INTERNATIONAL AUDIO LABORATORIES ERLANGEN
A joint institution of Fraunhofer IIS and Universität Erlangen-Nürnberg

AUDIO LABS

Loss Functions Matter

Three Case Studies in Informed Loss Design

Meinard Müller

International Audio Laboratories Erlangen

meinard.mueller@audiolabs-erlangen.de

Lecture Series "Musical Informatics"

Linz, November 19, 2025

Friedrich-Alexander-Universität Erlangen-Nürnberg

Fraunhofer IIS

Meinard Müller

- Mathematics (Diplom/Master, 1997)
  Computer Science (PhD, 2001)
  Information Retrieval (Habilitation, 2007)

- Senior Researcher (2007-2012)

- Professor Semantic Audio Processing (since 2012)

- Former President of the International Society for
  Music Information Retrieval (MIR)

- IEEE Fellow for contributions to
  Music Signal Processing

International Audio Laboratories Erlangen

- Fraunhofer Institute for Integrated Circuits IIS
- Largest Fraunhofer institute with > 1000 members
- Applied research for sensor, audio, and media technology

- Friedrich-Alexander Universität Erlangen-Nürnberg (FAU)
- One of Germany’s largest universities with ≈ 40,000 students
- Strong Technical Faculty

International Audio Laboratories Erlangen

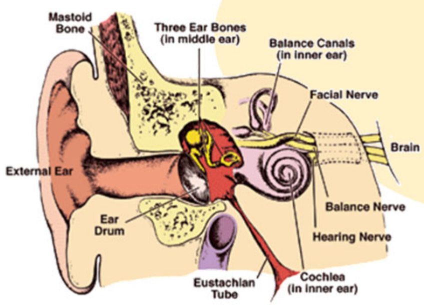
Audio

International Audio Laboratories Erlangen

Audio Coding

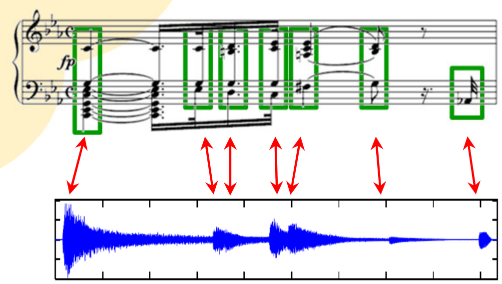
3D Audio

Audio



Psychoacoustics

Internet of Things



Music Processing

Meinard Müller: Research Group

- Ben Maman
- Simon Schwär
- Johannes Zeitler
- Peter Meier

- Sebastian Strahl
- Uli Berendes
- Vlora Arifi-Müller
- Stefan Balke

- Ching-Yu Chiu (Sunny)
- Yigitcan Özer
- Michael Krause
- Christof Weiß
- Sebastian Rosenzweig
- Frank Zalkow

- Hendrik Schreiber
- Christian Dittmar
- Stefan Balke
- Jonathan Driedger
- Thomas Prätzlich
- …

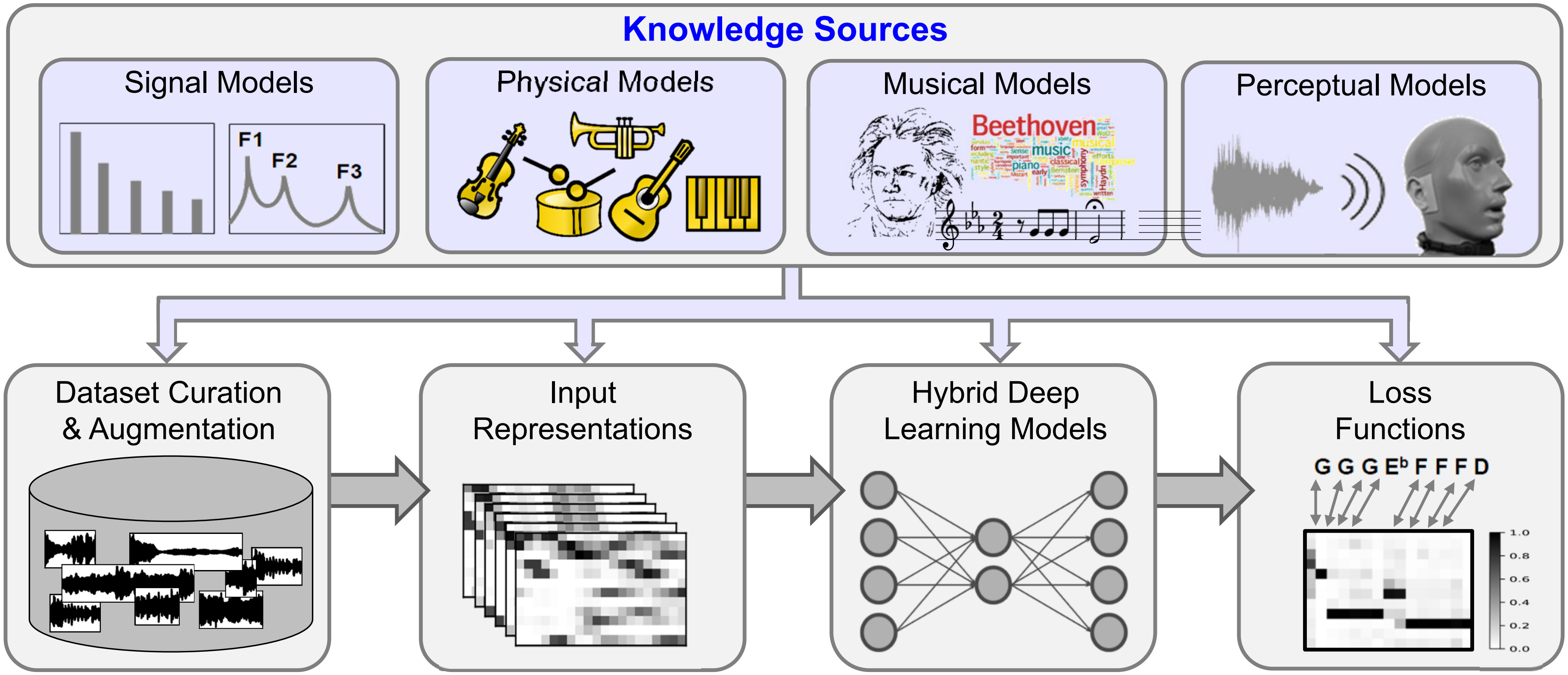
Meinard Müller: Research Group

- Ben Maman
- Simon Schwär
- Johannes Zeitler
- Peter Meier

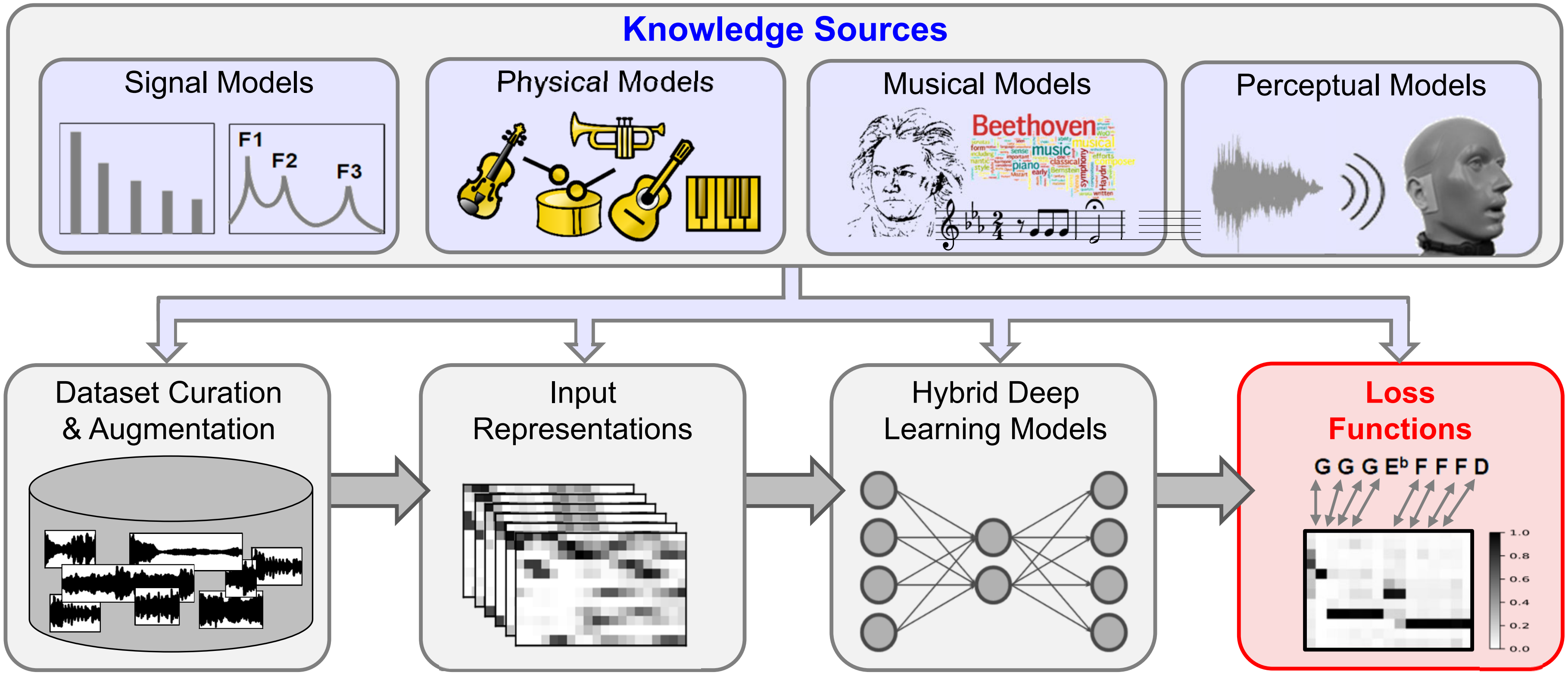
- Sebastian Strahl
- Uli Berendes
- Vlora Arifi-Müller
- Stefan Balke

- Ching-Yu Chiu (Sunny)
- Yigitcan Özer
- Michael Krause
- Christof Weiß
- Sebastian Rosenzweig
- Frank Zalkow

- Hendrik Schreiber
- Christian Dittmar
- Stefan Balke
- Jonathan Driedger
- Thomas Prätzlich
- …



Richard, Lostanlen, Yang, Müller: Model-Based Deep Learning for Music Information Research: Leveraging Diverse Knowledge Sources to Enhance Explainability, Controllability, and Resource Efficiency. IEEE Signal Processing Magazine, 41(6): 51–59, 2024



Richard, Lostanlen, Yang, Müller: Model-Based Deep Learning for Music Information Research: Leveraging Diverse Knowledge Sources to Enhance Explainability, Controllability, and Resource Efficiency. IEEE Signal Processing Magazine, 41(6): 51–59, 2024.

Overview

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations

- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies

- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

Johannes Zeitler

Overview

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations
- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies
- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

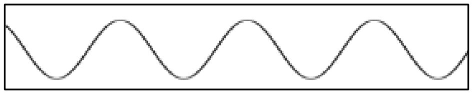
Johannes Zeitler

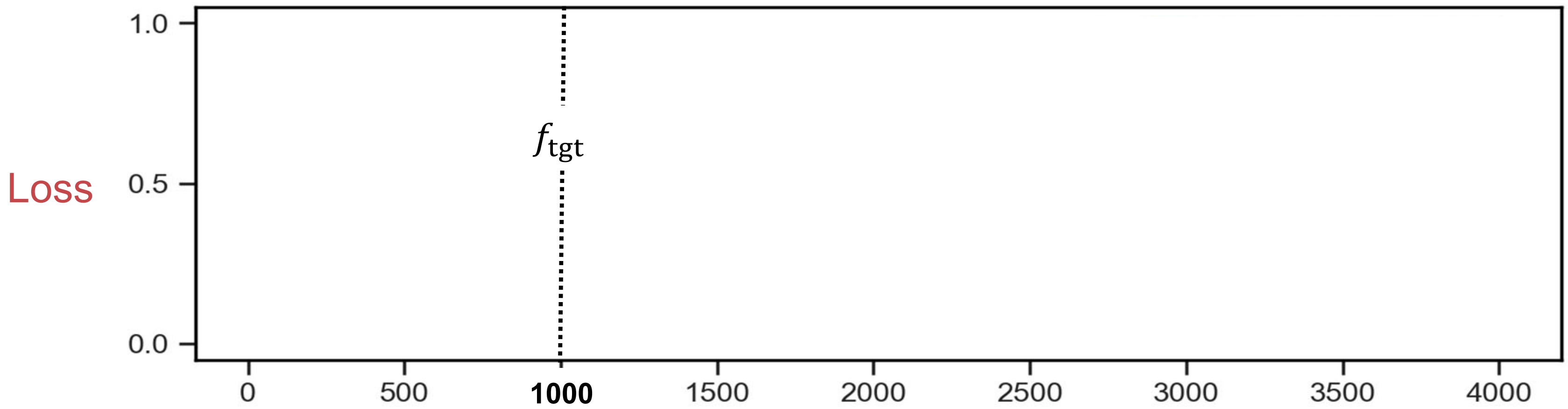
Literature

- Turian, Henry: I'm sorry for your loss: Spectrally-based audio distances are bad at pitch. Proc. Adv. Neural Inf. Process. Syst., 2020.
- Hayes, Saitis, Fazekas: Sinusoidal frequency estimation by gradient descent. Proc. ICASSP, 2023.
- Torres, Peeters, Richard: Unsupervised Harmonic Parameter Estimation Using DDSP and Spectral Optimal Transport. Proc. ICASSP, 2024
- **Schwär**, Müller: Multi-Scale Spectral Loss Revisited. IEEE Signal Processing Letters, 30: 1712–1716, 2023.

Example Scenario: Sinusoidal Frequency Estimation

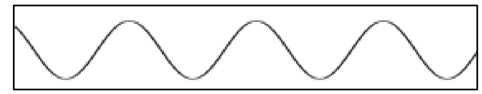
Sinusoid with target frequency: $f_{\text{tgt}} = 1000$ Hz



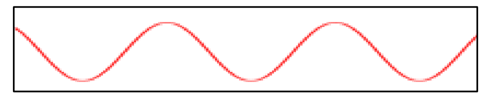


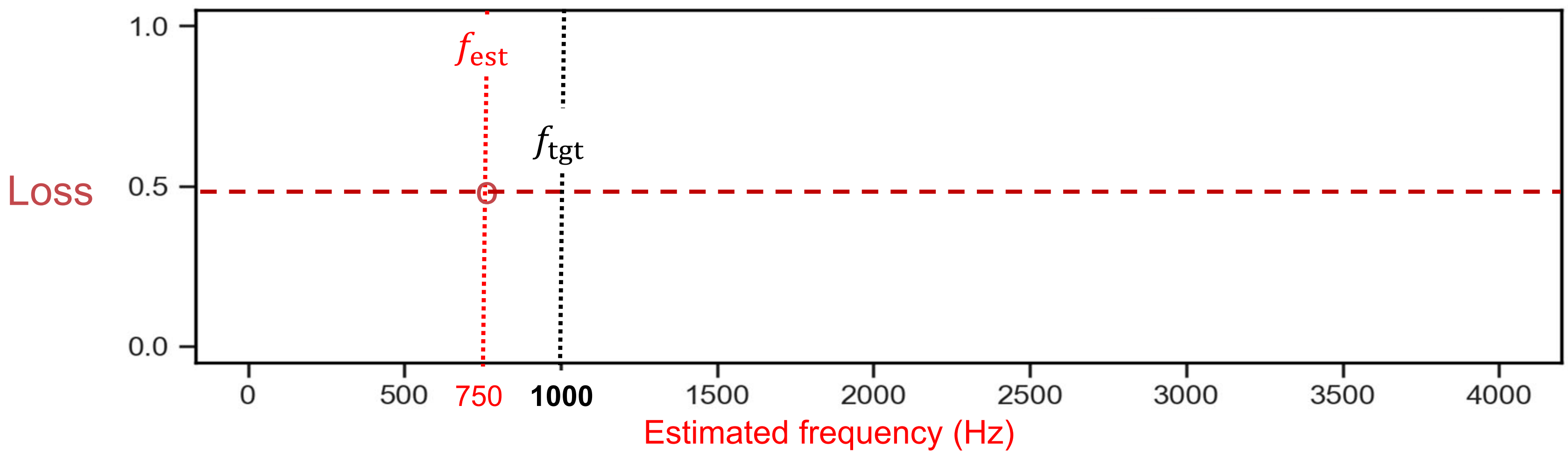
Example Scenario: Sinusoidal Frequency Estimation

Sinusoid with target frequency: $f_{\mathrm{tgt}} = 1000$ Hz



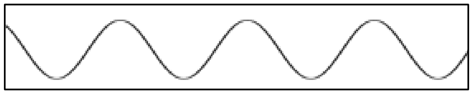
Sinusoid with estimated frequency: $f_{\mathrm{est}} = 750$ Hz



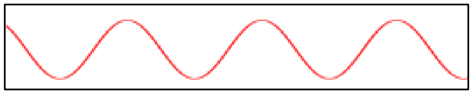


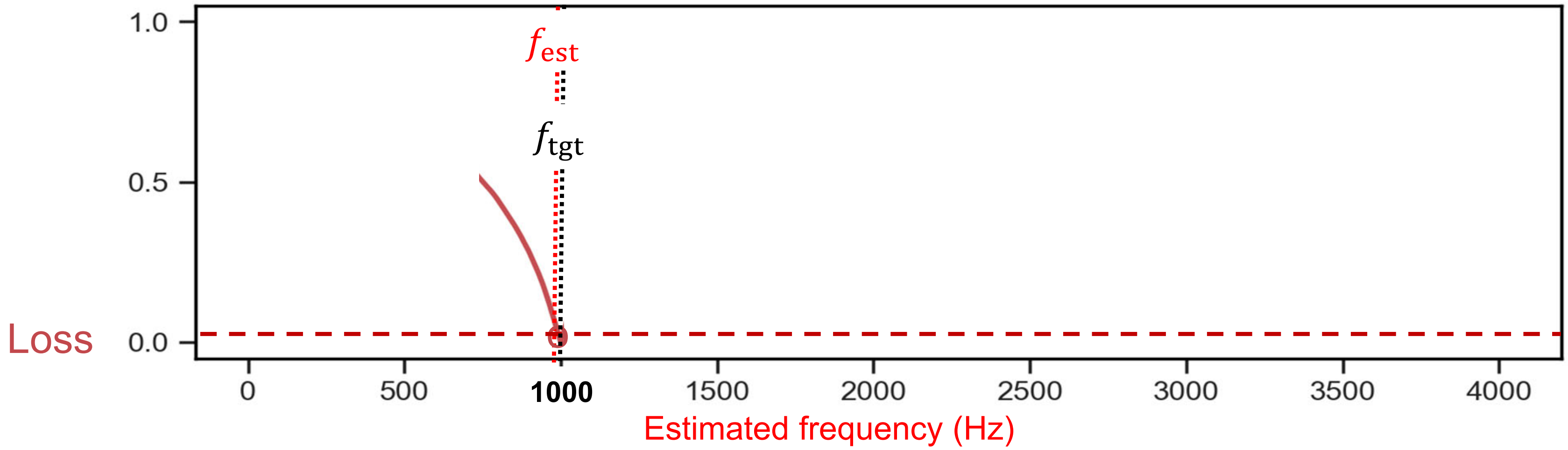
Example Scenario: Sinusoidal Frequency Estimation

Sinusoid with target frequency: $f_{\mathrm{tgt}} = 1000$ Hz



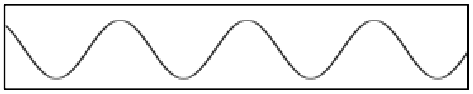
Sinusoid with estimated frequency: $f_{\mathrm{est}} = 972$ Hz



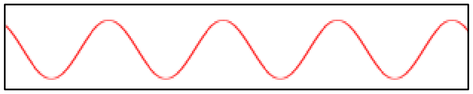


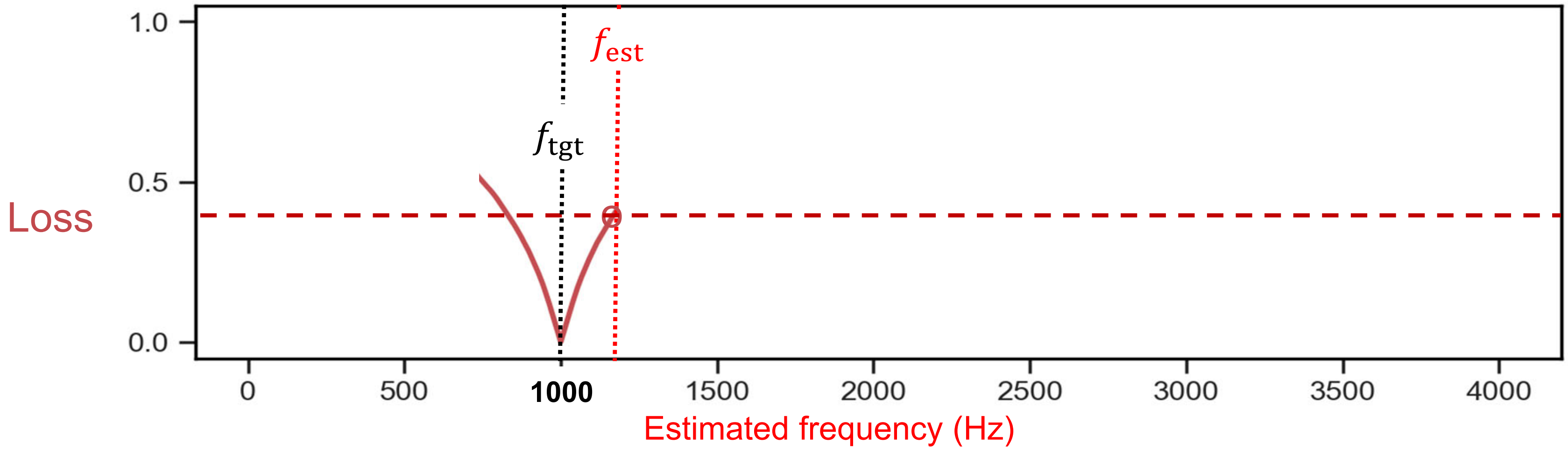
Example Scenario: Sinusoidal Frequency Estimation

Sinusoid with target frequency: $f_{\mathrm{tgt}} = 1000$ Hz



Sinusoid with estimated frequency: $f_{\mathrm{est}} = 1100$ Hz

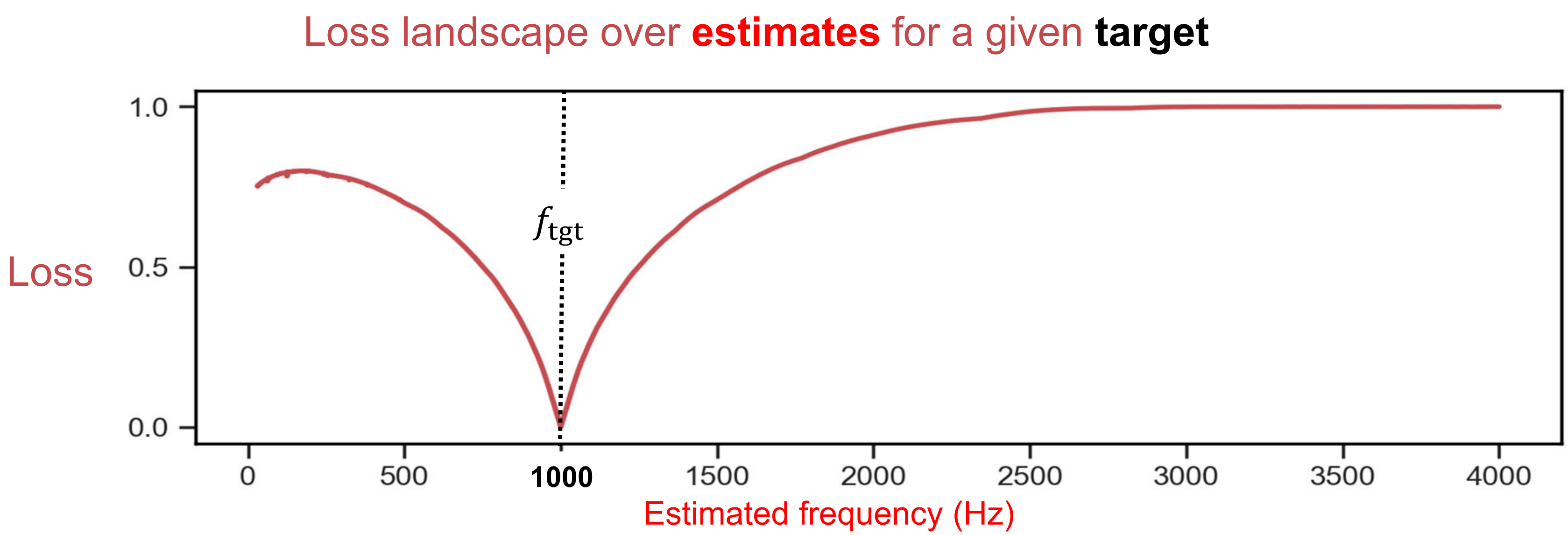




Example Scenario: Sinusoidal Frequency Estimation

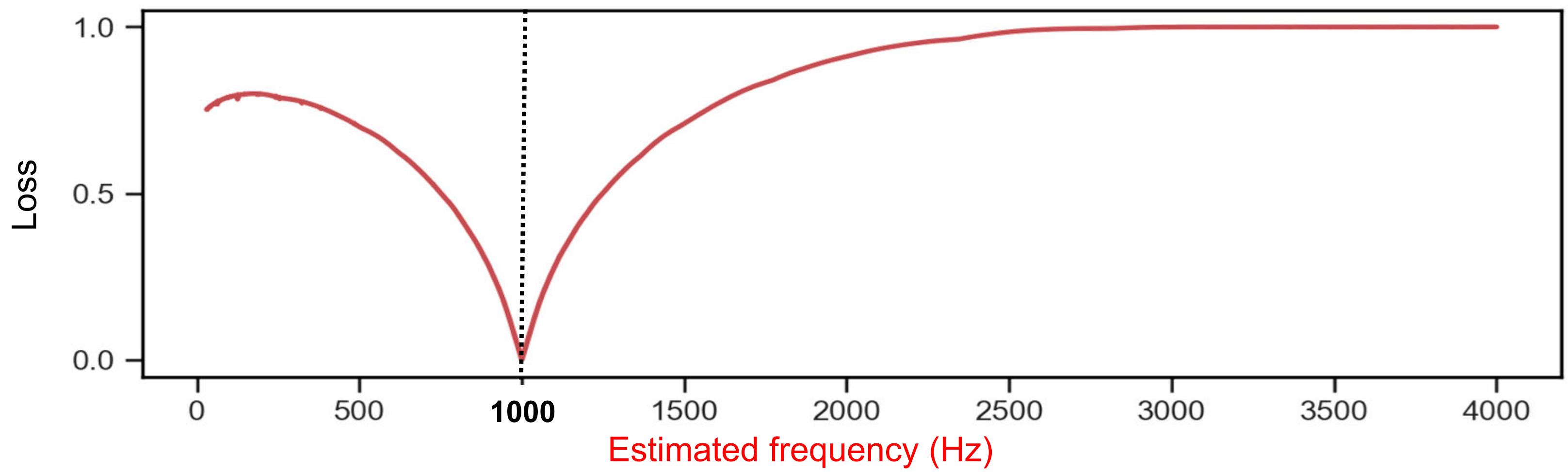
Sinusoid with target frequency: $f_{\mathrm{tgt}} = 1000$ Hz

