In the name of GOD

SELF-SUPERVISED SPEECH MODELS

Presented By : Maryam Afshari

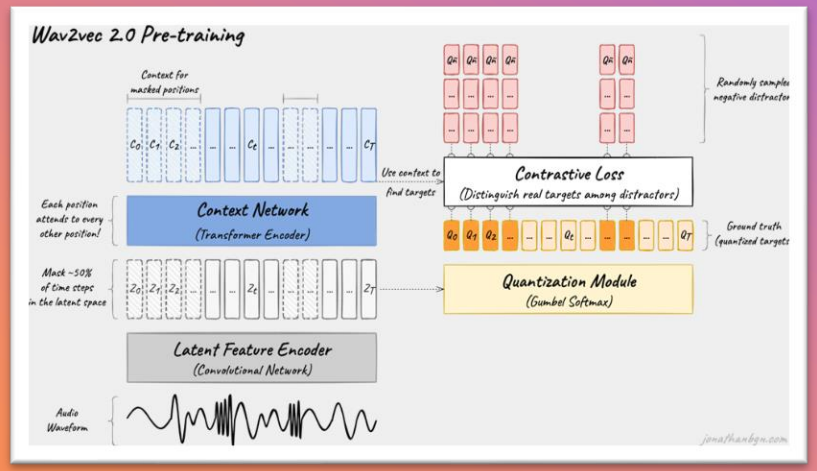
Supervisor : Dr. Hossein Sameti

Advisor : Dr. Hossein Zeinali

Lab : Sharif SPL

12/12/2022

TOPICS :



1. The Rise of Self-Supervised Learning
2. Wav2vec 2.0 Model
3. HuBERT Model
4. Realistic Brain Speech Processing with SSL

THE RISE OF SSL LEARNING

The Rise of Self-Supervised Learning

- Since the deep learning wave started in the early 201
- a big part of this is due to **high expectations driven b progress that do not translate so well in real-world**
- it has seen a resurgence of interest over the past few to mediatized successes like **GPT-3 or BERT**.

- Facebook's AI chief Yann Lecun with his cake analogy

How Much Information is the Machine Given during Learning?

Y. LeCun

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

What is self-supervised learning?

- Today's deep learning success is mostly about **labeled** data
  - ImageNet (Alex Net (2012) »» Le-net5-(1998))
- Problem: Labels don't scale
  - large clean labeled datasets like ImageNet
  - **real-world applications is incredibly messy, and labels are extremely scarce**.
  - **labeling data can be extremely costly and time-consuming**
  - Labels are expensive. Fortunately, unlabeled data is free!

Reducing the need for human labels

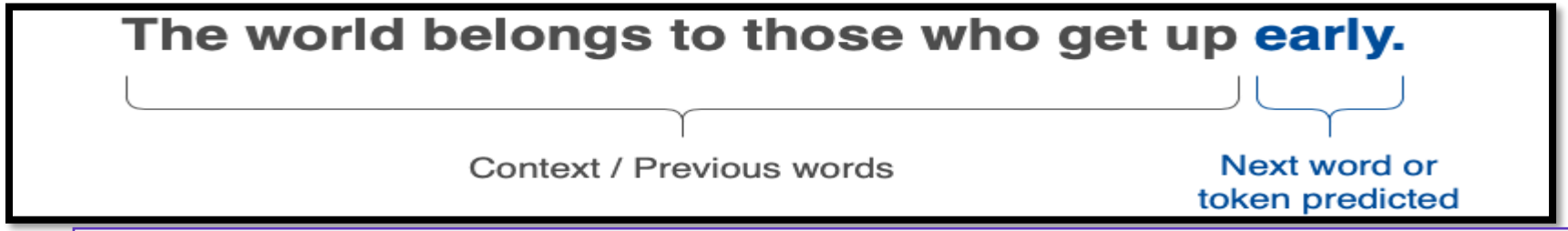
- **Self-supervised learning allows training on unlabeled data**, thanks to **automatic label generation.**
  - How to generate these labels then?
    - **use one part of the input as the label itself**.

  - Remove a word from a sentence, which becomes the target label
  - Hide one part of an image and let the model re-generate it
  - Ask the model to predict the next frame in a video

- **This is a game-changing paradigm that effectively shifts the bottleneck from data quantity to compute capacity.**

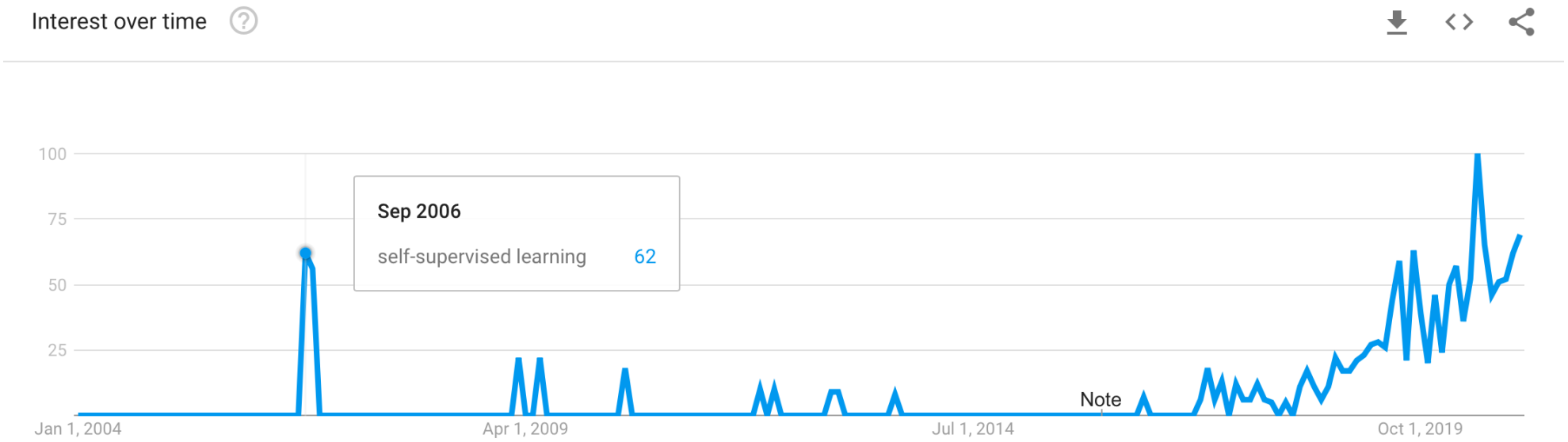
a model trained to predict the next word in a sentence.

- A good example is a language model: **GPT-3** by yourself with current cloud platform offerings would require 355 years and $4,600,000!



A language model like the GPT family predicts the next token based on what came before.

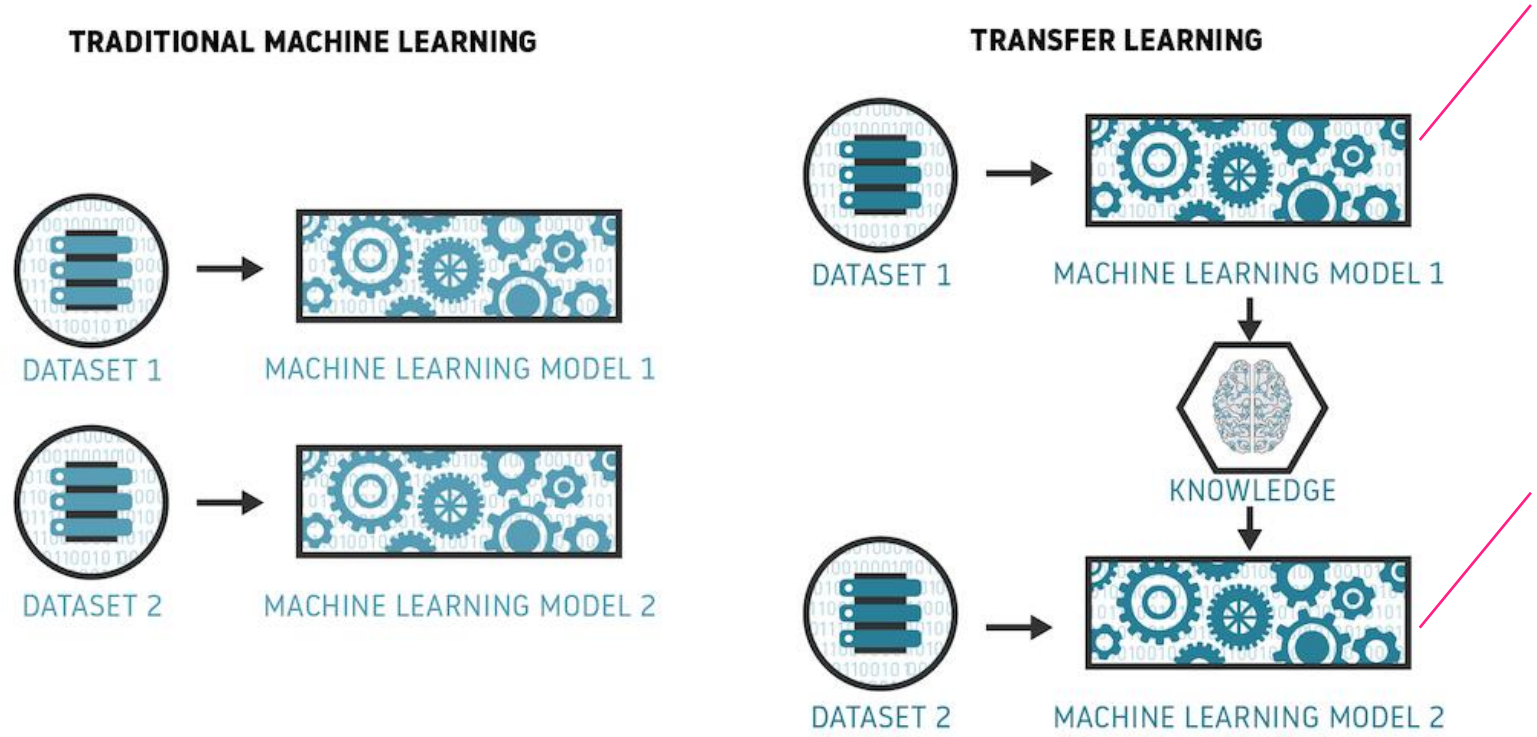
A very short history of self-supervised learning



Transfer knowledge to generalize better

True Power of self-supervised learning

a language model trained in a self-supervised way needs to learn about meaning and grammar to effectively predict the next word. This linguistic knowledge can be re-used in a downstream task like predicting the sentiment of a text



Pre-text Task (*auxiliary task*)

Downstream Task

An important point here is the choice of the pretext task.
This task must make sense such that solving it must be in some way useful to the downstream task.

