Soft-DTW: A Differentiable Loss Function for Time Series

Marco Cuturi

université PARIS-SACLAY

ENSAE ParisTech
École nationale de la statistique et de l'administration économique
université PARIS-SACLAY

Mathieu Blondel

NTT

In proceedings of ICML'17

From: Plume App

Ground truth
(reality)

From: Plume App

How wrong was this prediction?

This depends on the loss function used to train the algorithm.

Ground truth (reality)

From: Plume App

- In this talk we propose to use the celebrated **Dynamic Time Warping** discrepancy as a loss.

- Loss functions should be **differentiable**. We show that an **appropriate smoothing** , **soft-DTW**, helps

- We apply this to **several problems**:
  - Computation of **barycenters**,
  - Clustering of time series,
  - Learning with structured (time series) output

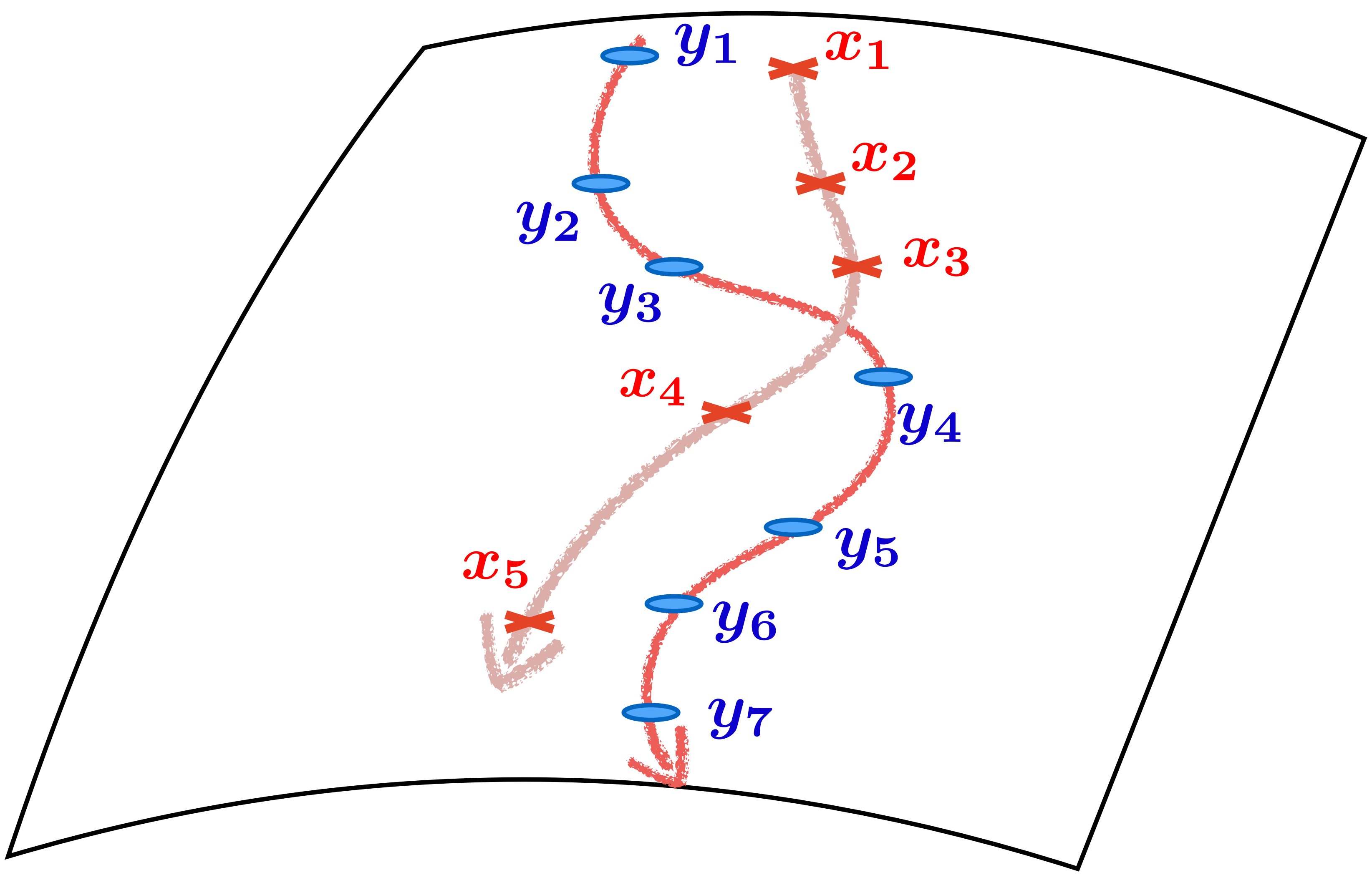
0. The DTW Geometry

1. Soft-DTW

2. Soft-DTW as a Loss Function

Dynamic Time Warping [Sakoe&Chiba'78]

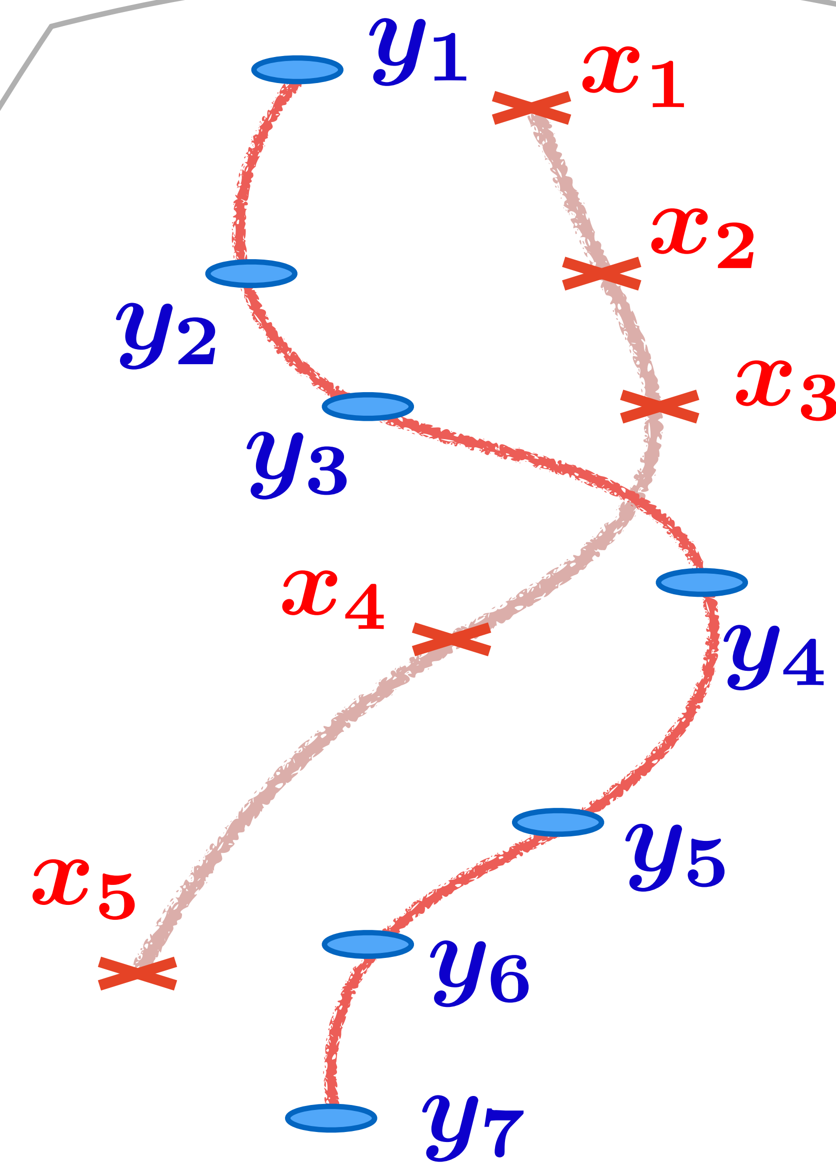
A discrepancy function between two **time series of observations** supported **on a metric space.**



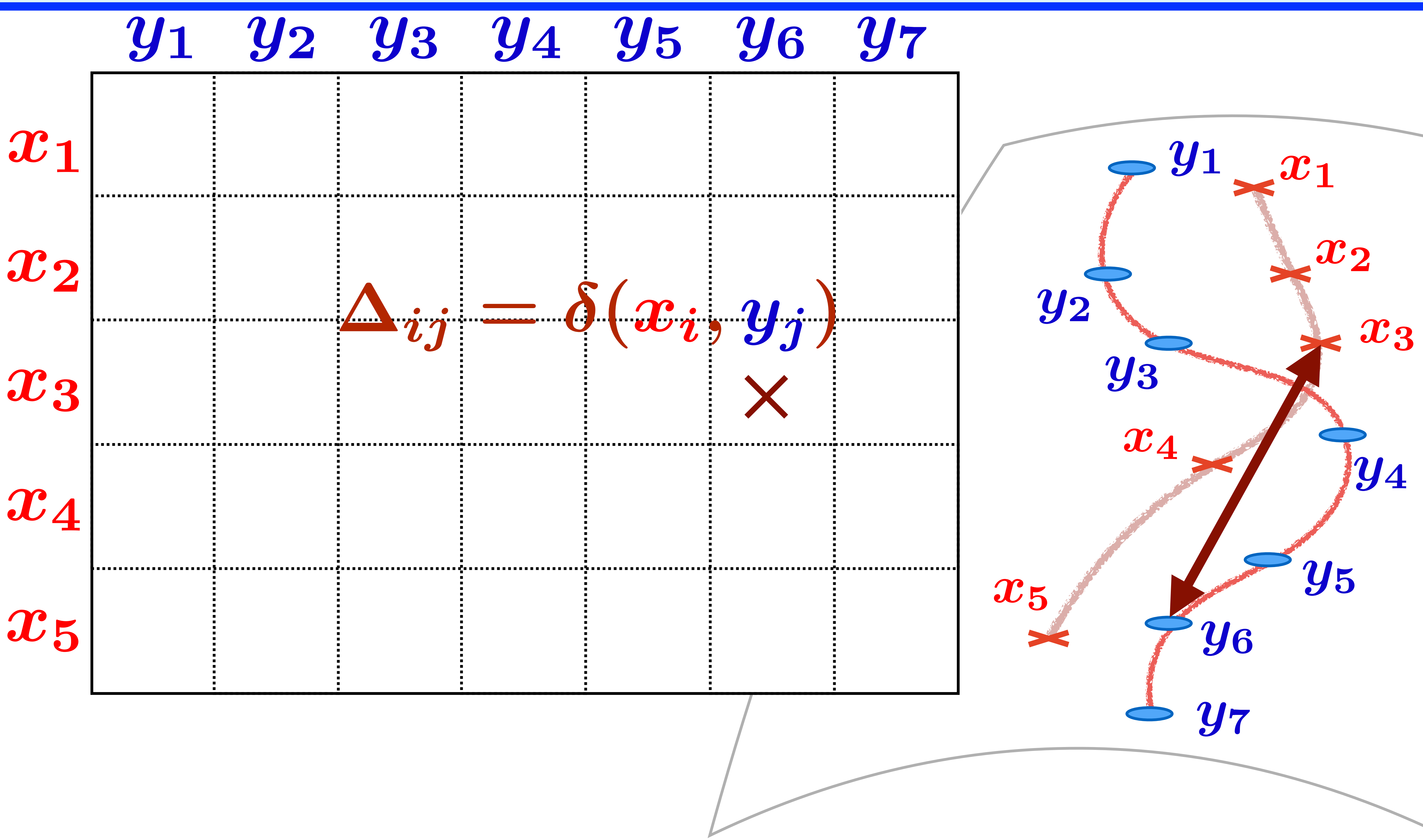
(Ω, δ)

Pairwise Distance Matrix

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1							
x_2							
x_3							
x_4							
x_5							

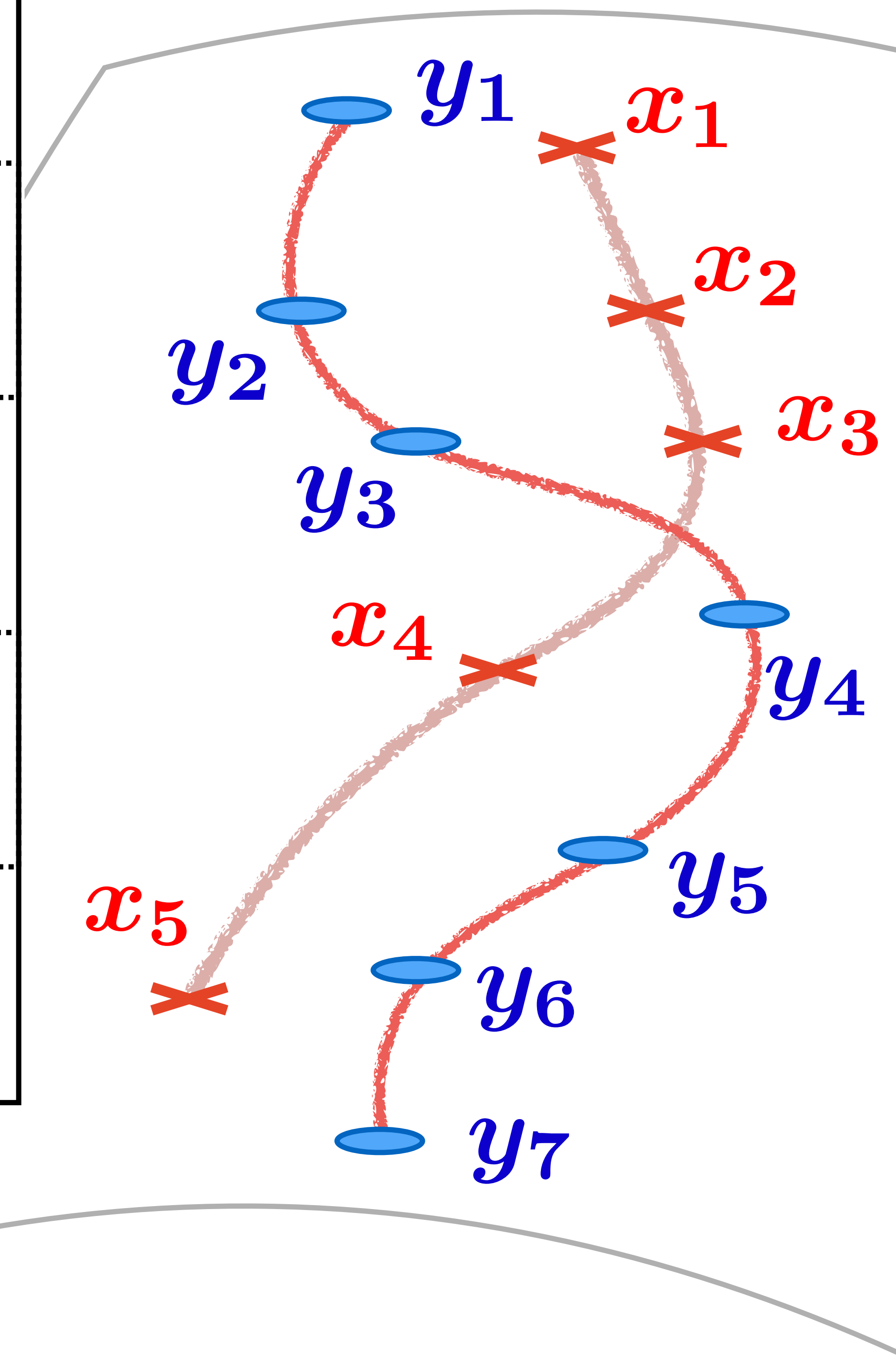


Pairwise Distance Matrix



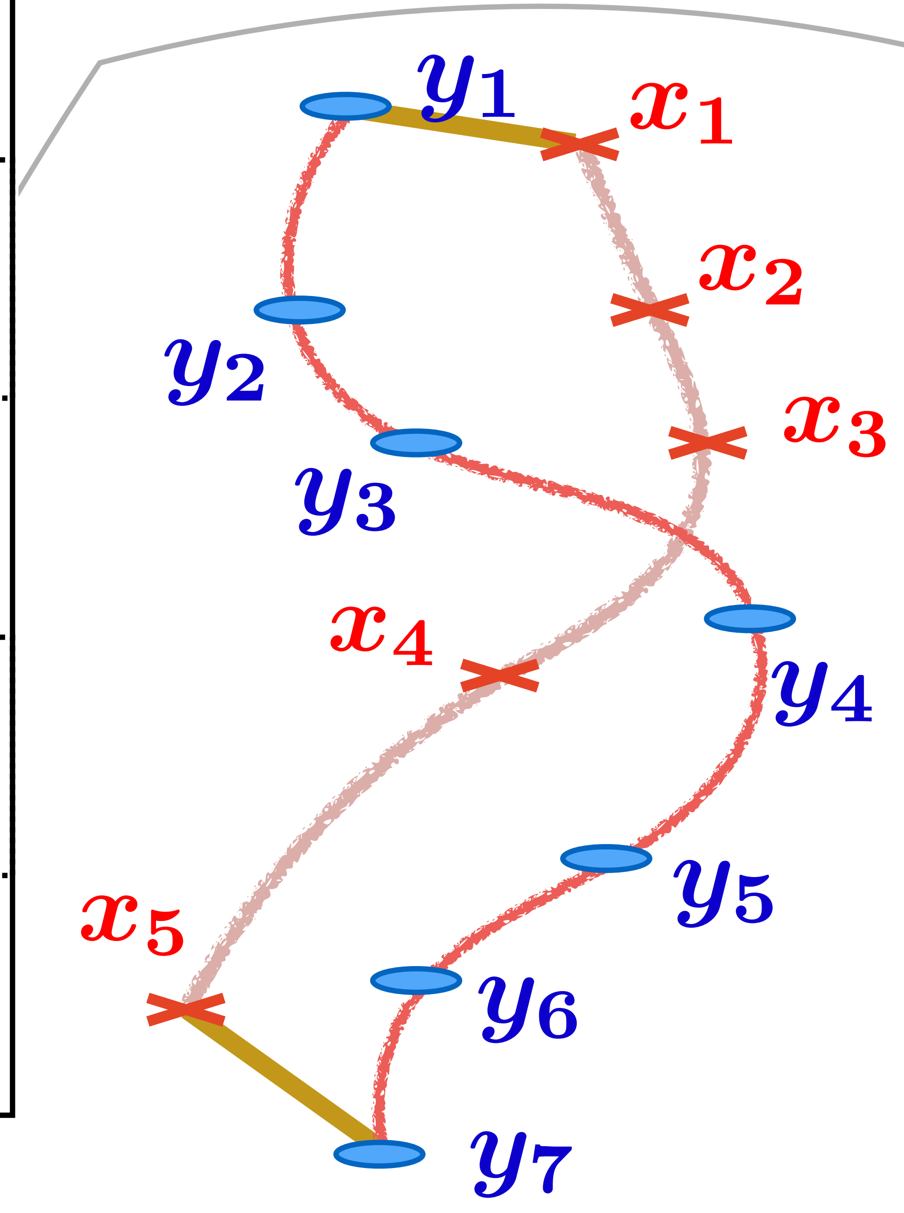
Pairwise Distance Matrix

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}

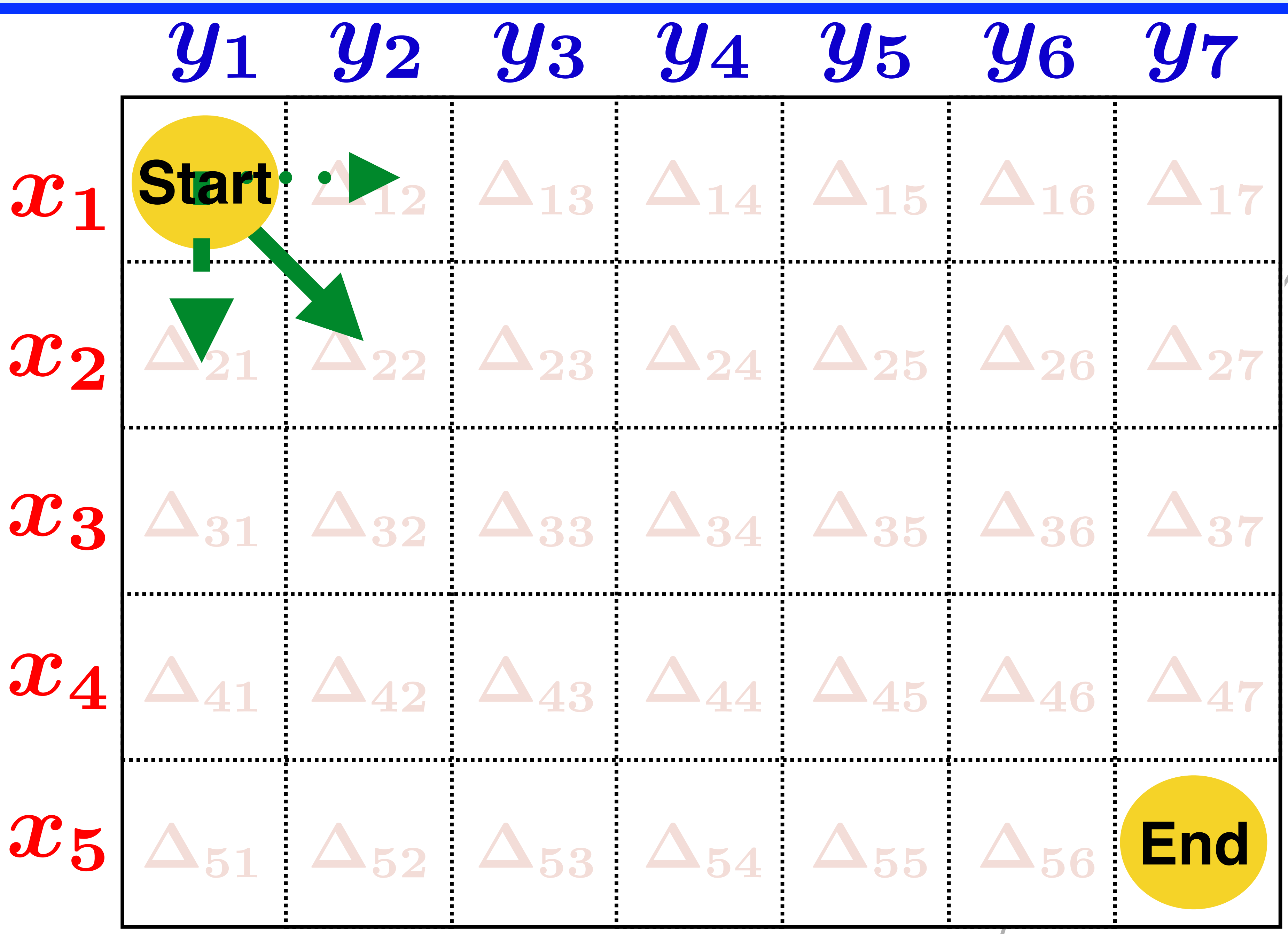


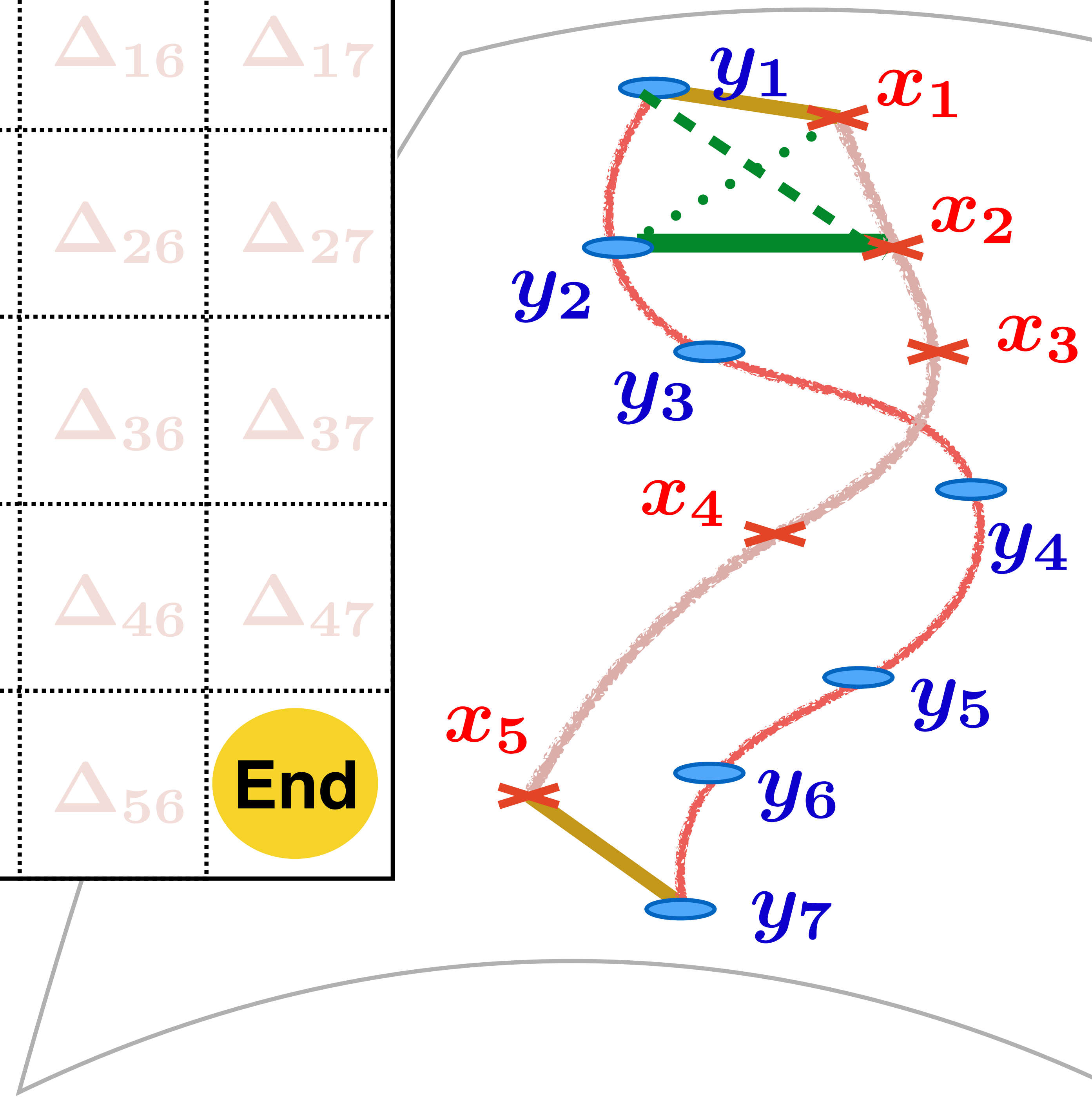
Alignment Path

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Start	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	End

