Beyond NMF:
Time-Domain Audio Source Separation without Phase Reconstruction

PSDTF

Kazuyoshi Yoshii (AIST, Japan)
Ryota Tomioka (Univ. of Tokyo , Japan)
Daichi Mochihashi (ISM , Japan)
Masataka Goto (AIST , Japan)

2013/11/06 ISMIR 2013

Summary

If you do not care about

computational time,

you could forget NMF and

use our method PSDTF

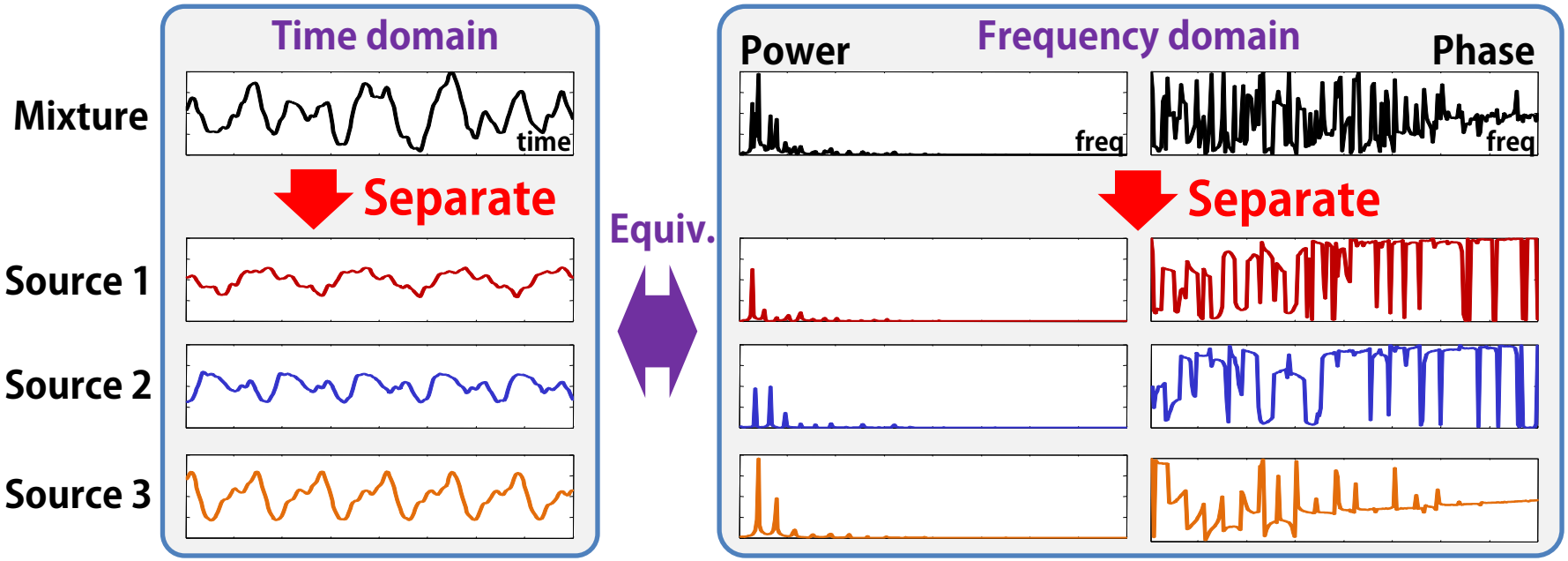
for audio source separation

We provide MATLAB source codes!

Overview

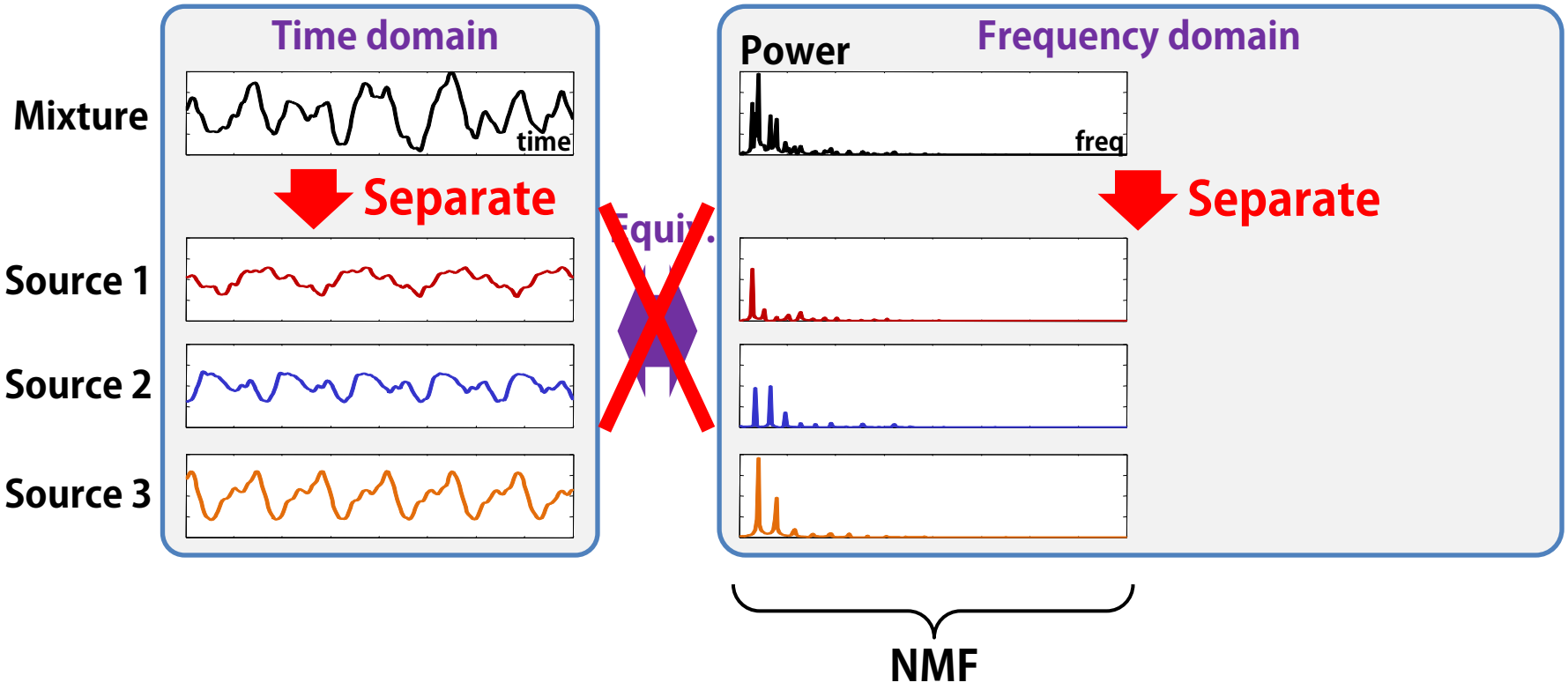
- Our goal: high-quality source separation of monaural audio signals

Time-domain separation
is equivalent to
frequency-domain separation



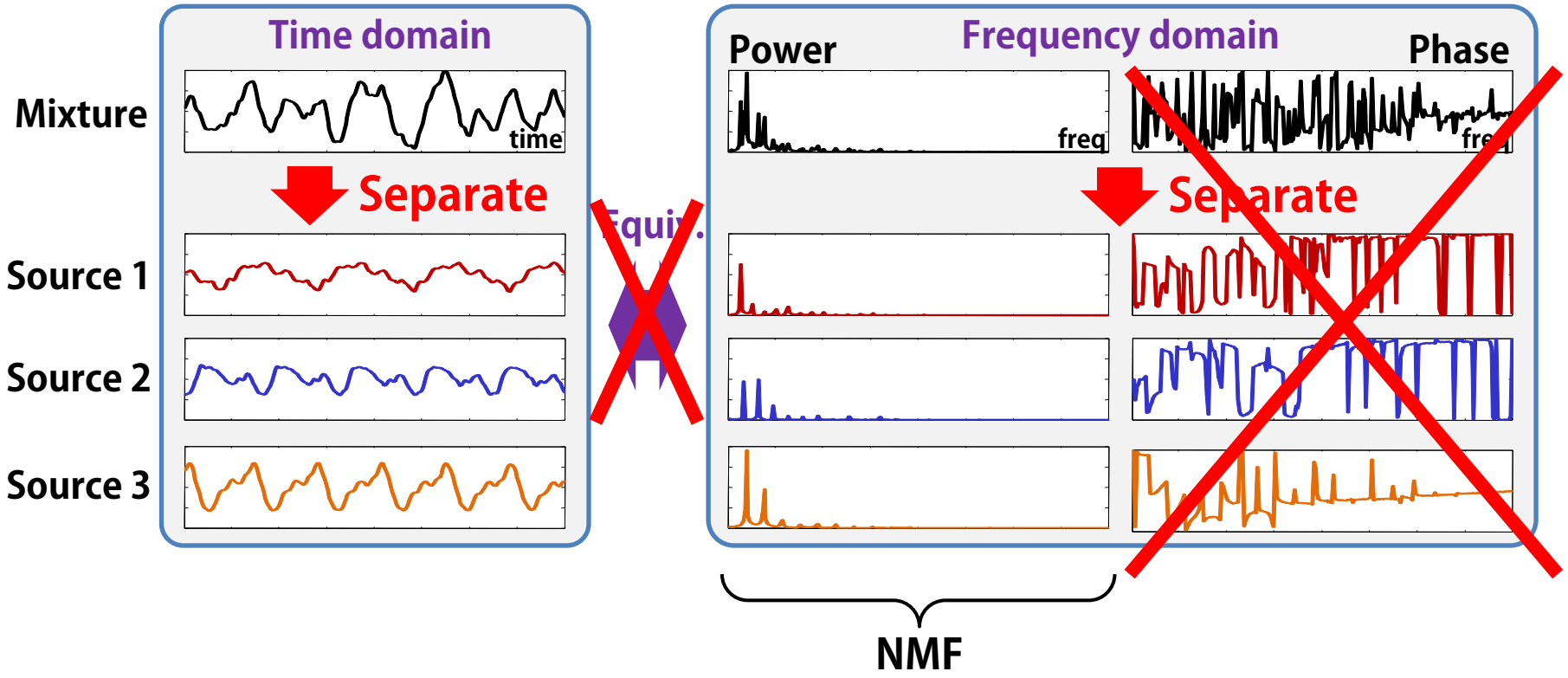
Overview

- Our goal: high-quality source separation of monaural audio signals
  - Nonnegative matrix factorization (NMF) is very popular
    - Frequency-domain decomposition for power spectrogram



Overview

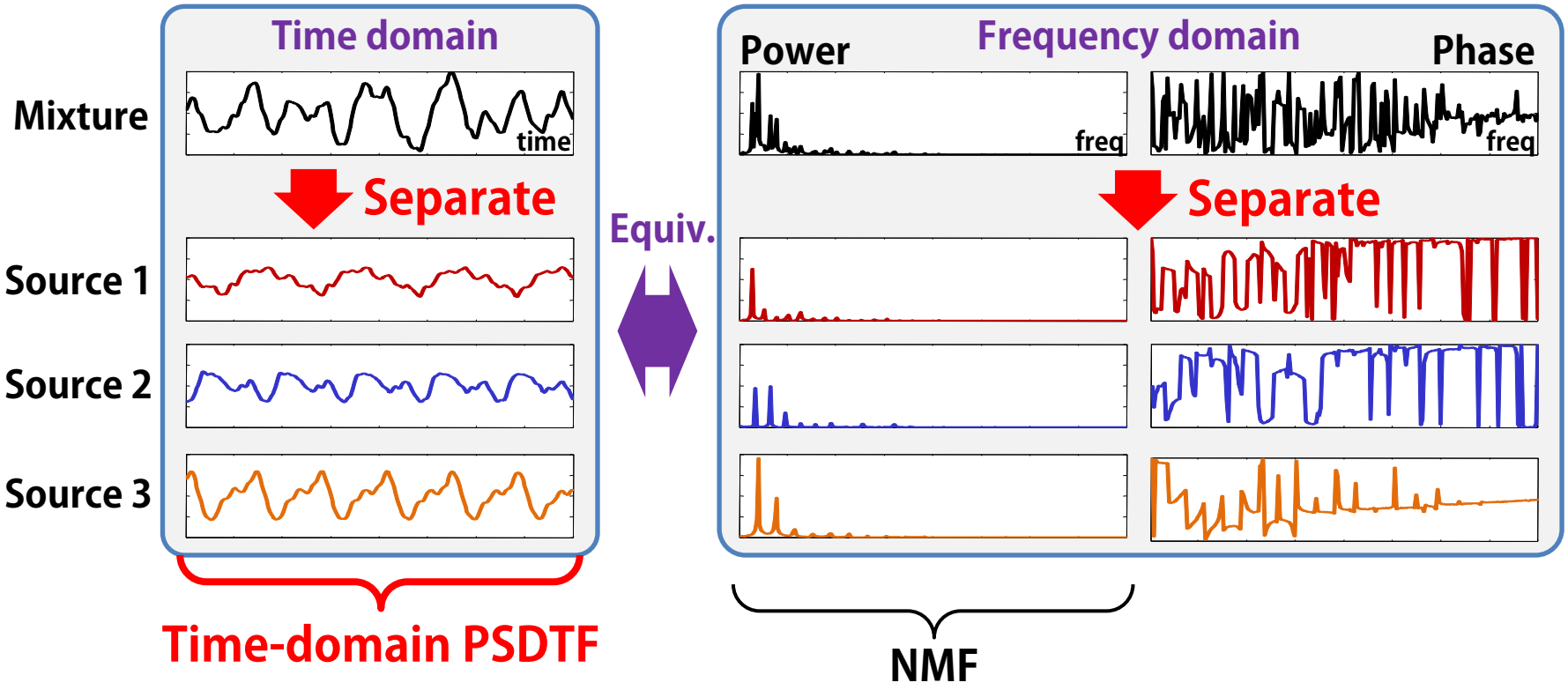
- Our goal: high-quality source separation of monaural audio signals
  - Nonnegative matrix factorization (NMF) is very popular
    - Frequency-domain decomposition for power spectrogram
    - Ignore phase → Cannot separate time-domain signals well



Overview

- Our goal: high-quality source separation of monaural audio signals

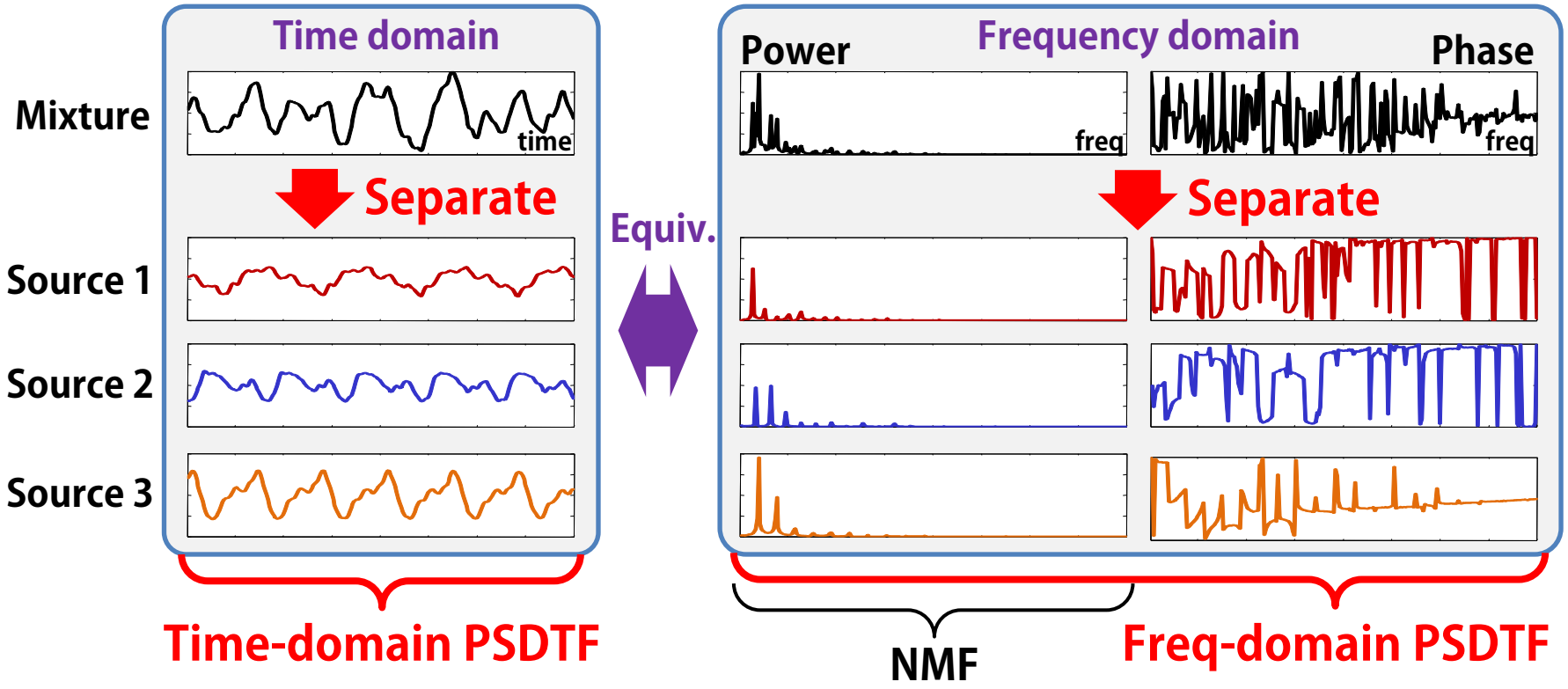
Our PSDTF can solve this problem by time-domain separation!



