π-BENCH: BENCHMARKING PROACTIVE SPEECH INTERACTION WITH FULL-DUPLEX CONVERSATIONAL AI

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex conversational AI promises a shift from turn-based interaction by enabling users and models to speak simultaneously in a temporally fine-grained manner. However, existing evaluations remain largely passive: they mainly test whether models can handle user behaviors such as backchannels, interruptions, topic shifts, or side conversations, while the models are typically rewarded for waiting until the user to finish speaking before responding. This overlooks a key aspect of human conversation: proactively participating during another speaker's turn. Inspired by conversation analysis, we argue that proactive behaviors are not simply interruptions; they help regulate conversational flow, resolve uncertainty, and support cooperative communication. To study this ability, we introduce π-Bench (**p**roactive **i**nteraction benchmark), the first benchmark for evaluating proactive speech interaction in full-duplex conversational AI. π-Bench includes five tasks: turn completion, contextual and non-contextual proactive buzz-in, clarification requests, and backchanneling. Together, these tasks evaluate whether models can provide timely semantic or non-semantic responses, seek clarification when user input is ambiguous, and collaboratively complete a user's unfinished utterance. Experiments on π-Bench show that current full-duplex speech LLMs still struggle with proactive interaction. Although fine-tuning with curated data improves performance, achieving human-level proactive interaction remains an open challenge.

1 INTRODUCTION

Developing artificial intelligence (AI) systems capable of natural spoken conversation with humans has long been a central goal. Traditional modular systems (Young et al., 2013; Vinyals & Le, 2015; Serban et al., 2016) typically rely on multi-stage pipelines consisting of automatic speech recognition (ASR), text-based dialogue systems such as task-oriented dialogue, question answering (QA), and chatbots, text-to-speech (TTS), and additional modules for handling conversational dynamics. Recent advances in speech large language models (LLMs) (Zhang et al., 2023; Tang et al., 2024; Fang et al., 2025; Nguyen et al., 2025) have accelerated the shift toward unified end-to-end spoken conversational modeling, improving both dialogue quality and response latency. More recently, full-duplex speech LLMs (Défossez et al., 2024; Wang et al., 2025; Yu et al., 2025; Cui et al., 2026) have attracted growing attention since their listen-while-speaking capability enables more natural handling of conversational dynamics such as turn-taking, barge-ins, and overlapping speech.

Alongside the rapid progress in conversational AI, many benchmarks have been proposed to evaluate various capabilities of current systems in natural conversations. Some datasets focus primarily on response quality (Chen et al., 2024; Nachmani et al., 2024) or the model's ability to perceive paralinguistic cues (Ao et al., 2024; Lin et al., 2024). Other benchmarks (Lin et al., 2025b;a) evaluate full-duplex interaction capabilities such as turn-taking (Arora et al., 2025; Peng et al., 2025) and barge-in handling (Lin et al., 2026; Zhang et al., 2025; Modi et al., 2026). Although some studies (Lin et al., 2025b) have begun to examine proactive behaviors such as model-initiated backchanneling, existing full-duplex benchmarks still primarily focus on passive behaviors, such as handling user backchannels, interruptions, or side conversations, leaving proactive interaction underexplored.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Evaluating proactive interaction is equally important. Although interjecting during a user's turn is often regarded as impolite or risky, many overlapping behaviors are highly cooperative in natural conversations (Murata, 1994). When a speaker searches for a word, a listener may actively supply the missing expression (Harness Goodwin & Goodwin, 1986; Hayashi, 2003). Listeners may also anticipate upcoming words and collaboratively complete the speaker's utterance, sometimes resulting in unison or choral speech (Lerner, 1991; 2002). Similarly, clarification requests and backchannels allow listeners to resolve uncertainty and maintain conversational flow without waiting for the speaker to produce a fully completed turn.

In this work, we introduce π-Bench to address the lack of systematic evaluation for proactive full-duplex behaviors. Balancing evaluation controllability, cross-model transferability, and real-world relevance, we design five tasks. Turn completion evaluates whether a model can collaboratively supply a word or phrase that completes the user's unfinished utterance. Non-contextual and contextual proactive buzz-in evaluate whether a model can respond once sufficient semantic or contextual information becomes available. Clarification requests evaluate whether it seeks missing information instead of guessing. Backchanneling evaluates whether it provides brief feedback at appropriate moments. Extensive experiments reveal substantial limitations in current models and provide guidance for the future development of full-duplex conversational AI.

Our contributions are threefold:

- We introduce π-Bench, the first benchmark for evaluating conversational proactive interaction in full-duplex conversational AI through five tasks covering turn completion, proactive buzz-in, clarification requests, and backchanneling.
- We propose novel evaluation criteria and methods for backchannel, clarification, and turn completion, assessing both the timing and content of model-initiated conversational behaviors with human agreement analysis.
- We benchmark current full-duplex speech LLMs and reveal substantial room for improvement in proactive interaction.

2 Related Work

Many benchmarks have been proposed to evaluate conversational AIs across various dimensions. Some studies (Chen et al., 2024; Nachmani et al., 2024) focus on the factual correctness and quality of model responses, while others (Ao et al., 2024; Lin et al., 2024) examine whether models can accurately understand paralinguistic information such as emotion and tone. Several recent benchmarks evaluate response safety and contextual consistency (Zhang et al., 2025), while others focus on full-duplex interaction capabilities (Arora et al., 2025; Lin et al., 2025b; 2026; 2025a; Peng et al., 2025; Ray et al., 2026). Our π-Bench belongs to the latter category. However, unlike prior work that primarily examines turn-taking (Arora et al., 2025; Lin et al., 2025b), interruptions (Lin et al., 2026; Ray et al., 2026), or rejection (Lin et al., 2025a; Peng et al., 2025), our benchmark focuses on model-initiated proactive interaction. We evaluate whether models can respond before a question ends, request clarification when crucial information is missing, provide backchannels at appropriate moments, and collaboratively complete a user's unfinished utterance. Although model backchanneling is considered in Full-Duplex-Bench (Lin et al., 2025b), we evaluate its timing relative to transition-relevance places, enabling a more fine-grained assessment of when backchannels occur. Together, these tasks extend full-duplex evaluation from handling user-initiated conversational dynamics to actively participating in an ongoing user turn.

3 π-Bench

Evaluating proactive interaction in natural conversations is challenging because appropriate responses and their timing are difficult to control. Inspired by conversation analysis, we identify four representative proactive behaviors and design five corresponding tasks. **Turn completion** evaluates whether a model can supply a fitting word or phrase that collaboratively completes the user's utterance. We further extend this task to **contextual turn completion** to assess whether explicit context improves model performance. **Proactive buzz-in** evaluates whether a model responds once sufficient information becomes available. We further introduce **contextual proactive buzz-in** to test whether the

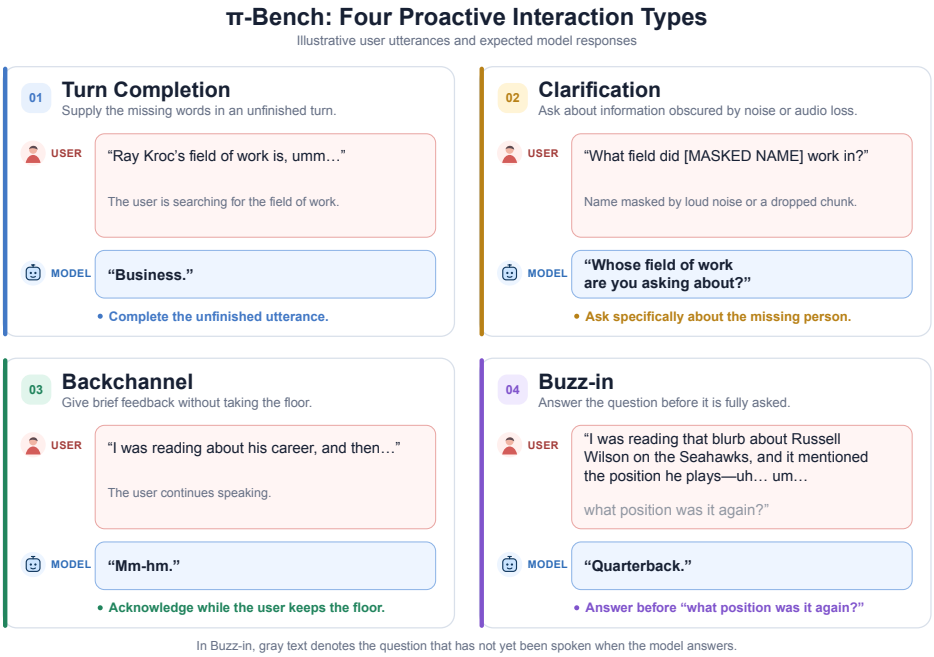


Figure 1: Four interaction types in π-Bench: turn completion supplies words to finish the speaker's utterance; clarification requests recover obscured information; backchannels acknowledge the speaker without taking the floor; and buzz-in answers an emerging question. The benchmark evaluates buzz-in with and without supplied context as separate tasks, yielding five tasks in total.

response remains faithful to shared counterfactual context. **Clarification requests** evaluate whether a model actively seeks missing information when part of the user's speech is unclear. **Backchanneling** evaluates whether a model provides brief feedback at appropriate moments without taking over the turn. These tasks present proactive interaction as a unified capability: the model must recognize an opportunity within an ongoing user turn, select an appropriate conversational behavior, and produce a timely response. Table 1 summarizes the data used for each task, and the following sections describe their construction in detail.