Sinusoidal sweep of estimated frequencies f_{est}



Example Scenario: Sinusoidal Frequency Estimation

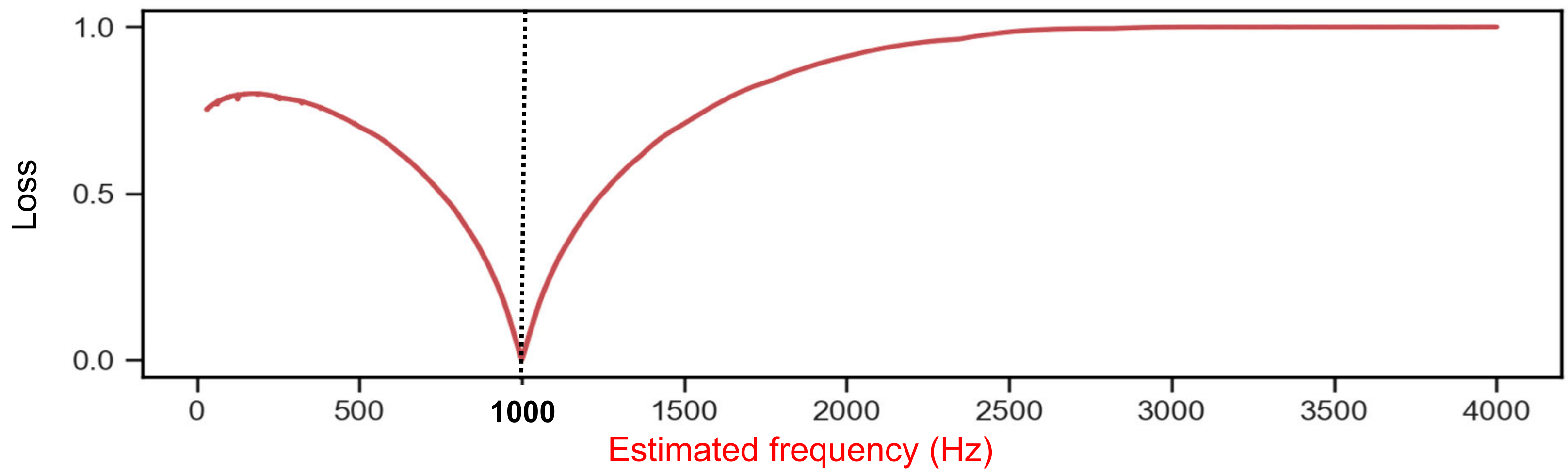
Loss landscape depends a lot on the chosen loss function to compare **estimated** and **target** signal



Example Scenario: Sinusoidal Frequency Estimation

Loss landscape depends a lot on the chosen loss function to compare **estimated** and **target** signal

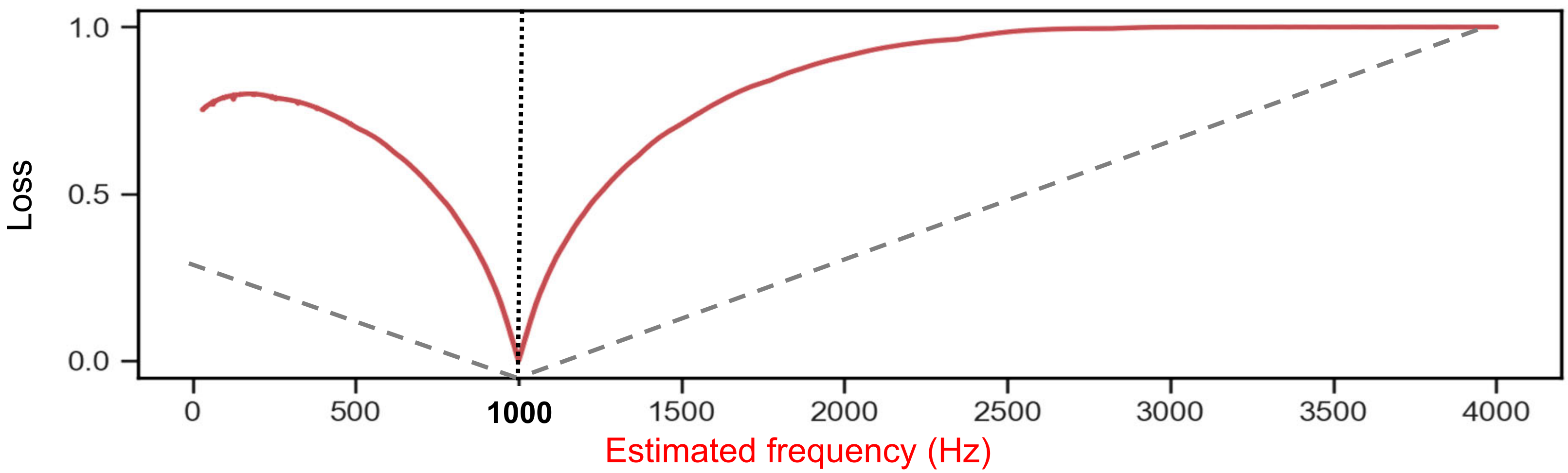
- Loss function discussed later



Example Scenario: Sinusoidal Frequency Estimation

Loss landscape depends a lot on the chosen loss function to compare **estimated** and **target** signal

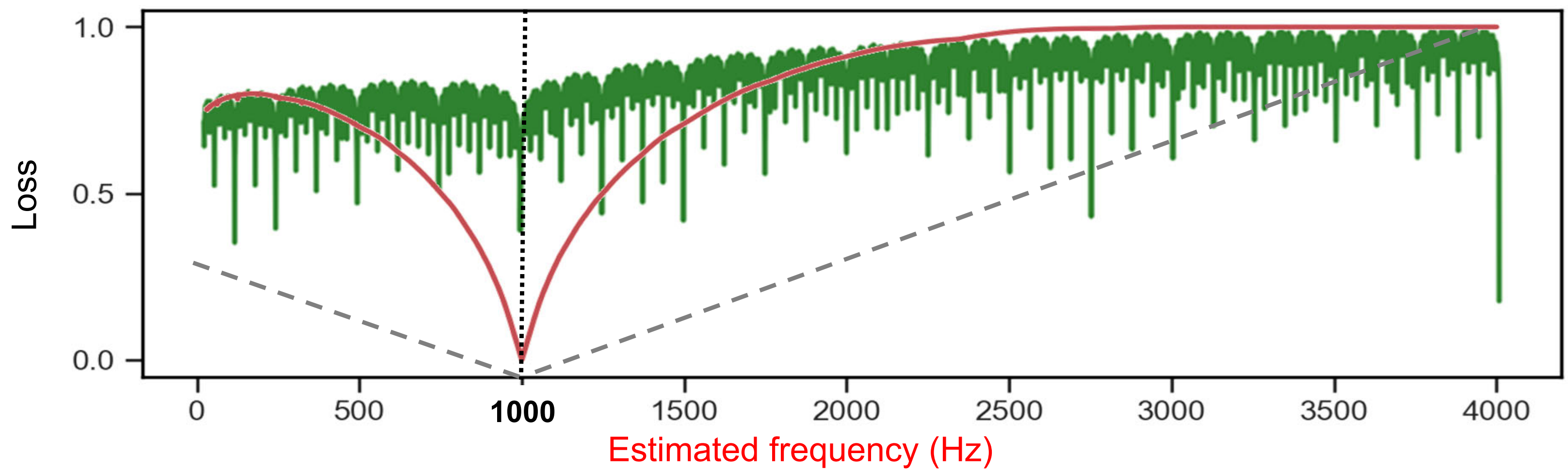
- Loss function discussed later
- Ideal convex loss



Example Scenario: Sinusoidal Frequency Estimation

Loss landscape depends a lot on the chosen loss function to compare **estimated** and **target** signal

- Loss function discussed later
- Ideal convex loss
- Multi-Scale Spectral (MSS) loss with standard settings

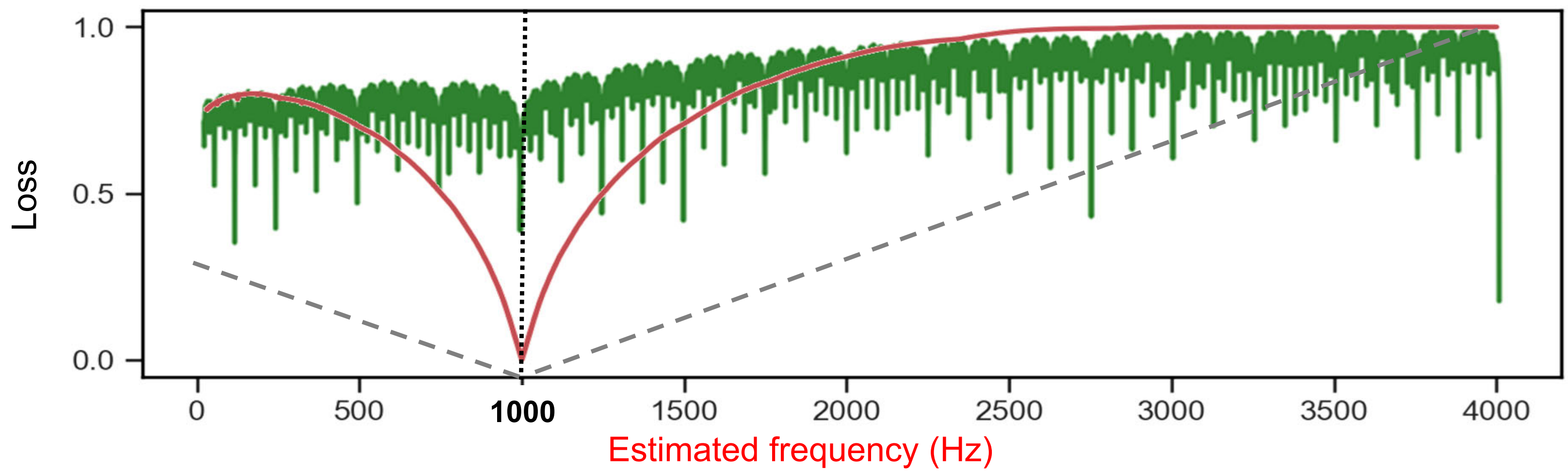


Example Scenario: Sinusoidal Frequency Estimation

Loss landscape depends a lot on the chosen loss function to compare **estimated** and **target** signal

- Loss function discussed later
- Ideal convex loss
- Multi-Scale Spectral (MSS) loss with standard settings

The MSS loss is what we widely use in audio processing (e.g., DDSP)



Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_2^2$

Spectrum

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n + mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x, \hat{x}) := \sum_{N\in\mathcal{N}} \sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_2^2$

Spectrum
$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n + mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss
$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_2^2$

Spectrum

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n + mH]\, w[n] \exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x, \hat{x}) := \sum_{N \in \mathcal{N}} \sum_{p \in \mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\}, \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\}, \gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\}, \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x, \ \log(x+\varepsilon)\}, \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_2^2$

Spectrum

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n+mH]\,w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \Vert\mathcal{Y} - \hat{\mathcal{Y}}\Vert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \Vert\mathcal{Y} - \hat{\mathcal{Y}}\Vert_2^2$

Spectrum
$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n + mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss
$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}, \gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_2^2$

Spectrum

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n+mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_2^2$

Spectrum
$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n+mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss
$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\},\ \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\},\ \gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\},\ \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\},\ \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_2^2$

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

Spectrum

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n + mH]w[n] \exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x, \hat{x}) := \sum_{N \in \mathcal{N}} \sum_{p \in \mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Multi-Scale Spectral Loss

- x input signal
- N window size
- H hop size
- w window function
- p compression function
- d distance function
- $\mathcal{N}$ set of window sizes
- $\mathcal{P}$ set of compression function

MSS loss with standard settings: (WH, S4, C4, D1)

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\}, \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\}, \gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\}, \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x, \ \log(x+\varepsilon)\}, \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert\mathcal{Y} - \hat{\mathcal{Y}}\rVert_2^2$

Spectrum

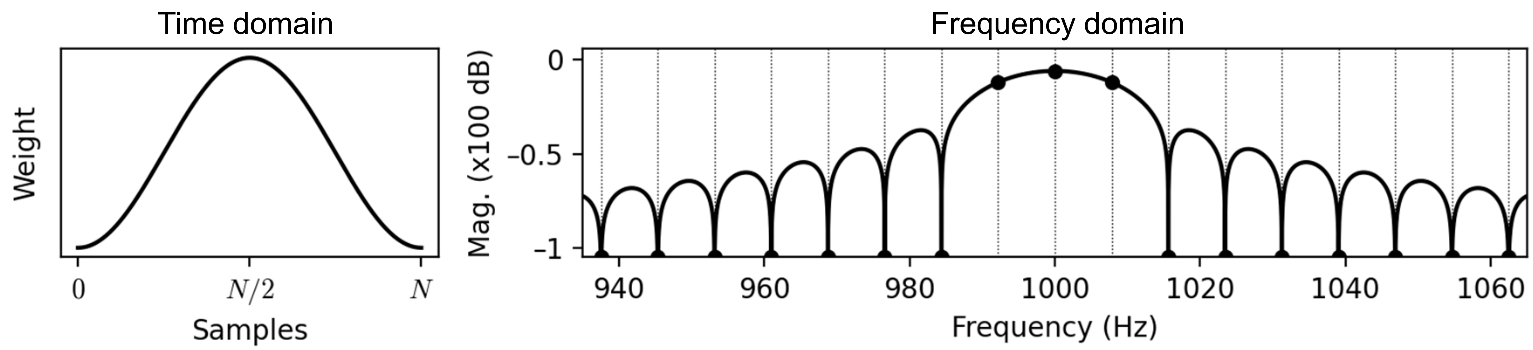
$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n+mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right)$$

MSS loss

$$\mathcal{L}_{\mathrm{MSS}}(x,\hat{x}) := \sum_{N\in\mathcal{N}}\sum_{p\in\mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p})$$

Spectrum-Based Distance

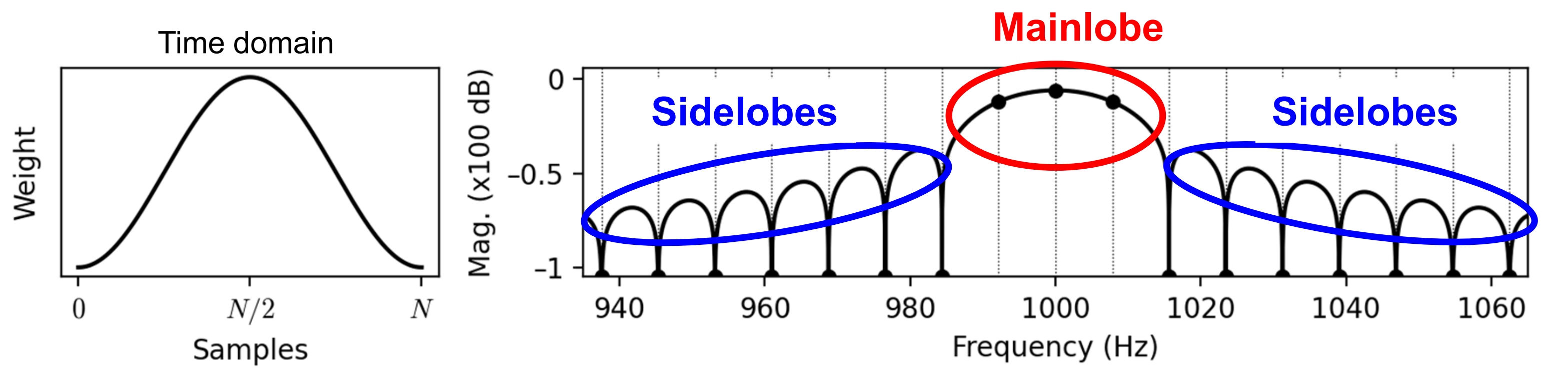
Hann window



- Input signal: Sinusoid with frequency $f = 1000$ Hz

Spectrum-Based Distance

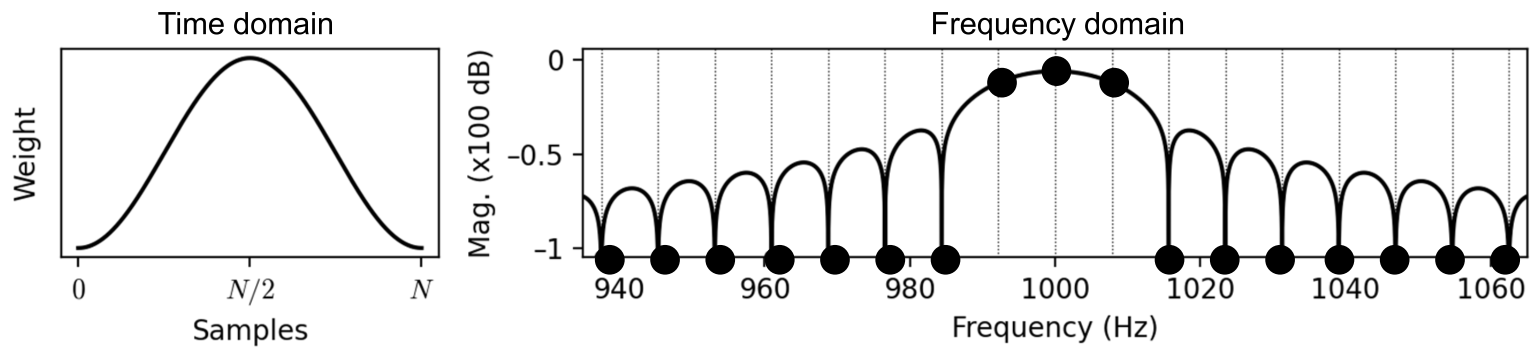
Hann window



- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing

Spectrum-Based Distance

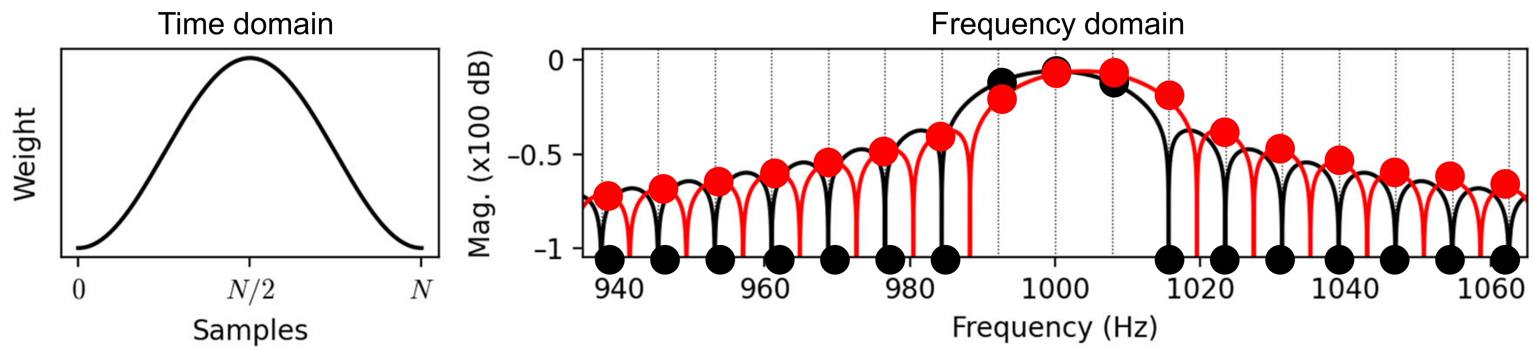
Hann window



- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing
- Discrete STFT → **Frequency grid**

Spectrum-Based Distance

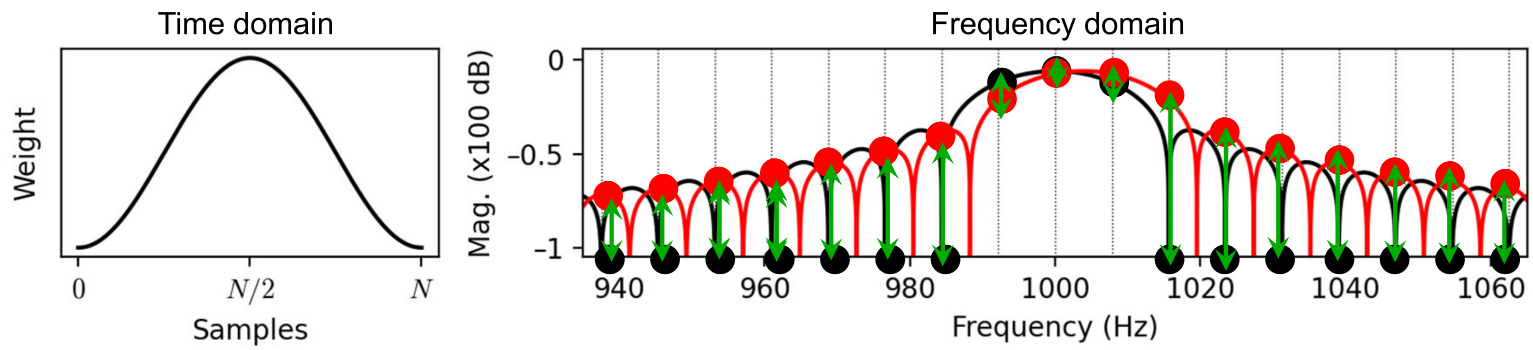
Hann window



- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT $\rightarrow$ Spectral leakage due to windowing
- Discrete STFT $\rightarrow$ **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1003.9$ Hz

Spectrum-Based Distance

Hann window

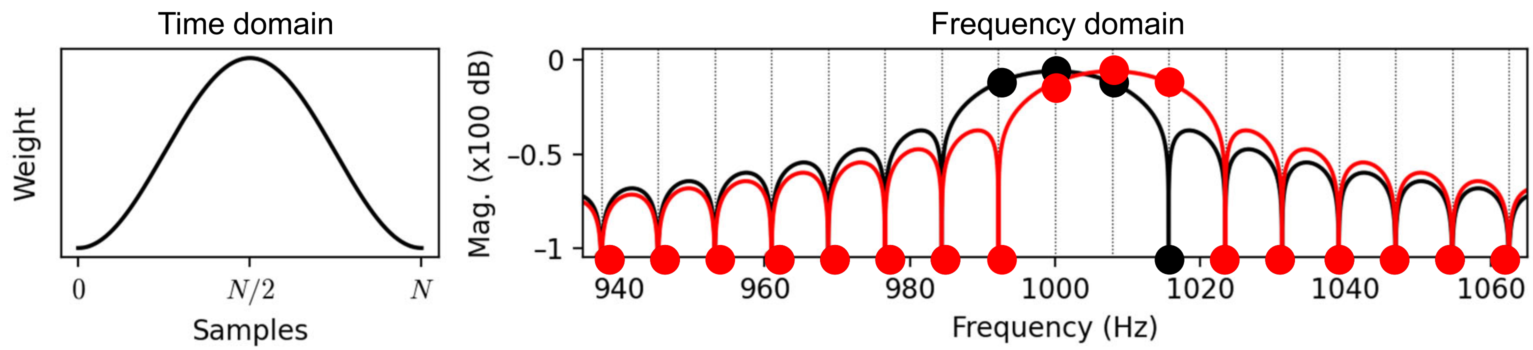


- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT $\rightarrow$ Spectral leakage due to windowing
- Discrete STFT $\rightarrow$ **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1003.9$ Hz

