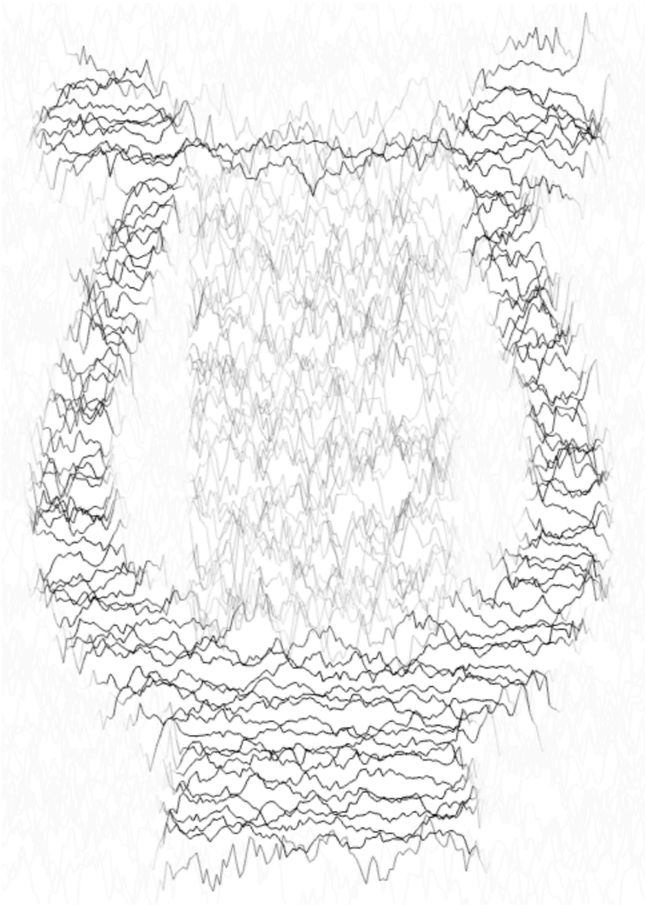
University of Illinois @ Urbana-Champaign



CS 498PS – Audio Computing Lab

Audio Mixtures and Spectral Factorizations

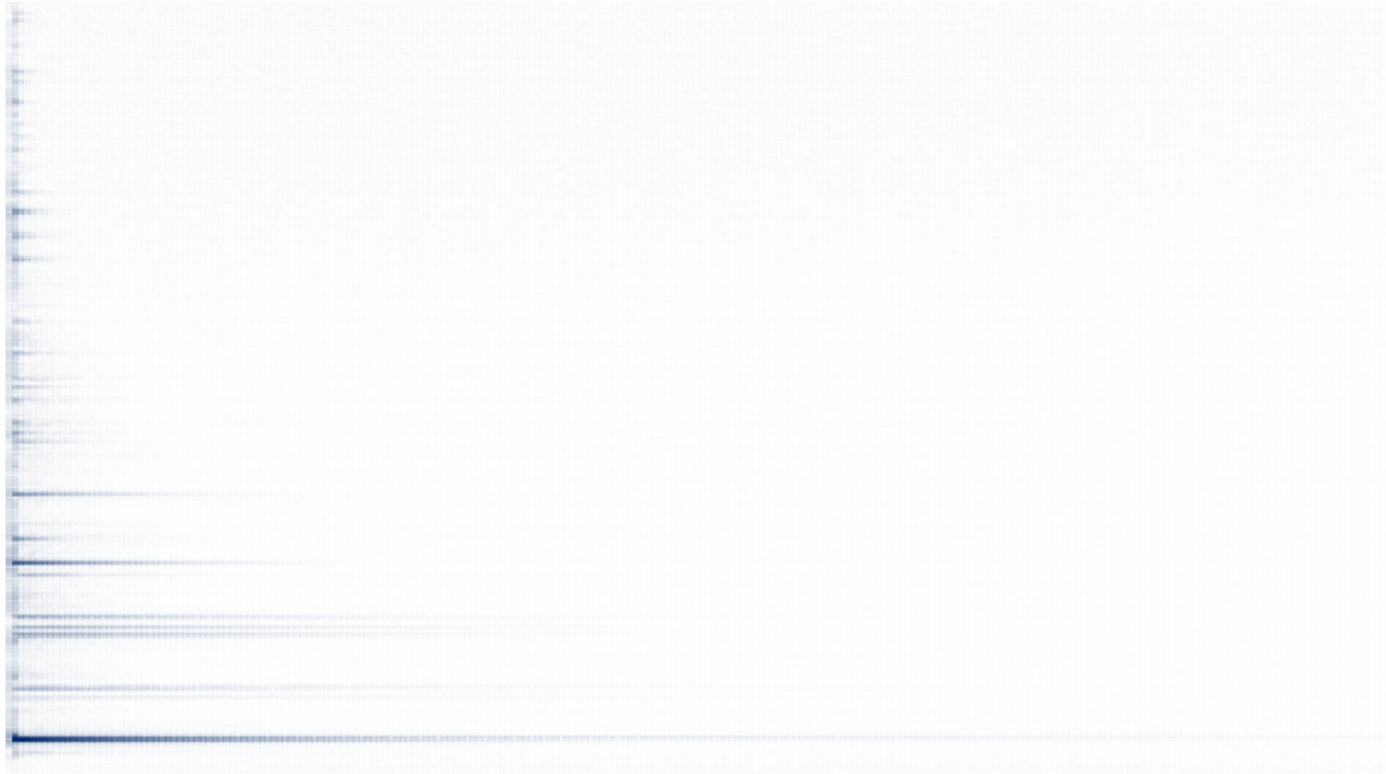
Paris Smaragdis
paris@illinois.edu
paris.cs.illinois.edu

Overview

- Factorizing spectra
- Audio mixture problems
  - Scene analysis
  - Source separation
  - Denoising
  - Multisource manipulation
  - …

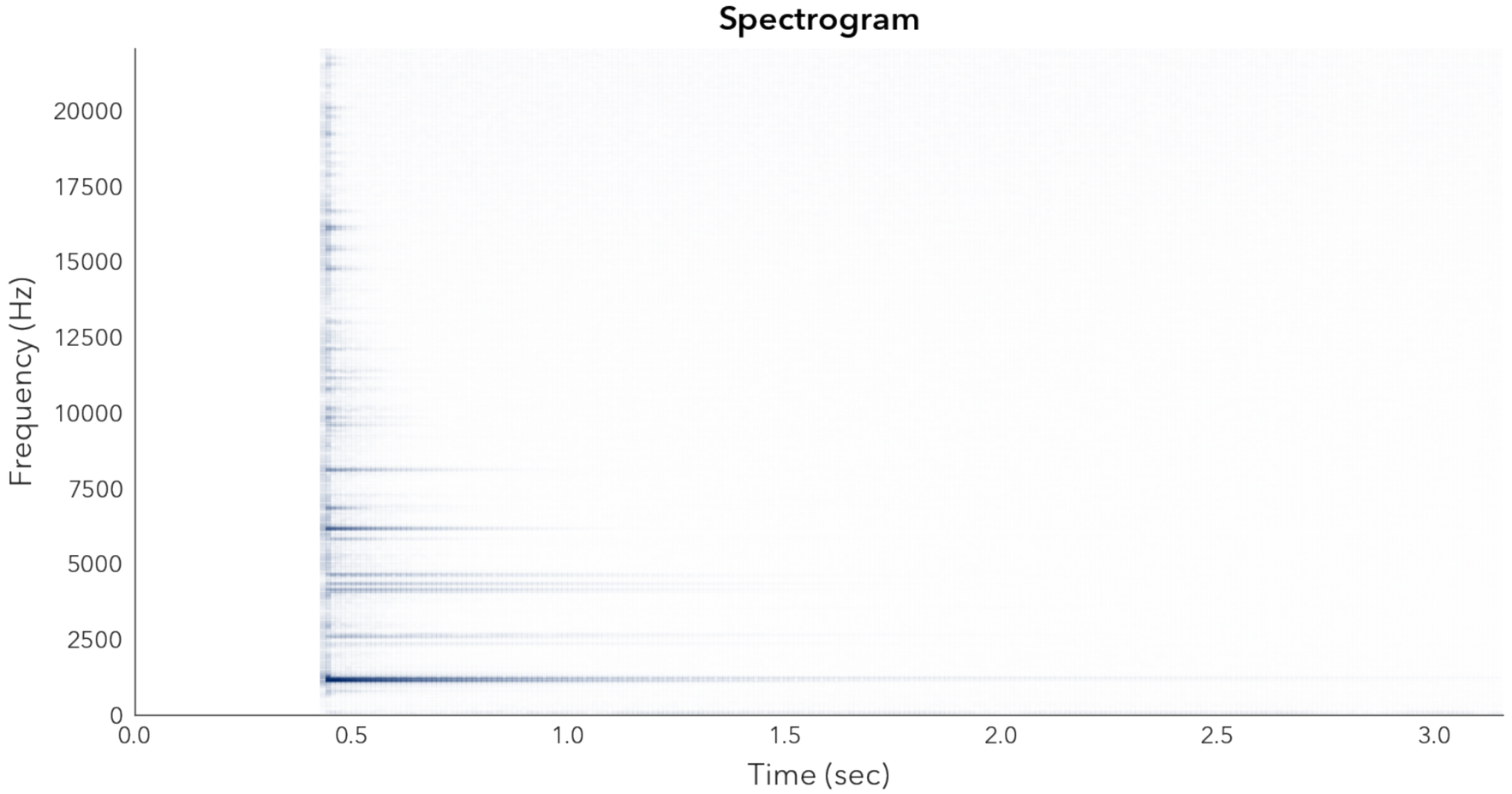
A starter

- What is this?



A magnitude spectrogram

- Representing energy over time & frequency



Factorizing a spectrogram

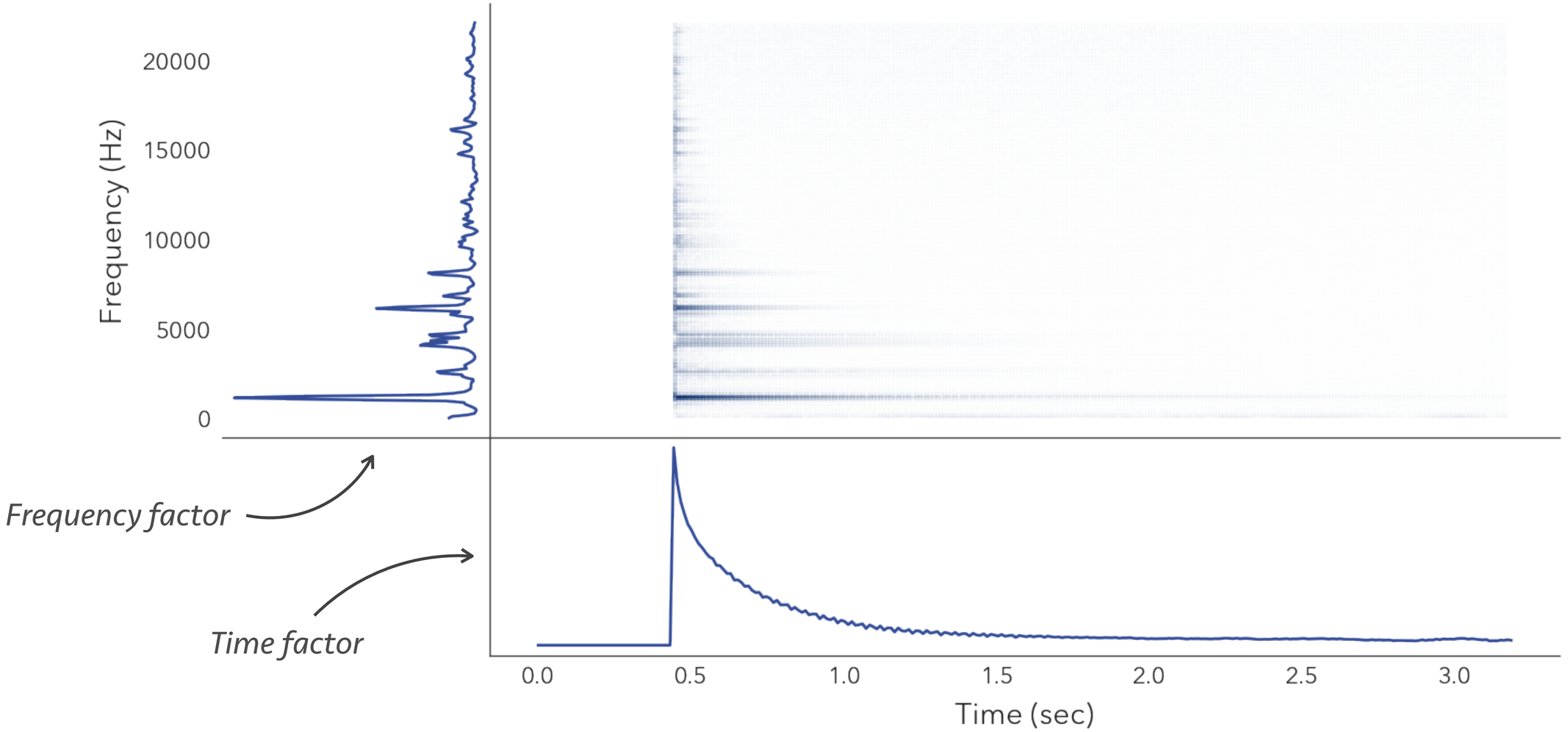
- We can approximate the 2D magnitude spectrogram as a product of two 1D functions
  - A *frequency factor* and a *time factor*

$$S(\omega,\tau) = F(\omega)A(\tau)$$

Spectrogram *Frequency factor* *Time factor*

Estimating the factors

- We average over each dimension



Nothing new here

- Frequency factor == Signal spectrum
  - Energy distribution over frequency

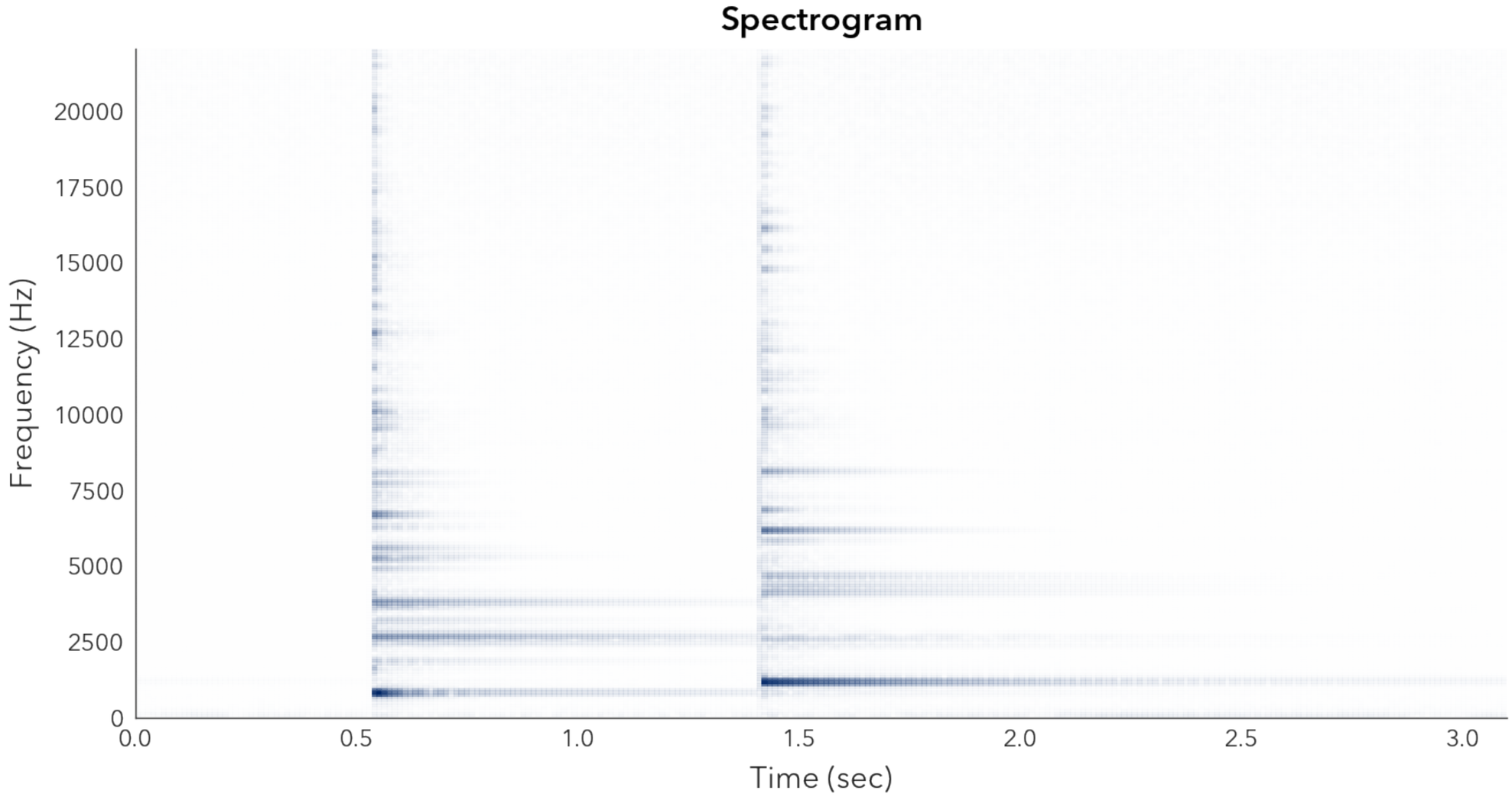
$$F(\omega) = \sum_{\tau} S(\omega, \tau)$$

