Speech synthesis
Where did the signal processing go?

Simon King, Centre for Speech Technology Research, University of Edinburgh, UK

CSTR website: `www.cstr.ed.ac.uk`

Teaching website: `speech.zone`

Motivation

arXiv:1609.03499 (unreviewed manuscript)

WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

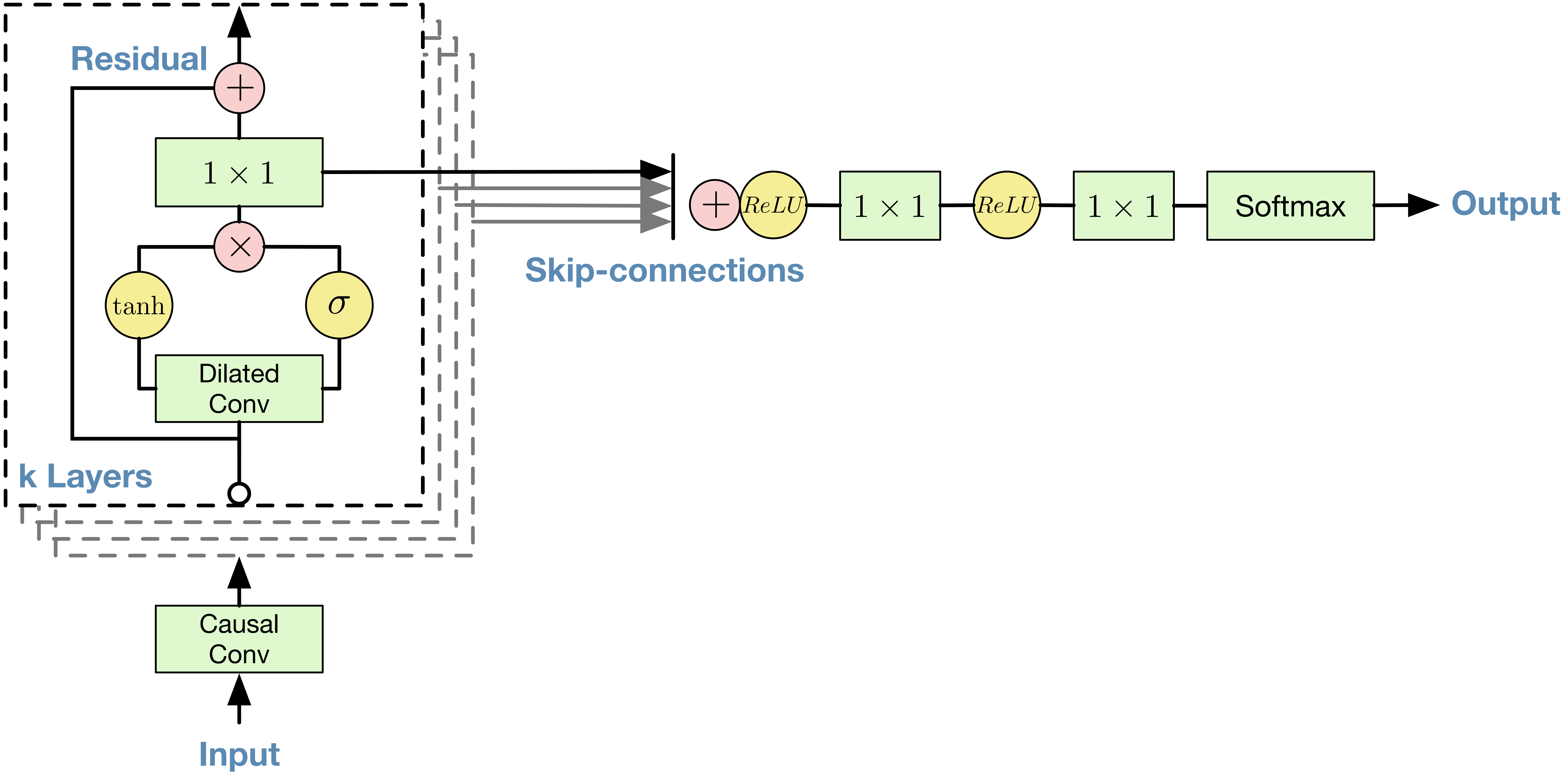
Karen Simonyan **Oriol Vinyals** **Alex Graves**

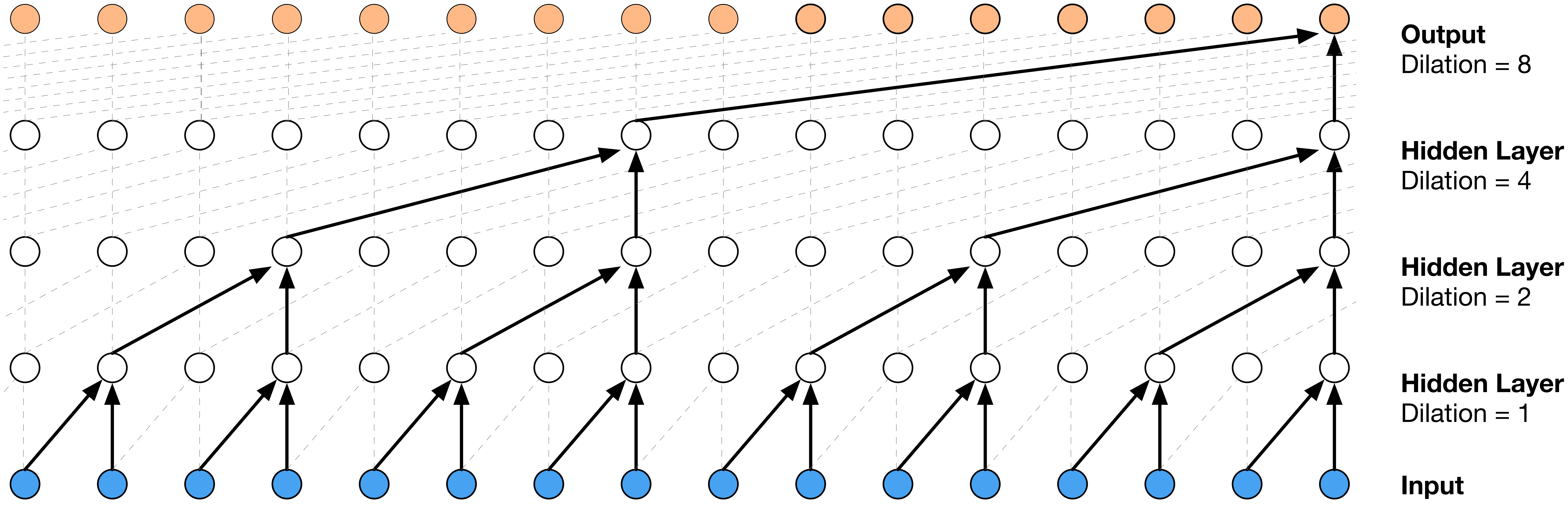
Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK

19 Sep 2016

Abstract


Residual
1×1
tanh
σ
Dilated Conv
k Layers
Causal Conv
Input
Skip-connections
$ReLU$
1×1
$ReLU$
1×1
Softmax
Output


Output
Dilation = 8
Hidden Layer
Dilation = 4
Hidden Layer
Dilation = 2
Hidden Layer
Dilation = 1
Input

DOI: 10.21437/Interspeech.2017-1452

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Tacotron: Towards End-to-End Speech Synthesis

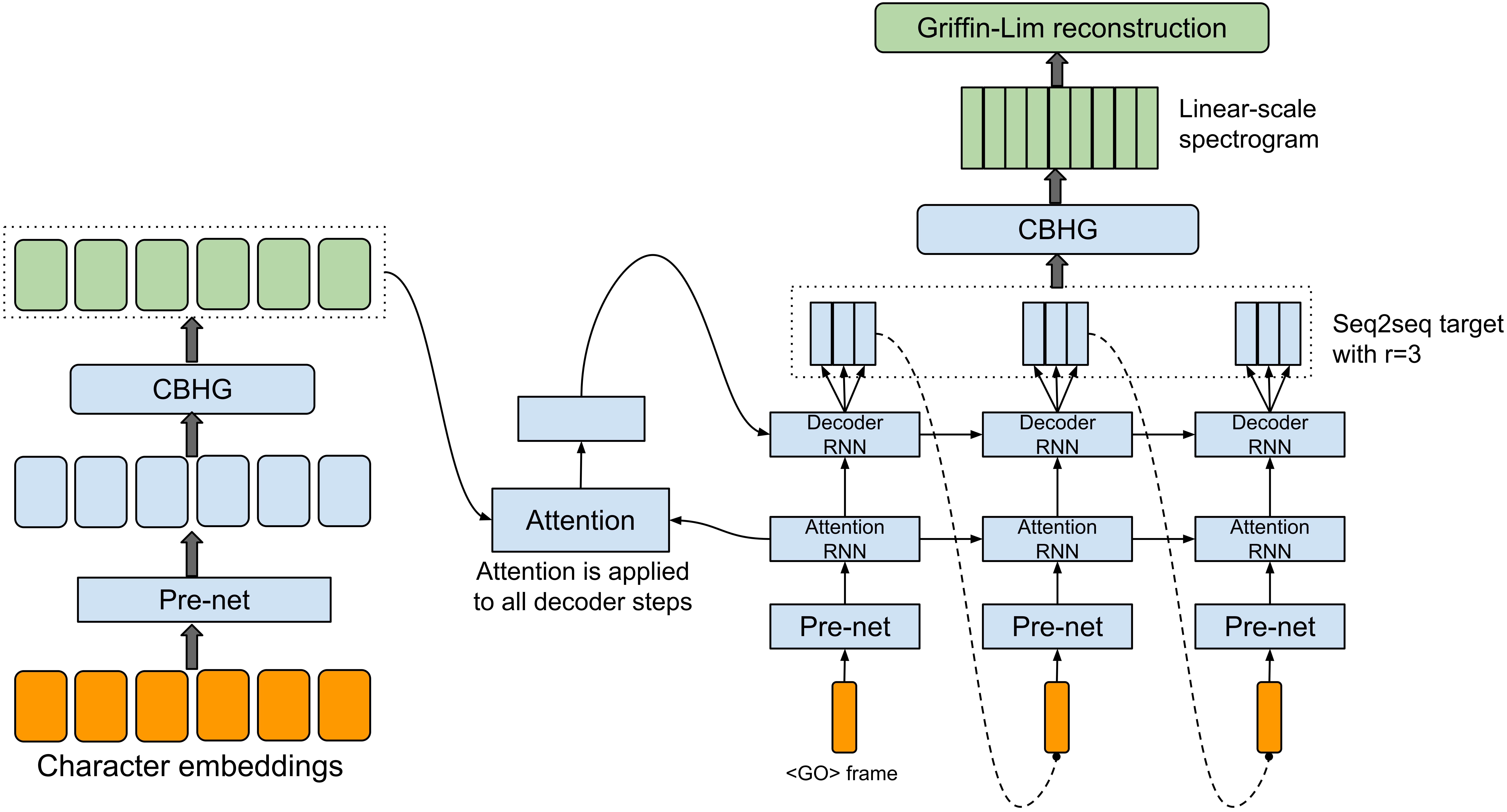
Yuxuan Wang, RJ Skerry-Ryan*, Daisy Stanton, Yonghui Wu, Ron J. Weiss†,*
Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio†, Quoc Le,*
*Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous**

Google, Inc.
{yxwang, rjryan, rif}@google.com

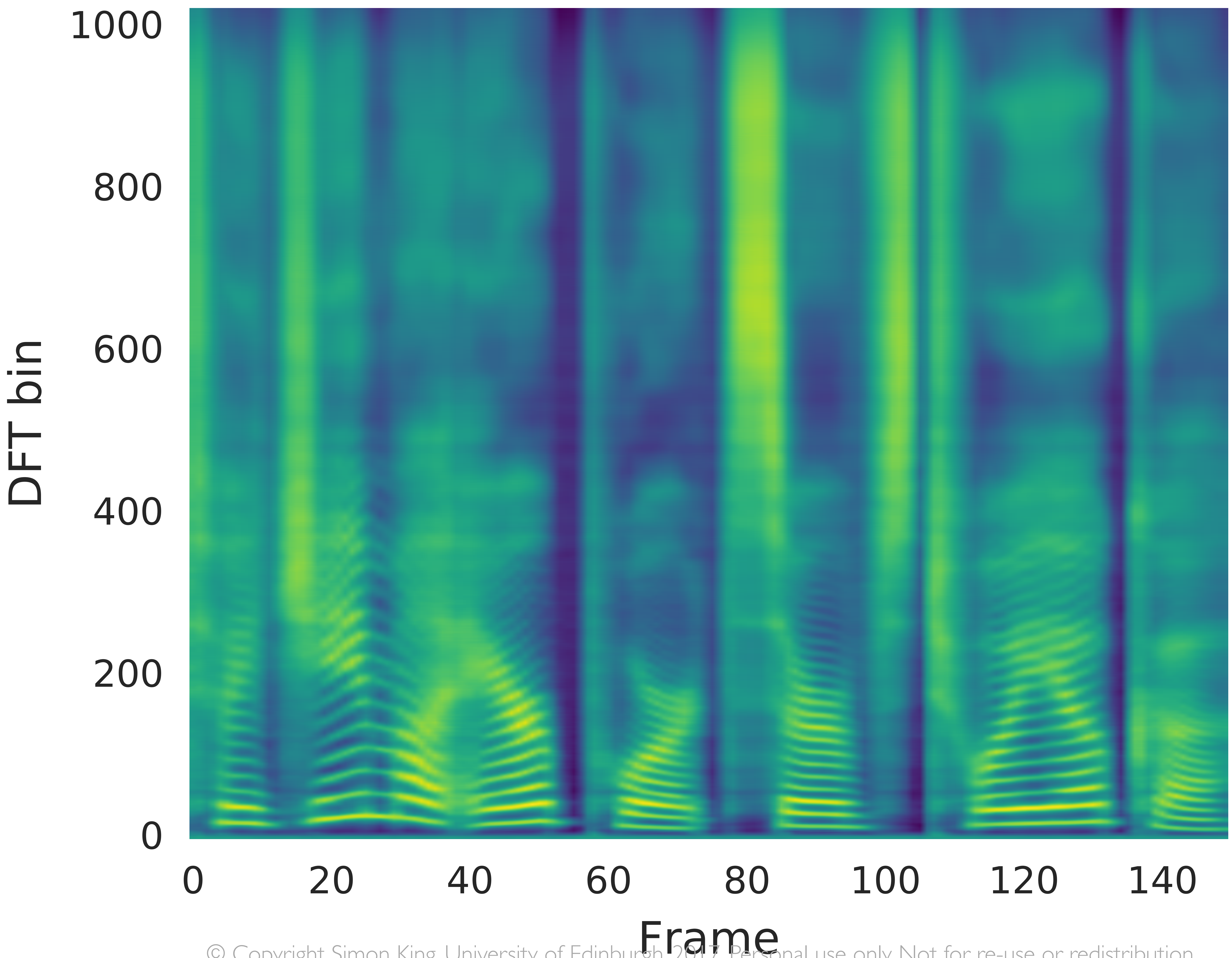
Abstract

A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module. Building these components often requires extensive domain expertise and may contain brittle

this is a particularly difficult learning task for an end-to-end model: it must cope with large variations at the signal level for a given input. Moreover, unlike end-to-end speech recognition [4] or machine translation [5], TTS outputs are continuous, and output sequences are usually much longer than those of the input. These attributes cause prediction errors to accu-



Griffin-Lim reconstruction
Linear-scale spectrogram
CBHG
Seq2seq target with r=3
Decoder RNN
Decoder RNN
Decoder RNN
Attention RNN
Attention RNN
Attention RNN
Pre-net
Pre-net
Pre-net
<GO> frame
Attention
Attention is applied to all decoder steps
CBHG
Pre-net
Character embeddings

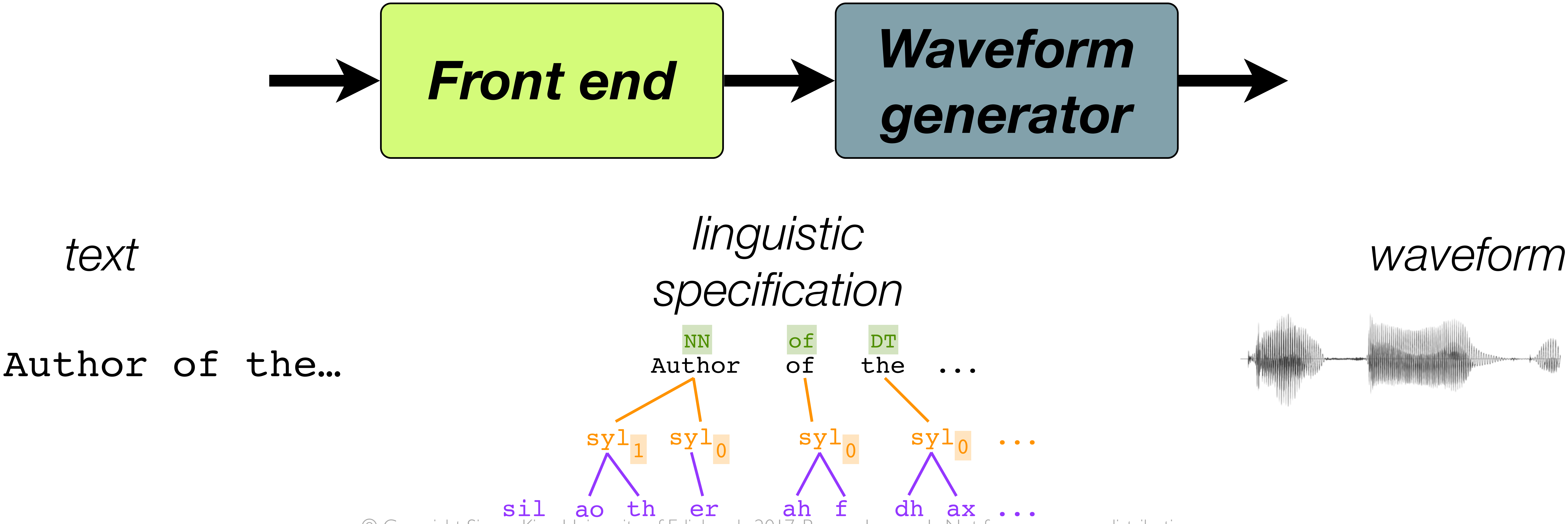

1000
800
600
400
200
0
DFT bin
0
20
40
60
80
100
120
140
Frame

Contents

1. Mini-tutorial
2. Conventional signal processing for speech synthesis
3. What do we want from our speech signal representation (a lot !)

Part 1 - Mini-tutorial - Text-to-speech using Deep Neural Networks

The classic two-stage pipeline of unit selection

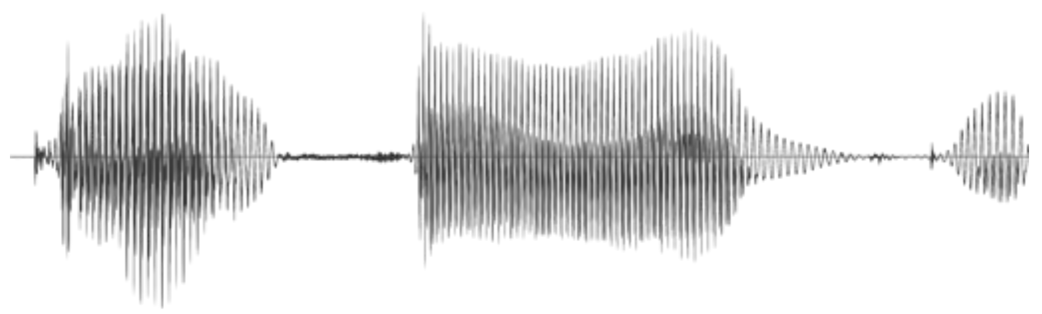


The end-to-end problem we want to solve

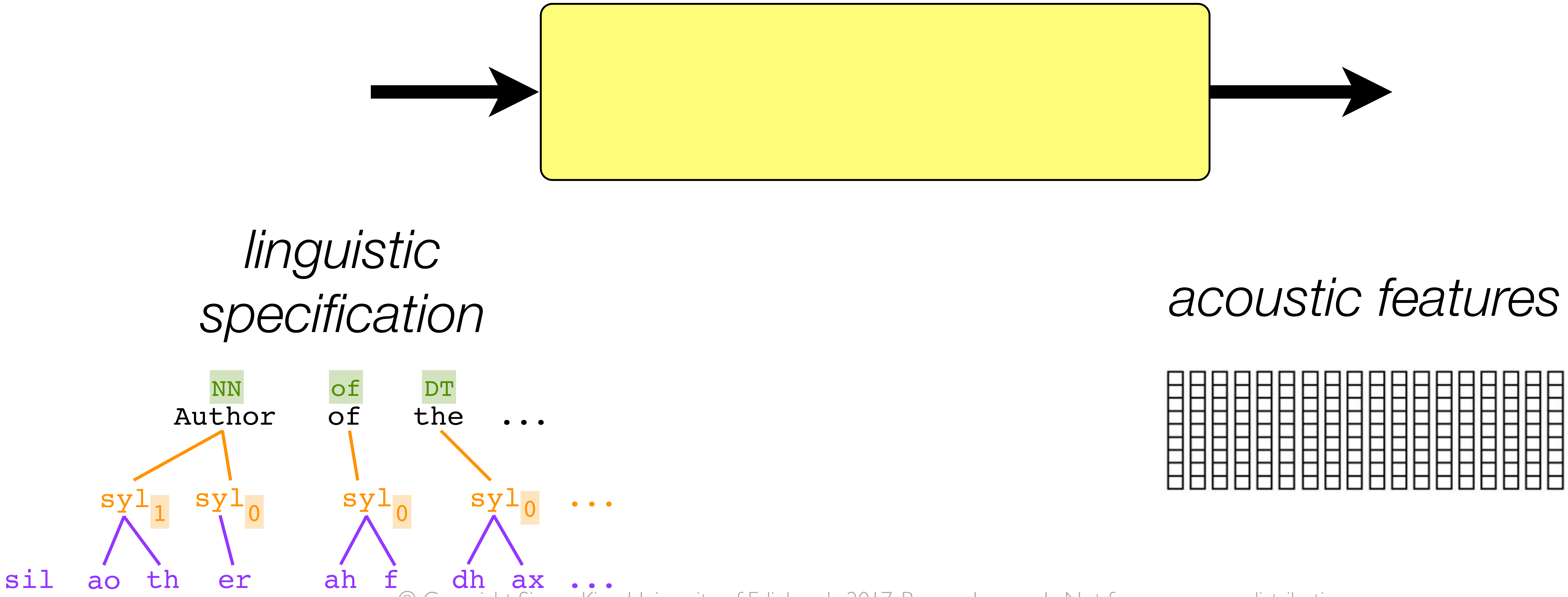
text

waveform

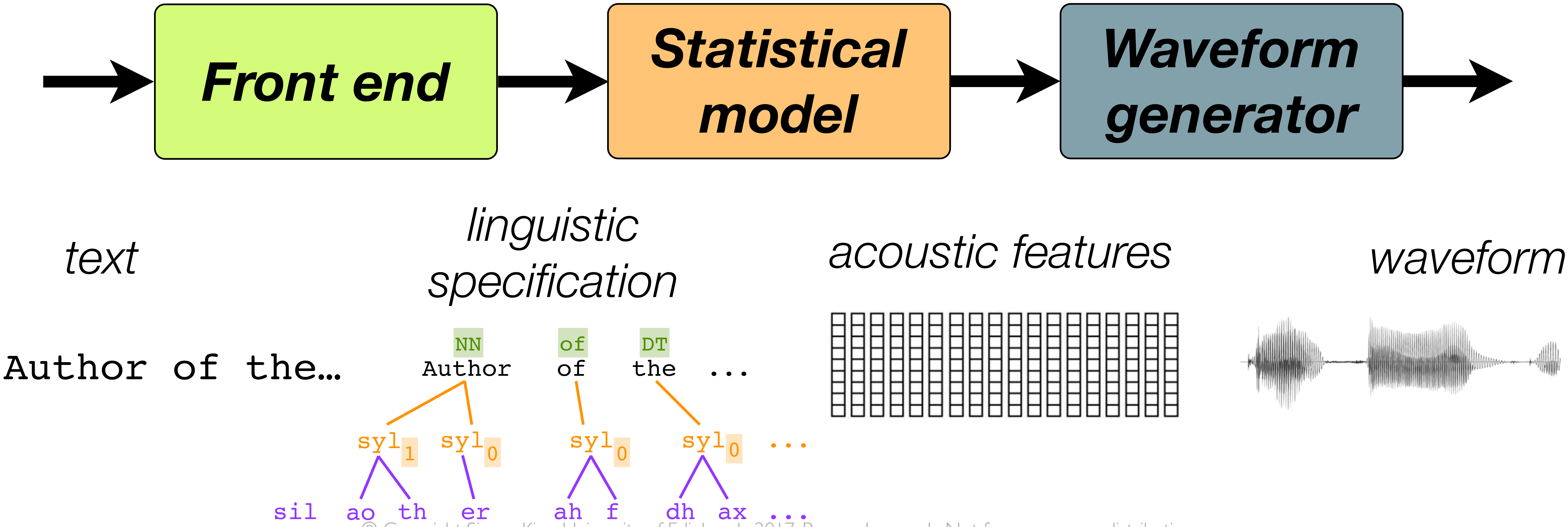
`Author of the…`



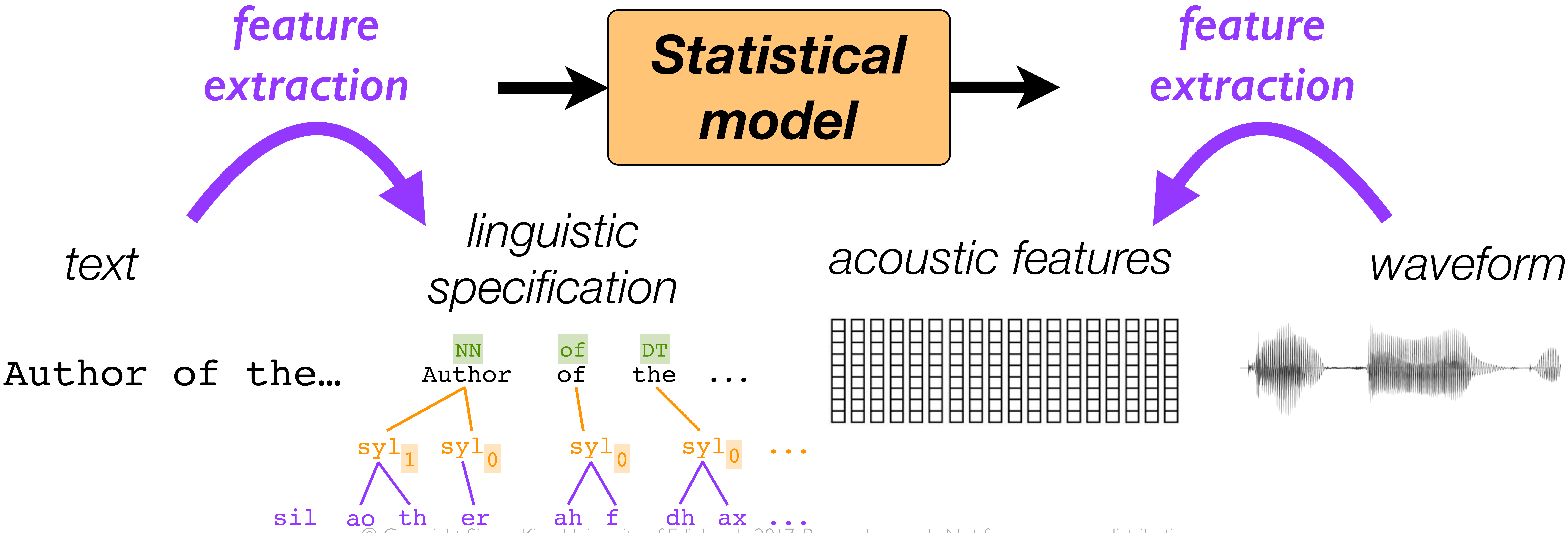
A problem we can actually solve with machine learning



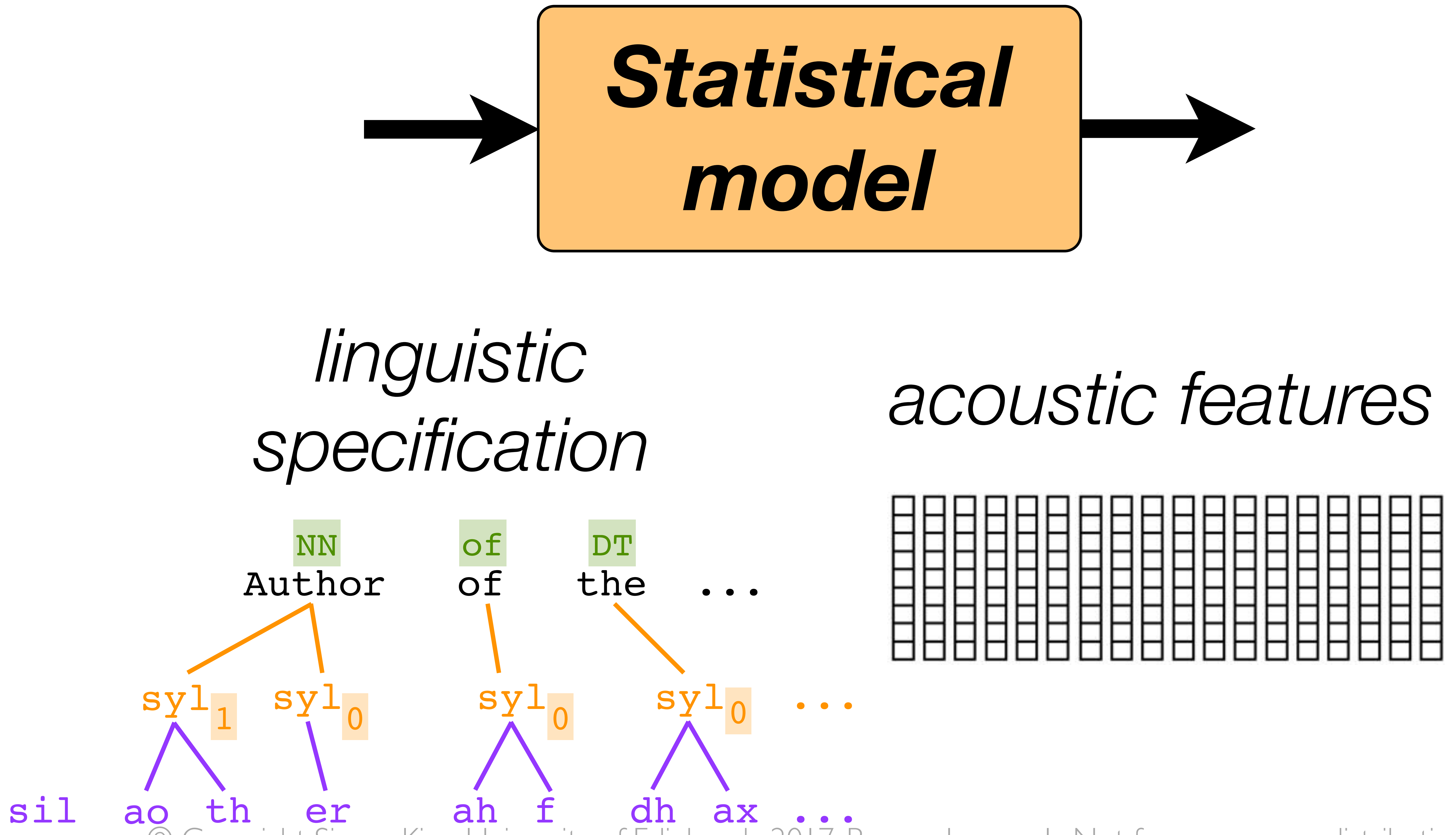
The classic three-stage pipeline of statistical parametric speech synthesis



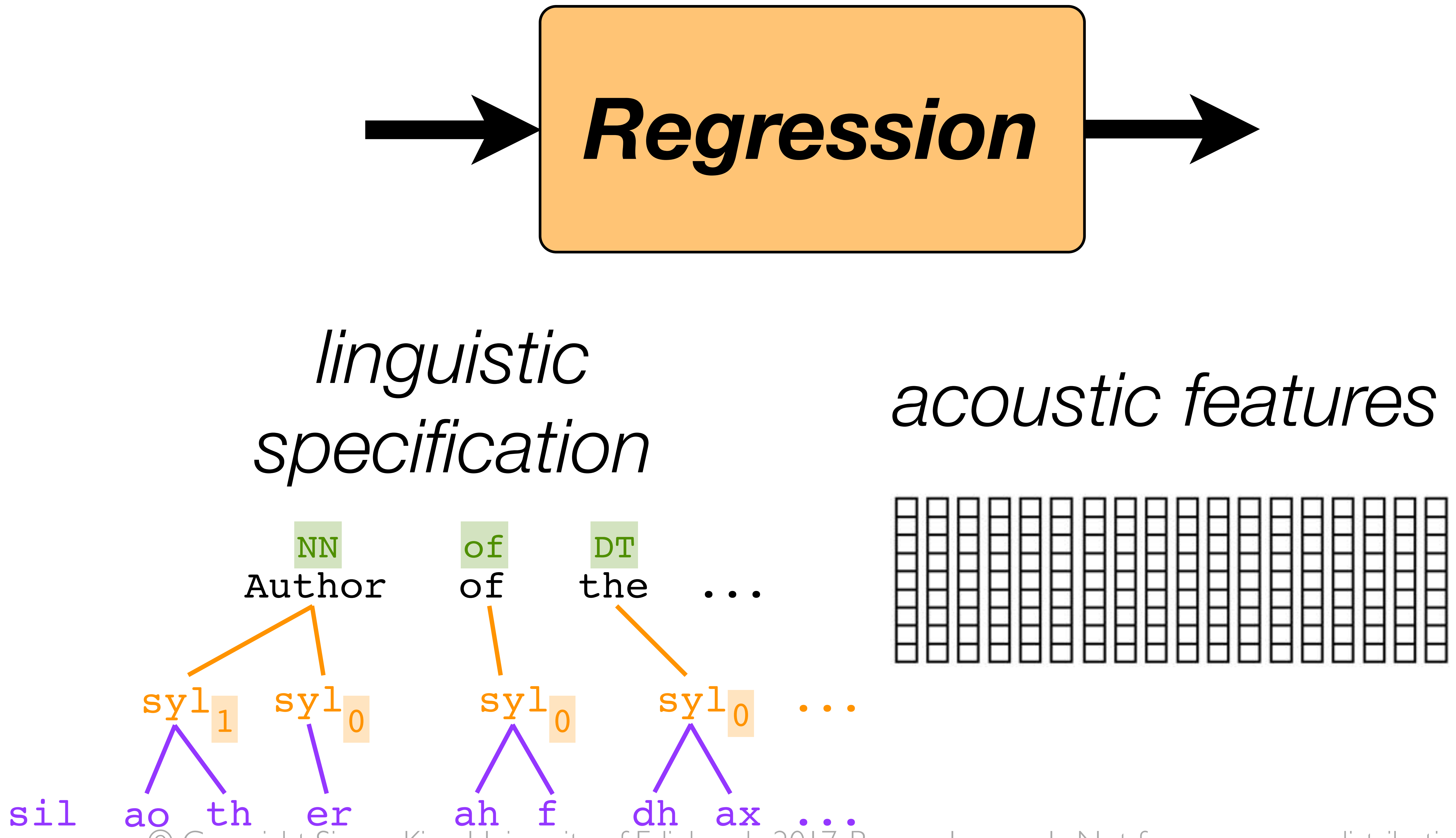
The classic three-stage pipeline of statistical parametric speech synthesis



The classic three-stage pipeline of statistical parametric speech synthesis

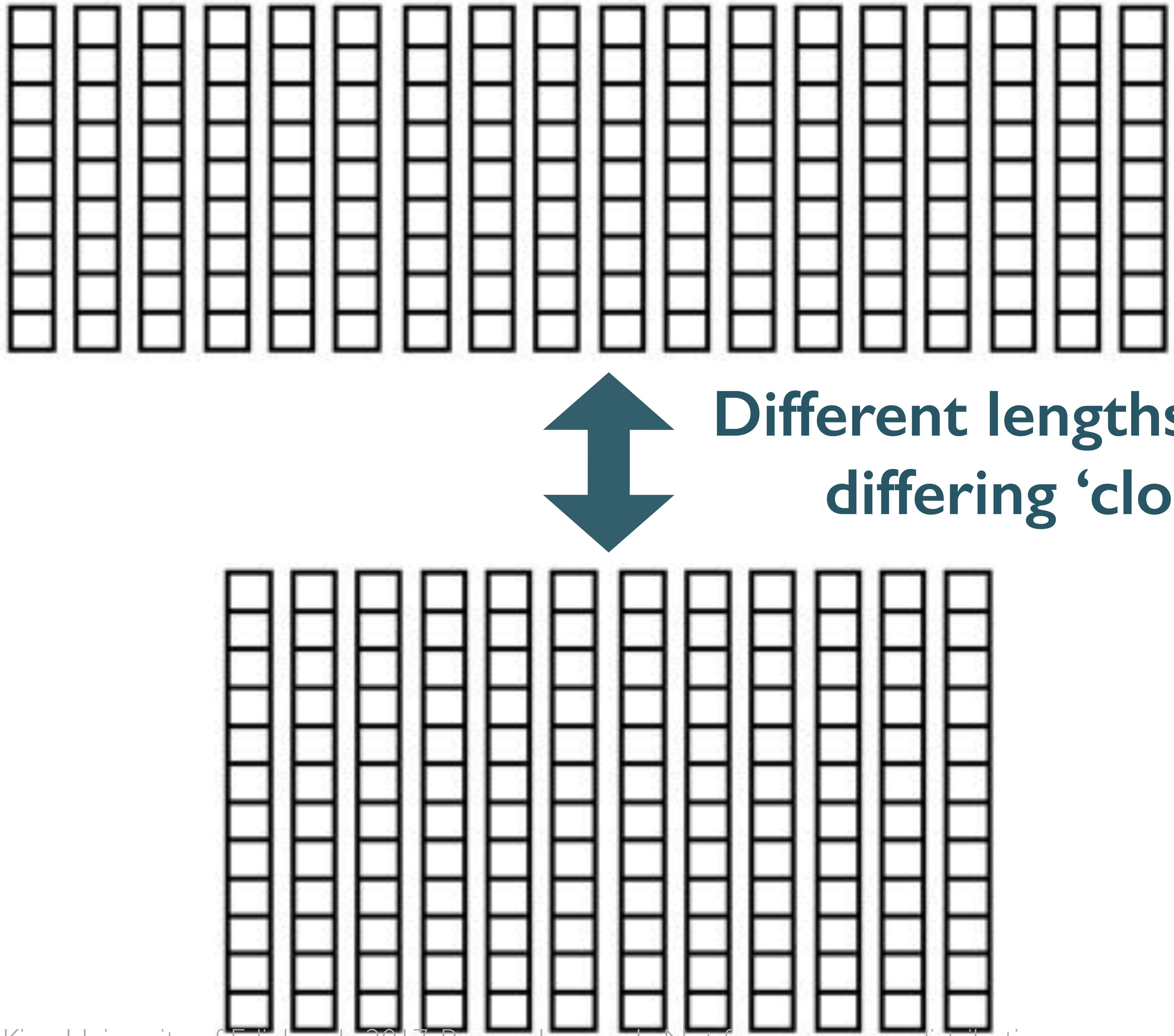


The classic three-stage pipeline of statistical parametric speech synthesis



We can describe the core problem as **sequence-to-sequence regression**

output sequence
(acoustic features)



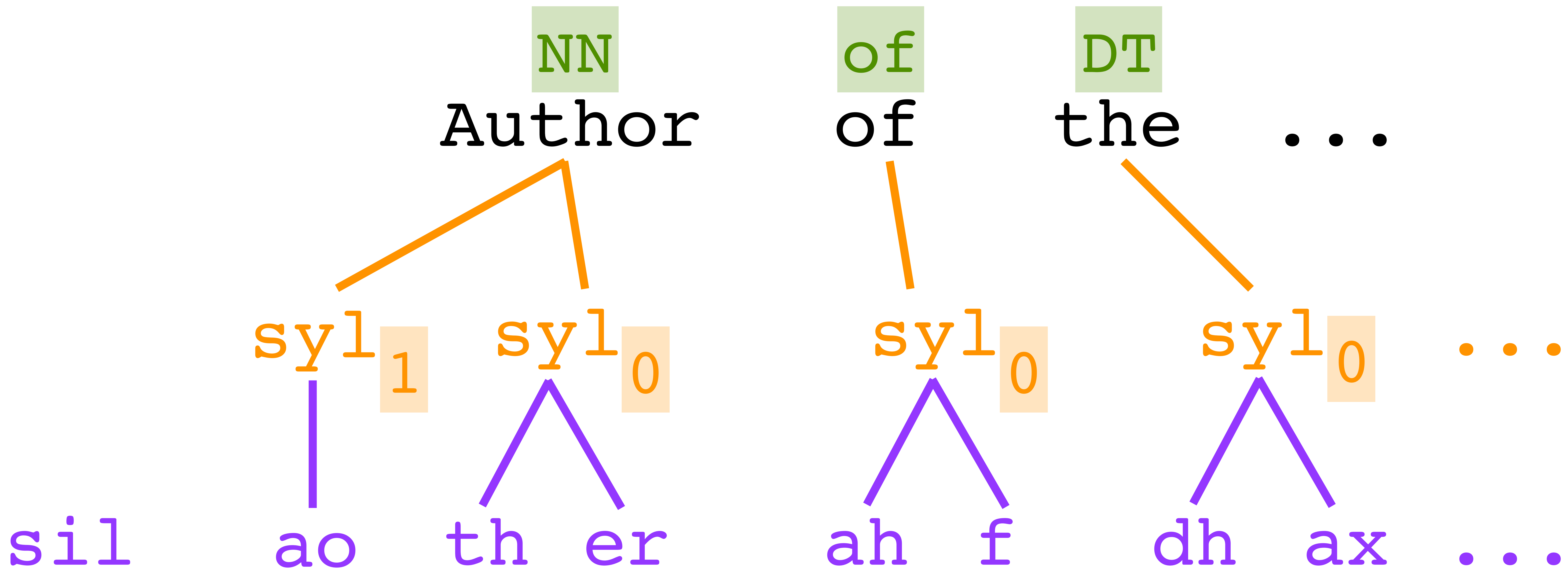
Different lengths, because of differing ‘clock rates’

input sequence
(linguistic features)

