TECHNICAL REPORT • October 2026

ShadowMe: Turn-Taking That Listens Only to Its User

A speaker-aware turn detector that follows one enrolled user while other people talk nearby (released as voxturn-personal)

Yang Xiao
The University of Melbourne

Model & plugin · Live replay · Hosted demo & API · GitHub

Abstract

A voice agent decides at every pause of its user whether the user has finished, and while it speaks, whether the user is interrupting. Turn detectors hear whoever is speaking. When a family member, a colleague or a television talks near the microphone, the agent waits for them, answers the moment they stop, or stops talking because of them. We present ShadowMe, a speaker-aware turn detector that follows one user like a shadow. It extends voxturn's multiple-choice speech LLM to Qwen3-ASR-1.7B and conditions it on about 10 s of the user's voice, recorded in advance or captured from the start of the call, through four zero-initialised paths: reference tokens, per-token voice matching, a target-speaker VAD and feature-wise modulation inside the audio encoder. Training combines synthetic user and bystander mixtures, real meetings seen from the agent's side, synthetic and real counterfactual enrollment pairs, a clean-input teacher and voxturn v1 as a second teacher. On phone calls with a bystander as loud as the user, ShadowMe cuts the user off in 20.5% of pauses at a 300 ms latency budget, against 47.3% for voxturn v1; on dinner-party recordings never used in training, 46.0% against 61.7%. While the agent speaks, it stops for 90% of the user's interruptions and for 2% of bystanders' speech. Without a bystander it matches or beats voxturn v1 on every accuracy test of v1, including eot-bench in 13 of 14 languages, at 22 ms per decision. The model, a Pipecat plugin and a hosted API are released.

20.5% false cut-offs on phone calls with a bystander at 0 dB (v1: 47.3%)

46.0% DiPCo far field, a corpus never trained on (v1: 61.7%)

90 / 2% barge-ins stopped for: the user's vs a bystander's (v1: 54 / 45%)

13 / 14 eot-bench languages better than v1, without an enrollment

1 Introduction

End-of-turn (EOT) detection decides when a cascaded voice agent may answer: answer too early and the agent cuts the user off mid-thought, answer too late and the conversation drags [1]. Recent detectors read the user's audio, the transcript or both [2–5], and our voxturn [6] answers several turn-taking questions in one forward pass of a speech LLM. All of them are trained and evaluated on one speaker at a time: eot-bench [7] scores single-speaker user turns and TurnBench [8] gives each speaker a channel of their own.

Deployed agents rarely hear one speaker. A phone held in a living room, a laptop in an open-plan office or a car cabin picks up other people, and a speaker-agnostic detector cannot tell their speech from the user's. It fails in three ways.

1. **Waiting for the bystander.** Someone keeps talking through the user's final pause; the VAD hears speech, so the agent does not answer.

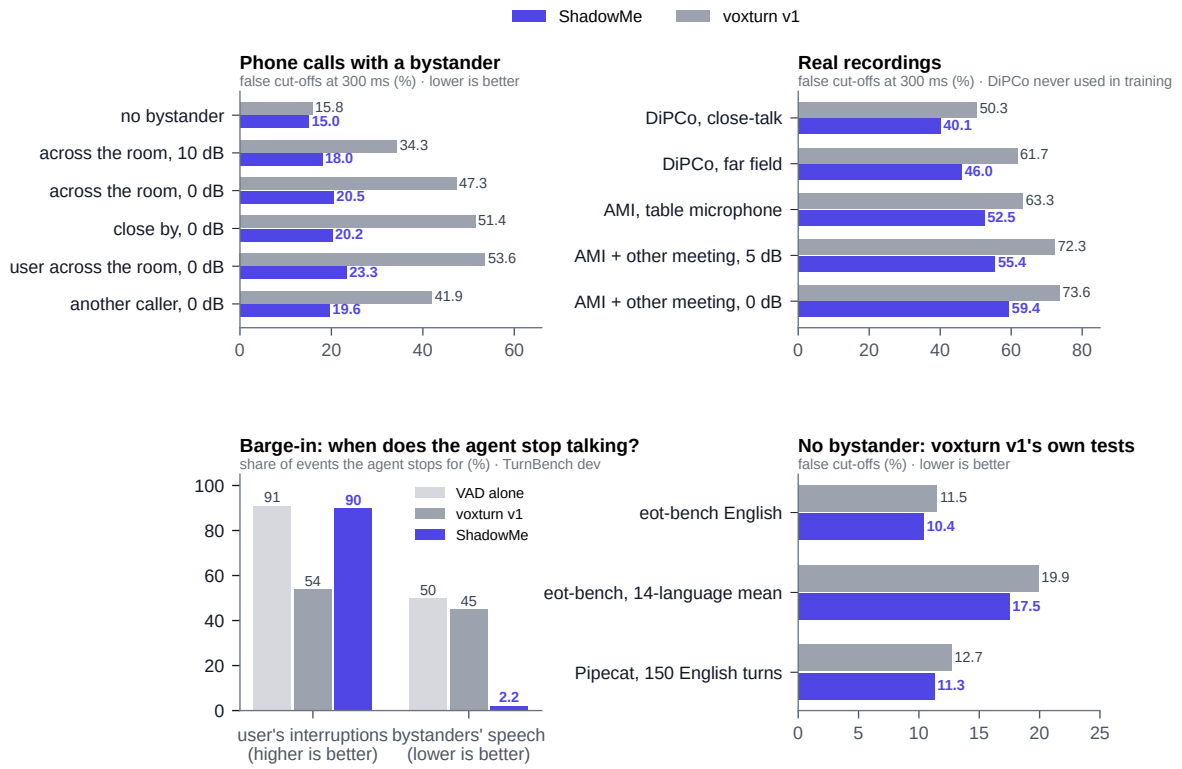


Figure 1. ShadowMe at a glance. Top left: phone calls (SpokenWOZ users) with a bystander, the share of the user's mid-turn pauses in which the agent would answer within a 300 ms latency budget. Top right: real recordings turned into agent-view turn-taking (DiPCo is never used in training; "other meeting" adds a real conversation from another AMI meeting). Bottom left: barge-in while the agent speaks, each detector letting through at most 10% of the user's backchannels. Bottom right: no bystander, voxturn v1's own tests (eot-bench at 300 ms; Pipecat with Silero VAD, share of turns cut off).

