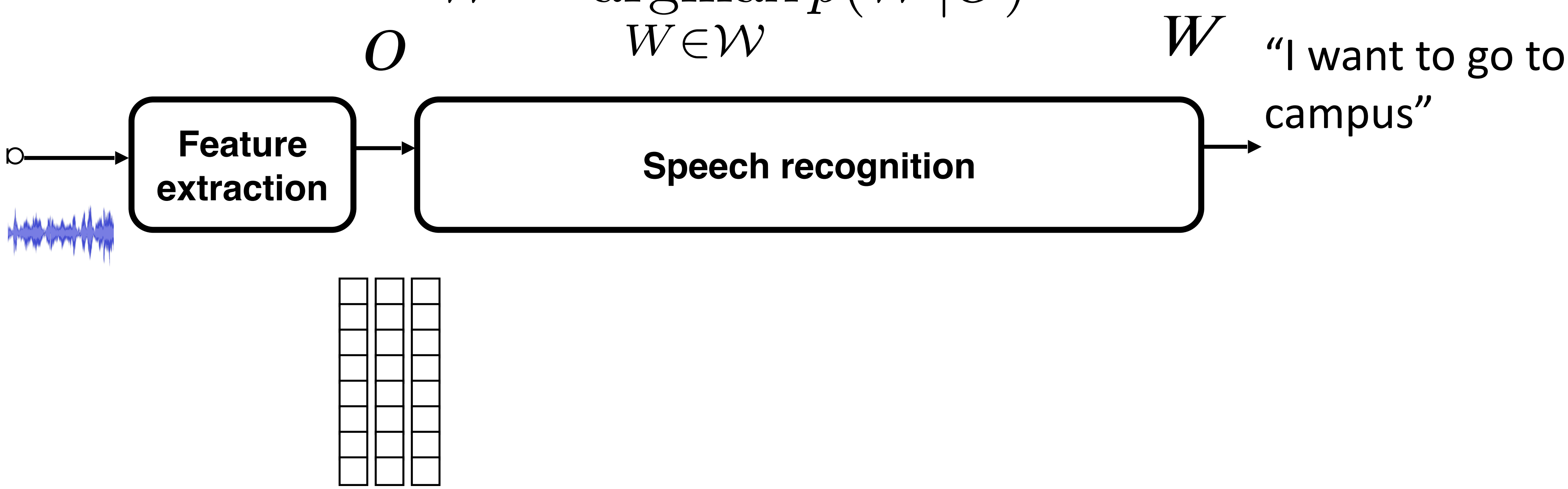
Self-Supervised Learning for Speech

EECS 183/283a: Natural Language Processing

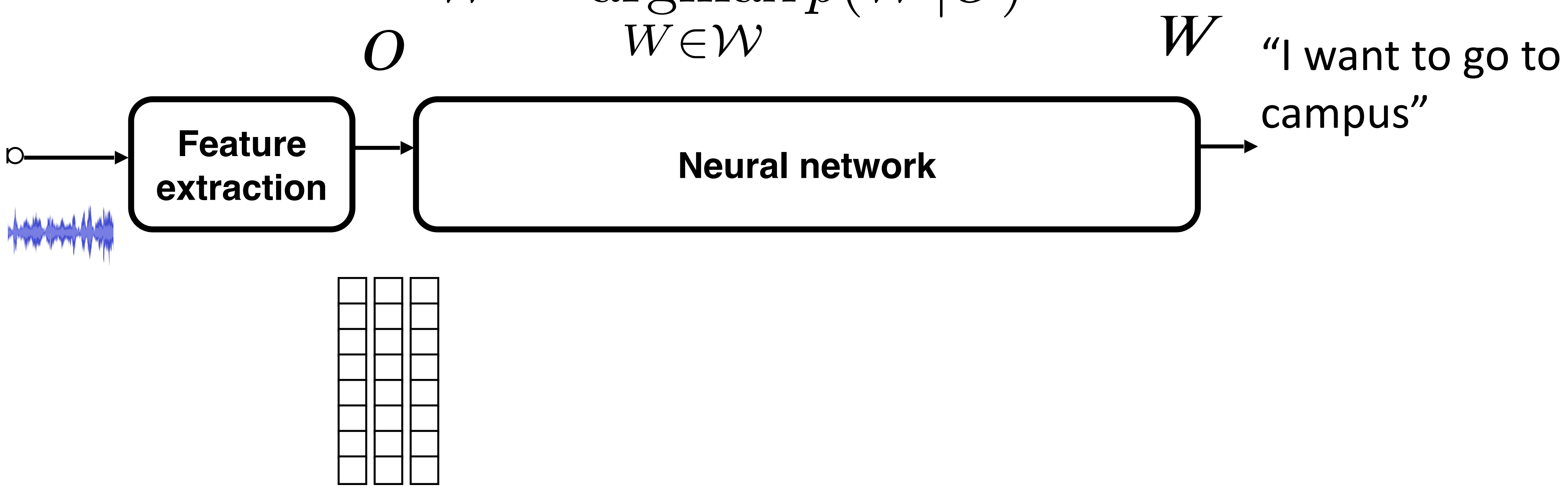
Recap: Speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$



End-to-end speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$



How to obtain the posterior $p(W|O)$

- We just replace it with a neural network-based function $f^{\mathrm{nn}}(\cdot)$

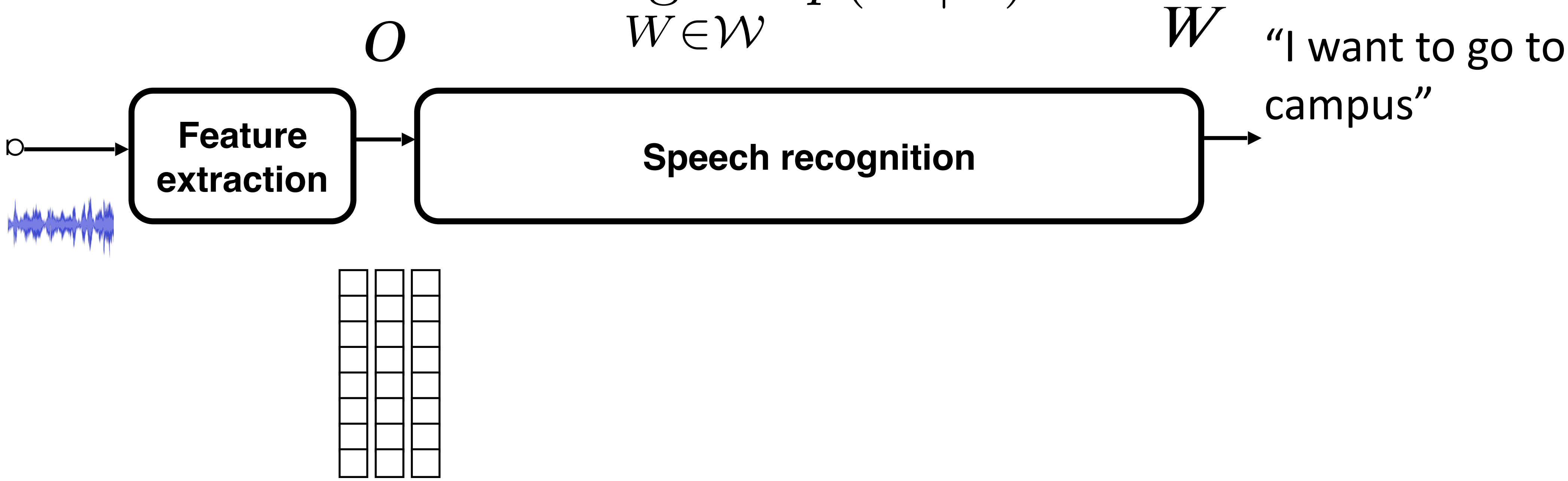
$$\underset{W}{\operatorname{argmax}}\, p(W|O) = \underset{W}{\operatorname{argmax}}\, f^{\mathrm{nn}}(W|O)$$

  - Easy and simple, no math (in this level), however
- W is a **sequence**! $W = (w_n \in \mathcal{V} | n = 1, \ldots, N)$
  - Very difficult to deal with it
  - Say $N = 10$, $|\mathscr{V}| = 100$, we have to deal with 100^{10} possible sequences
  - Also, the length N is variable
  - We have to use a **special neural network** (e.g., attention, CTC, and RNN-transducer)

- Classical speech recognition
  - Pipeline

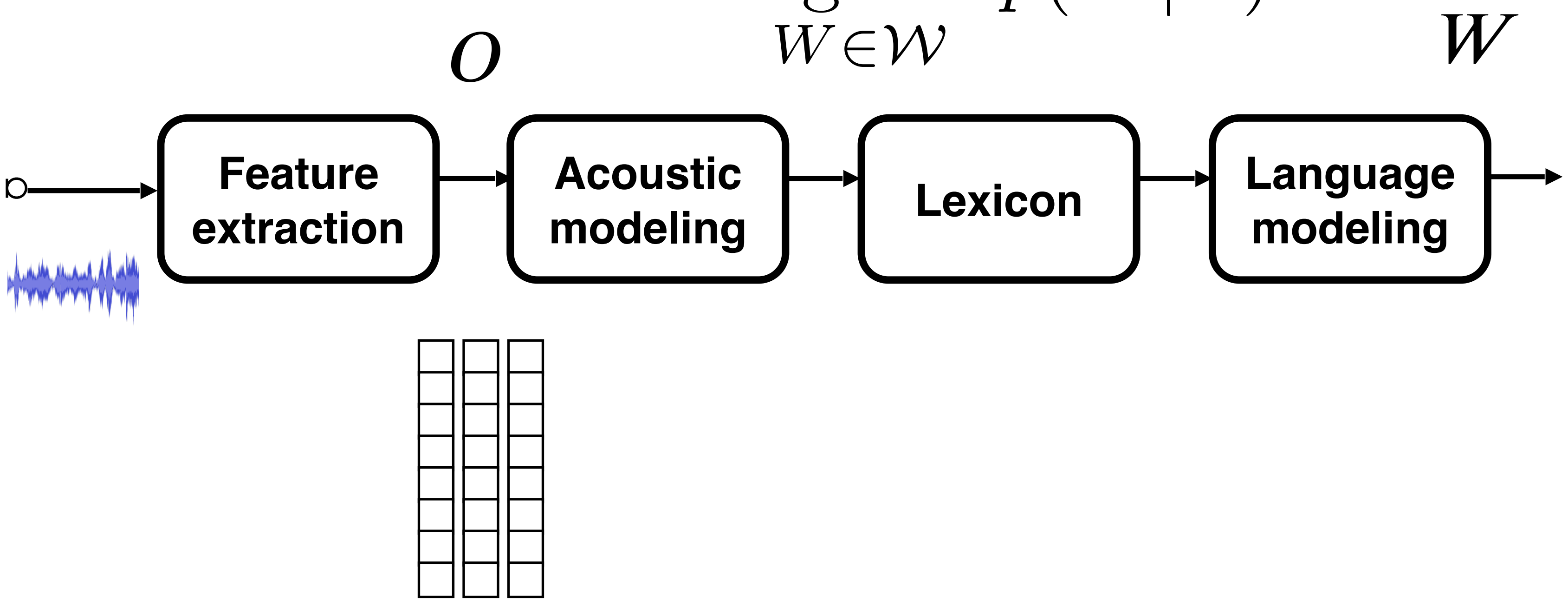
Speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$



Classical (non-end-to-end) speech recognition

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, p(W|O)$$



"I want to **go to** campus"

Speech → Text

Speech O:

Text W: I want to go to the campus

Speech → Phoneme → Text

Speech O:

ASR

Phoneme L: AY W AA N T T UW G OW T UW K AE M P AH S

ASR

Text W: I want to go to campus

How to obtain the posterior $p(W|O)$

- Factorize the model with **phoneme**
  - Let $L = (l_i \in \{/\mathrm{AA}/, /\mathrm{AE}/, \cdots\} | i = 1, \cdots, J)$ be a phoneme sequence

$$\arg\max_W p(W|O) = \arg\max_W \sum_L p(W, L|O)$$ **Sum rule**

$$= \arg\max_W \sum_L \frac{p(O|W,L)p(L|W)p(W)}{p(O)}$$ **Bayes+ Product rule**

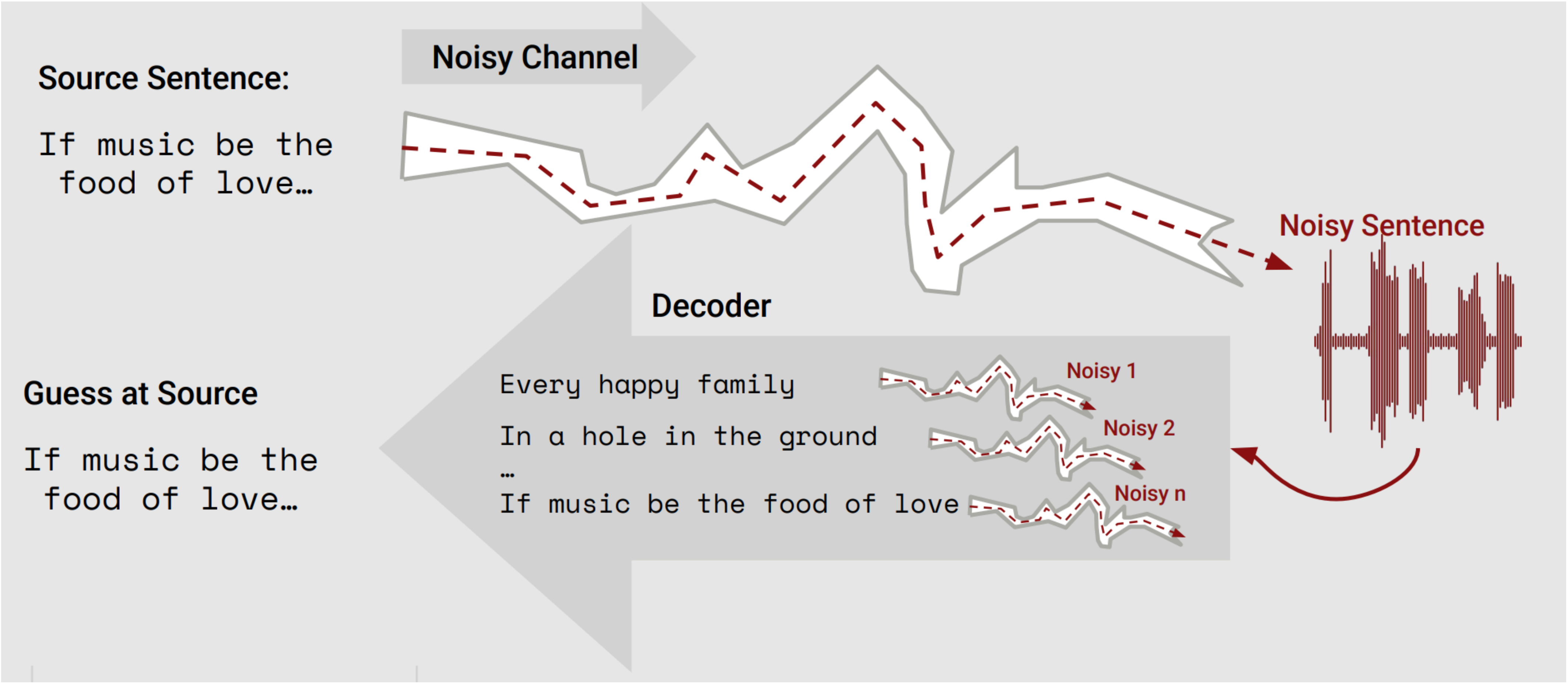
Ignore $p(O)$ as it does not depend on W

$$= \arg\max_W \sum_L p(O|W,L)p(L|W)p(W)$$

$$= \arg\max_W \sum_L p(O|L)p(L|W)p(W)$$ **Conditional independence assumption**

Note: the right hand side is not the probability as it lacks a sum to one constraint

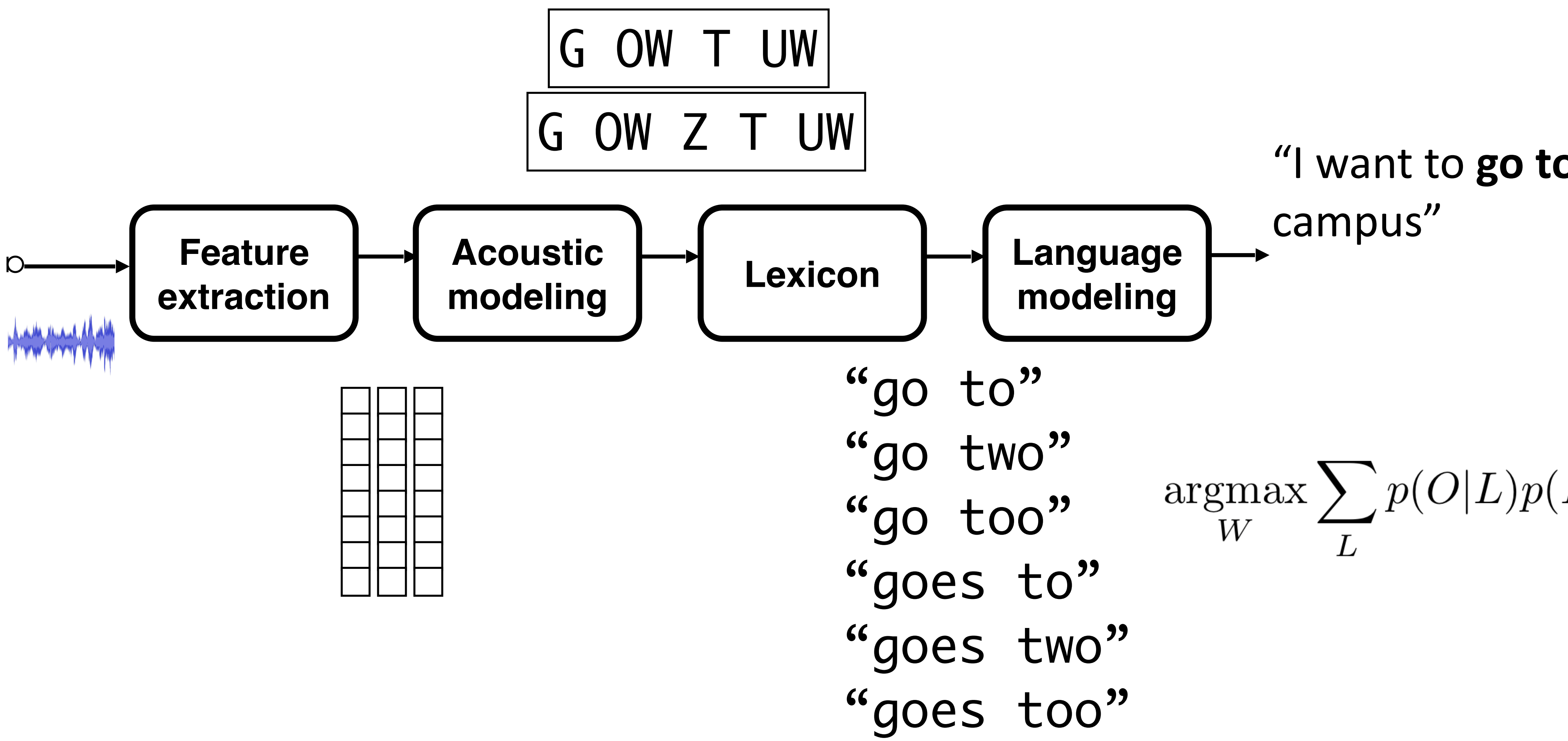
Noisy channel model



$$\underset{W}{\operatorname{argmax}}\, p(W \mid O) = \underset{W}{\operatorname{argmax}}\, p(O \mid W)p(W) \approx \underset{W}{\operatorname{argmax}} \sum_{L} p(O \mid L)p(L \mid W)p(W)$$

- **Speech recognition**
  - $p(O \mid L)$: Acoustic model (Hidden Markov model)
  - $p(L \mid W)$: Lexicon
  - $p(W)$: Language model (n-gram)

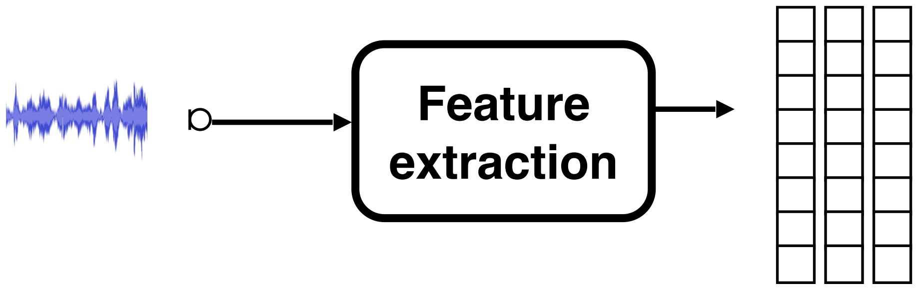
Speech recognition pipeline



$$\operatorname*{argmax}_{W} \sum_{L} p(O|L)p(L|W)p(W)$$

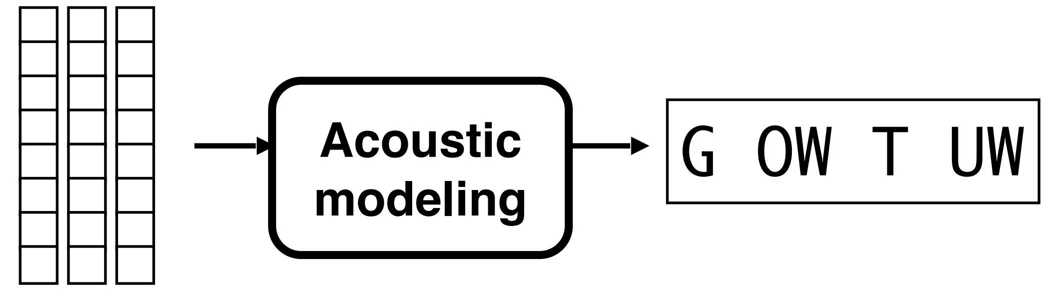
$$p(O|L) \qquad p(L|W) \qquad p(W)$$

Waveform to speech feature



- Performed by so-called **feature extraction** module
  - Mel-frequency cepstral coefficient (MFCC), Perceptual Linear Prediction (PLP) used for **Gaussian mixture model (GMM)**
  - Log Mel filterbank used for **deep neural network (DNN)**
- Time scale
  - 0.0625 milliseconds (16kHz) to 10 milliseconds
- Type of values
  - Scalar (or discrete) to 12—40 dimensional vector

Speech feature to phoneme

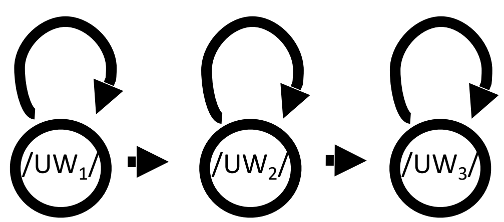


- Performed by so-called **acoustic modeling** module
  - Hidden Markov model (HMM) with **GMM** as an emission probability function
  - Hidden Markov model (HMM) with **DNN** as an emission probability function
- Time scale
  - 10 milliseconds to ~100 milliseconds (depending on a phoneme)
- Type of values
  - 12-dimensional continuous vector to 50 categorical value (~6bit)
- The most critical component to get the ASR performance
- It can be a probability of possible phoneme sequences, e.g.,

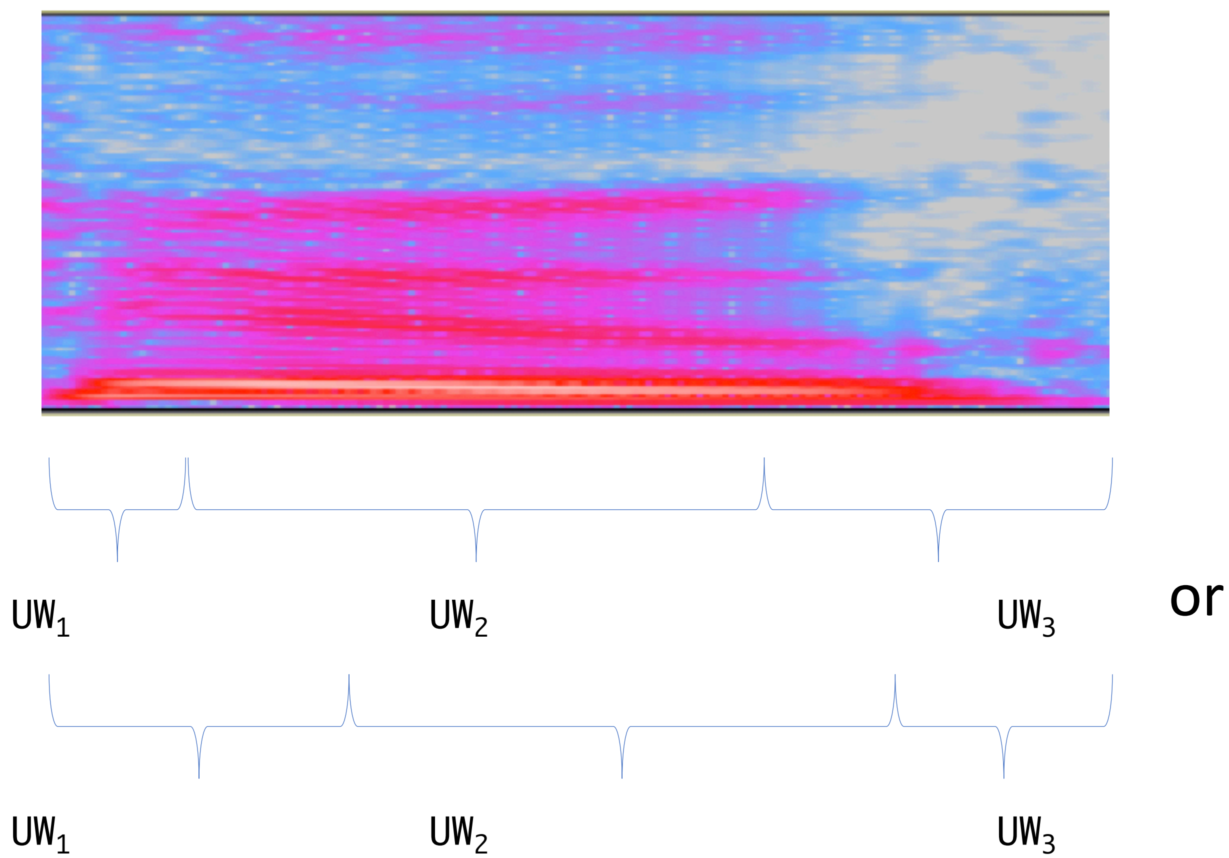
  G OW T UW or G OW Z T UW with some scores

Acoustic model $p(O \mid L)$

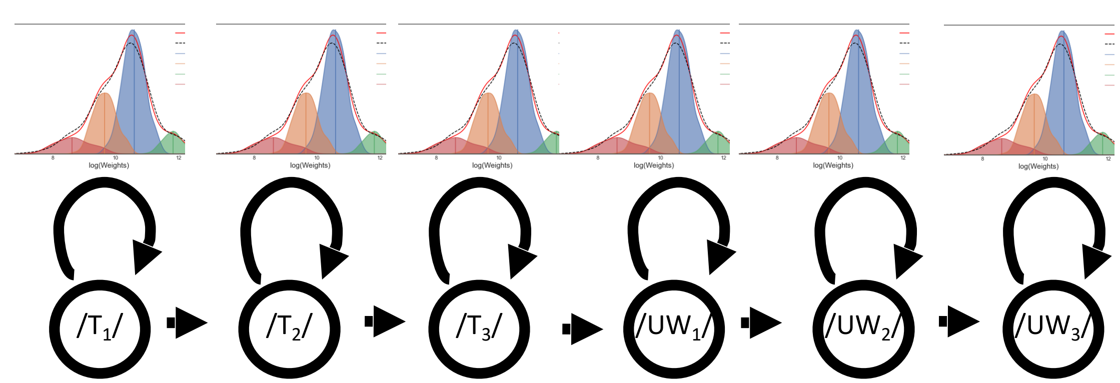
- O and L are **different lengths**
- **Align** speech features and phoneme sequences by using HMM



