

Beyond Streaming: Conversational Floor Control in Speech LLMs and a Reproducible Telugu Speech-Pipeline Evaluation

Banuru Mohammed Fawaz

Technical assignment, Speech AI Research Internship, Vaani Research Lab

Code: <https://github.com/f4w4z/vaani-technical-assignment>

Abstract

Speech language models are moving from turn-based cascades to streaming, full-duplex systems, yet “real-time” is usually reported as response-onset latency. We make two contributions. *First*, we compare two recent speech LLMs, Moshi (full-duplex, speech-to-speech) and Qwen2.5-Omni (streaming, omni-modal, speech output), along seven architectural axes, record what their reports leave unspecified, and argue that streaming generation alone does not solve conversational *floor control*: deciding to continue, hold, or yield while still listening. We split “end-of-speech” into four distinct decisions and specify, as a falsifiable experiment that we have *not* run, a small controller on frozen Moshi that uses the model’s own generation state to choose among them. *Second*, we build and evaluate a reproducible Telugu speech-to-text-response cascade (Telugu-finetuned Whisper-small $\rightarrow$ Qwen2.5-1.5B-Instruct) with from-scratch metrics on 129 synthesized utterances in clean and 0 dB-SNR conditions. Noise raises WER from 0.293 to 0.338 (paired 95% CI of the increase: +0.014 to +0.077); a spectral-subtraction front end does not help (Δ WER +0.001, CI -0.014 to +0.017). The dominant failure is the small LLM: it states the correct fact in 0 of 129 answers from speech and 3 of 129 from perfect transcripts, and never abstains. Code and configurations are released.

1 Introduction

A speech assistant feels conversational only if it answers quickly *and* behaves sensibly about who holds the floor. Cascaded systems (ASR, text LLM, TTS) wait for an endpoint, so they cannot represent overlap or interruption. End-to-end streaming models such as Moshi (Défossez et al., 2024) and Qwen2.5-Omni (Qwen Team, 2025) remove that turn boundary and report low latency, but a latency number measures how soon speech *starts*, not whether the system keeps talking through a listener’s “mm-hm” or stops when it is genuinely interrupted.

This report has two parts that share a theme but make different kinds of claims, and we keep them apart.

Part I (analysis and proposal; no experiments). We analyze Moshi and Qwen2.5-Omni along the seven axes of the assignment brief, answer the research question of what architectural limitations prevent low-latency, natural, real-time conversation, and propose an improvement. Every statement about the models carries a section and page citation to the source report; everything beyond that is labeled interpretation or proposal.

Part II (implementation and measurement). We build and measure a *cascaded* Telugu pipeline with our own evaluation code. This part does not

test Part I’s hypothesis: a cascade processes whole utterances and has no floor-control decision to evaluate. It supplies a measured baseline for what a modest, reproducible Indic system does today.

Contributions. (1) A comparative architectural analysis with an explicit list of unreported quantities that block matched comparison (§3). (2) A decomposition of end-of-speech into four decisions and a ranked account of three limitations (§4–5). (3) A fully specified, refutable proposal: a generation-state-aware yielding controller, with baselines, a constrained primary metric, and ablations (§6). (4) A reproducible Telugu cascade evaluation with from-scratch WER/CER/RTF, a defined correctness and hallucination rate, paired bootstrap intervals, and an honest negative result for a denoising extension (§7–8).

2 Related Work

Duplex and streaming speech LMs. dGSLM (Nguyen et al., 2023) models two-channel dialogue directly. Moshi (Défossez et al., 2024) adds a text-initialized backbone, a causal codec, and an “inner monologue”. Mini-Omni (Xie and Wu, 2024) generates speech while reasoning in streaming fashion, and Qwen2.5-Omni (Qwen Team, 2025) pairs a Thinker with a Talker for text and speech output.

Turn-taking prediction. TurnGPT (Ekstedt and Skantze, 2020) predicts turn shifts from text, and Voice Activity Projection (Ekstedt and Skantze, 2022) predicts future voice activity from audio. Turn prediction therefore predates our proposal; the question we pose is narrower (§6). We have not exhaustively searched for prior controllers that read a speech generator’s hidden state, so we make no priority claim.

Indic ASR and denoising. Whisper (Radford et al., 2023) and MMS (Pratap et al., 2024) cover Telugu at differing quality, and community fine-tunes exist for individual Indic languages. Spectral subtraction (Boll, 1979) is a classical noise-suppression front end that we evaluate as a pre-ASR step.

3 Architectural Analysis of Two Speech LLMs

We chose Moshi and Qwen2.5-Omni because both are recent, both stream, and both produce speech (so the brief’s speech-to-speech requirement is met twice over). Table 1 gives the seven-axis comparison and Figure 1 the data flow. Citations are to the printed page numbers of the attached reports (M = Moshi, Q = Qwen2.5-Omni).

What the reports do not specify. Neither report fully specifies input-endpoint, playback-completion/flush, or session-stop rules, and Qwen2.5-Omni additionally leaves interruption handling unspecified. Qwen2.5-Omni does not report output-codec rate, codebooks or bitrate; adapter dimensions or depth; output-block duration; numerical end-to-end latency; audio training hours or named corpora; or complete tokenizer objectives. Moshi’s practical-latency figure comes without a complete measurement protocol, and Qwen’s “causal decoder” wording sits beside a DiT with one lookahead block. These omissions prevent matched codec, latency, data-efficiency and stopping comparisons. They do not show a performance failure.