Distance depends on
- Grid sampling
- Mainlobe & sidelobes
- Window type
- STFT parameters

Spectrum-Based Distance

Hann window

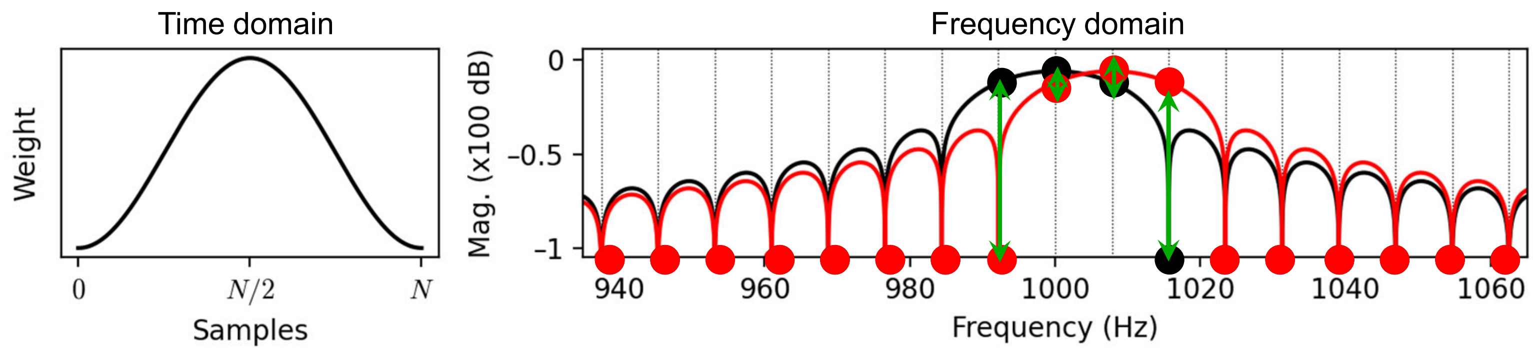


- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing
- Discrete STFT → **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1007.8$ Hz

Distance depends on
- Grid sampling
- Mainlobe & sidelobes
- Window type
- STFT parameters

Spectrum-Based Distance

Hann window

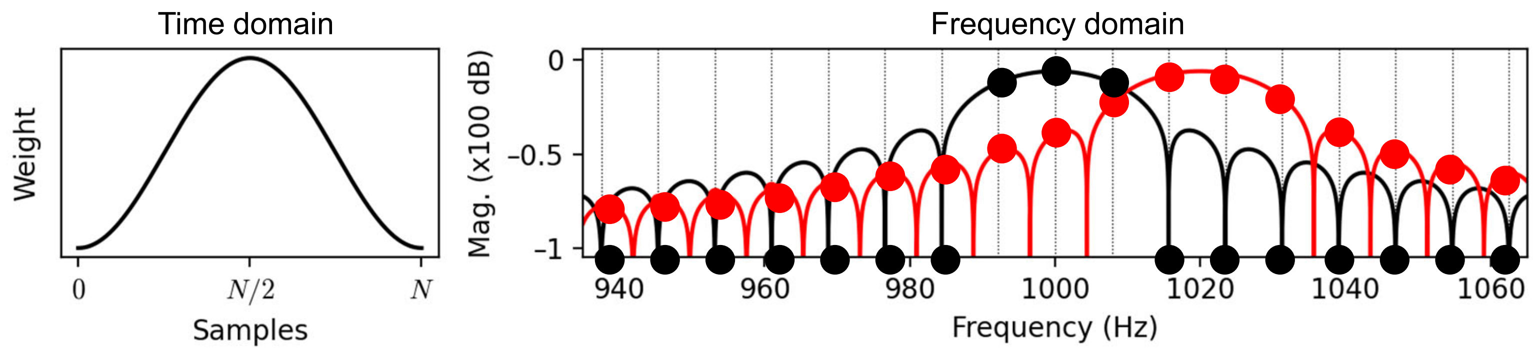


- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing
- Discrete STFT → **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1007.8$ Hz

Distance depends on
- Grid sampling
- Mainlobe & sidelobes
- Window type
- STFT parameters

Spectrum-Based Distance

Hann window

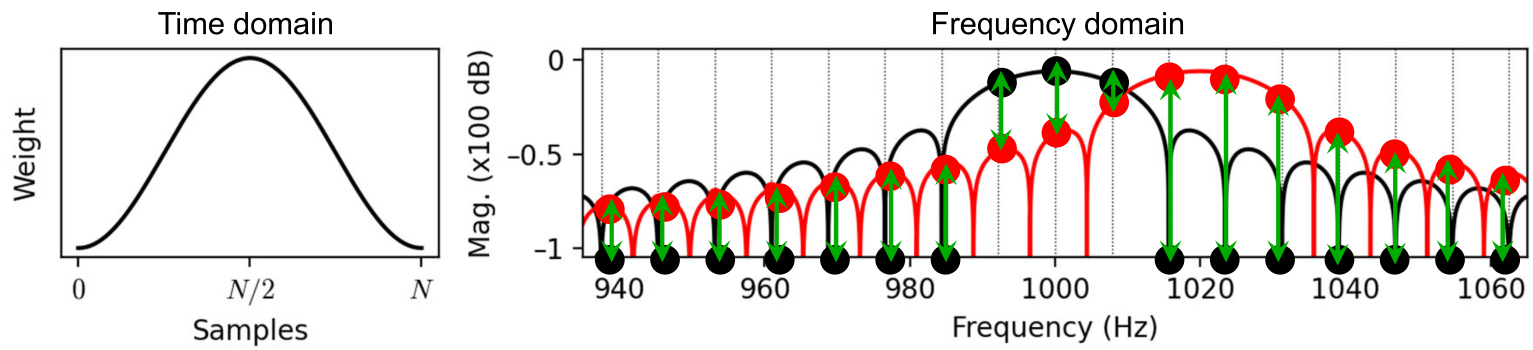


- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing
- Discrete STFT → **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1020$ Hz

Distance depends on
- Grid sampling
- Mainlobe & sidelobes
- Window type
- STFT parameters

Spectrum-Based Distance

Hann window

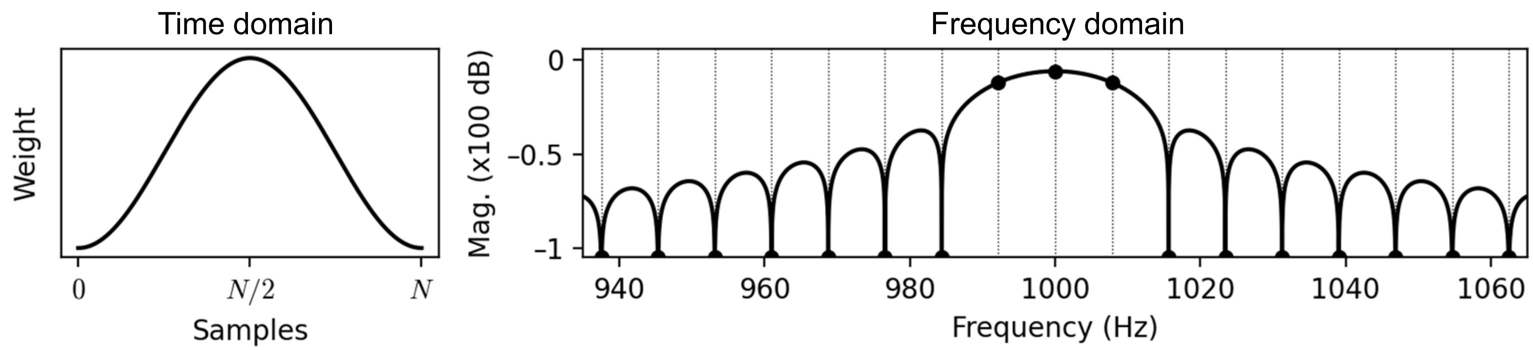


- Input signal: Sinusoid with frequency $f = 1000$ Hz
- STFT → Spectral leakage due to windowing
- Discrete STFT → **Frequency grid**
- Second signal: Sinusoid with frequency $f = 1020$ Hz

Distance depends on
- Grid sampling
- Mainlobe & sidelobes
- Window type
- STFT parameters