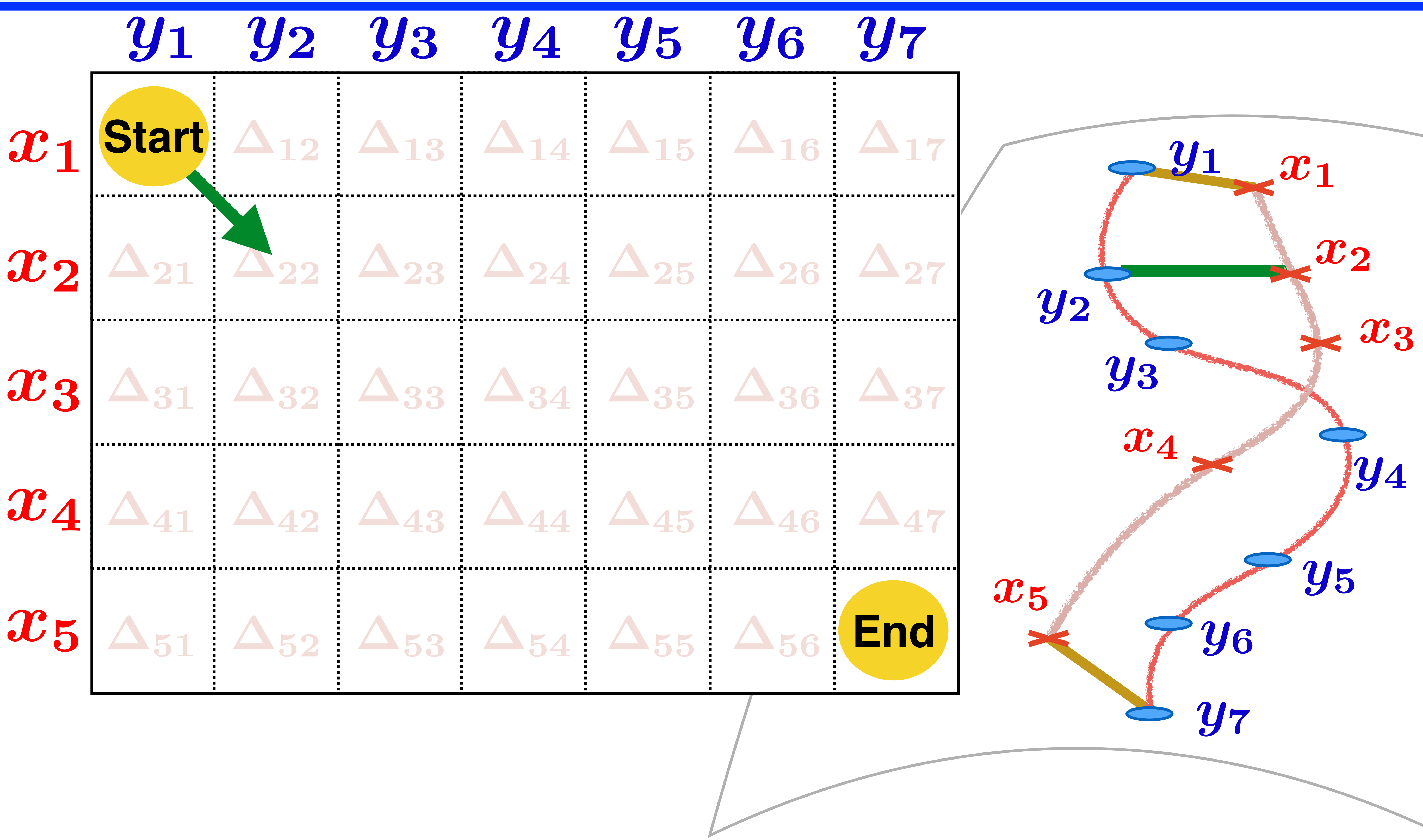


Alignment Path

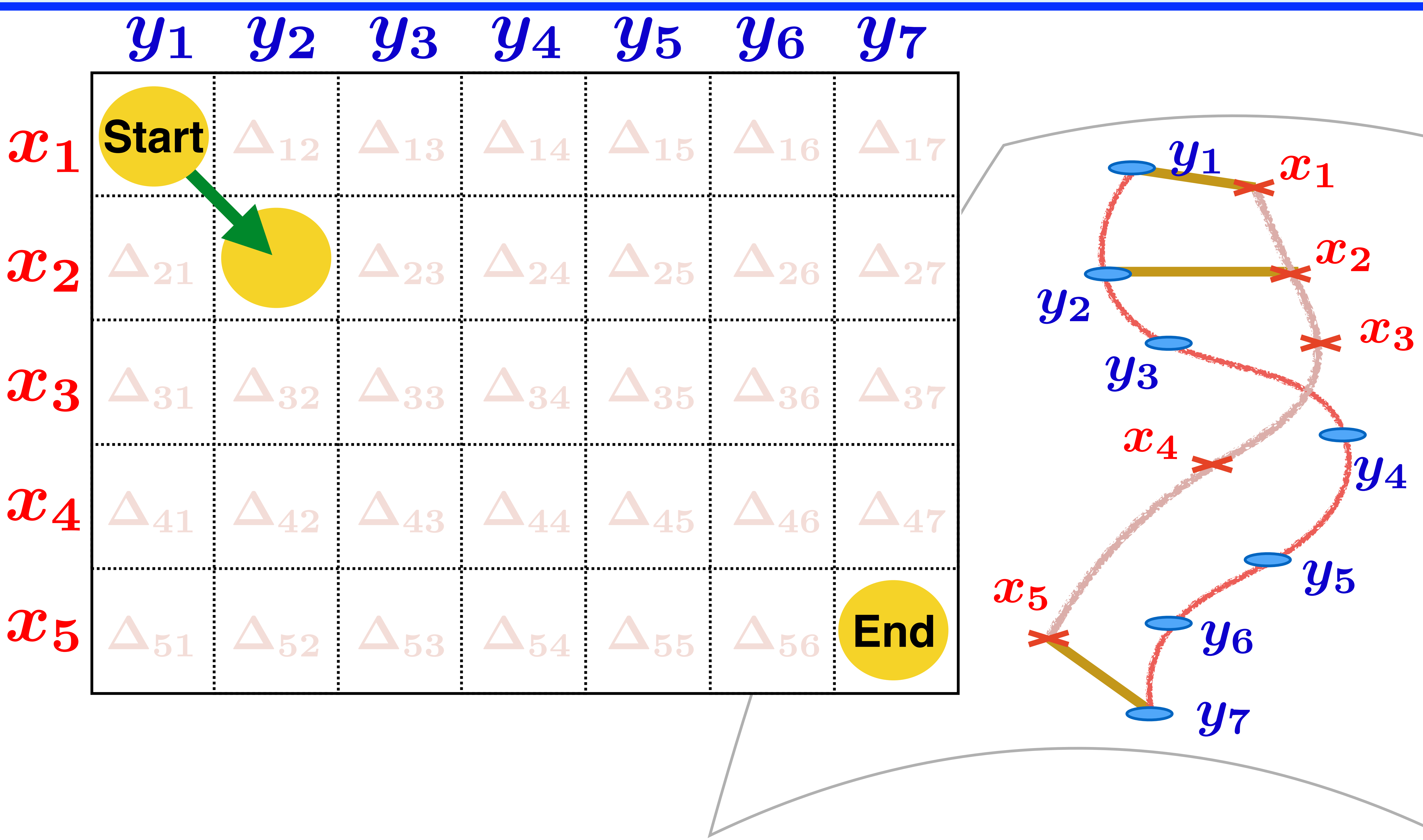




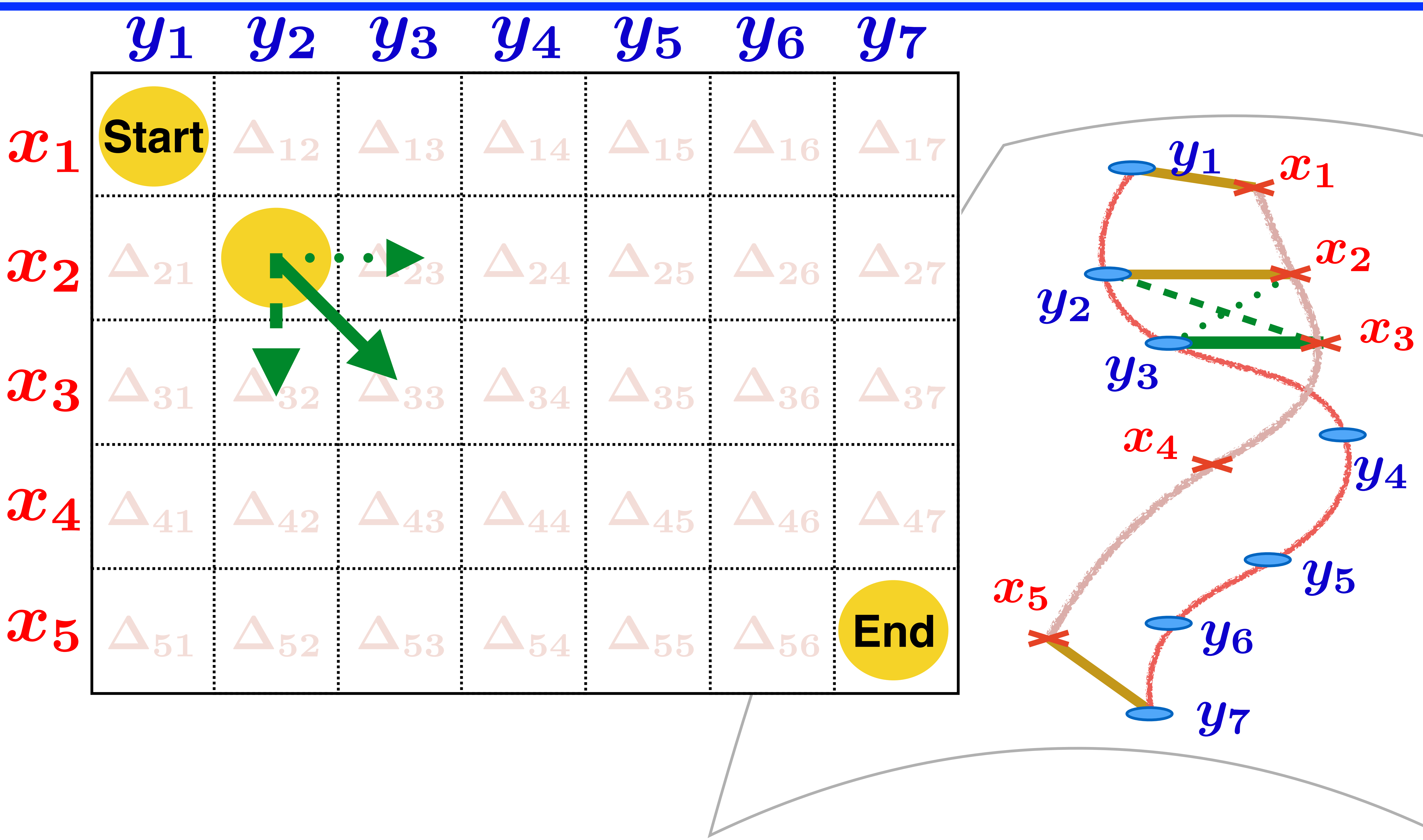
Alignment Path



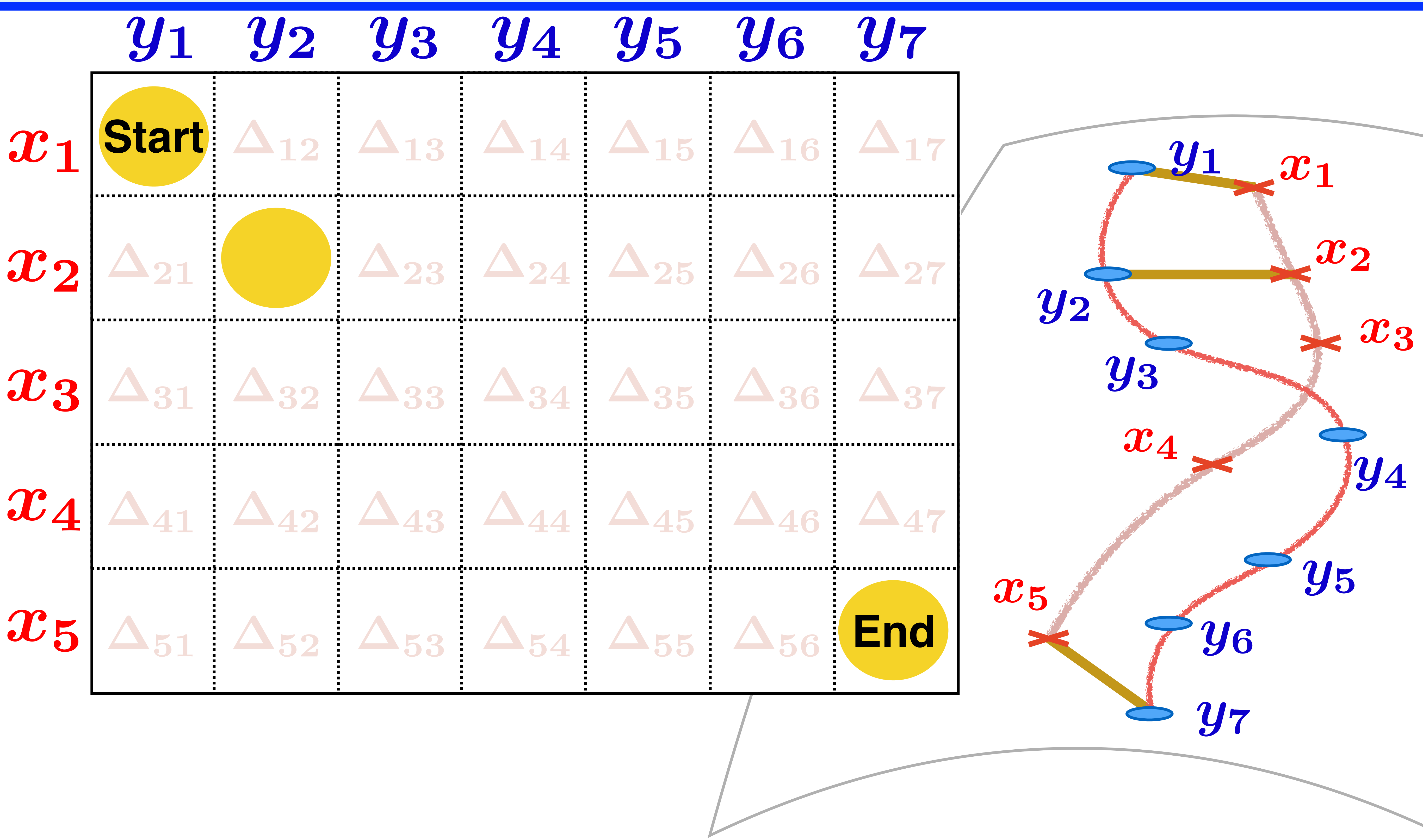
Alignment Path



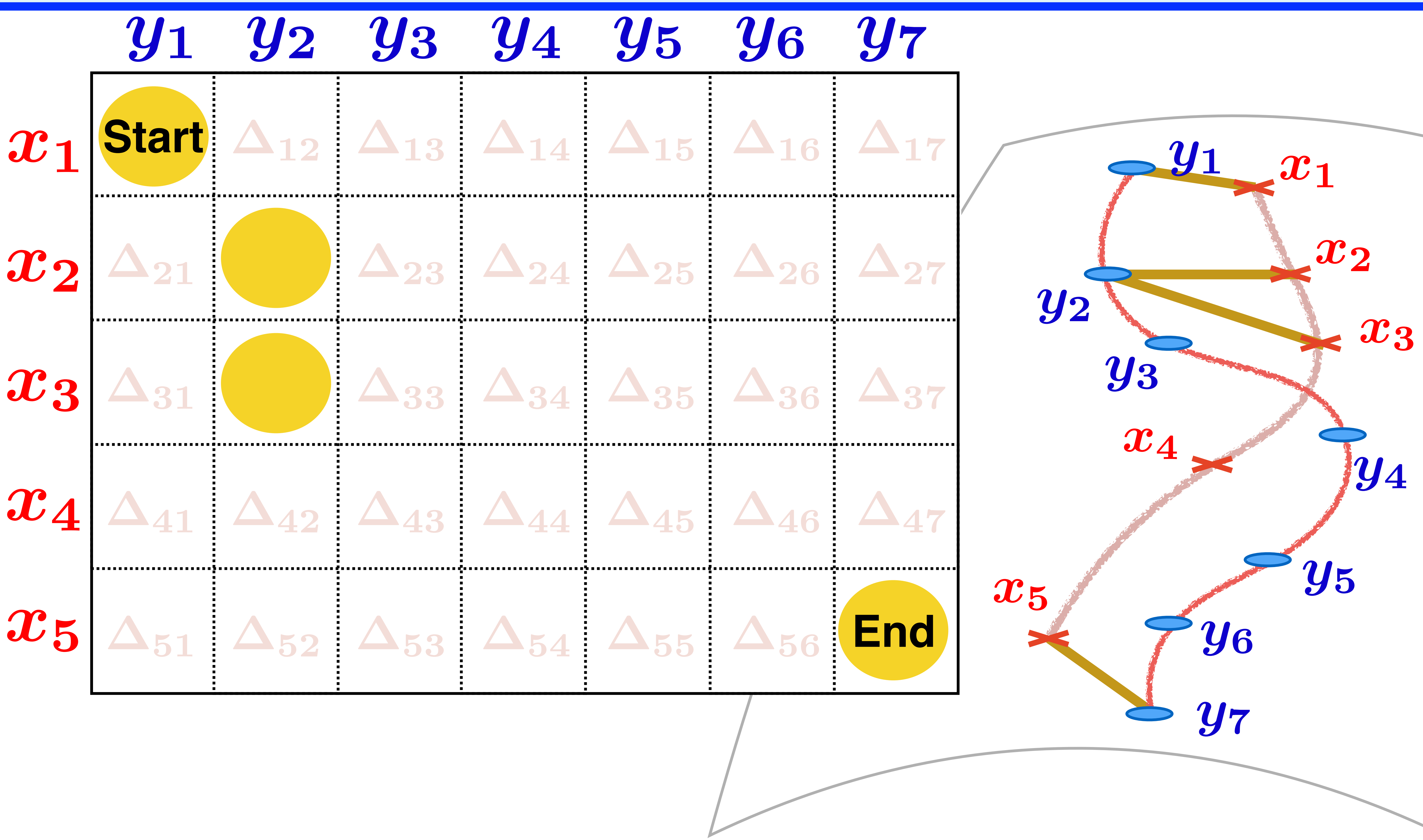
Alignment Path



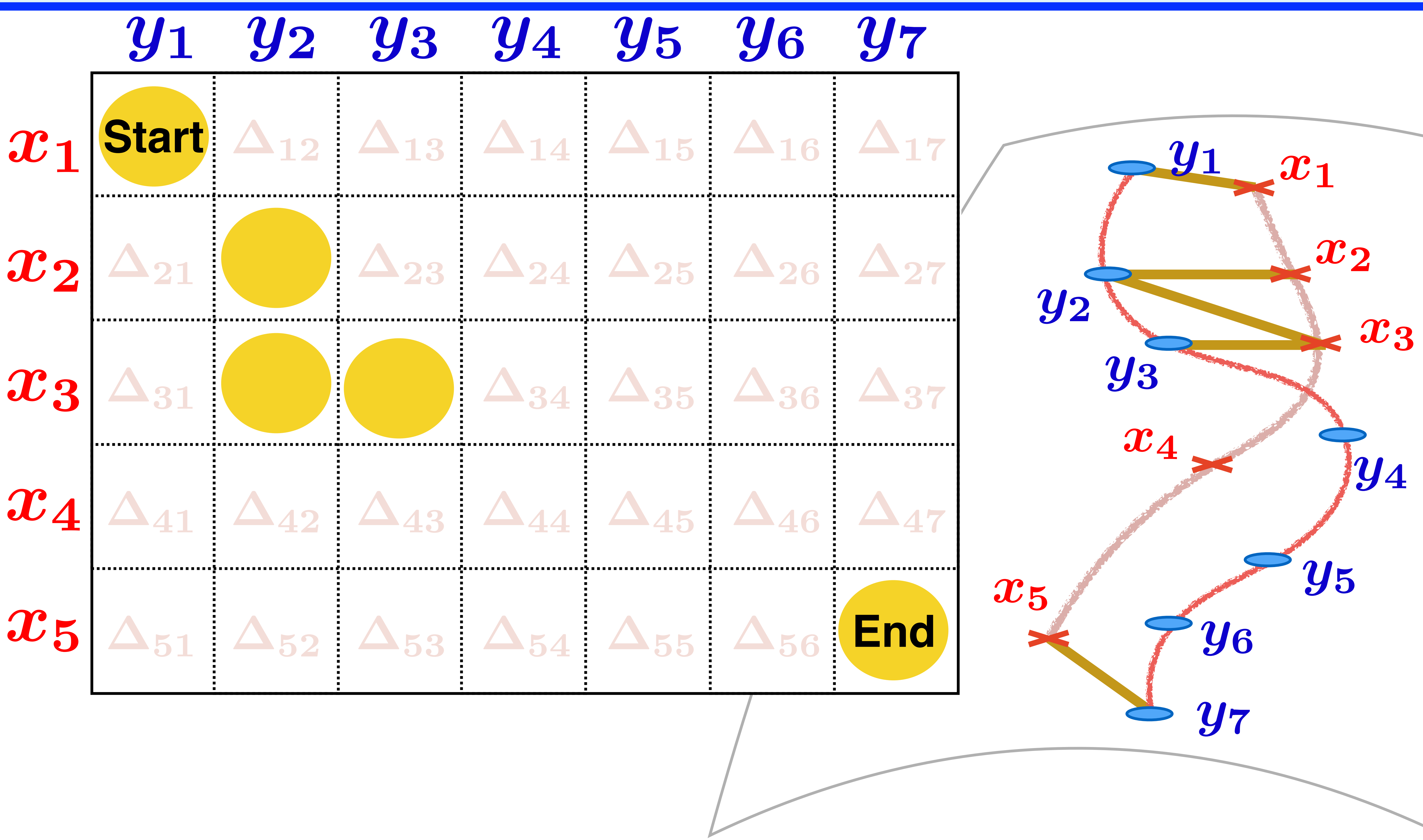
Alignment Path



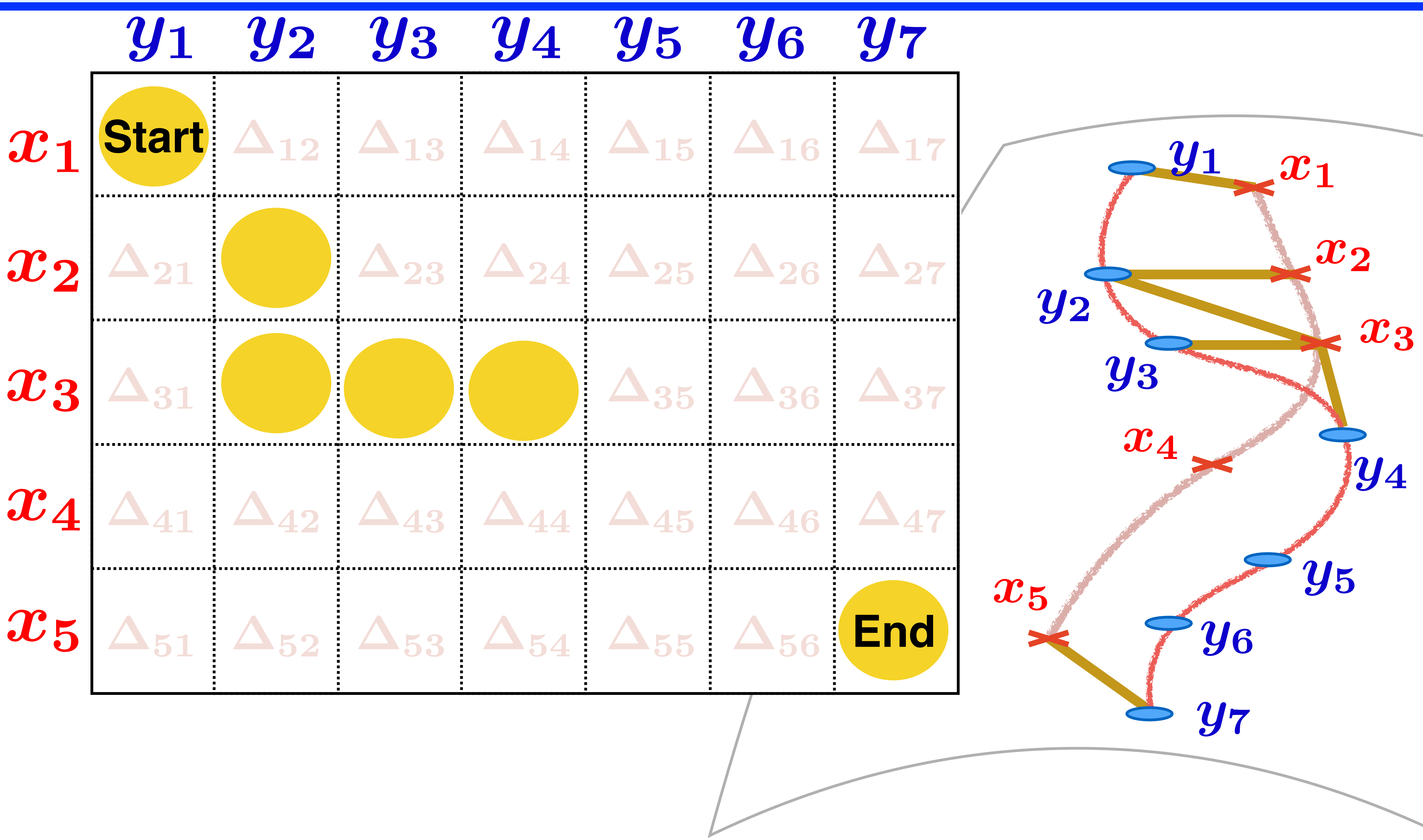
Alignment Path



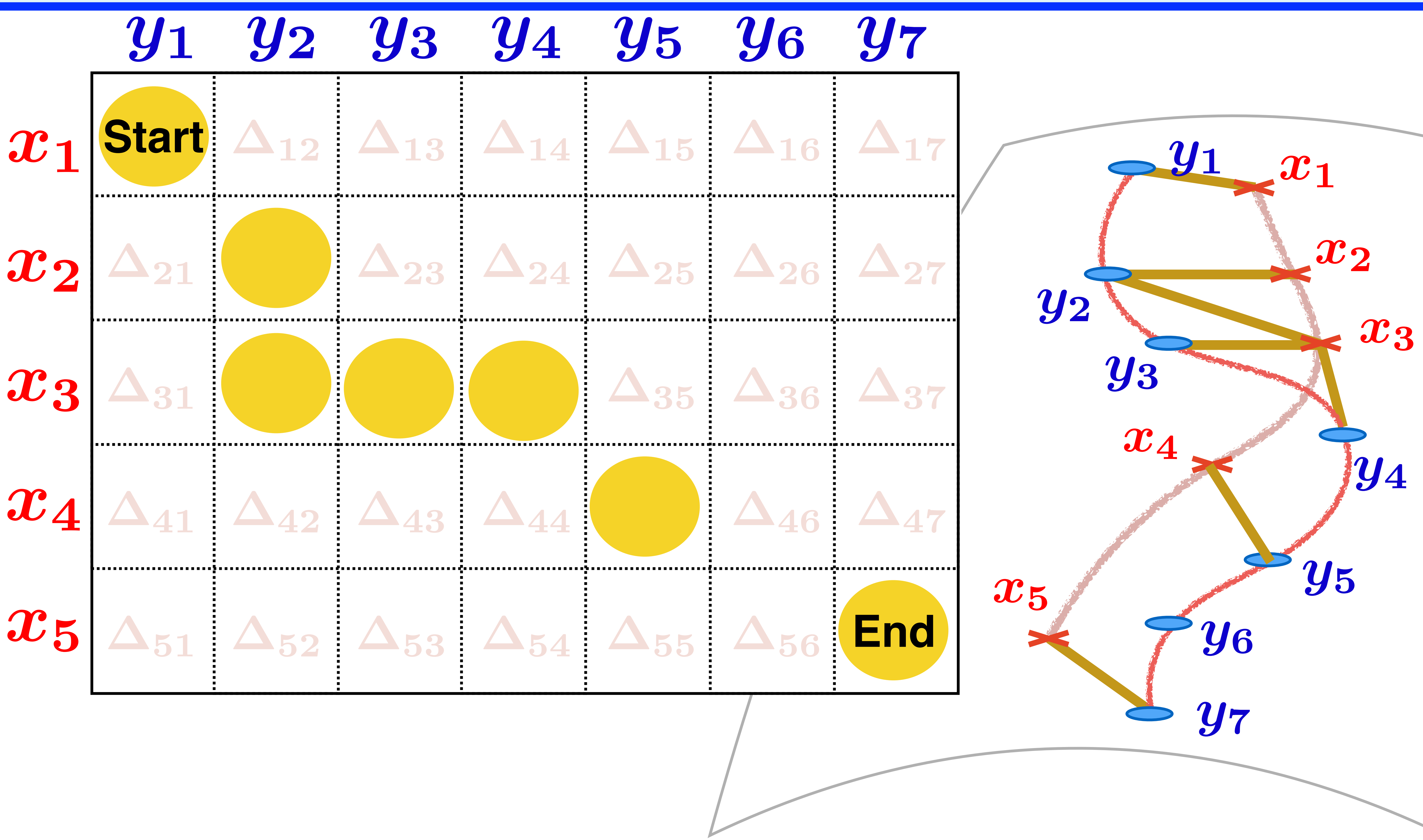
Alignment Path



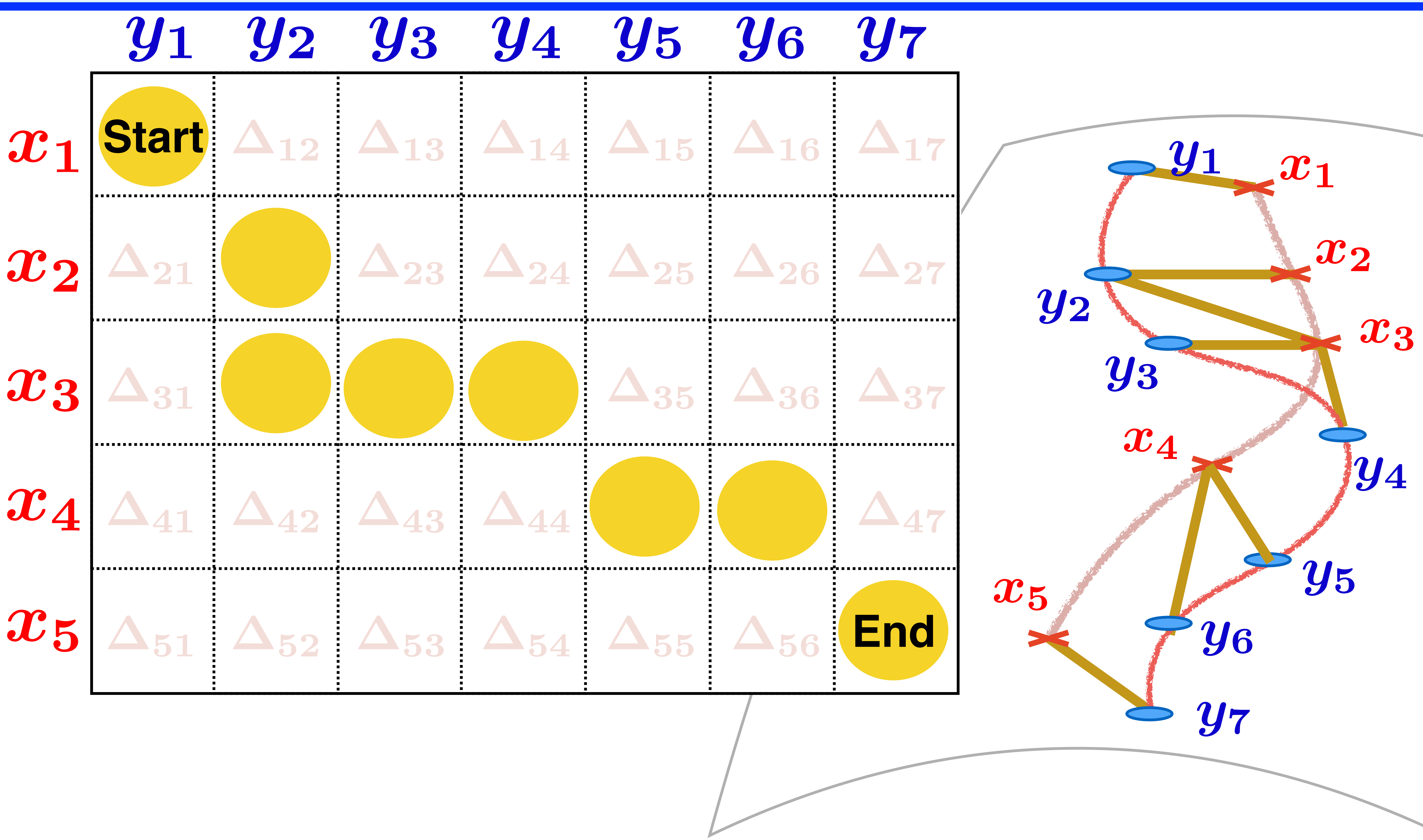
Alignment Path



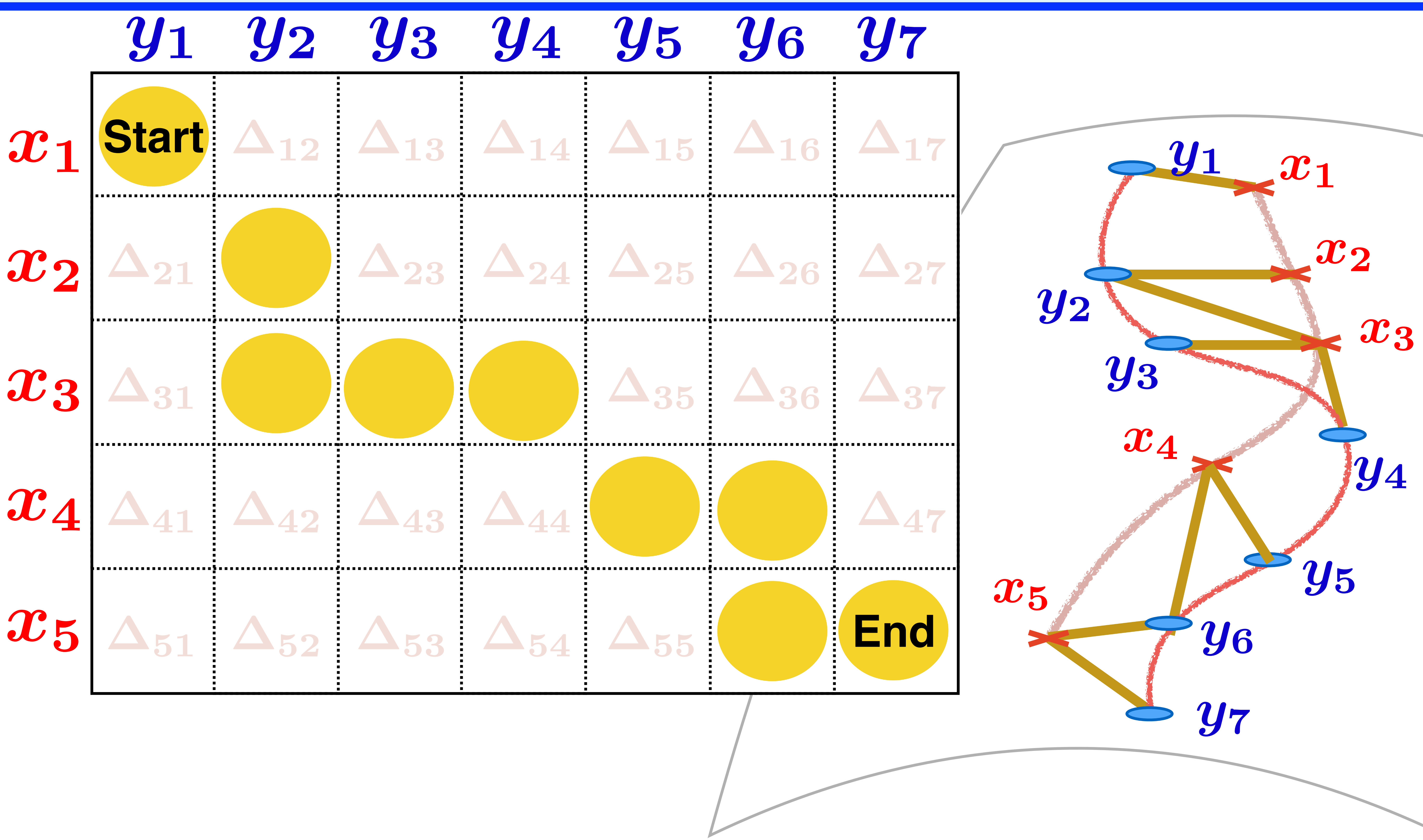
