# ELEC-C5220
# Lecture 5:
# Metrics and loss functions

## Machine learning in information technology

Aalto University
School of Electrical
Engineering

Lauri Juvela

8.2.2024

# Clarifications to Exercise 04

- **Do not use the state parameters self.c0 and self.h0 for anything (these should have been removed)**
- **LSTM uses a default all-zero initial state when passed None**

```
class LSTMModel(torch.nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(LSTMModel, self).__init__()
        self.lstm = torch.nn.LSTM(input_size, hidden_size, batch_first=False)
        self.linear = torch.nn.Linear(hidden_size, output_size)

        self.c0 = torch.nn.Parameter(torch.zeros(1, 1, hidden_size))
        self.h0 = torch.nn.Parameter(torch.zeros(1, 1, hidden_size))

    def forward(self, x, state_0=None):
        """
        Args:
            x: input tensor of shape (batch_size, input_size, timesteps)
            state_

        Returns:
            y: output tensor of shape (batch_size, output_size, timesteps)
            state_out: tuple containing (h_out, c_out)
        """
        # YOUR CODE HERE
        raise NotImplementedError()
        return out, state_out
```

# Clarifications to Exercise 04

- Process the frame one time-step at a time

(batch, C=1, frame_len)

- Processing the whole frame as single time-step is possible, but don't

(batch, frame_len, T=1)

```
model = torch.compile(model)
iteration = 0
losses_ma = []
while iteration < max_iterations:
    state = None # initial state
    for j in range(segment_len // frame_len):
        # create input_frame by slicing waveform_input
        # create target_frame by slicing waveform_target
        # YOUR CODE HERE
        raise NotImplementedError()

        output_frame, (h_0, c_0) = model(input_frame, state)
        # detach gradient tracking from state for next iteration
        # YOUR CODE HERE
        raise NotImplementedError()

        if j == 0:
            continue

        loss = criterion(output_frame, target_frame)
```

# Lecture 05 content

- Metrics and loss functions
- Differentiable programs and requirements for useful gradients
- Subjetive evaluation
- Objective metrics and loss functions for speech and audio

# Metrics and loss functions

- Both are used for evaluating how well the a machine learning method performs
- Sometimes they can be the same, but now always.
- What is the difference?
- Loss function needs to provide useful learning signals to adjust parameters
- Metric should be somehow easy to interpret by humans

# Metrics requirements

- Intuitive and interpretable for humans
- Correlates with human perception
- Numerically stable computation

# Loss function requirements

- ~~Intuitive and interpretable for humans~~
- ~~Correlates with human perception~~
- Numerically stable for forward and *backward* computation
- Differentiable and has useful gradients
- Fast to compute! Needs to be computed at every iteration

# Accuracy as a metric

- Make classifications and count the number of correct classifications
- Easy to interpret
- Compare this to the cross-entropy numbers we saw in the exercises previously

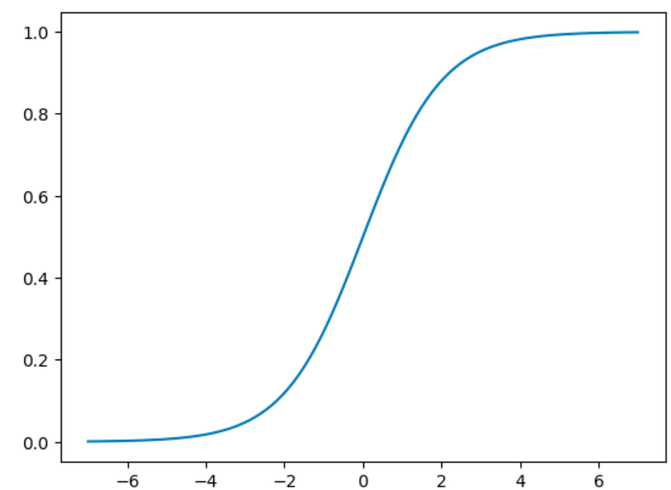
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Accuracy} = \frac{\#\ \text{correct classifications}}{\#\ \text{total classifications}}$$

# Accuracy as a loss-function?

- **A binary decision function outputs class value 1 when its input is above the decision boundary (0.5 in this case)**
- **These outputs are needed for counting!**

$$f(x) = \begin{cases} 1 & , x > 0.5 \\ 0 & , x \leq 0.5 \end{cases}$$



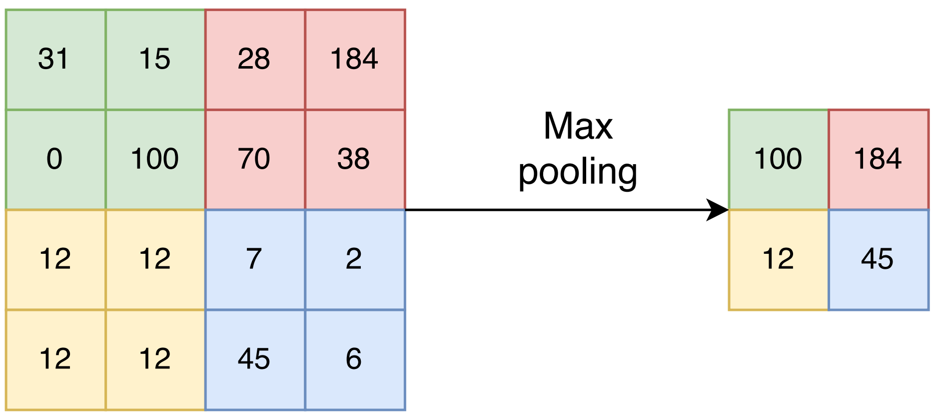
# Accuracy as a loss-function?