4 Limitations Blocking Natural Real-Time Conversation

Our answer to the research question is three ranked limitations. Each is interpretation built on the cited mechanisms, not a result reported by either paper.

L1. Continuous generation leaves conversational decisions implicit. Moshi’s joint streams remove the turn boundary and let the model listen

while speaking (M §1, p.2; §3.4.4, p.17). What remains open is *when* that learned behavior correctly keeps or gives up the floor. Moshi evaluates gaps, overlaps and dialogue quality (M §5.6, p.31), but aggregate distributions cannot show whether one backchannel caused an inappropriate stop. Continuous modeling is a prerequisite for studying floor control, not evidence that it is handled well. Qwen2.5-Omni’s streaming perception and generation likewise do not establish event-level behavior.

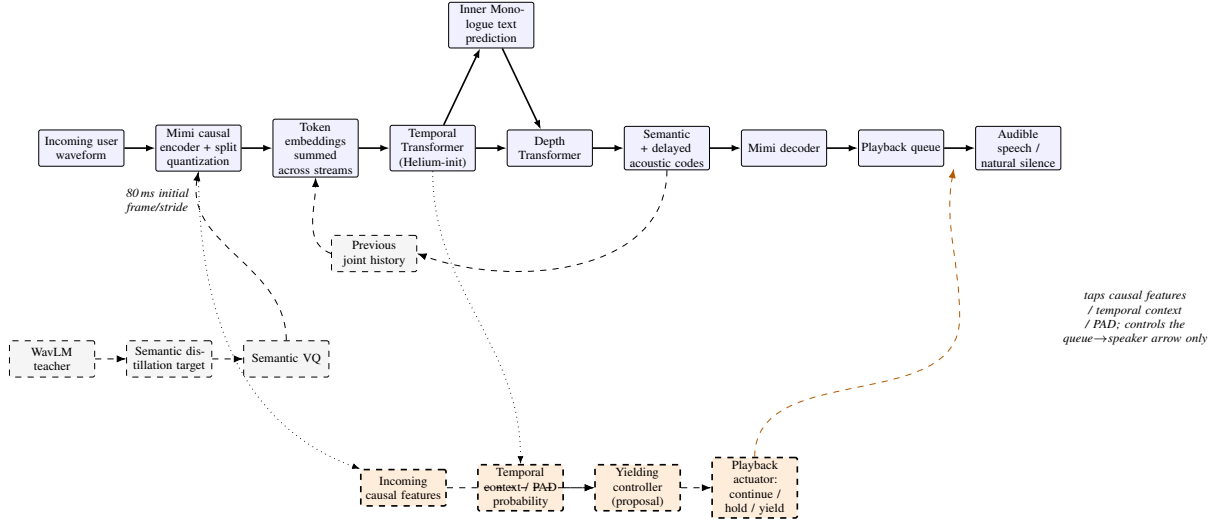
L2. Low transport delay does not establish timely social action. Causal frames and hierarchical prediction (Moshi) or block-wise processing and bounded-context waveform decoding (Qwen) reduce the delay between input and output audio. Yet first-audio latency measures onset, not whether the system stops after a real interruption while talking through a backchannel. Suppressing output whenever speech appears yields fast stopping trivially, and refusing to stop yields few false yields trivially; neither is a better conversation. Any evaluation must therefore compare false yields only under common interruption-latency and missed-interruption limits, and the timing account must include input buffering, controller computation and playback queues. Because stage durations are unreported (above), a component-level timing diagram is more informative than comparing headline figures.

L3. Available cues and supervision may be insufficient. Speech activity identifies that someone is making sound, not why. Distinguishing acknowledgment, hesitation and interruption needs context, and generation state (is the assistant finishing a phrase, waiting, or mid-explanation?) is a candidate cue whose usefulness must be tested. Supervision matters too: Moshi notes limits of simulated dialogue without overlap and Qwen acknowledges noisy speech supervision (M §4.4, pp.20–21; Q §4.3, p.7). A measured gain must therefore be attributed to a controlled intervention, not to differences between training corpora.

5 End-of-Speech Is Four Decisions

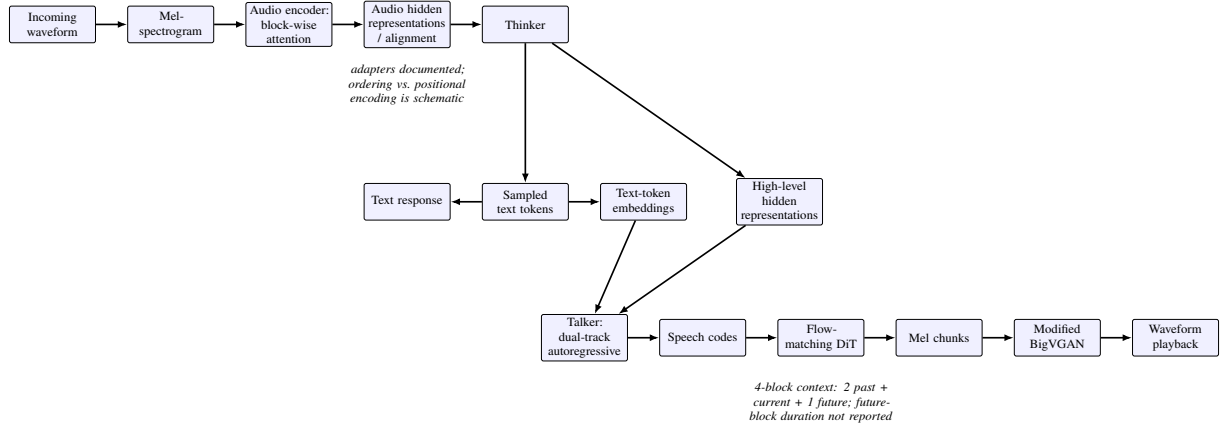
“End of speech” conflates decisions that a conversational system must make separately (Table 2). Moshi establishes continuous listening and generation including silence, while PAD/EPAD concern text timing, not a stand-alone stop command (M §3.4.4, pp.16–17). Qwen2.5-Omni establishes text and message boundaries and streaming waveform

