Learning with Regularized Distances: **Optimal Transport** and **Dynamic Time Warping**

Marco Cuturi

Joint work with many people, including:
G. Peyré, A. Genevay *(ENS)*, A. Doucet *(Oxford)*, J. Solomon *(MIT)*,
J.D. Benamou, N. Bonneel, F. Bach, L. Nenna *(INRIA)*,
G. Carlier (*Dauphine*), M. Blondel *(NTT)*.

What is Optimal Transport?

A geometric toolbox to
compare probability measures
supported on a metric space.


Monge


Kantorovich


Dantzig


Wasserstein


Brenier

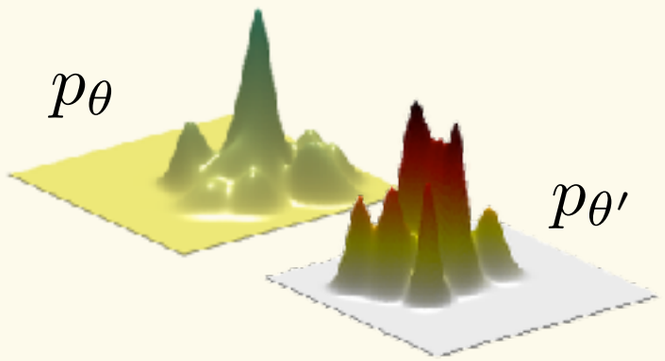

Otto


McCann


Villani

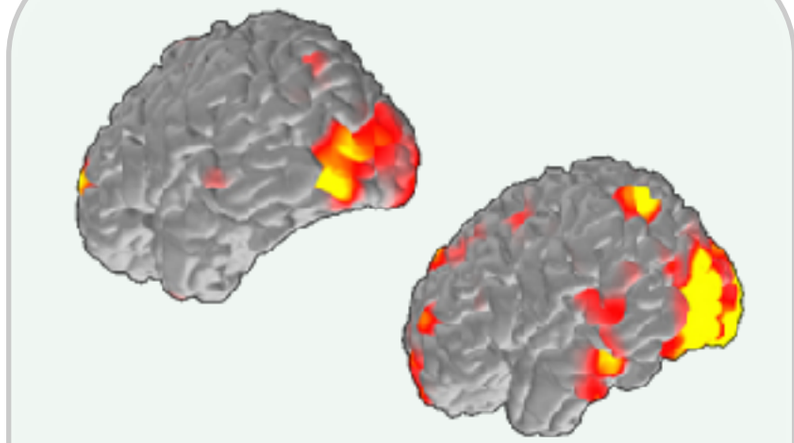
What is Optimal Transport?

A geometric toolbox to
compare **probability measures**
supported on a metric space.



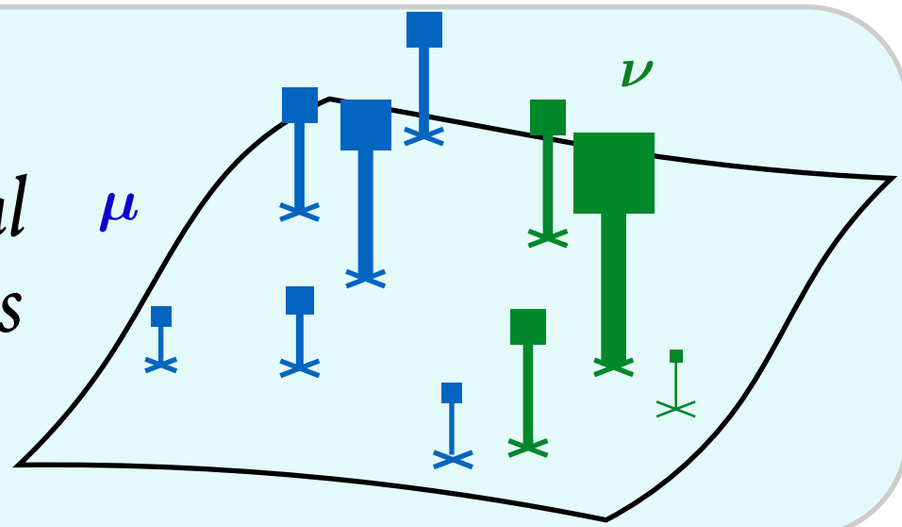
Statistical Models

Bags of features



Brain Activation Maps

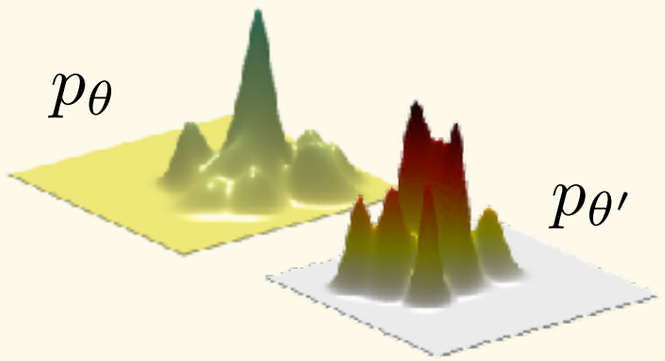
Empirical Measures



Color Histograms

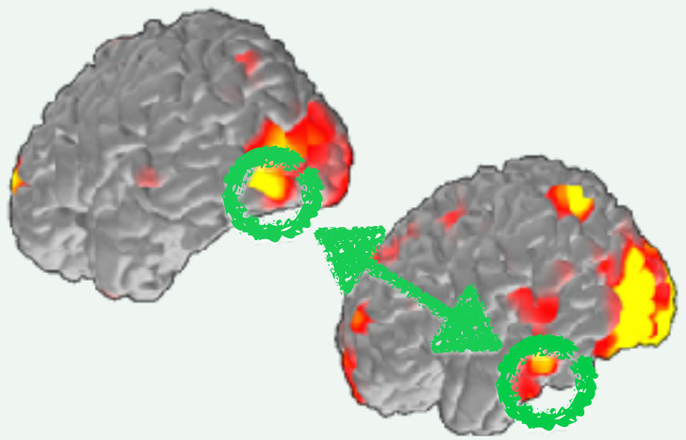
What is Optimal Transport?

A geometric toolbox to
compare probability measures
supported on a **metric space**.



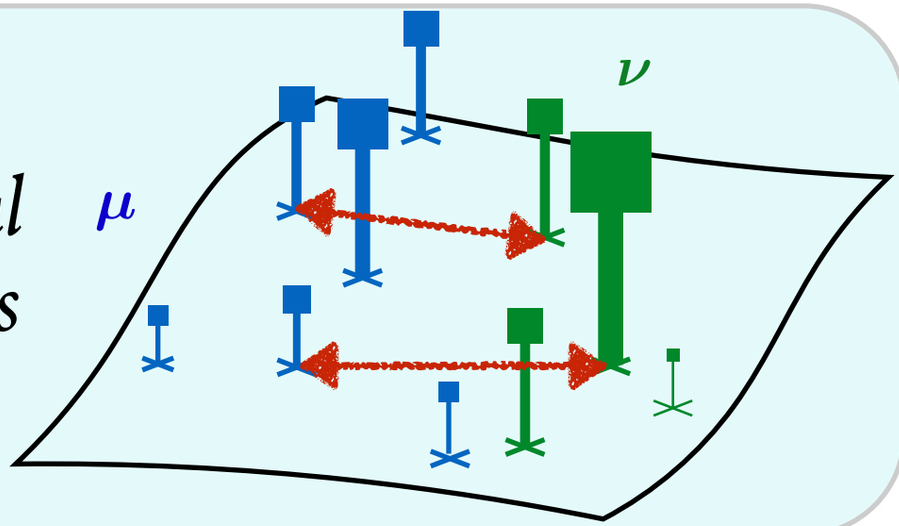
Statistical Models

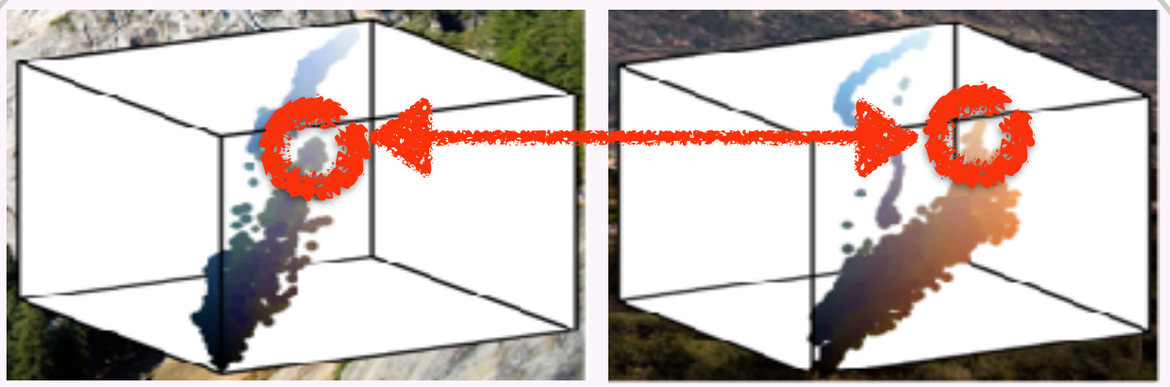
Bags of features



Brain Activation Maps

Empirical Measures

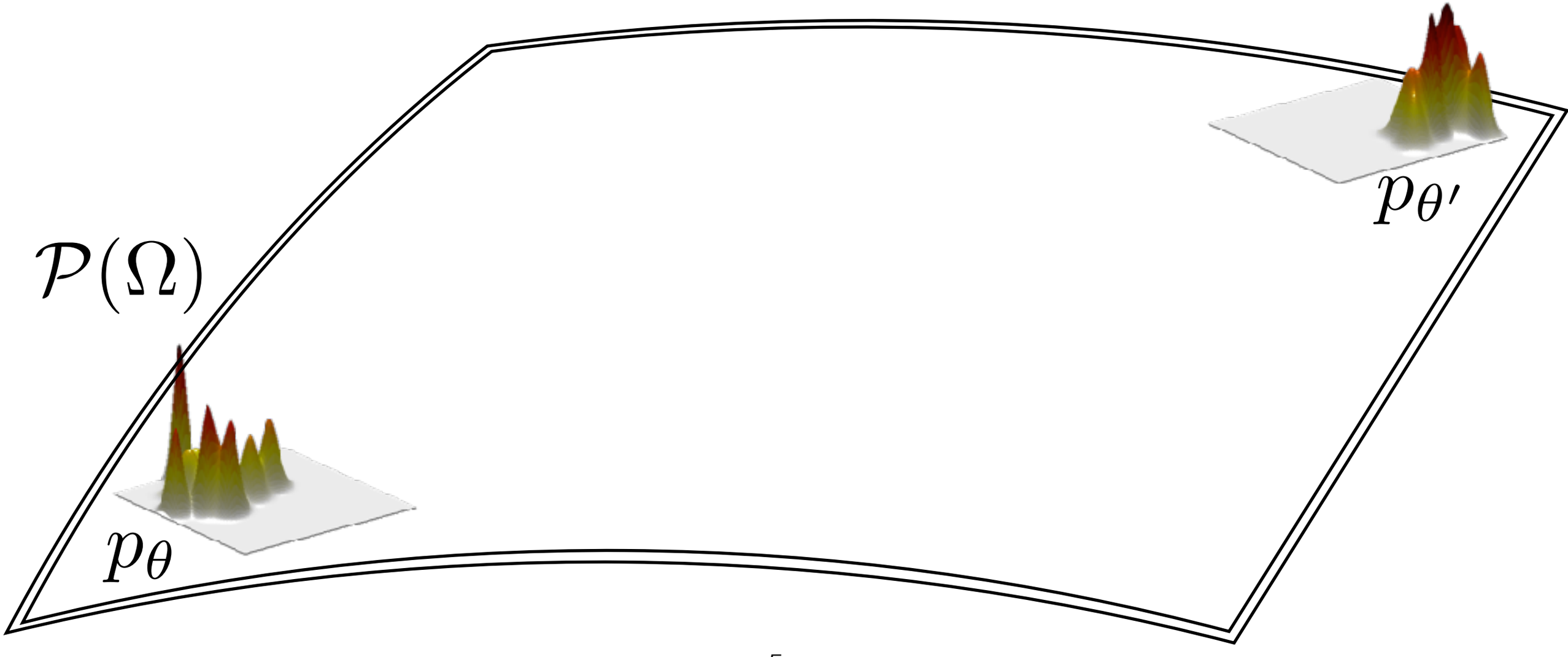




Color Histograms

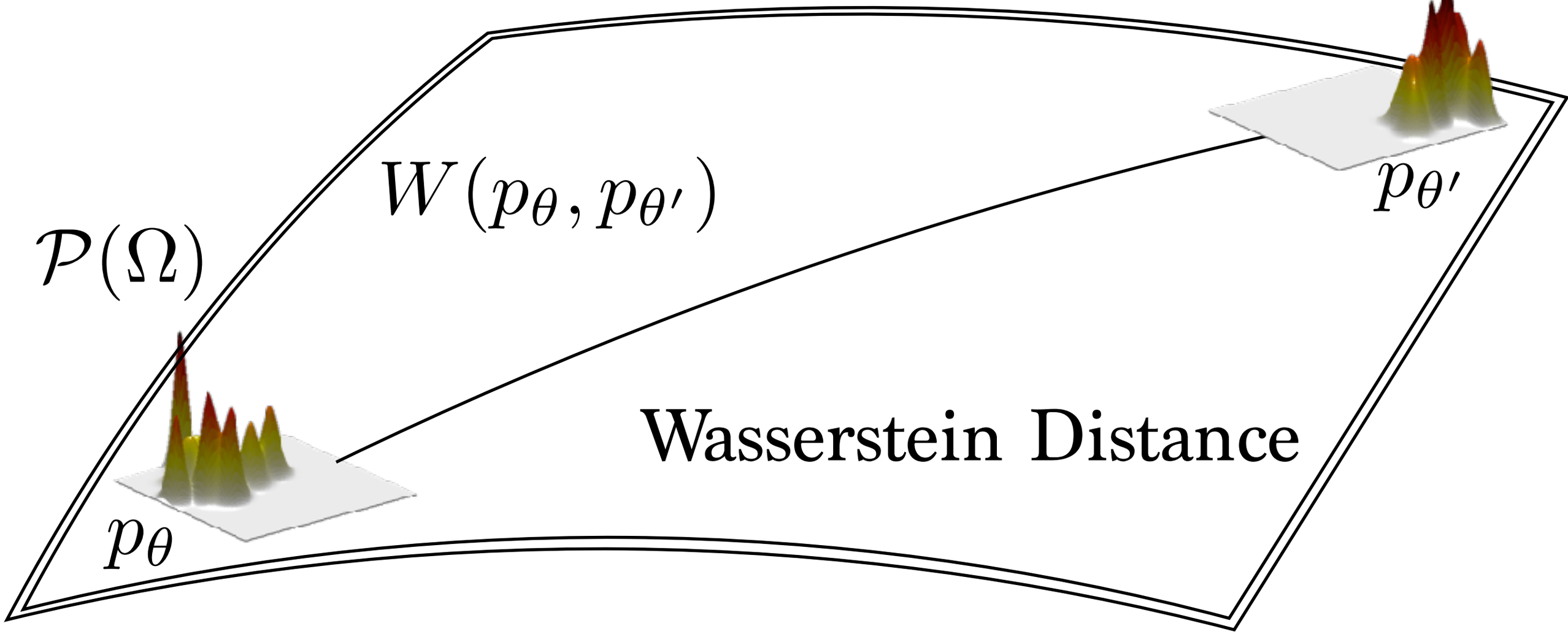
Optimal Transport Geometry

A **geometric toolbox** to
compare probability measures
supported on a metric space.



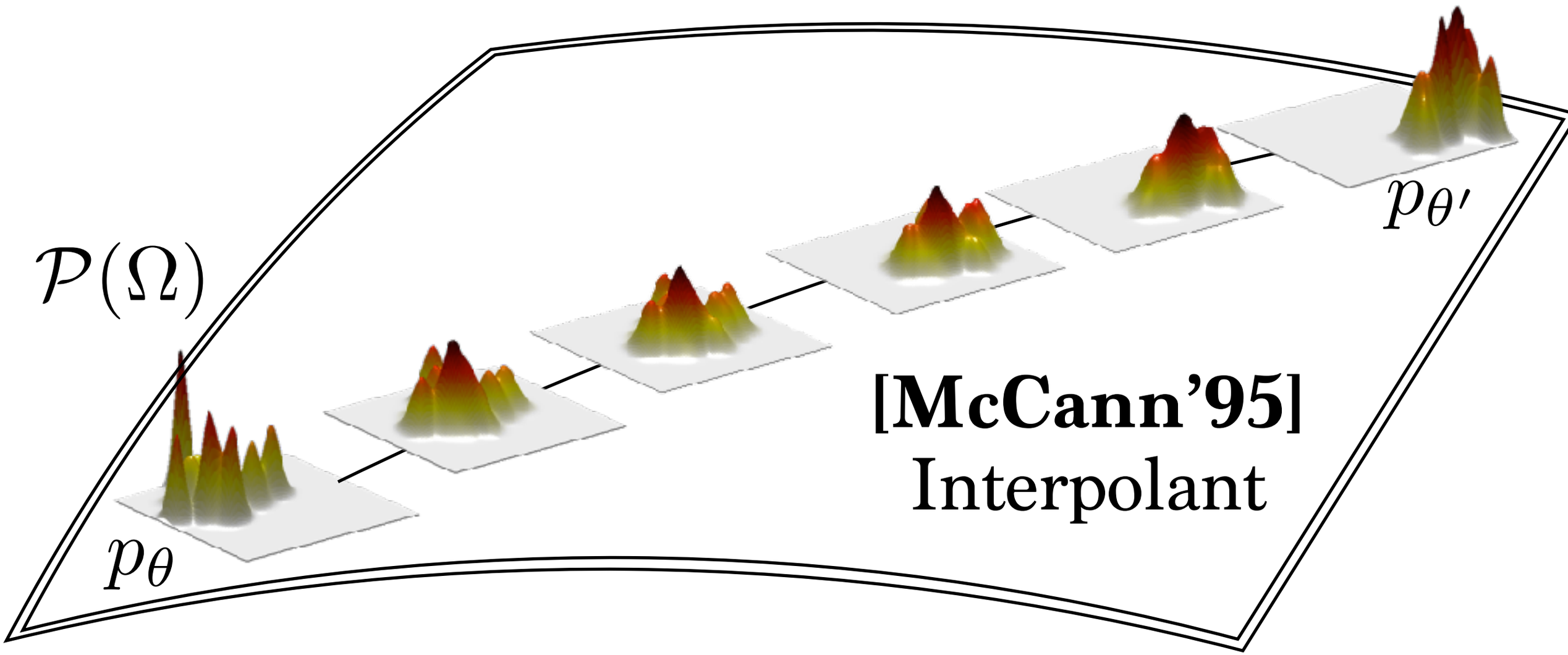
Optimal Transport Geometry

A **geometric toolbox** to
compare probability measures
supported on a metric space.



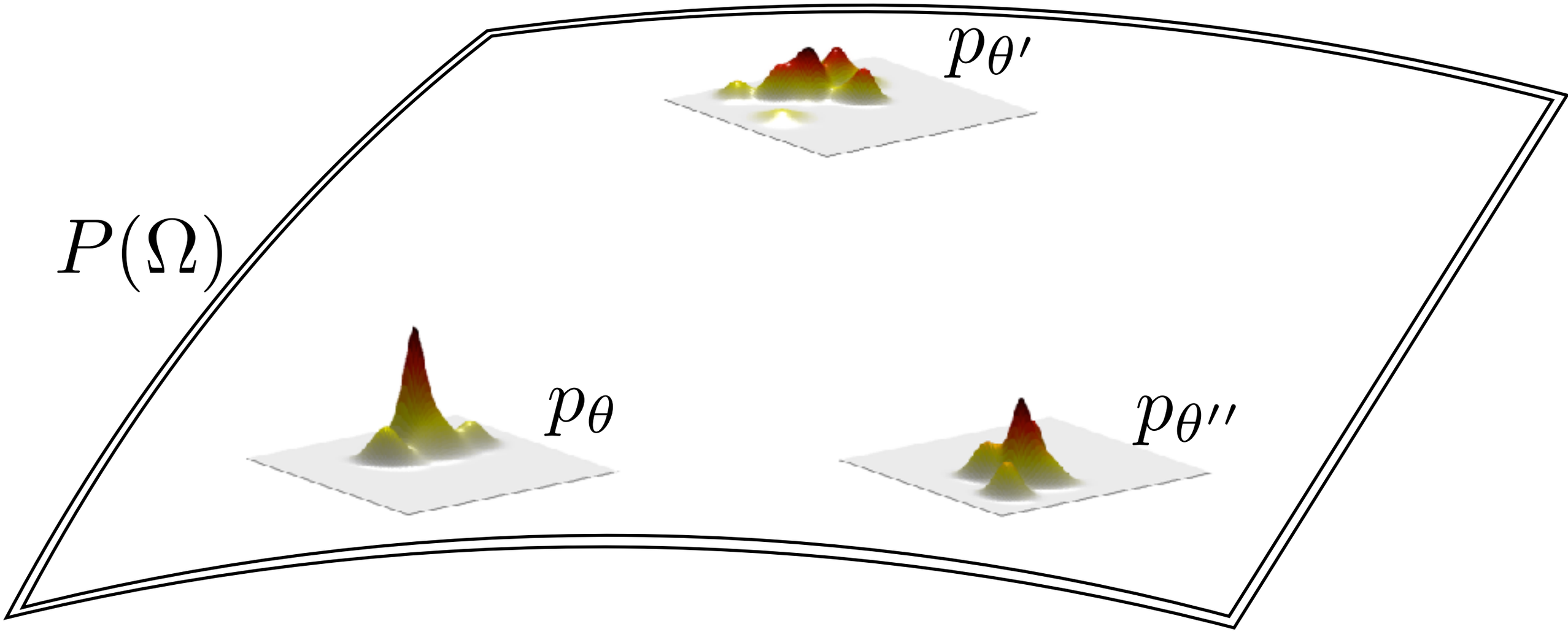
Optimal Transport Geometry

A **geometric toolbox** to
compare probability measures
supported on a metric space.



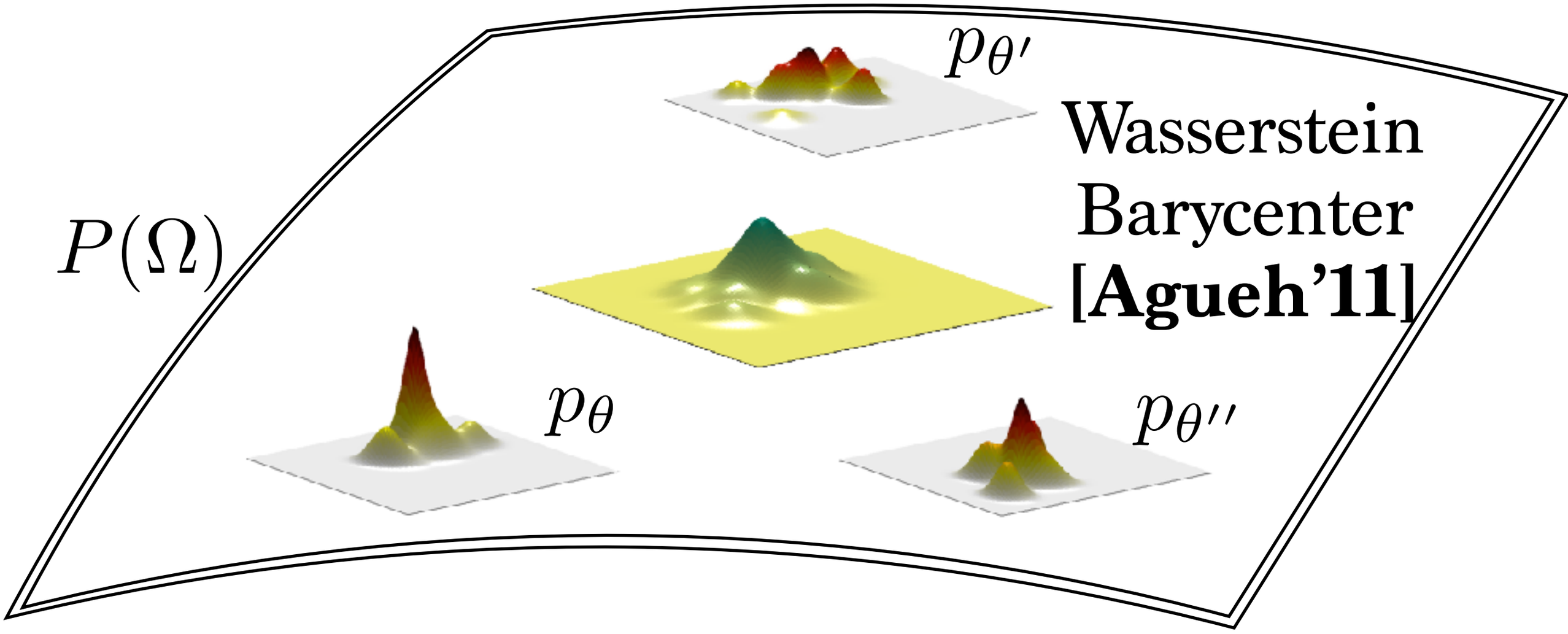
Optimal Transport Geometry

A **geometric toolbox** to
compare probability measures
supported on a metric space.



Optimal Transport Geometry

A **geometric toolbox** to
compare probability measures
supported on a metric space.



OT and data-analysis

- Key developments in (applied) maths ~'90s
  [McCann'95], [JKO'98], [Benamou'98], [Gangbo'98], [Ambrosio'06], [Villani'03/'09].

- Key developments in TCS / graphics since '00s
  [Rubner'98], [Indyk'03], [Naor'07], [Andoni'15].

◉ Small to *no-impact* in large-scale data analysis:

- ✦ **computationally heavy;**
- ✦ **Wasserstein distance is not differentiable**

OT and data-analysis

Today's talk: Entropy Regularized OT

- **Very fast** compared to usual approaches, GPGPU parallel.
- **Differentiable**, important if we want to use OT distances as **loss functions**.
- Can be **automatically differentiated**, simple iterative process, *DL*-toolboxes compatible.
- OT can become a building block in ML.

✦ Wasserstein distance is not differentiable

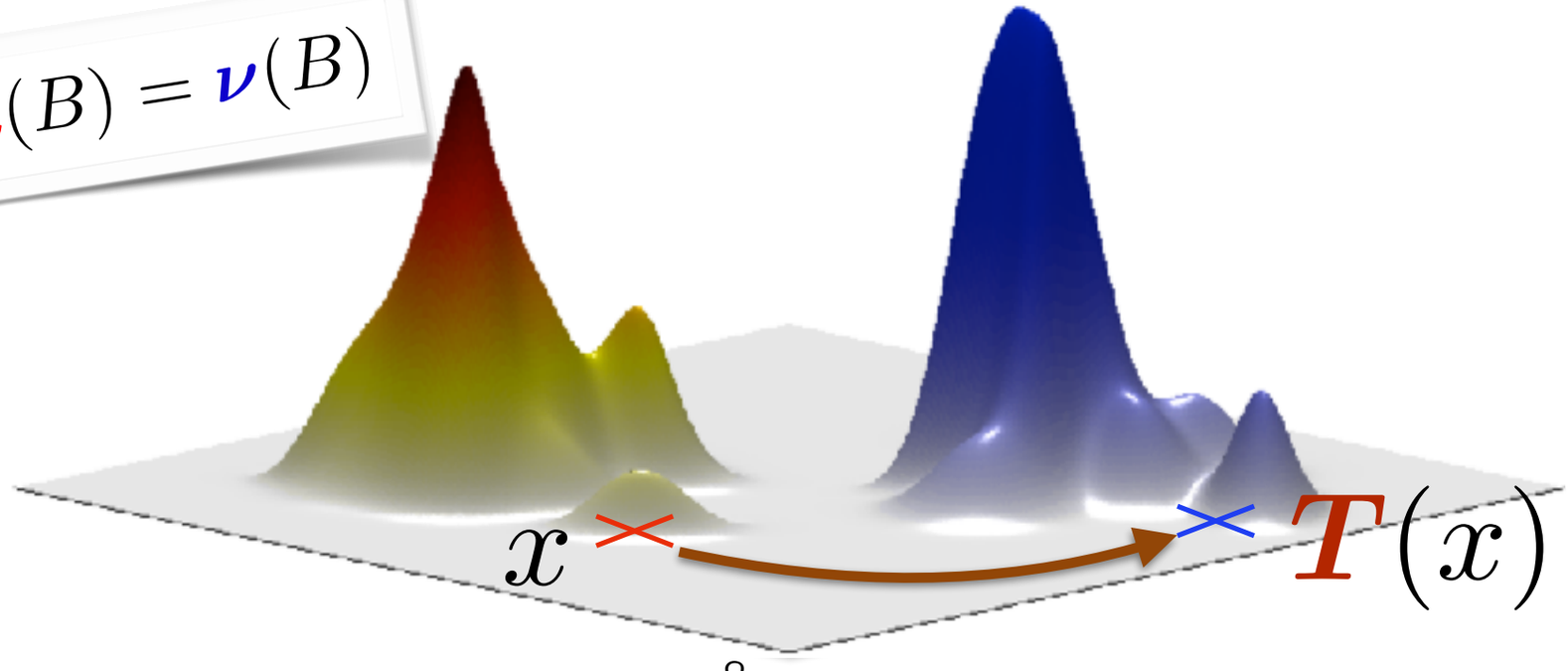
Background: OT Geometry

Ω a probability space, $\boldsymbol{c} : \Omega \times \Omega \to \mathbb{R}$.
$\boldsymbol{\mu}, \boldsymbol{\nu}$ two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $\boldsymbol{T} : \Omega \to \Omega$

$$\inf_{\boldsymbol{T}\#\boldsymbol{\mu}=\boldsymbol{\nu}} \int_\Omega \boldsymbol{c}(x, \boldsymbol{T}(x))\boldsymbol{\mu}(dx)$$

$\forall B \subset \Omega, \boldsymbol{T}\#\boldsymbol{\mu}(B) = \boldsymbol{\nu}(B)$

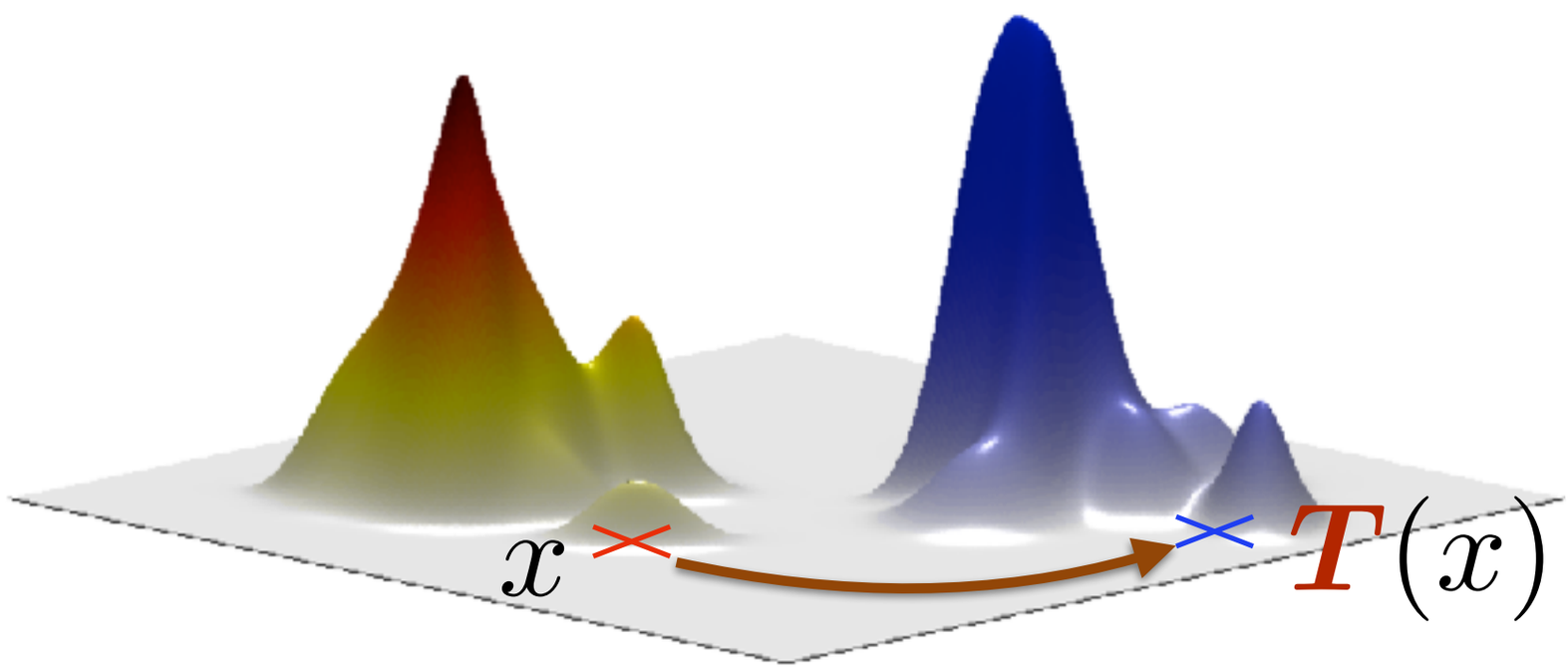


Background: OT Geometry

Ω a probability space, $\boldsymbol{c} : \Omega \times \Omega \to \mathbb{R}$.
$\boldsymbol{\mu}, \boldsymbol{\nu}$ two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $\boldsymbol{T} : \Omega \to \Omega$

[Brenier'87] If $\Omega = \mathbb{R}^d, \boldsymbol{c} = \| \cdot - \cdot \|^2$,
$\boldsymbol{\mu}, \boldsymbol{\nu}$ a.c., then $\boldsymbol{T} = \nabla \boldsymbol{u}$, $\boldsymbol{u}$ **convex.**

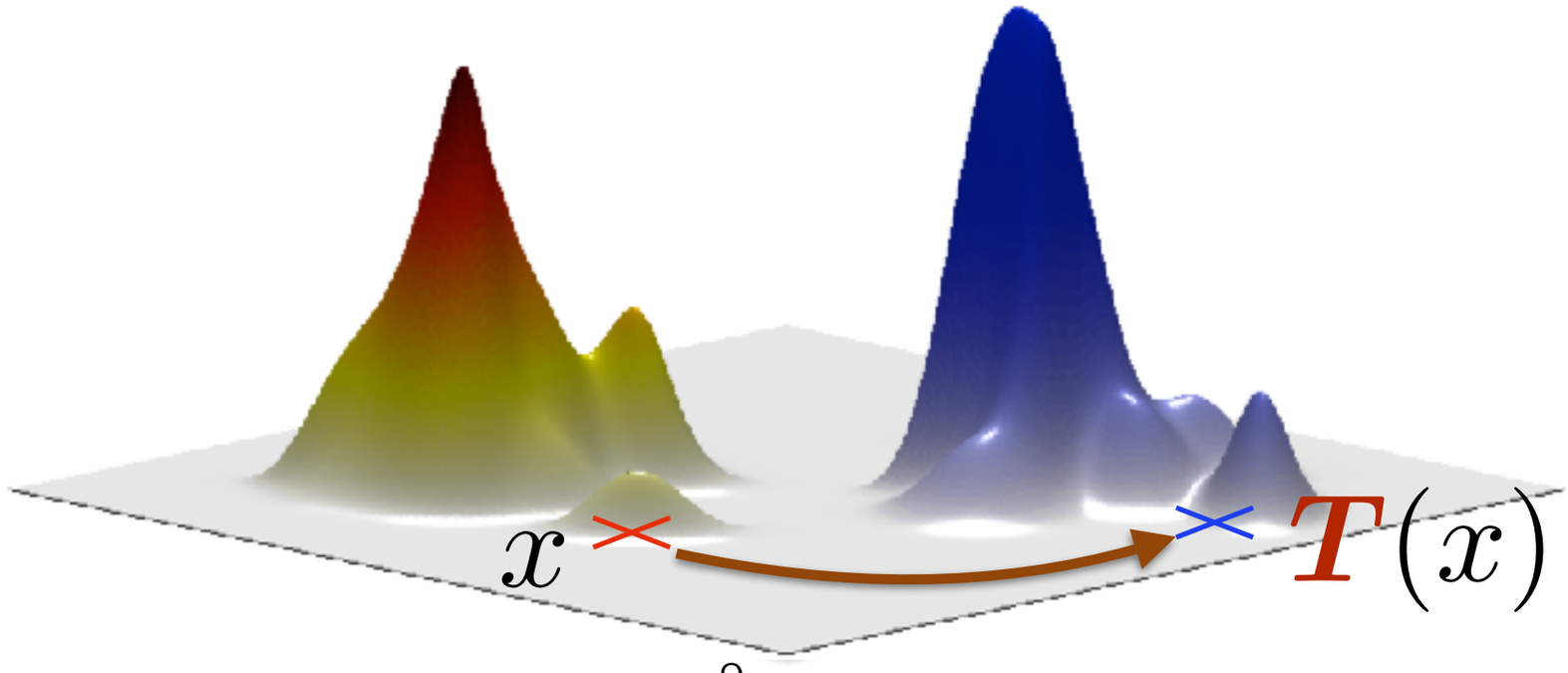


Background: OT Geometry

Ω a probability space, $\boldsymbol{c} : \Omega \times \Omega \to \mathbb{R}$.
$\boldsymbol{\mu}, \boldsymbol{\nu}$ two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $\boldsymbol{T} : \Omega \to \Omega$

$$\inf_{\boldsymbol{T}\#\boldsymbol{\mu}=\boldsymbol{\nu}} \int_\Omega \boldsymbol{c}(x, \boldsymbol{T}(x))\boldsymbol{\mu}(dx)$$

