Selected Topics in Deep Learning for Audio, Speech, and Music Processing

Nonnegative Autoencoders with Applications to Music Audio Decomposing

Meinard Müller, Yigitcan Özer

International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

31.05.2021

Thanks

- Tim Zunner (Master Thesis 2021)
- Edgar Suárez Guarnizo (Master Thesis 2020)
- Christian Dittmar (PhD 2018, Fraunhofer IIS)
- Michael Krause (PhD student)
- Yigitcan Özer (PhD student)

Literature

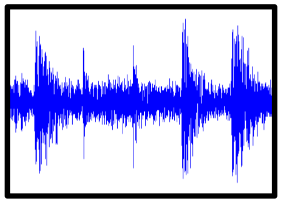
- Daniel Lee and Sebastian Seung: **Algorithms for Non-Negative Matrix Factorization.** Proc. NIPS, 2000.
- Sebastian Ewert and Meinard Müller: **Using Score-Informed Constraints for NMF-Based Source Separation.** Proc. ICASSP, 2012.
- Paris Smaragdis and Shrikant Venkataramani: **A Neural Network Alternative to Non-Negative Audio Models.** Proc. ICASSP, 2017.
- Sebastian Ewert and Mark B. Sandler: **Structured Dropout for Weak Label and Multi-Instance Learning and Its Application to Score-Informed Source Separation.** Proc. ICASSP, 2017.
- Tim Zunner: **Neural Networks with Nonnegativity Constraints for Decomposing Music Recordings.** Master Thesis, FAU, 2021.
- Edgar Andrés Suárez Guarnizo: **DNN-Based Matrix Factorization with Applications to Drum Sound Decomposition.** Master Thesis, FAU, 2020.

Score-Informed Source Separation

Exploit musical score to support decomposition process

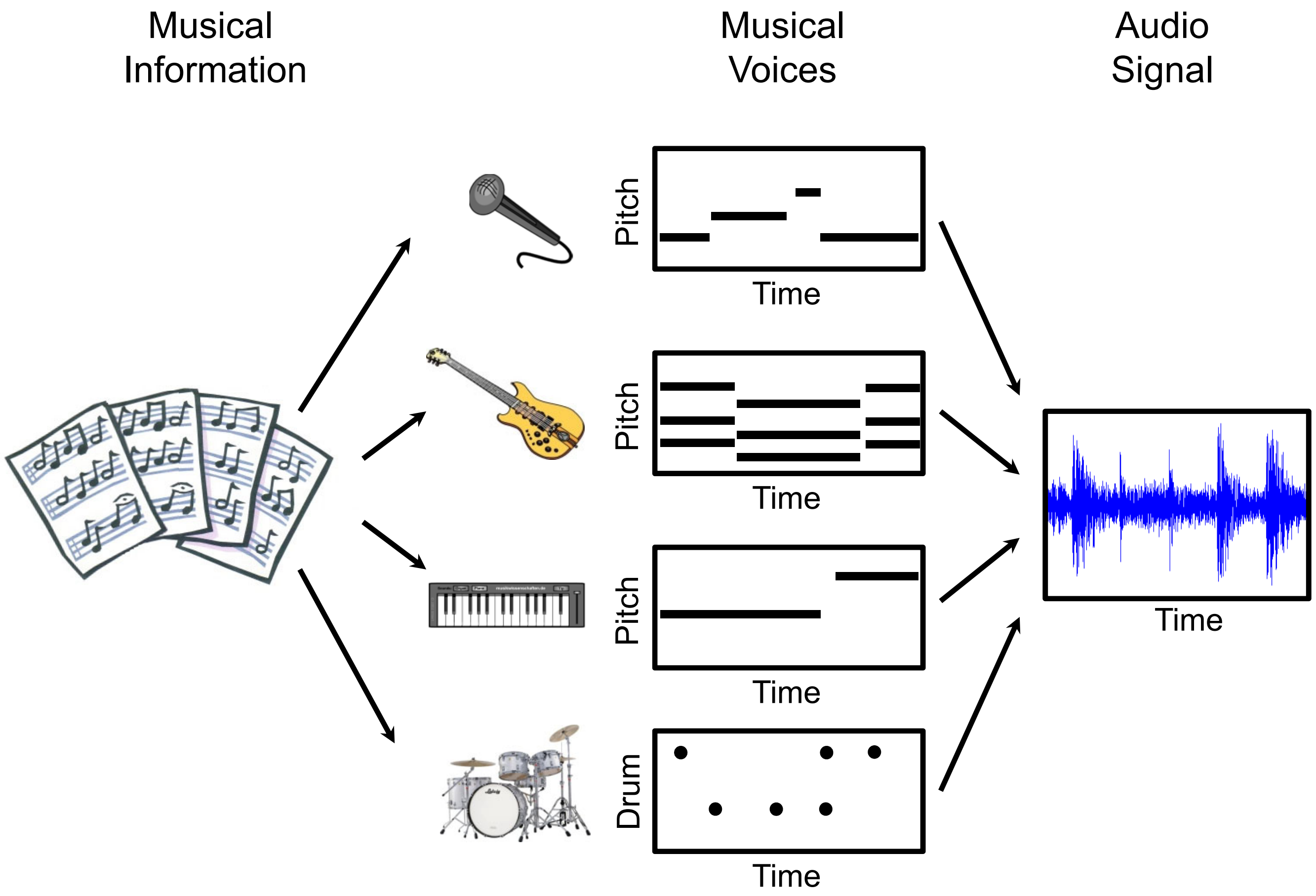
Musical
Information

Audio
Signal



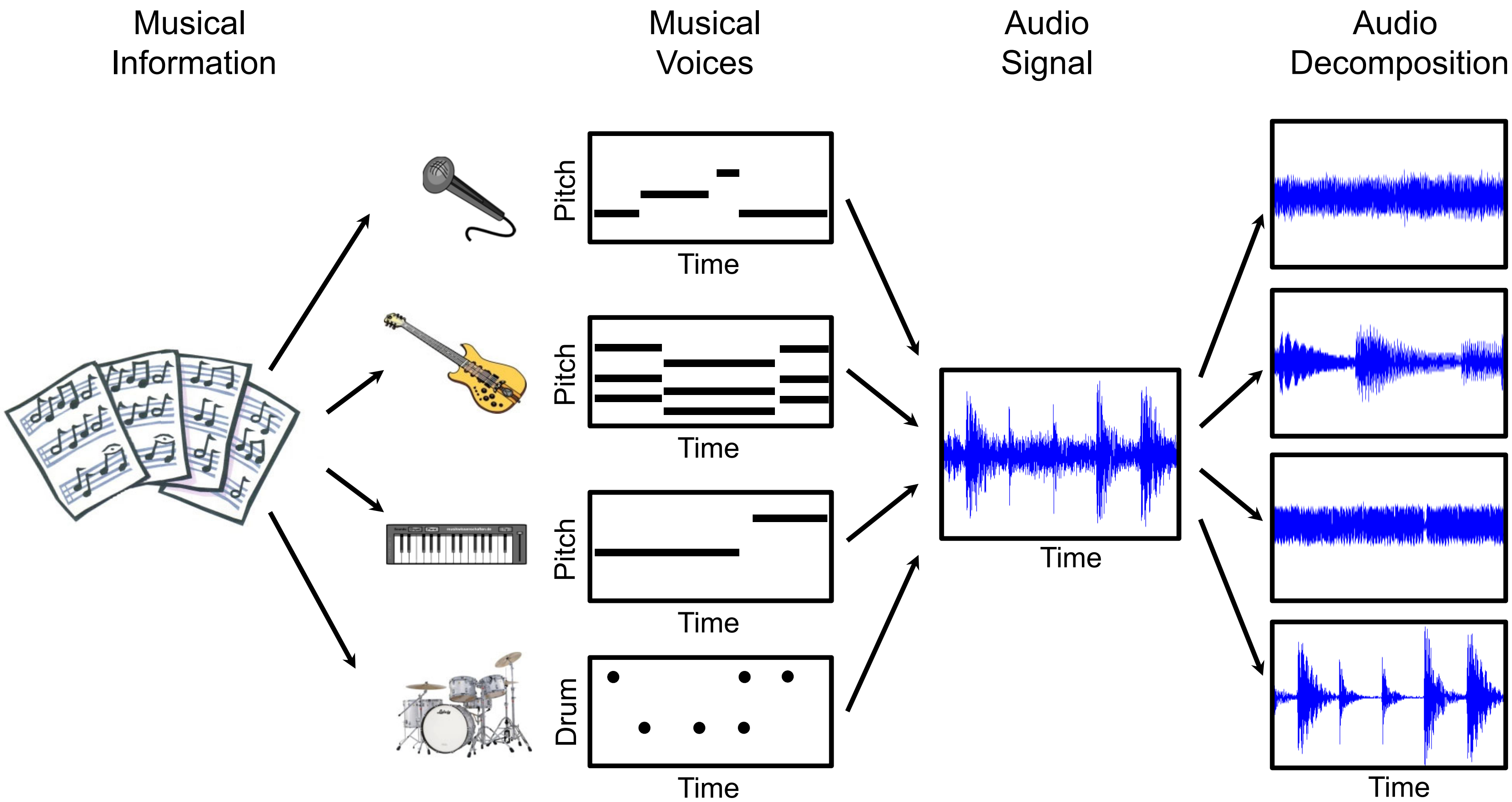
Score-Informed Source Separation

Exploit musical score to support decomposition process



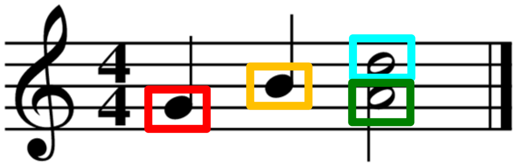
Score-Informed Source Separation

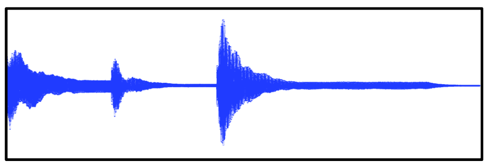
Exploit musical score to support decomposition process