- Provide $p(O \mid L)$ based on this alignment and model

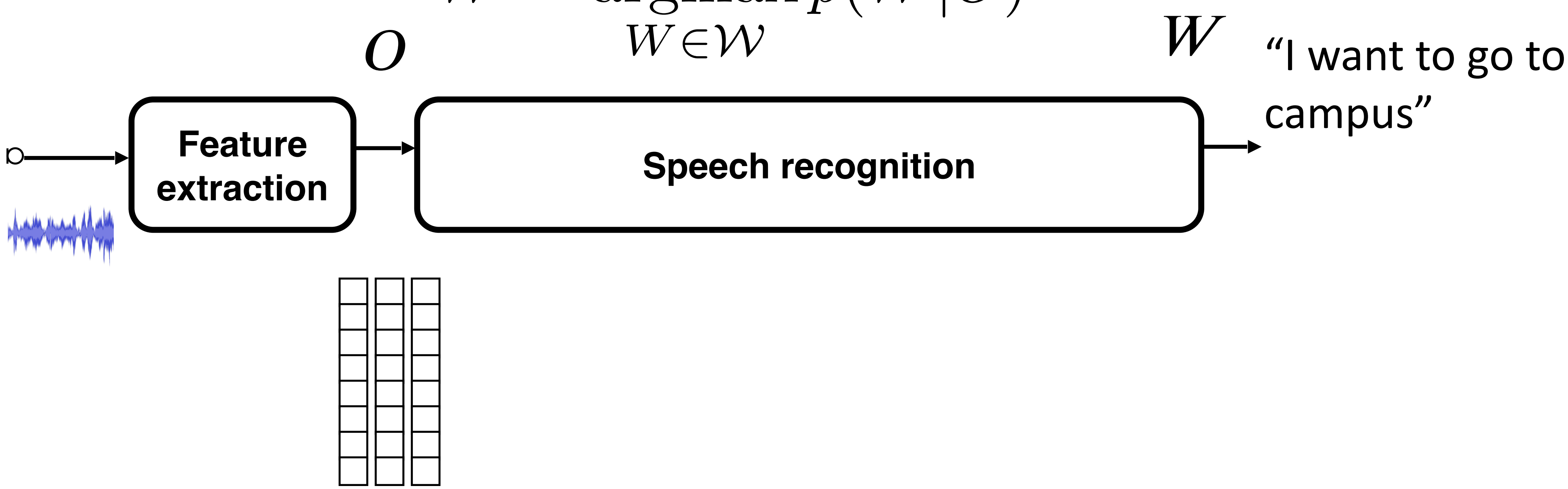


- Output distributions can be modeled using Gaussian Mixture Models (Max. Likelihood parameters of the distributions)



Speech recognition

$$\hat{W} = \operatorname*{argmax}_{W \in \mathcal{W}} p(W|O)$$



End-to-End ASR

- Direct mapping function from speech feature sequence O to text W
- Usually, it does **not** deal with the phoneme-based intermediate representation
- Mainly three architectures
  - Attention-based
    - Decoder-only is also included here (not fully end-to-end)
  - Connectionist temporal classification (CTC)
  - Recurrent neural network transducer (RNN-T)

HMM-based (Classical)

CTC

RNN-transducer

Attention-based

Attention-based ASR

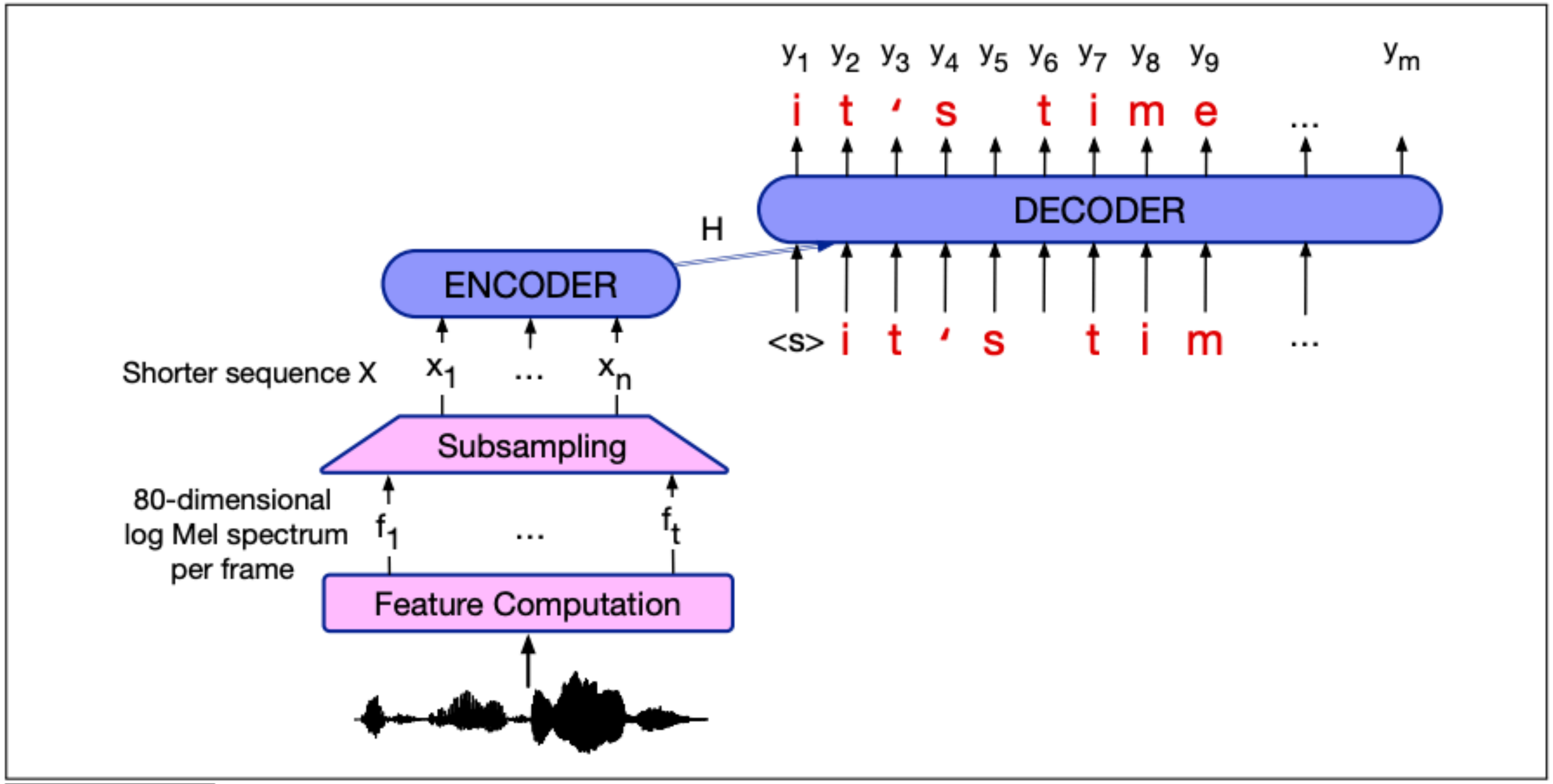


Figure 15.5 Schematic architecture for an encoder-decoder speech recognizer.

Listen Attend & Spell (2016)

Attention-based ASR

- Our staring point $p(W \mid O)$
  - Input: $O = (\mathbf{o}_t \mid t = 1, \ldots, T)$
  - It is difficult to deal with $W = \left(w_i \in \mathscr{V} \,\middle|\, i = 1, \ldots, N\right)$
  - $\boldsymbol{T \neq N}$
- Instead, we **factorize** $p(W \mid O)$ as follows based on a **probabilistic chain rule**

$$p(W|O) = \prod_{i=1}^{N} p(w_i | w_{1:i-1}, O)$$

Attention-based ASR

- Our staring point $p(W \mid O)$
  - Input: $O = (\mathbf{o}_t \mid t = 1, \ldots, T)$
  - It is difficult to deal with $W = \left(w_i \in \mathcal{V} \middle| i = 1, \ldots, N\right)$
  - $T \neq N$
- Instead, we **factorize** $p(W \mid O)$ as follows based on a **probabilistic chain rule**

$$p(W|O) = \prod_{i=1}^{N} p(w_i|w_{1:i-1}, O)$$

- This neural network is handled by an **attention-based method** to align the input and output **(soft alignments)**
- We usually do not use the phoneme

Whisper (OpenAI)

$$p(W|O) = \prod_{i=1}^{N} p(w_i|w_{1:i-1}, O)$$

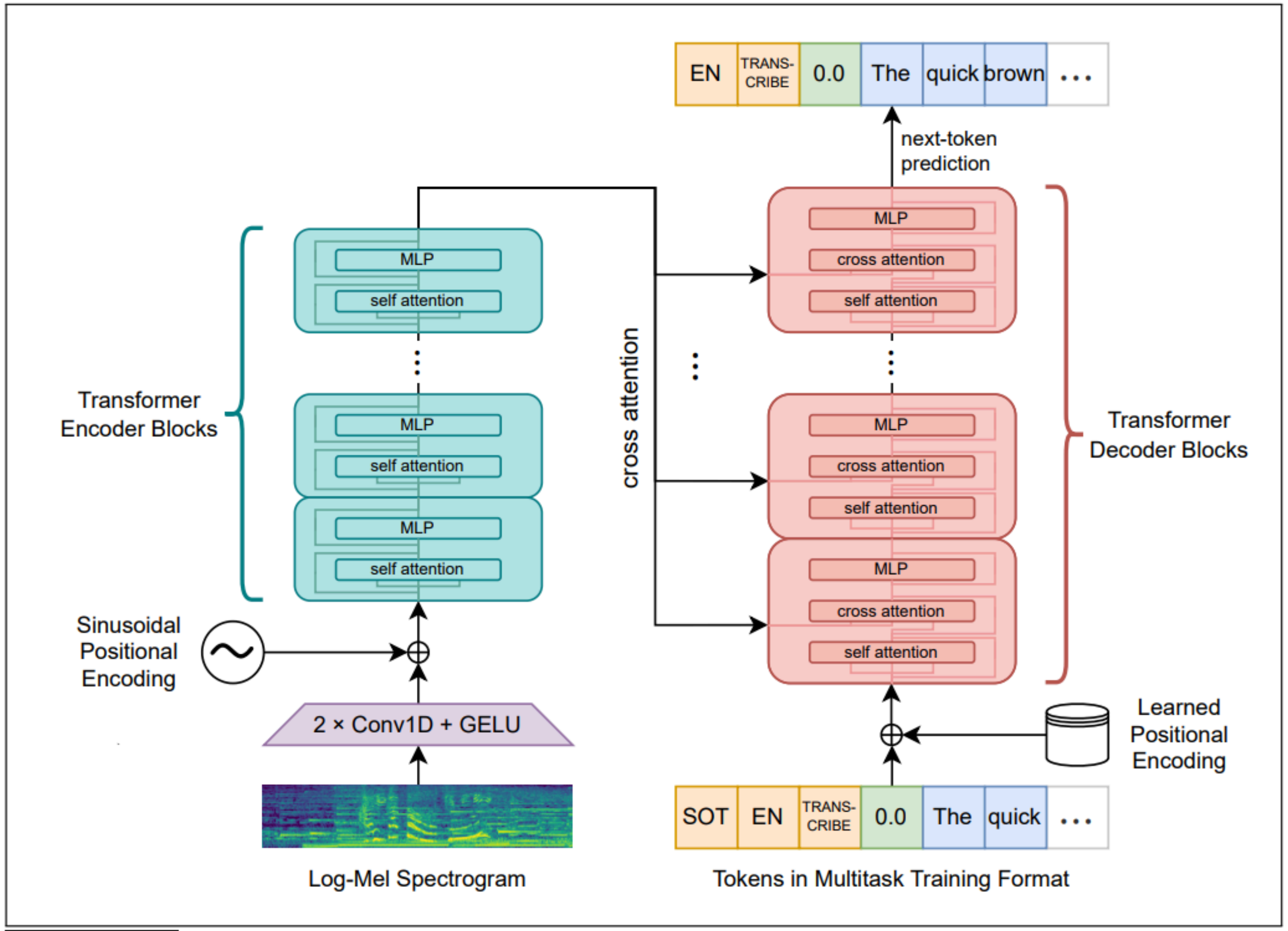
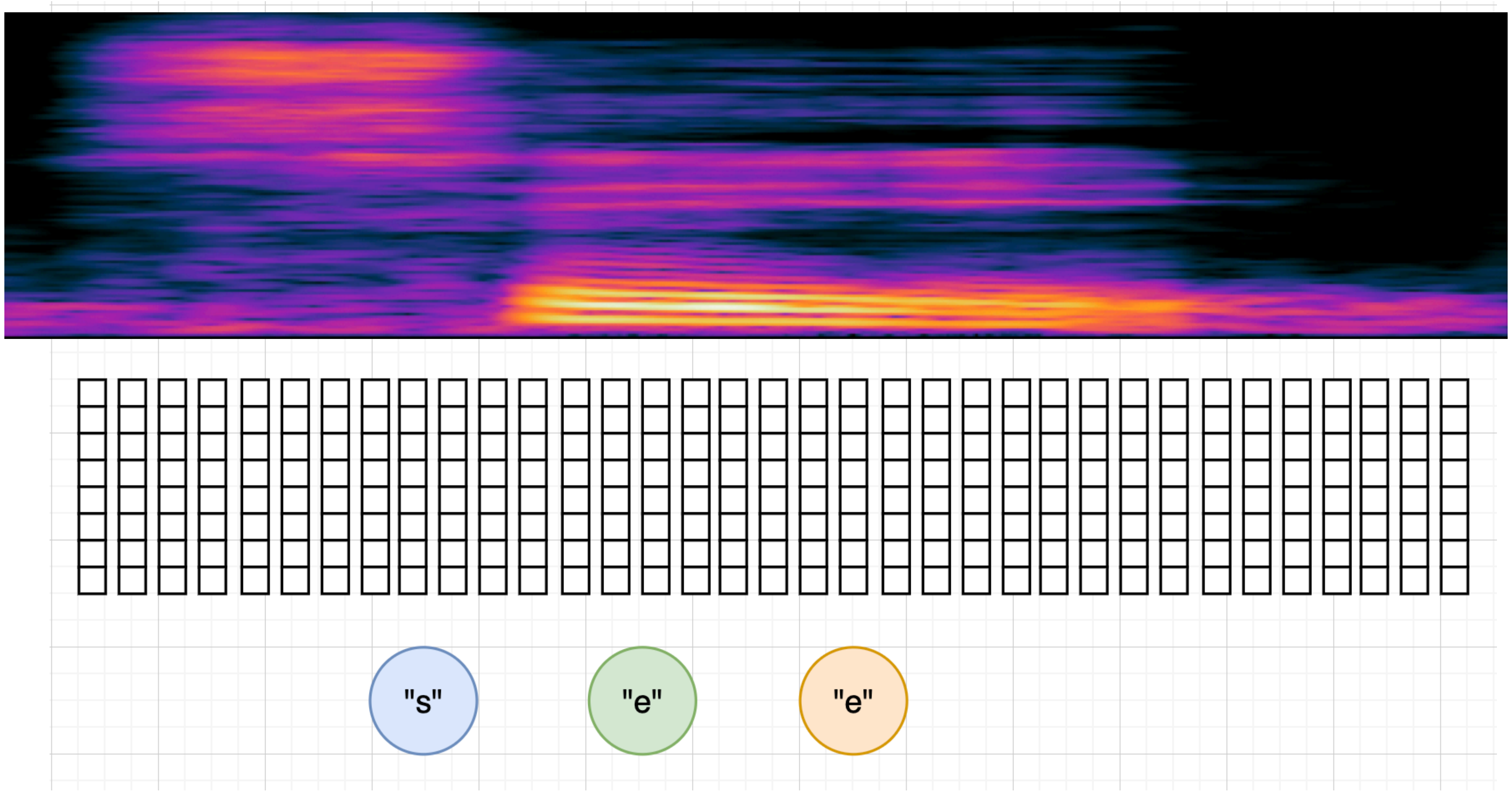


Figure 15.6 A sketch of the Whisper architecture from Radford et al. (2023). Because Whisper is a multitask system that also does translation and diarization (we'll discuss these non-ASR tasks in the following chapter), Whisper's transcription format has a Start of Transcript (SOT) token, a language tag, and then an instruction token for whether to transcribe or translate.

Alignments



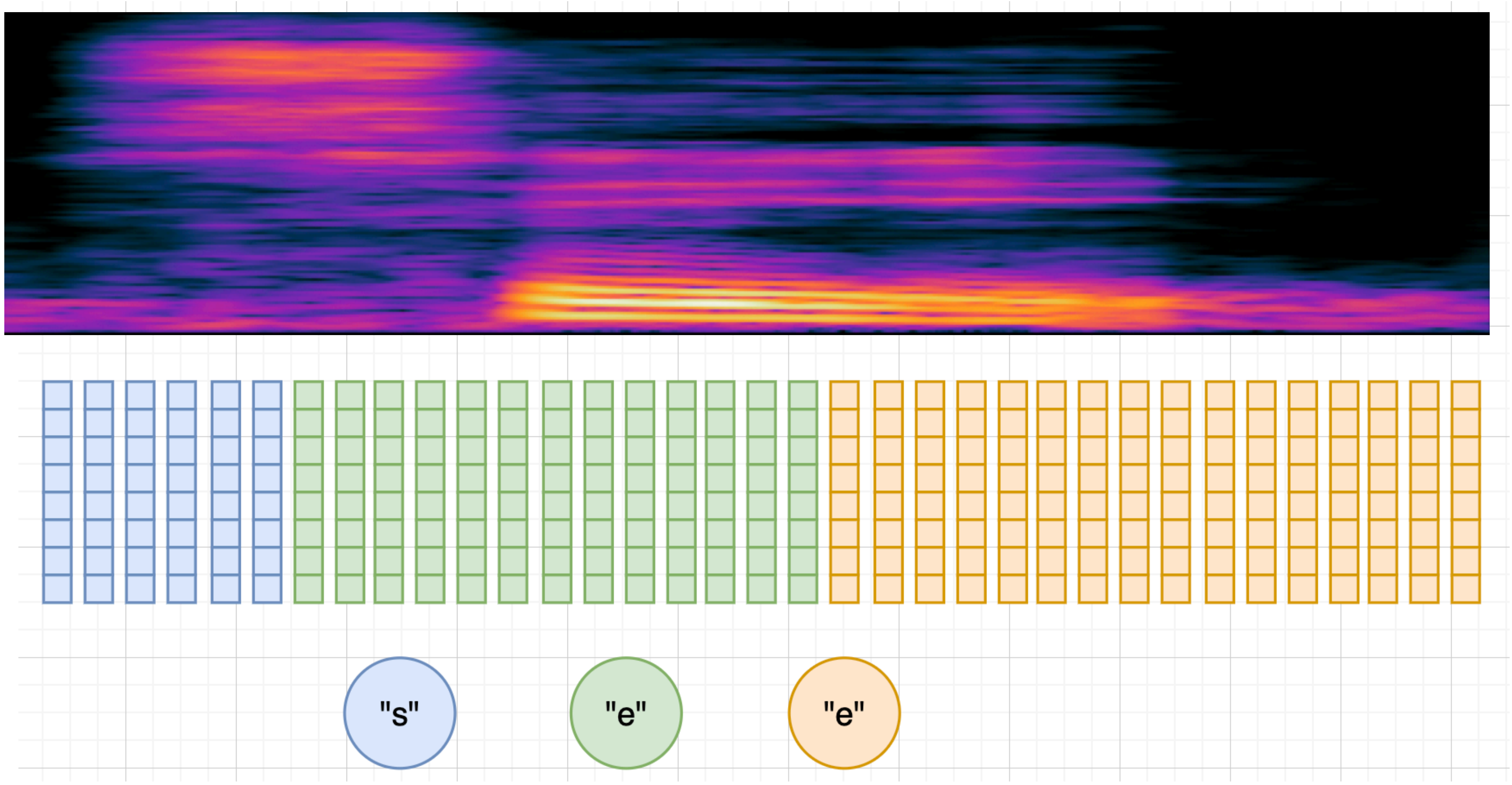
Speech features

O ($|O| = 35$)

Token

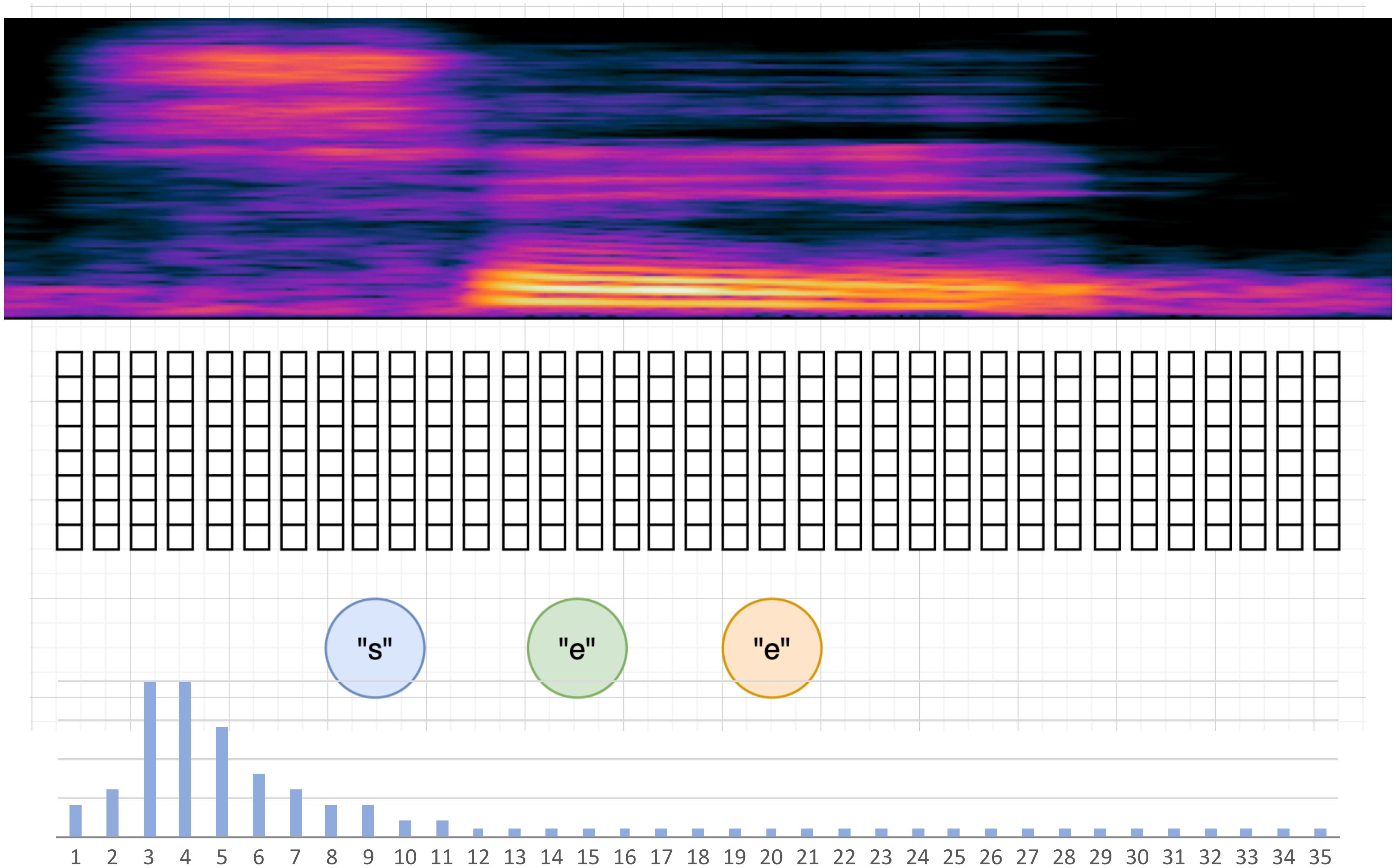
$W(|W| = 3)$

Hard Alignments

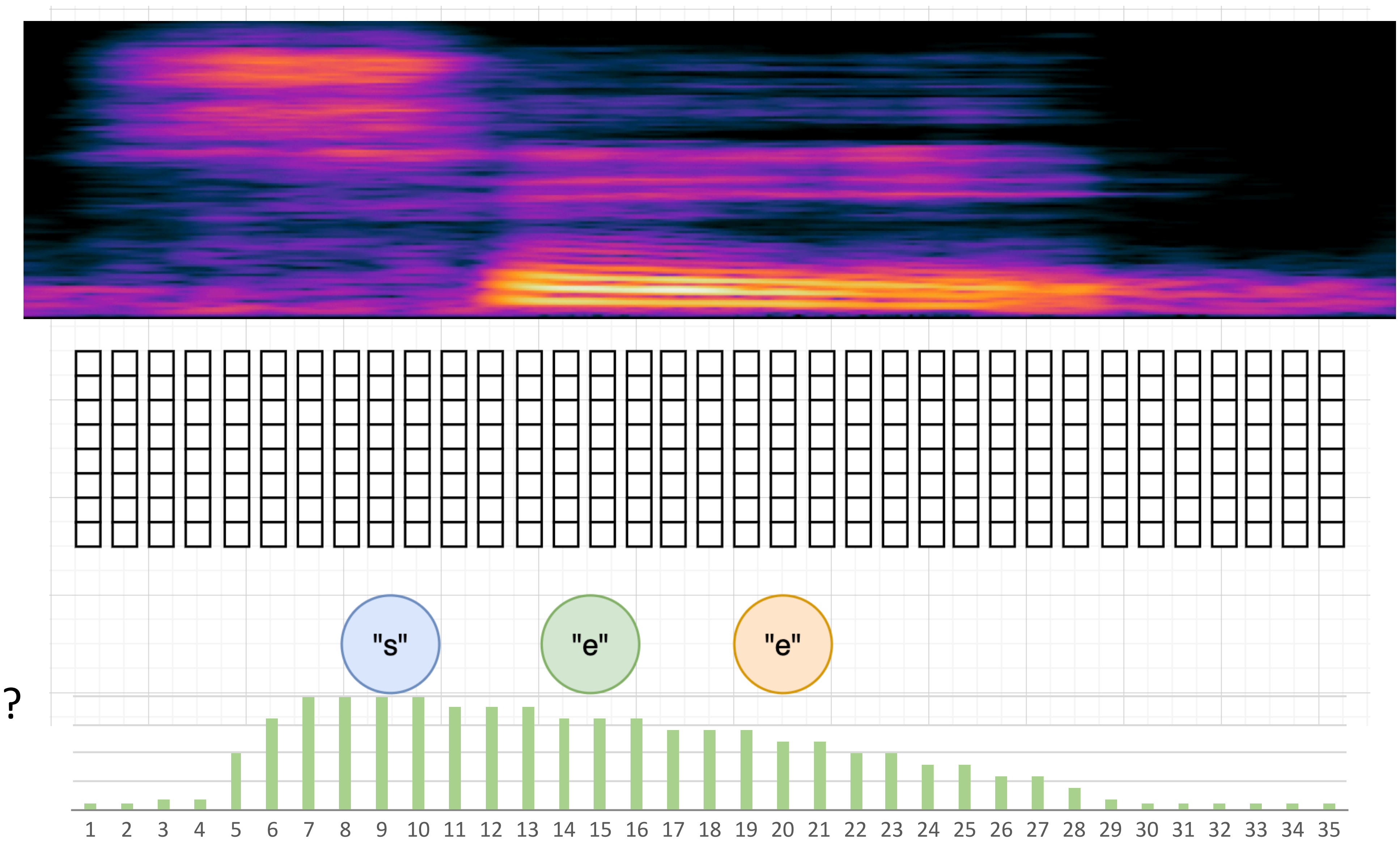


Soft alignments

How many %
“s” is aligned?

