BATCHED SPEECH DECISIONS WITHOUT DECODING: SINGLE-TOKEN SUPERVISION LETS A FROZEN LLM HEAR BEYOND THE TRANSCRIPT

Jie Jin[1,⋆], *Ziyin Ma*[2], *Min Yin*[3], *Jinyu Chen*[4], *Haigang Song*[1], *Zhikun Pang*[1], *Xiaowen Zhang*[1]

[1]Adventists.ai, Melbourne, Australia
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3]Future Design Lab, Innovation Center of Yangtze River Delta, Zhejiang University, Jiaxing, China
[4]University of Virginia, Charlottesville, VA, USA
⋆Corresponding author: jiejin@adventists.ai

ABSTRACT

Full-duplex voice agents make many small, closed decisions, which current systems answer by slow autoregressive decoding. We propose DuplexJev, which feeds ASR-encoder hidden states through a small connector into a frozen LLM and reads each question as a single-token distribution over its options. Nothing is decoded, and an 8-GPU node answers 80 decisions about eight utterances in about 0.1 s. With a last-layer connector, spoken QA stays close to reading the transcript (90% vs. 91%). DuplexJev also hears the speaker: gender and emotion accuracy both reach 90% (from 55% and 28%) with a cross-attention connector, whose spoken QA drops by only 1 point (83% to 82%). We train decisions with cross-entropy on the read-out answer token, instead of the usual transcript distillation, whose teacher never hears the voice, and keep distillation for content. Encoders and LLMs are interchangeable; we release weights, training recipe, a batched-inference pipeline for full-duplex serving and a bilingual spoken-QA set.

***Index Terms*—** full-duplex dialogue, speech LLM, speech-to-decision, decision models, paralinguistics

1. INTRODUCTION

A full-duplex voice agent listens while it speaks. Around every user utterance it must answer small, closed questions: *Is the turn complete? Is this a backchannel or a barge-in? Which pre-recorded filler fits? Who is speaking?* Most of this work is not *speech-to-text* but *speech-to-decision*: the agent rarely needs the transcript, only the answer to a question about it. The cost is real: in a deployed in-car assistant (Sec. 3), such decisions are made by generating text and consume 30% of all LLM input tokens. These decisions block the reply: the agent cannot speak until they are answered, so their latency adds to every turn. Keeping them fast under peak load requires spare capacity, which makes them a major share of both latency and cost.

Current systems pay for these decisions with autoregressive decoding. End-to-end full-duplex models [1, 2, 3] decode on every frame of every call, so each call holds its own decoding stream and batched serving is expensive; their decisions are also not exposed as typed outputs. Cascades transcribe and then generate an answer, which is slow and discards everything in the voice. Dedicated turn-taking classifiers [4] are cheap but answer one fixed question each.

We ask whether a frozen, general-purpose LLM can answer such questions directly from speech, in the spirit of the "System-One" decision models [5, 6] that motivated this work: given a state and a question declared at run time, return a distribution over its options without generating text. If speech enters the LLM as continuous embeddings, no ASR decoder is needed. If each question lists its options under letters, the answer is the next-token distribution over those letters, so nothing is decoded. Such one-step questions batch across questions and calls, and questions that share a long dialogue context can be packed into one row (Fig. 1).

Reading decisions from audio exposes a second problem. Connectors that feed speech into a frozen LLM are usually trained by distillation from the LLM's own output on the transcript [7, 8]. This teacher never hears the voice, so the student cannot learn what is not in the words: after 6k steps, speaker gender is still at chance, although a linear probe reads it at 97% from the encoder (Fig. 2). Supervising only the answer token that is read out, while keeping distillation for content, fixes this.

Contributions.

(1) **Decoding-free batched decisions.** One forward pass, no decode steps, and exact prefix sharing for long contexts; an 8-GPU node answers 80 decisions about eight utterances in about 0.1 s.
(2) **Why speech LLMs cannot hear the speaker, and a fix.** Answer-token supervision teaches a frozen LLM gender and emotion through two connector designs; controlled comparisons of objectives and layer-wise probes locate the bottleneck in the objective, not the connector.
(3) **Modularity and release.** Any ASR encoder and any frozen LLM are joined by a small connector. We release the weights, training recipe, a batched-inference pipeline for full-duplex serving and the qa100 test set.[1]

2. METHOD

2.1. Connector

A frozen ASR encoder produces layer-wise hidden states $h^{\ell}_{1:T}$. We compare two ways of turning them into LLM input embeddings (Fig. 1). **B (last layer)** feeds h^{L} to an Ultravox-style projector [7] (frame stacking + MLP) into the LLM embedding space ($d = 5120$). **A (cross-attention fusion)** adds a fusion block before the projector: queries come from the final layer ($\ell = 18$ for Qwen3-ASR-0.6B [9]), keys from layer 14 and values from layer 9. The output projection is zero-initialised and added residually to h^{L}, so A starts identical to B. A tests the common intuition that ASR fine-tuning

[1]`https://github.com/adventists-ai/duplexjev`; `pip install duplexjev`.