Dependency: Window Type

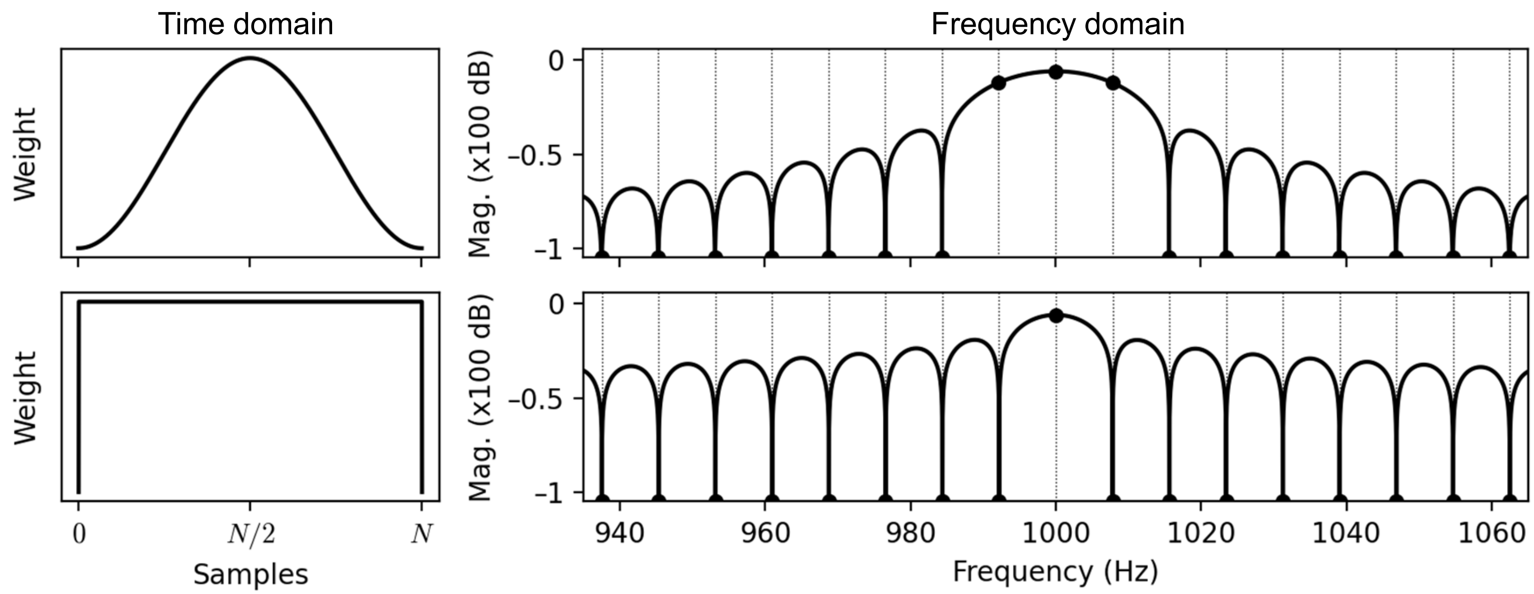
Hann window



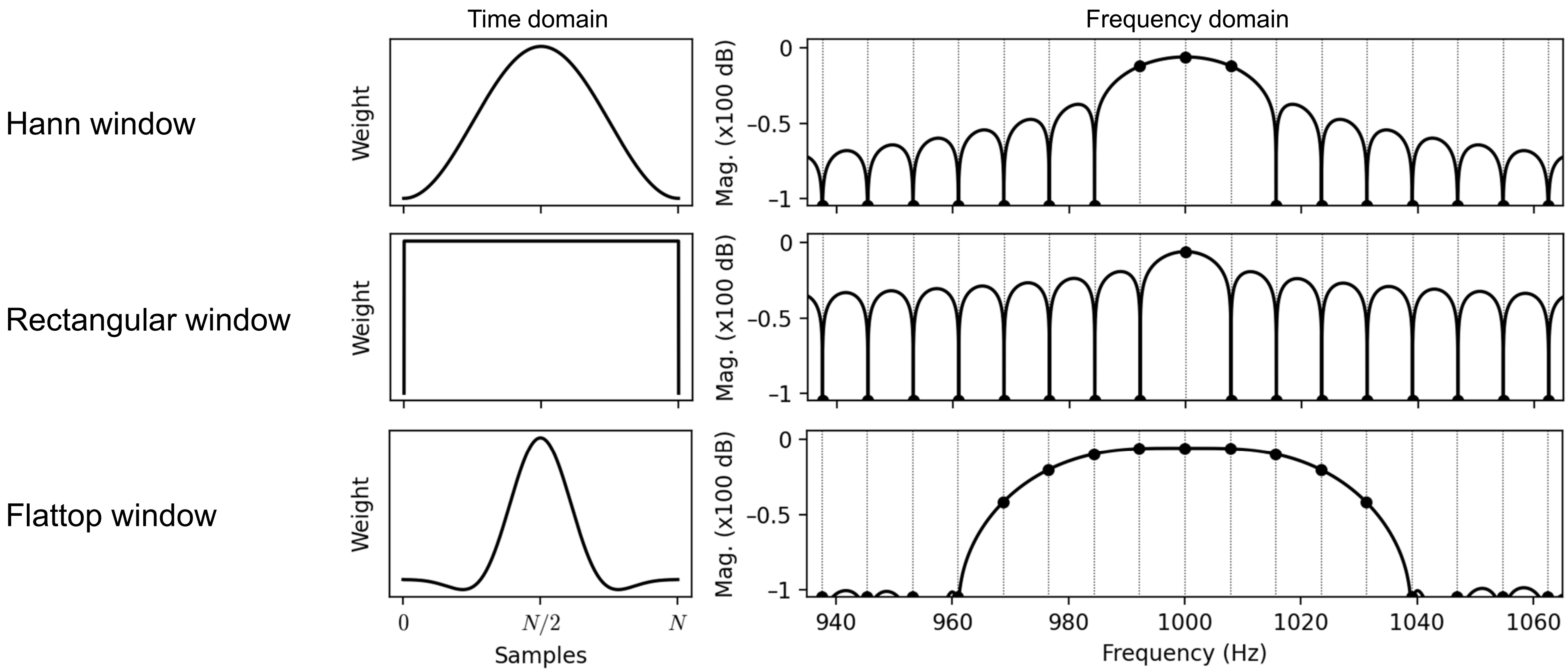
Dependency: Window Type

Hann window

Rectangular window



Dependency: Window Type

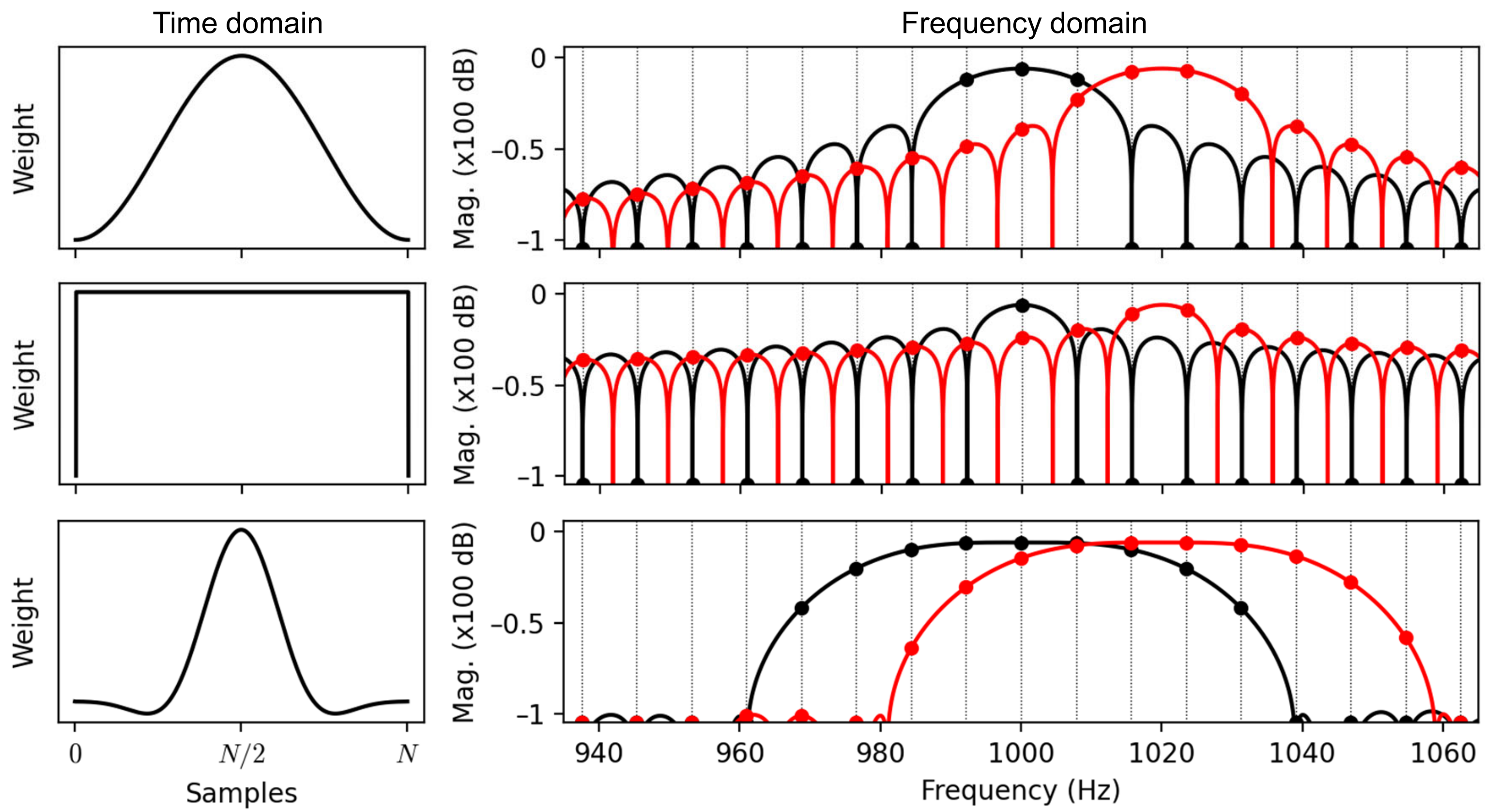


Dependency: Window Type

Hann window

Rectangular window

Flattop window



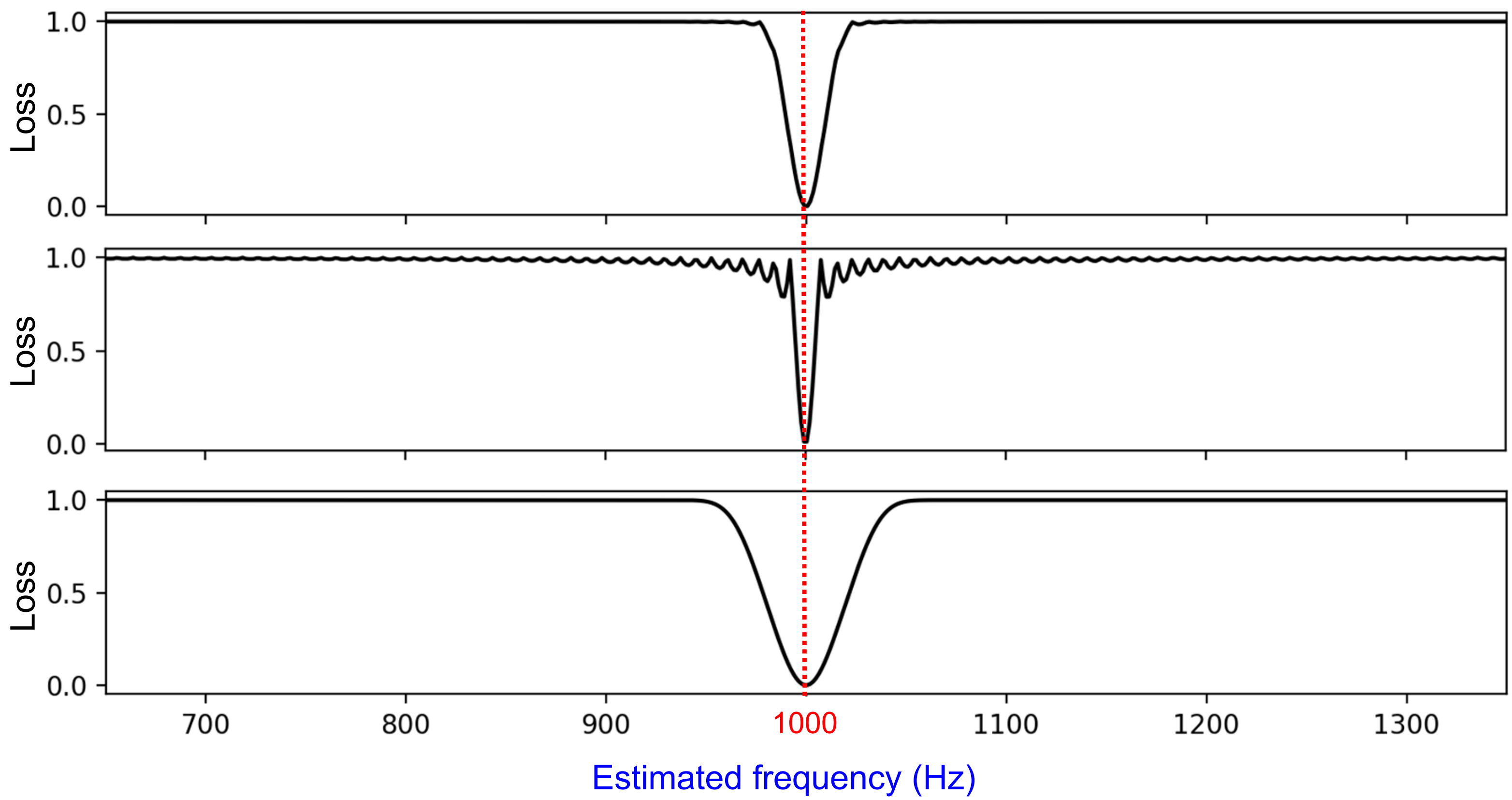
Dependency: Window Type

Loss landscape over estimates for a given target

Hann window

Rectangular window

Flattop window

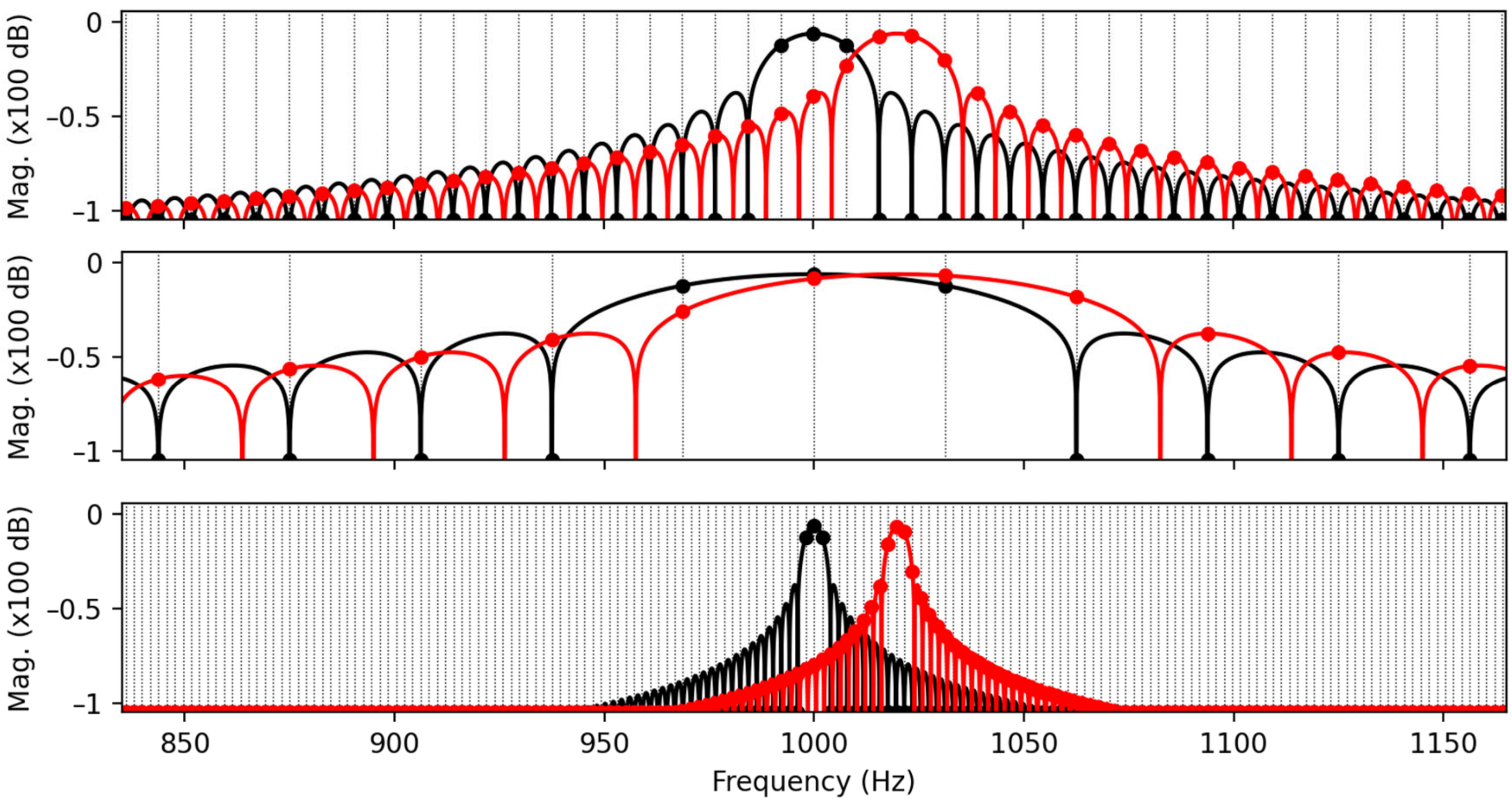


Dependency: Window Size

$N = 2048$

$N = 512$

$N = 8192$



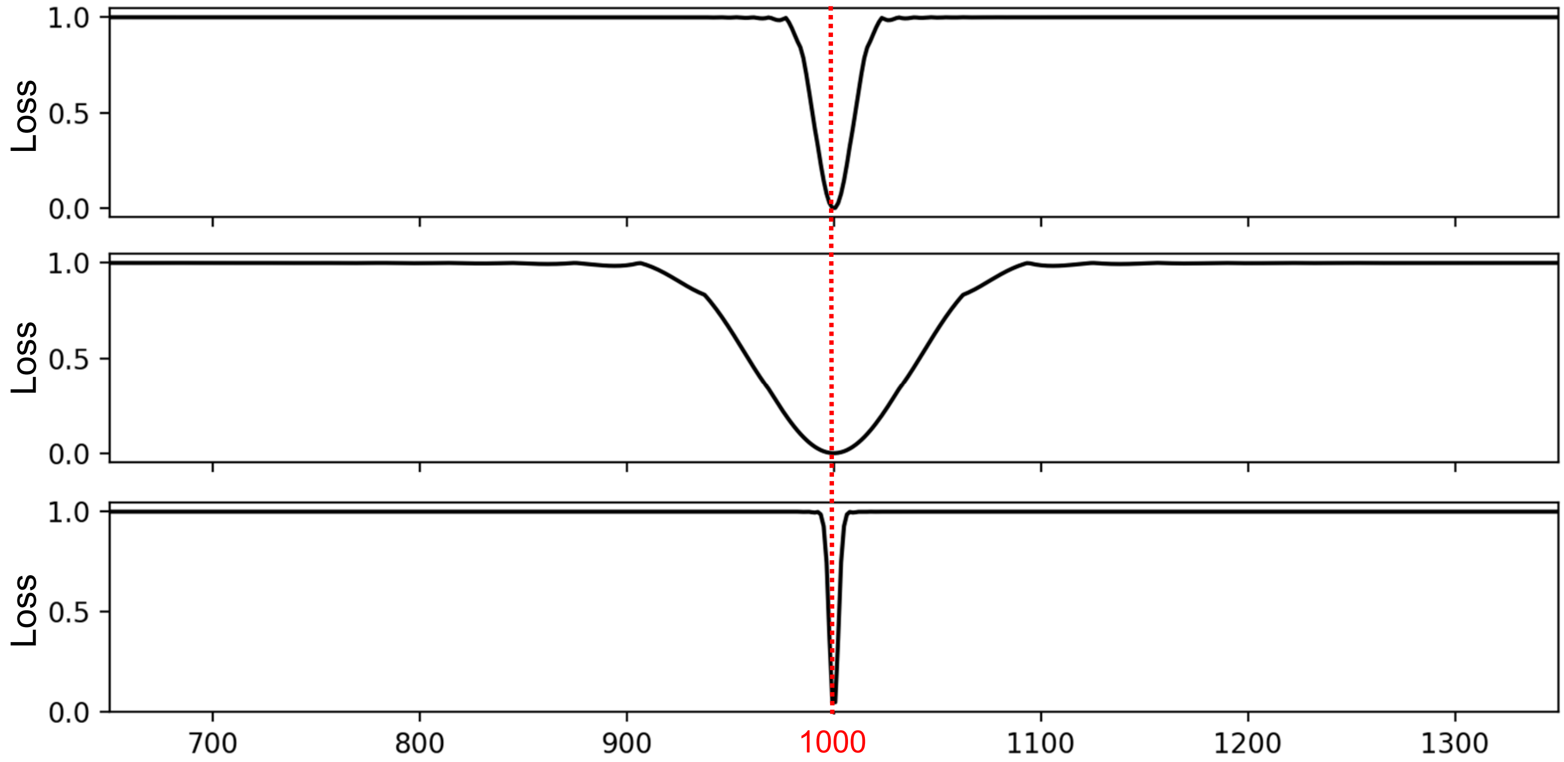
Dependency: Window Size

Loss landscape over estimates for a given target

$N = 2048$

$N = 512$

$N = 8192$

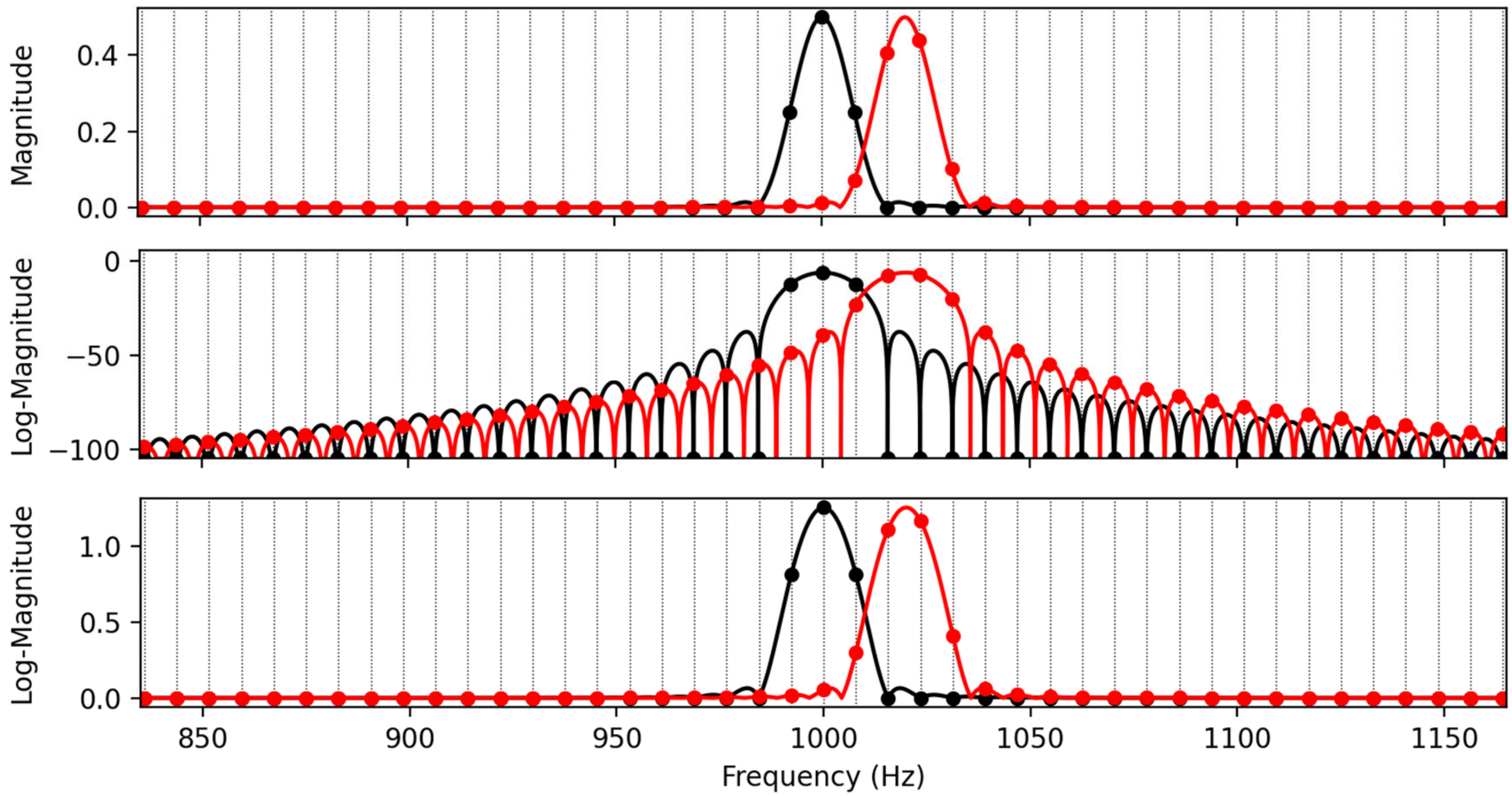


Dependency: Magnitude Compression

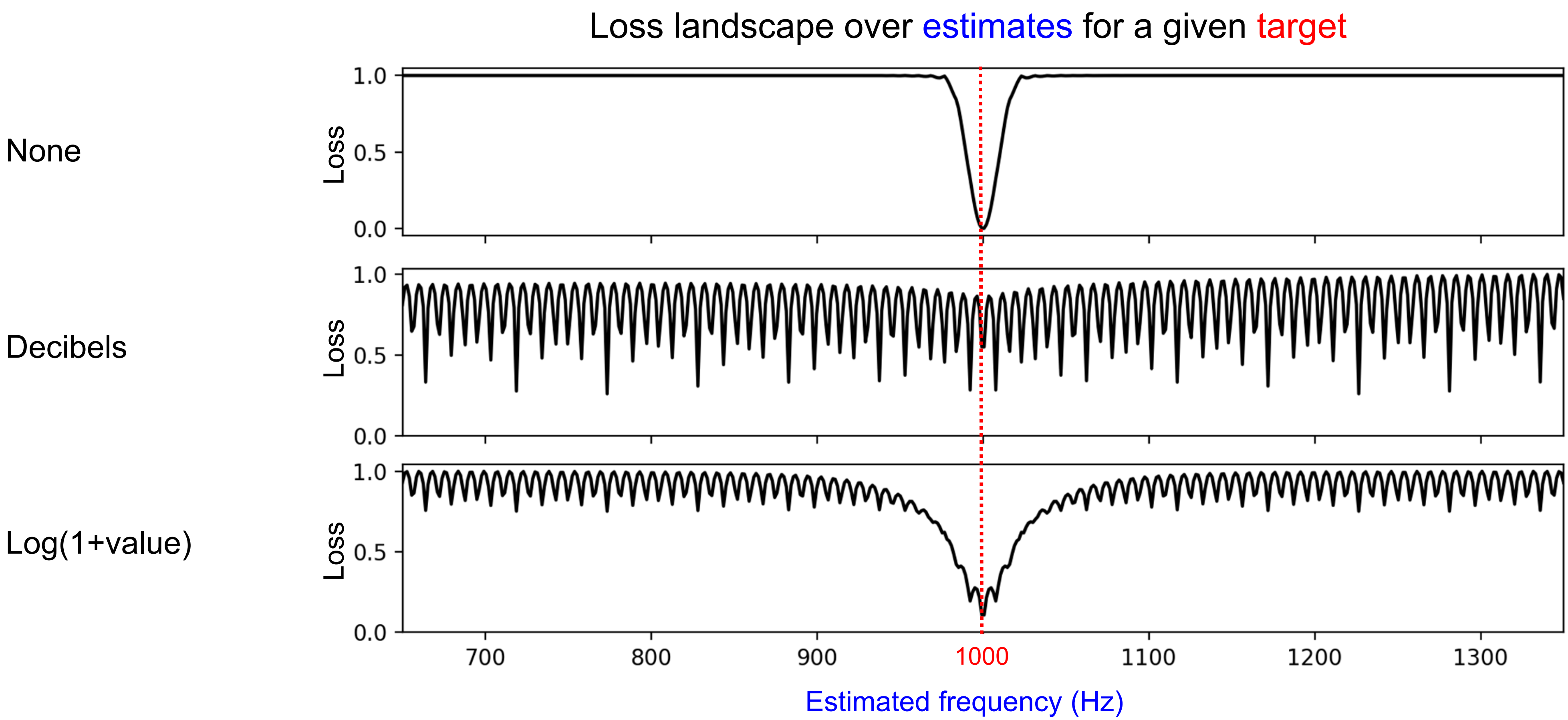
None

Decibels

Log(1+value)



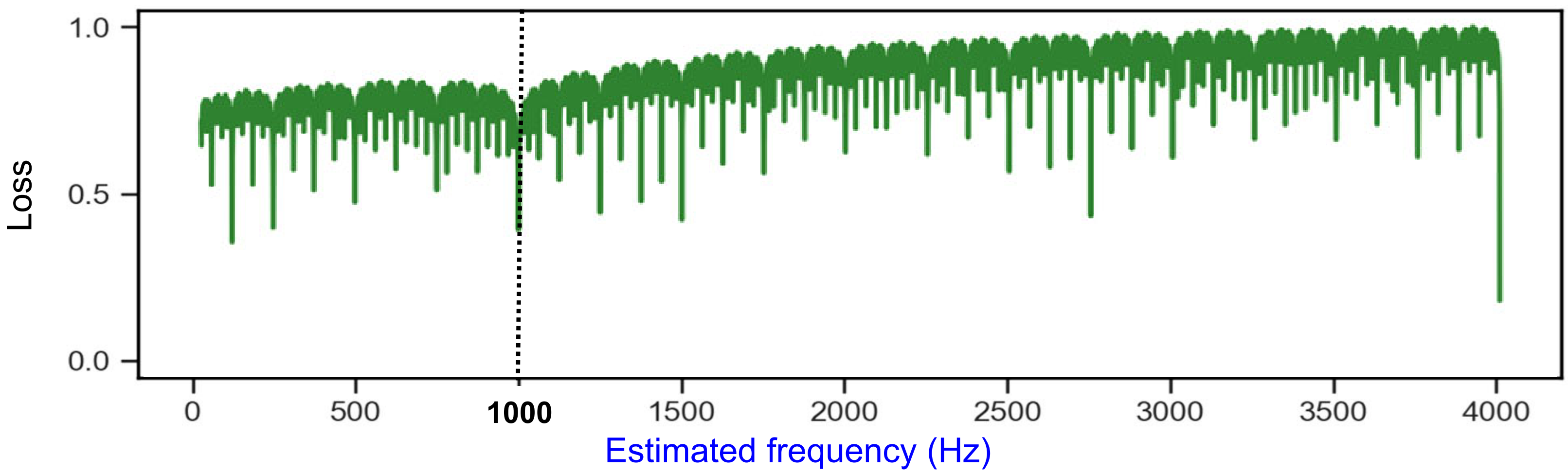
Dependency: Magnitude Compression



Experiments

- MSS loss with standard settings (WH, S4, C4, D1)

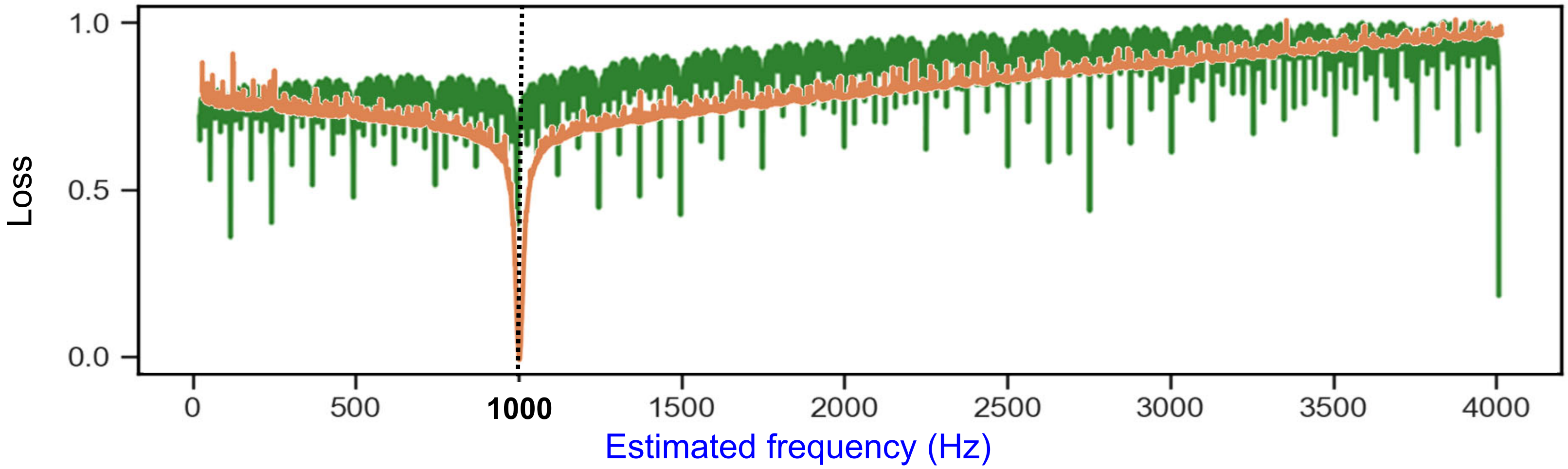
Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_2^2$



Experiments

- MSS loss with standard settings (WH, S4, C4, D1)
- Modified Hann MSS (WH, S5, C4, D2)

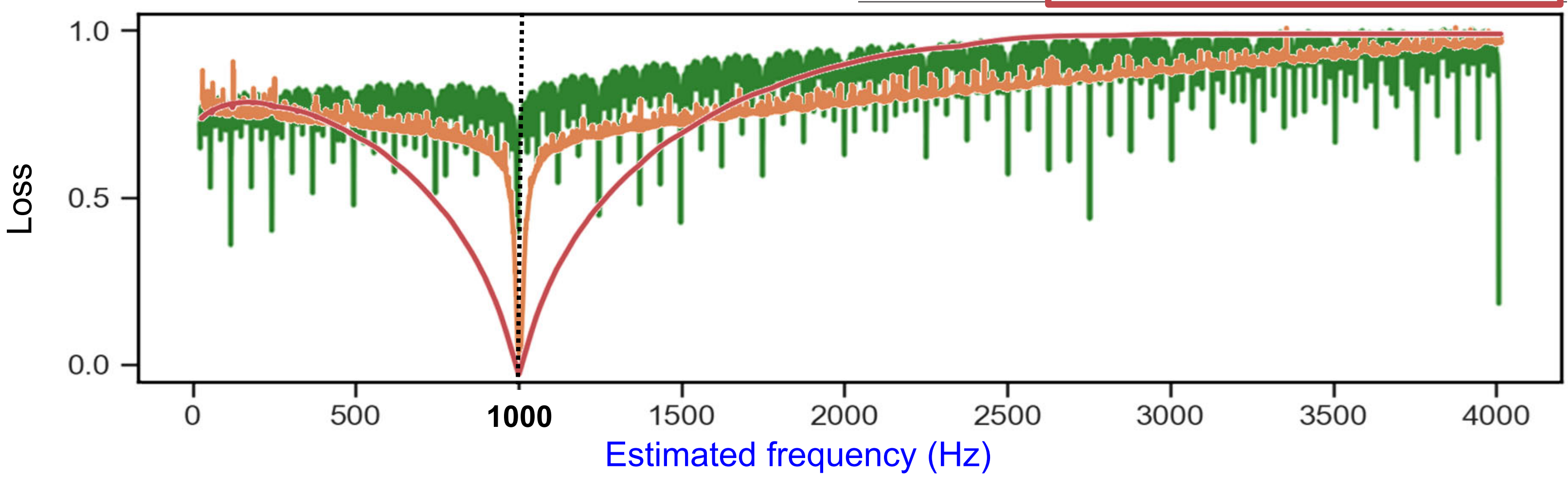
Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\},\ \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\},\ \gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\},\ \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x+\varepsilon)\},\ \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y},\hat{\mathcal{Y}}) = \Vert\mathcal{Y}-\hat{\mathcal{Y}}\Vert_1$
	D2	$d(\mathcal{Y},\hat{\mathcal{Y}}) = \Vert\mathcal{Y}-\hat{\mathcal{Y}}\Vert_2^2$



Experiments

- MSS loss with standard settings (WH, S4, C4, D1)
- Modified Hann MSS (WH, S5, C4, D2)
- Smooth MSS (WF, S5, C2, D2)

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}, \gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \lVert \mathcal{Y} - \hat{\mathcal{Y}} \rVert_2^2$

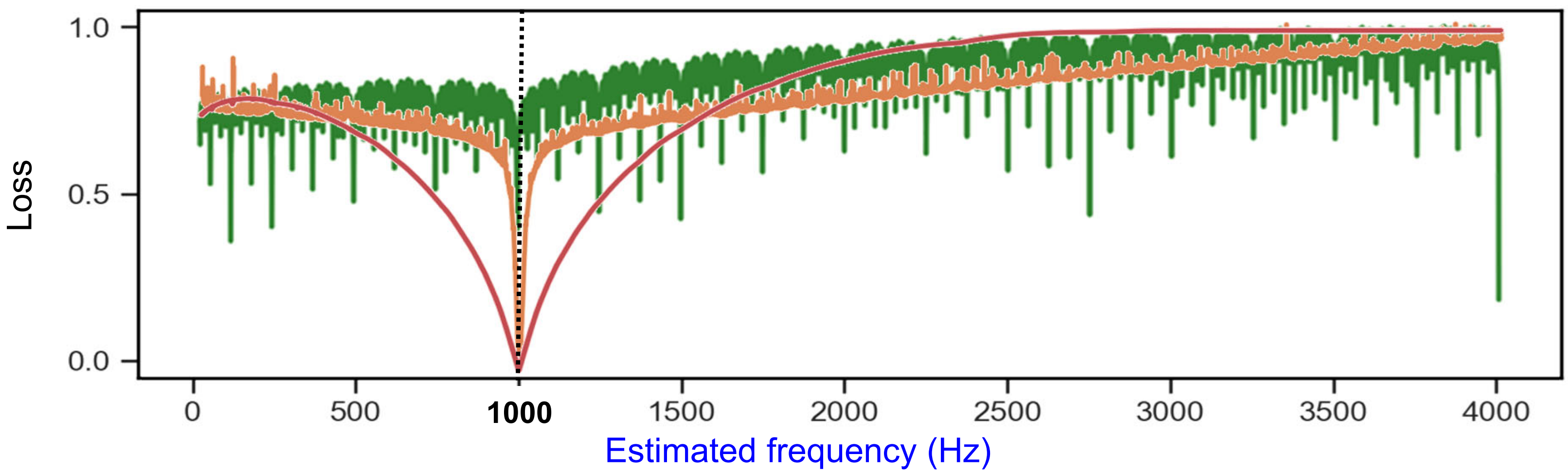


Experiments

GRA (Gradient-Sign Ranking Accuracy)

- Measures how often the loss gradient points in the correct direction.
- **Step size** distinguishes local gradient behavior from global trend.

Configuration	GRA			
Step Size	0.3 ct.	3 ct.	30 ct.	300 ct.
Standard MSS	0.523	0.529	0.573	0.775
Modified Hann MSS	0.613	0.635	0.708	0.923
Smooth MSS	0.999	0.993	0.952	0.860



Overview

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations

- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies

- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

Johannes Zeitler

Literature

- Silla, Freitas: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 22(1-29: 31–72, 2011.
- Wehrmann, Cerri, Barros: Hierarchical multi-label classification networks. Proc. ICML, 2018.
- **Krause**, Müller: Hierarchical Classification for Singing Activity, Gender, and Type in Complex Music Recordings. Proc. ICASSP, 2022.
- **Krause**, Müller: Hierarchical Classification for Instrument Activity Detection in Orchestral Music Recordings. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31: 2567–2578, 2023.
- Weiß, Arifi-Müller, **Krause**, Zalkow, Klauk, Kleinertz, Müller: Wagner Ring Dataset: A Complex Opera Scenario for Music Processing and Computational Musicology. Transaction of the International Society for Music Information Retrieval (TISMIR), 6(1): 135–149, 2023.