Contents

1 Introduction 1
2 Related work 3
3 ShadowMe 4
3.1 Turn-taking decisions for an enrolled user . . . 4
3.2 Model . . . 4
3.3 Four ways to listen for the user . . . 4
3.4 Training objective . . . 5
3.5 Enrollment from the call and deployment . . . 6
4 Training data and recipe 6
5 Evaluation suite 7
6 Results 8
6.1 Phone calls with a bystander . . . 8
6.2 Real recordings . . . 8
6.3 In a voice-agent framework . . . 9
6.4 Barge-in: only the user stops the agent . . . 9
6.5 Does it follow the enrolled voice? . . . 10
6.6 What drives the gains . . . 10
6.7 Everything voxturn v1 does . . . 11
6.8 Speed . . . 12
7 Limitations 12
8 Conclusion 13
Acknowledgements 13
A Per-language eot-bench results 16
B Training configuration 16
C Pipecat replay of eot-bench English 16

2. **Answering the bystander.** Someone stops talking during the user's pause, which looks like the end of a turn, so the agent cuts the user off.
3. **Yielding to the bystander.** While the agent speaks, any voice on the microphone looks like a barge-in, so the agent stops for people who are not talking to it.

These are not rare corner cases. On phone calls with a bystander as loud as the user, voxturn v1 cuts the user off in 47% of mid-turn pauses instead of 16% (Section 6.1); with a real conversation from another meeting across the room, it is no better than a detector without information.

Telling the detector who the user is addresses all three failures. Personal VAD [9] and target-speaker VAD [10] do this for voice activity, but voice activity is not a turn decision. A turn detector has to read the user's last words and their prosody, which an overlapping voice masks, and it has to keep doing everything a speaker-agnostic detector does when nobody else is there. ShadowMe therefore conditions the whole turn model on an enrollment of the user, so that every question it answers (has the user finished, will they go on, is this an interruption) is about that person. The name describes the behaviour: like a shadow, it follows one person. ShadowMe is released as *voxturn-personal*. Our contributions:

- a speaker-aware turn detector that answers end of turn, who is speaking and barge-in for one enrolled user in one pass, with an enrollment that can be captured automatically from the start of the call;
- four zero-initialised conditioning paths, so that the model starts exactly as a speaker-agnostic detector, and a training recipe of synthetic mixtures, real meetings seen from the agent's side, synthetic and real counterfactual enrollment pairs, a clean-input teacher and voxturn v1 as a second teacher;
- an evaluation on phone calls with bystanders and on real multi-party recordings turned into agent-view turn-taking (AMI, and DiPCo, never used in training), a reference-swap test on real recordings, and a barge-in test with real interruptions and backchannels;
- 20.5% instead of 47.3% false cut-offs with a bystander at 0 dB, 10–16 points fewer on real recordings, 90% of the user's interruptions and 2% of bystanders' speech stopping the agent, and no regression on any accuracy test of voxturn v1;
- open weights, a Pipecat [11] turn analyzer that treats bystander speech as the user's silence, and a hosted demo and API.

2 Related work

Turn-taking models. Voice Activity Projection [12] predicts the future voice activity of both speakers, and full-duplex models such as Moshi [13] make turn-taking implicit. Cascaded agents use a dedicated detector on the user channel: text models over the transcript [2, 4], audio classifiers [3], larger audio models [5, 14, 15], and models that also label dialogue states such as backchannel and wait [16, 17]. voxturn [6] answers several of these questions in one pass; ShadowMe keeps its readout and adds the user.

Turn-taking with other people around. TPI-Bench [18] measures how voice assistants handle third-party interruptions, without an enrollment. MultiPartyTurnBench [19] scores floor control among several synthetic speakers, with no designated user. Helwani et al. [20] segment a primary speaker online, the person who first answers the agent, before an end-of-turn model, and Foreground VAD [21] learns to keep only the dominant speaker; neither uses an enrollment, so a bystander who answers first or speaks louder takes the user's place. FireRedChat [22] blocks bystander barge-ins with a personal VAD inside a full-duplex system. To our knowledge, ShadowMe is the first open turn detector that conditions end-of-turn and barge-in decisions on an enrolled user and is evaluated on real multi-party recordings.

Target-speaker activity. Personal VAD [9] conditions frame-level voice activity on a speaker embedding; TS-VAD [10] estimates each enrolled speaker's activity from the whole window for diarization. ShadowMe uses both ideas as internal paths and supervises them with their per-frame labels, but its

Table 1. The questions ShadowMe answers in one pass. With an enrollment, "the user" is the enrolled person; a speaking-state marker in the prompt says whether the agent is speaking.

Question	What it asks	Options
`eot`	Has the user finished their turn?	yes / no
`target_active`	Who is speaking now? (new)	user / someone else / nobody
`resume`	When will the user go on?	within 0.5 s / later / not in this turn
`state4`	Easy Turn dialogue state	complete / incomplete / backchannel / wait
`intent`, `invalid`	as in voxturn: interruption vs backchannel; not valid input	

outputs are turn decisions. Feature-wise linear modulation (FiLM) [23] conditions the audio encoder, and the speaker embeddings come from ReDimNet [24].

3 ShadowMe

3.1 Turn-taking decisions for an enrolled user

At a decision time t, ShadowMe hears the last 8 s of the microphone, left-padded with silence, possibly with other people talking, and an enrollment of the user. As in voxturn, each decision is a multiple-choice question: the model scores the hidden state at each option instead of generating text, and a block-isolating attention mask answers any set of questions after a shared prefix in one pass, exactly as if they were asked one at a time [6]. Questions refer to "the user", the enrolled person (Table 1). Without an enrollment, the questions are voxturn's and ShadowMe is a speaker-agnostic turn detector.

Barge-in. While the agent speaks, the speaking-state marker says so, and the model is asked once the VAD has heard about 0.2 s of speech and then every 100–200 ms:

$$p_{\text{barge}} = p(\text{target_active} = \text{user})\,\big(1 - p(\text{state4} = \text{backchannel})\big)\,\big(1 - p(\text{invalid} = \text{yes})\big). \qquad (1)$$

The agent stops above a threshold (about 0.8). A bystander's speech has a low first factor, the user's "mm-hm" a low second factor, and a cough a low third one.

3.2 Model

The backbone is Qwen3-ASR-1.7B [25]: the AuT audio encoder (frozen, 24 layers) turns 128-bin log-mel features of the 8 s window into 104 audio tokens, a projector maps them into Qwen3-1.7B [26], and LoRA [27] (rank 16) adapts every linear layer of the language model. The sequence is [header][audio][reference][text][speaking state] followed by the question blocks (Figure 2). A frozen ReDimNet-B3 [24] gives a 192-dimensional embedding e_{ref} of the enrollment, computed once per user, and frame features of the 8 s input (one per 15 ms).

3.3 Four ways to listen for the user

The enrollment enters through four paths. Each is initialised to have no effect, so training starts from a speaker-agnostic model and the enrollment only changes an answer once the model has learnt to use it.

1. **Reference tokens** tell the language model whom to follow: an MLP maps e_{ref} to four tokens placed after the audio, behind a "the user's voice" marker.
2. **Voice matching** tells it which parts of the window are that person. ReDimNet's pooling head embeds about 1 s of frame features around each audio token as e_t, in the same space as e_{ref}; the features $[e_t \odot e_{\text{ref}},\ e_t,\ \cos(e_t, e_{\text{ref}})]$ pass through a small convolution and a zero-initialised projection that is added to that audio token. A per-token head predicts user / someone else / nobody and is supervised densely.