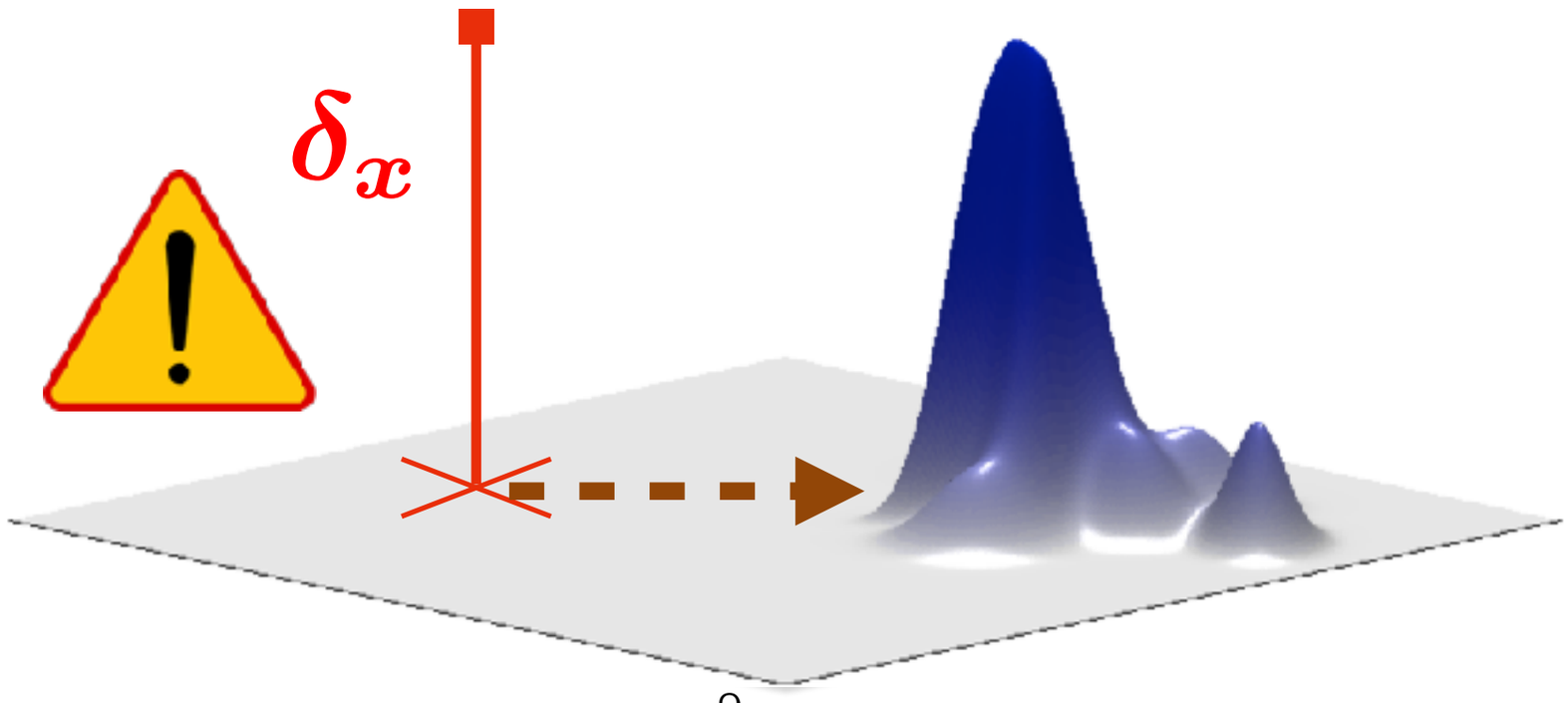


Background: OT Geometry

Ω a probability space, $\boldsymbol{c} : \Omega \times \Omega \to \mathbb{R}$.
$\boldsymbol{\mu}, \boldsymbol{\nu}$ two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $\boldsymbol{T} : \Omega \to \Omega$

$$\inf_{\boldsymbol{T}\#\boldsymbol{\mu}=\boldsymbol{\nu}} \int_{\Omega} \boldsymbol{c}(x, \boldsymbol{T}(x))\boldsymbol{\mu}(dx)$$



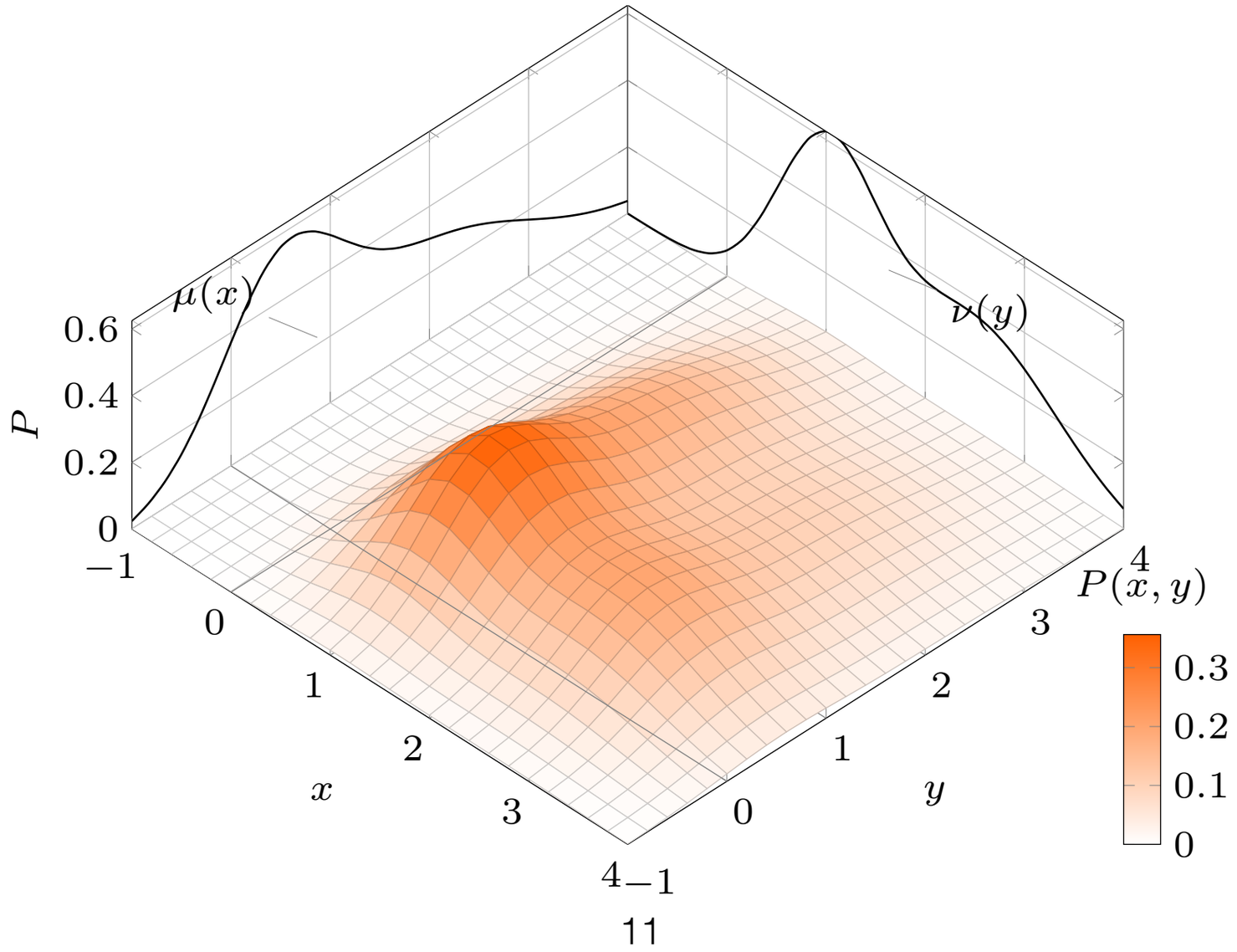
[Kantorovich'42] Relaxation

- Instead of maps $\boldsymbol{T} : \Omega \to \Omega$, consider probabilistic maps, i.e. **couplings** $\boldsymbol{P} \in \mathcal{P}(\Omega \times \Omega)$**:**

$$\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathcal{P}(\Omega \times \Omega) | \forall \boldsymbol{A}, \boldsymbol{B} \subset \Omega,$$
$$\boldsymbol{P}(\boldsymbol{A} \times \Omega) = \boldsymbol{\mu}(\boldsymbol{A}),$$
$$\boldsymbol{P}(\Omega \times \boldsymbol{B}) = \boldsymbol{\nu}(\boldsymbol{B})\}$$

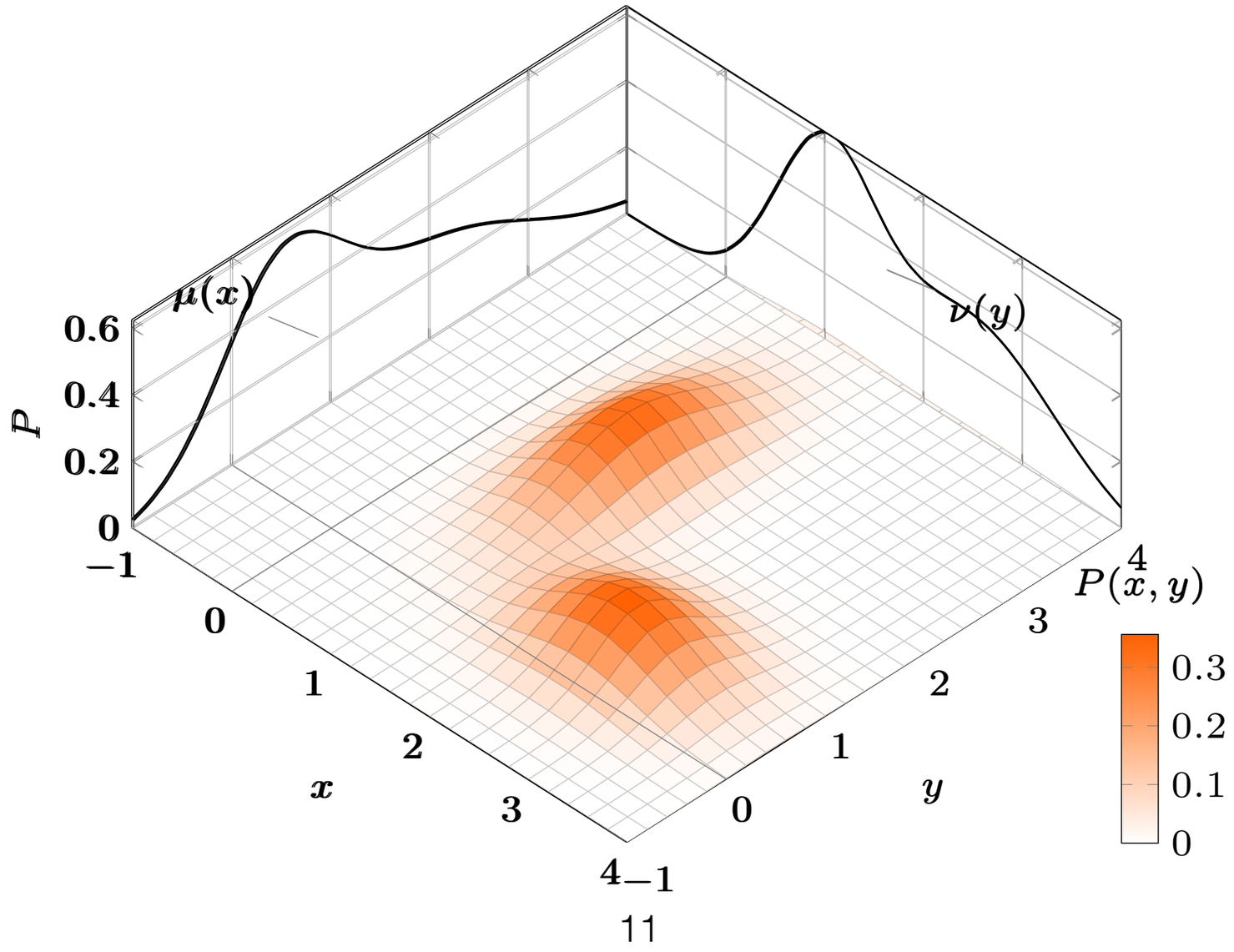
[Kantorovich'42] Relaxation

$$\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathcal{P}(\Omega \times \Omega) | \forall \boldsymbol{A}, \boldsymbol{B} \subset \Omega,$$
$$\boldsymbol{P}(\boldsymbol{A} \times \Omega) = \boldsymbol{\mu}(\boldsymbol{A}), \boldsymbol{P}(\Omega \times \boldsymbol{B}) = \boldsymbol{\nu}(\boldsymbol{B})\}$$



[Kantorovich'42] Relaxation

$$\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathcal{P}(\Omega \times \Omega) | \forall \boldsymbol{A}, \boldsymbol{B} \subset \Omega,$$
$$\boldsymbol{P}(\boldsymbol{A} \times \Omega) = \boldsymbol{\mu}(\boldsymbol{A}), \boldsymbol{P}(\Omega \times \boldsymbol{B}) = \boldsymbol{\nu}(\boldsymbol{B})\}$$

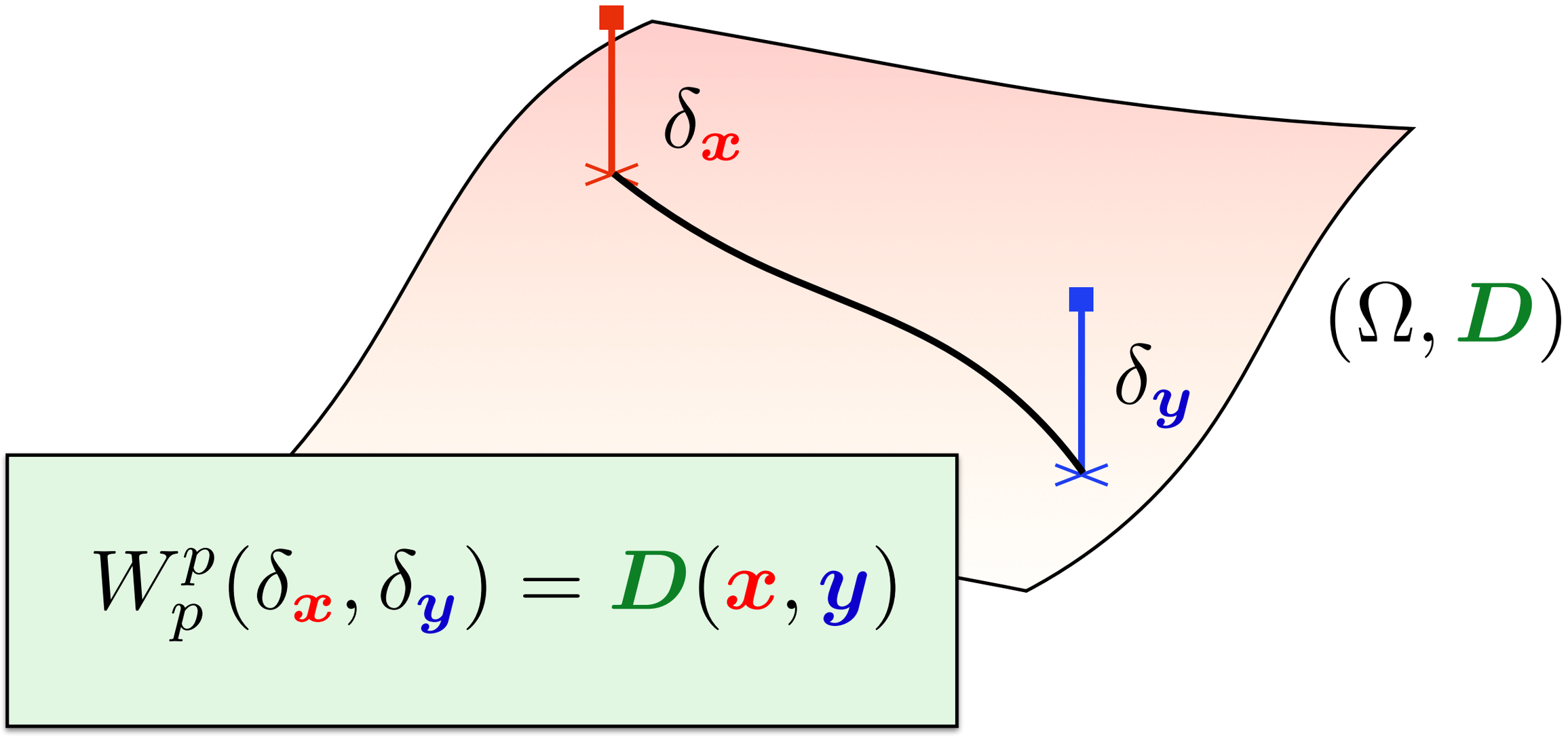


Wasserstein Distance

Def. For $p \geq 1$, the p-Wasserstein distance between $\boldsymbol{\mu}, \boldsymbol{\nu}$ in $\mathcal{P}(\Omega)$ is

$$W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \left(\inf_{\boldsymbol{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathbb{E}_{\boldsymbol{P}}[D(X, Y)^p] \right)^{1/p}.$$

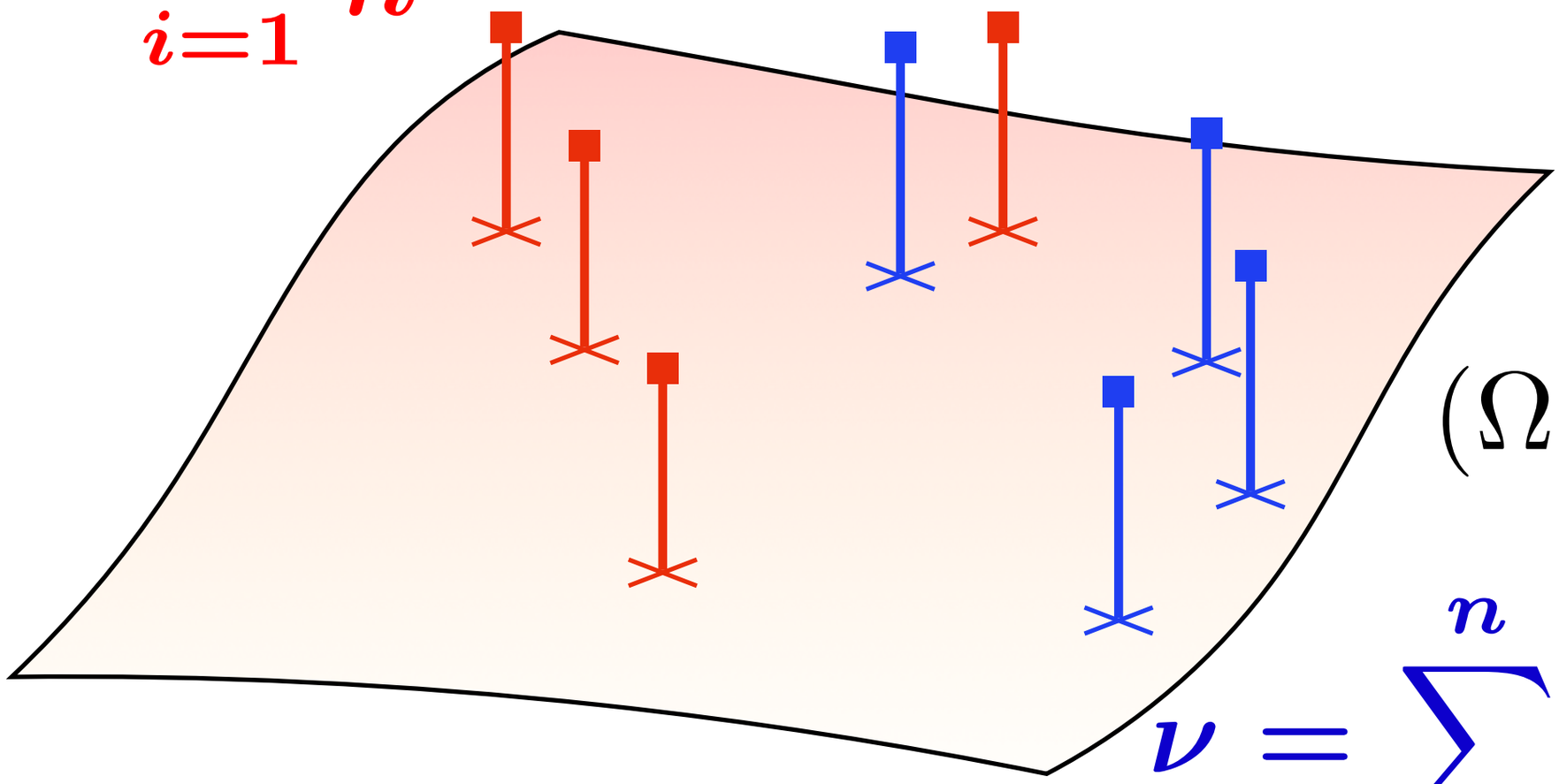
Wasserstein between 2 Diracs



$$W_p^p(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}}) = \boldsymbol{D}(\boldsymbol{x}, \boldsymbol{y})$$

Wasserstein on Uniform Measures

$$\mu = \sum_{i=1}^{n} \frac{1}{n} \delta_{x_i}$$

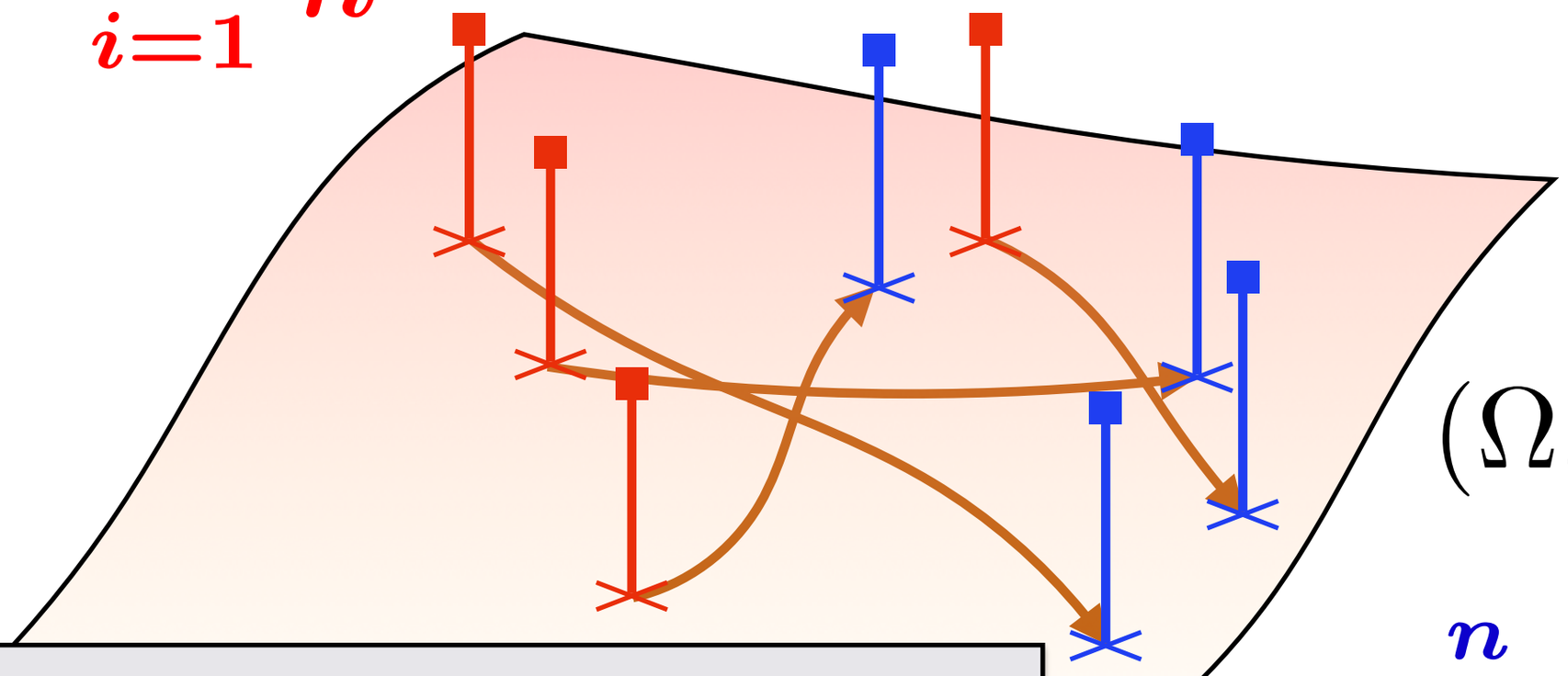


(Ω, D)

$$\nu = \sum_{j=1}^{n} \frac{1}{n} \delta_{y_j}$$

Wasserstein on Uniform Measures

$$\mu = \sum_{i=1}^{n} \frac{1}{n} \delta_{x_i}$$



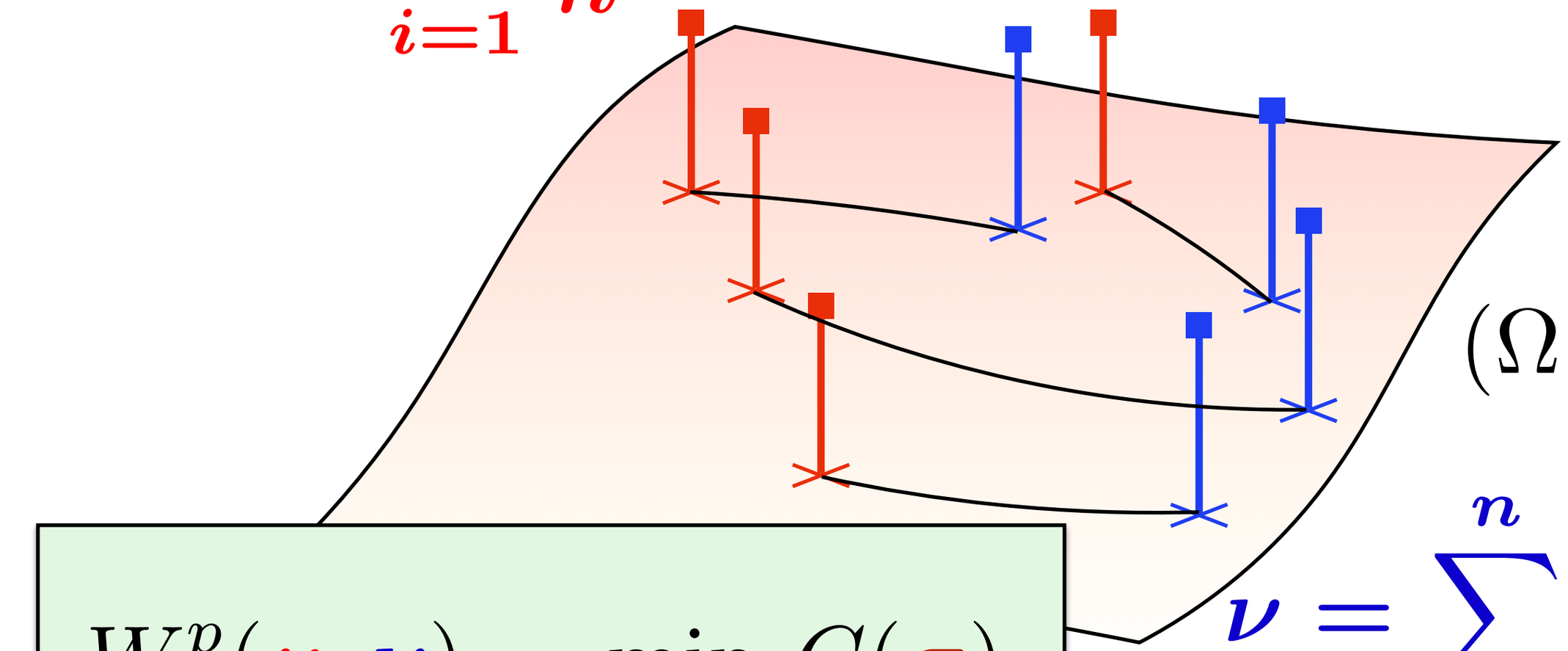
(Ω, D)

$$\nu = \sum_{j=1}^{n} \frac{1}{n} \delta_{y_j}$$

$$C(\sigma) = \frac{1}{n} \sum_{i=1}^{n} D(x_i, y_{\sigma_i})^p$$

Optimal Assignment $\subset$ Wasserstein

$$\mu = \sum_{i=1}^{n} \frac{1}{n} \delta_{x_i}$$



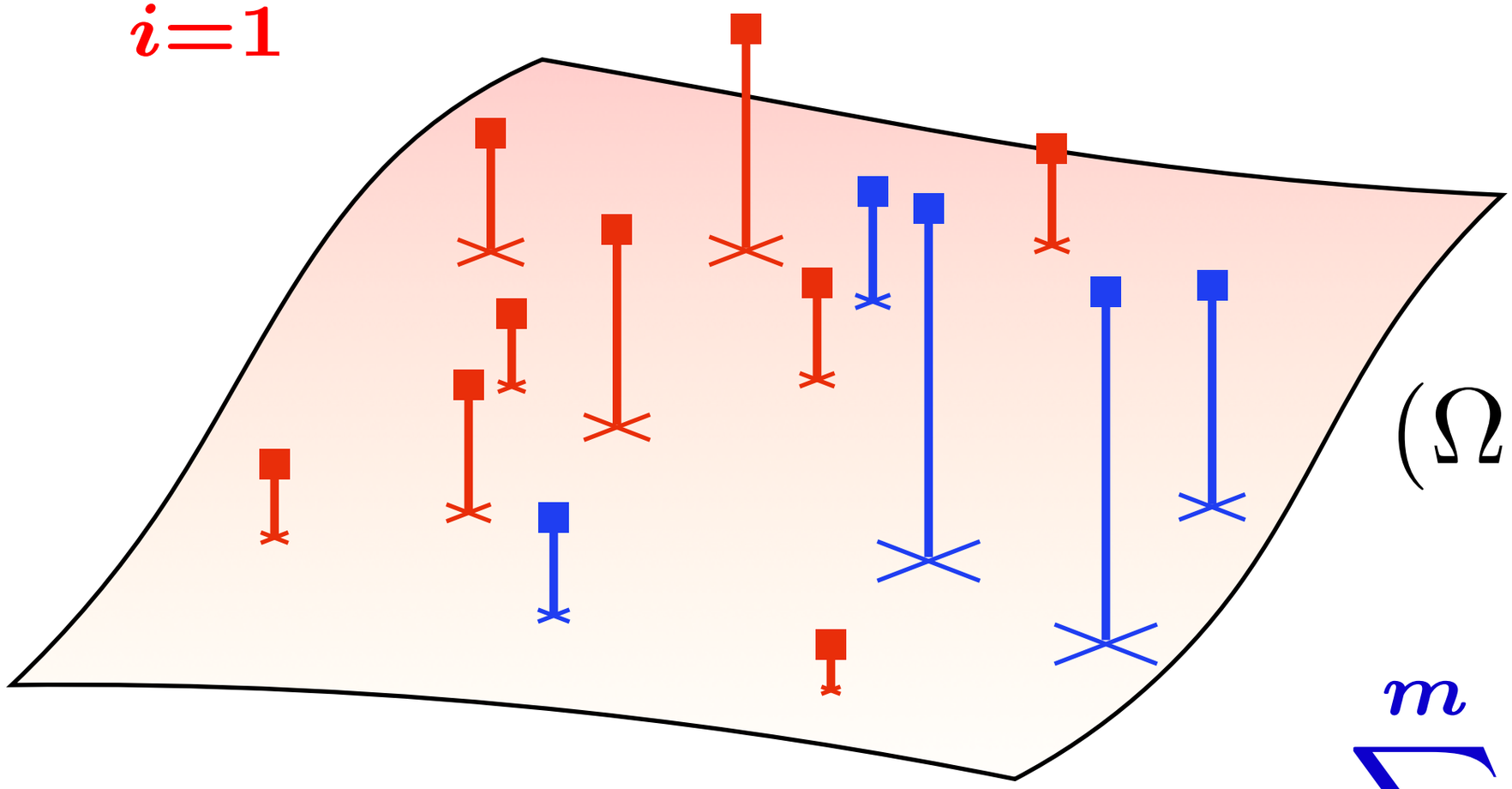
$$(\Omega, D)$$

$$\nu = \sum_{j=1}^{n} \frac{1}{n} \delta_{y_j}$$

$$W_p^p(\mu, \nu) = \min_{\sigma \in S_n} C(\sigma)$$

Wasserstein on Empirical Measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$$



(Ω, D)

$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

Wasserstein on Empirical Measures

Consider $\boldsymbol{\mu = \sum_{i=1}^{n} a_i \delta_{x_i}}$ and $\boldsymbol{\nu = \sum_{j=1}^{m} b_j \delta_{y_j}}$.

$$M_{\boldsymbol{XY}} \stackrel{\text{def}}{=} [D(\boldsymbol{x}_i, \boldsymbol{y}_j)^p]_{ij}$$

$$U(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} | \boldsymbol{P}\mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T\mathbf{1}_n = \boldsymbol{b}\}$$

$$\begin{array}{c} \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_m \end{array} \\ \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & D(\boldsymbol{x}_i, \boldsymbol{y}_j)^p & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_n \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_m \end{array} \\ \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \boldsymbol{P}\mathbf{1}_m = \boldsymbol{a} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

Wasserstein on Empirical Measures

Consider $\boldsymbol{\mu} = \sum_{i=1}^{n} \boldsymbol{a_i \delta_{x_i}}$ and $\boldsymbol{\nu} = \sum_{j=1}^{m} \boldsymbol{b_j \delta_{y_j}}$.

$$M_{\boldsymbol{XY}} \stackrel{\text{def}}{=} [D(\boldsymbol{x}_i, \boldsymbol{y}_j)^p]_{ij}$$

$$U(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} | \boldsymbol{P}\mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T\mathbf{1}_n = \boldsymbol{b}\}$$

	$\boldsymbol{y_1}$	$\cdots$	$\boldsymbol{y_m}$
$\boldsymbol{x_1}$			
$\vdots$		$D(\boldsymbol{x_i}, \boldsymbol{y_j})^p$	
$\boldsymbol{x_n}$			

	$\boldsymbol{b_1}$	$\cdots$	$\boldsymbol{b_m}$
$\boldsymbol{a_1}$			
$\vdots$		$\boldsymbol{P}^T\mathbf{1}_n = \boldsymbol{b}$	
$\boldsymbol{a_n}$			

Wasserstein on Empirical Measures

Consider $\boldsymbol{\mu} = \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{\delta}_{\boldsymbol{x}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^{m} \boldsymbol{b}_j \boldsymbol{\delta}_{\boldsymbol{y}_j}$.

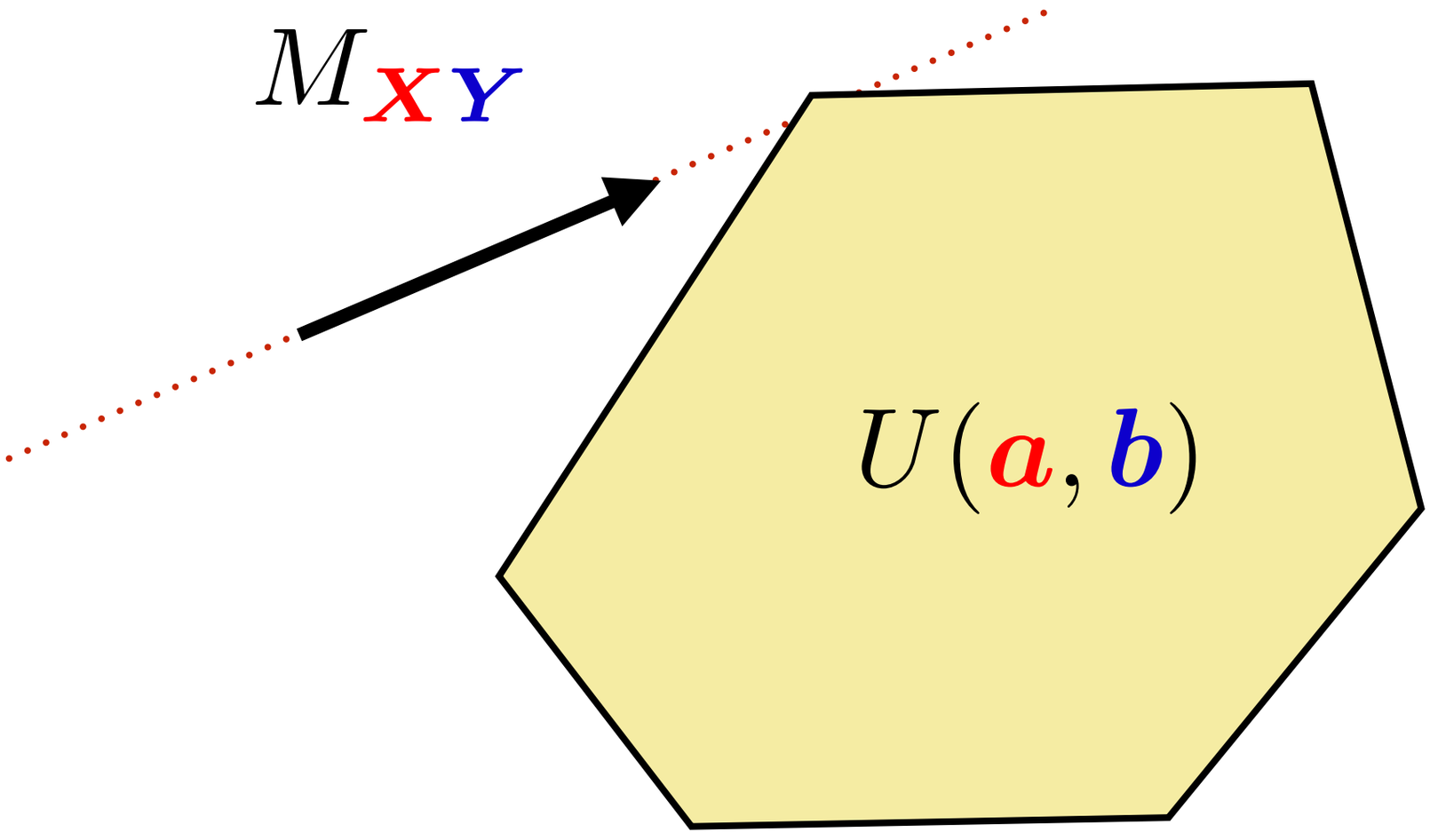
$$M_{\boldsymbol{X}\boldsymbol{Y}} \stackrel{\text{def}}{=} [D(\boldsymbol{x}_i, \boldsymbol{y}_j)^p]_{ij}$$

$$U(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} | \boldsymbol{P}\mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T\mathbf{1}_n = \boldsymbol{b}\}$$

