10-423/10-623 Generative AI

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Audio Understanding and Synthesis

Matt Gormley & Henry Chai
Lecture 24
Dec. 2, 2024

Reminders

- HW623
  - Only for students registered in 10-623
  - Due: Mon, Dec 2 at 11:59pm
  - Submit form: https://forms.gle/azrmUR9KrFexnASi7
- Project Poster
  - Upload Due: Tue, Dec 10 at 11:59pm
  - Presentations: Fri, Dec 13 at 1pm-4pm
- Project Final Report
  - Due: Fri, Dec 13 at 11:59pm
- Project Code Upload
  - Due: Fri, Dec 13 at 11:59pm

Poster Printing

- if you need to modify your poster to include a small update, consider paper and tape
- or reprint the entire thing if larger changes are needed

Audio Understanding and Synthesis

- Outline
  - working with audio data
    - Mel spectrogram
  - speech transcription (speech-to-text)
    - Whisper
  - speech generation (audio continuation)
    - autoregressive audio codec models (AudioLM)
  - music generation
    - MIDI generation (MusicTransformer)
    - audio tokenization (EnCodec)
    - Text-to-music generation (e.g. MusicGen)
  - combining speech and text models
    - speech+text-to-text (SpeechVerse)

AUDIO UNDERSTANDING & SYNTHESIS

Working with Audio Data

- Pulse-code modulation (PCM) representation
  - an audio recording device takes many samples of pressure (amplitude)
  - a raw audio file has several parameters:
    - # of channels (e.g. stereo has two)
    - bit depth (# of bits used to represent each amplitude)
    - sampling rate (how many samples are taken per second)
  - **example:** a 44.1 kHz 16-bit stereo recording has 44,100 samples per second, each sample consists of two 16-bit integers, one for each channel

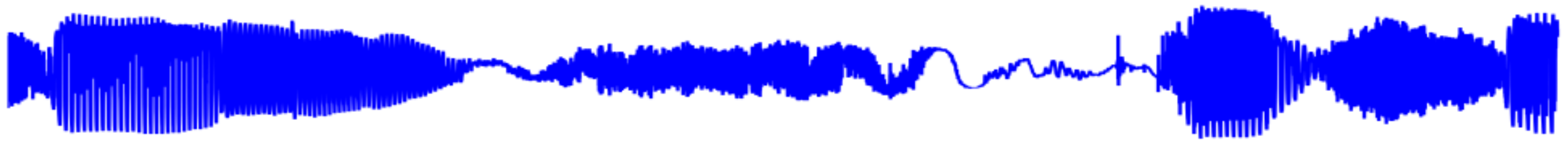
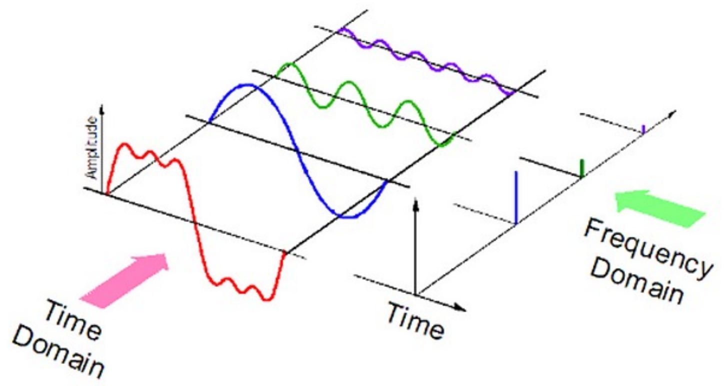


Figure 1: A second of generated speech.

Figure from https://arxiv.org/abs/1609.03499

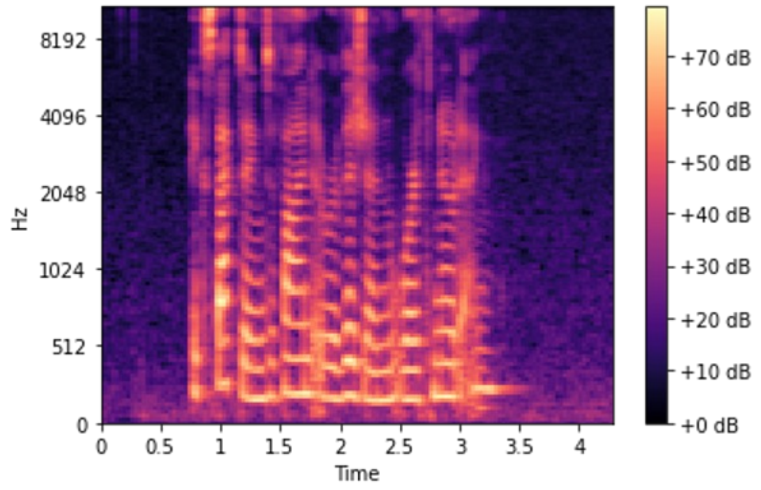
Working with Audio Data

Figure from https://kinder-chen.medium.com/denoising-data-with-fast-fourier-transform-a81d9f38cc4c



- Sound wave representation
  - if we were recording audio of a **single** unchanging sound, we could run a **single** Fast Fourier Transform (FFT) to extract the sinusoidal waves that gave rise to the samples we observe
- Mel-spetrogram representation
  - in practice sound changes over time and so we need to run **many FFTs of overlapping windows** to extract a usable representation of real audio
  - the output of this process is a **spectrogram,** and can be easily visualized as an image
  - to obtain a **mel-spectrogram,** we pass the frequencies through a frequency-to-mel map

Figure from https://towardsdatascience.com/audio-deep-learning-made-simple-part-2-why-mel-spectrograms-perform-better-aad889a93505



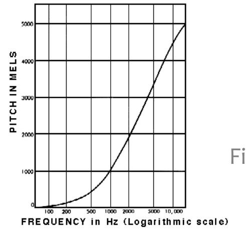


Figure from https://www.sfu.ca/sonic-studio-webdav/handbook/Mel.html

Speech Transcription: Whisper

- **Speech transcription** is the task of taking in the audio of speech and generating the text transcript
- **Whisper** is an encoder-decoder Transformer model for speech transcription
- The **input** is the log-mel-spectrogram of the audio (30 second chunk) [16kHz, 80 channels, 25 ms window, 10 ms stride]

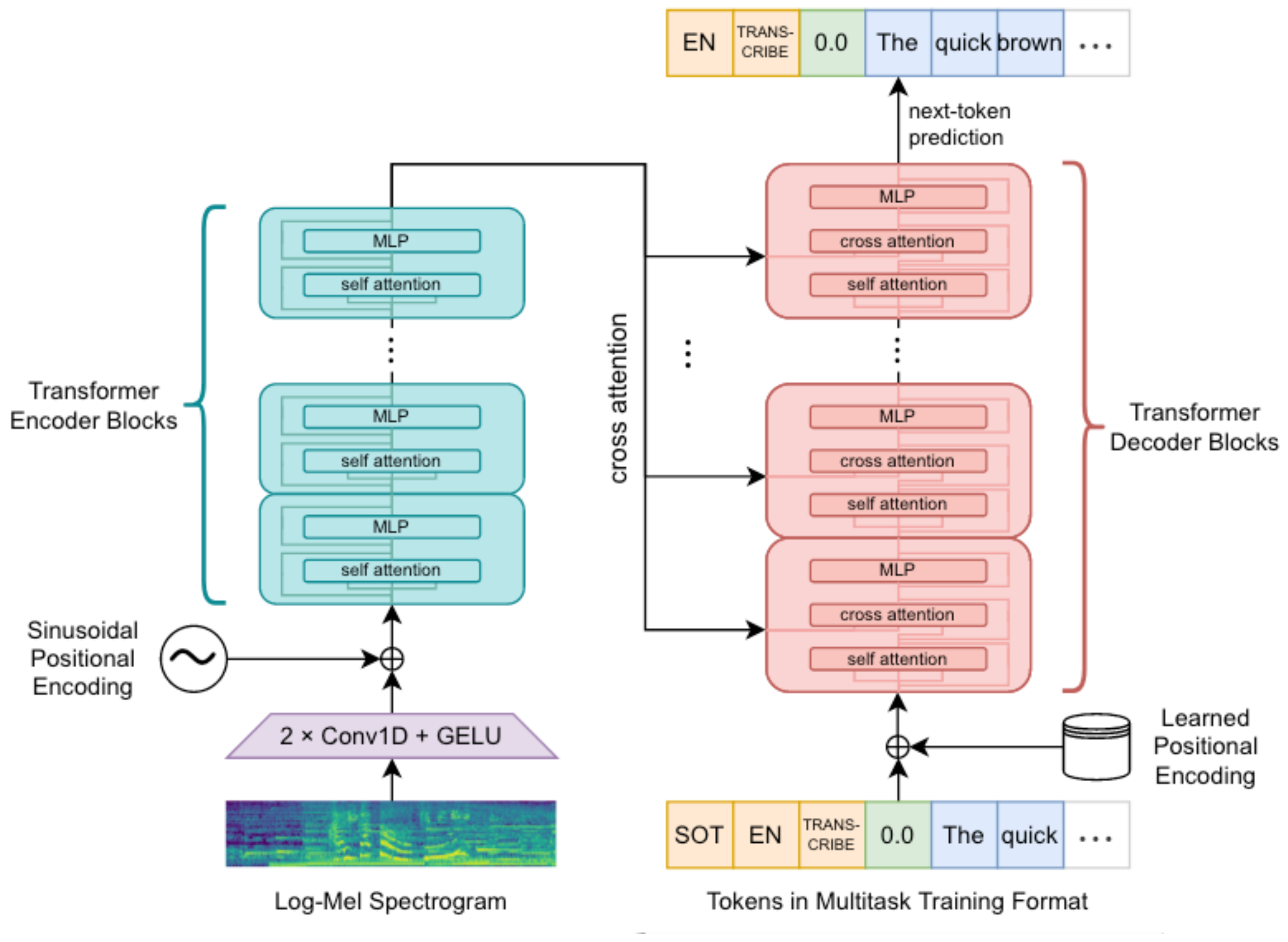


Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Whisper

Whisper

- model is *almost* identical to Vaswani et al. (2017)
- encoder: output of two convolution layers (filter={3,3}, stride={1,2}) + sinusoidal positional embeddings
- decoder: learned positional embeddings
- same # of transformer blocks in encoder / decoder (32)