Improve performance on small datasets

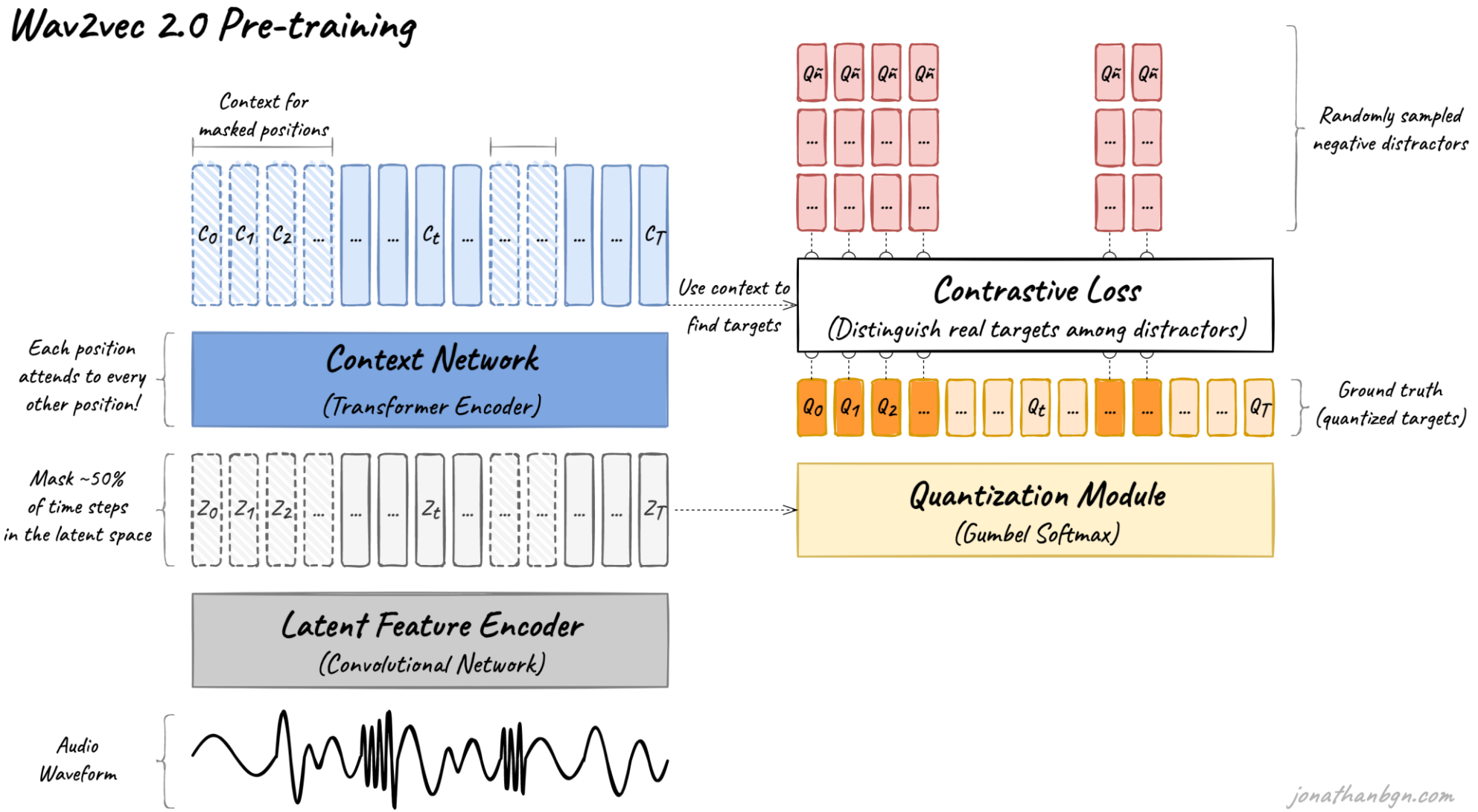
While we still need some domain-specific labeled data for the second fine-tuning step, pre-training first in a self-supervised way has been shown to immensely improve the performance on the downstream task, even when very few labeled data is available for fine-tuning. For example, the ULMFiT paper showed that it is possible to reach great performance for text classification with only 100 labeled examples. More recently, a new paper from DeepMind outperformed the original AlexNet performance on ImageNet with **only 13 labeled examples per class**.

Self-supervised learning does not remove the need for labeled data, but it greatly reduces the need for it and makes it practical to deploy deep learning models for use cases in which labeled data is (very) limited.