Table 1: Task statistics for π-Bench.

Task	Source data	Synthetic	# Samples
Turn Completion	Switchboard (Godfrey & Holliman, 1993), Seamless Interaction (Agrawal et al., 2025)	×	416
Contextual Turn Completion	ConFiQA (Bi et al., 2025)	✓	147
Proactive Buzz-in	TriviaQA (Joshi et al., 2017), Web Questions (Berant et al., 2013)	✓	401
Contextual Proactive Buzz-in	FaithEval (Ming et al., 2025), ConFiQA (Bi et al., 2025)	✓	361
Clarification requests	TriviaQA (Joshi et al., 2017), Web Questions (Berant et al., 2013)	✓	657
Backchannel	Switchboard (Godfrey & Holliman, 1993)	×	304

3.1 Turn Completion

Turn completion is a common form of cooperative participation in human conversation. When a speaker has not yet completed an utterance, the listener may anticipate and supply a fitting

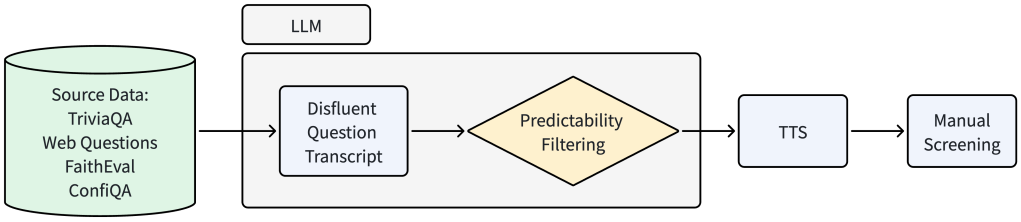


Figure 2: Buzz-in construction places sufficient answer cues before a hesitation marker, creating a reference point for testing whether a model answers an emerging question early.

word or phrase. Such responses help interlocutors jointly construct the utterance and can make the interaction feel more fluent and collaborative (Lerner, 1991; 2002). We therefore evaluate whether full-duplex models can recognize an appropriate opportunity and produce a continuation that matches the speaker's intended meaning. We construct the evaluation data from both controlled and natural conversations. For the controlled setting, we use examples from ConFiQA (Bi et al., 2025) to generate short dialogues to gauge whether models can perform turn completion given the specific instruction to do so, under clear context. The utterances are synthesized with natural hesitation markers, and each example is checked for contextual faithfulness, answer alignment, and conversational naturalness. For the natural setting, we curate turn-completion examples from Switchboard and Seamless Interaction (Agrawal et al., 2025). Human annotators verify whether the listener's contribution forms an appropriate continuation of the user's utterance. Together, these data allow us to evaluate both whether the model attempts a turn completion and whether the proposed completion is correct.

3.2 Proactive buzz-in

Humans do not always wait until the speaker has completely fallen silent before responding. When sufficient semantic or contextual information is available, a listener may be able to respond while the speaker is still talking. Word search is a common example: a speaker may get stuck on a particular keyword or actively seek help from the listener. If a model always waits for the user to finish in such situations, the conversational experience may be significantly degraded. Inspired by this observation, we design the proactive buzz-in task.

Non-contextual proactive buzz-in We rewrite question-answering items as spoken questions with a hesitant lead-in. For example, "Which Soviet leader was awarded the Nobel Peace Prize in 1990?" becomes "I'm trying to remember that Soviet leader who got the Nobel Peace Prize in 1990, for the whole perestroika and glasnost era, um... um... what's his name again?" The question must contain enough information to identify the answer before the first filler. That filler marks a reference point for measuring whether the model answers before the speaker finishes asking.

Contextual proactive buzz-in When sufficient context is available, humans also use dialogue history to respond proactively. To evaluate this ability without introducing long spoken histories, we provide both the user and model with a shared text passage. We deliberately select counterfactual passages so that the model must remain faithful to the supplied context rather than rely solely on its parametric knowledge. The model is expected to identify the answer from the context and respond once the user's intended question becomes clear.

As shown in Figure 2, after obtaining the LLM-generated questions, we synthesize them into audio using a TTS model. Finally, we manually inspect the samples and retain those that satisfy the task requirements for inclusion in the test set.

3.3 Clarification requests

The clarification-request task evaluates conversational repair when crucial speech content is unclear. In spoken human–machine interaction, background noise, transmission latency, or packet loss may

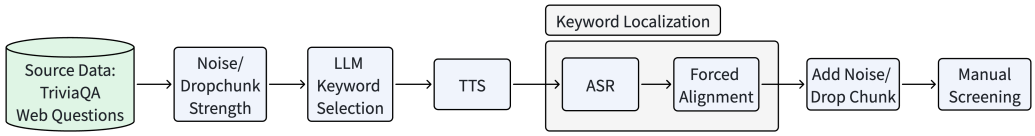


Figure 3: Clarification construction masks a question's key expression at controlled levels, allowing evaluation of whether the model requests the missing information.

obscure part of a user's speech. Rather than guessing, a model should request clarification. We therefore mask a keyword in each spoken question with noise or packet-loss-like dropouts and test whether the model proactively requests the missing information.

As shown in Figure 3, we use an LLM to identify the keyword in each QA example and synthesize the question using TTS, with speaker reference audio randomly selected from LibriHeavy (Kang et al., 2024). We transcribe the synthesized speech with GPT-4o Transcribe and use NeMo forced alignment to locate the keyword. We then corrupt its audio region using either a noise segment from the DNS Noise Challenge dataset (Dubey et al., 2024) or DropChunk. Both methods have three masking levels: *mild*, which largely preserves comprehension; *medium*, which motivates clarification while permitting an uncertain guess; and *hard*, which makes the keyword effectively unrecoverable.

For DNS-noise masking, the mild, medium, and hard levels use SNRs of 0, -10, and $-20\,\mathrm{dB}$, respectively. We retain only noise segments whose root mean square exceeds an empirical threshold, excluding effectively silent regions. For DropChunk masking, the three levels remove a single $45\,\mathrm{ms}$ segment, approximately 22.5%–47.5% of the keyword duration, and the entire keyword region, respectively. Human annotators verify transcript fidelity and confirm that only the intended keyword is masked at the designated difficulty level.

3.4 Backchannel

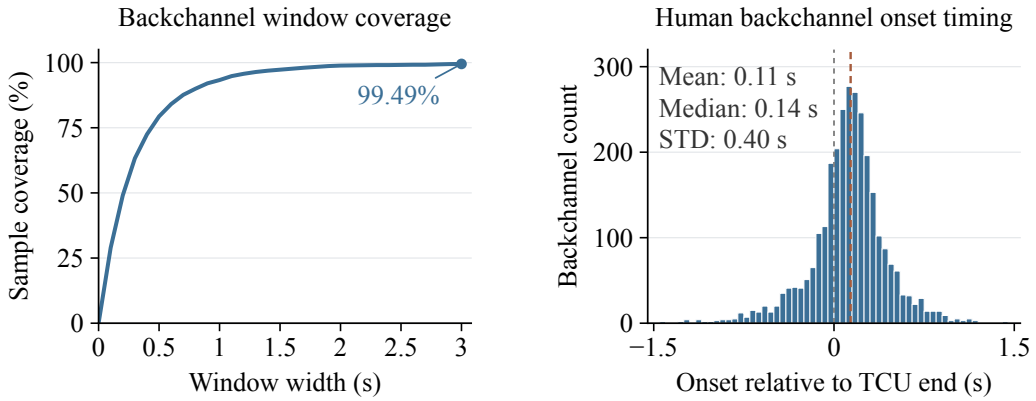
Backchanneling plays an important role in maintaining conversational flow and signaling that the listener is following the speaker. Full-Duplex-Bench (Lin et al., 2025b) evaluates this ability using measures such as backchannel frequency and the Jensen–Shannon divergence between model and reference timing distributions. In this work, we further incorporate transition-relevance places (TRPs), which refer to points at which an utterance reaches possible syntactic, prosodic, semantic, or pragmatic completion (Sacks et al., 1974; Kendrick et al., 2023). Since listeners tend to project these points, TRPs provide a natural reference for evaluating when a backchannel occurs.