Alignment Path



Alignment Path

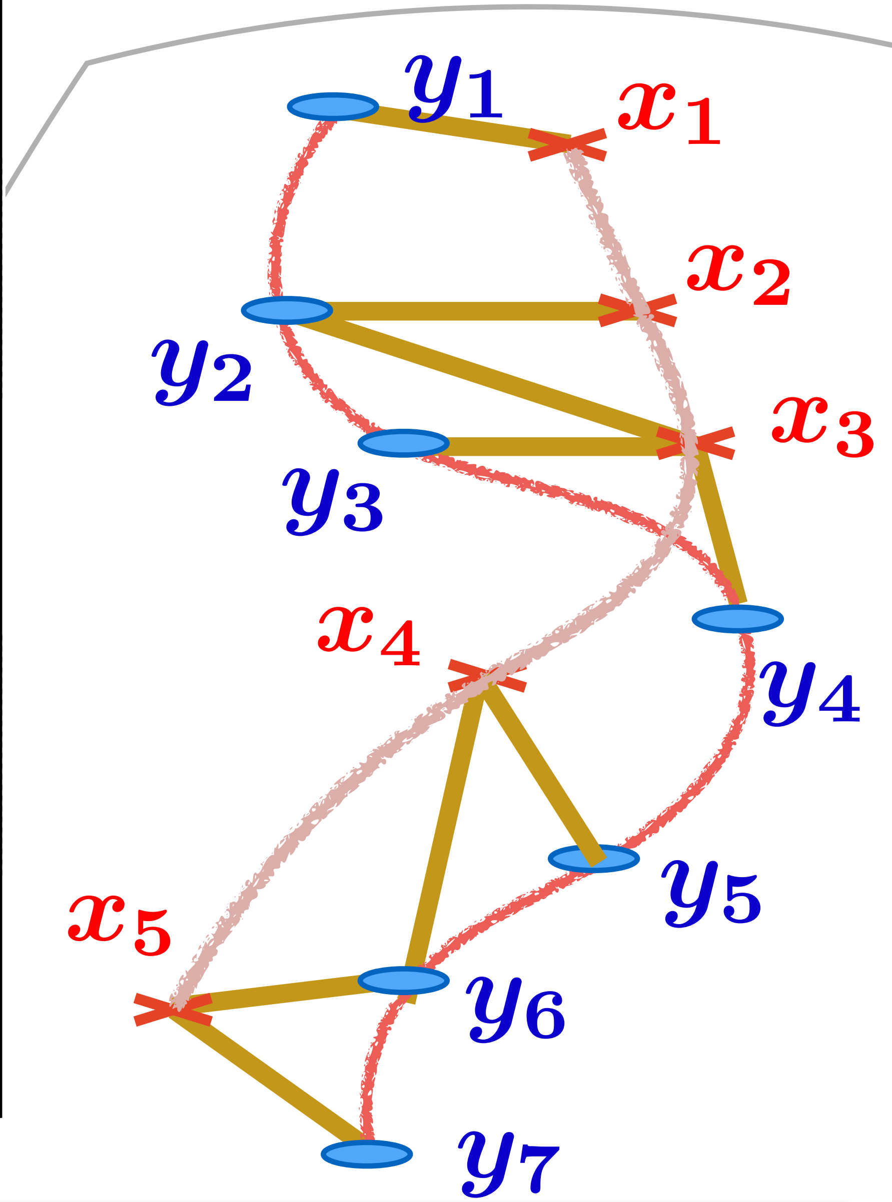


Alignment Path



Path Cost

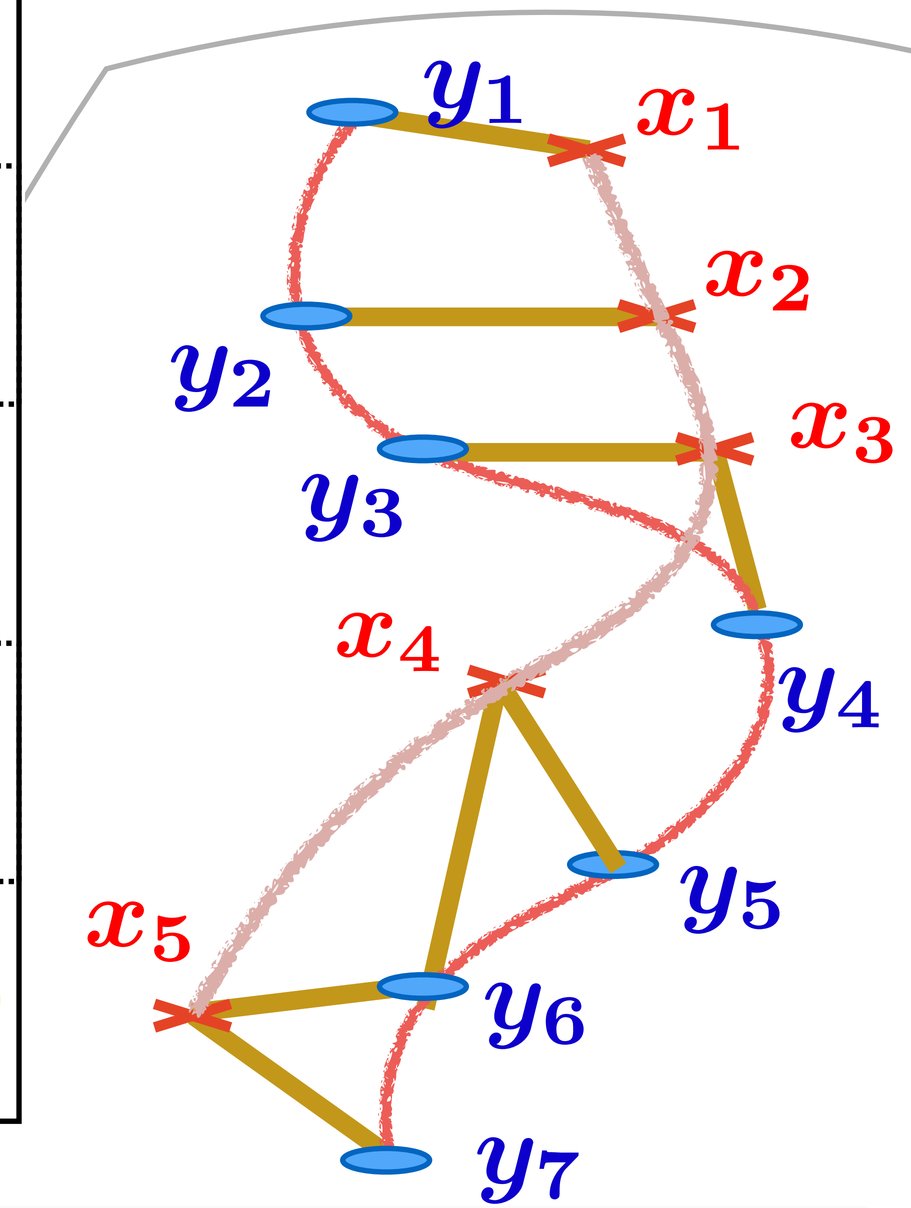
	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}



$$\text{Cost} = \Delta_{11} + \Delta_{22} + \Delta_{32} + \Delta_{33} + \Delta_{34} + \Delta_{46} + \Delta_{56} + \Delta_{57}$$

Path Cost

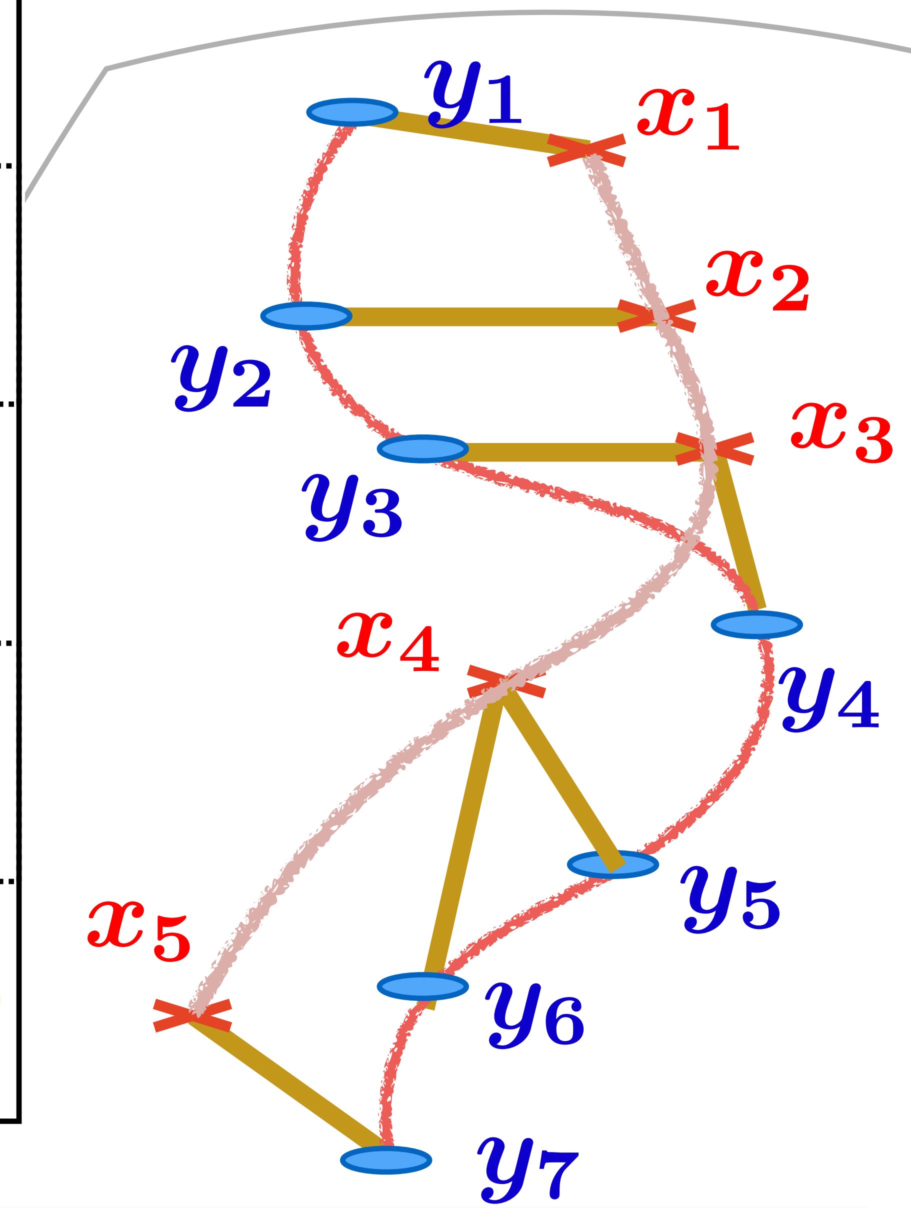
	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}



$$\text{Cost} = \Delta_{11} + \Delta_{22} \quad + \quad \Delta_{33} + \Delta_{34} + \Delta_{46} + \Delta_{56} + \Delta_{57}$$

