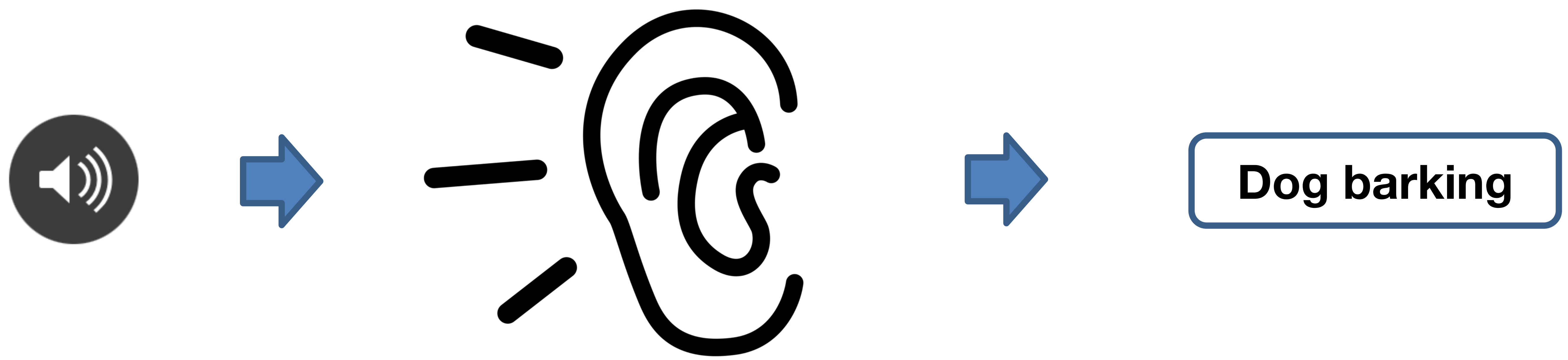
Sound Object Labeling

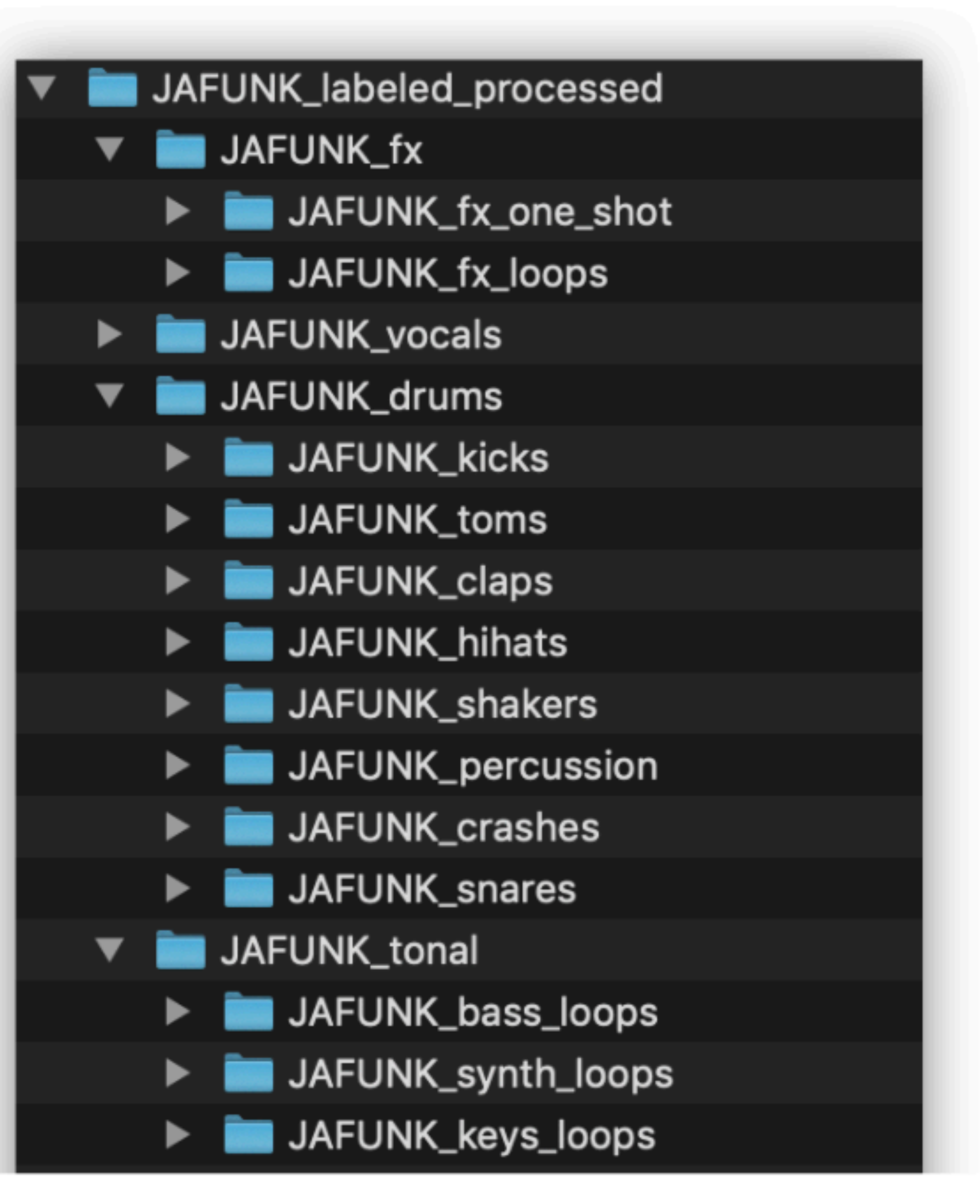
Hugo Flores García

Fall 2021

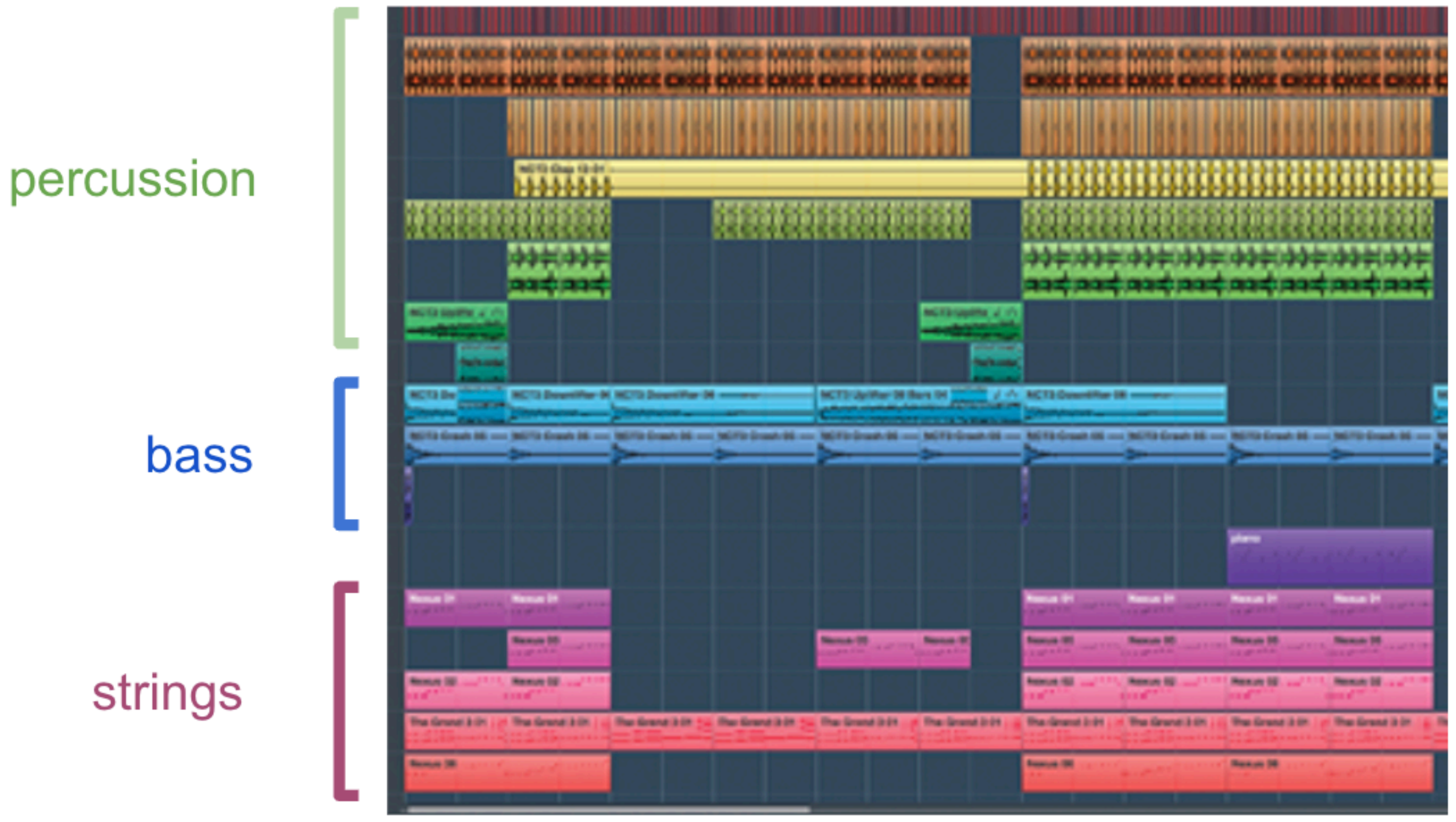
Sound object labeling



Bongjun Kim (Winter 2019)



Organizing large sample libraries



Grouping tracks in a DAW

Navigating large music recordings by content

Goal

- Building a system that automatically labels an audio event

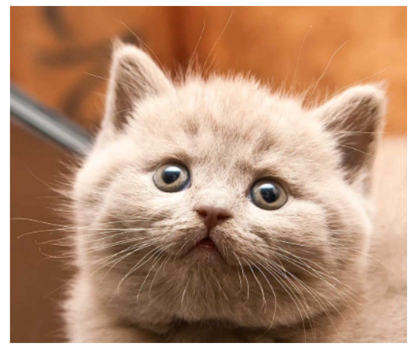
Bongjun Kim (Winter 2019)

Machine Learning: Classification

Bongjun Kim (Winter 2019)

Overview of general classification tasks

Input data ➡ **Feature representation** ➡ **Classifier** ➡ **Label**



A vector of numbers

$\vec{x}$ = $<a_1, a_2, \ldots, a_n>$

…that represent attributes of the example.

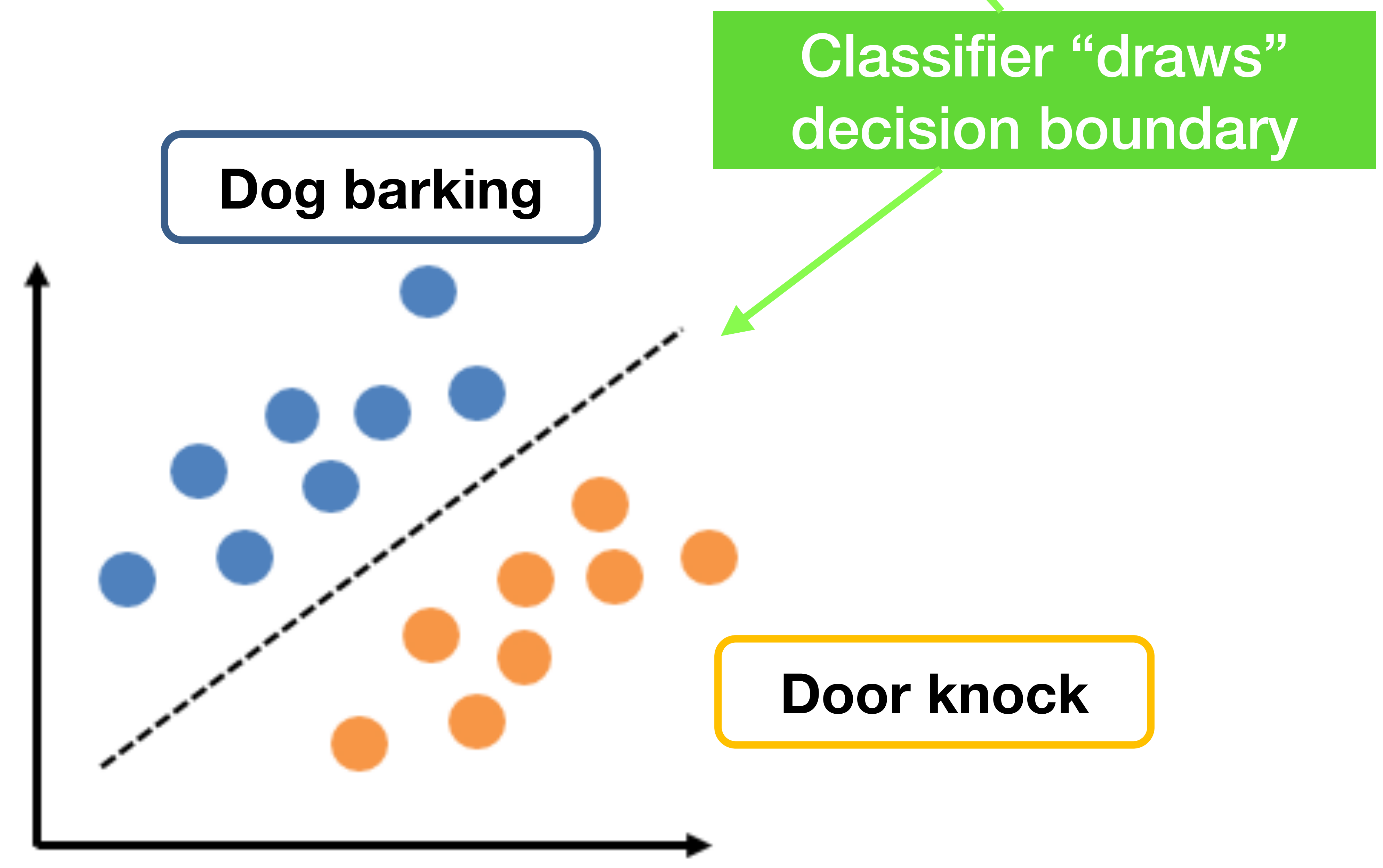
- **Decision Tree**
- **Nearest Neighbor**
- **Neural Networks**

“Cat”

“Piano”

