Large-Scale Self-Supervised Pre-Training for Automatic Speech Recognition

李雨鑫

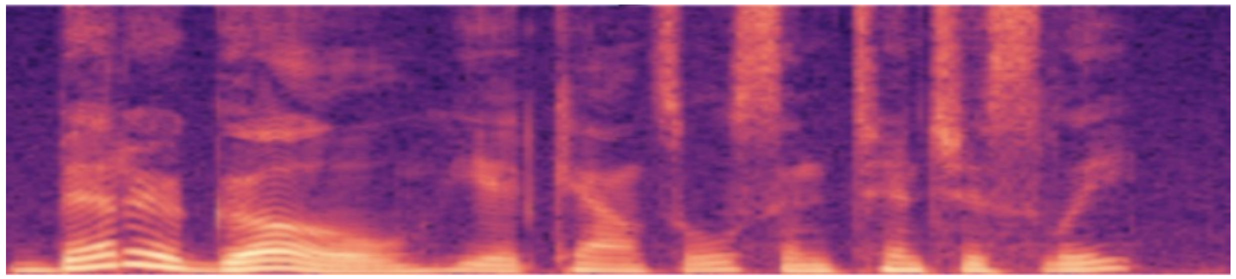
W. -N. Hsu, B. Bolte, Y. -H. H. Tsai, K. Lakhotia, R. Salakhutdinov and A. Mohamed, "**HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units**," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 3451-3460, 2021, doi: 10.1109/TASLP.2021.3122291.

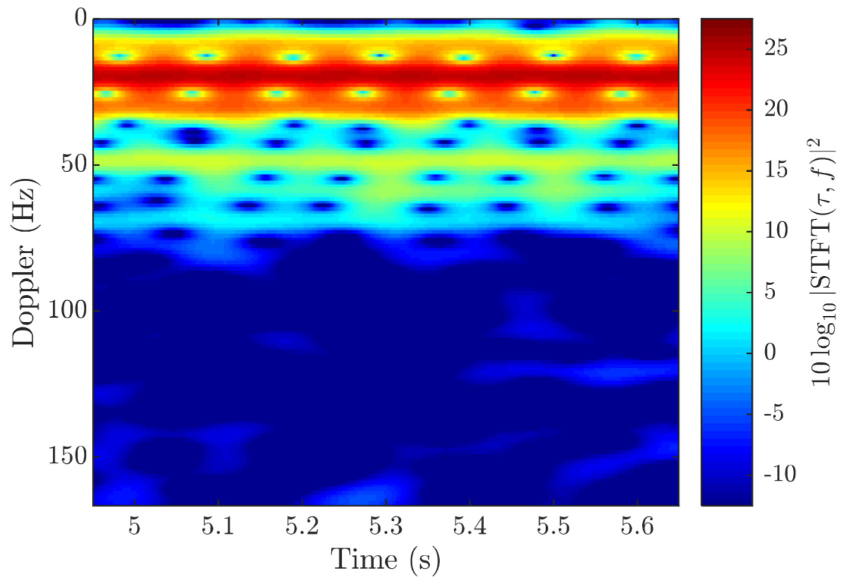
Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., ... & Wei, F. (2022). **Wavlm: Large-scale self-supervised pre-training for full stack speech processing**. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1505-1518.

Chiu, C. C., Qin, J., Zhang, Y., Yu, J., & Wu, Y. (2022, June). **Self-supervised learning with random-projection quantizer for speech recognition.** In *International Conference on Machine Learning* (pp. 3915-3924). PMLR.

Why ASR：

There are some common features between speech signal and HRRP.


speech



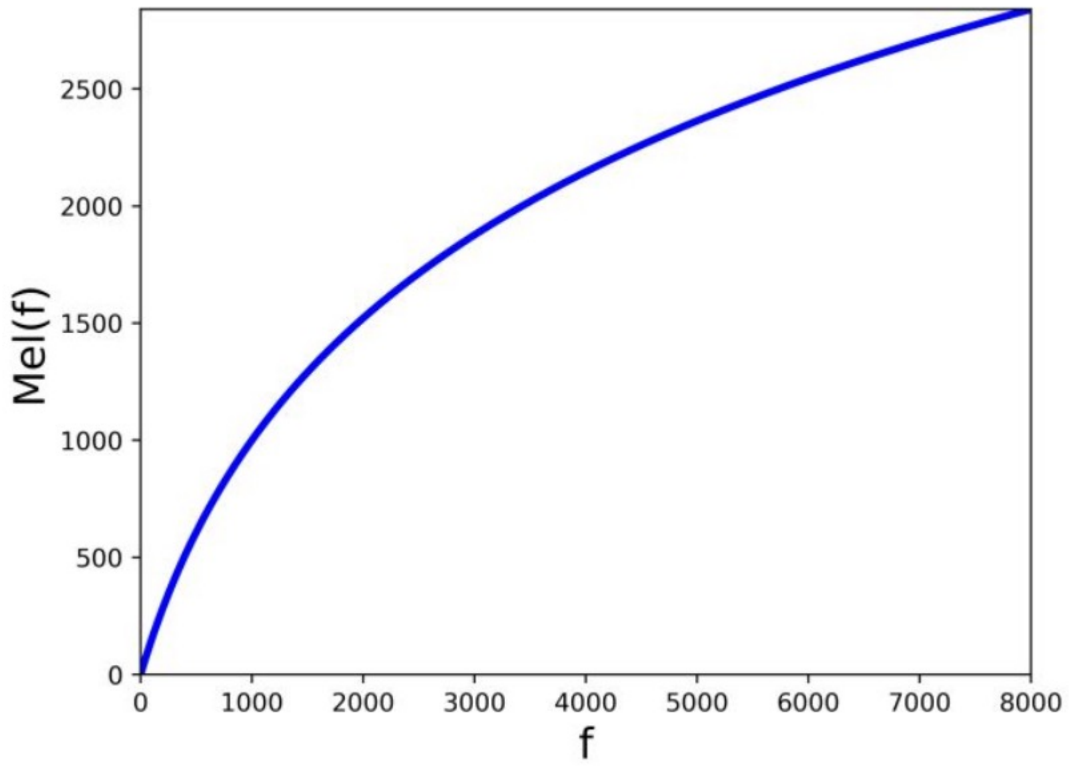
Challenge：

One challenge in building BERT-style self-supervised learning for speech is to bridge the gap between continuous speech signals and the discrete text tokens.

MFCC features：

MFCC是在Mel标度频率域提取出来的倒谱参数

$$Mel(f) = 2585 \times log(1 + \frac{f}{700})$$



pre–emphasis (High–pass)

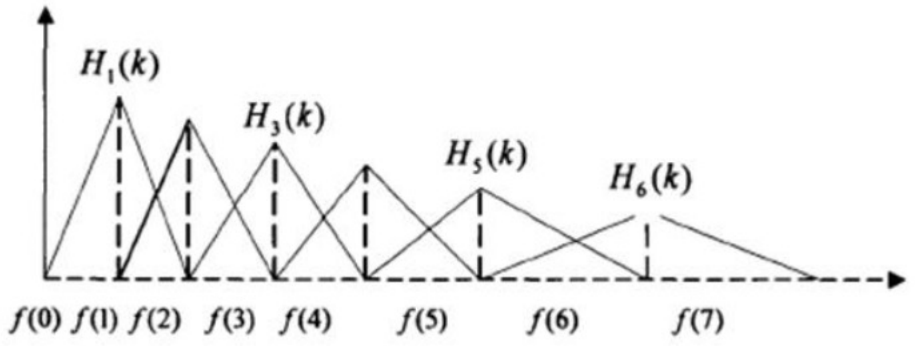
framing

windowing (STFT)

FFT

Triangular bandpass filter

DCT



HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed

Facebook

Representation Learning via Masked Prediction:

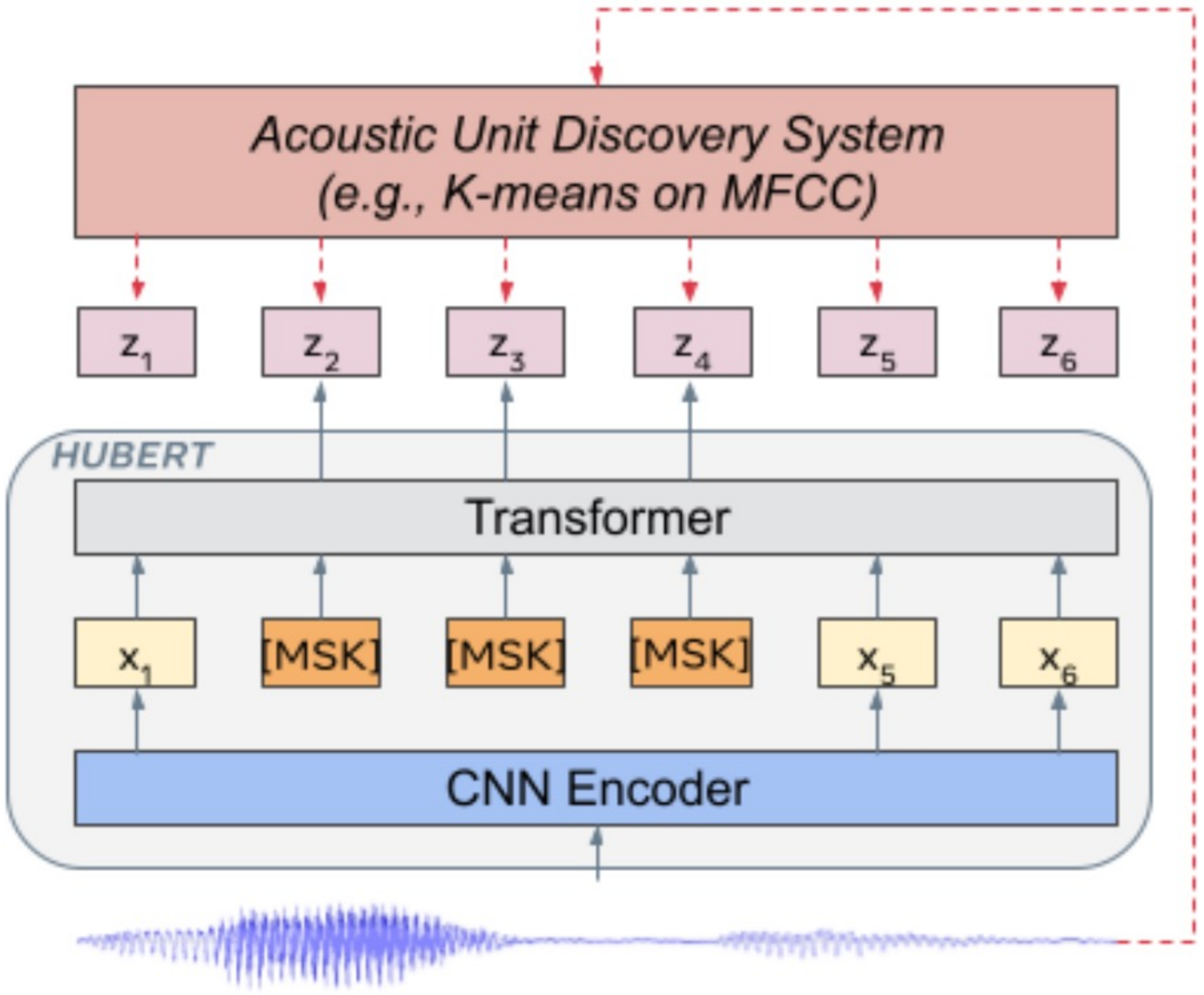
$X = [x_1, \cdots, x_T]$

$h(X) = Z = [z_1, \cdots, z_T]$

$z_t \in [C]$ is a C-class categorical variable

h is a clustering model, e.g. k-means.

The convolutional waveform encoder generates a feature sequence at a 20ms framerate for audio sampled at 16kHz (CNN encoder down-sampling factor is 320x).

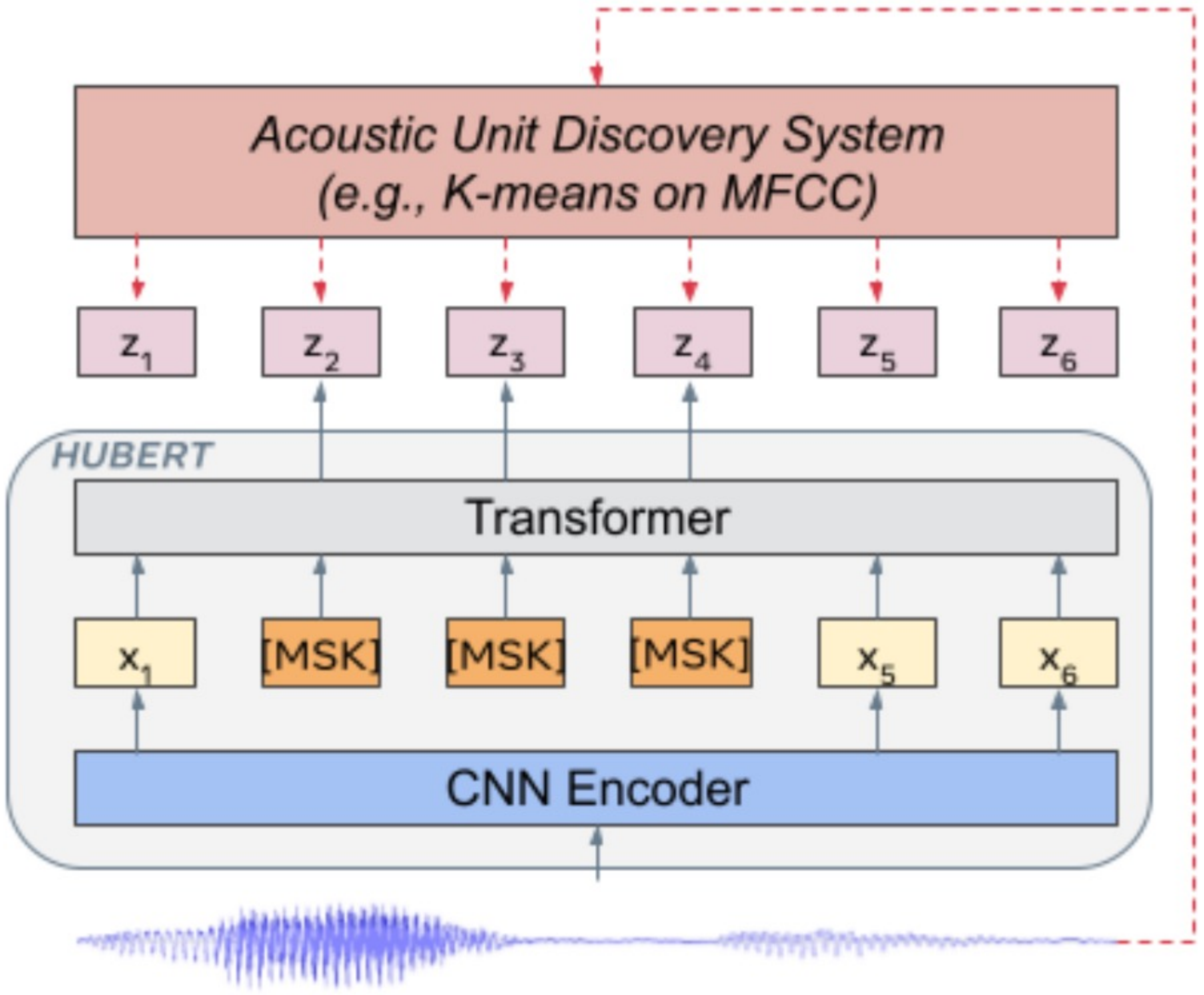


Representation Learning via Masked Prediction:

Let $M \subset [T]$ denote the set of indices to be masked for a length-T sequence X, and $\tilde{X} = r(X, M)$ denote a corrupted version of X where x_t is replaced with a mask embedding $\tilde{x}$ if $t \in M$. A masked prediction model f takes as input $\tilde{X}$ and predicts a distribution over the target indeces at each timestep $p_f(\cdot \mid \tilde{X}, t)$. There are two decisions to be made for masked prediction: *how to mask* and *where to apply the prediction loss*.

$$L_m(f; X, M, Z) = \sum_{t \in M} \log p_f(z_t \mid \tilde{X}, t),$$

$$L = \alpha L_m + (1 - \alpha) L_u$$


Acoustic Unit Discovery System
(e.g., K-means on MFCC)
z_1 z_2 z_3 z_4 z_5 z_6
HUBERT
Transformer
x_1 [MSK] [MSK] [MSK] x_5 x_6
CNN Encoder

Representation Learning via Masked Prediction:

$$L_m(f; X, M, Z) = \sum_{t \in M} \log p_f(z_t \mid \tilde{X}, t),$$

$$L = \alpha L_m + (1 - \alpha) L_u$$

In the other extreme with $\alpha = 1$, the loss is only computed over the masked timesteps where the model has to predict the targets corresponding to the unseen frames from context, analogous to language modeling. It forces the model to learn both the acoustic representation of unmasked segments and the long-range temporal structure of the speech data.

teacher	C	PNMI	dev-other WER (%) $\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 0.0$
Chenone (supervised top-line)	8976	0.809	10.38	9.16	9.79
K-means on MFCC	50	0.227	18.68	31.07	94.60
	100	0.243	17.86	29.57	96.37
	500	0.276	18.40	33.42	97.66
K-means on BASE-it1-layer6	500	0.637	11.91	13.47	23.29
K-means on BASE-it2-layer9	500	0.704	10.75	11.59	13.79

Learning with Cluster Ensembles

A simple idea to improve target quality is to utilize multiple clustering models.

$$L_m(f; X, \{Z^{(k)}\}_k, M) = \sum_{t \in M} \sum_k \log p_f^{(k)}(z_t^{(k)} \mid \tilde{X}, t)$$

The distribution over codewords is parameterized with:

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\mathrm{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(A^{(k)} o_t, e_{c'})/\tau)}$$

Where A is the projection matrix.