- Time factor == Signal envelope
  - Energy distribution over time

$$A(\tau) = \sum_{\omega} S(\omega, \tau)$$

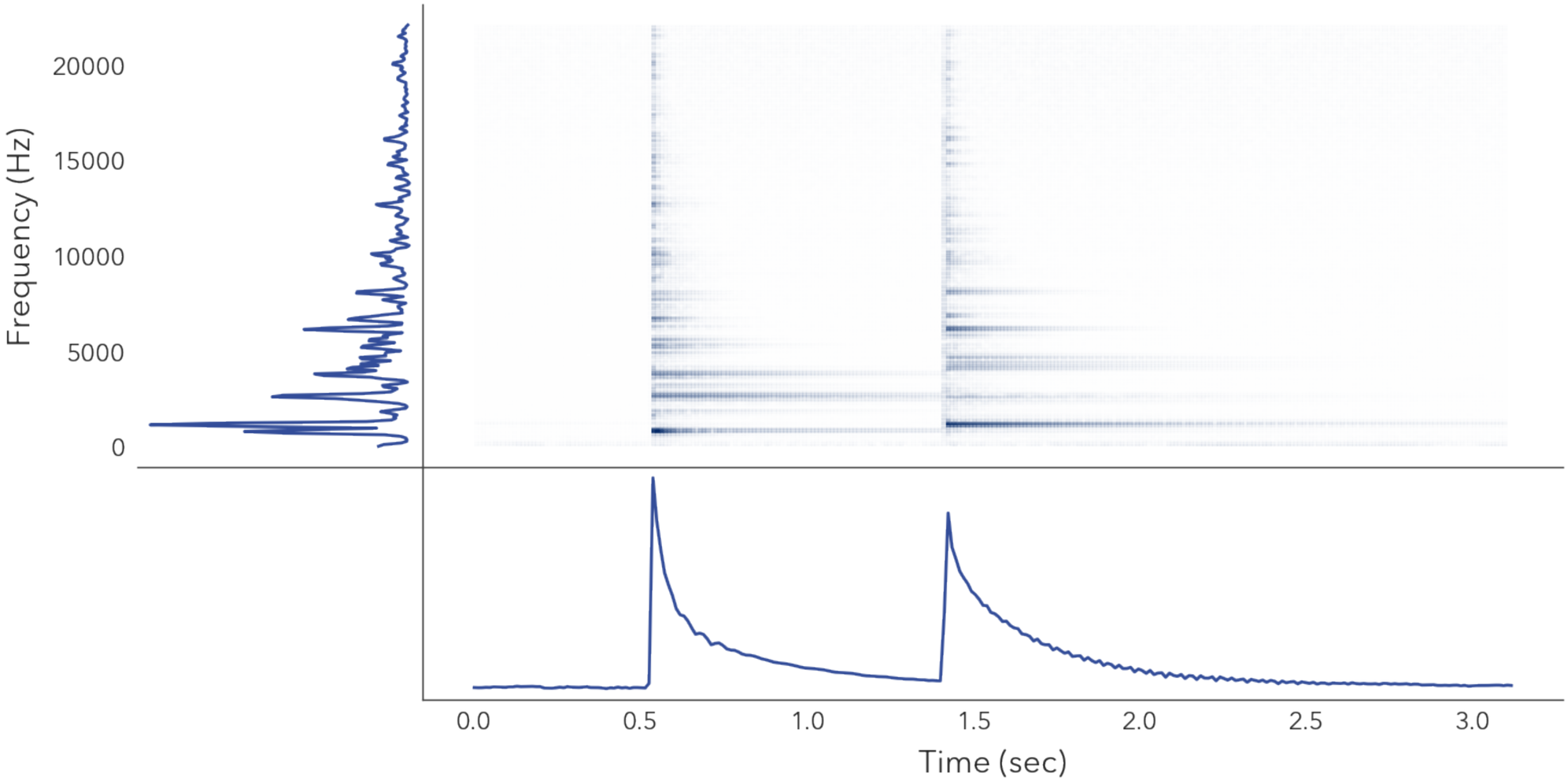
Another example

- What now?



Doesn’t factorize in a useful way

- The learned factors are “mixed”



What if?

- Could we get two sets of factors?
  - One for each element in the input

- Initial factor model:

$$S(\omega,\tau) = F(\omega)A(\tau)$$

- A "multiple factors" model:

$$S(\omega,\tau) = F_1(\omega)A_1(\tau) + F_2(\omega)A_2(\tau)$$

Learning this model

- Simple 2-stage process:
  - See how much each “pixel” is explained by each factor set

$$\gamma_i(\omega,\tau) = \frac{F_i(\omega)A_i(\tau)}{F_1(\omega)A_1(\tau) + F_2(\omega)A_2(\tau)}$$

  - Re-estimate factors using that weight

$$F_i(\omega) = \sum_{\tau} \gamma_i(\omega,\tau)S(\omega,\tau)$$

$$A_i(\omega) = \sum_{\omega} \gamma_i(\omega,\tau)S(\omega,\tau)$$

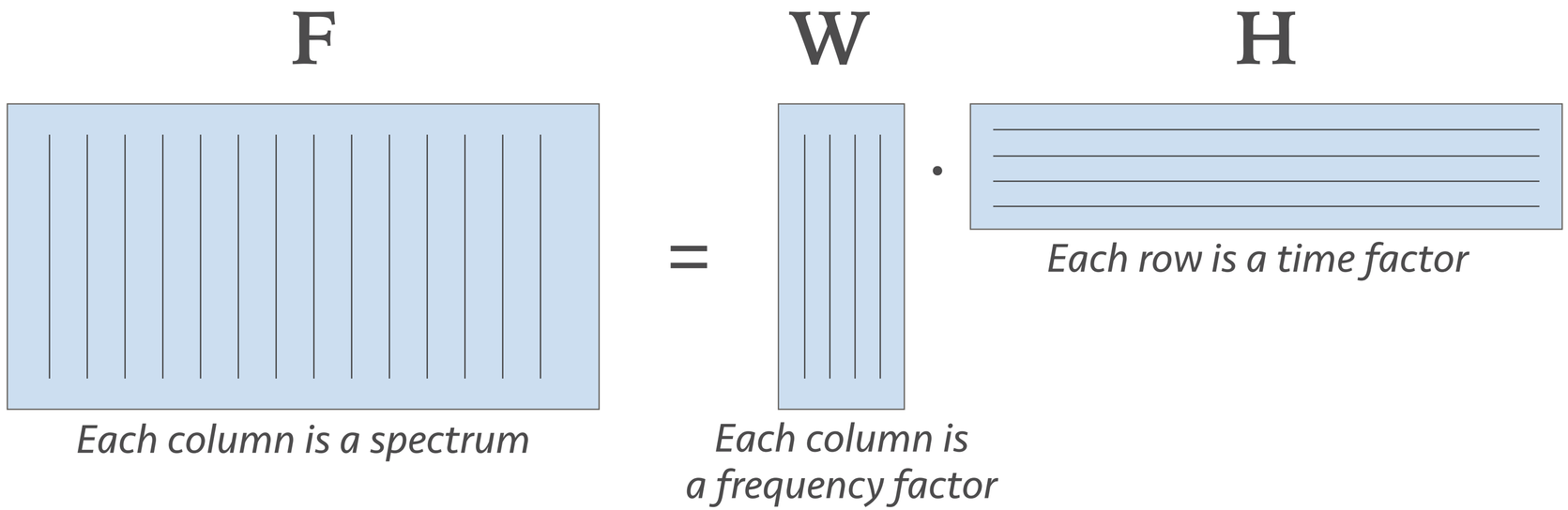
  - Repeat …

General algorithm

- Non-Negative Matrix Factorization

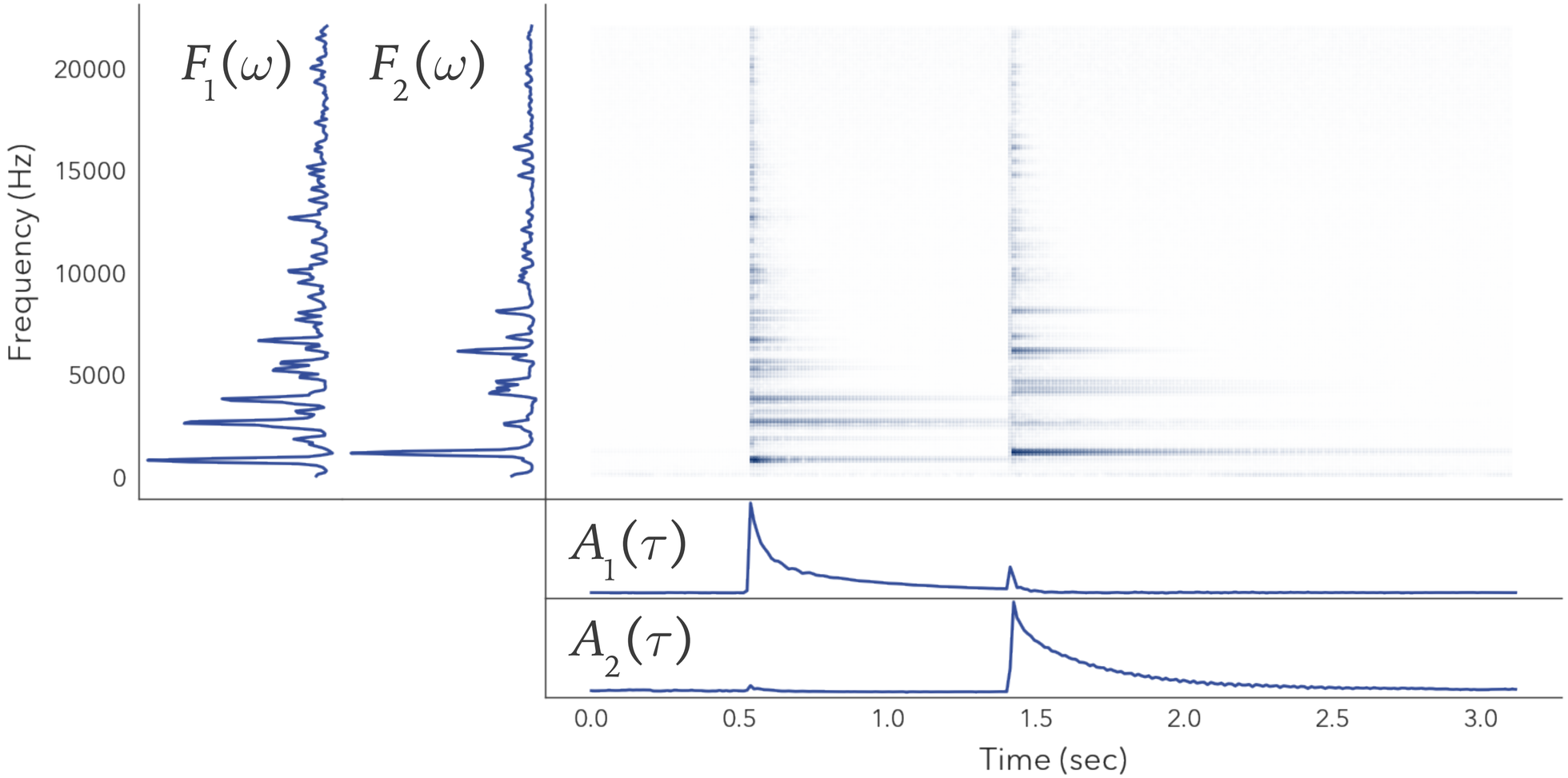
$$\mathbf{F} \approx \mathbf{W} \cdot \mathbf{H}$$

$$\mathbf{F} \in \mathbb{R}_{+}^{M \times N}, \mathbf{W} \in \mathbb{R}_{+}^{M \times K}, \mathbf{H} \in \mathbb{R}_{+}^{K \times N}$$

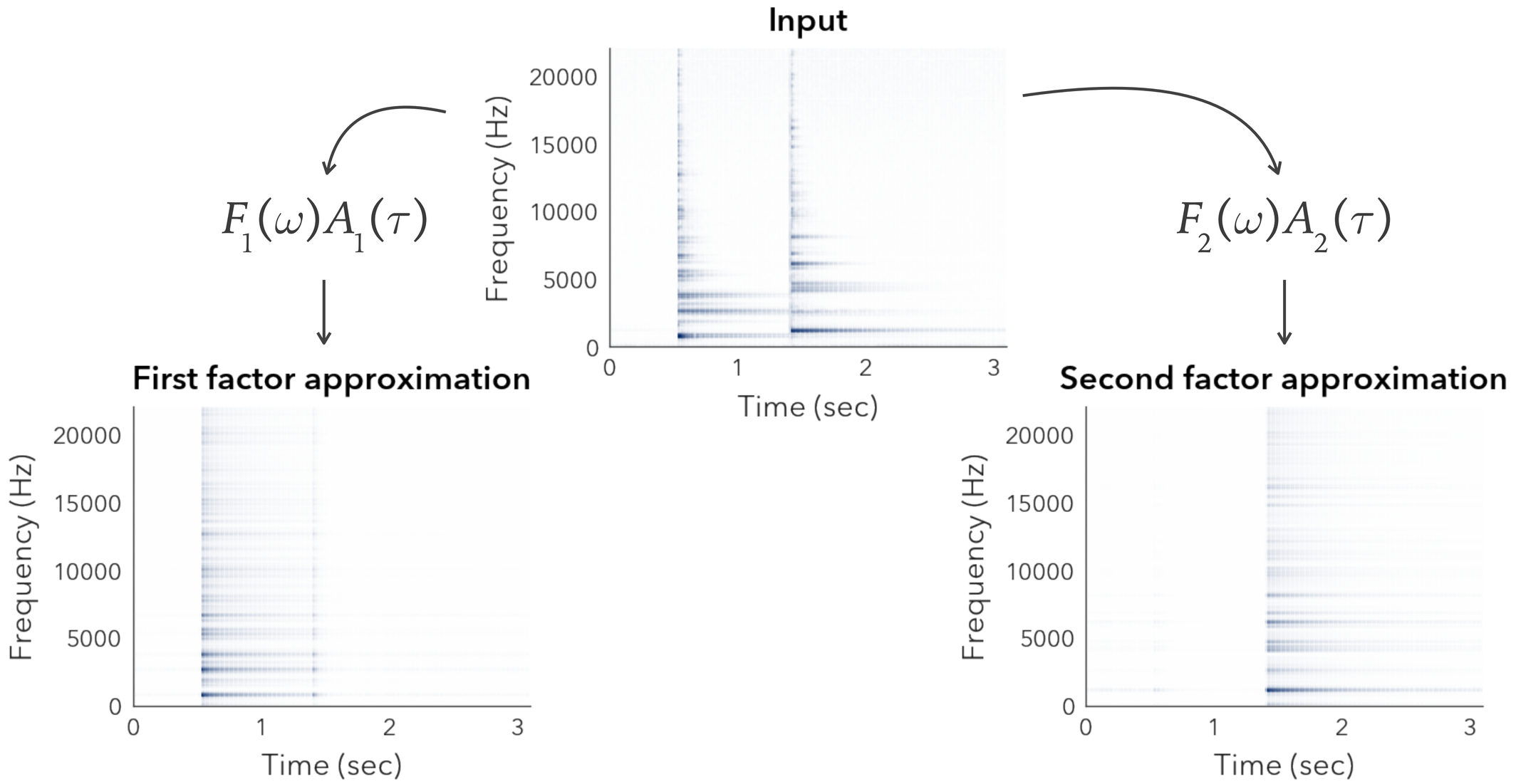


What does this do?

