Discrete Audio Tokens for Multimodal LLMs

Mirco Ravanelli

“Traditional” LLMs

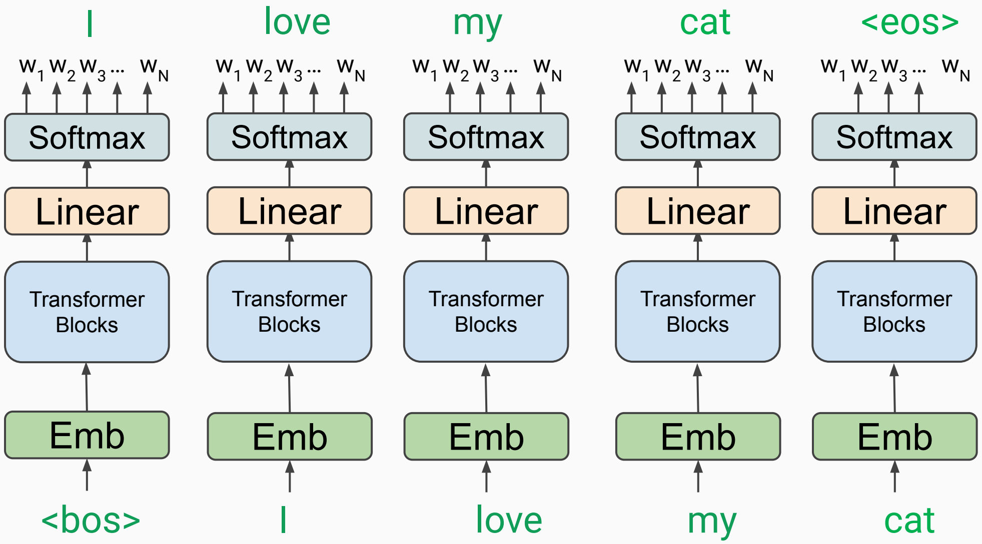
- Language Models have a **long history**.
- **Goal**: Assign a probability to every sequence of words **w**

$$P(w_1, w_2, ..., w_N) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_N|w_1, ..., w_{N-1})$$
$$= \prod_{i=1}^{N} P(w_i|w_1, .., w_{i-1})$$

- A language model can be trained to predict the **next word** given **all the previous ones**.

N-grams	→	MLPs	→	RNNs	→	Transformers
1980~2010		*2002~2010*		*2010~2018*		*2018~Now*

- **Dominant Approach**: autoregressive transformer trained to predict the next word.



Brute Force: Massively scaling up the model with more data and parameters results in impressive performance.

Why Multimodality?

- A multimodal system can offer several advantages:

Deeper Understanding

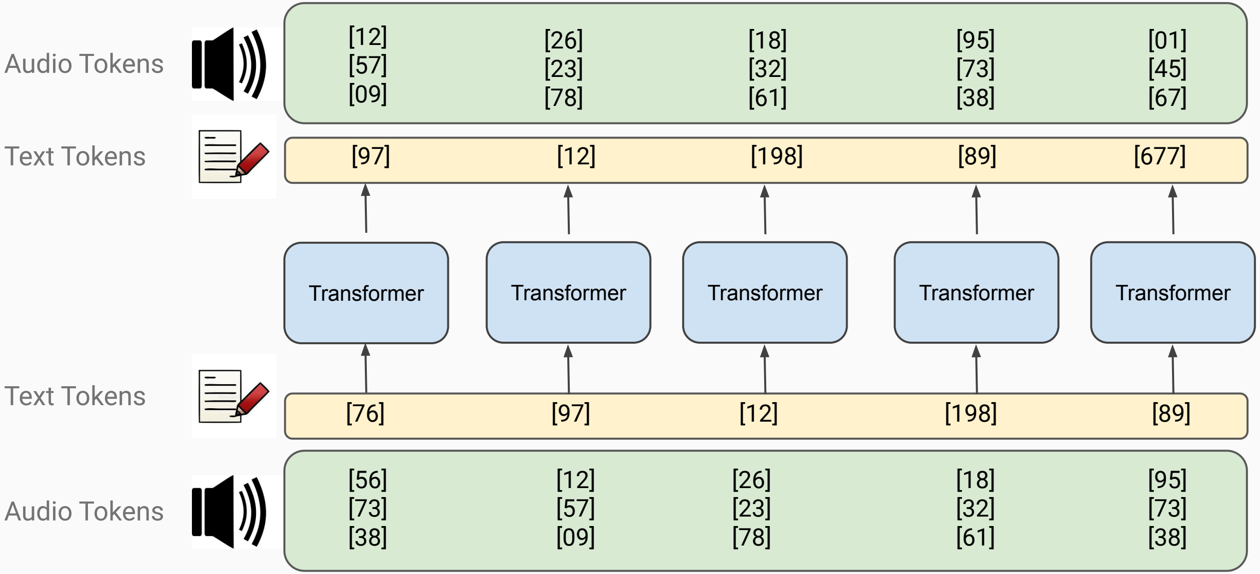
Better Generalization

Fewer Hallucinations

More Flexible
Human-Machine Interaction

Multimodal LLMs

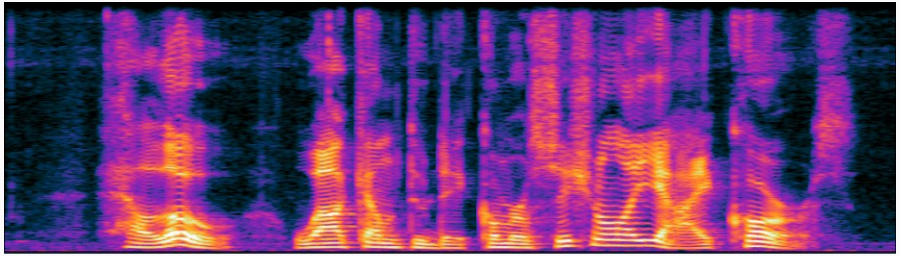
- One way to train multimodal LLMs? **Tokenize** all!



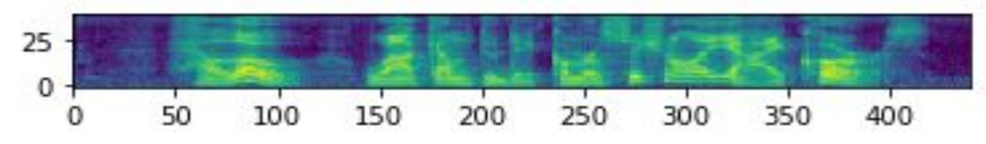
How to Tokenize Audio?

- Audio and Speech are **continuous**!

