

Computational Optimal Transport

Marco Cuturi



Google AI
Brain Team



Found. & Trends in ML
survey with Gabriel Peyré

<https://optimaltransport.github.io/>

Foundations and Trends® in
Machine Learning
11:5-6

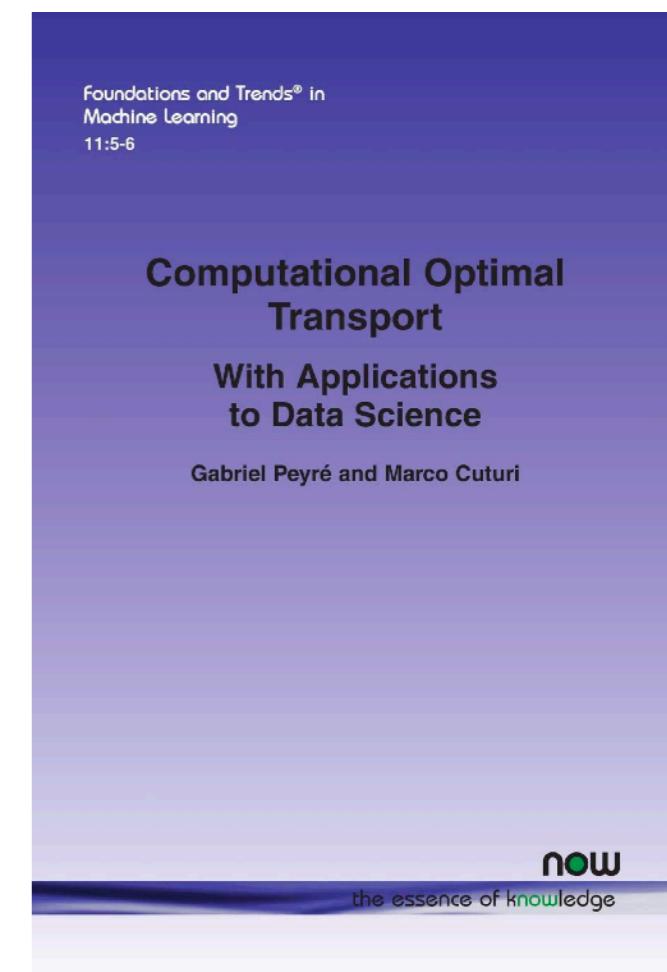
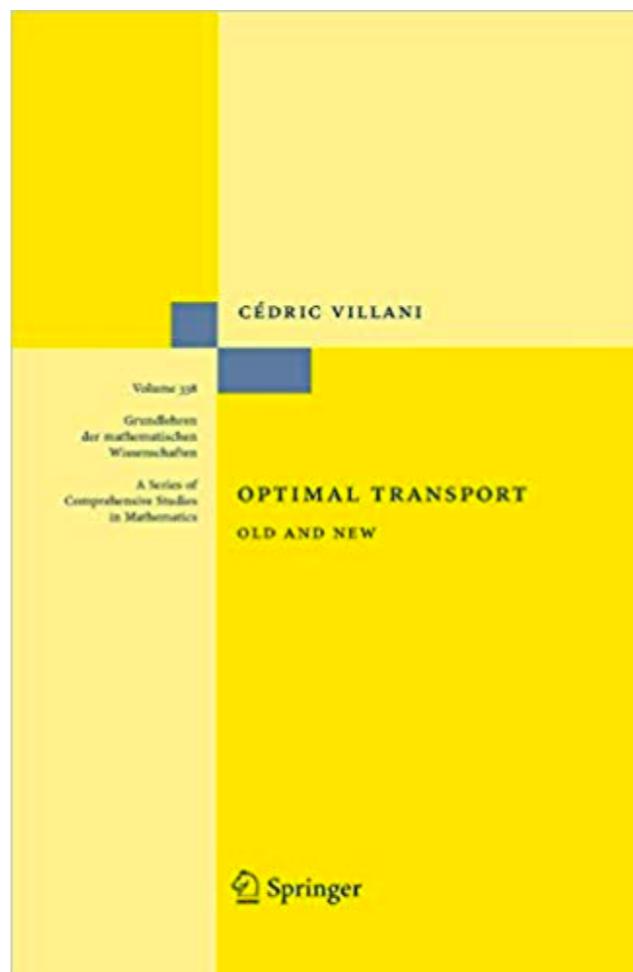
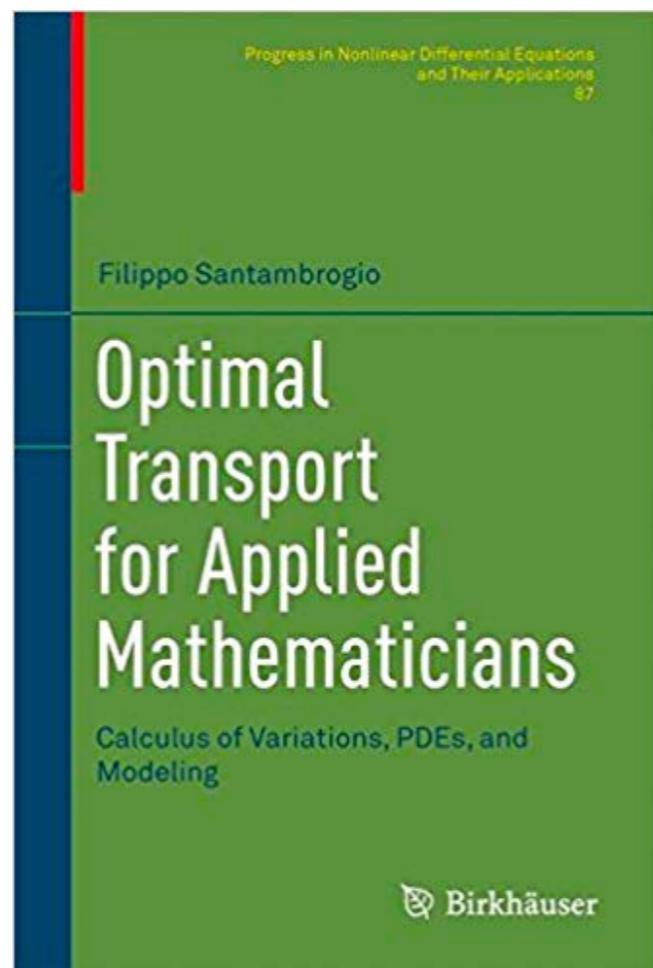
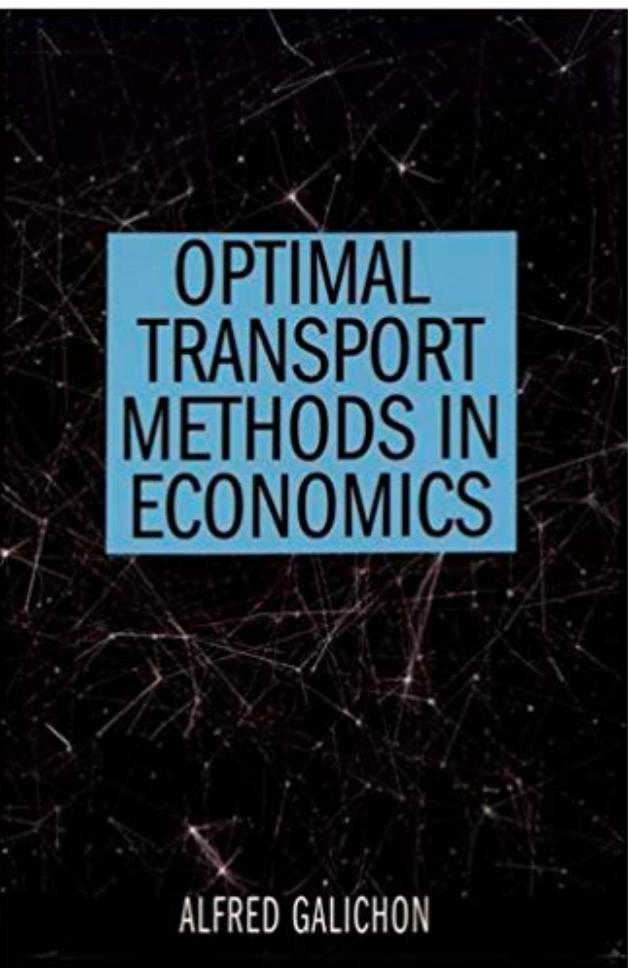
Computational Optimal
Transport

With Applications
to Data Science

Gabriel Peyré and Marco Cuturi

Why Optimal Transport?

Everybody seems to like it: economics, *applied* maths, *pure* maths, and even ML people!



Why Optimal Transport?

It has a bit of everything! theory, algorithms and applications, all building on endless supply of maths



Monge



Kantorovich



Koopmans

Nobel'75



Dantzig



Gangbo



Brenier



Otto



McCann



Caffarelli



Villani



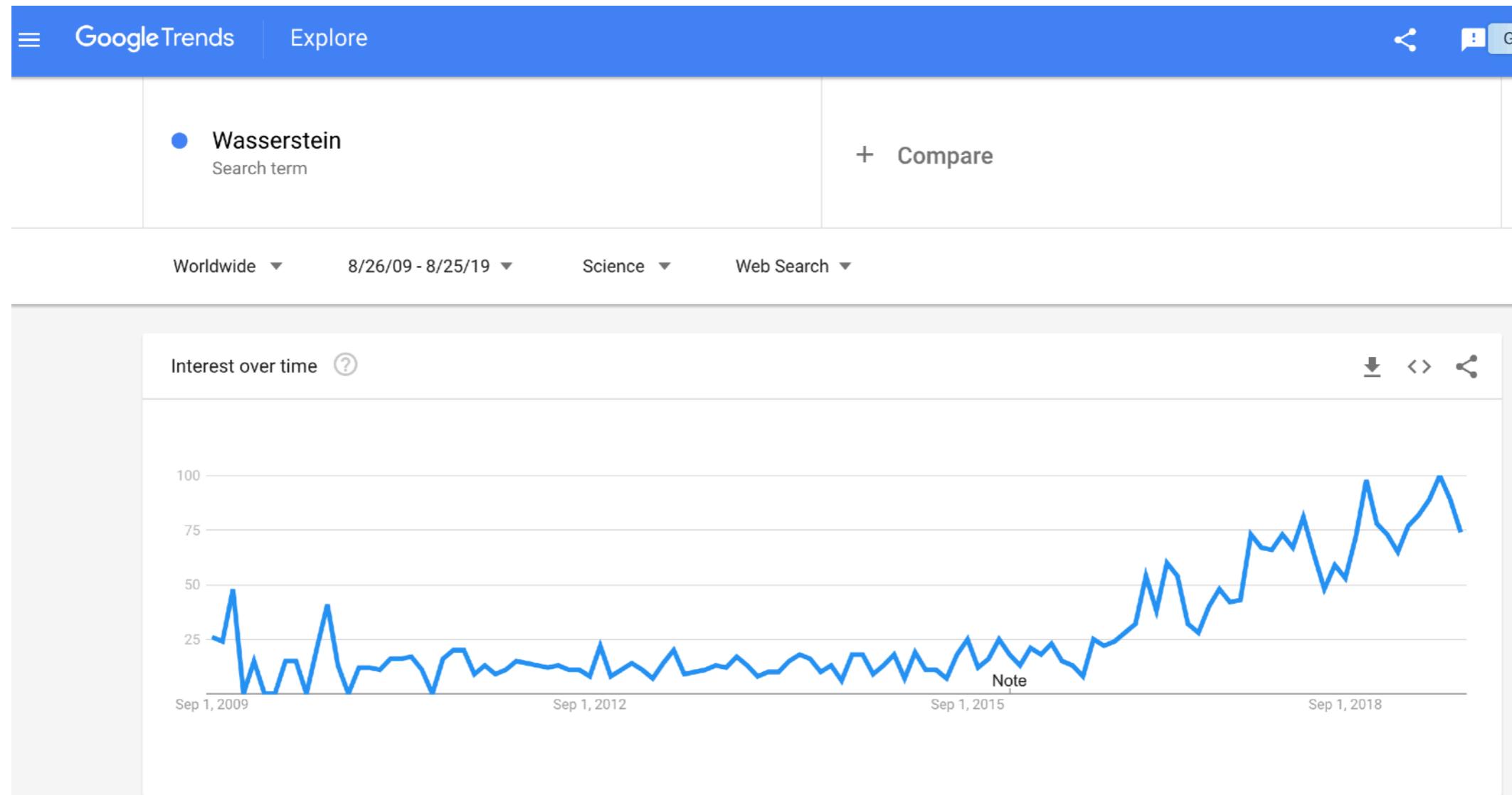
Figalli

Fields'10

Fields'18

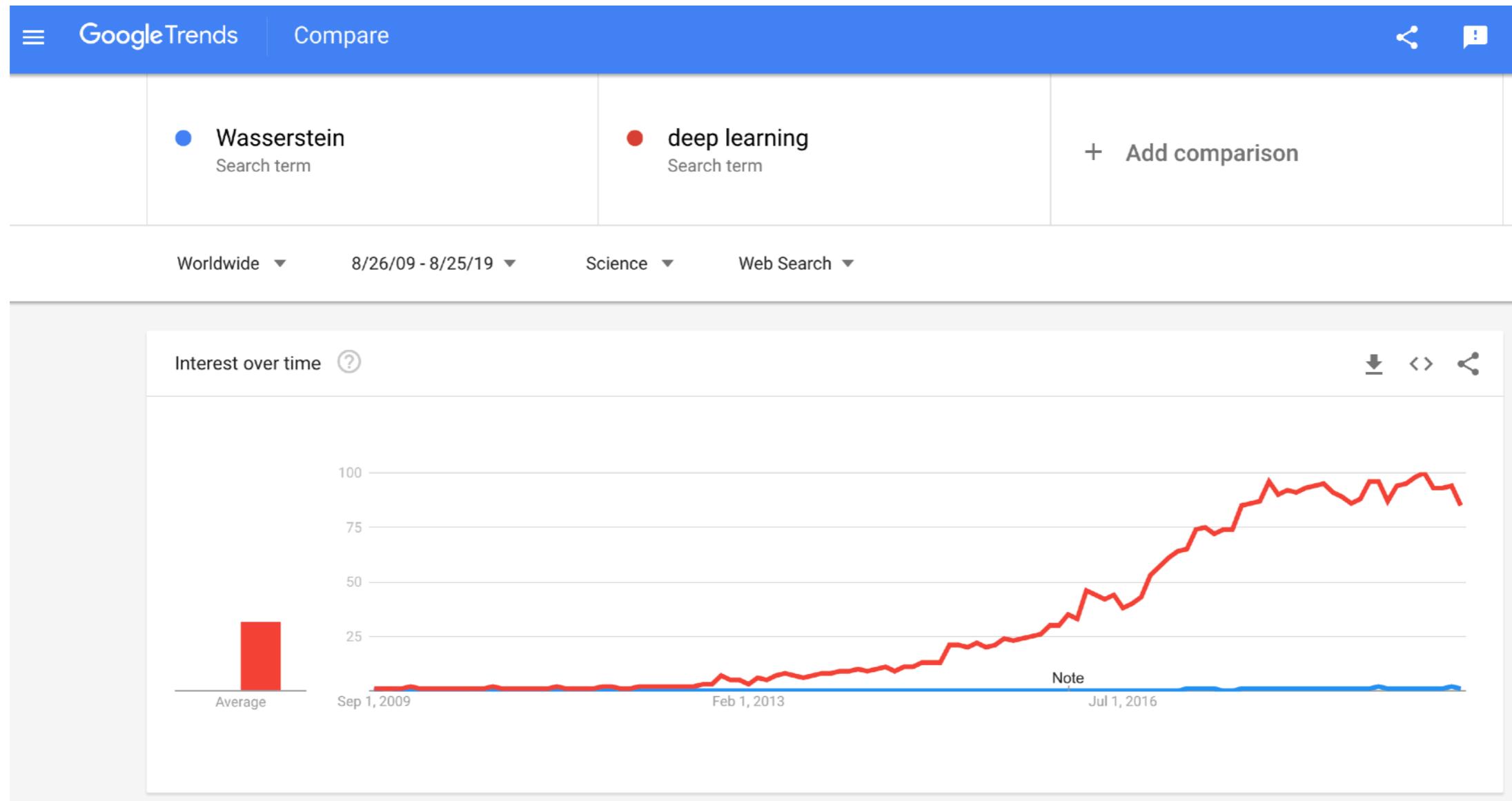
Why Optimal Transport?

It's trendy!



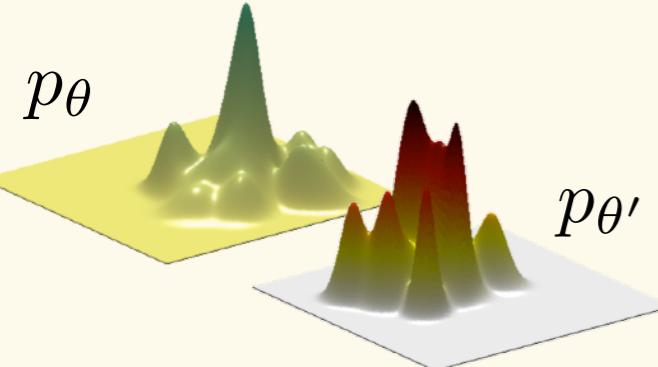
Why Optimal Transport?

Well... everything is relative



Optimal Transport in Data Sciences

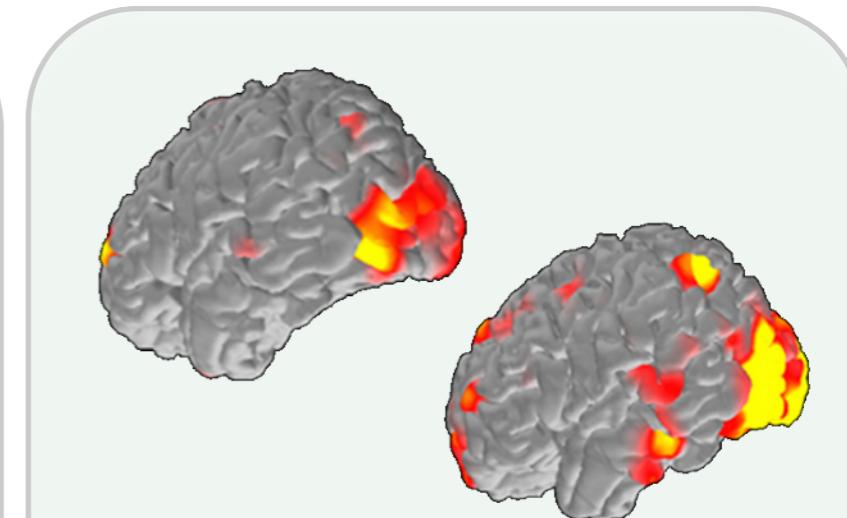
The natural geometry for probability measures



Statistical Models

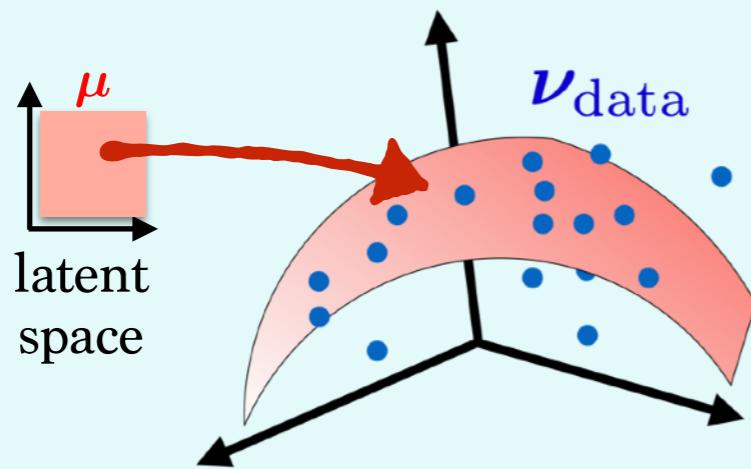


Bags
of features



Brain Activation Maps

Generative
Models
vs. data



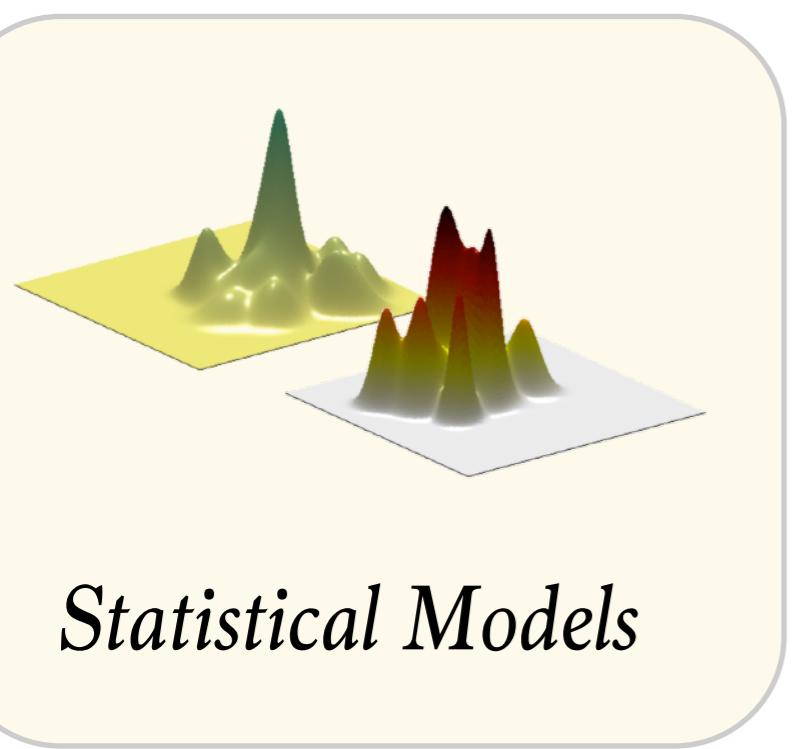
6



Color Histograms

Optimal Transport in Data Sciences

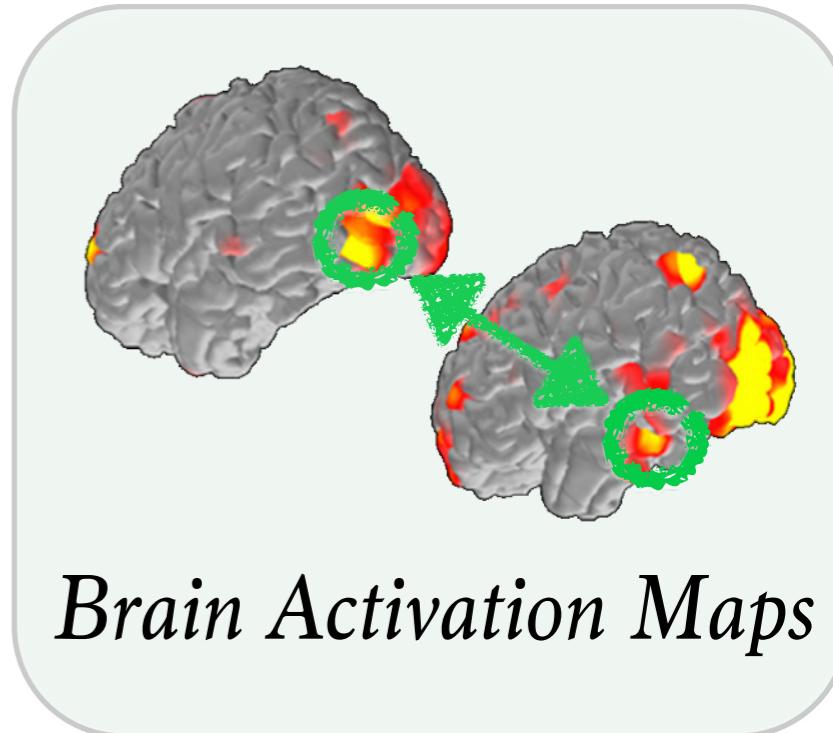
The natural geometry for probability measures supported on a geometric space.



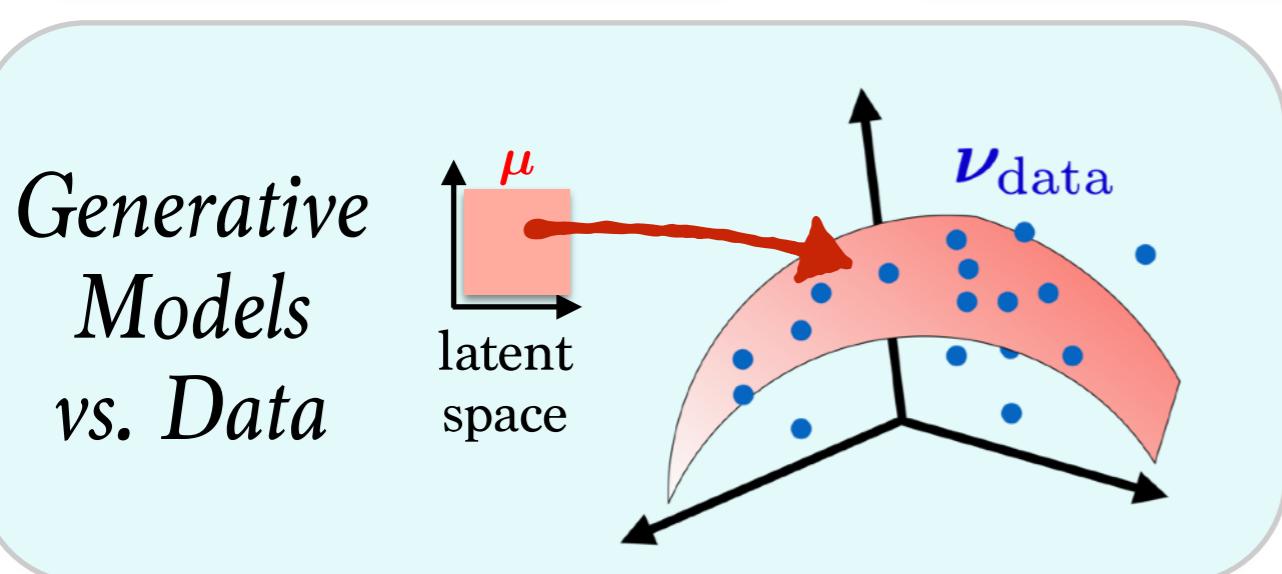
Statistical Models



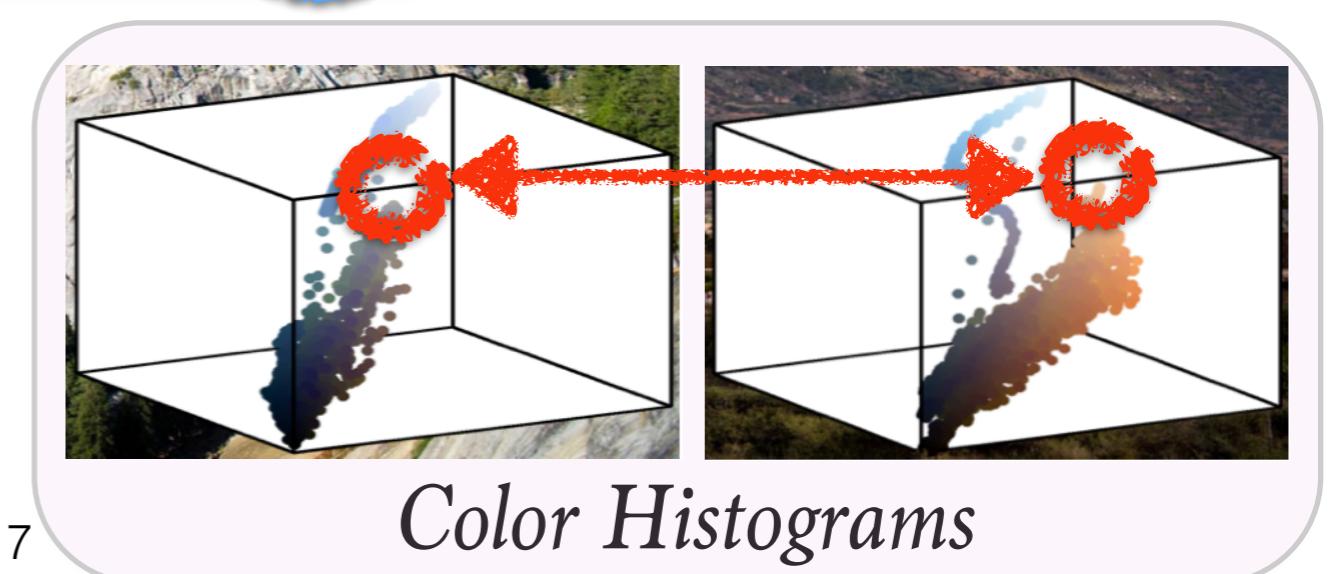
Bags of Features



Brain Activation Maps



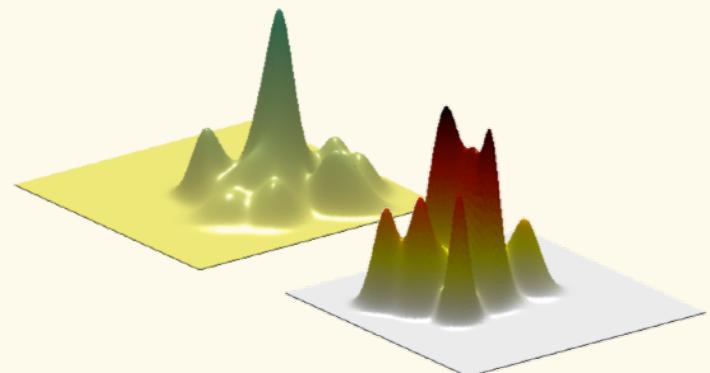
Generative Models vs. Data



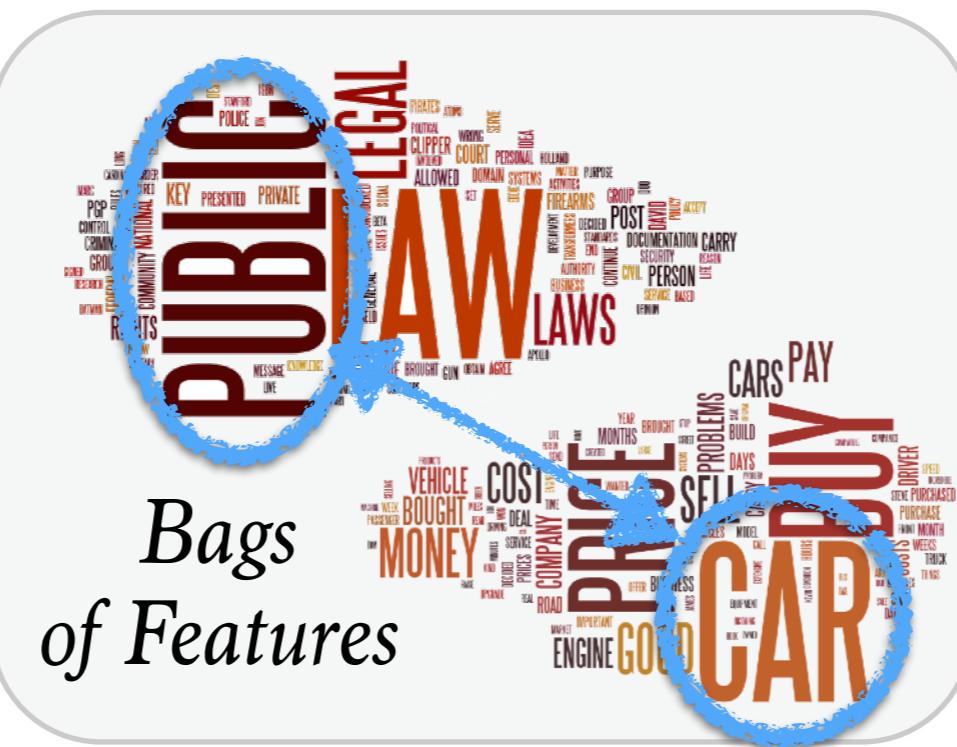
Color Histograms

Optimal Transport in Data Sciences

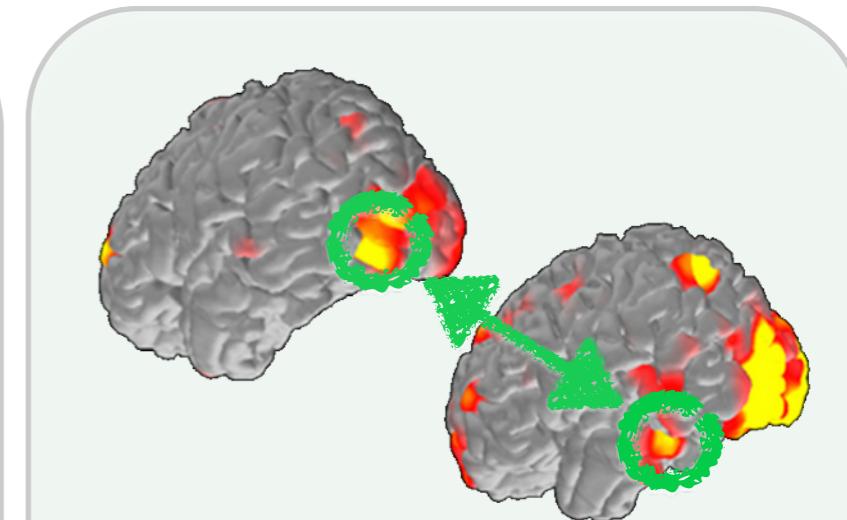
The natural geometry for probability measures supported on a geometric space.



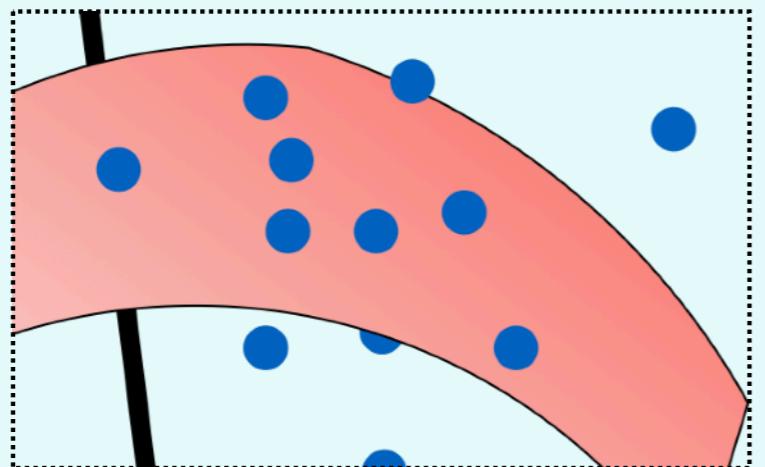
Statistical Models



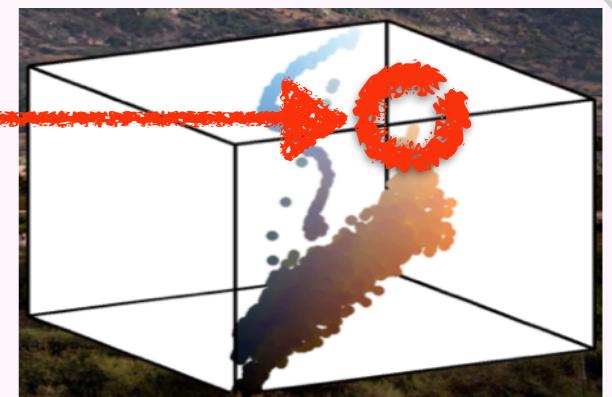
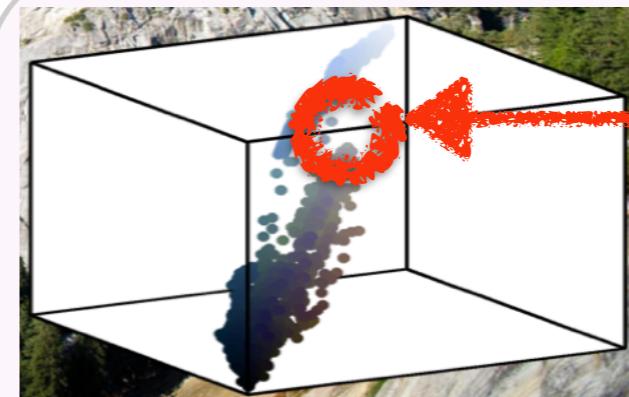
*Bags
of Features*



Brain Activation Maps



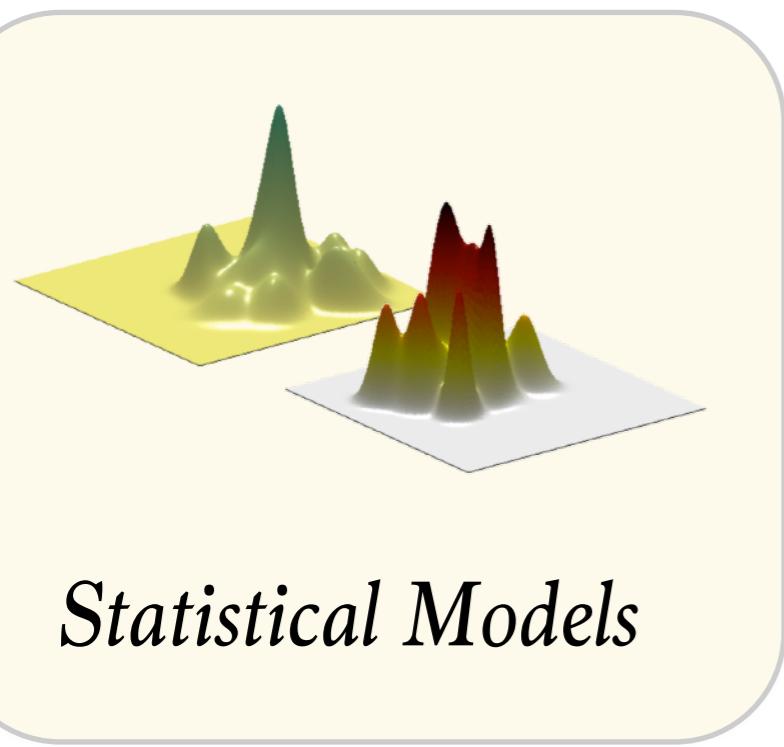
*Generative
Models
vs. Data*



Color Histograms

Optimal Transport in Data Sciences

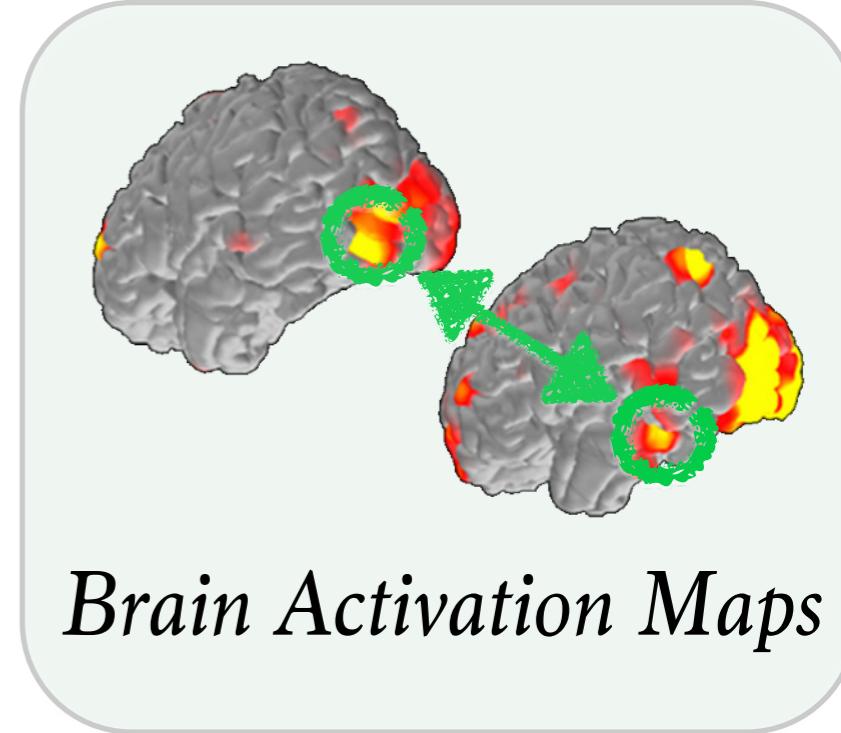
The natural geometry for probability measures supported on a geometric space.



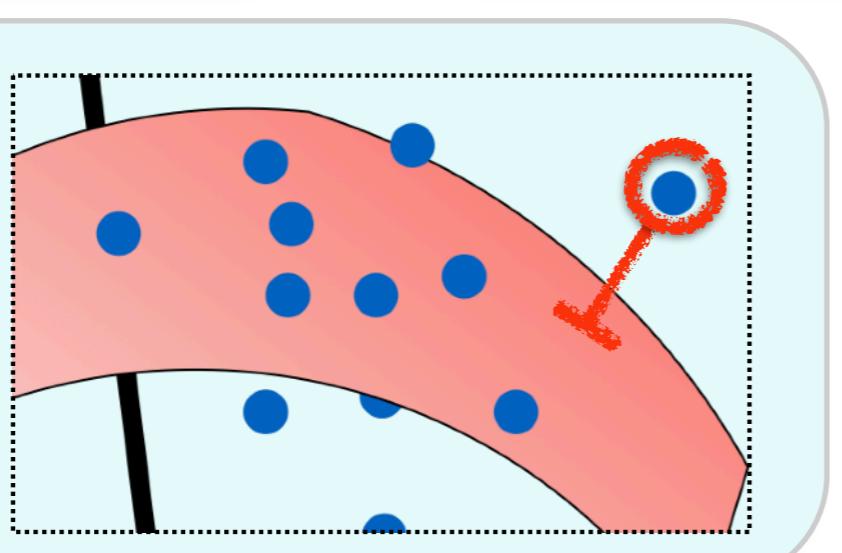
Statistical Models



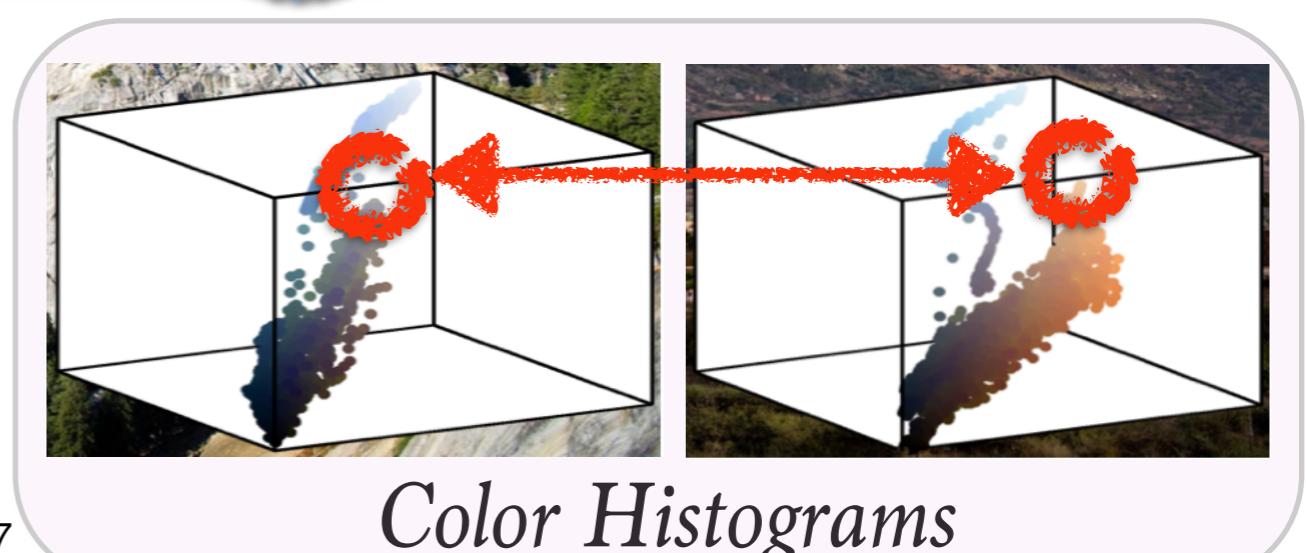
Bags of Features



Brain Activation Maps



Generative Models vs. Data



Color Histograms

Short Course Outline

1. Introduction to optimal transport
2. Computing OT exactly
3. Computing OT for data sciences
4. Some Applications

Introduction to OT

- Two examples: moving earth & soldiers
- Monge problem, Kantorovich problem
- OT as geometry, OT as a loss function

Origins: Monge Problem (1781)



Gaspard Monge (1746 - 1818)

Origins: Monge Problem (1781)

SOUTH AFRICA

Paris ▾

Basket Trips Inbox Profile Join Search

Travel feed: Paris Hotels Things to do Restaurants Flights Holiday Homes Shopping Car Hire •••

Europe > France > Ile-de-France > Paris > Things to do in Paris > Place Monge Place Monge, Paris: Address, Place Monge Reviews: 4/5

Place Monge

84 Reviews | #323 of 2 272 things to do in Paris | Shopping, Flea & Street Markets

Pl. Monge, Paris, France

Save Share

Review Highlights

"Good local market."

Place Monge market is one of our regular haunts when staying in Paris, it has a good range of... [read more](#)

 Reviewed 26 September 2018
johngl8492UH , Middle Park via mobile

"One of the most amazing streets we have be..."

We loved place monge. All the shops slide their products out to the streets, it has a real Parisian... [read more](#)

 Reviewed 11 October 2018
Steve L , Pacific Coast Australia, Australia via mobile

[Read all 84 reviews](#)



All photos (35)



Origins: Monge Problem (1781)



MÉMOIRES DE L'ACADEMIE ROYALE

*MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.*

Par M. MONGE.

Lorsqu'on doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Origins: Monge Problem (1781)



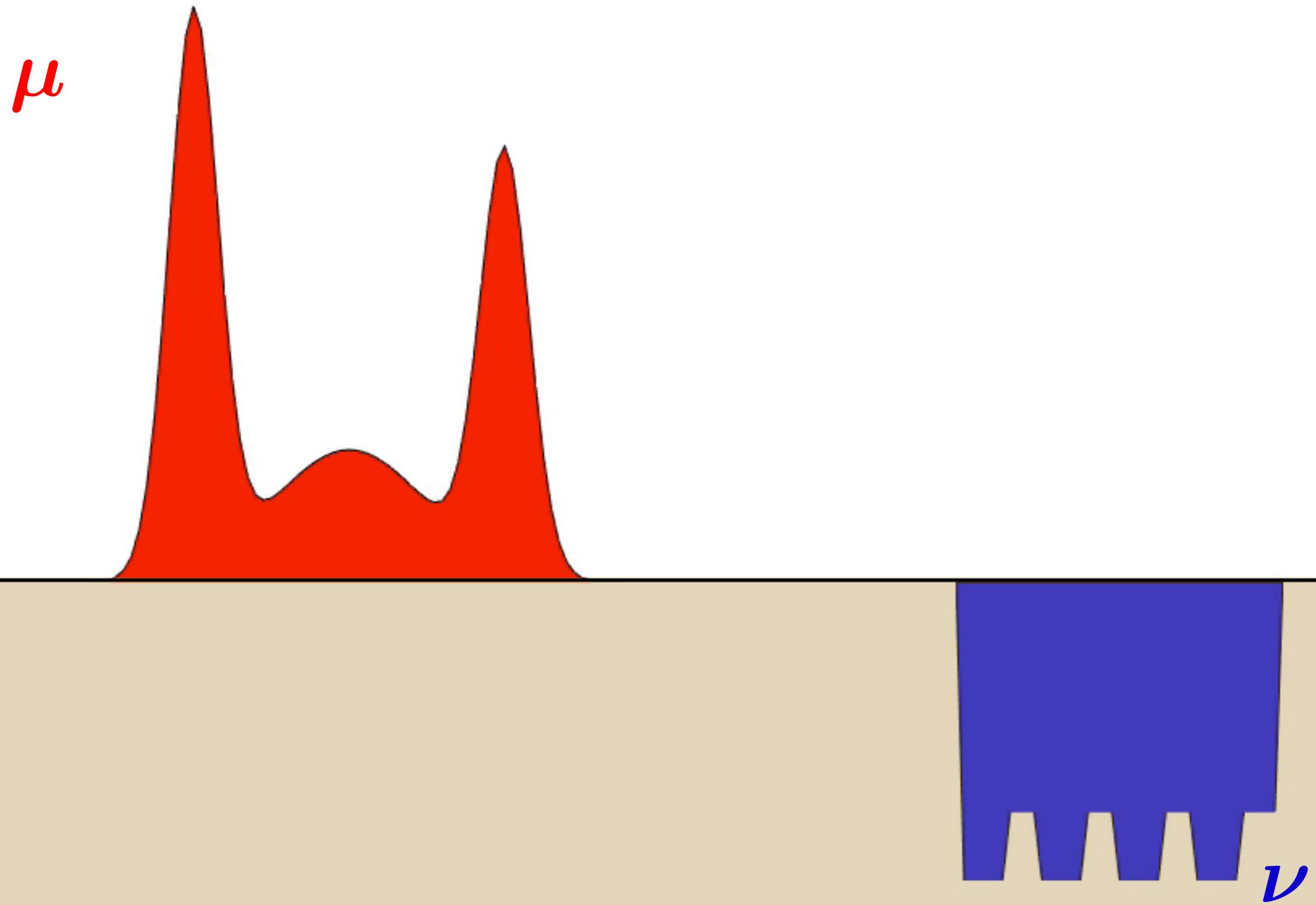
60 MÉMOIRES DE L'ACADEMIE ROYALE

MÉMOIRE
SUR LA
THÉORIE DES NÉBLAIS

*When one has to bring earth
from one place to another...*

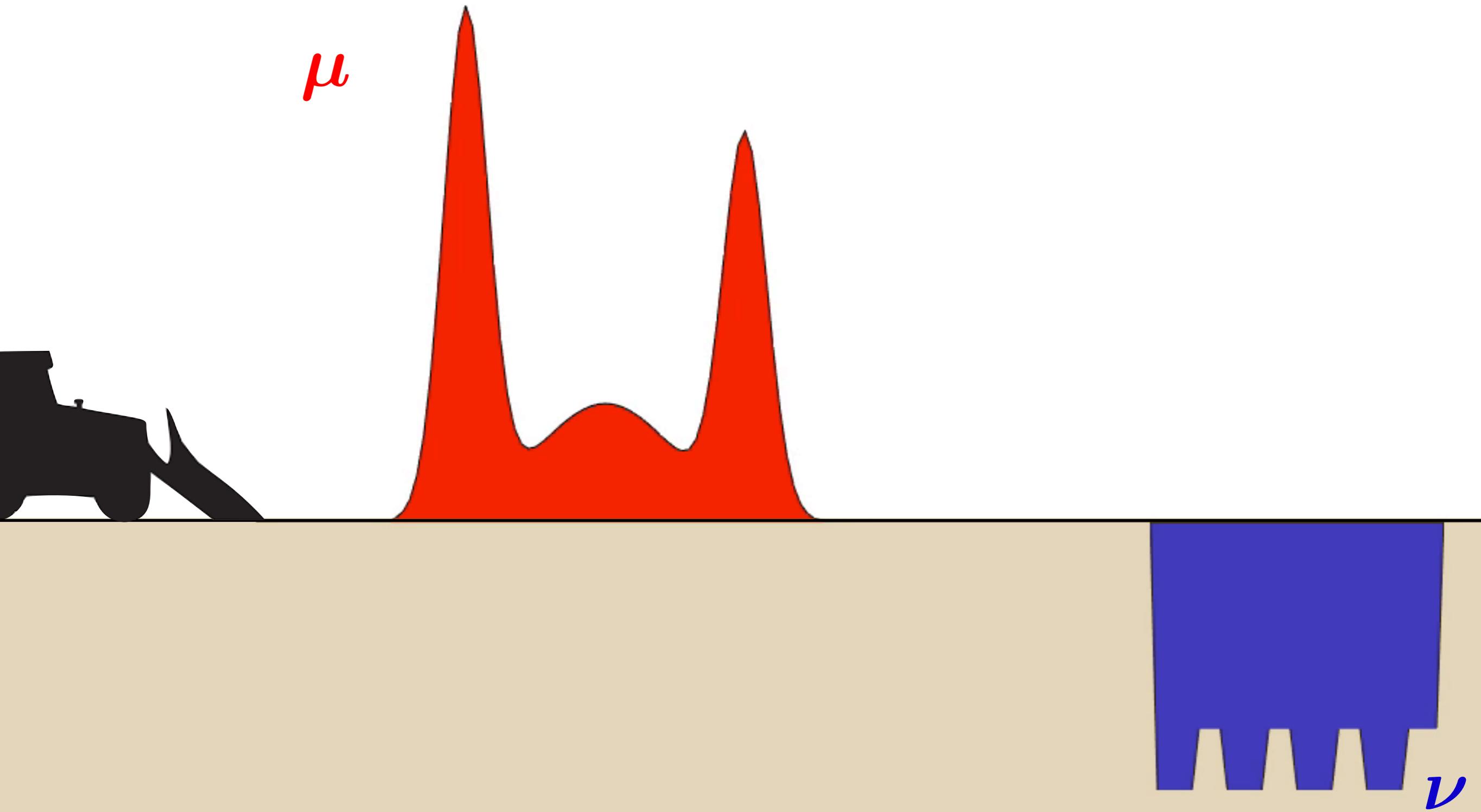
LORSQU'ON doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Origins: Monge Problem



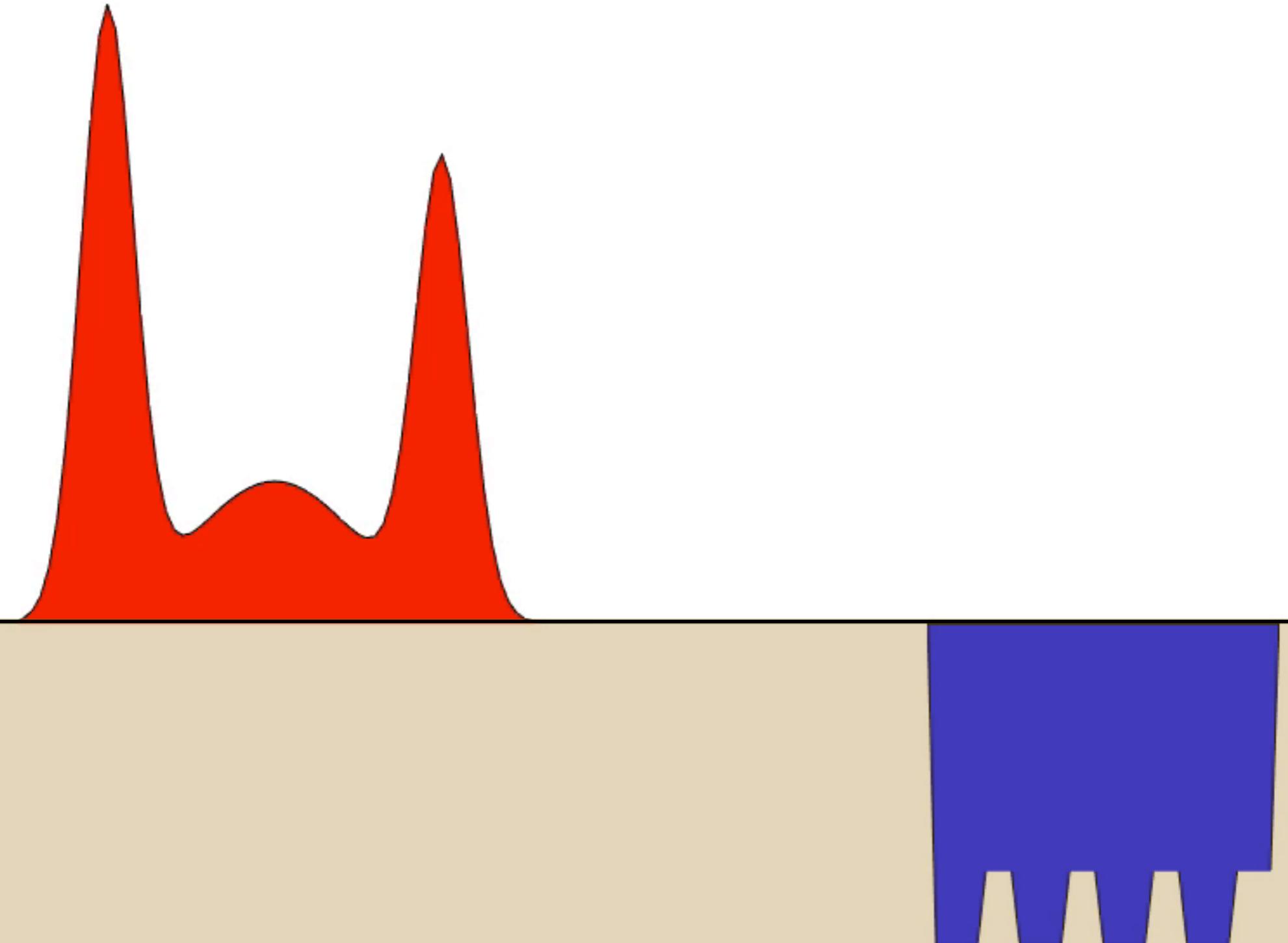
Origins: Monge Problem

In the 21st Century...



