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

AUDIO LABS

Winter Term 2026/27

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

Lecture

Differentiable Alignment Techniques for Music Processing

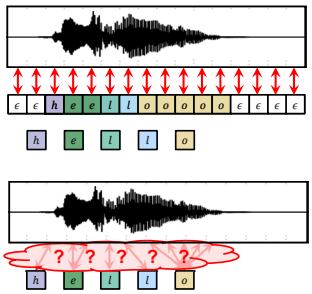
Stefan Balke, Andreas Brendel, Ben Maman, Meinard Müller

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

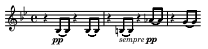
Fraunhofer IIS

Motivation: Learning with Weakly Aligned Labels
Training data in speech recognition

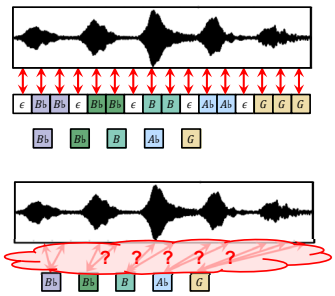
- Strongly aligned training data
  - Character annotations (labels) for each time step
  - Can be used for training in a standard classification setup
  - Tedious to annotate
- Weakly aligned training data
  - Globally corresponding character sequence without local alignment
  - Cannot be used for training in a standard classification setup
  - Easier to annotate
- Aim: Employ only weakly aligned data for training



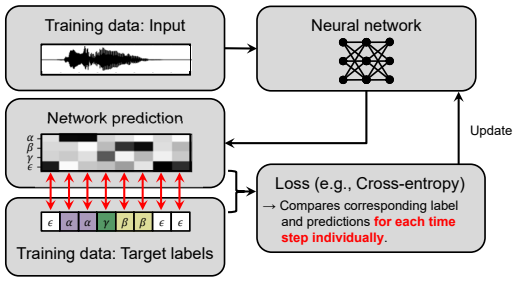
Motivation: Learning with Weakly Aligned Labels
Training data in theme-based music retrieval



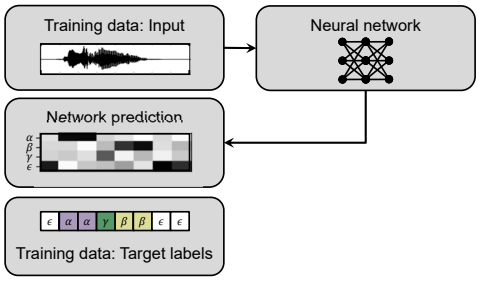
- Strongly aligned training data
  - Character annotations (labels) for each time step
  - Can be used for training in a standard classification setup
  - Tedious to annotate
- Weakly aligned training data
  - Globally corresponding character sequence without local alignment
  - Cannot be used for training in a standard classification setup
  - Easier to annotate
- Aim: Employ only weakly aligned data for training



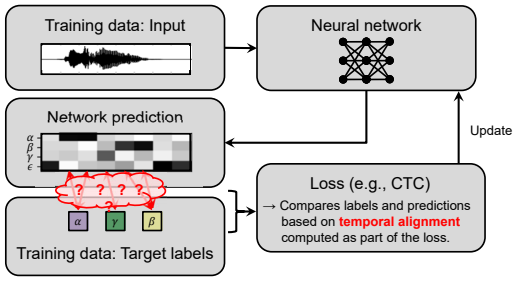
Motivation: Learning with Weakly Aligned Labels
Standard deep learning setup: Strongly aligned training data



Motivation: Learning with Weakly Aligned Labels
Standard deep learning setup: Strongly aligned training data



Motivation: Learning with Weakly Aligned Labels
Non-standard deep learning setup: Weakly aligned training data



Not a single alignment, but a suitably weighted combination of all valid alignments.

Overview
Differentiable Alignment Techniques for Music Processing

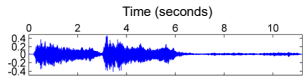
- **Motivation: Learning with Weakly Aligned Labels**
- Dynamic Time Warping (DTW)
- Differentiable DTW (Soft DTW)
- Applications

Overview
Differentiable Alignment Techniques for Music Processing

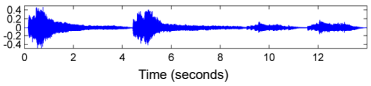
- Motivation: Learning with Weakly Aligned Labels
- **Dynamic Time Warping (DTW)**
- Differentiable DTW (Soft DTW)
- Applications

Motivation: Audio-Audio Alignment
Beethoven's Fifth

Karajan (Orchester)

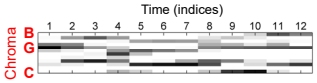


Gould (Piano)



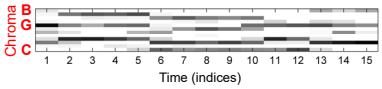
Motivation: Audio-Audio Alignment
Beethoven's Fifth

Karajan (Orchester)



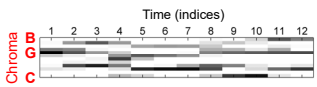
Time–chroma representations

Gould (Piano)



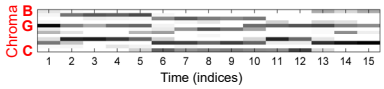
Motivation: Audio-Audio Alignment
Beethoven's Fifth

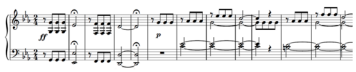
Karajan (Orchester)



Time–chroma representations

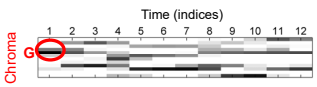
Gould (Piano)





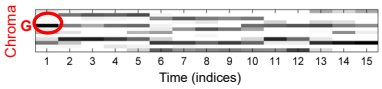
Motivation: Audio-Audio Alignment
Beethoven's Fifth

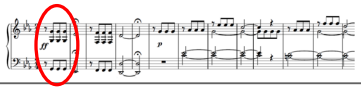
Karajan (Orchester)



Time–chroma representations

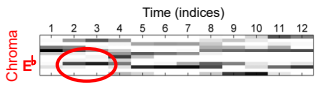
Gould (Piano)





