

Speech Recognition

Speech Recognition Introduction

Multimodal LLMs: speech

LLMs can now deal with text, images, video and **speech**!

 **OpenAI Whisper**

 **nvidia / canary-1b-v2**
 **Gemini 2.5 Pro**

Microsoft[®] **PHI-4**

 **Meta**
wav2vec 2.0

Common Speech tasks

Automatic Speech Recognition (ASR): Speech in, text out

Text-to-Speech (TTS/Speech Synthesis): Text in, speech out

Voice Morphing: Speech in, speech out

Automatic Speech Translation: Speech in, speech out

Language ID: Speech in, language name out

Speaker ID: Speech in, speaker name out

Diarization: Speech in, a script (who talked when) out

Voice Activity Detection: audio in, output: speech regions

Automatic Speech Recognition

The task: Map from a wavefile to a text string.

How do we do it: Transformers! Plus encoder-decoder!

Conversational speech is harder than you realize



A piece of an utterance
without context



A piece of an utterance
with context

Why are accents hard?



A word by itself



The word in context

A metric for errors: Word Error Rate (WER)

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

= (length-normalized) Minimum Edit Distance

Reference: Um I left this phone upstairs last night

Hypothesis: My sister her phone it stores less nine

We'll need a metric for errors:

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

Use MinEditDist to compute best alignment of reference and hypothesis, then count errors:

Reference:	Um	I	left	this	---	phone	--	upstairs	last	night
Hypothesis:	My	--	---	sister	her	phone	it	stores	less	nine
Score	S		D D	S	I		I	S	S	S

$$\text{WER} = 100 \times 9/8 = 112.5$$

We use WER despite many problems

Isn't semantically aware

- Doesn't distinguish between function words (of, they, he, she) and more important content words

Not necessarily comparable across languages (assumes a shared notion of words)

Requires really careful text normalization details

- Spelling conventions must be identical
- Apostrophes for contractions like "can't",
- Lots of non-standard words uh-huh/m-hmm/

Librispeech

Read speech from public domain audio books LibriVox

Latest version (Multilingual Librispeech) has

- 44,000 hours of English
- 6,000 hours of German, Dutch, French, Spanish, Italian, Portuguese, and Polish.



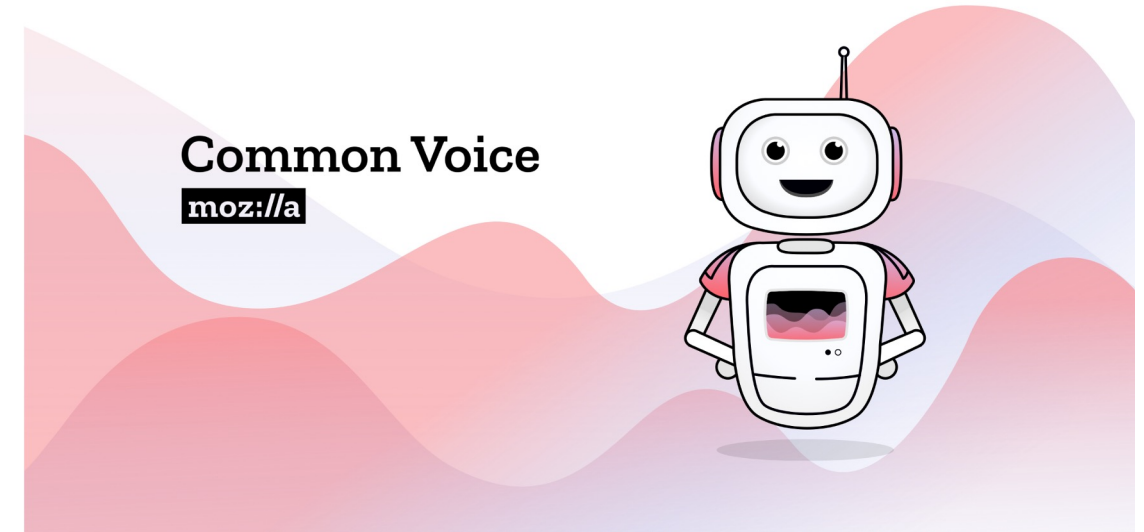
Common Voice

Multilingual crowdsourced dataset

33,000 hours in 133 languages

Still read speech, but anyone can record utterances

Noisier than LibriSpeech



終有一日，所有嘢 都會變

FLEURS: Parallel speech and text in 100+ languages

Speech version of FloRes-101 benchmark

2009 sentences extracted from English Wikipedia and translated into 101 other languages, with 3 read recordings of each sentence

12 hours of speech per language.

CHiME

Audio in difficult situations

CHiME 6: ASR of
conversational speech in
real home environments



Modern word error rates

English Tasks	WER%
LibriSpeech audiobooks 960hour clean	1.4
LibriSpeech audiobooks 960hour other	2.6
Switchboard telephone conversations between strangers	5.8
CALLHOME telephone conversations between family	11
AMI meetings	11
Sociolinguistic interviews, CORAAL (AAE)	16.2
CHiME6 dinner parties with distant microphones	25.5
Sample tasks in other languages	WER%
Common Voice 15 Vietnamese	39.8
Common Voice 15 Swahili	51.2
FLEURS Bengali	50
Chinese (Mandarin) Tasks	CER%
AISHELL-1 Mandarin read speech corpus	3.9
HKUST Mandarin Chinese telephone conversations	18.5

Automatic Speech Recognition

As we'll see, ASR still fails for many tasks and domains

- Noisy speech
- Multiple overlapping speakers
- Low-resource languages
- Some speakers (children, old people, people who stutter)

Can't fix what we don't understand!

- Understanding errors can help inform how to fix them

We should build speech technology that works well for everyone!

Some basic ethics when working on speech technologies



**Don't record
someone without
their consent**

In California, all parties to any phone conversation (or any live confidential conversation) must give their consent to be recorded.



**Don't use
someone's speech
data without
consent. Especially
with synthesis/
voice cloning**

It might be fun, but it's creepy.
People get upset.
Use existing speech datasets



**Do consider
subgroup and
language bias in
systems**

Poor performance on subgroups e.g. non-native speakers. Many languages are under-served relative to English/Mandarin.

Speech Recognition

Speech Recognition Introduction

Speech Recognition

Phonetics: Transcription

Why phonetics?

Understanding phonetics and speech production helps with:

- Interpretability
 - Gives us a language to describe system internal representations
- Debugging.
 - If you can't describe the error you can't measure it.
- Generality
 - For multilinguality, helps to know the space of possible speech sounds.

ARPAbet

An alphabet for transcribing American English phones
Developed for early English-focused ASR research

that	dh ae t
Thad	th ae d
bead	b iy d
clack	k l ae k
clock	k l aa k
supper	s ah p axr
super	s uw p axr

ARPAbet

A simple subset of the ARPAbet

<http://web.stanford.edu/class/cs124/arpabetsmall.html>

The CMU Pronouncing Dictionary

<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

Practice

stage: s t ey d zh

hacky: h ae k iy

Jackie:

haste:

plaid

Practice

stage: s t ey d zh

hacky: h ae k iy

Jackie: j ae k iy

haste: h ey s t

played : p l ey d

English Vowels in ARPAbet

	b_d	ARPA		b_d	ARPA
1	bead	iy	9	bode	ow
2	bid	ih	10	booed	uw
3	bayed	ey	11	bud	ah
4	bed	eh	12	bird	er
5	bad	ae	13	bide	ay
6	body	aa	14	bowed	aw
7	(baud), board	ao	15	Boyd	oy
8	pudding	uh			

IPA – International Phonetic Alphabet

ARPAbet only has American English phones!

- Not used in multi-lingual systems

What we use instead for phonetic analysis

- International Phonetic Alphabet:
- http://en.wikipedia.org/wiki/International_Phonetic_Alphabet

CONSONANTS (PULMONIC)

© 2020 IPA

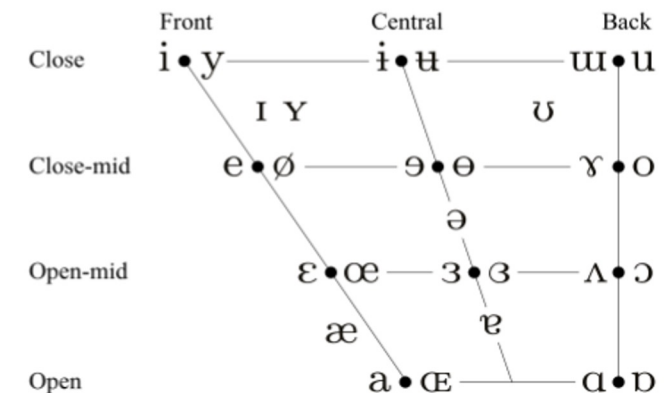
	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
◌ǀ Bilabial	◌ɓ Bilabial	◌' Examples:
◌ǃ Dental	◌ɗ Dental/alveolar	◌p' Bilabial
◌ǂ (Post)alveolar	◌ɟ Palatal	◌t' Dental/alveolar
◌ǁ Palatoalveolar	◌ɡ Velar	◌k' Velar
◌ǁ Alveolar lateral	◌ɠ Uvular	◌s' Alveolar fricative

VOWELS



Speech Recognition

Phonetics: Transcription

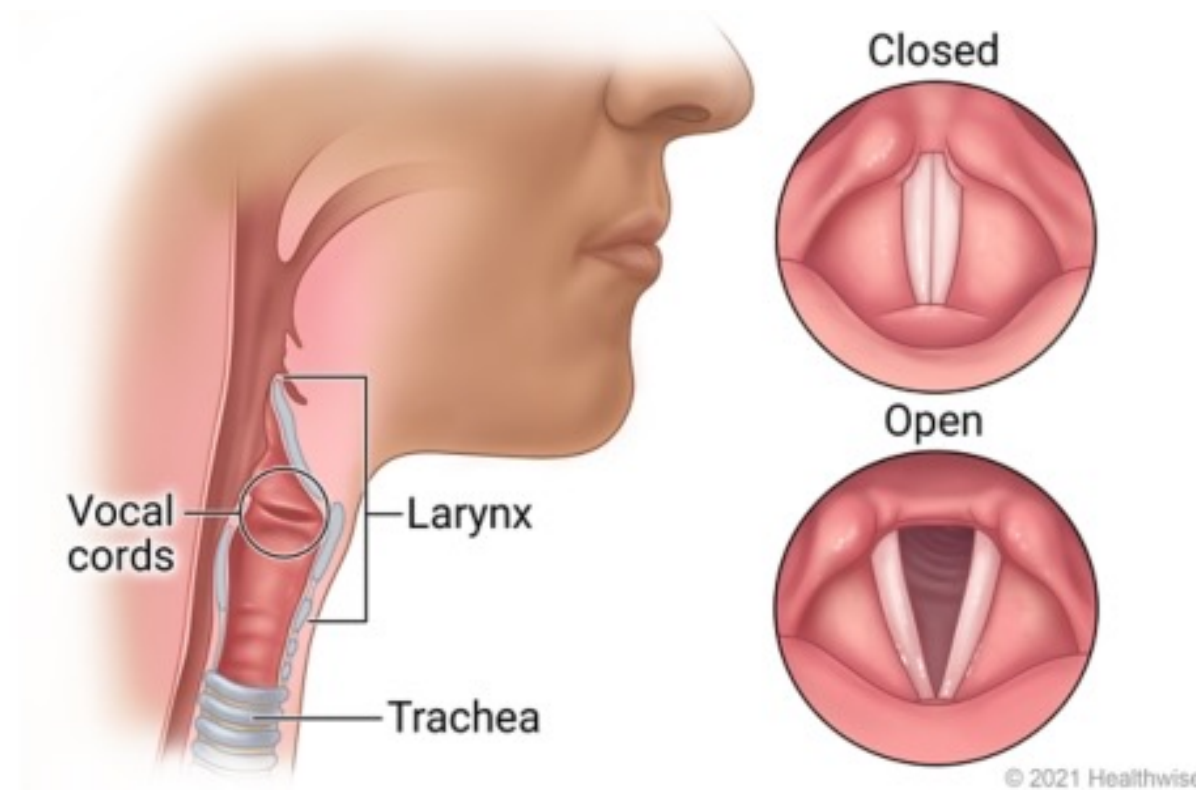
Speech Recognition

Phonetics: Articulatory Phonetics

How speech sounds are
made by articulators
(biology in motion)

How speech is made!

1. Air come up from from the **lungs**
2. Makes the **vocal cords** vibrate



3. Resulting pressure waves shaped by **tongue, mouth, lips**

Vocal Fold Vibration

- Air comes up from lungs
- Forces its way through vocal cords, pushing open (2,3,4)
- This causes air pressure in glottis to fall:
 - when gas runs through constricted passage, its velocity increases (Venturi tube effect)
 - this increase in velocity results in a drop in pressure (Bernoulli principle)
- Because of drop in pressure, vocal cords snap together again (6-10)
- Single cycle: $\sim 1/100$ of a second



Warning: gross video on next slide

Vocal Folds



Thanks to SPS, the Society of Physics students <https://www.youtube.com/watch?v=wYnPA7IXFIU>

Voicing

The vocal cords vibrate for some sounds

- All vowels
- "voiced" consonants like d/b/g/z/dh/m/n

But not for others

- t/p/k/s/th/f

Exercise:

- hold your hand on your larynx and say:
- ssssssss vs zzzzzzzzzzzzzzzzzzzzz

Sagittal View of Vocal Tract

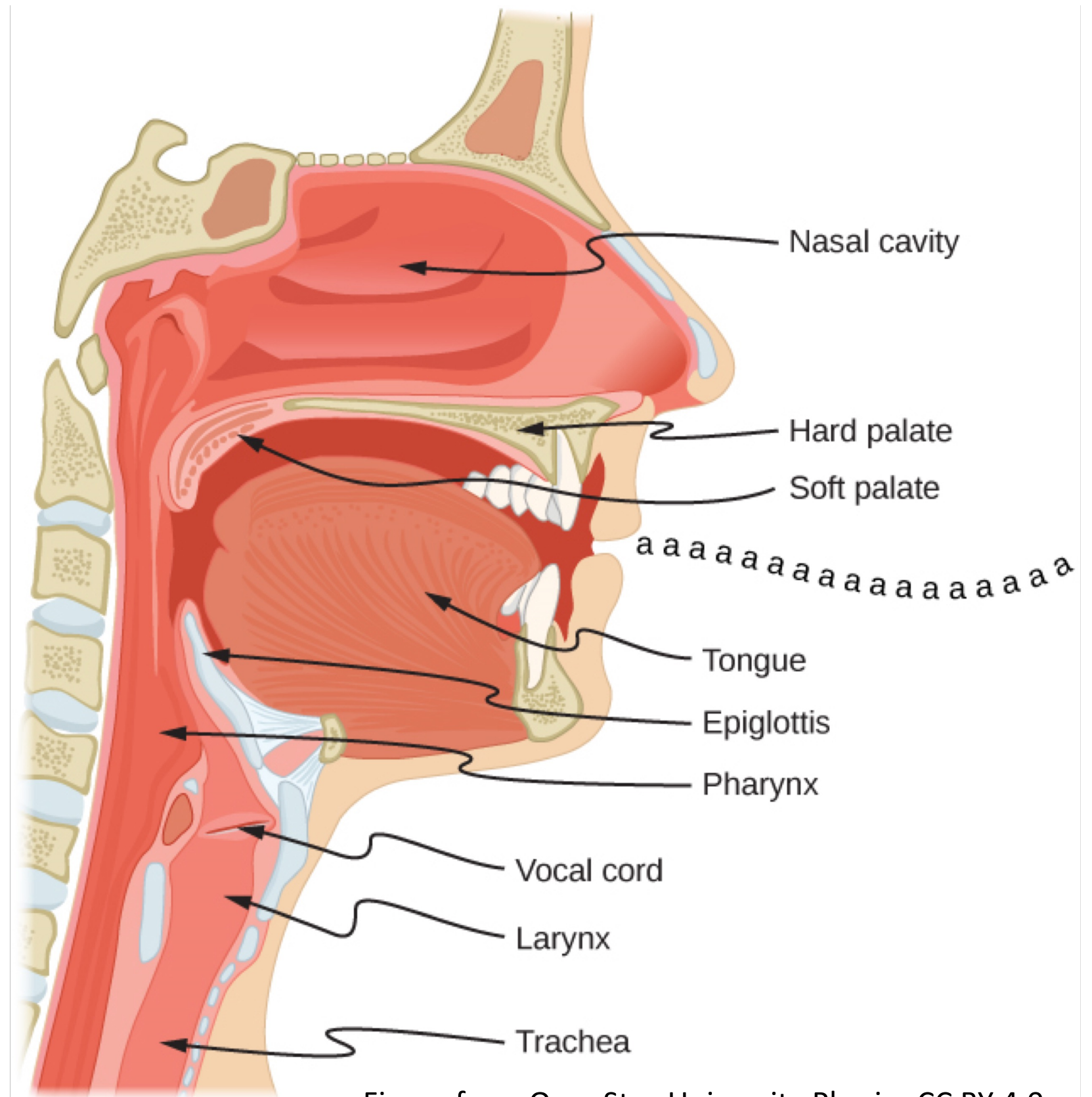


Figure from OpenStax University Physics CC BY 4.0

Consonants and Vowels



Consonants:

sounds with
audible noise produced by a
constriction



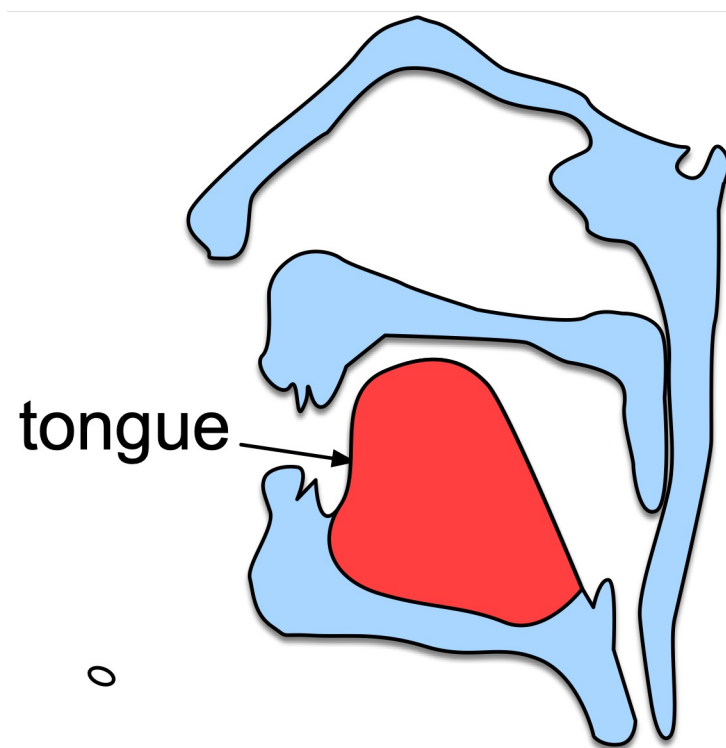
Vowels:

sounds without
audible noise produced by
a constriction

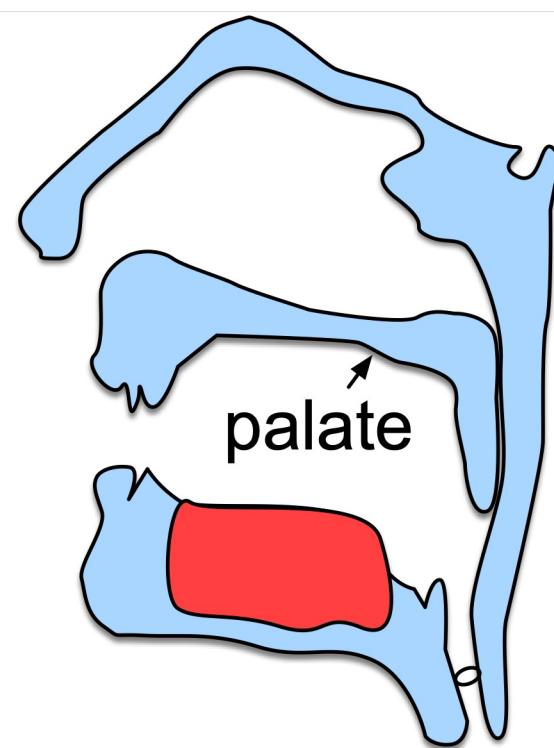
Vowels and the Tongue



USC SAIL



beet [iy]



bat [ae]



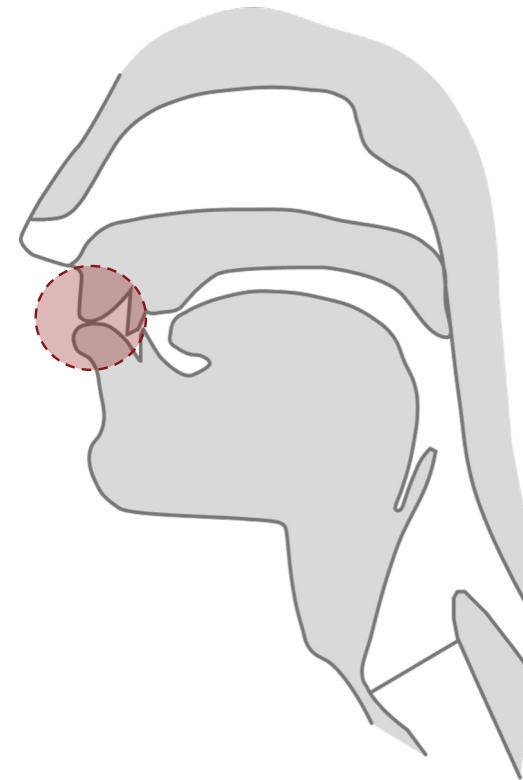
boot [uw]

Stop consonant example "p": passage is closed, then released

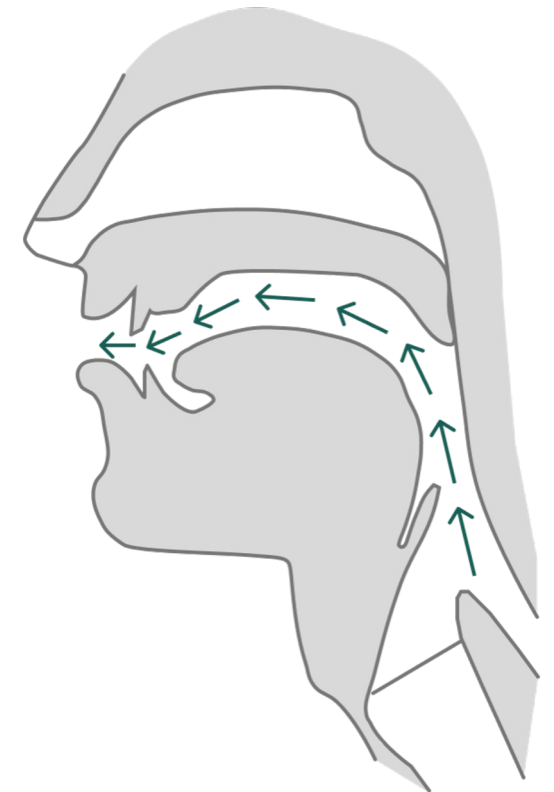
Closure: Lips close so no air escapes through mouth