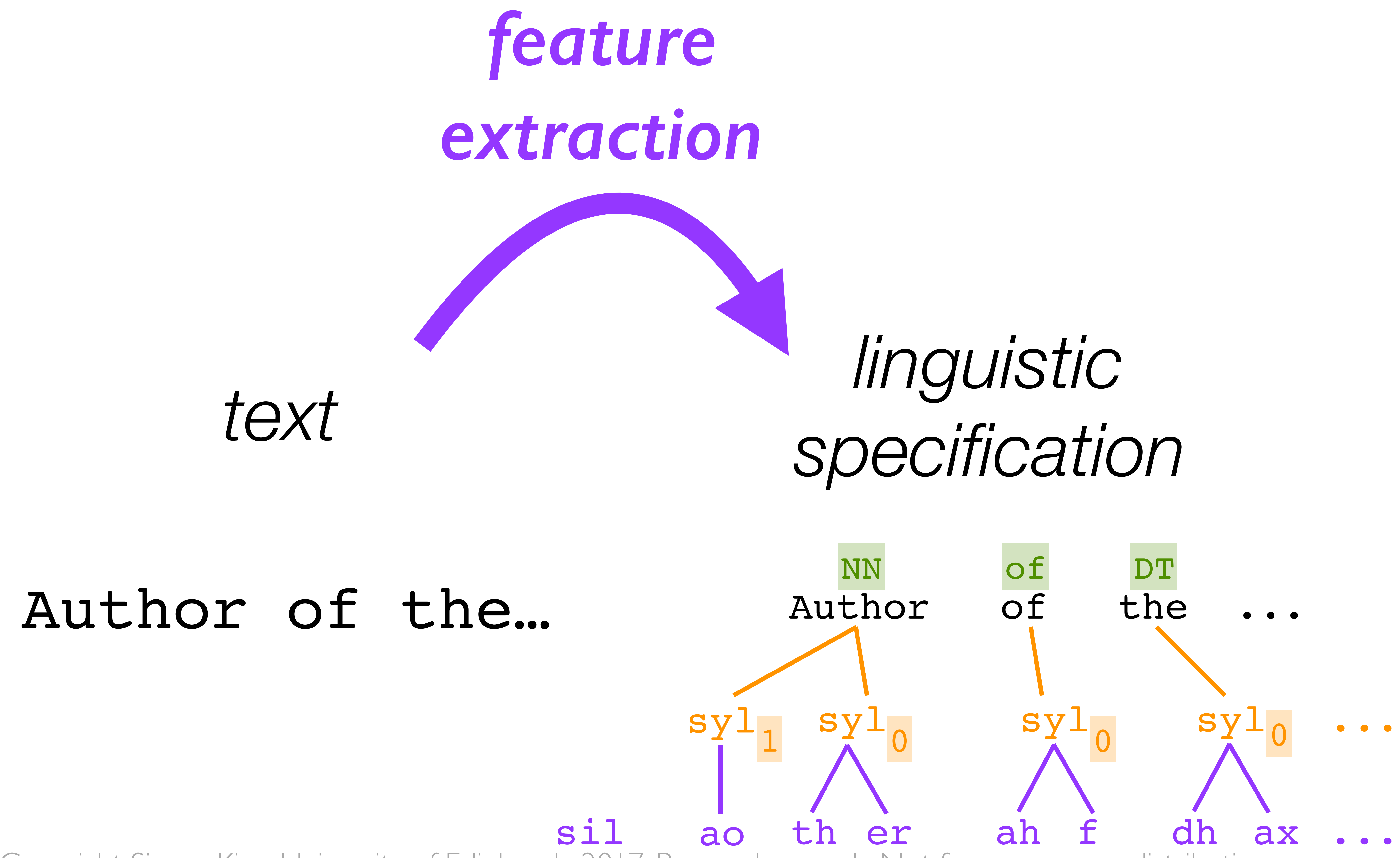
From text to speech

- <u>Text processing</u>
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- Waveform generation
  - acoustic features
  - signal processing

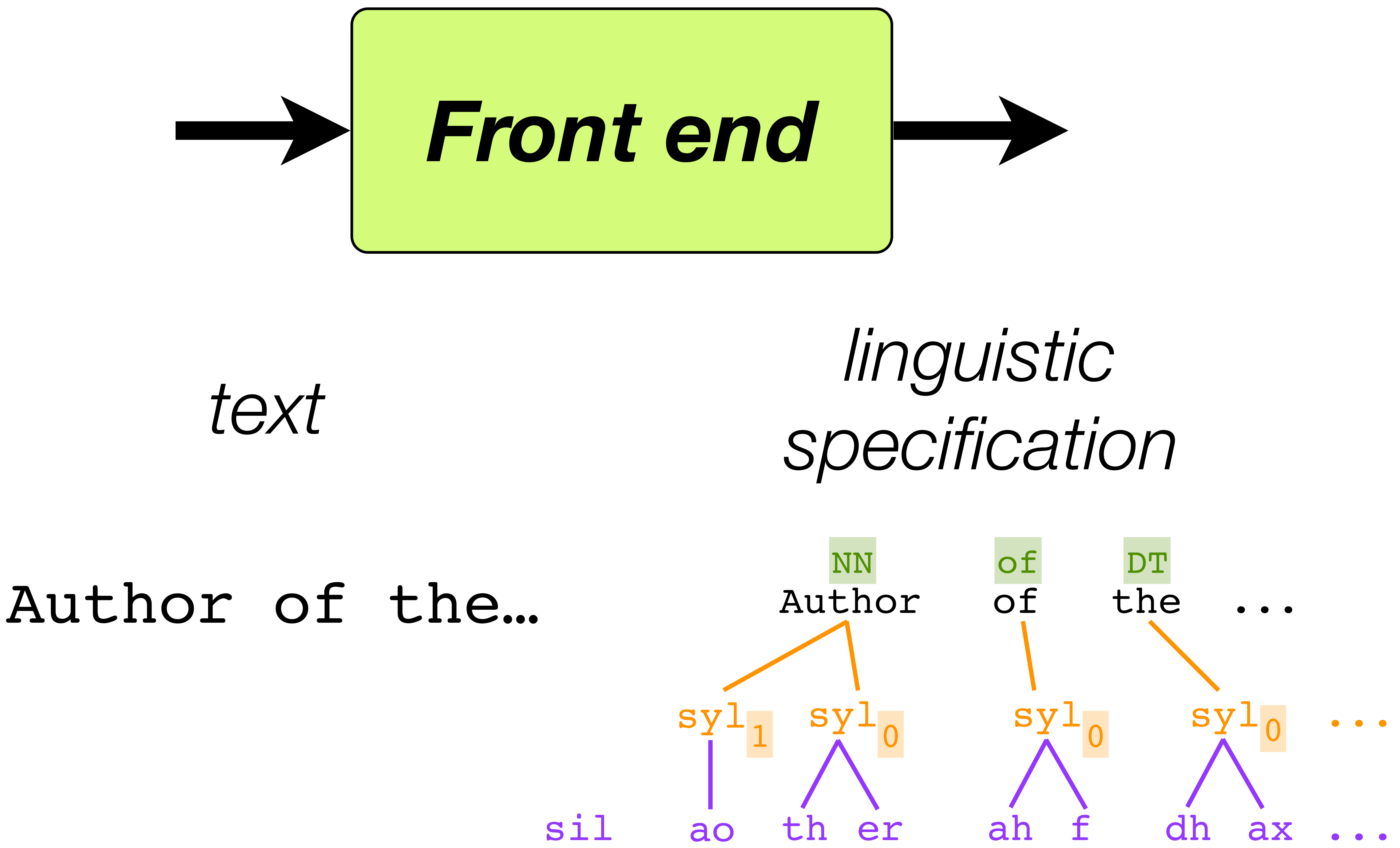
The linguistic specification



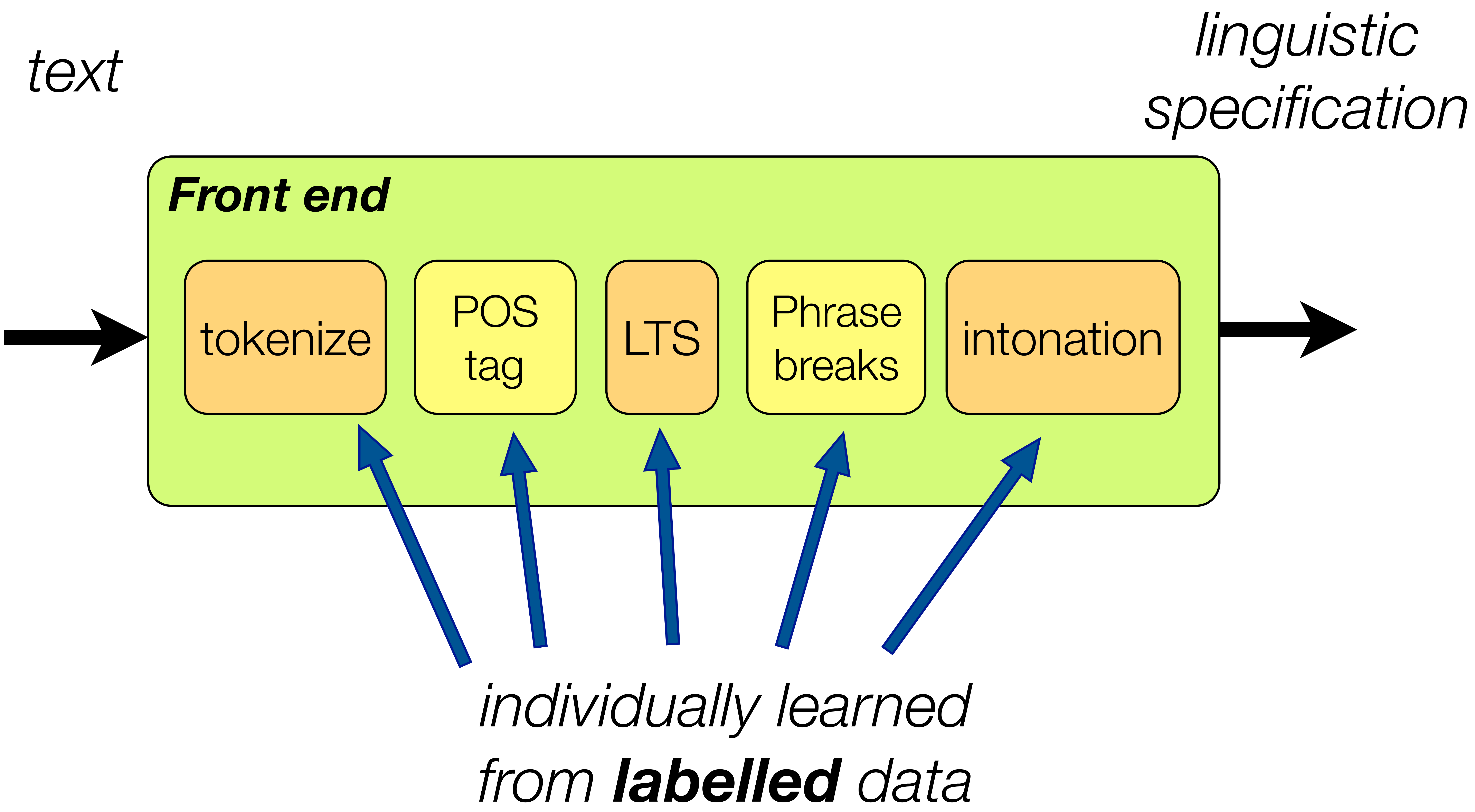
Extracting features from text using the front end



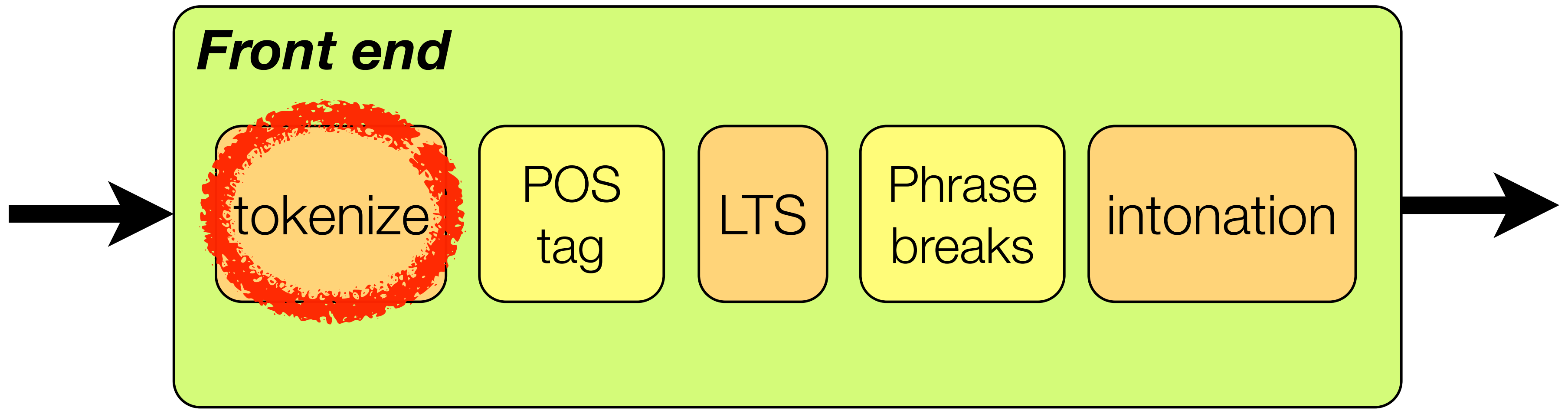
Extracting features from text using the front end



Text processing pipeline

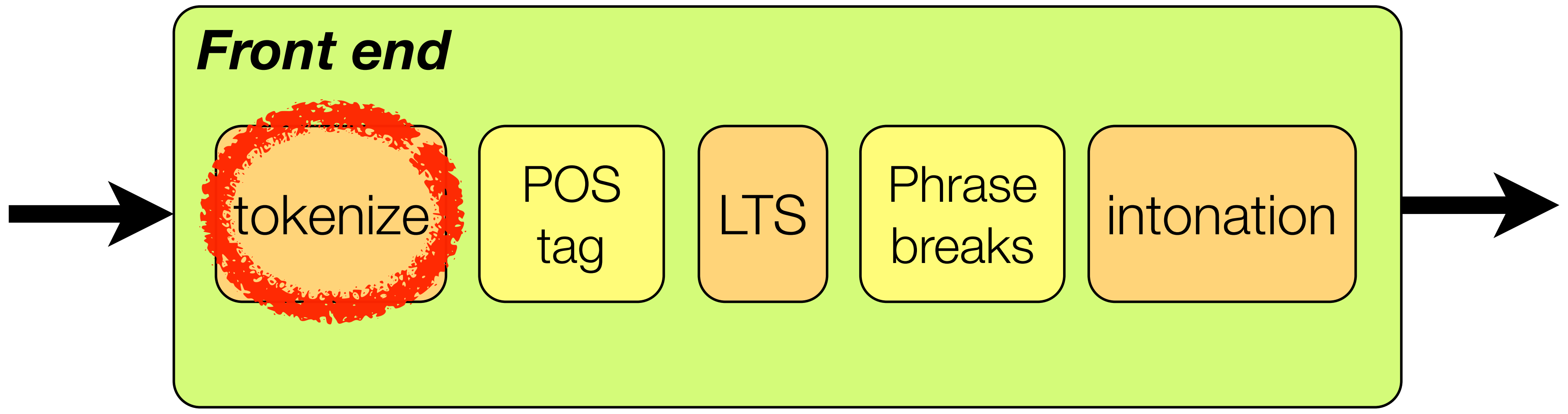


Tokenize & Normalize



- Step 1: divide input stream into tokens, which are potential words
- For English and many other languages
  - rule based
  - whitespace and punctuation are good features
- For some other languages, especially those that don't use whitespace
  - may be more difficult
  - other techniques required (out of scope here)

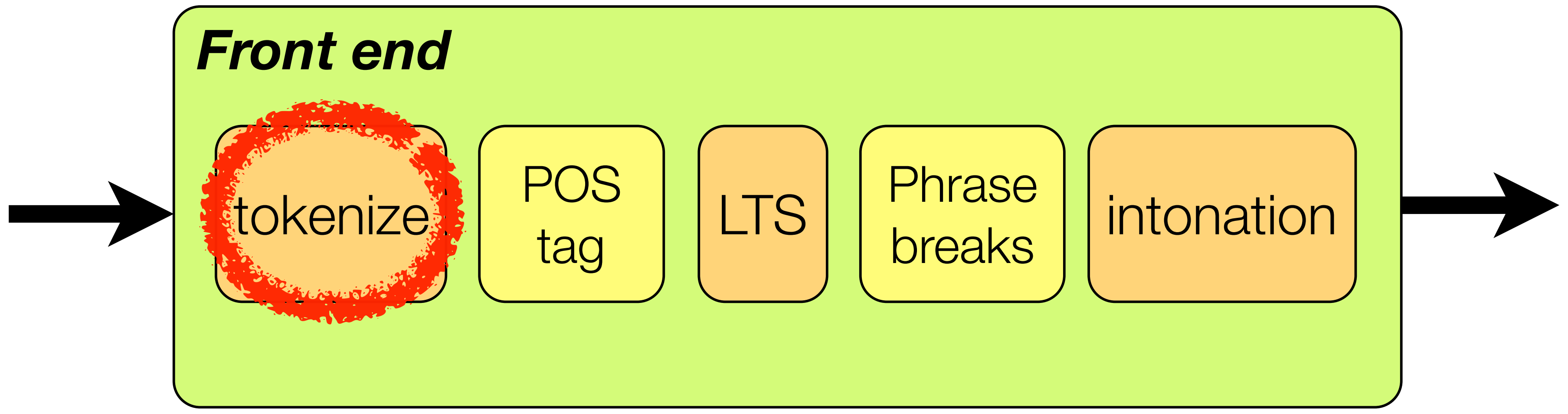
Tokenize & Normalize



- Step 2: classify every token, finding **Non-Standard Words** that need further processing

```
In 2011, I spent £100 at IKEA on 100 DVD holders.
```

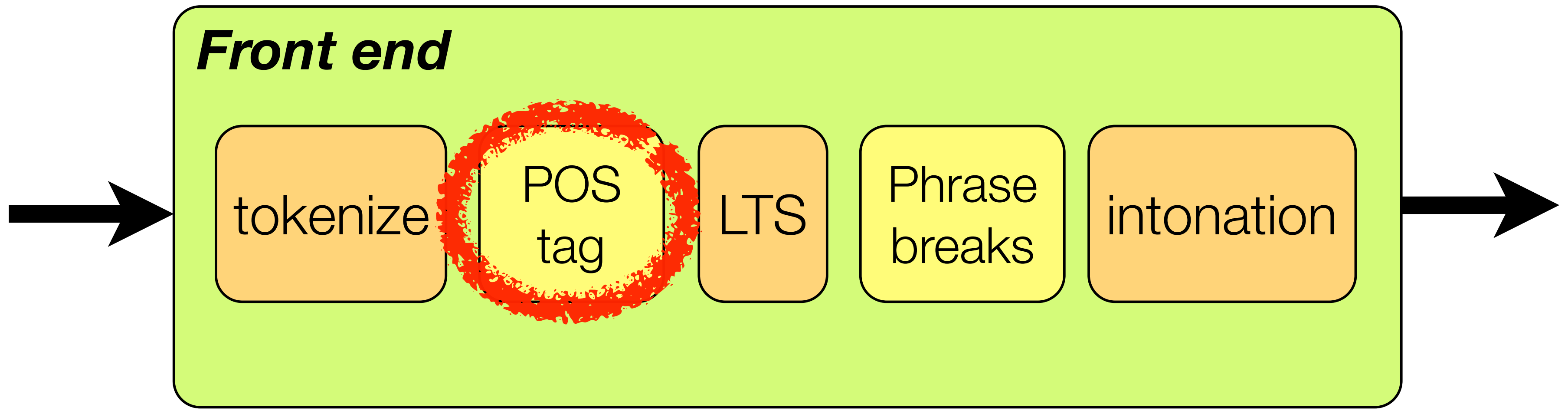
NYER MONEY ASWD NUM LSEQ



Tokenize & Normalize

- Step 3: a set of specialised modules to process NSWs of a each type

2011	⇨	NYER	⇨	twenty eleven		
£100	⇨	MONEY	⇨	one hundred pounds		
IKEA	⇨	ASWD	⇨	*apply letter-to-sound*		
100	⇨	NUM	⇨	one hundred		
DVD	⇨	LSEQ	⇨	D. V. D.	⇨	dee vee dee

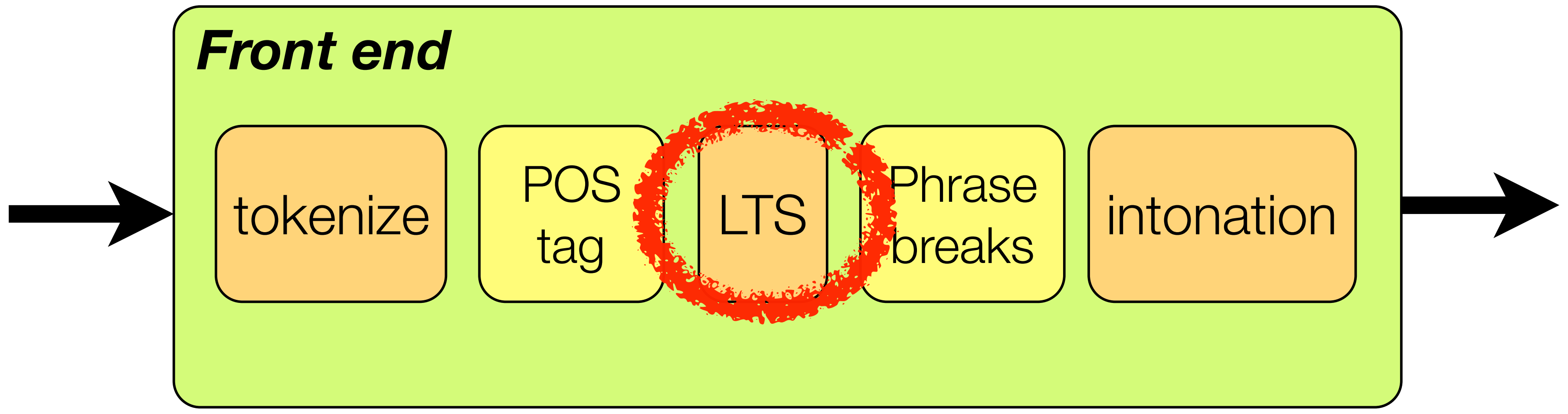


POS tagging

- Part-of-speech tagger
- Accuracy can be very high
- Trained on **annotated** text data
- **Categories** are designed for text, not speech

```
NP  Ed
NP  Beard,
VBZ says
DT  the
NN  push
IN  for
VBP do
PP  it
PP  yourself
NN  lawmaking
VBZ comes
IN  from
NNS voters
WP  who
VBP feel
VBN frustrated
IN  by
PP$ their
JJ  elected
NNS officials.
CC  But
DT  the
NN  initiative
```

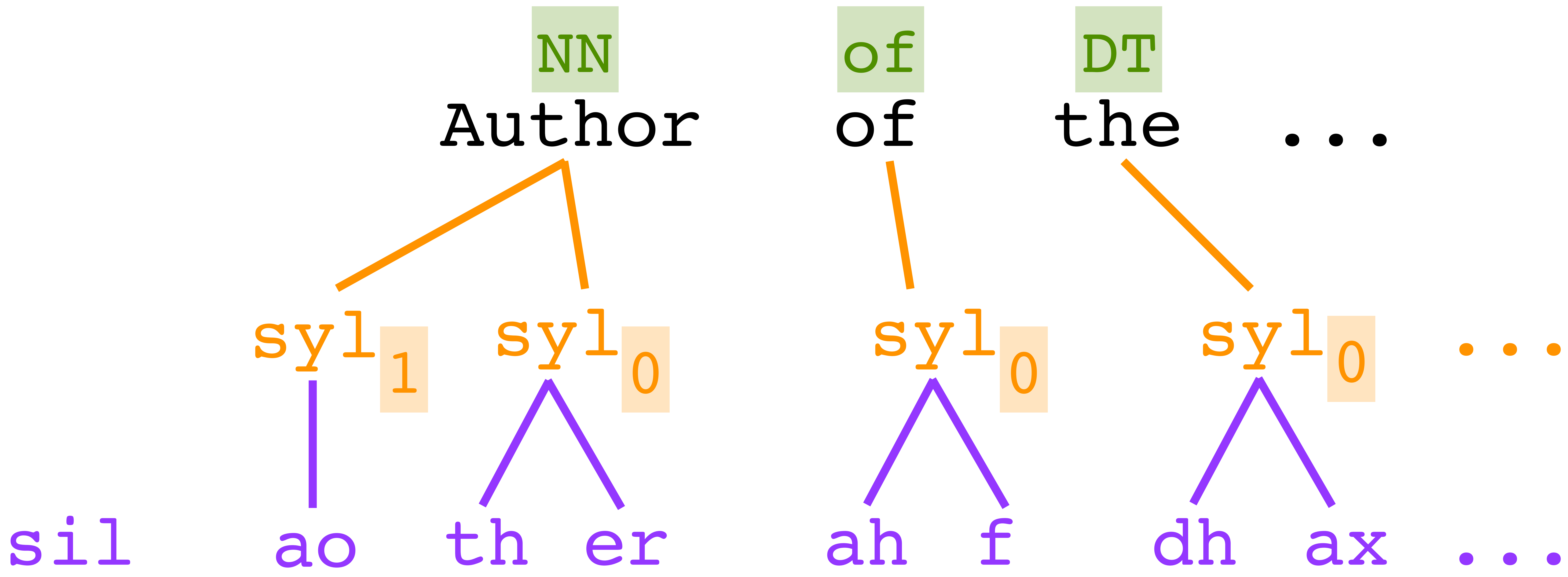
Pronunciation / LTS



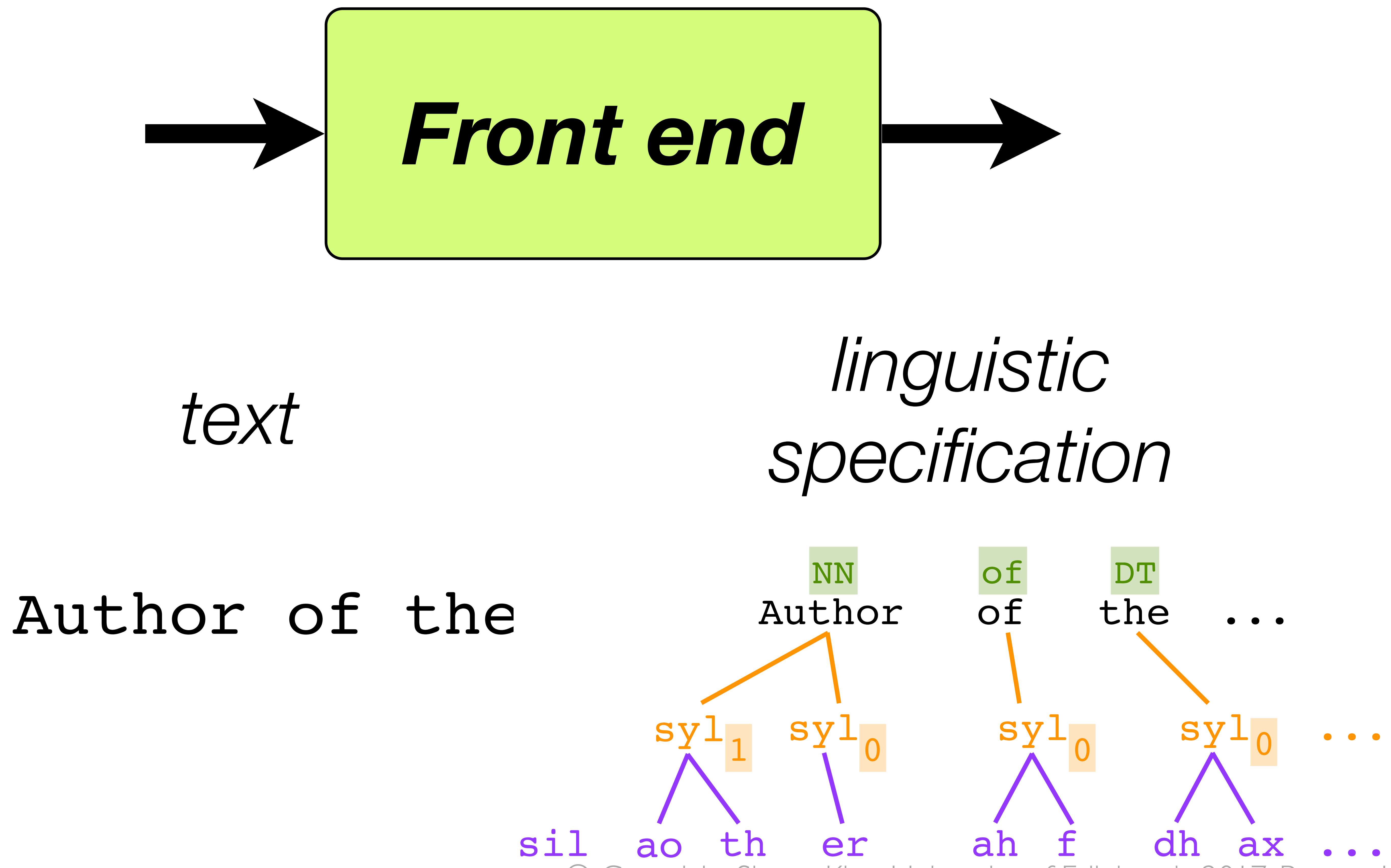
- Pronunciation model
  - dictionary look-up, *plus*
  - letter-to-sound model
- <u>But</u>
  - need deep **knowledge** of the language to design the phoneme set
  - human **expert** must write dictionary

```
AERIALS   EH1 R IY0 AH0 L Z
AERIE   EH1 R IY0
AERIEN   EH1 R IY0 AH0 N
AERIENS   EH1 R IY0 AH0 N Z
AERITALIA   EH2 R IH0 T AE1 L Y AH0
AERO   EH1 R OW0
AEROBATIC   EH2 R AH0 B AE1 T IH0 K
AEROBATICS   EH2 R AH0 B AE1 T IH0 K S
AEROBIC   EH0 R OW1 B IH0 K
AEROBICALLY   EH0 R OW1 B IH0 K L IY0
AEROBICS   ER0 OW1 B IH0 K S
AERODROME   EH1 R AH0 D R OW2 M
AERODROMES   EH1 R AH0 D R OW2 M Z
AERODYNAMIC   EH2 R OW0 D AY0 N AE1 M IH0 K
AERODYNAMICALLY   EH2 R OW0 D AY0 N AE1 M IH0 K L
AERODYNAMICIST   EH2 R OW0 D AY0 N AE1 M IH0 S IH
AERODYNAMICISTS   EH2 R OW0 D AY0 N AE1 M IH0 S I
AERODYNAMICISTS(1)   EH2 R OW0 D AY0 N AE1 M IH0
AERODYNAMICS   EH2 R OW0 D AY0 N AE1 M IH0 K S
AERODYNE   EH1 R AH0 D AY2 N
AERODYNE'S   EH1 R AH0 D AY2 N Z
AEROFLOT   EH1 R OW0 F L AA2 T
```