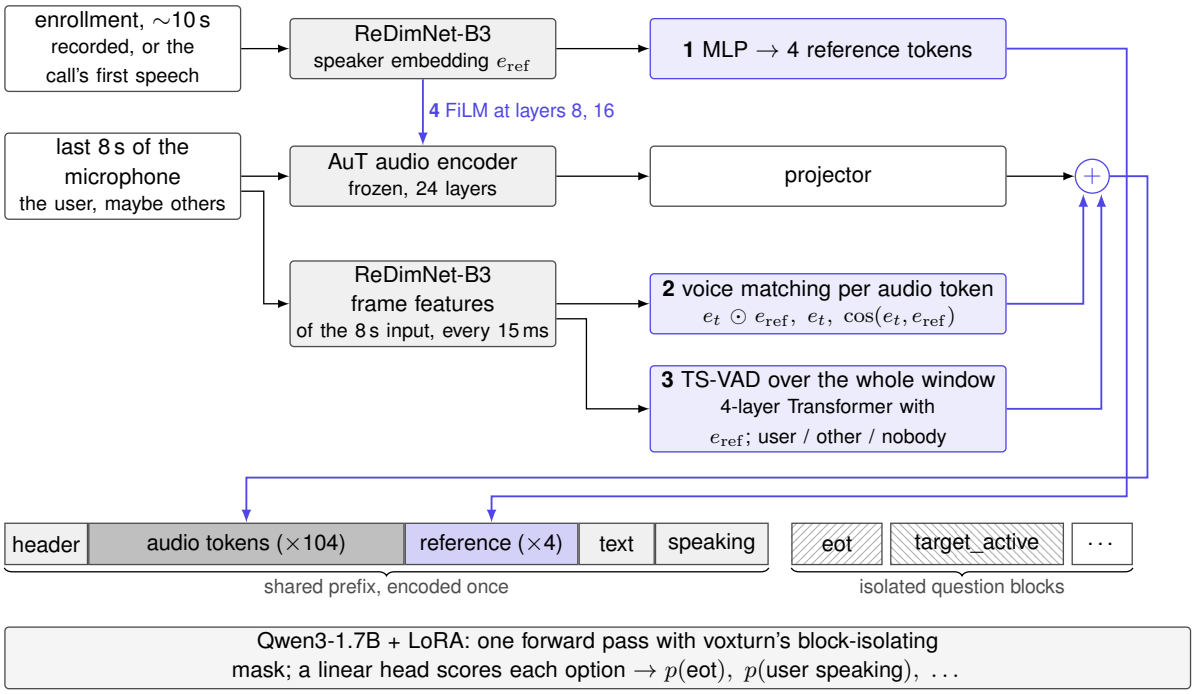


Figure 2. ShadowMe. A ReDimNet-B3 embedding of the enrollment conditions voxturn's speech LLM through four zero-initialised paths (indigo): (1) reference tokens in the shared prefix; (2) per-token voice matching and (3) a TS-VAD module, both added to the audio tokens; (4) FiLM inside the audio encoder. Grey boxes are frozen. Without an enrollment the input is exactly voxturn's.

3. **TS-VAD** decides the same per token from the whole window, which matters where voices overlap and a 1 s voice match is ambiguous: a 4-layer Transformer ($d = 256$) over the frame features, the matching features and e_{ref}, in the manner of target-speaker VAD [10]. Its logits replace the per-token head, and its hidden states reach the audio tokens and the encoder modulation through zero-initialised maps.

4. **AuT FiLM** lets the speaker information shape the audio representation itself: the speaker features scale and shift the hidden states of encoder layers 8 and 16 [23], before the user's last words are mixed with the bystander's in deeper layers.

The pooling head of ReDimNet is fine-tuned at 0.3 times the learning rate; the rest of ReDimNet and the audio encoder stay frozen. The model has 2.1B parameters (Qwen3-ASR-1.7B and ReDimNet-B3).

3.4 Training objective

ShadowMe keeps voxturn's objective: the main question of each sample, every auxiliary question whose label its timing determines, and a consistency term between two phrasings of the main question, with $\lambda_{\text{aux}} = \lambda_{\text{cons}} = 0.5$ [6]. Four terms are added.

- **Per-token activity** (weight 0.5): cross-entropy of the TS-VAD output against the user / someone else / nobody label of every audio token, from the sources' own timings.
- **Target-speaker transcription** (weight 0.3): an extra isolated block after the shared prefix asks the model to write out the user's last 12 words before the pause; on a counterfactual pair, each side writes its own person's words. It teaches the model to separate the user's words from the bystander's and is never used at inference.
- **Clean-input teacher** [28] (weight 1.0): on mixtures, every answer except `target_active` is pulled towards a frozen earlier checkpoint's answer on the user's audio alone.
- **voxturn v1 as a second teacher** (weight 1.0, added in the last stages): on all rows without a bystander,

Table 2. Training sources added to voxturn v1's mixture (Smart Turn, Easy Turn, MagicData-RAMC, SpokenWOZ and Seamless Interaction pauses, SemanticVAD, invalid sounds [6]). Synthetic mixtures are built on the fly from the user's pause clips and other people's speech.

Source	Content
User + bystander mixtures	SpokenWOZ [29] and Seamless Interaction [30] pauses of the user; another person's speech at -5 to 10 dB in random rooms, often covering the user's last words or ending just before the decision
Counterfactual pairs	two people's pauses mixed and aligned; the same audio asked with each one's enrollment, so that only the voice decides the answer
Clean and absent references	the user alone with an enrollment (no regression without a bystander); an enrollment of someone who is not in the audio
AMI, agent view	104 AMI [31] training meetings; for each user turn the participant who speaks next is the agent, and the user is heard on their headset, on the summed headsets and on the table microphone
AMI counterfactual pairs	the same real window, asked with the user's enrollment and with another participant's

the answers are pulled towards voxturn v1's. Two kinds of *probe questions* carry v1's behaviour where no training data asks: dialogue state and invalid sound under the agent-speaking marker (the barge-in questions), and text-only questions on transcripts. Without them the speaker-aware model loses v1's interruption handling, which no source supervises directly (TurnBench interruption recall 0.51, Table 7).

3.5 Enrollment from the call and deployment

An agent rarely has a recording of its caller. ShadowMe can capture the enrollment from the call itself: it keeps the speech that the agent's VAD hears on the user's input channel, from the start of the call, until 10 s are collected. Until 4 s are collected it answers speaker-agnostically; it then enrolls what it has, refreshes the enrollment every 2 s of new speech, and freezes it at 10 s. This assumes that the first speech on the channel is the user's.

Pipecat. The turn analyzer replaces Pipecat's default Smart Turn analyzer. The VAD reports anyone's speech; while it does, the analyzer asks `target_active` when the speech starts and every 0.2 s, and speech that is not the user's ($p(\text{user}) < 0.3$) counts as the user's silence, so the user's turn can end while a bystander talks on. While the user is silent, `eot` is asked every 0.2 s and the turn ends once $p(\text{eot}) \geq 0.85$, with the framework's timeout as a fallback, as in voxturn.