Overview

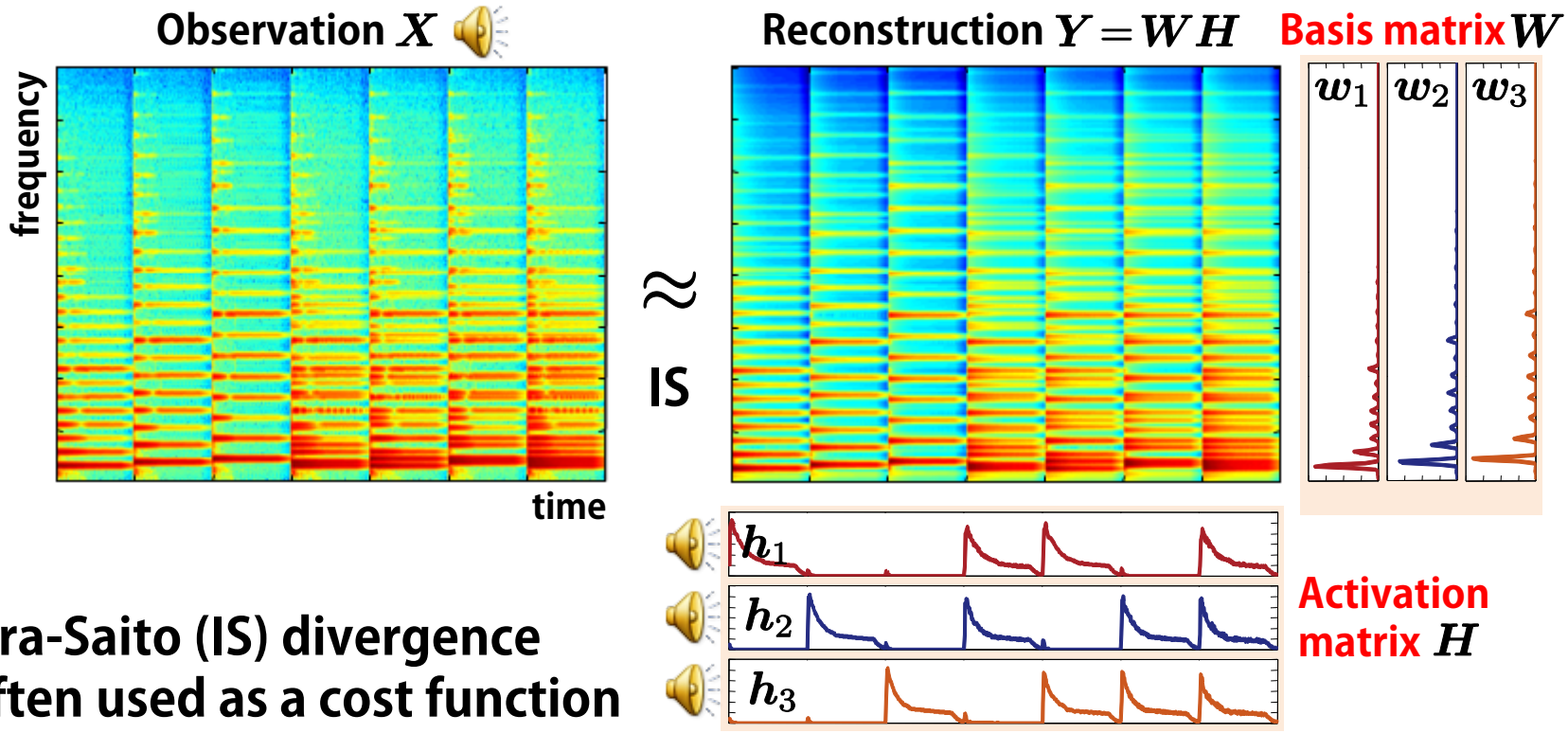
- Our goal: high-quality source separation of monaural audio signals

Time-domain PSDTF has an equivalent form of frequency-domain PSDTF considering the phase



Nonnegative Matrix Factorization (NMF)

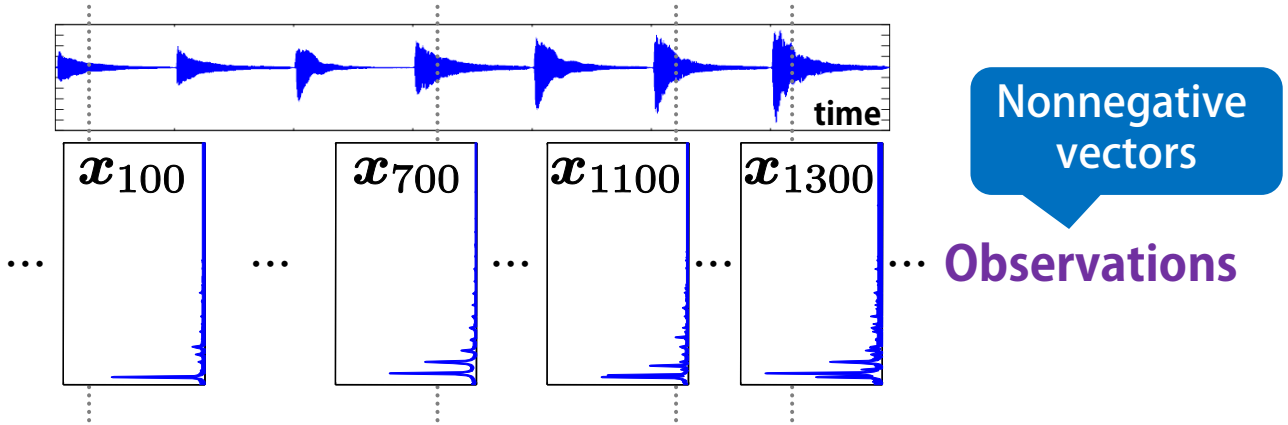
- A nonnegative matrix (power spectrogram X) is approximated by the product of two nonnegative matrices WH
  - Given the observation matrix X, NMF estimates W and H so that their product Y can be similar to X



Itakura-Saito (IS) divergence is often used as a cost function

Nonnegative Matrix Factorization (NMF)

- Nonnegative vectors are approximated by convex combinations of fewer nonnegative vectors

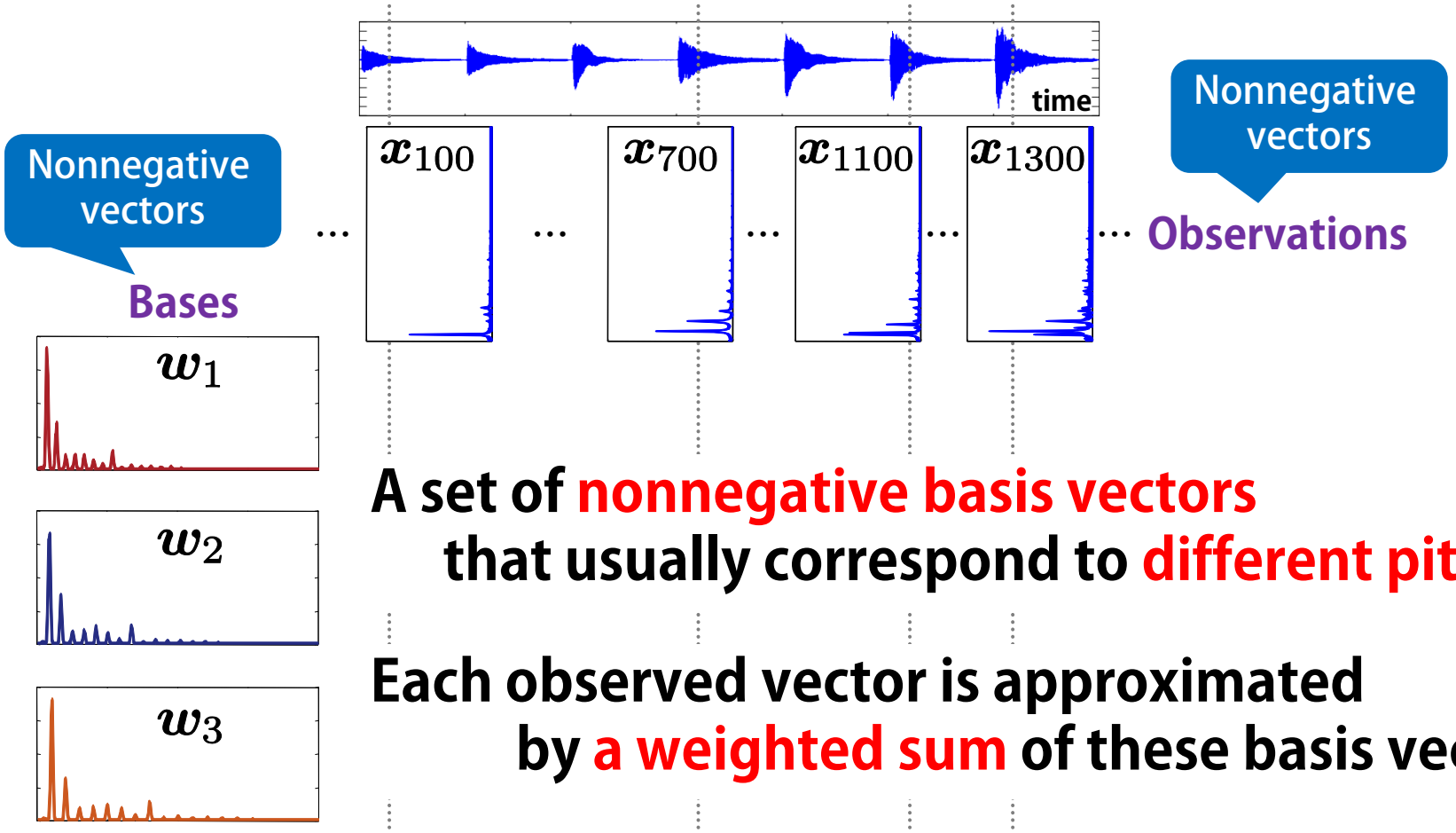


A series of nonnegative observed vectors along the time

Each vector corresponds to the power spectrogram of a locally observed mixture signal at each frame

Nonnegative Matrix Factorization (NMF)

- Nonnegative vectors are approximated by convex combinations of fewer nonnegative vectors

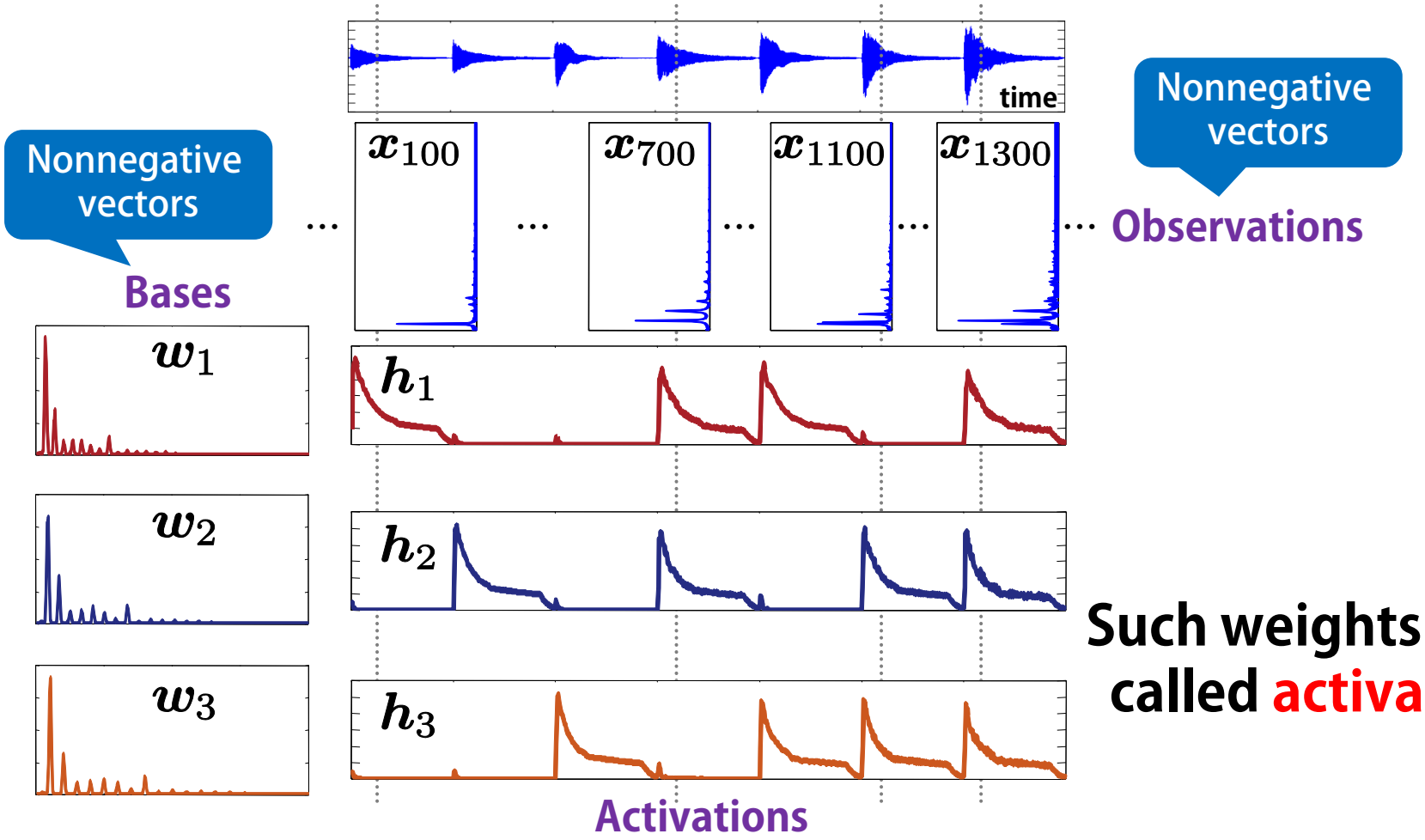


A set of nonnegative basis vectors that usually correspond to different pitches

Each observed vector is approximated by a weighted sum of these basis vectors

Nonnegative Matrix Factorization (NMF)

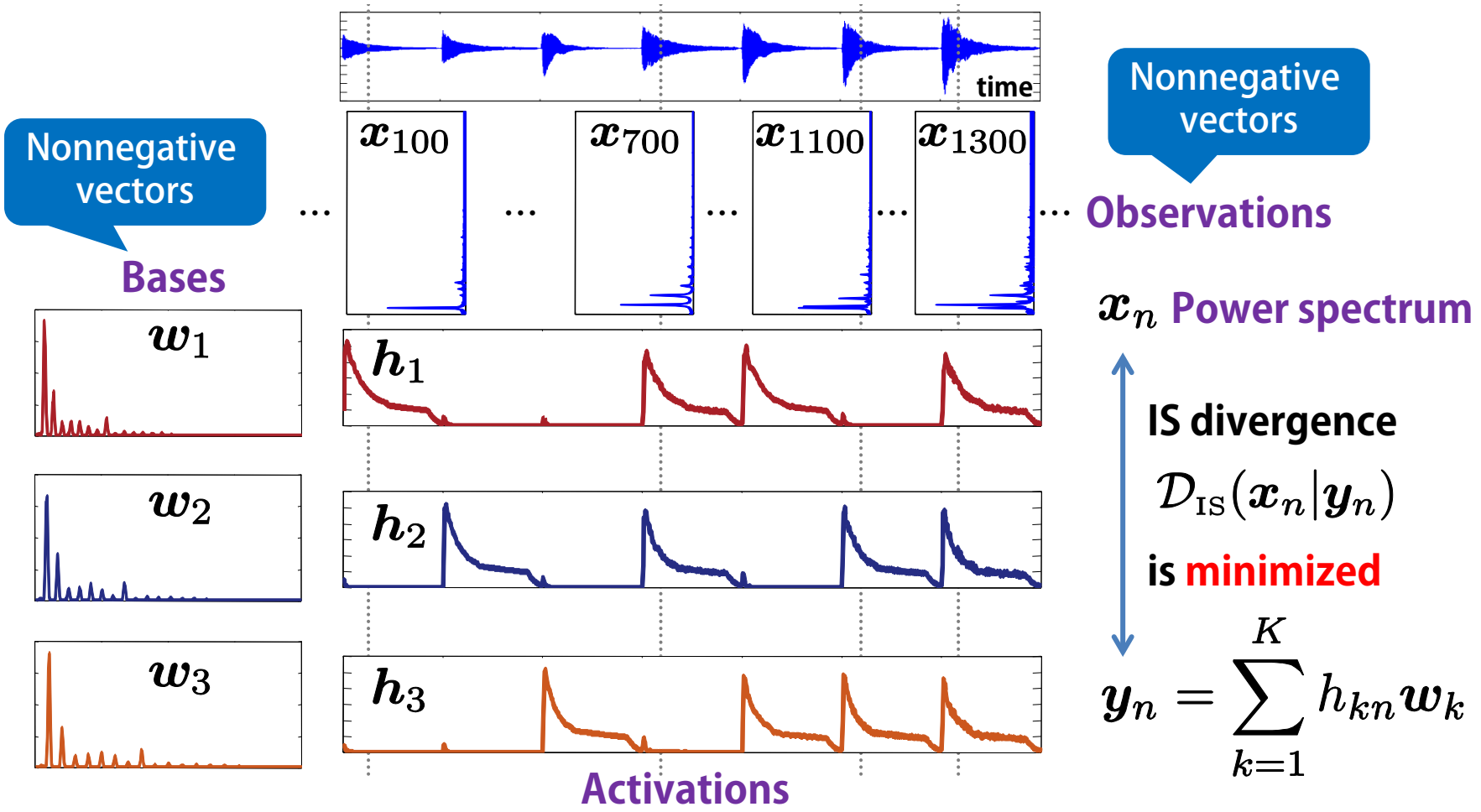
- Nonnegative vectors are approximated by convex combinations of fewer nonnegative vectors