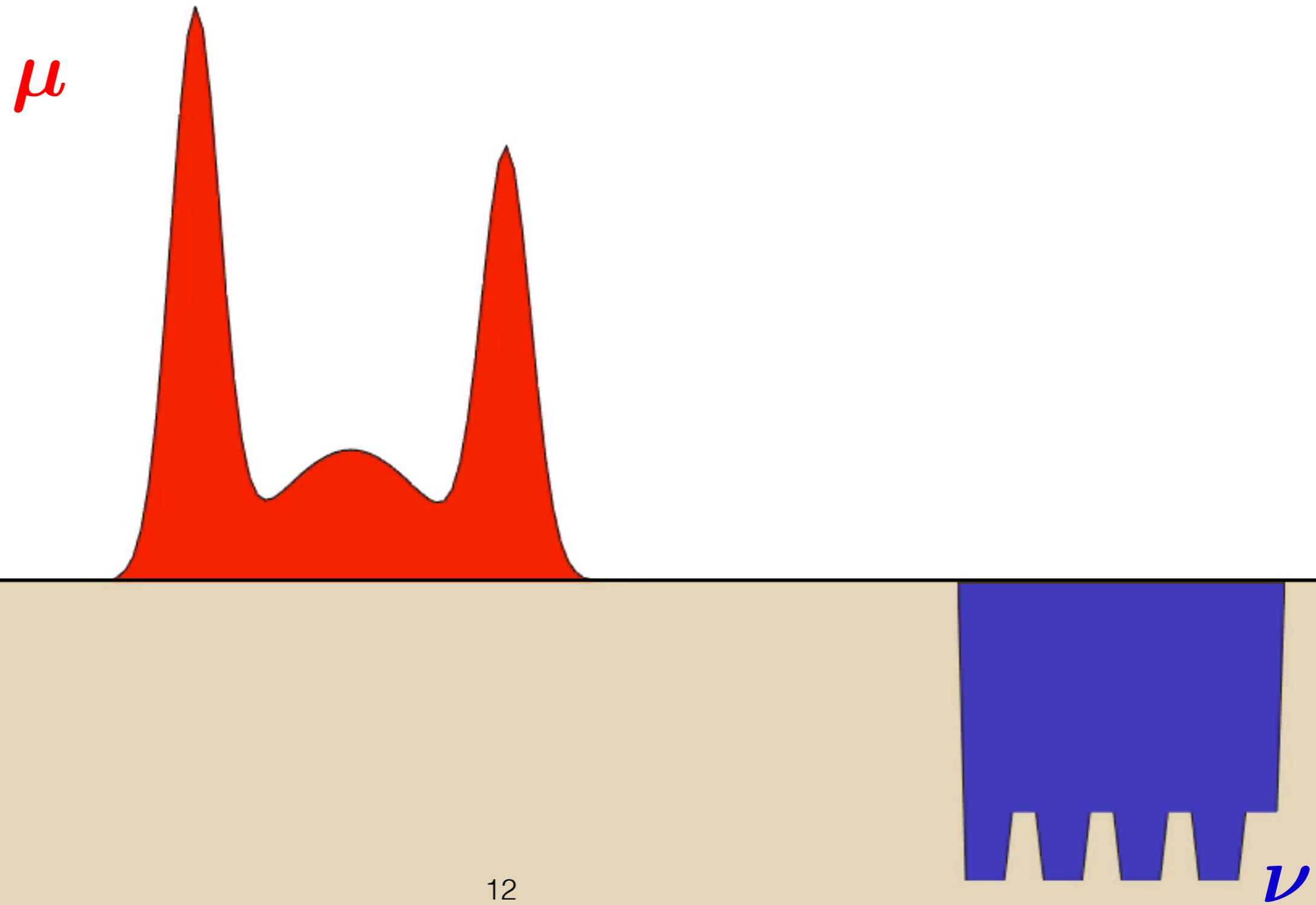
Origins: Monge Problem

In the 21st Century...



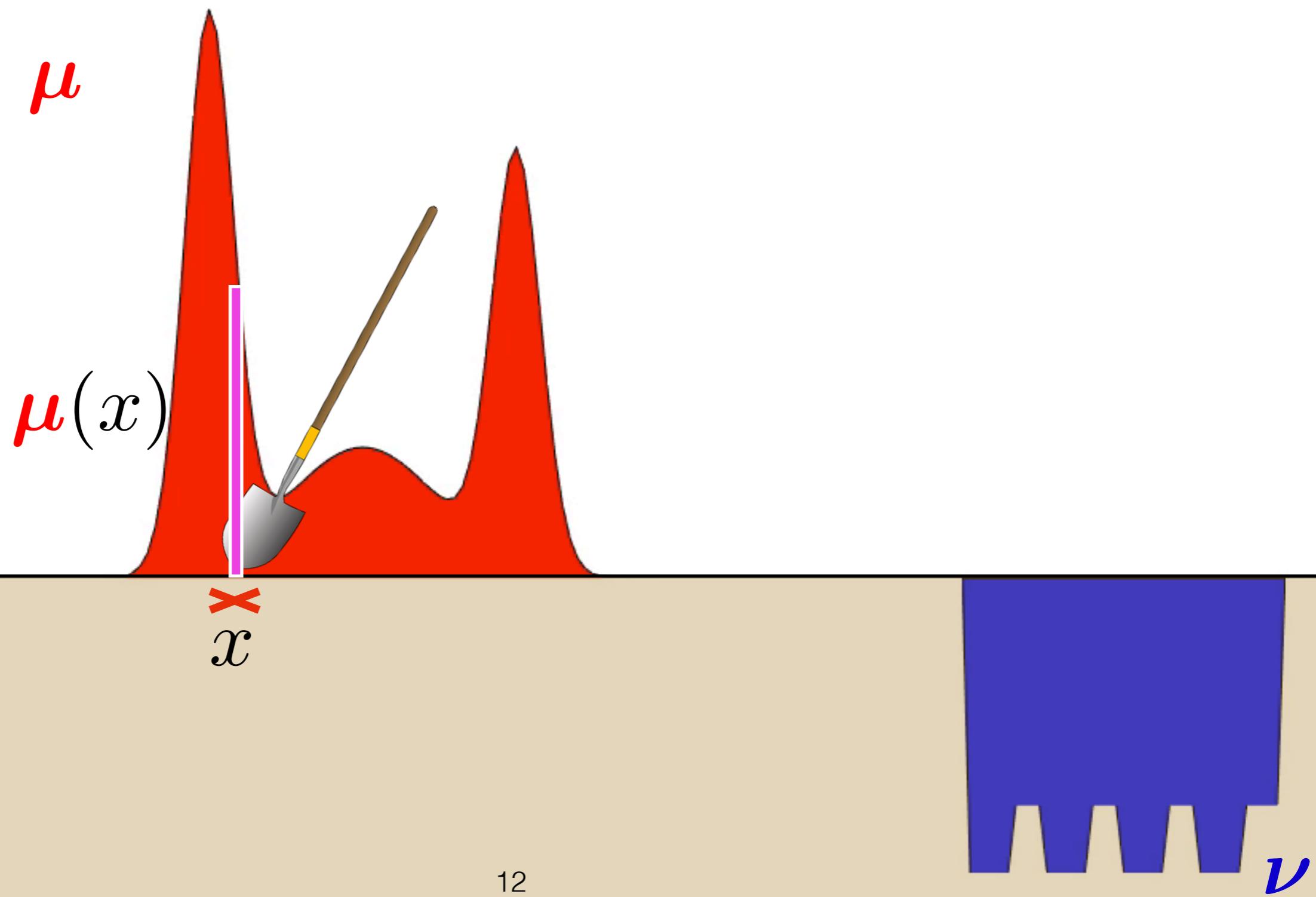
Origins: Monge's Problem

In 1781 however...



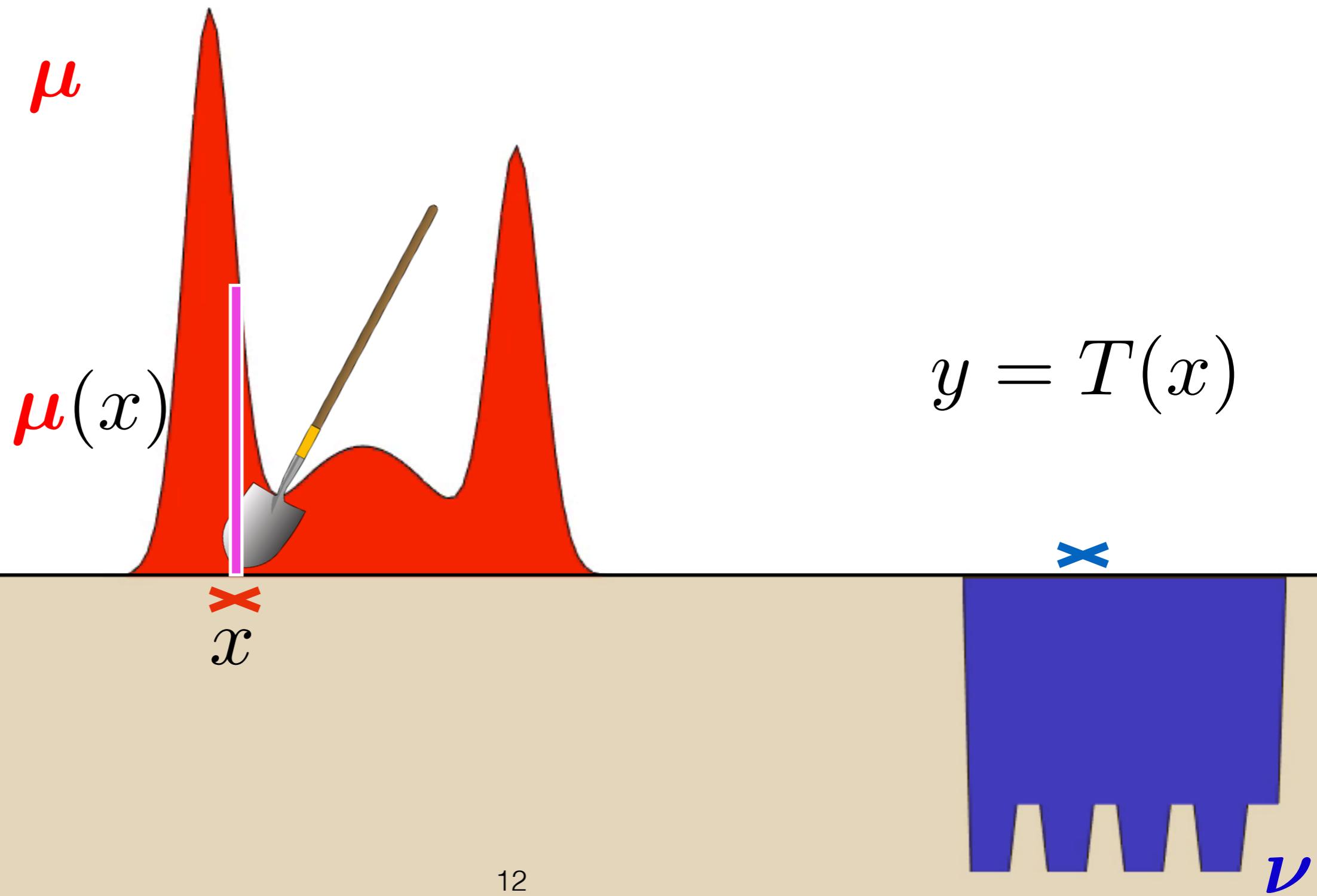
Origins: Monge's Problem

In 1781 however...



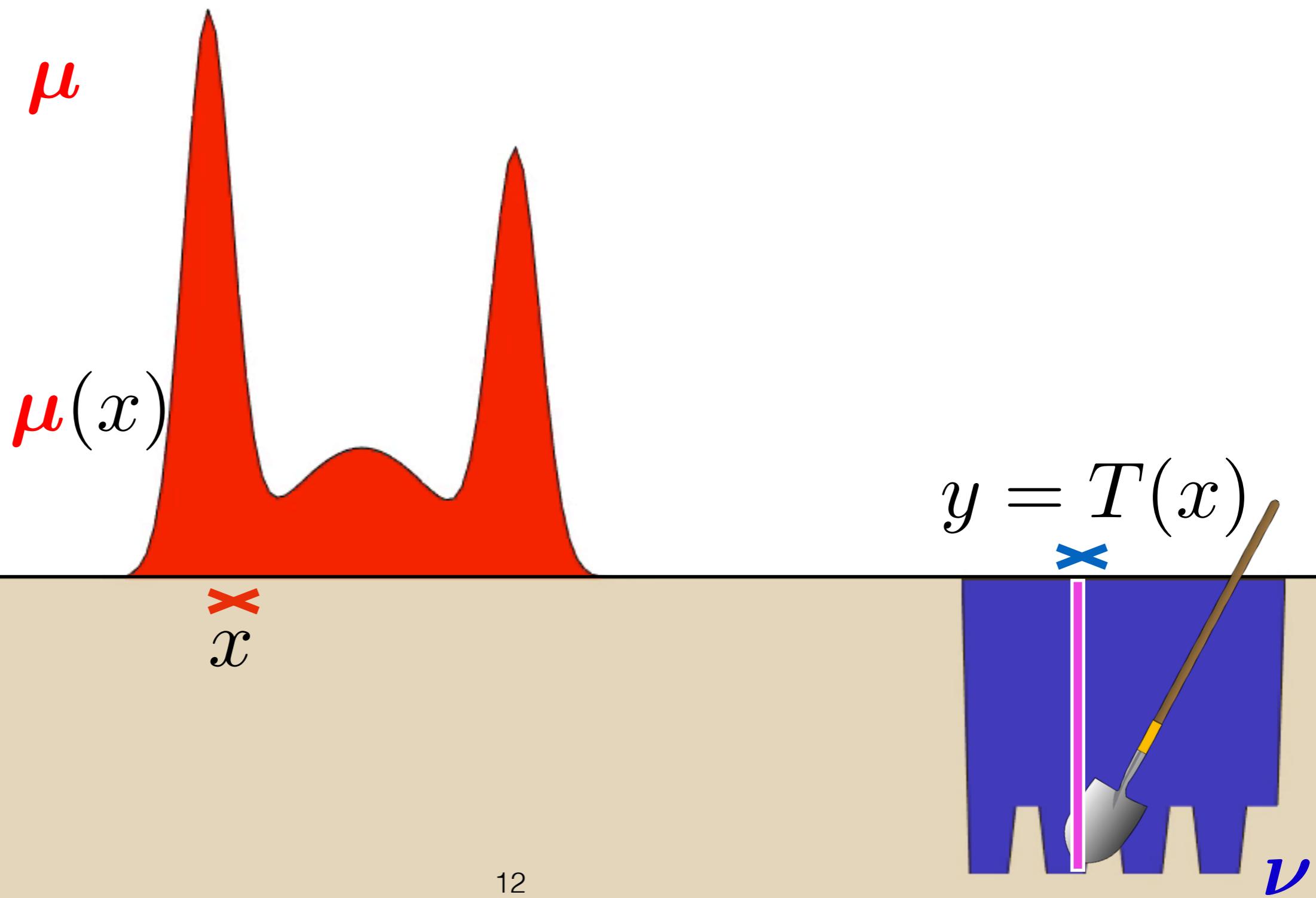
Origins: Monge's Problem

In 1781 however...



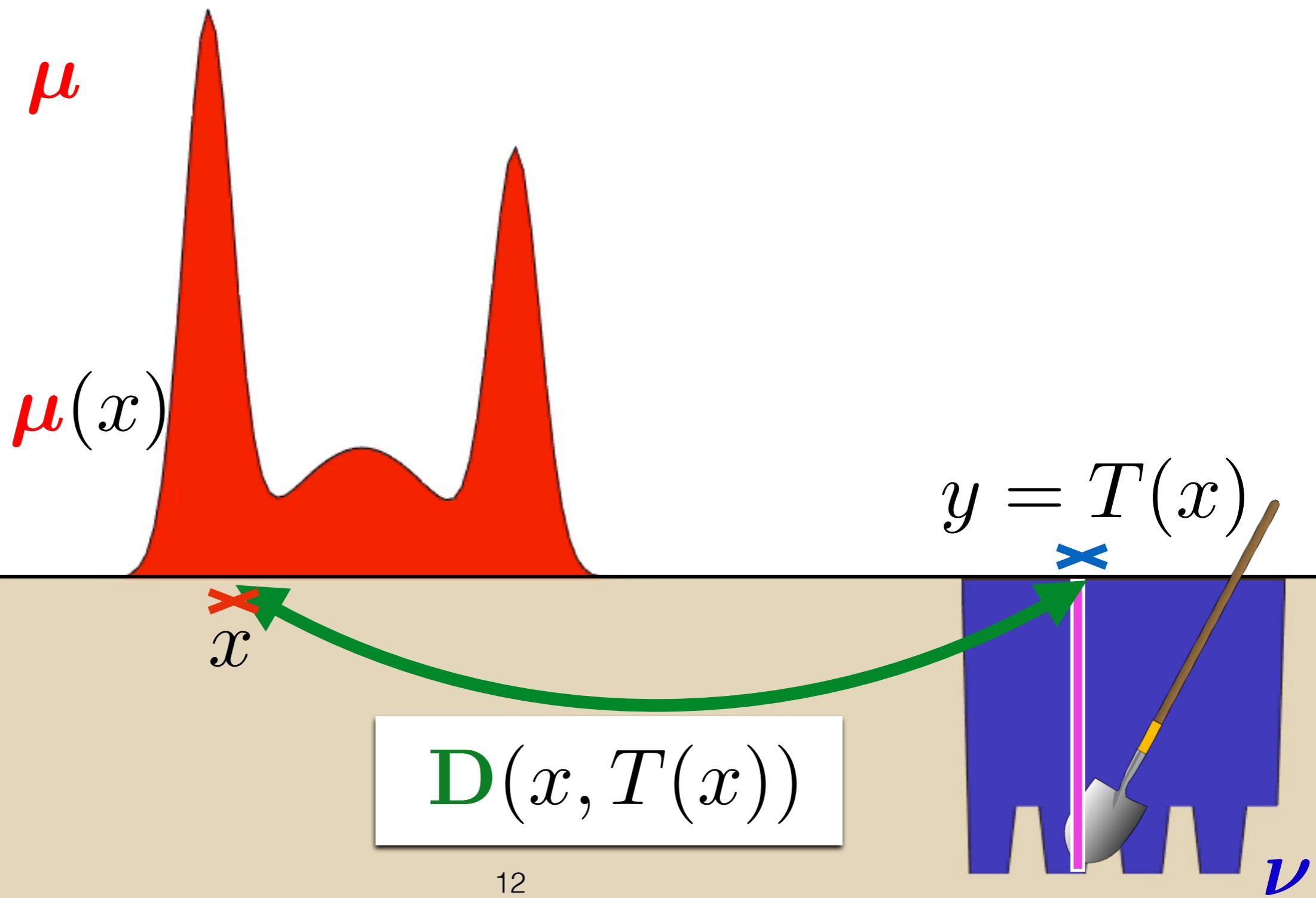
Origins: Monge's Problem

In 1781 however...



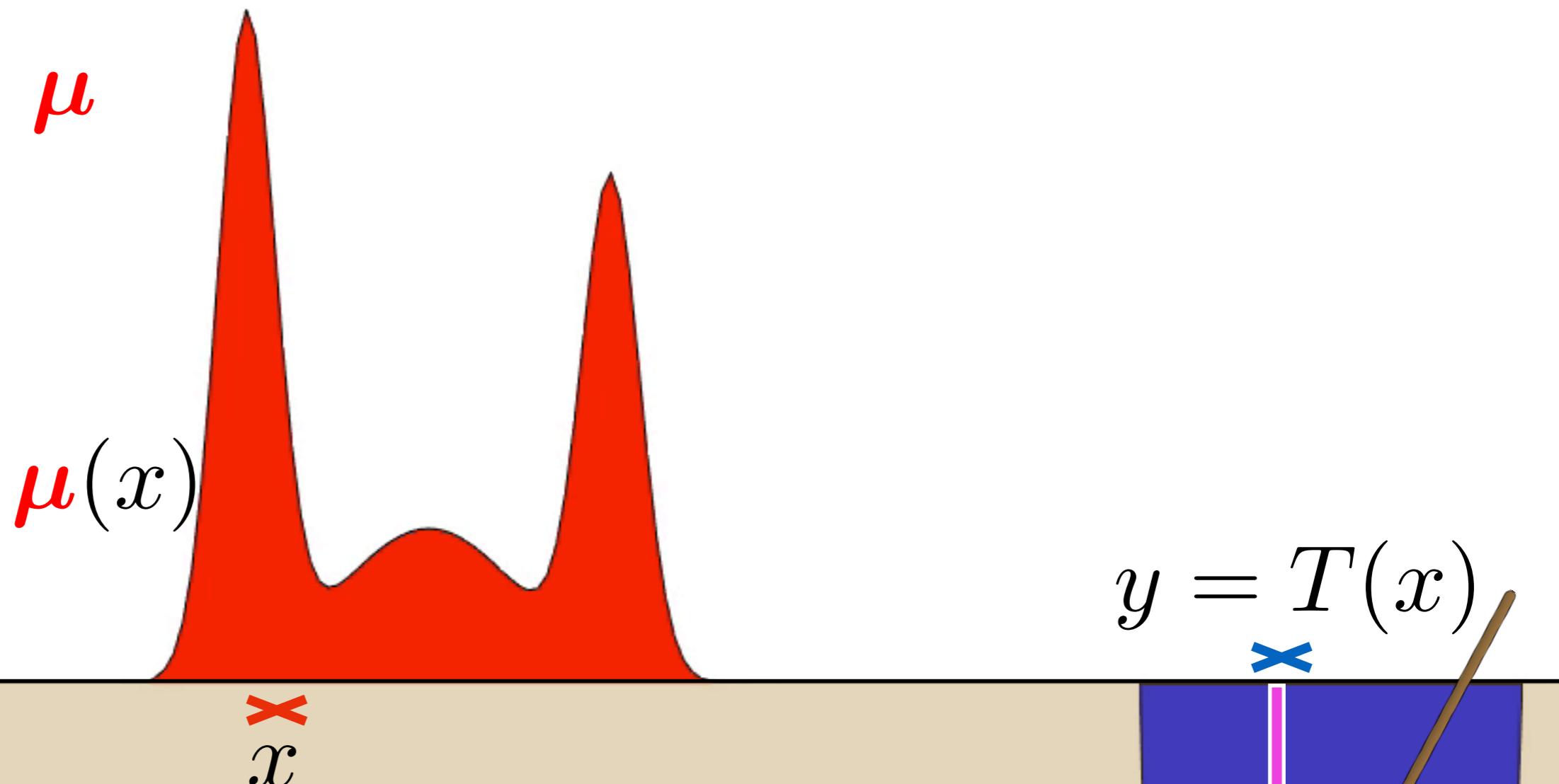
Origins: Monge's Problem

In 1781 however...



Origins: Monge's Problem

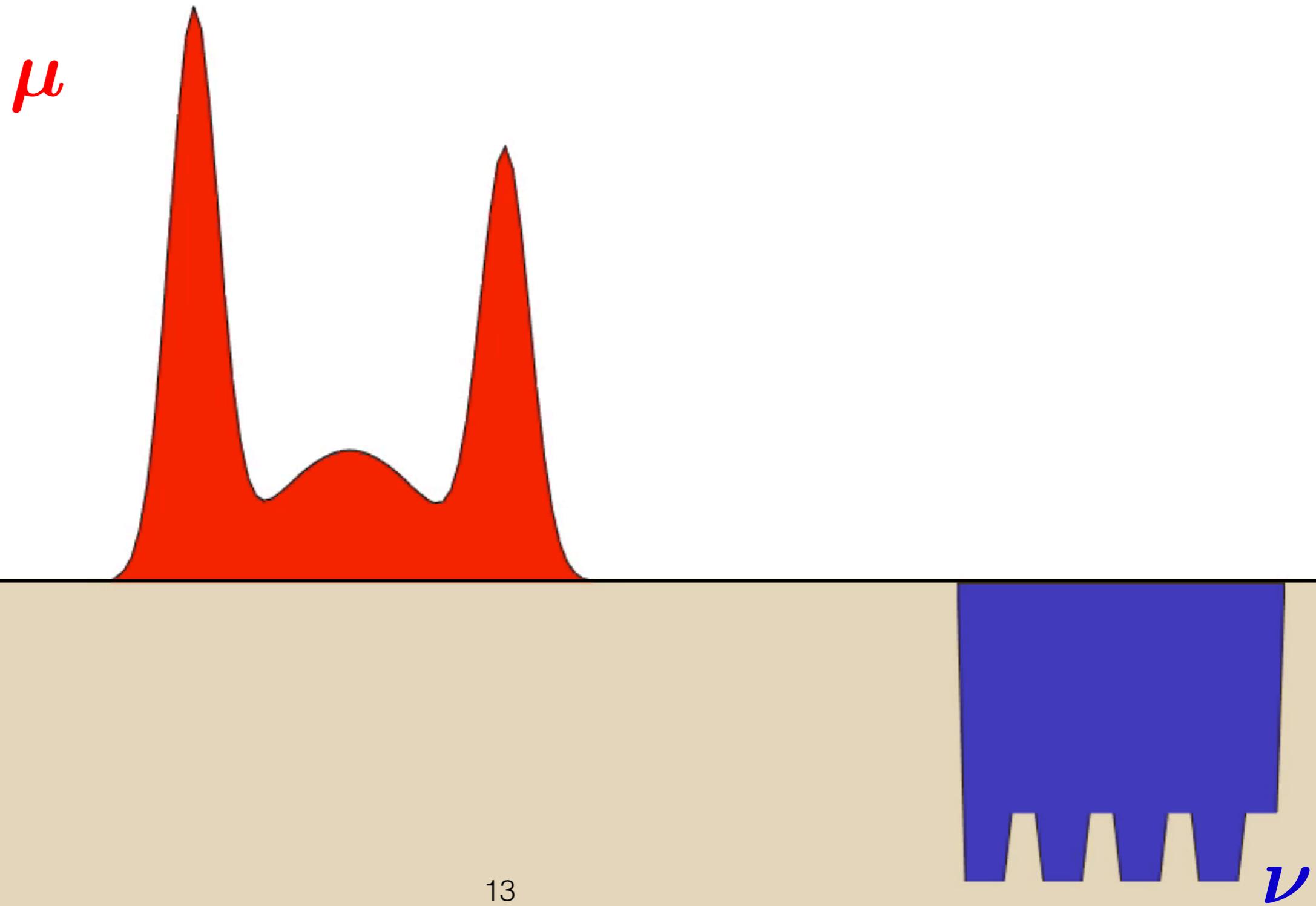
In 1781 however...



work: $\mu(x) \mathcal{D}(x, T(x))$

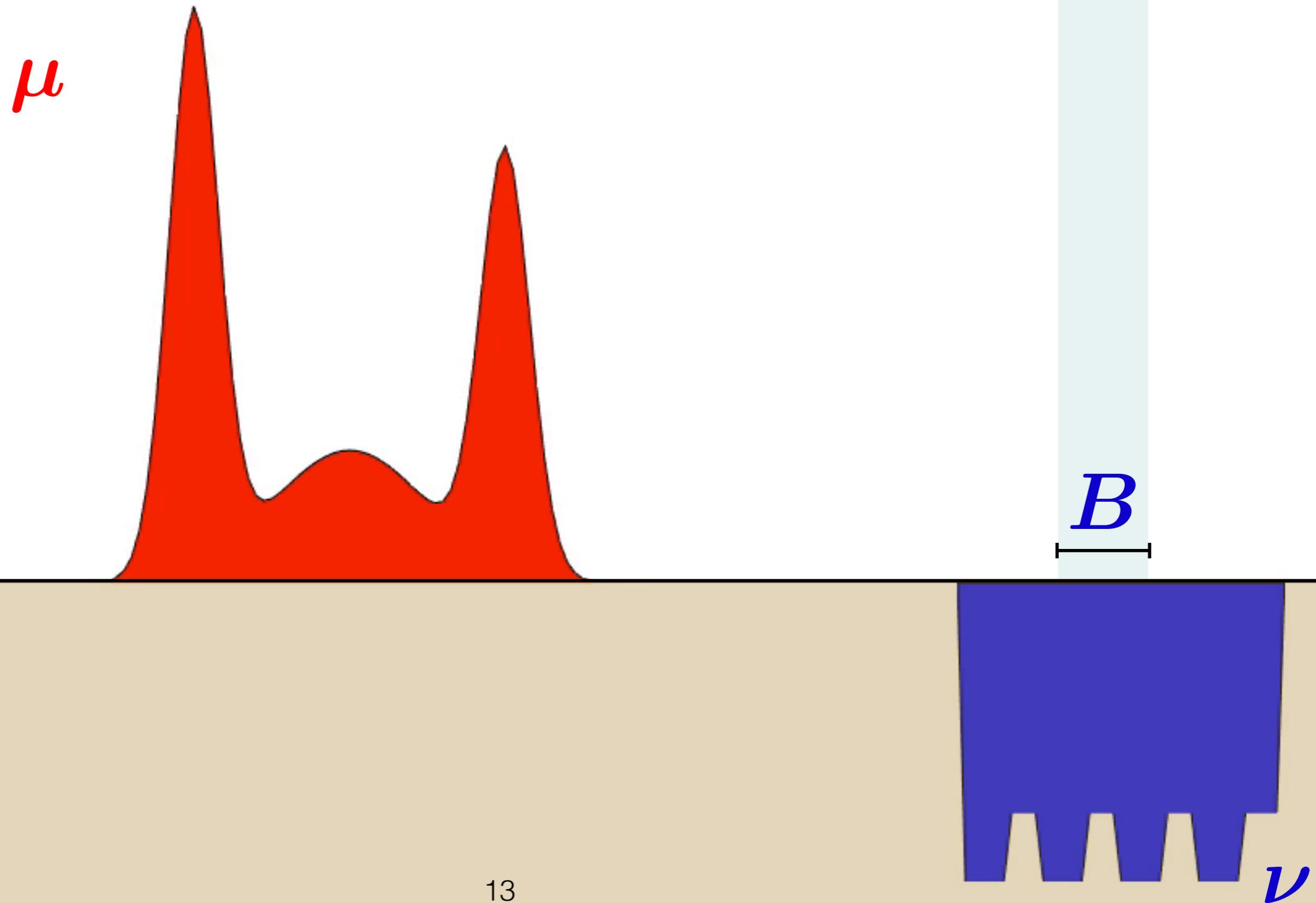
Origins: Monge's Problem

T must map red to blue.



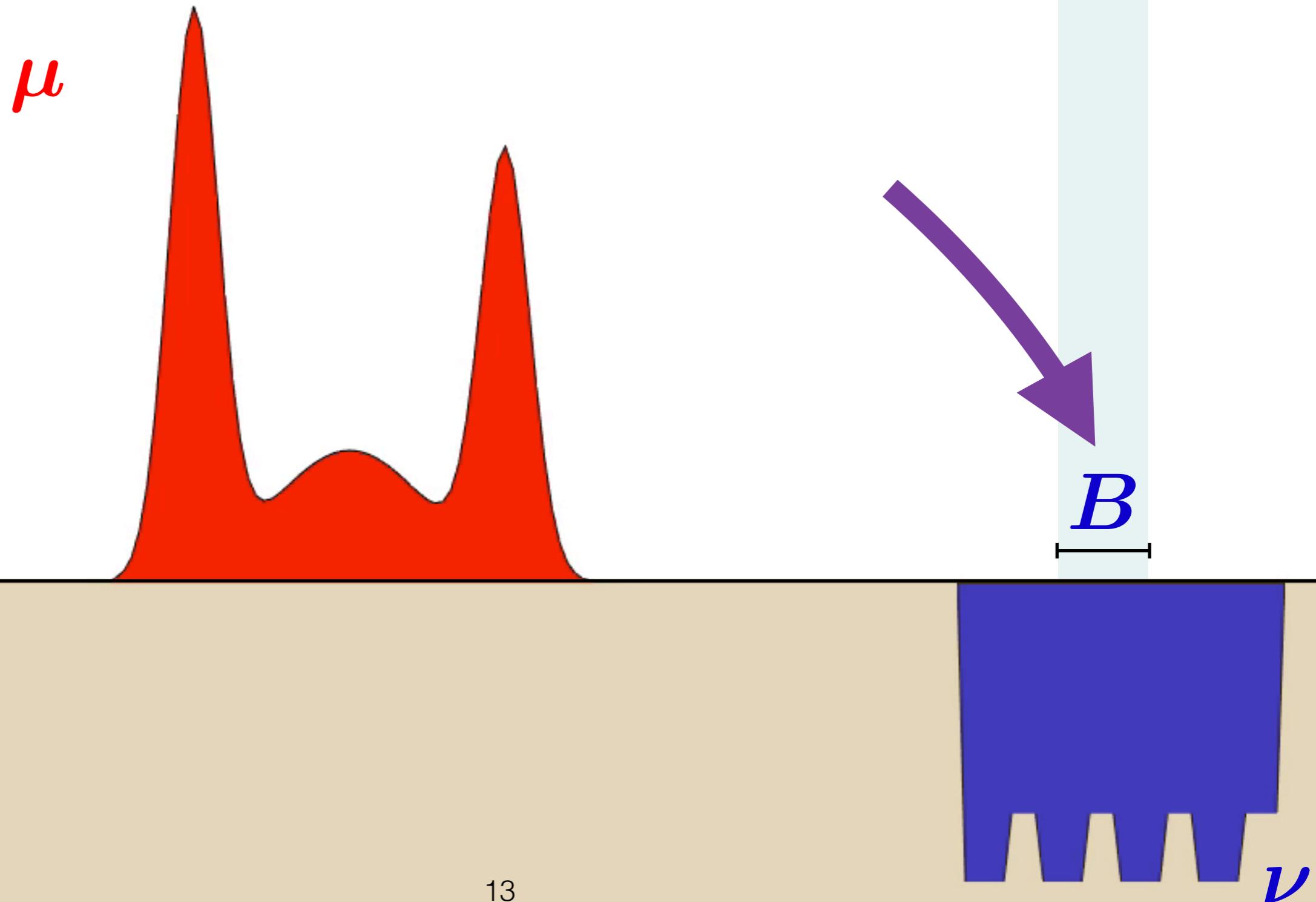
Origins: Monge's Problem

T must map red to blue.



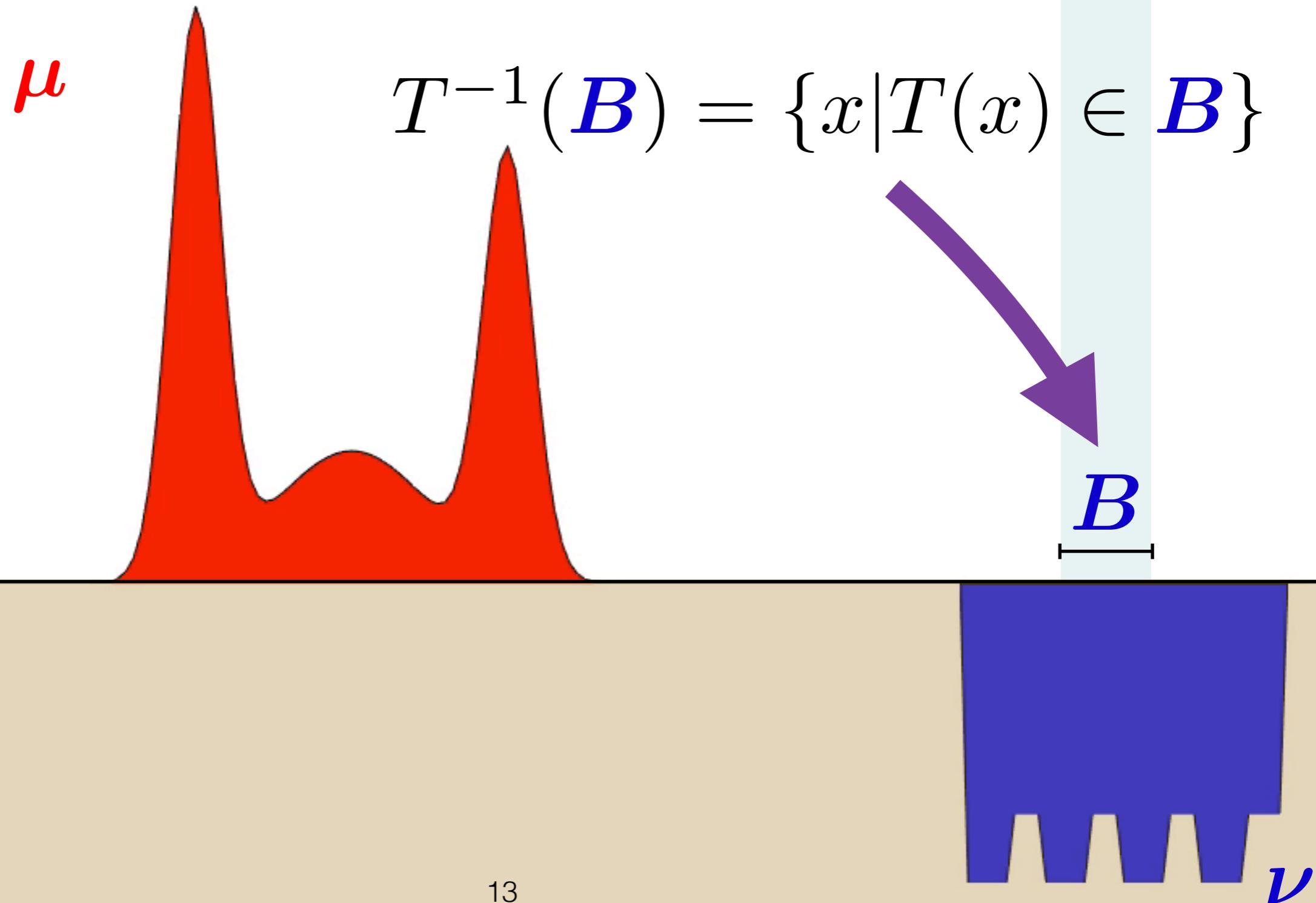
Origins: Monge's Problem

T must map red to blue.



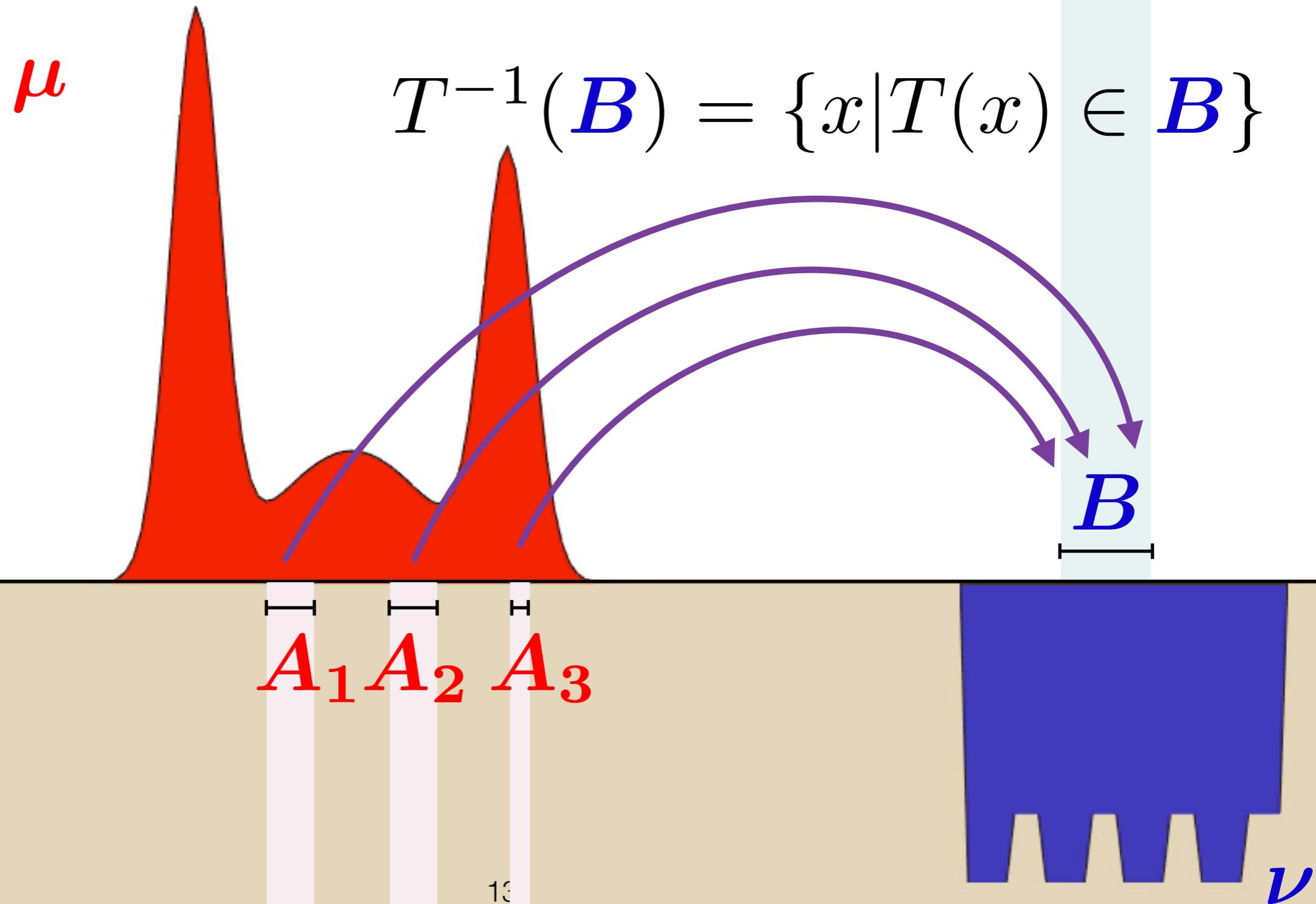
Origins: Monge's Problem

T must map red to blue.



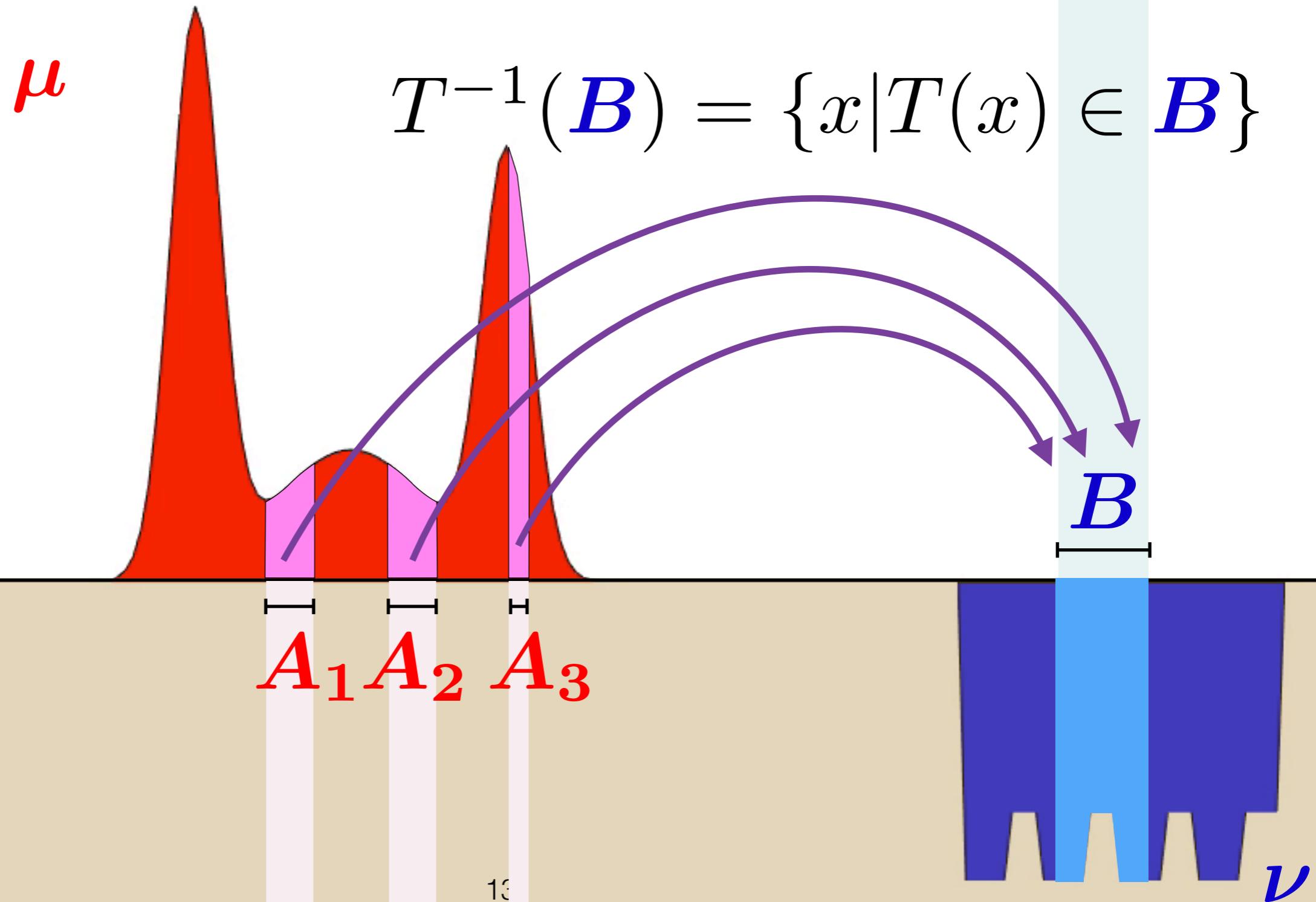
Origins: Monge's Problem

T must map red to blue.



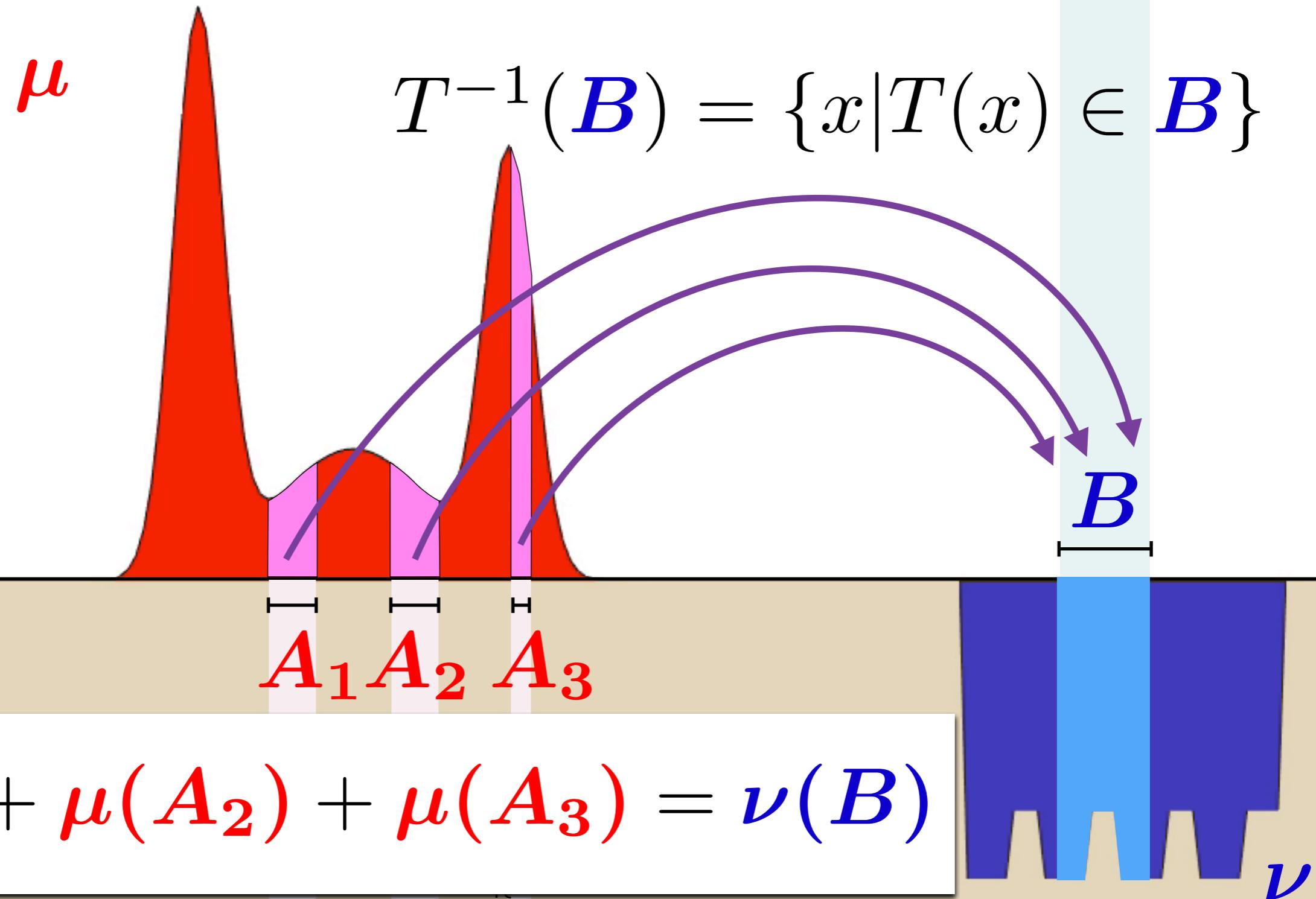
Origins: Monge's Problem

T must map red to blue.