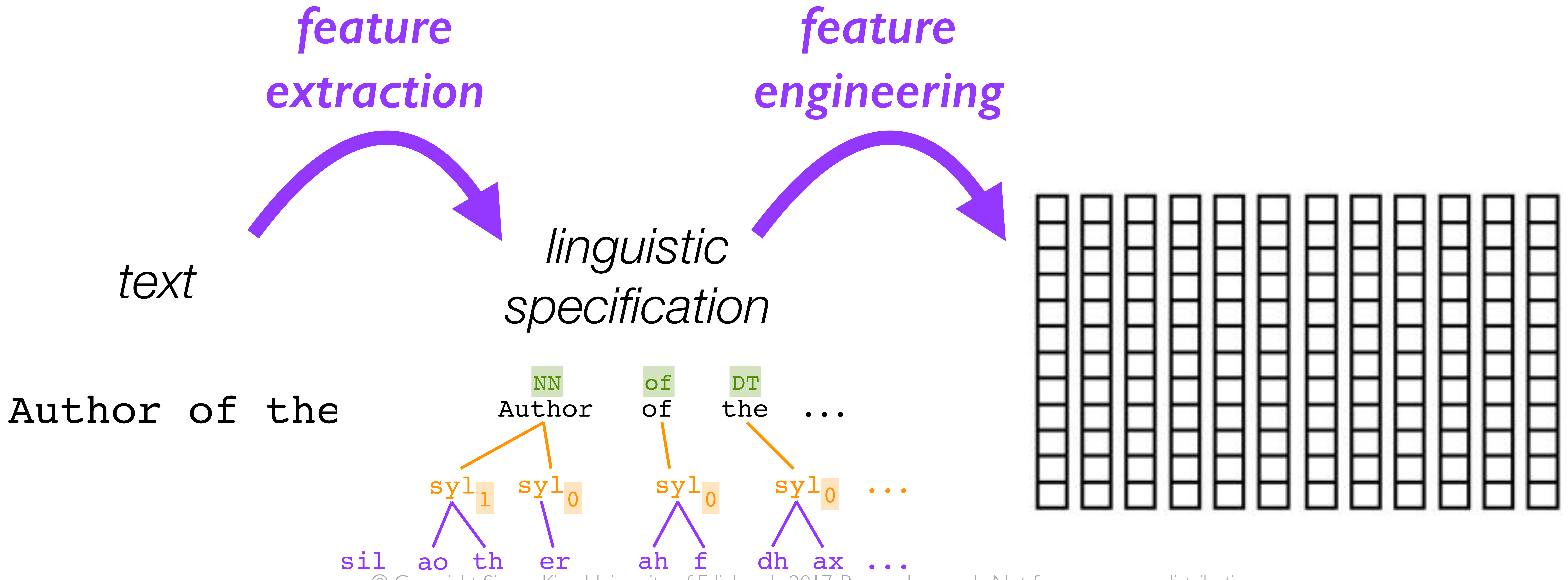
The linguistic specification



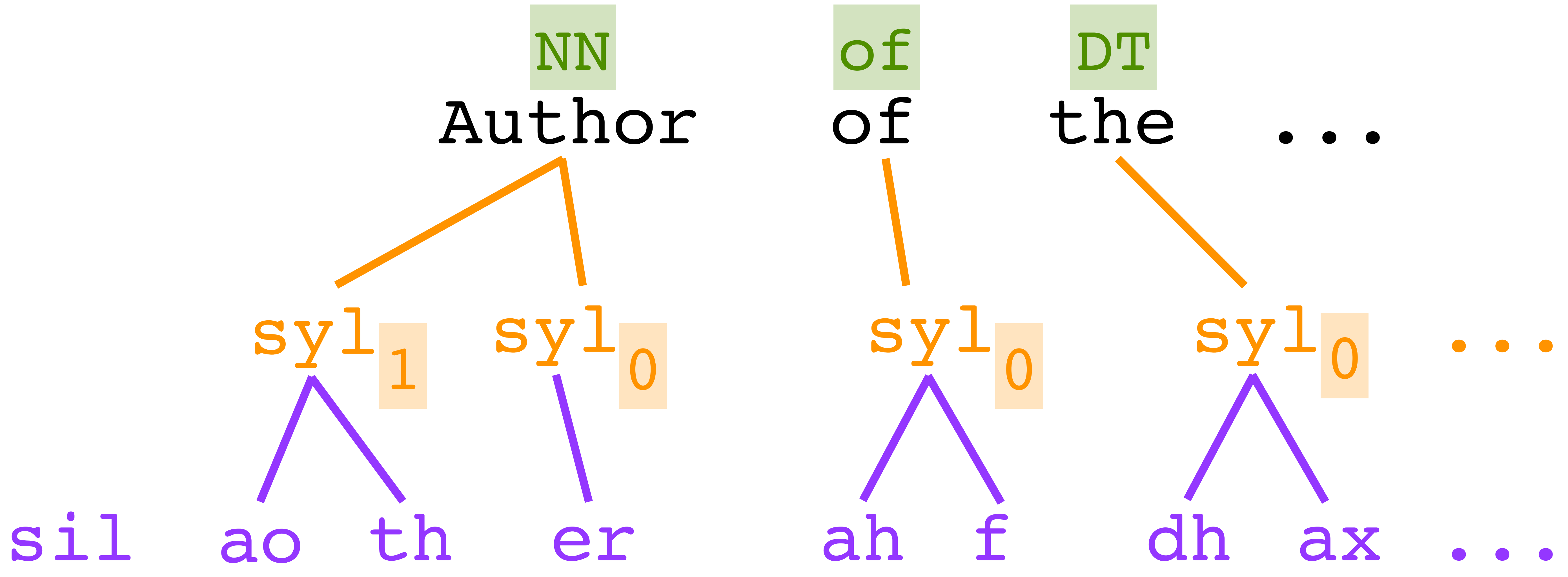
Linguistic feature engineering

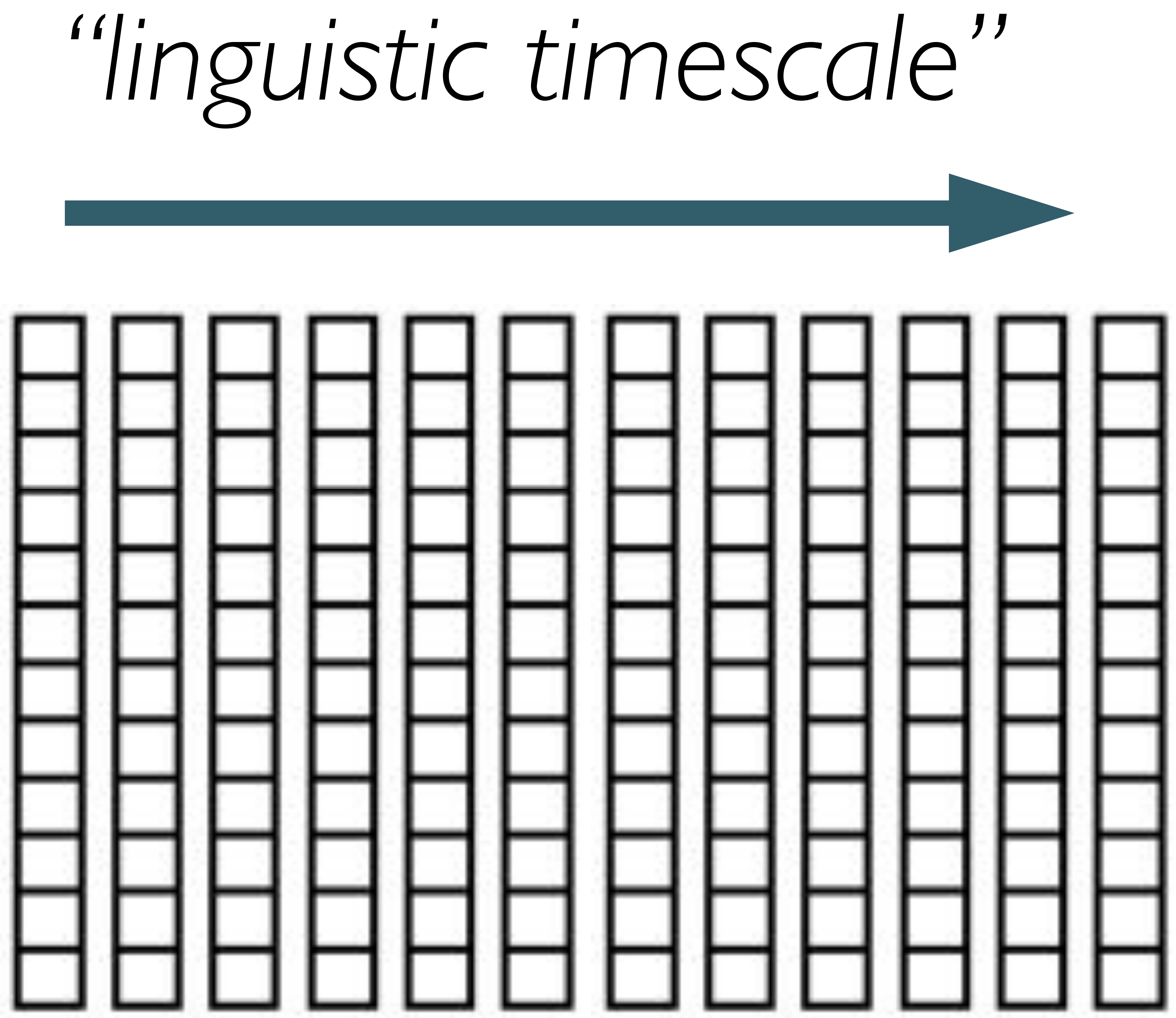


Linguistic feature engineering



Flatten & encode: convert linguistic specification to vector sequence

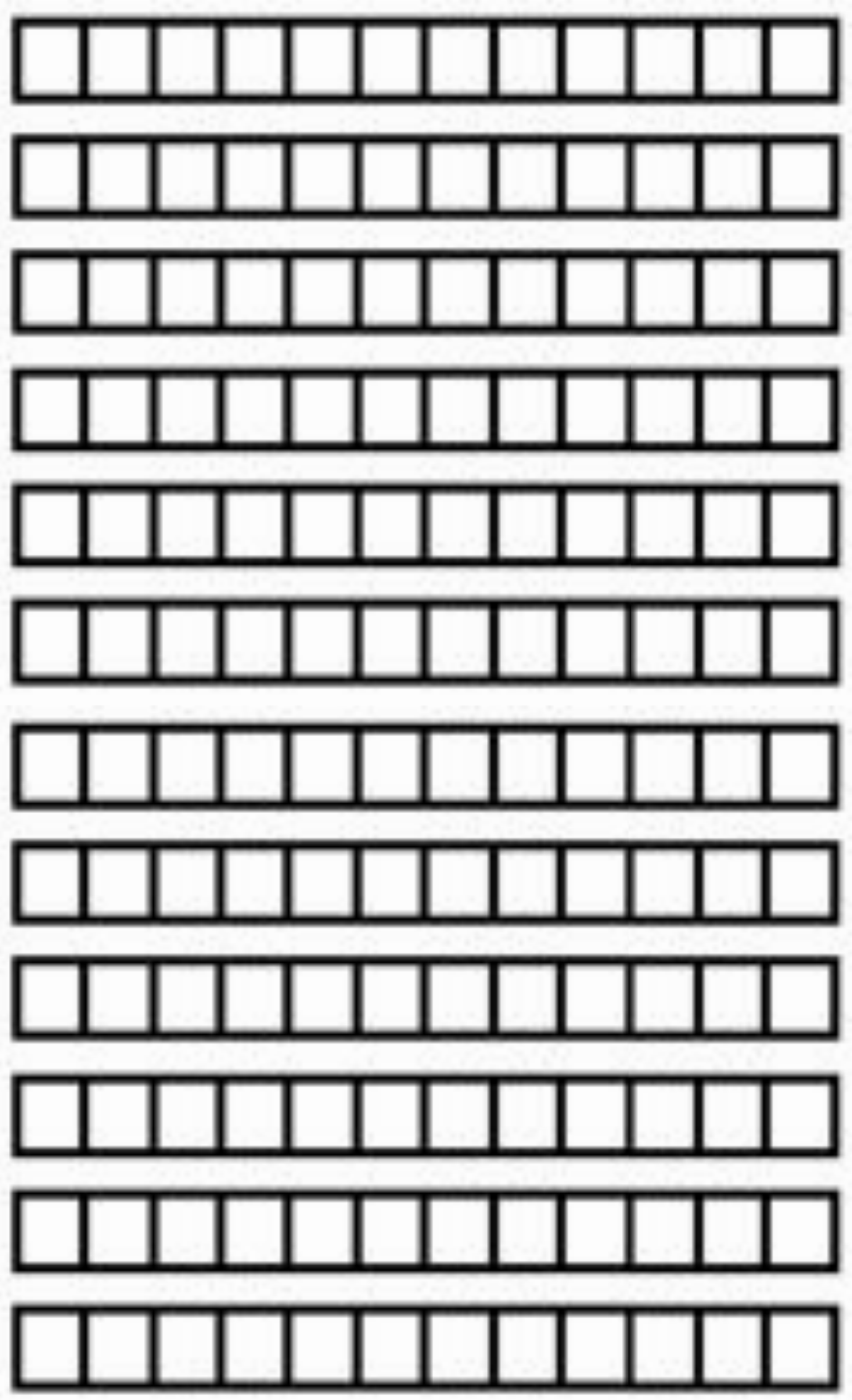




Upsample: add duration information

linguistic timescale → acoustic framerate

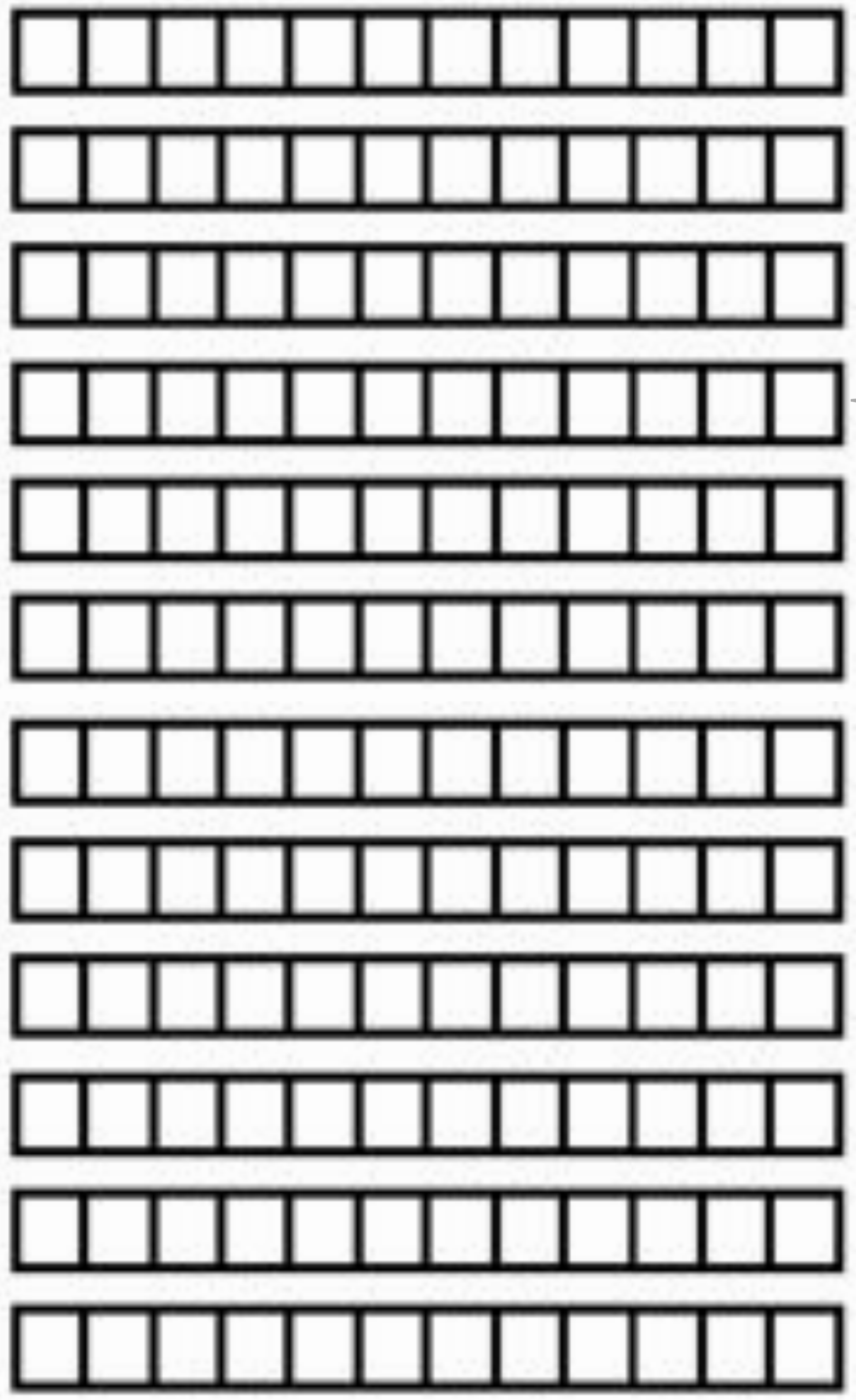
predict durations



Upsample: add duration information

linguistic timescale → acoustic framerate

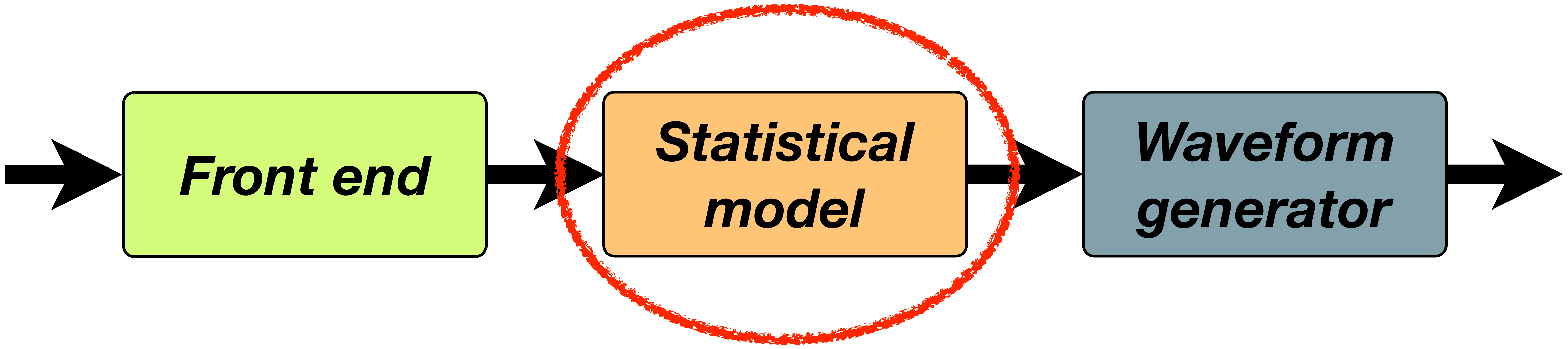
predict durations



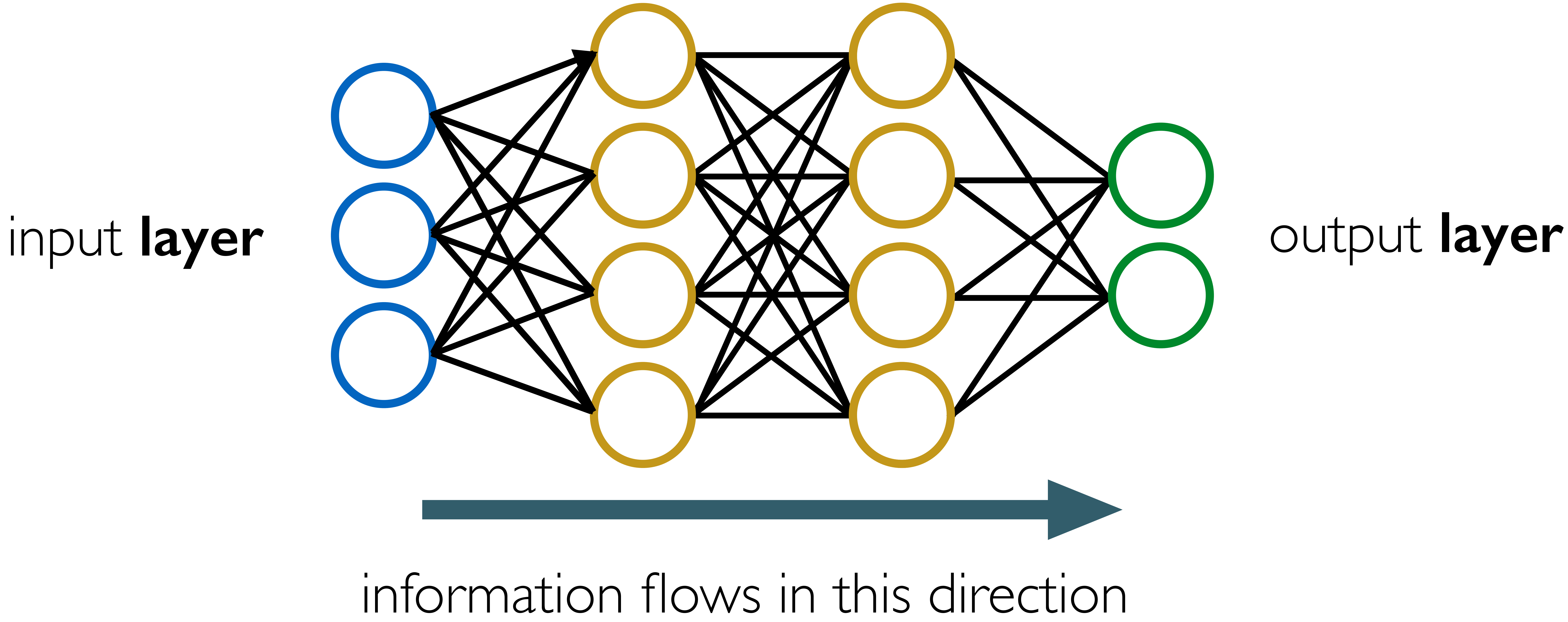
```
[0 0 1 0 0 1 0 1 1 0 … 0.2  0.0]
[0 0 1 0 0 1 0 1 1 0 … 0.2  0.1]
…
[0 0 1 0 0 1 0 1 1 0 … 0.2  1.0]
[0 0 1 0 0 1 0 1 1 0 … 0.4  0.0]
[0 0 1 0 0 1 0 1 1 0 … 0.4  0.5]
[0 0 1 0 0 1 0 1 1 0 … 0.4  1.0]
…
[0 0 1 0 0 1 0 1 1 0 … 1.0  1.0]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.0]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.2]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.4]
…
```

From text to speech

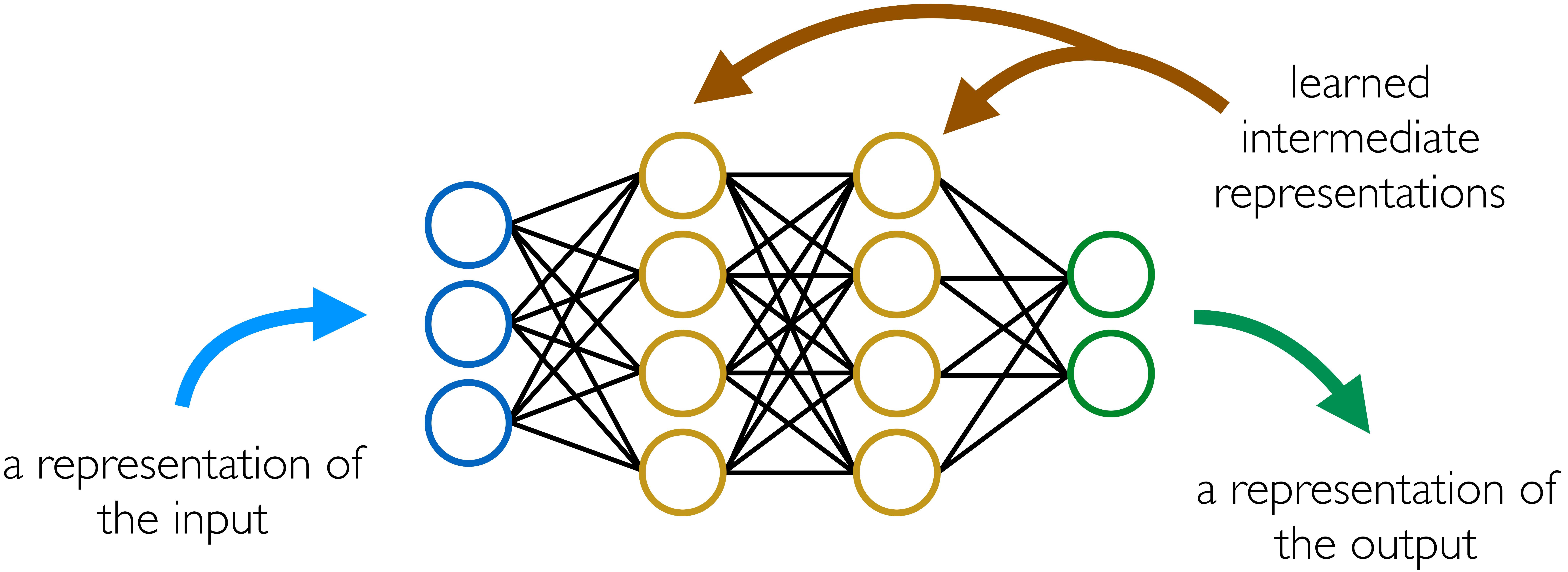
- Text processing
    - pipeline architecture
    - linguistic specification
- Regression
    - duration model
    - acoustic model
- Waveform generation
    - acoustic features
    - signal processing



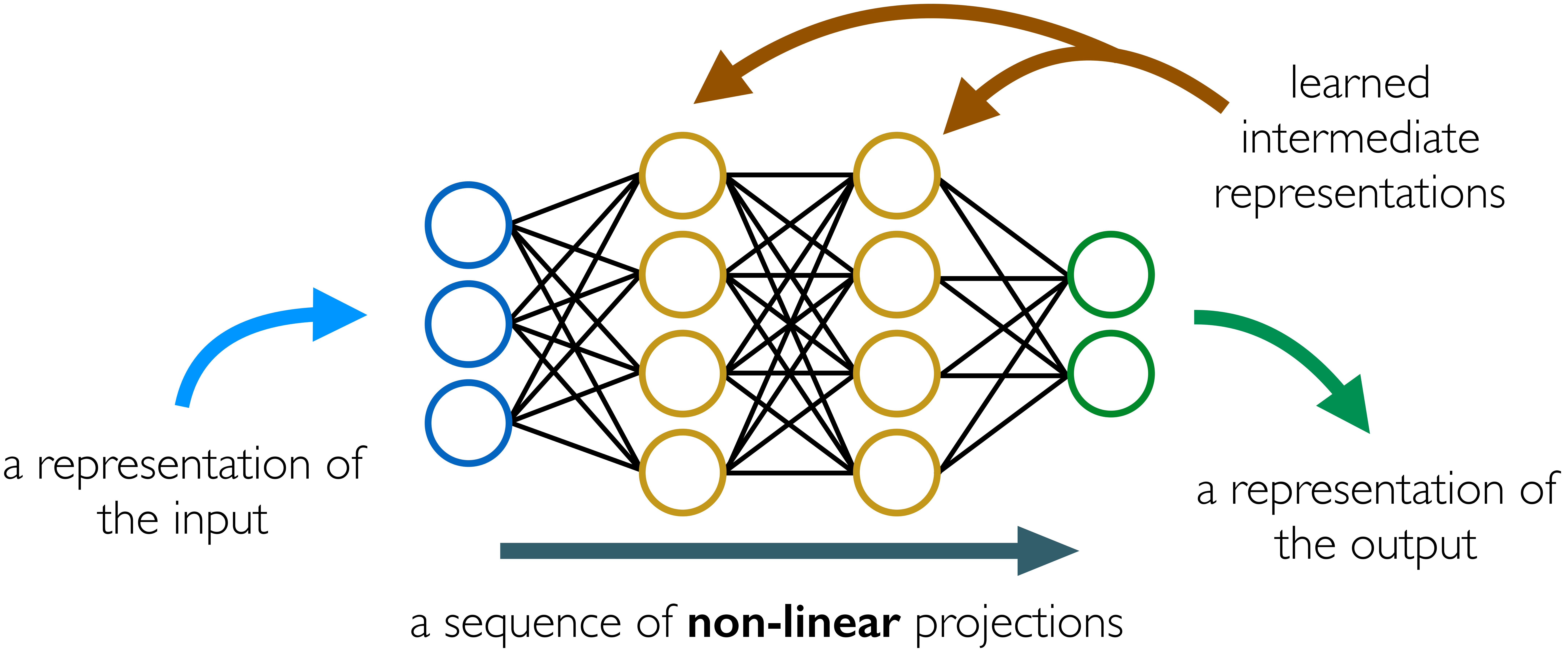
Acoustic model: a simple "feed forward" neural network



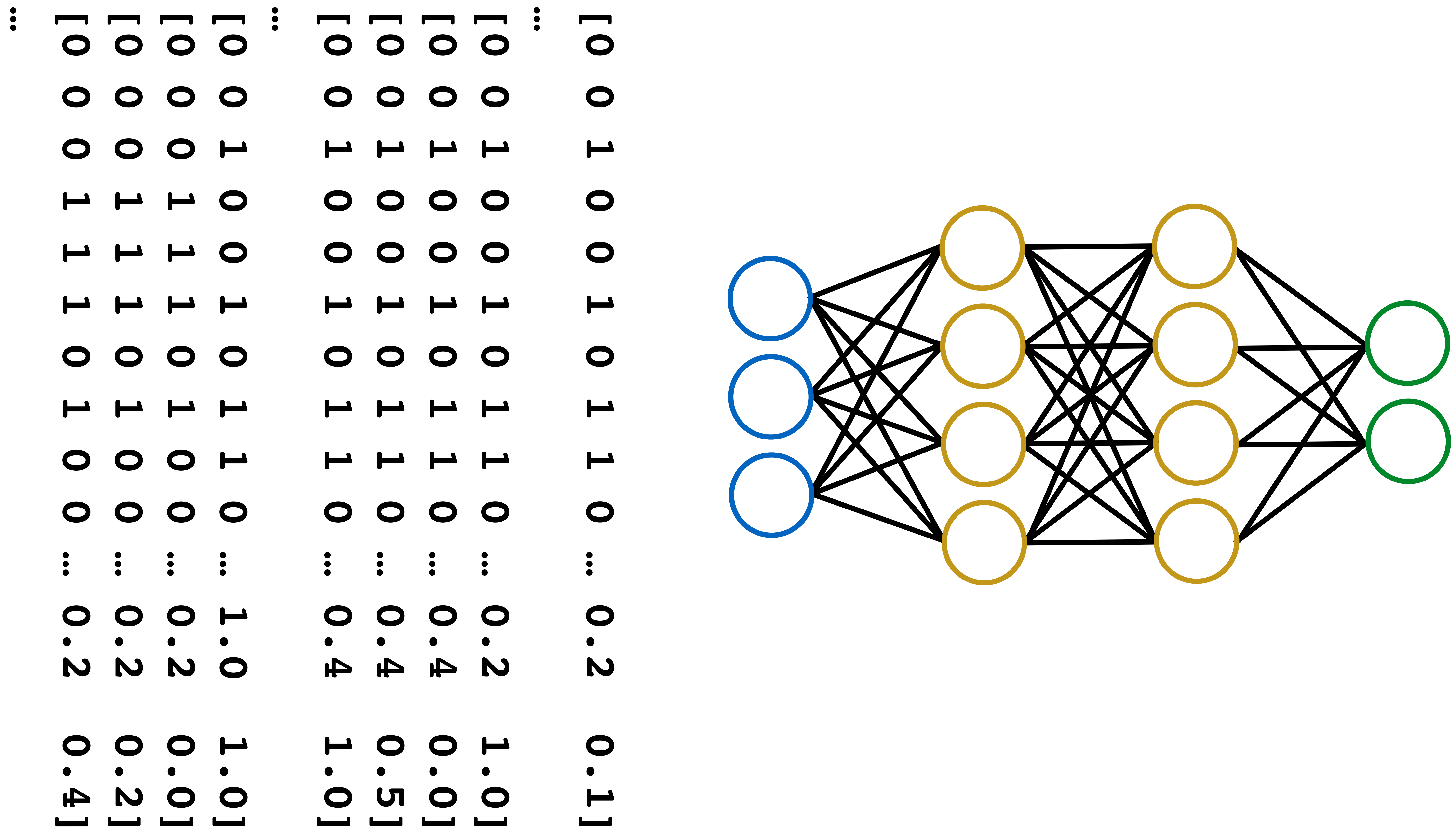
What are all those layers for?



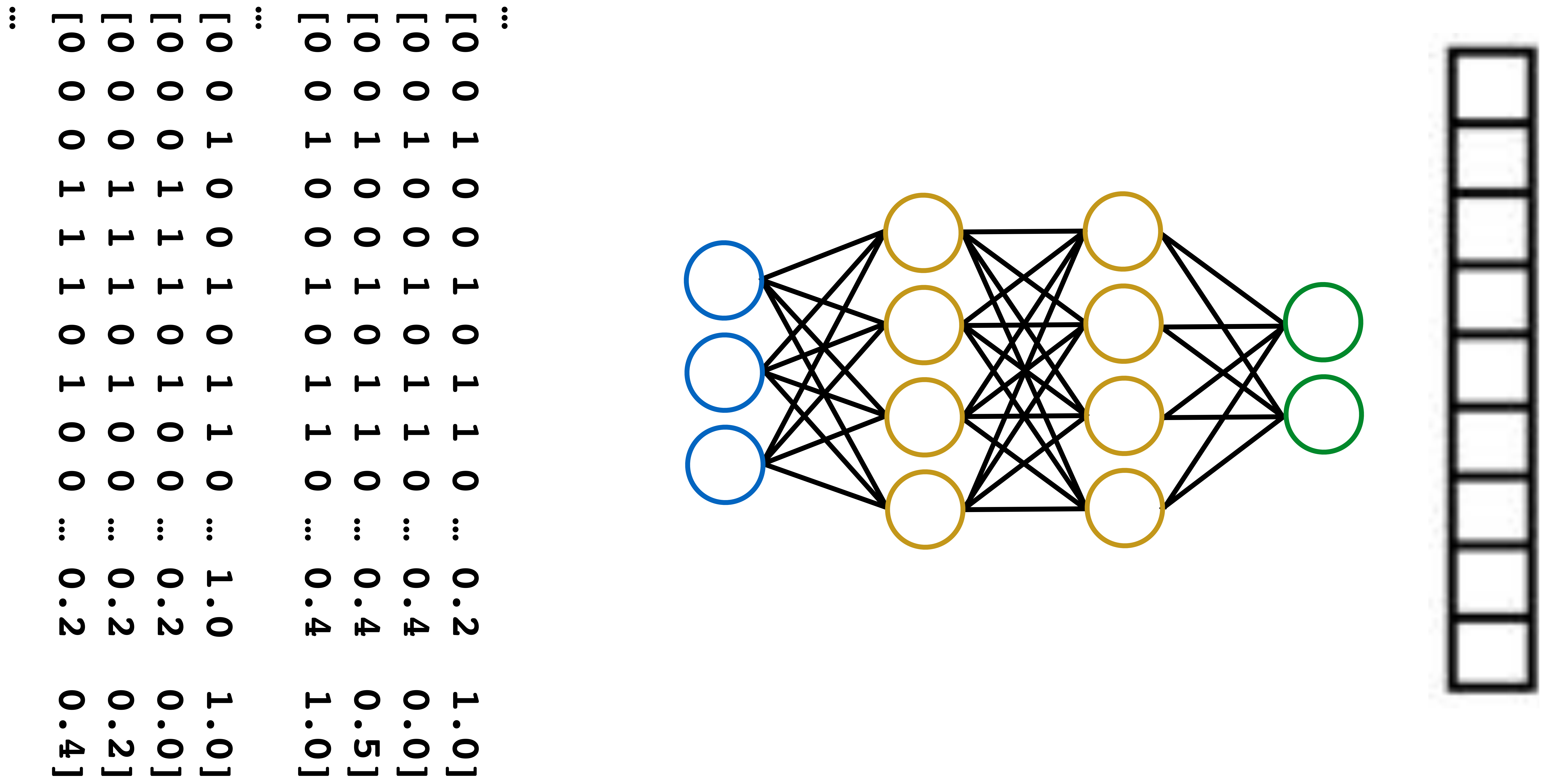
What are all those layers for?



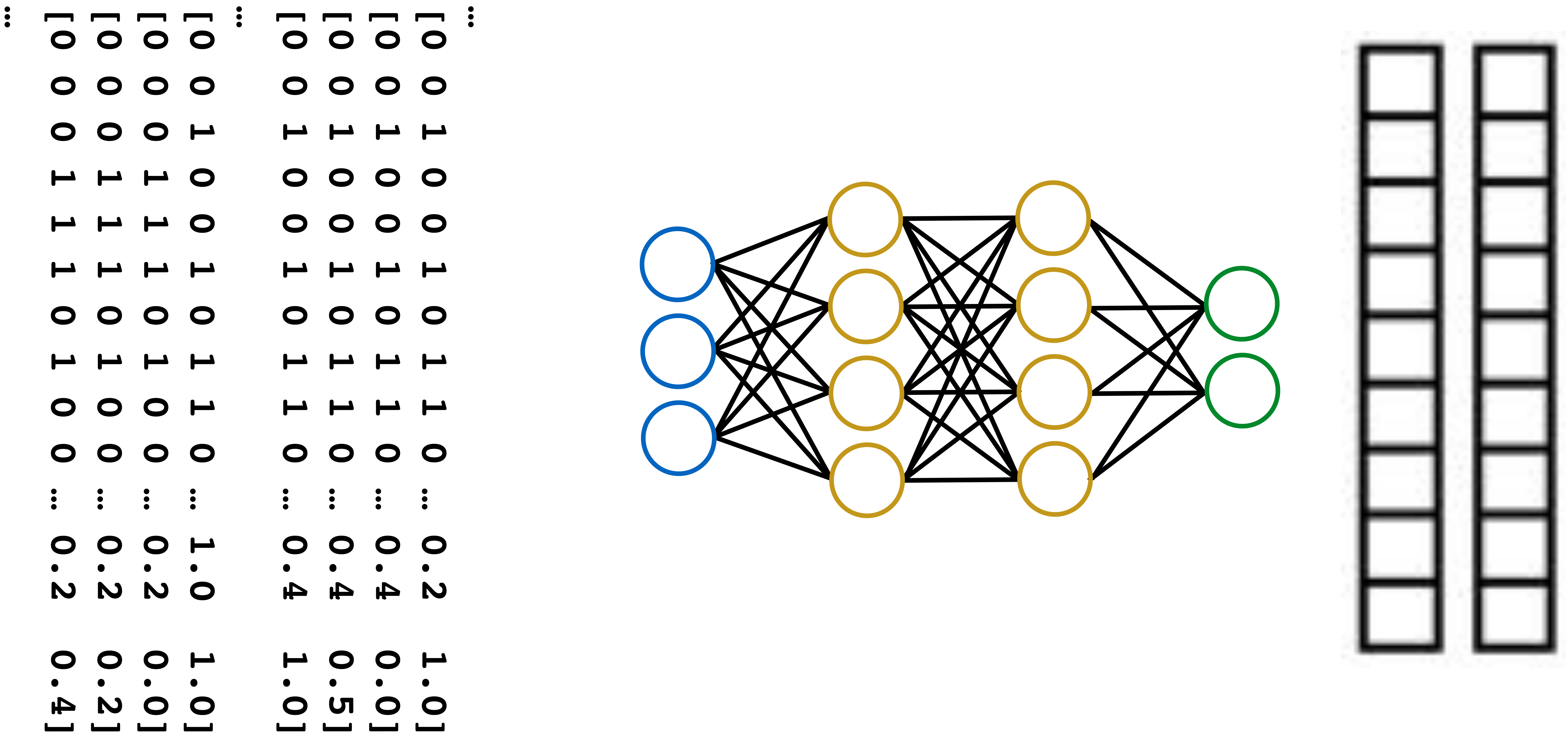
Synthesis with a neural network



Synthesis with a neural network



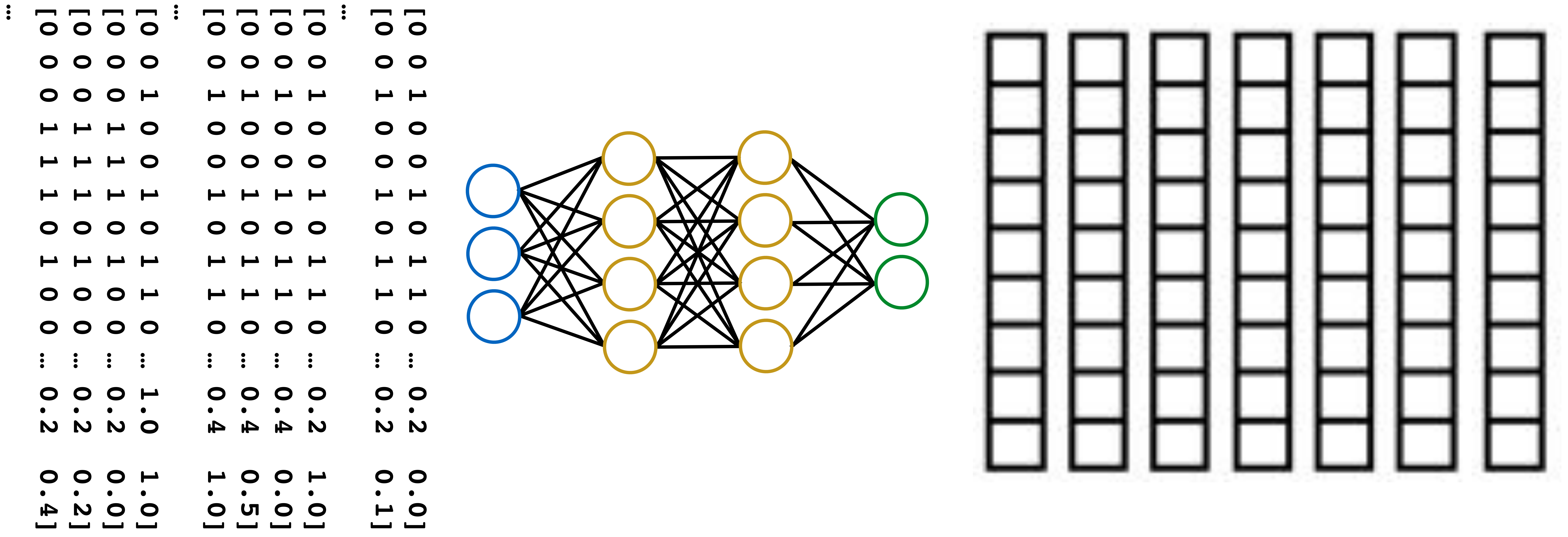
Synthesis with a neural network



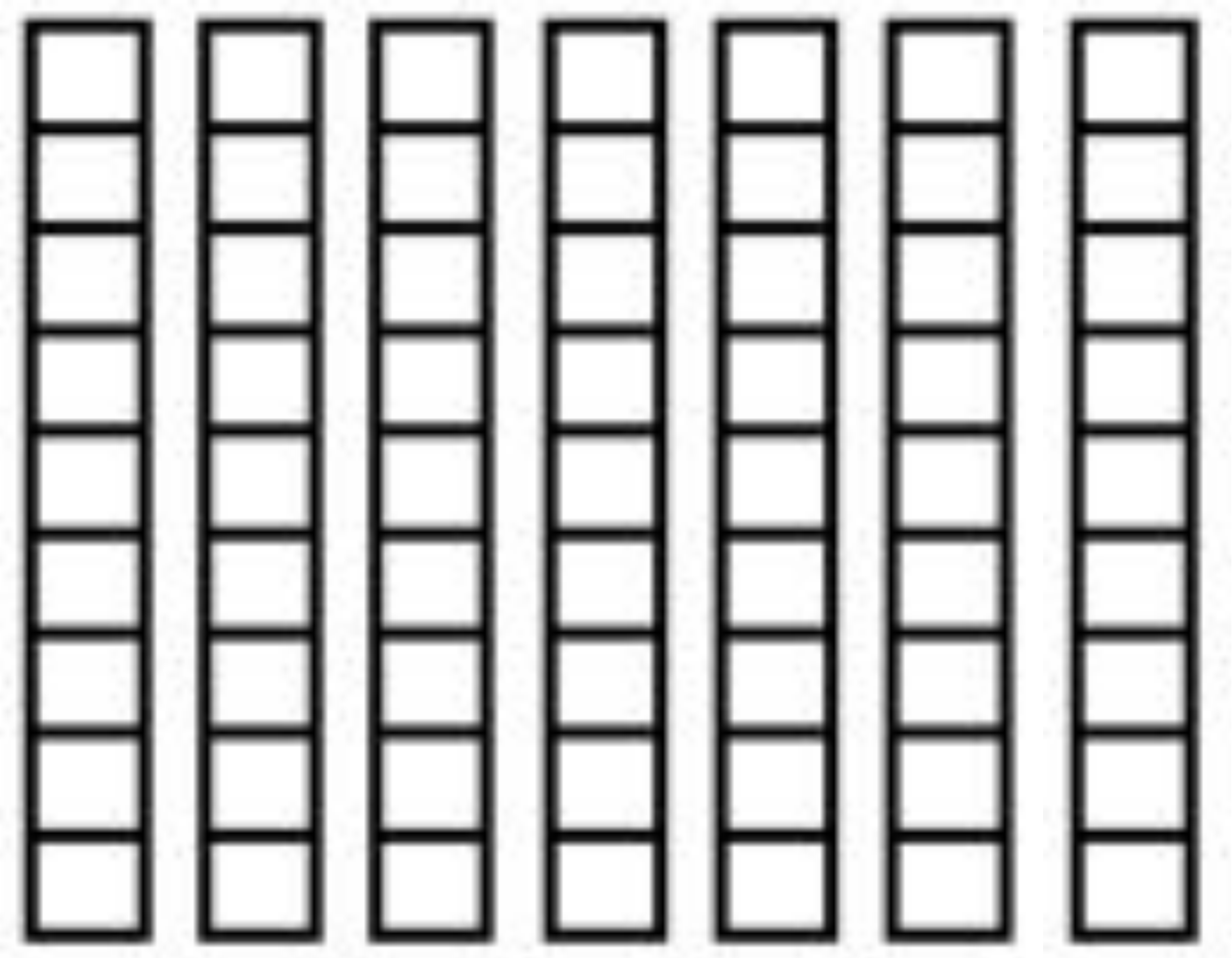
From text to speech

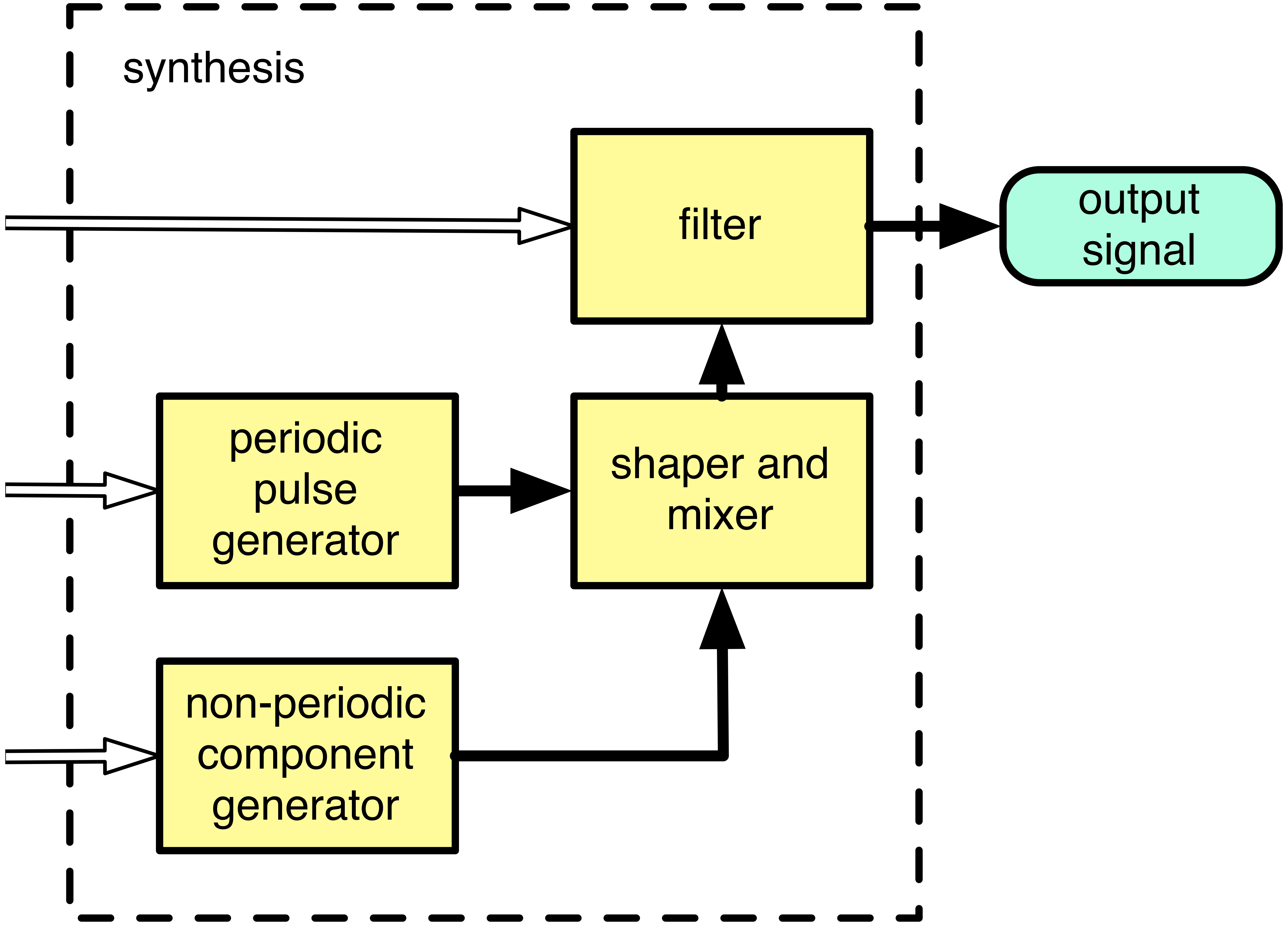
- Text processing
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- <u>Waveform generation</u>
  - acoustic features
  - signal processing

What are the acoustic features?

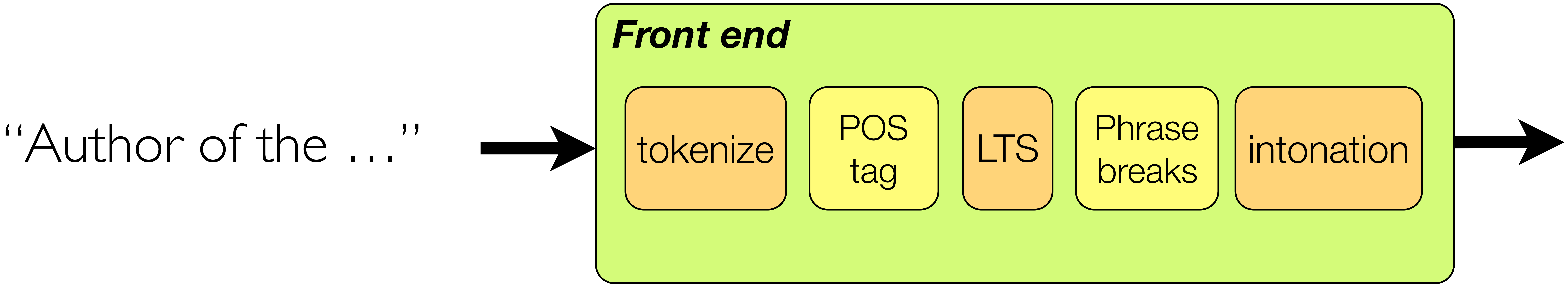


What are the acoustic features?