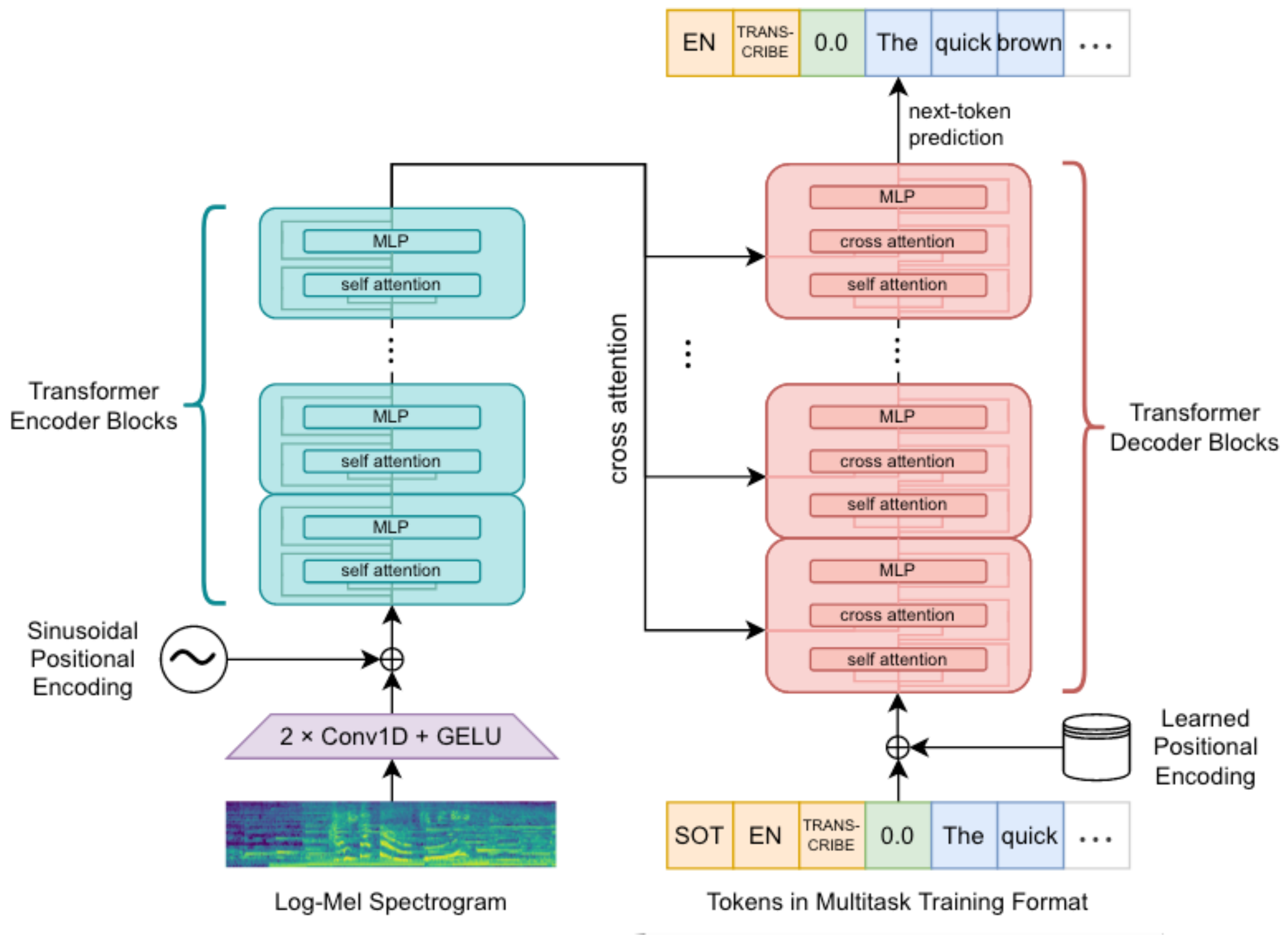
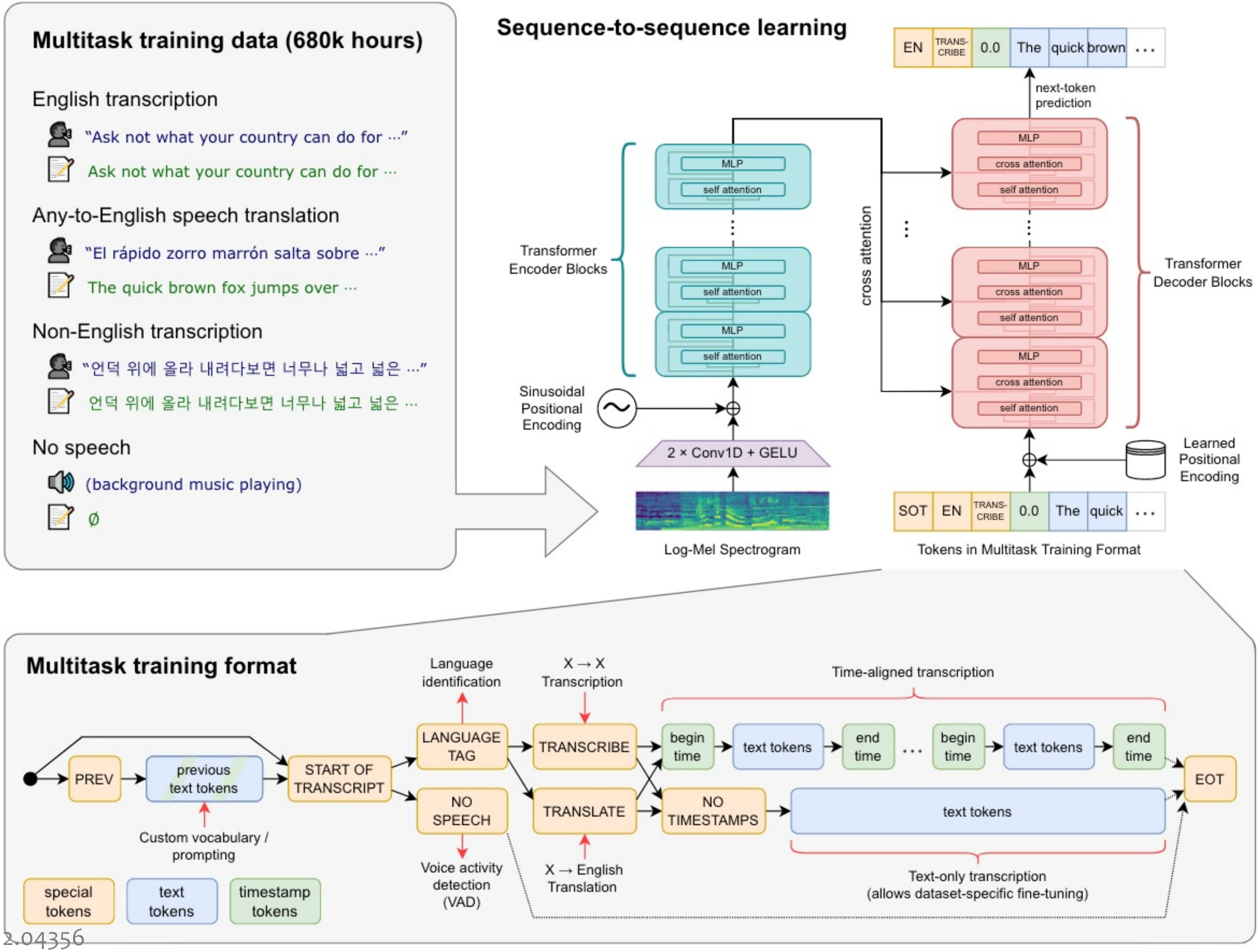


Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Whisper

- many tasks are used to train a single model
- special tokens are used to indicate the type of task, the language, etc.
- timestamp tokens allow the generation of time-aligned transcripts