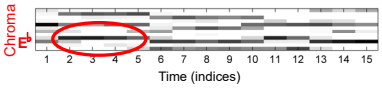
Motivation: Audio-Audio Alignment
Beethoven's Fifth

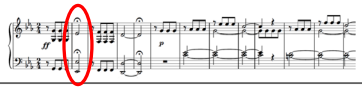
Karajan (Orchester)



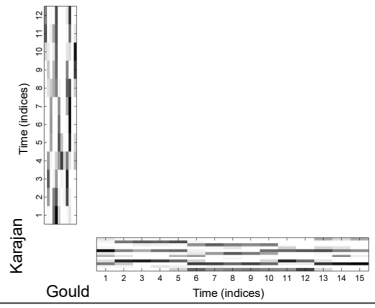
Time–chroma representations

Gould (Piano)

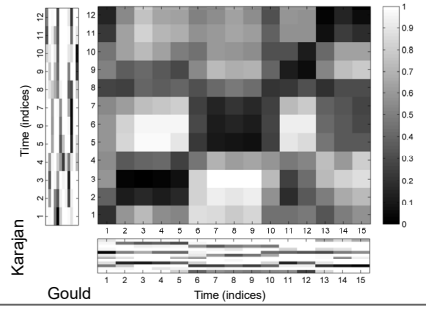




Motivation: Audio-Audio Alignment
Beethoven's Fifth

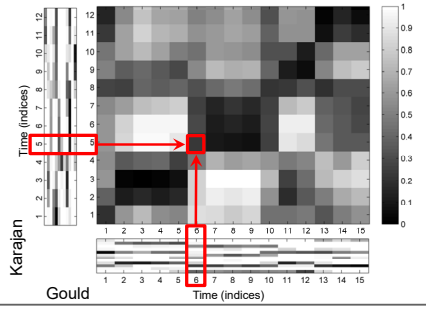


Motivation: Audio-Audio Alignment
Beethoven's Fifth



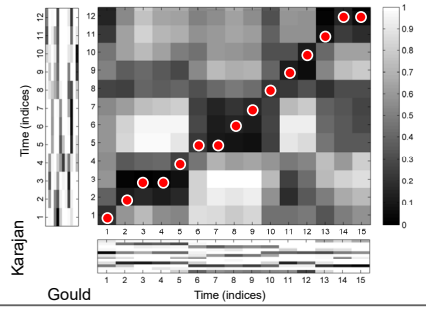
Cost matrix

Motivation: Audio-Audio Alignment
Beethoven's Fifth



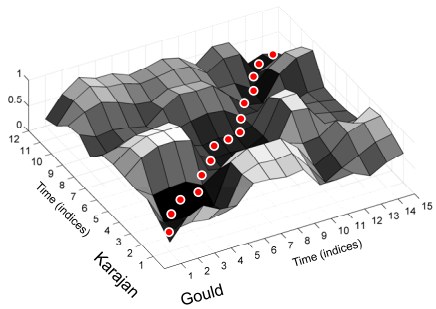
Cost matrix

Motivation: Audio-Audio Alignment
Beethoven's Fifth



Cost-minimizing warping path

Motivation: Audio-Audio Alignment
Beethoven's Fifth

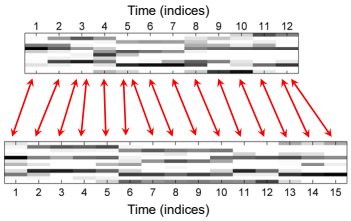


Cost-minimizing warping path

Motivation: Audio-Audio Alignment
Beethoven's Fifth

Karajan (Orchester)

Gould (Piano)



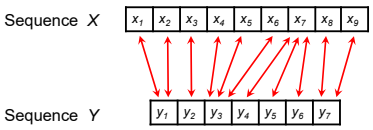
Cost-minimizing warping path

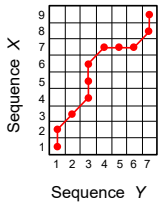
→ Strong alignment

Dynamic Time Warping

- Well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions.
- Intuitively, sequences are warped in a non-linear fashion to match each other.
- Originally used to compare different speech patterns in automatic speech recognition

Dynamic Time Warping





Time alignment of two time-dependent sequences, where the aligned points are indicated by the arrows.

Dynamic Time Warping

The objective of DTW is to compare two (time-dependent) sequences

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

of length $N \in \mathbb{N}$ and

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

of length $M \in \mathbb{N}$. Here,

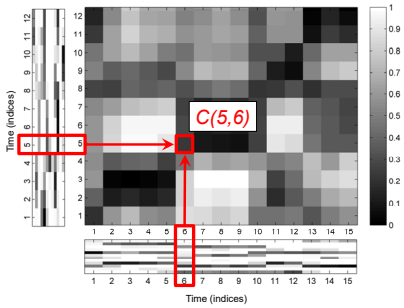
$$x_n, y_m \in \mathcal{F},\ n \in [1:N],\ m \in [1:M],$$

are features that are elements from a given feature $\mathcal{F}$.

Dynamic Time Warping
Cost Matrix

- Local cost measure $c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$ for comparing two features
- Typically, $c(x, y)$ is small (low cost) if $x, y \in \mathcal{F}$ are similar
- Define $C(n, m) := c(x_n, y_m)$ for $n \in [1:N],\ m \in [1:M]$
- Cost matrix $C \in \mathbb{R}^{N \times M}$

Dynamic Time Warping
Cost Matrix



Dynamic Time Warping
Warping Path

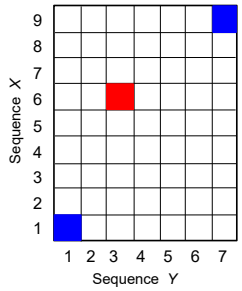
A **warping path** is a sequence $p = (p_1, \ldots, p_L)$ with

$$p_\ell = (n_\ell, m_\ell) \in [1:N] \times [1:M]$$

for $\ell \in [1:L]$ satisfying the following three conditions:

- Boundary condition: $p_1 = (1,1)$ and $p_L = (N,M)$
- Monotonicity condition: $n_1 \leq n_2 \leq \ldots \leq n_L$ and $m_1 \leq m_2 \leq \ldots \leq m_L$
- Step size condition: $p_{\ell+1} - p_\ell \in \{(1,0),(0,1),(1,1)\}$ for $\ell \in [1:L-1]$

Dynamic Time Warping
Warping Path



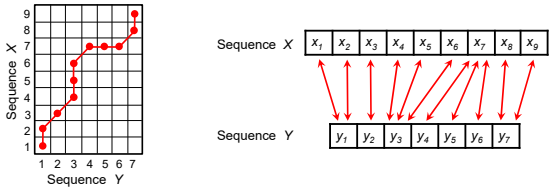
Each matrix entry (cell) corresponds to a pair of indices.