Serving. As for voxturn, the LoRA weights are merged and one CUDA graph is replayed per input layout. A decision takes 22 ms on an NVIDIA L40S (15 ms without an enrollment), four questions in one pass 23.5 ms, and the enrollment embedding 19 ms once per user; GPU memory is 4.8 GB (Section 6.8).

4 Training data and recipe

Synthetic mixtures. Table 2 lists what ShadowMe adds to voxturn's mixture. User clips are the labelled pauses of SpokenWOZ (task-oriented phone calls, user channel) and Seamless Interaction (face-to-face conversations, each speaker's own track), with holds and turn ends as in voxturn. A bystander from another call or another Seamless participant is mixed at a signal-to-interference ratio of -5 to 10 dB, with room impulse responses for one or both talkers. Many mixtures place the bystander over the decision time (the hardest case), and others end one of the bystander's utterances just before it, the moment that a speaker-agnostic detector mistakes for a turn end. Enrollments are 2–10 s of the user from another part of the same call or another session of the same person, sometimes with other voices in them.

Counterfactual pairs. In a counterfactual pair two people's pause clips are mixed and aligned so that both are pausing at the decision time, and the same audio is asked twice, once with each person's enrollment; the `eot` answers usually differ. The two halves of a pair are in the same micro-batch. Without such pairs a model can answer from cues such as loudness or distance ("the closer voice is the user") instead of comparing voices.

Real meetings from the agent's side. Synthetic mixtures differ from real multi-party speech more in their turn structure than in their acoustics: real people pause, overlap and take the floor in ways that cut-and-mix data does not reproduce. We therefore turn 104 AMI training meetings into agent-view turn-taking. For each user turn, the participant who speaks next is the agent; decisions in the user's pauses are labelled hold or end of turn as in voxturn, and the decision window closes when the agent starts, so the answer never leaks into the input. The user is heard on their own headset, on the summed headsets without the agent (as after echo cancellation) and on the table microphone. AMI counterfactual pairs ask the same real window, in which only the user speaks, with the user's enrollment (answer: the user) and with another participant's (answer: someone else). The other participant's voice is really in the room, through crosstalk or the table microphone, which makes these pairs hard negatives under real acoustics. The 30 AMI meetings used to build bystander test sets are excluded, and no SpokenWOZ test call, held-out Seamless speaker or DiPCo session is used in training.

Schedule. ShadowMe was trained in stages, each starting from an earlier checkpoint (Table 7, Appendix B): synthetic mixtures with the enrollment paths and counterfactual pairs; the clean-input teacher and TS-VAD (200k samples); real AMI turns and counterfactual pairs (100k); then voxturn v1 as second teacher and the probe questions (228k samples over four short stages). The later stages use batch 128, AdamW with learning rate 10^{-4} and bf16 autocast. They ran on NVIDIA H200, A100 and L40S GPUs.

5 Evaluation suite

Phone calls with a bystander. The users of 110 SpokenWOZ test calls (1,496 turns) are mixed with a bystander at a signal-to-interference ratio of 0, 5 or 10 dB measured on active speech: a held-out Seamless speaker across the room (reverberant) or close by, the user across the room with the bystander close by, or another phone caller on the same kind of channel. The bystander, offset and room of a turn are the same in every condition, so the conditions are paired. The enrollment is about 10 s of the user from elsewhere in the same call.

Real recordings, agent view. We build test sets from real meetings in the same way as the AMI training data. In AMI test meetings (16 users never seen in training) the participant who speaks next after the user is the agent and the other two are bystanders. The user is heard on their own headset, on the summed headsets without the agent, or on the table microphone. Two further sets add a stretch of another AMI meeting, recorded by its table microphone (other people, another room), to the user's headset at 0 and 5 dB: every sound is a real recording and only the level is set. DiPCo [32] dinner parties (5 evaluation sessions, 16 speakers, 441 turns) are built the same way, on the close-talk microphones, their sum and a far-field device, and are never used in training. Enrollments come from another meeting of the same person.

In a meeting, whether the floor passes after a finished sentence is decided by the other participants, so absolute cut-off rates are high for every model; a detector without information scores 73.6% on the AMI sets. We therefore also report the *decidable layer*: turn ends after a complete sentence against pauses after an unfinished one, the comparison that matters for an agent.

Reference swap on real recordings. On real windows in which only the user speaks, `target_active` must answer "user" with the user's enrollment and "someone else" with the enrollment of a participant of the same meeting, whose voice is present in the room. We report the share of 300 pairs per set answered correctly both times (pair accuracy).

Table 3. Phone calls with a bystander: false cut-offs at a 300 ms budget (%, lower is better) and the paired difference of ShadowMe to voxturn v1 [95% CI]. dB: user-to-bystander level; 0 dB = as loud as the user.

Condition	Smart Turn v3.2	voxturn v1	ShadowMe	Δ to v1
no bystander	38.5	15.8	**15.0**	−0.8 [−2.3, 0.3]
bystander across the room, 10 dB	48.4	34.3	**18.0**	−16.3 [−20.1, −13.8]
bystander across the room, 5 dB	48.8	40.6	**18.8**	−21.8 [−26.9, −17.8]
bystander across the room, 0 dB	52.9	47.3	**20.5**	−26.8 [−31.4, −22.7]
bystander close by, 0 dB	54.7	51.4	**20.2**	−31.2 [−35.0, −26.9]
user across the room, 0 dB	54.7	53.6	**23.3**	−30.3 [−33.0, −27.1]
another phone caller, 0 dB	54.7	41.9	**19.6**	−22.3 [−28.3, −19.3]

Barge-in. From TurnBench dev [8] (real two-party calls with three annotators), each speaker in turn is the user, enrolled from their first 10 s of speech in the call. Events are the user's real interruptions and backchannels, clean and with a far-field bystander talking throughout, and a bystander who starts talking while the user is silent: another speaker's ordinary speech, real interruption or backchannel, as loud as the user, close by or across the room (4,932 events). A detector is asked every 100 ms once the VAD has heard 0.2 s of speech, and its threshold lets through at most 10% of the user's backchannels (2-fold cross-validation over conversations).

voxturn v1's tests. Every test of the voxturn v1 report is rerun for both models with the same scripts, without an enrollment where the test has none: eot-bench in 14 languages [7], TurnBench dev [8], Easy Turn [16], OpenSLR 99 [33], TEN [17], SemanticVAD [34], and 150 eot-bench English turns replayed through Pipecat.

Metrics and protocol. Turn decisions follow the eot-bench protocol: the detector is asked every 100 ms in each silence, and a policy sweep over threshold, action delay and timeout gives the lowest false cut-off rate at a 300 ms mean latency (cut300, the share of mid-turn pauses in which the agent would answer; lower is better), the lowest latency at 5% false cut-offs, and the AUC between turn ends and holds. Differences to voxturn v1 are paired on the same rows, with 95% cluster bootstrap intervals (200 draws; clusters are calls or participants). Thresholds for frame-level detection are fixed on separate calibration sets with other people.