- **The decision function does not have useful gradients**
- **This problem extends to sampling from binary and categorical distributions: what if we want to use the outcome of a coin-flip as input to another model?**

$$f(x) = \begin{cases} 1 & , \ x > 0.5 \\ 0 & , \ x \leq 0.5 \end{cases}$$

$$\frac{\partial f(x)}{\partial x} = \begin{cases} 0 & , \ x > 0.5 \\ 0 & , \ x < 0.5 \\ ? & , \ x = 0.5 \end{cases}$$

# Max operation is differentiable




$$\max(x) = x_i, \text{ if } x_i >= x_j, \forall j$$

$$\frac{\partial \max(x)}{\partial x_j} = \begin{cases} 1 & j = i \\ 0 & \text{otherwise} \end{cases}$$

# Argmax is not differentiable

- **Argmax is used when picking the most likely class from class probabities**
- **Decisions, decoding and sampling from categorical distributions is not generally differentiable**

$$\max(x) = x_i, \text{ if } x_i >= x_j, \forall j$$

$$\frac{\partial \max(x)}{\partial x_j} = \begin{cases} 1 & j = i \\ 0 & \text{otherwise} \end{cases}$$

$$\operatorname{argmax}(x) = i, \text{ if } x_i \geq x_j, \forall j$$

$$\frac{\partial \operatorname{argmax}(x)}{\partial x_j} = \begin{cases} 0 & j = i \\ 0 & \text{otherwise} \end{cases}$$

# Word Error Rate

- **Common metric in Automatic Speech Recognition (ASR)**
- **Intuitive: compare model output to reference text and count the number of correctly recognised words**
- **No useful gradient – not directly usable as a loss function**

$$\text{WER} = \frac{S + D + I}{N}$$

$$N = \text{Number of words in reference}$$

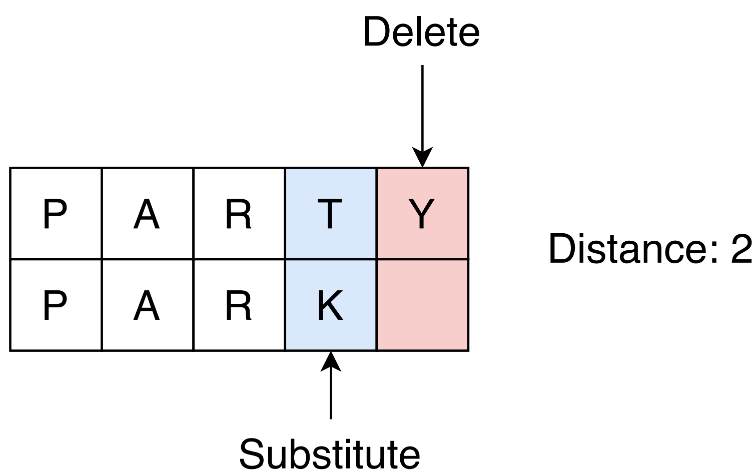
$$S = \text{Number of substitutions}$$

$$D = \text{Number of deletions}$$

$$I = \text{Number of insertions}$$

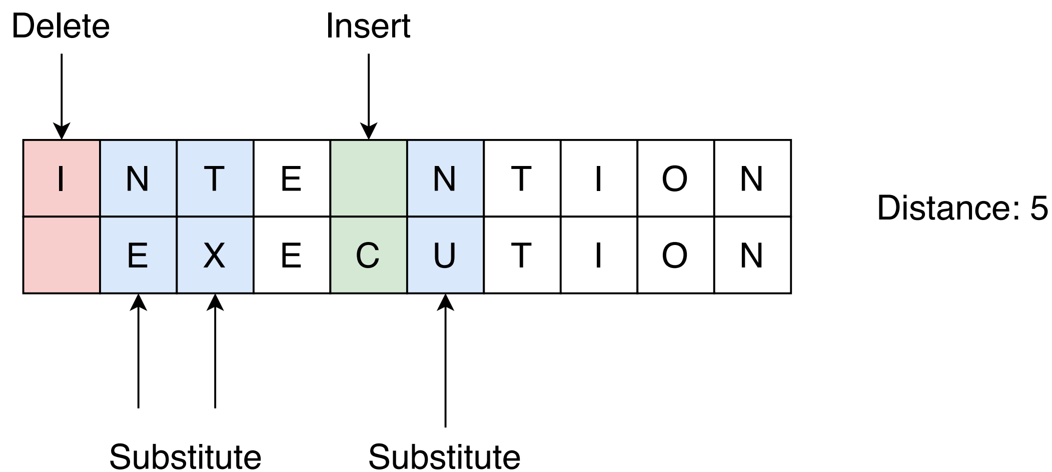
# Character Error Rate (CER)

- **Same distance metric as WER, but on characters**
- **Also known as the Levenshtein edit distance**



# Character Error Rate (CER)

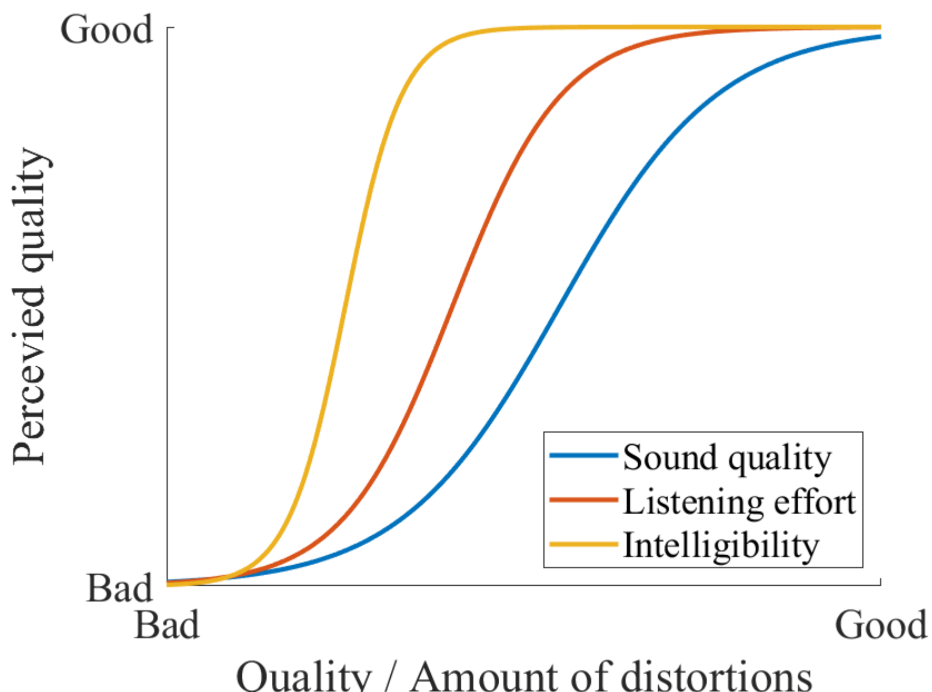
- Same distance metric as WER, but on characters
- Also known as the Levenshtein edit distance



# Subjective evaluation

- **Gold standard for measuring system performance in many deep learning applications: only humans can judge when the model is good enough**
- **Very expensive and noisy to measure; not differentiable**
- **Applications: speech synthesis, coding, enhancement, audio effects modeling etc.**
- **Evaluating generative model outputs is also subjective**
  - RLHF – ChatGPT is trained using reinforcement learning from human feedback

# Subjective quality depends on the context and question



# A-B preference testing

- Which do prefer A or B?
- Number of pairings grows quickly when comparing multiple systems