- More useful representation!



Factors learn sources



Going back to time

- Original input had amplitude *and* phase

$$S(\omega,\tau) = \left|\mathrm{STFT}\left(x(t)\right)\right|$$

$$\Phi(\omega,\tau) = \angle\mathrm{STFT}\left(x(t)\right)$$

- To get each source y_i in the time domain we assume:

$$\left|\mathrm{STFT}\left(y_i(t)\right)\right| = F_i(\omega)A_i(\tau)$$

$$\angle\mathrm{STFT}\left(y_i(t)\right) = \Phi(\omega,\tau)$$

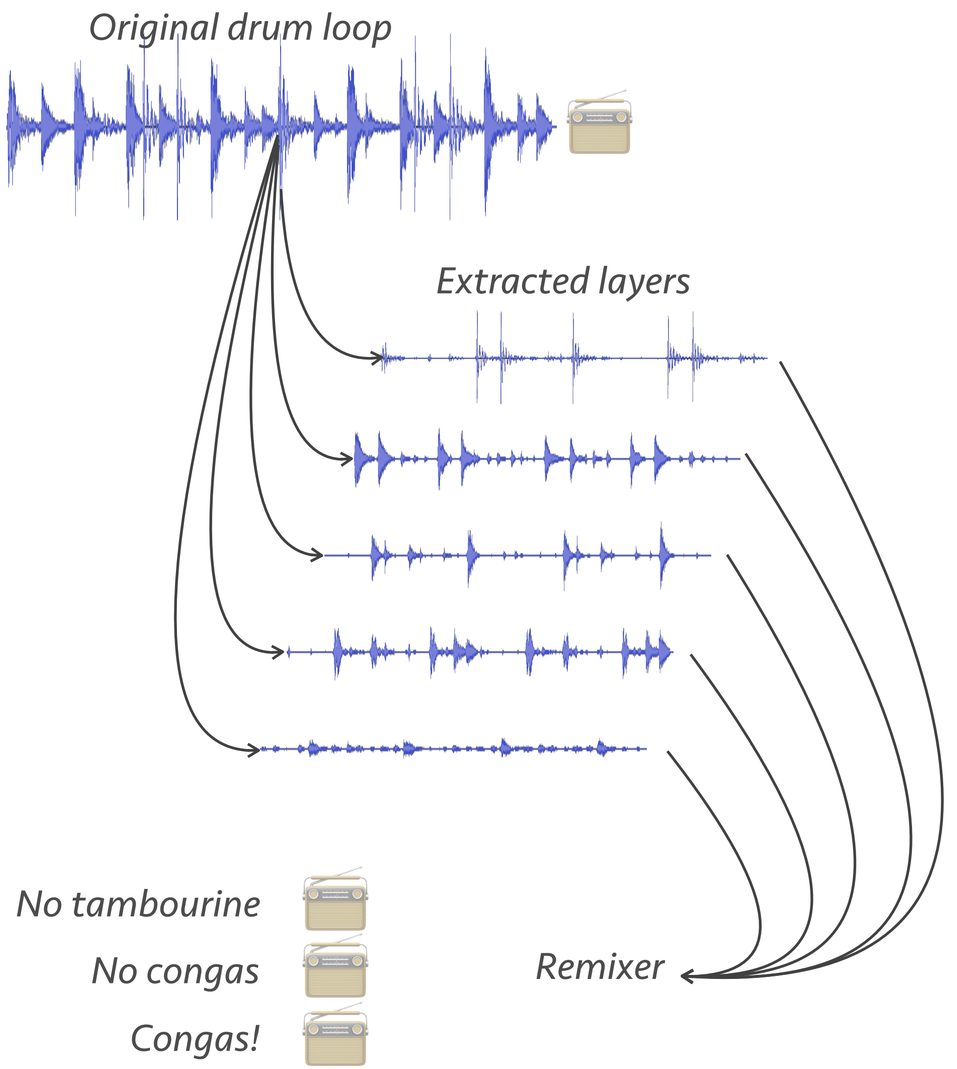
- And we can now extract the two bells:

Source 1

Source 2

Audio remixing

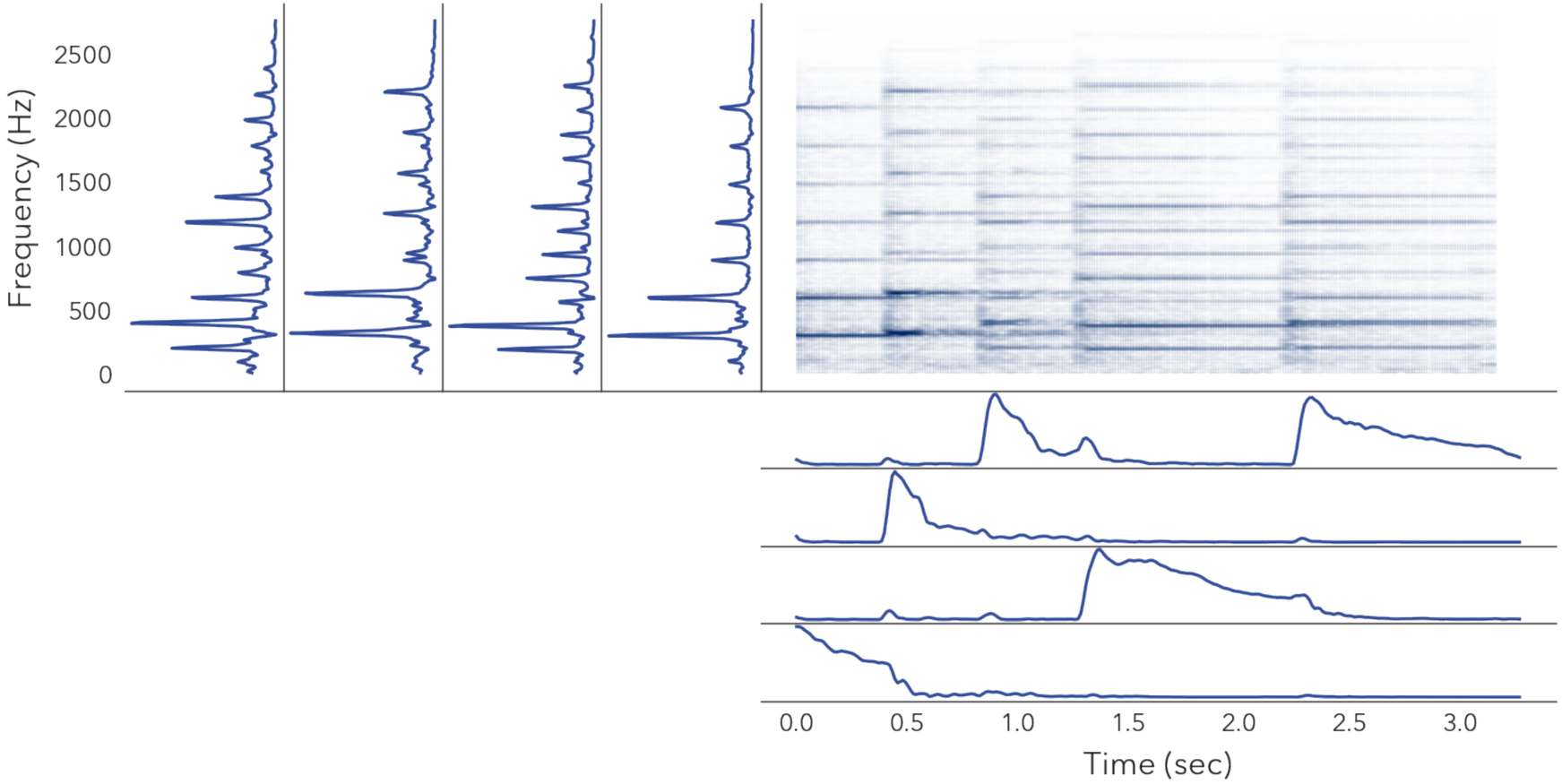
- We can use this idea to remix different sounds in a recording
- Factorize the input
  - Learns salient sounds in it
- Remix factors as desired
  - E.g. remove a sound, or make it louder/softer



Trying this on music

- Analyzing a piano clip
  - What are the learned factors?

5 notes, 4 of them unique

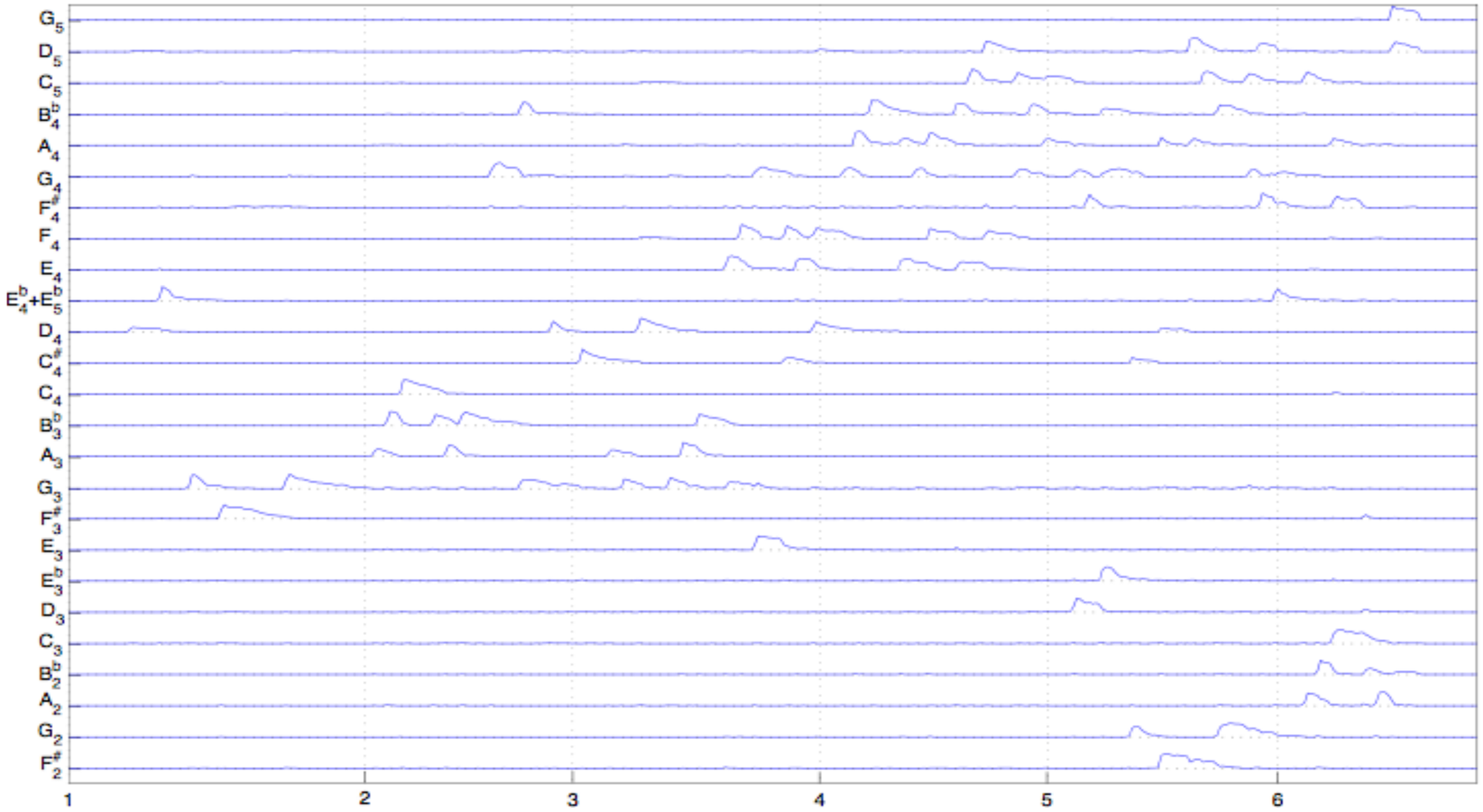


Polyphonic music transcription

- Assuming simple note spectra
  - We can decompose music into notes
    - Each factor $F_i(\omega)$ is a note spectrum
- Problems
  - How many notes do we need to extract?
  - Notes need to be represented by a single spectrum

