Speech Processing with LLMs

Matthew Wiesner

Speech processing with LLMs

01 **Language models in ASR**
Decoding, rescoring, and end-to-end recognition

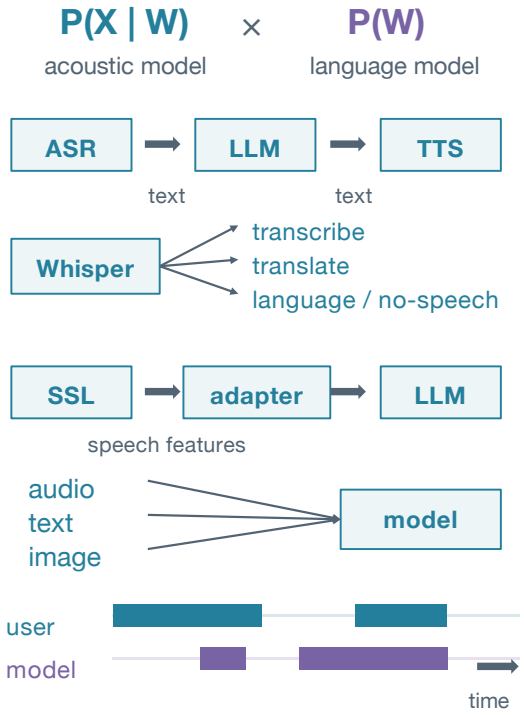
02 **Speech pipelines with LLMs**
Text-mediated understanding and speech generation

03 **Whisper and multitask speech**
Transcription, translation, language ID, no-speech

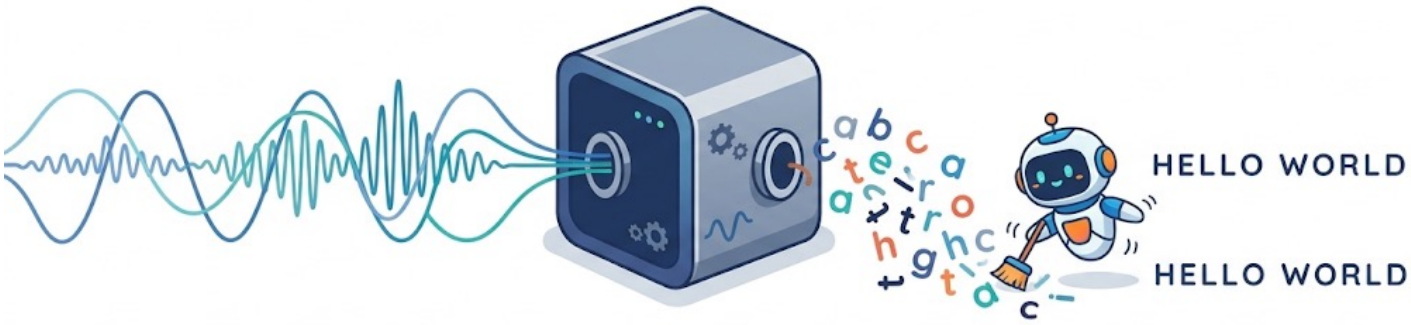
04 **Speech representations as LLM inputs**
SSL encoders, adapters, and audio tokens

05 **Integrated multimodal models**
Gemma, Qwen-Omni, and native audio interfaces

06 **Full-duplex spoken interaction**
Streaming, interruptions, and backchannels

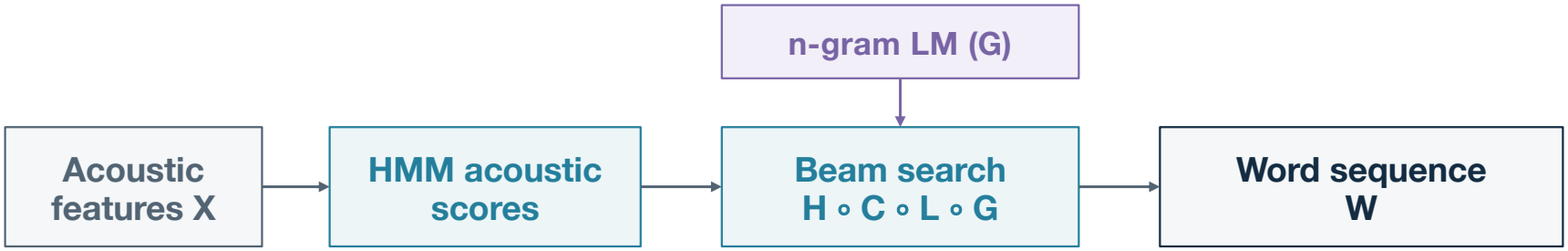


01 Language Models in ASR



HMM decoding with an n-gram LM

Combine acoustic evidence with a prior over word sequences during search.



H: HMM topology C: phone context L: lexicon G: language model

$$\text{score}(W) = \log P(X \mid W) + \lambda \log P(W) + \beta |W|$$

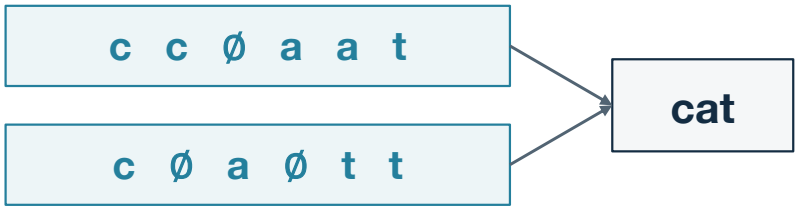
Trigram LM: $P(W) \approx \prod_i P(w_i \mid w_{i-2}, w_{i-1})$

Keep the best-scoring partial paths. LM scores affect which paths survive.

CTC decoding with n-gram shallow fusion

Transcript probabilities from CTC.
External LM helps select among CTC transcripts.

Many alignments, one transcript



Merge repeats, then remove blanks (∅).

$$P(y \mid X) = \sum_{\pi : B(\pi) = y} \prod_{t} p(\pi_t \mid X)$$

Prefix beam search

Sum alignments that share a prefix.
Score LM extensions at token or word boundaries.
Keep blank and nonblank states separate.

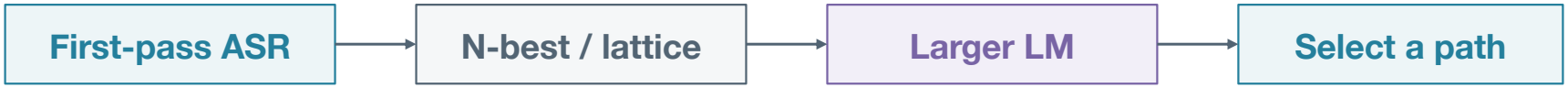
$$\text{score}(y) = \log P(y \mid X) + \lambda \log P(\text{words}(y)) + \beta \, |\text{words}(y)|$$

End-to-end training does not remove the option of an external language model.

Graves & Jaitly (2014) · Hannun et al. (2014) · Hannun (2017)

Second-pass LM rescoring

Decode first, then spend a larger language model on the surviving hypotheses.



$$S(W) = A(W) + \lambda \log P(W) + \beta |W|$$

Illustrative N-best scores: $\lambda = 0.5$, $\beta = 0$. Higher is better.

