Artificial Intelligence
EECS 491

Non-Gaussian latent variable models: ICA and blind source separation

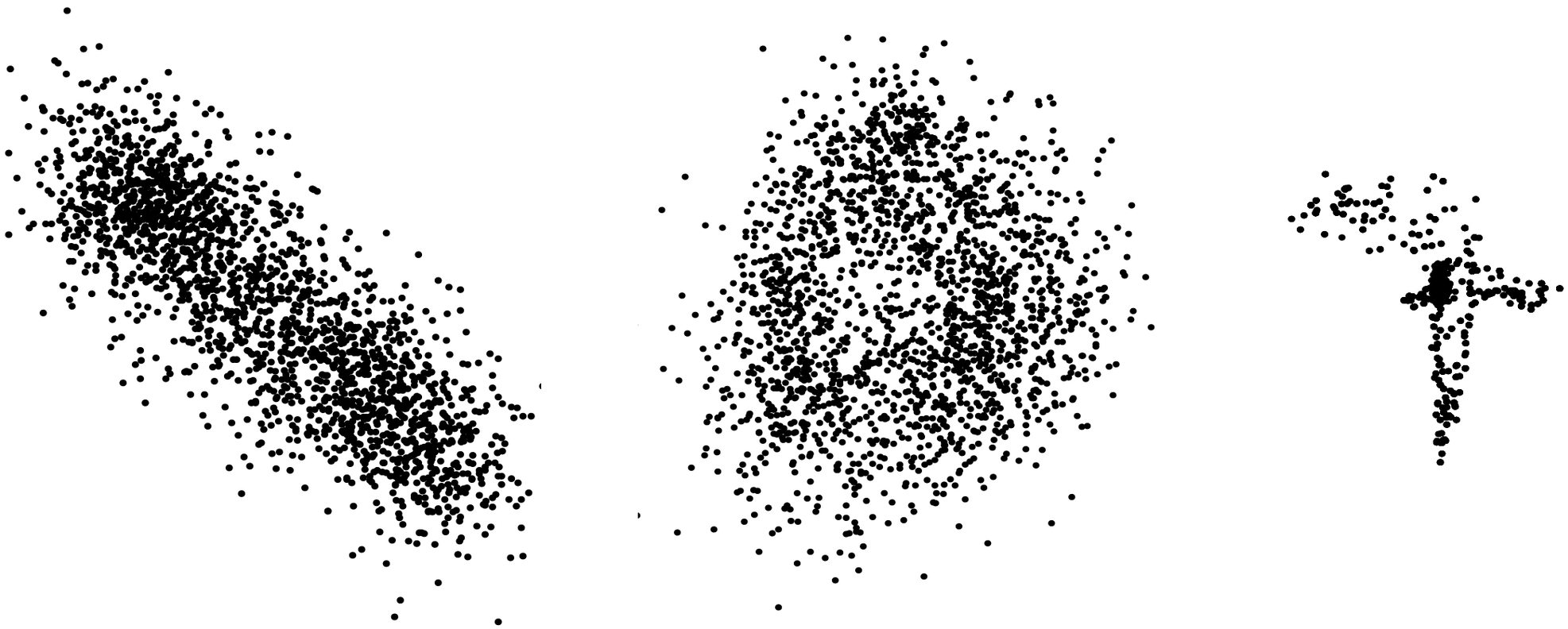
Previous lecture

- Covariance matrices and principal component analysis
- The process of probabilistic inference
  1. *define* model of problem
  2. *derive* posterior distributions and estimators
  3. *estimate* parameters from data
  4. *evaluate* model accuracy
- Applying Bayes rule to probability density functions

$$\underbrace{p(\theta|y,n)}_{\text{posterior}} = \frac{\overbrace{p(y|\theta,n)}^{\text{likelihood}}\overbrace{p(\theta|n)}^{\text{prior}}}{\underbrace{p(y|n)}_{\text{normalizing constant}} = \int p(y|\theta,n)p(\theta|n)d\theta}$$

Density estimation

- Density estimation is the general problem of fitting probabilistic models to data.
- In the coin toss example
  - there was only a single parameter, θ the probability of a "head"
  - the data consisted of the number of trials, and the number of heads.
- The procedure for more complex models is the same



Applications of density estimation

- unsupervised learning
  - clustering
  - language modeling
- regression (underlying deterministic model + random variation)
  - prediction, interpolation, extrapolation
  - time series modeling (e.g. stocks, climate)
- anomaly detection
- undirected learning of kinematic control
- texture modeling and synthesis
- signal denoising
- signal compression and coding

In each of these examples, the objective is to infer the underlying structure in the data from incomplete or noisy examples.

Multivariate Gaussians

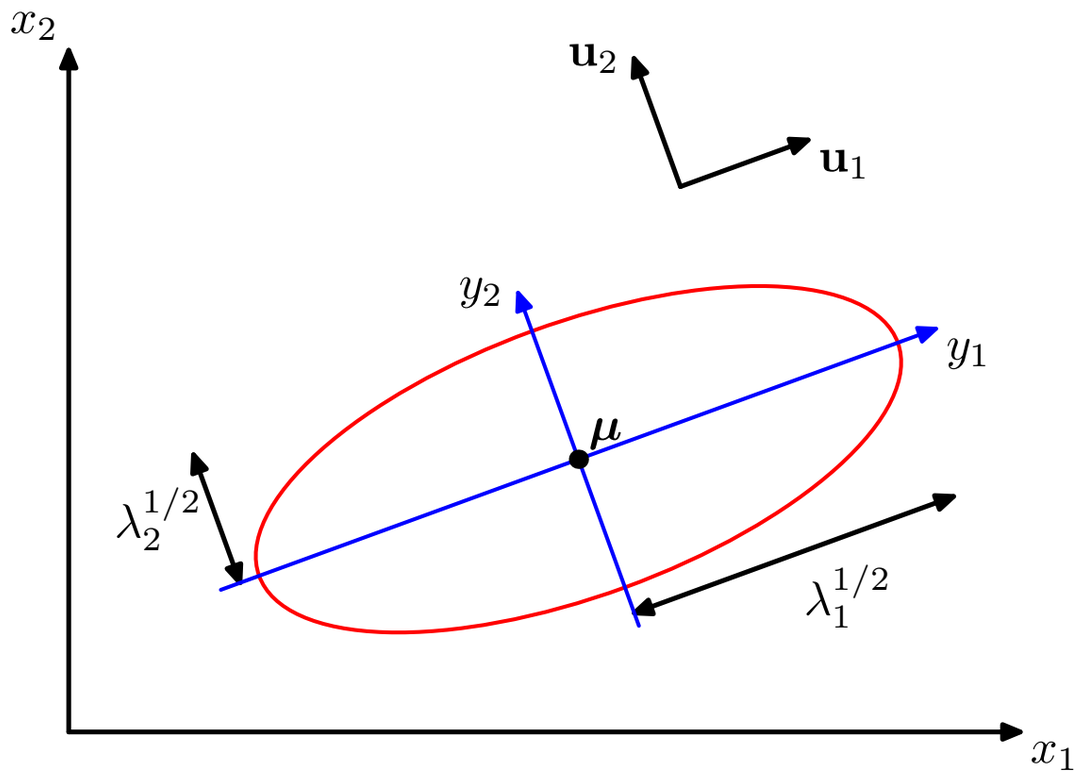
- Recall the univariate Gaussian

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

- The multivariate Gaussian is defined as follows:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Recall: This can be interpreted as transforming into a new coordinate system defined by the *eigenvectors*.



Recall example from previous lecture: waveform data

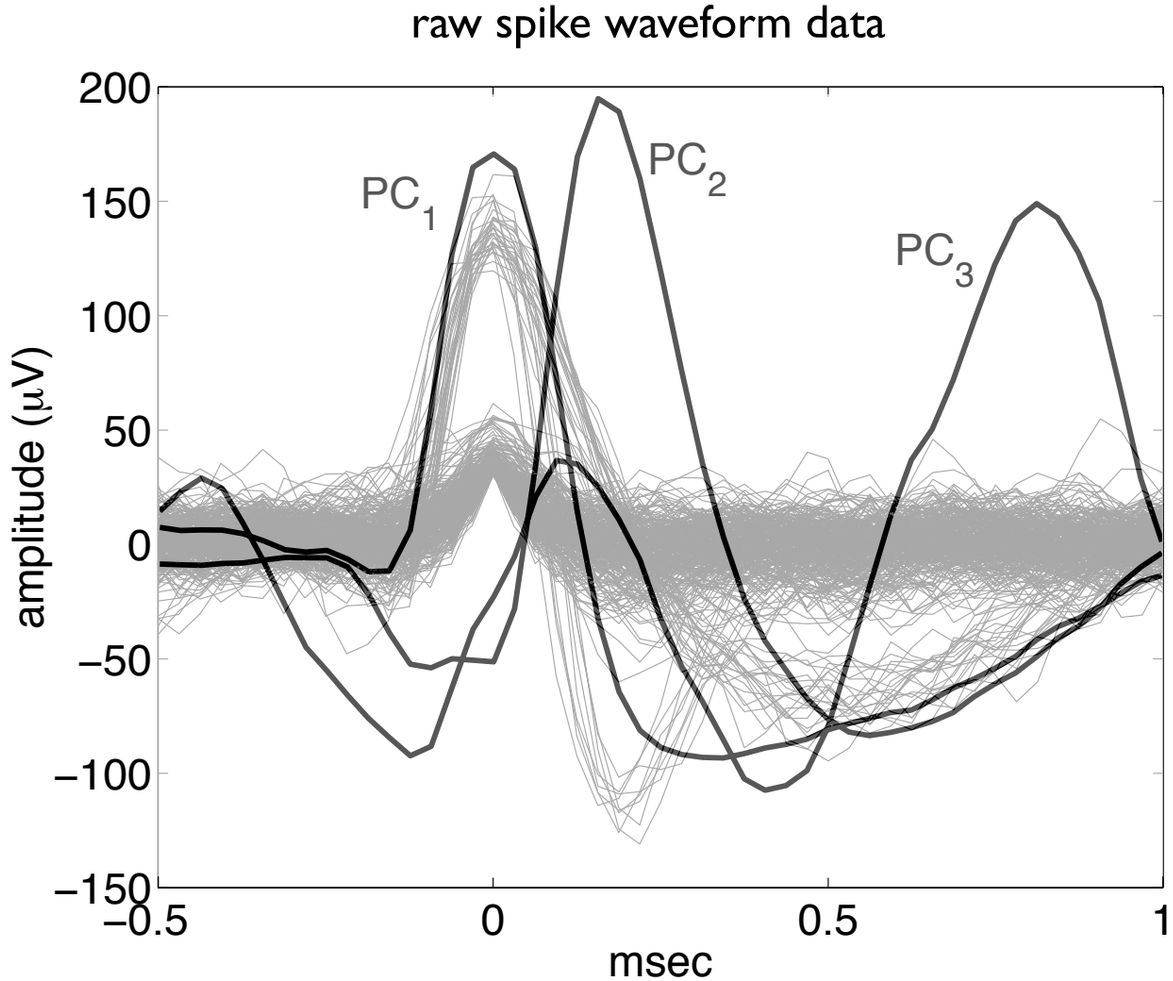
- The waveform x is modeled with a multivariate Gaussian

$$x \sim p(x|\mu, \Sigma)$$

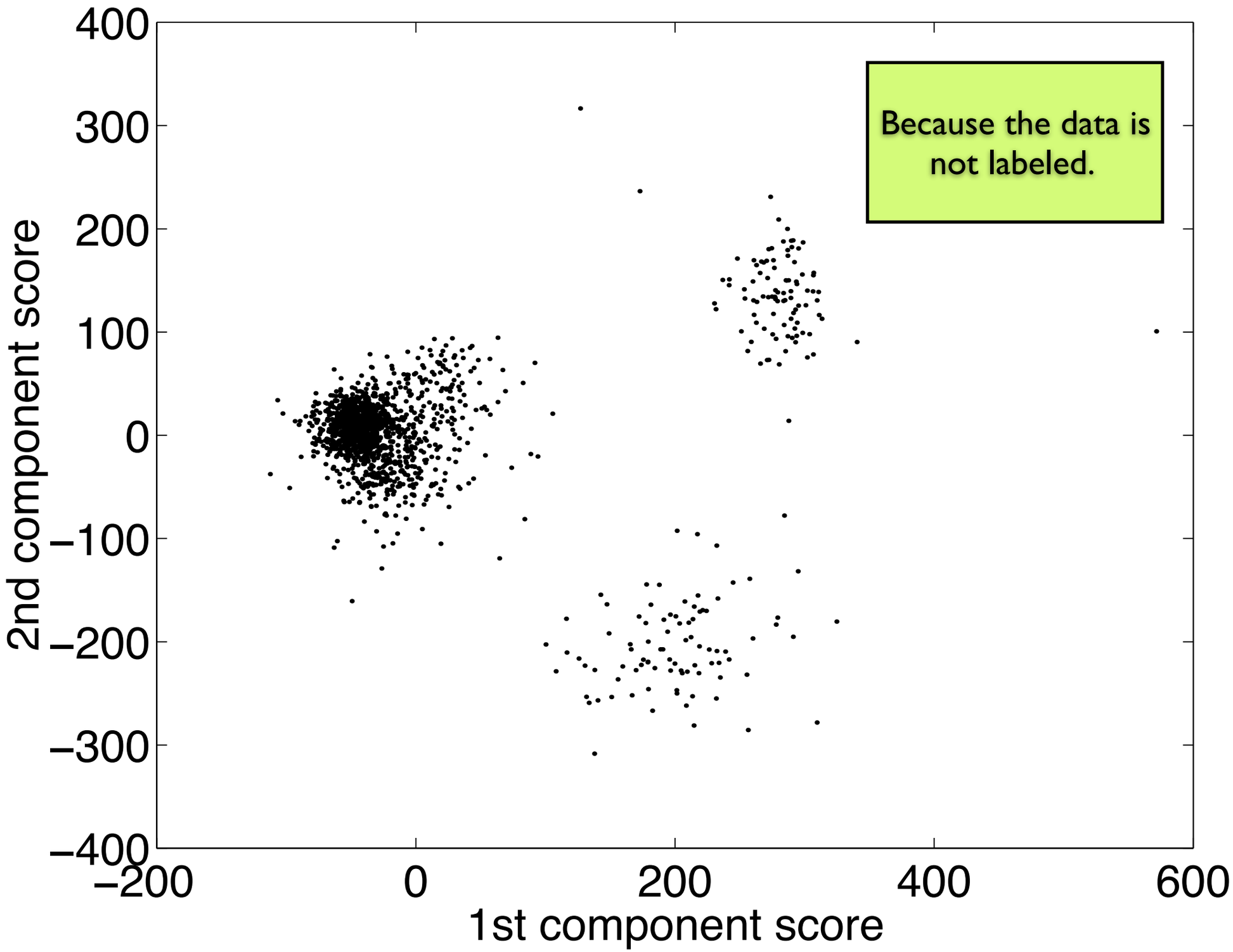
- μ and Σ are mean and covariance of the distribution.
- Principal components can be used to form a low-dimensional approximation

$$x^{(n)} = \sum_{i=1}^{T} c_i^{(n)} \phi_i$$

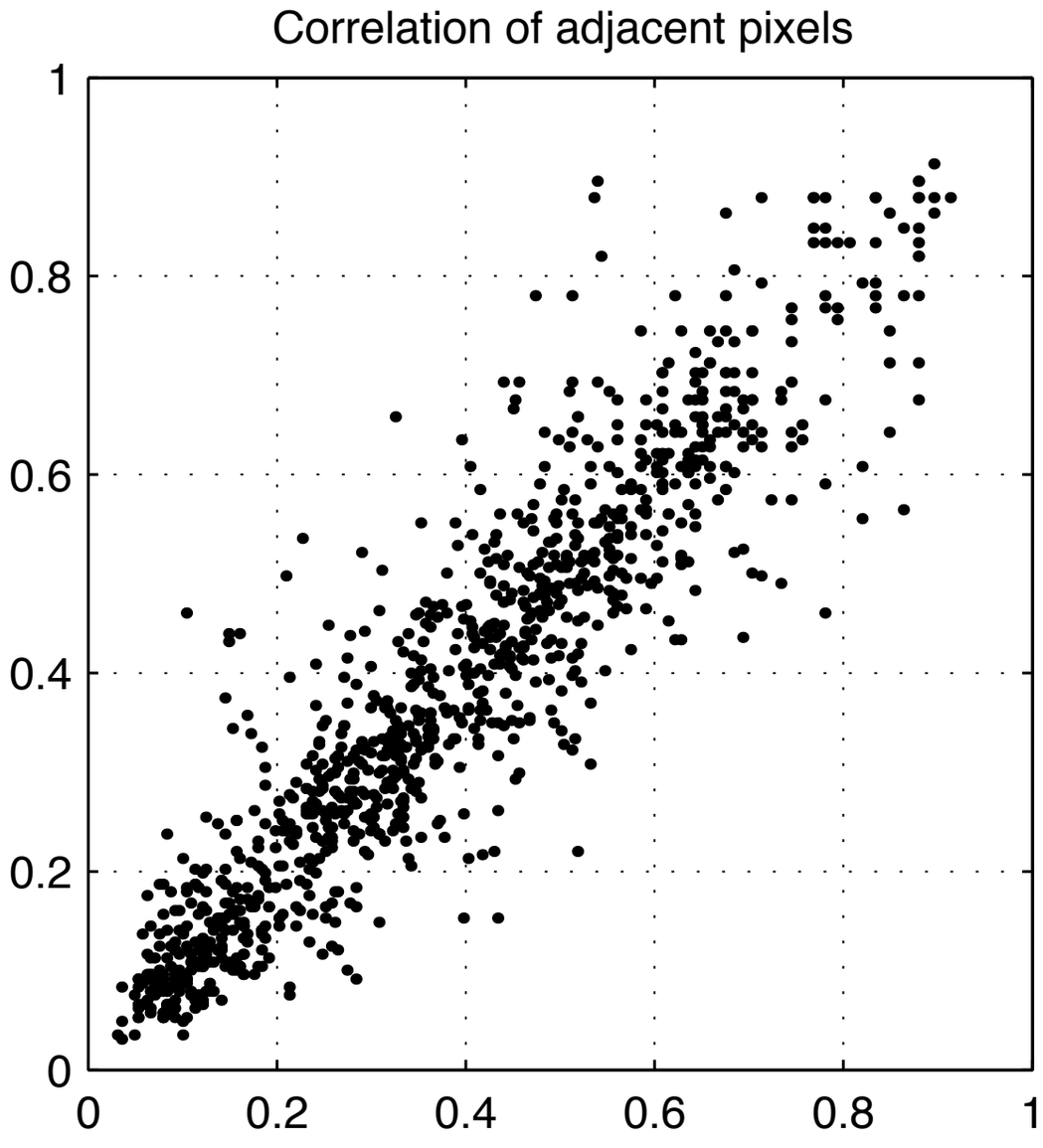
- The vectors $\{\phi_1, \ldots, \phi_T\}$ are the eigenvectors of Σ.
- keeping on the first two terms in the sum is an adequate approximation of the full T-dimensional density.



Why do we have to do density estimation for this problem?



Learning densities of images

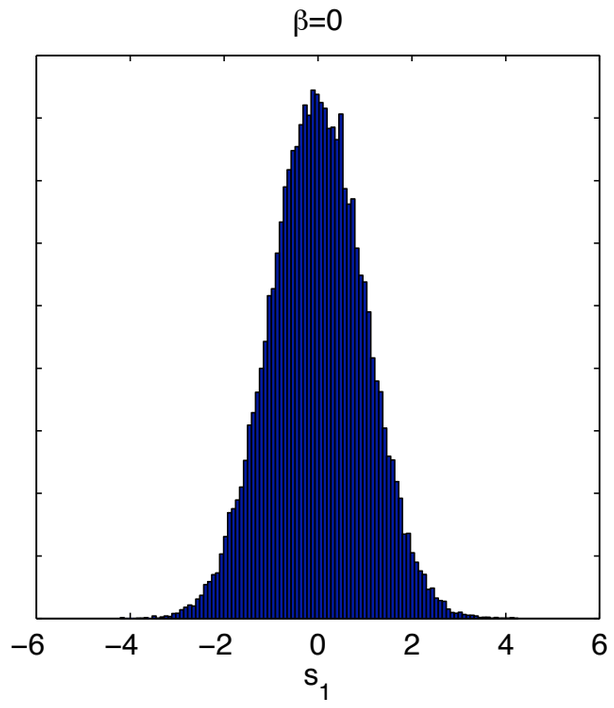


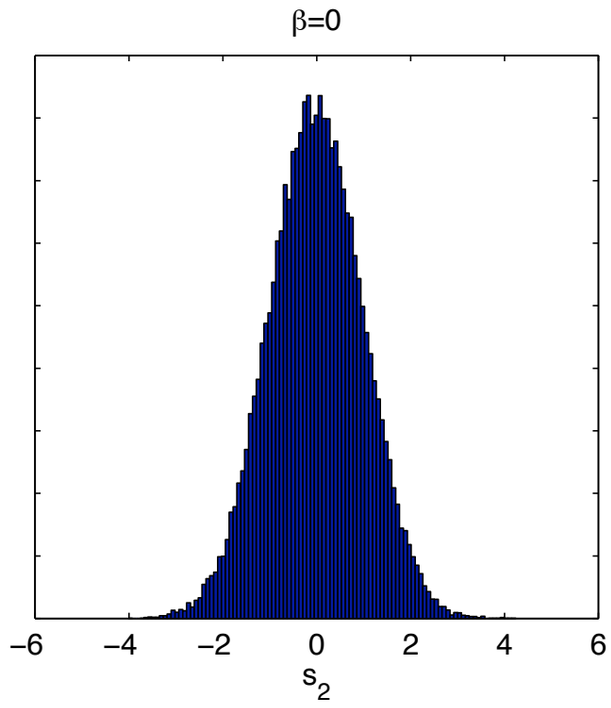
Idea:

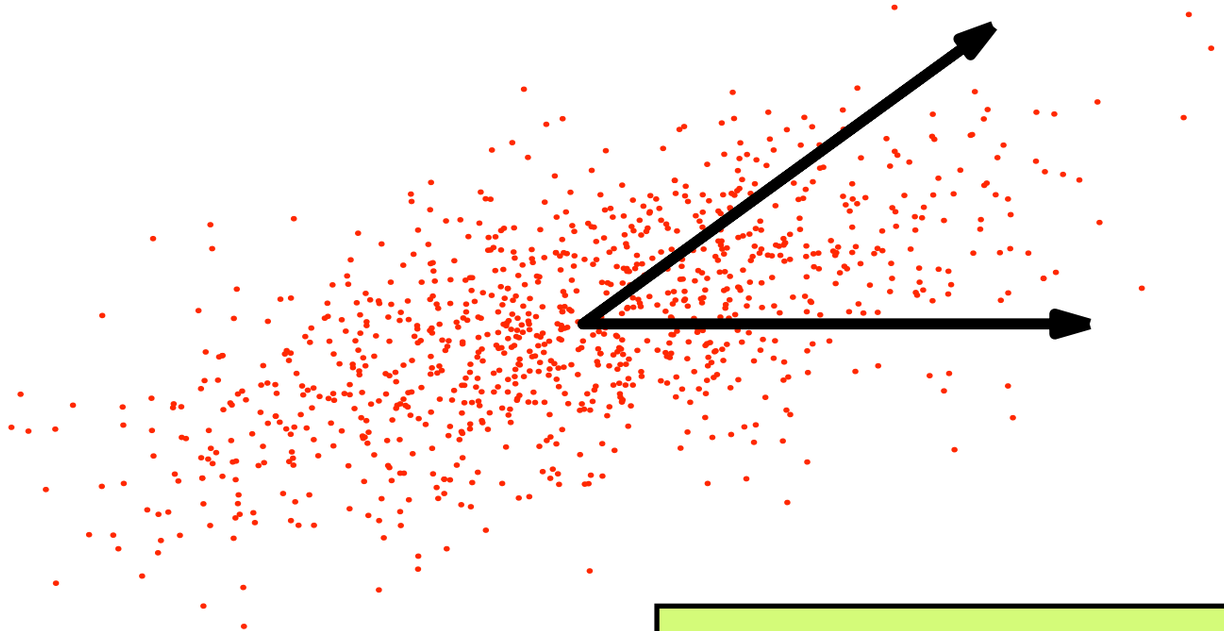
Good codes capture the statistical distribution of sensory patterns

greater coding efficiency ⇔ more learned structure

A distribution of two Gaussian sources





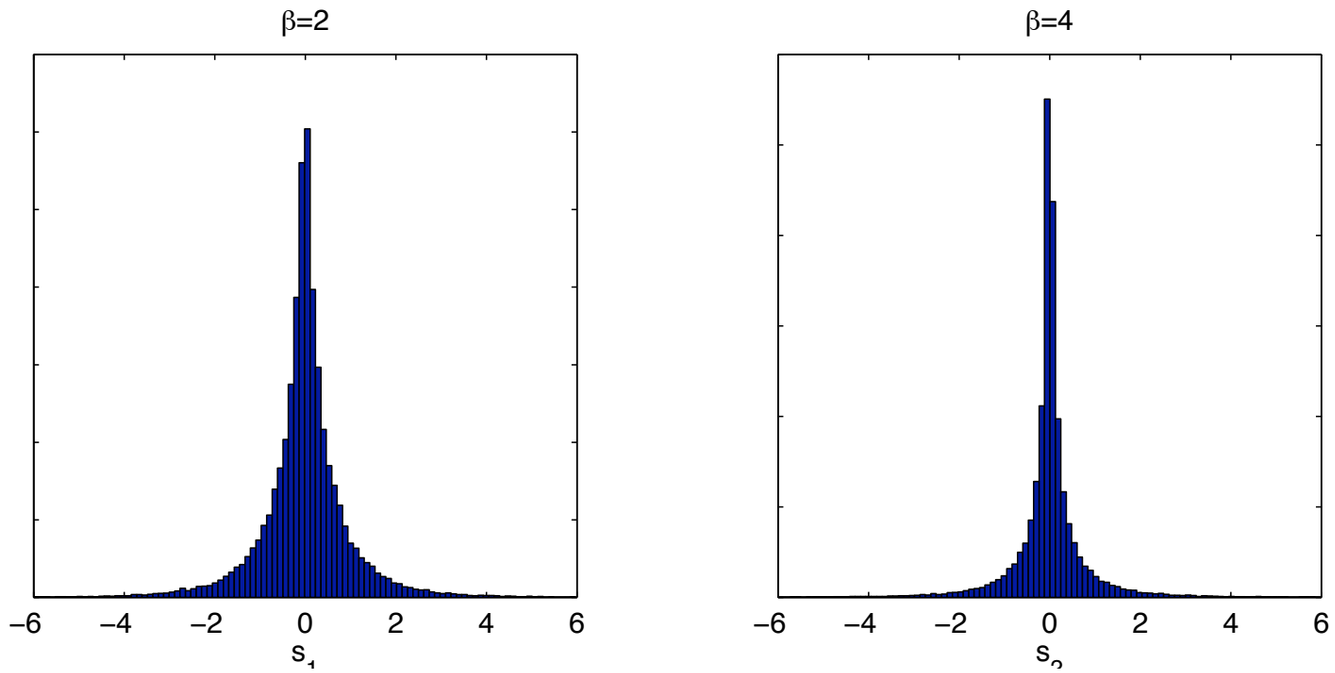


There's just one problem:
Image densities aren't Gaussian!