Such weights are called activations

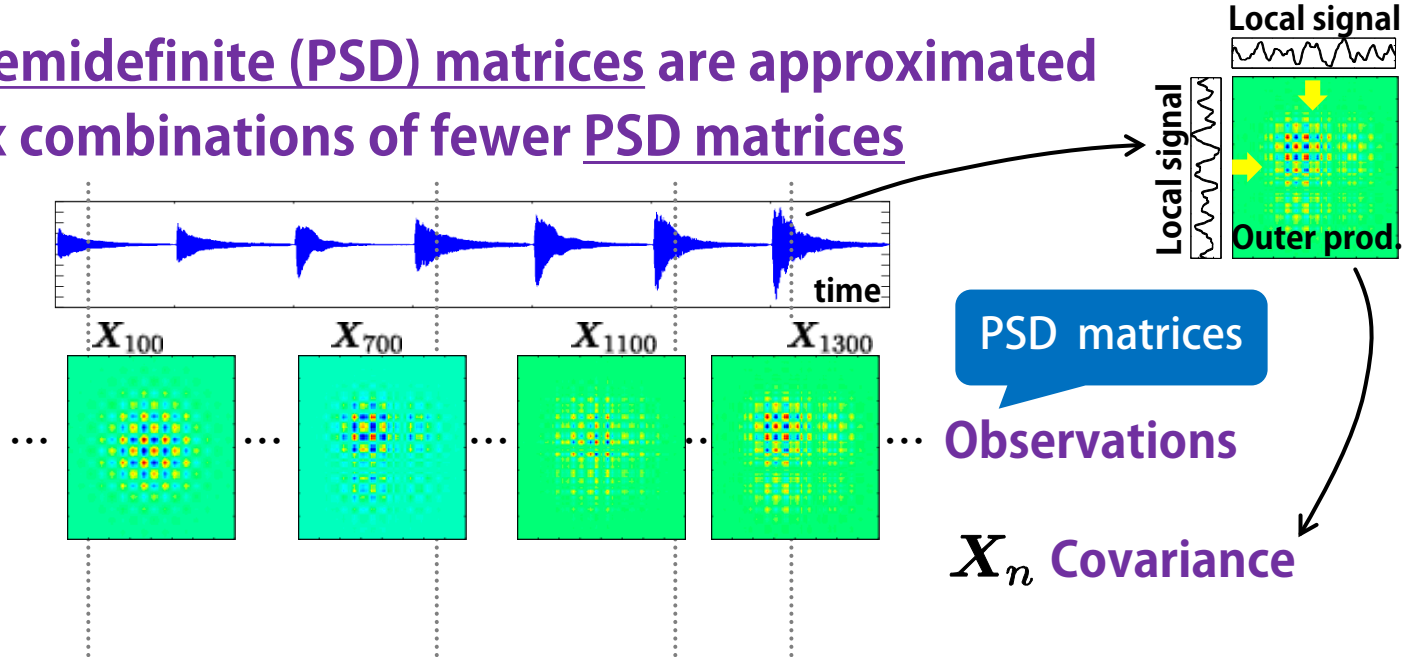
Nonnegative Matrix Factorization (NMF)

The goal of NMF is to estimate basis and activation vectors so that the total reconstruction error can be minimized



Positive SemiDefinite Tensor Factorization (PSDTF)

- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices

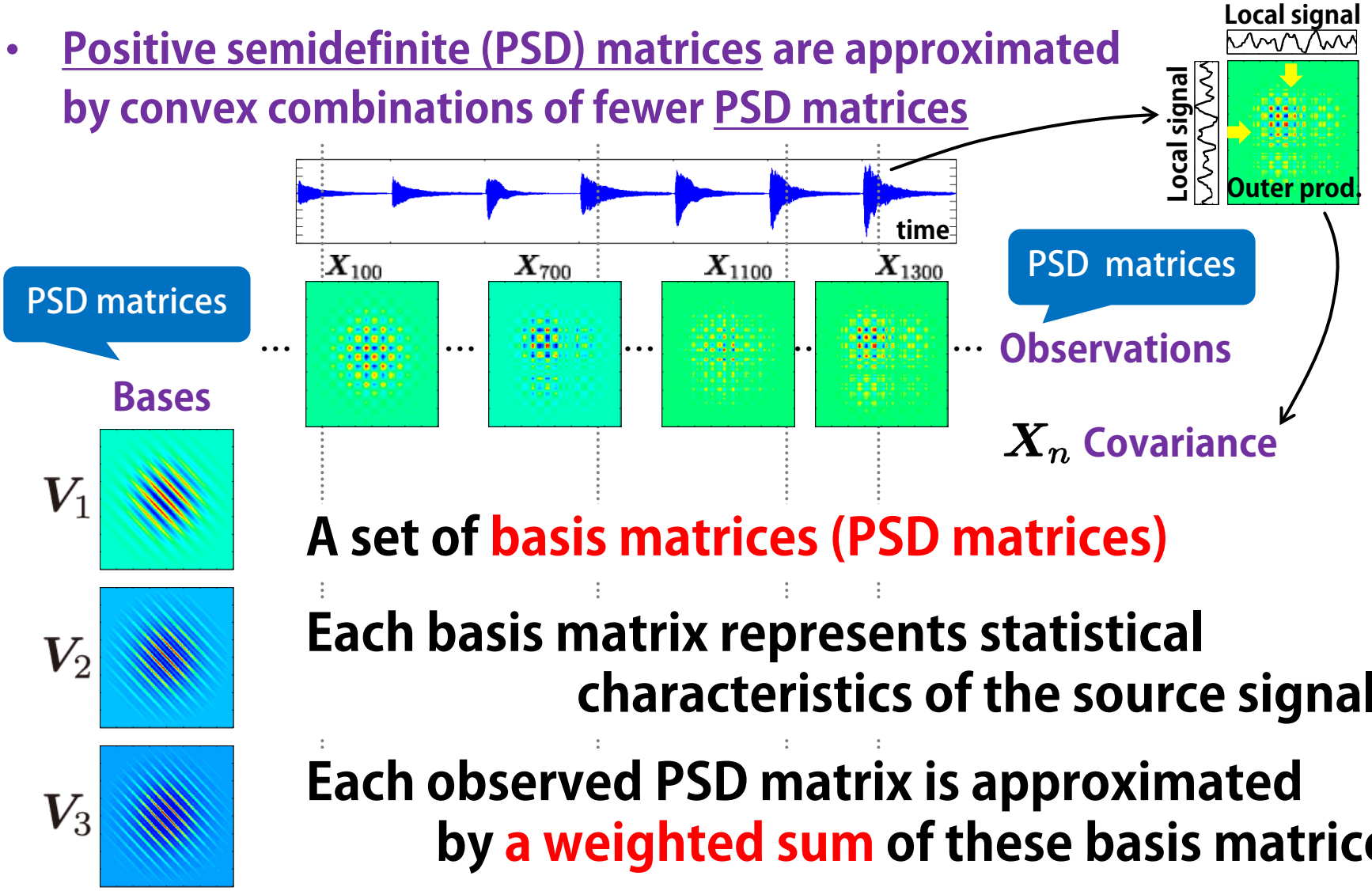


The observed PSD matrices are a set of local covariance matrices calculated from a locally observed mixture signal at each frame

By calculating the outer product of the local signal, we can make the PSD matrix X_n

Positive SemiDefinite Tensor Factorization (PSDTF)

- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices



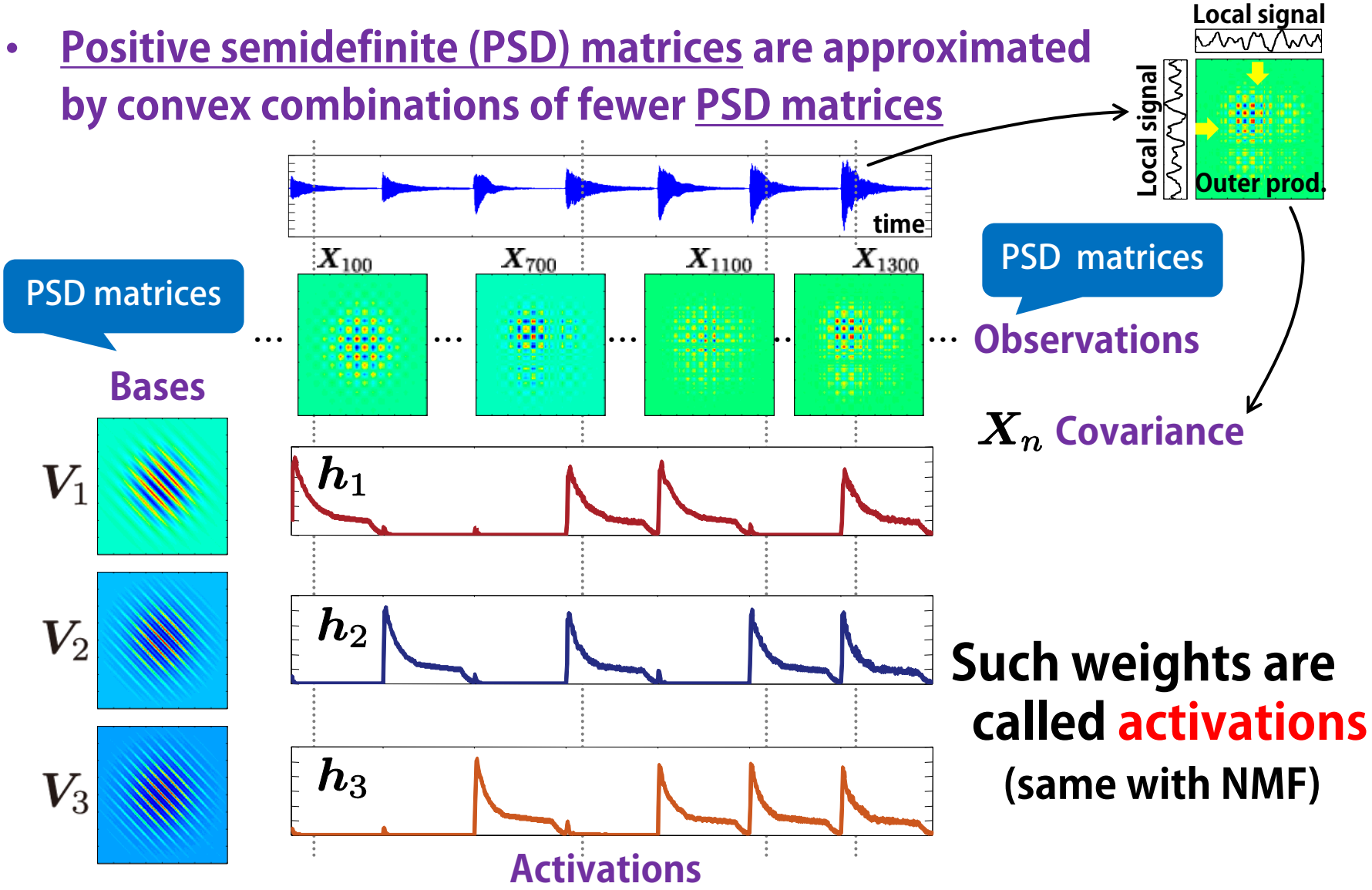
A set of basis matrices (PSD matrices)

Each basis matrix represents statistical characteristics of the source signal

Each observed PSD matrix is approximated by a weighted sum of these basis matrices

Positive SemiDefinite Tensor Factorization (PSDTF)

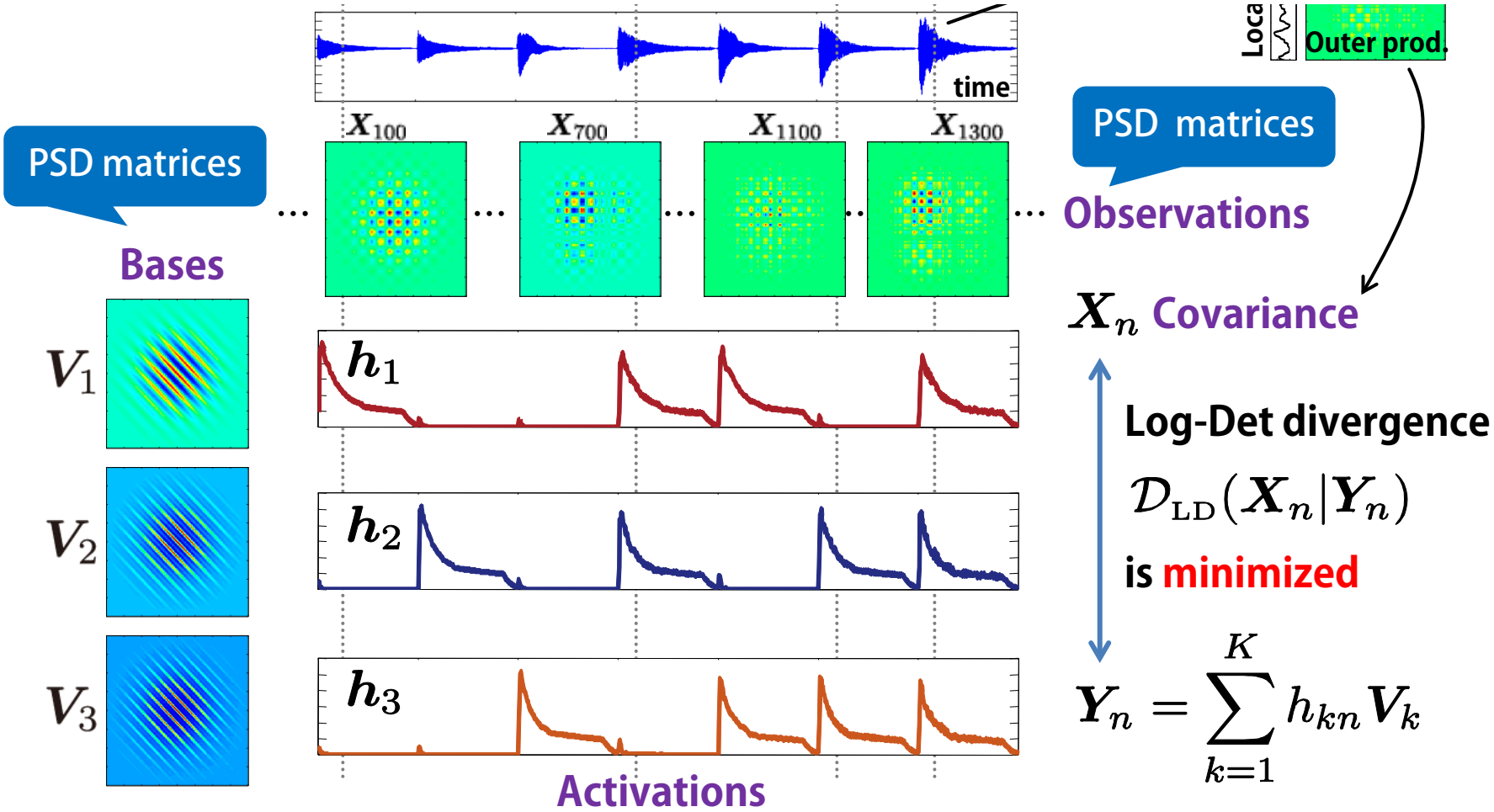
- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices



Such weights are called activations (same with NMF)

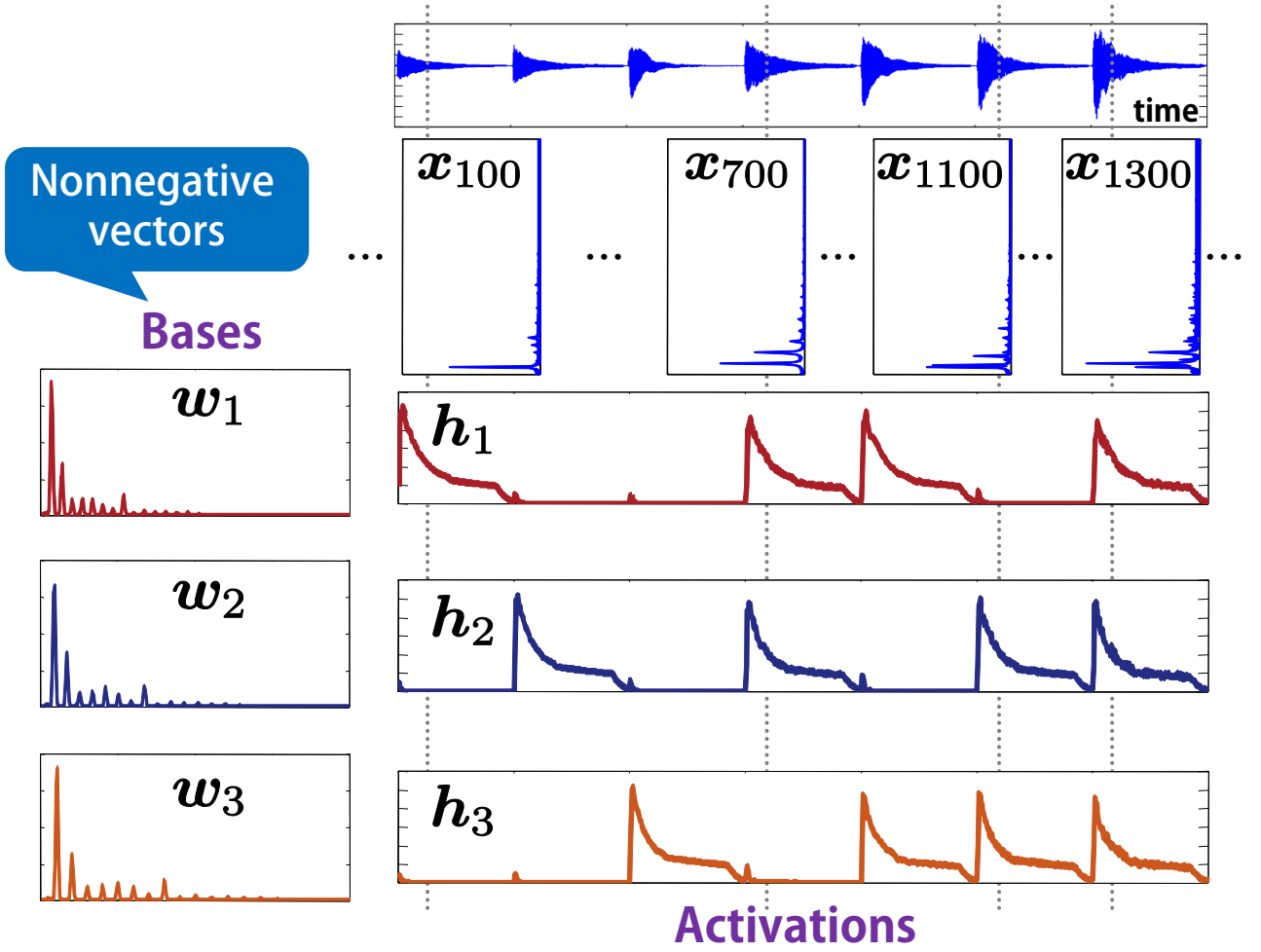
Positive SemiDefinite Tensor Factorization (PSDTF)