Candidate transcript	ASR A(W)	LM log P(W)	Combined
wreck a nice beach	−3.0	−7.0	−6.5
recognize speech	**−4.0**	**−2.0**	**−5.0**
recognize peach	−4.5	−6.0	−7.5

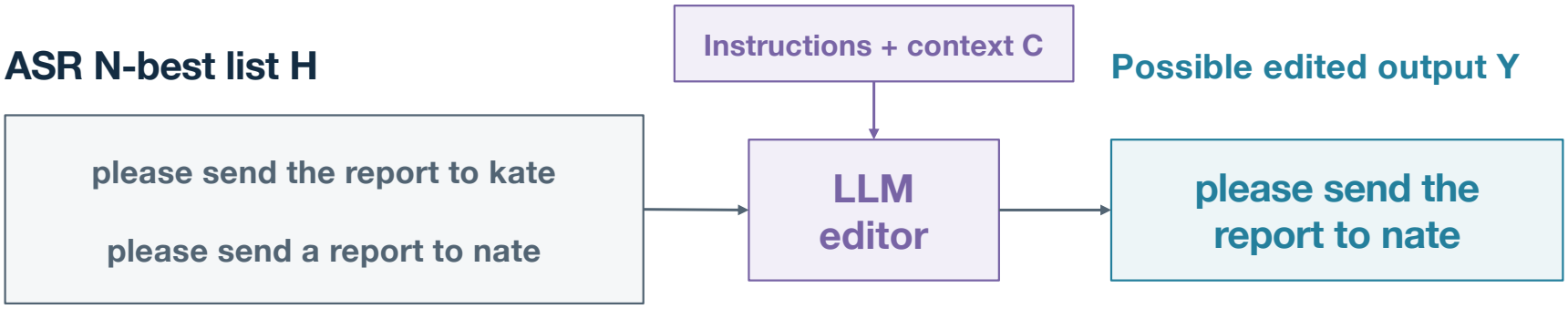
If the first-pass score already includes an LM, remove or retune that contribution.

Fixed N-best rescoring cannot beat its oracle WER: it only selects from that list.

Kaldi decoder documentation · Chen et al., Large-scale LM Rescoring (2023)

LLMs as transcript editors

Generative error correction models the clean transcript given ASR hypotheses.



Illustrative target absent from the input list

$$Y^* = \arg\max_{Y} P(Y \mid H, C)$$

The editor can combine fragments, delete errors, or generate new tokens.

It can escape the N-best oracle bound, but plausible rewrites can introduce errors.

Ma et al. (2023) · Chen et al., HyPoradise (NeurIPS 2023)

Training and using an LLM editor

Prompt a pretrained model, or teach it the error patterns made by ASR.

Prompt a frozen LLM

Provide 1-best or N-best text and relevant context.
Ask for minimal corrections that preserve the words spoken.

Fine-tune an editor

Train on (ASR hypotheses, reference transcript) pairs.
Use supervised fine-tuning, including LoRA.

Synthetic supervision: the Denoising LM recipe



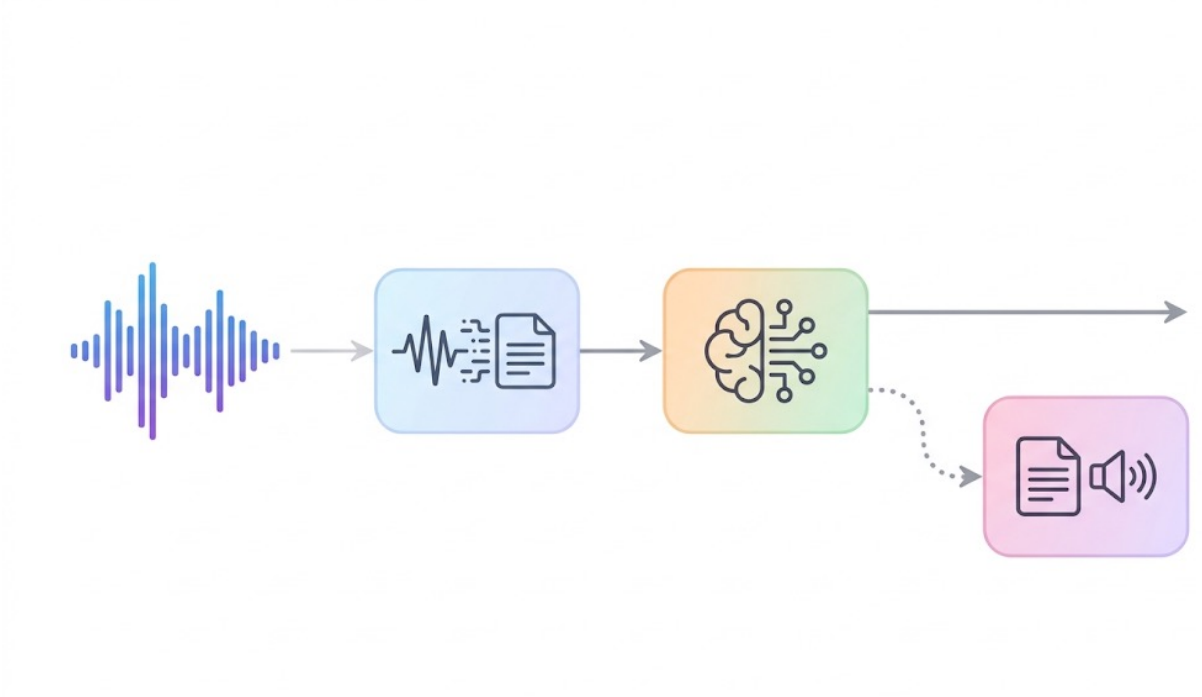
Pair the hypotheses with the original clean text to train the correction model.

Constrain or check proposed edits against ASR evidence. Measure errors introduced, too.

HyPoradise (2023) · Gu et al., Denoising LM (2024) · Ma et al. (2024/2025)

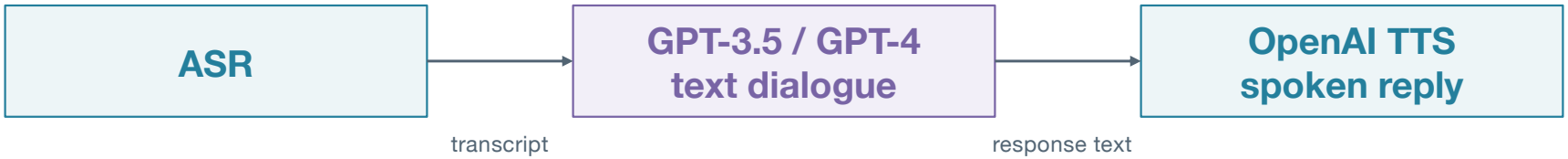
02

Speech Pipelines with LLMs



ChatGPT Voice before GPT-4o (2023–24)

A shipped consumer assistant built from three separately trained models.



Architectural strengths

Reuse an existing conversational LLM.
Give users a hands-free speech interface.
Develop recognition and voice generation separately.

Documented limitations

The text LLM could not directly hear tone, multiple speakers, or background sounds.
The handoffs also added response delay.

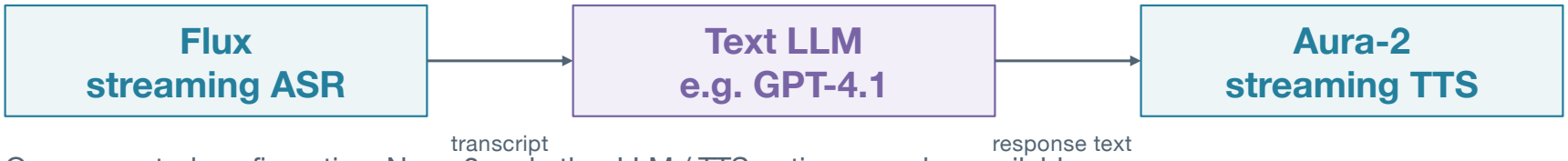
Reported average latency: GPT-3.5 2.8 s GPT-4 5.4 s

A text transcript carries the words, but leaves out much of how they were spoken.

OpenAI: ChatGPT voice launch (Sep 2023) and Hello GPT-4o (May 2024)

Deepgram Voice Agent: a streaming cascade

Production API with separate Listen, Think, and Speak stages.



One supported configuration. Nova-3 and other LLM / TTS options are also available.