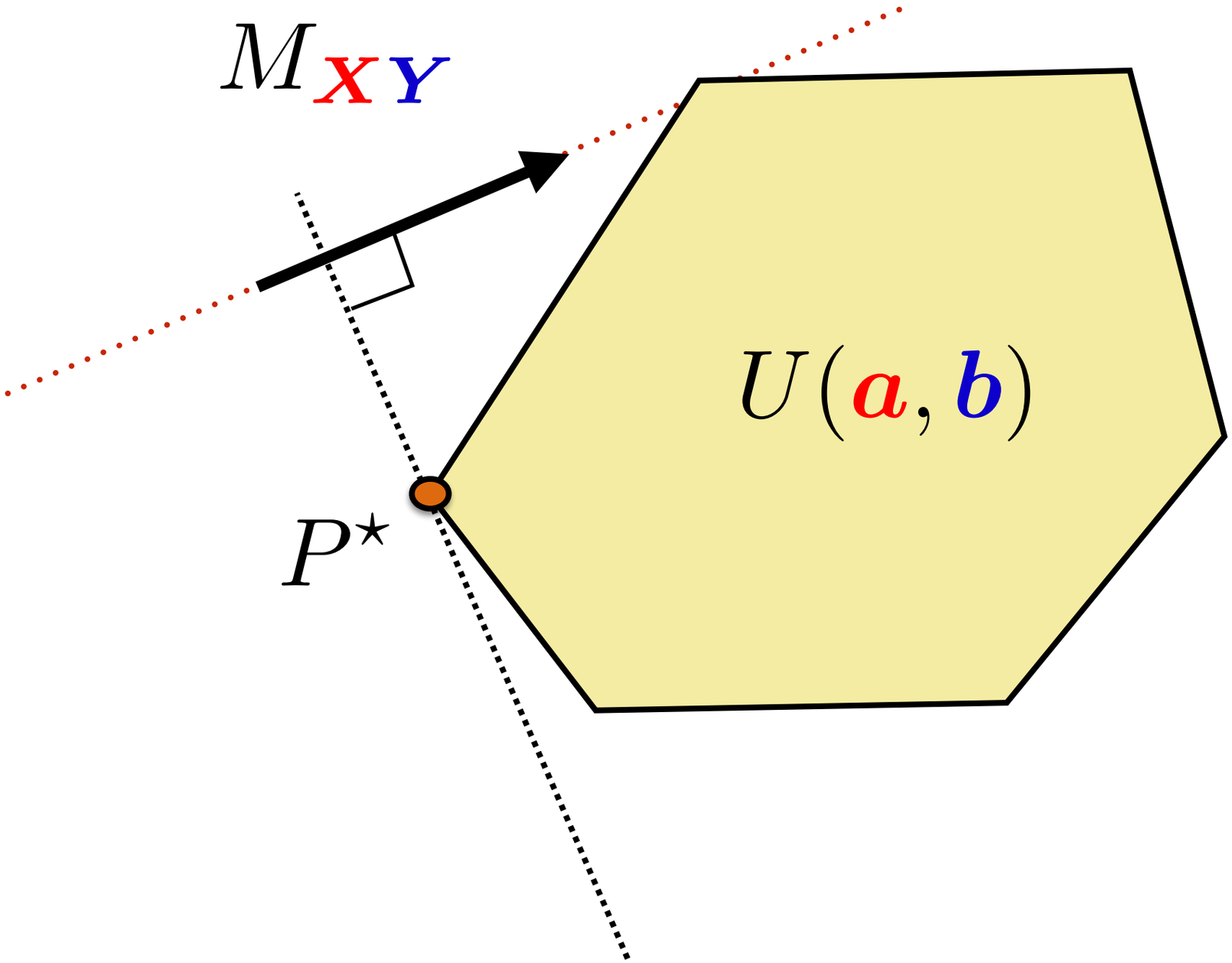
Def. Optimal Transport Problem

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

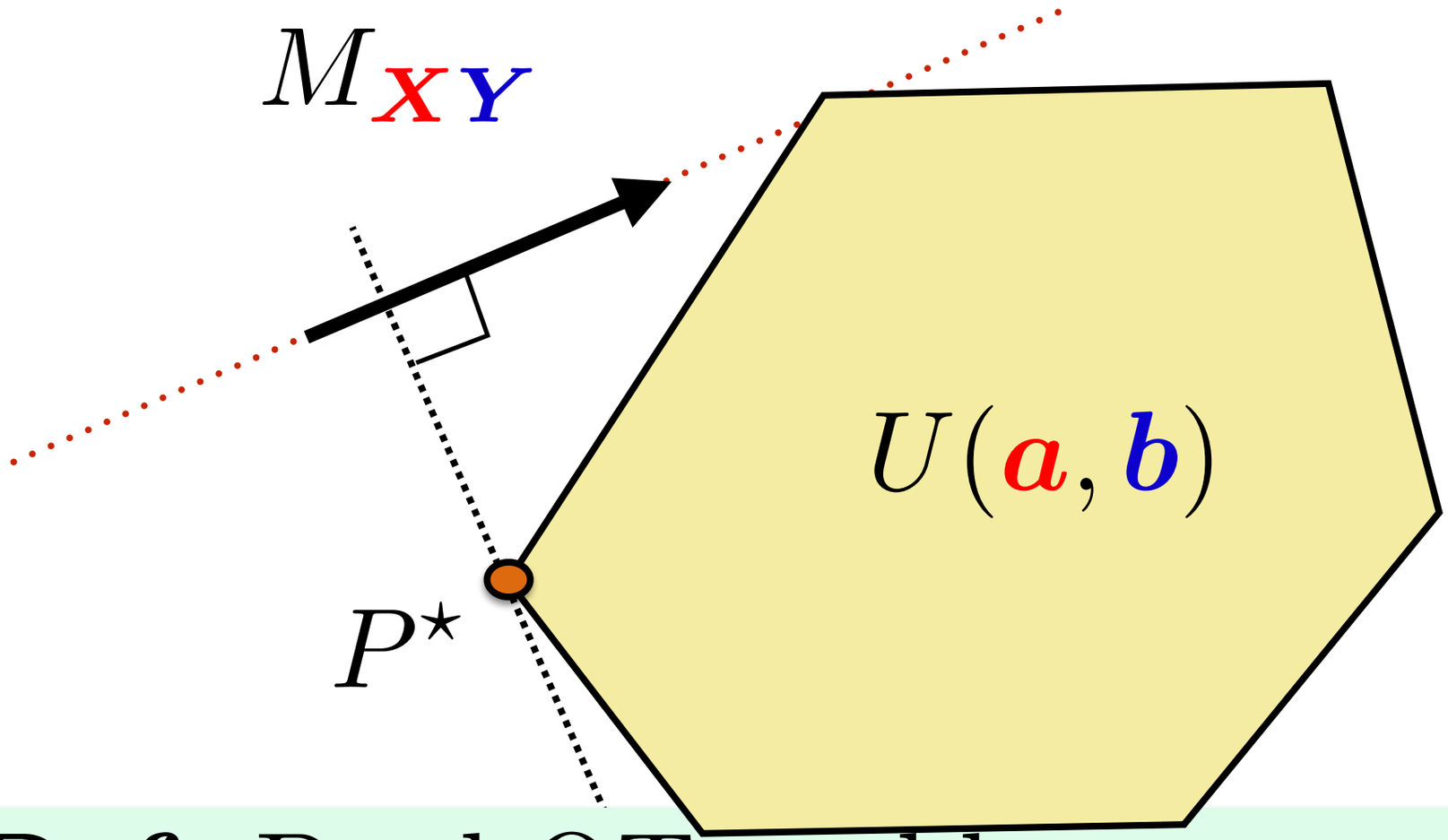
Discrete OT Problem



Discrete OT Problem



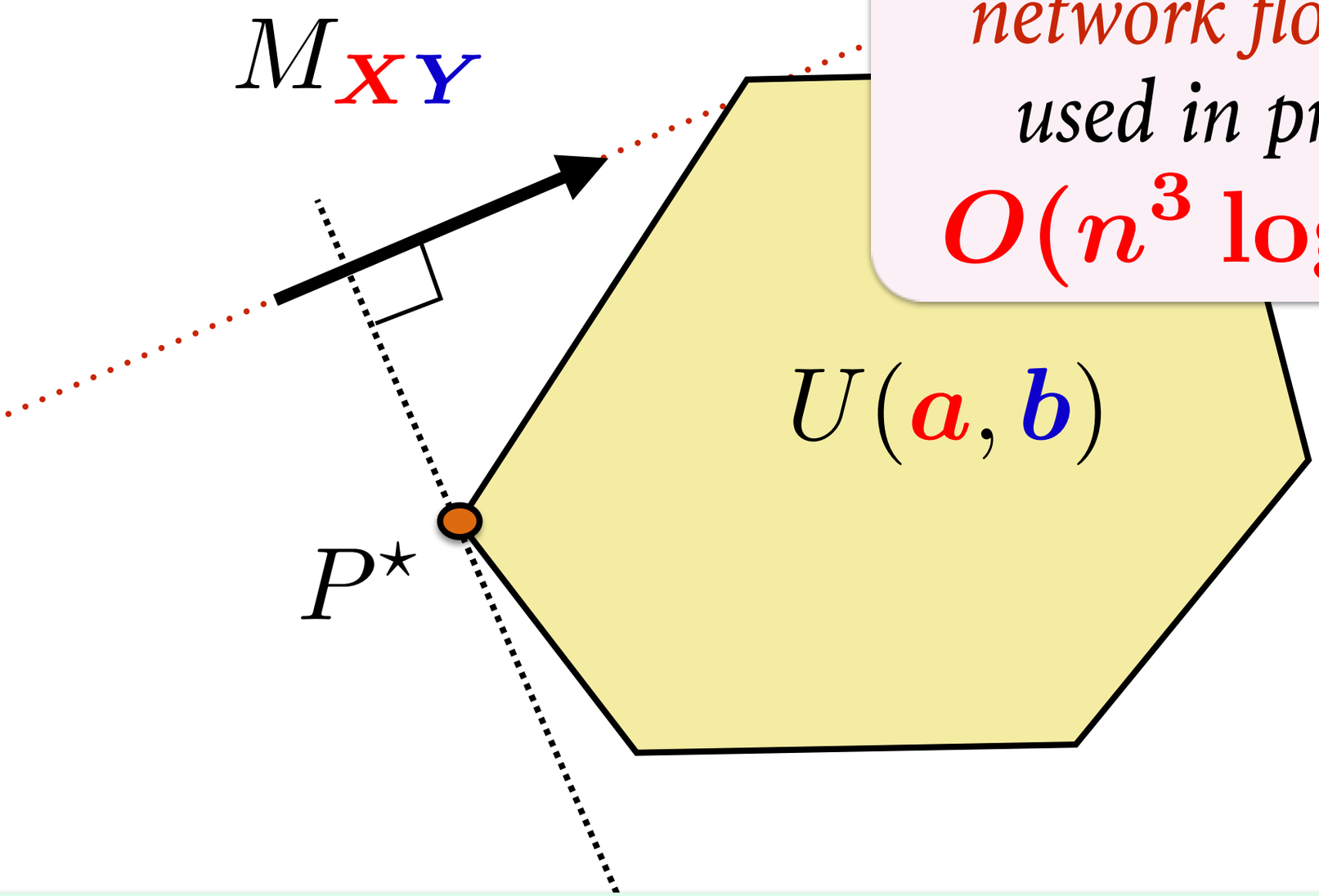
Discrete OT Problem



Def. Dual OT problem

$$W_p^p(\mu, \nu) = \max_{\substack{\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(x_i, y_j)^p}} \alpha^T a + \beta^T b$$

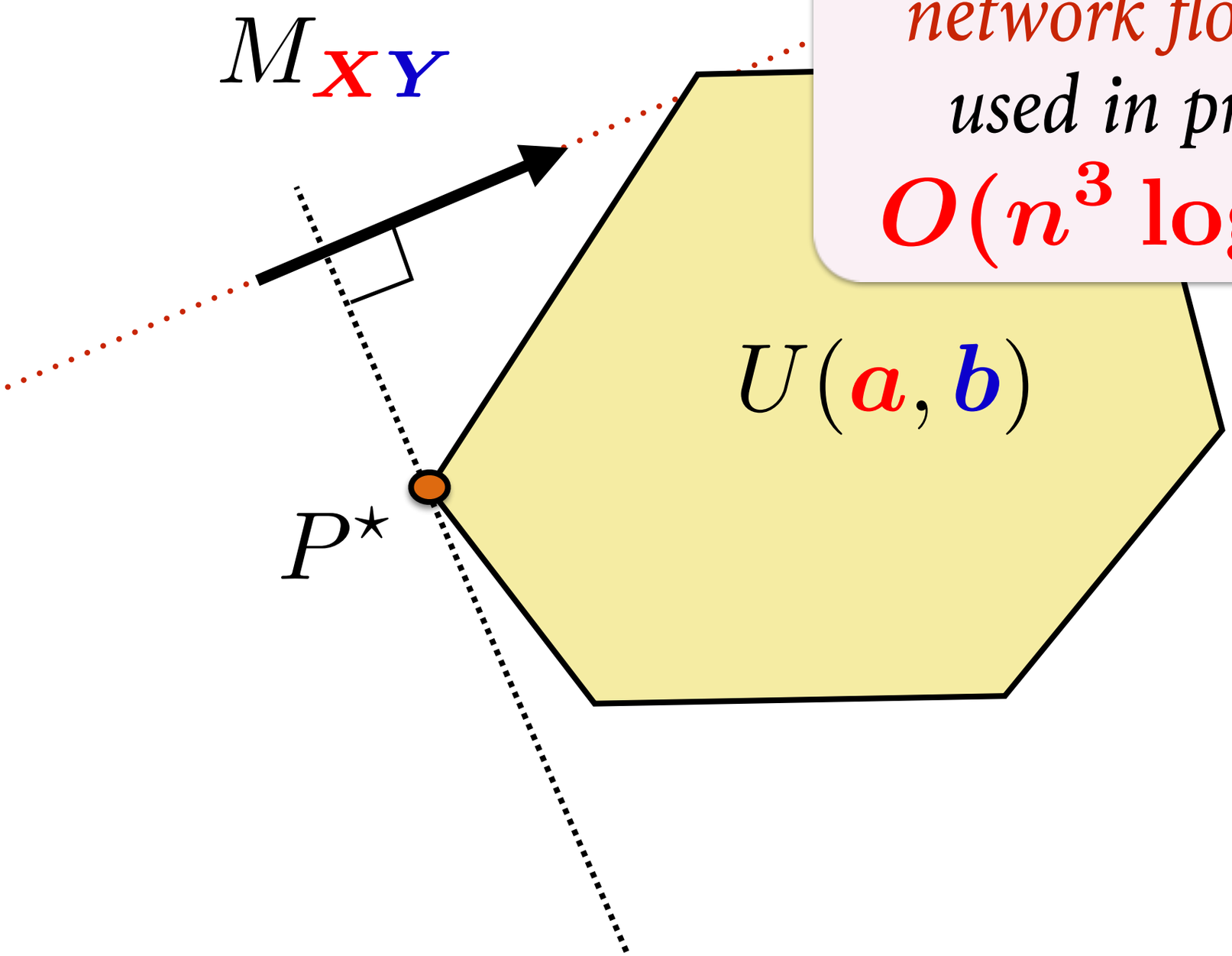
Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

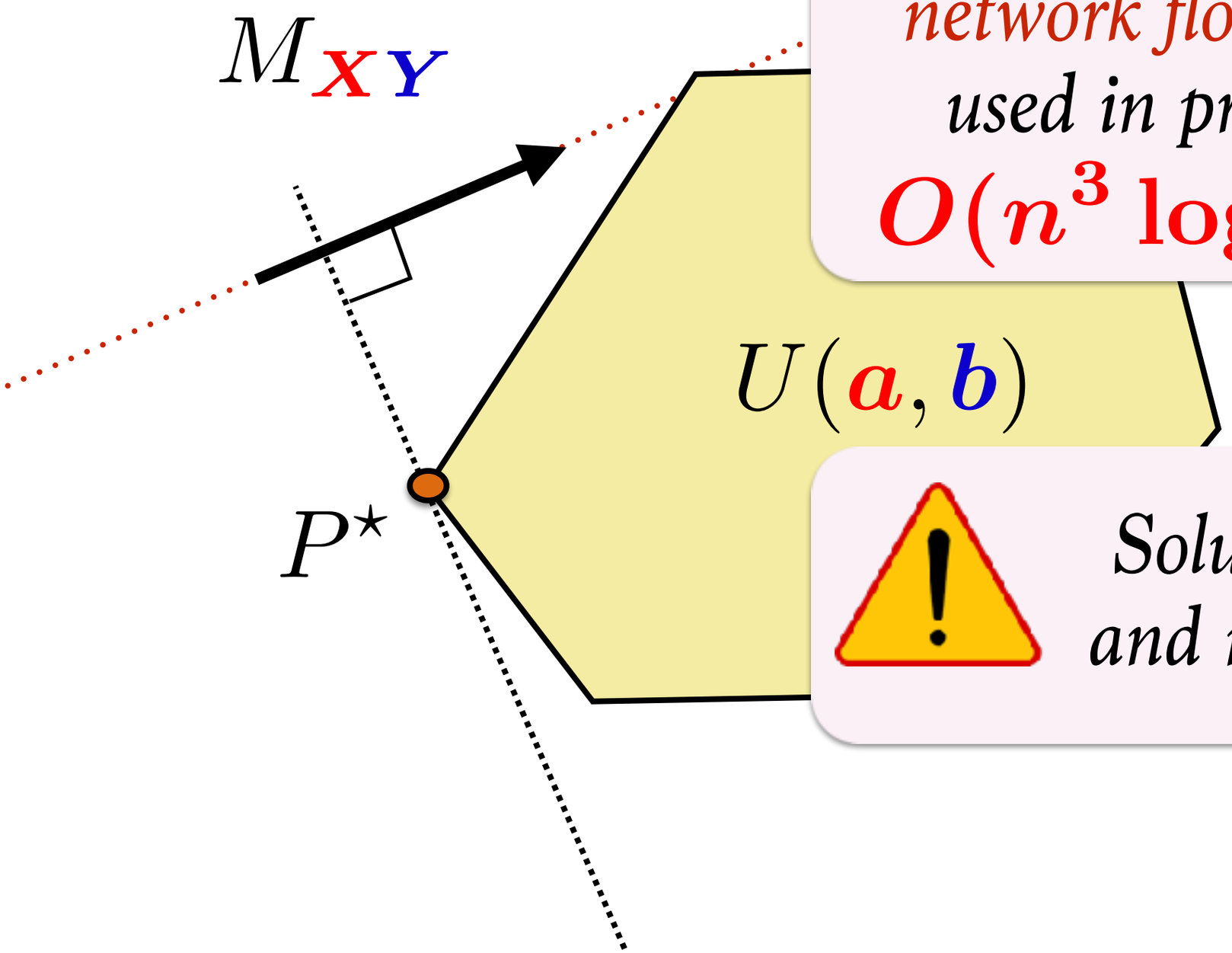
Note: flow/PDE formulations **[Beckman'61]/[Benamou'98]** can be used for *p=1/p=2* for a sparse-graph metric/Euclidean metric.

Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

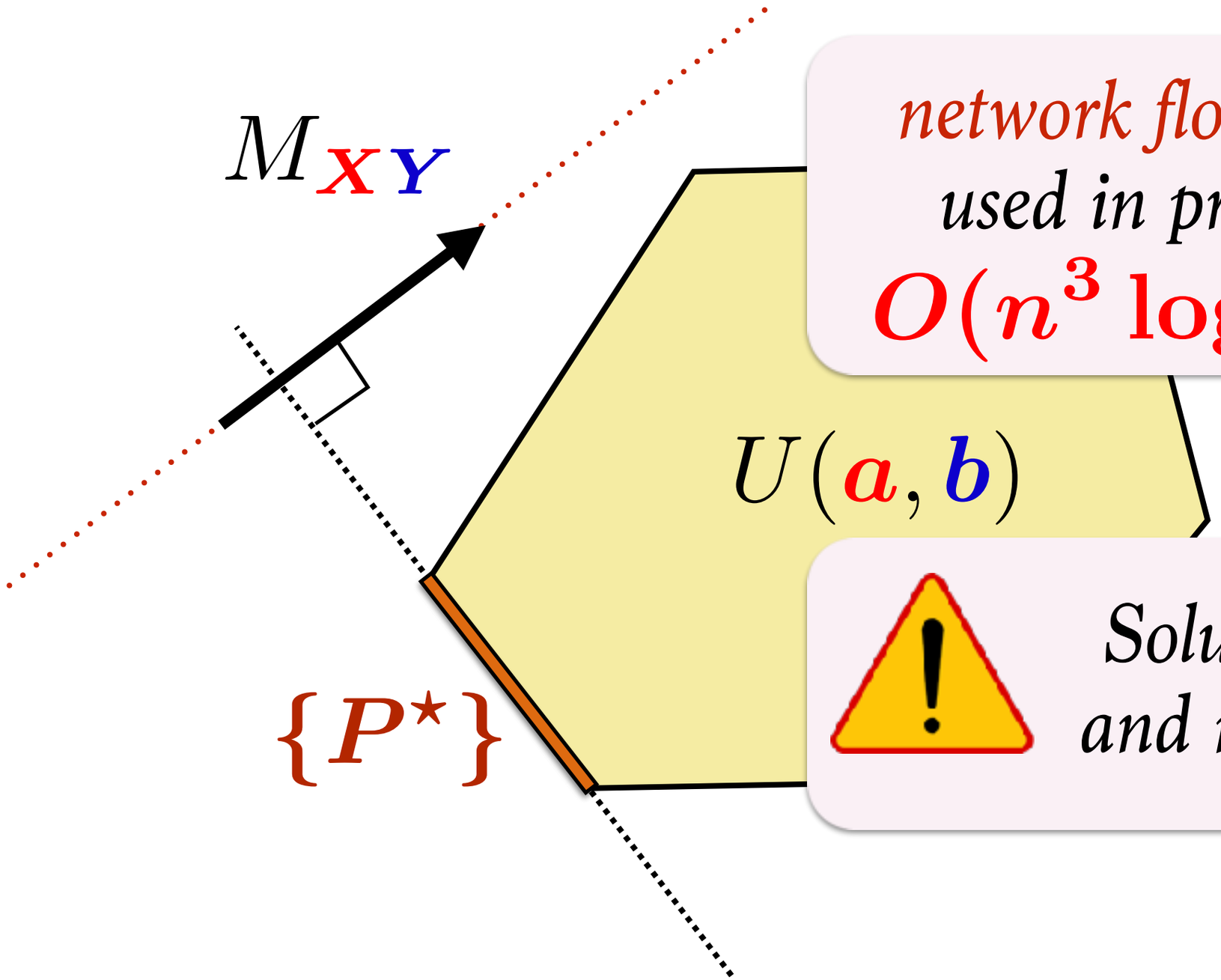
Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

Solution $P^\star$ unstable
and not always unique.

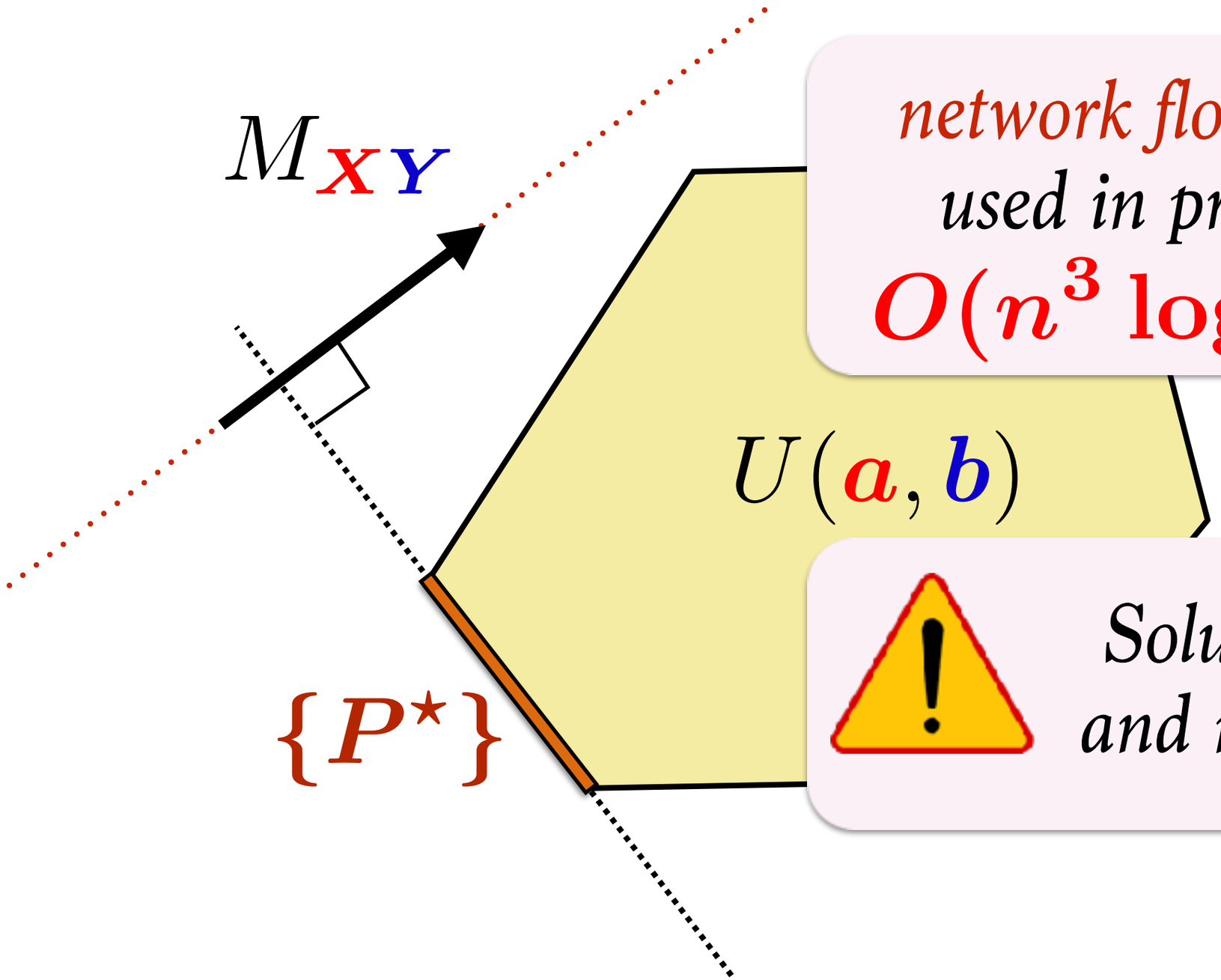
Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

Solution $P^\star$ unstable
and not always unique.

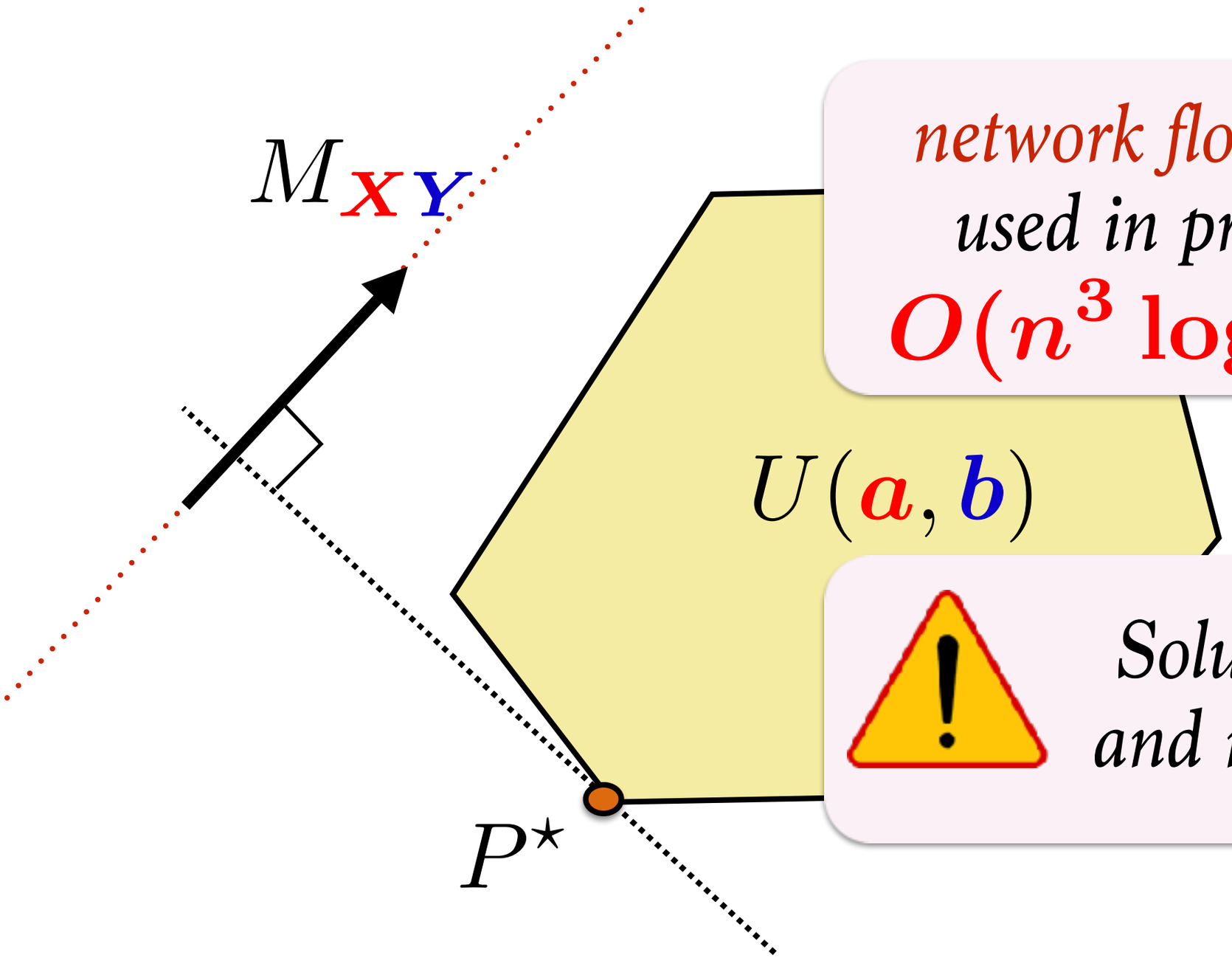
Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

Solution $P^\star$ unstable
and not always unique.

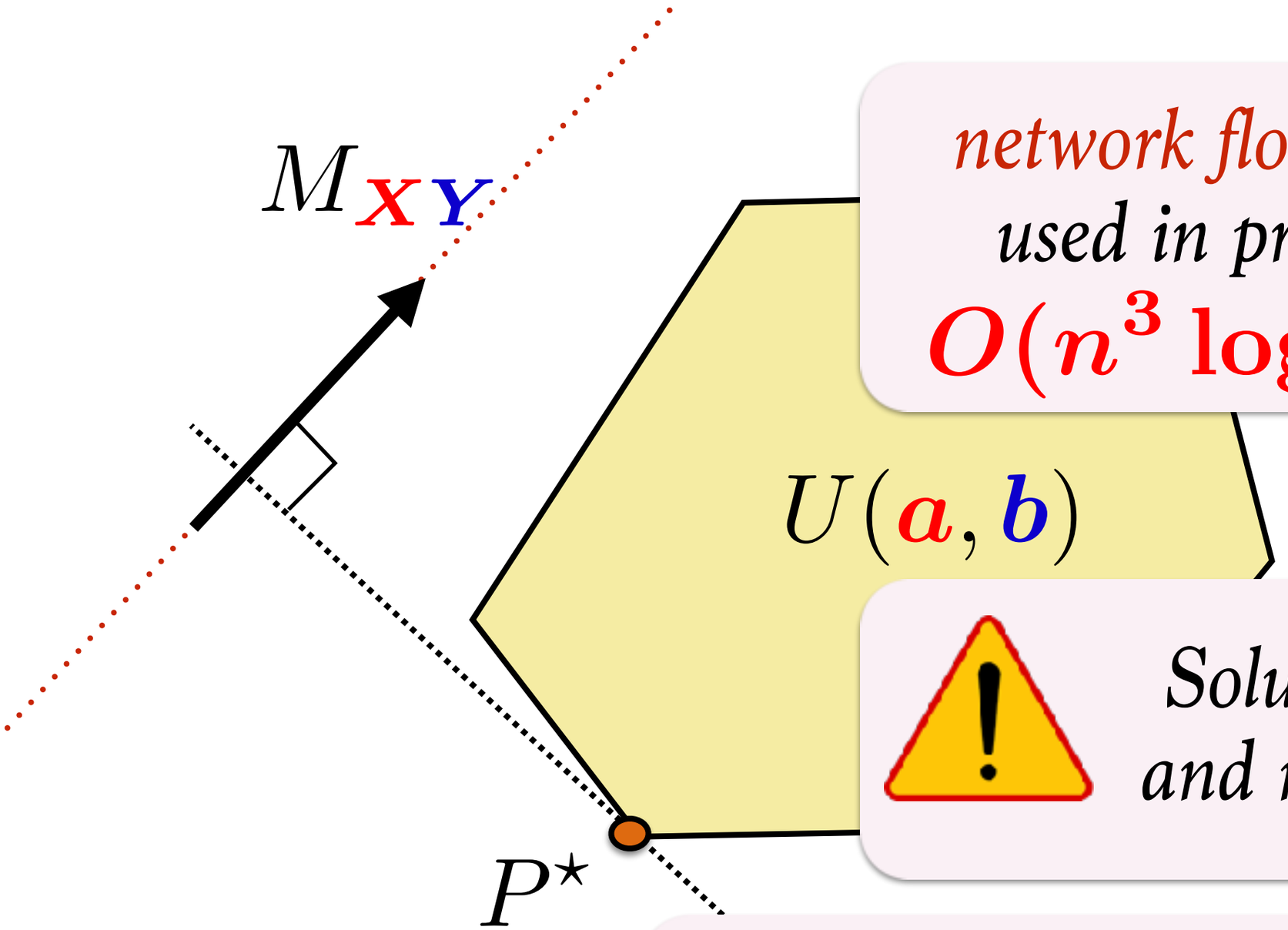
Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

Solution $P^\star$ unstable
and not always unique.

Discrete OT Problem



network flow solver
used in practice.
$O(n^3 \log(n))$

Solution $P^\star$ unstable and not always unique.

$W_p^p(\mu, \nu)$ not differentiable.

Entropic Regularization [Wilson'62]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$$

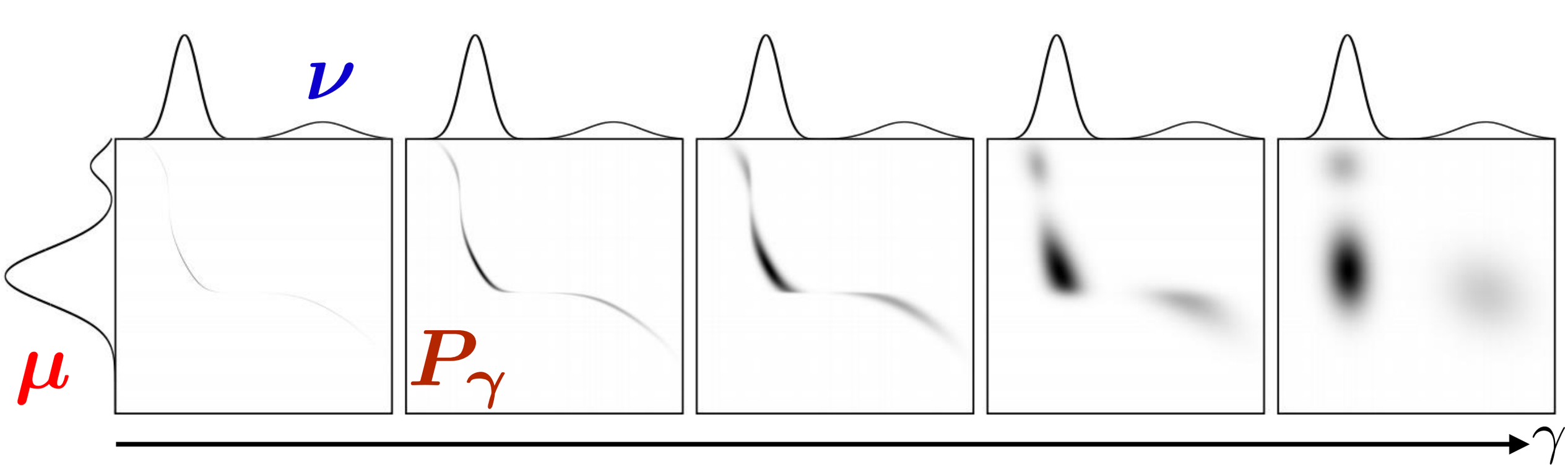
$$E(P) \stackrel{\text{def}}{=} -\sum_{i,j=1}^{nm} P_{ij}(\log P_{ij})$$

Note: Unique optimal solution because of strong concavity of Entropy

Entropic Regularization [Wilson'62]

Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$$



Note: Unique optimal solution because of strong concavity of Entropy

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists ! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

$$L(P, \alpha, \beta) = \sum_{ij} P_{ij} M_{ij} + \gamma P_{ij} \log P_{ij} + \alpha^T (P\mathbf{1} - \boldsymbol{a}) + \beta^T (P^T \mathbf{1} - \boldsymbol{b})$$

$$\partial L / \partial P_{ij} = M_{ij} + \gamma(\log P_{ij} + 1) + \alpha_i + \beta_j$$

$$(\boldsymbol{\partial L / \partial P_{ij}} = 0) \Rightarrow P_{ij} = e^{\frac{\alpha_i}{\gamma} + \frac{1}{2}} \; e^{-\frac{M_{ij}}{\gamma}} \; e^{\frac{\beta_j}{\gamma} + \frac{1}{2}} = \boldsymbol{u_i} \; K_{ij} \boldsymbol{v_j}$$

Fast & Scalable Algorithm

Prop. If $P_\gamma \stackrel{\text{def}}{=} \underset{\boldsymbol{P} \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$

then $\exists ! \boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \mathbf{diag}(\boldsymbol{u}) K \mathbf{diag}(\boldsymbol{v}), \quad K \stackrel{\text{def}}{=} e^{-M_{\boldsymbol{XY}}/\gamma}$$

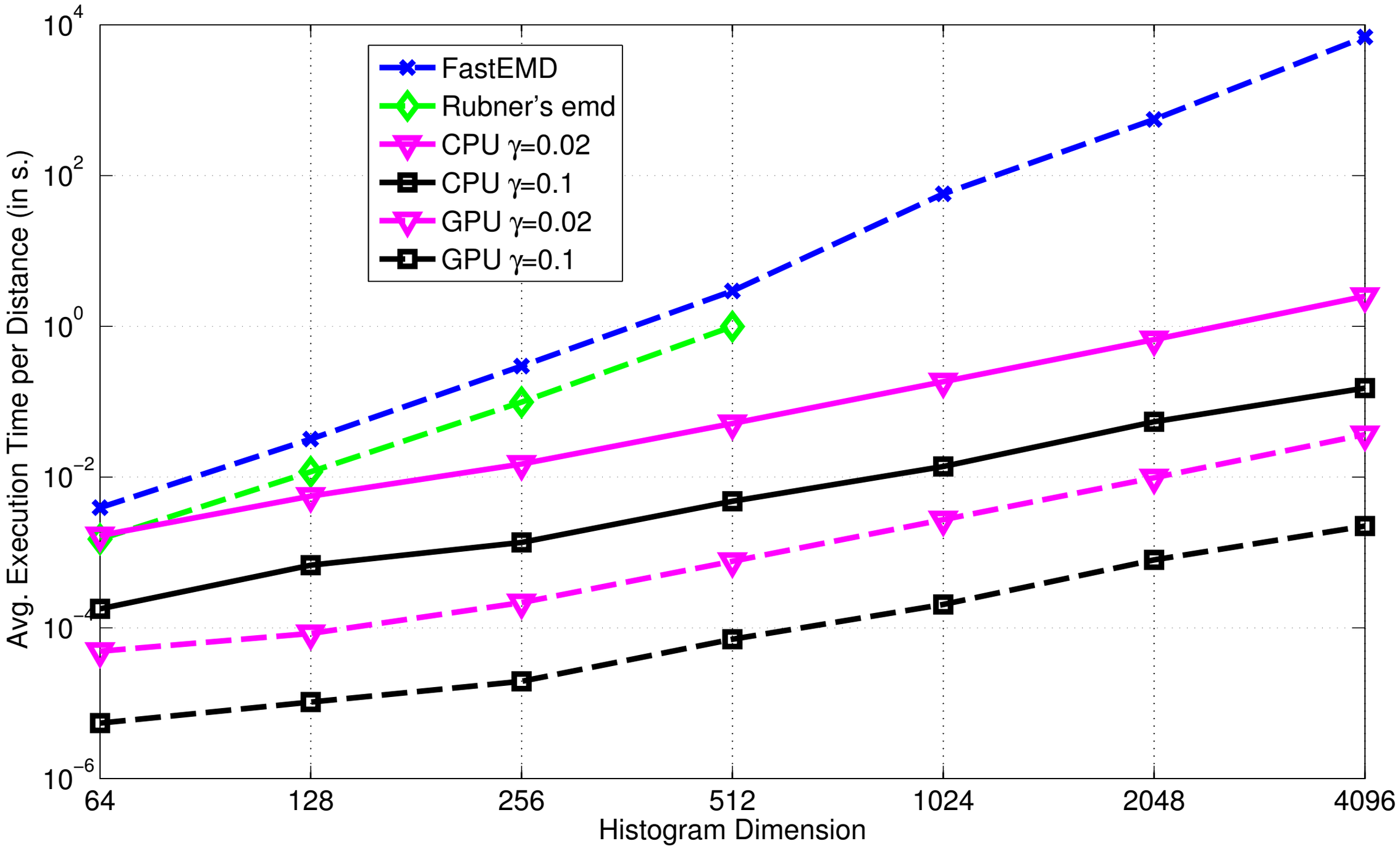
- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$

- $O(nm)$ complexity, GPGPU parallel **[C'13]** .
- $O(n^{d+1})$ if $\Omega = \{1, \ldots, n\}^d$ and $\boldsymbol{D}^p$ separable.