WAV2VEC2.0

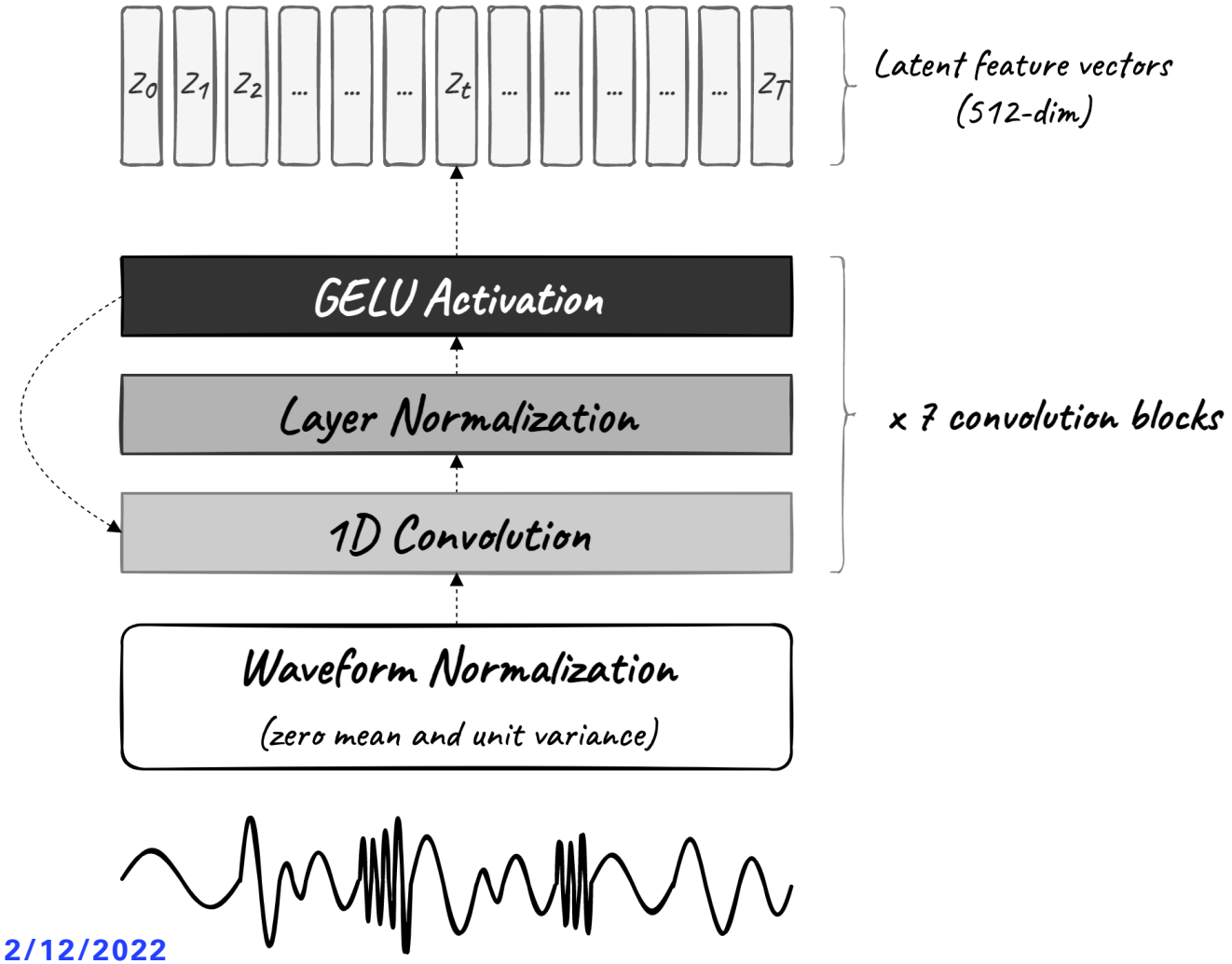
Model

Pre-training



Feature encoder

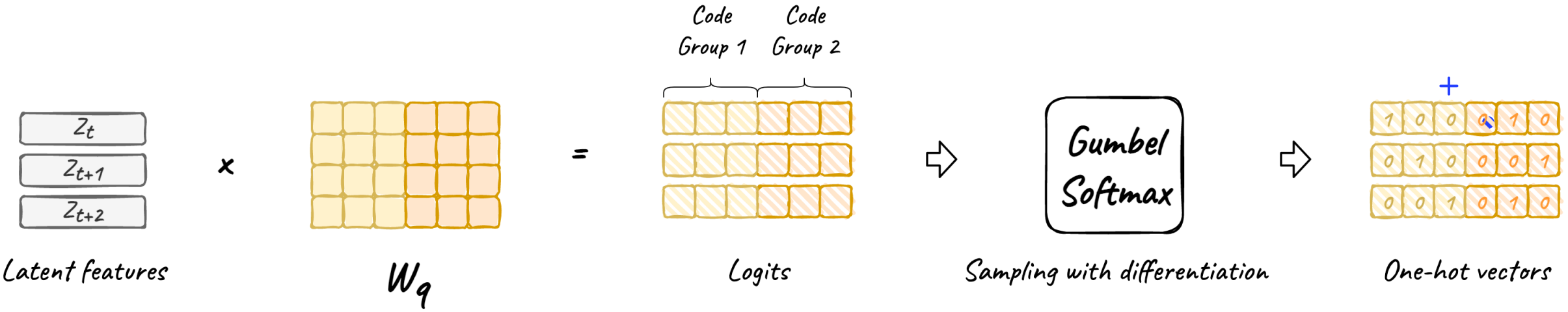
Wav2vec 2.0 Latent Feature Encoder

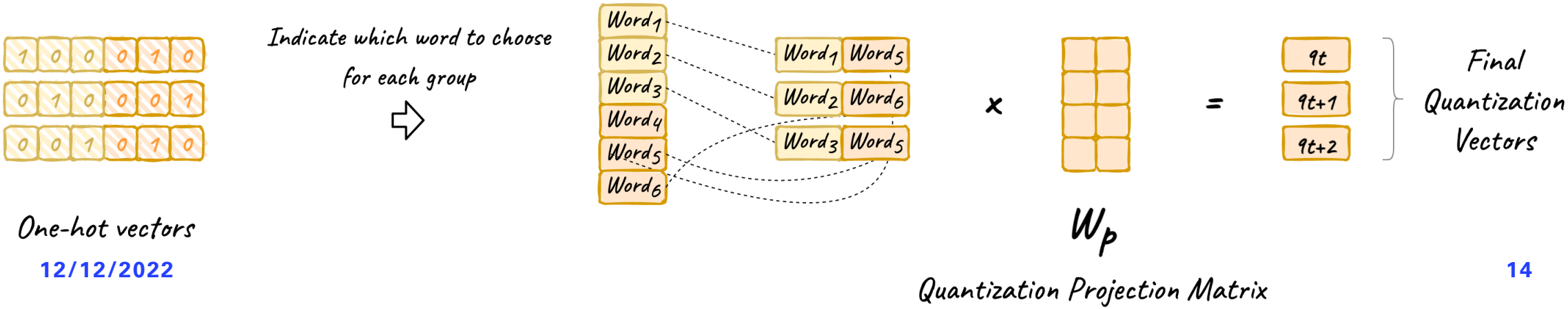


Block	Channels	Kernel width	Stride
7	512	2	2
6			
5		3	
4			
3			
2			
1		10	5

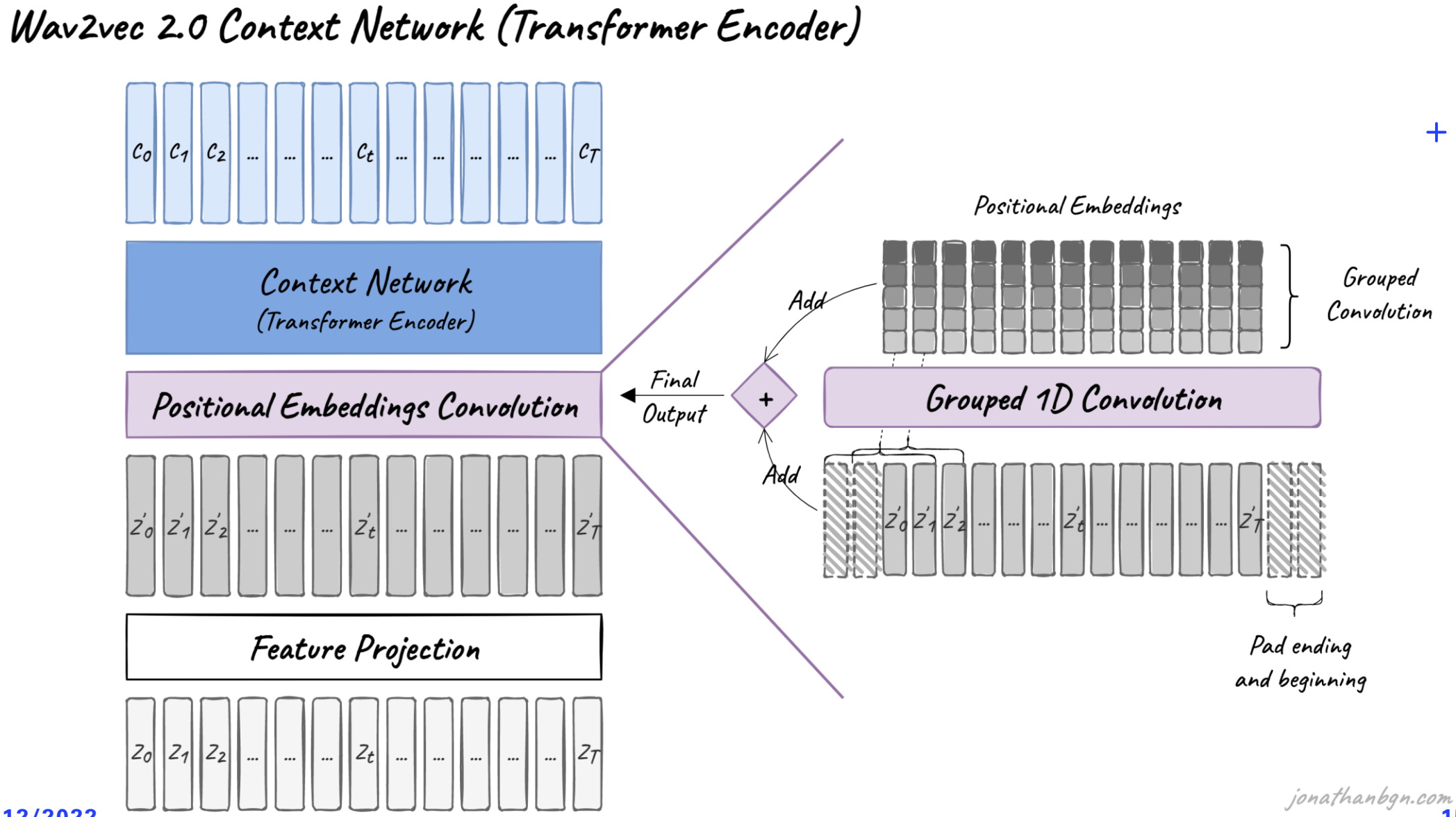
Quantization module

Wav2vec 2.0 Quantization Module



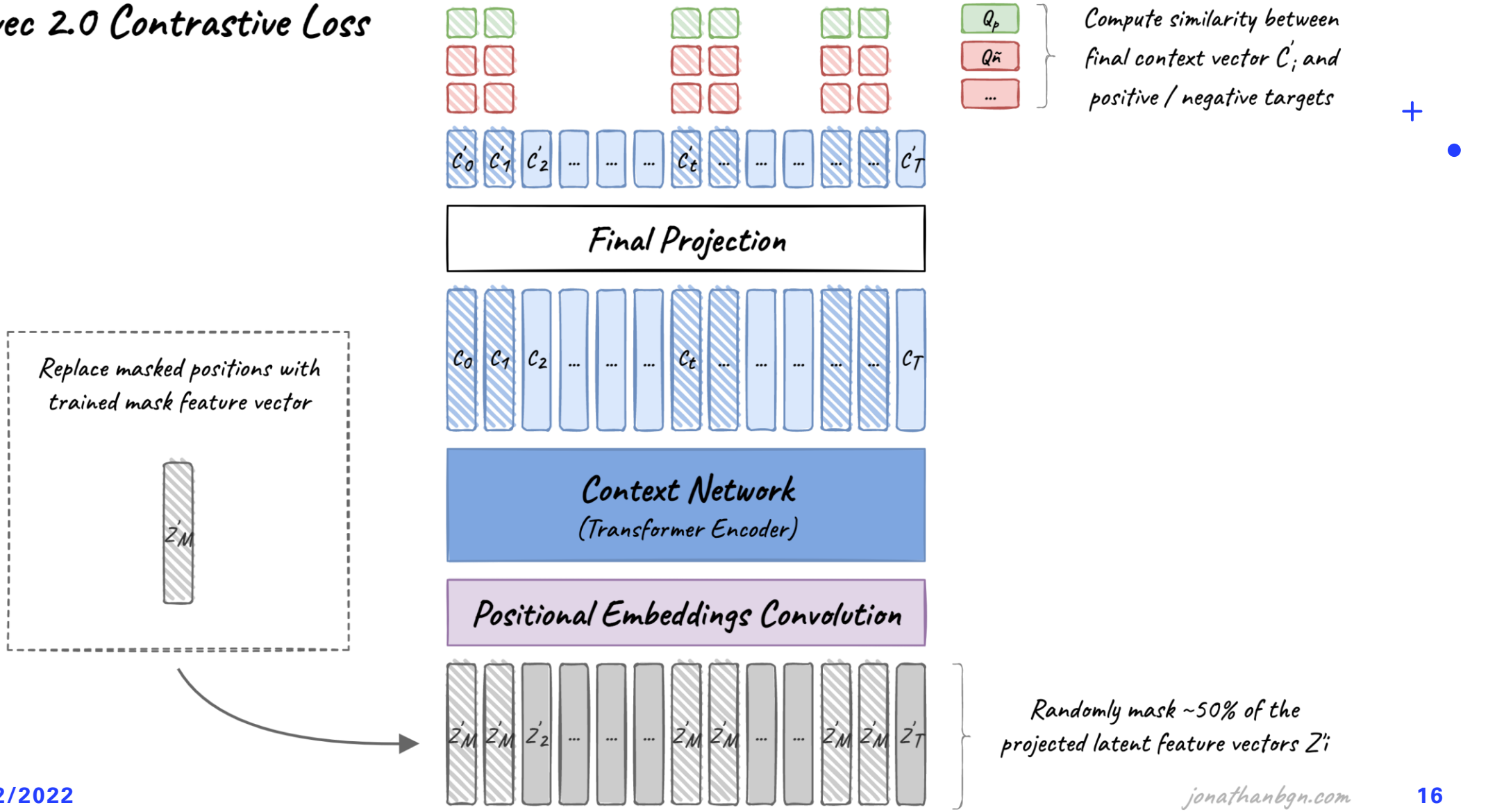


Context network



Pre-training & contrastive loss

Wav2vec 2.0 Contrastive Loss



Diversity loss

During pre-training, another loss is added to the contrastive loss to encourage the model to use all codewords equally often. This works by maximizing the entropy of the Gumbel-Softmax distribution, preventing the model to always choose from a small sub-group of all available codebook entries.

When lowering the amount of labeled data to one hour, wav2vec 2.0 outperforms the previous state of the art on the 100 hour subset while using 100 times less labeled data.