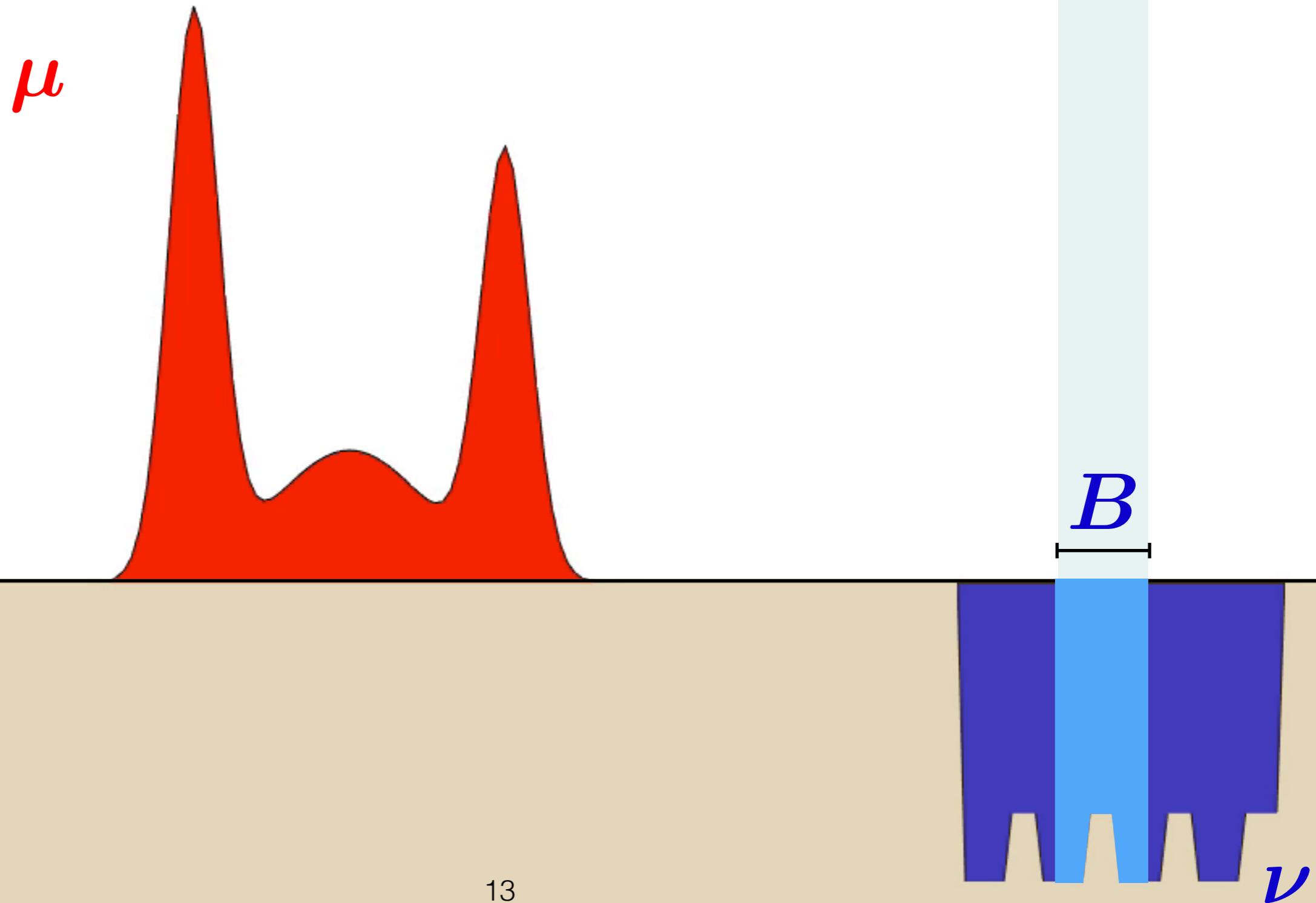
Origins: Monge's Problem

T must map red to blue.



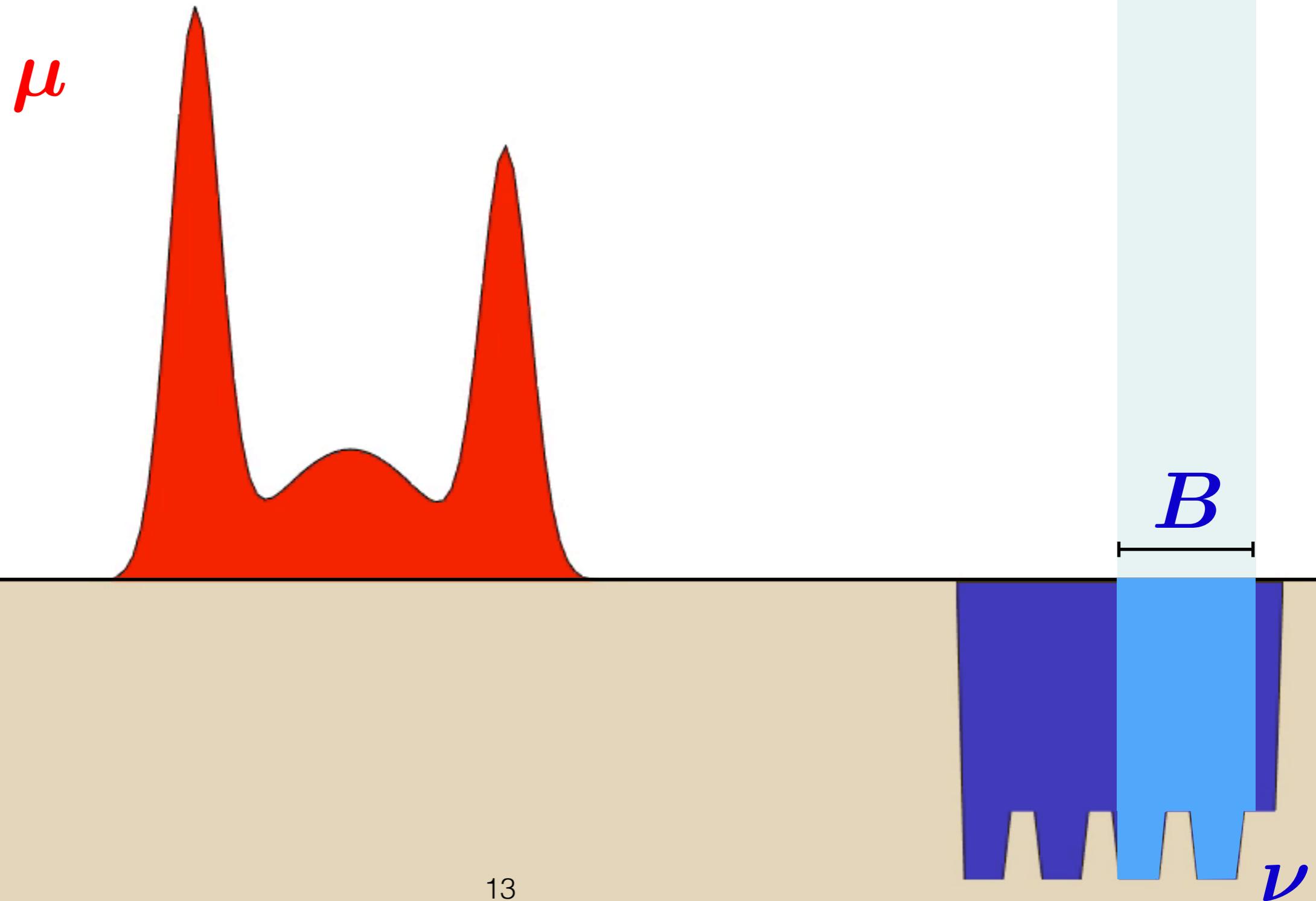
Origins: Monge's Problem

T must map red to blue.



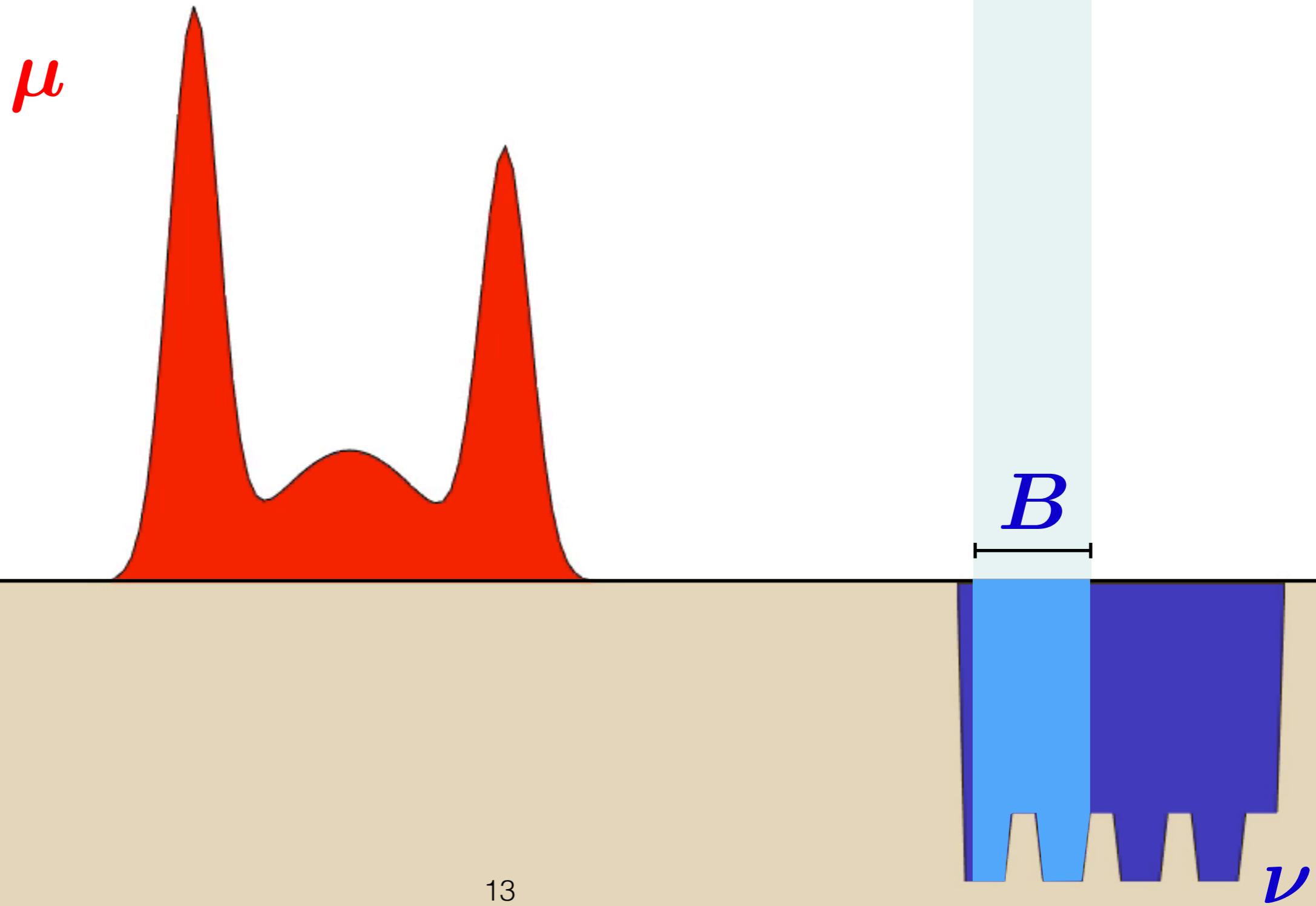
Origins: Monge's Problem

T must map red to blue.



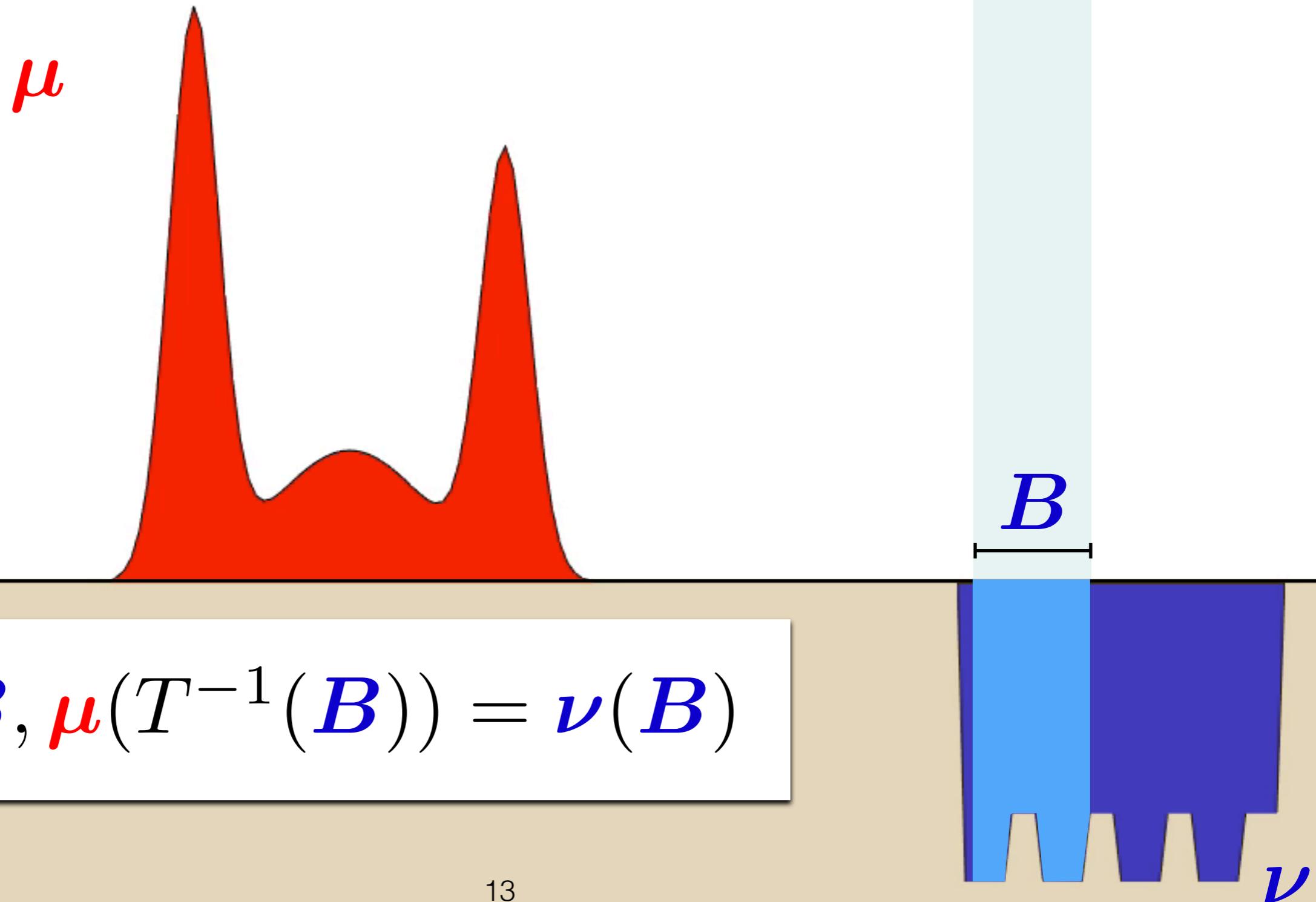
Origins: Monge's Problem

T must map red to blue.



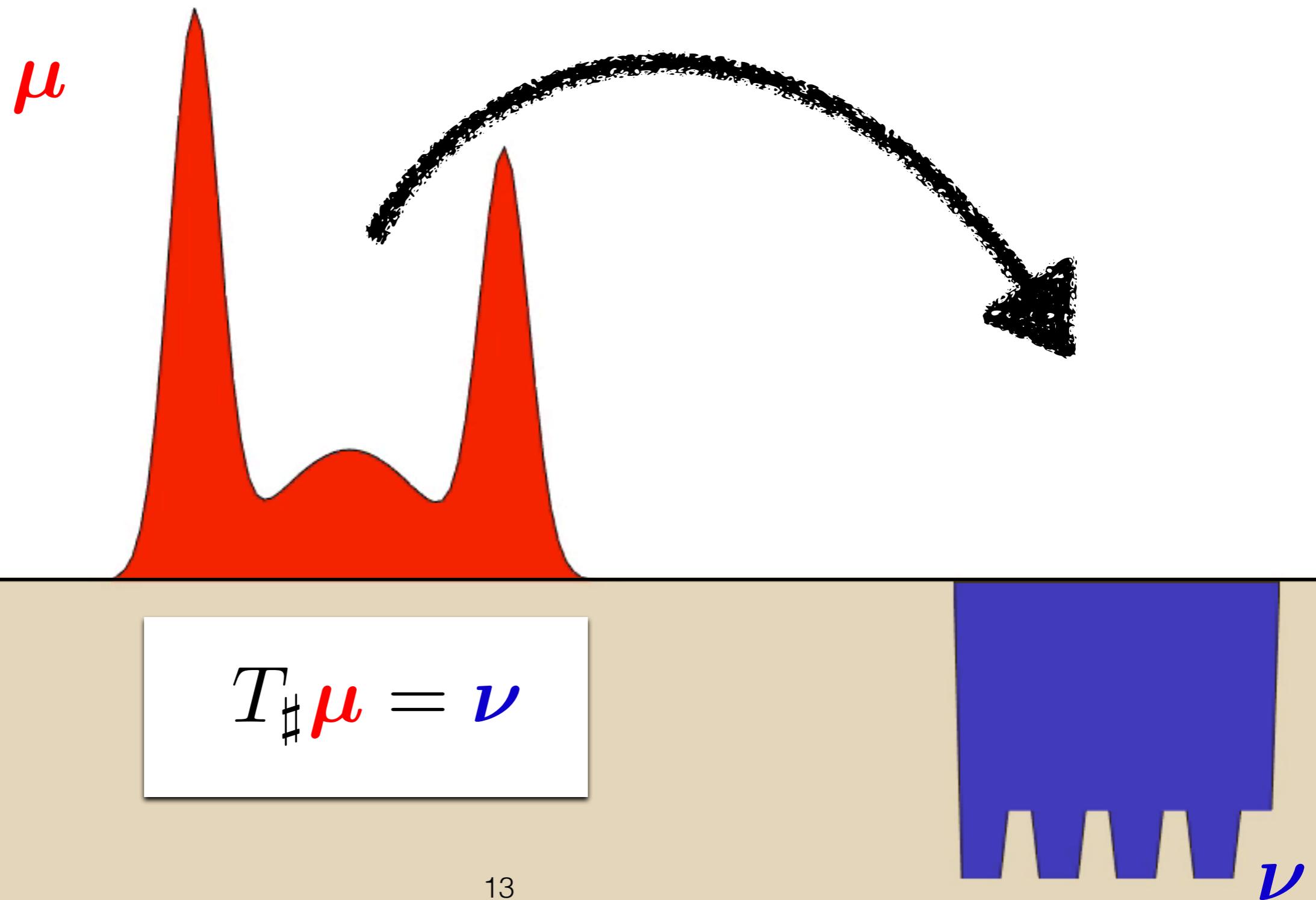
Origins: Monge's Problem

T must map red to blue.



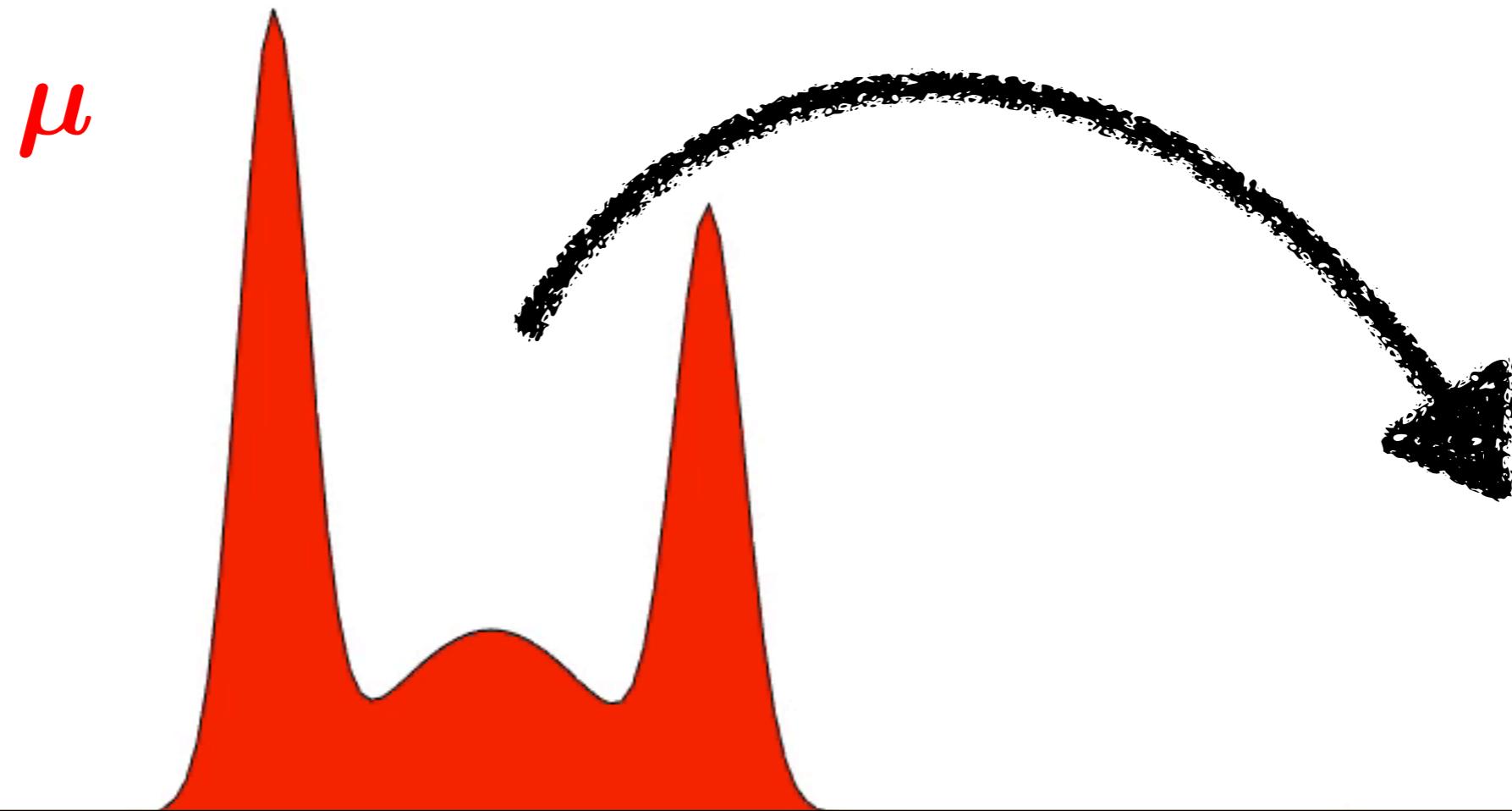
Origins: Monge's Problem

T must push-forward the red measure towards the blue



Origins: Monge's Problem

T must push-forward the red measure towards the blue



What T s.t. $T_{\sharp}\mu = \nu$
minimizes $\int D(x, T(x))\mu(dx)$?

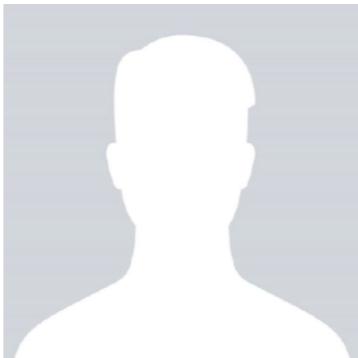
Kantorovich Problem



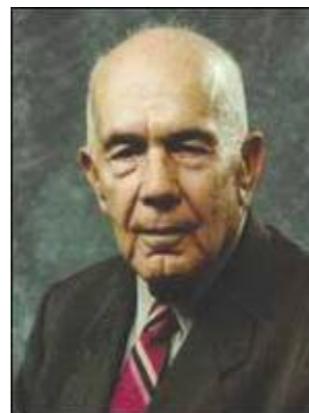
Kantorovich



1939



Tolstoi
1930



Hitchcock

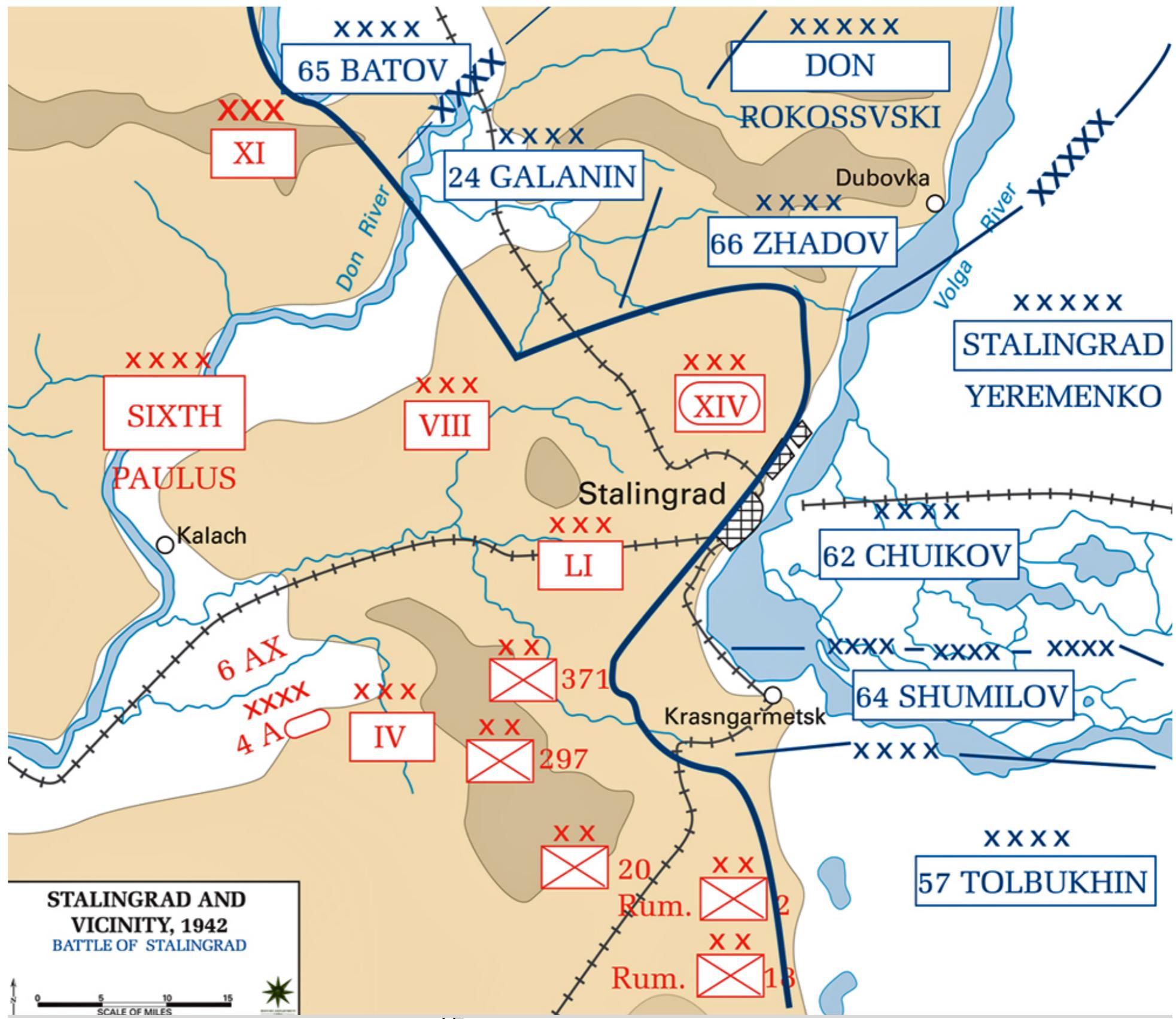
THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

BY FRANK L. HITCHCOCK

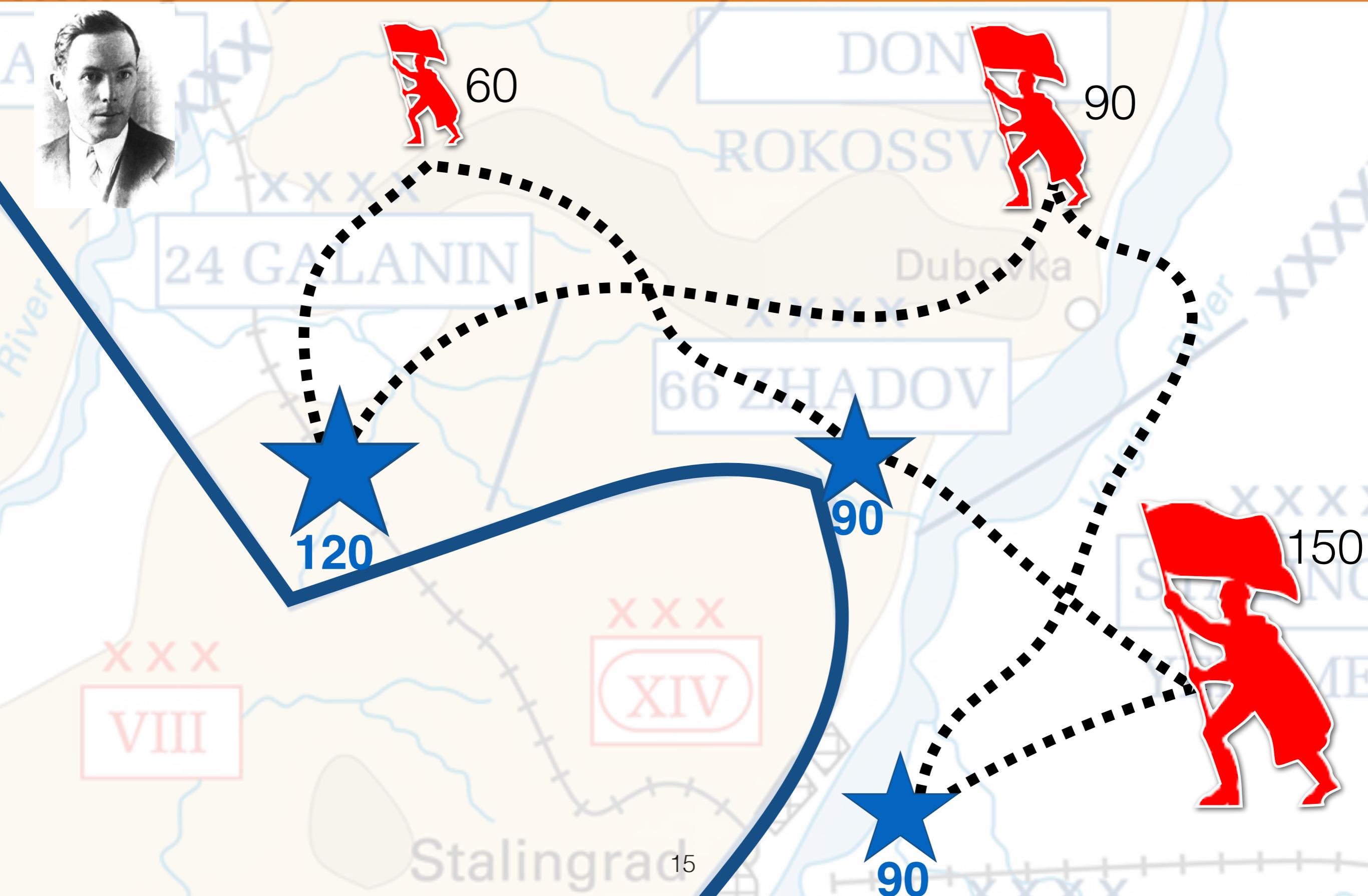
1. Statement of the problem. When several factories supply a product to a number of cities we desire the least costly manner of distribution. Due to freight rates and other matters the cost of a ton of product to a particular city will vary according to which factory supplies it, and will also vary from city to city.

1941

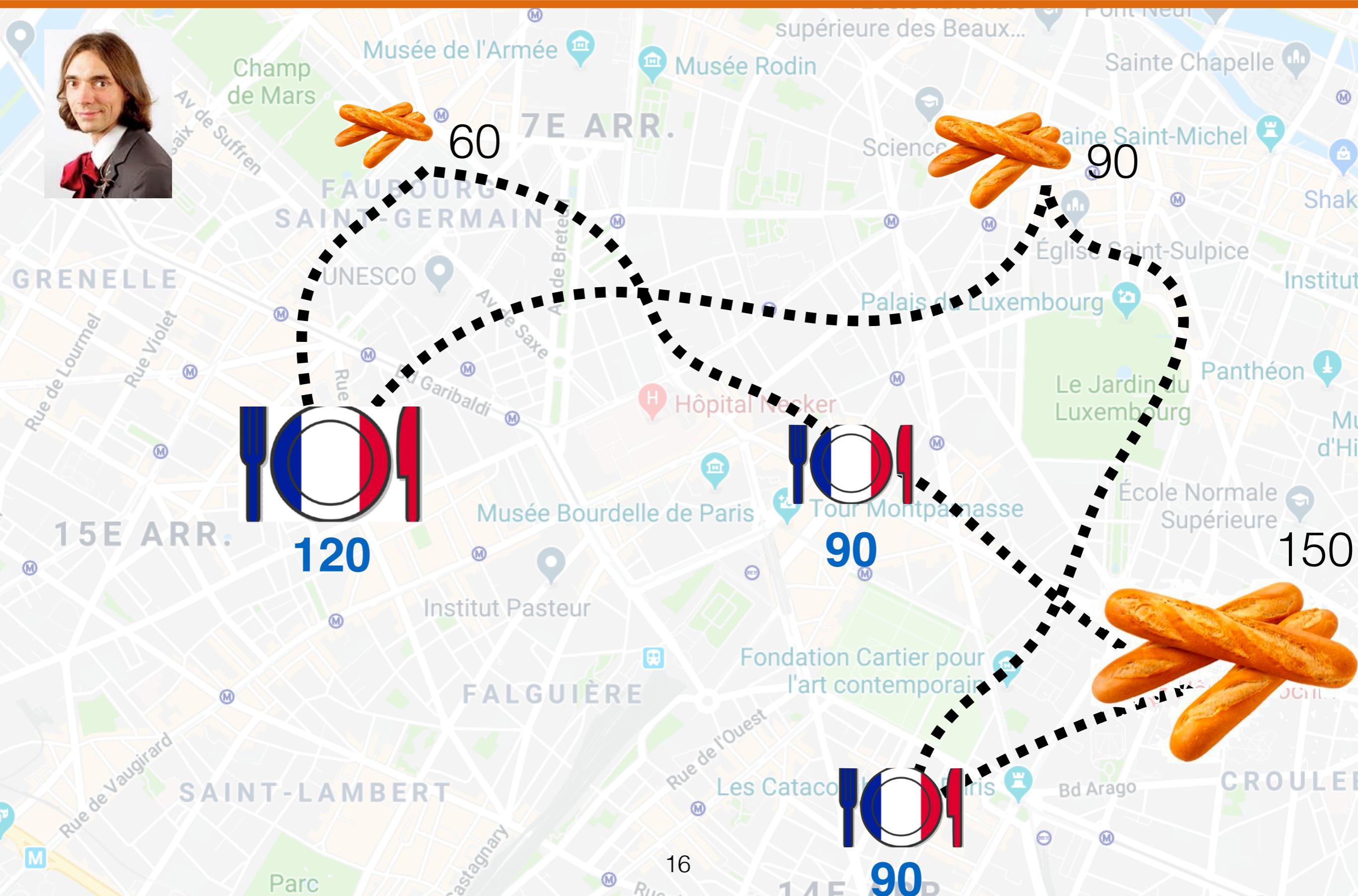
Kantorovich Problem



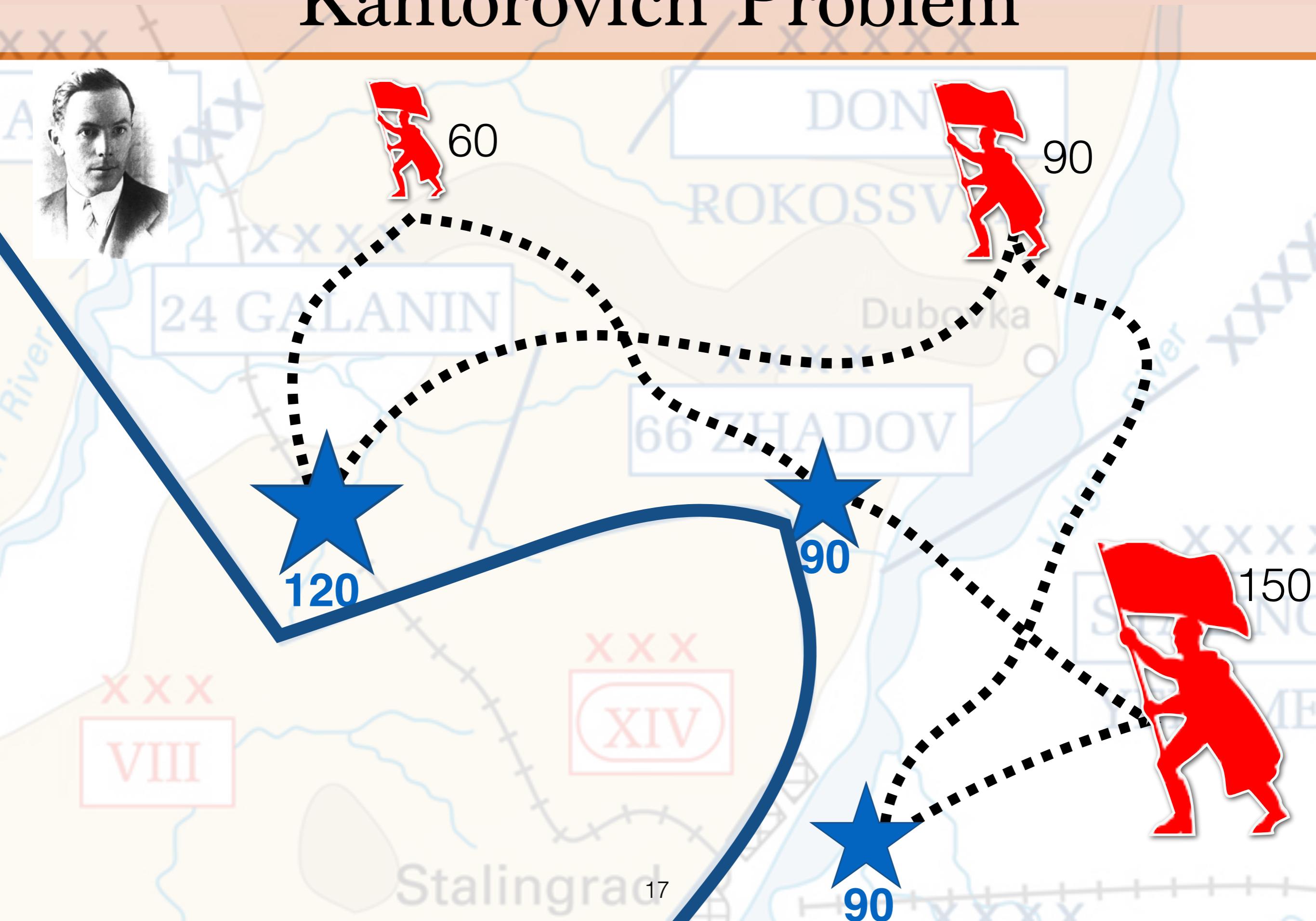
Kantorovich Problem



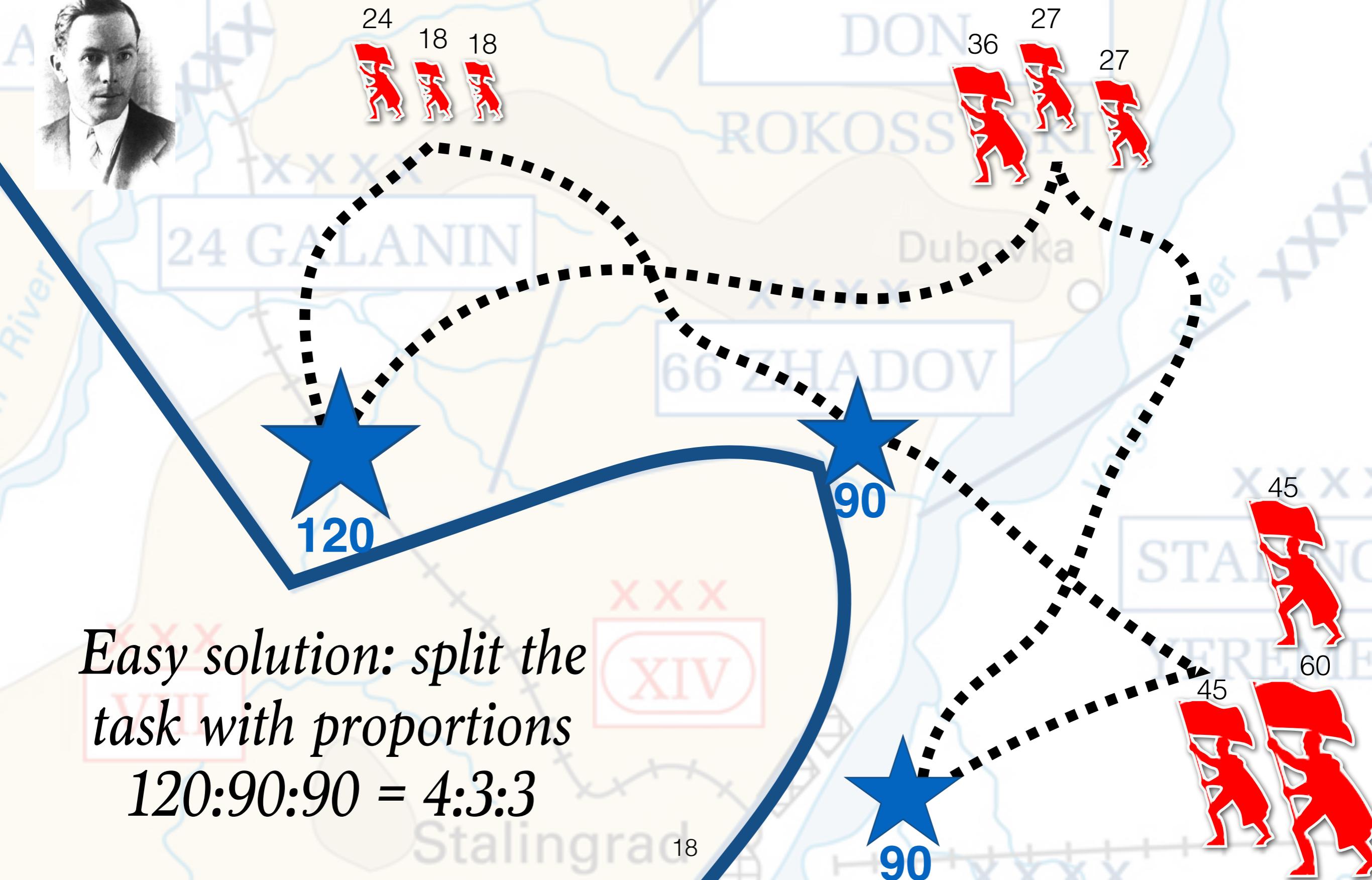
Kantorovich Problem à la française



Kantorovich Problem



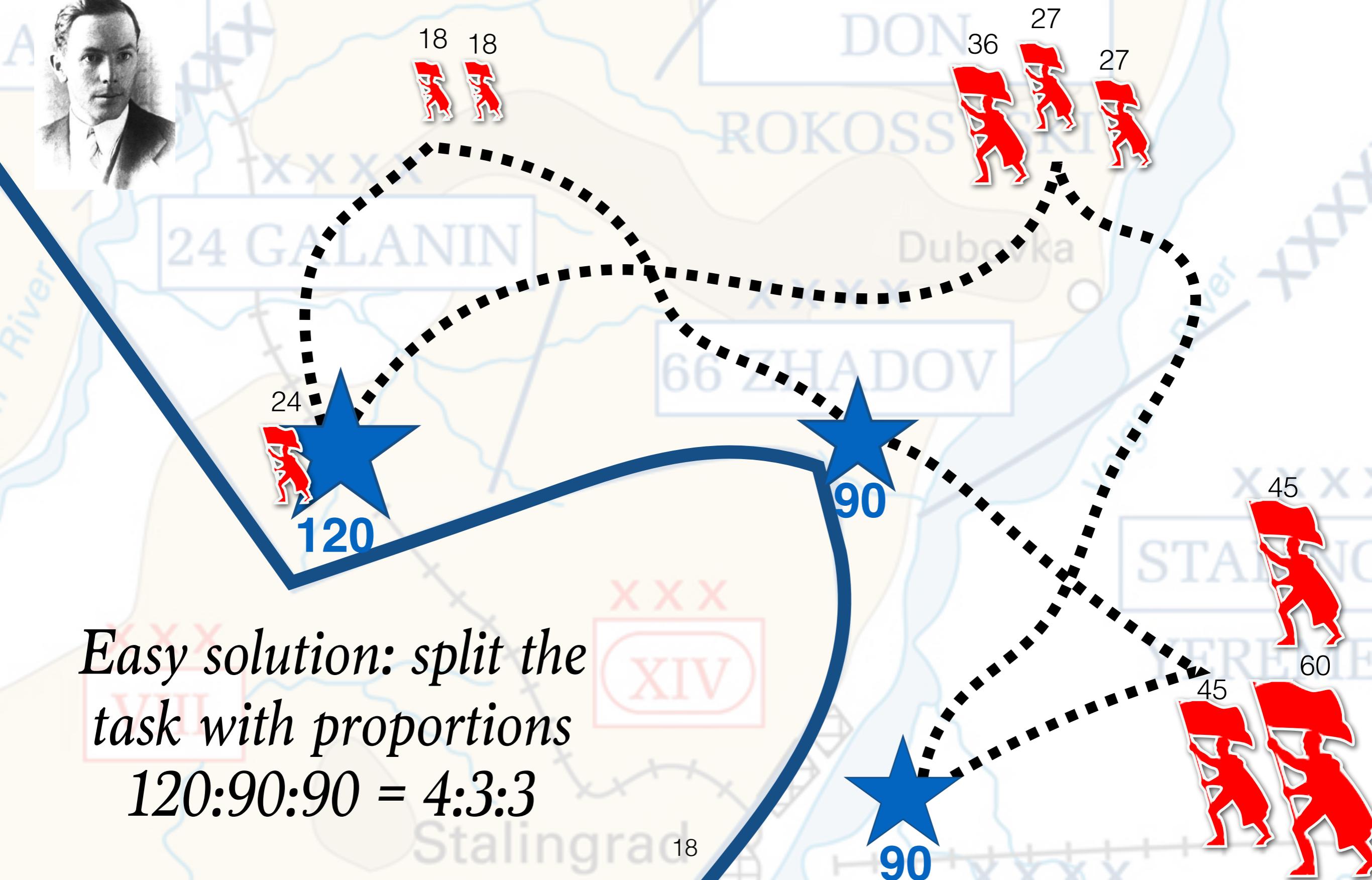
Kantorovich Problem



Easy solution: split the task with proportions

$$120:90:90 = 4:3:3$$

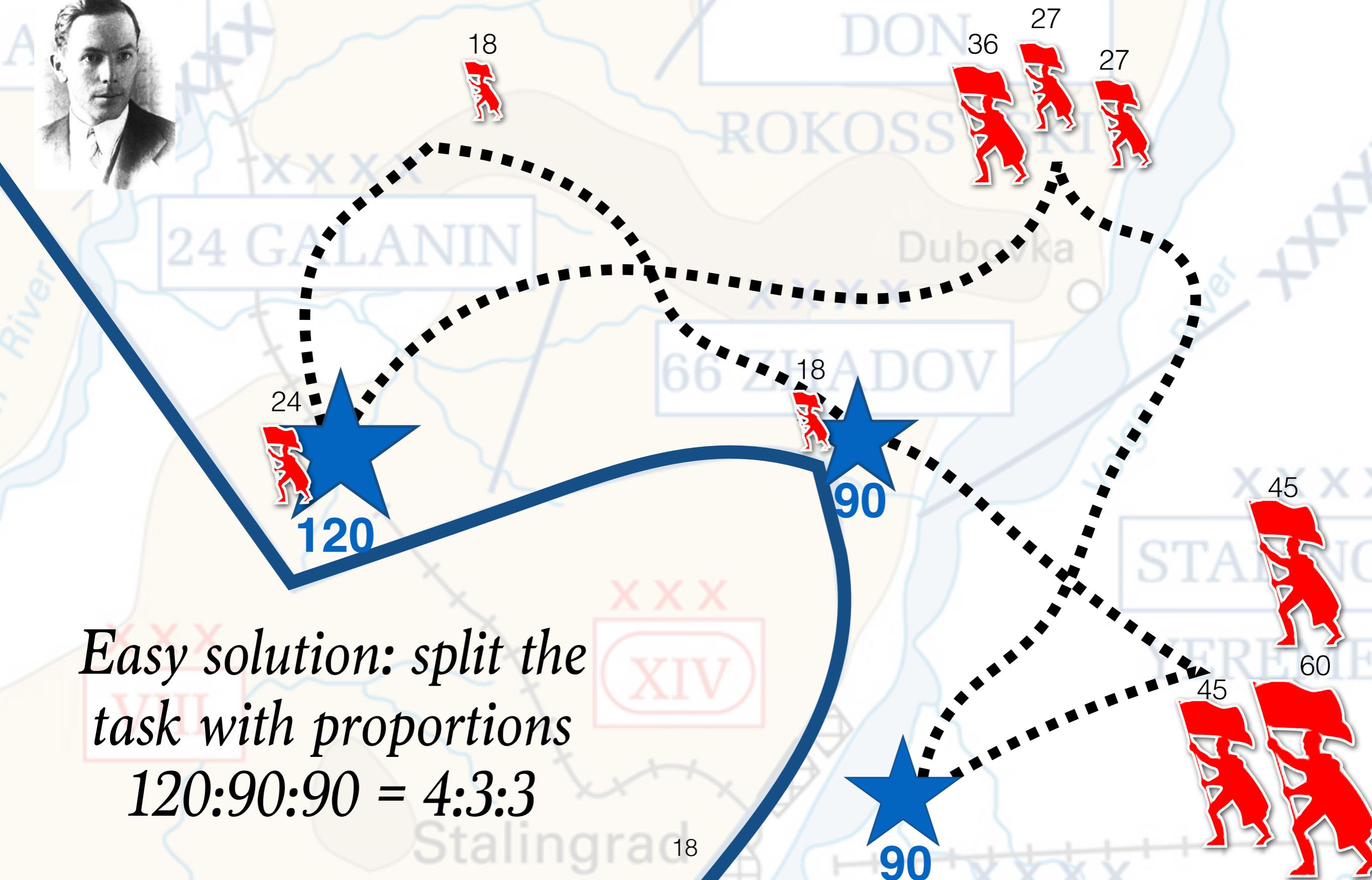
Kantorovich Problem



Easy solution: split the task with proportions

$$120:90:90 = 4:3:3$$

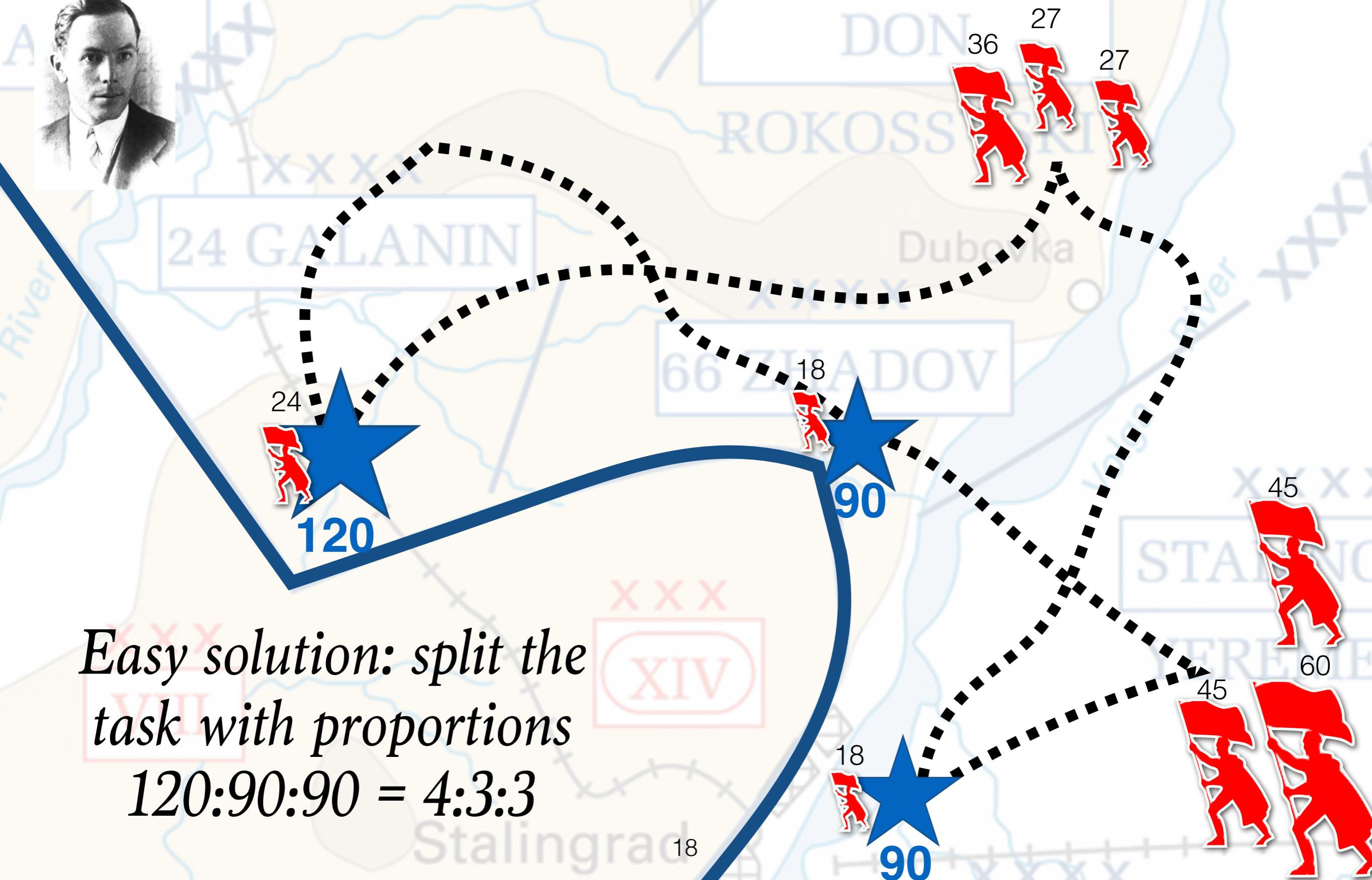
Kantorovich Problem



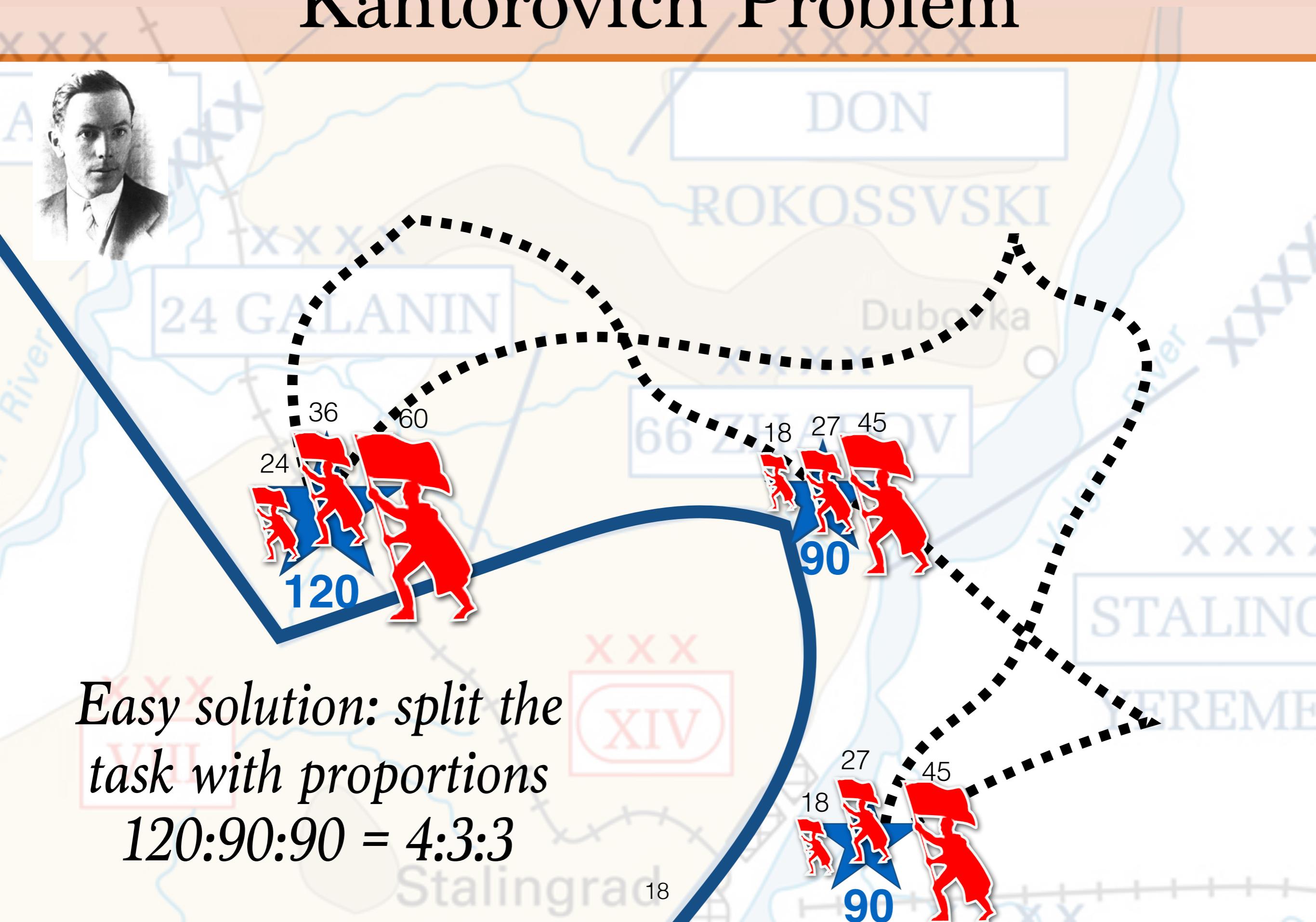
Easy solution: split the task with proportions

$$120:90:90 = 4:3:3$$

Kantorovich Problem



Kantorovich Problem



Easy solution: split the task with proportions

$$120:90:90 = 4:3:3$$

Kantorovich Problem



Naive approach results in
many displacements...