Learning with Cluster Ensembles

In addition to using cluster ensembles, another direction for improved representation is refining the cluster assignments throughout the learning process.

Representation Learning via Masked Prediction:

Step1:

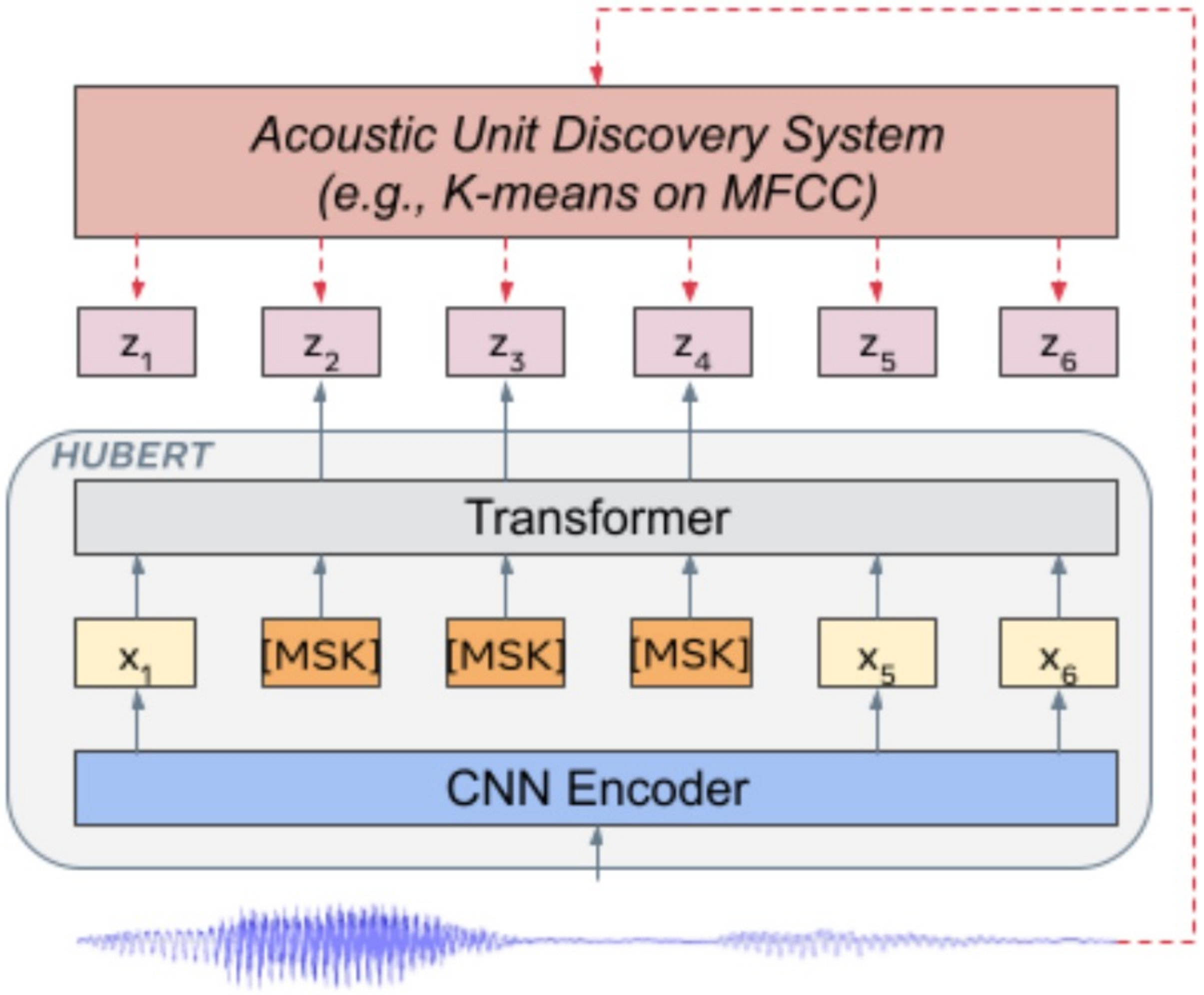
960 hour data on 39-dim MFCC features with 100 clusters.

Step2:

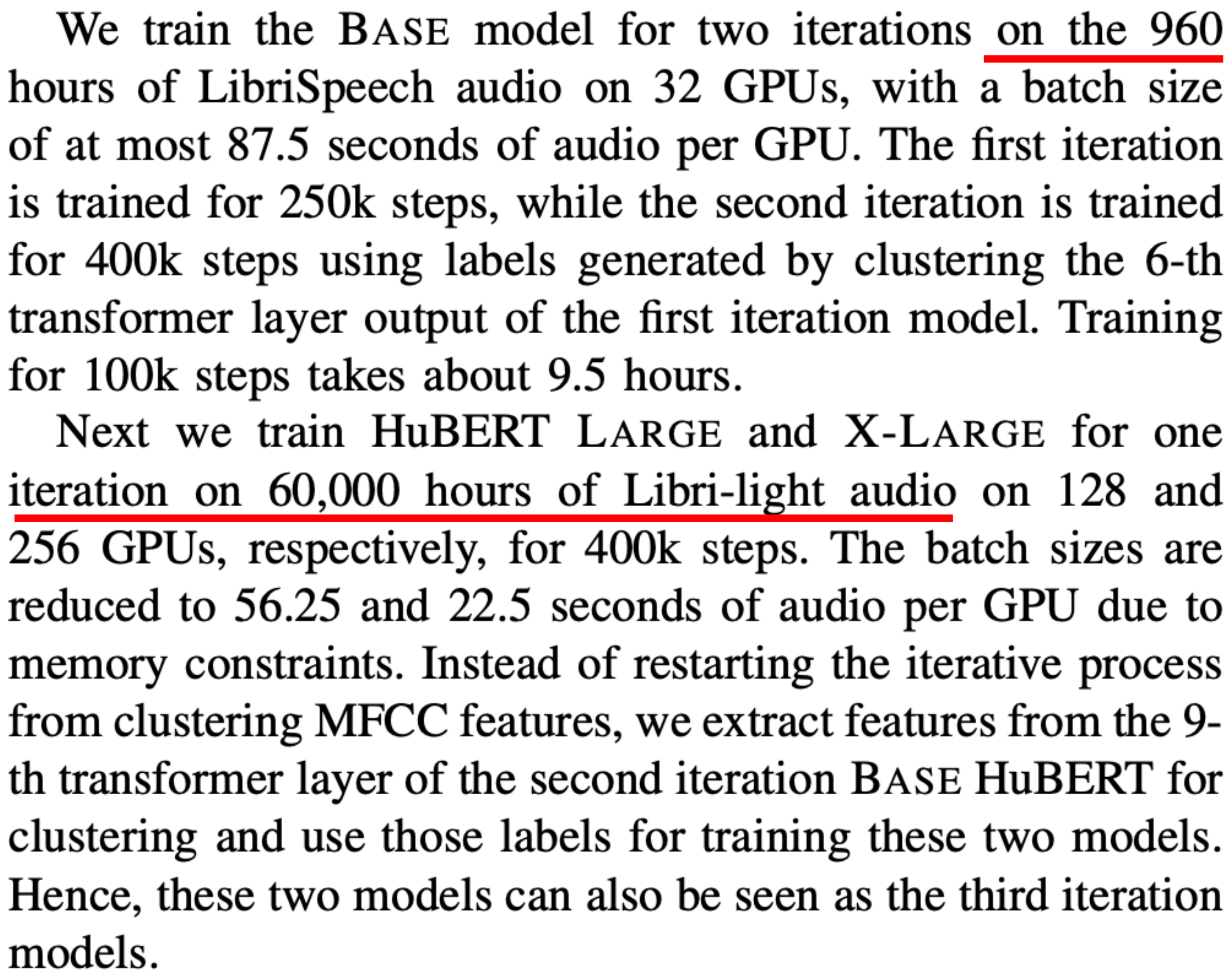
10% data on 768-D HuBERT features with 500 clusters.