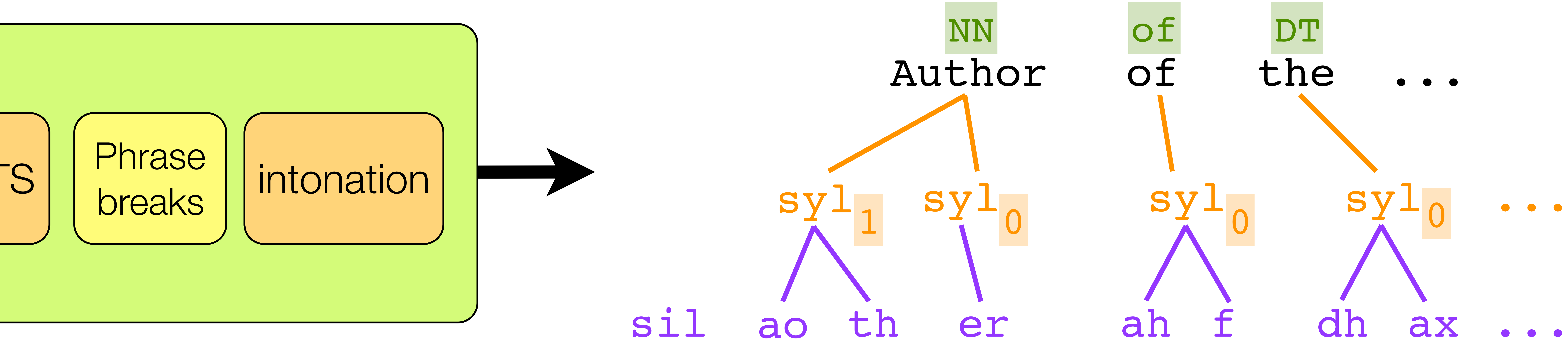




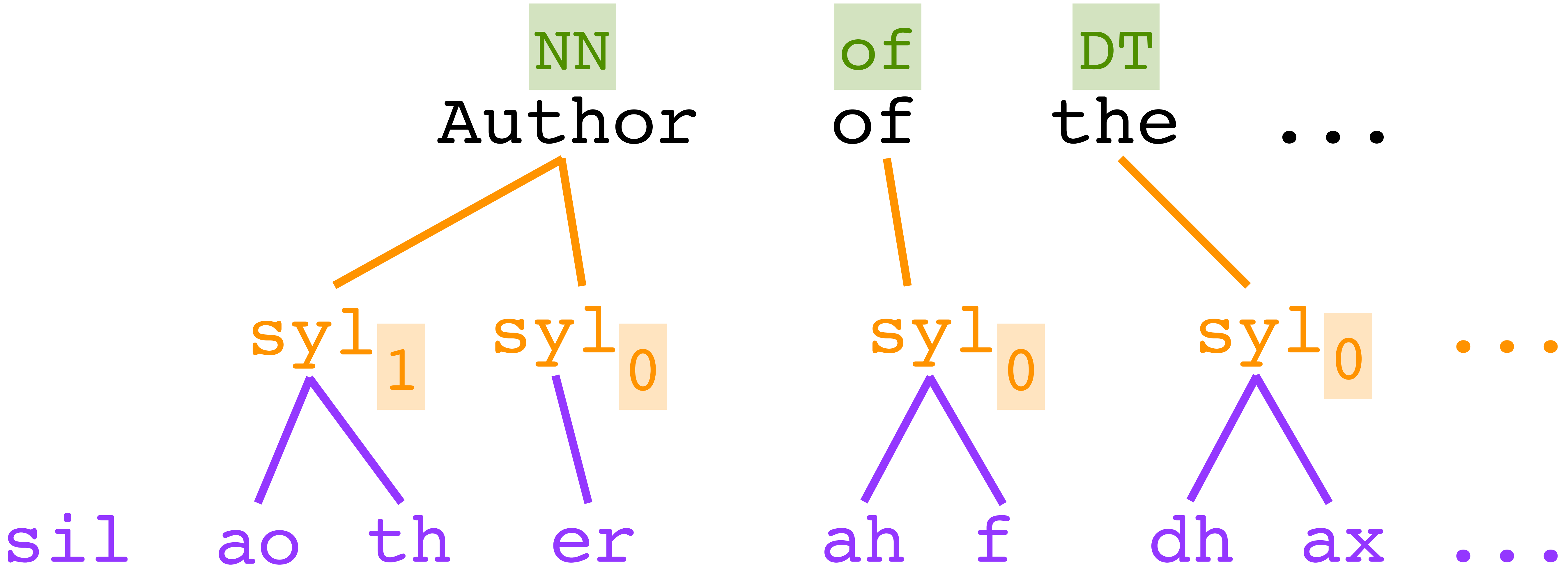
Putting it all together: text-to-speech with a neural network



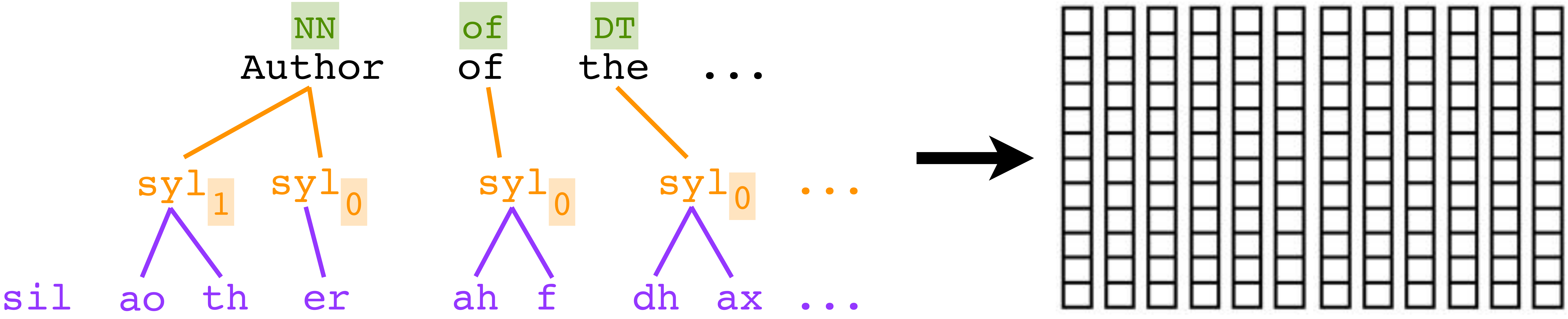
Putting it all together: text-to-speech with a neural network



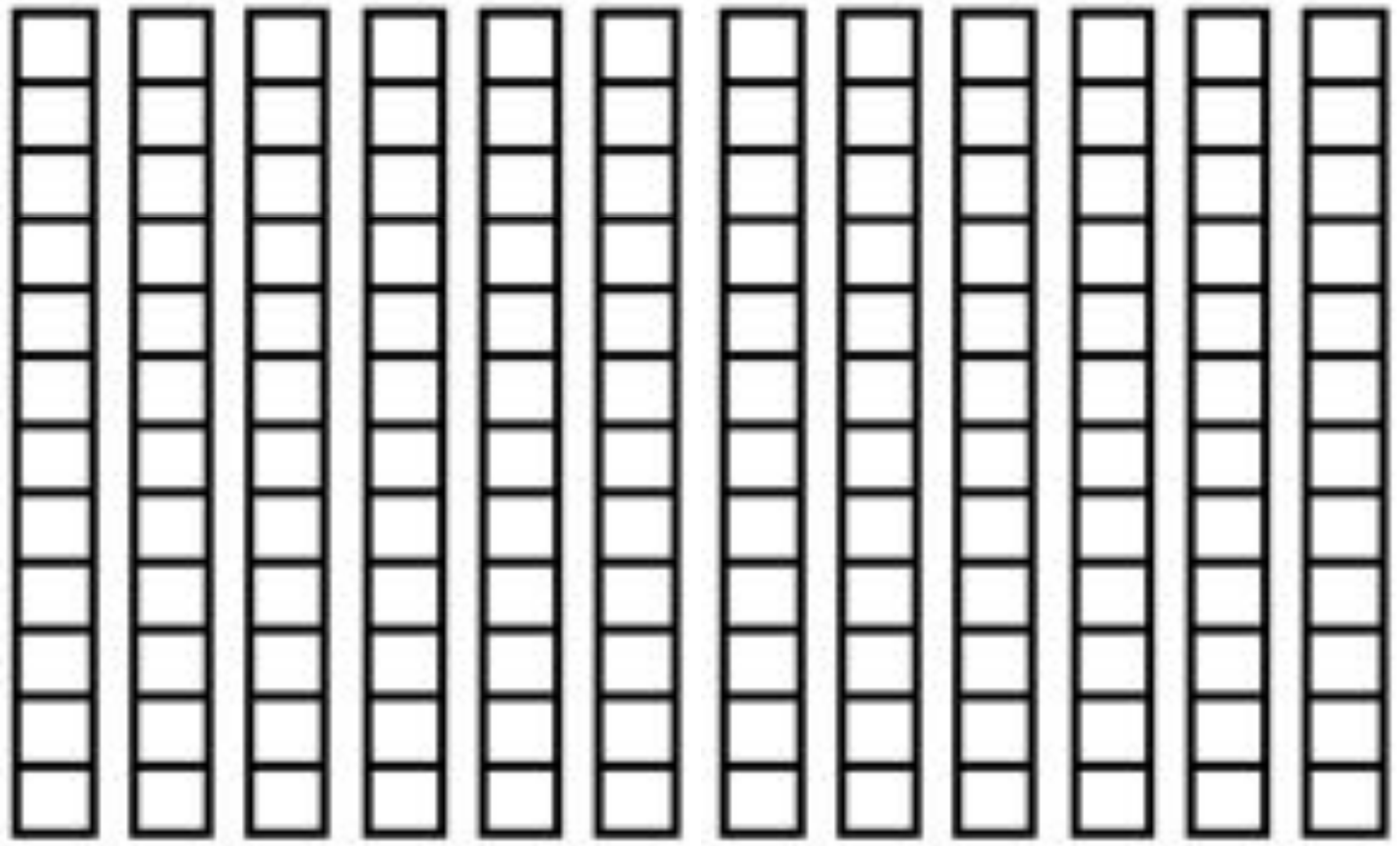
Putting it all together: text-to-speech with a neural network



Putting it all together: text-to-speech with a neural network

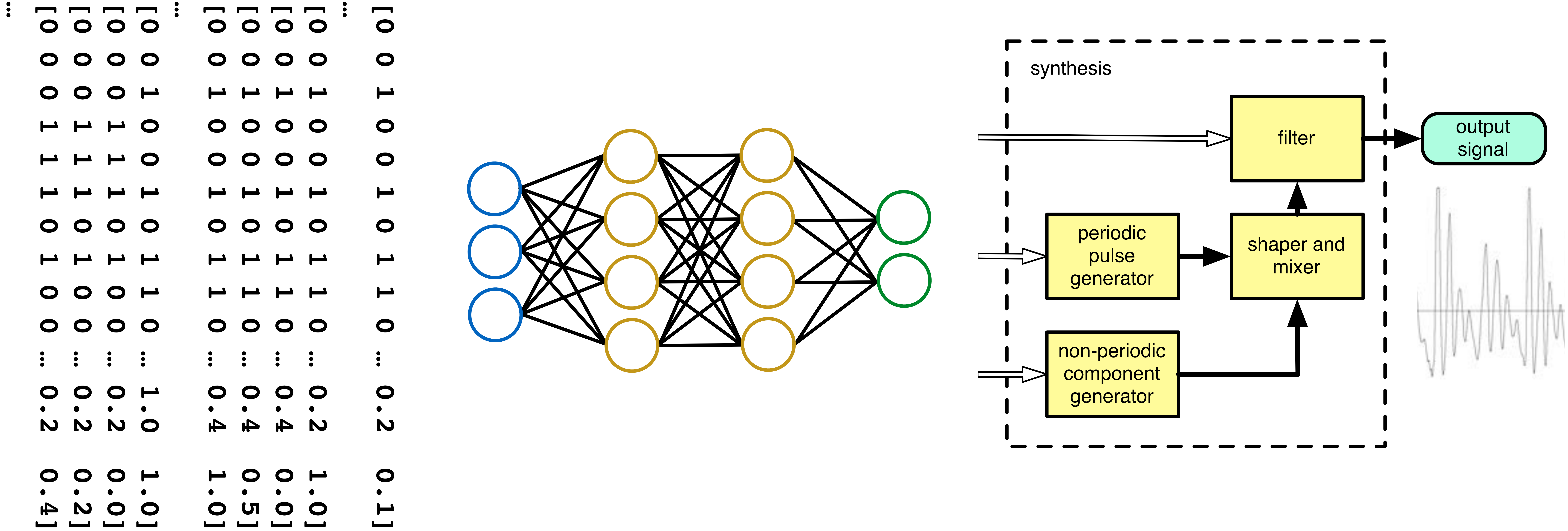


Putting it all together: text-to-speech with a neural network



```
[0 0 1 0 0 1 0 1 1 0 … 0.2  0.0]
[0 0 1 0 0 1 0 1 1 0 … 0.2  0.1]
…
[0 0 1 0 0 1 0 1 1 0 … 0.2  1.0]
[0 0 1 0 0 1 0 1 1 0 … 0.4  0.0]
[0 0 1 0 0 1 0 1 1 0 … 0.4  0.5]
[0 0 1 0 0 1 0 1 1 0 … 0.4  1.0]
…
[0 0 1 0 0 1 0 1 1 0 … 1.0  1.0]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.0]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.2]
[0 0 0 1 1 1 0 1 0 0 … 0.2  0.4]
…
```

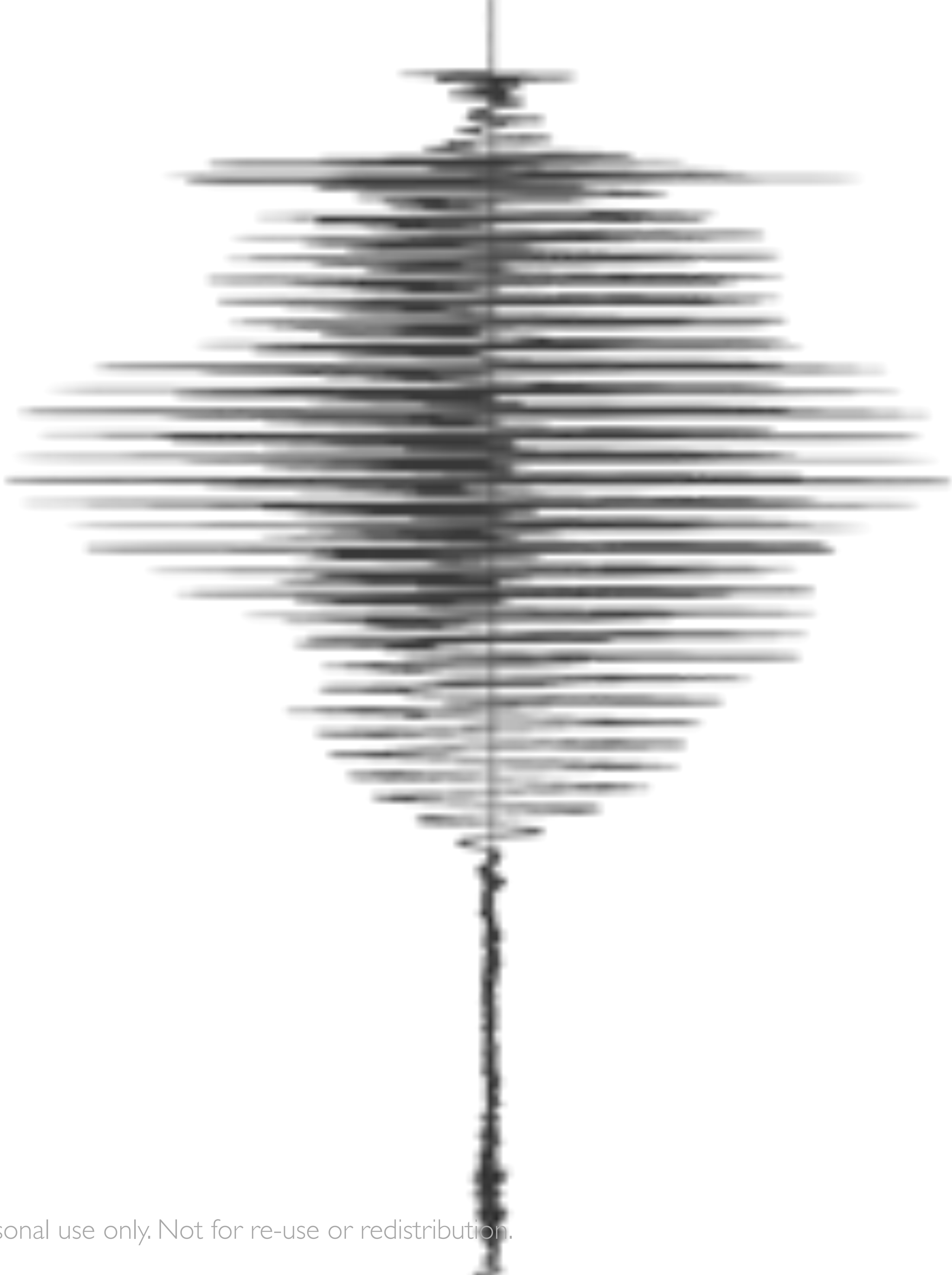
Putting it all together: text-to-speech with a neural network



Part 2 - Conventional signal processing for speech synthesis

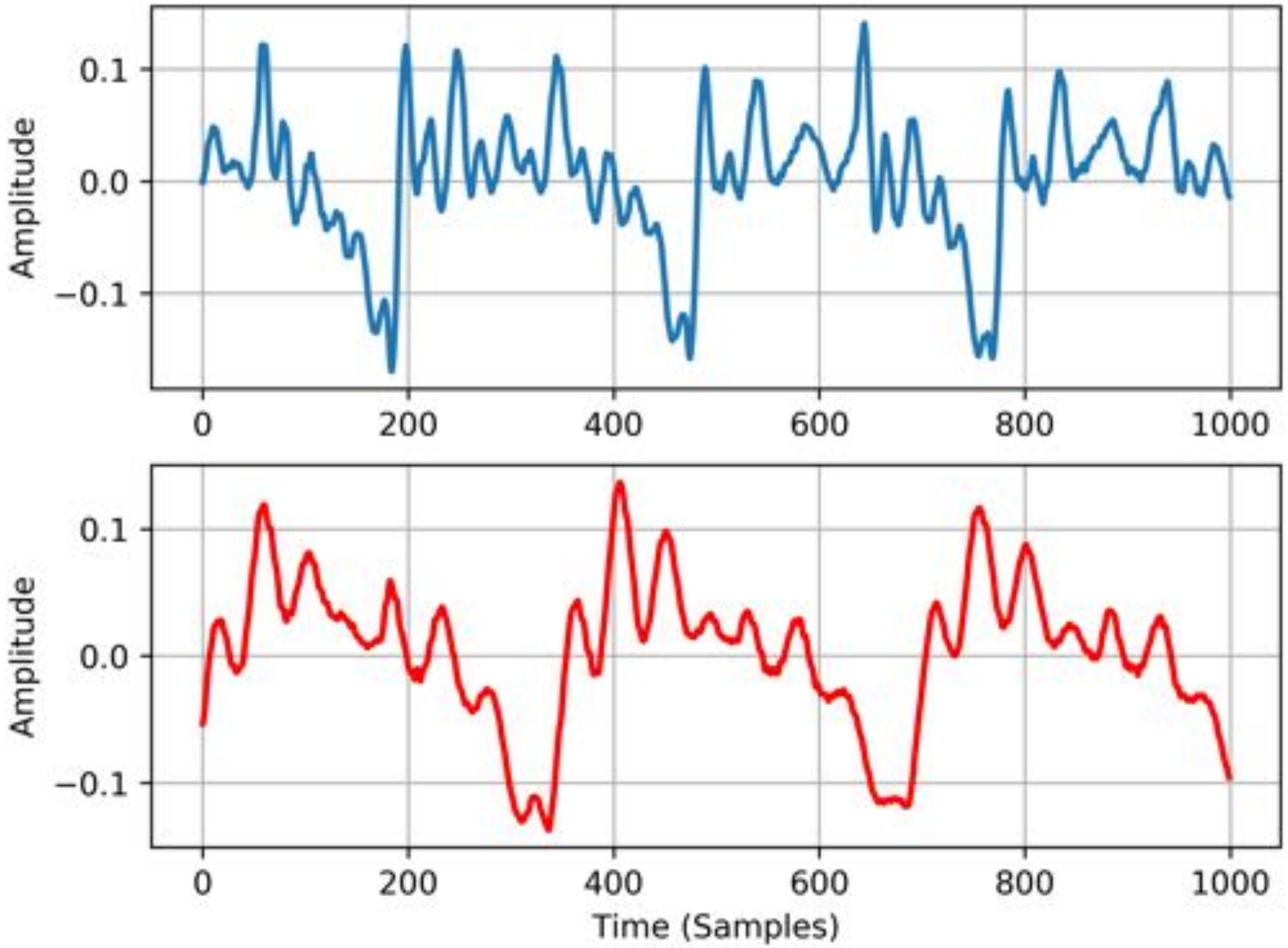
Signal processing for speech synthesis

- A typical vocoder: WORLD
  - Acoustic feature extraction
  - Feature engineering
  - Waveform generation



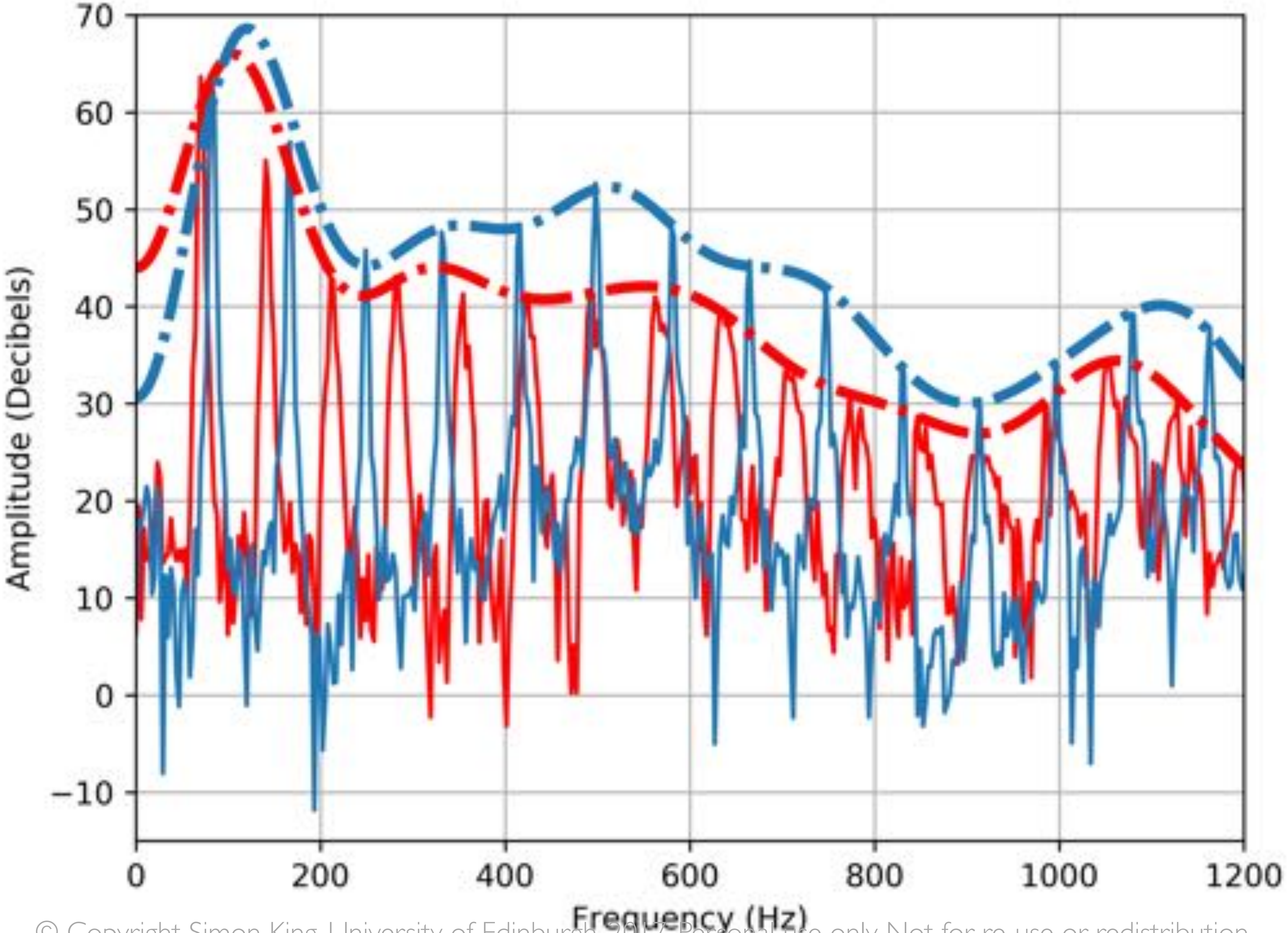
Why we use acoustic feature extraction - waveform

- Phoneme /a:/



Why we use acoustic feature extraction - magnitude spectrum

- Phoneme /a:/



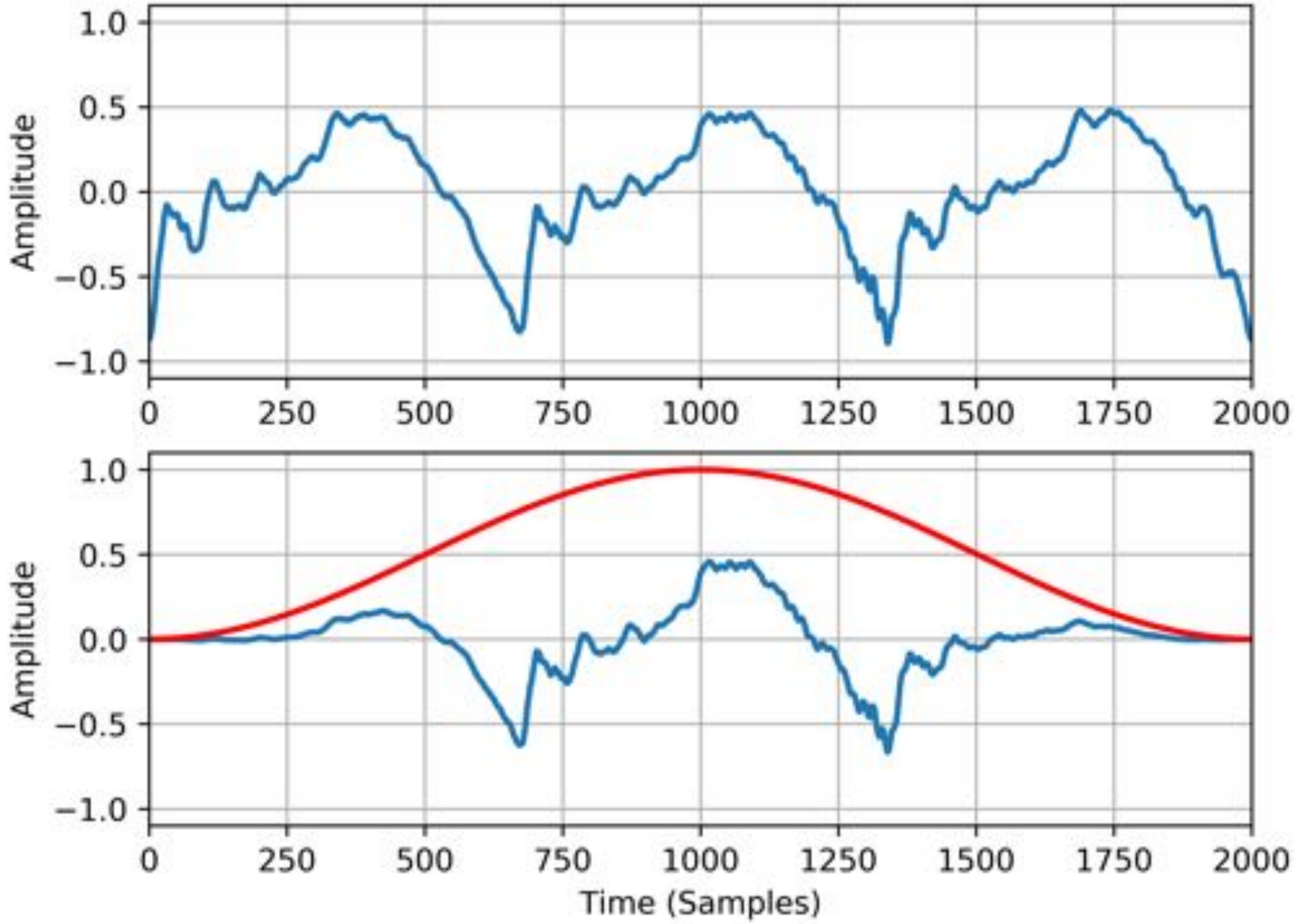
A typical vocoder: WORLD

- Developed by Masanori Morise since 2009
- Free and Open Source (modified BSD licence)
- Speech Features:
  - **Spectral Envelope** (estimated using CheapTrick)
  - **F0** (estimated using DIO)
  - **Band aperiodicities** (estimated using D4C)

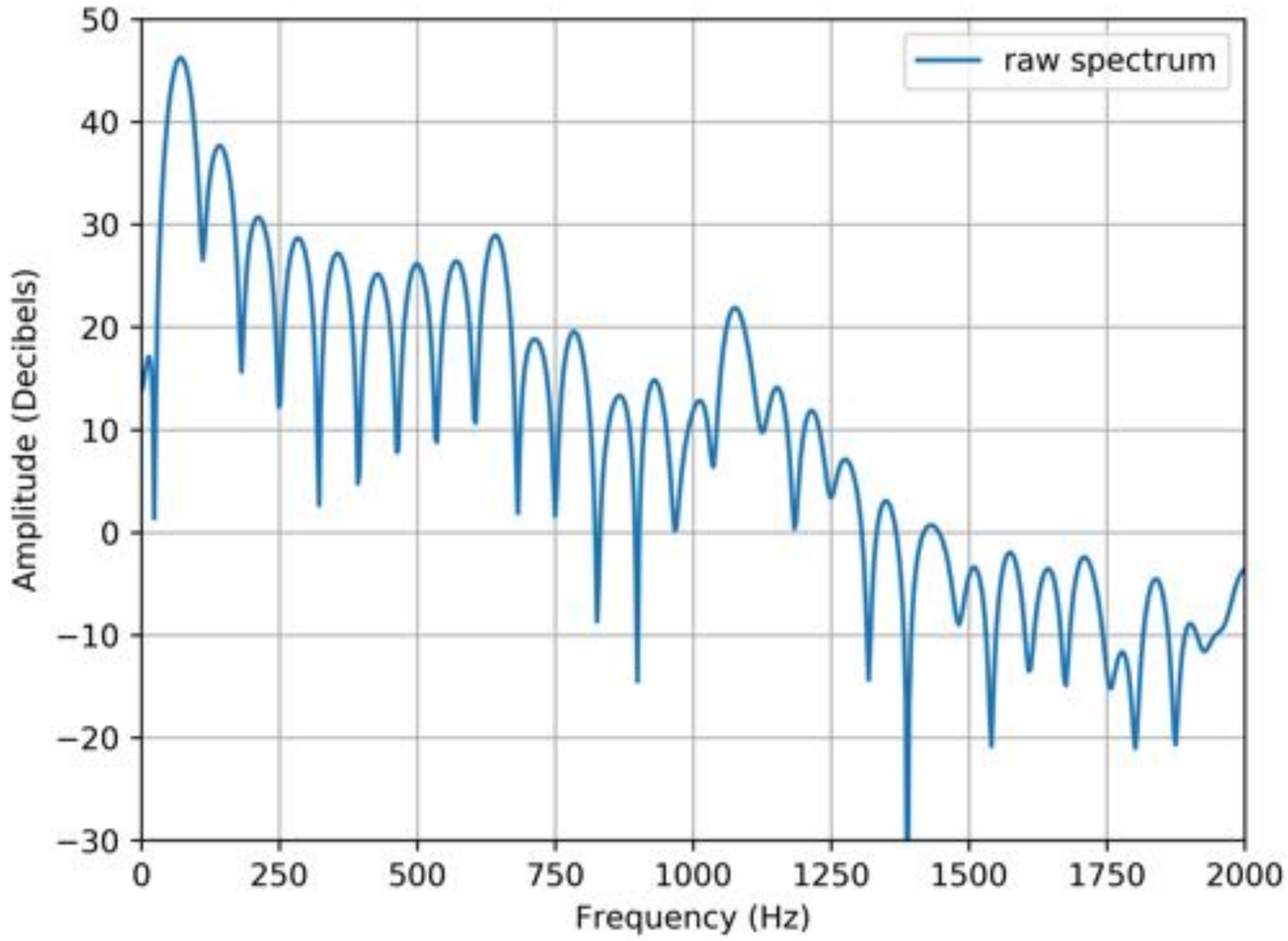
WORLD: spectral envelope estimation

- Hanning window length 3 T0

Power is temporally stable

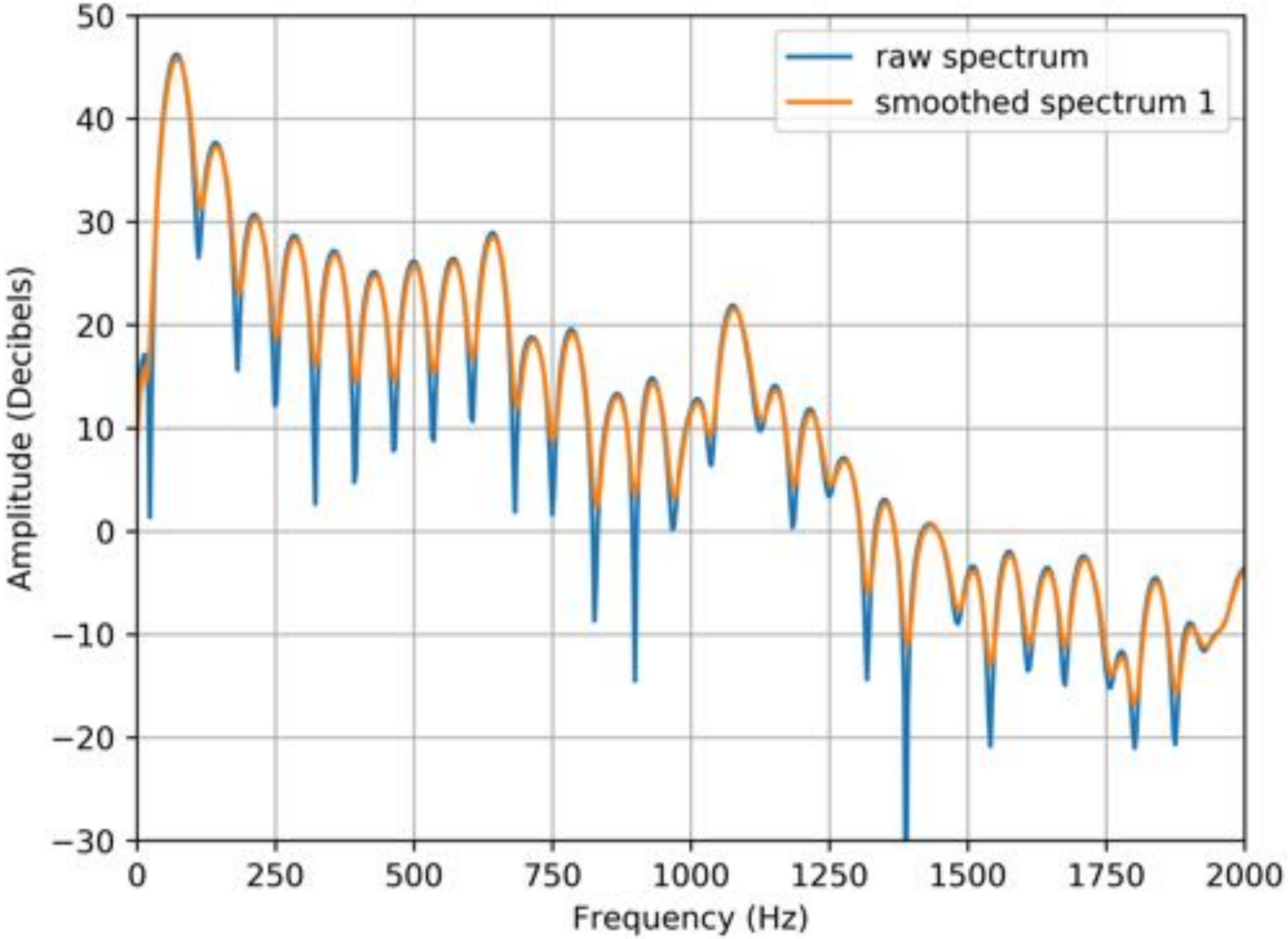


WORLD: spectral envelope estimation



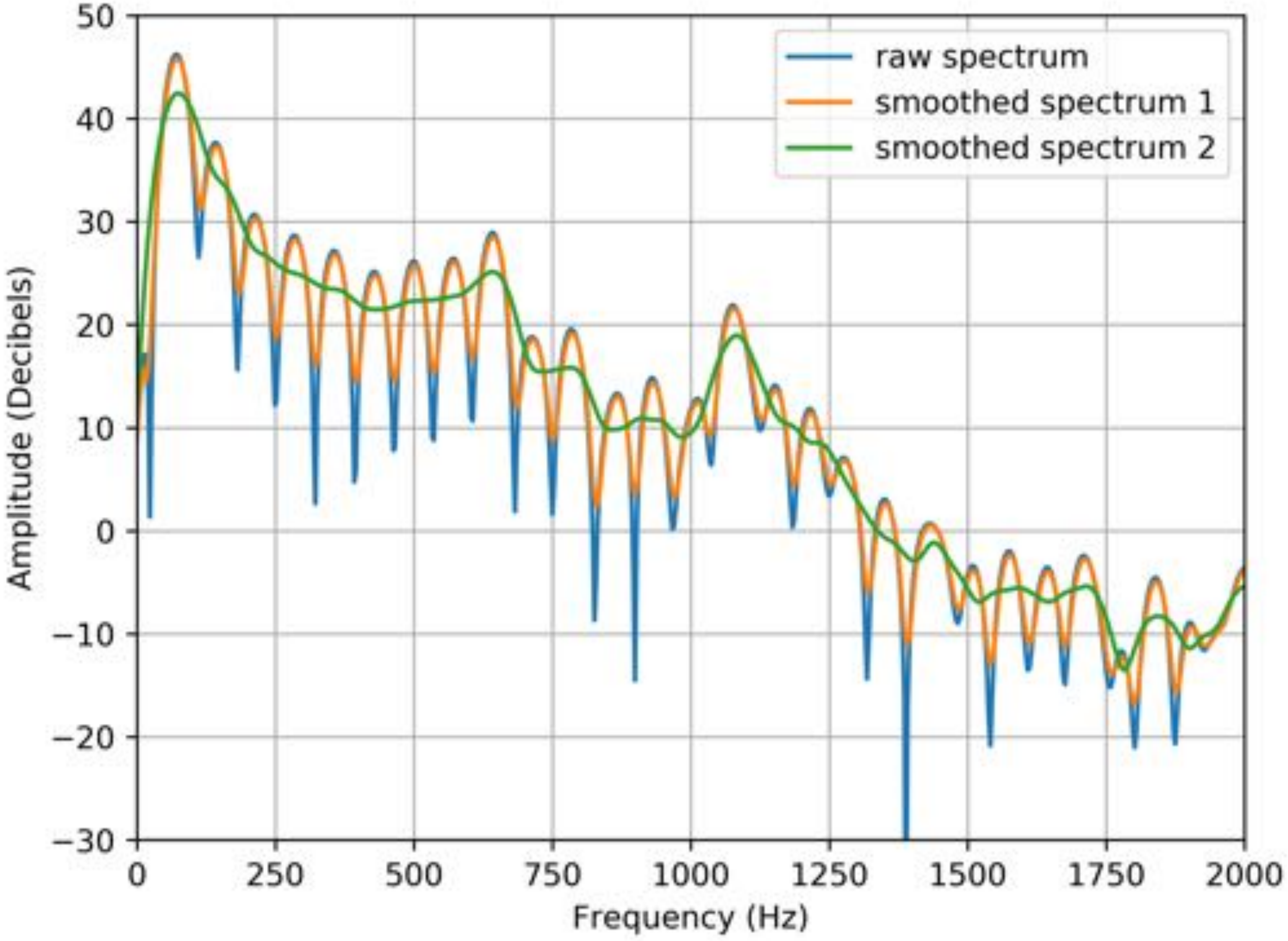
WORLD: spectral envelope estimation

- Apply a moving average filter
  - length (2/3) F0



WORLD: spectral envelope estimation

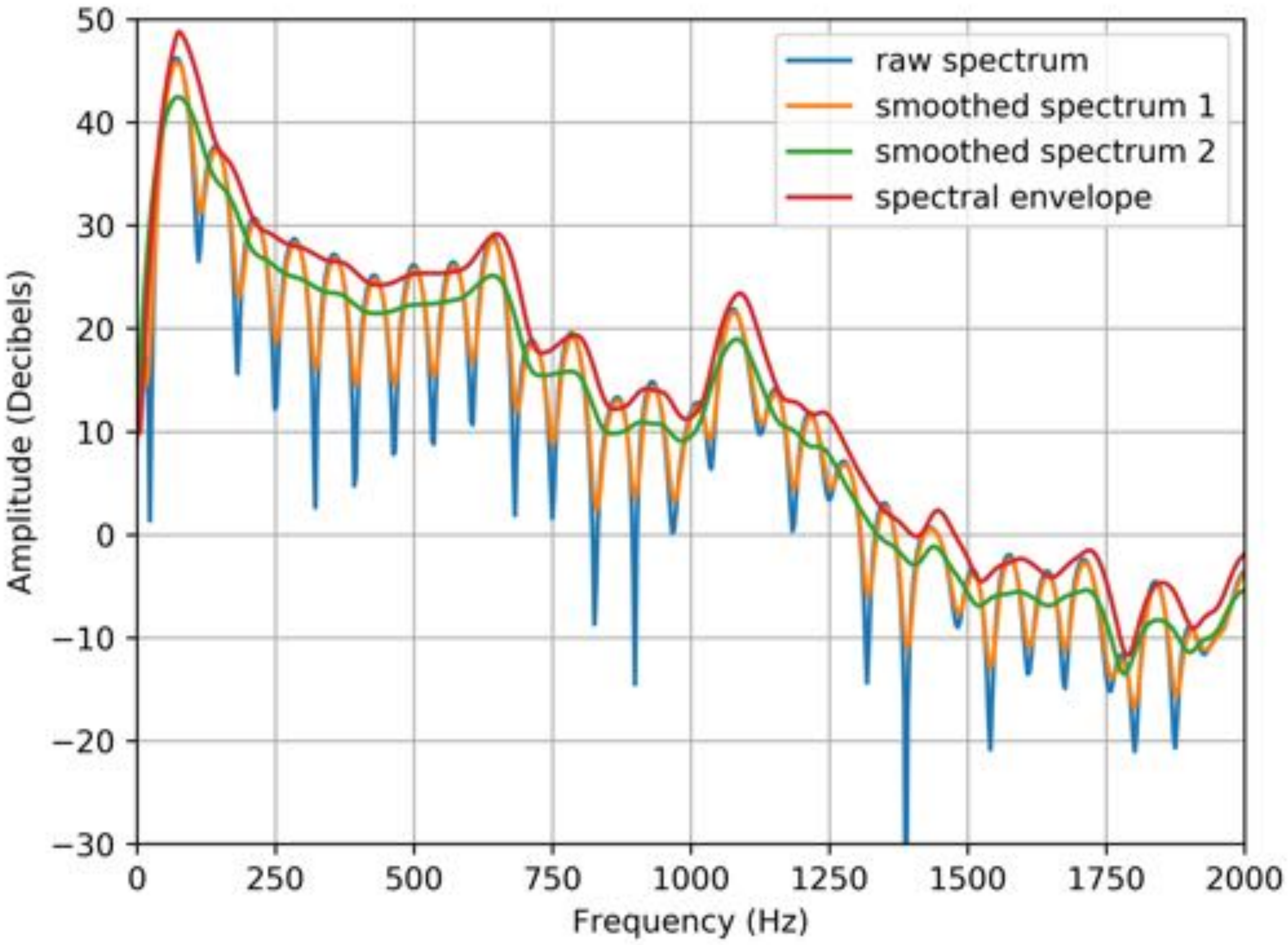
- Apply another moving average filter
  - length 2 F0