Waveform



Spectrogram



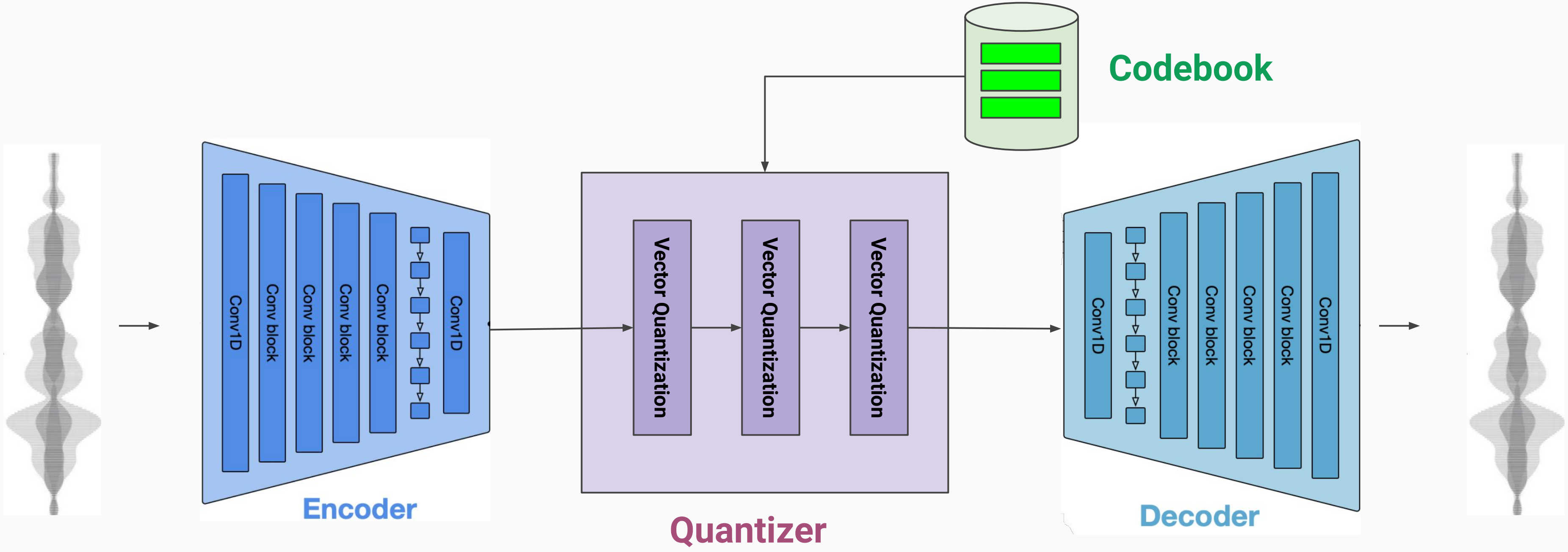
FBANKs

How to Discretize Audio and Speech Without Losing Information?

- SSL models (e.g., wav2vec, HuBERT, WavLM) also produce continuous representations!

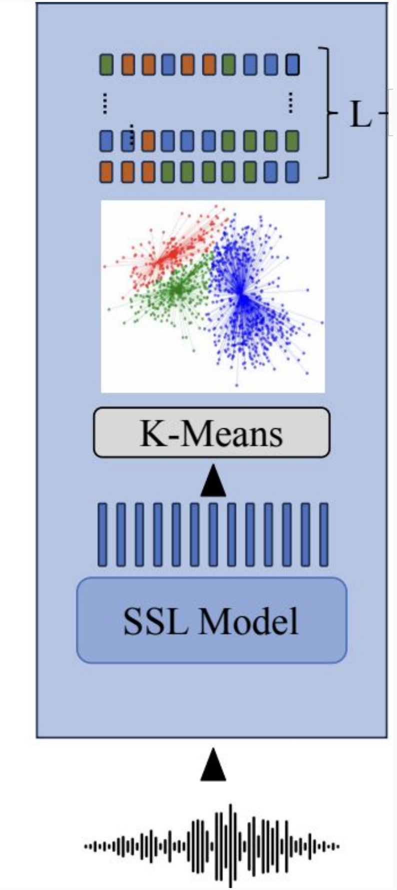
Audio Codes

- **Compression tokens** (Audio Codec)



Examples: *SoundStream, Encodec*

Semantic Tokens



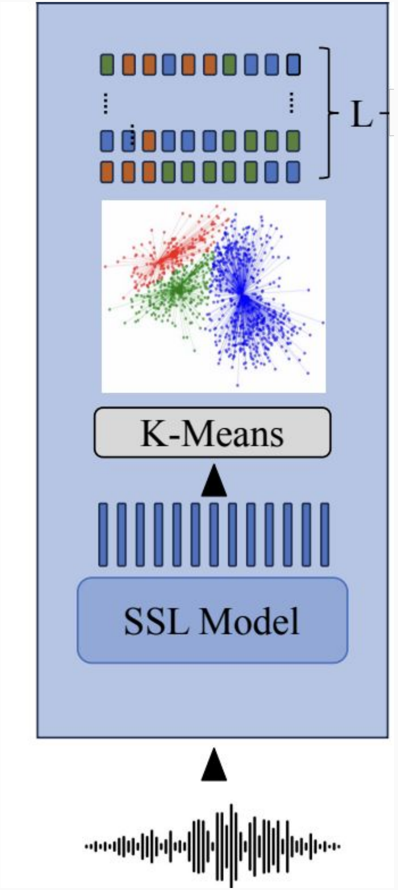
Semantic Tokens

- Large self-supervised models (e.g., *Wav2vec*, *HuBERT*, *WaLM*) achieve **state-of-the-art** performance.
- These models are based on **continuous** representations
- We convert continuous representations into discrete tokens through **clustering**.

How Should We Extract Discrete Audio Tokens from Self-Supervised Models?