Release: Air pressure builds up behind closure, explodes when released

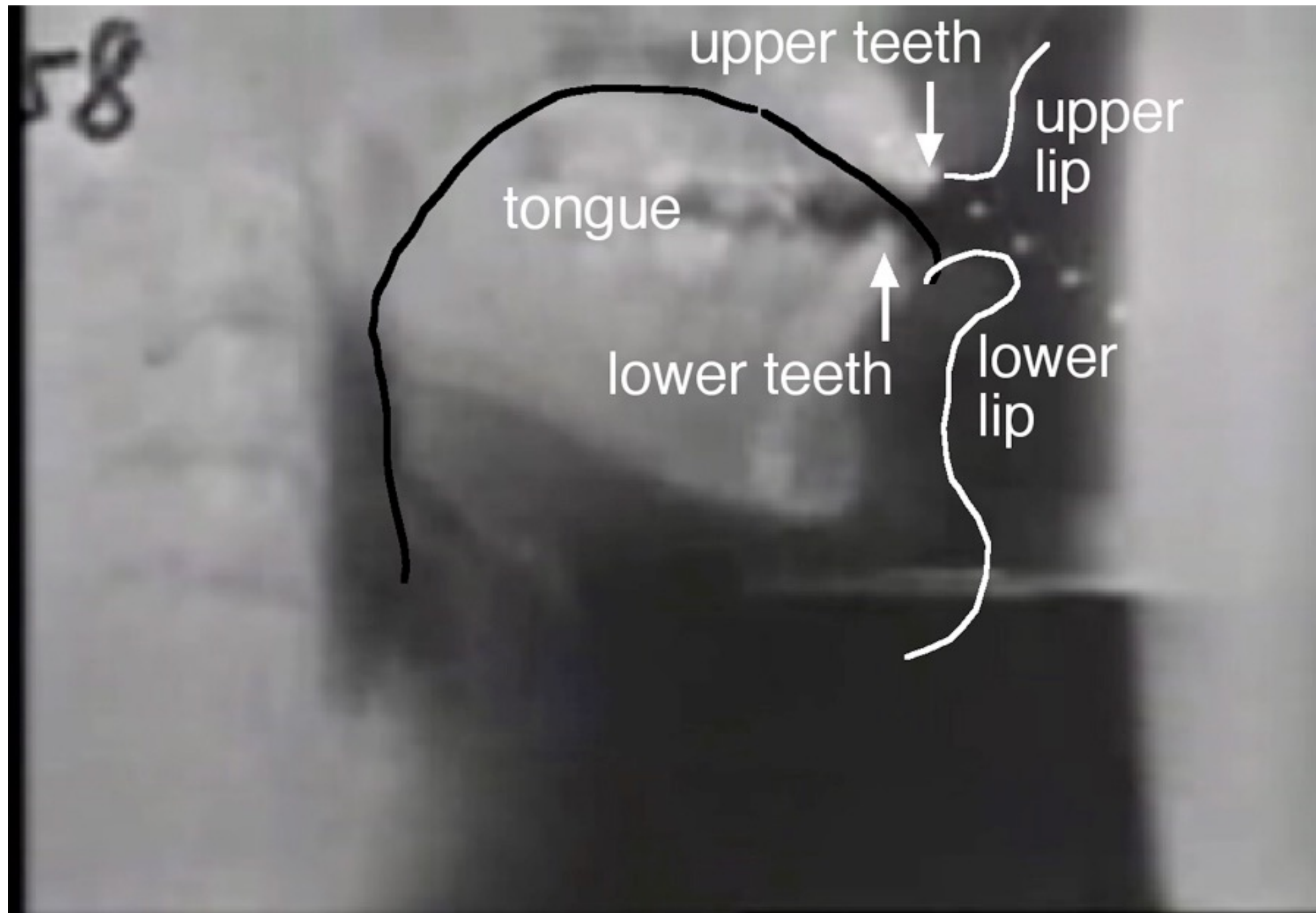
- p, t, k, b, d, g



p
closure



p
release



X-Ray of Ken Stevens, labels from Peter Ladefoged's web site

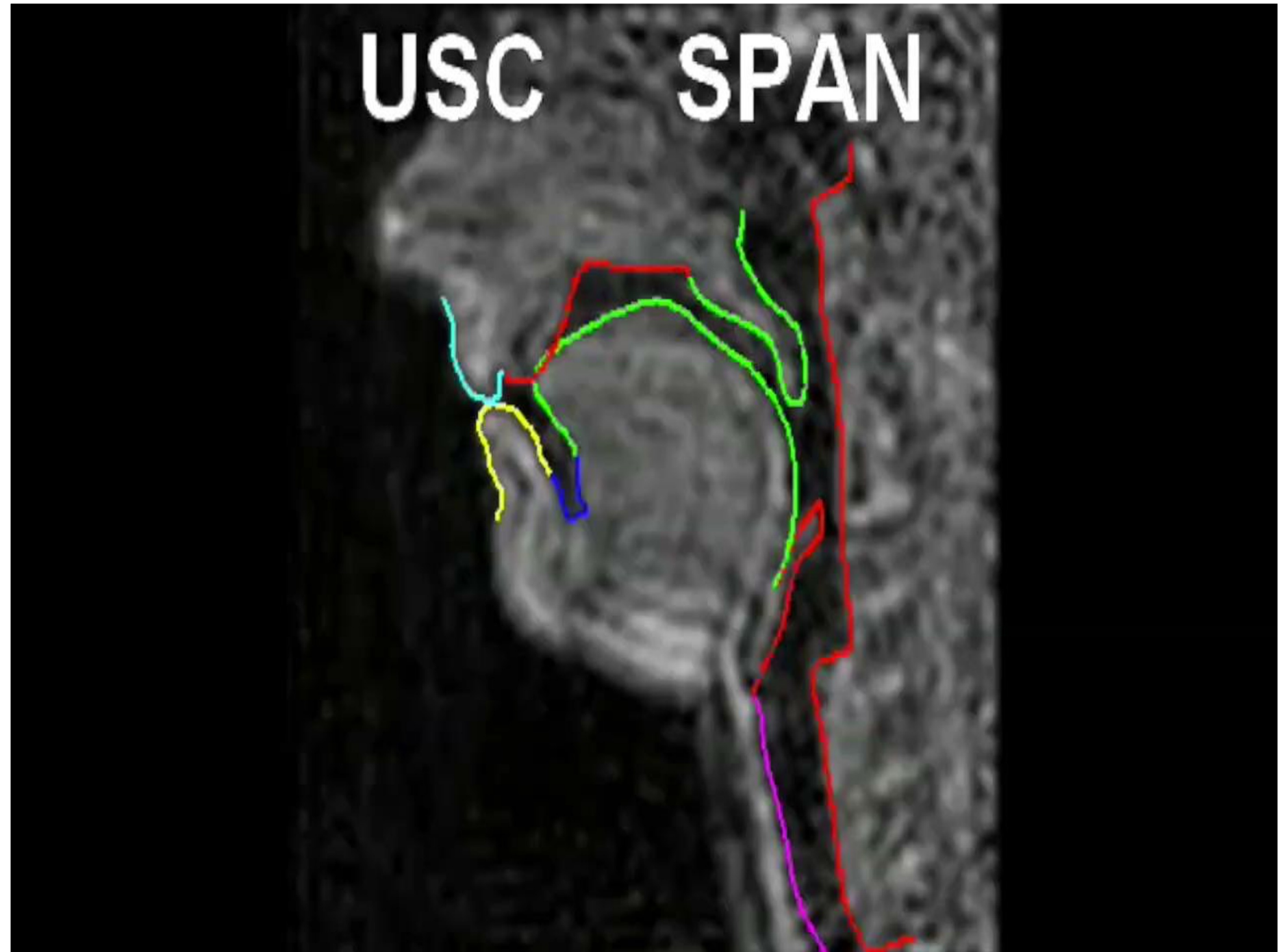
20th Century Vocal tract movie (high speed x-ray)



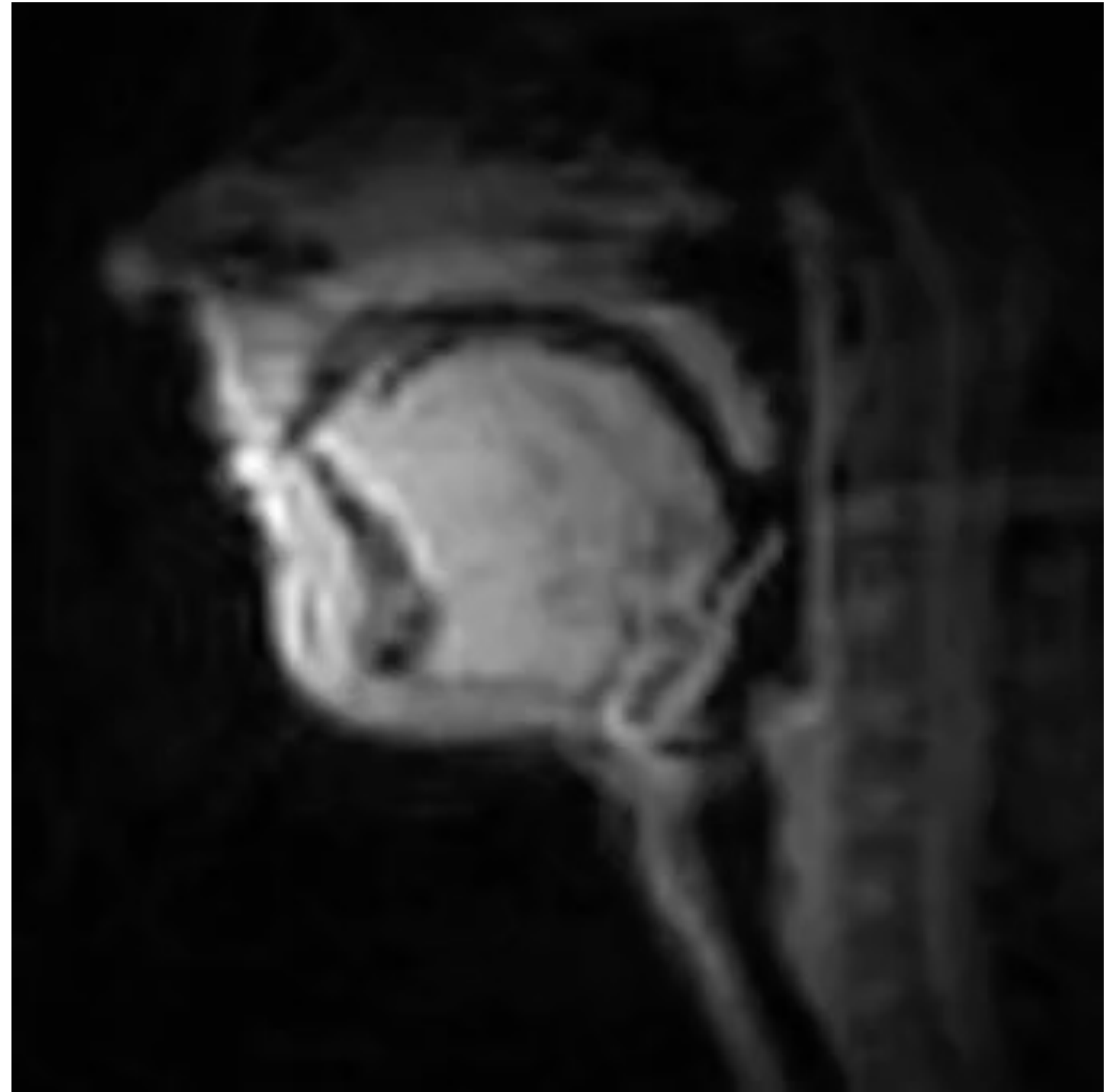
Figure of Ken Stevens, from Peter Ladefoged's web site at UCLA

USC's SAIL Lab Shri Narayanan

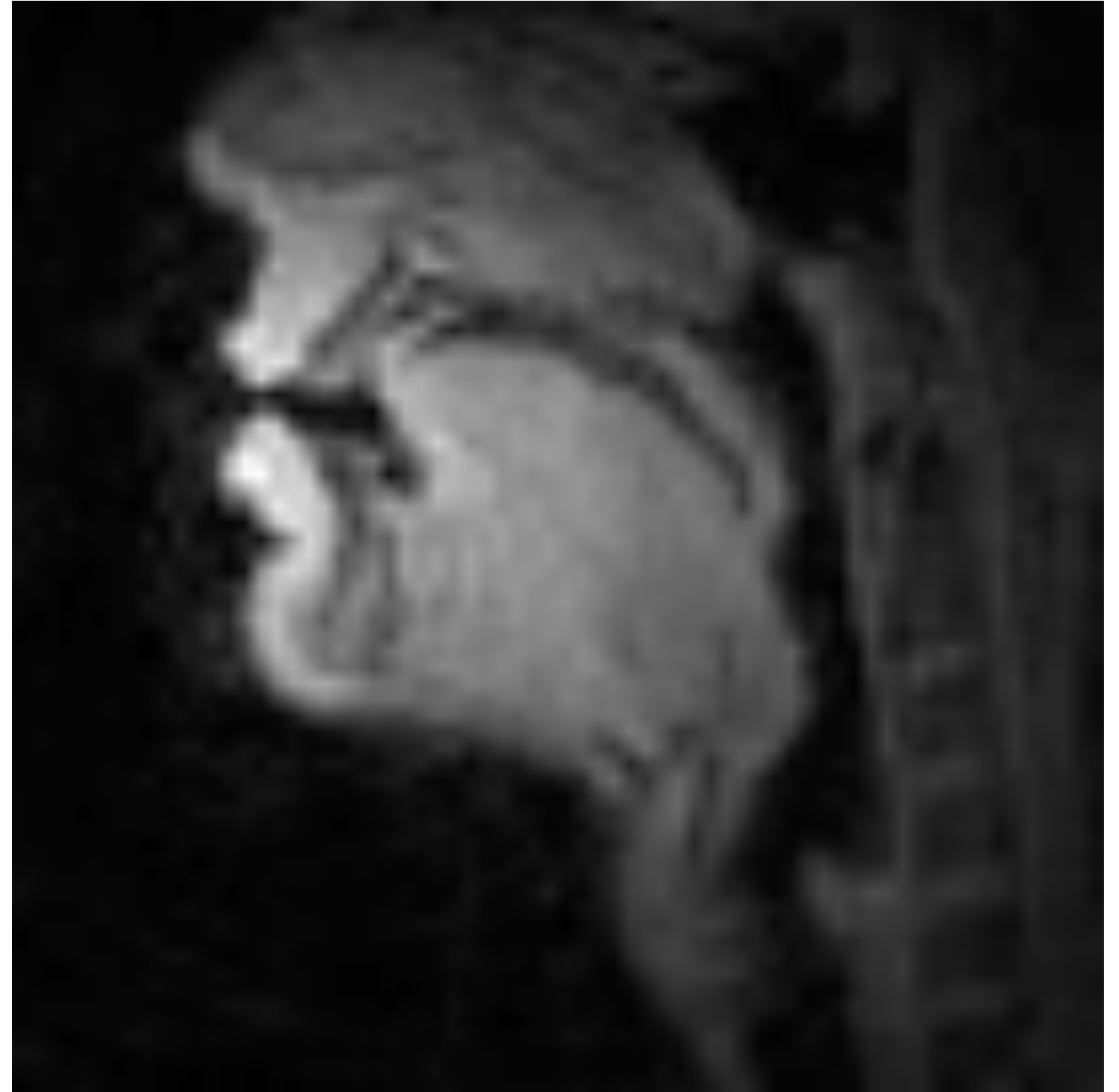
Modern
MRI
Analysis



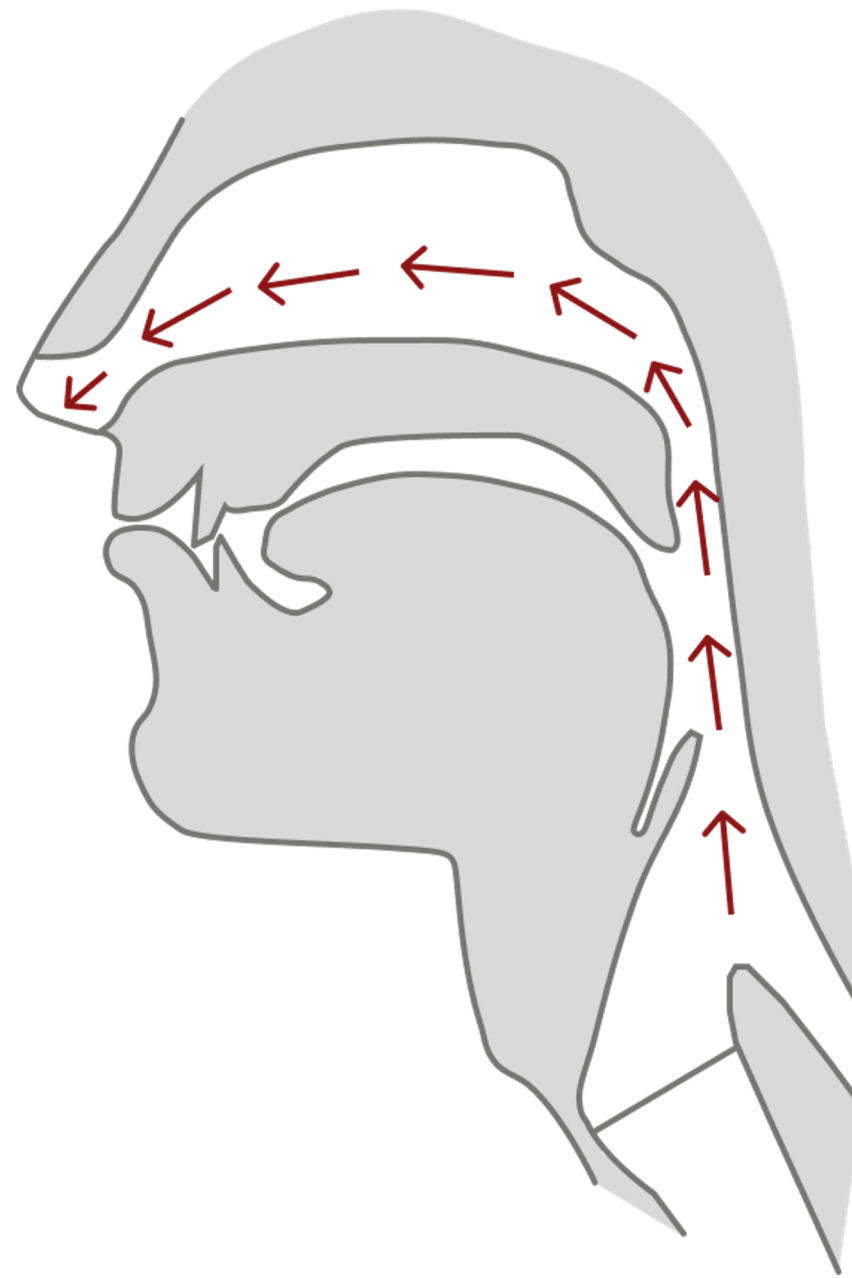
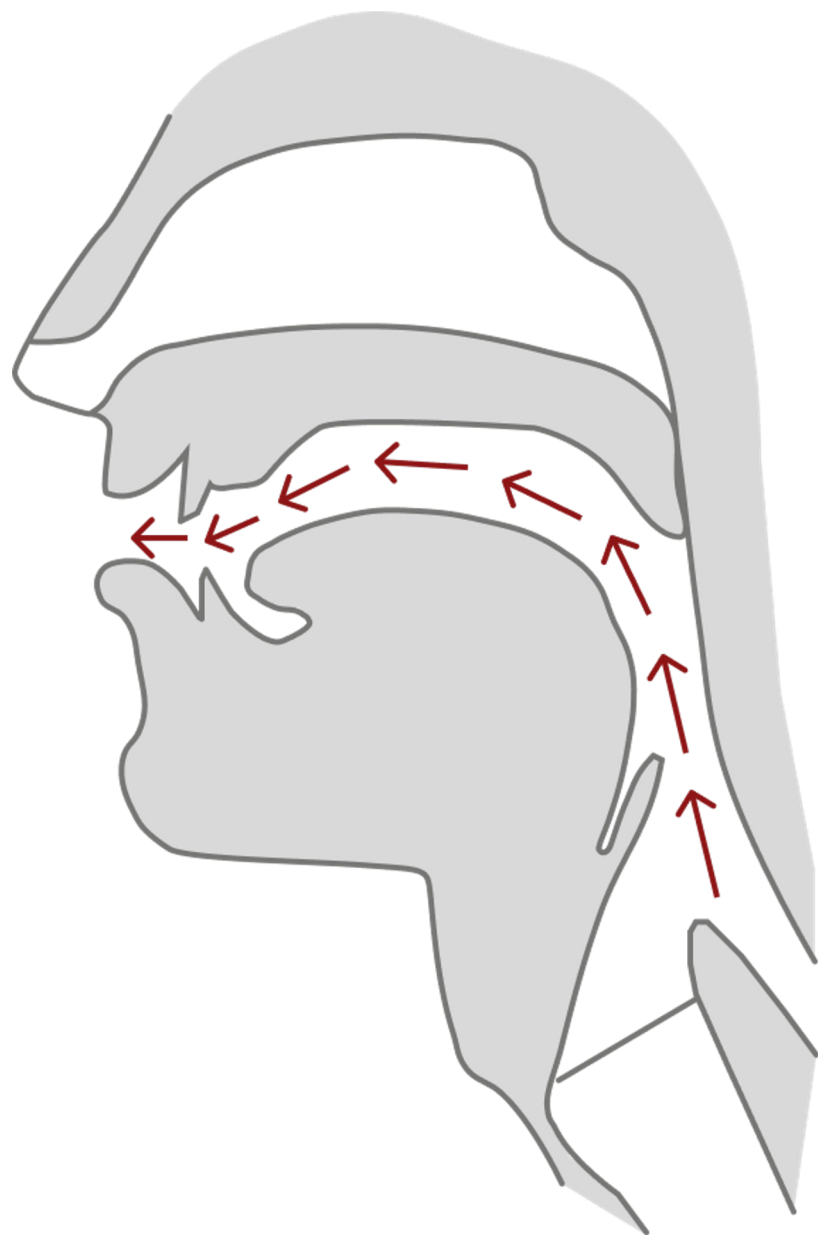
English



Tamil

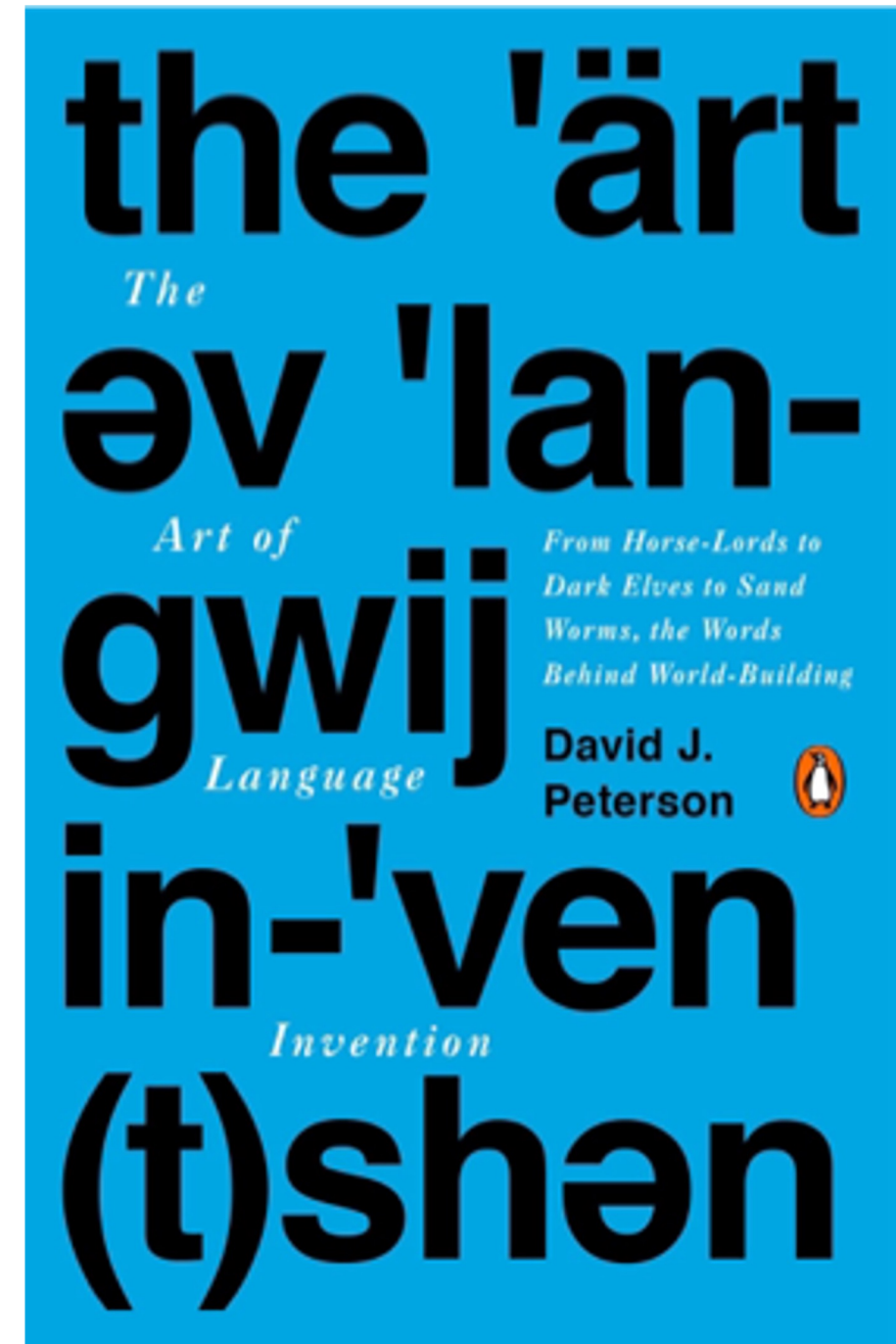


Oral vs Nasal Sounds



The Art of Language Invention

- Fun, informative book that includes phonetics and phonotactics
- [David Peterson Talk at Google](#)



Articulatory Phonetics

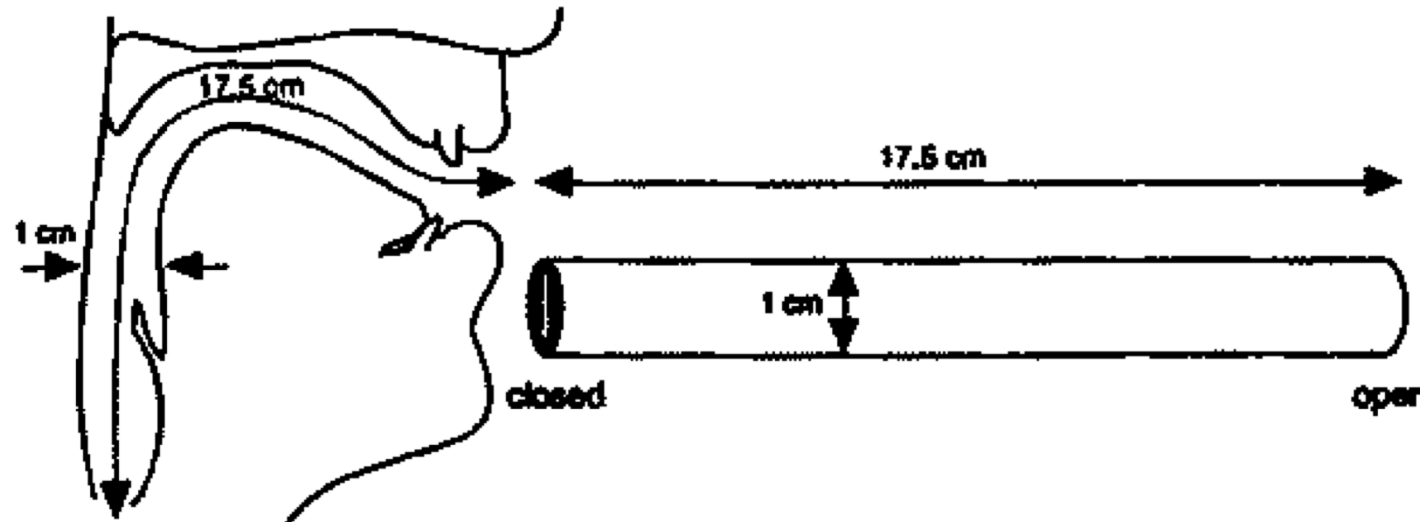
Speech
Recognition

Speech
Recognition

Acoustic Phonetics

The waveform: resonances of the vocal tract

The human vocal tract as an open tube



Air in a tube of a given length will tend to vibrate at the resonance frequency of tube.