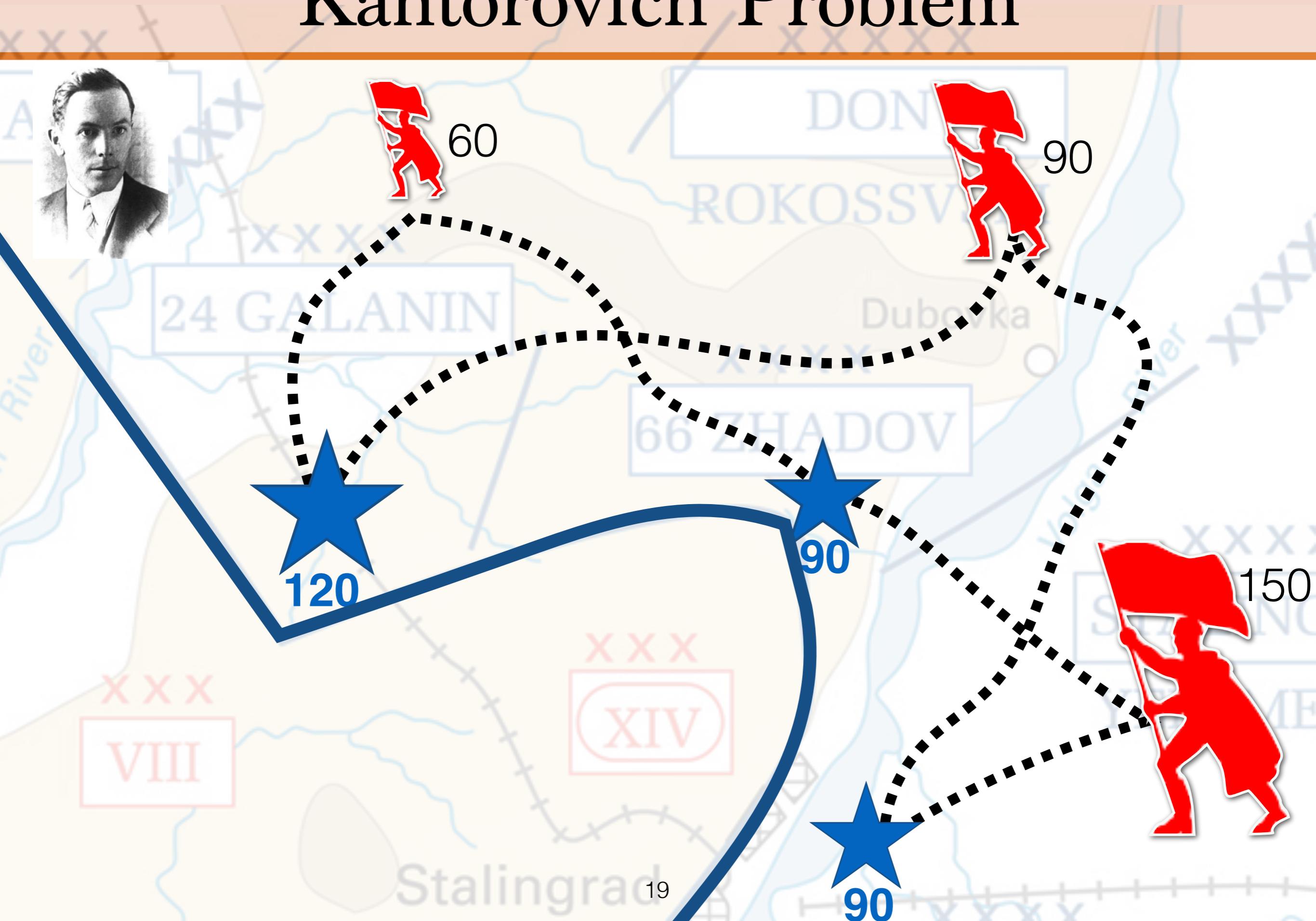
Can we find a cheaper
alternative?

*Easy solution: split the
task with proportions*

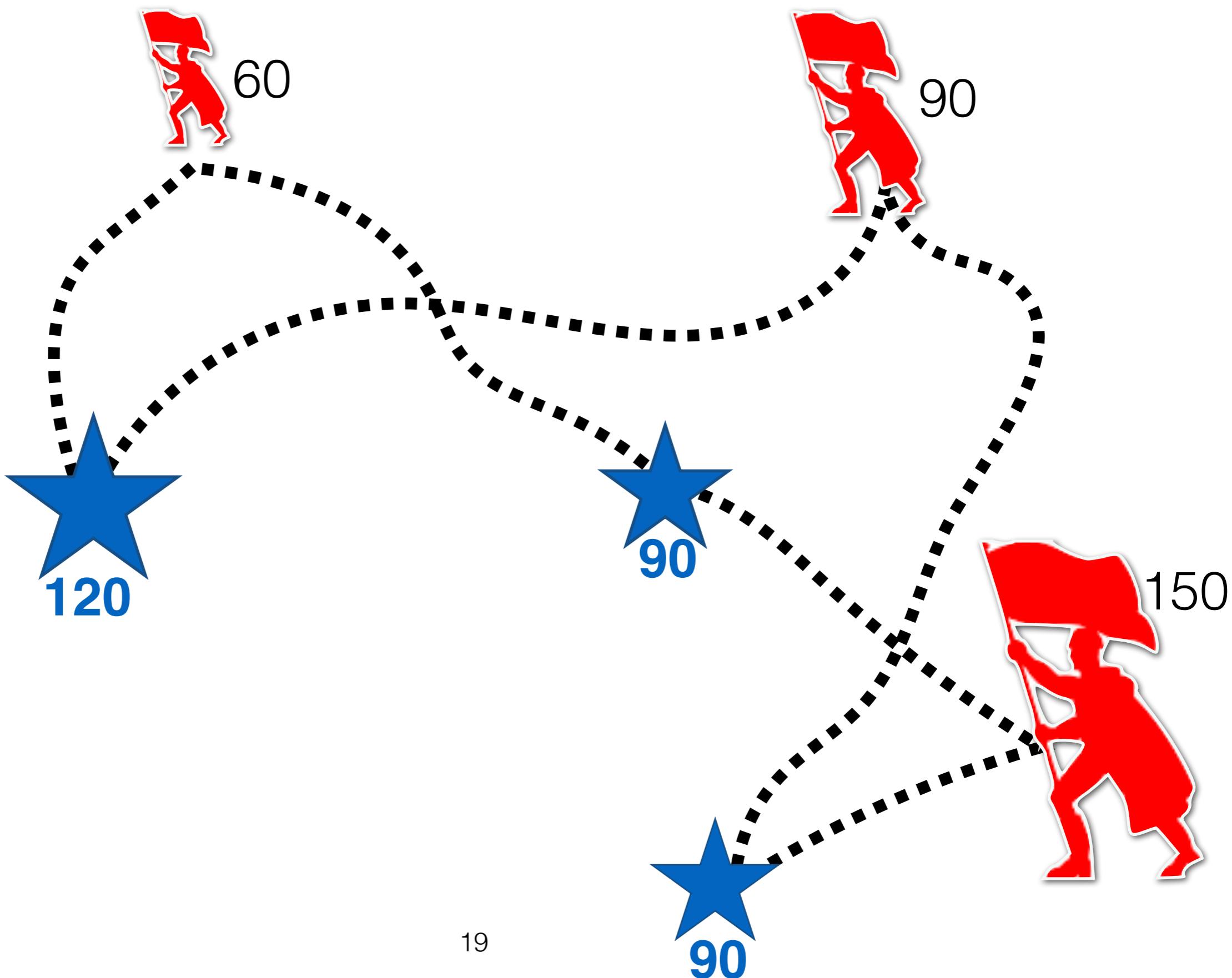
$$120:90:90 = 4:3:3$$



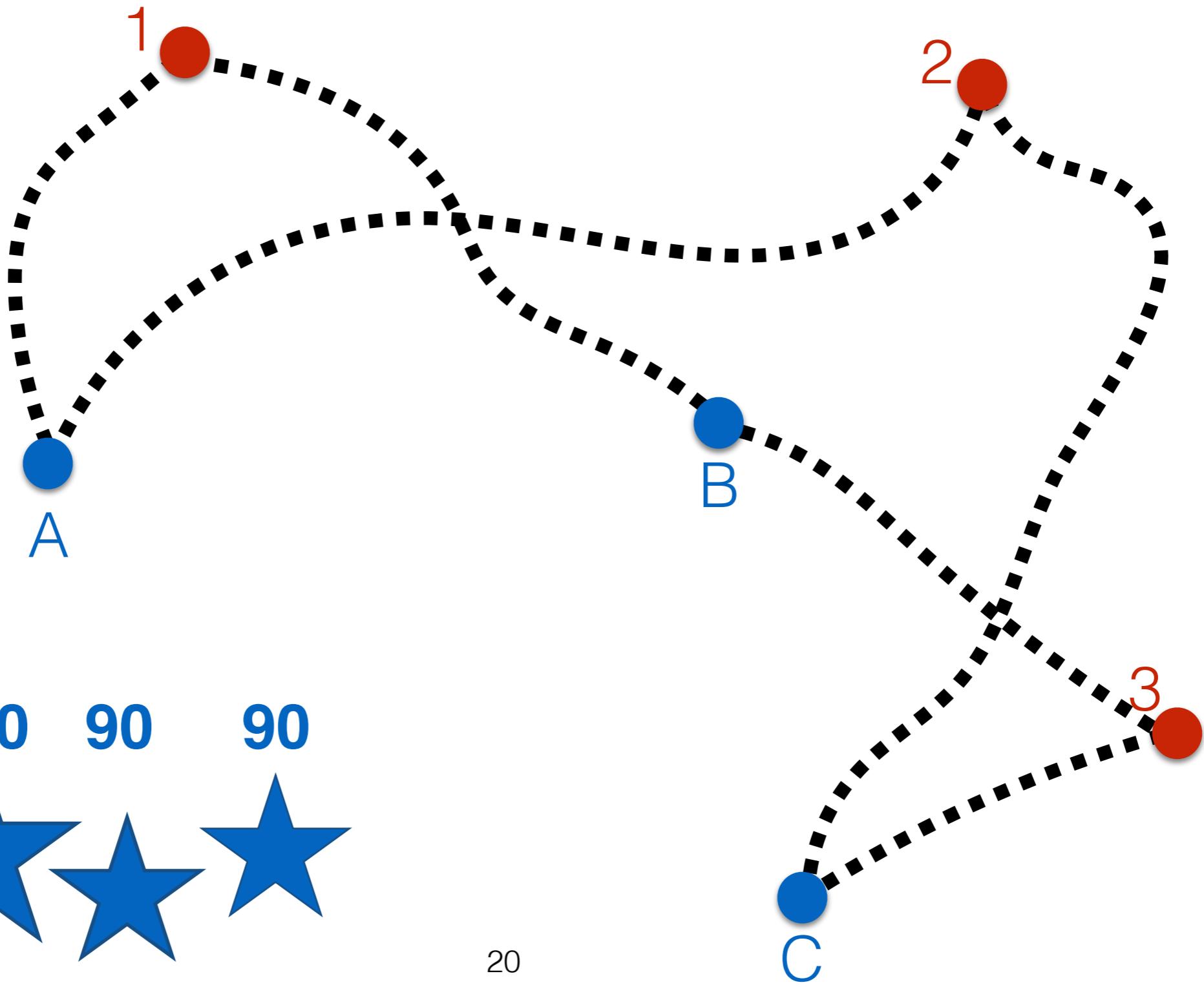
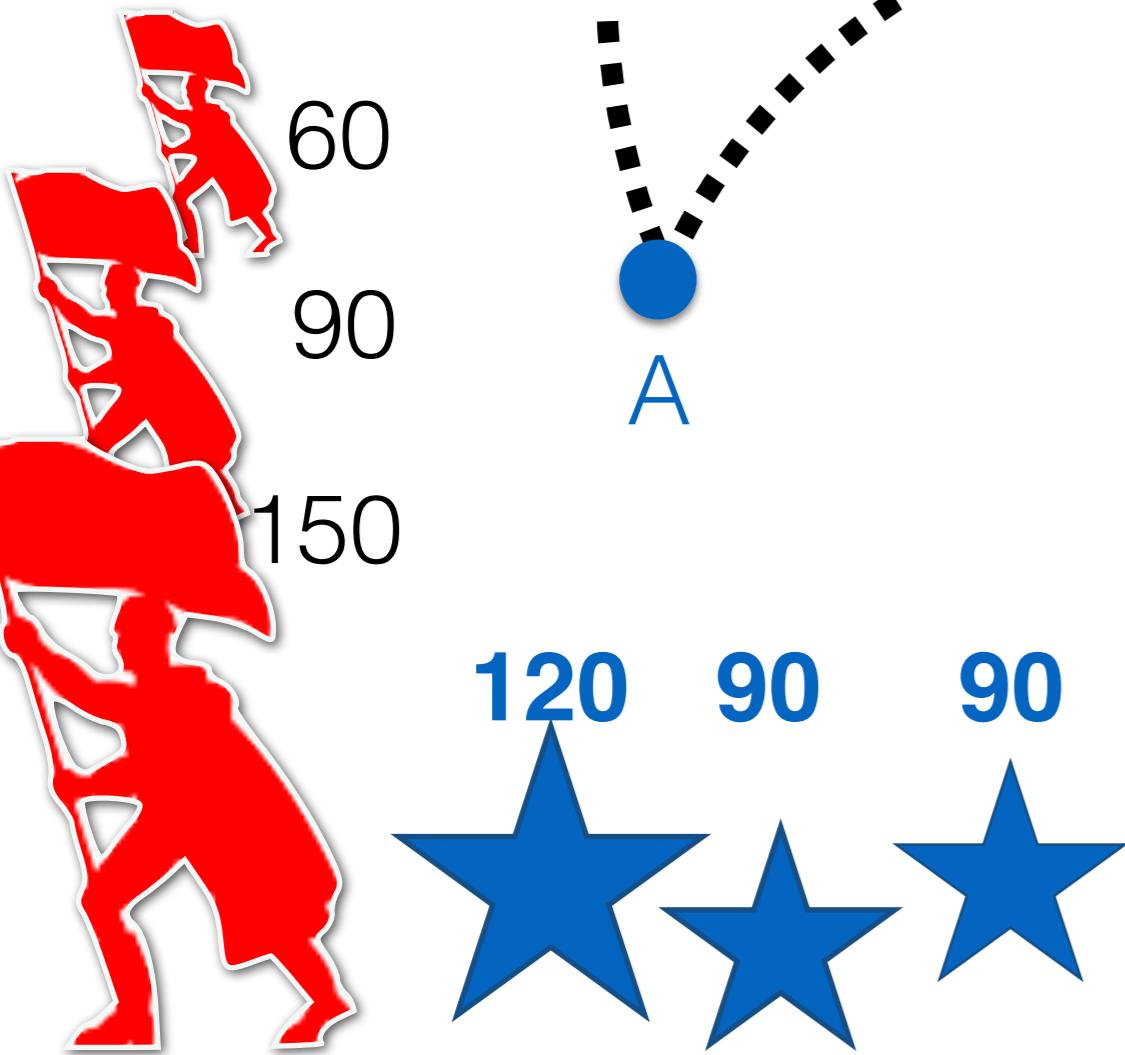
Kantorovich Problem



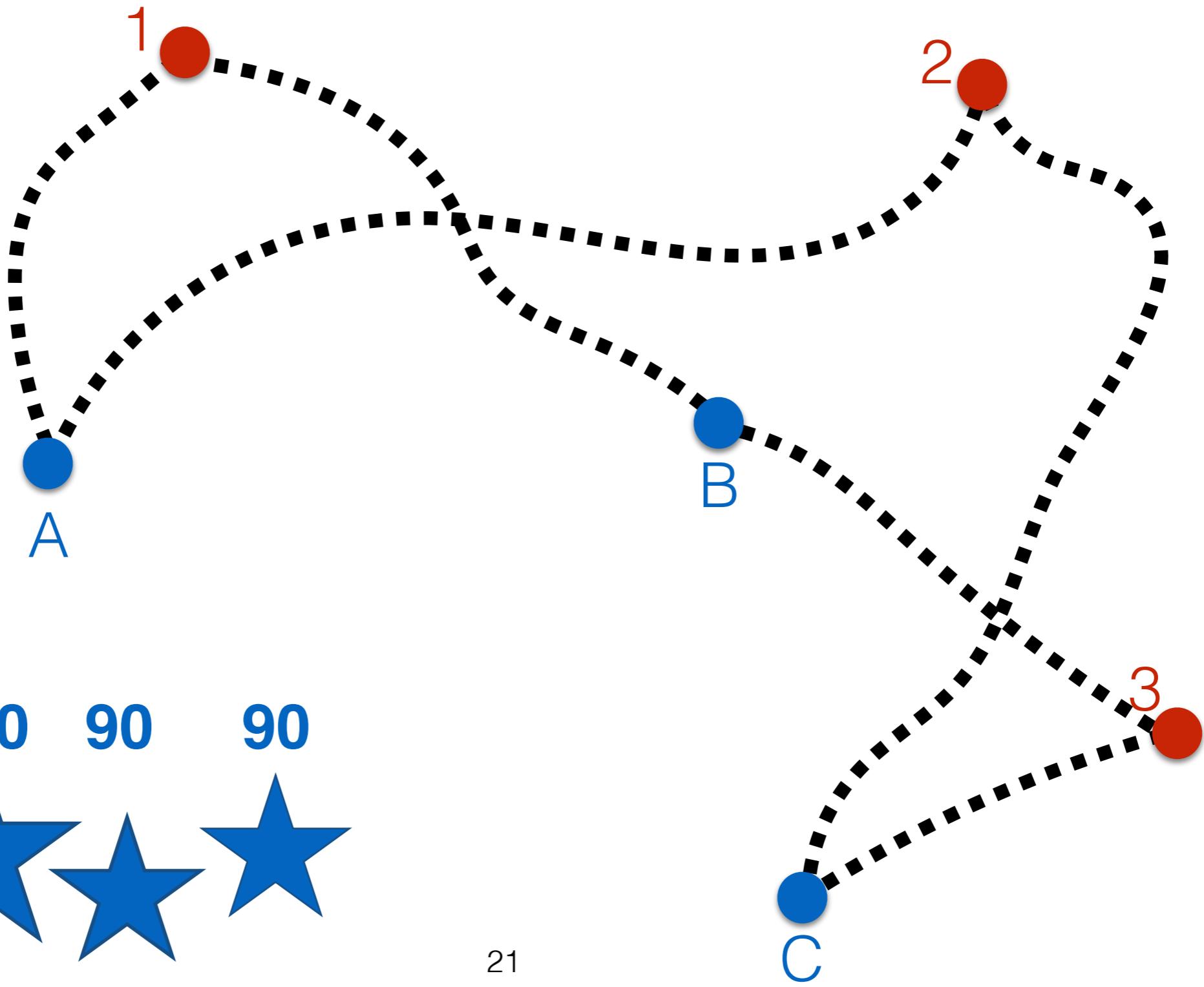
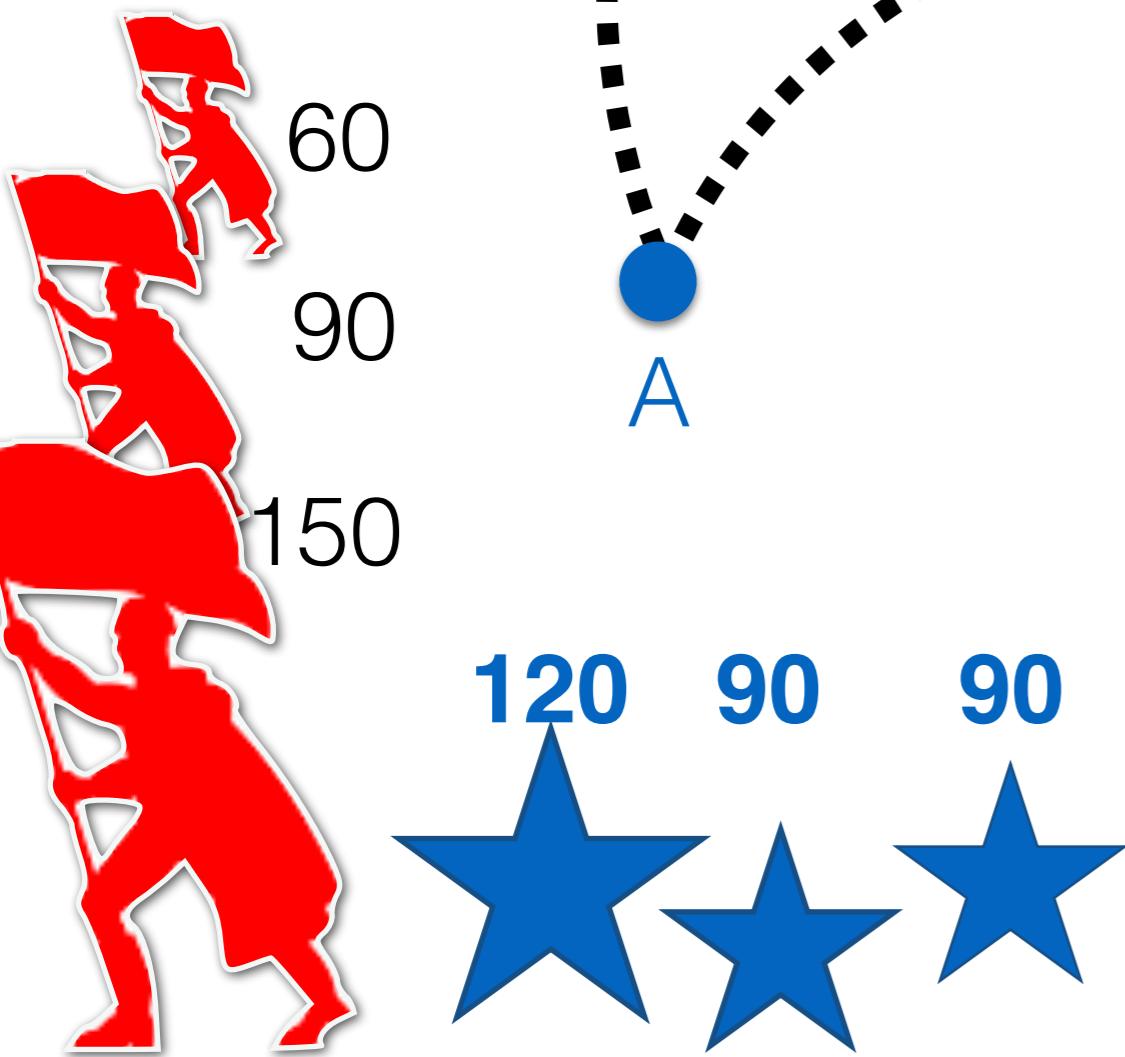
Kantorovich Problem



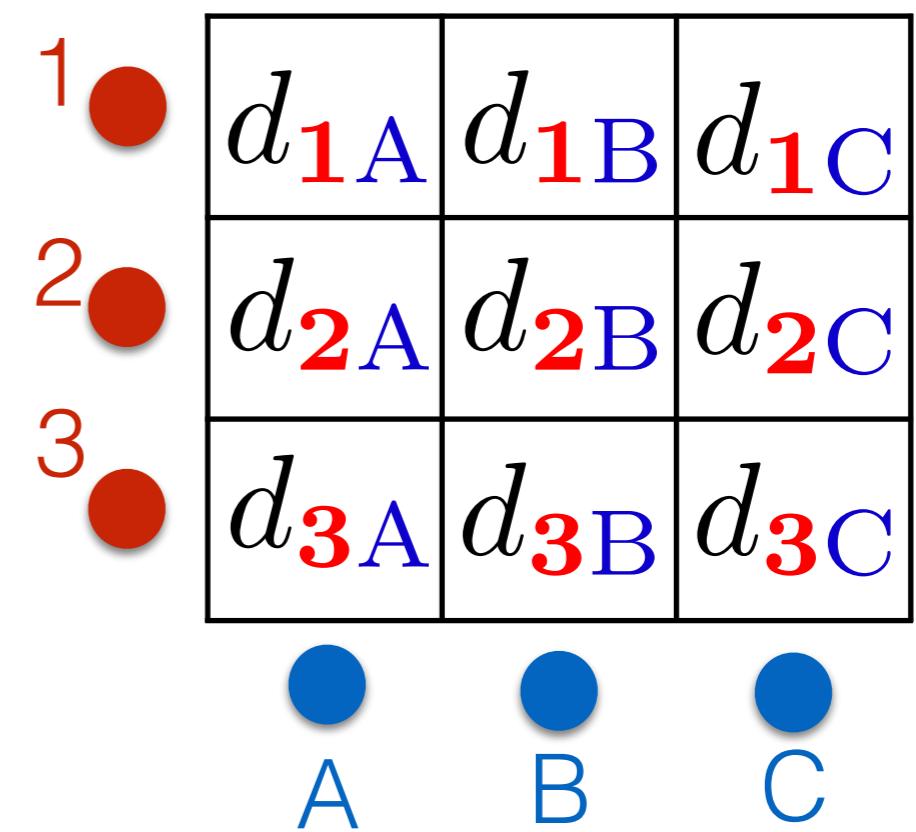
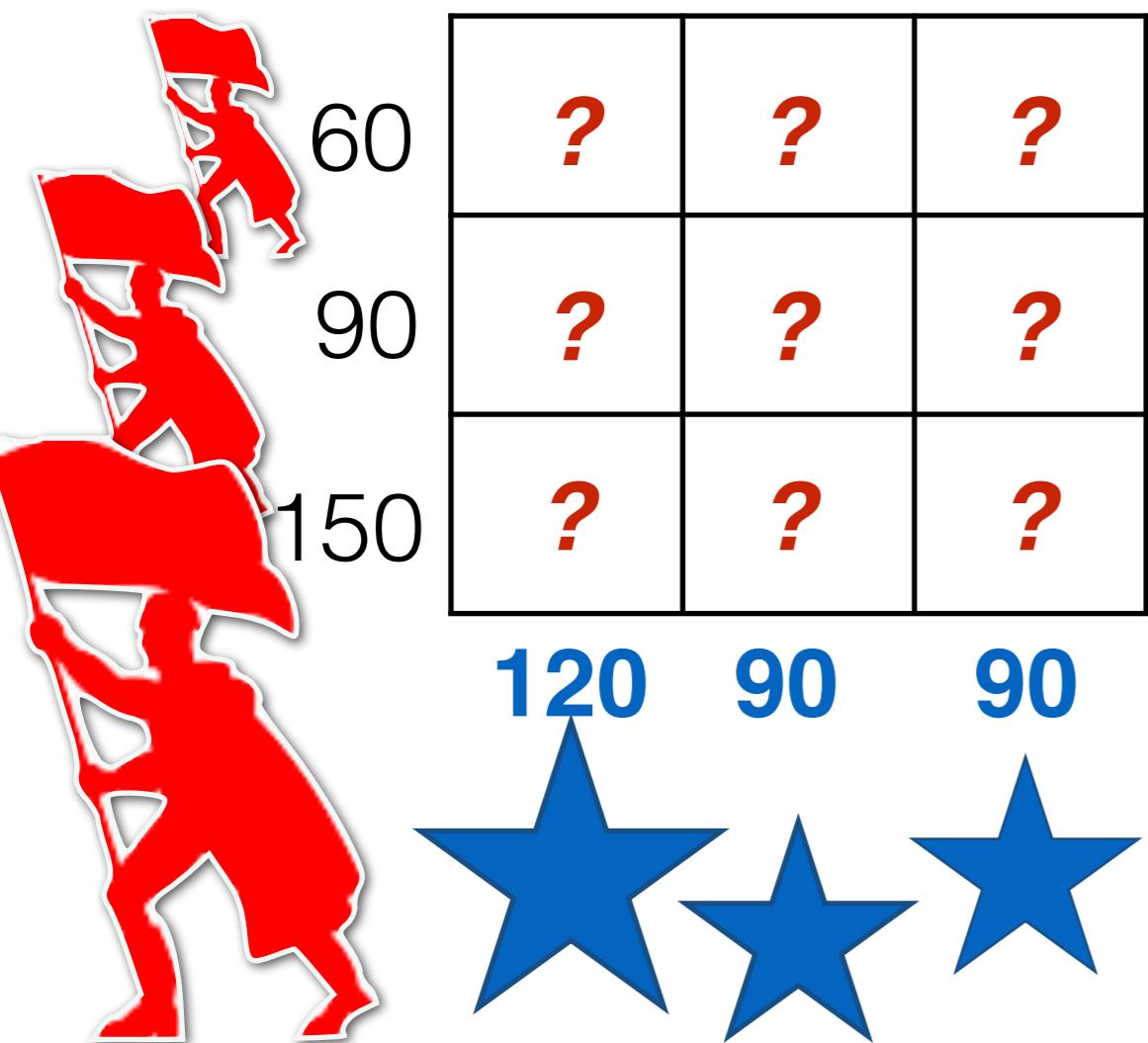
Kantorovich Problem



Kantorovich Problem



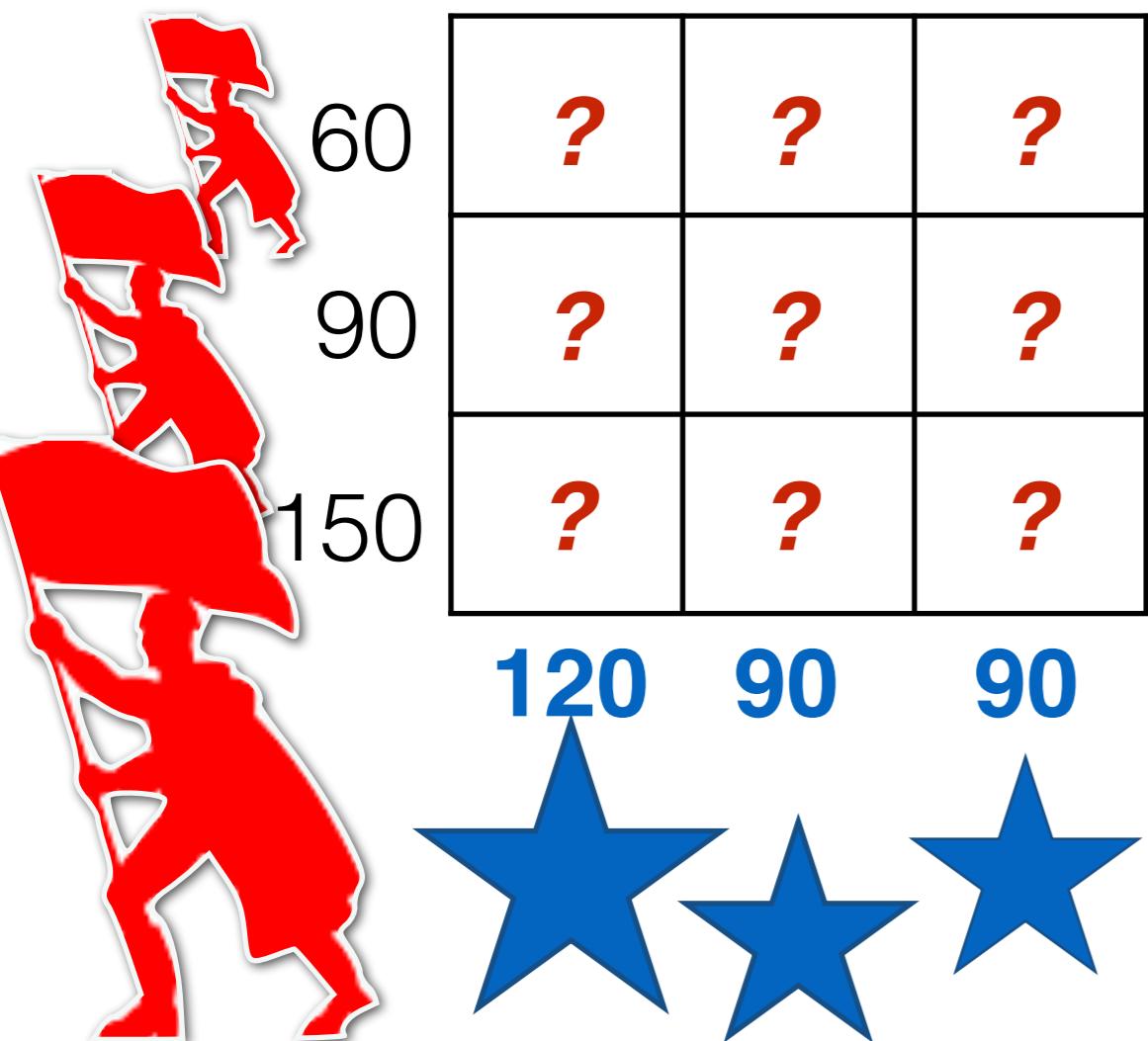
Kantorovich Problem



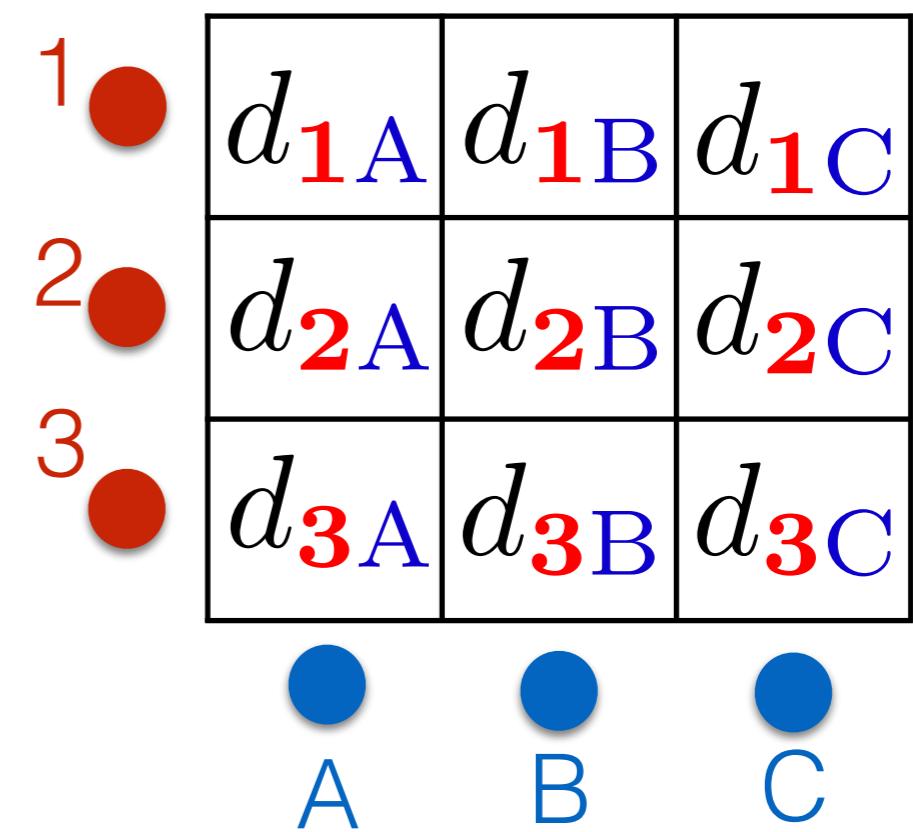
Kantorovich Problem



Transportation matrix



Distance matrix

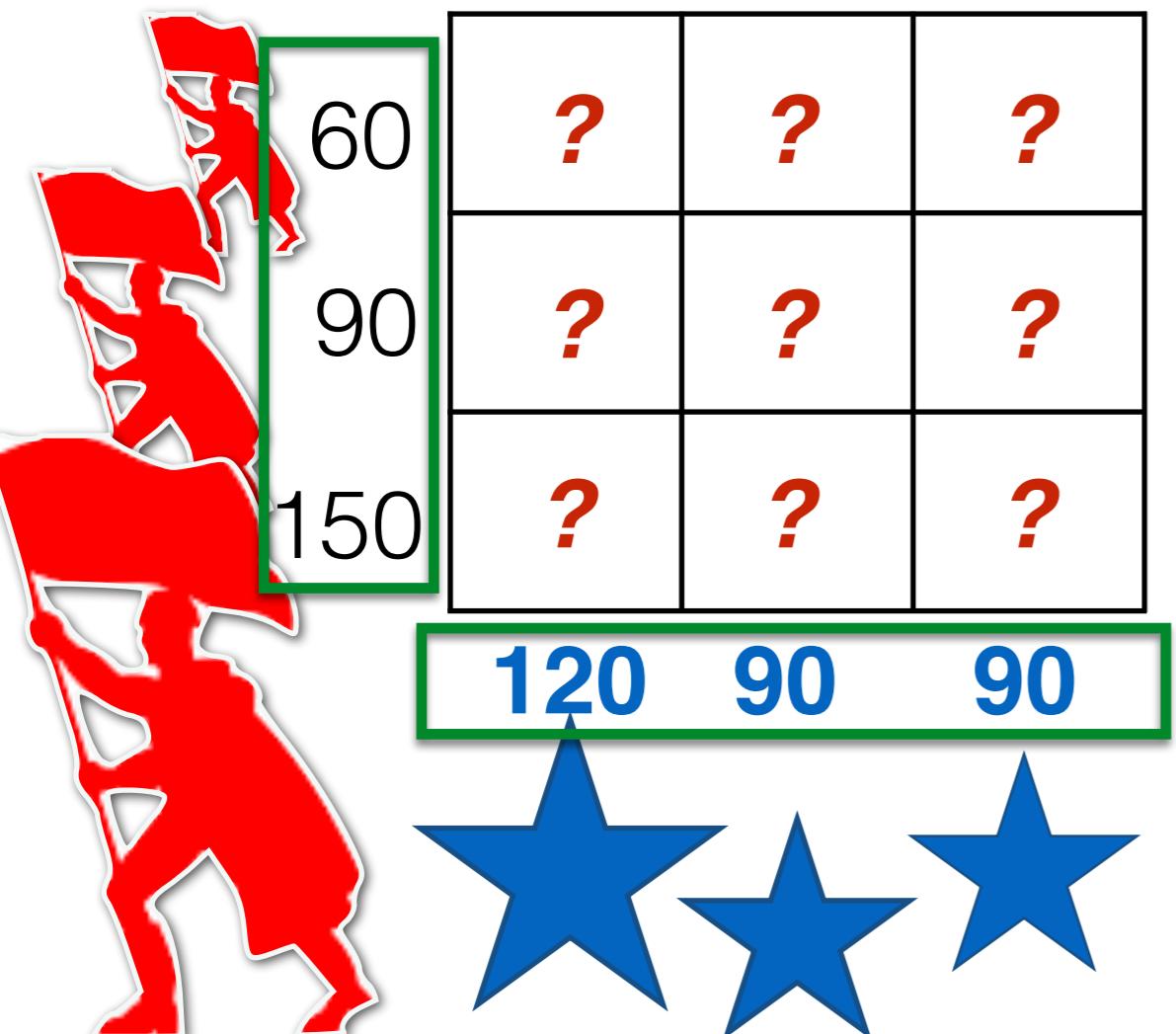


Kantorovich Problem



The problem is entirely described by counts and a cost/distance matrix

Transportation matrix



60	?	?	?
90	?	?	?
150	?	?	?
	120	90	90

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}

A diagram showing three red circles labeled 1, 2, and 3, each connected by a line to a blue circle labeled A, B, or C respectively. The circles are arranged vertically on the left, and the corresponding letters are arranged horizontally at the bottom.

Kantorovich Problem

Transportation matrix

60	?	?	?
90	?	?	?
150	?	?	?
	120	90	90

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Kantorovich Problem

Transportation matrix

60	p_{1A}	p_{1B}	p_{1C}
90	p_{2A}	p_{2B}	p_{2C}
150	p_{3A}	p_{3B}	p_{3C}
	120	90	90

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Kantorovich Problem

Transportation matrix

a_1	p_{1A}	p_{1B}	p_{1C}
a_2	p_{2A}	p_{2B}	p_{2C}
a_3	p_{3A}	p_{3B}	p_{3C}
	b_A	b_B	b_C

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}

A B C

Kantorovich Problem

Transportation matrix

a_1	p_{1A}	p_{1B}	p_{1C}
a_2	p_{2A}	p_{2B}	p_{2C}
a_3	p_{3A}	p_{3B}	p_{3C}
	b_A	b_B	b_C

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Constraints

$$\forall i \in \{1, 2, 3\}, \quad \sum_{j \in \{A, B, C\}} p_{ij} = a_i$$

$$\forall j \in \{A, B, C\}, \quad \sum_{i \in \{1, 2, 3\}} p_{ij} = b_j$$

$$p_{ij} \geq 0$$

Kantorovich Problem

Transportation matrix

a_1	p_{1A}	p_{1B}	p_{1C}
a_2	p_{2A}	p_{2B}	p_{2C}
a_3	p_{3A}	p_{3B}	p_{3C}

b_A b_B b_C

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}

A B C

Constraints

$$\forall i \in \{1, 2, 3\}, \quad \sum_{j \in \{A, B, C\}} p_{ij} = a_i$$

$$\forall j \in \{A, B, C\}, \quad \sum_{i \in \{1, 2, 3\}} p_{ij} = b_j$$

$$p_{ij} \geq 0$$

Cost function

$$C(\mathbf{P}) = \sum_{j \in \{A, B, C\}} \sum_{i \in \{1, 2, 3\}} p_{ij} d_{ij}$$

Kantorovich Problem

Transportation matrix

a_1	p_{1A}	p_{1B}	p_{1C}
a_2	p_{2A}	p_{2B}	p_{2C}
a_3	p_{3A}	p_{3B}	p_{3C}

b_A b_B b_C

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}

A B C

Constraints

$$\forall i \in \{1, 2, 3\}, \sum_{j \in \{A, B, C\}} p_{ij} = a_i$$

$$\forall j \in \{A, B, C\}, \sum_{i \in \{1, 2, 3\}} p_{ij} = b_j$$

$$p_{ij} \geq 0$$

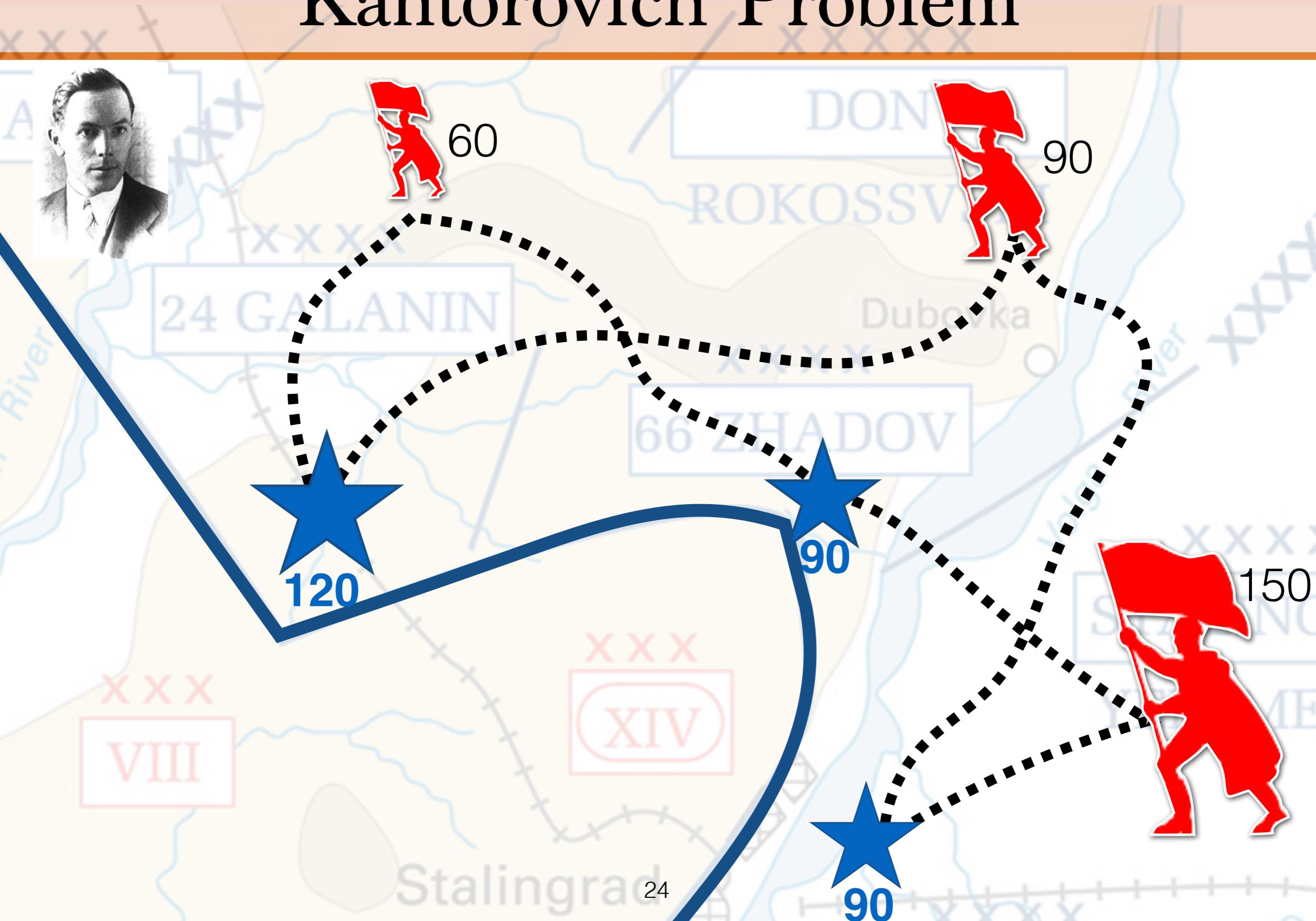
Cost function

$$C(\mathbf{P}) = \sum_{j \in \{A, B, C\}} \sum_{i \in \{1, 2, 3\}} p_{ij} d_{ij}$$

