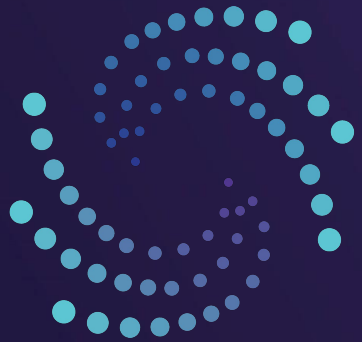


Deep Learning School

DLS

Speech LLMs

Part 1: Audio-Conditioned LLMs

Максименко
Александр

Senior DLE, GigaChat Audio

Plan

- Why LLM
- Why AudioLLM
- What is Audio LLM
- Types of AudioLLM
- Audio-conditioned LLM tasks
- Audio Input: encoders & speech tokens
- Integration Audio embeddings into LLM
- Training stages and data
- Case Study

Why LLM

Human-like Interaction

Models like ChatGPT showed millions that AI could understand nuance, write coherently, and generate human-like text.

Accessibility

APIs and open-source models allow developers and businesses to easily integrate powerful AI, fueling rapid adoption.

Versatility

One model can translate, summarize, write code, and answer questions—a "Swiss Army knife" for knowledge tasks.

Why LLM

Scalable Architecture

The "Transformer" architecture is highly parallel and scales predictably. More data & compute = better performance.

Massive Data

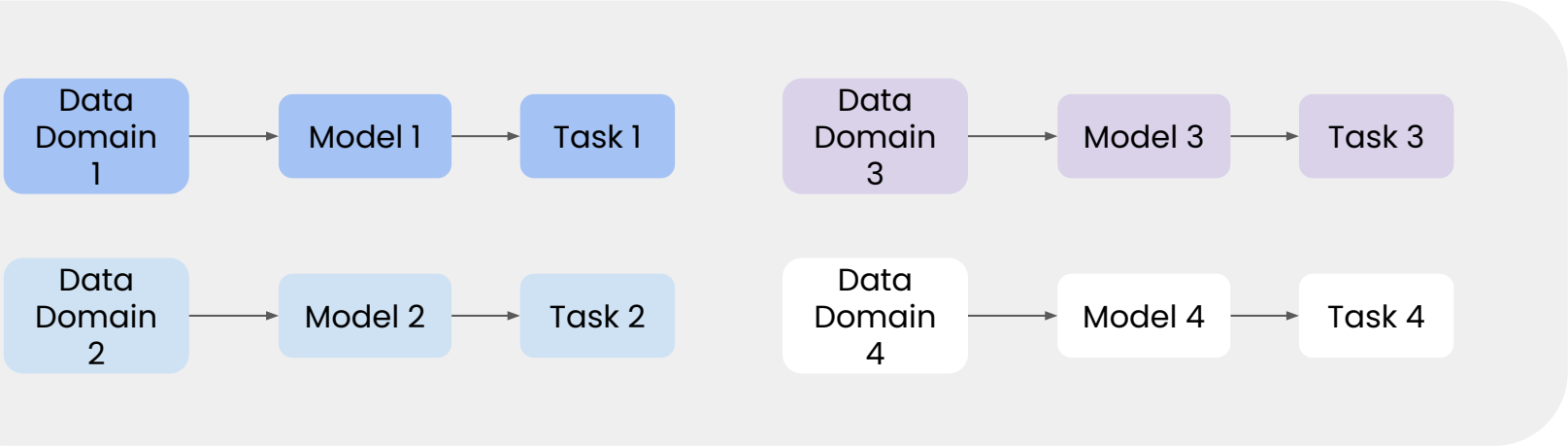
Trained on internet-scale text, models learn vast knowledge, language patterns, and reasoning abilities.

Emergent Abilities

At scale, models develop new skills they weren't explicitly trained for, such as "zero-shot" learning and reasoning.

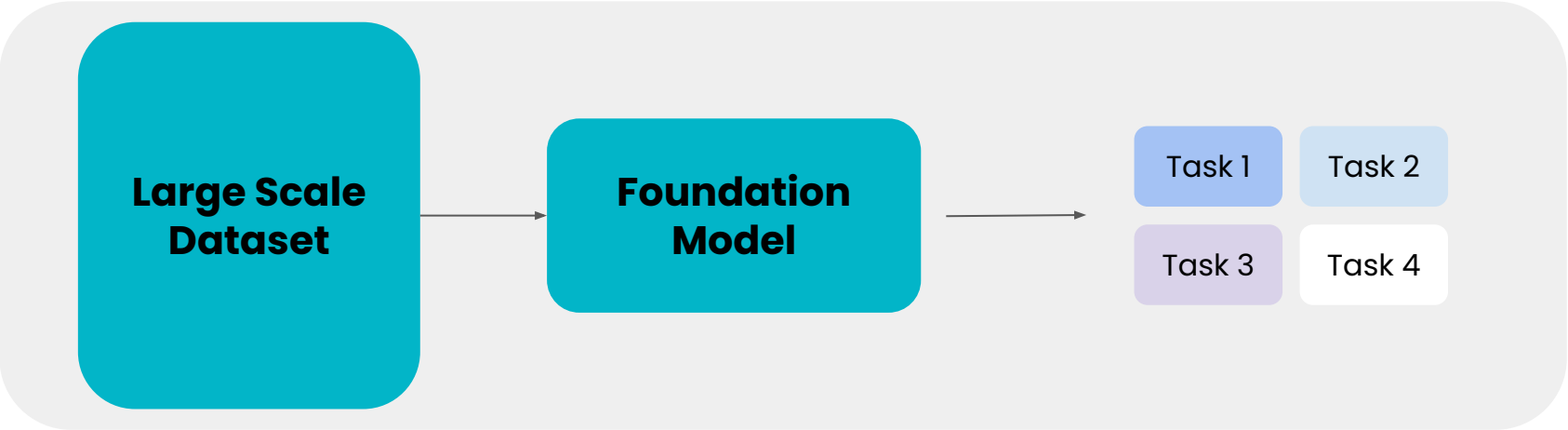
Why LLM

Traditional Approach: Specialized Models



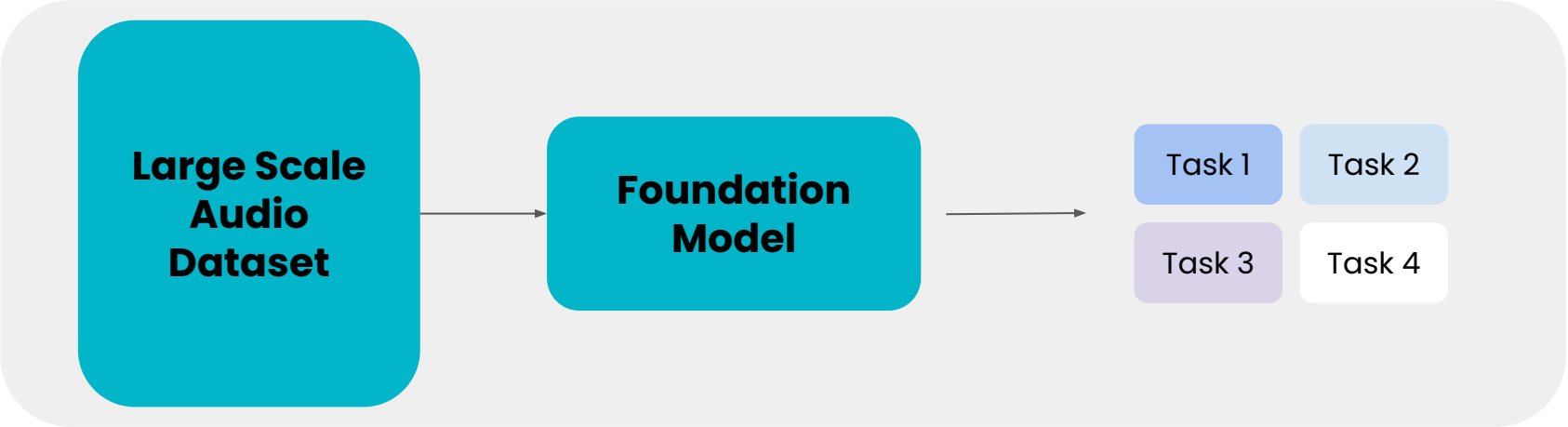
Why LLM

LLM-based Approach: One for All

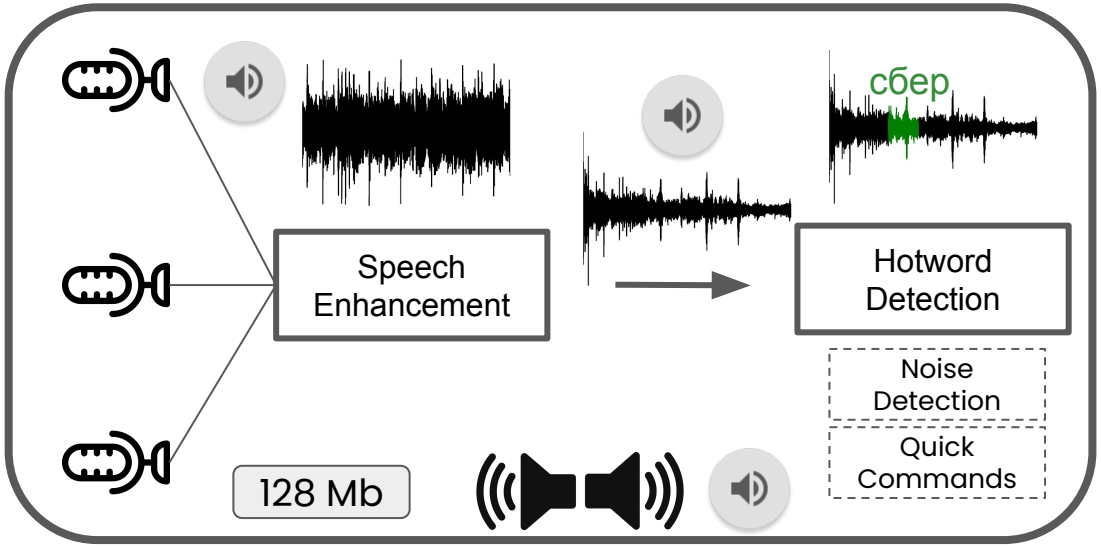


Why AudioLLM

AudioLLM-based Approach: One for All



Voice Assistant Pipeline



Partial Recognition
Prefetching
Punctuation

Hotword Validation
Intended Query Detection
Endpoint Detection
Speech Recognition
Speaker Identification
Whisper Detection
Speech Separation
Emotion Recognition

сбер измени пожалуйста условия по моему кредиту

Intent Recognition

Dialog Management

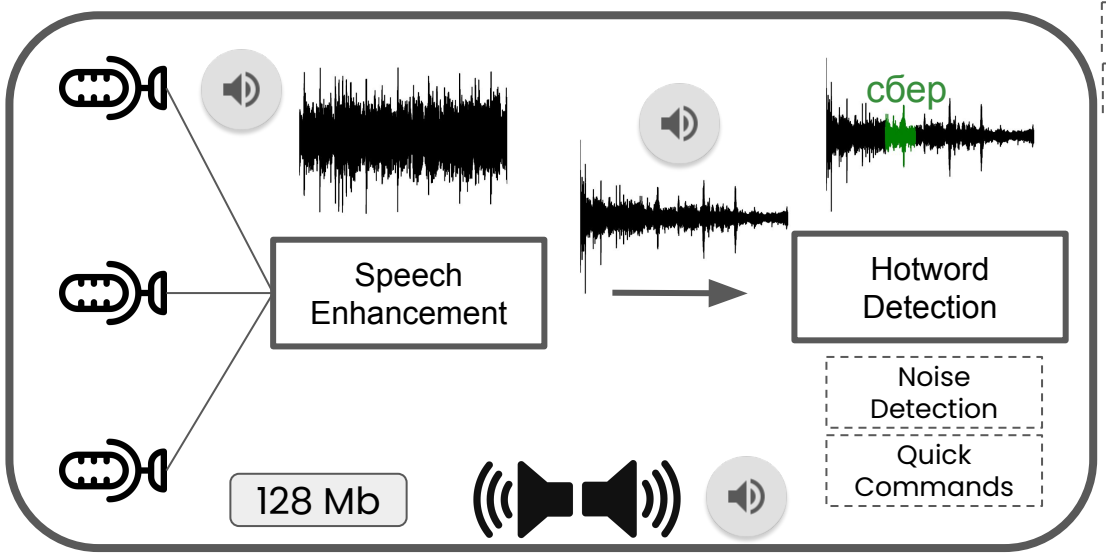
Что нужно изменить?

Language Generation

Text Normalization

Speech Synthesis

Voice Assistant Pipeline



Partial Recognition
Prefetching
Punctuation

Hotword Validation
Intended Query Detection
Endpoint Detection
Speech Recognition
Speaker Identification
Whisper Detection
Speech Separation
Emotion Recognition

сбер измени пожалуйста условия по моему кредиту

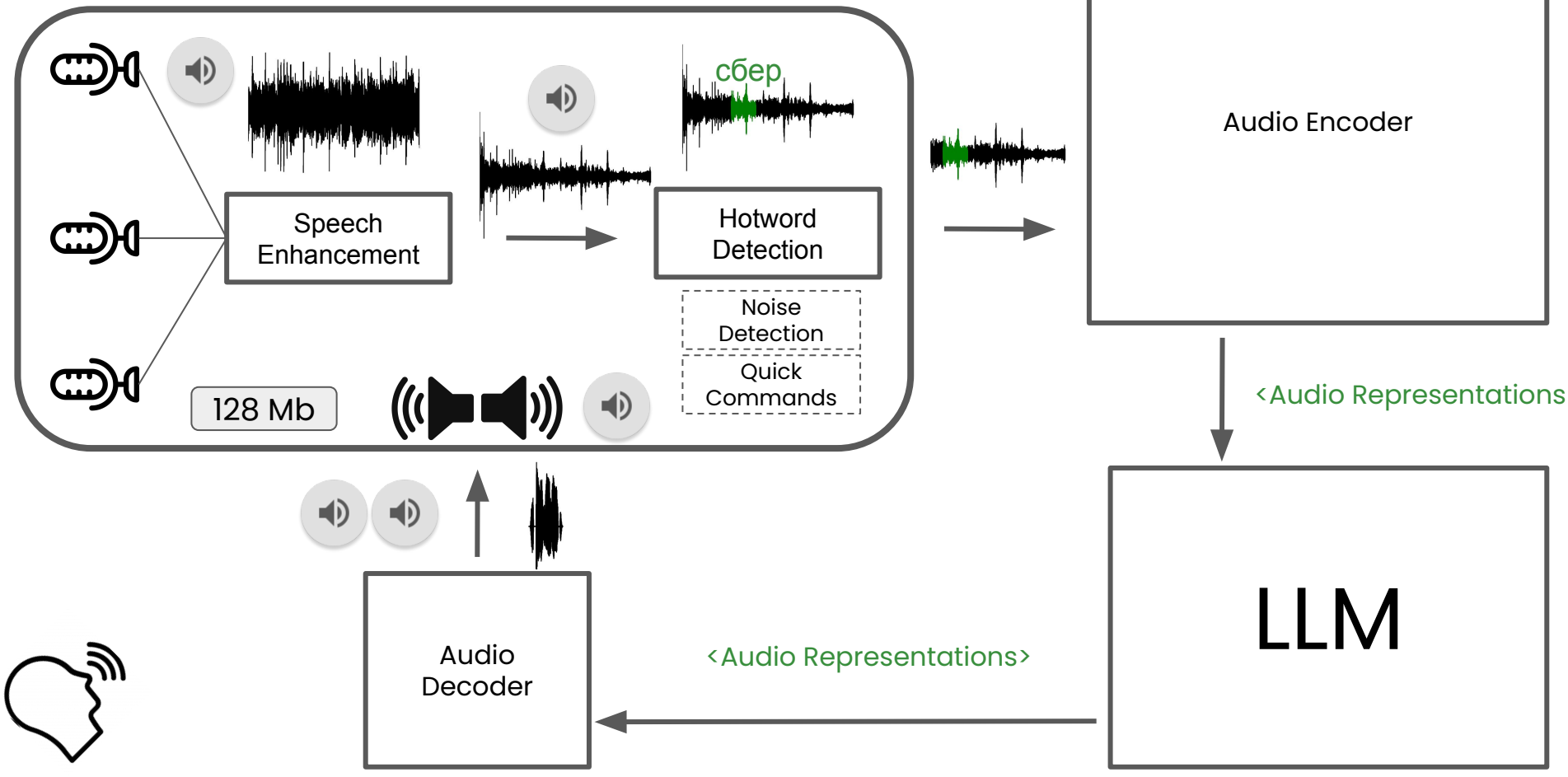
LLM

Что нужно изменить?

Text Normalization

Speech Synthesis

Voice Assistant Pipeline: Endgame



AudioLLM For ASR

- Hard to compare Approaches
- Hard enough to compare Models
- Insufficiency of WER

AudioLLM For ASR

Task	Llama 3 8B	Llama 3 70B	Whisper	SeamlessM4T	Gemini 1.0 Ultra	Gemini 1.5 Pro
MLS (English)	4.9	4.4	6.2 (v2)	6.5	4.4	**4.2**
LibriSpeech (test-other)	3.4	**3.1**	4.9 (v2)	6.2	-	-
VoxPopuli (English)	6.2	**5.7**	7.0 (v2)	7.0	-	-
FLEURS (34 languages)	9.6	**8.2**	14.4 (v3)	11.7	-	-

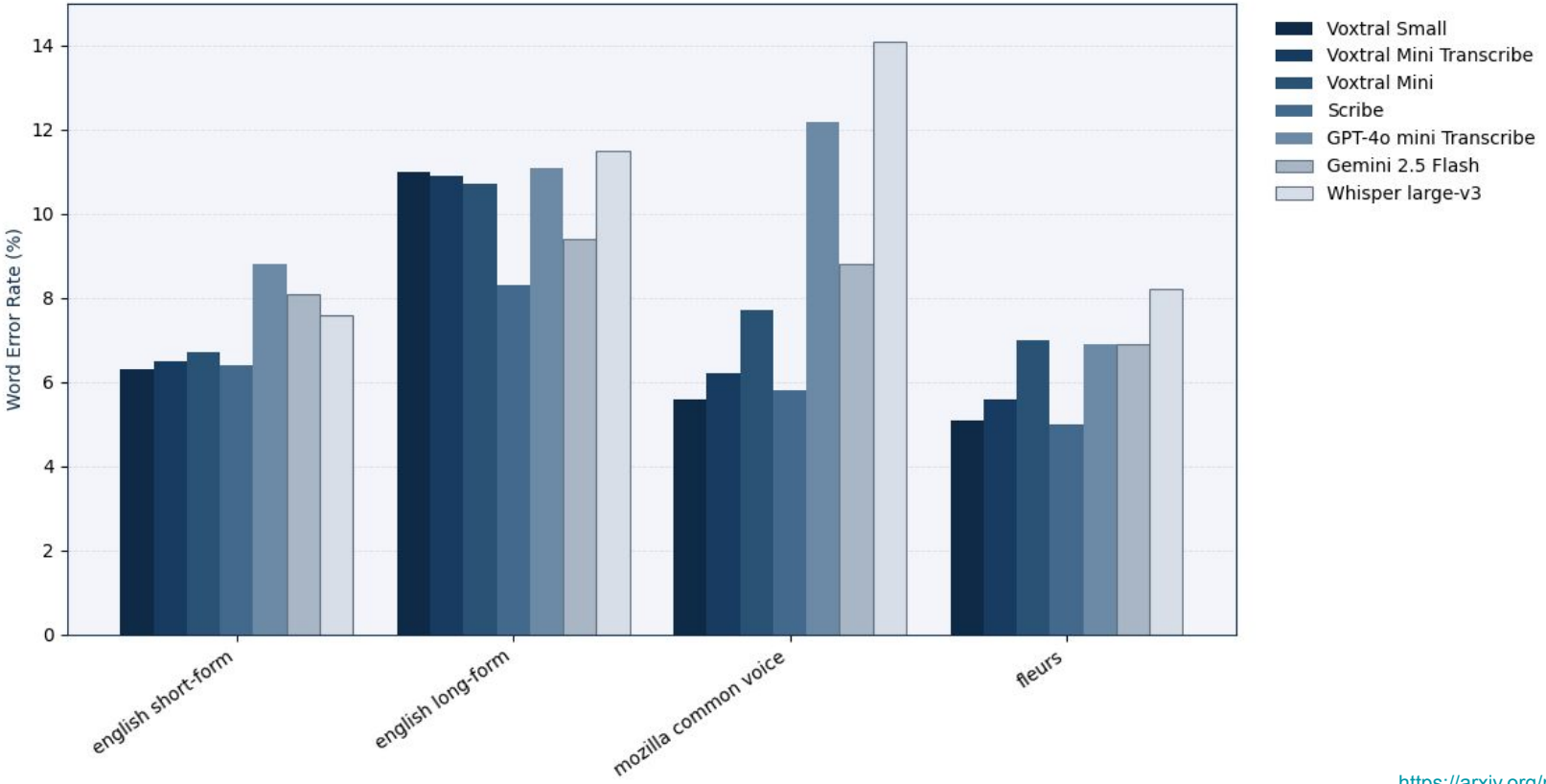
https://arxiv.org/pdf/2407.21783

AudioLLM For ASR

Category	Task	Metric	USM	Whisper	1.0 Pro	1.0 Ultra	1.5 Flash	1.5 Pro
ASR	YouTube (en-us)	WER	5.8%	6.5%	4.8%	**4.7%**	4.9%	4.8%
ASR	YouTube (52 lang)	WER	22.8%	41.4%	22.5%	**21.0%**	23.8%	22.6%
ASR	Multilingual LibriSpeech (en-us)	WER	7.0%	6.2%	4.8%	4.4%	5.2%	**4.2%**
ASR	FLEURS (55 lang)	WER	11.2%	16.6%	6.4%	**6.0%**	9.8%	6.5%
AST	Covost 2 (20 lang)	BLEU (↑)	31.5	29.4	40.0	**41.0**	36.1	39.4