Step3:

fine-tune.



Results:



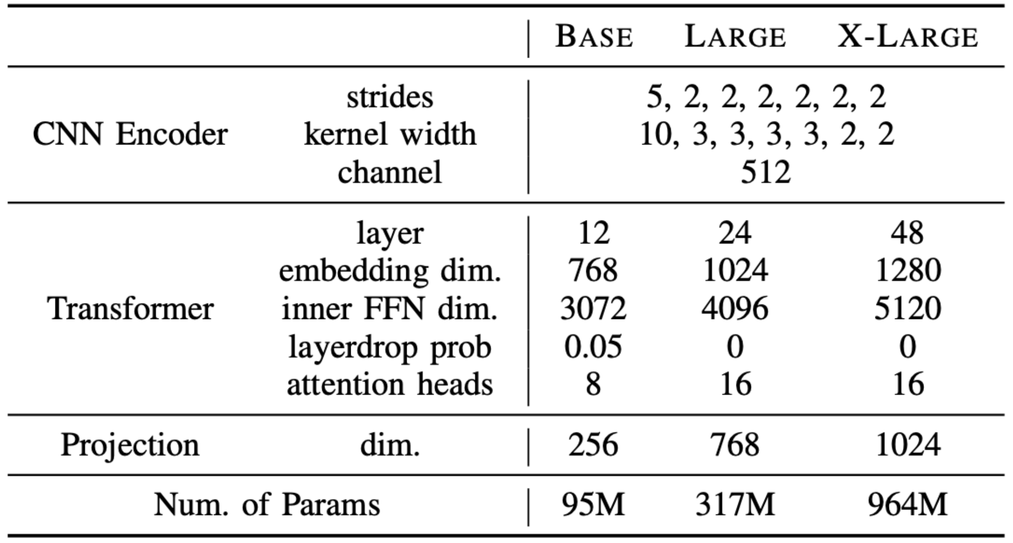
We train the BASE model for two iterations on the 960 hours of LibriSpeech audio on 32 GPUs, with a batch size of at most 87.5 seconds of audio per GPU. The first iteration is trained for 250k steps, while the second iteration is trained for 400k steps using labels generated by clustering the 6-th transformer layer output of the first iteration model. Training for 100k steps takes about 9.5 hours.

Next we train HuBERT LARGE and X-LARGE for one iteration on 60,000 hours of Libri-light audio on 128 and 256 GPUs, respectively, for 400k steps. The batch sizes are reduced to 56.25 and 22.5 seconds of audio per GPU due to memory constraints. Instead of restarting the iterative process from clustering MFCC features, we extract features from the 9-th transformer layer of the second iteration BASE HuBERT for clustering and use those labels for training these two models. Hence, these two models can also be seen as the third iteration models.

Sample rate: 16KHZ

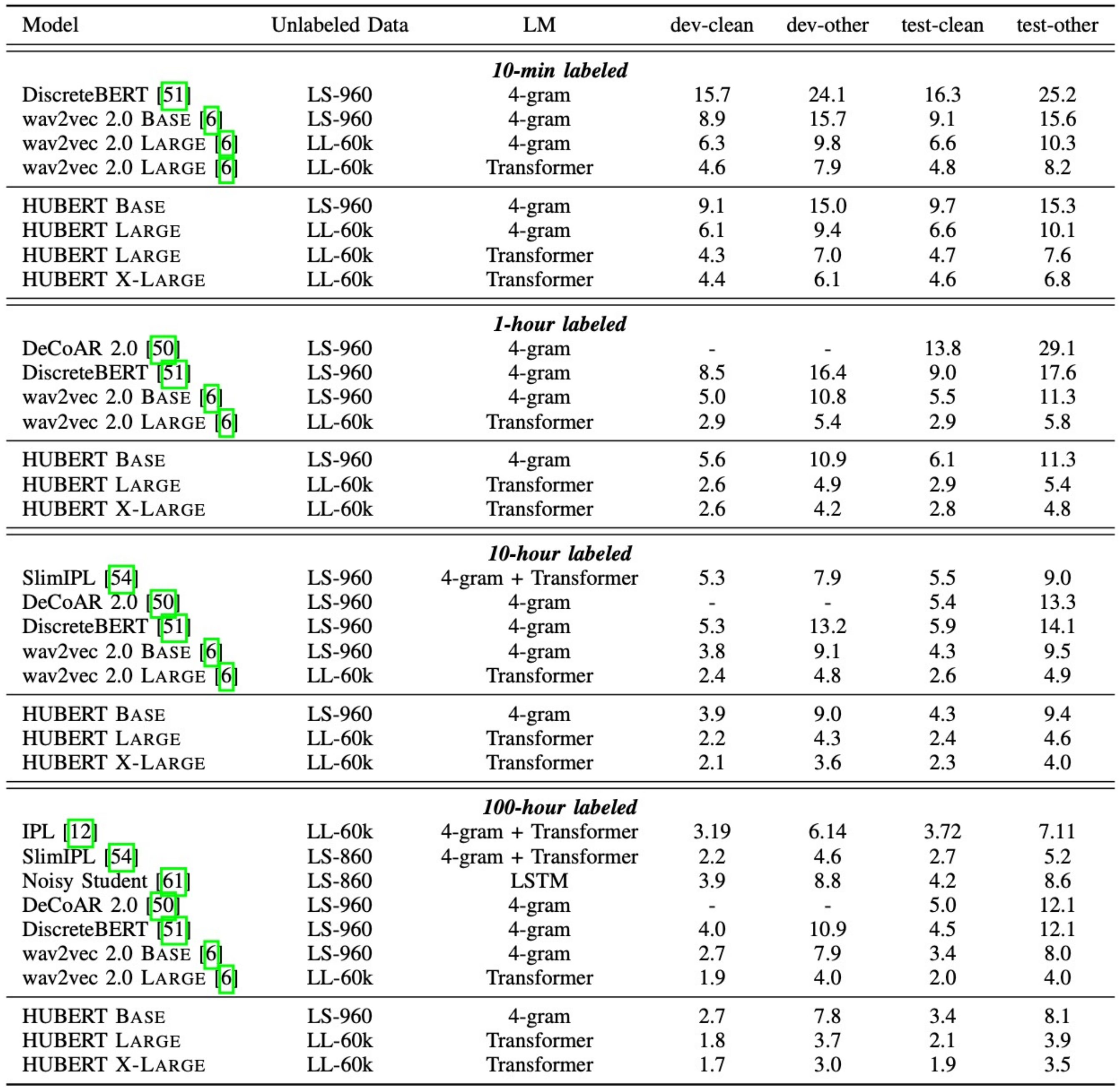
16000*60*60*960= 5.5296e+10

400000*256 = 1.024e+8



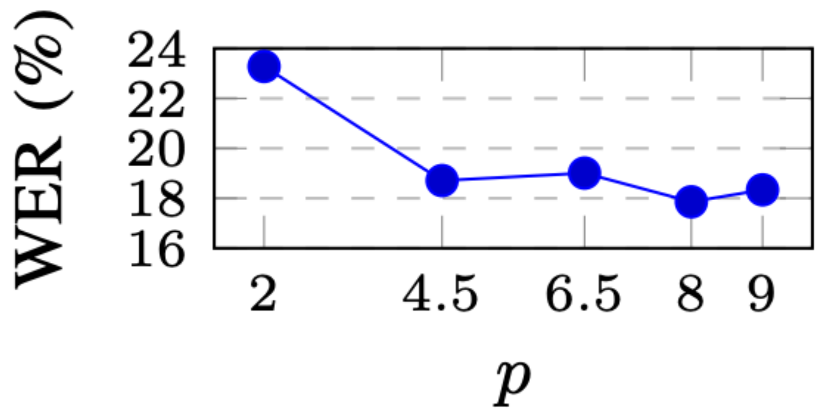
		BASE	LARGE	X-LARGE
CNN Encoder	strides	5, 2, 2, 2, 2, 2, 2		
	kernel width	10, 3, 3, 3, 3, 2, 2		
	channel	512		
Transformer	layer	12	24	48
	embedding dim.	768	1024	1280
	inner FFN dim.	3072	4096	5120
	layerdrop prob	0.05	0	0
	attention heads	8	16	16
Projection	dim.	256	768	1024
Num. of Params		95M	317M	964M