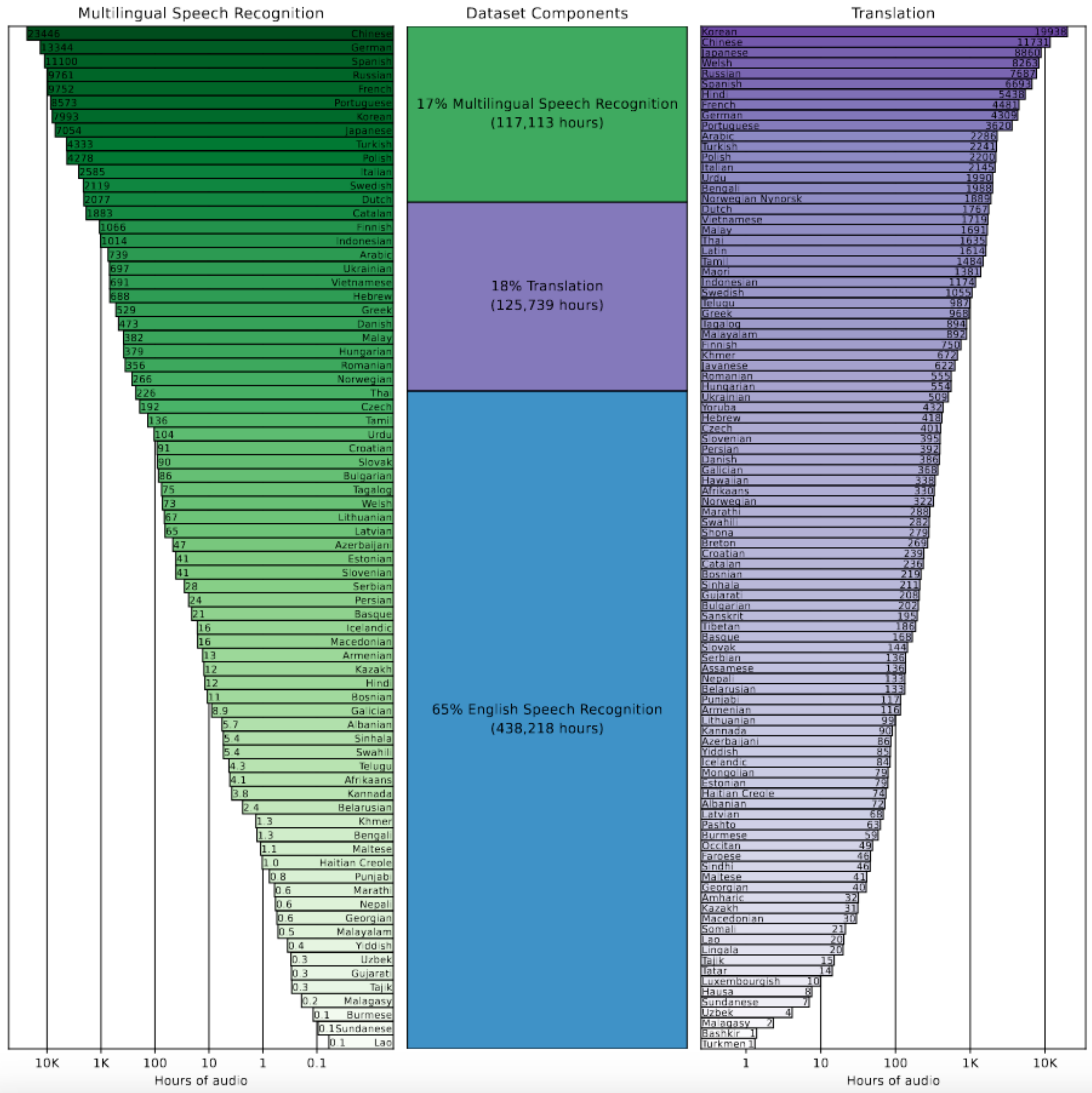


Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ···"
Ask not what your country can do for ···
Any-to-English speech translation
"El rápido zorro marrón salta sobre ···"
The quick brown fox jumps over ···
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고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

Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Whisper

- Large model has 1.5B parameters
- Training dataset is so large that only 2-3 epochs are used



Speech Transcription: Whisper

Results:

- Whisper closes the gap to human level performance on LibriSpeech English WER

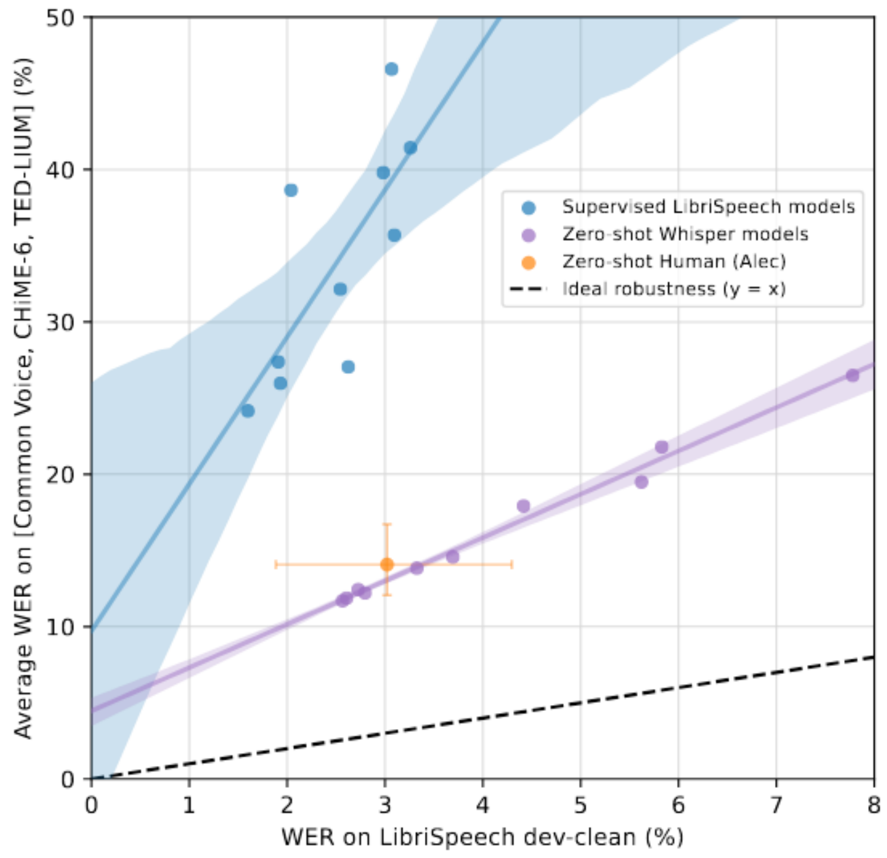


Figure 2. **Zero-shot Whisper models close the gap to human robustness.** Despite matching or outperforming a human on LibriSpeech dev-clean, supervised LibriSpeech models make roughly twice as many errors as a human on other datasets demonstrating their brittleness and lack of robustness. The estimated robustness frontier of zero-shot Whisper models, however, includes the 95% confidence interval for this particular human.

Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Whisper

Results:

- Whisper closes the gap to human level performance on LibriSpeech English WER

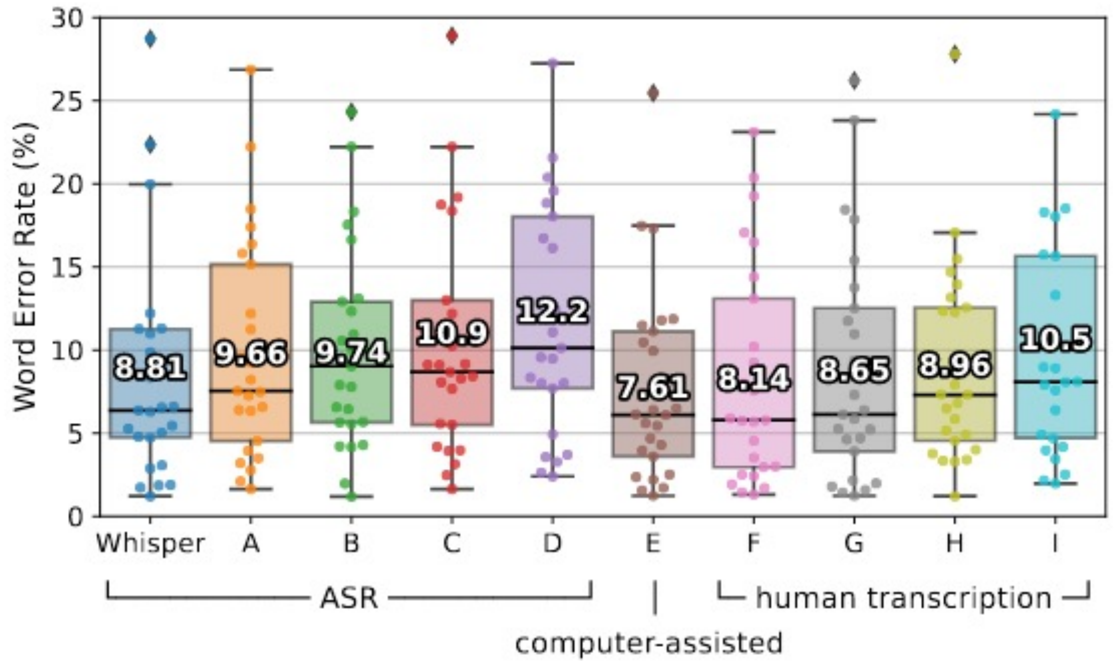


Figure 7. **Whisper's performance is close to that of professional human transcribers.** This plot shows the WER distributions of 25 recordings from the Kincaid46 dataset transcribed by Whisper, the same 4 commercial ASR systems from Figure 6 (A-D), one computer-assisted human transcription service (E) and 4 human transcription services (F-I). The box plot is superimposed with dots indicating the WERs on individual recordings, and the aggregate WER over the 25 recordings are annotated on each box.