Classification Tasks

Input data ➡ Feature representation ➡ Classifier ➡ Label

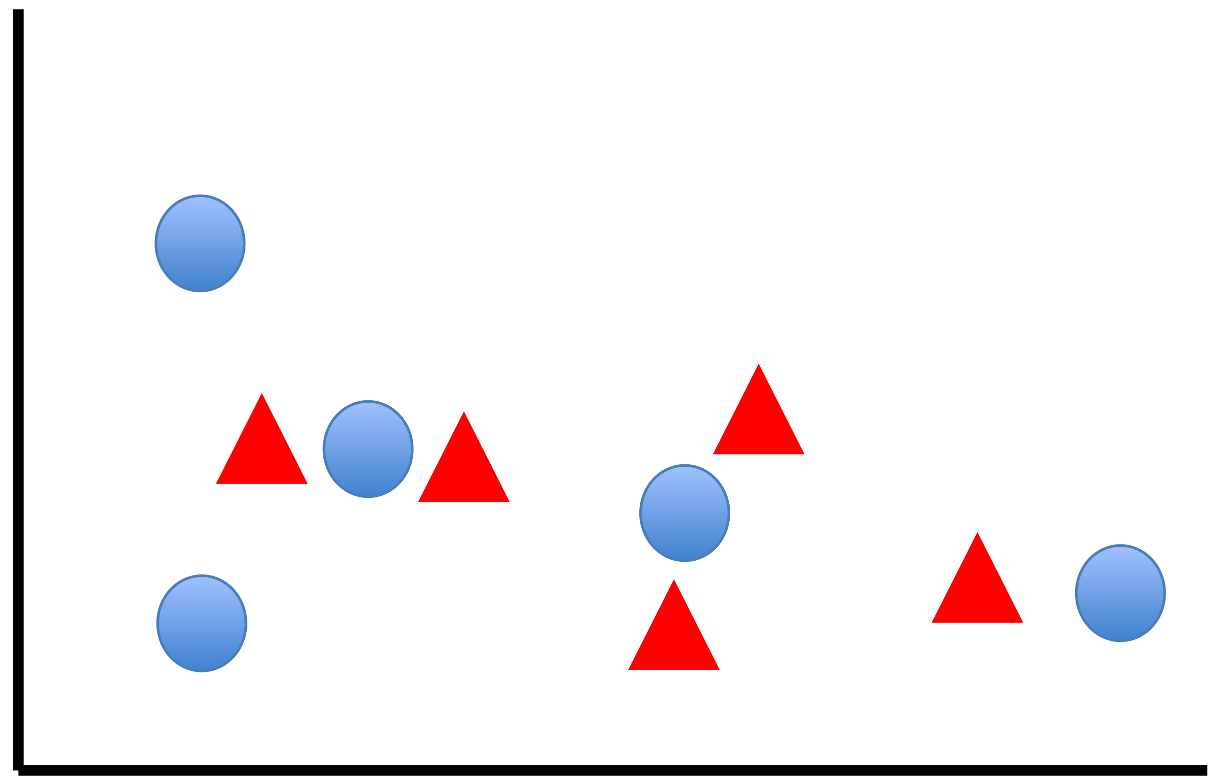


Feature space should easily discriminate between classes

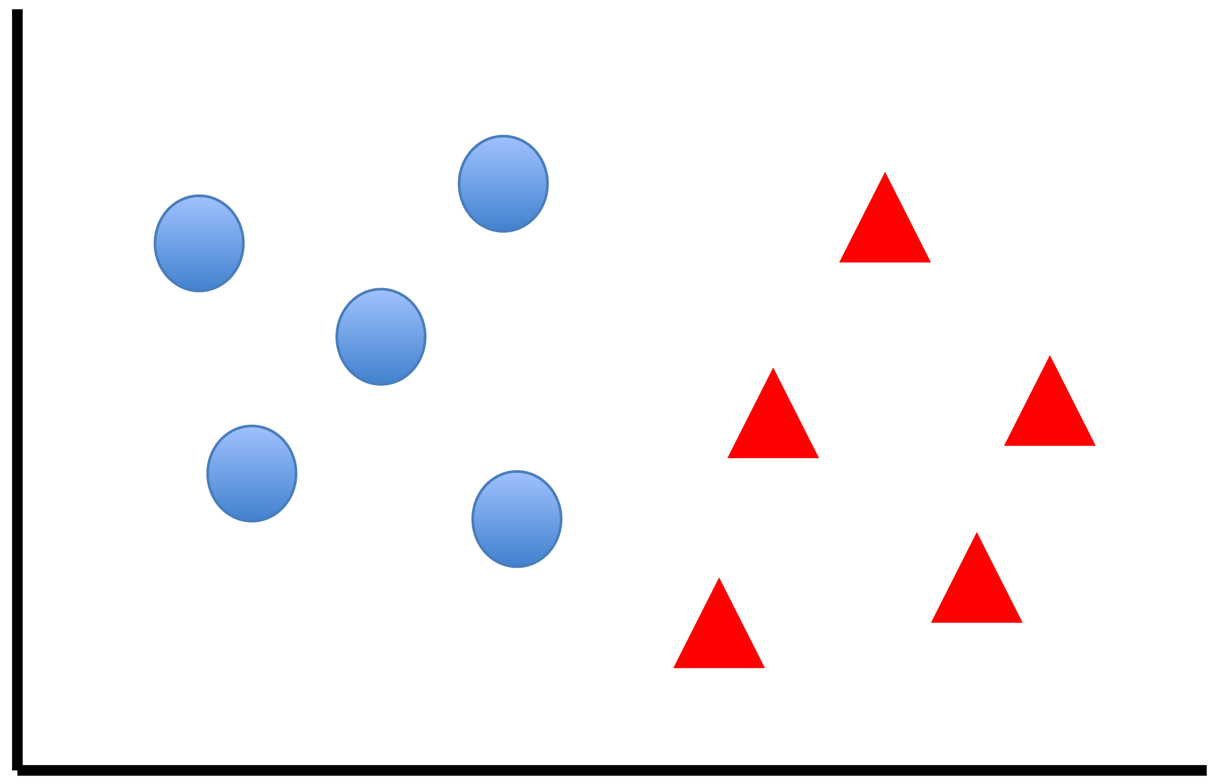
Bongjun Kim and Hugo Flores García

Feature selection is important

- how points in the feature space cluster is important



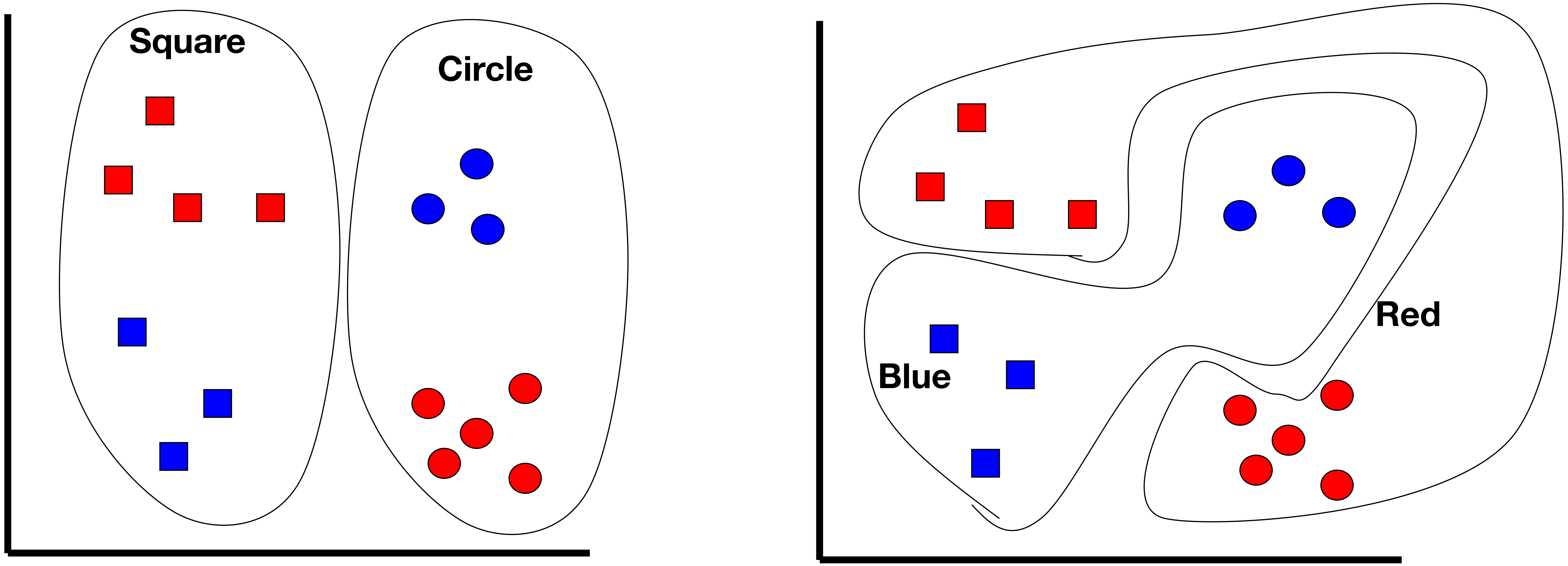
bad feature representation



good feature representation

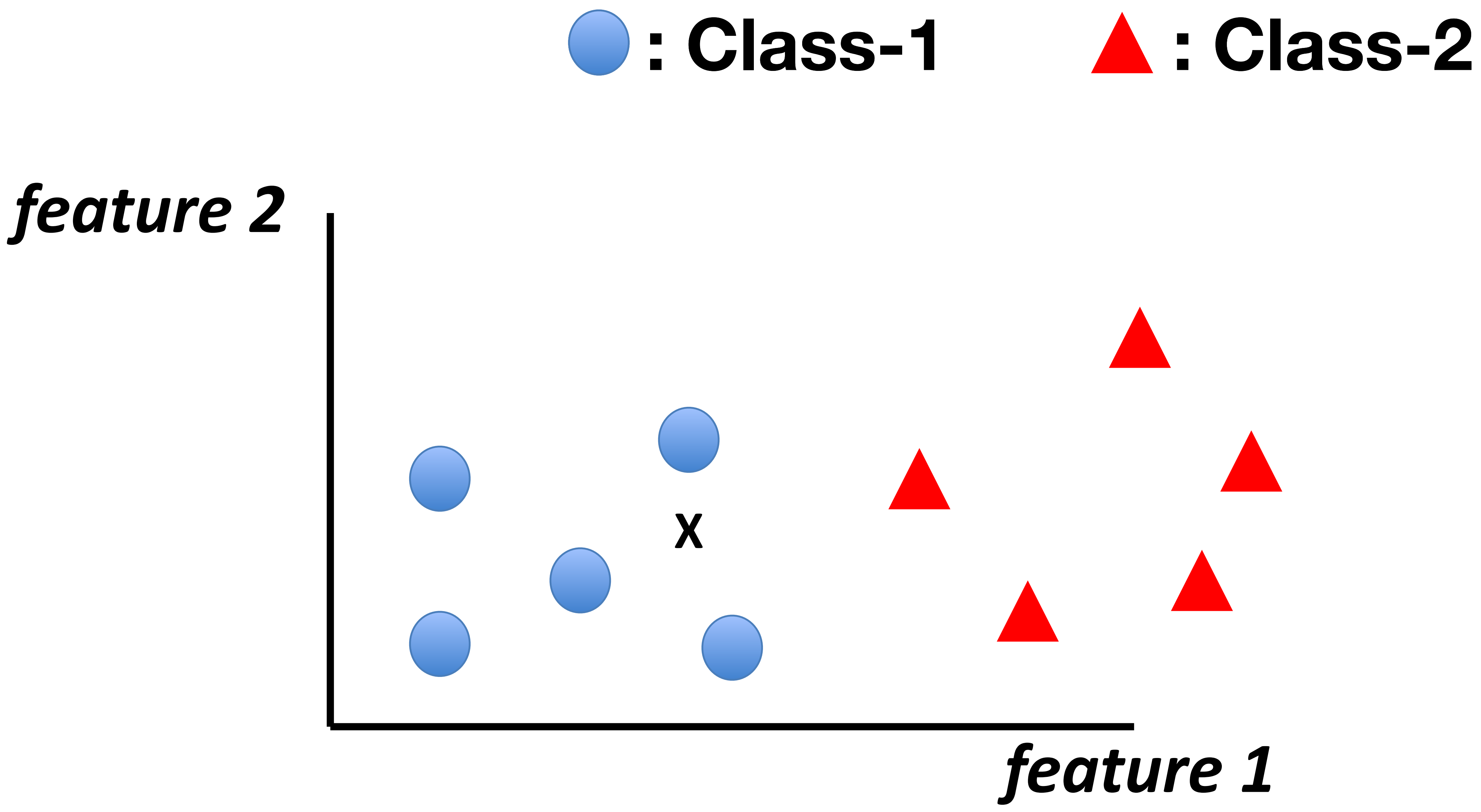
Different Classifiers

The same feature space could be meaningful for different ways of classifying data.

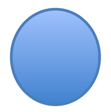


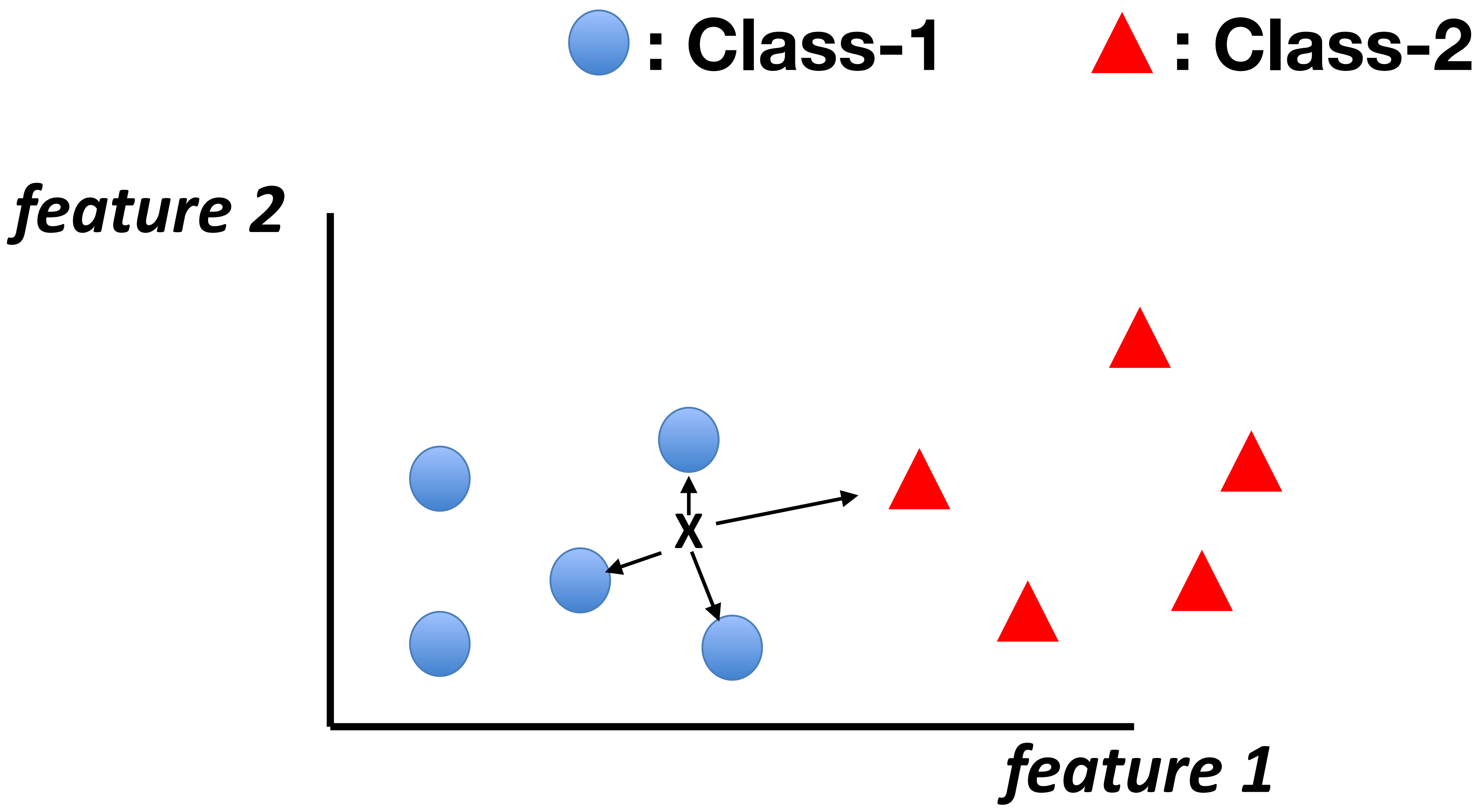
K-Nearest Neighbor (KNN) Classifier

- When you see a new instance x to classify, find **the most similar training example(s)** and assign their label to the instance.
- How do you tell what things are similar?
  1. Extract proper features.
  2. Measure distance / similarity in the feature space.

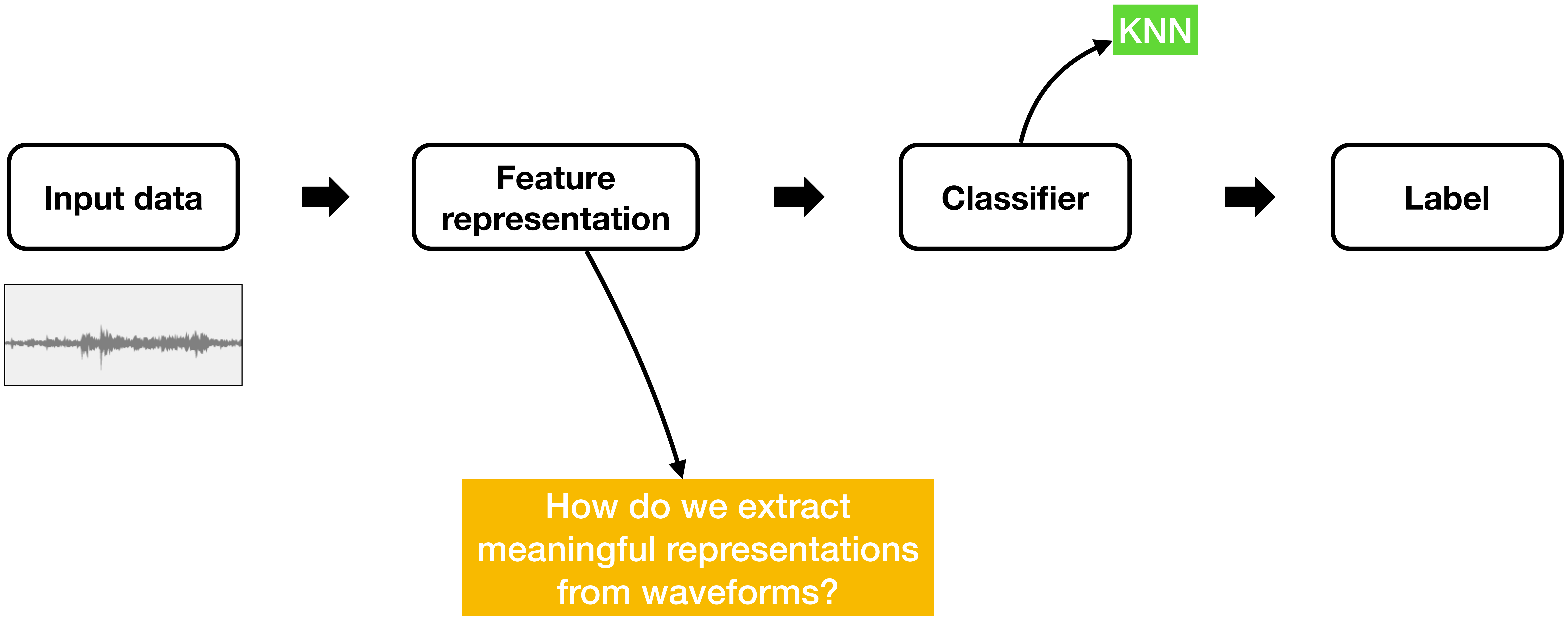


K-Nearest Neighbor (KNN) Classifier

Considering 4 nearest neighbors (k=4),
X is probably a 

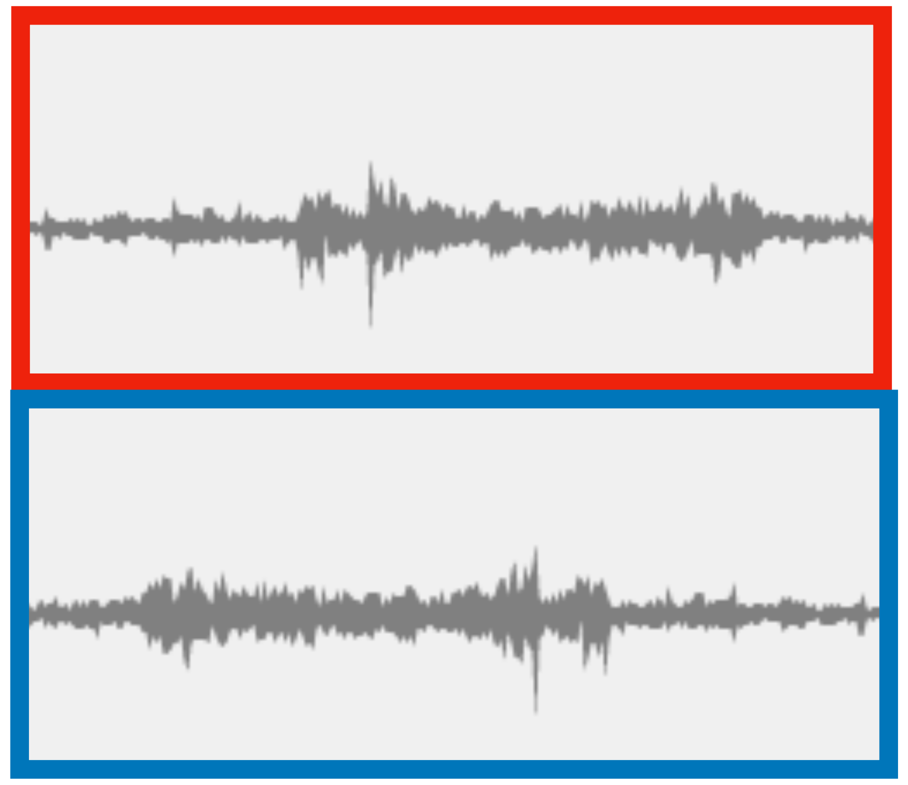


Now that we know..