Cell = (6,3)

Boundary cells:
p_1 = (1,1)
p_L = (N,M) = (9,7)

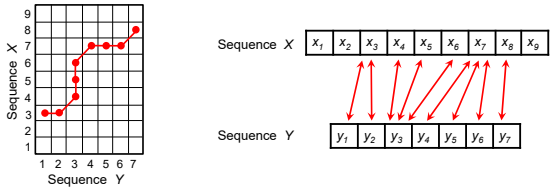
Dynamic Time Warping
Warping Path

Correct warping path



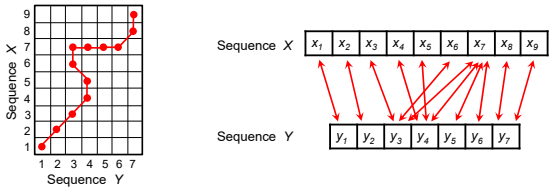
Dynamic Time Warping
Warping Path

Violation of boundary condition



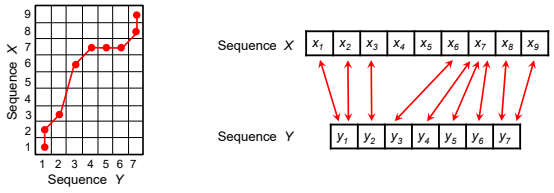
Dynamic Time Warping
Warping Path

Violation of monotonicity condition



Dynamic Time Warping
Warping Path

Violation of step size condition



Dynamic Time Warping
Warping Path

- The total cost of a warping path p between X and Y with respect to the local cost measure c is defined as $c_p(X,Y) := \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell})$
- An optimal warping path between X and Y is a warping path p^* having minimal total cost among all possible warping paths.
- The DTW distance between X and Y is defined as the total cost of p^*:

$$\mathrm{DTW}(X,Y) := c_{p^*}(X,Y) = \min\{c_p(X,Y) \mid p \text{ is a warping path}\}$$

Dynamic Time Warping

- There are exponentially many warping paths. How many?
- The optimal warping path p^*need not to be unique. Give an example?
- DTW does generally not satisfy the triangle inequality.
- How can $\mathrm{DTW}(X,Y)$ and p^* be computed efficiently?

Dynamic Time Warping
Dynamic Programming: Bellman Recursion

- Prefix Notation:

$$\begin{aligned} X(1:n) &:= (x_1,\ldots,x_n), \quad 1 \le n \le N \\ Y(1:m) &:= (y_1,\ldots,y_m), \quad 1 \le m \le M \\ D(n,m) &:= \mathrm{DTW}(X(1:n), Y(1:m)) \end{aligned}$$

- The $N \times M$ matrix D is called the accumulated cost matrix.
- The entry $D(n,m)$ specifies the cost of an optimal warping path that aligns the prefix sequences $X(1:n)$ with $Y(1:m)$.

Dynamic Time Warping
Dynamic Programming: Bellman Recursion

Initialization:

$$\begin{aligned} D(1,1) &= C(1,1) \\ D(n,1) &= \textstyle\sum_{k=1}^{n} C(k,1) \quad \text{for} \quad 1 \le n \le N \quad \text{“First column”} \\ D(1,m) &= \textstyle\sum_{k=1}^{m} C(1,k) \quad \text{for} \quad 1 \le m \le M \quad \text{“First row”} \end{aligned}$$

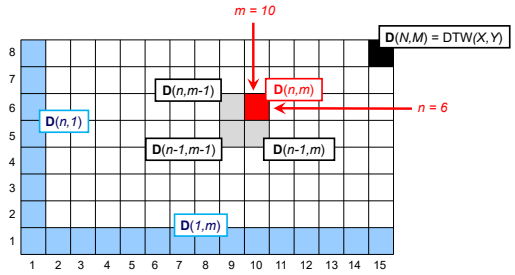
Recursions:

$$D(n,m) = \min \begin{pmatrix} D(n-1,m-1), \\ D(n-1,m), \\ D(n,m-1) \end{pmatrix} + C(n,m) \quad \text{for} \quad n > 1,\ m > 1$$

Time and memory complexity: $O(NM)$

Result: $D(N,M) = \mathrm{DTW}(X,Y)$

Dynamic Time Warping
Dynamic Programming: Bellman Recursion



Dynamic Time Warping
Dynamic Programming: Bellman Recursion

- An optimal warping path $p^* = (p_1,\ldots,p_L)$ is obtained via **backtracking**.
- Start with last path element $p_L = (N,M)$
- Compute in revers order:

$$p_{\ell-1} := \begin{cases} (1,m-1), & \text{if } n = 1 \\ (n-1,1), & \text{if } m = 1 \\ \mathrm{argmin}\{D(n-1,m-1), & \\ \quad D(n-1,m), D(n,m-1)\}, & \text{otherwise,} \end{cases}$$

- In case $(n,m) = (1,1)$, one must have $\ell = 1$.
- Take the lexicographically smallest pair in case “argmin” is not unique.