# ABX testing

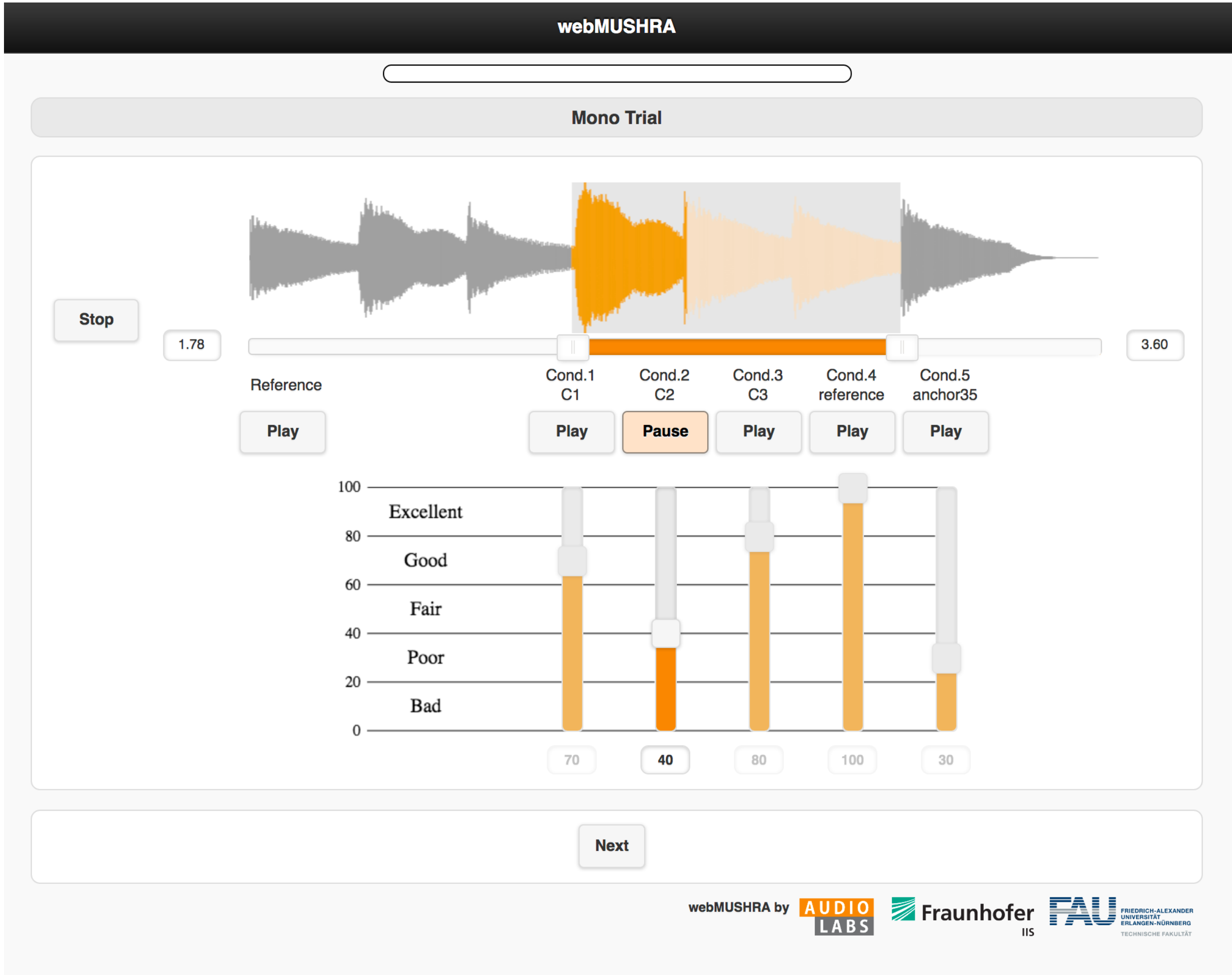
- **Version 1:**
  - Here is a test sample X and reference samples A and B
  - Is X the same as A or B?
- **Version 2:**
  - Here are three samples, find which on is the odd one out?

# Mean opinion score (MOS)

- **Rate the quality (naturalness) of the following sample on a five level scale**
  - 5: Excellent
  - 4: Good
  - 3: Fair
  - 2: Poor
  - 1: Bad
- **Mean opinion scores are averaged over multiple subjects and test items**

# MUSHRA tests

- Multiple stimulus
- Hidden reference
- Hidden anchor



# Objective metrics

- **Emulate the preceptual relevance of subjective evaluation**
- **Can be actually computed**
  - Differentiable?

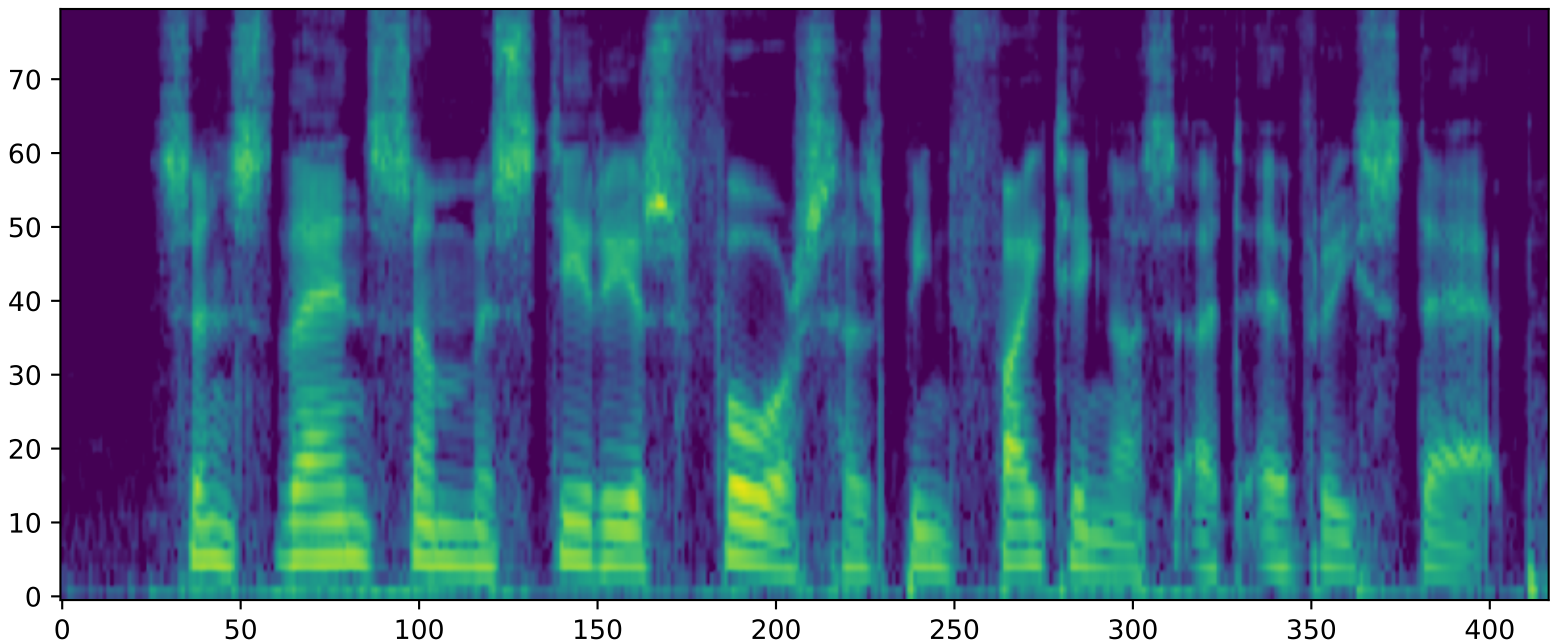
# Signal-to-Noise Ratio (SNR)