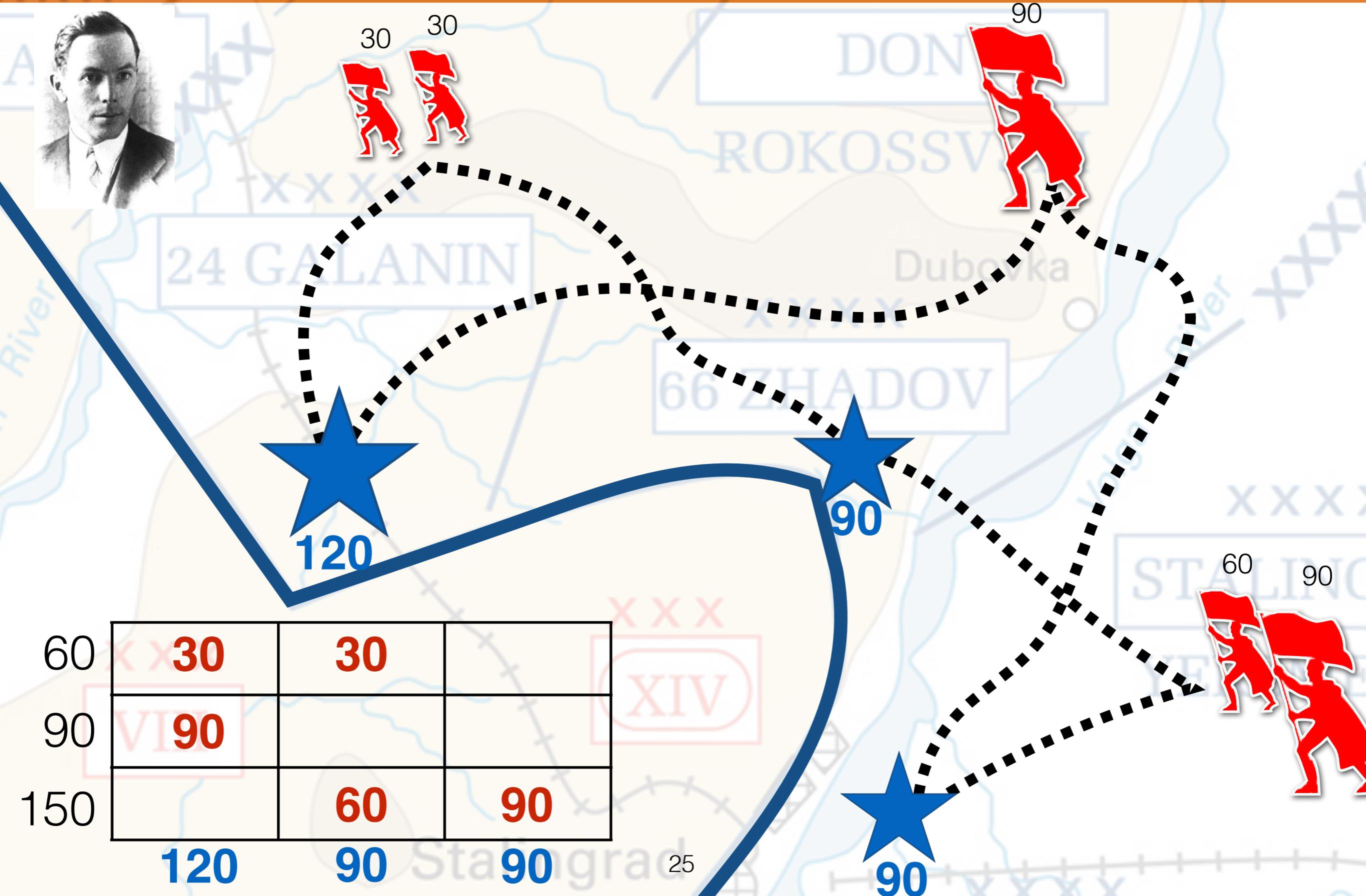
Problem

$$\min_{\text{all valid } \mathbf{P}} C(\mathbf{P})$$

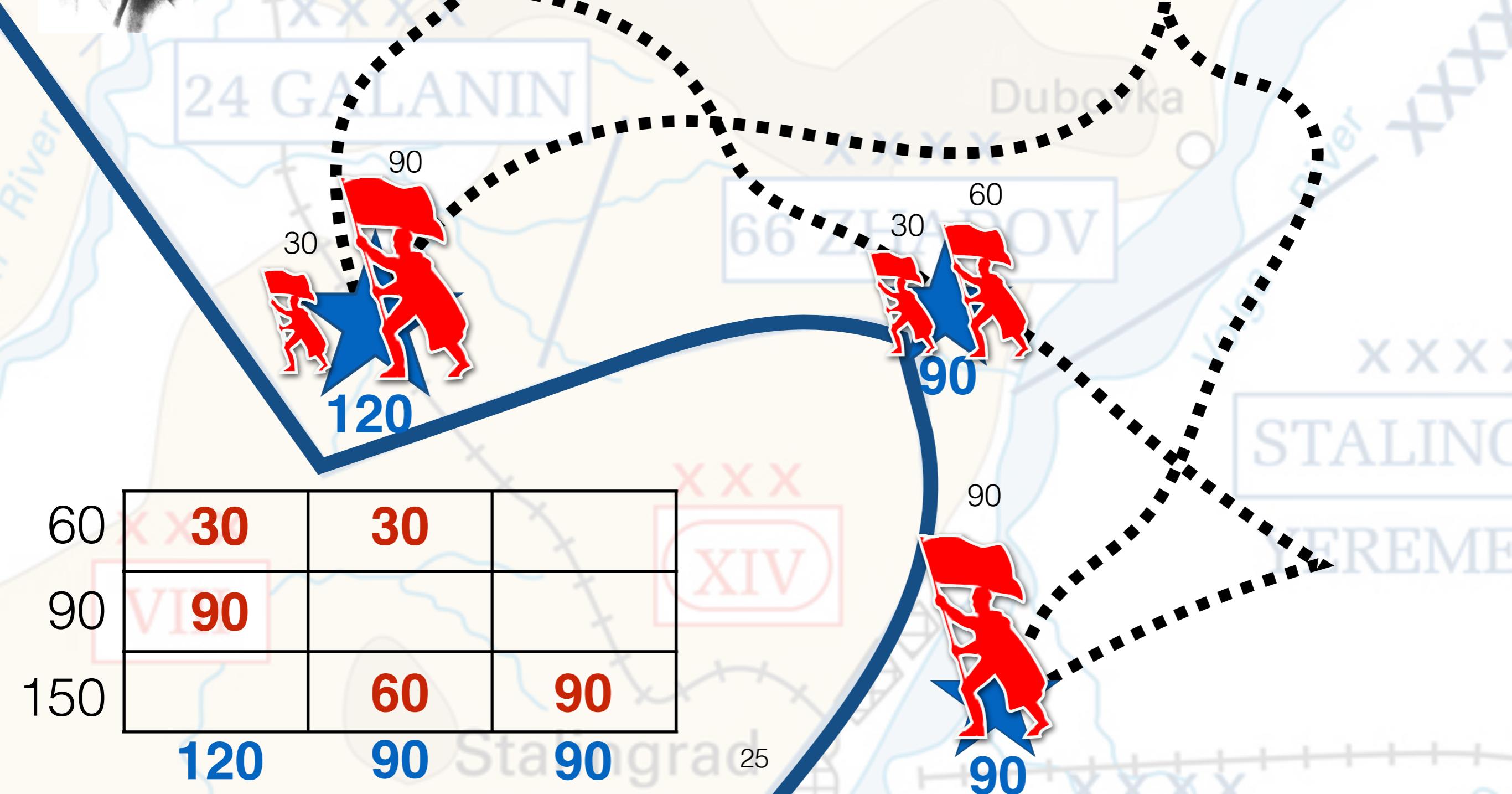
Kantorovich Problem



Kantorovich Problem

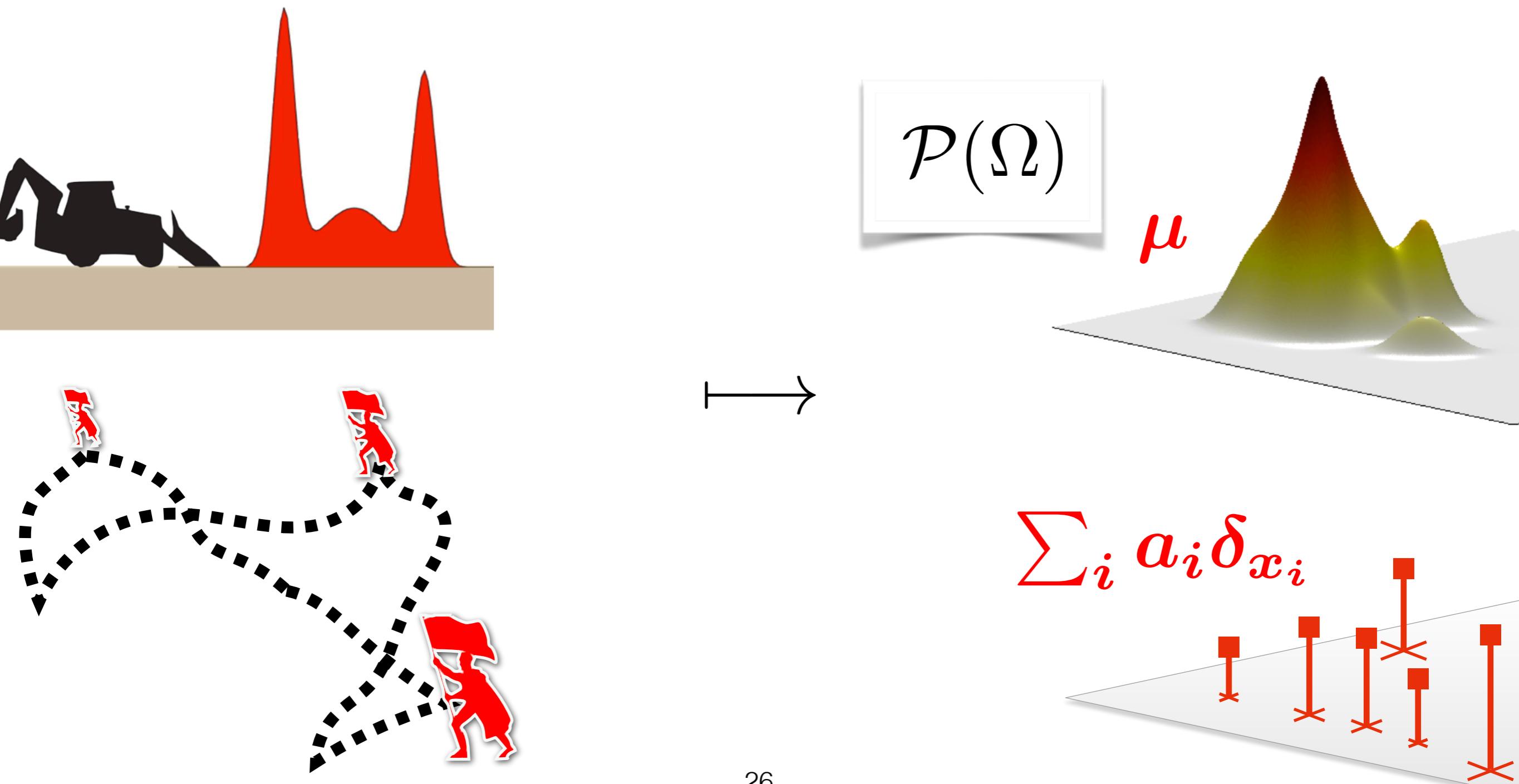


Kantorovich Problem



Mathematical Formalism

These problems involve discrete and continuous probability measures on a geometric space Ω

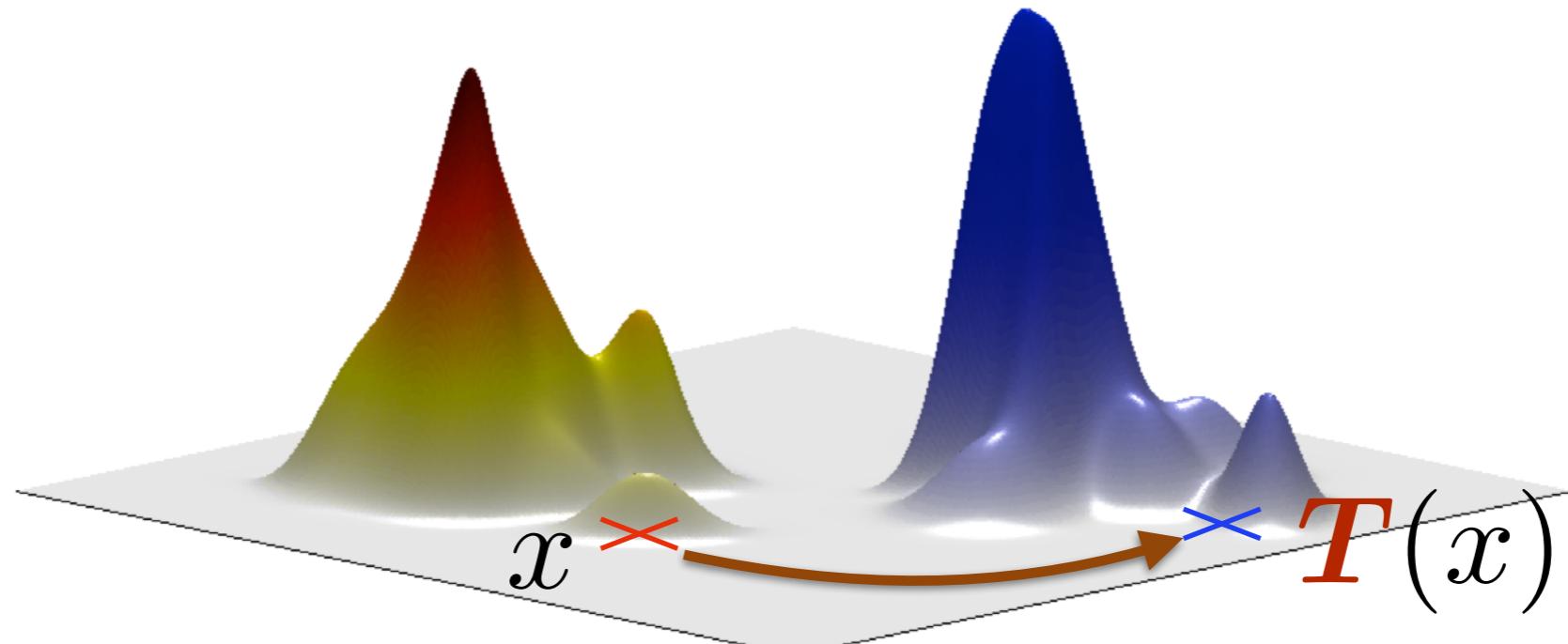


Monge Problem

Ω a measurable space, $c : \Omega \times \Omega \rightarrow \mathbb{R}$.
 μ, ν two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $T : \Omega \rightarrow \Omega$

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



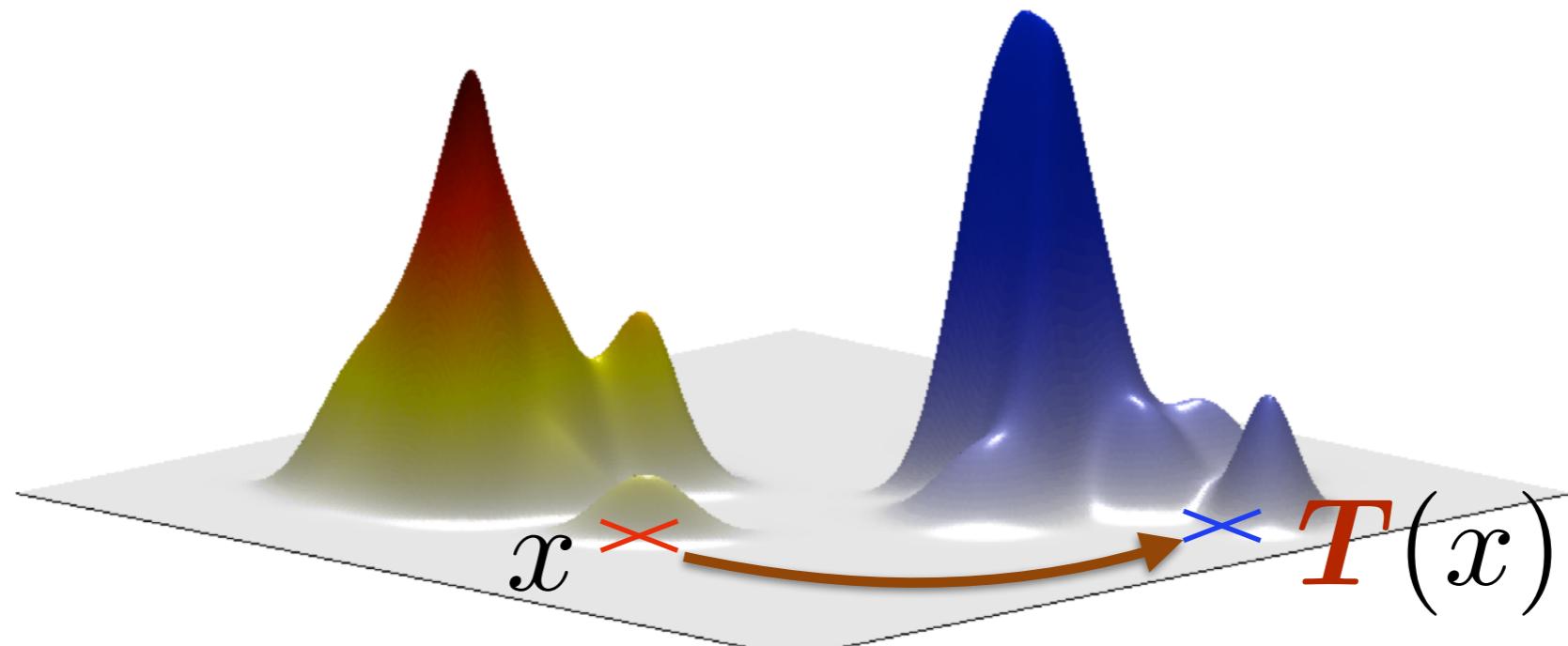
Monge Problem

Ω a measurable space, $c : \Omega \times \Omega \rightarrow \mathbb{R}$.
 μ, ν two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $T : \Omega \rightarrow \Omega$

[Brenier'87] If $\Omega = \mathbb{R}^d$, $c = \|\cdot - \cdot\|^2$,

μ, ν a.c., then $T = \nabla u$, u convex.



Monge Problem

Ω a measurable space, $c : \Omega \times \Omega \rightarrow \mathbb{R}$.
 μ, ν two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $T : \Omega \rightarrow \Omega$

[Brenier'87] If $\Omega = \mathbb{R}^d$, $c = \|\cdot - \cdot\|^2$,

μ, ν a.c., then $T = \nabla u$, u convex.

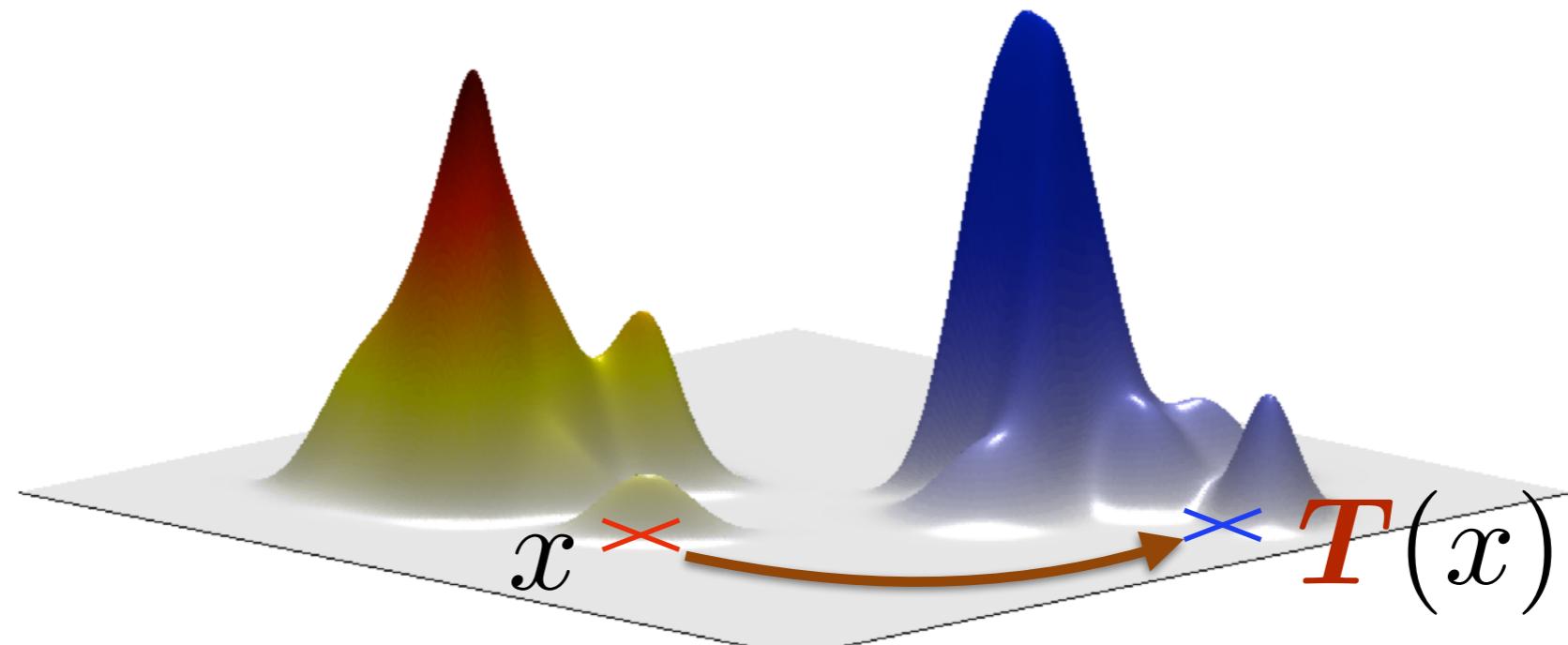
[Brenier'87]: For any u convex, ∇u is the OT
Monge map between μ and $\nabla u_\# \mu$.

Monge Problem

Ω a measurable space, $c : \Omega \times \Omega \rightarrow \mathbb{R}$.
 μ, ν two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $T : \Omega \rightarrow \Omega$

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

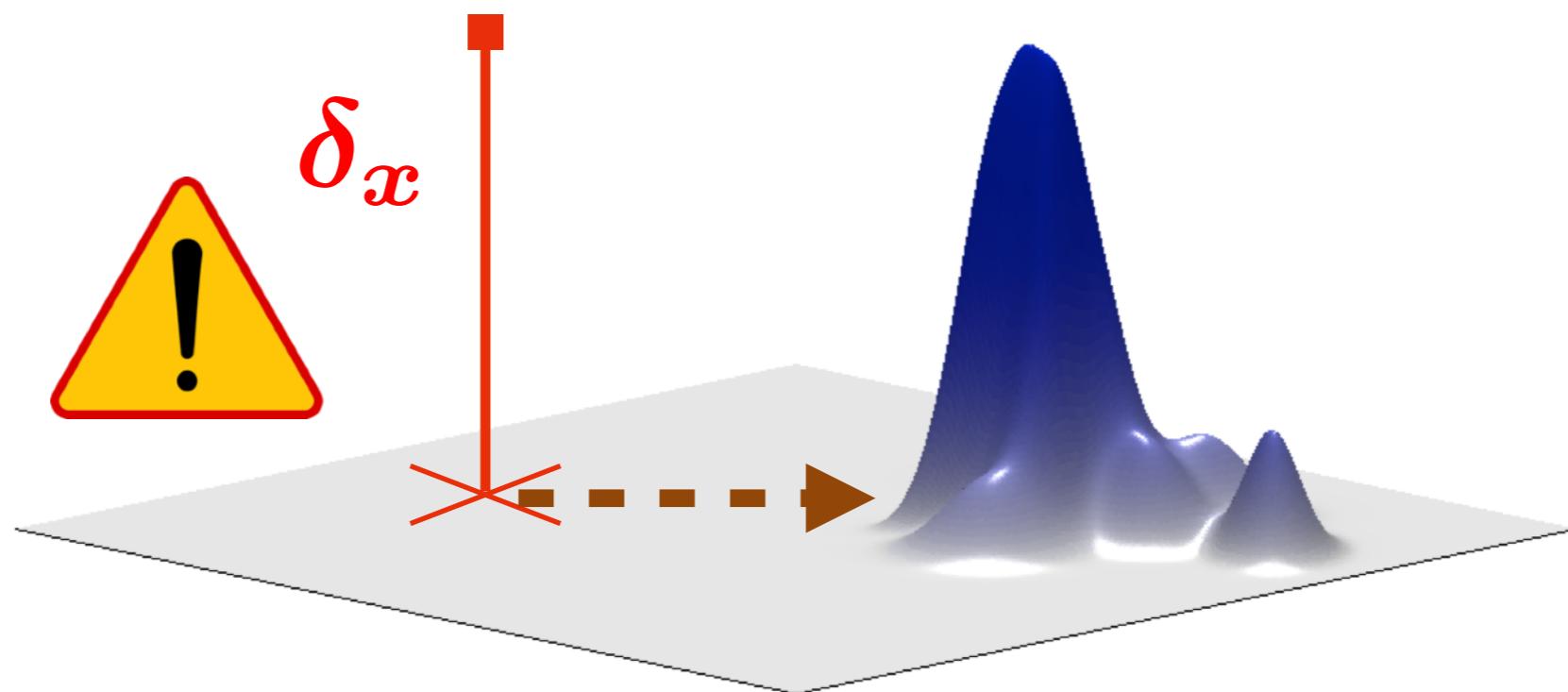


Monge Problem

Ω a measurable space, $c : \Omega \times \Omega \rightarrow \mathbb{R}$.
 μ, ν two probability measures in $\mathcal{P}(\Omega)$.

[Monge'81] problem: find a map $T : \Omega \rightarrow \Omega$

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

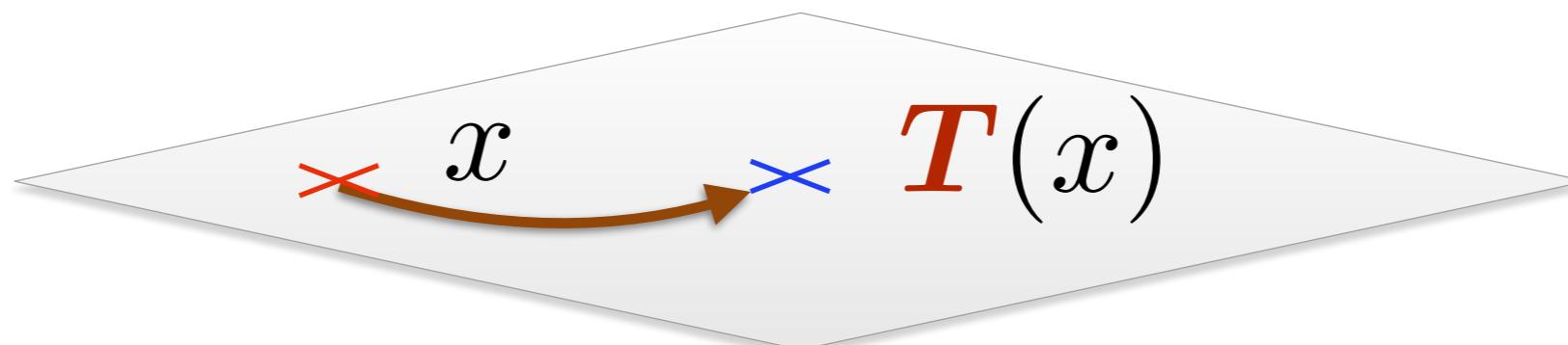


Kantorovich Relaxation

Instead of maps $\textcolor{red}{T} : \Omega \rightarrow \Omega$,

consider probabilistic maps,

i.e. **couplings** $\textcolor{red}{P} \in \mathcal{P}(\Omega \times \Omega)$:



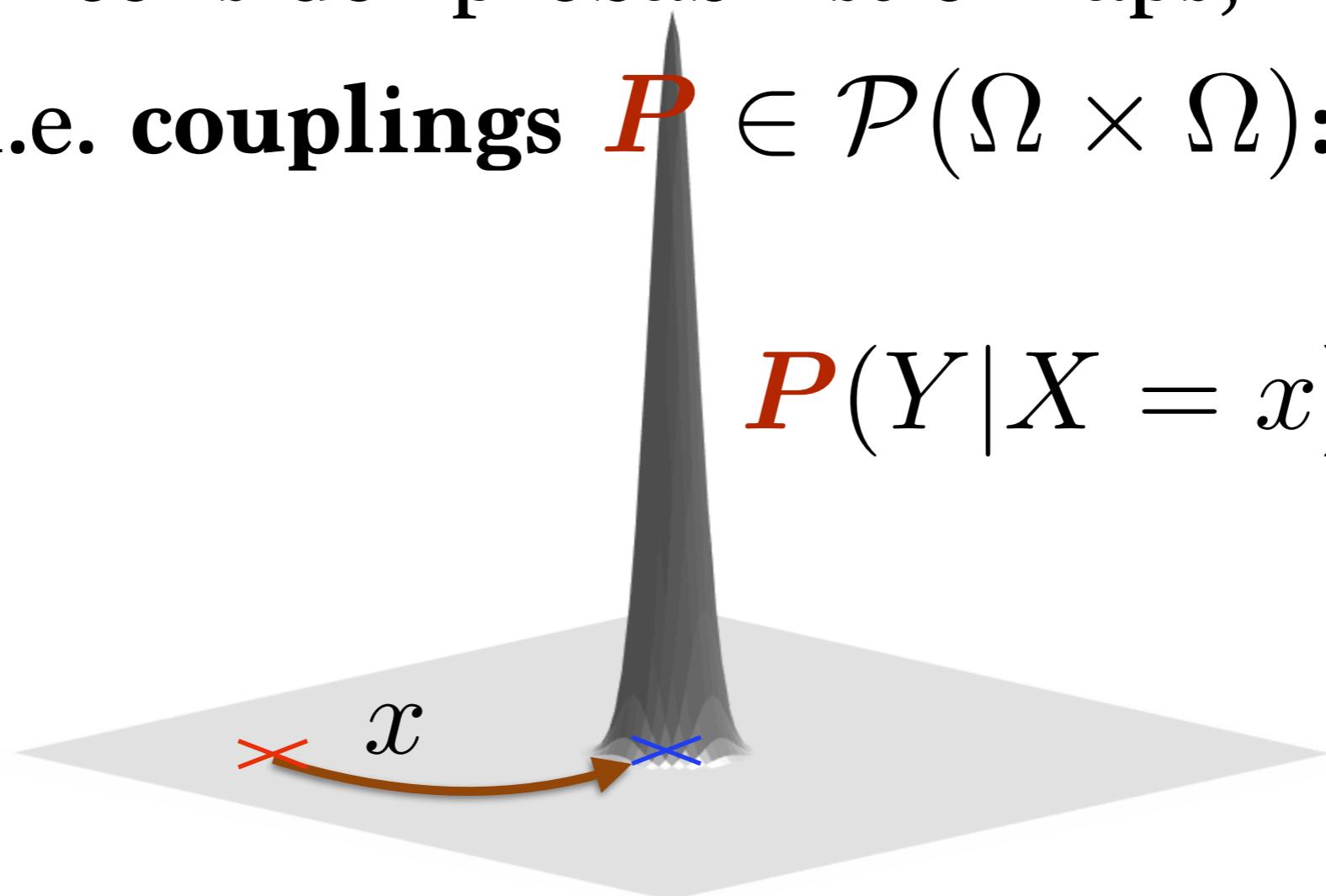
Kantorovich Relaxation

Instead of maps $\textcolor{red}{T} : \Omega \rightarrow \Omega$,

consider probabilistic maps,

i.e. **couplings** $\textcolor{red}{P} \in \mathcal{P}(\Omega \times \Omega)$:

$$\textcolor{red}{P}(Y|X = x)$$



Kantorovich Relaxation

Instead of maps $\textcolor{red}{T} : \Omega \rightarrow \Omega$,

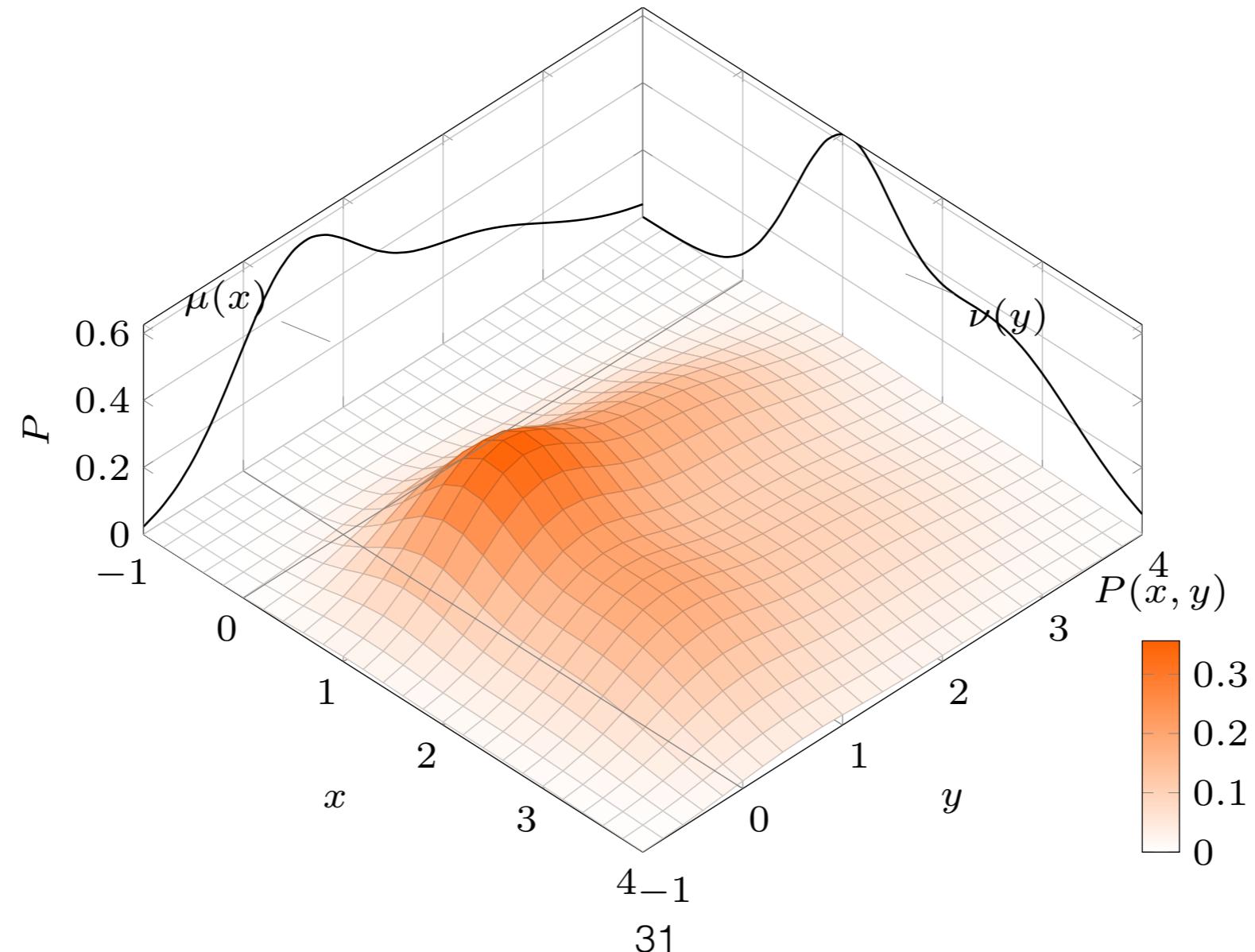
consider probabilistic maps,

i.e. **couplings** $\textcolor{red}{P} \in \mathcal{P}(\Omega \times \Omega)$:

$$\Pi(\mu, \nu) \stackrel{\text{def}}{=} \{ \textcolor{red}{P} \in \mathcal{P}(\Omega \times \Omega) \mid \forall \textcolor{red}{A}, \textcolor{blue}{B} \subset \Omega, \\ \textcolor{red}{P}(A \times \Omega) = \mu(A), \\ \textcolor{red}{P}(\Omega \times B) = \nu(B) \}$$

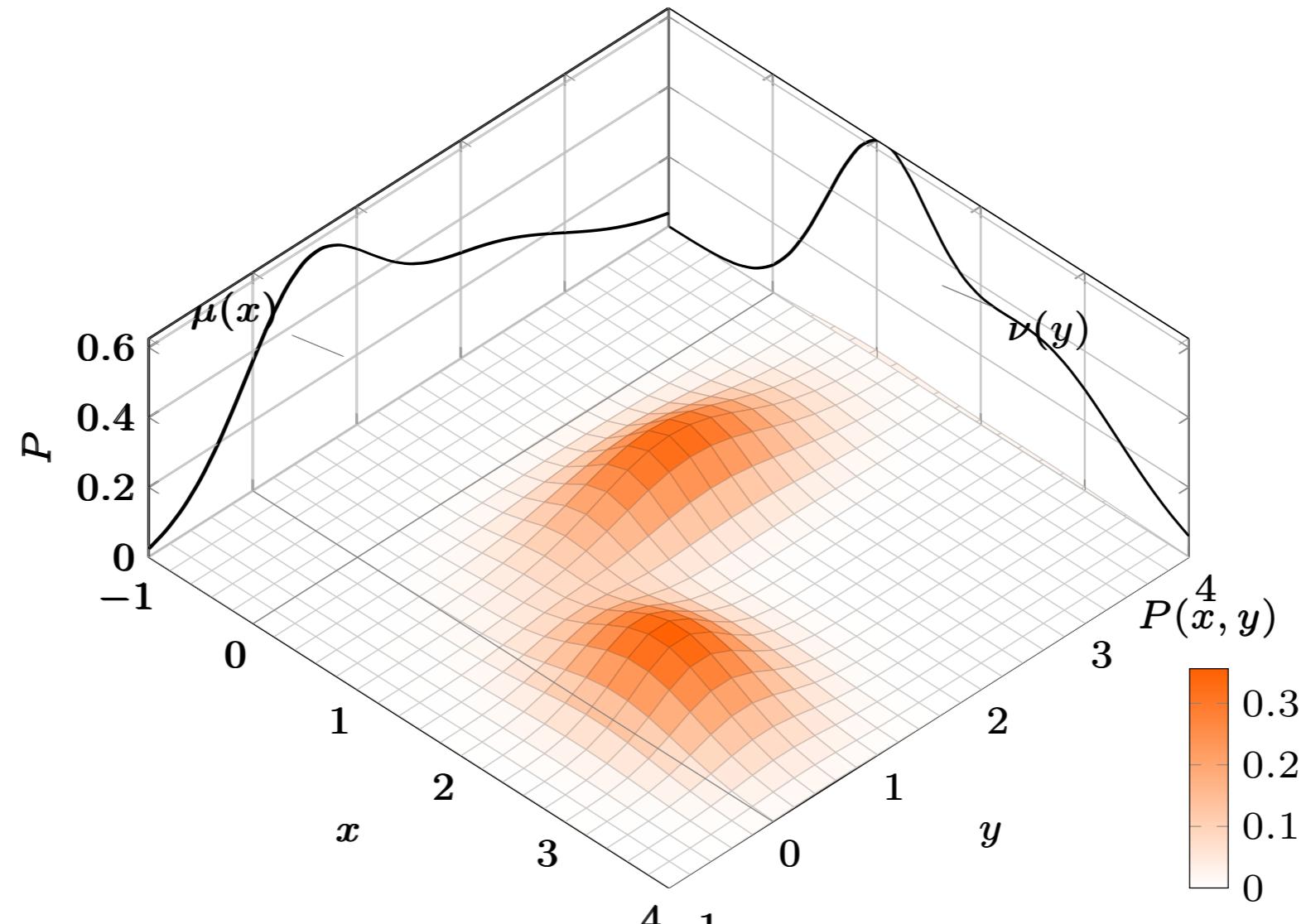
Kantorovich Relaxation

$$\begin{aligned}\Pi(\mu, \nu) &\stackrel{\text{def}}{=} \{P \in \mathcal{P}(\Omega \times \Omega) \mid \forall A, B \subset \Omega, \\ P(A \times \Omega) &= \mu(A), P(\Omega \times B) = \nu(B)\}\end{aligned}$$



Kantorovich Relaxation

$$\begin{aligned}\Pi(\mu, \nu) &\stackrel{\text{def}}{=} \{P \in \mathcal{P}(\Omega \times \Omega) \mid \forall A, B \subset \Omega, \\ P(A \times \Omega) &= \mu(A), P(\Omega \times B) = \nu(B)\}\end{aligned}$$



Kantorovich Problem

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

MONGE

Def. Given μ, ν in $\mathcal{P}(\Omega)$; a cost function c on $\Omega \times \Omega$, the Kantorovich problem is

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

PRIMAL

Kantorovich Problem

Def. Given μ, ν in $\mathcal{P}(\Omega)$; a cost function c on $\Omega \times \Omega$, the Kantorovich problem is

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint c(x, y) \mathbf{P}(dx, dy).$$

PRIMAL

Kantorovich Problem

Def. Given μ, ν in $\mathcal{P}(\Omega)$; a cost function c on $\Omega \times \Omega$, the Kantorovich problem is

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint c(x, y) \mathbf{P}(dx, dy).$$

PRIMAL

$$\sup_{\begin{array}{l} \varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq c(x, y) \end{array}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Kantorovich Problem

Def. Given μ, ν in $\mathcal{P}(\Omega)$; a cost function c on $\Omega \times \Omega$, the Kantorovich problem is

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint c(x, y) \mathbf{P}(dx, dy).$$

PRIMAL

For two real-valued functions φ, ψ on Ω ,

$$(\varphi \oplus \psi)(x, y) \stackrel{\text{def}}{=} \varphi(x) + \psi(y)$$

Kantorovich Problem

Def. Given μ, ν in $\mathcal{P}(\Omega)$; a cost function c on $\Omega \times \Omega$, the Kantorovich problem is

$$\inf_{P \in \Pi(\mu, \nu)} \iint c(x, y) P(dx, dy).$$

PRIMAL

$$\sup_{\begin{array}{c} \varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi \oplus \psi \leq c \end{array}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

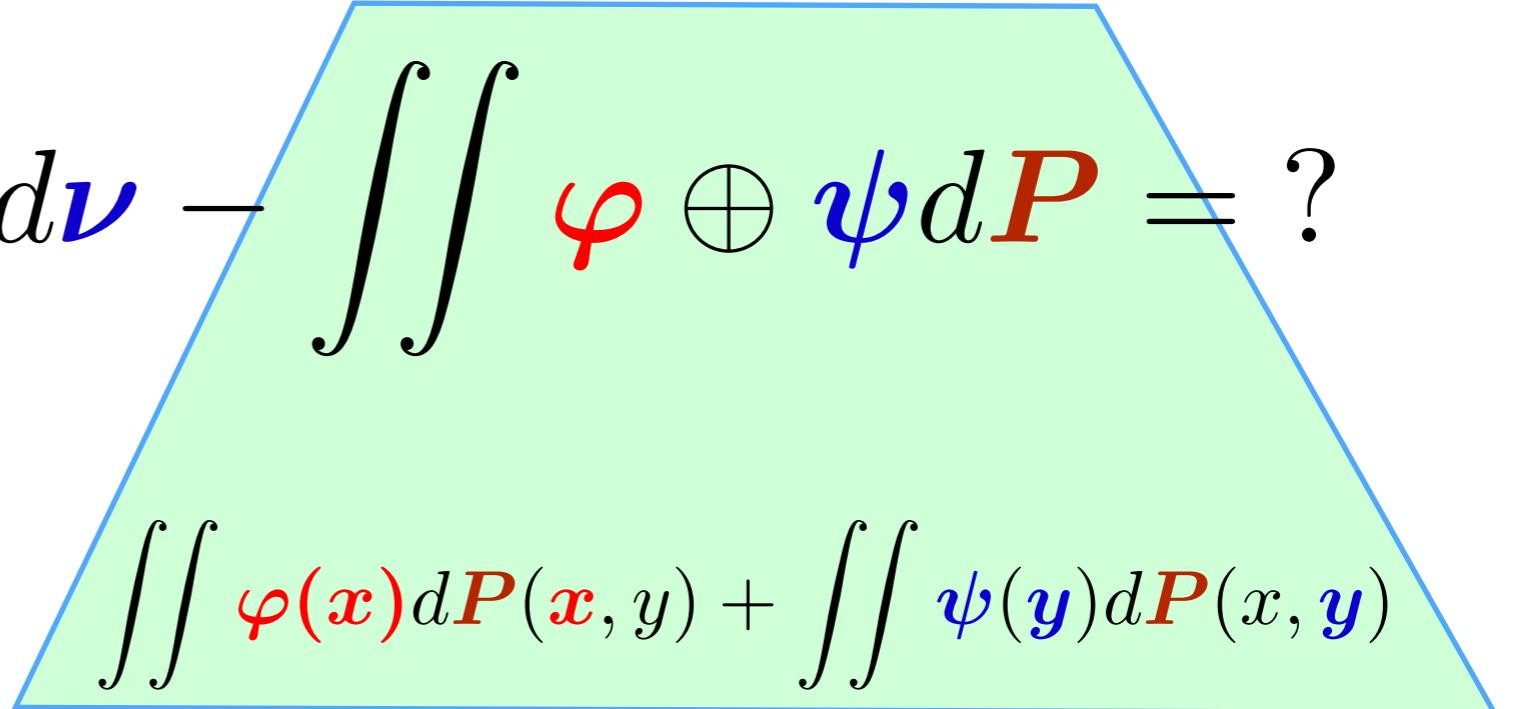
Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = ?$$

Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = ?$$

$$\iint \varphi(x) dP(x, y) + \iint \psi(y) dP(x, y)$$

Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = ?$$

The diagram illustrates the derivation of Kantorovich duality. It features a light green parallelogram representing the space of measures P between two probability measures μ and ν . On the left, a red arrow points from the expression $\int \varphi d\mu + \int \psi d\nu$ towards the parallelogram. On the right, a blue arrow points from the expression $\iint \varphi(x) dP(x,y) + \iint \psi(y) dP(x,y)$ away from the parallelogram. The question mark at the end of the equation indicates that the goal is to find the minimum value of the expression on the left.

Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = 0$$

Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = 0$$

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \mathcal{P}_+(\Omega^2)$.

Deriving Kantorovich Duality

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \Pi(\mu, \nu)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP = 0$$

Let $\varphi, \psi : \Omega \rightarrow \mathbb{R}$, and $P \in \mathcal{P}_+(\Omega^2)$.

$$\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi dP =$$

$$\int \varphi d(\underbrace{\mu - P_X}_{\neq 0}) + \int \psi d(\underbrace{\nu - P_Y}_{\neq 0}) \quad \text{and / or}$$

Deriving Kantorovich Duality

$$\begin{aligned}\iota_\Pi(\mathbf{P}) &= \sup_{\varphi, \psi} \left[\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P} \right] \\ &= \begin{cases} 0 & \text{if } \mathbf{P} \in \Pi(\mu, \nu), \\ +\infty & \text{otherwise.} \end{cases}\end{aligned}$$

Deriving Kantorovich Duality

$$\begin{aligned}\iota_\Pi(\mathbf{P}) &= \sup_{\varphi, \psi} \left[\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P} \right] \\ &= \begin{cases} 0 & \text{if } \mathbf{P} \in \Pi(\mu, \nu), \\ +\infty & \text{otherwise.} \end{cases}\end{aligned}$$

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint c d\mathbf{P}$$

Deriving Kantorovich Duality

$$\begin{aligned}\iota_\Pi(\mathbf{P}) &= \sup_{\varphi, \psi} \left[\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P} \right] \\ &= \begin{cases} 0 & \text{if } \mathbf{P} \in \Pi(\mu, \nu), \\ +\infty & \text{otherwise.} \end{cases}\end{aligned}$$

$$\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint c d\mathbf{P}$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint c d\mathbf{P} + \boxed{\iota_\Pi(\mathbf{P})}$$

Deriving Kantorovich Duality

$$\begin{aligned}\iota_\Pi(\mathbf{P}) &= \sup_{\varphi, \psi} \left[\int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P} \right] \\ &= \begin{cases} 0 & \text{if } \mathbf{P} \in \Pi(\mu, \nu), \\ +\infty & \text{otherwise.} \end{cases}\end{aligned}$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P}$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \sup_{\varphi, \psi} \iint \mathbf{c} d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu - \iint \varphi \oplus \psi d\mathbf{P}$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \sup_{\varphi, \psi} \iint \mathbf{c} d\mathbf{P} - \iint \varphi \oplus \psi d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \sup_{\varphi, \psi} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\sup_{\varphi, \psi} \inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} \quad + \int \varphi d\mu + \int \psi d\nu$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\sup_{\varphi, \psi} \inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\sup_{\varphi, \psi} \inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} = \begin{cases} 0 & \text{if } \mathbf{c} - \varphi \oplus \psi \geq 0. \\ -\infty & \text{otherwise} \end{cases}$$

Deriving Kantorovich Duality

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint \mathbf{c} d\mathbf{P} + \iota_\Pi(\mathbf{P})$$

$$\sup_{\varphi, \psi} \inf_{\mathbf{P} \in \mathcal{P}_+(\Omega^2)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} + \int \varphi d\mu + \int \psi d\nu$$

$$\inf_{\mathbf{P} \in \mathcal{P}_+(\Omega)} \iint (\mathbf{c} - \varphi \oplus \psi) d\mathbf{P} = \begin{cases} 0 & \text{if } \mathbf{c} - \varphi \oplus \psi \geq 0. \\ -\infty & \text{otherwise} \end{cases}$$

$$\sup_{\varphi \oplus \psi \leq \mathbf{c}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Wasserstein Distances

Let $p \geq 1$. Let $\mathbf{c}(x, y) := \mathbf{D}^p(x, y)$, a metric.

Def. The p -Wasserstein distance between μ, ν in $\mathcal{P}(\Omega)$ is

$$W_p(\mu, \nu) \stackrel{\text{def}}{=} \left(\inf_{\mathbf{P} \in \Pi(\mu, \nu)} \iint \mathbf{D}(x, y)^p \mathbf{P}(dx, dy) \right)^{1/p}.$$

Wasserstein Distances

Let $p \geq 1$. Let $\mathbf{c}(x, y) := \mathbf{D}^p(x, y)$, a metric.

Def. The p -Wasserstein distance between μ, ν in $\mathcal{P}(\Omega)$ is

$$W_p(\mu, \nu) \stackrel{\text{def}}{=} \left(\inf_{P \in \Pi(\mu, \nu)} \iint \mathbf{D}(x, y)^p P(dx, dy) \right)^{\frac{1}{p}}.$$

Kantorovich Duality

$$W_p^p(\mu, \nu) = \sup_{\begin{array}{l} \varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y) \end{array}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

- Kantorovich Duality is **interesting** from a computational perspective: easier to store 2 functions than a whole coupling.
- D transforms: go from **two** to **one** dual potential.

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Imagine we choose a φ . Can we find a good ψ ?

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Imagine we choose a φ . Can we find a good ψ ?

We need that ψ satisfies for all x, y

$$\varphi(x) + \psi(y) \leq D^p(x, y)$$

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Imagine we choose a φ . Can we find a good ψ ?

We need that ψ satisfies for all x, y

$$\varphi(x) + \psi(y) \leq D^p(x, y)$$

$$\psi(y) \leq D^p(x, y) - \varphi(x)$$

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

Imagine we choose a φ . Can we find a good ψ ?

We need that ψ satisfies for all x, y

$$\varphi(x) + \psi(y) \leq D^p(x, y)$$

$$\psi(y) \leq D^p(x, y) - \varphi(x)$$

$$\psi(y) \leq \inf_x D^p(x, y) - \varphi(x)$$

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

For given φ , cannot get a better ψ than

$$\bar{\varphi}(y) \stackrel{\text{def}}{=} \inf_x D^p(x, y) - \varphi(x).$$

D transforms

$$W_p^p(\mu, \nu) = \sup_{\substack{\varphi \in L_1(\mu), \psi \in L_1(\nu) \\ \varphi(x) + \psi(y) \leq D^p(x, y)}} \int \varphi d\mu + \int \psi d\nu.$$

DUAL

For given φ , cannot get a better ψ than

$$\bar{\varphi}(y) \stackrel{\text{def}}{=} \inf_x D^p(x, y) - \varphi(x).$$

$$W_p^p(\mu, \nu) = \sup_{\varphi} \int \varphi d\mu + \int \bar{\varphi} d\nu.$$

SEMI-DUAL

D transforms

$$\overline{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\overline{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

D transforms

$$\overline{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\overline{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\varphi} \int \overline{\varphi} d\boldsymbol{\mu} + \int \overline{\varphi} d\boldsymbol{\nu}.$$

D transforms

$$\overline{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\overline{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\varphi} \int \overline{\varphi} d\boldsymbol{\mu} + \int \overline{\varphi} d\boldsymbol{\nu}.$$



D transforms

$$\overline{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\overline{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\varphi} \int \overline{\varphi} d\boldsymbol{\mu} + \int \overline{\varphi} d\boldsymbol{\nu}.$$

For all φ , we have $\overline{\overline{\varphi}} = \overline{\varphi}$

D transforms

$$\bar{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\bar{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\varphi} \int \bar{\varphi} d\boldsymbol{\mu} + \int \bar{\varphi} d\boldsymbol{\nu}.$$

For all φ , we have $\overline{\overline{\varphi}} = \bar{\varphi}$

φ is D^p -concave if $\exists \phi : \varphi = \bar{\phi}$

φ is D^p -concave $\Rightarrow \overline{\overline{\varphi}} = \varphi$

D transforms

$$\overline{\varphi}(\mathbf{y}) \stackrel{\text{def}}{=} \inf_{\mathbf{x}} D^p(\mathbf{x}, \mathbf{y}) - \varphi(\mathbf{x}).$$

$$\overline{\psi}(\mathbf{x}) = \inf_{\mathbf{y}} D^p(\mathbf{x}, \mathbf{y}) - \psi(\mathbf{y}).$$

$$W_p^p(\mu, \nu) = \sup_{\varphi} \int \overline{\varphi} d\mu + \int \overline{\varphi} d\nu.$$

$$W_p^p(\mu, \nu) = \sup_{\varphi \text{ is } D^p\text{-concave}} \int \varphi d\mu + \int \overline{\varphi} d\nu.$$

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.
 $\bar{\varphi}_x(y) - \bar{\varphi}_x(y') = D(x, y) - D(x, y') \leq D(y, y')$

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.
 $\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.
 $\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.
 $\Rightarrow \bar{\varphi}(y) - \bar{\varphi}(x) \leq D(x, y)$

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) - \bar{\varphi}(x) \leq D(x, y)$

$\Rightarrow -\bar{\varphi}(x) \leq D(x, y) - \bar{\varphi}(y)$

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) - \bar{\varphi}(x) \leq D(x, y)$

$\Rightarrow -\bar{\varphi}(x) \leq D(x, y) - \bar{\varphi}(y)$

$\Rightarrow -\bar{\varphi}(x) \leq \inf_y D(x, y) - \bar{\varphi}(y)$

D transforms, W_1

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.

$$\Rightarrow \bar{\varphi}(y) - \bar{\varphi}(x) \leq D(x, y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq D(x, y) - \bar{\varphi}(y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq \inf_y D(x, y) - \bar{\varphi}(y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq \inf_y D(x, y) - \bar{\varphi}(y) \leq -\bar{\varphi}(x)$$

\$D\$ transforms, \$W_1\$

Prop. If $c = D$, namely $p = 1$, then
 φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

For given x , $\bar{\varphi}_x(y) \stackrel{\text{def}}{=} D(x, y) - \varphi(x)$ is 1-Lipschitz.