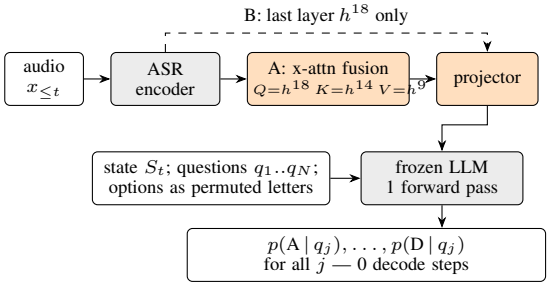

Fig. 1. DuplexJev. Only the connector (fusion block and projector, orange) is trained. Variant A fuses three encoder layers by cross-attention; variant B projects the final layer only. Every question is read as a closed-set next-token distribution. Content is aligned by transcript distillation; decisions are trained with answer-token cross-entropy (Sec. 2.1).

pushes the final layer towards linguistic content while intermediate layers keep speaker and prosodic cues [10]; for this encoder our probes do not support it (Sec. 4.3). Both LLM and encoder stay frozen; only fusion and projector are trained.

Training. For content we align the connector by transcript distillation [11, 7] (R1–R2): given the audio embeddings, the frozen LLM should match the distribution it produces when the audio is replaced by the gold transcript (token-level KL, temperature 2). This keeps the LLM's behaviour on content intact, but the teacher never hears the voice: for any question whose answer is not in the words (speaker, emotion), its target is the text-only prior, and the student is trained to be exactly as uninformed. DiVA [8] aligns a connector to a frozen LLM with the same kind of text-conditioned distillation; DeSTA2 [12] writes paralinguistic metadata into generated text descriptions used as targets; speech LLMs trained on labelled tasks [13, 10] learn such cues with cross-entropy on generated answers. We differ in supervising only the single token the readout uses, which needs neither descriptions nor decoding, in showing with probes that the objective and not the connector blocks the cue, and in a mixed objective that keeps distillation for content. For decisions we therefore introduce two ingredients. *Answer-token supervision*: cross-entropy on the single answer letter, at the position and under the template that the readout of Eq. (1) uses, with the fixed chat-template tokens before it (e.g. an empty reasoning block) excluded, so the letter is the only supervised token. *Mixed objective*: each sample keeps its own loss; content samples (transcription, continuation) are distilled, decision samples use answer-token cross-entropy, and the two averages are added.

2.2. Typed single-token readout

A request holds a state S_t (the audio up to time t, plus optional agent state) and N typed questions. Each *choice* question with K options is rendered as a prompt that lists the options under letters A, B, …, with the letter order randomly permuted to remove position bias. *Boolean* questions use N/Y and *score* questions use $1..K$. We run a single forward pass, take the logits at the answer position, restrict them to the K option tokens and renormalise:

$$p(o_k \mid S_t, q_j) = \mathrm{softmax}_k\left(z_{\pi_j(k)}\right), \tag{1}$$

where π_j maps options to letters. Nothing is sampled: the typed output is always valid, and $\max_k p$ gives a confidence score that can be used to abstain.

2.3. Batching and cost

Because each question needs one step, N questions over the same state, or over different calls, can be read together. The simplest implementation copies the whole prompt into every row (*batch-repeat*), so each row re-encodes the audio and any dialogue context. With *prefix sharing*, the longest common prefix of the N prompts (template, dialogue context, audio) of length P is encoded once into a KV cache. The N question suffixes are then concatenated into a single row; a block-diagonal 4-D mask lets each suffix attend to the prefix and causally to itself, but not to other suffixes, and position ids restart at P for every suffix. The answer to question j is read at the last token of its suffix. The pass costs $\approx C(P) + \sum_j C(L_j)$ instead of $N\,C(P+\bar{L})$, and the KV cache holds $P+\sum_j L_j$ positions rather than $N(P+\bar{L})$, so long shared contexts do not multiply memory. The construction is exact up to numerical precision (Sec. 4).