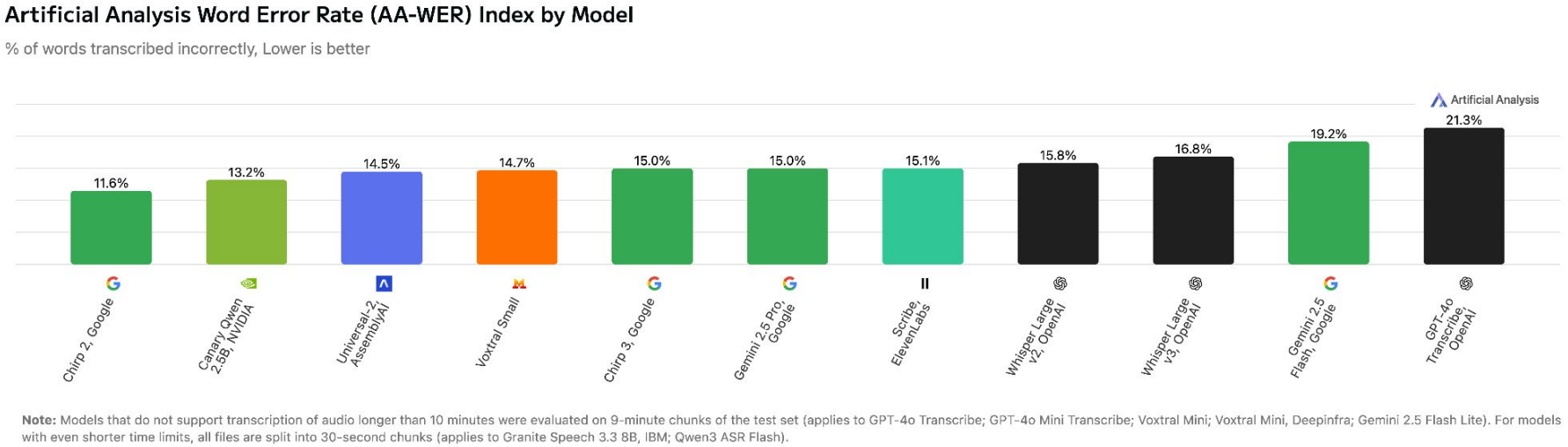
https://arxiv.org/pdf/2312.11805

AudioLLM For ASR



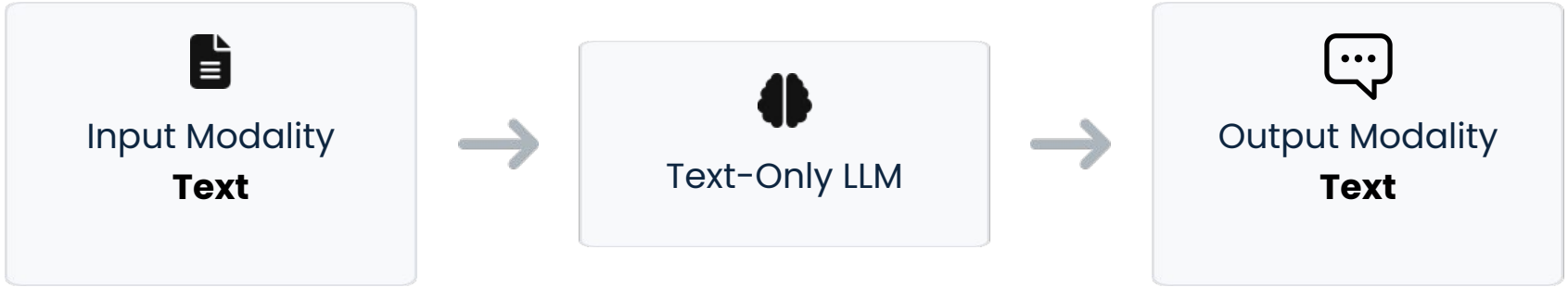
https://arxiv.org/pdf/2507.13264v1

AudioLLM For ASR

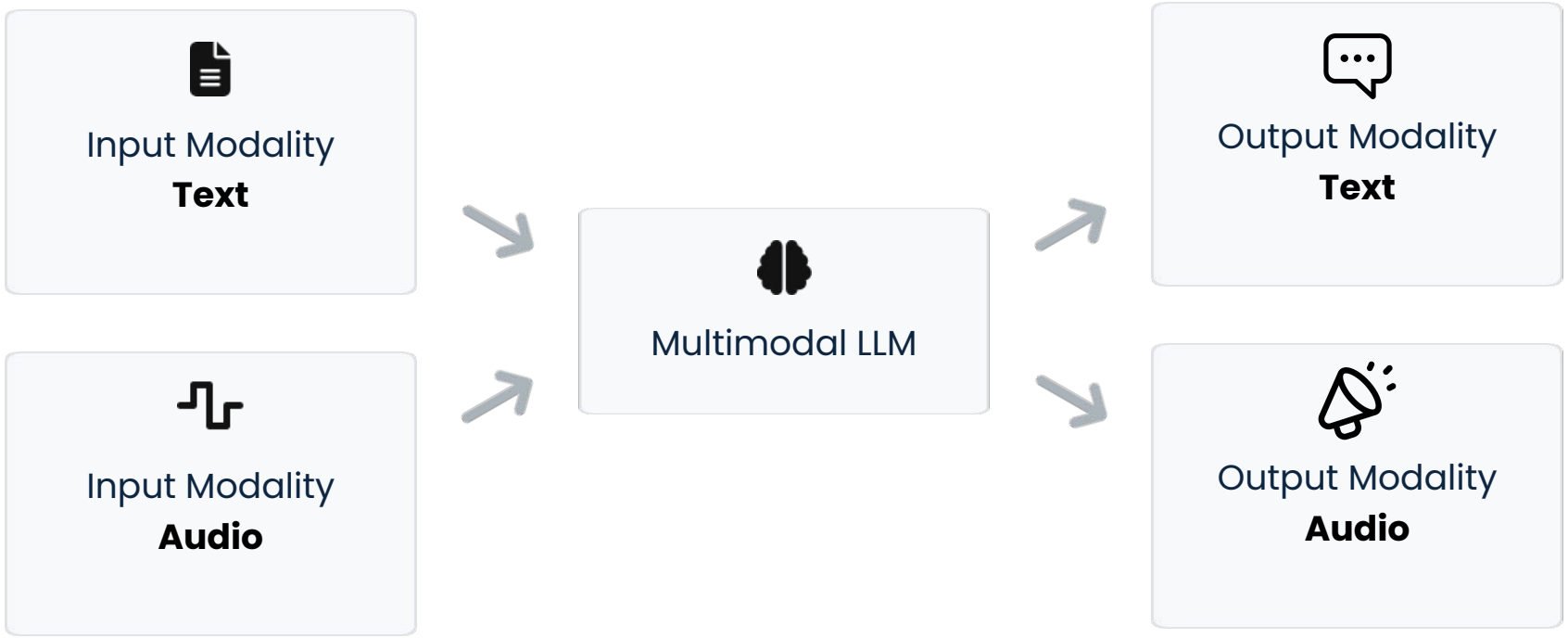


https://artificialanalysis.ai/speech-to-text?error-rate=word-error-rate#artificial-analysis-word-error-rate-aa-wer-index-by-model