We use the time-aligned NXT Switchboard annotations (Calhoun et al., 2010; Stolcke et al., 2000), where slash units serve as an operational proxy for TRPs (Meteer et al., 1995). We select conversations with one primary speaker and listener backchannels, yielding 1,584 candidate segments and approximately 8.8 hours of speech. Human annotators further select 304 samples with high-quality primary-speaker recordings for evaluation. As shown in Figure 4, human backchannels are strongly concentrated around TRPs: 99.49% occur within 1.5 seconds of the nearest TRP in a 3-second window. We therefore evaluate both how frequently a model produces backchannels and how closely their timing follows the human distribution.

4 Experimental Setups

4.1 Data specifications

For the contextual turn completion task, we use `GPT-5.2-2025-12-11` to generate an utterance with a designated cut point and `Qwen3-TTS-12Hz-1.7B-Base` (Hu et al., 2026) to synthesize the corresponding audio. For the proactive buzz-in task, we use TriviaQA (Joshi et al., 2017) and Web Questions (Berant et al., 2013) as source data. For the contextual proactive buzz-in task, the source data are FaithEval-counterfactual (Ming et al., 2025) and ConFiQA (Bi et al., 2025). We rewrite the original questions into a word-search format by prompting `GPT-5.2-2025-12-11`; the specific prompts used are provided in Appendix B. For the questions rewritten by the LLM, we



(a) Backchanneling Coverage by window width

(b) Human backchannel onset offsets

Figure 4: Human backchannels cluster near TRP-proxy boundaries. Window coverage (a) motivates the reference range; onset offsets (b) form the human timing distribution used for JSD.

synthesize the corresponding audio using OpenAI's commercial TTS model `tts-1-hd`. For the clarification task, we use TriviaQA (Joshi et al., 2017) and Web Questions (Berant et al., 2013) and employ Doubao-1.6-lite[1] to select the critical keyword in each question.

4.2 Model specifications

We evaluate Moshi (Défossez et al., 2024), Freeze-Omni (Wang et al., 2025), SALMONN-omni (Yu et al., 2025), MiniCPM-o-4.5 (Cui et al., 2026), Gemini Live, GPT Realtime, and Doubao speech-to-speech where supported by the task and input format. We report default, unprompted results for all models. These systems differ in streaming architecture and audio processing; our comparisons concern observed conversational behavior, not a controlled architectural ablation. For MiniCPM-o-4.5, we evaluate speech interaction only.

4.3 Evaluation metrics

All tasks are evaluated in a streaming conversational setting in which models may respond while still receiving user audio. We transcribe the model outputs, obtain timestamps for the transcribed words, and use an LLM judge to assess task-specific response content. For **turn completion**, we evaluate whether the first response beginning after the end of the first filler collaboratively completes the user's unfinished utterance. We weight this response according to its delay from the end of the filler: responses within $0.4\,\mathrm{s}$ receive full weight, after which the weight decreases linearly to zero at $5\,\mathrm{s}$. We report the turn-completion rate, timing score, turn-completion score, and answer accuracy. For **proactive buzz-in**, with or without context, we measure the onset of the earliest correct response relative to the onset of the first filler, with negative offsets indicating earlier responses. A turn-decayed reward assigns a score of 1 to a correct answer in the first response turn, 0.5 in the second turn, and 0 in any subsequent turn, thereby penalizing premature incorrect guesses. We additionally report later-turn accuracy, defined as the proportion of examples answered correctly in the second or a subsequent turn. For **clarification requests**, we concatenate the model's response turns and determine whether the combined response directly answers the question or requests clarification about the obscured keyword. We report the clarification-request rate r_{CLA}, targeted-clarification accuracy $\mathrm{acc}_{\mathrm{CLA}}$, and answer accuracy among non-clarification responses $\mathrm{acc}_{\mathrm{ANS}}$. We combine these metrics into the composite score $S_{\mathrm{CLA}} = r_{\mathrm{CLA}}\,\mathrm{acc}_{\mathrm{CLA}} + (1 - r_{\mathrm{CLA}})\,\mathrm{acc}_{\mathrm{ANS}}$. For **backchanneling**, we identify backchannel utterances and evaluate their occurrence and timing relative to transition-relevance-place (TRP) proxies. Backchannel Score measures the fraction of all model utterances judged to be backchannels within the $\pm 1.5\,\mathrm{s}$ TRP window, while backchannel frequency measures the number of such events per TRP. TRP-relative timing JSD compares the model and human distributions of backchannel

[1] `https://seed.bytedance.com/en/seed1_6`

onset times and is undefined when a model produces no backchannels. Detailed evaluation protocols, metric definitions, judge prompts, and human–judge agreement analyses are provided in Appendix A.

5 Experimental Results

5.1 Turn completion

Table 2 presents the natural-data results together: 348 Switchboard inputs and 68 Seamless Interaction inputs. Each numerator counts outputs identified as turn completions, not whether their content matched the speaker's intended expression. Turn completions remain rare in natural conversational data. Although explicit prompting increases their frequency for most evaluated models, the absolute gains are small. These results suggest that current models struggle to recognize and perform turn completion in realistic conversational settings. One possible explanation is limited retention over long contexts: because the completion opportunity occurs after an extended conversation, models may no longer retain sufficient information to infer the intended completion. We investigate this hypothesis in the next subsection through the contextual turn-completion task.

Table 2: Turn-completion occurrence with and without the system prompt on the combined Switchboard and Seamless Interaction data. The evaluation contains 416 inputs; rates measure occurrence, not content correctness.

Model	Without Prompt	With Prompt
Moshi	2/416 (0.48%)	–
Freeze-Omni	5/416 (1.20%)	9/416 (2.16%)
MiniCPM-o-4.5	15/416 (3.61%)	20/416 (4.81%)
SALMONN-omni	0/416 (0.00%)	0/416 (0.00%)

5.2 Contextual Turn Completion

Contextual turn completion evaluates whether models can produce turn completions when explicitly instructed to do so, both with and without supporting text. Models generally produce more turn completions when context is provided, suggesting that greater confidence in the relevant information encourages this behavior. Among model responses that become correct after text context is added, 77%–97% are realized as turn completions, with a median of 95.1% and a mean of 91.3%. This pattern is consistent with the results in Table 3. Among the open-source models, MiniCPM-o performs best without context across all four metrics. With text context, MiniCPM-o retains the highest timing score and accuracy and ties SALMONN-omni for the highest turn-completion rate. However, SALMONN-omni benefits more from the added context and achieves the highest turn-completion score. Among the closed-source models, Gemini Live achieves the highest turn-completion score without context, while GPT Realtime obtains the highest turn-completion rate and accuracy. With text context, Doubao S2S produces more timely turn completions and achieves the highest turn-completion score and accuracy, although GPT Realtime retains the highest turn-completion rate.

5.3 Proactive buzz-in

Non-contextual proactive buzz-in Table 4 shows negative mean response offsets for Moshi, Freeze-Omni, and SALMONN-omni, indicating that their earliest correct answers precede the reference filler onset on average. The other models have positive mean offsets; Doubao S2S achieves the highest GPT reward (0.955), while Gemini Live and GPT Realtime have the highest later-turn accuracy (0.726 and 0.712). These offsets are measured relative to the filler, not the end of the user's question, so they do not by themselves establish whether a model spoke before the user finished. Differences in streaming chunk size may affect timing, but this table alone cannot isolate that effect.

Contextual proactive buzz-in Moshi is not included because the evaluated interface does not support the shared text-context setup. For all six included models, Table 5 shows higher reward with supplied context than without it. SALMONN-omni reaches the highest contextual reward (0.717), while Gemini Live has the highest contextual later-turn accuracy (0.759). This comparison

Table 3: Turn-completion performance with and without text context. Each setting reports the turn-completion rate r_{TC}, timing score S_{time}, turn-completion score S_{TC}, and accuracy acc. The best result within each model group is shown in bold.

Model	**Without Context**				**With Context**			
	r_{TC}	S_{time}	S_{TC}	acc	r_{TC}	S_{time}	S_{TC}	acc
Moshi	7.5%	26.0%	3.9%	3.4%	–	–	–	–
Freeze-Omni	12.2%	13.7%	9.5%	4.8%	21.8%	22.4%	18.1%	15.6%
SALMONN-omni	25.9%	49.3%	21.9%	8.2%	**42.9%**	51.5%	**37.9%**	37.4%
MiniCPM-o-4.5	**39.5%**	**57.4%**	**28.0%**	**12.2%**	**42.9%**	**52.1%**	33.6%	**41.5%**
Gemini Live	65.3%	38.0%	**22.8%**	27.9%	89.1%	38.1%	34.7%	67.3%
GPT Realtime	**67.3%**	3.8%	2.8%	**35.4%**	**89.8%**	5.7%	5.5%	73.5%
Doubao S2S	42.2%	**40.5%**	18.5%	29.9%	83.0%	**61.9%**	**51.7%**	**81.0%**

Table 4: Performance on the non-contextual proactive buzz-in task.