- Classic signal processing metric
- Compare the signal energy with noise energy
- In deep learning loss functions, noise is equivalent to model error

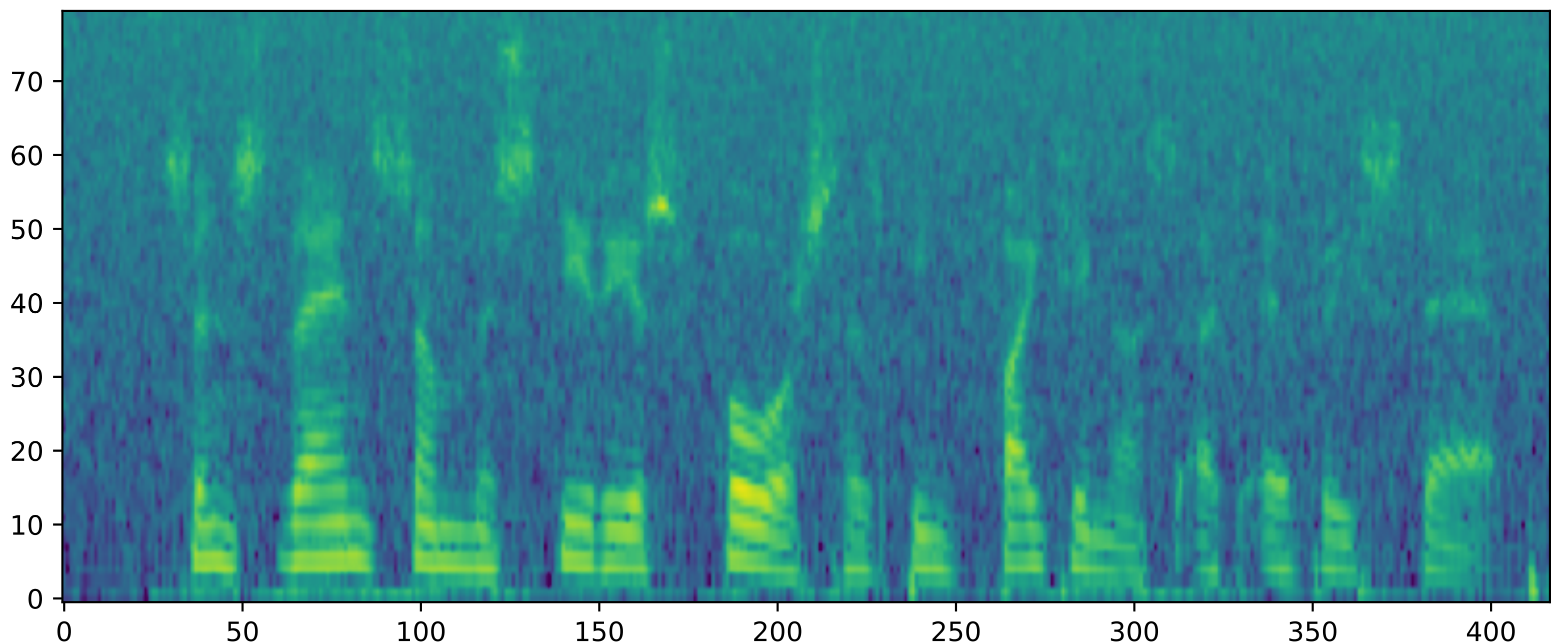
$$\mathrm{SNR} = \frac{\sum_t x^2[t]}{\sum_t n^2[t]}$$