By contrast, a full-duplex model spends a decoding step on every frame of every call whether or not a decision is due, and a cascade spends $O(\text{transcript} + \text{answer})$ autoregressive steps per decision.

2.4. Deployment pattern

Because a pass is one step with no decoding, DuplexJev could also run on a fixed clock rather than per detected utterance: every tick (e.g. 160–200 ms) the server collects the newest audio of all connected calls and answers their decision sets in one batched pass, with turn-taking (turn complete, backchannel, barge-in) as one of the questions, so that it could replace a separate VAD or end-pointer; we do not evaluate streaming input here. Business decisions (filler, routing, history) share the pass. Fillers show the gain: instead of generating one (the deployed assistant spends a 5k-token LLM call on a 20-token filler), a fitting filler is *selected* from a large pre-recorded library, by boolean rows over a shortlist or a coarse-to-fine tree of lettered questions that grows without retraining, at the cost of one readout (Table 1).

3. EXPERIMENTAL SETUP

Models. The LLM is Qwen3-32B [14], frozen, in bf16. The encoder is Qwen3-ASR-0.6B, frozen. Trainable parameters are 21.1 M for A (fusion 3.2 M, projector 17.8 M) and 17.8 M for B. A and B are trained from scratch with identical data and schedule (A on 8×H200, B on 8×A800; global batch 16, lr $2{\times}10^{-4}$, 1k warm-up, cosine decay, 32k steps per round). **R1–R2** use only content tasks, transcription and continuation, with the distillation objective; the data are a public ASR/translation mixture of WenetSpeech [15], GigaSpeech, LibriSpeech [16], Common Voice [17] and CoVoST, split into 100 disjoint packs of which R1 and R2 each use one (0.5 M utterances per round). **R3** is a short paralinguistic stage that starts from the R2 connector (batch 16, lr 10^{-4}). The *gender pack* has 32k real utterances from AISHELL-1 [18] and LibriSpeech with corpus-provided labels; the *emotion pack* has 26k utterances with four classes (neutral, happy, angry, sad) from ESD [19] (Chinese and English) and CREMA-D [20] (English). Every clip serves three tasks in equal shares: transcription, continuation, and one closed-set question with permuted option letters and paraphrased stems. Test speakers are disjoint from all training speakers and no synthetic voices are used. With everything else fixed we compare distillation (6k steps) with answer-token supervision for gender (6k) and emotion (3k), and a four-task mixture (a content pack plus both decision tasks, 4k steps) under all-cross-entropy (MIX-CE) or the mixed objective (MIX-KD). To test an encoder swap we plug in the released

Whisper-large-v3-turbo encoder and projector [21, 7], trained for the same Qwen3-32B, without modification.

qa100 (released). 100 four-way multiple-choice questions: 50 Chinese and 50 English, each language split into 25 logic and 25 factual items, with the gold letter balanced over A–D. Question stems are synthesised with VoxCPM2 [22]; options are given as text. Every clip is verified by an independent ASR (exact normalised match, re-synthesised otherwise). Three conditions share the same prompt: *audio*; *text*, where the oracle transcript replaces the audio (an upper bound for hearing); and *options-only* (a lower bound, 0.36).

Content understanding. Besides qa100 we use the main-language part of the ZJU audio benchmark v2.0.0 [23] (ZJU-ML): 100 four-way questions, 50 Chinese and 50 English, whose 50 factual items are real recordings (ODSQA, SD-QA) and 50 math and logic items TTS. We also use Easy-Turn [4] for zero-shot turn-taking.

Paralinguistics. *Gender*: our 800-utterance real-speech set (AISHELL-1, LibriSpeech, Common Voice; balanced by gender and language) and the independently constructed ZJU audio-gender-benchmark v1.2.0 [23] described next. *Emotion*: 800 utterances balanced over the four classes from held-out ESD speakers (two per language) and CREMA-D actors. Transcript-only controls are at chance by construction, since both emotion corpora read the same sentences in every emotion.

Gender (ZJU). The gender part of the same benchmark (v1.2.0) [23]: 100 clips, 50 TTS and 50 real speech (MagicData-RAMC Chinese; SLURP and AMI English), balanced by gender, scored with the official script. We report accuracy on all 100 clips and on the real-speech subset, together with a *text-only* control, which is at chance by construction and confirms the decision is made from the voice.