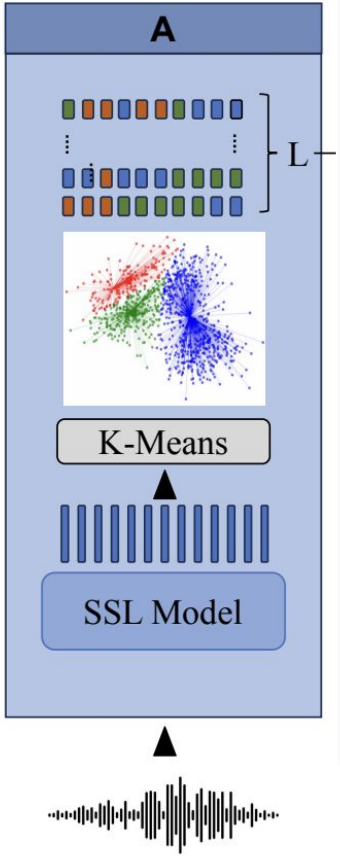
Pooneh Mousavi[1,2], Jarod Duret[3], Salah Zaiem[4], Luca Della Libera[1,2], Artem Ploujnikov[5,2], Cem Subakan[6,2,1], Mirco Ravanelli[1,2,5]

Extracting Semantic Tokens



1. *Which layers should we cluster?*
2. *What is the optimal number of clusters?*
3. *Which datasets are we using for clustering?*
4. *What is the best approach to train the decoder (vocoder)?*
5. *How should we initialize the embeddings effectively?*
6. *Can we extract universal tokens for both discriminative and generative tasks?*

Extracting Semantic Tokens



Extracting Semantic Tokens

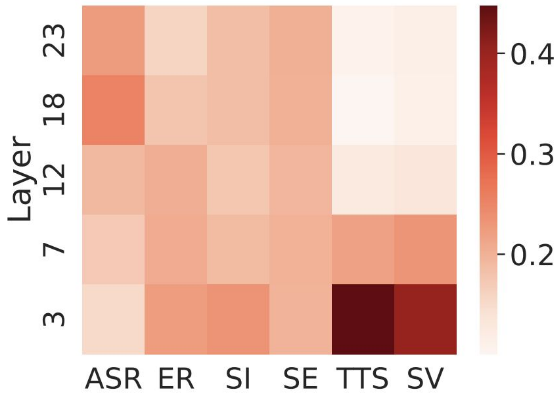


Figure 3: *Attention analysis across various tasks and layers of the discrete WavLM model with in-domain tokenizers.*

- Automatic Speech Recognition (ASR)
- Emotion Recognition (ER)
- Speaker Identification (SI)

- Speech Enhancement (SE)
- Text-to-Speech (TTS)
- Scalable Vocoder (SV)

Extracting Semantic Tokens

Table 1: *Assessing the impact of the number of clusters and embedding initialization on discrete WavLM-Large across different tasks.*

Setting	ASR (EN)	ASR (FR)	SID	ER	SE		TTS	
	WER ↓	*WER* ↓	*ACC* ↑	*ACC* ↑	*DNSMOS*↑	*dWER*↓	*UTMOS*↑	*WER* ↓
Effect of Number Of Clusters								
1000	7.15	34.61	79.0	61.8	3.93	6.75	3.65	5.76
2000	6.96	32.94	79.5	67.2	3.93	6.58	3.55	5.62
Effect of Embedding Initialization								
Random	6.96	32.94	81.0	67.2	3.93	6.75	3.65	5.76
PreTrained & finetune	8.93	35.81	77.5	63.9	3.93	6.82	3.64	6.62
PreTrained & freeze	9.26	35.12	73.1	67.0	3.93	6.98	3.66	6.42

- The optimal number of clusters varies by task.
- Typically, using 1000 to 2000 clusters yields good performance.

- There is no advantage observed in initializing the embedding with pretrained centroid embeddings.

Extracting Semantic Tokens

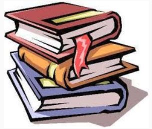
Table 2: *Out-of-domain and in-domain performance of discrete HuBERT and WavLM models across the downstream tasks.*

SSL Model	Tokenizer	ASR (EN)	ASR (FR)	SID	ER	SE		TTS		Vocoder	
		WER ↓	*WER* ↓	*ACC* ↑	*ACC* ↑	*DNSMOS*↑	*dWER*↓	*UTMOS*↑	*WER* ↓	*UTMOS*↑	*dWER*↓
HuBERT Large [4]	In-Domain	7.89	38.29	67.2	64.5	3.98	17.64	3.61	6.46	3.50	4.49
	Out-Of-Domain	N/A	39.50	67.8	61.7	3.95	15.92	3.54	5.45	3.48	2.92
WavLM Large [3]	In-Domain	6.96	32.94	81.0	67.2	3.93	6.75	3.65	5.76	3.49	2.98
	Out-Of-Domain	N/A	36.25	79.0	61.9	3.96	6.49	3.61	5.73	3.68	2.95

- As expected, the in-domain tokenizer outperforms its OOD counterpart.
- However, the performance drop is not always huge.