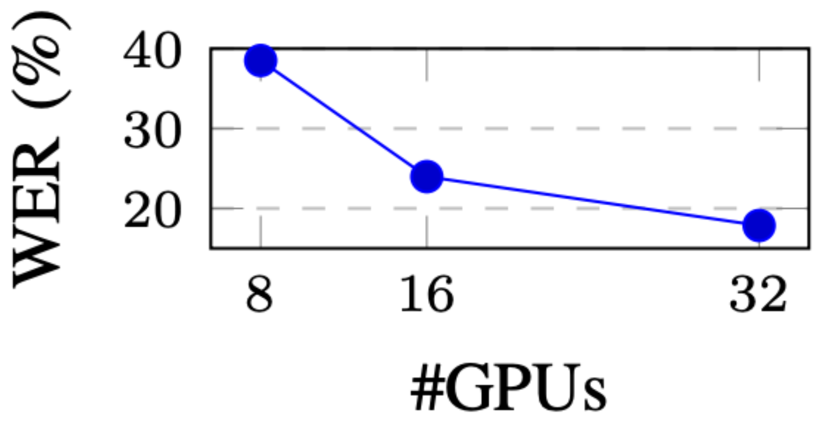
Results:



Model	Unlabeled Data	LM	dev-clean	dev-other	test-clean	test-other
		10-min labeled				
DiscreteBERT [51]	LS-960	4-gram	15.7	24.1	16.3	25.2
wav2vec 2.0 BASE [6]	LS-960	4-gram	8.9	15.7	9.1	15.6
wav2vec 2.0 LARGE [6]	LL-60k	4-gram	6.3	9.8	6.6	10.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	4.6	7.9	4.8	8.2
HUBERT BASE	LS-960	4-gram	9.1	15.0	9.7	15.3
HUBERT LARGE	LL-60k	4-gram	6.1	9.4	6.6	10.1
HUBERT LARGE	LL-60k	Transformer	4.3	7.0	4.7	7.6
HUBERT X-LARGE	LL-60k	Transformer	4.4	6.1	4.6	6.8
		1-hour labeled				
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	13.8	29.1
DiscreteBERT [51]	LS-960	4-gram	8.5	16.4	9.0	17.6
wav2vec 2.0 BASE [6]	LS-960	4-gram	5.0	10.8	5.5	11.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.9	5.4	2.9	5.8
HUBERT BASE	LS-960	4-gram	5.6	10.9	6.1	11.3
HUBERT LARGE	LL-60k	Transformer	2.6	4.9	2.9	5.4
HUBERT X-LARGE	LL-60k	Transformer	2.6	4.2	2.8	4.8
		10-hour labeled				
SlimIPL [54]	LS-960	4-gram + Transformer	5.3	7.9	5.5	9.0
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.4	13.3
DiscreteBERT [51]	LS-960	4-gram	5.3	13.2	5.9	14.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	3.8	9.1	4.3	9.5
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.4	4.8	2.6	4.9
HUBERT BASE	LS-960	4-gram	3.9	9.0	4.3	9.4
HUBERT LARGE	LL-60k	Transformer	2.2	4.3	2.4	4.6
HUBERT X-LARGE	LL-60k	Transformer	2.1	3.6	2.3	4.0
		100-hour labeled				
IPL [12]	LL-60k	4-gram + Transformer	3.19	6.14	3.72	7.11
SlimIPL [54]	LS-860	4-gram + Transformer	2.2	4.6	2.7	5.2
Noisy Student [61]	LS-860	LSTM	3.9	8.8	4.2	8.6
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.0	12.1
DiscreteBERT [51]	LS-960	4-gram	4.0	10.9	4.5	12.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	2.7	7.9	3.4	8.0
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	1.9	4.0	2.0	4.0
HUBERT BASE	LS-960	4-gram	2.7	7.8	3.4	8.1
HUBERT LARGE	LL-60k	Transformer	1.8	3.7	2.1	3.9
HUBERT X-LARGE	LL-60k	Transformer	1.7	3.0	1.9	3.5

Results:





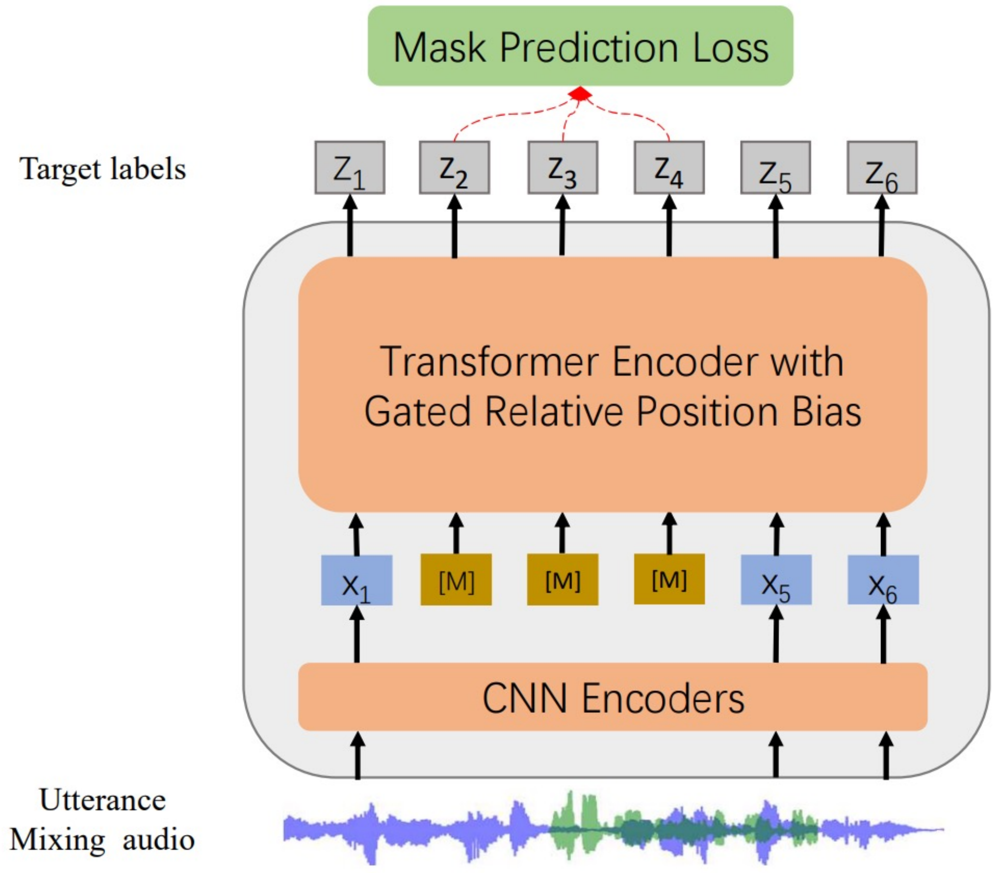
WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing

Sanyuan Chen*, Chengyi Wang*, Zhengyang Chen*, Yu Wu*, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, Furu Wei

Microsoft

Improves:

1. We scale up unlabeled pre-training data to 94k hours of public audios.
2. WavLM sheds light on a general pre-trained model for full stack speech processing tasks.
3. WavLM jointly learns masked speech prediction and denoising in pre-training.



- **Speaker verification** is a task to verify the speaker' s identity from the voice characteristics.
- **Speech separation** is a classic multi-speaker task, which is the key to solving the cocktail party problem.
- **Speaker diarization** is a task to recognize "who spoke when" from an input audio stream .
- **Speech recognition** .