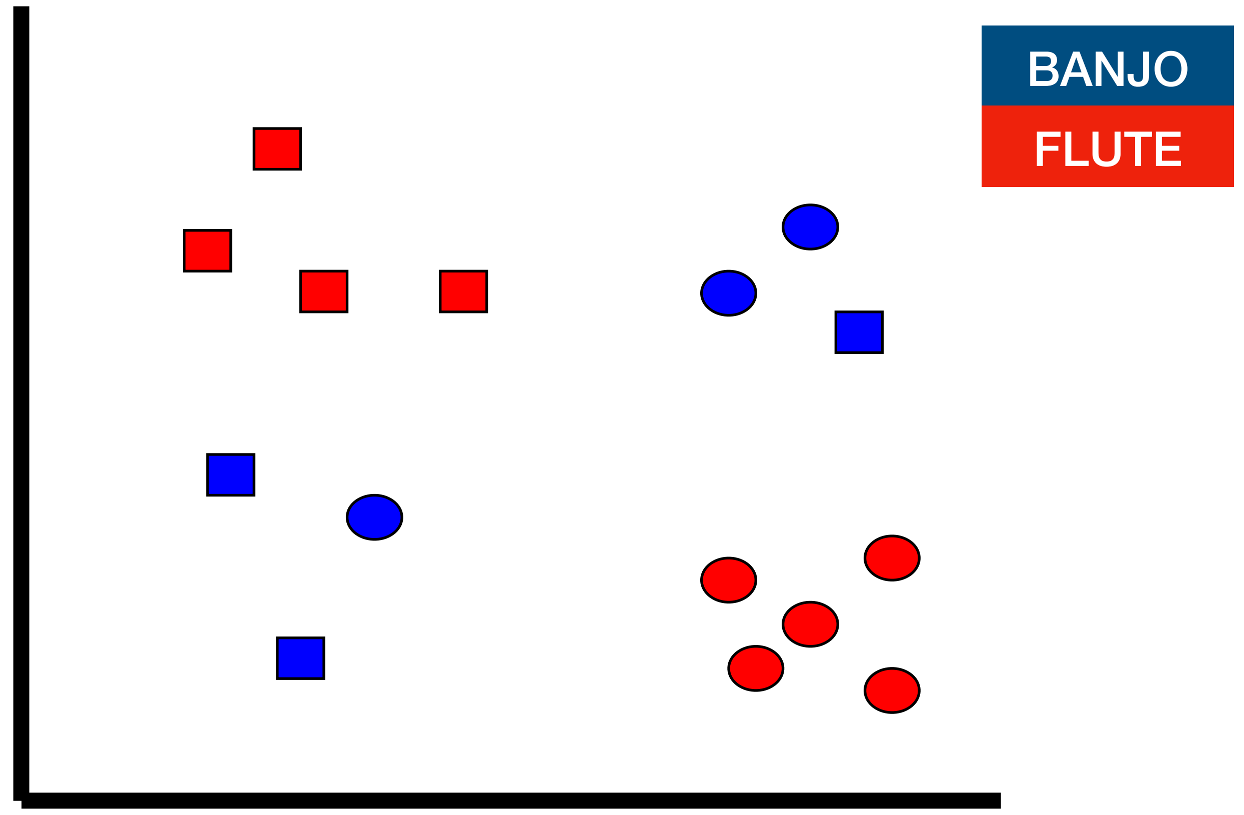
Audio event classification

Input data



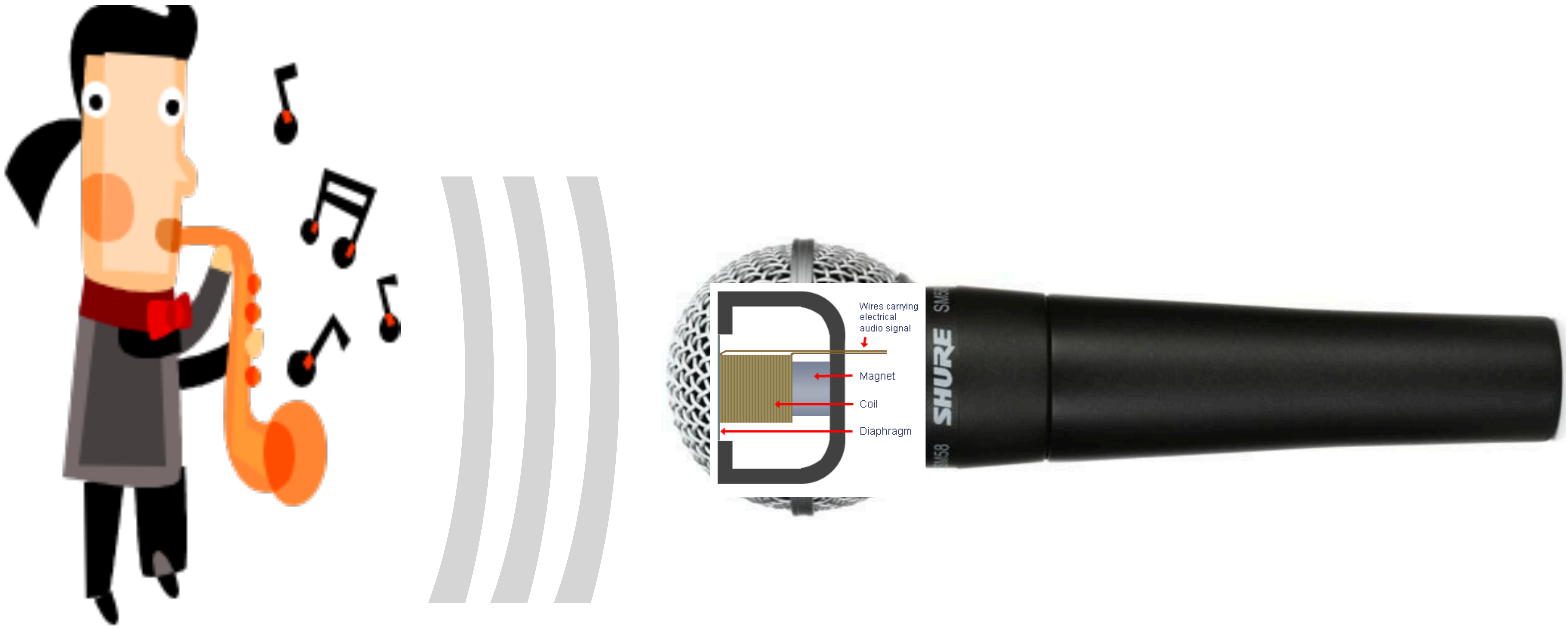
Feature representation

$\vec{x} = \langle a_1, a_2, \ldots, a_n \rangle$

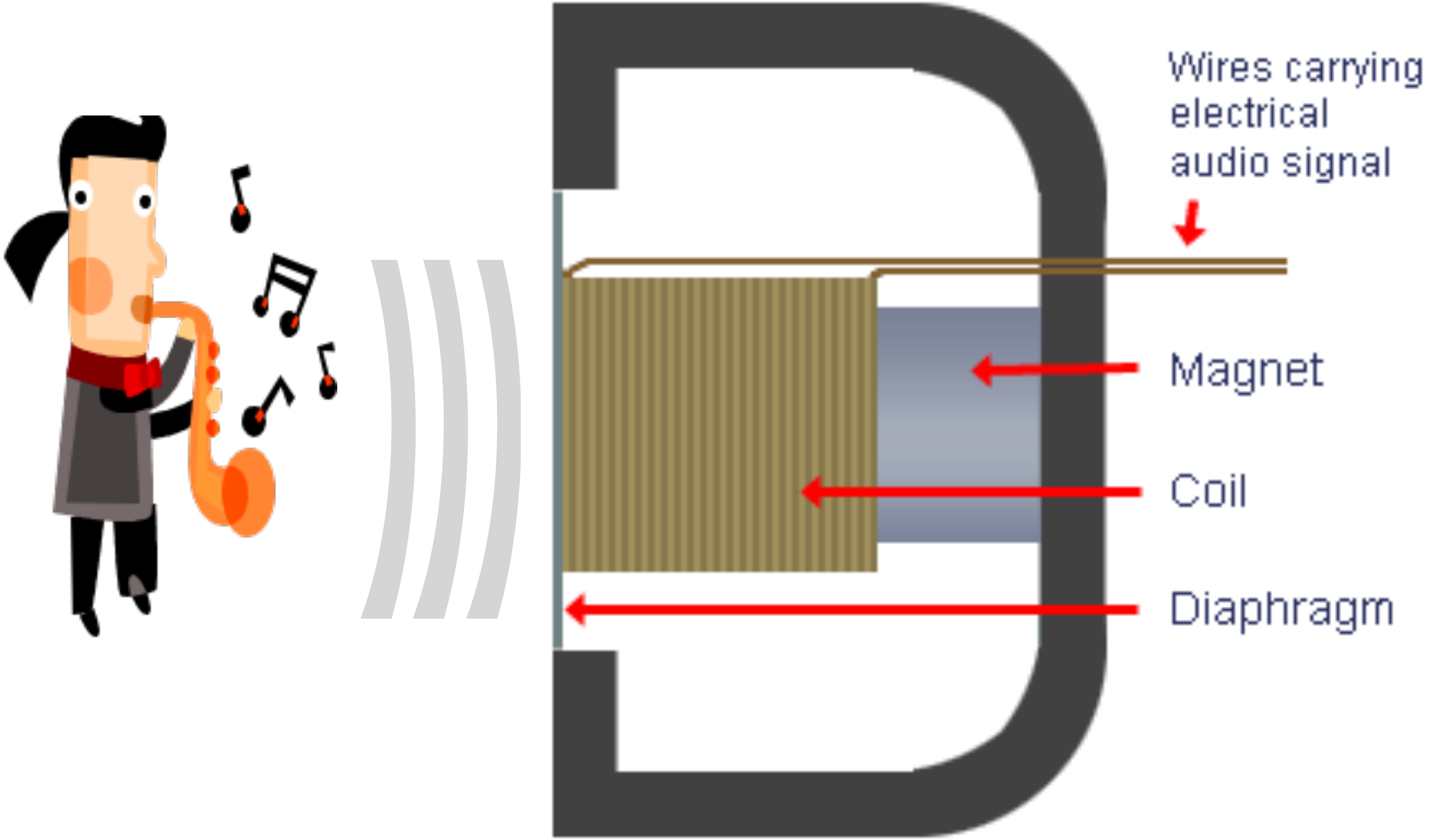


How do we extract meaningful representations from waveforms?