Wav2vec2.0 Summary

composed of **seven blocks** of temporal **convolutions** (output dimension 512)

transforms the **speech input S** (raw mono waveform at 16 kHz) into a **latent representation z** (output dimension of 512, frequency 49 Hz, stride of 20 ms between each frame, receptive field of 25 ms).

Feature encoder

discretizes z into q,

a dictionary of discrete and latent representations of sounds.

Quantization module

z is input to a

“context network”

consisting of 12 transformer blocks (model dimension 768, inner dimension 3072, and 8 attention heads), which together yield a contextualized embedding c, of the same dimensionality of q.

.

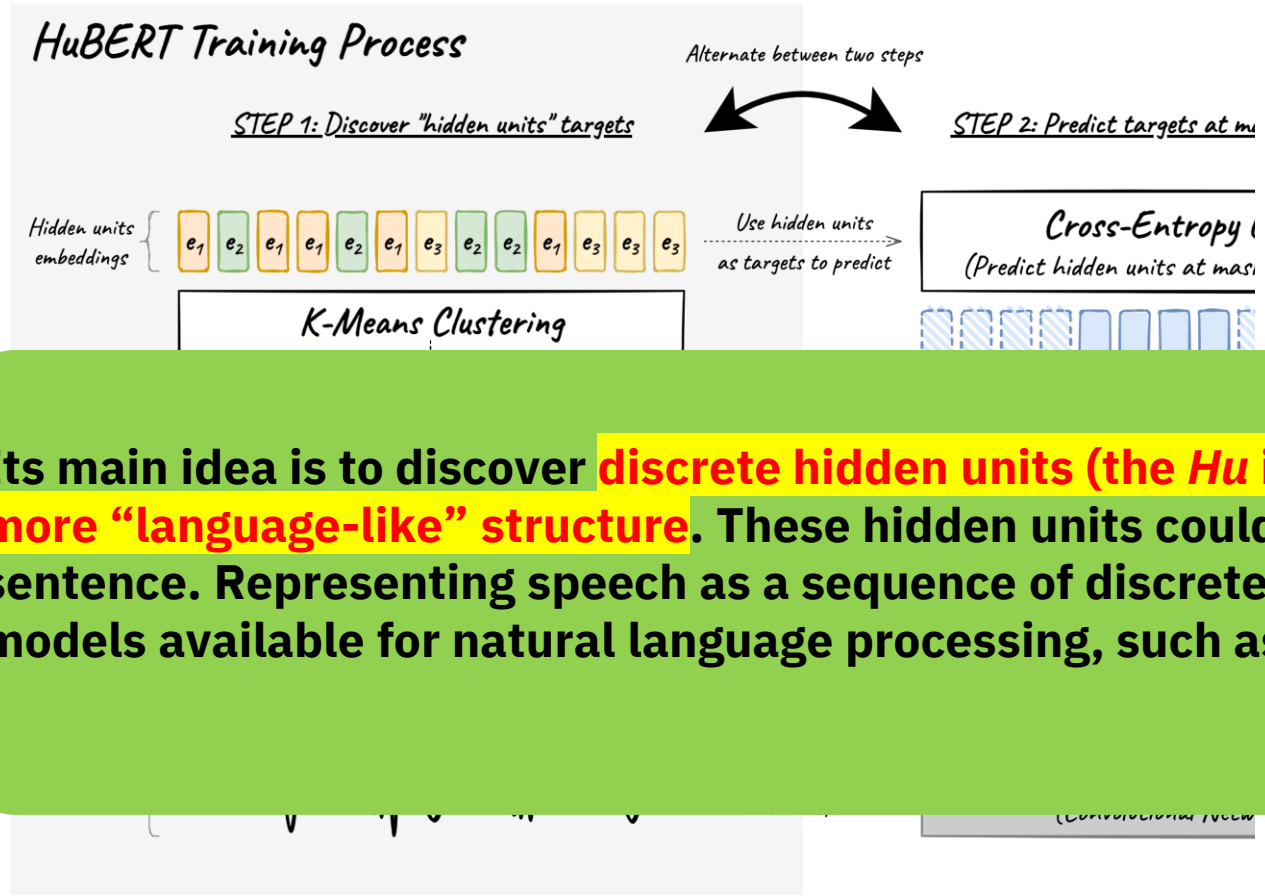
Context network

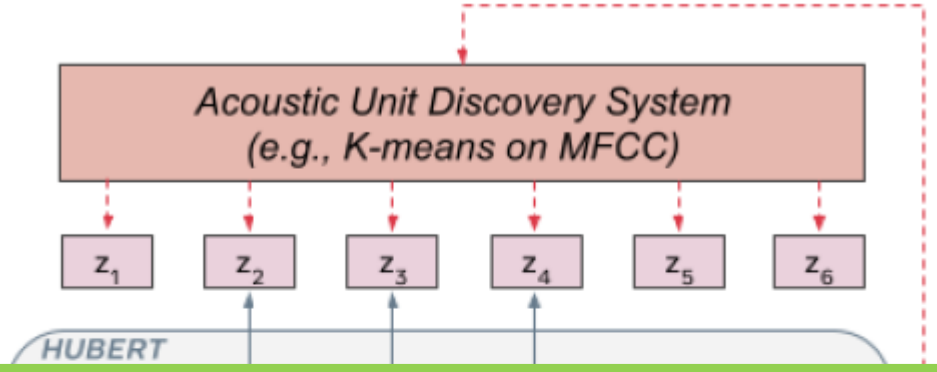
HUBERT

Model

Hubert:

How to Apply BERT to Speech





Its main idea is to discover discrete hidden units (the *Hu* in the name) to transform speech data into a more "language-like" structure. These hidden units could be compared to words or tokens in a text sentence. Representing speech as a sequence of discrete units enables us to apply the same powerful models available for natural language processing, such as BERT.

ments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Hubert Training process:

1

Step 1: a clustering step

- to create pseudo-targets
- (Discover “hidden units” targets through clustering)