[S..C..'15]

Very Fast EMD Approx. Solver

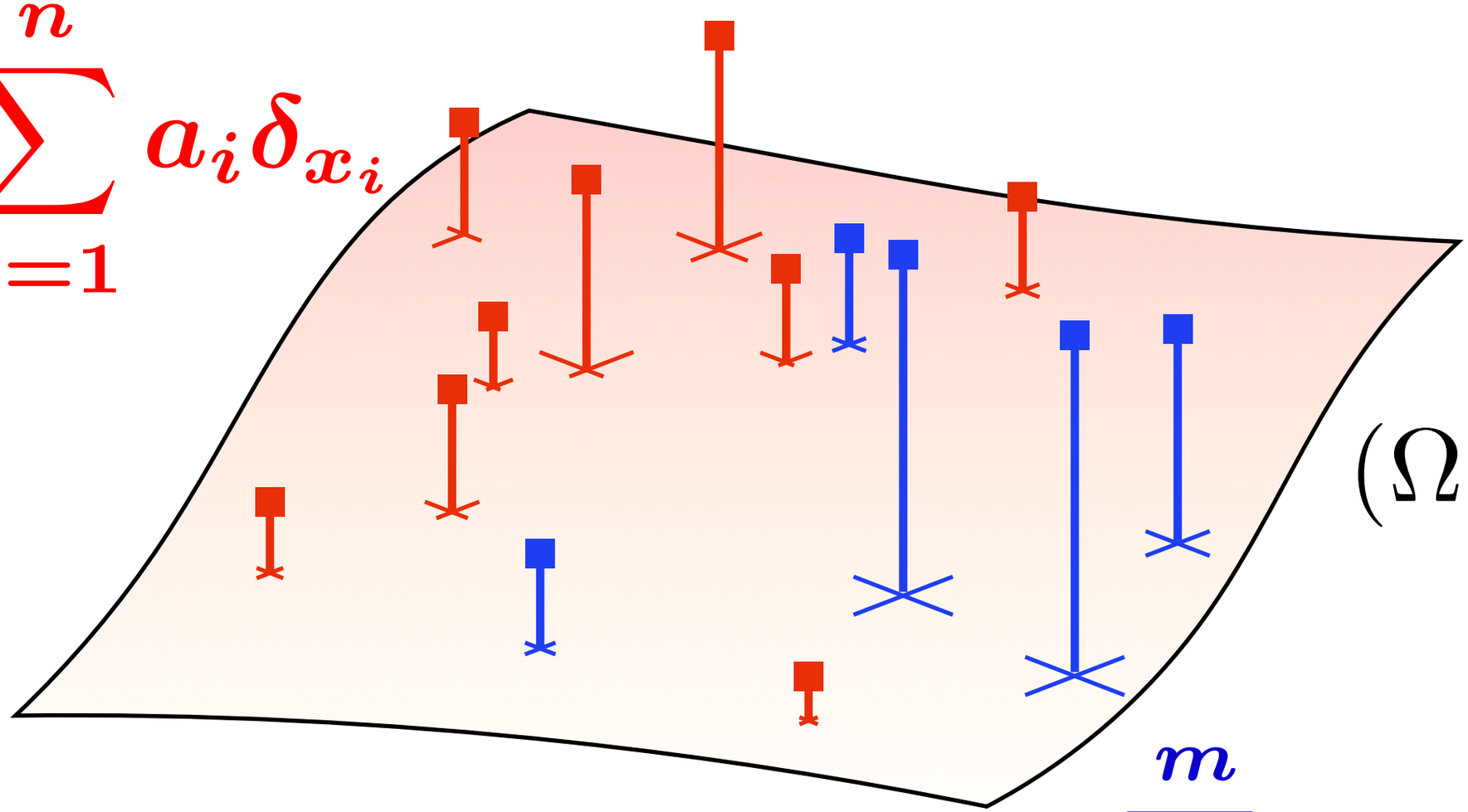


Note. $(\Omega, \boldsymbol{D})$ is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance 10^{-2}.

Regularization ⇢ *Differentiability*

$$W_\gamma((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y})) = \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle - \gamma E(\boldsymbol{P})$$

$$\boldsymbol{\mu} = \sum_{i=1}^{\boldsymbol{n}} \boldsymbol{a}_i \boldsymbol{\delta}_{\boldsymbol{x}_i}$$



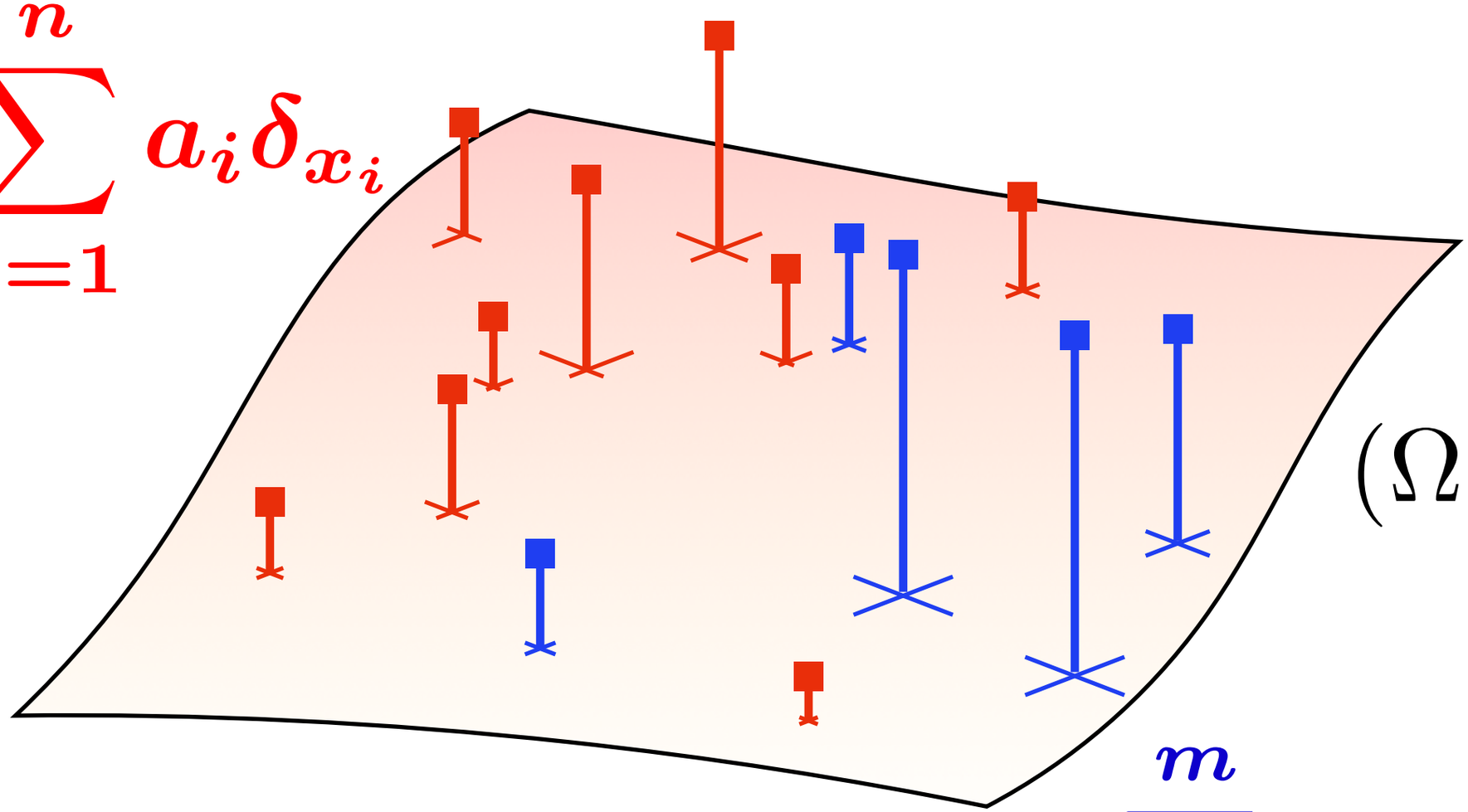
$$(\Omega, \boldsymbol{D})$$

$$\boldsymbol{\nu} = \sum_{\boldsymbol{j}=1}^{\boldsymbol{m}} \boldsymbol{b}_j \boldsymbol{\delta}_{\boldsymbol{y}_j}$$

Regularization ⇢ *Differentiability*

$$W_\gamma((a + \Delta a, X), (b, Y)) = W_\gamma((a, X), (b, Y)) + ??$$

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$$



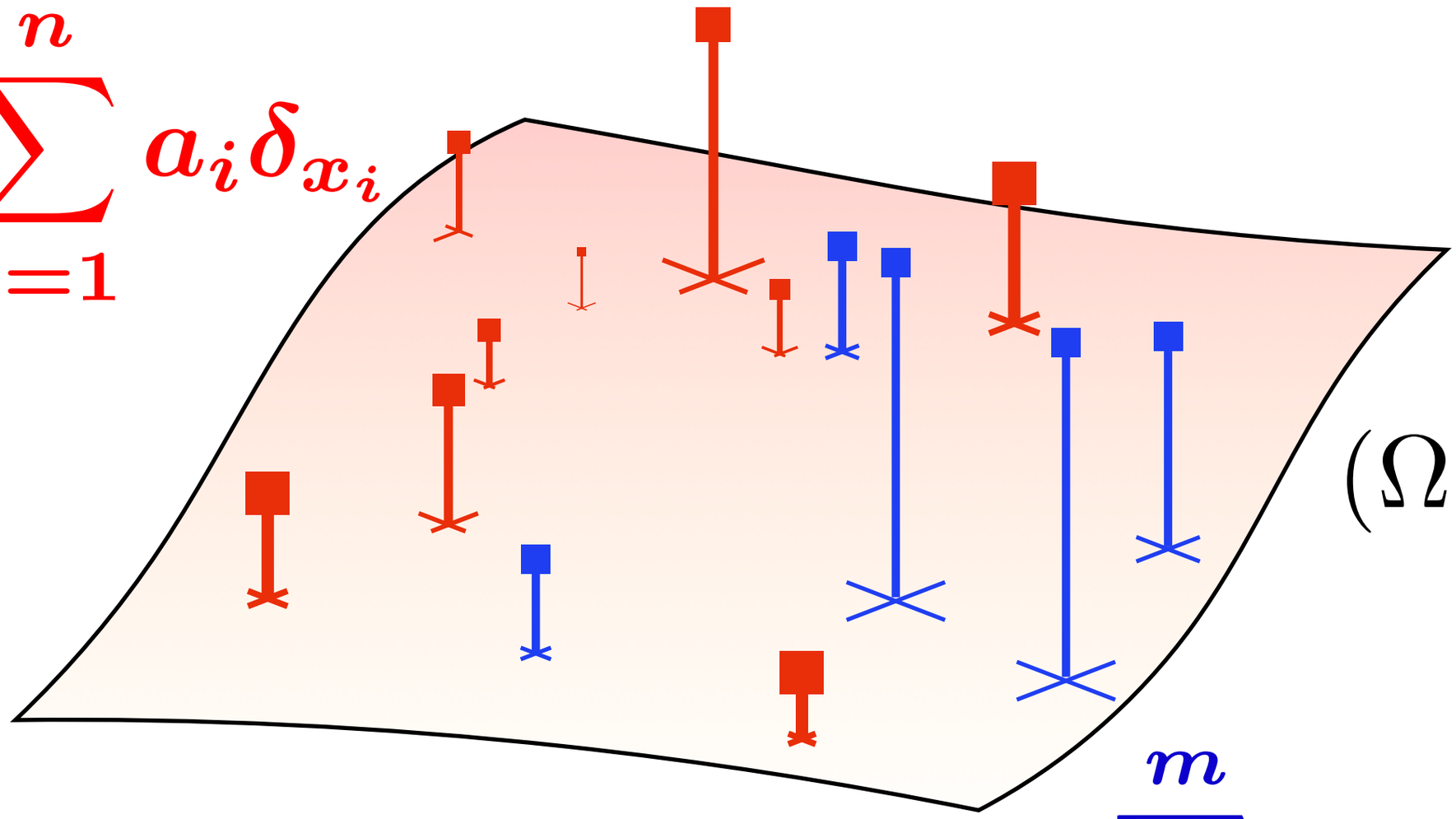
$$(\Omega, D)$$

$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

Regularization ⇢ *Differentiability*

$$W_\gamma((a + \Delta a, X), (b, Y)) = W_\gamma((a, X), (b, Y)) + ??$$

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$$



$$(\Omega, D)$$

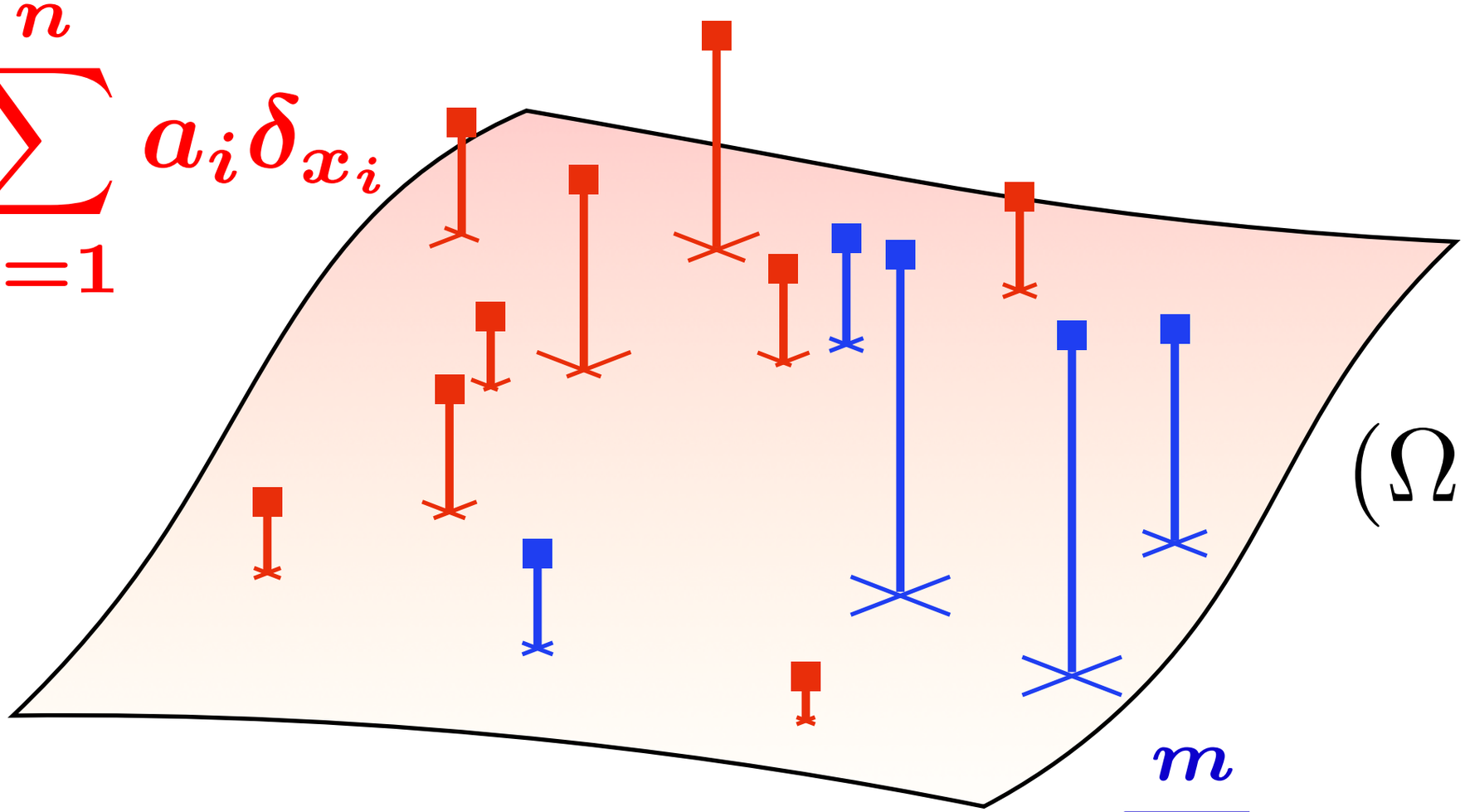
$$a \leftarrow a + \Delta a$$

$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

Regularization ⇢ *Differentiability*

$$W_\gamma((\boldsymbol{a}, \boldsymbol{X} + \boldsymbol{\Delta X}), (\boldsymbol{b}, \boldsymbol{Y})) = W_\gamma((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y})) + ??$$

$$\boldsymbol{\mu} = \sum_{i=1}^{\boldsymbol{n}} \boldsymbol{a_i \delta_{x_i}}$$



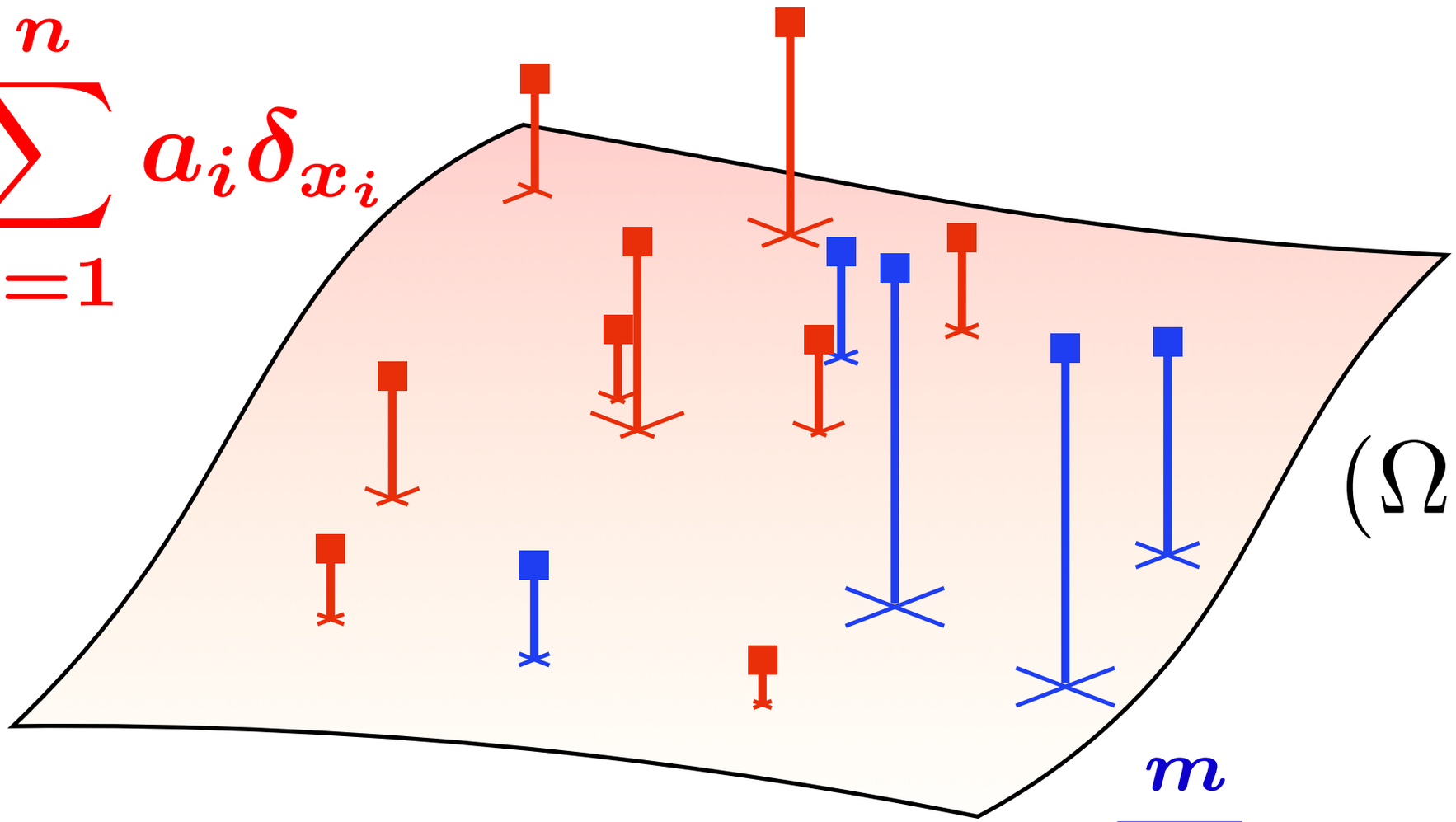
$$(\Omega, \boldsymbol{D})$$

$$\boldsymbol{\nu} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{m}} \boldsymbol{b_j \delta_{y_j}}$$

Regularization ⇢ *Differentiability*

$$W_\gamma((\boldsymbol{a}, \boldsymbol{X} + \boldsymbol{\Delta X}), (\boldsymbol{b}, \boldsymbol{Y})) = W_\gamma((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y})) + ??$$

$$\boldsymbol{\mu} = \sum_{i=1}^{\boldsymbol{n}} \boldsymbol{a_i \delta_{x_i}}$$



$$(\Omega, \boldsymbol{D})$$

$$\boldsymbol{X} \leftarrow \boldsymbol{X} + \boldsymbol{\Delta X}$$

$$\boldsymbol{\nu} = \sum_{\boldsymbol{j}=1}^{\boldsymbol{m}} \boldsymbol{b_j \delta_{y_j}}$$

Crucial for "min *data* + W" problems

- Quantization, k-means problem **[Lloyd'82]**

$$\min_{\substack{\boldsymbol{\mu}\in\mathcal{P}(\mathbb{R}^d)\\ |\operatorname{supp}\boldsymbol{\mu}|=k}} W_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}_{\text{data}})$$

- **[McCann'95]** Interpolant

$$\min_{\boldsymbol{\mu}\in\mathcal{P}(\Omega)} (1-t)W_2^2(\boldsymbol{\mu}, \boldsymbol{\nu_1}) + tW_2^2(\boldsymbol{\mu}, \boldsymbol{\nu_2})$$

- **[JKO'98]** PDE's as gradient flows in $(\mathcal{P}(\Omega), W)$.

$$\mu_{t+1} = \operatorname*{argmin}_{\boldsymbol{\mu}\in\mathcal{P}(\Omega)} J(\boldsymbol{\mu}) + \lambda_t W_p^p(\boldsymbol{\mu}, \mu_t)$$

Crucial for "min *data* + *W*" problems

Any (ML) problem involving a **KL** or **L2** loss between (parameterized) histograms or probabilility measures can be easily *Wasserstein-ized* if we can differentiate *W* efficiently.

1. Differentiability of Regularized OT

Def. Dual regularized OT Problem

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\alpha,\beta} \alpha^T \boldsymbol{a} + \beta^T \boldsymbol{b} - \frac{1}{\gamma}(e^{\alpha/\gamma})^T \, \boldsymbol{K} e^{\beta/\gamma}$$

Prop. $W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu})$ is **[CD'14]**

1. convex w.r.t. $\boldsymbol{a}$ (Danskin),

$$\nabla_{\boldsymbol{a}} W_\gamma = \alpha^\star = \gamma \log(\boldsymbol{u}).$$

2. decreased, when $p = 2, \Omega = \mathbb{R}^d$, using

$$\boldsymbol{X} \leftarrow \boldsymbol{Y} P_\gamma^T \mathbf{D}(\boldsymbol{a}^{-1}).$$

2. Duality for Regularized OT's

Prop. Writing $H_{\boldsymbol{\nu}} : \boldsymbol{a} \mapsto W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu})$, **[CP'16]**

1. $H_{\boldsymbol{\nu}}$ has simple Legendre transform:

$$H^*_{\boldsymbol{\nu}} : \boldsymbol{g} \in \mathbb{R}^n \mapsto \gamma \left(E(\boldsymbol{b}) + \boldsymbol{b}^T \log(\boldsymbol{K} e^{\boldsymbol{g}/\gamma}) \right)$$

2. If $A \in \mathbb{R}^{n \times d}$, f convex on $\mathbb{R}^d$,

$$\min_{\boldsymbol{a} \in \Sigma_n} H_{\boldsymbol{\nu}}(\boldsymbol{a}) + f(A\boldsymbol{a}) = \max_{\boldsymbol{g} \in \mathbb{R}^d} -H^*_{\boldsymbol{\nu}}(A^T \boldsymbol{g}) - f^*(-\boldsymbol{g})$$

3. Stochastic Formulation

$$
\begin{aligned}
W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) &= \max_{\alpha,\beta} \alpha^T \boldsymbol{a} + \beta^T \boldsymbol{b} - \frac{1}{\gamma}(e^{\alpha/\gamma})^T \; \boldsymbol{K} e^{\beta/\gamma} \\
&= \max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \boldsymbol{a} - \gamma(\log \; \boldsymbol{K} e^{\boldsymbol{\alpha}/\gamma})^T \boldsymbol{b} \\
&= \max_{\boldsymbol{\alpha}} \sum_{j=1}^{m} \boldsymbol{b_j} \left(\boldsymbol{\alpha}^T \boldsymbol{a} - \gamma \log \; \boldsymbol{K}_{\cdot \boldsymbol{j}}^T e^{\boldsymbol{\alpha}/\gamma} \right) \\
&= \max_{\boldsymbol{\alpha}} \sum_{j=1}^{m} \; \boldsymbol{f_j}(\boldsymbol{\alpha})
\end{aligned}
$$

- **[GCPB'16]** shows that incremental gradient methods are competitive with Sinkhorn.

4. Algorithmic Formulation

Def. For $L \geq 1$, define

$$W_L(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \langle \boldsymbol{P_L}, M_{\boldsymbol{XY}} \rangle,$$

where $\boldsymbol{P_L} \stackrel{\text{def}}{=} \mathbf{diag}(\boldsymbol{u_L}) K \mathbf{diag}(\boldsymbol{v_L})$,

$\boldsymbol{v_0} = \mathbf{1}_m; l \geq 0, \boldsymbol{u_l} \stackrel{\text{def}}{=} \boldsymbol{a}/K\boldsymbol{v_l}, \boldsymbol{v_{l+1}} \stackrel{\text{def}}{=} \boldsymbol{b}/K^T\boldsymbol{u_l}.$

Prop. $\frac{\partial W_L}{\partial \boldsymbol{X}}, \frac{\partial W_L}{\partial \boldsymbol{a}}$ can be computed recursively, in $O(L)$ kernel $K \times$vector products.

Algorithmic Formulation of Reg. OT

Example: Differentiability w.r.t. a

$$\left(\frac{\partial \boldsymbol{v_0}}{\partial a}\right)^T = \mathbf{0}_{m\times n},$$

$$\left(\frac{\partial \boldsymbol{u_l}}{\partial a}\right)^T \boldsymbol{x} = \frac{\boldsymbol{x}}{K\boldsymbol{v_l}} - \left(\frac{\partial \boldsymbol{v_l}}{\partial a}\right)^T K^T \frac{\boldsymbol{x}\circ a}{(K\boldsymbol{v_l})^2},$$

$$\left(\frac{\partial \boldsymbol{v_{l+1}}}{\partial a}\right)^T \boldsymbol{y} = -\left(\frac{\partial \boldsymbol{u_l}}{\partial a}\right)^T K \frac{\boldsymbol{y}\circ b}{(K^T\boldsymbol{u_l})^2}.$$

Algorithmic Formulation of Reg. OT

Example: Differentiability w.r.t. a

$$N = K \circ M_{\boldsymbol{XY}}$$

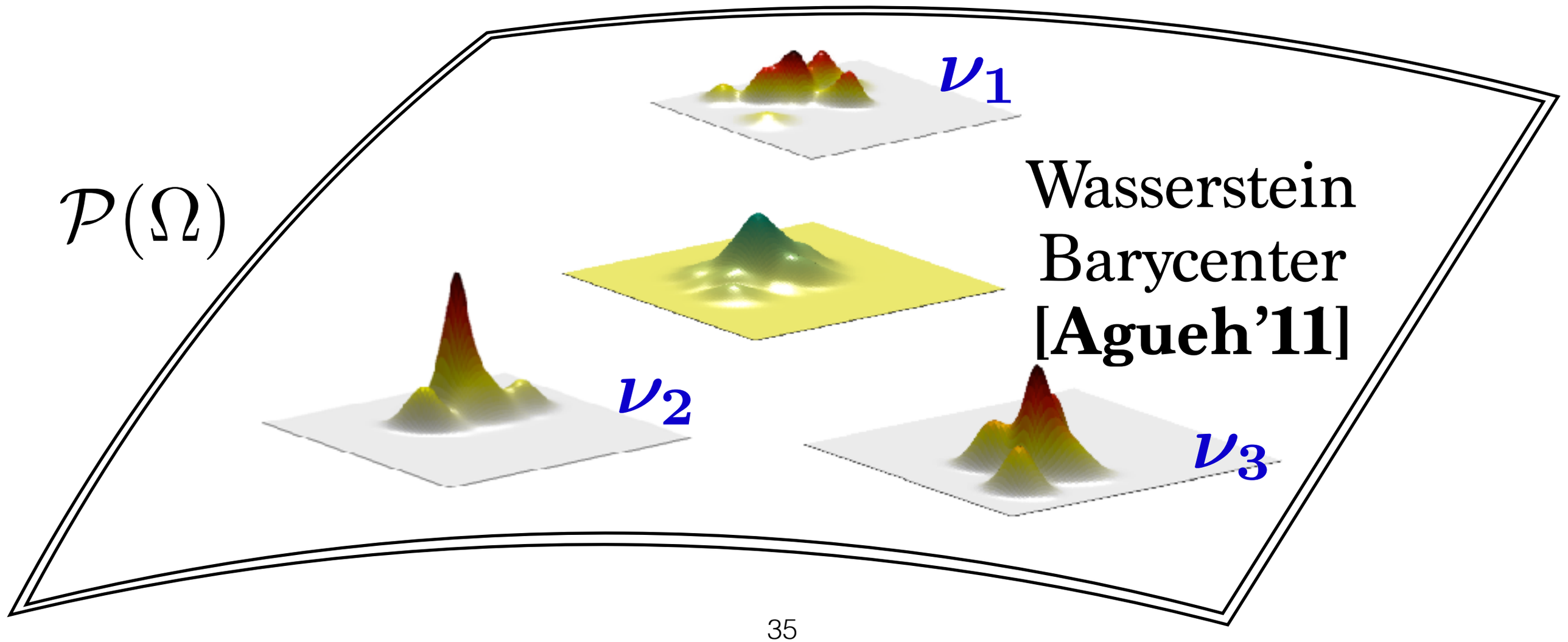
$$\nabla_{\boldsymbol{a}} W_L(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left(\frac{\partial \boldsymbol{u_L}}{\partial a}\right)^T N \boldsymbol{v_L} + \left(\frac{\partial \boldsymbol{v_L}}{\partial a}\right)^T N^T \boldsymbol{u_L}$$

Thanks to these tricks...

- **[Agueh'11]** Barycenters **[CD'14][BCCNP'15] [GCP'15][S..C..'15]**
- **[Burger'12]** TV gradient flow using duality **[CP'16]**
- Dictionary Learning / Latent Factors **[RCP'16]**
- **[Bigot'15]** *W*-PCA **[SC'15]**
- Inverse problems / Wasserstein regression **[BPC'16]**
- Density fitting / parameter estimation **[MMC'16]**

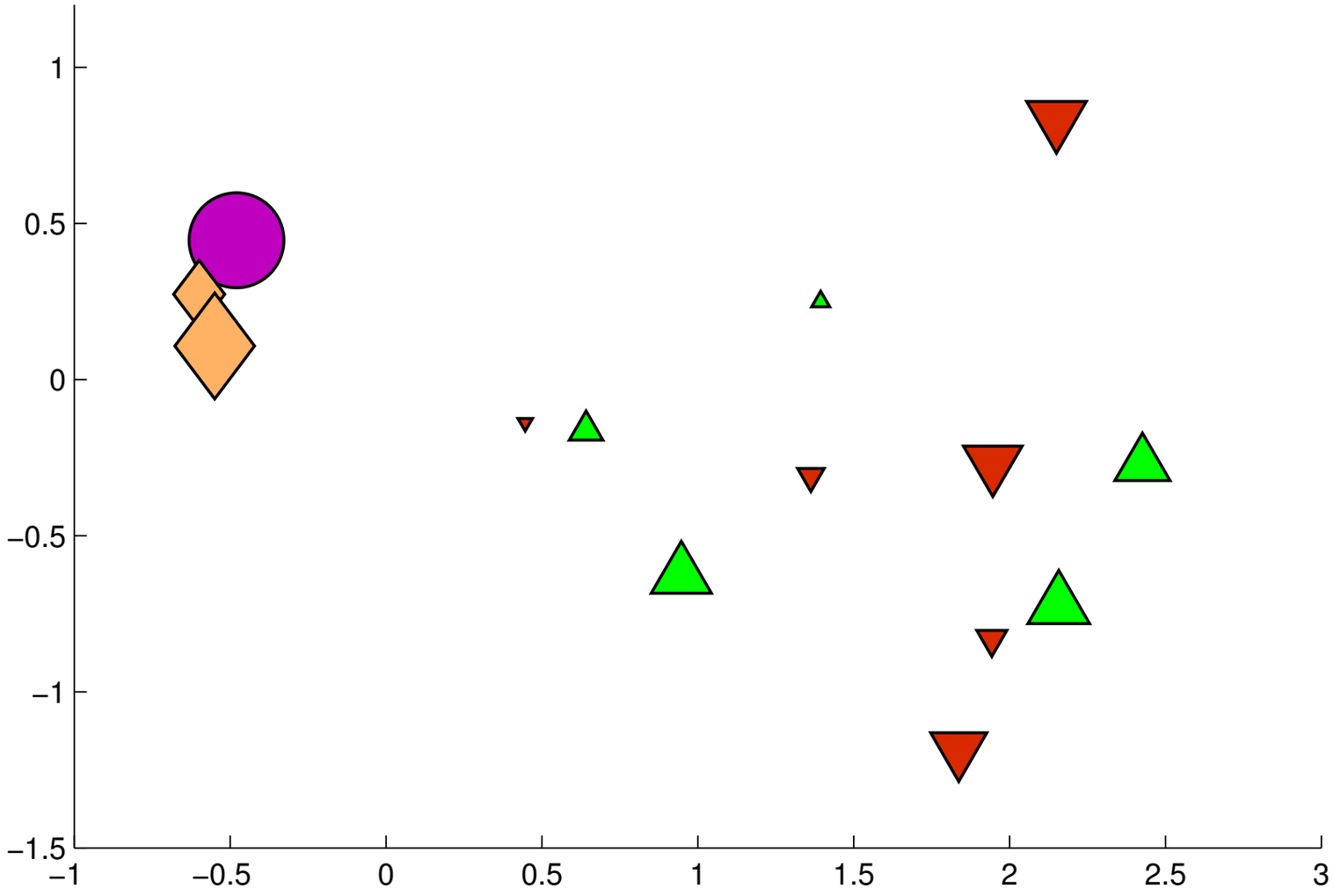
Wasserstein Barycenters

$$\min_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$



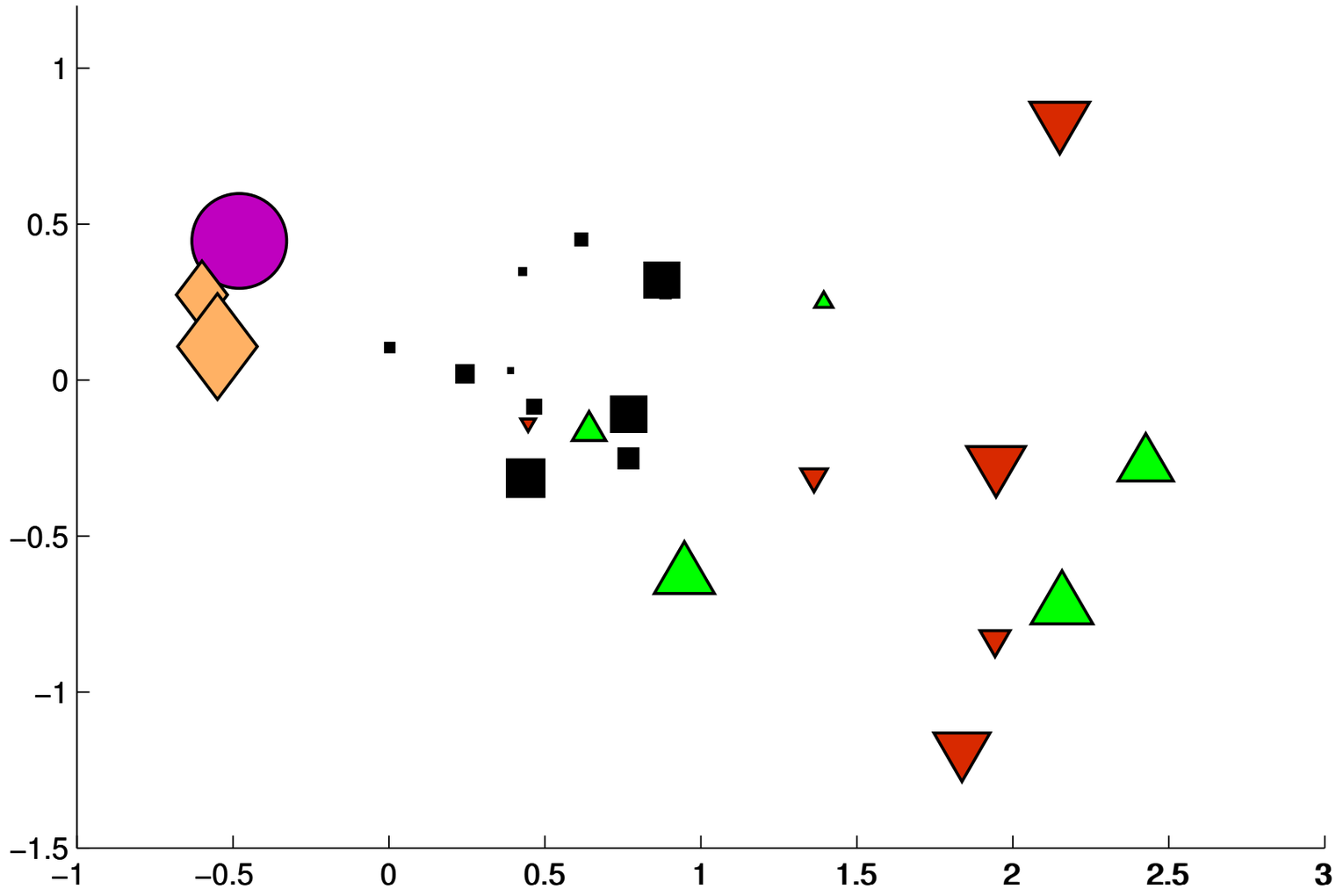
Multimarginal Formulation

- Exact solution (W_2) using MM-OT. **[Agueh'11]**



Multimarginal Formulation

- Exact solution (W_2) using MM-OT. **[Agueh'11]**



$$\text{If } |\operatorname{supp} \boldsymbol{\nu_i}| = \boldsymbol{n_i}, \text{ LP of size } \left(\prod_i \boldsymbol{n_i}, \sum_i \boldsymbol{n_i}\right)$$