- The histograms show the amplitude distribution of each source.
- The scatter plot shows the 2D distribution of $s_1(t)$ vs $s_2(t)$.
- What are the principal components?
- The vectors form a *basis* for the distribution of data:

  any point in the set can be represented by linear combinations of these vectors

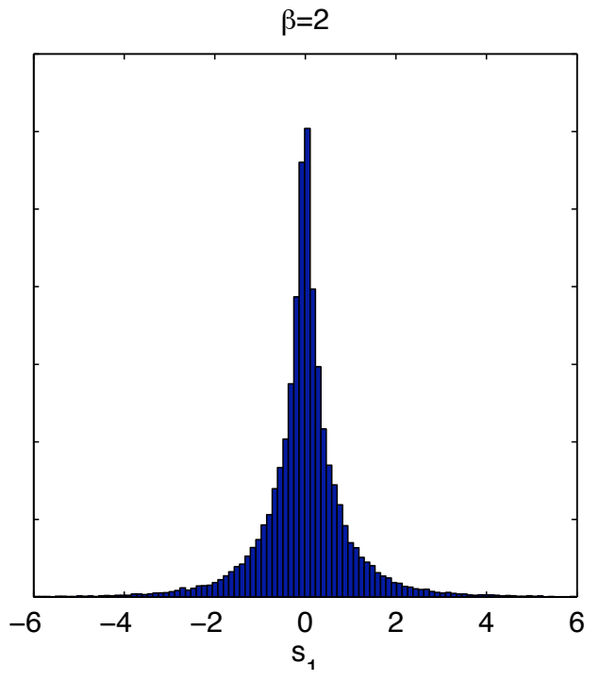
The distribution of sample points for non-Gaussian sources

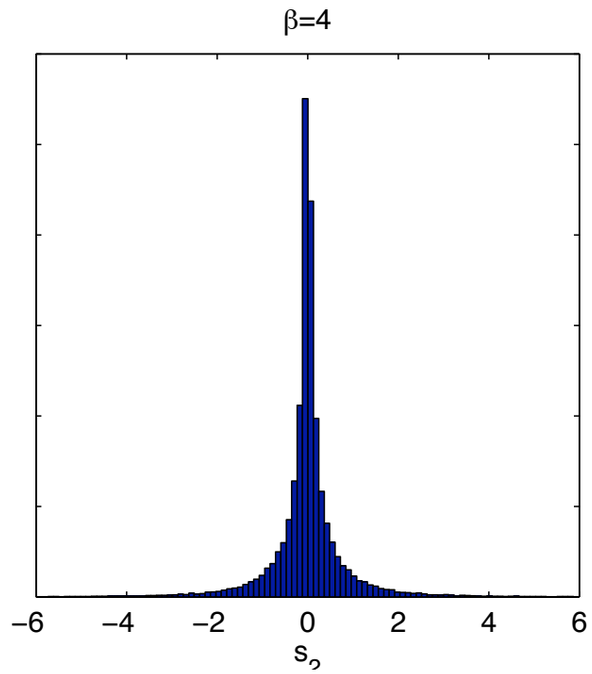


The actual densities look more like this.

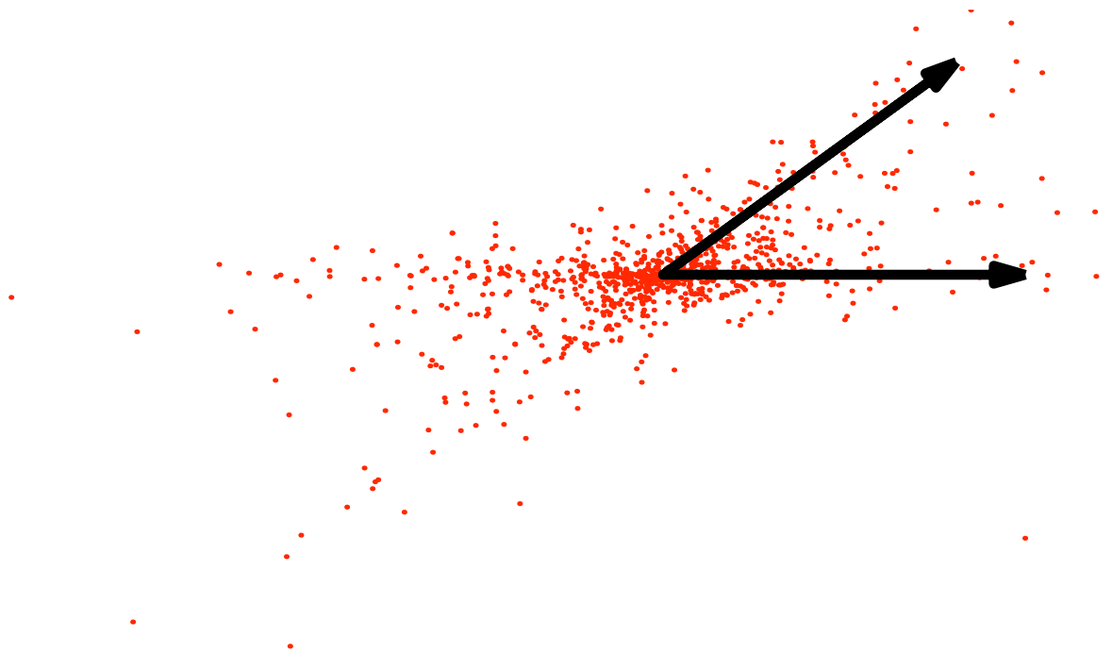
What will the joint distribution look like now?

The distribution of sample points for non-Gaussian sources

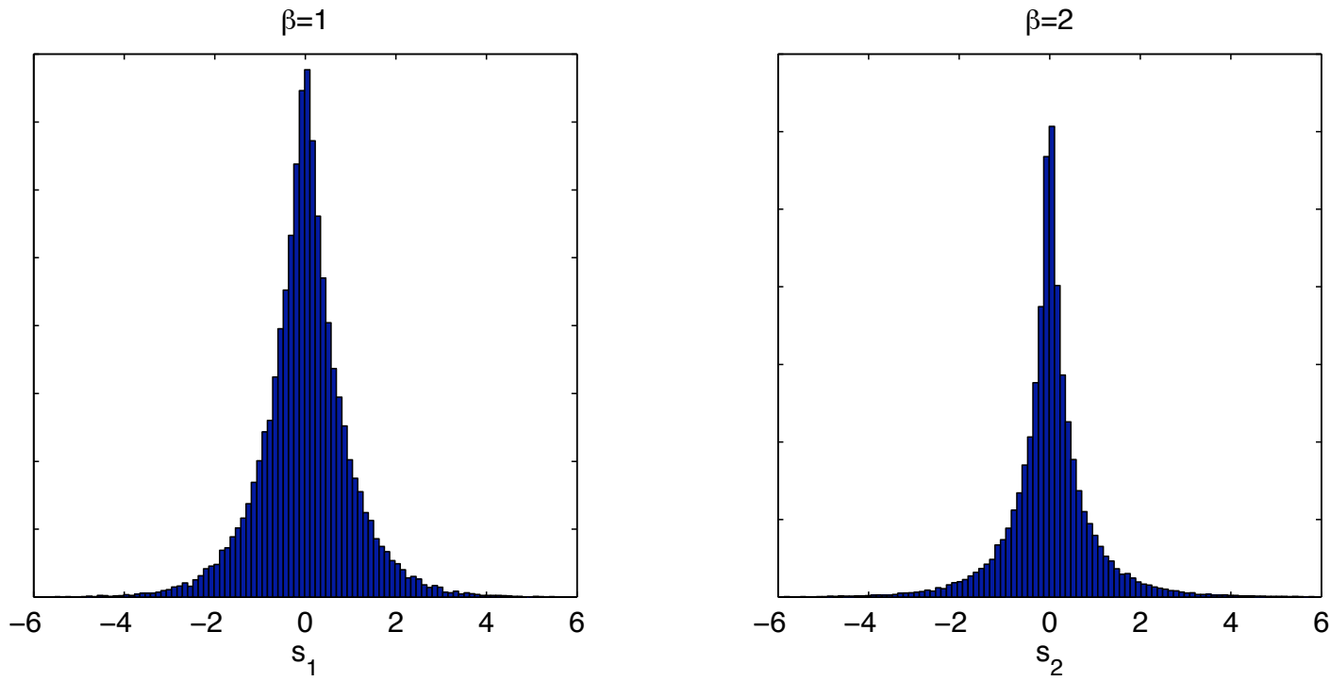




What will the joint distribution look like now?

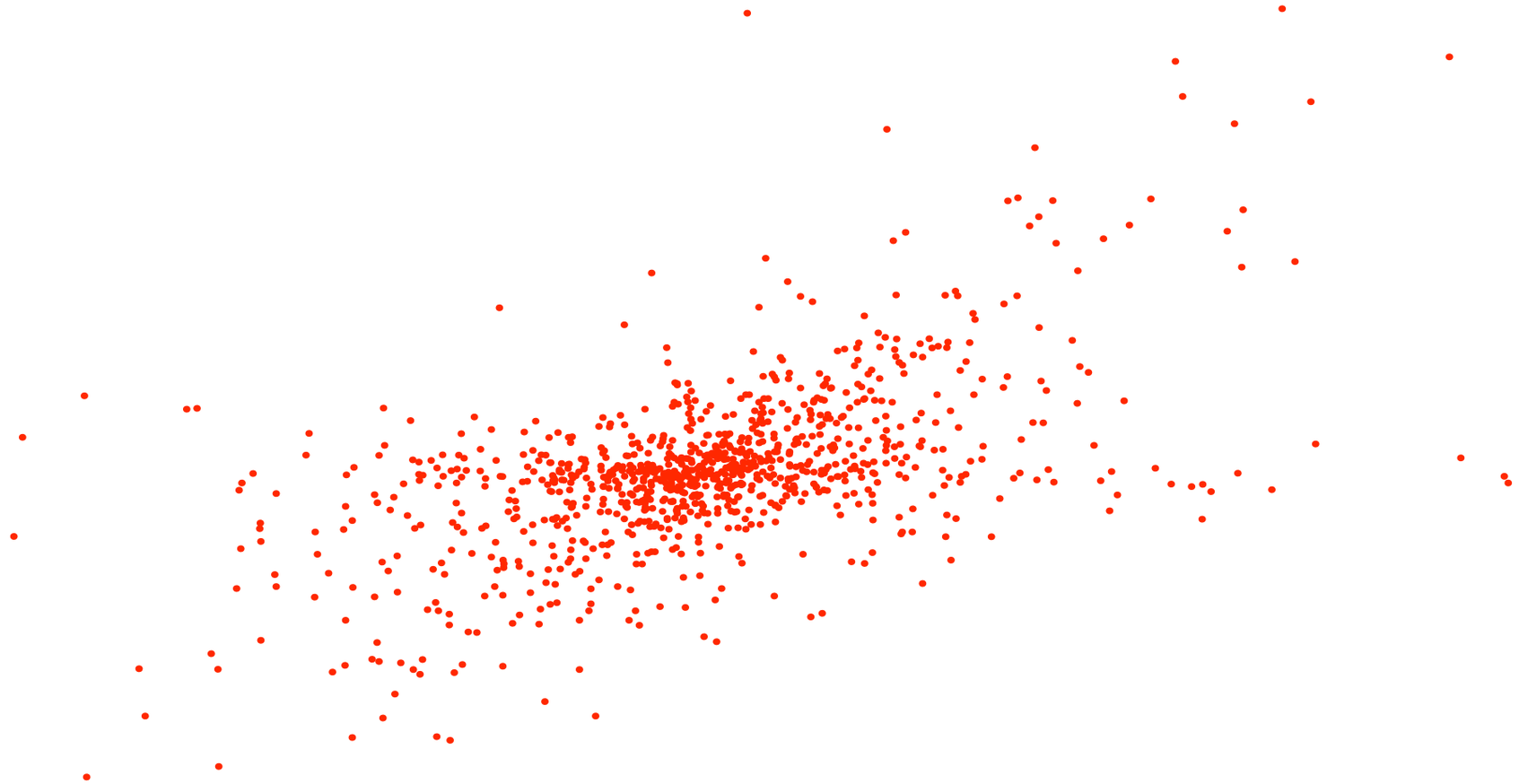


The distribution of sample points



- Sources can have a wide range of amplitude distributions.
- Different directions correspond to different mixing matrices.
- A mixture of non-Gaussian sources create a distribution whose axes can be uniquely determined (within a sign change).

Checking the model



- How would a Gaussian model fit this distribution?
- What would the principal components be?
- Idea: Let's do better by allowing non-Gaussian models with non-orthogonal basis vectors.

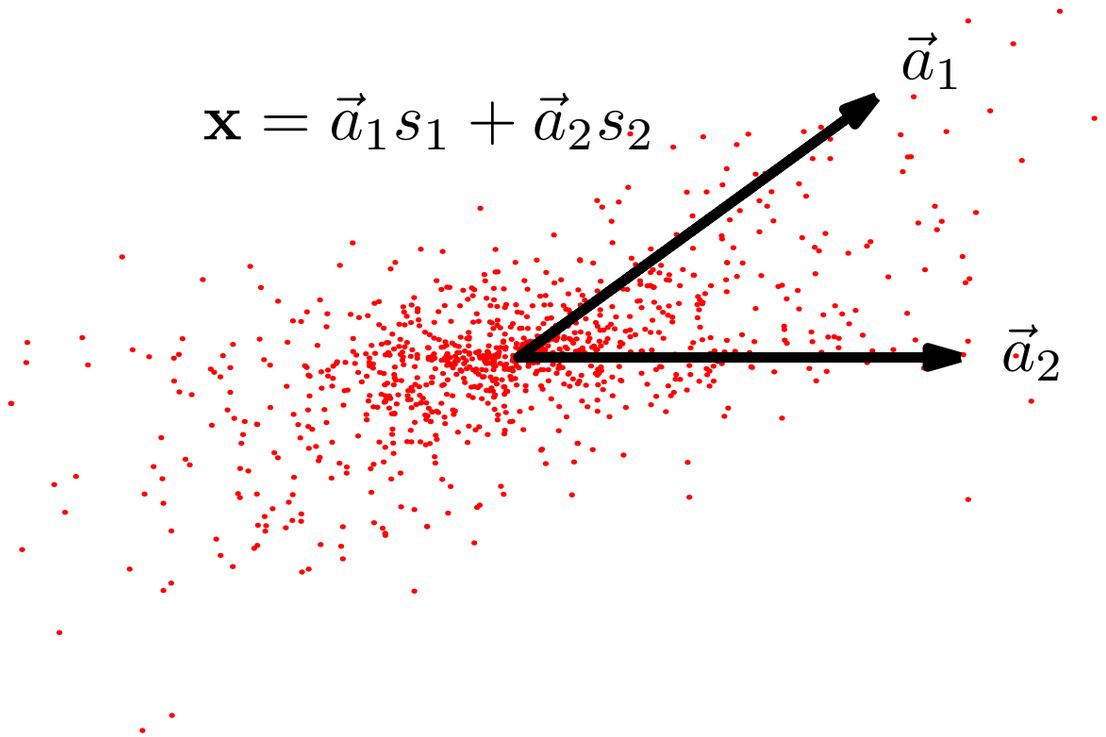
Coding images with a statistical model

Goal: Encode the patters to desired precision:

$$\begin{aligned}\mathbf{x} &= \vec{a}_1 s_1 + \cdots + \vec{a}_L s_L + \vec{\epsilon} \\ &= \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}\end{aligned}$$

Columns of **A** are *basis functions*

Here is a 2D example:



Coding images with a statistical model

Goal: Encode the patters to desired precision:

$$\begin{aligned}\mathbf{x} &= \vec{a}_1 s_1 + \cdots + \vec{a}_L s_L + \vec{\epsilon} \\ &= \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}\end{aligned}$$

Columns of **A** are *basis functions*

Posterior:

$$P(\mathbf{s}|\mathbf{x}, \mathbf{A}) = \frac{P(\mathbf{s})P(\mathbf{x}|\mathbf{s}, \mathbf{A})}{P(\mathbf{x}|\mathbf{A})}$$

Prior: s_i's are *independent* and *sparse*:

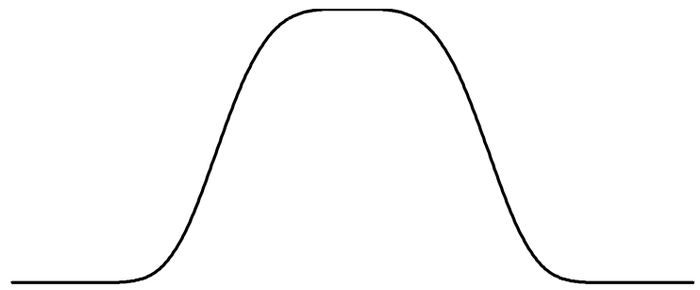
$$P(\mathbf{s}) = \prod_i P(s_i)$$

$$P(s_i) \propto \exp\left[-\left|\frac{s_i}{\lambda_i}\right|^{q_i}\right] \Leftarrow$$

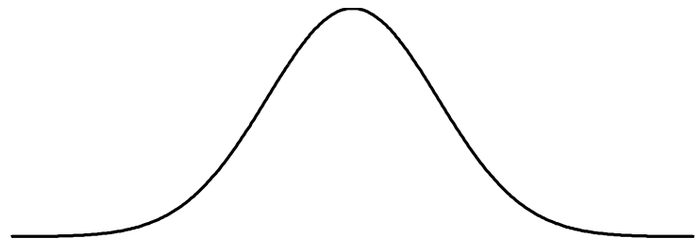
This is a *generalized* Gaussian distribution

The pdfs look like this:

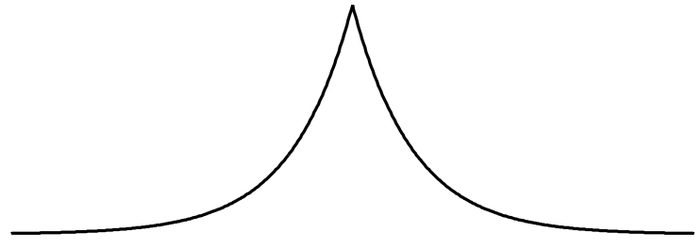
q=4



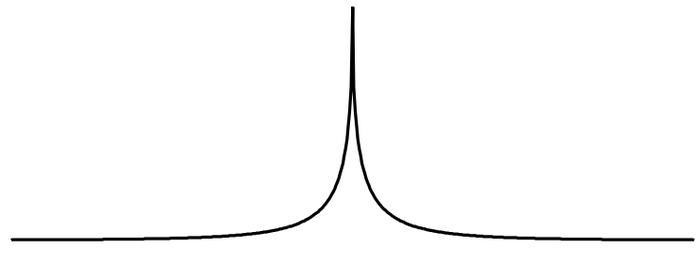
q=2 (Gaussian)



q=1 (Laplacian)



q=0.5



Coding images with a statistical model

Goal: Encode the patters to desired precision:

$$\begin{aligned}\mathbf{x} &= \vec{a}_1 s_1 + \cdots + \vec{a}_L s_L + \vec{\epsilon} \\ &= \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}\end{aligned}$$

Posterior:

$$P(\mathbf{s}|\mathbf{x}, \mathbf{A}) = \frac{P(\mathbf{s})P(\mathbf{x}|\mathbf{s}, \mathbf{A})}{P(\mathbf{x}|\mathbf{A})}$$

Prior: s_i's are *independent* and *sparse*:

$$P(\mathbf{s}) = \prod_i P(s_i)$$

$$P(s_i) \propto \exp\left[-\left|\frac{s_i}{\lambda_i}\right|^{q_i}\right]$$

Likelihood: Assume $\boldsymbol{\epsilon} \sim$ Gaussian,

$$P(\mathbf{x}|\mathbf{s}, \Sigma) \propto \exp\left[-\frac{1}{2}\boldsymbol{\epsilon}^T \Sigma^{-1} \boldsymbol{\epsilon}\right]$$

Inference: use the MAP value:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}, \mathbf{A})$$

Simple special case: no noise (ICA)

$$\hat{\mathbf{s}} = \mathbf{A}^{-1}\mathbf{x}$$

Inference (or recognition or coding):

finds most efficient representation of pattern $\mathbf{x}$ *in a given basis* $\mathbf{A}$

Learning: Optimizing the model parameters

Learning objective:

maximize coding efficiency
$\Rightarrow$ maximize $P(\mathbf{x}|\mathbf{A})$ over $\mathbf{A}$.

Probability of pattern ensemble is:

$$P(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N|\mathbf{A}) = \prod_k P(\mathbf{x}_k|\mathbf{A})$$

$P(\mathbf{x}|\mathbf{A})$ is obtained by marginalization:

$$\begin{aligned} P(\mathbf{x}|\mathbf{A}) &= \int d\mathbf{s}\, P(\mathbf{x}|\mathbf{A}, \mathbf{s}) P(\mathbf{s}) \\ &= \frac{P(\mathbf{s})}{|\det \mathbf{A}|} \end{aligned}$$

Use *independent component analysis* (ICA) to learn $\mathbf{A}$:

$$\begin{aligned} \Delta\mathbf{A} \propto &\quad \mathbf{A}\mathbf{A}^T \frac{\partial}{\partial \mathbf{A}} \log P(\mathbf{x}|\mathbf{A}) \\ = &\quad -\mathbf{A}(\mathbf{z}\mathbf{s}^T + \mathbf{I}), \end{aligned}$$

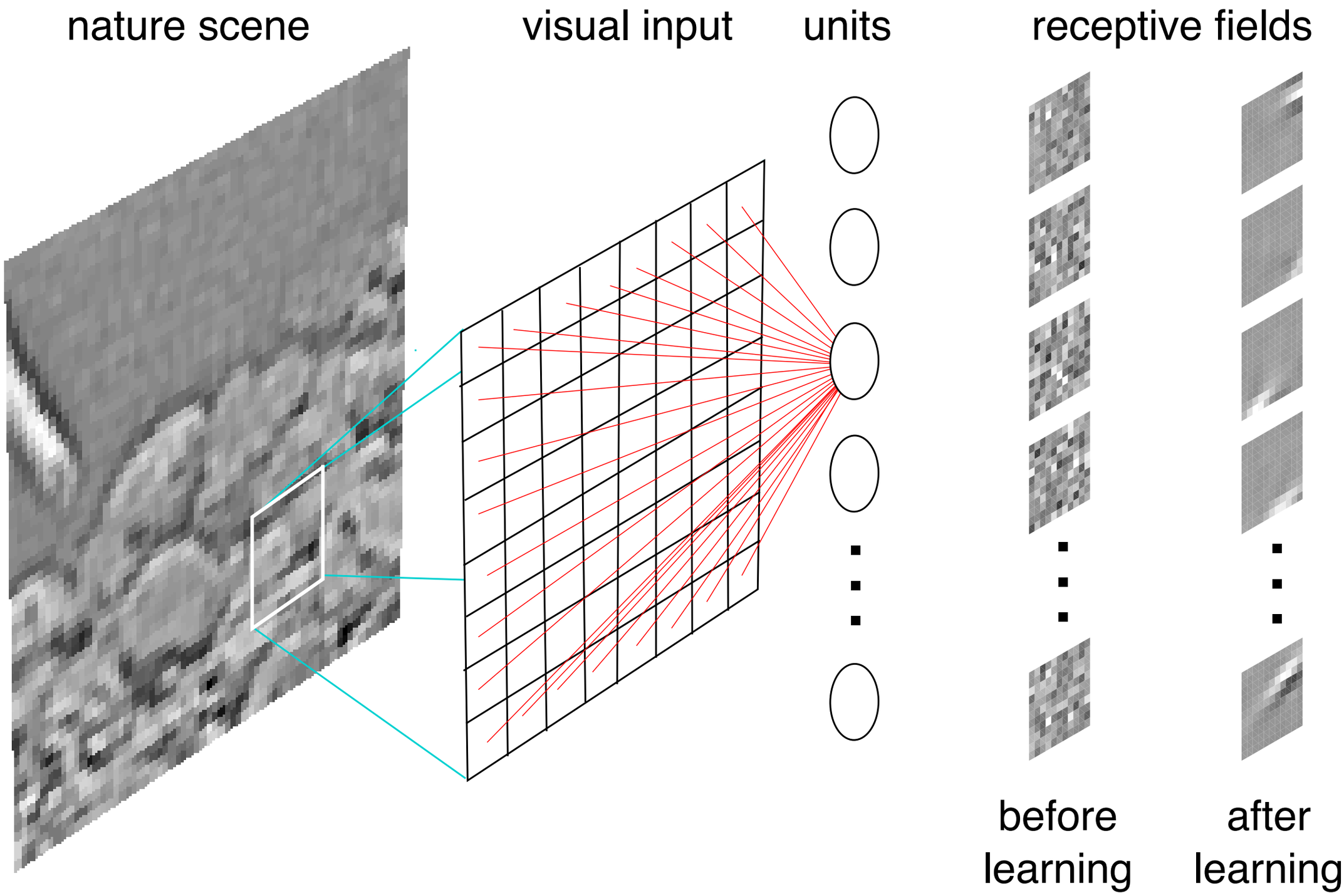
where $z_i = d \log P(s_i)/ds$. Assume generalized Gaussians:

$$P(s_i) \sim \mathcal{N}^{q_i}(s_i|\mu, \sigma).$$

This learning rule:

- *learns the feature set that captures the most structure*
- *optimizes basis to maximize the efficiency of the code*

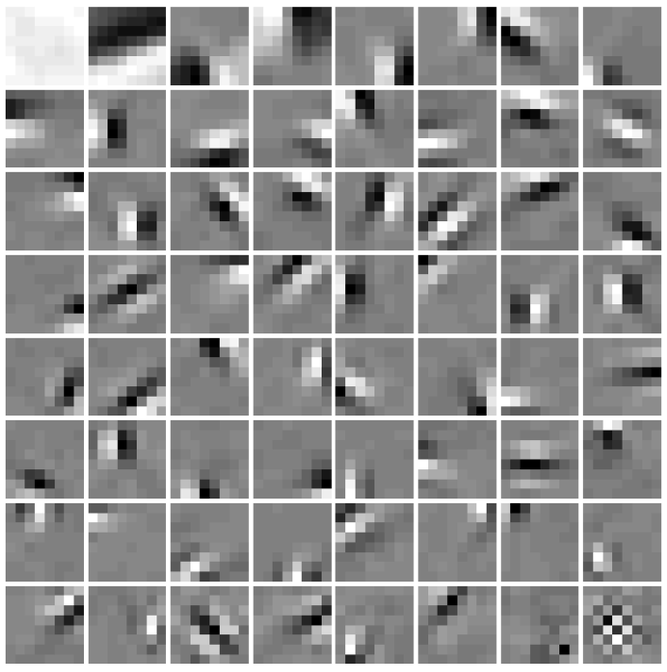
Efficient coding of natural images



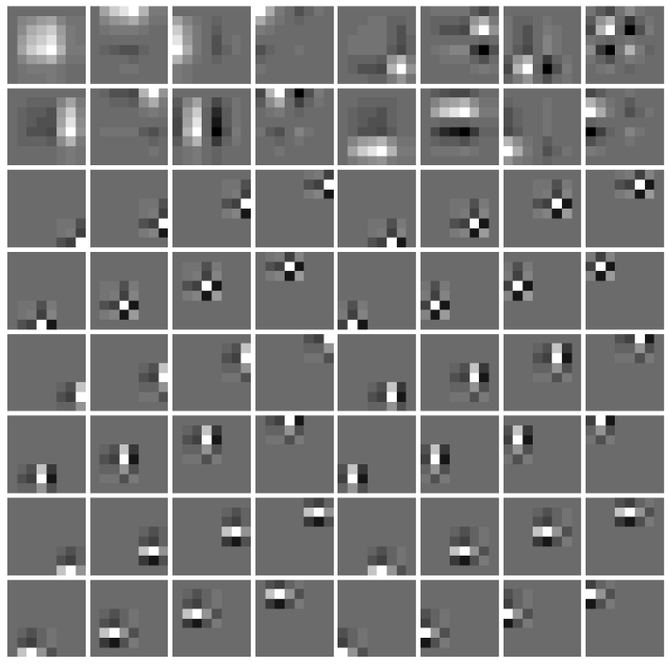
Olshausen and Field, 1996; Bell and Sejnowski, 1997; Lewicki and Olshausen, 1999

Which code is best for natural images?