Figure from https://arxiv.org/abs/2212.04356

Whisper Demo

- **Transcript:** Whisper is an automatic speech recognition (ASR) system trained on 680,000 hours of multilingual and multitask supervised data collected from the web. We show that the use of such a large and diverse dataset leads to improved robustness to accents, background noise and technical language. Moreover, it enables transcription in multiple languages, as well as translation from those languages into English. We are open-sourcing models and inference code to serve as a foundation for building useful applications and for further research on robust speech processing.

https://www.youtube.com/watch?v=zLP6oT3uqV8

Speech Transcription: Whisper

Results:

- strong performance on high resource languages
- not-so-strong performance on low resource languages

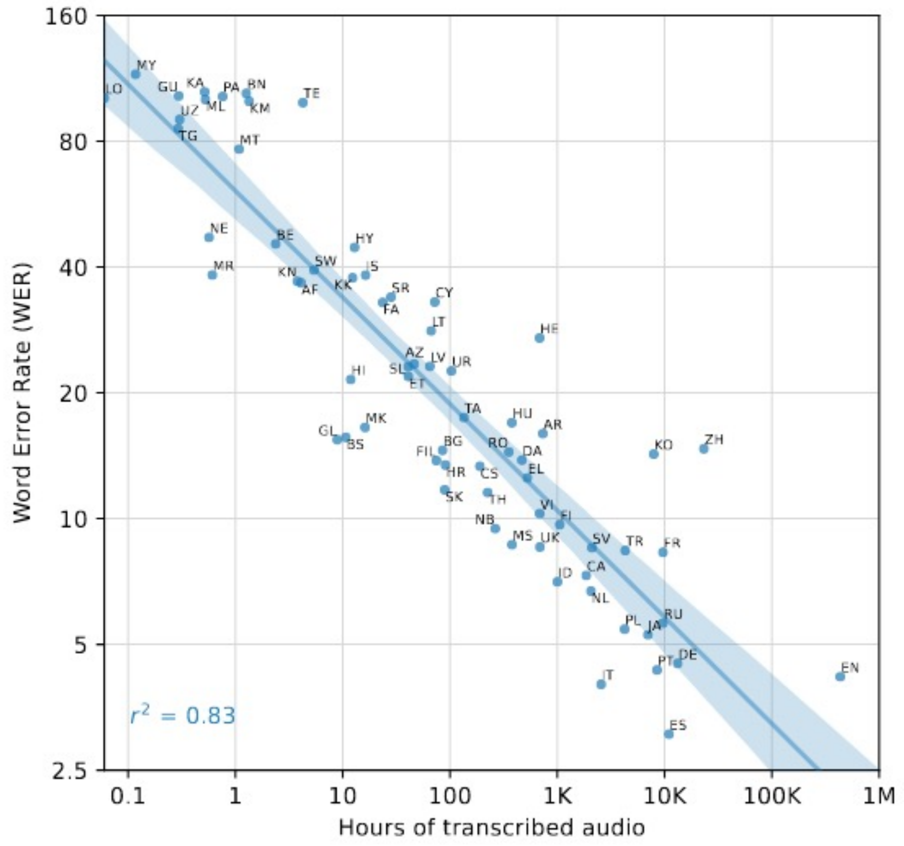


Figure 3. **Correlation of pre-training supervision amount with downstream speech recognition performance.** The amount of pre-training speech recognition data for a given language is very predictive of zero-shot performance on that language in Fleurs.

Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Whisper

Results:

- capable of good speech translation performance on a wide variety of languages

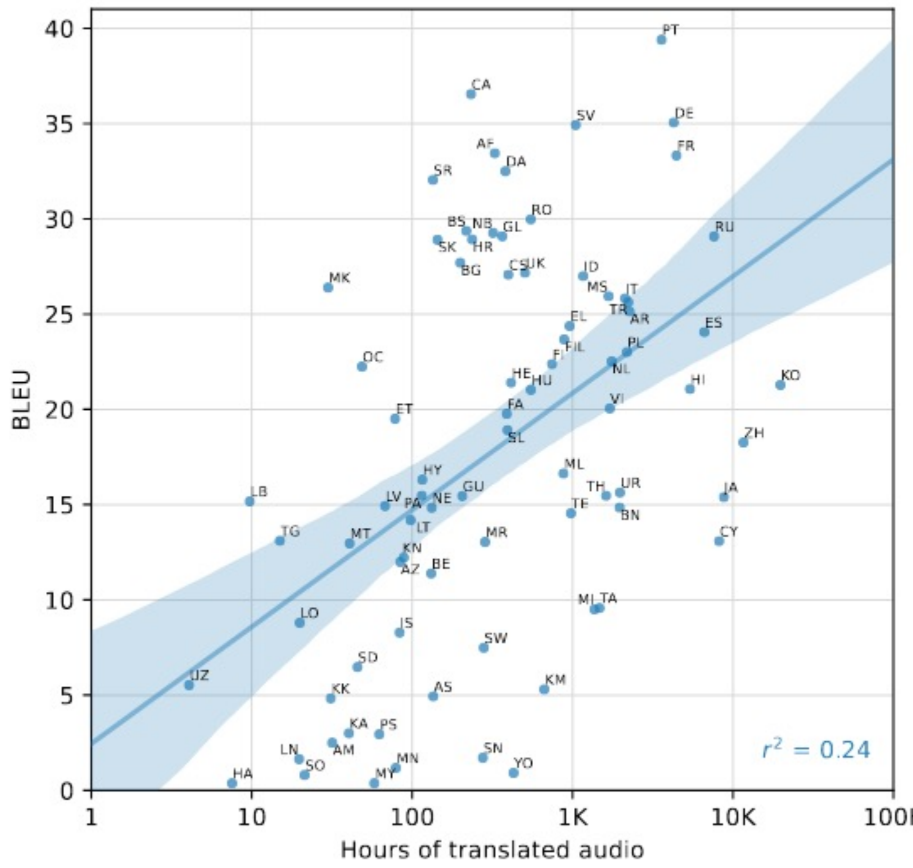


Figure 4. **Correlation of pre-training supervision amount with downstream translation performance.** The amount of pre-training translation data for a given language is only moderately predictive of Whisper's zero-shot performance on that language in Fleurs.

Figure from https://arxiv.org/abs/2212.04356

Speech Transcription: Gemini

- Multimodal large language models (e.g. Gemini) are trained with many different modalities as input
- Gemini converts each audio input to 16kHz, then each second of audio is converted to 25 tokens
- Speech transcription follows naturally from this setup and allows interleaving of a text prompt with audio tokens

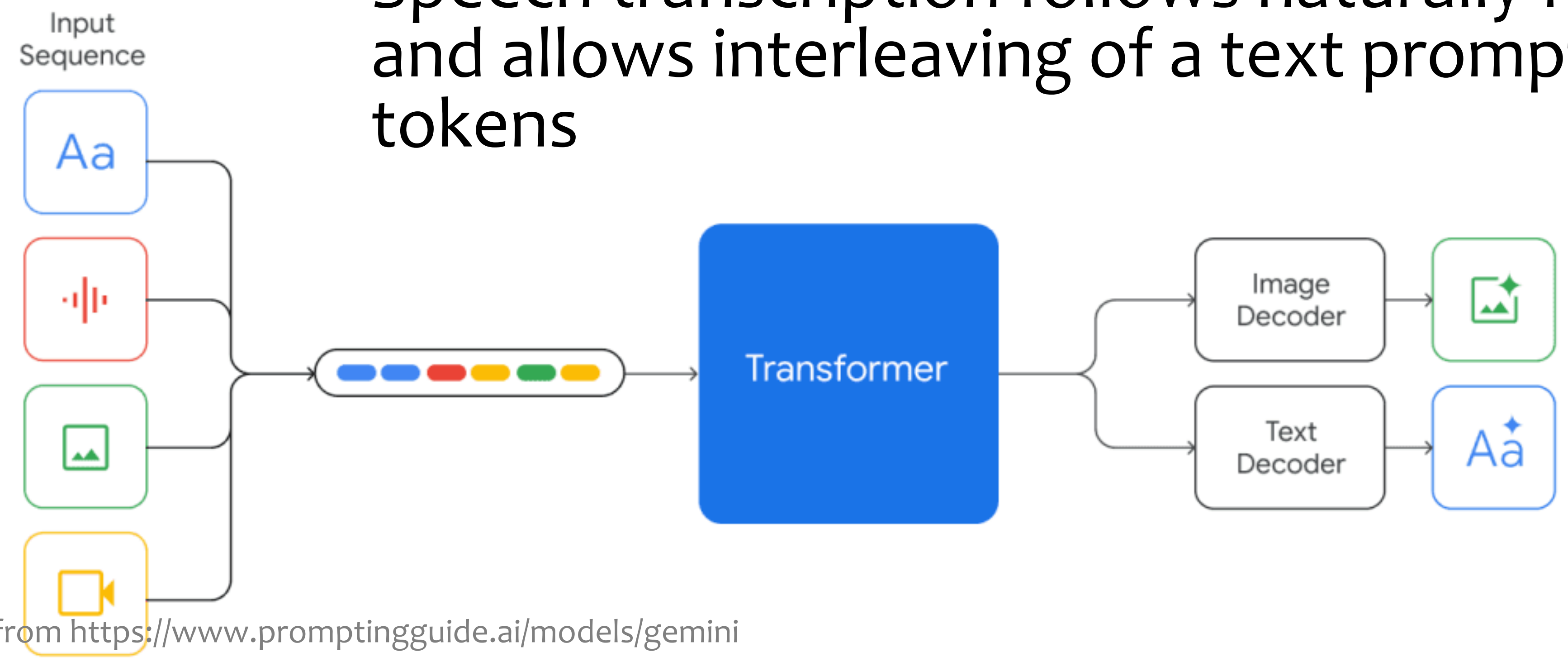


Figure from https://www.promptingguide.ai/models/gemini

Audio Continuation: AudioLM

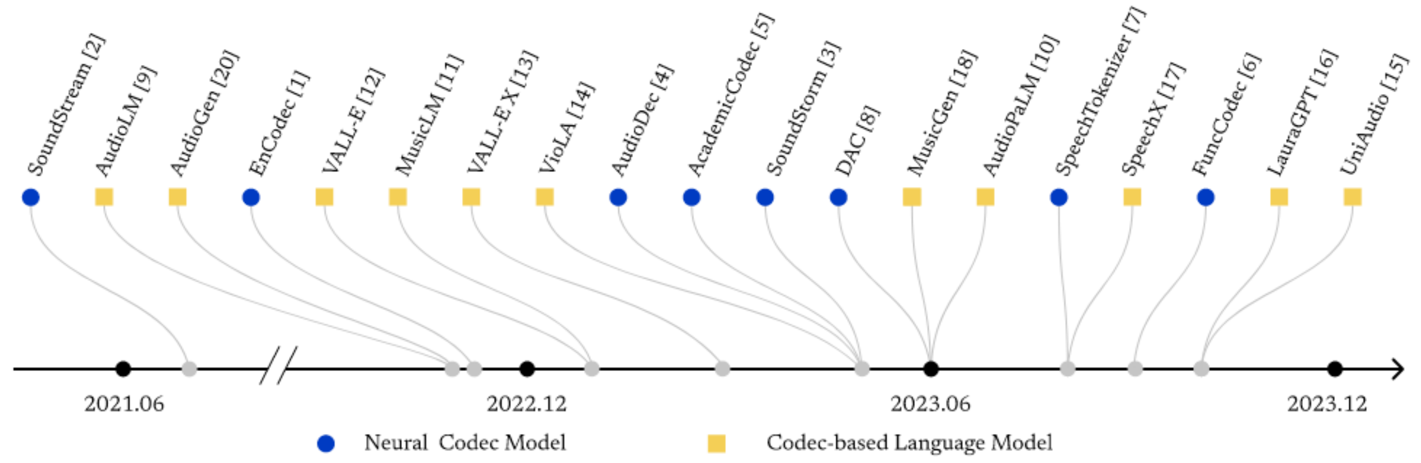


Fig. 1. Timeline of current neural codec models and codec-based language models.