Finite Case, LP Formulation

- When Ω is a finite set, metric M, another LP.

$$\min_{\boldsymbol{\mu}} \sum_i \lambda_i W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$

Finite Case, LP Formulation

- When Ω is a finite set, metric M, another LP.

$$\min_{\boldsymbol{P_1},\cdots,\boldsymbol{P_N},\boldsymbol{a}} \sum_{i=1}^{N} \lambda_i \langle \boldsymbol{P_i}, M \rangle$$

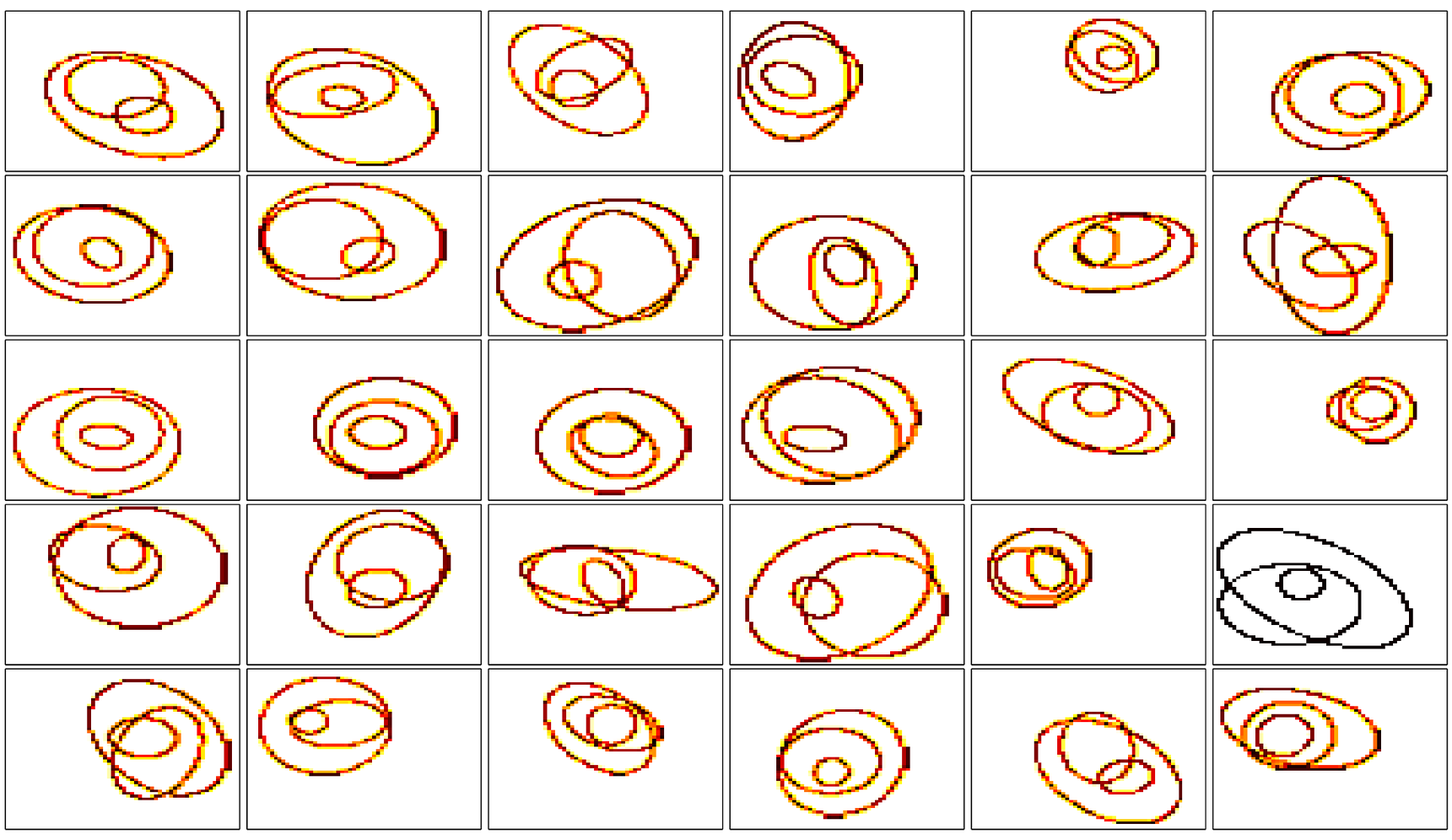
$$\text{s.t. } \boldsymbol{P_i}^T \mathbf{1}_n = \boldsymbol{b_i}, \forall i \leq N,$$

$$\boldsymbol{P_1}\mathbf{1}_n = \cdots = \boldsymbol{P_N}\mathbf{1}_d = \boldsymbol{a}.$$

If $|\Omega| = n$, LP of size $(Nn^2, (2N-1)n)$; unstable

Primal Descent on Regularized W

$$\min_{\boldsymbol{\mu}\in\boldsymbol{Q}\subset\mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_{\boldsymbol{\gamma}}(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$

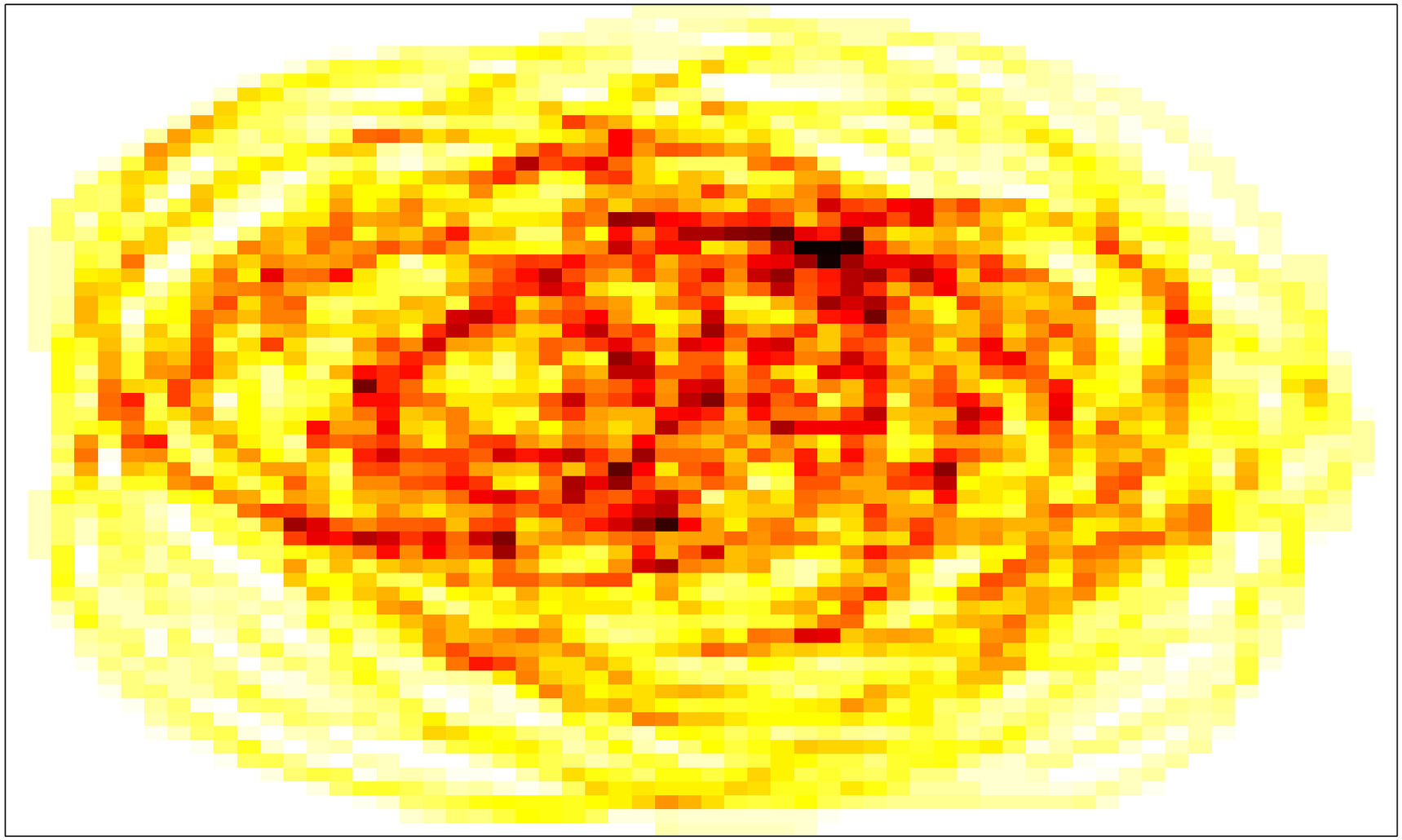


Fast Computation of Wasserstein Barycenters
International Conference on Machine Learning 2014

[CD'14]

Primal Descent on Regularized W

$$\min_{\boldsymbol{\mu} \in \boldsymbol{Q} \subset \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_{\boldsymbol{\gamma}}(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$

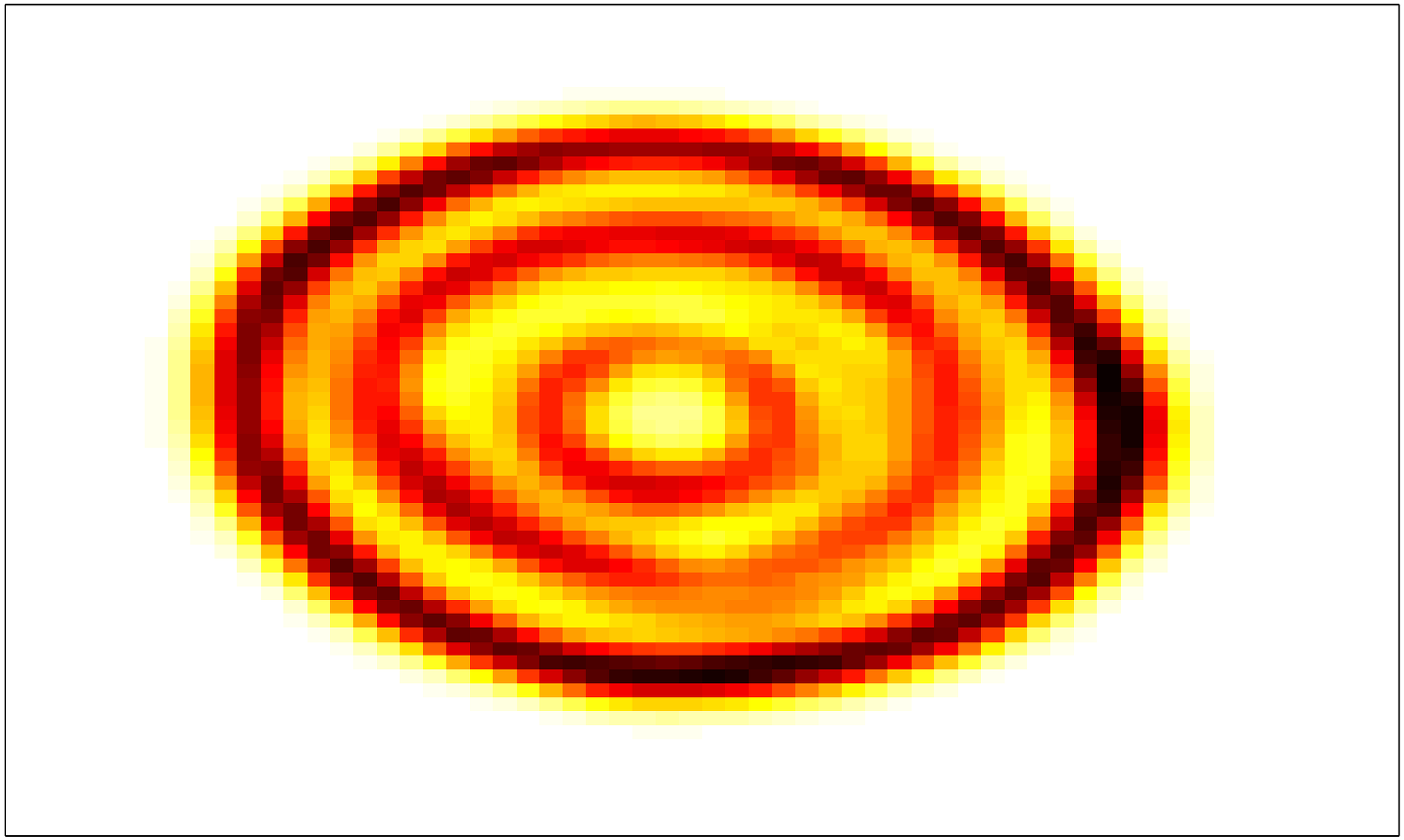


Fast Computation of Wasserstein Barycenters
International Conference on Machine Learning 2014

[CD'14]

Primal Descent on Regularized W

$$\min_{\mu \in Q \subset \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_\gamma(\mu, \nu_i)$$



Fast Computation of Wasserstein Barycenters
International Conference on Machine Learning 2014

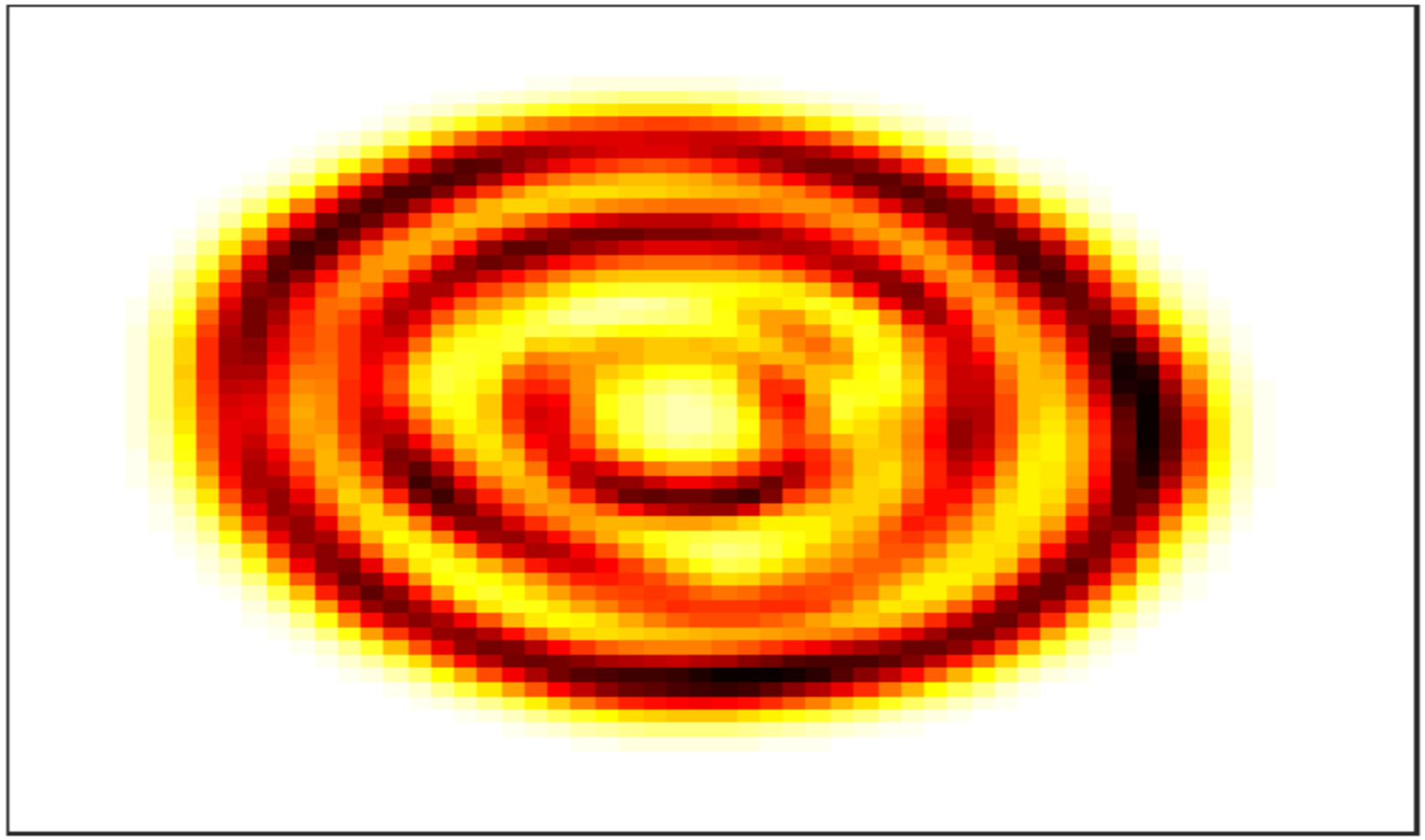
[CD'14]

Primal Descent on Algorithmic W

$$\min_{\boldsymbol{\mu} \in \boldsymbol{Q} \subset \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_{\boldsymbol{L}}(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$

Primal Descent on Algorithmic W

$$\min_{\boldsymbol{\mu} \in \boldsymbol{Q} \subset \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_{\boldsymbol{L}}(\boldsymbol{\mu}, \boldsymbol{\nu_i})$$



Wasserstein Barycenter = KL Projections

$$\langle P, M_{\boldsymbol{XY}} \rangle - \gamma E(P) = \gamma \mathbf{KL}(P|\ K)$$

$$\min_{\boldsymbol{a}} \sum_{i=1}^{N} \lambda_i W_\gamma(\boldsymbol{a}, \boldsymbol{b_i}) = \min_{\substack{\mathbf{P}=[\boldsymbol{P_1},\ldots,\boldsymbol{P_N}] \\ \mathbf{P}\in \boldsymbol{C_1}\cap\boldsymbol{C_2}}} \sum_{i=1}^{N} \lambda_i \mathbf{KL}(\boldsymbol{P_i}|K)$$

$$\boldsymbol{C_1} = \{\mathbf{P}|\exists \boldsymbol{a}, \forall i, P_i \mathbf{1}_m = \boldsymbol{a}\}$$

$$\boldsymbol{C_2} = \left\{\mathbf{P}|\forall i, P_i^T \mathbf{1}_n = \boldsymbol{b_i}\right\}$$

Wasserstein Barycenter = KL Projections

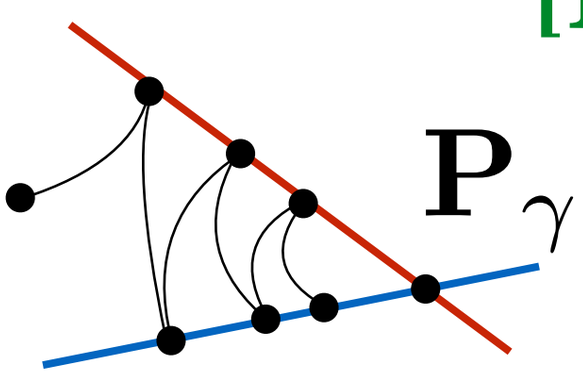
$$\min_{\boldsymbol{a}} \sum_{i=1}^{N} \lambda_i W_\gamma(\boldsymbol{a}, \boldsymbol{b_i}) = \min_{\substack{\mathbf{P}=[\boldsymbol{P_1},\ldots,\boldsymbol{P_N}] \\ \mathbf{P}\in \boldsymbol{C_1}\cap\boldsymbol{C_2}}} \sum_{i=1}^{N} \lambda_i \mathbf{KL}(\boldsymbol{P_i}|K)$$

$$\boldsymbol{C_1} = \{\mathbf{P}|\exists \boldsymbol{a}, \forall i, P_i \mathbf{1}_m = \boldsymbol{a}\}$$

$$\boldsymbol{C_2} = \{\mathbf{P}|\forall i, P_i^T \mathbf{1}_n = \boldsymbol{b_i}\}$$

[BCCNP'15]

$$[\ K \cdots\ K]$$



$$\mathbf{P}_\gamma$$

Wasserstein Barycenter = KL Projections

$$\min_{\boldsymbol{a}} \sum_{i=1}^{N} \lambda_i W_\gamma(\boldsymbol{a}, \boldsymbol{b_i}) = \min_{\substack{\mathbf{P}=[\boldsymbol{P_1},\ldots,\boldsymbol{P_N}] \\ \mathbf{P}\in \boldsymbol{C_1}\cap\boldsymbol{C_2}}} \sum_{i=1}^{N} \lambda_i \mathbf{KL}(\boldsymbol{P_i}|K)$$

$$\boldsymbol{C_1} = \{\mathbf{P}|\exists \boldsymbol{a}, \forall i, P_i \mathbf{1}_m = \boldsymbol{a}\}$$

$$\boldsymbol{C_2} = \left\{\mathbf{P}|\forall i, P_i^T \mathbf{1}_n = \boldsymbol{b_i}\right\}$$

```
u=ones(size(B)); % d x N matrix
while not converged
   v=u.*(K'*(B./(K*u))); % 2(Nd^2) cost
   u=bsxfun(@times,u,exp(log(v)*weights))./v;
end
a=mean(v,2);
```

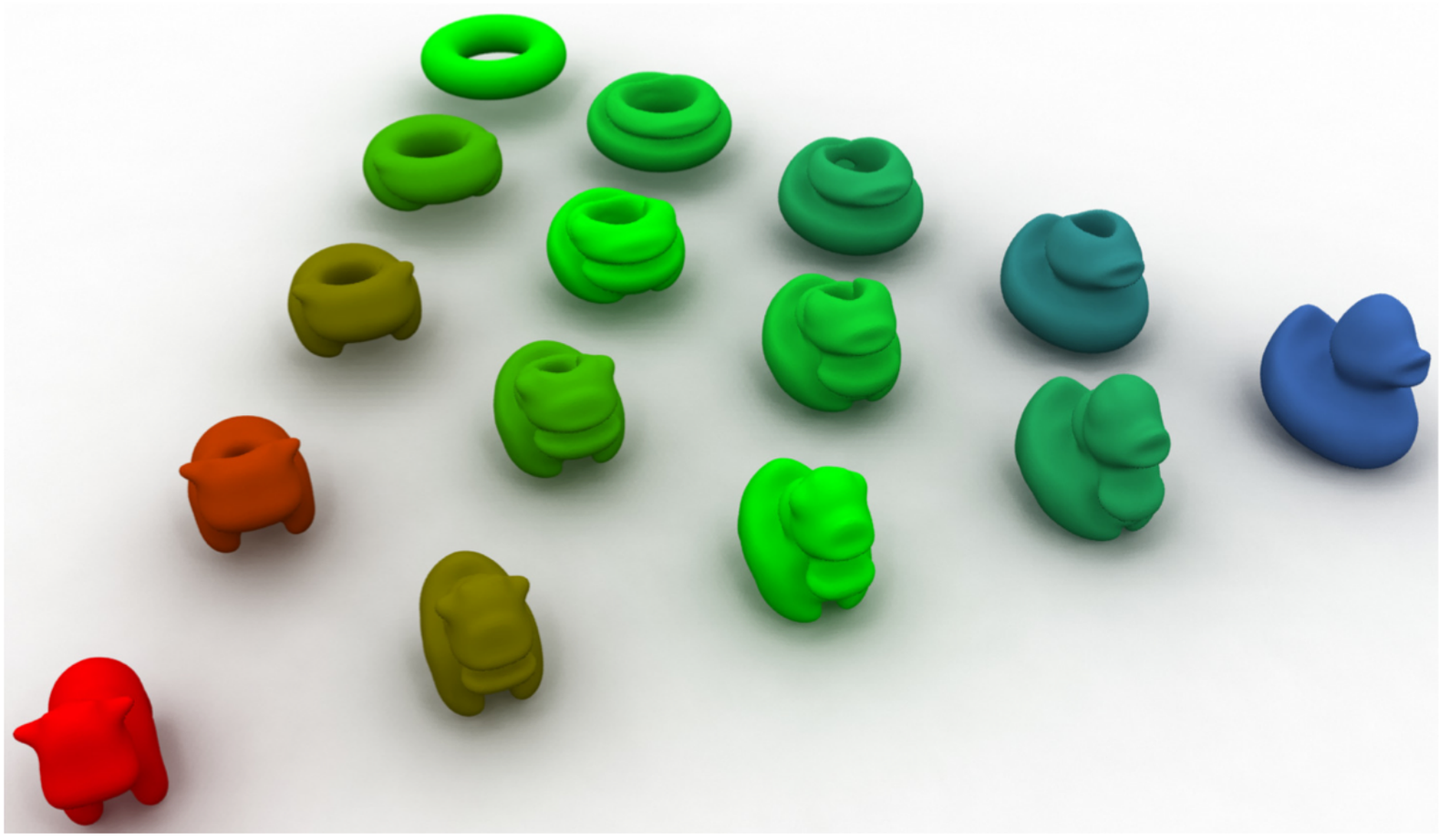
[BCCNP'15]

Iterative Bregman Projections for Regularized Transportation Problems
SIAM J. on Sci. Comp. 2015

Application: Graphics

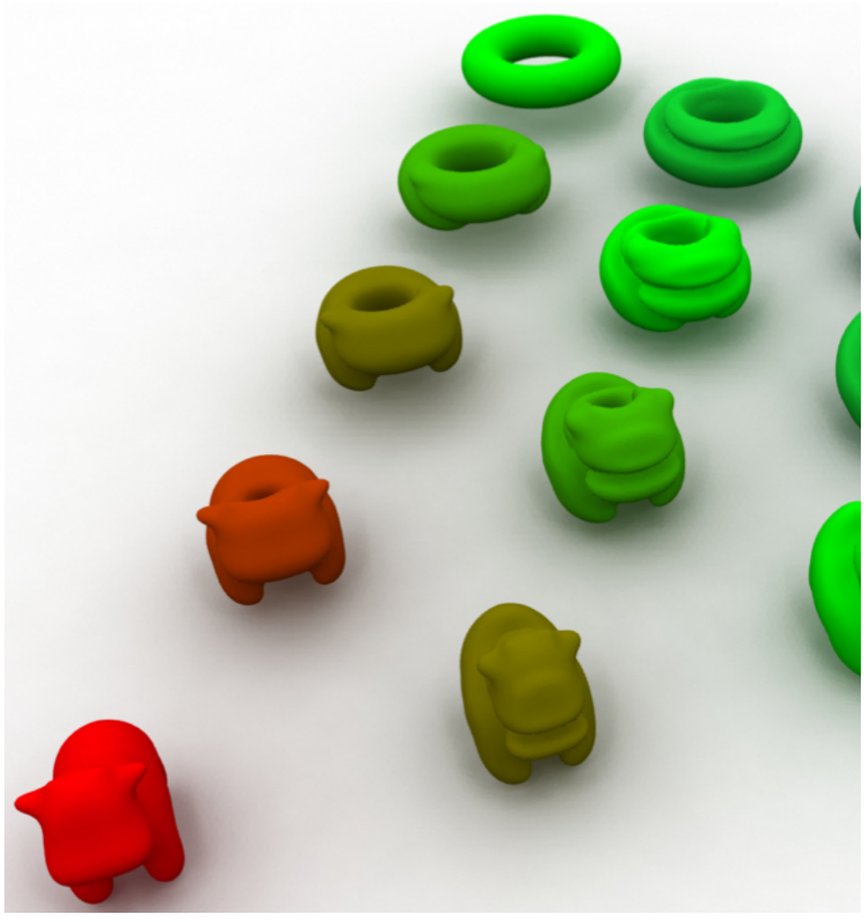
Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains,
SIGGRAPH'15 **[S..C..'15]**

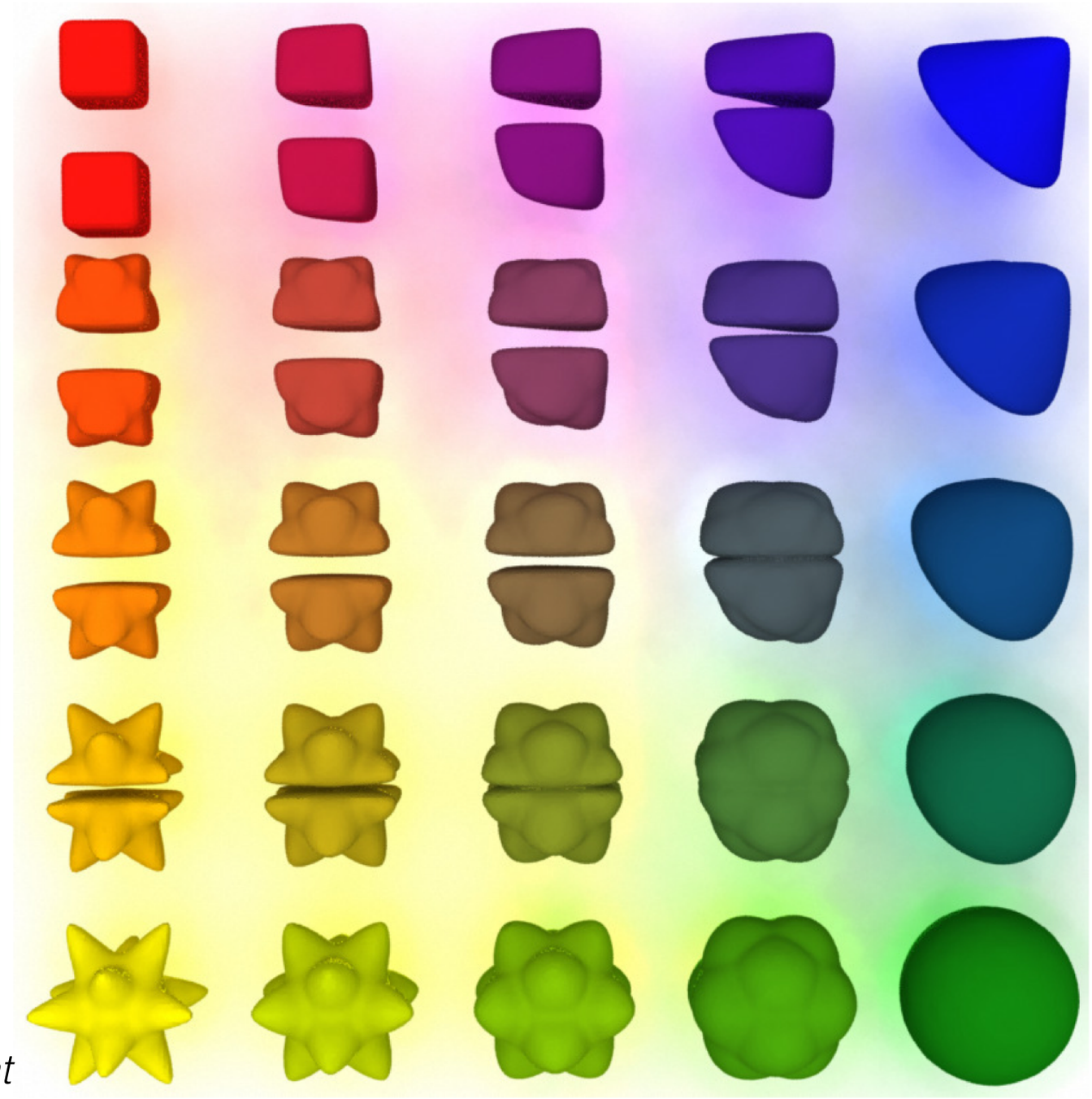
Application: Graphics



Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains,
SIGGRAPH'15 **[S..C..'15]**

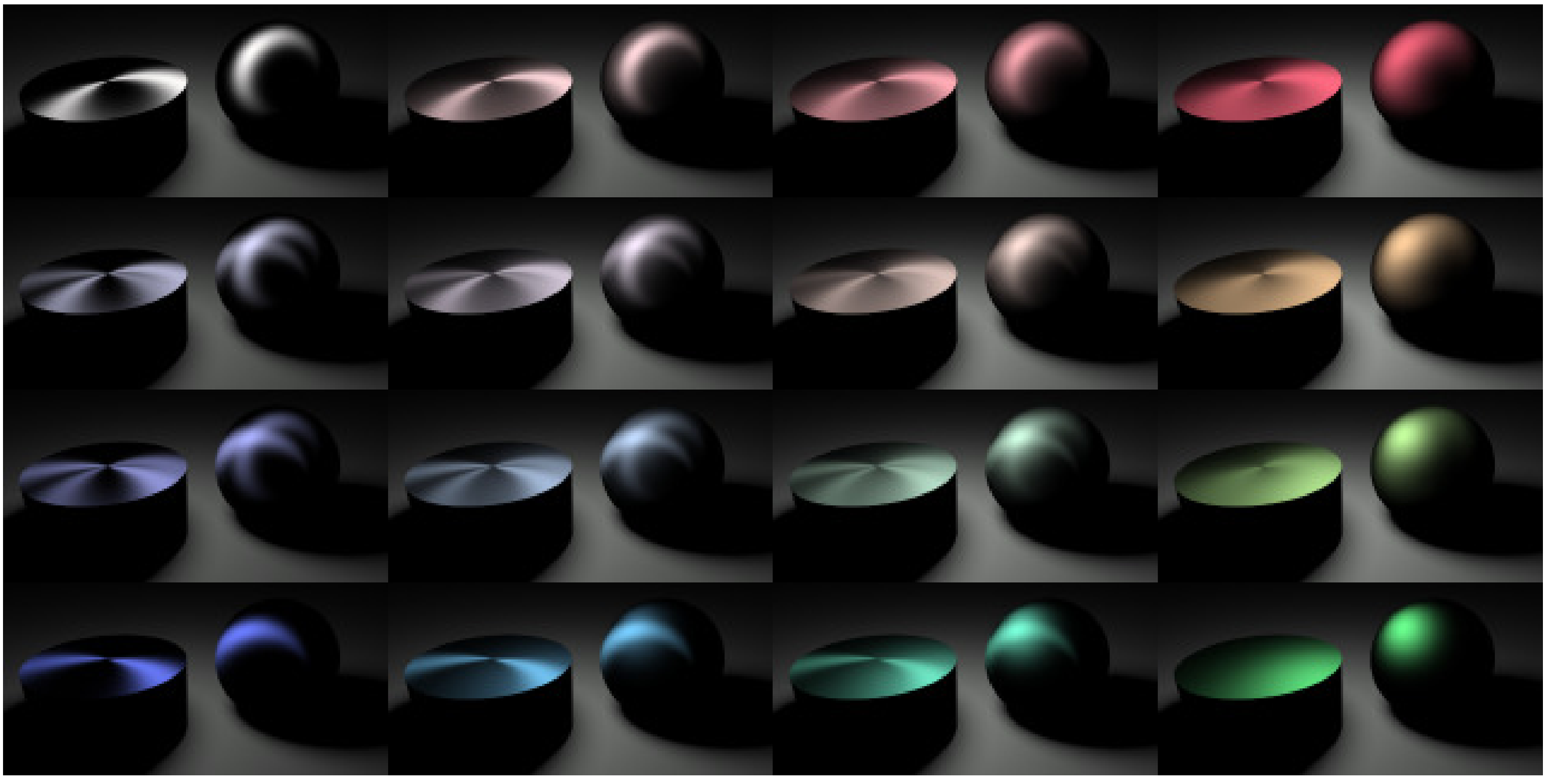
Application: Graphics





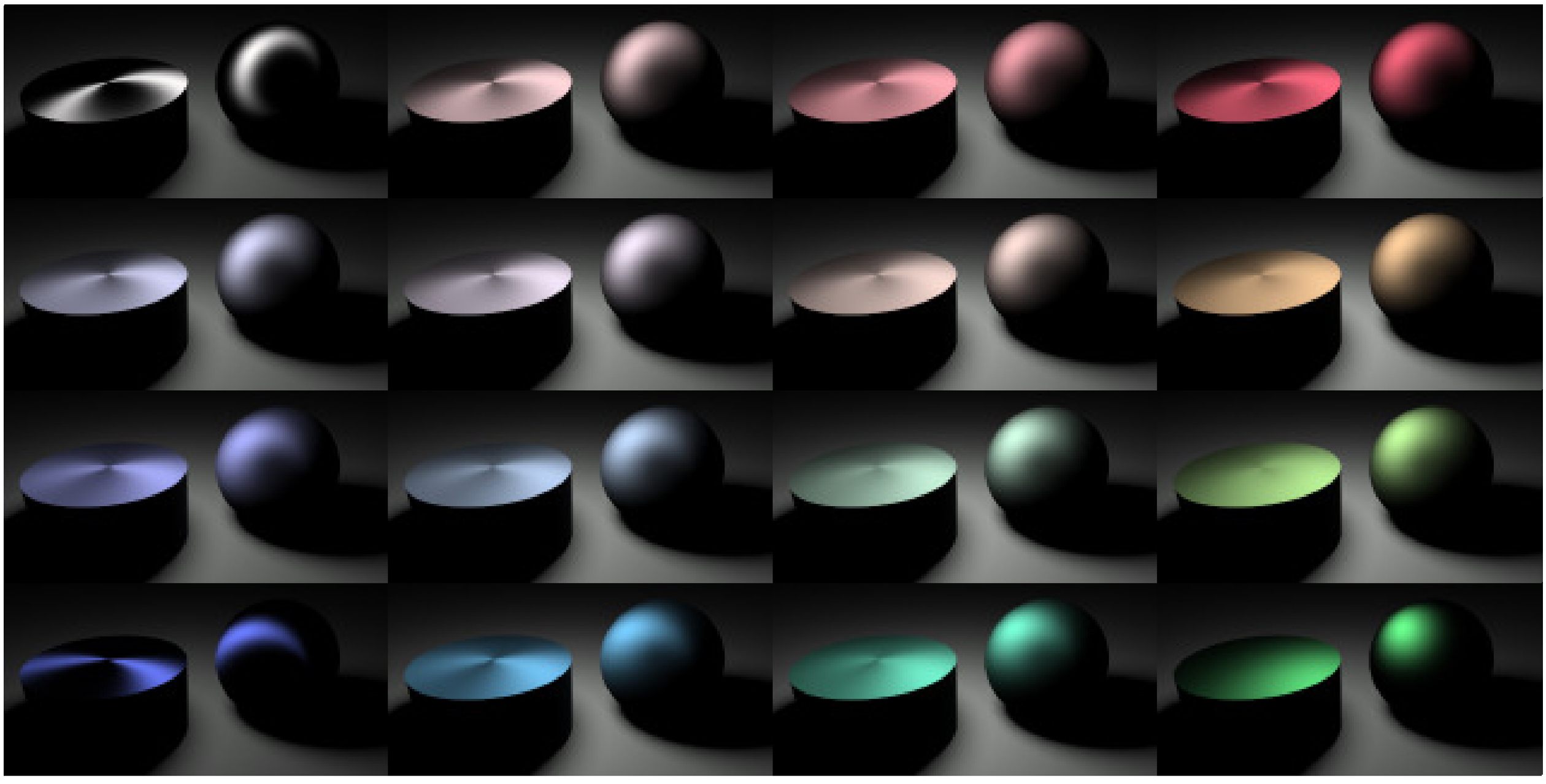
Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains,
SIGGRAPH'15 **[S..C..'15]**