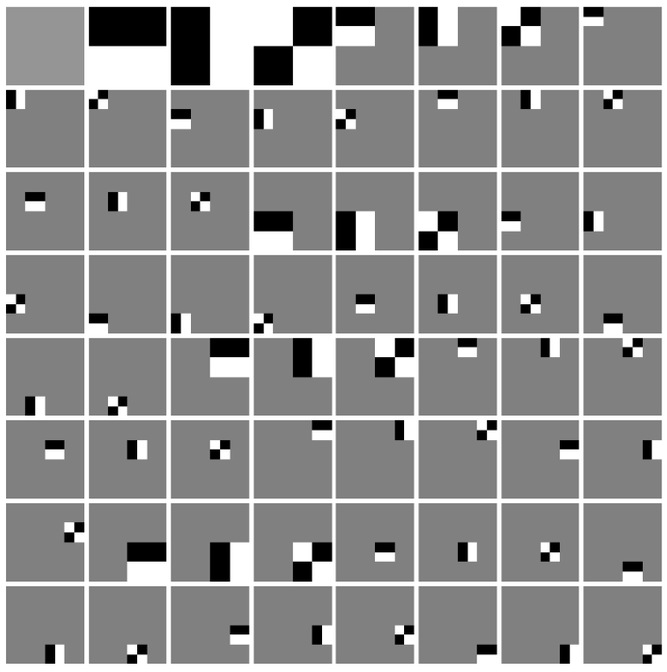
Learned



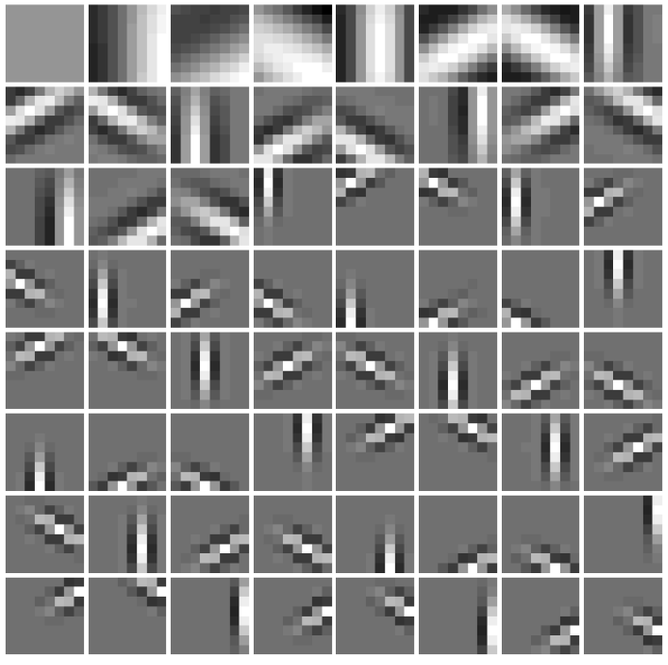
Wavelet



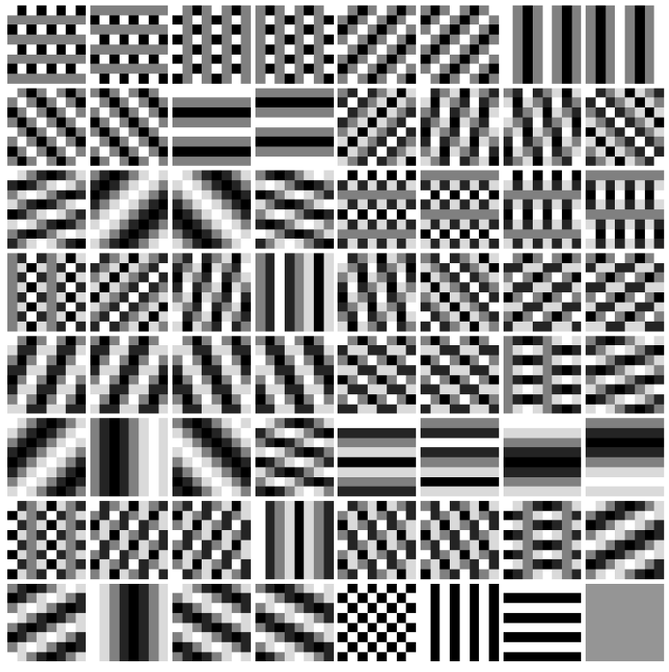
Haar



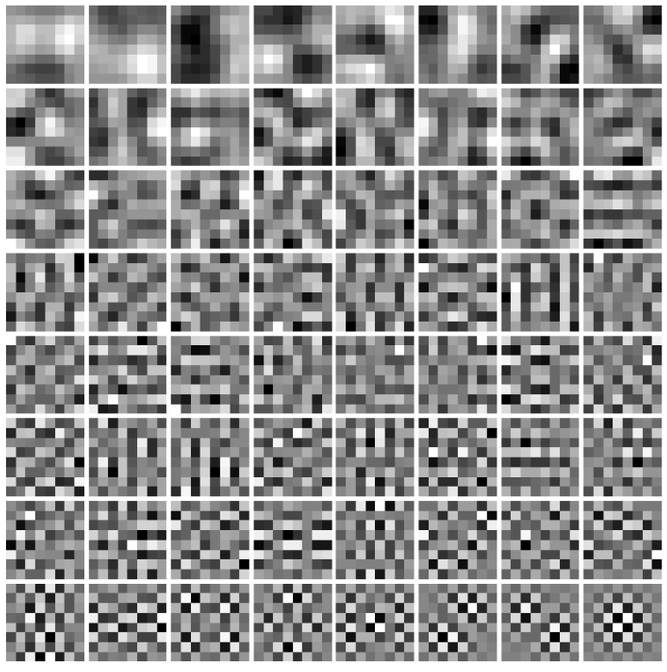
Gabor



Fourier

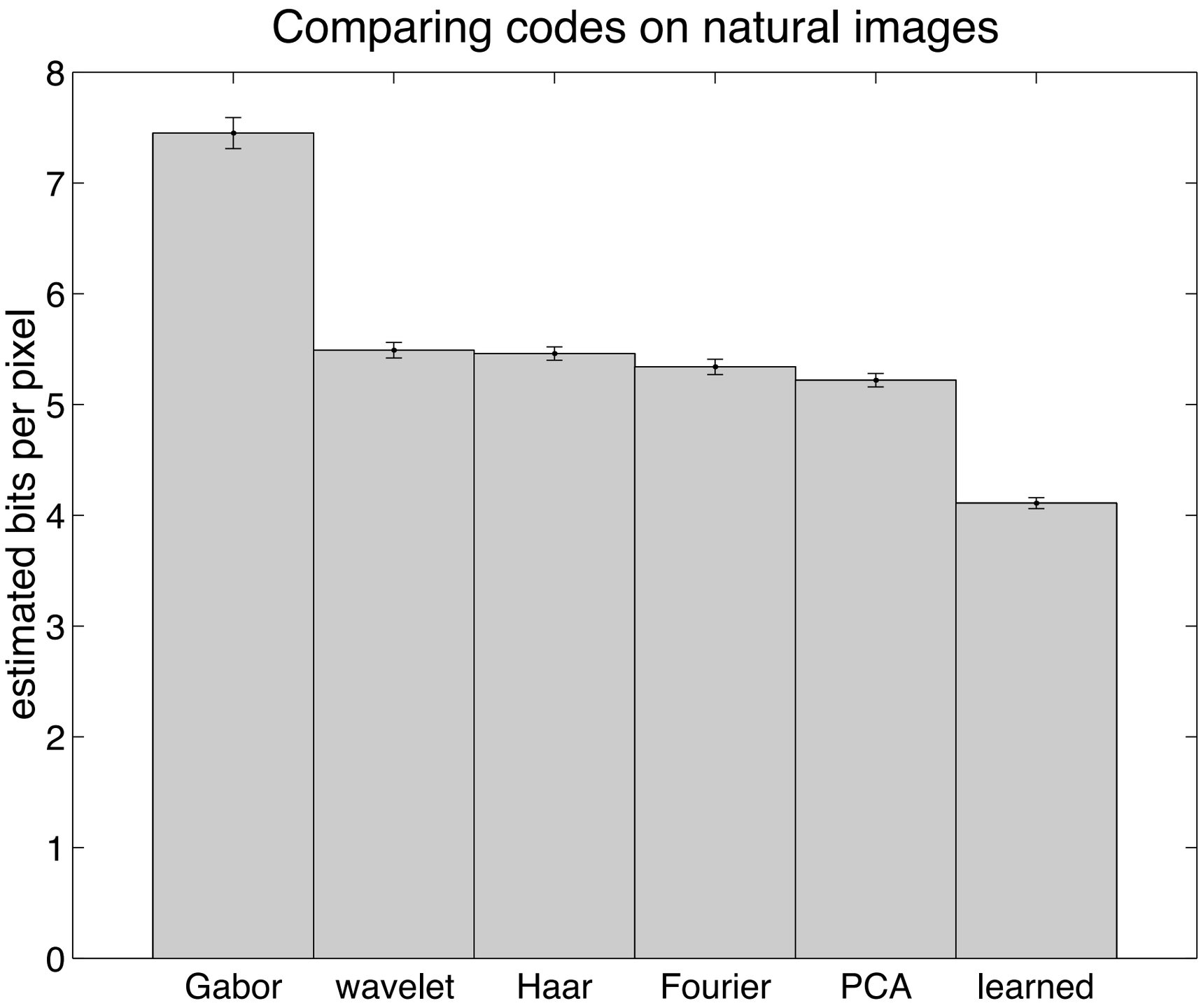


PCA



Each plot shows the basis functions for different image representations.

Lossless coding efficiency for different statistical models



Learning non-Gaussian (ICA) mixture models

Natual Images

Scanned Newspapers

ıy Picture-Perfect Slice of the

ing on Route 4 over a dark and winding mountain pass, the visitor suddenly emerges into a "Lost Horizon" world of hot springs, trout streams and meadows of wildflowers, where cattle and the state's largest elk herd graze side by side.

But the same sense of wide-open Western independence evoked by the vistas has prevented the sale of the land for years. And the deal that is being negotiated for the ranch, which has been owned by one family for almost 40 years, is as much about Western attitudes toward public land as it is about money.

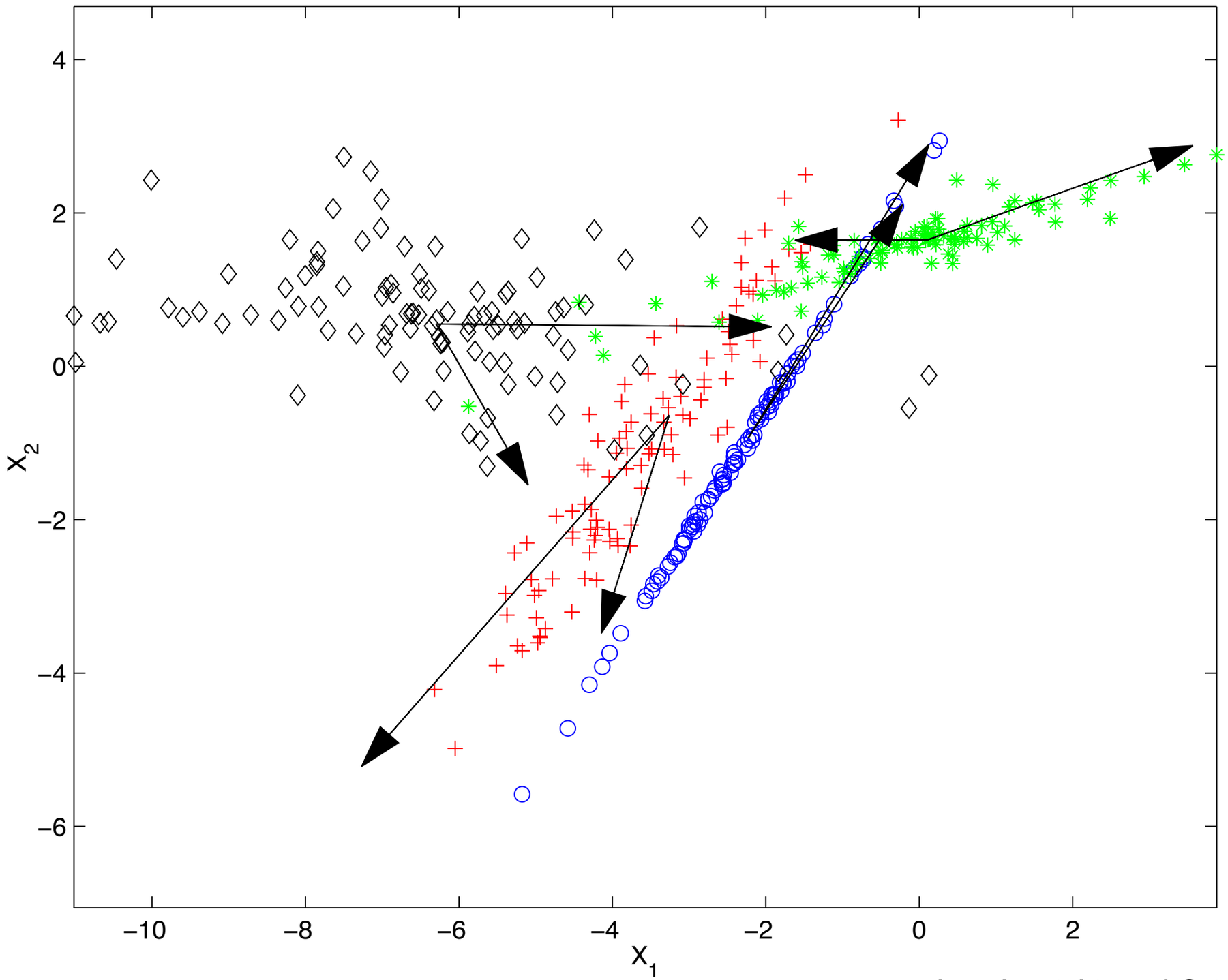
The Administration has long supported the purchase of the ranch, which has been called "the hole in the doughnut" because it is an island surrounded by the Santa Fe National Forest. Last February on a visit to New Mexico, President had Air Force One mak fly over the ranch for a l its dominant feature — wide crater of the dorm

Republicans, noting one-third of New Mexi owned by the Federal have long opposed chases. But in August sentiment began shift cally.

Under legislation dra tor Pete V. Domenici, R New Mexico, the Baca separate unit of the Na system, owned by the Forest Service, but m trust, comprised of n pointed by the Presiden