Score-Informed Audio Decomposition

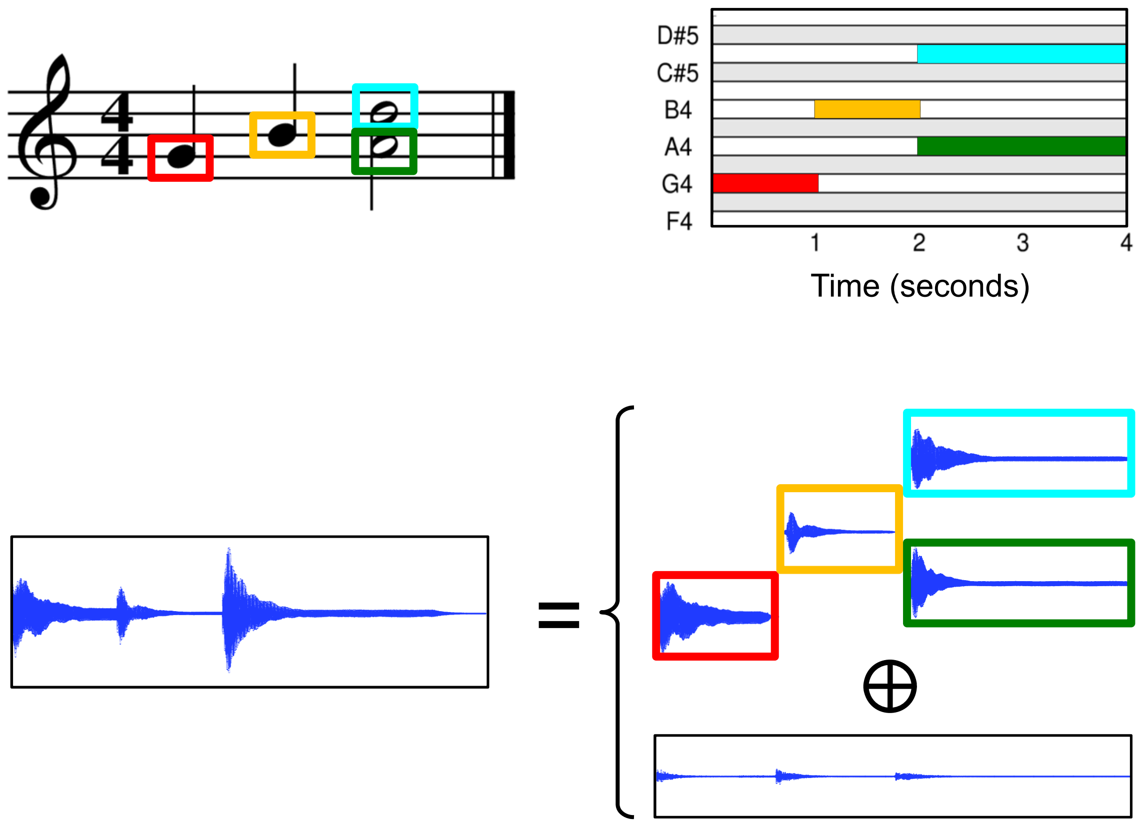
Notewise decomposition





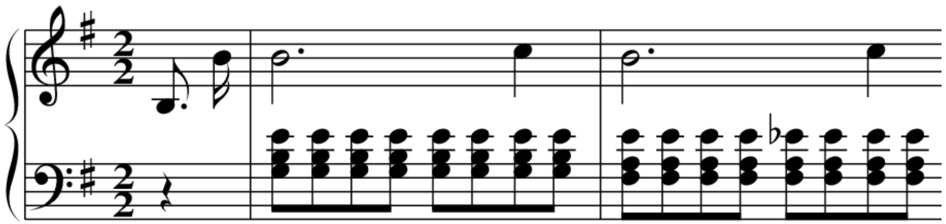
Score-Informed Audio Decomposition

Notewise decomposition

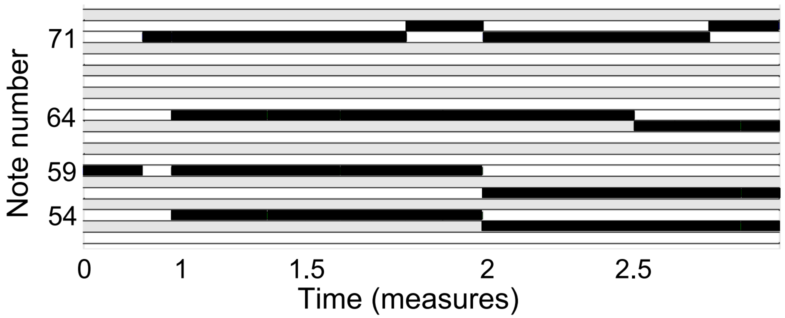


Score-Informed Audio Decomposition

Sheet music

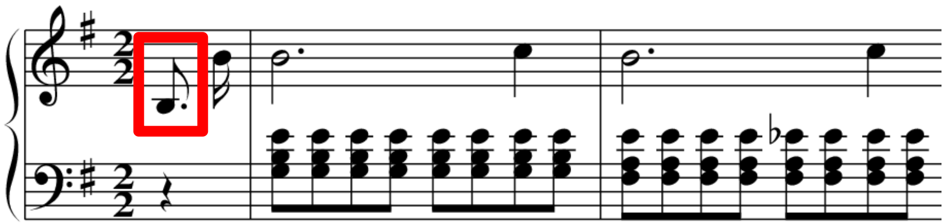


Piano roll



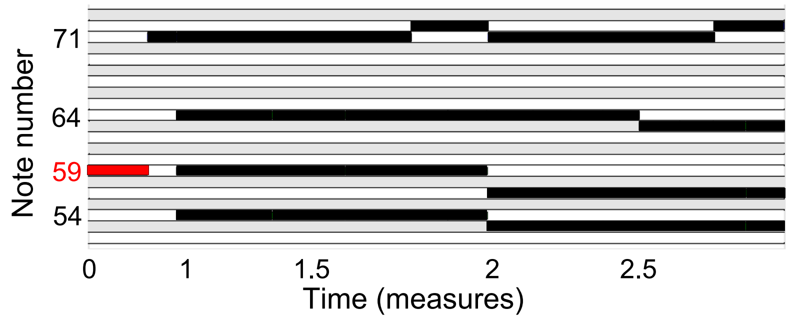
Score-Informed Audio Decomposition

Sheet music



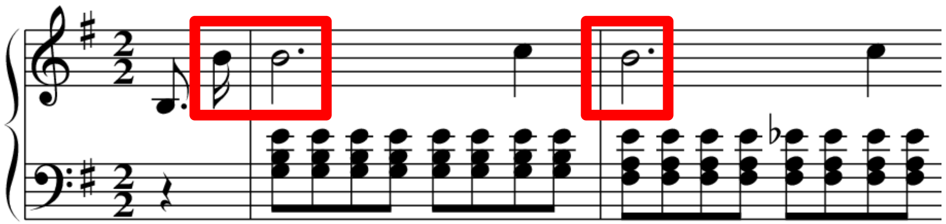
p = 59

Piano roll



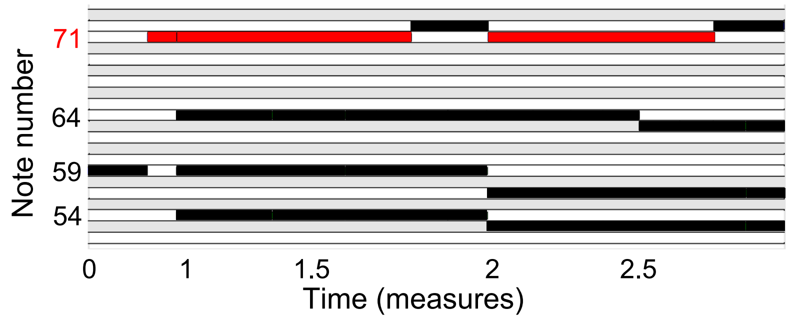
Score-Informed Audio Decomposition

Sheet music



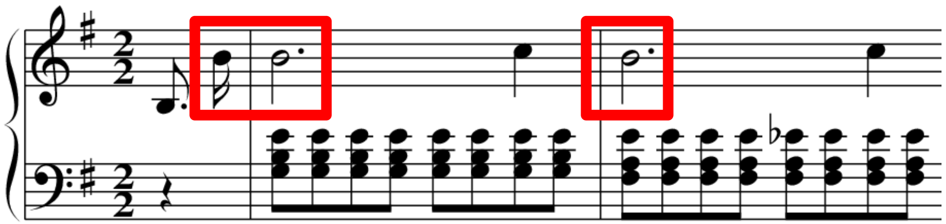
p = 71

Piano roll



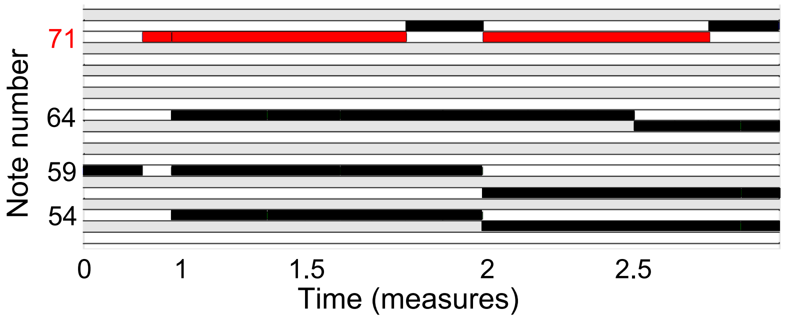
Score-Informed Audio Decomposition

Sheet music

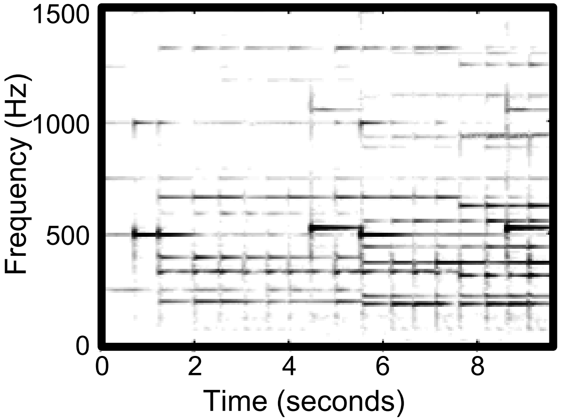


p = 71

Piano roll

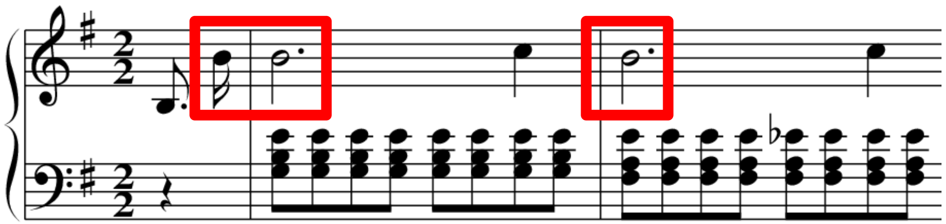


Spectrogram

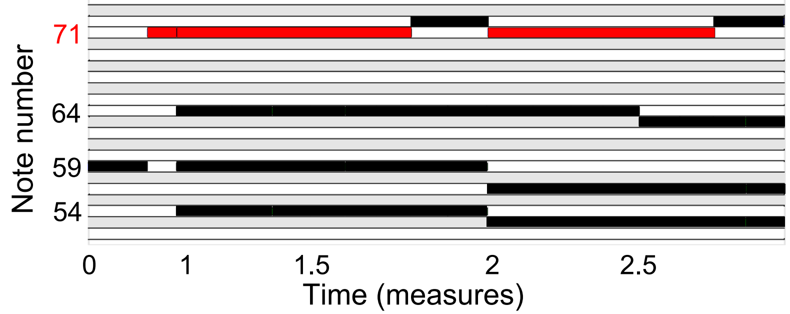


Score-Informed Audio Decomposition

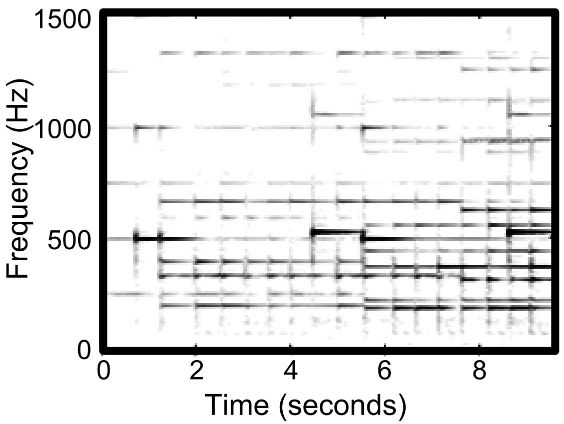
Sheet music



p = 71 Piano roll

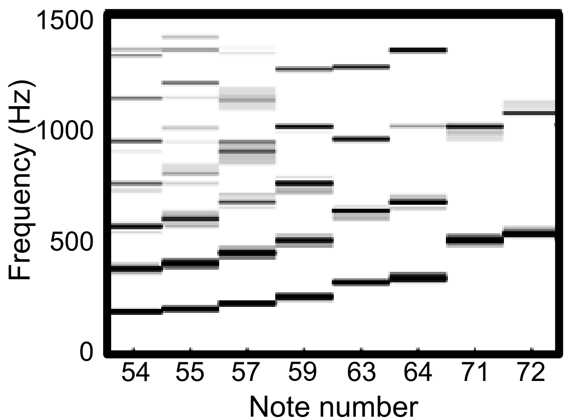


Spectrogram



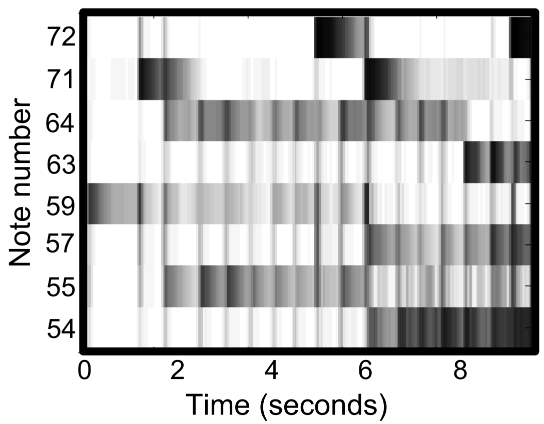
≈

Spectral patterns

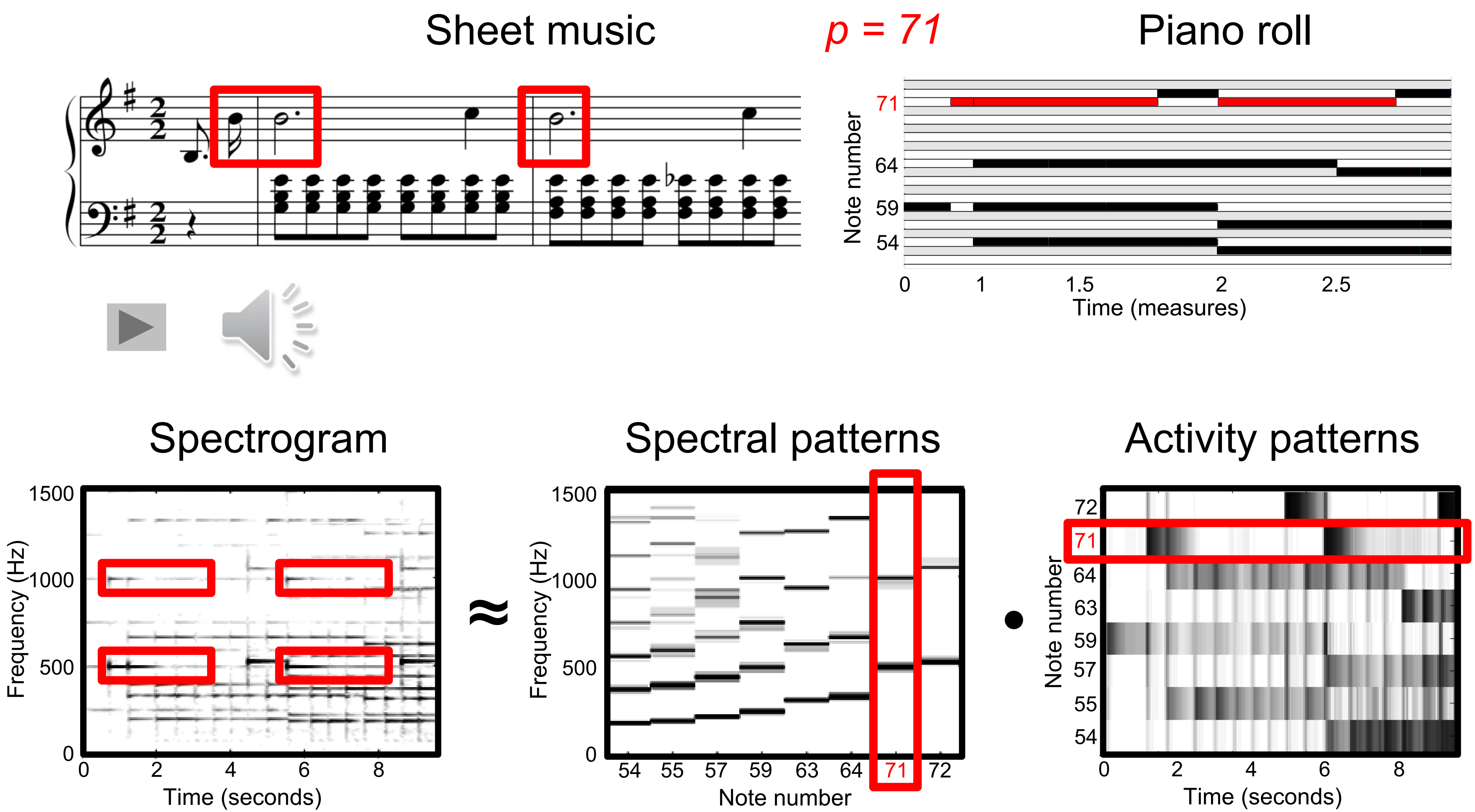


•

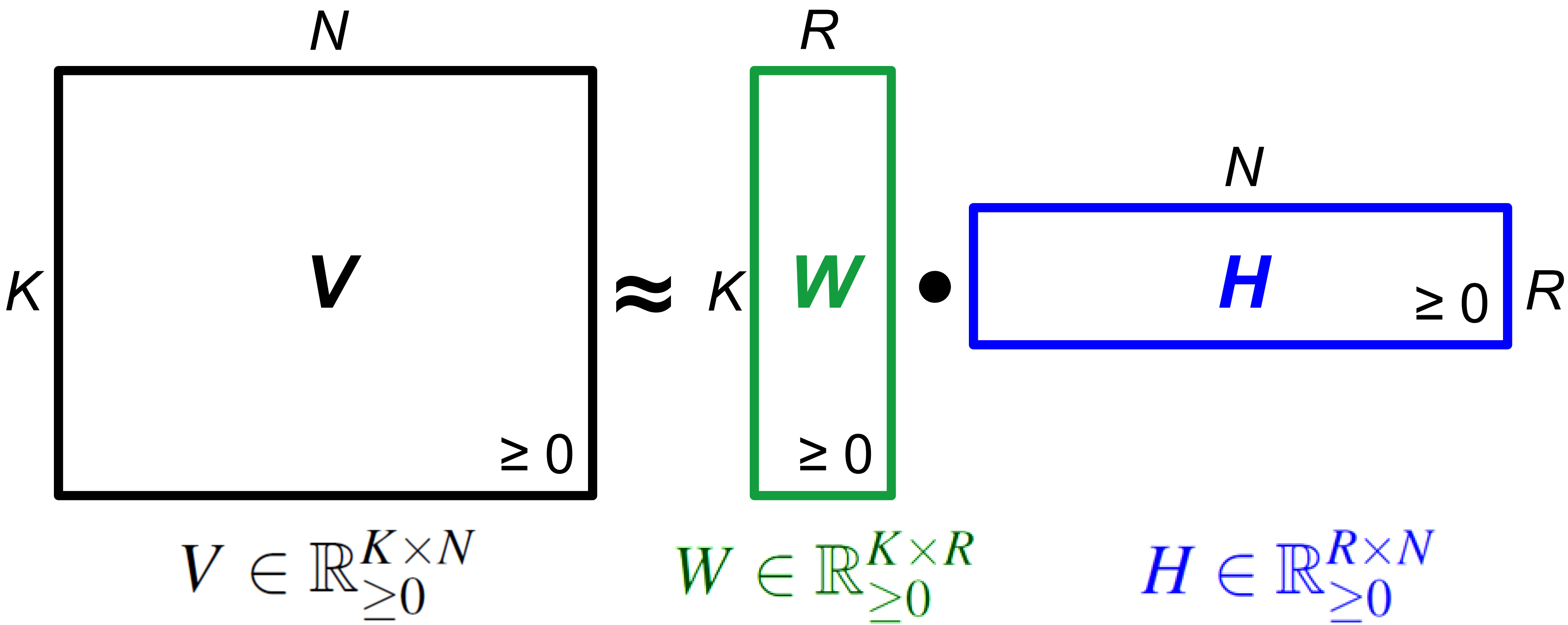
Activity patterns



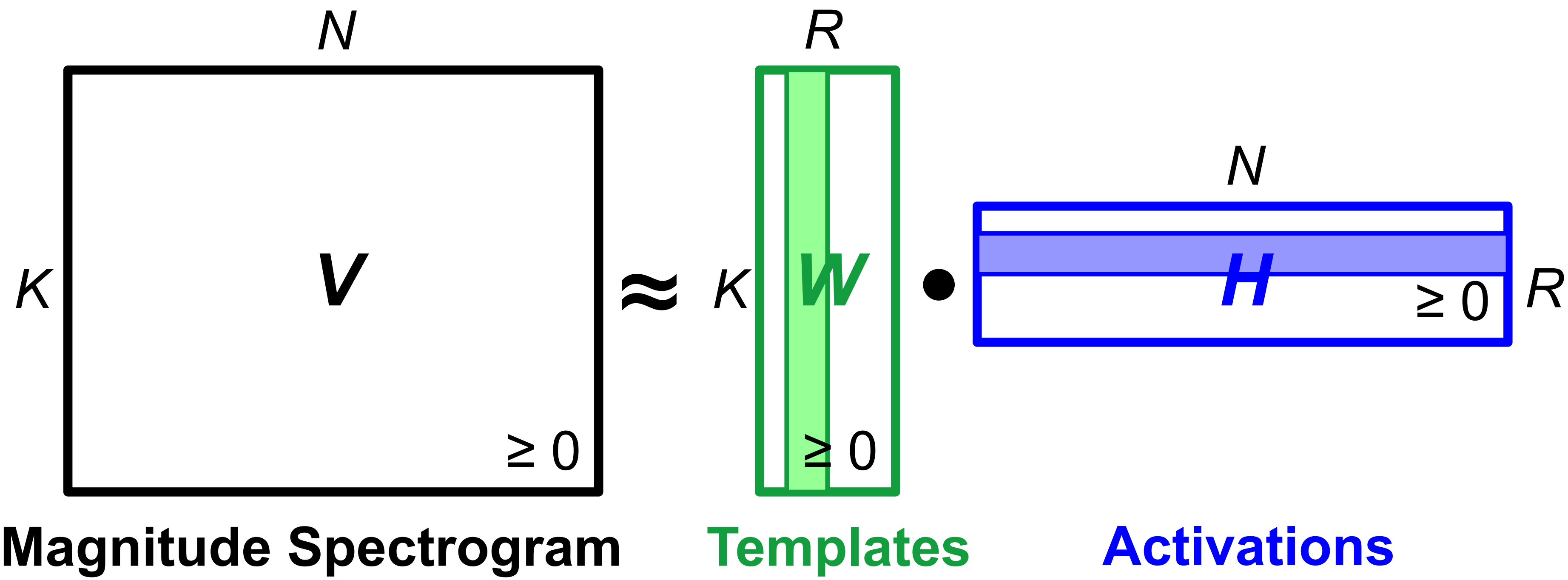
Score-Informed Audio Decomposition



Nonnegative Matrix Factorization (NMF)



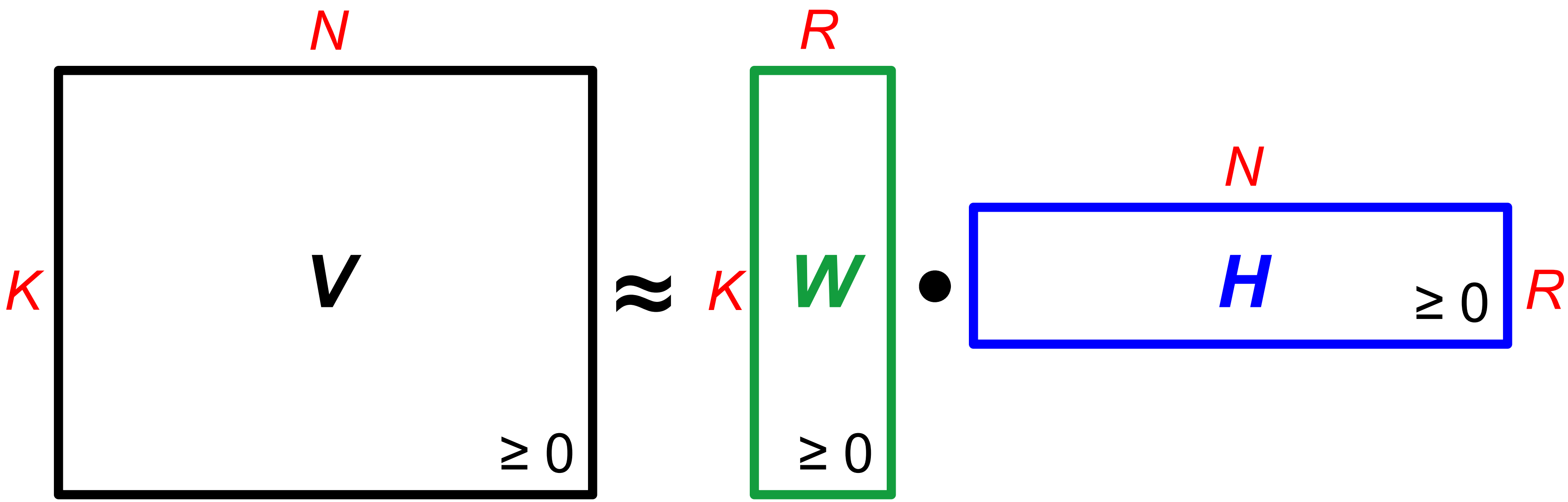
Nonnegative Matrix Factorization (NMF)