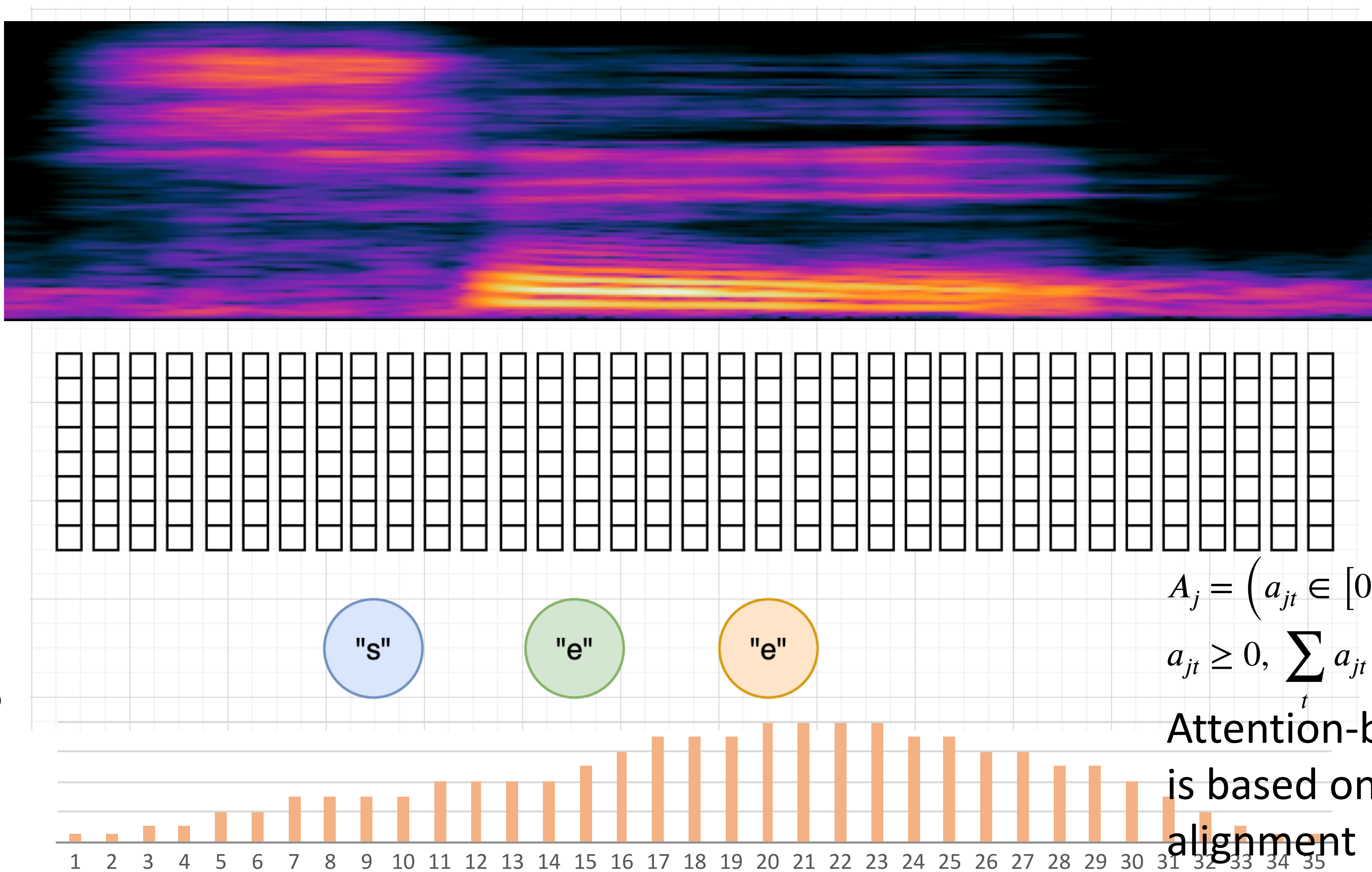


Soft alignments



How many %
“e” is aligned?

Soft alignments



How many %
“e” is aligned?

$$A_j = \left(a_{jt} \in [0,1] \,\middle|\, t = 1,\ldots,T\right)$$

$$a_{jt} \geq 0, \ \sum_t a_{jt} = 1$$

Attention-based approach is based on the **soft** alignment

Connectionist Temporal Classification

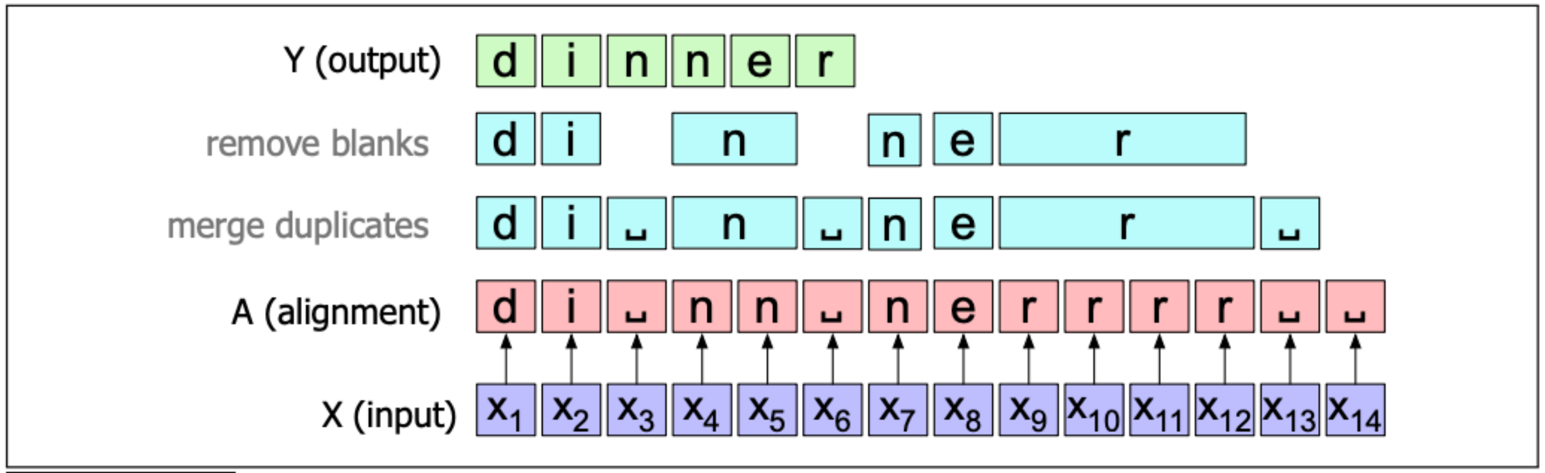


Figure 15.13 The CTC collapsing function B, showing the space blank character ␣; repeated (consecutive) characters in an alignment A are removed to form the output Y.

The CTC collapsing function is many-to-one; lots of different alignments map to the same output string. For example, the alignment shown in Fig. 15.13 is not the only alignment that results in the string *dinner*. Fig. 15.14 shows some other alignments that would produce the same output.

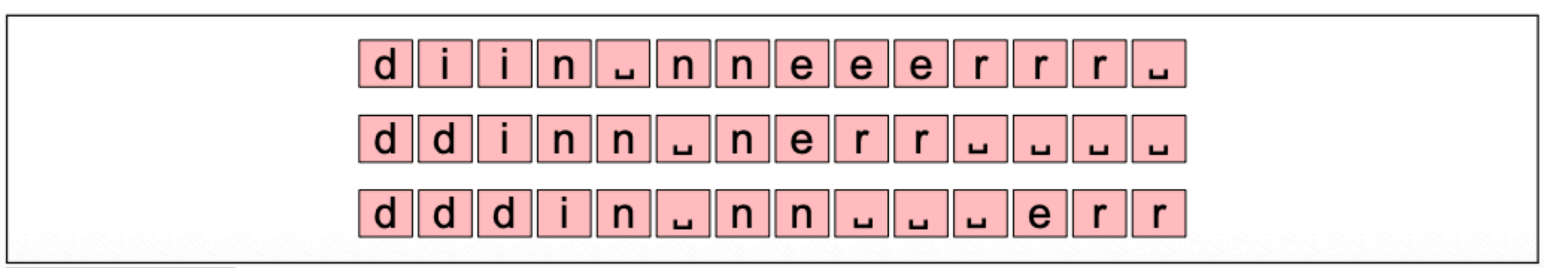


Figure 15.14 Three other legitimate alignments producing the transcript *dinner*.

RNN-Transducer

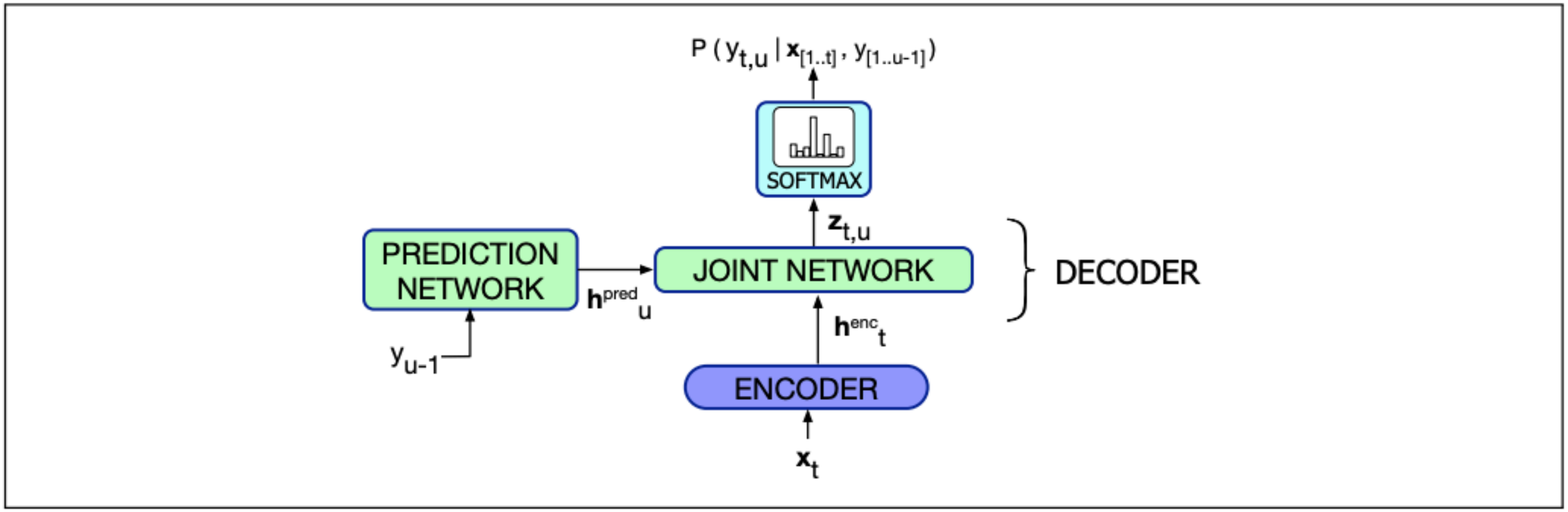
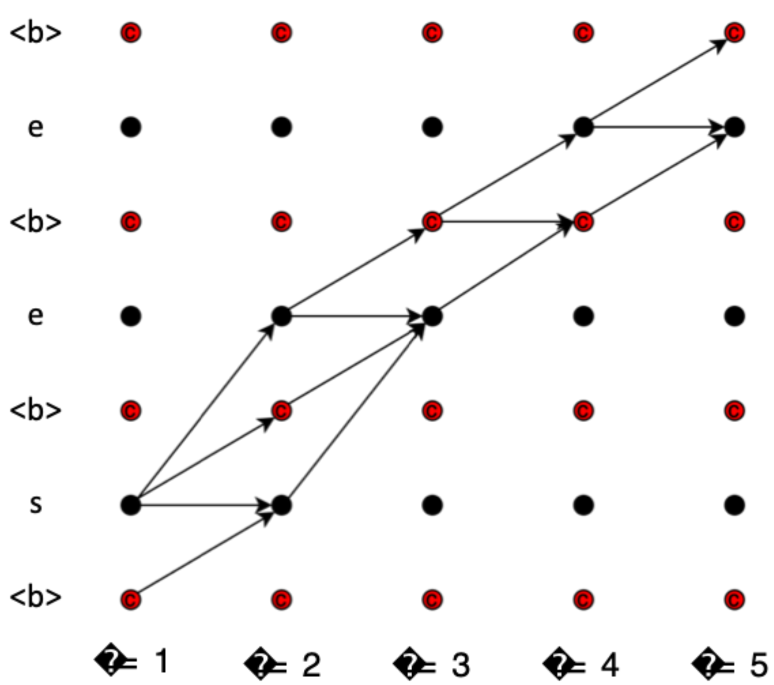


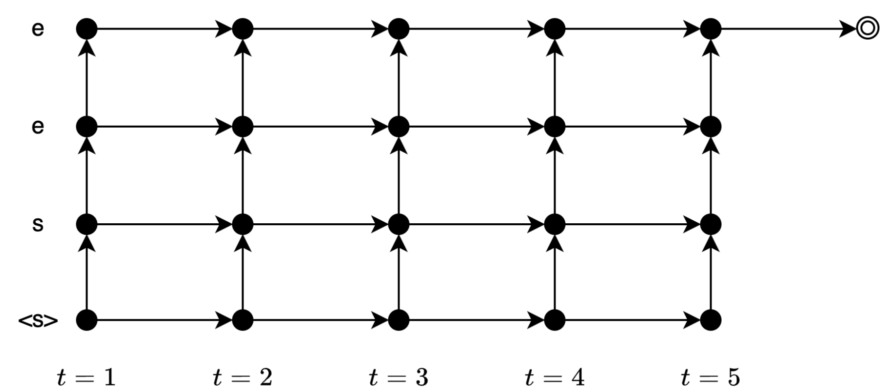
Figure 15.17 The RNN-T model computing the output token distribution at time *t* by integrating the output of a CTC acoustic encoder and a separate 'predictor' language model.

CTC vs. RNN-T

- CTC



- RNN-transducer

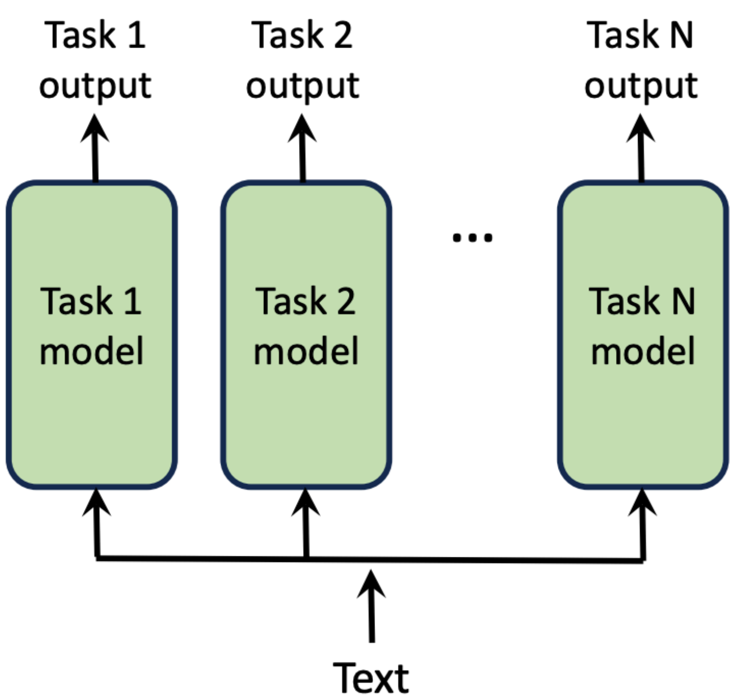


Today's Agenda

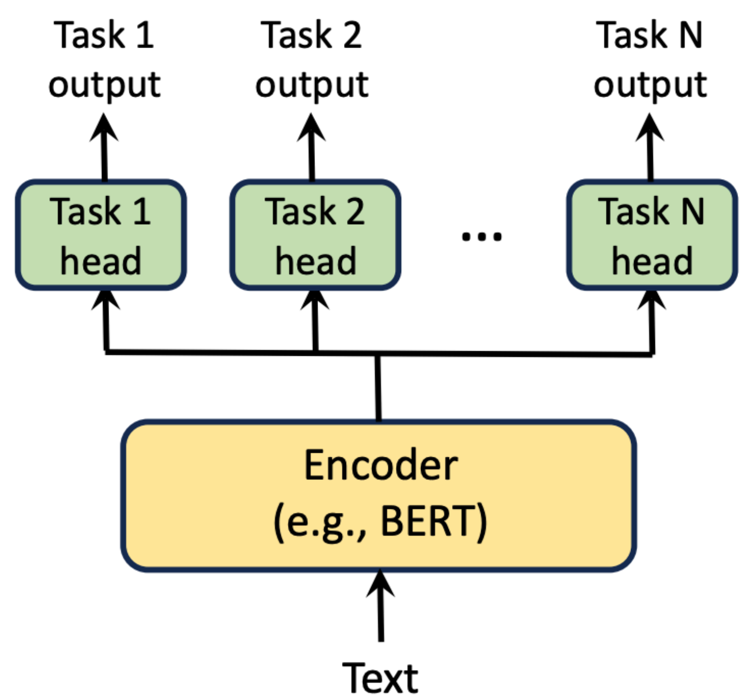
Self-Supervised Learning
for Speech

Evolution of (Text) Foundation Models

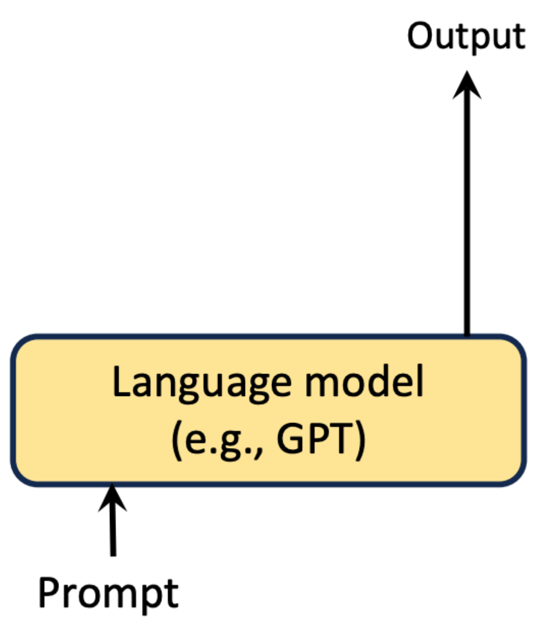
The task-specific model era (- 2018)



The encoder era (2018 - 2022)



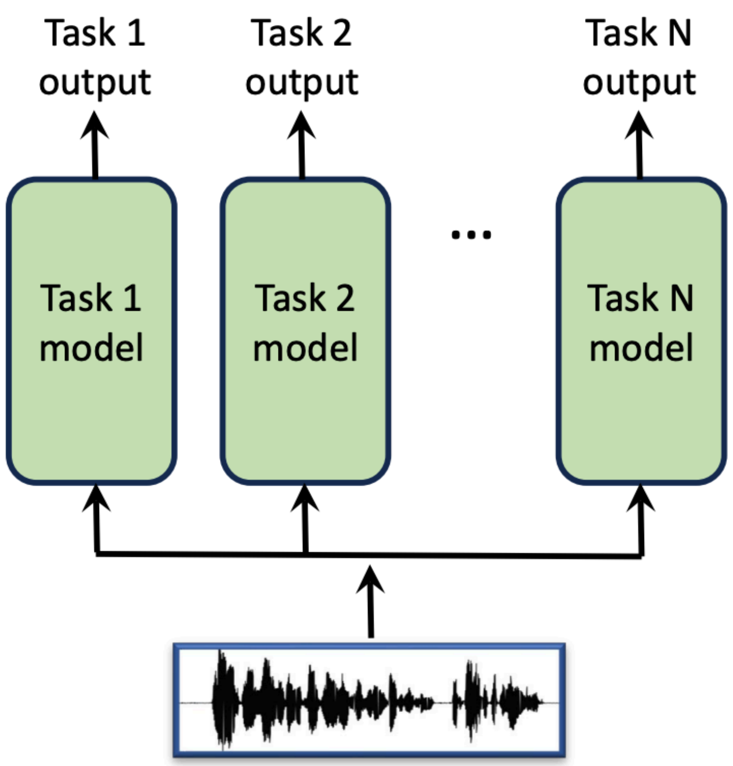
The large language model era (2022 -)



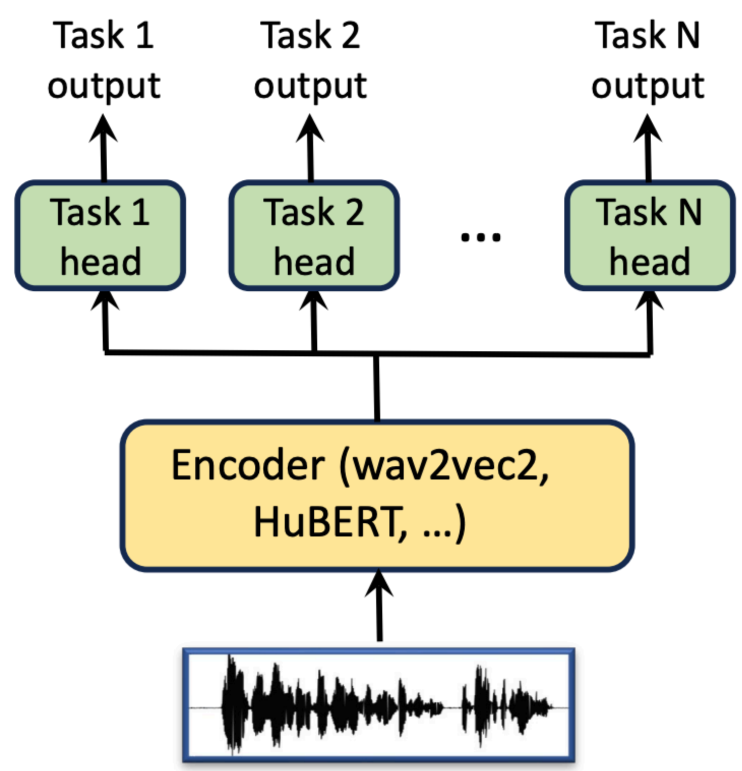
More task-universality, less human effort

Evolution of (Speech) Foundation Models

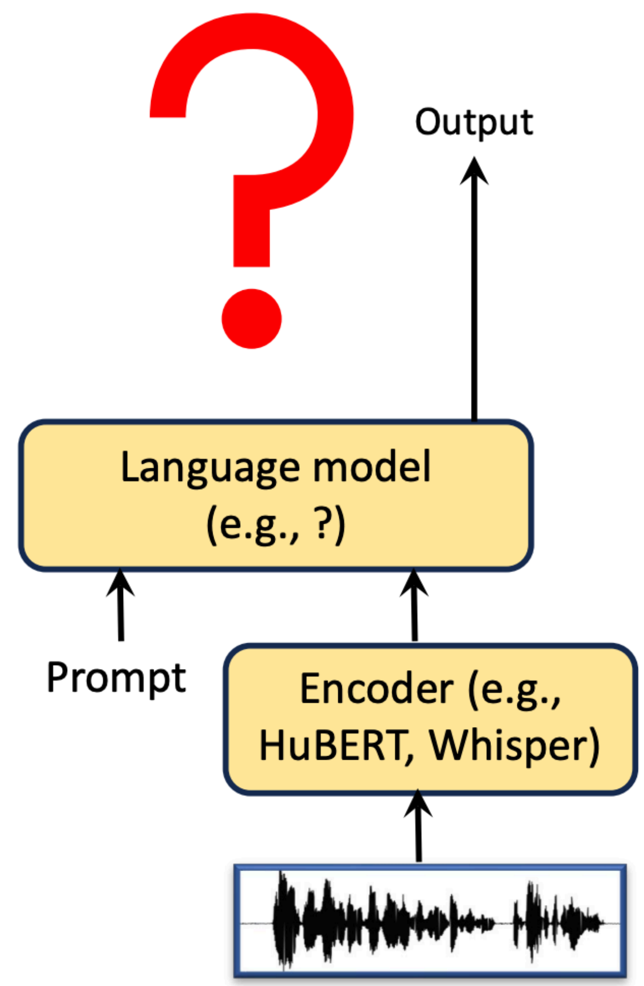
The task-specific model era (- 2020)



The speech encoder era (2020 -)



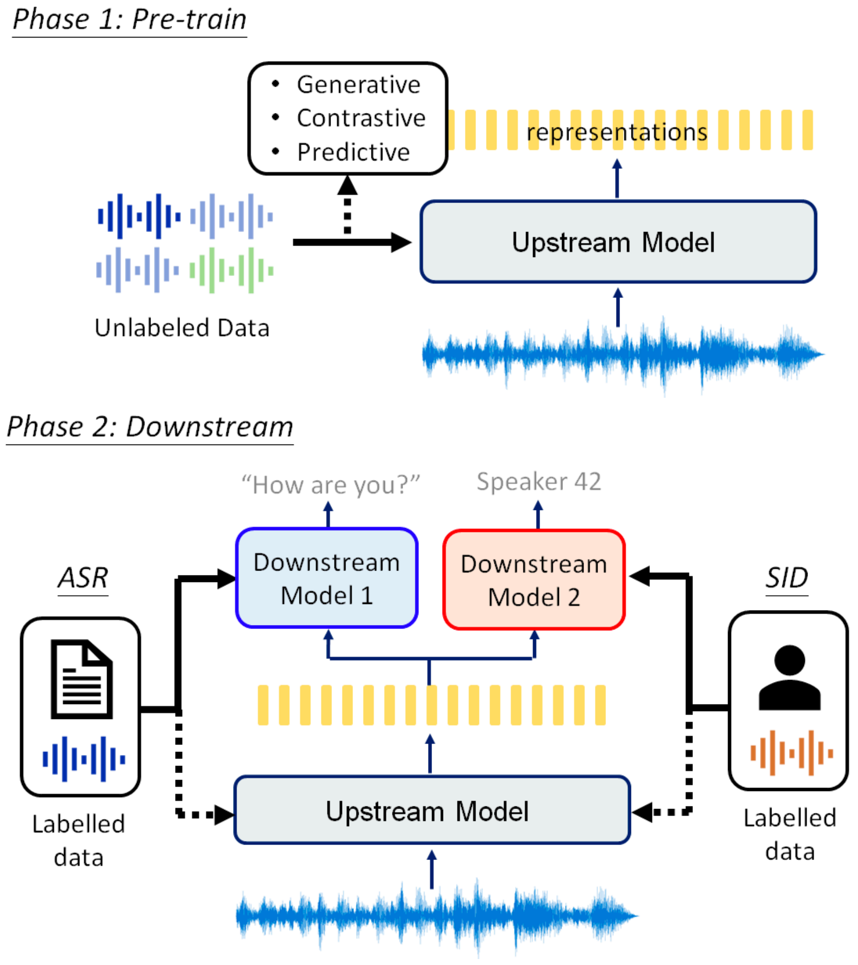
The spoken large language model era (2024? -)



Self-supervised learning

The setting:

1. A family of downstream tasks in mind *(The main motivational difference)*.
2. Lots of unlabeled in-domain data.
3. A small fraction of labeled data.



Self-supervised learning

One basic procedure:

1. Design a "pretext" task over the input space. Examples:
   + Predict the future part of the signal given the past
   + Predict a masked portion of the signal given the unmasked one.
   + Reconstruct the input through a quantization or extreme compression.
   + Distinguish similar "positive" samples from different "negative" ones.