Continued on Pag

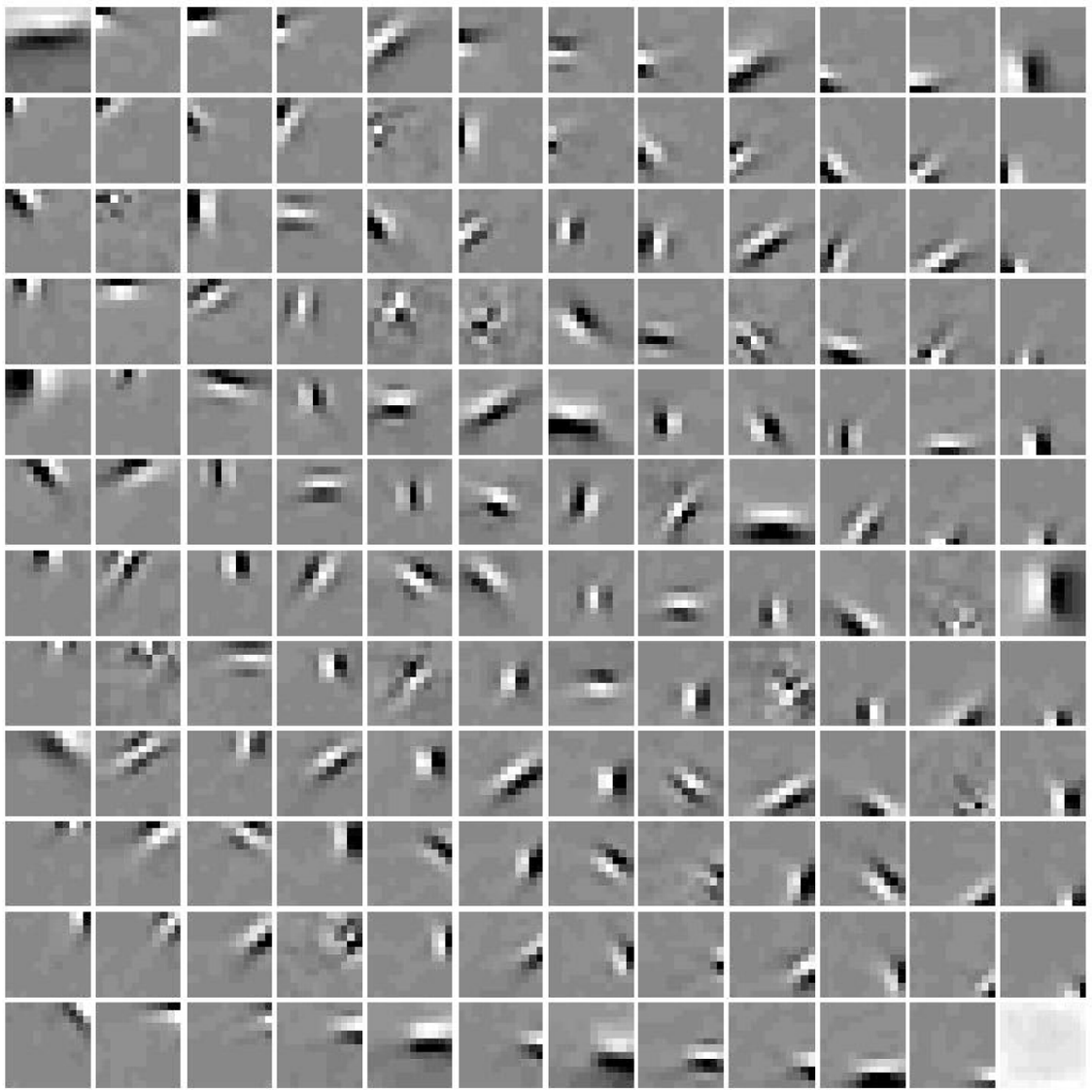
ICA mixture models



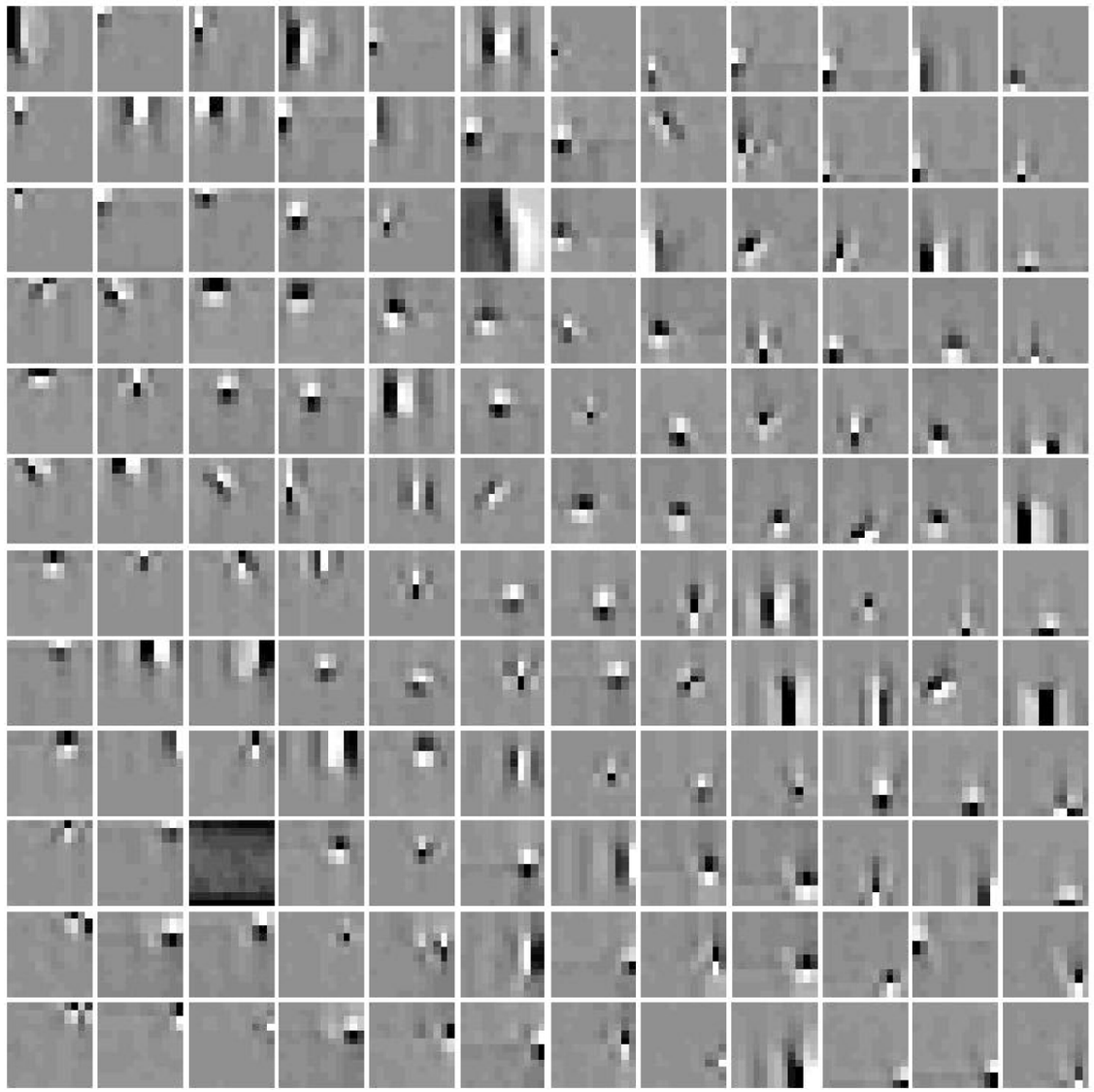
Lee, Lewicki, and Sejnowski, 2000

Learing ICA mixtures for natural images and newpapers

"Cluster" 1: Natural Image Basis

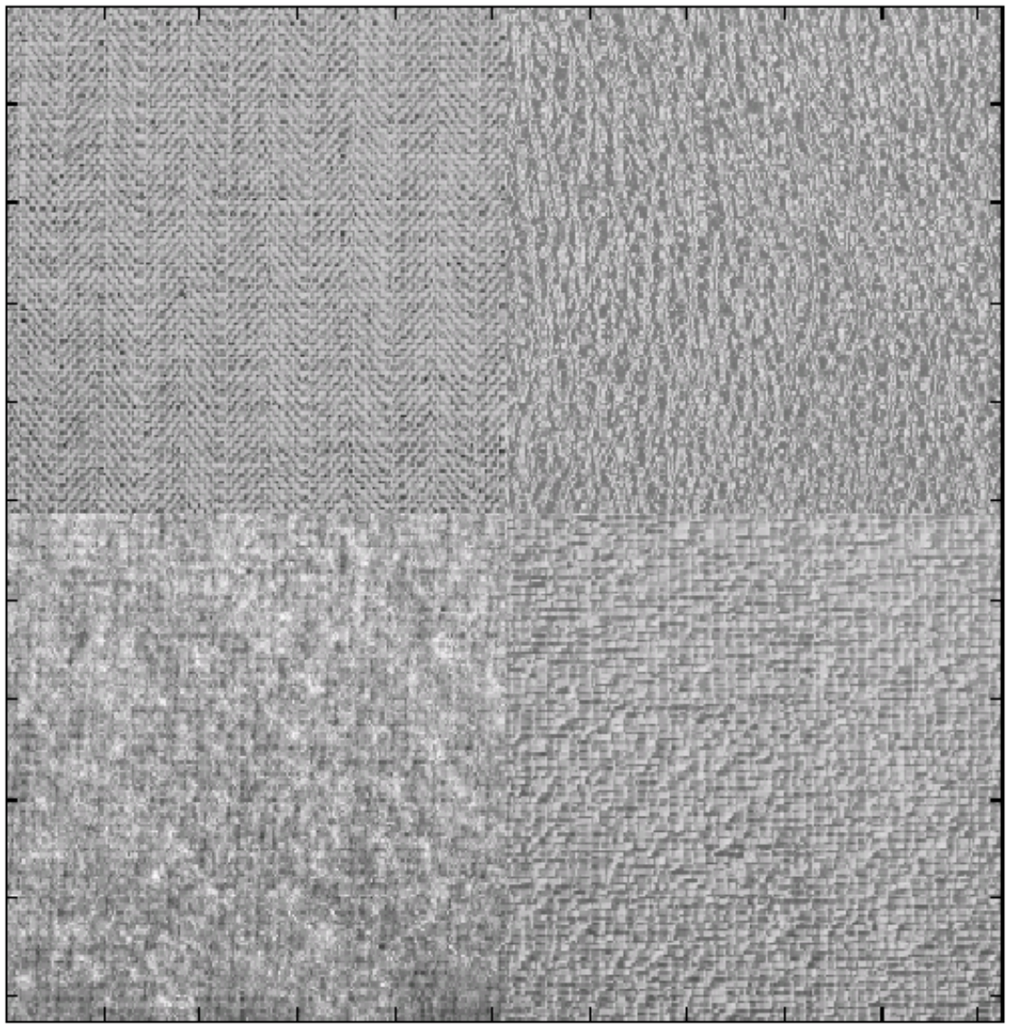


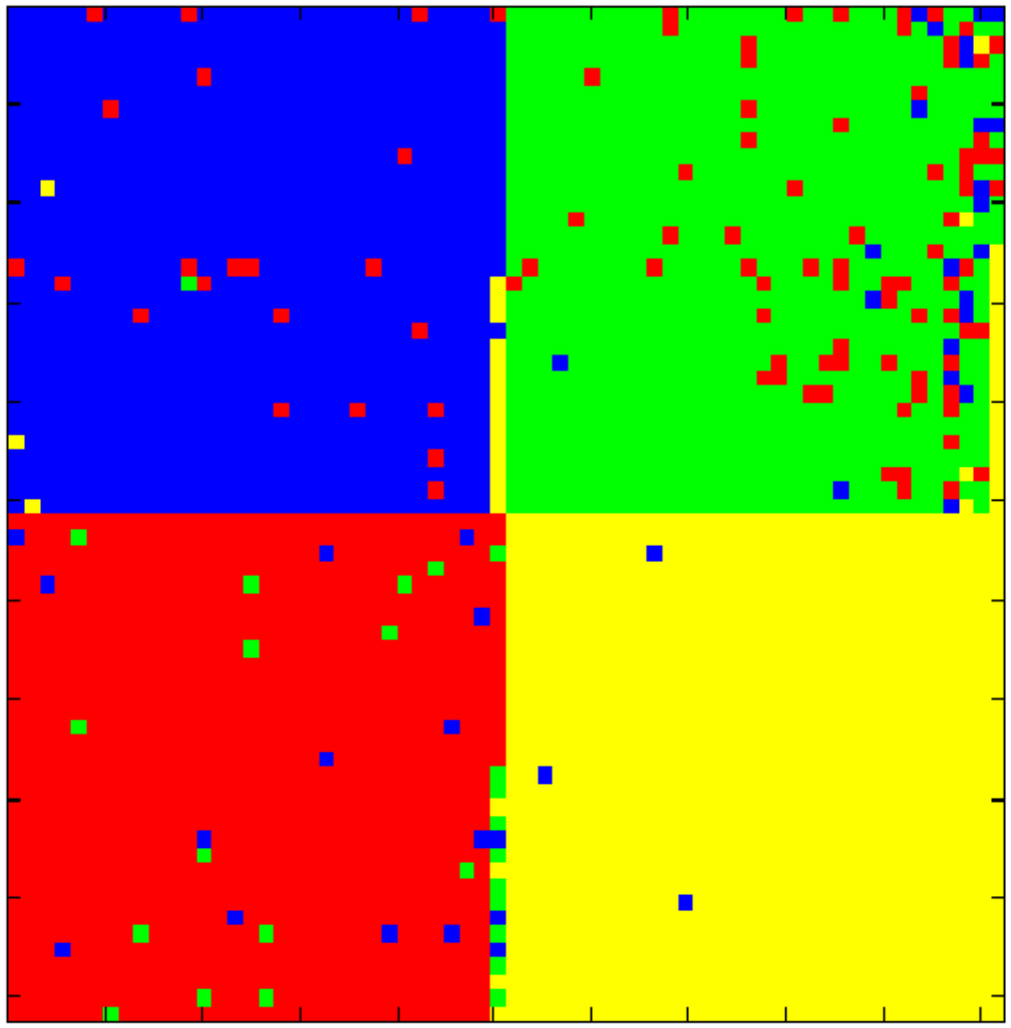
"Cluster" 2: Scanned Newspaper Basis



Lee, Lewicki, and Sejnowski, 2000

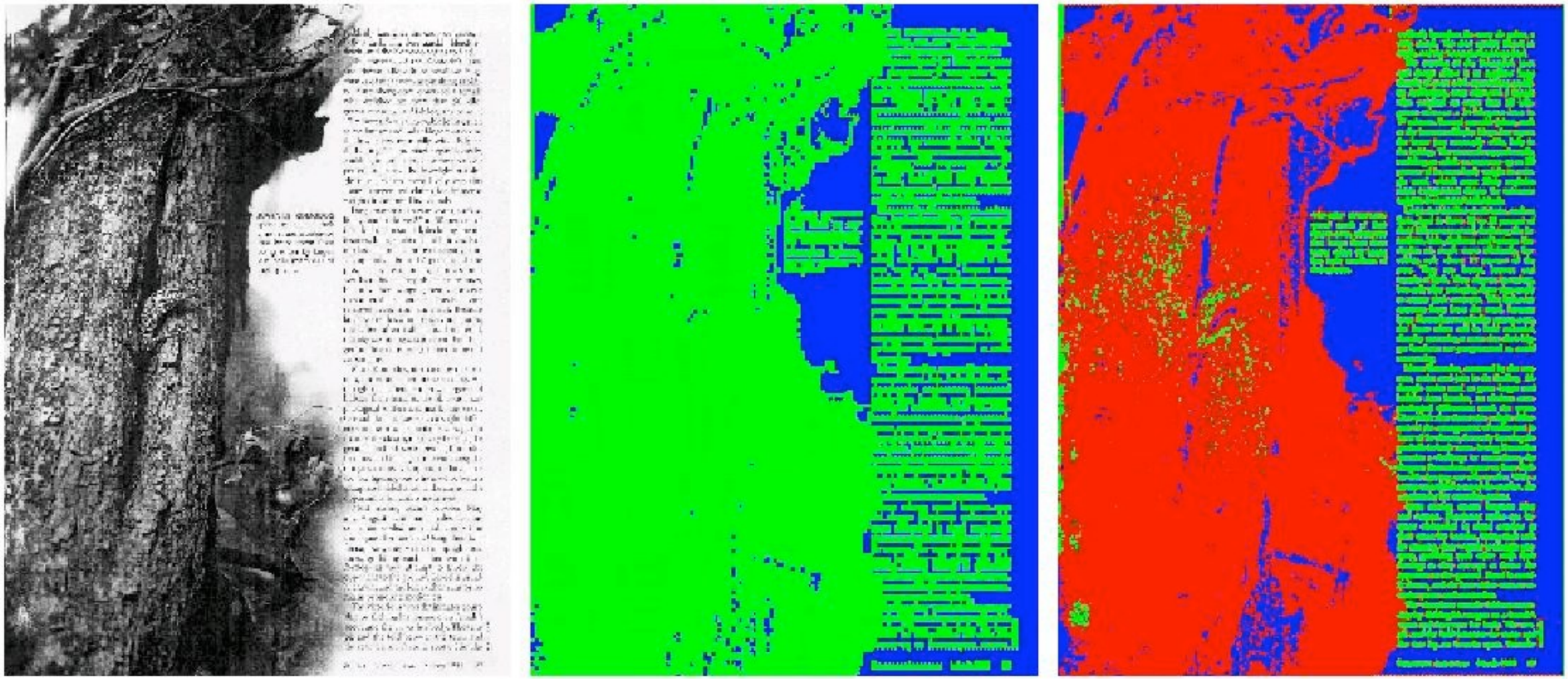
ICA mixtures for similar textures





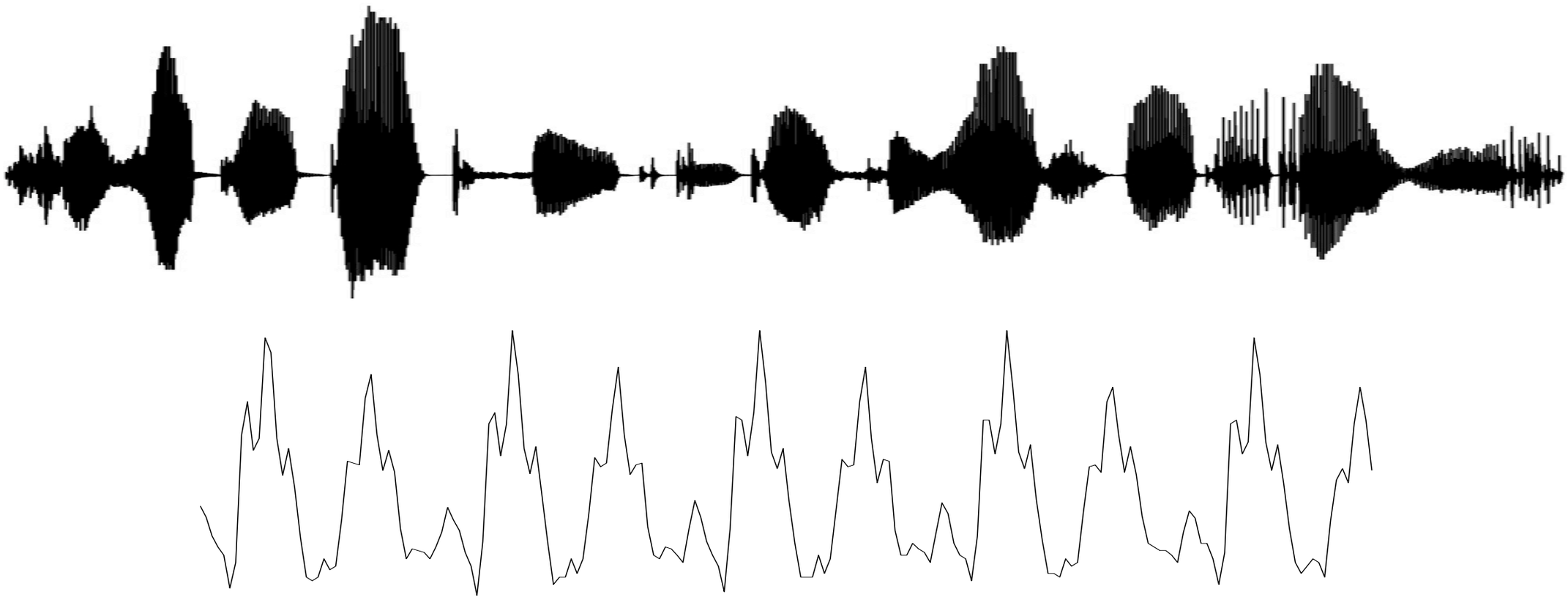
Lee, Lewicki, and Sejnowski, 2000

Fine-grained image segmentation



Lee and Lewicki, 2002

Example: Efficient coding of speech waveforms



- We do *not* assume a Fourier or spectral representation.
- Goal:

  Predict optimal transformation of acoutsic waveform from statistics of the acoustic environment.
- Use a simple model: bank of linear filters

Learning the optimal codes

Goal:

Predict optimal transformation of sound waveform from statistics of the acoustic environment

Learning procedure:

- random sound segments (8 msec)
- optimize features using ICA

What sounds to use?

What tasks are auditory systems adapted to do?

- localization ⇒ environmental sounds

- communication ⇒ vocalizations

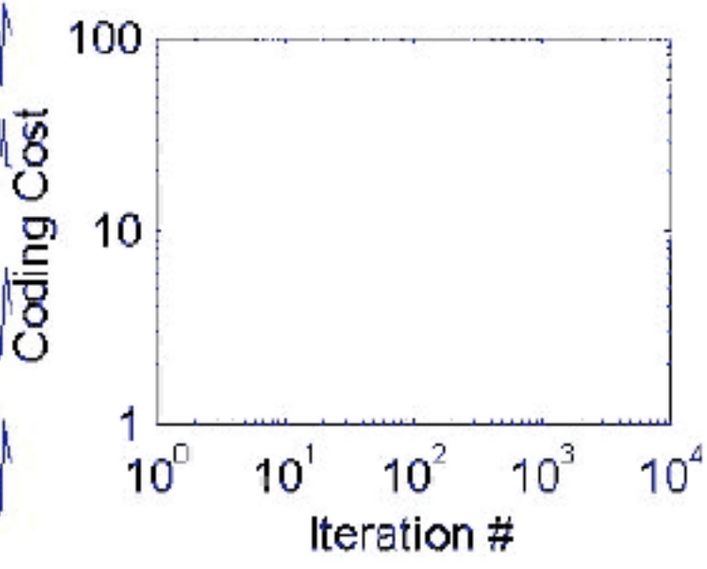
- general sound recognition

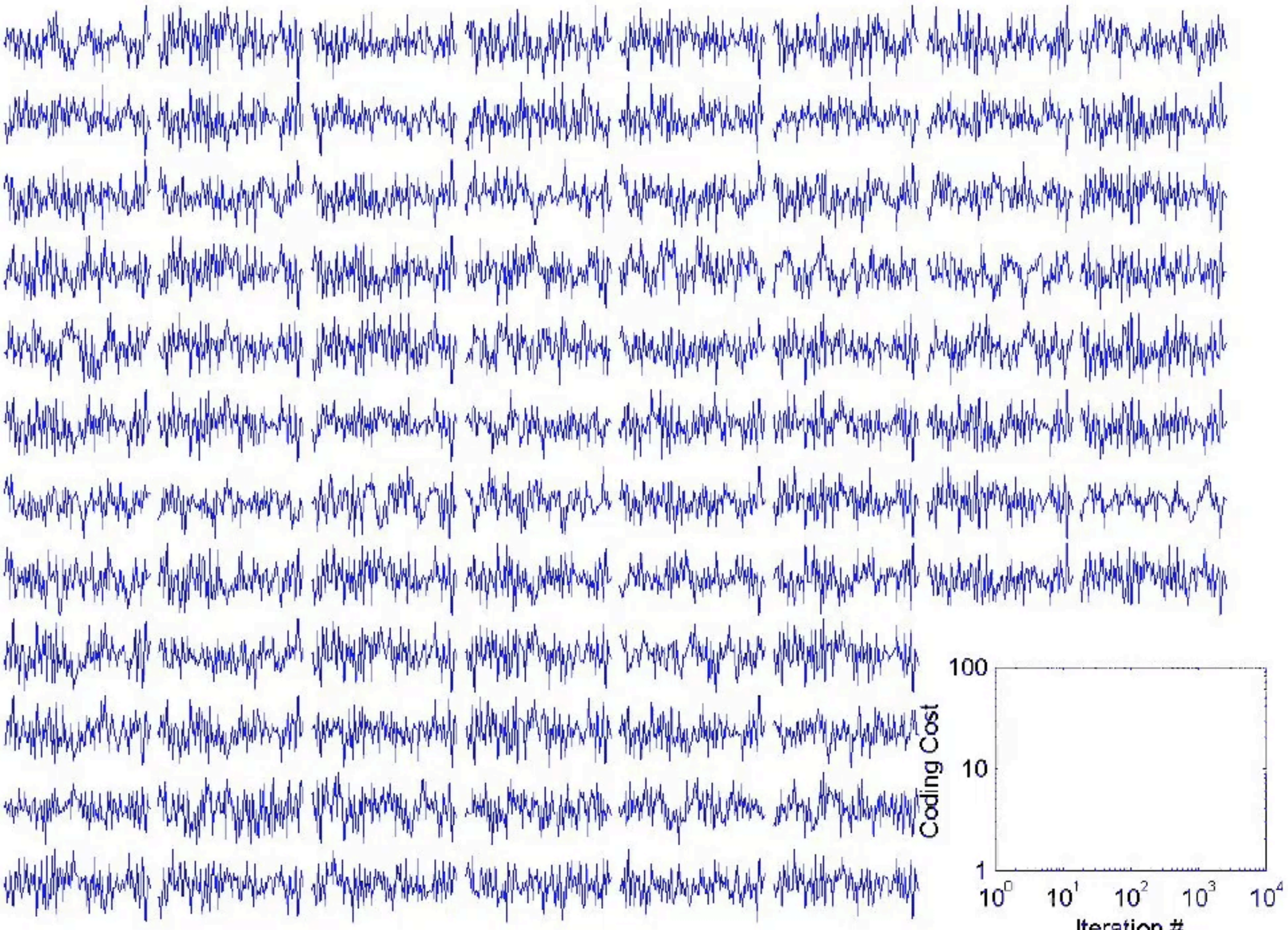
Use a variety of sound ensembles:

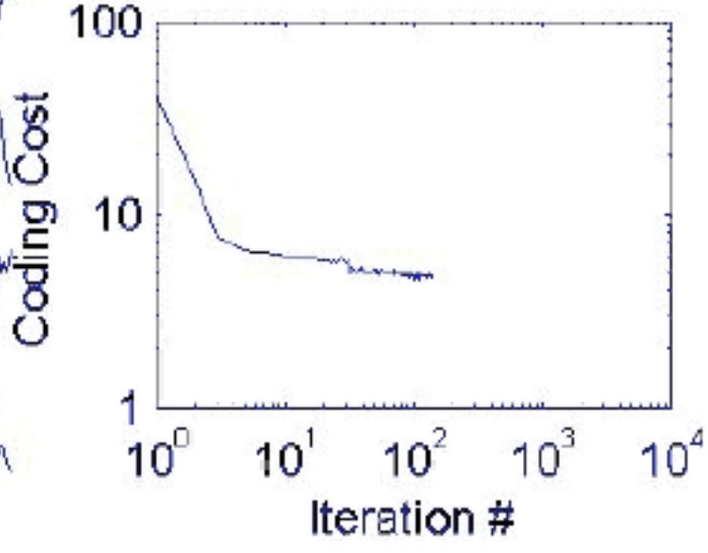
- non-harmonic *environmental sounds* (e.g. footsteps, stream sounds, etc.)

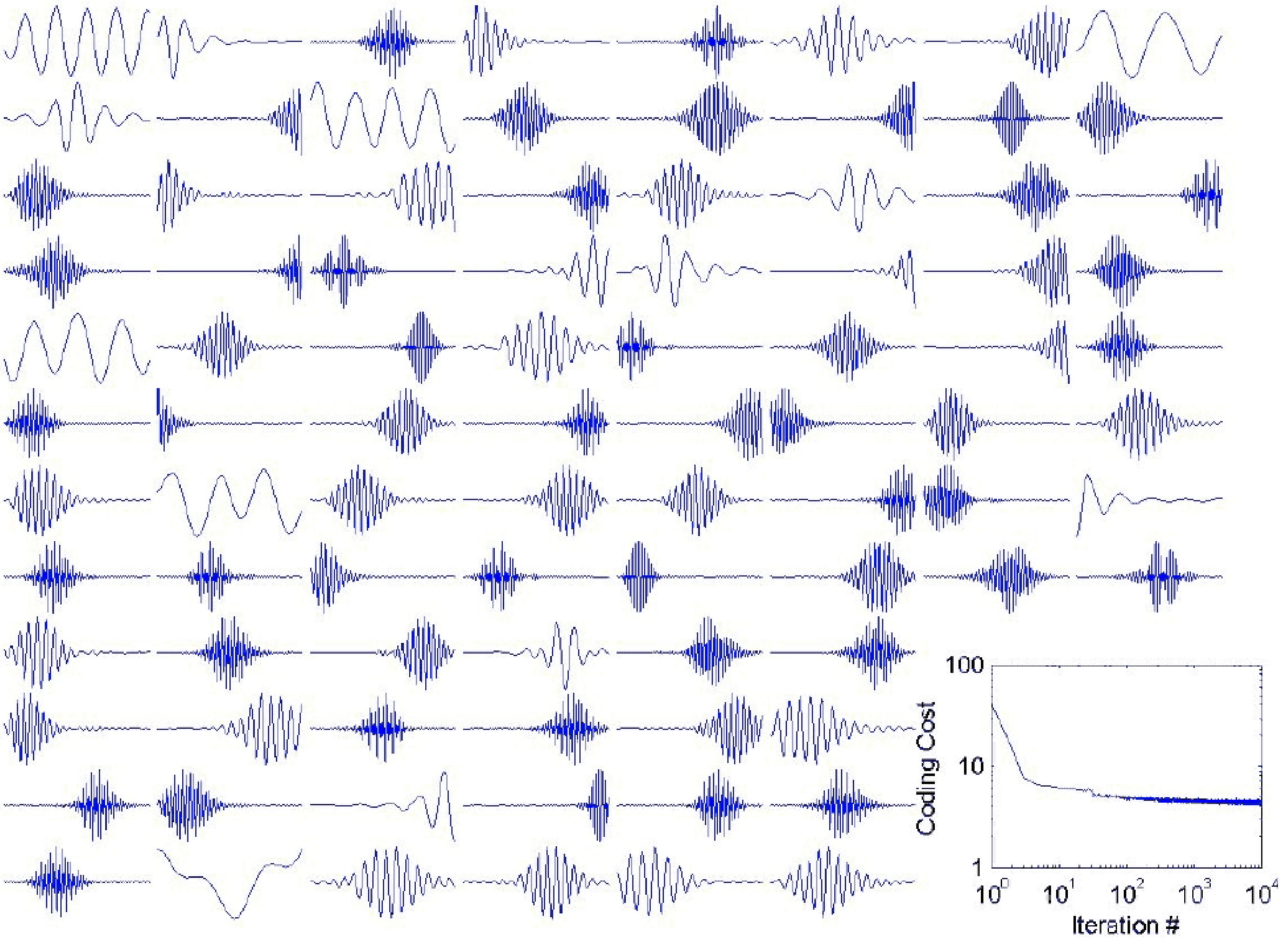
- *animal vocalizations* (rainforest mammals, e.g. chirps, screeches, cries, etc.)

- *speech* (samples from 100 male & female speakers from the TIMIT corpus)


100
10
1
Coding Cost
10^0
10^1
10^2
10^3
10^4
Iteration #


100
10
1
Coding Cost
10^0
10^1
10^2
10^3
10^4
Iteration #

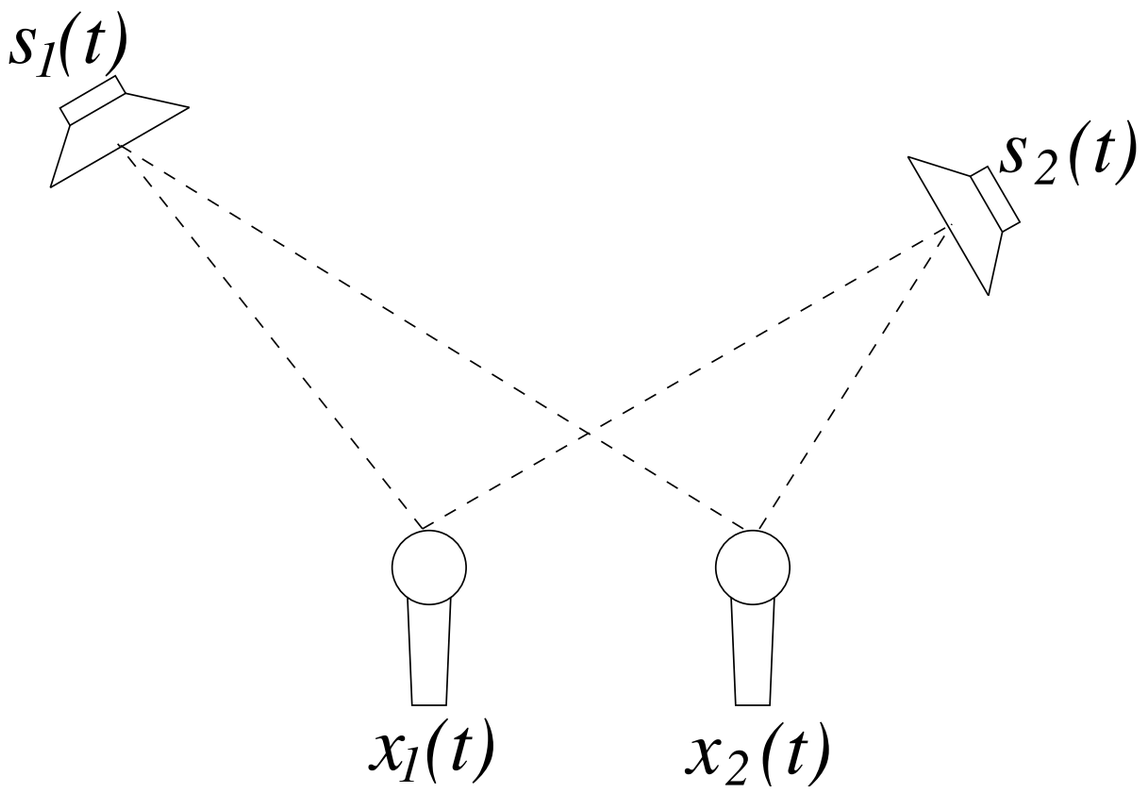

100
10
1
Coding Cost
10^0
10^1
10^2
10^3
10^4
Iteration #


Coding Cost
100
10
1
10^0
10^1
10^2
10^3
10^4
Iteration #

Blind source separation

Modeling the cocktail party problem

Suppose we have two speakers (sources), $s_1(t)$ and $s_2(t)$ and two microphones (mixtures), $x_1(t)$ and $x_2(t)$:



The general problem is called *blind source separation*. We only observe the mixtures and are "blind" to the sources.