$\Rightarrow \bar{\varphi}(y) = \inf_x \bar{\varphi}_x(y)$ is 1-Lipschitz.

$$\Rightarrow \bar{\varphi}(y) - \bar{\varphi}(x) \leq D(x, y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq D(x, y) - \bar{\varphi}(y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq \inf_y D(x, y) - \bar{\varphi}(y)$$

$$\Rightarrow -\bar{\varphi}(x) \leq \inf_y D(x, y) - \bar{\varphi}(y) \leq -\bar{\varphi}(x)$$

$$\Rightarrow -\bar{\varphi}(x) \leq \bar{\varphi}(x) \leq -\bar{\varphi}(x) \text{ and } \bar{\varphi}(x) = -\bar{\varphi}(x)$$

D transforms, W_1

$$W_1(\mu, \nu) = \sup_{\varphi \text{ is } D\text{-concave}} \int \varphi d\mu + \int \bar{\varphi} d\nu.$$

SEMI-DUAL

Prop. If $c = D$, then

φ is D -concave $\Leftrightarrow \bar{\varphi} = -\varphi$, φ is 1-Lipschitz

$$W_1(\mu, \nu) = \sup_{\varphi \text{ 1-Lipschitz}} \int \varphi(d\mu - d\nu).$$

W1

Links between Monge & Kantorovich

Prop. For “well behaved” costs c , if μ has a density then an *optimal* Monge map T^* between μ and ν must exist.

Prop. In that case

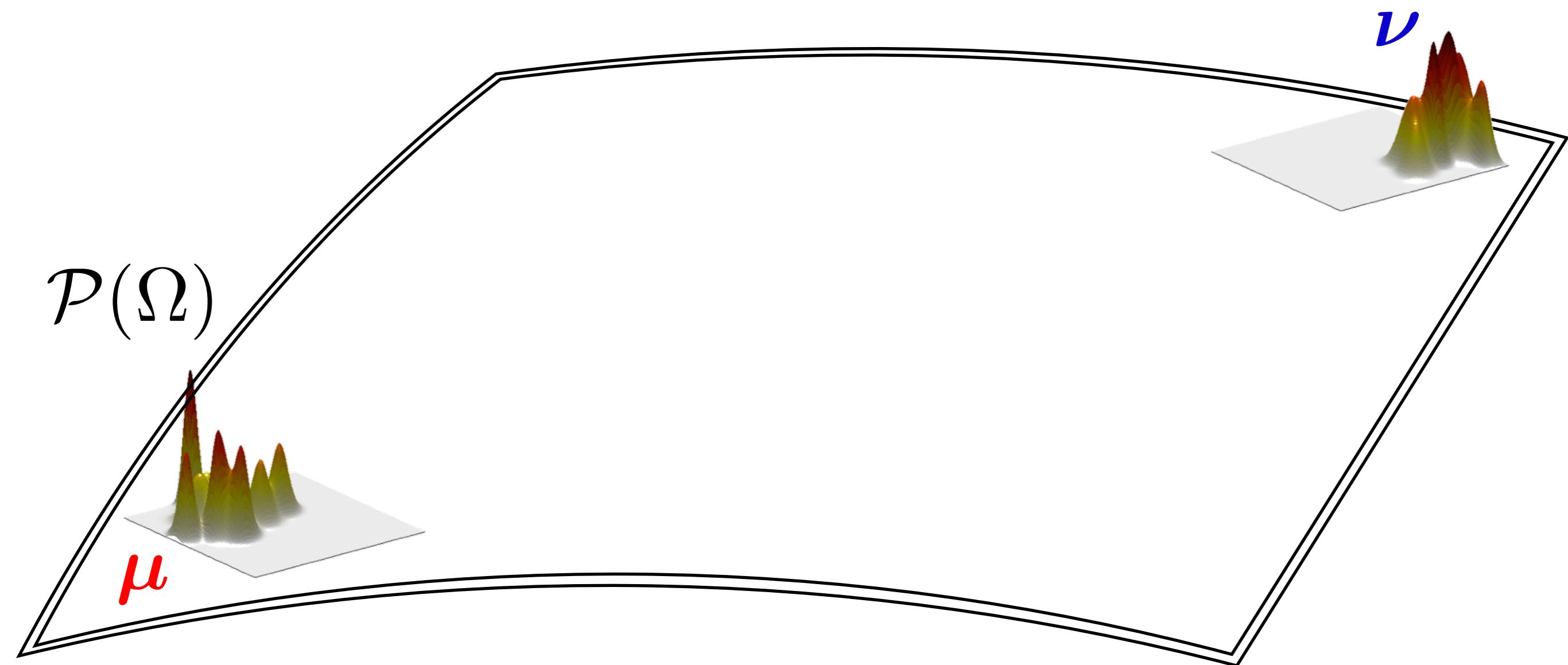
$$P^* := (\text{Id}, T^*)_{\sharp} \mu \in \Pi(\mu, \nu)$$

is also *optimal* for the Kantorovich problem.

[Brenier'91] [Smith&Knott'87] [McCann'01]

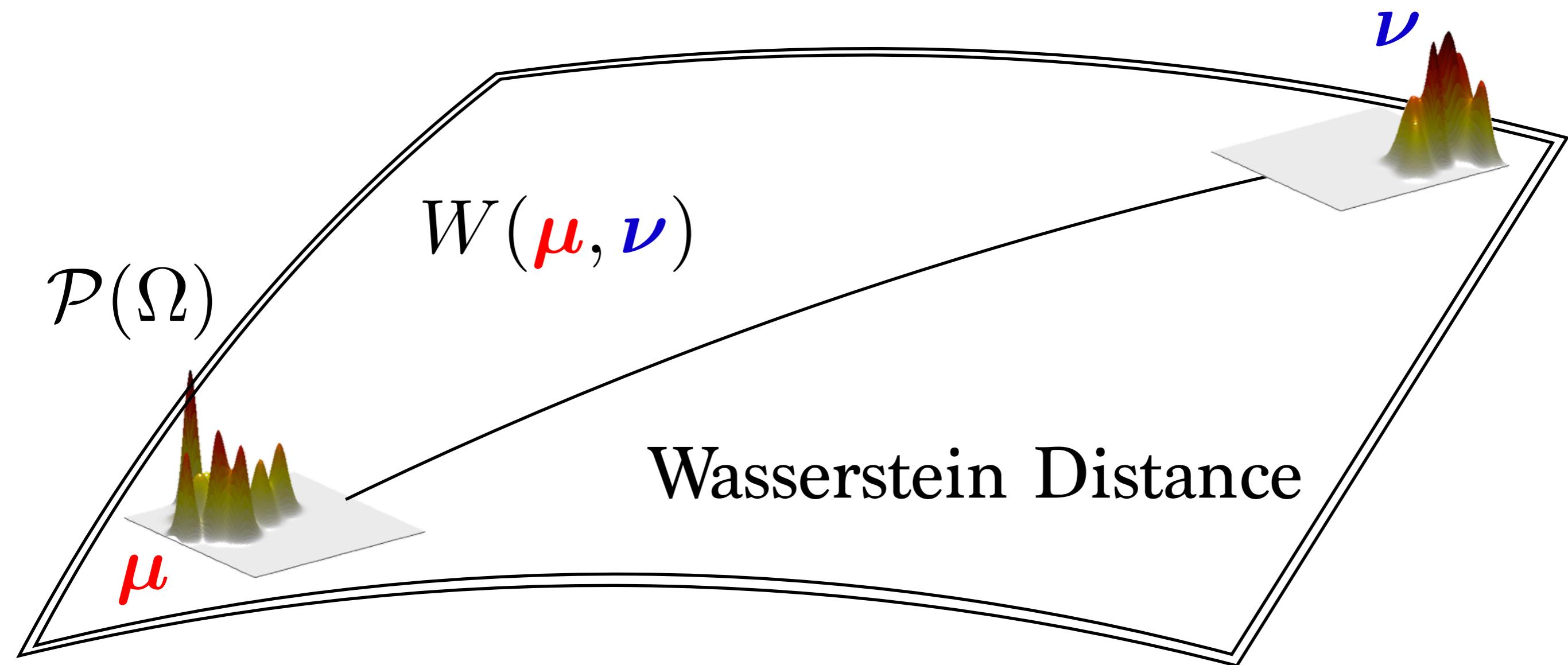
Optimal Transport Geometry

Very different geometry than standard information divergences (KL, Euclidean)



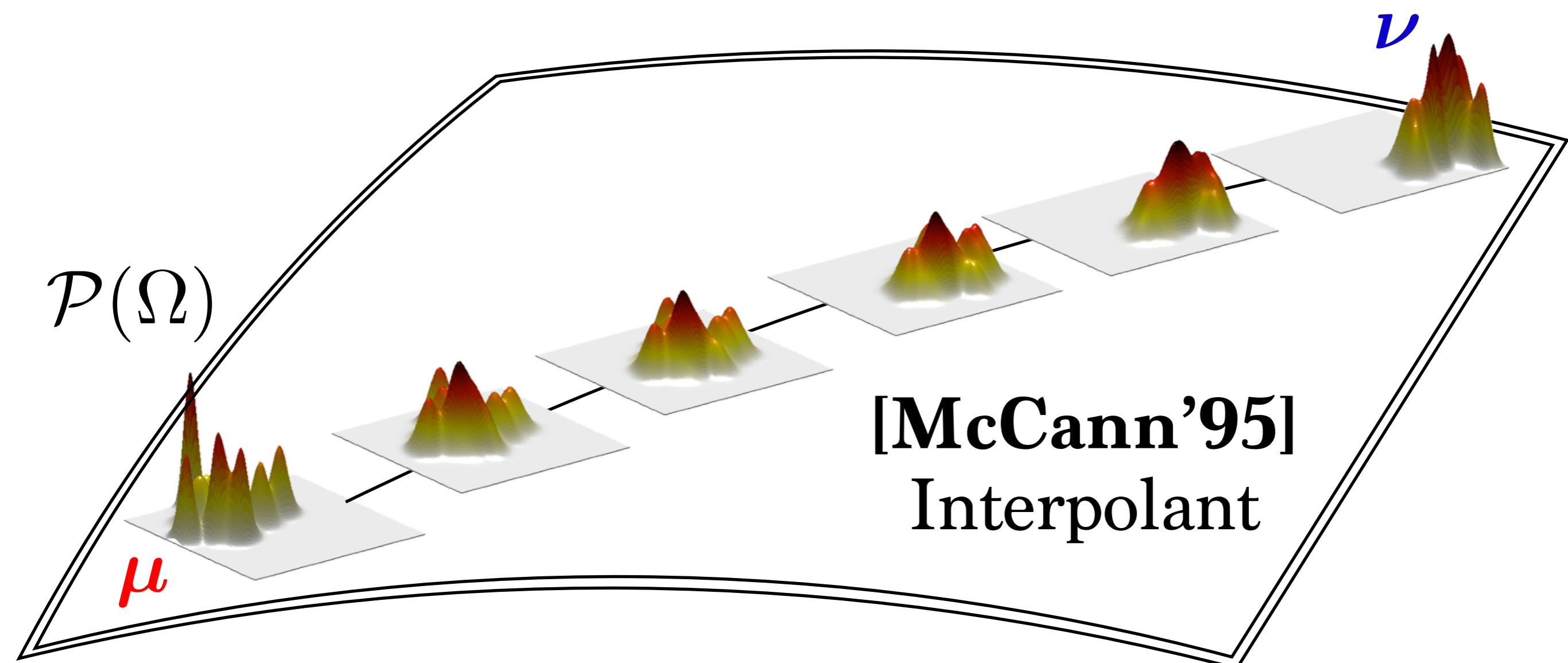
Optimal Transport Geometry

Very different geometry than standard information divergences (KL, Euclidean)



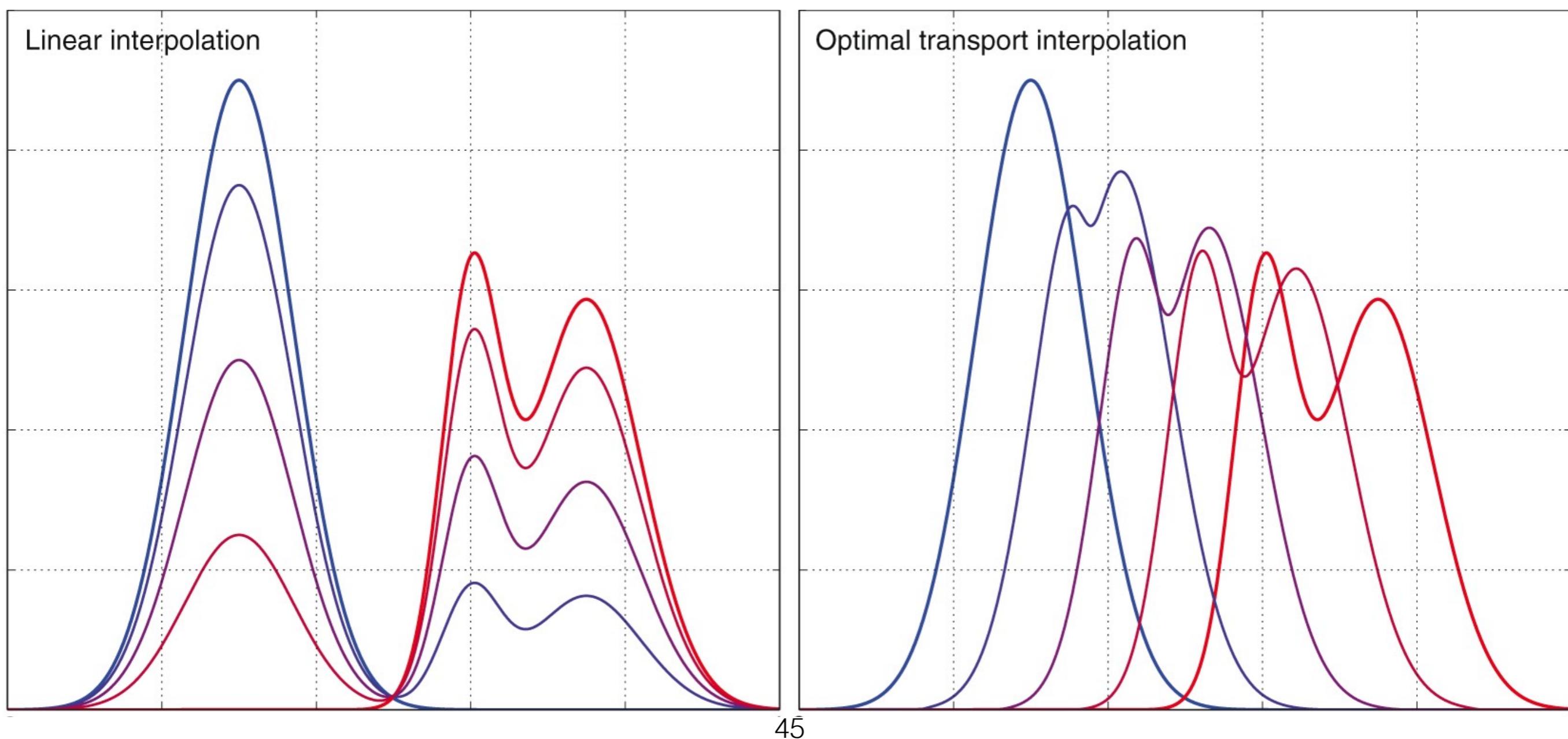
Optimal Transport Geometry

Very different geometry than standard information divergences (KL, Euclidean)



Optimal Transport Geometry

Very different geometry than standard information divergences (KL, Euclidean)



Optimal Transport Geometry

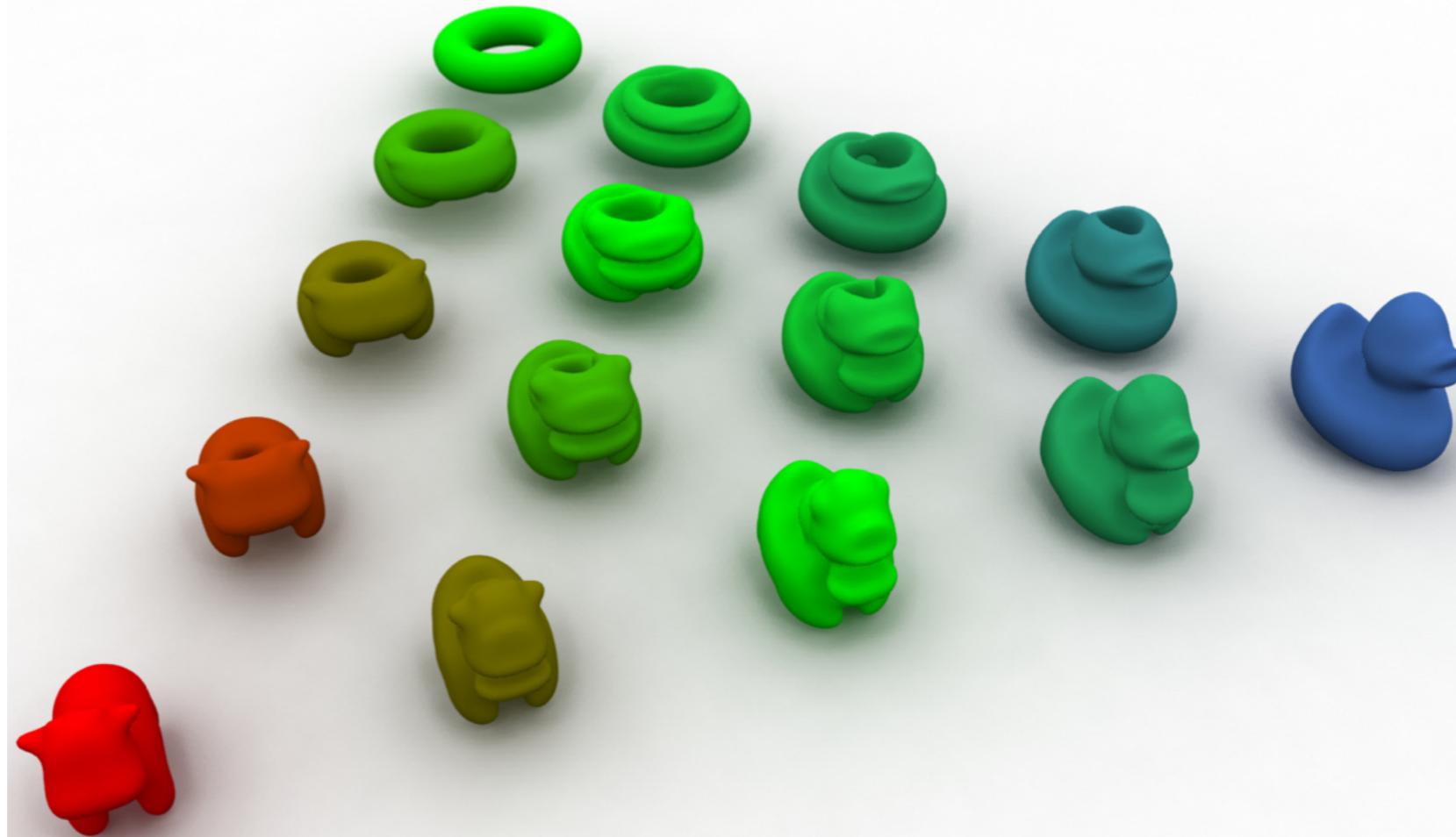
Very different geometry than standard information divergences (KL, Euclidean)



[SDPC..'15]

Optimal Transport Geometry

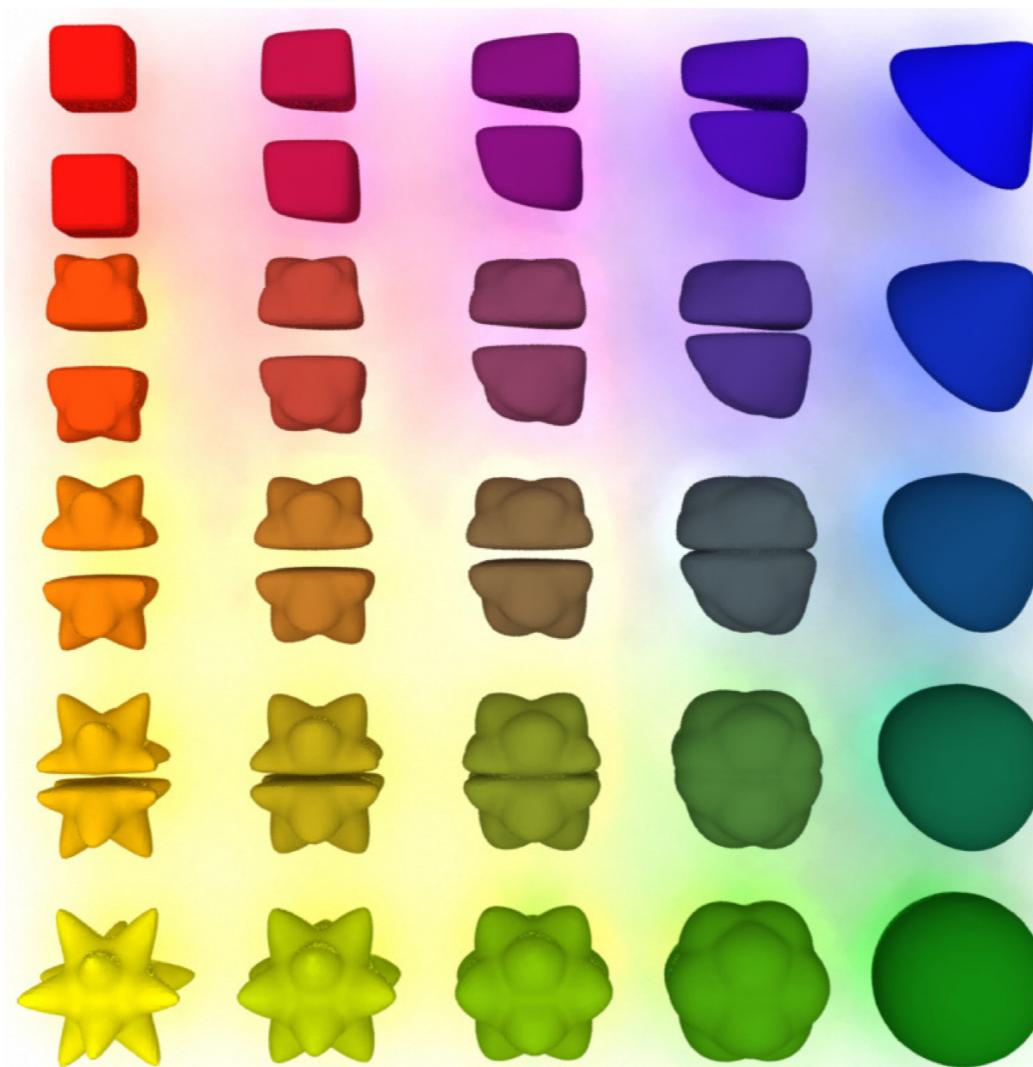
Very different geometry than standard information divergences (KL, Euclidean)



[SDPC..'15]

Optimal Transport Geometry

Very different geometry than standard information divergences (KL, Euclidean)



[SDPC..'15]

Variational OT Problems in ML

Up to 2010: OT solvers
used mostly for retrieval
in databases of histograms

$$W_p(\mu, \nu) = ?$$

$$W_p(\mu, \nu) \leq \dots ?$$

OT is now used as a **loss** or **fidelity** term:

$$\operatorname{argmin}_{\mu \in \mathcal{P}(\Omega)} F(W_p(\mu, \nu_1), W_p(\mu, \nu_2), \dots, \mu) = ?$$

[Jordan Kinderlehrer Otto'98]

“ ∇_{μ} ” $W_p(\mu, \nu) = ?$

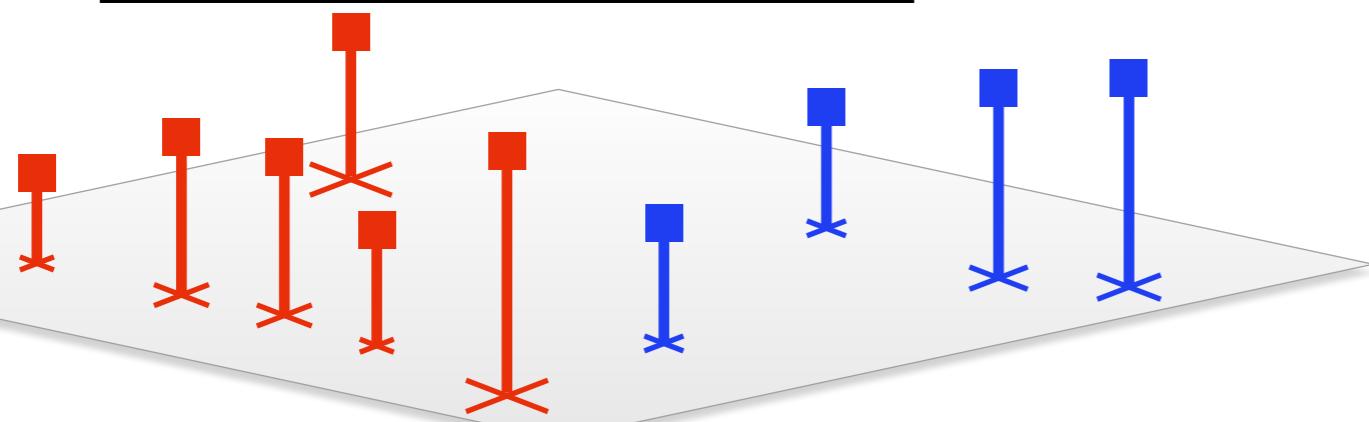
[Ambrosio Gigli Savaré'05]

2. Computing OT exactly

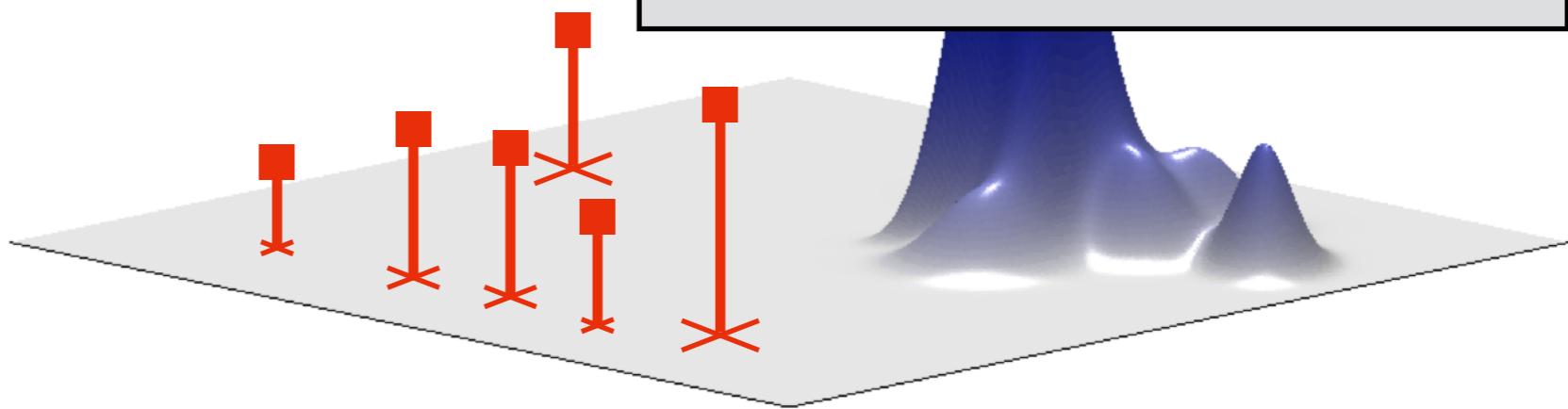
- Typology: discrete/continuous problems
- Easy cases and exact solvers for discrete measures.

When can we compute OT?

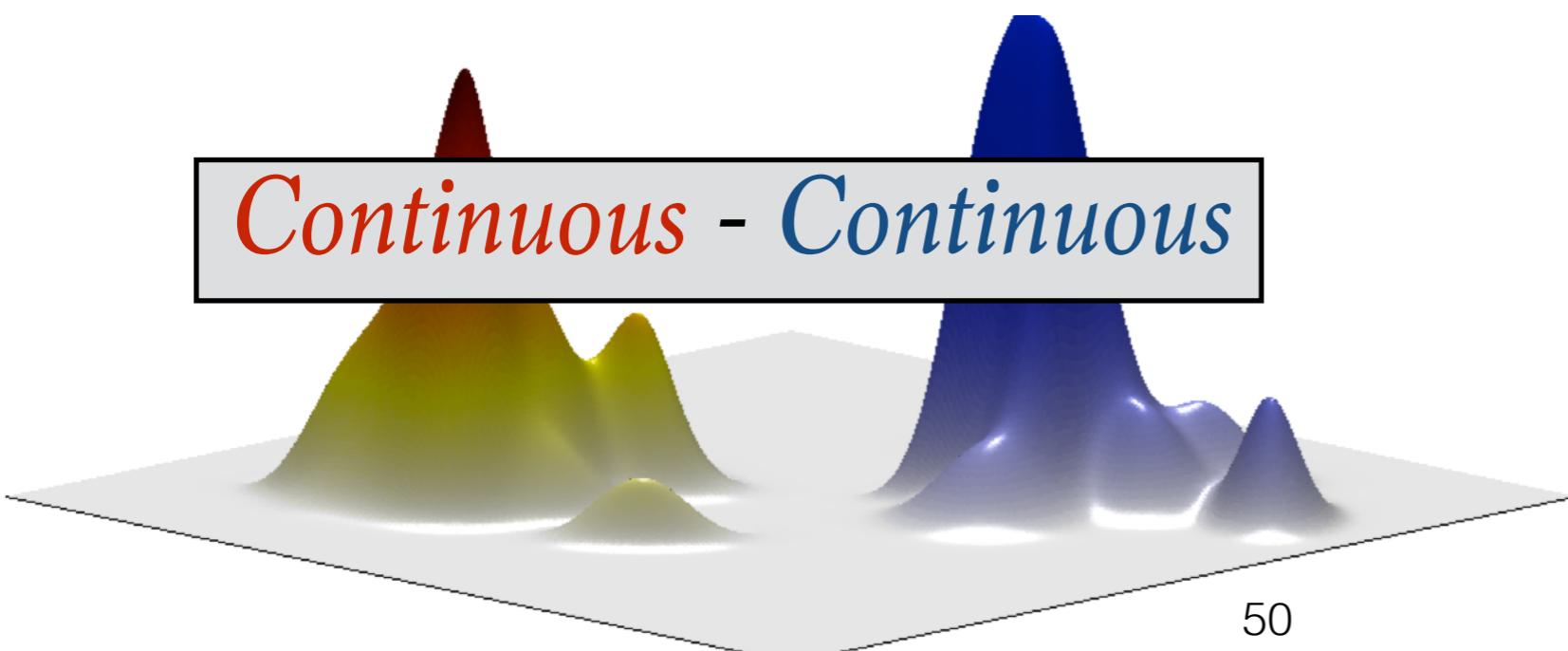
Discrete - Discrete



Discrete - Continuous

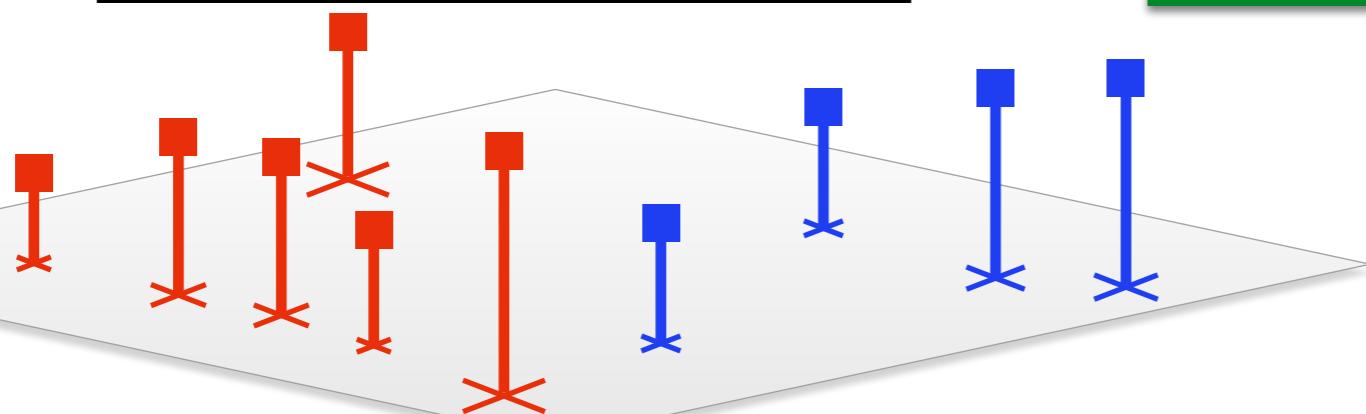


Continuous - Continuous



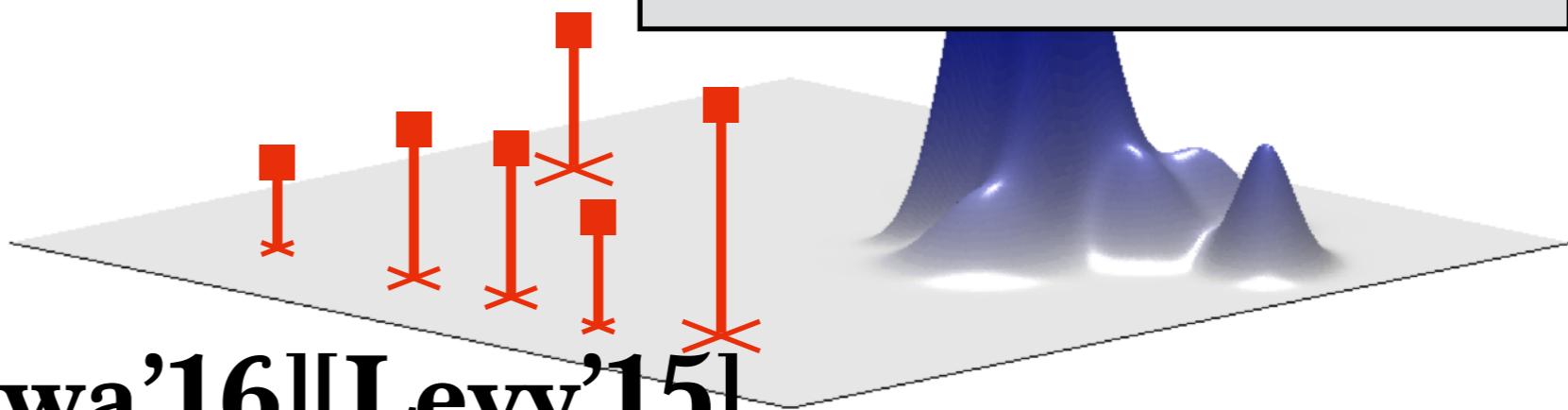
When can we compute OT?

Discrete - Discrete



Network flow solvers

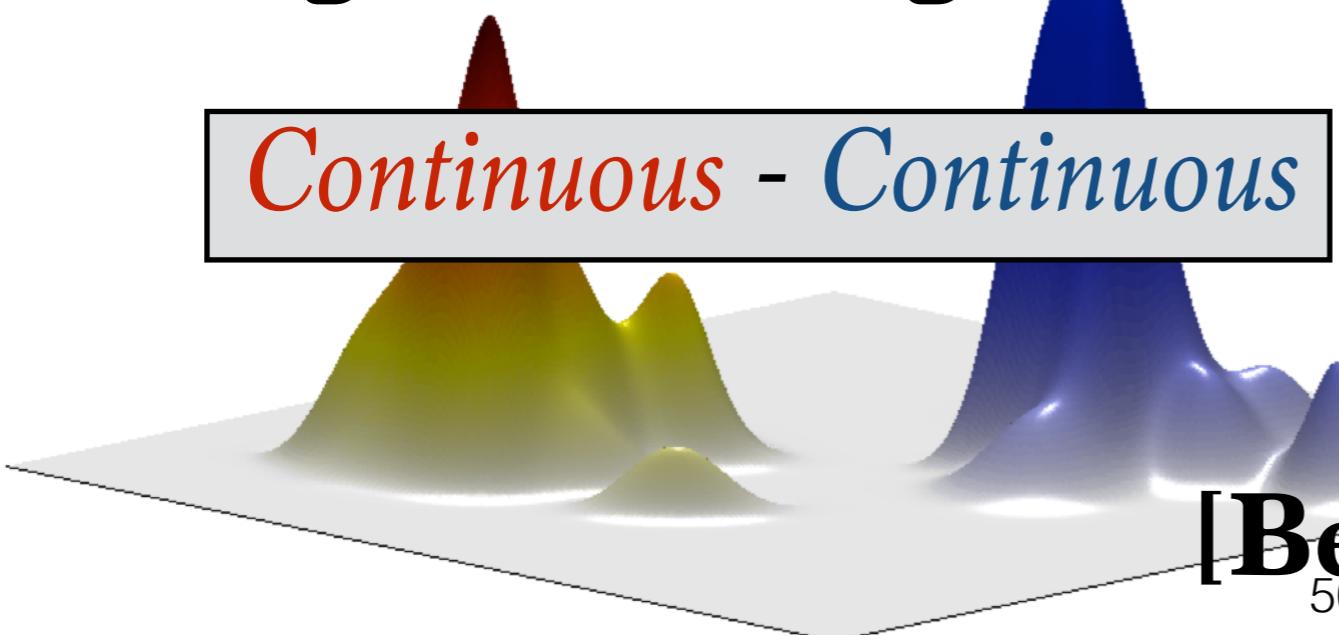
Discrete - Continuous



low dim.

[Mérigot'11][Kitagawa'16][Levy'15]

Continuous - Continuous



PDE's

[Benamou'98]

Stochastic
Optimization

[Genevay'16]

[Arjovsky'17]

Easy (1): Univariate Measures

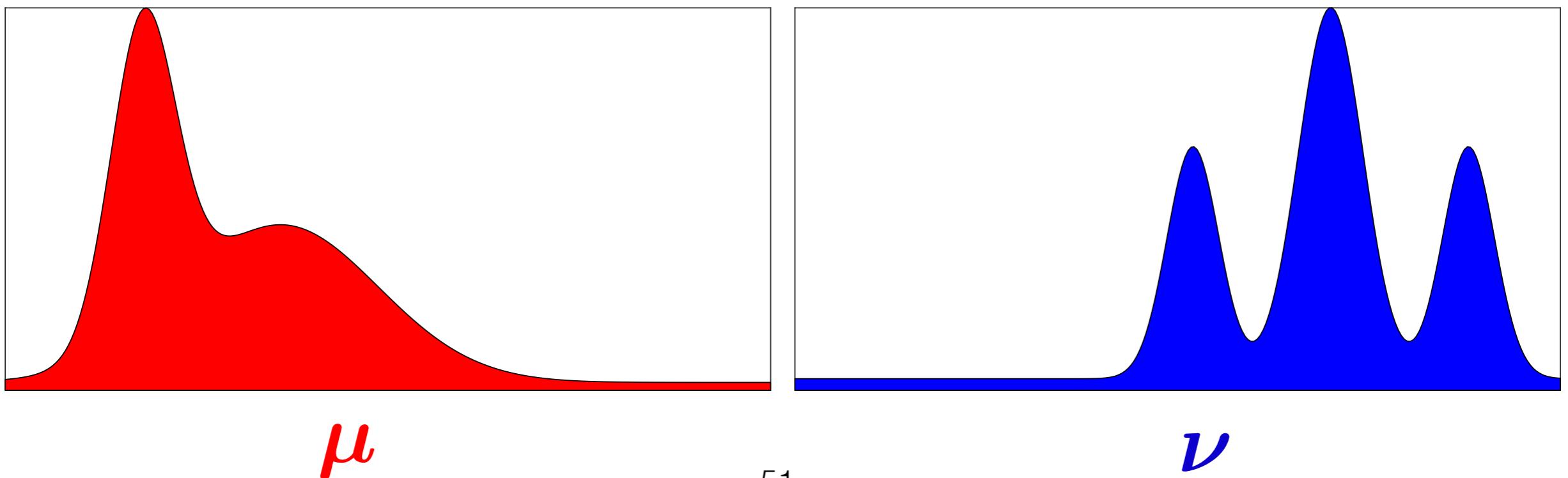
Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\boldsymbol{\mu}}^{-1}, F_{\boldsymbol{\nu}}^{-1}$ quantile functions,

$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \int_0^1 \textcolor{green}{c}(|F_{\boldsymbol{\mu}}^{-1}(x) - F_{\boldsymbol{\nu}}^{-1}(x)|) dx$$

Easy (1): Univariate Measures

Remark. If $\Omega = \mathbb{R}$, $\textcolor{red}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\boldsymbol{\mu}}^{-1}, F_{\boldsymbol{\nu}}^{-1}$ quantile functions,

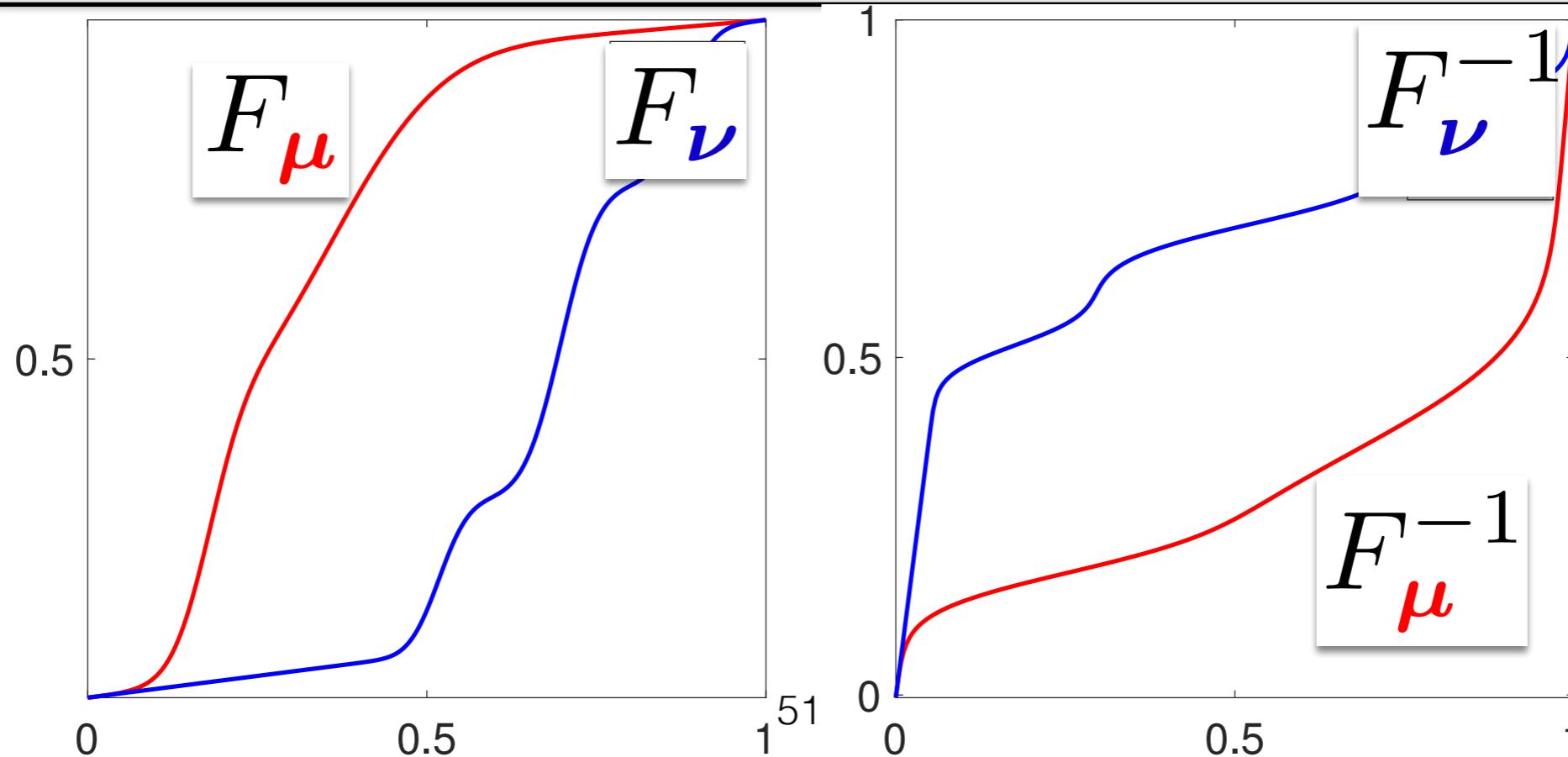
$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \int_0^1 \textcolor{green}{c}(|F_{\boldsymbol{\mu}}^{-1}(x) - F_{\boldsymbol{\nu}}^{-1}(x)|) dx$$



Easy (1): Univariate Measures

Remark. If $\Omega = \mathbb{R}$, $c(x, y) = c(|x - y|)$,
 c convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

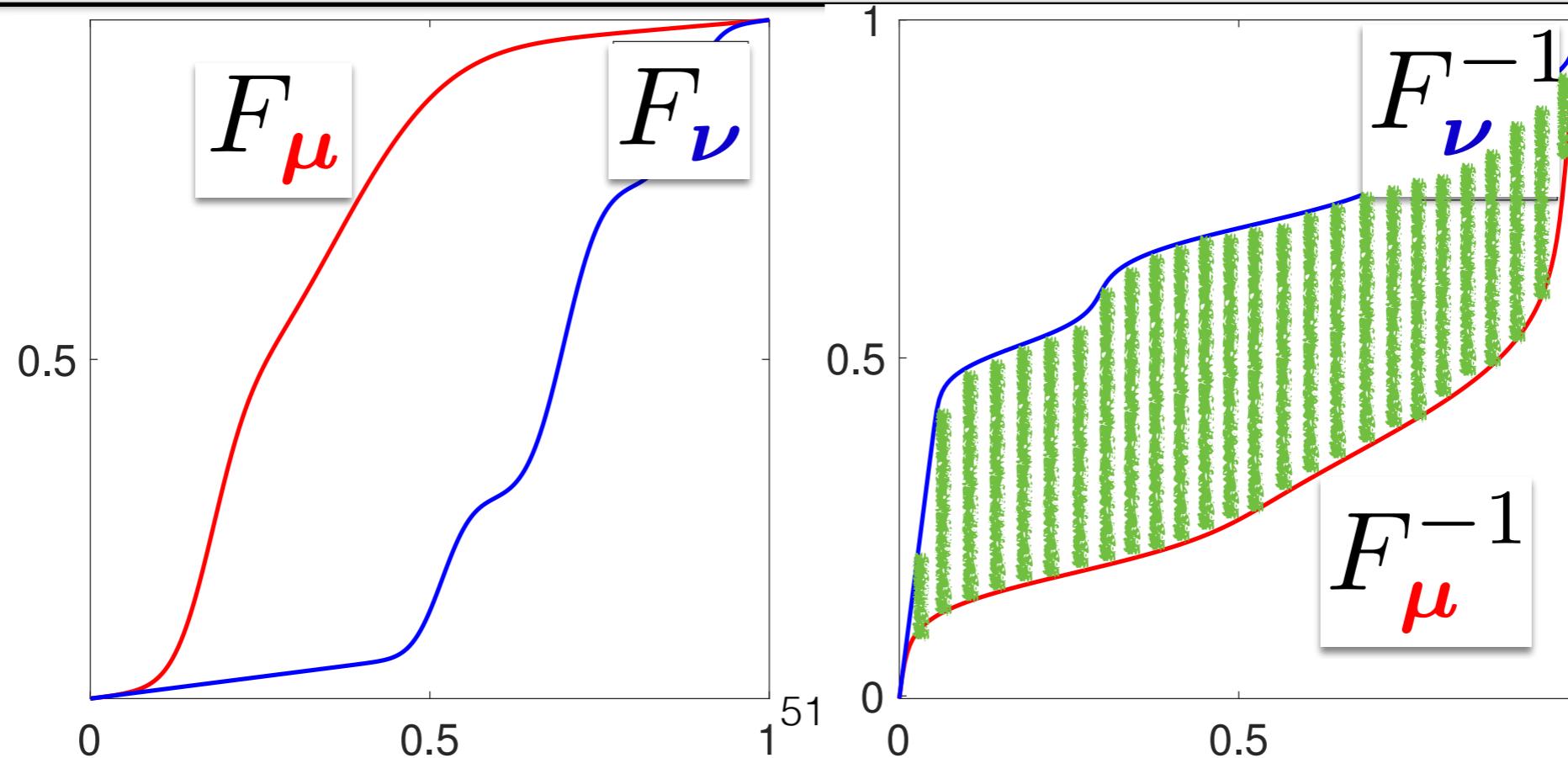
$$W(\mu, \nu) = \int_0^1 c(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Easy (1): Univariate Measures

Remark. If $\Omega = \mathbb{R}$, $c(x, y) = c(|x - y|)$,
 c convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 c(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Easy (2): Gaussian Measures

Remark. If $\Omega = \mathbb{R}^d$, $\textcolor{green}{c}(x, y) = \|x - y\|^2$, and $\mu = \mathcal{N}(\mathbf{m}_\mu, \Sigma_\mu)$, $\nu = \mathcal{N}(\mathbf{m}_\nu, \Sigma_\nu)$ then

$$W_2^2(\mu, \nu) = \|\mathbf{m}_\mu - \mathbf{m}_\nu\|^2 + B(\Sigma_\mu, \Sigma_\nu)^2$$

where B is the Bures metric

$$B(\Sigma_\mu, \Sigma_\nu)^2 = \text{trace}(\Sigma_\mu + \Sigma_\nu - 2(\Sigma_\mu^{1/2} \Sigma_\nu \Sigma_\mu^{1/2})^{1/2}).$$

Easy (2): Gaussian Measures

Remark. If $\Omega = \mathbb{R}^d$, $c(x, y) = \|x - y\|^2$, and $\mu = \mathcal{N}(\mathbf{m}_\mu, \Sigma_\mu)$, $\nu = \mathcal{N}(\mathbf{m}_\nu, \Sigma_\nu)$ then

$$W_2^2(\mu, \nu) = \|\mathbf{m}_\mu - \mathbf{m}_\nu\|^2 + B(\Sigma_\mu, \Sigma_\nu)^2$$

where B is the Bures metric

$$B(\Sigma_\mu, \Sigma_\nu)^2 = \text{trace}(\Sigma_\mu + \Sigma_\nu - 2(\Sigma_\mu^{1/2} \Sigma_\nu \Sigma_\mu^{1/2})^{1/2}).$$

The map $T : x \mapsto \mathbf{m}_\nu + A(x - \mathbf{m}_\mu)$ is optimal,

$$\text{where } A = \Sigma_\mu^{-\frac{1}{2}} \left(\Sigma_\mu^{\frac{1}{2}} \Sigma_\nu \Sigma_\mu^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_\mu^{-\frac{1}{2}}.$$

Easy (2): Gaussian Measures

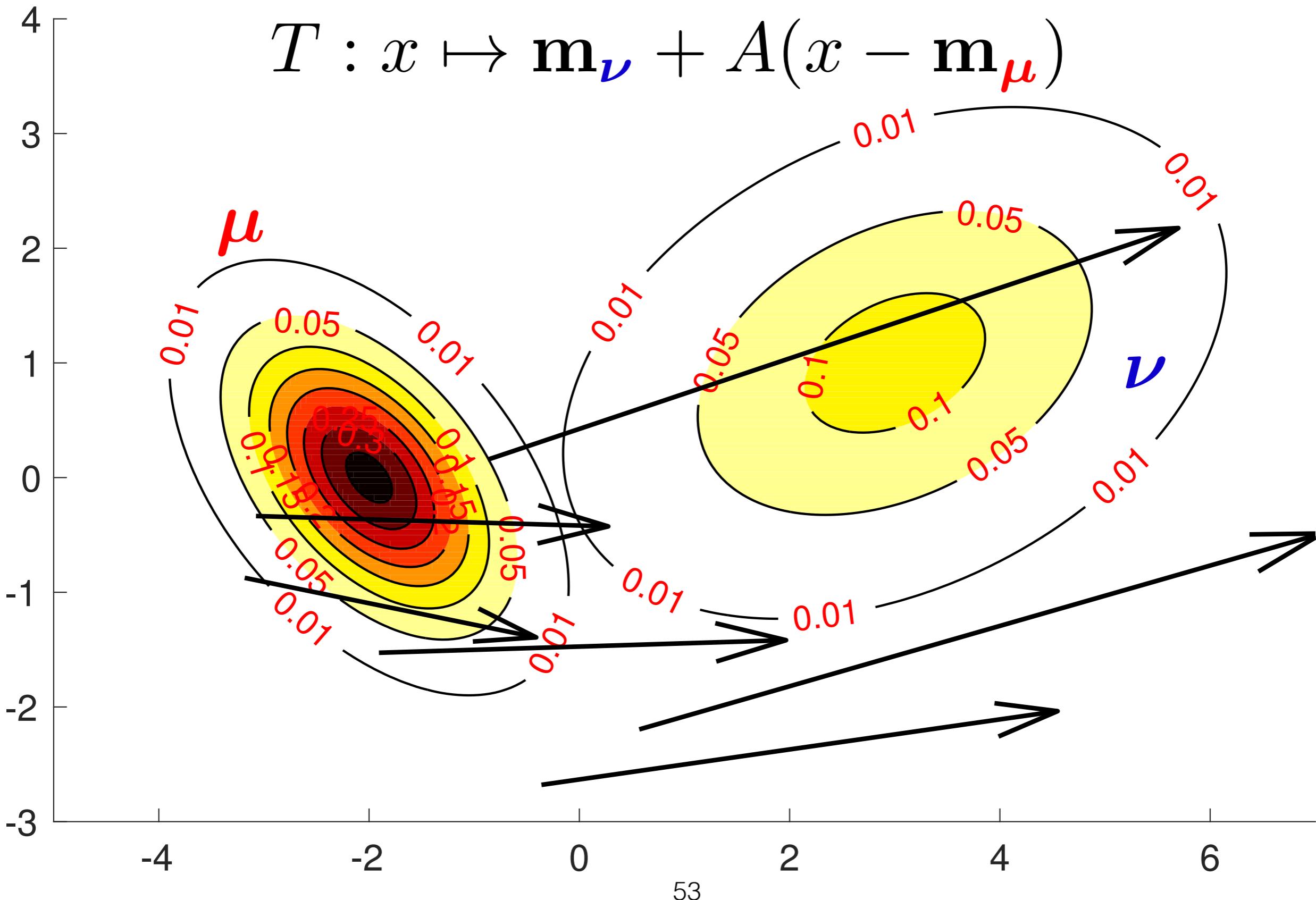
Remark. If $\Omega = \mathbb{R}^d$, $\textcolor{green}{c}(x, y) = \|x - y\|^2$, and $\mu = \mathcal{N}(\mathbf{m}_\mu, \Sigma_\mu)$, $\nu = \mathcal{N}(\mathbf{m}_\nu, \Sigma_\nu)$ then

$$W_2^2(\mu, \nu) = \|\mathbf{m}_\mu - \mathbf{m}_\nu\|^2 + B(\Sigma_\mu, \Sigma_\nu)^2$$

where B is the Bures distance $B = \sqrt{\text{trace}(\Sigma_\mu + \Sigma_\nu - 2(\Sigma_\mu^{1/2} \Sigma_\nu \Sigma_\mu^{1/2})^{1/2})}$.