- a variety of models use an audio code to convert the audio signal to a sequence of discrete audio tokens
- we can then train a deep neural LM on these audio tokens and use it to generate speech, music, nature sounds, etc.

Audio Continuation: AudioLM

AudioLM consists of

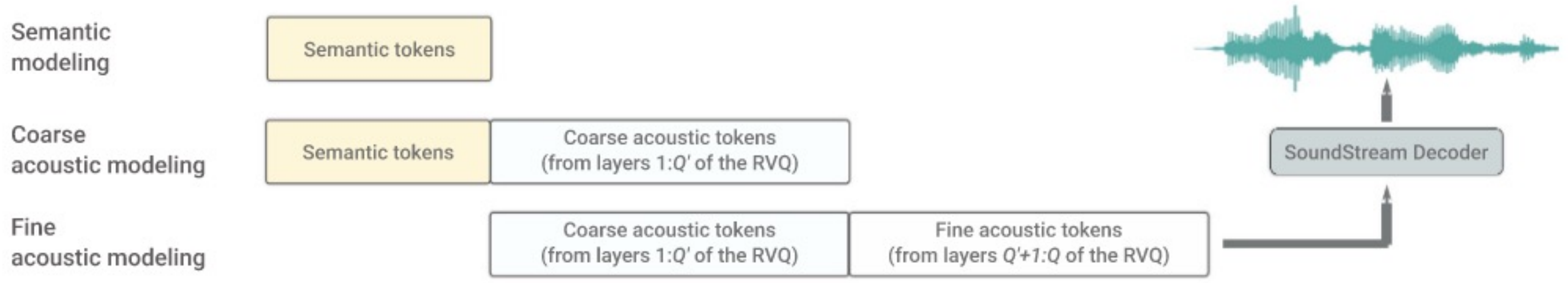
- **input**: **x** a single channel audio input of length T=16000
- **tokenizer model(s)**:
  - *acoustic*: SoundStream neural audio codec
    - vocab size: N=1024
  - *semantic*: w2v-BERT, a self-supervised model that returns a sequence of T’ discrete tokens
    - vocab size: K=1024
    - length T’ = T/640
- **decoder-only Transformer model**:
  - trained to maximize the sequence of discrete tokens from the tokenizer
  - at test time: autoregressively decodes one token at a time
- **detokinizer model**: SoundStream decoder

Figure from https://ieeexplore.ieee.org/document/10158503/?arnumber=10158503

Audio Continuation: AudioLM

AudioLM consists of

- **decoder-only Transformer model**:
  - consists of three stages in a hierarchy
  - each stage conditions on the output of the previous stage
  - separate model for each stage to keep the lengths shorter



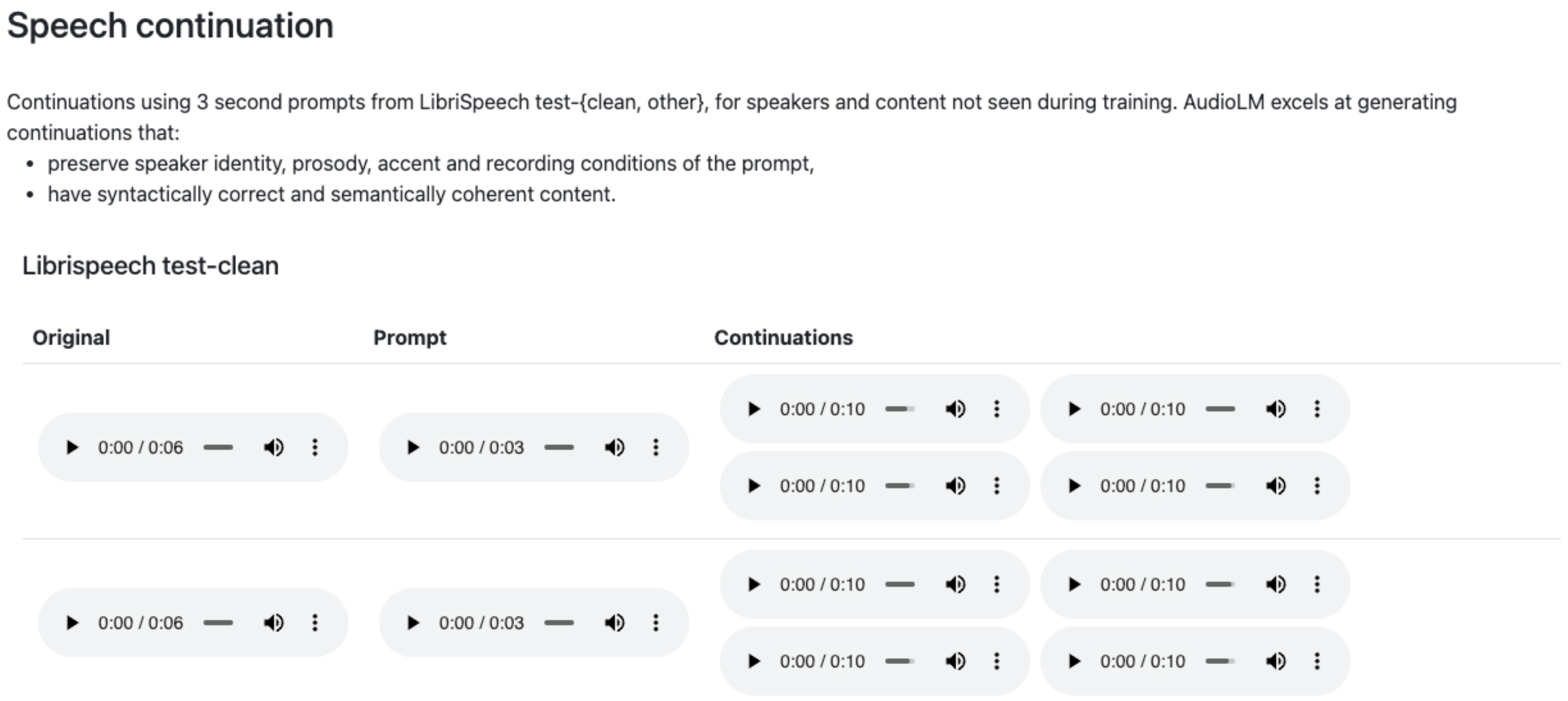
Semantic modeling
Semantic tokens
Coarse acoustic modeling
Semantic tokens
Coarse acoustic tokens (from layers 1:Q' of the RVQ)
Fine acoustic modeling
Coarse acoustic tokens (from layers 1:Q' of the RVQ)
Fine acoustic tokens (from layers Q'+1:Q of the RVQ)
SoundStream Decoder

Fig. 2. Three stages of the hierarchical modeling of semantic and acoustic tokens in AudioLM : i) semantic modeling for long-term structural coherence, ii) coarse acoustic modeling conditioned on the semantic tokens and iii) fine acoustic modeling. With the default configuration, for every semantic token there are $2Q'$ acoustic tokens in the second stage and $2(Q - Q')$ tokens in the third stage. The factor of 2 comes from the fact that the sampling rate of SoundStream embeddings is twice as that of the w2v-BERT embeddings.

Figure from https://ieeexplore.ieee.org/document/10158503/?arnumber=10158503

Audio Continuation: AudioLM

- Demos: https://google-research.github.io/seanet/audiolm/examples/



Speech continuation

Continuations using 3 second prompts from LibriSpeech test-{clean, other}, for speakers and content not seen during training. AudioLM excels at generating continuations that:

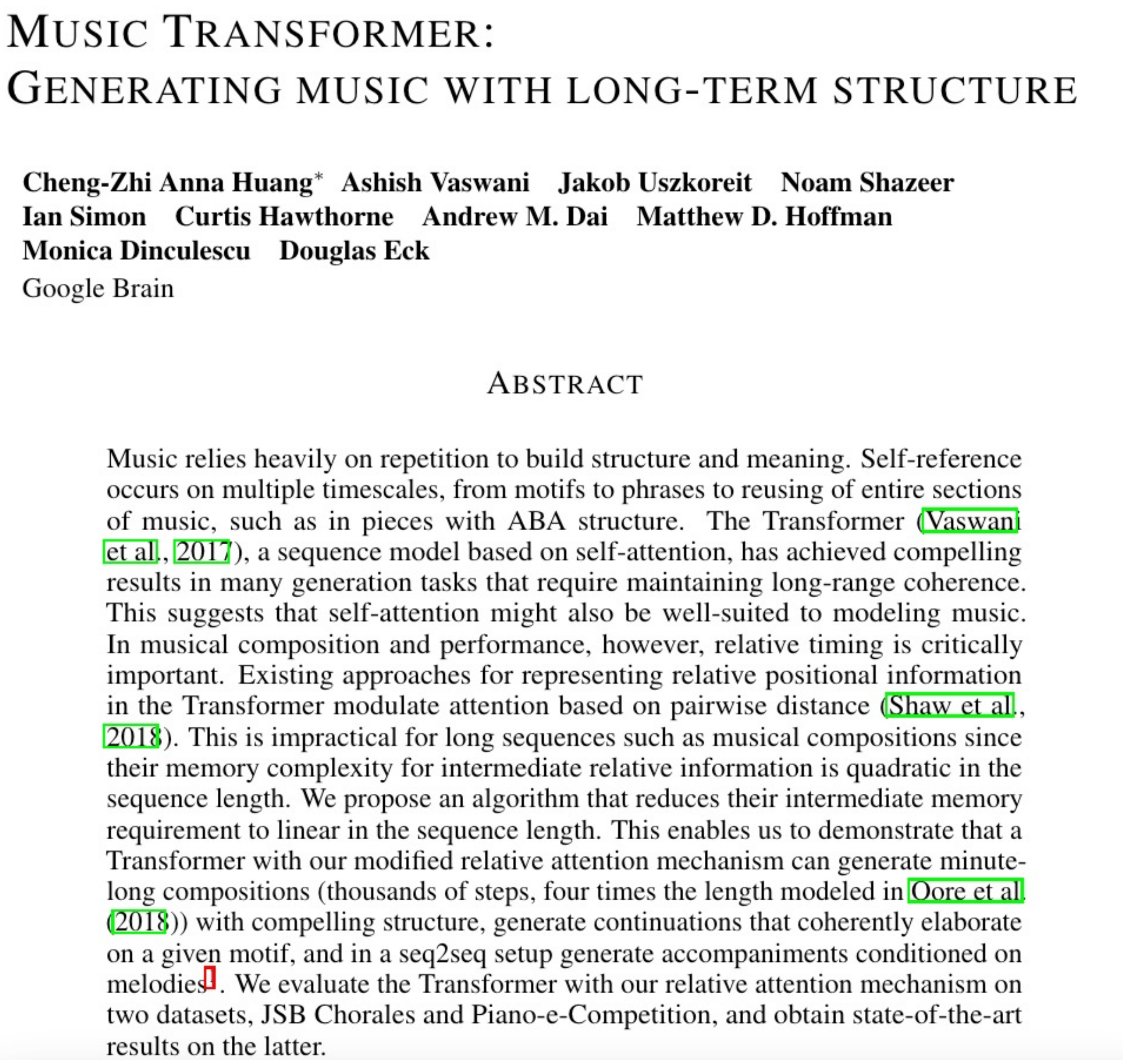
- preserve speaker identity, prosody, accent and recording conditions of the prompt,
- have syntactically correct and semantically coherent content.

Librispeech test-clean

Original	Prompt	Continuations	
0:00 / 0:06	0:00 / 0:03	0:00 / 0:10	0:00 / 0:10
		0:00 / 0:10	0:00 / 0:10
0:00 / 0:06	0:00 / 0:03	0:00 / 0:10	0:00 / 0:10
		0:00 / 0:10	0:00 / 0:10

Figure from

Music Generation: Music Transformer



MUSIC TRANSFORMER:
GENERATING MUSIC WITH LONG-TERM STRUCTURE

Cheng-Zhi Anna Huang* **Ashish Vaswani** **Jakob Uszkoreit** **Noam Shazeer**
Ian Simon **Curtis Hawthorne** **Andrew M. Dai** **Matthew D. Hoffman**
Monica Dinculescu **Douglas Eck**
Google Brain

ABSTRACT

Music relies heavily on repetition to build structure and meaning. Self-reference occurs on multiple timescales, from motifs to phrases to reusing of entire sections of music, such as in pieces with ABA structure. The Transformer (Vaswani et al., 2017), a sequence model based on self-attention, has achieved compelling results in many generation tasks that require maintaining long-range coherence. This suggests that self-attention might also be well-suited to modeling music. In musical composition and performance, however, relative timing is critically important. Existing approaches for representing relative positional information in the Transformer modulate attention based on pairwise distance (Shaw et al., 2018). This is impractical for long sequences such as musical compositions since their memory complexity for intermediate relative information is quadratic in the sequence length. We propose an algorithm that reduces their intermediate memory requirement to linear in the sequence length. This enables us to demonstrate that a Transformer with our modified relative attention mechanism can generate minute-long compositions (thousands of steps, four times the length modeled in Oore et al. (2018)) with compelling structure, generate continuations that coherently elaborate on a given motif, and in a seq2seq setup generate accompaniments conditioned on melodies[1]. We evaluate the Transformer with our relative attention mechanism on two datasets, JSB Chorales and Piano-e-Competition, and obtain state-of-the-art results on the latter.

Figure from http://arxiv.org/abs/1809.04281

Music Generation: Music Transformer

3.4 Memory efficient implementation of relative position-based attention

We improve the implementation of relative attention by reducing its intermediate memory requirement from $O(L^2D)$ to $O(LD)$, with example lengths shown in Table 1. We observe that all of the terms we need from $QR^\top$ are already available if we directly multiply Q with E^r, the relative position embedding. After we compute $QE^{r\top}$, its (i_q, r) entry contains the dot product of the query in position i_q with the embedding of relative distance r. However, each relative logit (i_q, j_k) in the matrix S^{rel} from Equation 3 should be the dot product of the query in position i_q and the embedding of the relative distance $j_k - i_q$, to match up with the indexing in $QK^\top$. We therefore need to "skew" $QE^{r\top}$ so as to move the relative logits to their correct positions, as illustrated in Figure 1 and detailed in the next section. The time complexity for both methods are $O(L^2D)$, while in practice our method is 6x faster at length 650.

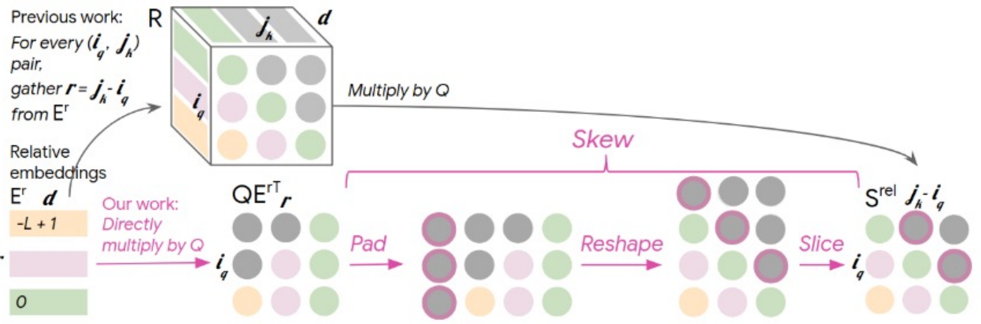


Figure 1: Relative global attention: the bottom row describes our memory-efficient "skewing" algorithm, which does not require instantiating R (top row, which is $O(L^2D)$). Gray indicates masked or padded positions. Each color corresponds to a different relative distance.

3.5 Relative local attention

For very long sequences, the quadratic memory requirement of even baseline Transformer is impractical. Local attention has been used for example in Wikipedia and image generation (Liu et al., 2018; Parmar et al., 2018) by chunking the input sequence into non-overlapping blocks. Each block then attends to itself and the one before, as shown by the smaller thumbnail on the top right corner of Figure 2.

To extend relative attention to the local case, we first note that the right block has the same configuration as in the global case (see Figure 1) but much smaller: $(\frac{L}{M})^2$ (where M is the number of blocks, and N be the resulting block length) as opposed to L^2. The left block is unmasked with relative indices running from -1 (top right) to -$2N+1$ (bottom left). Hence, the learned E^r for the local case has shape $(2N-1, N)$.

Similar to the global case, we first compute $QE^{r\top}$ and then use the following procedure to skew it to have the same indexing as $QK^\top$, as illustrated in Figure 2.

1. Pad a dummy column vector of length N after the rightmost column.
2. Flatten the matrix and then pad with a dummy row of length $N-1$.
3. Reshape the matrix to have shape $(N+1, 2N-1)$.
4. Slice that matrix to retain only the first N rows and last N columns, resulting in a (N, N) matrix.

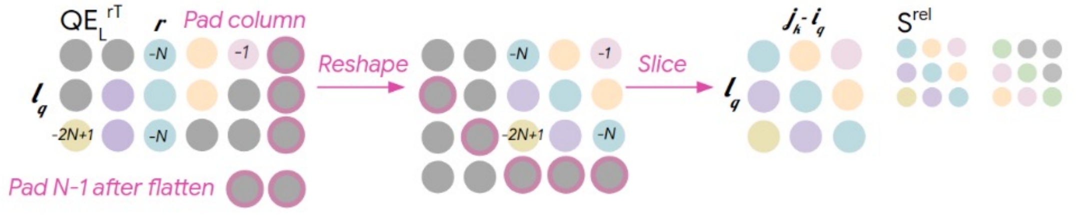


Figure 2: Relative local attention: the thumbnail on the right shows the desired configuration for S^{rel}. The "skewing" procedure is shown from left to right.