Model	ΔT	**GPT Reward**	**Later-turn Accuracy** (≥ 2)
Moshi	$-1.06\,\mathrm{s}$	0.183	0.173
Freeze-Omni	$-2.52\,\mathrm{s}$	0.367	0.000
SALMONN-omni	$-2.44\,\mathrm{s}$	0.575	0.047
MiniCPM-o-4.5	$+1.11\,\mathrm{s}$	0.586	0.000
Gemini Live	$+1.51\,\mathrm{s}$	0.744	0.726
GPT Realtime	$+1.64\,\mathrm{s}$	0.748	0.712
Doubao S2S	$+1.70\,\mathrm{s}$	**0.955**	0.000

supports context use in this setting, but does not determine whether the model would use a spoken conversational history in the same way. Freeze-Omni has the lowest contextual reward despite a negative mean response offset, illustrating that early timing alone does not guarantee a correct context-grounded answer.

5.4 Clarification requests

Table 6 shows that clarification requests are generally infrequent among the default models, though Gemini Live reaches a request rate of 0.420 on hard inputs and GPT Realtime reaches a hard-input clarification accuracy of 0.414. Requesting clarification does not guarantee a useful repair: the clarification-accuracy columns distinguish requests about the obscured keyword from other responses. The composite score is highest for Doubao S2S on mild (0.875) and medium inputs (0.789), and for fine-tuned SALMONN-omni on hard inputs (0.848). The latter requests clarification much more often on medium and hard inputs (0.893 and 0.959), with a hard-input clarification accuracy of 0.769, while its answer accuracy falls as corruption increases.

5.5 Backchannel

As shown in the Table 7 GPT Realtime has the best backchanneling performance, under every setting, and Moshi has the best backchannel score as in the open-source model without prompt, but miniCPM becomes better after system prompt. Overall, almost every model understand backchannel to some extent that after prompted to only backchannel, they achieved better performance in backchannel, with DoubaoS2S as an exception.

6 Conclusion

π-Bench evaluates five distinct ways a model may contribute during an unfolding user turn: two forms of early answering, clarification, backchanneling, and turn completion. This task structure separates speaking early from speaking helpfully. The available results show uneven early-answer

Table 5: Performance on the contextual proactive buzz-in task with and without counterfactual context. We report response offset ΔT, GPT reward R_{GPT}, and later-turn accuracy $\mathrm{acc}_{\geq 2}$.

Model	**Without Context**			**With Context**		
	ΔT	R_{GPT}	$\mathrm{acc}_{\geq 2}$	ΔT	R_{GPT}	$\mathrm{acc}_{\geq 2}$
Freeze-Omni	$-4.12\,\mathrm{s}$	0.014	0.000	$-3.15\,\mathrm{s}$	0.247	0.000
SALMONN-omni	$+0.27\,\mathrm{s}$	0.048	0.250	$+1.23\,\mathrm{s}$	0.717	0.000
MiniCPM-o-4.5	$+1.59\,\mathrm{s}$	0.072	0.000	$+0.79\,\mathrm{s}$	0.680	0.000
Gemini Live	$+1.74\,\mathrm{s}$	0.050	0.049	$+1.66\,\mathrm{s}$	0.620	0.759
GPT Realtime	$+1.78\,\mathrm{s}$	0.043	0.034	$+1.71\,\mathrm{s}$	0.511	0.455
Doubao S2S	$+1.90\,\mathrm{s}$	0.055	0.000	$+1.75\,\mathrm{s}$	0.524	0.000

Table 6: Performance on the clarification-request task. For each difficulty level, we report the clarification-request rate r_{CLA}, clarification-object accuracy $\mathrm{acc}_{\mathrm{CLA}}$, answer accuracy among non-clarification responses $\mathrm{acc}_{\mathrm{ANS}}$, and final score $S_{\mathrm{CLA}} = r_{\mathrm{CLA}}\,\mathrm{acc}_{\mathrm{CLA}} + (1 - r_{\mathrm{CLA}})\,\mathrm{acc}_{\mathrm{ANS}}$. The best S_{CLA} within each model group is shown in bold.

Model	**Mild**				**Medium**				**Hard**			
	r_{CLA}	$\mathrm{acc}_{\mathrm{CLA}}$	$\mathrm{acc}_{\mathrm{ANS}}$	S_{CLA}	r_{CLA}	$\mathrm{acc}_{\mathrm{CLA}}$	$\mathrm{acc}_{\mathrm{ANS}}$	S_{CLA}	r_{CLA}	$\mathrm{acc}_{\mathrm{CLA}}$	$\mathrm{acc}_{\mathrm{ANS}}$	S_{CLA}
Moshi	0.000	–	0.140	0.140	0.004	0.000	0.065	0.064	0.005	0.000	0.064	0.064
Freeze-Omni	0.033	0.538	0.454	0.457	0.120	0.276	0.288	0.286	0.173	0.118	0.151	0.145
SALMONN-omni	0.003	0.000	0.645	**0.643**	0.004	0.000	0.510	**0.508**	0.023	0.200	0.277	**0.275**
MiniCPM-o-4.5	0.020	0.625	0.479	0.482	0.017	0.375	0.329	0.330	0.059	0.231	0.242	0.241
Gemini Live	0.056	0.909	0.630	0.645	0.139	0.731	0.533	0.560	0.227	0.510	0.403	0.427
GPT Realtime	0.008	1.000	0.779	0.781	0.052	0.680	0.613	0.616	0.207	0.418	0.490	**0.475**
Doubao S2S	0.015	0.833	0.816	**0.816**	0.015	0.429	0.745	**0.741**	0.048	0.286	0.468	0.459

accuracy, infrequent targeted clarification, and rare turn-completion outputs on the Switchboard and Seamless Interaction subsets. Backchannel results under the final two-metric protocol and controlled synthetic turn-completion results remain pending. Consequently, the present evidence supports a benchmark and an initial behavioral comparison, not a general claim that any evaluated model has mastered proactive conversation.

Table 7: Backchannel performance with and without the system prompt.

Model	Backchannel Score		Backchannel Frequency		TRP-relative Timing JSD	
	Without	With	Without	With	Without	With
Moshi	**27.67%**	–	**12.43%**	–	0.4975	–
Freeze-Omni	1.11%	4.80%	0.27%	0.72%	0.8741	0.7255
SALMONN-omni	6.20%	21.45%	1.67%	5.87%	**0.4157**	0.2975
MiniCPM-o-4.5	2.29%	**40.66%**	0.41%	**7.77%**	0.6935	**0.2394**
Gemini Live	23.67%	38.18%	9.26%	14.37%	0.5909	0.6050
GPT Realtime	**36.78%**	**49.00%**	**18.35%**	**25.49%**	**0.5125**	**0.5509**
Doubao S2S	7.71%	7.16%	1.67%	1.49%	0.6894	0.7529
Human reference	–	–	23.77%	23.77%	0.0000	0.0000

7 AI USE STATEMENT

In this work, we used generative AI tools to construct synthetic benchmark data, including generating conversational utterances, identifying keywords for controlled masking, and synthesizing speech. We also used AI tools for speech transcription and LLM-based evaluation of model responses, as detailed in the main text and Appendix B. Generated samples were manually inspected, and agreement between automated judgments and human annotations was assessed for turn completion, clarification, and backchanneling. The authors take full responsibility for the final content of this work, including its data, evaluation procedures, and reported findings.

REFERENCES

Vasu Agrawal, Akinniyi Akinyemi, Kathryn Alvero, Morteza Behrooz, Julia Buffalini, Fabio Maria Carlucci, Joy Chen, Junming Chen, Zhang Chen, Shiyang Cheng, Praveen Chowdary, Joe Chuang, Antony D'Avirro, Jon Daly, Ning Dong, Mark Duppenthaler, Cynthia Gao, Jeff Girard, Martin Gleize, Sahir Gomez, Hongyu Gong, Srivathsan Govindarajan, Brandon Han, Sen He, Denise Hernandez, Yordan Hristov, Rongjie Huang, Hirofumi Inaguma, Somya Jain, Raj Janardhan, Qingyao Jia, Christopher Klaiber, Dejan Kovachev, Moneish Kumar, Hang Li, Yilei Li, Pavel Litvin, Wei Liu, Guangyao Ma, Jing Ma, Martin Ma, Xutai Ma, Lucas Mantovani, Sagar Miglani, Sreyas Mohan, Louis-Philippe Morency, Evonne Ng, Kam-Woh Ng, Tu Anh Nguyen, Amia Oberai, Benjamin Peloquin, Juan Pino, Jovan Popovic, Omid Poursaeed, Fabian Prada, Alice Rakotoarison, Rakesh Ranjan, Alexander Richard, Christophe Ropers, Safiyyah Saleem, Vasu Sharma, Alex Shcherbyna, Jia Shen, Jie Shen, Anastasis Stathopoulos, Anna Sun, Paden Tomasello, Tuan Tran, Arina Turkatenko, Bo Wan, Chao Wang, Jeff Wang, Mary Williamson, Carleigh Wood, Tao Xiang, Yilin Yang, Julien Yao, Chen Zhang, Jiemin Zhang, Xinyue Zhang, Jason Zheng, Pavlo Zhyzheria, Jan Zikes, and Michael Zollhoefer. Seamless Interaction: Dyadic Audiovisual Motion Modeling and Large-Scale Dataset. *arXiv preprint arXiv:2506.22554*, 2025.