Reported model latency: 160 ms theoretical; 200 ms practical.



Per-stage compute duration (encoder, Transformer, decoder, queue) is not individually reported; only the end-to-end figures above are stated. Dashed gray boxes are training-only; dashed orange boxes are the proposed controller, not an evaluated component.

Input: 2-second attention blocks (not an end-to-end latency measurement). DiT: 4-block context, 2 past + current + 1 future; future-block duration not reported.



Per-stage compute duration and total onset delay are not reported in the source for either path.

Figure 1. Architecture and timing dependencies. Top: Moshi (M §§3.3–3.4, pp.9–17; abstract, p.1). Bottom: Qwen2.5-Omni (Q §§2.1–2.4, pp.3–5; §3, p.6). Arrows show dependencies, not proportional execution durations. Solid = established pipeline; dashed gray = training-only path; dashed orange = the proposed yielding controller (a proposal, not a reported or evaluated component).

Axis	Moshi (Défossez et al., 2024)	Qwen2.5-Omni (Qwen Team, 2025)
1. Encoder and representation learning	Mimi: causal SeaNet convolutional codec (strides 4,5,6,8,2) with 8-layer causal Transformers before and after quantization; 24 kHz audio to 512-d at 12.5 Hz; 8 codebooks of 2,048 entries, 1.1 kbps (M §3.3.1, pp.10–11).	128-channel mel input (25 ms window, 10 ms hop), about 40 ms per representation; audio encoder initialized from Whisper-large-v3 with block-wise attention (Q §2.2, p.4; §3, p.6).
2. Semantic vs. acoustic representations	Split quantizer: one semantic VQ distilled from WavLM (cosine loss, non-causal teacher used in training only) plus a 7-level acoustic RVQ, summed for reconstruction (M §3.3.2, p.12).	Input: encoder representations. Output: discrete codes from qwen-tts-tokenizer; Talker is conditioned on Thinker hidden states (semantic) plus sampled text-token embeddings (phonetic) (Q §2.3, p.5).
3. Speech-to-LLM alignment	No adapter: parallel token streams (both speakers’ audio plus Moshi’s own text) enter a Helium-initialized Temporal Transformer as summed learned embeddings; “inner monologue” aligns self-text via Whisper timestamps and PAD/EPAD tokens (M §3.4.1, §3.4.4, pp.13–17).	Adapters align the audio encoder to the Thinker LLM; they are trained first while the LLM is frozen. Talker consumes Thinker hidden states and text embeddings as a dual-track decoder (Q §2.1, p.3; §3, p.6).
4. Streaming vs. non-streaming	Causal 80 ms frames; user and system streams modeled jointly so listening and speaking overlap; reported latency 160 ms theoretical and 200 ms practical (M abstract, p.1; §3.3.1, p.11).	Block-wise input attention over 2 s blocks; waveform decoding by a DiT over 4 blocks (2 lookback, 1 lookahead) and chunked BigVGAN; no numerical end-to-end latency reported (Q §2.4, p.5).
5. Autoregressive vs. non-autoregressive	Hierarchical autoregression: Temporal Transformer over time, Depth Transformer over codebooks within a step; Mimi decodes the codes (M §3.4.1, pp.12–14).	Thinker and Talker are autoregressive; mel generation is flow-matching (iterative, not token-by-token) followed by BigVGAN (Q §2.1–2.4, pp.3–5).
6. End-of-speech and stopping	No utterance-end token: continuous streams represent speech or silence; PAD/EPAD structure text timing rather than command a stop (M §3.4.4, pp.16–17).	Text EOS and ChatML message boundaries; interruption handling is not specified (Q Fig. 2, p.3; §4.1, p.6).
7. Training objectives and data	Helium text pretraining (2.1T tokens), audio pretraining (7M hours, mostly English), multi-stream post-training, Fisher fine-tuning (2,000 h), synthetic instruction tuning (>20,000 h); cross-entropy with semantic weight 100, acoustic 1 (M §4, pp.17–21).	Frozen-LLM alignment, then all-parameter training (800B image/video, 300B audio, 100B video-audio tokens), context 8k to 32k; Talker: continuation, DPO ranked by WER and punctuation-pause error, speaker tuning (Q §3, §4.3, pp.6–7).

Table 1. Seven-axis comparison. Every cell is a statement from the cited report; consequences are discussed in the text.