Example

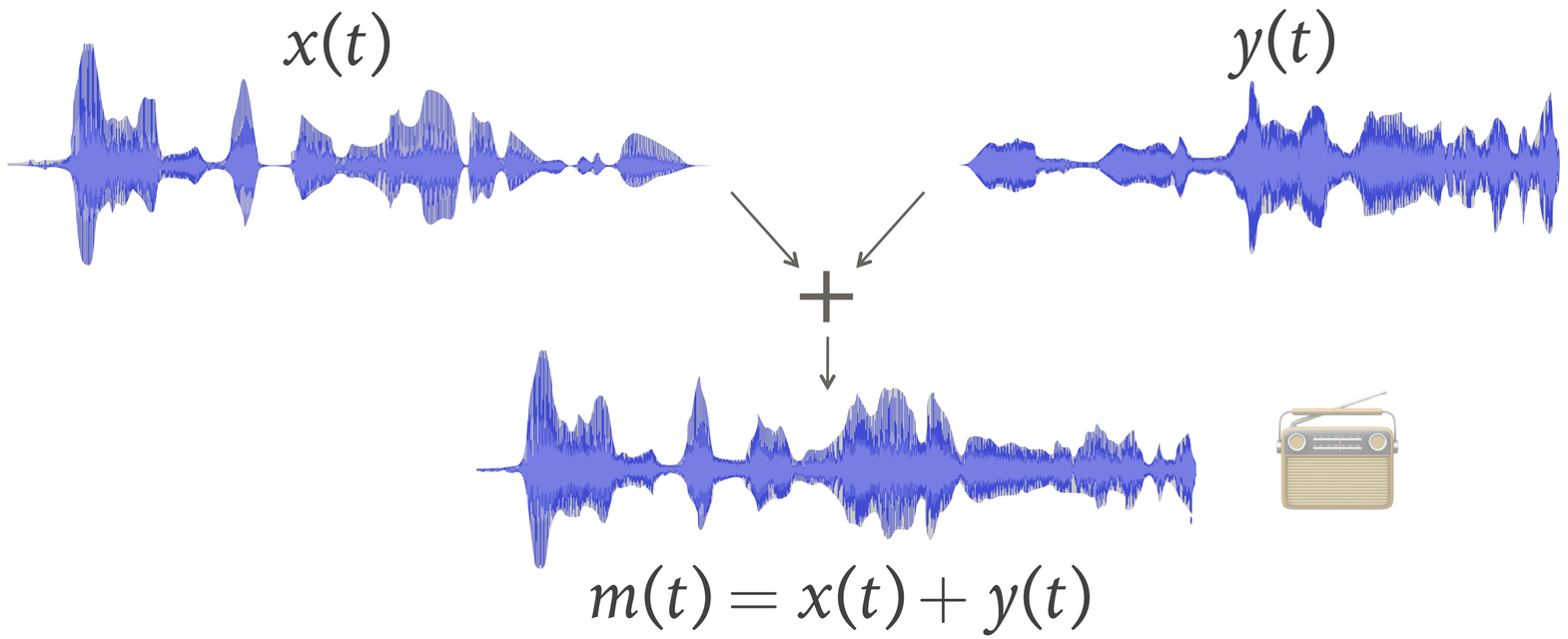
- Bach Fugue XVII in Gm



But …

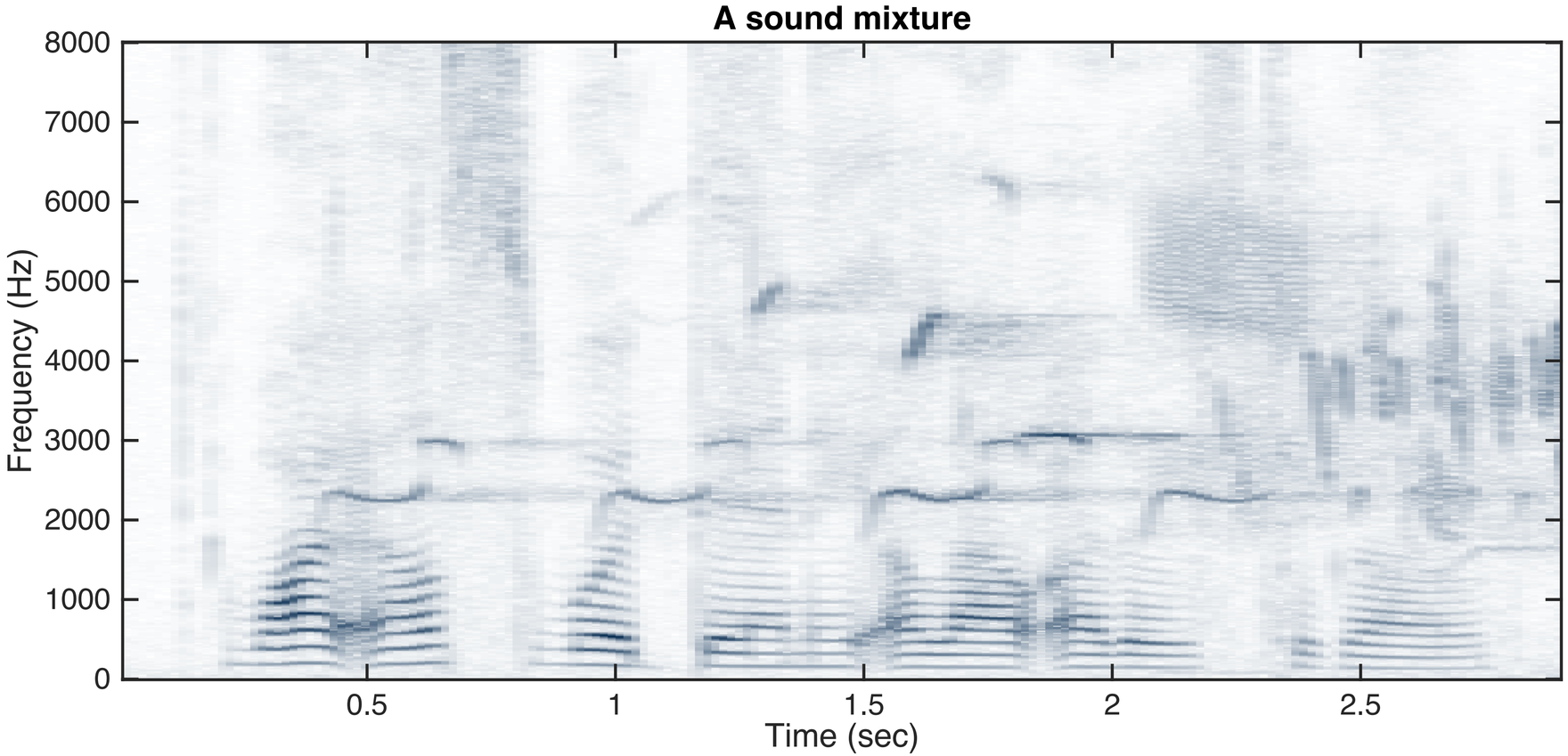
- We can only extract/detect simple sounds
  - Why is that?
- How about real-world source separation?

Defining the problem



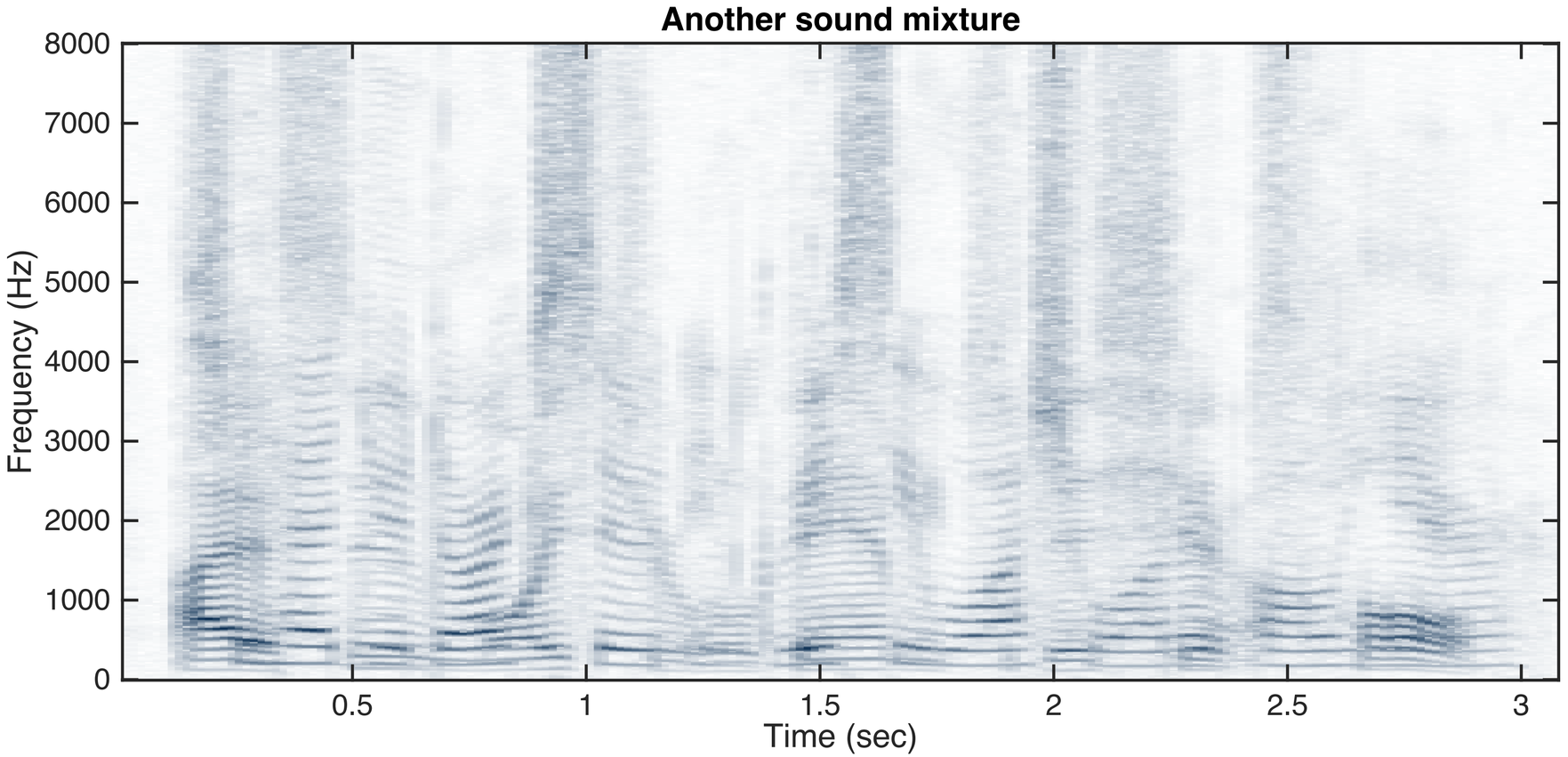
- Very ill-defined problem!
  - *Single-channel source separation*

The name of the game



- Finding signal priors to perform separation
  - School a: Perceptual-minded approaches
  - School b: Statistical approaches

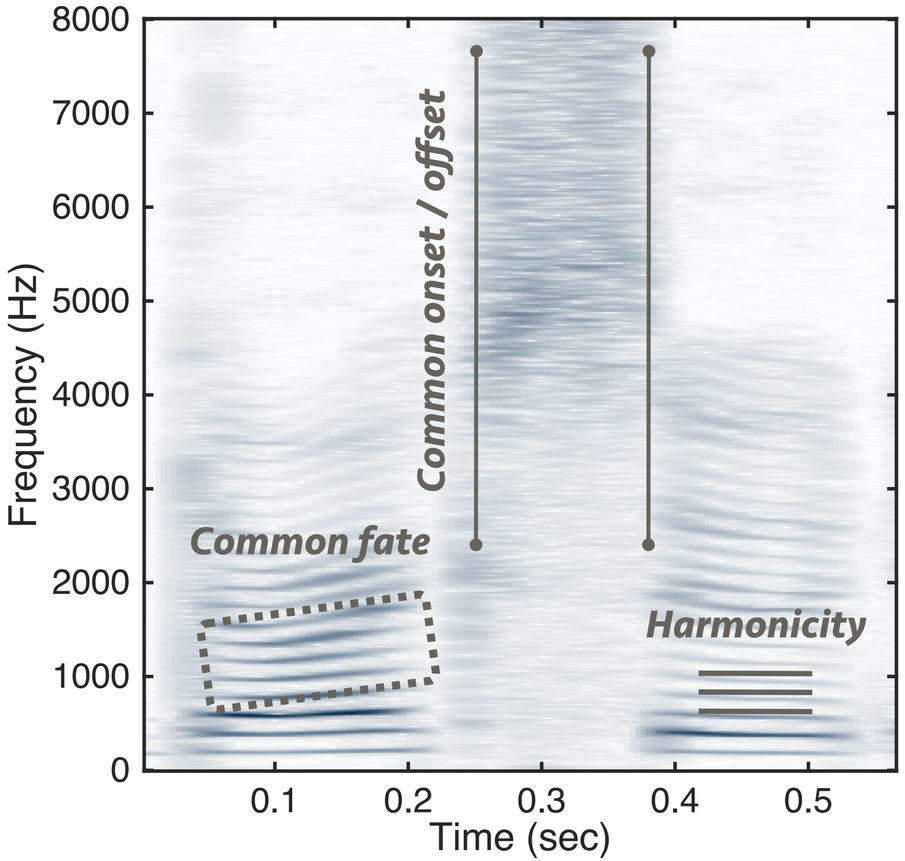
The name of the game



- Finding signal priors to perform separation
  - School a: Perceptual-minded approaches
  - School b: Statistical approaches

Perceptual approaches

- Computational Auditory Scene Analysis
  - Driven by psychoacoustic experiments



Brown mix *Sep*

Ellis mix *Sep*

Wang mix *Sep*

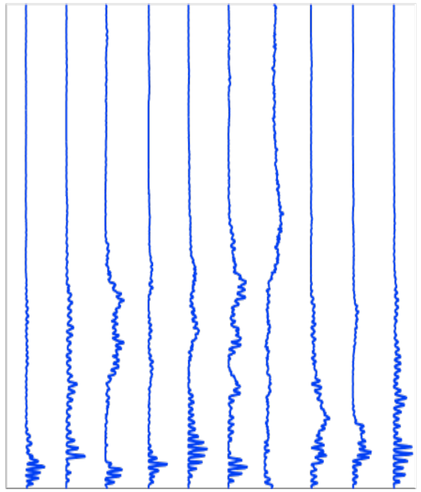
Some (general) statistical approaches

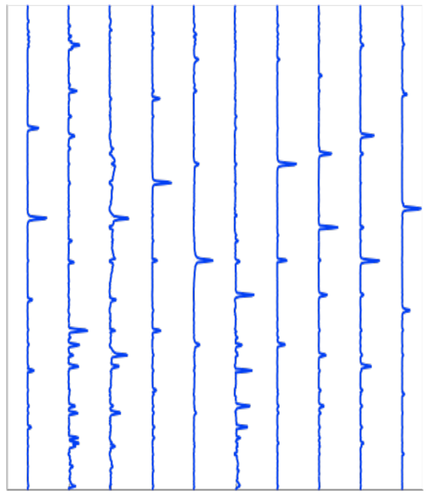
- Approaches with general source assumptions
  - Lee and Jang
    - ICA dictionaries of time waveforms
  - Reyes, Jojic and Ellis
    - Graphical model on TF distributions
  - Lagrange, et al.
    - Normalized cuts
  - Bach and Jordan
    - Spectral clustering for perceptual grouping
- Things aren't great …

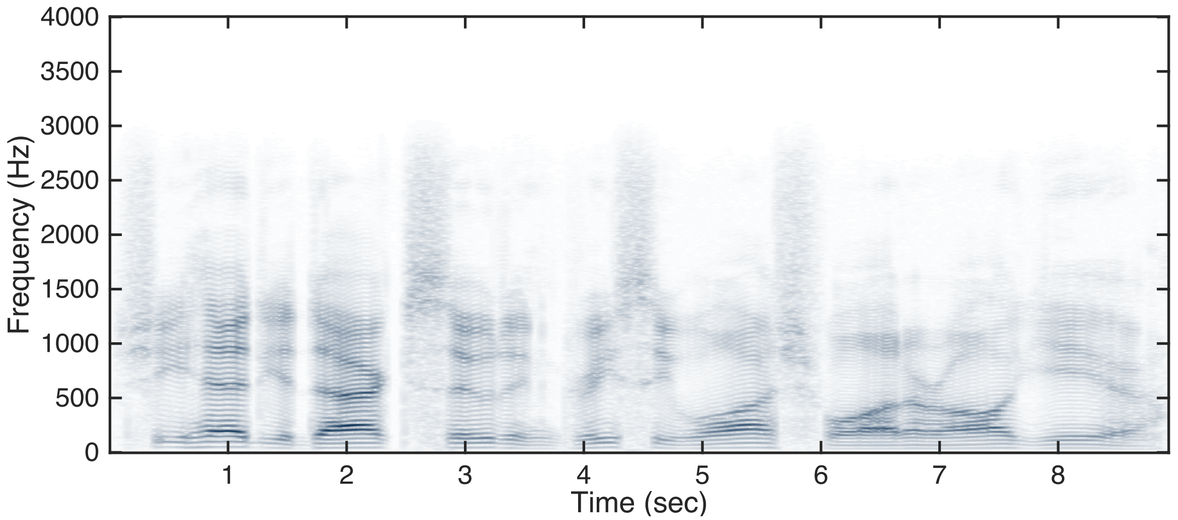
Giving up on unsupervised methods

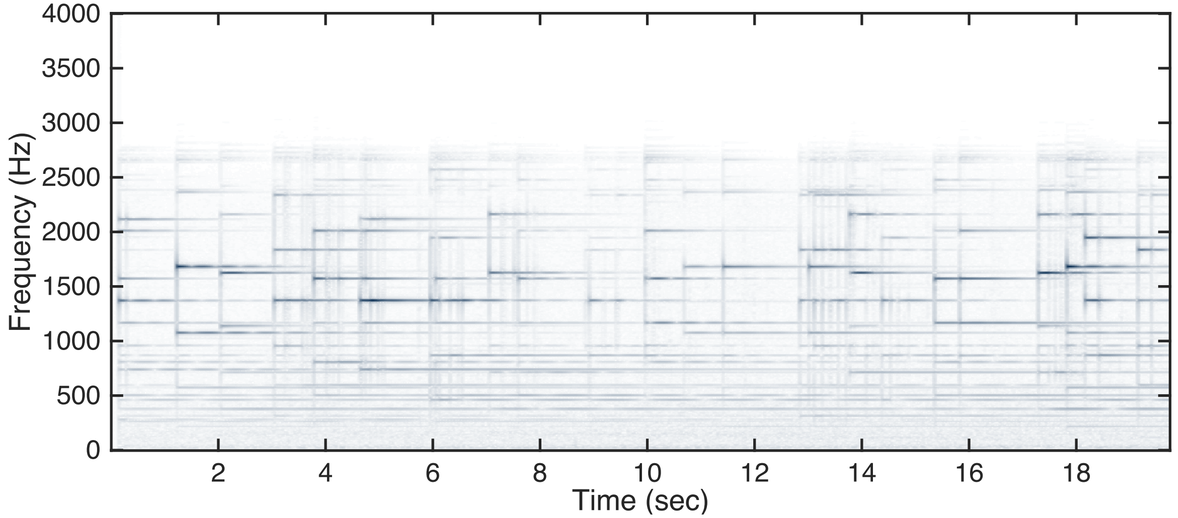
- It is hard to define a source structure
  - We should learn it instead
- Supervised source separation
  - Use training data as a hint towards what you want