Discrete Audio and Speech Benchmark (DASB)

Are compression tokens better than semantic tokens?



Literature offers no clear answer

DASB - Discrete Audio and Speech Benchmark

Pooneh Mousavi[1,2], Luca Della Libera[1,2], Jarod Duret[3], Artem Ploujnikov[4,2], Cem Subakan[5,2,1], Mirco Ravanelli[1,2,4]
[1]Concordia University [2]Mila - Quebec AI Institute [3]Avignon Université
[4]Université de Montréal [5]Université Laval

[DASB Code Repo]

Discrete Audio and Speech Benchmark (DASB)

Tokenizer	Type
Discrete HuBERT	Semantic
Discrete WavLM	Semantic
Discrete Wav2Vec2	Semantic
EnCodec	Compression
DAC	Compression
SpeechTokenizer	Hybrid

Task	Type
Automatic Speech Recognition (ASR)	Discriminative
Speaker Identification/Verification (SID, SV)	Discriminative
Emotion Recognition (ER)	Discriminative
Intent Classification (IC)	Discriminative
Keyword Spotting (KS)	Discriminative
Speech Enhancement (SE)	Generative
Speech Separation (SS)	Generative
Text-to-Speech (TTS)	Generative

Discrete Audio and Speech Benchmark (DASB)

Table 2: Benchmarking results for discriminative tasks.

Models/Tasks	ASR-En WER ↓		ASR-multiling WER ↓		ER ACC ↑	IC ACC ↑	KS ACC ↑	SI ACC ↑	SV 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

- **Semantic tokens** outperform compression tokens in most discriminative tasks.
- The exception is **speaker recognition**, where EnCodec excels.
- **Big gap** compared to **continuous** baselines!

Discrete Audio and Speech Benchmark (DASB)

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

- Semantic tokens show the best performance for generative tasks as well
- **Big gap** compared to **continuous** baselines

Discrete Audio and Speech Benchmark (DASB)

TTS Samples

Ranking aggregation for models (medium bitrate)

Model	Disc.	Gen.	Comb.
Discrete HuBERT	2.66	3.62	3.11
Discrete WavLM	**2.00**	2.75	**1.94**
Discrete Wav2Vec2	3.33	**2.68**	3.41
EnCodec	4.11	3.93	4.23
DAC	5.55	4 .06	4.64
SpeechTokenizer	3.44	3.81	3.64

Discrete Audio and Speech Benchmark (DASB)

- We are still **far** from an ideal solution!

- Semantic tokens are computationally expensive
- They don’t preserve speaker identities well
- Performance drops significantly compared to continuous representations

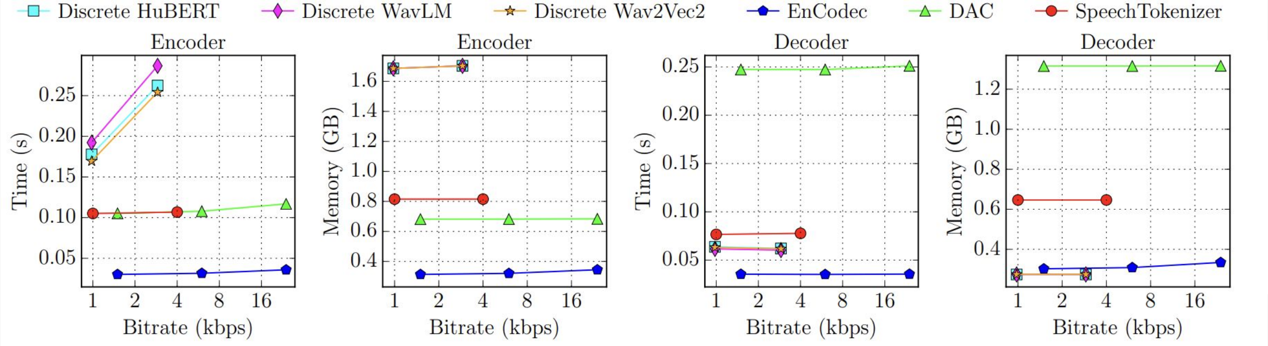


Figure 2: Time and memory required to process an utterance of 16 seconds for encoders and decoders of the considered audio tokenizers on an NVIDIA GeForce RTX 3070 GPU @ 8 GB.