Wagner Ring Dataset

- Tetralogy (four operas)

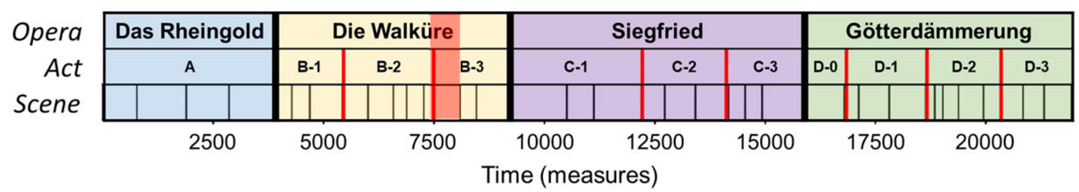
Wagner Ring Dataset

- Tetralogy (four operas)
- 11 Acts

Opera	Das Rheingold	Die Walküre			Siegfried			Götterdämmerung			
Act	A	B-1	B-2	B-3	C-1	C-2	C-3	D-0	D-1	D-2	D-3

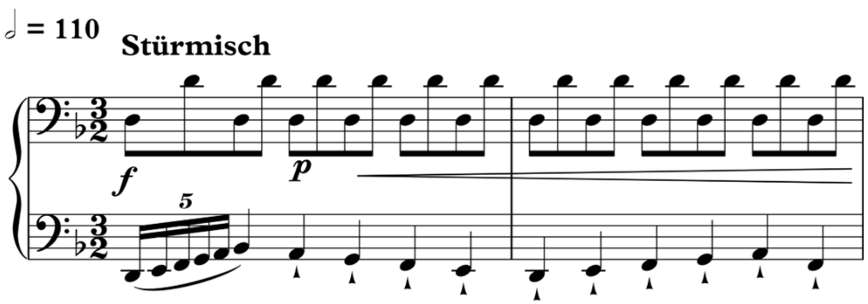
Wagner Ring Dataset

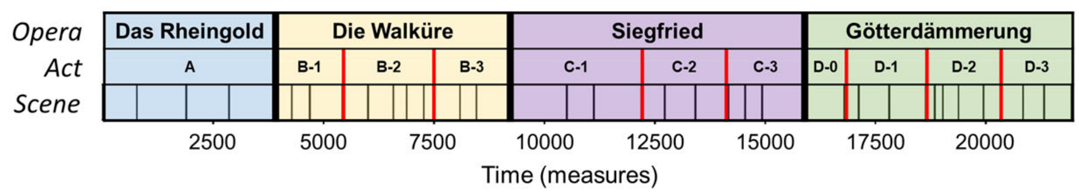
- Tetralogy (four operas)
- 11 Acts
- 21,939 measures



Wagner Ring Dataset
Raw Data

- Symbolic score:
  - Piano reduction
  - 822 pages

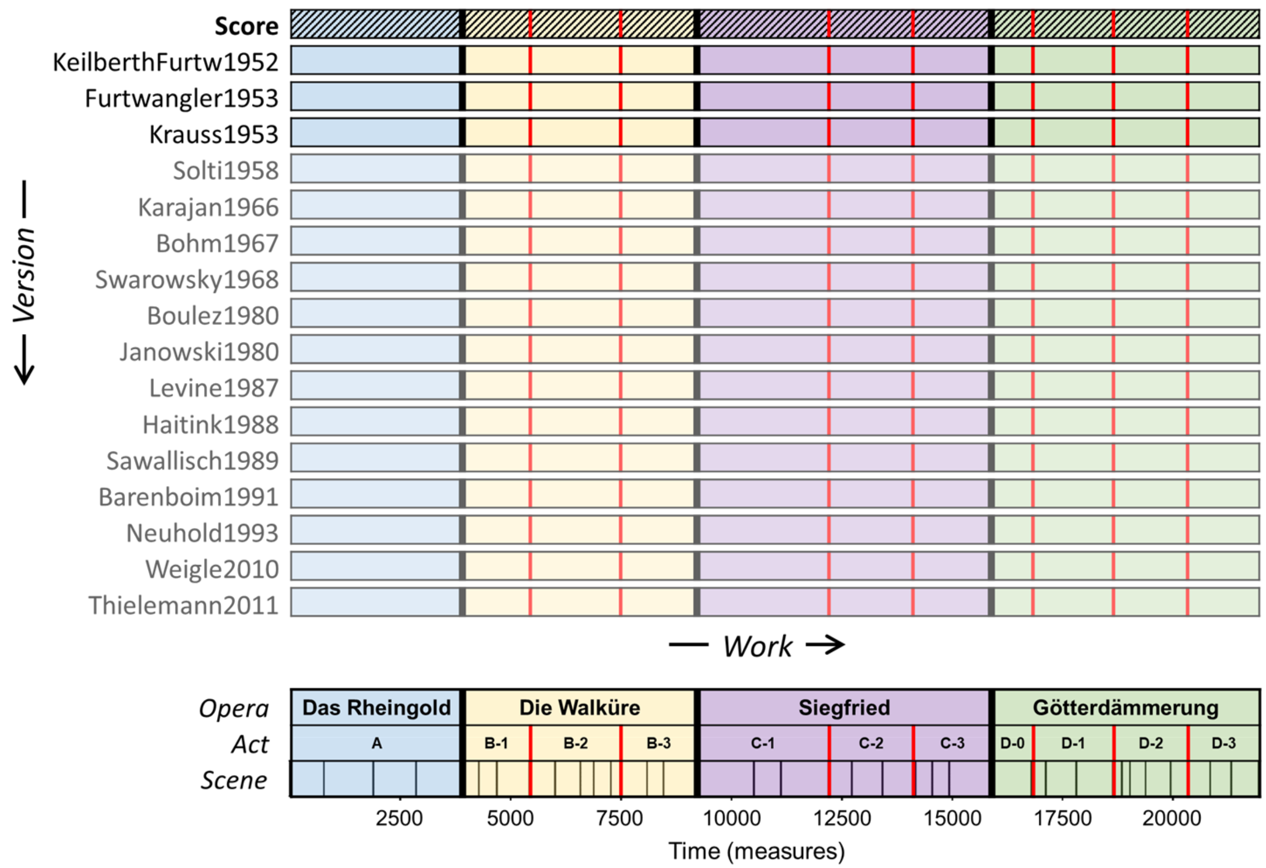




Wagner Ring Dataset

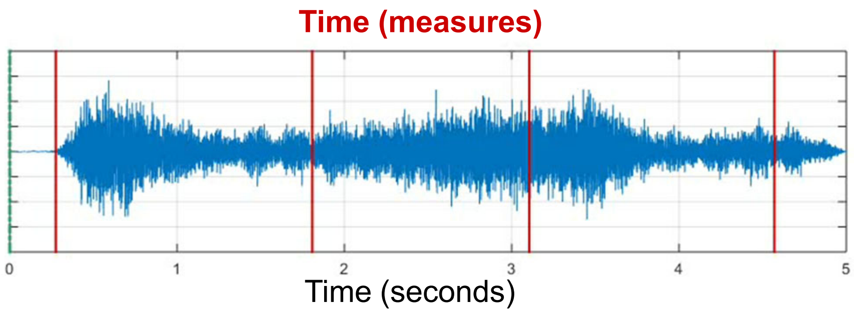
Raw Data

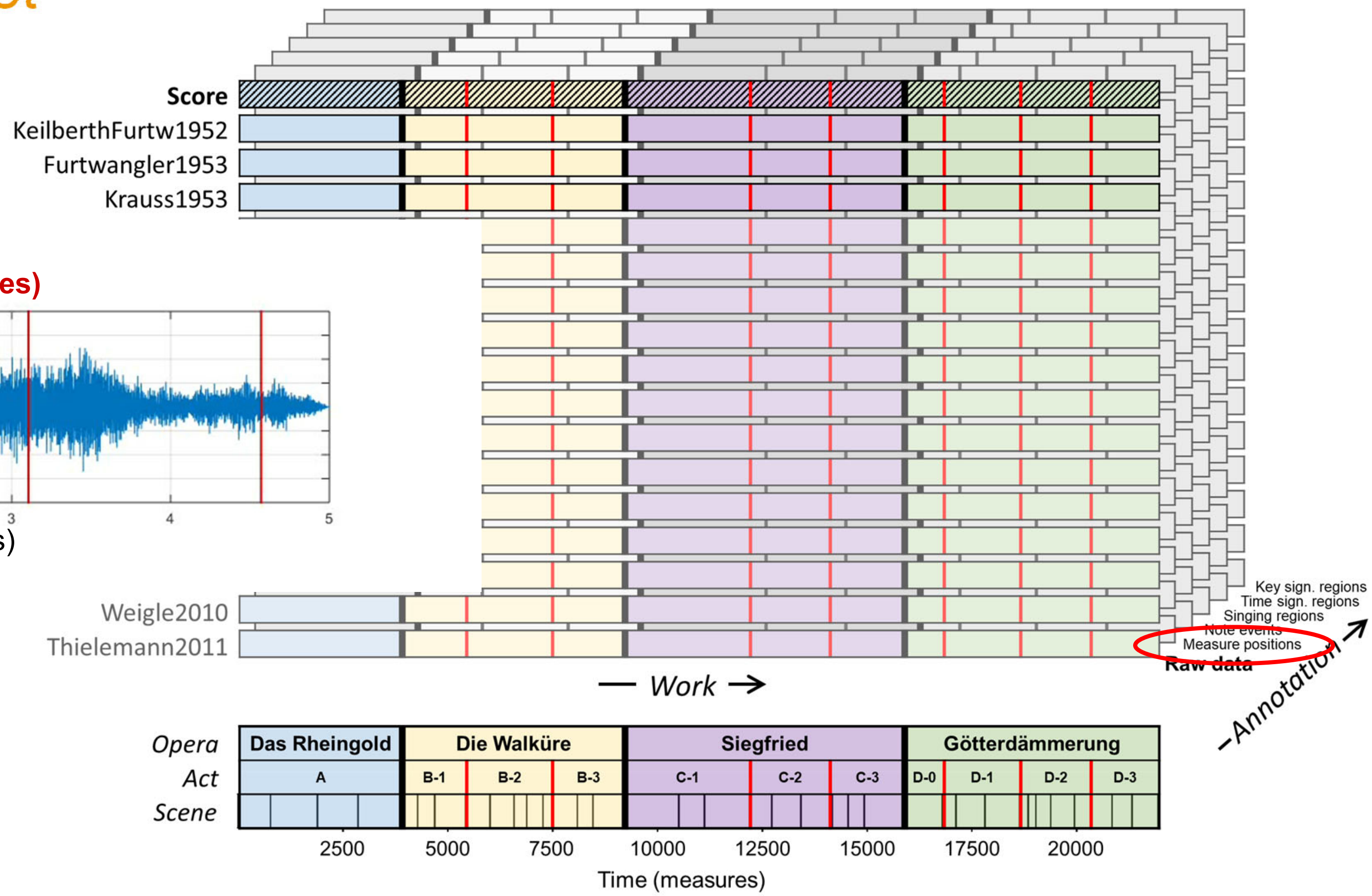
- Symbolic score:
  - Piano reduction
  - 822 pages
- Audio recordings:
  - 16 performances
  - 232 hours
  - 3 performances in Public Domain (EU)



Wagner Ring Dataset
Annotations

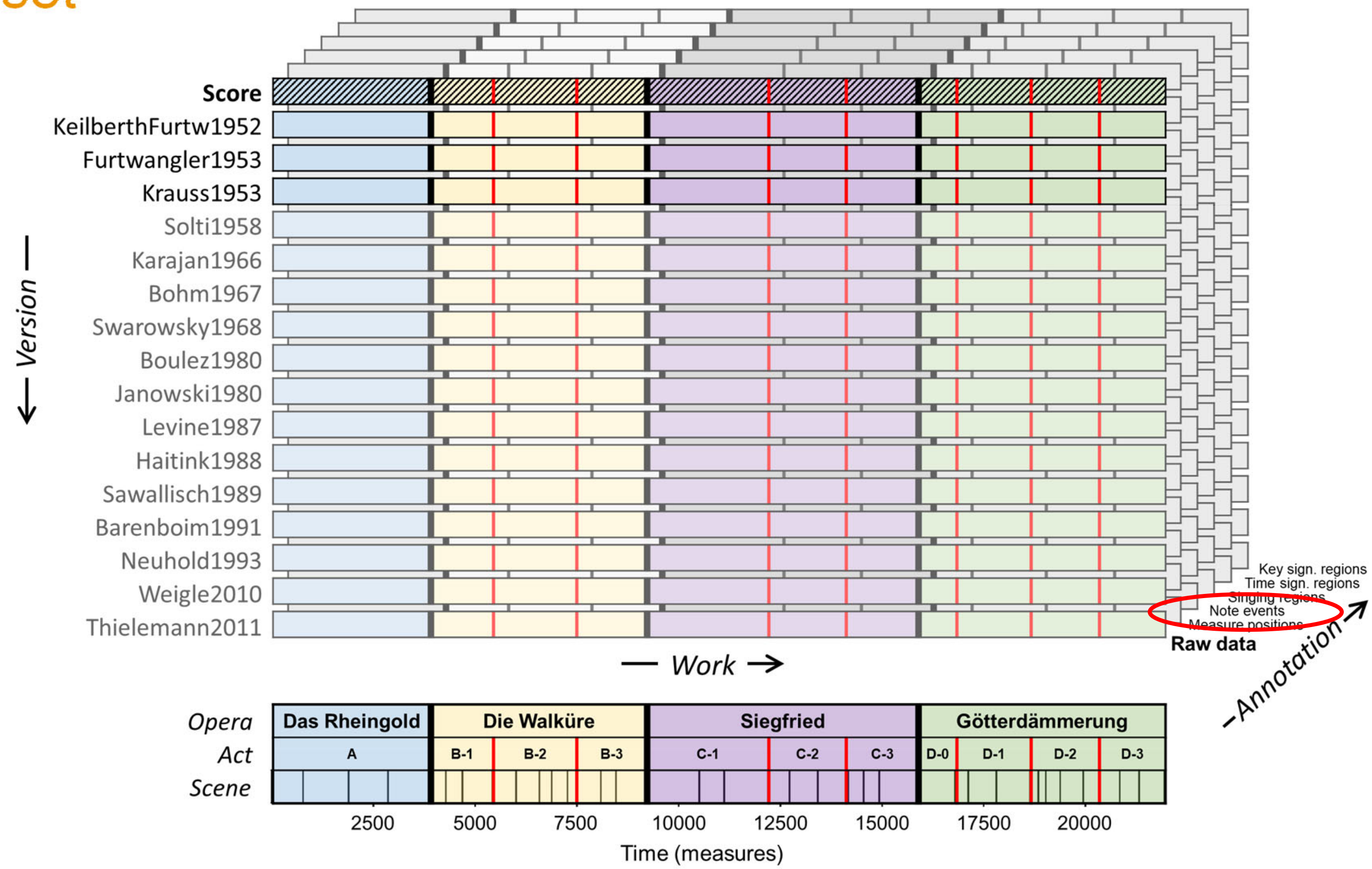
- Measure positions





Wagner Ring Dataset
Annotations

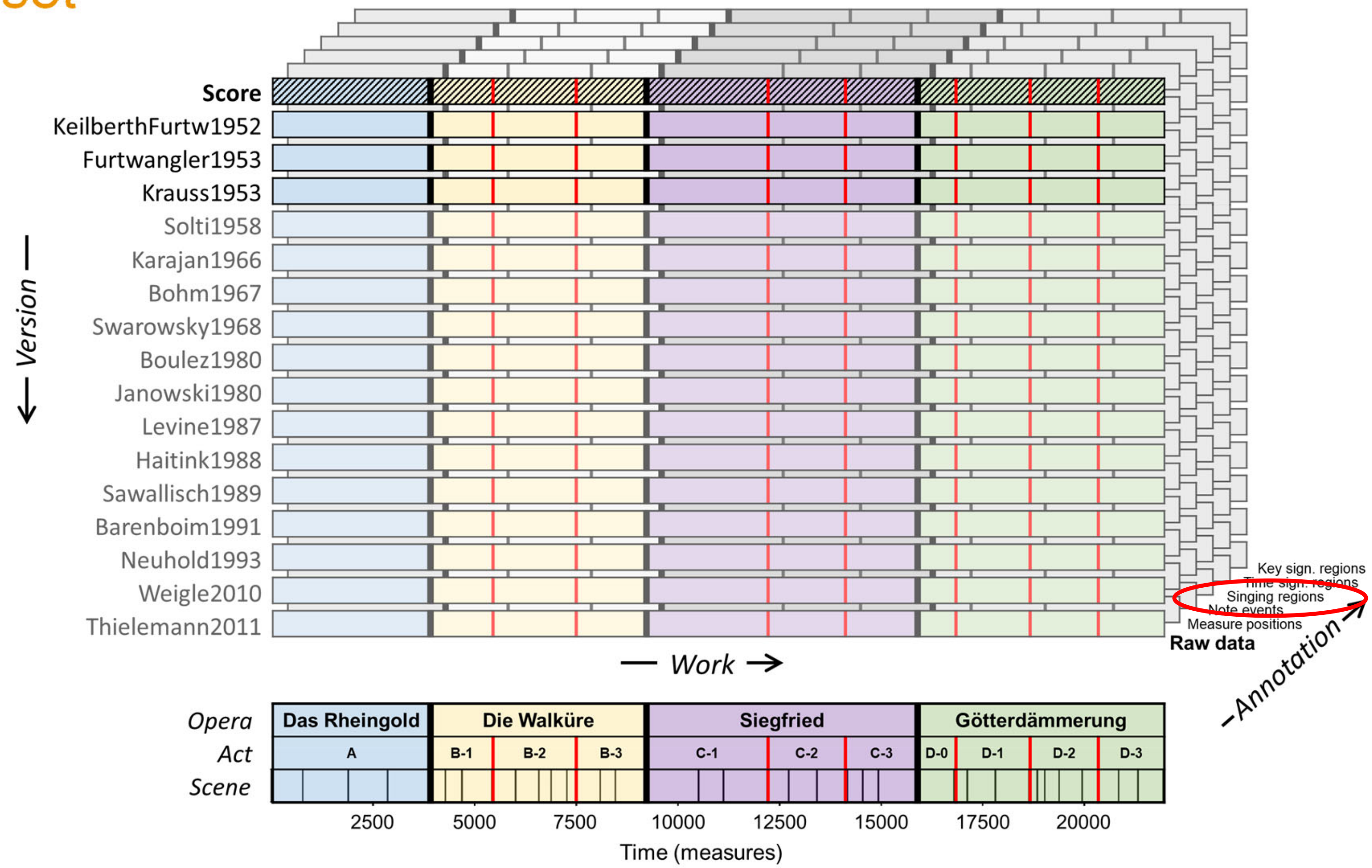
- Measure positions
- Note events



Wagner Ring Dataset

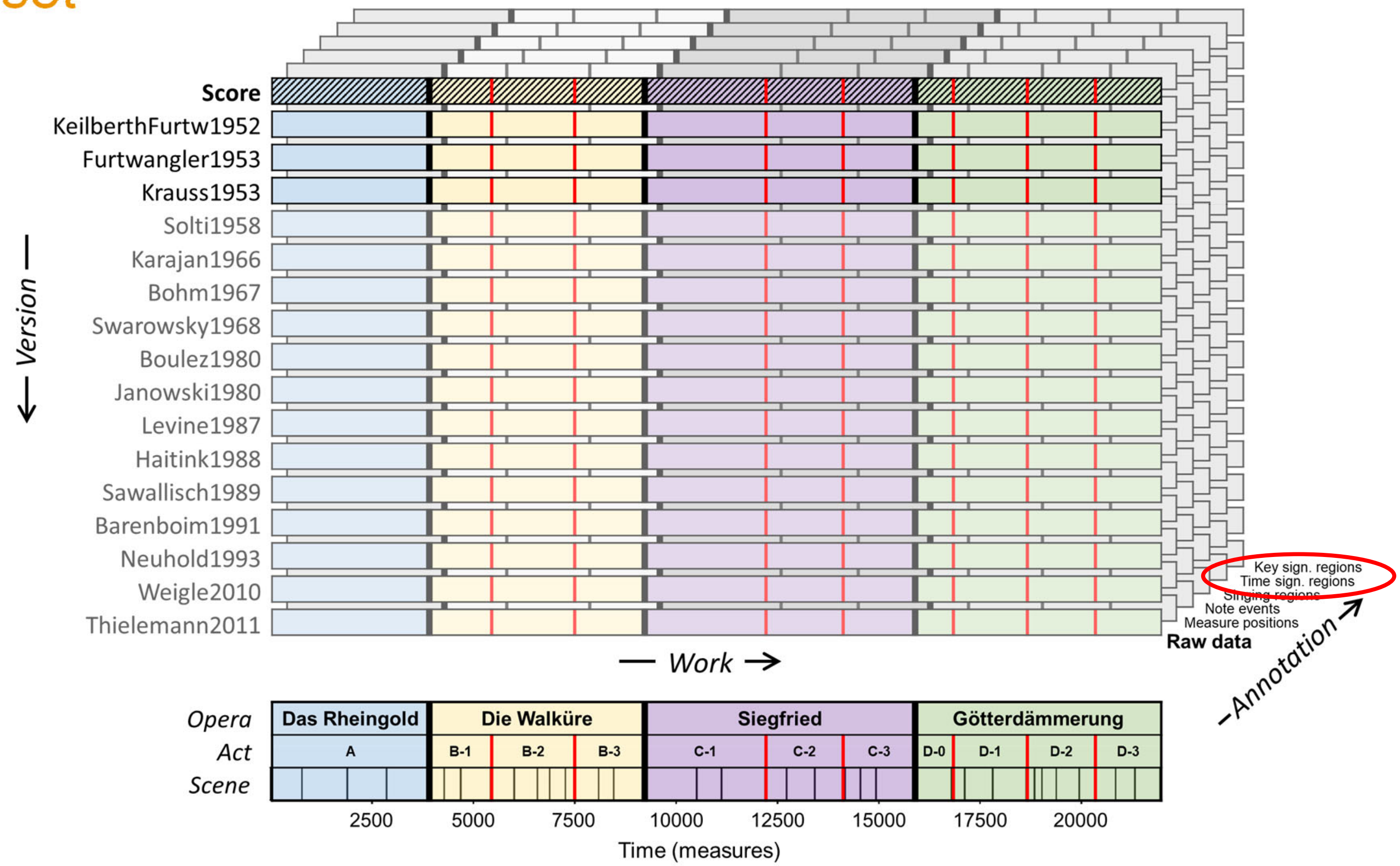
Annotations

- Measure positions
- Note events
- Singing regions



Wagner Ring Dataset
Annotations

- Measure positions
- Note events
- Singing regions
- Time signatures
- Key signatures



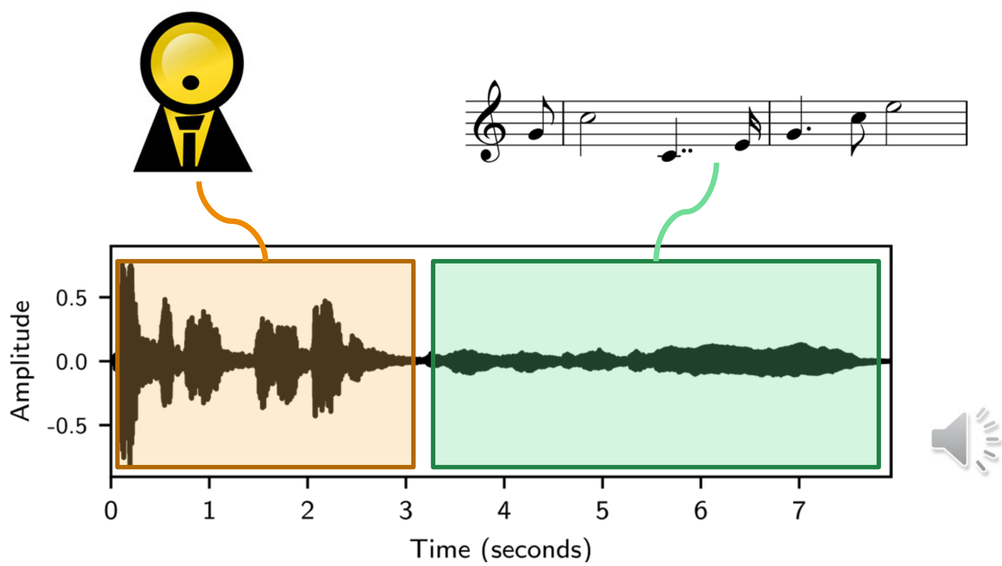
PhD Thesis by Michael Krause (2023)

Activity Detection for Sound Events in Orchestral Music Recordings

Singing Voice Detection

Leitmotif Detection

Instrument Detection



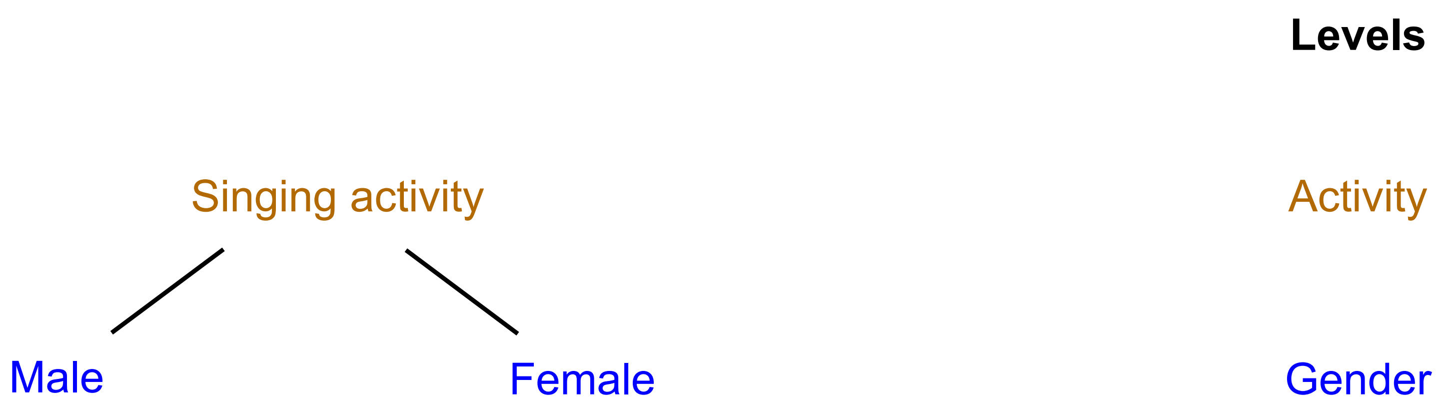
Hierarchical Classification
Singing Voice Detection