Strengths

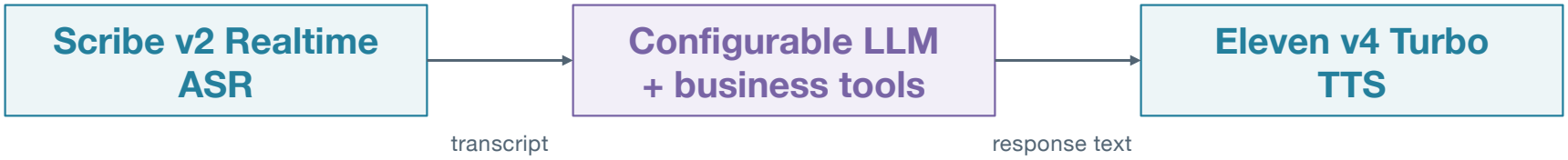
Model-integrated turn detection.
Prepare an LLM reply before the turn fully ends.
Managed orchestration with configurable model choices.

Documented tradeoffs

Earlier turn guesses can create false starts.
If speech resumes, cancel the draft reply.
Speculation adds LLM calls, cost, and state handling.

Streaming reduces delay, while turn detection decides when a reply is safe to play.

ElevenAgents: voice support at Revolut (Banking)



Current platform example (Nov 2026). Revolut's exact model configuration is undisclosed.

Strengths

Keep existing business logic and tool integrations.
Use multilingual voices and language switching.
Add prosodic cues to turn-taking and expressive delivery.

Documented platform issues

Background voices can corrupt transcripts or cause false interruptions.
Expressiveness varies by voice and language.
Response timing still needs tuning.

Modern cascades can carry prosodic and timing signals alongside the transcript.

ElevenLabs: Revolut case study (Jan 2026), Expressive mode and Conversation flow docs

Pipeline strengths and failure modes

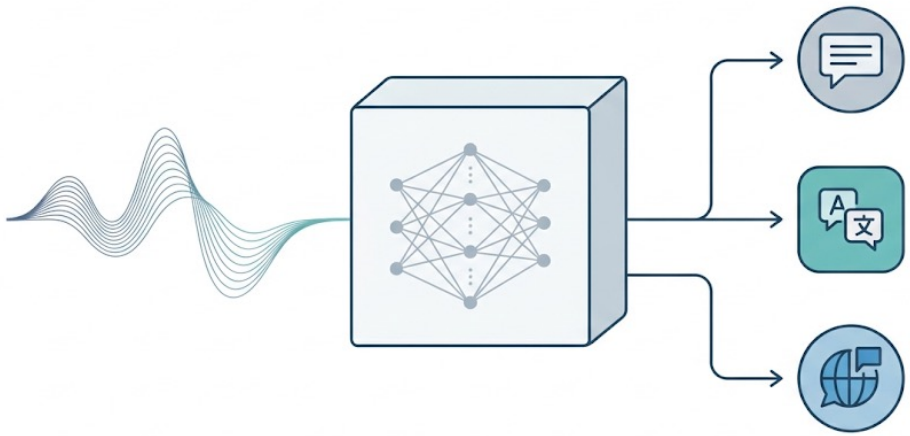
The interfaces between models provide useful controls

Design choice	Production strength	Failure mode / cost
Text between stages	Inspect transcripts and responses. Connect retrieval and business tools.	ASR errors propagate. Plain text omits prosody and other sounds.
Separate models	Choose or upgrade the LLM, recognizer, and voice independently.	Coordinate services, failures, cancellation, and playback state.
Streaming + early drafts	Overlap stages to reduce the delay before a response.	A premature turn boundary can waste work or interrupt.
Dedicated TTS	Control the voice and spoken delivery separately.	Wording and delivery need coordination across models.

ASR mistakes and LLM hallucinations are different errors. A fluent voice can conceal both.

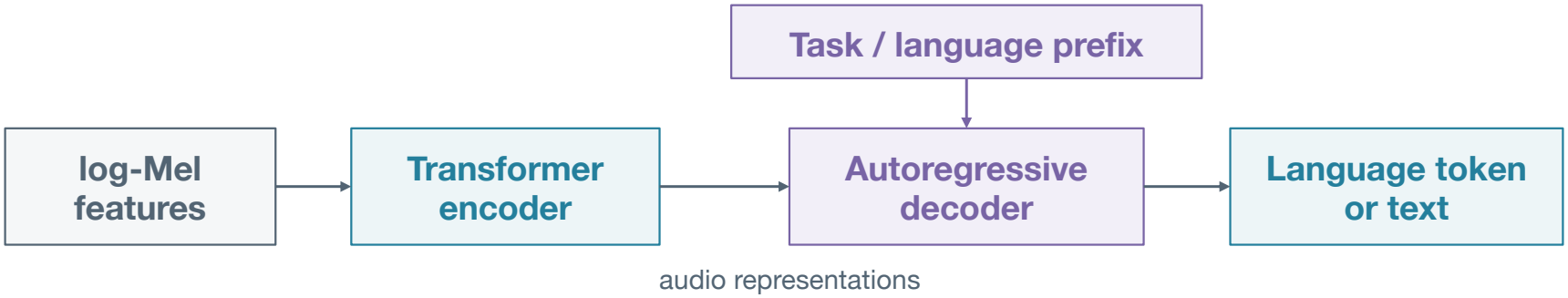
Measure turn-end to first audible reply, including tools, network, and playback.

03 Whisper: A bridge from ASR to Natural Language Audio Interface



Whisper: one model, a fixed set of tasks

Multilingual speech tasks share an encoder–decoder and a token vocabulary.



LID

Which language is spoken?

ASR

Transcribe in that language.

Translation

Translate speech into English.

P(output tokens | audio, decoder prefix)

Task specification becomes part of the sequence the decoder reads.

Radford et al., Robust Speech Recognition via Large-Scale Weak Supervision (2022)

Three tasks, three decoding setups

Same illustrative Spanish audio: “El tren sale a las ocho.”

	Purple: supplied prefix	**Teal: predicted output**
LID	<|startoftranscript|>	→ **<|es|> (Spanish)**
ASR	<|startoftranscript|> <|es|> <|transcribe|> <|notimestamps|>	**El tren sale a las ocho.**
Translation	<|startoftranscript|> <|es|> <|translate|> <|notimestamps|>	**The train leaves at eight.**

LID predicts <|es|>. ASR and translation then use it as a conditioning token.

OpenAI Whisper reference implementation: tokenizer.py and decoding.py

Natural-language instructions for audio

Keep audio-conditioned generation. Broaden how the user describes the task.

Whisper: fixed task vocabulary

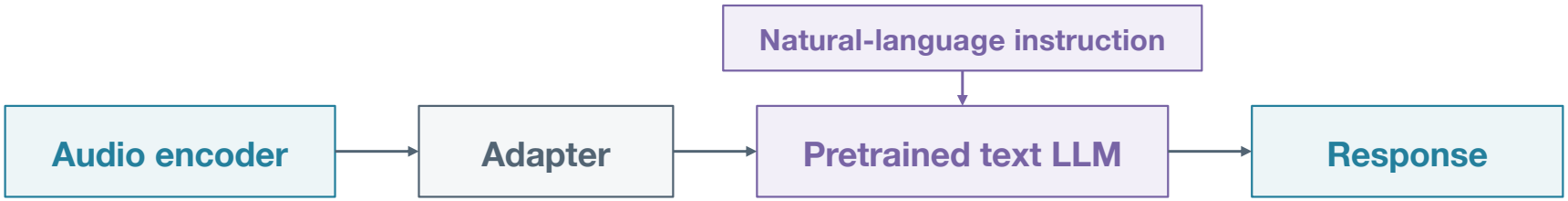
<|es|> <|translate|>

Select a task learned during training.
Free-text prompts provide transcript context.

Audio LLM: task described in words

“Translate this speech into English.”
“What time does the train leave?”