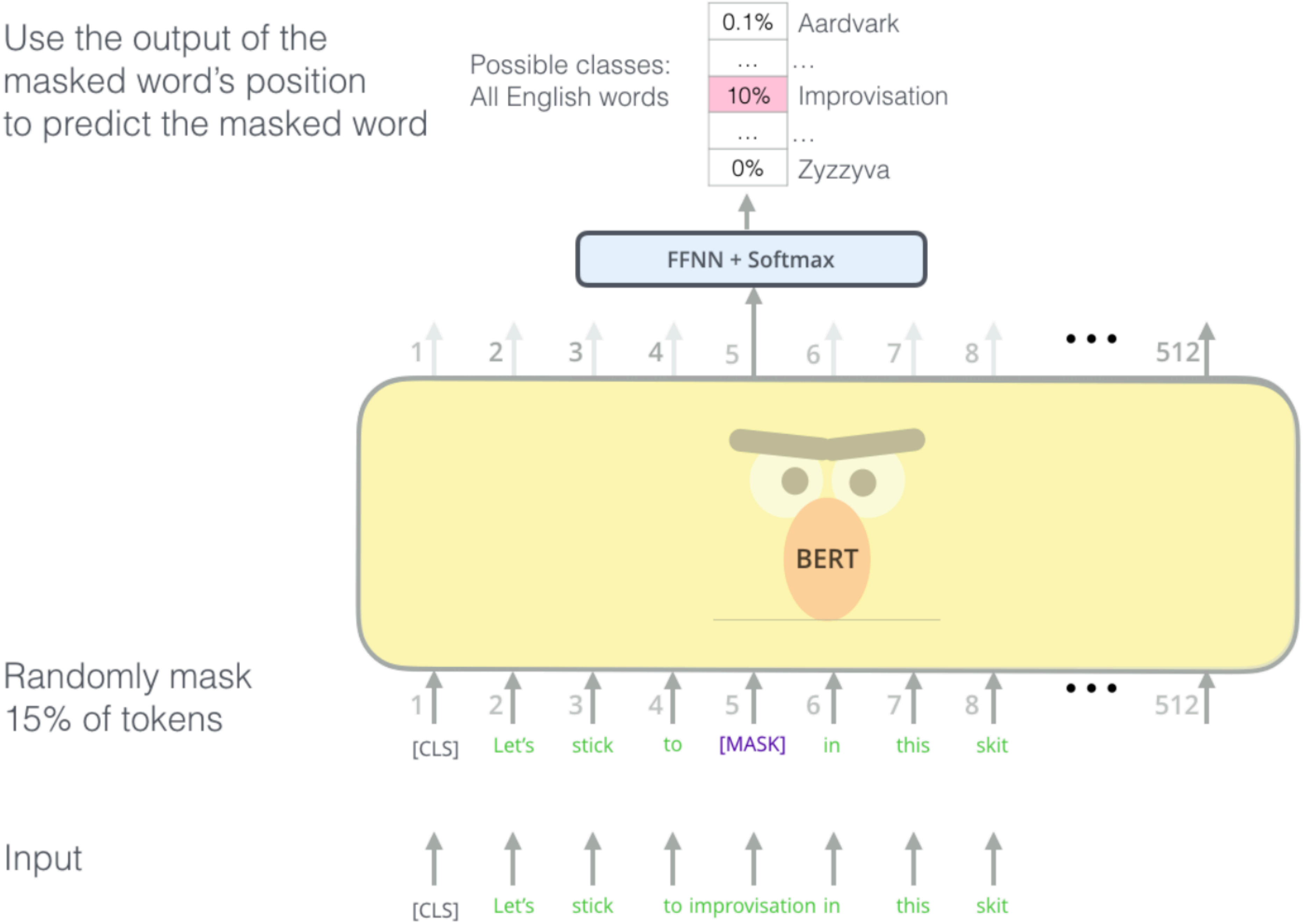
2. Use the pretext task for training the neural network using the unlabeled data.
3. Fine-tune the pre-trained network on the labeled data.

Self Supervised Learning

Sometimes learners set up tasks for themselves to solve…

- Even if we only have unlabeled data, we may be able to define "pretext tasks" from the data alone
- A good pretext task is one that requires us to represent the "useful" information in the input in order to solve it well

Self Supervised Learning -Text



What makes speech representation learning unique?

- Speech inputs have a variable number of lexical units per sequence.

What makes speech representation learning unique?

- Speech inputs have a variable number of lexical units per sequence.
- Speech is a long sequence that doesn't have segment boundaries.

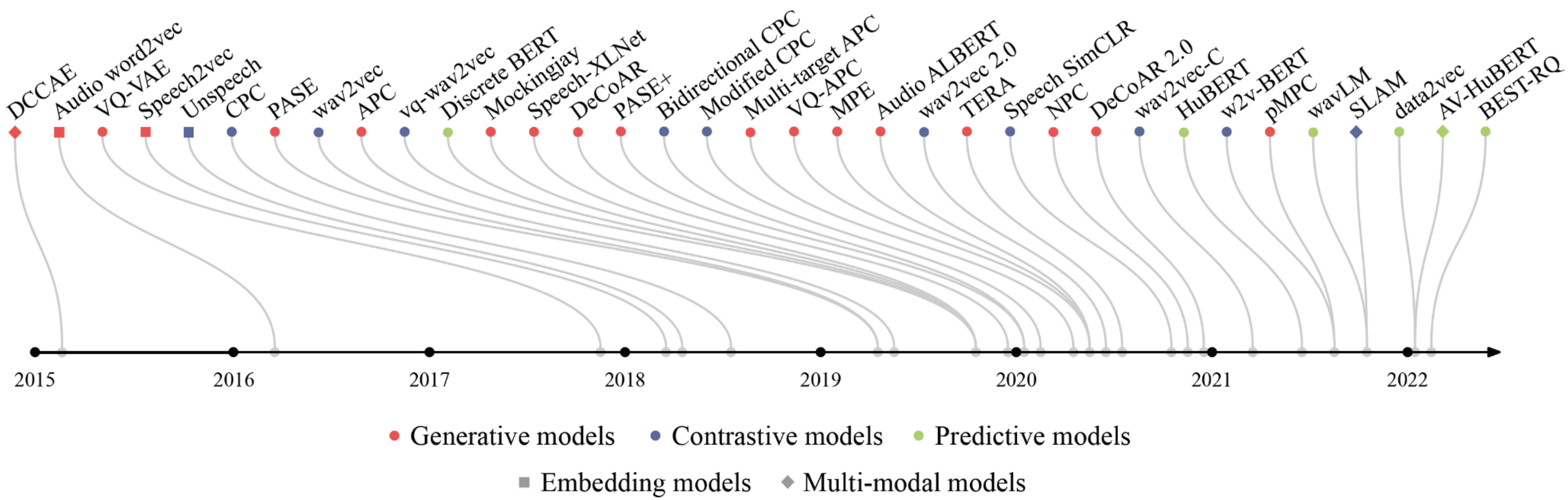
What makes speech representation learning unique?

- Speech inputs have a variable number of lexical units per sequence.
- Speech is a long sequence that doesn't have segment boundaries.
- Speech is continuous without a predefined dictionary of units to explicitly model in the self-supervised setting.

What makes speech representation learning unique?

- Speech inputs have a variable number of lexical units per sequence.
- Speech is a long sequence that doesn't have segment boundaries.
- Speech is continuous without a predefined dictionary of units to explicitly model in the self-supervised setting.
- Speech processing tasks might require orthogonal information, e.g., ASR and Speaker ID.

SSL methods for Speech



Contrastive approaches

Predictive approaches

Generative approaches

Contrastive Predictive Coding (CPC)

van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC

van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC

- The first successful representation learning approach for speech data.

van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC

- The first successful representation learning approach for speech data.
- It triggered lots of research in speech representation learning.

van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC

- Distinguish correct (positive) samples from wrong (negative) ones.

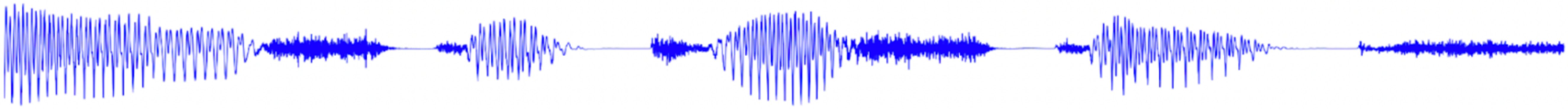
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC

- Distinguish correct (positive) samples from wrong (negative) ones.
- **But, how do we choose positive and negative examples?**

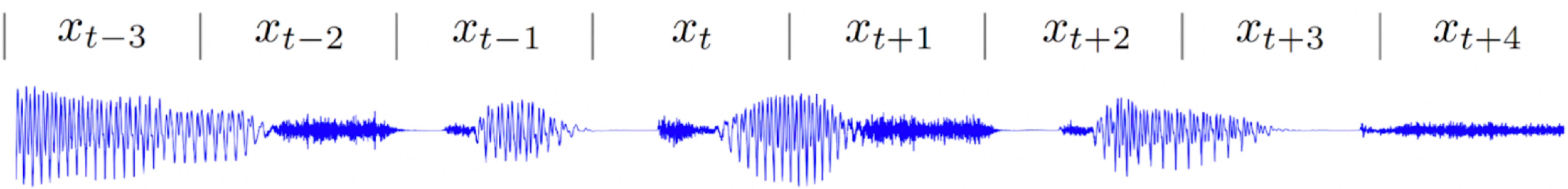
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC



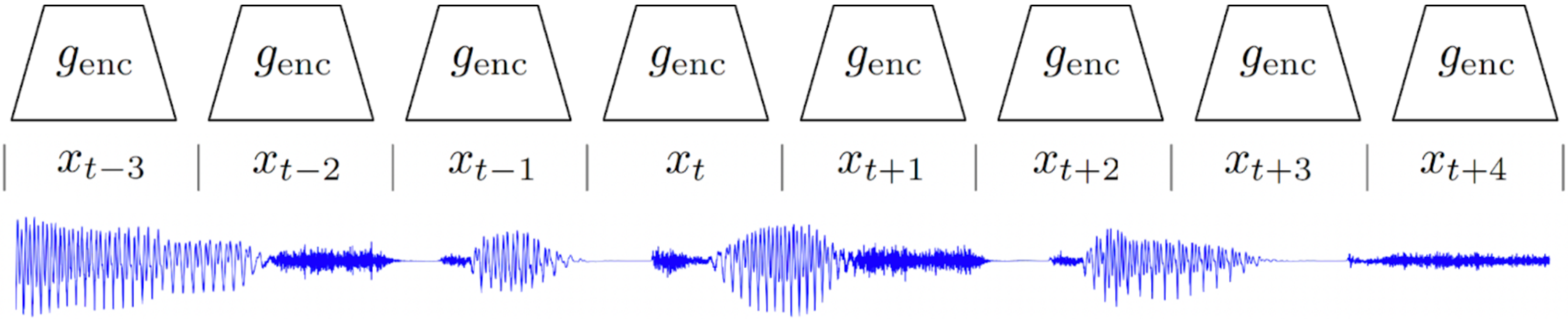
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC



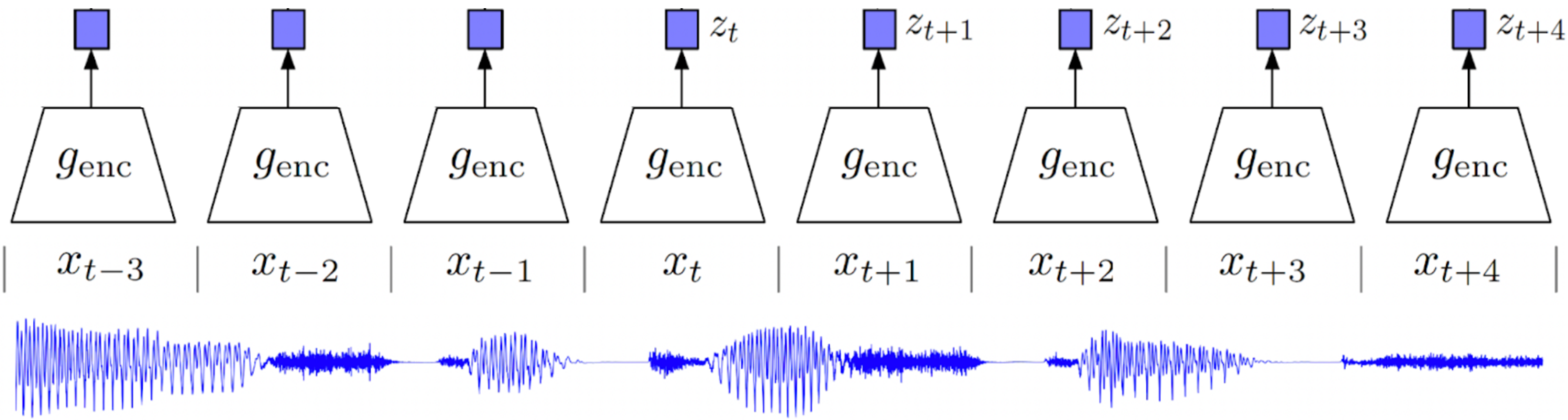
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC



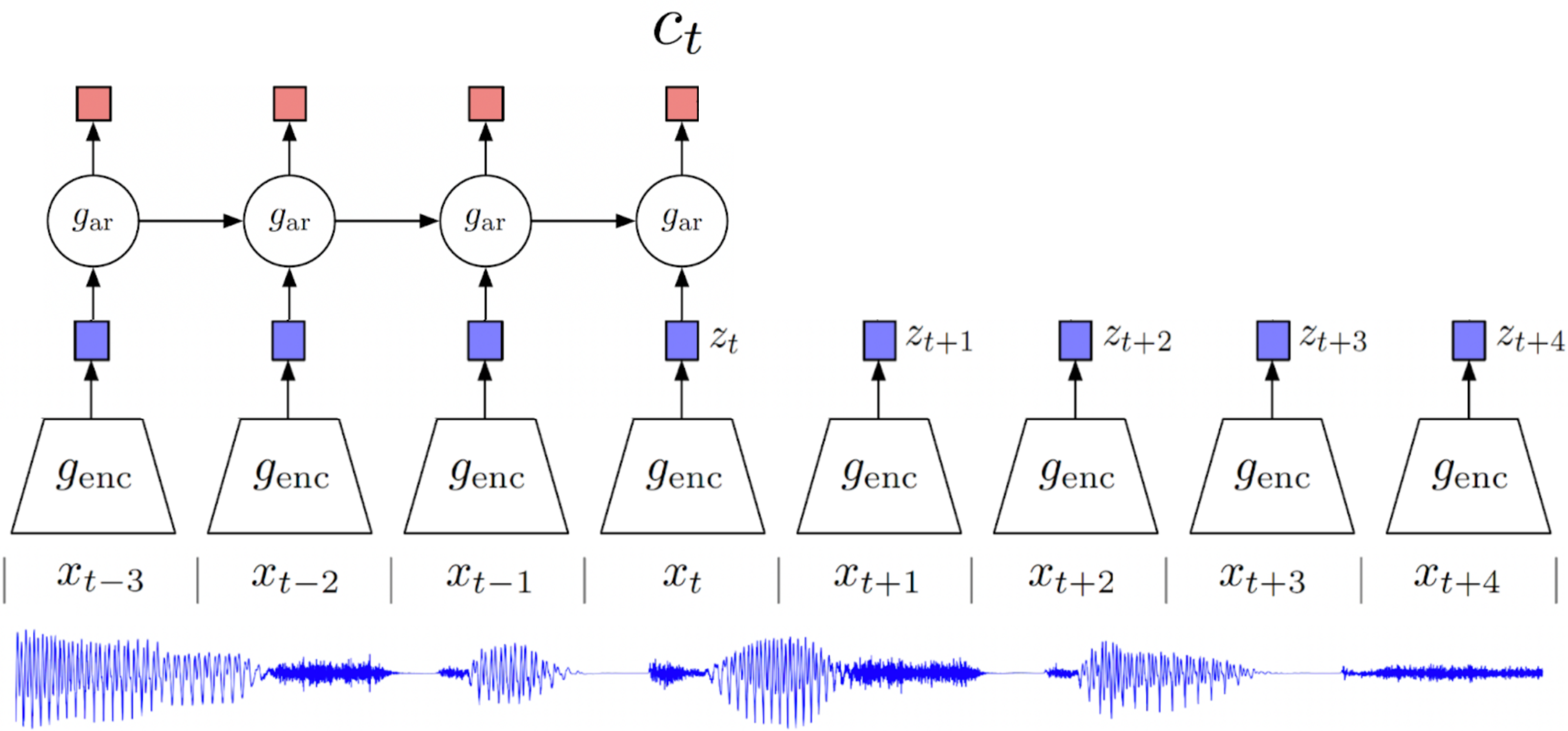
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC



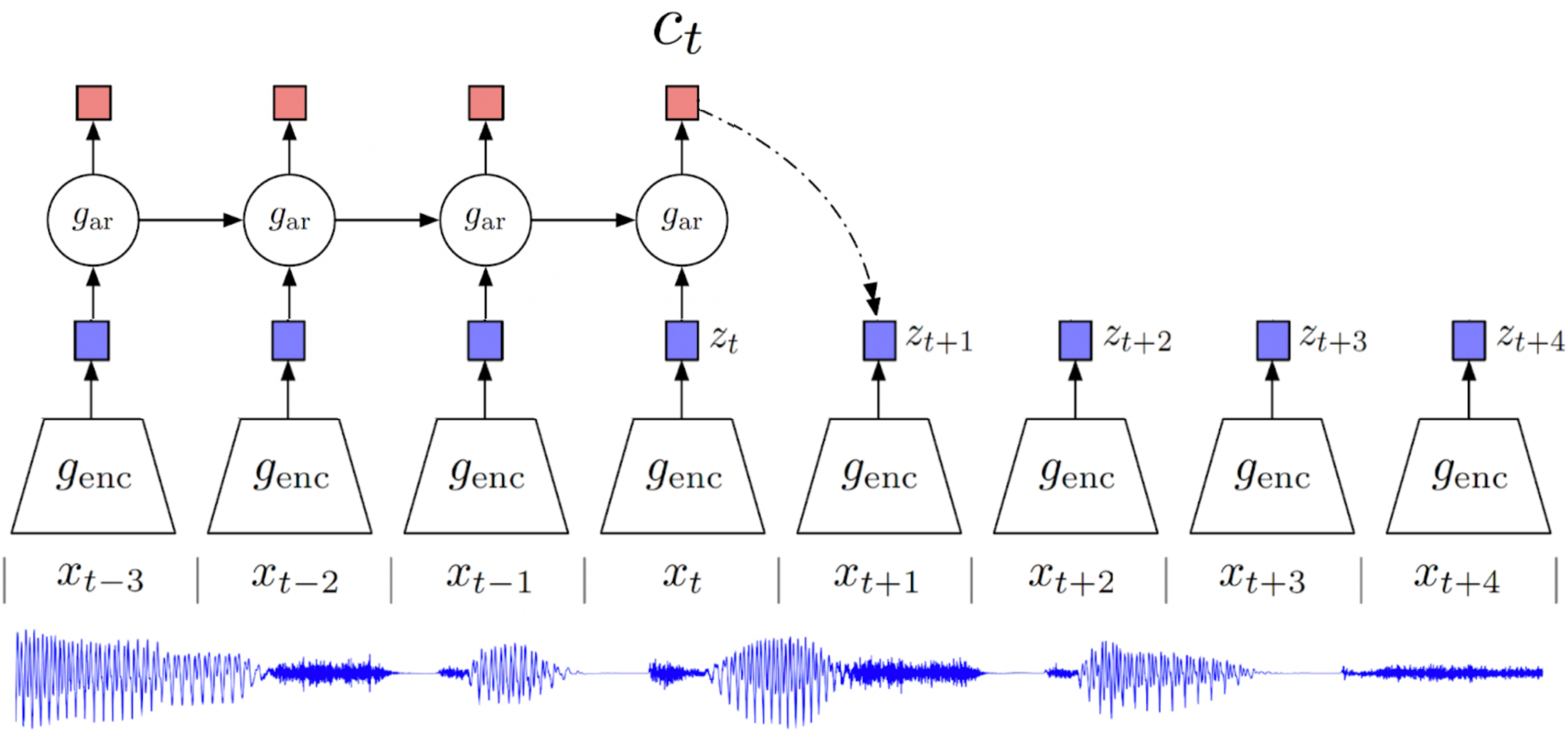
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

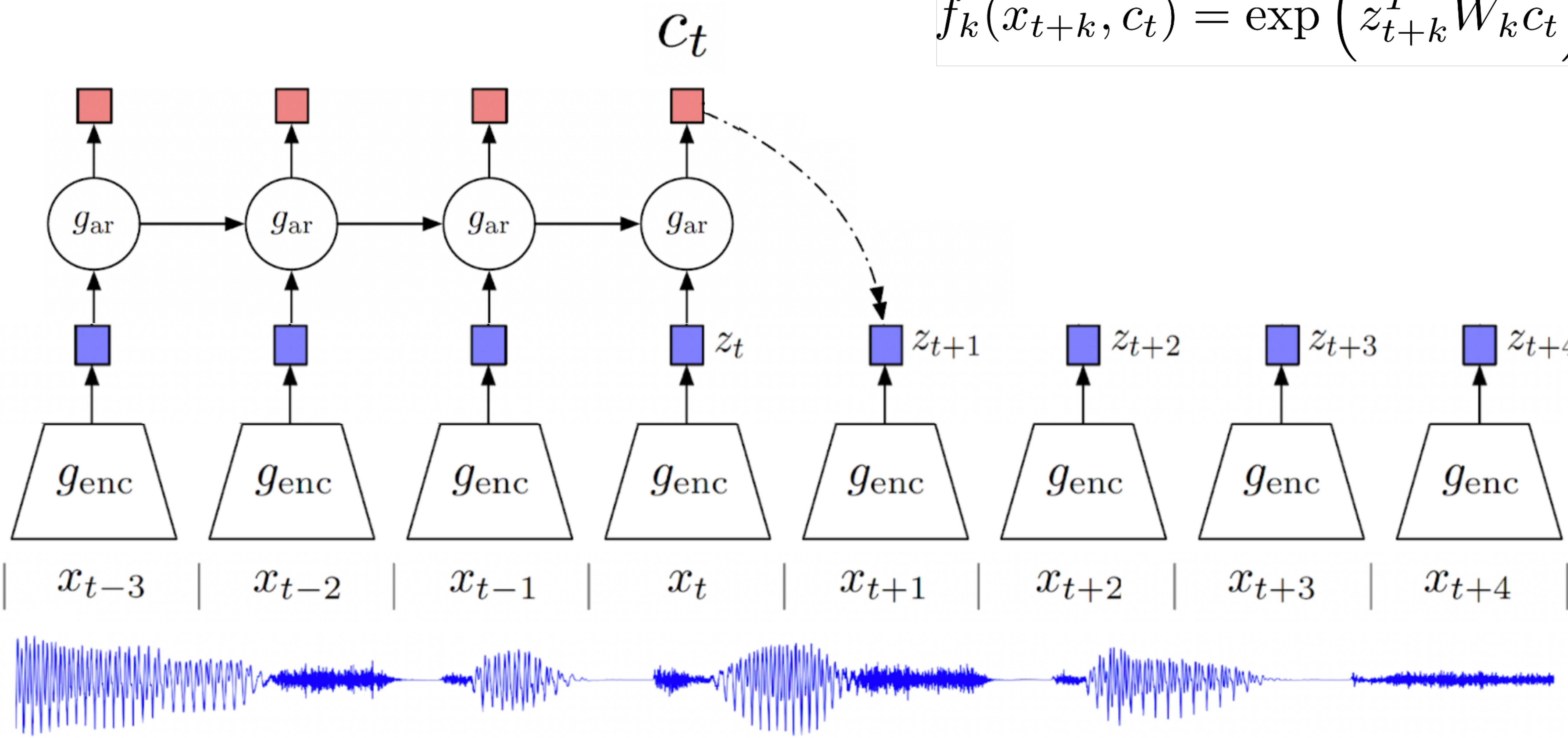
CPC: The pretext task



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

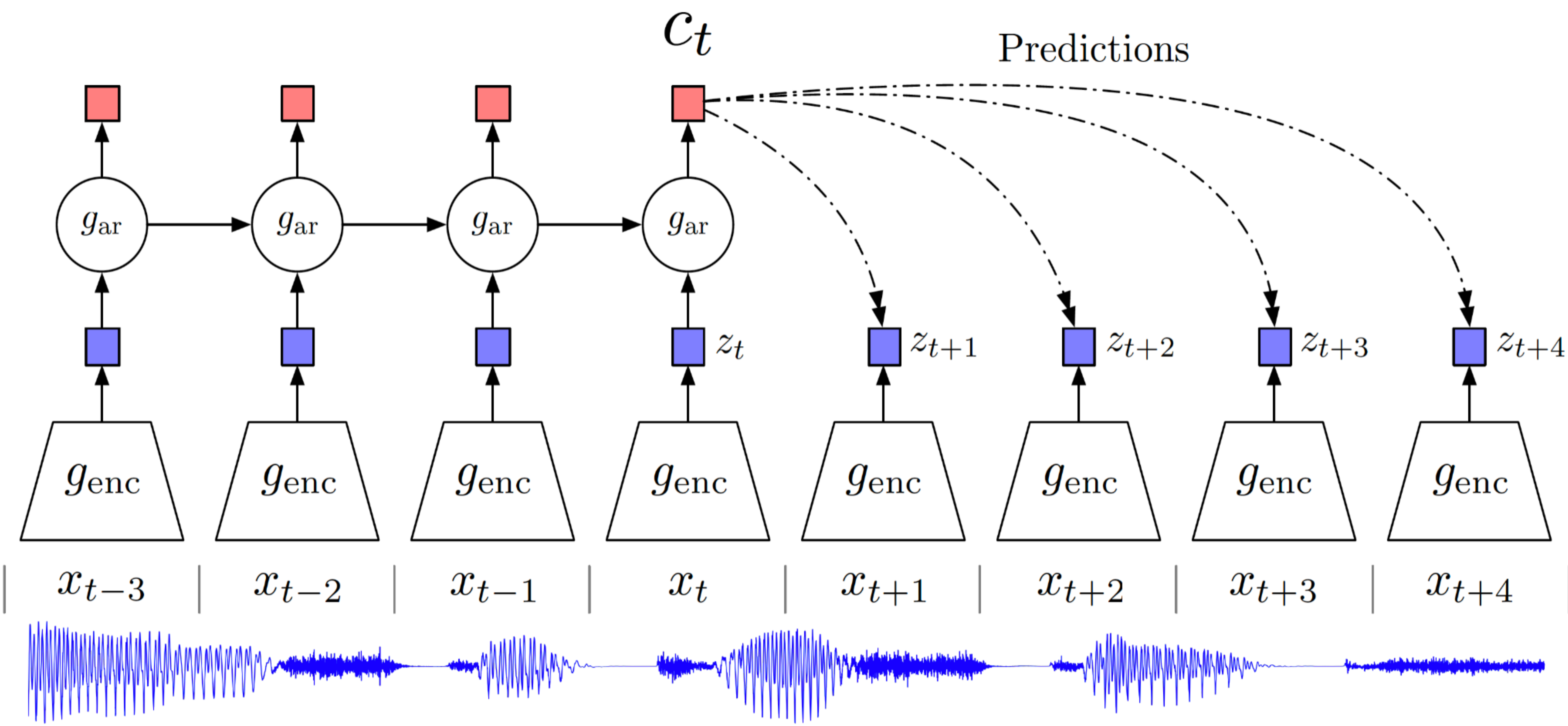
CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$