Dynamic Time Warping
Dynamic Programming: Bellman Recursion

Algorithm: DTW

Input: Cost matrix $\mathbf{C}$ of size $N \times M$
Output: Accumulated cost matrix $\mathbf{D}$
Optimal warping path P^*

Procedure: Initialize $(N \times M)$ matrix $\mathbf{D}$ by $\mathbf{D}(n,1) = \sum_{k=1}^{n} \mathbf{C}(k,1)$ for $n \in [1:N]$ and $\mathbf{D}(1,m) = \sum_{k=1}^{m} \mathbf{C}(1,k)$ for $m \in [1:M]$. Then compute in a nested loop for $n = 2,\ldots,N$ and $m = 2,\ldots,M$:

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

Set $\ell = 1$ and $q_\ell = (N,M)$. Then repeat the following steps until $q_\ell = (1,1)$:

Increase ℓ by one and let $(n,m) = q_{\ell-1}$.
If $n = 1$, then $q_\ell = (1,m-1)$,
else if $m = 1$, then $q_\ell = (n-1,m)$,
else $q_\ell = \operatorname{argmin}\{\mathbf{D}(n-1,m-1), \mathbf{D}(n-1,m), \mathbf{D}(n,m-1)\}$.
(If 'argmin' is not unique, take lexicographically smallest cell.)

Set $L = \ell$ and return $P^* = (q_L, q_{L-1}, \ldots, q_1)$ as well as $\mathbf{D}$.

Dynamic Time Warping
Dynamic Programming: Bellman Recursion

$X = (1,3,3,8,1) \quad Y = (2,0,0,8,7,2)$
$c(x,y) = |x-y|, \; x,y \in \mathbb{R}$

C

1	1	1	1	7	6	1
8	6	8	8	0	1	6
3	1	3	3	5	4	1
3	1	3	3	5	4	1
1	1	1	1	7	6	1
	2	0	0	8	7	2

D

1	10	10	11	14	13	9
8	9	11	13	7	8	14
3	3	5	7	10	12	13
3	2	4	5	8	12	13
1	1	2	3	10	16	17
	2	0	0	8	7	2

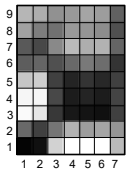
Alignment

X	1	3	3	8	1	
Y	2	0	0	8	7	2

Optimal warping path: $p^* = ((1,1),(2,2),(3,3),(4,4),(4,5),(5,6))$

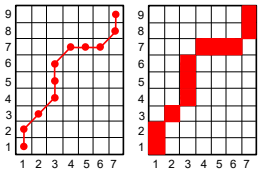
Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$



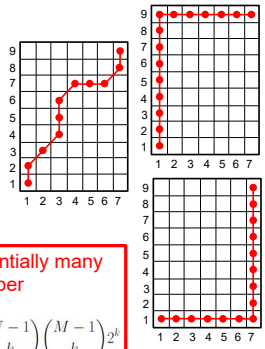
Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$



Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$



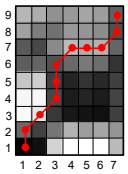
$M \backslash N$	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	3	5	7	9	11	13
3	1	5	13	25	41	61	85
4	1	7	25	63	129	231	377
5	1	9	41	129	321	681	1289
6	1	11	61	231	681	1683	3653
7	1	13	85	377	1289	3653	8989

There are exponentially many
→ Delannoy number

$$|\mathcal{A}_{N,M}| = \sum_{k=0}^{\min(N-1,M-1)} \binom{N-1}{k}\binom{M-1}{k} 2^k$$

Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- Cost of alignment: $\langle A, C \rangle$



Frobenius inner product

$$\langle \mathbf{A}, \mathbf{C} \rangle = \sum_{n=1}^{N} \sum_{m=1}^{M} a_{n,m} c_{n,m}$$

Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- Cost of alignment: $\langle A, C \rangle$
- DTW cost: $\mathrm{DTW}(C) = \min\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- 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: $A^* = \operatorname{argmin}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$

Dynamic Time Warping

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- 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: $A^* = \operatorname{argmin}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$
- Dynamic programming: $\mathrm{DTW}(C) = D(N, M)$

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

Dynamic Time Warping

Local – Global
Global definition of DTW(C) vs. local compuation of DTW(C)

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- Cost of alignment: $\langle A, C \rangle$
- DTW cost: $\mathrm{DTW}(C) = \min\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$ — **Global**
- Optimal alignment: $A^* = \operatorname{argmin}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$
- Dynamic programming: $\mathrm{DTW}(C) = D(N, M)$

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

Dynamic Time Warping

Local – Global
Global definition of DTW(C) vs. local compuation of DTW(C)

Forward – Backward
Forward DTW computation and backward optimal-alignment backtracking

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- 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: $A^* = \operatorname{argmin}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$ — **Backward**
- Dynamic programming: $\mathrm{DTW}(C) = D(N, M)$

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

Dynamic Time Warping

Local – Global
Global definition of DTW(C) vs. local compuation of DTW(C)

Forward – Backward
Forward DTW computation and backward optimal-alignment backtracking

Differentiability
The min and argmin operations make DTW and A^* computation nondifferentiable

- Cost matrix: $C \in \mathbb{R}^{N \times M}$
- Alignment matrix: $A \in \{0,1\}^{N \times M}$
- Set of all alignment matrices: $\mathcal{A}_{N,M} \subset \{0,1\}^{N \times M}$
- 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: $A^* = \operatorname{argmin}\left(\{\langle A, C\rangle \mid A \in \mathcal{A}_{N,M}\}\right)$
- Dynamic programming: $\mathrm{DTW}(C) = D(N, M)$

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

Overview
Differentiable Alignment Techniques for Music Processing

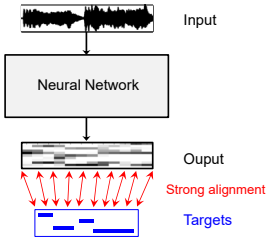
- Motivation: Learning with Weakly Aligned Labels
- Dynamic Time Warping (DTW)
- **Differentiable DTW (Soft DTW)**
- Applications

Special Thanks



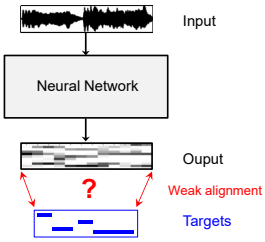
Johannes Zeitler (PhD 2026)