Templates: Pitch + Timbre “How does it sound”

Activations: Onset time + Duration “When does it sound”

Nonnegative Matrix Factorization (NMF)

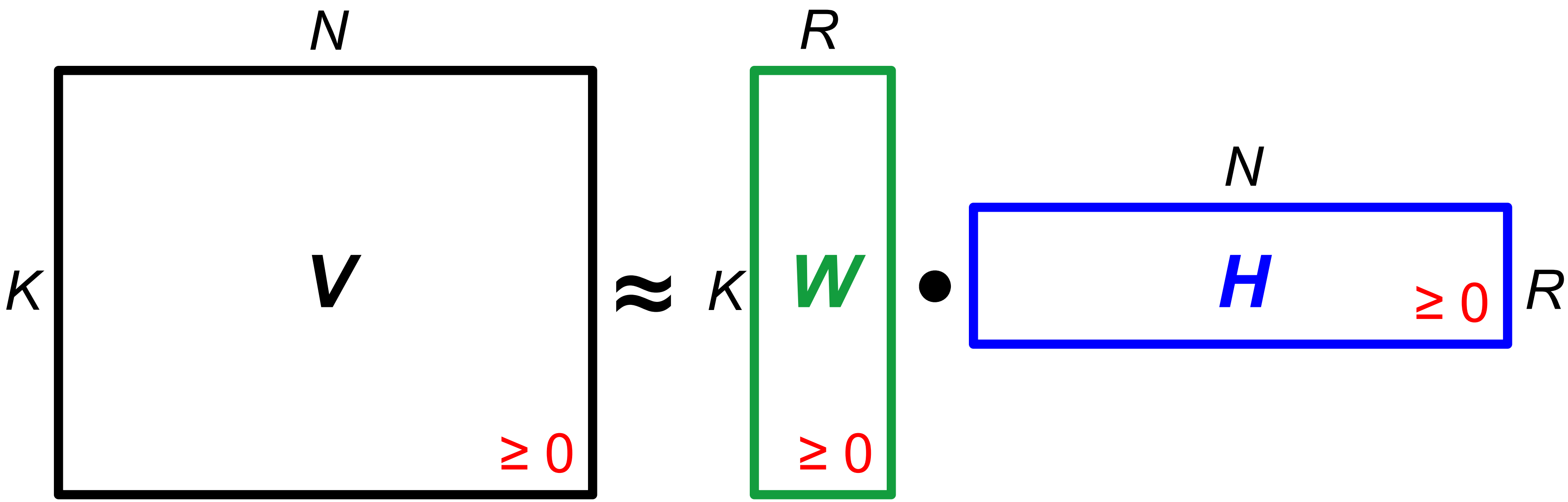


$V \in \mathbb{R}_{\geq 0}^{K \times N}$ $W \in \mathbb{R}_{\geq 0}^{K \times R}$ $H \in \mathbb{R}_{\geq 0}^{R \times N}$

Dimensionality reduction

- *K*, *N* typically much larger than *R* (maximal rank)
- Example: *N = 1000, K = 500, R = 20*
  K x N = 500,000, *K x R = 10,000,* *R x N = 20,000*

Nonnegative Matrix Factorization (NMF)



$V \in \mathbb{R}_{\geq 0}^{K \times N}$ $W \in \mathbb{R}_{\geq 0}^{K \times R}$ $H \in \mathbb{R}_{\geq 0}^{R \times N}$

Nonnegativity:

- Prevents mutual cancellation of template vectors
- Encourages semantically meaningful decomposition

NMF Optimization

Optimization problem:

Given $V \in \mathbb{R}_{\geq 0}^{K \times N}$ and rank parameter *R* minimize

$$\|V - WH\|^2$$

with respect to $W \in \mathbb{R}_{\geq 0}^{K \times R}$ and $H \in \mathbb{R}_{\geq 0}^{R \times N}$.

Optimization not easy:

- Nonnegativity constraints
- Nonconvexity when jointly optimizing *W* and *H*

Strategy: Iteratively optimize *W* and *H* via gradient descent

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \to \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1:R]$$

$$\nu \in [1:N]$$

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^{K} \sum_{n=1}^{N} \left(V_{kn} - \sum_{r=1}^{R} W_{kr} H_{rn} \right)^2 \right)}{\partial H_{\rho\nu}}$$

Variables

$H \in \mathbb{R}^{R\times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^{K} \sum_{n=1}^{N} \left(V_{kn} - \sum_{r=1}^{R} W_{kr} H_{rn} \right)^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^{K} \left(V_{k\nu} - \sum_{r=1}^{R} W_{kr} H_{r\nu} \right)^2 \right)}{\partial H_{\rho\nu}}$$

Summand that does not depend on $H_{\rho\nu}$ must be zero

Variables

$H \in \mathbb{R}^{R \times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^{K} \sum_{n=1}^{N} \left(V_{kn} - \sum_{r=1}^{R} W_{kr} H_{rn}\right)^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^{K} \left(V_{k\nu} - \sum_{r=1}^{R} W_{kr} H_{r\nu}\right)^2 \right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^{K} 2 \left(V_{k\nu} - \sum_{r=1}^{R} W_{kr} H_{r\nu} \right) \cdot (-W_{k\rho})$$

Variables

$H \in \mathbb{R}^{R \times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

Apply chain rule from calculus

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial\left(\sum_{k=1}^K \sum_{n=1}^N \left(V_{kn} - \sum_{r=1}^R W_{kr} H_{rn}\right)^2\right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial\left(\sum_{k=1}^K \left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu}\right)^2\right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^K 2\left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu}\right) \cdot (-W_{k\rho})$$

$$= 2\left(\sum_{r=1}^R \sum_{k=1}^K W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^K W_{k\rho} V_{k\nu}\right)$$

Rearrange summands

Variables

$H \in \mathbb{R}^{R \times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^{K} \sum_{n=1}^{N} \left(V_{kn} - \sum_{r=1}^{R} W_{kr} H_{rn}\right)^2\right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^{K} \left(V_{k\nu} - \sum_{r=1}^{R} W_{kr} H_{r\nu}\right)^2\right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^{K} 2\left(V_{k\nu} - \sum_{r=1}^{R} W_{kr} H_{r\nu}\right) \cdot (-W_{k\rho})$$

$$= 2\left(\sum_{r=1}^{R} \sum_{k=1}^{K} W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^{K} W_{k\rho} V_{k\nu}\right)$$

$$= 2\left(\sum_{r=1}^{R} \left(\sum_{k=1}^{K} W_{\rho k}^{\top} W_{kr}\right) H_{r\nu} - \sum_{k=1}^{K} W_{\rho k}^{\top} V_{k\nu}\right)$$

Variables

$H \in \mathbb{R}^{R \times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

Introduce transposed $W^{\top}$

NMF Optimization

Computation of gradient with respect to *H* (fixed *W*)

$D := RN$

$\varphi^W : \mathbb{R}^D \to \mathbb{R}$

$\varphi^W(H) := \|V - WH\|^2$

Variables

$H \in \mathbb{R}^{R \times N}$

$H_{\rho\nu}$

$\rho \in [1:R]$

$\nu \in [1:N]$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N \left(V_{kn} - \sum_{r=1}^R W_{kr} H_{rn}\right)^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K \left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu}\right)^2 \right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^K 2\left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu}\right) \cdot (-W_{k\rho})$$

$$= 2\left(\sum_{r=1}^R \sum_{k=1}^K W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^K W_{k\rho} V_{k\nu}\right)$$

$$= 2\left(\sum_{r=1}^R \left(\sum_{k=1}^K W_{\rho k}^\top W_{kr}\right) H_{r\nu} - \sum_{k=1}^K W_{\rho k}^\top V_{k\nu}\right)$$

$$= 2\left((W^\top W H)_{\rho\nu} - (W^\top V)_{\rho\nu}\right).$$

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \ldots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left(\left(W^\top W H^{(\ell)}\right)_{rn} - \left(W^\top V\right)_{rn} \right)$$

with suitable learning rate $\gamma_{rn}^{(\ell)} \geq 0$

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \ldots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left(\left(W^\top W H^{(\ell)} \right)_{rn} - \left(W^\top V \right)_{rn} \right)$$

with suitable learning rate $\gamma_{rn}^{(\ell)} \geq 0$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

NMF Optimization

Gradient descent

Choose adaptive learning rate:

$$\gamma_{rn}^{(\ell)} := \frac{H_{rn}^{(\ell)}}{\left(W^\top W H^{(\ell)}\right)_{rn}}$$

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \ldots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left(\left(W^\top W H^{(\ell)}\right)_{rn} - \left(W^\top V\right)_{rn} \right)$$

$$= H_{rn}^{(\ell)} \cdot \frac{\left(W^\top V\right)_{rn}}{\left(W^\top W H^{(\ell)}\right)_{rn}}$$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \ldots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left(\left(W^\top W H^{(\ell)}\right)_{rn} - \left(W^\top V\right)_{rn} \right)$$

$$= H_{rn}^{(\ell)} \cdot \frac{\left(W^\top V\right)_{rn}}{\left(W^\top W H^{(\ell)}\right)_{rn}}$$

Choose adaptive learning rate:

$$\gamma_{rn}^{(\ell)} := \frac{H_{rn}^{(\ell)}}{\left(W^\top W H^{(\ell)}\right)_{rn}}$$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

- Update rule become multiplicative
- Nonnegative values stay nonnegative

NMF Optimization

Algorithm: NMF ($V \approx WH$)

Input: Nonnegative matrix V of size $K \times N$
Rank parameter $R \in \mathbb{N}$
Threshold ε used as stop criterion

Output: Nonnegative template matrix W of size $K \times R$
Nonnegative activation matrix H of size $R \times N$

Procedure: Define nonnegative matrices $W^{(0)}$ and $H^{(0)}$ by some random or informed initialization. Furthermore set $\ell = 0$. Apply the following update rules (written in matrix notation):

(1) $H^{(\ell+1)} = H^{(\ell)} \odot \left(((W^{(\ell)})^\top V) \oslash ((W^{(\ell)})^\top W^{(\ell)} H^{(\ell)})\right)$

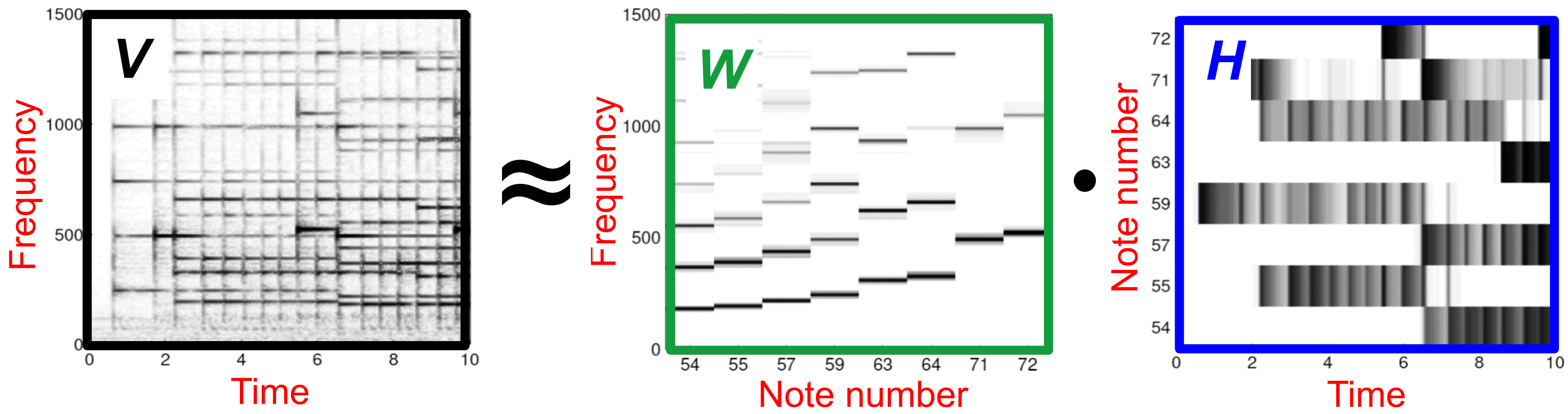
(2) $W^{(\ell+1)} = W^{(\ell)} \odot \left((V(H^{(\ell+1)})^\top) \oslash (W^{(\ell)} H^{(\ell+1)} (H^{(\ell+1)})^\top)\right)$

(3) Increase ℓ by one.

Repeat the steps (1) to (3) until $\|H^{(\ell)} - H^{(\ell-1)}\| \leq \varepsilon$ and $\|W^{(\ell)} - W^{(\ell-1)}\| \leq \varepsilon$ (or until some other stop criterion is fulfilled). Finally, set $H = H^{(\ell)}$ and $W = W^{(\ell)}$.

Lee, Seung: Algorithms for Non-Negative Matrix Factorization. Proc. NIPS, 2000.

NMF-based Spectrogram Decomposition

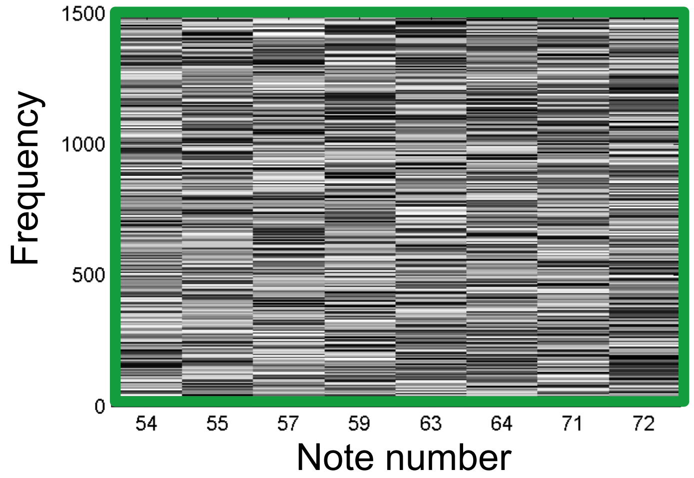


Templates: Pitch + Timbre **“How does it sound”**

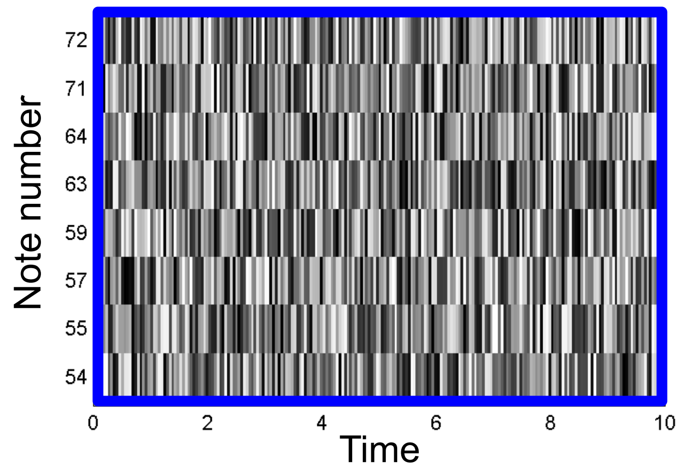
Activations: Onset time + Duration **“When does it sound”**