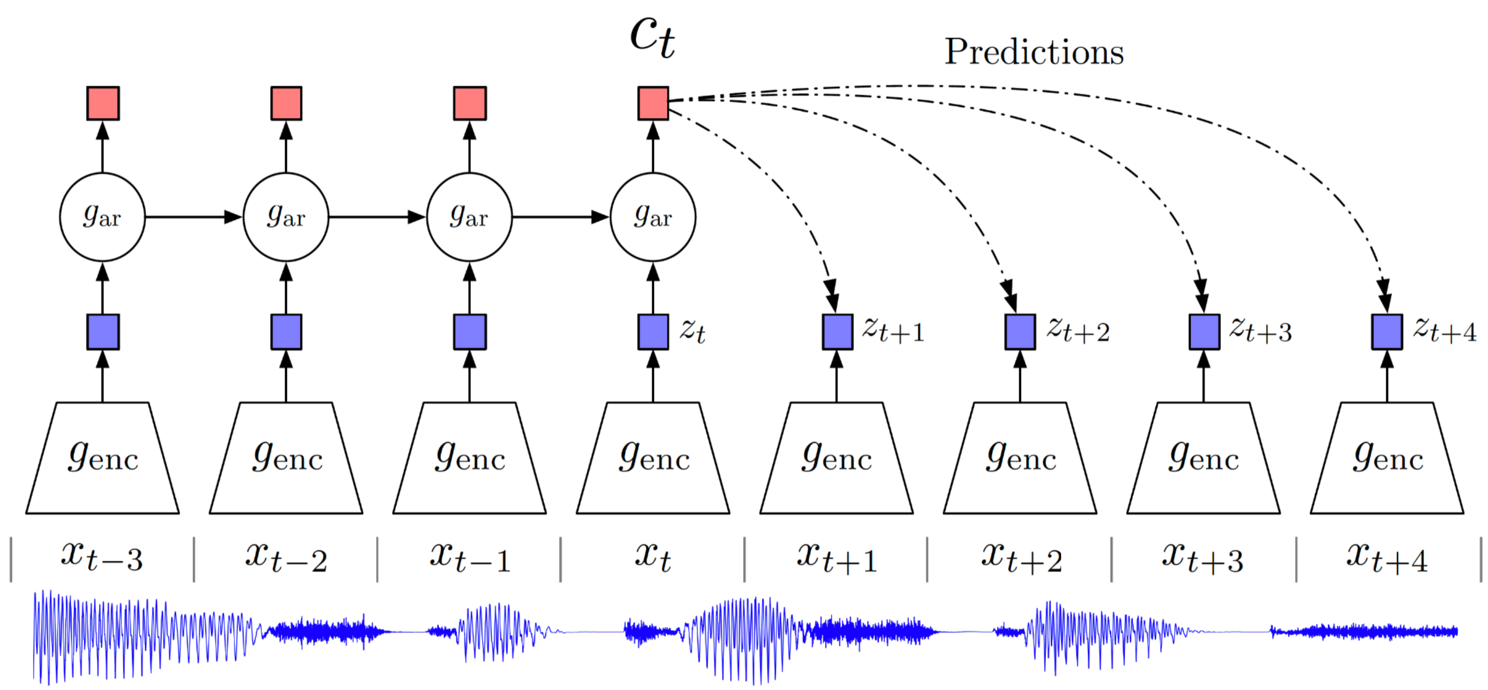
van den Oord et al, 2019 "Representation Learning with Contrastive Predictive Coding"

CPC: The pretext task



van den Oord et al, 2019 "Representation Learning with Contrastive Predictive Coding"

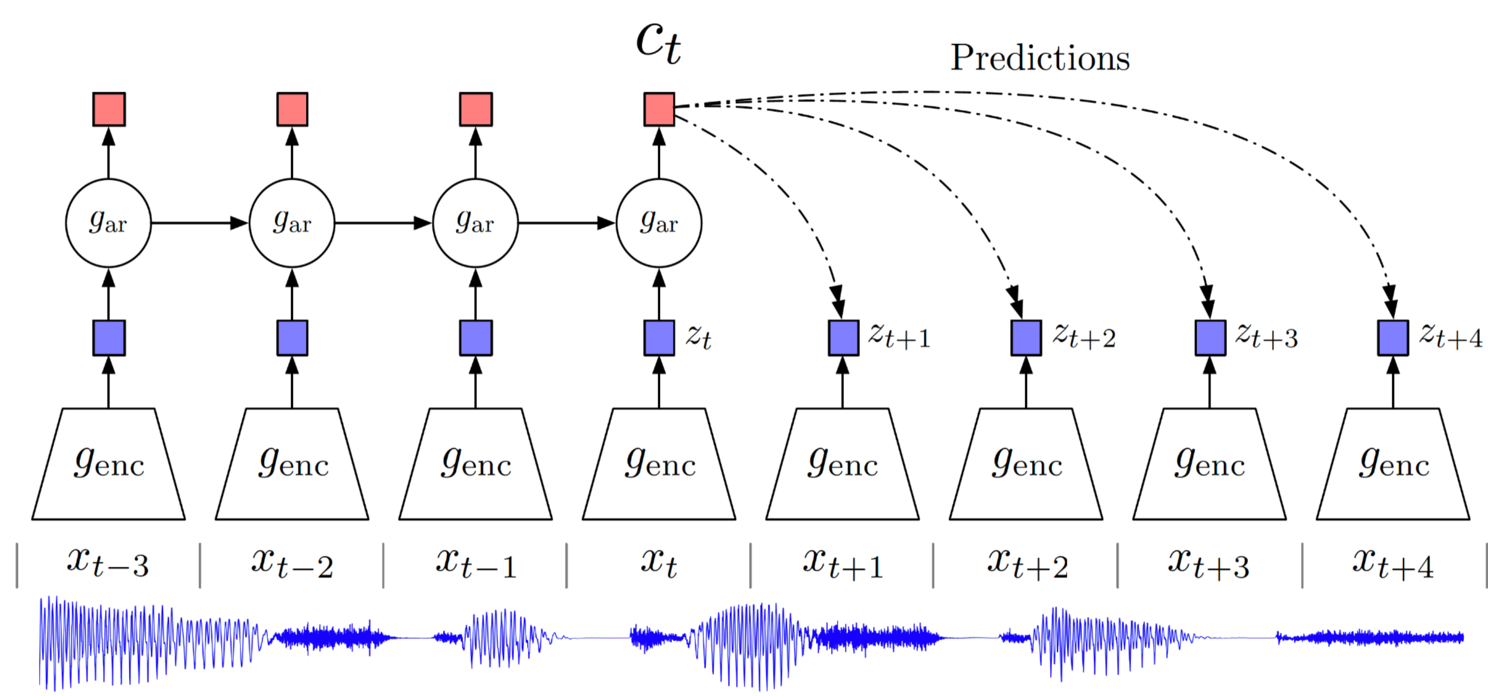
CPC: The pretext task



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

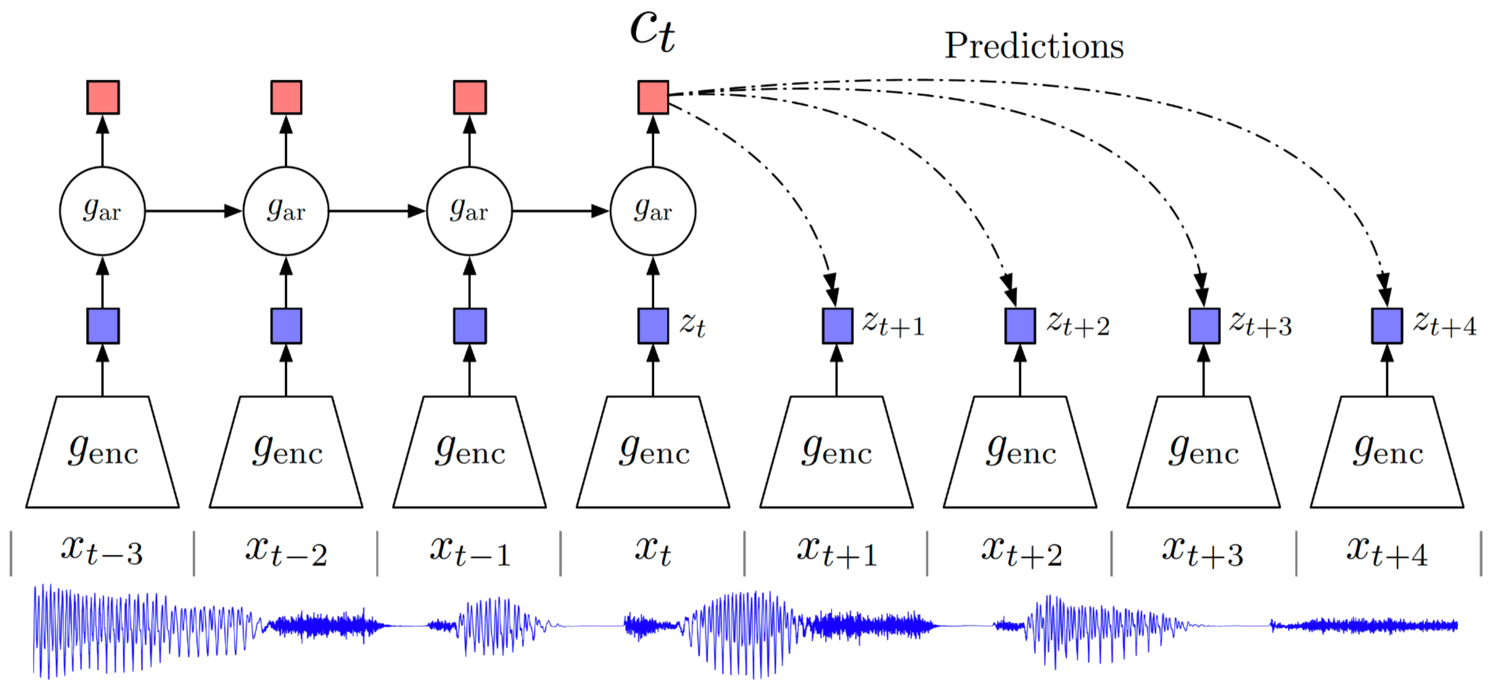


van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

$$\mathcal{L}_\mathrm{N} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$



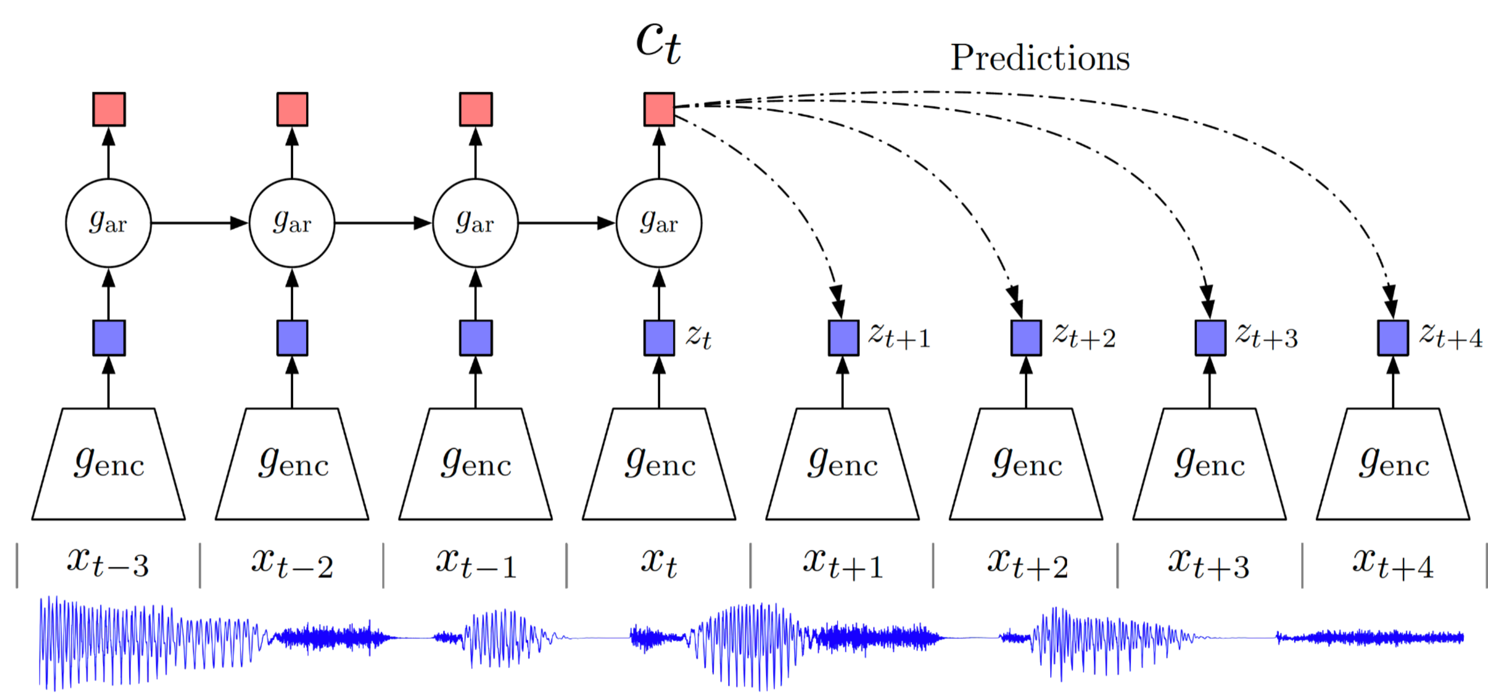
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

- InfoNCE maximizes the mutual information between the input signal and the learned latent variables C.

$$\mathcal{L}_{\mathrm{N}} = - \mathop{\mathbb{E}}_{X} \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$



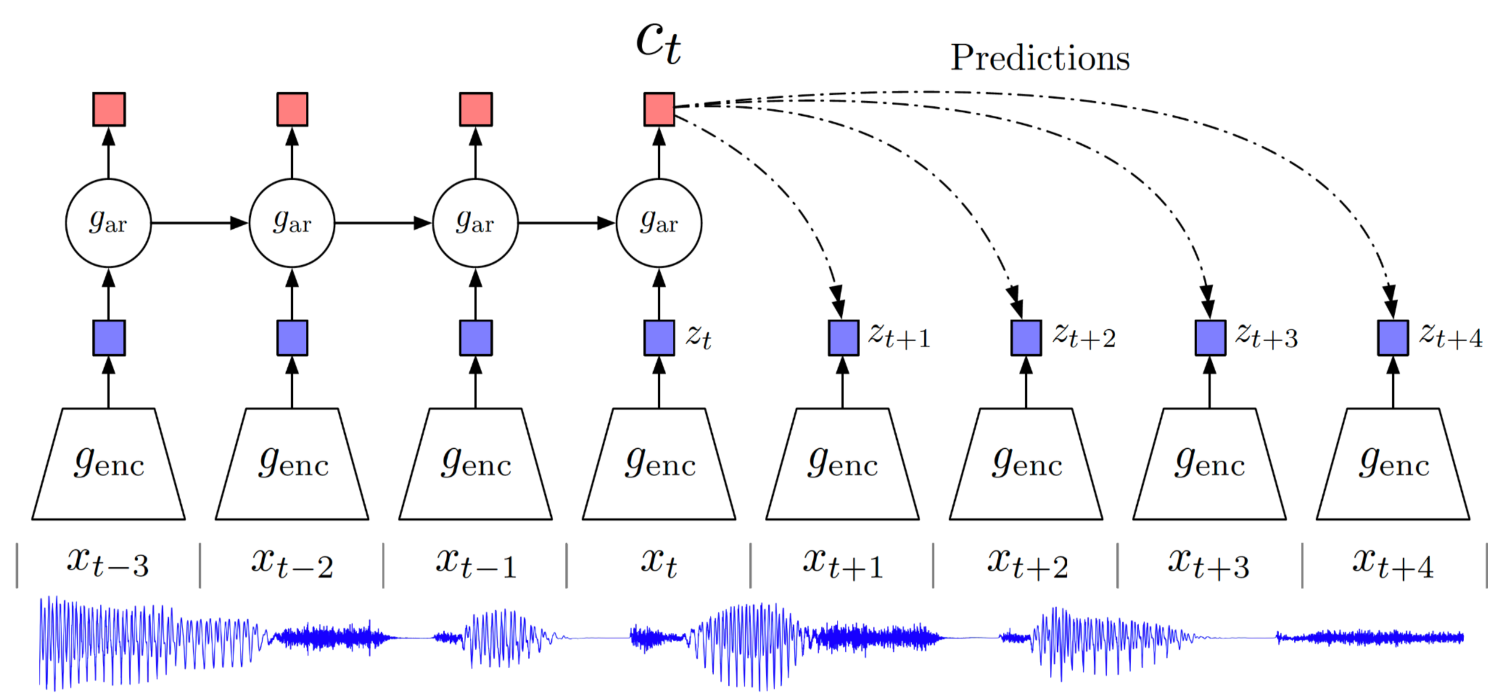
van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

CPC: The pretext task

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

- InfoNCE maximizes the mutual information between the input signal and the learned latent variables C.
- Strategies for sampling negative and positive examples determine the nature of representations, e.g., whether they are good for ASR or Speaker ID.



van den Oord et al, 2019 “Representation Learning with Contrastive Predictive Coding”

- The CPC approach inspired many follow up work

- The CPC approach inspired many follow up work:
  - With better architectures and normalization for stable training.

Schneider et. al., 2019 "wav2vec: Unsupervised Pre-training for Speech Recognition"
Kawakami et. al., 2020 "Learning Robust and Multilingual Speech Representations"
Rivi`ere et. al., 2020 "Unsupervised pretraining transfers well across languages"

- The CPC approach inspired many follow up work:
  - With better architectures and normalization for stable training.
  - Bidirectional autoregressive components.

Schneider et. al., 2019 "wav2vec: Unsupervised Pre-training for Speech Recognition"
Kawakami et. al., 2020 "Learning Robust and Multilingual Speech Representations"
Rivi`ere et. al., 2020 "Unsupervised pretraining transfers well across languages"

- The CPC approach inspired many follow up work:
  - With better architectures and normalization for stable training.
  - Bidirectional autoregressive components.
  - Which investigates the multilingual transfer of representations.
  - ….

Schneider et. al., 2019 "wav2vec: Unsupervised Pre-training for Speech Recognition"
Kawakami et. al., 2020 "Learning Robust and Multilingual Speech Representations"
Rivi`ere et. al., 2020 "Unsupervised pretraining transfers well across languages"

wav2vec 2.0

wav2vec 2.0: The pretext task

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- The first approach to show significant improvements for low-resource ASR.

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

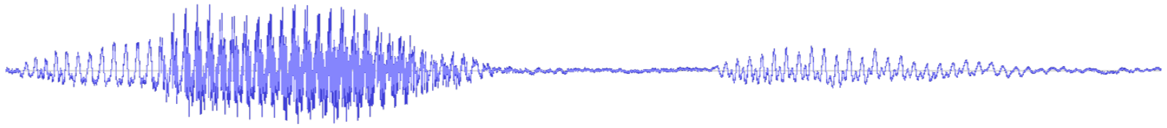
- The first approach to show significant improvements for low-resource ASR.
- Impressive results on multilingual representations.

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- The first approach to show significant improvements for low-resource ASR.
- Impressive results on multilingual representations.
- Strong performance on a wide range of downstream speech tasks.

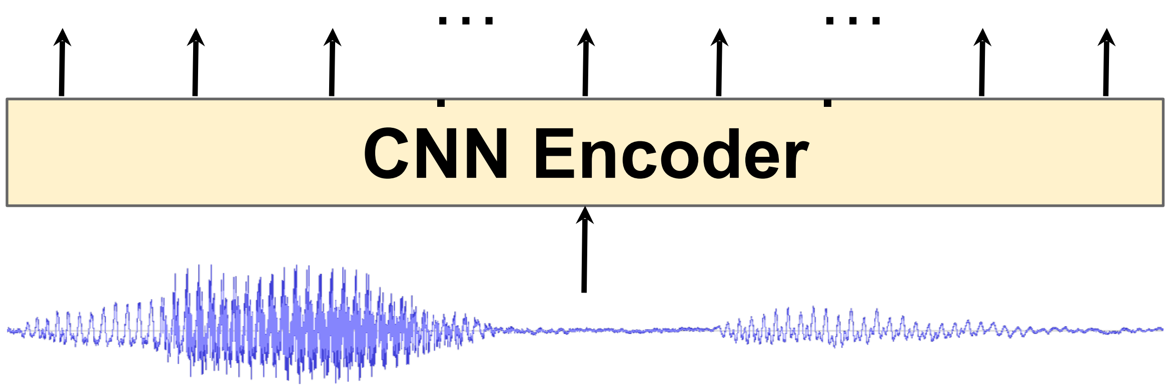
Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task



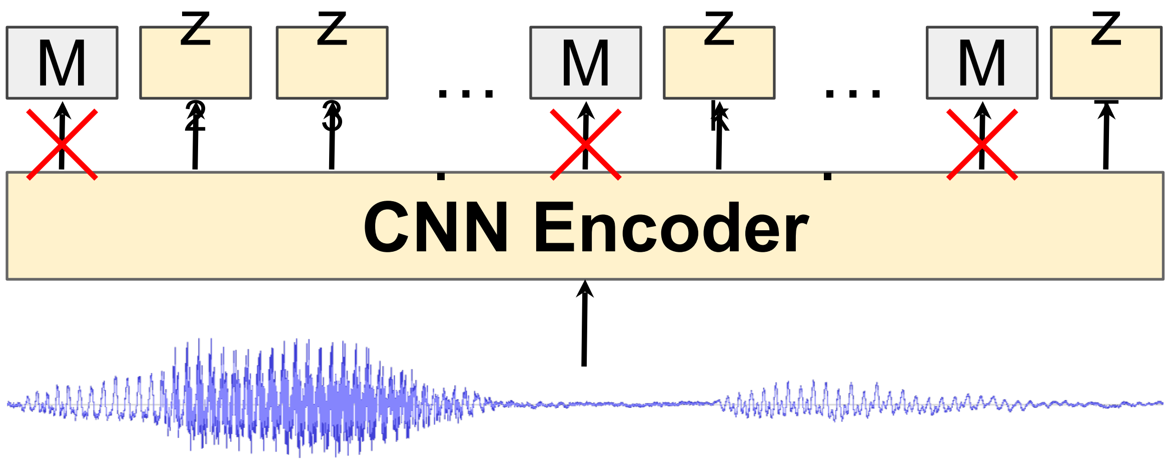
Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task



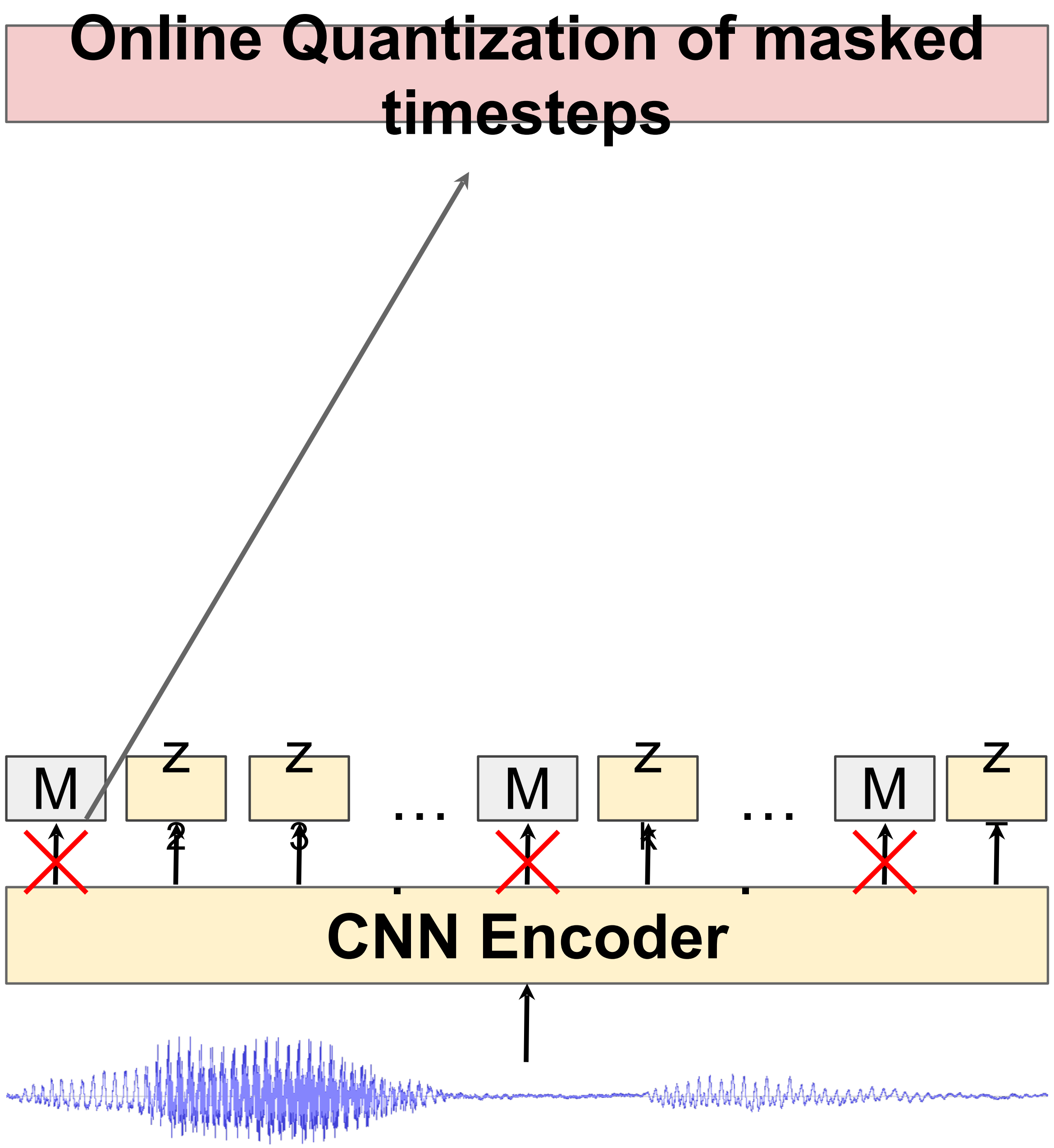
Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task



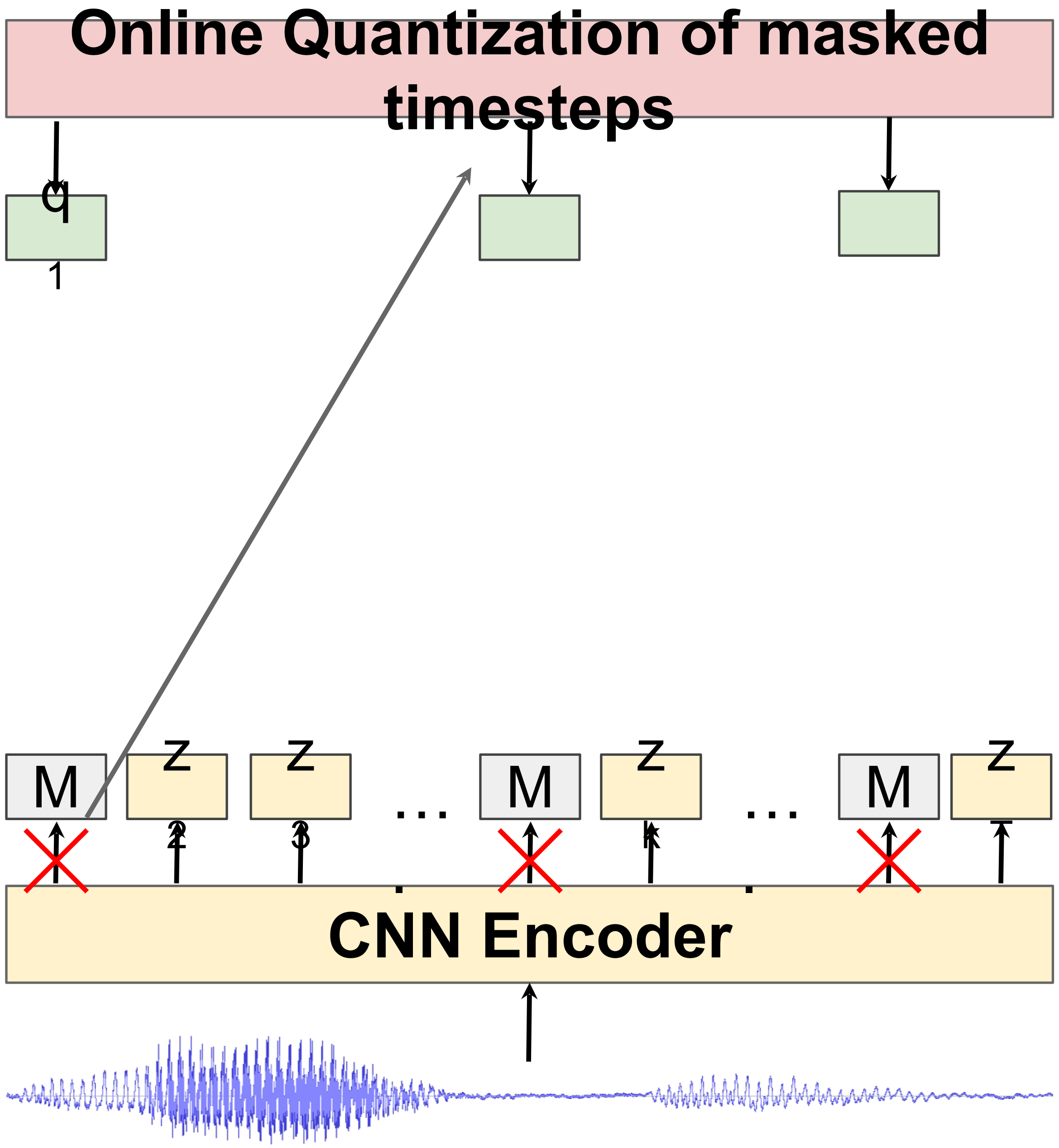
Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task