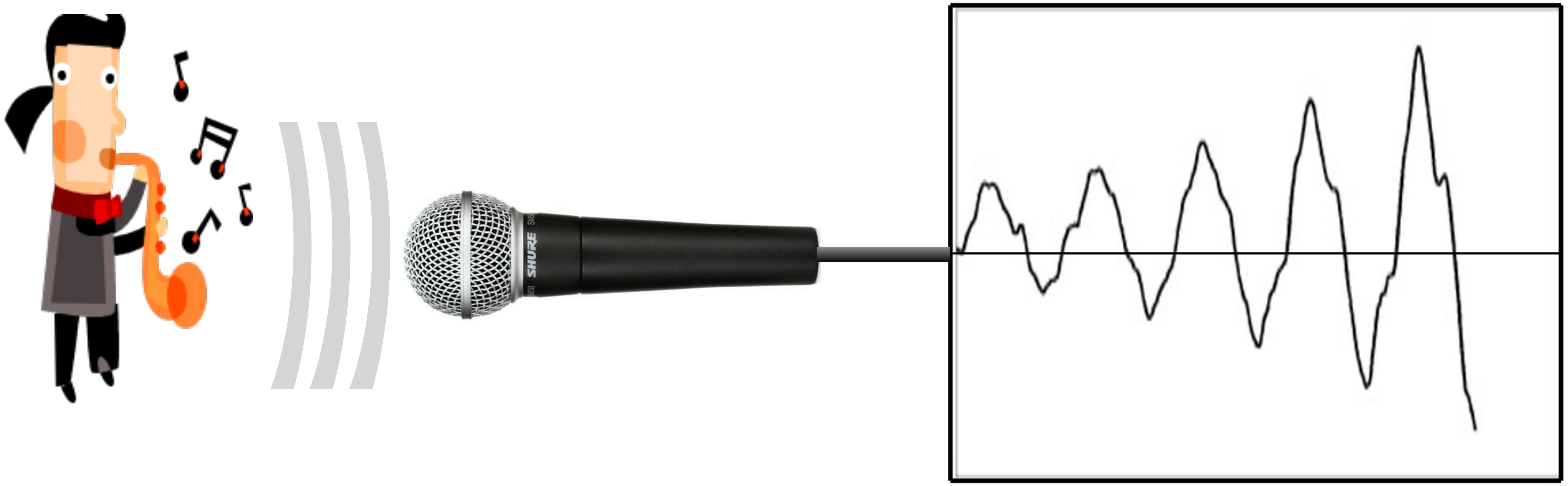
Some audio recording basics


Wires carrying electrical audio signal
Magnet
Coil
Diaphragm
SHURE

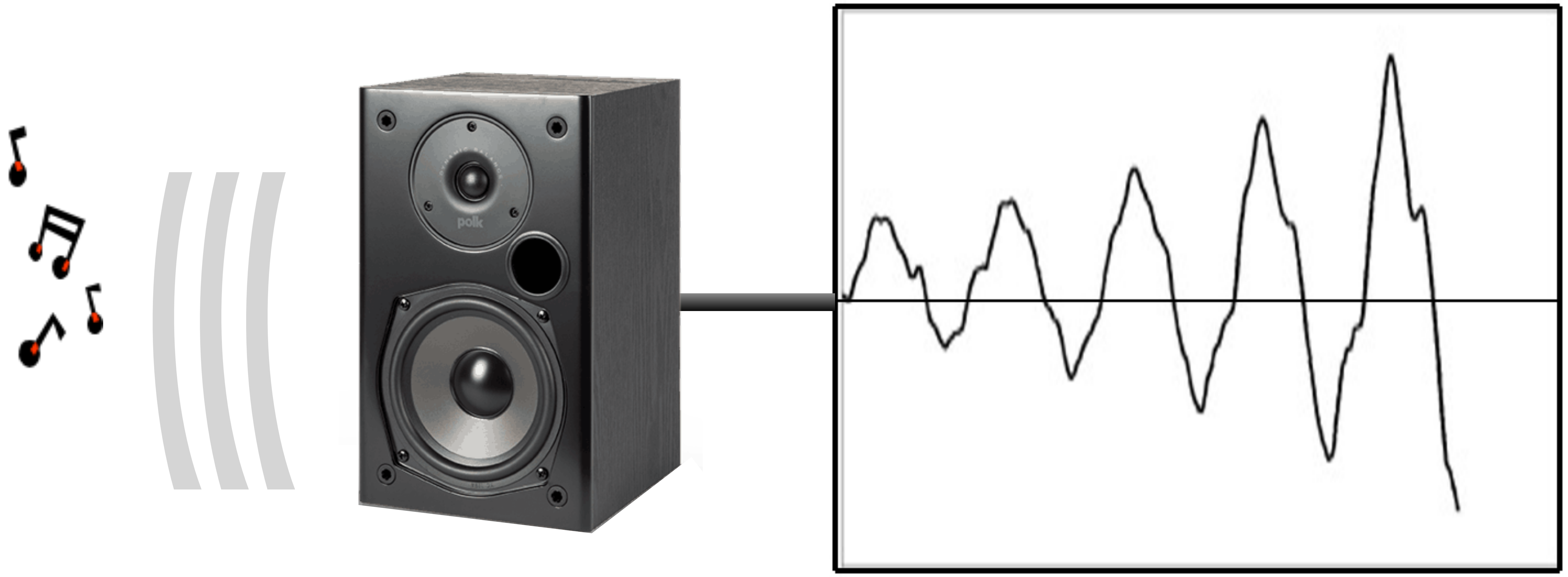
Bryan Pardo


Wires carrying
electrical
audio signal
Magnet
Coil
Diaphragm

Bryan Pardo



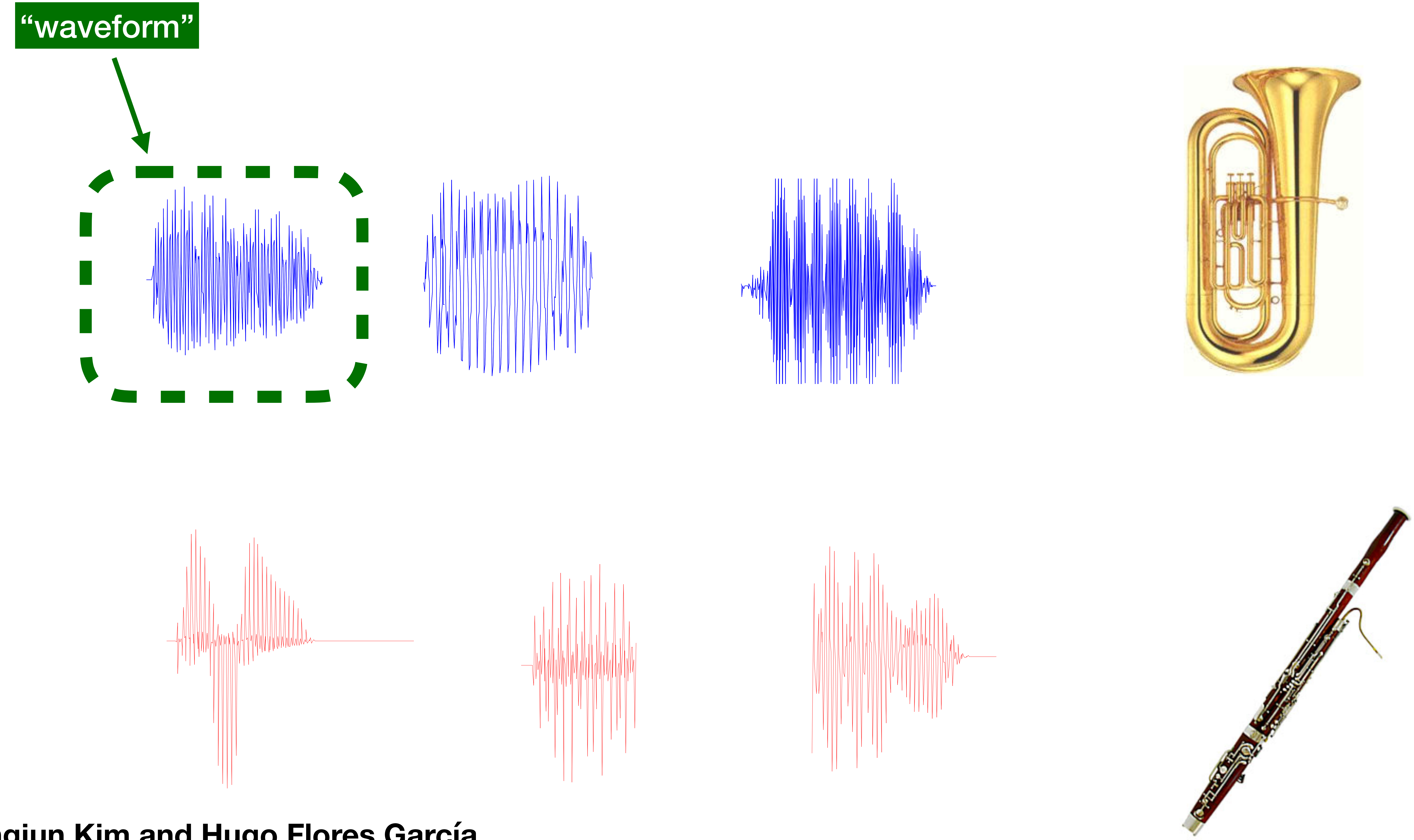
Voltage over time



Voltage over time

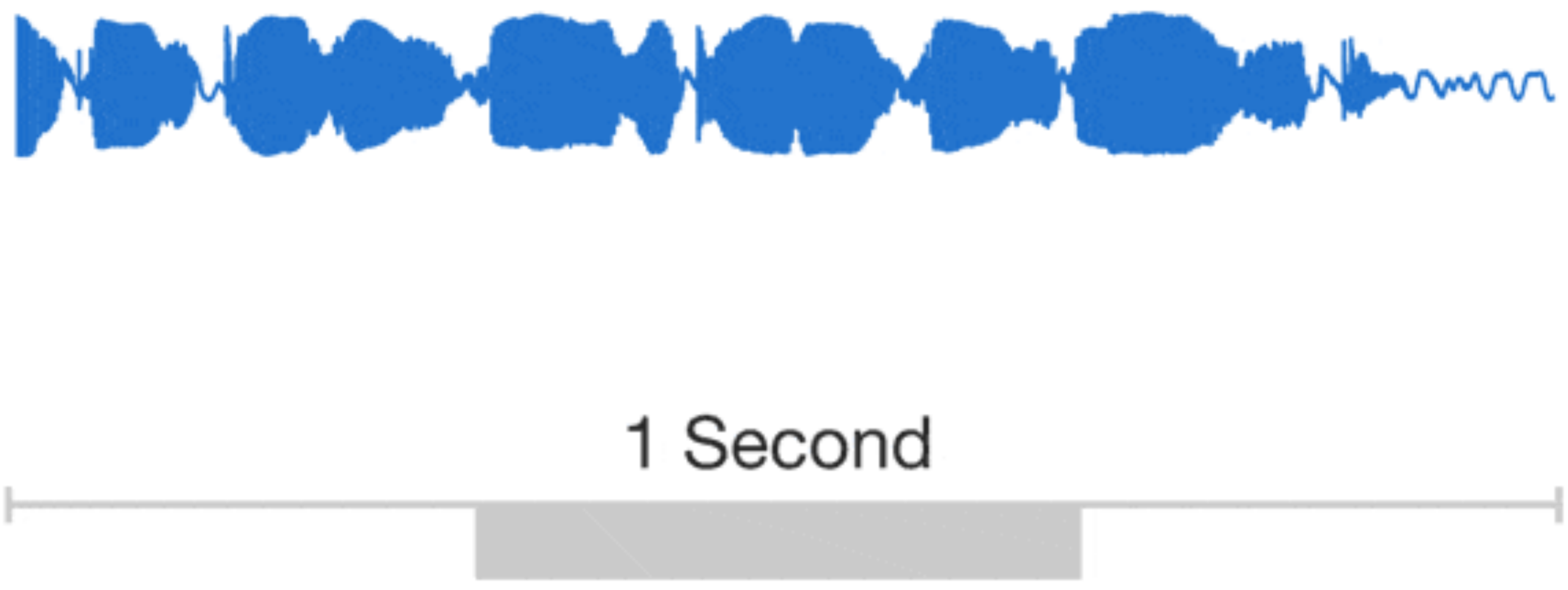
Bryan Pardo

Why not use the waveform as a feature representation?



Bongjun Kim and Hugo Flores García

Why not use the waveform as a feature representation?



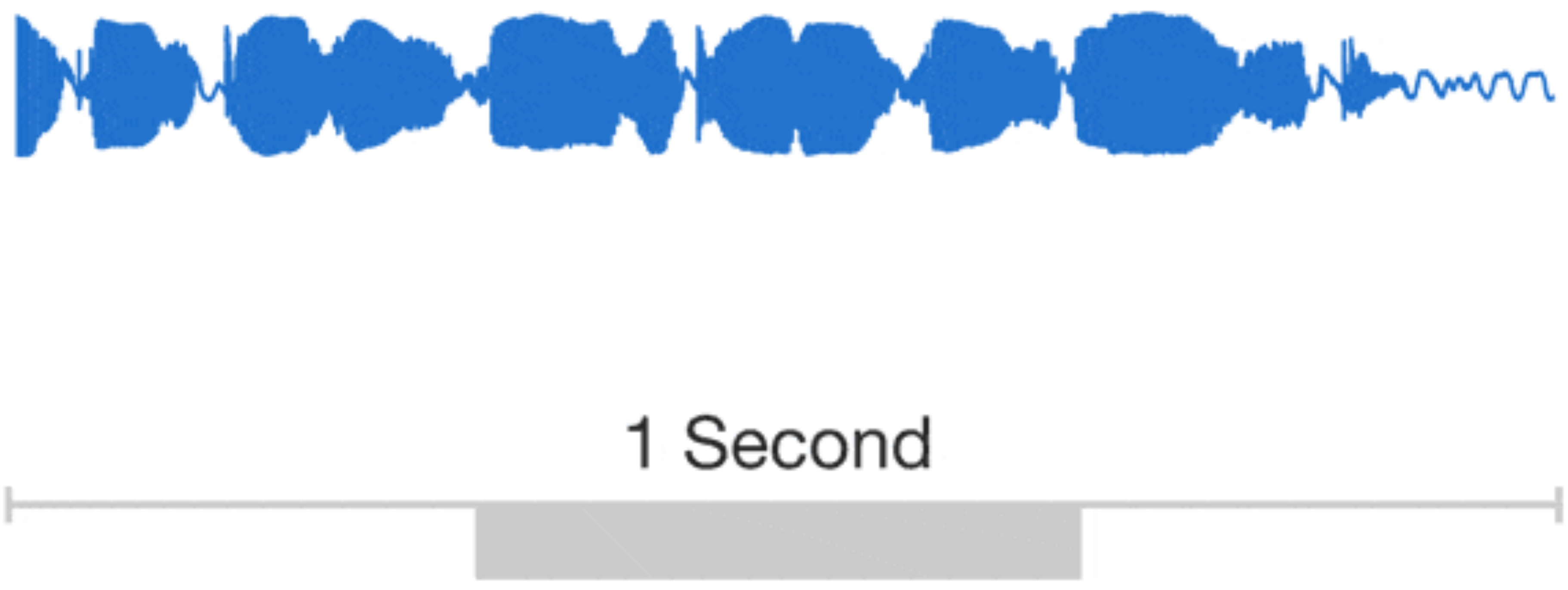
van den Oord et al. 2016

1 second of audio at 44.1kHZ → 44,100 values!

Need a very powerful model (like a deep neural net) which requires millions of training examples.

It's hard to find meaningful patterns!

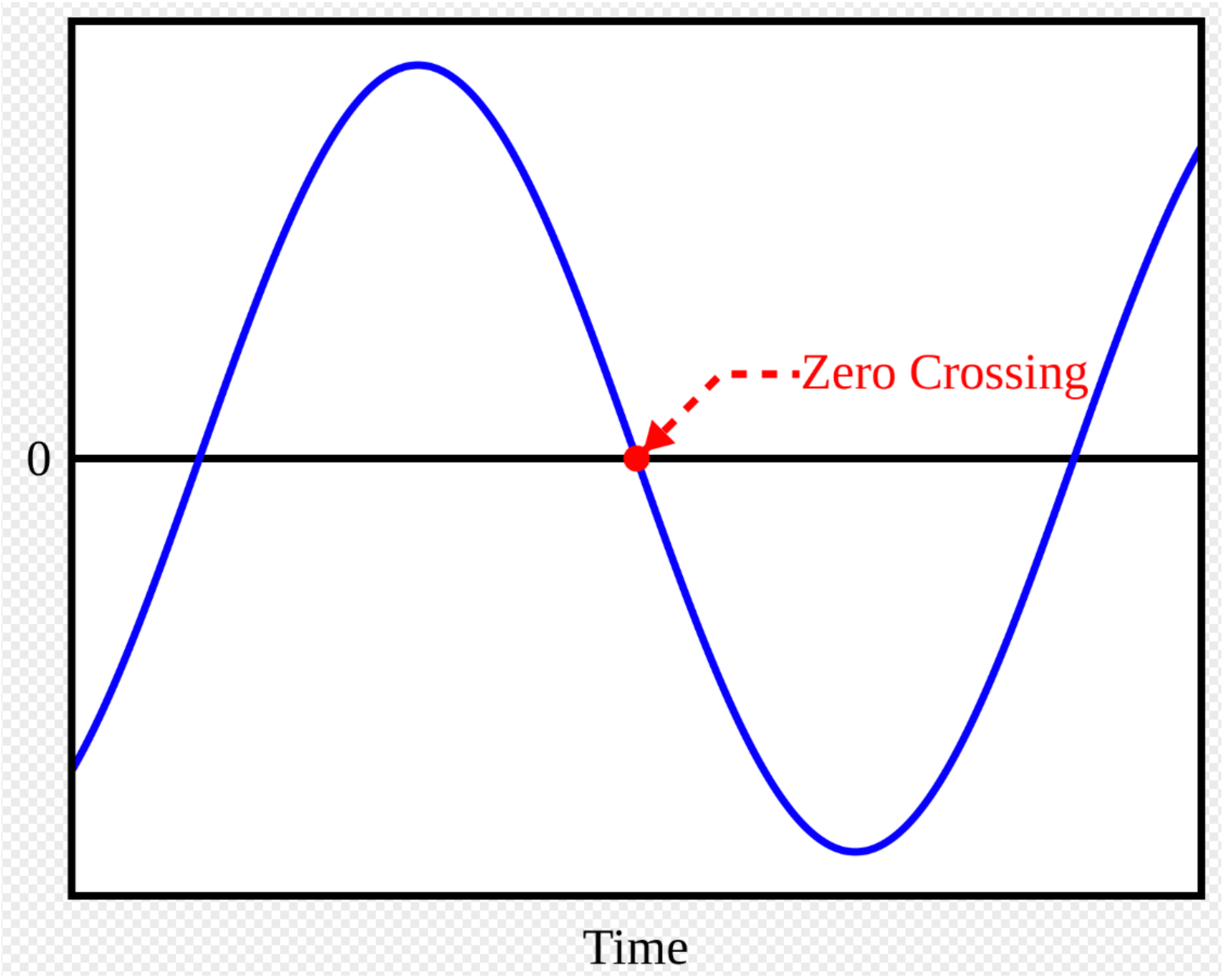
Why not use the waveform as a feature representation?



van den Oord et al. 2016

How do we preprocess the audio waveform to obtain meaningful representations?

Commonly used audio features

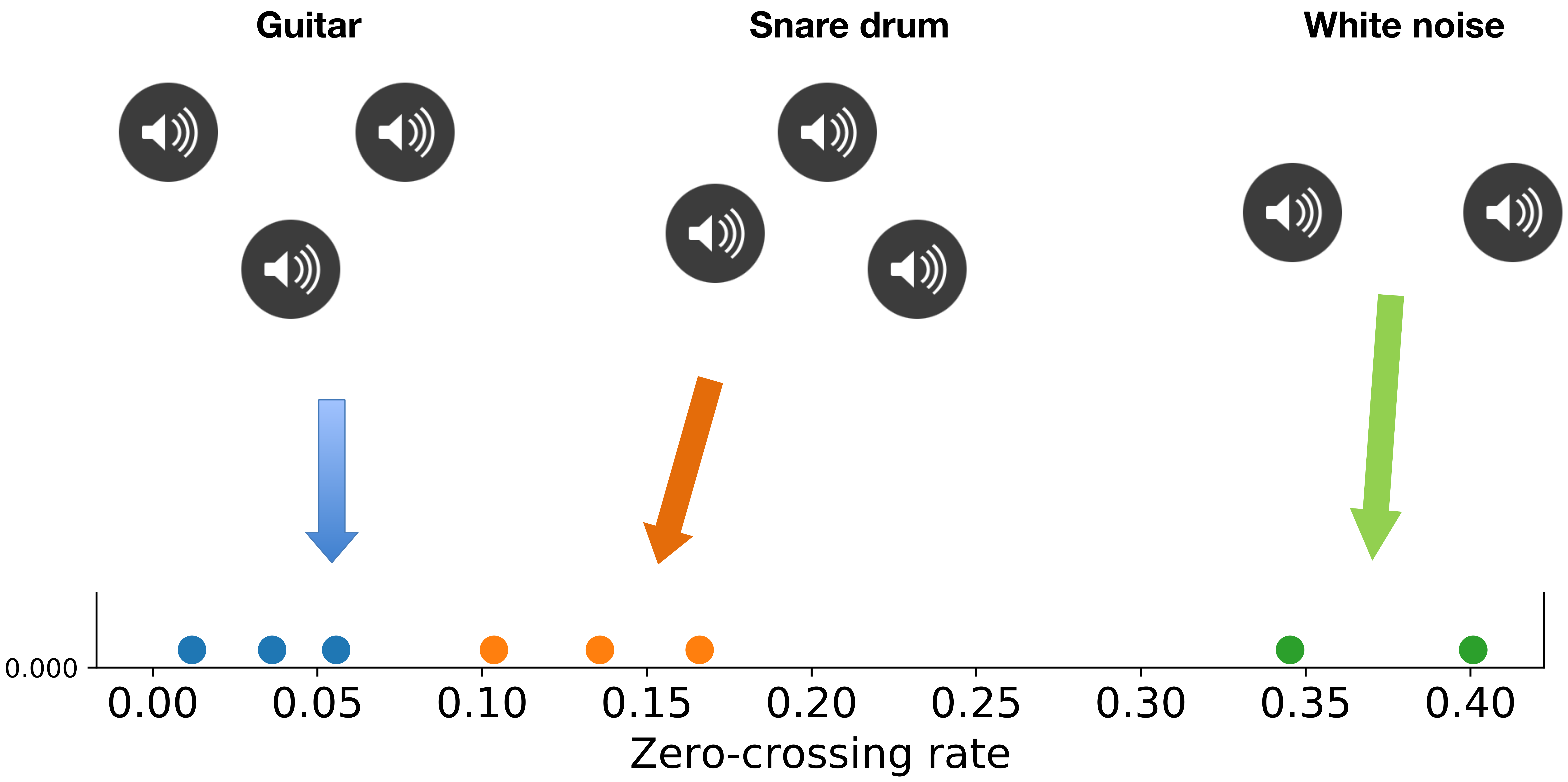


- **Zero-crossing rate**
  - Time-domain feature
  - Rate of sign changes in a signal
  - Low for harmonic sounds, high for noisy sounds