Illustrative instructions for the next model family

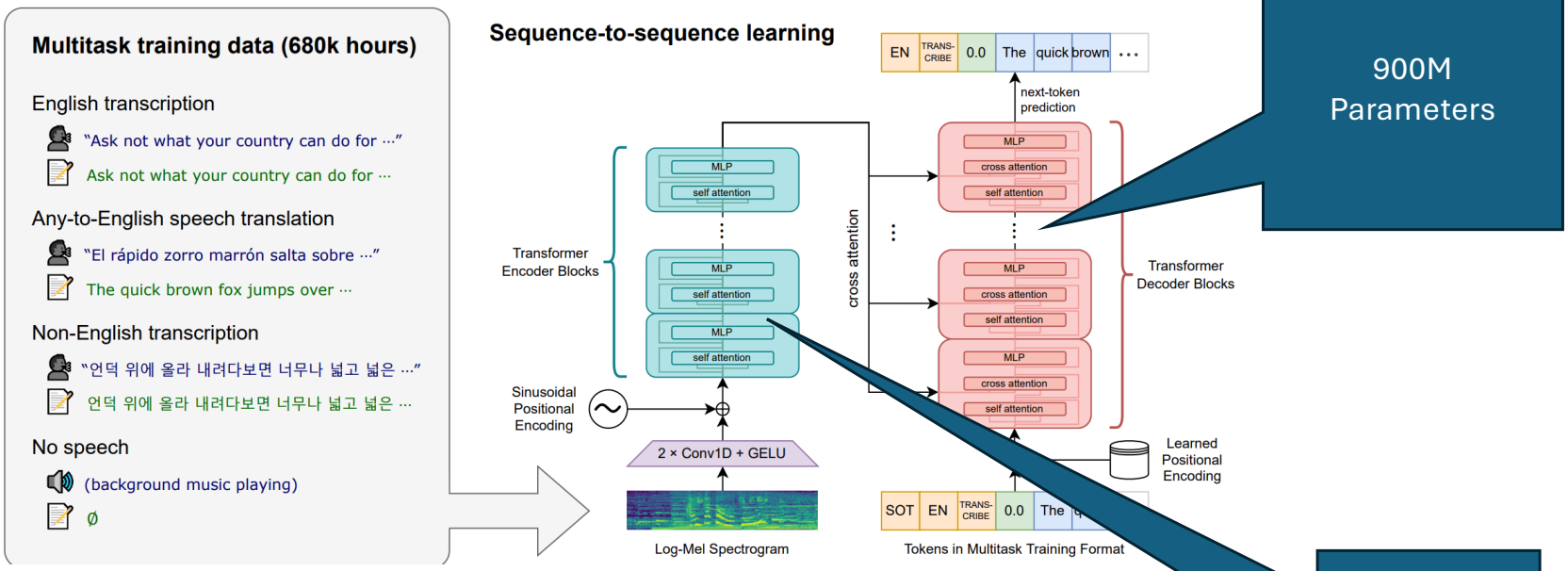


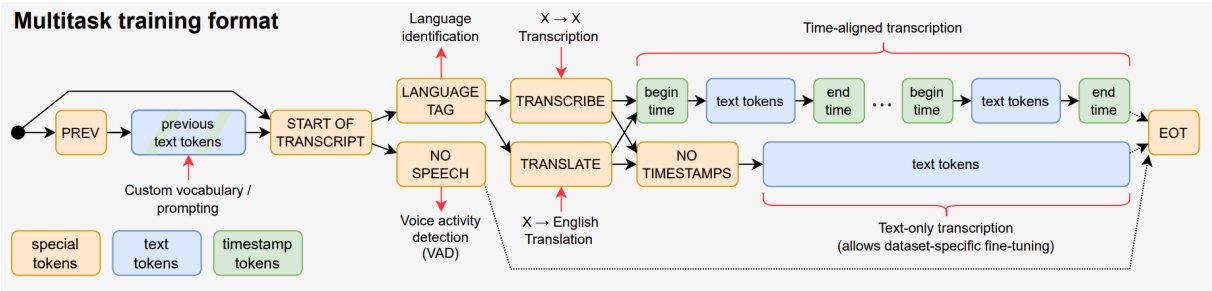
A common audio-LLM architecture

Broader instructions require audio–text alignment and instruction training.

Whisper reference implementation · Tang et al., SALMONN (2023/2024)

Whisper


Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ···"
Ask not what your country can do for ···
Any-to-English speech translation
"El rápido zorro marrón salta sobre ···"
The quick brown fox jumps over ···
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···
No speech
(background music playing)
Ø
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Learned Positional Encoding
SOT
Log-Mel Spectrogram
Tokens in Multitask Training Format
900M Parameters
630M Parameters

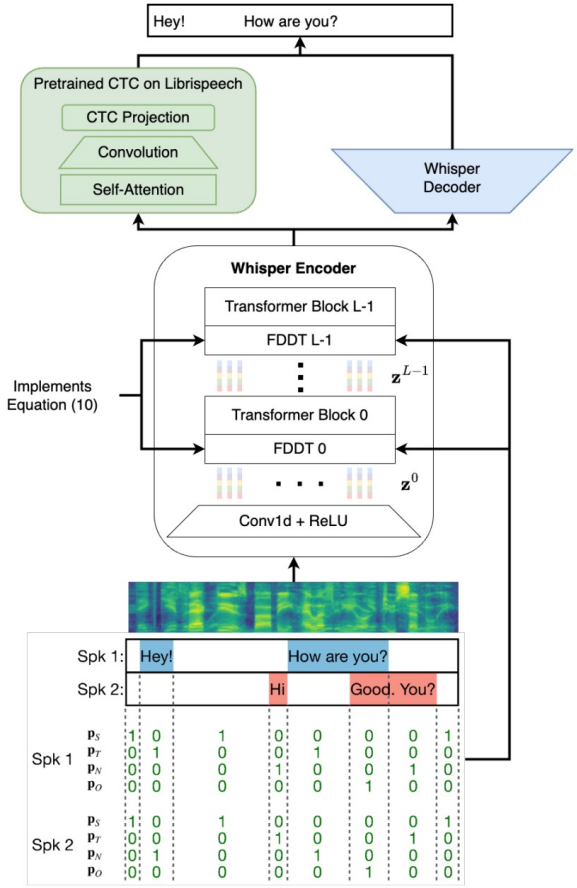

Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
NO TIMESTAMPS
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Whisper

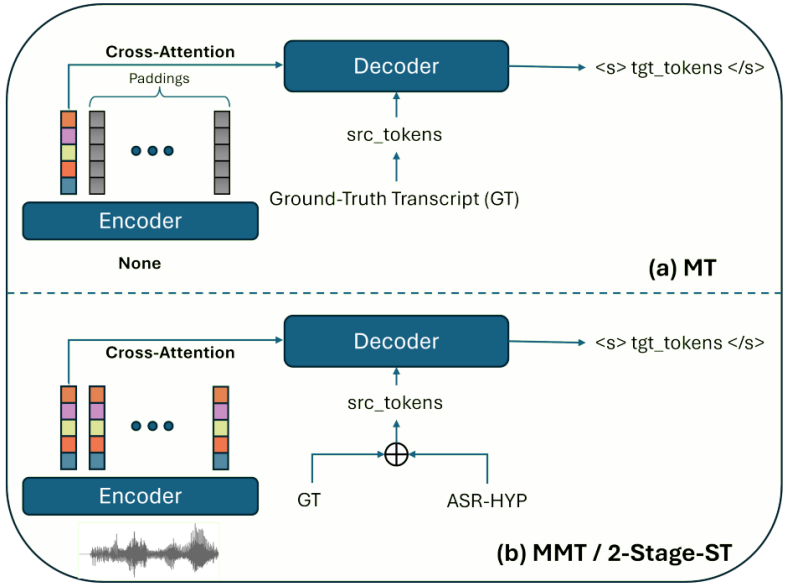
Components are still heavily used.