Levels

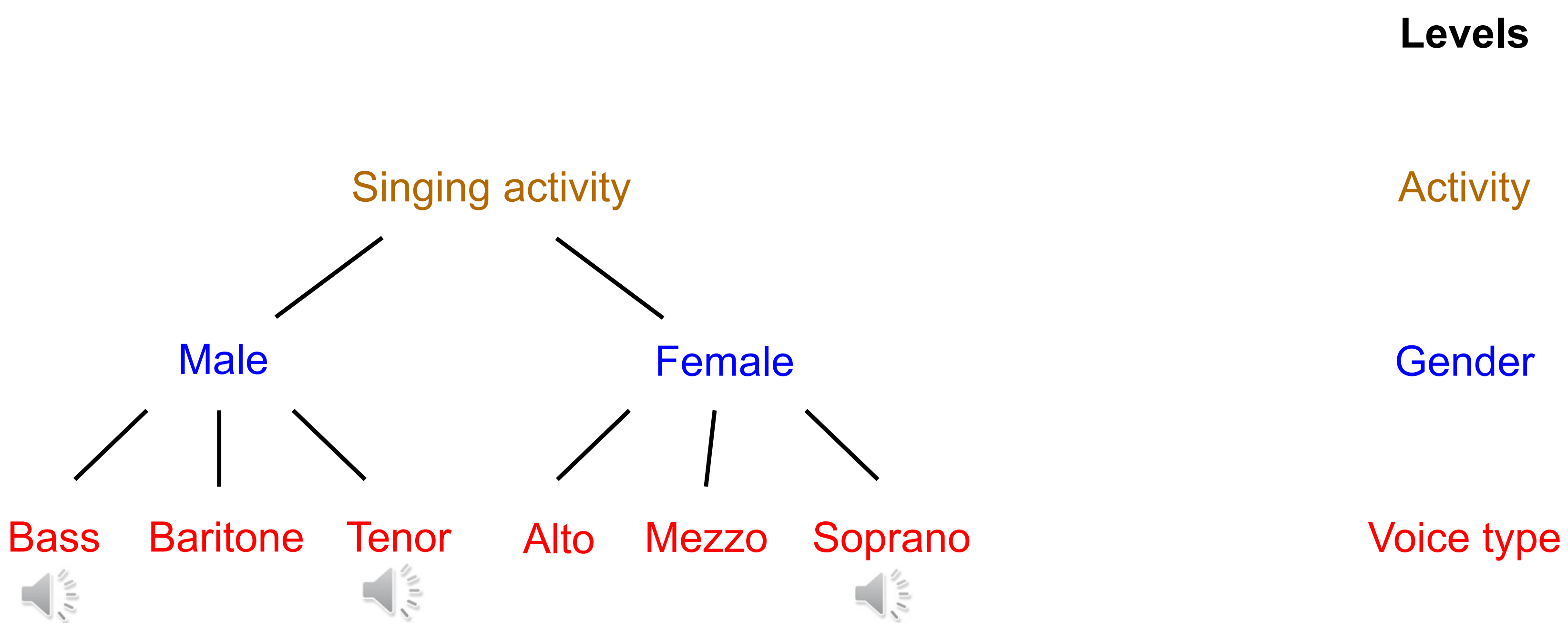
Singing activity

Activity

Hierarchical Classification
Singing Voice Detection



Hierarchical Classification
Singing Voice Detection

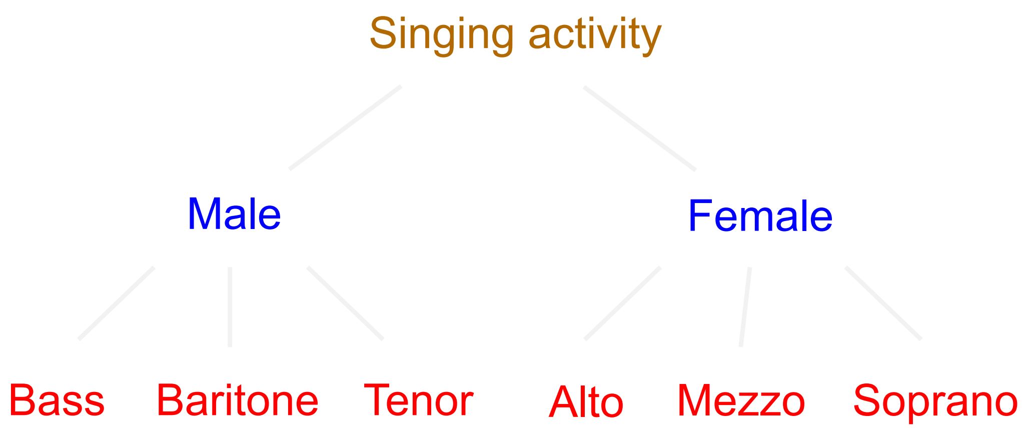


Hierarchical Strategies for Activity Detection

- Strategy A: Independent Decisions
- Strategy B: Bottom-Up Aggregation
- Strategy C: Top-Down Divide-and-Conquer
- Strategy D: Joint Classification
- Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses

Hierarchical Strategies for Activity Detection

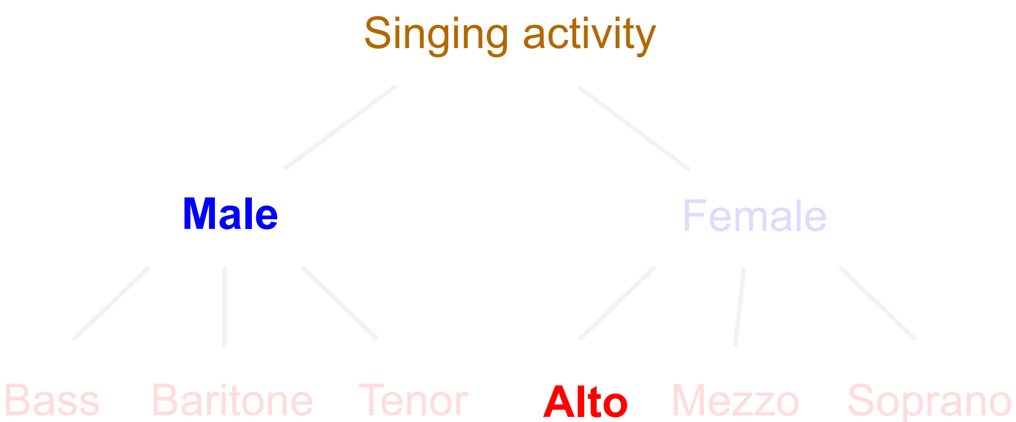
Strategy A: Independent Decisions



- Train and evaluate separate models for each hierarchy level
  - Activity classifier
  - Gender classifier
  - Voice type classifier

Hierarchical Strategies for Activity Detection

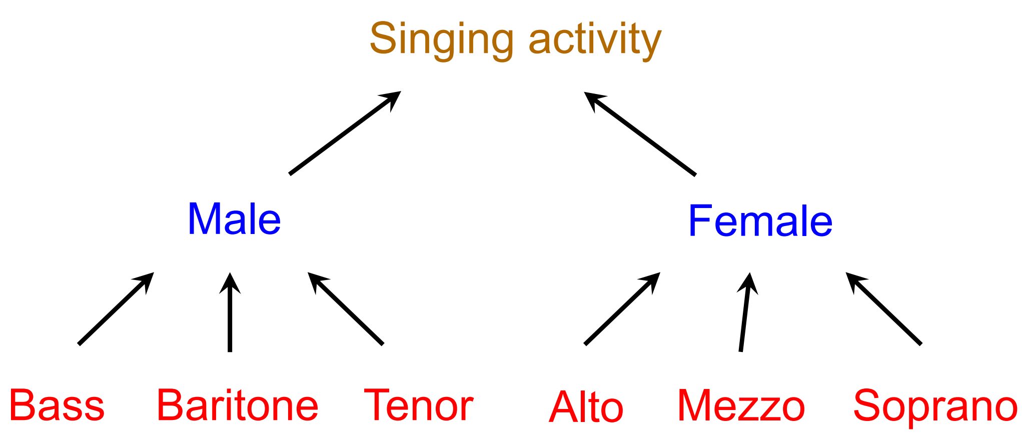
Strategy A: Independent Decisions



- Train and evaluate separate models for each hierarchy level
  - Activity classifier
  - Gender classifier
  - Voice type classifier
- Outputs may be inconsistent

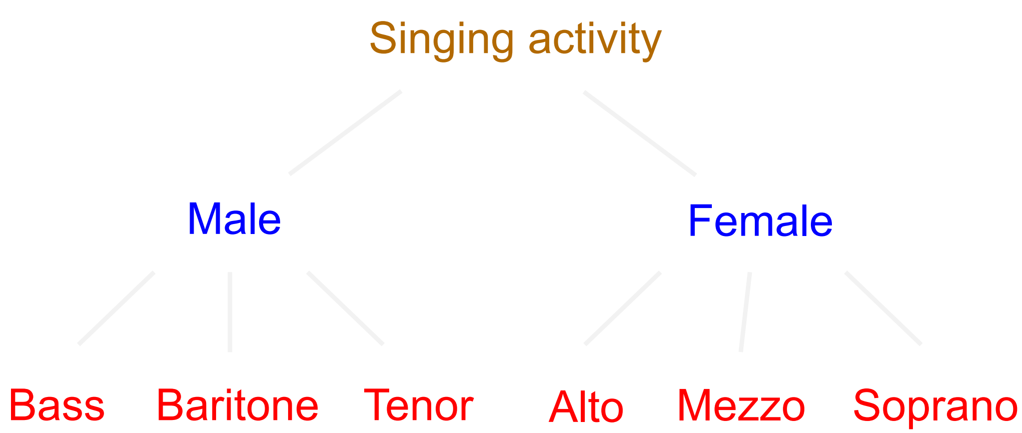
Hierarchical Strategies for Activity Detection

Strategy B: Bottom-Up Aggregation



- Train and evaluate a single model for the lowest hierarchy level
  - Voice type classifier
- Aggregate results from lower levels
- Consistency is trivially fulfilled
- May cause poor predictions on upper levels due to error propagation

Hierarchical Strategies for Activity Detection
Strategy D: Joint Classification



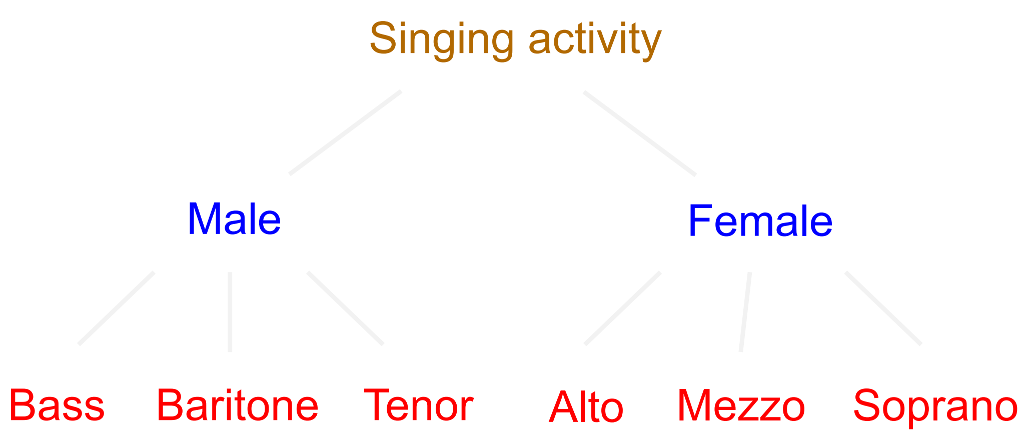
- Train and evaluate a single model for all classes

  → **Multi-task model**

- Need additional loss terms to promote consistent predictions

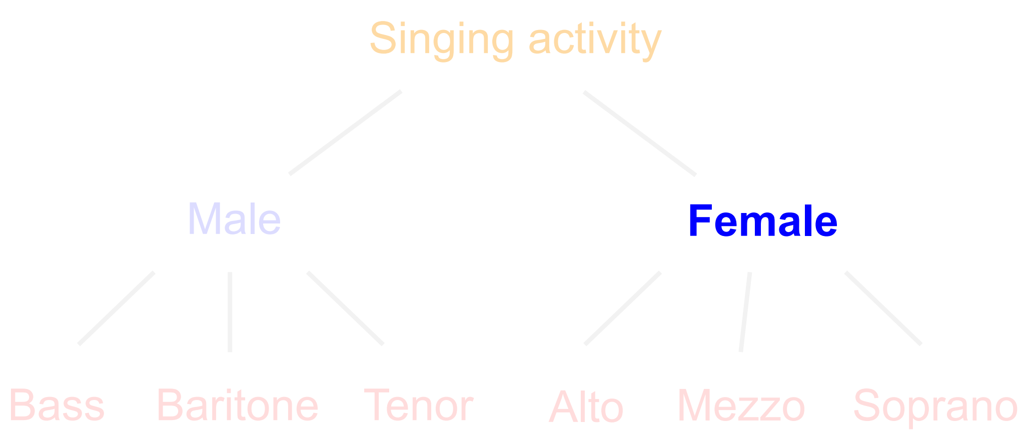
Hierarchical Strategies for Activity Detection

Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses



Hierarchical Strategies for Activity Detection

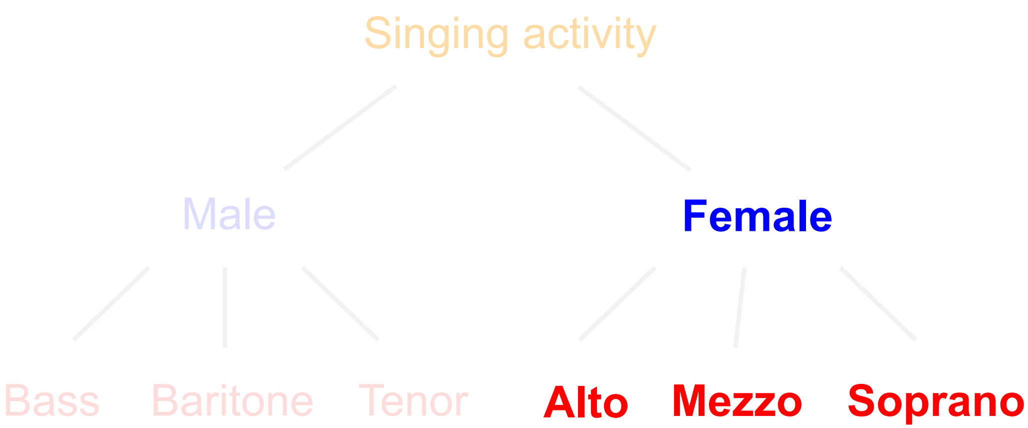
Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses



- Notation:
  - c: a class
  - p_c: probability of c

Hierarchical Strategies for Activity Detection

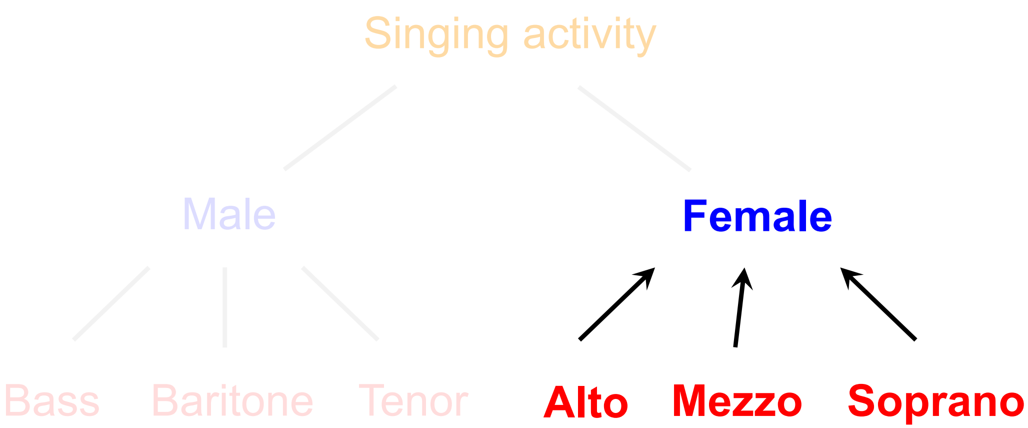
Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses



- Notation:
  - c: a class
  - p_c: probability of c
  - $c\downarrow$: child classes of c

Hierarchical Strategies for Activity Detection

Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses



- Notation:
  - c: a class
  - p_c: probability of c
  - $c{\downarrow}$: child classes of c

- For **bottom-up** consistency, minimize

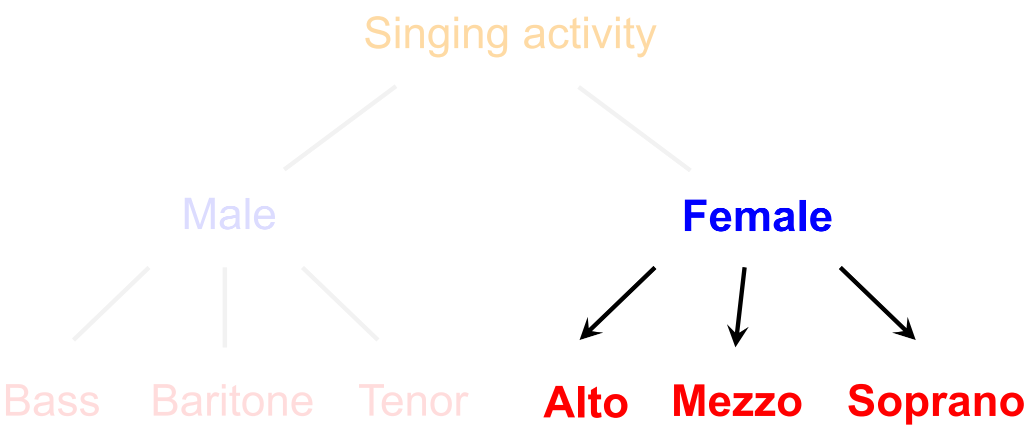
$$\sum_{c' \in c\downarrow} \max\{0, p_{c'} - p_c\}^2$$

p_c should be at least as high as any $p_{c'}$

$\rightarrow$ penalty for every $p_{c'} > p_c$

Hierarchical Strategies for Activity Detection

Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses



- Notation:
  - c: a class
  - p_c: probability of c
  - $c{\downarrow}$: child classes of c
- For **top-down** consistency, minimize

$$\max\{0, p_c - \max_{c' \in c\downarrow} p_{c'}\}^2$$

p_c should not be above largest $p_{c'}$

Hierarchical Strategies for Activity Detection

Strategy $D^{\alpha,\beta}$: Joint Classification with Consistency Losses

Bottom-up loss term:

$$\mathcal{L}_{\uparrow} = \frac{1}{|\mathbf{C} \setminus \mathbf{C}^H|} \sum_{h=2}^{H} \sum_{c \in \mathbf{C}^h} \sum_{c' \in c\downarrow} \max\{0, p_{c'} - p_c\}^2$$

Top-down loss term:

$$\mathcal{L}_{\downarrow} = \frac{1}{|\mathbf{C} \setminus \mathbf{C}^1|} \sum_{h=2}^{H} \sum_{c \in \mathbf{C}^h} \max\{0, p_c - \max_{c' \in c\downarrow} p_{c'}\}^2$$

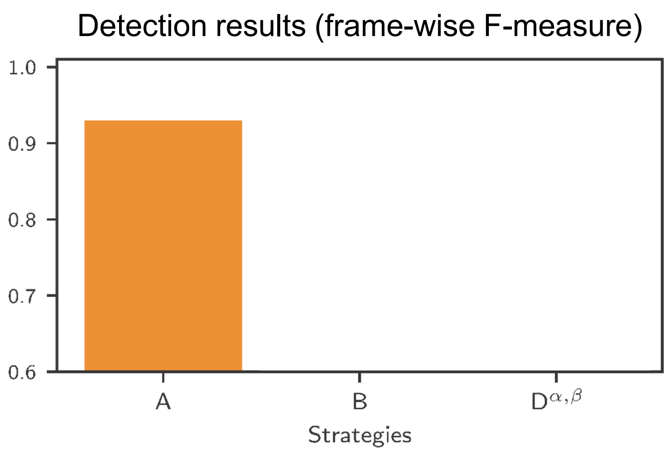
Joint loss term:

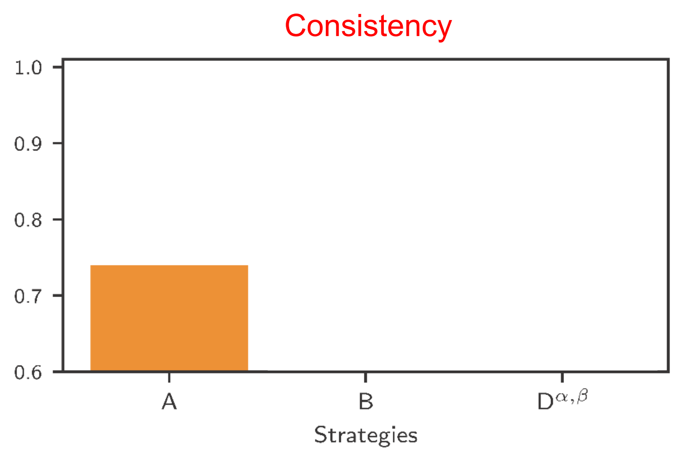
$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \alpha \mathcal{L}_{\downarrow} + \beta \mathcal{L}_{\uparrow}$$

Notation

- $\mathbf{C}$ All classes
- $\mathbf{C}^h$ Classes at level h
- H Number of levels
- $c\downarrow$ Children of c
- p_c Probability for c

Results: Female Singing





Consistency

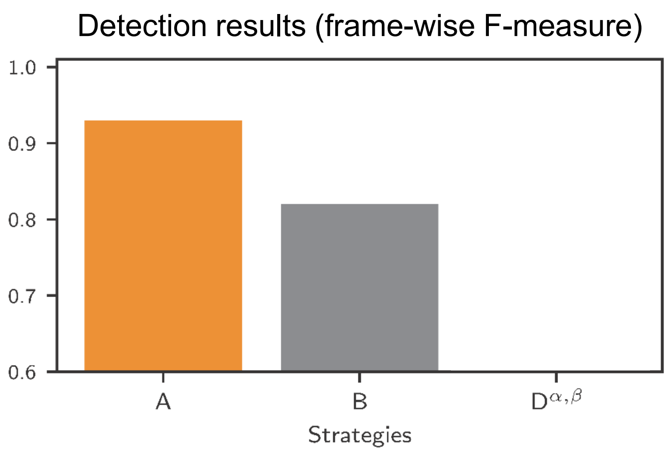
$\mathcal{I}_c^{\mathrm{Est}}$ Frames predicted as *c*

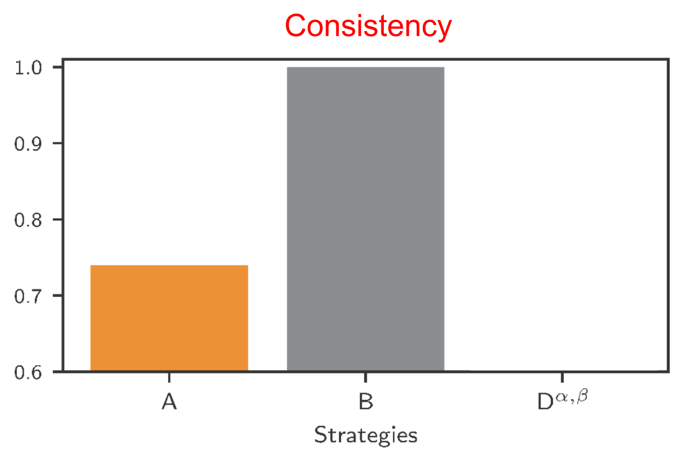
$\mathcal{I}_{c\downarrow}^{\mathrm{Est}}$ Frames predicted as child of *c*

$$\gamma_c = \frac{|\mathcal{I}_c^{\mathrm{Est}} \cap \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}{|\mathcal{I}_c^{\mathrm{Est}} \cup \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}$$

- Strategy A (Independent Decisions) yields good but inconsistent results

Results: Female Singing





Consistency

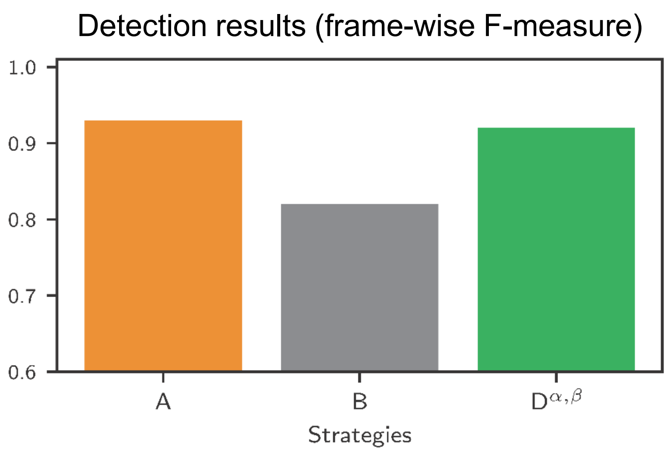
$\mathcal{I}_c^{\mathrm{Est}}$ Frames predicted as *c*