Learning factor dictionaries









Speech recording

Chime recording

Mixtures of sounds

- Use spectrogram additivity
  - combine models to explain mixture

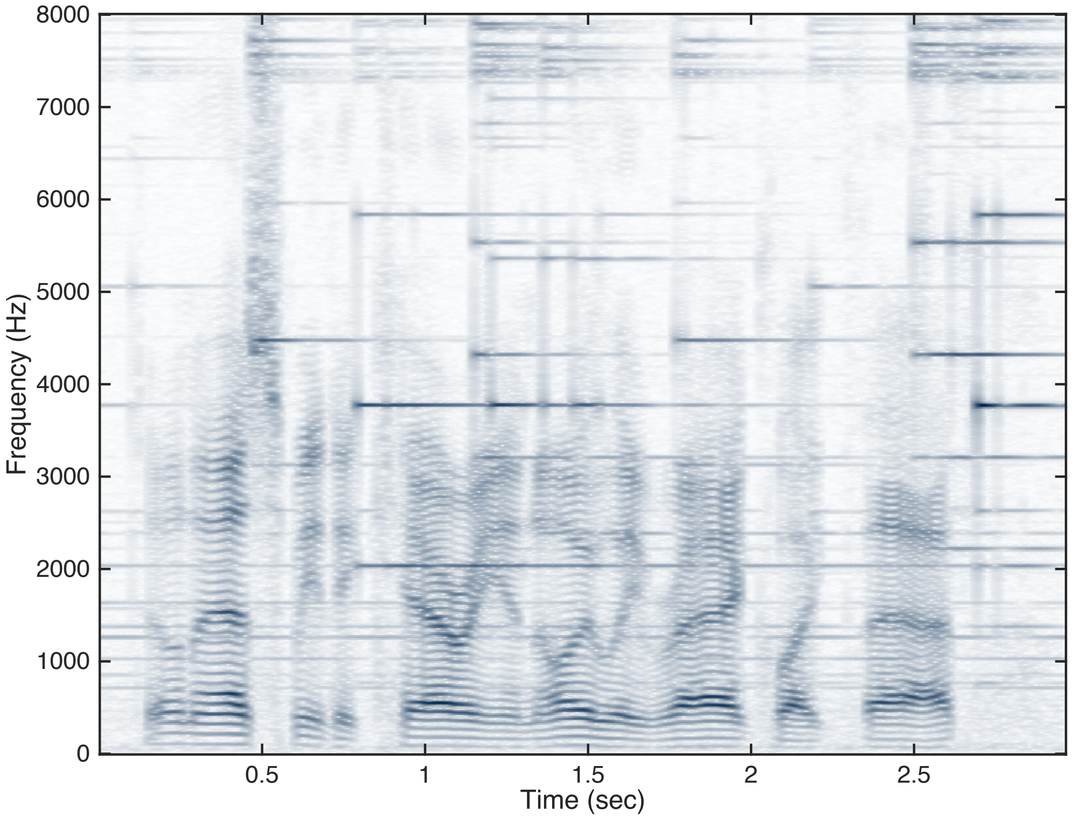
$$\mathbf{F} = \begin{bmatrix} \mathbf{W}_{chimes} & \mathbf{W}_{speech} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{chimes} \\ \mathbf{H}_{speech} \end{bmatrix}$$

Known/fixed *Estimated*

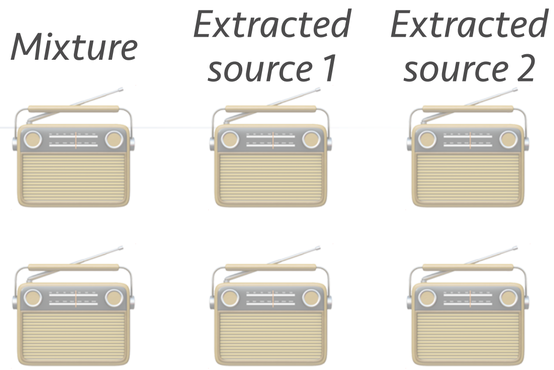
- We estimate only $\mathbf{H}$

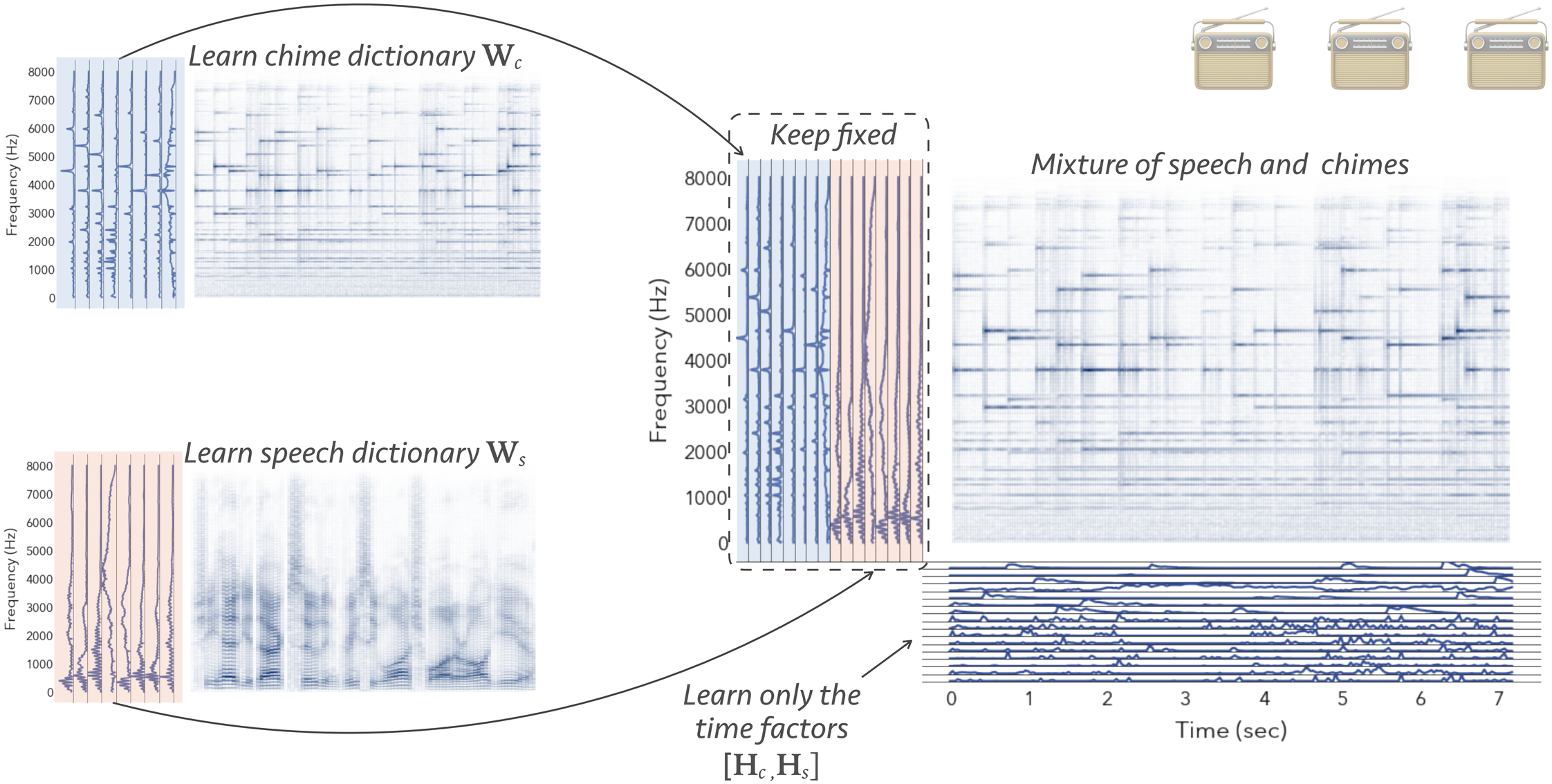
- The known frequency factors claim only the parts that they can fit best

Speech & chimes



Huh?


Mixture
Extracted source 1
Extracted source 2


Learn chime dictionary $\mathbf{W}_c$
Frequency (Hz)
8000
7000
6000
5000
4000
3000
2000
1000
0
Keep fixed
Mixture of speech and chimes
Learn speech dictionary $\mathbf{W}_s$
Learn only the time factors $[\mathbf{H}_c, \mathbf{H}_s]$
0
1
2
3
4
5
6
7
Time (sec)

What if we do not have all the models?

- Same as before, only one model:

$$\mathbf{F} = \begin{bmatrix} \mathbf{W}_{known} & \mathbf{W}_{unknown} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{known} \\ \mathbf{H}_{unknown} \end{bmatrix}$$

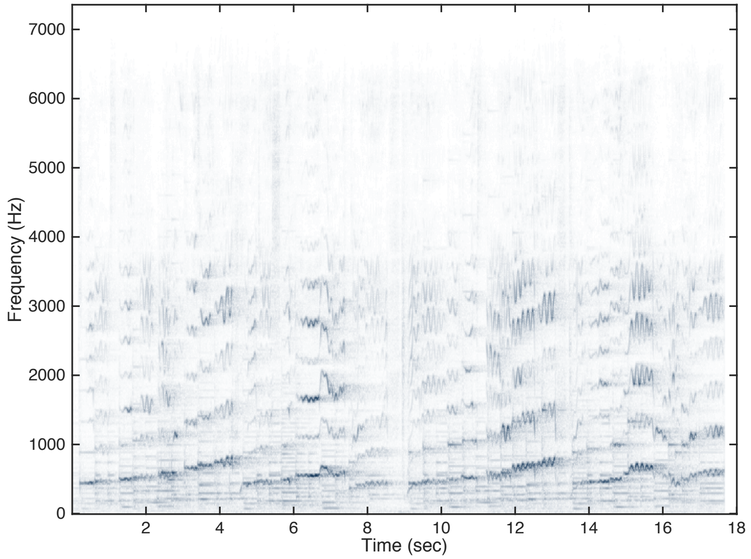
Known/fixed — $\mathbf{W}_{known}$

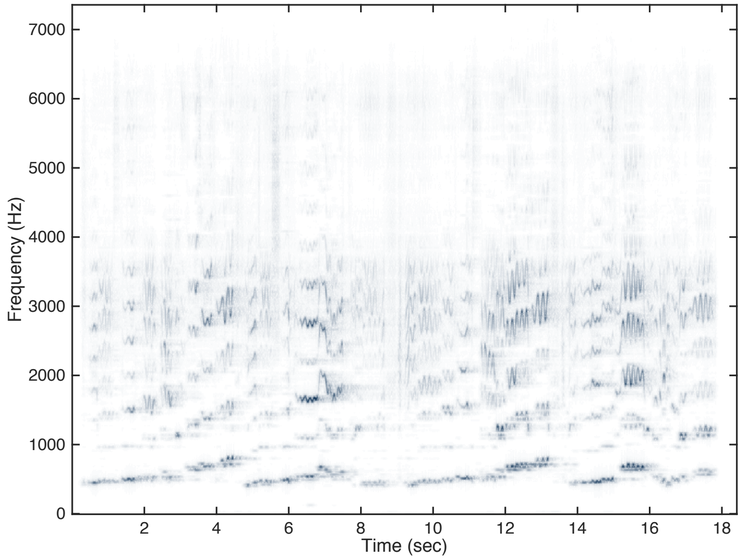
Estimated — $\mathbf{H}_{unknown}$

- Learn weights and unknown bases
  - Extra learned factors converge to unmodeled sounds in the mixture

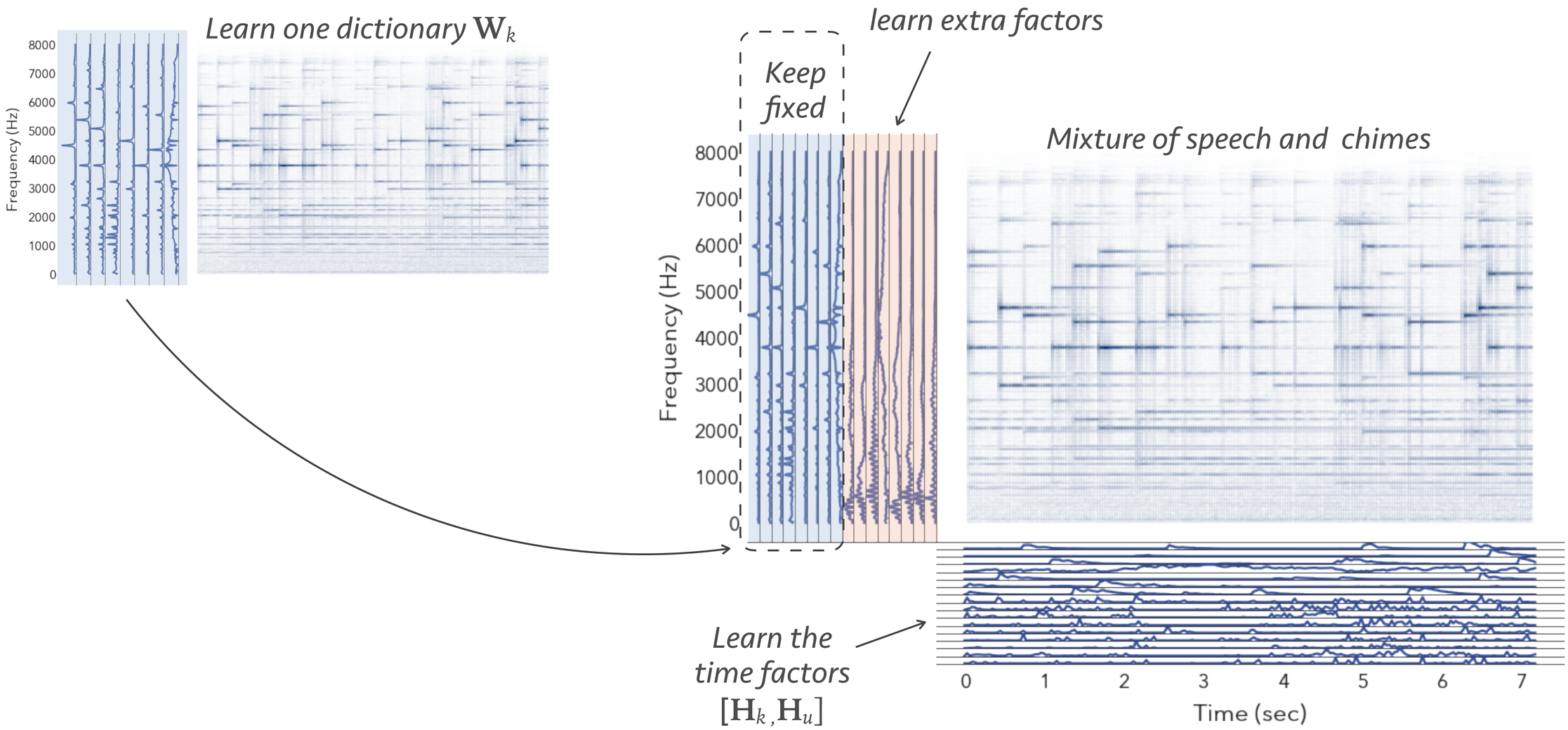
Soprano & Piano

Extracted soprano





Huh again?