Improves:

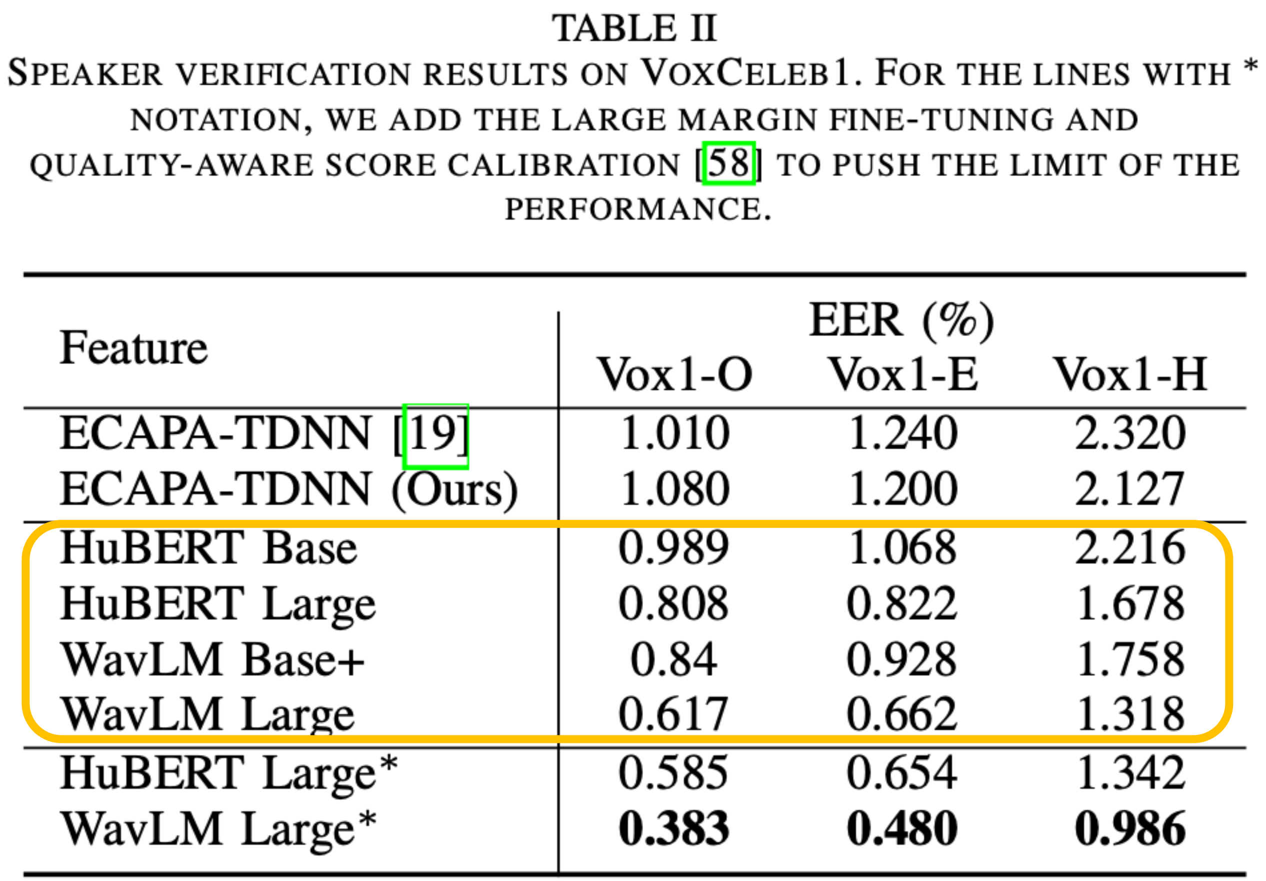
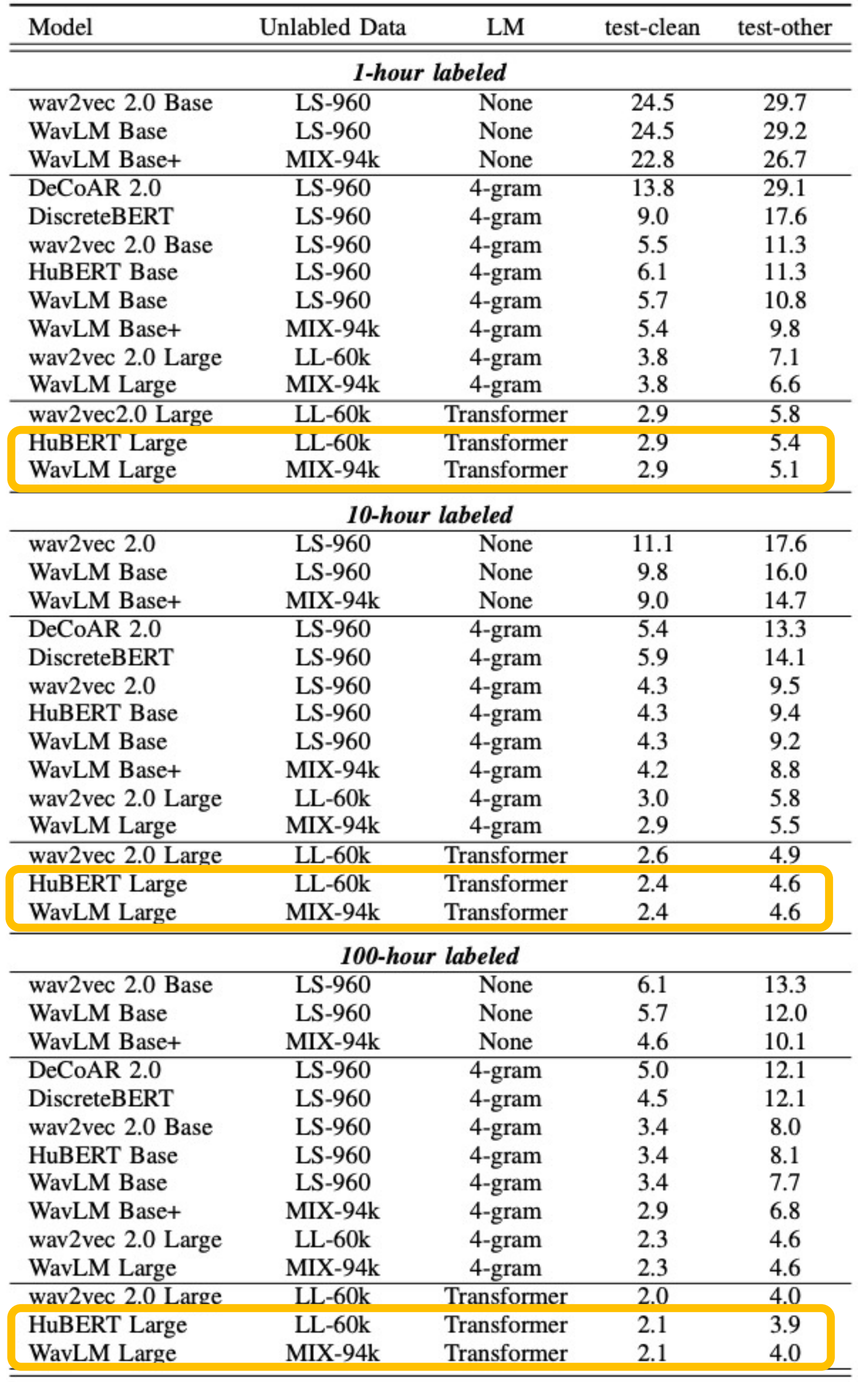


TABLE II
SPEAKER VERIFICATION RESULTS ON VOXCELEB1. FOR THE LINES WITH * NOTATION, WE ADD THE LARGE MARGIN FINE-TUNING AND QUALITY-AWARE SCORE CALIBRATION [58] TO PUSH THE LIMIT OF THE PERFORMANCE.