Sound is compression waves of air particles

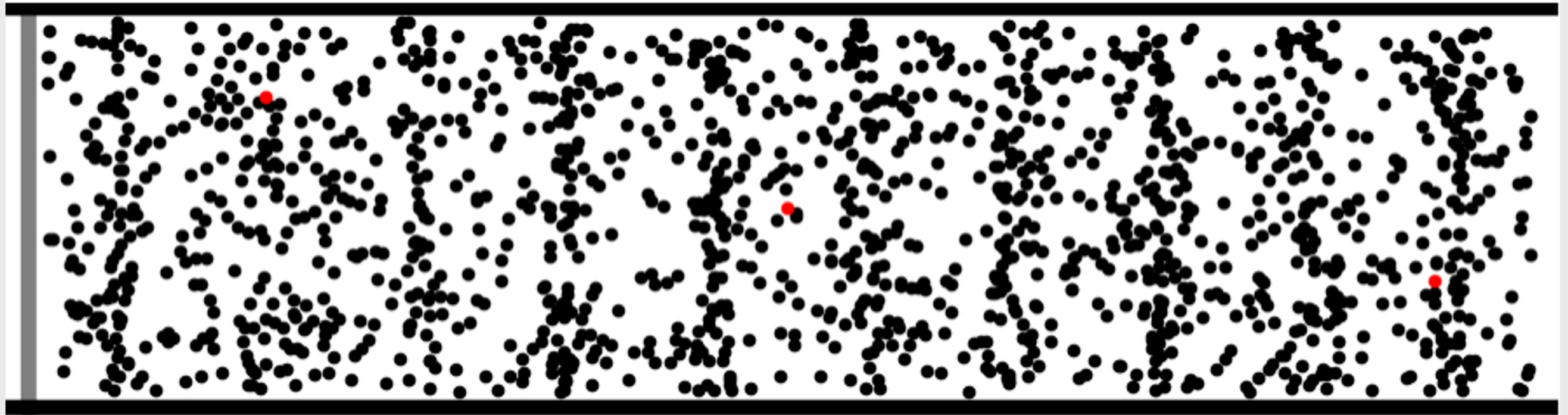
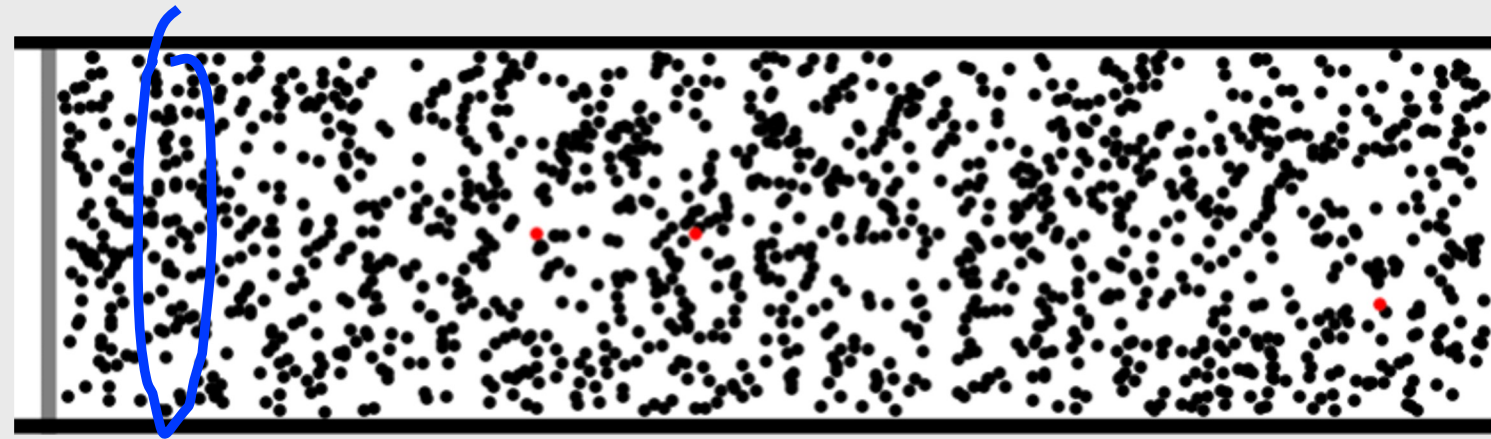
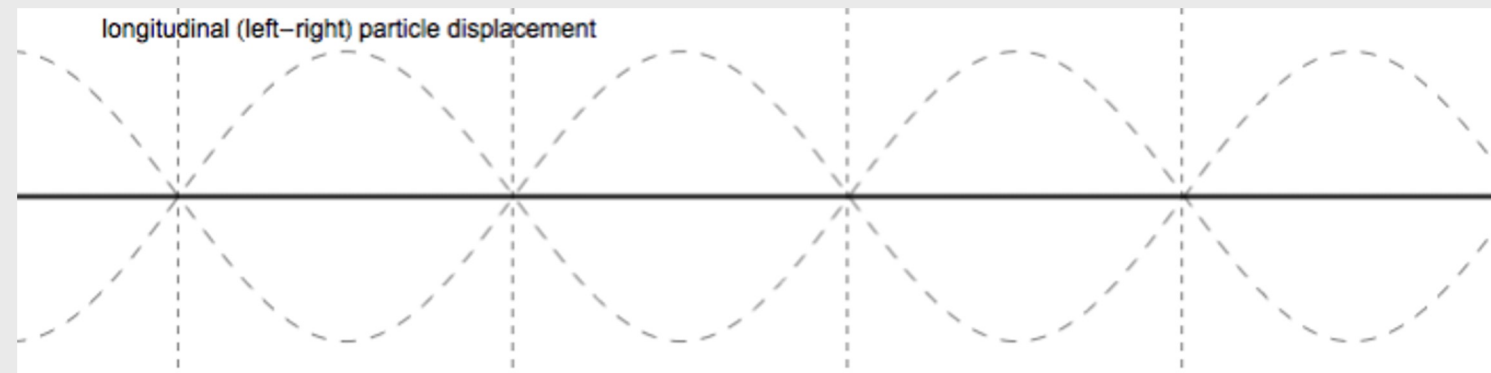


Image: Dan Russell (2011)

Sound Waves are Longitudinal Waves



**Particle
Displacement**



Pressure

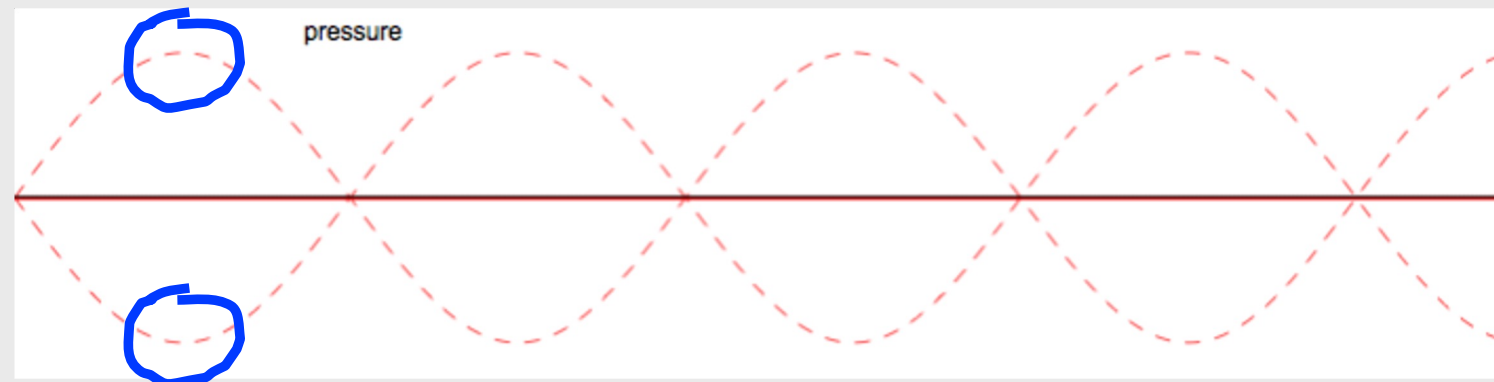
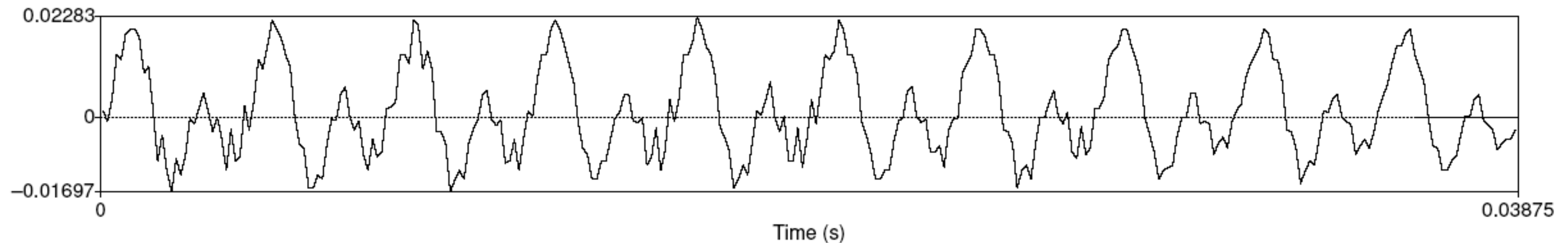


Image: Dan Russell
(2011)

Speech sound waves

The shape of the mouth enhances some frequencies and dampens others

A little piece from the waveform of the vowel [iy]



X axis: time.

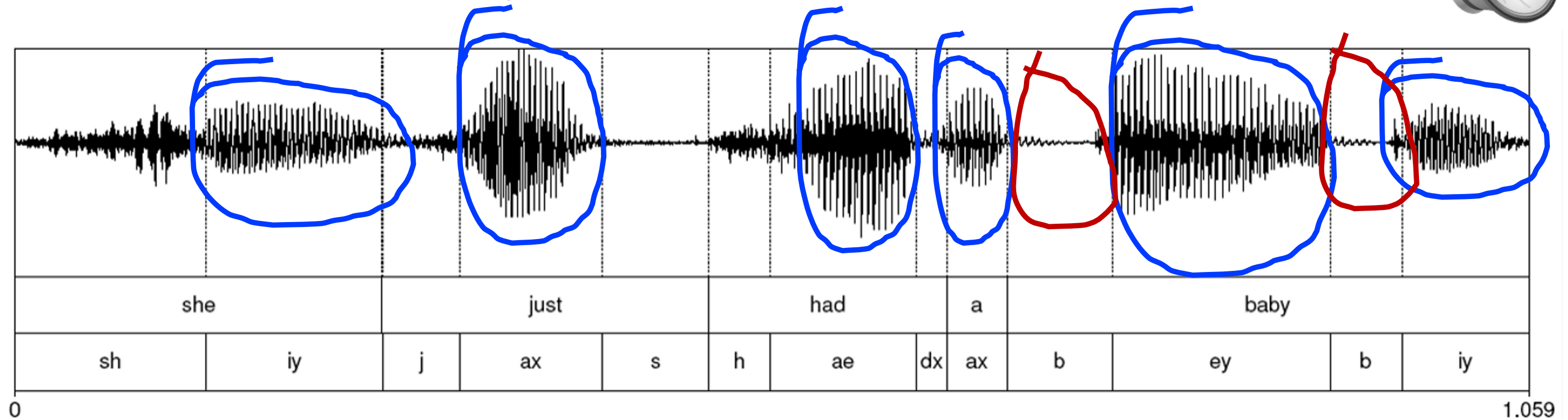
Y axis: Amplitude = air pressure at that time

+: compression

0: normal air pressure,

-: rarefaction

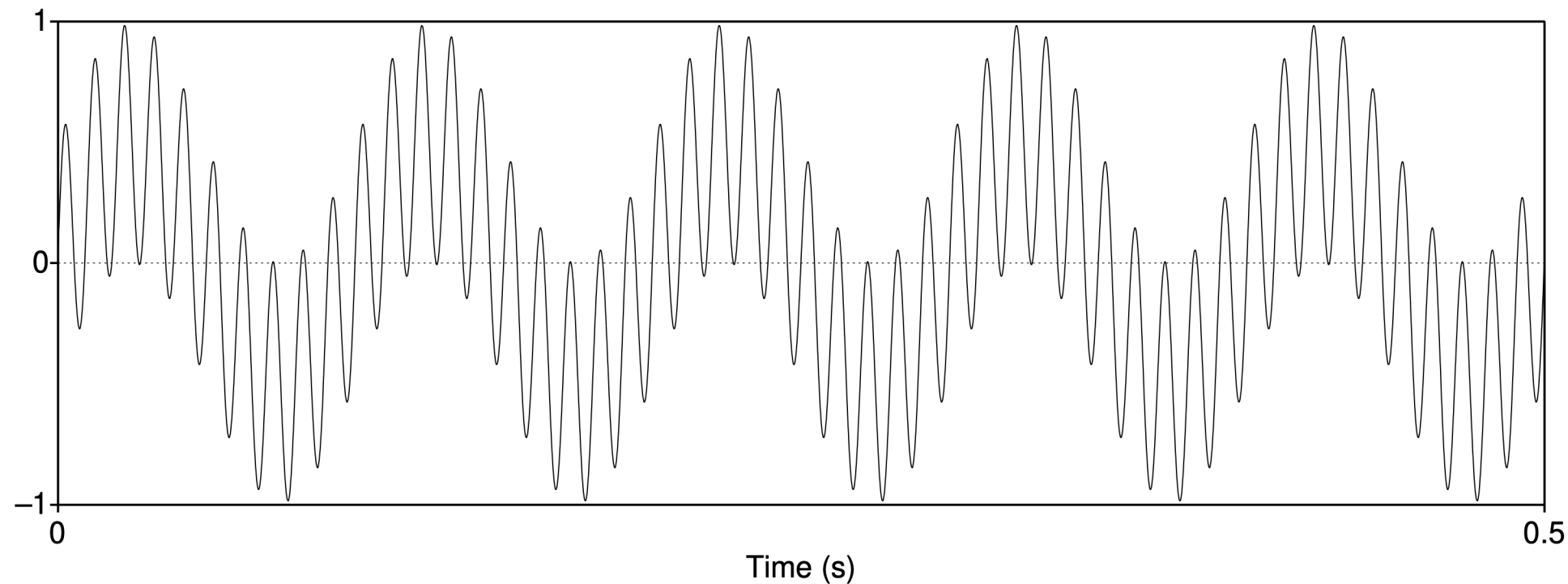
She Just Had a Baby



- Vowels vowels have voicing, regular peaks
- Stop consonants like "b" have closure followed by release

The concept of a spectrum

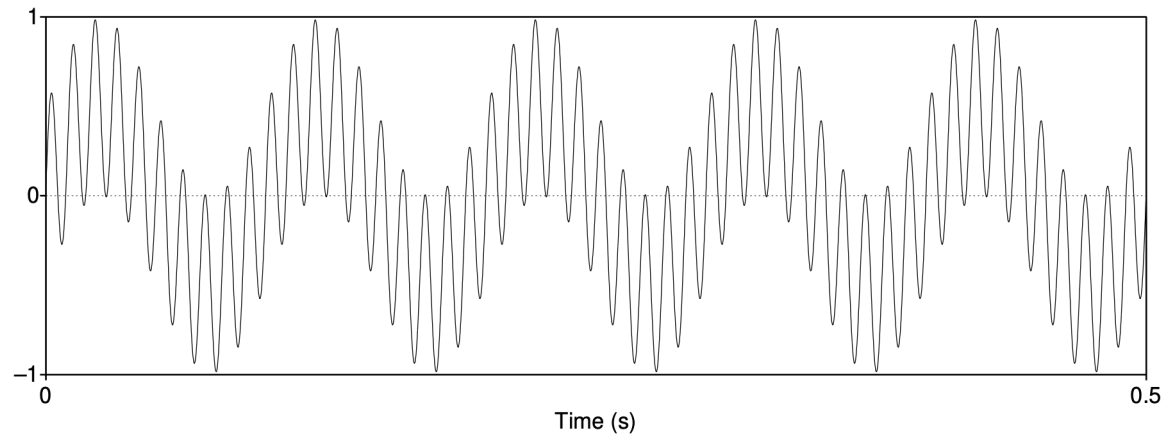
Every complex wave can be represented as a sum of sine waves of different frequencies



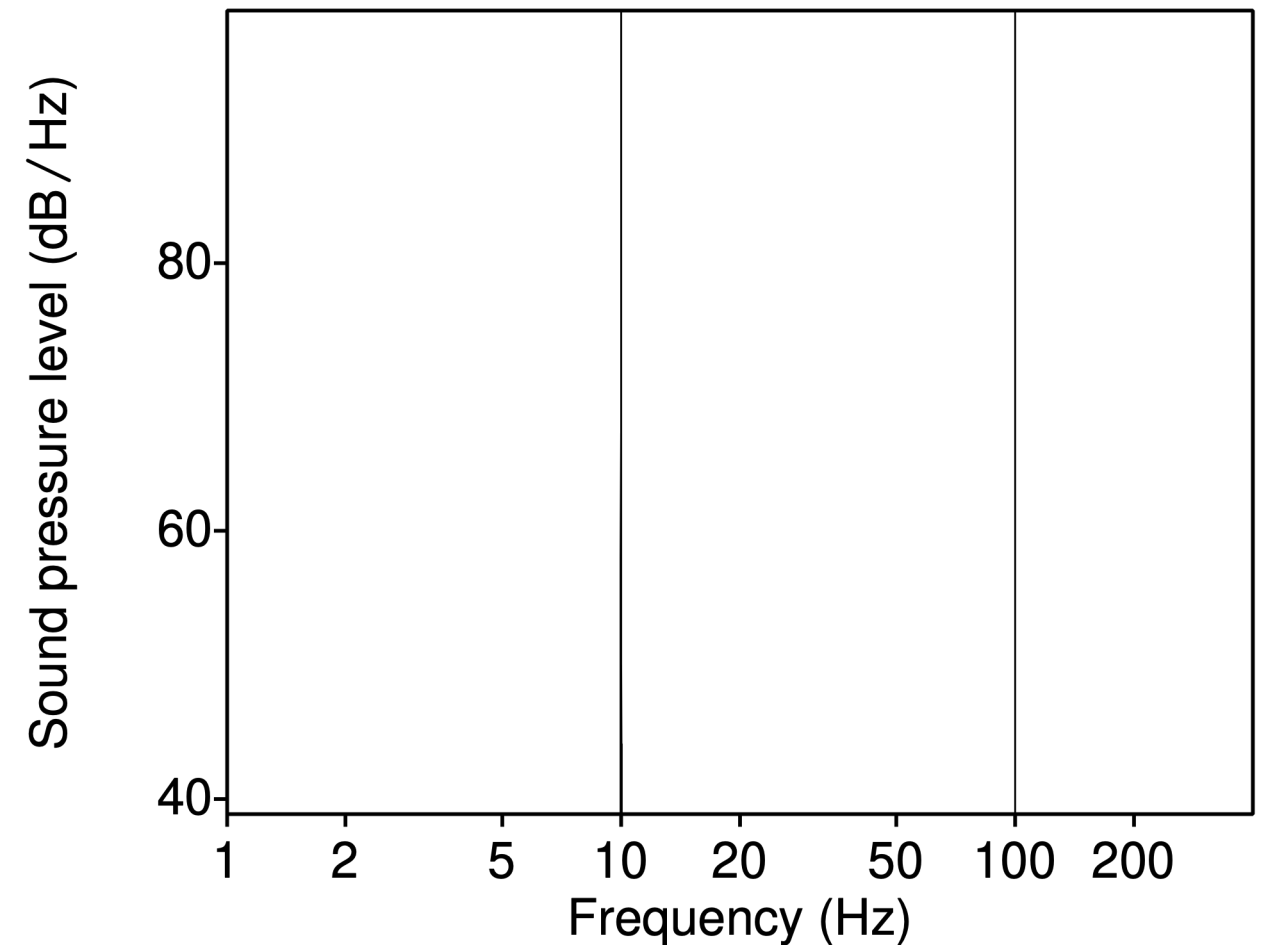
- One of 10 Hz (repeats 5 times in 0.5 seconds)
- One of 100 Hz

The concept of a spectrum

Every complex wave can be represented as a sum of sine waves of different frequencies

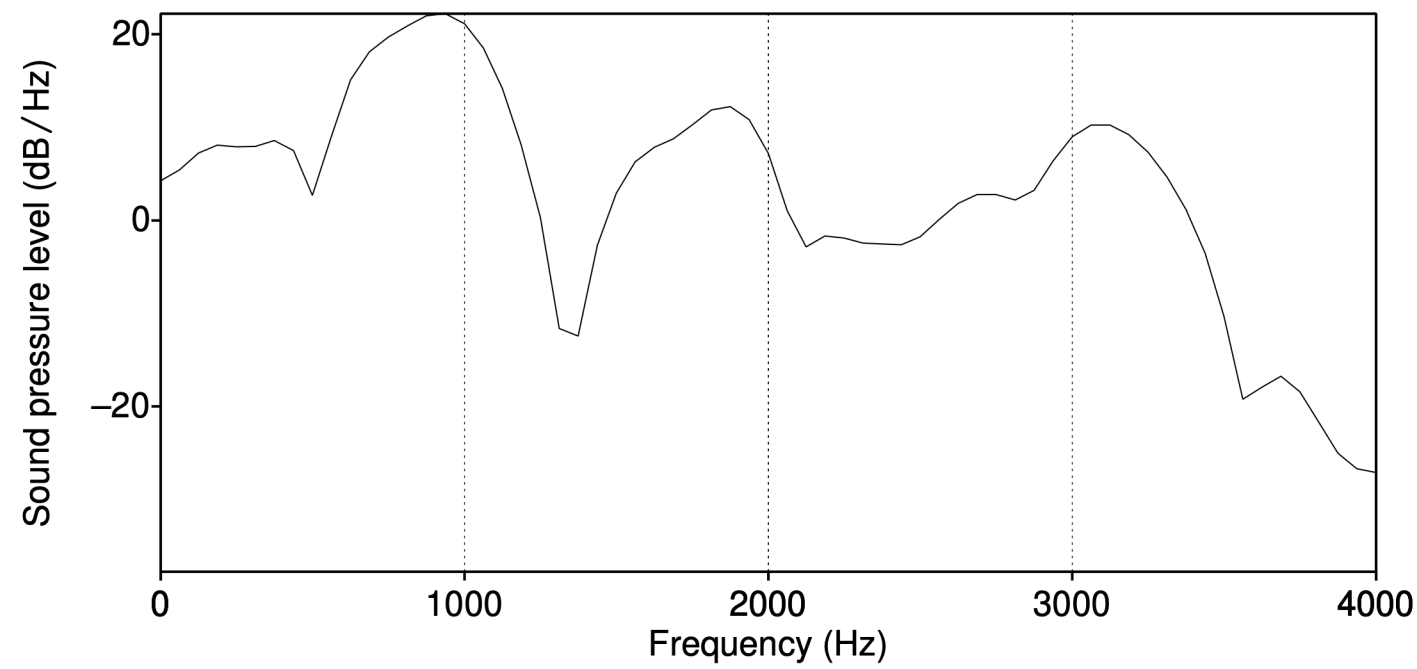
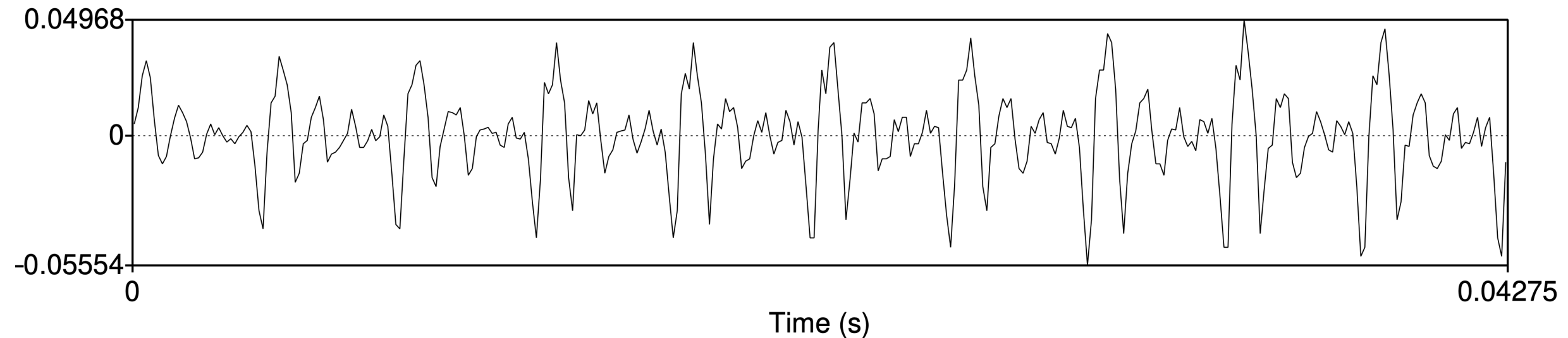