Learn one dictionary $\mathbf{W}_k$
Frequency (Hz)
8000
7000
6000
5000
4000
3000
2000
1000
0
learn extra factors
Keep fixed
Mixture of speech and chimes
Learn the time factors $[\mathbf{H}_k, \mathbf{H}_u]$
0
1
2
3
4
5
6
7
Time (sec)

Using in denoising

- Two cases
  - Have noise model, extract target
  - Have target model, extract noise
- More flexible than traditional spectral subtraction
  - It can model changing sound spectra
    - Remember the wind denoising example?
- Applies to many problems
  - Denoising, separation, karaoke, …

Speech + the beauty of mechanics

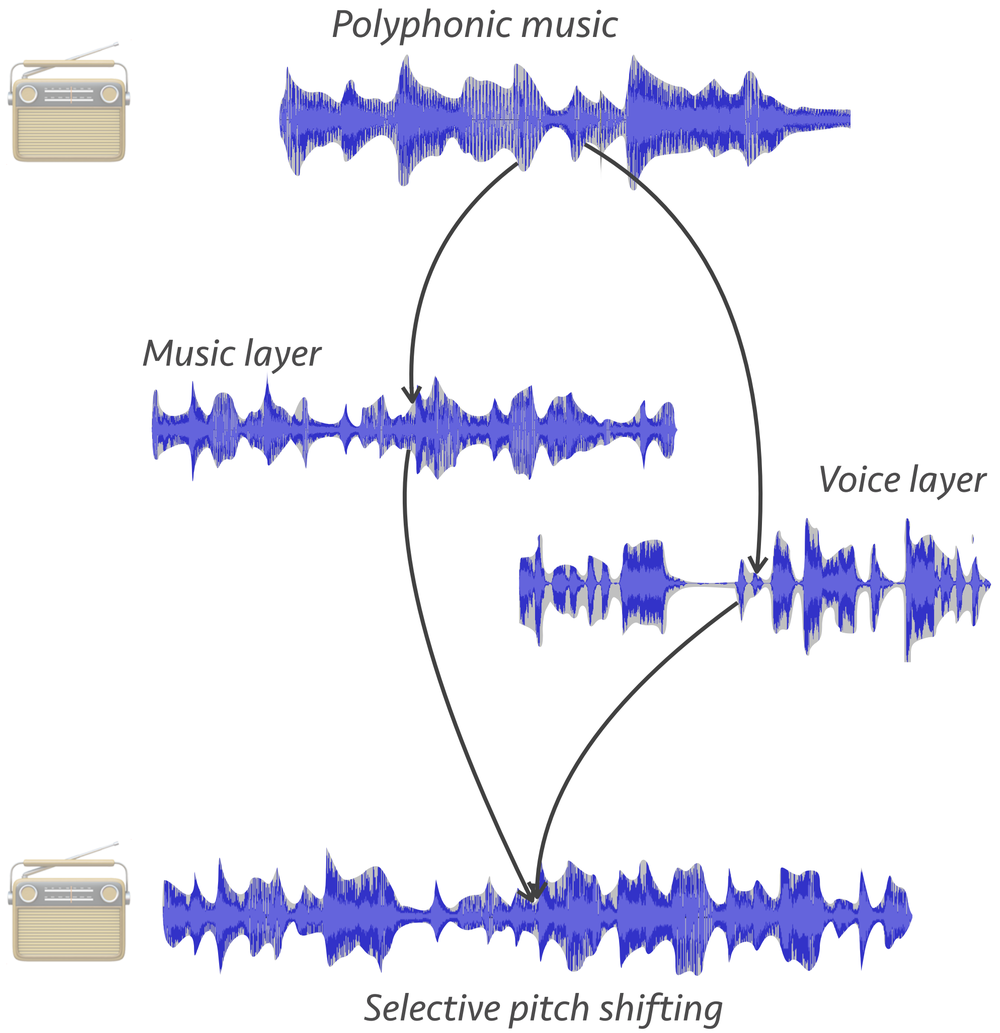
Wideband noise

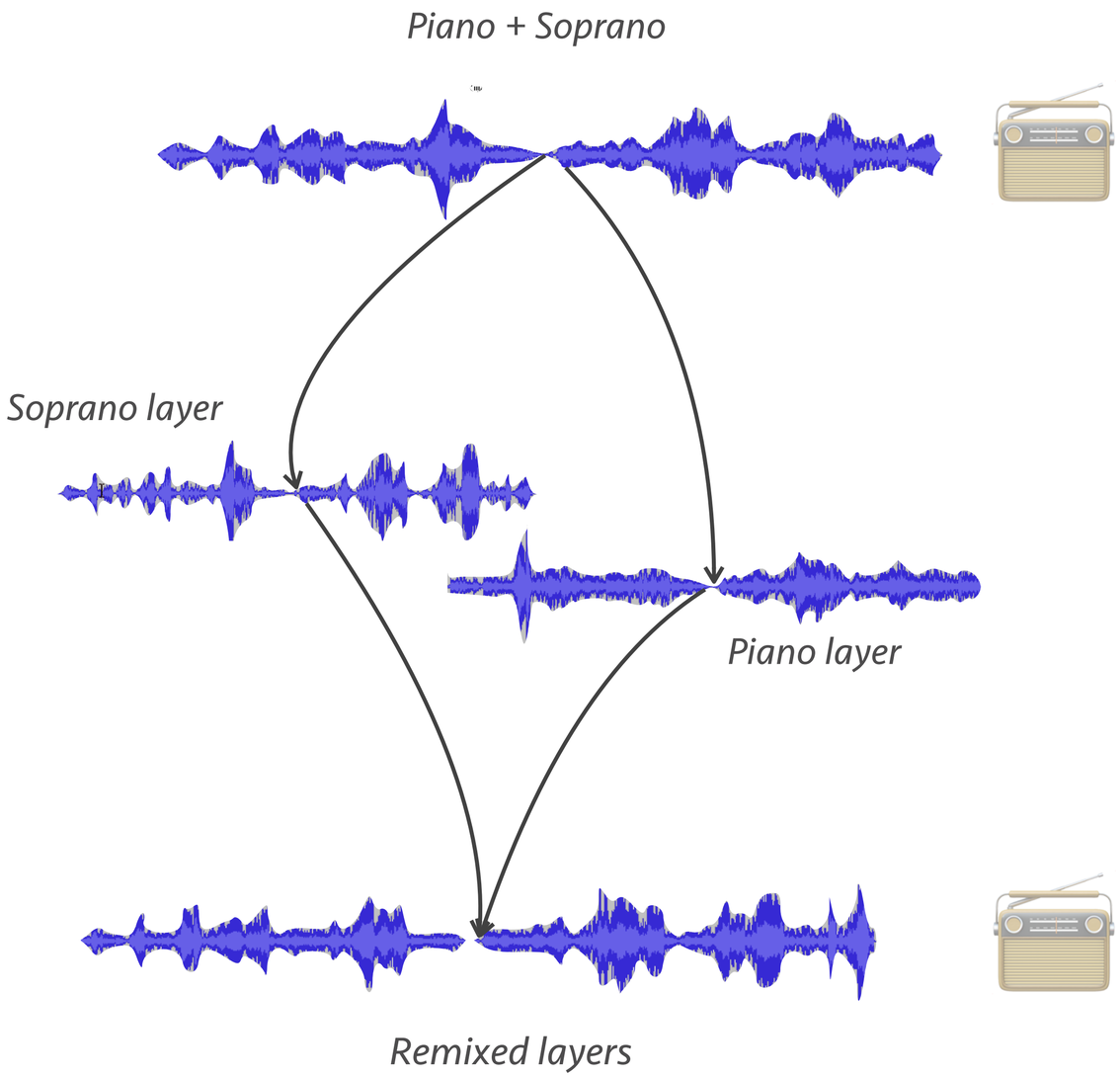
Extracted speech

Loosely correlated "noise"

Denoised sound

Audio layer editing





Going A/V

- We can also extract sources audio/visually

Output

Input

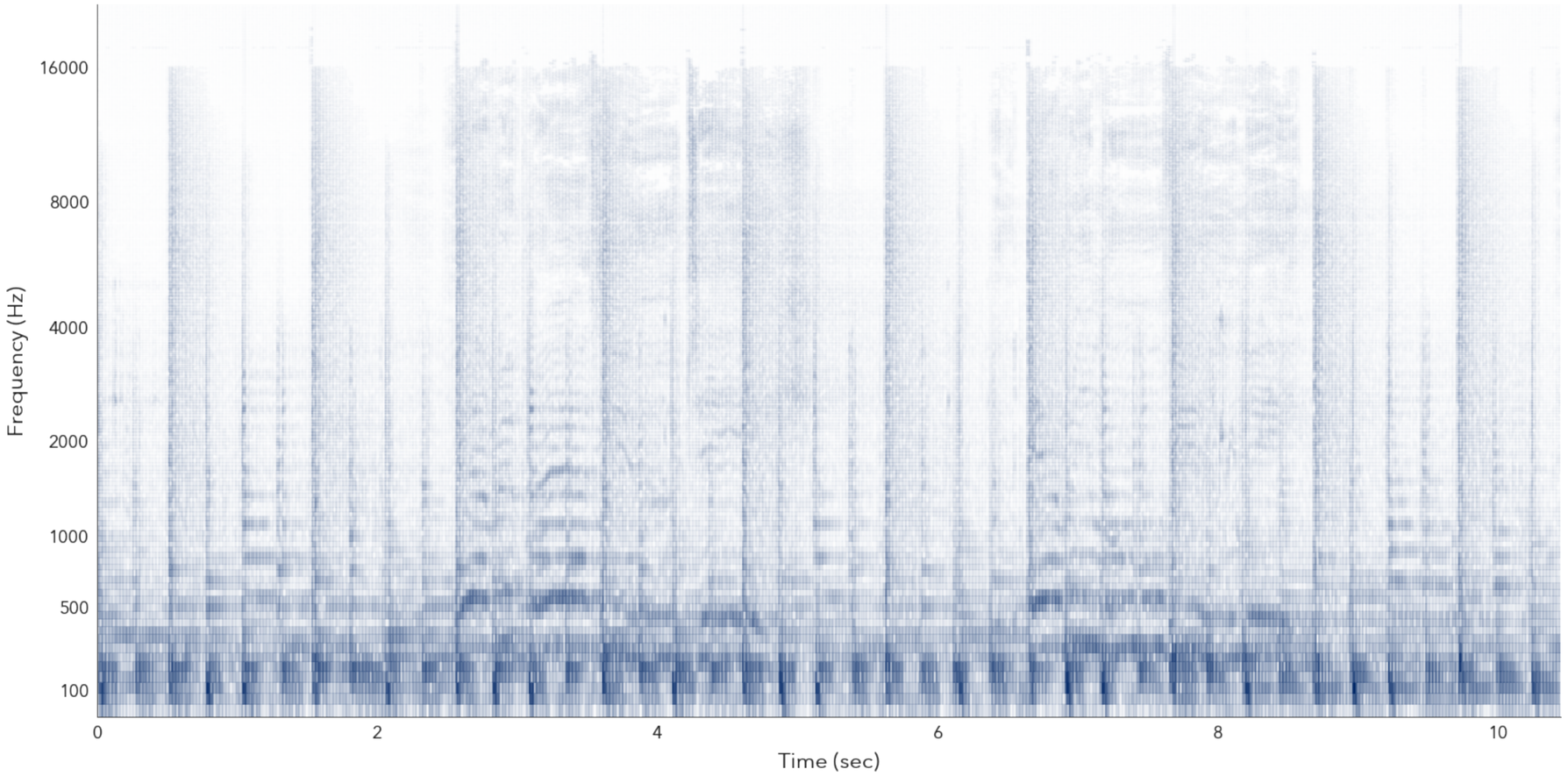
What about really difficult cases?

- How do we select a specific source?
  - One for which we do not have a dictionary?

- We can provide hints from a user input
  - E.g. have the user vocalize the target
  - Give both temporal and spectral hints

Some motivation

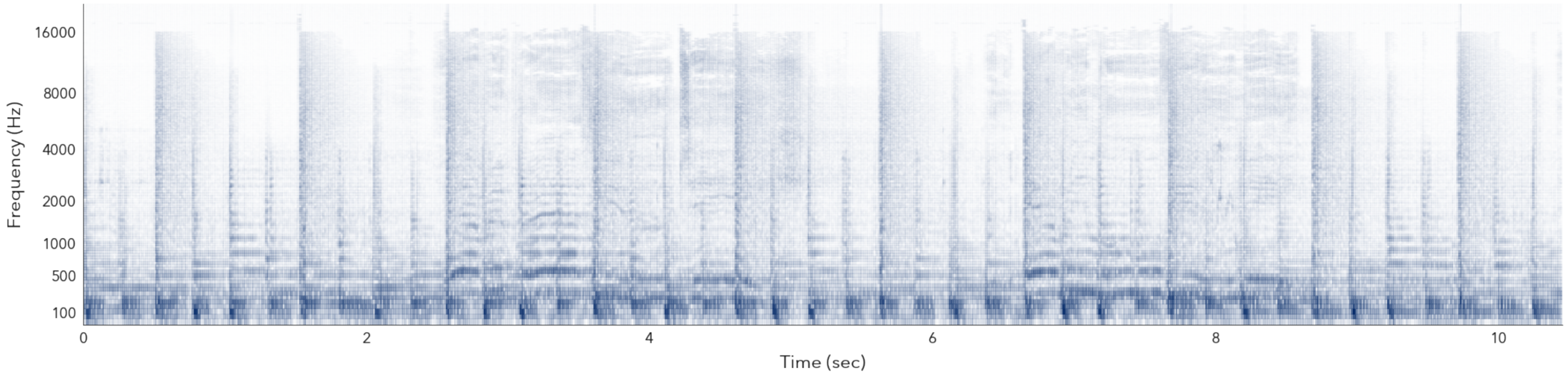
- Can we remove the main source?
  - What is the main source? How do we select it?

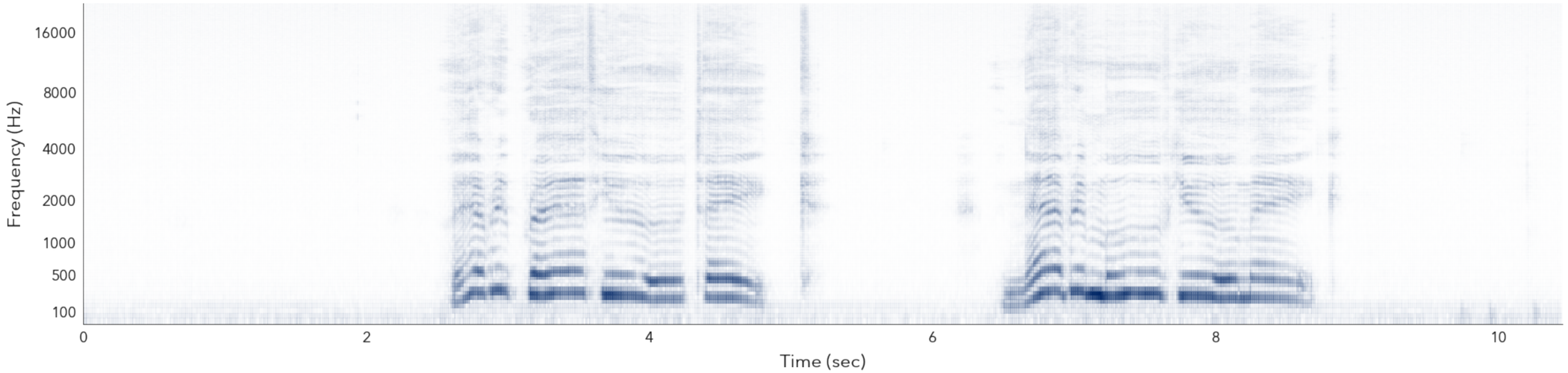


Basic idea

- Vocalize the target sound
  - Provides spectral/time factors similar to the ones in the target
- Perform unsupervised separation
  - Start with the user-provided factors as starting point
    - These will quickly explain what is closest to them
    - Extra factors will model the rest