Decision	Meaning	Observable contrast
Input end-point	User finished the current contribution	Hesitation pause vs. completed question
Conversational yield	Assistant gives the user the floor	Backchannel vs. genuine interruption
Output completion	Intended response finished playing	Intended end vs. playback and queue state
Session termination	Processing stops running	Explicit stop vs. ongoing listening

Table 2. Four decisions hidden inside “end of speech”.

generation (Q Fig. 2, p.3; §§2.3–2.4, p.5; §4.1, p.6). Neither fully specifies the remaining boundaries.

Figure 2 fixes the annotation targets. They are desired behaviors for a test, not assumed model behavior. A hesitation while the assistant has no active or pending output tests premature response onset and cannot count as a false yield; it is reported separately.

6 Proposal: Generation-State-Aware Yielding

Hypothesis. *A frozen speech LLM’s own generation state improves continue/hold/yield decisions beyond incoming audio and observable playback history, predicting fewer false yields on eligible backchannel and hesitation events than speech-activity rules and an audio-only learned controller, under matched interruption-to-yield latency and missed-interruption constraints.* It is refuted if the state-aware controller cannot improve that constrained comparison on held-out speakers and conversations, or if removing generation state preserves the benefit. This is a proposal: **no result is claimed.**

Controller. Attach a small causal controller to frozen Moshi. Inputs are current Mimi representations and causal speech-activity features, observable assistant playback activity, elapsed activity and silence durations, and queue occupancy. The

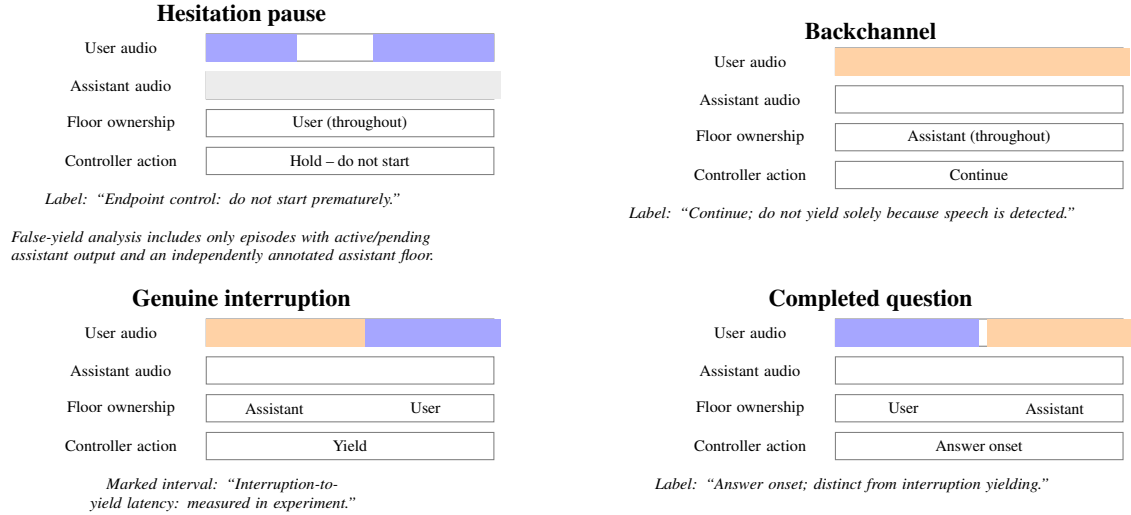


Figure 2. Conversational events and desired actions. Four lanes per panel: user audio, assistant audible audio, desired floor ownership, desired controller action. Time is symbolic (no numerical durations). These are desired behaviors for the proposed test, not measured traces.

generation-state additions are the Temporal Transformer context vector and the self-text PAD probability (M §3.4.1, pp.13–14; §3.4.4, pp.16–17); whether an implementation exposes them must be verified. Only values available at decision time enter the controller. A learned projection and recurrent classifier output **continue** (release audio), **hold** (briefly suspend release, keep listening) or **yield** (suppress output, clear pending playback, keep listening, resume only on a later decision), all through one bounded playback actuator. Hold duration is capped on development data and inappropriate holding counts as disruption, not as a successful non-yield. The contribution under test is the use of generation state, not turn-taking prediction itself.

Training and data. Only the projection and classifier are trained, with action cross-entropy. Labels give event type, floor ownership, onset, acceptable action intervals and interruption deadlines; annotators hear surrounding dialogue and mark ambiguity independently of controller output, and future context may inform labels but never inputs. Data are consented human–Moshi conversations and controlled reenactments recorded as separate user and played-assistant channels, replayed through frozen Moshi for causal feature traces, with Fisher’s separate channels a possible auxiliary source for human cues (M §4.2, p.18). Splits are by speaker, conversation and scenario; thresholds and hold limits are chosen on development data only.

Baselines and primary metric. We compare unmodified Moshi, a prespecified fixed-threshold

speech-activity rule, a development-tuned rule, an audio-only learned controller, and the state-aware controller (Figure 3), all with the same actuator. The primary metric is the unweighted mean of false-yield rates over eligible backchannels and hesitations, where a hold exceeding the annotated tolerance counts as equivalent disruption. Operating points are compared only under common limits on interruption-to-yield latency (annotated onset to actual playback cessation) and missed-interruption rate, set from development-set baseline behavior *before* testing. If no operating point meets both, the benefit is unsupported. Uncertainty is estimated by resampling conversations.