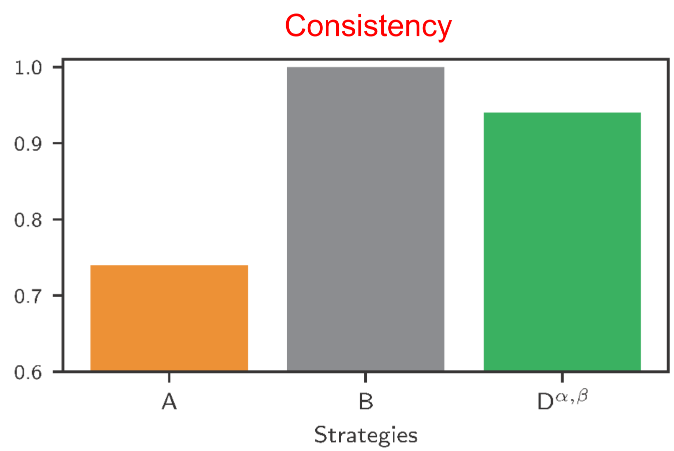
$\mathcal{I}_{c\downarrow}^{\mathrm{Est}}$ Frames predicted as child of *c*

$$\gamma_c = \frac{|\mathcal{I}_c^{\mathrm{Est}} \cap \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}{|\mathcal{I}_c^{\mathrm{Est}} \cup \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}$$

- Strategy A (Independent Decisions) yields good but inconsistent results
- Strategy B (Bottom-Up Aggregation) gives worse but consistent results

Results: Female Singing





Consistency

$\mathcal{I}_c^{\mathrm{Est}}$ Frames predicted as *c*

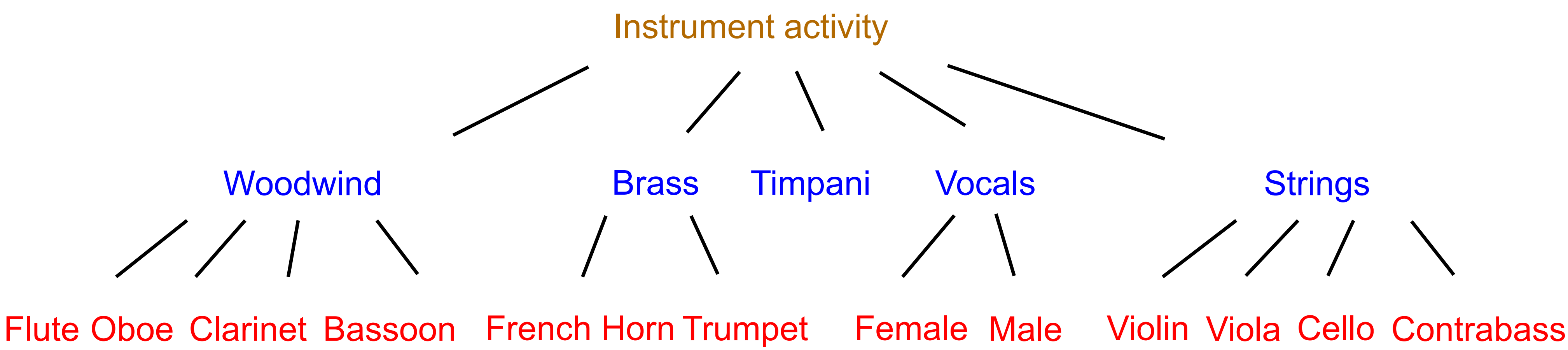
$\mathcal{I}_{c\downarrow}^{\mathrm{Est}}$ Frames predicted as child of *c*

$$\gamma_c = \frac{|\mathcal{I}_c^{\mathrm{Est}} \cap \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}{|\mathcal{I}_c^{\mathrm{Est}} \cup \mathcal{I}_{c\downarrow}^{\mathrm{Est}}|}$$

- Strategy A (Independent Decisions) yields good but inconsistent results
- Strategy B (Bottom-Up Aggregation) gives worse but consistent results
- Strategy $D^{\alpha,\beta}$ (Joint with Consistency Losses) provides good trade-off

Scenario: Hierarchical Instrument Classification

- Musical instruments can naturally be arranged into hierarchies



- Instrument-level annotations hard to obtain

Overview

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations

- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies

- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

Johannes Zeitler

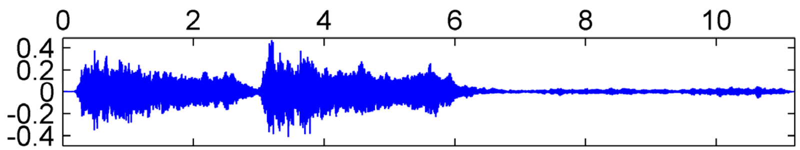
Literature

- Cuturi, Blondel: Soft-DTW: A Differentiable Loss Function for Time-Series. ICML, 2017.
- Blondel, Mensch, Vert: Differentiable Divergences Between Time Series. AISTATS, 2021.
- **Krause**, Weiß, Müller: Soft Dynamic Time Warping For Multi Pitch Estimation And Beyond. Proc. ICASSP, 2023.
- **Zeitler**, Deniffel, **Krause**, Müller: Stabilizing Training with Soft Dynamic Time Warping: A Case Study for Pitch Class Estimation with Weakly Aligned Targets. Proc. ISMIR, 2023.
- **Zeitler**, **Krause**, Müller: Soft Dynamic Time Warping with Variable Step Weights. Proc. ICASSP, 2024.

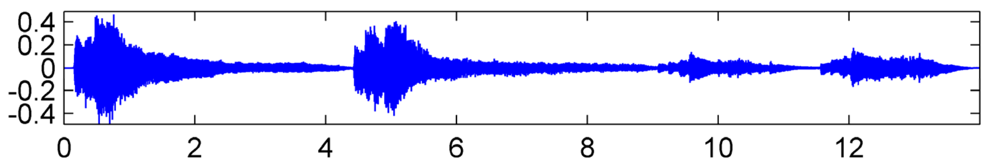
Motivation: Audio-Audio Alignment

Beethoven's Fifth

Karajan
(Orchester)



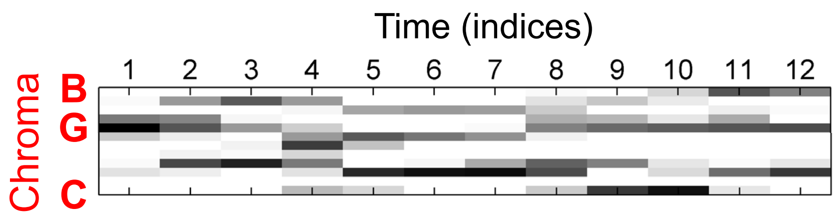
Gould
(Piano)



Motivation: Audio-Audio Alignment

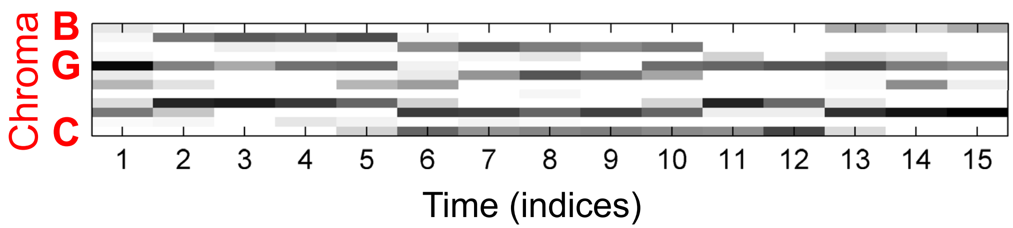
Beethoven’s Fifth

Karajan
(Orchester)



Time–chroma representations

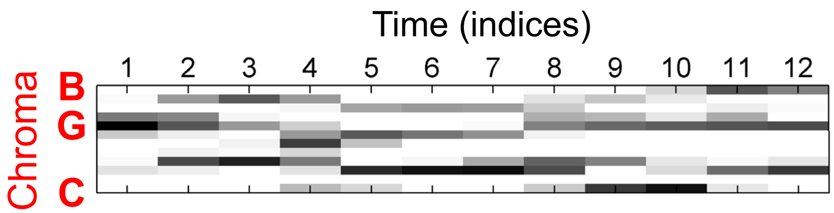
Gould
(Piano)



Motivation: Audio-Audio Alignment

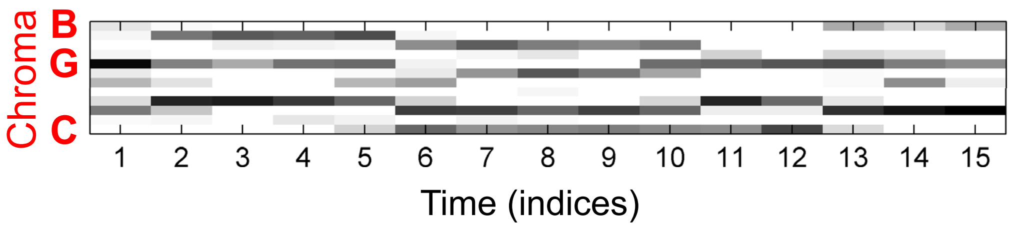
Beethoven's Fifth

Karajan
(Orchester)



Time–chroma representations

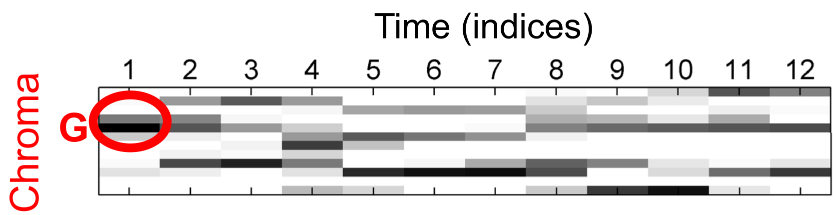
Gould
(Piano)



Motivation: Audio-Audio Alignment

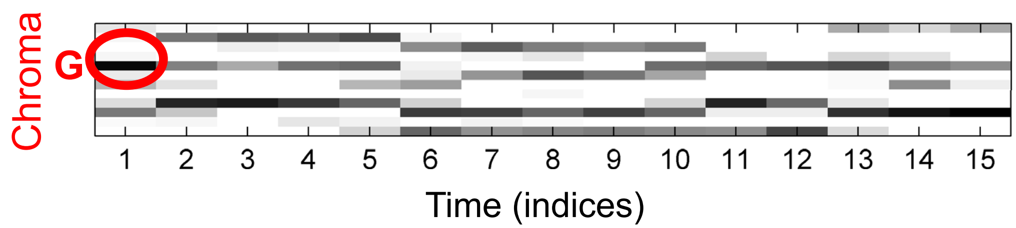
Beethoven’s Fifth

Karajan
(Orchester)



Time–chroma representations

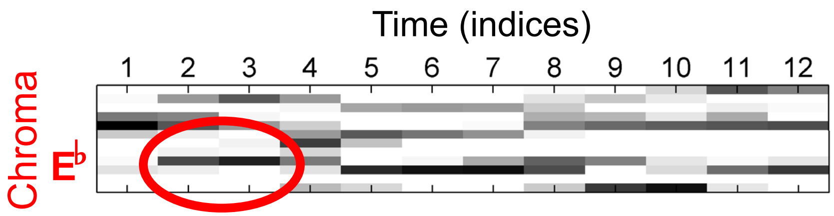
Gould
(Piano)



Motivation: Audio-Audio Alignment

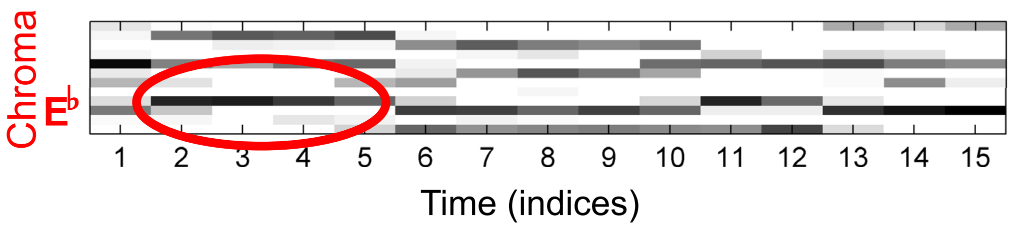
Beethoven's Fifth

Karajan (Orchester)



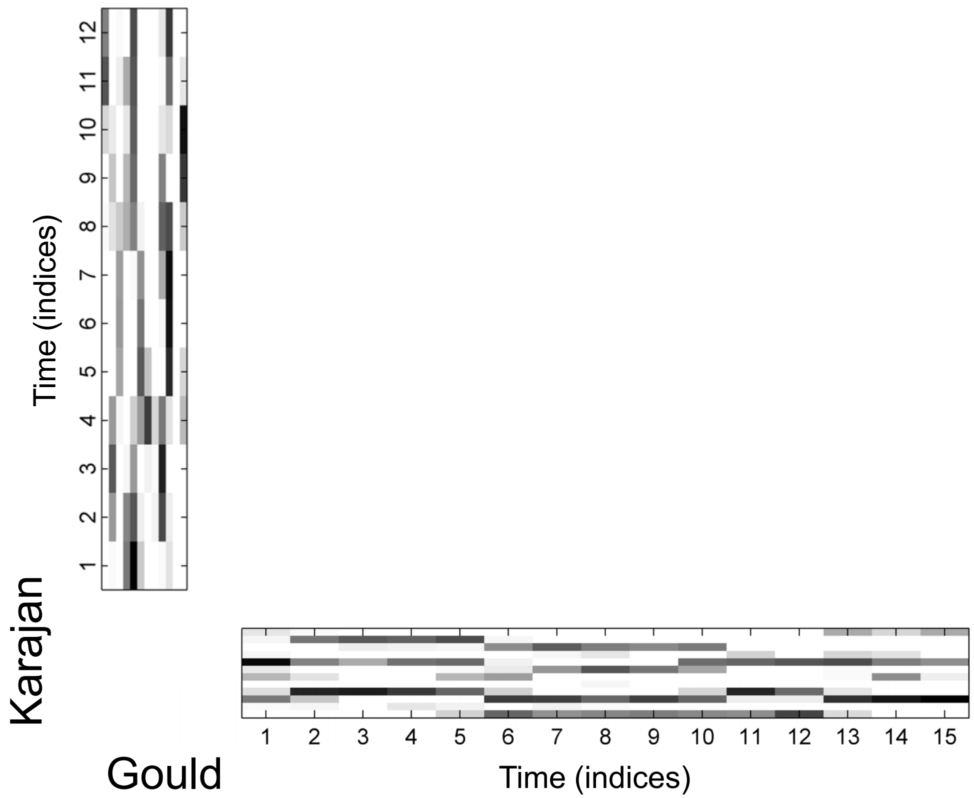
Time–chroma representations

Gould (Piano)



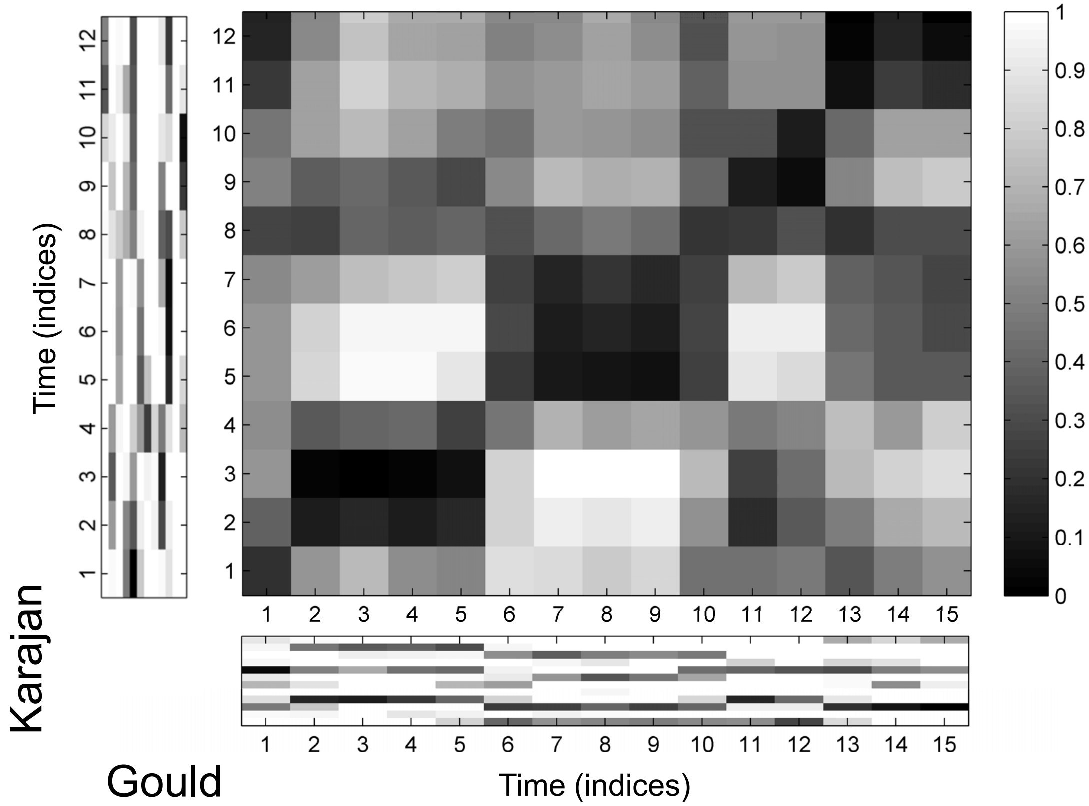
Motivation: Audio-Audio Alignment

Beethoven's Fifth



Motivation: Audio-Audio Alignment

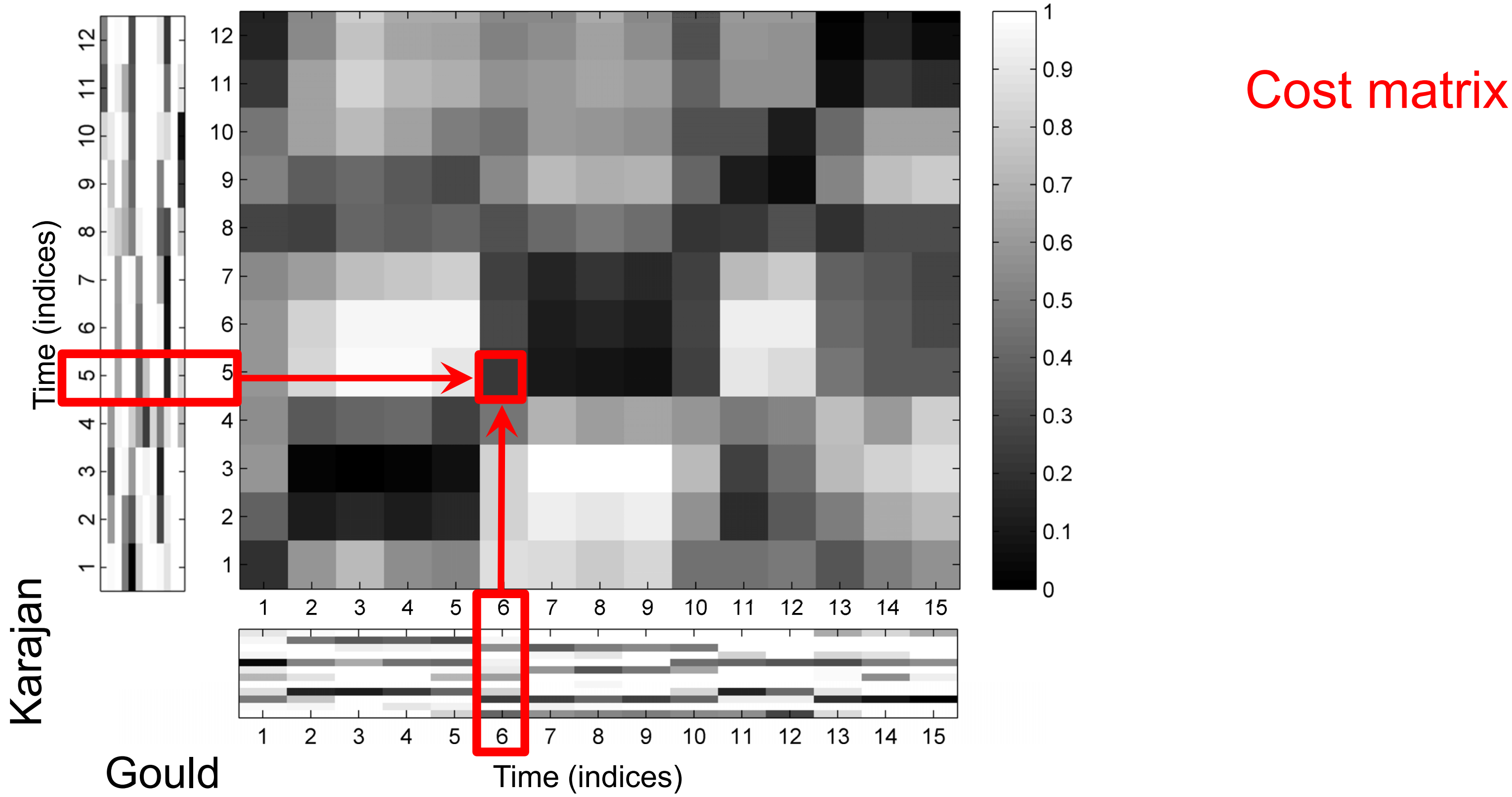
Beethoven’s Fifth



Cost matrix

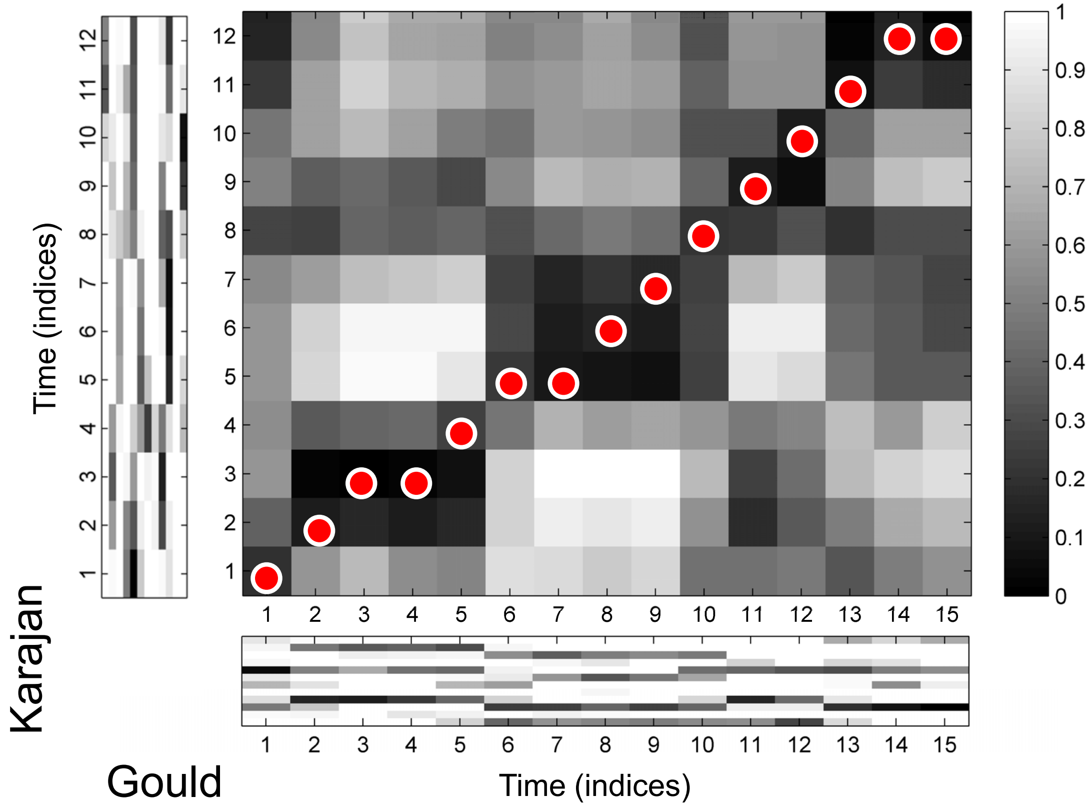
Motivation: Audio-Audio Alignment

Beethoven's Fifth



Motivation: Audio-Audio Alignment

Beethoven's Fifth

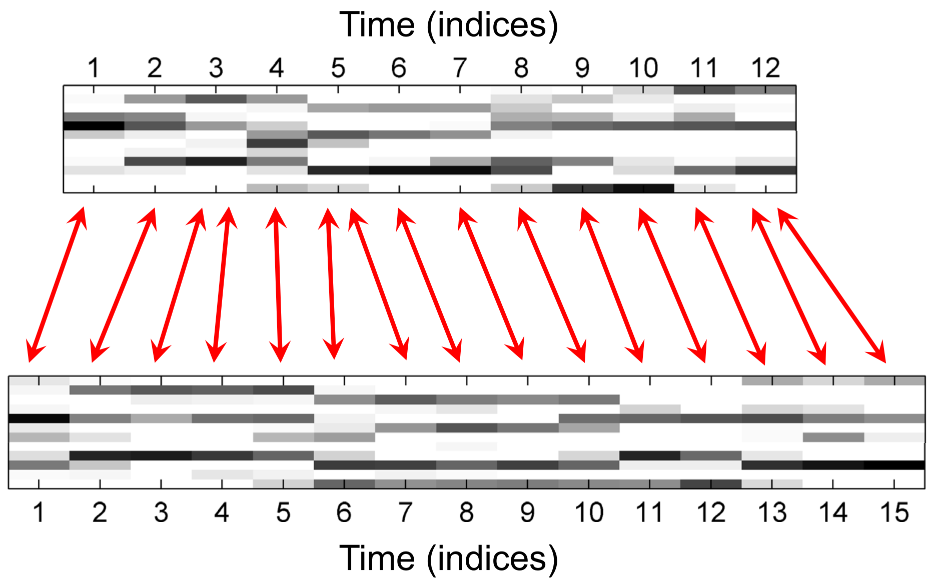


Cost-minimizing warping path

Motivation: Audio-Audio Alignment

Beethoven's Fifth

Karajan (Orchester)