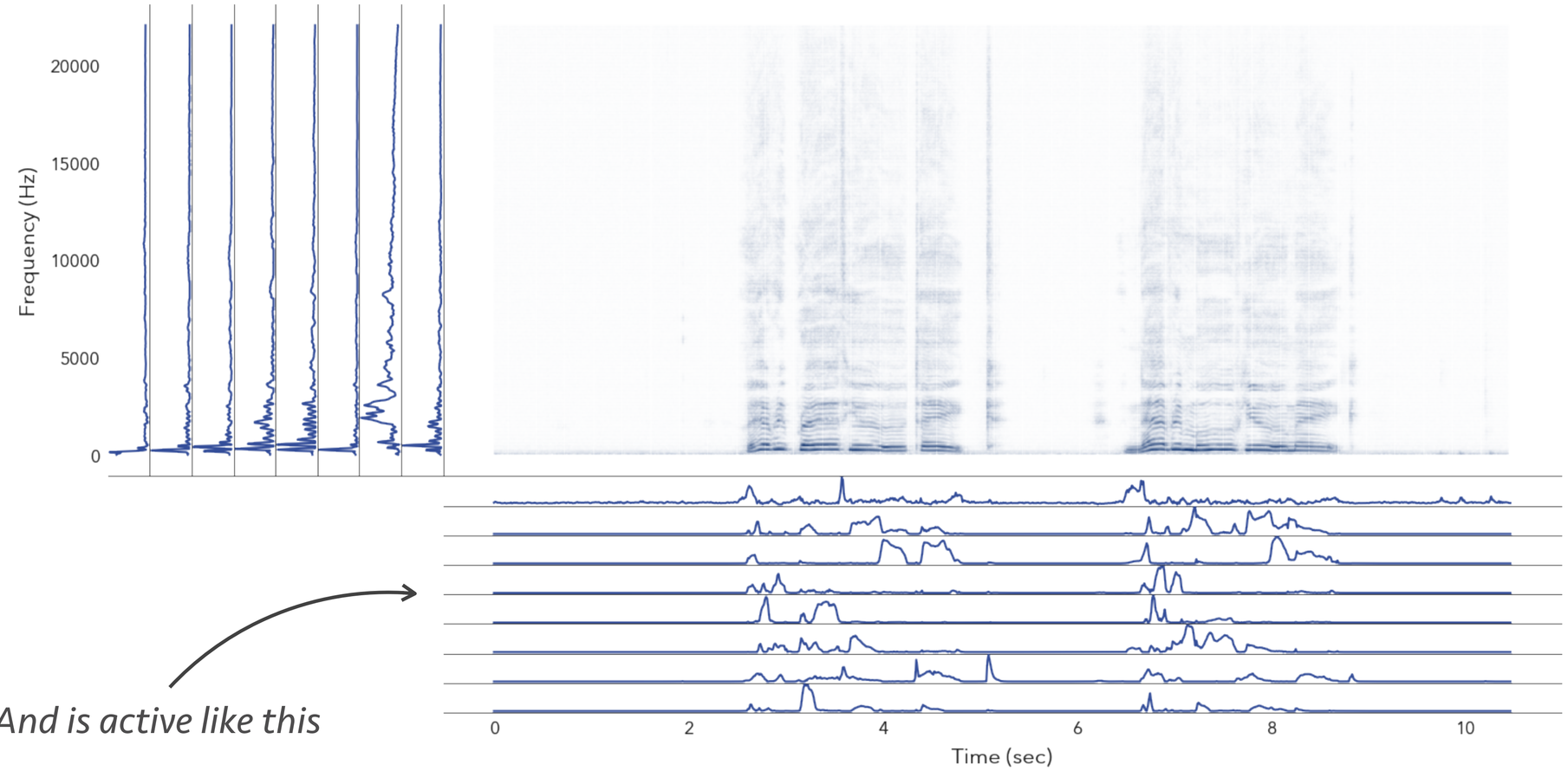
Pointing to a source in the mix





Learning something useful

- Get spectrum and timing info for target source

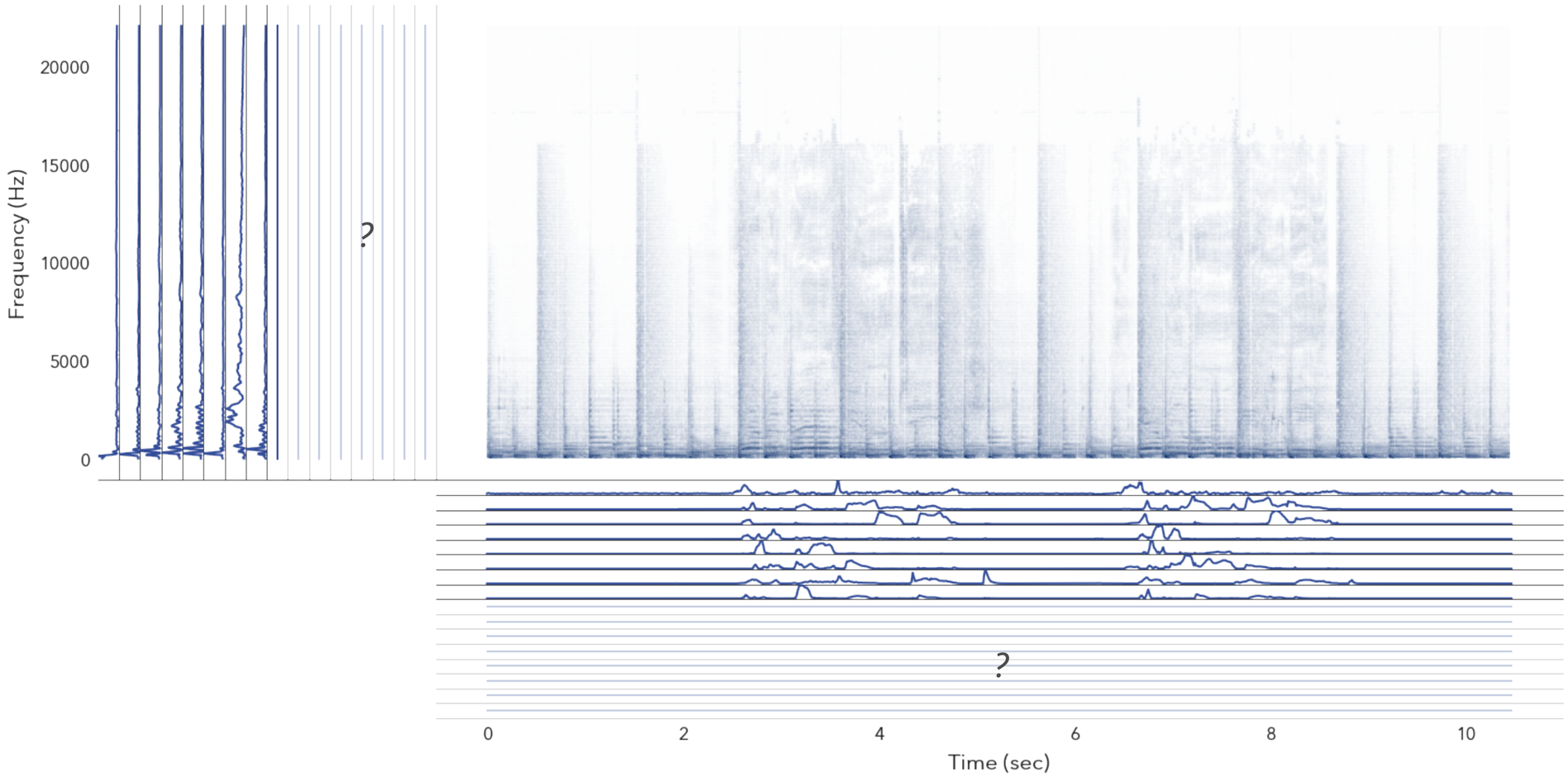


We need something that sounds like this

And is active like this

Use learned model for separation

- Add extra factors to explain the rest of the sounds

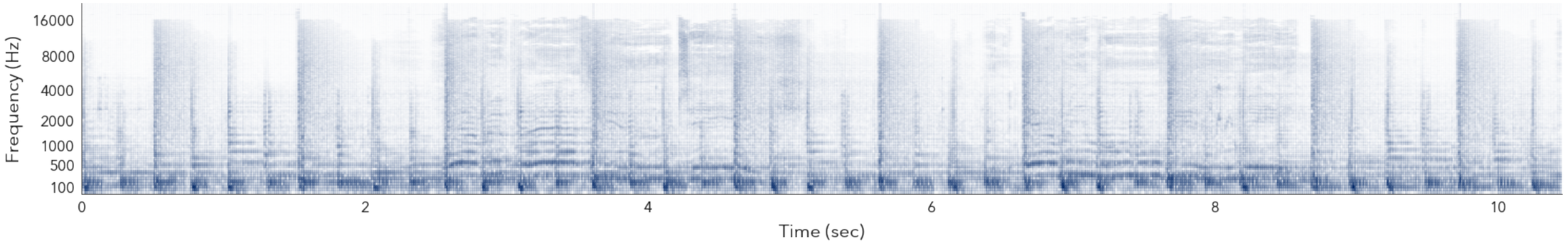


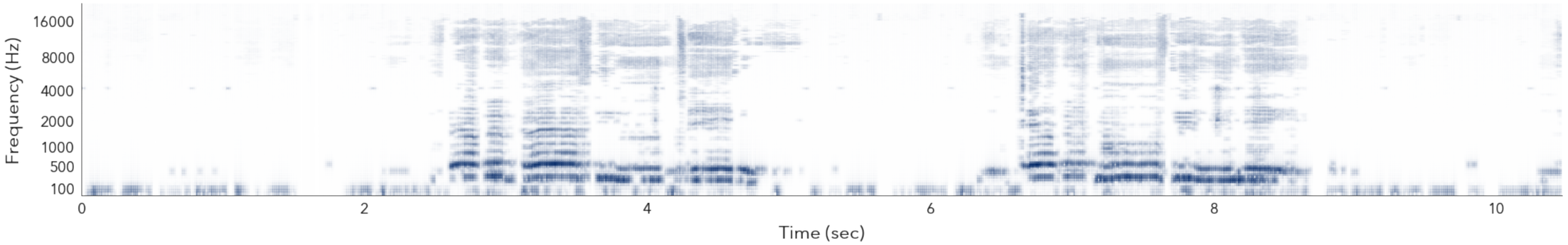
Same model as before

- Known factors will describe the target best
  - Similar spectra and activations
    - We can additionally fine-tune them to fit the input best
  - Using that factor subset we approximate the target

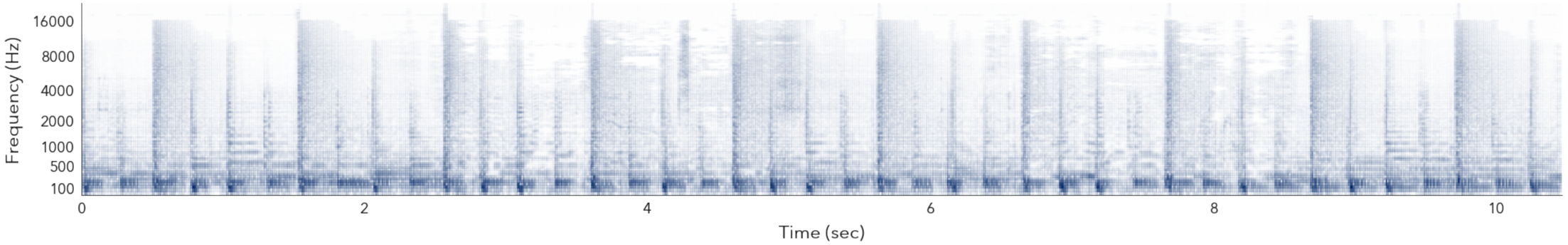
- Extra factors will explain everything else
  - Which we can use to resynthesize the rest of the mix