Ablation, robustness, controls. The diagnostic ablation removes generation-state features but keeps acoustic features, playback history, queue state, capacity and training budget, so any remaining benefit is attributable to a learned audio policy or to knowing that the assistant is speaking. Robustness repeats held-out events under identical noise and echo perturbations and adds a Hindi/Telugu subset with fluent annotators, reporting zero-shot transfer separately from adaptation. Checkpoint, generation settings, hardware, recordings, actuator and annotation protocol are held fixed.

Threats. Generation state may encode only timing or speaker identity (addressed by supplying playback history to both learned variants and by speaker-disjoint splits); gains may come from extra waiting (latency constraints and hold accounting); and scripted cues may not transfer (natural held-out conversations). Moshi can also generate

speech internally that suppression keeps the user from hearing, so freezing the model does not repair its internal dialogue state; we track unheard output, resumption coherence and controller overhead as follow-up outcomes.

7 A Reproducible Telugu Speech Pipeline

System. Speech $\rightarrow$ Whisper-family ASR $\rightarrow$ instruction LLM $\rightarrow$ Telugu text response, with both models resident on one GPU and reused across the run. Latency is wall-clock per sample with CUDA synchronization, including audio decoding and re-sampling but excluding model loading.

Model choice under a memory budget. The evaluation GPU (RTX 4060 Laptop, 8 GB) had about 6.5 GB free, too little for the originally configured Whisper-large-v3 (about 3 GB fp16) with Qwen2.5-7B-Instruct (about 15 GB). We used `vasista22/whisper-telugu-small`, a Telugu-only community fine-tune of Whisper-small, and Qwen2.5-1.5B-Instruct (Qwen Team, 2024). The substitution of the fine-tune was forced: the general `openai/whisper-small`, even with Telugu forced, emitted Devanagari-looking text for Telugu audio and looped on short clips. Our first full run under it had corpus WER 2.56 and was discarded. The fine-tune’s `generation_config.json` predates transformers’ language/task validation and raises `ValueError` if either is passed at all (reproduced on transformers 5.15.0 and 4.57.6), so our wrapper treats language: `null` as “pass neither”.

Data. No Telugu speech corpus was available locally, so we synthesized one: 129 template questions with reference answers, comprising 29 Indian state/UT capitals and 100 addition problems spoken as Telugu number words (1–10), rendered with `facebook/mms-tts-tel` (Pratap et al., 2024). Two corpora share this text: **clean**, and **noisy**, with additive white Gaussian noise at 0 dB SNR (noise power equals signal power). An earlier 8 dB corpus changed WER by under 0.002 and was replaced because it exposed no bottleneck. Both exceed the 100-example requirement and give two acoustic conditions. The corpora are *not* byte-identical pairs: TTS duration prediction was unseeded and the noisy set was regenerated, so the clean/noisy gap includes rendering variance. The noisy-baseline vs. denoised comparison reuses identical audio and is exactly matched.

	Clean	Noisy	Noisy+DN
Corpus WER	0.293	0.338	0.339
Corpus CER	0.075	0.096	0.099
ASR time (s)	1.74	2.05	2.03
LLM time (s)	12.71	15.42	15.28
Total time (s)	14.45	17.47	17.31
Pipeline RTF	4.15	4.99	4.95
ASR-only RTF	0.50	0.59	0.58
F1 (ASR input)	0.166	0.155	0.155
F1 (human input)	0.169	0.169	0.169

Table 3. Baseline results (129 samples each). Noisy is 0 dB SNR; DN is spectral-subtraction denoising. Times are means. The human-transcript F1 is the same reference arm in every column.

Evaluation infrastructure (from scratch). WER and CER come from our own unit-cost Levenshtein implementation over NFC-normalized text; corpus rates divide total edits by total reference length, and CER counts Unicode code points without whitespace. Real-time factor (RTF) is pipeline seconds over audio seconds. Response quality is multiset word-overlap F1, computed both for the answer from the ASR transcript and for the answer from the human transcript, which isolates ASR’s downstream effect. Greedy decoding keeps the paired comparison deterministic.

Correctness and hallucination rate. Every reference answer states one checkable fact (a capital name or a sum word). An answer is *correct* if it contains that fact (sums also accept the digit form, matched as a whole number), *abstained* if not correct but containing an uncertainty marker (Telugu for “don’t know”, “sorry”, “not clear”), and *hallucinated* otherwise: a wrong or off-topic answer asserted without hedging despite the prompt’s “ask for clarification, do not invent facts”. The test is lexical, so a correct answer in a different inflection is miscounted, making the rate an upper bound.

Uncertainty. Differences between systems are reported with 95% paired bootstrap intervals (10,000 resamples over the 129 questions, paired by question).

8 Results

All runs completed 129 of 129 samples with no failures.