- One of 10 Hz
- One of 100 Hz

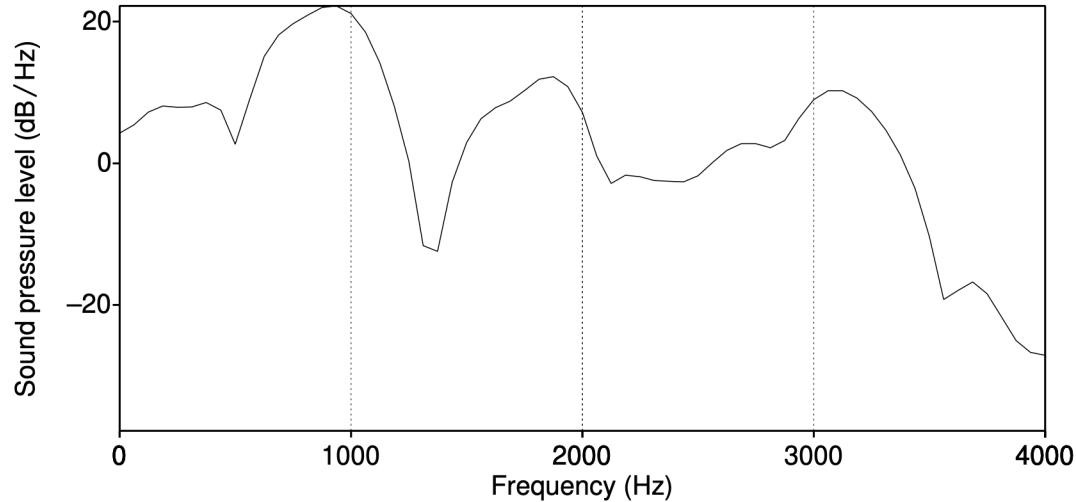
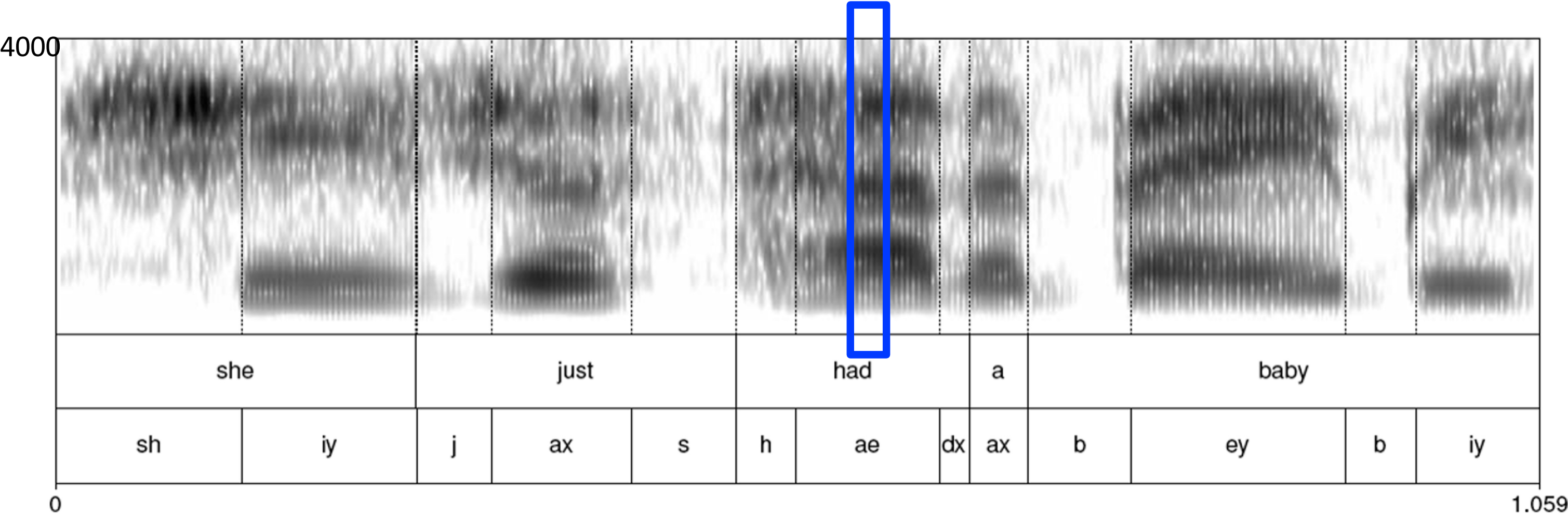


A wavfile frame and its spectrum

Part of the vowel ae in "had"

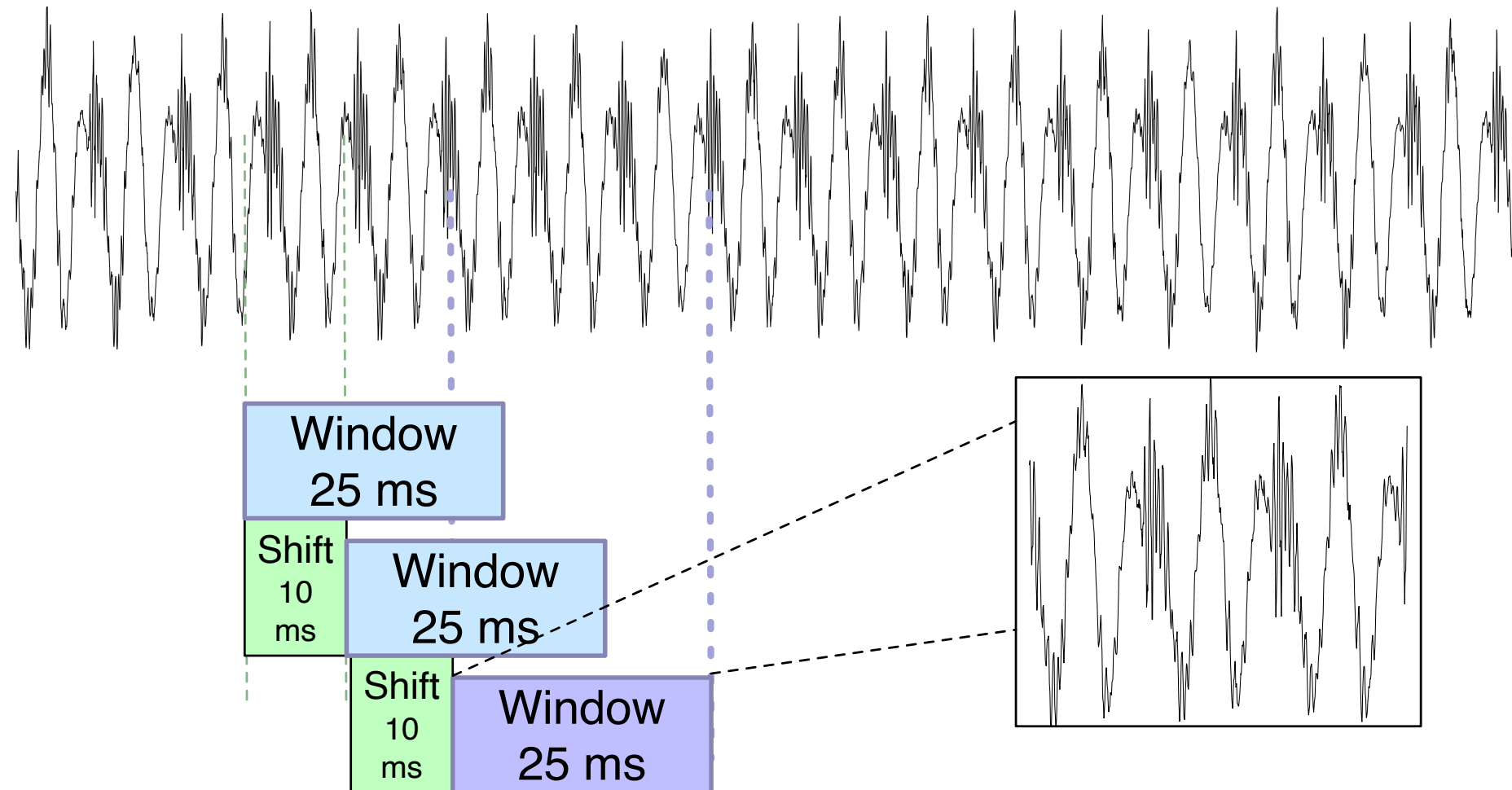


Spectrogram: Spectrum + Time Dimension

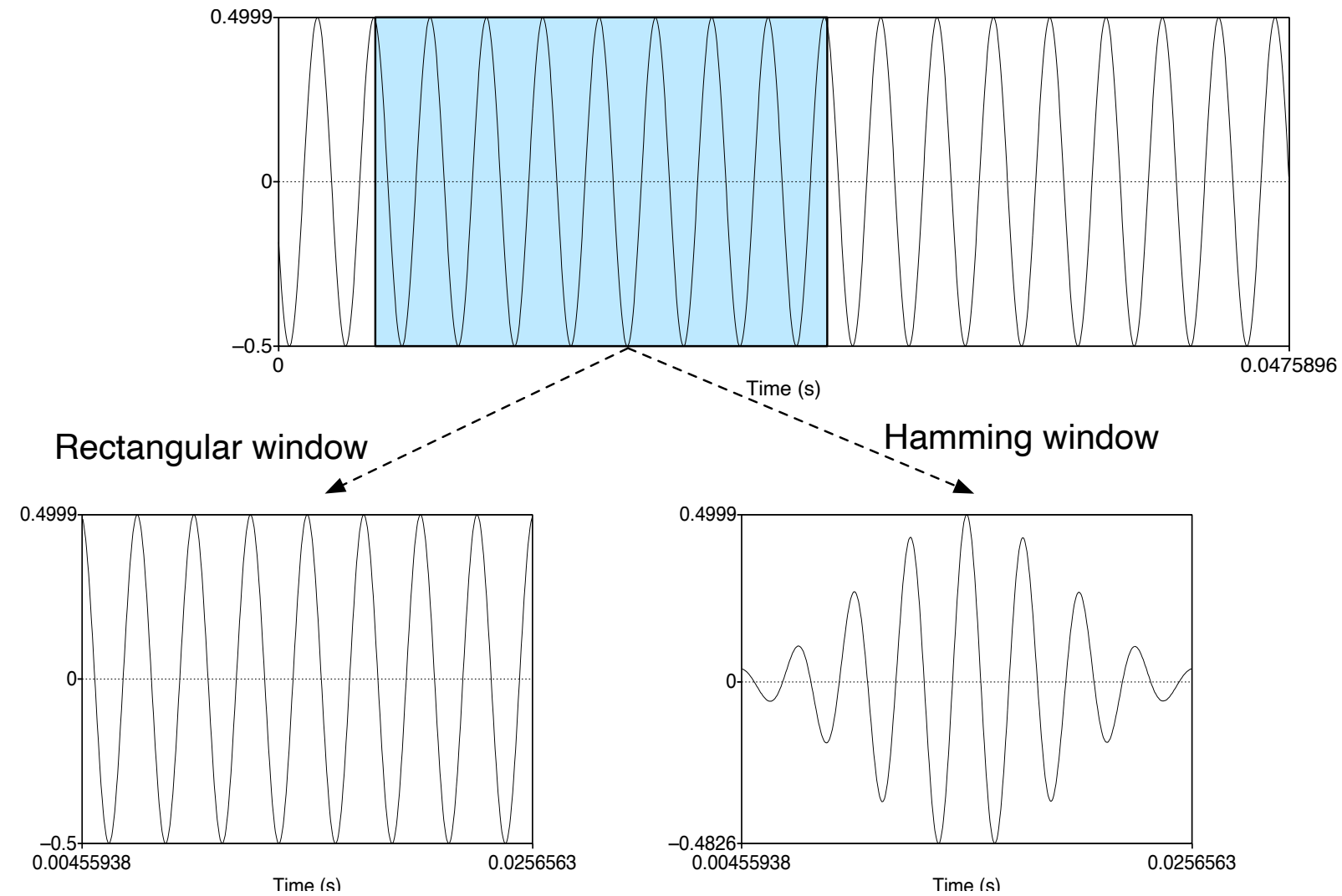


Summary steps : getting a frame of speech

Extracting a frame of speech



Windowing the frame

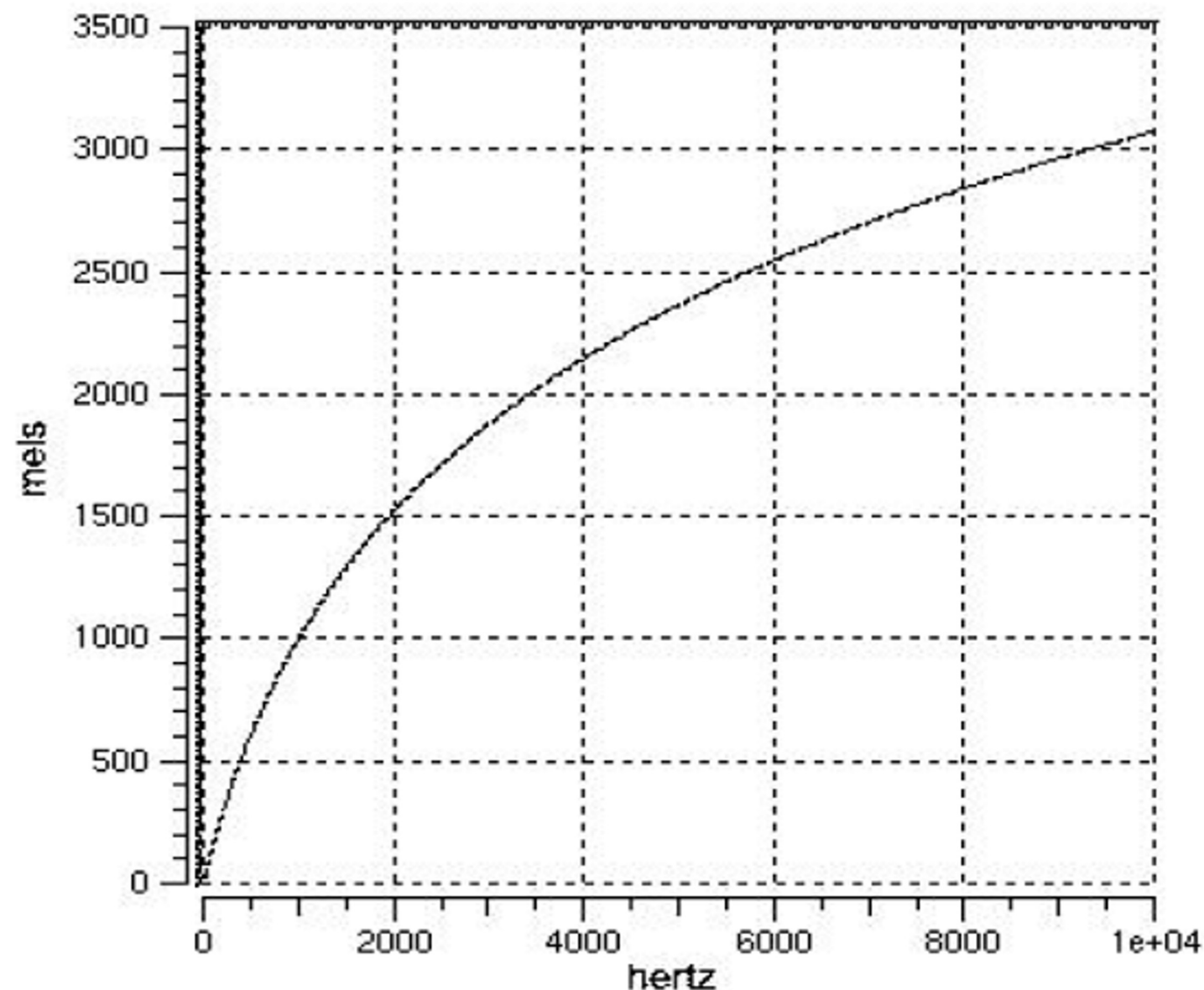


Mel-scale spectrum

Human hearing is not equally sensitive to all frequency bands

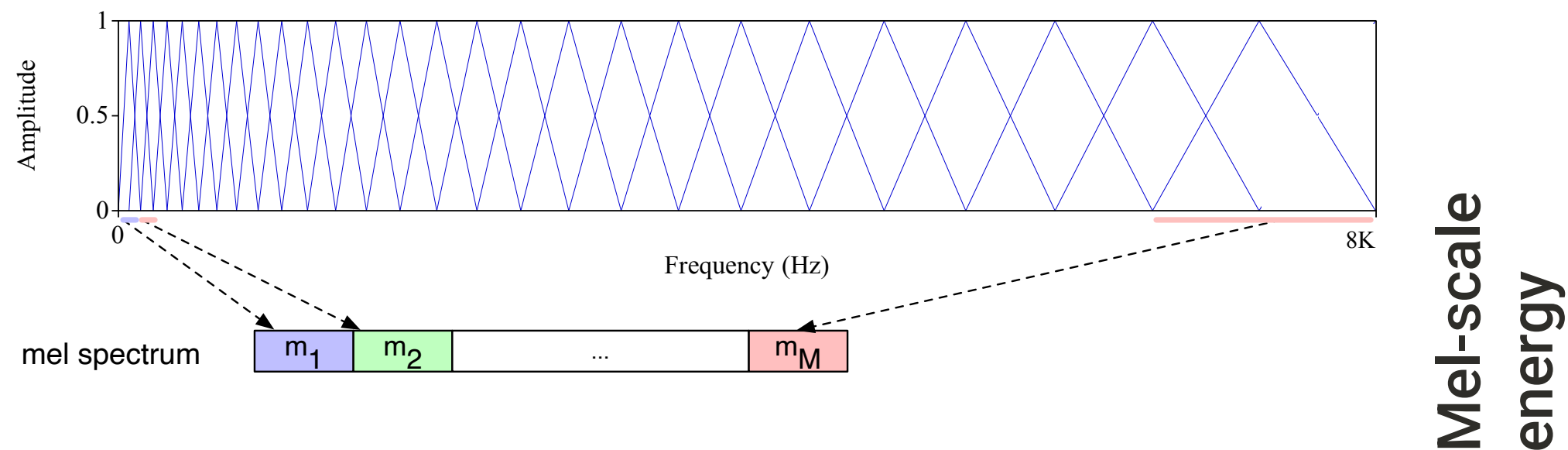
Less sensitive at higher frequencies, > 1000 Hz

I.e. human perception of frequency is non-linear:



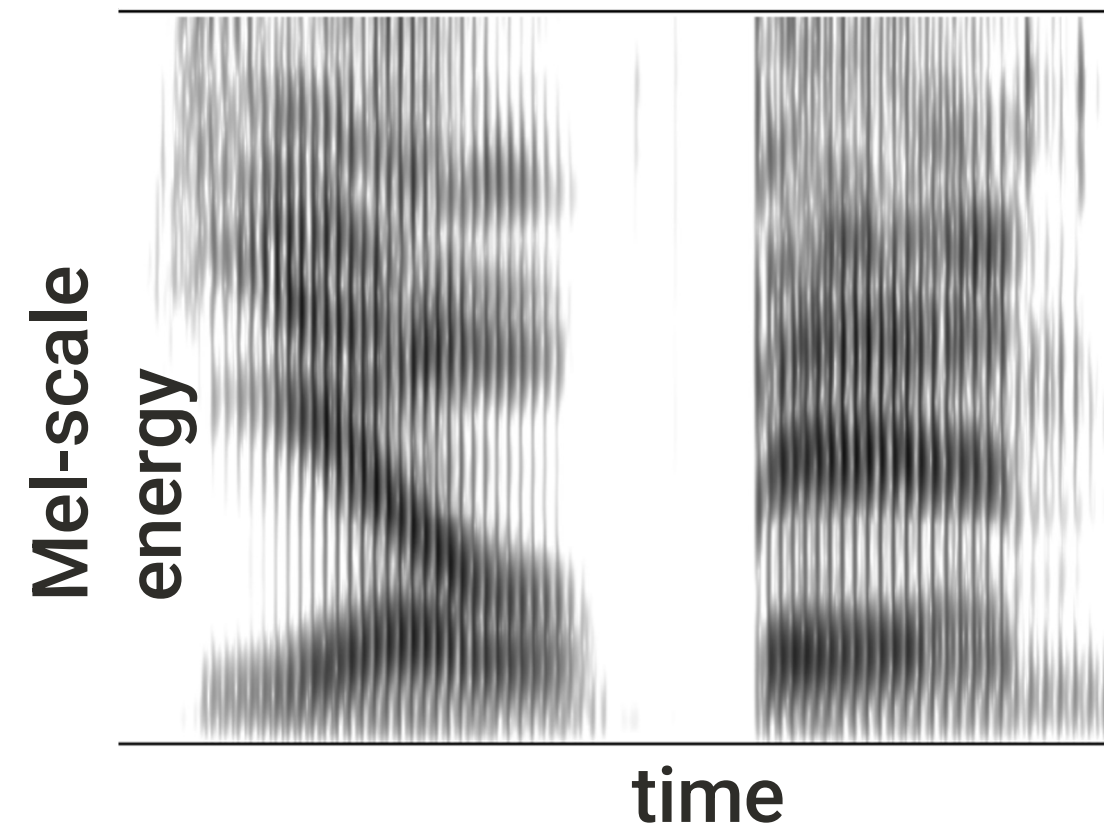
Mel Filter Bank Processing

- Mel Filter bank
 - Fine resolution at low frequencies
 - Less resolution at high frequencies



Mel Filter Bank Processing

- Mel Filter bank
 - Fine resolution at low frequencies
 - Less resolution at high frequencies



Speech
Recognition

Acoustic Phonetics

Speech
Models

Convolution Networks

A convolutional layer

Passes a "kernel" over the input to extract features

Speech: 1 dimensional, slide kernel over signal in time dimension

Images: 2 dimensional, slide kernel horizontally and vertically over input

Basic assumptions

Input a sequence of vectors $\mathbf{x}_1 \dots \mathbf{x}_t$

Output a sequence of vectors $\mathbf{z}_1 \dots \mathbf{z}_t$ (same length t)

A kernel = small vector of weights $\mathbf{w}_1 \dots \mathbf{w}_k$

We'll **convolve** the kernel with the input

1. Flip the kernel left-to-right
2. Pass the kernel frame by frame (temporally) across the input
 - At each frame take dot product of kernel with input
3. The output is the resulting sequence of dot products

Basic assumptions

Input a sequence of vectors $\mathbf{x}_1 \dots \mathbf{x}_t$

Output a sequence of vectors $\mathbf{z}_1 \dots \mathbf{z}_t$ (same length t)

A kernel = small vector of weights $\mathbf{w}_1 \dots \mathbf{w}_k$

We'll **convolve** the kernel with the input

- ~~1. Flip the kernel left-to-right~~
2. Pass the kernel frame by frame (temporally) across the input
 - At each frame take dot product of kernel with input
3. The output is the resulting sequence of dot products

Intuition of convolution

What is convolution?