Extracted sources





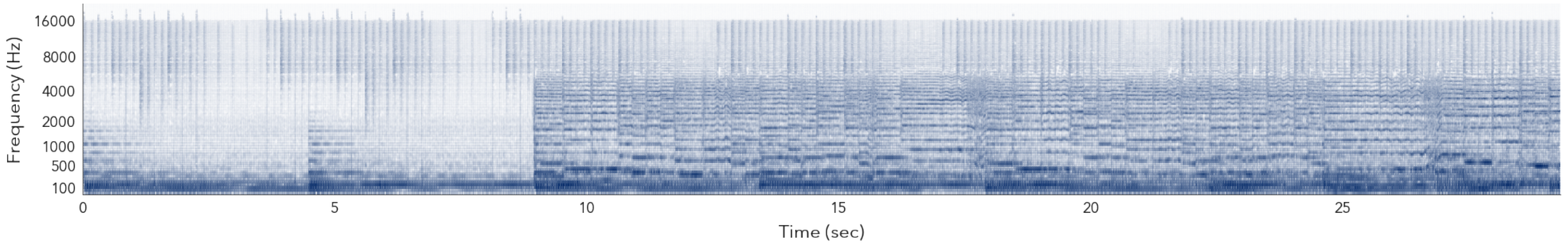
Vocals



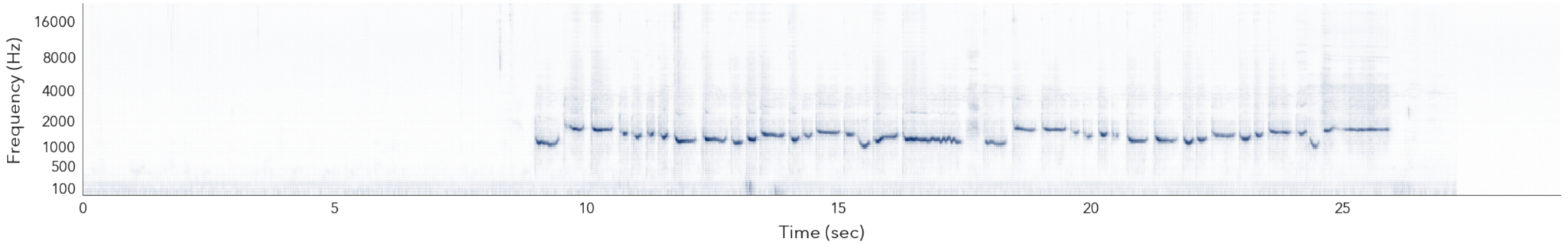
Music

Non-matching spectra

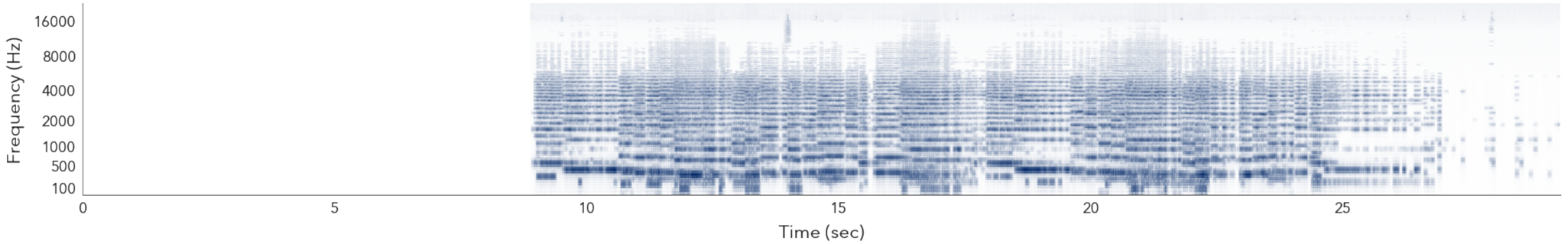
- No need to be too exact



Input mix



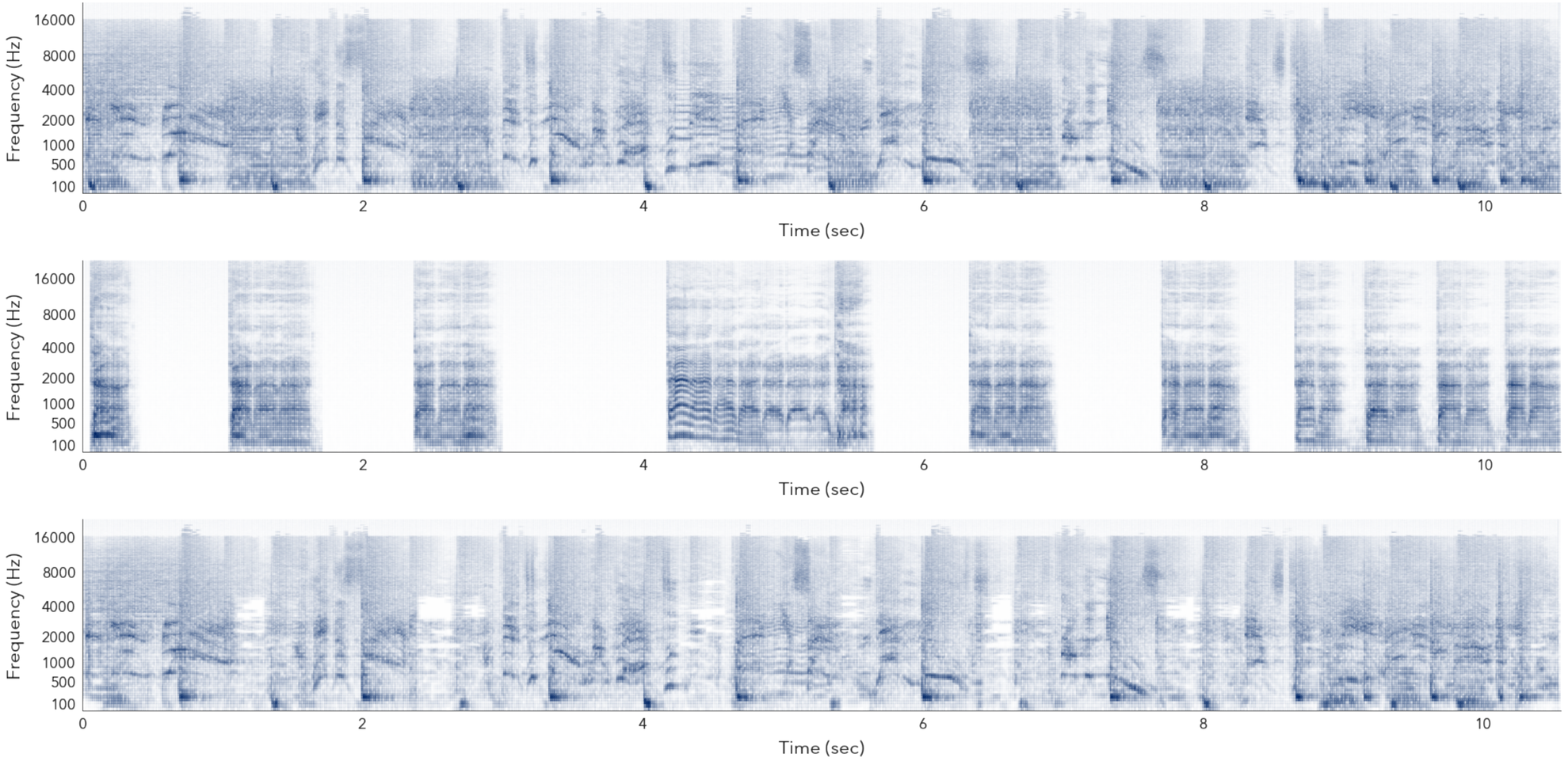
User input



Extraction

Pushing it a bit …

- Very rough approximation to input



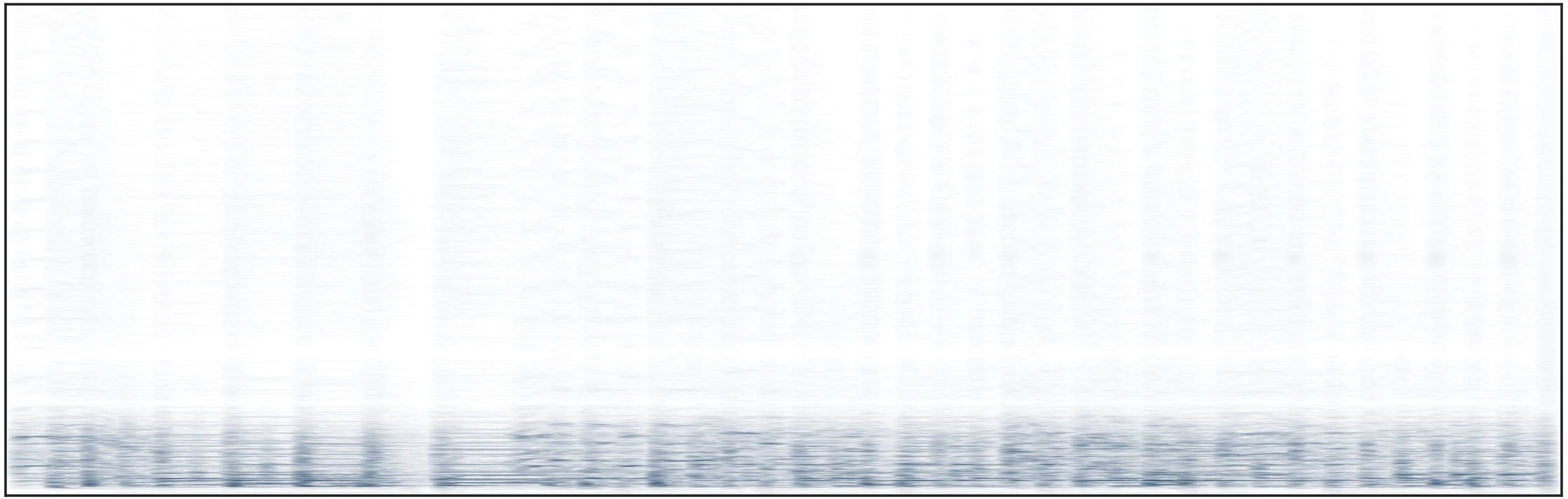
Inputs

Removed guitars

Audio superresolution

- Suppose that we are missing some frequencies
  - Can we ever recover them?

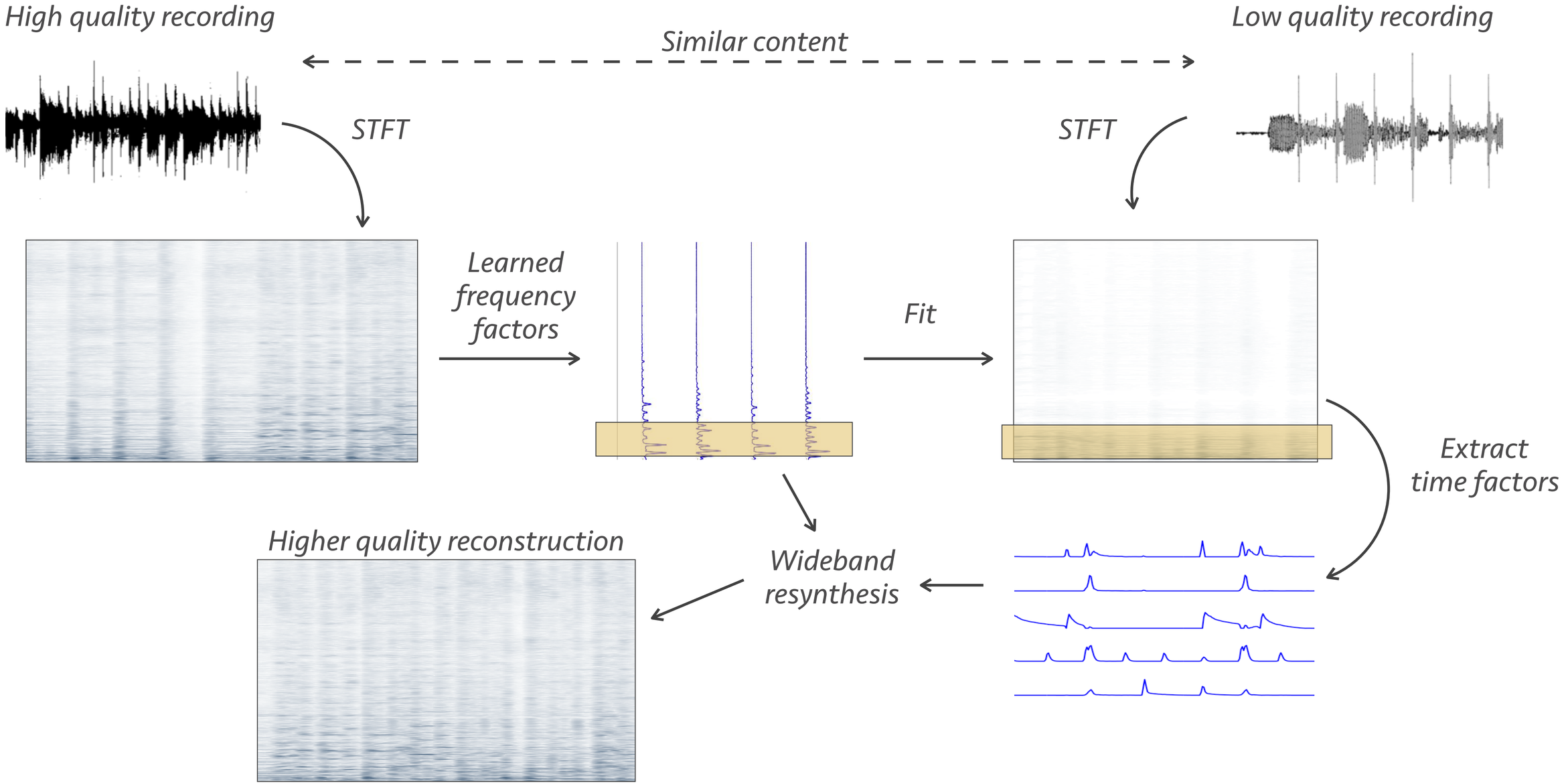
A bandlimited recording



Recovery by example

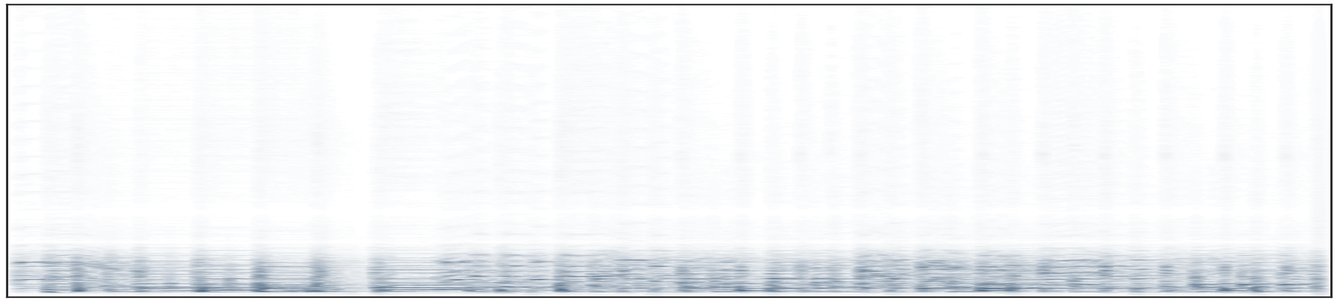
- We need a way to “make up” the missing data
  - We can use hints from other recordings
- Extract model of a sound that serves as example
  - Use that model to reconstruct low-quality recording

Overview of the process

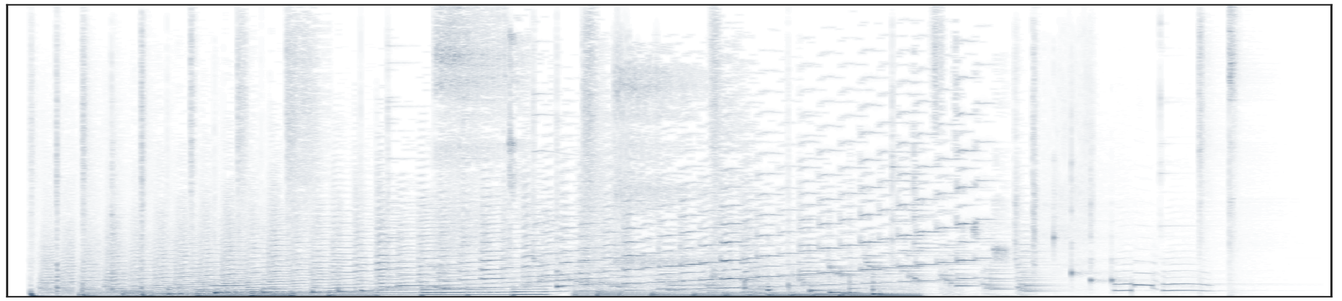


Expansion example

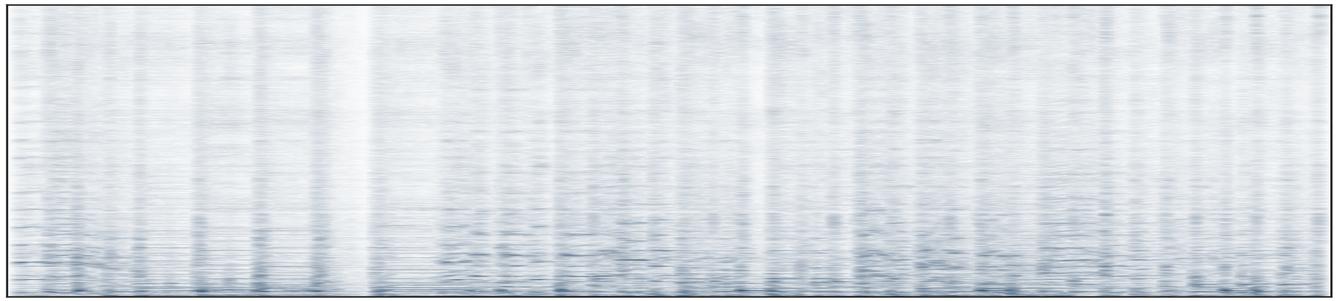
Low quality recording (400Hz-1600Hz)



High quality example

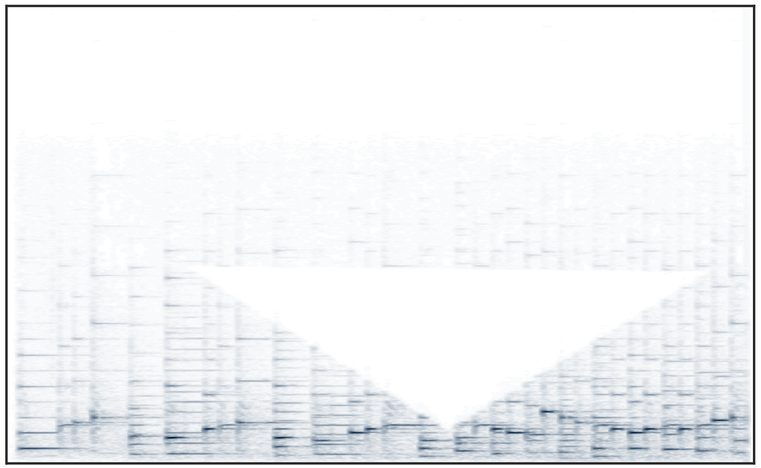


Expanded reconstruction

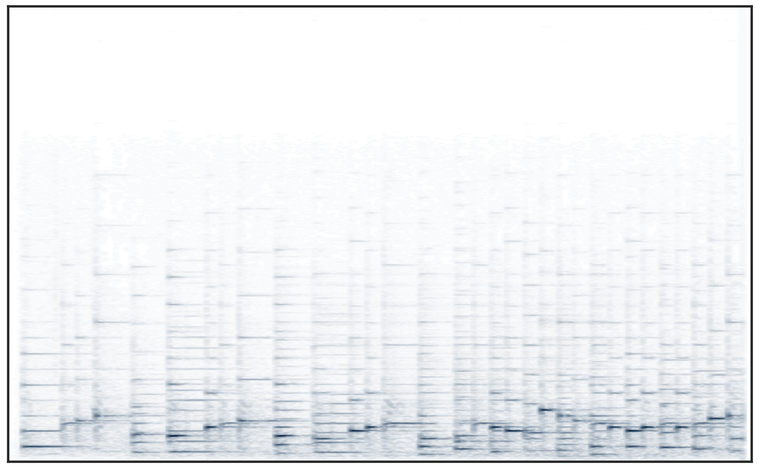


Also works for missing data

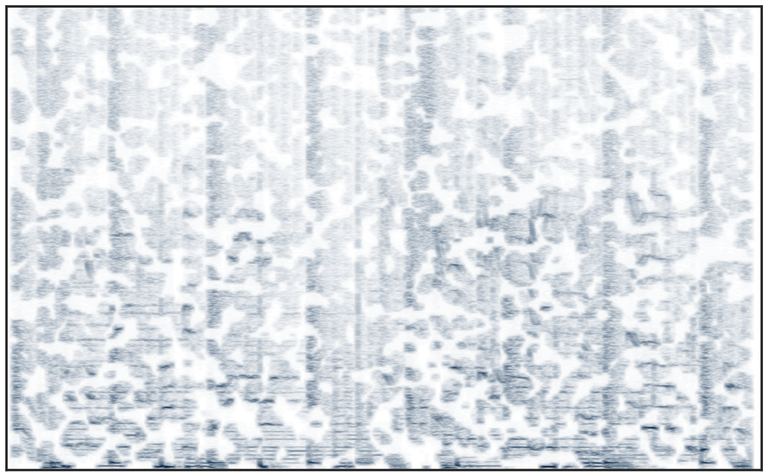
Input with missing data



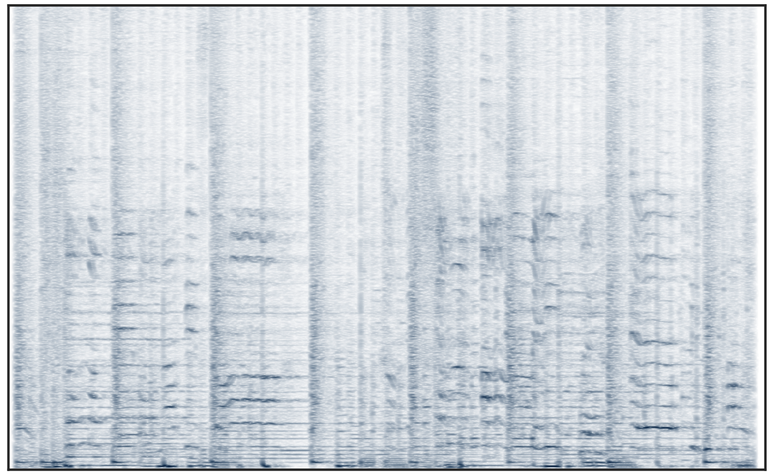
Recovered output



Extreme compression input



Recovered output



Many more applications

- Mixtures are very common in the real-world
  - Medical measurements, mechanical readings, …
- I only talked about audio here
  - Lots of work to do in the field of mixtures
  - Lots of cool things coming out recently

Recap

- Source separation
  - Different approaches
- Spectral factorizations
  - Non-negative models
- Example-driven processing
  - Source selection, reconstruction, missing data, …

Reading material

- Spectral factorizations:
  - http://paris.cs.illinois.edu/pubs/T-SA-00816-2005.pdf
- Bandwidth expansion:
  - http://paris.cs.illinois.edu/pubs/smaragdis-waspaa07.pdf
- User-guided separation:
  - http://paris.cs.illinois.edu/pubs/smaragdis-waspaa09.pdf

This week's lab

- Lab on spectral factorizations
  - Basic supervised source separation
- Also the last lab for this course! 🎉

Last bit of administrivia

- No classes next week
  - I will be out of town April 16-22