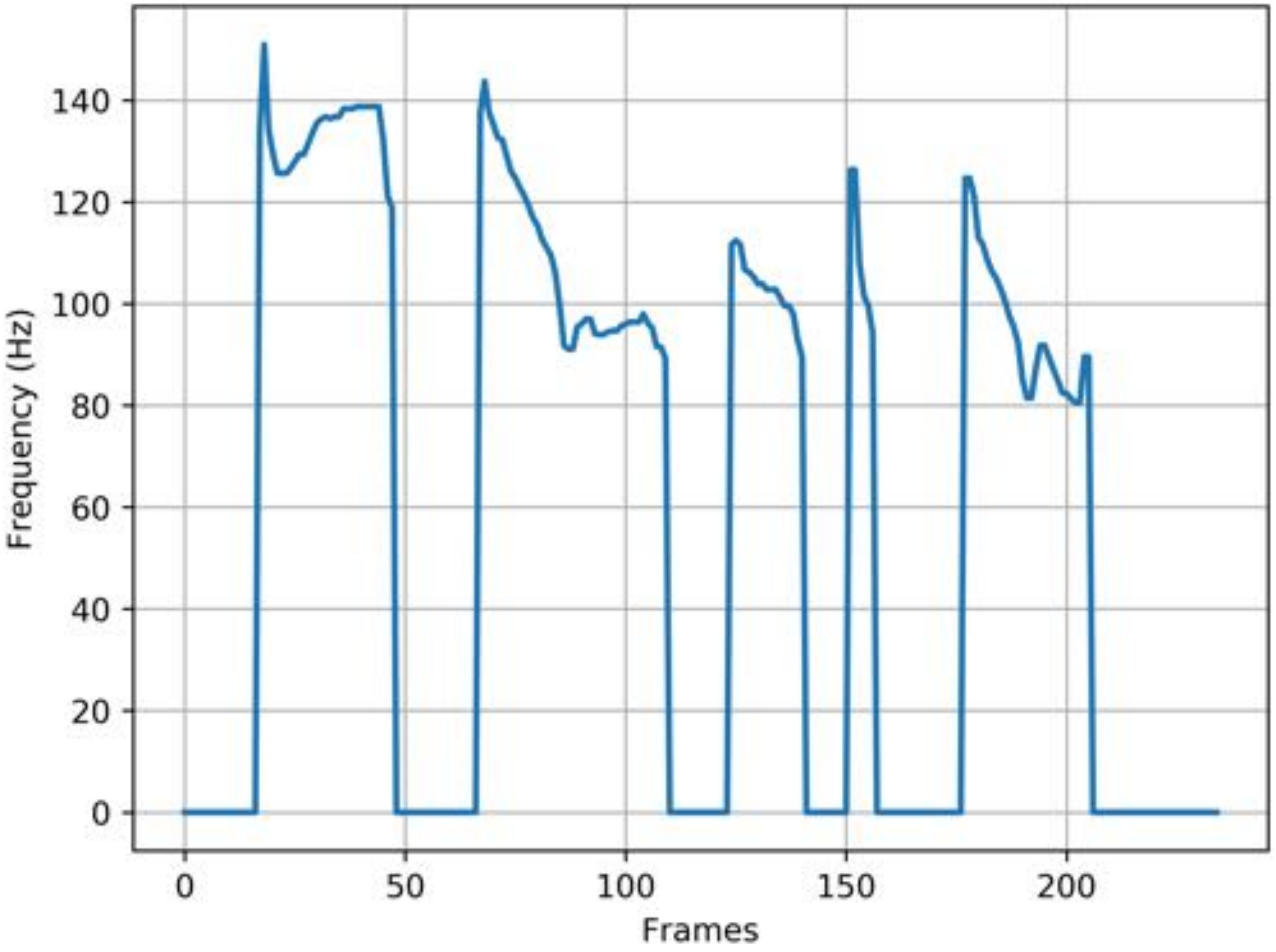
WORLD: spectral envelope estimation

- SpEnv = q0 logSp(F) + q1 logSp(F+F0)+q1 logSp(F-F0)

- *actually done in the cepstral domain*
- *illustrated here in the spectral domain*

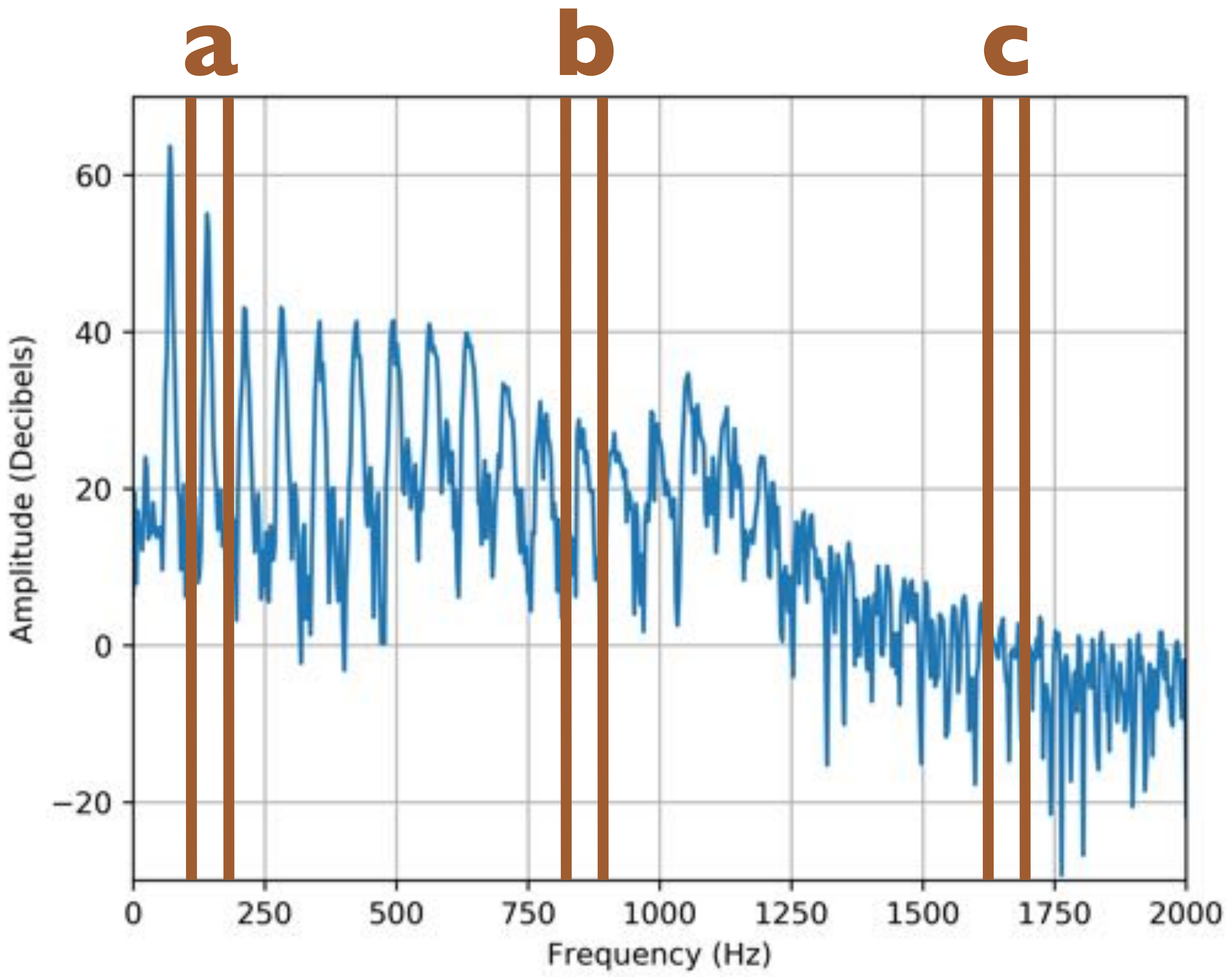


WORLD: F0 estimation



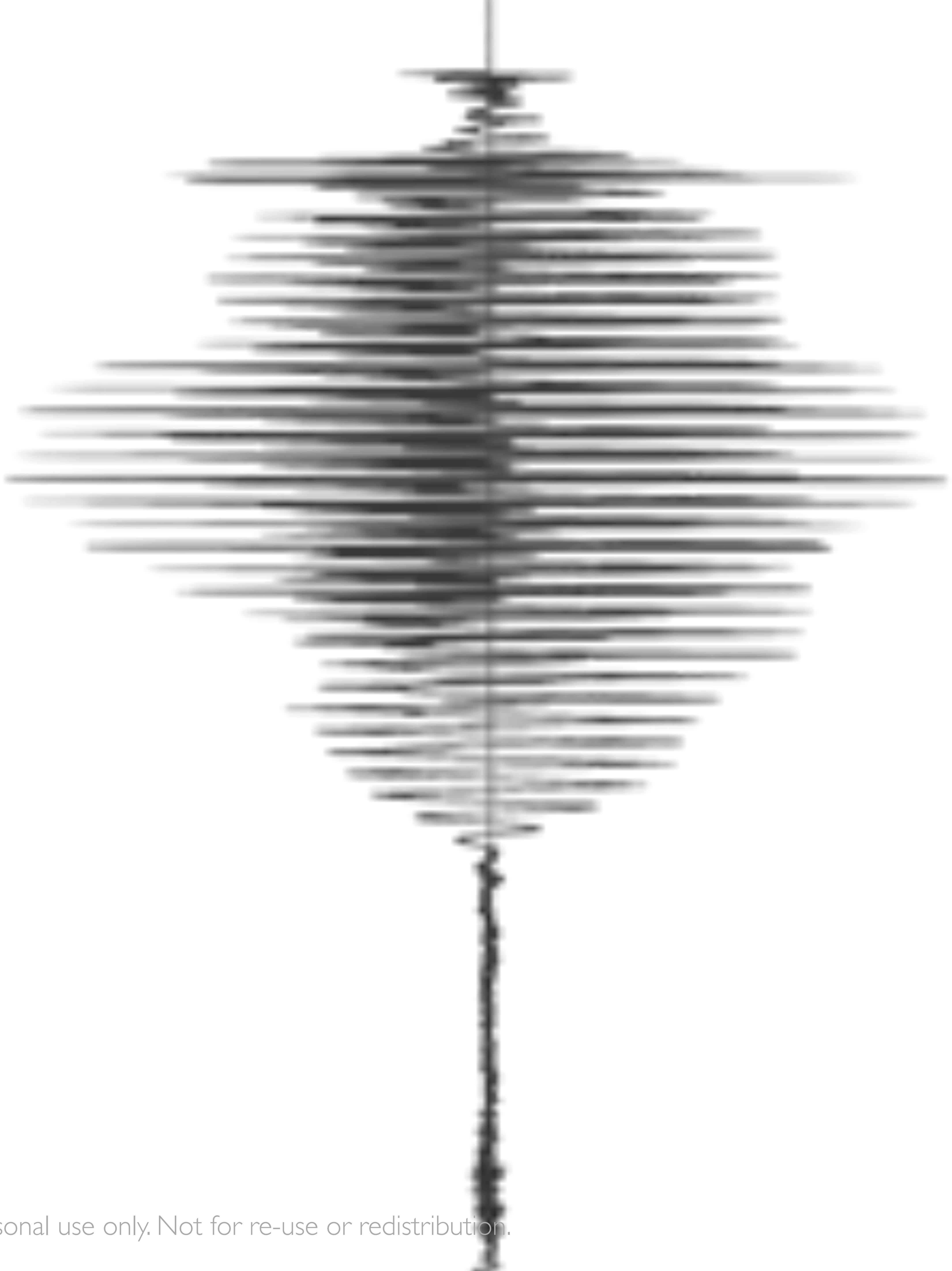
WORLD: band aperiodicities

- The **ratio** between aperiodic and periodic energy, averaged over certain frequency bands
- i.e., total power / sine wave power
- In the example, this ratio is
  - lowest in band **a**
  - more in band **b**
  - highest in band **c**

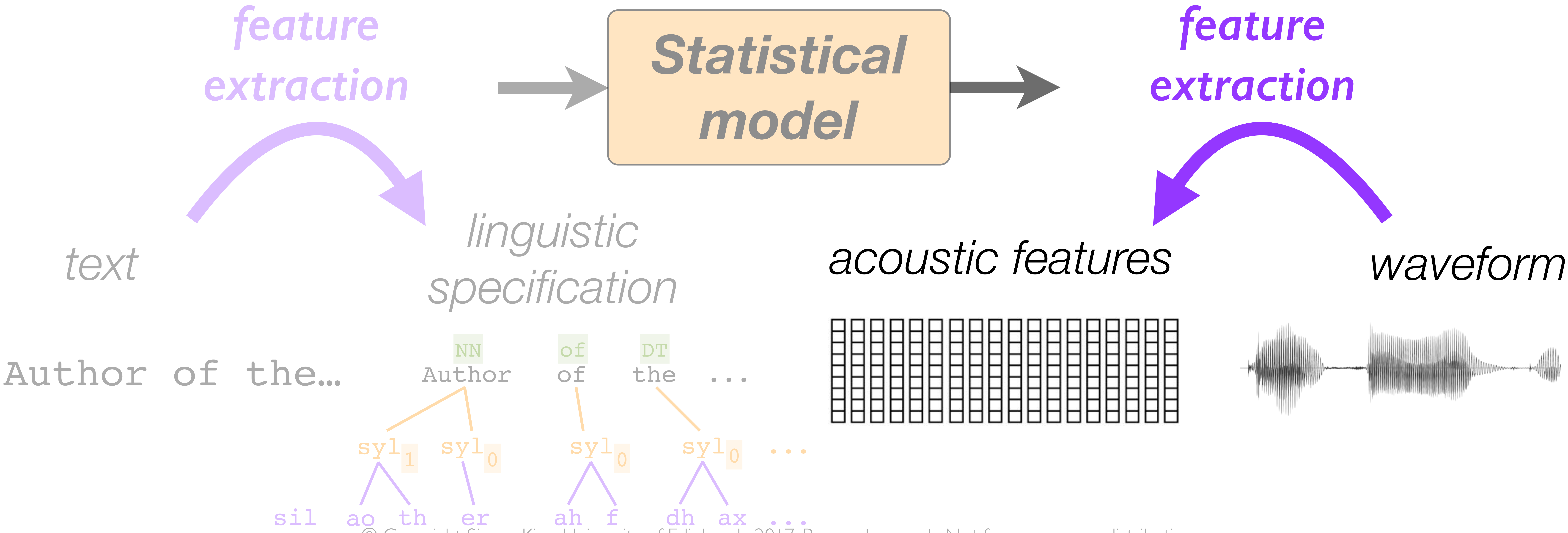


Signal processing for speech synthesis

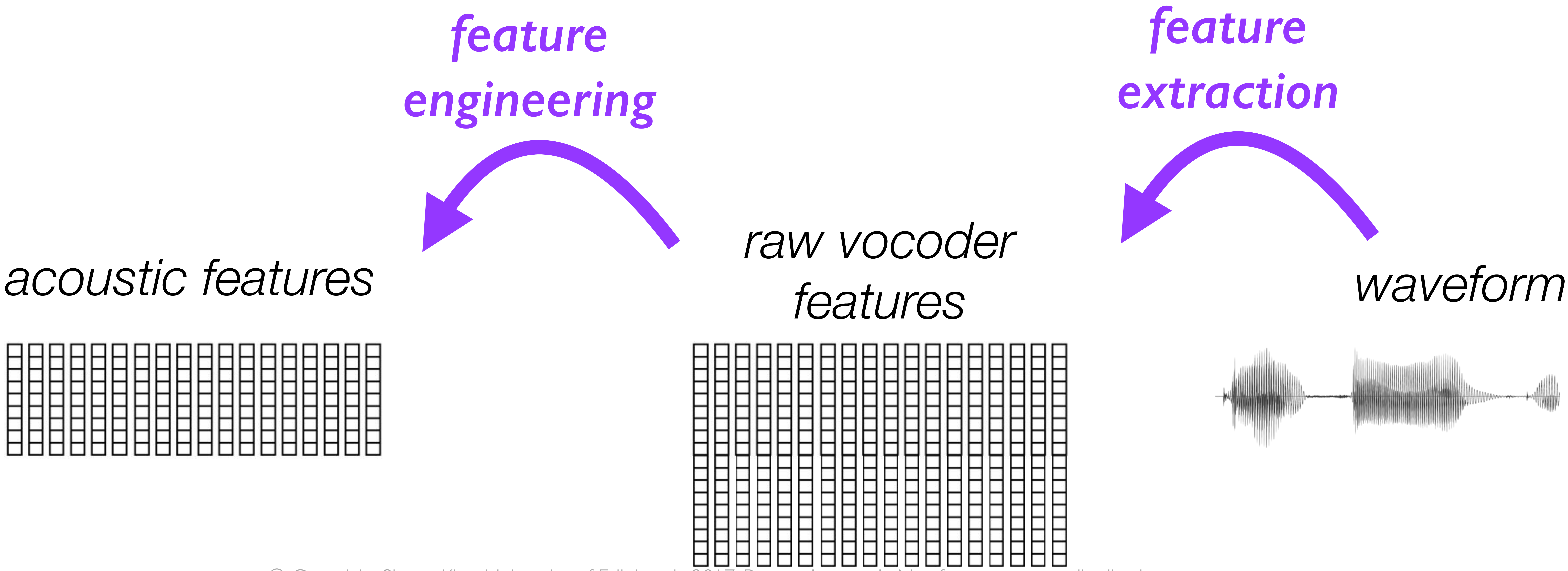
- A typical vocoder: WORLD
  - Acoustic feature extraction
  - Feature engineering
  - Waveform generation



Acoustic feature extraction

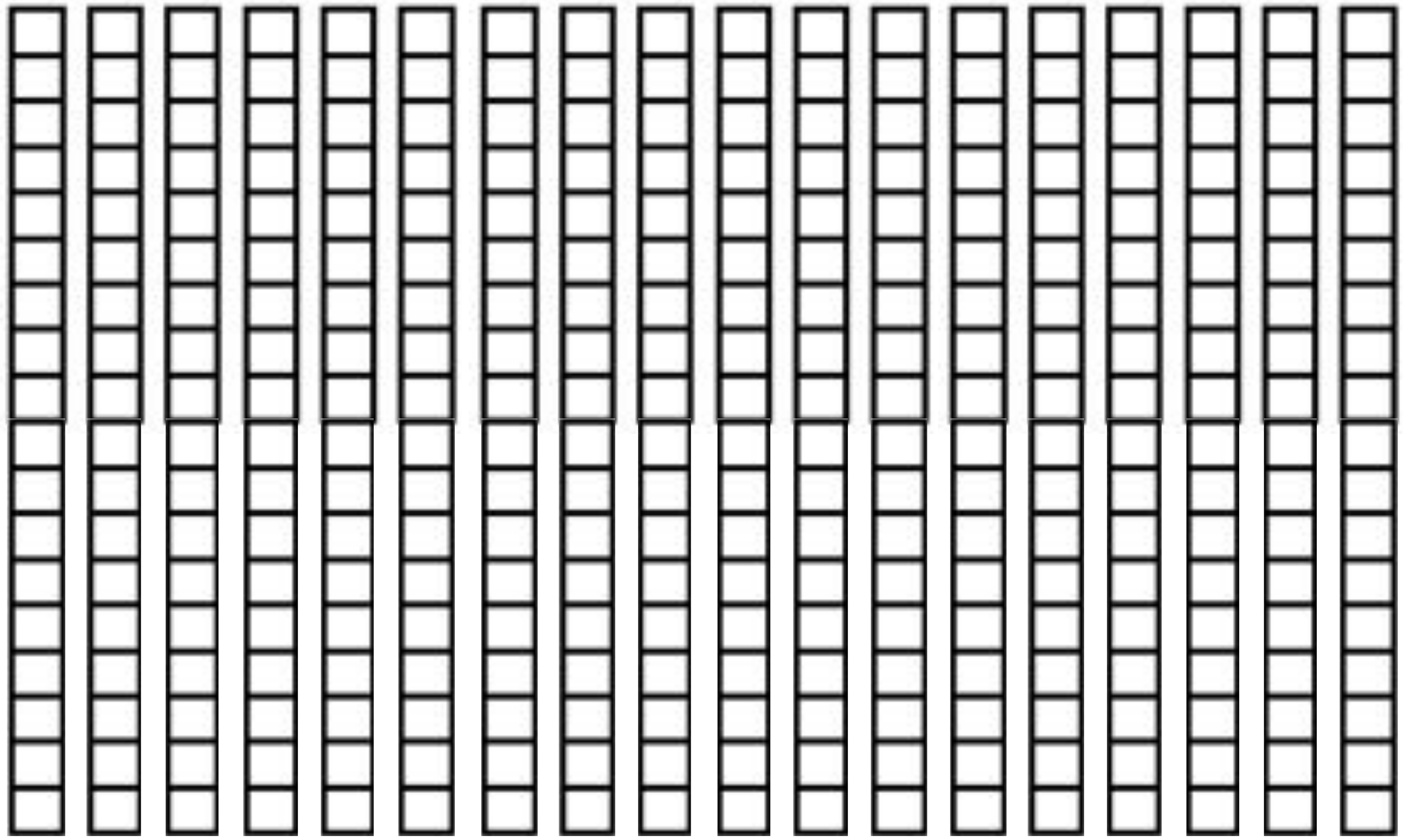


Acoustic feature extraction & engineering

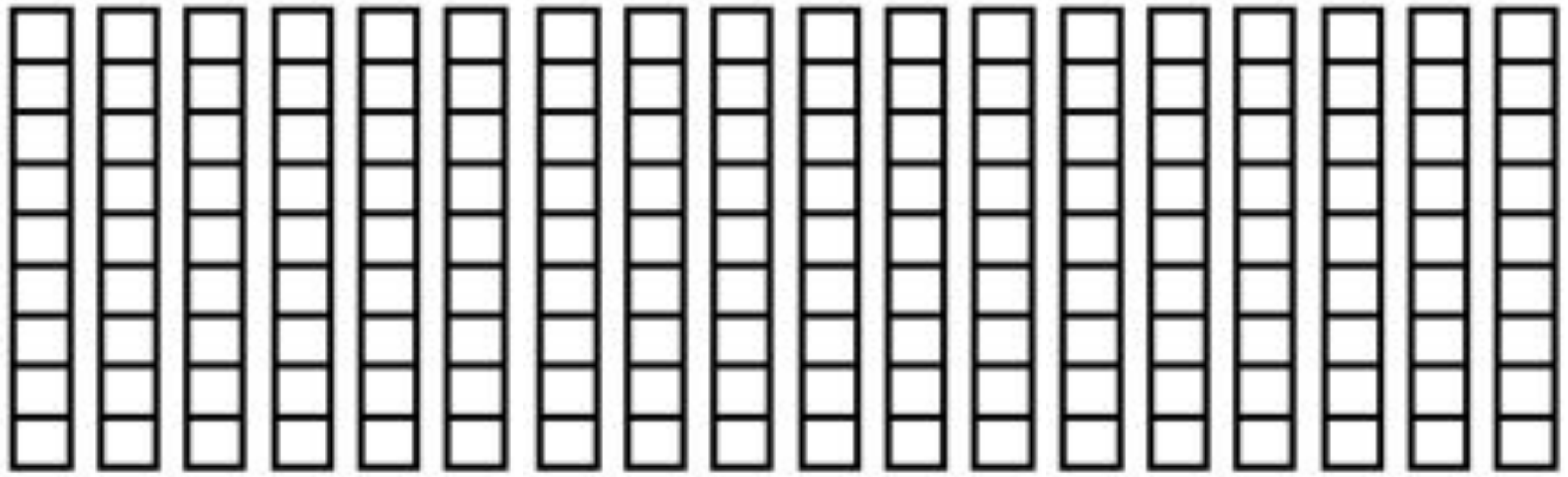


Acoustic feature engineering

raw vocoder features



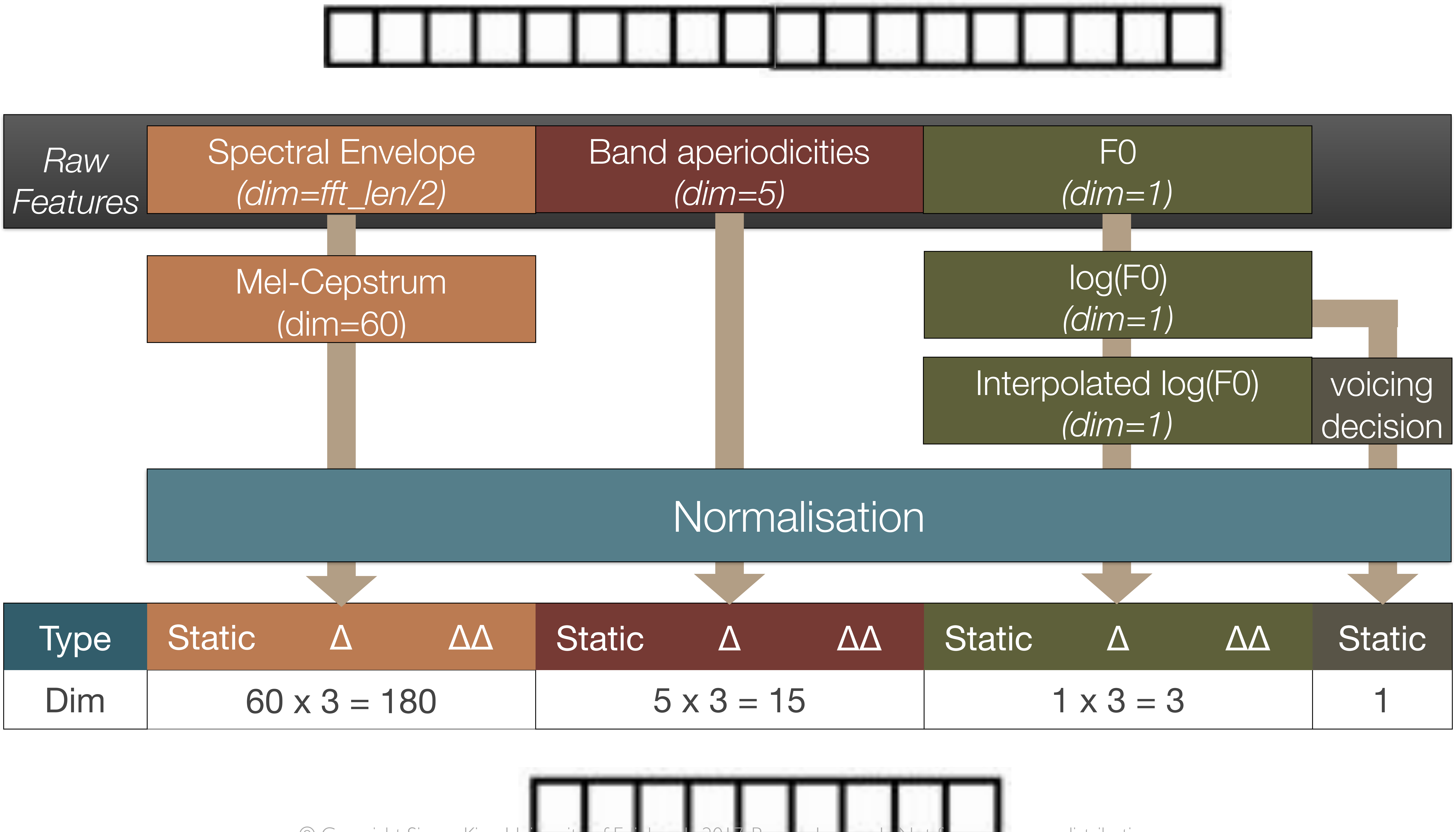
acoustic features



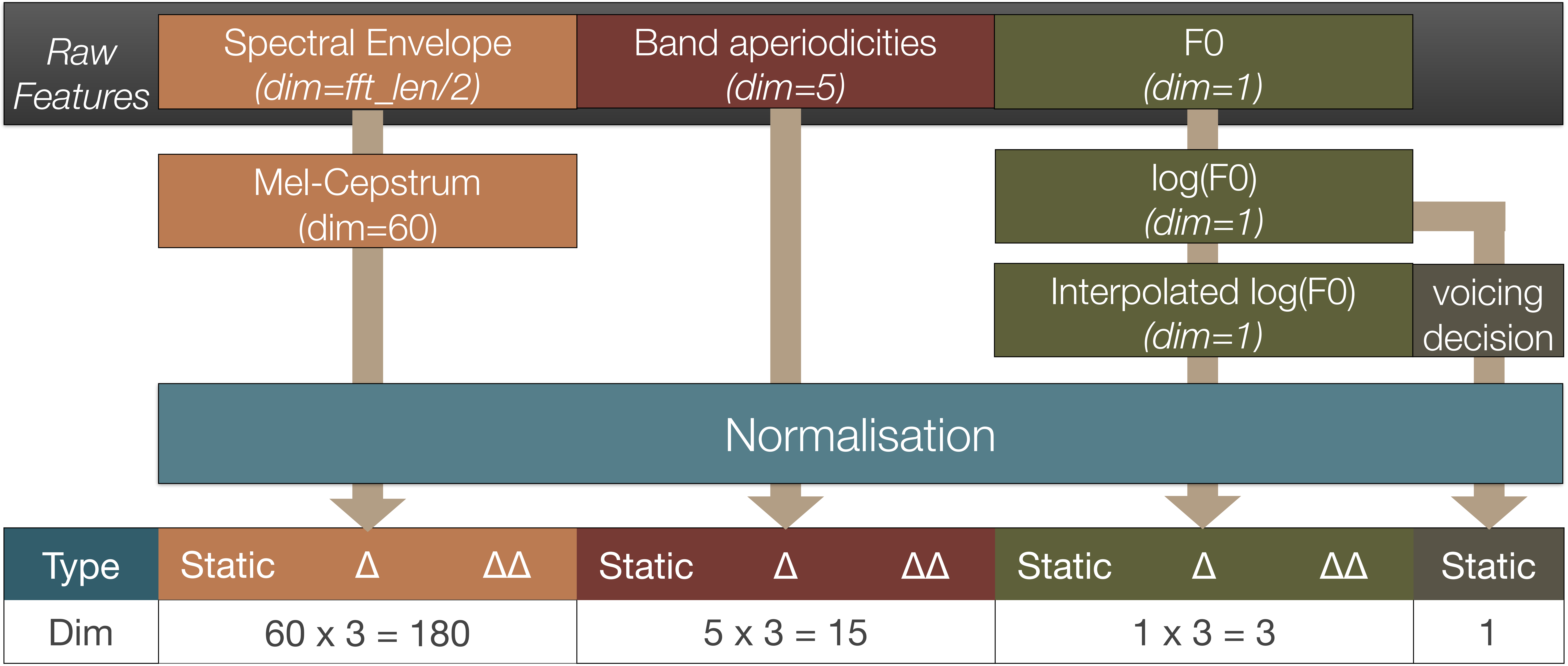
Acoustic feature engineering

raw vocoder features

acoustic features

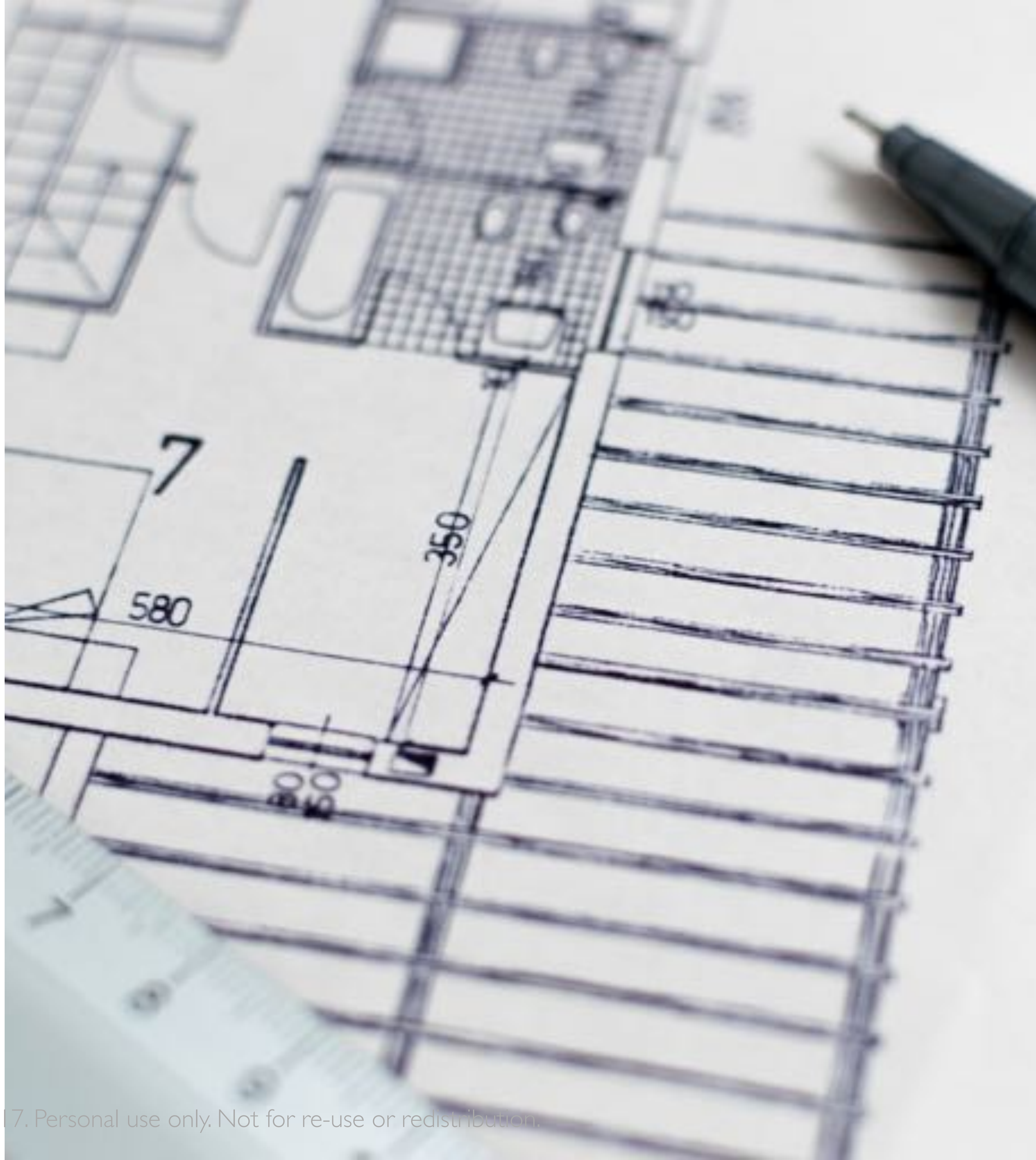

Raw Features
Spectral Envelope (dim=fft_len/2)
Band aperiodicities (dim=5)
F0 (dim=1)
Mel-Cepstrum (dim=60)
log(F0) (dim=1)
Interpolated log(F0) (dim=1)
voicing decision
Normalisation
Type
Static
Δ
ΔΔ
Static
Δ
ΔΔ
Static
Δ
ΔΔ
Static
Dim
60 x 3 = 180
5 x 3 = 15
1 x 3 = 3
1

Acoustic feature engineering



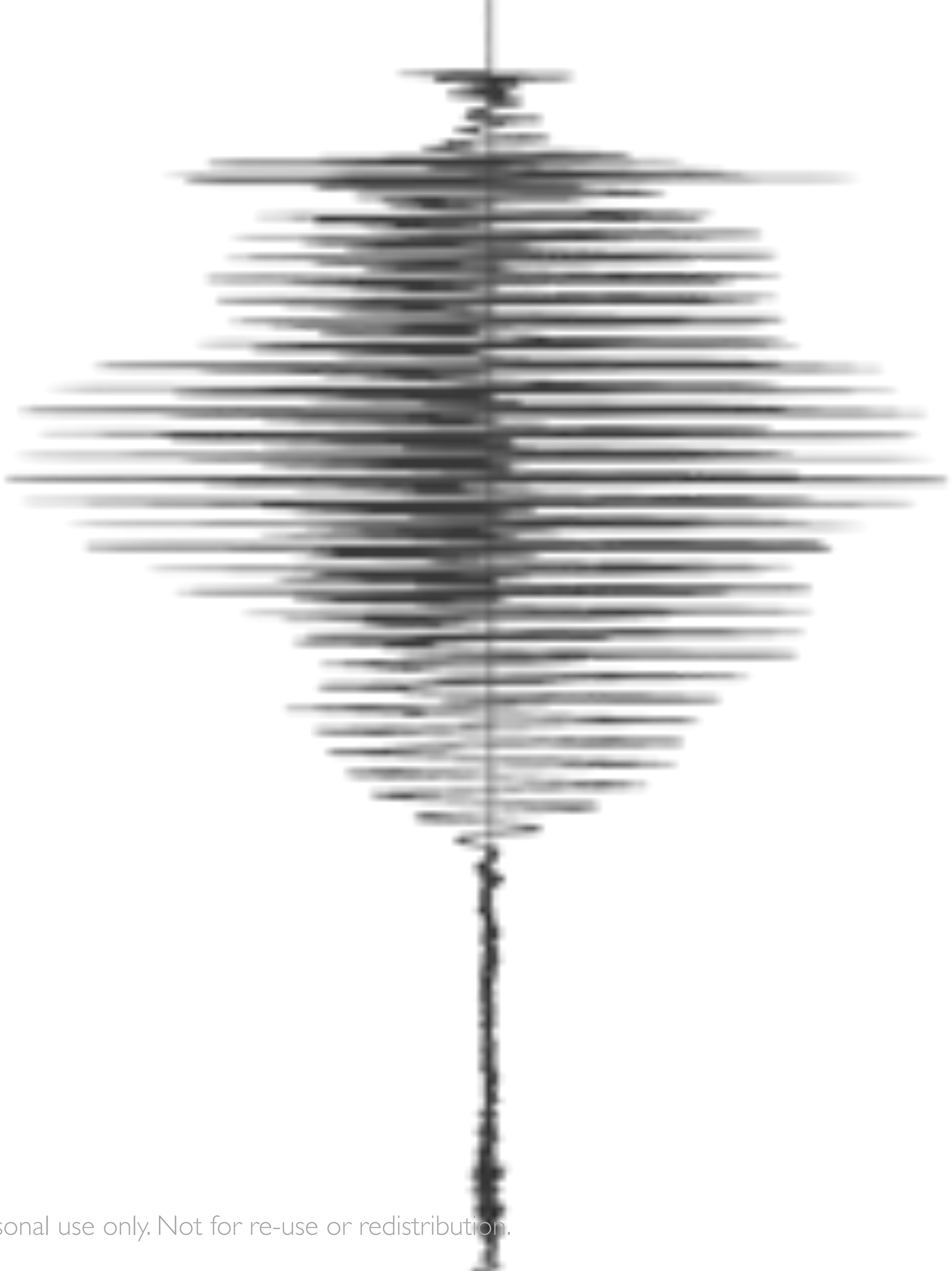
Design choices: acoustic features

- fixed framerate *or* pitch synchronous
- cepstrum *or* spectrum
- linear *or* warped frequency (e.g., Mel)
- order
- interpolate F0
- phase modelling
  - no: e.g., Tacotron
  - yes: e.g., Espic, Valentini-Botinhao, King, Interspeech 2017



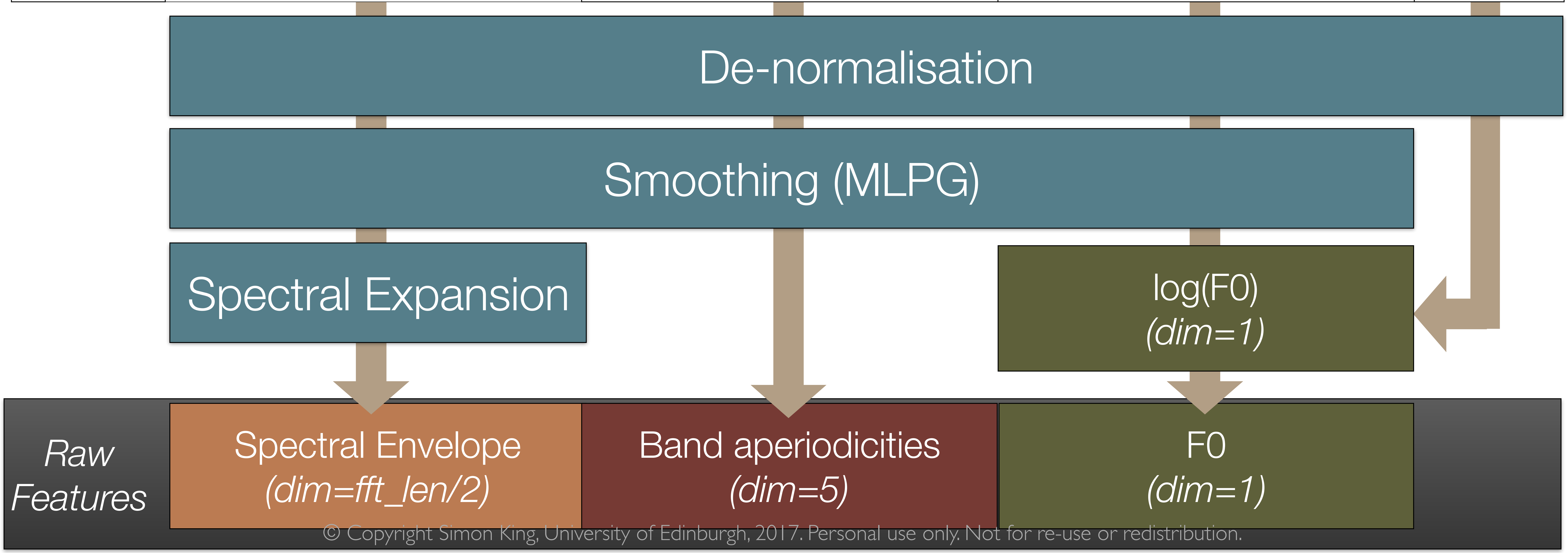
Signal processing for speech synthesis

- A typical vocoder: WORLD
  - Acoustic feature extraction
  - Feature engineering
  - Waveform generation



From acoustic features back to raw vocoder features

Feat Type	Mel-Cepstrum			Band aperiodicities			Interpolated log(F0)			voicing
	Static	Δ	ΔΔ	Static	Δ	ΔΔ	Static	Δ	ΔΔ	Static
Dim	60 x 3 = 180			5 x 3 = 15			1 x 3 = 3			1