- Pass the kernel frame by frame (temporally) across the input, at each frame taking dot product with input

Convolution is **finding regions in the signal that are similar to the kernel**

- The dot product is high when two vectors are similar

Example

How to compute $\mathbf{x} * \mathbf{w}$:

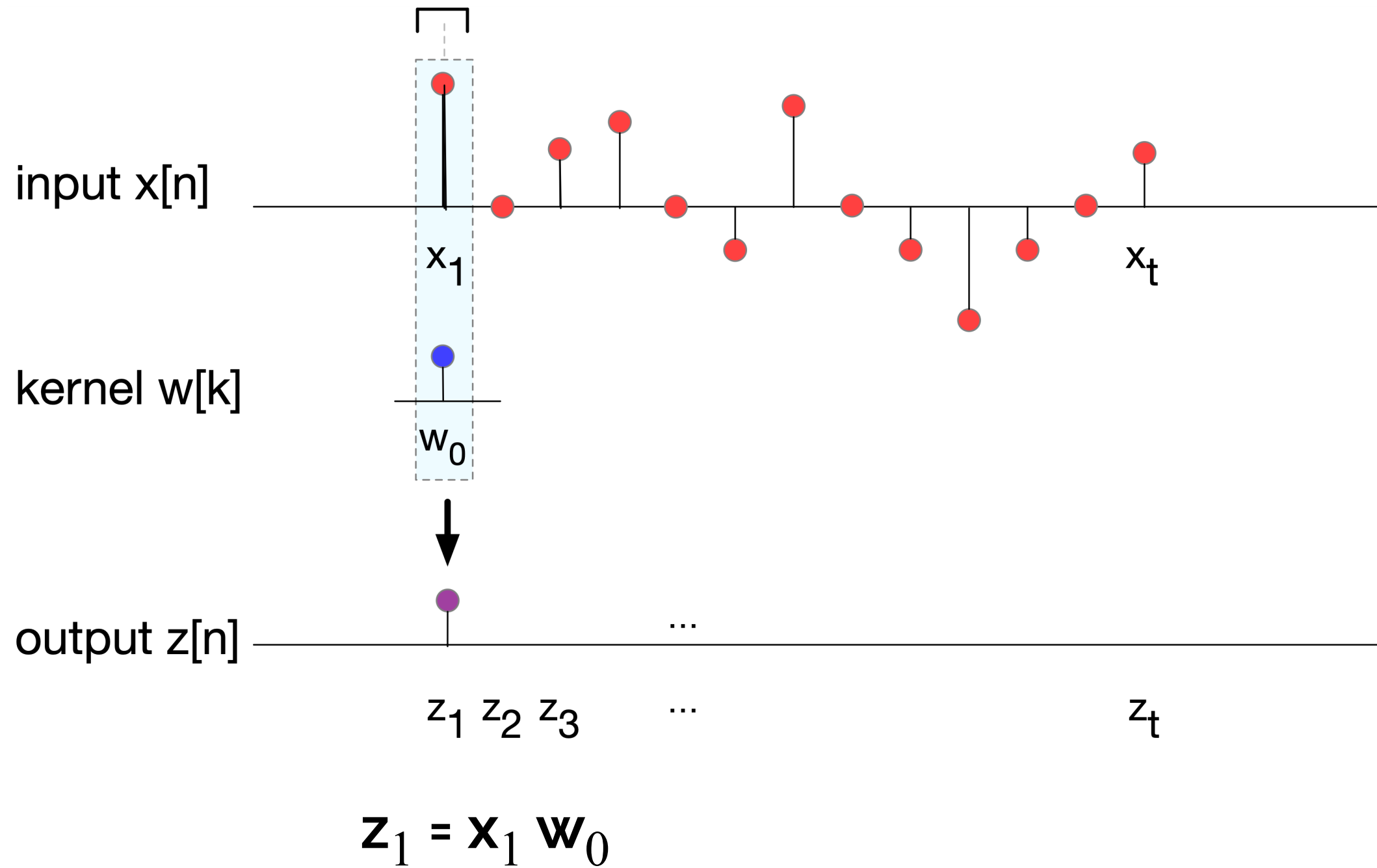
- the convolution of
- a single vector $\mathbf{x}$
- with a kernel vector $\mathbf{w}$ of width (receptive field) k

Simplest case: *Convolution with width-1 kernel:*

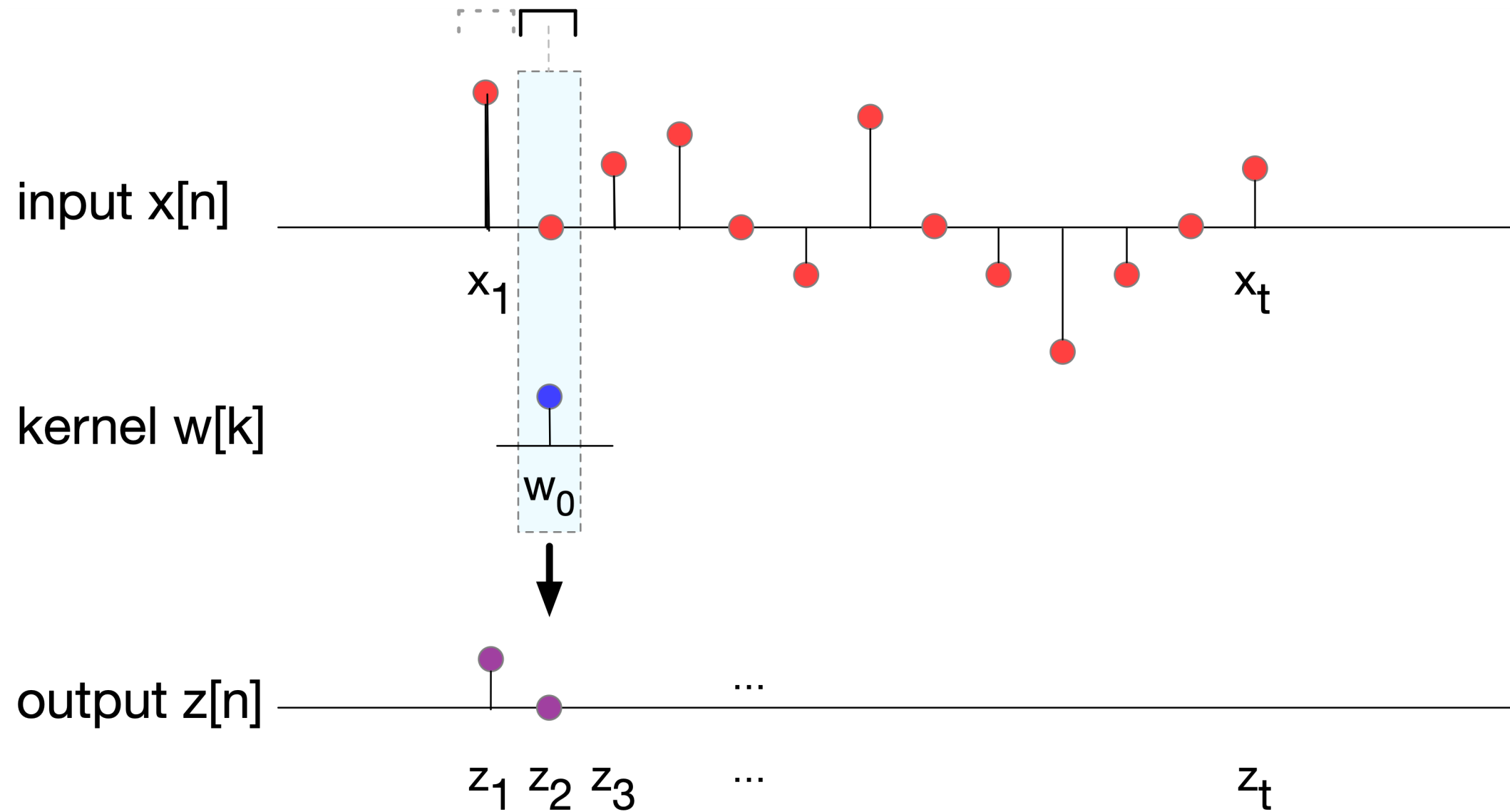
$$\mathbf{z}_j = \mathbf{x}_j \mathbf{w}_0 \quad \forall j : 1 \leq j \leq t$$

(dot product is just scalar product)

Computing $\mathbf{x} * \mathbf{w}$: $\mathbf{z}_j = \mathbf{x}_j \mathbf{w}_0 \quad \forall j : 1 \leq j \leq t$

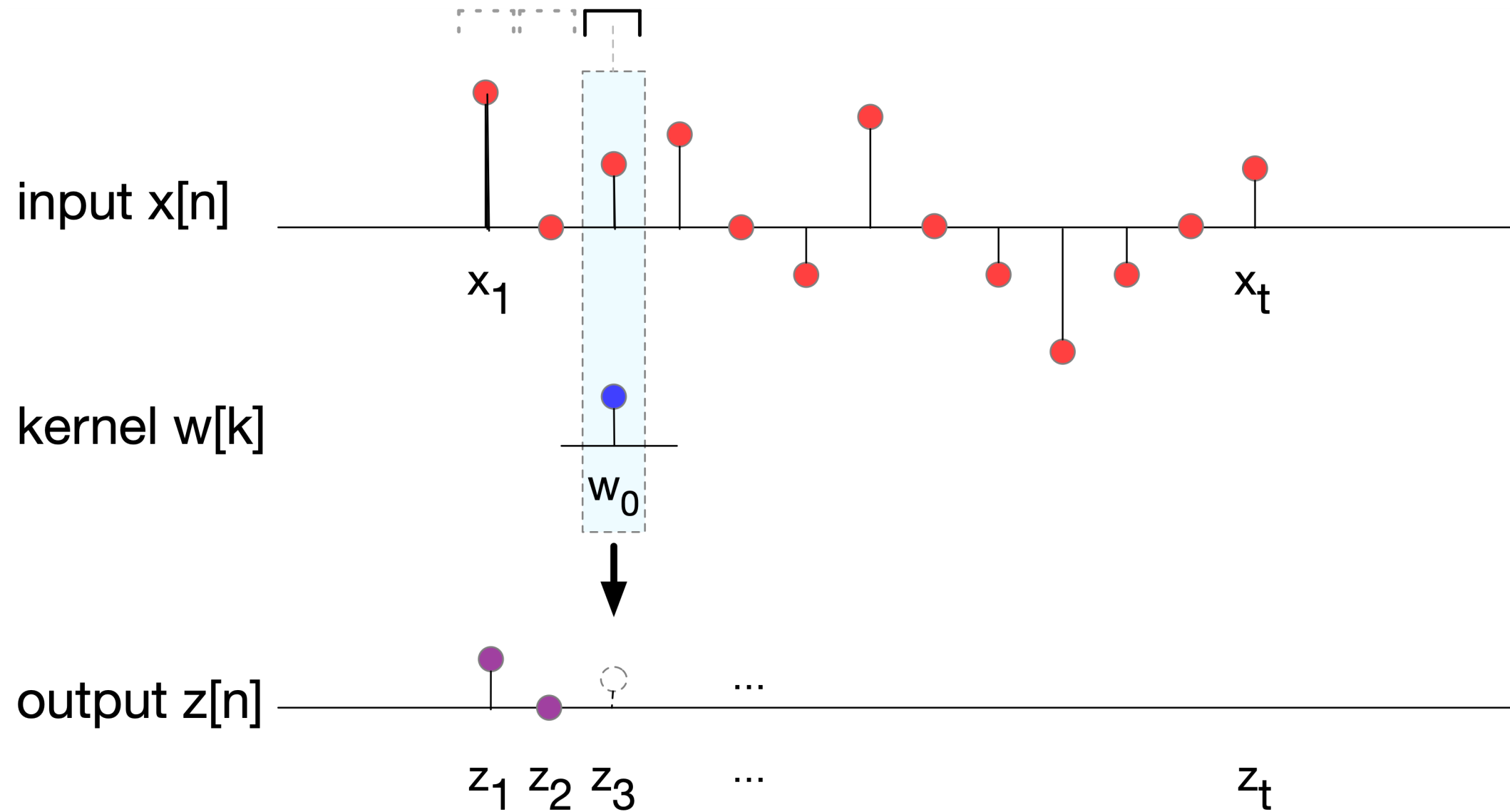


Computing $\mathbf{x} * \mathbf{w}$: $\mathbf{z}_j = \mathbf{x}_j \mathbf{w}_0 \quad \forall j : 1 \leq j \leq t$



$$\mathbf{z}_2 = \mathbf{x}_2 \mathbf{w}_0$$

Computing $\mathbf{x} * \mathbf{w}$: $\mathbf{z}_j = \mathbf{x}_j \mathbf{w}_0 \quad \forall j : 1 \leq j \leq t$



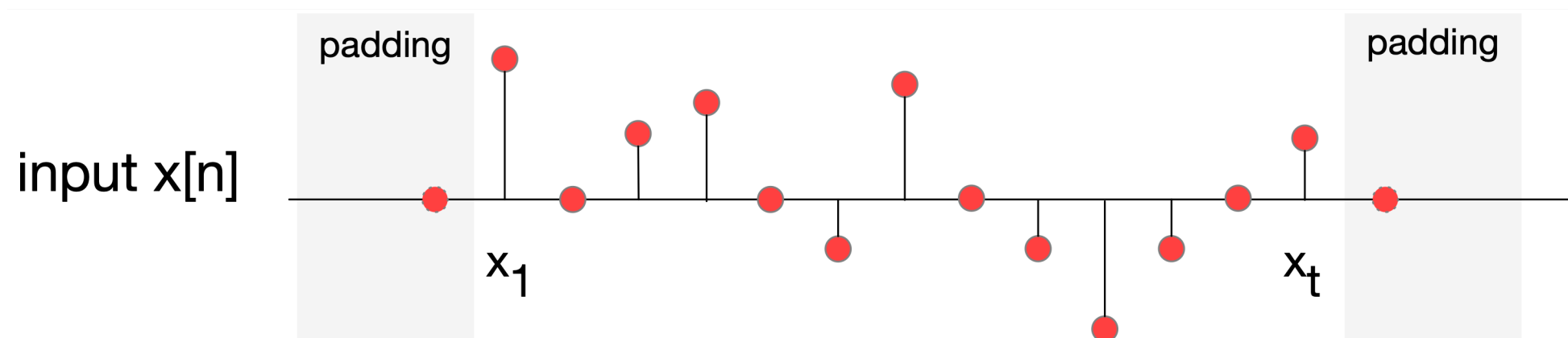
$$\mathbf{z}_3 = \mathbf{x}_3 \mathbf{w}_0$$

More complex example with a longer kernel

Convenient to have output length = input length

We **pad** the input by adding **p** zeros at the beginning and end of the signal.

- We start the center of the kernel at the first element x_1 ,



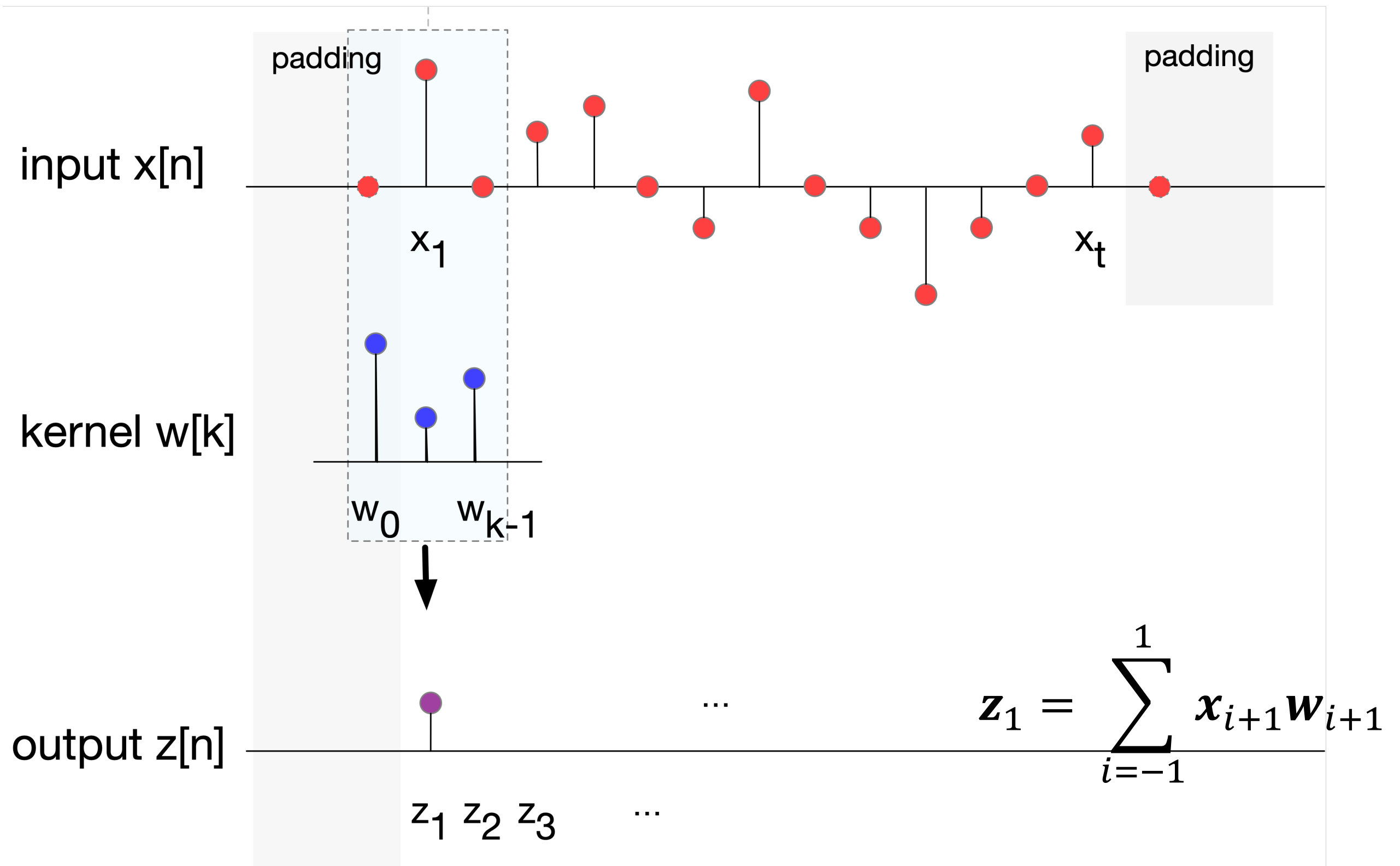
More complex example with a longer kernel

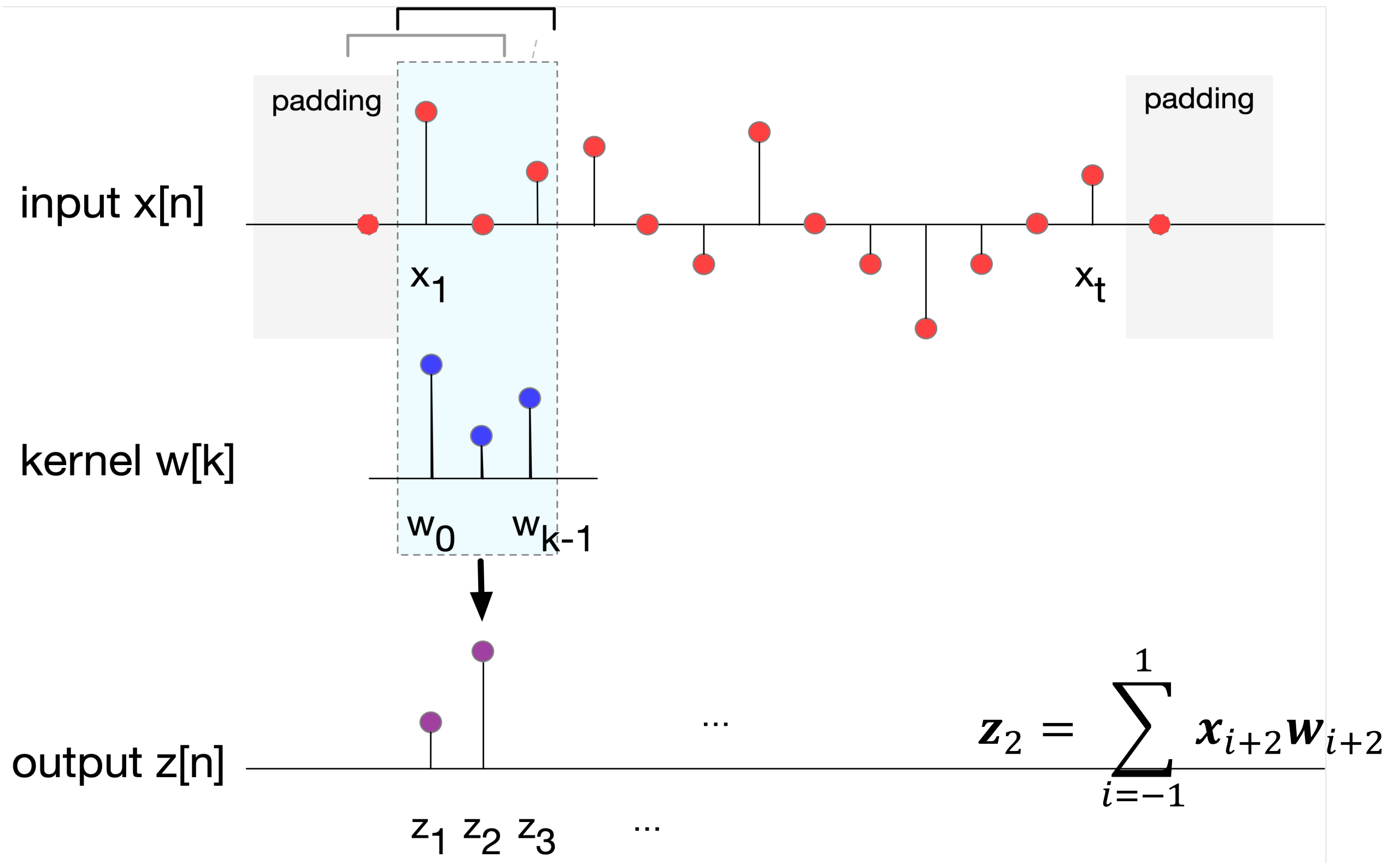
Convenient to define the kernel vector **w** as having an odd number of elements of length $k = 2p + 1$,

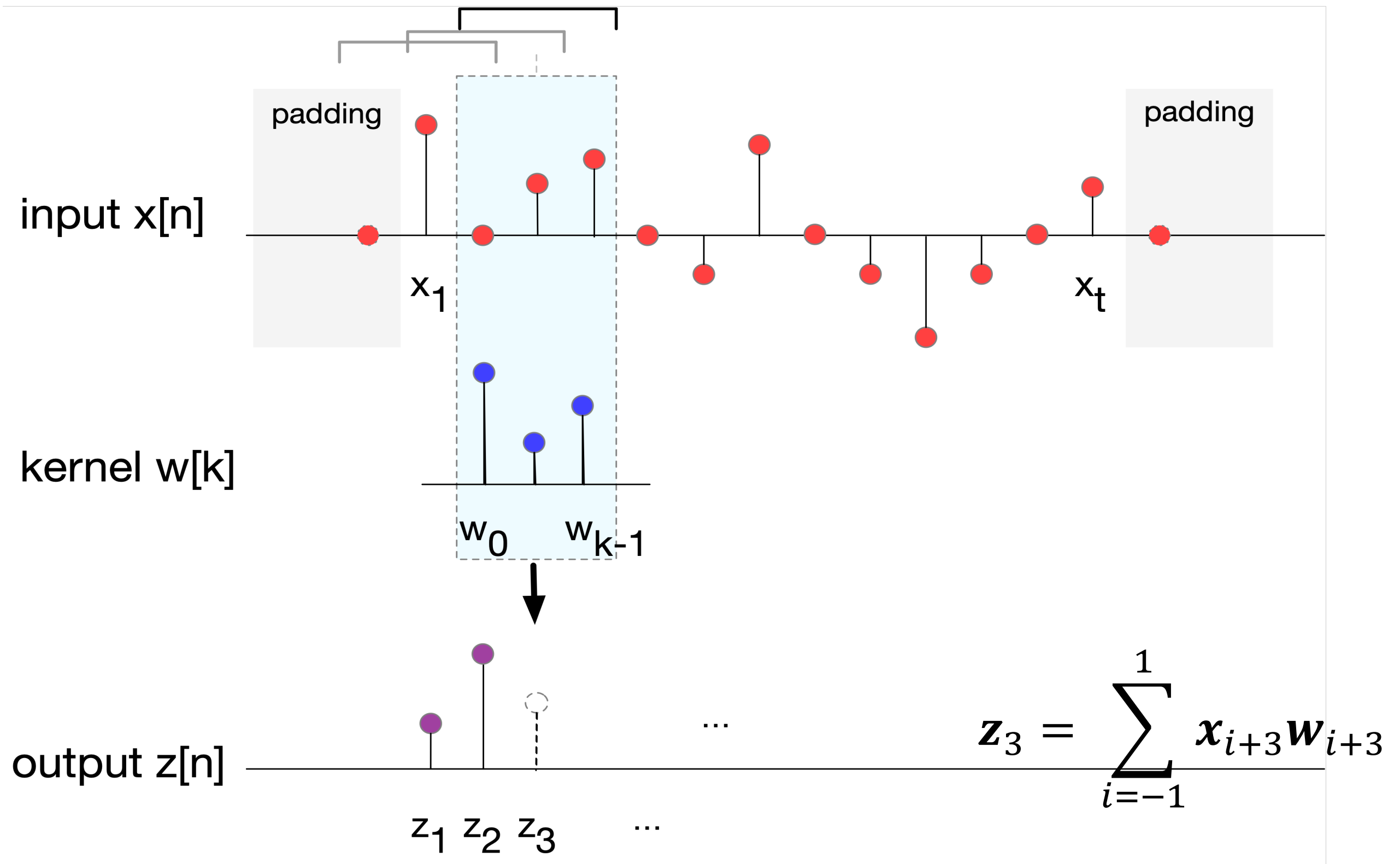
- center element has p elements on either side.

$$\mathbf{z}_j = \sum_{i=-p}^p \mathbf{x}_{j+i} \mathbf{w}_{i+p}$$

- $\mathbf{z}_j$ = dot product of $\mathbf{x} [j-p : j+p+1]$ with $\mathbf{w} [0 : 2p+1]$
- So for a kernel **w** of a receptive field of 3, $p = 1$

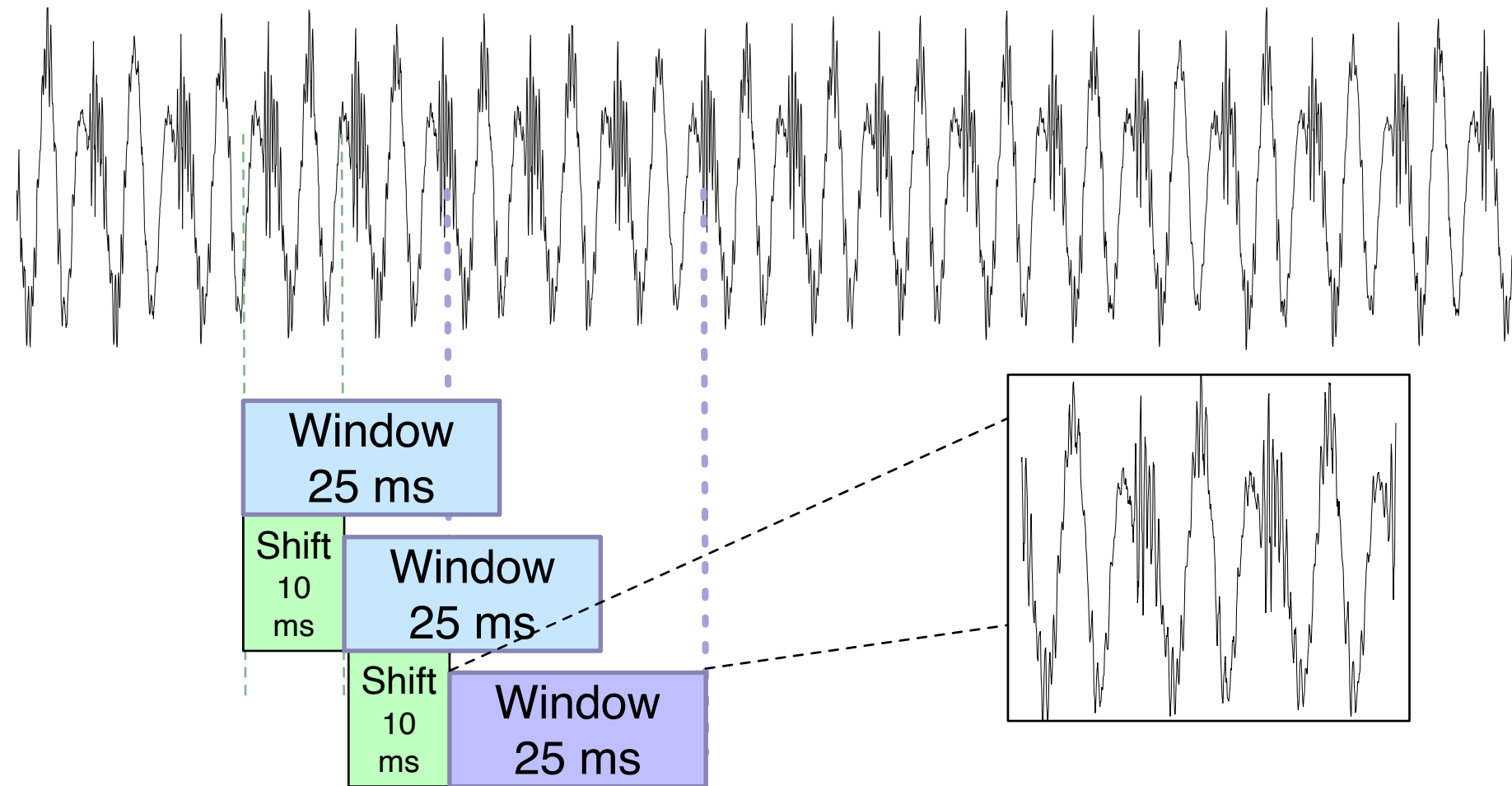






Convolution in Whisper

Receptive field is 3, so kernel is compared to 3 frames of speech



Whisper frame is 25 ms of speech, separated by 10 ms stride

So each kernel extracts 45 ms of speech

Input is not one number per frame!