Feature	EER (%)		
	Vox1-O	Vox1-E	Vox1-H
ECAPA-TDNN [19]	1.010	1.240	2.320
ECAPA-TDNN (Ours)	1.080	1.200	2.127
HuBERT Base	0.989	1.068	2.216
HuBERT Large	0.808	0.822	1.678
WavLM Base+	0.84	0.928	1.758
WavLM Large	0.617	0.662	1.318
HuBERT Large*	0.585	0.654	1.342
WavLM Large*	**0.383**	**0.480**	**0.986**



Model	Unlabled Data	LM	test-clean	test-other
1-hour labeled				
wav2vec 2.0 Base	LS-960	None	24.5	29.7
WavLM Base	LS-960	None	24.5	29.2
WavLM Base+	MIX-94k	None	22.8	26.7
DeCoAR 2.0	LS-960	4-gram	13.8	29.1
DiscreteBERT	LS-960	4-gram	9.0	17.6
wav2vec 2.0 Base	LS-960	4-gram	5.5	11.3
HuBERT Base	LS-960	4-gram	6.1	11.3
WavLM Base	LS-960	4-gram	5.7	10.8
WavLM Base+	MIX-94k	4-gram	5.4	9.8
wav2vec 2.0 Large	LL-60k	4-gram	3.8	7.1
WavLM Large	MIX-94k	4-gram	3.8	6.6
wav2vec2.0 Large	LL-60k	Transformer	2.9	5.8
HuBERT Large	LL-60k	Transformer	2.9	5.4
WavLM Large	MIX-94k	Transformer	2.9	5.1
10-hour labeled				
wav2vec 2.0	LS-960	None	11.1	17.6
WavLM Base	LS-960	None	9.8	16.0
WavLM Base+	MIX-94k	None	9.0	14.7
DeCoAR 2.0	LS-960	4-gram	5.4	13.3
DiscreteBERT	LS-960	4-gram	5.9	14.1
wav2vec 2.0	LS-960	4-gram	4.3	9.5
HuBERT Base	LS-960	4-gram	4.3	9.4
WavLM Base	LS-960	4-gram	4.3	9.2
WavLM Base+	MIX-94k	4-gram	4.2	8.8
wav2vec 2.0 Large	LL-60k	4-gram	3.0	5.8
WavLM Large	MIX-94k	4-gram	2.9	5.5
wav2vec 2.0 Large	LL-60k	Transformer	2.6	4.9
HuBERT Large	LL-60k	Transformer	2.4	4.6
WavLM Large	MIX-94k	Transformer	2.4	4.6
100-hour labeled				
wav2vec 2.0 Base	LS-960	None	6.1	13.3
WavLM Base	LS-960	None	5.7	12.0
WavLM Base+	MIX-94k	None	4.6	10.1
DeCoAR 2.0	LS-960	4-gram	5.0	12.1
DiscreteBERT	LS-960	4-gram	4.5	12.1
wav2vec 2.0 Base	LS-960	4-gram	3.4	8.0
HuBERT Base	LS-960	4-gram	3.4	8.1
WavLM Base	LS-960	4-gram	3.4	7.7
WavLM Base+	MIX-94k	4-gram	2.9	6.8
wav2vec 2.0 Large	LL-60k	4-gram	2.3	4.6
WavLM Large	MIX-94k	4-gram	2.3	4.6
wav2vec 2.0 Large	LL-60k	Transformer	2.0	4.0
HuBERT Large	LL-60k	Transformer	2.1	3.9
WavLM Large	MIX-94k	Transformer	2.1	4.0

Noisy/Overlapped Speech Simulation:

Algorithm 1 Noisy/Overlapped Speech Simulation

```
1:  given a batch of speech utterances $\mathbf{U} = \{\mathbf{u}^i\}_{i=1}^{B}$ with batch size $B$ and length $L$, mixing probability $p$, a set of DNS noises $\mathbf{N} = \{\mathbf{n}^i\}_{i=1}^{M}$ with size $M$, mixing noise probability $p_n$
2:  Choose $S$ utterances $\mathbf{U}^S \subset \mathbf{U}$ by Bernoulli sampling with probability $p$
3:  for each primary utterance $\mathbf{u}^{\text{pri}} \in \mathbf{U}^S$ do
4:      Sample a random value $v$ from the continuous uniform distribution $\mathcal{U}(0, 1)$
5:      if $v > p_n$ then
6:          Sample a secondary utterance $\mathbf{u}^{\text{sec}}$ from discrete uniform distribution with probability $P(\mathbf{u}^{\text{sec}} = \mathbf{x}) = \frac{1}{B}, \mathbf{x} \in \mathbf{U}$
7:          Sample the mixing energy ratio $r$ from the continuous uniform distribution $\mathcal{U}(-5, 5)$
8:      else
9:          Sample a noise $\mathbf{u}^{\text{sec}}$ from discrete uniform distribution with probability $P(\mathbf{u}^{\text{sec}} = \mathbf{x}) = \frac{1}{M}, \mathbf{x} \in \mathbf{N}$
10:         Sample the mixing energy ratio $r$ from the continuous uniform distribution $\mathcal{U}(-5, 20)$
11:     end if
12:     Sample the mix length $l$ from discrete uniform distribution with probability $P(l = x) = \frac{2}{L}, x \in \{1, \cdots, \frac{L}{2}\}$
13:     Sample a start position $s^{\text{pri}}$ of $\mathbf{u}^{\text{pri}}$ from discrete uniform distribution with probability $P(s^{\text{pri}} = x) = \frac{1}{L-l}, x \in \{1, \cdots, L - l\}$
14:     Sample a start position $s^{\text{sec}}$ of $\mathbf{u}^{\text{sec}}$ from discrete uniform distribution with probability $P(s^{\text{sec}} = x) = \frac{1}{L-l}, x \in \{1, \cdots, L - l\}$
15:     Calculate the energy of the primary utterance $E^{\text{pri}} \leftarrow \frac{\sum \mathbf{u}^{\text{pri}} \cdot \mathbf{u}^{\text{pri}}}{L}$
16:     Calculate the energy of the secondary utterance $E^{\text{sec}} \leftarrow \frac{\sum \mathbf{u}^{\text{sec}} \cdot \mathbf{u}^{\text{sec}}}{L}$
17:     Calculate the mixing scale $scl \leftarrow \sqrt{\frac{E^{\text{pri}}}{10^{\frac{r}{10}} E^{\text{sec}}}}$
18:     $\mathbf{u}^{\text{pri}}[s^{\text{pri}} : s^{\text{pri}} + l] \leftarrow \mathbf{u}^{\text{pri}}[s^{\text{pri}} : s^{\text{pri}} + l] + scl \cdot \mathbf{u}^{\text{sec}}[s^{\text{sec}} : s^{\text{sec}} + l]$
19: end for
20: return $\mathbf{U}$
```

TABLE II
SPEAKER VERIFICATION RESULTS ON VOXCELEB1. FOR THE LINES WITH * NOTATION, WE ADD THE LARGE MARGIN FINE-TUNING AND QUALITY-AWARE SCORE CALIBRATION [58] TO PUSH THE LIMIT OF THE PERFORMANCE.

Feature	EER (%)		
	Vox1-O	Vox1-E	Vox1-H
ECAPA-TDNN [19]	1.010	1.240	2.320
ECAPA-TDNN (Ours)	1.080	1.200	2.127
HuBERT Base	0.989	1.068	2.216
HuBERT Large	0.808	0.822	1.678
WavLM Base+	0.84	0.928	1.758
WavLM Large	0.617	0.662	1.318
HuBERT Large*	0.585	0.654	1.342
WavLM Large*	**0.383**	**0.480**	**0.986**

TABLE IV
SEPARATION RESULTS ON LIBRICSS DATASET. WE FREEZE THE PRE-TRAINED PARAMETERS BY DEFAULT FOR THE SEPARATION TASK. THE RESULTS DENOTE %WER SCORE EVALUATED WITH E2E TRANSFORMER BASED ASR MODEL [69]. 0S AND 0L ARE UTTERANCES WITH SHORT/LONG INTER-UTTERANCE SILENCE. THE AVG IS THE WEIGHTED AVERAGED WER OF DIFFERENT OVERLAPPED TESTSETS.