$$n[t] = e[t] = x[t] - \hat{x}[t]$$

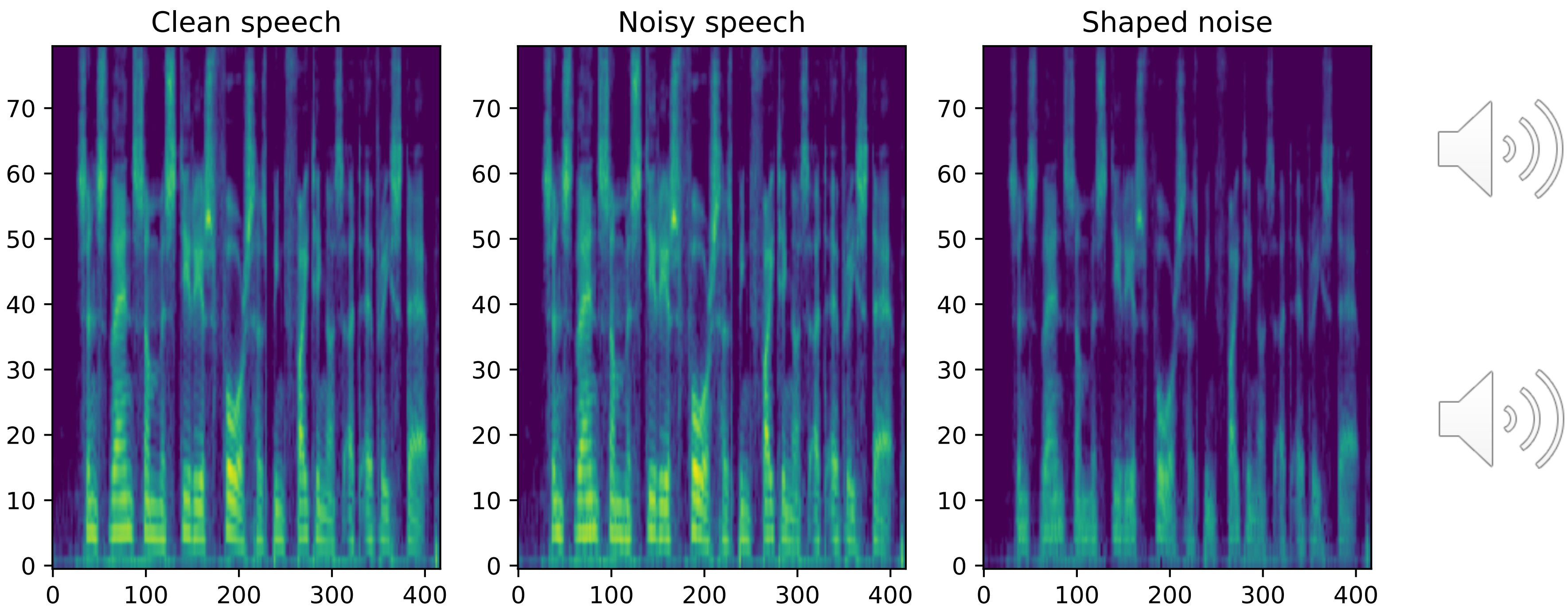
# Clean speech



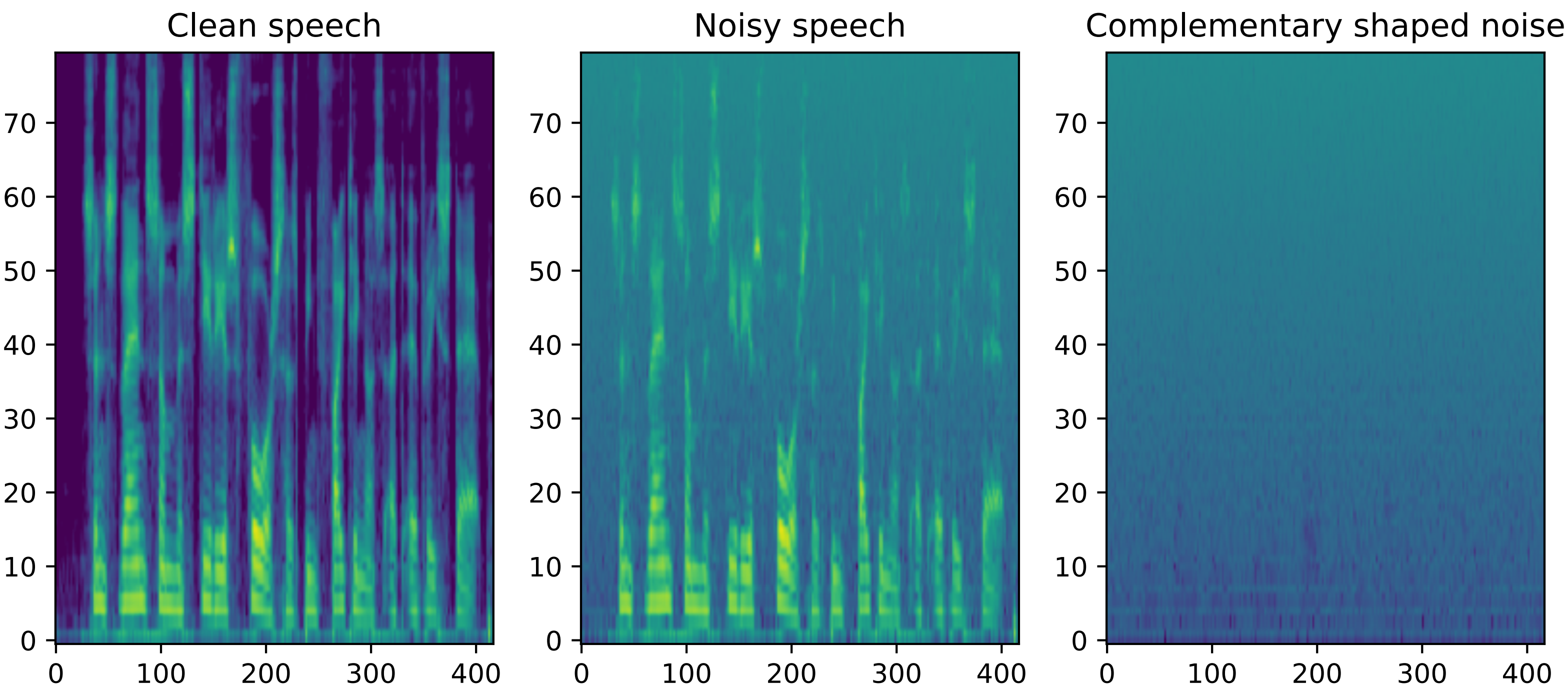
# Noisy speech at 10dB SNR



# Speech shaped noise at 10dB SNR is perceptually masked



# Noisy speech at 10 dB SNR, complementary noise spectrum

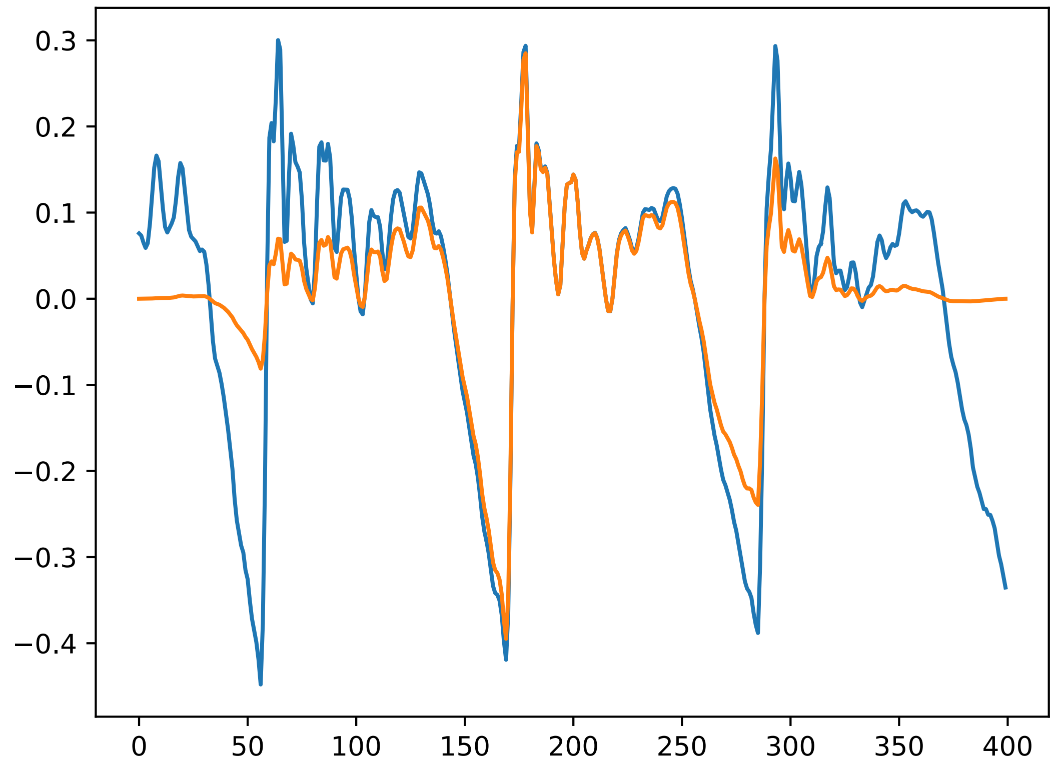


# Perceptual loss functions for audio

- **Mel spectrum is differentiable and has useful gradients**