The goal of PSDTF is to estimate basis matrices and activation vectors so that the total reconstruction error can be minimized



Nonnegative Matrix Factorization (NMF)

- Nonnegative vectors are approximated by convex combinations of fewer nonnegative vectors



x_n Power spectrum

IS divergence

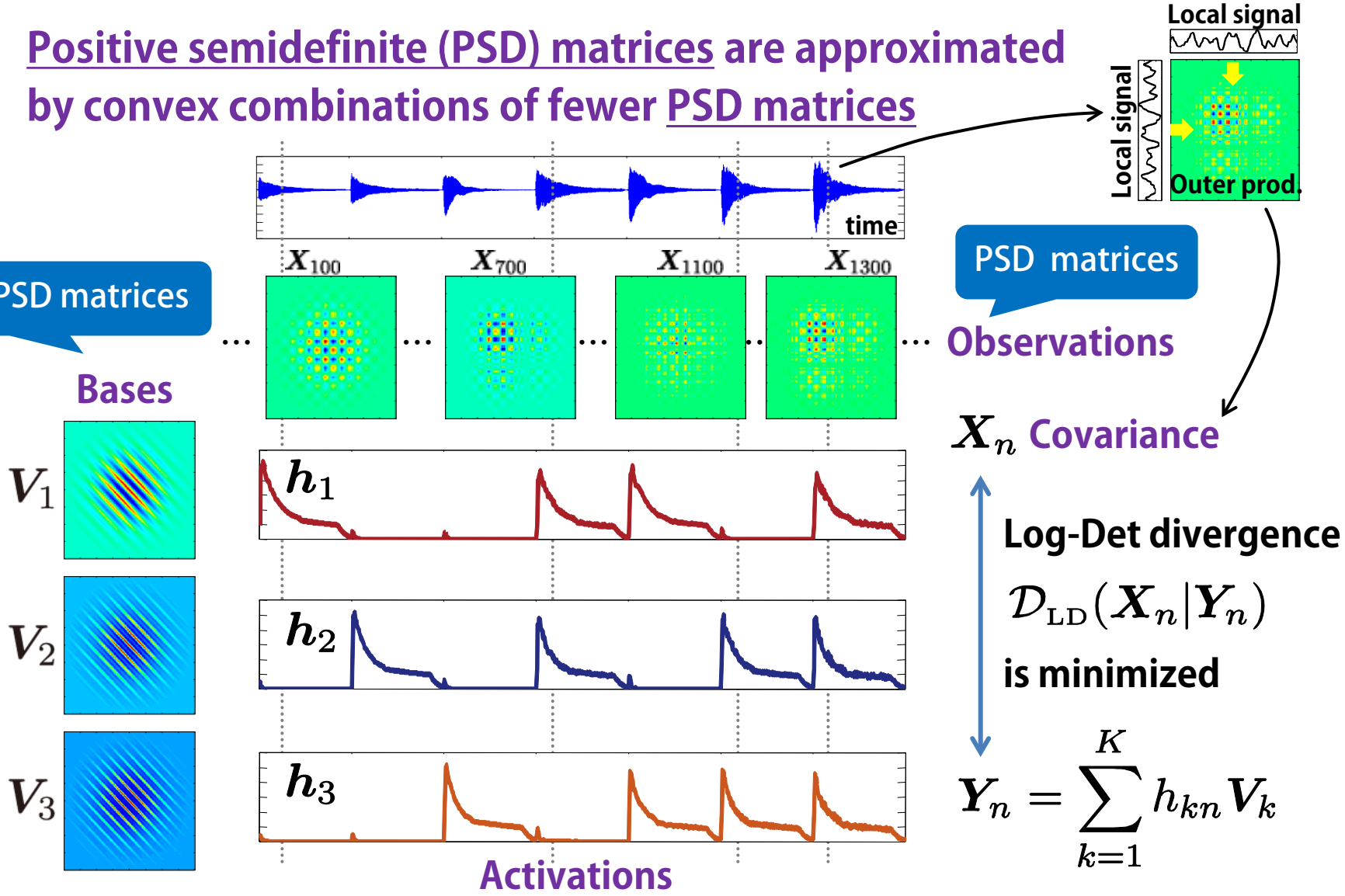
$\mathcal{D}_{\mathrm{IS}}(x_n|y_n)$

is minimized

$$y_n = \sum_{k=1}^{K} h_{kn} w_k$$

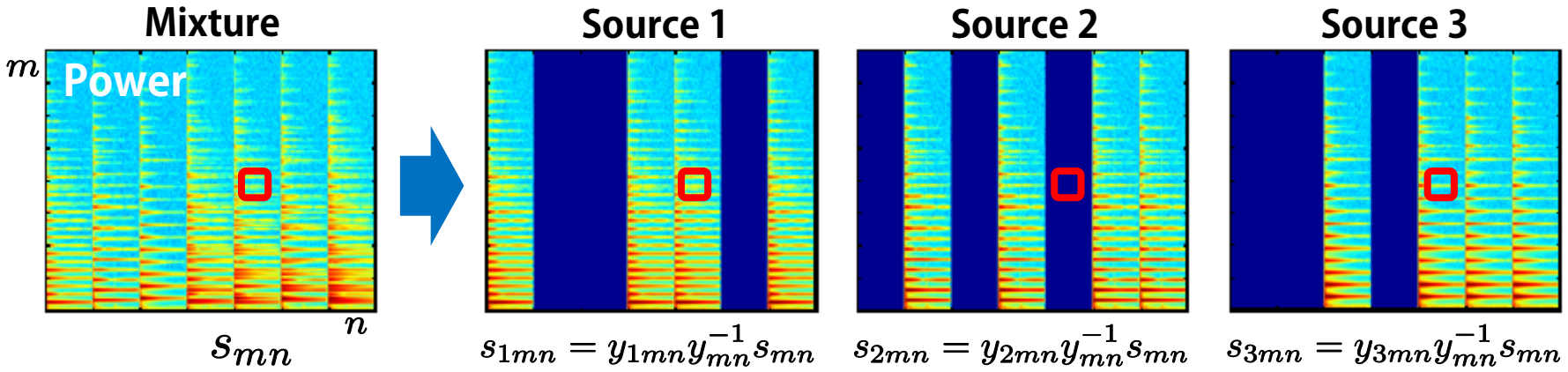
Positive SemiDefinite Tensor Factorization (PSDTF)

- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices



Source Separation based on NMF

- **Wiener filtering based on NMF results** $s_{kmn} = y_{kmn} y_{mn}^{-1} s_{mn}$
  - **Element-wise decomposition of nonnegative power spectrogram**

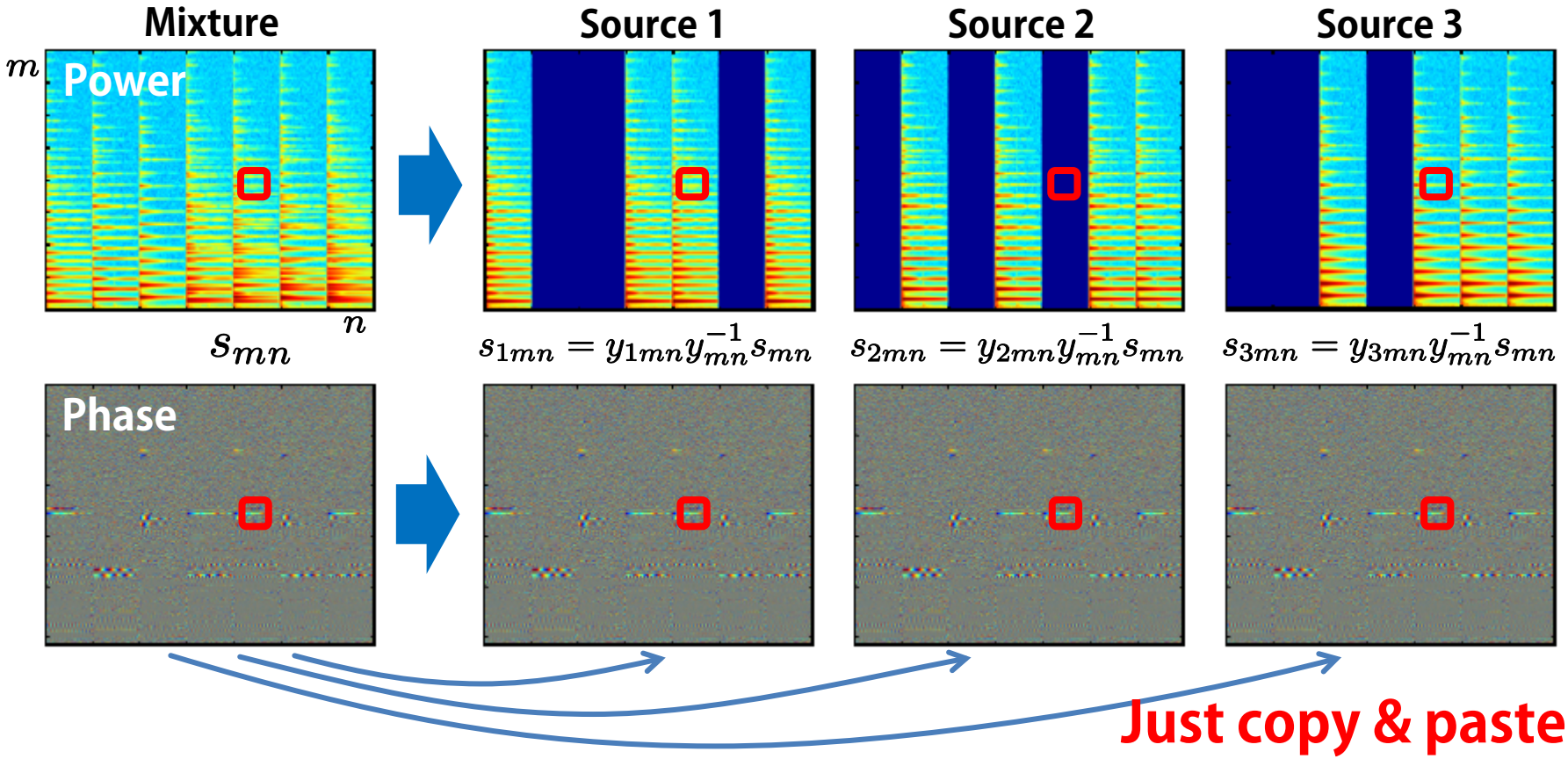


As shown in this equation, Wiener filtering can be regarded as a simple power partition function

The power of each element s_{mn} is distributed into different sources $s_{1mn}, s_{2mn}, s_{3mn}$

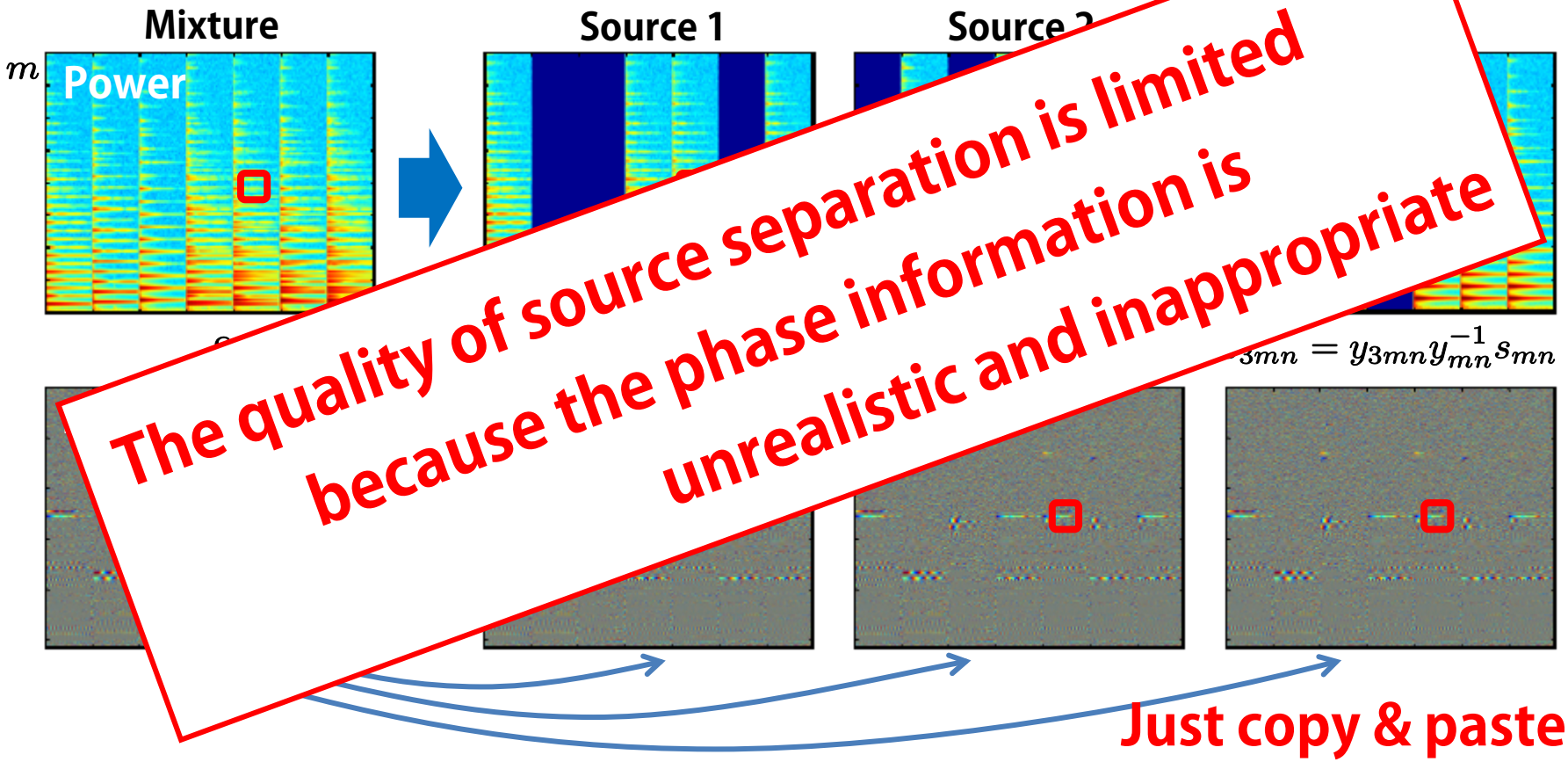
Source Separation based on NMF

- **Wiener filtering based on NMF results** $s_{kmn} = y_{kmn} y_{mn}^{-1} s_{mn}$
  - Element-wise decomposition of nonnegative power spectrogram
  - Ignore phases and treat frequency bins independently