Motivation: DTW Loss
Feature Learning



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

Motivation: DTW Loss
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

Motivation: DTW Loss

- **DTW loss:** Gradients required for neural-network training
- **Hard minimum:** Discontinuous gradients, unstable optimization
- **Sparse gradients:** Only one optimal path; alternatives ignored
- **Smoothing approach:** Replace the hard minimum with a soft minimum
- **Main idea:** Replace hard decision by weighted combination of alternatives
- **Foundation:** Differentiable soft-minimum function

From Hard Minimum to Soft Minimum

From Hard Minimum to Soft Minimum

- Input vector:
  $\mathbf{a} = (a_1, \ldots, a_K)^\top \in \mathbb{R}^K$
- Min function:
  $\min(\mathbf{a}) = \min(\{a_1, \ldots, a_K\})$
- One-hot-encoded argmin for a unique minimum at k^*:
  $\operatorname{argmin}(\mathbf{a}) \in \{0,1\}^K \subset \mathbb{R}^K$

  $$[\operatorname{argmin}(\mathbf{a})]_k = \begin{cases} 1, & k = k^* \\ 0, & k \neq k^* \end{cases}$$

- Softmin function:
  $\min_{\text{soft}}(\mathbf{a}) = -\log\left(\sum_{k=1}^{K} \exp(-a_k)\right)$
- Gradient of softmin function:

  $\nabla \min_{\text{soft}}(\mathbf{a}) \in \mathbb{R}^K$

  $$[\nabla \min_{\text{soft}}(\mathbf{a})]_k = \frac{\exp(-a_k)}{\sum_{j=1}^{K} \exp(-a_j)}$$

From Hard Minimum to Soft Minimum

- Input vector:
  $\mathbf{a} = (a_1, \ldots, a_K)^\top \in \mathbb{R}^K$
- Min function:
  $\min(\mathbf{a}) = \min(\{a_1, \ldots, a_K\})$
- One-hot-encoded argmin for a unique minimum at k^*:
  $\operatorname{argmin}(\mathbf{a}) \in \{0,1\}^K \subset \mathbb{R}^K$

  $$[\operatorname{argmin}(\mathbf{a})]_k = \begin{cases} 1, & k = k^* \\ 0, & k \neq k^* \end{cases}$$

- Temperature parameter
  $\gamma > 0$
- Softmin function:
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = -\gamma \log\left(\sum_{k=1}^K \exp\left(-\frac{a_k}{\gamma}\right)\right)$
- Gradient of softmin function:

  $\nabla \min^\gamma_{\text{soft}}(\mathbf{a}) \in \mathbb{R}^K$

  $$[\nabla \min^\gamma_{\text{soft}}(\mathbf{a})]_k = \frac{\exp(-a_k/\gamma)}{\sum_{j=1}^K \exp(-a_j/\gamma)}$$

From Hard Minimum to Soft Minimum

- γ controls degree of smoothing
- Small γ: Closer to hard minimum
  $\lim_{\gamma \to 0^+} \min^\gamma_{\text{soft}}(\mathbf{a}) = \min(\mathbf{a})$
- Large γ: Closer to average
  $\lim_{\gamma \to \infty} \min^\gamma_{\text{soft}}(\mathbf{a}) = -\infty$
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = \frac{1}{K}\sum_{k=1}^K a_k - \gamma \log K + \mathcal{O}\left(\frac{1}{\gamma}\right)$

- Temperature parameter
  $\gamma > 0$
- Softmin function:
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = -\gamma \log\left(\sum_{k=1}^K \exp\left(-\frac{a_k}{\gamma}\right)\right)$
- Gradient of softmin function:

  $\nabla \min^\gamma_{\text{soft}}(\mathbf{a}) \in \mathbb{R}^K$

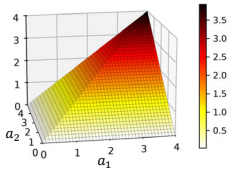
  $$[\nabla \min^\gamma_{\text{soft}}(\mathbf{a})]_k = \frac{\exp(-a_k/\gamma)}{\sum_{j=1}^K \exp(-a_j/\gamma)}$$

From Hard Minimum to Soft Minimum

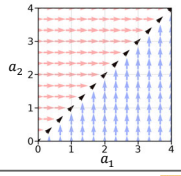
$\gamma = 0$

- γ controls degree of smoothing
- Small γ: Closer to hard minimum
  $\lim_{\gamma \to 0^+} \min^\gamma_{\text{soft}}(\mathbf{a}) = \min(\mathbf{a})$
- Large γ: Closer to average
  $\lim_{\gamma \to \infty} \min^\gamma_{\text{soft}}(\mathbf{a}) = -\infty$
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = \frac{1}{K}\sum_{k=1}^K a_k - \gamma \log K + \mathcal{O}\left(\frac{1}{\gamma}\right)$

$\min^\gamma_{\text{soft}}(\mathbf{a}) = \min(\mathbf{a})$



$\nabla \min^\gamma_{\text{soft}}(\mathbf{a}) = \operatorname{argmin}(\mathbf{a})$

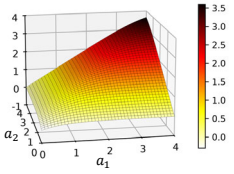


From Hard Minimum to Soft Minimum

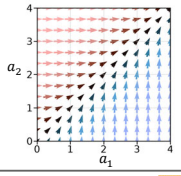
$\gamma = 0.5$

- γ controls degree of smoothing
- Small γ: Closer to hard minimum
  $\lim_{\gamma \to 0^+} \min^\gamma_{\text{soft}}(\mathbf{a}) = \min(\mathbf{a})$
- Large γ: Closer to average
  $\lim_{\gamma \to \infty} \min^\gamma_{\text{soft}}(\mathbf{a}) = -\infty$
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = \frac{1}{K}\sum_{k=1}^K a_k - \gamma \log K + \mathcal{O}\left(\frac{1}{\gamma}\right)$