* Figure: https://en.wikipedia.org/wiki/Zero_crossing

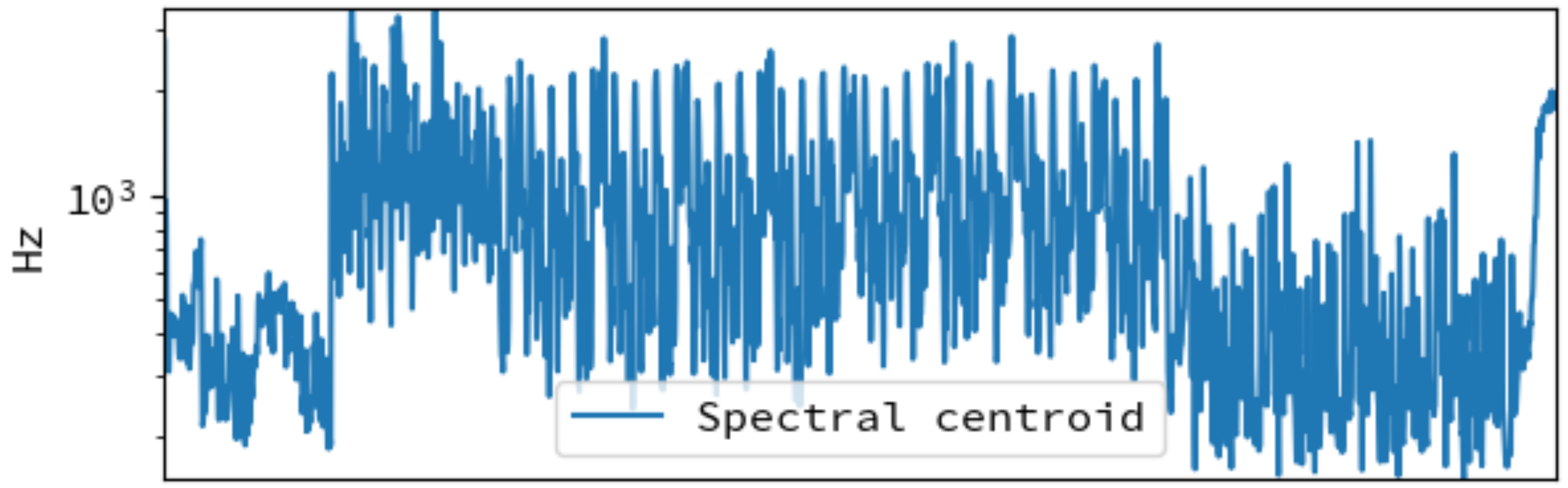
Commonly used audio features

- Zero-crossing rate



Commonly used audio features

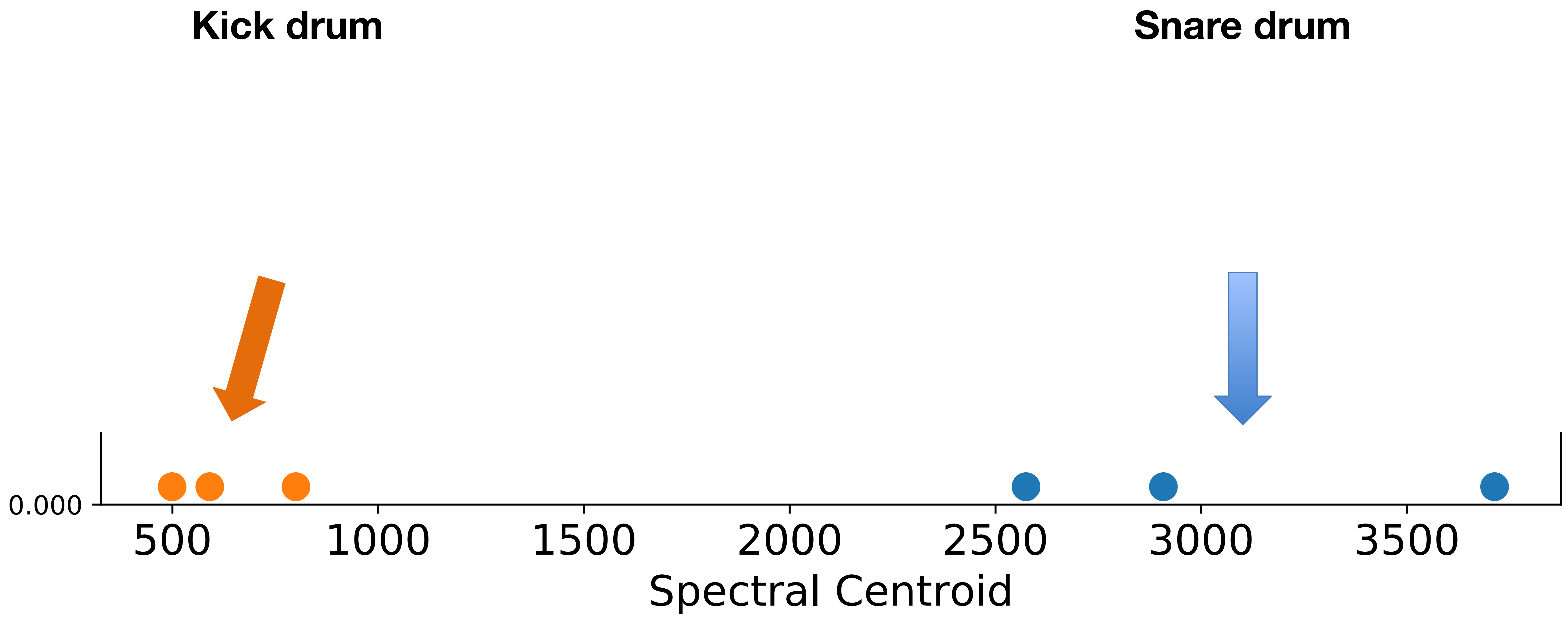
- Spectral centroid
  - Frequency domain feature
  - The weighted mean of the frequencies in the signal
  - Known as a predictor of the “brightness” of a sound



*** figure: https://librosa.github.io/librosa/generated/librosa.feature.spectral_centroid.html**

Commonly used audio features

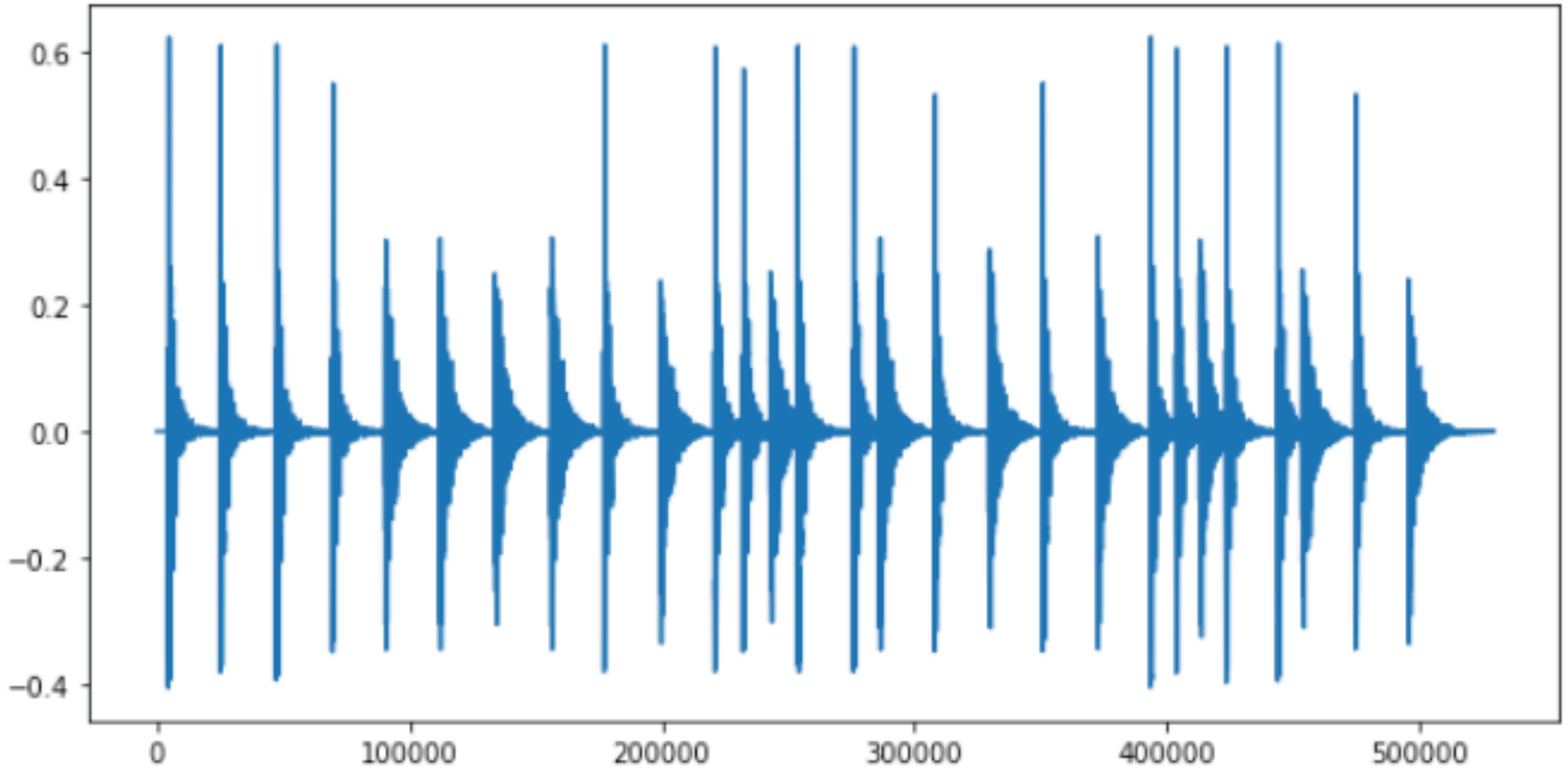
- Spectral centroid



Example: Drum Transcription

Automatic drum transcription

- Let's build a drum transcription machine only using spectral centroid features

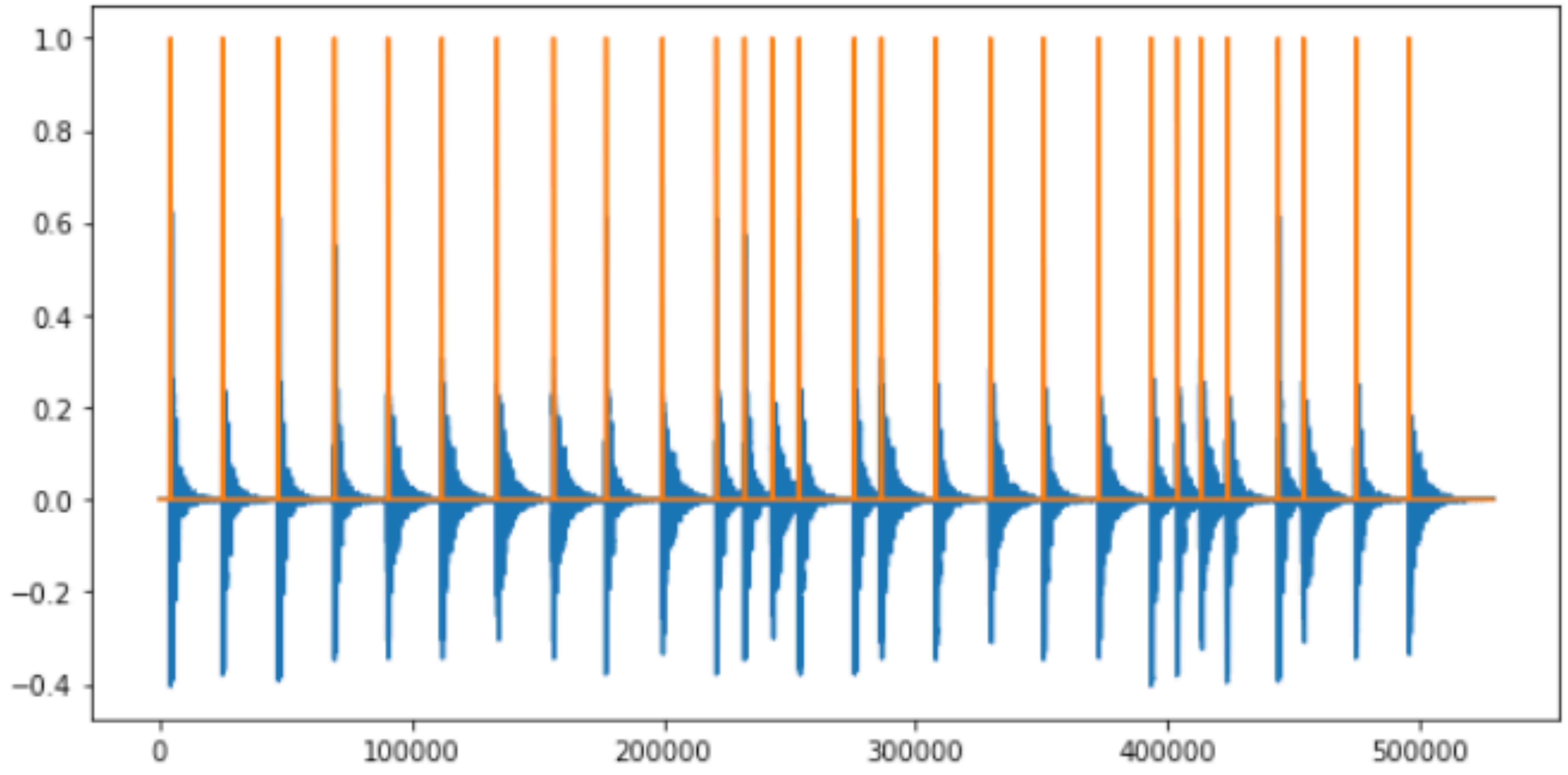


Bongjun Kim (Winter 2019)

Automatic drum transcription

Onset detection:

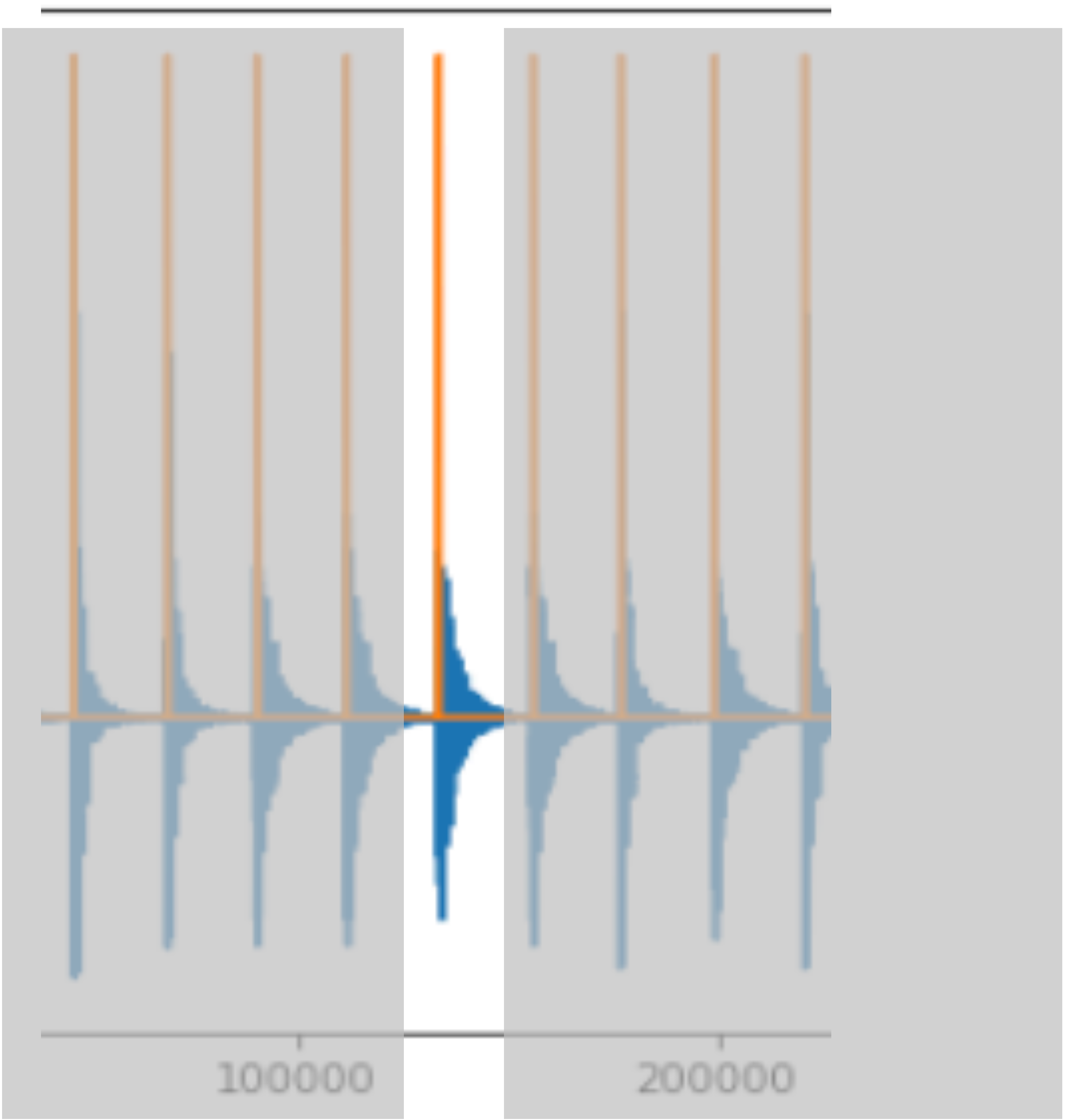
`librosa.onset.onset_detect`



Bongjun Kim (Winter 2019)