Application: Graphics



Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains,
SIGGRAPH'15 **[S..C..'15]**

Application: Graphics



Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains,
SIGGRAPH'15 **[S..C..'15]**

Inverse Wasserstein Problems

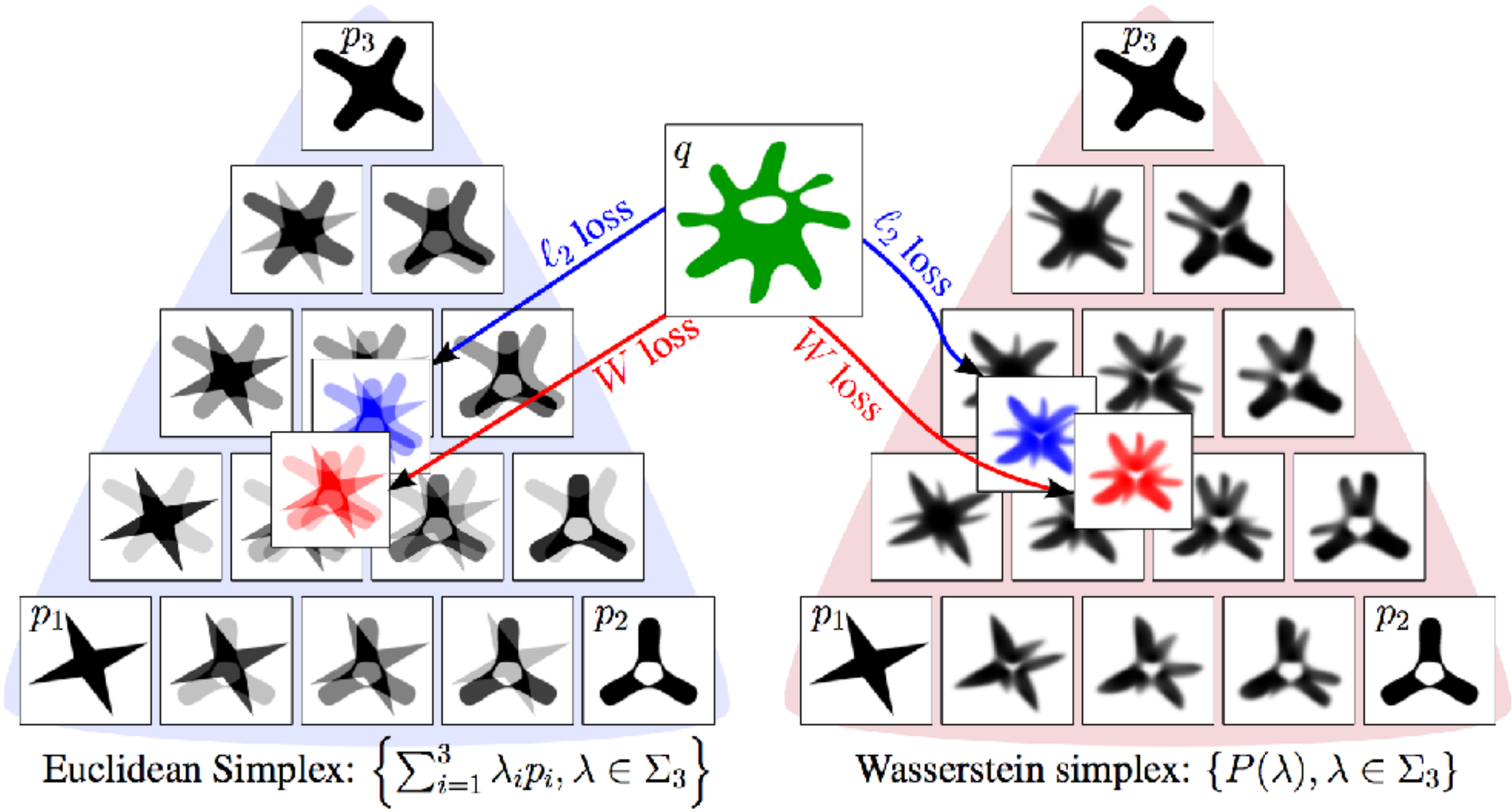
- consider Barycenter operator:

$$\boldsymbol{b}(\lambda) \stackrel{\text{def}}{=} \underset{\boldsymbol{a}}{\text{argmin}} \sum_{i=1}^{N} \lambda_i W_\gamma(\boldsymbol{a}, \boldsymbol{b_i})$$

- address now **Wasserstein inverse problems**:

$$\text{Given } \boldsymbol{a}, \text{ find } \underset{\lambda \in \Sigma_N}{\text{argmin}}\, \mathcal{E}(\lambda) \stackrel{\text{def}}{=} \text{Loss}(\boldsymbol{a}, \boldsymbol{b}(\lambda))$$

The Wasserstein Simplex



Euclidean Simplex: $\left\{\sum_{i=1}^{3} \lambda_i p_i, \lambda \in \Sigma_3\right\}$

Wasserstein simplex: $\{P(\lambda), \lambda \in \Sigma_3\}$

Barycenters = Fixed Points

Prop. **[BCCNP'15]** Consider $\boldsymbol{B} \in \Sigma_d^N$ and let $\boldsymbol{U_0} = \mathbf{1}_{\boldsymbol{d}\times \boldsymbol{N}}$, and then for $l \geq 0$:

$$\boldsymbol{b}^l \stackrel{\text{def}}{=} \exp\left(\log\left(K^T \boldsymbol{U_l}\right) \lambda\right); \begin{cases} \boldsymbol{V_{l+1}} \stackrel{\text{def}}{=} \frac{\boldsymbol{b}^l \mathbf{1}_N^T}{K^T \boldsymbol{U_l}}, \\ \boldsymbol{U_{l+1}} \stackrel{\text{def}}{=} \frac{\boldsymbol{B}}{K \boldsymbol{V_{l+1}}}. \end{cases}$$

Using Truncated Barycenters

- instead of using the exact barycenter

$$\underset{\lambda \in \Sigma_N}{\operatorname{argmin}} \mathcal{E}(\lambda) \stackrel{\text{def}}{=} \operatorname{Loss}(\boldsymbol{a}, \boldsymbol{b}(\lambda))$$

- use instead the L-iterate barycenter

$$\underset{\lambda \in \Sigma_N}{\operatorname{argmin}} \mathcal{E}^{(L)}(\lambda) \stackrel{\text{def}}{=} \operatorname{Loss}(\boldsymbol{a}, \boldsymbol{b}^{(L)}(\lambda))$$

- Differente using **the chain rule.**

$$\nabla \mathcal{E}^{(L)}(\lambda) = [\partial \boldsymbol{b}^{(L)}]^T(\boldsymbol{g}),\ \boldsymbol{g} \stackrel{\text{def}}{=} \nabla \operatorname{Loss}(\boldsymbol{a}, \cdot)|_{\boldsymbol{b}^{(L)}(\lambda)}.$$

Gradient / Barycenter Computation

function SINKHORN-DIFFERENTIATE($(p_s)_{s=1}^S, q, \lambda$)

$\forall\, s, b_s^{(0)} \leftarrow \mathbb{1}$

$(w, r) \leftarrow (0^S, 0^{S \times N})$

for $\ell = 1, 2, \ldots, L$ *// Sinkhorn loop*

$\forall\, s, \varphi_s^{(\ell)} \leftarrow K^\top \frac{p_s}{K b_s^{(\ell-1)}}$

$p \leftarrow \prod_s \left(\varphi_s^{(\ell)}\right)^{\lambda_s}$

$\forall\, s, b_s^{(\ell)} \leftarrow \frac{p}{\varphi_s^{(\ell)}}$

$g \leftarrow \nabla \mathcal{L}(p, q) \odot p$

for $\ell = L, L-1, \ldots, 1$ *// Reverse loop*

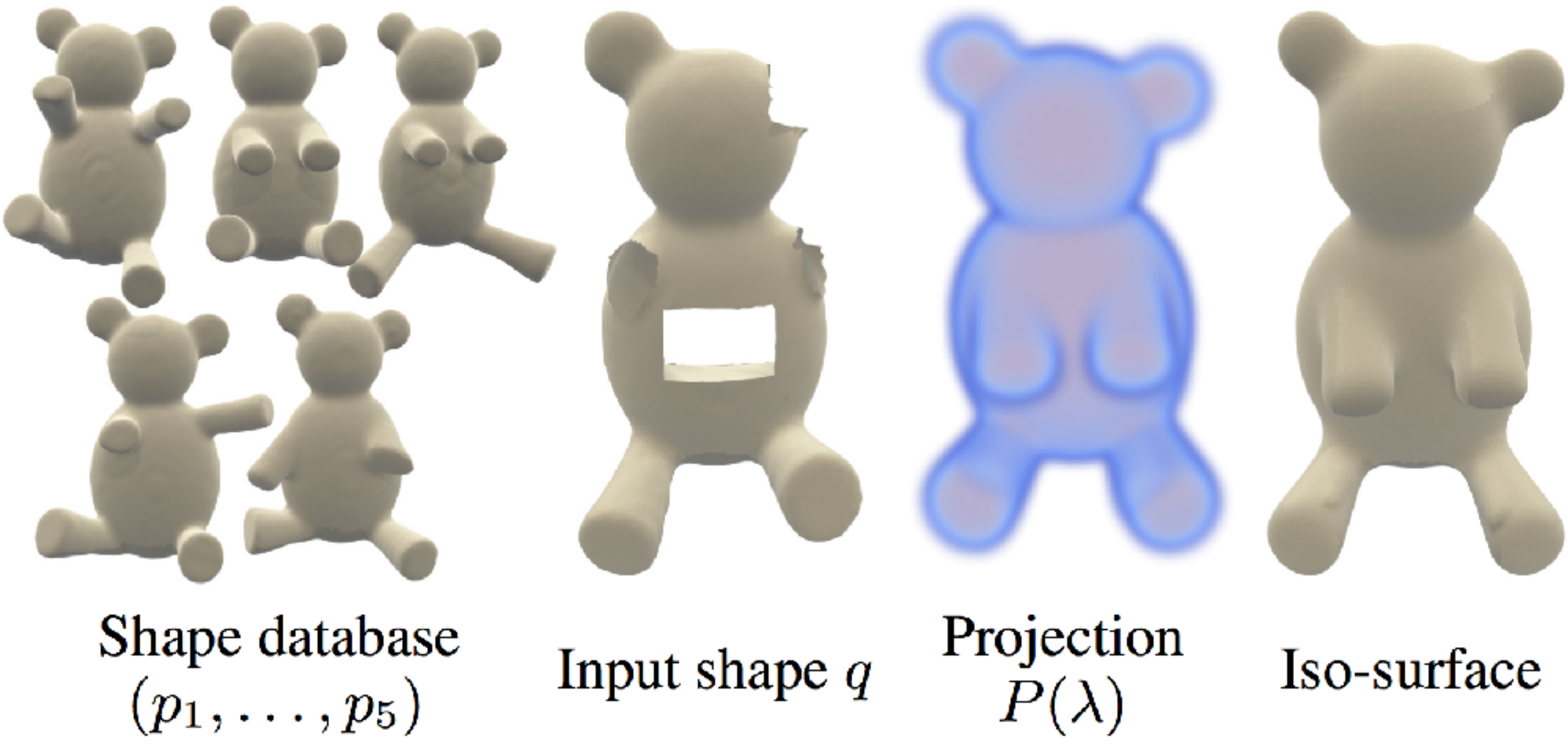
$\forall\, s, w_s \leftarrow w_s + \langle \log \varphi_s^{(\ell)}, g \rangle$

$\forall\, s, r_s \leftarrow -K^\top \left(K\left(\frac{\lambda_s g - r_s}{\varphi_s^{(\ell)}}\right) \odot \frac{p_s}{(K b_s^{(\ell-1)})^2}\right) \odot b_s^{(\ell-1)}$

$g \leftarrow \sum_s r_s$

return $P^{(L)}(\lambda) \leftarrow p, \nabla \mathcal{E}_L(\lambda) \leftarrow w$

Application: Volume Reconstruction



Wasserstein Barycentric Coordinates: Histogram Regression using Optimal Transport, **SIGGRAPH'16**

[BPC'16]

Application: Color Grading

Application: Color Grading

$\lambda_0 = 0.03$

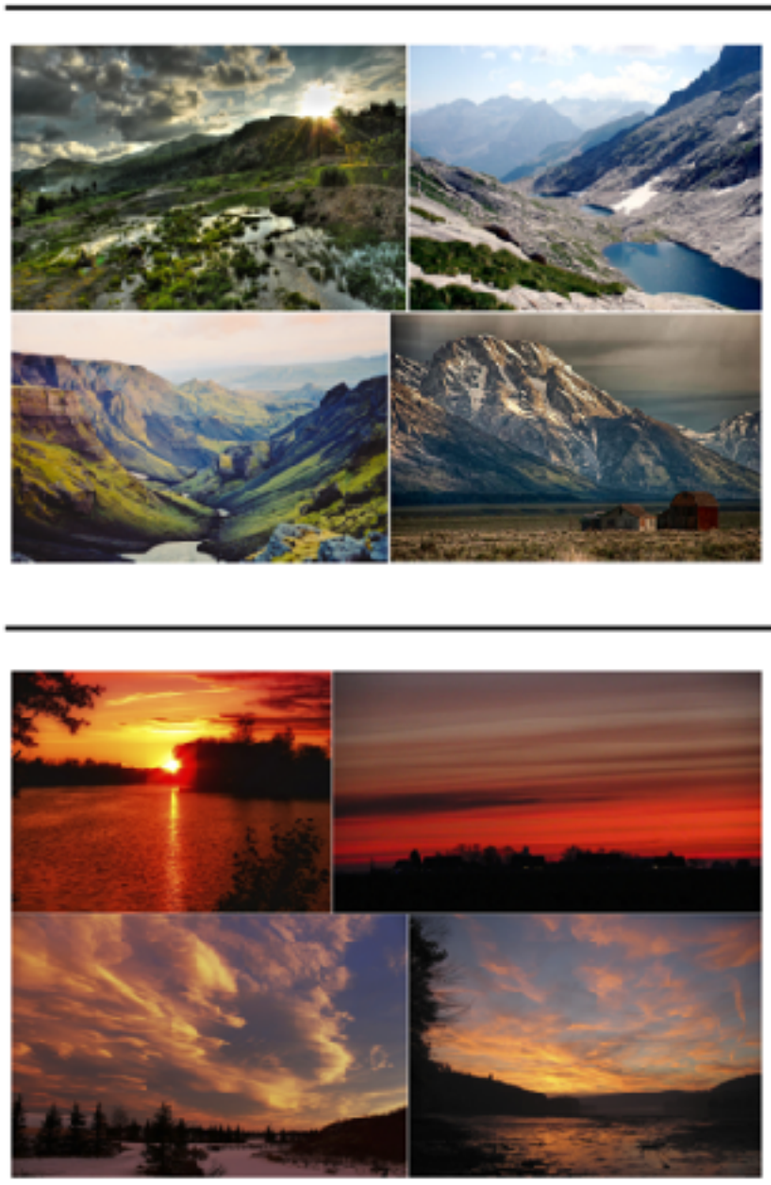
$\lambda_1 = 0.12$

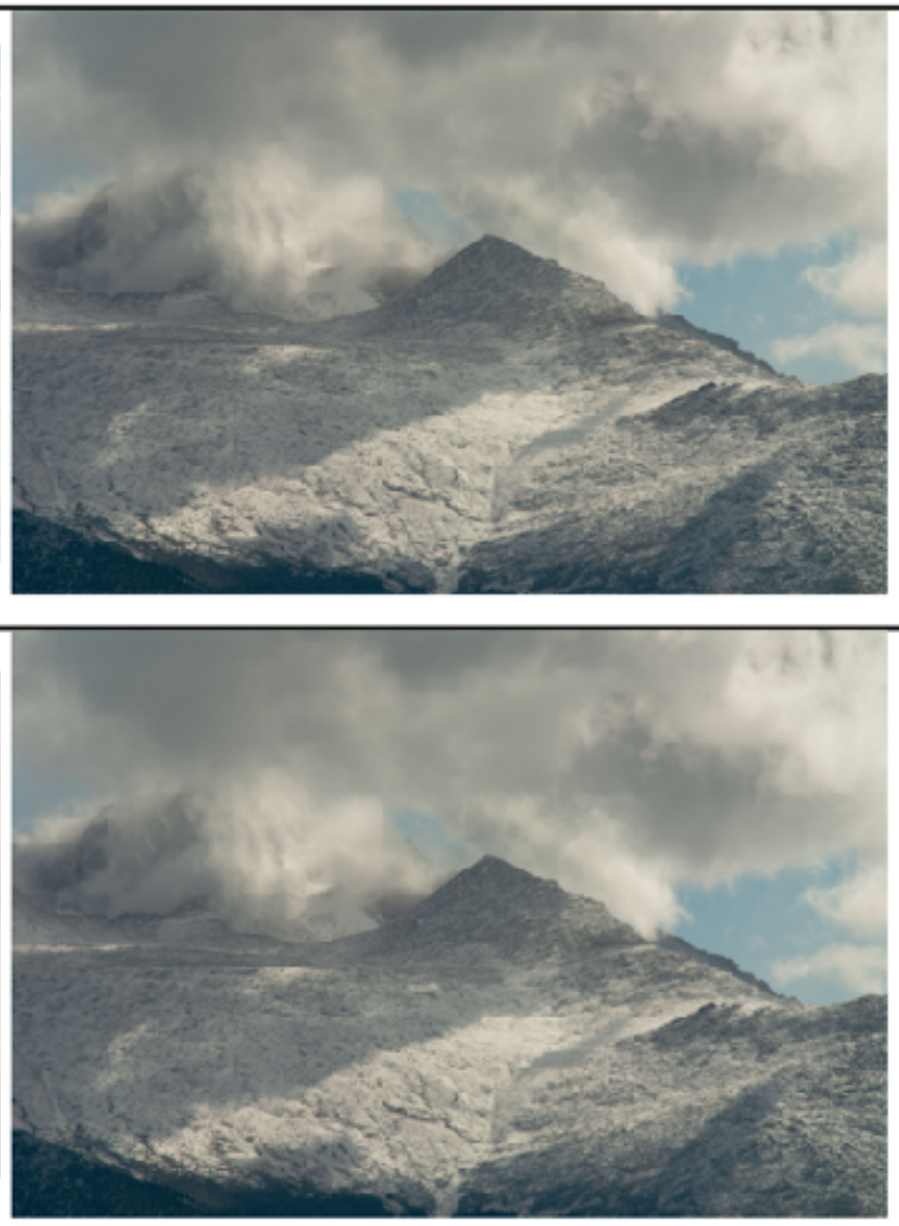
$\lambda_2 = 0.40$

$\lambda_3 = 0.43$

Application: Color Grading

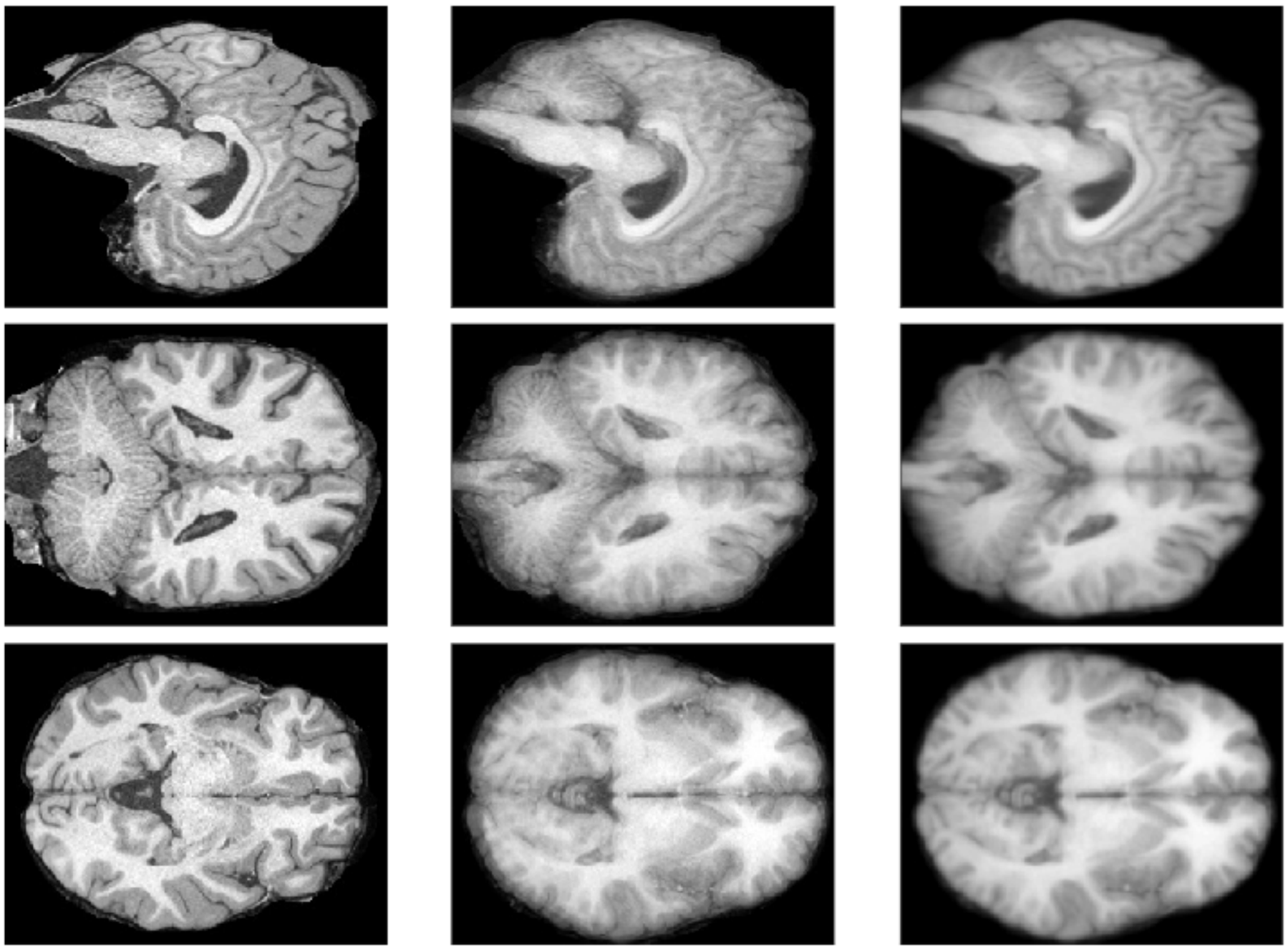
Application: Color Grading





Wasserstein Barycentric Coordinates: Histogram Regression using Optimal Transport, **SIGGRAPH'16** **[BPC'16]**

Application: Brain Mapping



Original | Euclidean projection | Wasserstein projection

Minimum Kantorovich Estimation

$$\theta^\star = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\, W_p^p(\boldsymbol{p_\theta}, \boldsymbol{\nu}_{\mathrm{data}}) \quad \textbf{[Bassetti'06]}$$

$$W_\gamma(p_{\boldsymbol{\theta}}, \boldsymbol{\nu}_{\mathrm{data}}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \langle \boldsymbol{\alpha}, p_{\boldsymbol{\theta}} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\nu}_{\mathrm{data}} \rangle - \gamma \langle e^{\boldsymbol{\alpha}/\gamma},\ K e^{\boldsymbol{\beta}/\gamma} \rangle$$

$$\nabla_\theta W_\gamma = \left(\frac{\partial p_\theta}{\partial \theta}\right)^T \boldsymbol{\alpha}^\star$$

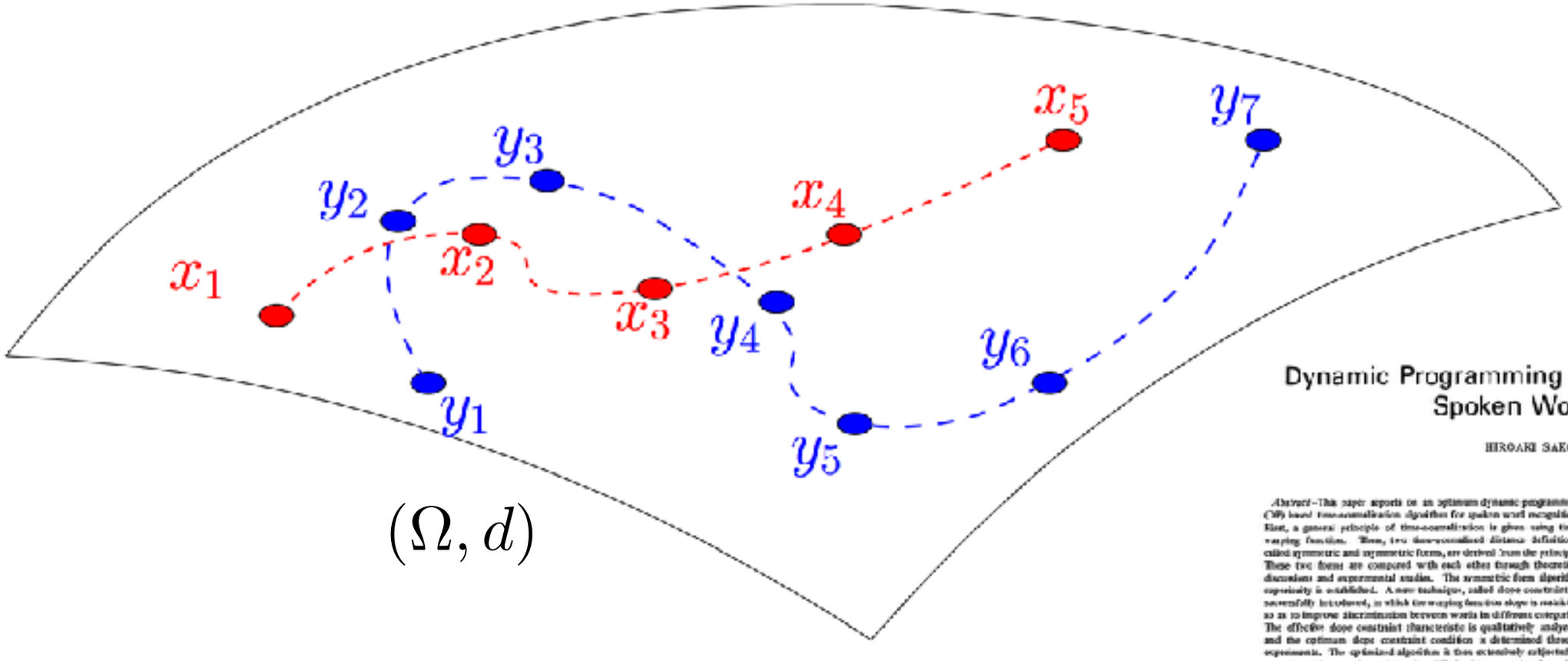
- Application to parameter estimation in discrete models **[MMC'16]**.
- Stochastic methods for semi-discrete OT **[GCPB'16]**

To conclude on Wasserstein

- *Entropy* regularization is a very effective way to get OT to work as a generic loss.
- Many recent extensions:
- **[Schmitzer'16]**: fast multiscale approaches
- **[ZFMAP'15] [CSPV'16]**: Unbalanced transport
- **[SPKS'16]** **[PCS'16]** extensions to *Gromov-W.*
- **[FCTR'15]** Domain adaptation in ML

Dynamic Time Warping

A distance to
compare **time series of observations**
supported on a metric space.



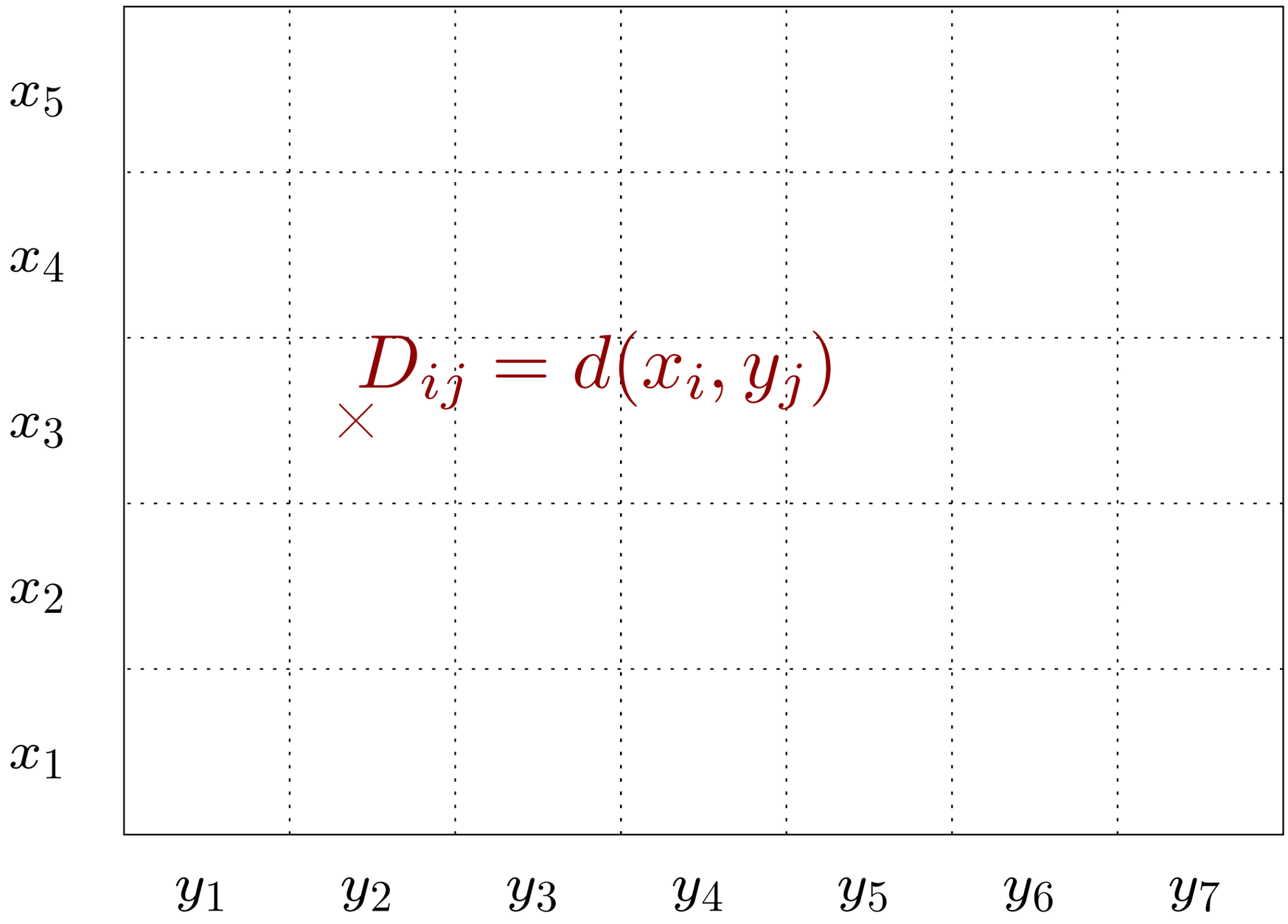
Dynamic Programming Algorithm Optimization for Spoken Word Recognition

HIROAKI SAKOE and SEIBI CHIBA

Alignment Grid

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

Fill in Metric Information



Fill in Metric Information

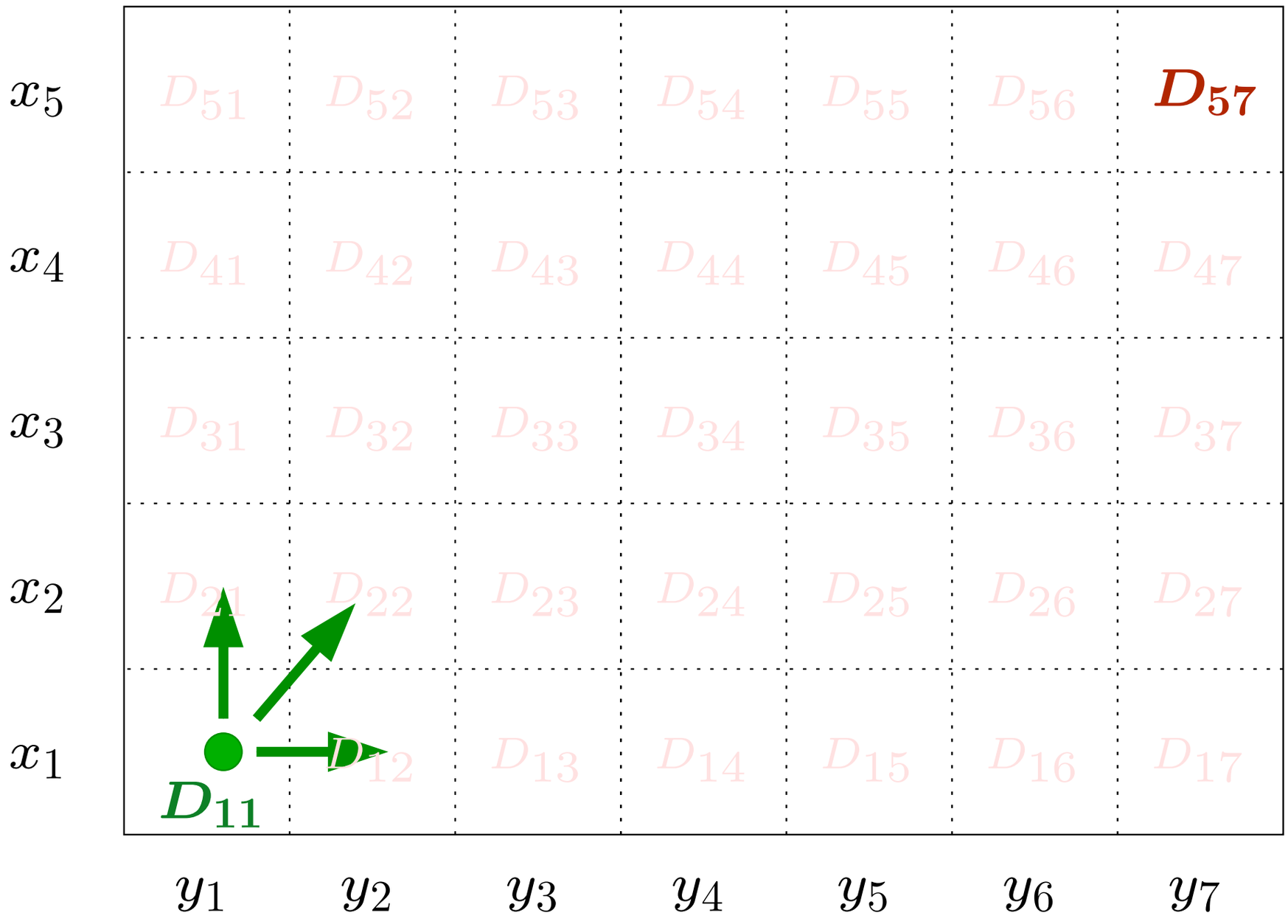
x_5	D_{51}	D_{52}	D_{53}	D_{54}	D_{55}	D_{56}	D_{57}
x_4	D_{41}	D_{42}	D_{43}	D_{44}	D_{45}	D_{46}	D_{47}
x_3	D_{31}	D_{32}	D_{33}	D_{34}	D_{35}	D_{36}	D_{37}
x_2	D_{21}	D_{22}	D_{23}	D_{24}	D_{25}	D_{26}	D_{27}
x_1	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}
	y_1	y_2	y_3	y_4	y_5	y_6	y_7

Alignment Paths

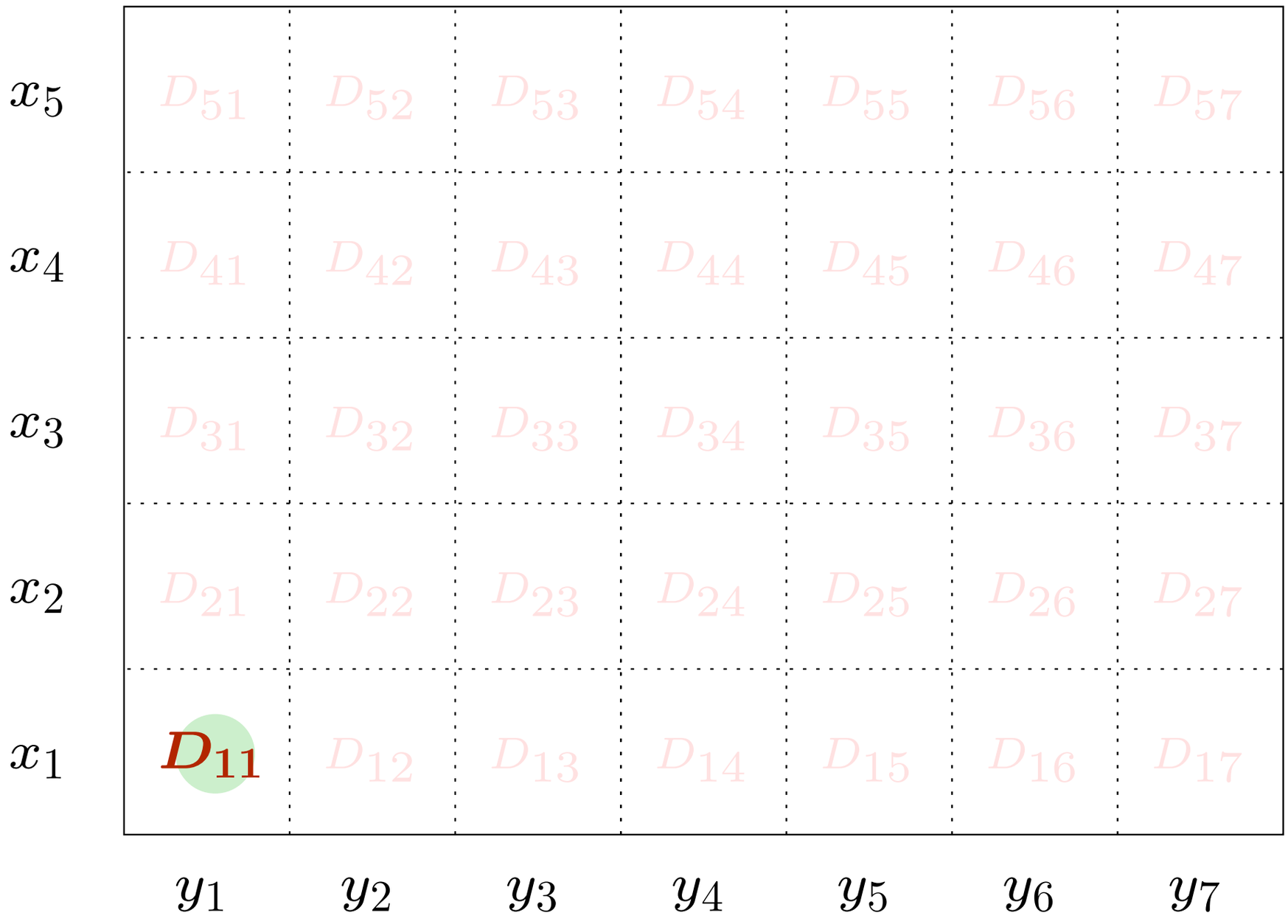
start from (1,1) and ends at (5,7)