$\min^\gamma_{\text{soft}}(\mathbf{a})$



$\nabla \min^\gamma_{\text{soft}}(\mathbf{a})$

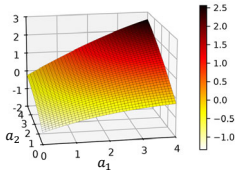


From Hard Minimum to Soft Minimum

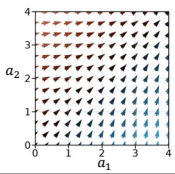
$\gamma = 2$

- γ controls degree of smoothing
- Small γ: Closer to hard minimum
  $\lim_{\gamma \to 0^+} \min^\gamma_{\text{soft}}(\mathbf{a}) = \min(\mathbf{a})$
- Large γ: Closer to average
  $\lim_{\gamma \to \infty} \min^\gamma_{\text{soft}}(\mathbf{a}) = -\infty$
  $\min^\gamma_{\text{soft}}(\mathbf{a}) = \frac{1}{K}\sum_{k=1}^K a_k - \gamma \log K + \mathcal{O}\left(\frac{1}{\gamma}\right)$

$\min^\gamma_{\text{soft}}(\mathbf{a})$



$\nabla \min^\gamma_{\text{soft}}(\mathbf{a})$



Soft Dynamic Time Warping (SDTW)

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

- 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.

  $\operatorname{DTW}(C) = D(N,M)$ **Local**
- Problem: DTW(C) is not differentiable with regard to *C*
- Idea: Replace min-function by a smooth version

  $$\min^\gamma(\mathcal{S}) = -\gamma \log \sum_{s \in \mathcal{S}} \exp(-s/\gamma)$$

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

Soft Dynamic Time Warping (SDTW)

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

Global

- 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.

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

Local

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

Differentiability

- 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)

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

Forward

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

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

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

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

- The gradient is given by:

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

Backward

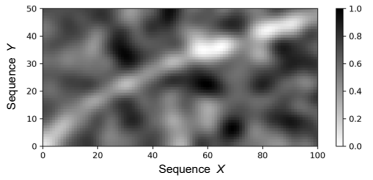
- The gradient can be computed efficiently in O(*NM*) via a recursive algorithm.

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

Cost matrix C

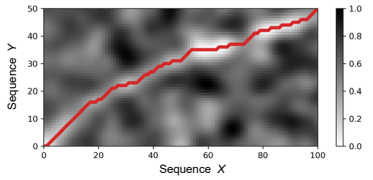


Soft Dynamic Time Warping (SDTW)

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

Cost matrix C Optimal alignment A^*

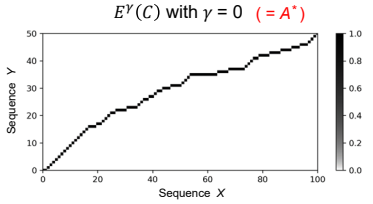


Soft Dynamic Time Warping (SDTW)

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 $\gamma = 0$ ($= A^*$)

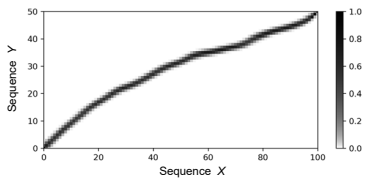


Soft Dynamic Time Warping (SDTW)

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 $\gamma = 0.1$

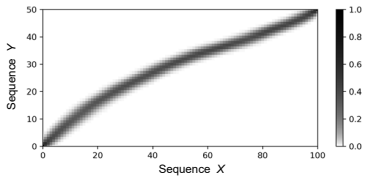


Soft Dynamic Time Warping (SDTW)

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

$E^{\gamma}(C)$ with γ = 1

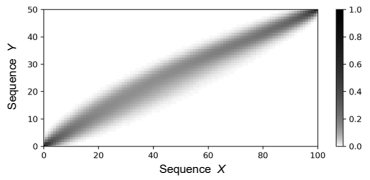


Soft Dynamic Time Warping (SDTW)

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

$E^{\gamma}(C)$ with γ = 10

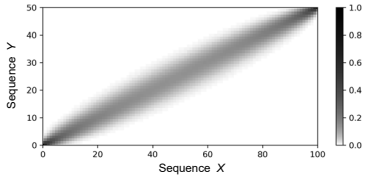


Soft Dynamic Time Warping (SDTW)

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

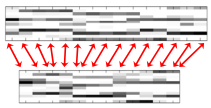
Differentiability

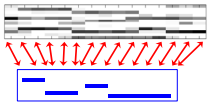
Forward

Backward

Soft Dynamic Time Warping (SDTW)
Conclusions

- 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
- SDTW can be generalized in may ways
  - Weight, step size & boundary conditions
  - Simulation of CTC-loss using SDTW possible



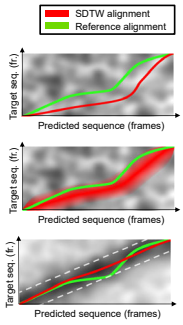


Graves, Fernández, Gomez, Schmidhuber: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. ICML, 2006.

Zeitler, Müller: A Unified Perspective on CTC and Soft-DTW Using Differentiable DTW. IEEE TASLPRO, 2026.