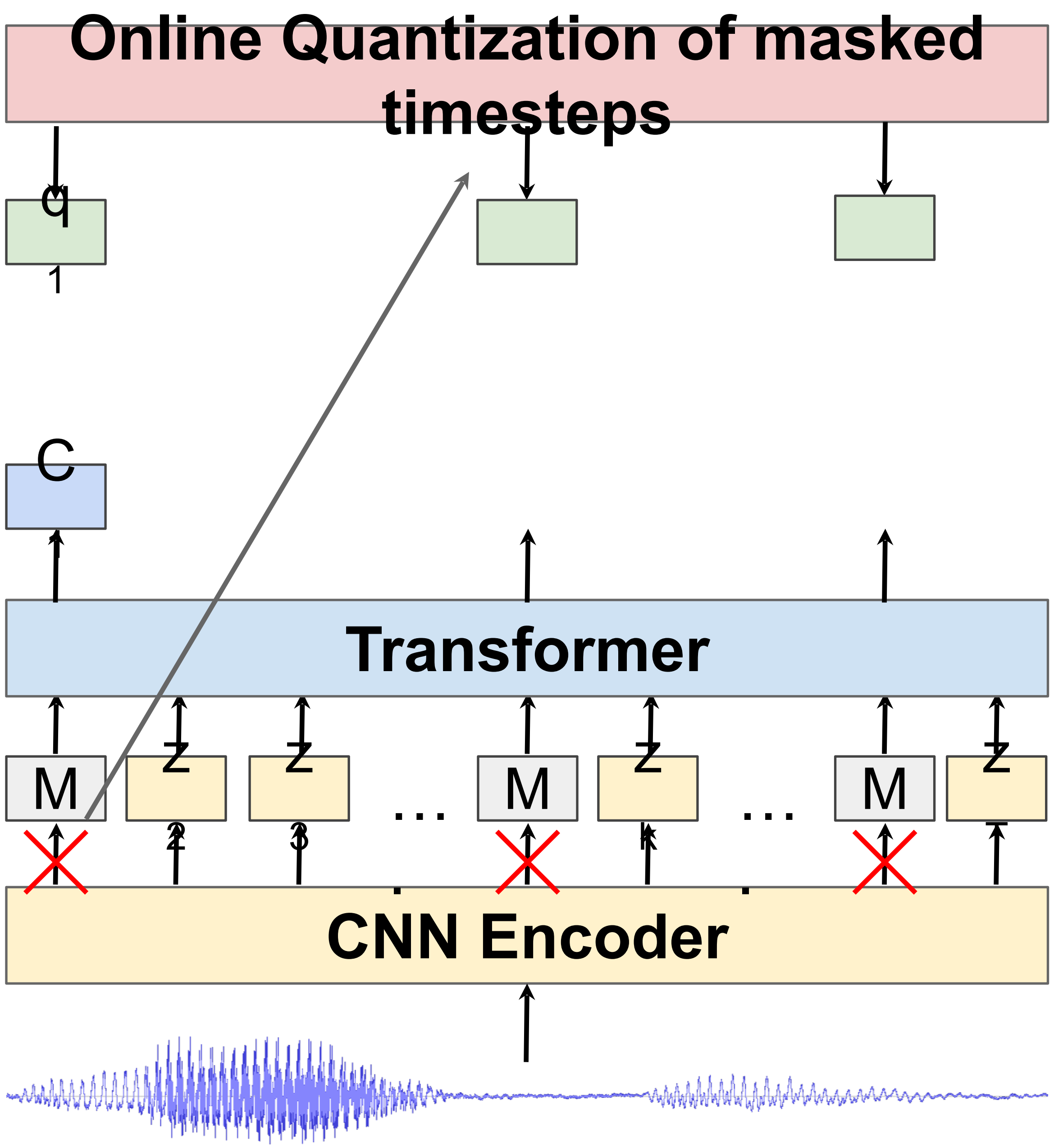
Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task



Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

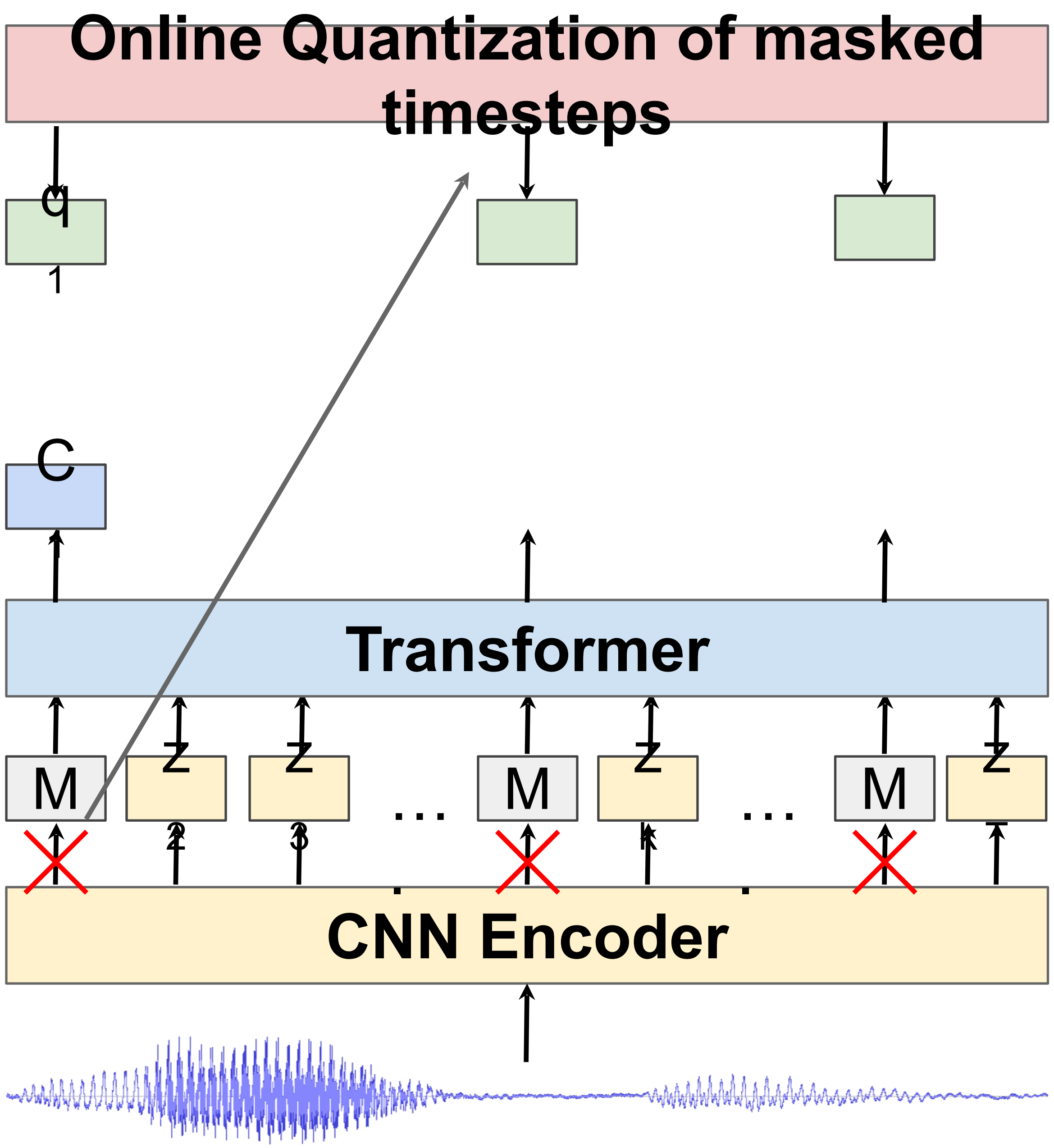
wav2vec 2.0: The pretext task



Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task

- The goal is to maximize the similarity between the learned contextual representation and the quantized input features at the same position.

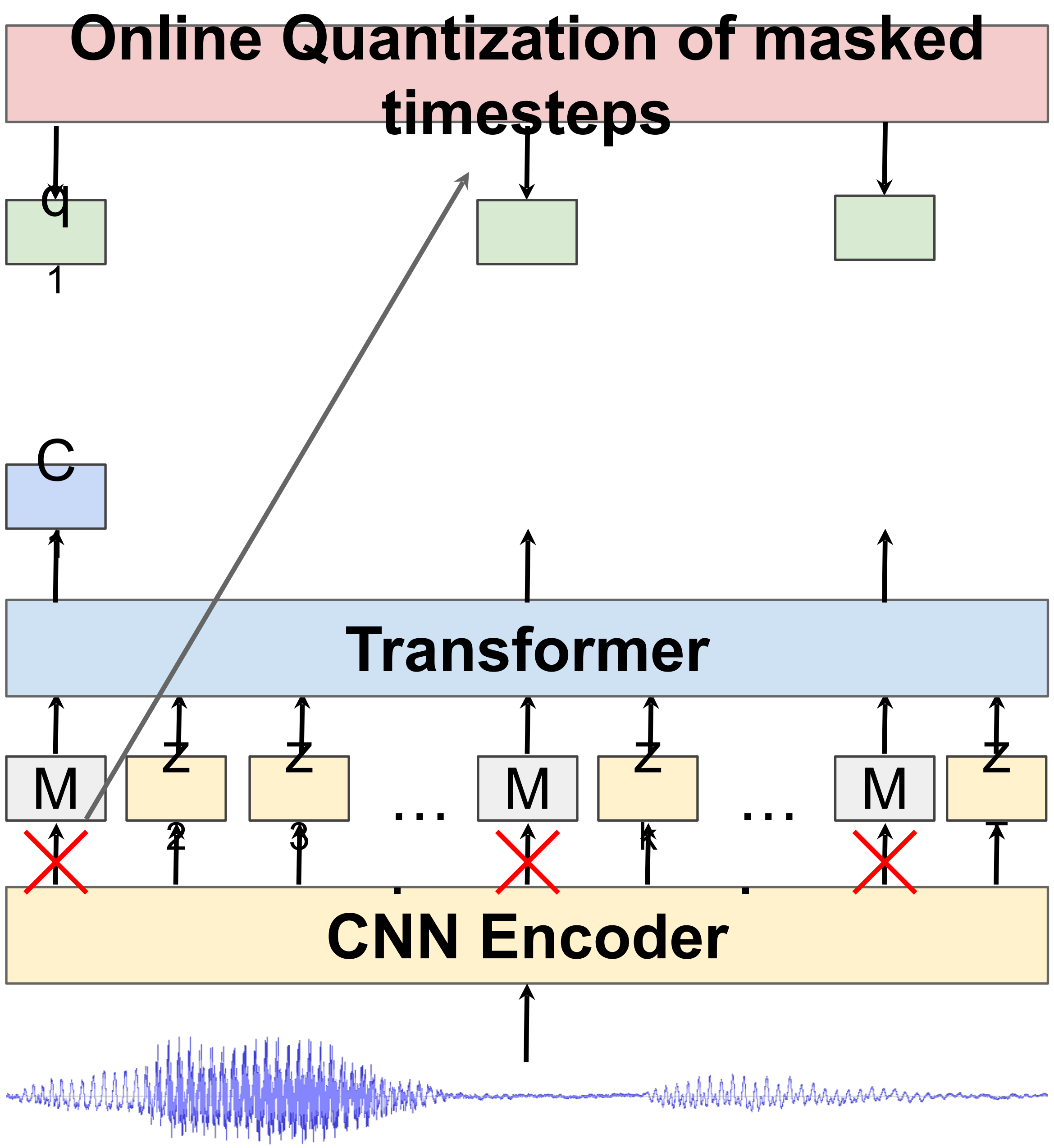


Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: The pretext task

- The goal is to maximize the similarity between the learned contextual representation and the quantized input features

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$



Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: Gumbel softmax

- The model learns an online quantization of audio representations using a Gumbel softmax.

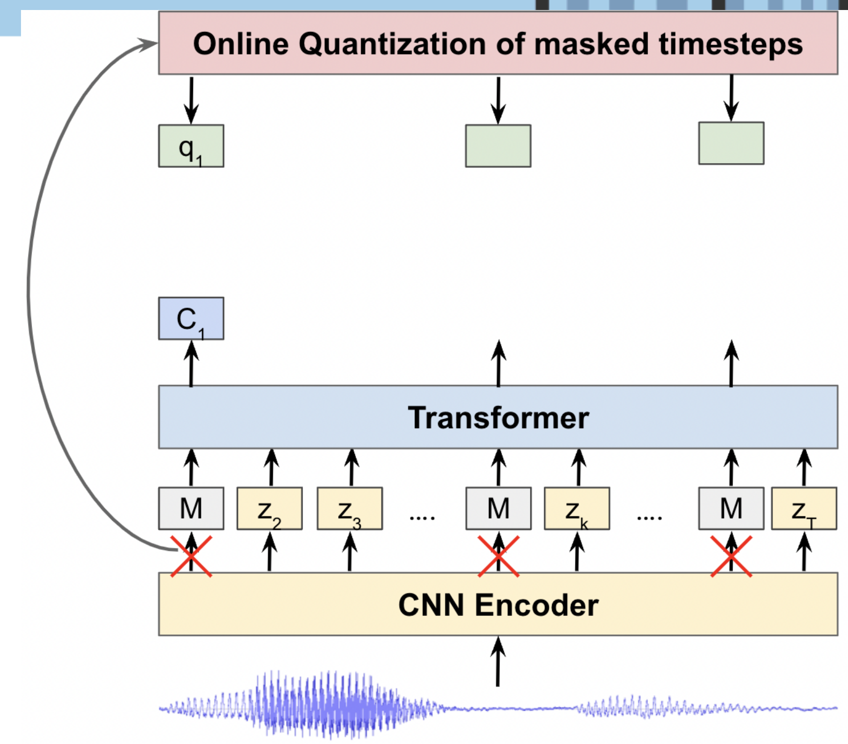
```
gumbels = (logits + gumbels) / tau  # ~Gumbel(logits,tau)
y_soft = gumbels.softmax(dim)

if hard:
    # Straight through.
    index = y_soft.max(dim, keepdim=True)[1]
    y_hard = torch.zeros_like(logits, memory_format=torch.legacy_contiguous_format).scatter_(dim, index, 1.0)
    ret = y_hard - y_soft.detach() + y_soft
else:
    # Reparametrization trick.
    ret = y_soft
return ret
```

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: Implementation details

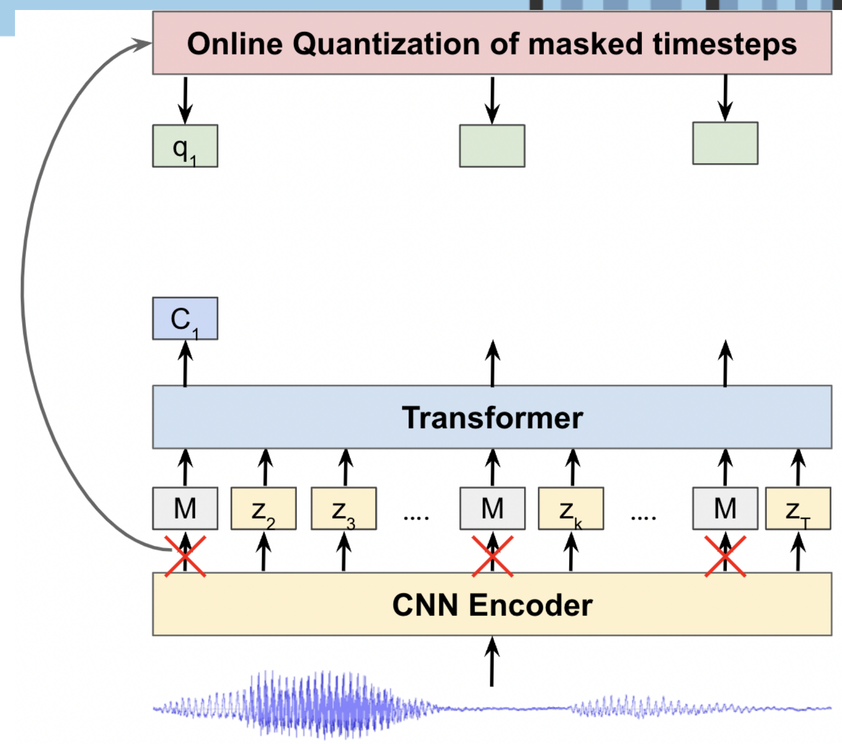
- Product quantization with more than one codebook yields better results.



Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: Implementation details

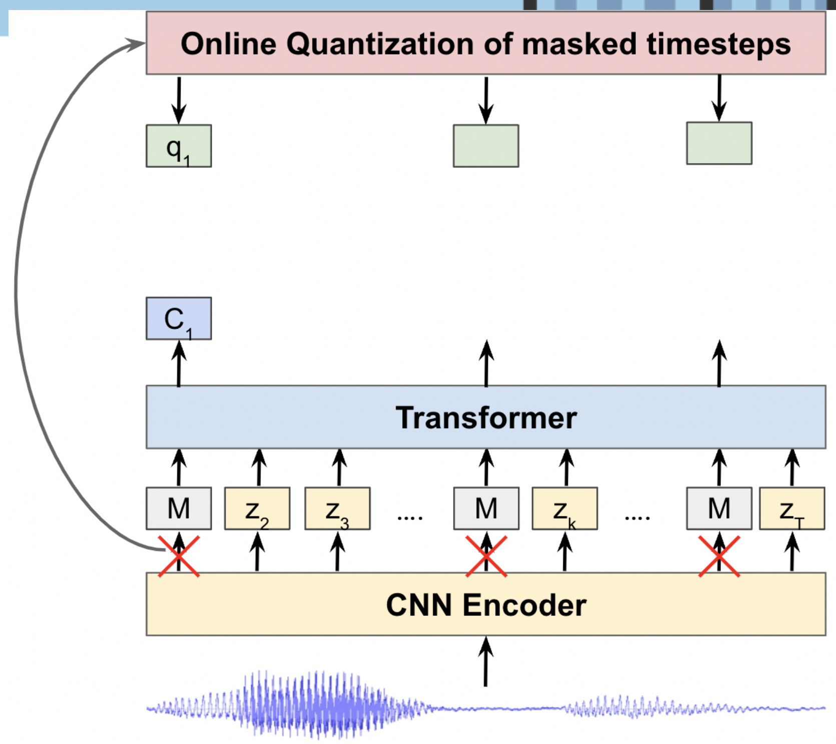
- Product quantization with more than one codebook yields better results.
- An entropy loss is added to the Gumbel softmax distribution to maximize codebook diversity.



Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: Implementation details

- Product quantization with more than one codebook yields better results.
- An entropy loss is added to the Gumbel softmax distribution to maximize codebook diversity.
- Negative examples are chosen from masked segments in the same utterance that don't belong to the same codeword.



Baevski et al, 2020 "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations"

- The first approach to get into single-digit WER on Librispeech test-other using only 10 mins of labels.

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- The first approach to get into single-digit WER on Librispeech test-other using only 10 mins of labels.

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
10 min labeled						
Discrete BERT [4]	LS-960	4-gram	15.7	24.1	16.3	25.2
BASE	LS-960	4-gram	8.9	15.7	9.1	15.6
		Transf.	6.6	13.2	6.9	12.9
LARGE	LS-960	Transf.	6.6	10.6	6.8	10.8
	LV-60k	Transf.	4.6	7.9	4.8	8.2

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- It is the first self-supervised approach to produce competitive results compared to semi-supervised learning approaches.

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

wav2vec 2.0: Results

- It is com app ...arning

Model	Unlabeled data	LM	dev clean	dev other	test clean	test other
Supervised						
CTC Transf [51]	-	CLM+Transf.	2.20	4.94	2.47	5.45
S2S Transf. [51]	-	CLM+Transf.	2.10	4.79	2.33	5.17
Transf. Transducer [60]	-	Transf.	-	-	2.0	4.6
ContextNet [17]	-	LSTM	1.9	3.9	1.9	4.1
Conformer [15]	-	LSTM	2.1	4.3	1.9	3.9
Semi-supervised						
CTC Transf. + PL [51]	LV-60k	CLM+Transf.	2.10	4.79	2.33	4.54
S2S Transf. + PL [51]	LV-60k	CLM+Transf.	2.00	3.65	2.09	4.11
Iter. pseudo-labeling [58]	LV-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
Noisy student [42]	LV-60k	LSTM	1.6	3.4	1.7	3.4
This work						
LARGE - from scratch	-	Transf.	1.7	4.3	2.1	4.6
BASE	LS-960	Transf.	1.8	4.7	2.1	4.8
LARGE	LS-960	Transf.	1.7	3.9	2.0	4.1
	LV-60k	Transf.	1.6	3.0	1.8	3.3

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- It is ... com ... arning ... app

Model	Unlabeled data	LM	dev clean	dev other	test clean	test other
Supervised						
CTC Transf [51]	-	CLM+Transf.	2.20	4.94	2.47	5.45
S2S Transf. [51]	-	CLM+Transf.	2.10	4.79	2.33	5.17
Transf. Transducer [60]	-	Transf.	-	-	2.0	4.6
ContextNet [17]	-	LSTM	1.9	3.9	1.9	4.1
Conformer [15]	-	LSTM	2.1	4.3	1.9	3.9
Semi-supervised						
CTC Transf. + PL [51]	LV-60k	CLM+Transf.	2.10	4.79	2.33	4.54
S2S Transf. + PL [51]	LV-60k	CLM+Transf.	2.00	3.65	2.09	4.11
Iter. pseudo-labeling [58]	LV-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
Noisy student [42]	LV-60k	LSTM	1.6	3.4	1.7	3.4
This work						
LARGE - from scratch	-	Transf.	1.7	4.3	2.1	4.6
BASE	LS-960	Transf.	1.8	4.7	2.1	4.8
LARGE	LS-960	Transf.	1.7	3.9	2.0	4.1
	LV-60k	Transf.	1.6	3.0	1.8	3.3

Baevski et al, 2020 “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations”

- wav2vec 2.0 inspired many follow up work:

Conneau et. al., 2020 "Unsupervised Cross-lingual Representation Learning for Speech Recognition"
Sadhu et. al., 2021 "Wav2vec-C: A Self-supervised Model for Speech Representation Learning"
Chung et. al., 2021 "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training"
Babu et. al., 2021 "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale"

- wav2vec 2.0 inspired many follow up work:
  - Multilingual pretraining.

Conneau et. al., 2020 "Unsupervised Cross-lingual Representation Learning for Speech Recognition"
Sadhu et. al., 2021 "Wav2vec-C: A Self-supervised Model for Speech Representation Learning"
Chung et. al., 2021 "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training"
Babu et. al., 2021 "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale"

- wav2vec 2.0 inspired many follow up work:
  - Multilingual pretraining.
  - With more effective quantization.

Conneau et. al., 2020 "Unsupervised Cross-lingual Representation Learning for Speech Recognition"
Sadhu et. al., 2021 "Wav2vec-C: A Self-supervised Model for Speech Representation Learning"
Chung et. al., 2021 "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training"
Babu et. al., 2021 "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale"

- wav2vec 2.0 inspired many follow up work:
  - Multilingual pretraining.
  - With more effective quantization.
  - Combining contrastive and predictive losses.
  - …

Conneau et. al., 2020 "Unsupervised Cross-lingual Representation Learning for Speech Recognition"
Sadhu et. al., 2021 "Wav2vec-C: A Self-supervised Model for Speech Representation Learning"
Chung et. al., 2021 "W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training"
Babu et. al., 2021 "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale"

Speech representation learning methods

Contrastive approaches

Predictive approaches

Generative approaches

Hidden Unit BERT (HuBERT)

HuBERT

Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT

- A simple method to apply BERT style representation learning for speech.

Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT

- A simple method to apply BERT style representation learning for speech.
- Matched or beat the SOTA on ASR while being the best for many speech tasks.

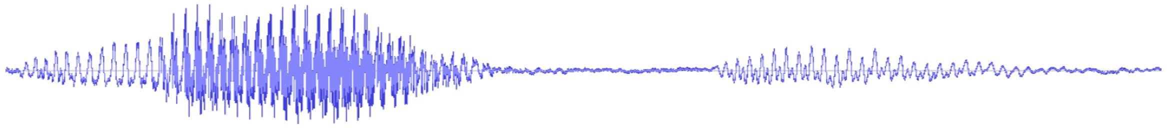
Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT

- A simple method to apply BERT style representation learning for speech.
- Matched or beat the SOTA on ASR while being the best for many speech tasks.
- With its high-quality discrete units, HuBERT facilitated Textless NLP research.