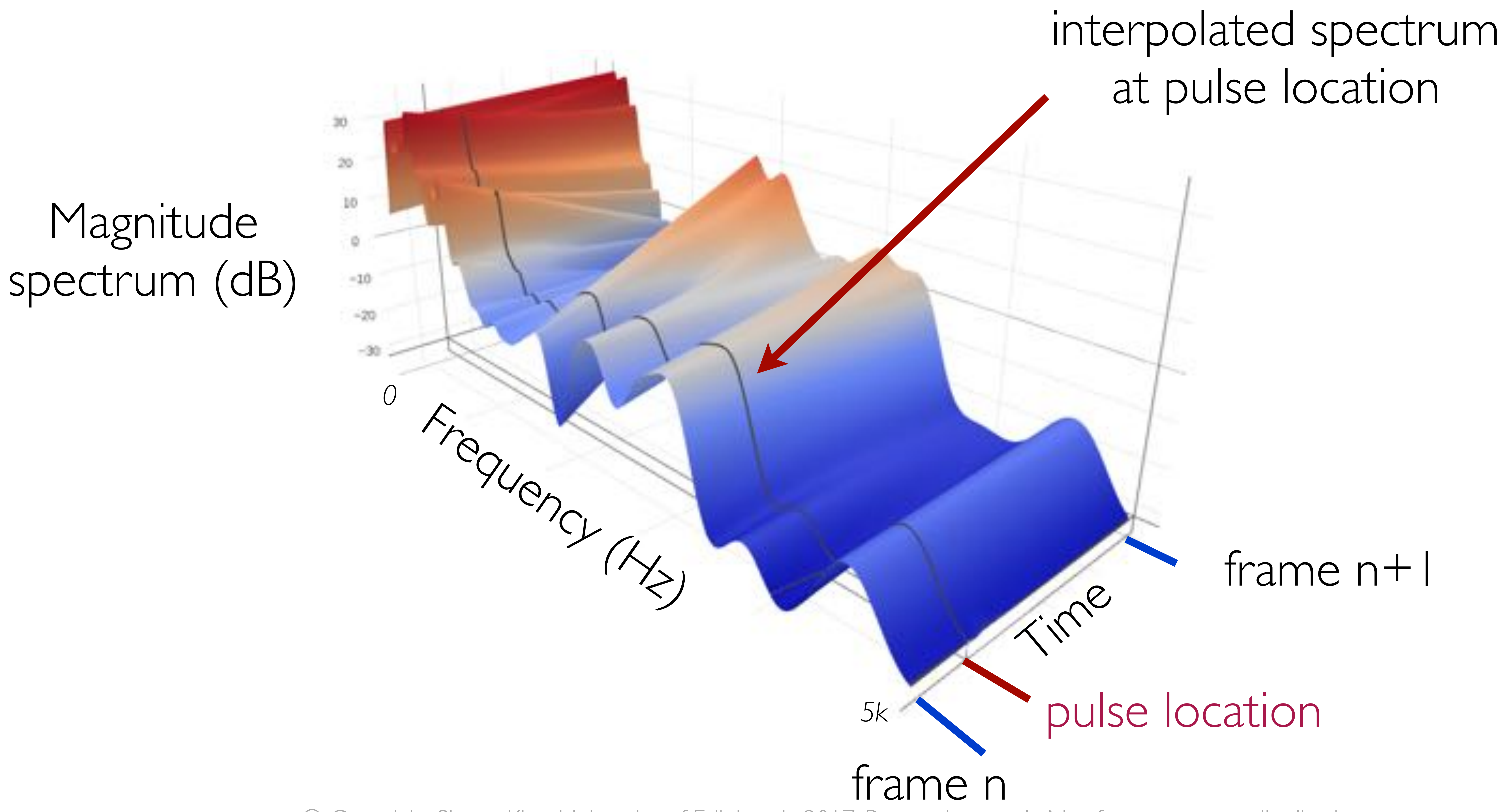
WORLD: periodic excitation using a pulse train

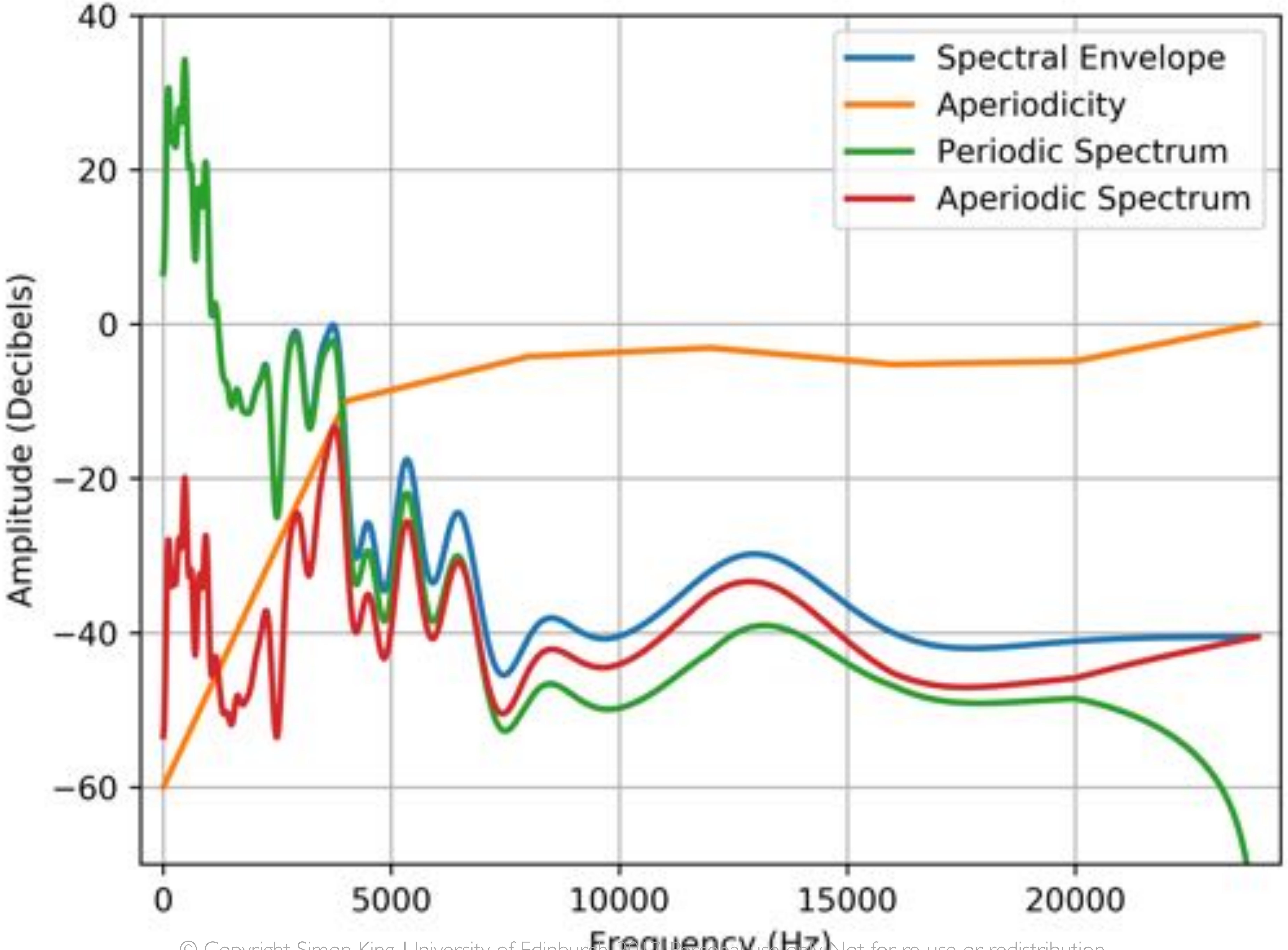
- Computation of pulse locations

  - Voiced segments: create one pulse every **fundamental period**, T0
    - calculate T0 from F0, which has been predicted by the acoustic model

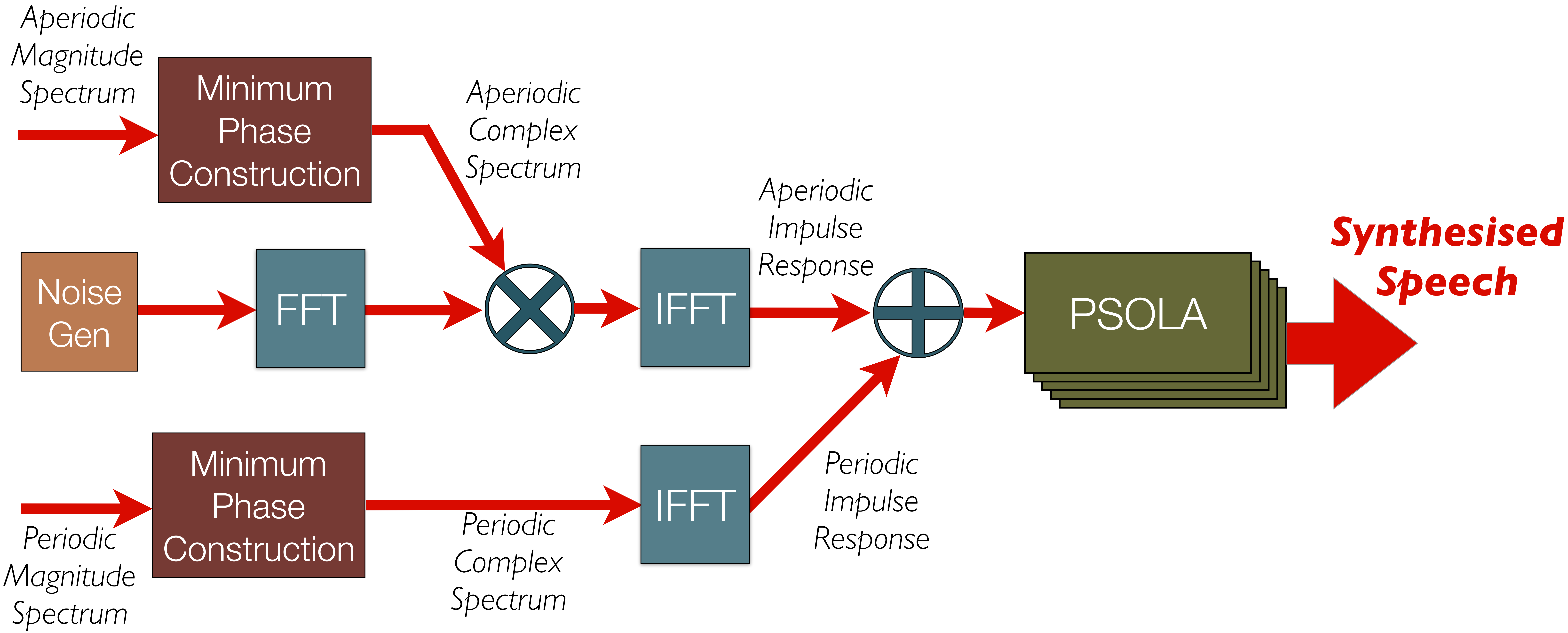
  - Unvoiced segments: fixed rate T0 = 5ms

WORLD: obtain spectral envelope at exact pulse locations, by interpolation



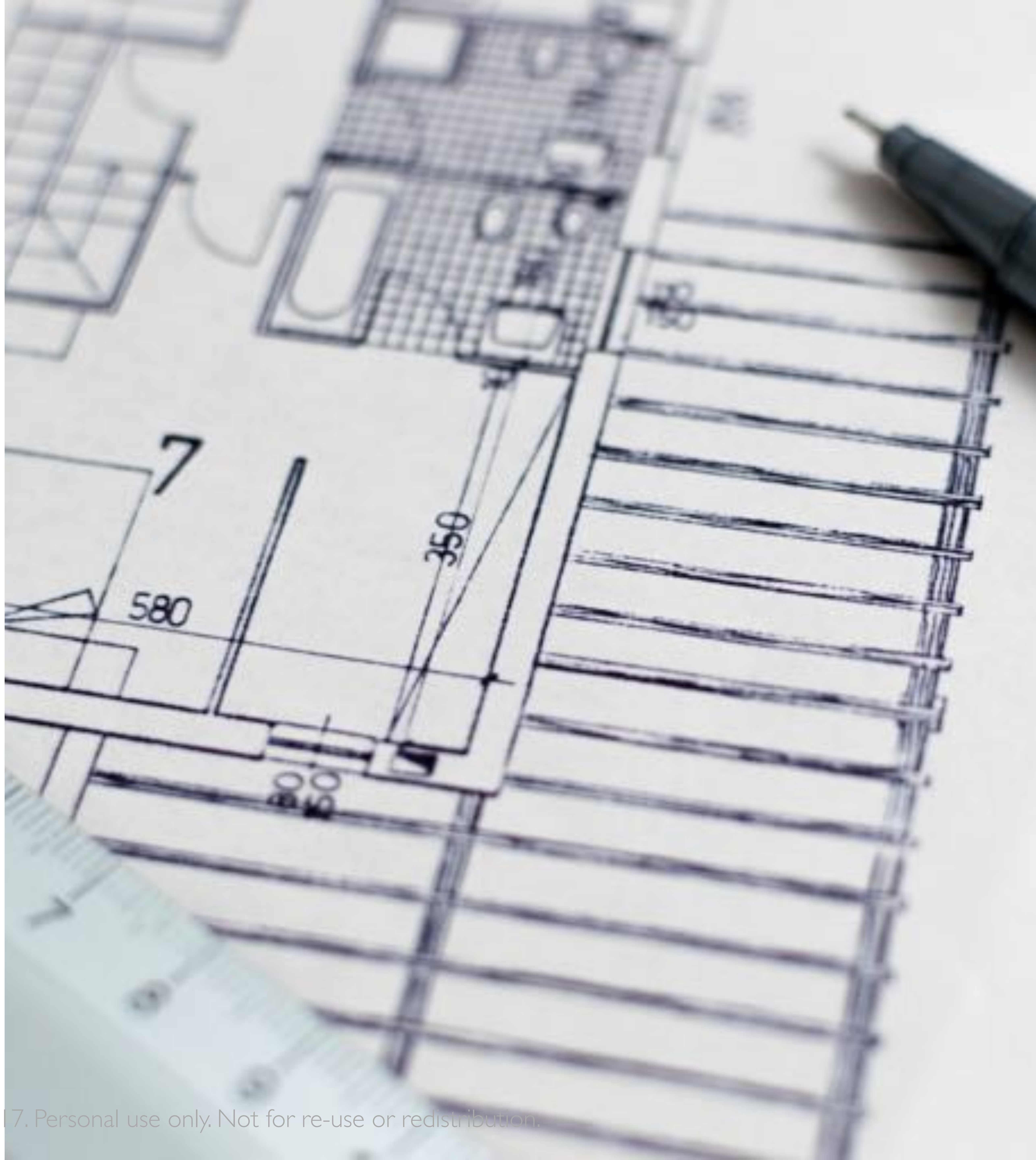

Spectral Envelope
Aperiodicity
Periodic Spectrum
Aperiodic Spectrum
Amplitude (Decibels)
40
20
0
−20
−40
−60
0
5000
10000
15000
20000
Frequency (Hz)

WORLD: generate waveform



Design choices: waveform generation

- fixed framerate *or* pitch synchronous
  - may be different from what you used in acoustic feature extraction
- cepstrum *or* spectrum
- source
  - pulse/noise *or* mixed *or* sampled
- phase
  - synthetic (e.g., pulse train + minimum phase filter) *or*
  - predict using acoustic model



Examples

System	feedforward	BLSTM
Merlin + WORLD		
Merlin + STRAIGHT		

So, what happened next … ?

arXiv:1609.03499 (unreviewed manuscript)

WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

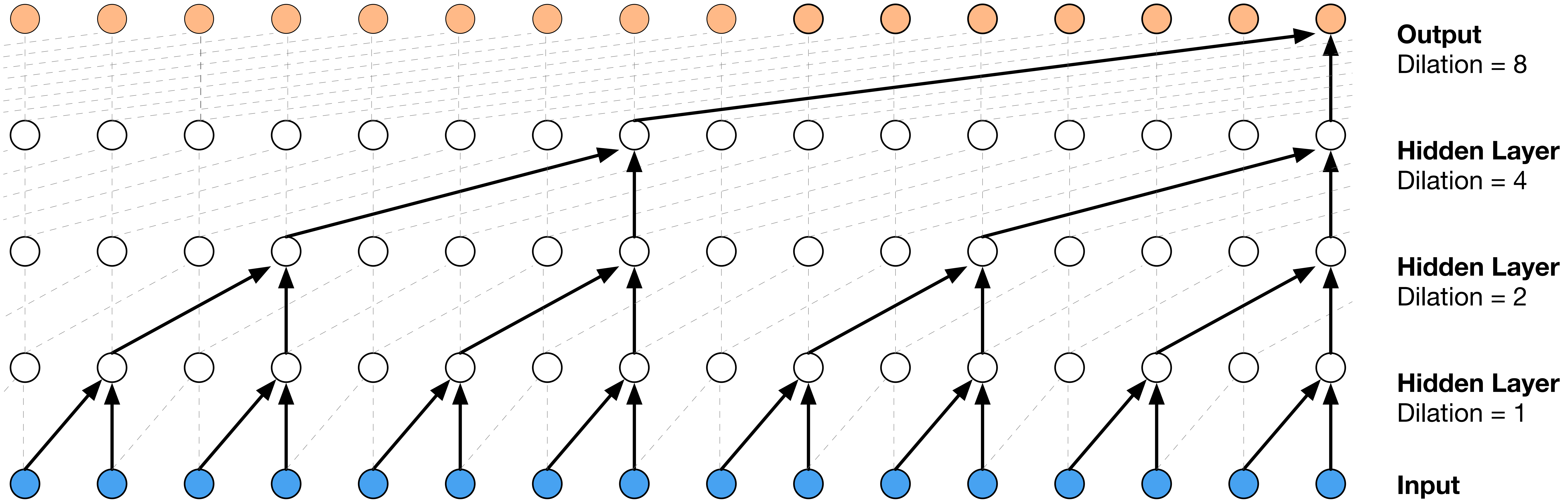
Karen Simonyan **Oriol Vinyals** **Alex Graves**

Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK

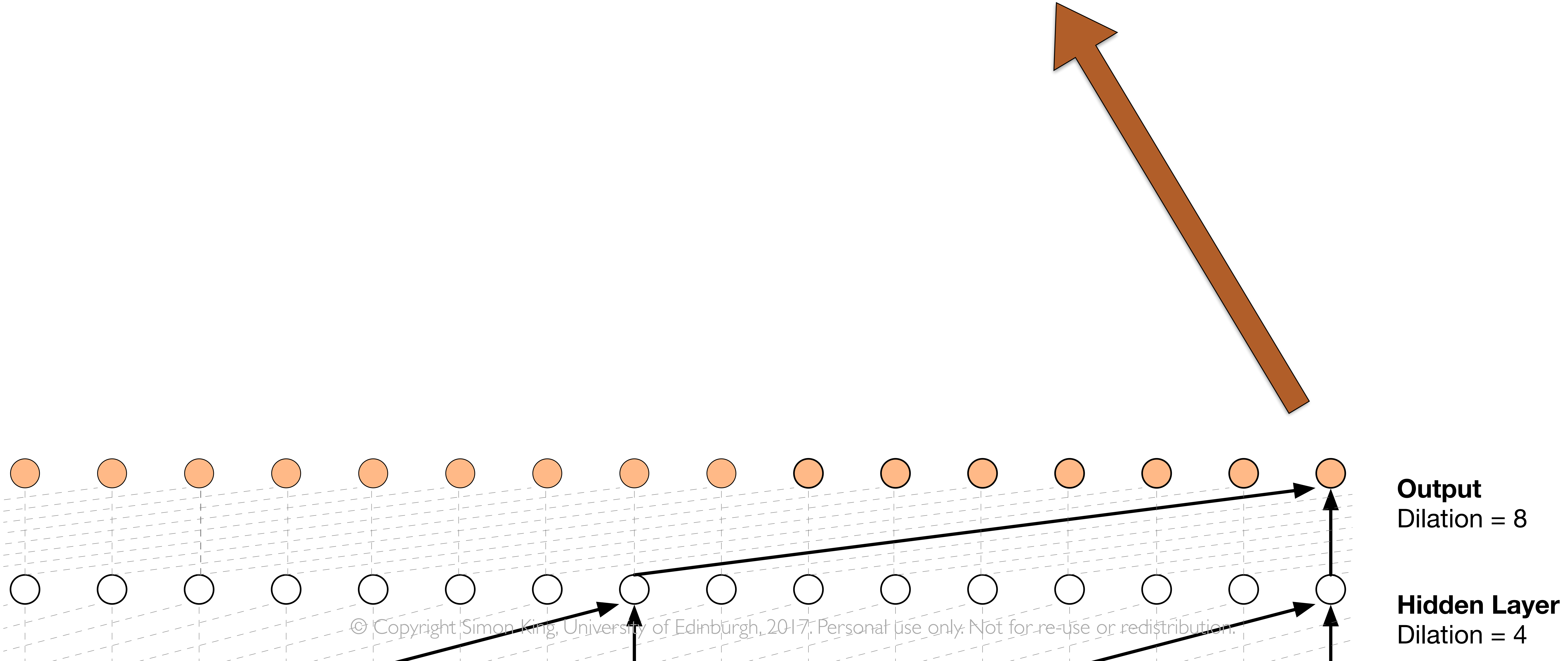
19 Sep 2016

ABSTRACT


Output
Dilation = 8
Hidden Layer
Dilation = 4
Hidden Layer
Dilation = 2
Hidden Layer
Dilation = 1
Input

"one-hot" coding of 8 bit quantised waveform sample = **1-of-256**

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0



DOI: 10.21437/Interspeech.2017-1452

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Tacotron: Towards End-to-End Speech Synthesis

Yuxuan Wang, RJ Skerry-Ryan*, Daisy Stanton, Yonghui Wu, Ron J. Weiss†, Navdeep Jaitly, Zongheng Yang, Ying Xiao*, Zhifeng Chen, Samy Bengio†, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous**

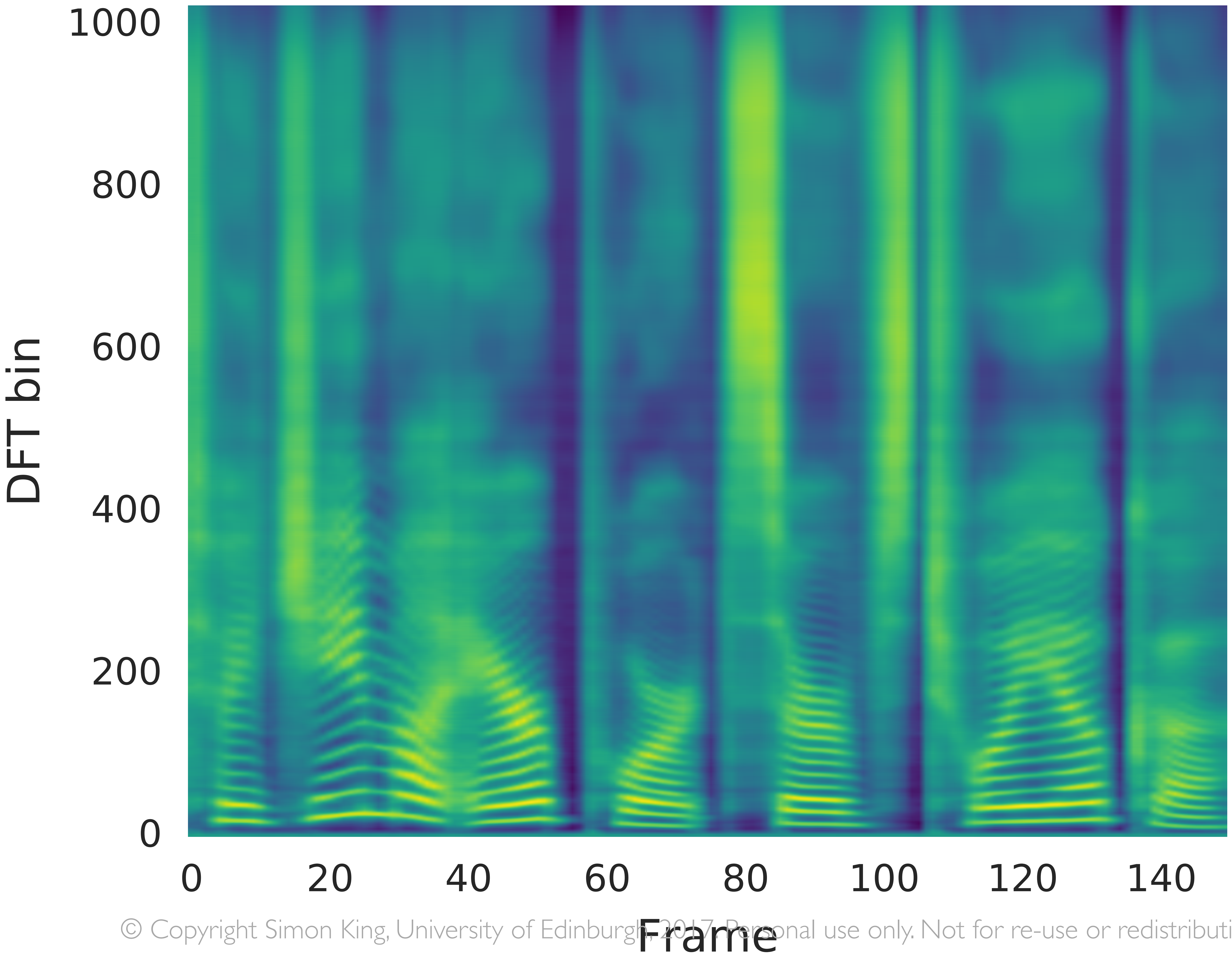
Google, Inc.
{yxwang, rjryan, rif}@google.com

Abstract

A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module. Building these components often requires extensive domain expertise and may contain brittle

this is a particularly difficult learning task for an end-to-end model: it must cope with large variations at the signal level for a given input. Moreover, unlike end-to-end speech recognition [4] or machine translation [5], TTS outputs are continuous, and output sequences are usually much longer than those of the input. These attributes cause prediction errors to accu-

DOI: 10.21437/Interspeech.2017-1452



DOI: 10.1109/TASSP.1984.1164317

Signal Estimation from Modified Short-Time Fourier Transform

DANIEL W. GRIFFIN AND JAE S. LIM, SENIOR MEMBER, IEEE

Abstract—In this paper, we present an algorithm to estimate a signal from its modified short-time Fourier transform (STFT). This algorithm is computationally simple and is obtained by minimizing the mean squared error between the STFT of the estimated signal and the modified STFT. Using this algorithm, we also develop an iterative algorithm to estimate a signal from its modified STFT magnitude. The iterative algorithm is shown to decrease, in each iteration, the mean squared error between the STFT magnitude of the estimated signal and the modified STFT magnitude. The major computation involved in the iterative algorithm is the discrete Fourier transform (DFT) computation, and the algorithm appears to be real-time implementable with current hardware technology. The algorithm developed in this paper

mated signal and the MSTFT. The resulting algorithm is quite simple computationally. In Section III, the algorithm in Section II is used to develop an iterative algorithm that estimates a signal from the MSTFTM. The iterative algorithm is shown to decrease, in each iteration, the mean squared error between the STFTM of the estimated signal and the MSTFTM. In Section IV, we present an example of the successful application of our theoretical results. Specifically, we develop a time-scale speech modification system by modifying the STFTM first and then estimating a signal from the MSTFTM using the

Part 3 - What do we want from our speech signal representation?

We ask a lot of our representation

- Easy to **extract** from speech waveforms
- **Compact** (low dimensional)
- "**Well-behaved**" because statistical modelling will introduce **errors**
  - **reconstruction** of waveforms from corrupted parameters must be possible

- Statistical model training aims to **minimise error** (loss) function *in the domain of the representation*

What will statistical modelling do to our acoustic features?

Some things that statistical models might do to our acoustic features

- Incorrect variance of acoustic feature trajectories (too much or too little variance)
- Failure to capture covariance between features
- Temporal smoothing
- Averaging of features (e.g., within a cluster of HMM states)

Investigating the shortcomings of HMM synthesis

Thomas Merritt, Simon King

The Centre for Speech Technology Research, The University of Edinburgh, U.K.

T.Merritt@ed.ac.uk, Simon.King@ed.ac.uk

Abstract

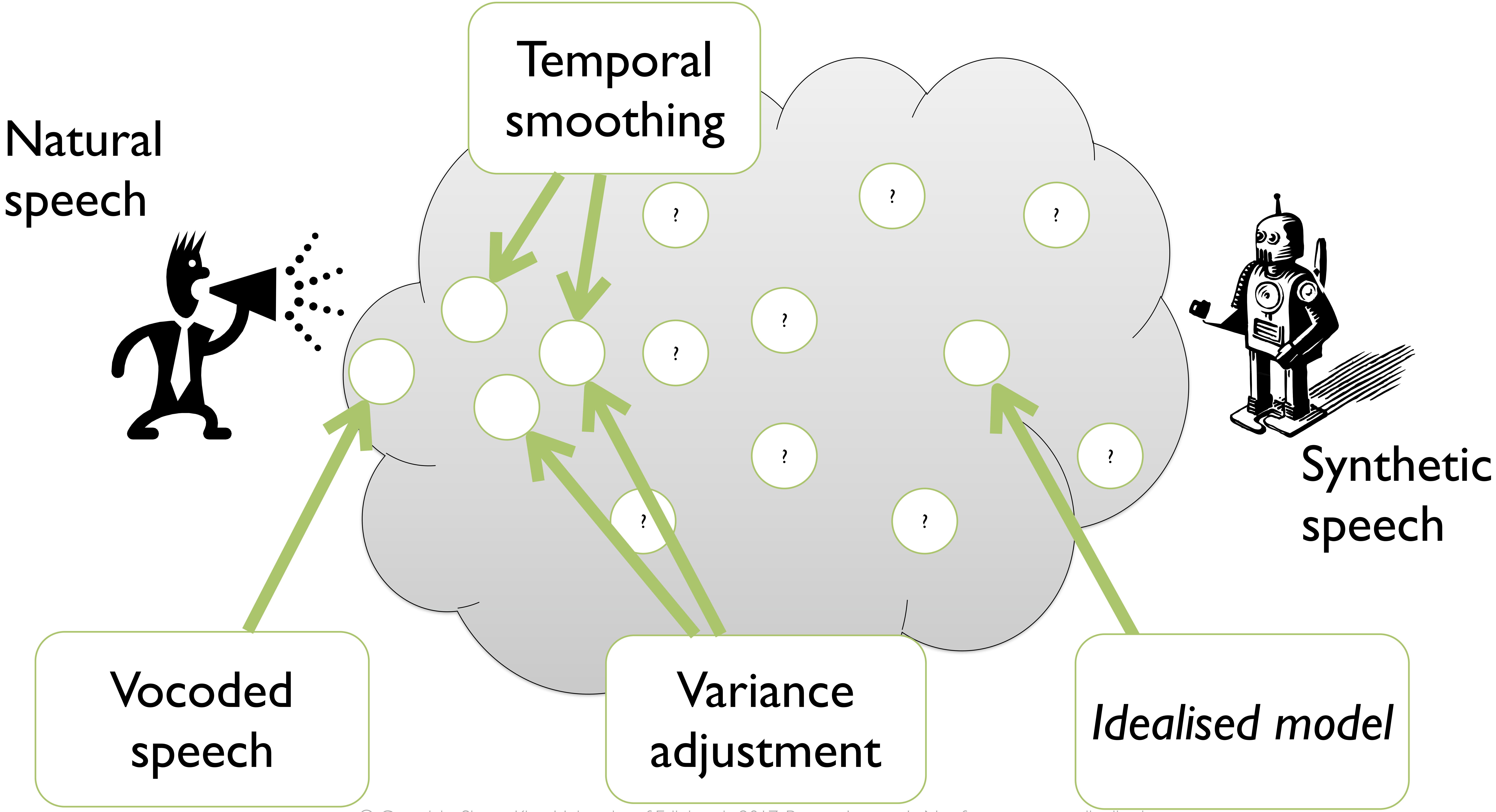
This paper presents the beginnings of a framework for formal testing of the causes of the current limited quality of HMM (Hidden Markov Model) speech synthesis. This framework separates each of the effects of modelling to observe their independent effects on vocoded speech parameters in order to address the issues that are restricting the progression to highly intelligible and natural-sounding speech synthesis.

The simulated HMM synthesis conditions are performed on spectral speech parameters and tested via a pairwise listening test, asking listeners to perform a "same or different" judgement on the quality of the synthesised speech produced between these

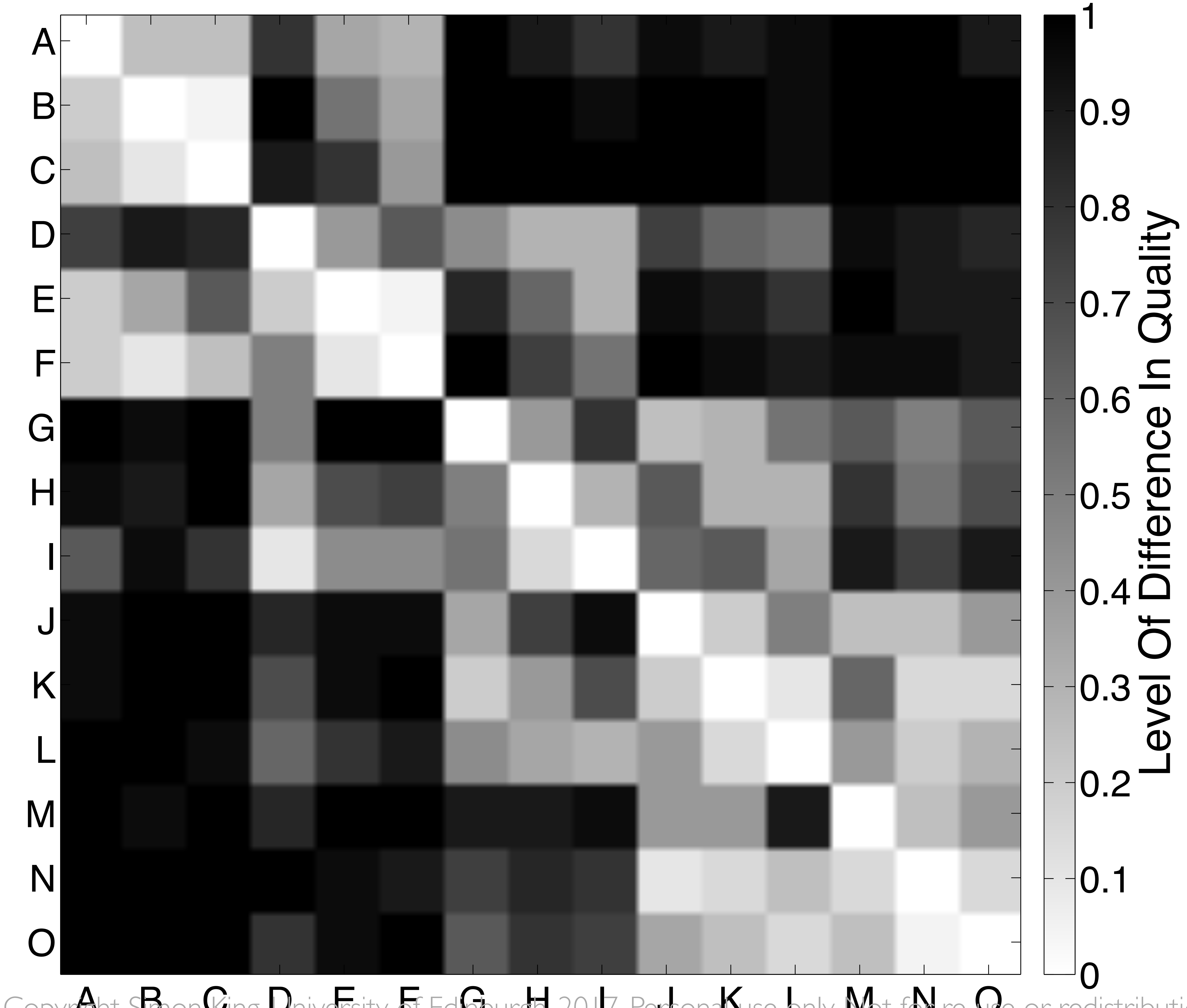
1.1. A simulation framework

This paper introduces such a framework and – as a first illustration of its use – tests a couple of the potential causes of the degradation in naturalness introduced by the use of statistical models. The framework is general and could be applied to many different aspects of the problem. The idea is to *simulate* the effects of modelling vocoded speech, in a carefully controlled manner. Knowledge obtained by such experiments could then be used to identify those areas that are causing the problem, and to eventually rectify them.

Current HMM-based synthesisers are large, complex sys-


Temporal smoothing
Natural speech
?
?
?
?
?
?
?
?
?
Synthetic speech
Vocoded speech
Variance adjustment
Idealised model

Listeners rate **pairwise differences**. Construct **matrix** of differences. Analyse with Multi-Dimensional Scaling (**MDS**).



DOI: 10.1109/ICASSP.2015.7178766

ATTRIBUTING MODELLING ERRORS IN HMM SYNTHESIS BY STEPPING GRADUALLY FROM NATURAL TO MODELLED SPEECH

Thomas Merritt[1], *Javier Latorre*[2], *Simon King*[1]

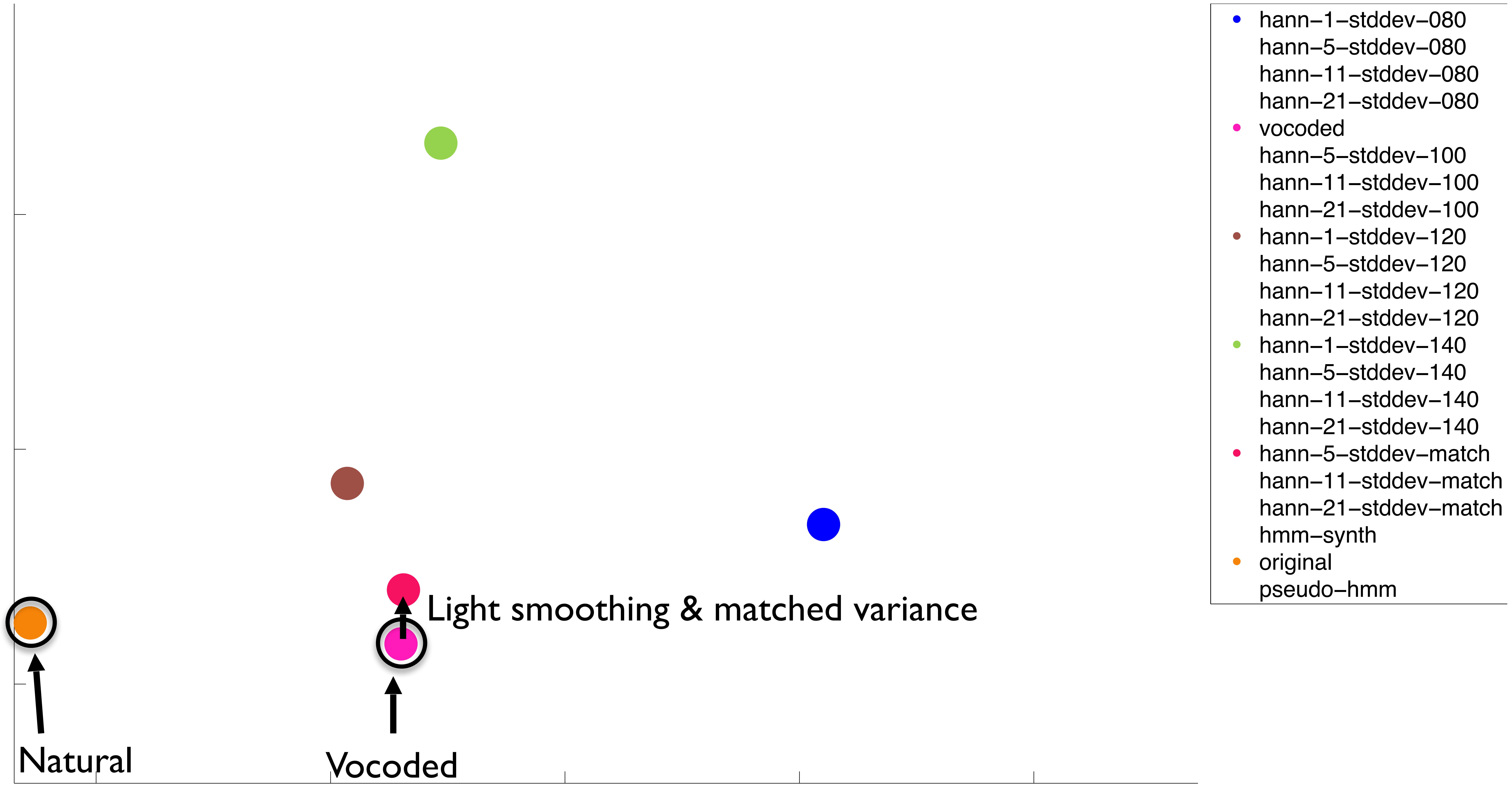
[1] The Centre for Speech Technology Research, University of Edinburgh, UK.
[2] Toshiba Research Europe Ltd., Cambridge Research Lab, Cambridge, UK.