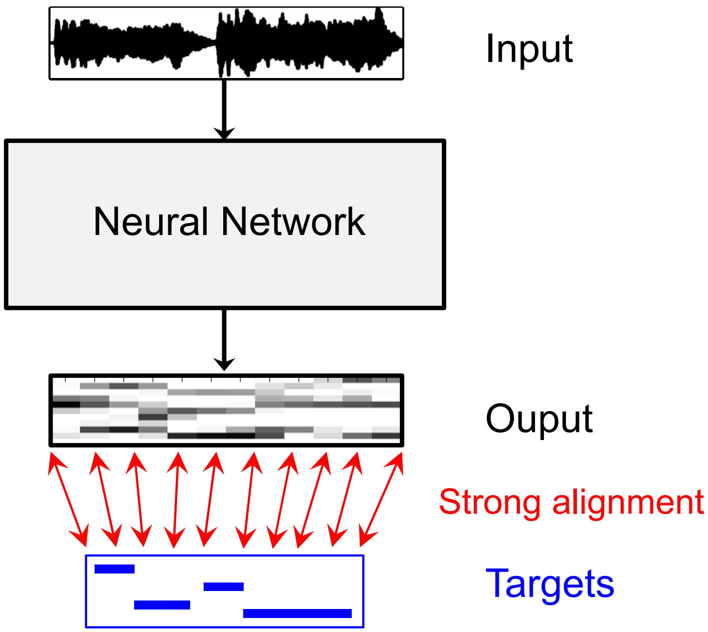


Gould (Piano)

Cost-minimizing warping path

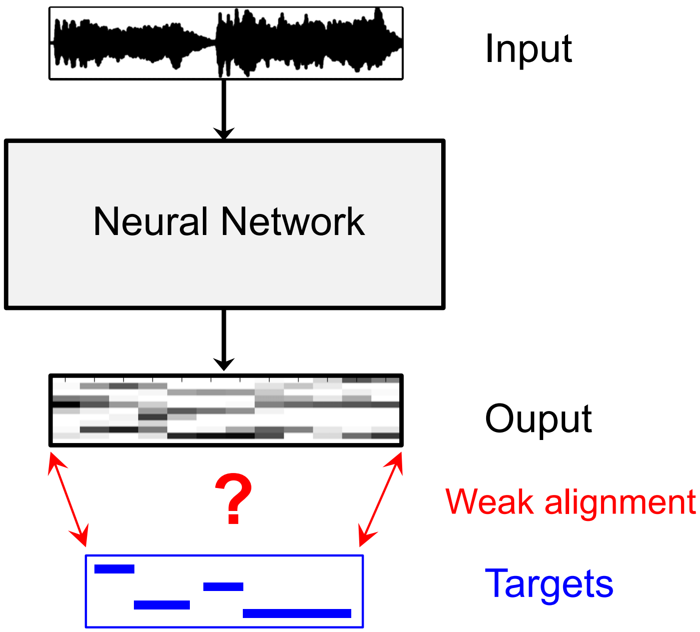
→ Strong alignment

Feature Learning



- Task: Learn audio features using a neural network
- Loss: Binary cross-entropy
  - framewise loss
  - requires strongly aligned targets
  - hard to obtain

Feature Learning



- Task: Learn audio features using a neural network
- Loss: Binary cross-entropy
  - framewise loss
  - requires strongly aligned targets
  - hard to obtain
- Alignment as part of loss function
  - requires only weakly aligned targets
  - needs to be differentiable
- Problem: DTW is not differentiable
  → Soft DTW

Dynamic Time Warping (DTW)

$X := (x_1, x_2, \ldots, x_N)$

$Y := (y_1, y_2, \ldots, y_M)$

$x_n, y_m \in \mathcal{F},\ n \in [1:N],\ m \in [1:M]$

$\mathcal{F}$ = Feature space

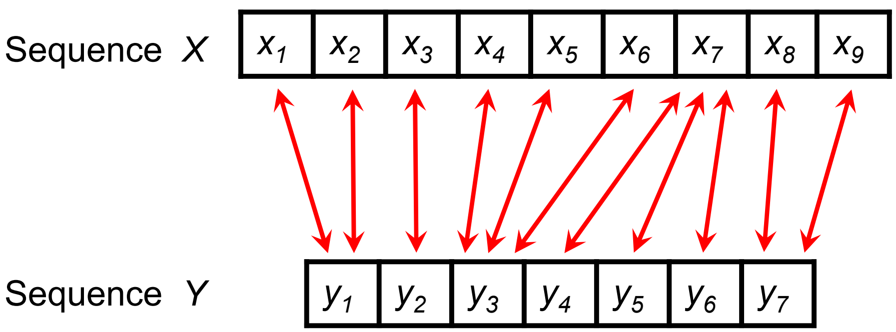
Alignment matrix

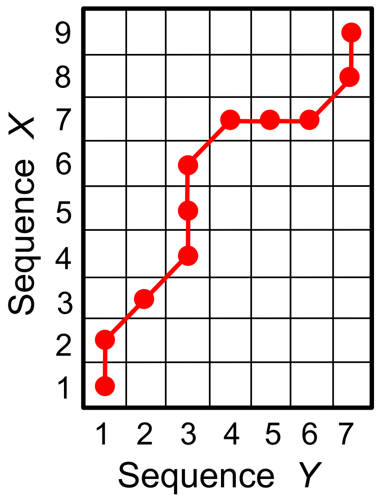
$A \in \{0,1\}^{N \times M}$

Set of all possible alignment matrices

$\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$

Alignment





Dynamic Time Warping (DTW)

$X := (x_1, x_2, \ldots, x_N)$

$Y := (y_1, y_2, \ldots, y_M)$

$x_n, y_m \in \mathcal{F},\ n \in [1:N],\ m \in [1:M]$

$\mathcal{F}$ = Feature space

Alignment matrix

$A \in \{0,1\}^{N \times M}$

Set of all possible alignment matrices

$\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$

Cost measure: $c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$

Cost matrix: $C \in \mathbb{R}^{N \times M}$ with $C(n,m) := c(x_n, y_m)$

Cost of alignment: $\langle A, C \rangle$

DTW cost: $\mathrm{DTW}(C) = \min\left(\{\langle A, C \rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

Optimal alignment: $A^* = \mathrm{argmin}\left(\{\langle A, C \rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

Dynamic Time Warping (DTW)

DTW cost: $$\mathrm{DTW}(C) = \min\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$$

- Efficient computation via Bellman's recursion in O(*NM*)

$$D(n,m) = \min\{D(n-1,m), D(n,m-1), D(n,m)\} + C(n,m)$$

for *n>1* and *m>1* and suitable initialization.

$$\mathrm{DTW}(C) = D(N,M)$$

- Problem: DTW(*C*) is not differentiable with regard to *C*
- Idea: Replace min-function by a smooth version

$$\min{}^{\gamma}(\mathcal{S}) = -\gamma \log \sum\nolimits_{s\in\mathcal{S}} \exp\left(-s/\gamma\right)$$

for set $\mathcal{S} \subset \mathbb{R}$ and temperature parameter $\gamma \in \mathbb{R}$

Soft Dynamic Time Warping (SDTW)

SDTW cost: $\mathrm{SDTW}^{\gamma}(C) = \min^{\gamma}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

- Efficient computation via Bellman's recursion in O(*NM*) still works:

  $$D^{\gamma}(n,m) = \min^{\gamma}\{D^{\gamma}(n{-}1,m), D^{\gamma}(n,m{-}1), D^{\gamma}(n,m)\}{+}C(n,m)$$

  for $n>1$ and $m>1$ and suitable initialization.

  $$\mathrm{SDTW}^{\gamma}(C) = D^{\gamma}(N,M)$$

- Limit case: $\mathrm{SDTW}^{\gamma}(C) \xrightarrow{\gamma\to 0} \mathrm{DTW}(C)$

- SDTW(*C*) is differentiable with regard to *C*

- Questions:
  - How does the gradient look like?
  - Can it be computed efficiently?
  - How does SDTW generalize the alignment concept?

Soft Dynamic Time Warping (SDTW)

Soft-DTW
Cuturi, Blondel: Soft-DTW: A Differentiable Loss Function for Time-Series. ICML, 2017

SDTW cost: $\mathrm{SDTW}^{\gamma}(C) = \min^{\gamma}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

- Define $p^{\gamma}(C)$ as the following "probability" distribution over $\mathcal{A}_{N,M}$:

$$p^{\gamma}(C)_A = \frac{\exp\left(-\langle A, C\rangle/\gamma\right)}{\sum_{A'\in\mathcal{A}_{N,M}} \exp\left(-\langle A', C\rangle/\gamma\right)} \qquad \text{for } A \in \mathcal{A}_{N,M}$$

- The expected alignment with respect to $p^{\gamma}(C)$ is given by:

$$E^{\gamma}(C) = \sum_{A\in\mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N\times M}$$

- The gradient is given by:

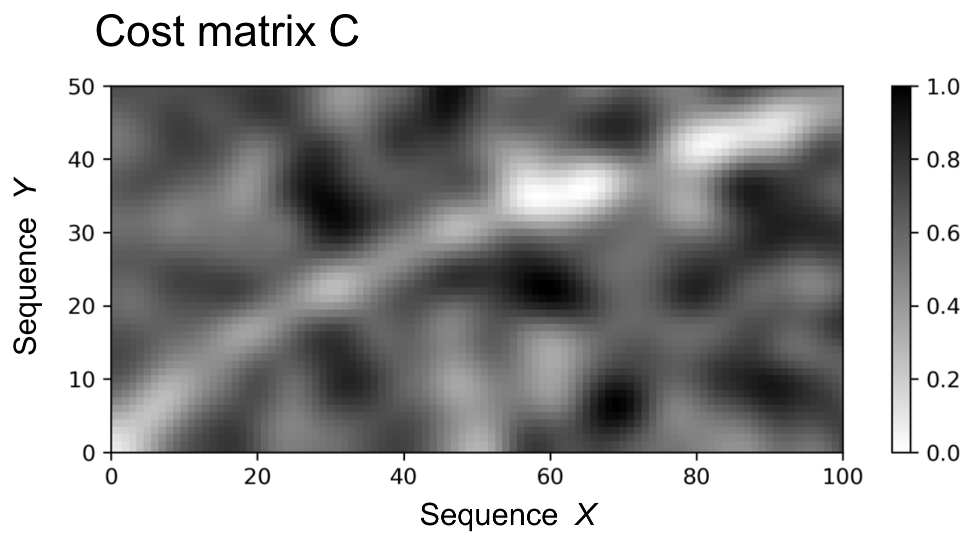
$$\nabla_C \mathrm{SDTW}^{\gamma}(C) = E^{\gamma}(C)$$

- The gradient can be computed efficiently in O(NM) via a recursive algorithm.

Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

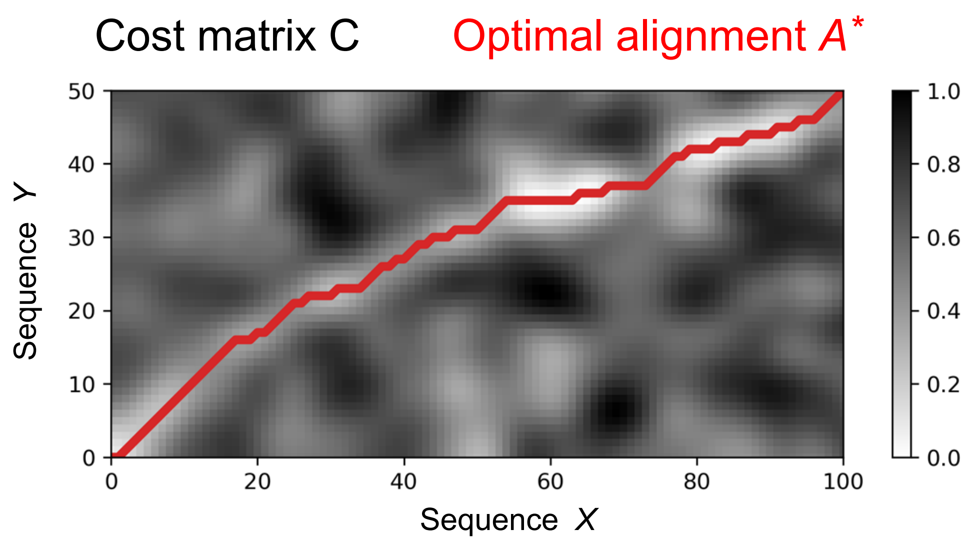
- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ



Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ

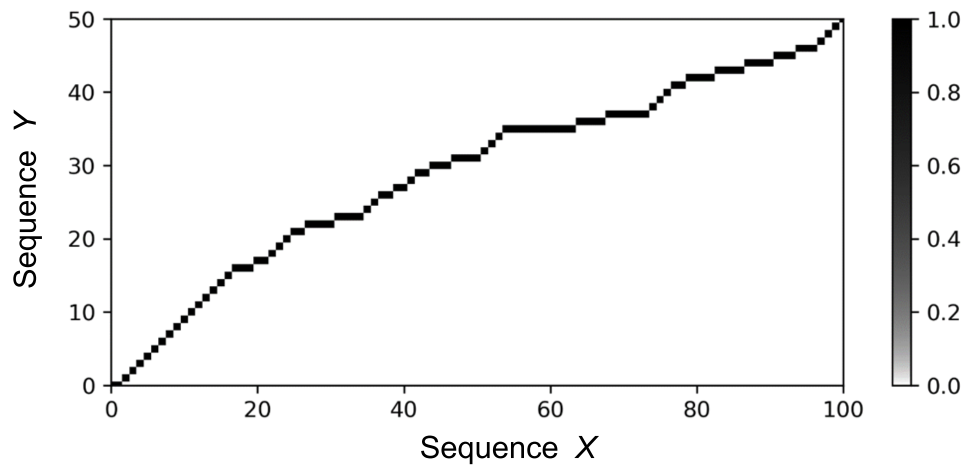


Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ

$E^{\gamma}(C)$ with $\gamma = 0$ $(= A^*)$

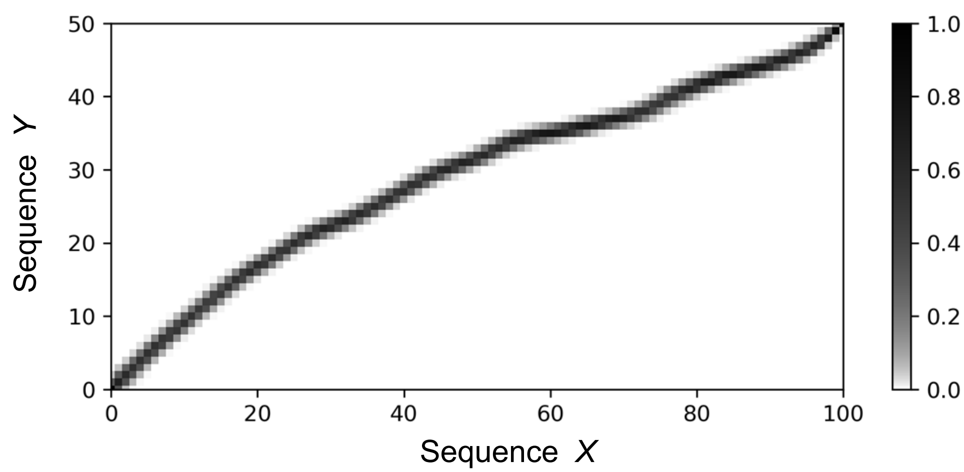


Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ

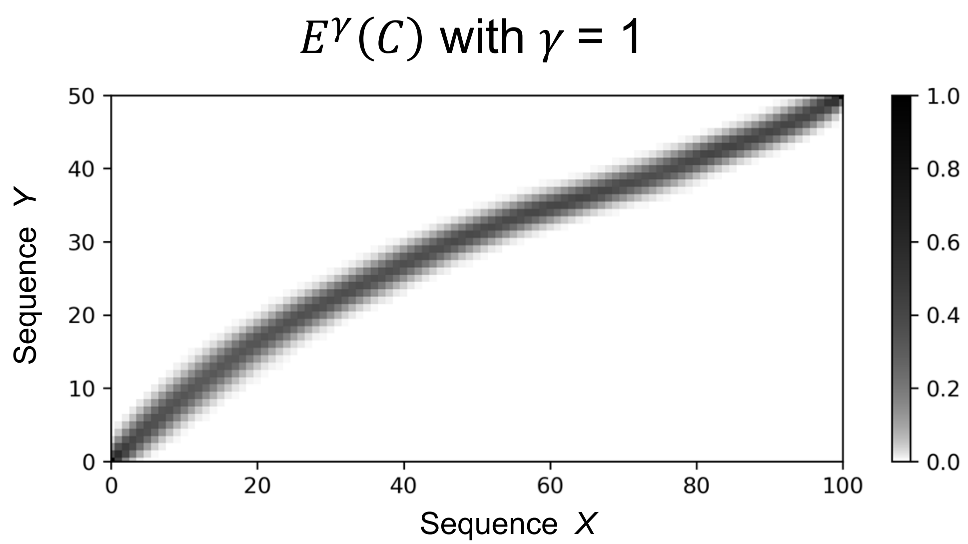
$E^{\gamma}(C)$ with γ = 0.1



Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

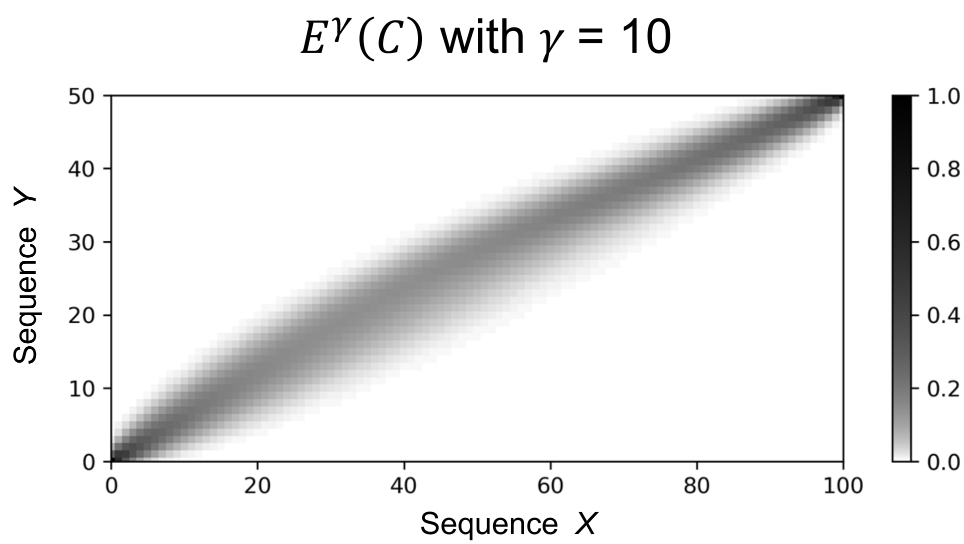
- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ



Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ

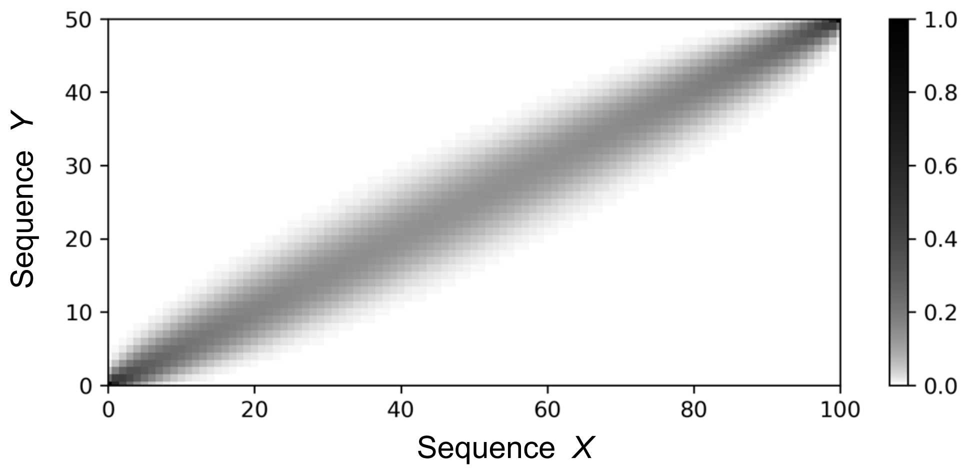


Soft Dynamic Time Warping (SDTW)

Expected alignment : $E^{\gamma}(C) = \sum_{A \in \mathcal{A}_{N,M}} p^{\gamma}(C)_A A \quad \in \mathbb{R}^{N \times M}$

- Can be interpreted as a smoothed version of an alignment
- Degree of smoothing depends on temperature parameter γ

$E^{\gamma}(C)$ with γ = 100

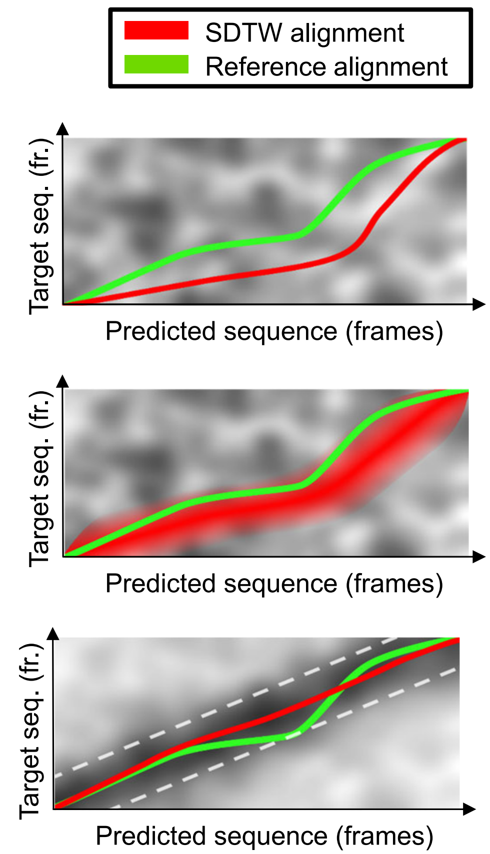


Soft Dynamic Time Warping (SDTW)
Conclusions

- Direct generalization of DTW (replacing min by smooth variant)
- Gradient is given by expected alignment
- Fast forward algorithm: O(NM)
- Fast gradient computation: O(NM)
- SDTW yields a (typically) poor lower bound for DTW
- Can be used as loss function to learn from weakly aligned sequences

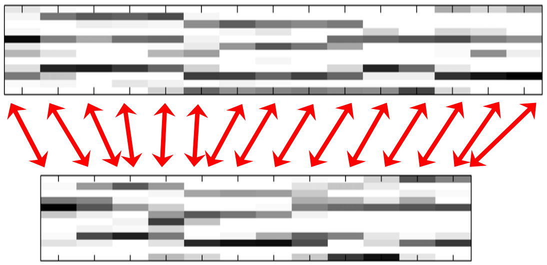
Soft Dynamic Time Warping (SDTW)
Stabilizing Training

- Standard SDTW often unstable
  - Unstable training in early stages
  - Degenerate output alignment
- Hyperparameter adjustment
  - High temperature to smooth alignments
  - Temperature annealing
- Diagonal prior
- Modified step size condition

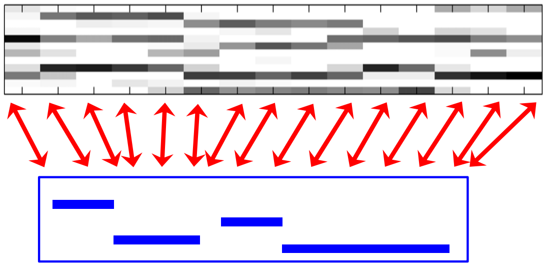


Soft Dynamic Time Warping (SDTW)
Representation Learning

- Symmetric application
  - Learn representation of both sequences
  - Needs a contrastive loss term



- Assymmetric application
  - Use fixed (e.g., binary) encoding of target
  - Learn representation of only one sequences
  - No contrastive loss term need



- Simulation of CTC-loss using SDTW possible
- Many DTW variants also possible for SDTW

Conclusions

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations
- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies
- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

Johannes Zeitler

Conclusions

- Multi-Scale Spectral Loss
  Knowledge Source: Signal Representations
- Hierarchical Classification Loss
  Knowledge Source: Musical Hierarchies
- Differentiable Alignment Loss
  Knowledge Source: Temporal Coherence

Simon Schwär

Michael Krause

Johannes Zeitler

Müller, Zeitler: **2025 ISMIR Tutorial**
Differentiable Alignment Techniques for Music Processing: Techniques and Applications