Production workload. Dimi, an in-car voice assistant operated by DoumaoAI (Qwen3-ASR-0.6B $\rightarrow$ 72B LLM $\rightarrow$ TTS; 1.45 M user turns from 3,721 devices over 91 days), issues up to three LLM calls per turn only to make small decisions: a tool-need check (median 1,471 input tokens), an intent label (530) and a 20-token opening filler (5,020). The median time from end of speech to first audio is 1.29 s, and 1.9% of utterances already call a separate speaker-identification model.

Cost. All timings use one H200 and bf16. DuplexJev latency is the median of three timed passes after one warm-up in plain PyTorch (eager attention, no CUDA graphs), excluding audio loading. The *cascade* baseline answers ten questions per event (e.g. filler among 8, route among 6, urgency 1–5, turn complete): Qwen3-ASR-0.6B transcribes greedily, then Qwen3-32B served with vLLM [24] (prefix caching, greedy) generates a 10-field JSON; all 30 outputs parse. Events are 30 real utterances, 10 each of 2–3, 4–6 and 8–12 s, from the speaker-disjoint gender test set. To separate the readout from the implementation, we also run the same ten questions on the same vLLM engine as single-token readouts (one lettered prompt per question, one output token, answer from the logprobs), with the transcript in place of the audio embeddings. Our connectors feed 6.25 audio tokens/s (stack factor 2), 10–33 more tokens per row than the transcript here (rows of 47–67 tokens), so this slightly underestimates the cost of DuplexJev on an optimised engine, before the saved ASR; the PyTorch timings and prefix-sharing runs use earlier connectors with stack factor 8 ($\approx$1.6 tokens/s). We repeat both with a 1,471-token dialogue context, the median input of the production tool-need check.

Table 1. Ten decisions per event, one H200, Qwen3-32B on vLLM, 30 real utterances. *JSON*: one JSON answer generated from the ASR transcript (ASR, 411 ms, not included). *Readout*: one single-token prompt per question on the transcript (approximates DuplexJev). Capacity: events/s whose batch finishes within the budget.

		decode	1 event	events/s within		
ctx	method	steps	(ms)	0.25 s	0.5 s	2 s
0	JSON	86	1567	0	0	16.9
0	readout	0	92	17.3	20.4	33.5
1.5k	JSON	87	1598	0	0	8.4
1.5k	readout	0	144	9.7	12.0	18.4

4. RESULTS

4.1. Batched readout and cost

Latency and capacity. On the same engine, reading ten decisions as single tokens takes 92 ms per event, against 1.57 s to generate them as JSON, a 17$\times$ reduction before counting the 411 ms of autoregressive transcription the cascade also needs (Table 1). With a 1,471-token dialogue context, the median of the production tool-need check, the gap is 11$\times$. The JSON cascade cannot serve any load within 0.5 s, whereas the readout serves 17–20 events/s (10–12 with context) on one GPU. Replicated on one 8$\times$H200 node (one engine per GPU), eight events (80 decisions) finish in 94–103 ms, and the readout serves 161 events/s, i.e. 1,613 decisions/s, within 0.5 s (100 events/s with the 1.5k context) and saturates at 3,684 decisions/s. At full saturation (64 simultaneous events) the two are on par (335 vs. 311 decisions/s without context), since batched decoding is cheap and one JSON prompt amortises the dialogue context over all ten questions, while our readout repeats the transcript and instructions in every question row. The advantage is latency and capacity under a conversational budget, not raw throughput.

Prefix sharing is exact and removes the context cost. Shared and one-by-one readouts give the same answer on 195 (A) and 198 (B) of 200 question–utterance pairs (20 utterances $\times$ 10 questions), and on 40 of 40 with 0.5–5k-token contexts. All differences but one (0.31) are near-ties (margin $\leq$ 0.06; median $|\Delta p|$ 0.01–0.02, bf16 noise). Without context the audio prefix is short (stack factor 8: $\approx$1.6 audio tokens/s, 36–66 prefix tokens), so the question text dominates and sharing halves the cost of 100 questions over one 2.6 s utterance (3.1 s batch-repeat vs. 1.6 s; 7.6 s sequential). With a production-sized context, batch-repeat re-encodes the context in every row (158k tokens for 100 questions over 1.5k) and runs out of memory already at 5 questions over a 5k context, while the packed row keeps the KV at $P + \sum_j L_j$ positions: 100 questions over a 1.5k context take 5.1 s instead of 37.9 s sequentially, and 50 over a 5k context 5.5 s instead of 112 s.

Batching does not change answers. Across different utterances in one batch, batched and one-by-one answers agree on 790 of 800 real-speech gender items and on 99–100 of 100 ZJU items, and qa100 scores move by at most one item.