A really good encoder, and a good decoder

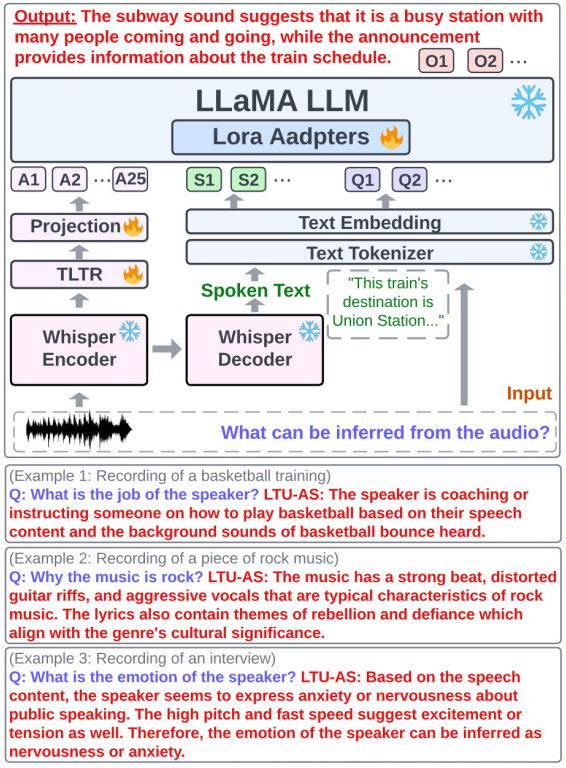
DiCoW: Multi-talker ASR



Whisper-UT: Speech and Text Translation from Speech Models



04 Speech Representations as LLM Inputs



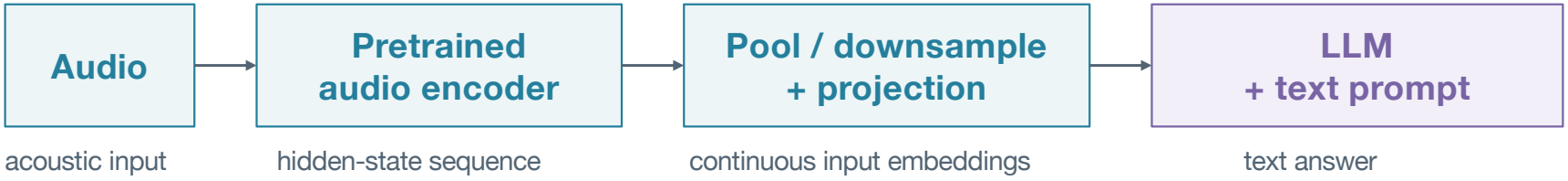
(Example 1: Recording of a basketball training)
Q: What is the job of the speaker? LTU-AS: The speaker is coaching or instructing someone on how to play basketball based on their speech content and the background sounds of basketball bounce heard.

(Example 2: Recording of a piece of rock music)
Q: Why the music is rock? LTU-AS: The music has a strong beat, distorted guitar riffs, and aggressive vocals that are typical characteristics of rock music. The lyrics also contain themes of rebellion and defiance which align with the genre's cultural significance.

(Example 3: Recording of an interview)
Q: What is the emotion of the speaker? LTU-AS: Based on the speech content, the speaker seems to express anxiety or nervousness about public speaking. The high pitch and fast speed suggest excitement or tension as well. Therefore, the emotion of the speaker can be inferred as nervousness or anxiety.

Speech representations as LLM inputs

An audio encoder supplies continuous features that a learned adapter maps into the LLM.



Self-supervised speech encoders

wav2vec 2.0: contrastive masked prediction
HuBERT: predict clustered speech targets
WavLM: masked prediction + denoising

Whisper encoder

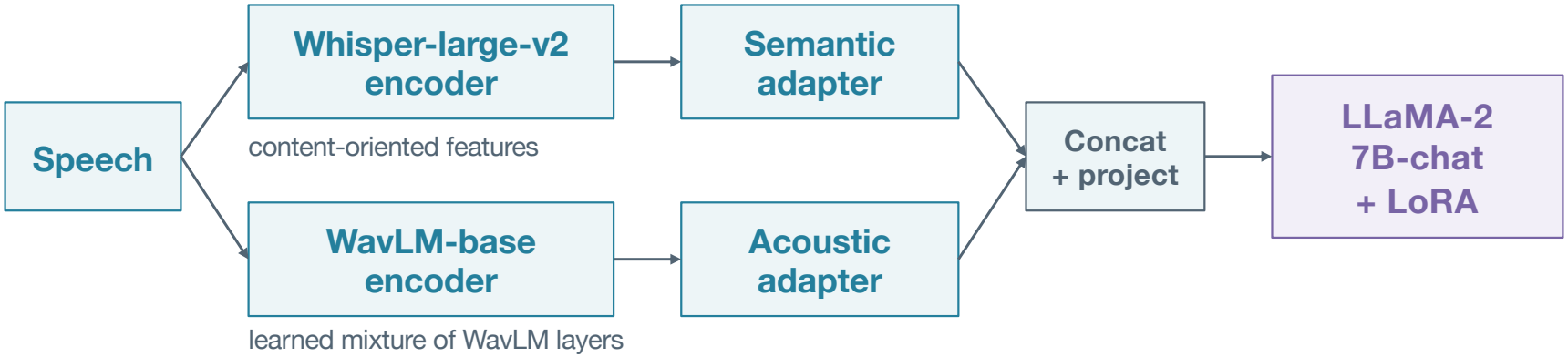
Weakly supervised ASR / translation training
Reuse features without decoding a transcript
Layer choice affects the information retained

The adapter must learn alignment. Audio features are not text-token IDs.

Baevski et al. (2020), Hsu et al. (2021), Chen et al. (2022), Radford et al. (2022)

WavLLM: two complementary speech encoders

Hu et al. (2024) combine Whisper features with self-supervised WavLM features.



Training

Freeze both encoders and base LLM.
Train modality adapters and LoRA.
A prompt adapter scales LoRA by task.

Tasks

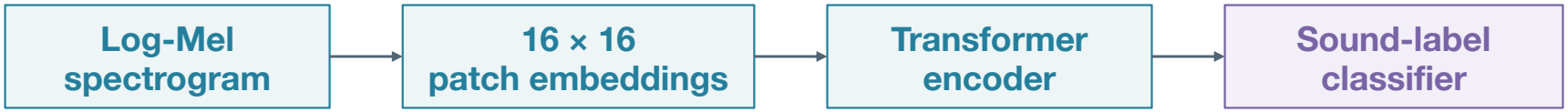
ASR, translation, speaker verification, emotion recognition, and speech QA.
Then train on combinations of tasks.

Complementary training objectives motivate the two branches. Their information overlaps.

AST and LTU: sound features as language-model input

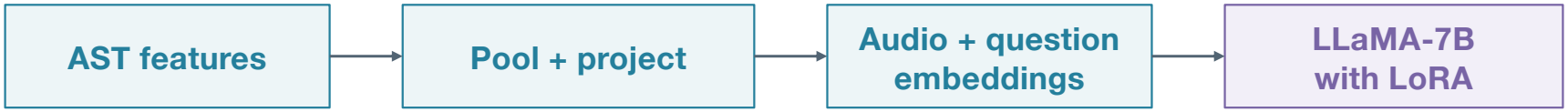
Yuan Gong and collaborators extend sound classification to questions about audio.

AST: Audio Spectrogram Transformer



Original AST transfers a vision Transformer and learns from audio labels.
SSAST is the related self-supervised variant.

LTU: Listen, Think, and Understand



LTU uses CAV-MAE pretraining + AudioSet tuning for its AST encoder.
32 audio embeddings represent a 10-second clip. Training uses audio-question-answer pairs.

A learned connector lets an LLM answer open-ended questions about non-speech sounds.

Gong et al., AST (Interspeech 2021), LTU (2023 / ICLR 2024), SSAST (AAAI 2022)

Whisper features beyond transcription

Whisper-AT and LTU-AS reuse acoustic information that the transcript can omit.

Whisper-AT: supervised audio-event tagging



Intermediate layers retain background-sound information.
Train a lightweight tagging branch while preserving the original ASR model.

LTU-AS: joint audio and speech understanding



The frozen Whisper decoder also supplies the transcript.
The LLM can combine spoken content with sounds and paralinguistic cues.

Whisper-AT learns sound labels. LTU-AS learns answers from audio features and text.

Gong et al., Whisper-AT (Interspeech 2023) and Joint Audio and Speech Understanding (ASRU 2023)

Qwen and Gemma: concrete encoder-based models

A multimodal model can still use a dedicated audio encoder before its language backbone.