Figure from http://arxiv.org/abs/1809.04281

Music Generation: Music Transformer

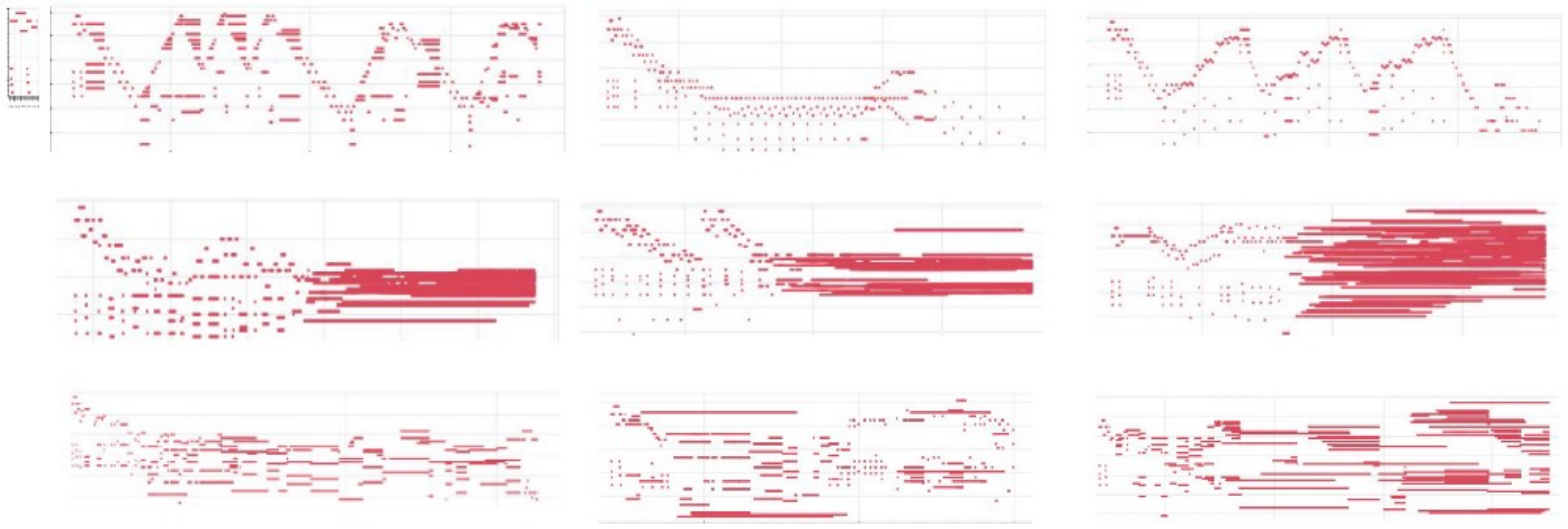
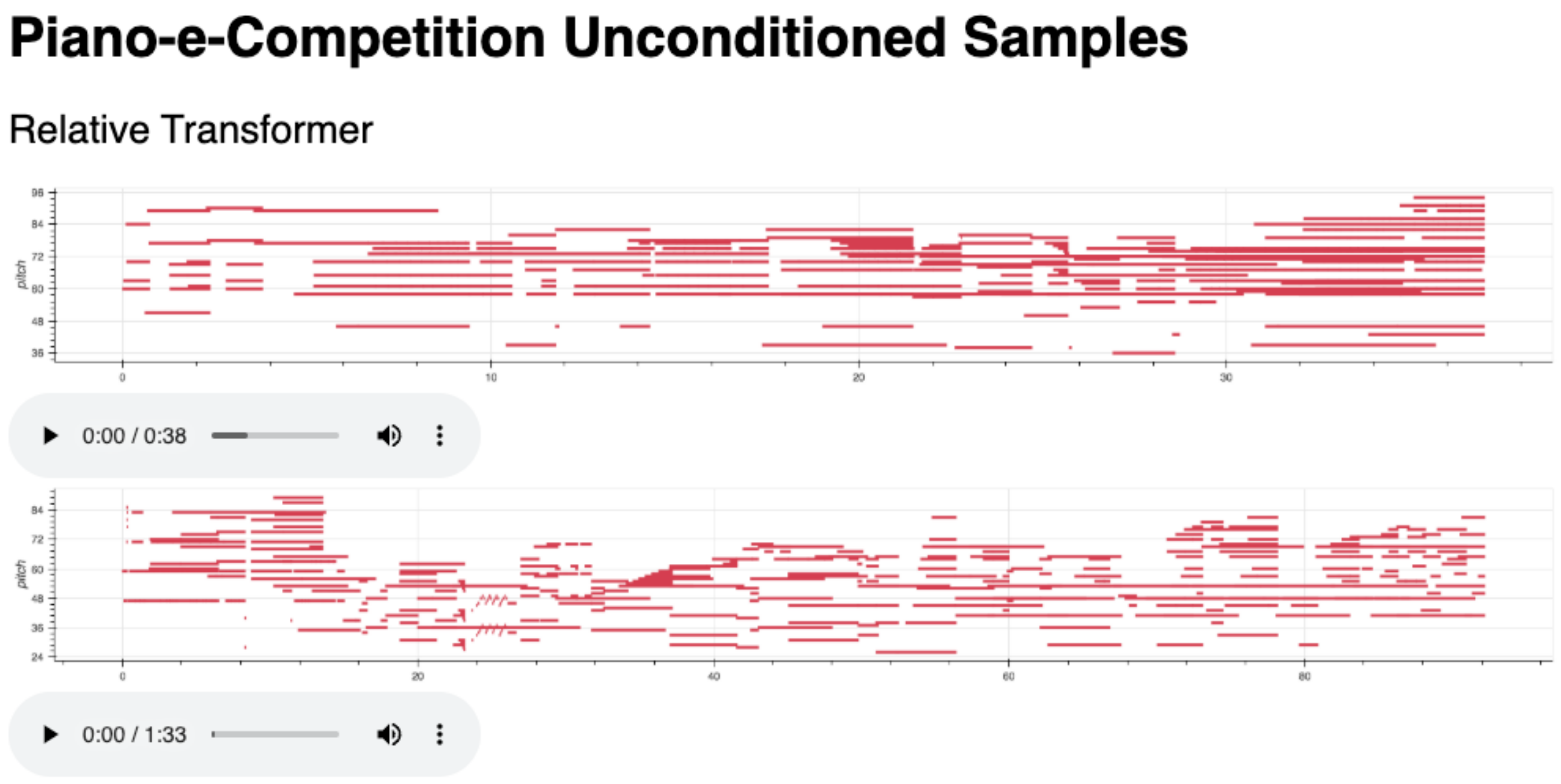


Figure 4: Comparing how models continue a prime (top left). Repeated motives and structure are seen in samples from Transformer with relative attention (top row), but less so from baseline Transformer (middle row) and PerformanceRNN (LSTM) (bottom row).

Figure from http://arxiv.org/abs/1809.04281

Music Generation: Music Transformer

- Demos: https://storage.googleapis.com/music-transformer/index.html


Piano-e-Competition Unconditioned Samples

Relative Transformer

Figure from http://arxiv.org/abs/1809.04281

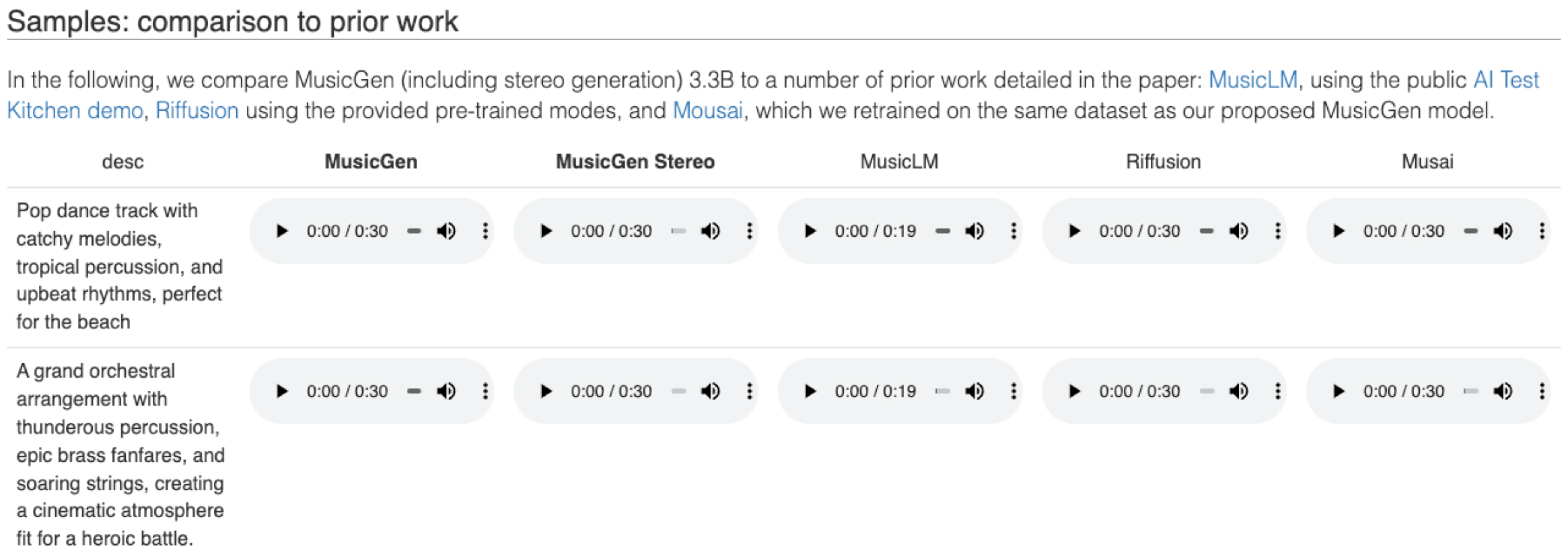
Text-to-Music Generation: MusicGen

- MusicGen Model
  - **audio tokenizer**: EnCodec [Défossez et al., 2022], which is a convolutional auto-encoder
  - **codebook interleaving**: multiple token sequences are predicted in parallel
  - **text prompt model**: T5, Flan-T5, or CLAP
  - **melody prompt model**: information bottleneck
  - **decoder model**: transformer LM up to 3.3B parameters

Figure from https://proceedings.neurips.cc/paper_files/paper/2023/hash/94b472a1842cd7c56dcb125fb2765fbd-Abstract-Conference.html