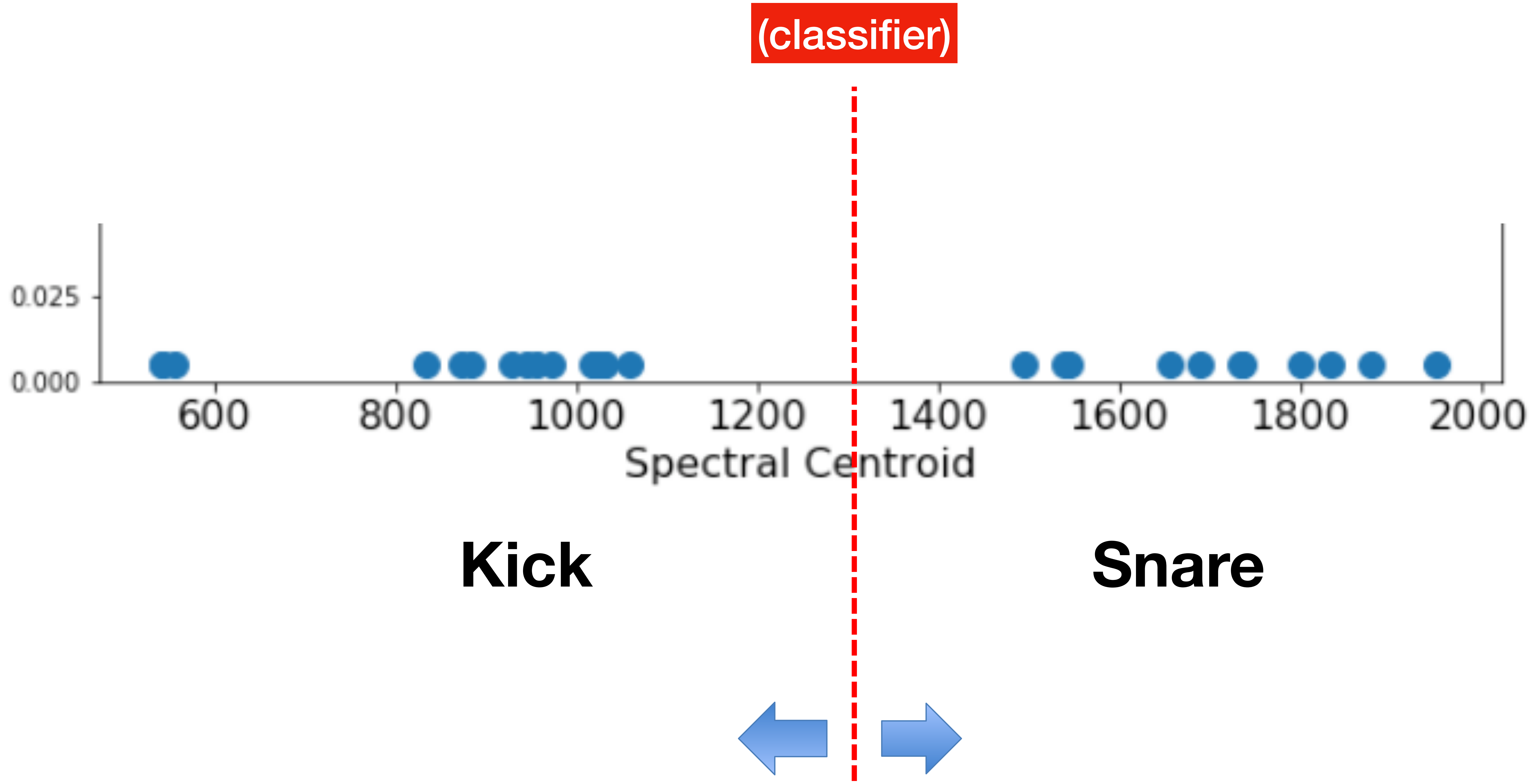
Automatic drum transcription

- Segmentation
  - Cutting the recording every *<onset−2048 samples>*



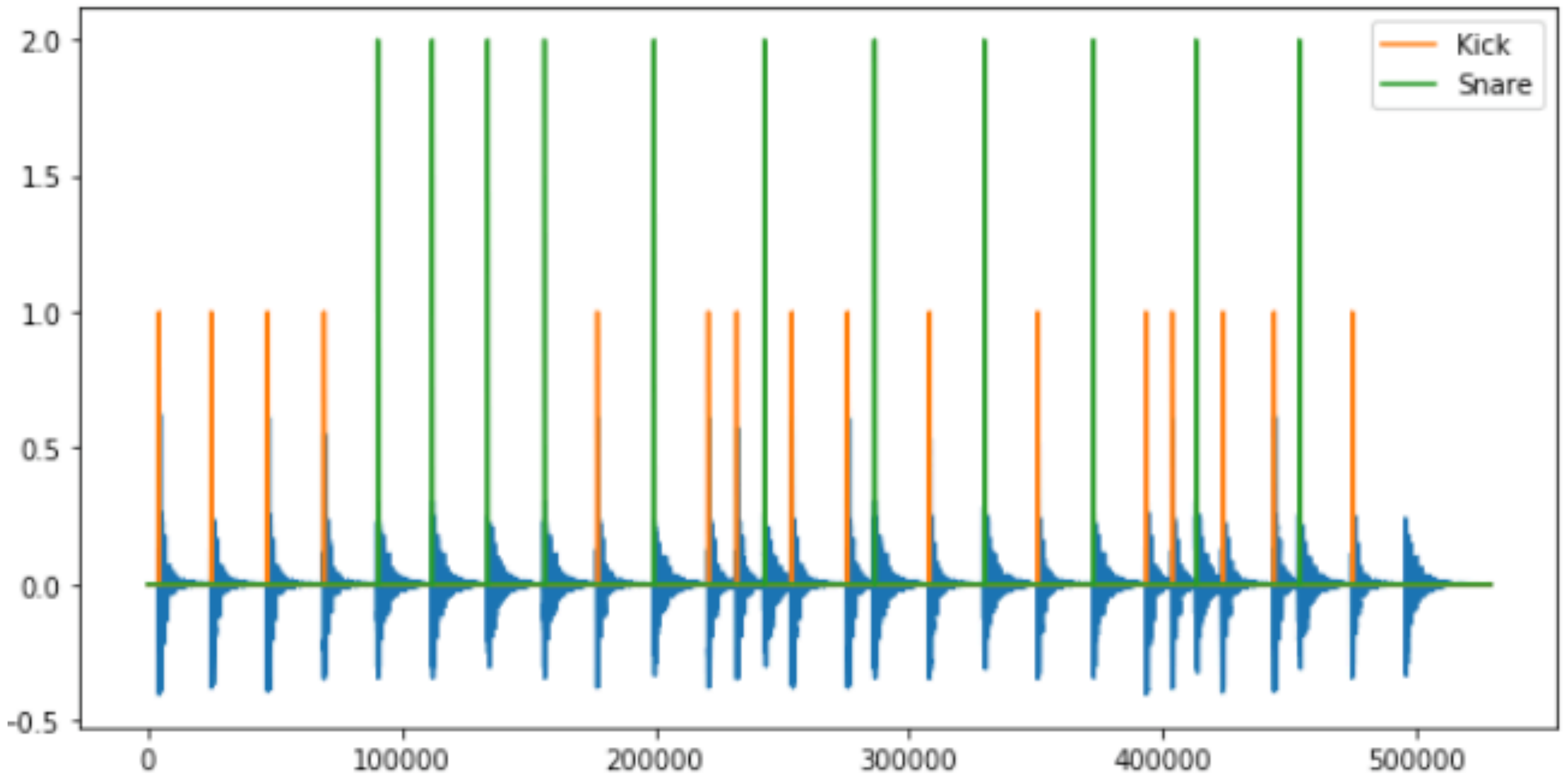
Bongjun Kim (Winter 2019)

Automatic drum transcription



- Extracting spectral centroid from each segment

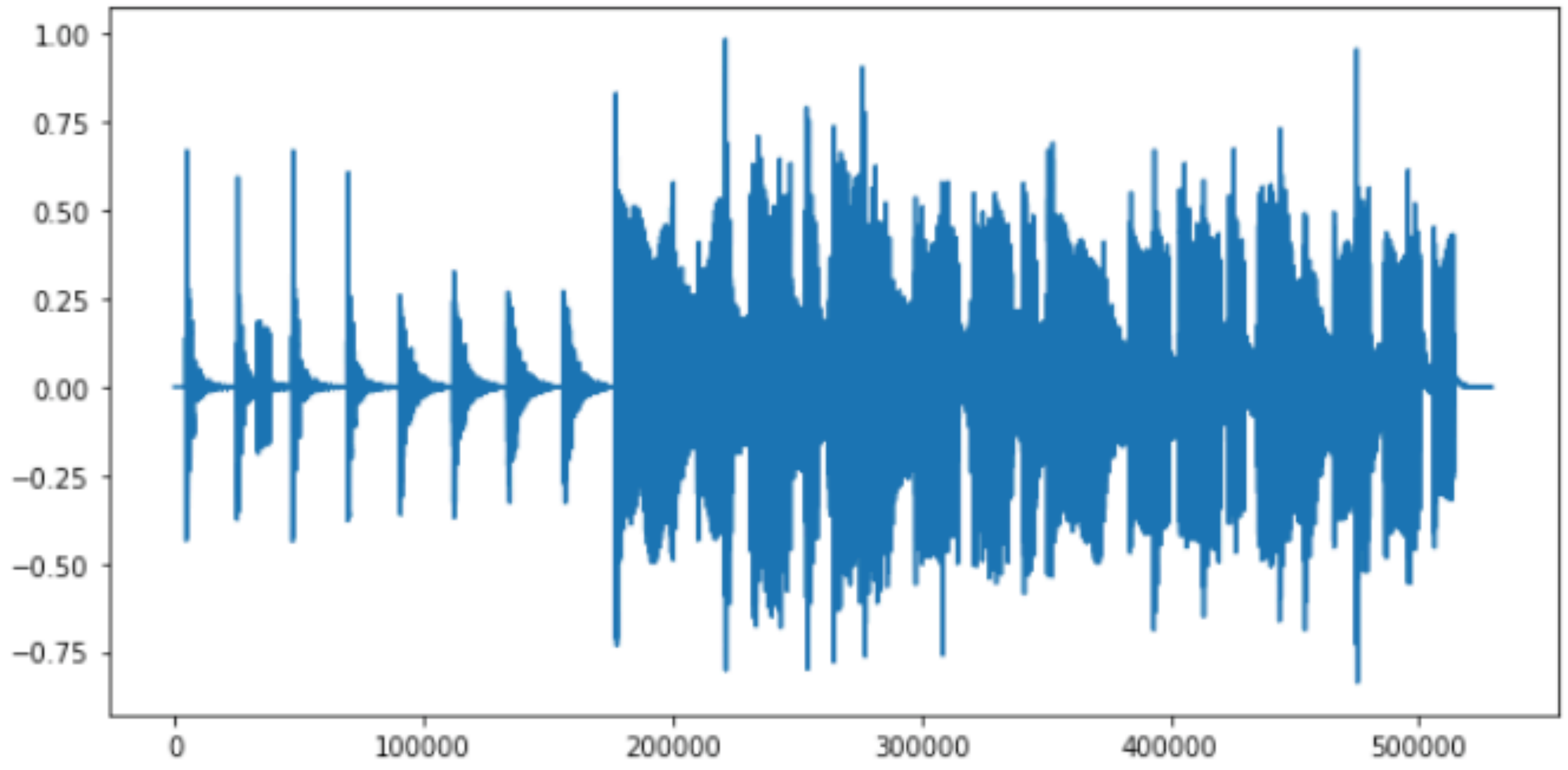
Automatic drum transcription



Bongjun Kim (Winter 2019)

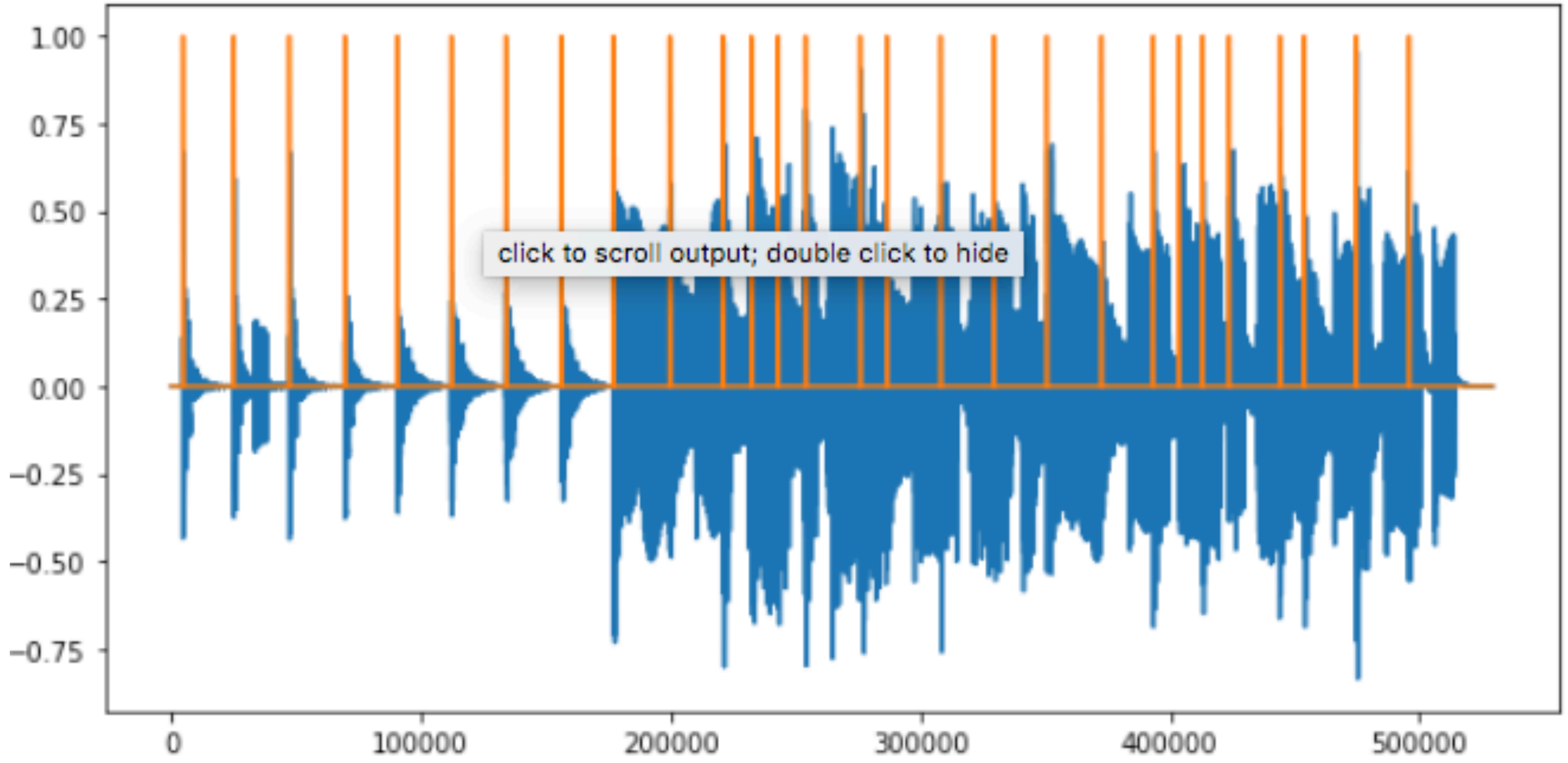
Automatic drum transcription-2

- More challenging example



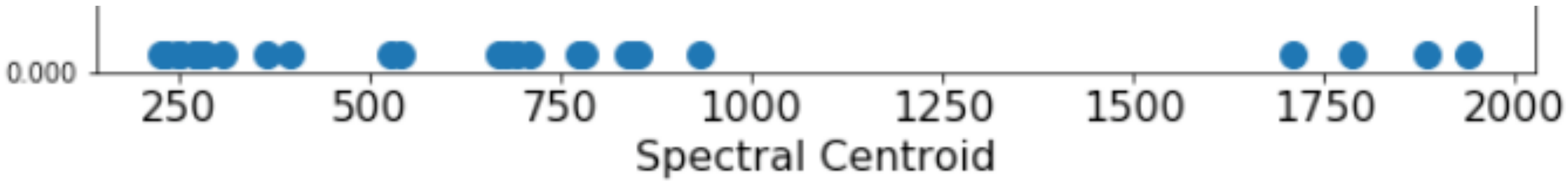
Automatic drum transcription-2

- Onset detection might not work that well on this example, but let's assume we have perfect onset info