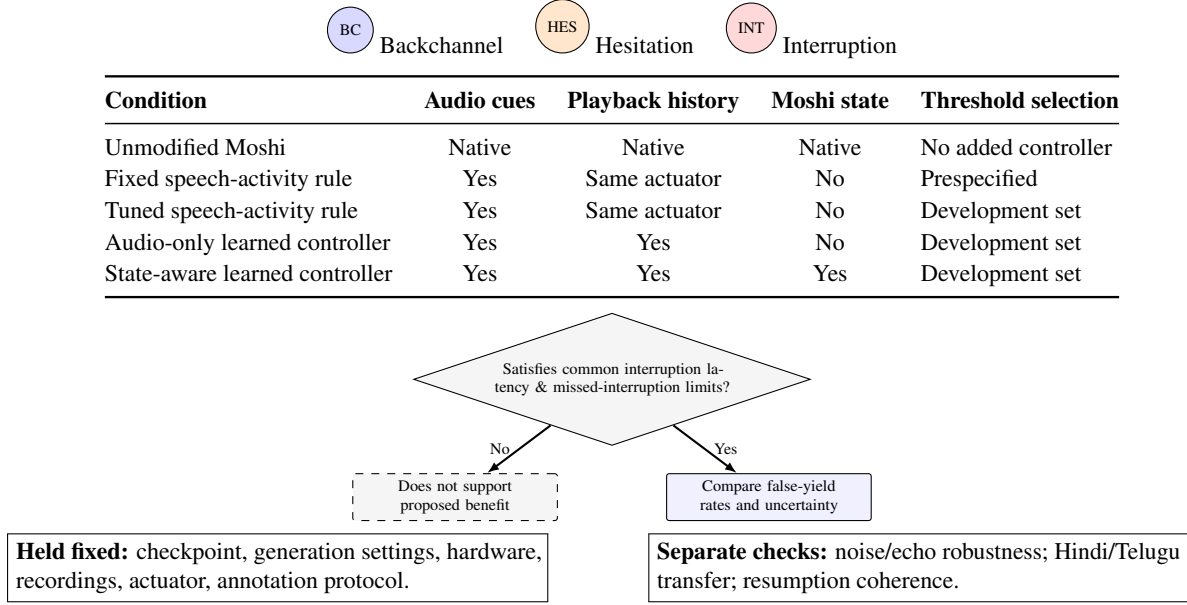


Figure 3. Experimental design matrix. No empirical curve is implied; all rows are proposed conditions, not measured results.

Condition	Input	Corr.	Abst.	Halluc.
Clean	ASR	0	0	129
Clean	human	3	0	126
Noisy	ASR	0	0	129
Noisy	human	3	0	126
Noisy+DN	ASR	0	0	129
Noisy+DN	human	3	0	126

Table 4. Answer correctness out of 129. The hallucination rate is 1.000 from ASR input and 0.977 from human transcripts in every condition.

8.1 Noise Degrades ASR; the LLM Dominates Latency

Noise raises corpus WER from 0.293 to 0.338 and CER from 0.075 to 0.096 (Table 3). The paired bootstrap places the WER increase at +0.045 (95% CI +0.014 to +0.077) and the CER increase at +0.022 (+0.008 to +0.036), so the effect is real even though the renderings differ. The ASR stage is comfortably real time (RTF 0.50), but the full pipeline is not (RTF 4.15 to 4.99) because the LLM stage accounts for about 88% of pipeline time in both conditions. That share is a property of our setup (a 1.5B model on a laptop GPU, greedy decoding, up to 256 new tokens), not of cascades in general.

8.2 Correctness and Hallucination

The LLM, not the ASR, is the dominant failure. Even given the perfect transcript, only 3 of 129

answers state the right fact and mean F1 is 0.169; exact-match is 0 of 129 everywhere. The model *never abstains* (0 of 774 answers across six arms), so every miss is a confident non-answer or wrong answer, typically fluent, repetitive Telugu that re-states the question without naming the capital. ASR errors add little on top: the F1 gap between human and ASR input is 0.003 (clean) to 0.014 (noisy). The smallest Qwen2.5 checkpoint therefore fails at Telugu factual recall and instruction following in this setup, and a larger model, ruled out here by memory, would be the first remedy.

8.3 Research Extension: Denoising as a Bottleneck Fix

Given the measurable noise penalty, we implemented magnitude spectral subtraction (Boll, 1979) as a pre-ASR step: a per-frequency noise floor from a low time-percentile of STFT magnitude, subtracted with a floor to limit musical noise, then resynthesis with the original phase. Table 5 compares it with the baseline on identical audio.

The fix did not help. At default settings WER and CER moved by +0.001 and +0.003, with intervals comfortably containing zero, so we cannot distinguish the effect from sampling noise. A small subset sweep shows mild subtraction helping slightly and aggressive subtraction hurting, consistent with over-subtraction introducing musical-noise artifacts that a neural recognizer handles worse than the original stationary noise; with only

Comparison	Δ	95% CI
Corpus WER	+0.0014	[−0.014, +0.017]
Corpus CER	+0.0028	[−0.004, +0.010]
<i>25-question ASR-only sweep (mean WER)</i>		
No denoising	0.543	
Mild ($\alpha=1.5$, floor 0.05)	0.530	
Aggressive ($\alpha=3.0$, floor 0.02)	0.557	

Table 5. Denoising ablation on identical noisy audio; Δ is denoised minus not denoised. Both intervals contain zero. The sweep covers the first 25 questions only (capitals), and “mild” is the shipped default.

25 questions this is suggestive, not established. The filter does what it should at signal level (RMS 0.228 to 0.177; squared error to an independent clean rendering 0.094 to 0.073 on one sample), so the negative result concerns whether that cleanup survives through the recognizer, not whether the code works. Latency overhead was negligible (mean ASR time 2.05 s vs. 2.03 s).

9 Discussion

The two parts meet at one observation: neither a faster first token nor a cleaner front end addresses what actually limits these systems. In Part I, headline latency says little about floor control; in Part II, where a cascade has no floor-control decision at all, the measured bottlenecks were the LLM’s factual reliability and its generation time, with acoustic noise a smaller and (via our denoiser) not-yet-remedied third. We caution against reading Part II as support for Part I: they test different systems.