How do we model this in general?

A general formulation

Suppose we have M sources

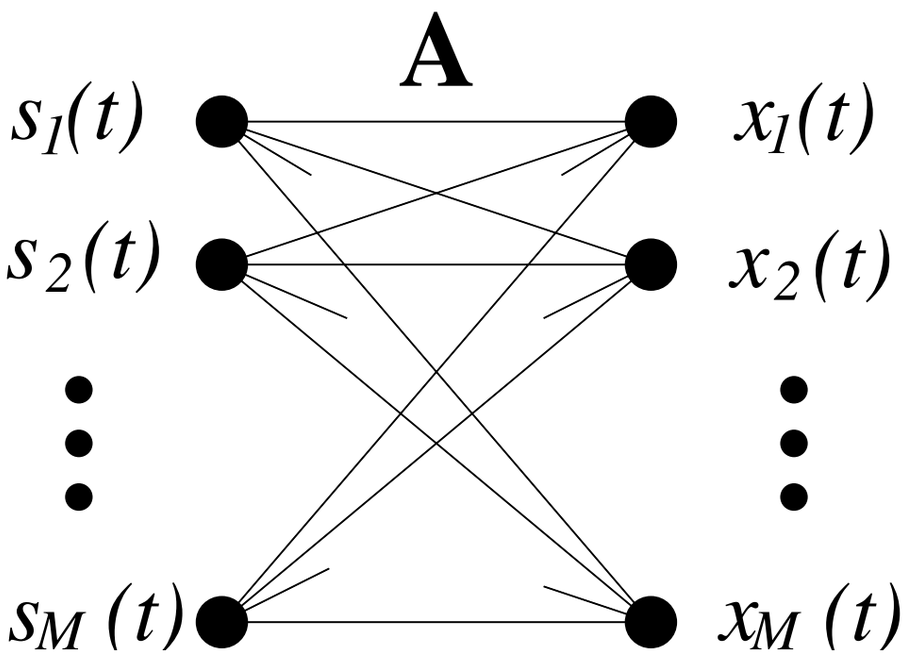
$$s_1(t), \ldots, s_M(t)$$

and M mixtures

$$x_1(t), \ldots, x_2(t)$$

What's the simplest mathematical model?

This can represented diagramatically as:



A general formulation

Suppose we have M sources

$$s_1(t), \ldots, s_M(t)$$

and M mixtures

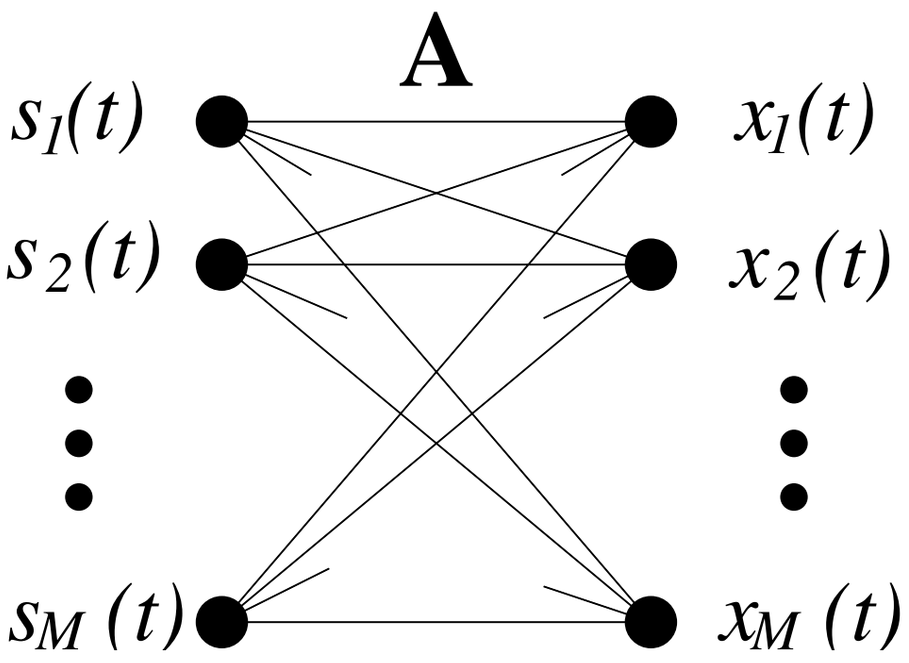
$$x_1(t), \ldots, x_2(t)$$

What's the simplest mathematical model?

Assume linear, instantaneous mixing:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

This can represented diagramatically as:



A general formulation

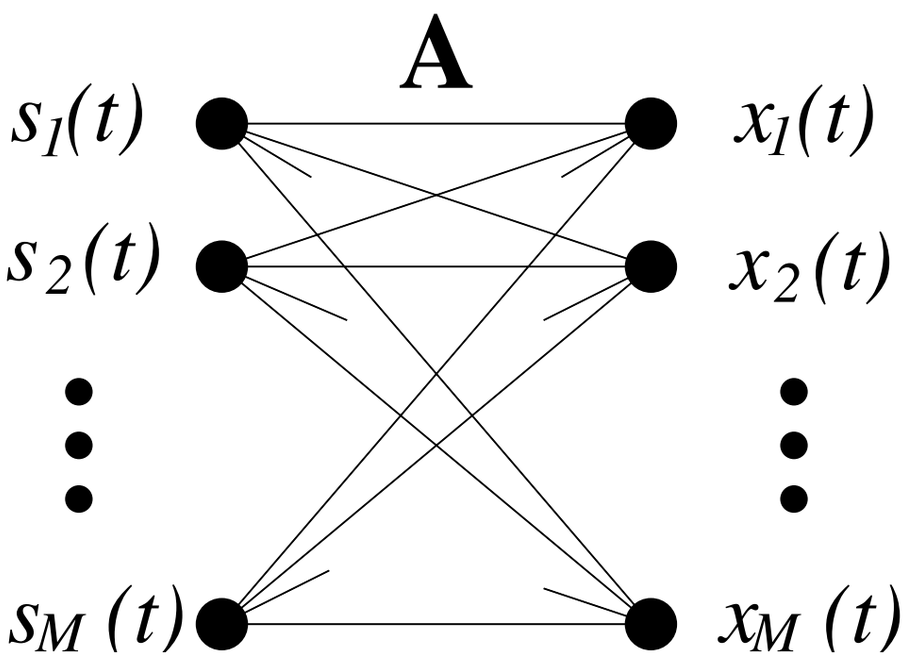
Suppose we have M sources

$$s_1(t), \ldots, s_M(t)$$

and M mixtures

$$x_1(t), \ldots, x_2(t)$$

This can represented diagramatically as:



What's the simplest mathematical model?

Assume linear, instantaneous mixing:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

A is a called the *mixing matrix*.

What does this model ignore?

A general formulation

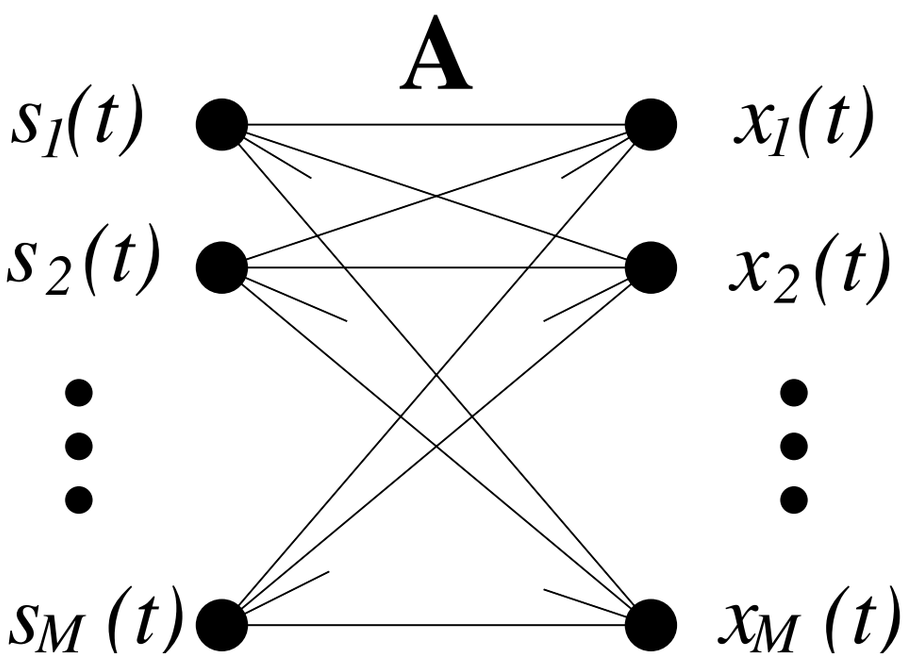
Suppose we have M sources

$$s_1(t), \ldots, s_M(t)$$

and M mixtures

$$x_1(t), \ldots, x_2(t)$$

This can represented diagramatically as:



What's the simplest mathematical model?

Assume linear, instantaneous mixing:

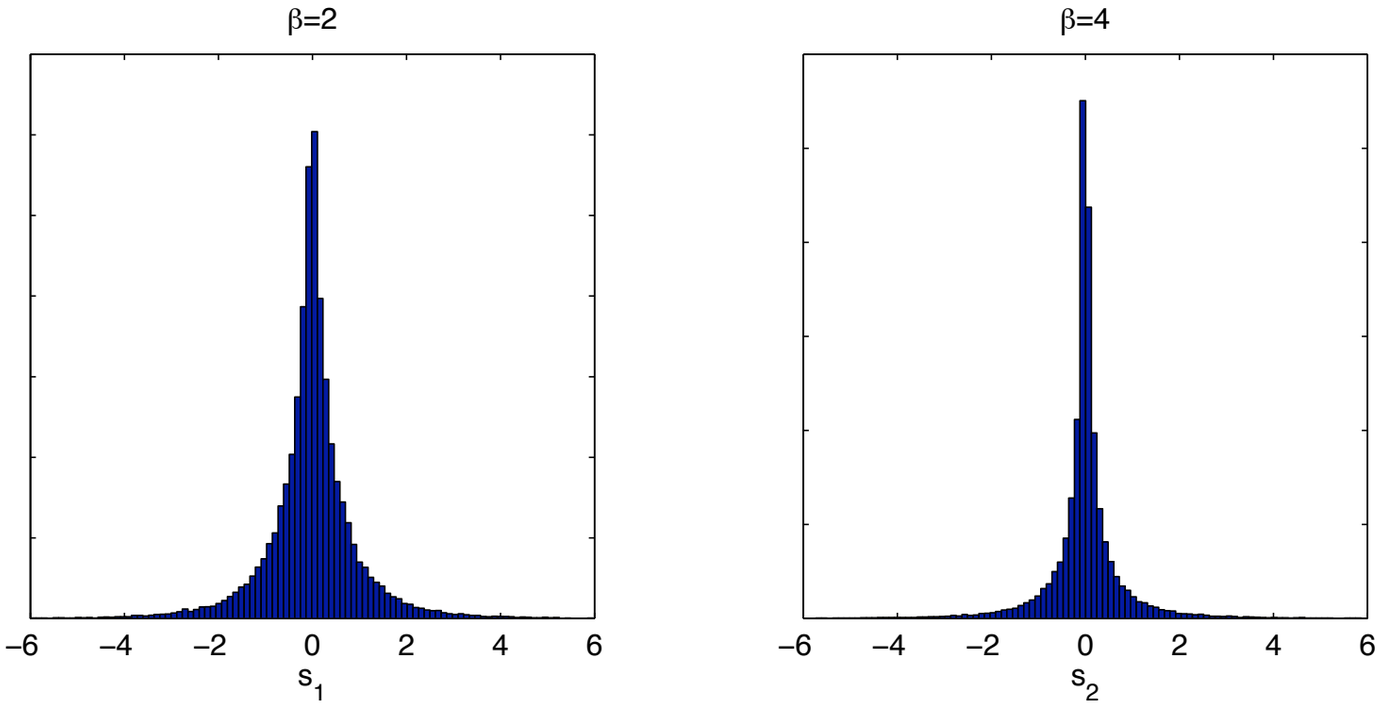
$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

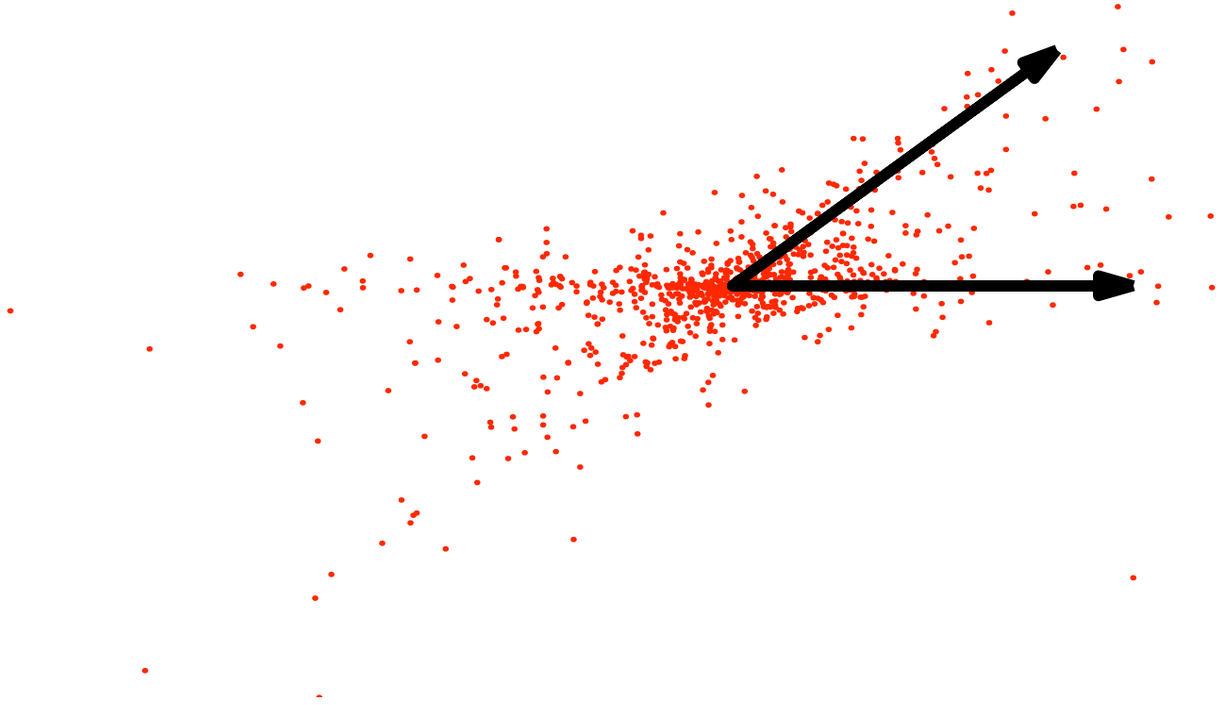
A is a called the *mixing matrix*.

What does this model ignore?

- room acoustics, reverberation, echos
- filtering, noise
- might have more than two sounds
- sounds might not come from a single source
- sound sources could change location

The distribution of sample points



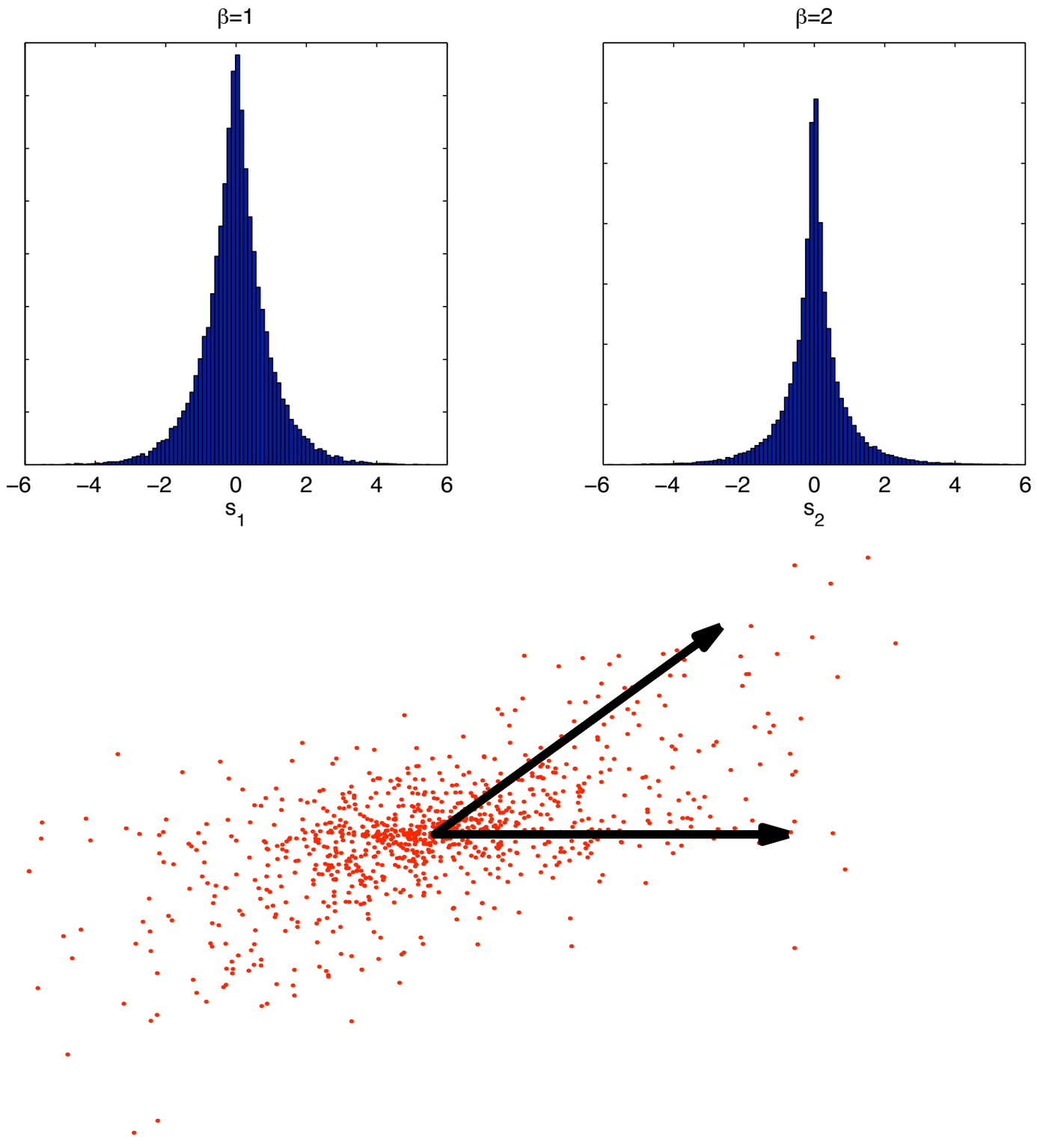


- The histograms show the amplitude distribution of each source.
- The scatter plot shows the 2D distribution of $x_1(t)$ vs $x_2(t)$.
- The parameter β characterizes the sharpness of the data using the distribution

$$P(s) \propto e^{-|s|^{2/(1+\beta)}/2}$$

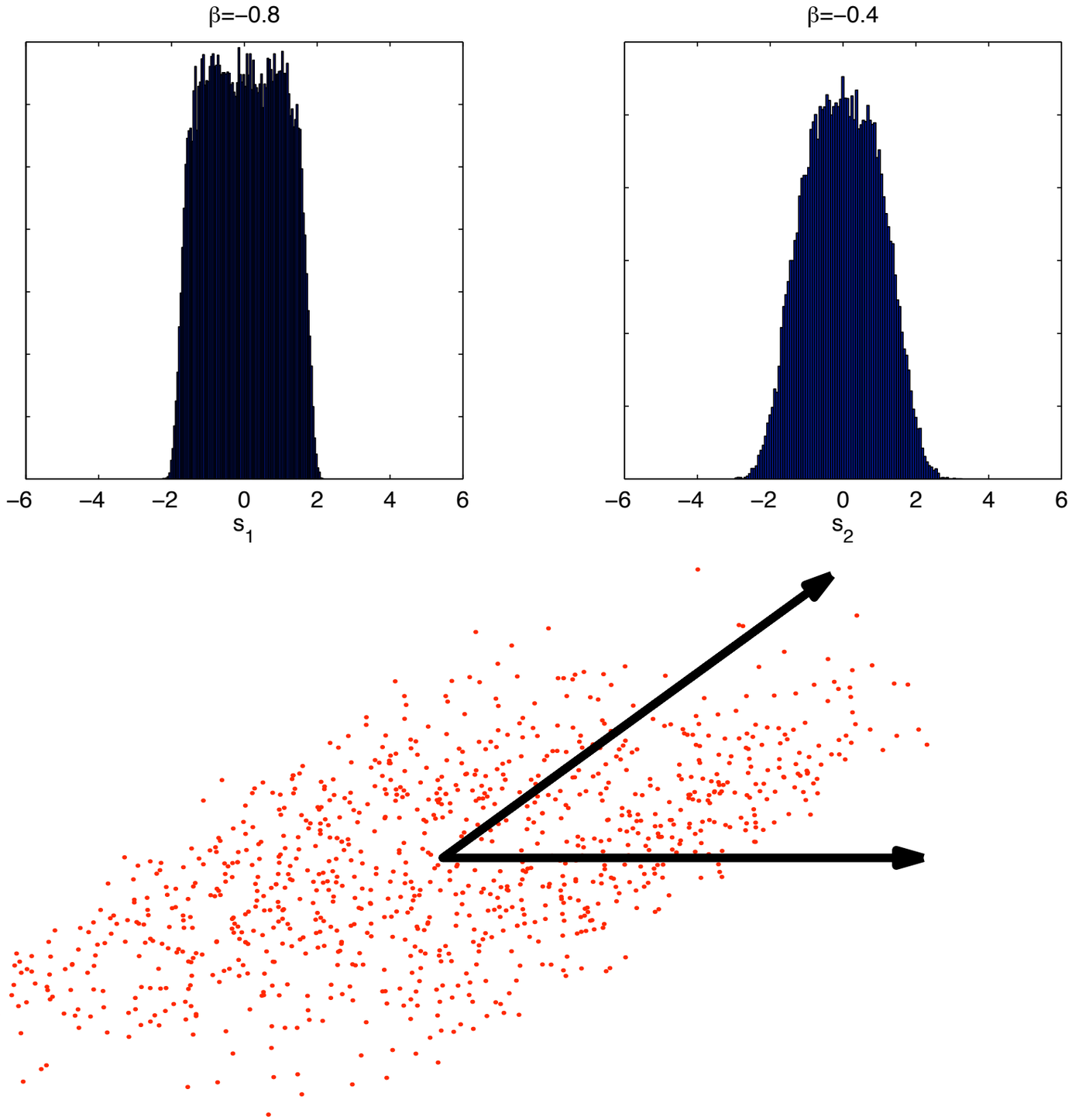
- $\beta = 0$ corresponds to a Gaussian distribution

The distribution of sample points



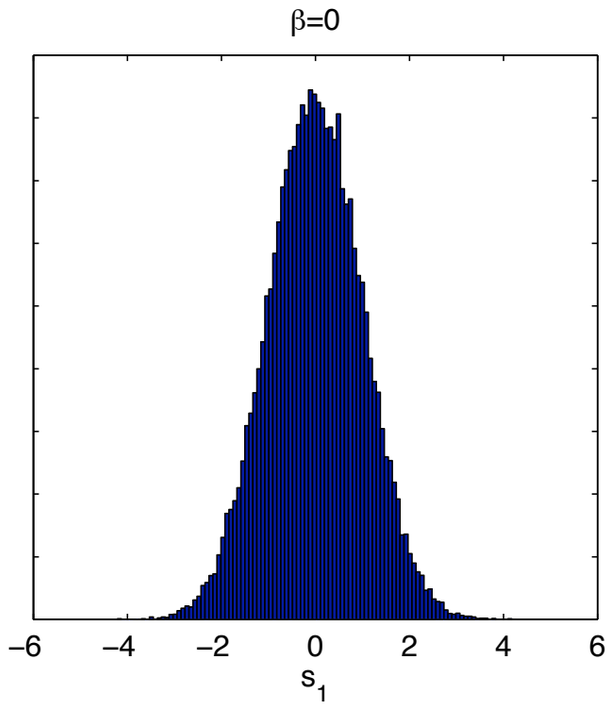
- Sources can have a wide range of amplitude distributions.
- Different directions correspond to different mixing matrices.
- A mixture of non-Gaussian sources create a distribution whose axes can be uniquely determined (within a sign change).