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

Three Possible Directions

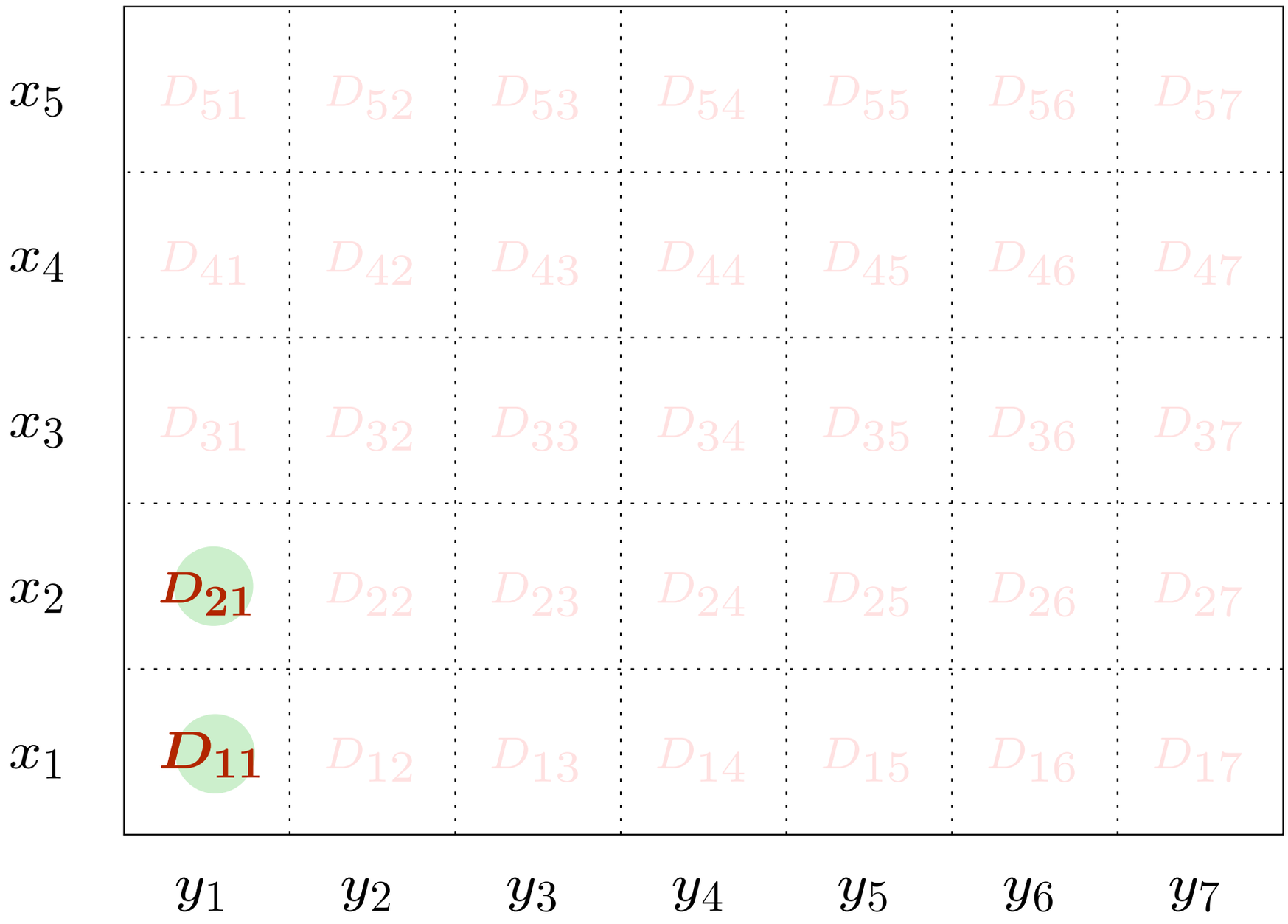


Example Path



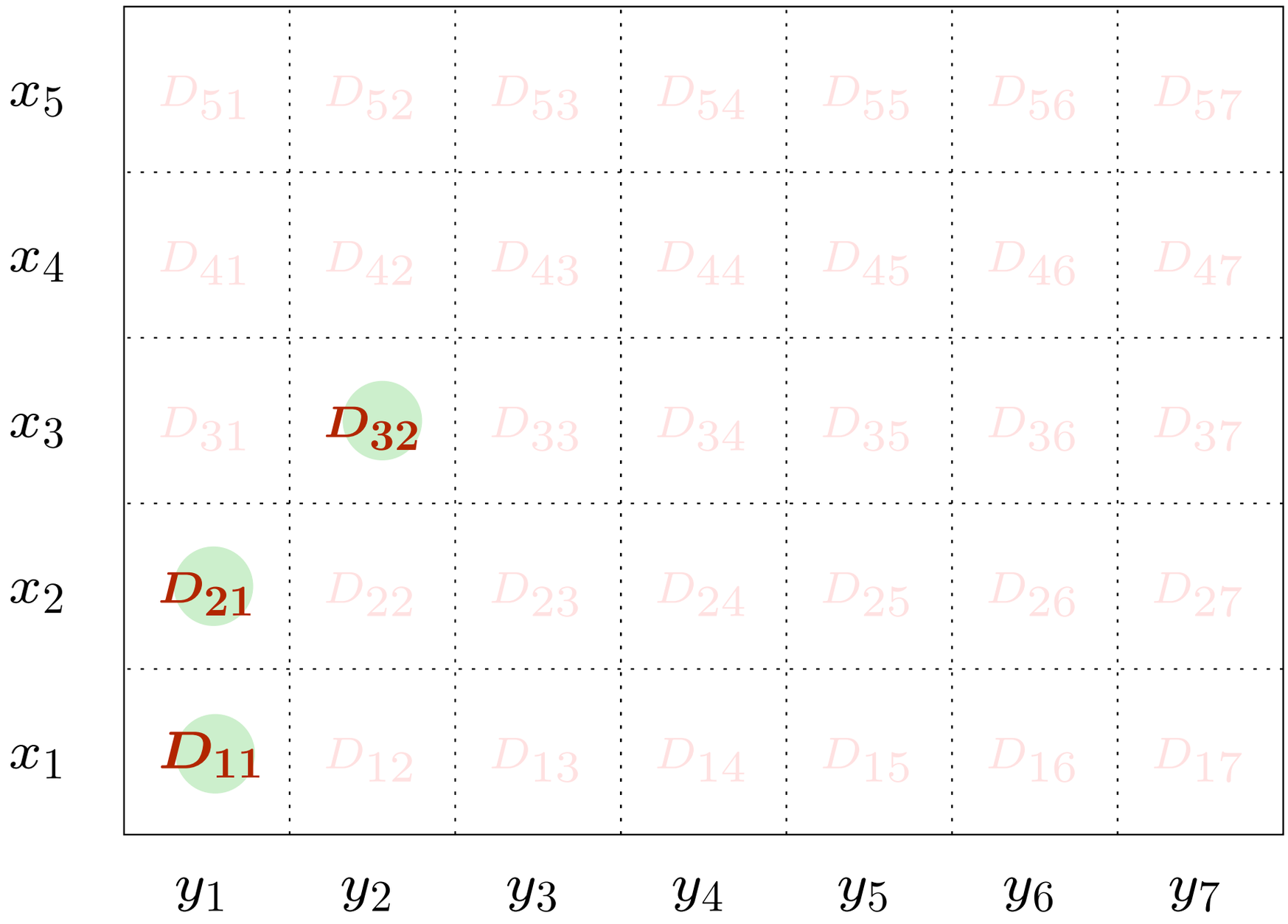
Path Cost = Sum of Visited Distances

$$C = D_{11} + \boldsymbol{D_{21}}.$$



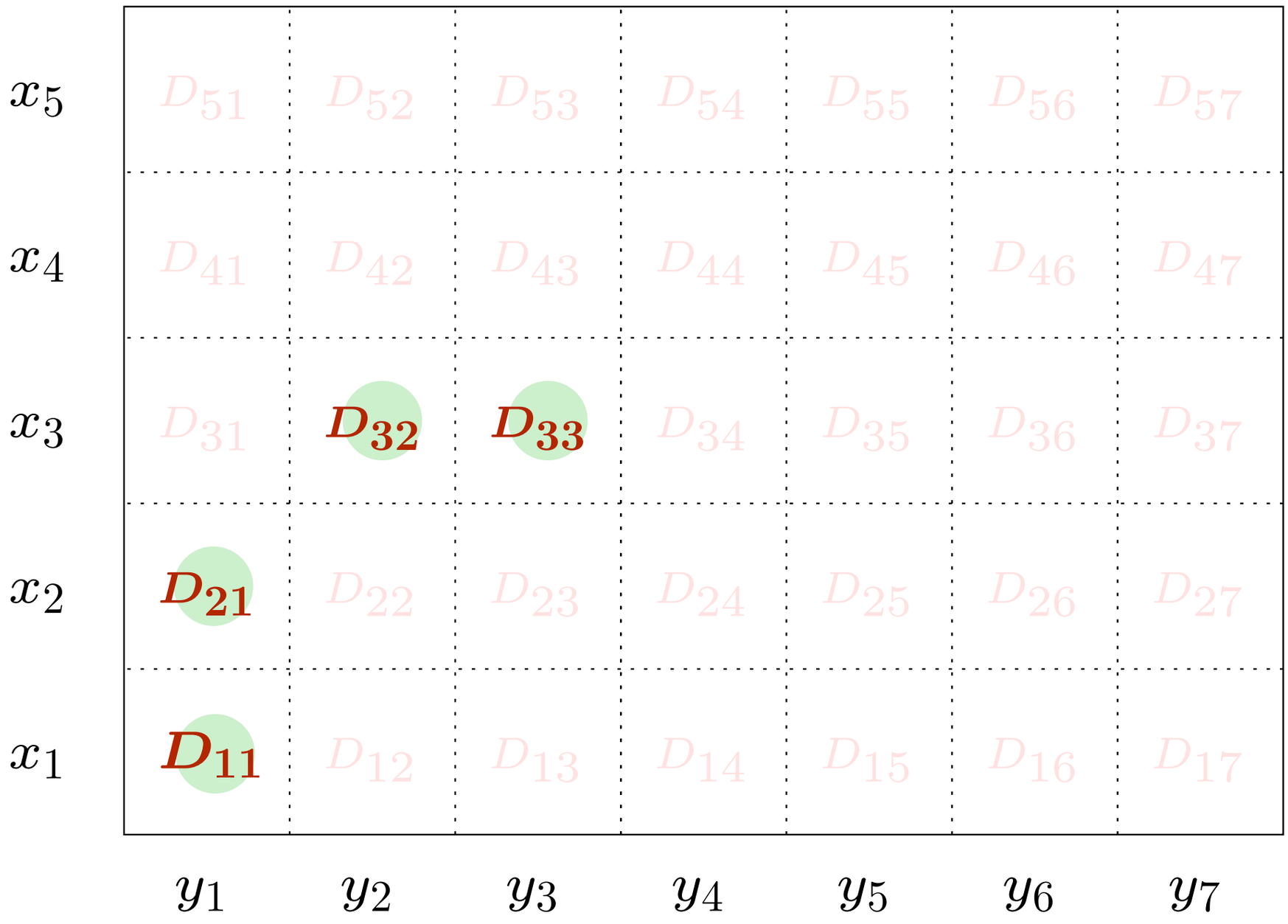
Path Cost = Sum of Visited Distances

$$C = D_{11} + D_{21} + \boldsymbol{D_{32}}.$$



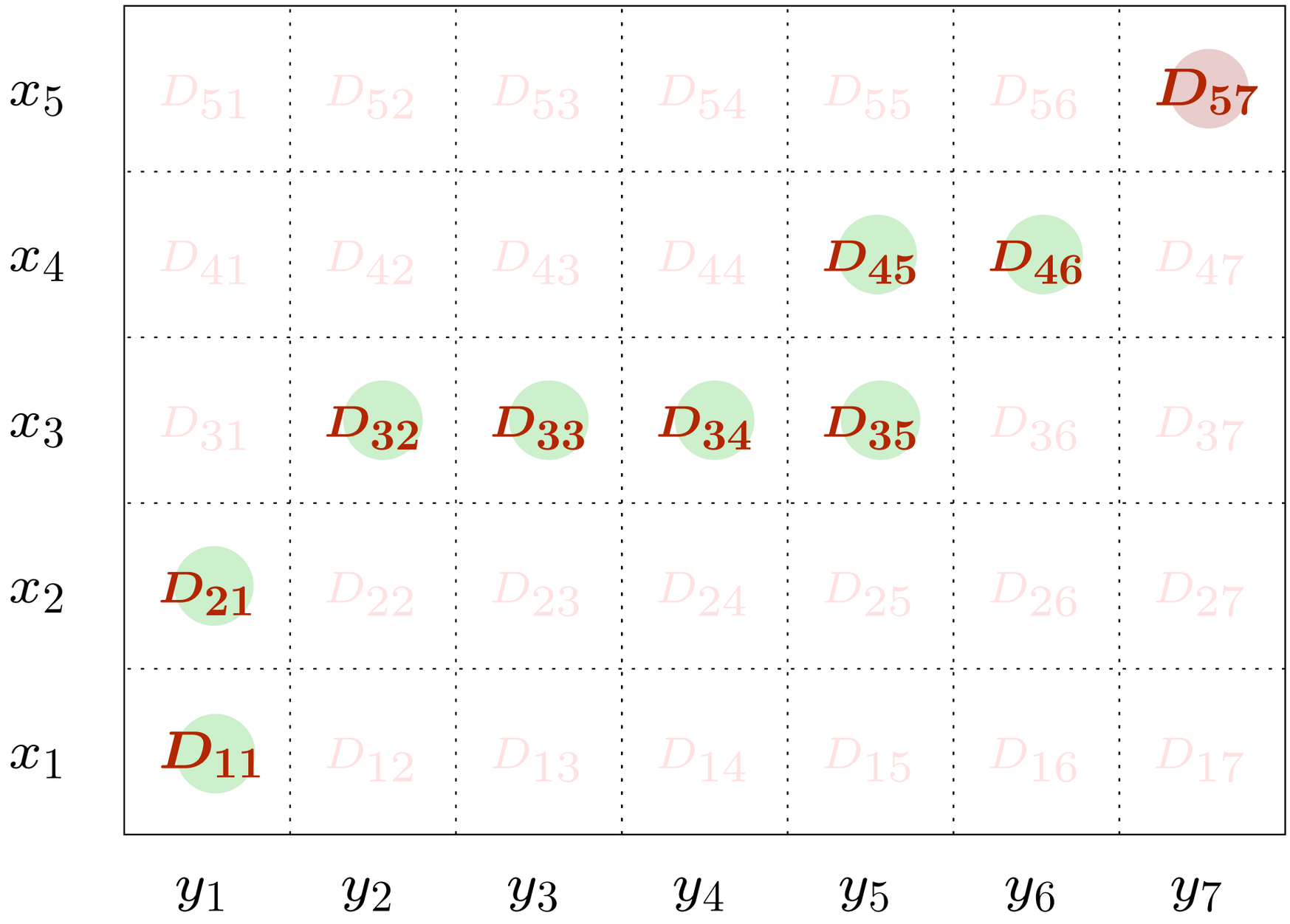
Path Cost = Sum of Visited Distances

$$C = D_{11} + D_{21} + D_{32} + \boldsymbol{D_{33}}.$$



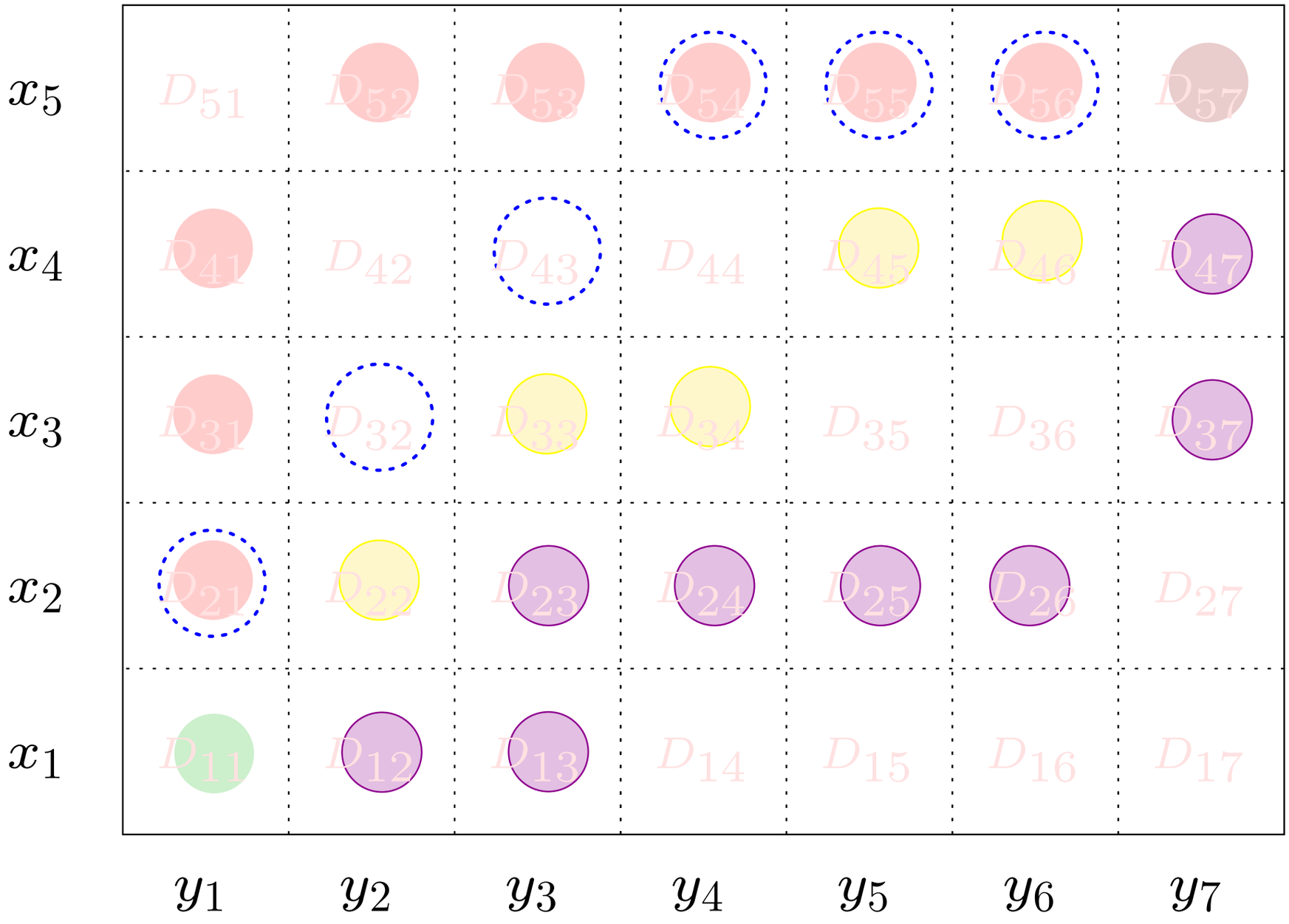
Path Cost = Sum of Visited Distances

$$C = D_{11} + D_{21} + D_{32} + D_{33} + D_{34} + D_{35} + D_{45} + D_{46} + \boldsymbol{D_{57}}.$$



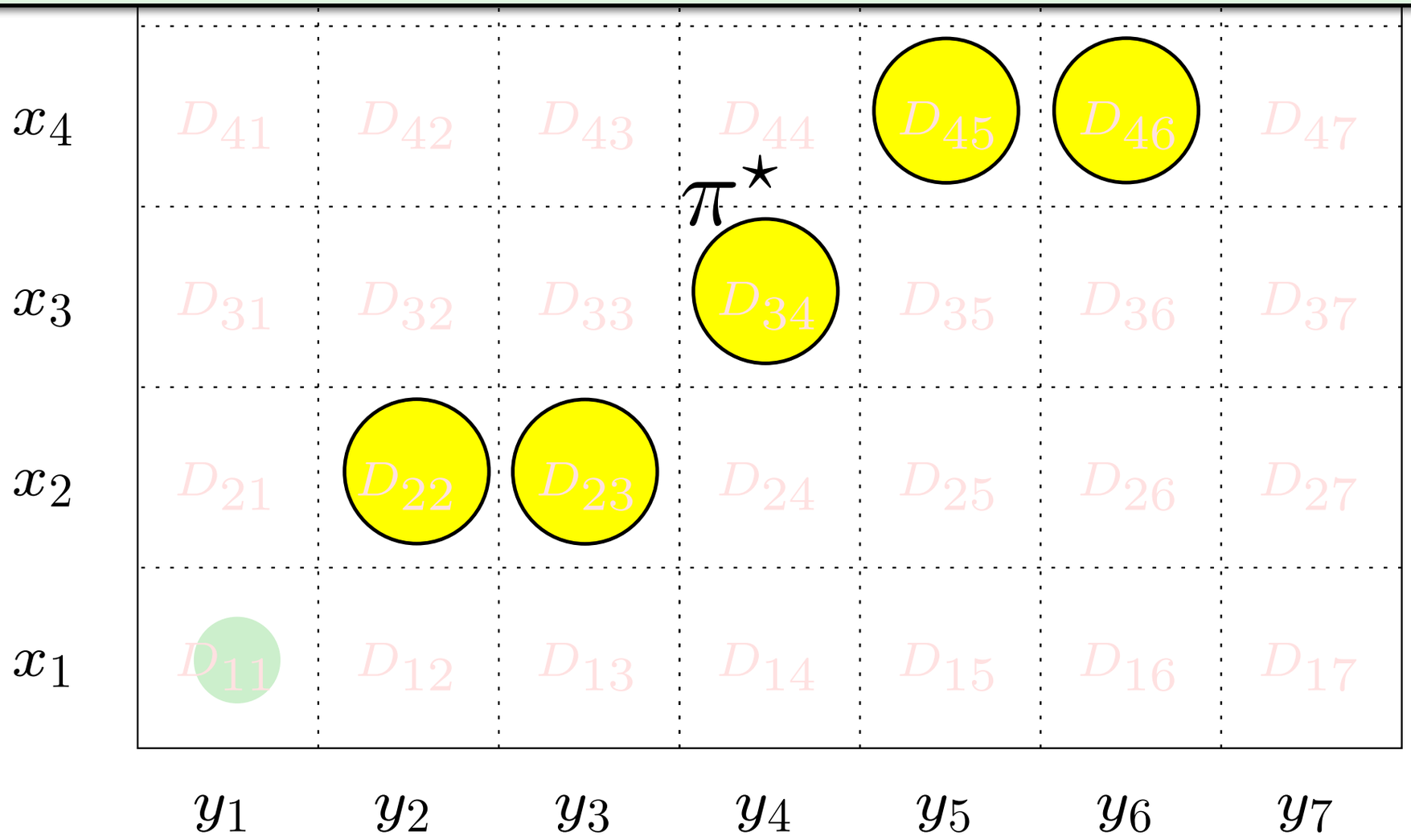
#All Paths = *Delannoy(5,7)*

Delannoy(5,7) = 2,241 ; Delannoy(20,20)= 4.53e13



Dynamic Time Warping (Distance)

$$d_{\mathrm{DTW}}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\pi \in \mathcal{A}(\boldsymbol{X}, \boldsymbol{Y})} \sum_{i=1} d\left(\boldsymbol{x}_{\pi_1(i)}, \boldsymbol{y}_{\pi_2(i)}\right)$$

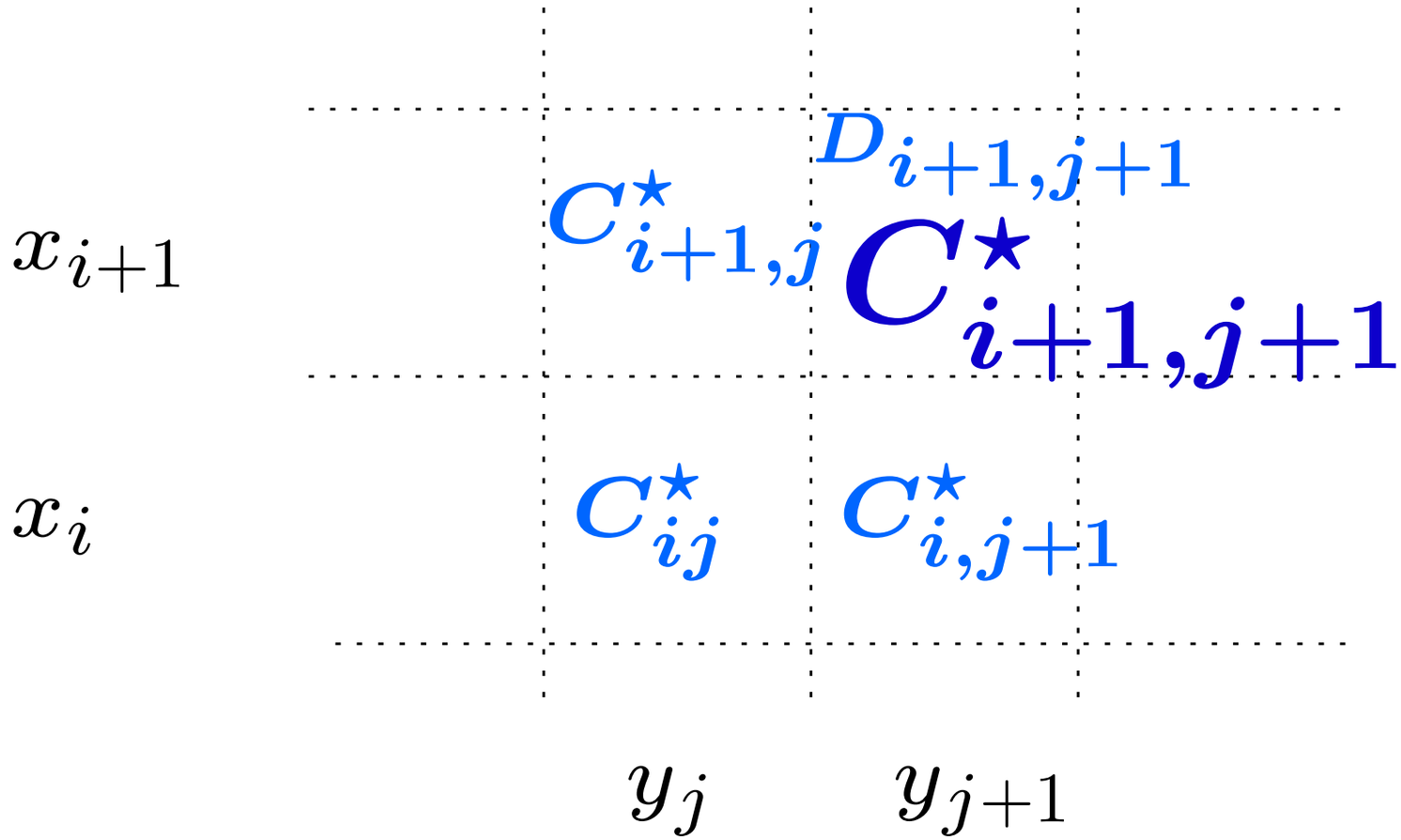


DP Computation

$$C^{\star}_{ij} = \min_{\pi \in \mathcal{A}(i,j)} C_{\mathbf{x}_1^i, \mathbf{y}_1^j}(\pi).$$

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

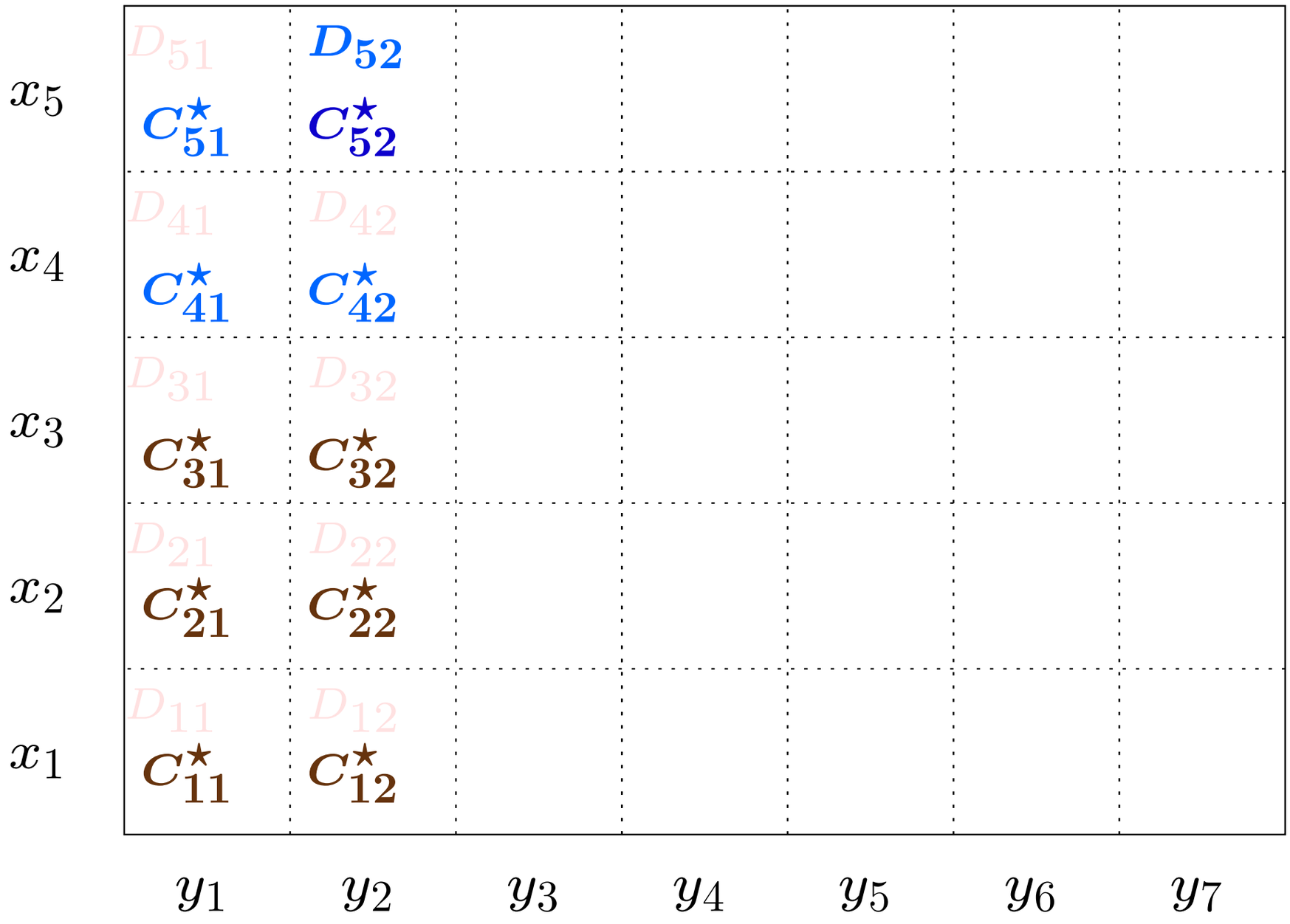
Bellman Recursion



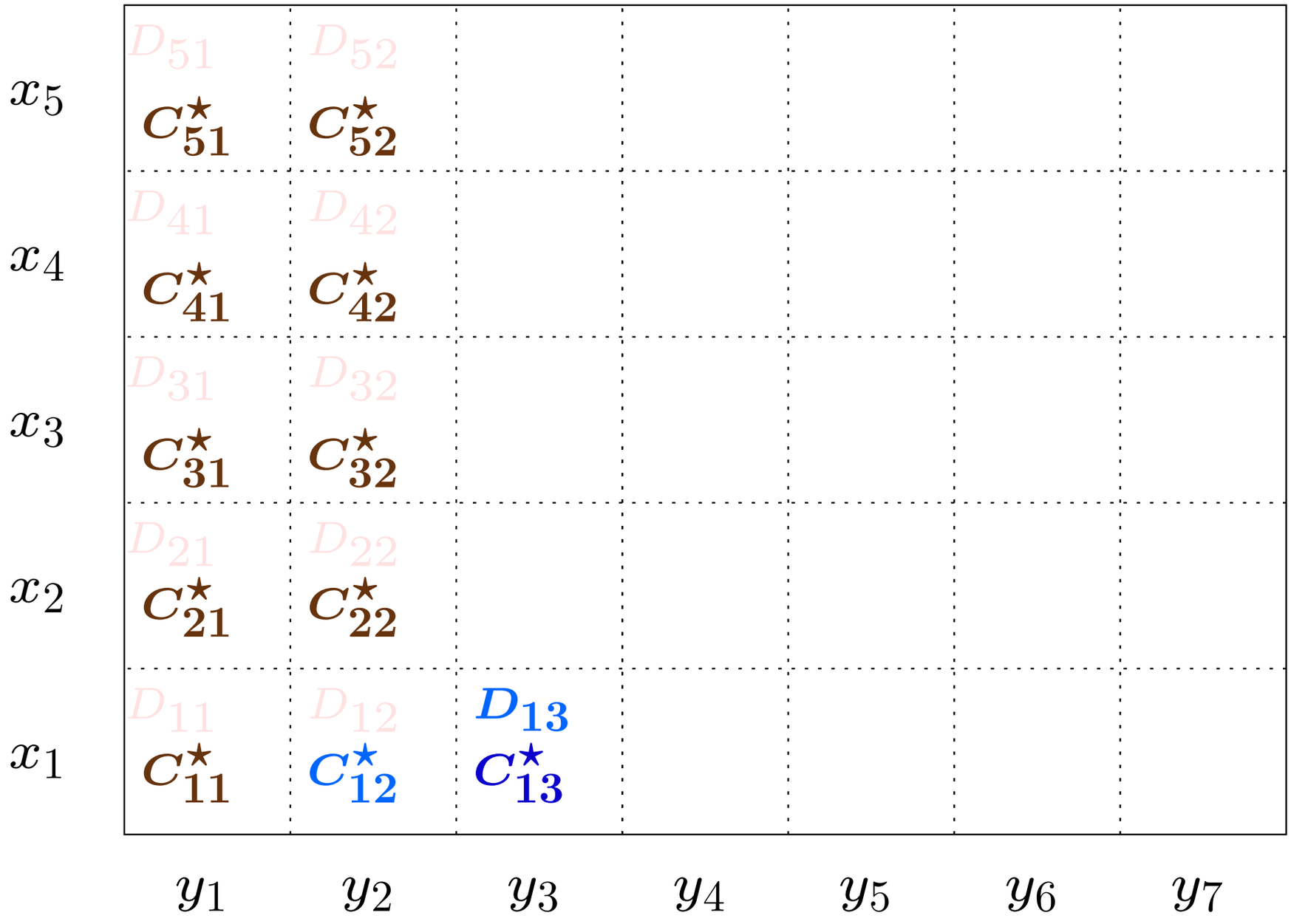
Bellman recursion: for all $i \leq n-1, j \leq m-1$,

$$C^\star_{i+1,j+1} = \min(C^\star_{i+1,j}, C^\star_{ij}, C^\star_{i,j+1}) + D_{i+1,j+1}$$