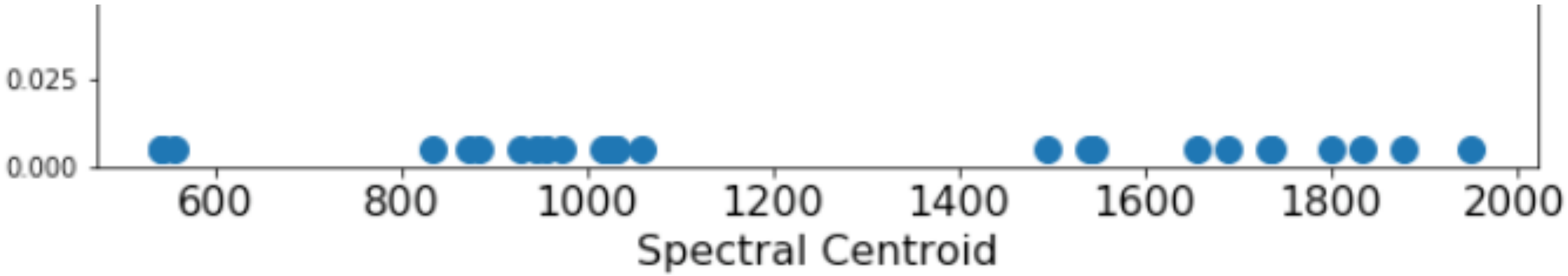


Automatic drum transcription-2

- Segmentation and feature extraction

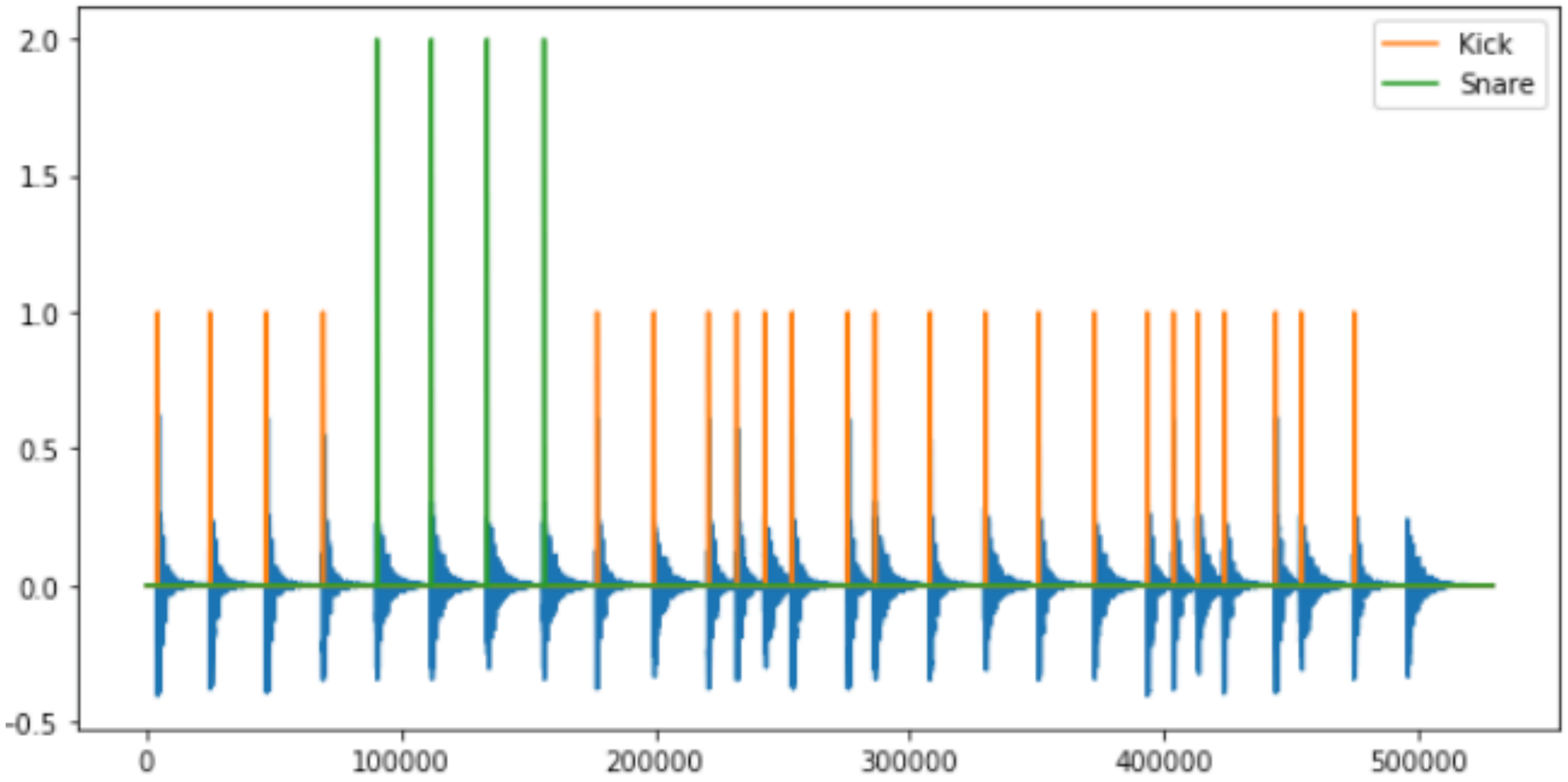


- The previous example



Automatic drum transcription-2

- More challenging example

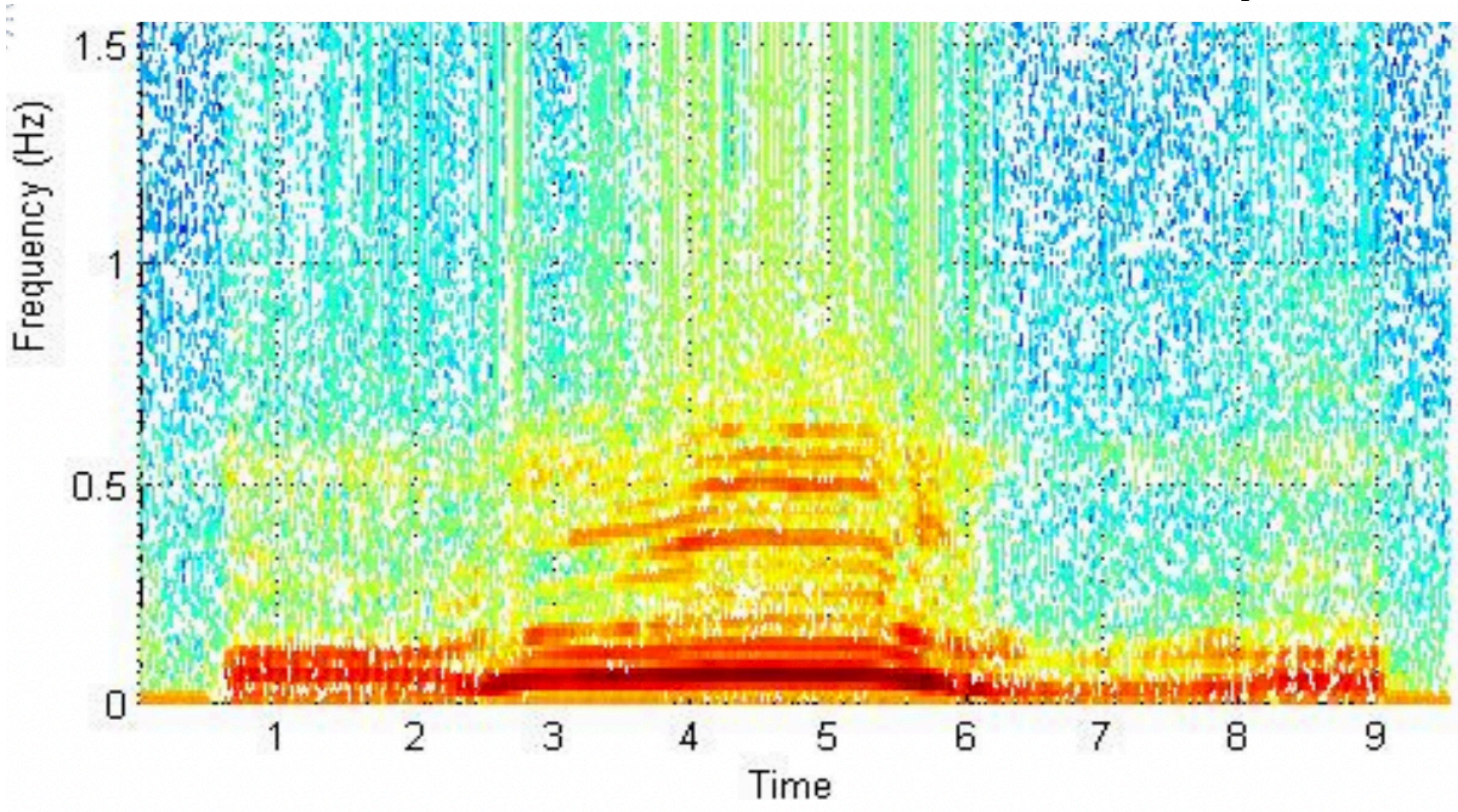


You can find more feature extraction functions in the Librosa package

Commonly used audio features

Bryan Pardo

- Spectrogram
  - Plots the magnitude of the frequency spectrum as a function of time.



Lo-res image (Usually 256x199 for 1 second of audio)

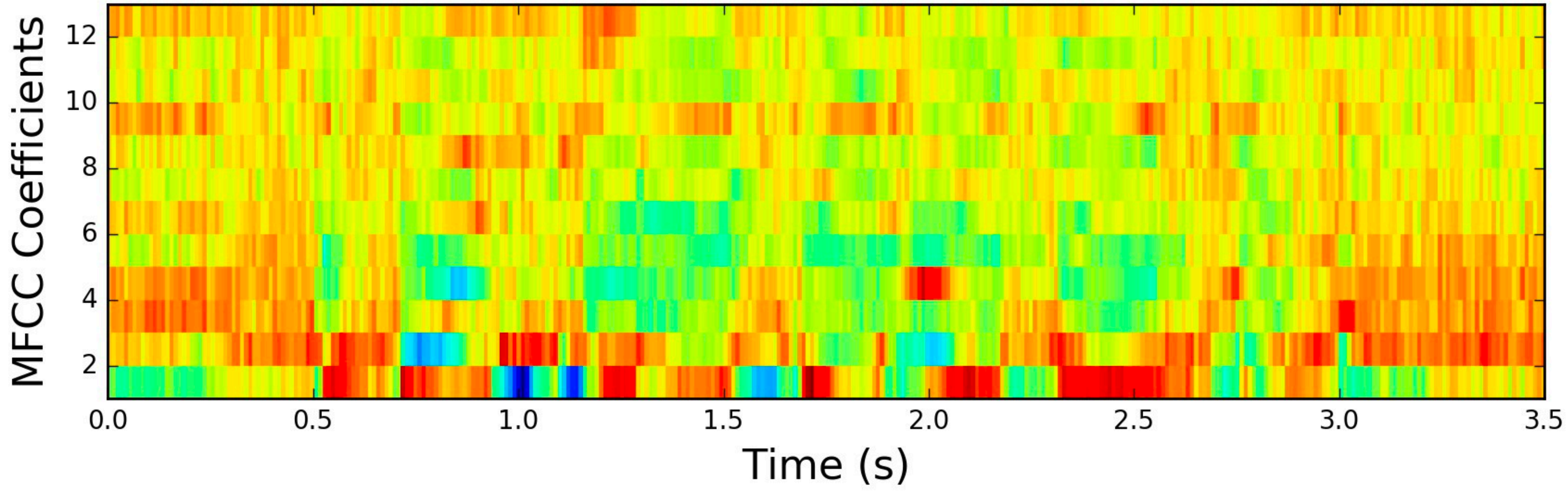
Lower dimensionality than a pure waveform, but it is still high dimensional!

Bongjun Kim

Commonly used audio features

- Mel Frequency Cepstral Coefficients (MFCCs)

~10 times smaller than a spectrogram!



Plots the envelope of the spectrum with just a few coefficients (usually 13)

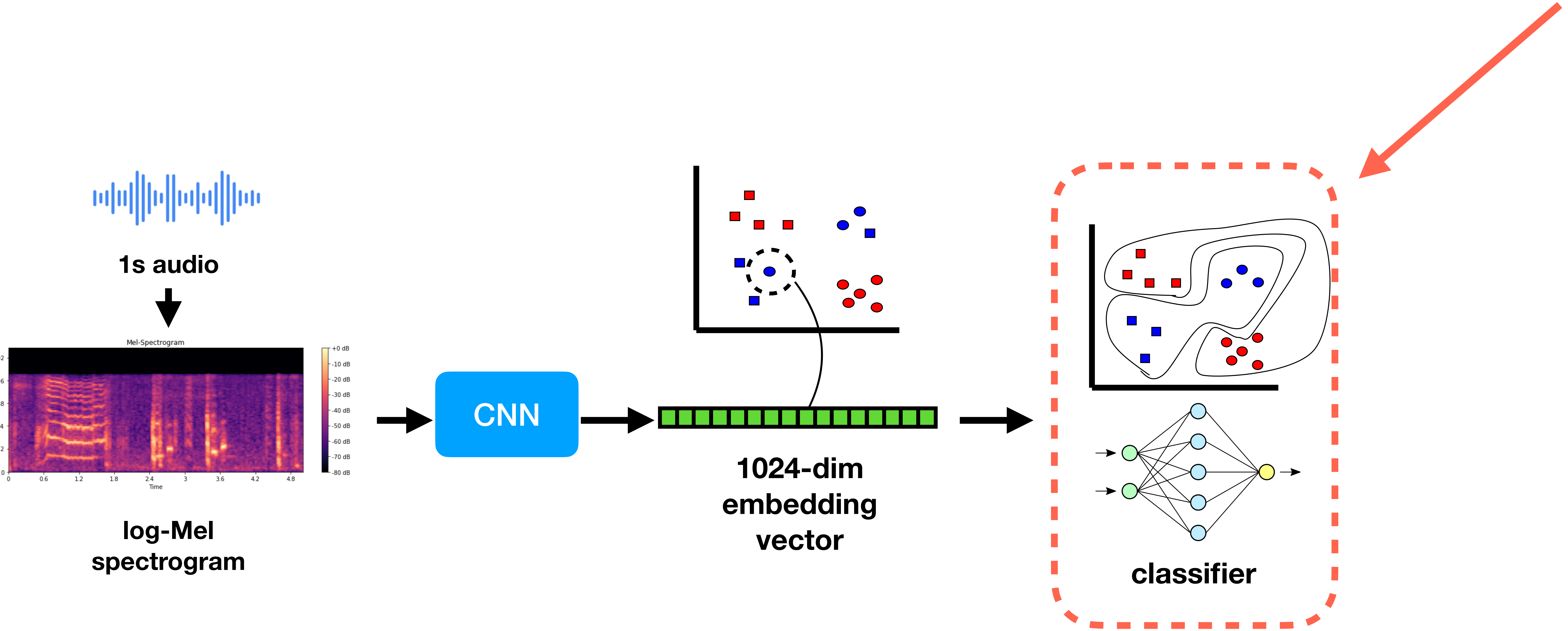
The standard for speech recognition before deep learning!

Deep Embeddings

- Can we use a neural net to generate meaningful features?