Text-to-Music Generation: MusicGen

- Demo: https://ai.honu.io/papers/musicgen/



Samples: comparison to prior work

In the following, we compare MusicGen (including stereo generation) 3.3B to a number of prior work detailed in the paper: MusicLM, using the public AI Test Kitchen demo, Riffusion using the provided pre-trained modes, and Mousai, which we retrained on the same dataset as our proposed MusicGen model.

desc	MusicGen	MusicGen Stereo	MusicLM	Riffusion	Musai
Pop dance track with catchy melodies, tropical percussion, and upbeat rhythms, perfect for the beach	0:00 / 0:30	0:00 / 0:30	0:00 / 0:19	0:00 / 0:30	0:00 / 0:30
A grand orchestral arrangement with thunderous percussion, epic brass fanfares, and soaring strings, creating a cinematic atmosphere fit for a heroic battle.	0:00 / 0:30	0:00 / 0:30	0:00 / 0:19	0:00 / 0:30	0:00 / 0:30

Figure from https://proceedings.neurips.cc/paper_files/paper/2023/hash/94b472a1842cd7c56dcb125fb2765fbd-Abstract-Conference.html

Speech+Text to Text: SpeechVerse

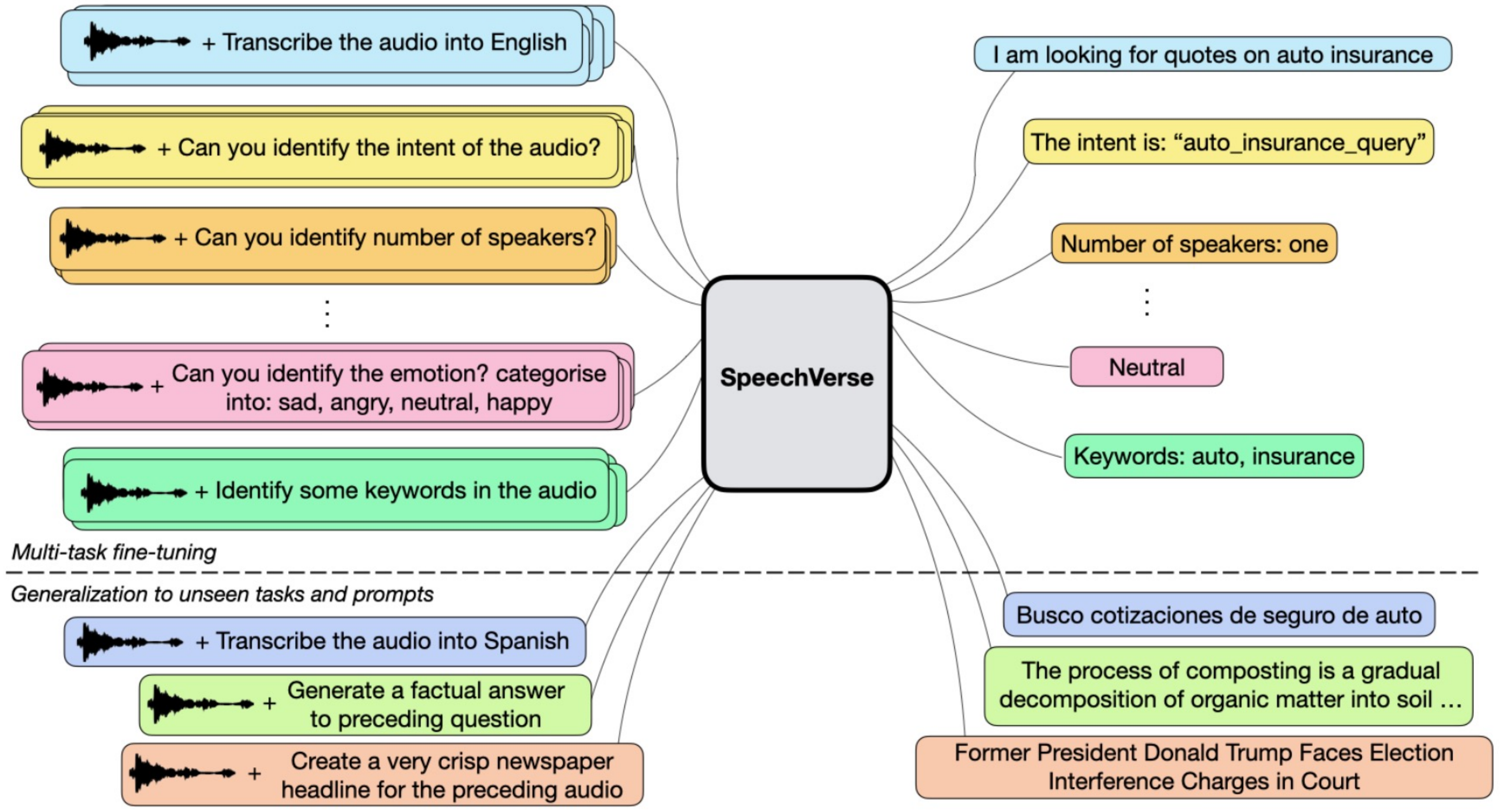

+ Transcribe the audio into English
+ Can you identify the intent of the audio?
+ Can you identify number of speakers?
⋮
+ Can you identify the emotion? categorise into: sad, angry, neutral, happy
+ Identify some keywords in the audio
Multi-task fine-tuning
Generalization to unseen tasks and prompts
+ Transcribe the audio into Spanish
+ Generate a factual answer to preceding question
+ Create a very crisp newspaper headline for the preceding audio
SpeechVerse
I am looking for quotes on auto insurance
The intent is: "auto_insurance_query"
Number of speakers: one
⋮
Neutral
Keywords: auto, insurance
Busco cotizaciones de seguro de auto
The process of composting is a gradual decomposition of organic matter into soil …
Former President Donald Trump Faces Election Interference Charges in Court

Figure 1: Schematic diagram of the SpeechVerse framework.

Figure from https://arxiv.org/abs/2405.08295

Speech+Text to Text: SpeechVerse

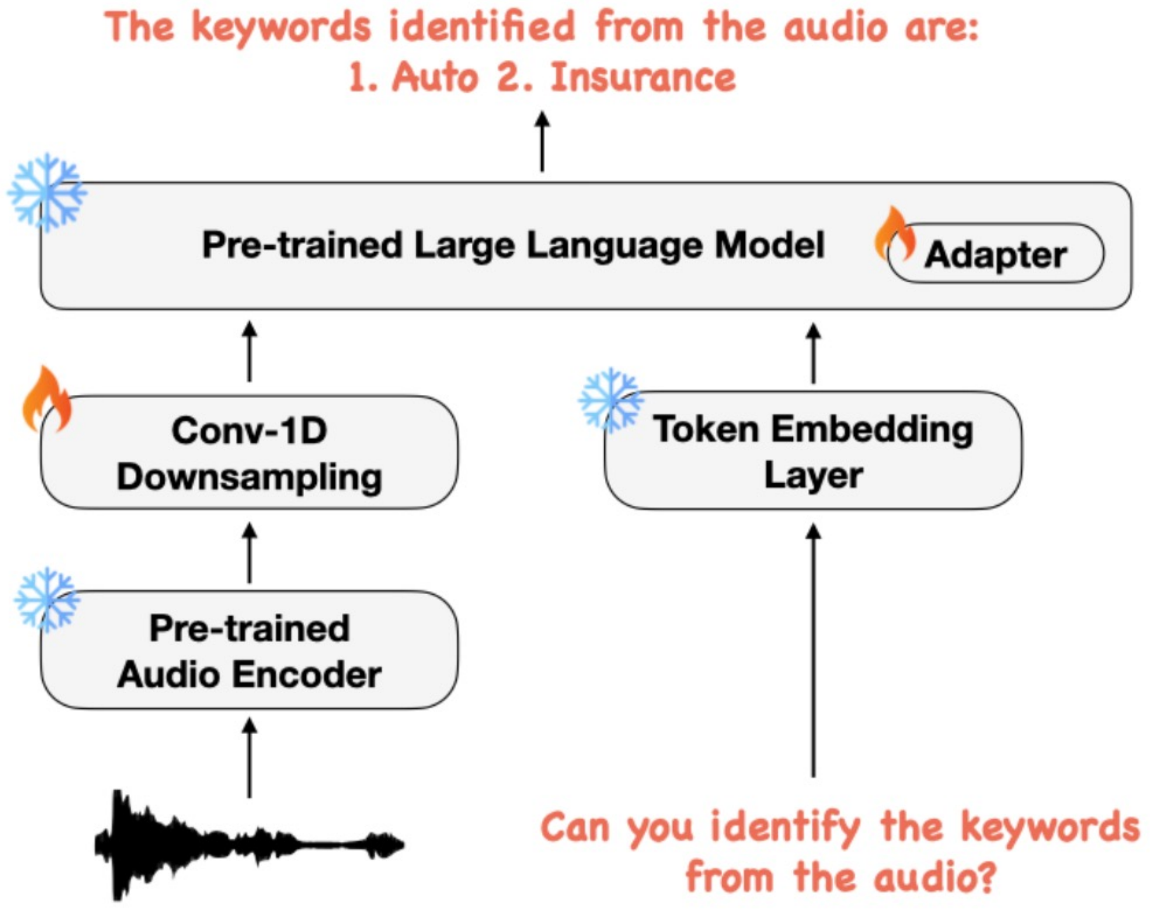

The keywords identified from the audio are: 1. Auto 2. Insurance
Pre-trained Large Language Model
Adapter
Conv-1D Downsampling
Token Embedding Layer
Pre-trained Audio Encoder
Can you identify the keywords from the audio?

Figure 2: Block diagram of the SpeechVerse architecture.

Figure from https://arxiv.org/abs/2405.08295