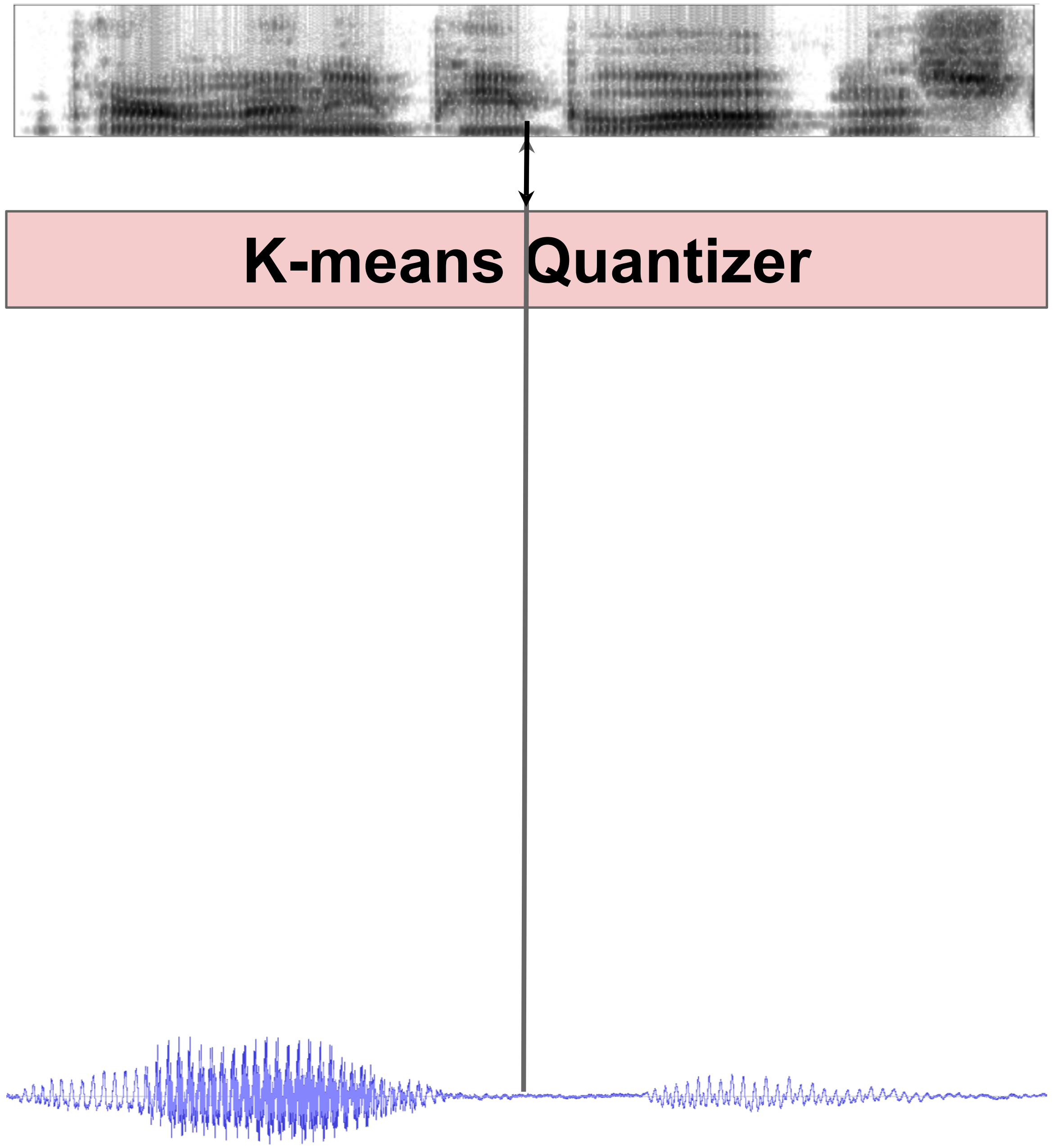
Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: The pretext task



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

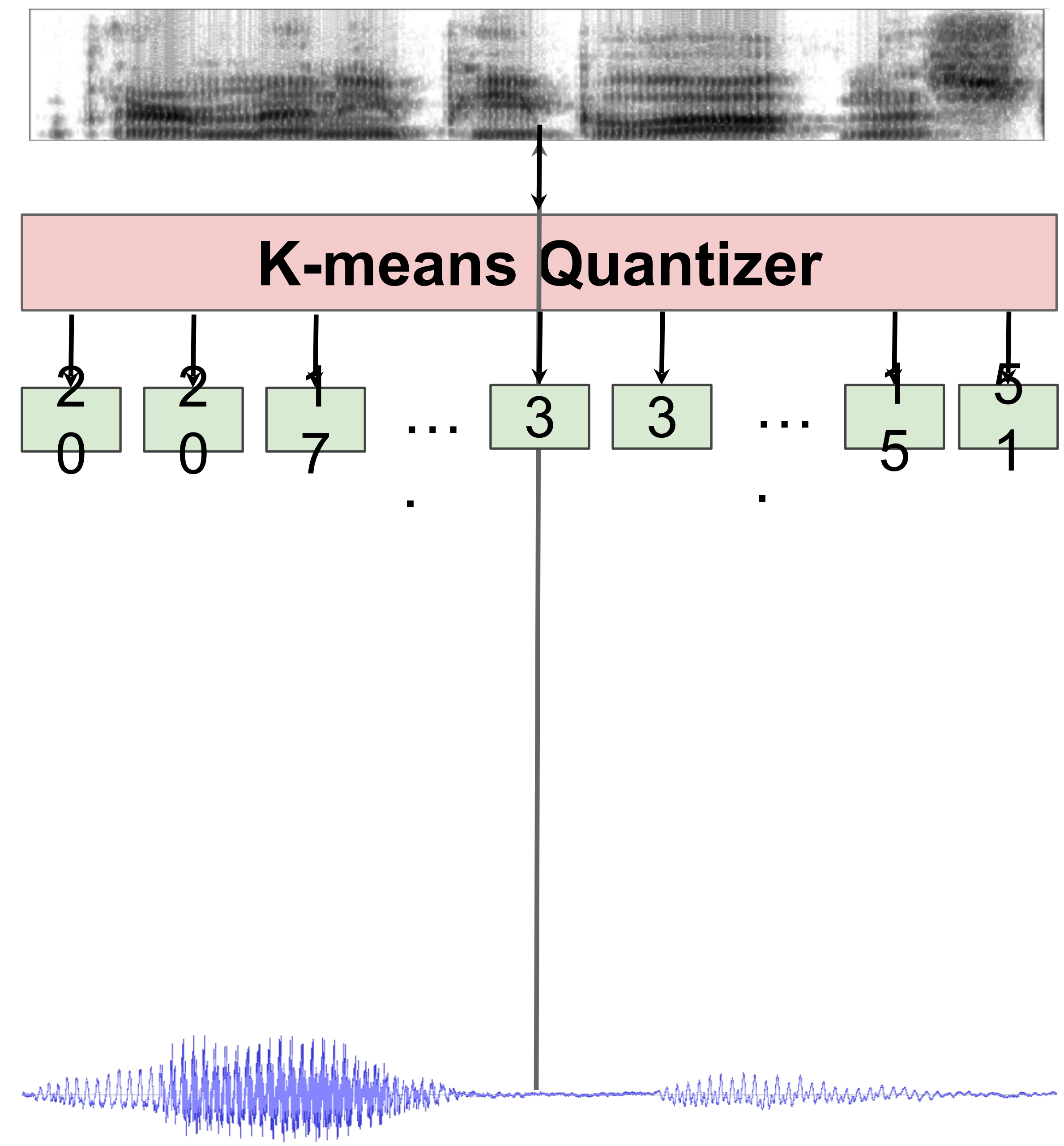
HuBERT: The pretext task



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

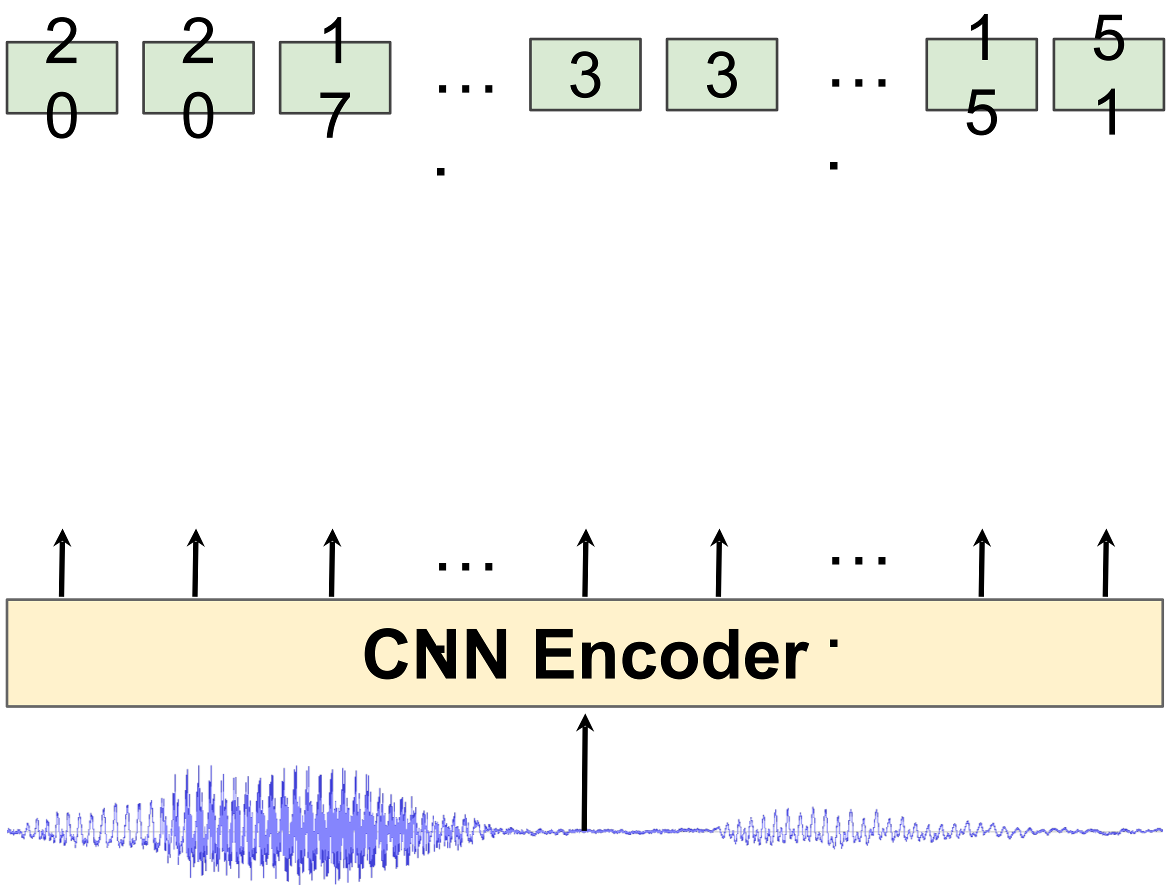
HuBERT: The pretext task

- The K-means quantizer produces frame-level labels.



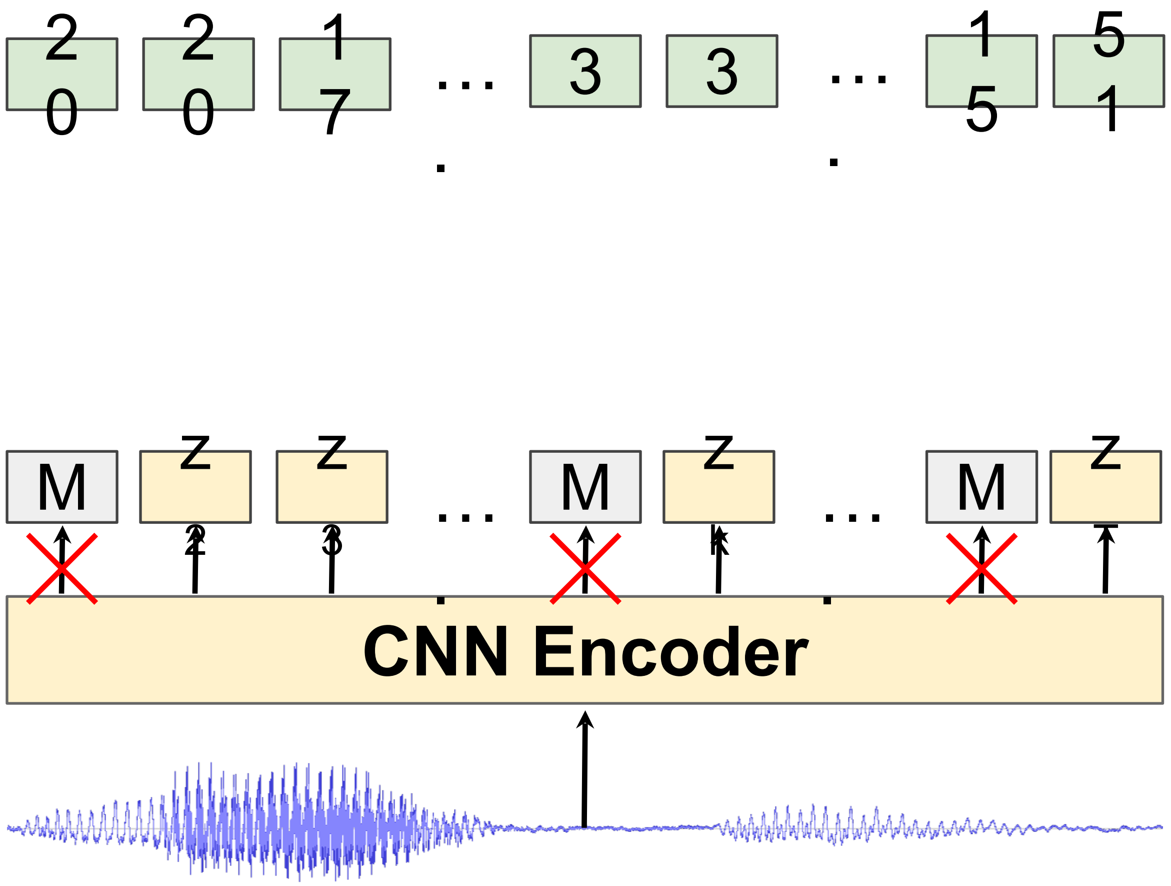
Hsu et al 2021, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units"

HuBERT: The pretext task



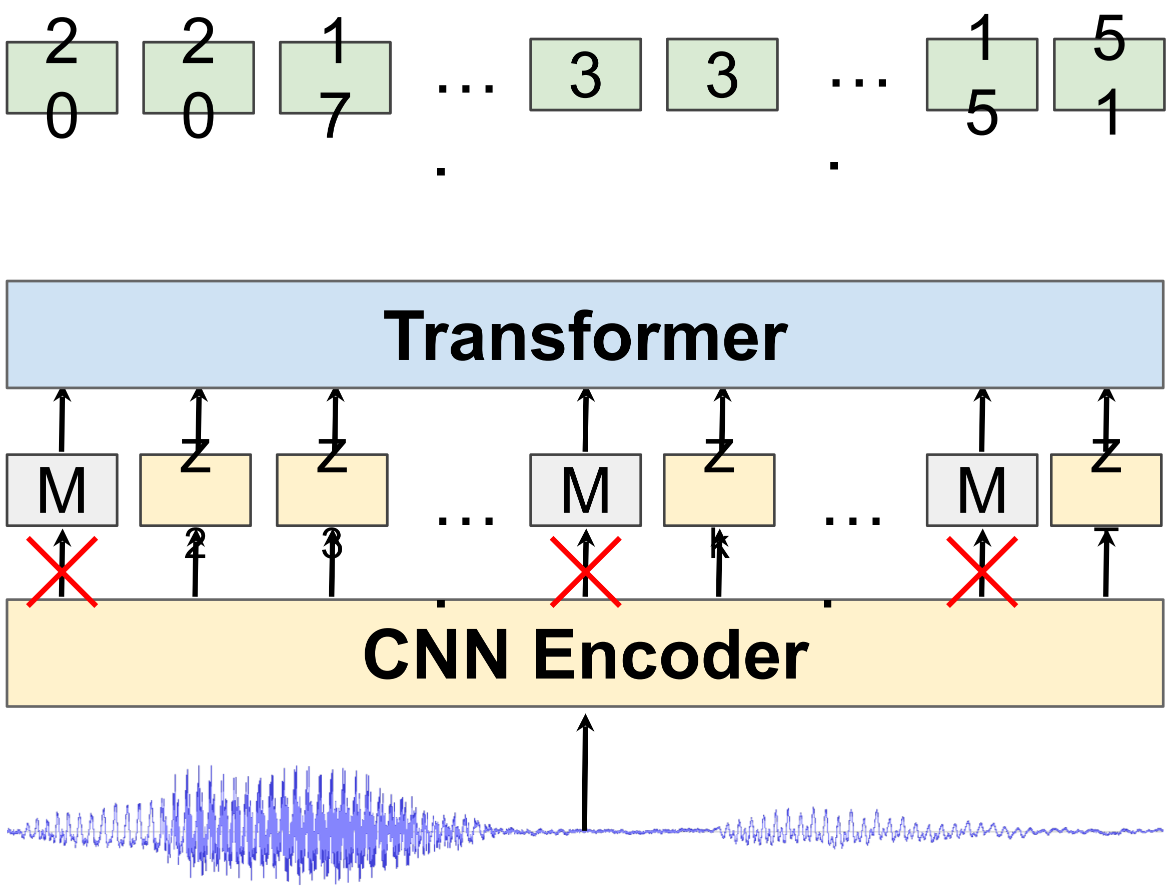
Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: The pretext task



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

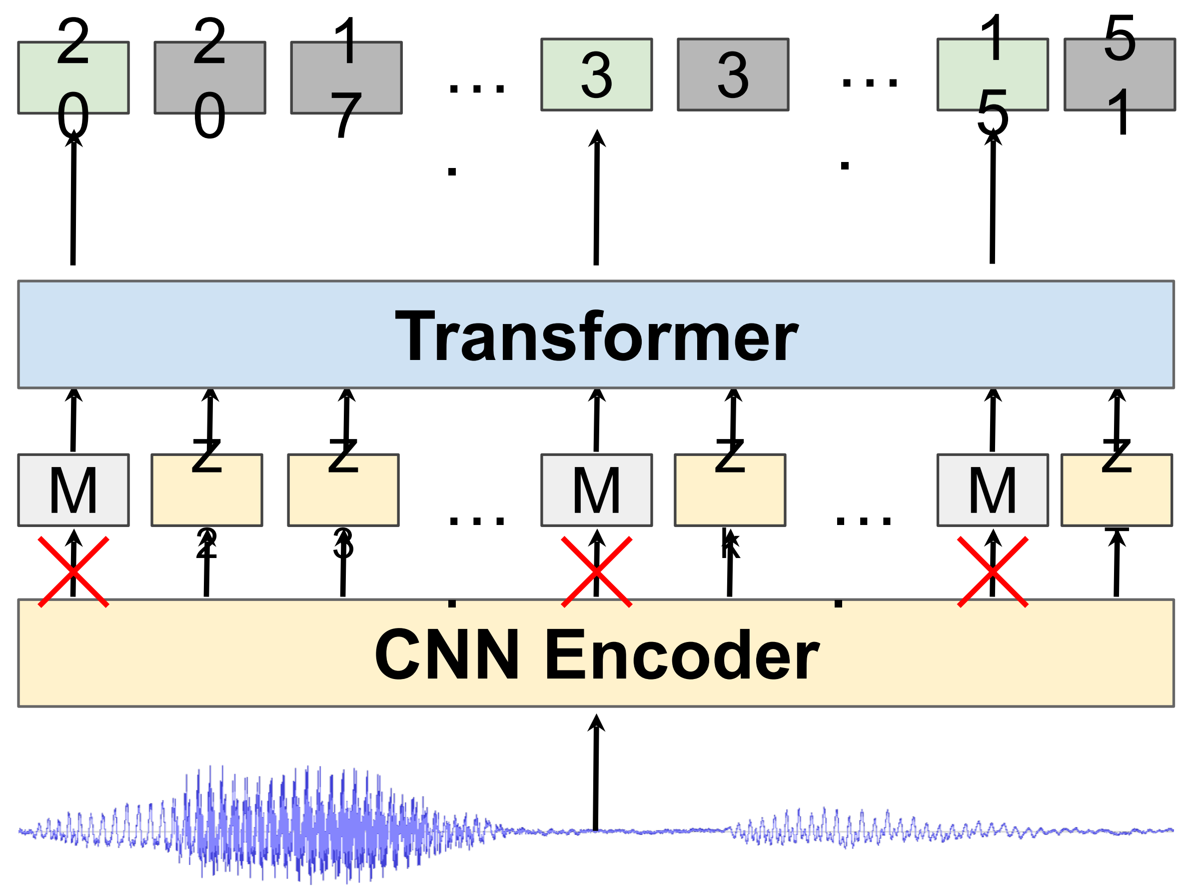
HuBERT: The pretext task



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: The pretext task

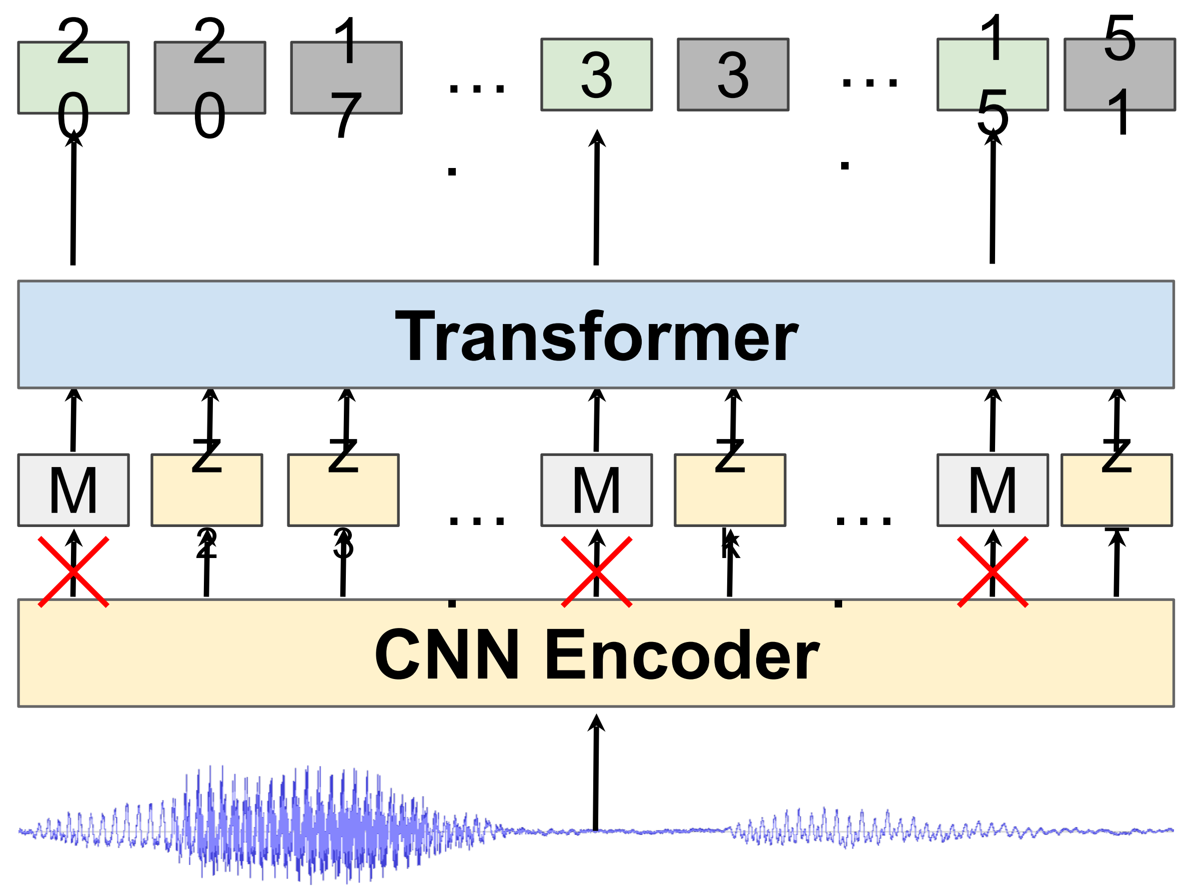
- Although the frame labels are imperfect, their consistency is more important!



Hsu et al 2021, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units"

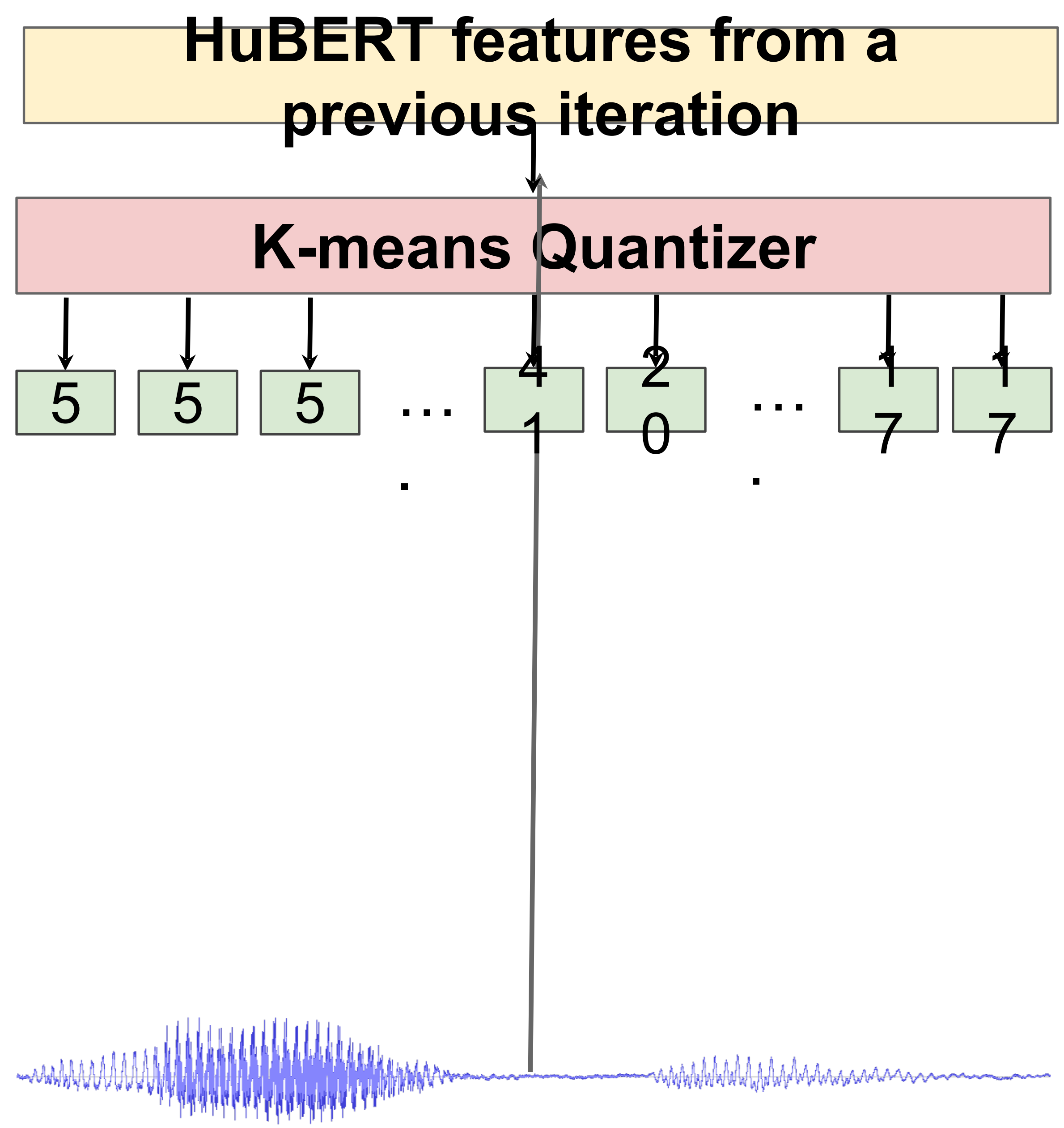
- Although the frame labels are imperfect, their consistency is more important!
- Th
  us
  prediction:

$$\mathcal{L}_m = \sum_{t \in M} -\log p(y_t \mid X)$$



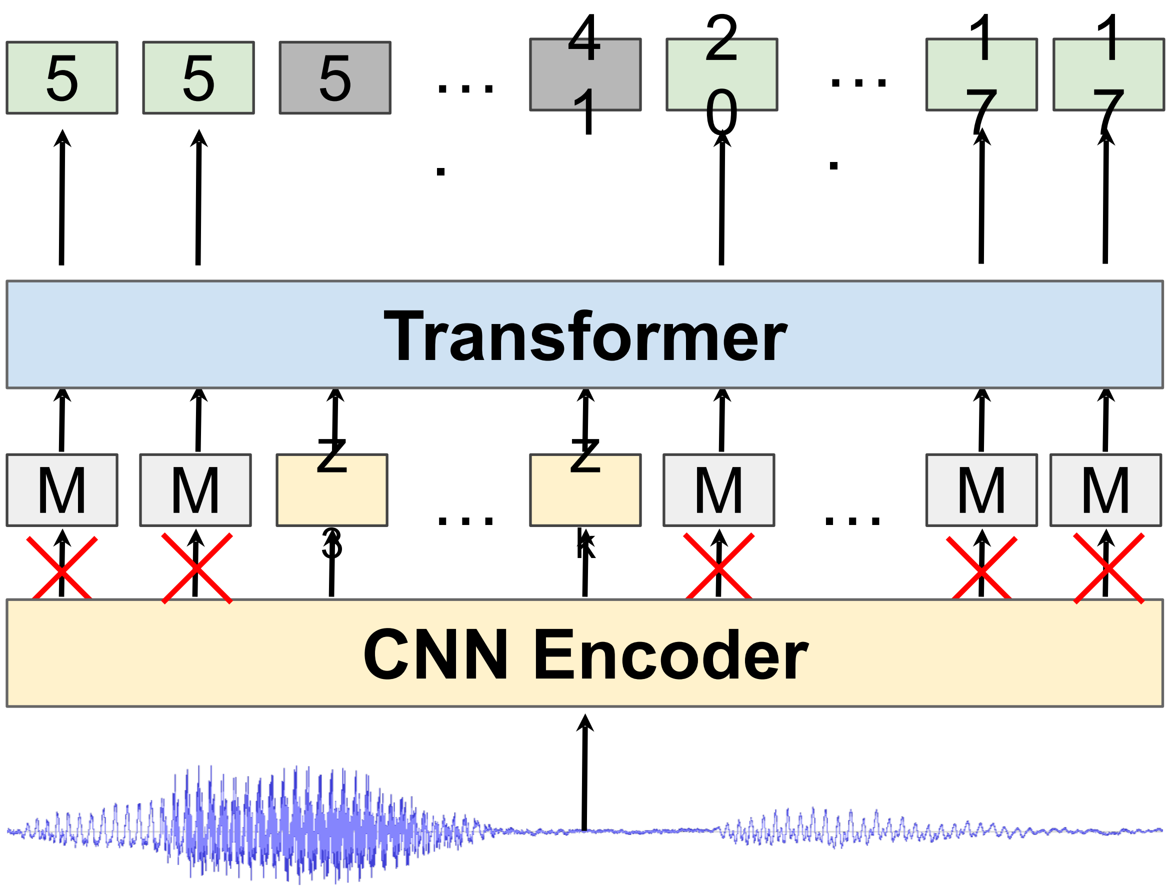
Hsu et al 2021, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units"

- Then the process can be repeated using learned HuBERT features from a previous iteration.