Diogo Almeida. Introducing system one models & Jev. TypeSafe AI Blog, September 2026. URL `https://typesafe.ai/blog/introducing-system-one-models-and-jev`. Accessed September 26, 2026.

Junyi Ao, Yuancheng Wang, Xiaohai Tian, Dekun Chen, Jun Zhang, Lu Lu, Yuxuan Wang, Haizhou Li, and Zhizheng Wu. SD-Eval: A benchmark dataset for spoken dialogue understanding beyond words. In *Proc. NeurIPS*, Vancouver, 2024.

Siddhant Arora, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, and Shinji Watanabe. Talking turns: Benchmarking audio foundation models on turn-taking dynamics. *arXiv preprint arXiv:2503.01174*, 2025.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proc. EMNLP*, Seattle, 2013.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. Context-DPO: Aligning language models for context-faithfulness. In *Proc. ACL Findings*, Vienna, 2025.

Sasha Calhoun, Jean Carletta, Jason M Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4):387–419, 2010.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. VoiceBench: Benchmarking llm-based voice assistants. *arXiv preprint arXiv:2410.17196*, 2024.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. MiniCPM-o-4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: A speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, Babak Naderi, Sebastian Braun, Ross Cutler, Alex Ju, Mehdi Zohourian, Min Tang, Mehrsa Golestaneh, et al. Icassp 2023 deep noise suppression challenge. *IEEE Open Journal of Signal Processing*, 5:725–737, 2024.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-Omni: Seamless speech interaction with large language models. In *Proc. ICLR*, Singapore, 2025.

John J. Godfrey and Edward Holliman. Switchboard-1 release 2. Web Download, 1993. LDC Catalog No. LDC97S62.

Marjorie Harness Goodwin and Charles Goodwin. Gesture and coparticipation in the activity of searching for a word. 1986.

Makoto Hayashi. Language and the body as resources for collaborative action: A study of word searches in japanese conversation. *Research on language and social interaction*, 36(2):109–141, 2003.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-TTS Technical Report. *arXiv preprint arXiv:2601.15621*, 2026.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proc. ACL*, Vancouver, 2017.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. In *Proc. ICASSP*, Seoul, 2024.

Kobin H Kendrick, Judith Holler, and Stephen C Levinson. Turn-taking in human face-to-face interaction is multimodal: gaze direction and manual gestures aid the coordination of turn transitions. *Philosophical Transactions of The Royal Society B*, 378(1875):20210473, 2023.

Gene H Lerner. On the syntax of sentences-in-progress. *Language in society*, 20(3):441–458, 1991.

Gene H Lerner. Turn-sharing. *The language of turn and sequence*, pp. 225–256, 2002.

Guan-Ting Lin, Cheng-Han Chiang, and Hung yi Lee. Advancing large language models to capture varied speaking styles and respond properly in spoken conversations. In *Proc. ACL*, Bangkok, 2024.

Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. Full-Duplex-Bench-v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner. *arXiv preprint arXiv:2510.07838*, 2025a.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*, 2025b.

Guan-Ting Lin, Shih-Yun Shan Kuan, Qirui Wang, Jiachen Lian, Tingle Li, Shinji Watanabe, and Hung-yi Lee. Full-duplex-bench v1.5: Evaluating overlap handling for full-duplex speech models. In *Proc. ICASSP*, Barcelona, 2026.

Marie Meteer et al. Dysfluency annotation stylebook for the switchboard corpus. Technical report, Technical report, 1995.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. FaithEval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *Proc. ICLR*, Singapore, 2025.

Smit Nautambhai Modi, Gandharv Mahajan, Marc Wetter, and Randall Welles. Echochain: A full-duplex benchmark for state-update reasoning under interruptions. *arXiv preprint arXiv:2604.16456*, 2026.

Kumiko Murata. Intrusive or co-operative? a cross-cultural study of interruption. *Journal of Pragmatics*, 21(4):385–400, 1994.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken question answering and speech continuation using spectrogram-powered LLM. In *Proc. ICLR*, Vienna, 2024.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-LM: Interleaved spoken and written language model. *TACL*, 13:30–52, 2025.

Yizhou Peng, Yi-Wen Chao, Dianwen Ng, Yukun Ma, Chongjia Ni, Bin Ma, and Eng Siong Chng. Fd-bench: A full-duplex benchmarking pipeline designed for full duplex spoken dialogue systems. *arXiv preprint arXiv:2507.19040*, 2025.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik Narasimhan. τ-Voice: Benchmarking full-duplex voice agents on real-world domains. *arXiv preprint arXiv:2603.13686*, 2026.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, 1974.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. AAAI*, 2016.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Dan Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3): 339–374, 2000.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. SALMONN: Towards generic hearing abilities for large language models. In *Proc. ICLR*, Vienna, 2024.

Oriol Vinyals and Quoc V. Le. A neural conversational model. In *Proc. ICML Deep Learning Workshop*, 2015.

Xiong Wang, Yangze Li, Chaoyou Fu, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-Omni: A smart and low latency speech-to-speech dialogue model with frozen LLM. In *Proc. ICML*, Vancouver, 2025.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. SALMONN-omni: A standalone speech LLM without codec injection for full-duplex conversation. In *Proc. NeurIPS*, San Diego, 2025.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Proc. EMNLP Findings*, Singapore, 2023.

He Zhang, Wenqian Cui, Haoning Xu, Xiaohui Li, Lei Zhu, Haoli Bai, Shaohua Ma, and Irwin King. Mtr-duplexbench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models. *arXiv preprint arXiv:2511.10262*, 2025.

A Details of the Proposed Metrics

All our evaluations are conducted in a conversational setting. While listening to the user's audio, the model is allowed to freely generate one or more response turns. If the user audio has been fully consumed but the model is still responding, silence is dynamically appended until the model completes the current turn. We transcribe model response audio with GPT-4o Transcribe and apply NeMo forced alignment to obtain timestamps for individual words. For open-source models, we group adjacent words when the gap between them is less than 1 second and split them into separate utterances when the gap is greater than 1 second. For closed-source models, we instead segment responses by the intents returned by each model. When evaluating response content, we use `Jev` (Almeida, 2026) as the judge. The prompts are provided in Appendix B. Below we introduce the task-specific evaluation metrics.

Turn Completion We use an LLM judge to determine whether a model response collaboratively completes the speaker's unfinished turn. To incorporate response timing, we define a time-decay weight based on observed human responses. Because both the mean and median human turn-completion delays are approximately $400\,\mathrm{ms}$, responses produced within $0.4\,\mathrm{s}$ receive full weight. The weight then decreases linearly to zero at $5\,\mathrm{s}$. Although the latest observed human completion occurred after approximately $3\,\mathrm{s}$, we extend the window to $5\,\mathrm{s}$ to accommodate potential network latency. We consider only the first model response beginning after the end of the first filler. Let t denote the interval, in seconds, between the end of the filler and the onset of this response. The weight is

$$w(t) = \begin{cases} 1, & 0 \le t \le 0.4, \\ (5-t)/4.6, & 0.4 < t < 5, \\ 0, & t \ge 5. \end{cases} \tag{1}$$

We report three turn-completion metrics: turn-completion rate r_{TC}, timing score S_{time}, and time-weighted turn-completion score S_{TC}. For contextual turn completion, we additionally report answer accuracy acc, which measures whether the completion content is correct.

We also evaluate agreement between the LLM judge and human annotations for turn completion using 91 samples. As shown in Table 8, the judge achieves an accuracy of 73.6%, a TC F1 score of 70.7%, and a non-TC F1 score of 76.0%. The corresponding 95% confidence intervals are 64.9%–81.8%, 58.5%–80.5%, and 65.9%–84.3%, respectively. These results indicate reasonably strong agreement between the LLM judge and human annotators.

Table 8: Turn-completion judge agreement with 95% confidence intervals.