Future Directions

- Universal Audio Tokens

Emotion Recognition

Speaker Recognition

Speech Recognition

Audio Editing

Text-to-Speech

Enhancement

......

UNIVERSAL©
Audio Tokens

Some Ideas

- **Massive Multitask** learning (similar to PASE)
- **Hierarchical** codebooks with **Dynamic Allocation** (more details for Music)
- **Perceptual Loss Optimization** (Similar to MetricGAN)
- Better **Multi-scale** processing**.**

Future Directions

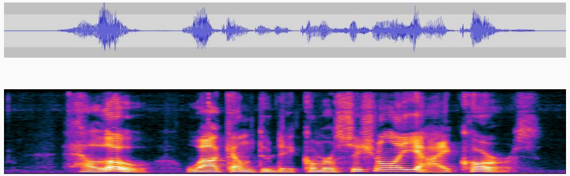
- Can we learn “**interpretable**” audio tokens?

Text: “The City of Montréal”

Tokenized: “[The] [City] [of] [Mont] [ré] [al]”

Each token is **easily interpretable**, as it maps to a specific part of the text.

Audio

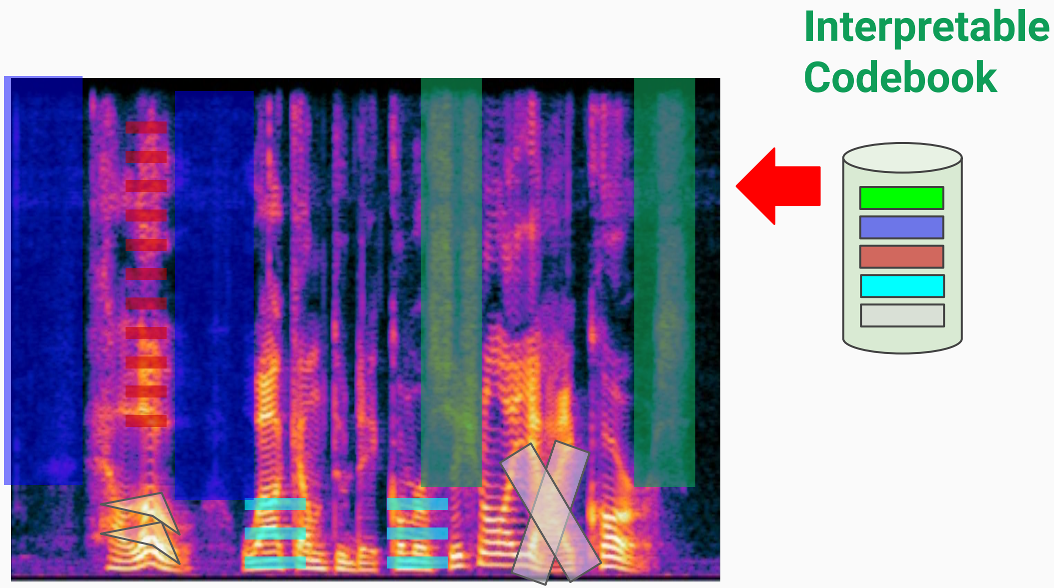


Tokenized Audio

[56]	[12]	[26]	[18]
[73]	[57]	[23]	[32]
[38]	[09]	[78]	[61]

No clear mapping to the original signal

Future Directions



Possible Advantages

- Increased **Transparency**
- Easier **Error Analyses**
- Interpretability might act as a **powerful regularizer**, enhancing both generalization and performance.

Our Related Paper: NeurIPS 2024, ICML 2024

Future Directions

- Is **audio tokenization** the best way to go?

- Survey Paper and Extended Benchmark (in Progress)

Thank you!

Pooneh Mousavi

Artem Ploujnikov

Luca Della Libera

Jarod Duret

Salah Zaiem

Cem Subakan