Conv layer input has **128 input channels**, each $\mathbf{x}^c$

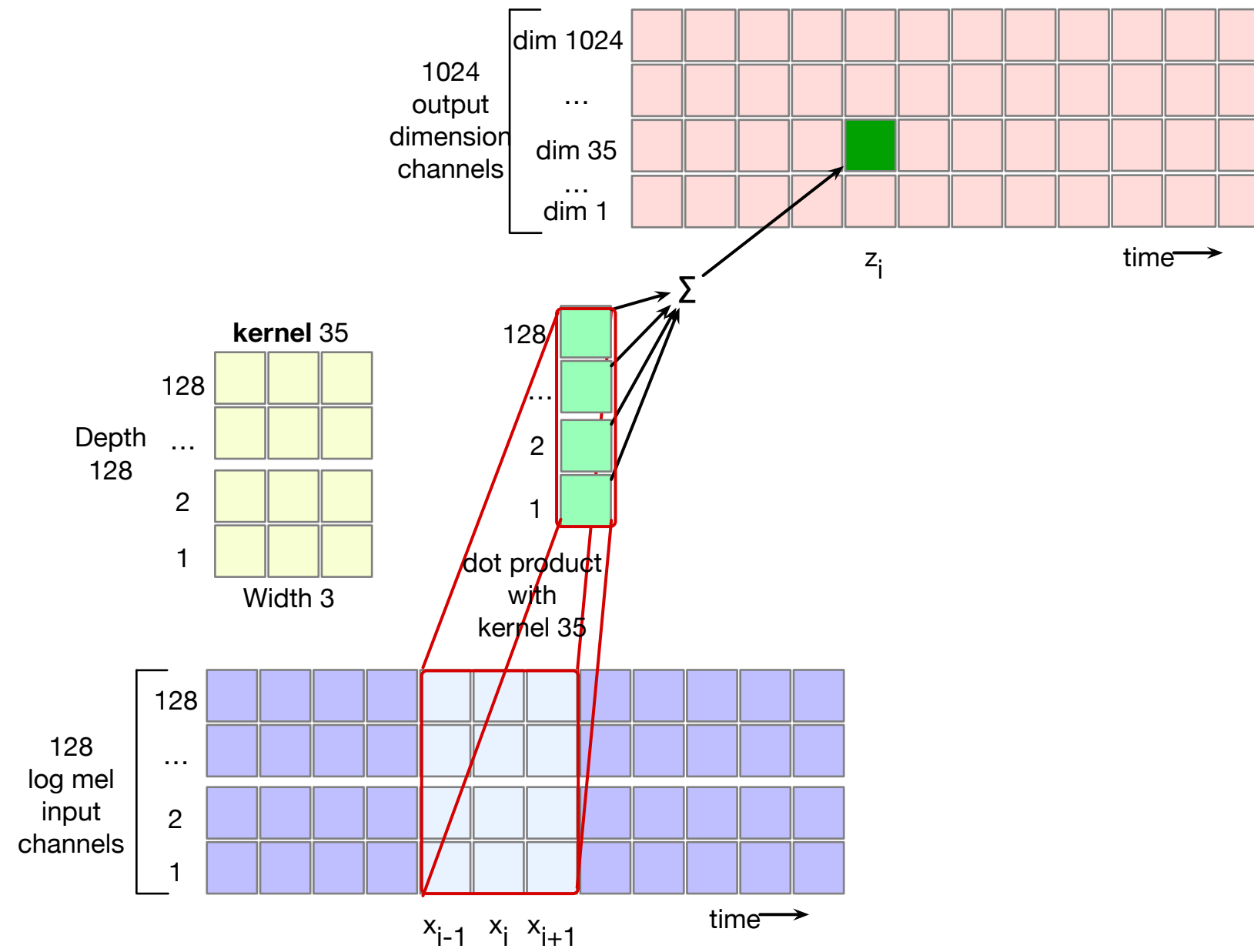
- Each of the 128 values from the 128 log mel filters in spectrum.
- Kernel has separate vector for each of these input channels
- Kernel has depth 128

The convolution sums over those 128 dot products

$$\mathbf{z} = \sum_{c=1}^{C_i} \mathbf{x}^c * \mathbf{w}$$

Output is 1024-dim embedding per frame

Convnet has **1024 output channels**



Speech
Models

Convolution Networks

Speech
Models

Whisper Introduction

Many architectures for ASR

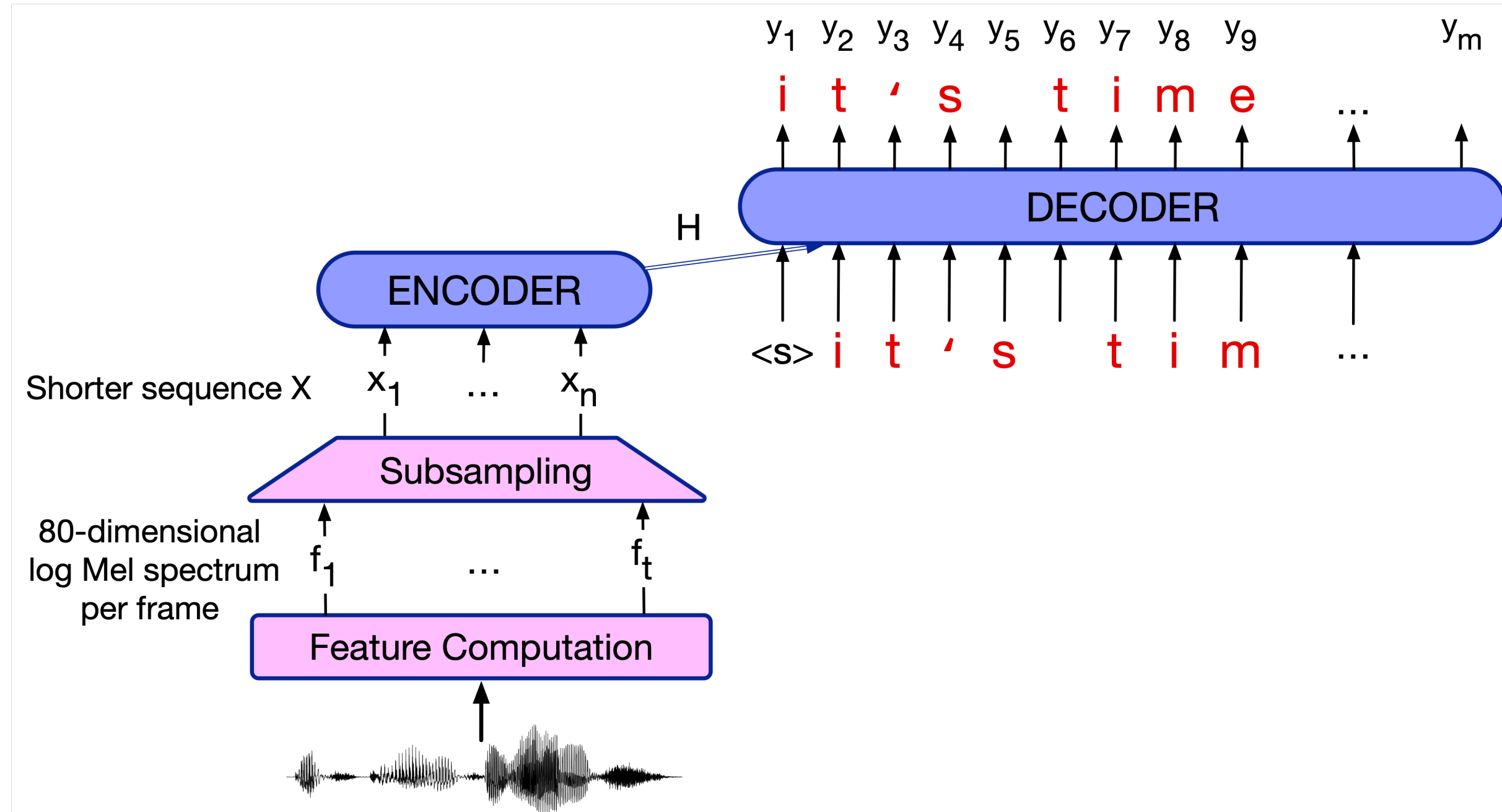
Encoder-decoder

Self-supervised + CTC

Speech encoder bolted onto LLM
backend

The Encoder-Decoder architecture for speech

Output: sequence of BPE tokens (or letters)

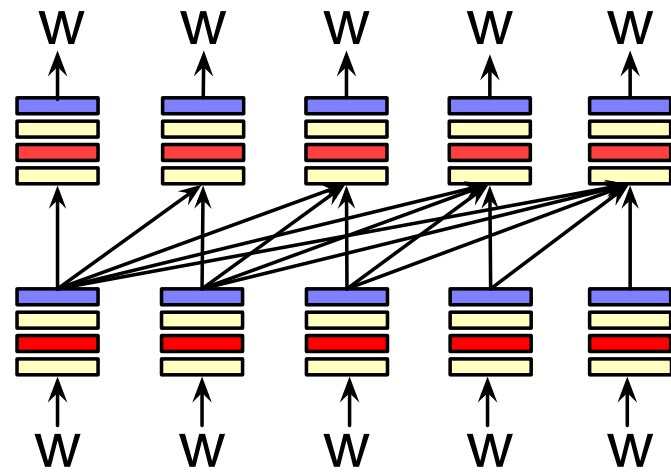


Input: sequence of acoustic vectors $X = x_1, x_2, x_3, \dots, x_t$ for each 10ms frames

Whisper

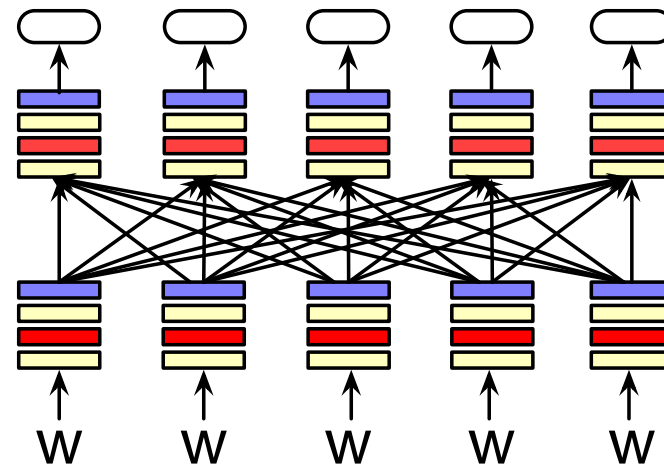
Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. "Robust speech recognition via large-scale weak supervision." *ICML 2022*

Three architectures for large language models



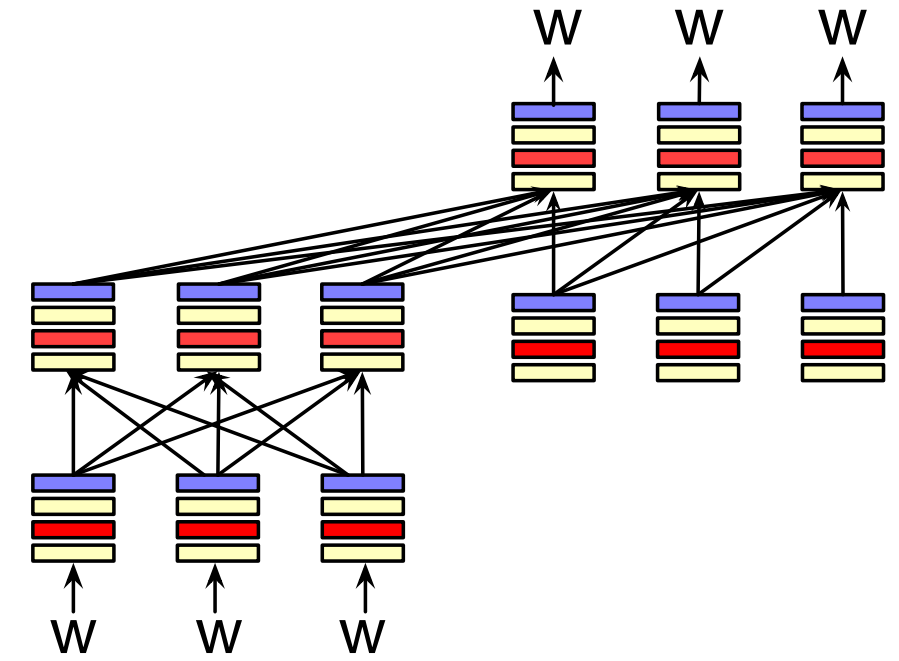
Decoders

GPT, Claude,
Llama
Mixtral



Encoders

BERT family,
HuBERT



Encoder-decoders

Flan-T5, Whisper

What is an encoder-decoder?

Enc-dec is useful when the input and output sequences are very different

- different in length
- different in vocabulary

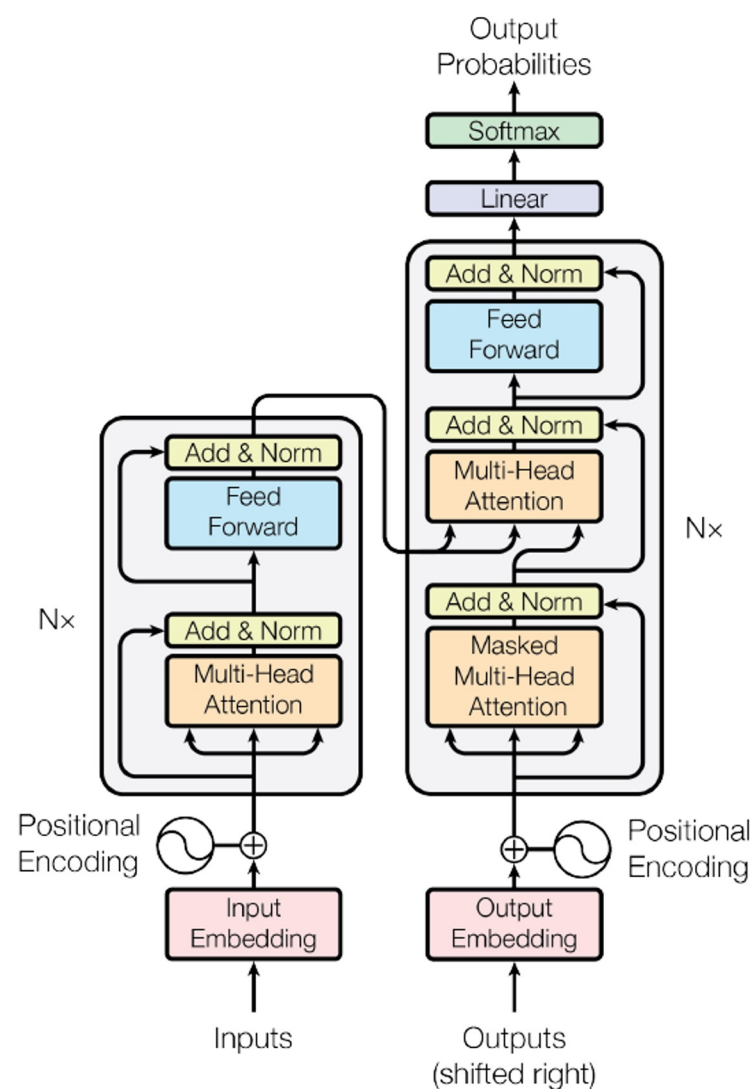
Written words are on average **1.3 tokens (BPE)**

Spoken words 250 milliseconds = **25 frames** of 10 ms

- So speech can be **19 times longer** ($=25/1.3$) than text

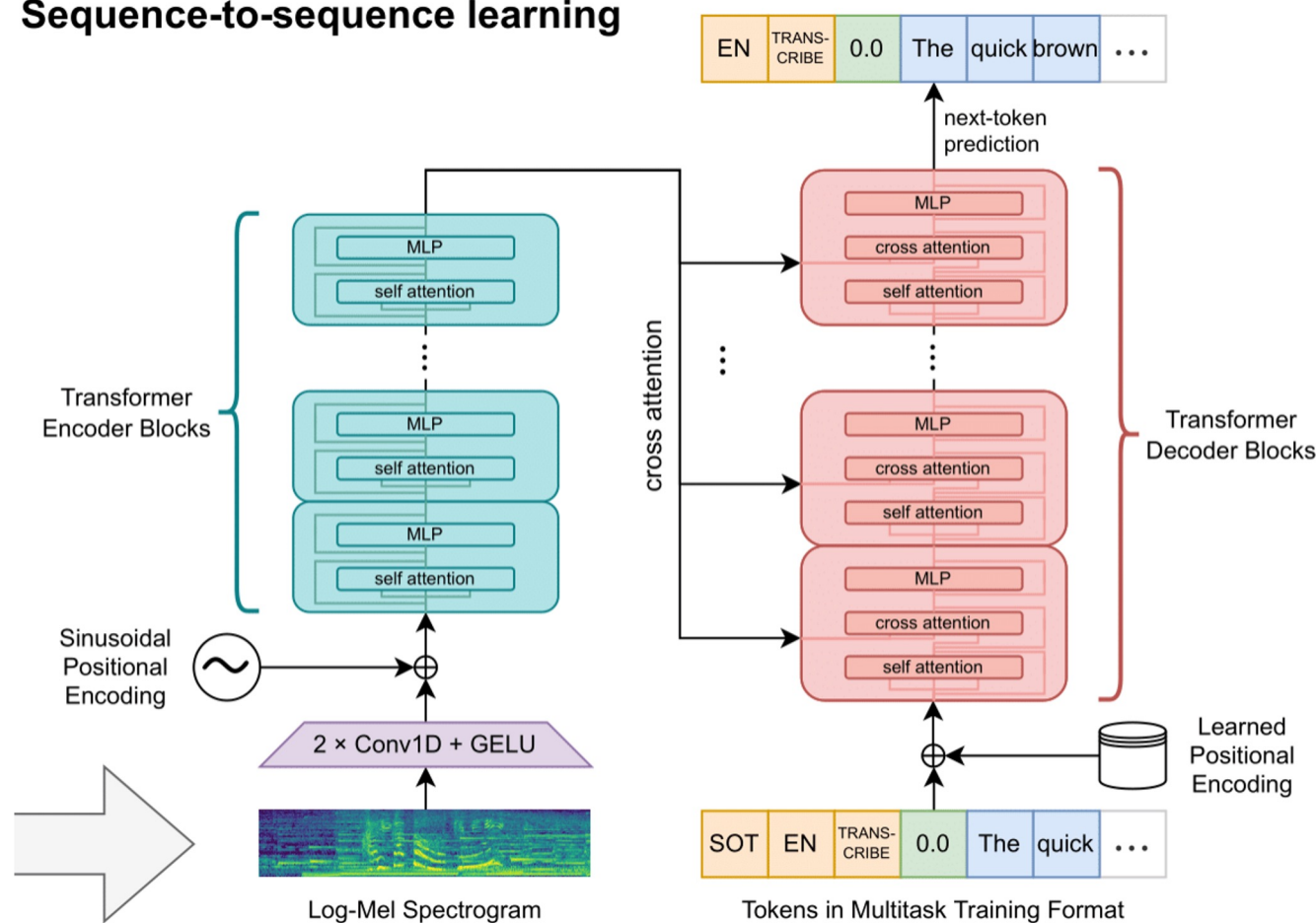
Whisper is a encoder-decoder transformer model

Text encoder-decoder transformer for translation

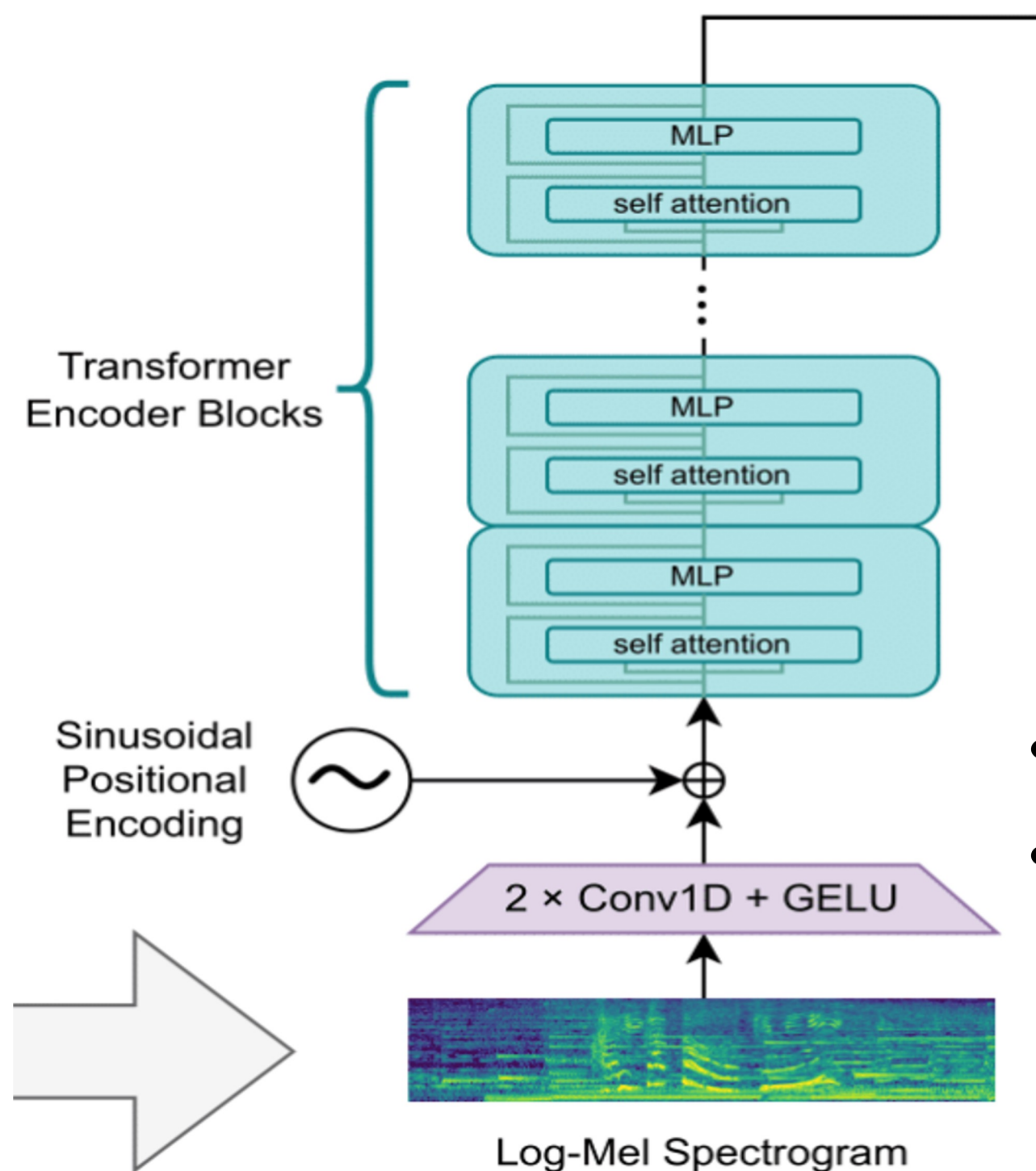


Whisper speech encoder-decoder transformer

Sequence-to-sequence learning



Whisper encoder



- Pass through transformer blocks
- Add position embeddings
- Pass through conv nets to extract features, downsample to every 20 ms
- Run a 128-channel Mel spectrogram, every 10 ms

Output of Whisper encoder

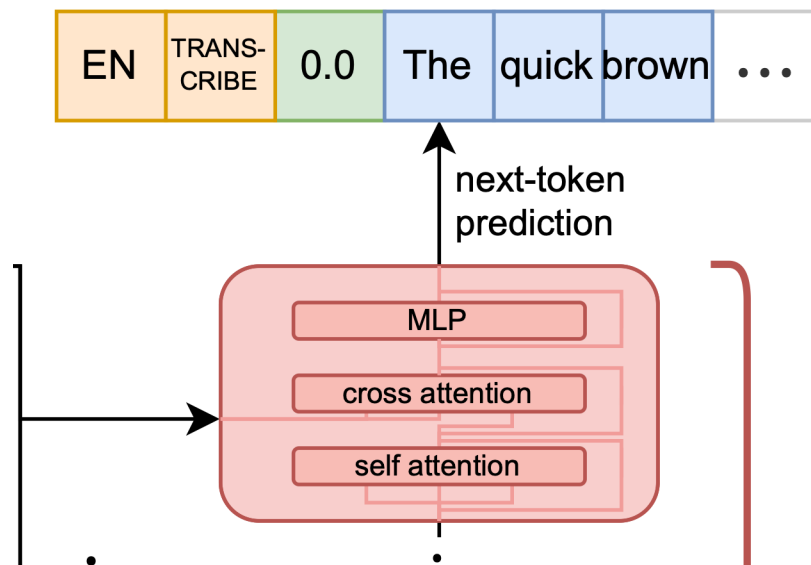
The language modeling head

The output of the last block is an embedding

We need to instead get a distribution over tokens.