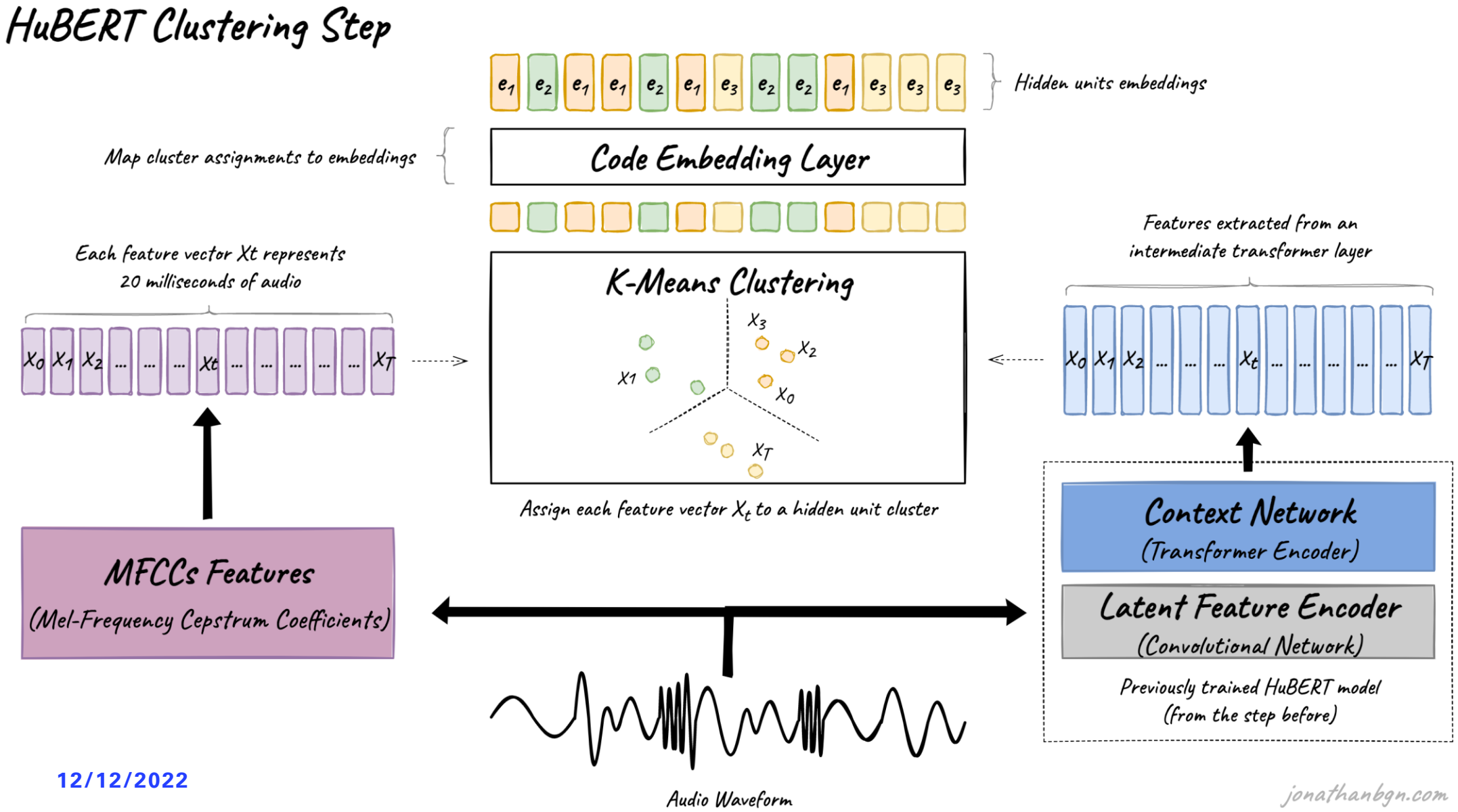
2

Step 2: prediction step

- where the model tries to guess these targets at masked positions

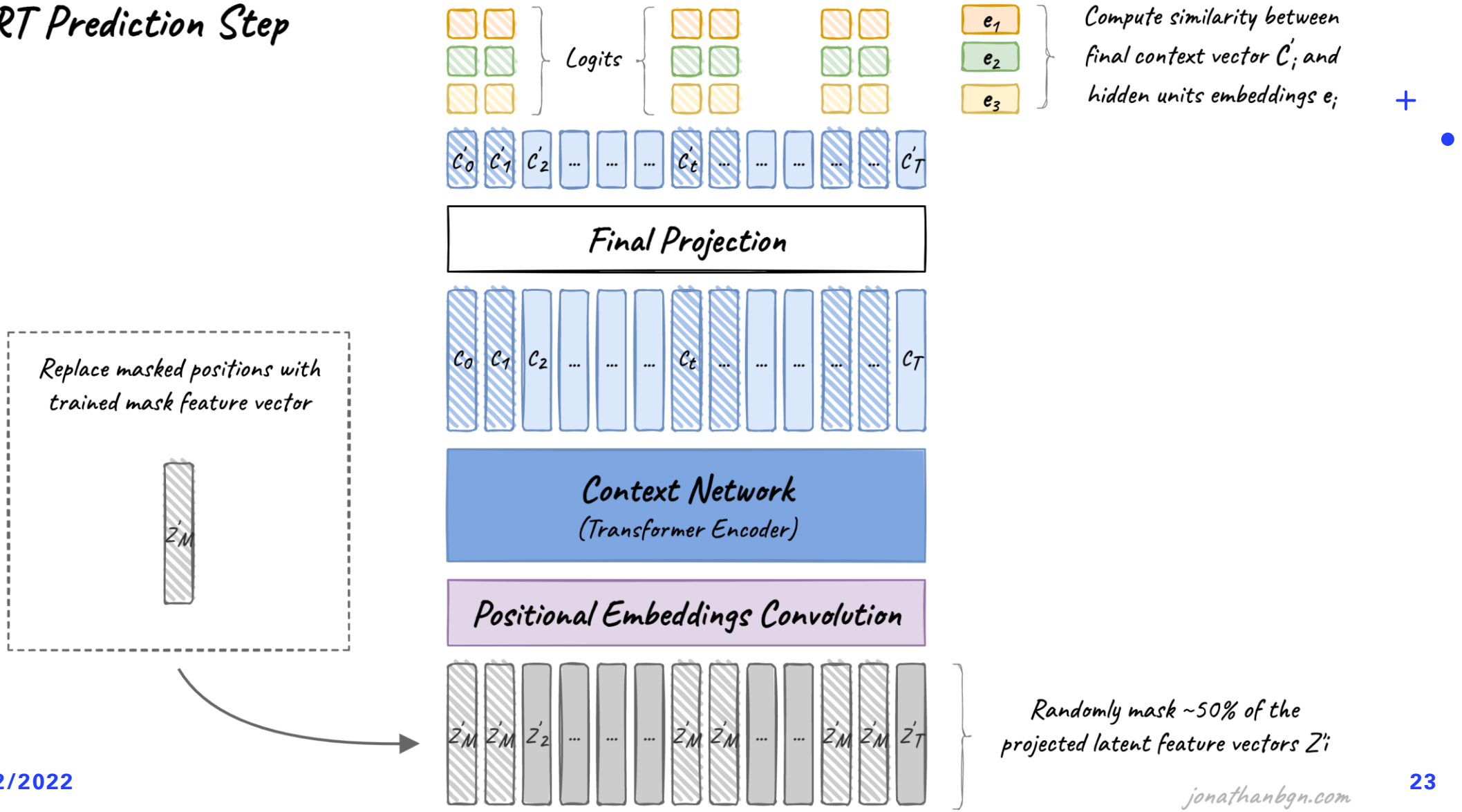
(Predict noisy targets from the context)

Clustering Step



Prediction Step

HuBERT Prediction Step



Differences with wav2vec 2.0

At first glance, HuBERT looks very similar to **wav2vec 2.0**: both models use the same **convolutional network followed by a transformer encoder**. However, their training processes are very different, and HuBERT's performance, when fine-tuned for automatic speech recognition, either matches or improves upon wav2vec 2.0.

In terms of model architecture, the BASE and LARGE versions of HuBERT have the same configuration as the BASE and LARGE versions of wav2vec 2.0 (95 million and 317 million of parameters respectively).

However, an X-LARGE version of HuBERT is also used with twice as many transformer layers as in the LARGE version, with almost 1 billion parameters.

Differences with wav2vec 2.0

HuBERT uses the cross-entropy loss

- Instead of the more complex combination of contrastive loss + diversity loss used by wav2vec 2.0.
- This makes training easier and more stable since this is the same loss that was used in the original BERT paper.

HuBERT builds targets via a separate clustering process

- while wav2vec 2.0 learns its targets simultaneously while training the model (via a **quantization process** using **Gumbel-softmax**).
- While wav2vec 2.0 training could seem simpler as it consists of only a single step, in practice, it can become more complex as the temperature of the Gumbel-softmax must be carefully adjusted during training to prevent the model from sticking to a small subset of all available targets

HuBERT re-uses embeddings from the BERT encoder to improve targets

- while wav2vec 2.0 only uses the output of the convolutional network for quantization.
- In the HuBERT paper, the authors show that using such embeddings from intermediate layers of the BERT encoder leads to better targets quality than using the CNN output.

WAVLM

Model

WavLM:

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

***Abstract*—Self-supervised learning (SSL) achieves great success in speech recognition, while limited exploration has been attempted for other speech processing tasks. As speech signal contains multi-faceted information including speaker identity, paralinguistics, spoken content, etc., learning universal representations for all speech tasks is challenging. To tackle the problem, we propose a new pre-trained model, WavLM, to solve full-stack downstream speech tasks. WavLM jointly learns masked speech prediction and denoising in pre-training. By this means, WavLM does not only keep the speech content modeling capability by the masked speech prediction, but also improves the potential to non-ASR tasks by the speech denoising. In addition, WavLM employs gated relative position bias for the Transformer structure to better capture the sequence ordering of input speech. We also scale up the training dataset from 60k hours to 94k hours. WavLM Large achieves state-of-the-art performance on the SUPERB benchmark, and brings significant improvements for various speech processing tasks on their representative benchmarks. The code and pre-trained models are available at https://aka.ms/wavlm.**

***Index Terms*—Self-Supervised Learning, Speech Pre-Training**

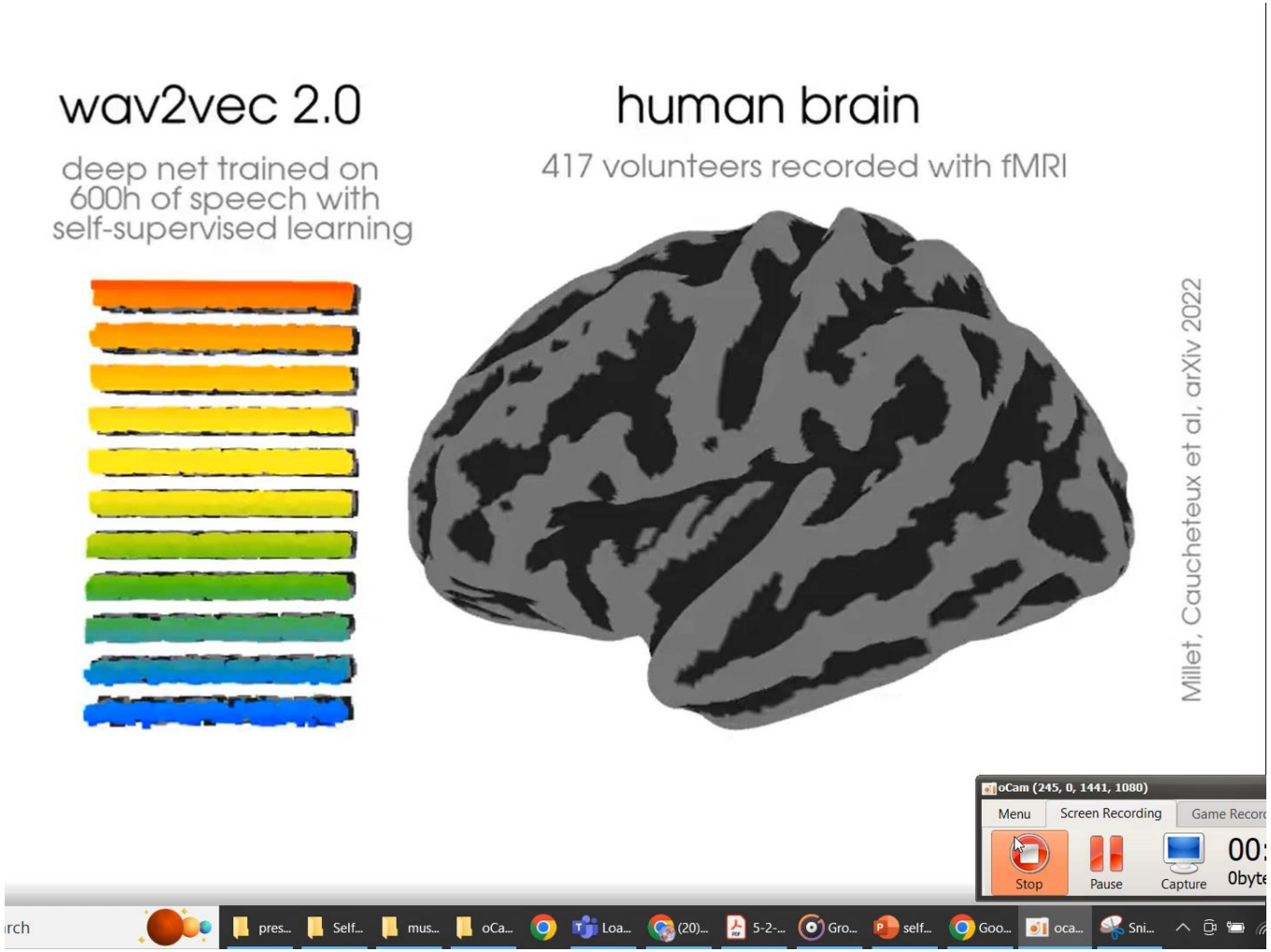
the network to learn the speaker characteristic regardless of the spoken content, while speech recognition demands the network to discard speaker characteristics and focus only on the content information. Meanwhile, unlike verification and recognition tasks, speaker diarization and speech separation involve multiple speakers, which creates additional obstacles to learning general speech representations. Recent advances fueled by large-scale pre-trained models have changed the situation. [8] proves the potential of pre-trained models on full-stack speech tasks by using the weighted sum of embeddings from different layers.[1] They find different layers containing information useful for different tasks. For instance, the hidden states of the top layers are useful for ASR, while the bottom layers are more effective for speaker verification.