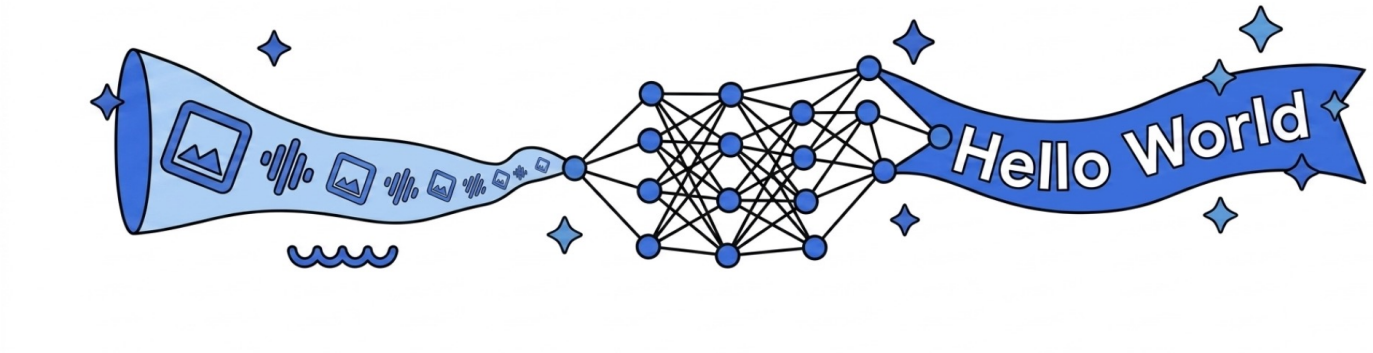
Model	Audio encoder	Representation / design
Qwen2-Audio (2024)	Whisper-large-v3 initialization	Pooled audio features 25 embeddings / second
Qwen2.5-Omni (2025)	Whisper-derived encoder	Blockwise audio processing Thinker consumes features
Qwen3-Omni (2025)	AuT encoder, about 0.6B	New supervised audio model 12.5 embeddings / second
Gemma 3n E2B / E4B	USM-based audio encoder	Designed for device use 6.25 embeddings / second
Gemma 4 E2B / E4B	12-layer Conformer, ~300M	Dedicated audio front end for the small model variants

Whisper initialization describes the starting weights. Subsequent training changes the encoder.

A contrasting direction: Gemma 4 12B uses direct audio projection, without a separate encoder.

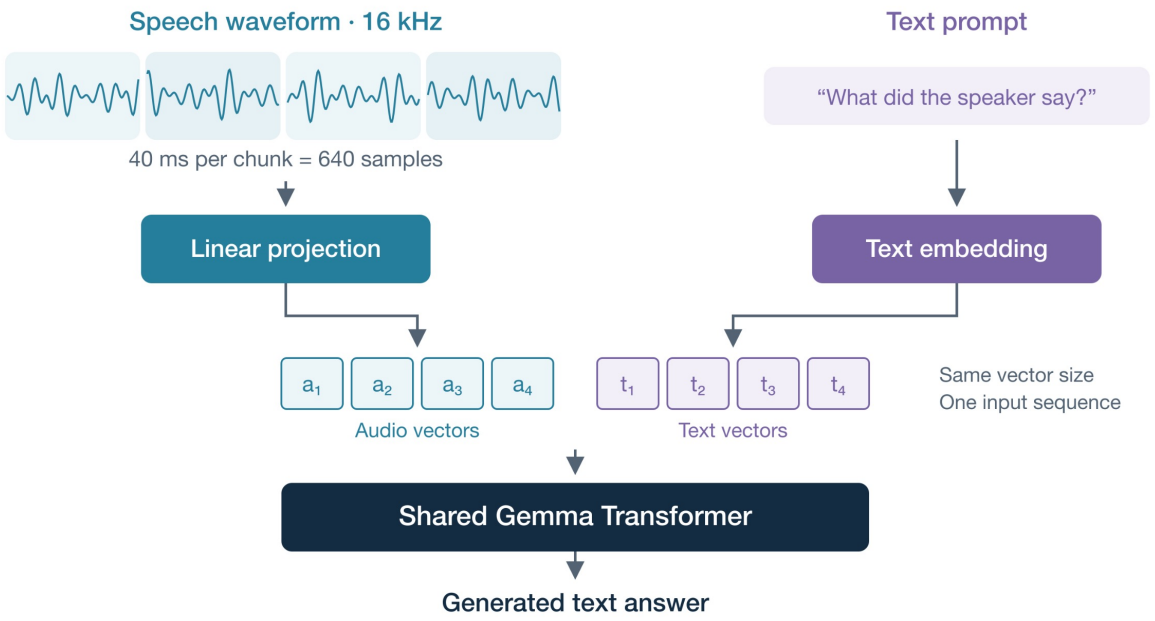
Qwen technical reports (2024–25), Google model documentation (2025–26). Checked Nov 3, 2026.

05 Integrated Multimodal Models



Integrated Models

Speech and text are treated almost identically!



Integrated Models

All of the work is on creating interesting training examples.

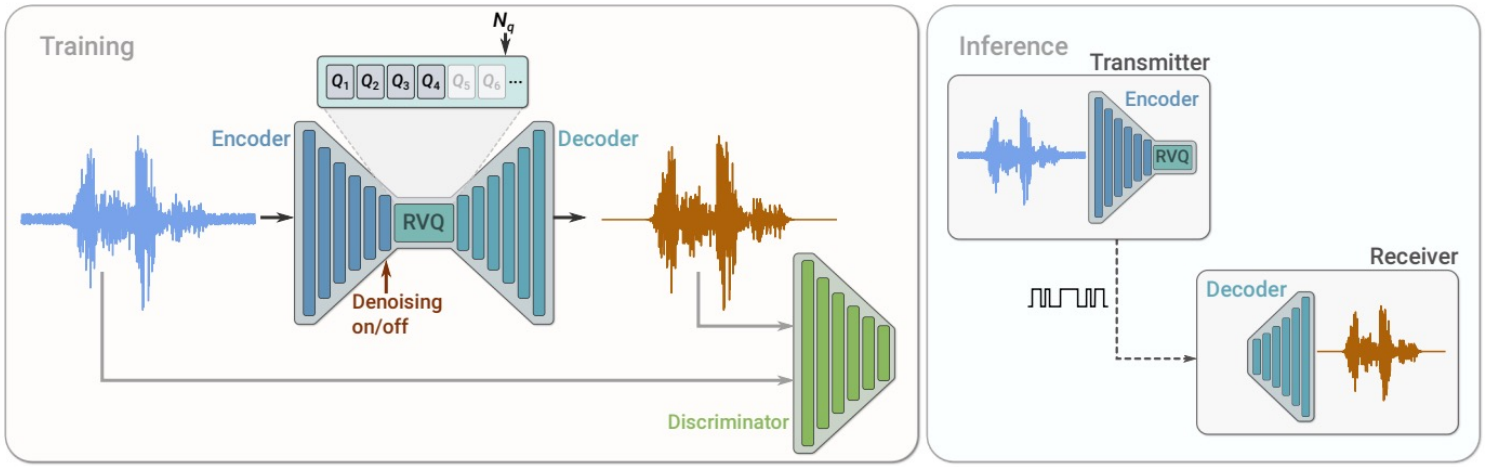


My name is George Washington. I picked up an ax one day.