We multiply by a matrix that turns an embedding $[1 \times 1280]$ into the vocabulary dimension $[1 \times 50,000]$

Then we do softmax over that vector to get probabilities



Speech
Models

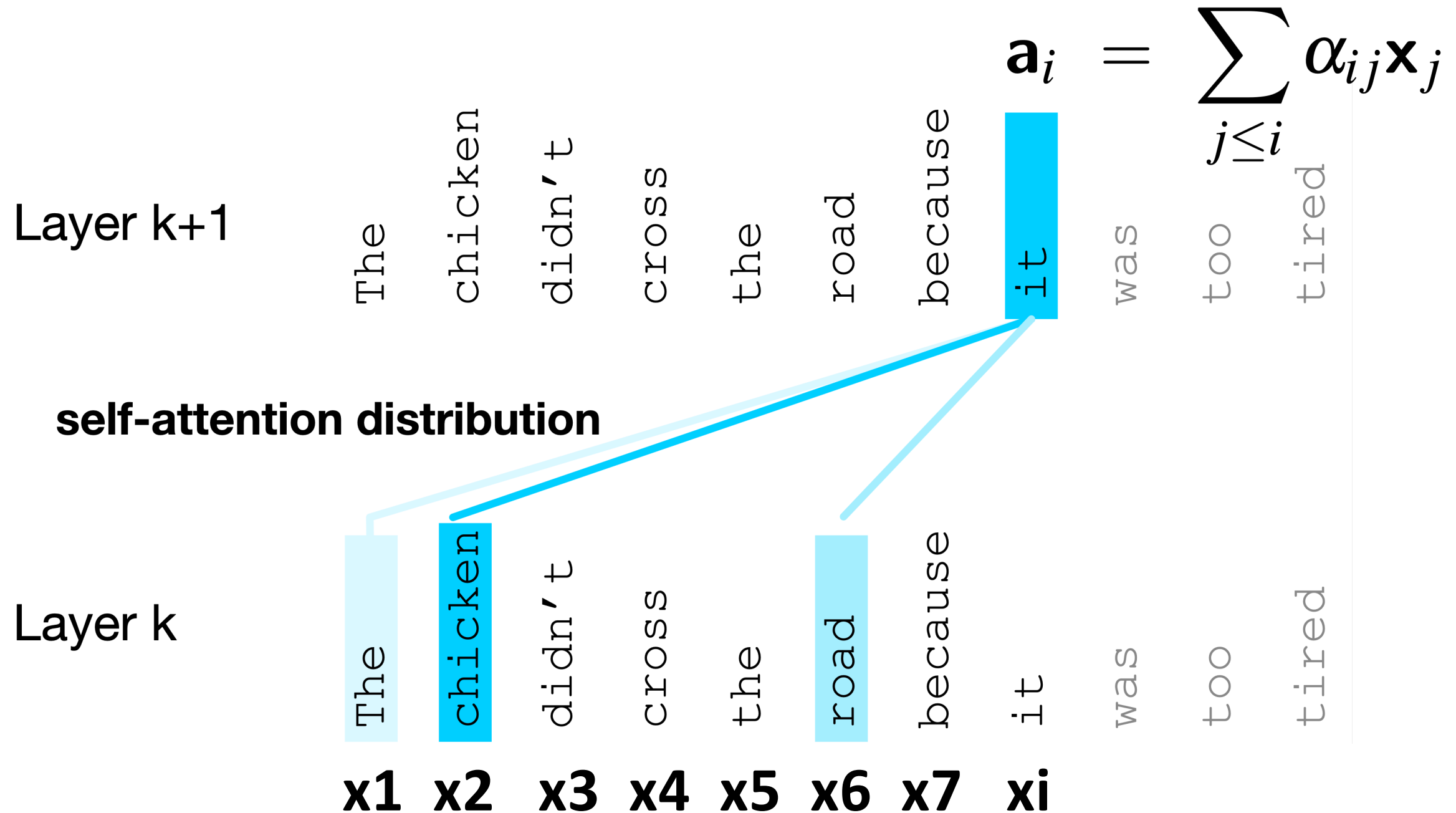
Whisper Introduction

Whisper: Cross-attention in the Encoder-Decoder

Reminder of attention

Attention:

$$\alpha_{ij} = \text{softmax}(\mathbf{x}_i \cdot \mathbf{x}_j) \quad \forall j \leq i$$

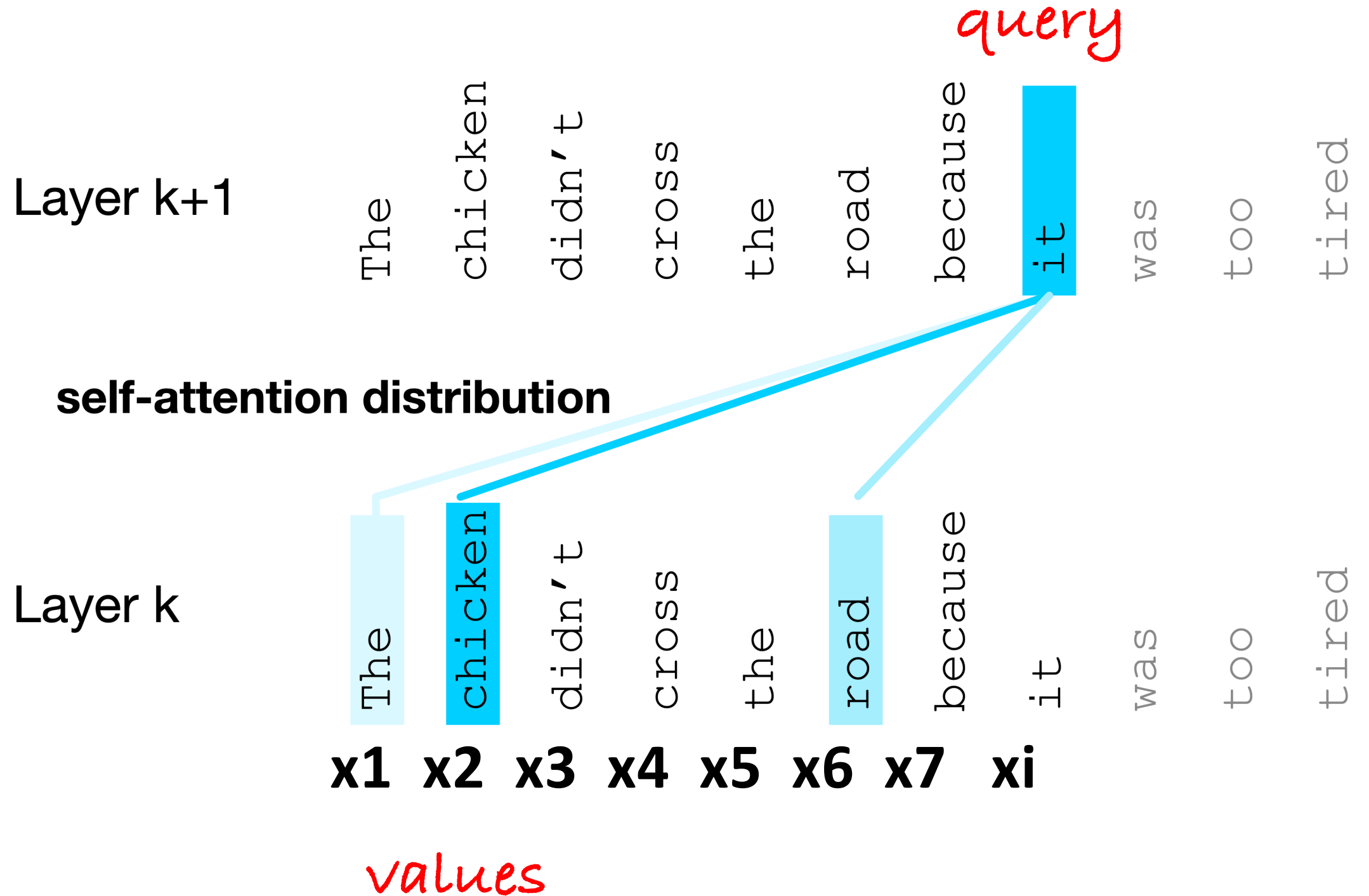


An Actual Attention Head: slightly more complicated

High-level idea: instead of using vectors (like x_i and x_4) directly, we'll represent 3 separate roles each vector $\mathbf{x}_i$ plays:

- **query**: *As the current element* being compared to the preceding inputs.
- **key**: *as a preceding input* that is being compared to the current element to determine a similarity
- **value**: a value of a preceding element that gets weighted and summed

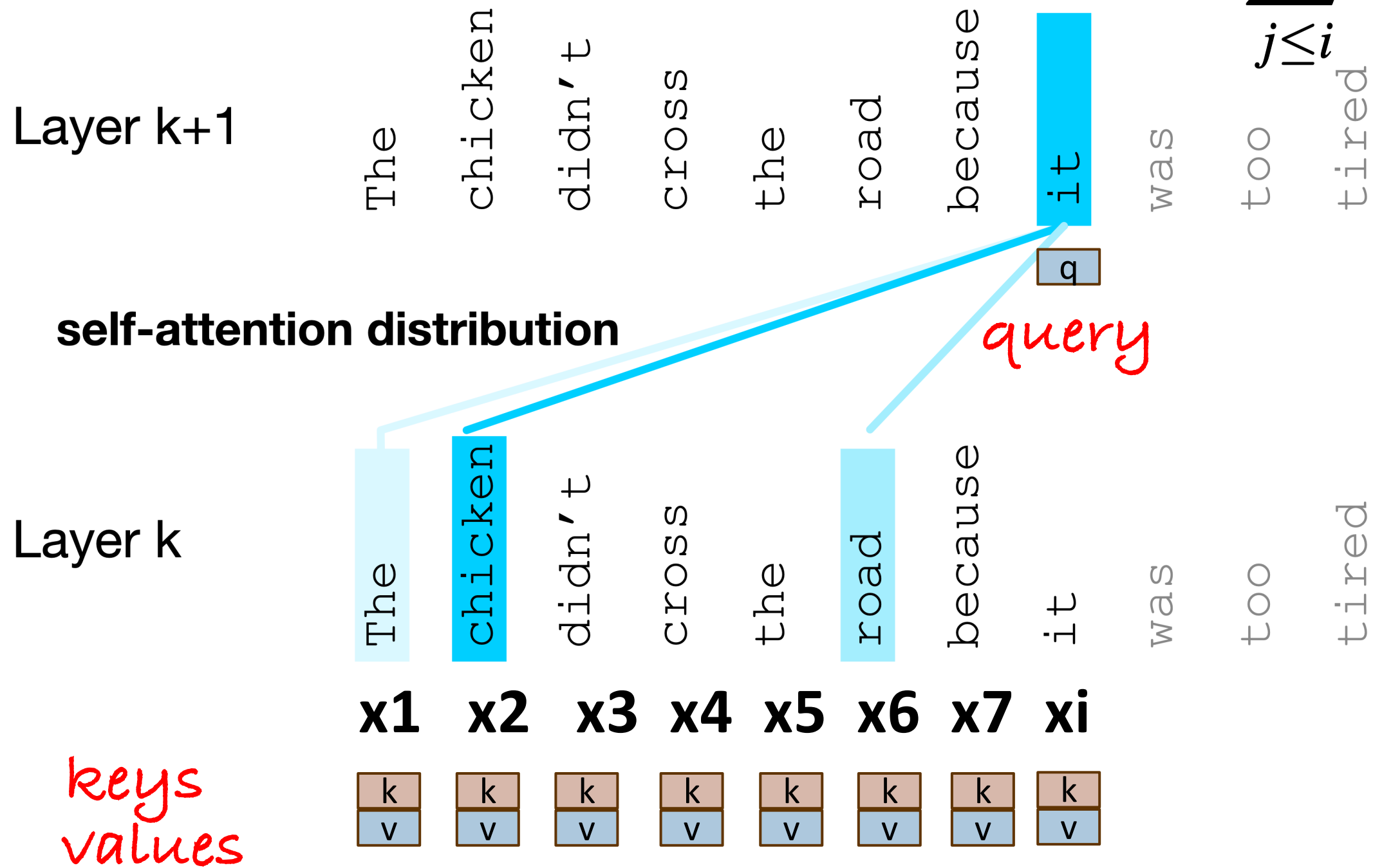
Attention intuition



Attention:

$$\alpha_{ij} = \text{softmax}(\alpha_{1i} \cdot \mathbf{x}_{j,j}) \quad \forall j \leq i$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{x}_j$$



How do we get 3 vectors $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$ for each $\mathbf{x}_i$?

We'll use matrices to project each vector $\mathbf{x}_i$ into a representation of its role as query, key, value:

- query: $\mathbf{W}^Q$
- key: $\mathbf{W}^K$
- value: $\mathbf{W}^V$

$$\mathbf{q}_i = \mathbf{x}_i \mathbf{W}^Q; \quad \mathbf{k}_i = \mathbf{x}_i \mathbf{W}^K; \quad \mathbf{v}_i = \mathbf{x}_i \mathbf{W}^V$$

Final equations for one attention head

we also normalize the dot product

Parallelizing computation using **X**

We can pack the N tokens of the input sequence into a single matrix **X** of size $[N \times d]$.

- Each **row** of **X** is the embedding of **one token** of the input.
- **X** can have 1K-32K rows, each of the dimensionality of the embedding d (the **model dimension**)

Attention:

$$\alpha_{ij} = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} \quad \forall j \leq i$$

$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j$$

$$\mathbf{Q} = \mathbf{XW}^Q; \quad \mathbf{K} = \mathbf{XW}^K; \quad \mathbf{V} = \mathbf{XW}^V$$

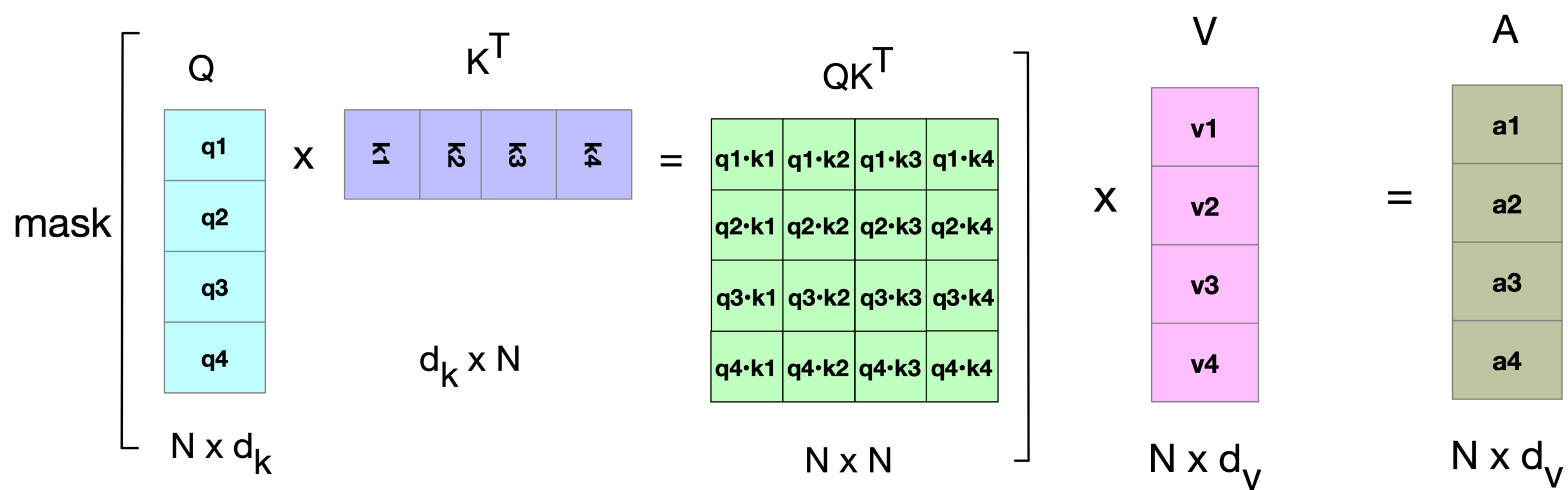
	X	V	K	Q
Thanks	x1	v1	k1	q1
for	x2	v2	k2	q2
all	x3	v3	k3	q3
the	x4	v4	k4	q4
fish	x5	v5	k5	q5

$$QK^T$$

We can do a single matrix multiply to combine Q and K^T

N	q1•k1	q1•k2	q1•k3	q1•k4
	q2•k1	q2•k2	q2•k3	q2•k4
	q3•k1	q3•k2	q3•k3	q3•k4
	q4•k1	q4•k2	q4•k3	q4•k4
N				

$$\begin{array}{c}
 \text{mask} \left[\begin{array}{c} \text{Q} \\ \begin{array}{|c|} \hline \text{q1} \\ \hline \text{q2} \\ \hline \text{q3} \\ \hline \text{q4} \\ \hline \end{array} \\ N \times d_k \end{array} \right] \times \begin{array}{c} K^T \\ \begin{array}{|c|c|c|c|} \hline k1 & k2 & k3 & k4 \\ \hline \end{array} \\ d_k \times N \end{array} = \begin{array}{c} QK^T \\ \begin{array}{|c|c|c|c|} \hline \text{q1} \cdot \text{k1} & \text{q1} \cdot \text{k2} & \text{q1} \cdot \text{k3} & \text{q1} \cdot \text{k4} \\ \hline \text{q2} \cdot \text{k1} & \text{q2} \cdot \text{k2} & \text{q2} \cdot \text{k3} & \text{q2} \cdot \text{k4} \\ \hline \text{q3} \cdot \text{k1} & \text{q3} \cdot \text{k2} & \text{q3} \cdot \text{k3} & \text{q3} \cdot \text{k4} \\ \hline \text{q4} \cdot \text{k1} & \text{q4} \cdot \text{k2} & \text{q4} \cdot \text{k3} & \text{q4} \cdot \text{k4} \\ \hline \end{array} \\ N \times N \end{array} \right]
 \end{array}$$



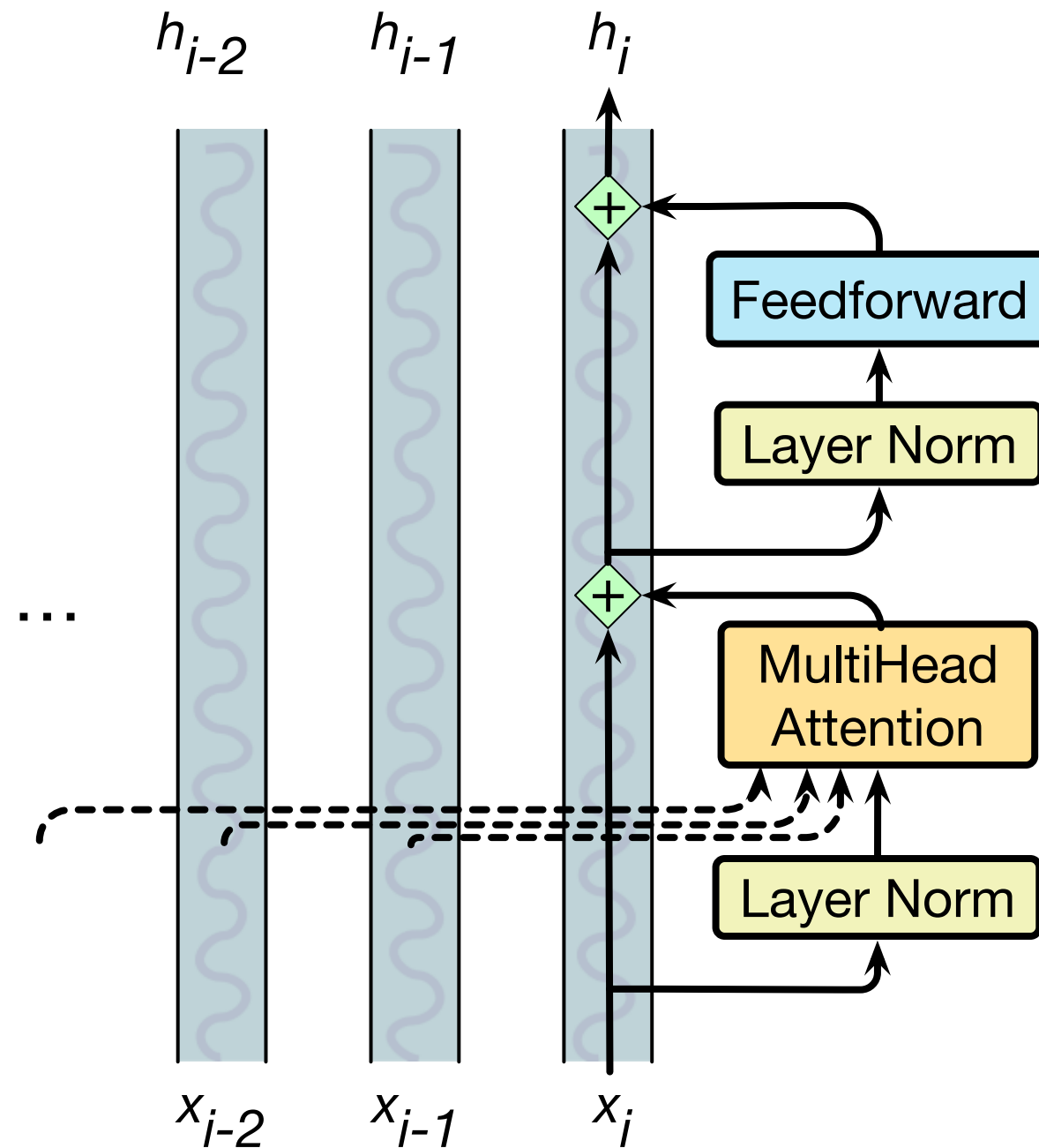
$$\mathbf{a}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{v}_j$$

Parallelizing attention

- Scale the scores, take the softmax, and then multiply the result by V resulting in a matrix of shape $N \times d$
 - An attention vector for each input token

$$\mathbf{A} = \text{softmax} \left(\text{mask} \left(\frac{\mathbf{QK}^\top}{\sqrt{d_k}} \right) \right) \mathbf{V}$$