The map $T : x \mapsto \mathbf{m}_\nu + A(x - \mathbf{m}_\mu)$ is optimal, where $A = \Sigma_\mu^{-\frac{1}{2}} \left(\Sigma_\mu^{\frac{1}{2}} \Sigma_\nu \Sigma_\mu^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_\mu^{-\frac{1}{2}}$.

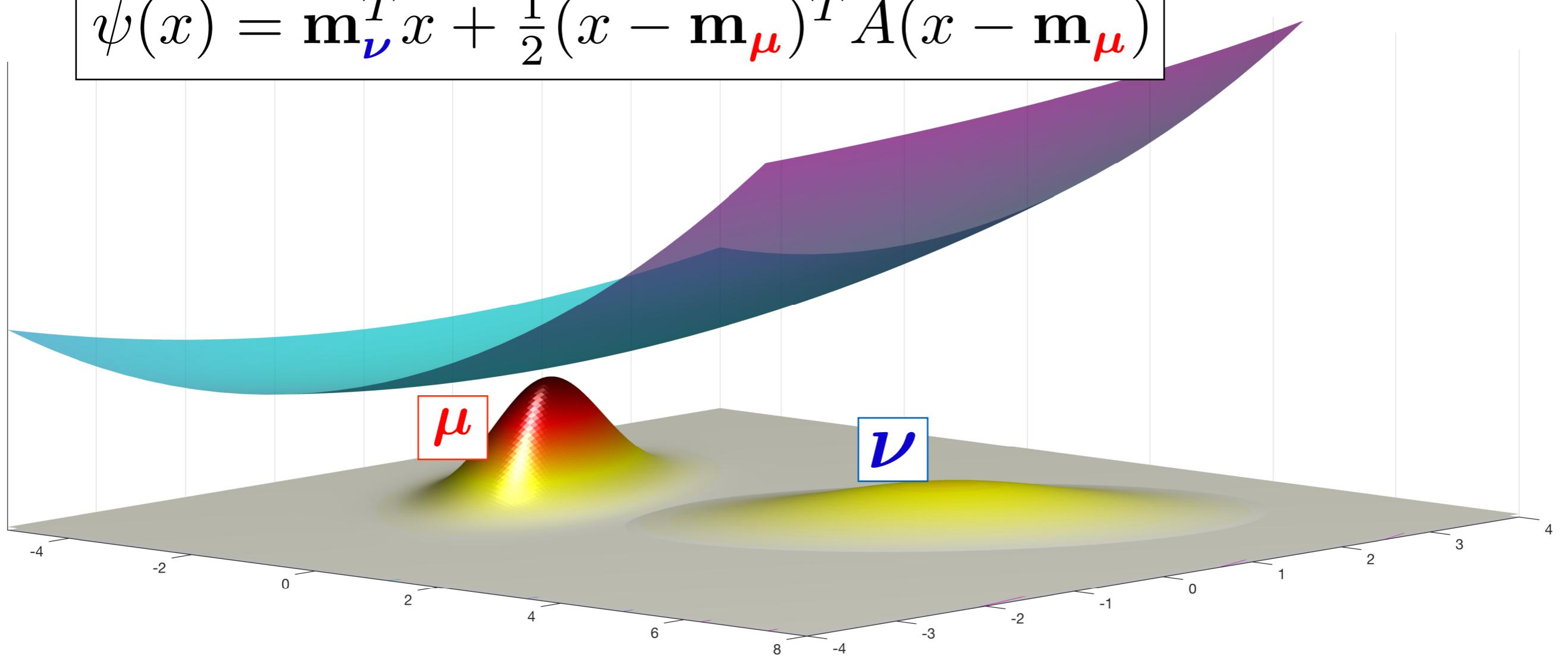
Easy (2): Gaussian Measures



Easy (2): Gaussian Measures

$$T = \nabla \psi : x \mapsto \mathbf{m}_{\nu} + A(x - \mathbf{m}_{\mu})$$

$$\psi(x) = \mathbf{m}_{\nu}^T x + \frac{1}{2}(x - \mathbf{m}_{\mu})^T A(x - \mathbf{m}_{\mu})$$



Easy (3): Elliptical Distributions

$$T = \nabla \psi : x \mapsto \mathbf{m}_{\nu} + A(x - \mathbf{m}_{\mu})$$

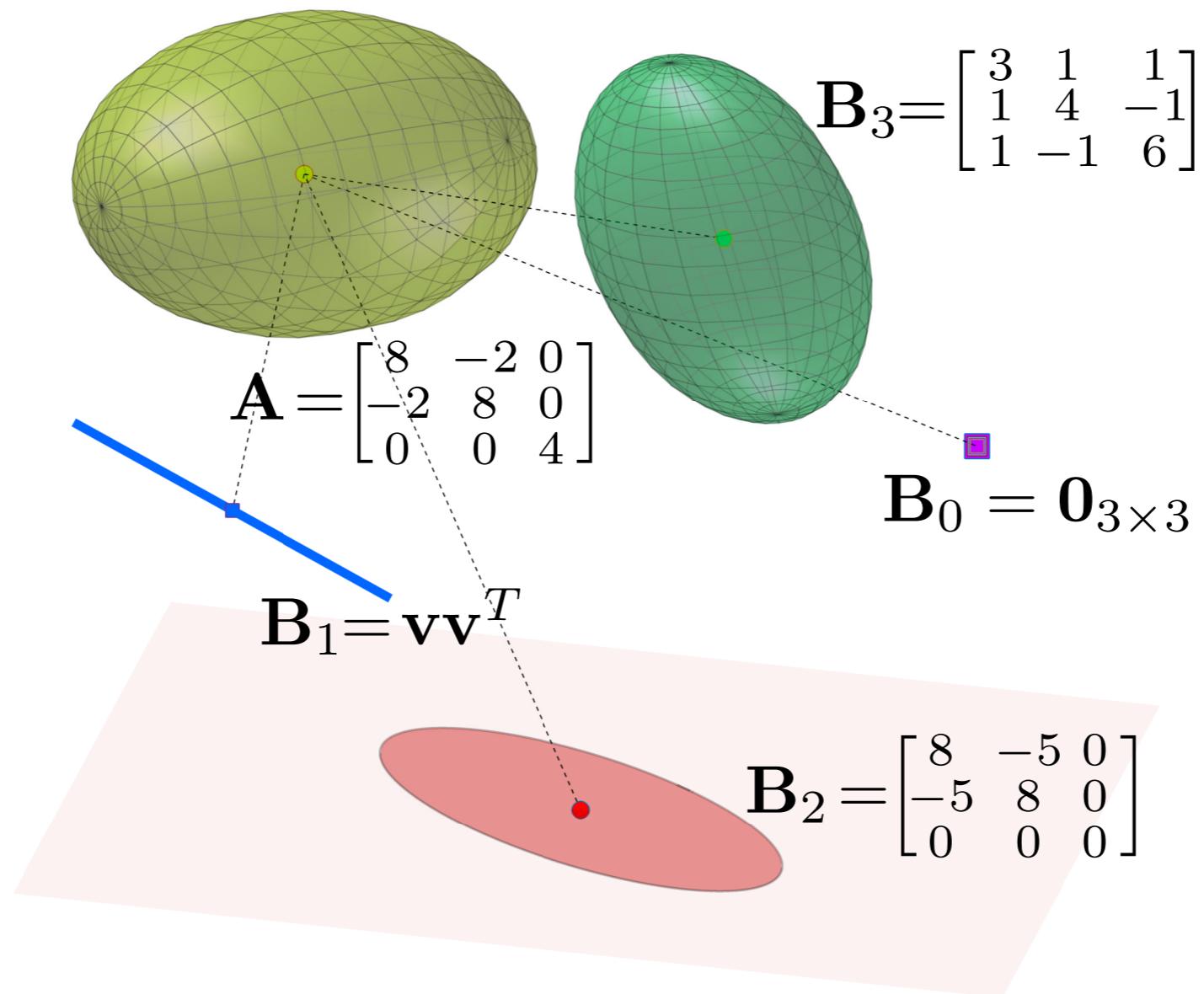
[Gelbrich'92] shows that the linear map T is also **optimal** for elliptically contoured distributions, *i.e.* distributions whose MGF are

$$\phi_X(\mathbf{t}) = \mathbb{E} \left[e^{\sqrt{-1} \mathbf{t}^T X} \right] = e^{\sqrt{-1} \mathbf{t}^T \mathbf{m}} g(\mathbf{t}^T \mathbf{C} \mathbf{t})$$

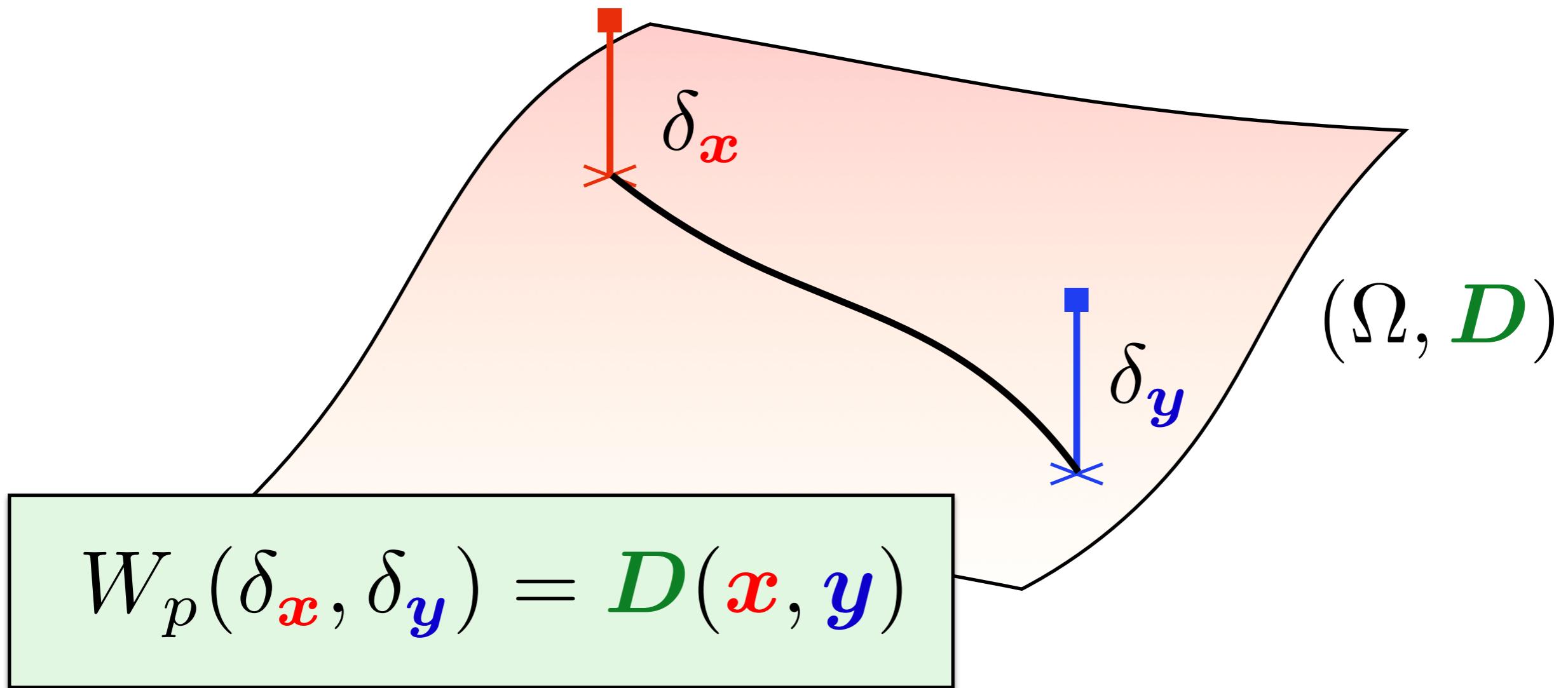
g of positive type.

Same formula applies, but variance is a factor (depends on g) of \mathbf{C} , hence Bures factor is scaled.

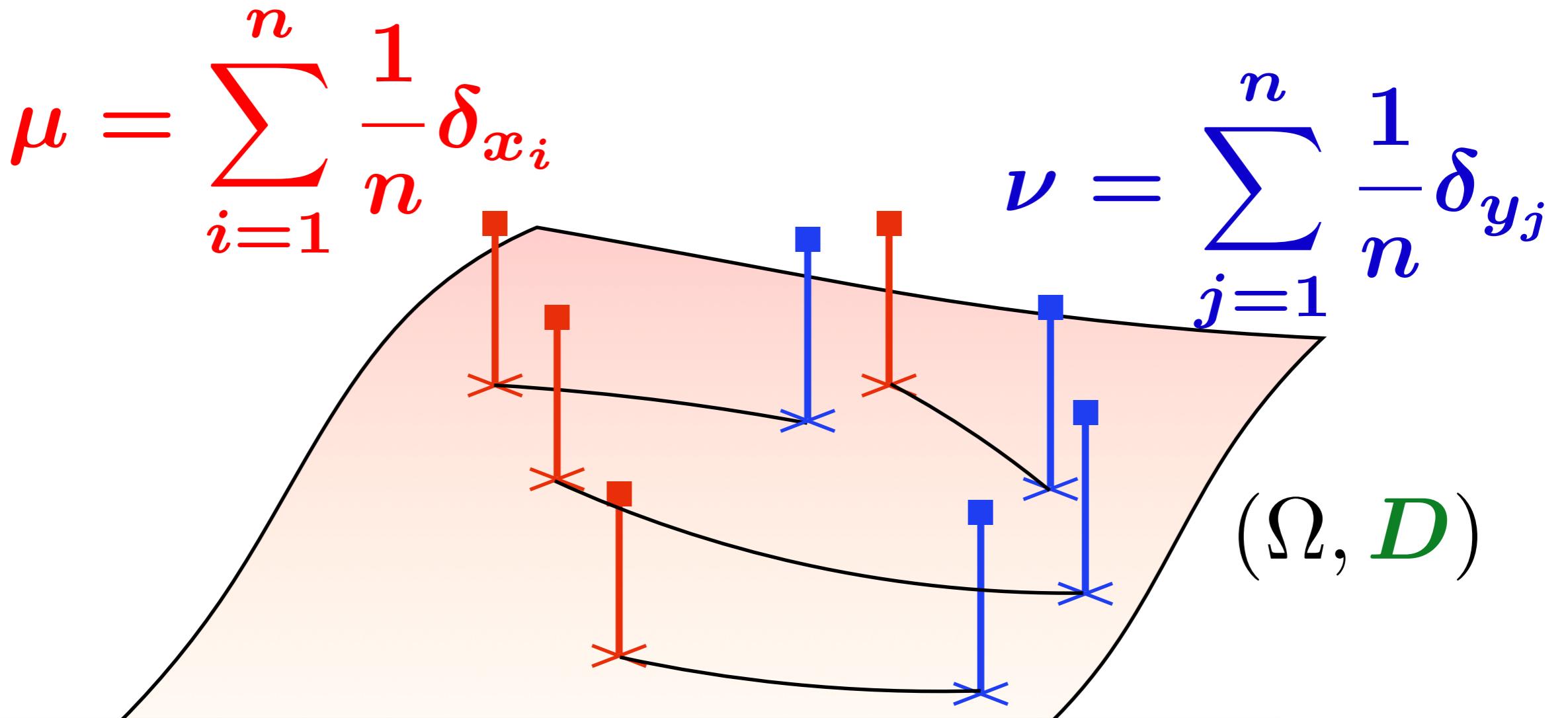
Easy (3): Uniform Ellipses



Wasserstein Between Two Diracs

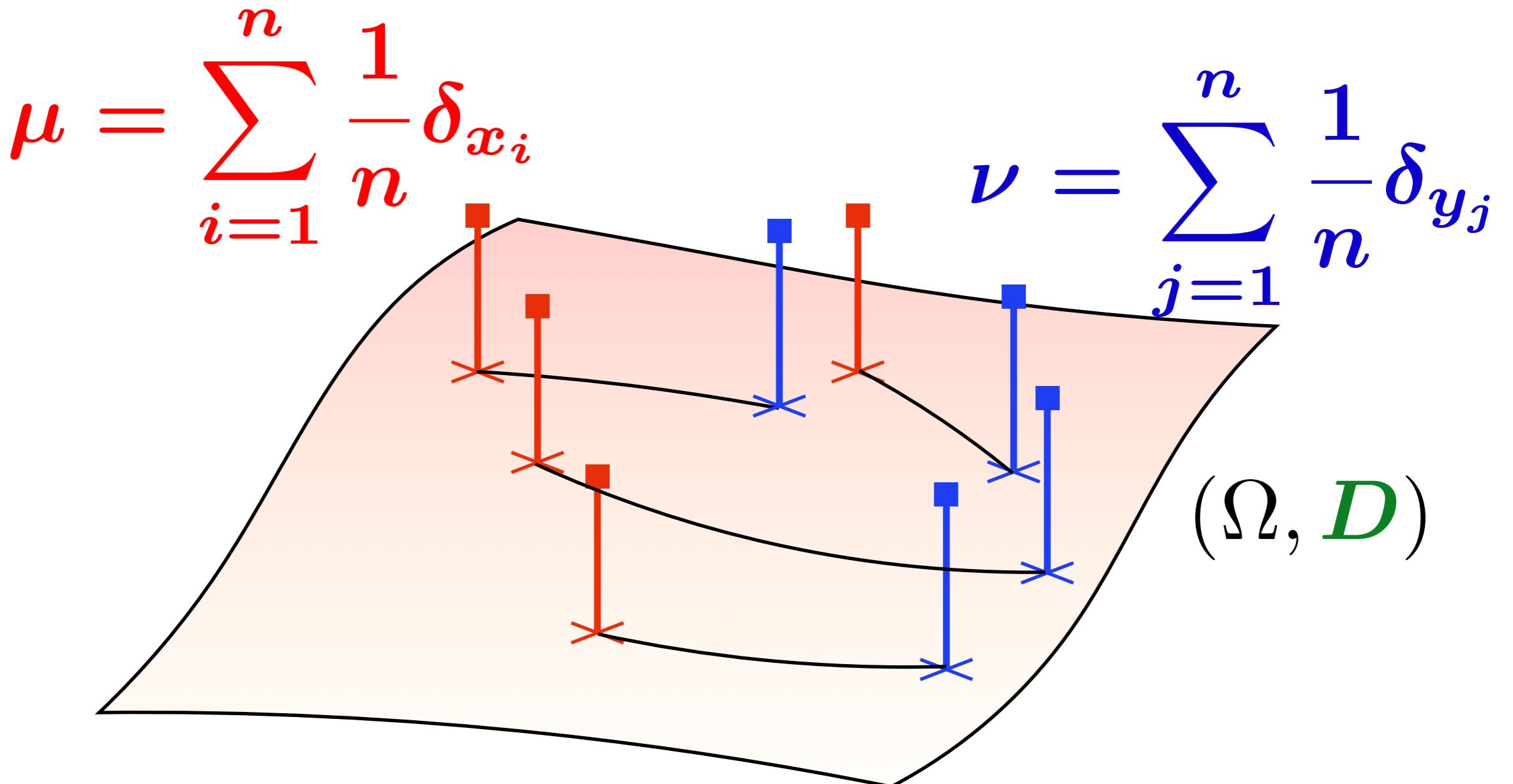


Linear Assignment \subset Wasserstein



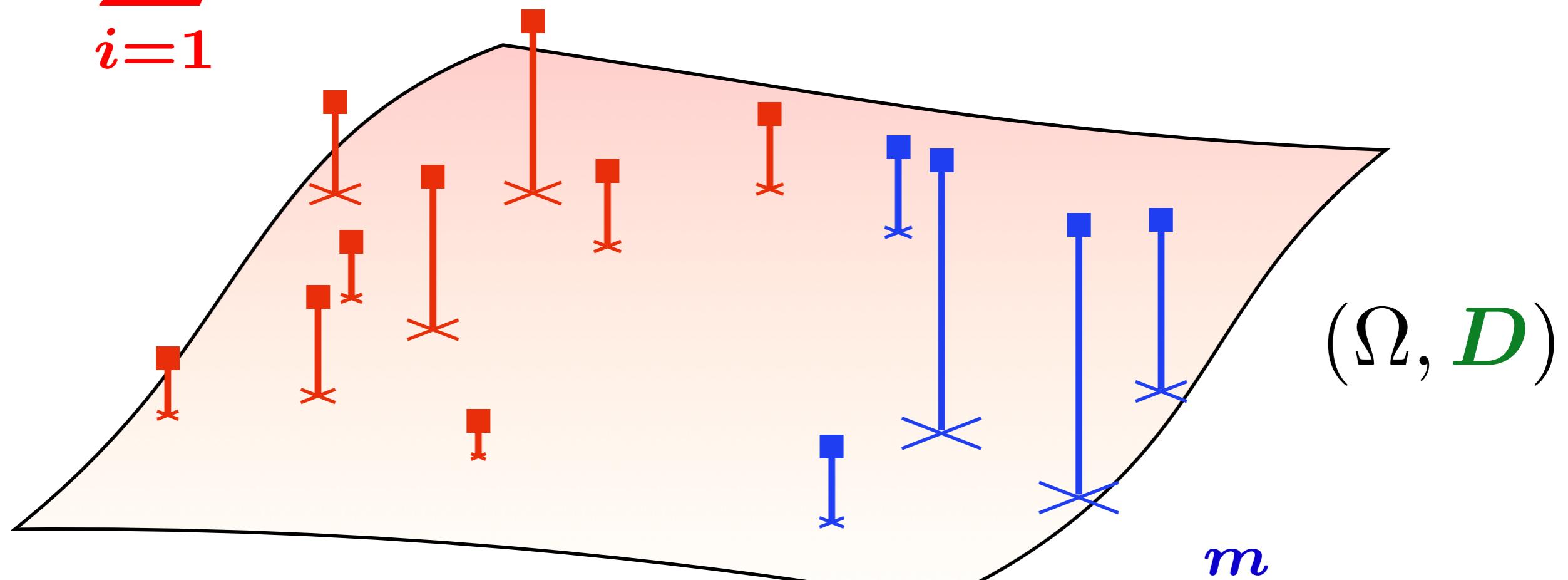
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{\sigma} \in S_n} \frac{1}{n} \sum_{i=1}^n D(\mathbf{x}_i, \mathbf{y}_{\boldsymbol{\sigma}_i})^p$$

Linear Assignment \subset Wasserstein



OT on Two Empirical Measures

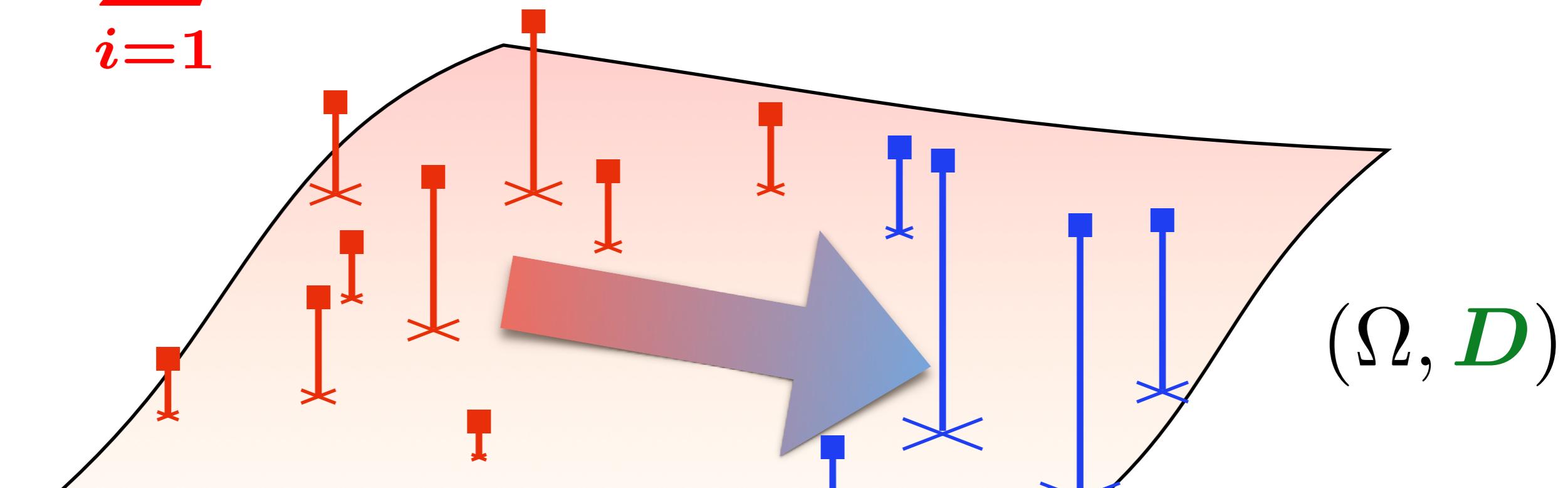
$$\mu = \sum_{i=1}^n a_i \delta_{x_i}$$



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

OT on Two Empirical Measures

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



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

Wasserstein on Empirical Measures

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

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

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

Def. Optimal Transport Problem

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

Dual Kantorovich Problem

$$W_p^p(\mu, \nu) = \min_{\substack{\mathbf{P} \in \mathbb{R}_+^{n \times m} \\ \mathbf{P} \mathbf{1}_m = \mathbf{a}, \mathbf{P}^T \mathbf{1}_n = \mathbf{b}}} \langle \mathbf{P}, M_{\mathbf{XY}} \rangle$$

Dual Kantorovich Problem

$$W_p^p(\mu, \nu) = \min_{\substack{\mathbf{P} \in \mathbb{R}_+^{n \times m} \\ \mathbf{P} \mathbf{1}_m = \mathbf{a}, \mathbf{P}^T \mathbf{1}_n = \mathbf{b}}} \langle \mathbf{P}, M_{\mathbf{XY}} \rangle$$

Def. Dual OT problem

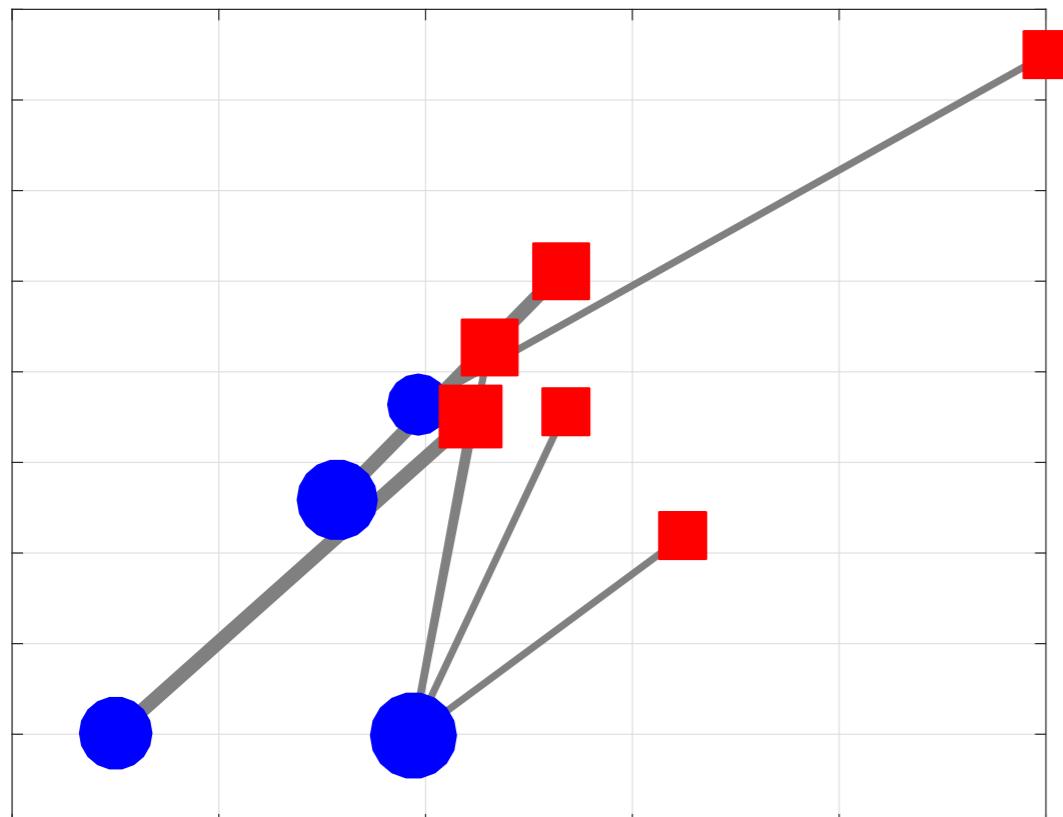
$$W_p^p(\mu, \nu) = \max_{\substack{\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(x_i, y_j)^p}} \alpha^T \mathbf{a} + \beta^T \mathbf{b}$$

Dual Kantorovich Problem

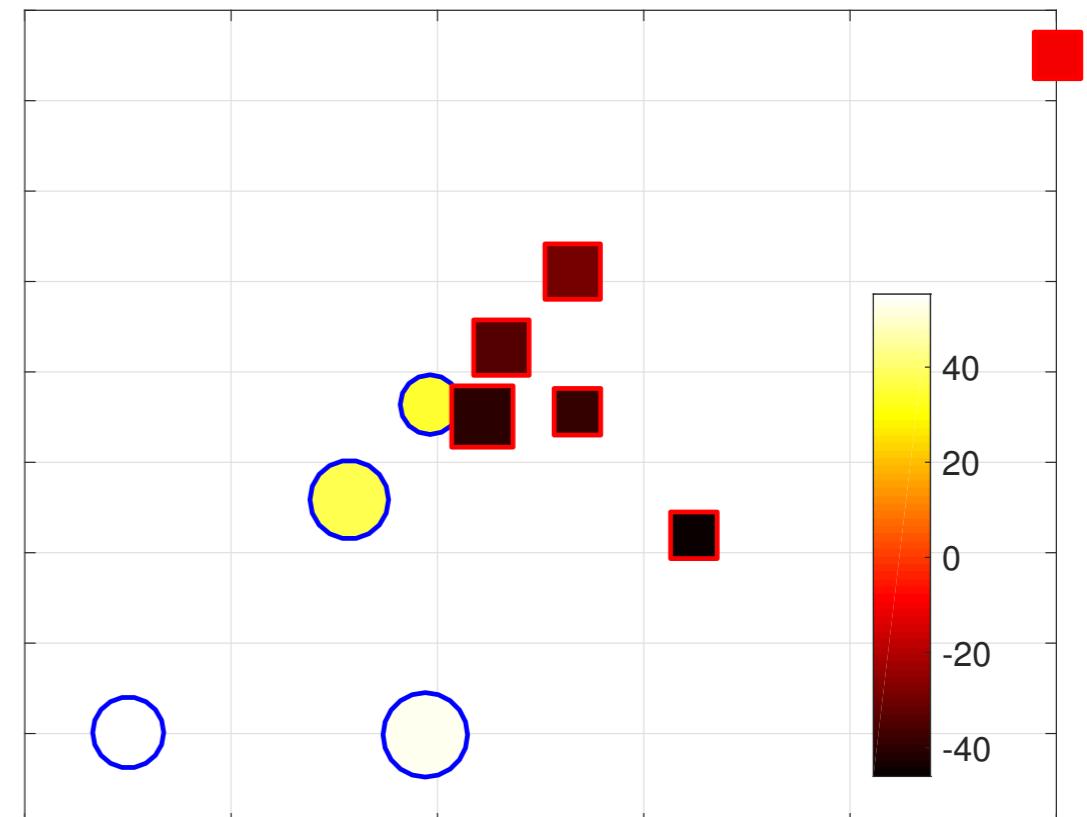
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\substack{\mathbf{P} \in \mathbb{R}_+^{n \times m} \\ \mathbf{P} \mathbf{1}_m = \mathbf{a}, \mathbf{P}^T \mathbf{1}_n = \mathbf{b}}} \langle \mathbf{P}, M_{\mathbf{XY}} \rangle$$

Def. Dual OT problem

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(\mathbf{x}_i, \mathbf{y}_j)^p}} \alpha^T \mathbf{a} + \beta^T \mathbf{b}$$



62

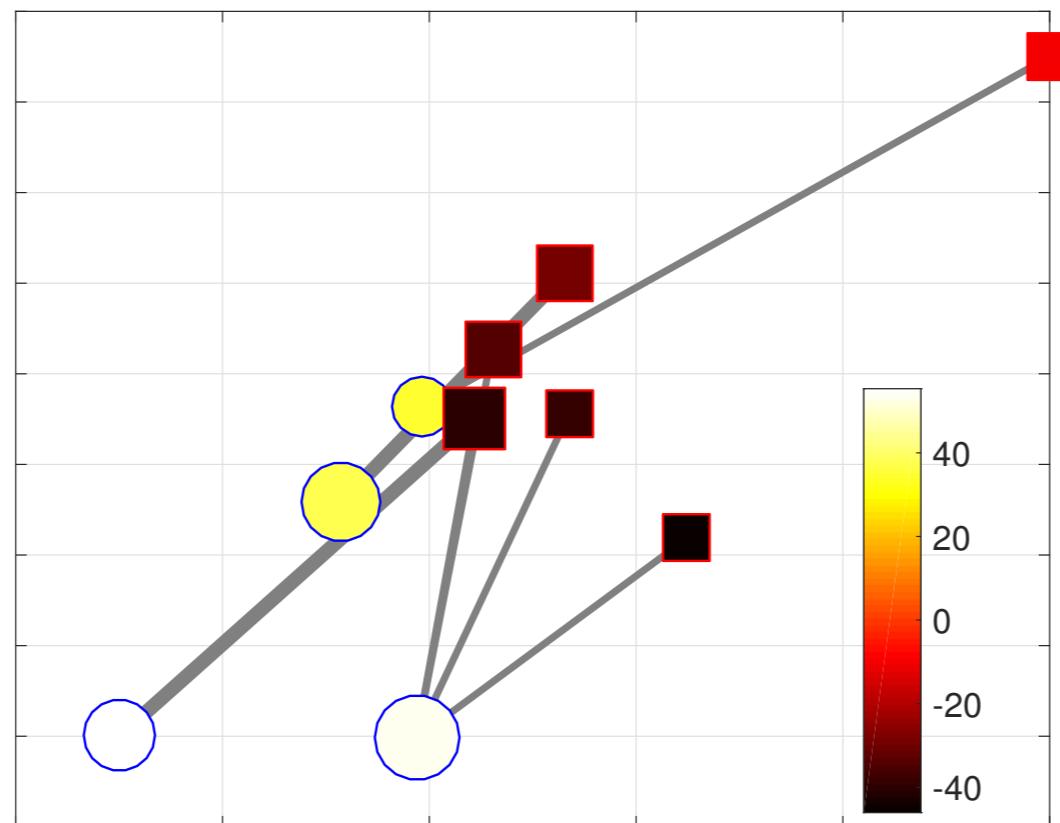


Dual Kantorovich Problem

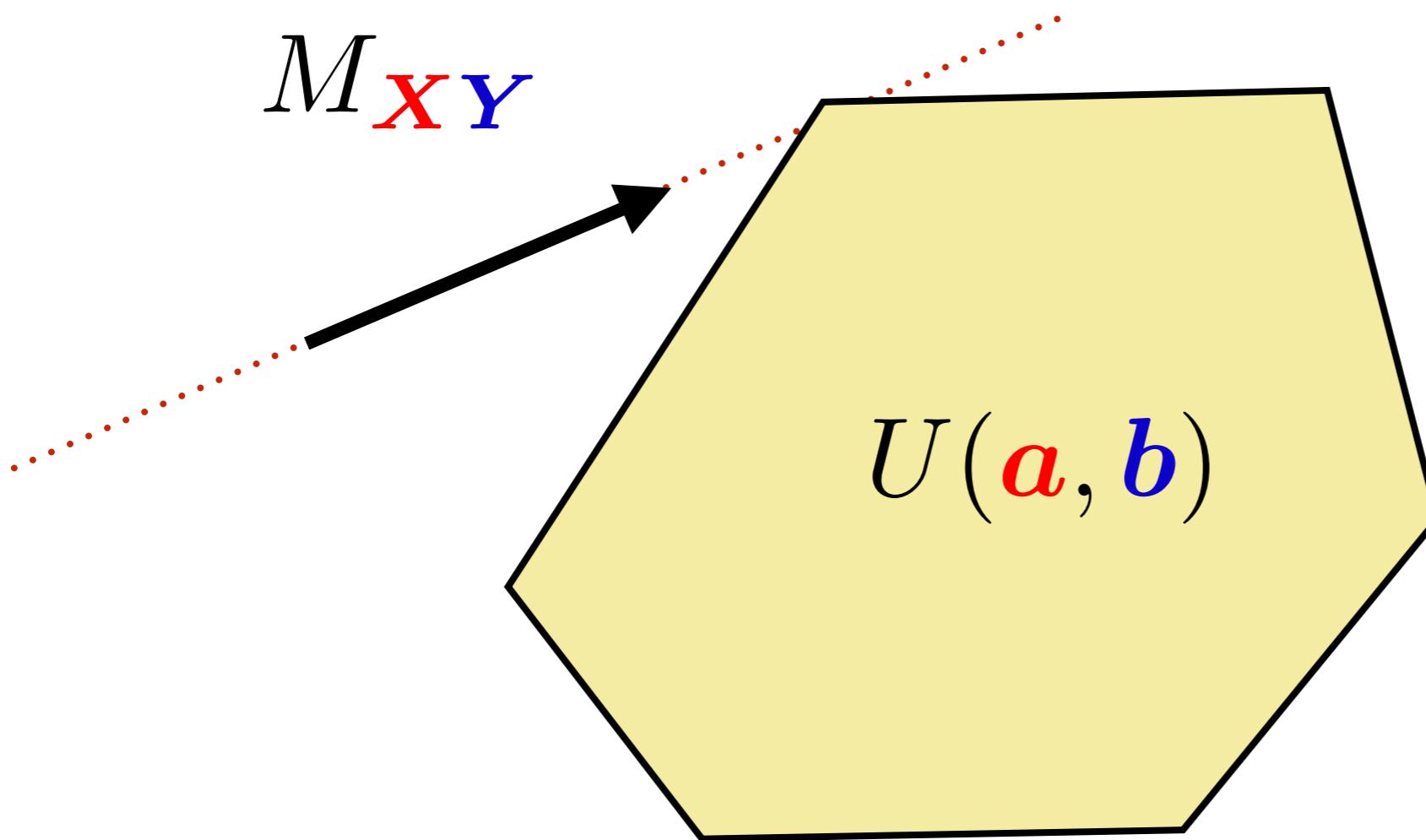
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\begin{array}{c} \boldsymbol{P} \in \mathbb{R}_+^{n \times m} \\ \boldsymbol{P} \mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T \mathbf{1}_n = \boldsymbol{b} \end{array}} \langle \boldsymbol{P}, M_{\mathbf{XY}} \rangle$$

Def. Dual OT problem

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\begin{array}{c} \boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(\mathbf{x}_i, \mathbf{y}_j)^p \end{array}} \alpha^T \boldsymbol{a} + \beta^T \boldsymbol{b}$$



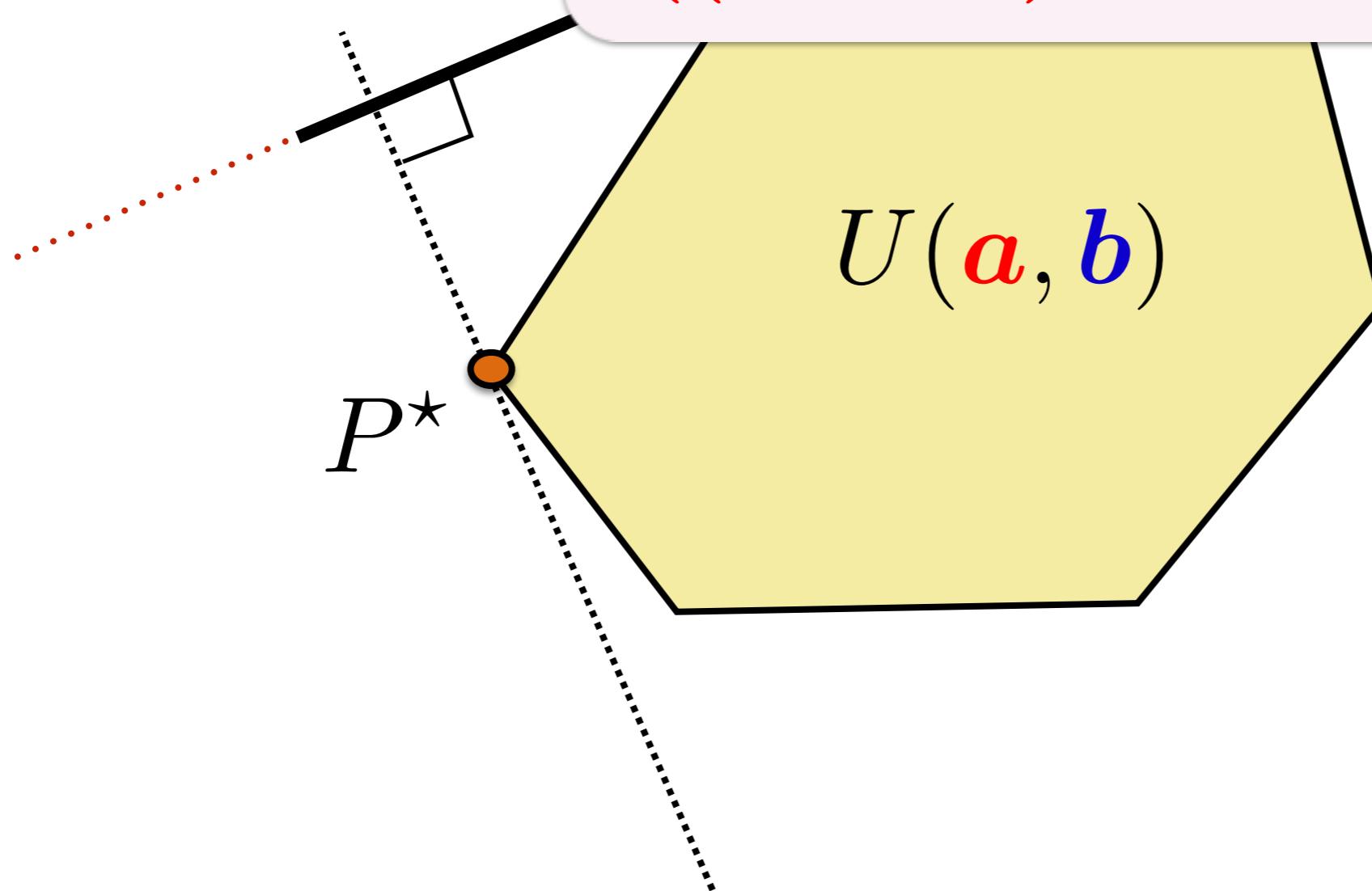
Solving the OT Problem



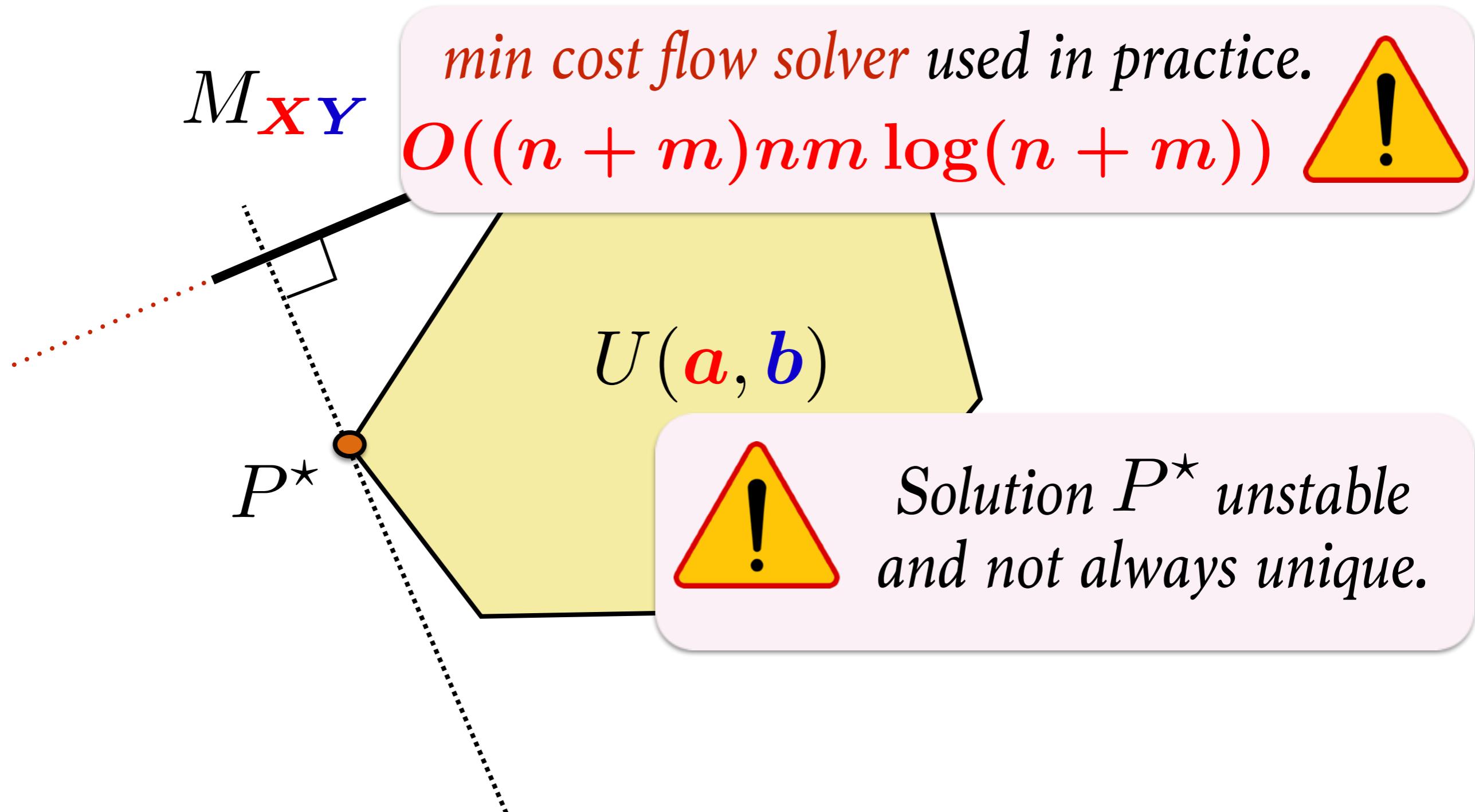
Solving the OT Problem

M_{XY}

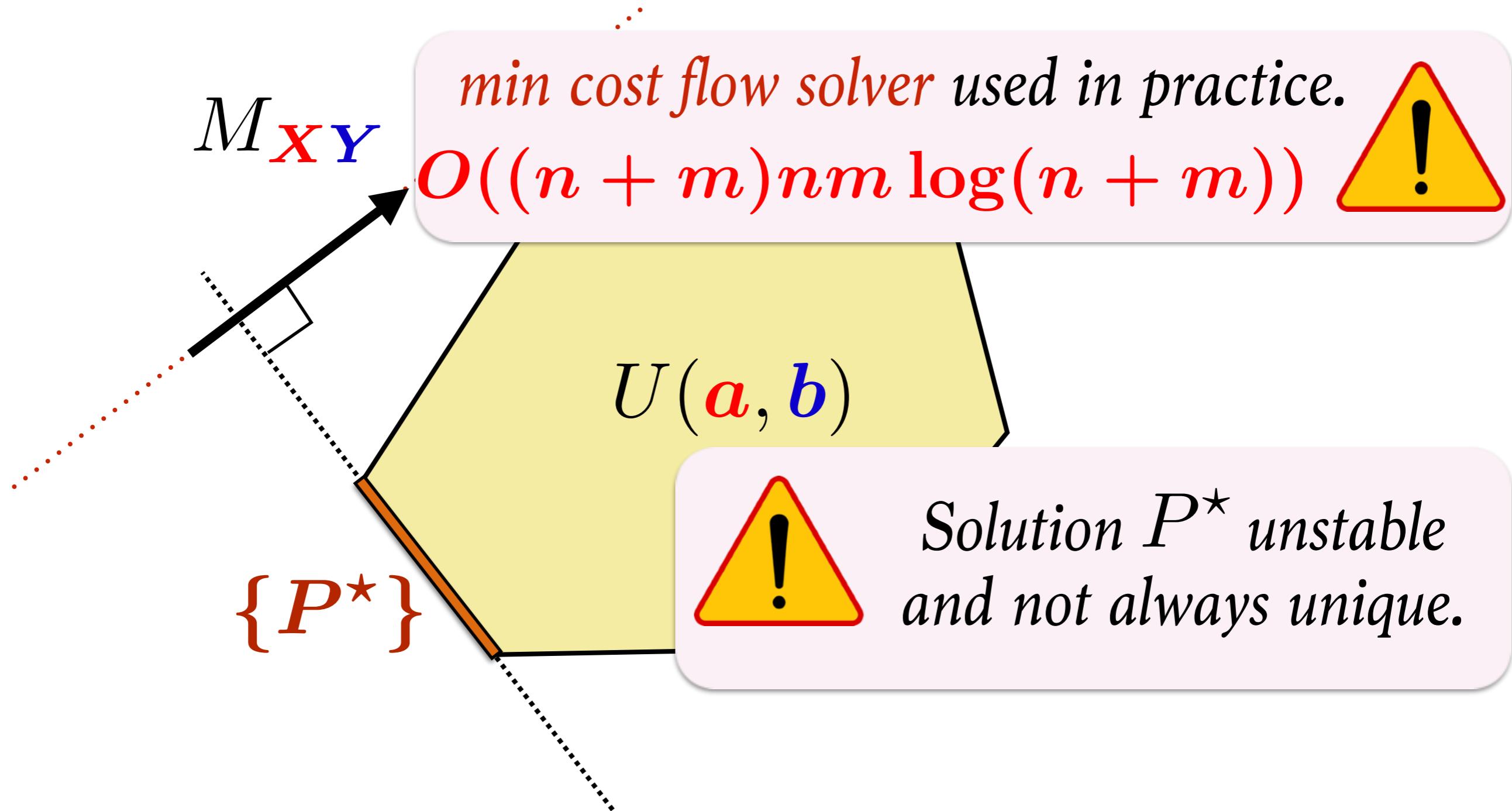
min cost flow solver used in practice.
 $O((n + m)nm \log(n + m))$