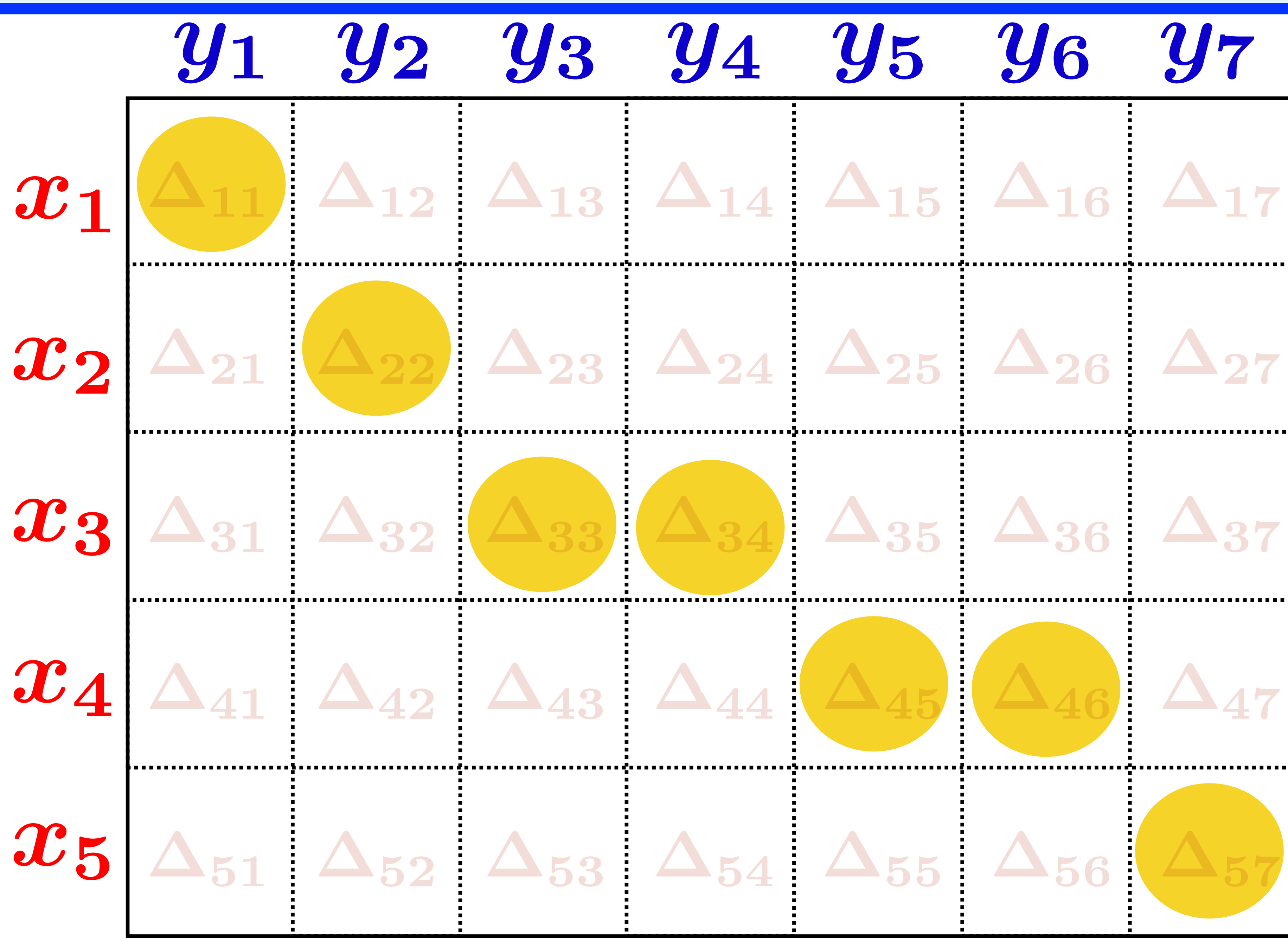
Path Cost

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}



$$\text{Cost} = \Delta_{11} + \Delta_{22} \quad + \quad \Delta_{33} + \Delta_{34} + \Delta_{46} \quad + \quad \Delta_{57}$$

Path Cost



y_1 y_2 y_3 y_4 y_5 y_6 y_7
x_1 x_2 x_3 x_4 x_5
Δ_{11} Δ_{12} Δ_{13} Δ_{14} Δ_{15} Δ_{16} Δ_{17}
Δ_{21} Δ_{22} Δ_{23} Δ_{24} Δ_{25} Δ_{26} Δ_{27}
Δ_{31} Δ_{32} Δ_{33} Δ_{34} Δ_{35} Δ_{36} Δ_{37}
Δ_{41} Δ_{42} Δ_{43} Δ_{44} Δ_{45} Δ_{46} Δ_{47}
Δ_{51} Δ_{52} Δ_{53} Δ_{54} Δ_{55} Δ_{56} Δ_{57}

Path Cost

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	**1**	0	0	0	0	0	0
x_2	0	**1**	0	0	0	0	0
x_3	0	0	**1**	**1**	0	0	0
x_4	0	0	0	0	**1**	**1**	0
x_5	0	0	0	0	0	0	**1**

$$= A$$

Path Cost

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	**1**	**0**	**0**	**0**	**0**	**0**	**0**
x_2	**0**	**1**	**0**	**0**	**0**	**0**	**0**
x_3	**0**	**0**	**1**	**1**	**0**	**0**	**0**
x_4	**0**	**0**	**0**	**0**	**1**	**1**	**0**
x_5	**0**	**0**	**0**	**0**	**0**	**0**	**1**

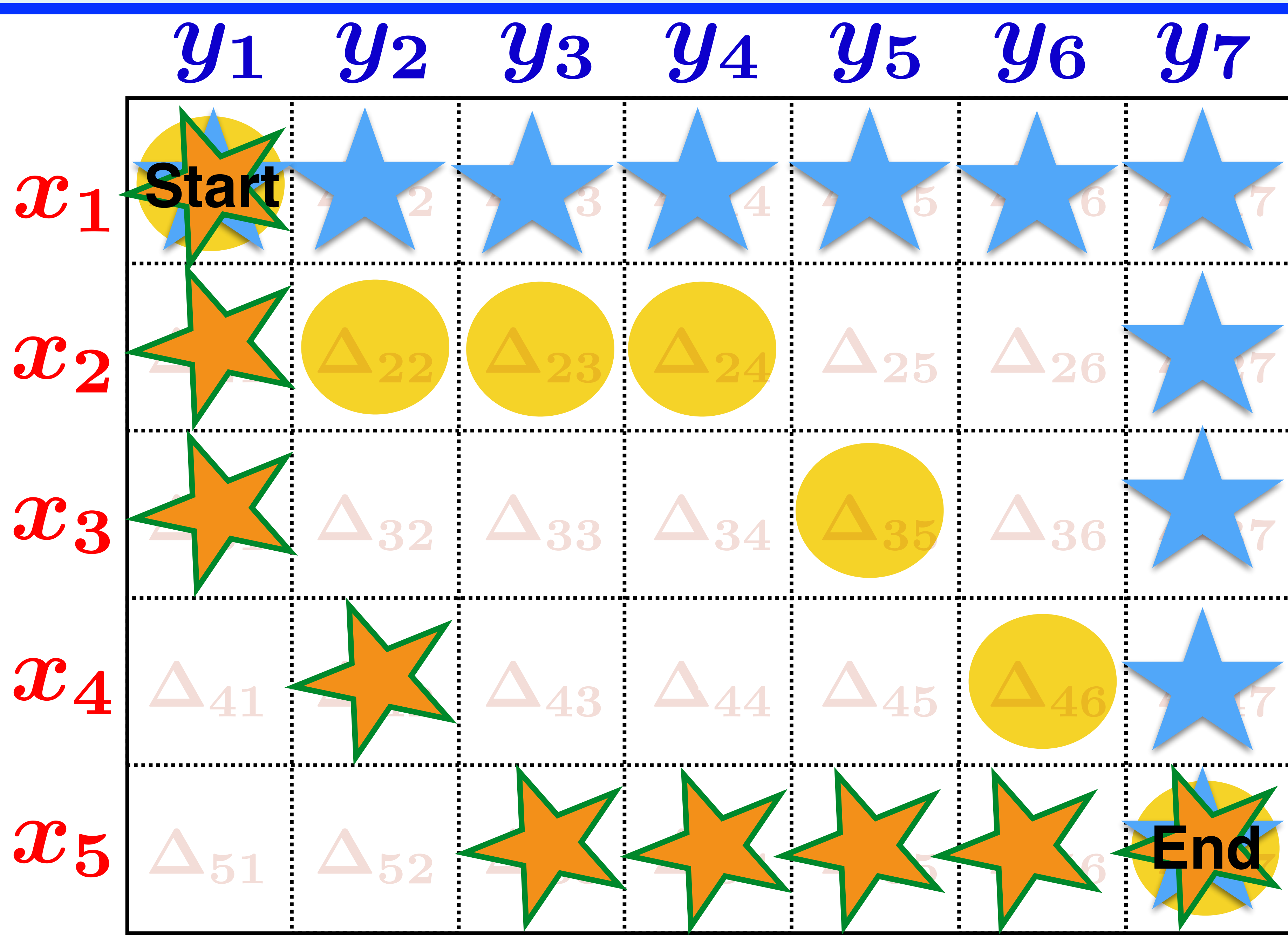
$= A$

$$\text{Cost} = \langle A, \mathbf{\Delta} \rangle, A \in \{0, 1\}^{n \times m}$$

Minimum Cost Alignment Matrix?

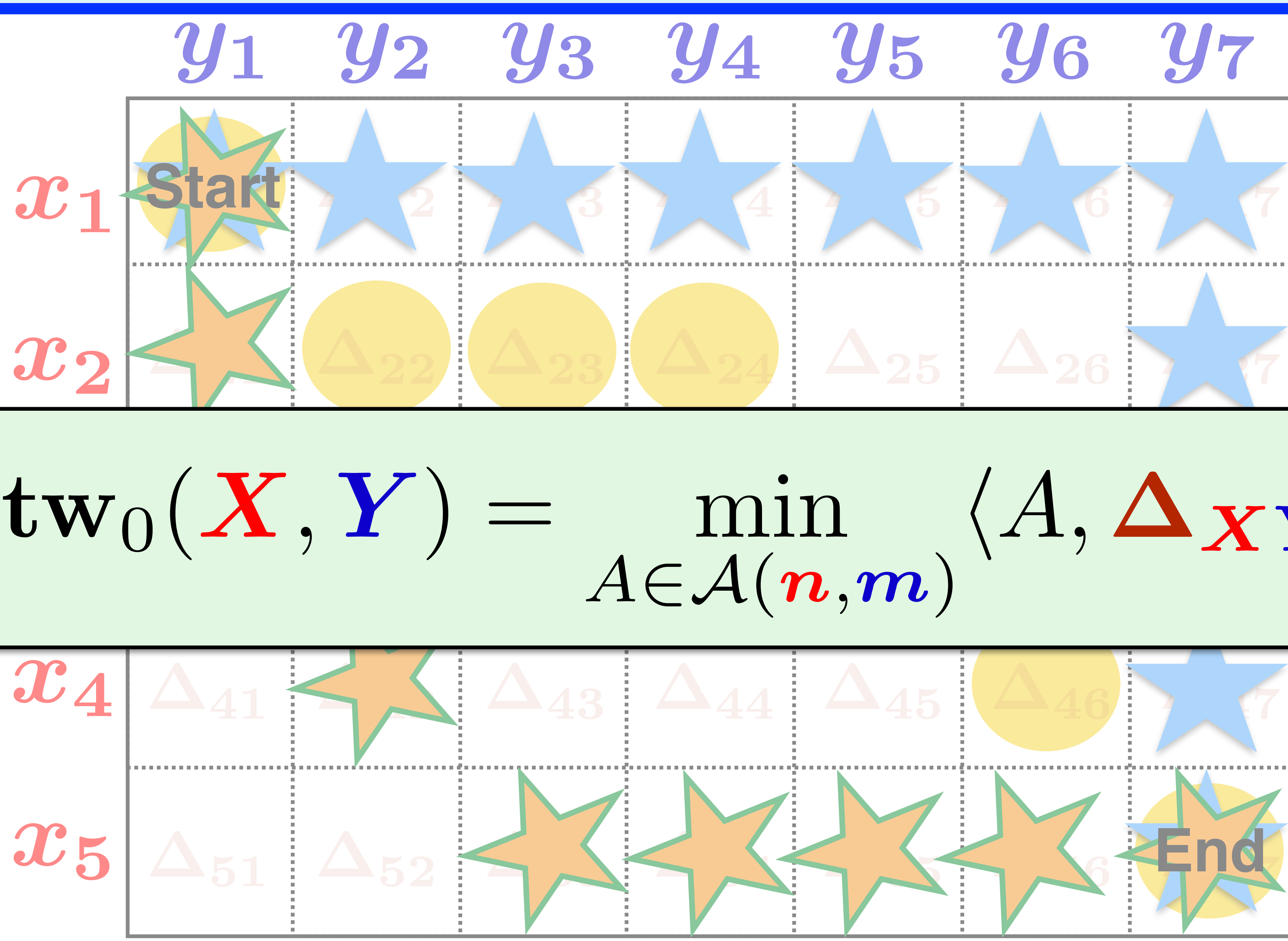
	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	**Start** Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	**End** Δ_{57}

Minimum Cost Alignment Matrix?



Set of all valid path matrices: $\mathcal{A}(n, m) \subset \{0,1\}^{n \times m}$

Dynamic Time Warping [Sakoe&Chiba'78]



$$\mathbf{dtw}_0(\mathbf{X},\mathbf{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\mathbf{XY}} \rangle$$

Set of all valid path matrices: $\mathcal{A}(\boldsymbol{n},\boldsymbol{m}) \subset \{0,1\}^{\boldsymbol{n}\times\boldsymbol{m}}$

Number of valid paths

Size of $\mathcal{A}(\boldsymbol{n}, \boldsymbol{m})$ is exponential in $\boldsymbol{n}, \boldsymbol{m}$.

$$\#\mathcal{A}(\boldsymbol{n}, \boldsymbol{m}) = \text{Delannoy}(\boldsymbol{n} - 1, \boldsymbol{m} - 1)$$

n=m=3	13
n=m=5	321
n=m=10	1462563

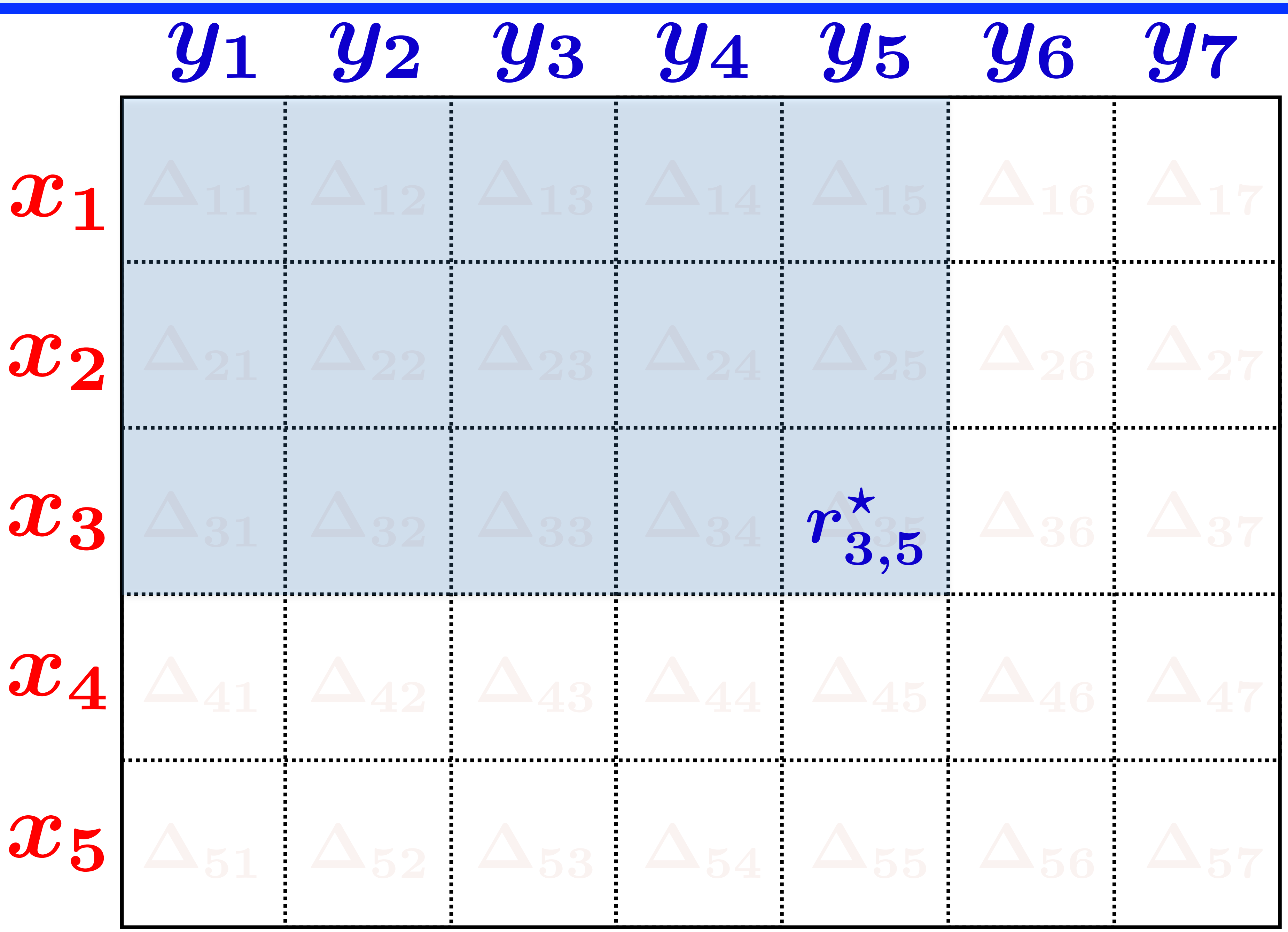
$\cdots$

Set of all valid path matrices: $\mathcal{A}(\boldsymbol{n}, \boldsymbol{m}) \subset \{0, 1\}^{\boldsymbol{n} \times \boldsymbol{m}}$

Best Path: Bellman Recursion

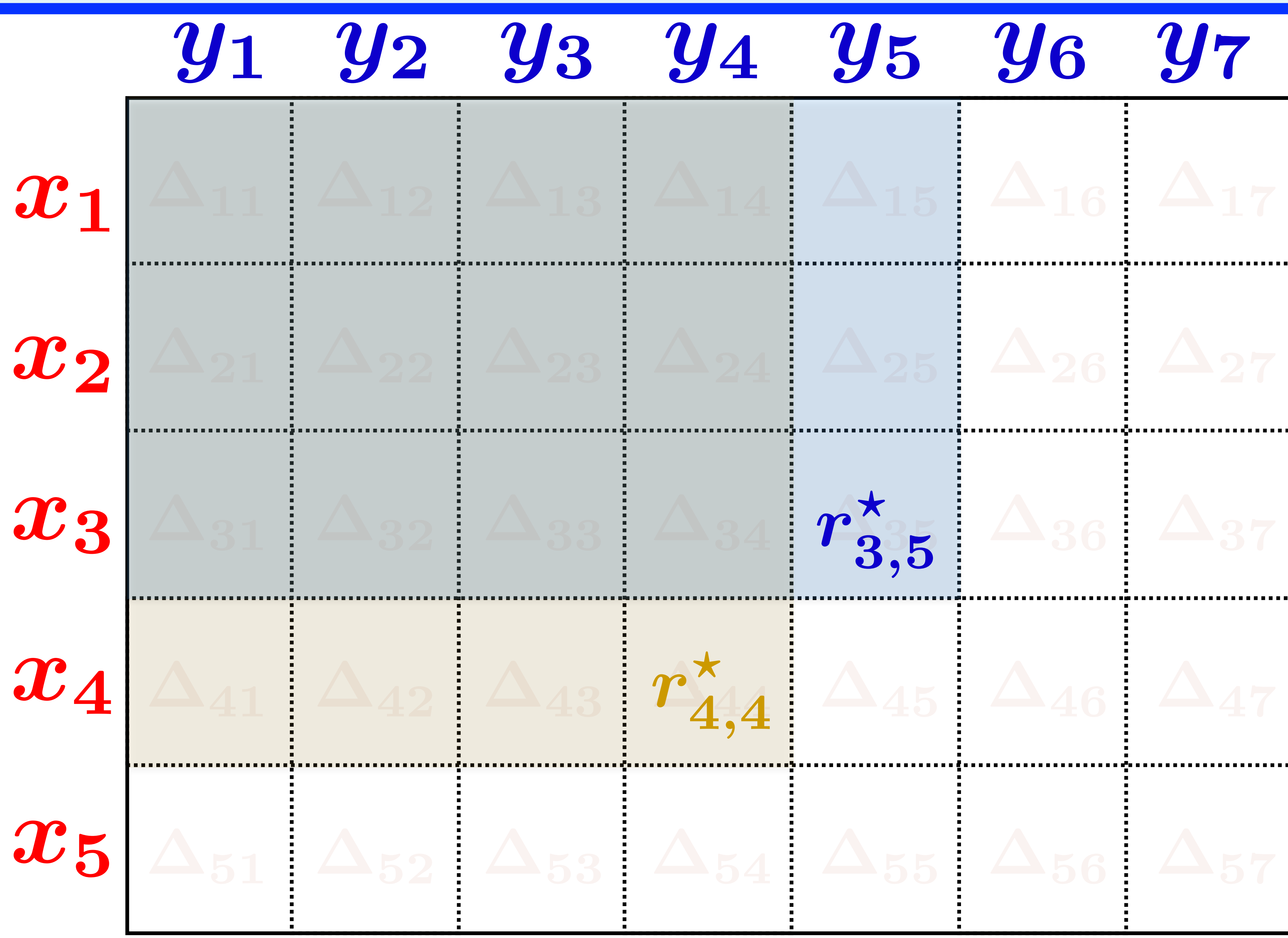
	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
x_2	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
x_3	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
x_4	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
x_5	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}

Best Path: Bellman Recursion

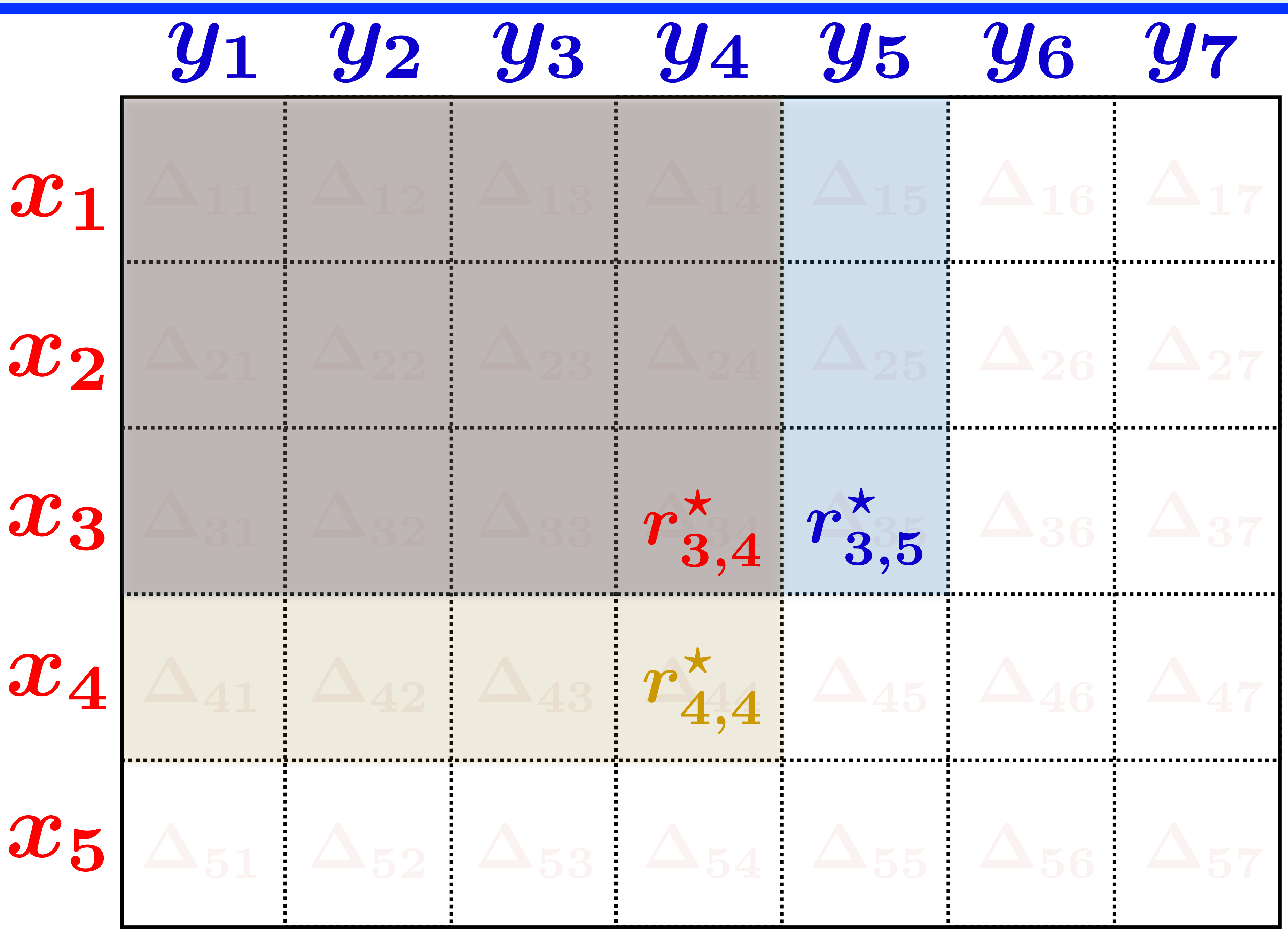


$$r^\star_{3,5} = \min_{A \in \mathcal{A}(3,5)} \langle A, [\Delta_{ij}]_{i \leq 3, j \leq 5} \rangle$$