4.2. Content understanding

The interface does not cost understanding. With the last-layer connector (B), after only two 1% packs, DuplexJev answers 90 of 100 spoken questions against 91 when reading the oracle transcript (Table 2); English is identical, and the text-condition errors are arith-

Table 2. Content-understanding accuracy (%). qa100: released spoken QA (/100, zh/en /50 each). ZJU-ML: spoken questions from [23], half real, half TTS. Easy-Turn: zero-shot 4-way turn state. Our connectors are shown after R2 (two 1% packs).

input	qa100	zh / en	ZJU-ML	Easy-Turn
oracle transcript	91	45 / 46	89	80.5
options only	36	18 / 18	39	–
Whisper, released (Qwen3-32B)	89	43 / 46	85	76.1
Whisper, released (Gemma-3-27B)	92	–	77	–
B last layer, R2	90	44 / 46	79	76.1
A x-attn, R2	83	39 / 44	77	77.1

Table 3. Paralinguistic accuracy (%) and change in content understanding. Gender: our 800 real utterances / ZJU native / ZJU lettered. Emotion: 4-way, 800 utterances. Δ: change vs. the R2 start of the same connector. Probes: logistic regression on pooled features.

	gender	ZJU n/l	emotion	Δqa100	ΔZJU-ML
probe, encoder last layer	97.5	–	92.5		
probe, connector A / B	98.1 / 96.5	–	87.9 / 88.3		
Whisper, released connector	55.4	51	27.1		
A after R2	55	53 / 50	28		
B after R2	54	50 / 51	27		
A, distillation, 6k	53	49 / 50	–		
B, distillation, 6k	53	52 / 52	–		
A, answer CE, gender, 6k	89.4	73 / 67	27	+3	+2
B, answer CE, gender, 6k	87.9	60 / 57	26*	−3	+2
A, answer CE, emotion, 3k	49†	40 / 43	71.8	+1	−6
B, answer CE, emotion, 3k	–	–	85.5	−1	−10
A, MIX-CE, 4k	89.5	81 / 76	88.9	−4	−17
A, MIX-KD, 4k	89.9	90 / 86	90.0	−1	−16

*B with the 80%-gender variant. †below chance: emotion training alone biases the gender answer.

metic or date items a single token without reasoning misses. A reaches 83; both reach 77–79 on ZJU-ML, below the released Whisper connector (85), trained on far more data. Swapping parts leaves the readout unchanged: the Whisper connector gives 89, and its Gemma-3-27B version 92 (transcript bound 93). Zero-shot turn-taking reaches 76–77% on Easy-Turn [4] from audio against 80.5% from text; recall on *wait* stays near 0.3 even from text, a limit of the LLM's reading rather than of hearing. Confidence gates abstention: answering only at $p \geq 0.95$ covers 87% of qa100 at 93% accuracy (B).

4.3. Paralinguistics: distillation vs. answer supervision

Distillation cannot teach what the transcript lacks. Starting from R2, both connectors were trained for 6k steps on the gender pack with the distillation objective. Gender stays at chance (53%) on real speech and on ZJU, and the answers keep the text prior (female recall 0.3, male 0.8). The failure is not a matter of data, capacity or schedule: a gender-only mix, a 5× higher learning rate, a re-start from a random connector, and evaluation on the *training* utterances with the training prompt (56%) all leave it at chance, while the loss keeps falling as the student matches its uninformed teacher.

Answer-token supervision teaches it. Changing only the loss, the same data and schedule take gender to 89% (A) and 88% (B) and emotion to 72% (A) and 86% (B) (Table 3, Fig. 2). With 80% gender questions the cue is learned within 250 steps (A: 92%, ZJU native 97%). Gender training moves content scores by −3 to +3 points.

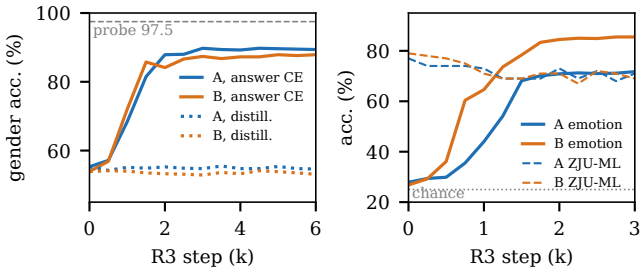

Fig. 2. R3 from the R2 connector, same data and schedule. Left: gender on 800 real utterances under distillation (flat) and answer-token supervision, for A and B. Right: emotion under answer-token supervision, with ZJU-ML (dashed).