6 Results

6.1 Phone calls with a bystander

A bystander raises voxturn v1's false cut-offs from 15.8% to 34–54% (Table 3); Smart Turn v3.2 [3], already at 38.5% without one, rises to 48–55%. ShadowMe stays at 18–23%: 16 to 31 points fewer cut-offs than v1 in every condition, with AUC higher by 0.11–0.19. It loses nothing without a bystander (15.0%; AUC +0.009 [0.004, 0.014]).

The gain does not rest on a distance or loudness cue. It is as large when the user is the one across the room and the bystander is close to the microphone (53.6 → 23.3%), and when the bystander is another phone caller on the same kind of channel (41.9 → 19.6%). Figure 3 shows that the advantage holds at every latency budget. What remains is the cost of overlap itself: at 0 dB ShadowMe cuts off 5.5 points more than without a bystander, and its AUC is lowest where the bystander overlapped at least 30% of the user's last second of speech (0.934, against 0.963 without overlap).

6.2 Real recordings

Every difference on real recordings is significant (Table 4). On DiPCo, which ShadowMe never saw, it cuts the user off 10–16 points less often than v1, and AUC rises by 0.08–0.11; the far-field device, where everyone is across the room, gains most. With a real conversation from another meeting across

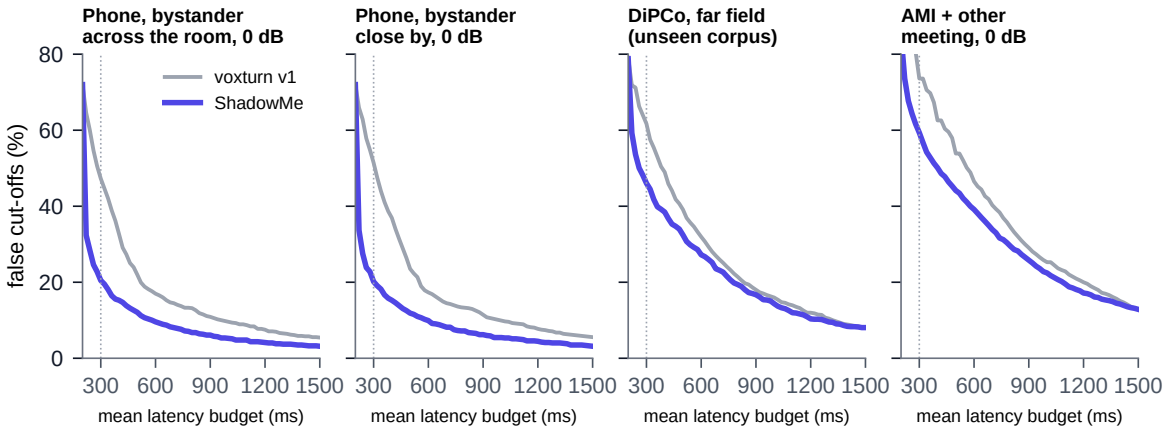


Figure 3. False cut-offs vs latency budget with a bystander: the lowest false cut-off rate at each mean response latency (dotted line: 300 ms), voxturn v1 and ShadowMe on the same rows.

Table 4. Real recordings, agent view: false cut-offs at 300 ms (%) and the paired difference of ShadowMe to voxturn v1 [95% CI]; AUC difference in the last column. A detector without information scores 73.6% on AMI.

Recording	voxturn v1	ShadowMe	Δ cut300	Δ AUC
DiPCo dinner parties (never used in training)				
user's close-talk microphone	50.3	**40.1**	−10.2 [−16.6, −4.9]	+0.088
summed close-talk microphones	53.3	**41.3**	−12.0 [−19.8, −4.3]	+0.078
far-field device	61.7	**46.0**	−15.7 [−18.6, −8.1]	+0.105
AMI test meetings (16 users never seen in training)				
user's headset	54.0	**46.4**	−7.6 [−10.5, −5.3]	+0.066
summed headsets without the agent	55.2	**48.9**	−6.3 [−10.8, −3.9]	+0.060
table microphone	63.3	**52.5**	−10.8 [−14.3, −7.3]	+0.085
user's headset + other meeting, 5 dB	72.3	**55.4**	−16.9 [−20.2, −13.3]	+0.156
user's headset + other meeting, 0 dB	73.6	**59.4**	−14.2 [−20.2, −10.0]	+0.174

the room, v1 is no better than a detector without information (73.6% at 0 dB), while ShadowMe reaches 59.4%. On the decidable layer, its AUC rises from 0.580 to 0.763 at 0 dB and from 0.648 to 0.799 at 5 dB: v1 can hardly tell a finished sentence from a hesitation any more, and ShadowMe still can. In the AMI meetings themselves, where the other participants seldom speak during the user's turn, the gain is smaller (6–11 points) but consistent across the three microphones.

6.3 In a voice-agent framework

We replayed 80 phone calls with someone talking nearby through Pipecat with Silero VAD [35], once with voxturn v1 and once with ShadowMe as the turn analyzer (Section 3.5), each with the enrollment captured from the start of the call. With ShadowMe, the agent answered within 1 s of the caller finishing on 54% of calls (voxturn v1: 35%), jumped in before the caller was done on 33% (35%) and kept the caller waiting for more than 1 s on 14% (30%). Replays of such calls, with both models' probabilities over time and the audio of the caller, the bystander and their mix, are in the live replay.

6.4 Barge-in: only the user stops the agent

A VAD stops the agent for nearly every interruption but also for half of the bystanders, and voxturn v1, which hears no difference between people, trades one error for the other (Table 5). ShadowMe stops for 90% of the user's interruptions, sooner than either (median 0.8 s), and for 2.2% of bystander events. The hardest case is a bystander's real interruption close to the microphone, which still stops the agent 4.5%

Table 5. Barge-in while the agent speaks (TurnBench dev, 4,932 events): share of the user's real interruptions the agent stops for (median delay), and share of bystander events it stops for. Every detector lets through at most 10% of the user's backchannels.

Detector	User interruptions stopped for ↑	Bystanders stopped for ↓
VAD alone (0.2 s of speech)	91% (1.4 s)	50%
voxturn v1, interruption vs backchannel	54% (0.9 s)	45%
ShadowMe, p_{barge}	**90% (0.8 s)**	**2.2%**

Table 6. Reference swap on real recordings: share of 300 pairs per set in which `target_active` answers correctly with both the user's and a co-participant's enrollment. s6 is an earlier checkpoint trained with synthetic counterfactual pairs only (Table 7, row D).

	DiPCo			AMI		
Model	far field	summed	close-talk	table mic	+ other meeting	summed
s6 (synthetic pairs only)	0.30	0.51	0.50	0.37	0.45	0.66
ShadowMe	**0.49**	**0.74**	**0.79**	**0.69**	**0.73**	**0.91**

of the time.

