jokinen and M. McTear, *Spoken Dialogue Systems*, ser. Synthesis Lectures on Human Language Technologies. Cham: Springer International Publishing, 2010.

[12] E. Ekstedt and G. Skantze, "Voice Activity Projection: Self-supervised Learning of Turn-taking Events," May 2022, arXiv:2205.09812 [eess].

[13] ——, "TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, 2020, pp. 2981–2990.

[14] "Introducing the Realtime API."

[15] Q. Zhang *et al.*, "OmniFlatten: An End-to-end GPT Model for Seamless Voice Conversation," Jan. 2025, arXiv:2410.17799 [cs].

[16] R. Luo *et al.*, "OpenOmni: Advancing Open-Source Omnimodal Large Language Models with Progressive Multimodal Alignment and Real-Time Self-Aware Emotional Speech Synthesis," Feb. 2025, arXiv:2501.04561 [cs].

[17] A. Défossez *et al.*, "Moshi: a speech-text foundation model for real-time dialogue," Oct. 2024, arXiv:2410.00037 [eess].

[18] L. Mai and J. Carson-Berndsen, "Real-Time Textless Dialogue Generation," Jan. 2025, arXiv:2501.04877 [cs].

[19] A. Raux and M. Eskenazi, "A Finite-State Turn-Taking Model for Spoken Dialog Systems," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, M. Ostendorf *et al.*, Eds. Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 629–637.

[20] S. Duncan, "Some signals and rules for taking speaking turns in conversations," *Journal of Personality and Social Psychology*, vol. 23, pp. 283–292, 1972, place: US Publisher: American Psychological Association.

[21] S. Duncan and G. Niederehe, "On signalling that it's your turn to speak," *Journal of Experimental Social Psychology*, vol. 10, pp. 234–247, 1974, place: Netherlands Publisher: Elsevier Science.

[22] G. Skantze *et al.*, "Exploring Turn-taking Cues in Multi-party Human-robot Discussions about Objects," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: Association for Computing Machinery, Nov. 2015, pp. 67–74.

[23] T.-E. Lin *et al.*, "Duplex Conversation: Towards Human-like Interaction in Spoken Dialogue Systems," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 3299–3308.

[24] A. Raux and M. Eskenazi, "Optimizing the turn-taking behavior of task-oriented spoken dialog systems," *ACM Transactions on Speech and Language Processing*, vol. 9, 2012.

[25] C. Jin *et al.*, "Duplex Conversation in Outbound Agent System," 2021, pp. 4866–4867.

[26] G. Skantze and B. Irfan, "Applying General Turn-taking Models to Conversational Human-Robot Interaction," Jan. 2025, arXiv:2501.08946 [cs].

[27] P. Wang *et al.*, "A Full-duplex Speech Dialogue Scheme Based On Large Language Model," Nov. 2024.

[28] Y. Chen *et al.*, "VoiceBench: Benchmarking LLM-Based Voice Assistants," Dec. 2024, arXiv:2410.17196 [cs].

[29] M. J. Pinto and T. Belpaeme, "Predictive Turn-Taking: Leveraging Language Models to Anticipate Turn Transitions in Human-Robot Dialogue," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, Aug. 2024, pp. 1733–1738, iSSN: 1944-9437.

[30] "Google Gemini."

[31] S. Team, "Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier," 2024, original-date: 2020-11-23T09:54:16Z.

[32] "openai/openai-realtime-console," Mar. 2025, original-date: 2024-09-30T19:00:38Z.

[33] "Humanoid Robot | Robotics Institute - The Hong Kong University of Science and Technology."

[34] K. E. Newcomer *et al.*, Eds., *Handbook of Practical Program Evaluation*, 1st ed. Wiley, Aug. 2015.