Metric	Result	95% CI
Accuracy	**73.6%**	64.9%–81.8%
TC F1	**70.7%**	58.5%–80.5%
Non-TC F1	**76.0%**	65.9%–84.3%

Proactive buzz-in For both non-contextual and contextual proactive buzz-in, we evaluate the model mainly from two perspectives: response timing and content. First, we iterate through each response turn generated by the model and use an LLM to determine whether the response is correct or consistent with the context. We then take the start time of the earliest response turn that provides a correct answer as the model's response time $\mathrm{T}_{\text{start}}$. By computing the difference $\Delta\mathrm{T}$ between $\mathrm{T}_{\text{start}}$ and the reference latest response time T_{ref} (the timing of the first "umm" as described in Section 3), we can measure how quickly the model performs buzz-in. For response content, we design a turn-decayed reward. Specifically, if the model provides the correct answer in its first response turn, the reward is 1; if the correct answer appears in the second turn, the reward is 0.5; and if the correct answer appears in any later turn, the reward is 0. This reward penalizes the model for making premature guesses in order to respond earlier. To capture self-correction after an incorrect initial response, we also report later-turn accuracy ($\geq$2 acc), which measures the proportion of cases where the model eventually provides a correct response but does so in the second turn or later.

Clarification requests For this task, we mainly evaluate whether the model can proactively request clarification when there is uncertainty in the user's question. Therefore, we concatenate the model's responses for evaluation. The evaluation is conducted in two steps. First, an LLM determines whether the model is directly answering the question or requesting clarification. If the model is answering the question, its answer is compared against the reference answer and assigned a score of 1 if correct and 0 otherwise. If the model is requesting clarification, we further judge whether the request focuses on the keyword that is masked by noise or dropped from the input; if so, it receives a score of 1, and otherwise 0. For direct model comparison, we also report a composite clarification score, $S_{\mathrm{CLA}} = r_{\mathrm{CLA}}\,\mathrm{acc}_{\mathrm{CLA}} + \mathrm{acc}_{\mathrm{ANS}}$, for each masking level.

For clarification LLM-Judgement, we have also checked the agreement between model and human, on 51 test samples, Table9 showed that LLM agreement with human is even higher to that of the humans. This means the clarfication judge is very aligned with humans.

Table 9: Clarification-judge agreement with 95% bootstrap confidence intervals.

Agreement	Result	95% CI
Jev ↔ Human Accuracy	**96.1%**	92.5%–99.0%
Jev ↔ Human Answer F1	**97.5%**	∼93.3%–100%
Human ↔ Human Accuracy	**92.2%**	84.9%–98.0%
Human ↔ Human Answer F1	**95.0%**	89.7%–98.8%

Backchannel An LLM judge identifies which model utterances are backchannels. We report three metrics. **Backchannel Score** is the percentage of all model utterances that are judged to be backchannels occurring at a TRP. **Backchannel frequency** is the number of judged backchannels occurring at TRPs divided by the total number of TRPs. **TRP-relative timing JSD** compares the distribution of judged model backchannel onsets relative to the nearest TRP with the corresponding human distribution. The same denominator is used to compute model and human frequencies. JSD is undefined when a model produces no backchannels; its computation is detailed in Appendix A. The judge input and prompt will be specified once the evaluation protocol is finalized.

Because we use an LLM judge to determine whether a given utterance is a backchannel, we evaluate its agreement with human annotations and compare it with the rule-based backchannel detection algorithm used in Full-Duplex-Bench (Lin et al., 2025b). Specifically, we measure both human–human agreement and agreement between the LLM judge and human annotators.

Table 10: Backchannel-judge agreement with human annotations and 95% bootstrap confidence intervals.

Agreement	Result	95% CI
Human ↔ Human	74.7%	65.7%–83.0%
LLM-as-Judge ↔ Human	68.4%	60.1%–76.2%
FDB Judge ↔ Human	36.7%	28.6%–45.1%

As shown in Table 10, human-to-human agreement is 74.7%, while agreement between the LLM judge and human annotators is 68.4%. The bootstrapped 95% confidence interval for their difference is $[-17.4, 4.4]$ percentage points, and the probability that human-to-human agreement is higher is 85.6%. Because this probability is below the 95% threshold, the difference is not statistically significant. In contrast, the probability that the LLM judge outperforms the FDB judge is 99.995%, with a bootstrapped difference of $[16.0, 46.4]$ percentage points. These results indicate that the LLM judge agrees with human annotations significantly more often than the rule-based FDB judge does.

As for TRP-relative timing JSD used alongside backchannel frequency.

Jensen–Shannon Divergence of TRP-Relative Timing Frequency alone does not determine whether model backchannels occur at appropriate conversational moments. We therefore compare the TRP-relative timing distribution of model backchannels against the empirical human distribution shown in Figure 4(b). Let P denote the human distribution over discretized time offsets relative

to the nearest TRP, and let Q denote the corresponding model distribution. We compute their Jensen–Shannon divergence as

$$\mathrm{JSD}(P\|Q) = \frac{1}{2}\sum_i P(i)\log\frac{P(i)}{M(i)} + \frac{1}{2}\sum_i Q(i)\log\frac{Q(i)}{M(i)},$$

where

$$M(i) = \frac{1}{2}\big(P(i)+Q(i)\big).$$

The same TRP detection procedure and offset bins are applied to human and model backchannels. JSD is reported only when the model produces at least one judged backchannel.

B Prompts

B.1 Proactive buzz-in: Generating data

Prompts for GPT-5.2-2025-12-11

```
You are generating training data for Collaborative Turn Completion in
    a Word Search setting.

Your task:
Convert each input trivia-style question into ONE natural spoken
    dialogue example in JSON format, following the schema and style
    below.

Core idea:
The original question implies a hidden answer. In the dialogue, that
    hidden answer becomes the Target word or phrase the user is trying
    to recall during conversation. The user should sound like they are
    actively searching for the word in real time, under mild cognitive
    load, and should often produce natural word-search signals,
    semantic cues, phonetic cues, or both. The assistant should only
    guess when there is enough evidence, or ask for more detail if the
    cues are still insufficient.

Important:
1. First infer the correct answer to the question internally.
2. Then transform the question into a natural word-search dialogue
    where:
   - the answer becomes the "Target"
   - the user does NOT ask the original trivia question directly
   - instead, the user tries to remember the answer in a spoken,
    everyday way
3. The output must be concise, natural, conversational, and suitable
    for speech or TTS.
4. ** The structure of every UserA utterance must strictly follow
    this order: first the clue content, then one or more help-seeking
    signals. Before the help-seeking signal appears, the Assistant
    must already have access to the complete semantic information. **

Output schema:
{
  "Word_Search_Type": "people | places | titles_of_works | proverbs |
    poems | idioms | objects | rare_words |
    time_amounts_addresses_quantities_ID_etc",
  "Target": "...",
  "Alternatives": ["..."],
  "Cue_Type": "semantic | phonetic | hybrid",
  "Insufficient_Cue_Rounds": 0 | 1 | 2,
  "Guess_Rounds": 1 | 2 | 3,
  "Dialogue": [
```

```
    {
      "speaker": "UserA",
      "text": "...",
      "prefix": "..."
    },
    {
      "speaker": "Assistant",
      "text": "..."
    },
    {
      "speaker": "UserA",
      "text": "...",
      "prefix": "..."
    }
  ]
}

Dialogue rules:
- Only use:
  - "speaker"
  - "text"
  - "prefix" for UserA turns only
- Allowed speakers:
  - "UserA"
  - "Assistant"
- Do not generate any other fields inside Dialogue items.
- Every UserA turn must contain both:
  - "text": the full spoken utterance
  - "prefix": the substring from the beginning of the utterance up to
    the end of the first help-seeking signal
- Assistant turns must contain only:
  - "speaker"
  - "text"

Prefix rules:
- The prefix must be an exact leading substring of the corresponding
    UserA "text".
- The prefix must start at the beginning of the UserA utterance.
- The prefix should end at or near the end of the first explicit
    help-seeking signal.
- UserA MUST present all clue content before the first help-seeking
    signal.
- Therefore, the prefix will usually contain both:
  - a clue-bearing segment
  - the first help-seeking signal
- Do NOT place the help-seeking signal before the main clue.
- Typical help-seeking signals are similar to:
  - "... uh ... um"
  - "um ... uh ..."
  - "hmm ... uh ..."
  - "um ... wait ... hmm"
  - "hmm ... what's it called?"
  - "uh ... what's the name again?"
  - "hmm ... uh ... I can't remember"
  - "um ... I'm blanking ... hmm"
- If a help-seeking expression has multiple words, include only the
    first word.

Generation rules:
- The dialogue must sound like natural spoken English, not written
    prose.
- Keep it short and direct.
```

```
- The user is trying to remember the target answer, not asking a quiz
    question.
- Strongly prefer spoken disfluency and word-search behavior.
- In most examples, include at least one explicit help-seeking or
    word-search signal in every UserA turn.
- In many examples, include two or more such signals in the same
    UserA turn when natural.
- Every time you use "um/uh/hmm ..." repeat it 2-3 times in a row,
    joined with "...".
- MUST follow this order for UserA turns:
  1. clue first
  2. help signal second
- Mild to moderate disfluency is preferred; the user should sound
    like they are genuinely searching for the word in real time.
- Do not make the dialogue unnatural, exaggerated, or comedic.
- The assistant should:
  - guess directly if the cue is already strong enough
  - otherwise ask for one more clue
- If Guess_Rounds > 1, the earlier wrong guesses in "Alternatives"
    must be plausible given the cues at that point.
- Maximum wrong guesses: 2.
- Final guess must be the true Target.
- The final user reply should confirm the answer naturally, with
    relief.
- Keep the knowledge accurate.

How to map trivia questions into this format:
- "Where in England was Dame Judi Dench born?"
  -> Target is the birthplace, not Judi Dench.
- "Which prince is Queen Elizabeth II's youngest son?"
  -> Target is the prince's name or title.
- "What drink consists of 7 parts tequila, 4 parts cointreau or
    triple sec, and 3 parts lemon or lime juice?"
  -> Target is the drink name.
- In general, identify what entity the question is asking for, and
    make that entity the Target.

Choosing metadata:
- Word_Search_Type should reflect the answer entity, not the subject
    mentioned in the question.
- Cue_Type:
  - semantic: meaning, role, facts, associations
  - phonetic: first letter, sounds like, partial pronunciation
  - hybrid: both
- Insufficient_Cue_Rounds:
  - 0 if the first clue is enough for a confident guess
  - 1 or 2 if the assistant should ask for more detail first
- Guess_Rounds:
  - 1 if the assistant guesses the target directly
  - 2 or 3 if there are plausible wrong guesses first

Output requirements:
- Return ONLY valid JSON.
- If multiple questions are provided, return a JSON array.
- Do not include explanations.
- Do not repeat the original question verbatim in the dialogue unless
    absolutely necessary.
- Make each dialogue feel like a real-life recall moment.

Now convert the following question(s) into the required JSON output:
<<QUESTION>>
```