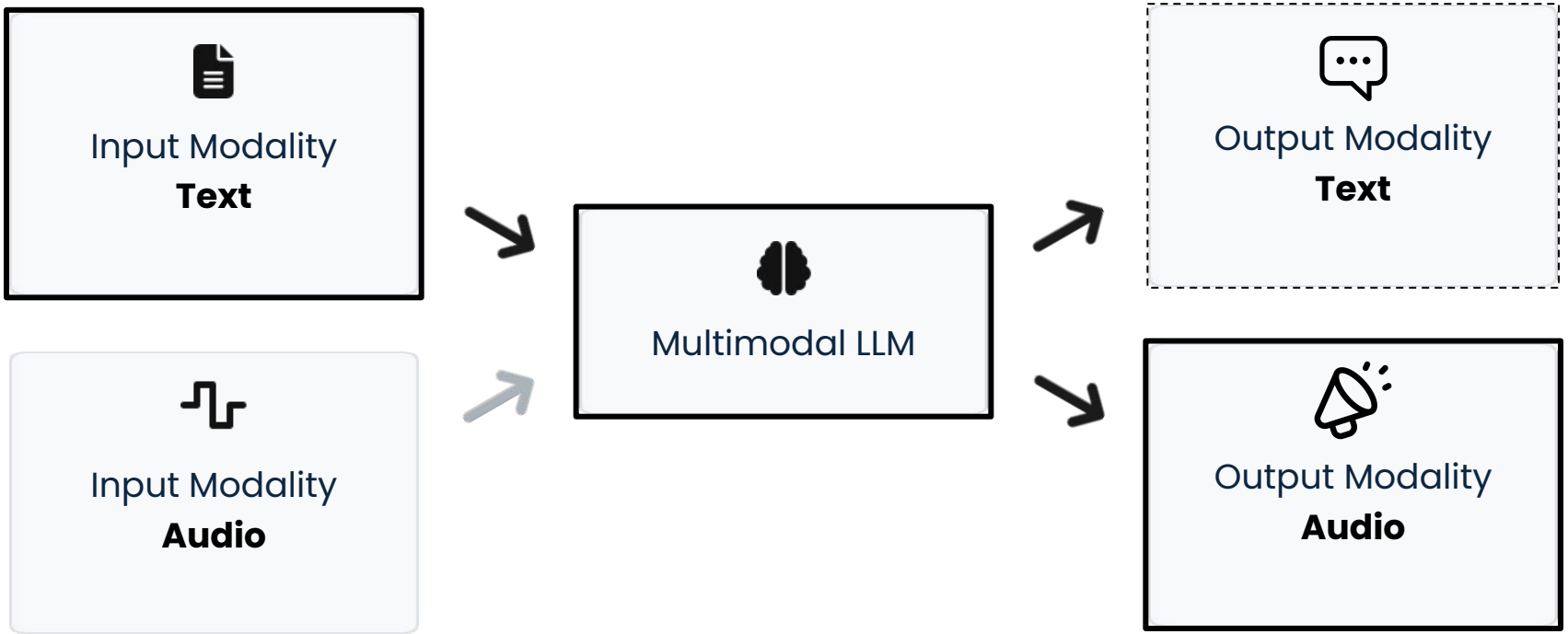
What is LLM



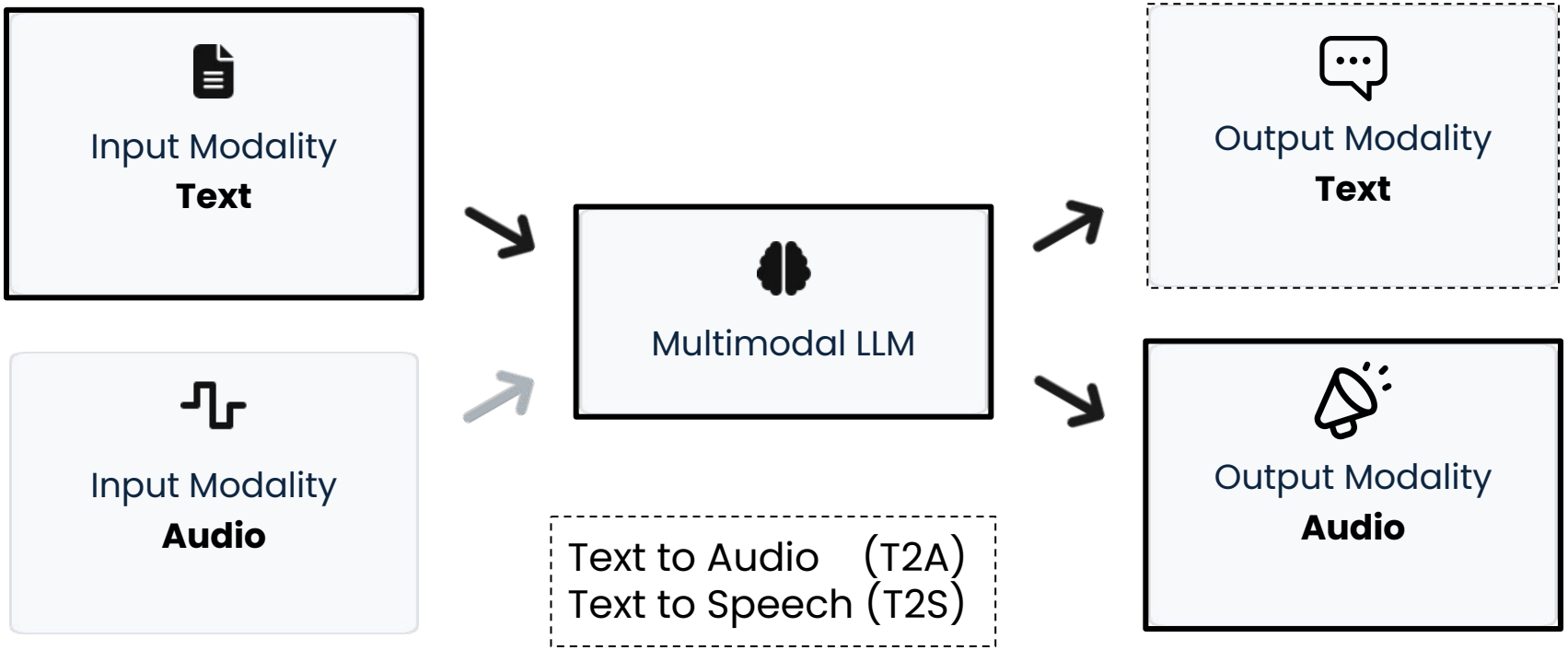
What is AudioLLM



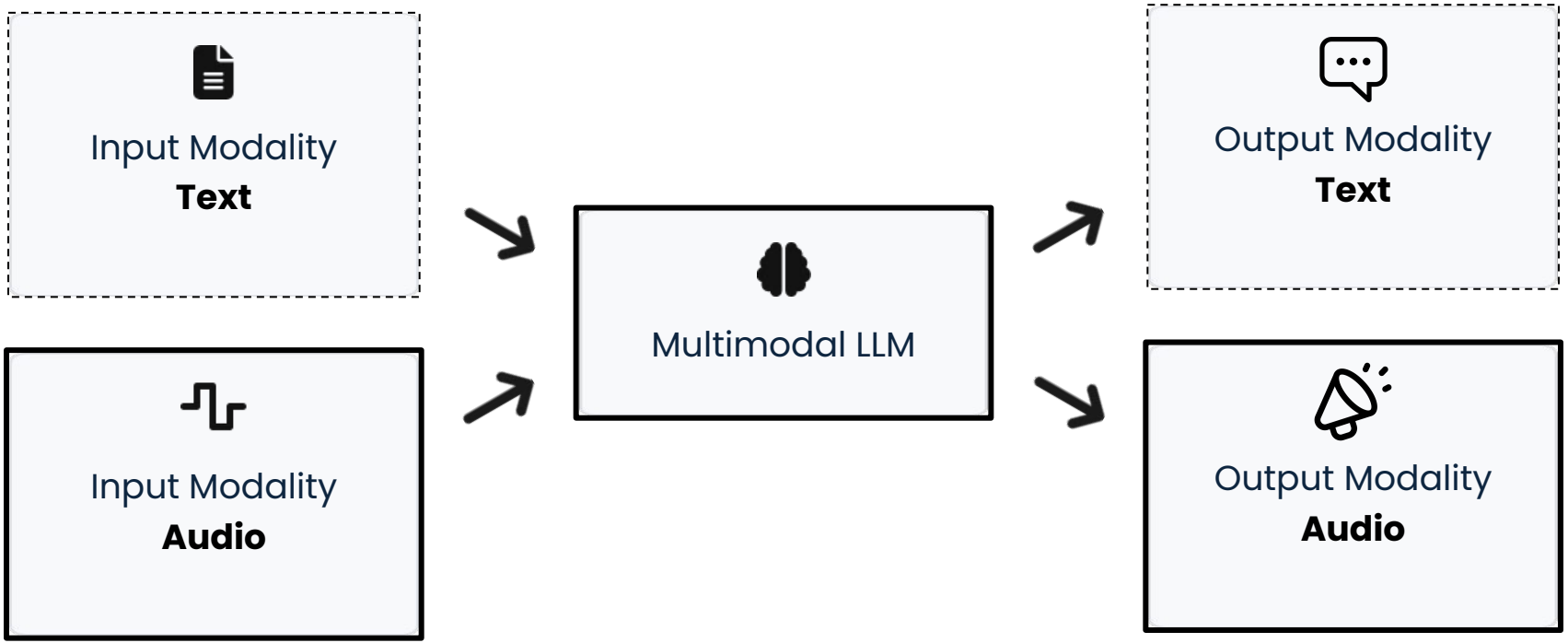
What is AudioLLM



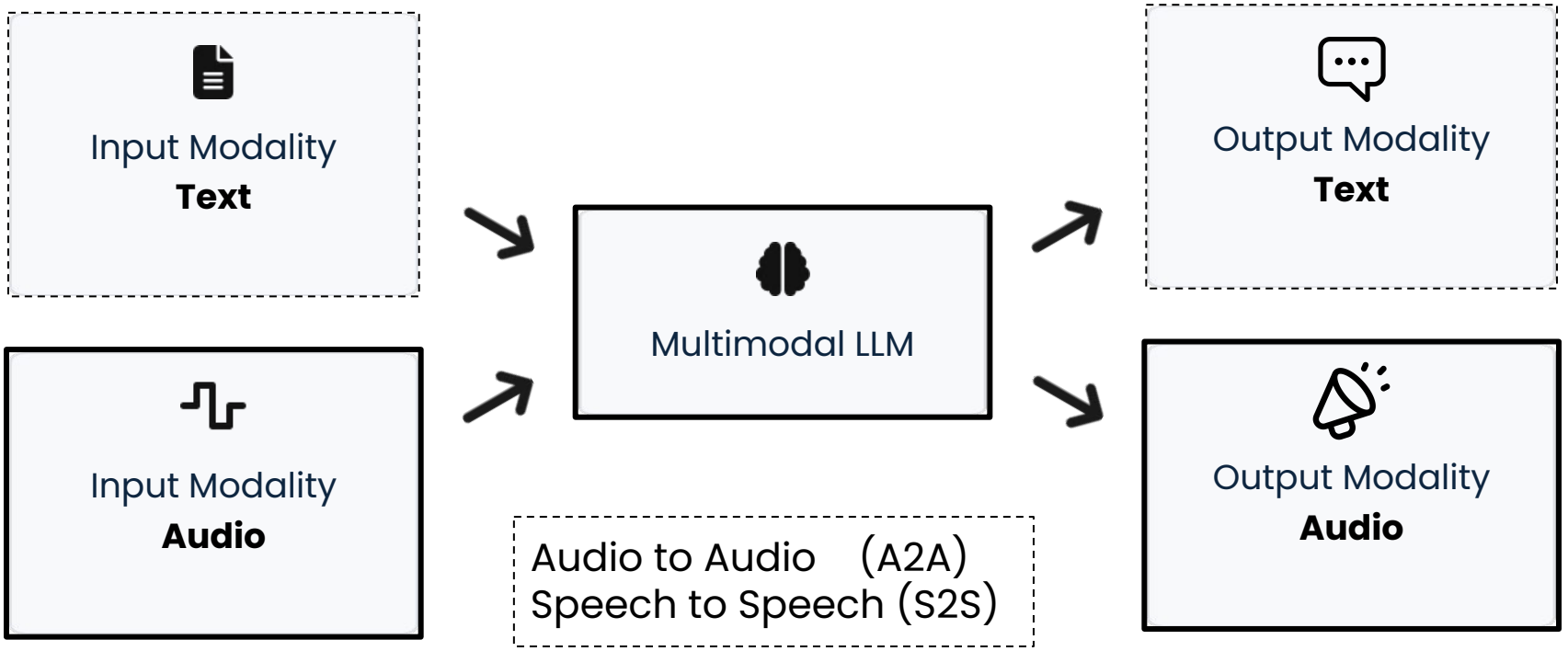
What is AudioLLM



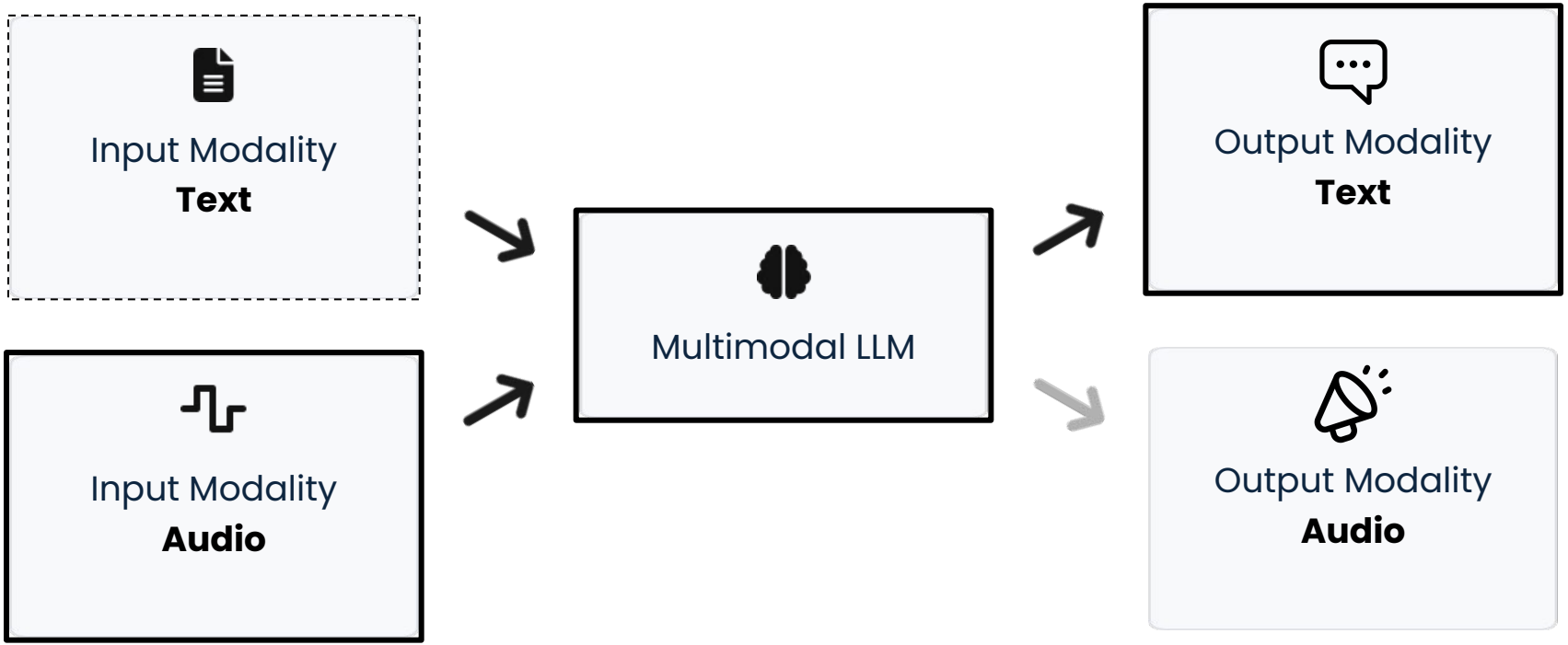
What is AudioLLM



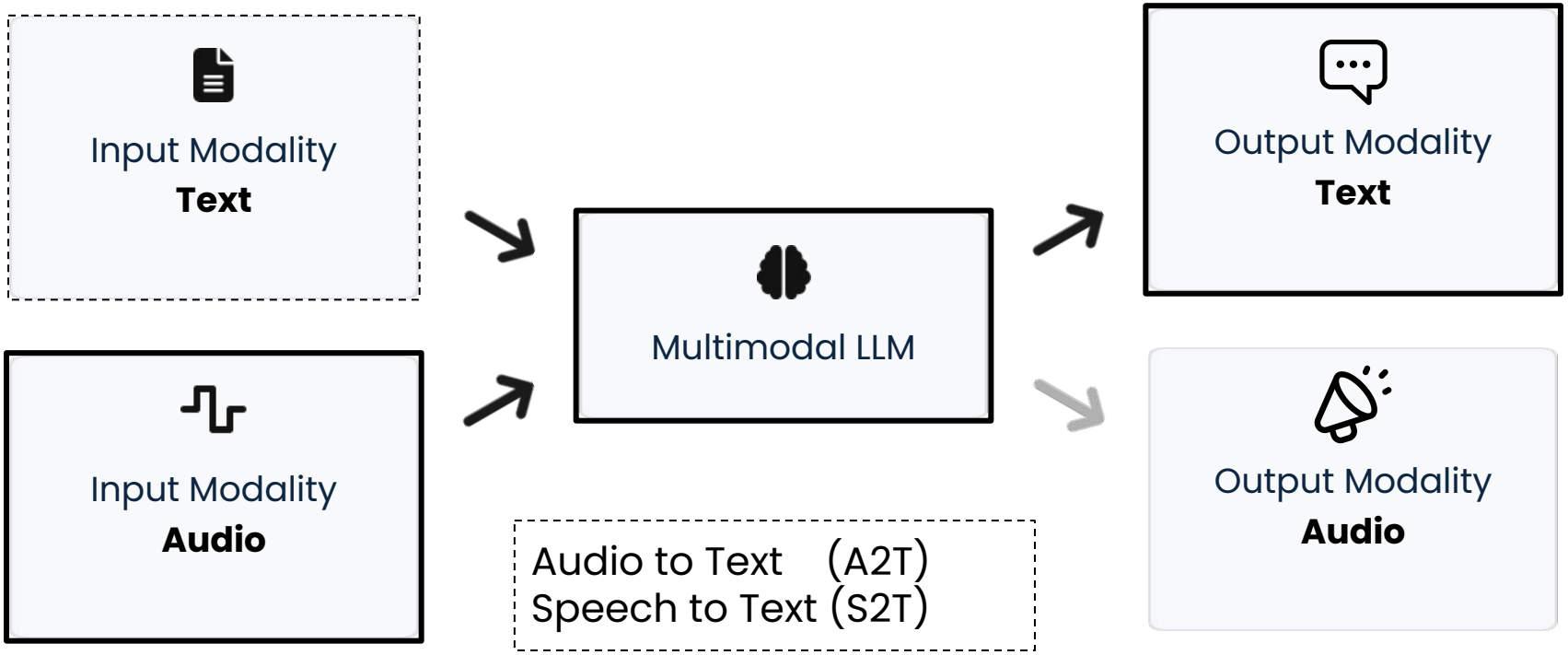
What is AudioLLM



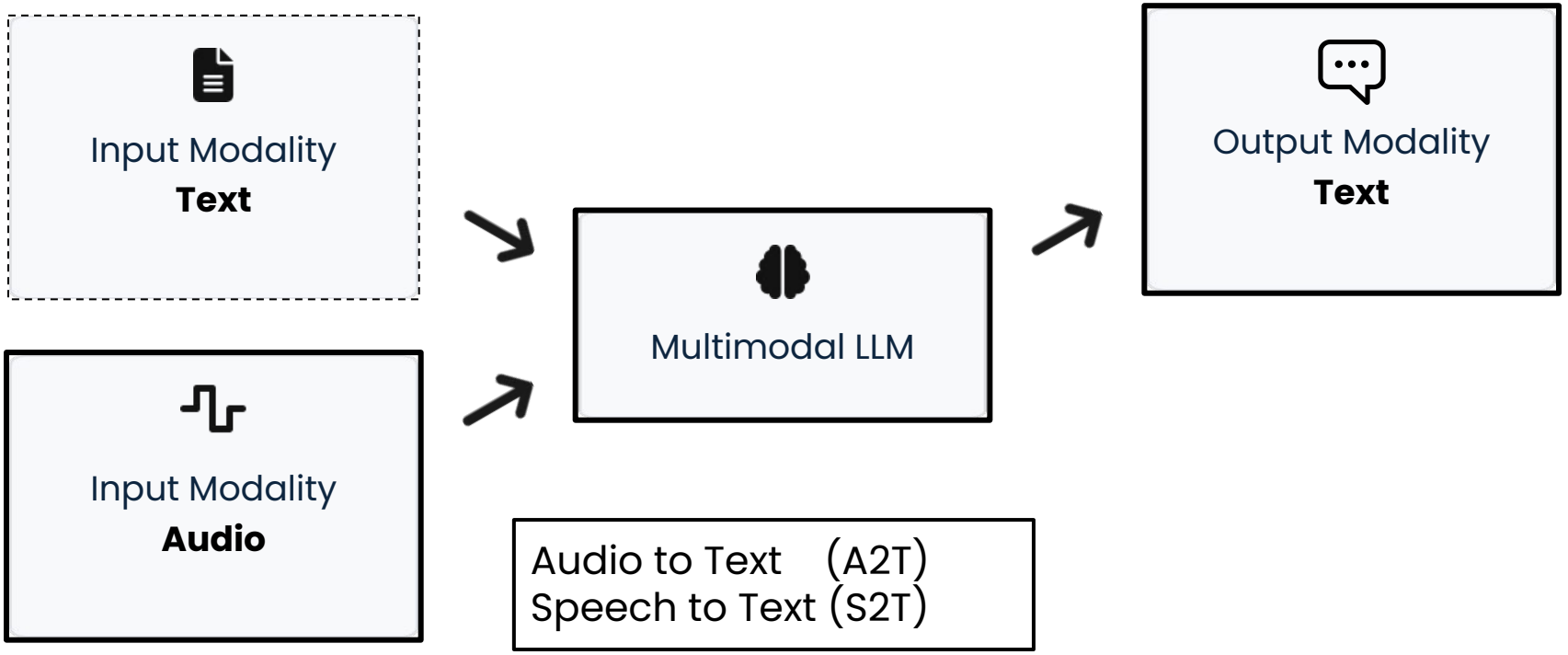
What is AudioLLM



What is AudioLLM



What is AudioLLM



Audio-Conditioned LLM abilities

ASR

Automatic Speech Recognition (Speech-to-Text)

AST

Automatic Speech Translation (Speech-to-Translation)

AQA

Audio Question Answering (Answering questions about audio, Close-Ended)

Speaker Tasks

Speaker Diarization & Identification (Who spoke and when?)

AED

Audio Event Detection (Identifying non-speech sounds)

Any other task

Captioning, Spoken Language Understanding, Speaker Emotion Recognition, ...

Audio-Conditioned LLM abilities

ASR

WORD ERROR RATE (WER)

AST

BLEU, ROGUE, METEOR

AQA

MMLU-Like setup: Accuracy

Speaker Tasks

Diarization Metrics, SID Metrics, ...

AED

Classification metrics, mAP

Any other task

?

Audio-Conditioned LLM abilities

Summarization ?	Contextuality ?	Empathy ?
Usefullness ?	Beauty ?	•••

Human Preferences: Pairwise Comparison

Rater Instructions

You will be shown a user prompt and two anonymous answers, **Answer A** and **Answer B**.

Your Task: Read both answers carefully and decide which answer is **better**. If they are very similar or equally good/bad, you may choose 'Tie'.

Criteria: Judge based on helpfulness, accuracy, and clarity. A better answer directly addresses the user's prompt, is factually correct, and is well-written.

Example: An answer that is comprehensive is better than one that is too brief.

User Prompt:

"What is an AudioLLM?"

[Answer A]

```
It's an LLM for audio.
```

[Answer B]

```
An AudioLLM is a system that combines an audio encoder
with an LLM core to understand and process audio tasks like
ASR and translation.
```

[A is better] [B is better] [Tie]

Human Preferences: Single Answer Grading

Rater Instructions

You will be shown a user prompt and a single model's answer.

Your Task: Rate the quality of the answer on a scale from **1 (Bad)** to **10 (Excellent)** based on the provided rubric.

Rubric:

- **1-3:** Incorrect, harmful, or irrelevant.
- **4-6:** Partially correct but incomplete or poorly written.
- **7-8:** Helpful and accurate, but could be more detailed.
- **9-10:** Excellent, comprehensive, and well-written.

Please provide a brief justification for your score.

User Prompt:

"What is WER?"

[Answer]

```
WER is Word Error Rate, a metric for ASR.
```

Overall Score:

6 (Select 1-10)

Justification:

Correct, but very incomplete. It doesn't explain what the metric means or how it's used. Falls into the 'partially correct' category.

Human Preferences: Reference-Guided Grading

Rater Instructions

You will be shown a prompt, a model's **"Answer"**, and a "gold" **"Reference Answer"**.

Your Task: Compare the model's "Answer" to the "Reference". Rate how well the model's answer captures the *meaning* of the reference, on a scale of 1-5.

Rubric (Meaning):

- **1:** Contradicts the reference.
- **3:** Captures some, but not all, of the meaning.
- **5:** Captures all of the meaning. Paraphrasing is acceptable.

User Prompt:

"Define ASR."

[Reference Answer]

```
Automatic Speech Recognition is a task of converting
spoken language into written text.
```

[Model Answer]

```
ASR is when a computer writes down what you say.
```

Meaning Score:

5 (Select 1-5)

Justification:

The model's answer is a perfect simple paraphrase of the reference. It captures the full meaning.

Human Preferences

Pros:

Cons:

LLM-as-a-Judge: Pairwise Comparison

System Prompt

You are a meticulous and objective evaluator. You will be given a user prompt and two AI-generated answers (A and B).

Your Task: Your goal is to determine which answer is better. First, provide a step-by-step critique of both answers. Second, compare them directly. Finally, provide your verdict in the format [[Winner: A]], [[Winner: B]], or [[Winner: Tie]].

Criteria: Judge based on helpfulness, accuracy, and clarity. Do not let the length or writing style of one answer overly influence your decision.

User Input:

```
[Prompt] "What is an AudioLLM?"

       [Answer A] "It's an LLM for audio."

       [Answer B] "An AudioLLM is a system that combines..."
```

LLM Judge Output:

Critique: *Answer A is technically correct but lacks any useful information. Answer B is comprehensive, defines the components, and lists example tasks.*

Comparison: *Answer B is clearly superior as it is far more helpful and detailed. Answer A is a low-quality response.*

```
[[Winner: B]]
```

LLM-as-a-Judge: Single Answer Grading

System Prompt

You are an evaluator. You will be given a user prompt and one AI answer. You must rate the answer on a scale of 1-10 based on the provided rubric.

Rubric:

- **[1-3]:** Incorrect, harmful, or irrelevant.
- **[4-6]:** Partially correct but incomplete or poorly written.
- **[7-8]:** Helpful and accurate, but could be more detailed.
- **[9-10]:** Excellent, comprehensive, and well-written.

First, provide your reasoning for the score. Then, output the score in the exact format [[Score: X]].

User Input:

```
[Prompt] "What is WER?"

       [Answer] "WER is Word Error Rate, a metric for ASR."
```

LLM Judge Output:

Reasoning: *The answer is correct and provides the definition of the acronym, but it lacks any depth. It doesn't explain what the metric means or how it's used, stopping it from being a truly helpful answer. This fits the "Partially correct but incomplete" category.*

```
[[Score: 6]]
```

LLM-as-a-Judge: Reference-Guided Grading

System Prompt

You are an evaluator. You will be given a 'Reference Answer' and a 'Model Answer' for a given prompt.

Your Task: Score the 'Model Answer' from 1-5 based *only* on its semantic equivalence to the 'Reference Answer'.

Rubric (Meaning):

- **[1]:** Contradicts the reference.
- **[3]:** Captures some, but not all, of the reference's meaning.
- **[5]:** Captures all of the reference's meaning. Paraphrasing is acceptable.

First, provide your reasoning. Then, output the score in the format [[Score: X]].

User Input:

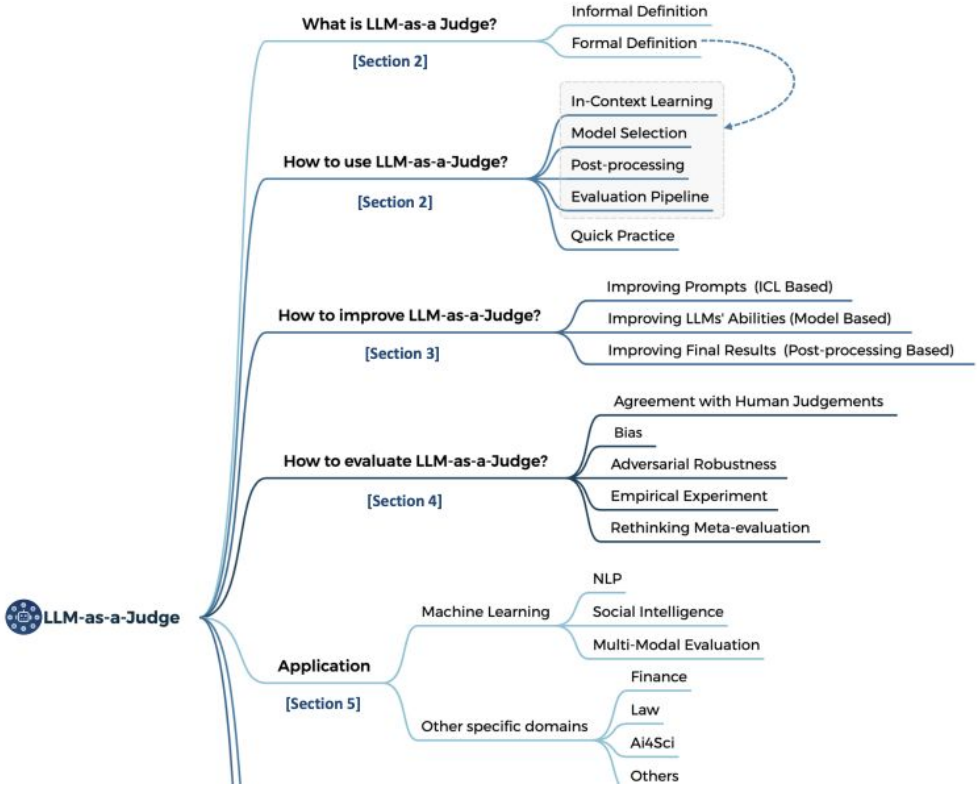
```
[Reference] "Automatic Speech Recognition is a task of
converting spoken language into written text."


      [Model Answer] "ASR is when a computer writes down
what you say."
```

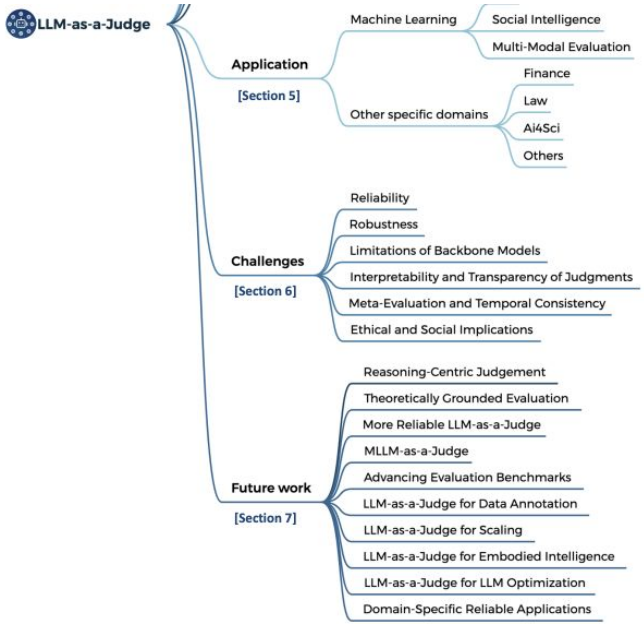
LLM Judge Output:

Reasoning: *The model's answer uses simpler language ("writes down what you say") but it is a correct and complete paraphrase of the reference ("converting spoken language into written text"). It fully captures the semantic meaning.*

[[Score: 5]]

LLM-as-a-Judge



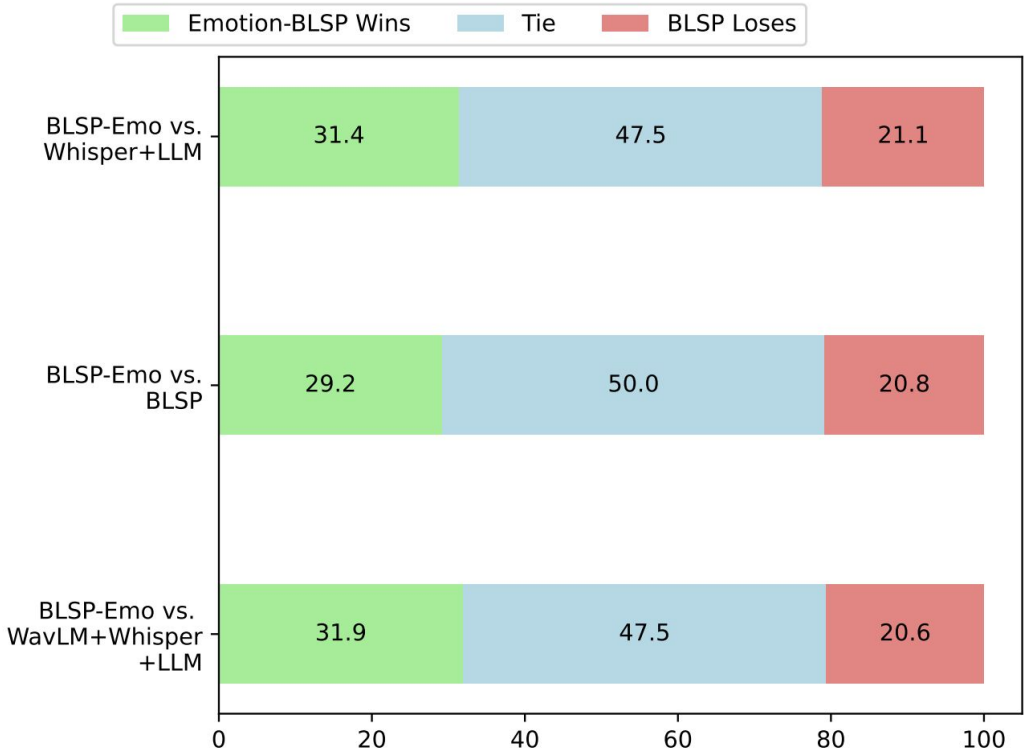


https://arxiv.org/pdf/2411.15594v6

LLM-as-a-Judge: Response Empathicity Case

Synthetic Emotion-Aware Speech Instruction dataset

1. Generate plausible emotion label for text sample with LLM
2. Synthesize Instruction with Emotional TTS
3. Assess AudioLLM out with Empathy-targeted prompt



https://arxiv.org/pdf/2406.03872v1

LLM-as-a-Judge: Response Empathicity Case

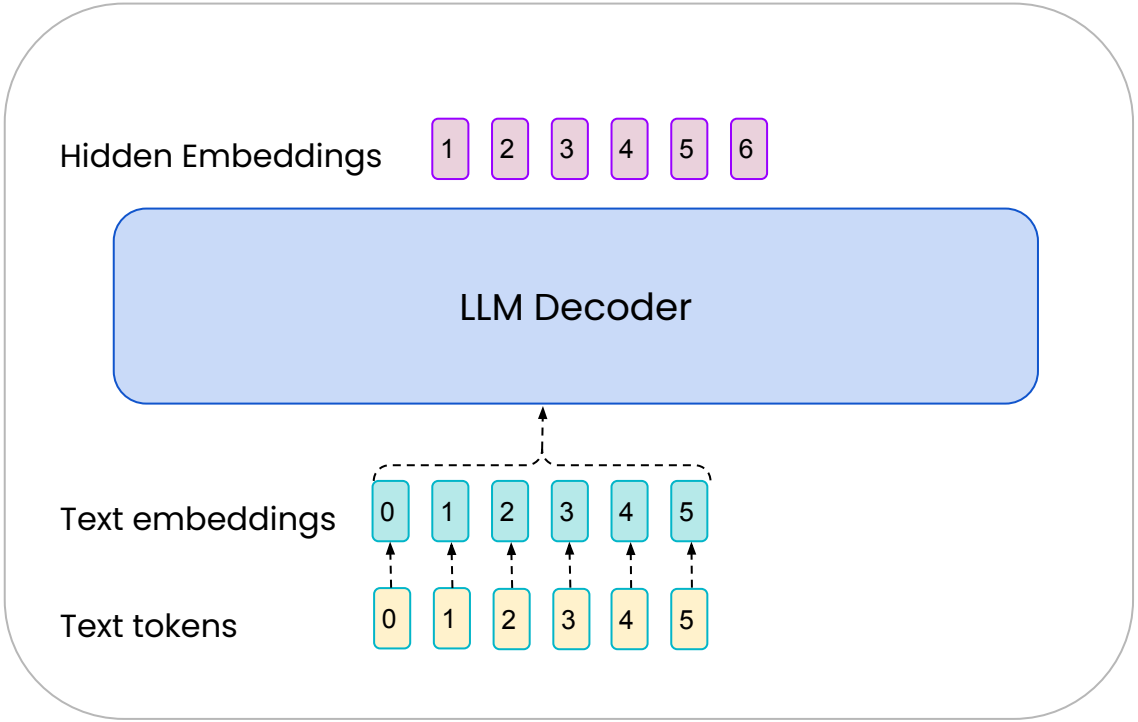
Synthetic Emotion-Aware Speech Instruction dataset

1. Generate plausible emotion label for text sample with LLM
2. Synthesize Instruction with Emotional TTS
3. Assess AudioLLM out with Empathy-targeted prompt

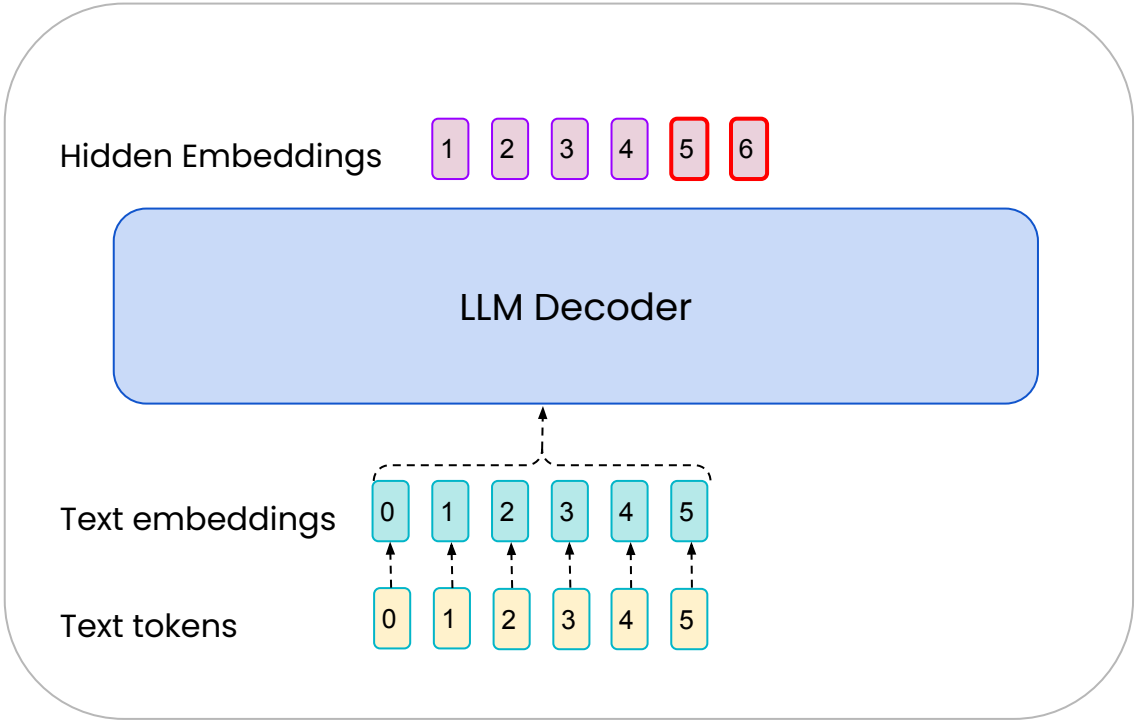
Method	SER	Empathetic Response	
		Quality	Empathy
Text+LLM	40.0	8.9	7.4
Whisper+LLM	40.1	8.9	7.4
BLSP	36.8	8.6	7.1
BLSP-SER	80.3	1.9	2.1
BLSP-Emo	83.8	8.8	7.7
HuBERT+Whisper+LLM	76.3	8.9	7.6
wav2vec2+Whisper+LLM	83.3	9.0	7.7
WavLM+Whisper+LLM	80.8	8.9	7.8
SALMONN-7B	43.8	2.4	1.9