NMF-based Spectrogram Decomposition

Template initialization

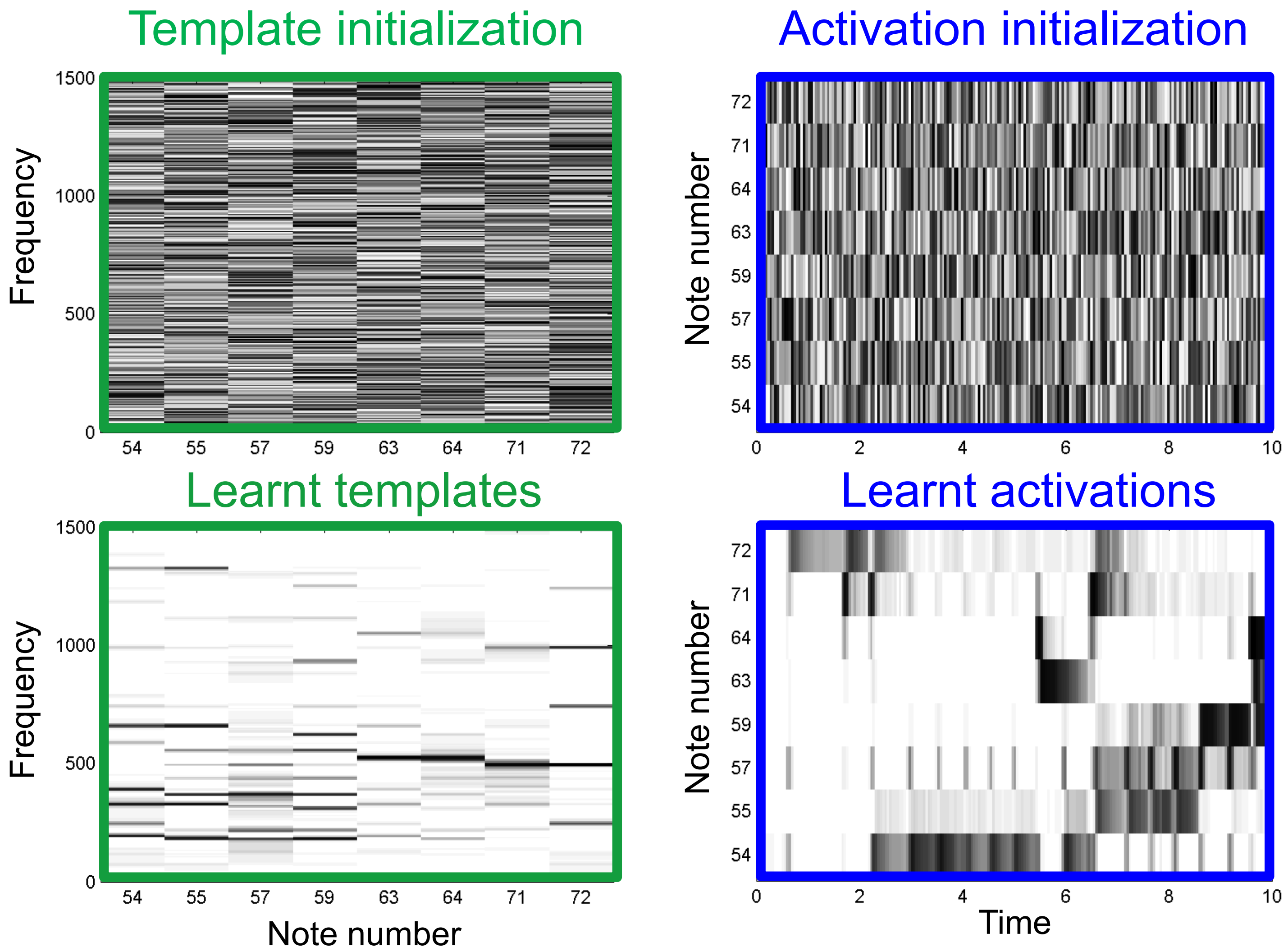


Activation initialization



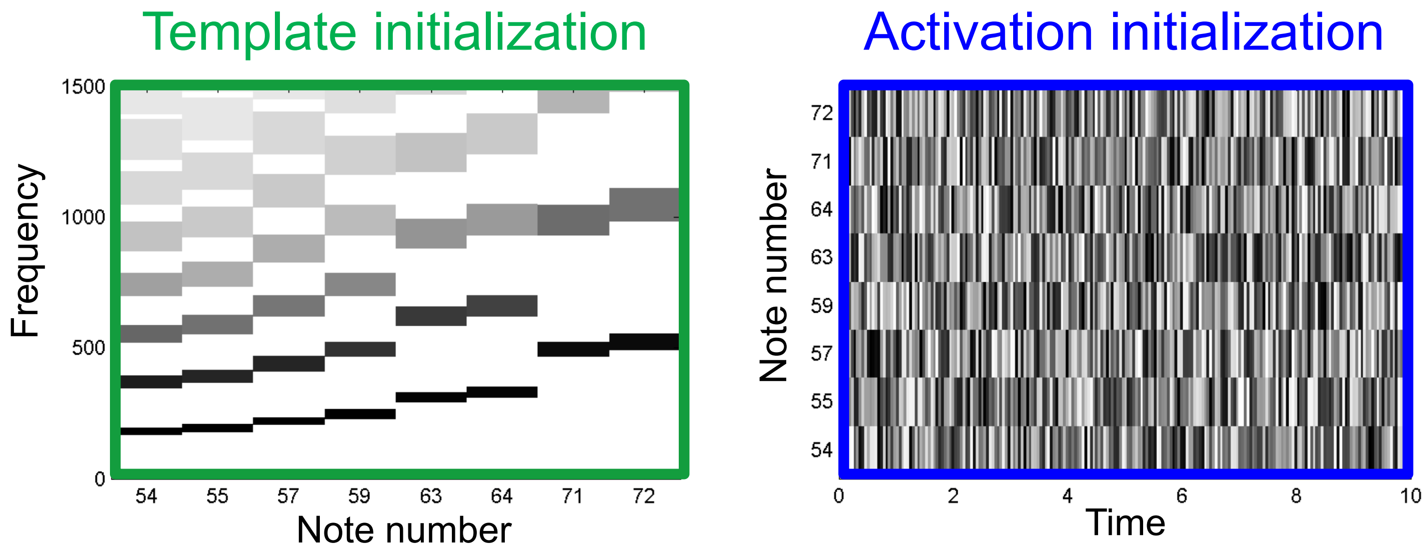
Random initialization

NMF-based Spectrogram Decomposition



Random initialization → No semantic meaning

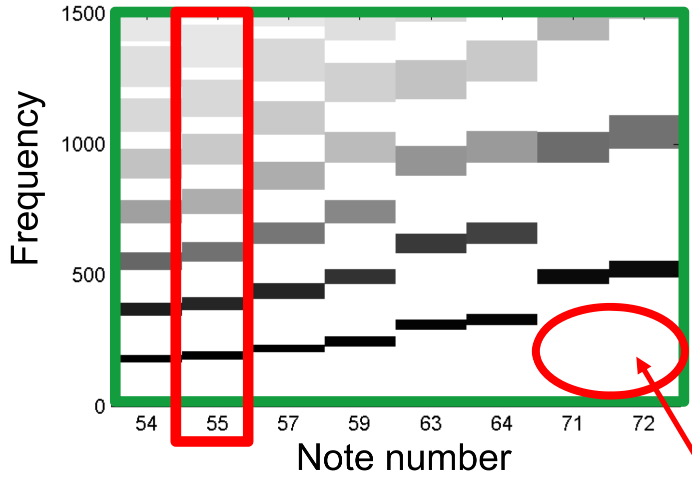
Constrained NMF: Templates



Enforce harmonic structure with zero-valued entries

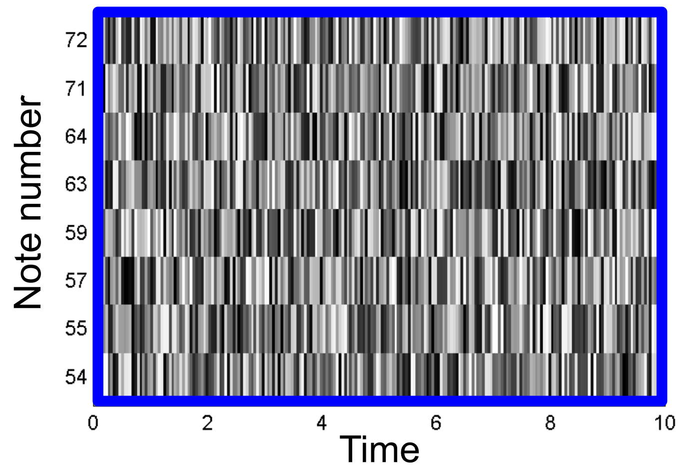
Constrained NMF: Templates

Template initialization



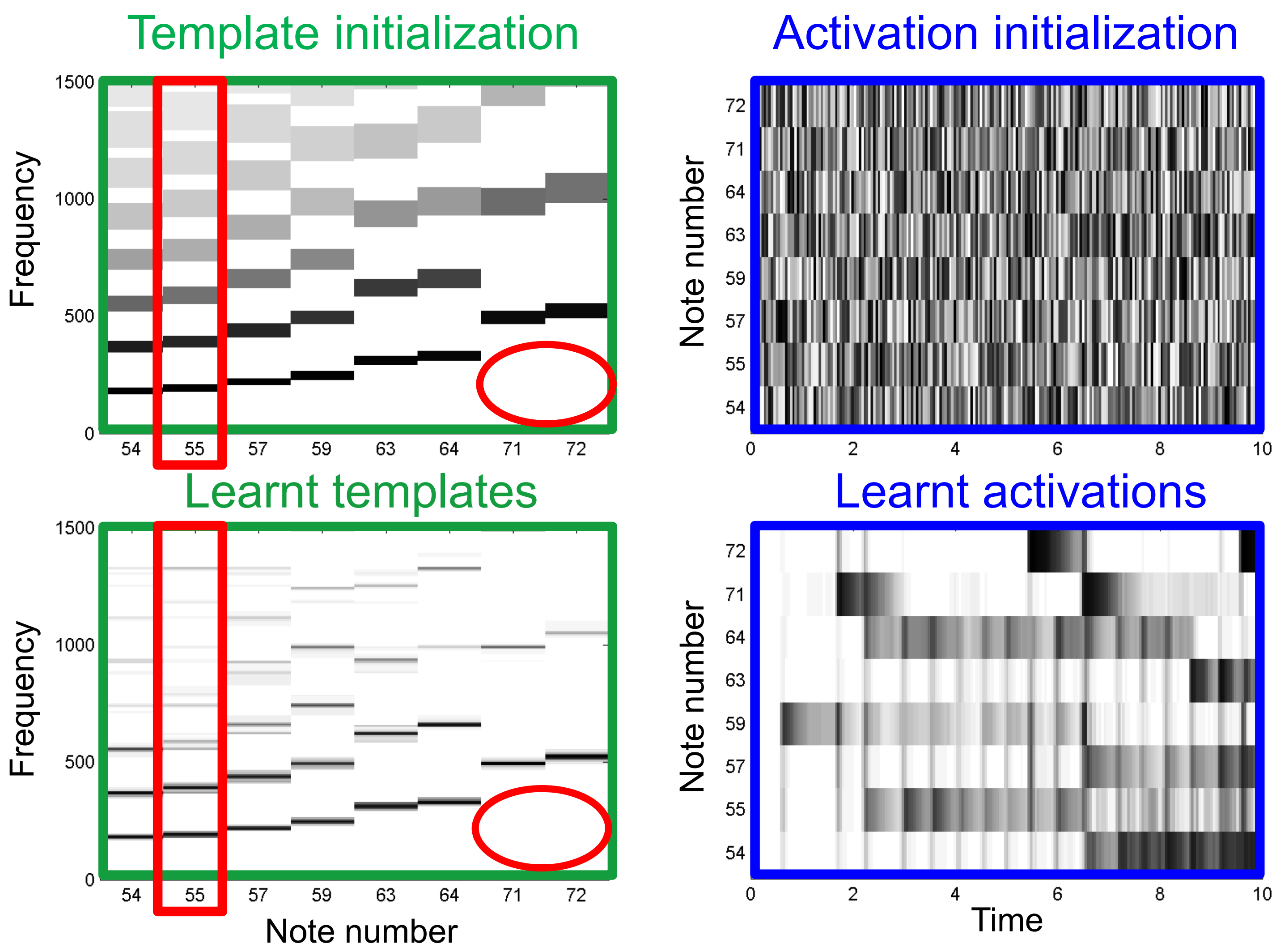
Template constraint for p=55

Activation initialization



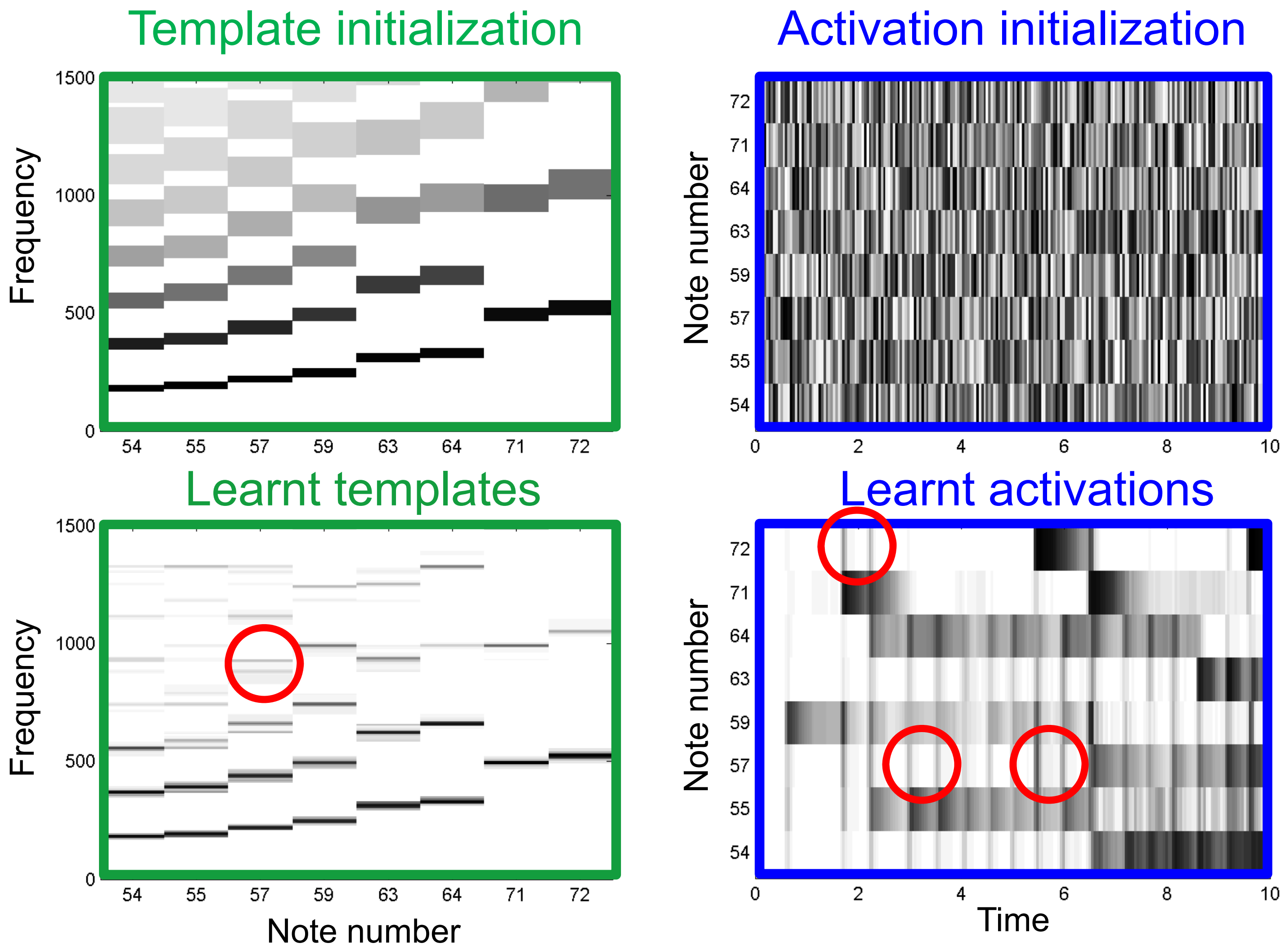
Enforce harmonic structure with zero-valued entries

Constrained NMF: Templates



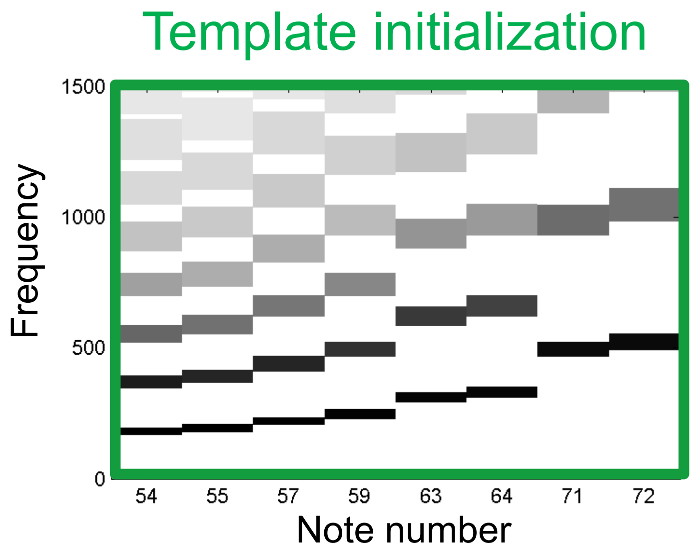
Zero-valued entries remain zero-valued entries!

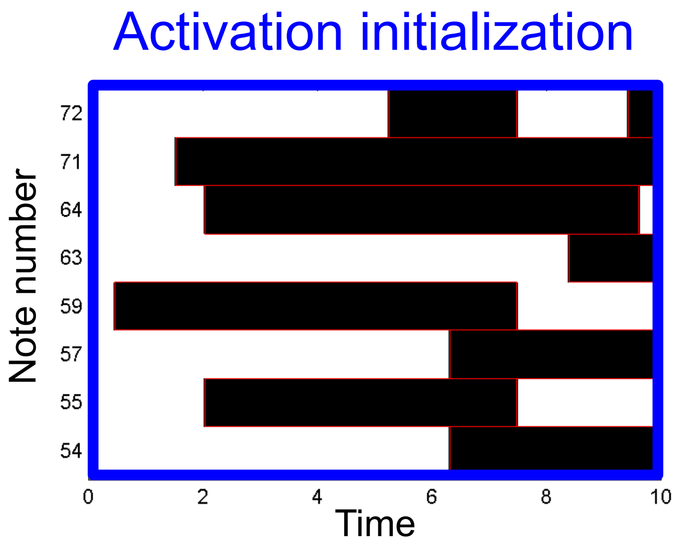
Constrained NMF: Templates



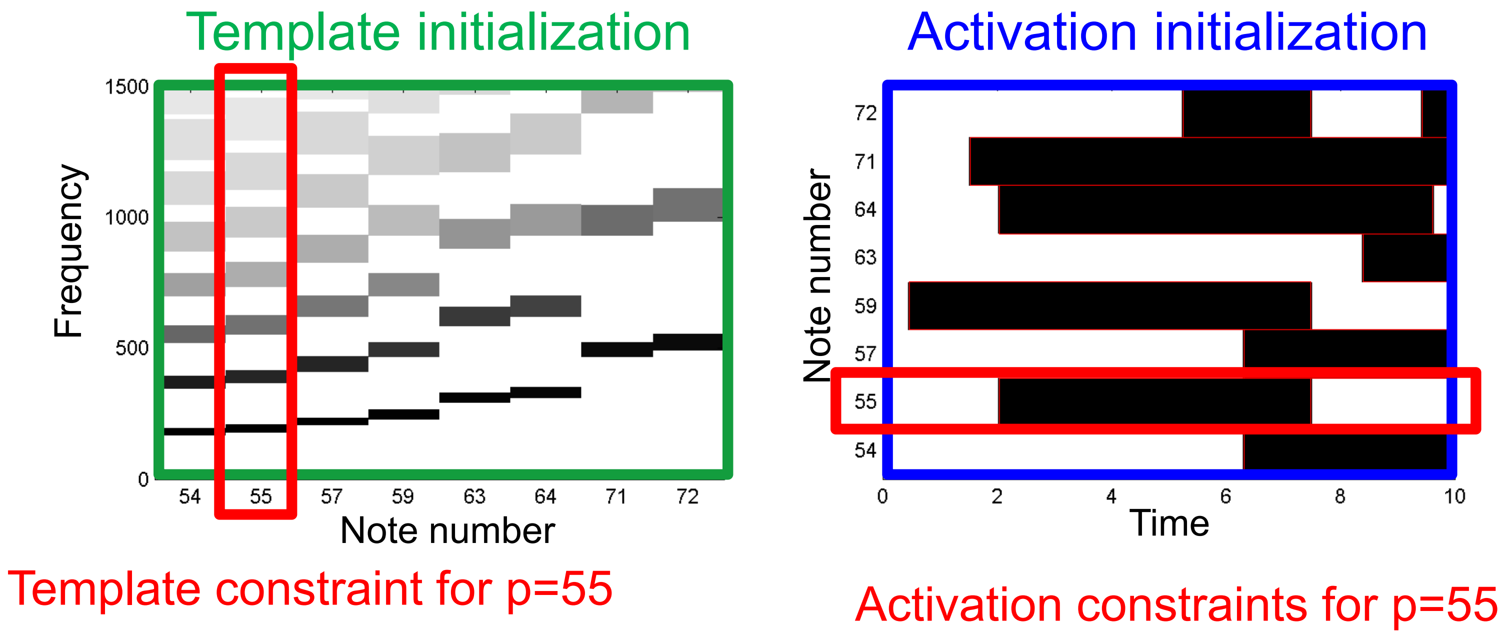
Pitch templates misused to represent onsets