6.5 Does it follow the enrolled voice?

A model can lower cut-offs with a bystander without comparing voices, for instance by learning that the louder or closer voice is the user. The reference swap tests whether the answer really follows the enrollment (Table 6). Synthetic swap tests overstate this ability: the checkpoint trained on synthetic pairs only (s6) answered 0.86 of synthetic pairs correctly, but only 0.30–0.66 of real ones. Real counterfactual pairs from AMI raised the real score by 0.10–0.22 on every set, including DiPCo, which is never trained on, and two seeds agreed within 0.02. ShadowMe answers 0.69–0.91 of AMI pairs and 0.49–0.79 of DiPCo pairs correctly; the far field, where every voice reaches the device through the room, remains the hardest.

As a frame-level detector, `target_active` alone, with the threshold for a user recall of 0.95 fixed on calibration mixtures, takes 12.5% of the moments in which only a phone bystander speaks for the user at 0 dB and 7.0% at 10 dB. This is why the Pipecat analyzer only asks it while the VAD hears speech and uses a conservative threshold (0.3) before treating speech as a bystander's.

6.6 What drives the gains

Table 7 traces the development. It is a record of how the model was built, not a controlled ablation: the stages differ in more than the change named, and each was trained once.

Mixtures first. Most of the gain on synthetic phone mixtures comes before the enrollment. A larger backbone trained on user and bystander mixtures already halves v1's cut-offs with a bystander across the room (A: 47.3 → 23.9%): with level and distance cues, many bystanders can be ignored without knowing who the user is.

The enrollment where cues fail. The enrollment matters where loudness and distance stop identifying the user (A → B): close by (26.5 → 23.5%) and with the user across the room (33.1 → 25.9%). It is also what makes `target_active`, barge-in and the reference swap possible at all; a model without it cannot answer them. The clean-input teacher and TS-VAD add a few points more (D).

Real data for real recordings. Real AMI turn labels lowered DiPCo cut-offs by 4–8 points (D → E; 3–6 points against the same amount of extra training without them), although DiPCo is never trained on. Refinements trained on synthetic or unlabelled data, such as a target speech extraction front end, channel and layer-wise distillation, or real far-field interference without labels, improved in-domain results but

Table 7. The path to ShadowMe. False cut-offs at 300 ms (%) on phone calls (no bystander; bystander across the room, close by, and the user across the room, all at 0 dB), real recordings (DiPCo far field and close-talk; AMI headset + other meeting, 0 dB), eot-bench English, and TurnBench dev interruption recall. A and B are trained separately on voxturn's recipe; the other rows start from the model named. One seed each; –: not evaluated.

	Phone				DiPCo		AMI	EB	TB
Model	**clean**	**far**	**near**	**rev**	**far**	**ihm**	**bys**	**en**	**INT↑**
voxturn v1 (0.6B backbone)	15.8	47.3	51.4	53.6	61.7	50.3	73.6	11.5	.960
A: 1.7B + mixtures, no enrollment	15.7	23.9	26.5	33.1	56.7	46.0	–	11.1	–
B: 1.7B + enrollment paths, CF pairs	15.4	23.1	23.5	25.9	–	–	–	11.1	–
C: B + clean teacher	15.0	22.7	22.1	25.6	–	–	–	11.2	–
D: B + clean teacher + TS-VAD (s6)	15.3	21.1	20.5	22.9	52.1	46.3	60.8	10.6	.395
E: D + AMI agent-view turns	14.9	21.5	21.6	22.9	47.8	38.5	60.3	10.6	–
F: D + AMI turns and CF pairs	15.9	20.8	20.6	23.1	48.6	40.9	60.9	11.6	.507
ShadowMe: F + v1 teacher, probes	15.0	**20.5**	**20.2**	23.3	**46.0**	40.1	**59.4**	**10.4**	.951

Table 8. voxturn v1's tests, rerun for both models with the same scripts (no enrollment where the test has none). Text tests are paired per sample (McNemar); "level": no significant difference ($p > 0.05$). TurnBench dev at FPR ≤ 0.1 with the EOT decision policy of the v1 report.

Test	**voxturn v1**	**ShadowMe**	
eot-bench English: cut300 / latency at 5% cut-offs	11.5% / 669 ms	**10.4% / 563 ms**	better
eot-bench, 14-language mean cut300	19.9%	**17.5%**	better in 13 of 14
TurnBench dev, end of turn: recall / median latency	0.930 / 565 ms	0.930 / 716 ms	level, later
TurnBench dev, interruptions: recall / median latency	0.960 / 337 ms	0.951 / 506 ms	level, later
Easy Turn test (Chinese, dialogue state)	88.3%	**89.4%**	better
SLR99 sounds flagged invalid	99.6%	99.6%	level
valid speech flagged invalid	0.06%	**0.0%**	better
TEN, text only: English / Chinese	80.5% / 94.4%	**94.9%** / 93.4%	en better
SemanticVAD, text only: end of turn, en / zh	100% / 99.0%	99.5% / 98.5%	level
SemanticVAD, text only: interrupt vs backchannel	80.5% / 88.4%	82.0% / 88.2%	level
Pipecat, 150 English turns: cut off / median reply	12.7% / 0.30 s	**11.3%** / 0.30 s	better
One decision, NVIDIA L40S	**8.5 ms**	22 ms	slower

did not transfer to DiPCo. Real counterfactual pairs added speaker following (Section 6.5) at a small cost in turn decisions (E vs F).

Keeping what voxturn did. The speaker-aware stages lost v1's interruption handling (TurnBench recall 0.395 and 0.507): no training source asks the dialogue-state questions under the agent-speaking marker, and v1 had learnt this behaviour by generalisation. voxturn v1 as a second teacher with probe questions restored it (0.951) and improved all but one of the other columns.

6.7 Everything voxturn v1 does

Without a bystander and without an enrollment, ShadowMe keeps or improves every accuracy result of voxturn v1 (Table 8). On eot-bench it is better in English (10.4%, and 106 ms sooner at 5% cut-offs) and in 13 of 14 languages (Figure 4; mean 17.5%), although following an enrolled user was trained in English only; the exception is Italian (21.7 vs 21.0%). It is also better on Easy Turn and on TEN's English text test ($p < 0.001$); the other differences are not significant. On TurnBench dev it detects ends of turns and interruptions as reliably as v1 but later. In Pipecat with Silero VAD, on the 150 English turns of the v1 report, it cuts off 11.3% of turns at its default threshold against 12.7% for v1 and 41.3% for Smart Turn v3, the framework's default, at the same median reply time (Appendix C).