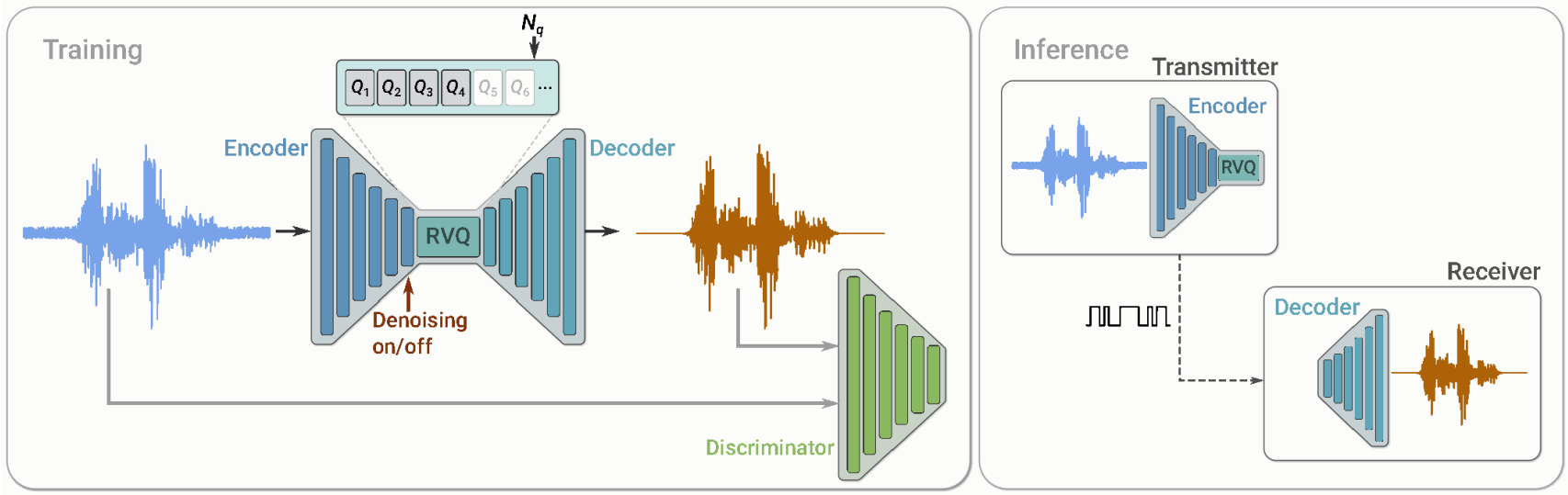
What is the gender of the speaker who picked up the axe?

06a Full Duplex Model (and Audio Codecs First)



SoundStream: a learned audio codec

Joint training teaches the encoder, RVQ and decoder to reconstruct sound.



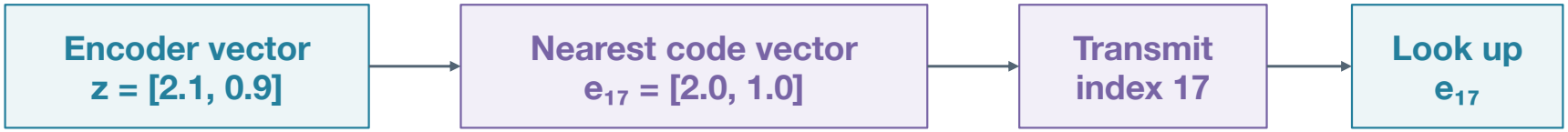
During training, losses compare real and reconstructed audio. At inference, only the codec is needed.

24 kHz waveform → encoder → 75 latent frames/s → RVQ indices → decoder → waveform

Figure: Zeghidour et al., SoundStream (2021), Fig. 2

Vector quantization: replace a vector with an index

Learn a finite dictionary of vectors; choose the nearest entry for each encoder frame.



$$k = \arg\min_j \|z - e_j\|^2 \qquad \hat{z} = e_k$$

The codebook is shared

The encoder and decoder know the same K learned vectors. Only the selected index must be transmitted.

An index has a bit cost

K = 1,024 entries → 10 bits per index. These IDs are learned acoustic symbols; they are not words or phonemes.

Quantization trades exact reconstruction for a finite vocabulary of reusable audio codes.

van den Oord et al., Neural Discrete Representation Learning (2017); SoundStream (2021)

Residual vector quantization, one stage at a time

Each codebook approximates the residual left by the previous codebooks.

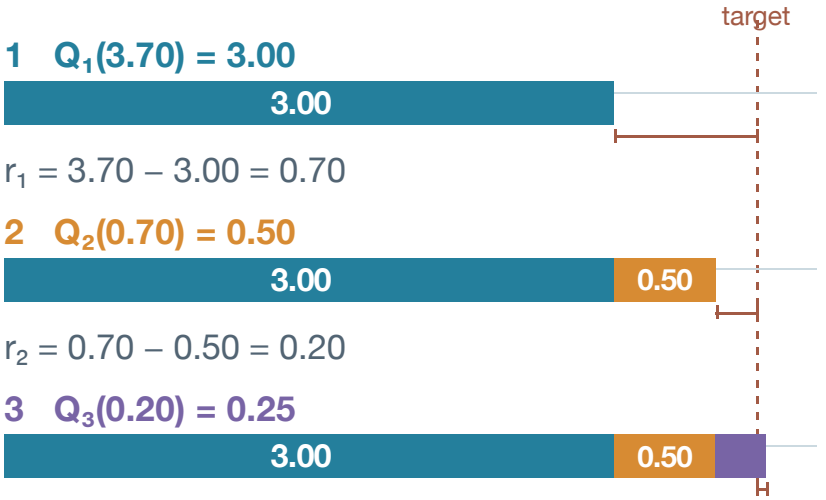
SoundStream's algorithm

Algorithm 1: Residual Vector Quantization

Input: $y = \text{enc}(x)$ the output of the encoder, vector quantizers Q_i for $i = 1..N_q$
Output: the quantized $\hat{y}$
$\hat{y} \leftarrow 0.0$
residual $\leftarrow y$
for $i = 1$ *to* N_q **do**
  $\hat{y}$ += Q_i(residual)
  residual −= Q_i(residual)
return $\hat{y}$

Encode: send one index per codebook.
Decode: retrieve the vectors and add.

Scalar example: z = 3.70



Reconstruction: 3.00 + 0.50 + 0.25 = 3.75 **Final error: 3.70 – 3.75 = –0.05**

Source: SoundStream (2021), Algorithm 1. Colored bars: illustrative scalar example.

SoundStream: train for quality, choose a bitrate

Joint training makes the discrete bottleneck useful; quantizer dropout enables variable rates.

Training the codec

Reconstruction / spectral losses
keep the decoded signal close to the input.

Adversarial + feature-matching losses
encourage perceptually realistic sound.

Straight-through gradients train the encoder;
codebook vectors use moving-average updates.

Quantizer dropout trains with a variable
prefix of RVQ stages: 1, …, Q.

Fixed-width bitrate

$$R = \text{frames/s} \times Q \times \log_2 K$$

RVQ stages Q	Bits/frame	Bitrate
4	40	3 kb/s
8	80	6 kb/s
16	160	12 kb/s

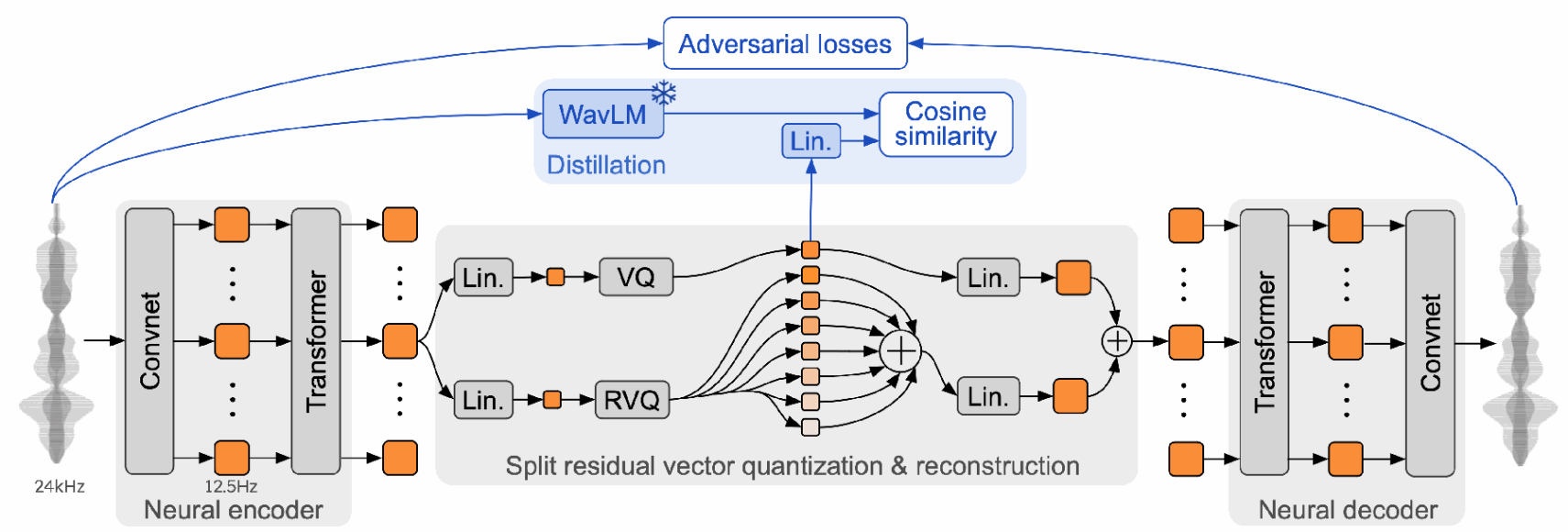
Here: 75 frames/s, K = 1,024 per stage.
Rates exclude headers and entropy coding.