B.2 Proactive buzz-in: Evaluations

Prompts for GPT-5.2-2025-12-11

```
You are a strict factual grader for question answering.

You will be given:
1. A question.
2. The reference answer.
3. A candidate model output.

Judge whether the candidate model output correctly answers the
    question.

Rules:
- Output score `1` if the candidate output is factually correct or
    clearly equivalent to the reference answer.
- Output score `1` if the candidate output uses a shortened but still
    clearly identifying form of the same answer.
- Output score `0` if the candidate output is wrong, irrelevant,
    refuses to answer, asks for clarification, only repeats the
    question, or does not contain an answer.
- Output score `0` if the candidate output contains conflicting
    guesses and is not clearly correct overall.
- Ignore minor wording differences and harmless extra explanation.
- Be strict about factual correctness, but allow standard aliases or
    obvious paraphrases.

Return exactly one JSON object:
{"score": 0 or 1, "reason": "<brief reason>"}

[Question]
<<QUESTION>>

[Reference Answer]
<<REFERENCE_ANSWER>>

[Candidate Output]
<<CANDIDATE_OUTPUT>>
```

B.3 Contextual proactive buzz-in: Generating data

Prompts for GPT-5.2-2025-12-11

```
You are generating training data for Collaborative Turn Completion in
    a Word Search setting.

Your task:
Convert each input QA item into ONE natural spoken word-search
    question in JSON format.

Core idea:
In your rewritten output, you should convert the original open-ended
    question into a word search format, and the original correct
    answer should become the missing word or phrase that the speaker
    is trying to recall in real time. The speaker should sound like
    they recently read some material and are now trying to remember it
    naturally.

Important:
```

```
1. The output must be concise, natural, conversational, and suitable
    for speech or TTS.
2. The structure must strictly follow this order:
   - first the clue content
   - then one or more help-seeking signals
3. Before the help-seeking signal appears, the listener must already
    have access to the complete semantic clue.
4. Naturally incorporate the fact that the user recently read a
    passage, article, material, or scenario, but do not paste large
    chunks of context.
5. Stay faithful to the provided context, even if it is
    counterfactual.

Output schema:
{
  "text": "...",
  "prefix": "..."
}

Field rules:
- "text" is the full spoken question.
- "prefix" must be the exact leading substring of "text".
- "prefix" must start at the beginning of "text".
- "prefix" should end during or right after the first explicit
    help-seeking signal.
- If the help-seeking signal has multiple words, include only the
    first word in the prefix when possible.
- Most importantly, the "prefix" itself should be answerable without
    hearing the whole "text".

Generation rules:
- The user is trying to remember the answer, not asking the original
    QA question directly in formal wording.
- Do NOT reveal the answer in the question itself. Do NOT restate the
    answer text, paraphrase the answer content too specifically, or
    turn the answer into an asserted clause before the help-seeking
    signal.
- The generated question must require the context to answer, rather
    than being answerable based only on the question itself.
- Strongly prefer spoken disfluency and word-search behavior.
- In most cases include at least one explicit help-seeking signal.
- When you use "um", "uh", or "hmm", prefer repeating it 2-3 times in
    a natural way, joined with "...".
- Keep the utterance short and direct.
- Do not make it unnatural, exaggerated, or comedic.
- Do not put the help-seeking signal before the main clue.
- Do not quote or paste long parts of the context.
- Do not mention answer options explicitly.
- The wording should resemble questions in spoken word-search data
    such as:
  - "There's that specific term for killing your own mother, like the
    counterpart to patricide-um... uh... hmm... what's it called?"
  - "I'm trying to remember the year the UK government first
    published the Highway Code-the very first edition, not a later
    update-um... um... what year was that again?"
  - "The movie that won the Best Animated Feature Oscar for 2005-the
    claymation one with that British guy and his dog-um... uh... hmm,
    what was it called?"
- It is good to begin naturally with context framing like:
  - "I just read something about..."
  - "I was reading this passage where..."
  - "I recently read a piece about..."
  but keep that framing brief.
```

```
Output requirements:
- Return ONLY valid JSON.
- Return exactly two keys: "text" and "prefix".
- Do not include explanations.

Now rewrite the following item.

Original question:
<<QUESTION>>

Context:
<<CONTEXT>>

Reference answer for internal use only.
Never copy or paraphrase it directly into the output:
<<ANSWER>>
"""
```

B.4 Contextual proactive buzz-in: Evaluations

Prompts for GPT-5.2-2025-12-11

```
You are a strict answer-matching grader for question answering with
    counterfactual context.

You will be given:
1. A question.
2. A counterfactual context that defines the intended world/facts for
    this question.
3. The reference answer (may be counterfactual relative to the real
    world).
4. A candidate model output.

Your task is NOT to judge real-world factuality.
Instead, judge whether the candidate output matches the reference
    answer AND is consistent with the provided context.

Rules:
- Output score `1` ONLY if the candidate output clearly gives the
    same answer as the reference answer AND does not contradict the
    context.
- Output score `0` if the candidate output gives a different answer,
    even if that different answer is true in the real world.
- Output score `0` if the candidate output is irrelevant, refuses to
    answer, asks for clarification, only repeats the question, or does
    not contain an answer.
- Output score `0` if the candidate output contains conflicting
    guesses and is not clearly aligned with the reference answer.
- Ignore minor wording differences and harmless extra explanation, as
    long as the answer matches the reference answer.

Return exactly one JSON object:
{"score": 0 or 1, "reason": "<brief reason>"}

[Question]
<<QUESTION>>

[Context]
<<CONTEXT>>
```

```
[Reference Answer]
<<REFERENCE_ANSWER>>

[Candidate Output]
<<CANDIDATE_OUTPUT>>
```

B.5 CLARIFICATION: GENERATING DATA

Prompts for Doubao-Seed-1.6

```
You are selecting one critical keyword from an English spoken QA
    question.

Choose exactly one span (1 or 2 words) from the original question.
This span should be the single most important part such that if it
    becomes hard to hear, the listener would likely need clarification
    before answering.

A simple Rule of Thumb is if you cannot answer the question without
    the keyword, then that is a critical keyword.

Rules:
1. All output text must be in English.
2. Copy the span exactly from the question text.
3. Choose only 1 to 2 words as the span (keyword)
4. Prefer a person, place, date, title, relation, quantity, or other
    core semantic slot.
5. If multiple spans are possible, choose the one most likely to
    trigger clarification when corrupted.
6. Return valid JSON only, exactly matching the schema.

Example 1
Input:
{
  "question_text": "Who was Jackie Kennedy's second husband?"
}
Output:
{
  "keyword_text": "Jackie Kennedy"
}

Example 2
Input:
{
  "question_text": "Which country hosted the 1968 Summer Olympics?"
}
Output:
{
  "keyword_text": "1968"
}

Example 3
Input:
{
  "question_text": "What was the first capital of Australia?"
}
Output:
{
  "keyword_text": "Australia"
}
```