Soft Dynamic Time Warping (SDTW)
Conclusions: 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



Overview
Differentiable Alignment Techniques for Music Processing

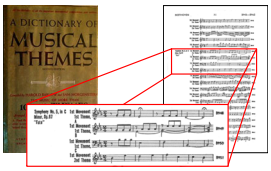
- Motivation: Learning with Weakly Aligned Labels
- Dynamic Time Warping (DTW)
- Differentiable DTW (Soft DTW)
- **Applications**

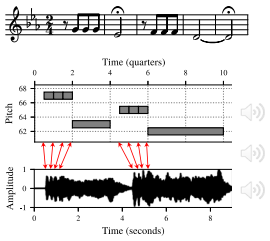
Special Thanks



Frank Zalkow (PhD 2021)

Theme-Based Audio Retrieval
Barlow & Morgenstern (1949): A Dictionary of Musical Themes

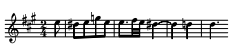




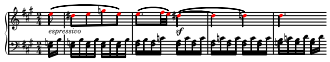
- 2067 themes by 54 different composers
- Recordings (1126 recordings, ~ 120 hours)
- Theme occurences (~ 5 hours)

Theme-Based Audio Retrieval
Monophony–Polyphony Challenge

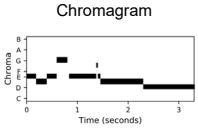
Monophonic symbolic musical theme

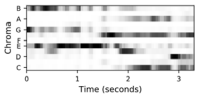


Audio recording of polyphonic music



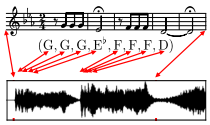
Chromagram





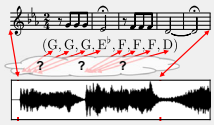
Goal: Compute “enhanced” chromagram from polyphonic audio recording that better matches the symbolic monophonic theme

Theme-Based Audio Retrieval
Strongly Aligned Training Data



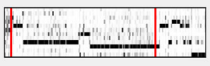
Theme-Based Audio Retrieval
Weakly Aligned Training Data

Network input for training:
Weakly aligned score–audio pairs

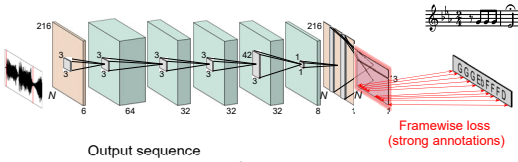


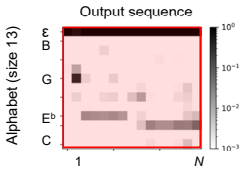
Convolutional neural network:
Train with the differentiable alignment loss

Post-processed output of trained network:
Enhanced chroma representation

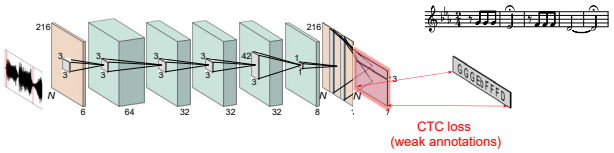


Theme-Based Audio Retrieval





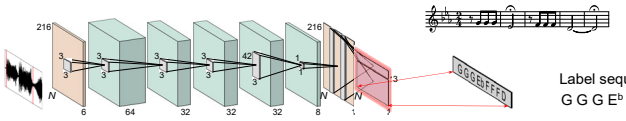
Theme-Based Audio Retrieval



- Idea of CTC loss similar to SDTW
- Theme is given as label sequence over finite alphabet (size 13 including blank symbol)
- Expand label sequence to match audio feature sequence
  → valid alignment
- CTC loss considers probability over **all** valid alignments
  → differentiable

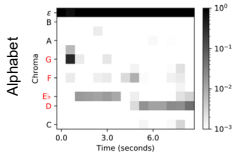
Graves, Fernández, Gomez, Schmidhuber: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. ICML, 2006.

Theme-Based Audio Retrieval
CTC-Based Training

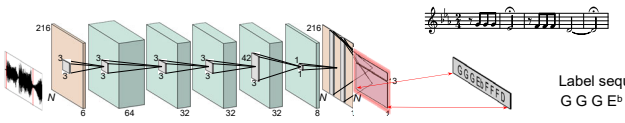


Label sequence Y
G G G E$^{\flat}$ F F F D

Output sequence X

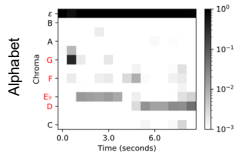


Theme-Based Audio Retrieval
CTC-Based Training



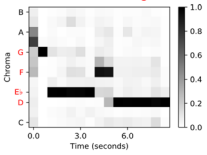
Label sequence Y
G G G E$^{\flat}$ F F F D

Output sequence X

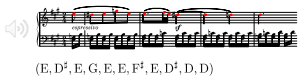


Post processing

Final Chromagram



Theme-Based Audio Retrieval
Evaluation Results



$(\mathrm{E}, \mathrm{D}^{\sharp}, \mathrm{E}, \mathrm{G}, \mathrm{E}, \mathrm{E}, \mathrm{F}^{\sharp}, \mathrm{E}, \mathrm{D}^{\sharp}, \mathrm{D}, \mathrm{D})$

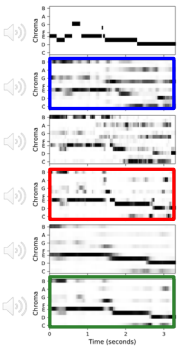
Chroma Variant	Top-1	Top-10
Standard chromagram	0.561	0.723
Enhanced chromagram (baseline)	0.824	0.861
DNN-based chromagram (CTC)	0.867	0.942
DNN-based chromagram (linear scaling)	0.829	0.914
DNN-based chromagram (strong alignment)	0.882	0.939

Target

Traditional: Lower bound

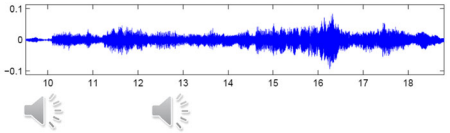
Weak case

Strong case: Upper bound



Lyrics–Audio Alignment

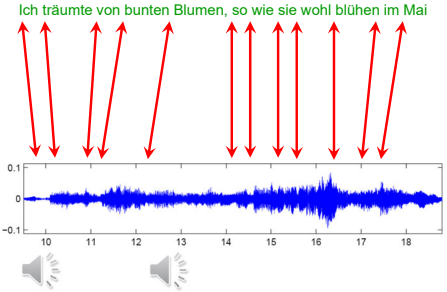
Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai



Stoller, Durand, Ewert: End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. Proc. ICASSP 2019

Lyrics–Audio Alignment

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai

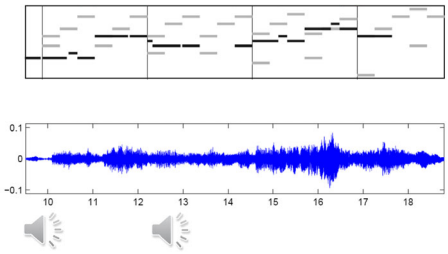


Lyrics-Audio

Stoller, Durand, Ewert: End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. Proc. ICASSP 2019