- **Steps**
  - Framing
  - Windowing
  - FFT
  - Magnitude
  - Mel filterbank
  - log

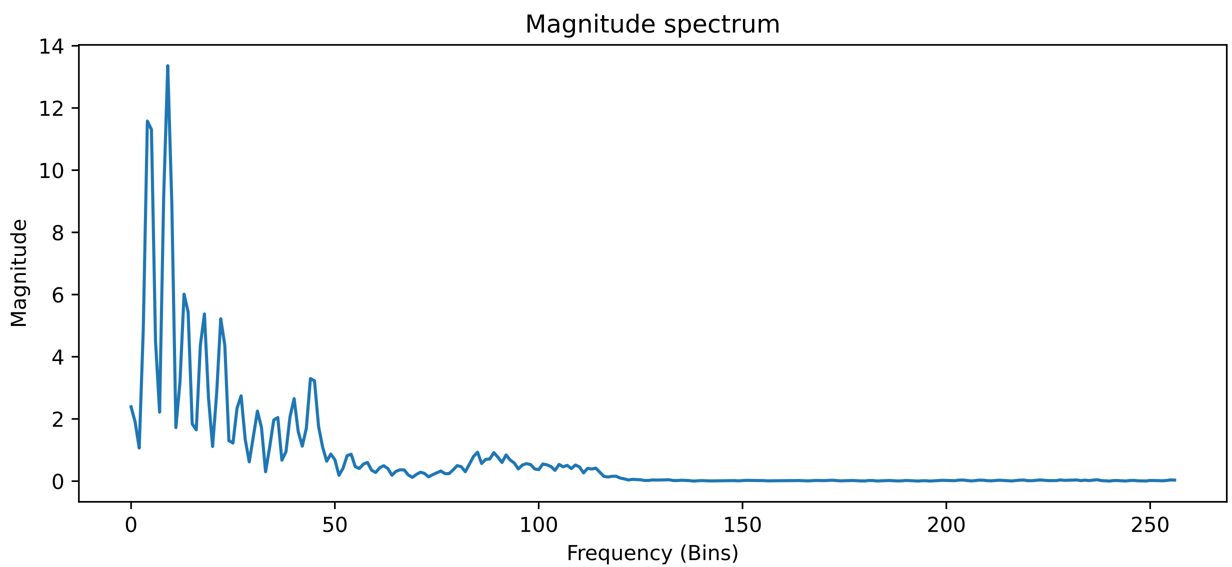
# Framing and windowing

- **Slice waveform into short-time frames**
- **Multiply with a cosine window to taper the edges**



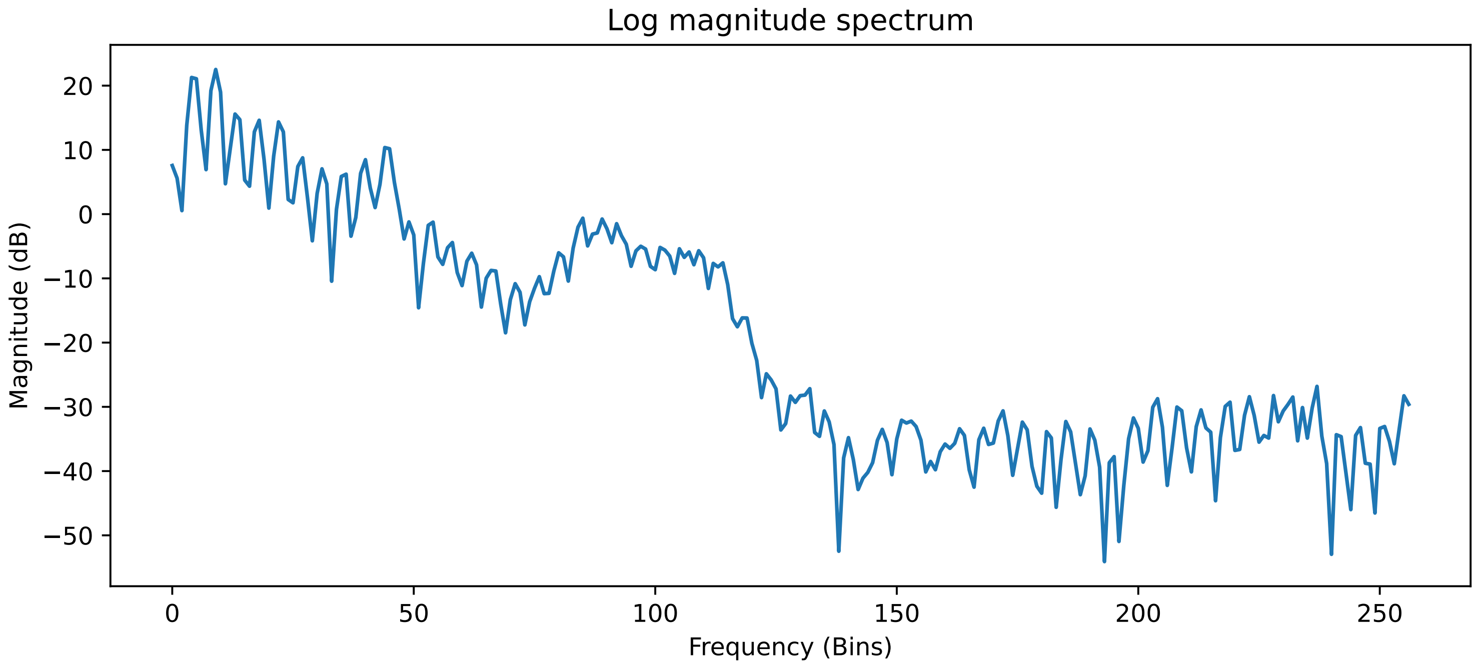
# Short-time spectrum with FFT

- **FFT is linear and differentiable**
- **Magnitude of complex number (absolute value)**