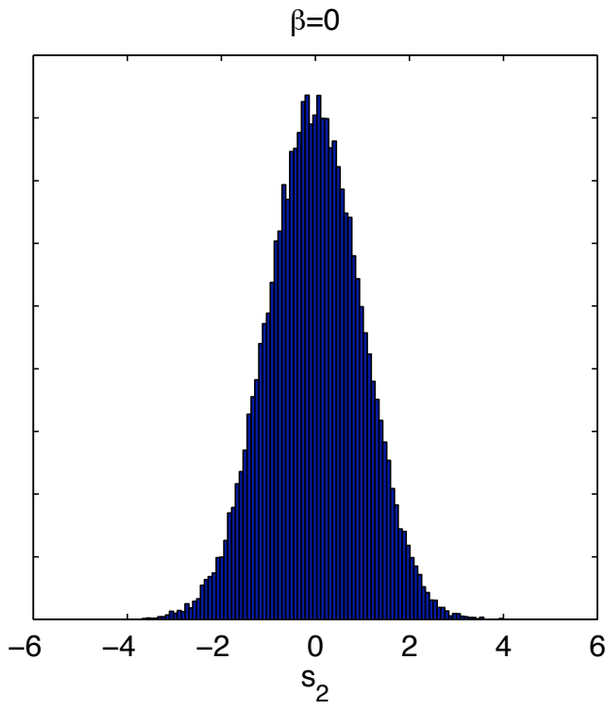
The distribution of sample points

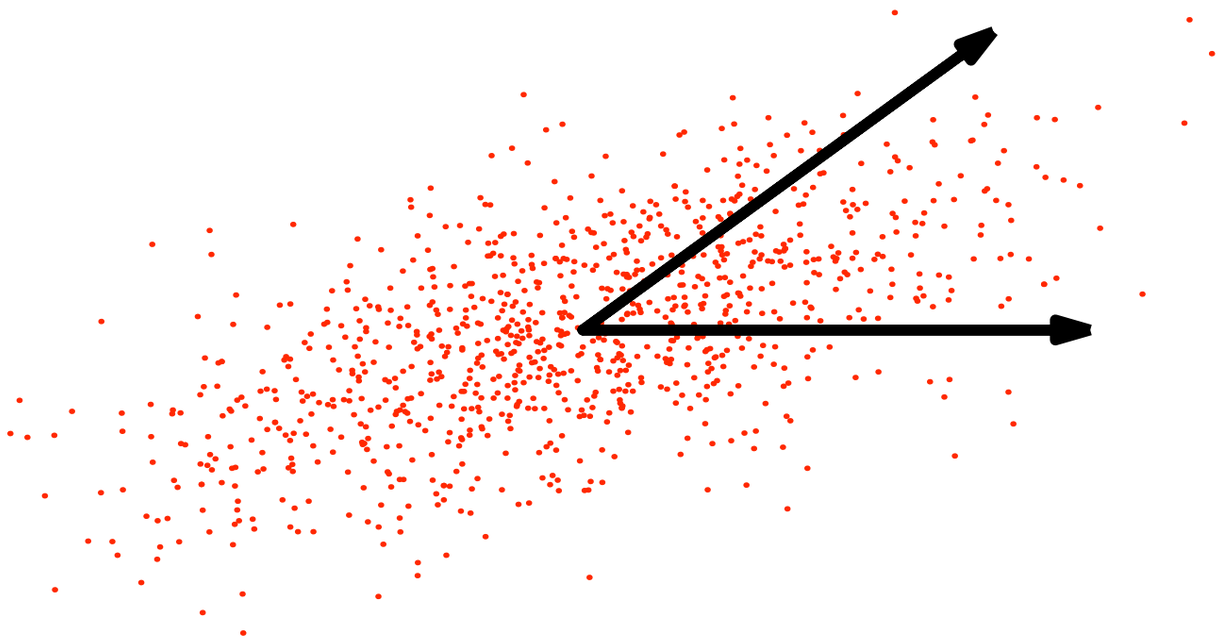


The axes of Sub-Gaussian ources, i.e $\beta < 0$ or negative kurtosis, can still be determined.

A Gaussian distribution of sample points





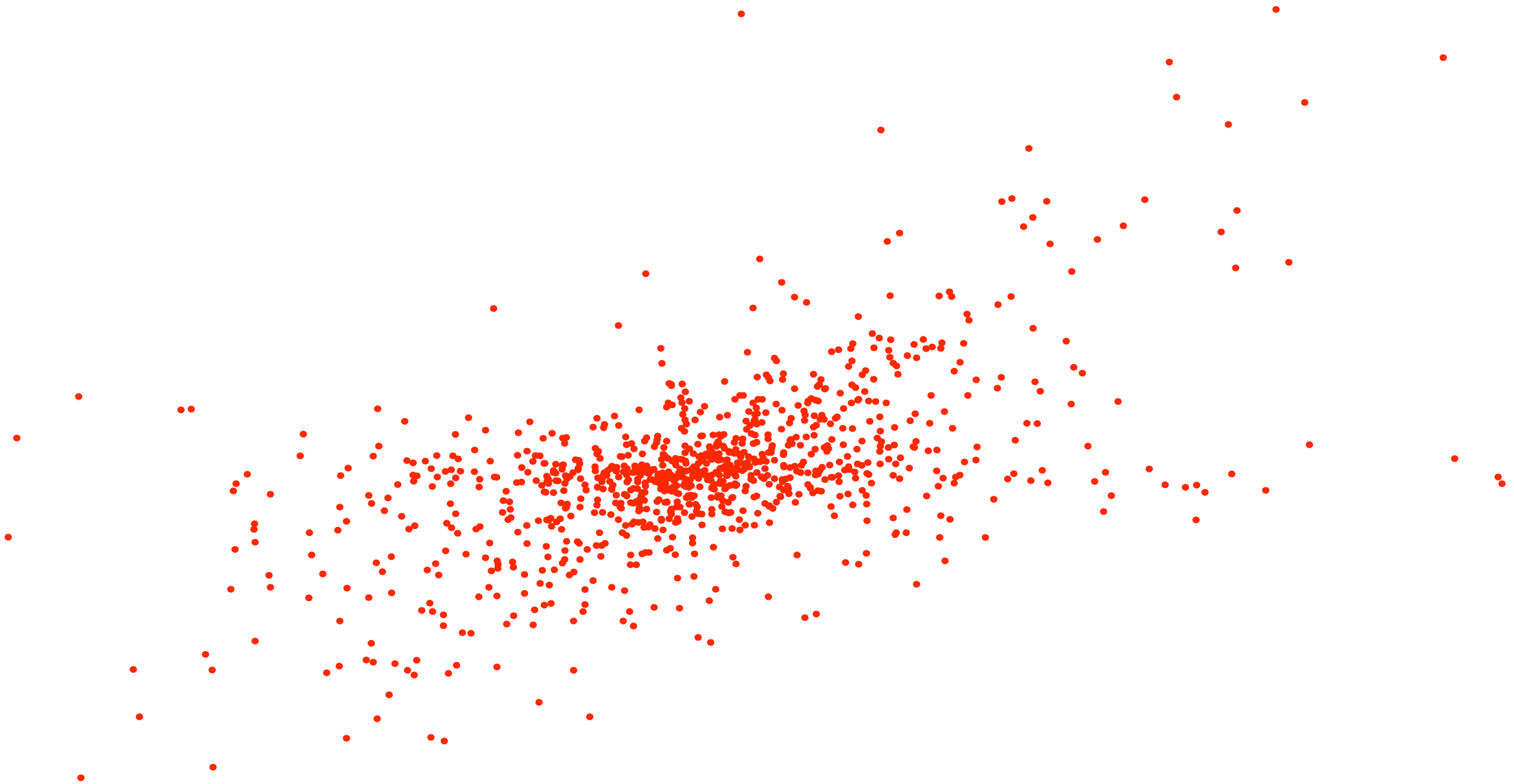


The principal axes of two Gaussian sources are ambiguous.

- Why?

- Because the product of two Gaussians is still a Gaussian, so there are an infinite number of directions that fit the 2D distribution.

Inferring the (un)mixing matrix



How do we determine the axes from just the data?

Modeling non-Gaussian distributions

Learning objective: model statistical density of sources:

$\Rightarrow$ maximize $P(\mathbf{x}|\mathbf{A})$ over $\mathbf{A}$.

Probability of pattern ensemble is:

$$P(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N|\mathbf{A}) = \prod_k P(\mathbf{x}_k|\mathbf{A})$$

To obtain $P(\mathbf{x}|\mathbf{A})$ marginalize over $\mathbf{s}$:

$$\begin{aligned} P(\mathbf{x}|\mathbf{A}) &= \int d\mathbf{s}\, P(\mathbf{x}|\mathbf{A}, \mathbf{s}) P(\mathbf{s}) \\ &= \frac{P(\mathbf{s})}{|\det \mathbf{A}|} \end{aligned}$$

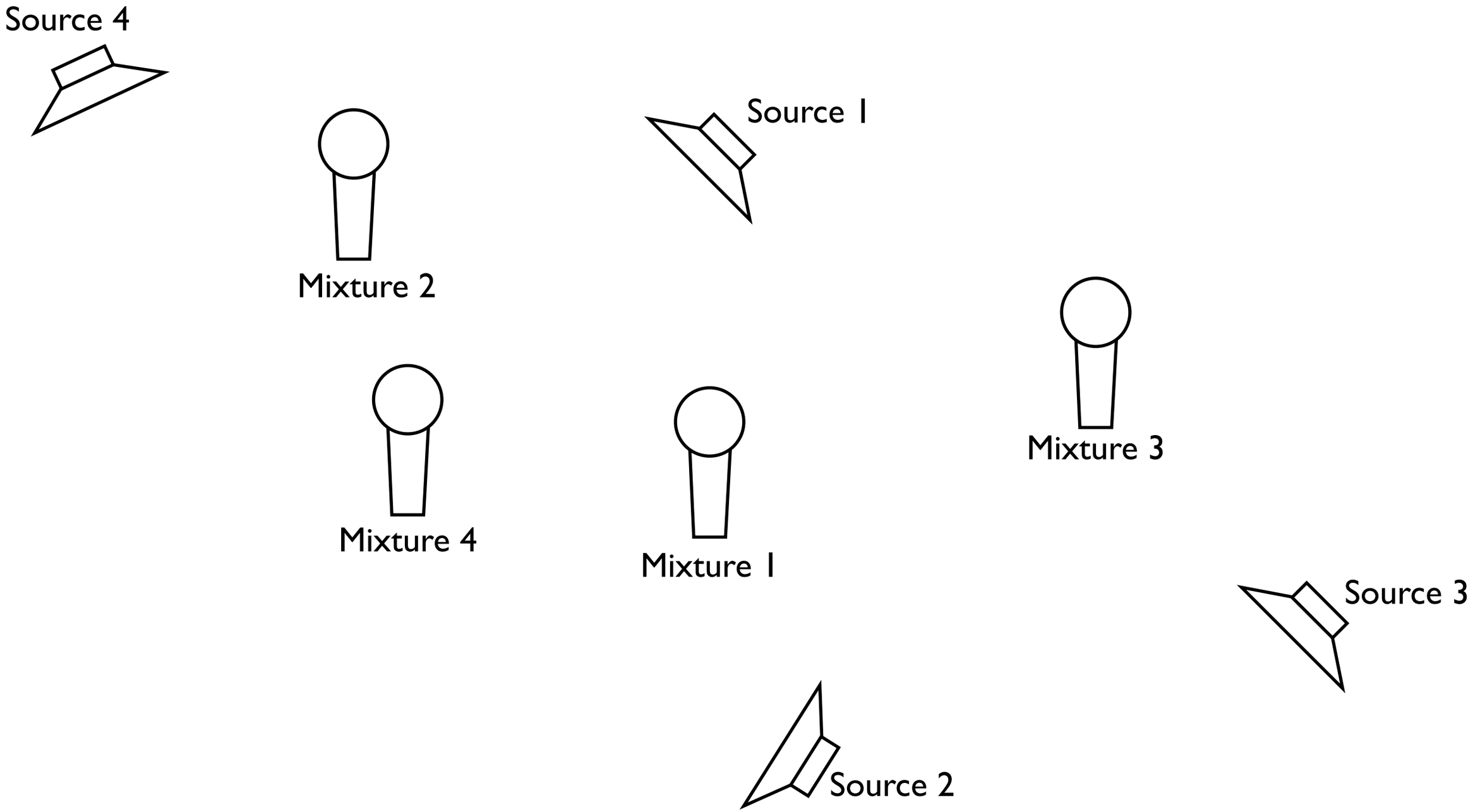
Learning rule (ICA):

$$\begin{aligned} \Delta\mathbf{A} &\propto \mathbf{A}\mathbf{A}^T \frac{\partial}{\partial \mathbf{A}} \log P(\mathbf{x}|\mathbf{A}) \\ &= -\mathbf{A}(\mathbf{z}\mathbf{s}^T - \mathbf{I}), \end{aligned}$$

where $\mathbf{z} = (\log P(\mathbf{s}))'$. Use $P(s_i) \sim \text{ExPwr}(s_i|\mu, \sigma, \beta_i)$.

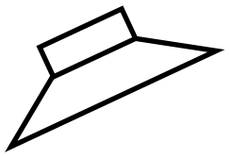
This is identical to the procedure that was used to learn efficient codes, i.e *independent component analysis*.