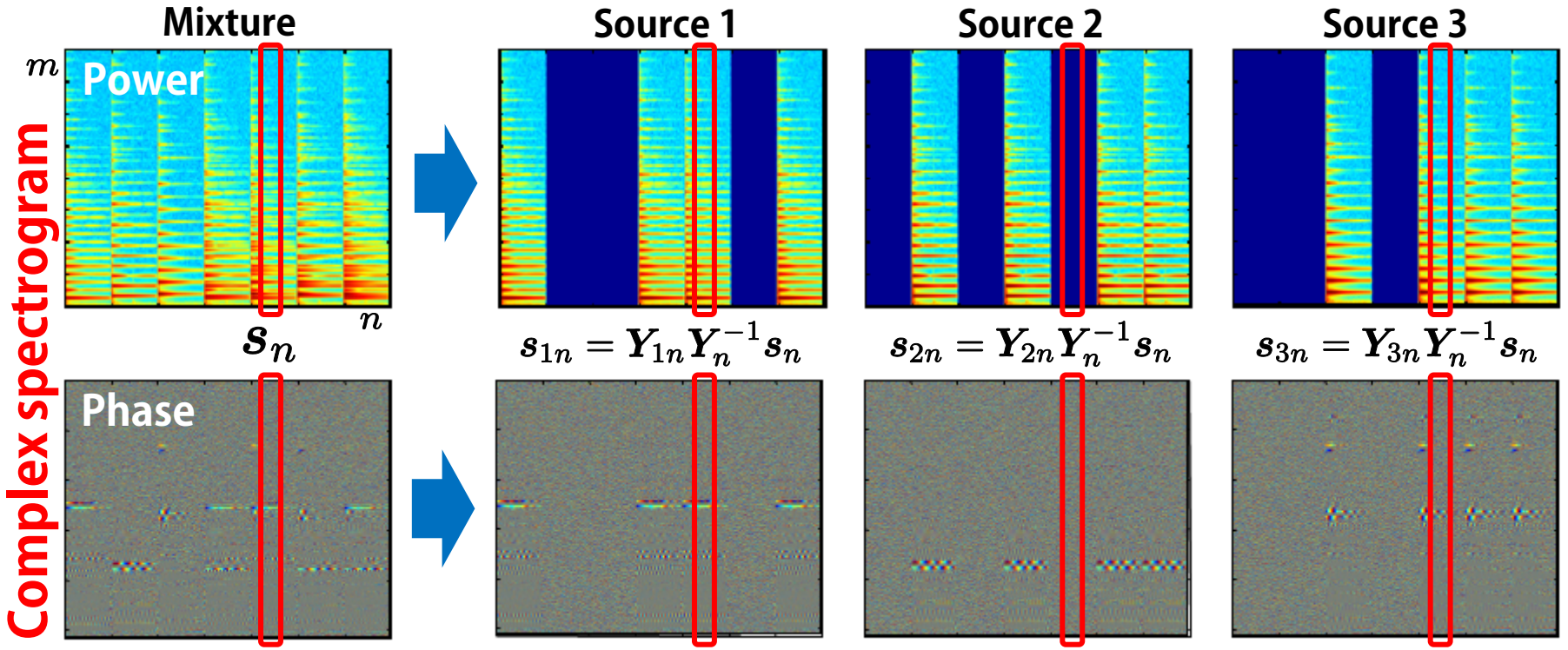
Source Separation based on NMF

- Wiener filtering based on NMF results $s_{kmn} = y_{kmn} y_{mn}^{-1} s_{mn}$
  - Element-wise decomposition of nonnegative power spectrogram
  - Ignore phases and treat frequency bins independently



Source Separation based on PSDTF

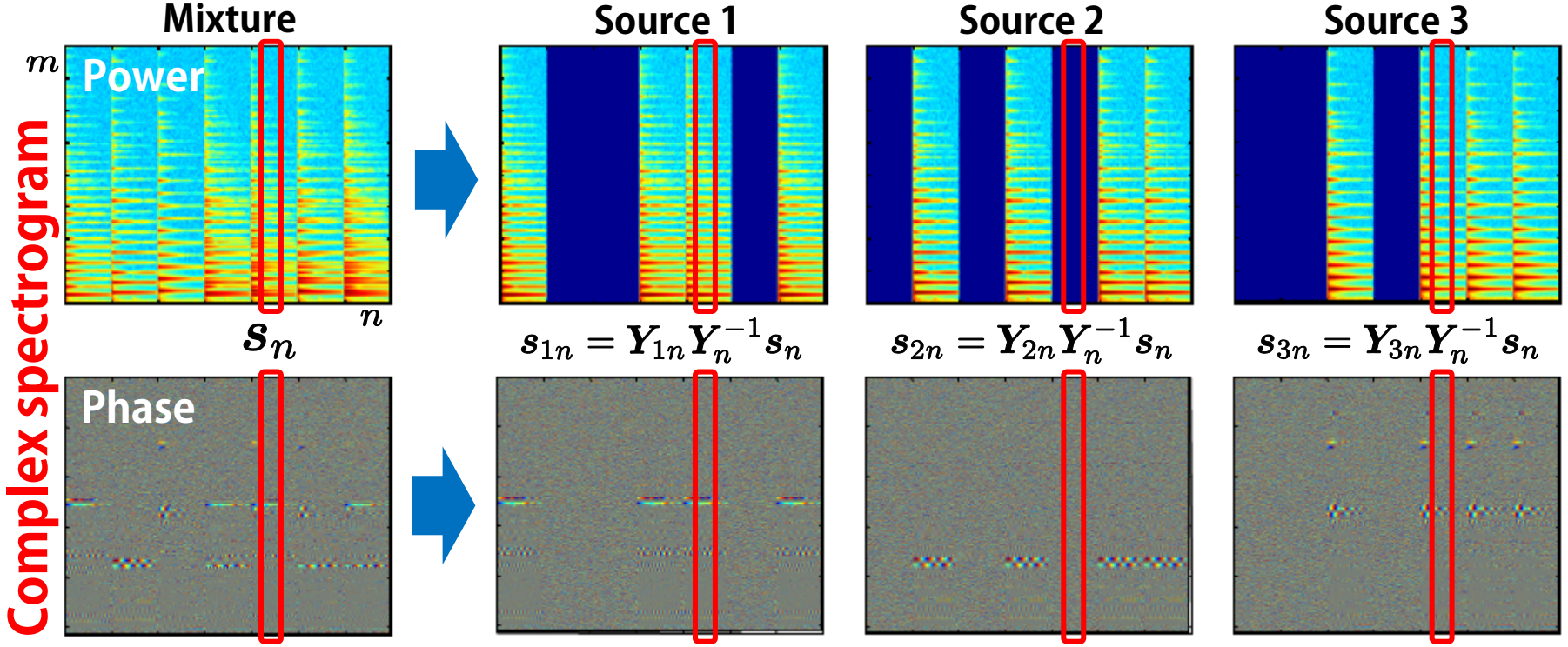
- Wiener filtering based on PSDTF results $s_{kn} = Y_{kn} Y_n^{-1} s_n$
  - Vector-wise decomposition of complex spectrogram



Source Separation based on PSDTF

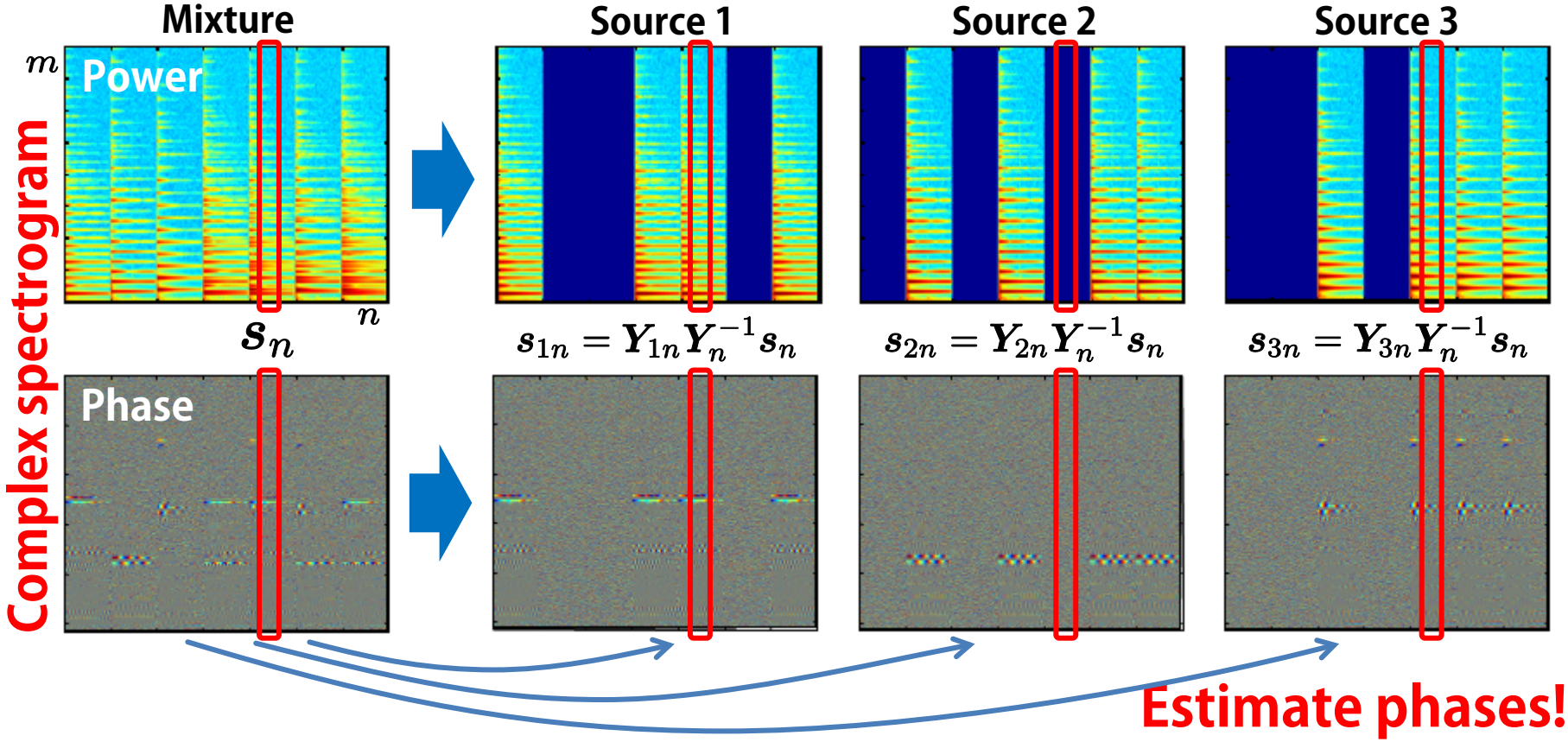
- Wiener filtering based on PSDTF results $\boldsymbol{s}_{kn} = \boldsymbol{Y}_{kn}\boldsymbol{Y}_{n}^{-1}\boldsymbol{s}_{n}$

The complex spectrogram of the observed mixture signal is decomposed in a vector-wise/frame-wise manner unlike NMF



Source Separation based on PSDTF

- Wiener filtering based on PSDTF results $\boldsymbol{s}_{kn} = \boldsymbol{Y}_{kn}\boldsymbol{Y}_{n}^{-1}\boldsymbol{s}_{n}$
  - Vector-wise decomposition of complex spectrogram
  - Consider phases and correlations between frequency bins

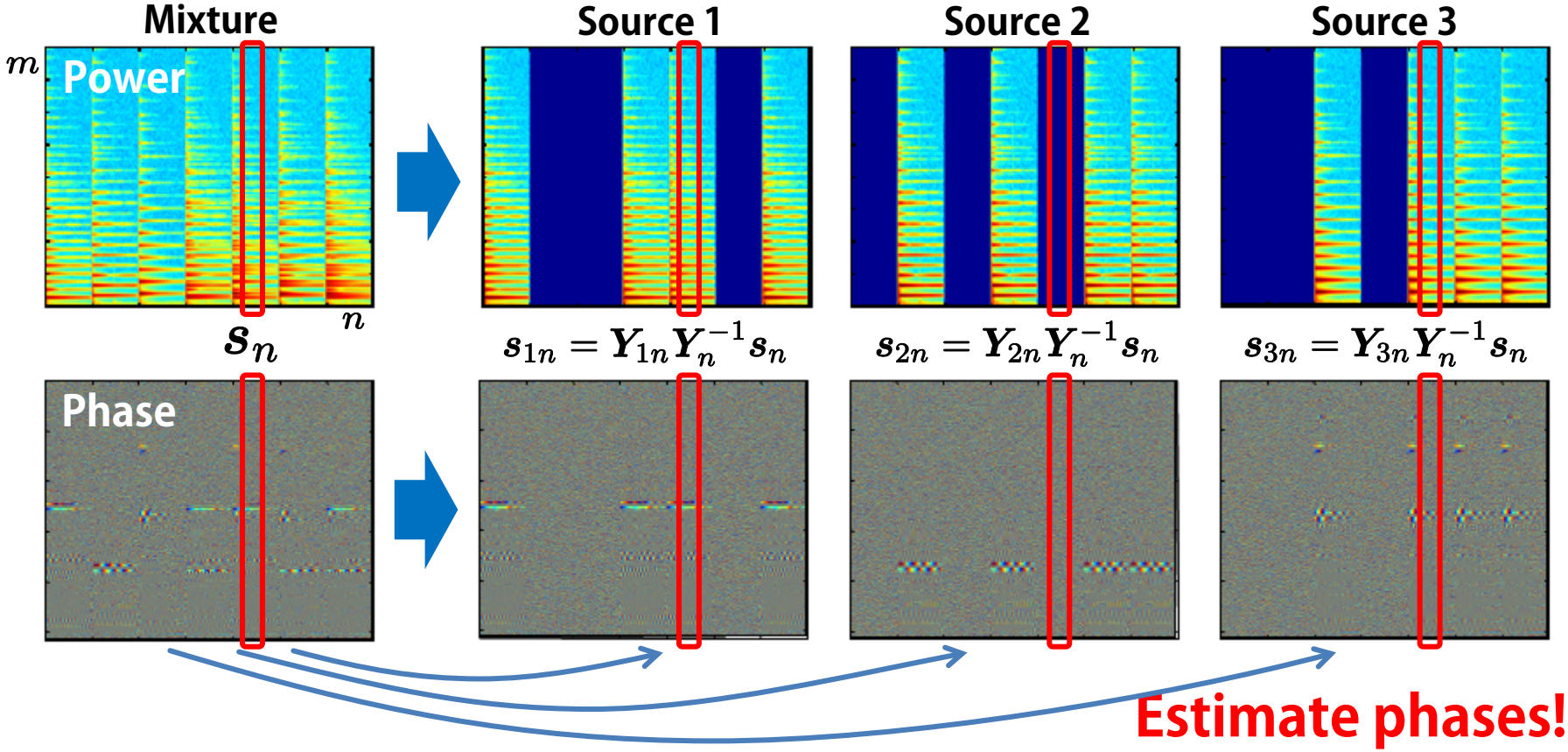


Source Separation based on PSDTF

- Wiener filtering based on PSDTF results $s_{kn} = Y_{kn} Y_n^{-1} s_n$

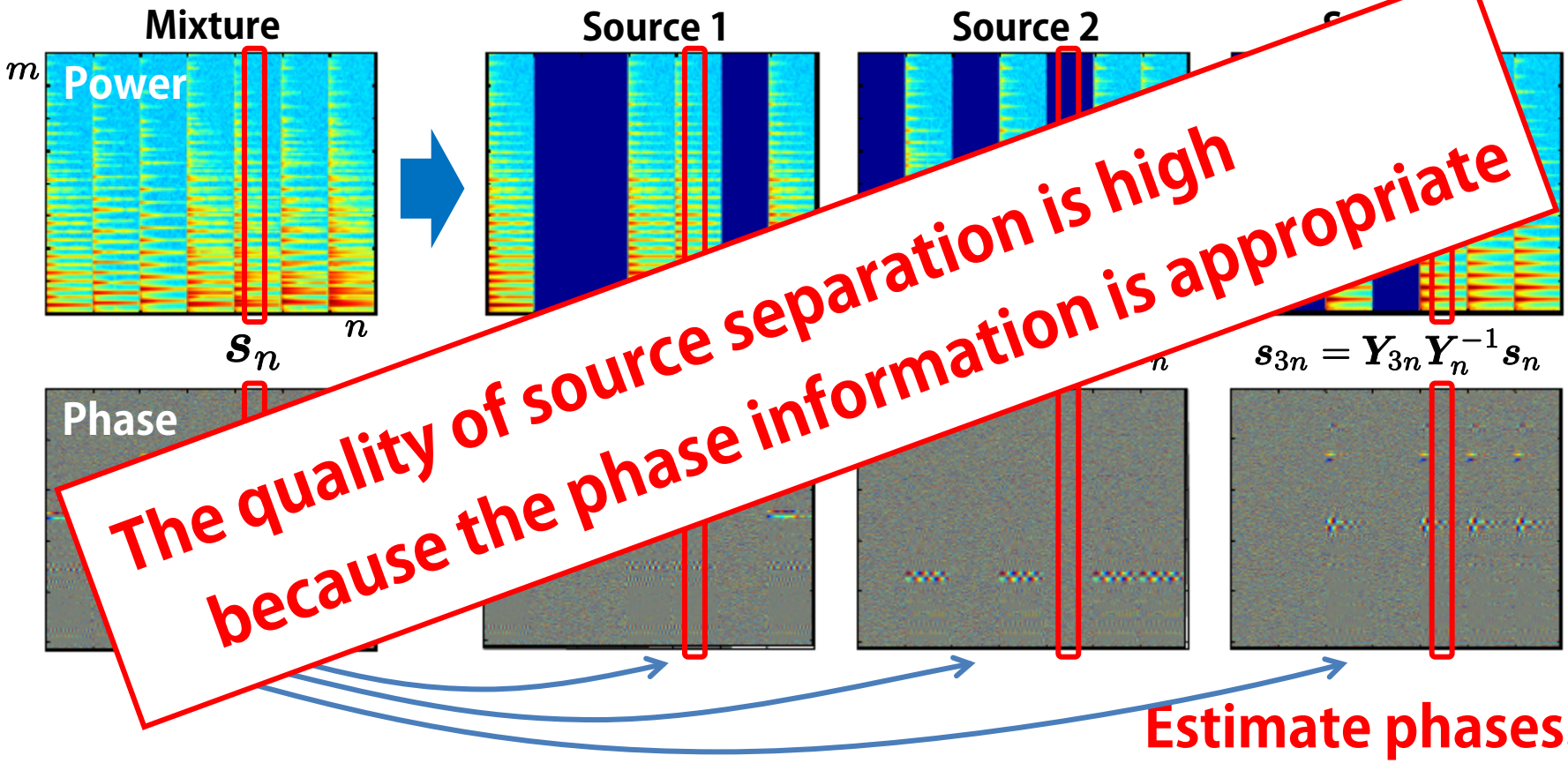
PSDTF considers the correlations between frequency bins!