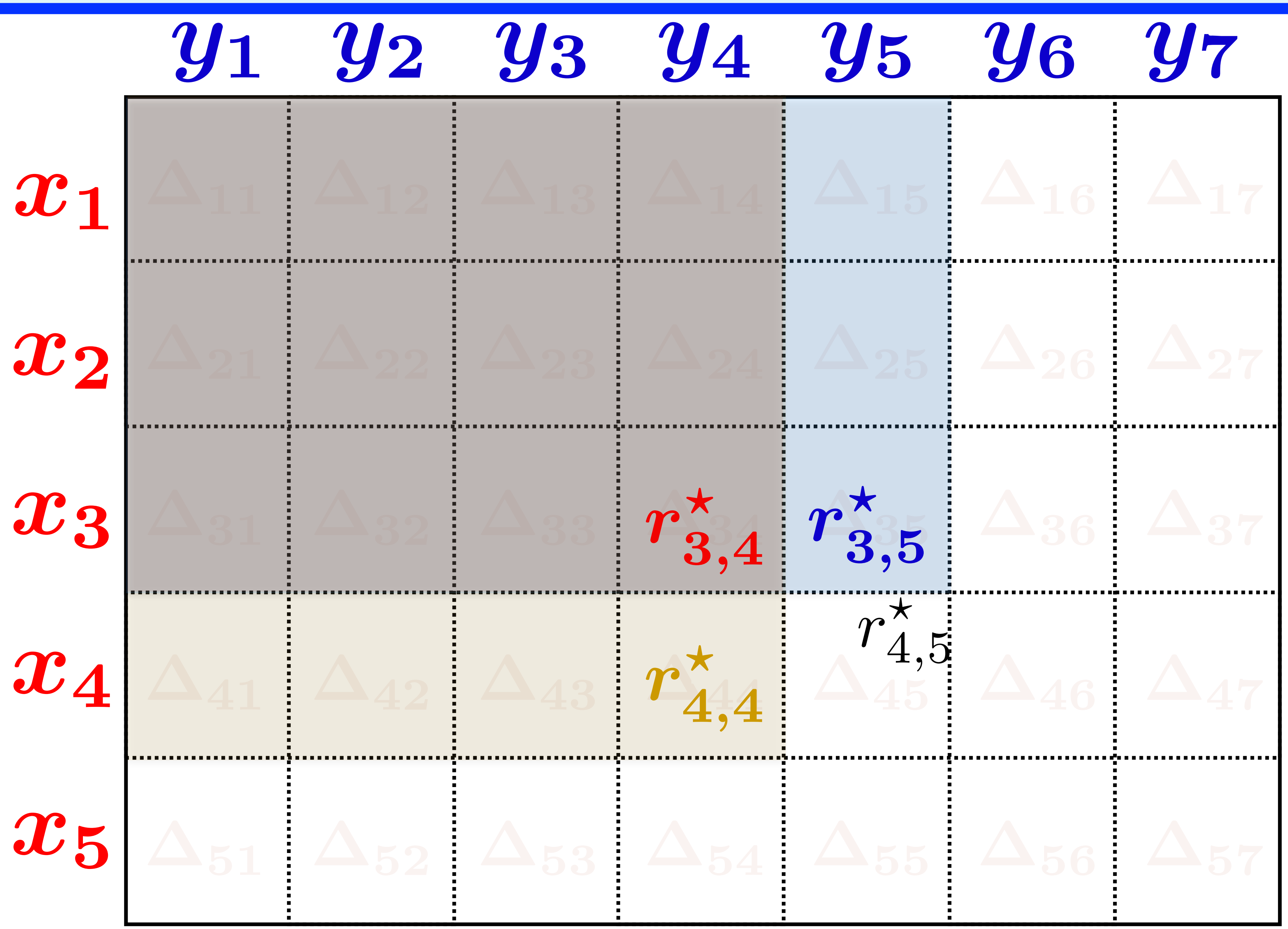
Best Path: Bellman Recursion



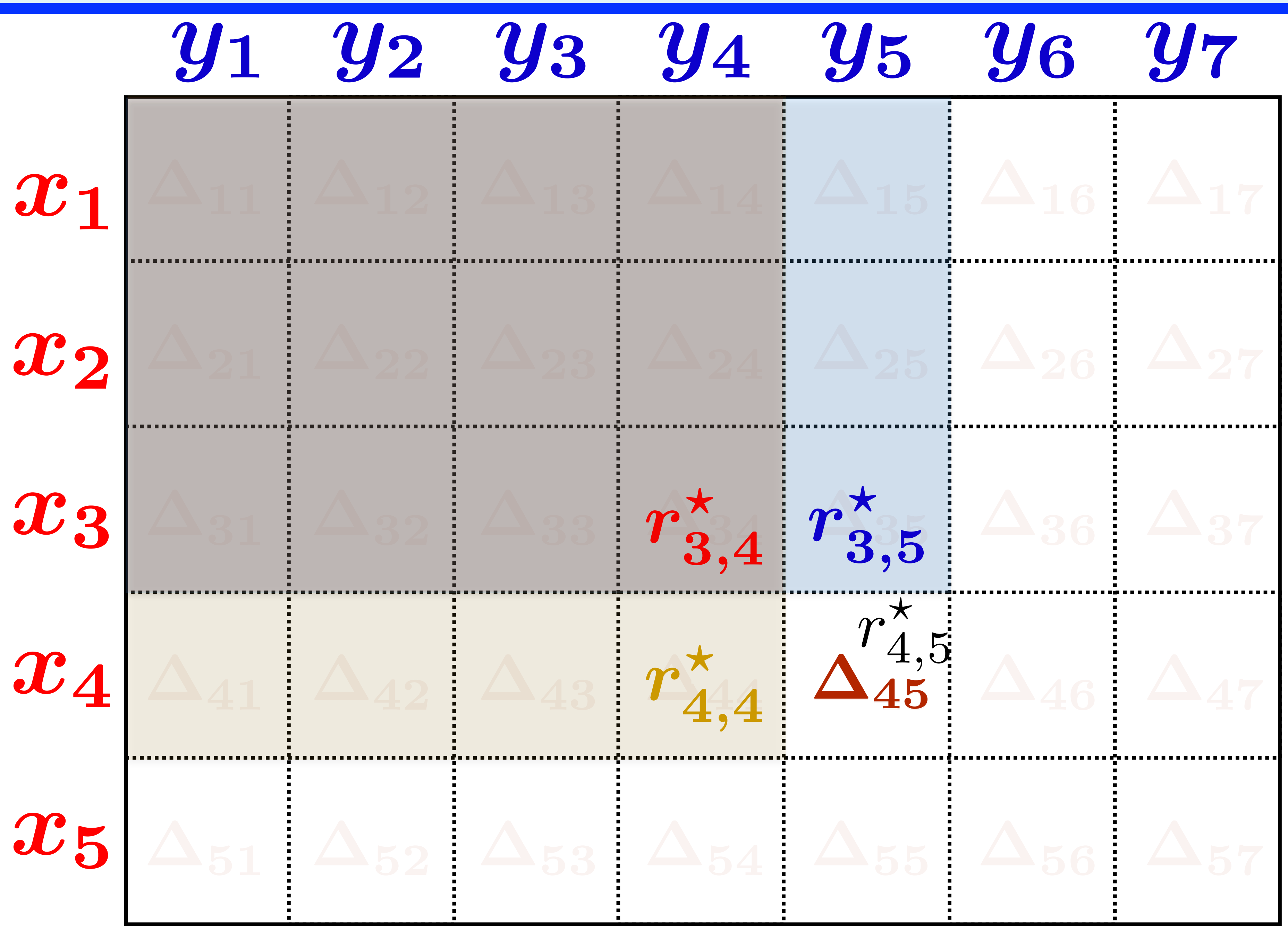
Best Path: Bellman Recursion



Best Path: Bellman Recursion

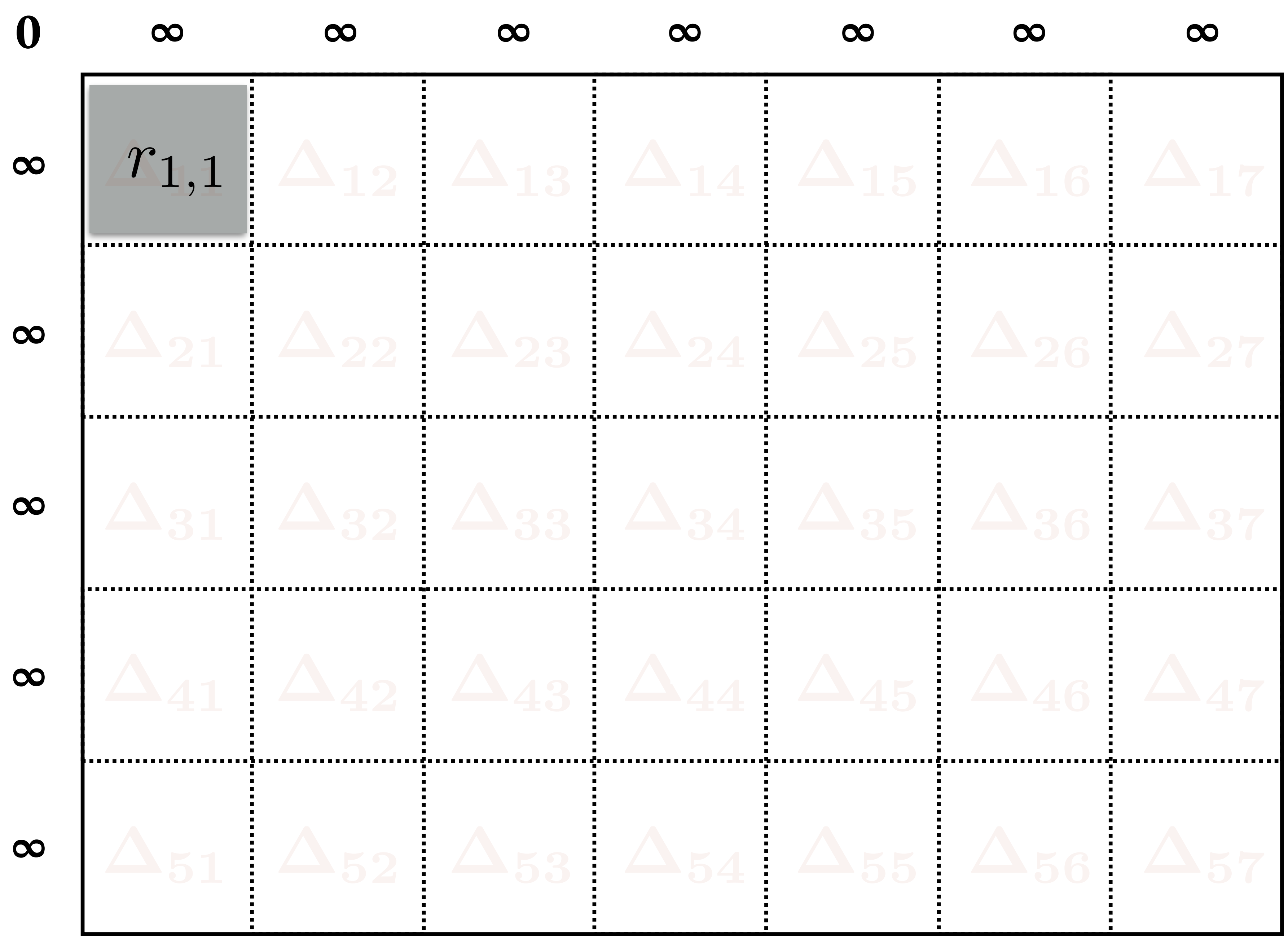


Best Path: Bellman Recursion



$$r^\star_{4,5} = \min(r^\star_{3,5}, r^\star_{4,4}, r^\star_{3,4}) + \Delta_{4,5}$$

Best Path: Bellman Recursion



$$r_{1,1} = \Delta_{11} \qquad r_{0,j} = r_{i,0} = \infty$$

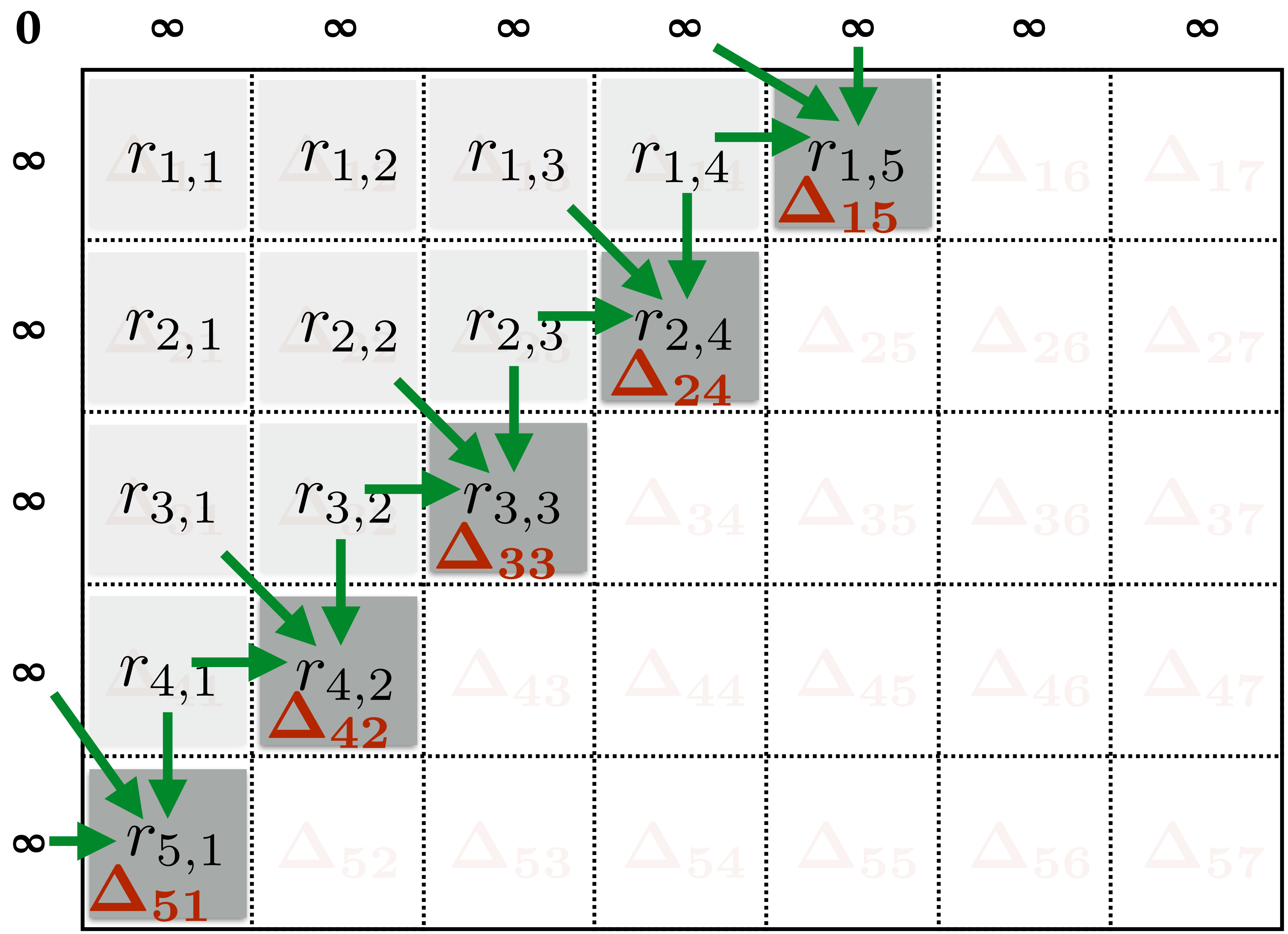
$$r_{0,0} = 0$$

Best Path: Bellman Recursion

0	∞	∞	∞	∞	∞	∞	∞
∞	$r_{1,1}$	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}	Δ_{16}	Δ_{17}
∞	Δ_{21}	Δ_{22}	Δ_{23}	Δ_{24}	Δ_{25}	Δ_{26}	Δ_{27}
∞	Δ_{31}	Δ_{32}	Δ_{33}	Δ_{34}	Δ_{35}	Δ_{36}	Δ_{37}
∞	Δ_{41}	Δ_{42}	Δ_{43}	Δ_{44}	Δ_{45}	Δ_{46}	Δ_{47}
∞	Δ_{51}	Δ_{52}	Δ_{53}	Δ_{54}	Δ_{55}	Δ_{56}	Δ_{57}

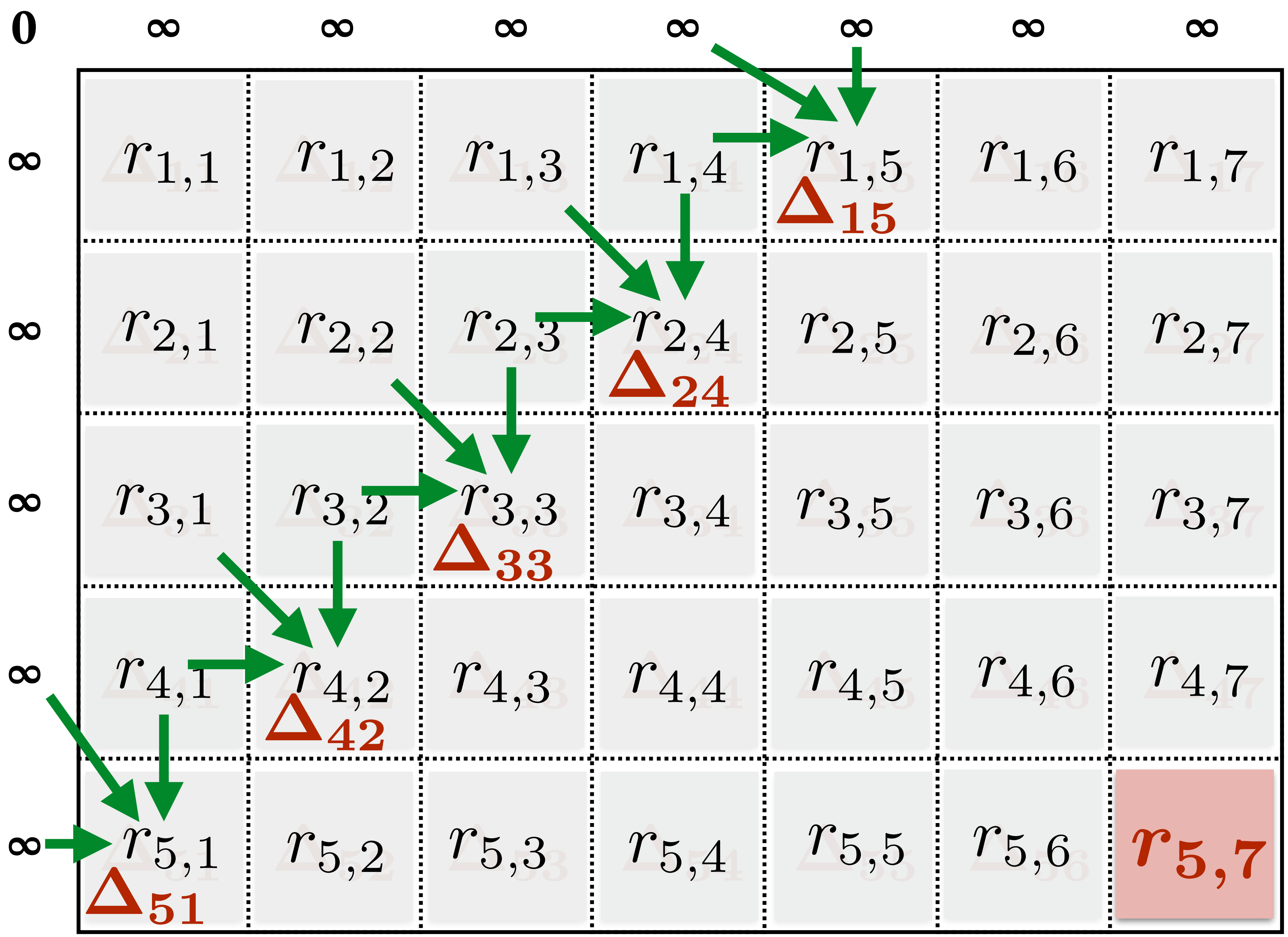
$$r_{i,j} = \min(r_{i-1,j-1}\,,\ r_{i-1,j}\,,\ r_{i,j-1}) + \boldsymbol{\Delta_{i,j}}$$

Best Path: Bellman Recursion



$$r_{i,j} = \min(r_{i-1,j-1}\,,\ r_{i-1,j}\,,\ r_{i,j-1}) + \boldsymbol{\Delta_{i,j}}$$

Best Path: Bellman Recursion



$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{r_{n,m}}$$

Optimal Path

	0	∞	∞	∞	∞	∞	∞	∞
	∞	1	0	0	0	0	0	0
	∞	0	1	0	0	0	0	0
$A^\star$	∞	0	0	1	1	0	0	0
	∞	0	0	0	0	1	0	0
	∞	0	0	0	0	0	1	1

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{r_{n,m}} = \langle A^\star, \boldsymbol{\Delta} \rangle$$

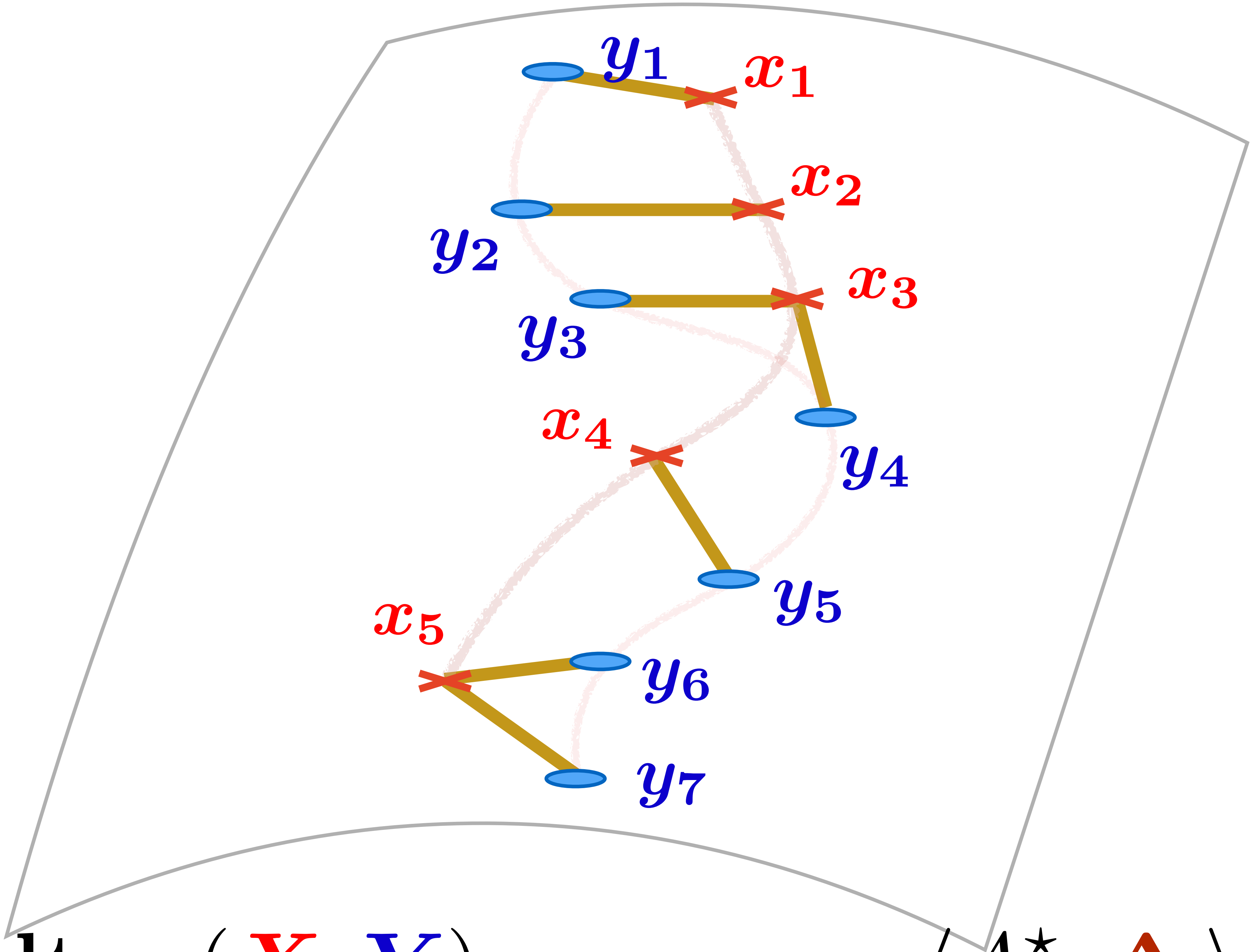
Computational cost: O(nm)