Quantizer dropout lets one model support several bitrates using different RVQ prefixes.

SoundStream (2021); Google Research, SoundStream overview; VQ-VAE (2017)

Mimi: semantic and acoustic codes

The semantic branch learns from WavLM while the codec learns to reconstruct audio.



Moshi configuration: 1 semantic + 7 acoustic codes per 80 ms frame, at 1.1 kb/s

Figure: Défossez et al., Moshi (2024), Fig. 2. Blue paths operate during training.

Mimi turns speech into targets an LM can predict

Each 80 ms frame carries a small bundle of discrete codes, including a semantic code.

Code / frame	t	t + 1	t + 2
Semantic	s_t	s_{t+1}	s_{t+2}
Acoustic 1	$a_{t,1}$	$a_{t+1,1}$	$a_{t+2,1}$
…	…	…	…
Acoustic 7	$a_{t,7}$	$a_{t+1,7}$	$a_{t+2,7}$

Frames ≠ code indices

12.5 frames/s × 8 codes/frame
= 100 code IDs/s per audio stream.

2,048 entries/codebook → 11 bits/ID
100 × 11 = 1.1 kb/s.

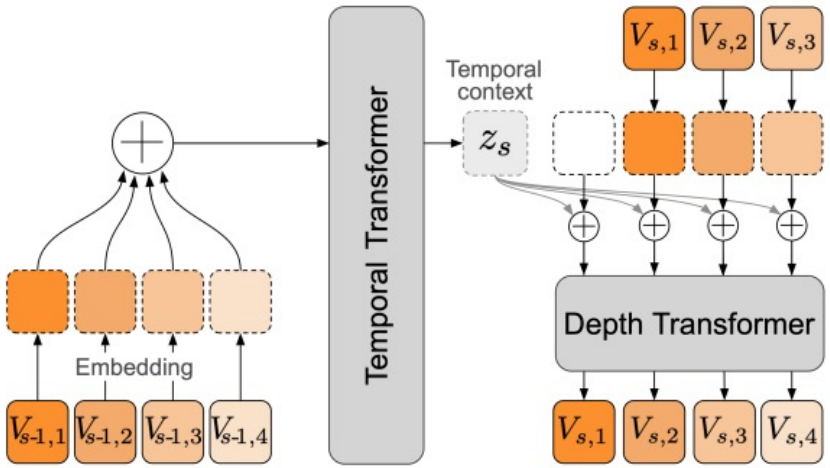


Moshi: a temporal Transformer models frames; a depth Transformer predicts codes within each frame.

Train the codec to represent sound; train the speech LM to predict sequences of those codes.

Moshi (2024), §§3–4; Kyutai Moshi / Mimi documentation; 8-codebook configuration

06b Full Duplex Spoken Interaction (Moshi)

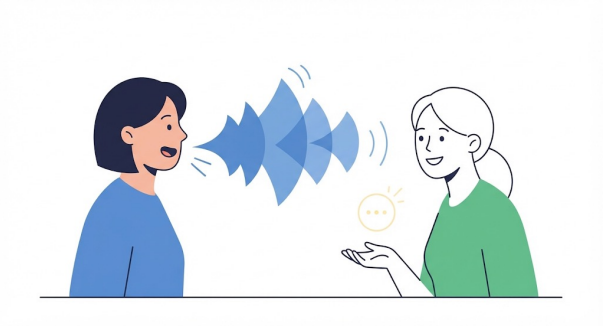


Problems with the ASR - LLM – TTS Pipeline

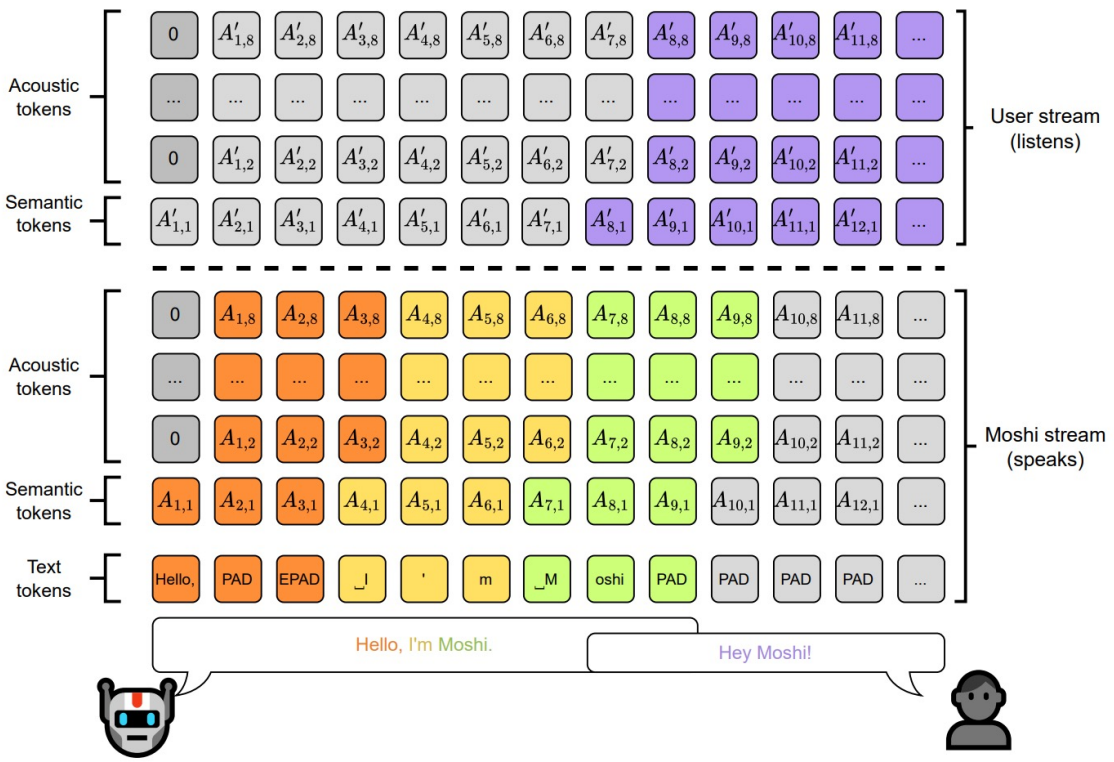
Why should I have to wait for the TTS to synthesize a giant block of text?

How do I know the LLM is listening?

We want a model that actually mimics human conversations!



Full Duplex


Acoustic tokens
Semantic tokens
User stream (listens)
Acoustic tokens
Semantic tokens
Text tokens
Moshi stream (speaks)
Hello,
PAD
EPAD
_I
'
m
_M
oshi
PAD
PAD
PAD
PAD
Hello, I'm Moshi.
Hey Moshi!