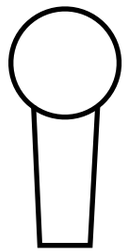
Separating mixtures of real sources



Separating mixtures of real sources

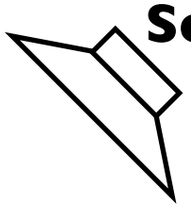
Source 4

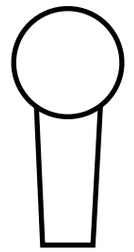




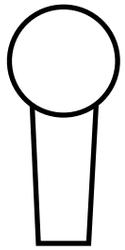
Mixture 2

Source 1

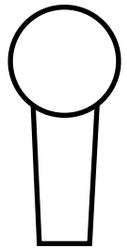




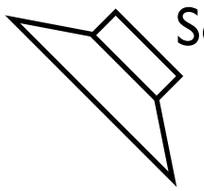
Mixture 3



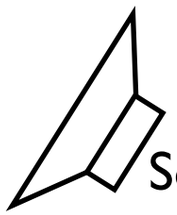
Mixture 4



Mixture 1



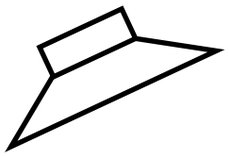
Source 3

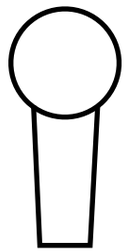


Source 2

Separating mixtures of real sources

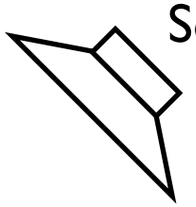
Source 4

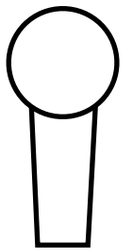




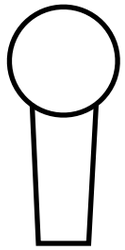
Mixture 2

Source 1

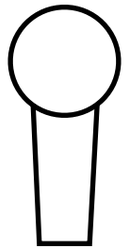




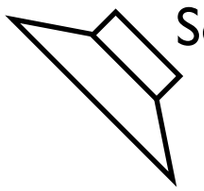
Mixture 3



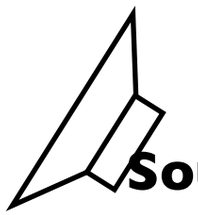
Mixture 4



Mixture 1

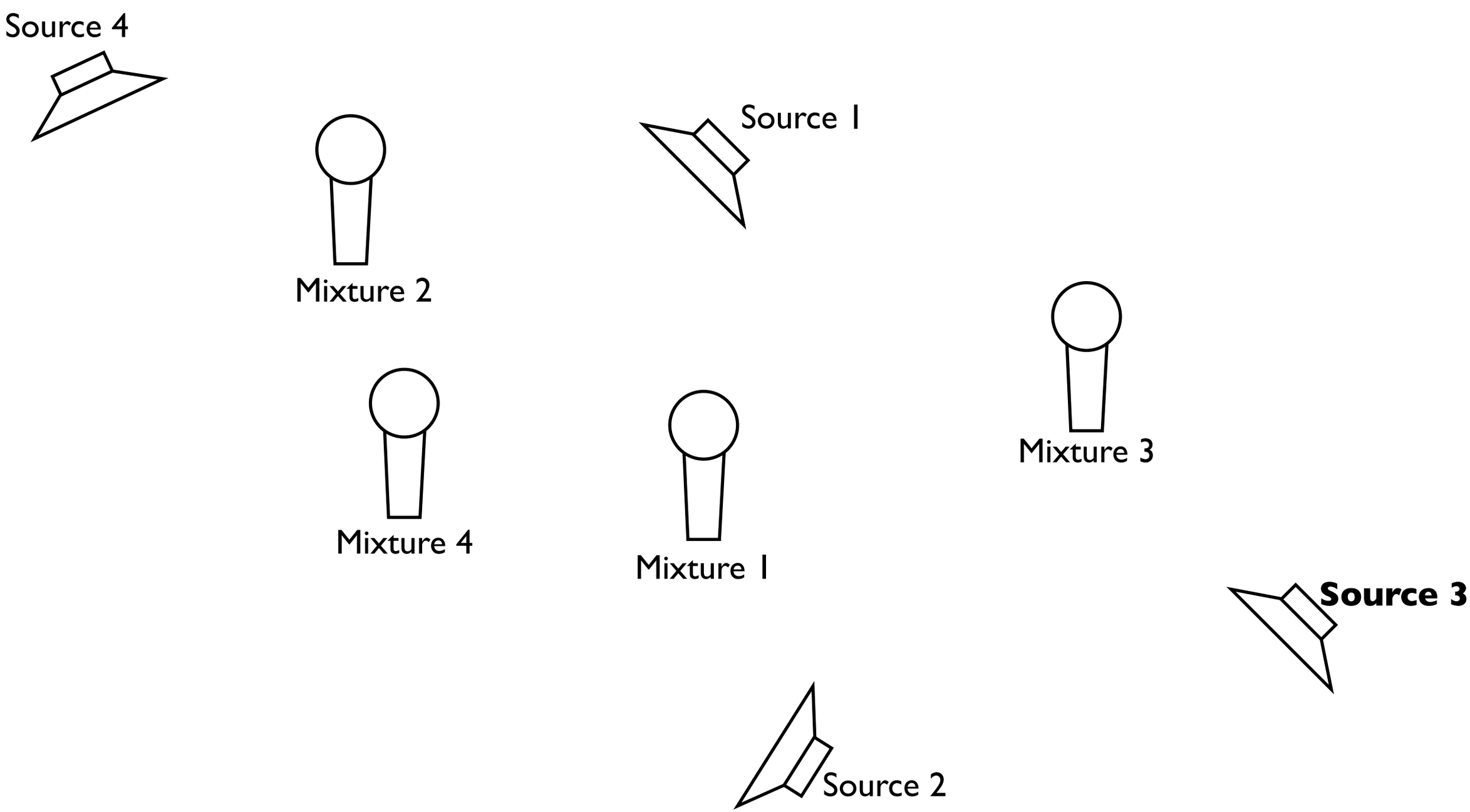


Source 3



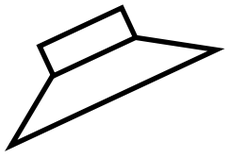
Source 2

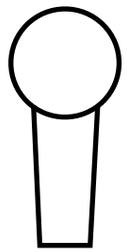
Separating mixtures of real sources



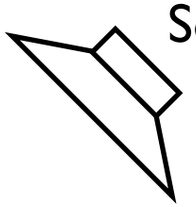
Separating mixtures of real sources

Source 4

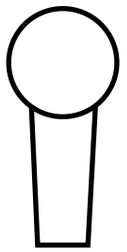




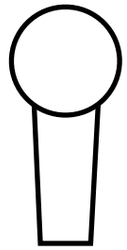
Mixture 2



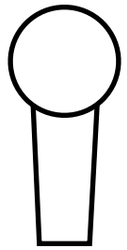
Source 1



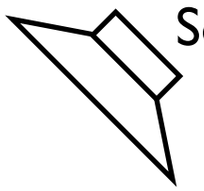
Mixture 3



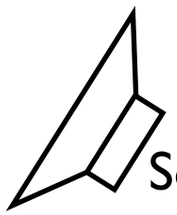
Mixture 4



Mixture 1



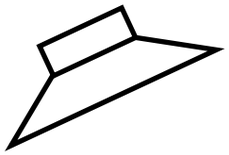
Source 3

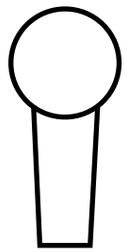


Source 2

Separating mixtures of real sources

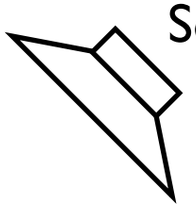
Source 4

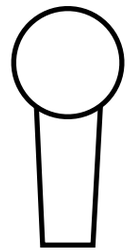




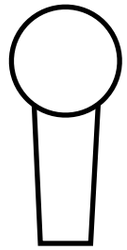
Mixture 2

Source 1

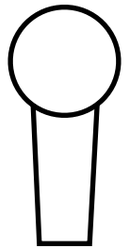




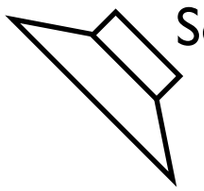
Mixture 3



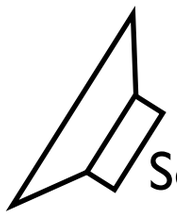
Mixture 4



Mixture 1

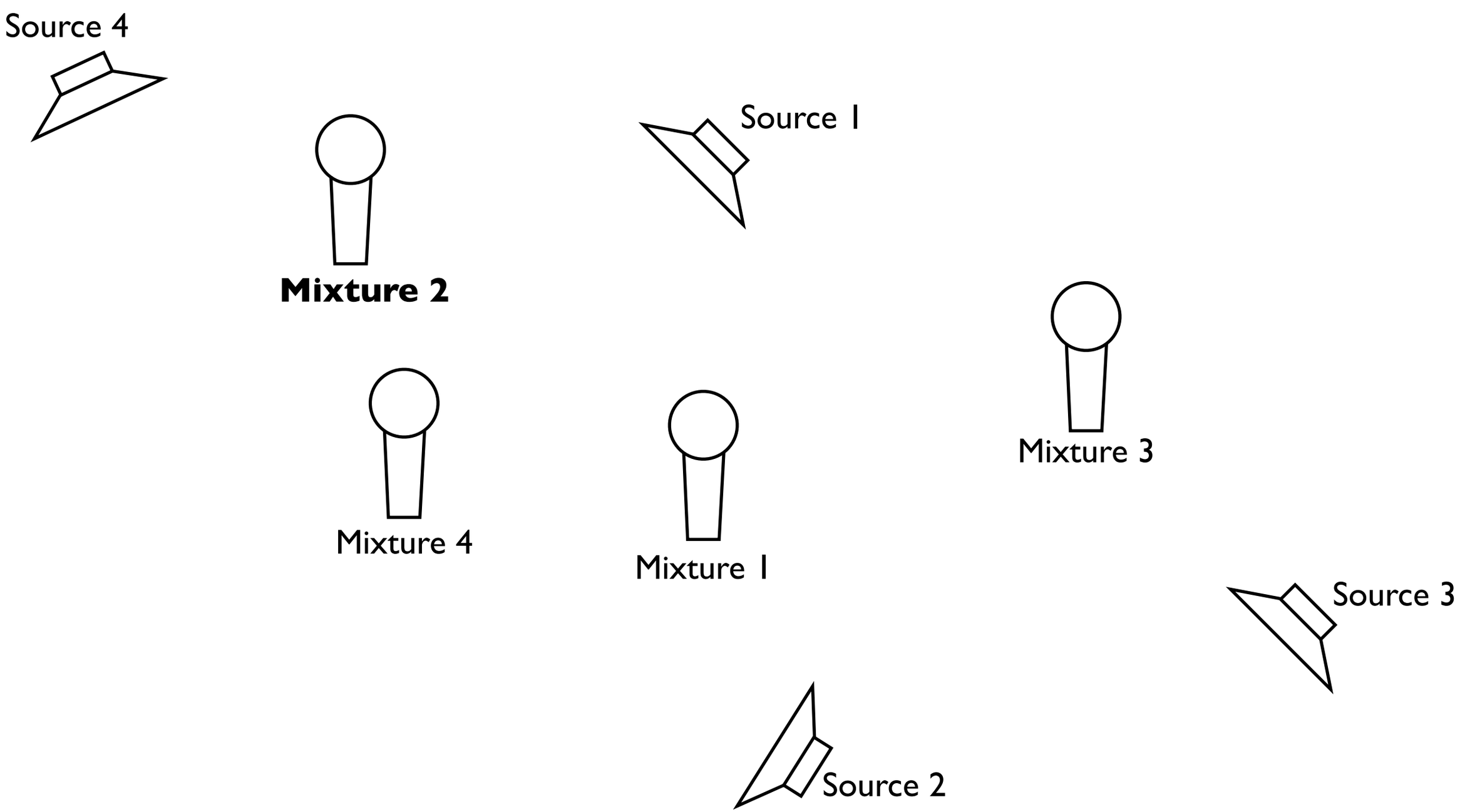


Source 3



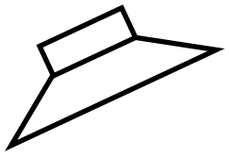
Source 2

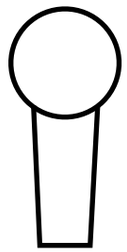
Separating mixtures of real sources



Separating mixtures of real sources

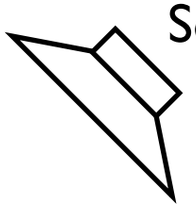
Source 4

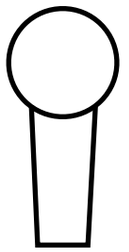




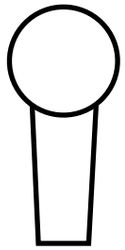
Mixture 2

Source 1

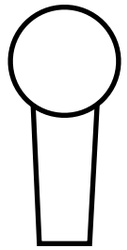




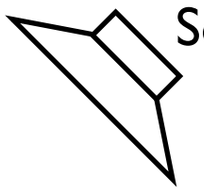
Mixture 3



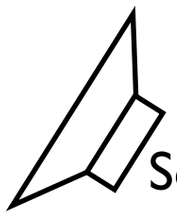
Mixture 4



Mixture 1

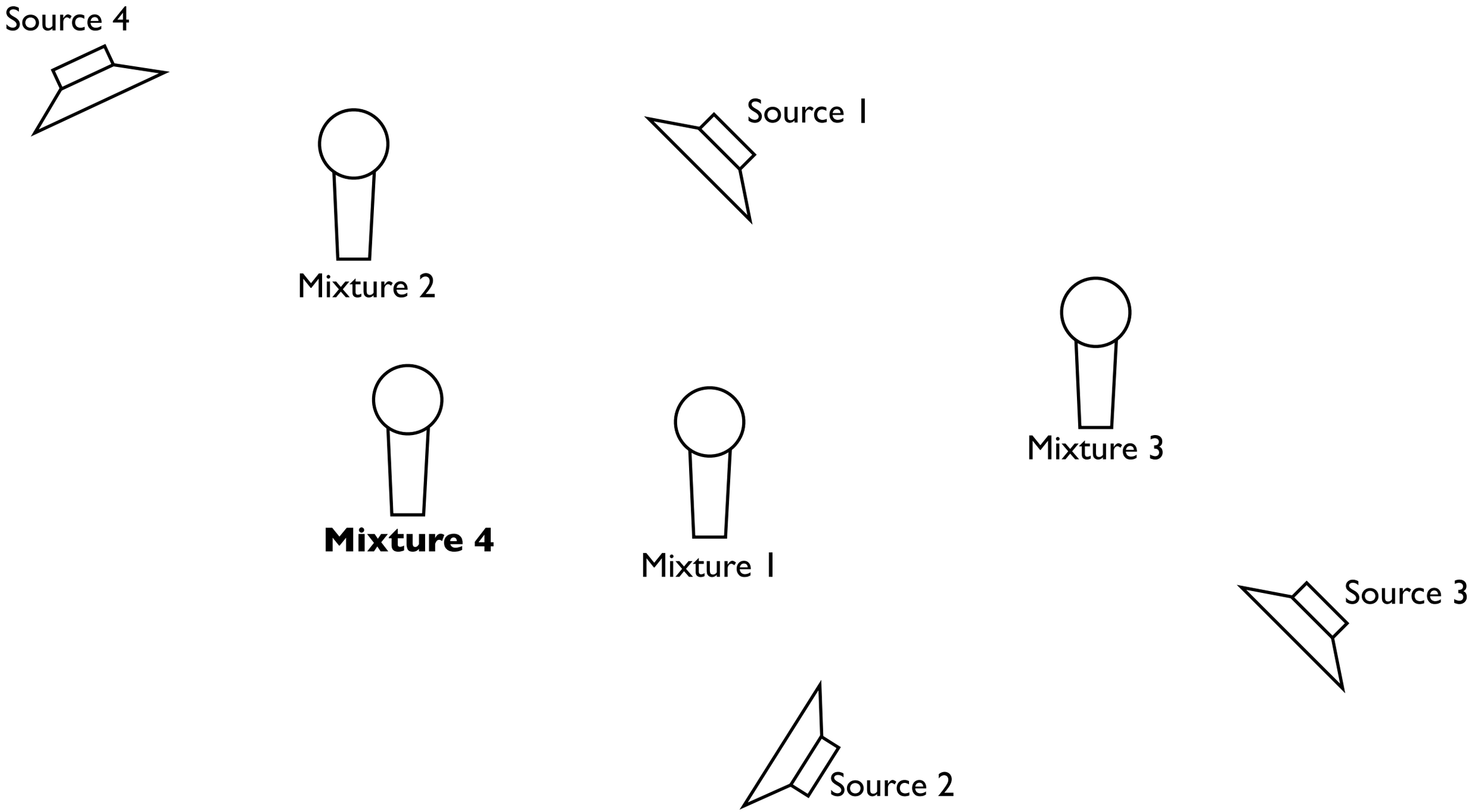


Source 3

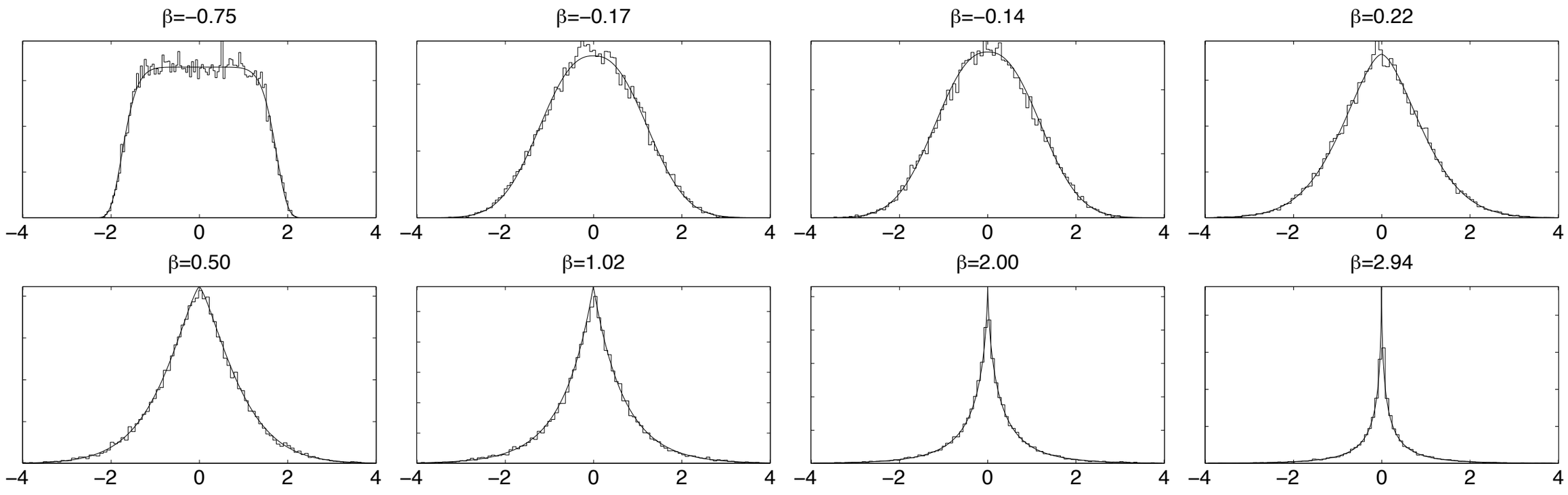


Source 2

Separating mixtures of real sources



Separating mixtures of synthetic sources: 4 sources, 4 mixtures



source	1	2	3	4	5	6	7	8
avg. SNR (dB)	40.48	4.84	4.71	17.29	35.03	43.07	45.85	44.10
std. dev.	1.35	0.39	0.42	2.08	1.49	0.50	1.75	2.00

- Experiment: recover synthetic sources from a random mixing matrix. Repeat 5 times.
- SNR reflects the accuracy of in inferring the mixing matrix.
- The near-Gaussian sources cannot be accurately recovered.

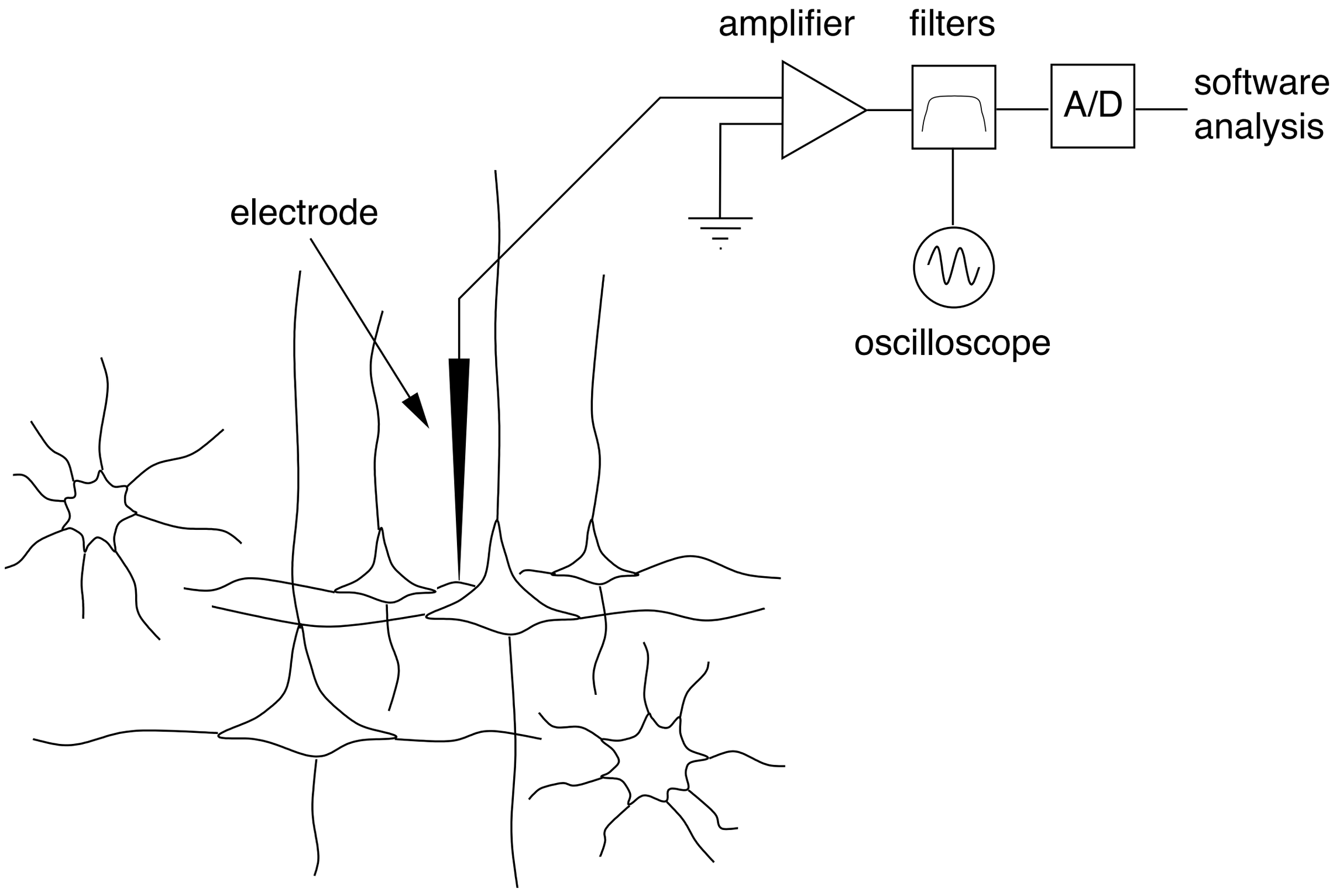
Limitations for computational auditory scene analysis

How do we incorporate these ideas in a computational algorithm?

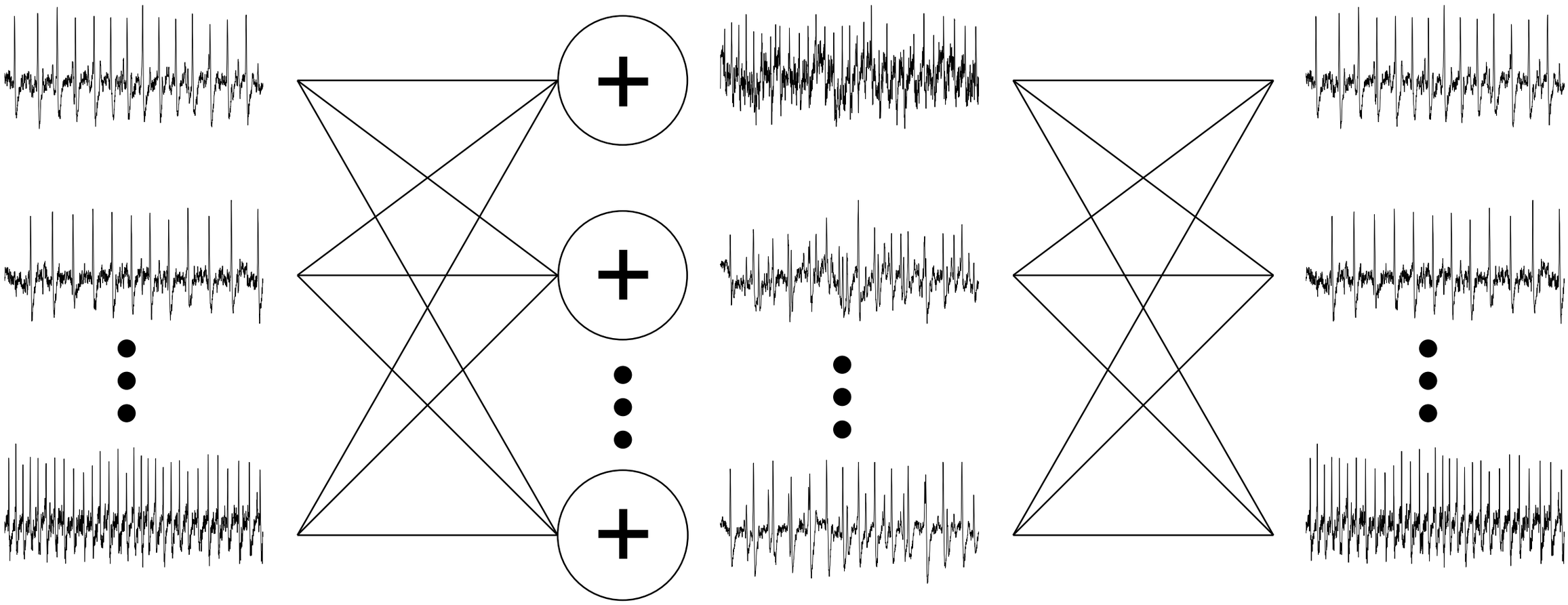
Blind source separation (ICA) only solves special case

- non-Gaussian sources
- linear, stationary mixtures
- equal number of sources and mixtures
- need at least two mixtures

Spike sorting



Blind source separation of multi-channel spike recordings



unknown sources

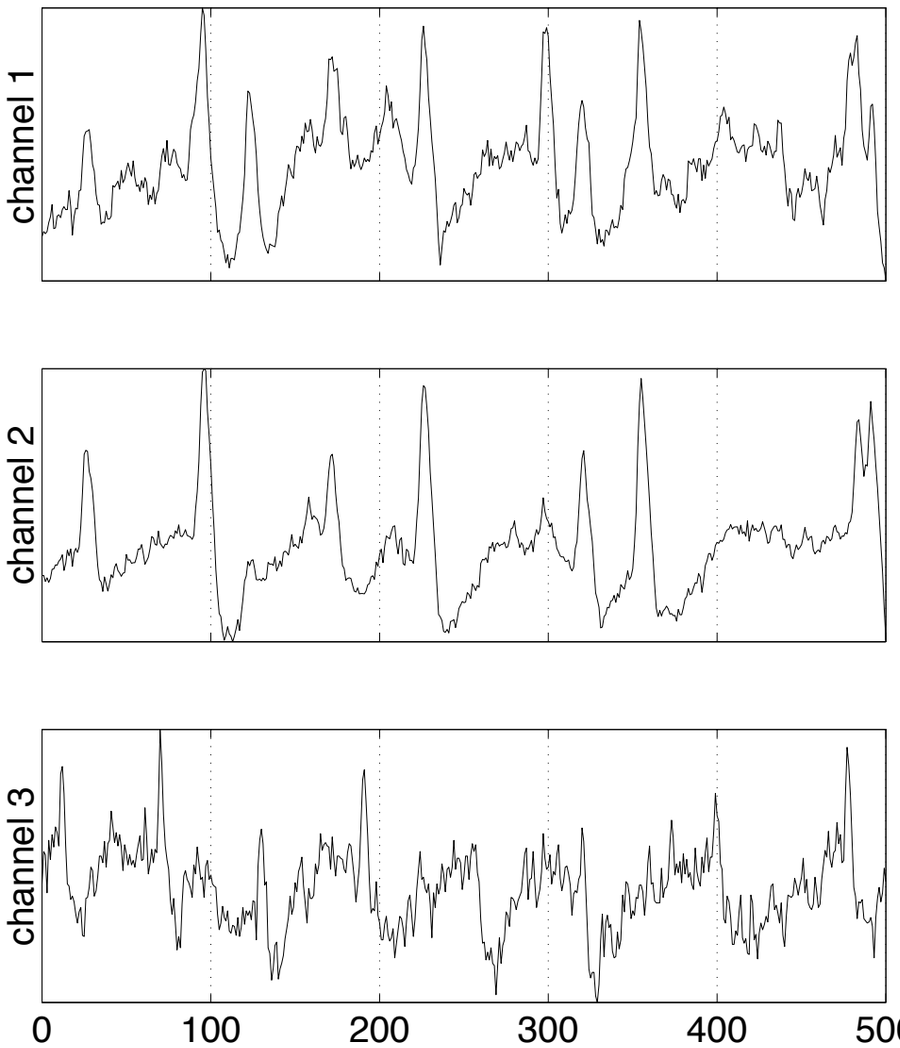
unknown mixing weights

observed mixtures

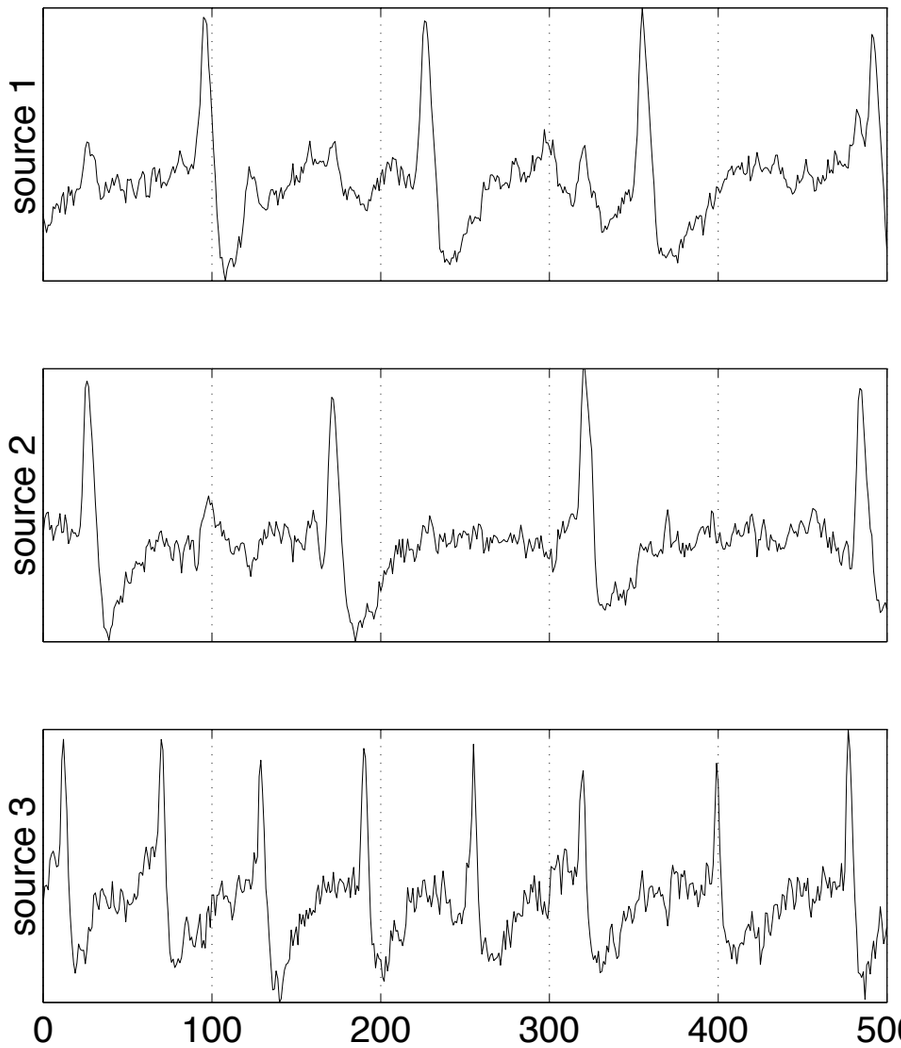
learned unmixing weights

inferred sources

Blind source separation of multi-channel spike recordings

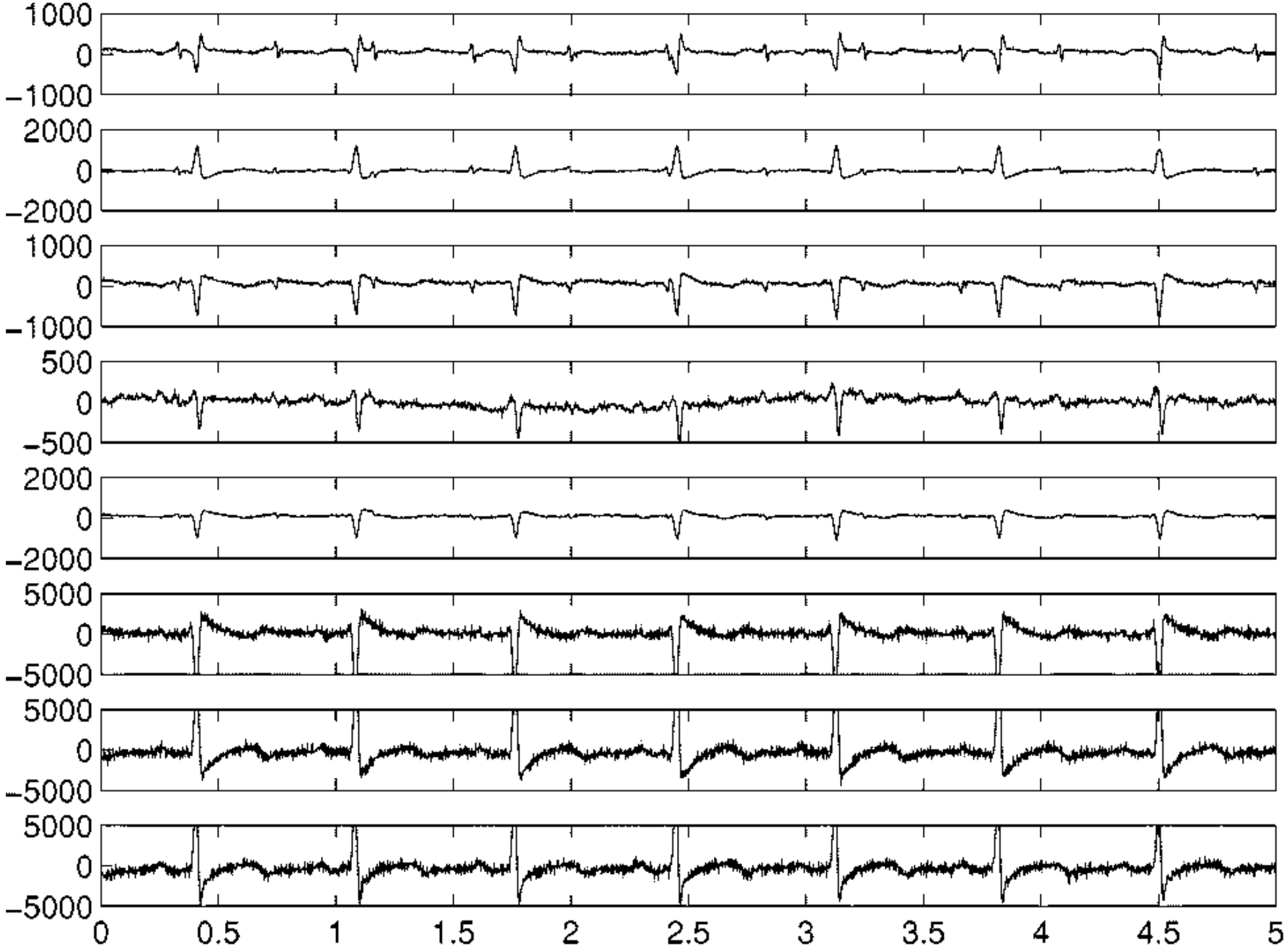


Raw waveforms of an optical recording of voltage sensitive dye in *Tritonia*.