Constrained NMF: Double Constraints



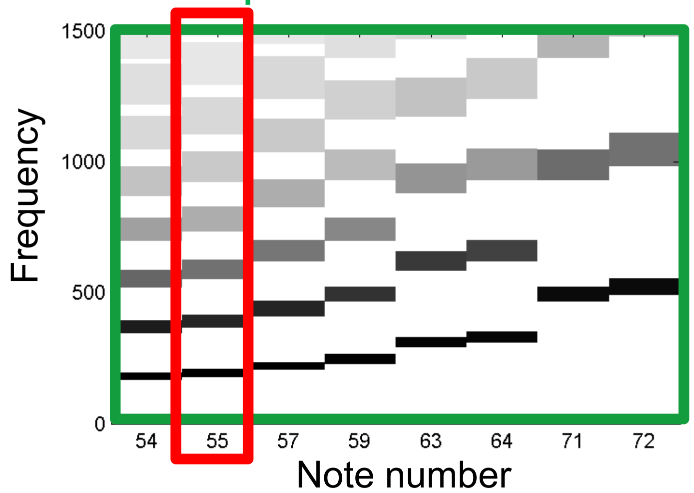


Constrained NMF: Double Constraints



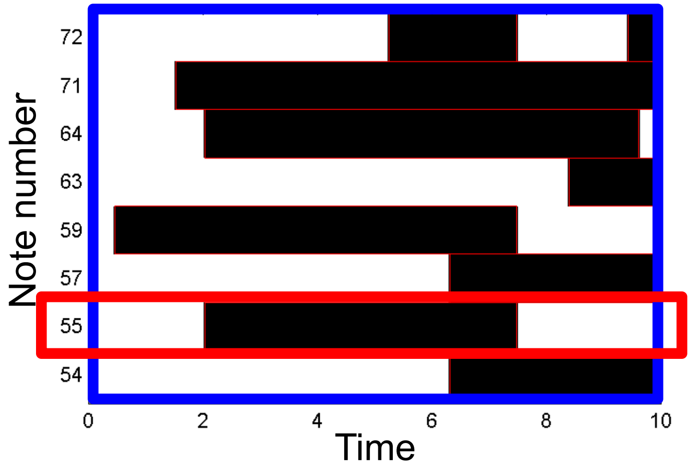
Constrained NMF: Double Constraints

Template initialization



Template constraint for p=55

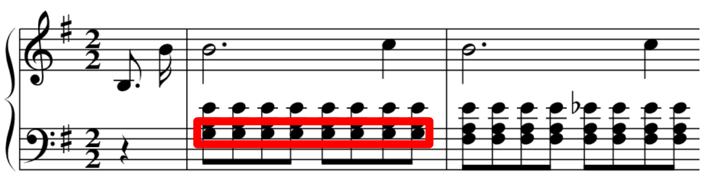
Activation initialization

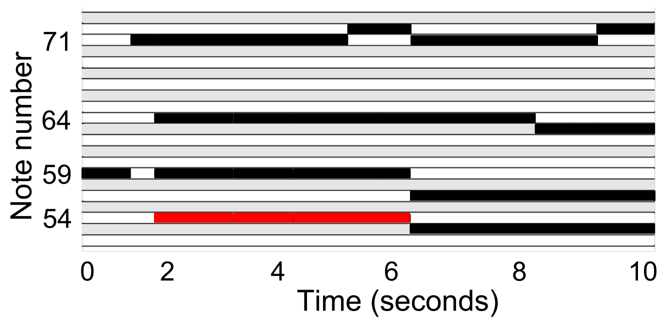


Activation constraints for p=55

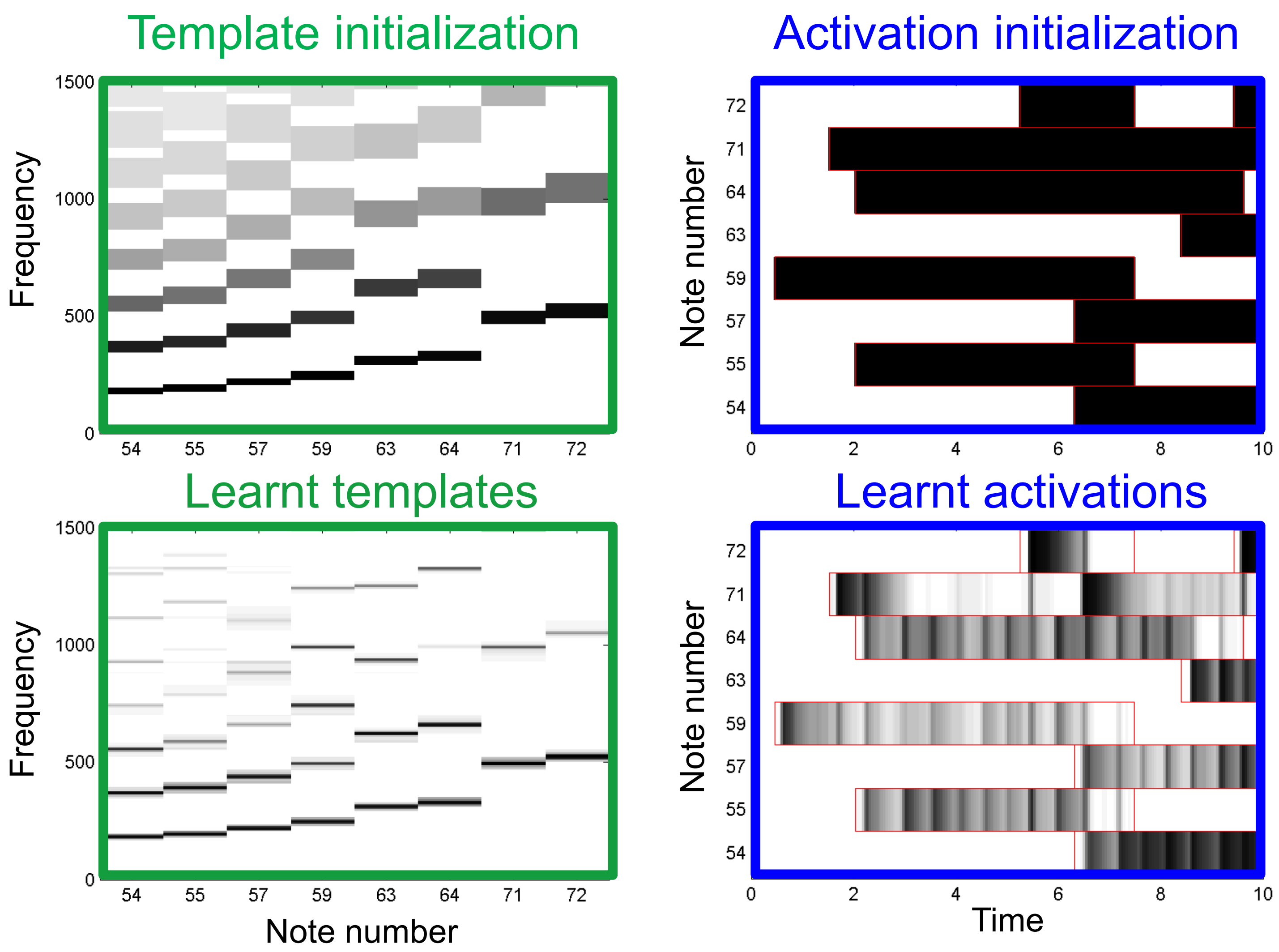
Such information may come from a synchronized score

Sheet music





Constrained NMF: Double Constraints

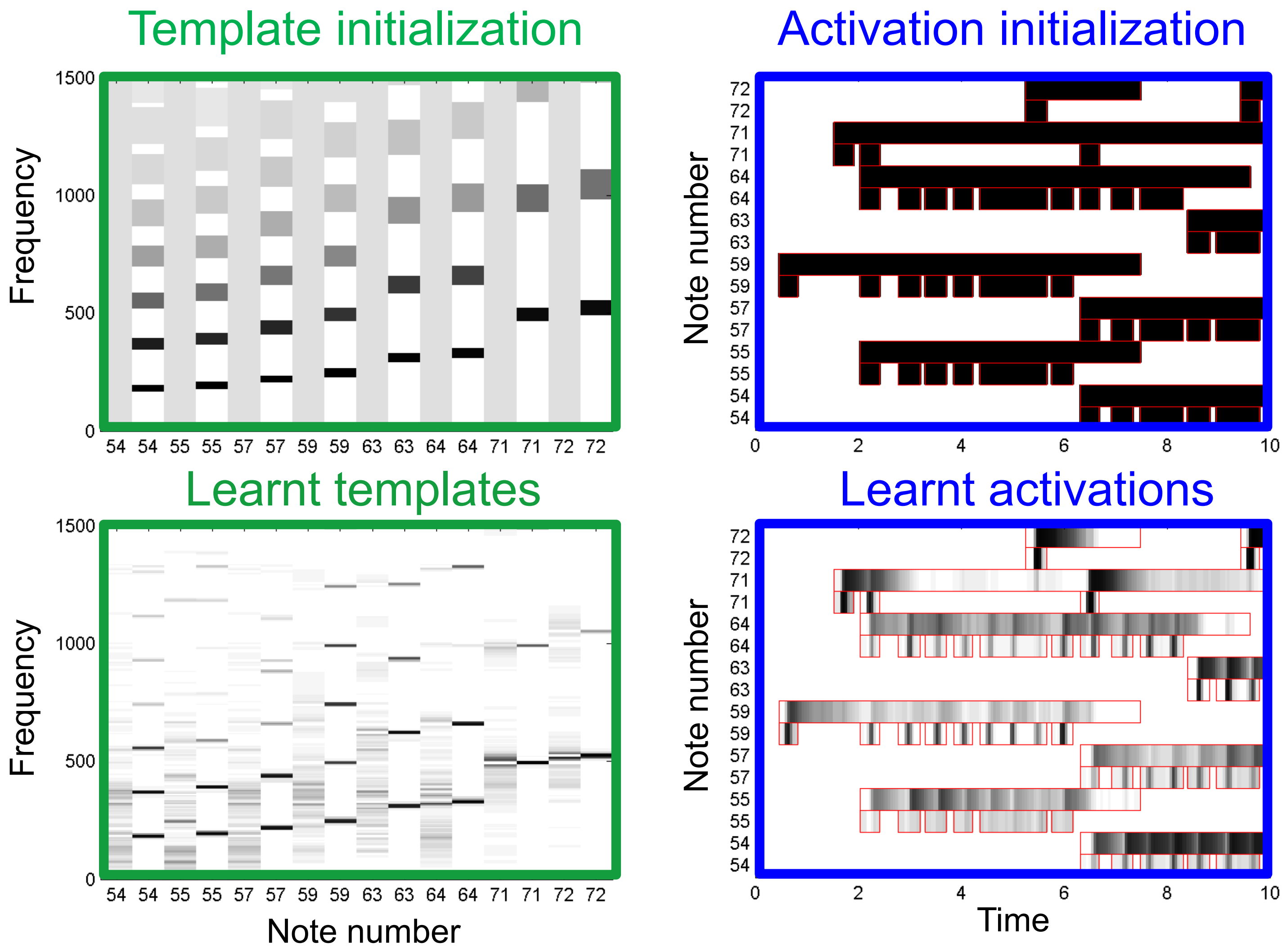


Original

Model

Significant gain in structure, but onsets are missing

Constrained NMF: Onset Templates



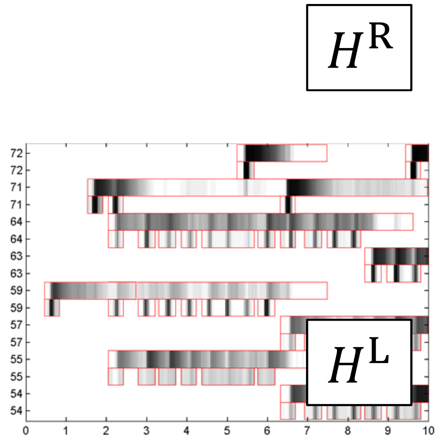
Original

Model Onset

Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

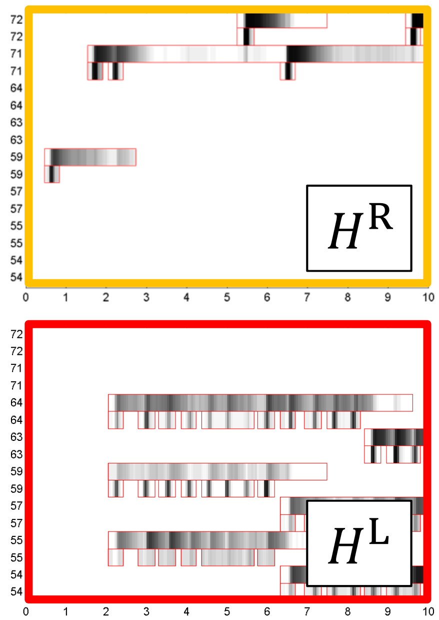
1. Split activation matrix



Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

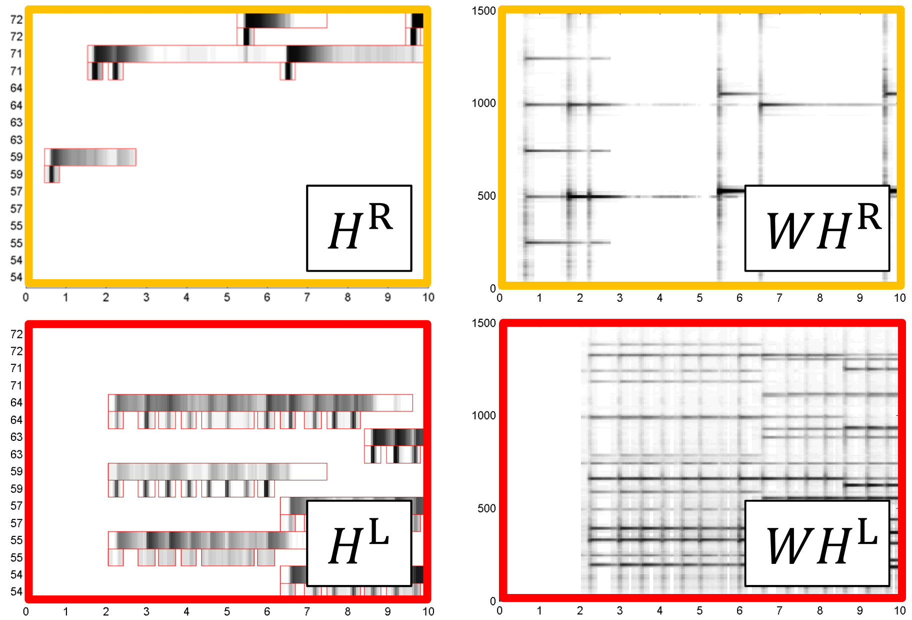
1. Split activation matrix



Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

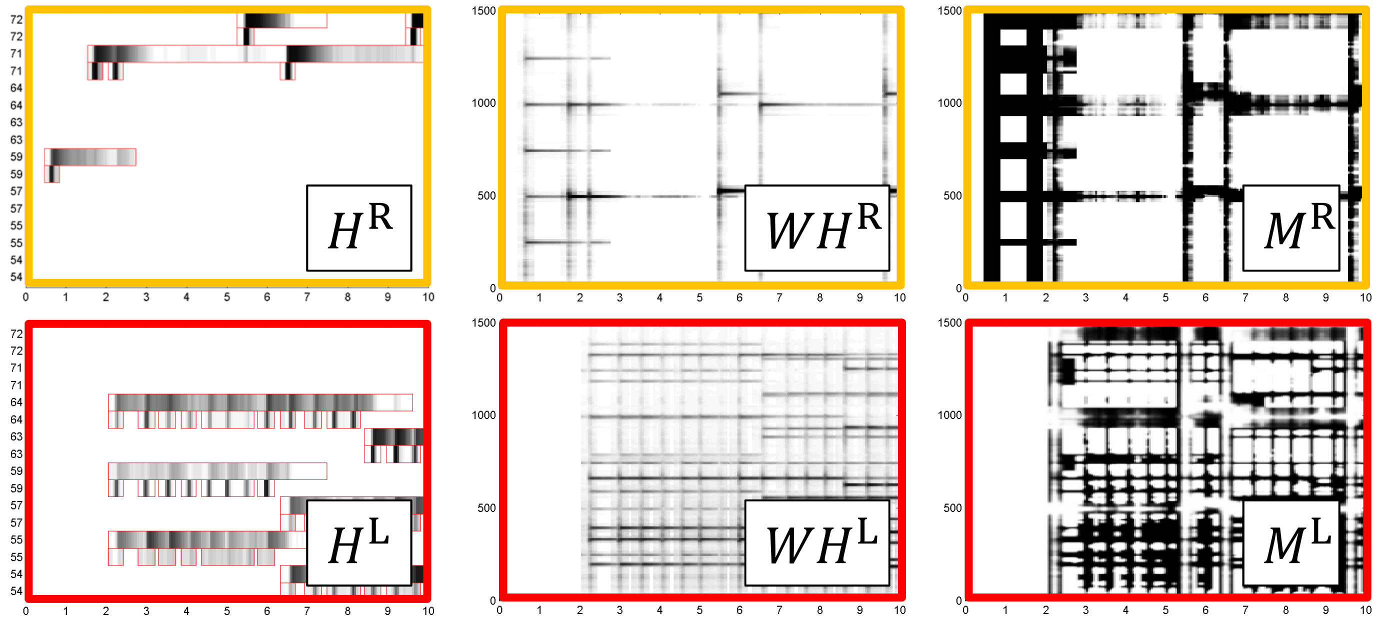
1. Split activation matrix
2. Model spectrogram for left/right



Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

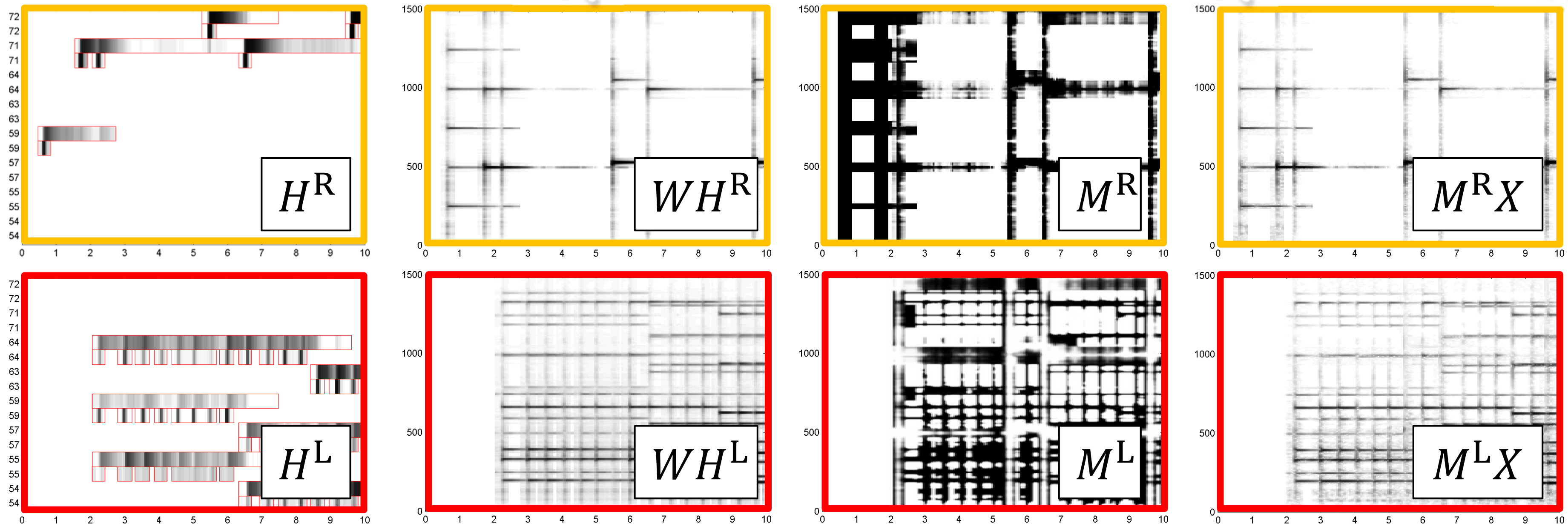
1. Split activation matrix
2. Model spectrogram for left/right
3. Separation masks for left/right



Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

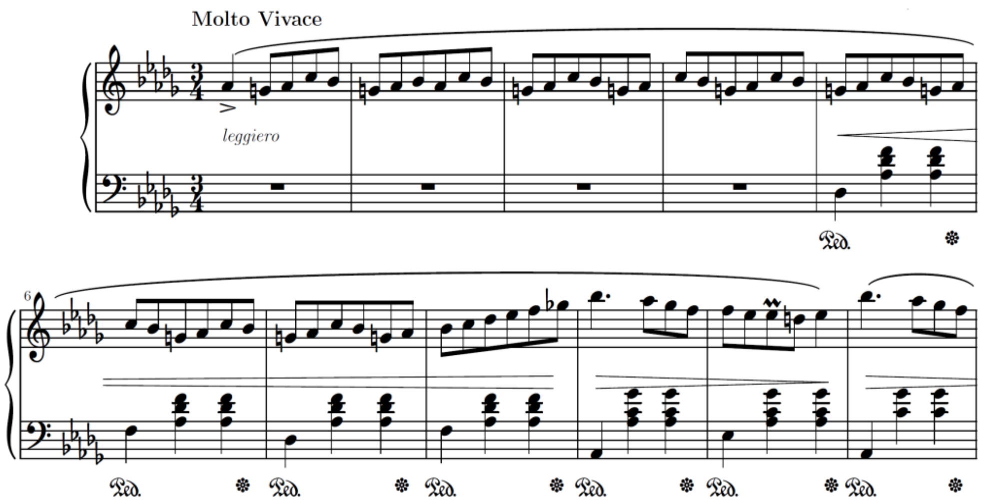
1. Split activation matrix
2. Model spectrogram for left/right
3. Separation masks for left/right
4. Estimated spectrograms for left/right



Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

Chopin, Waltz Op. 64, No. 1



Original

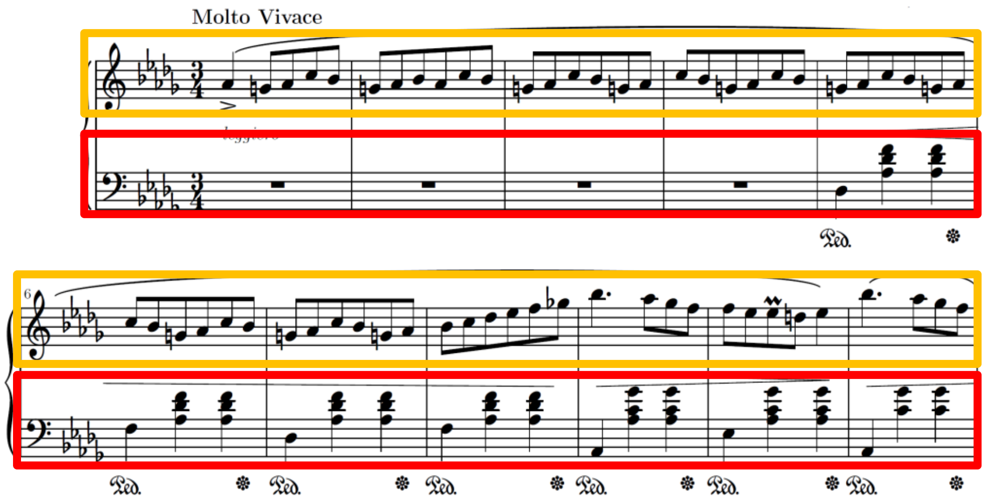
Ewert, Müller: Using Score-Informed Constraints for NMF-based Source Separation. Proc. ICASSP, 2012.

Further results available at
http://www.mpi-inf.mpg.de/resources/MIR/ICASSP2012-ScoreInformedNMF/

Score-Informed Audio Decompostion

Application: Separating left and right hands for piano

Chopin, Waltz Op. 64, No. 1



Original

Left/right hand

Right hand

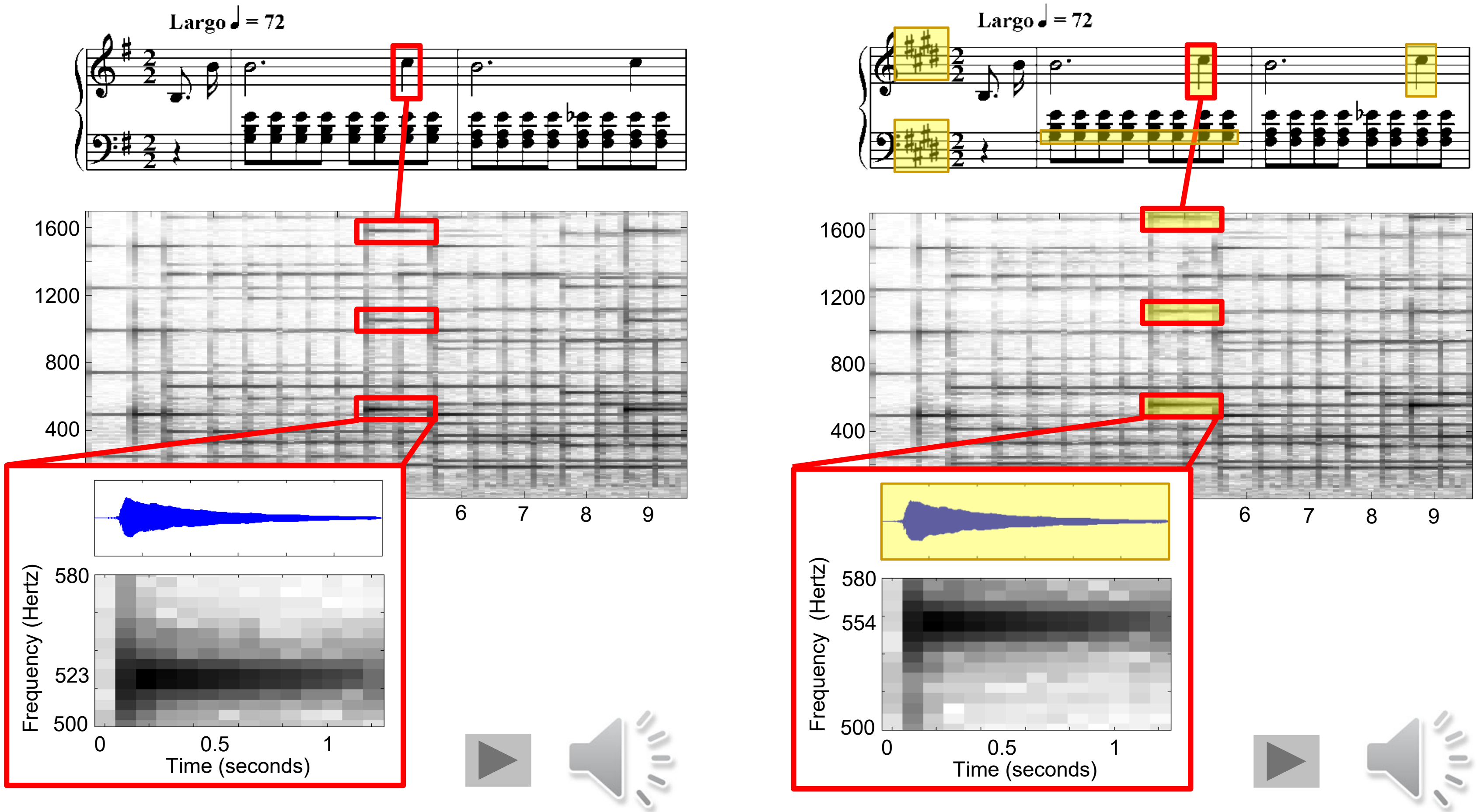
Left hand

Ewert, Müller: Using Score-Informed Constraints for NMF-based Source Separation. Proc. ICASSP, 2012.

Further results available at
http://www.mpi-inf.mpg.de/resources/MIR/ICASSP2012-ScoreInformedNMF/

Score-Informed Audio Decomposition

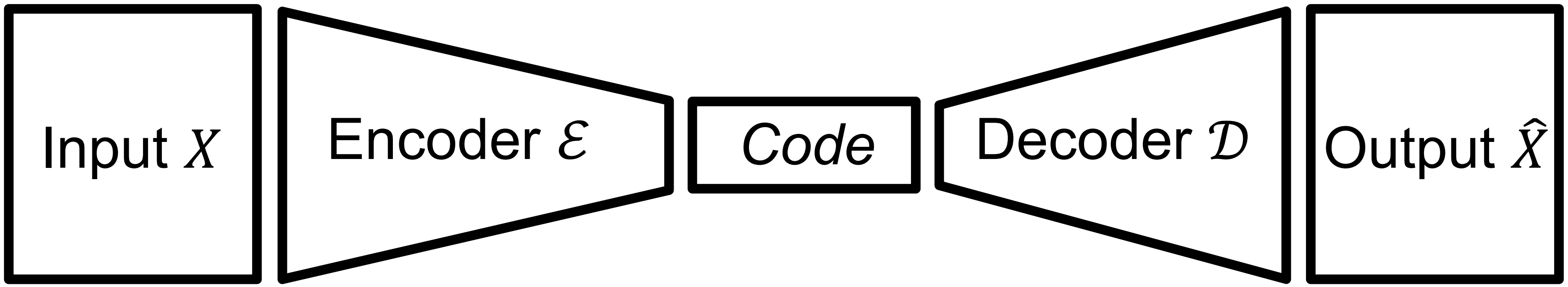
Application: Audio editing



Conclusions (NMF)

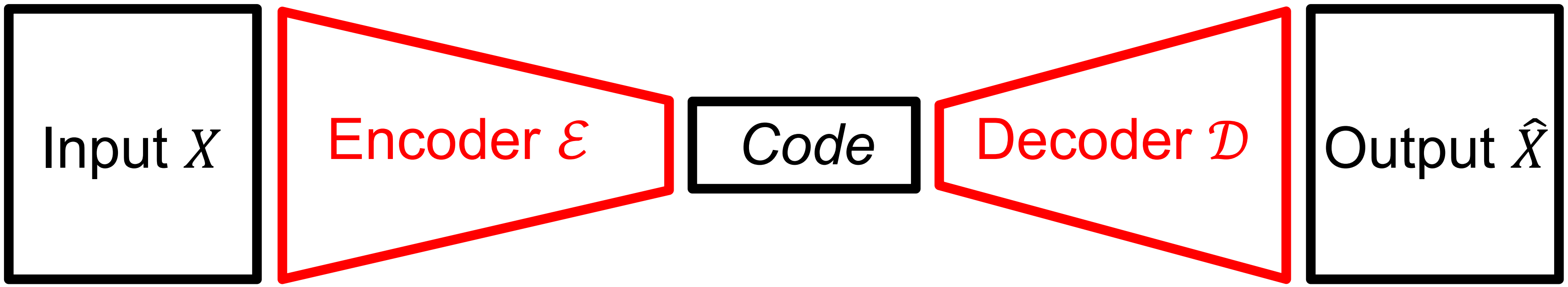
- NMF used for spectrogram decomposition
- Multiplicative update rules make it easy to constrain NMF model via zero initialization
- Exploiting score information to guide separation process (requires score–audio synchronization)
- Application: Separation of arbitrary note groups from given audio recording

Autoencoder



- Specific type of neural network
- Encoder: Compress input X into a low-dimensional code
- Decoder: Reconstruct output $\hat{X}$ from code

Autoencoder



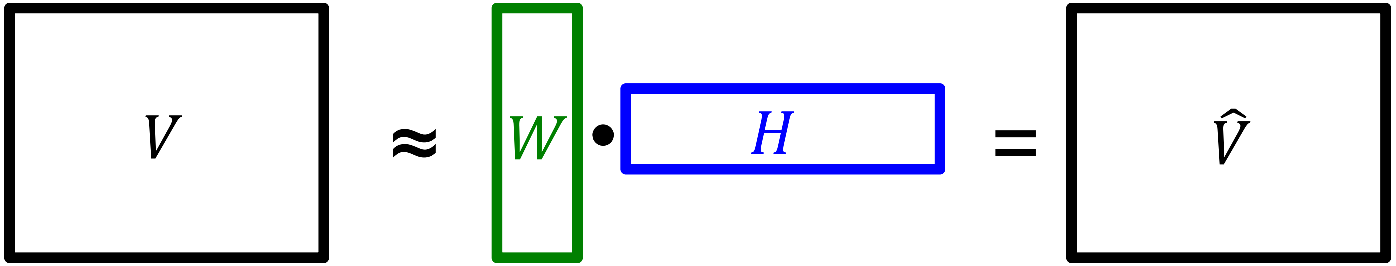
- Specific type of neural network
- Encoder: Compress input X into a low-dimensional code
- Decoder: Reconstruct output $\hat{X}$ from code
- Goal: Learn parameters for encoder and decoder such that output is close to input with respect to some loss function:

$$\mathcal{L}(X, \hat{X}) \approx 0$$

NMF and Autoencoder (AE)

Smaragdis, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.

NMF

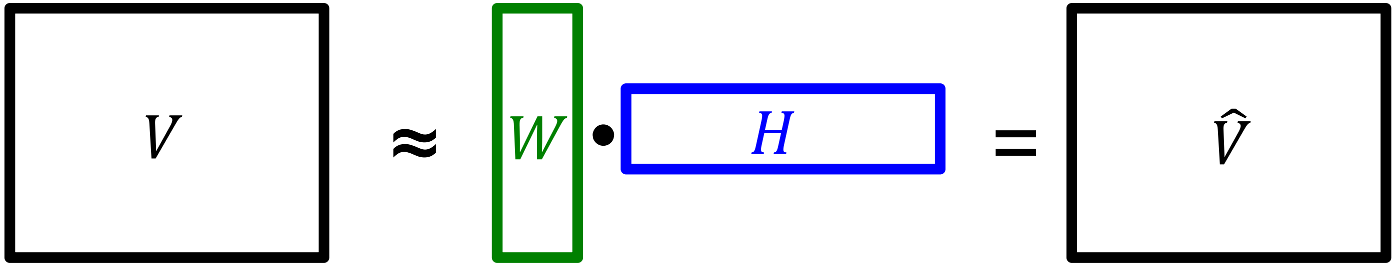


$V \approx WH$ implies $W^{+}V \approx H$ with pseudoinverse W^{+}

NMF and Autoencoder (AE)

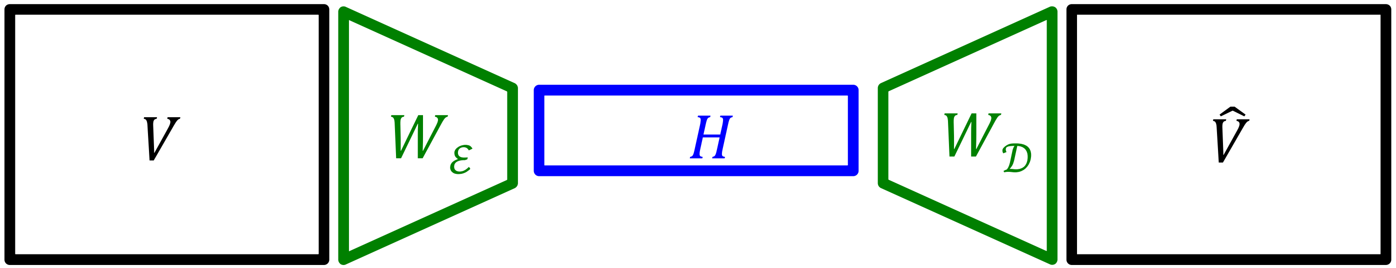
Smaragdis, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.