While exciting as a proof of concept, there are still some drawbacks in existing pre-trained models: 1) Current pre-trained models are unsatisfactory for multi-speaker tasks, such as speaker diarization and speech separation. Our experiments show that speech separation models trained on top of HuBERT

ARE DEEP NET REALLY DIFFERENT FROM OUR BRAIN?

Toward a realistic model of speech processing in the brain with self-supervised learning

Wav2vec2.0 VS. humain brain



Introduction

- **Several deep neural networks have recently been shown to generate activations similar to those of the brain in response to the same input.**
- **These algorithms,however, remain largely implausible.**

(1) extraordinarily large amounts of data (40GB for GPT-2 Radford et al. [2019], the equivalent of multiple lifetimes of reading),

(2) supervised labels which are rare in human experience (e.g. Yamins and DiCarlo)

(3) data in a textual rather than a raw sensory format, and/or

(4) Considerable memory (e.g., language models typically have parallel access to thousands of context words to process text).

These differences highlight the pressing necessity to **identify architectures and learning objectives** which, subject to these four constraints, would be sufficient to account for both behavior and brain responses.

Wav2vec2.0 VS. humain brain

compare a recent self-supervised architecture, Wav2Vec 2.0, to the brain activity of 412 English, French, and Mandarin individuals recorded with **functional Magnetic Resonance Imaging (fMRI)**, while they listened to =1 h of audio books. Our results are four-fold.

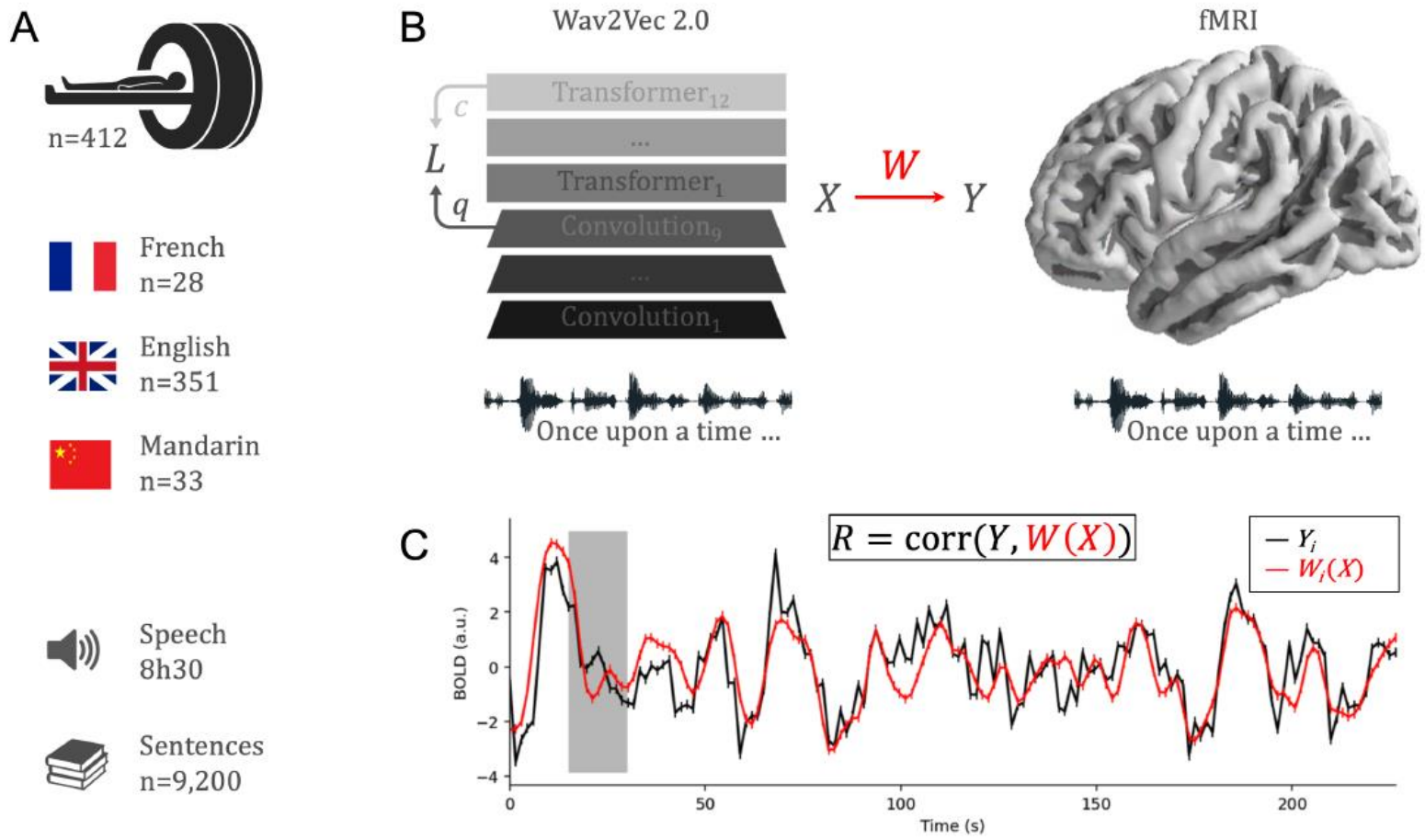
First, we show that this algorithm **learns brain-like representations** with as little as 600 hours of unlabelled speech – a quantity comparable to what infants can be exposed to during language acquisition.

Second, its **functional hierarchy aligns** with the cortical hierarchy of speech processing.

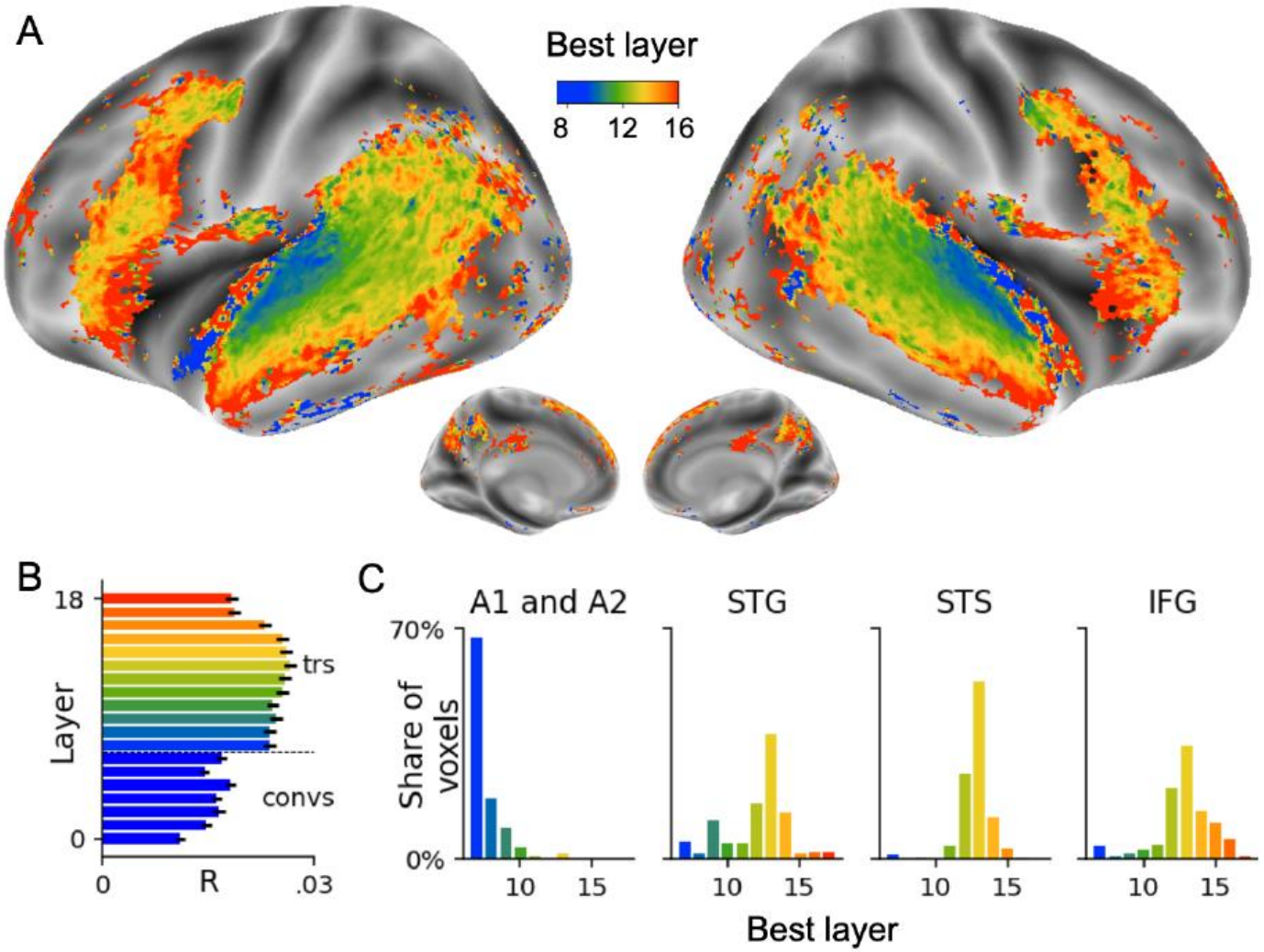
Third, different training regimes reveal a functional specialization akin to the cortex: Wav2Vec 2.0 learns **sound-generic, speech-specific and language-specific representations** similar to those of the prefrontal and temporal cortices.

Fourth, we confirm the similarity of this specialization with the behavior of 386 additional participants.