0. The DTW Geometry

1. Soft-DTW

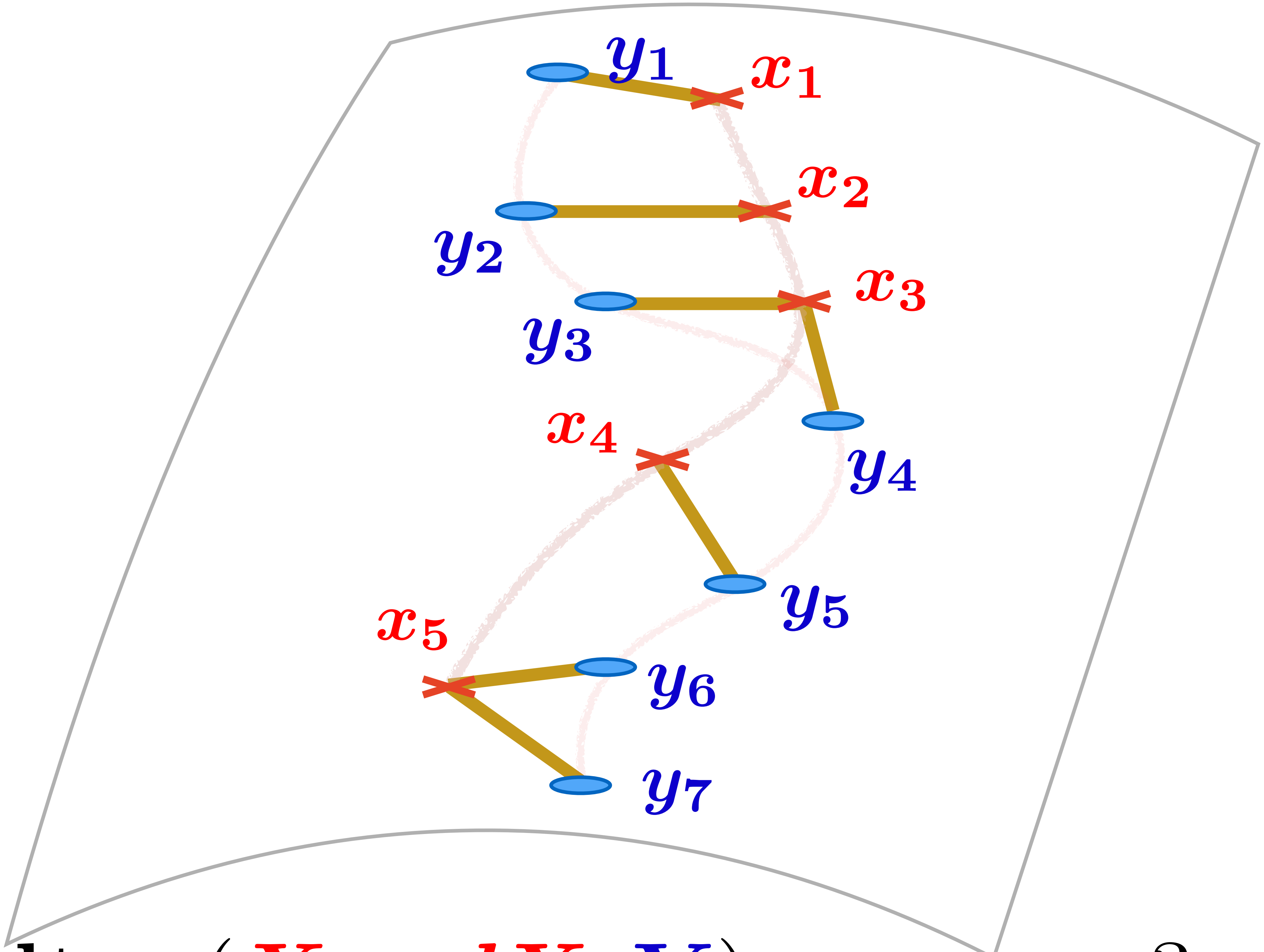
2. Soft-DTW as a Loss Function

DTW as a Loss: Differentiability?



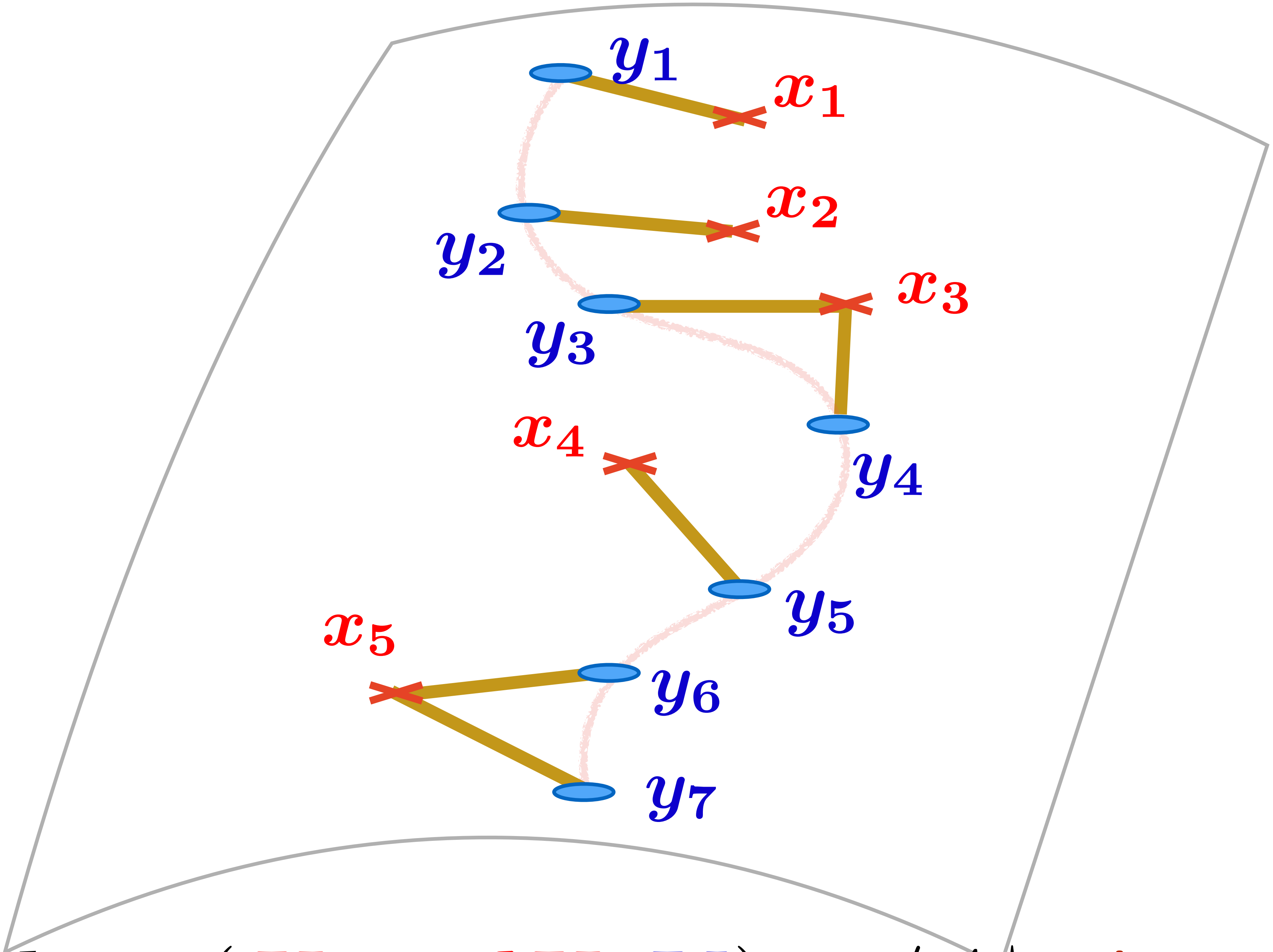
$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{r_{n,m}} = \langle A^\star, \boldsymbol{\Delta} \rangle$$

DTW as a Loss: Differentiability?



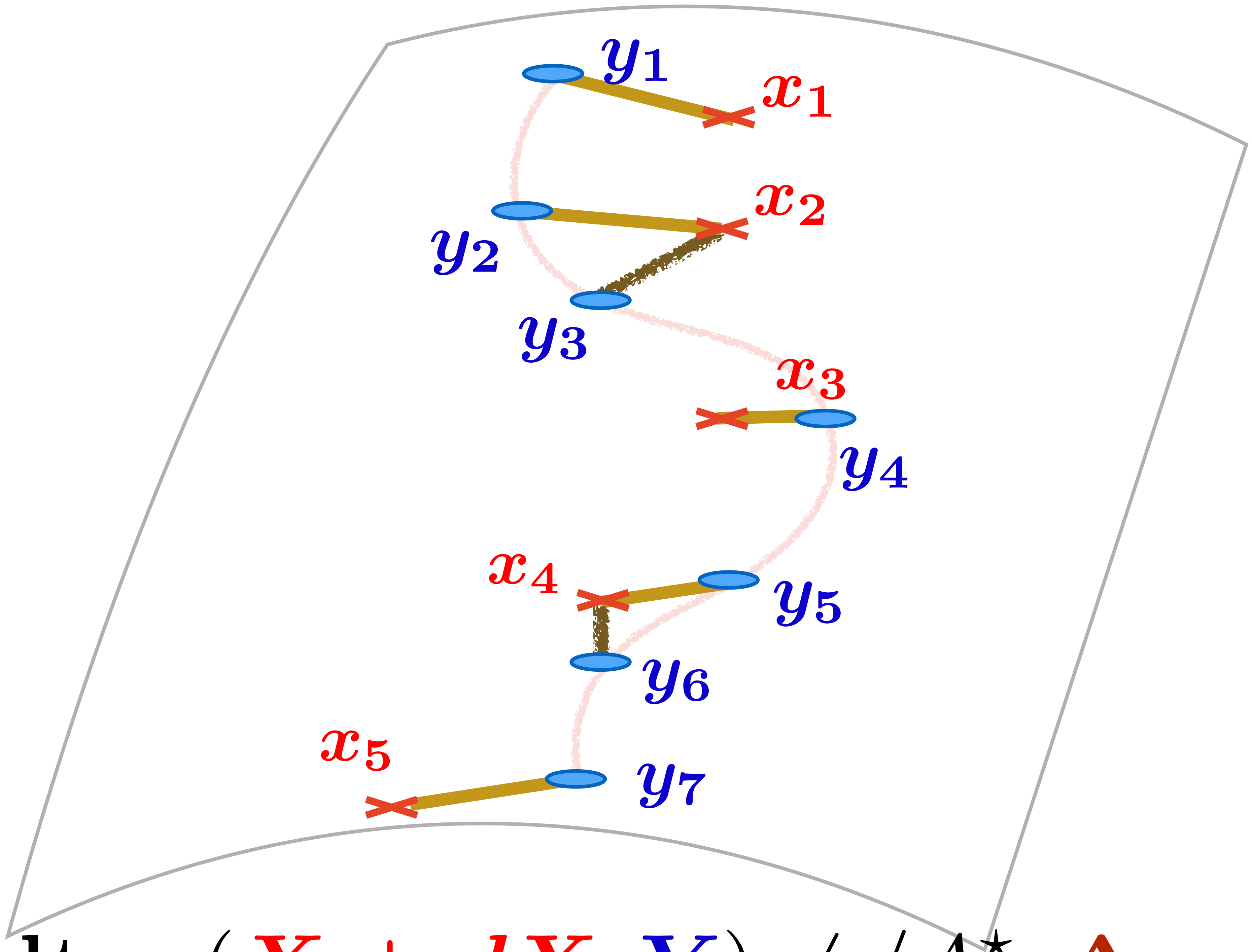
$$\mathbf{dtw}_0(\boldsymbol{X} + \boldsymbol{dX}, \boldsymbol{Y}) = \boldsymbol{r_{n,m}} + ?$$

DTW as a Loss: Differentiability?



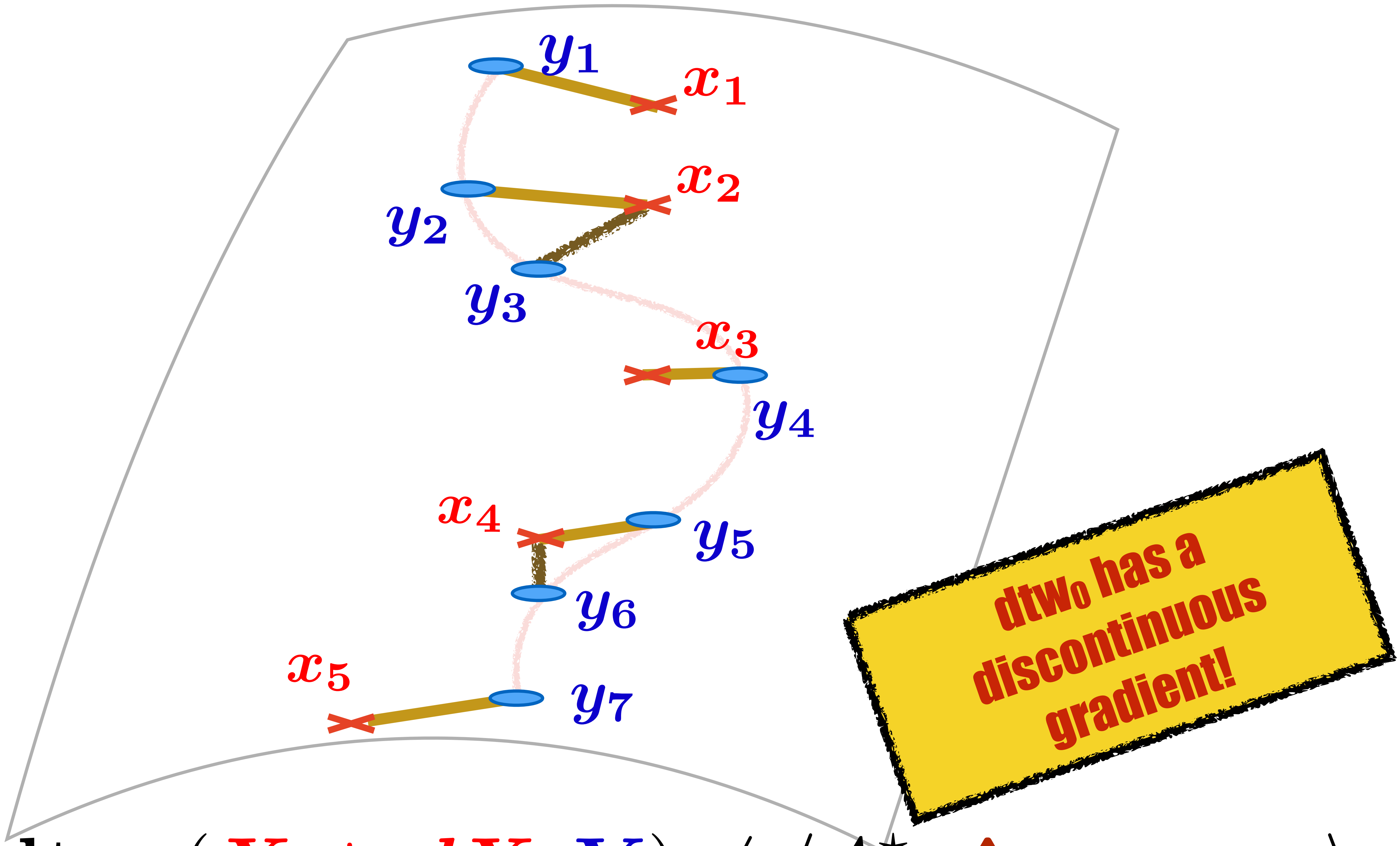
$$\mathbf{dtw}_0(\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y}) = \langle A^\star, \boldsymbol{\Delta}_{\boldsymbol{X}+d\boldsymbol{X}\boldsymbol{Y}} \rangle$$

DTW as a Loss: Differentiability?



$$\mathbf{dtw}_0(\boldsymbol{X} + \boldsymbol{dX}, \boldsymbol{Y}) \neq \langle A^\star, \boldsymbol{\Delta}_{\boldsymbol{X}+\boldsymbol{dX}\boldsymbol{Y}} \rangle$$

DTW as a Loss: Differentiability?

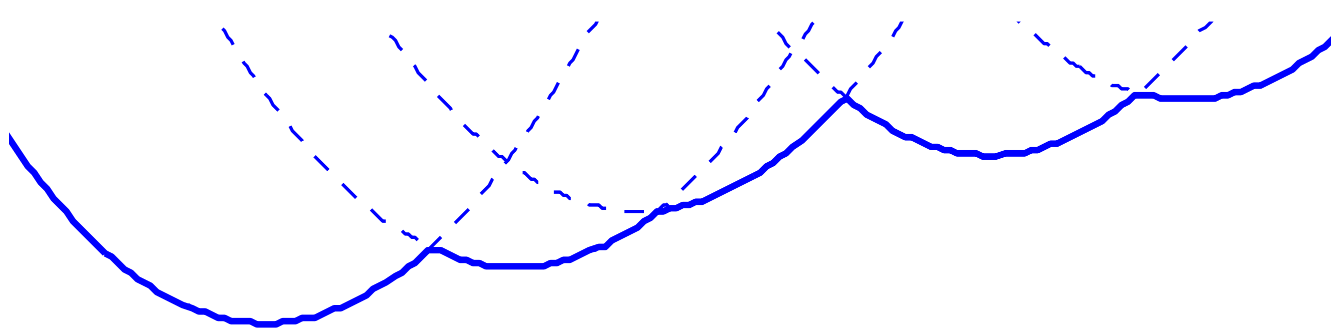


$$\mathbf{dtw}_0(\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y}) \neq \langle A^\star, \boldsymbol{\Delta}_{\boldsymbol{X}+d\boldsymbol{X}\boldsymbol{Y}} \rangle$$

DTW as a Loss: Differentiability?

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

- $\mathbf{dtw}_0$ is piecewise **linear** w.r.t $\boldsymbol{\Delta}$
- if $\boldsymbol{\Delta}_{ij} = \boldsymbol{\delta}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2$, $\mathbf{dtw}_0$ is piecewise **quadratic** w.r.t. $\boldsymbol{X}$.



DTW as a Loss: Differentiability?

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_{\boldsymbol{X}} \mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}}}{\partial \boldsymbol{X}}\right)^T \nabla_{\boldsymbol{\Delta}} \min_{\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle \cdot, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

DTW as a Loss: Differentiability?

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_{\boldsymbol{X}} \mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}}}{\partial \boldsymbol{X}} \right)^T \nabla_{\boldsymbol{\Delta}} \min_{\mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle \cdot, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Jacobian matrix of **Δ w.r.t. X**

DTW as a Loss: Differentiability?

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_{\boldsymbol{X}} \mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}}}{\partial \boldsymbol{X}}\right)^T \nabla_{\boldsymbol{\Delta}} \min_{\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle \cdot, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$= A^\star$$

Jacobian matrix of **Δ w.r.t. X**

iff optimal solution is unique

DTW as a Loss: Differentiability?

$$\mathbf{dtw}_0(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_{\boldsymbol{X}}\, \mathbf{dtw}_0(\boldsymbol{X},\boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}}}{\partial \boldsymbol{X}}\right)^T \nabla_{\boldsymbol{\Delta}} \min_{\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle \cdot, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$= A^\star$$

Jacobian matrix of **Δ w.r.t. X**

iff optimal solution is unique

When $A^\star$ is not unique, $\mathbf{dtw}_0$ has a discontinuous gradient!

Our proposal: smoothing the min

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Problem: non-differentiability of min operator over finite family of values.

Our proposal: smoothing the min

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

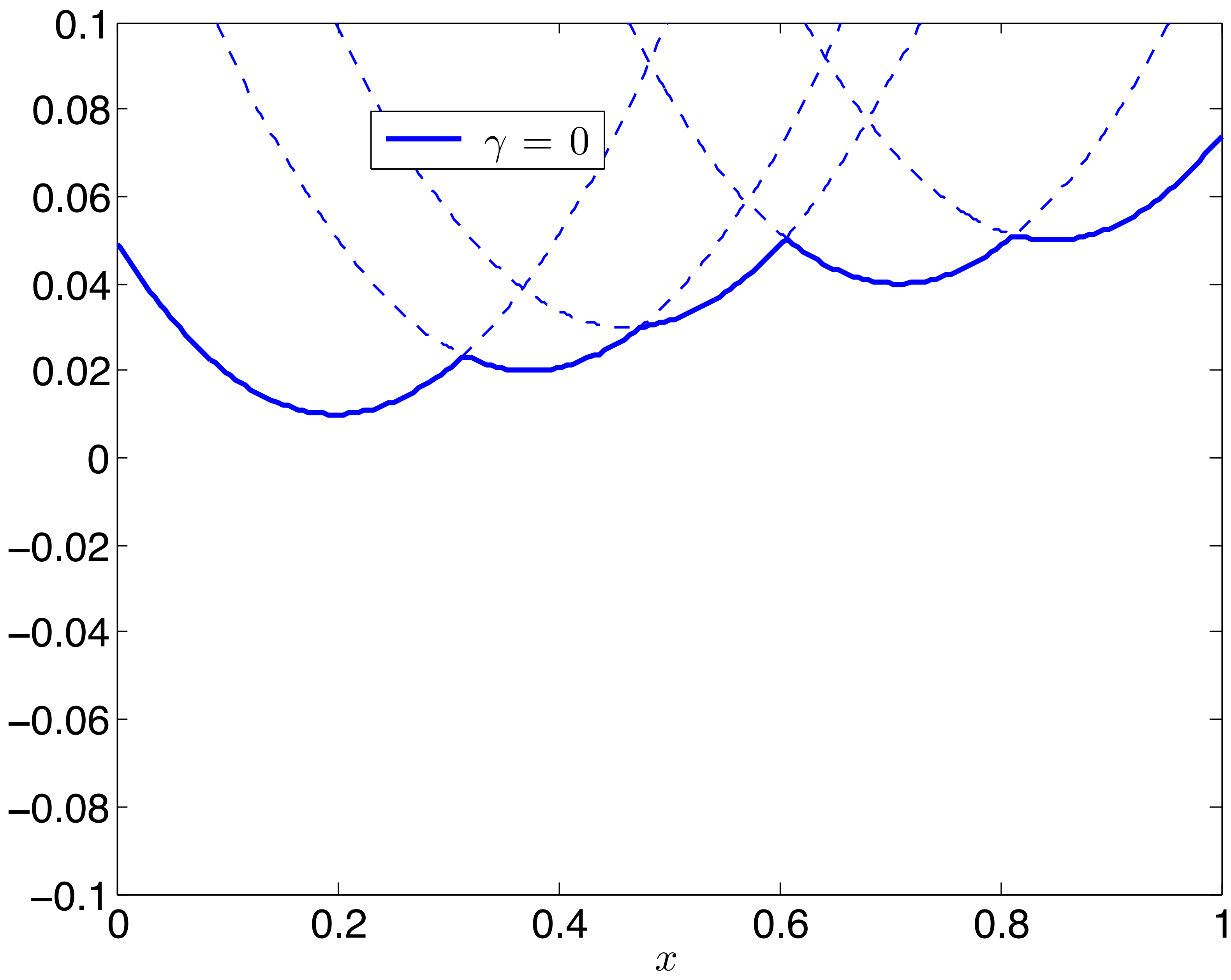
Problem: non-differentiability of min operator over finite family of values.