Lyrics–Audio Alignment

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai

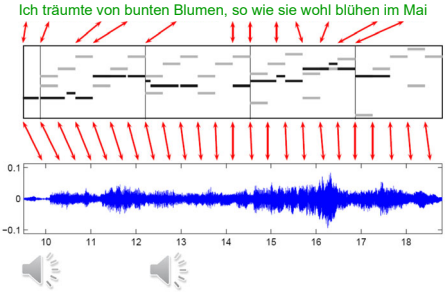


Lyrics-Audio

Stoller, Durand, Ewert: End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. Proc. ICASSP 2019

Lyrics–Audio Alignment

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai



Lyrics-MIDI

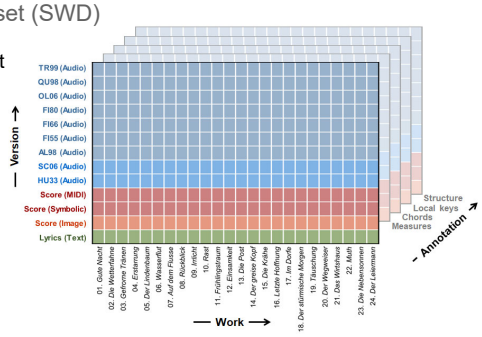
Lyrics-Audio

MIDI-Audio

Stoller, Durand, Ewert: End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. Proc. ICASSP 2019

Datasets
Schubert Winterreise Dataset (SWD)

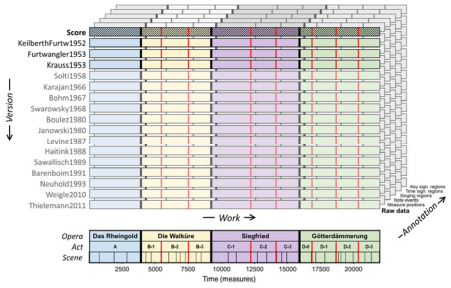
- Song cycle by Franz Schubert
- 24 songs
- 9 performances (versions)
- Annotations
  - Lyrics
  - Chords
  - Local keys
  - Structure



Weiß et al.: Schubert Winterreise Dataset: A Multimodal Scenario for Music Analysis ACM J. Computing & Cultural Heritage, 2021.

Datasets
Wagner Ring Dataset (WRD)

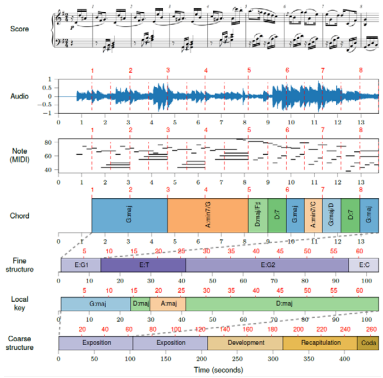
- Opera cycle by Richard Wagner
- 4 operas (ca. 22,000 meaures)
- 16 performances (versions)
- Annotations
  - Lyrics
  - Measure positions
  - Aligned reduced score
  - ...



Weiß et al.: Wagner Ring Dataset: A Complex Opera Scenario for Music Processing and Computational Musicology. TISMIR 2023.

Datasets
Beethoven Piano Sonata Dataset

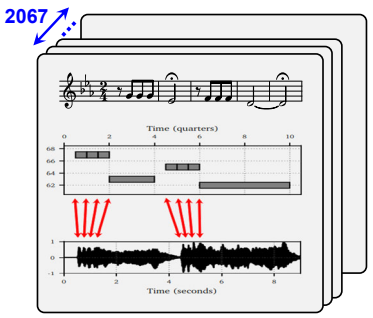
- Piano Sontats by Beethoven
- 32 first movments
- 11 performances (versions)
- Annotations
  - Notes
  - Measures and beats
  - Chords, local & global keys
  - Musical structures



Zeitler et al.: BPSD: A Coherent Multi-Version Dataset for Analyzing the First Movements of Beethoven's Piano Sonatas. TISMIR 2024.

Datasets
Musical Theme Dataset (MTD)

- Western classical music
- Inspired by Barlow & Morgenstern (1948)
- 2067 themes
  - Symbolic encodings
  - Audio excerpts
  - Strong alignments
  - ...



Zalkow et al.: MTD: A Multimodal Dataset of Musical Themes for MIR Research. TISMIR 2020.

References (Differentiable Alignment)

- Marco Cuturi, Mathieu Blondel: Soft-DTW: A Differentiable Loss Function for Time-Series. In Proceedings of the International Conference on Machine Learning (ICML), 894–903, 2017.
- Alex Graves, Santiago Fernández, Faustino Gomez, Jürgen Schmidhuber: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the International Conference on Machine Learning (ICML), 369–376, 2006.
- Michael Krause, Christof Weiß, Meinard Müller: Soft Dynamic Time Warping For Multi Pitch Estimation And Beyond. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1–5, 2023.
- Arthur Mensch, Mathieu Blondel: Differentiable Dynamic Programming for Structured Prediction and Attention. In Proceedings of the International Conference on Machine Learning (ICML), 3459–3468, 2018.
- Daniel Stoller, Simon Durand, Sebastian Ewert: End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 181–185, 2019
- Frank Zalkow, Meinard Müller: CTC-Based Learning of Chroma Features for Score-Audio Music Retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29: 2957–2971, 2021.
- Johannes Zeitler, Simon Deniffel, Michael Krause, Meinard Müller: Stabilizing Training with Soft Dynamic Time Warping: A Case Study for Pitch Class Estimation with Weakly Aligned Targets. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR): 433–439, 2023.
- Johannes Zeitler, Meinard Müller: A Unified Perspective on CTC and Soft-DTW Using Differentiable DTW. IEEE Transactions on Audio, Speech, and Language Processing (TASLPRO), 34: 936–951, 2026.