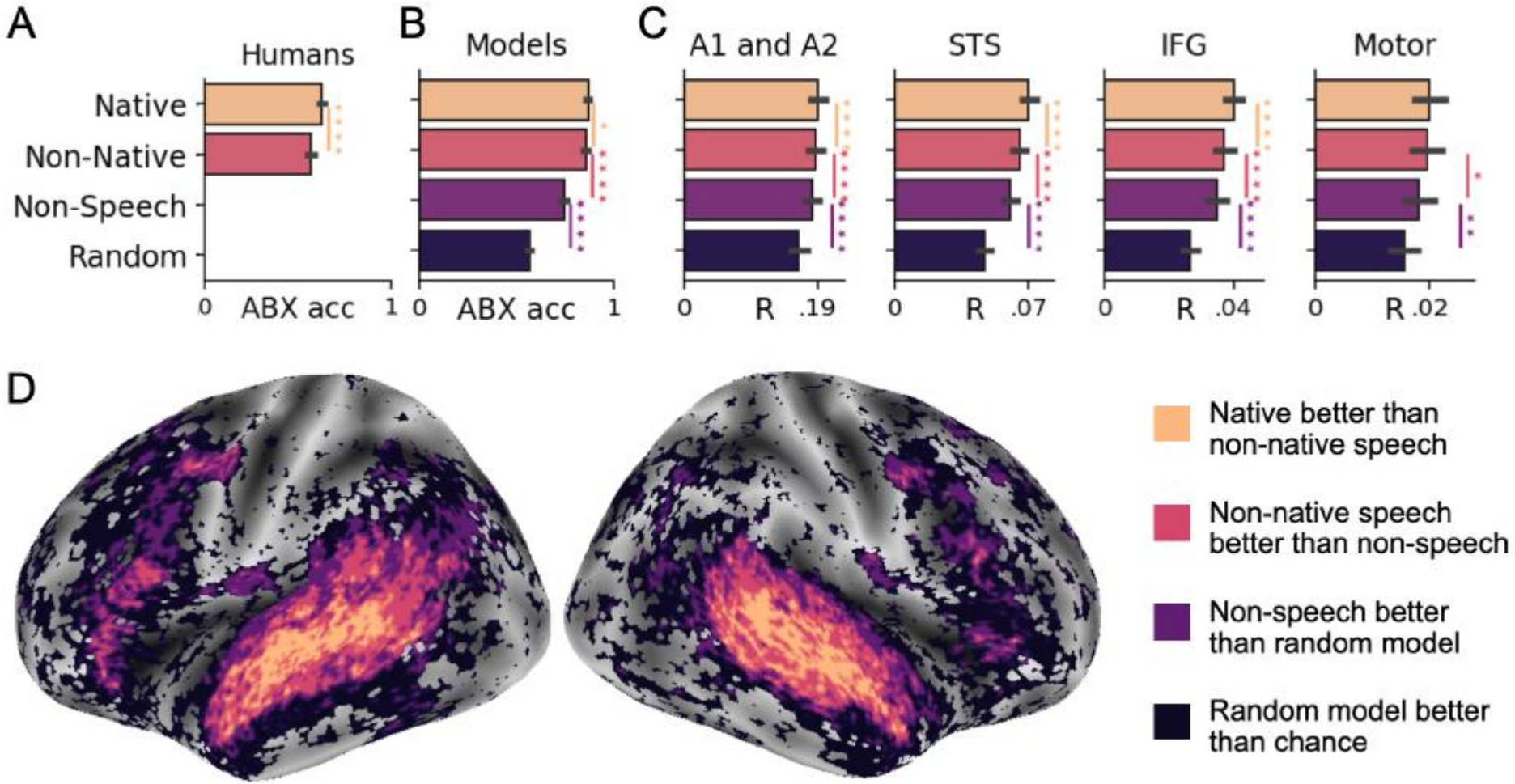
Comparing speech representations in brains and deep neural networks



The functional hierarchy of Wav2Vec 2.0 maps onto the speech hierarchy in the brain



The specialization ofWav2Vec 2.0's representations follows and clarifies the acoustic, speech, and language regions in the brain.



References:

1. https://arxiv.org/abs/2006.11477 (Wav2Vec2)
2. https://arxiv.org/abs/2106.07447 (Hubert)
3. https://arxiv.org/pdf/2206.01685.pdf (Toward a realistic model of speech processing in the brain with self-supervised learning)
4. https://jonathanbgn.com/2020/12/31/self-supervised-learning.html

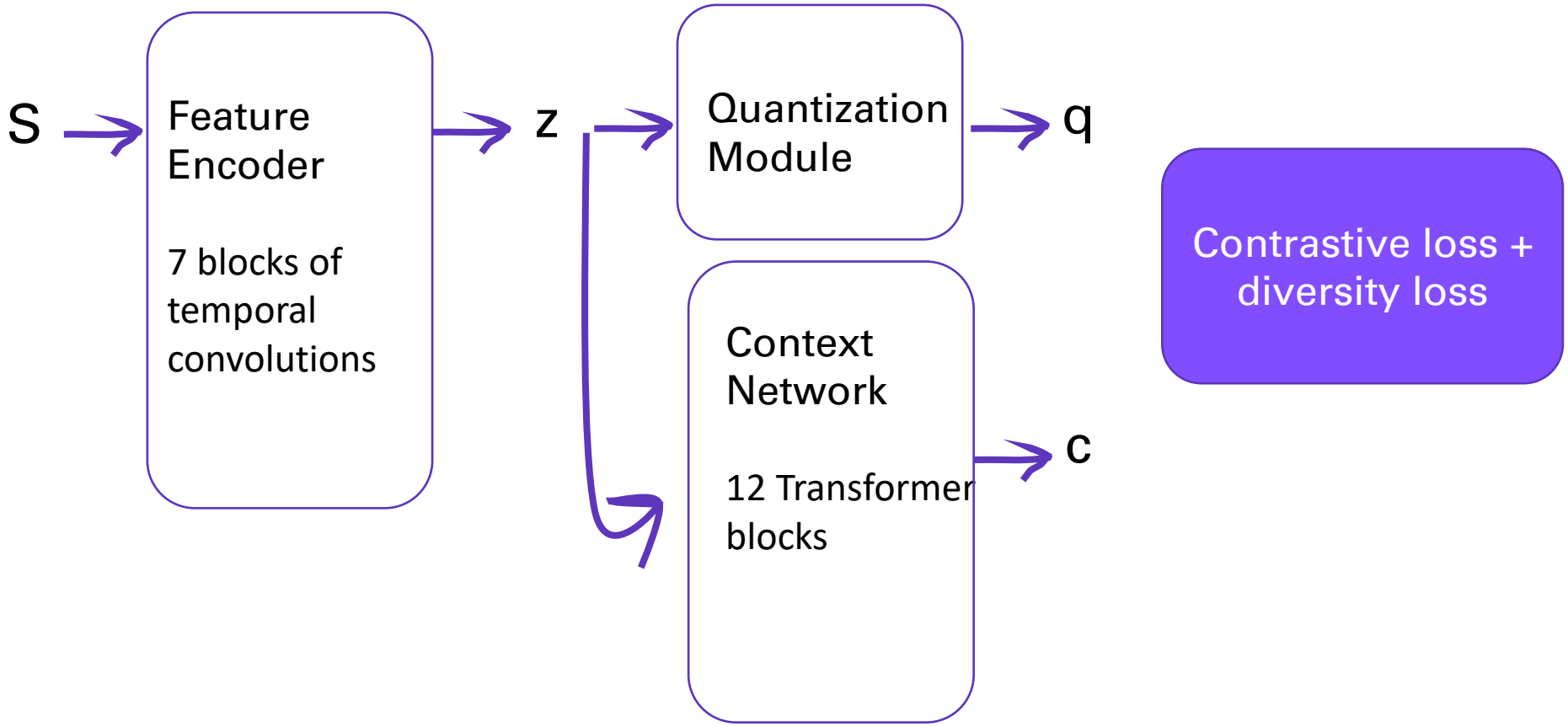
THANK YOU

https://github.com/afshari-maryam/SelfSupervised-models-repo

afshari1431@gmail.com

The END

Wav2vec2.0 Summary



Wav2vec2:

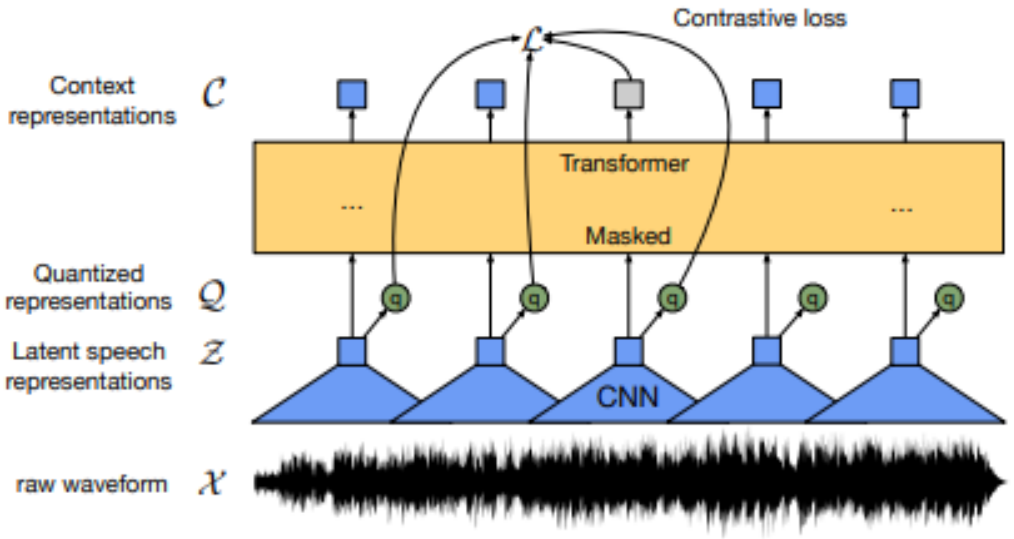


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

Wav2vec2:

The Gumbel softmax enables choosing discrete codebook entries in a fully differentiable way [16, 24, 35]. We use the straight-through estimator [26] and setup G hard Gumbel softmax operations [24]. The feature encoder output $\mathbf{z}$ is mapped to $\mathbf{l} \in \mathbb{R}^{G \times V}$ logits and the probabilities for choosing the v-th codebook entry for group g are

$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau}, \tag{1}$$

where τ is a non-negative temperature, $n = -\log(-\log(u))$ and u are uniform samples from $\mathcal{U}(0, 1)$. During the forward pass, codeword i is chosen by $i = \text{argmax}_j p_{g,j}$ and in the backward pass, the true gradient of the Gumbel softmax outputs is used.