Fix: smoothed min operator

$$\min{}^{\gamma}(u_1, \ldots, u_n) = \begin{cases} \min_{i \leq n} u_i, & \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} e^{-u_i/\gamma}, & \gamma > 0. \end{cases}$$

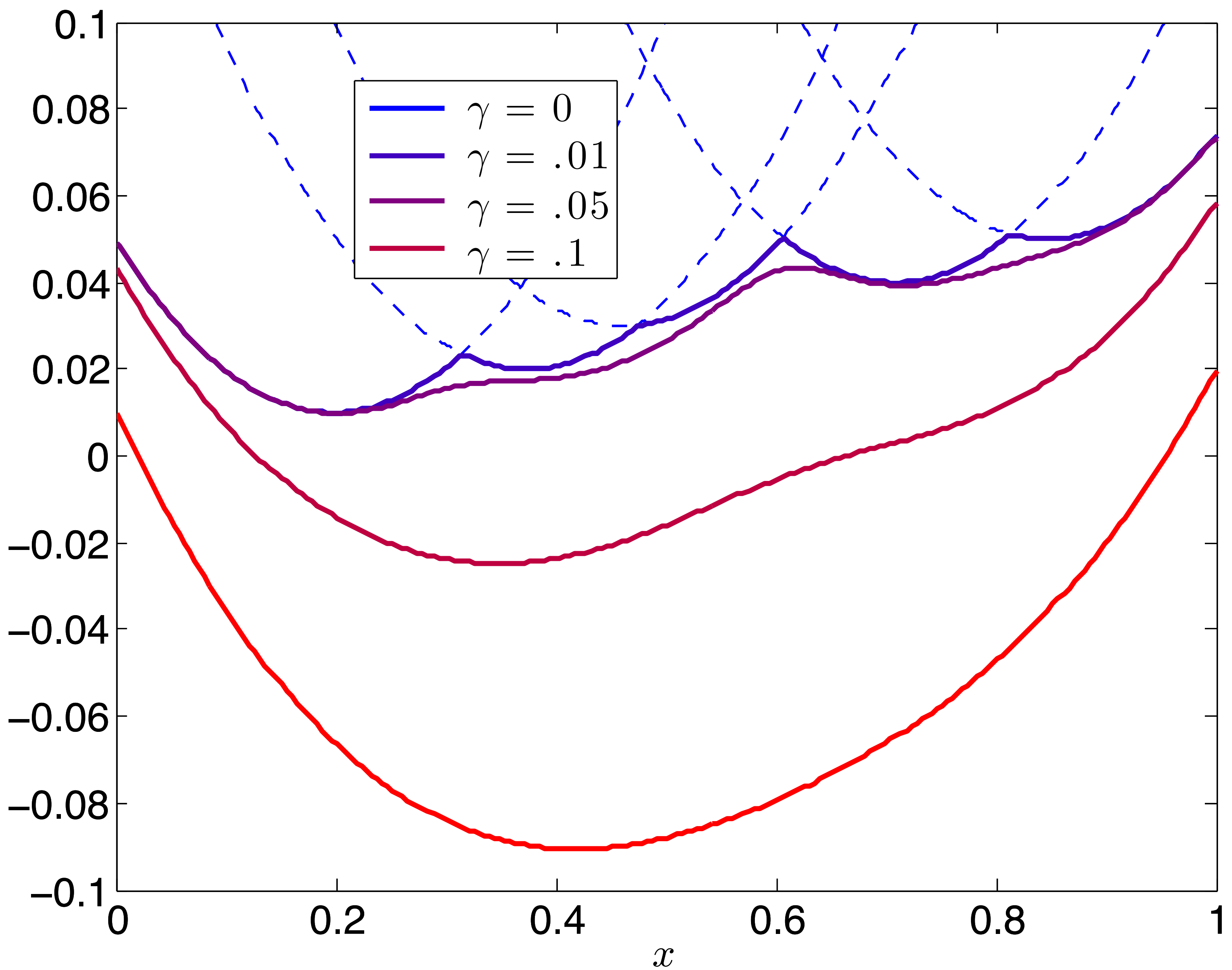
Example softmin of quadratic functions

$$f(\boldsymbol{x}) = \min_{i=1,\ldots,s}{}^{\gamma}\; a_i \boldsymbol{x}^{\mathbf{2}} + b_i \boldsymbol{x} + c_i$$



Example softmin of quadratic functions

$$f(\boldsymbol{x}) = \min_{i=1,\ldots,s}{}^{\gamma} \; a_i \boldsymbol{x}^{\mathbf{2}} + b_i \boldsymbol{x} + c_i$$



Soft-DTW

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Soft-DTW

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Fix: Replace min by $\min^{\gamma}, \gamma > 0$

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Soft-DTW

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Fix: Replace min by $\min^{\gamma}, \gamma > 0$

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})}{}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = -\gamma \log \sum_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}$$

Relation to Global Alignment kernels

$$k_{\mathrm{GA}} := \sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}$$

CVBM'07

A positive semi-definite **kernel** between time series

Relation to Global Alignment kernels

$$k_{\mathrm{GA}} := \sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}$$

CVBM'07

A positive semi-definite **kernel** between time series

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = -\gamma \log k_{\mathrm{GA}}$$

Computing soft-DTW is equivalent to computing k_{GA} in **log domain**

Recursive Computation

$$\mathbf{dtw}_0(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$r_{i,j} = \min(r_{i-1,j-1}\,,\, r_{i-1,j}\,,\, r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i},\boldsymbol{j}}$$

Recursive Computation

$$\mathbf{dtw}_0(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle$$

$$r_{i,j} = \min(r_{i-1,j-1}\,,\; r_{i-1,j}\,,\; r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i,j}}$$

$$\mathbf{dtw}_\gamma(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})}{}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle$$

$$r_{i,j} = \min{}^{\gamma}(r_{i-1,j-1}\,,\; r_{i-1,j}\,,\; r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i,j}}$$

Recursive Computation

$$\mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle$$

$$r_{i,j} = \min(r_{i-1,j-1}\,,\ r_{i-1,j}\,,\ r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i,j}}$$

$$\mathbf{dtw}_\gamma(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle$$

$$r_{i,j} = \min^{\gamma}(r_{i-1,j-1}\,,\ r_{i-1,j}\,,\ r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i,j}}$$

Simply replace min operator!

Recursive Computation

$$\mathbf{dtw}_0(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$r_{i,j} = \min(r_{i-1,j-1}\,,\; r_{i-1,j}\,,\; r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i},\boldsymbol{j}}$$

$$\mathbf{dtw}_\gamma(\boldsymbol{X},\boldsymbol{Y}) = \min^{\gamma}_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$r_{i,j} = \min{}^{\gamma}(r_{i-1,j-1}\,,\; r_{i-1,j}\,,\; r_{i,j-1}) + \boldsymbol{\Delta}_{\boldsymbol{i},\boldsymbol{j}}$$

Simply replace min operator!

Stable: recursion in log domain!

Differentiation

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n}, \boldsymbol{m})}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_X \mathbf{dtw}_0(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}} \right)^T A^\star$$

Differentiation

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})}{}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_X \, \mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}} \right)^T \mathbb{E}_{\gamma}[A]$$

Differentiation

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})}{}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_X \, \mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}} \right)^T \mathbb{E}_{\gamma}[A]$$

$$\mathbb{E}_{\gamma}[A] := \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}$$

Expectation of Path under Gibbs distribution

Differentiation

$$\mathbf{dtw}_{\gamma}(\boldsymbol{X},\boldsymbol{Y}) = \min_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})}{}^{\gamma} \langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\nabla_X \, \mathbf{dtw}_{\gamma}(\boldsymbol{X},\boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{X},\boldsymbol{Y})}{\partial \boldsymbol{X}}\right)^T \underbrace{\mathbb{E}_{\gamma}[A]}_{\nabla_{\Delta}\,\mathbf{dtw}_{\gamma}(\boldsymbol{X},\boldsymbol{Y})}$$

$$\mathbb{E}_{\gamma}[A] := \frac{\sum_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}{\sum_{A\in\mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}$$

Expectation of Path under Gibbs distribution

Computing the expectation $E_\gamma[A]$

$$\mathbb{E}_{\gamma}[A] := \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}$$

Naive computation
is intractable

Computing the expectation $E_\gamma[A]$

$$\mathbb{E}_\gamma[A] := \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}$$

Naive computation is intractable

$$= \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{X}\boldsymbol{Y}} \rangle}{\gamma}}}{k_{\mathrm{GA}}}$$

k_{GA} is the normalization constant (a.k.a. partition function)!

Computing the expectation $E_\gamma[A]$

$$\mathbb{E}_\gamma[A] := \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle}{\gamma}}}{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle}{\gamma}}}$$

Naive computation is intractable

$$= \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \boldsymbol{\Delta}_{\boldsymbol{XY}} \rangle}{\gamma}}}{k_{\mathrm{GA}}}$$

k_{GA} is the normalization constant (a.k.a. partition function)!

$$= \nabla_{\Delta} -\gamma \log k_{GA}$$

$E_\gamma[A]$ is the gradient of the log partition

Computing the expectation $E_\gamma[A]$

$$\mathbb{E}_\gamma[A] := \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \Delta_{XY} \rangle}{\gamma}}}{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} e^{-\frac{\langle A, \Delta_{XY} \rangle}{\gamma}}}$$

Naive computation is intractable

$$= \frac{\sum_{A \in \mathcal{A}(\boldsymbol{n},\boldsymbol{m})} A e^{-\frac{\langle A, \Delta_{XY} \rangle}{\gamma}}}{k_{\mathrm{GA}}}$$

k_{GA} is the normalization constant (a.k.a. partition fu on)!

$$= \nabla_\Delta -\gamma \log k_{GA}$$

nt

g partition

Classical result of exponential families

Computing the expectation $E_\gamma[A]$

To summarize, we want to compute:

$$E_\gamma[A] = \nabla_\Delta -\gamma \log k_{GA}$$

Computing the expectation $E_\gamma[A]$

To summarize, we want to compute:

$$E_\gamma[A] = \nabla_\Delta -\gamma \log k_{GA} = \nabla_\Delta \mathrm{dtw}_\gamma$$

Computing the expectation $E_\gamma[A]$

To summarize, we want to compute:

$$E_\gamma[A] = \nabla_\Delta -\gamma \log k_{GA} = \nabla_\Delta \, dtw_\gamma$$

$E_\gamma[A]$ can be computed by **backpropagation** in the same O(nm) cost as dtw_γ

Computing the expectation $E_\gamma[A]$

To summarize, we want to compute:

$$E_\gamma[A] = \nabla_\Delta -\gamma \log k_{GA} = \nabla_\Delta \, dtw_\gamma$$

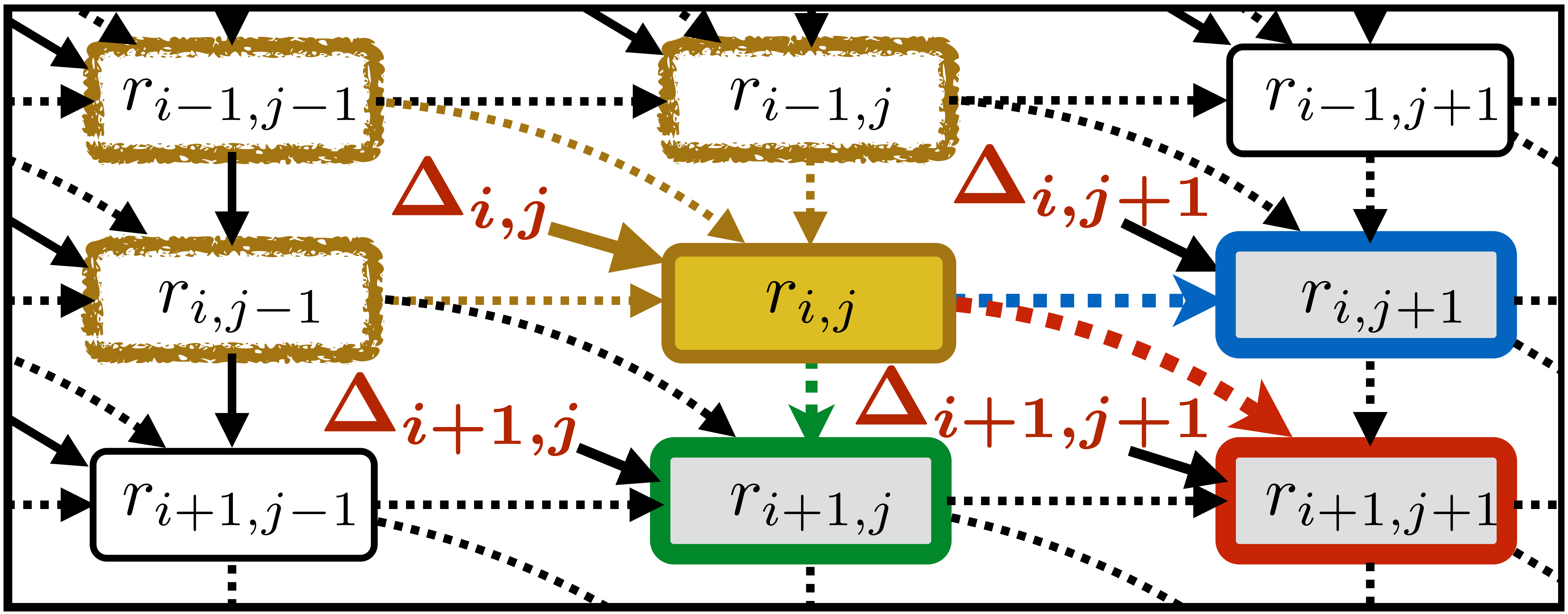
$E_\gamma[A]$ can be computed by **backpropagation** in the same O(nm) cost as dtw_γ

We derive a backward recursion **without resorting to autodiff**

Faster and more numerically stable

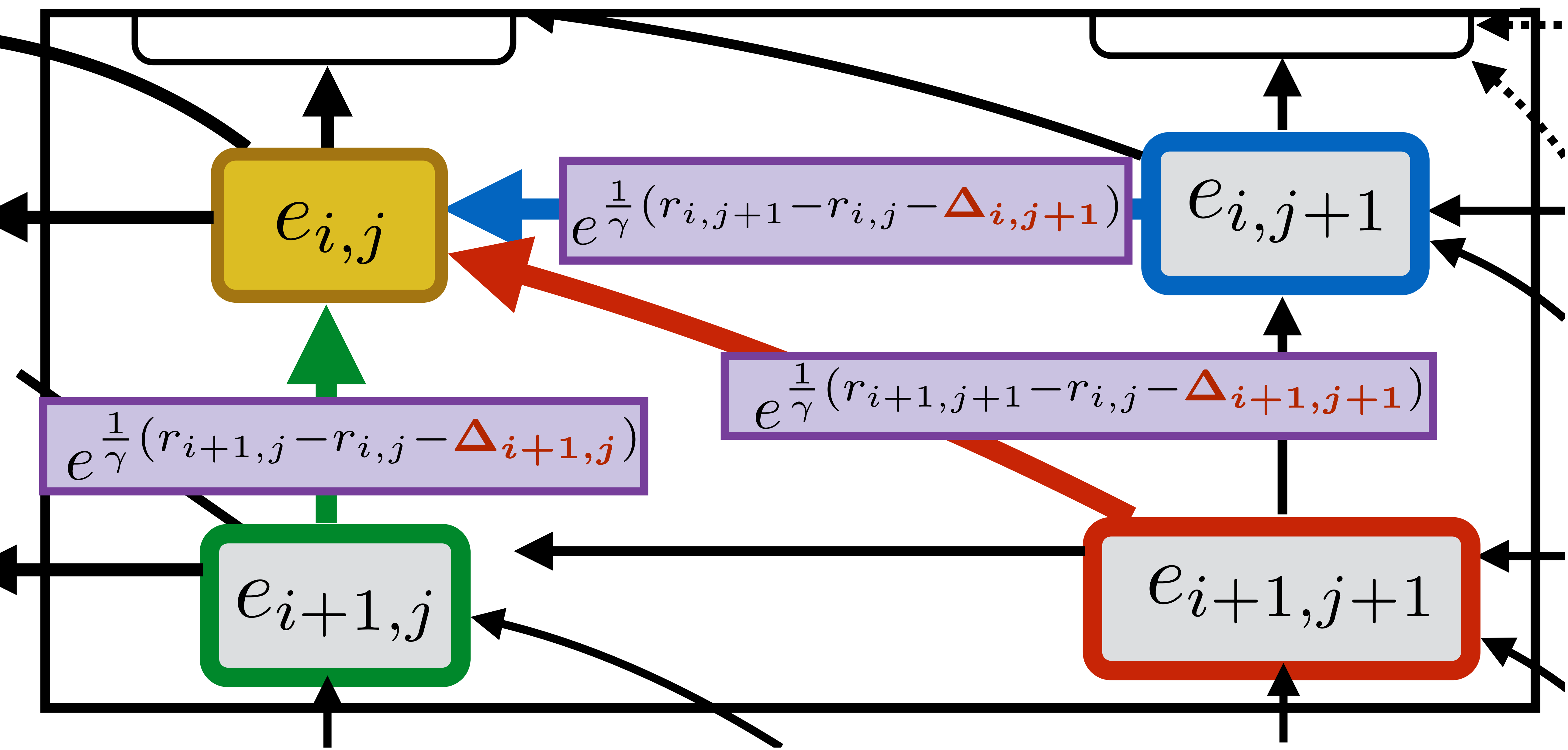
Forward Pass

Bellman's recursion has the following computational graph

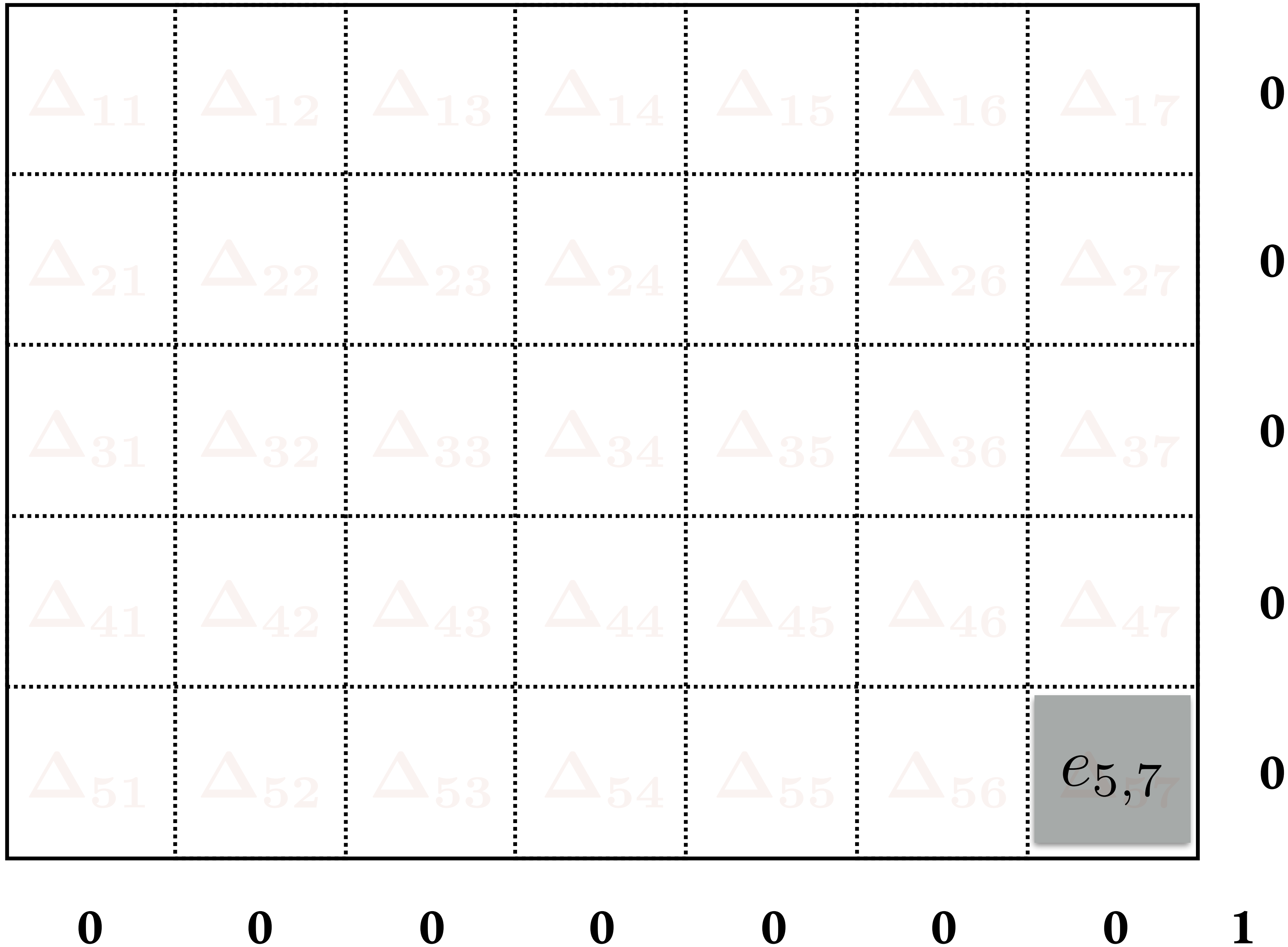


Backward Pass

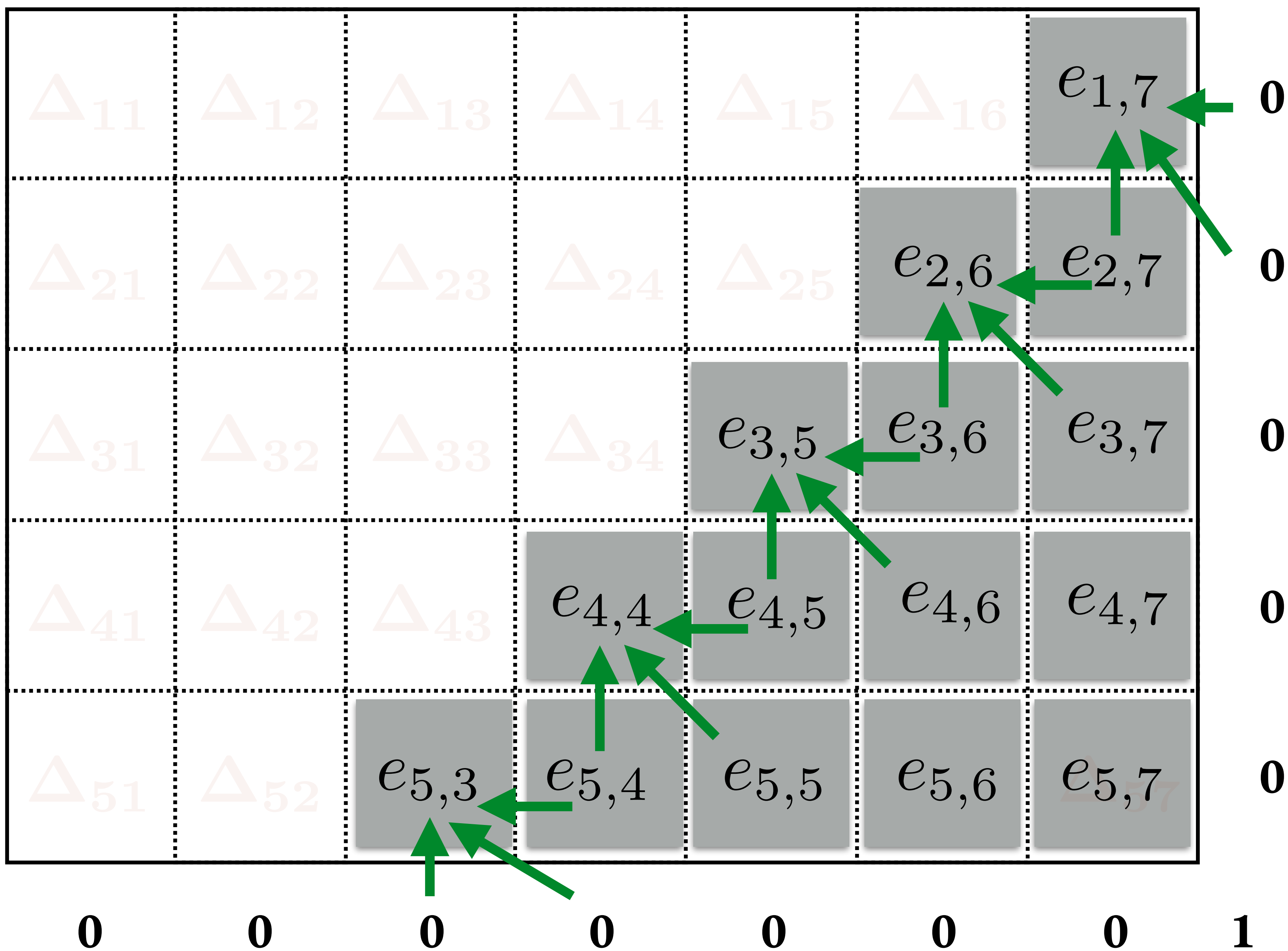
with a few simplifications, the backward pass boils down to the following updates