Harmonic structure has strong correlations between harmonics



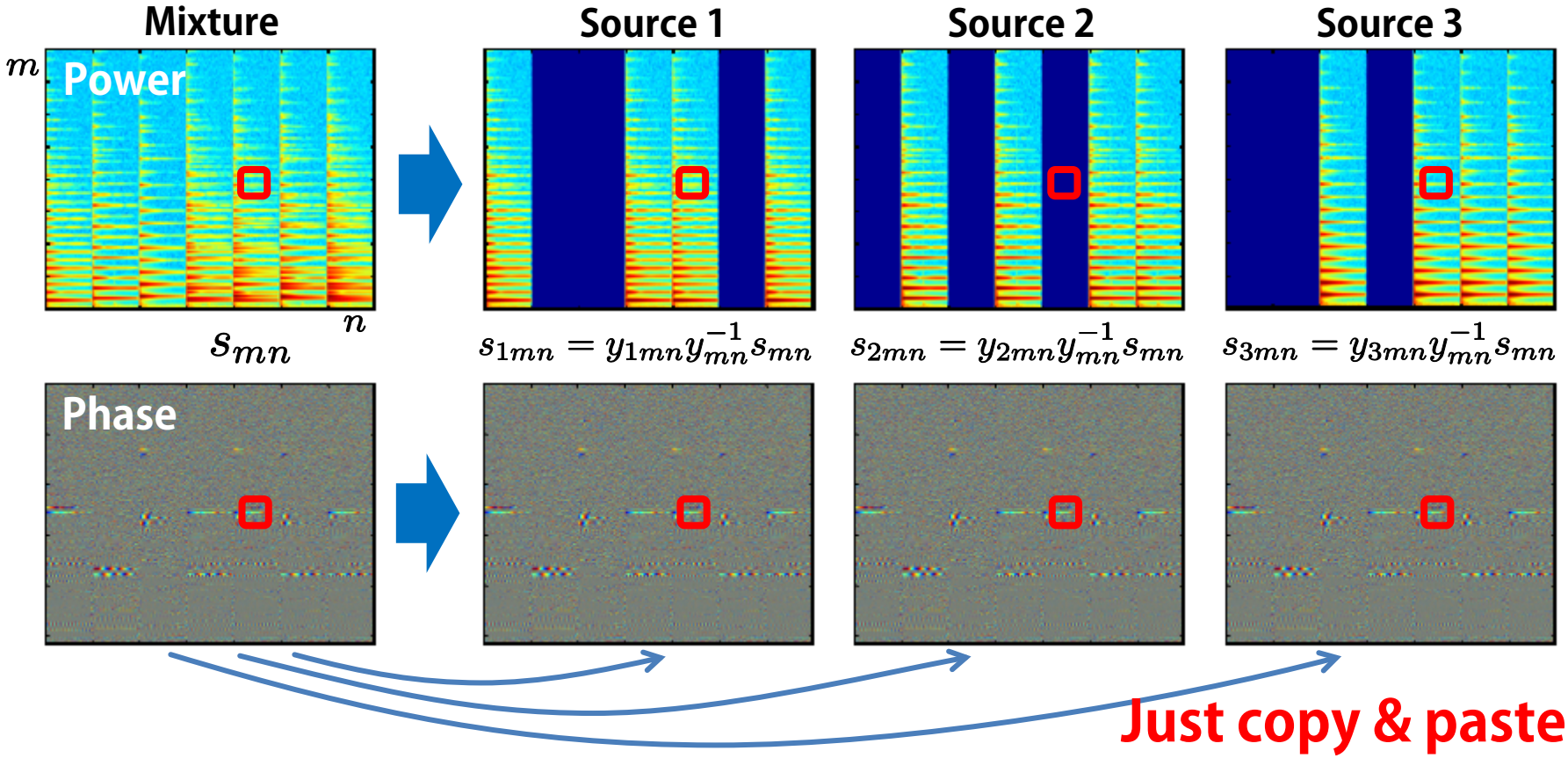
Source Separation based on PSDTF

- Wiener filtering based on PSDTF results $s_{kn} = Y_{kn} Y_n^{-1} s_n$
  - Vector-wise decomposition of complex spectrogram
  - Consider phases and correlations between frequency bins



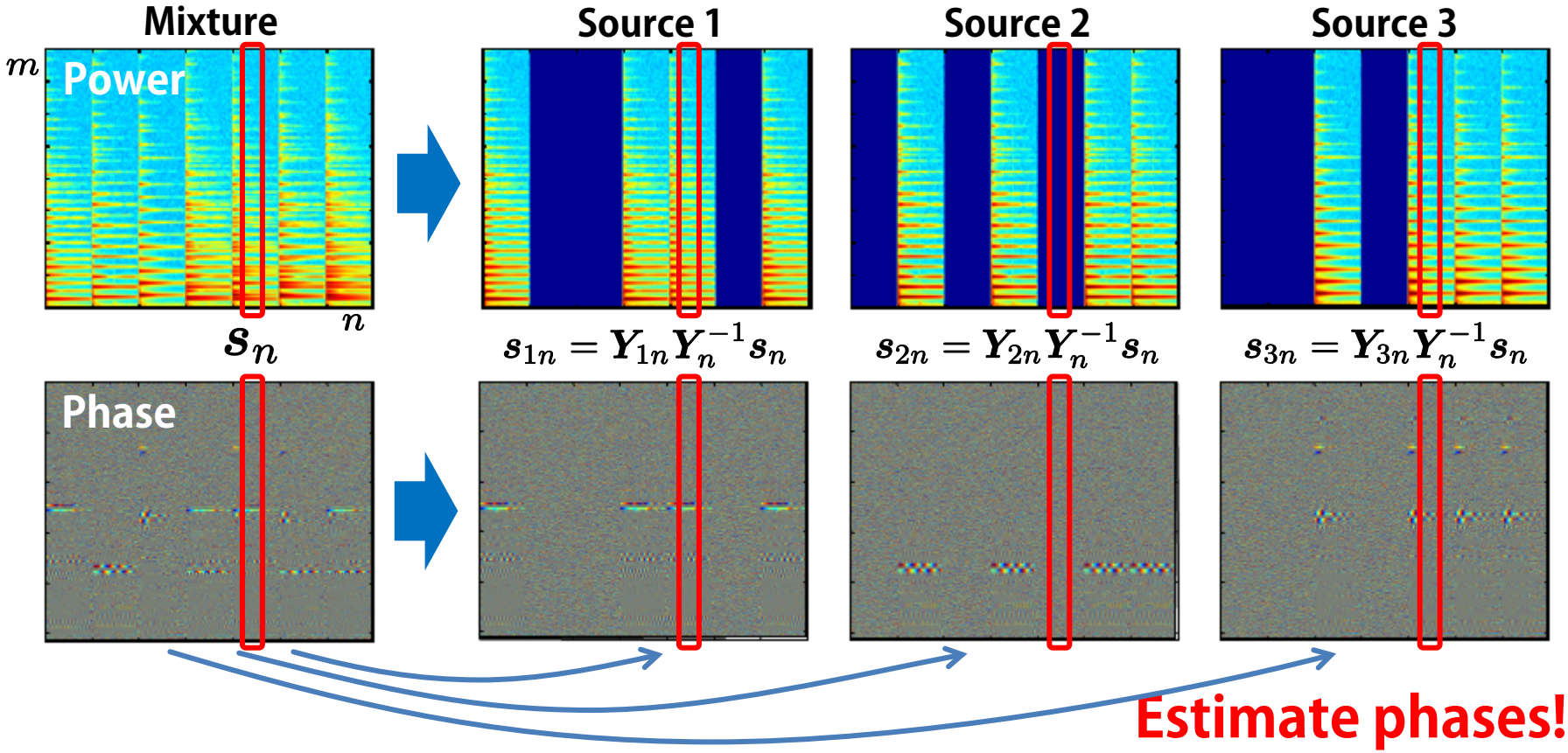
Source Separation based on NMF

- **Wiener filtering based on NMF results** $s_{kmn} = y_{kmn} y_{mn}^{-1} s_{mn}$
  - Element-wise decomposition of nonnegative power spectrogram
  - Ignore phases and treat frequency bins independently



Source Separation based on PSDTF

- Wiener filtering based on PSDTF results $s_{kn} = Y_{kn} Y_n^{-1} s_n$
  - Vector-wise decomposition of complex spectrogram
  - Consider phases and correlations between frequency bins



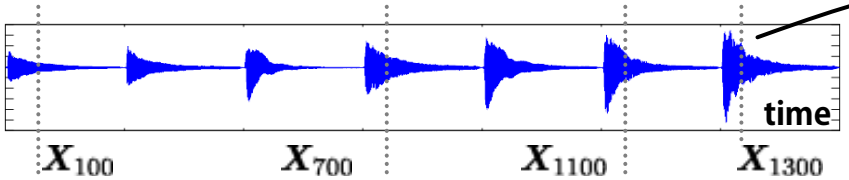
PSD Matrices

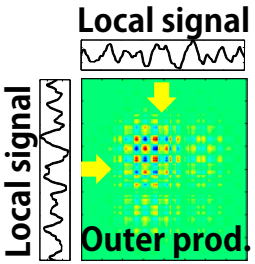
I intentionally skipped

a very important explanation

Positive SemiDefinite Tensor Factorization (PSDTF)

- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices

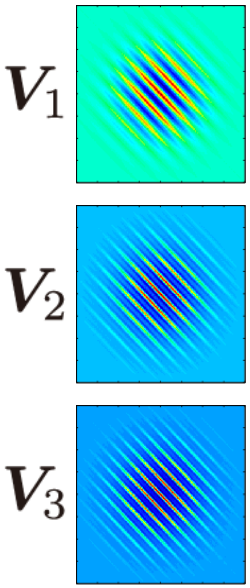




PSD matrices

PSD matrices

Bases



What are represented by PSD matrices?

Activations

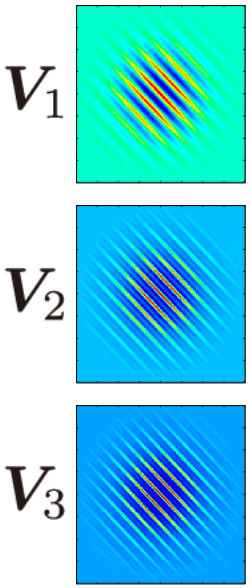
$k=1$

Gaussian Process (GP)

- A stochastic process over continuous time
  - GP can be interpreted as a probability distribution $p(f)$ over continuous functions $f : t \to x$

PSD matrices

Bases



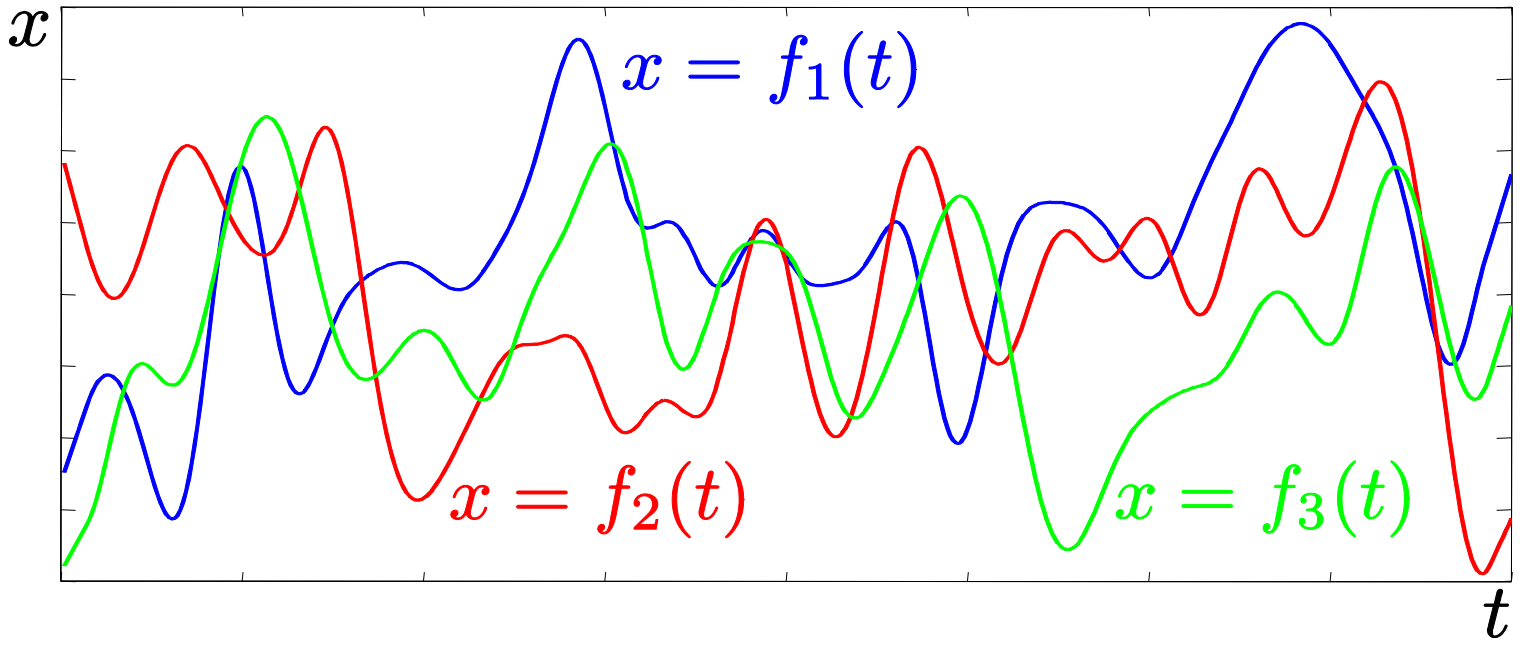
PSDTF matrix is
the kernel matrix (covariance matrix)
of Gaussian Process (GP)

$$\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$$

Kernel matrix
(PSD covariance matrix)

Gaussian Process (GP)

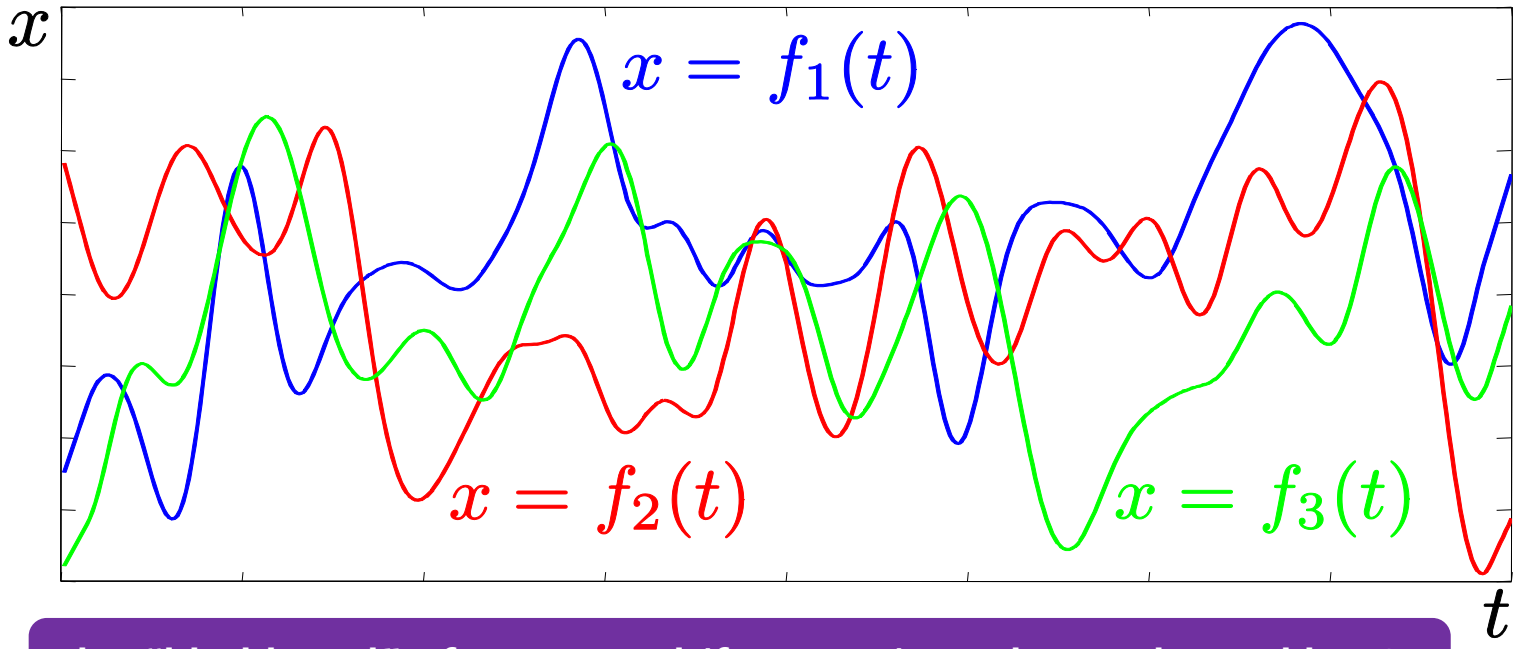
- A stochastic process over continuous time
  - GP can be interpreted as a probability distribution $p(f)$ over continuous functions $f : t \to x$



Three functions are stochastically sampled from the same GP

Gaussian Process (GP)

- A stochastic process over continuous time
  - GP can be interpreted as a probability distribution $p(f)$ over continuous functions $f: t \to x$

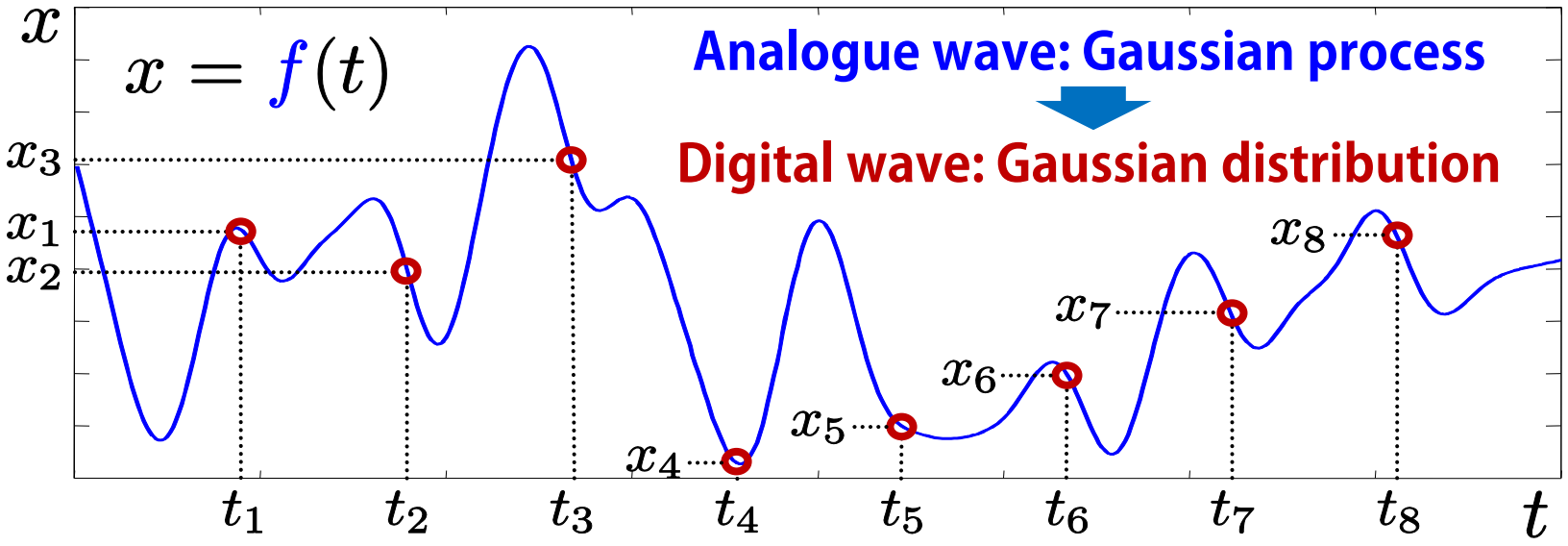


The "likelihood" of any signal (function) can be evaluated by GP

$p(f_1)$ $p(f_2)$ $p(f_3)$

How to Understand GP?