Deep Embeddings: Transfer Learning

Train on a "pretext" task to learn a meaningful internal representation!
(aka *transfer learning*)

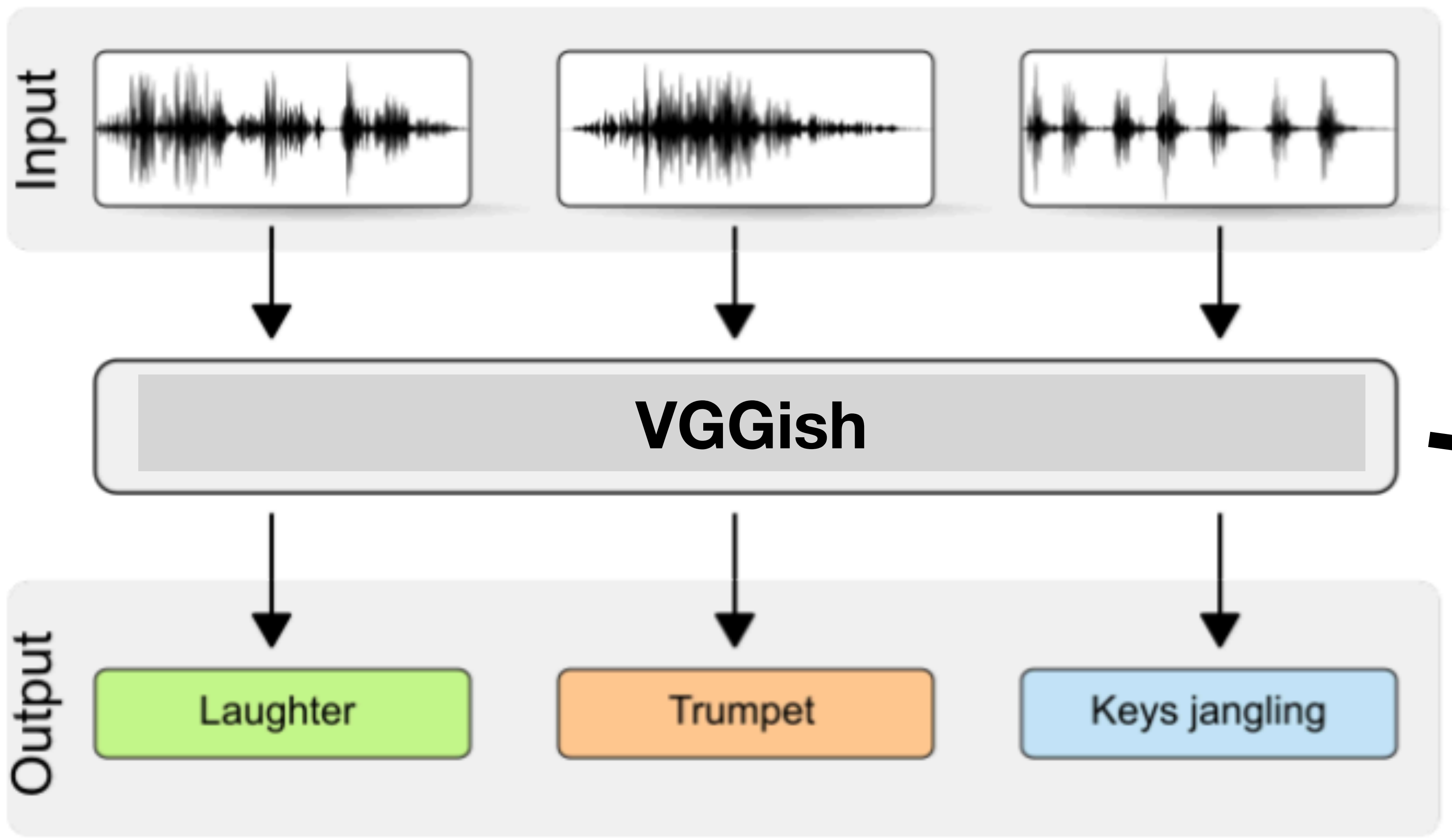


Deep Embeddings: VGGish

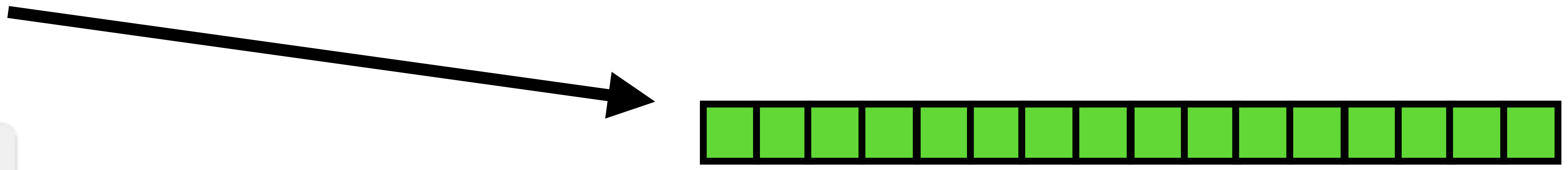
(Simoyan et al. 2015)

The original “deep audio embedding”

Trained on an Audio Tagging task on Audioset (subset of YouTube)



https://medium.com/@yxu71/freesound-tagging-by-vggish-with-knn-731dc3e1dc5a



128-dim embedding vector

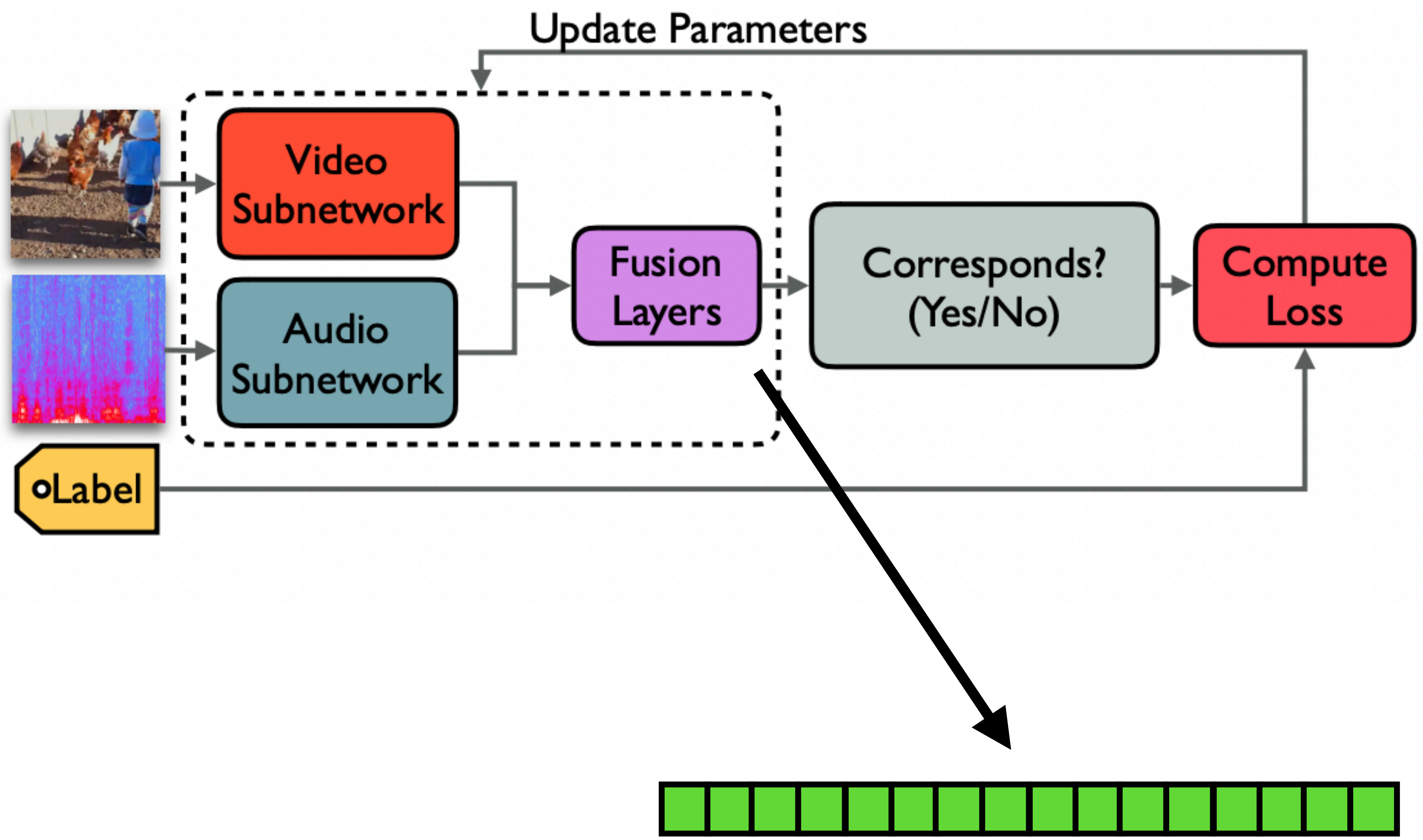
Deep Embeddings: OpenL3

R. Arandjelovic et al., 2017
and
Cramer et al., 2019

- L³ -Net (aka OpenL3)

Predict whether an audio clip and an image correspond to each other (audiovisual correspondence)

Train on LOTS of data (all of YouTube if you want!)



512-dim
or
6144-dim
embedding vector

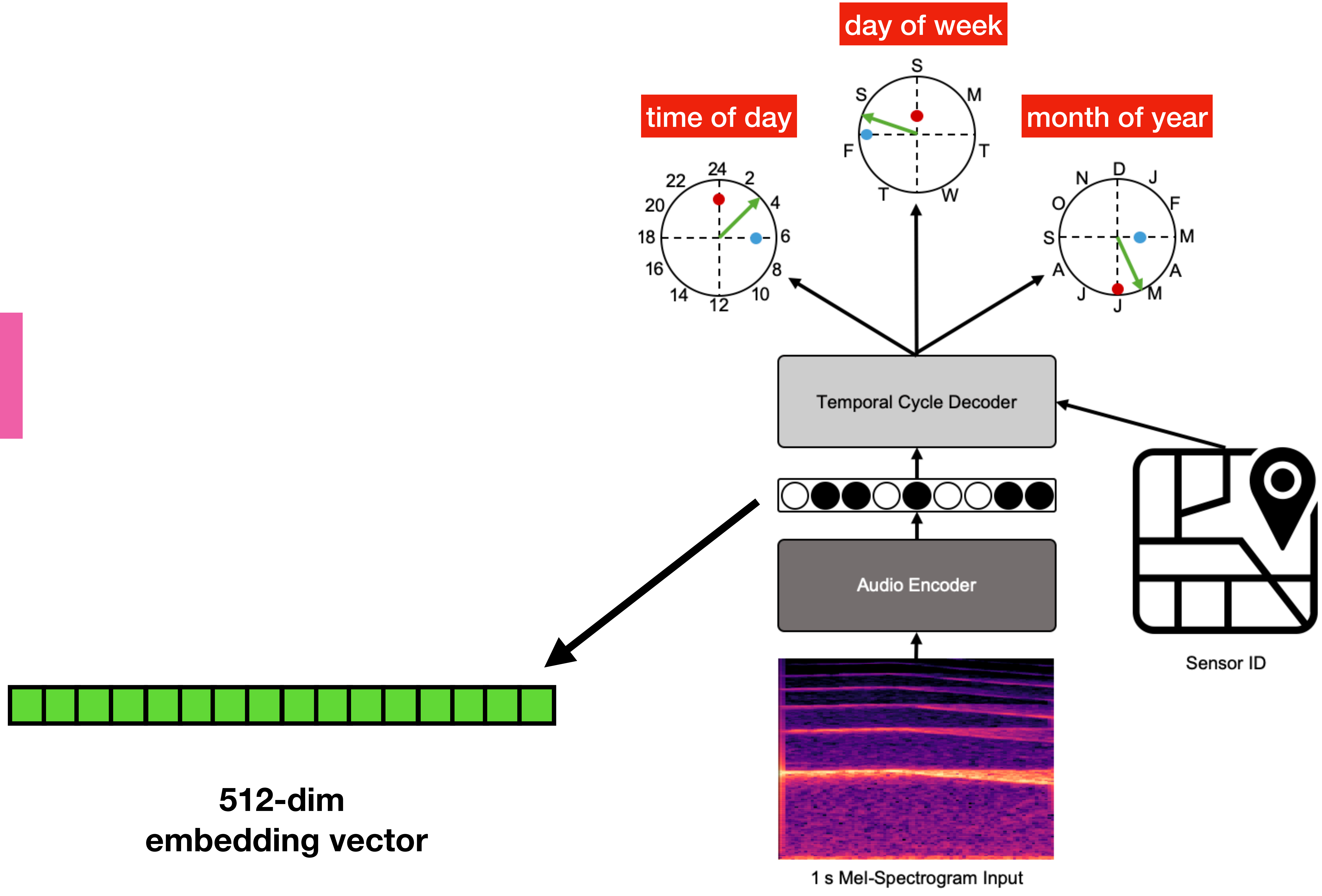
pretext task — only for learning a meaningful representation

No labels needed!
(Self-supervised)

Deep Embeddings: TriCycle (Cartwright et al. 2019)

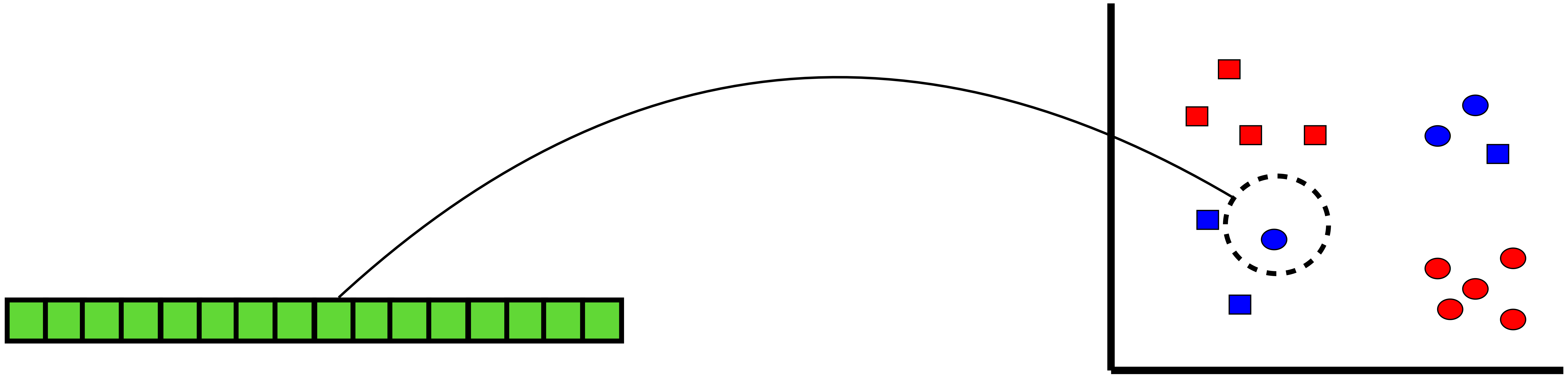
Given an audio clip, predict temporal cycles!

self-supervised:
all you need are the timestamps!



Dimensionality Reduction

Aka how can we visualize high-dimensionality embedding spaces?



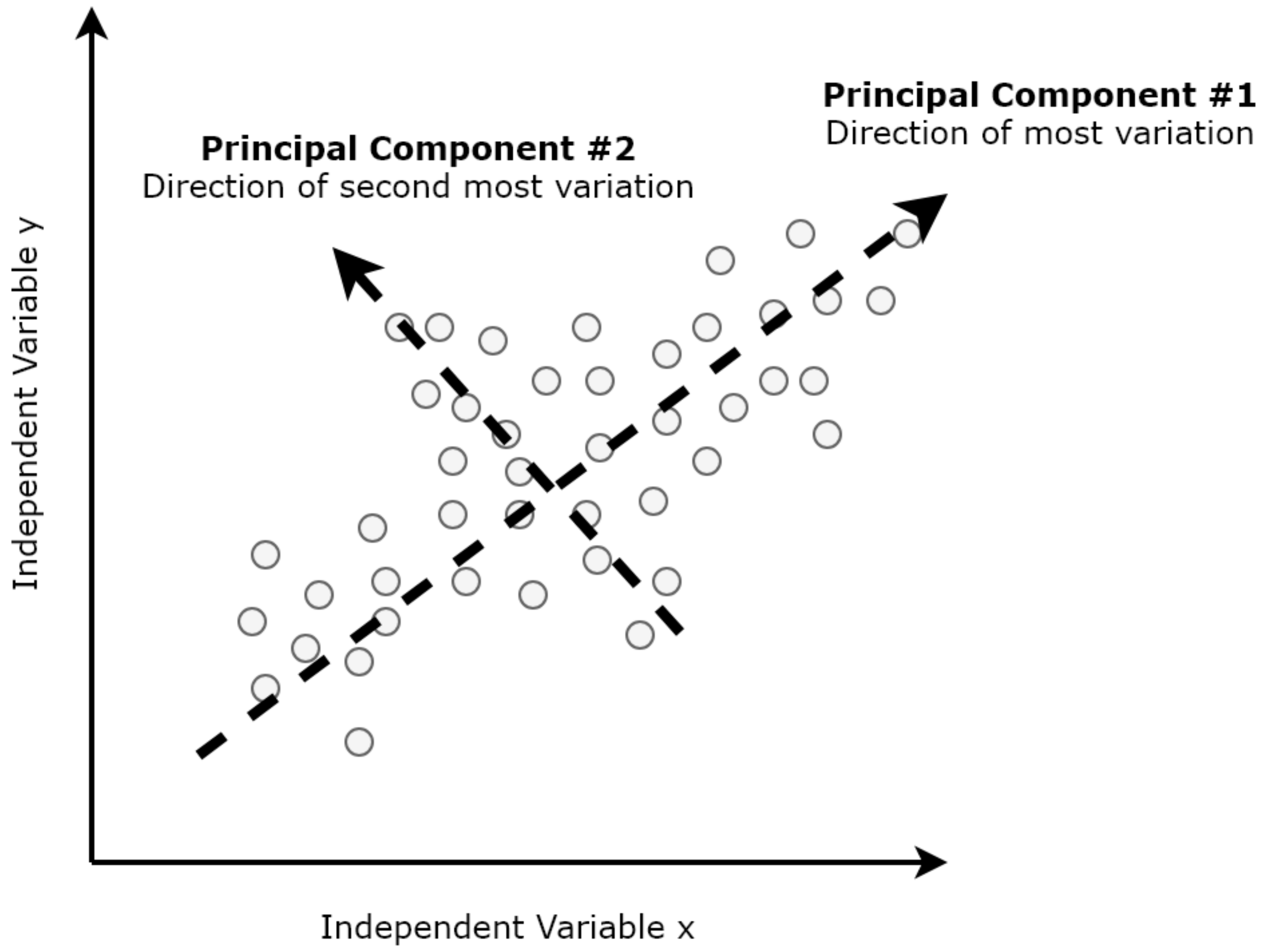
1024-dim embedding vector

2-dim projection of embedding space

PCA (Principal Component Analysis)

Goal: find a **linear** projection of your dataset, such that you can keep the axes with the **most** variation

Dimensions with most variation
===
“Principal components”



https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29

t-SNE
(T-distributed Stochastic Neighbor Embedding)

Goal: find a **nonlinear** projection of your dataset, such that the **local relationships** between points are preserved.

Iterative algorithm (slow)!

https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

Musical Instrument ID (MIID)

The Dataset

Philharmonia Dataset

- **14,000 sound samples of the Philharmonia Orchestra**
- **Mostly single notes of isolated instruments, 1-5s in length**
- **19 melodic instruments + many percussion instruments**

https://philharmonia.co.uk/resources/sound-samples/

Some Links I Shared

Google's infinite drum machine:
https://experiments.withgoogle.com/ai/drum-machine/view/

VQGAN + CLIP:
https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN#scrollTo=ix4T6qkRqZgi

huggingface spaces
https://huggingface.co/spaces