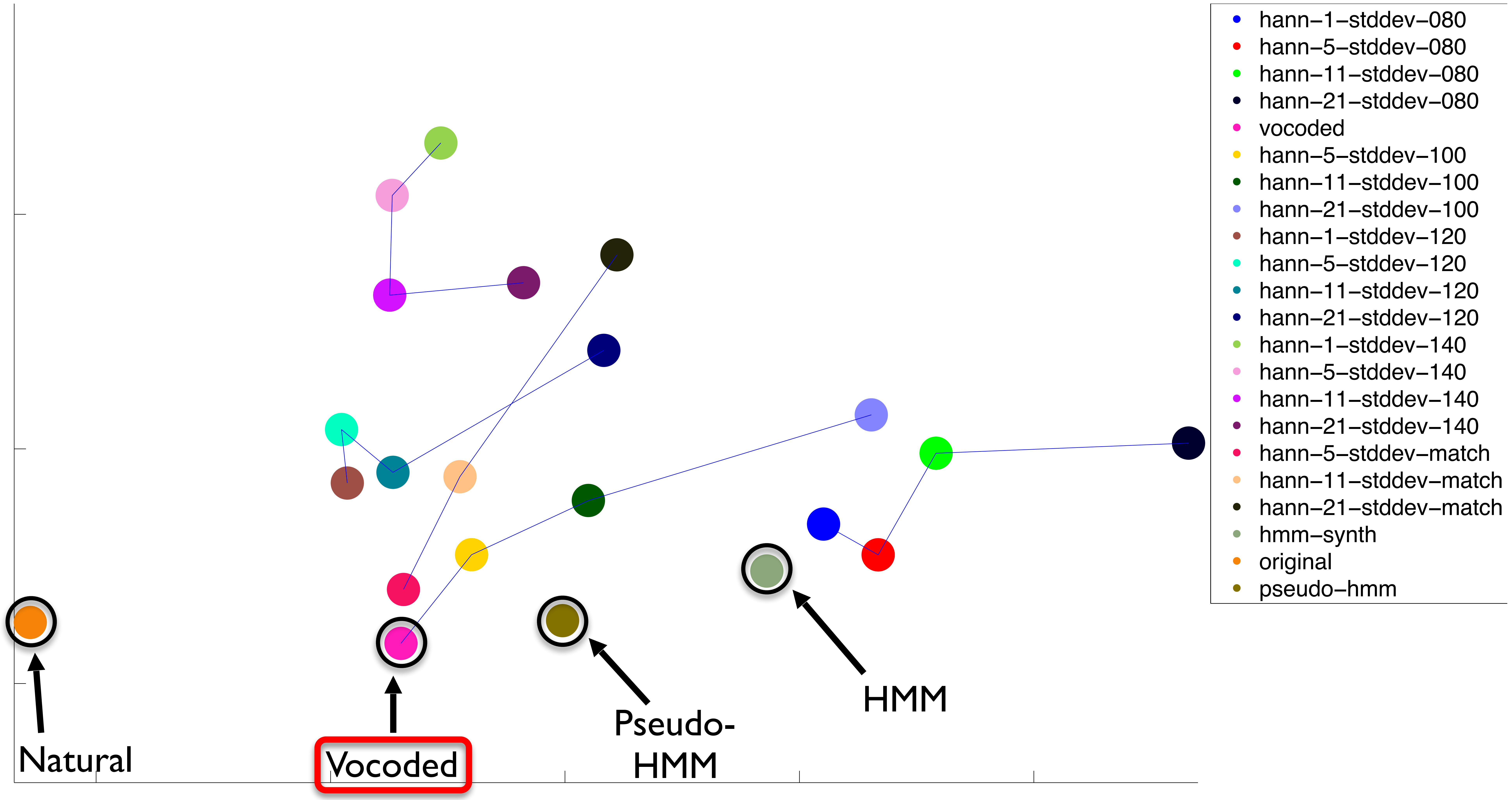
T.Merritt@ed.ac.uk, javier.latorre@crl.toshiba.co.uk, Simon.King@ed.ac.uk

ABSTRACT

Even the best statistical parametric speech synthesis systems do not achieve the naturalness of good unit selection. We investigated possible causes of this. By constructing speech signals that lie inbetween natural speech and the output from a complete HMM synthesis system, we investigated various effects of modelling. We manipulated the temporal smoothness and the variance of the spectral parameters to create stimuli, then presented these to listeners alongside natural and vocoded speech, as well as output from a full HMM-based text-to-speech system and from an idealised 'pseudo-HMM'. All speech signals, except the natural waveform, were created using

Condition	Speech signal origin	Hanning smoothing window duration (frames)	Standard deviation scaling (%)
hann-1-stddev-080	vocoded	none	80
hann-5-stddev-080	vocoded	5	80
hann-11-stddev-080	vocoded	11	80
hann-21-stddev-080	vocoded	21	80
Vocoded	vocoded	none	100
hann-5-stddev-100	vocoded	5	100
hann-11-stddev-100	vocoded	11	100
hann-21-stddev-100	vocoded	21	100


hann–1–stddev–080
hann–5–stddev–080
hann–11–stddev–080
hann–21–stddev–080
vocoded
hann–5–stddev–100
hann–11–stddev–100
hann–21–stddev–100
hann–1–stddev–120
hann–5–stddev–120
hann–11–stddev–120
hann–21–stddev–120
hann–1–stddev–140
hann–5–stddev–140
hann–11–stddev–140
hann–21–stddev–140
hann–5–stddev–match
hann–11–stddev–match
hann–21–stddev–match
hmm–synth
original
pseudo–hmm
Light smoothing & matched variance
Natural
Vocoded


hann–1–stddev–080
hann–5–stddev–080
hann–11–stddev–080
hann–21–stddev–080
vocoded
hann–5–stddev–100
hann–11–stddev–100
hann–21–stddev–100
hann–1–stddev–120
hann–5–stddev–120
hann–11–stddev–120
hann–21–stddev–120
hann–1–stddev–140
hann–5–stddev–140
hann–11–stddev–140
hann–21–stddev–140
hann–5–stddev–match
hann–11–stddev–match
hann–21–stddev–match
hmm–synth
original
pseudo–hmm
Natural
Vocoded
Pseudo-HMM
HMM

Behaviour of speech parameterisations

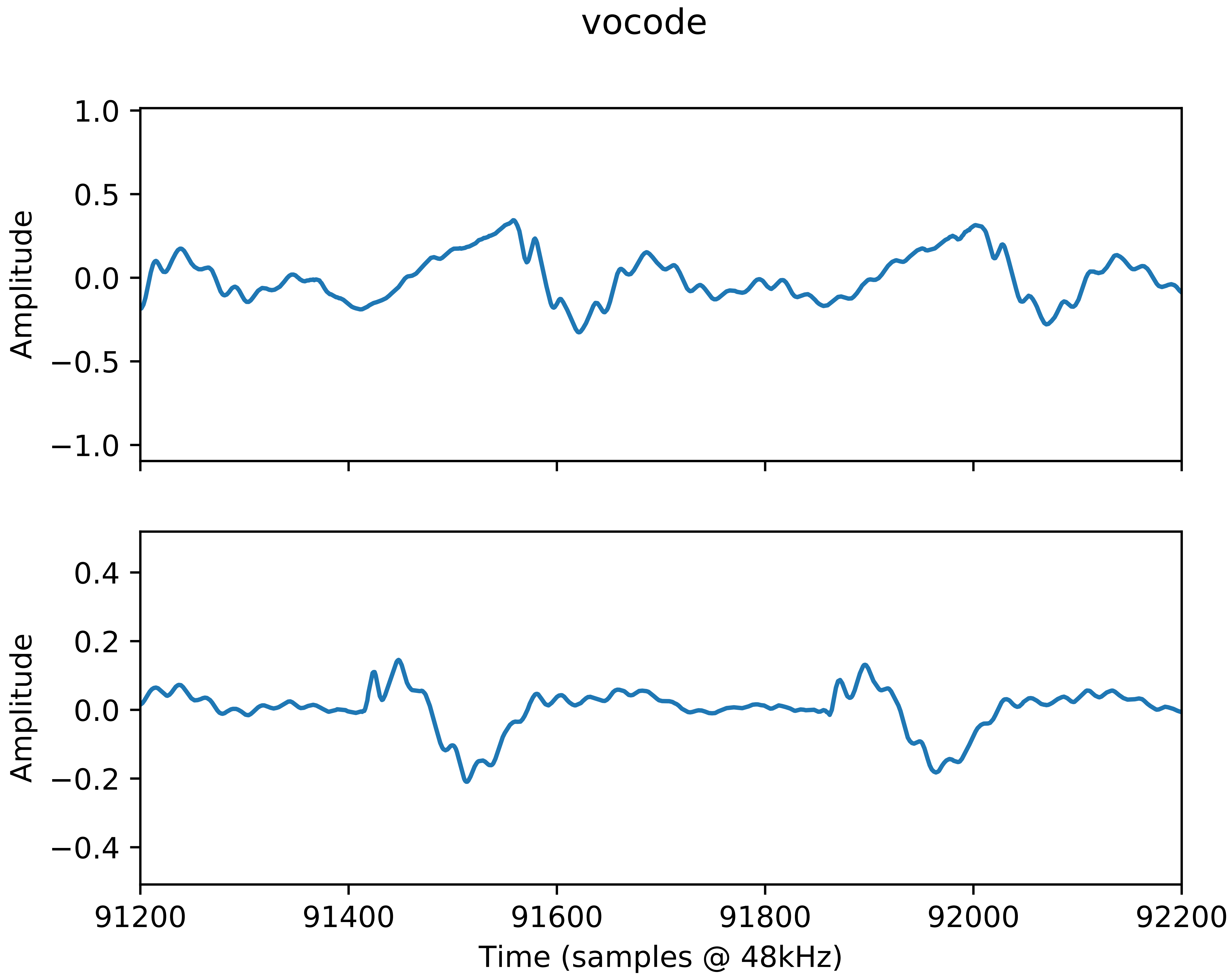
A toy experiment

- Parameterise a speech waveform using
  - vocoder features (high-dimensional)
  - engineered speech synthesis features (reduced dimension)
  - quantised waveform samples (like Wavenet)
- Corrupt the parameters in various ways, as modelling might do
  - isolated frame (or sample) corruption
  - moving average (temporal smoothing)
- Reconstruct waveform
- Listen to perceptual consequences

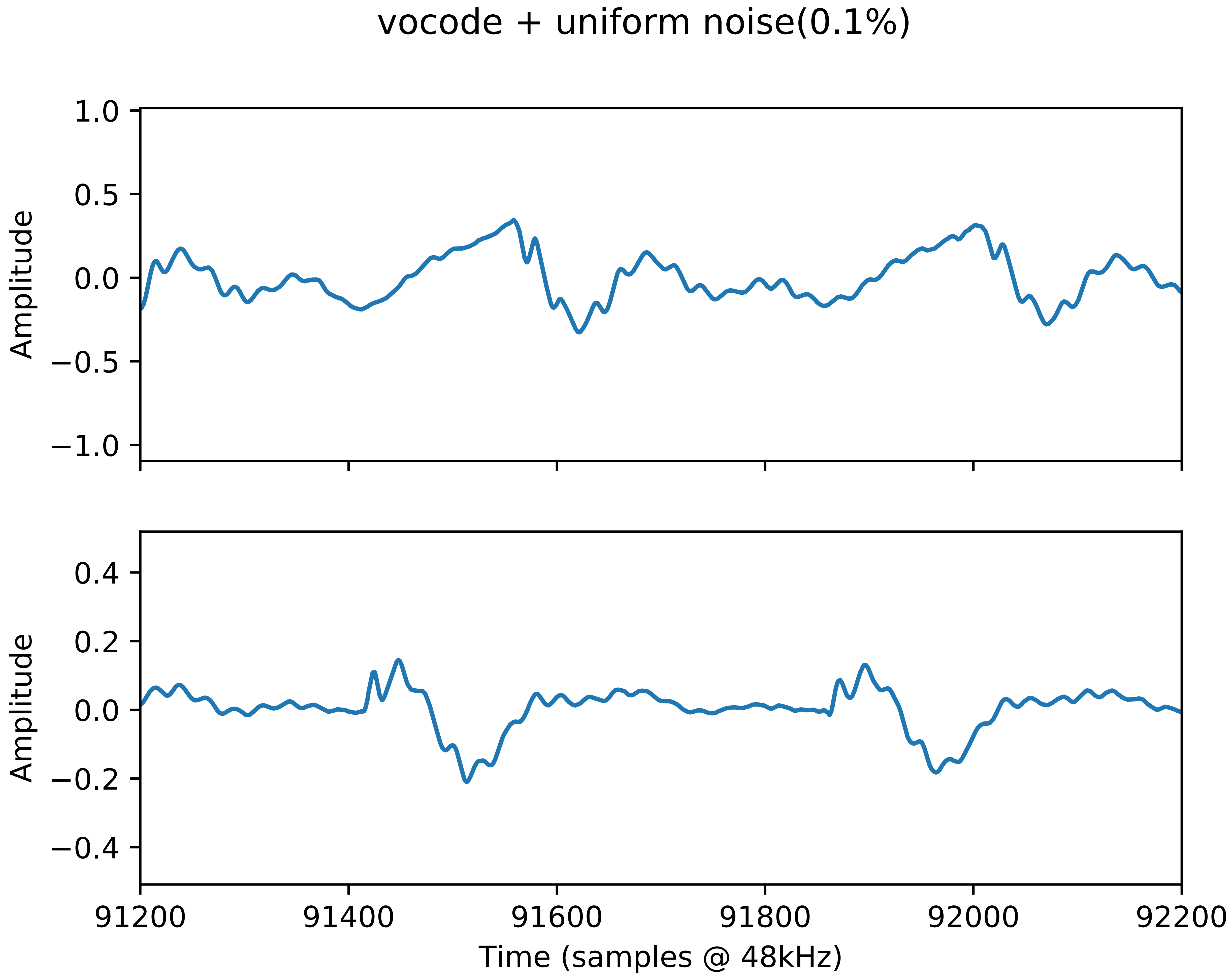
Typical vocoder features, from STRAIGHT

- **High-resolution** (i.e., half FFT length)
  - smooth spectral envelope
  - aperiodic energy ratio

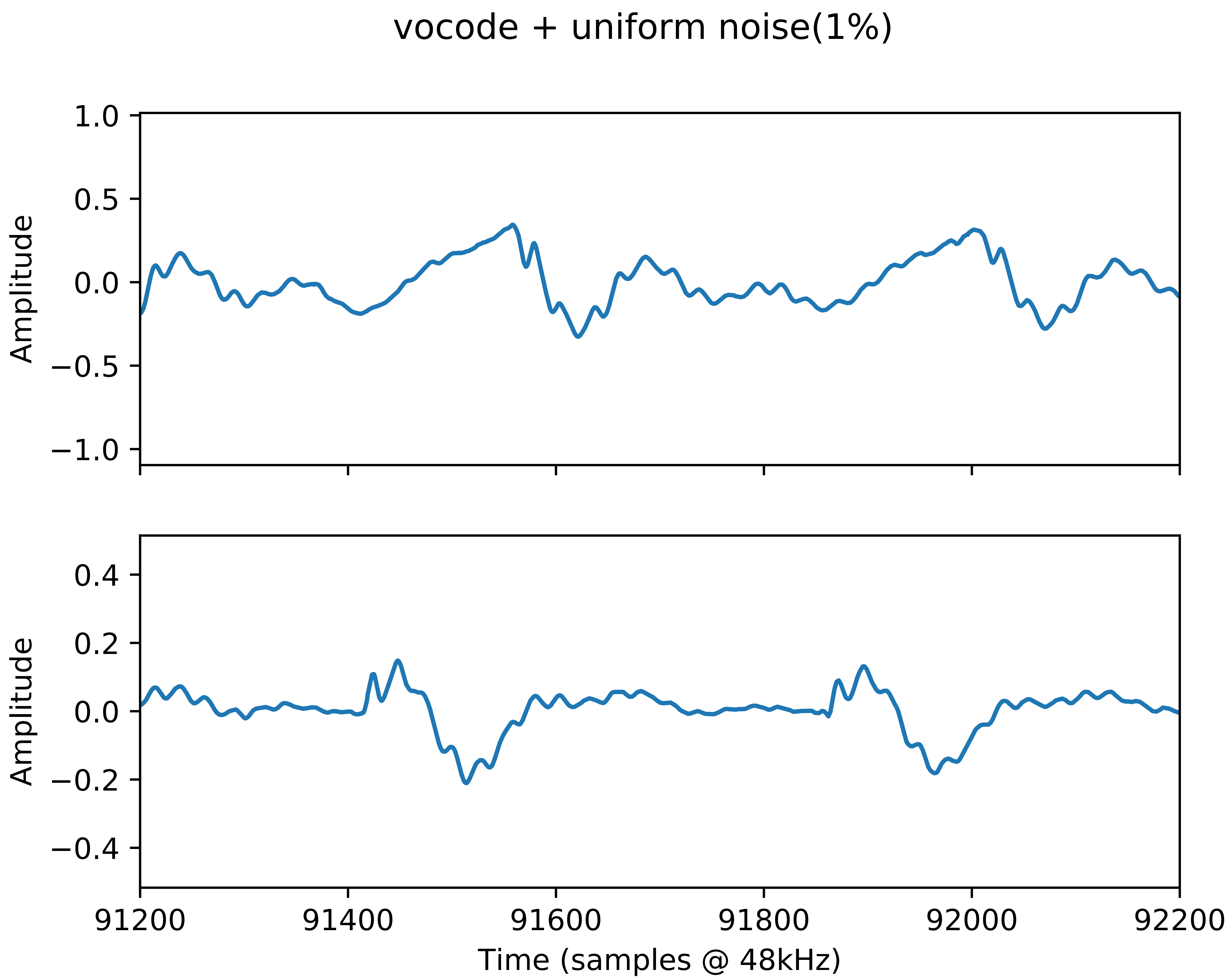
Natural speech vs Vocoded speech



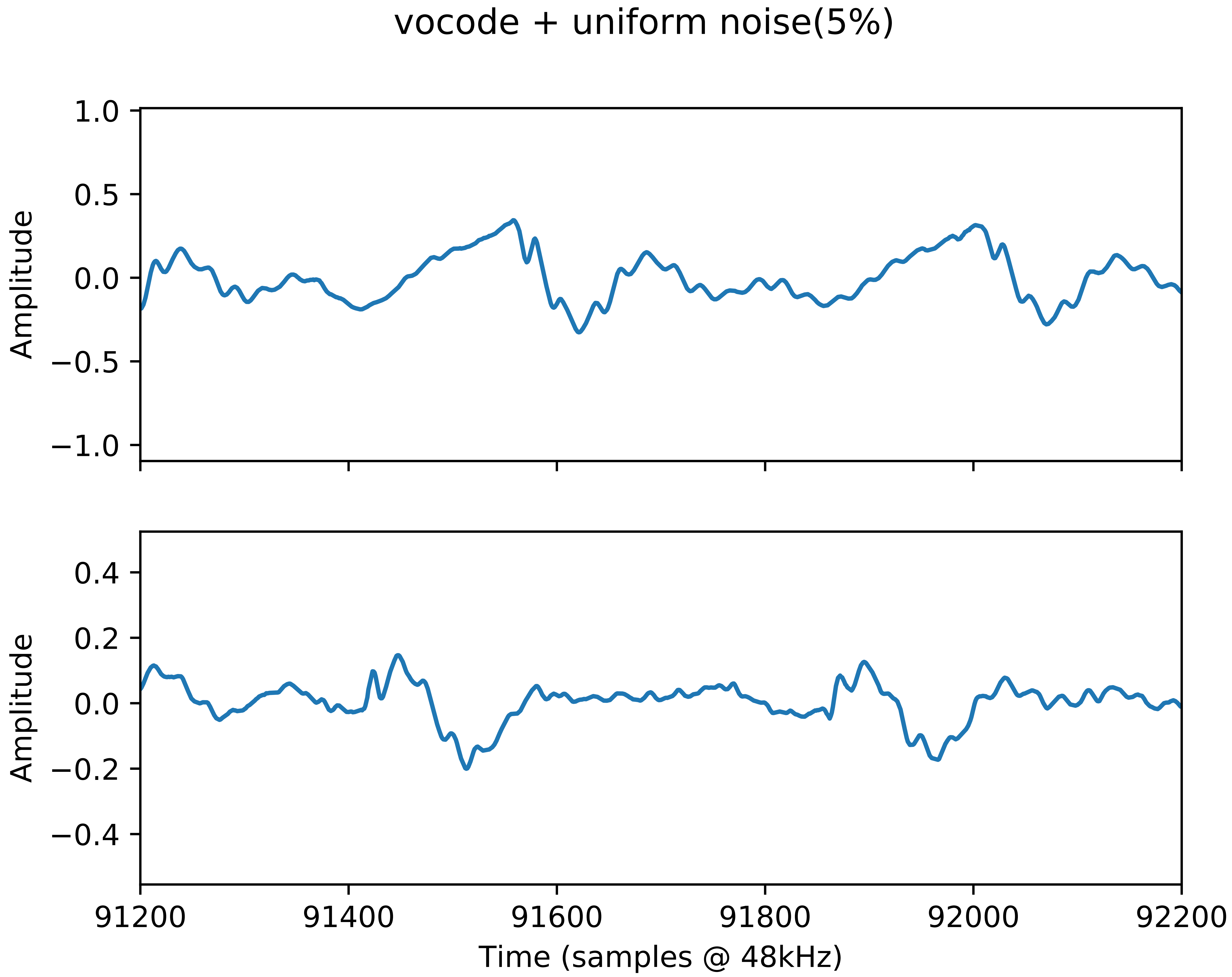
Vocoded speech - corrupt 0.1% of frames (200 frames per second)



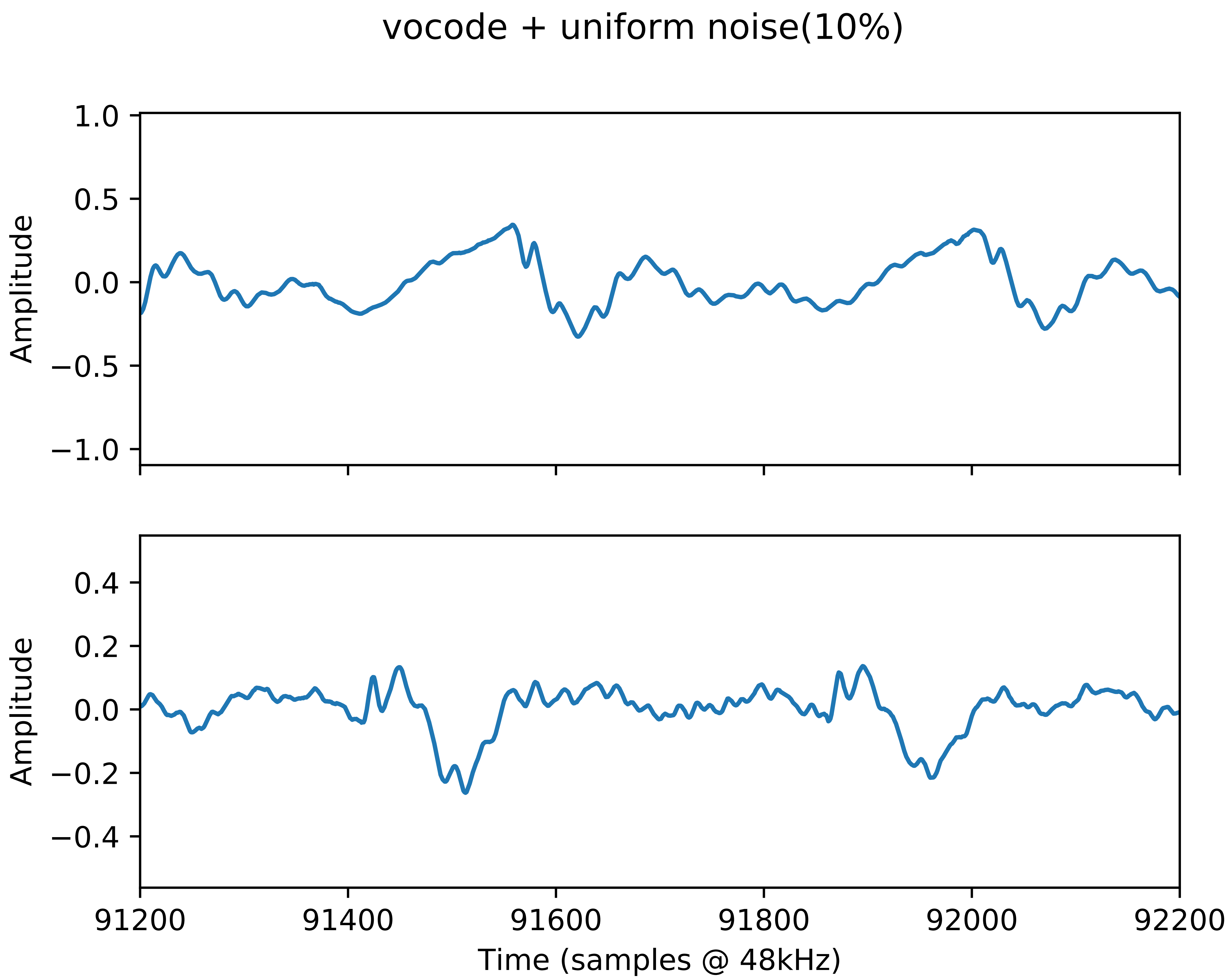
Vocoded speech - corrupt 1% of frames (200 frames per second)



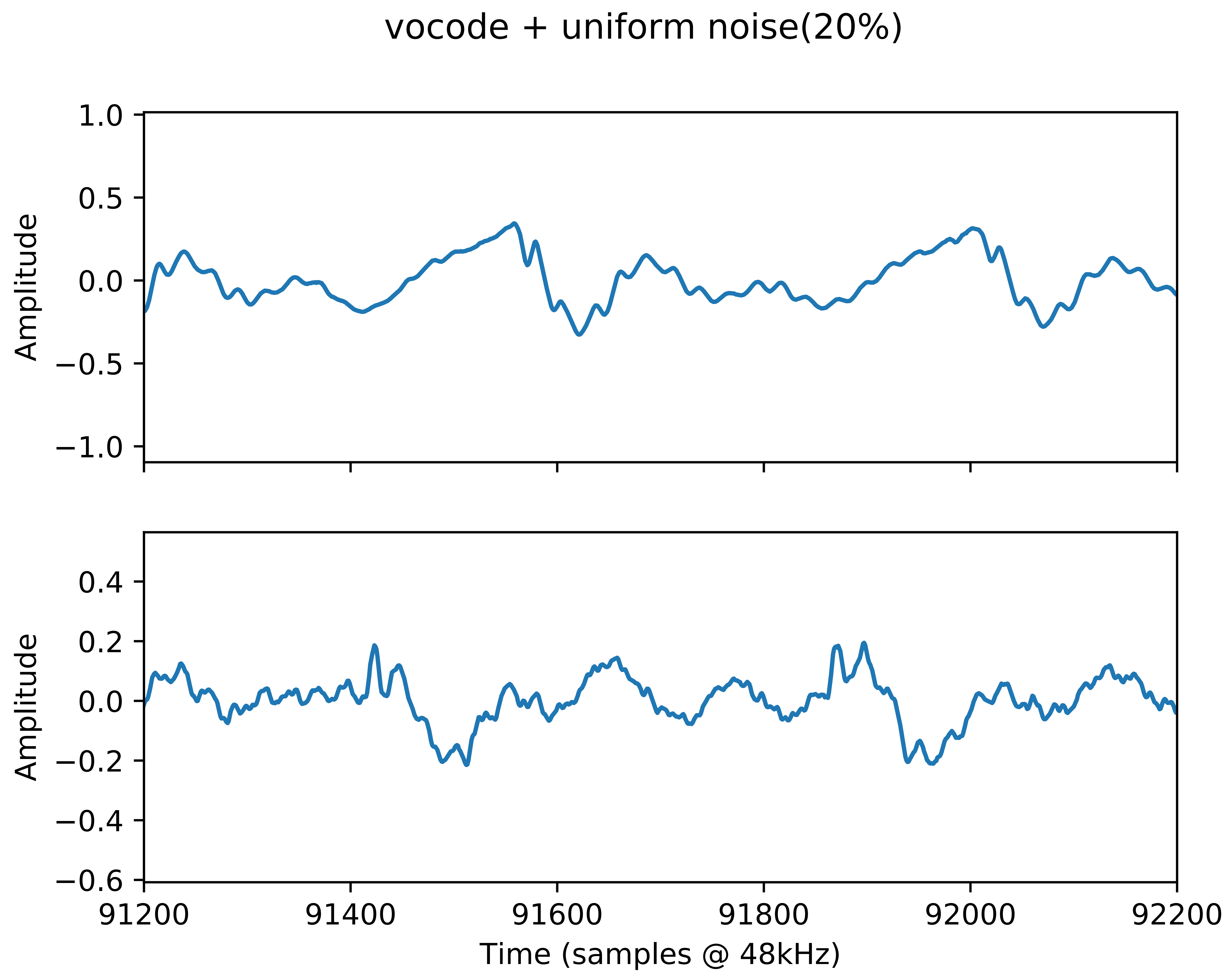
Vocoded speech - corrupt 5% of frames (200 frames per second)



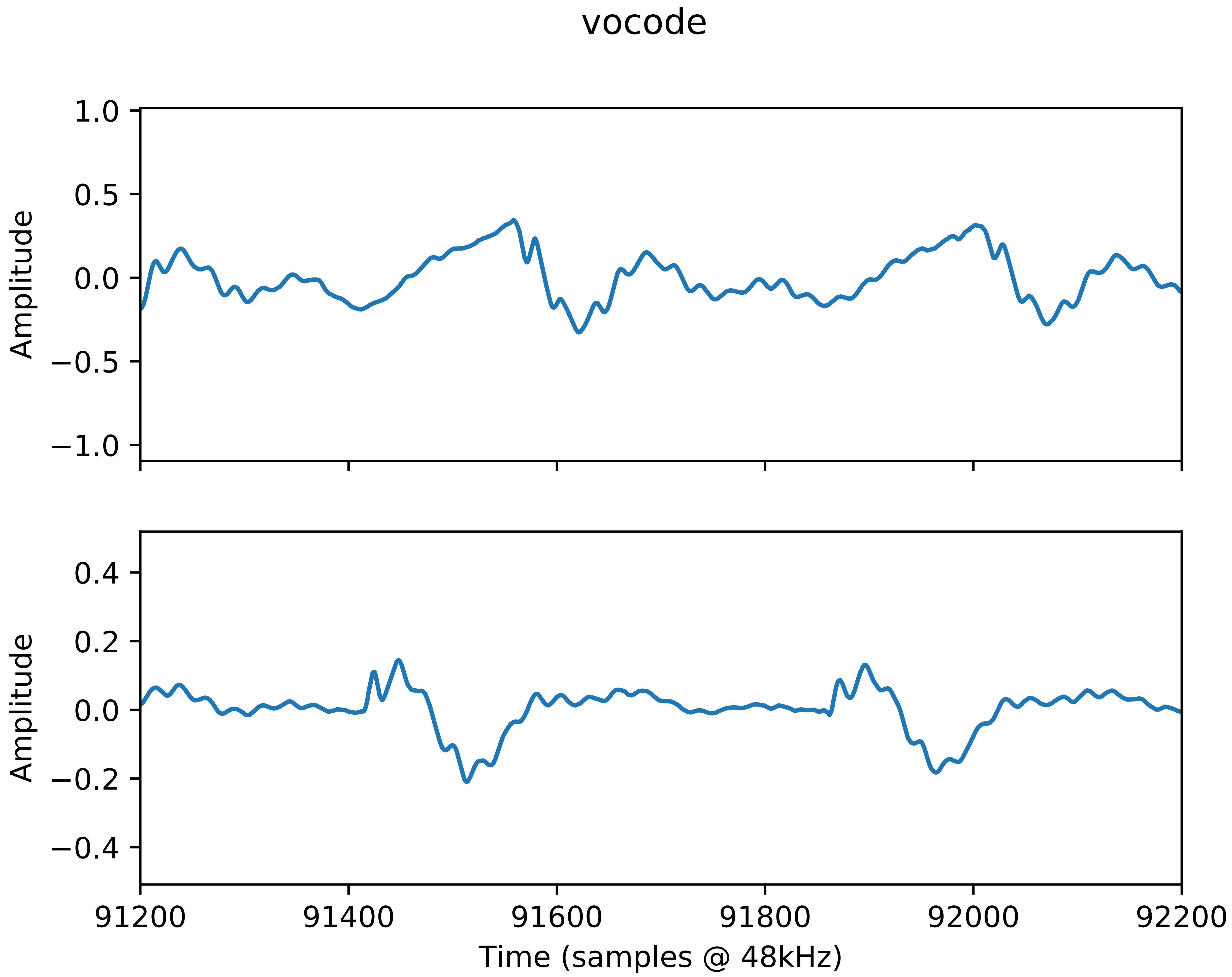
Vocoded speech - corrupt 10% of frames (200 frames per second)



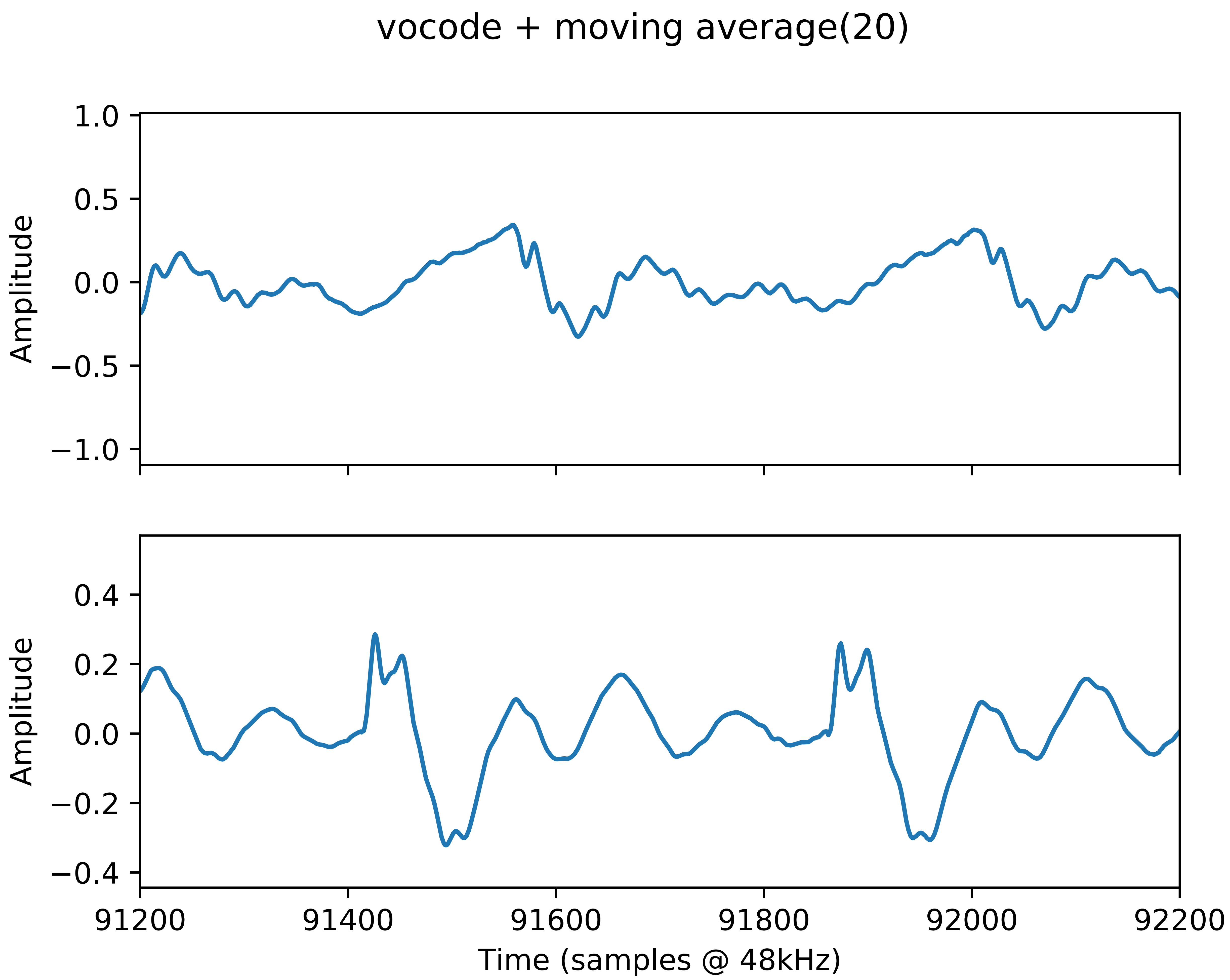
Vocoded speech - corrupt 20% of frames (200 frames per second)



Natural speech vs Vocoded speech



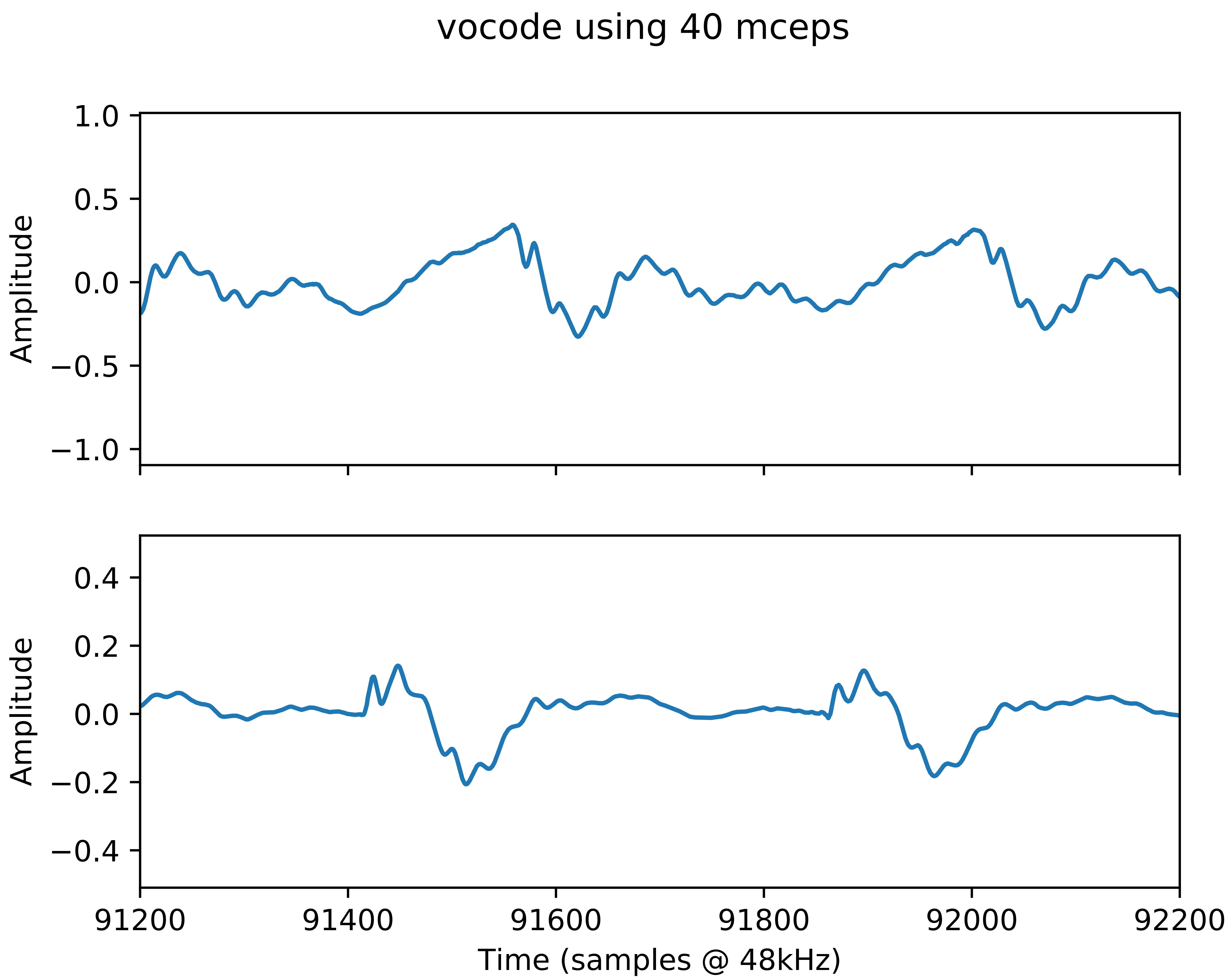
Vocoded speech - moving average, length 20 frames (100ms)



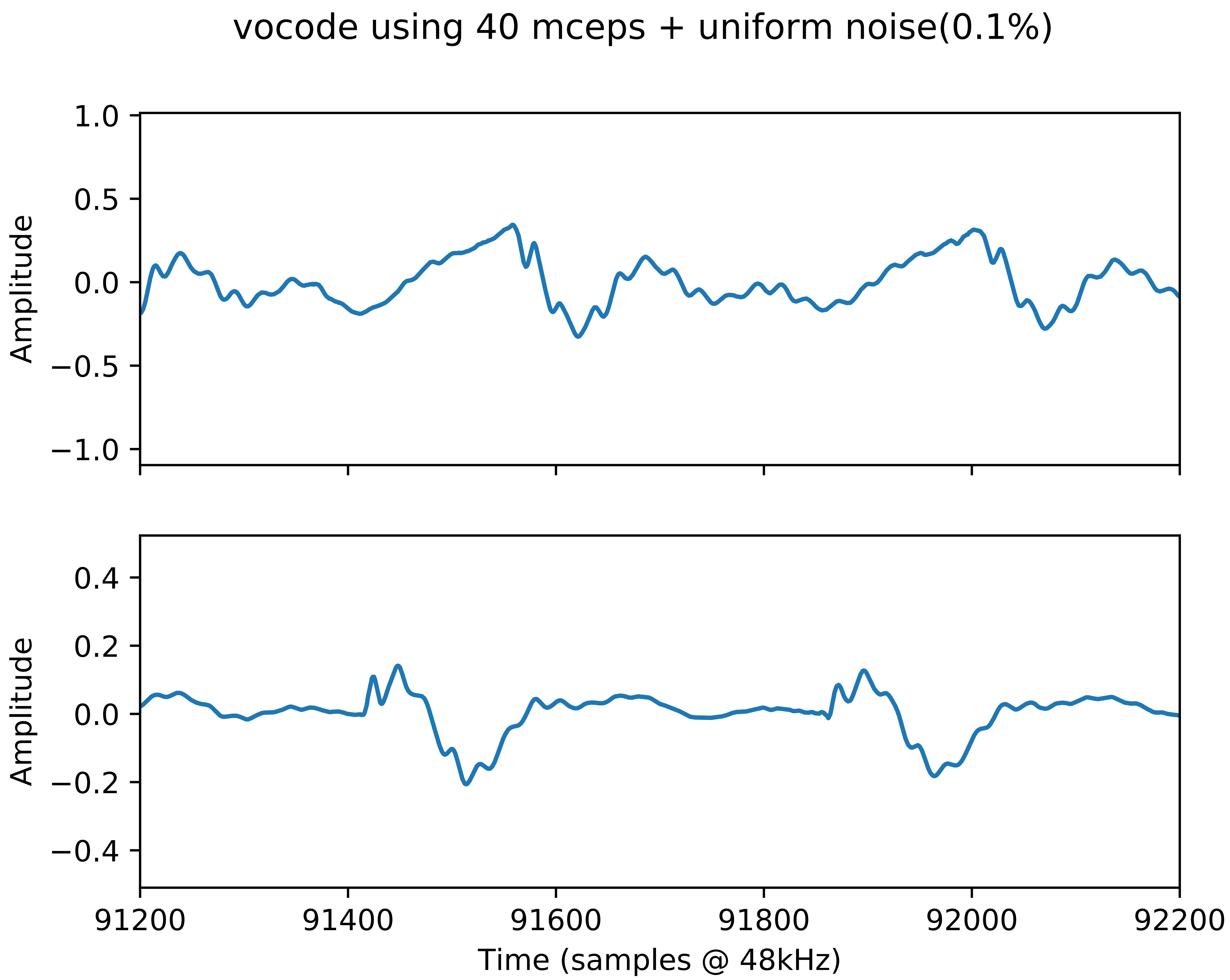
Typical acoustic features used in speech synthesis: Mel cepstrum