NMF



$V \approx WH$ implies $W^{+}V \approx H$ with pseudoinverse W^{+}

AE

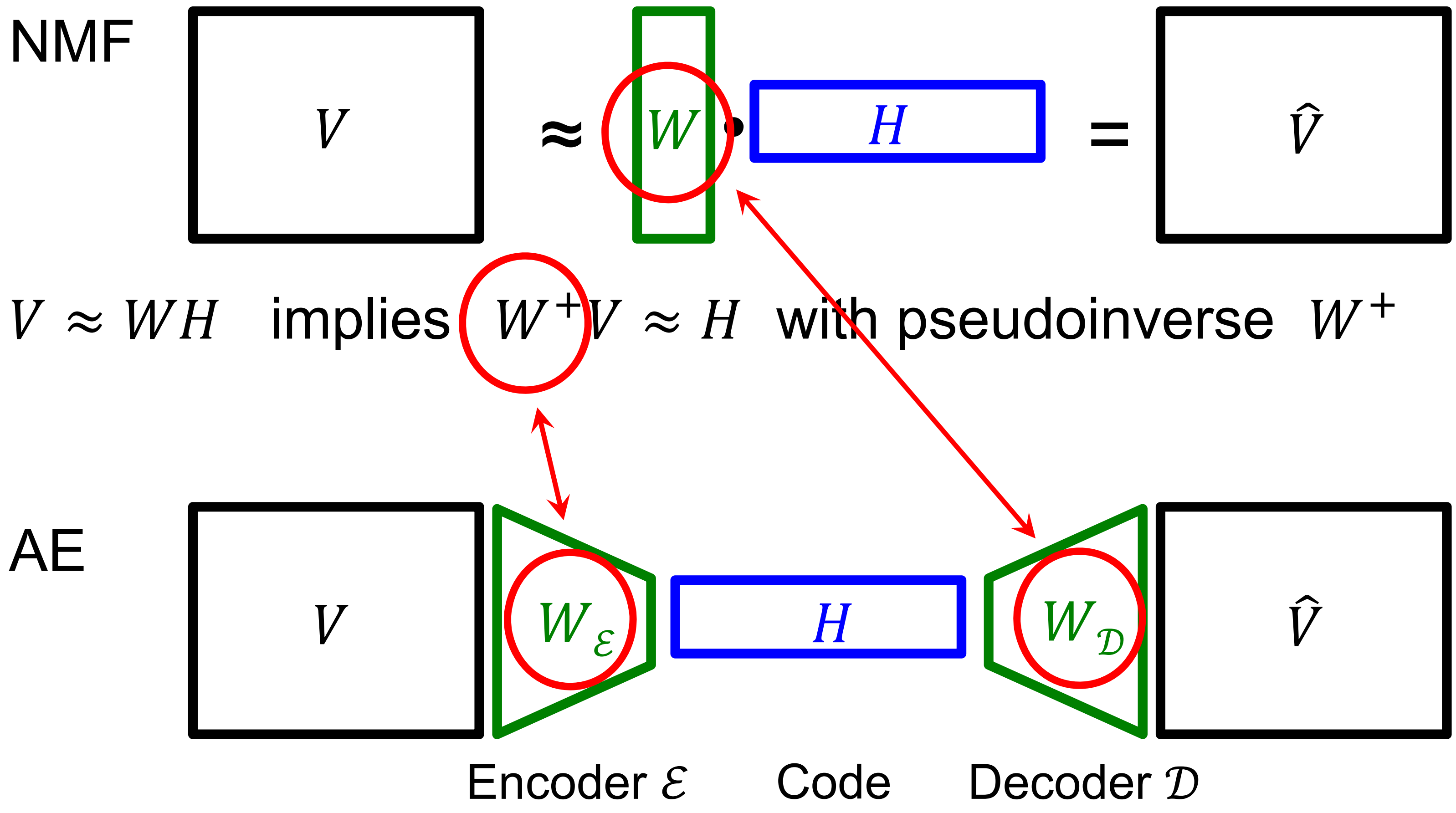


Encoder $\mathcal{E}$ Code Decoder $\mathcal{D}$

1. Layer: $H = W_{\mathcal{E}} V$
2. Layer: $\hat{V} = W_{\mathcal{D}} H$

NMF and Autoencoder (AE)

Smaragdis, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.



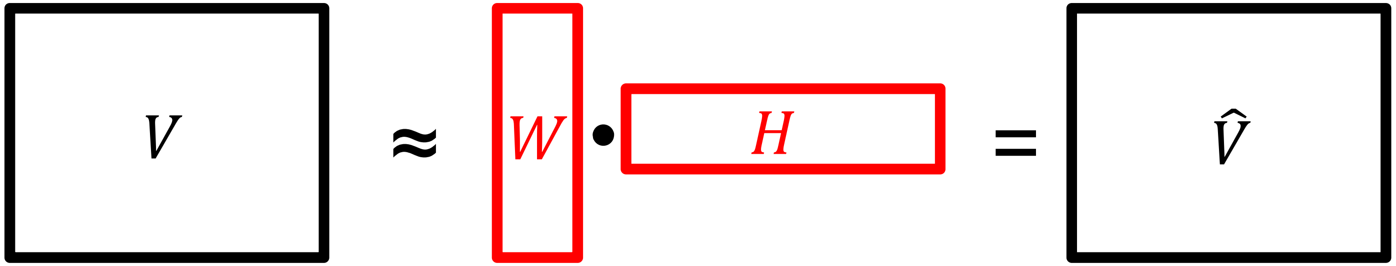
1. Layer: $H = W_{\mathcal{E}} V$
2. Layer: $\hat{V} = W_{\mathcal{D}} H$

Fully connected network

NMF and Autoencoder (AE)

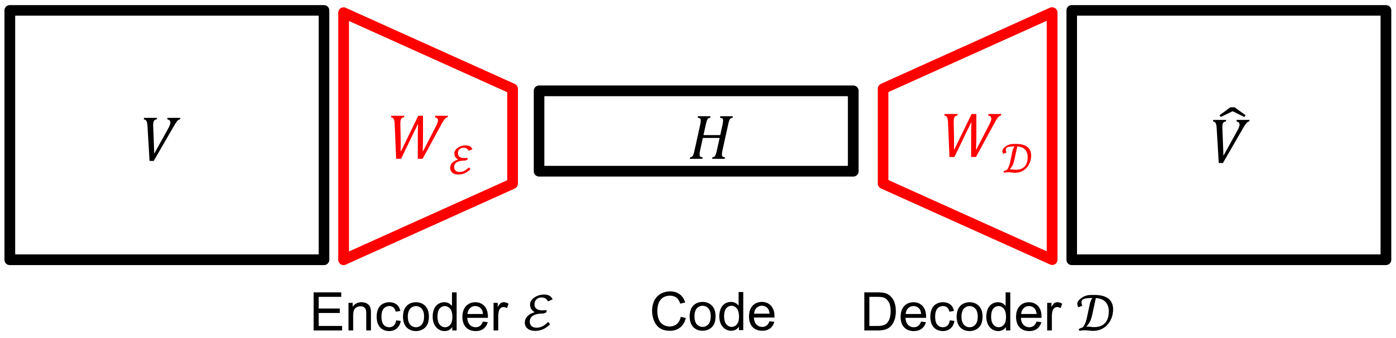
Smaragdis, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.

NMF



$V \approx WH$ implies $W^{+}V \approx H$ with pseudoinverse W^{+}

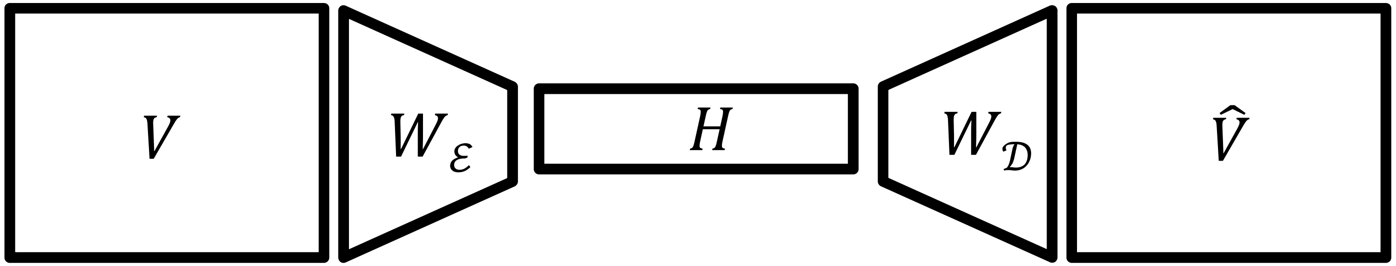
AE



1. Layer: $H = W_{\mathcal{E}} V$
2. Layer: $\hat{V} = W_{\mathcal{D}} H$

NMF: Learn H and W
AE: Learn $W_{\mathcal{E}}$ and $W_{\mathcal{D}}$

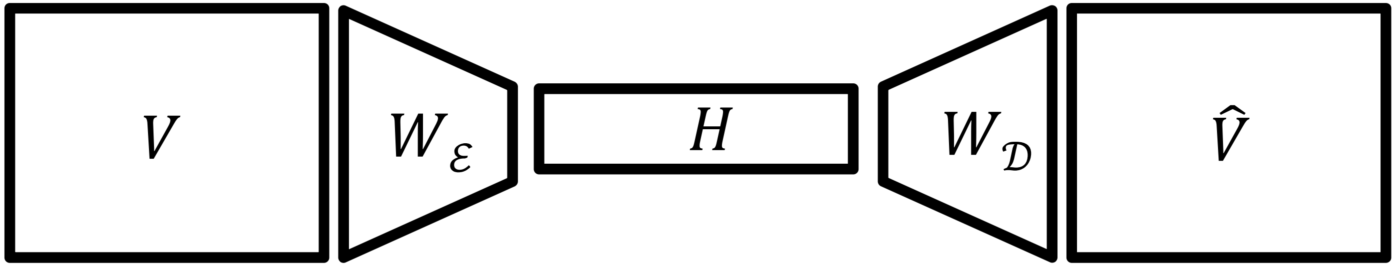
Nonnegative Autoencoder (NAE)



1. Layer: $H = W_{\mathcal{E}}\, V$
2. Layer: $\hat{V} = W_{\mathcal{D}}\, H$

- How can one adjust the AE to simulate NMF?
- How can one achieve nonnegativity?
- How can one incorporate musical knowledge?
- …

Nonnegative Autoencoder (NAE)

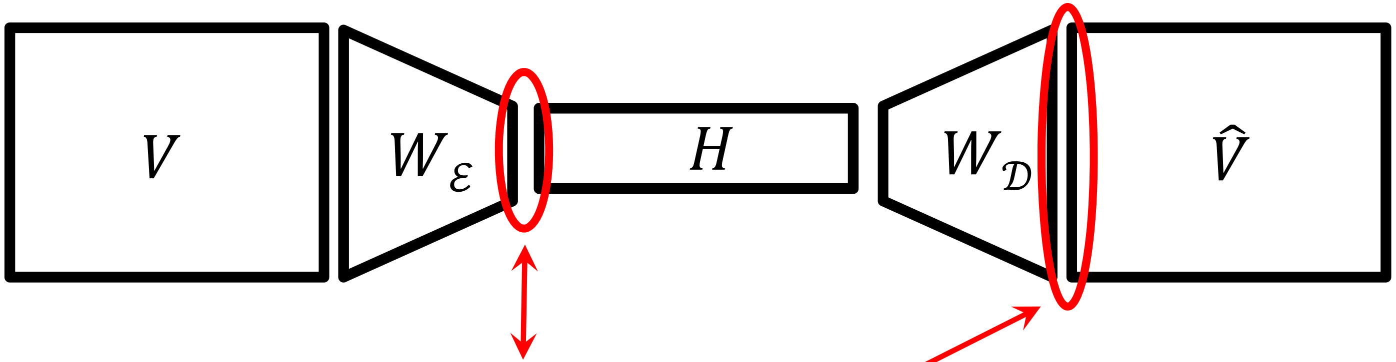


1. Layer: $H = W_{\mathcal{E}}\, V$
2. Layer: $\hat{V} = W_{\mathcal{D}}\, H$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

- Loss function: same as in NMF

Nonnegative Autoencoder (NAE)

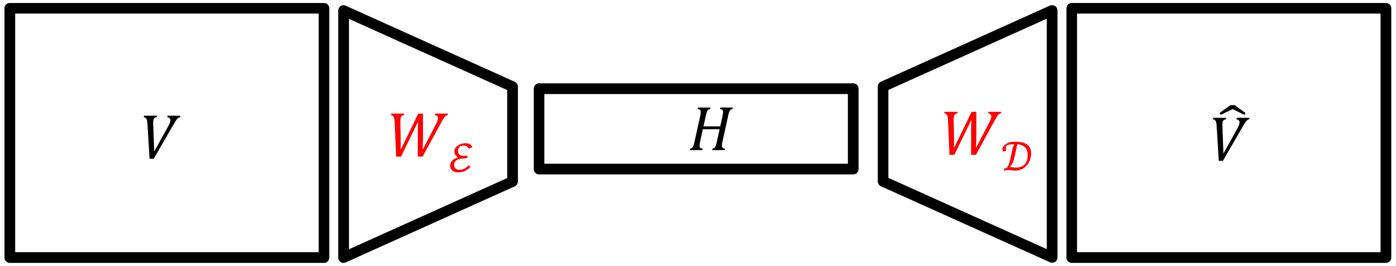


1. Layer: $H = \max(W_{\mathcal{E}}\, V, 0)$
2. Layer: $\hat{V} = \max(W_{\mathcal{D}}\, H, 0)$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

- Loss function: same as in NMF
- Activation function (ReLU) makes H and $\hat{V}$ nonnegative

Nonnegative Autoencoder (NAE)



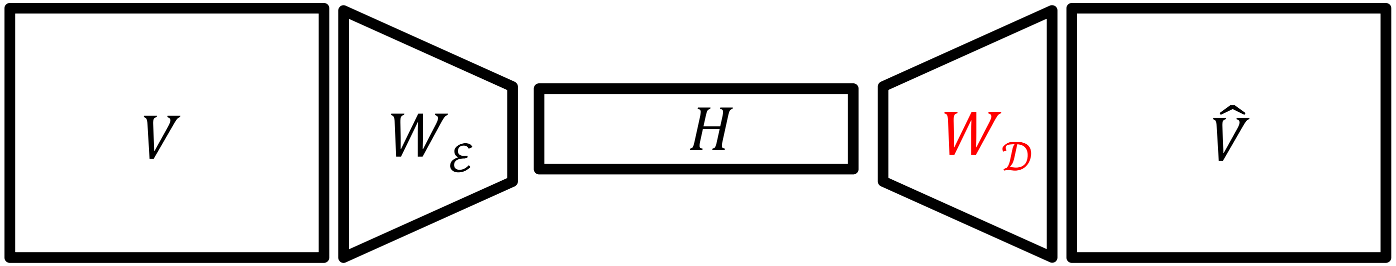
1. Layer: $H = \max(W_{\mathcal{E}}\, V, 0)$
2. Layer: $\hat{V} = \max(W_{\mathcal{D}}\, H, 0)$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

$$W_{\mathcal{D}} \leftarrow \max\left(W_{\mathcal{D}} - \gamma \frac{\partial \mathcal{L}}{\partial W_{\mathcal{D}}}, 0\right)$$

- Loss function: same as in NMF
- Activation function (ReLU) makes H and $\hat{V}$ nonnegative
- Projected gradient descent can be used to keep $W_{\mathcal{D}}$ (and $W_{\mathcal{E}}$) nonnegative

Musical Constraints

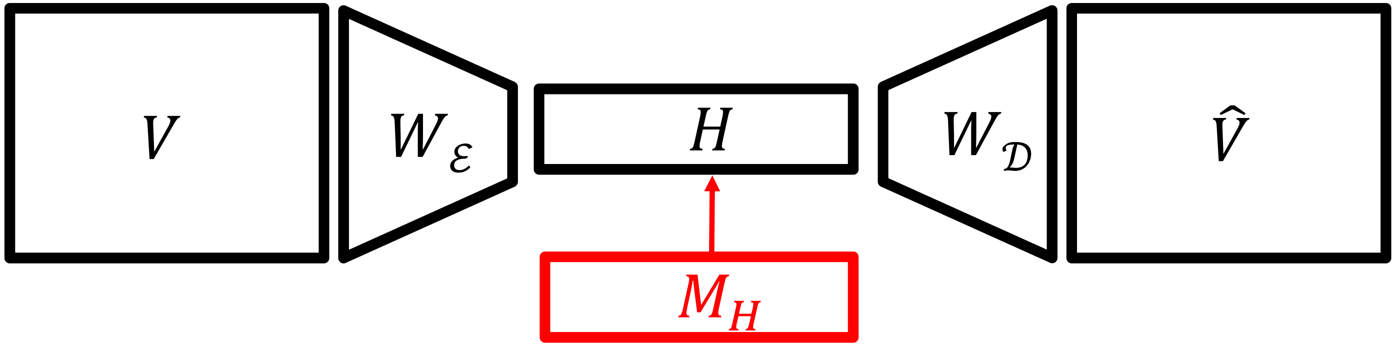


$$H = \max(W_{\mathcal{E}}\, V, 0)$$
$$\hat{V} = \max(W_{\mathcal{D}}\, H, 0)$$

- Template constraints: Project certain entries in $W_{\mathcal{D}}$ to zero values (using projected gradient decent)

Musical Constraints

Ewert, Sandler: Structured Dropout for Weak Label and Multi-Instance Learning and Its Application to Score-Informed Source Separation. Proc. ICASSP, 2017.



$$H' = H \odot M_H$$
$$\hat{V} = \max(W_{\mathcal{D}} H', 0)$$

- Template constraints: Project certain entries in $W_{\mathcal{D}}$ to zero values (using projected gradient decent)
- Activation constraints: Use structured dropout by applying pointwise multiplication with binary mask M_H

NAE with Multiplicative Update Rules

- Multiplicative update rules in NMF:
  - Preserve nonnegativity
  - Lead to fast convergence

- Question: Can one introduce multiplicative update rules to train network weights for NAE?