- **Gaussian process is characterized by its marginal distribution**
  - **A marginal distribution over any M points can be Gaussian**



$$\boldsymbol{t} = [t_1, t_2, \cdots, t_M]^T$$

f

$$\boldsymbol{x} = [x_1, x_2, \cdots, x_M]^T$$

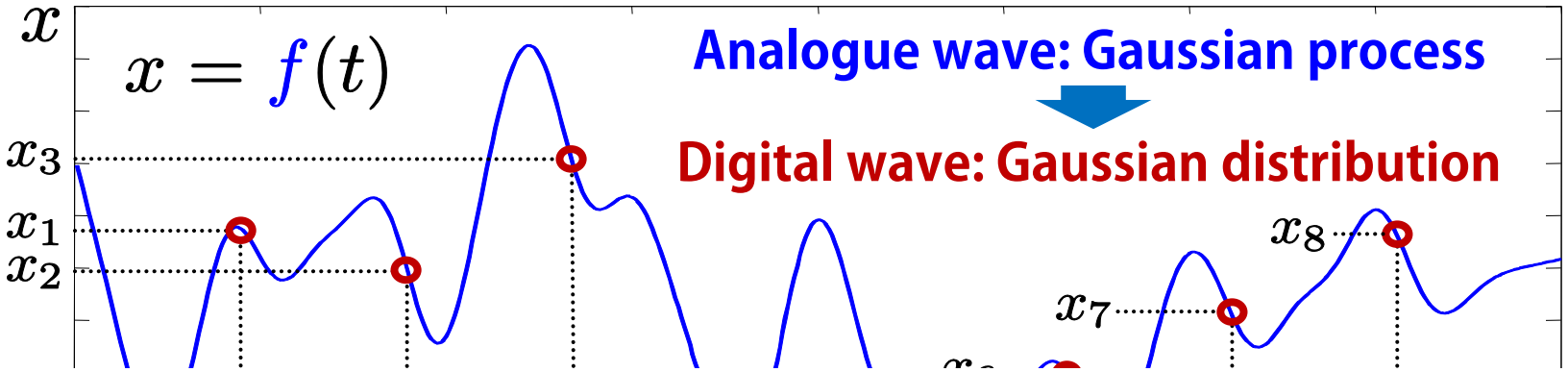
$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

Kernel matrix
(PSD covariance matrix)

The function dynamics are determined by the kernel matrix

How to Understand GP?

- Gaussian process is characterized by its marginal distribution
  - A marginal distribution over any M points can be Gaussian



This means that the uncertainty over $\boldsymbol{x}$ is modeled by a mean vector and a covariance matrix (kernel matrix)

$$\boldsymbol{t} = [t_1, t_2, \cdots, t_M]^T$$

$$f \rightarrow \boldsymbol{x} = [x_1, x_2, \cdots, x_M]^T$$

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$$

Kernel matrix (PSD covariance matrix)

The function dynamics are determined by the kernel matrix

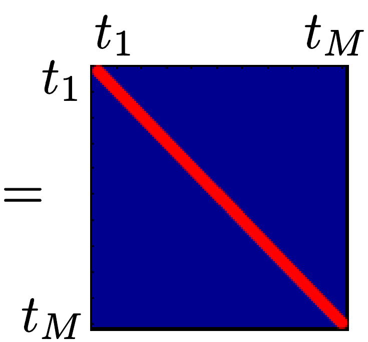
Example: Identity Kernel

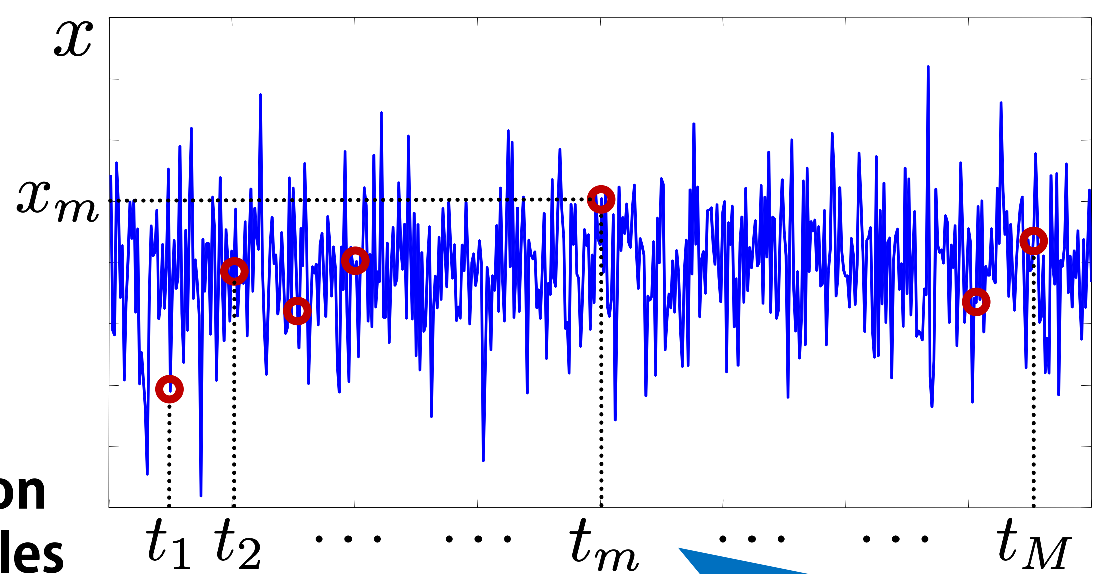
- **Stationary Gaussian white noise can be generated**
  - Marginal: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$

$$\boldsymbol{t} = [t_1, t_2, \cdots, t_M]^T$$

f

$$\boldsymbol{x} = [x_1, x_2, \cdots, x_M]^T$$

$\boldsymbol{K} =$ 



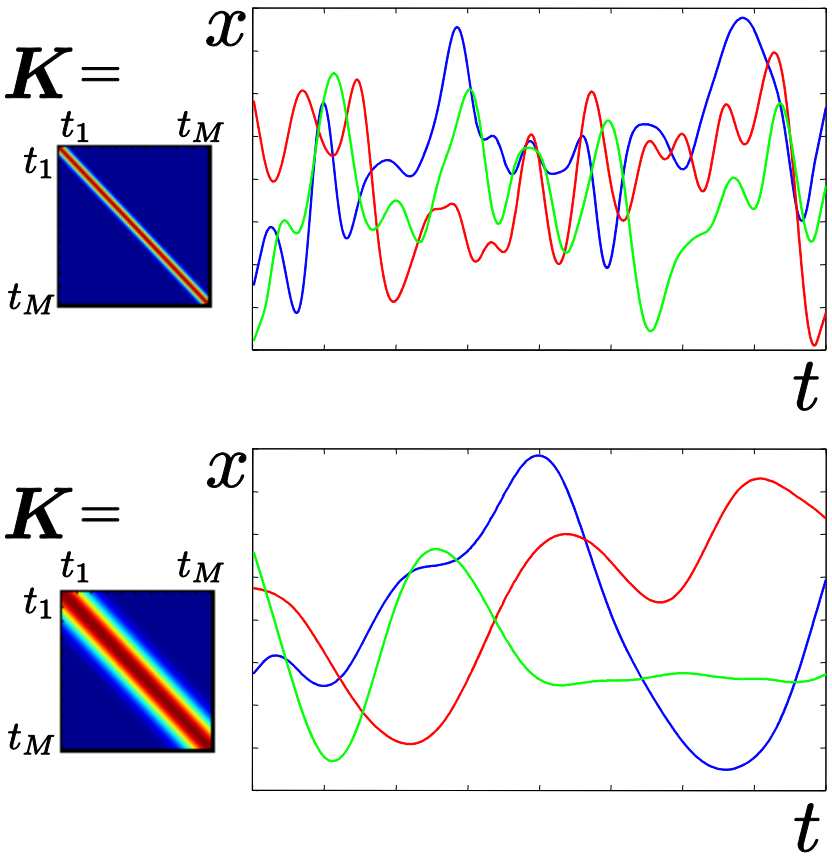
We consider equally-spanned sampling points

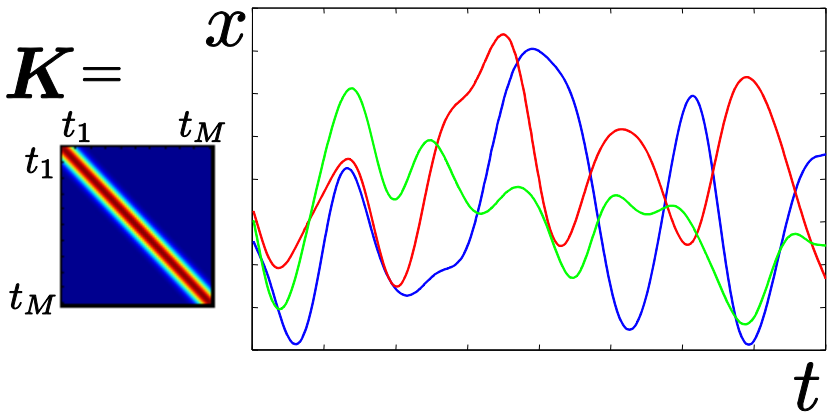
In this kernel, there is no correlation between samples
→ Its GP can generate white noise

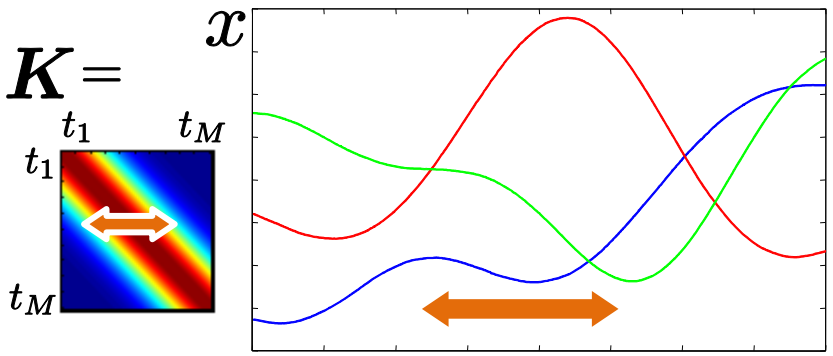
Example: Squared Exponential Kernel (RBF)

- **Smooth functions can be generated**
  - Marginal: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$

$$\boldsymbol{t} = [t_1, t_2, \cdots, t_M]^T$$
$$\boldsymbol{x} = [x_1, x_2, \cdots, x_M]^T = f(\boldsymbol{t})$$





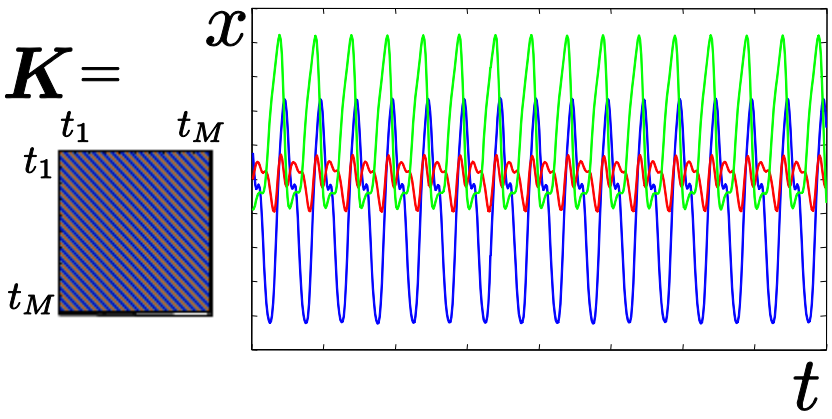


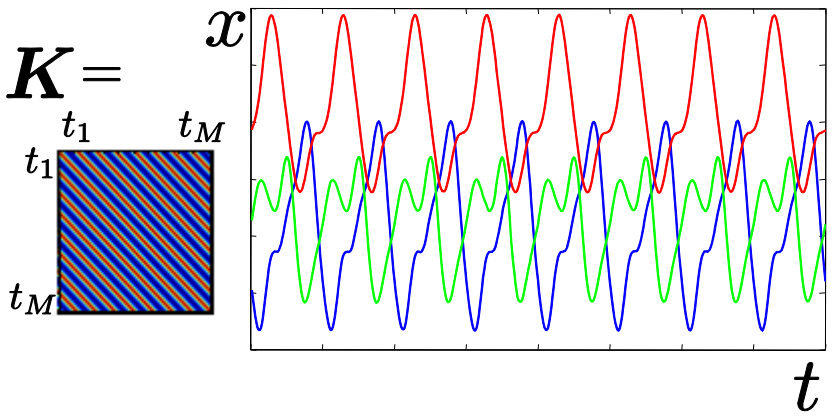
Strong correlation in the vicinity
→ Slowly changing function

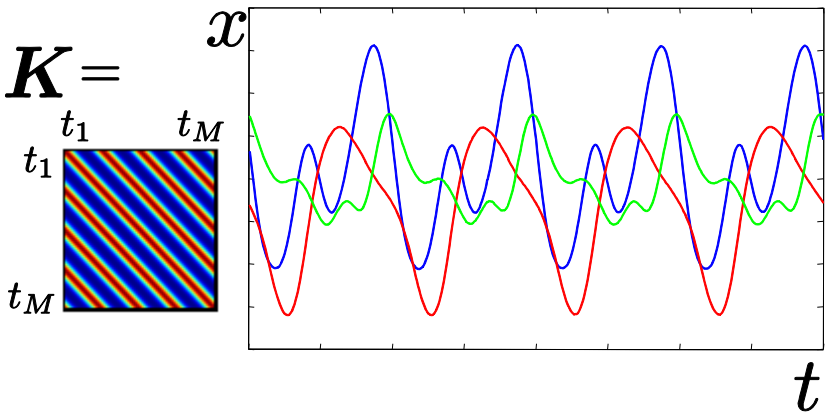
Example: Periodic Kernel (Stripe Pattern)

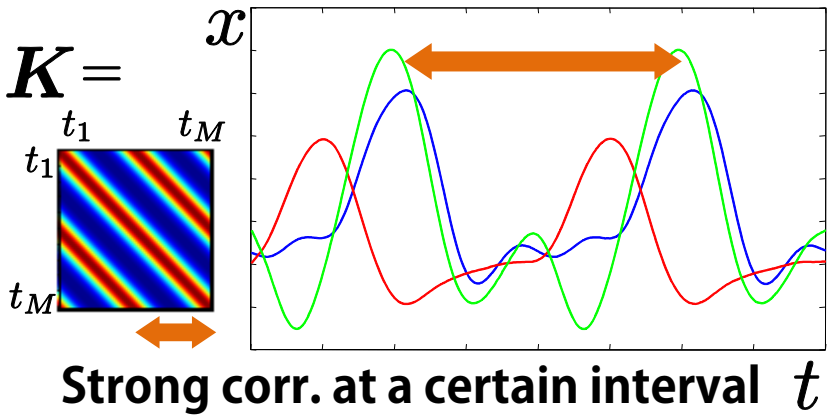
- **Periodic functions can be generated**
  - **Marginal:** $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$

$$\boldsymbol{t} = [t_1, t_2, \cdots, t_M]^T$$
$$\boldsymbol{x} = [x_1, x_2, \cdots, x_M]^T = f(\boldsymbol{t})$$