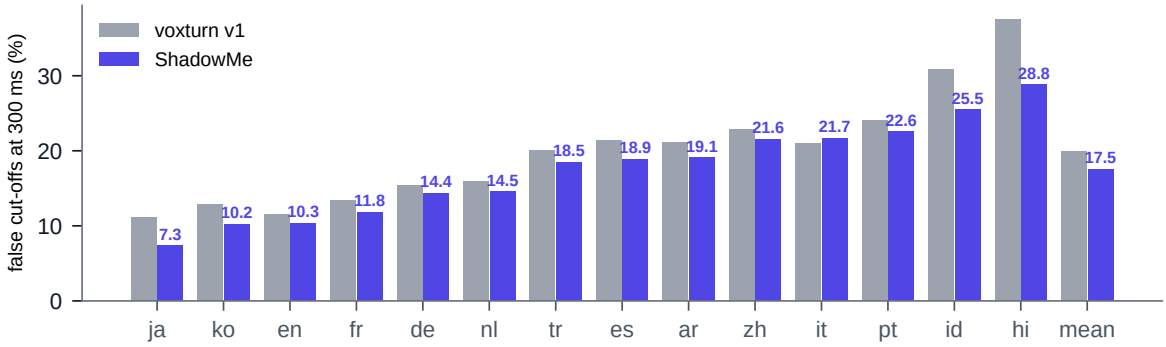


Figure 4. eot-bench per language without an enrollment: false cut-offs at a 300 ms budget, voxturn v1 and ShadowMe (Appendix A lists latencies).

Table 9. Latency per call (NVIDIA L40S, bf16, 8 s audio, through `predict`, end to end); GPU memory 4.8 GB.

Call	p50 (ms)	p95 (ms)
1 question (`eot`)	22.3	22.4
2 questions in one pass (`eot`, `target_active`)	22.6	22.7
4 questions in one pass	23.5	23.6
4 clips in one batch, 1 question each	54.4	55.0
without an enrollment	14.7	14.8
enrollment (10 s, once per user)	19.0	20.1
eager model, merged / unmerged LoRA	60.2	73.2
voxturn v1, serving / eager	8.5	40.8

6.8 Speed

A decision takes 22 ms in serving mode (Table 9), and asking `target_active` in the same pass costs 0.3 ms more. The 1.7B backbone and the speaker branch make ShadowMe 2.6 times slower than voxturn v1, but a detector asked every 100–200 ms keeps one GPU busy for 11–22% of the time of each call it serves, and `predict_batch` serves several calls at once.

7 Limitations

- **Language.** Following an enrolled user is trained and tested in English only; without an enrollment ShadowMe covers voxturn's 14 languages.
- **Overlap at the last words** still costs: at 0 dB the agent answers about 0.3 s later, and cut-offs are 5.5 points above the no-bystander level.
- **Who is speaking in the far field.** With loud bystanders across the room, `target_active` can take their speech for the user's. An agent with a better user-activity signal (a close-talk microphone with its own VAD) should use it for the pauses and ShadowMe for the turn decision.
- **Enrollment.** The enrollment captured from the call assumes that the first speech is the user's. Very similar voices, enrollments shorter than about 3 s and far-field enrollments are harder.
- **Evidence.** Each stage was trained once, and the last variants were chosen among four with the real test sets, so the DiPCo numbers carry a small selection effect. Barge-in bystanders are mixed in, not recorded. Multi-party floor control (who speaks next in a meeting) is out of scope.
- **Speed.** 22 ms per decision against 8.5 ms for v1; on TurnBench, interruptions are detected as reliably but later (median 506 vs 337 ms).

ShadowMe compares voices to decide turns. It is not a speaker verification system and must not be used to identify, verify or track people; enroll only users who have given informed consent, and store enrollments as personal data.

8 Conclusion

ShadowMe lets a voice agent follow one person in a room full of voices. Conditioning voxturn's multiple-choice speech LLM on an enrollment of the user, through four zero-initialised paths, and training it on synthetic mixtures, real meetings seen from the agent's side and counterfactual enrollment pairs, cuts false cut-offs with a bystander from 47.3% to 20.5% on phone calls and by 10–16 points on real recordings it never saw, and stops the agent for the user's interruptions but not for the room. With voxturn v1 as a second teacher it gives up none of v1's accuracy when nobody else is there. Weights, code and the Pipecat plugin are at https://huggingface.co/AustinXiao/voxturn-personal-turn-detector, the live replay at https://swagshaw.github.io/voxturn/personal/, and a hosted demo and API at https://huggingface.co/spaces/AustinXiao/voxturn-demo.

Acknowledgements

We thank Qiangju Chen of Macquarie University for naming ShadowMe. Experiments used computational resources of the National Computational Infrastructure (NCI Australia) and of the University of Melbourne. Generative AI tools were used to assist with writing code, running and summarising experiments, and drafting and editing this report; the authors reviewed and take responsibility for all content.

References

[1] Gabriel Skantze. Turn-taking in conversational systems and human-robot interaction: A review. *Computer Speech & Language*, 67:101178, 2021. doi: 10.1016/j.csl.2020.101178.

[2] Erik Ekstedt and Gabriel Skantze. TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020. arXiv:2010.10874.

[3] Pipecat. Smart turn: an open-source audio turn detection model. https://github.com/pipecat-ai/smart-turn, 2025.

[4] LiveKit. LiveKit turn detector. https://docs.livekit.io/agents/build/turns/turn-detector/, 2026.

[5] p99lab. turn-1. https://huggingface.co/p99lab/turn-1, 2026.

[6] Yang Xiao. VoxTurn: Answering several turn-taking questions in one pass with one model. Technical report, The University of Melbourne, 2026. URL https://github.com/swagshaw/voxturn/blob/main/tech_report.pdf.

[7] LiveKit. eot-bench: LiveKit end-of-turn benchmark. https://github.com/livekit/eot-bench, 2026.

[8] Freeman Jiang, Ramon Sanabria, Soham Deshmukh, Bandhav Veluri, Simon Michael Vuch Williams, Elliott K. Suen, Garreth Lee, Kevin Yoonho Choi, Takuya Umeki, Riku Kubo, Sathvik Udupa, Chien-yu Huang, Shih-Yun Shan Kuan, Zhuoyan Tao, Satyapriya Krishna, Sefik Emre Eskimez, Yu Tsao, Hung-yi Lee, and Shinji Watanabe. TurnBench: A multi-domain benchmark for

turn-taking dynamics in spoken dialogue. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2026. arXiv:2608.25218.

[9] Shaojin Ding, Quan Wang, Shuo yiin Chang, Li Wan, and Ignacio Lopez Moreno. Personal VAD: Speaker-conditioned voice activity detection. In *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2020. arXiv:1908.04284.

[10] Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, Aleksandr Laptev, and Aleksei Romanenko. Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario. In *Proc. Interspeech*, 2020. arXiv:2005.07272.

[11] Pipecat. Pipecat: open source framework for voice and multimodal conversational AI. https://github.com/pipecat-ai/pipecat, 2026.

[12] Erik Ekstedt and Gabriel Skantze. Voice activity projection: Self-supervised learning of turn-taking events. In *Proc. Interspeech*, 2022. arXiv:2205.09812.