https://arxiv.org/pdf/2406.03872v1

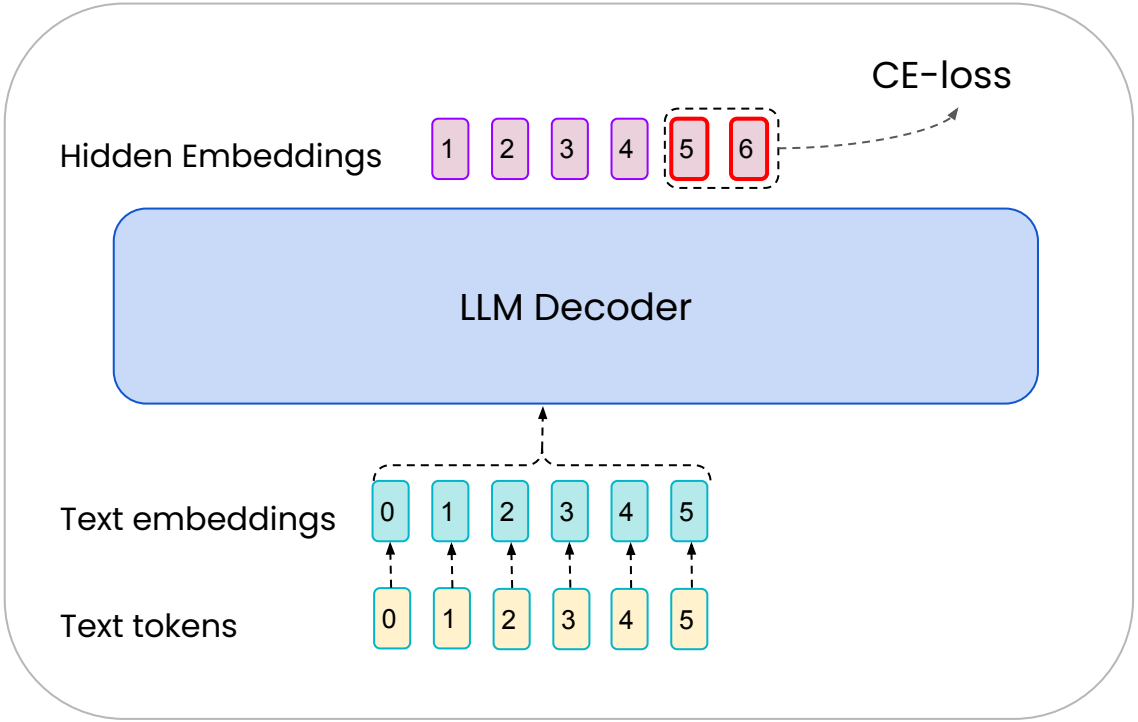
Audio-Conditioned LLM: Architecture



Audio-Conditioned LLM: Architecture



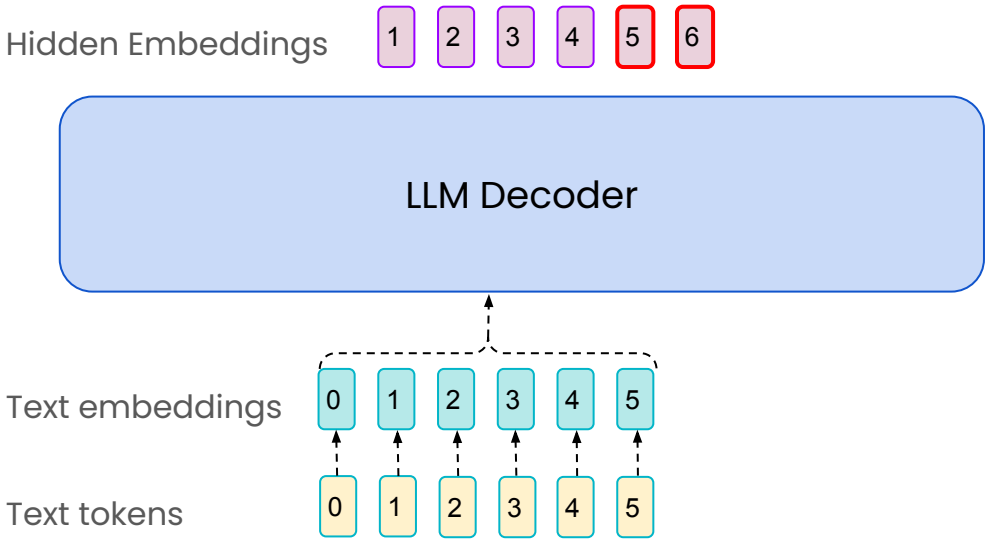
Audio-Conditioned LLM: Architecture



Audio-Conditioned LLM: Architecture

Sample

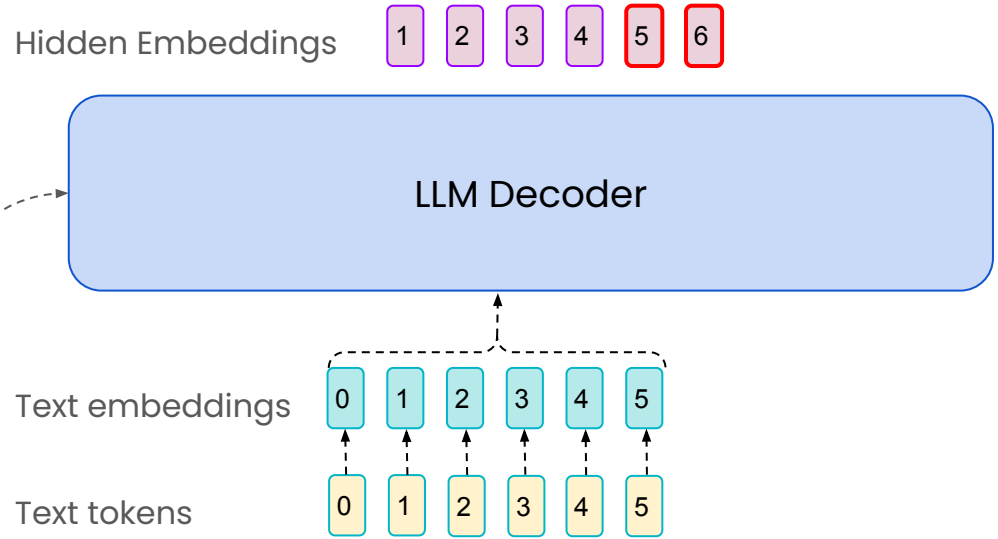
- What does this record say?
- There's a lecture on AudioLLM on the record.
- <audio_file>



Audio-Conditioned LLM: Architecture

Sample

- What does this record say?
- There's a lecture on AudioLLM on the record.
- <audio_file>



Audio-Conditioned LLM: Architecture

How to encode audio?

- Discrete / Continuous
- Semantic / Acoustic (debatable)

How to pass audio representations?

- Vocab Extension with discrete tokens
- Continuous Audio Embeddings in Prompt
- Continuous Audio Embeddings, Cross-Attention

Audio Encoding Classification

Semantic representations

Acoustic representations

Audio Encoding Classification

Semantic representations

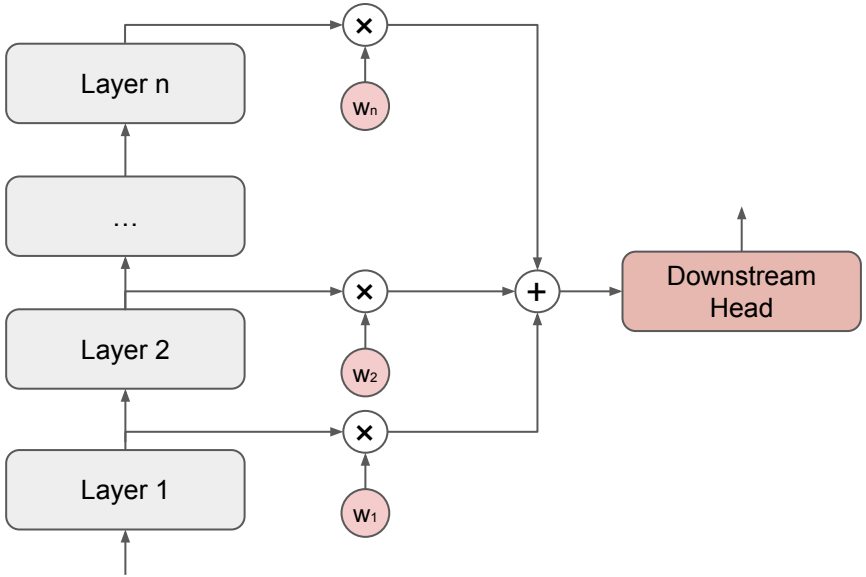
- Obtained from encoders, trained on semantic task - ASR / LM-based SSL
- Good for semantic tasks
- Support more aggressive downsampling

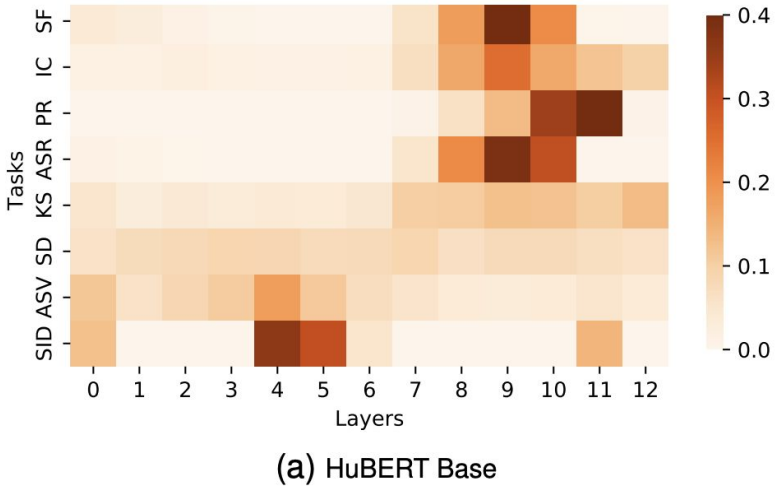
Acoustic representations

- Obtained from encoders, trained on recounstruction task - autoencoding
- Good for audio generation task
- Usually require complicated logic & higher rate

SUPERB benchmark

- SUPERB: **S**peech processing **U**niversal **PER**formance **B**enchmark
- superbbenchmark.org





(a) HuBERT Base

Audio-Conditioned LLM: Architecture

Sample

- What does this record say?
- There's a lecture on AudioLLM on the record.
-  <audio_file>

3 Main Approaches

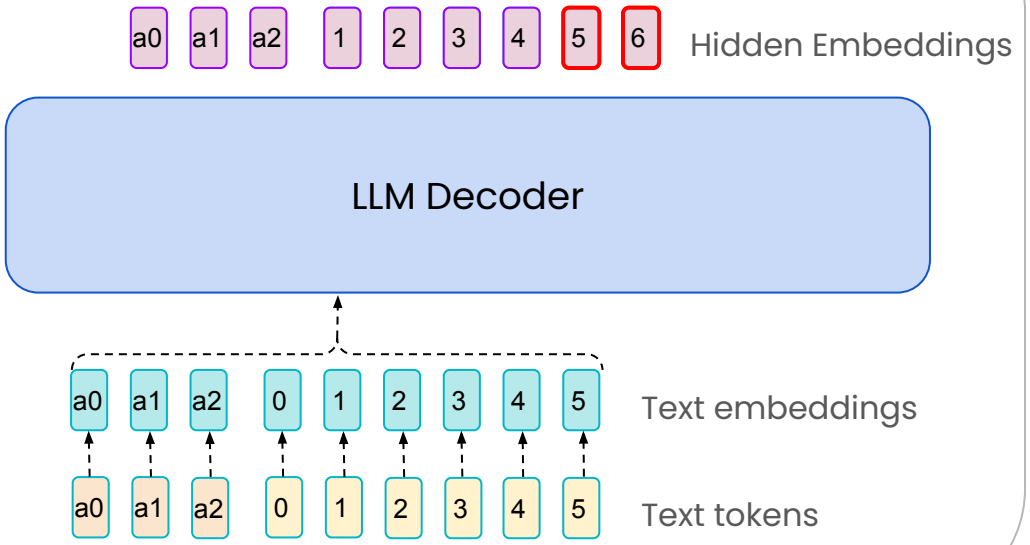
- Audio Tokens
- Continuous Audio Embeddings, Cross-Attention
- Continuous Audio Embeddings in Prompt

All 3 requires some Audio Encoding module

Audio-Conditioned LLM: Architecture

Approach 1: Audio Tokens

- Generation ability
- Lossy



AudioLM

- Semantic & Acoustic tokens
- Speech / Music Continuation Task
- 300M Params, Trained from scratch

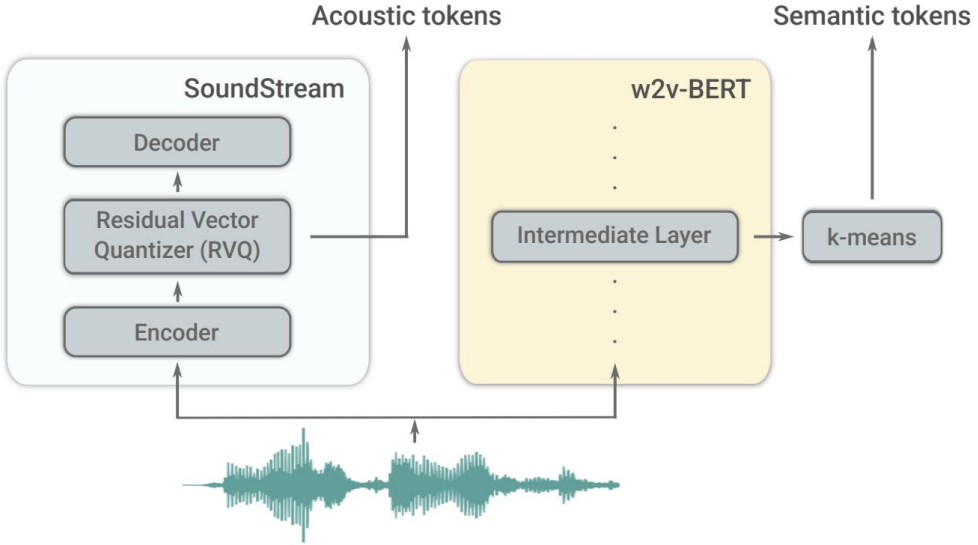
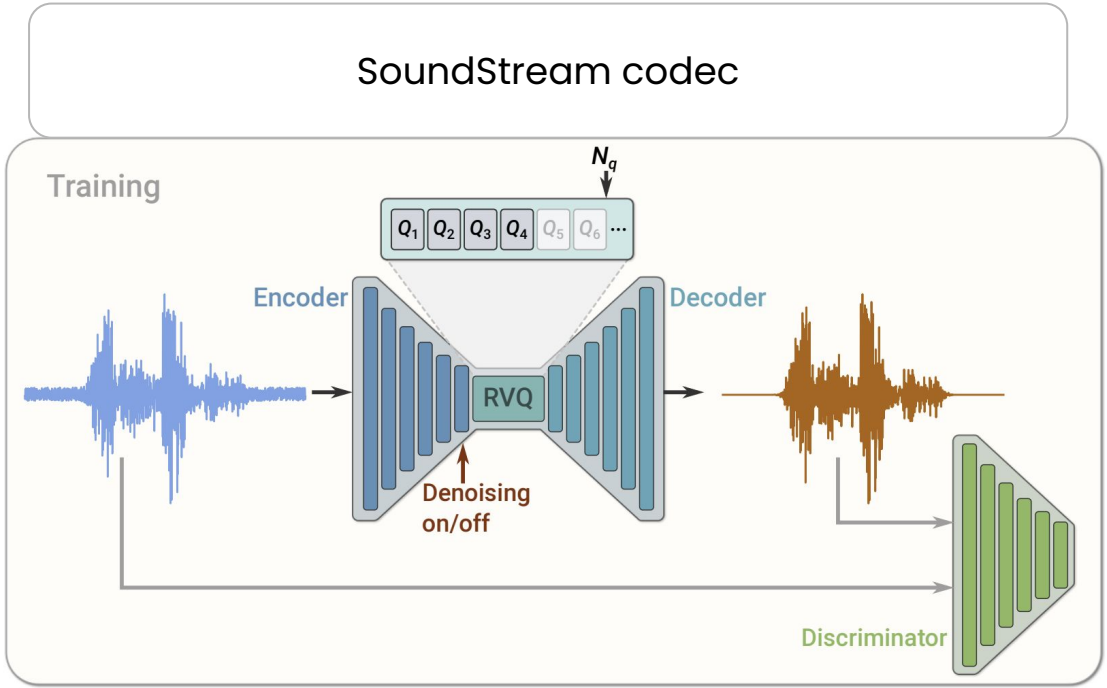


Fig. 1. Overview of the tokenizers used in AudioLM. The acoustic tokens are produced by SoundStream [16] and enable high-quality audio synthesis. The semantic tokens are derived from representations produced by an intermediate layer of w2v-BERT [17] and enable long-term structural coherence.

https://arxiv.org/pdf/2209.03143

AudioLM

- Semantic & Acoustic tokens
- Speech / Music Continuation Task
- 300M Params, Trained from scratch



https://arxiv.org/abs/2107.03312

AudioLM

TABLE I

COMPARISON OF TOKEN TYPES IN TERMS OF PHONETIC DISCRIMINABILITY WITHIN AND ACROSS SPEAKERS (LOWER IS BETTER) AND RECONSTRUCTION QUALITY (HIGHER IS BETTER). PHONETIC DISCRIMINABILITY IS MEASURED BY ABX, WHILE RECONSTRUCTION QUALITY IS REPORTED IN VISQOL UNITS.

Tokenization	Bitrate	Phonetic discriminability within/across (↓)	Reconstruction quality (↑)
Semantic (w2v-BERT)	250 bps	6.7 / 7.6	1.1
	6000 bps	5.6 / 6.2	1.4
Acoustic (SoundStream)	2000 bps	22.4 / 28.7	3.3
	6000 bps	17.8 / 26.6	3.9

- Semantic & Acoustic tokens
- Speech / Music Continuation Task
- 300M Params, Trained from scratch

https://arxiv.org/pdf/2209.03143

Discrete vs Continuous tokens

- Discrete tokens usually obtained from continuous embeddings
- We can compare continuous and corresponding discrete form

https://arxiv.org/pdf/2411.08742v1

Discrete vs Continuous tokens

- Discrete tokens usually obtained from continuous embeddings
- We can compare continuous and corresponding discrete form

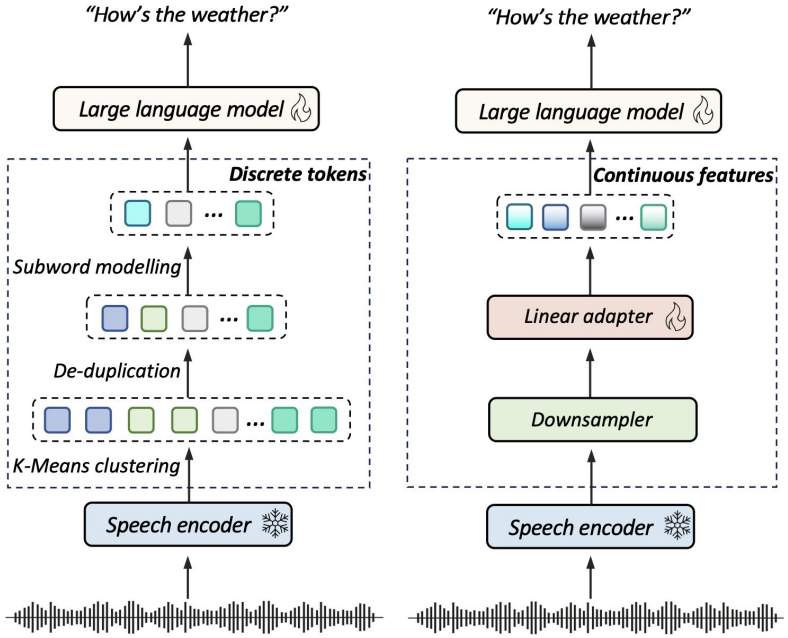


Fig. 1. Architectures of two approaches for integrating speech into Large Language Models (LLMs): **discrete token-based encoding** versus **continuous feature processing**.

https://arxiv.org/pdf/2411.08742v1

Discrete vs Continuous tokens

TABLE I
EFFECT OF DIFFERENT SETTINGS ON ASR WITH LIBRISPEECH 960H.

SSL model	*Manipulation*		WER ↓	
	K-Means	BPE size	test-clean	test-other
Hubert-Large	k=1000	-	7.48	12.82
	k=2000	-	5.02	10.55
	k=3000	-	5.01	10.51
	k=2000	4000	4.99	10.78
	k=2000	6000	**4.56**	**9.79**
	k=2000	8000	5.04	11.20
WavLM-large	k=1000	-	5.33	10.54
	k=2000	-	5.04	10.11
	k=3000	-	5.03	10.34
	k=2000	4000	4.88	10.62
	k=2000	6000	4.72	**10.45**
	k=2000	8000	**4.62**	10.82

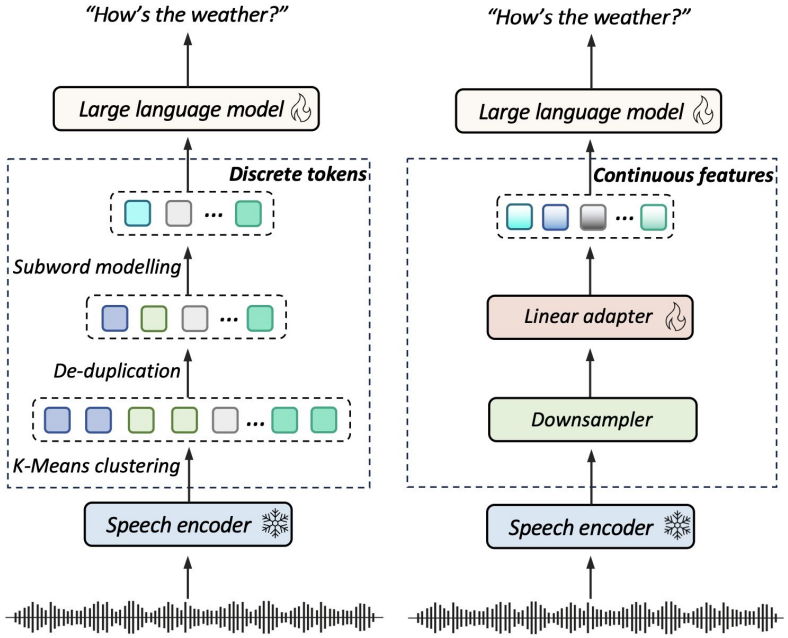


Fig. 1. Architectures of two approaches for integrating speech into Large Language Models (LLMs): **discrete token-based encoding** versus **continuous feature processing**.

https://arxiv.org/pdf/2411.08742v1

Discrete vs Continuous tokens

TABLE II
COMPARISON OF DISCRETE AND CONTINUOUS SPEECH TOKENS ON VARIOUS TASKS BASED ON QWEN-1.5-0.5B MODEL. DISCRETE TOKENS USE K-MEANS (2000 CLUSTERS) WITH BPE SIZE 6000 FOR ALL TASKS.