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

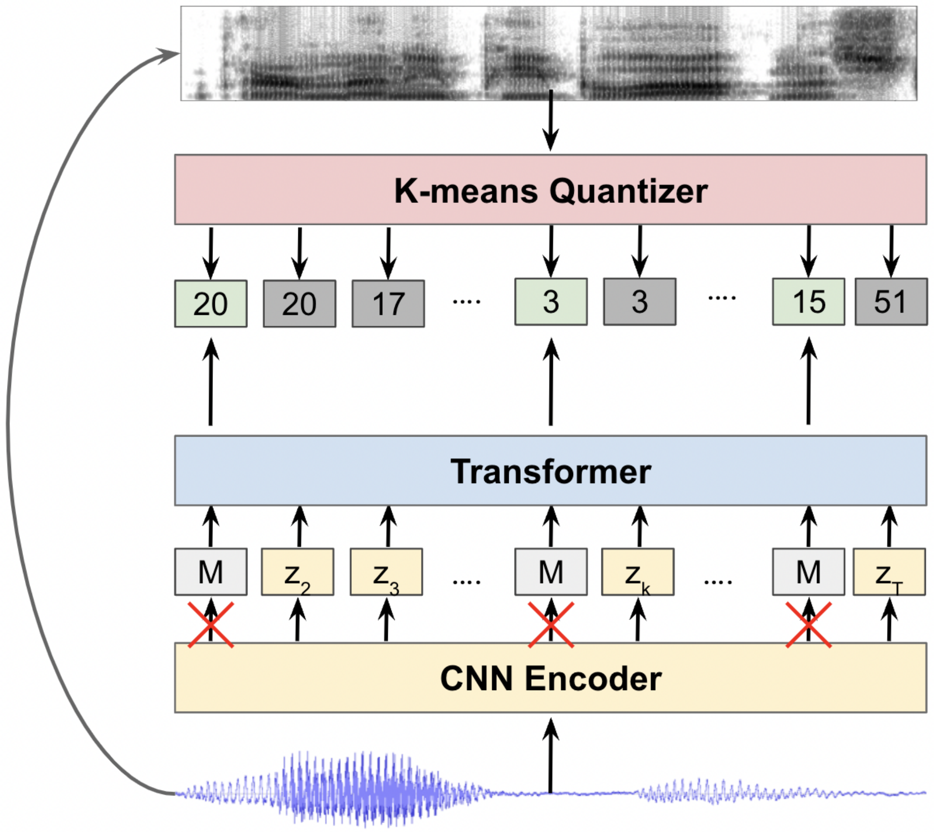
- Then the process can be repeated using learned HuBERT features from a previous iteration.



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

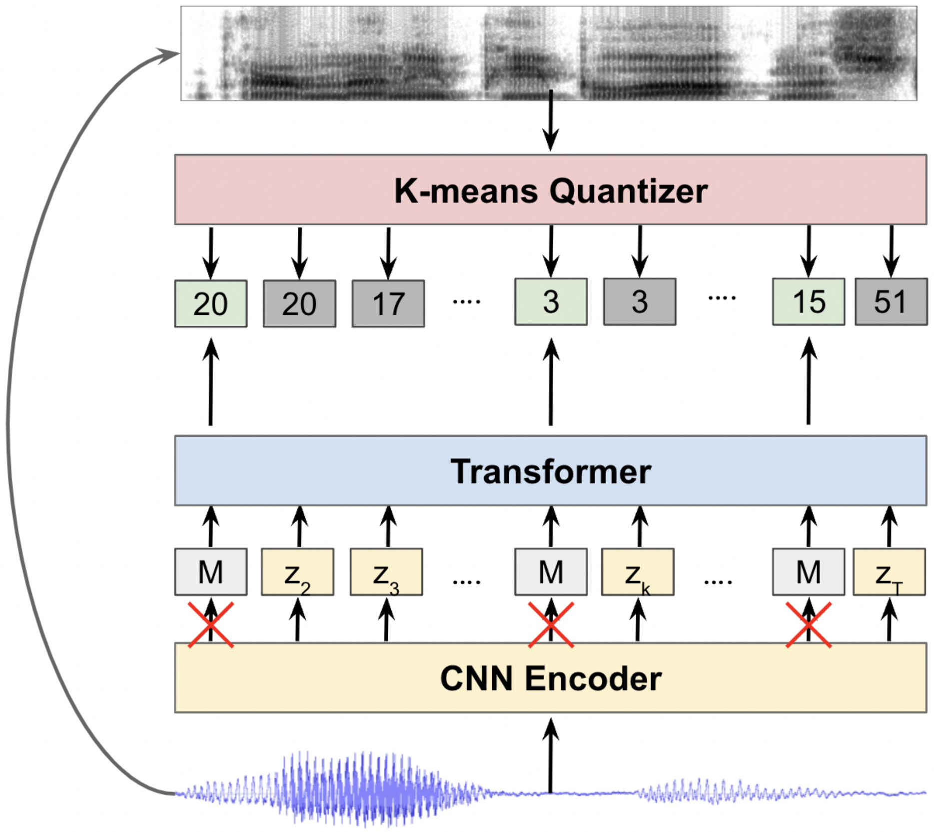
HuBERT: Implementation details

- A small codebook size, e.g., 50, 100, is used for the initial training iteration to focus on phonetic differences rather than speaker and style.



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: Implementation details



- A small codebook size, e.g., 50, 100, is used for the initial training iteration to focus on phonetic differences rather than speaker and style.
- For the subsequent two iterations, layers 6 and 9 of the base architecture (12 layers) are used for the clustering steps. They found empirically to contain higher quality features over many speech tasks.

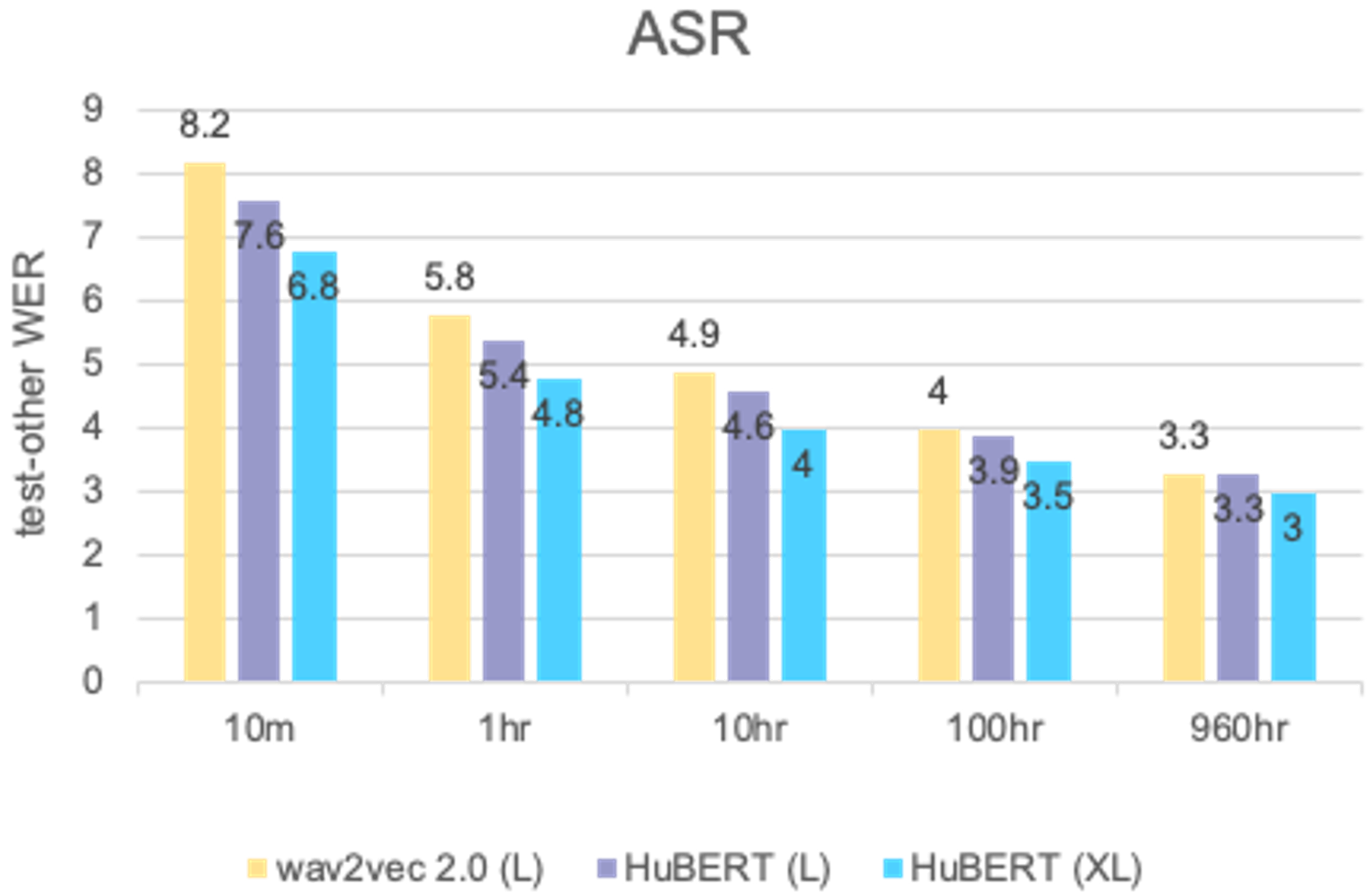
Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: Results

- Matched or beat the SOTA on ASR.

Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

- Matched or beat the SOTA on ASR.



Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

HuBERT: Results

- Matched or beat the SOTA on ASR.
- The best representations for multiple downstream tasks.

	PR	KS	IC	SID	ER	ASR (WER)		QbE	SF		ASV	SD
	PER ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	w/o ↓	w/ LM ↓	MTWV ↑	F1 ↑	CER ↓	EER ↓	DER ↓
FBANK	82.01	8.63	9.10	8.5E-4	35.39	23.18	15.21	0.0058	69.64	52.94	9.56	10.05
PASE+ [16]	58.95	82.54	29.82	37.99	57.86	24.92	16.61	0.0072	62.14	60.17	11.61	8.68
APC [7]	42.21	91.01	74.69	60.42	59.33	21.61	15.09	0.0310	70.46	50.89	8.56	10.53
VQ-APC [32]	41.49	91.11	74.48	60.15	59.66	21.72	15.37	0.0251	68.53	52.91	8.72	10.45
NPC [33]	43.69	88.96	69.44	55.92	59.08	20.94	14.69	0.0246	72.79	48.44	9.4	9.34
Mockingjay [8]	70.84	83.67	34.33	32.29	50.28	23.72	15.94	6.6E-04	61.59	58.89	11.66	10.54
TERA [9]	49.17	89.48	57.90	57.57	56.27	18.45	12.44	0.0013	67.50	54.17	15.89	9.96
modified CPC [34]	42.54	91.88	64.09	39.63	60.96	20.02	13.57	0.0326	71.19	49.91	12.86	10.38
wav2vec [12]	32.24	95.59	84.92	56.56	59.79	16.40	11.30	0.0485	76.37	43.71	7.99	9.9
vq-wav2vec [13]	34.24	93.38	85.68	38.80	58.24	18.70	12.69	0.0410	77.68	41.54	10.38	9.93
wav2vec 2.0 Base [14]	5.56	96.23	92.35	75.18	63.43	9.57	6.32	0.0233	88.30	24.77	6.02	6.08
wav2vec 2.0 Large [14]	4.75	**96.66**	95.28	86.14	65.64	3.75	3.10	0.0489	86.94	27.80	5.65	**5.62**
HuBERT Base [35]	5.05	96.30	98.34	81.42	64.92	6.74	4.93	**0.0736**	88.53	25.20	**5.11**	5.88
HuBERT Large [35]	**3.28**	95.29	**98.76**	**90.33**	**67.62**	**3.67**	**2.91**	0.0353	**89.81**	**21.76**	5.98	5.75

Hsu et al 2021, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”

- HuBERT inspired many follow up work:

Chen et. al., 2021 "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing"
Chiu et. al. 2022 "Self-supervised Learning with Random-projection Quantizer for Speech Recognition"
Shi et. al. 2022 "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction"
Lakhotia et. al., 2021 "Generative Spoken Language Modeling from Raw Audio"

- HuBERT inspired many follow up work:
  - Combined masked prediction and denoising pre-training.

Chen et. al., 2021 "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing"
Chiu et. al. 2022 "Self-supervised Learning with Random-projection Quantizer for Speech Recognition"
Shi et. al. 2022 "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction"
Lakhotia et. al., 2021 "Generative Spoken Language Modeling from Raw Audio"

- HuBERT inspired many follow up work:
  - Combined masked prediction and denoising pre-training.
  - Random clustering is as effective in learning representations as k-means.

Chen et. al., 2021 "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing"
Chiu et. al. 2022 "Self-supervised Learning with Random-projection Quantizer for Speech Recognition"
Shi et. al. 2022 "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction"
Lakhotia et. al., 2021 "Generative Spoken Language Modeling from Raw Audio"

- HuBERT inspired many follow up work:
    - Combined masked prediction and denoising pre-training.
    - Random clustering is as effective in learning representations as k-means.
    - Multimodal clustering for audio-visual ASR.

Chen et. al., 2021 "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing"
Chiu et. al. 2022 "Self-supervised Learning with Random-projection Quantizer for Speech Recognition"
Shi et. al. 2022 "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction"
Lakhotia et. al., 2021 "Generative Spoken Language Modeling from Raw Audio"

- HuBERT inspired many follow up work:
  - Combined masked prediction and denoising pre-training.
  - Random clustering is as effective in learning representations as k-means.
  - Multimodal clustering for audio-visual ASR.
  - High-quality discrete units facilitated textless NLP research for speech generation.
  - ….

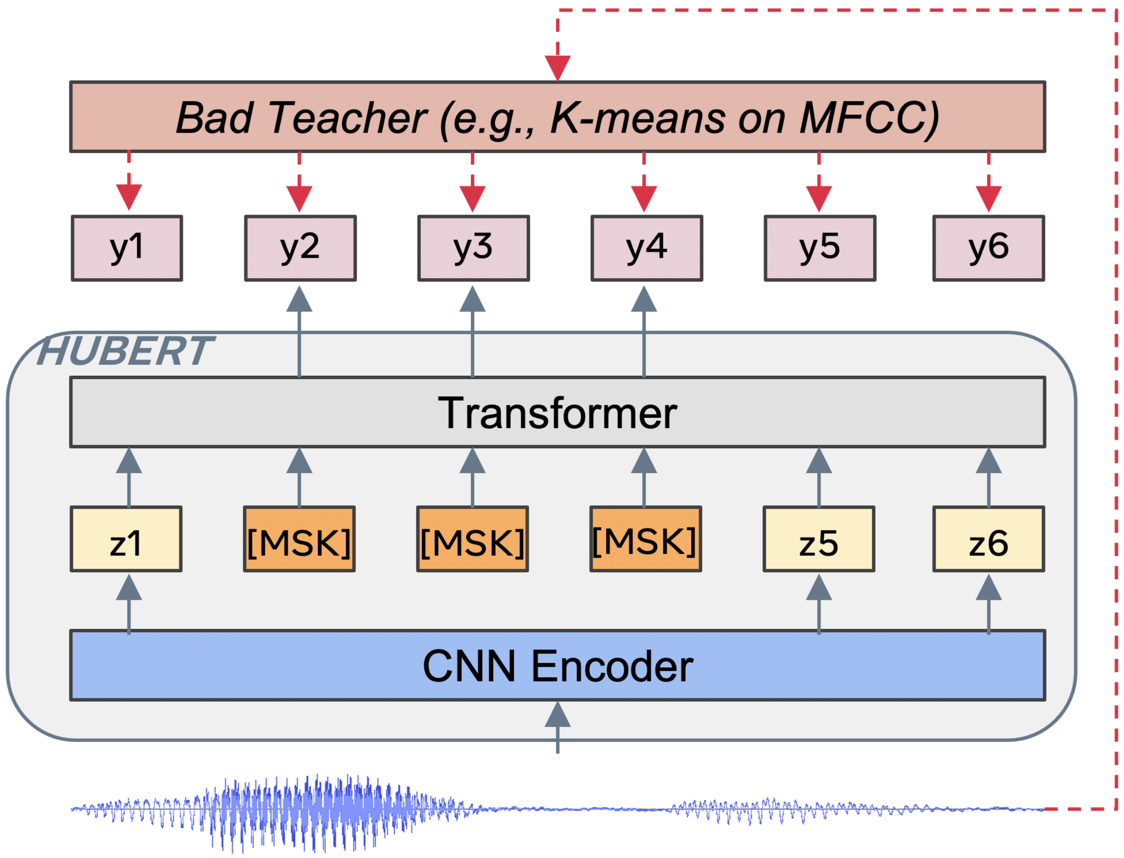
Chen et. al., 2021 "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing"
Chiu et. al. 2022 "Self-supervised Learning with Random-projection Quantizer for Speech Recognition"
Shi et. al. 2022 "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction"
Lakhotia et. al., 2021 "Generative Spoken Language Modeling from Raw Audio"

HuBERT: Details:

- Small codebook sizes, e.g. 100, 500.
- The loss is only applied over masked regions.

$$L_m(\theta; X, M, Y) = \sum_{t \in M} \log p(y_t \mid \tilde{X}, t)$$

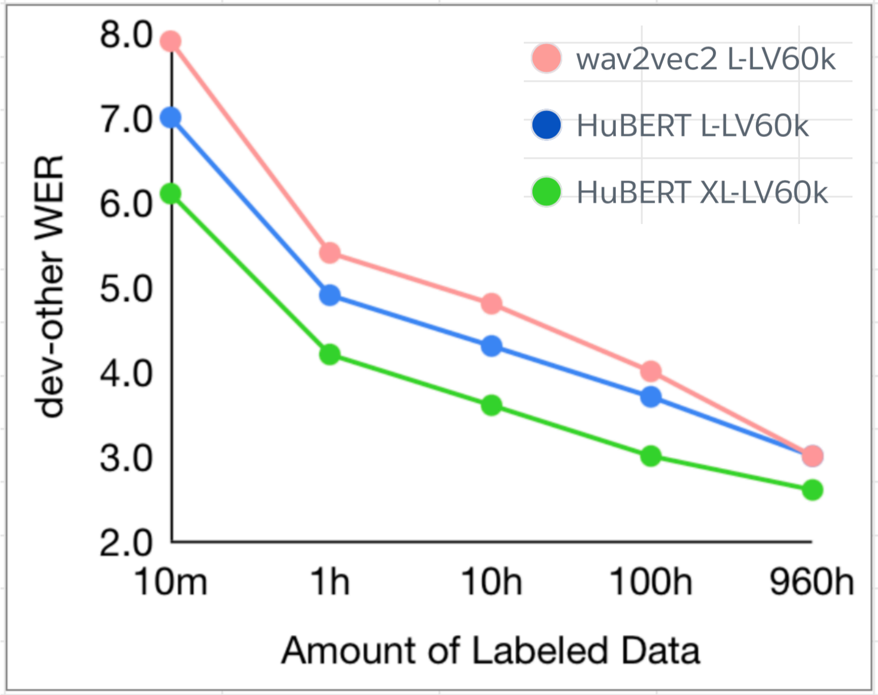
- The learned latent features can be quantized for another learning iteration.
- GMMs or HMMs may replace k-means for better initial labels.

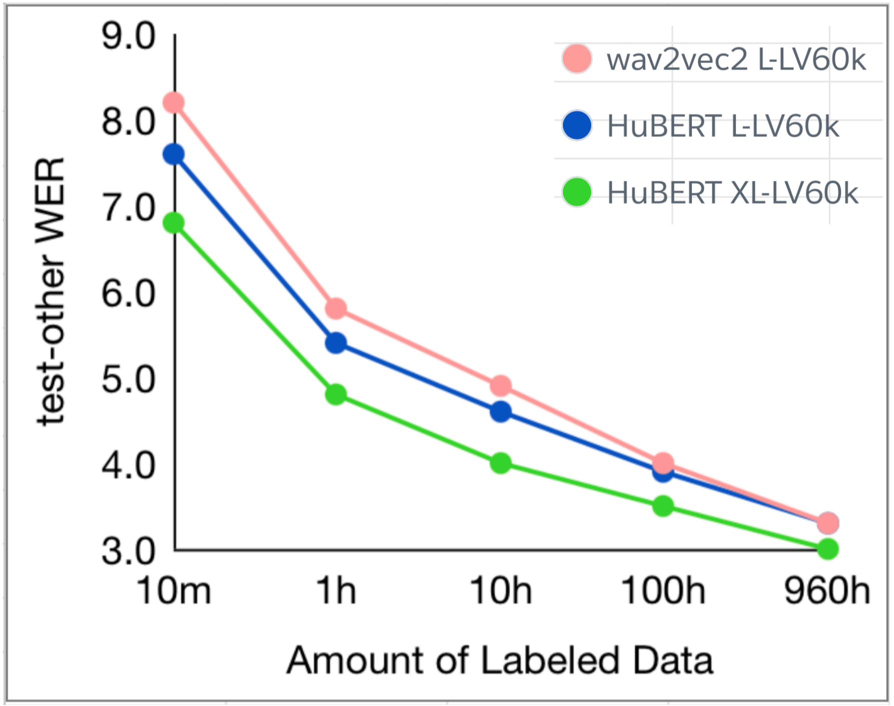


Hsu et al 2021, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units"

Results on Libri-light:

- Using at most three clustering steps, HuBERT is as effective or better than Wav2Vec 2.0
- Using a 1B model improves the performance across all sizes of labeled data for the challenging dev/test_other condition (up to 19% and13%).





Results on SUPERB – last layer

	PR	KS	IC	SID	ER	ASR (WER)		QbE	SF		SV	SD
	PER ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	w/o ↓	w/ LM ↓	MTWV ↑	F1 ↑	CER ↓	EER ↓	DER ↓
FBANK	82.01	8.63	9.10	8.5E-4	35.39	23.18	15.21	0.0058	69.64	52.94	9.56	10.05
PASE+ [16]	58.88	82.37	30.29	35.84	57.64	24.92	16.61	7.0E-4	60.41	62.77	10.91	8.52
APC [7]	41.85	91.04	74.64	59.79	58.84	21.61	15.09	0.0268	71.26	50.76	8.81	10.72
VQ-APC [32]	42.86	90.52	70.52	49.57	58.31	21.72	15.37	0.0205	69.62	52.21	9.29	10.49
NPC [33]	52.67	88.54	64.04	50.77	59.55	20.94	14.69	0.0220	67.43	54.63	10.28	9.59
Mockingjay [8]	80.01	82.67	28.87	34.50	45.72	23.72	15.94	3.1E-10	60.83	61.15	23.22	11.24
TERA [9]	47.53	88.09	48.8	58.67	54.76	18.45	12.44	8.7E-5	63.28	57.91	16.49	9.54
modified CPC [34]	41.66	92.02	65.01	42.29	59.28	20.02	13.57	0.0061	74.18	46.66	9.67	11.00
wav2vec [12]	32.39	94.09	78.91	44.88	58.17	16.40	11.30	0.0307	77.52	41.75	9.83	10.79
vq-wav2vec [13]	53.49	92.28	59.4	39.04	55.89	18.70	12.69	0.0302	70.57	50.16	9.50	9.93
wav2vec 2.0 Base [14]	28.37	92.31	58.34	45.62	56.93	9.57	6.32	8.8E-4	79.94	37.81	9.69	7.48
HuBERT Base [35]	6.85	**95.98**	95.94	64.84	62.94	6.74	4.93	**0.0759**	86.24	28.52	**7.22**	6.76
HuBERT Large [35]	**3.72**	93.15	**98.37**	**66.40**	**64.93**	**3.67**	**2.91**	0.0360	**88.68**	**23.05**	7.70	**6.23**

Results on SUPERB – Weighted sum of layers

	PR	KS	IC	SID	ER	ASR (WER)		QbE	SF		ASV	SD
	PER ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	w/o ↓	w/ LM ↓	MTWV ↑	F1 ↑	CER ↓	EER ↓	DER ↓
FBANK	82.01	8.63	9.10	8.5E-4	35.39	23.18	15.21	0.0058	69.64	52.94	9.56	10.05
PASE+ [16]	58.95	82.54	29.82	37.99	57.86	24.92	16.61	0.0072	62.14	60.17	11.61	8.68
APC [7]	42.21	91.01	74.69	60.42	59.33	21.61	15.09	0.0310	70.46	50.89	8.56	10.53
VQ-APC [32]	41.49	91.11	74.48	60.15	59.66	21.72	15.37	0.0251	68.53	52.91	8.72	10.45
NPC [33]	43.69	88.96	69.44	55.92	59.08	20.94	14.69	0.0246	72.79	48.44	9.4	9.34
Mockingjay [8]	70.84	83.67	34.33	32.29	50.28	23.72	15.94	6.6E-04	61.59	58.89	11.66	10.54
TERA [9]	49.17	89.48	57.90	57.57	56.27	18.45	12.44	0.0013	67.50	54.17	15.89	9.96
modified CPC [34]	42.54	91.88	64.09	39.63	60.96	20.02	13.57	0.0326	71.19	49.91	12.86	10.38
wav2vec [12]	32.24	95.59	84.92	56.56	59.79	16.40	11.30	0.0485	76.37	43.71	7.99	9.9
vq-wav2vec [13]	34.24	93.38	85.68	38.80	58.24	18.70	12.69	0.0410	77.68	41.54	10.38	9.93
wav2vec 2.0 Base [14]	5.56	96.23	92.35	75.18	63.43	9.57	6.32	0.0233	88.30	24.77	6.02	6.08
wav2vec 2.0 Large [14]	4.75	**96.66**	95.28	86.14	65.64	3.75	3.10	0.0489	86.94	27.80	5.65	**5.62**
HuBERT Base [35]	5.05	96.30	98.34	81.42	64.92	6.74	4.93	**0.0736**	88.53	25.20	**5.11**	5.88
HuBERT Large [35]	**3.28**	95.29	**98.76**	**90.33**	**67.62**	**3.67**	**2.91**	0.0353	**89.81**	**21.76**	5.98	5.75