Cross-task checks show that the connector learns the cue, not an answer bias: gender-trained models stay at chance on emotion (26–29%), and the emotion-trained model does not know gender.

The connector is not the bottleneck. Probes find gender (97–99%) and emotion (88–93%) already linearly decodable in the encoder's last layer and in the embeddings both connectors hand to the LLM. Consistently, A and B learn gender at the same rate (2k steps: 88 vs. 84%), and on emotion B is even better: the objective decides.

Keeping content understanding. On qa100 all answer-supervised runs stay within 3 points of their start (all-cross-entropy mixing: −4; Table 3). ZJU-ML is more sensitive. Gender training alone is neutral on it (+2), whereas emotion data cost 6–10 points and both four-task mixtures 16–17, almost all on its real-speech factual half (e.g. MIX-KD: −15 of 50 there, −1 on the TTS half), in Chinese and English alike. The mixed objective does not reduce this drift, but it keeps qa100 (−1 vs. −4) and gives the best paralinguistic scores (gender 90, ZJU 90/86, emotion 90).

5. CONCLUSION

DuplexJev reads many typed decisions from speech in one forward pass of a frozen LLM. Transcript-distilled speech LLMs are deaf to the speaker because their teacher cannot hear the voice; supervising the token that is read out lets the LLM use the gender and emotion cues the encoder carries. On-device gating with small LLMs is next.

6. COMPLIANCE WITH ETHICAL STANDARDS

Public corpora and benchmarks are used under their licences (emotion corpora for research only); no new human-subject data were collected. Labels are corpus-provided; the classifiers study information flow, not individuals.

7. ACKNOWLEDGEMENTS

This work was funded by Adventists.ai. Claude (Anthropic) assisted with code.

8. REFERENCES

[1] Tu Anh Nguyen et al., "Generative spoken dialogue language modeling," *Transactions of the Association for Computational Linguistics*, vol. 11, 2023.

[2] Alexandre Défossez et al., "Moshi: a speech-text foundation model for real-time dialogue," *arXiv preprint arXiv:2410.00037*, 2024.

[3] Bandhav Veluri et al., "Beyond turn-based interfaces: Synchronous LLMs as full-duplex dialogue agents," in *Proc. EMNLP*, 2024.

[4] Guojian Li 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.

[5] TypeSafe AI, "Jev: A system-one model for typed decisions," `https://typesafe.ai/`, 2026.

[6] "OpenJev: Typed decisions via diffusion canvas readout," `https://github.com/razorback16/openjev`, 2026.

[7] Fixie.ai, "Ultravox: A fast multimodal LLM for real-time voice," `https://github.com/fixie-ai/ultravox`, 2025.

[8] Will Held et al., "Distilling an end-to-end voice assistant without instruction training data," in *Proc. ACL*, 2025.

[9] Qwen Team, "Qwen3-ASR technical report," *arXiv preprint arXiv:2601.21337*, 2026.

[10] Changli Tang et al., "SALMONN: Towards generic hearing abilities for large language models," in *Proc. ICLR*, 2024.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] Ke-Han Lu et al., "DeSTA2: Developing instruction-following speech language model without speech instruction-tuning data," in *Proc. ICASSP*, 2025.

[13] Yunfei Chu et al., "Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[14] Qwen Team, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[15] Binbin Zhang et al., "WenetSpeech: A 10000+ hours multi-domain Mandarin corpus for speech recognition," in *Proc. ICASSP*, 2022.

[16] Vassil Panayotov et al., "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[17] Rosana Ardila et al., "Common voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2020.

[18] Hui Bu et al., "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, 2017.

[19] Kun Zhou et al., "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," in *Proc. ICASSP*, 2021.

[20] Houwei Cao et al., "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Transactions on Affective Computing*, vol. 5, no. 4, 2014.

[21] Alec Radford et al., "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.

[22] Yixuan Zhou et al., "VoxCPM: Tokenizer-free TTS for context-aware speech generation and true-to-life voice cloning," *arXiv preprint arXiv:2509.24650*, 2025.

[23] "ZJU audio benchmark: Gender (v1.2.0) and main-language (v2.0.0) parts," `https://github.com/Vsky-morigen/audio-gender-benchmark`, 2026.

[24] Woosuk Kwon et al., "Efficient memory management for large language model serving with PagedAttention," in *Proc. SOSP*, 2023.