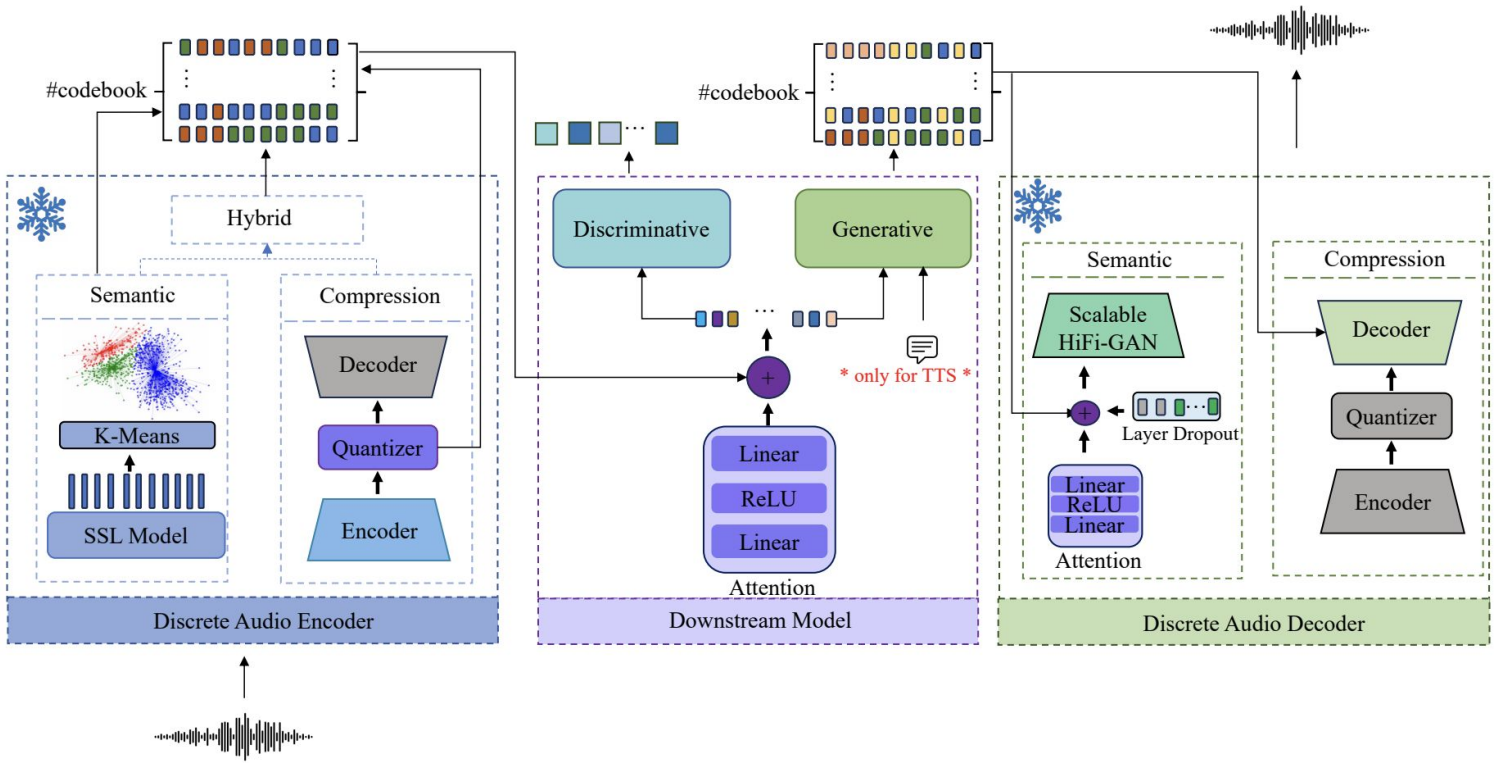
SSL model	*Token type*	ASR (WER↓)		PR (PER↓)	ST (BLEU↑)	KS (ACC↑)	IC (ACC↑)	ER (ACC↑)
		LibriSpeech (test-clean|other)	Gigaspeech-M (test)	Libri-100 (test-clean)	GigaST (En-Zh|En-De)	Speech Commands (test|val)	SLURP (test)	IEMOCAP (test)
HuBERT-Large	Discrete	4.56 / 9.79	19.40	9.69	22.75 / 20.14	93.70 / 93.85	57.04	38.65
WavLM-Large		4.72 / 10.45	16.34	**9.64**	24.62 / 21.22	92.87 / 92.45	59.96	37.98
HuBERT-Large	Continuous	4.91 / 6.43	17.45	12.84	26.63 / 25.42	95.38 / 95.70	76.84	56.72
WavLM-Large		**2.92 / 4.61**	**13.96**	12.62	**29.44 / 28.12**	**97.76 / 97.36**	**81.35**	**59.45**

TABLE III
PERFORMANCE COMPARISON OF DISCRETE SPEECH TOKENS USING LLAMA 3.1-8B. SAME SETTINGS AS IN QWEN 1.5-0.5B.

SSL model	*Token type*	ASR (WER↓)		PR (PER↓)	ST (BLEU↑)	KS (ACC↑)	IC (ACC↑)	ER (ACC↑)
		LibriSpeech (test-clean|other)	Gigaspeech-M (test)	Libri-100 (test-clean)	GigaST (En-Zh|En-De)	Speech Commands (test| val)	SLURP (test)	IEMOCAP (test)
HuBERT-Large	Discrete	**2.56 / 6.49**	**12.859**	7.85	26.32 / 25.24	96.75 / 96.69	63.44	**39.84**
WavLM-Large		2.96 / 7.48	13.35	**7.02**	**28.62 / 26.87**	**97.92 / 98.17**	**66.96**	36.12

https://arxiv.org/pdf/2411.08742v1

Discrete Audio and Speech Benchmark



https://arxiv.org/pdf/2406.14294

Discrete Audio and Speech Benchmark

Table 2: Benchmarking results for discriminative tasks.

Models/Tasks	ASR-En		ASR-multiling		ER	IC	KS	SI	SV
	WER ↓		WER ↓		ACC ↑	ACC ↑	ACC ↑	ACC ↑	EER ↓
	Clean	Other	Welsh	Basque					
Low Bitrate									
Discrete Hubert	**8.99**	**21.14**	**58.50**	**26.83**	57.20	68.70	90.54	0.90	24.99
Discrete WavLM	11.72	27.56	60.37	28.63	**59.80**	73.40	**97.94**	0.70	26.02
Discrete Wav2Vec2	12.14	28.65	66.30	32.25	57.80	**74.10**	96.16	0.40	33.53
EnCodec	52.37	77.04	92.01	58.20	44.70	31.50	86.00	**58.30**	**17.40**
DAC	63.96	83.61	94.86	66.29	49.20	22.10	81.00	45.10	20.62
SpeechTokenizer	19.77	43.12	76.67	47.92	49.10	57.90	95.09	47.40	20.41
Medium Bitrate									
Discrete Hubert	**7.91**	**18.95**	54.77	23.63	**62.10**	70.50	94.69	67.40	15.71
Discrete WavLM	8.52	20.35	**54.22**	**22.06**	57.60	**78.00**	**98.09**	80.80	8.00
Discrete Wav2Vec2	8.76	21.32	60.39	26.64	59.10	75.10	96.64	65.47	17.64
EnCodec	46.80	74.24	91.23	47.95	51.30	31.40	88.70	**91.90**	**7.81**
DAC	59.54	81.48	97.43	56.16	45.80	18.90	76.60	83.80	11.78
SpeechTokenizer	18.32	41.21	75.17	38.94	52.10	57.80	94.86	91.40	7.88
High Bitrate									
EnCodec	**45.18**	**72.56**	**93.40**	**87.65**	46.40	**19.60**	**83.60**	**92.81**	**7.18**
DAC	99.53	99.38	99.40	99.68	**46.00**	15.70	75.20	85.61	10.89
Continuous Baseline									
SSL	3.370	7.04	41.77	14.32	63.10	86.10	99.00	99.70	2.10

https://arxiv.org/pdf/2406.14294

Discrete Audio and Speech Benchmark

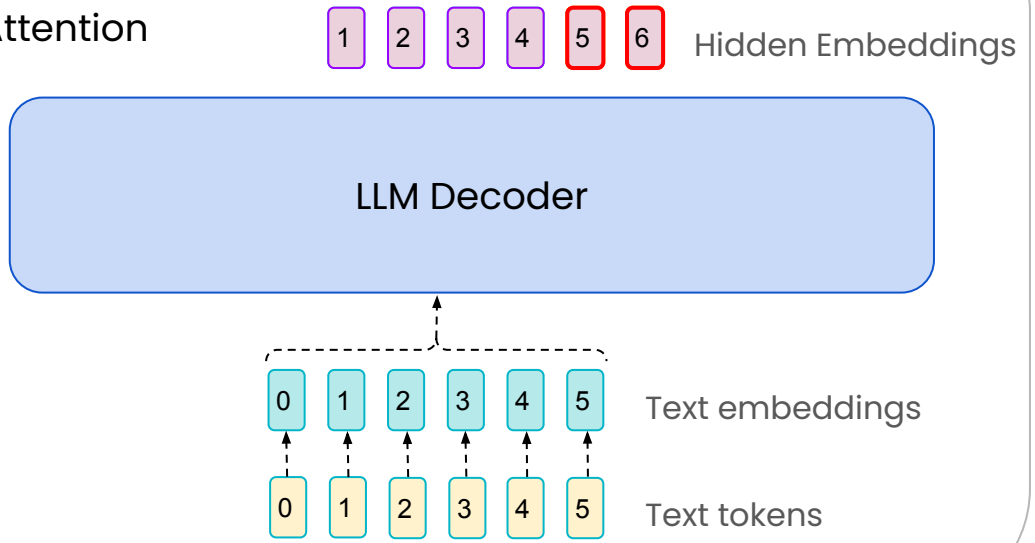
Table 3: Benchmarking results for generative tasks. N.C. indicates “Not Converged".

Models/Tasks	SE			SS			TTS	
	DNSMOS ↑	dWER ↓	SpkSim ↑	DNSMOS ↑	dWER ↓	SpkSim ↑	UTMOS ↑	dWER ↓
Low Bitrate								
Discrete HuBERT	3.33	**15.47**	0.824	3.52	80.86	0.840	3.24	**2.55**
Discrete WavLM	3.26	16.52	0.830	3.43	**62.34**	0.847	**3.84**	3.01
Discrete Wav2Vec2	**3.55**	18.86	0.779	**3.75**	96.70	0.787	3.32	3.45
EnCodec	3.15	34.35	0.852	3.11	83.55	**0.877**	1.46	8.85
DAC	3.30	57.41	0.853	3.01	102.00	0.854	1.97	10.68
SpeechTokenizer	3.18	30.13	**0.858**	3.13	85.25	0.874	2.51	3.69
Medium Bitrate								
Discrete HuBERT	3.48	12.62	0.875	3.70	66.29	0.891	3.80	3.40
Discrete WavLM	3.48	**10.18**	0.889	3.68	**34.03**	0.912	**3.82**	**2.45**
Discrete Wav2Vec2	**3.54**	17.60	0.858	**3.75**	78.42	0.866	3.68	2.89
EnCodec	3.10	19.07	0.885	3.09	48.57	0.906	1.50	94.6
DAC	3.49	31.14	**0.906**	3.26	55.43	**0.924**	1.71	71.26
SpeechTokenizer	3.49	23.44	0.876	3.42	60.75	0.906	1.96	53.26
High Bitrate								
EnCodec	2.87	68.22	0.814	**2.95**	**97.73**	**0.839**	N.C	N.C
DAC	**2.95**	**46.07**	**0.860**	2.53	208	0.784	N.C	N.C
Continuous Baseline								
SSL	3.49	4.92	0.928	3.68	9.97	0.939	3.71	2.94

https://arxiv.org/pdf/2406.14294

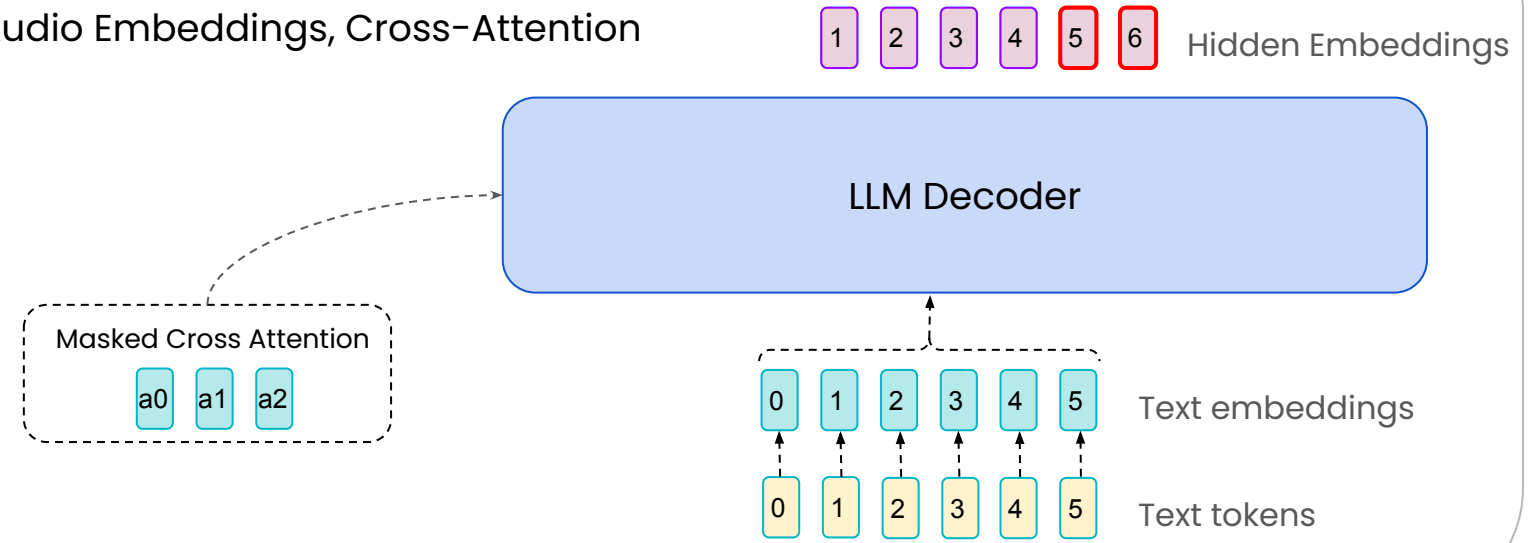
Audio-Conditioned LLM: Architecture

Approach 2:
Continuous Audio Embeddings, Cross-Attention

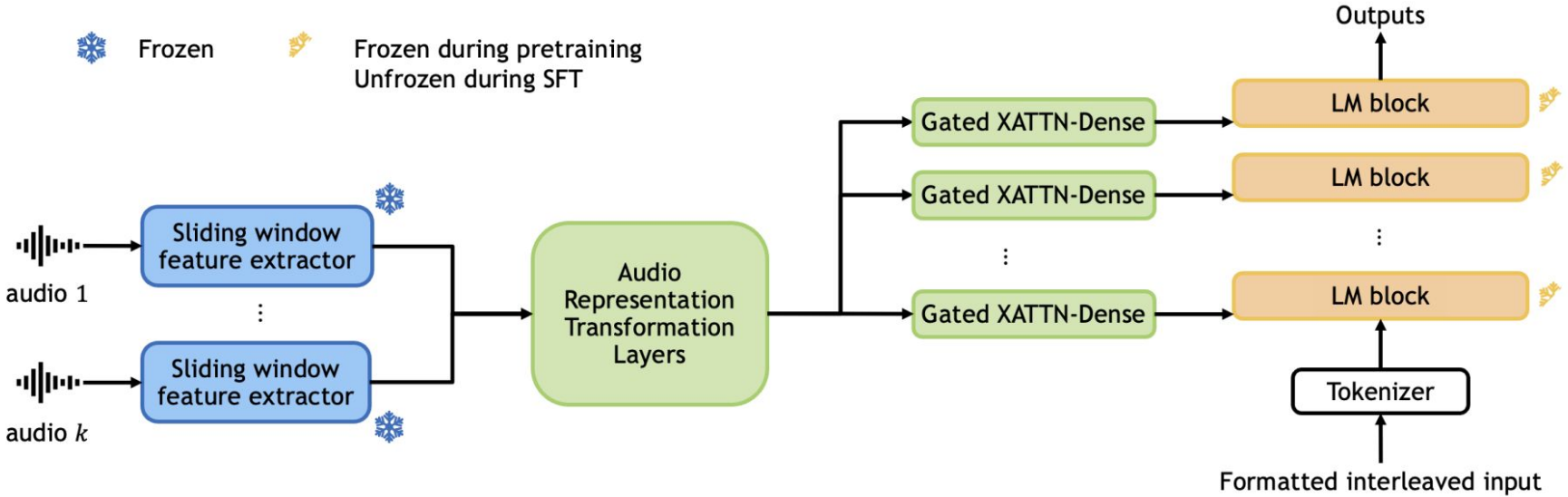


Audio-Conditioned LLM: Architecture

Approach 2:
Continuous Audio Embeddings, Cross-Attention

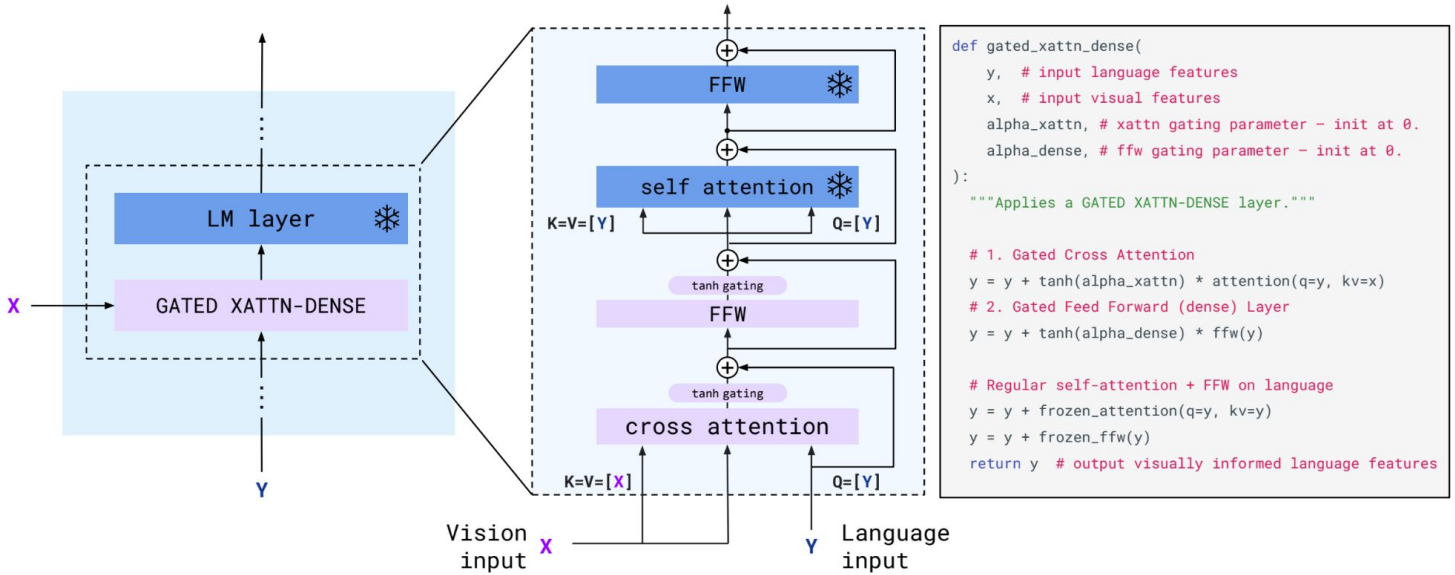


Audio-Conditioned LLM: Audio-FLamingo v1,2



https://arxiv.org/pdf/2503.03983

Audio-Conditioned LLM: Audio-FLamingo v1,2




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Figure 4: **GATED XATTN-DENSE layers.** To condition the LM on visual inputs, we insert new cross-attention layers between existing pretrained and frozen LM layers. The keys and values in these layers are obtained from the vision features while the queries are derived from the language inputs. They are followed by dense feed-forward layers. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

https://arxiv.org/pdf/2204.14198

Audio-Conditioned LLM: Audio-FLamingo v1,2

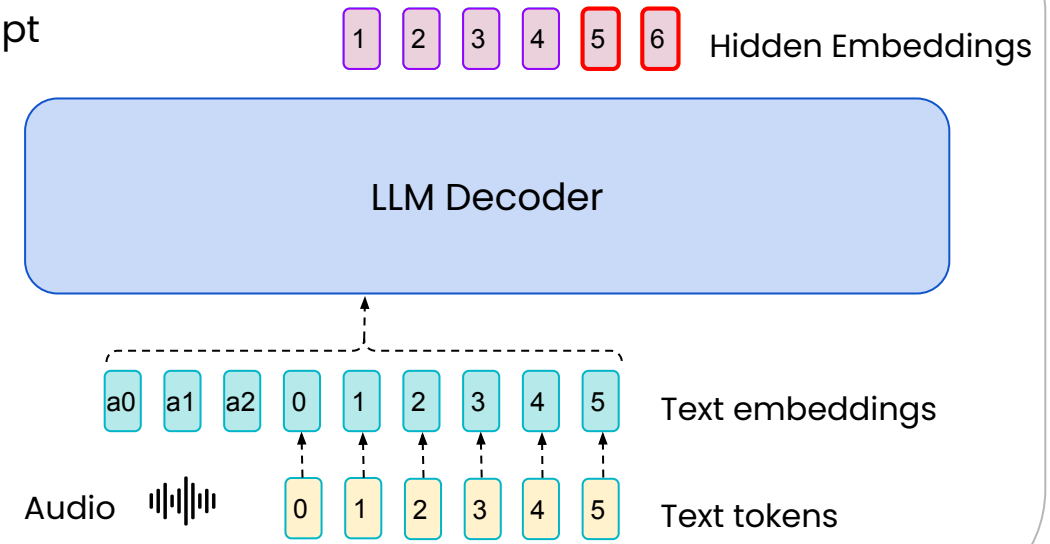
Architecture	Backbones training	Avg. score
Fully autoreg. no Perceiver	Frozen	51.8
Fully autoreg.	Frozen	60.3
Cross-attention	Frozen	66.7
Cross-attention	LoRA	67.3
Fully autoreg.	LoRA	69.5

Table 3: Ablation for the architecture and method of training.