Backward Recursion



Backward Recursion



Backward Recursion

$e_{1,1}$	$e_{1,2}$	$e_{1,3}$	$e_{1,4}$	$e_{1,5}$	$e_{1,6}$	$e_{1,7}$	**0**
$e_{2,1}$	$e_{2,2}$	$e_{2,3}$	$e_{2,4}$	$e_{2,5}$	$e_{2,6}$	$e_{2,7}$	**0**
$e_{3,1}$	$e_{3,2}$	$e_{3,3}$	$e_{3,4}$	$e_{3,5}$	$e_{3,6}$	$e_{3,7}$	**0**
$e_{4,1}$	$e_{4,2}$	$e_{4,3}$	$e_{4,4}$	$e_{4,5}$	$e_{4,6}$	$e_{4,7}$	**0**
$e_{5,1}$	$e_{5,2}$	$e_{5,3}$	$e_{5,4}$	$e_{5,5}$	$e_{5,6}$	$e_{5,7}$	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**

$$E_{\gamma}[A] = [e]_{ij}$$

Backward Recursion

$$a = e^{\frac{1}{\gamma}(r_{i+1,j} - r_{i,j} - \boldsymbol{\Delta}_{i+1,j})}$$

$$b = e^{\frac{1}{\gamma}(r_{i,j+1} - r_{i,j} - \boldsymbol{\Delta}_{i,j+1})}$$

$$c = e^{\frac{1}{\gamma}(r_{i+1,j+1} - r_{i,j} - \boldsymbol{\Delta}_{i+1,j+1})}$$

$$e_{i,j} = e_{i+1,j} \cdot a + e_{i,j+1} \cdot b + e_{i+1,j+1} \cdot c$$

$$\nabla_X \mathbf{dtw}_\gamma(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{\partial \boldsymbol{\Delta}(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}} \right)^T E$$

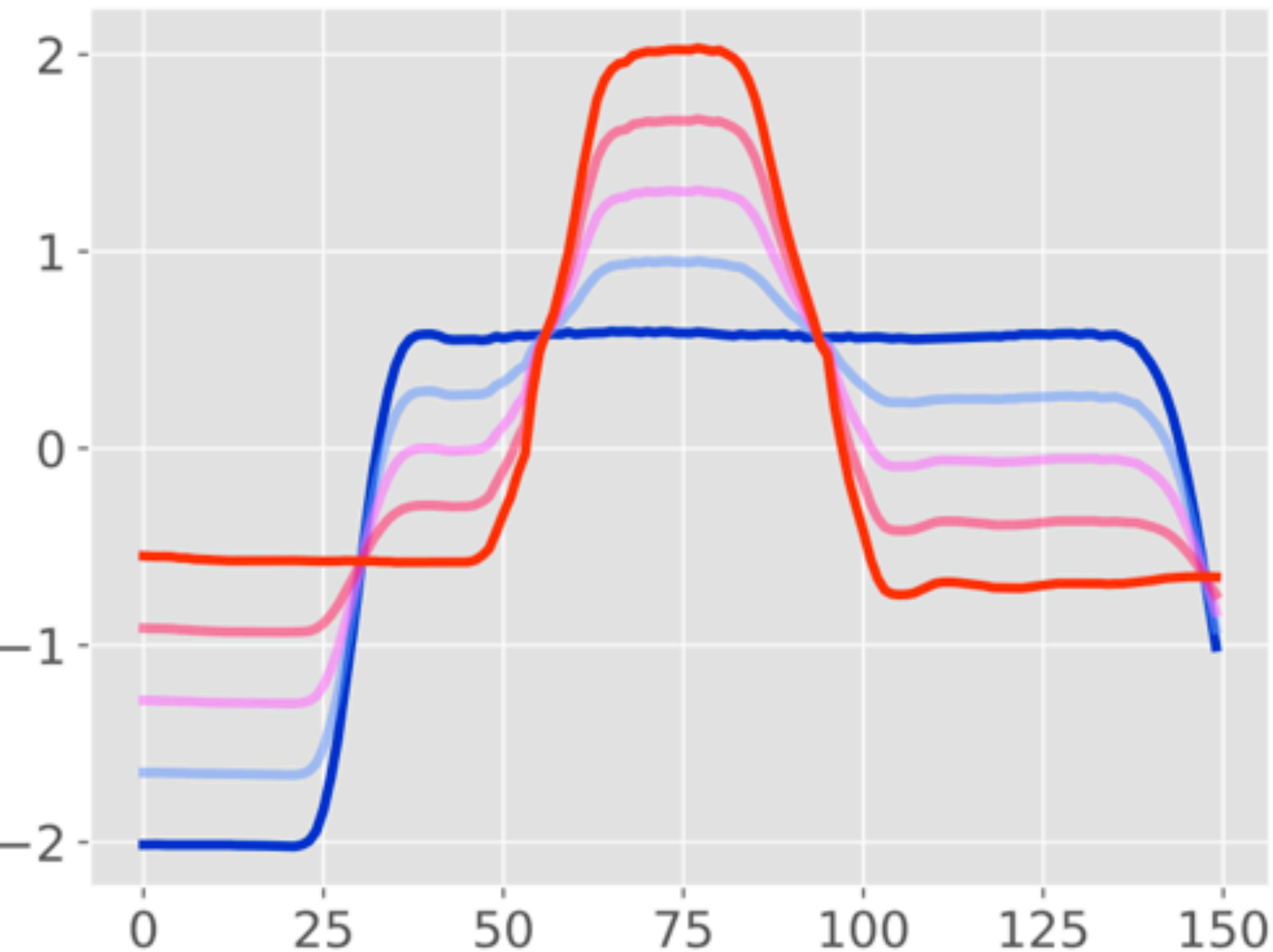
0. The DTW Geometry

1. Soft-DTW

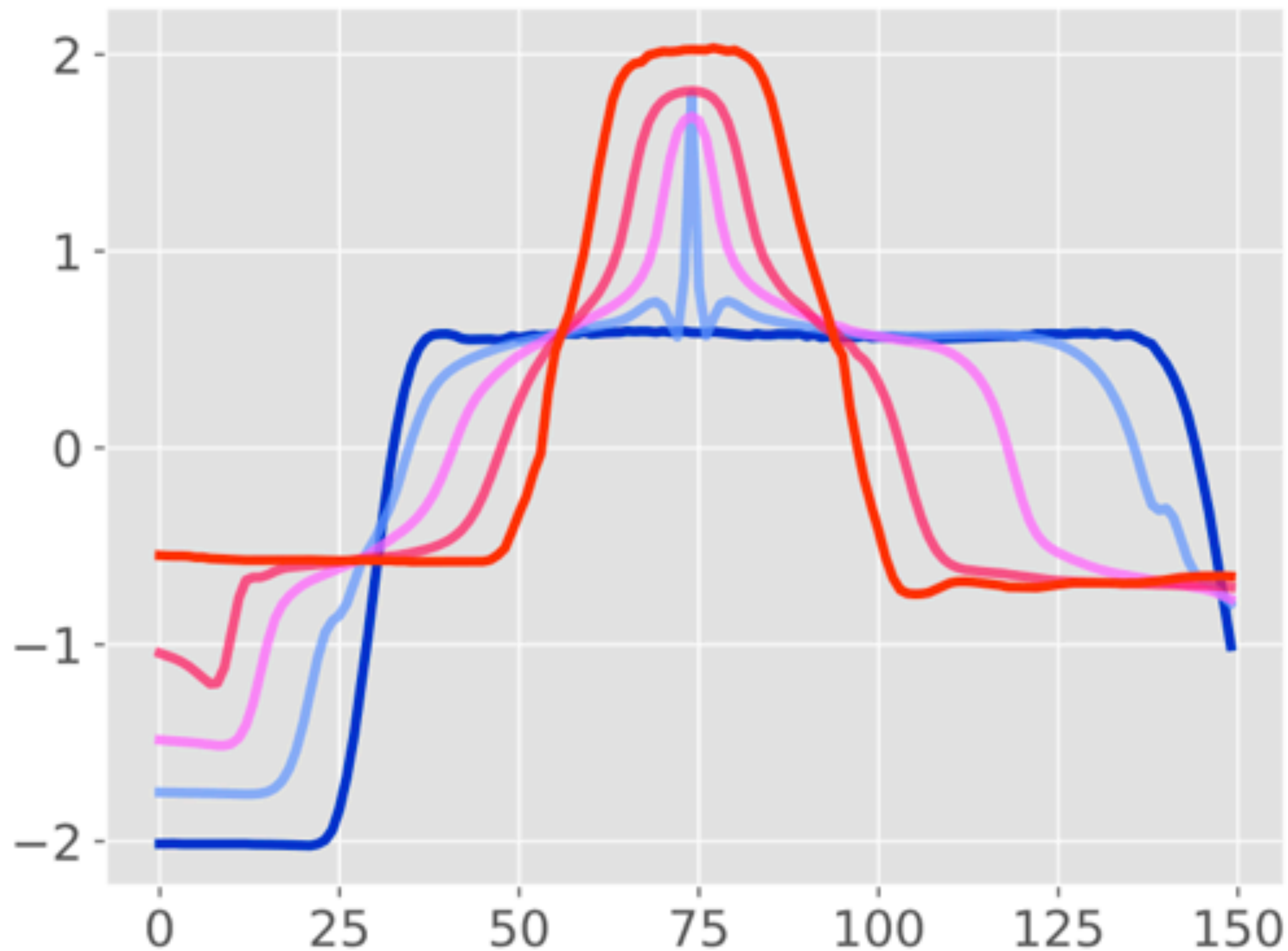
2. Soft-DTW as a Loss Function

Interpolation Between 2 Time Series

$$\min_{\mathbf{X}} \left[\lambda\, \mathbf{dtw}_{\gamma}(\mathbf{X}, \boldsymbol{Y_1}) + (1-\lambda)\, \mathbf{dtw}_{\gamma}(\mathbf{X}, \boldsymbol{Y_2})\right]$$



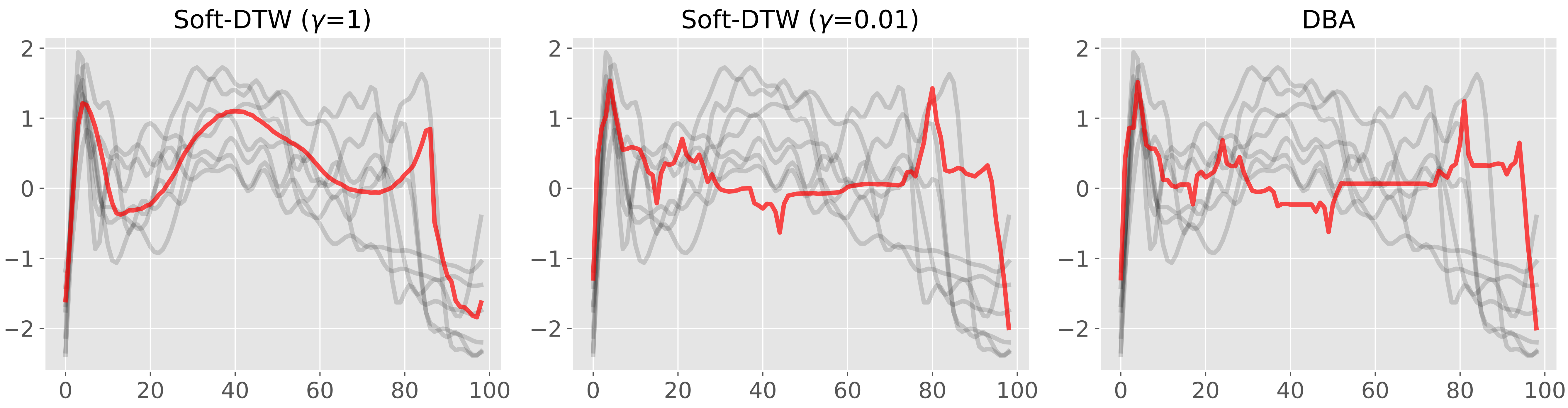
Euclidean loss



Soft-DTW loss ($\gamma = 1$)

sDTW Barycenter

$$\min_{\boldsymbol{X}} \sum_{j=1} \frac{\lambda_j}{m_j} \mathbf{dtw}_\gamma(\boldsymbol{X}, \boldsymbol{Y_j})$$



[DBA] Petitjean et al., A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44 (3):678–693, 2011.

sDTW Barycenter

$$\min_{\boldsymbol{X}} \sum_{j=1} \frac{\lambda_j}{m_j} \mathbf{dtw}_\gamma(\boldsymbol{X}, \boldsymbol{Y_j})$$

Table 1. Percentage of the datasets on which the proposed soft-DTW barycenter is achieving lower DTW loss (Equation (4) with $\gamma = 0$) than competing methods.

	Random initialization	Euclidean mean initialization
Comparison with DBA		
$\gamma = 1$	40.51%	3.80%
$\gamma = 0.1$	93.67%	46.83%
$\gamma = 0.01$	100%	79.75%
$\gamma = 0.001$	97.47%	89.87%
Comparison with subgradient method		
$\gamma = 1$	96.20%	35.44%
$\gamma = 0.1$	97.47%	72.15%
$\gamma = 0.01$	97.47%	92.41%
$\gamma = 0.001$	97.47%	97.47%

sDTW Barycenter

$$\min_{\boldsymbol{X}} \sum_{j=1} \frac{\lambda_j}{m_j} \mathbf{dtw}_{\gamma}(\boldsymbol{X}, \boldsymbol{Y_j})$$

Evaluation performed using $\mathbf{dtw}_0$

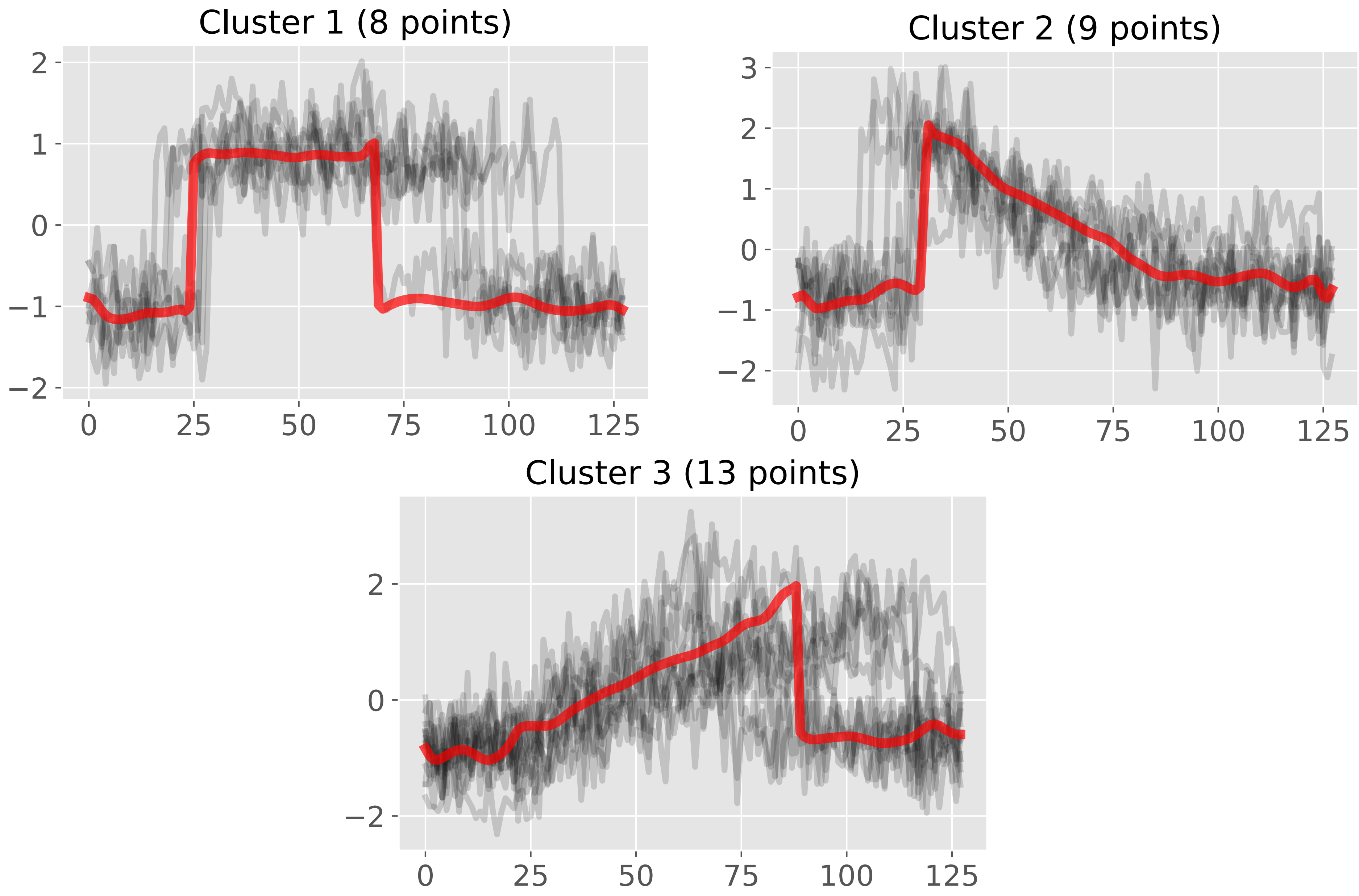
Table 1. Percentage of the datasets on which the propo
DTW barycenter is achieving lower DTW loss (Equation
$\gamma = 0$) than competing methods.

	Random initialization	Euclidean mean initialization
Comparison with DBA		
$\gamma = 1$	40.51%	3.80%
$\gamma = 0.1$	93.67%	46.83%
$\gamma = 0.01$	100%	79.75%
$\gamma = 0.001$	97.47%	89.87%
Comparison with subgradient method		
$\gamma = 1$	96.20%	35.44%
$\gamma = 0.1$	97.47%	72.15%
$\gamma = 0.01$	97.47%	92.41%
$\gamma = 0.001$	97.47%	97.47%

% of datasets where soft-dtw is winning

sDTW Clustering

$$\min_{\boldsymbol{X}_1,\ldots,\boldsymbol{X}_k} \sum_{j=1}^{N} \min_{\boldsymbol{i}=1,\ldots,k} \mathbf{dtw}_\gamma(\boldsymbol{X}_{\boldsymbol{i}}, \boldsymbol{Y}_{\boldsymbol{j}})$$



sDTW Clustering

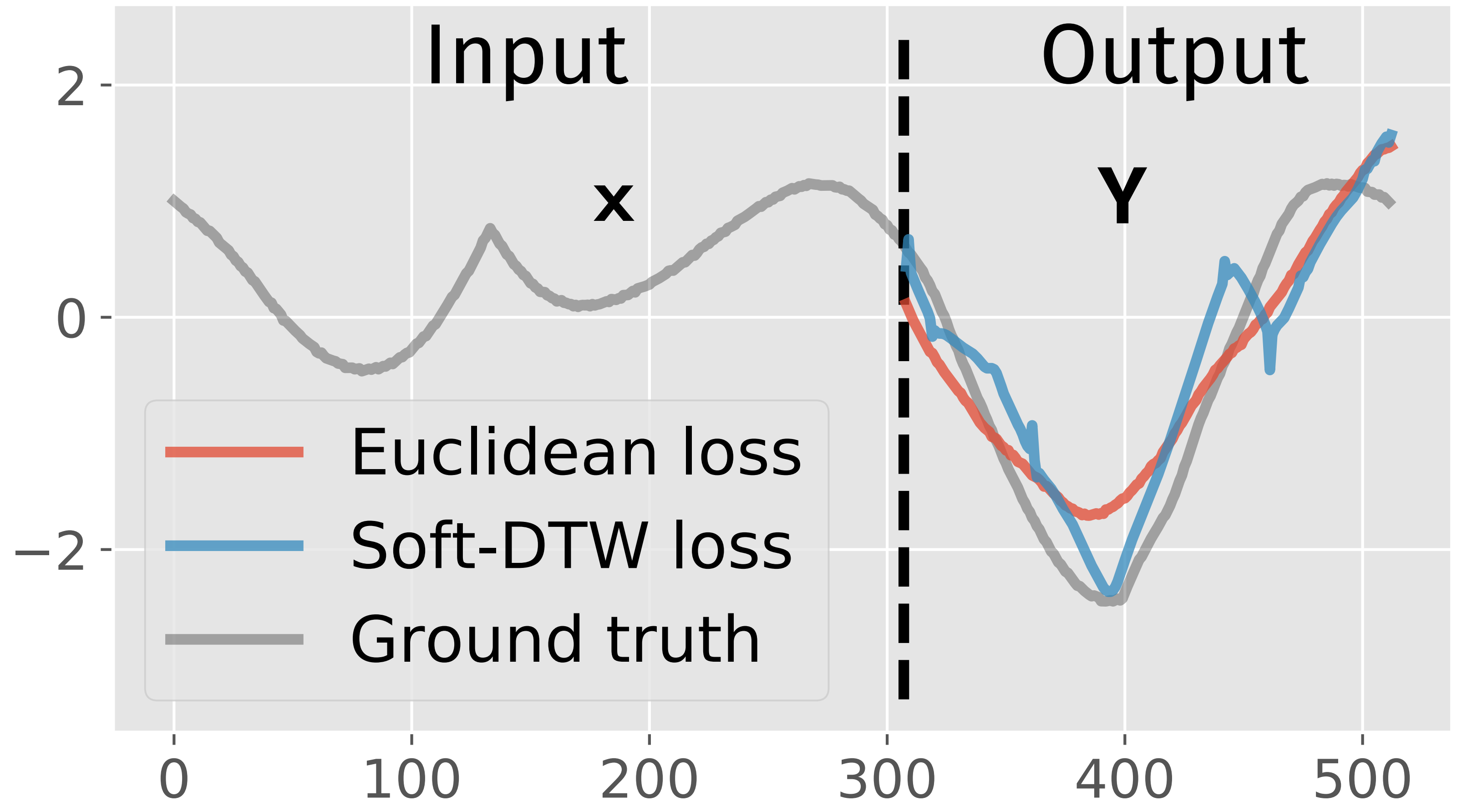
$$\min_{\boldsymbol{X_1},\ldots,\boldsymbol{X_k}} \sum_{j=1}^{N} \min_{\boldsymbol{i}=1,\ldots,k} \mathbf{dtw}_\gamma(\boldsymbol{X_i}, \boldsymbol{Y_j})$$

Table 2. Percentage of the datasets on which the proposed soft-DTW based k**-means** is achieving lower DTW loss (Equation (5) with $\gamma = 0$) than competing methods.

	Random initialization	Euclidean mean initialization
Comparison with DBA		
$\gamma = 1$	15.78%	29.31%
$\gamma = 0.1$	24.56%	24.13%
$\gamma = 0.01$	59.64%	55.17%
$\gamma = 0.001$	77.19%	68.97%
Comparison with subgradient method		
$\gamma = 1$	42.10%	46.44%
$\gamma = 0.1$	57.89%	50%
$\gamma = 0.01$	76.43%	65.52%
$\gamma = 0.001$	96.49%	84.48%

sDTW Prediction Loss

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \frac{1}{m_i} \mathbf{dtw}_{\gamma}(f_{\boldsymbol{\theta}}(x_i), \boldsymbol{Y_i})$$



sDTW Prediction Loss

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \frac{1}{m_i} \mathbf{dtw}_{\gamma}(f_{\boldsymbol{\theta}}(x_i), \boldsymbol{Y_i})$$

Table 3. Averaged rank obtained by a multi-layer perceptron (MLP) under Euclidean and soft-DTW losses. Euclidean initialization means that we initialize the MLP trained with soft-DTW loss by the solution of the MLP trained with Euclidean loss.

Training loss	Random initialization	Euclidean initialization
When evaluating with DTW loss (dtw_0)		
Euclidean	3.46	4.21
soft-DTW ($\gamma = 1$)	3.55	3.96
soft-DTW ($\gamma = 0.1$)	3.33	3.42
soft-DTW ($\gamma = 0.01$)	2.79	2.12
soft-DTW ($\gamma = 0.001$)	**1.87**	**1.29**
When evaluating with Euclidean loss		
Euclidean	**1.05**	**1.70**
soft-DTW ($\gamma = 1$)	2.41	2.99
soft-DTW ($\gamma = 0.1$)	3.42	3.38
soft-DTW ($\gamma = 0.01$)	4.13	3.64
soft-DTW ($\gamma = 0.001$)	3.99	3.29

averaged rank

Summary

- **Dynamic Time Warping** is a natural and flexible discrepancy to compare time series, yet it is **non-differentiable**
- **Soft-DTW** is a differentiable approximation, with better convexity properties
- Using **soft-DTW** typically results in better minima, even when measured with the original DTW
- Python code available on

https://github.com/mblondel/soft-dtw