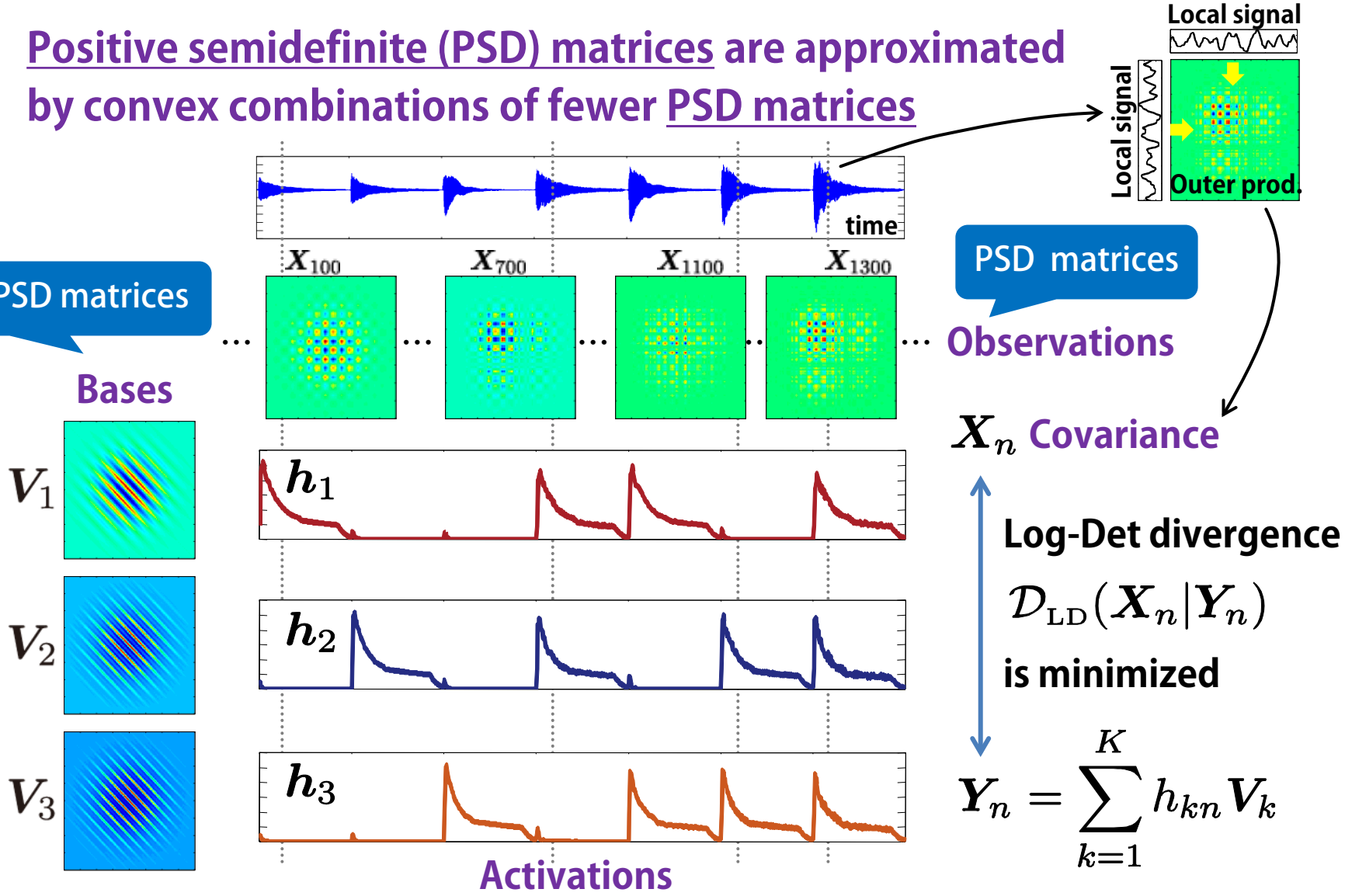




Strong corr. at a certain interval
→ Periodic function

Positive SemiDefinite Tensor Factorization (PSDTF)

- Positive semidefinite (PSD) matrices are approximated by convex combinations of fewer PSD matrices



NMF vs. PSDTF

- **Nonnegative Matrix Factorization (NMF)**
  - **Nonnegative vectors → sums of nonnegative vectors**
  - **Bregman divergence**

$$\boldsymbol{x}_n \approx \boldsymbol{y}_n = \sum_{k=1}^{K} h_{kn} \boldsymbol{w}_k$$

    - **Itakura-Saito (IS) divergence**

$$\mathcal{D}_{\mathrm{IS}}(\boldsymbol{x}_n | \boldsymbol{y}_n) = \sum\nolimits_m \left(-\log x_{mn} y_{mn}^{-1} + x_{mn} y_{mn}^{-1} - 1 \right)$$

    - **Kullback-Leibler (KL) divergence**

$$\mathcal{D}_{\mathrm{KL}}(\boldsymbol{x}_n | \boldsymbol{y}_n) = \sum\nolimits_m \left(x_{mn} \log x_{mn} y_{mn}^{-1} - x_{mn} + y_{mn} \right)$$

- **Positive Semidefinite Tensor Factorization (PSDTF)**
  - **PSD matrices → sums of PSD matrices**
  - **Bregman "matrix" divergence**

$$\boldsymbol{X}_n \approx \boldsymbol{Y}_n = \sum_{k=1}^{K} h_{kn} \boldsymbol{V}_k$$

    - **Log-Determinant (LD) divergence**

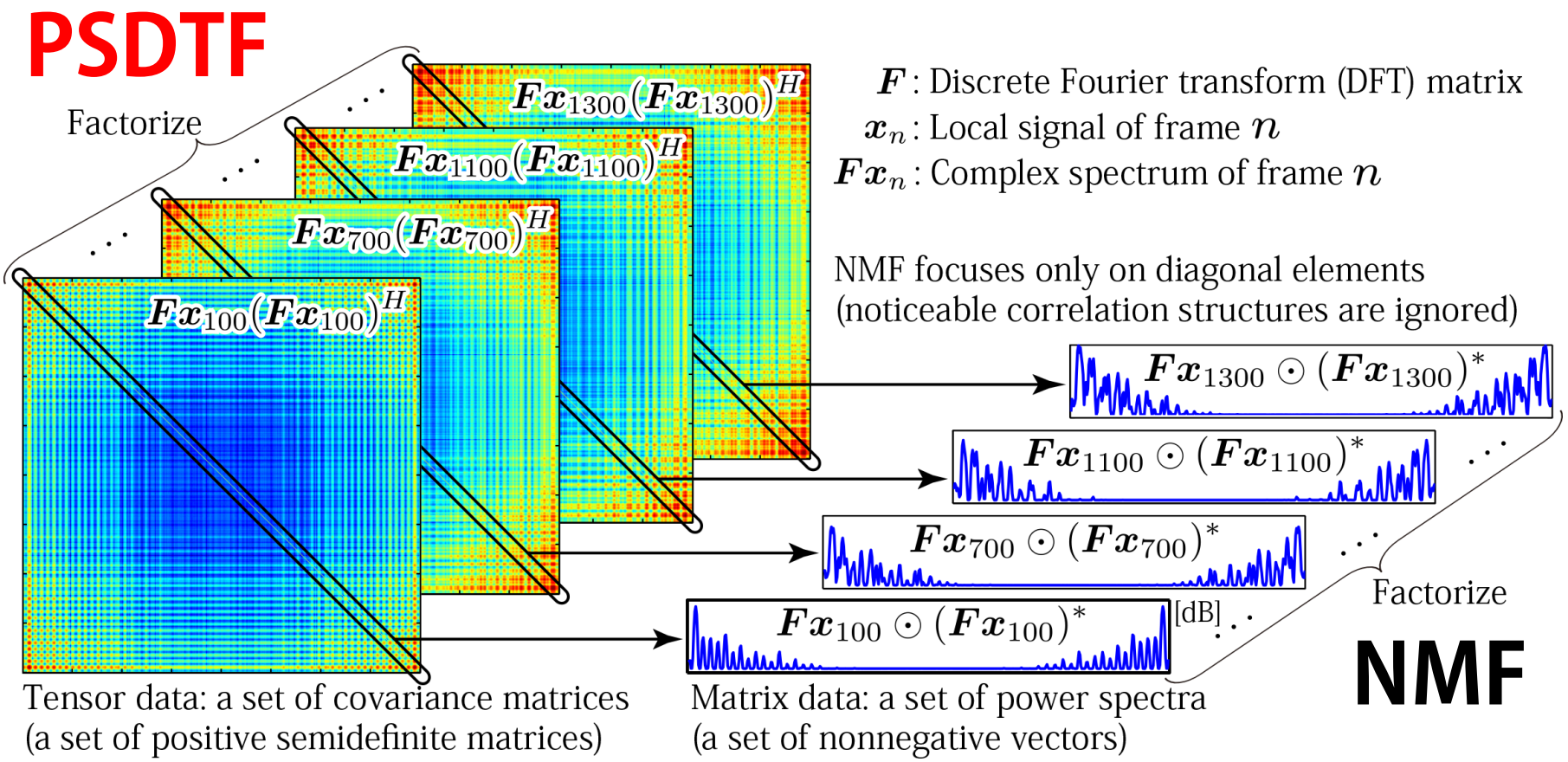
$$\mathcal{D}_{\mathrm{LD}}(\boldsymbol{X}_n | \boldsymbol{Y}_n) = -\log \left| \boldsymbol{X}_n \boldsymbol{Y}_n^{-1} \right| + \mathrm{tr} \left(\boldsymbol{X}_n \boldsymbol{Y}_n^{-1} \right) - M$$

    - **von Neumann (vN) divergence**

$$\mathcal{D}_{\mathrm{vN}}(\boldsymbol{X}_n | \boldsymbol{Y}_n) = \mathrm{tr} \left(\boldsymbol{X}_n \log \boldsymbol{X}_n - \boldsymbol{X}_n \log \boldsymbol{Y}_n - \boldsymbol{X}_n + \boldsymbol{Y}_n \right)$$

Frequency-domain PSDTF

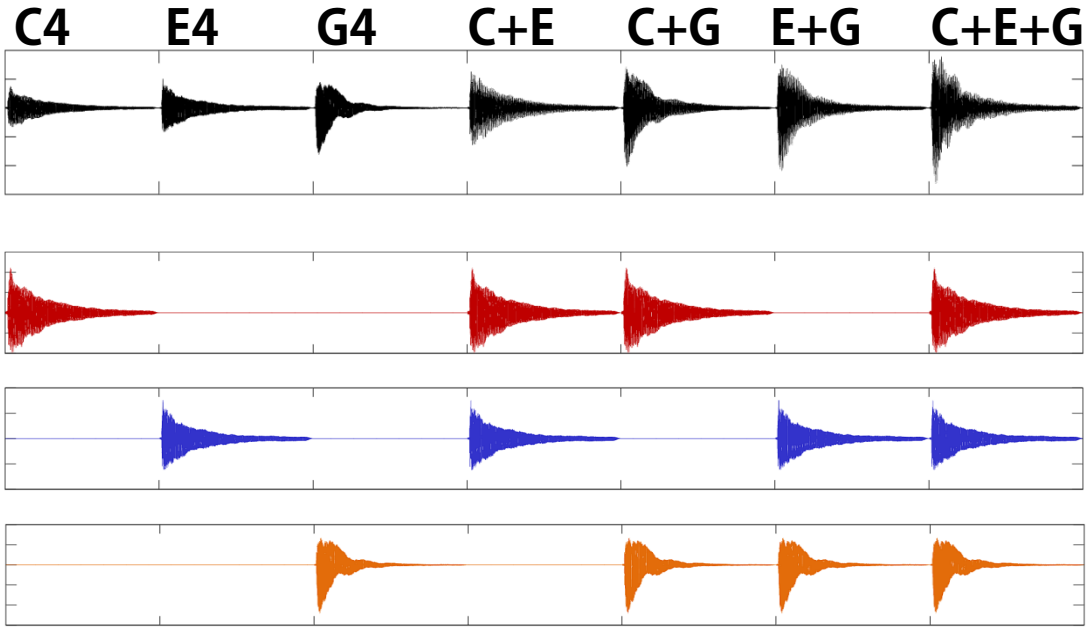
- **PSDTF is a natural extension of NMF**



Comparative Evaluation

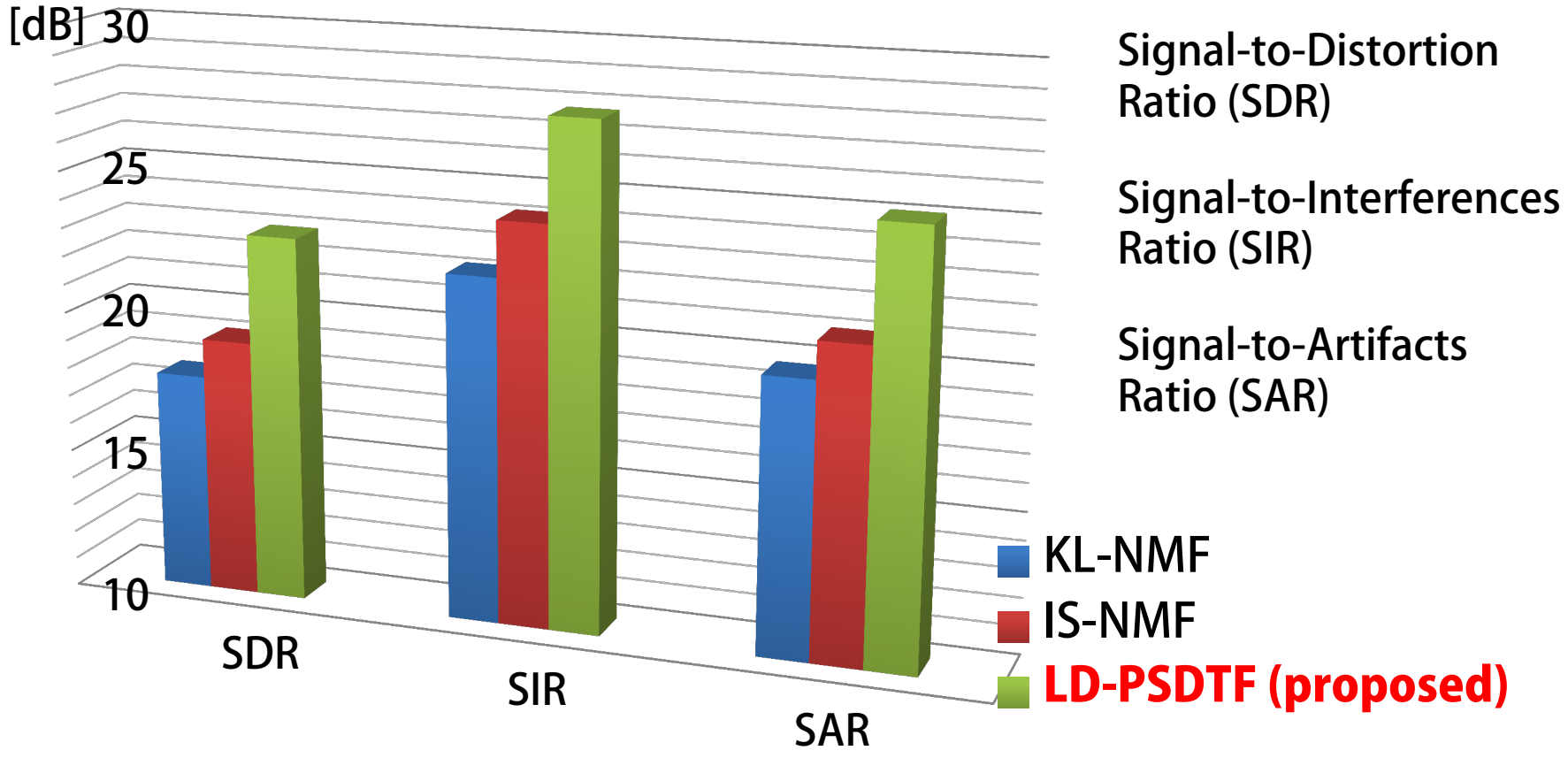
- Source separation performance on synthetic sounds
  - 3 bases (K=3)
  - 3 instruments (from RWC Music Database)
    - Piano / Electric guitar (decaying) / Clarinet (sustaining)
  - BSS Eval Toolbox [Vincent2006] was used

$$x = x_1 + x_2 + x_3$$



Experimental Results

- LD-PSDTF outperformed KL-NMF and IS-NMF



Conclusion and Future Work

- **Positive Semidefinite Tensor Factorization (PSDTF)**
  - A natural extension of Nonnegative Matrix Factorization (NMF)
  - Nonparametric Bayesian extension [Yoshii *et al.* ICML 2013]
    - Automatically optimize the number of bases

If you do not care about
computational time,
you could forget NMF and
use our method PSDTF
for audio source separation

We provide MATLAB source codes! *(2-Clause BSD License)*

http://staff.aist.go.jp/k.yoshii/psdtf/
or the last tweet of @MasatakaGoto

Beyond NMF:
Time-Domain Audio Source Separation without Phase Reconstruction

PSDTF

Kazuyoshi Yoshii (AIST, Japan)
Ryota Tomioka (Univ. of Tokyo , Japan)
Daichi Mochihashi (ISM , Japan)
Masataka Goto (AIST , Japan)

2013/11/06 ISMIR 2013

We Cannot Ignore Computational Time

- **Current issue: Big computational cost!**
  - **Comparison of computational cost**
    - **#samples: M=512 (frame size: 32ms)**
    - **#frames: N=1400 (duration: 14s)**
    - **#bases: K=3 (Mixtures of 3 sounds)**
    - **MATLAB / Intel Xeon 3.4GHz**

    - **LD-PSDTF: 600s/iteration** $O(M^3NK)$
      - **16.7 hours (60000s) / 100iterations**

    - **IS-NMF: 0.1s/iteration** $O(MNK)$
      - **10s / 100iterations**