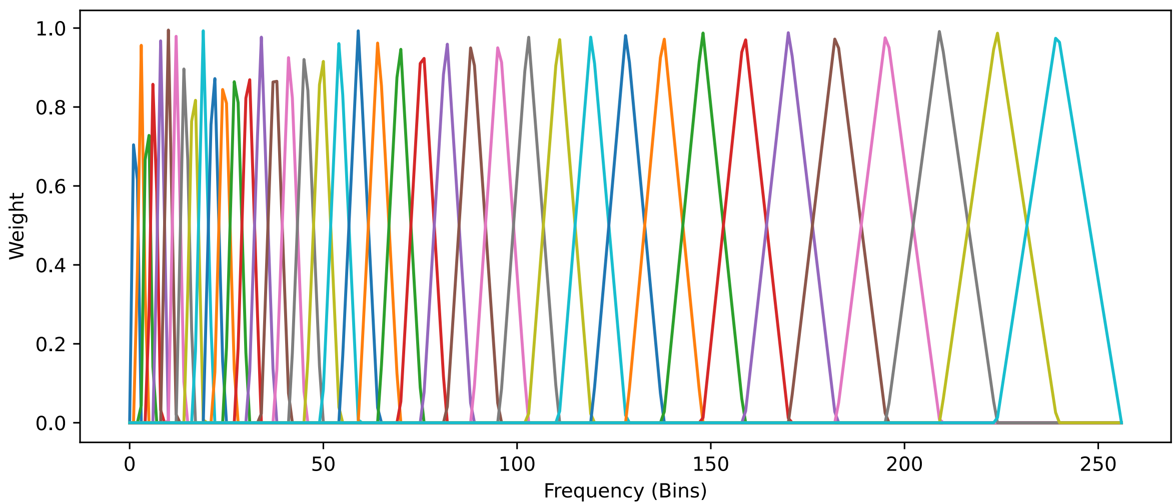

Aalto University
School of Electrical
Engineering

# Magnitude spectrum (dB)

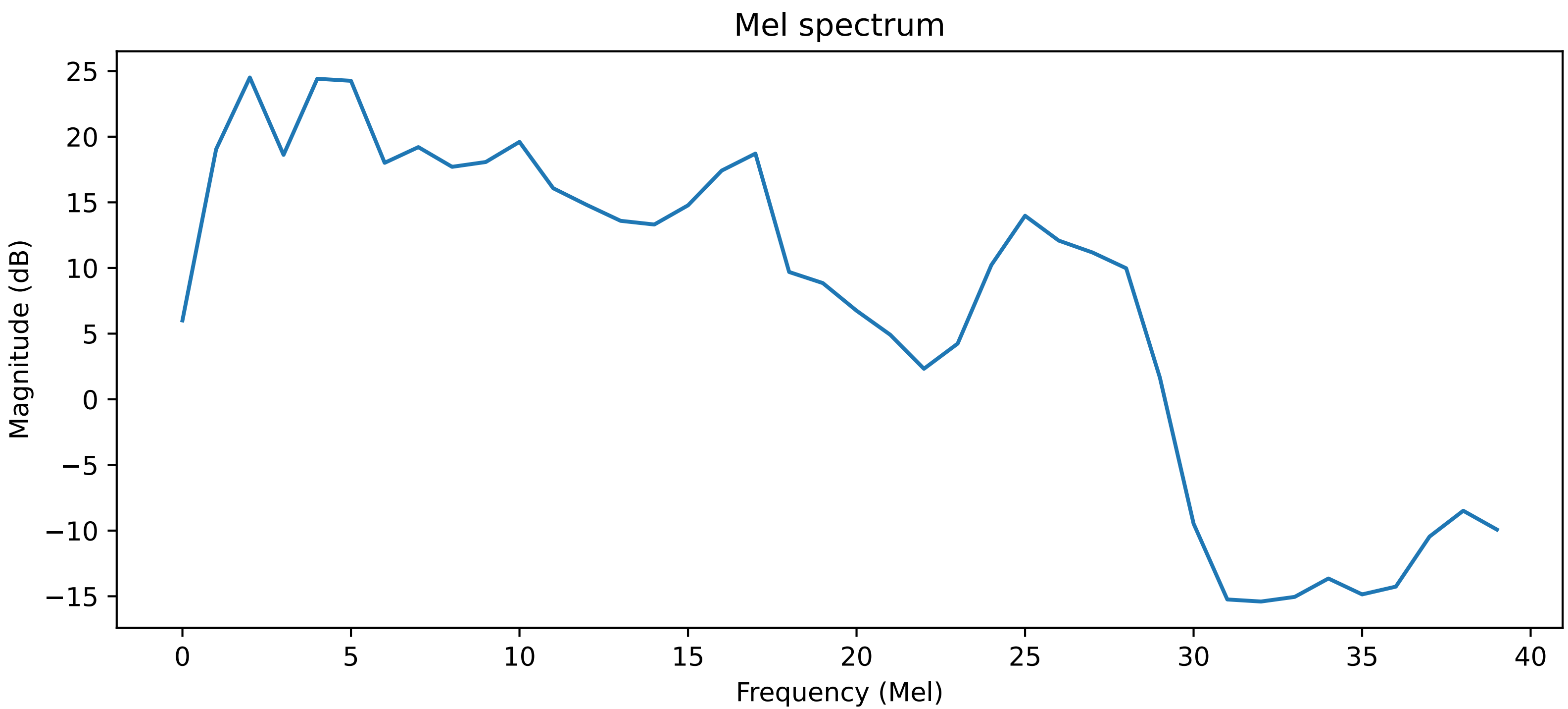


# Mel filterbank

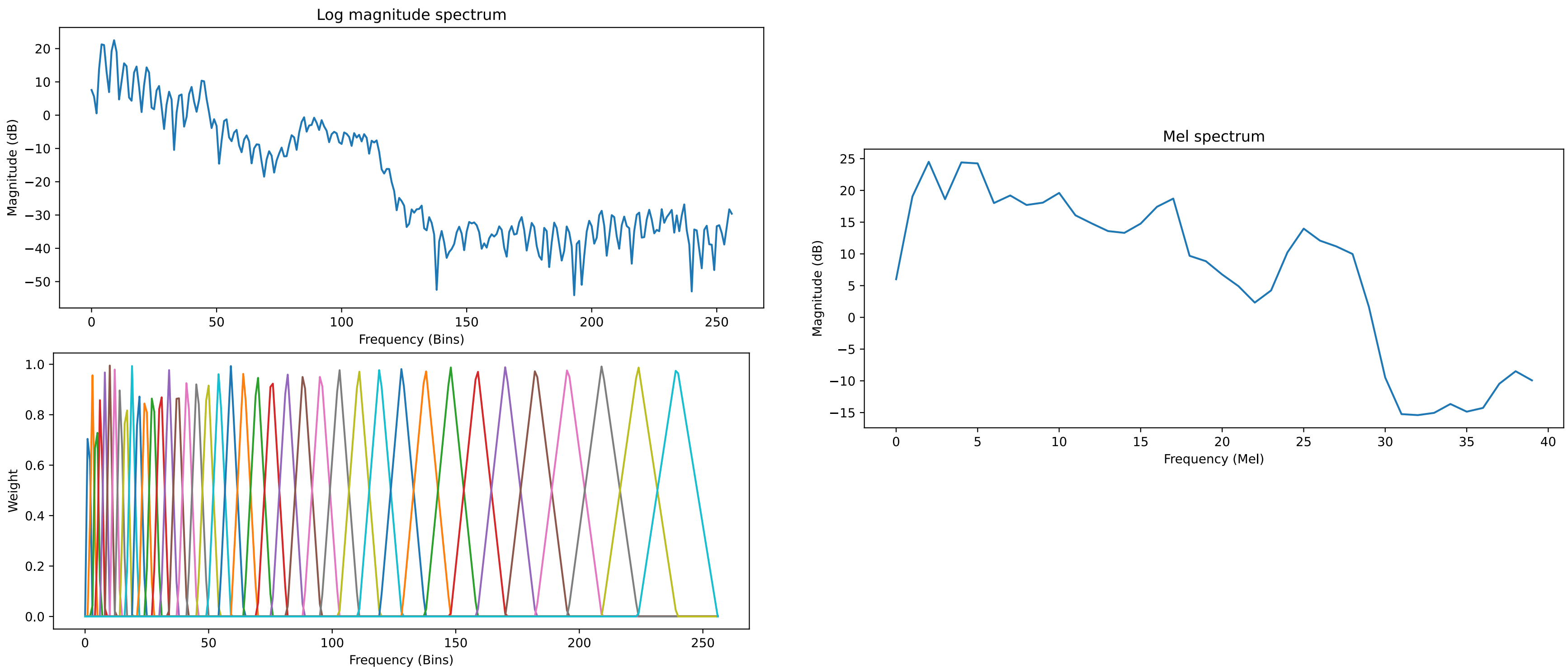
- **Represented by a 40 x 257 matrix, where 40 is the number of mel filter channels and 257 is the number of FFT frequency bins**
- **Linear and differentiable**



# Mel spectrum (dB)



# Multiply linear spectrum with mel filterbank to get mel spectrum

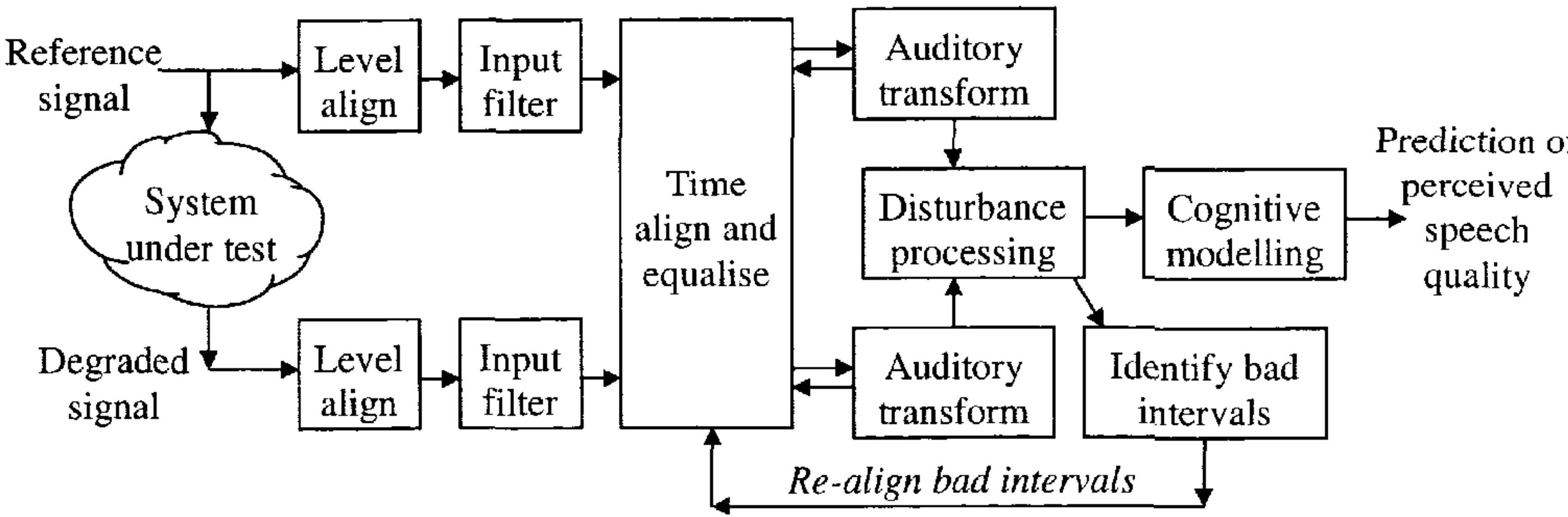



Aalto University
School of Electrical
Engineering

# Mel spectral distance

- **Compute mel spectrogram from model output and reference signals**
- **Use simple distances (MSE, MAE) to compare the spectrograms**
- **Pros:**
    - Differentiable
    - Cheap to compute
- **Cons**
    - No phase sensitivity
    - No perceptual masking model

# Perceptual evaluation of speech quality (PESQ)



# PESQ

- **Pros:**
  - Accurate perceptual model
  - Interpretable output (MOS score from 1 (bad) to 5 (excellent))
  - Differentiable!
- **Cons:**
  - Heavy to compute
  - Requires expert knowledge to judge when PESQ is applicable

# Recommended reading

- **Chapter 6 in the speech processing book**
    - 6.1. on subjective quality evaluation
    - 6.2. on objective quality evaluation
    - 6.4. on analysis of evaluation results

**https://speechprocessingbook.aalto.fi/Evaluation_of_speech_processing_methods.html**

# Lecture 05 recap

- Metrics and loss functions
- Subjective evaluation
- Objective evaluation
- Requirements and trade-offs