Hubert:

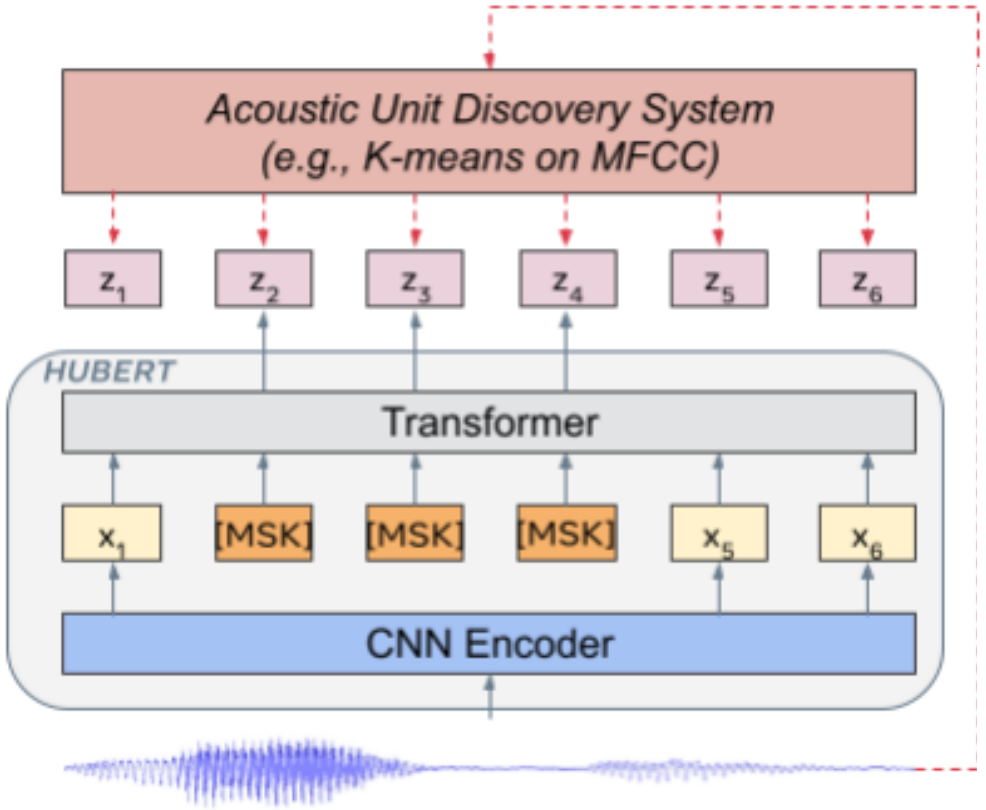


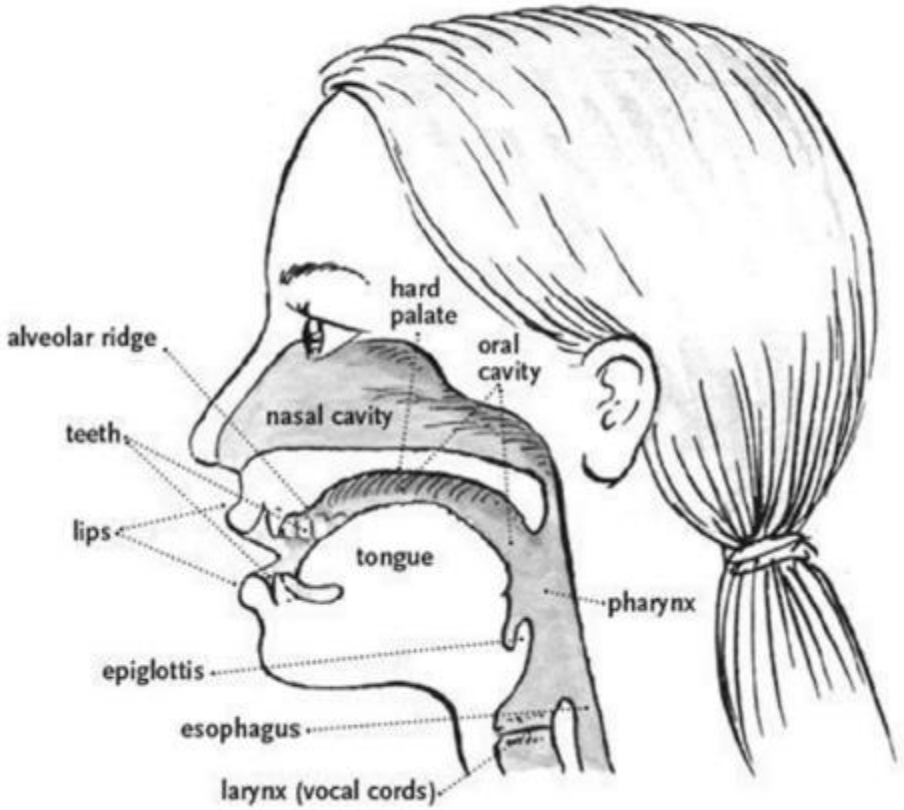
Fig. 1: The HuBERT approach predicts hidden cluster assig ments of the masked frames (y_2, y_3, y_4 in the figure) generat by one or more iterations of k-means clustering.

		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

TABLE I: Model architecture summary for BASE, LARGE, and X-LARGE HuBERT models

Hubert:

A Tour of the Vocal Tract



TABULAR DATA VS. SELF-SUPERVISED

Toward a realistic model of speech processing in the brain with self-supervised learning

Scarf:

SCARF: SELF-SUPERVISED CONTRASTIVE LEARNING USING RANDOM FEATURE CORRUPTION

Dara Bahri, Heinrich Jiang, Yi Tay, Donald Metzler
Google Research
{dbahri,heinrichj,yitay,metzler}@google.com

ABSTRACT

Self-supervised contrastive representation learning has proved incredibly successful in the vision and natural language domains, enabling state-of-the-art performance with orders of magnitude less labeled data. However, such methods are domain-specific and little has been done to leverage this technique on real-world *tabular* datasets. We propose SCARF, a simple, widely-applicable technique for contrastive learning, where views are formed by corrupting a random subset of features. When applied to pre-train deep neural networks on the 69 real-world, tabular classification datasets from the OpenML-CC18 benchmark, SCARF not only improves classification accuracy in the fully-supervised setting but does so also in the presence of label noise and in the semi-supervised setting where only a fraction of the available training data is labeled. We show that SCARF complements existing strategies and outperforms alternatives like autoencoders. We conduct comprehensive ablations, detailing the importance of a range of factors.

Scarf:

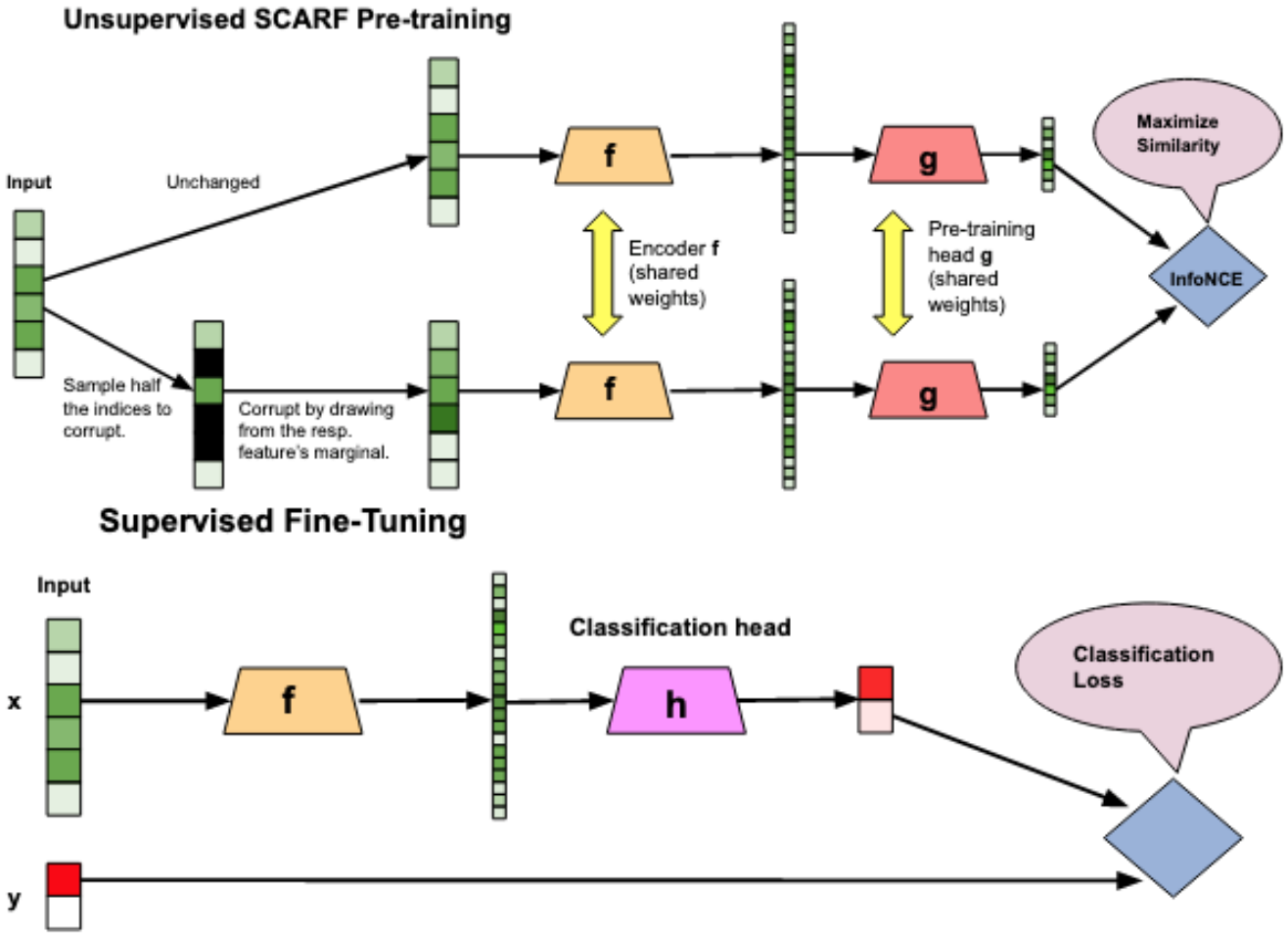


Figure 1: Diagram showing unsupervised SCARF pre-training (**Top**) and subsequent supervised fine-tuning (**Bottom**). During pre-training, networks f and g are learned to produce good representations of the input data. After pre-training, g is discarded and a classification head h is applied on top of the learned f and both f and h are subsequently fine-tuned for classification.