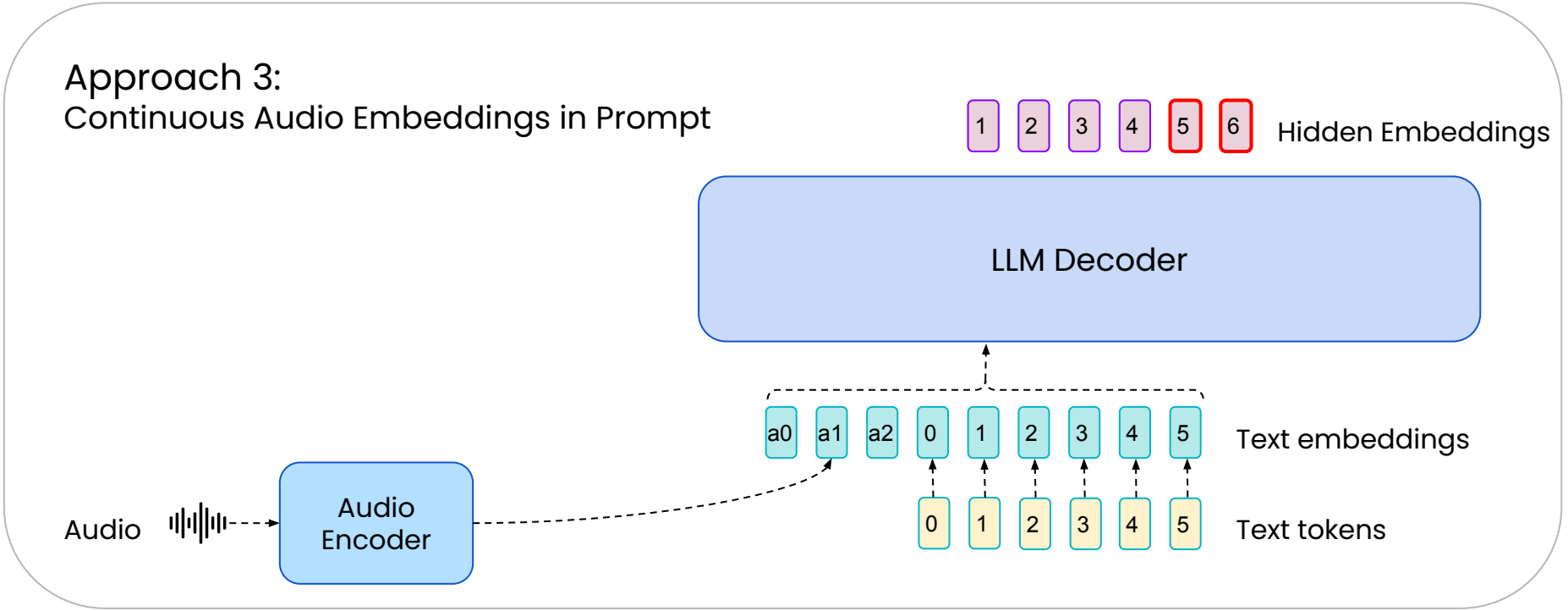
https://arxiv.org/pdf/2405.02246

Audio-Conditioned LLM: Main Architecture

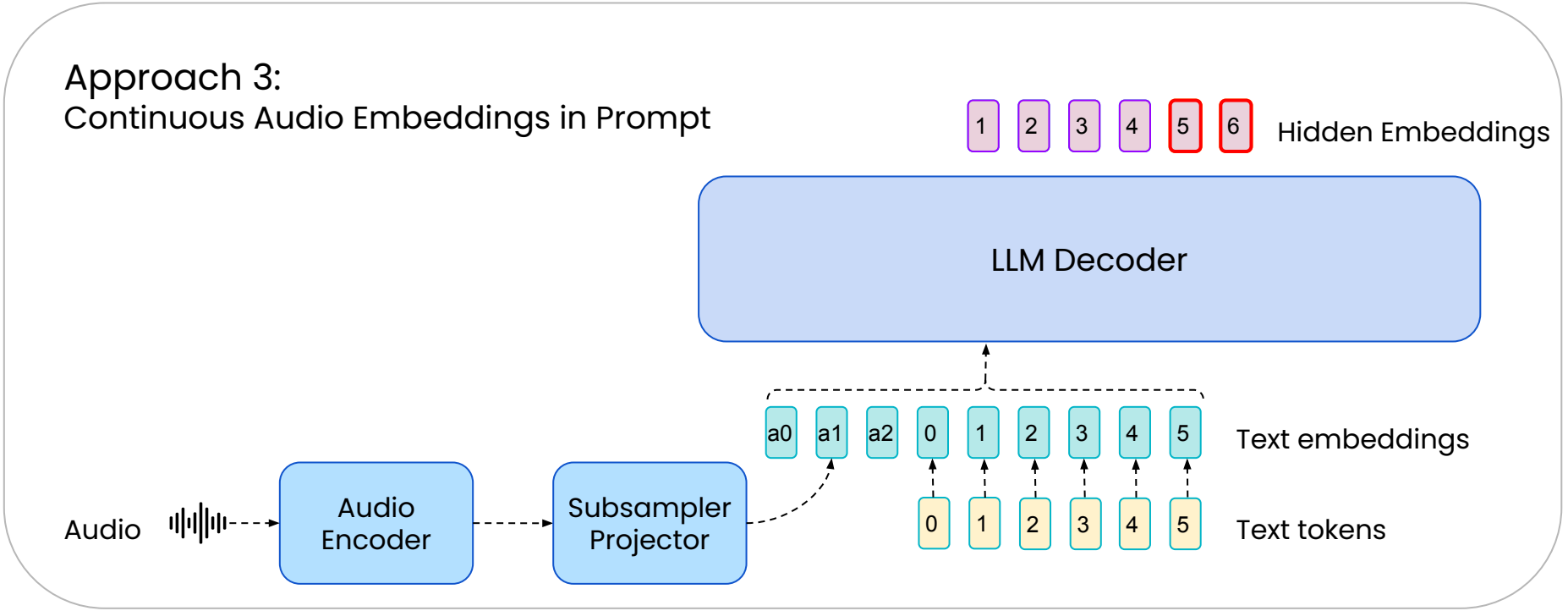
Approach 3:
Continuous Audio Embeddings in Prompt



Audio-Conditioned LLM: Main Architecture



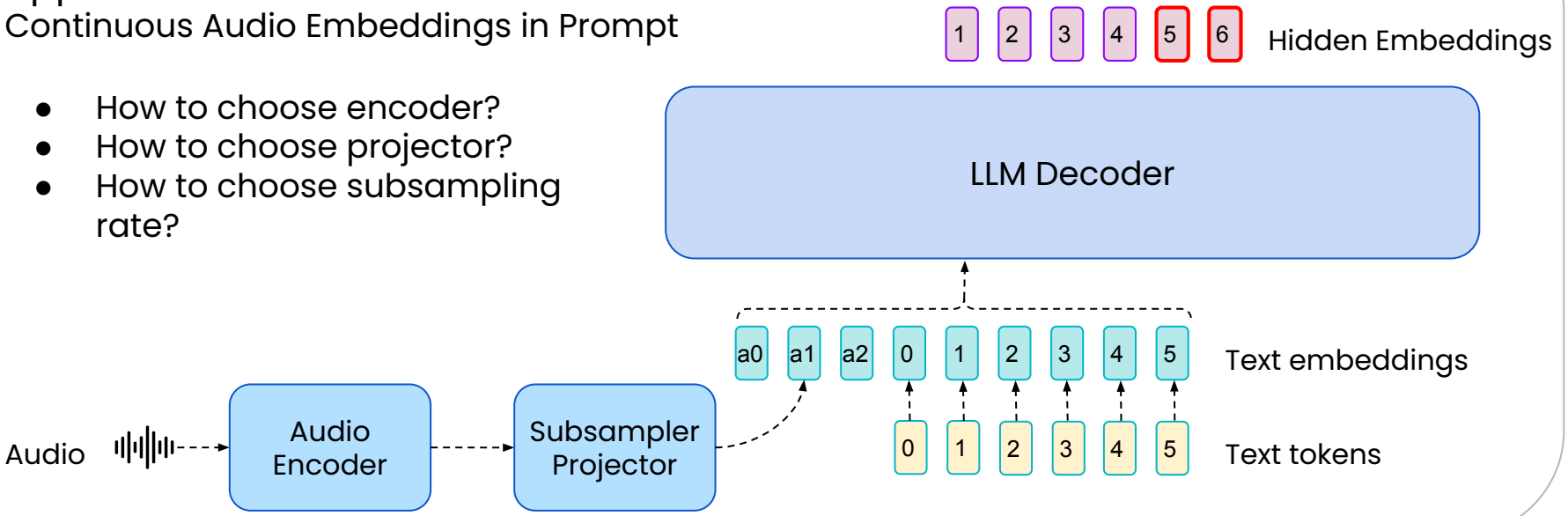
Audio-Conditioned LLM: Main Architecture



Audio-Conditioned LLM: Main Architecture

Approach 3:
Continuous Audio Embeddings in Prompt

- How to choose encoder?
- How to choose projector?
- How to choose subsampling rate?



Audio Encoding Classification

Semantic representations

- Obtained from encoders, trained on semantic task - ASR / LM-based SSL
- Good for semantic tasks
- Support more aggressive downsampling

Acoustic representations

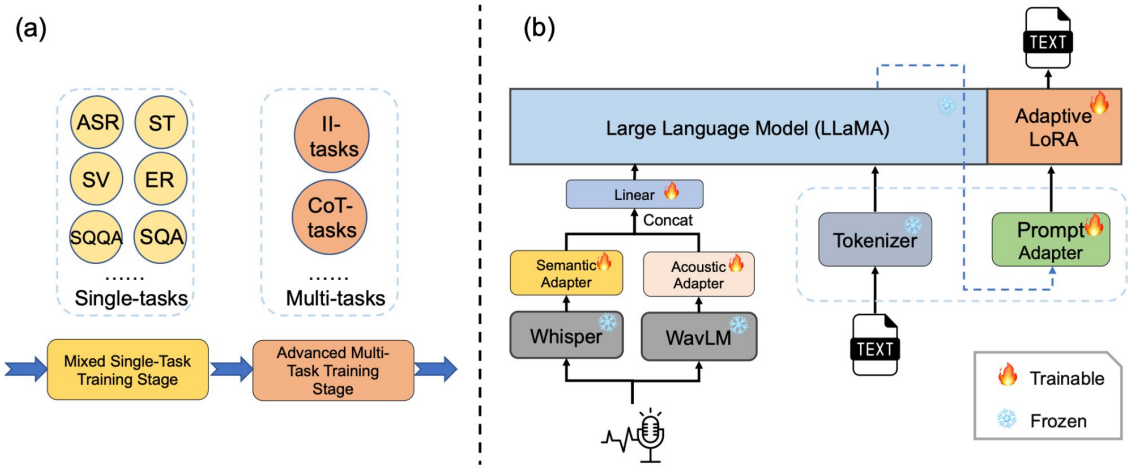
- Obtained from encoders, trained on recounstruction task - autoencoding
- Good for audio generation task
- Usually require complicated logic & higher rate

Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)

Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)



Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)

Table 9: Single-task instruction performance of models with or without WavLM encoder after the mixed training.

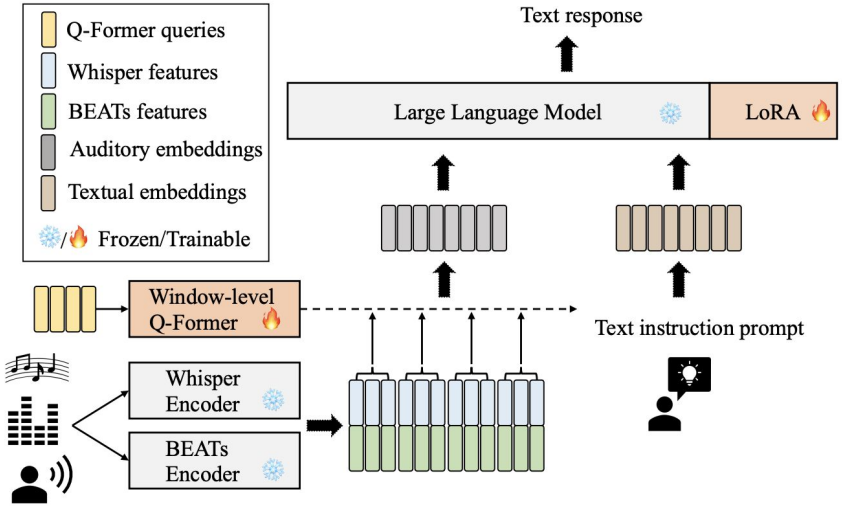
Models	ASR		ST (En2De)		SV	ER	SQQA	SQA
	test-clean	test-other	CoVoST2	MUSTC				
	WER↓		BLEU↑		Acc.↑	Acc.↑	Acc.↑	Acc.↑
WavLLM	2.0	4.8	23.9	21.9	0.91	0.72	0.55	67.30%
WavLLM w/o WavLM	2.3	5.4	23.4	21.0	0.89	0.73	0.55	68.55%

Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)

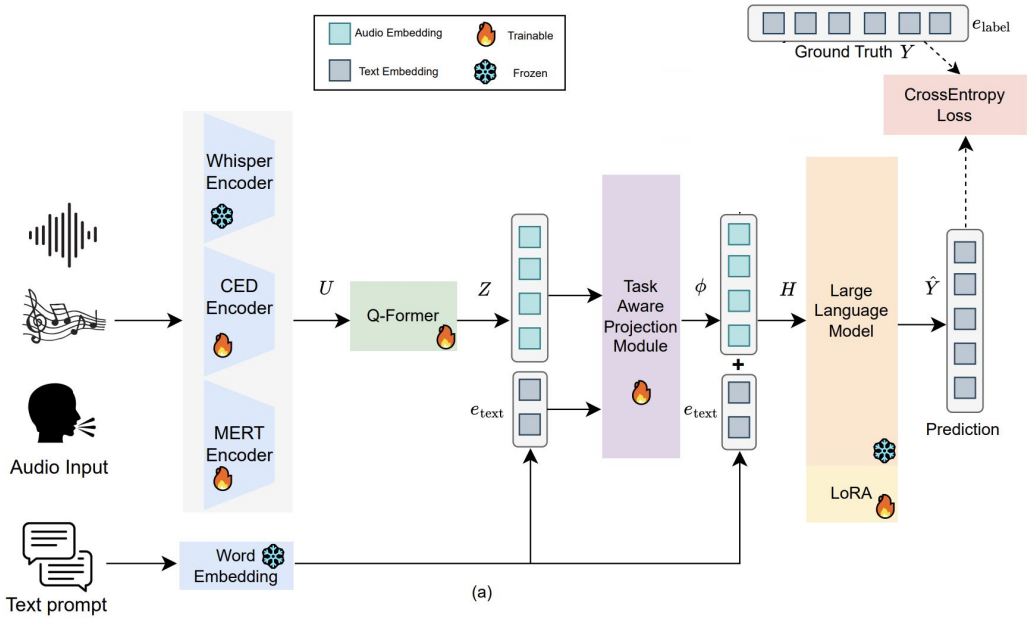
Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)



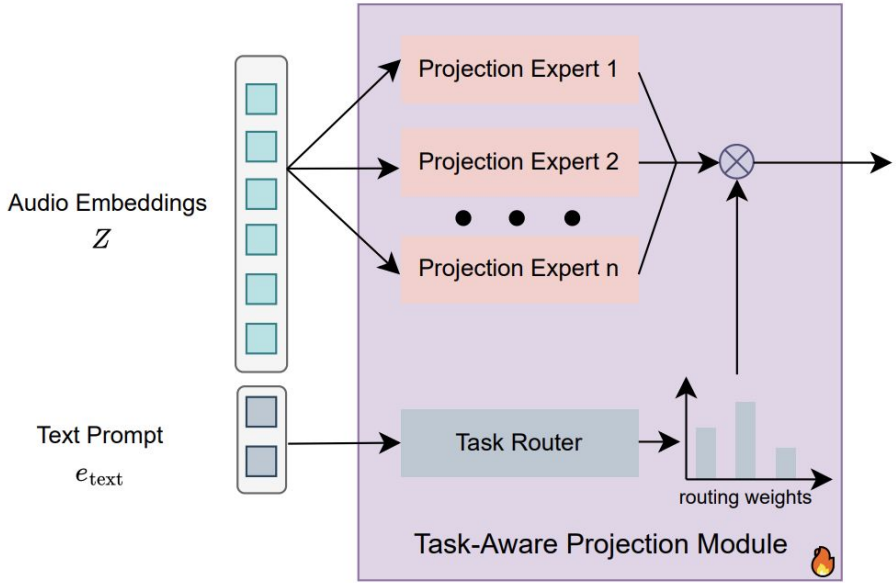
Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)



Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)



Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)

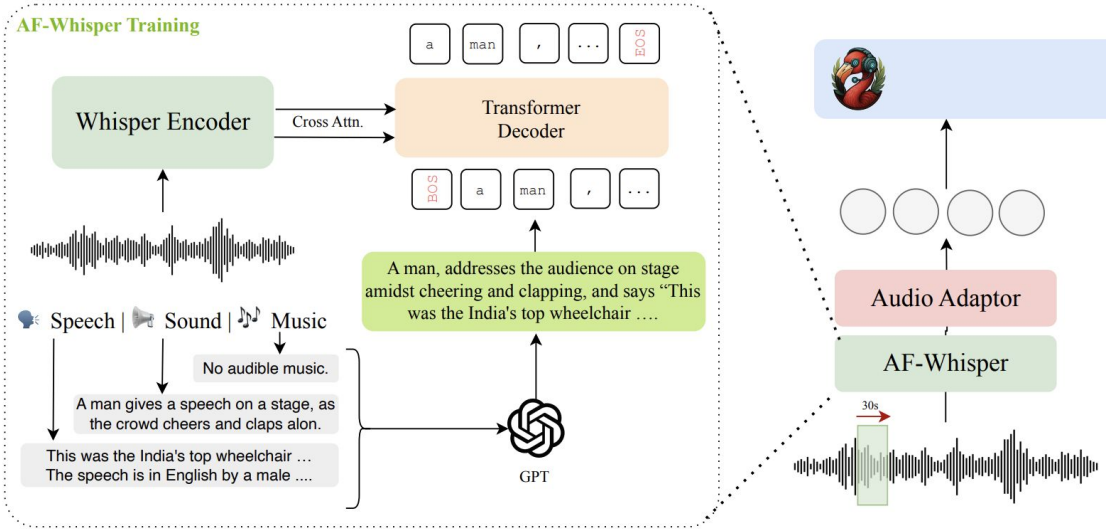
Models	ASR↓ %WER	En2Zh↑ BLEU4	AAC↑ METEOR|SPIDEr	MC↑ BLEU4|RougeL
U-SAM	(2.0, 5.0)	38.3	26.7|49.5	6.0|22.5
w/o Whisper Encoder	(39.1, 45.0)	14.7	27.2|50.1	6.8|23.8
w/o CED Encoder	(2.2, 5.1)	39.0	10.9|24.0	5.7|20.0
w/o MERT Encoder	(2.1, 5.2)	38.7	26.5|49.3	2.7|11.4
w/o TAPM	(5.1, 7.9)	31.0	28.0|51.5	4.0|16.8

Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)
- Audio-Flamingo v3: Comparison AF-Whisper vs. CLAP + Whisper

Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)
- Audio-Flamingo v3: Comparison AF-Whisper vs. CLAP + Whisper



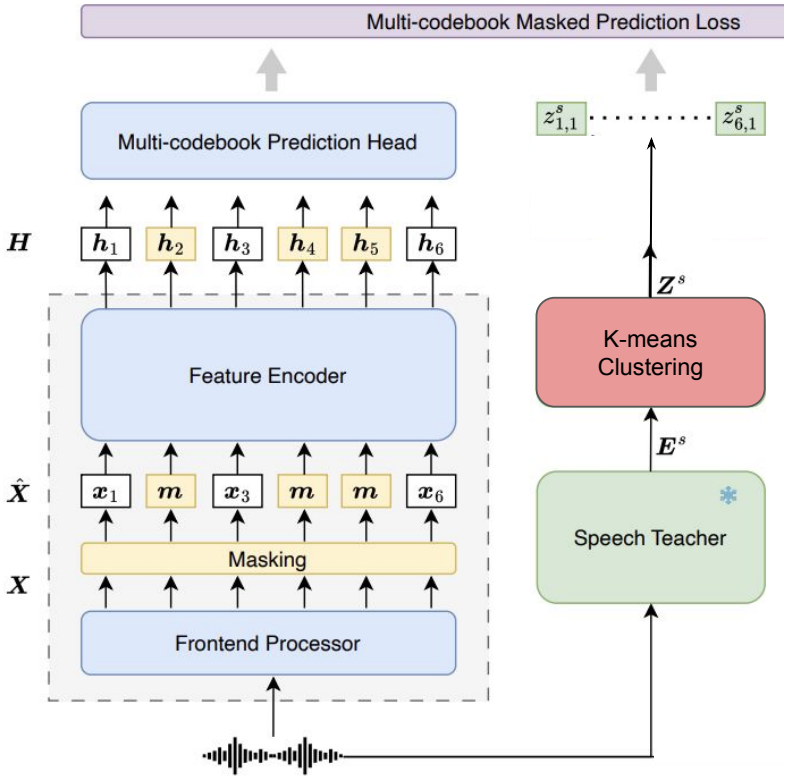
Audio-Conditioned LLM: Main Architecture

- WavLLM: Whisper (semantic) + WavLM (acoustic)
- SALMONN: Whisper (semantic) + BEATs (acoustic)
- U-SAM: Whisper (semantic) + CED (acoustic) + MERT (music)
- Audio-Flamingo v3: Comparison AF-Whisper vs. CLAP + Whisper

Table 4: Comparison of AF3 w/ 10% data, w/o AF-Whisper and w/o AudioSkills-XL.

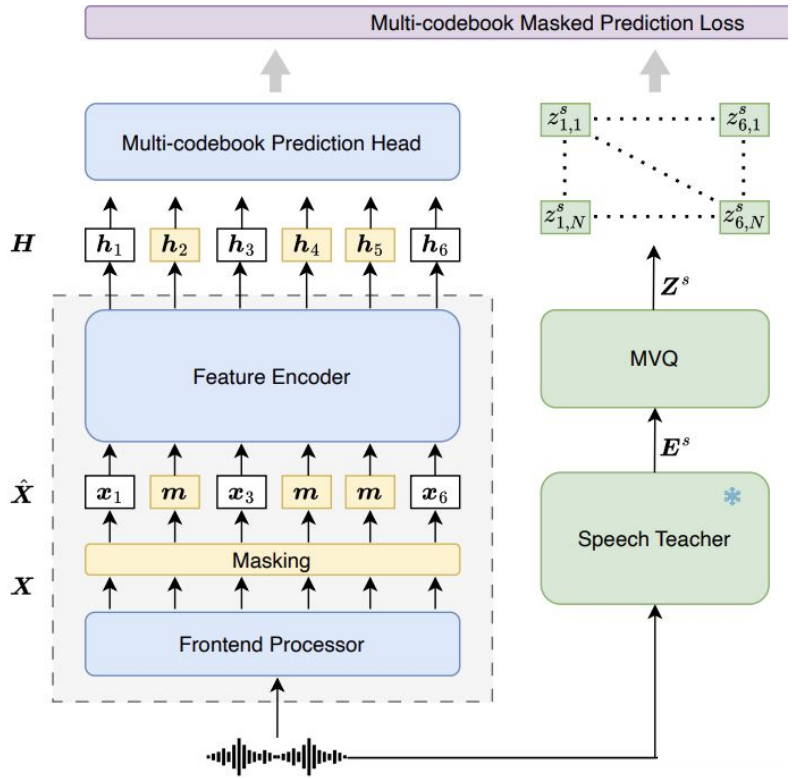
Model	**MMAU-Sound** ACC ↑	**MMAU-Music** ACC ↑	**MMAU-Speech** ACC ↑	**Librispeech-clean** WER ↓	**Librispeech-other** WER ↓
w/ 10% data	66.7	65.9	57.4	2.0	4.1
+ w/o AF-Whisper	63.7	68.3	45.2	3.7	7.2
w/o AudioSkills-XL	56.1	42.1	14.3	1.6	3.6
Audio Flamingo 3	**75.8**	**74.4**	**66.9**	**1.5**	**3.1**