For the proposal, Part II offers a practical lesson about evaluation design. Our first plausible-looking configuration produced a WER of 2.56 and our first noise level produced no effect; both were visible only because the harness reports per-sample outputs and a clean-versus-ASR comparison. We would build the same safeguards (state-removal ablation, matched constraints, held-out speakers) into any run of the yielding experiment before trusting its primary metric.

Limitations

Part I is analysis and a proposal; it contains no experiment and claims no gain, dataset size, parameter count or runtime. Statements about the models rest on the two attached reports and may not reflect later releases. Implementation access to Moshi’s temporal context and PAD probabil-

ity, Fisher licensing, and the effect of suppressing audio on Moshi’s internal state remain to be verified, and novelty of the state-aware combination is unchecked. **Part II** uses synthetic TTS speech, not real speakers, so it says nothing about accent, prosody or channel variation. The models are small because of a 6.5 GB memory budget and would likely change both the WER and the factual accuracy substantially at larger scale. Clean and noisy corpora are not matched recordings. Response quality is a lexical proxy and the hallucination check is a substring heuristic, not a semantic judge. Results are from a single deterministic greedy run per condition, with intervals reflecting only question sampling. The ASR checkpoint’s revision is unpinned because its configuration files and weights resolved to two different main commits within one session, and pinning either broke loading.

Ethical Considerations

All evaluation audio is machine-synthesized from template text, so no personal data or human recordings were collected. The proposed human–Moshi data collection (Part I) would require informed consent and a data-handling review before any run. Model checkpoints are used under their published licenses.

References

- Steven F. Boll. 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 27(2):113–120.
- Alexandre Défossez et al. 2024. Moshi: a speech-text foundation model for real-time dialogue. arXiv:2410.00037.
- Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a Transformer-based language model for predicting turn-taking in spoken dialog. In *Findings of EMNLP*.
- Erik Ekstedt and Gabriel Skantze. 2022. Voice Activity Projection: self-supervised learning of turn-taking events. arXiv:2205.09812.
- Tu Anh Nguyen et al. 2023. Generative spoken dialogue language modeling. *Transactions of the ACL*, 11.
- Vineel Pratap et al. 2024. Scaling speech technol-

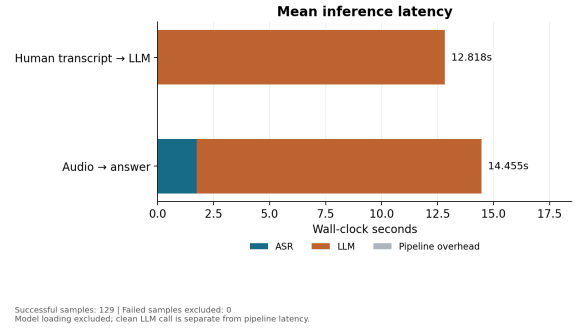
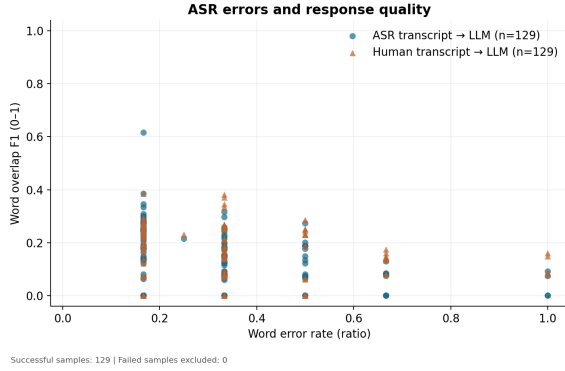


Figure 4. Clean condition. Left: per-sample WER against response F1 for ASR-transcript and human-transcript inputs (many samples share a WER because questions are short). Right: mean latency; the LLM stage dominates. Noisy-condition plots are in Appendix D.

ogy to 1,000+ languages. *Journal of Machine Learning Research*, 25(97).

Qwen Team. 2024. Qwen2.5 technical report. arXiv:2412.15115.

Qwen Team. 2025. Qwen2.5-Omni technical report. arXiv:2503.20215.

Alec Radford et al. 2023. Robust speech recognition via large-scale weak supervision. In *ICML*.

Zhifei Xie and Changqiao Wu. 2024. Mini-Omni: language models can hear, talk while thinking in streaming. arXiv:2408.16725.

A Reproducibility Details

Environment. Python 3.13.4, PyTorch 2.11.0+cu128, transformers 4.57.6 (the package pins $\geq 4.45, < 6$), fp16 on an RTX 4060 Laptop GPU (8 GB), greedy decoding with a warm-up pass. Qwen2.5-1.5B-Instruct is pinned to commit 989aa798. The Whisper fine-tune is left at main (see Limitations). The noise generator is seeded (42); TTS rendering is not.