System	Overlap ratio in %						
	0S	0L	10	20	30	40	avg
Conformer [22]	5.4	5.0	7.5	10.7	13.8	17.1	10.6
Conformer (rerun)	4.5	4.4	6.2	8.5	11.0	12.6	8.3
HuBERT Base	4.7	4.6	6.1	7.9	10.6	12.3	8.1
WavLM Base+	4.5	4.4	5.6	7.5	9.4	10.9	7.4
- unfreeze pre-trained parameters	4.5	4.3	5.9	8.3	11.1	12.5	8.2
WavLM Large	**4.2**	**4.1**	**4.8**	**5.8**	**7.4**	**8.5**	**6.0**

Self-Supervised Learning with Random-Projection Quantizer for Speech Recognition

Chung-Cheng Chiu, James , Yu Zhang , Jiahui Yu, Yonghui Wu

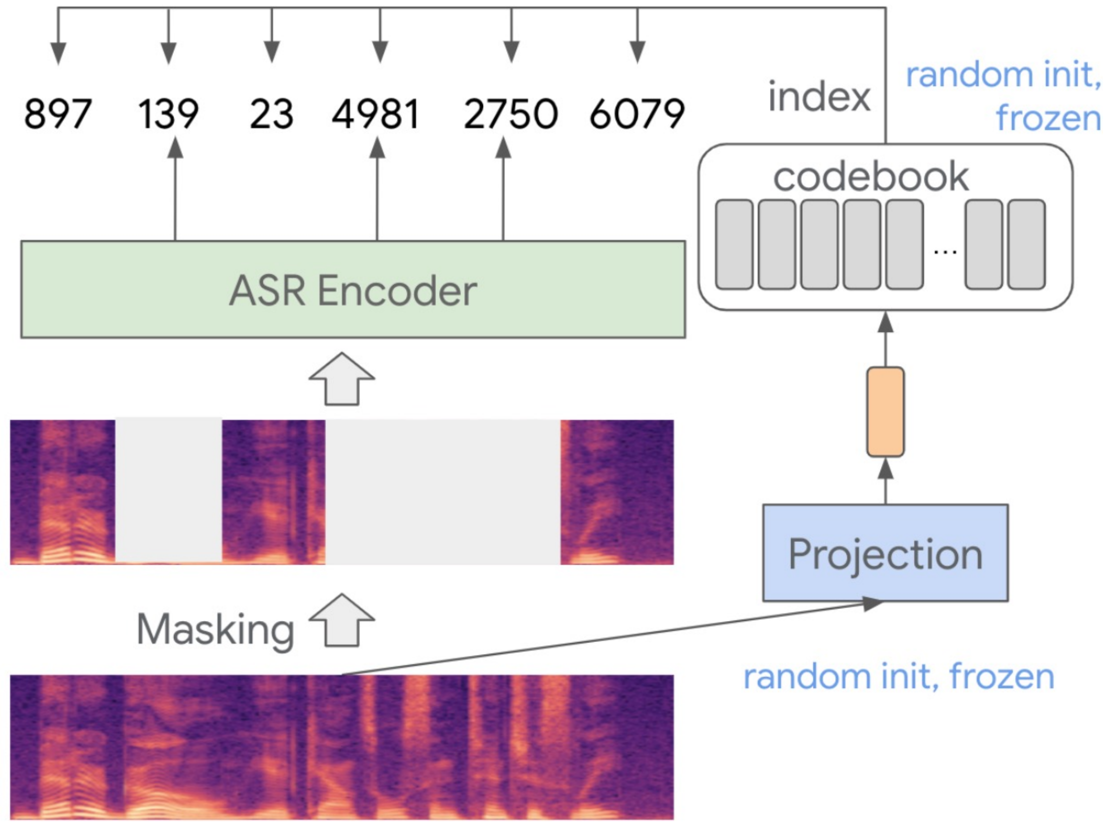
Google

The learning targets are labels provided by a random-projection quantizer.

The random projection quantizer projects speech signals to a randomly initialized matrix, and finds a nearest vector in a randomly initialized codebook. The index of that vector is the target label.

Neither the projection matrix nor the codebook is updated throughout the learning process.

$$y = \underset{i}{argmin} || norm_{l2}(c_i) - norm_{l2}(Ax) ||$$



- How good is the resulting quantization quality with this quantizer?
- How much does the quantization quality affect the effectiveness of the self-supervised learning?

We demonstrate that the quantization quality of the random projection quantizer is not ideal but yet effective for self-supervised learning.
We also show that the gap in the quantization quality is less an issue with the increase of the unsupervised data.

The random projection quantizer preserve the distribution of the speech data, and in order for the model to learn to predict the quantized token based on unmasked signals, the model needs to learn to process the raw signals and infer the contextual information among speech data.

Table 5. Quantizer quality's impact on ASR tasks. Although the Transformer-based quantizer gets much better performance when used as input directly, the random-projection quantizer is equally effective for self-supervised learning. The model used in the direct ASR task has size 25M. The self-supervised learning tasks use the same setup as the LibriSpeech non-streaming experiment, which use LibriLight for pre-training and LibriSpeech for fine-tuning and has 0.6B model size.

Configuration	Quantizer size (M)	Direct ASR WER				Pretrain-finetune WER			
		dev	dev-other	test	test-other	dev	dev-other	test	test-other
Random quantizer	1	58.8	78.8	57.9	72.8	1.5	**2.8**	**1.6**	**2.9**
Projection VQ-VAE	1	61.4	74.8	60.9	75.2	1.5	**2.8**	**1.6**	**2.9**
Transformer VQ-VAE	10	**17.8**	**35.8**	**17.6**	**36.1**	**1.4**	2.9	**1.6**	3.1

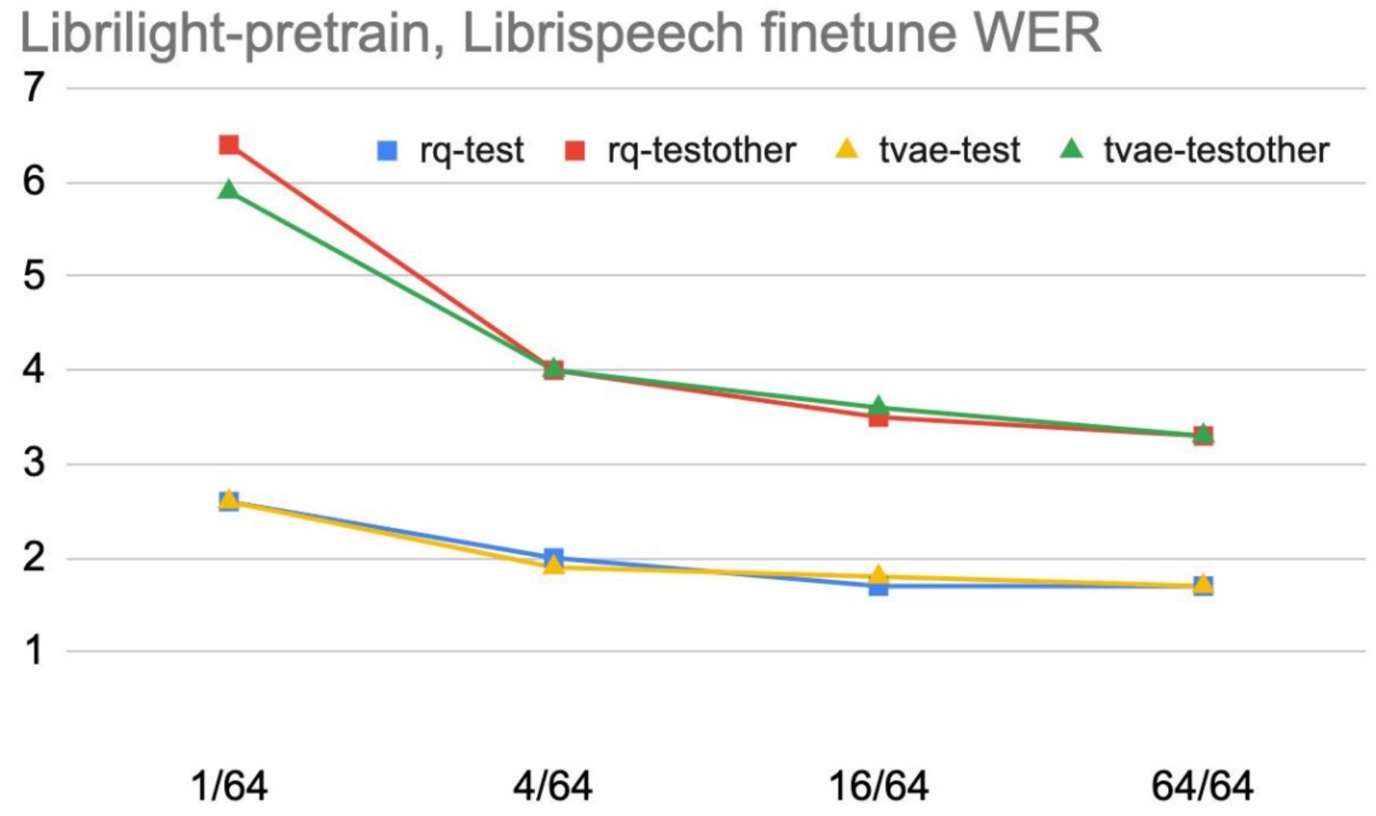


Figure 2. Comparing the self-supervised learning quality of the random-projection quantizer (rq) and the Transformer-based VQ-VAE quantizer (tvae) with different pre-training data size. Starting from low amount of pre-train data, the random-projection quantizer is behind the trained Transformer VQ-VAE quantizer. As the amount of pre-train data increases, the random-projection quantizer catches up.

The End.