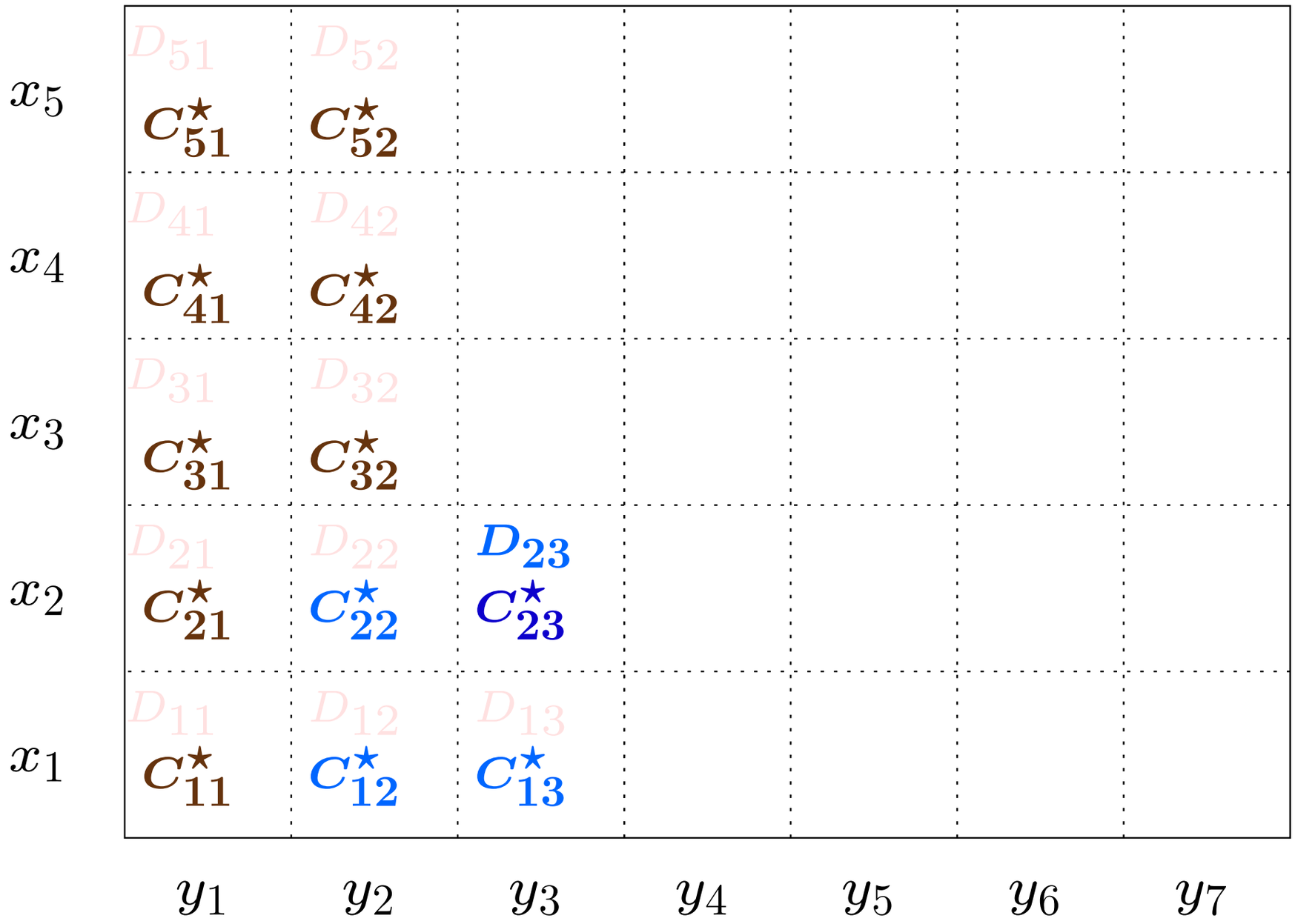
Bellman Recursion In Practice



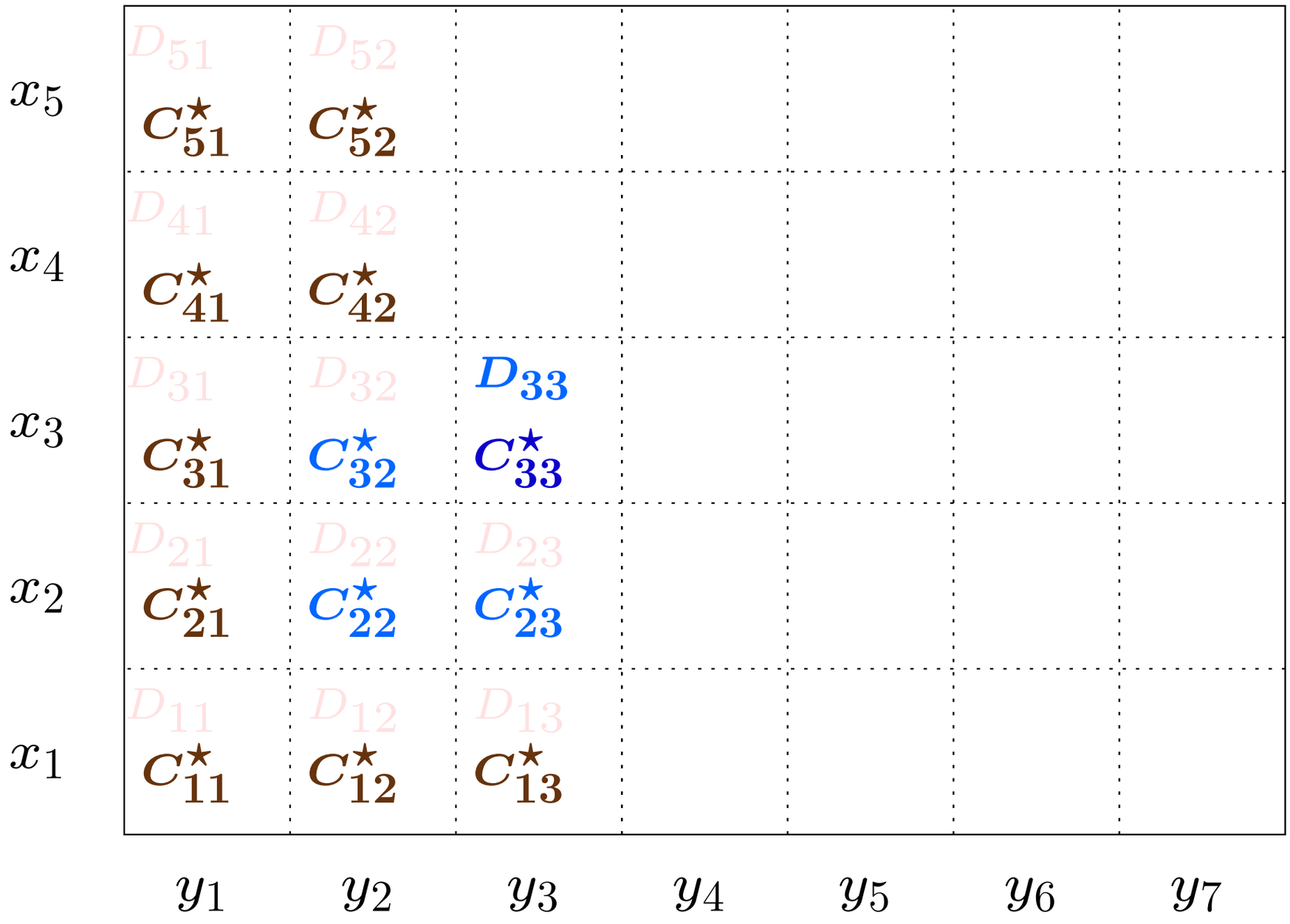
Bellman Recursion In Practice



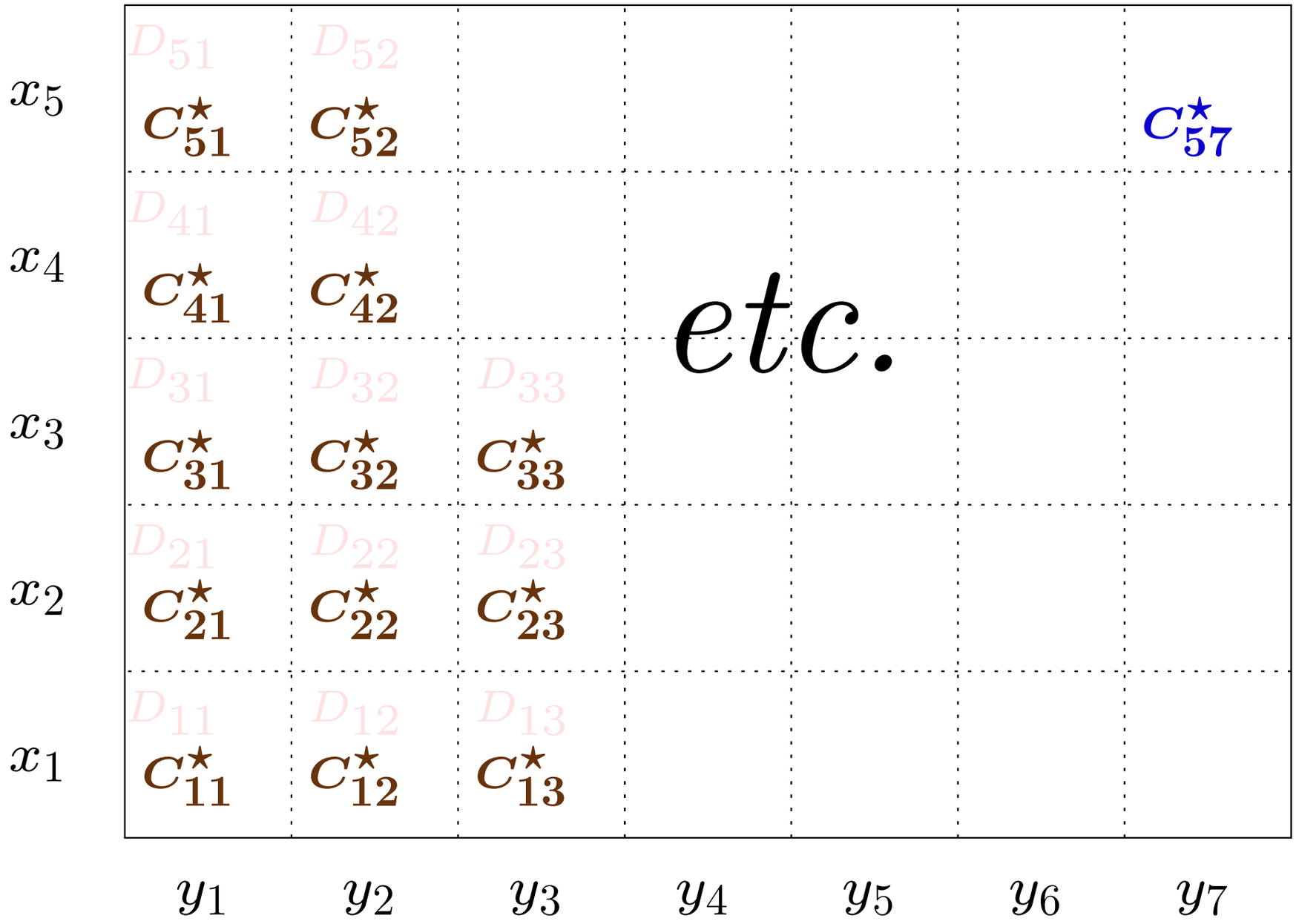
Bellman Recursion In Practice



Bellman Recursion In Practice

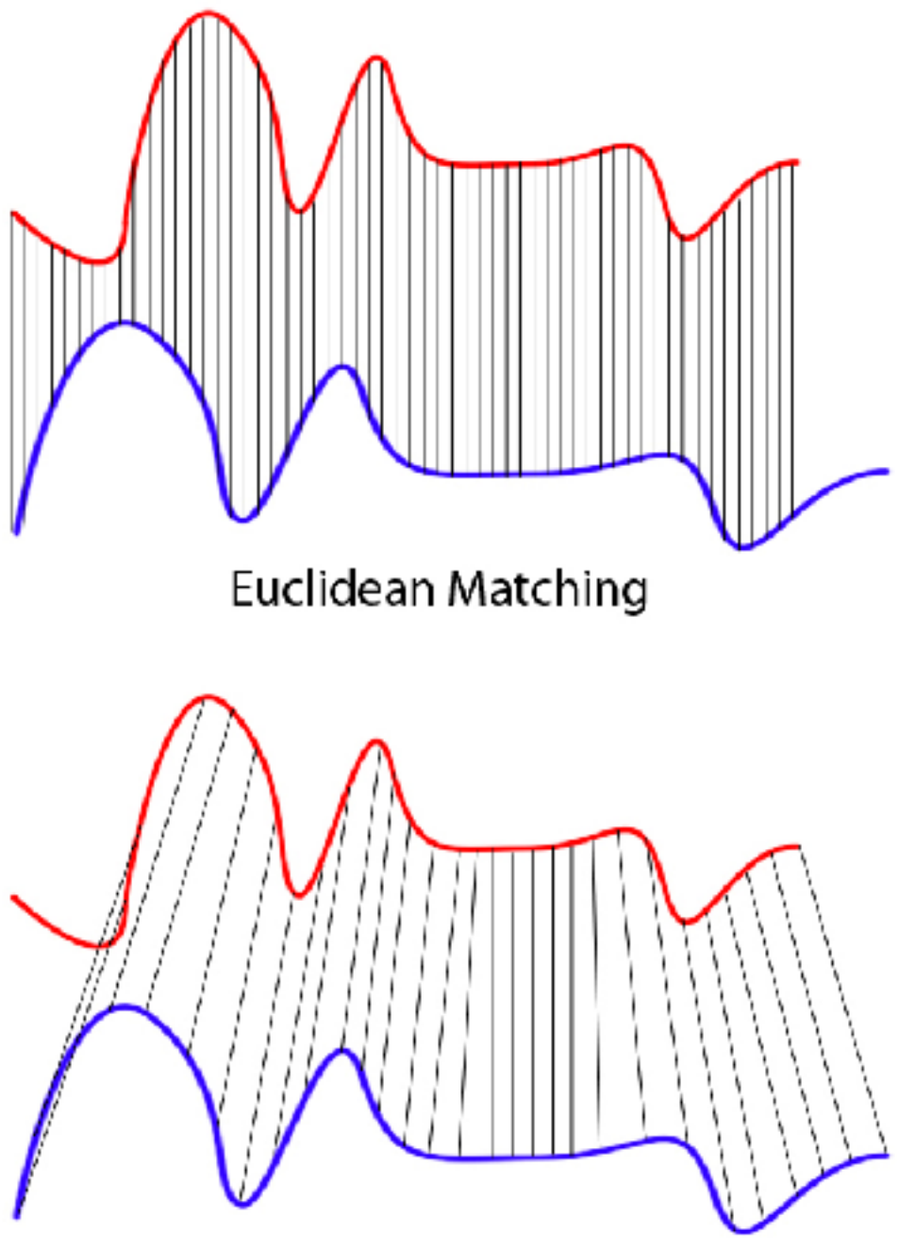


Bellman Recursion In Practice



DTW Strengths

Well documented!



DTW as LP

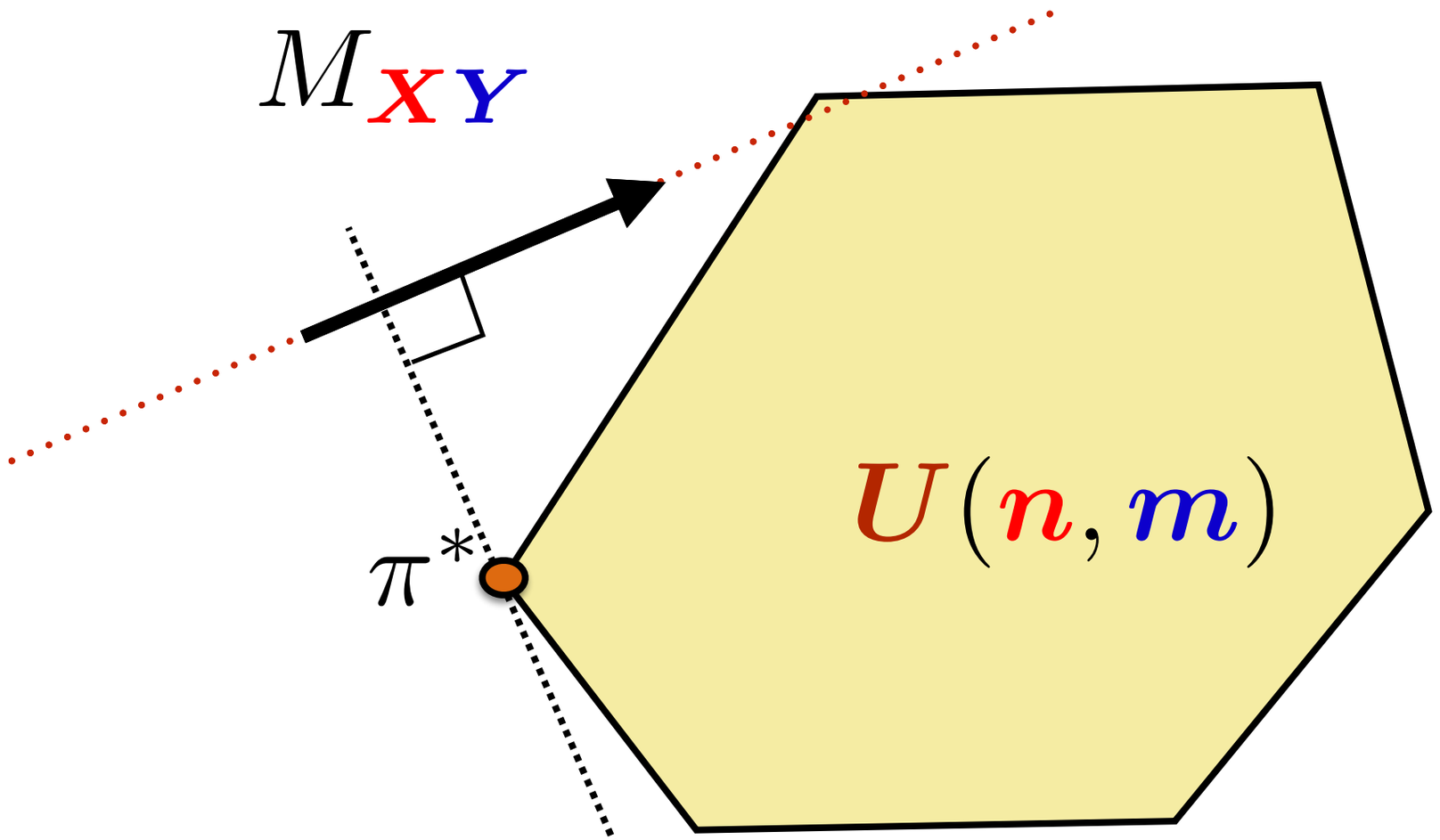
Let $\boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$ and $\boldsymbol{Y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_m})$.

$$U(\boldsymbol{n}, \boldsymbol{m}) \stackrel{\text{def}}{=} \text{co}\{\boldsymbol{\pi}, (\boldsymbol{n}, \boldsymbol{m}) \text{ alig. mat.}\} \subset [0,1]^{n \times m}$$

$$M_{\boldsymbol{XY}} \stackrel{\text{def}}{=} [D(\boldsymbol{x}_i, \boldsymbol{y}_j)^p]_{ij}$$

$$\boldsymbol{\pi} = \begin{bmatrix} \mathbf{1} & 0 & \cdots & \cdots & \cdots \\ \mathbf{1} & 0 & \cdots & \cdots & \cdots \\ 0 & \mathbf{1} & \mathbf{1} & 0 & \cdots \\ \vdots & \vdots & \mathbf{1} & \ddots & \vdots \\ \cdots & \cdots & \cdots & \mathbf{1} & \mathbf{1} \end{bmatrix} \qquad M_{\boldsymbol{XY}} \stackrel{\text{def}}{=} \begin{array}{c} \\ \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_i} \\ \vdots \\ \boldsymbol{x_n} \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y_1} & \cdots & \boldsymbol{y_j} & \cdots & \boldsymbol{y_m} \end{array} \\ \begin{bmatrix} & & & & \\ & & D(\boldsymbol{x_i}, \boldsymbol{y_j})^p & & \\ & & & & \end{bmatrix} \end{array}$$

DTW Problem

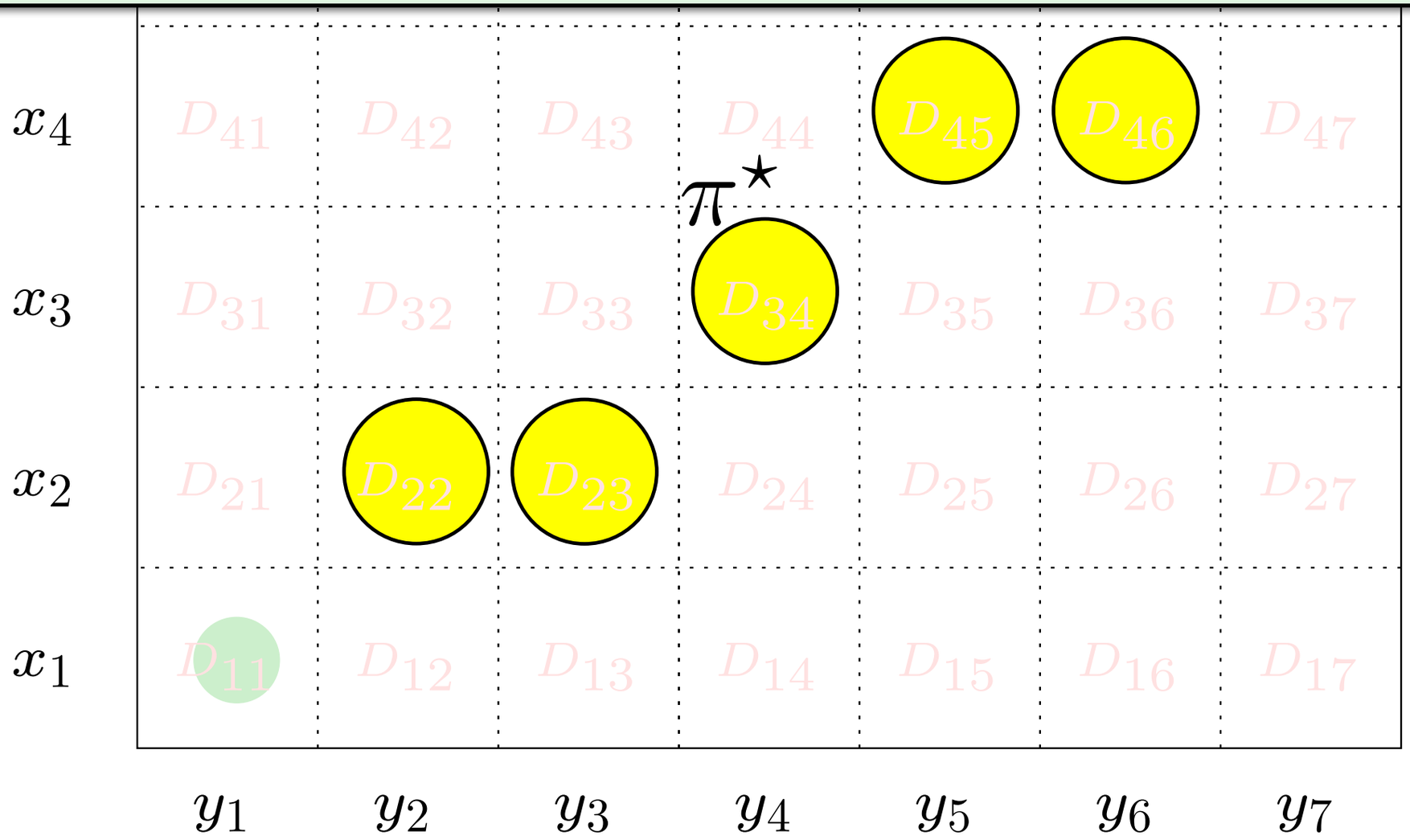


$$d_{\mathrm{DTW}}(X, Y) = \langle \pi^\star, M_{XY} \rangle$$

$$= \min_{\pi \in U(n,m)} \langle \pi, M_{XY} \rangle$$

Soft Dynamic Time Warping

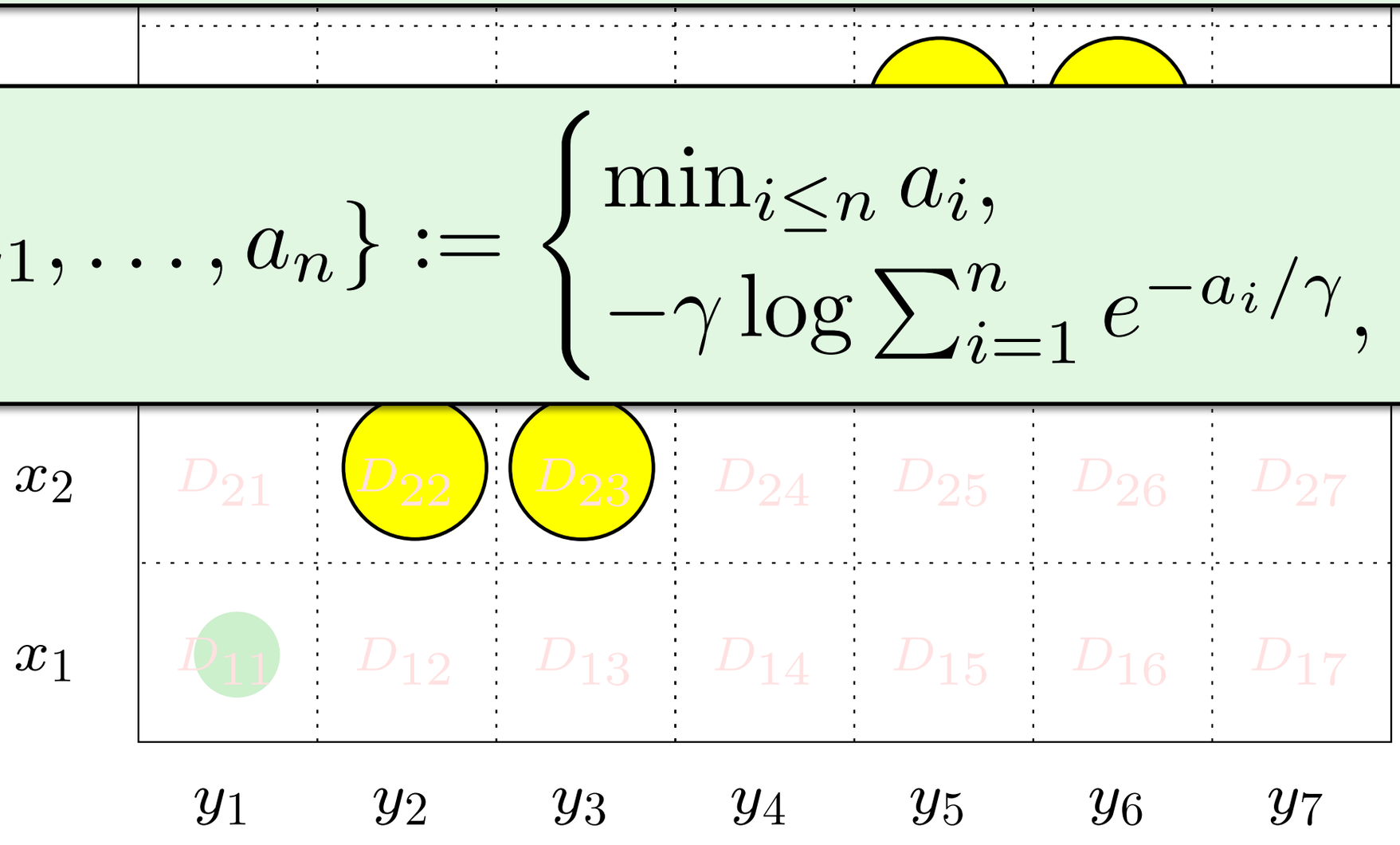
$$\mathbf{dtw}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{XY}} \rangle$$



Soft Dynamic Time Warping

$$\mathbf{dtw}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\min^{\gamma}\{a_1, \ldots, a_n\} := \begin{cases} \min_{i \le n} a_i, & \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0. \end{cases}$$



Soft Dynamic Time Warping

$$\mathbf{dtw}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\min^{\gamma}\{a_1, \ldots, a_n\} := \begin{cases} \min_{i \le n} a_i, & \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0. \end{cases}$$

$$\mathbf{dtw}^{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min^{\gamma}_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

Soft Dynamic Time Warping

$$\mathbf{dtw}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

$$\min^{\gamma}\{a_1, \ldots, a_n\} := \begin{cases} \min_{i \le n} a_i, & \gamma = 0, \\ -\gamma \log \sum_{i=1}^{n} e^{-a_i/\gamma}, & \gamma > 0. \end{cases}$$

$$\mathbf{dtw}^{\gamma}(\boldsymbol{X}, \boldsymbol{Y}) = \min^{\gamma}_{\boldsymbol{\pi} \in \boldsymbol{U}(\boldsymbol{n}, \boldsymbol{m})} \langle \boldsymbol{\pi}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle$$

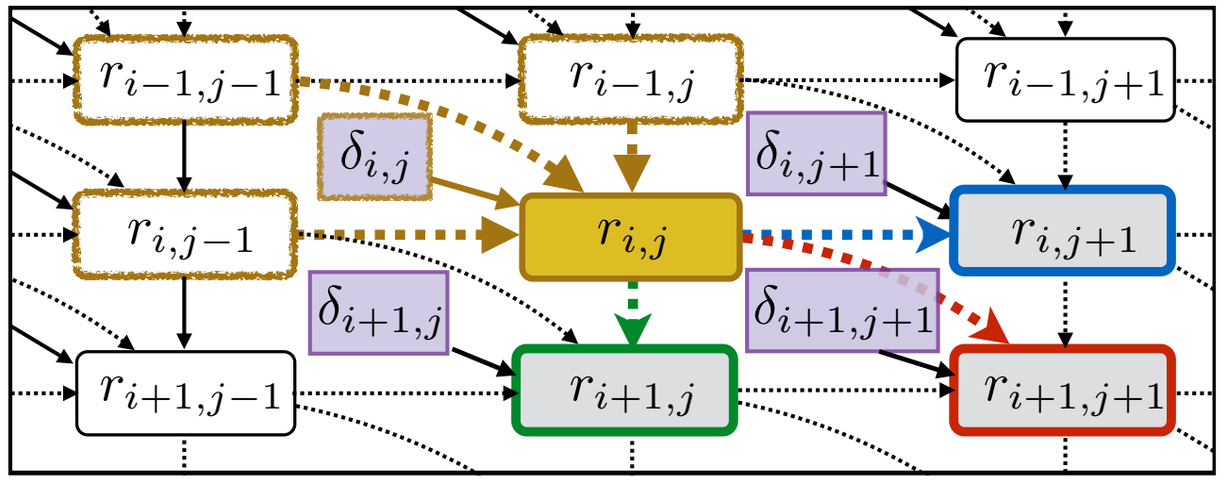
differentiable

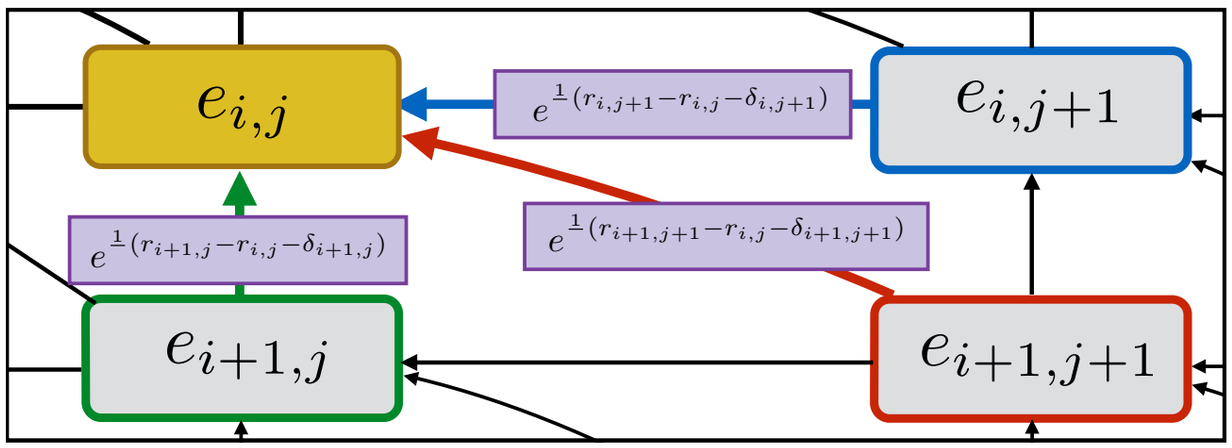
Differentiation of sDTW

Algorithm 2 Computes $\mathbf{dtw}_\gamma(\mathbf{x}, \mathbf{y})$ and $\nabla_{\mathbf{x}} \mathbf{dtw}_\gamma(\mathbf{x}, \mathbf{y})$

Inputs: $\mathbf{x}, \mathbf{y}$, smoothing $\gamma \geq 0$, distance function δ.
$\Delta = [\delta(x_i, y_j)]_{i,j}$.
$r_{0,0} = 0; r_{i,0} = r_{0,j} = \infty; i \in [\![n]\!], j \in [\![m]\!]$.
for $j = 1, \ldots, m$ **do** ▷ Forward recursion
  for $i = 1, \ldots, n$ **do**
    $r_{i,j} = \delta_{i,j} + \min^\gamma \{r_{i-1,j-1}, r_{i-1,j}, r_{i,j-1}\}$
  end for
end for
$\delta_{i,m+1} = \delta_{n+1,j} = 0, i \in [\![n]\!], j \in [\![m]\!]$
$e_{i,m+1} = e_{n+1,j} = 0, i \in [\![n]\!], j \in [\![m]\!]$
$r_{i,m+1} = r_{n+1,j} = -\infty, i \in [\![n]\!], j \in [\![m]\!]$
$\delta_{n+1,m+1} = 0, e_{n+1,m+1} = 1, r_{n+1,m+1} = r_{n,m}$
for $j = m, \ldots, 1$ **do** ▷ Backward recursion
  for $i = n, \ldots, 1$ **do**
    $a = \exp \frac{1}{\gamma}(r_{i+1,j} - r_{i,j} - \delta_{i+1,j})$
    $b = \exp \frac{1}{\gamma}(r_{i,j+1} - r_{i,j} - \delta_{i,j+1})$
    $c = \exp \frac{1}{\gamma}(r_{i+1,j+1} - r_{i,j} - \delta_{i+1,j+1})$
    $e_{i,j} = e_{i+1,j} \cdot a + e_{i,j+1} \cdot b + e_{i+1,j+1} \cdot c$
  end for
end for
Output: $\mathbf{dtw}_\gamma(\mathbf{x}, \mathbf{y}) = r_{n,m}$
$\nabla_{\mathbf{x}} \mathbf{dtw}_\gamma(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial \Delta(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}\right)^T E$

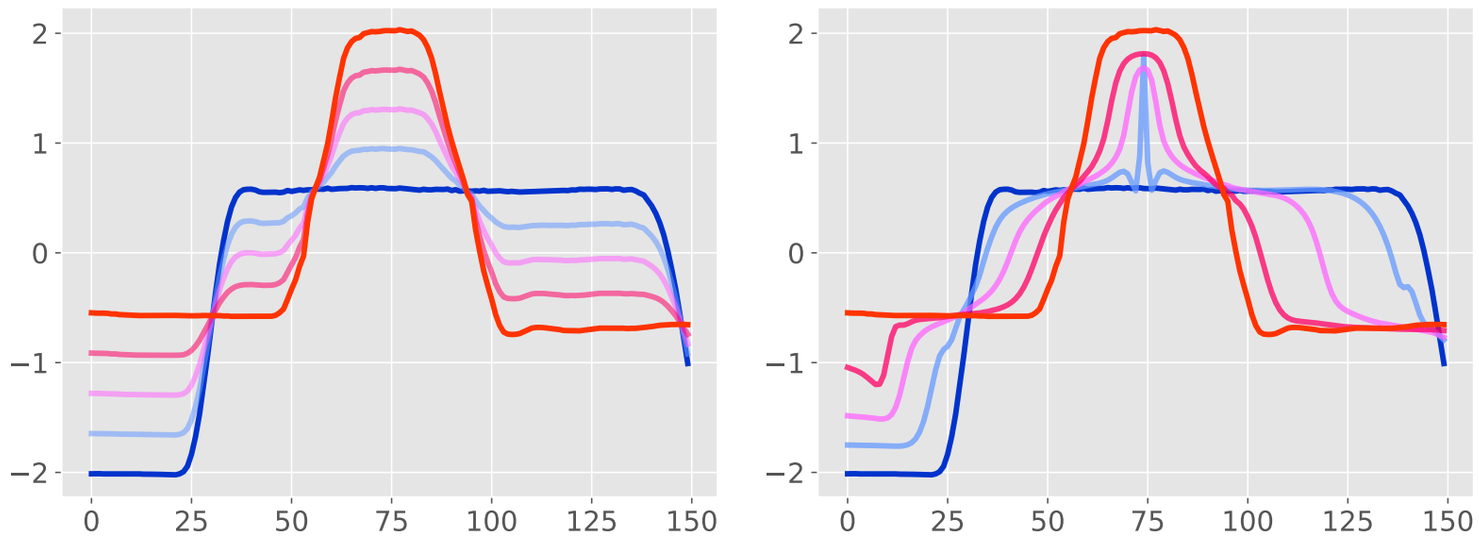
Automatic Differentiation





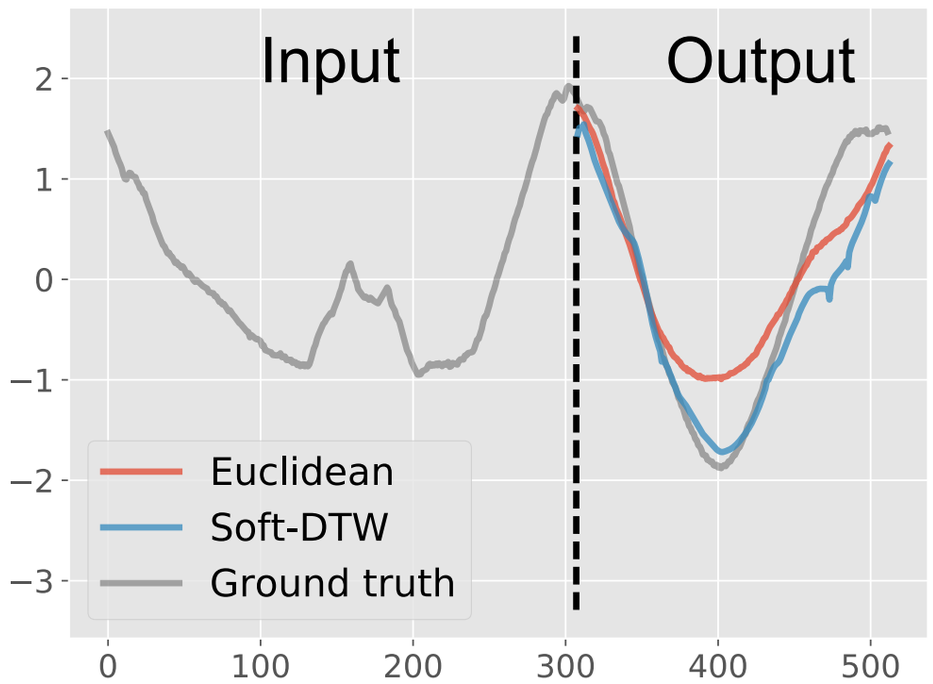
Applications: sDTW as Loss

Barycenters
Clustering



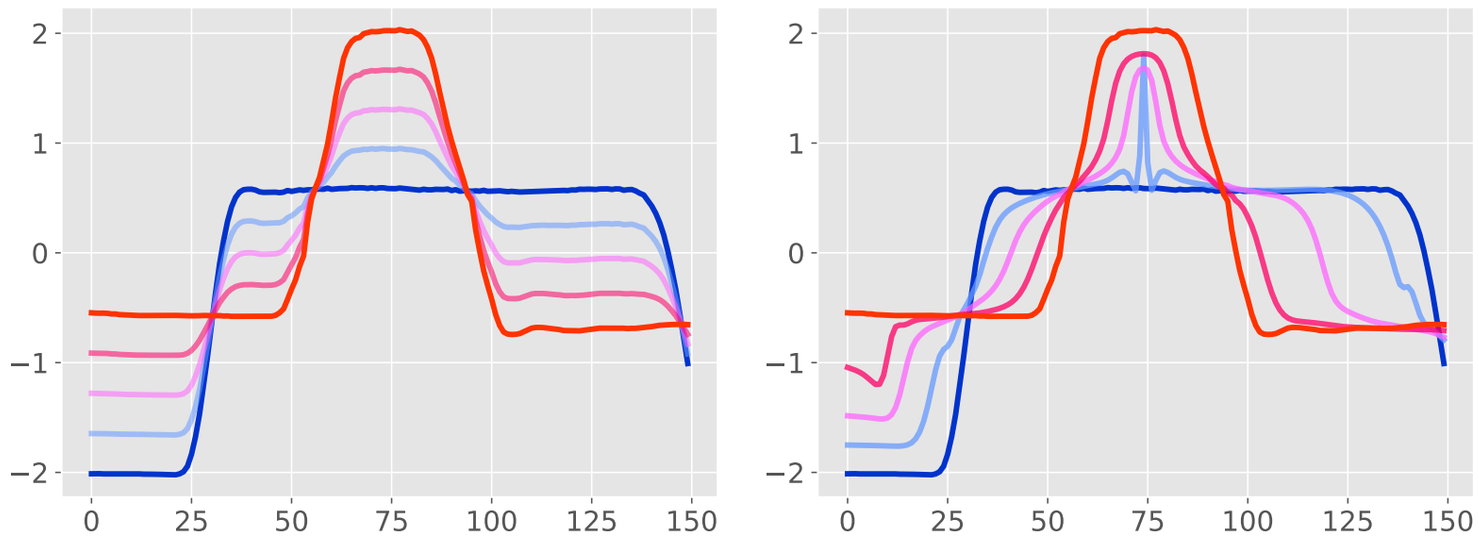
(a) Euclidean loss (b) Soft-DTW loss ($\gamma = 1$)

sDTW as a prediction loss



Applications: sDTW as Loss

Barycenters Clustering



(a) Euclidean loss (b) Soft-DTW loss ($\gamma = 1$)

sDTW as a prediction loss

https://arxiv.org/abs/1703.01541