[13] 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.

[14] JoinIn AI. Baton turn-taking engine. https://joinin.ai, 2026.

[15] Deepgram. Introducing Flux: Conversational speech recognition. https://deepgram.com/learn/introducing-flux-conversational-speech-recognition, 2025.

[16] Guojian Li, Chengyou Wang, Hongfei Xue, Shuiyuan Wang, Dehui Gao, Zihan Zhang, et al. Easy turn: Integrating acoustic and linguistic modalities for robust turn-taking in full-duplex spoken dialogue systems. *arXiv preprint arXiv:2509.23938*, 2025.

[17] TEN Framework. TEN turn detection. https://github.com/TEN-framework/ten-turn-detection, 2025.

[18] Dongwook Lee, Eunwoo Song, Che Hyun Lee, Heeseung Kim, et al. Still between us? evaluating and improving voice assistant robustness to third-party interruptions, 2026. arXiv:2604.17358.

[19] Puneet Mathur and Dinesh Manocha. MultiPartyTurnBench: Benchmarking floor control in multi-party spoken dialogue, 2026. arXiv:2610.20172.

[20] Karim Helwani, Hoang Do, James Luan, and Sriram Srinivasan. A hierarchical end-of-turn model with primary speaker segmentation for real-time conversational AI, 2026. arXiv:2603.13379.

[21] Guangzhao Yang, Muhammad Huzaifah, Yu Pan, Jinya Sakurai, et al. Foreground voice activity detection: Learning speaker selectivity from supervision, 2026. arXiv:2609.19856.

[22] Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, et al. FireRedChat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations, 2025. arXiv:2509.06502.

[23] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proc. AAAI*, 2018. arXiv:1709.07871.

[24] Ivan Yakovlev, Rostislav Makarov, Andrei Balykin, Pavel Malov, Anton Okhotnikov, and Nikita Torgashov. Reshape dimensions network for speaker recognition. In *Proc. Interspeech*, 2024. arXiv:2407.18223.

[25] Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, et al. Qwen3-ASR technical report. *arXiv preprint arXiv:2601.21337*, 2026.

[26] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022. arXiv:2106.09685.

[28] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. NIPS Deep Learning Workshop, 2014. arXiv:1503.02531.

[29] Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, et al. SpokenWOZ: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. In *Proc. NeurIPS Datasets and Benchmarks Track*, 2023. arXiv:2305.13040.

[30] Vasu Agrawal, Akinniyi Akinyemi, Kathryn Alvero, Morteza Behrooz, Julia Buffalini, Fabio Maria Carlucci, et al. Seamless interaction: Dyadic audiovisual motion modeling and large-scale dataset. *arXiv preprint arXiv:2506.22554*, 2025.

[31] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The AMI meeting corpus: A pre-announcement. In *Proc. Machine Learning for Multimodal Interaction (MLMI)*, 2005.

[32] Maarten Van Segbroeck, Ahmed Zaid, Ksenia Kutsenko, Cirenia Huerta, Tinh Nguyen, Xuewen Luo, Björn Hoffmeister, Jan Trmal, Maurizio Omologo, and Roland Maas. DiPCo – dinner party corpus, 2019. arXiv:1909.13447.

[33] Deeply Inc. Deeply nonverbal vocalization dataset (OpenSLR SLR99). https://www.openslr.org/99/.

[34] KE-Team. SemanticVAD dataset. https://huggingface.co/datasets/KE-Team/SemanticVAD-Dataset, 2025.

[35] Silero Team. Silero VAD: pre-trained enterprise-grade voice activity detector. https://github.com/snakers4/silero-vad, 2024.

A Per-language eot-bench results

Table 10. eot-bench per language, no enrollment: false cut-offs (%) at a 300 ms mean-latency budget and mean latency (ms) at 5% false cut-offs; lower is better.

	ar	de	en	es	fr	hi	id	it	ja	ko	nl	pt	tr	zh	mean
False cut-offs at 300 ms (%)															
voxturn v1	21.1	15.4	11.5	21.4	13.4	37.5	30.8	**21.0**	11.1	12.8	15.9	24.1	20.1	22.9	19.9
ShadowMe	**19.1**	**14.4**	**10.4**	**18.9**	**11.8**	**28.8**	**25.5**	21.7	**7.3**	**10.2**	**14.5**	**22.6**	**18.5**	**21.5**	**17.5**
Latency at 5% cut-offs (ms)															
voxturn v1	**751**	**801**	669	782	662	606	913	**912**	433	542	728	796	898	**814**	736
ShadowMe	758	823	**563**	**754**	**570**	**602**	**885**	939	**362**	**513**	**655**	**770**	**782**	841	**701**

B Training configuration

Table 11. ShadowMe training configuration (the final stages; earlier stages as in Table 7).

Setting	Value
Backbone	Qwen3-ASR-1.7B (AuT encoder frozen, Qwen3-1.7B language model with LoRA rank 16)
Speaker encoder	ReDimNet-B3, frozen except its pooling head (learning rate $\times 0.3$); 192-dimensional embedding
Conditioning	4 reference tokens; per-token voice matching (cosine features); TS-VAD (4 layers, $d = 256$); FiLM at AuT layers 8 and 16
Audio context	8 s, left-padded with silence; 104 audio tokens
Enrollment in training	2–10 s of the user, same call or another session, sometimes with other voices
Batch	128
Optimiser	AdamW, learning rate 10^{-4}, 100 warm-up steps
Precision	bf16 autocast
Loss weights	$\lambda_{\text{aux}} = \lambda_{\text{cons}} = 0.5$; per-token activity 0.5; target-speaker transcription 0.3; clean-input teacher 1.0; voxturn v1 teacher 1.0 (probe questions under the agent marker up to 4.0)
Stages	synthetic mixtures and enrollment paths; + clean teacher and TS-VAD (200k samples); + AMI agent-view turns and counterfactual pairs (100k); + voxturn v1 teacher and probe questions (100k, 64k, 40k, 24k)
Hardware	NVIDIA H200 (NCI Gadi), A100 (University of Melbourne), L40S

C Pipecat replay of eot-bench English

Table 12. 150 eot-bench English turns streamed through Pipecat's Silero VAD (no bystander; ShadowMe captures its enrollment from the turn itself). A turn counts as cut off if any of its pauses was; reply time from the start of the final silence.

Turn analyzer	Turns cut off	Reply: median / mean / p90 (s)
Smart Turn v3 (Pipecat default)	41.3%	0.14 / 1.37 / 3.10
voxturn v1, threshold 0.8 (its default)	12.7%	0.30 / 0.36 / 0.86
ShadowMe, threshold 0.85 (default)	**11.3%**	0.30 / 0.48 / 1.01
voxturn v1, threshold 0.9	8.7%	0.38 / 0.70 / 1.55
ShadowMe, threshold 0.9	**8.0%**	0.48 / 0.70 / 1.38