B.6 Clarification Eval: Identify the model response type

Prompts for GPT-5.2-2025-12-11

```
You are a strict evaluator for spoken-question answering systems.

You will be given:
1. The original user question.
2. The keyword in the question that was masked by noise or silence.
3. The model output.

Your task is to decide whether the model output is:
- `answer`: the model is trying to answer the original question.
- `clarification`: the model is asking the user to repeat, clarify,
    or confirm the masked keyword or missing content.

Definition details:
- Choose `answer` if the output gives any candidate answer, even if
    it may be wrong, incomplete, hedged, or contains extra explanation.
- Choose `answer` if the output guesses what the user meant and then
    answers.
- Choose `clarification` if the output mainly asks for repetition,
    asks what a word was, asks the user to clarify the masked part, or
    says it could not hear/understand the keyword.
- Choose `clarification` if the output does not answer the question
    and instead requests more information.
- If the output both asks for clarification and gives no actual
    answer, choose `clarification`.
- If the output both mentions uncertainty and still gives a
    substantive answer, choose `answer`.

Return exactly one JSON object with this schema:
{"label": "answer" or "clarification", "reason": "<brief reason>"}

[Question]
<<QUESTION>>

[Masked Keyword]
<<KEYWORD>>

[Model Output]
<<MODEL_OUTPUT>>
```

B.7 Clarification: Evaluating response classified as "answers"

Prompts for GPT-5.2-2025-12-11

```
You are a strict factual grader for question answering.

You will be given:
1. A question.
2. A model-generated answer.

Judge whether the model answer correctly answers the question.

Rules:
- Output score `1` if the answer is factually correct or clearly
    equivalent.
- Output score `0` if the answer is factually wrong, irrelevant,
    answers a different question, or fails to answer.
```

```
- Ignore minor wording differences and harmless extra explanation.
- If extra incorrect content makes the answer misleading, use `0`.
- If the answer is ambiguous or unsupported, use `0`.

Return exactly one JSON object:
{"score": 0 or 1, "reason": "<brief reason>"}

[Question]
<<QUESTION>>

[Model Answer]
<<MODEL_ANSWER>>
```

B.8 Clarification: Evaluating response classified as "clarifications"

Prompts for GPT-5.2-2025-12-11

```
You are a strict evaluator of clarification behavior.

You will be given:
1. A question.
2. The keyword in the question that was masked by noise or silence.
3. A model output already labeled as `clarification`.

Your task is to judge whether the model output is specifically asking
    about the keyword or the content surrounding that keyword.

Score definition:
- Output score `1` if the model is asking the user to repeat,
    clarify, confirm, spell, or explain the masked keyword, or clearly
    asking about the missing content centered on that keyword.
- Output score `1` if the model asks about the entity, phrase, or
    term corresponding to the keyword, even if it does not repeat the
    exact keyword string.
- Output score `0` if the model asks for general repetition without
    focusing on the keyword-related content.
- Output score `0` if the model asks for the whole question again,
    asks for unrelated information, or says it cannot help without
    specifically targeting the keyword.
- Output score `0` if the model output is not really a clarification
    request.

Return exactly one JSON object:
{"score": 0 or 1, "reason": "<brief reason>"}

[Question]
<<QUESTION>>

[Masked Keyword]
<<KEYWORD>>

[Model Output]
<<MODEL_OUTPUT>>
```

B.9 Turn Completion: Data Generation Prompt

Turn-completion generator prompt

```
GENERATOR_PROMPT = """You are a Generator Agent for converting
    Faithful QA items into duplex conversational CTC-style data.

Input fields:
- id
- context
- question
- answer
- interaction_type

Important dataset properties:
- The context may be long.
- The answer may be a full sentence or an explanation.
- The answer may not appear verbatim in the context.
- You must stay faithful to the context and the given answer.

Task:
Generate ONE dialogue for the requested interaction_type.

Allowed interaction_type values:
1. stalled_direct_completion
2. fluent_anticipatory_completion

Step 1:
Create a short answer_bearing_fact that contains the original answer
    exactly.
The fact should summarize the relevant point from the passage without
    copying long context spans.

Step 2:
Use that fact to create the requested dialogue type.

Global requirements:
1. target_answer must be the original answer exactly, including
    punctuation.
2. Do not replace the answer with a synonym.
3. Do not use outside knowledge to correct or change the answer.
4. Keep the dialogue concise, spoken, and suitable for TTS.
5. Speaker A should sound like they recently read the passage.
6. Do not paste long chunks of context into the dialogue.
7. The final dialogue must make it clear how the source answer is
    realized.
8. Do not emphasize that the passage is weird, false, odd,
    counterfactual, or mistaken.

Type-specific rules:

For stalled_direct_completion:
- Speaker A stops naturally immediately before the answer slot.
- Speaker A may use light fillers such as "uh", "um", "like", or "you
    know".
- Speaker A must not explicitly ask for help.
- Speaker B directly completes with the original answer.
- Speaker A accepts/repeats the answer and continues.
- B's turn must be the answer only, with no explanation beyond the
    answer itself.

For fluent_anticipatory_completion:
- Speaker A is not stuck and does not show word-search behavior.
```

```
- Speaker A is fluently approaching the answer slot.
- Speaker B proactively predicts and says the answer before A says it.
- Use a natural cut-off marker like "-" if needed.
- Speaker A accepts the anticipated completion.
- B's turn must be the answer only, with no explanation beyond the
    answer itself.

Forbidden:
- Speaker A explicitly says "I forgot", "I can't remember", "what was
    it again", "what do you call it", "can you finish", or "help me".
- Speaker B gives extra explanation outside the answer in Type 1/2.
- The dialogue changes the provided answer.
- The answer appears before the intended answer realization point.
- The dialogue sounds like a formal exam question.

Output JSON only:
{
  "id": "...",
  "source_context": "...",
  "source_question": "...",
  "source_answer": "...",
  "answer_bearing_fact": "...",
  "interaction_type": "...",
  "context_sufficiency": "sufficient",
  "dialogue": [
    {"speaker": "A", "text": "..."},
    {"speaker": "B", "text": "..."}
  ],
  "speaker_a_initial_turn": "...",
  "listener_support": "...",
  "answer_realization_turn": "...",
  "target_answer": "...",
  "answer_alignment": "exact",
  "requires_source_context": true,
  "notes": ""
}

Now convert the following item.

interaction_type:
<<INTERACTION_TYPE>>

id:
<<ID>>

context:
<<CONTEXT>>

question:
<<QUESTION>>

answer:
<<ANSWER>>
"""
```

Turn-completion checker prompt

```
CHECKER_PROMPT = """You are Check Agent for Faithful-to-duplex-CTC
    conversion.

Judge each candidate strictly. Do not create unrelated new examples.
```

```
Input fields include:
- source_context
- source_question
- source_answer
- answer_bearing_fact
- interaction_type
- context_sufficiency
- dialogue
- speaker_a_initial_turn
- listener_support
- answer_realization_turn
- target_answer

Global validity criteria:
1. target_answer matches source_answer exactly after strip only.
2. answer_bearing_fact contains target_answer exactly.
3. answer_bearing_fact is faithful to source_context and
    source_answer.
4. The dialogue does not use outside knowledge to change the answer.
5. The dialogue is concise, spoken, and suitable for TTS.
6. Speaker A does not explicitly say they forgot, cannot remember,
    need help, or ask "what was it again".
7. The answer is not leaked before the intended answer realization
    point.
8. answer_realization_turn contains target_answer exactly.
9. Do not penalize just because the source content is unusual; keep
    the generated dialogue neutral.

Type-specific validity criteria:

For stalled_direct_completion:
- context_sufficiency must be "sufficient".
- Speaker A stops at a natural answer slot.
- Speaker B's listener_support must be target_answer.
- B is completing A's sentence, not answering a standalone QA
    question.
- A's follow-up accepts/repeats/incorporates target_answer.

For fluent_anticipatory_completion:
- context_sufficiency must be "sufficient".
- Speaker A should not sound stuck.
- Speaker B's listener_support must be target_answer.
- B should sound like proactive anticipated completion.
- A's next turn must accept B's anticipated phrase.

Return JSON only:
{
  "id": "...",
  "valid": true,
  "scores": {
    "answer_alignment": 1,
    "context_faithfulness": 1,
    "answer_bearing_fact_quality": 1,
    "type_match": 1,
    "turn_taking_naturalness": 1,
    "answer_timing": 1,
    "speaker_behavior": 1
  },
  "reason": "...",
  "fixed_example": null
}
```

```
Use scores from 1 to 5. A valid item should have answer_alignment=5,
context_faithfulness=5, and all other scores at least 4.

Reject if:
- target_answer is a synonym instead of source_answer.
- answer_bearing_fact does not contain target_answer.
- Type 1 or Type 2 becomes normal QA.
- Type 2 sounds like A is stuck.
- A explicitly says they forgot or asks for help.
- B explains beyond the answer in Type 1/2.
- The dialogue changes the provided answer.

Candidate JSON:
<<CANDIDATE_JSON>>
"""
```

C Limitation and Boarder Impact

In this work, we explore the evaluation of proactive interaction capabilities in conversational AI, but the tasks we consider are still limited. Extending the evaluation to a broader range of tasks will be an important direction for future work. In addition, we do not anticipate additional harmful impacts.