Offtop: Universal SSL-Encoder



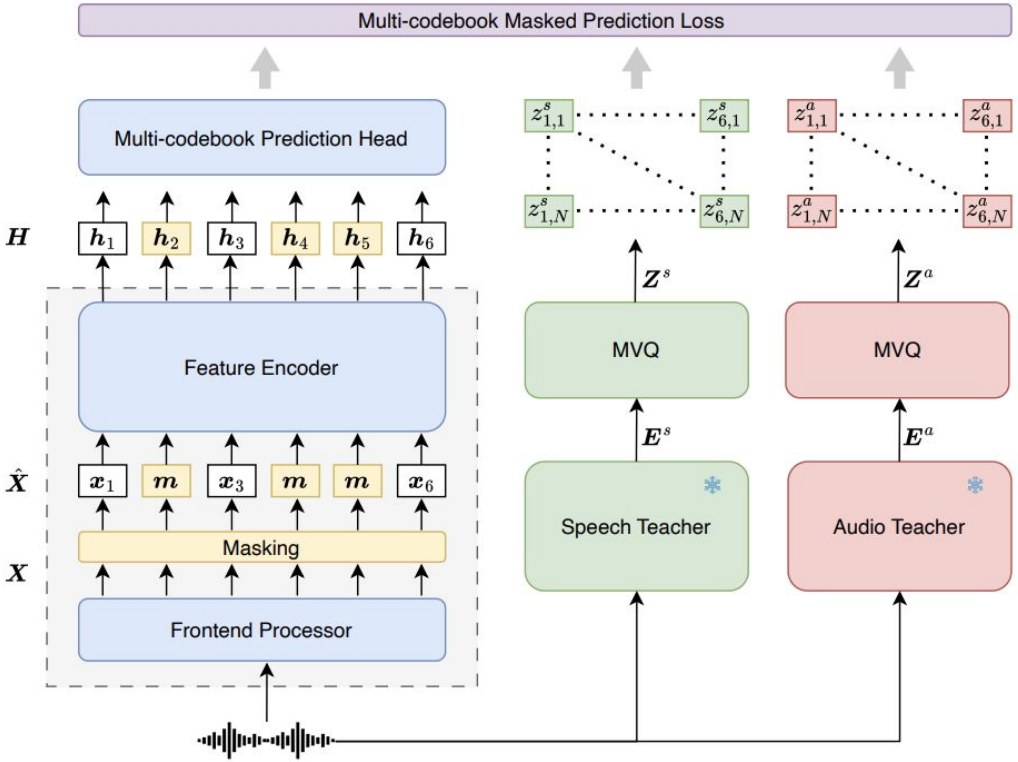
https://arxiv.org/pdf/2510.25955

Offtop: Universal SSL-Encoder



https://arxiv.org/pdf/2510.25955

Offtop: Universal SSL-Encoder



https://arxiv.org/pdf/2510.25955

Offtop: Universal SSL-Encoder

Model	# Params	Pre-train data	LS-100		LS-960		AS-20k	AS-2M
			clean	other	clean	other		
Speech SSL Models								
WavLM Base + △ (Chen et al., 2022)	95M	94k	4.0	8.4	2.9	5.4	-	-
HuBERT Large△ (Hsu et al., 2021)	317M	60k	-	-	1.8	3.9	-	-
WavLM Large△ (Chen et al., 2022)	317M	94k	3.0	6.1	1.8	3.8	-	-
Ours, SPEAR$_s$ Base	94M	84k	3.0	5.8	1.9	4.0	26.9	43.6
Ours, SPEAR$_s$ Large	327M	84k	2.6	4.7	1.7	3.3	26.4	43.9
Audio SSL Models								
BEATs (Chen et al., 2023)	90M	5k	-	-	-	-	38.9	48.6
EAT (Chen et al., 2024)	88M	5k	-	-	-	-	**40.2**	48.6
ATST Frame (Li et al., 2024a)	86M	5k	-	-	-	-	39.0	48.0
Dasheng-Base△ (Dinkel et al., 2024)	86M	272k	-	-	-	-	-	49.7
Ours, SPEAR$_a$ Base	94M	13k	11.2	23.0	-	-	39.2	49.3
Ours, SPEAR$_a$ Large	327M	13k	7.4	18.6	-	-	39.3	49.8
Speech & Audio SSL Models								
Ours, SPEAR$_{s+a}$ Base	94M	97k	3.1	6.1	1.9	4.2	39.1	48.4
Ours, SPEAR$_{s+a}$ Large	327M	97k	2.6	4.8	1.7	3.4	39.2	49.6
Ours, SPEAR$_{s+a}$ XLarge	600M	197k	**2.5**	**4.6**	**1.6**	**2.9**	39.4	**50.0**

https://arxiv.org/pdf/2510.25955

Encoders: Recap

- Different encoders perform betters on different tasks
- Depends mostly on model & data size AND pretraining task
- AF-Whisper training stage made Whisper Encoder better choice for music / audio tasks than specific encoders
- SSL-approaches that learn universal encoder through distillation

Audio-Conditoned LLM: Projectors & Subsampling

Projectors - we need to match LLM hidden dim with MLP

In some cases (frozen lm and encoder) - more complicated modules:

- QFormer
- Transofmer / Conformer
- Task-Aware Projector (U-SAM)

Subsampling - we want code audio with minimum frame rate

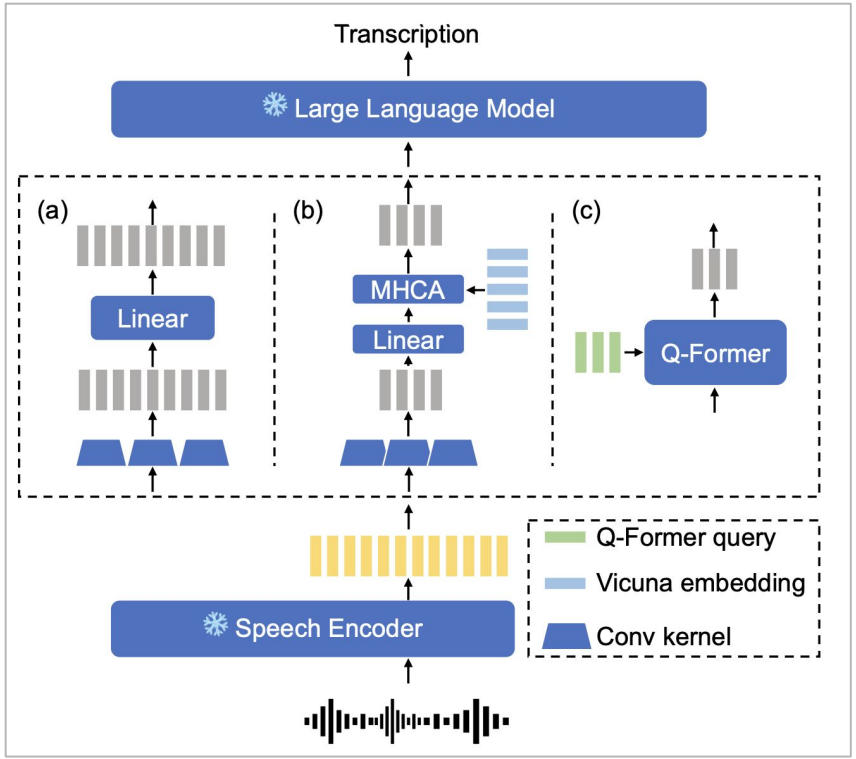
Usually for Audio understanding use from 6Hz to 50Hz embedding
Frame rate of some popular encoders:

- Whisper: 50Hz
- WavLM: 50Hz
- BEATs: 6.25Hz * 8 freq-patches
- CLAP: 6.25Hz * 4 freq-patches

Audio-Conditoned LLM: Projectors

Comparing quality with:

- Q-Former
- Linear layer
- Multi-Head Cross-Attention

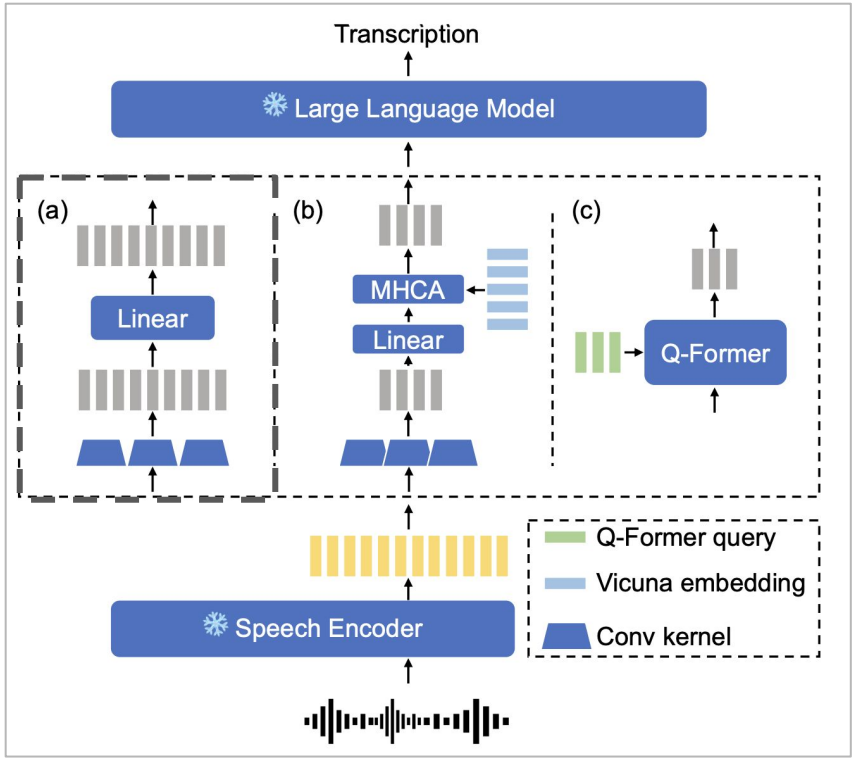


https://arxiv.org/pdf/2309.13963

Audio-Conditoned LLM: Projectors

Comparing quality with:

- Q-Former
- **Linear layer**
- Multi-Head Cross-Attention

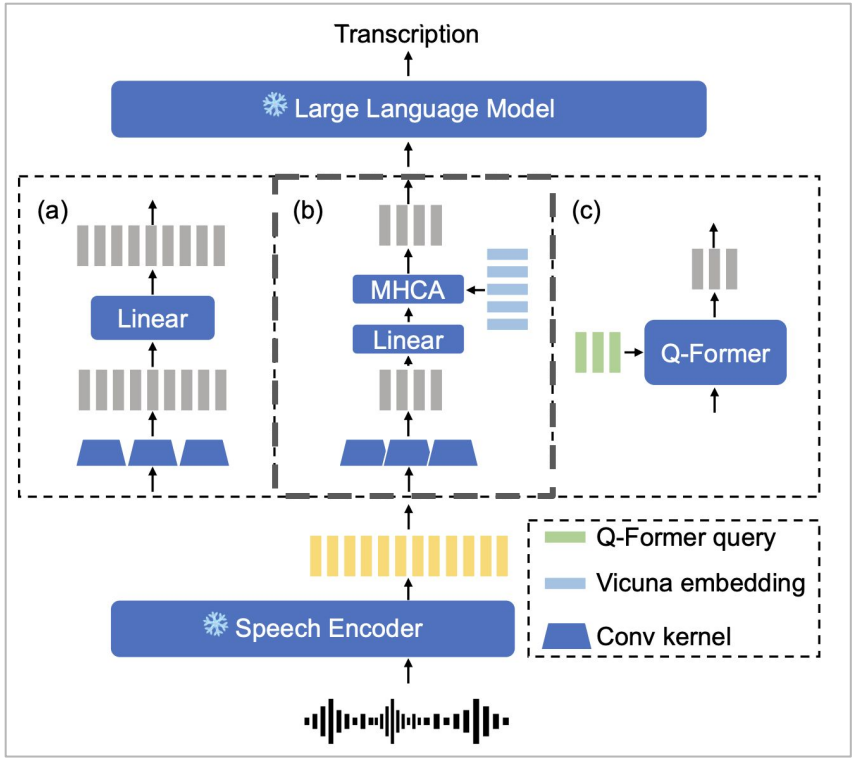


https://arxiv.org/pdf/2309.13963

Audio-Conditoned LLM: Projectors

Comparing quality with:

- Q-Former
- Linear layer
- **Multi-Head Cross-Attention**

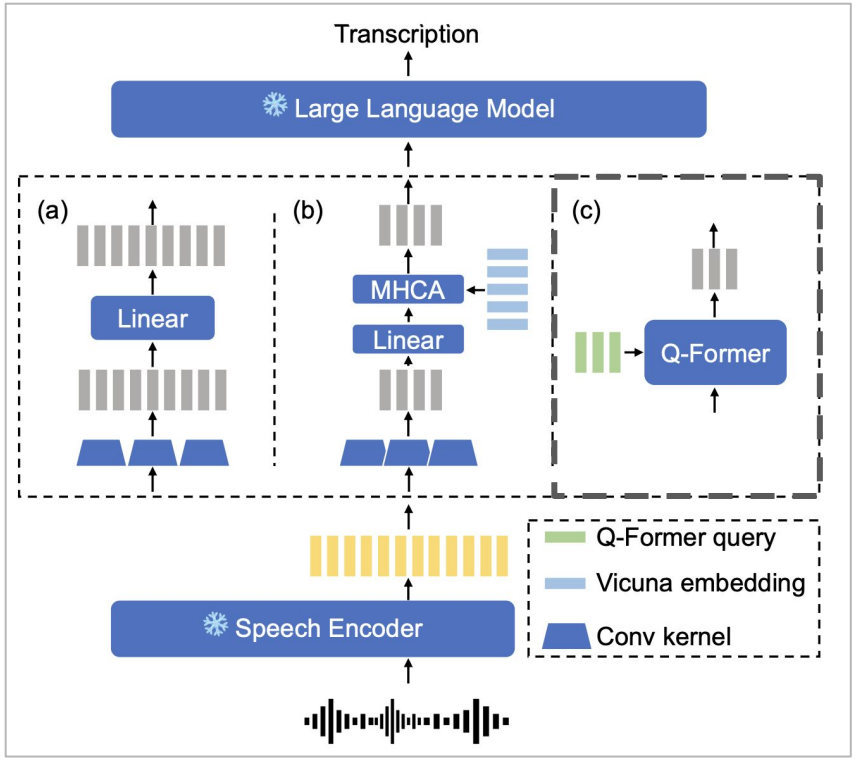


https://arxiv.org/pdf/2309.13963

Audio-Conditoned LLM: Projectors

Comparing quality with:

- **Q-Former**
- Linear layer
- Multi-Head Cross-Attention

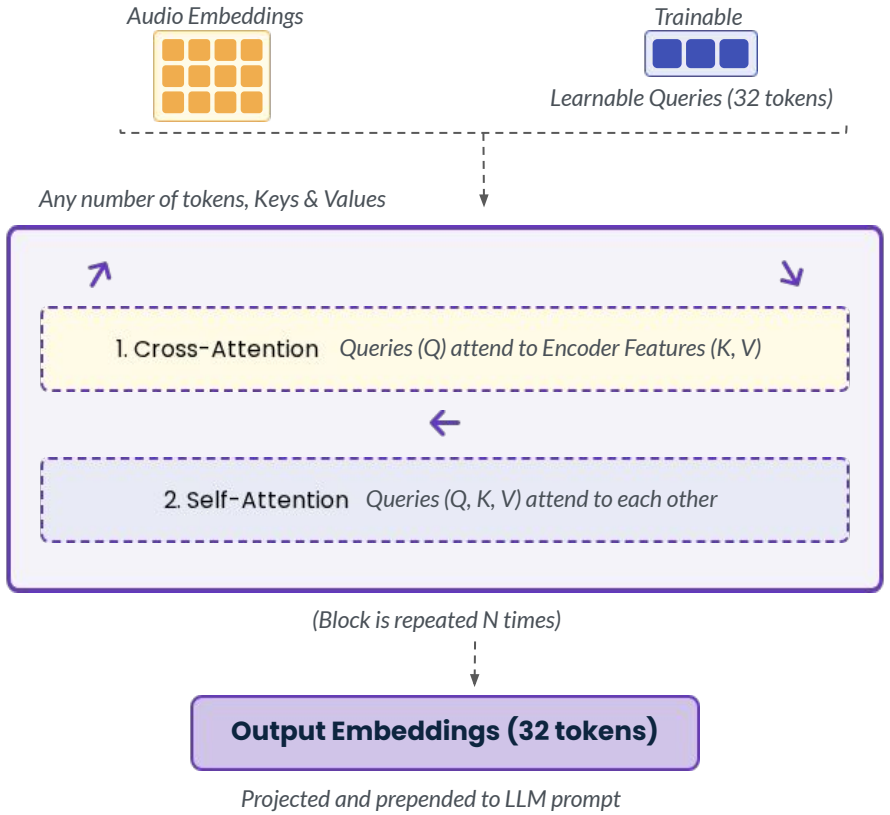


https://arxiv.org/pdf/2309.13963

Audio-Conditoned LLM: Projectors

Comparing quality with:

- **Q-Former**
- Linear layer
- Multi-Head Cross-Attention



Audio-Conditoned LLM: Projectors

Comparing quality with:

- Q-Former
- Linear layer
- Multi-Head Cross-Attention

Model	#Tokens	#Params	LibriSpeech	
			test-clean	test-other
FC	75	24.6M	3.00	6.70
FC	300	23.6M	2.29	5.44
CA	75	133.4M	3.22	7.54
QF	60	24.5M	2.33	5.43
QF	80	24.5M	**2.28**	**5.20**

Audio-Conditoned LLM: Projectors

Comparing quality with:

- Q-Former
- Linear layer
- Multi-Head Cross-Attention

Model	#Tokens	#Params	LibriSpeech	
			test-clean	test-other
FC	75	24.6M	3.00	6.70
FC	300	23.6M	2.29	5.44
CA	75	133.4M	3.22	7.54
QF	60	24.5M	2.33	5.43
QF	80	24.5M	**2.28**	**5.20**

Small difference
Mostly caused by frozen encoder & decoder

Audio-Conditoned LLM: Subsamplers

Reduce till the metrics start to fall

Audio-Conditoned LLM: Subsamplers

Reduce till the metrics start to fall

GigaChat-Audio Internal Ablations

Sub. Factor	Frame Dur.	WER ↓ (%)	BLEU ↑ (%)	Act. Mem. Saving	Rsrvd. Mem. Saving
1	40ms	33.4	9.6	0%	0%
2	80ms	**27.0**	**16.0**	6.4%	19.8%
4	160ms	28.5	15.4	9.6%	21.2%
8	320ms	33.7	11.4	**11.2%**	**30.1%**

Audio-Conditoned LLM: Subsamplers

Reduce the metrics start to fall

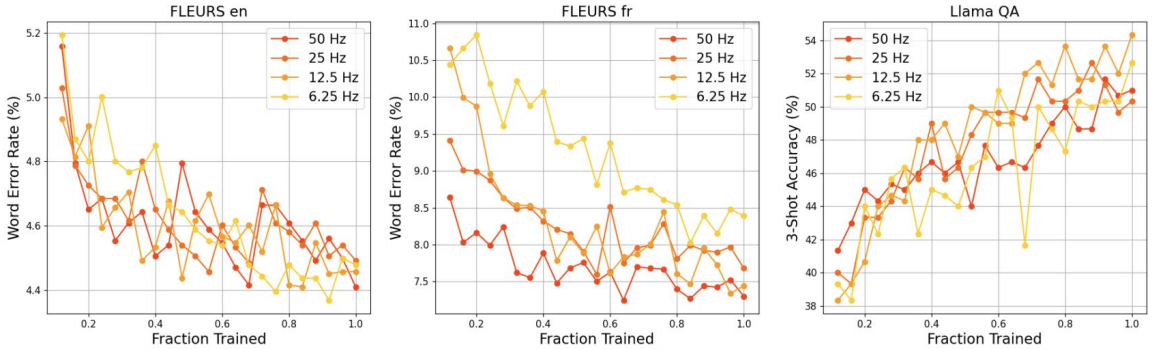


Figure 8: Effect of Downsampling. Word error rate results on FLEURS English (left) and FLEURS French (middle), alongside 3-shot Accuracy on Llama QA (right) for various frame-rates, achieved by increasing the downsampling factor by powers of 2.

https://arxiv.org/pdf/2507.13264v1

Projectors & Subsamplers Recap

Projectors:

- Often Linear / MLP module is good enough choice
- If LLM / encoder frozen - may be useful to use more complicated module
- There is the growing number of papers researching dynamic compressing of audio embeddings, but there are no large-scale with such methods yet

Projectors & Subsamplers Recap

Subsamplers:

- Reduce the metrics start to fall
- There is the growing number of papers researching dynamic compressing of audio embeddings, but there are no large-scale with such methods yet

AudioLLM Training Stages & Data

- Pre-training
  - LLM training (pretrain / sft / rl)
  - Encoder training (SSL / ASR / Specific Training)

AudioLLM Training Stages & Data

- Pre-training
  - LLM training (pretrain / sft / rl)
  - Encoder training (SSL / ASR / Specific Training)
- Modality Alignment
  - Usually - unfreeze adapter / encoder and train on simple tasks like ASR / AST / Classifications / Continutations / ...

AudioLLM Training Stages & Data

- Pre-training
  - LLM training (pretrain / sft / rl)
  - Encoder training (SSL / ASR / Specific Training)
- Modality Alignment
  - Usually - unfreeze adapter / encoder and train on simple tasks like ASR / AST / Classifications / Continutations / ...

AudioLLM Training Stages & Data

- Pre-training
  - LLM training (pretrain / sft / rl)
  - Encoder training (SSL / ASR / Specific Training)
- Modality Alignment
  - Usually - unfreeze adapter / encoder and train on simple tasks like ASR / AST / Classifications / Continutations / ...
- Supervised fine-tuning
  - Usually - finetuning on instructional datasets, unfreezing / adding Lora to LLM decoder

AudioLLM Training Stages & Data

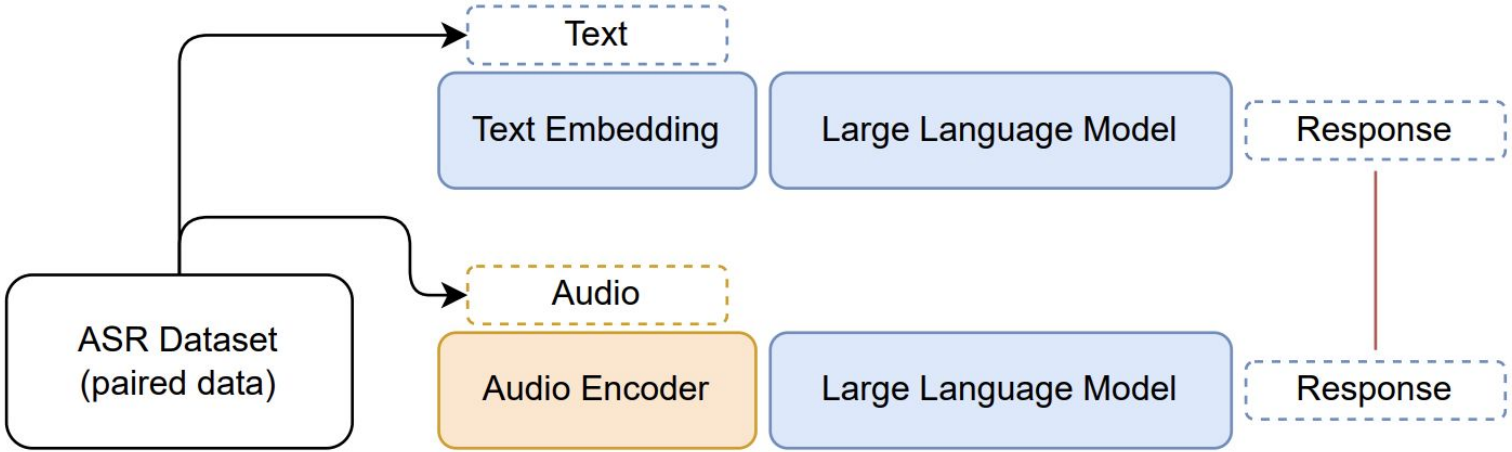
- Pre-training
  - LLM training (pretrain / sft / rl)
  - Encoder training (SSL / ASR / Specific Training)
- Modality Alignment
  - Usually - unfreeze adapter / encoder and train on simple tasks like ASR / AST / Classifications / Continutations / ...
- Supervised fine-tuning
  - Usually - finetuning on instructional datasets, unfreezing / adding Lora to LLM decoder
- Post training
  - RL
  - Context Extension
  - Reasoning