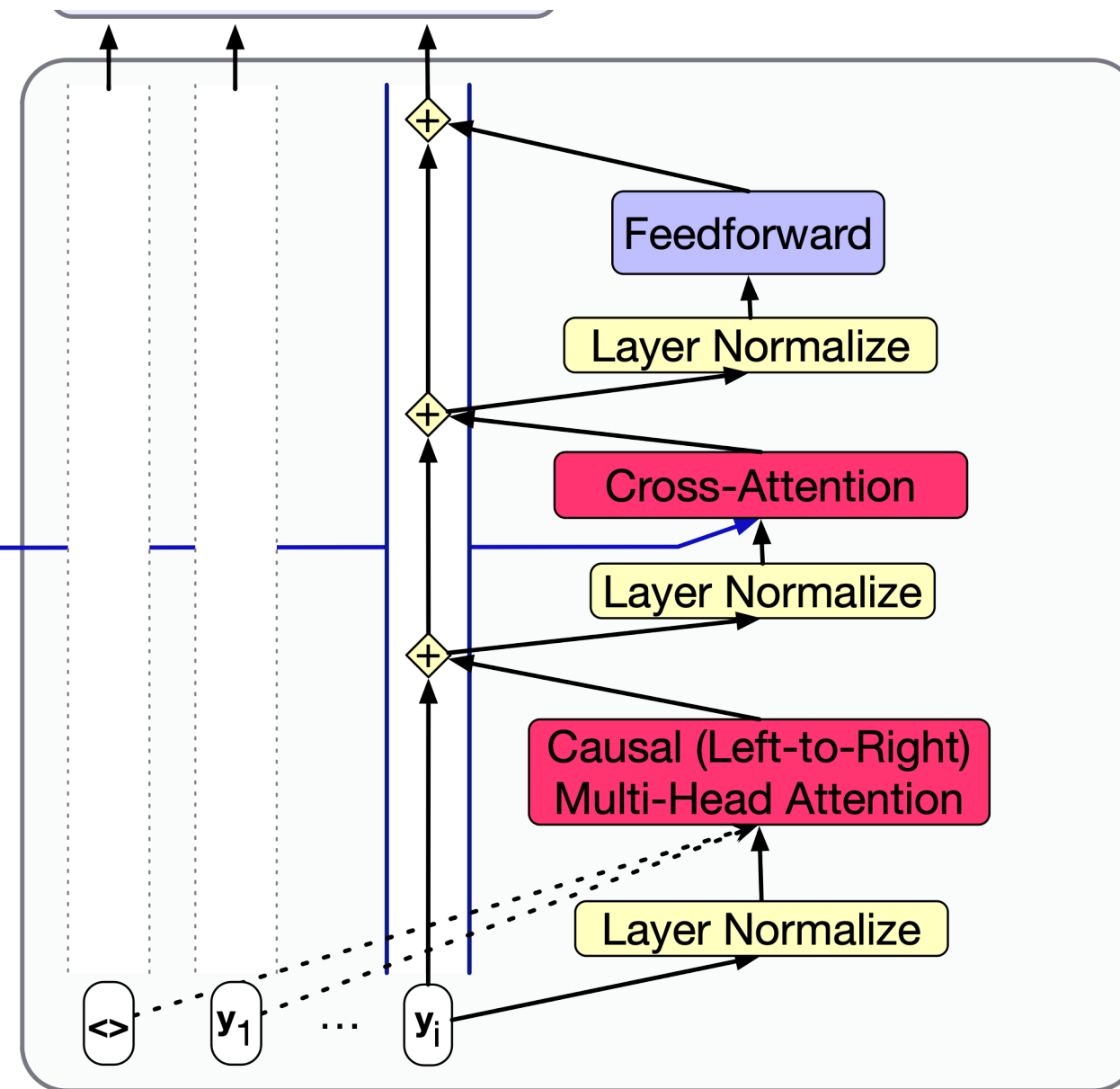
A regular transformer block



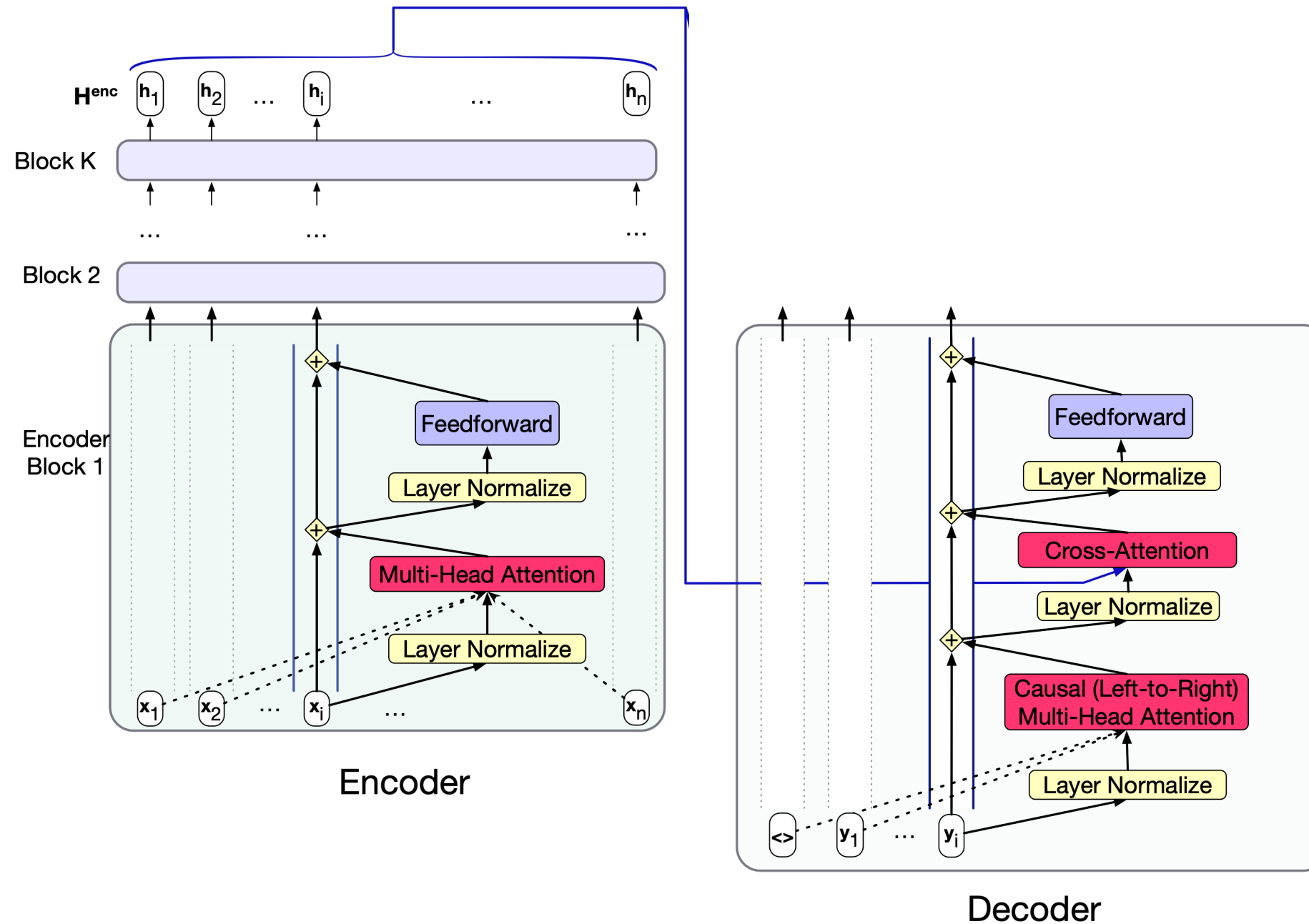
Reminder: Whisper

A encoder-decoder decoder-side block with cross-attention

Cross-attend
to the speech
encoder output



Whisper encoder/decoder components



Cross-attention

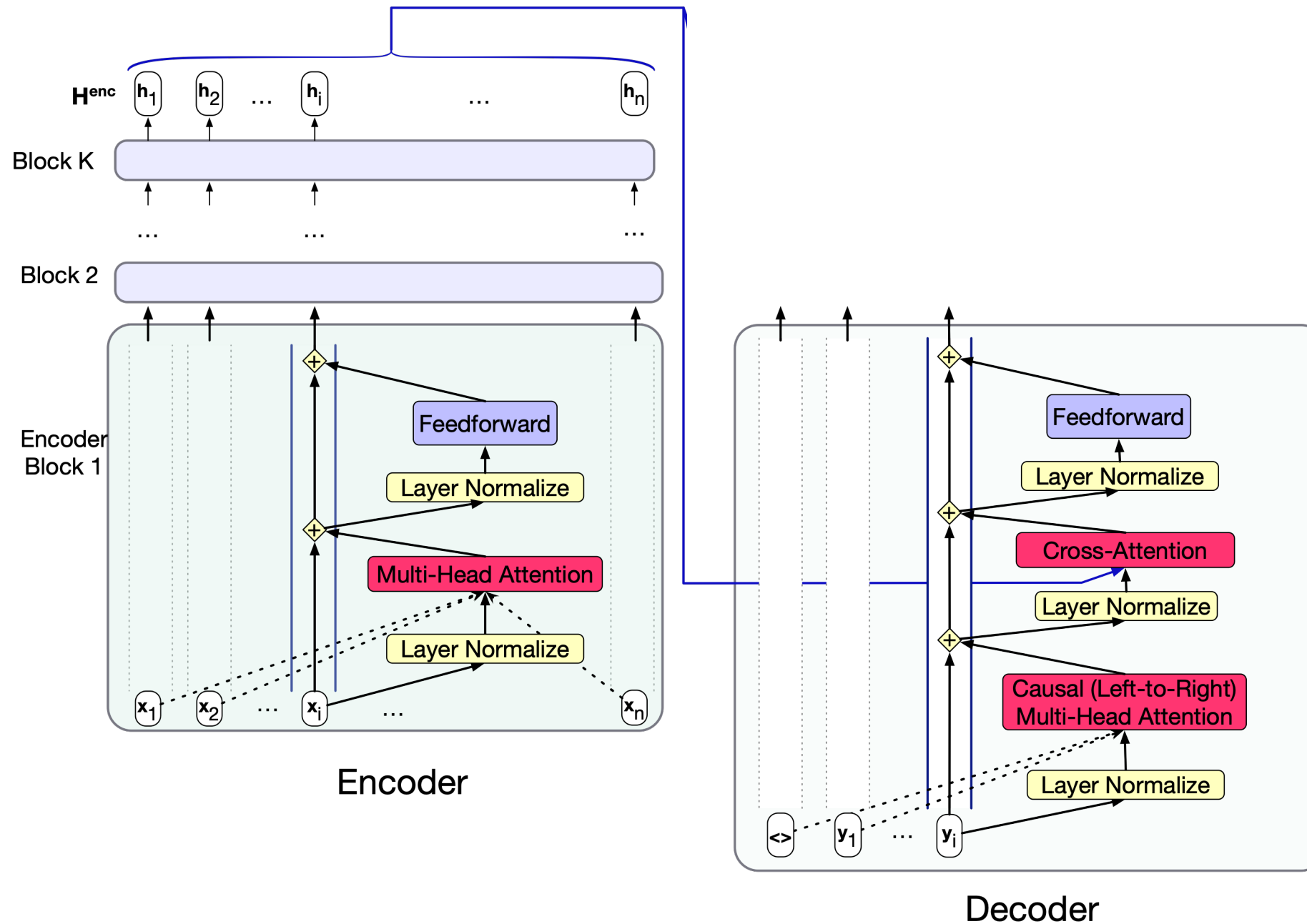
Attention in a decoder (or encoder) builds up a new representation at layer l from layer $l-1$

- By attending to all the **other tokens in $l-1$**

Cross-attention in a decoder builds up a new representation at layer l from layer $l-1$

- By attending to the **encoder output + other tokens in $l-1$**

Whisper encoder/decoder components



Cross-attention equations

The output of encoder is $\mathbf{H}_{enc} = \mathbf{h}_1, \dots, \mathbf{h}_n$ of shape $[n \times d]$

Cross attention takes the **query** from the decoder

$$\mathbf{Q} = \mathbf{H}^{dec[\ell-1]} \mathbf{W}^Q;$$

But the **key** and **value** from the encoder:

$$\mathbf{K} = \mathbf{H}^{enc} \mathbf{W}^K; \quad \mathbf{V} = \mathbf{H}^{enc} \mathbf{W}^V$$

So "attend to the speech input informed by the text representation from my previous layer"

Multiplies them all together to get the attention value

$$\text{CrossAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{QK}^T}{\sqrt{d_k}} \right) \mathbf{V}$$

Whisper: Cross-attention in the Encoder-Decoder

Speech
Models

Whisper: Training

Whisper training: Loss function

Standard language-modeling loss:

$$L_{CE} = -\log p(y_i | y_1, \dots, y_{i-1}, \mathbf{X})$$

The probability of the next token y_i given all the previous tokens and the acoustic input $\mathbf{X}$

Whisper training: Data

680,000 hours of multilingual and multitask supervision

- Including 118,000 hours from 96 other languages.

All the data is audio paired with a transcript

Scraped from the web, with lots of filtering

Broken into chunks:

- 30 second audio, paired with transcript of words

Actually, it's multitask training setup

Multitask training data (680k hours)

English transcription



"Ask not what your country can do for ..."



Ask not what your country can do for ...

Any-to-English speech translation



"El rápido zorro marrón salta sobre ..."



The quick brown fox jumps over ...

Non-English transcription



"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ..."



언덕 위에 올라 내려다보면 너무나 넓고 넓은 ...

No speech

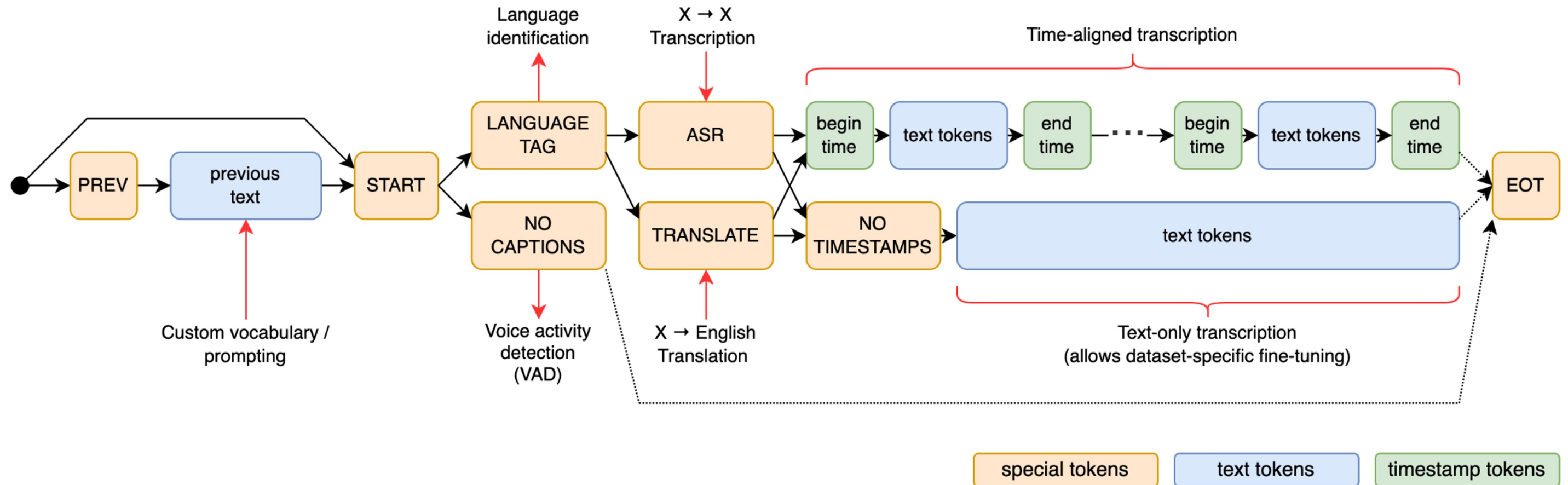


(background music playing)

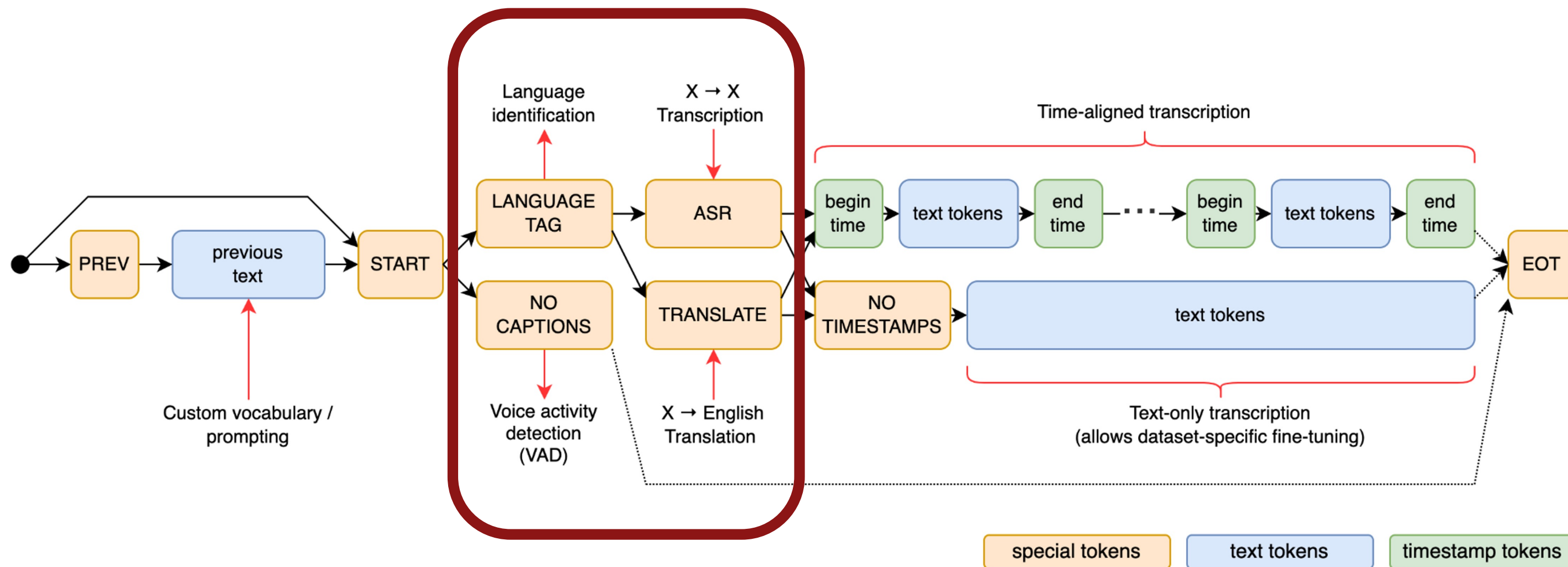


∅

Whisper (Multitask Training)



Whisper: task tags



Speech
Models

Whisper: Training

Speech
Models

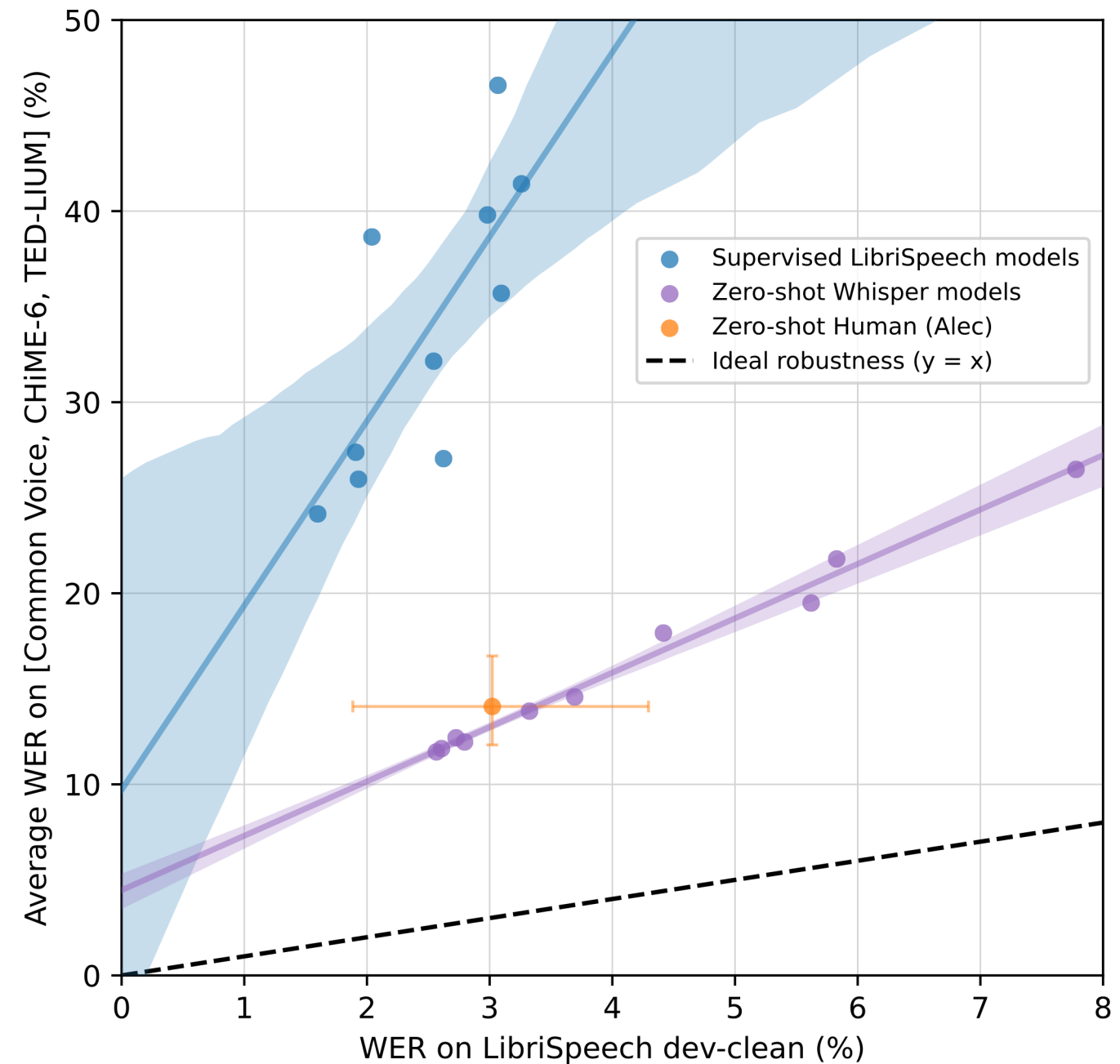
Whisper: Conclusion

Speech recognition!

- Transformers work for speech as well as text!
- Speech transcription performance is good enough to be useful in many practical situations
- Huge improvements in technology in 10 years.

So is speech recognition solved?

Not yet. Whisper is much worse than humans on slightly noisy speech



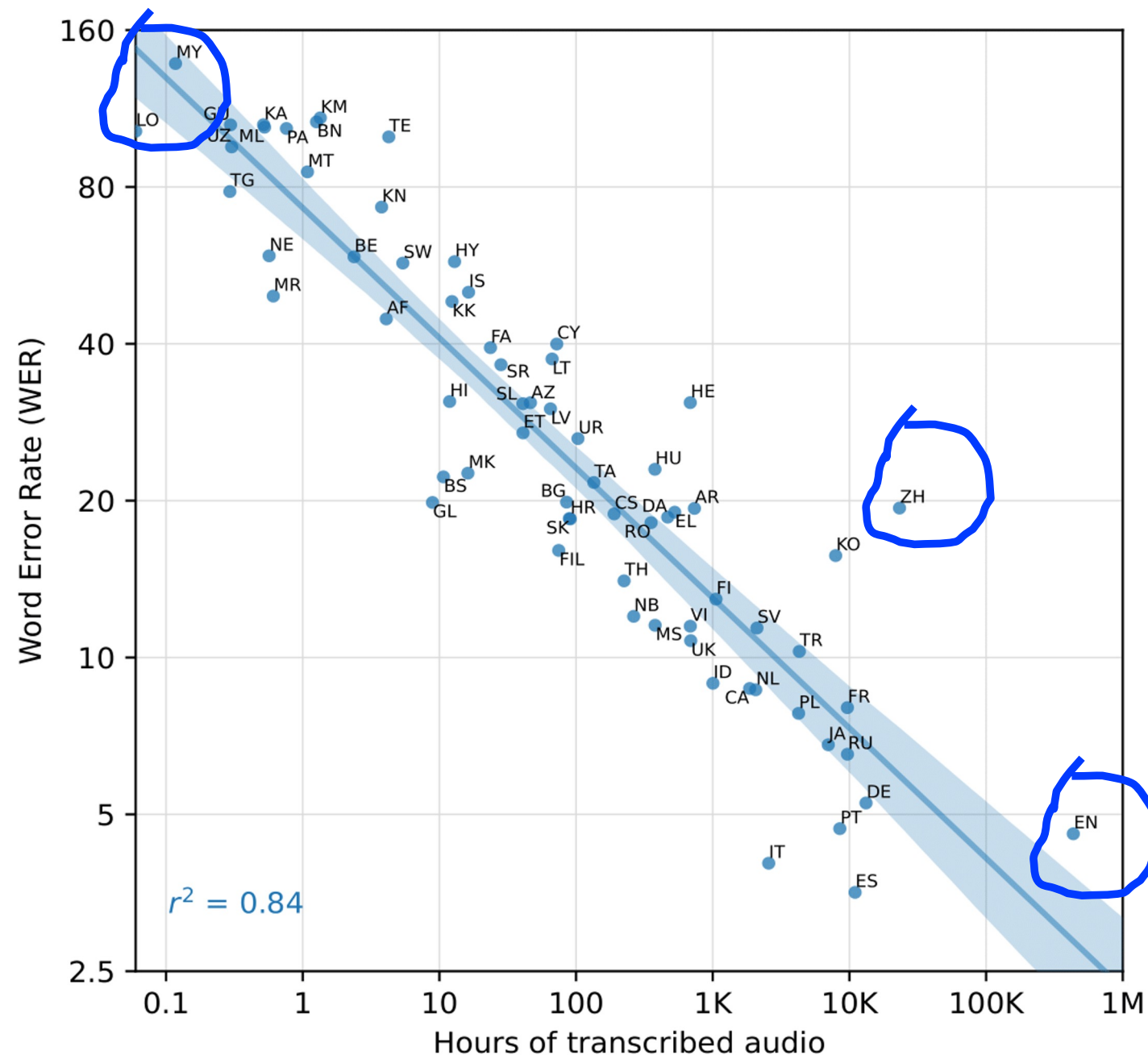
And even worse on very noisy speech

Ground truth: "Allez, allez, allez, alleeeeeez..... Martinez! Qui vient de sauver.... Face à Randal Kolo Muani. Mais c'est fou ! Et ça repart de l'autre côté, et **ça repart de l'autre côté !**

[illegible][illegible]

And only works on languages with enough data

Transcription performance on FLEURS dataset
(lower is better)



WER below 10 if 1000 hours of training data (English, Spanish, French, German, etc.)

Low resource languages (top left) have a WER above 100 (Burmese, Lao, Tajik)

High WER for Chinese

Speech recognition!

- ASR performance good enough to be useful in many practical situations
- Huge improvements over last 10 years.
- But still fails on many tasks and domains
 - Noisy speech
 - Multiple overlapping speakers
 - Low-resource languages
 - Some speakers (children, old people, people who stutter)

Speech Models

Whisper