Commands. From the repository root:

```
python synthesize_corpus.py --snr-db 0
python prepare_data.py --source corpus_clean \
  --output data/test_clean.json --limit 0
python prepare_data.py --source corpus_noisy \
  --output data/test_noisy.json --limit 0
python evaluate.py --dataset data/test_clean.json \
  --config config.yaml \
  --output-dir outputs/evaluation/clean --warmup 1
python evaluate.py --dataset data/test_noisy.json \
  --config config.yaml \
  --output-dir outputs/evaluation/noisy_baseline \
  --warmup 1
python evaluate.py --dataset data/test_noisy.json \
```

```
--config config_denoised.yaml \
--output-dir outputs/evaluation/noisy_denoised \
--warmup 1
python hallucination.py --reports <report.json ...>
python bootstrap.py --a <report.json> --b <report.json>
python -m unittest discover -s tests
```

The repository README documents each step; the unit tests cover the metric implementation, dataset preparation, plotting, the denoiser, the hallucination classifier and the bootstrap. The generated corpora and outputs/ directory are not committed and are regenerated by the commands above.

B Additional Architectural Evidence

Moshi. Mimi’s encoder uses four convolutional blocks with strides (4,5,6,8) and a final stride-2 convolution; WavLM teacher features are 1,024-d at 50 Hz pooled to 12.5 Hz. The joint sequence has $K = 2Q + 1$ streams with $Q = 8$, and actual user audio replaces predicted user-stream tokens during interaction (M §3.4.4, p.17). Acoustic delay is 2 in pretraining and 1 in fine-tuning (M §5.3, p.27). Separate ASR and TTS configurations use a 2 s text/audio delay and differ from the conversational configuration (M §5.7, pp.31–32). Training data figures (7M, 2,000, >20,000 and a separate 170-hour set used for TTS training) overlap and must not be summed to a unique-data total.

Qwen2.5-Omni. Audio is resampled to 16 kHz; the Qwen2-Audio encoder is initialized from Whisper-large-v3 and adapters are trained before the encoders against an initially frozen LLM (Q §2.2, p.4; §3, p.6). Input attention uses 2 s blocks, and audio-video time interleaving separately uses 2 s groups. Context grows from 8,192 to 32,768 tokens (Q §3, p.6). The exact position of adapters relative to positional encoding is schematic in Figure 1.

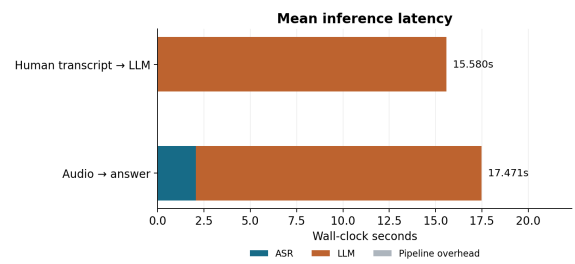
C Evaluation Protocol Definitions for the Proposal

Annotation. Event class and desired action are annotated separately: a brief utterance can be a takeover and a longer acknowledgment can be non-interrupting. Annotators record user and assistant activity and floor ownership, event onset and the earliest point the intended action becomes inferable, acceptable continue/hold/yield intervals and interruption deadlines, whether assistant output was active or queued, and ambiguity (including language-specific ambiguity). Eligibility is fixed before inspecting controller outcomes.

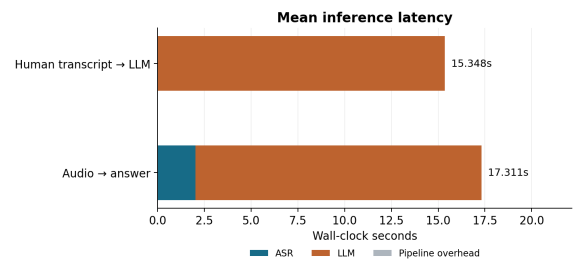
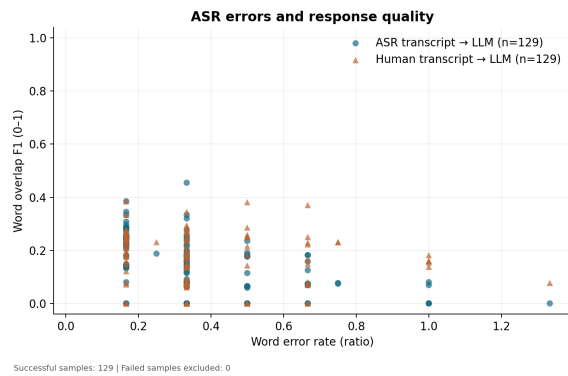
Metrics. False-yield rate = events with an inappropriate yield or excessive hold / eligible floor-retention events. Missed-interruption rate = genuine interruptions without timely playback cessation / genuine interruptions. Latency uses actual playback cessation including queued audio and actuator delay. Resumption errors are reported separately because yielding correctly does not guarantee coherent subsequent speech.

Replay vs. interactive. Immediate-action comparisons start systems from a common assistant prefix and replay the same user event, enabling paired evaluation. Replay cannot reproduce how people react to different assistant actions, so follow-up live trials with counterbalanced order must assess resumption and dialogue drift, and the two kinds of outcome are reported separately. For Hindi/Telugu, transfer is evaluated before adaptation, with agreement and instruction comprehension reported separately so poor performance is not automatically attributed to turn-taking control.

D Noisy-Condition Plots



Successful samples: 129 | Failed samples excluded: 0
Model loading excluded; clean LLM call is separate from pipeline latency.



Successful samples: 129 | Failed samples excluded: 0
Model loading excluded; clean LLM call is separate from pipeline latency.

Figure 5. Top: noisy baseline (0 dB SNR, no denoising). Bottom: noisy with spectral-subtraction denoising. Same axes and layout as Figure 4.