AudioLLM Training Stages & Data

- Alignment data:
  - ASR data
    - AudioChatLLAMa creation

AudioLLM Training Stages & Data

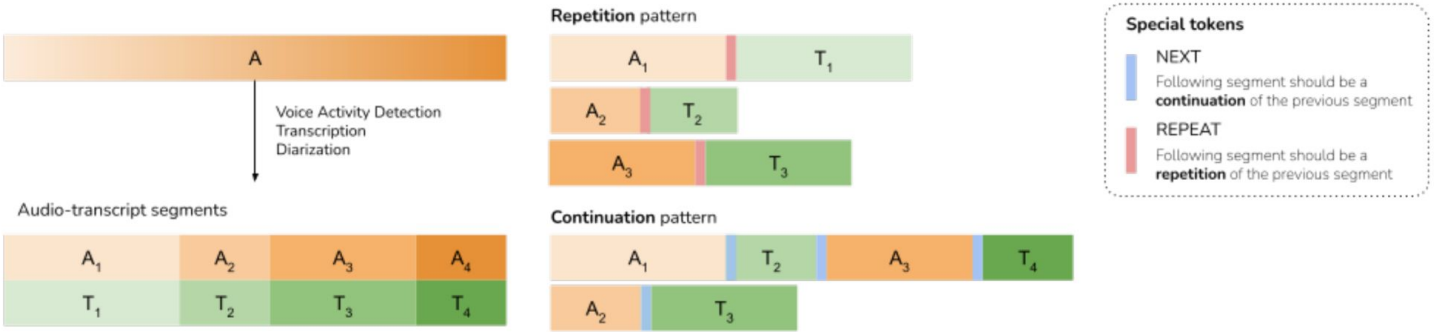
- Alignment data:
  - ASR data
    - AudioChatLLAMa creation



https://arxiv.org/pdf/2311.06753v2

AudioLLM Training Stages & Data

- Alignment data:
  - ASR data
    - AudioChatLLAMa creation
    - Voxtral Interleaving-style creation



https://arxiv.org/pdf/2507.13264v1

AudioLLM Training Stages & Data

- Alignment data:
  - ASR data
    - AudioChatLLAMa creation
    - Voxtral Interleaving-style creation
    - Transcription-based question & answer generation

AudioLLM Training Stages & Data

- Alignment data:
  - ASR data
    - AudioChatLLAMa creation
    - Voxtral Interleaving-style creation
    - Transcription-based question & answer generation
    - Traditional ASR

AudioLLM Training Stages & Data

- Alignment data:
  - ASR data
    - AudioChatLLAMa creation
    - Voxtral Interleaving-style creation
    - Transcription-based question & answer generation
    - Traditional ASR
  - Audio Captioning Data

AudioLLM Training Stages & Data

- Alignment data:
  - ASR & AST data
    - AudioChatLLAMa creation
    - Voxtral Interleaving-style creation
    - Transcription-based question & answer generation
    - Traditional ASR
  - Audio Captioning Data
  - Any Open-Source data

AudioLLM Training Stages & Data

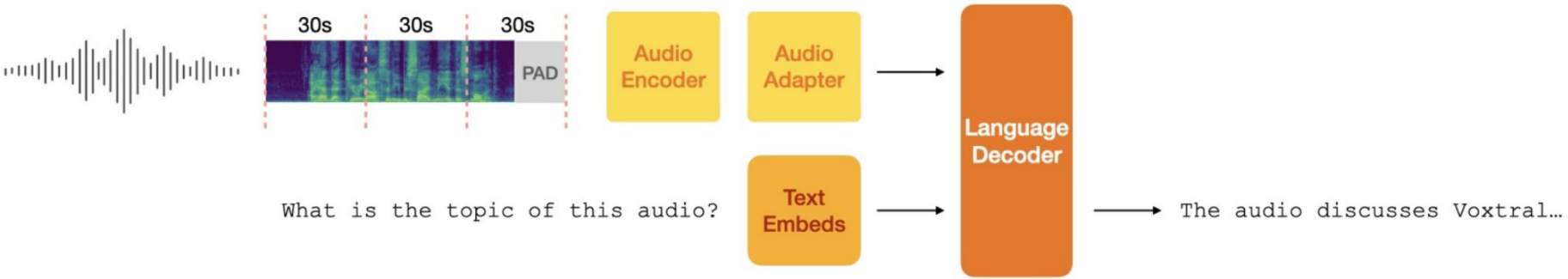
- SFT (supervised fine-tuning) data:
  - ASR & AST data
  - Captioning data
  - Synthetic data (LLM & TTS using)
    - Audio Question Answering
    - Multiturn dialogues
    - Summarizations

Case Study Voxtral

Model: Whisper Encoder + Linear Adapter + Transformer Decoder

Subsampling: 4x to 12.5Hz

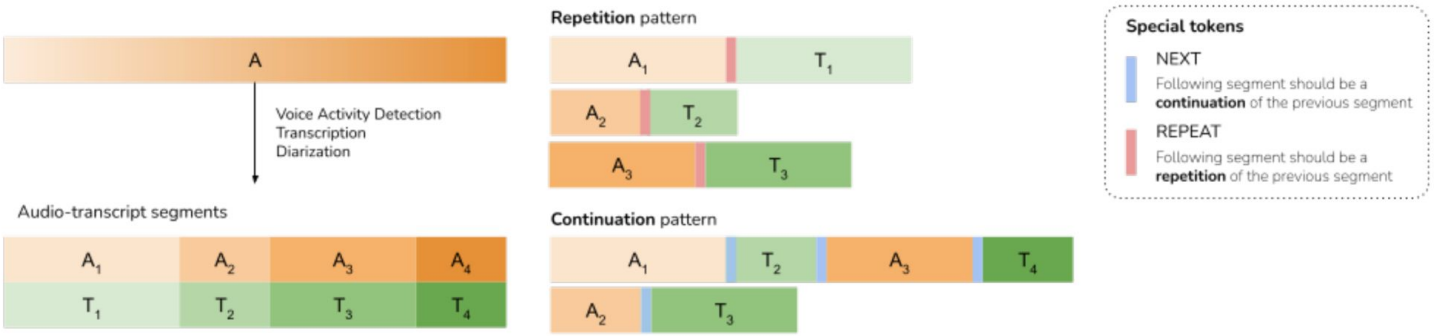
Decoder: 3B for Mini, 24B for Small



https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Pretraining Stage

- Freeze all, except adapter -> Unfreeze all?
- Interleaving-style data



https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: SFT Stage Data

Audio context, Text Query

- Synthetic QA created with Mistral Large on transcripts:
  - Factual questions
  - Needle-In-Haystack Retrieval
  - Reasoning
  - Multilingual
- Synthetic summarization & translation

Audio-Only Input

- Synthetic text sft data with TTS model
- Extract questions from ASR data and generate text answer with Mistral Large
- ASR with special token

https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: DPO

Online-DPO with text reward model, working on transcriptions + answers

https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Evaluation

- Speech Recognition
- Speech Translation

https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Evaluation

- Speech Recognition
- Speech Translation
- Custom sets:
  - Filtering & Synthesis:
    - GSM8K
    - TriviaQA
    - MMLU

https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Evaluation

- Speech Recognition
- Speech Translation
- Custom sets:
  - Filtering & Synthesis
  - LLM-Judge Speech Understanding Benchmark:

https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Evaluation

- Speech Recognition
- Speech Translation
- Custom sets:
  - Filtering & Synthesis
  - LLM-Judge Speech Understanding Benchmark:
    - Binary helpfulness score: correct / incorrect
    - Grade Score: from 0 to 5

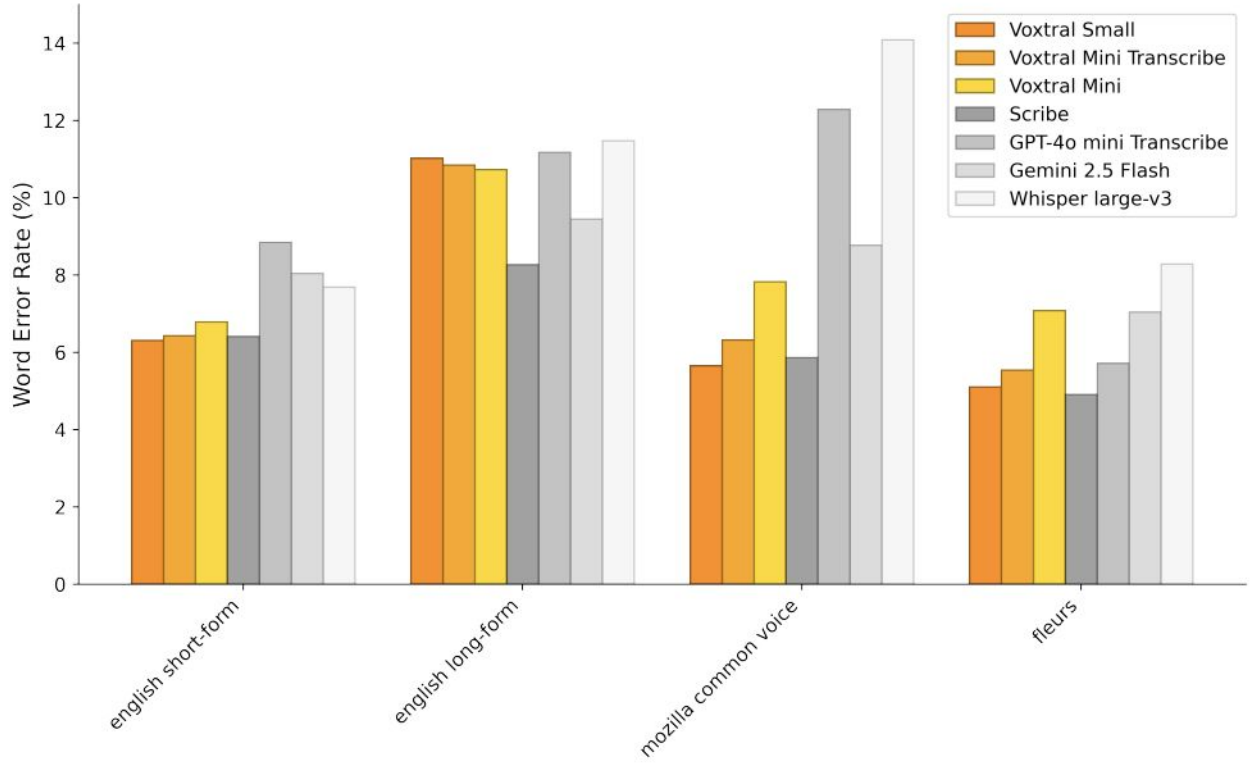
https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Evaluation

- Speech Recognition
- Speech Translation
- Custom sets:
  - Filtering & Synthesis
  - LLM-Judge Speech Understanding Benchmark:
    - Binary helpfulness score: correct / incorrect
    - Grade Score: from 0 to 5
- Text Benches

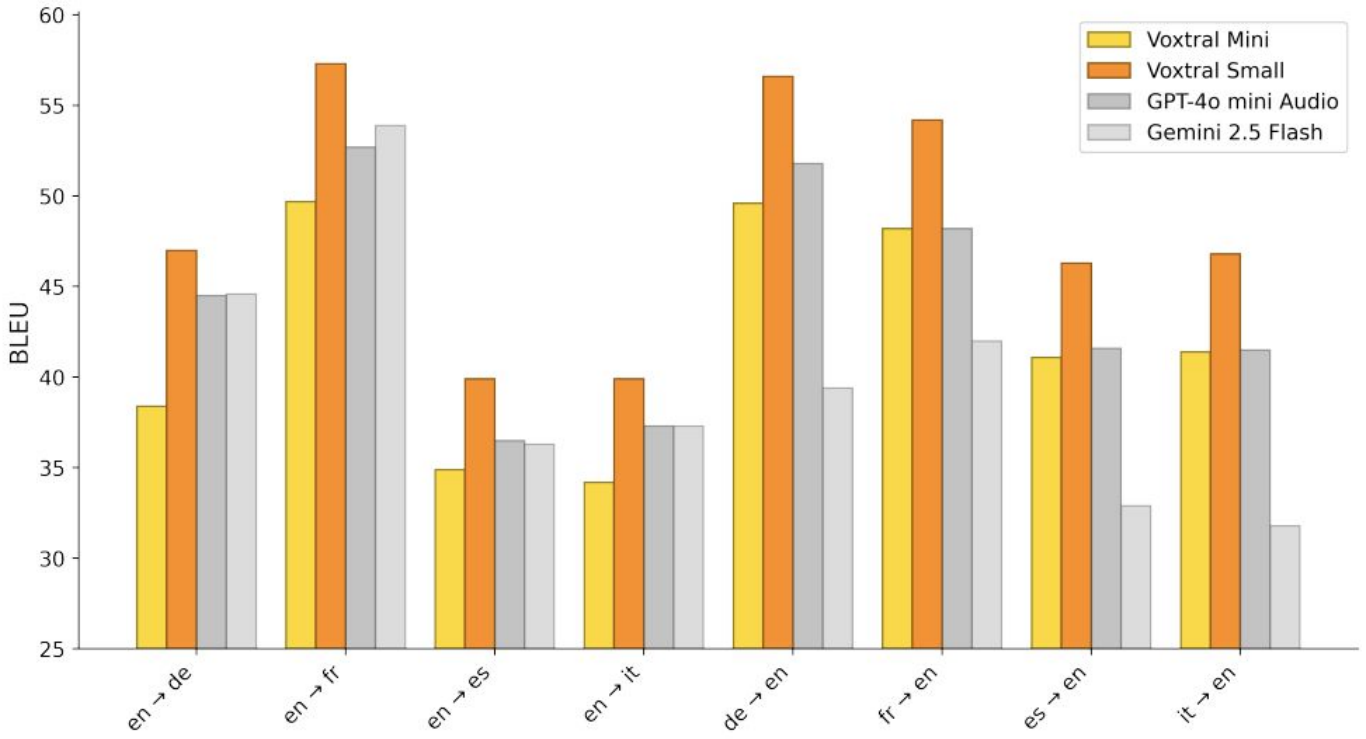
https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Speech Recognition Metrics



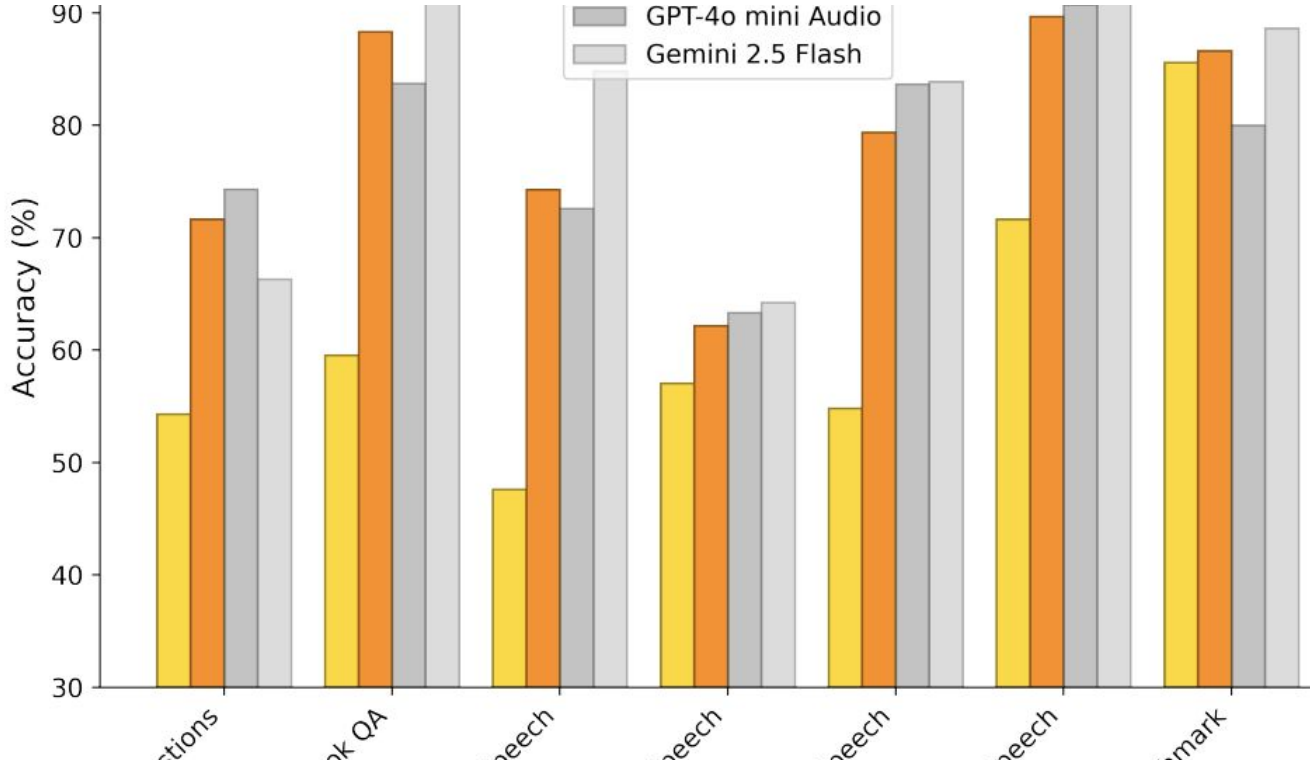
https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Speech Translation Metrics



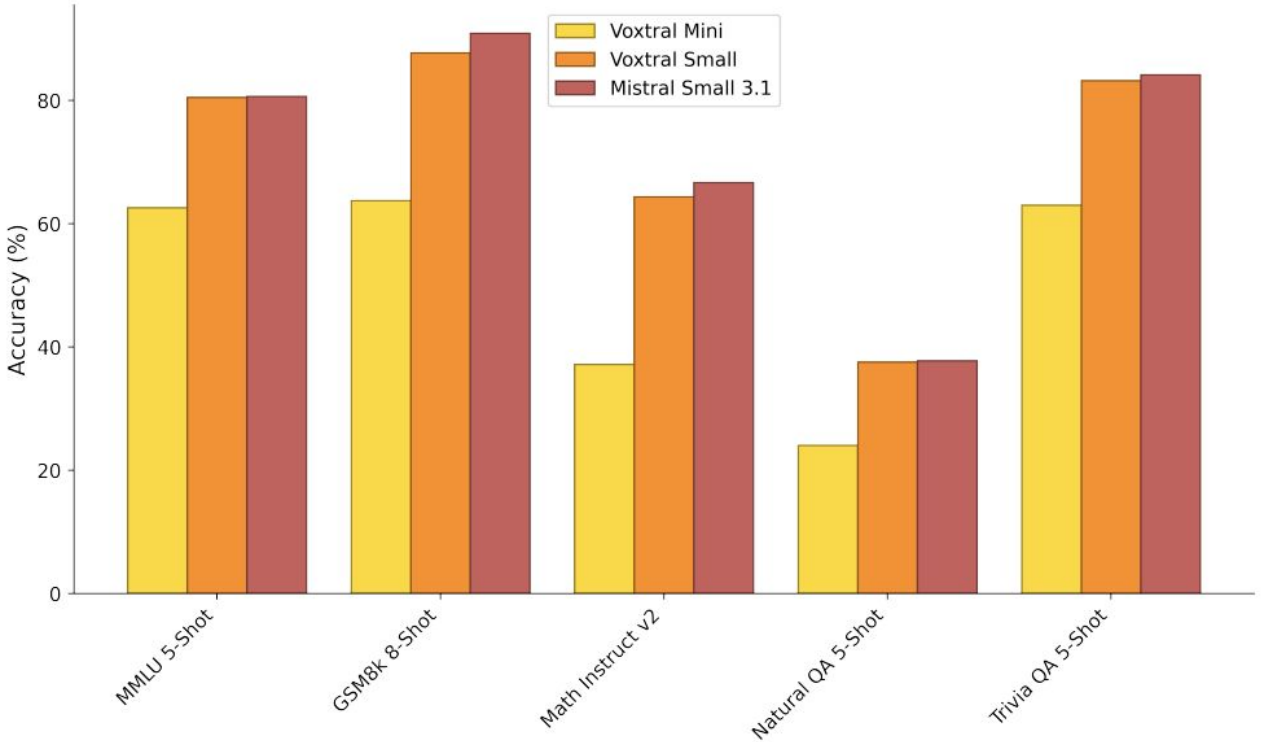
https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Speech Understanding Metrics



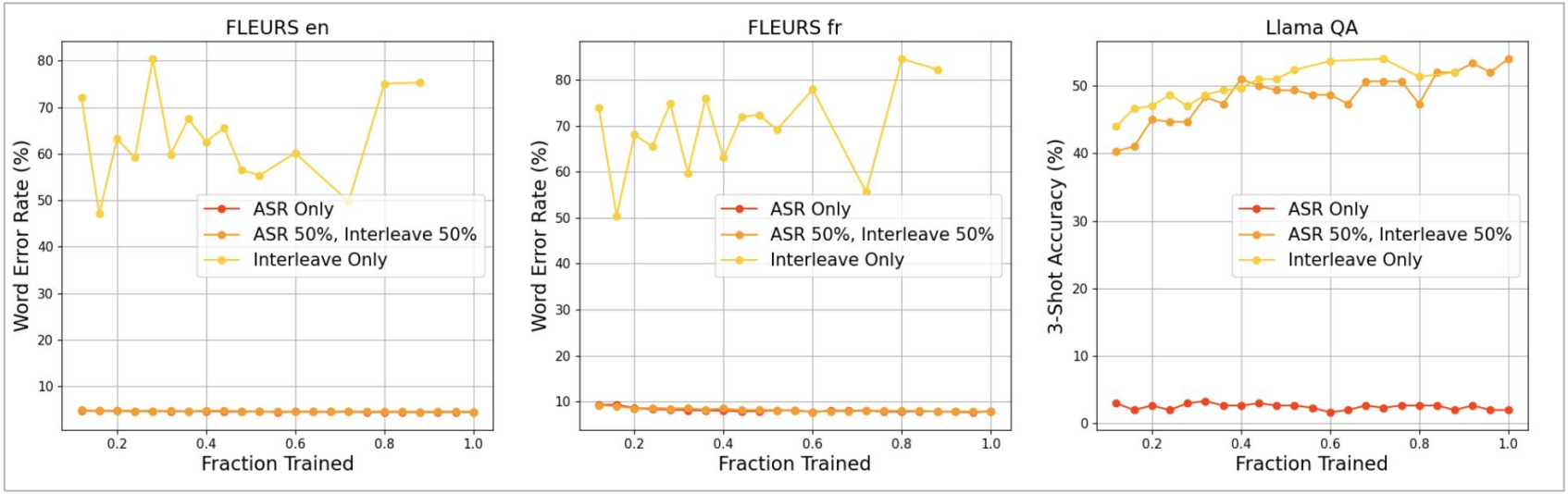
https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Speech Understanding Metrics



https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: Pretrain Taks Ablation



https://arxiv.org/pdf/2507.13264v1

Case Study Voxtral: DPO effect

Model	% LLM Judge ↑	Grade ↑	En Short WER ↓
Voxtral Mini SFT	83.47 ± 2.17	3.92 ± 0.04	**6.77**
Voxtral Mini Offline DPO	84.91 ± 3.21	3.92 ± 0.08	6.78
Voxtral Mini Online DPO	**85.59 ± 3.77**	**4.08 ± 0.07**	6.79
Voxtral Small SFT	86.61 ± 0.96	4.16 ± 0.03	**6.31**
Voxtral Small Offline DPO	87.29 ± 1.65	4.19 ± 0.04	6.32
Voxtral Small Online DPO	**88.31 ± 2.03**	**4.38 ± 0.06**	6.50
GPT-4o mini Audio	80.00 ± 2.97	3.97 ± 0.05	-
Gemini 2.5 Flash	88.64 ± 2.28	4.54 ± 0.07	8.04

https://arxiv.org/pdf/2507.13264v1

Recap

- Why LLM
- Why AudioLLM
- What is Audio LLM
- Types of AudioLLM
- Audio-conditioned LLM tasks
- Audio Input: encoders & speech tokens
- Integration Audio embeddings into LLM
- Training stages and data
- Case Study

Спасибо за внимание!

@Alexandr_Maximenko