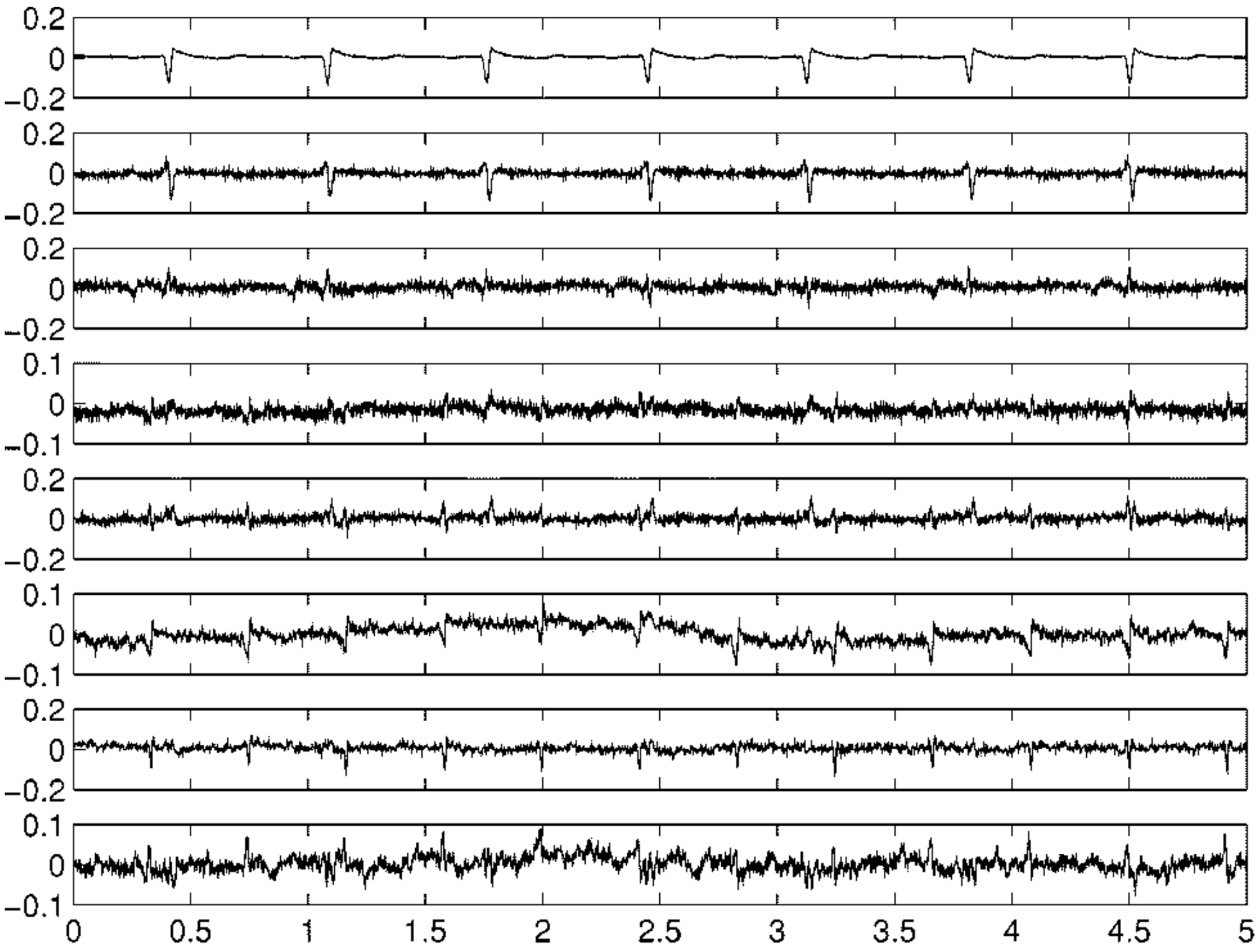
Unmixed sources.

Blind source separation of cardiac rhythms (De Lathauwer et al, 2000)

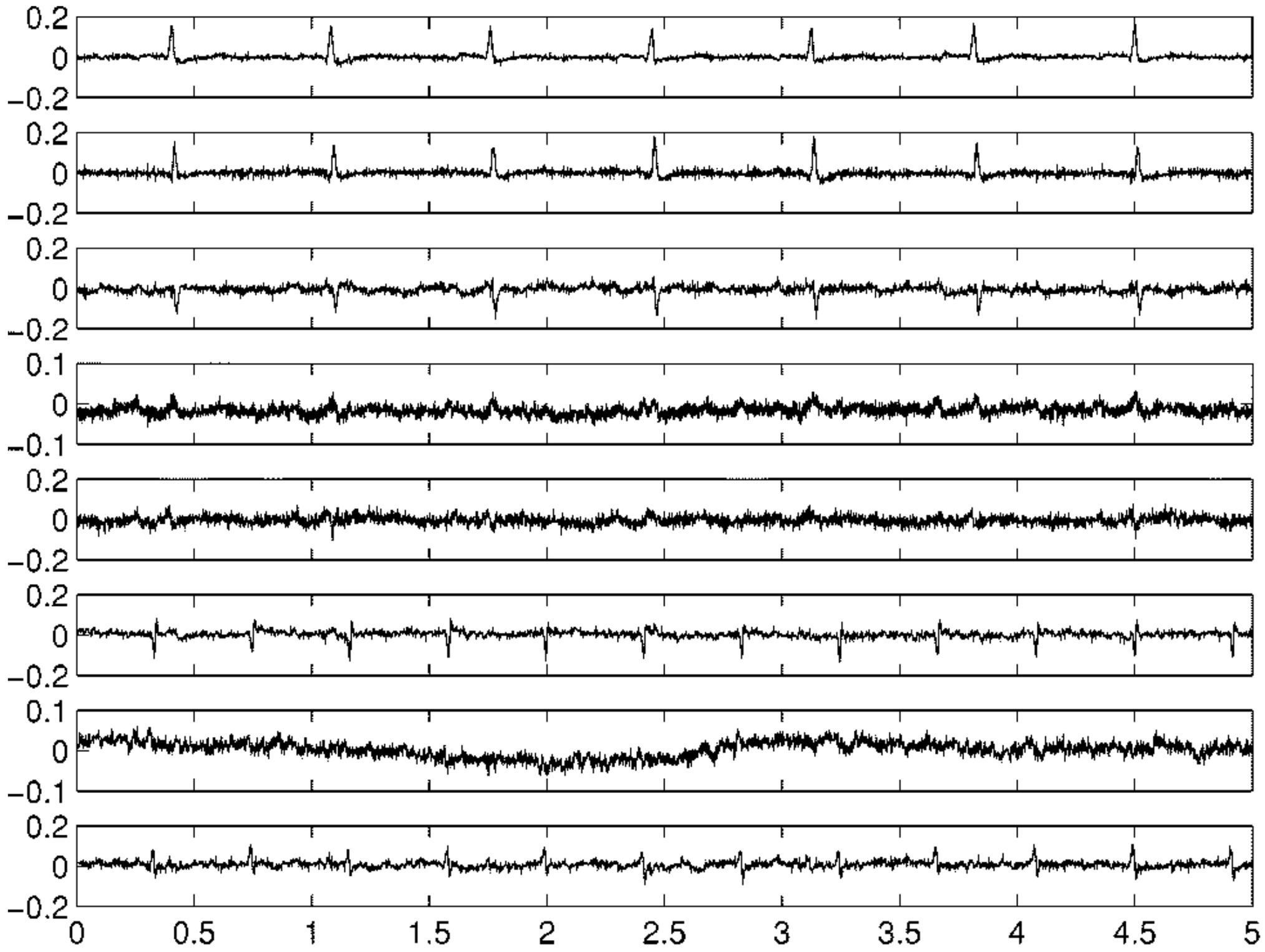


8 channel cutaneous recordings at 500 Hz from pregnant women

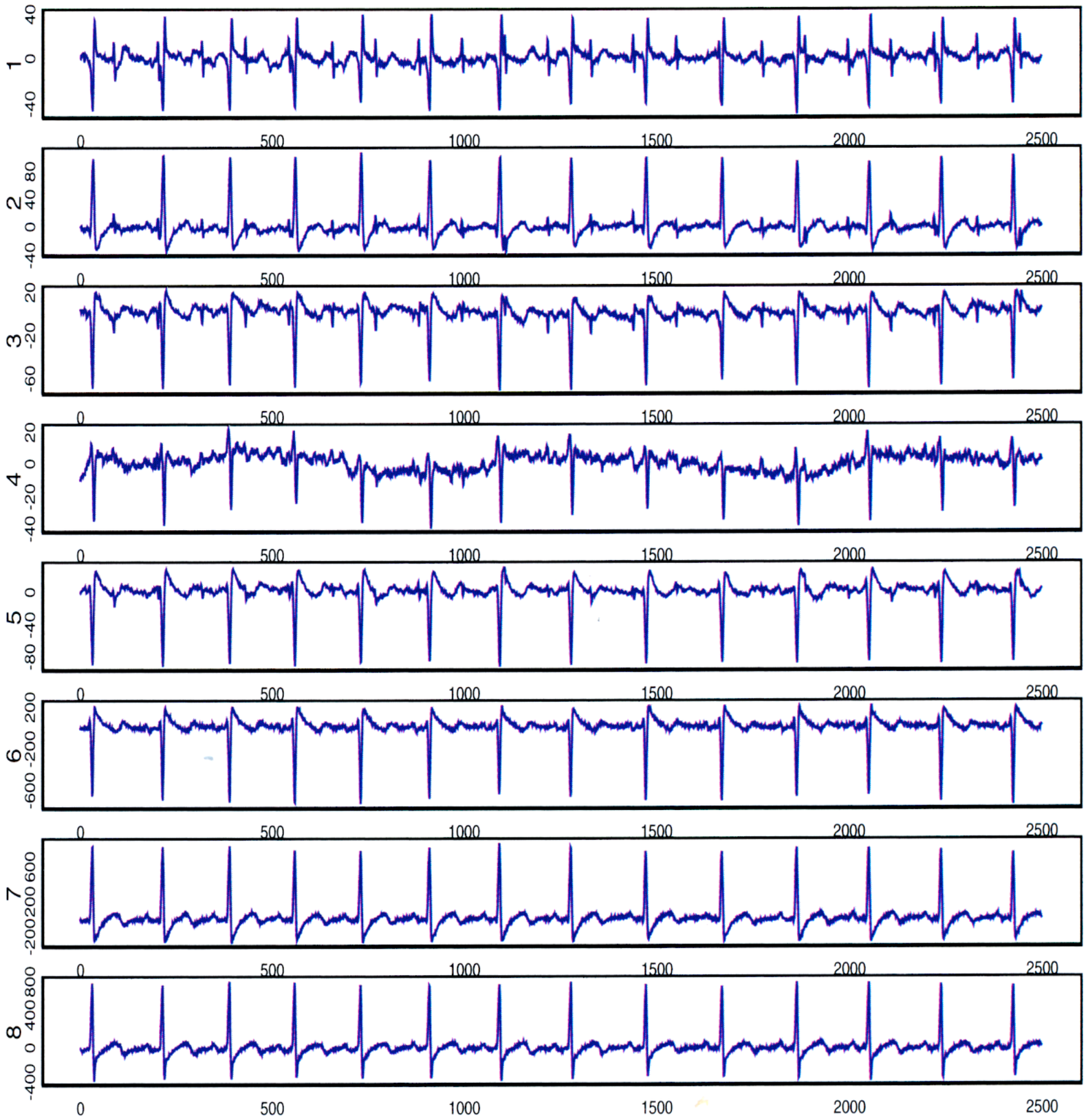
Source estimates by principal component analysis (PCA)



Source estimates by independent component analysis (ICA)

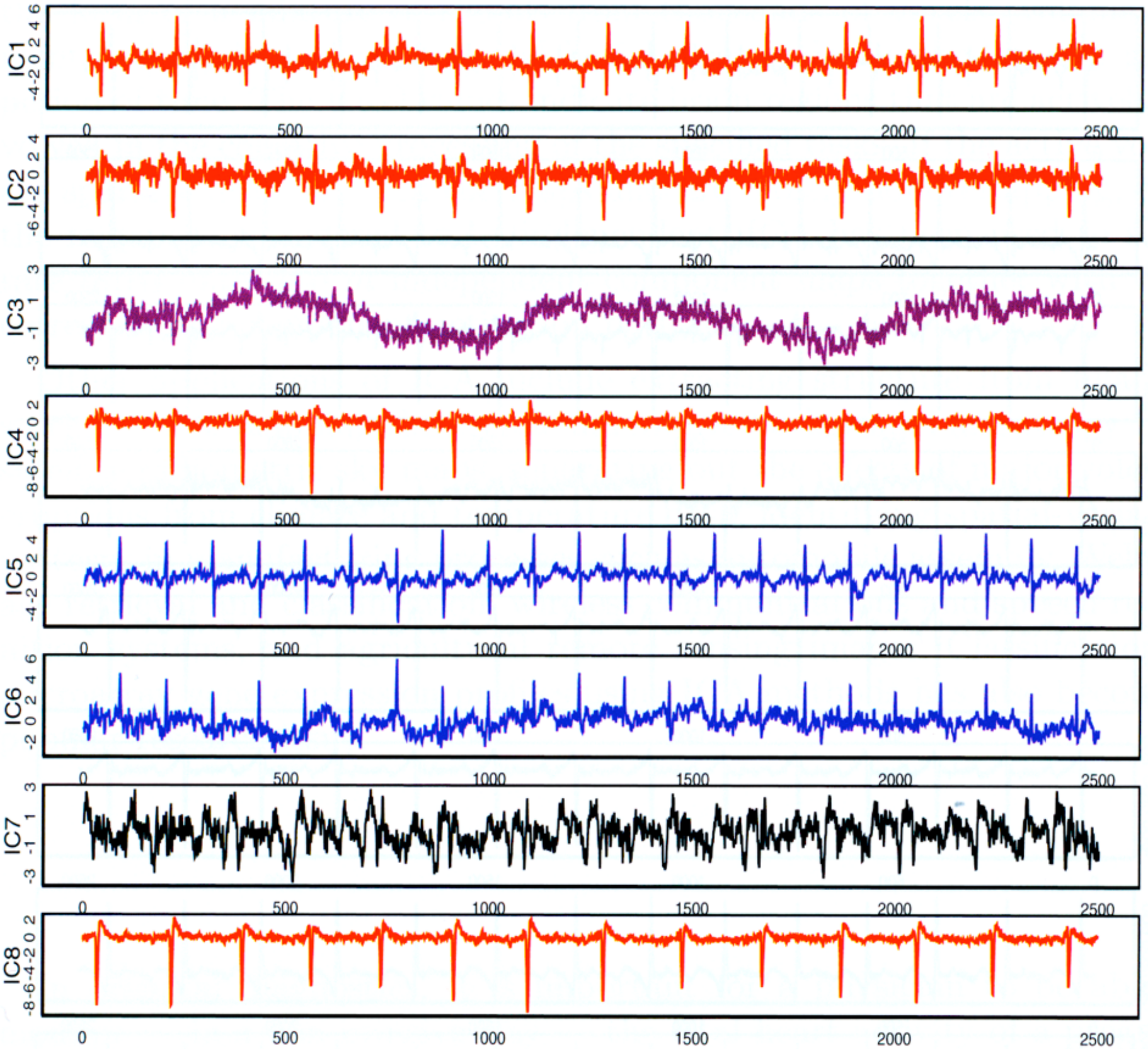


Blind source separation of cardiac rhythms (De Lathauwer et al, 2000)



- 2,500 ECG points from pregnant women
- eight channels of cutaneous data at 500 Hz

Blind source separation of cardiac rhythms (De Lathauwer et al, 2000)



mother

mother

respiration

mother

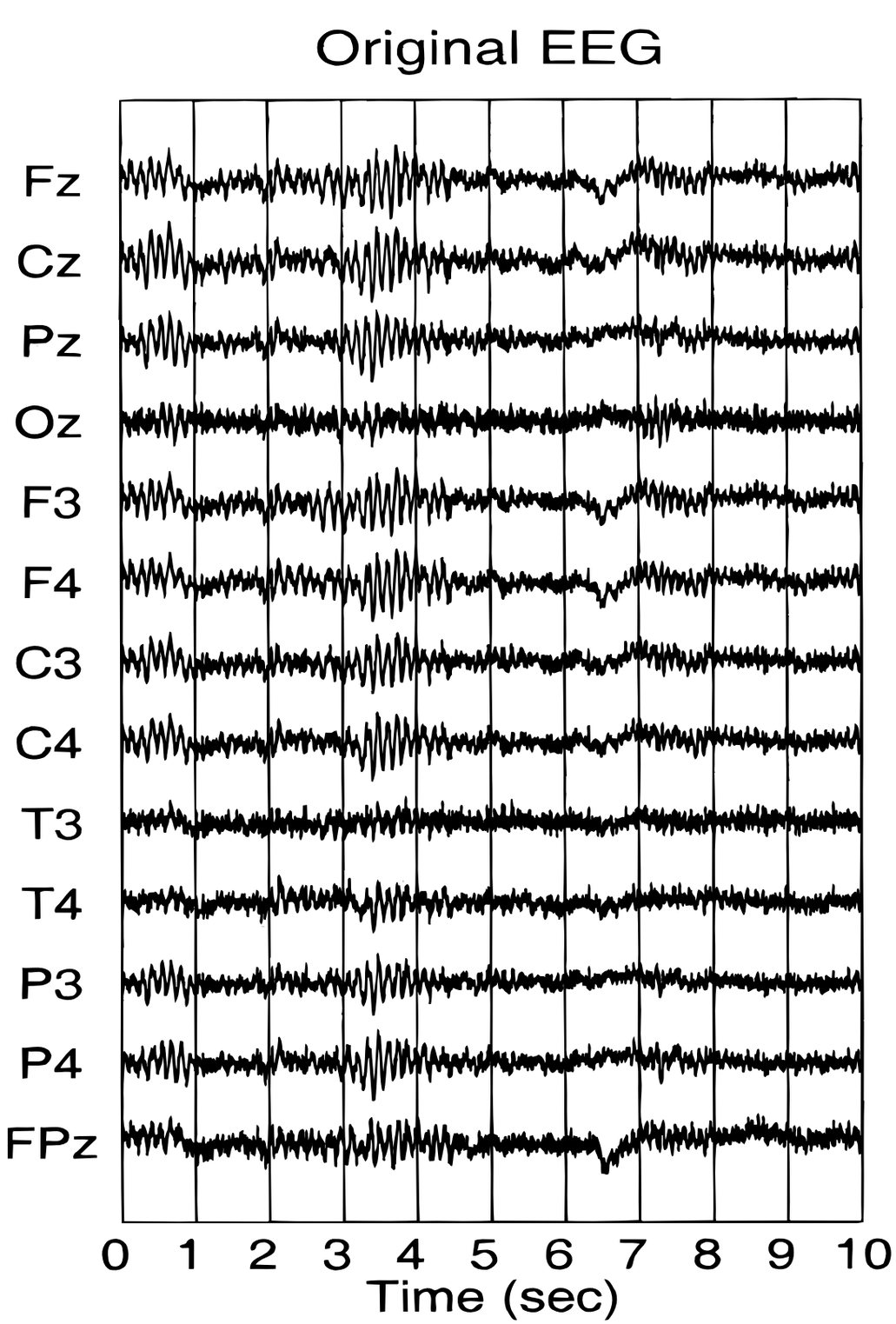
fetus

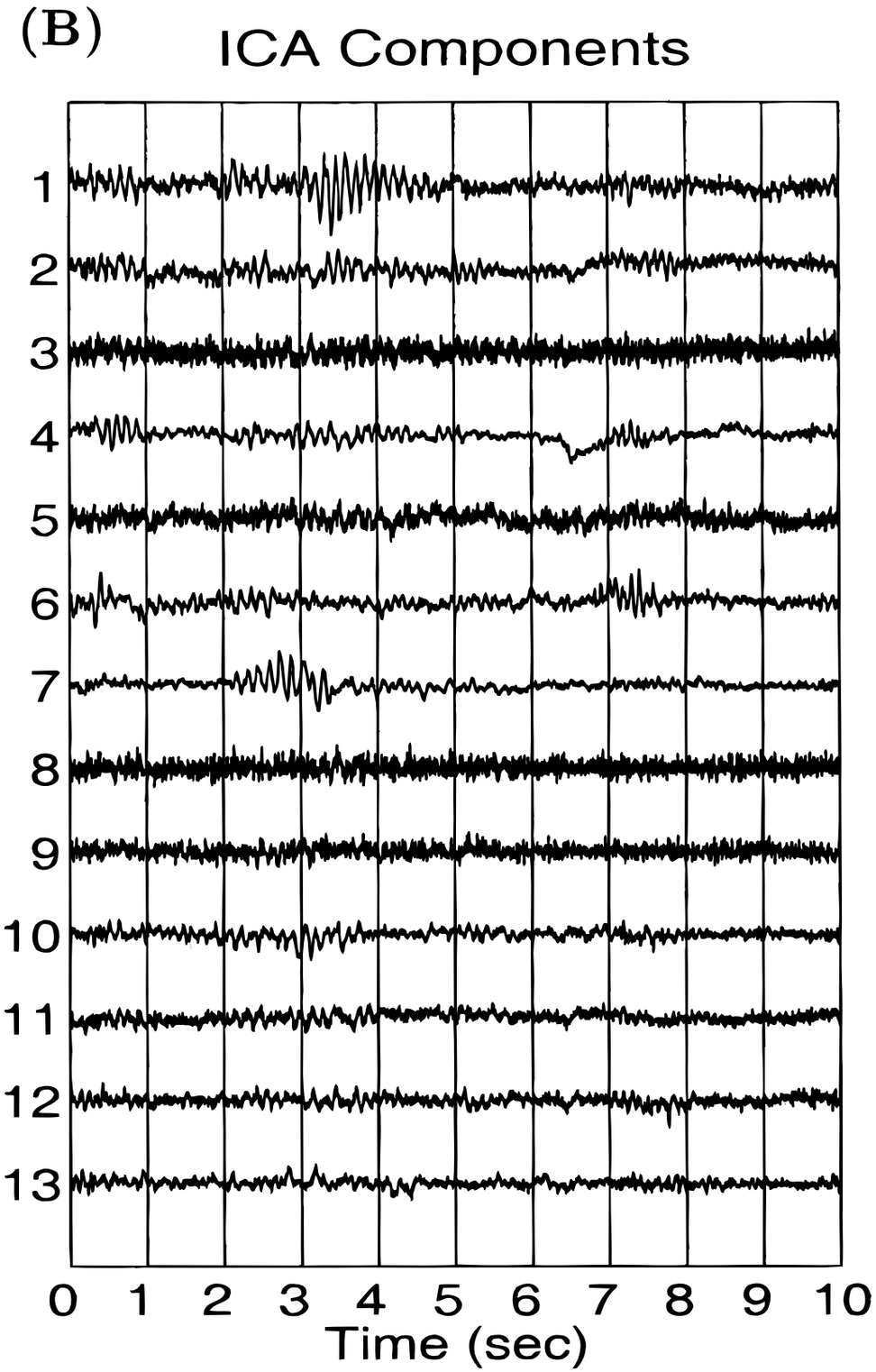
fetus

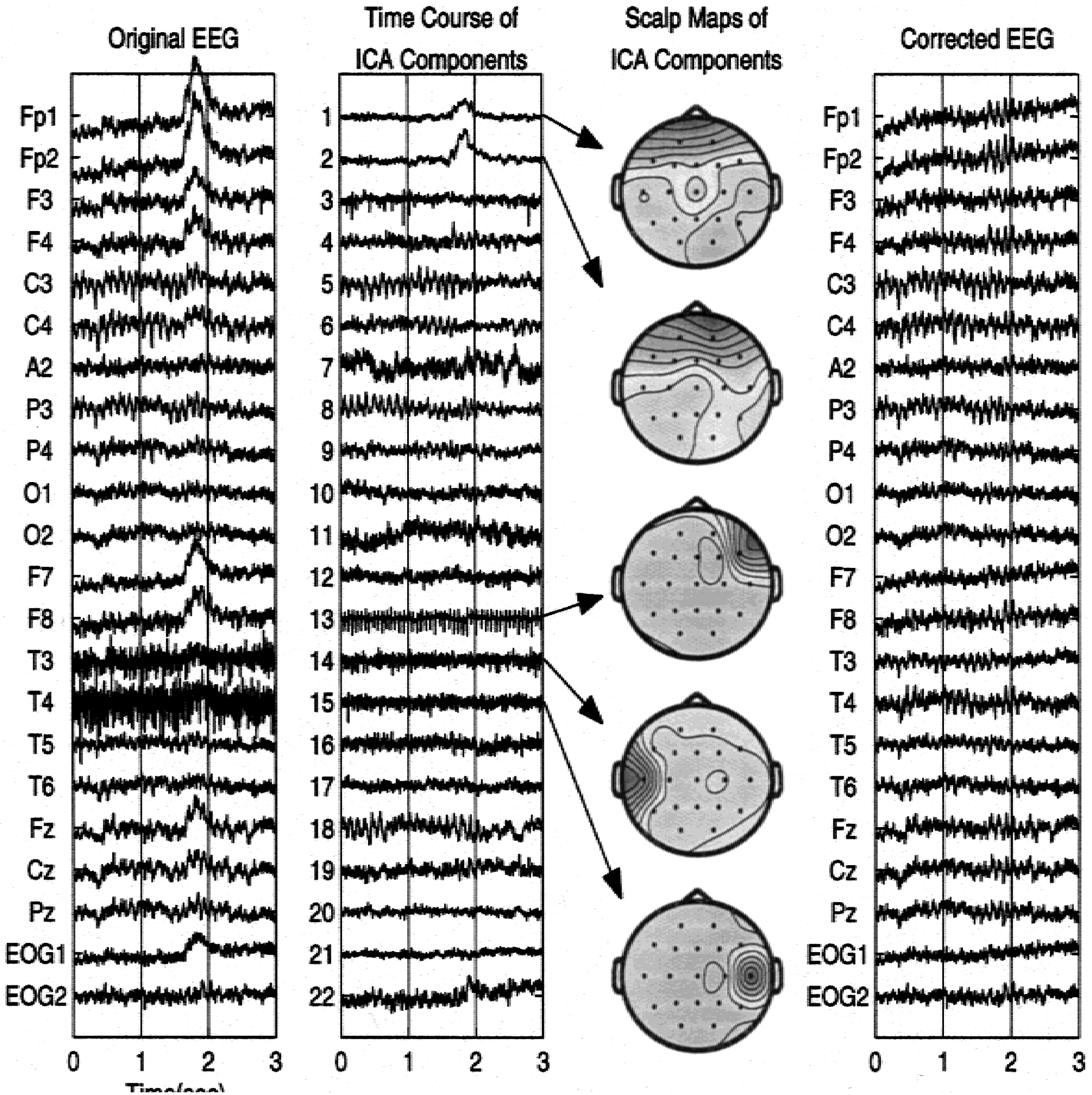
noise

mother

Blind source separation of EEG (T-P Jung et al, 200)






Original EEG
Time Course of
ICA Components
Scalp Maps of
ICA Components
Corrected EEG
Fp1
Fp2
F3
F4
C3
C4
A2
P3
P4
O1
O2
F7
F8
T3
T4
T5
T6
Fz
Cz
Pz
EOG1
EOG2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
0
1
2
3
Time(sec)