- Use in additive gradient descent

$$W^{(\ell+1)} = W^{(\ell)} - \gamma \cdot \frac{\partial \mathcal{L}}{\partial W}$$

a suitable (adaptive) learning rate γ .

NAE with Multiplicative Update Rules

- Encoder:

$$H = W_{\mathcal{E}} V$$

- Structured Dropout:

$$H' = H \odot M_H$$

- Decoder:

$$\hat{V} = W_{\mathcal{D}} H'$$

Zunner: Neural Networks with Nonnegativity Constraints for Decomposing Music Recordings. Master Thesis, FAU, 2021.

NAE with Multiplicative Update Rules

- Encoder:

$$H = W_{\mathcal{E}} V$$

$$W_{\mathcal{E},rk}^{(\ell+1)} = W_{\mathcal{E},rk}^{(\ell)} \cdot \frac{\left(\left((W_{\mathcal{D}}^{\top} V) \odot M_H\right) V^{\top}\right)_{rk}}{\left(\left((W_{\mathcal{D}}^{\top} W_{\mathcal{D}} H'^{(\ell)}) \odot M_H\right) V^{\top}\right)_{rk}}$$

- Structured Dropout:

$$H' = H \odot M_H$$

- Decoder:

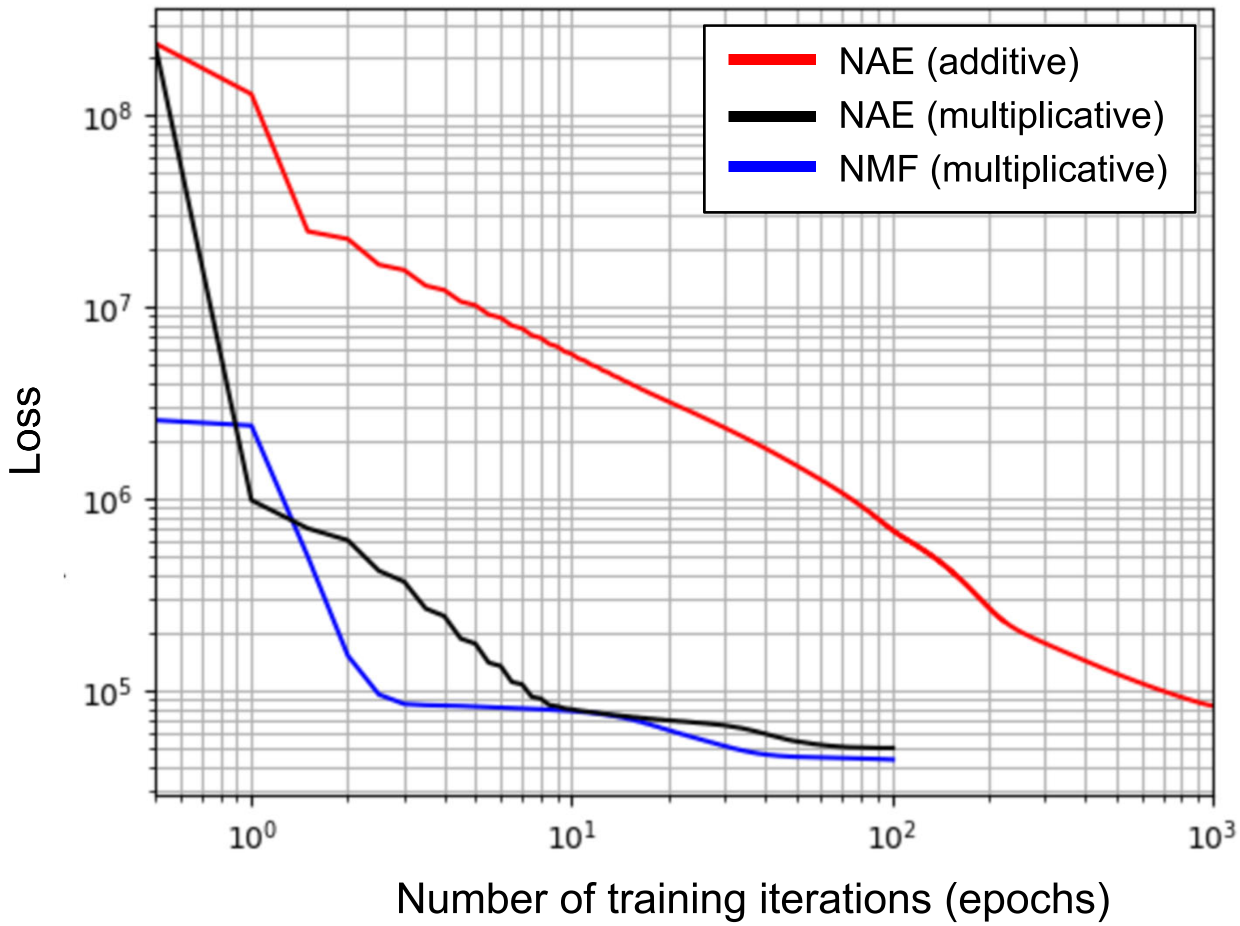
$$\hat{V} = W_{\mathcal{D}} H'$$

$$W_{\mathcal{D},kr}^{(\ell+1)} = W_{\mathcal{D},kr}^{(\ell)} \cdot \frac{\left(V H'^{\top}\right)_{kr}}{\left(W_{\mathcal{D}}^{(\ell)} H' H'^{\top}\right)_{kr}}$$

Similar idea and computation as for NMF.

Zunner: Neural Networks with Nonnegativity Constraints for Decomposing Music Recordings. Master Thesis, FAU, 2021.

Approximation Loss



Zunner: Neural Networks with Nonnegativity Constraints for Decomposing Music Recordings. Master Thesis, FAU, 2021.

Conclusions (NAE)

- Simulation of NMF:
  - Decoder corresponds to NMF templates
  - Encoder learns a kind of pseudo-inverse
  - Code corresponds to NMF activations

- Nonnegativity can be achieved via
  - activation function (ReLU)
  - projected gradient descent
  - multiplicative update rules

- Musical knowledge can be integrated via
  - removing network weights (template constraints)
  - structured dropout (activation constraints)

Outlook

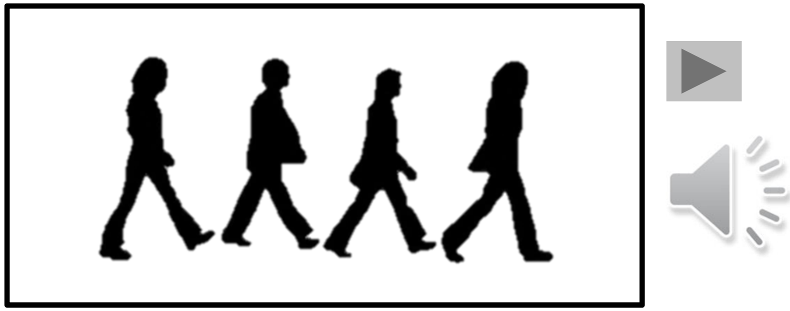
- More complex networks
  - Deeper networks (more layers)
  - Different layer types (CNN, RNN, …) and activation functions
  - Modification of loss function and regularization terms

- Understanding encoder – decoder relationship
  - Nonnegativity
  - Pseudo-inverse

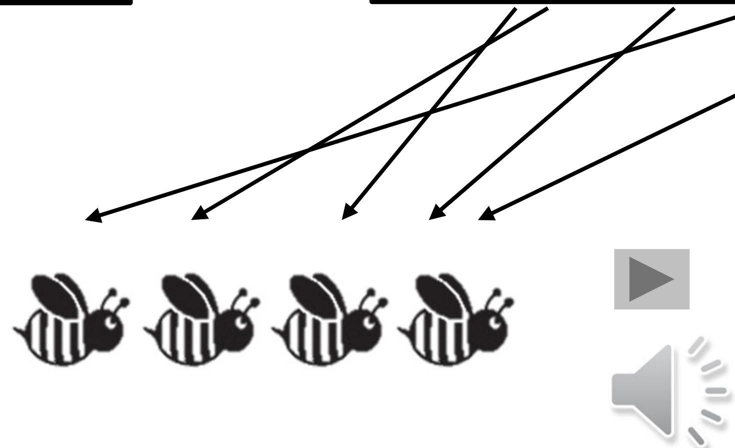
- Update rules
  - Constraints and conversion issues
  - Adaptive learning rates and projected gradient descent

Audio Mosaicing (Style Transfer)

Target signal: Beatles–Let it be



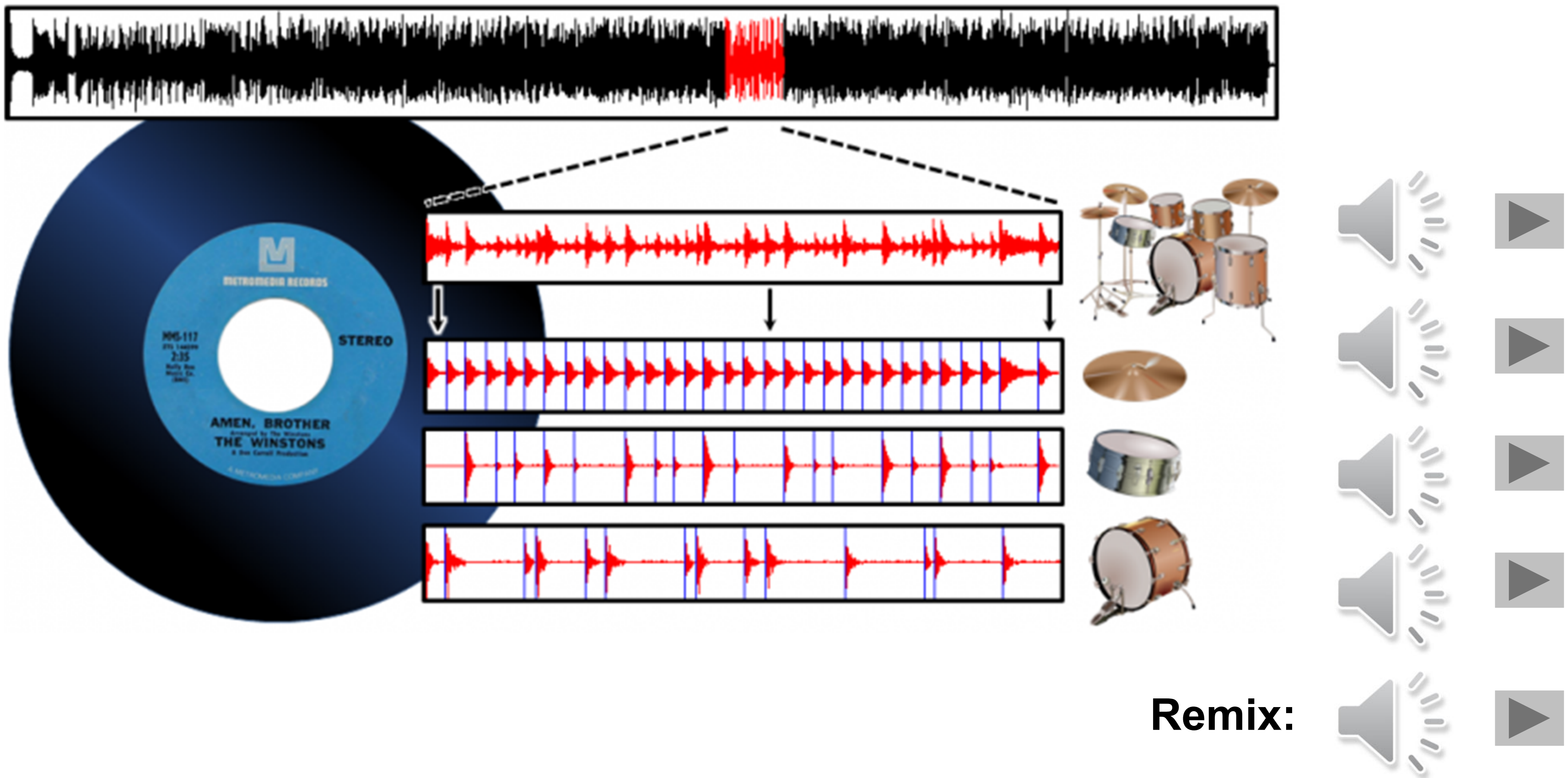
Source signal: Bees



Mosaic signal: Let it Bee

Driedger, Prätzlich, Müller: Let It Bee – Towards NMF-Inspired Audio Mosaicing, ISMIR 2015..

Informed Drum-Sound Decomposition



Dittmar, Müller: Reverse Engineering the Amen Break – Score-Informed Separation and Restoration Applied to Drum Recordings, IEEE/ACM TASLP, 2016.

Suárez: DNN-Based Matrix Factorization with Applications to Drum Sound Decomposition. Master Thesis, FAU, 2020.

Reconstruction of Sound Events

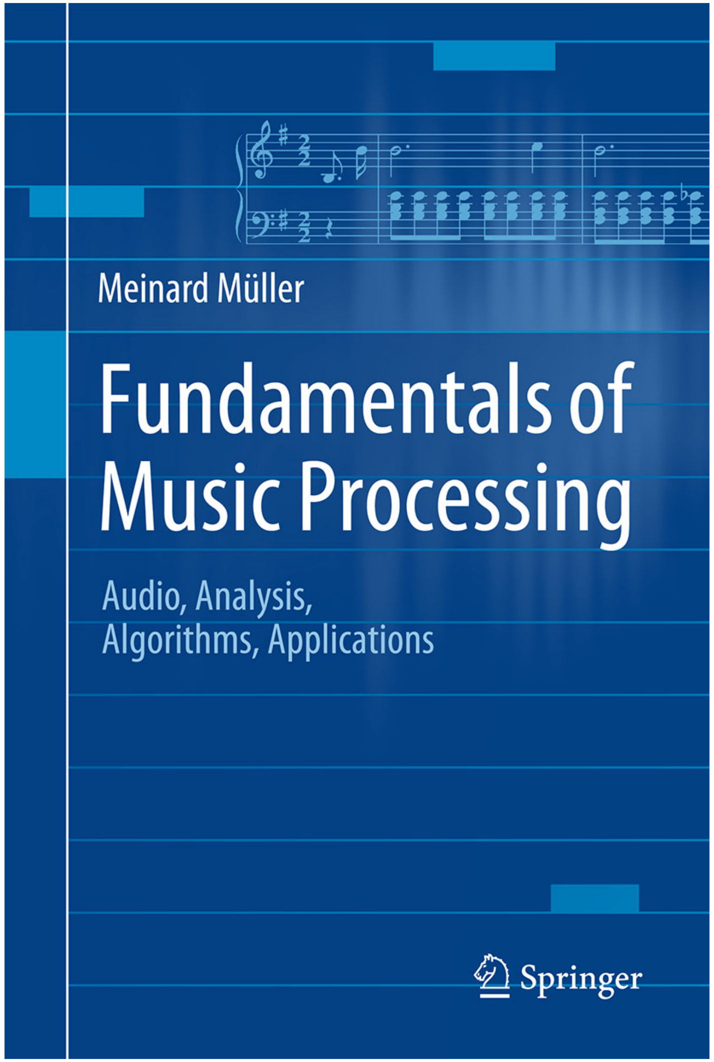
- Reconstruction via spectral masking (Wiener filtering)
- Alternative: Resynthesis approach
- Differentiable Digital Signal Processing (DDSP) combines classical DSP and deep learning
- Generative adversarial networks may help to reduce the artifacts

Lecture 8: Recurrent and Generative Adversarial Network Architectures for Text-to-Speech

Selected Topics in Deep Learning for Audio, Speech, and Music Processing

1. Introduction to Audio and Speech Processing
2. Introduction to Music Processing
3. Permutation Invariant Training Techniques for Speech Separation
4. Deep Clustering for Single-Channel Ego-Noise Suppression
5. Music Source Separation
6. Nonnegative Autoencoders with Applications to Music Audio Decomposing
7. Attention in Sound Source Localization and Speaker Extraction
8. Recurrent and Generative Adversarial Network Architectures for Text-to-Speech
9. Connectionist Temporal Classification (CTC) Loss with Applications to Theme-Based Music Retrieval
10. From Theory to Practise

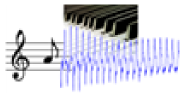
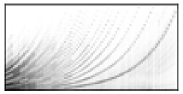
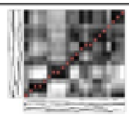
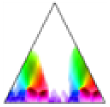
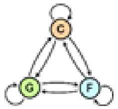
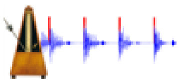
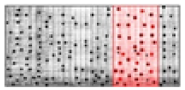
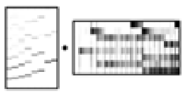
Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Book: Fundamentals of Music Processing

Chapter		Music Processing Scenario
1		Music Represenations
2		Fourier Analysis of Signals
3		Music Synchronization
4		Music Structure Analysis
5		Chord Recognition
6		Tempo and Beat Tracking
7		Content-Based Audio Retrieval
8		Musically Informed Audio Decomposition

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Software & Audio: FMP Notebooks

FMP Notebooks

Python Notebooks for Fundamentals of Music Processing

The FMP notebooks offer a collection of educational material closely following the textbook Fundamentals of Music Processing (FMP). This is the starting website, which is opened when calling https://www.audiolabs-erlangen.de/FMP. Besides giving an overview, this website provides information on the license, the main contributors, and some links.

https://www.audiolabs-erlangen.de/FMP