- **Dimensionality-reduced** smooth spectral envelope, represented as Mel cepstrum, order 40
- Aperiodic energy averaged across Mel-scaled **frequency bands**

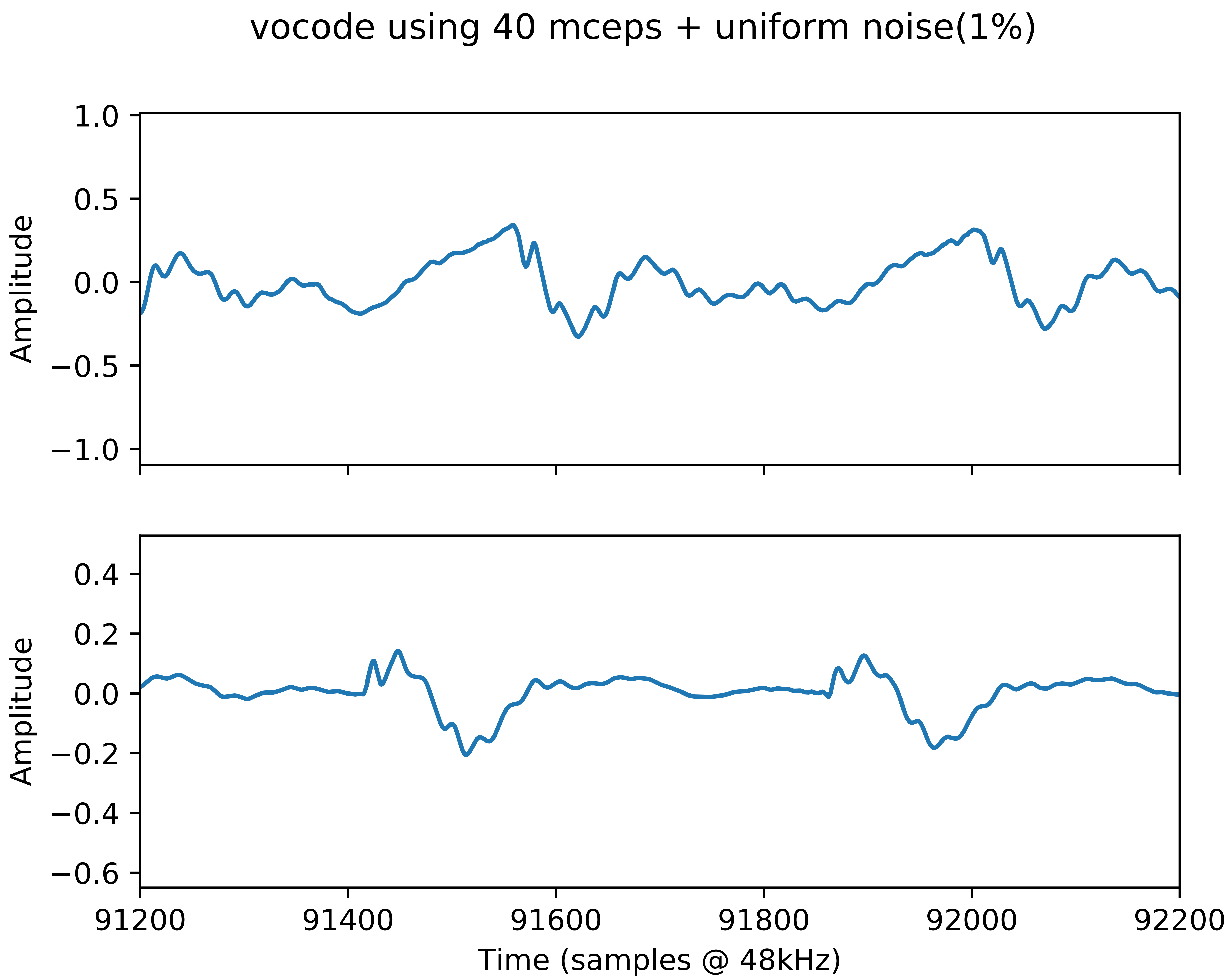
Natural Speech vs Vocoded speech via Mel-cepstrum



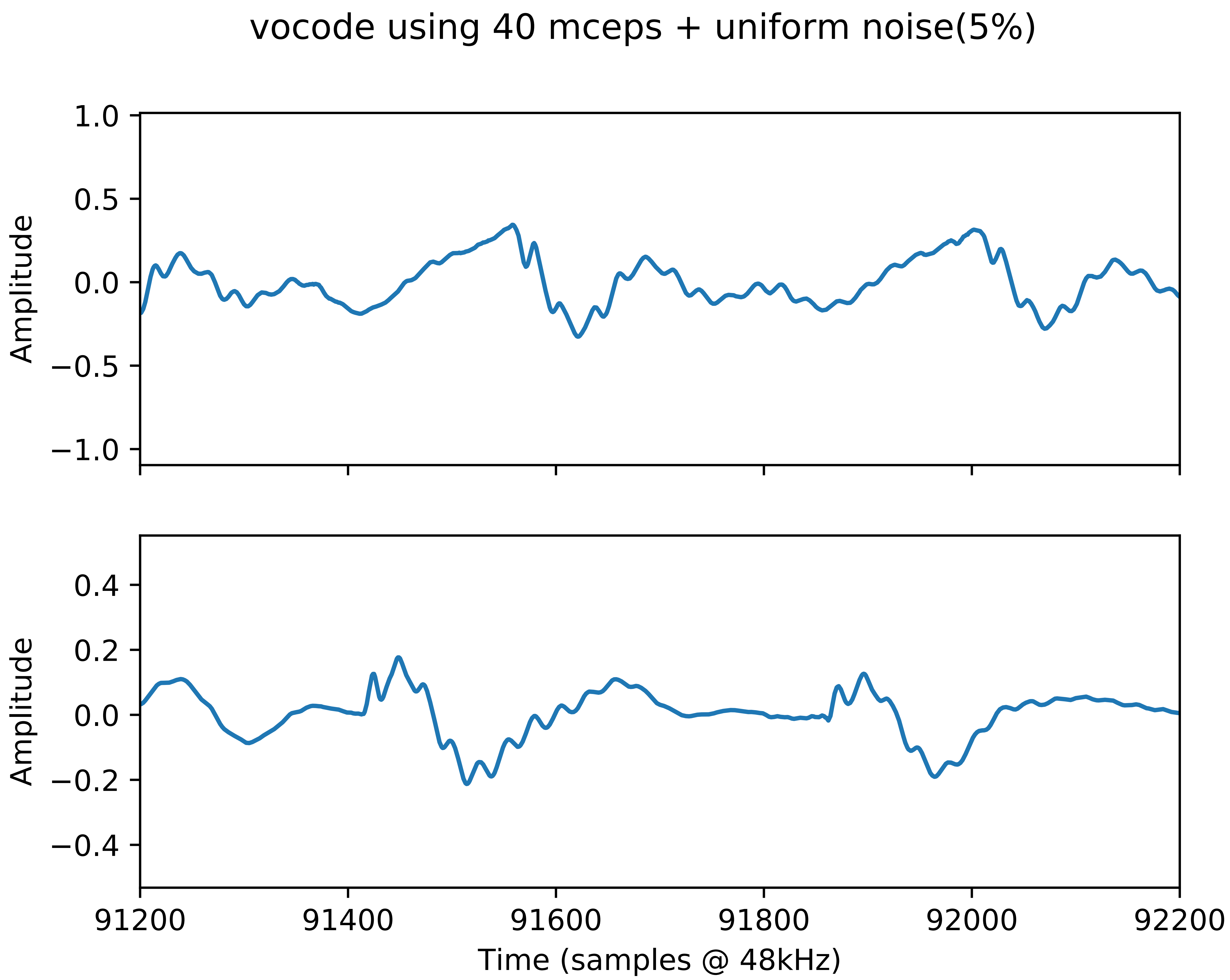
Vocoded speech via Mel-cepstrum - corrupt 0.1% of frames



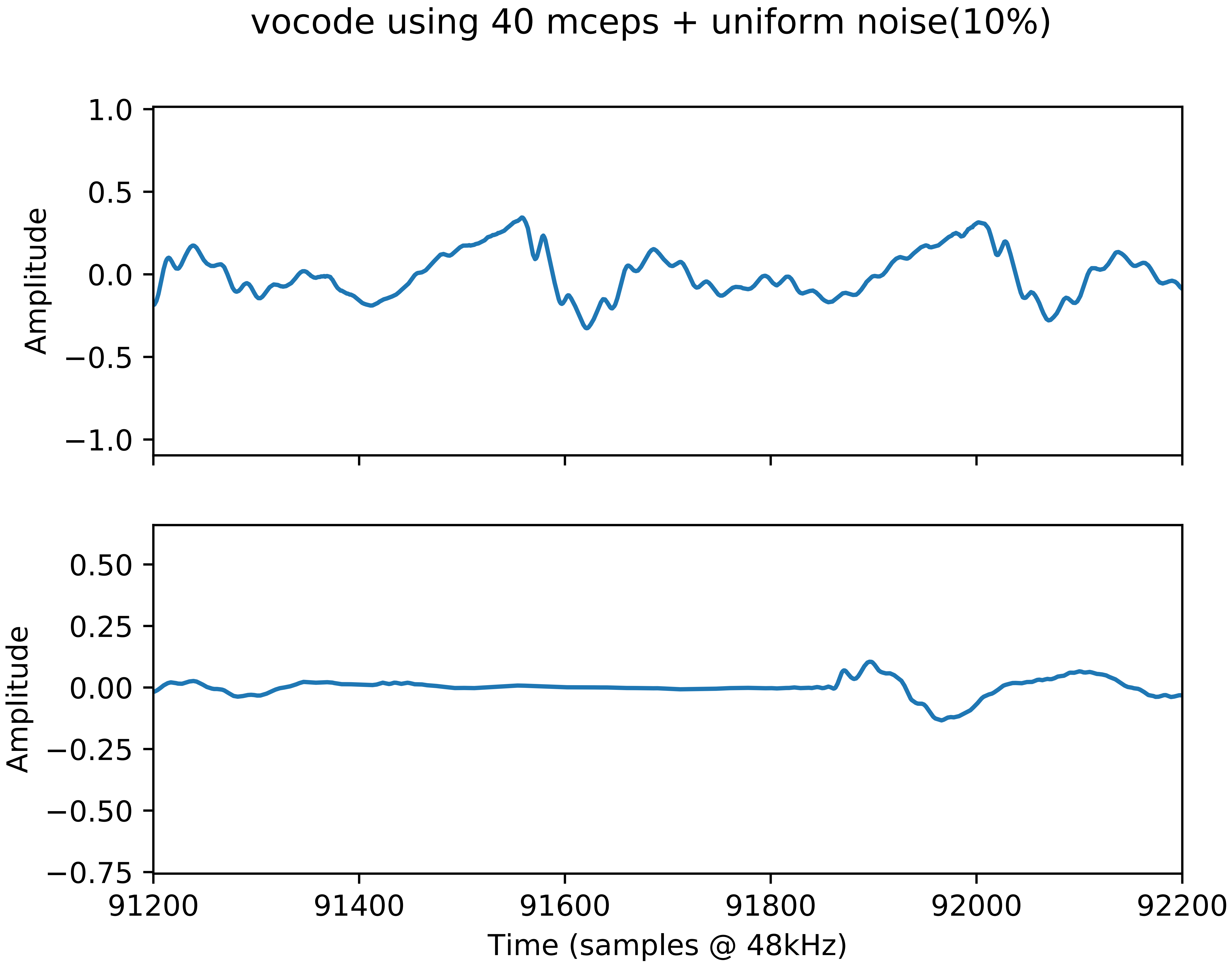
Vocoded speech via Mel-cepstrum - corrupt 1% of frames



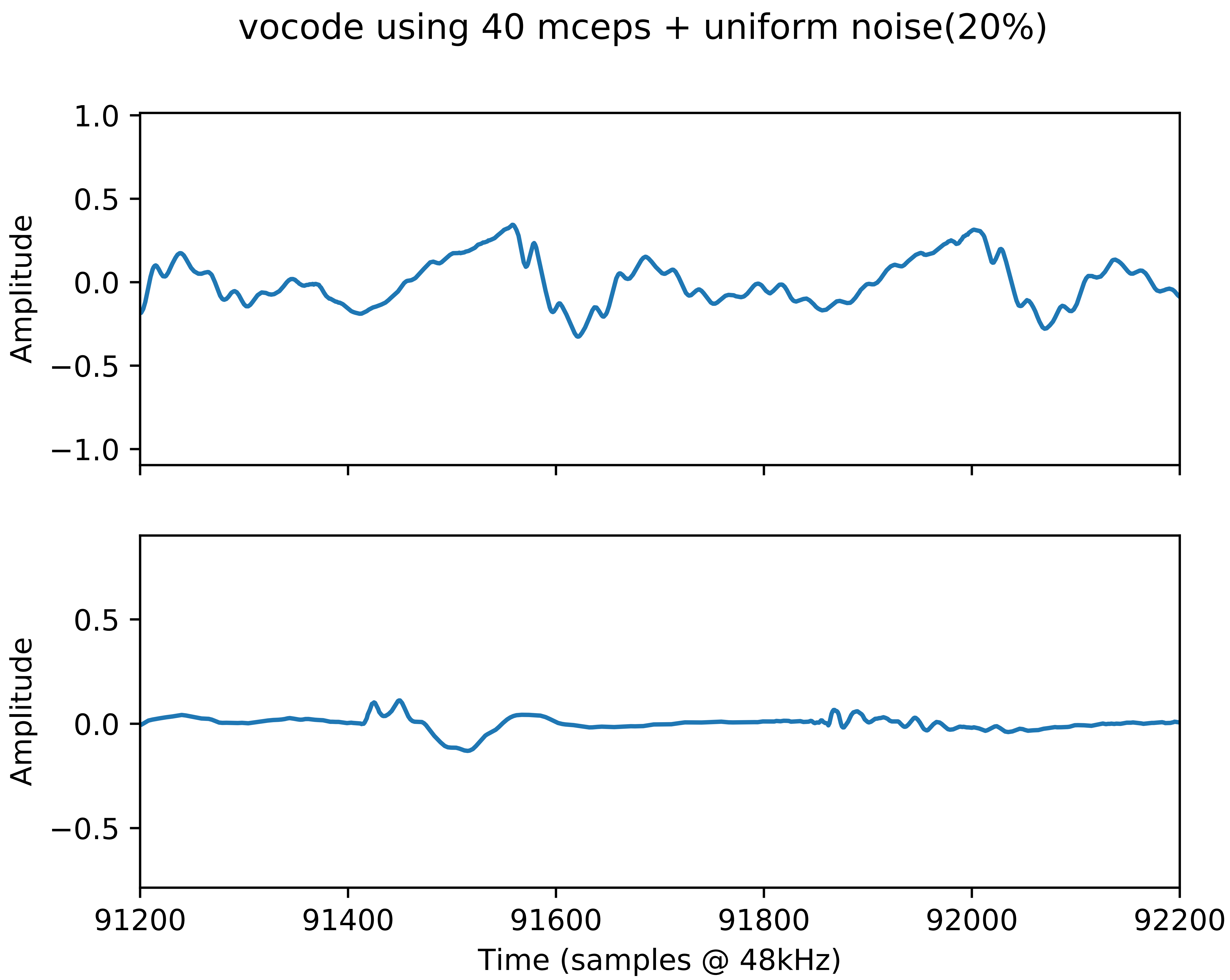
Vocoded speech via Mel-cepstrum - corrupt 5% of frames



Vocoded speech via Mel-cepstrum - corrupt 10% of frames



Vocoded speech via Mel-cepstrum - corrupt 20% of frames

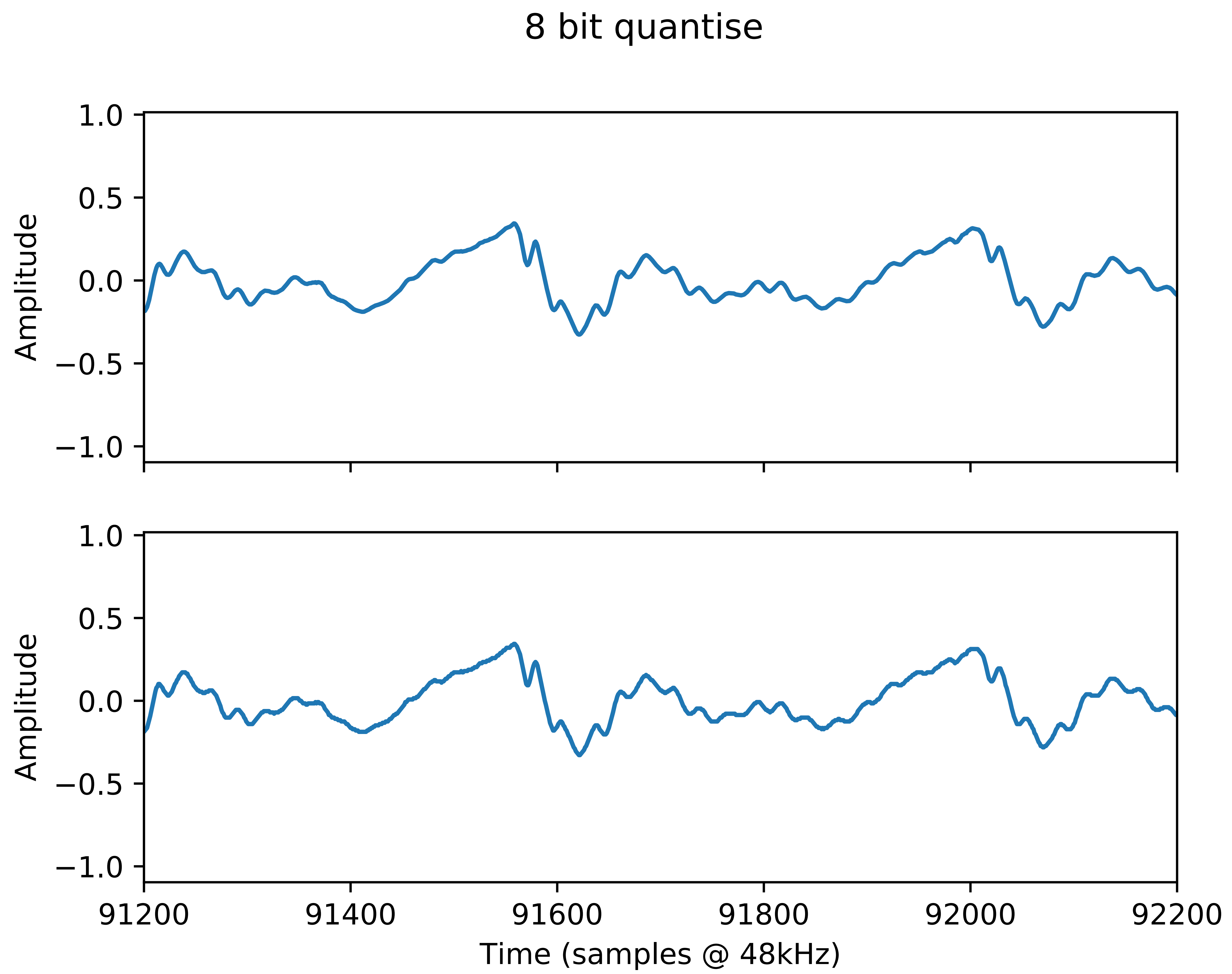


Quantised waveform

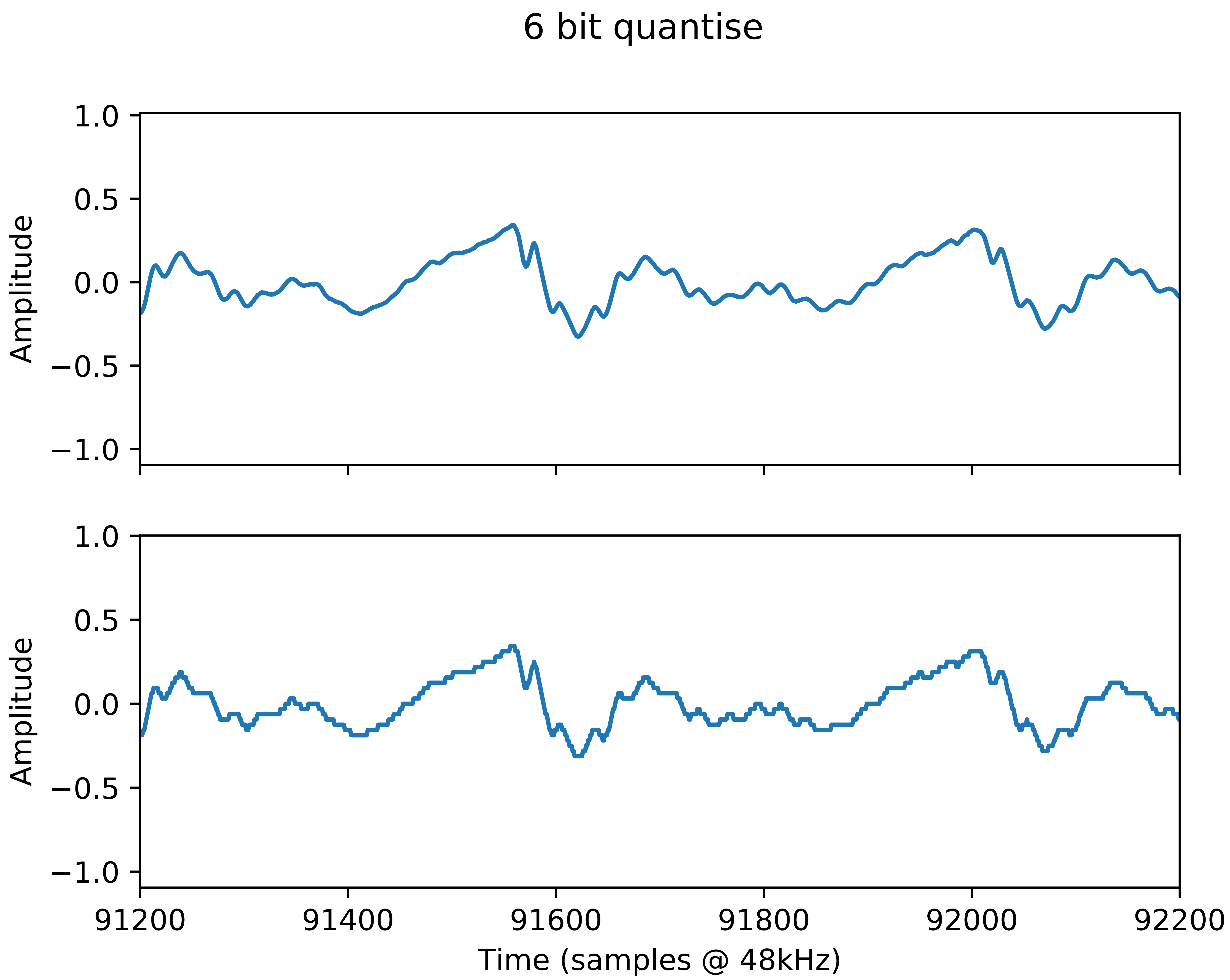
- *Reminder*: Wavenet represents samples using 1-of-256 encoding, which would scale badly with higher bit depth

- 1-of-256 is the most naive sparse code. Surely someone can do better (*cf keynote by Aggelos Katsaggelos*)

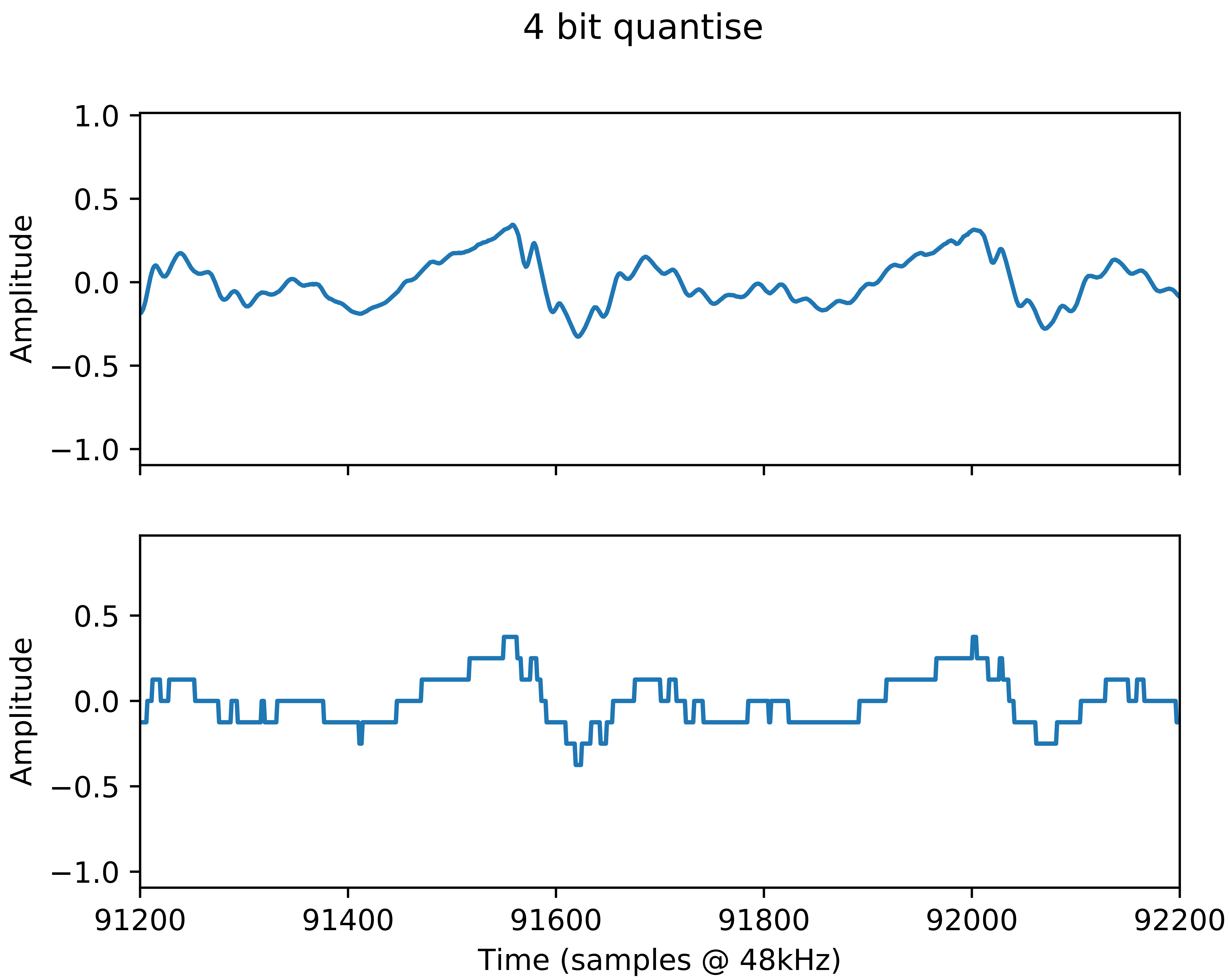
Natural Speech vs Quantised waveform - 8 bit



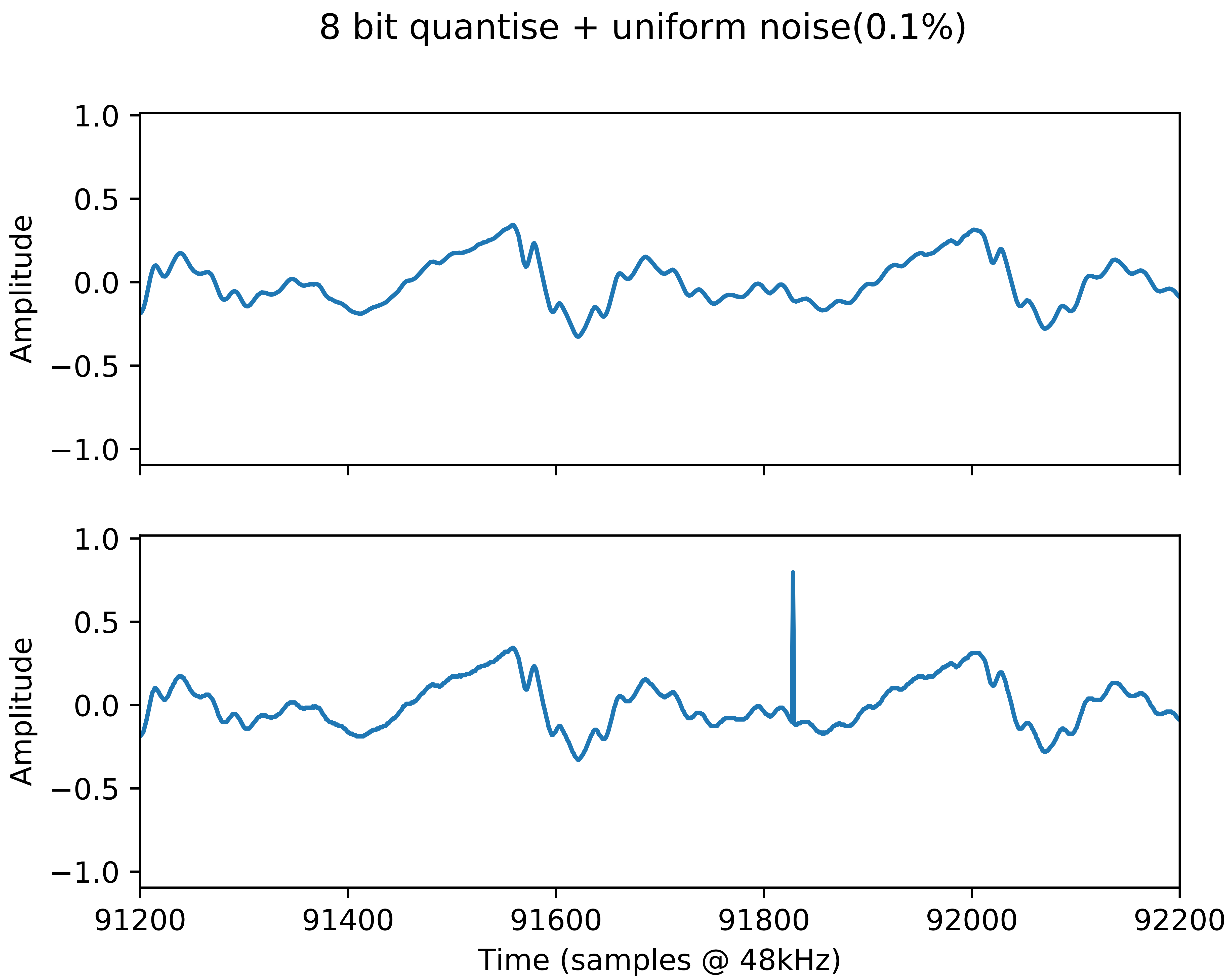
Quantised waveform - 6 bit



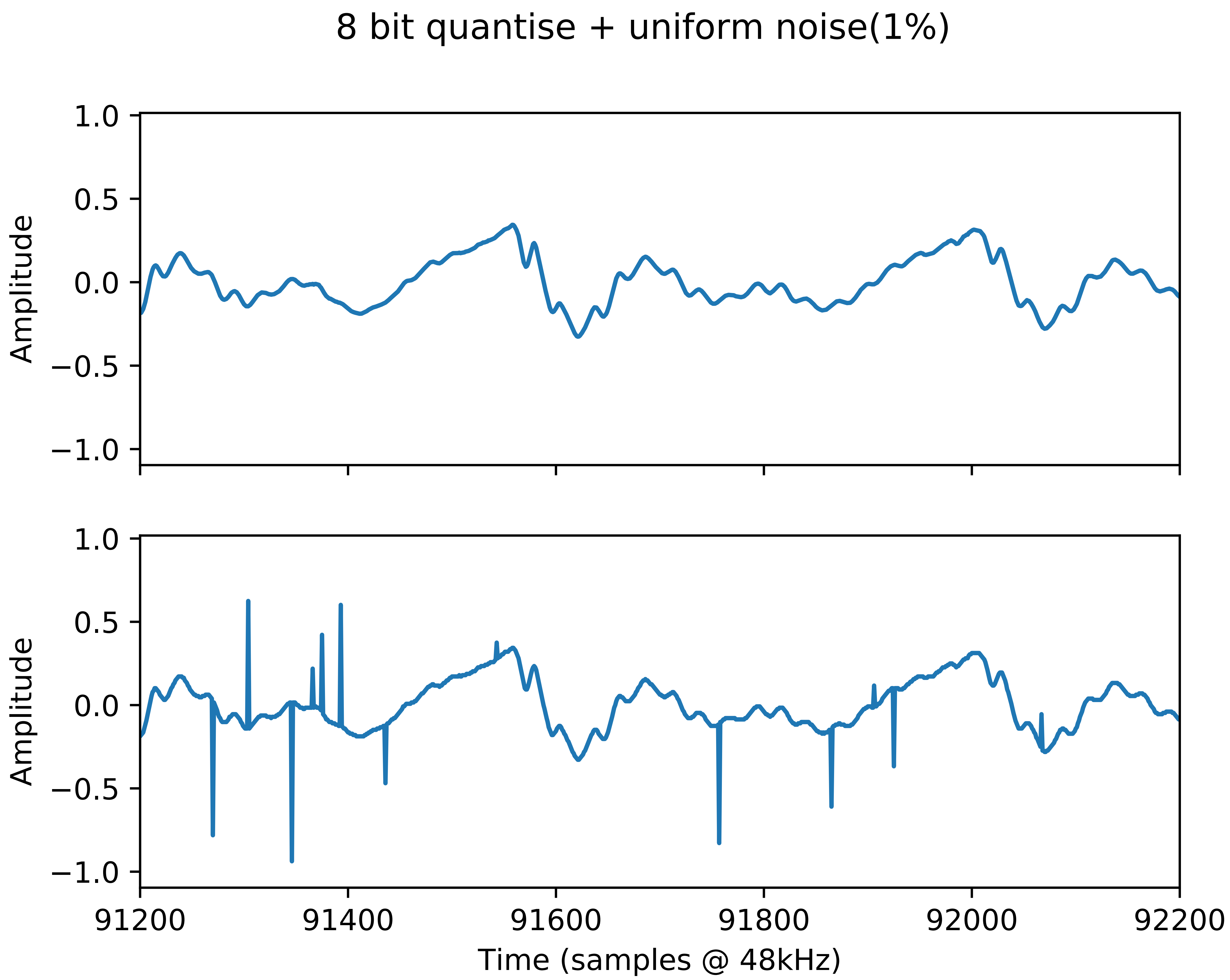
Quantised waveform - 4 bit



Quantised waveform - 8 bit - corrupt 0.1% of samples



Quantised waveform - 8 bit - corrupt 1% of samples



Some things to consider

- The choice of speech parameterisation affects many things
  - **quality** of synthetic speech, obviously
  - the **perceptual consequences** of modelling errors (which will always be present)
  - the available choices for the **objective (loss) function** of your chosen machine learning method (e.g., DNN)
    - *we have no idea what the error surface looks like !*

- The **shape of the error surface** is important for successfully learning a model from data
  - parameter initialisation, convergence properties, sensitivity to design choices and hyper-parameters, … (*SGD can be tricky to tune on hard problems - cf keynote by Francis Bach*)

Objective vs subjective error

- **Objective** measures
  - image/video reconstruction: PSNR, SSIM,…
  - machine translation / text summarisation: Bleu, METEOR, Rouge,…
  - speech transmission: PESQ, POLQA,…
  - speech synthesis: spectral distortion, F0 mean square error & correlation,…
- But these are not the same as **subjective** error (i.e., as perceived by a human)

INTERSPEECH 2015

Towards minimum perceptual error training for DNN-based speech synthesis

Cassia Valentini-Botinhao, Zhizheng Wu, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom

cvbotinh@inf.ed.ac.uk {zhizheng.wu, simon.king}@ed.ac.uk

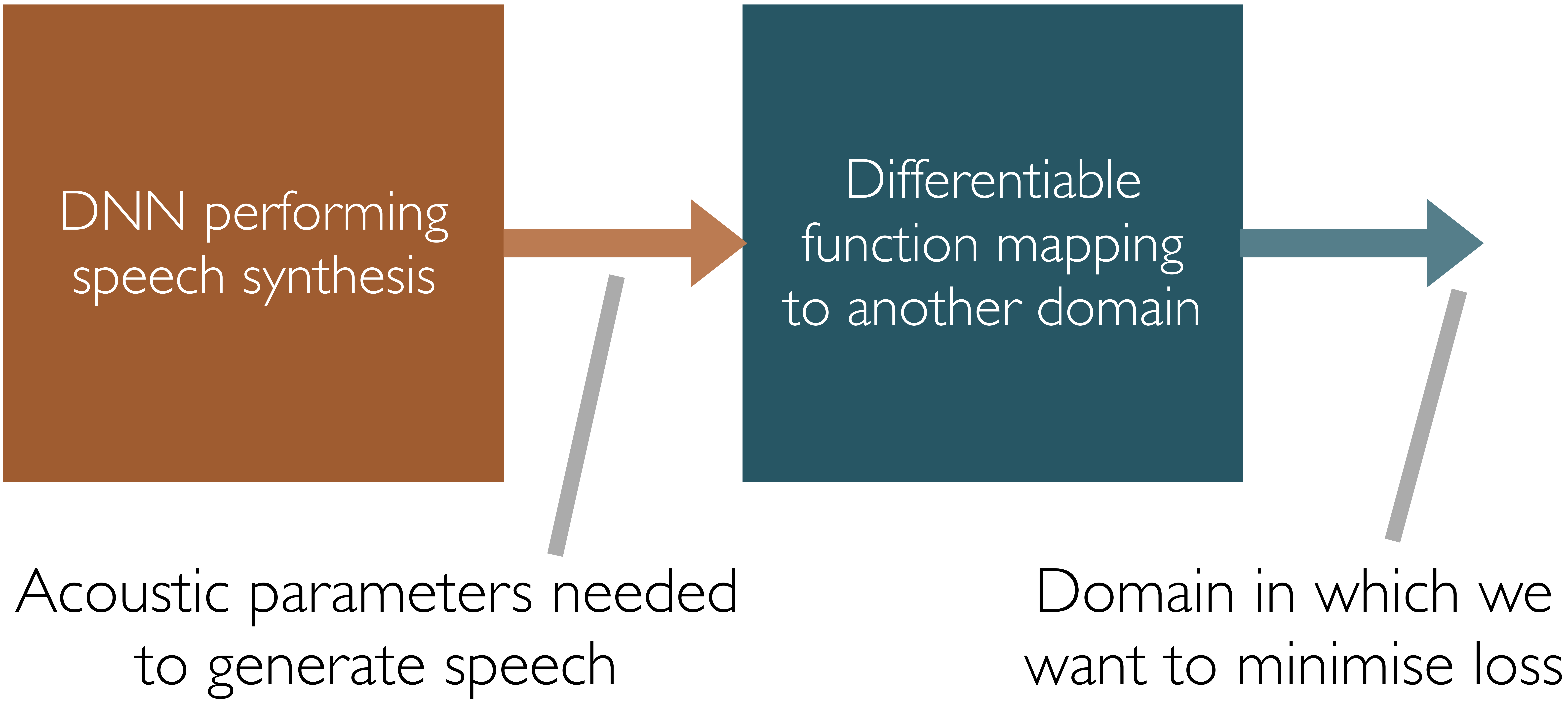
Abstract

We propose to use a perceptually-oriented domain to improve the quality of text-to-speech generated by deep neural networks (DNNs). We train a DNN that predicts the parameters required for speech reconstruction but whose cost function is calculated in another domain. In this paper, to represent this perceptual domain we extract an approximated version of the Spectro-Temporal Excitation Pattern that was originally proposed as part of a model of hearing speech in noise. We train DNNs that predict band aperiodicity, fundamental frequency and Mel cepstral

mised using a shared cost function, allowing the model potentially to learn dependencies between output parameters.

DNN training easily allows for different cost functions to be used. It is possible to train a DNN to predict Mel cepstral coefficients but to calculate the error in the higher-dimensional spectral domain, simply by reformulating the cost function. It is also possible to train a DNN to predict the spectrum directly.

There are, however, more perceptually relevant representations of speech that could be used to measure the error, but that do not allow for synthesis. So, we might measure the error not


DNN performing speech synthesis
Differentiable function mapping to another domain
Acoustic parameters needed to generate speech
Domain in which we want to minimise loss

Are Generative Adversarial Networks the answer ?

- Typically, the adversary is trying to **discriminate** between natural and synthetic speech
- The generative network is learning both
  - to do speech synthesis
  - to fool the adversary into classifying its output as 'natural'

- Unfortunately, there is no guarantee that the adversary will do its job in a **perceptually-relevant** way
  - e.g., discrimination might be possible from **inaudible** properties of the speech signal
  - the generative network might learn to beat the adversary, but the adversary is still only an **objective measure** and **not a human listener**

DOI: 10.21437/Interspeech.2017-962

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Generative Adversarial Network-based Postfilter for STFT Spectrograms

Takuhiro Kaneko[1], *Shinji Takaki*[2], *Hirokazu Kameoka*[1], *Junichi Yamagishi*[2]

[1]NTT Communication Science Laboratories, NTT Corporation, Japan
[2]National Institute of Informatics, Japan
{kaneko.takuhiro,kameoka.hirokazu}@lab.ntt.co.jp, {takaki,jyamagis}@nii.ac.jp

Abstract

We propose a learning-based postfilter to reconstruct the high-fidelity spectral texture in short-term Fourier transform (STFT) spectrograms. In speech-processing systems, such as speech synthesis, conversion, enhancement, separation, and

nents. A Wiener filter provides a conservative way of separating out a speech signal from a mixture signal so that the sum of the separated signals is ensured to be equal to the mixture; however, it often produces artifacts perceived as time-varying tones known as musical noise. To reduce artifacts or musical noise in processed speech, postprocessing methods using cep-

The take-home message

- We do not have very sophisticated models of human perception
- The best we can do at the moment is to minimise loss in an **appropriate domain**
- That means we still have to **choose** our signal representation carefully
  - **that's feature engineering !**

- GANs minimise loss in a **different domain** to the acoustic features - **very clever** !

- But, the adversary is **not constrained** to behave like a human listener - **less clever** !
  - …. so, can you find a way to do that…. ?

Where did the signal processing go?

Simon King

CSTR website: `www.cstr.ed.ac.uk`

Teaching website: `speech.zone`