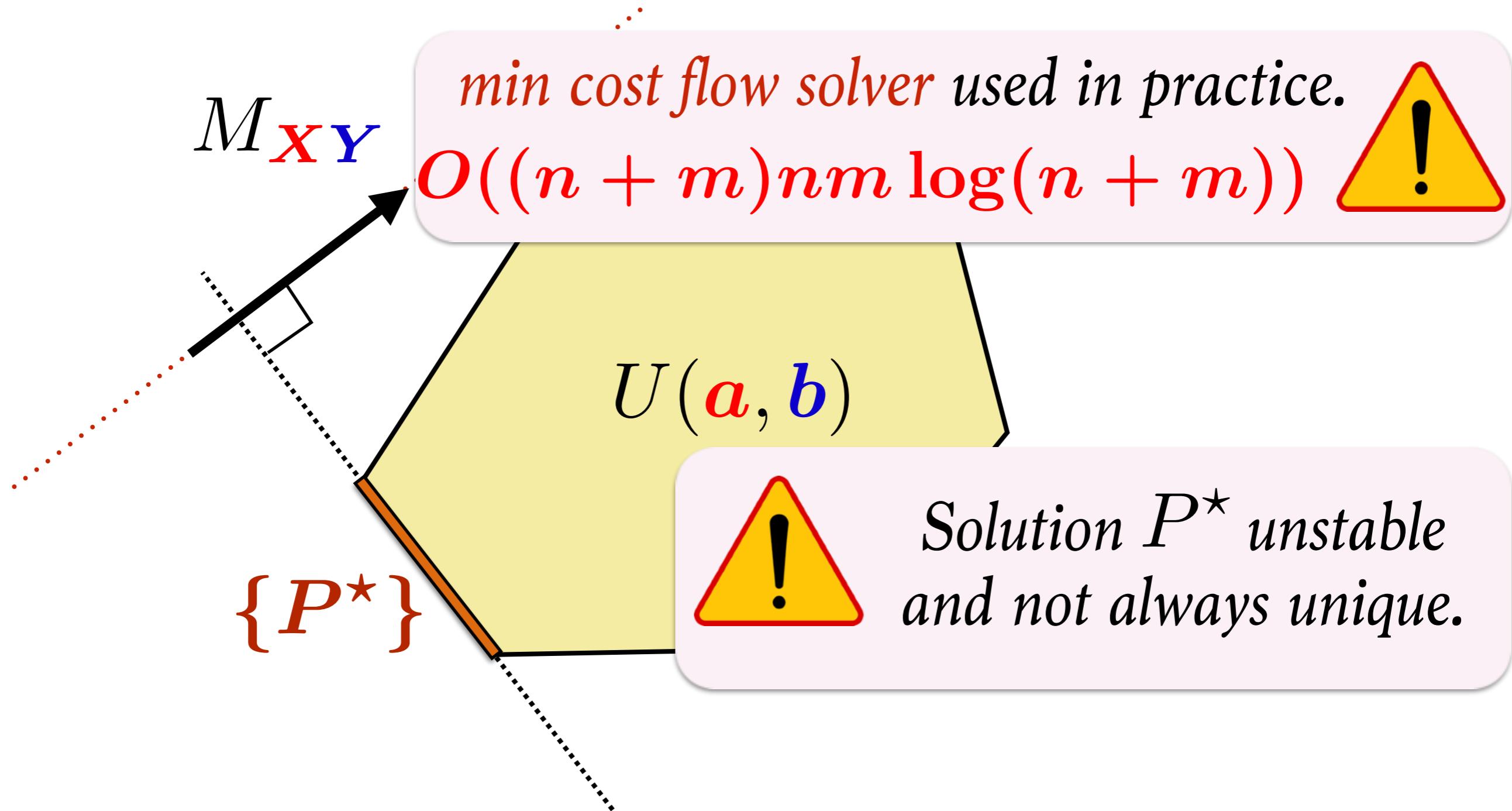
Solving the OT Problem



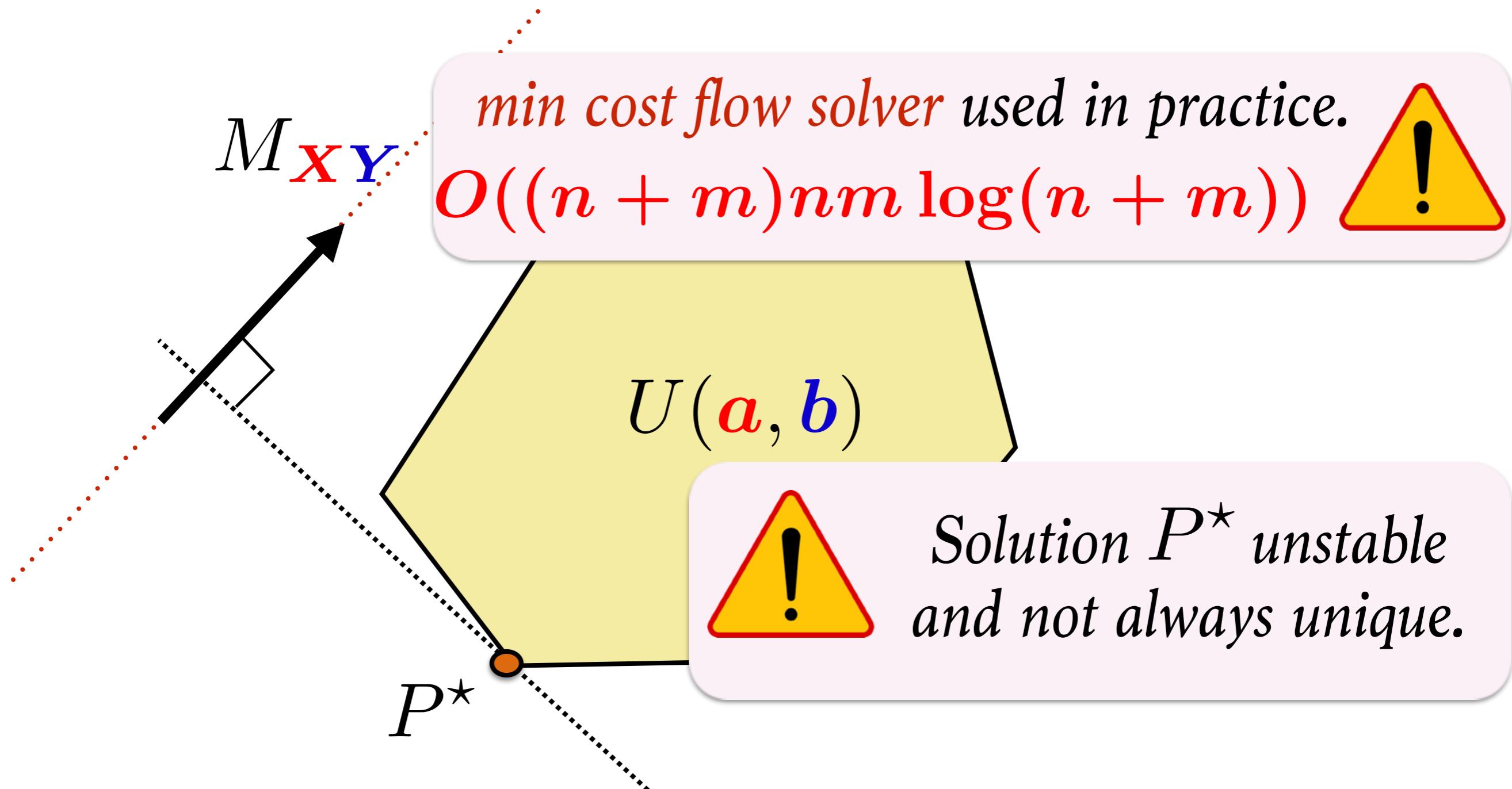
Solving the OT Problem



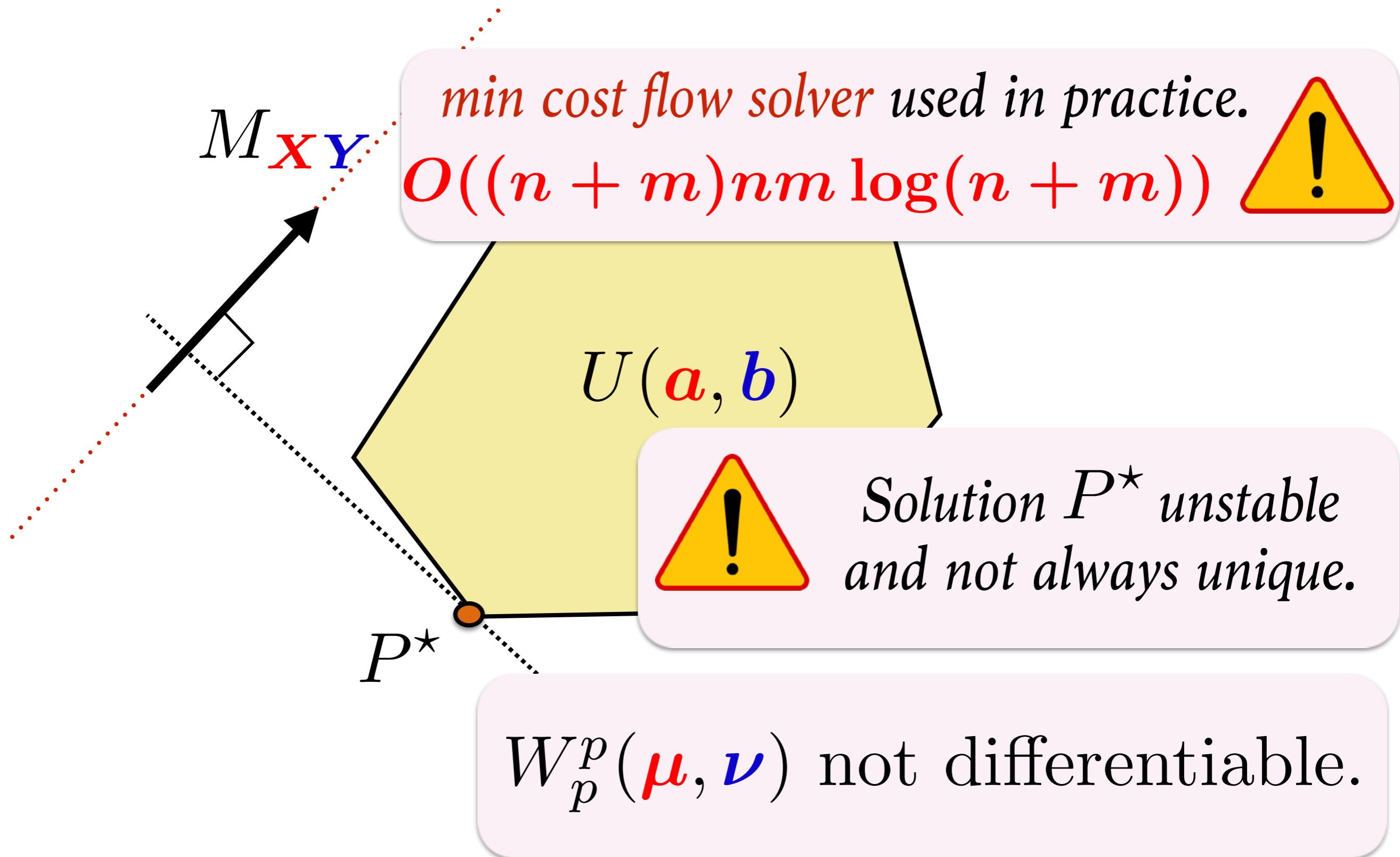
Solving the OT Problem



Solving the OT Problem



Solving the OT Problem



Discrete OT Problem

```
emd.c:6:1 <No selected symbol>
c emd.c
1 /*
2   emd.c
3
4   Last update: 3/14/98
5
6   An implementation of the Earth Movers Distance.
7   Based on the solution for the Transportation problem as described in
8   "Introduction to Mathematical Programming" by F. S. Hillier and
9   G. J. Lieberman, McGraw-Hill, 1990.
10
11  Copyright (C) 1998 Yossi Rubner
12  Computer Science Department, Stanford University
13  E-Mail: rubner@cs.stanford.edu URL: http://vision.stanford.edu/~rubner
14 */
15
16 /*#include <stdio.h>
17 #include <stdlib.h>/*
18 #include <math.h>
19
20 #include "emd.h"
21
22 #define DEBUG_LEVEL 0
23 /*
24 DEBUG_LEVEL:
25   0 = NO MESSAGES
26   1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
27   2 = PRINT THE RESULT AFTER EVERY ITERATION
28   3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
29   4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
30 */
31
32
33 #define MAX_SIG_SIZE1 (MAX_SIG_SIZE+1) /* FOR THE POSSIBLE DUMMY FEATURE */
34
35 /* NEW TYPES DEFINITION */
36
37 /* node1_t IS USED FOR SINGLE-LINKED LISTS */
38 typedef struct node1_t {
39   int i;
40   double val;
41   struct node1_t *Next;
42 } node1_t;
43
44 /* node2_t IS USED FOR DOUBLE-LINKED LISTS */
45 typedef struct node2_t {
46   int i, j;
47   double val;
48   struct node2_t *NextC;           /* NEXT COLUMN */
49   struct node2_t *NextR;           /* NEXT ROW */
50 } node2_t;
51
52
53
54 /* GLOBAL VARIABLE DECLARATION */
55 static int _n1, _n2;                  /* SIGNATURES SIZES */
56 static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1]; /* THE COST MATRIX */
57 static node2_t _X[MAX_SIG_SIZE1*2];    /* THE BASIC VARIABLES VECTOR */
58 /* VECTORS TO HANDLE THE TRANSPORTATION PROBLEM */
```

Discrete OT Problem

```
emd.c:6:1 <No selected symbol>
c emd.c
1 /*
2   emd.c
3
4   Last update: 3/14/98
5
6   An implementation of the Earth Movers Distance.
7   Based on the solution for the Transportation problem as described in
8   "Introduction to Mathematical Programming" by F. S. Hillier and
9   G. J. Lieberman, McGraw-Hill, 1990.
10
11  Copyright (C) 1998 Yossi Rubner
12  Computer Science Department, Stanford University
13  E-Mail: rubner@cs.stanford.edu URL: http://vision.stanford.edu/~rubner
14 */
15
16 /*#include <stdio.h>
17 #include <stdlib.h>/*
18 #include <math.h>
19
20 #include "emd.h"
21
22 #define DEBUG_LEVEL 0
23 /*
24 DEBUG_LEVEL:
25   0 = NO MESSAGES
26   1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
27   2 = PRINT THE RESULT AFTER EVERY ITERATION
28   3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
29   4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
30 */
31
32
33 #define MAX_SIG_SIZE1 (MAX_SIG_SIZE+1) /* FOR THE POSSIBLE DUMMY FEATURE */
34
35 /* NEW TYPES DEFINITION */
36
37 /* node1_t IS USED FOR SINGLE-LINKED LISTS */
38 typedef struct node1_t {
39   int i;
40   double val;
41   struct node1_t *Next;
42 } node1_t;
43
44 /* node2_t IS USED FOR DOUBLE-LINKED LISTS */
45 typedef struct node2_t {
46   int i, j;
47   double val;
48   struct node2_t *NextC;           /* NEXT COLUMN */
49   struct node2_t *NextR;           /* NEXT ROW */
50 } node2_t;
51
52
53
54 /* GLOBAL VARIABLE DECLARATION */
55 static int _n1, _n2;                  /* SIGNATURES SIZES */
56 static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1];/* THE COST MATRIX */
57 static node2_t _X[MAX_SIG_SIZE1*2];      /* THE BASIC VARIABLES VECTOR */
58 /* VECTORS TO HANDLE THE TRANSPORTATION PROBLEM */
```

Discrete OT Problem

```
emd.c:6:1 <No selected symbol>
1  /*
2   emd.c
3
4   Last update: 3/14/98
5
6   An implementation of the Earth Movers Distance.
7   Based on the solution for the Transportation problem as described in
8   "Introduction to Mathematical Programming" by F. S. Hillier and
9   G. J. Lieberman, McGraw-Hill, 1990.
10
11  Copyright (C) 1998 Yossi Rubner
12  Computer Science Department, Stanford University
13  E-Mail: rubner@cs.stanford.edu URL: http://vision.stanford.edu/~rubner
14 */
15
16 /*#include <stdio.h>
17 #include <stdlib.h>/*
18 #include <math.h>
19
20 #include "emd.h"
21
22 #define DEBUG_LEVEL 0
23 /*
24 DEBUG_LEVEL:
25 0 = NO MESSAGES
26 1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
27 2 = PRINT THE RESULT AFTER EVERY ITERATION
28 3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
29 4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
30 */
31
32
33 #define MAX_SIG_SIZE1 (MAX_SIG_SIZE+1) /* FOR THE POSSIBLE DUMMY FEATURE */
34
35 /* NEW TYPES DEFINITION */
36
37 /* node1_t IS USED FOR SINGLE-LINKED LISTS */
38 typedef struct node1_t {
39     int i;
40     double val;
41     struct node1_t *Next;
42 } node1_t;
43
44 /* node2_t IS USED FOR DOUBLE-LINKED LISTS */
45 typedef struct node2_t {
46     int i, j;
47     double val;
48     struct node2_t *NextC;           /* NEXT COLUMN */
49     struct node2_t *NextR;           /* NEXT ROW */
50 } node2_t;
51
52
53
54 /* GLOBAL VARIABLE DECLARATION */
55 static int _n1, _n2;                  /* SIGNATURES SIZES */
56 static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1];/* THE COST MATRIX */
57 static node2_t _X[MAX_SIG_SIZE1*2];    /* THE BASIC VARIABLES VECTOR */
58 /* VECTORS TO HANDLE THE TRANSPORTATION PROBLEM */
```

ice.
))



3. Computing OT for data sciences

- On an important misunderstanding
- Regularizations
- Entropic regularization
- Subspace based regularization

What matters for practitioners?

i.i.d samples $\textcolor{red}{x}_1, \dots, \textcolor{red}{x}_n \sim \mu$, $\textcolor{blue}{y}_1, \dots, \textcolor{blue}{y}_m \sim \nu$,

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\textcolor{red}{x}_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\textcolor{blue}{y}_j}$$

Computational properties

Compute/approximate $\textcolor{green}{W}_p(\hat{\mu}_n, \hat{\nu}_m)$?

Statistical properties

$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_m)|] \leq f(n, m)$?

What matters for practitioners?

i.i.d samples $\mathbf{x}_1, \dots, \mathbf{x}_n \sim \boldsymbol{\mu}$, $\mathbf{y}_1, \dots, \mathbf{y}_m \sim \boldsymbol{\nu}$,

$$\hat{\boldsymbol{\mu}}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\mathbf{x}_i}, \hat{\boldsymbol{\nu}}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\mathbf{y}_j}$$

Computational properties



$$O((n + m)nm \log(n + m))$$

Statistical properties

$$\mathbb{E} [|W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) - W_p(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\nu}}_m)|] \leq f(n, m)?$$

Sample Complexity



If $\Omega = \mathbb{R}^d, d > 3$

$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_n)|] = O(n^{-1/d})$$

- [Dudley'69][Dereich+'11][Fournier+'13] & others..
- [Weed/Bach'17]: sharper results when measures' support has “low effective d ” in metric spaces
- [Weed/Berthet'19] for smooth densities
- Lower bounds: optimal quantization error.

From theory to practice ?

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{x_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{y_j}$$

Computational properties



$$O((n + m)nm \log(n + m))$$

Statistical properties



$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_m)|] = O(n^{-1/d})$$

From theory to practice ?

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{x_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{y_j}$$

Computational properties 

$$O((n + m)nm \log(n + m))$$

Statistical properties 

$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_n)|] = O(n^{-1/d})$$

In data sciences we ***must*** regularize the problem to improve on either (or both) aspects!

Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

- RKHS for potentials [GCBP'16], dualize/smooth indicator constraint

Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

- RKHS for potentials [GCBP'16], dualize/smooth indicator constraint

$$W_1(\mu, \nu) = \sup_{\varphi \text{ 1-Lipschitz}} \int \varphi(d\mu - d\nu).$$

- Parameterize functions using ReLU Deep net with bounded weights [Arjovsky+'17] or use Wavelet decompositions [Shirdonkhar+'08] for low d .

Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

- Change cost function: threshold metric [Pele+'09], use geodesic distance on graphs [Beckman'52] [Lin+'07], [Solomon+'14], simplifies the LP.

Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

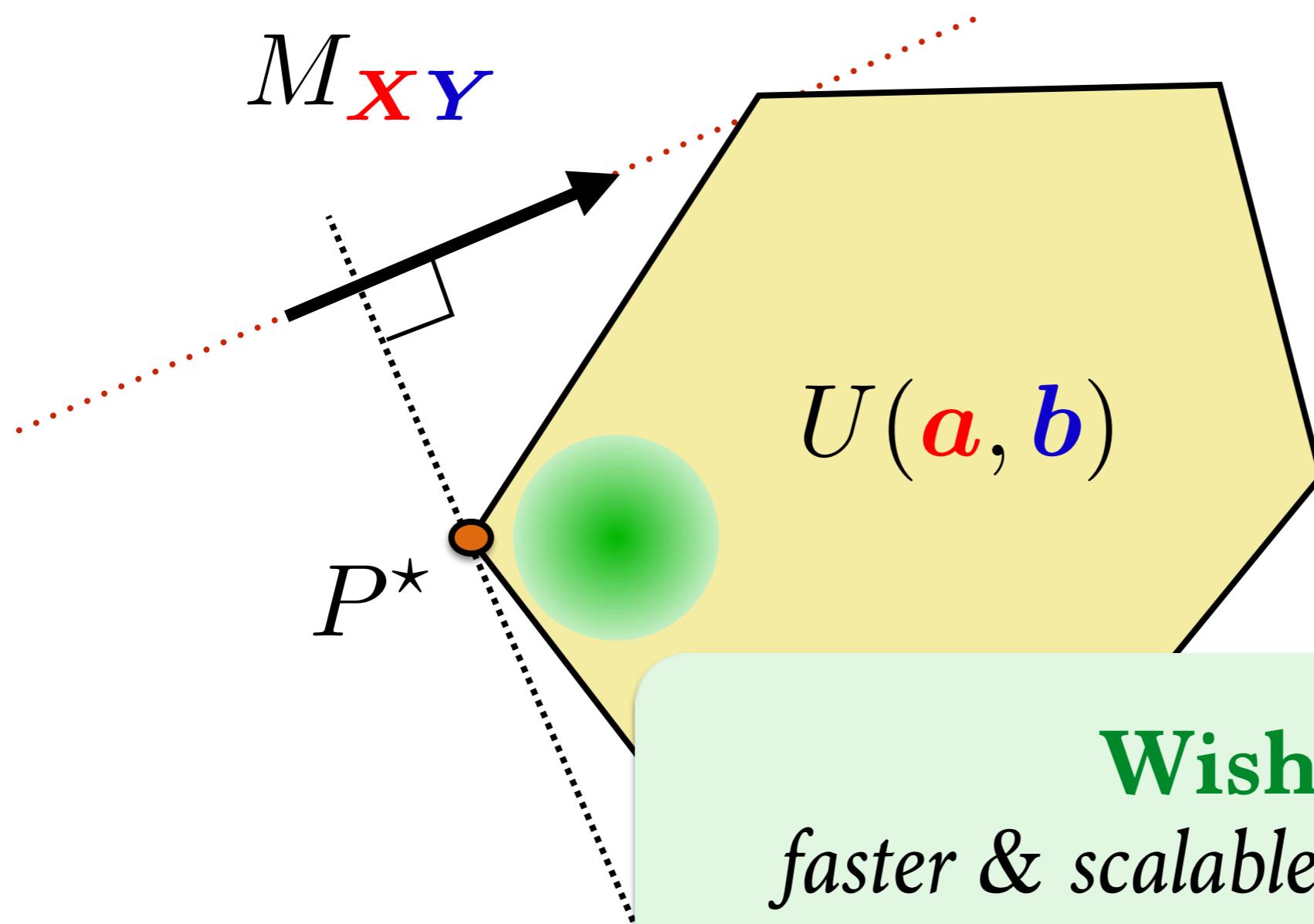
- Change cost function: threshold metric [Pele+'09], use geodesic distance on graphs [Beckman'52] [Lin+'07], [Solomon+'14], simplifies the LP.
- Change measures: Quantize first [Canas+'12]; use Gaussians [Gelbrich'92]; projections on random lines [Rabin+'11] & k -dimensional subspaces [Paty+'19][Weed+'19].

Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint c(x, y) \mathbf{P}(dx, dy).$$

- Change cost function: threshold metric [Pele+'09], use geodesic distance on graphs [Beckman'52] [Lin+'07], [Solomon+'14], simplifies the LP.
- Change measures: Quantize first [Canas+'12]; use Gaussians [Gelbrich'92]; projections on random lines [Rabin+'11] & k -dimensional subspaces [Paty+'19][Weed+'19].
- Add prior on coupling, e.g. (entropic) regularization [C'13][GP'16][GCBP'16] [GCBCP'19]

Regularization on the Primal



Wishlist:

*faster & scalable, more stable,
(automatically) differentiable*

Entropic Regularization [Wilson'62]

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

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

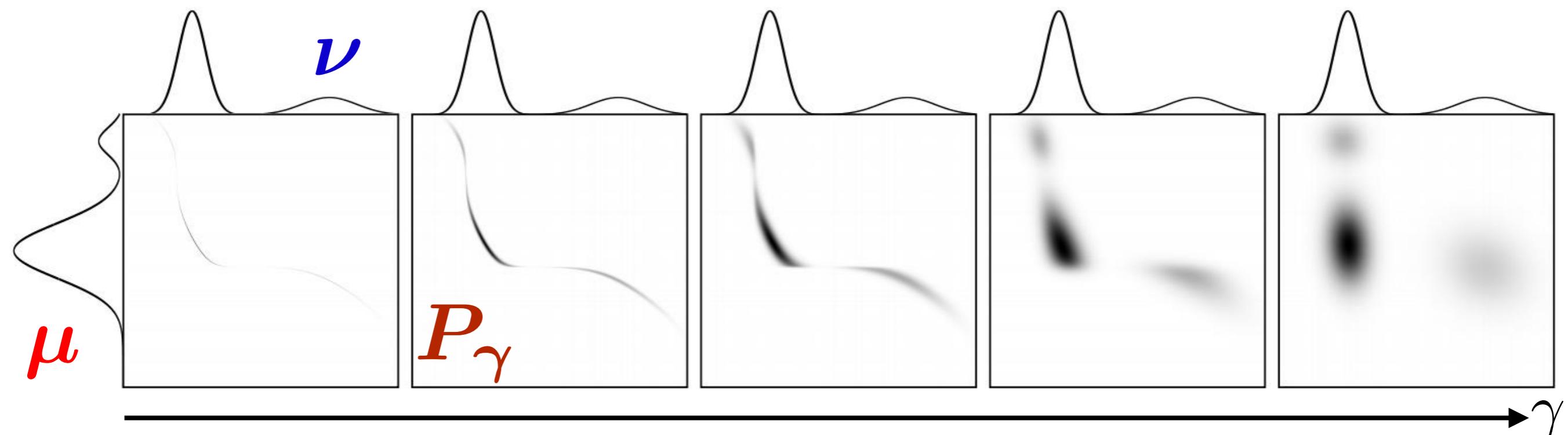
$$E(P) \stackrel{\text{def}}{=} - \sum_{i,j=1}^{nm} P_{ij} (\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

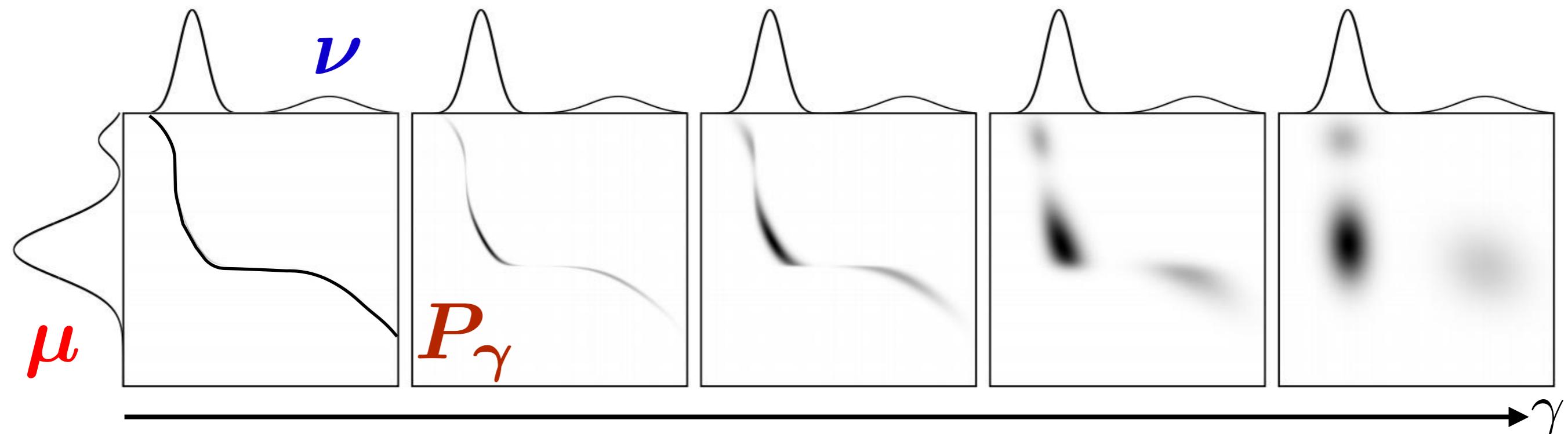


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

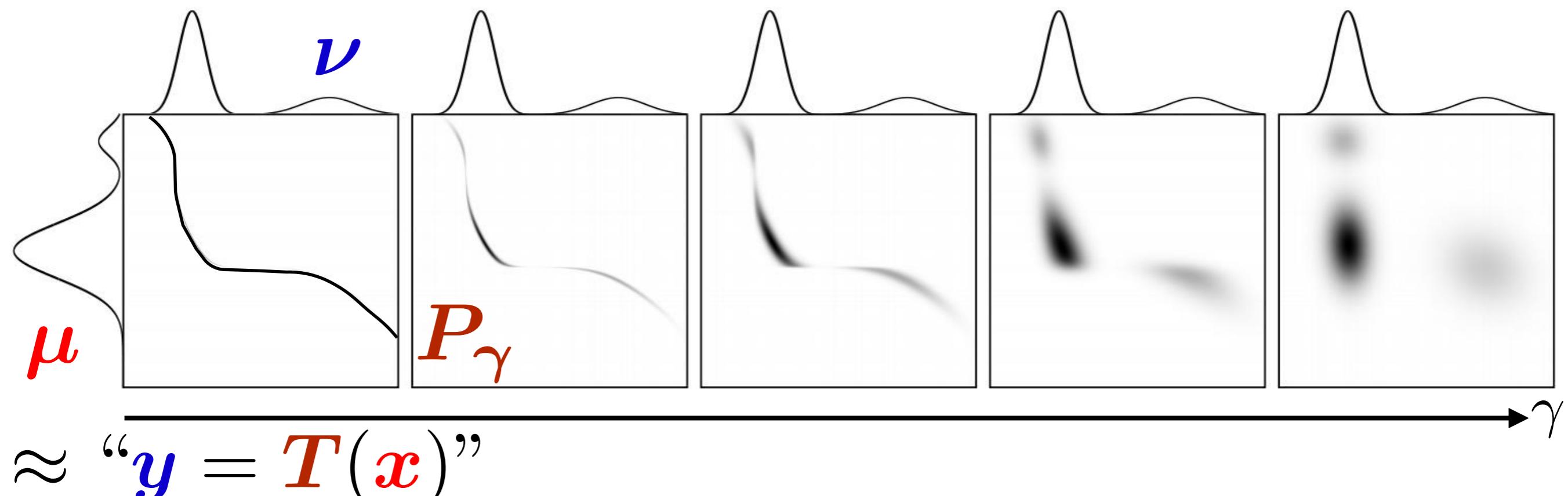


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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



Note: Unique optimal solution because of strong concavity of entropy

Fast & Scalable Algorithm

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

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

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

Fast & Scalable Algorithm

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

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

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

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

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

$$(\partial L / \partial P_{ij} = 0) \Rightarrow P_{ij} = e^{\frac{\alpha_i}{\gamma}} e^{-\frac{M_{ij}}{\gamma}} e^{\frac{\beta_j}{\gamma}} = \mathbf{u}_i K_{ij} \mathbf{v}_j$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1}_m = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \operatorname{diag}(\mathbf{u}) \mathbf{1}_n = \mathbf{b} \end{cases}$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1}_m = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \underbrace{\mathbf{K}^T \operatorname{diag}(\mathbf{u})}_{\mathbf{u}} \mathbf{1}_n = \mathbf{b} \end{cases}$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \underbrace{\operatorname{diag}(\mathbf{v}) \mathbf{1}_m}_{\mathbf{v}} = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \underbrace{\operatorname{diag}(\mathbf{u}) \mathbf{1}_n}_{\mathbf{u}} = \mathbf{b} \end{cases}$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \mathbf{v} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \mathbf{u} \end{cases} \begin{matrix} = \mathbf{a} \\ = \mathbf{b} \end{matrix}$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \mathbf{u} \odot \mathbf{K} \mathbf{v} \\ \mathbf{v} \odot \mathbf{K}^T \mathbf{u} \end{cases} \begin{matrix} = \mathbf{a} \\ = \mathbf{b} \end{matrix}$$

Fast & Scalable Algorithm

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

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

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

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v} \\ \mathbf{v} = \mathbf{b} / \mathbf{K}^T \mathbf{u} \end{cases}$$

Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

1. $\mathbf{u} = \mathbf{a}/K\mathbf{v}$
2. $\mathbf{v} = \mathbf{b}/K^T\mathbf{u}$

Fast & Scalable Algorithm

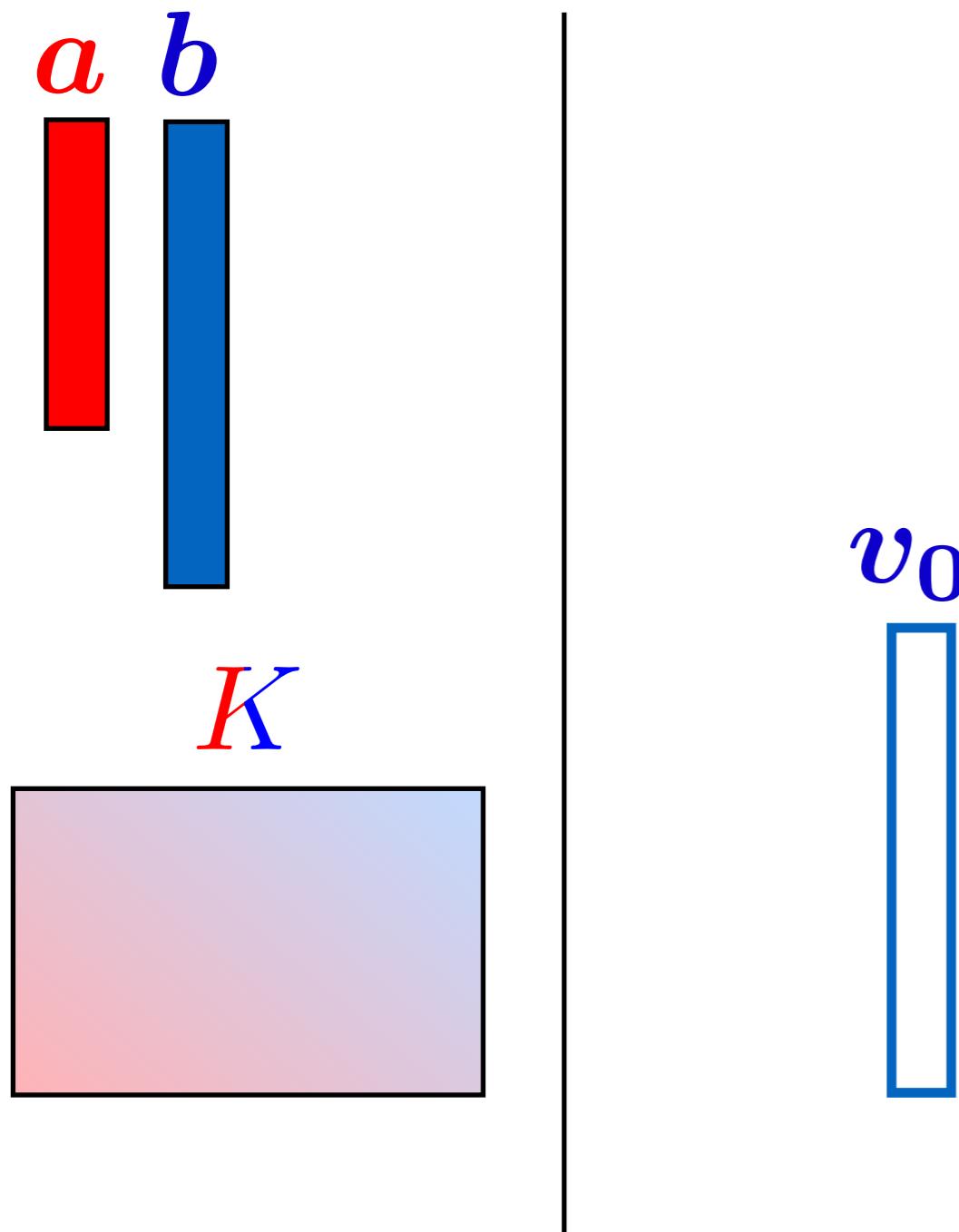
Sinkhorn's Algorithm : Repeat

1. $\mathbf{u} = \mathbf{a}/\mathbf{K}\mathbf{v}$
2. $\mathbf{v} = \mathbf{b}/\mathbf{K}^T \mathbf{u}$

- [Sinkhorn'64] proved first convergence result
[Lorenz+'89] characterised linear convergence
- Recent wave of great results by [Altschuler+'17]
[Dvurechensky+'18][Lin+'19]
- $O(nm)$ complexity, GPGPU parallel [C'13].
- $O(n \log n)$ on gridded spaces using convolutions.
[Solomon+'15]

Fast & Scalable Algorithm

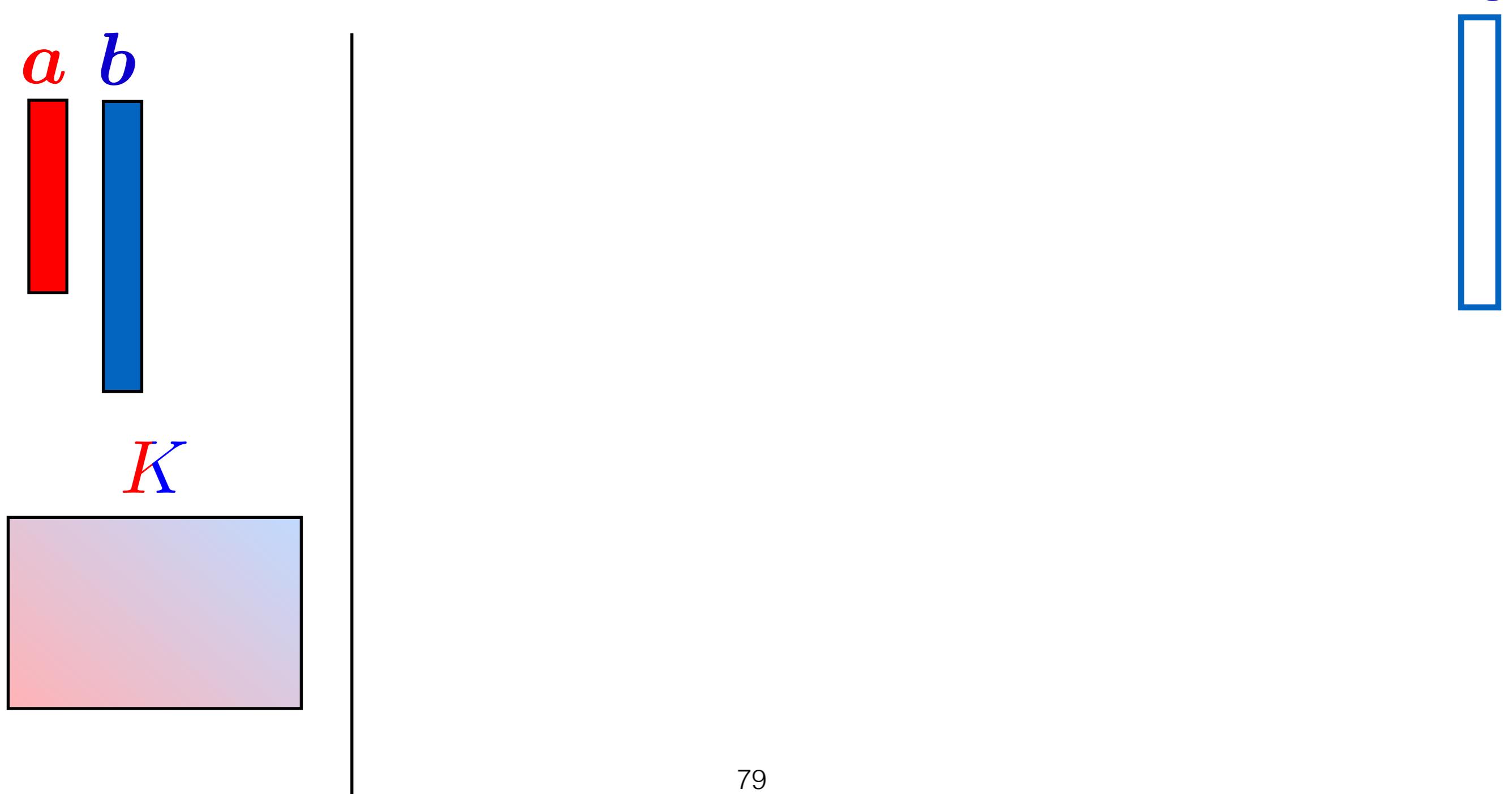
- [Sinkhorn'64] fixed-point iterations for (\mathbf{u}, \mathbf{v})
 $\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$



Fast & Scalable Algorithm

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

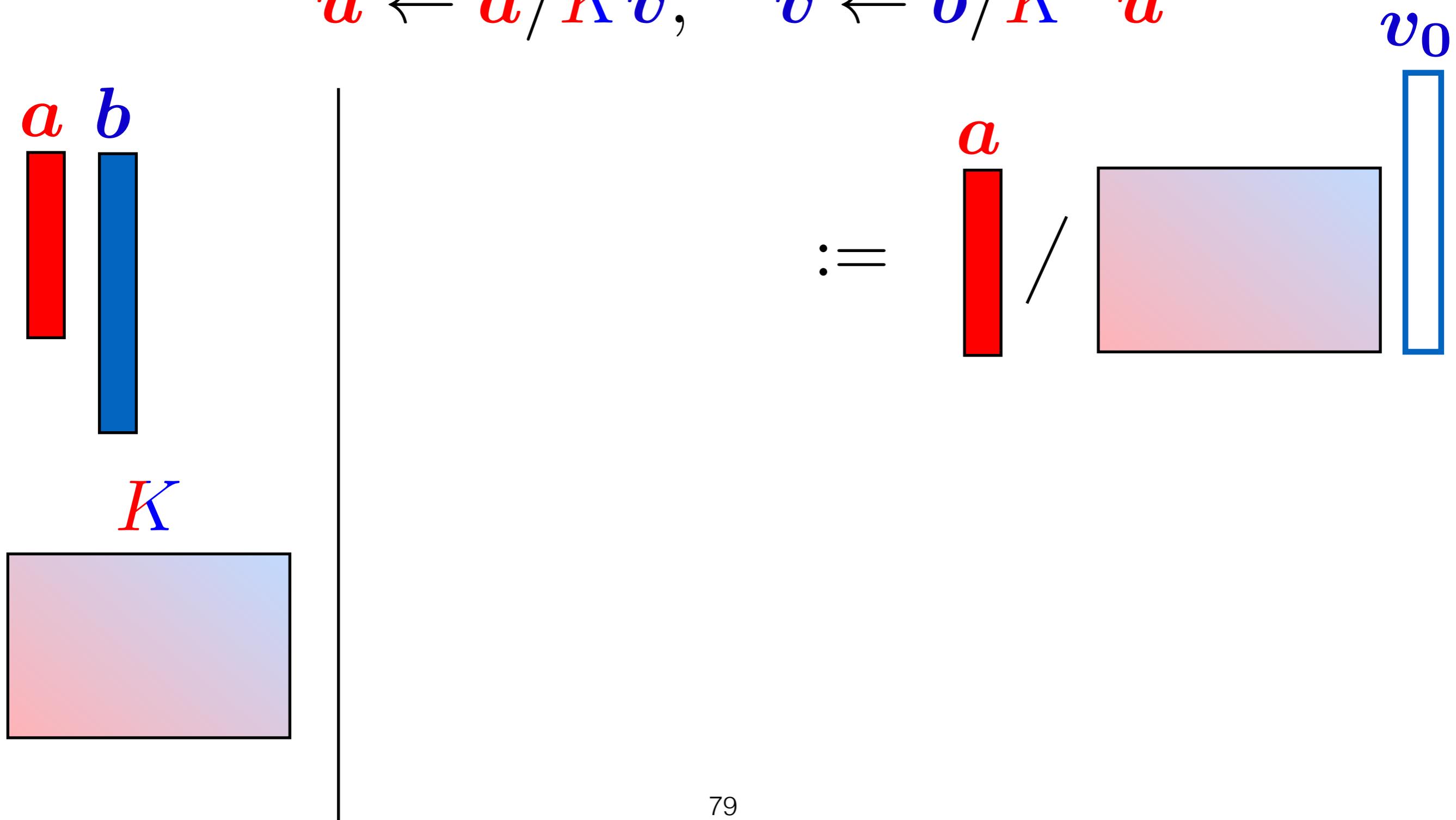
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

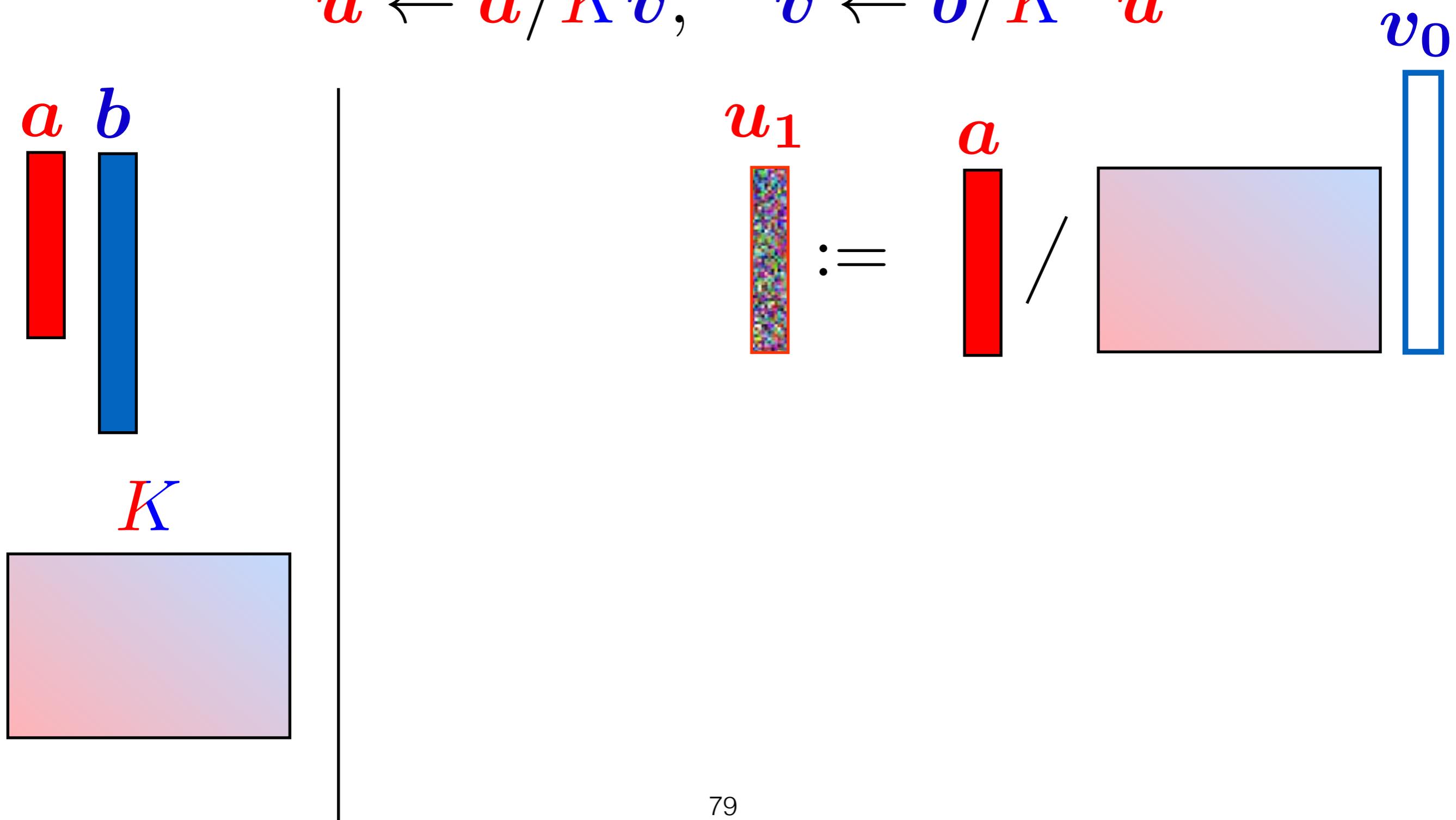
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

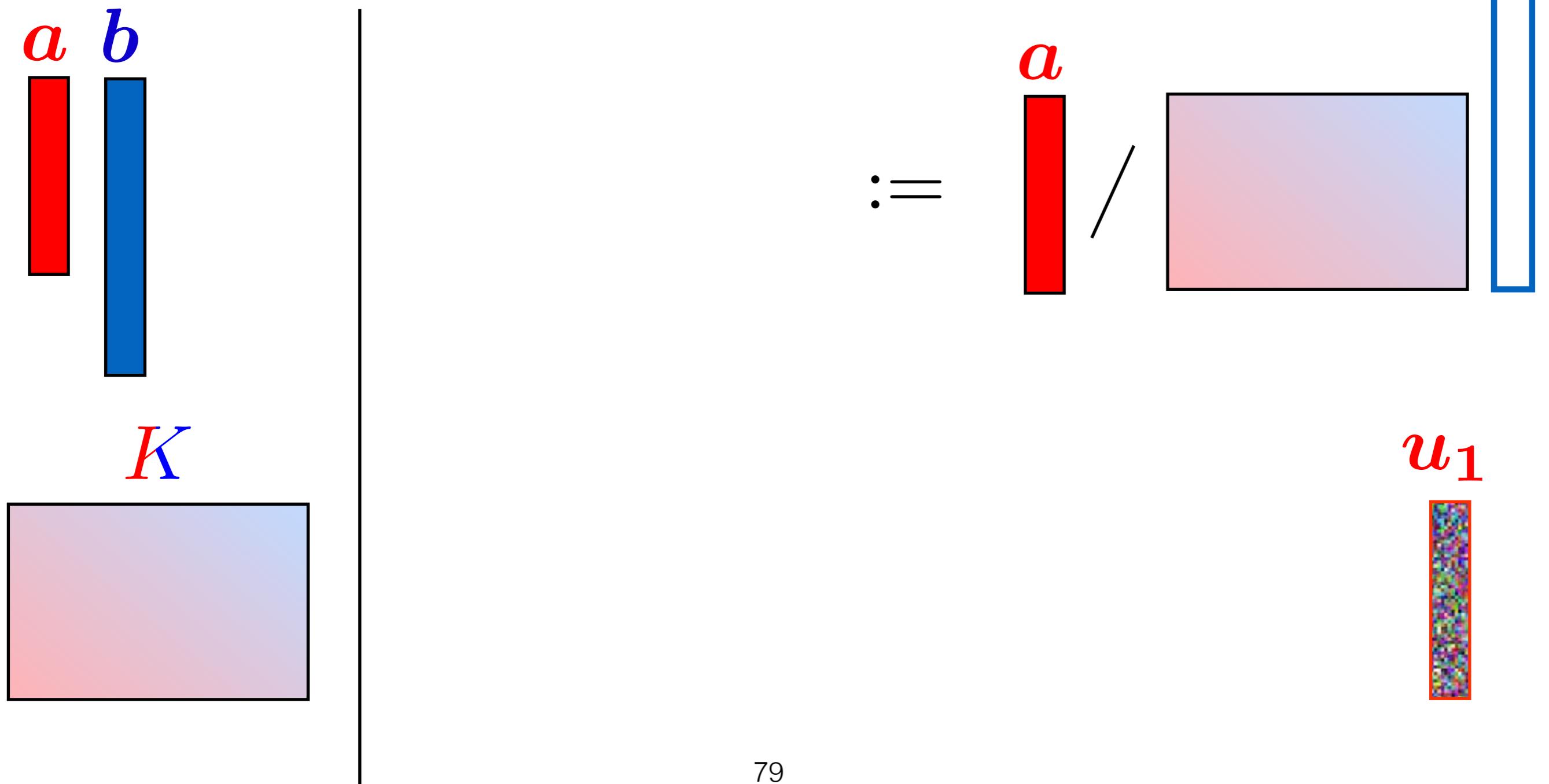
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

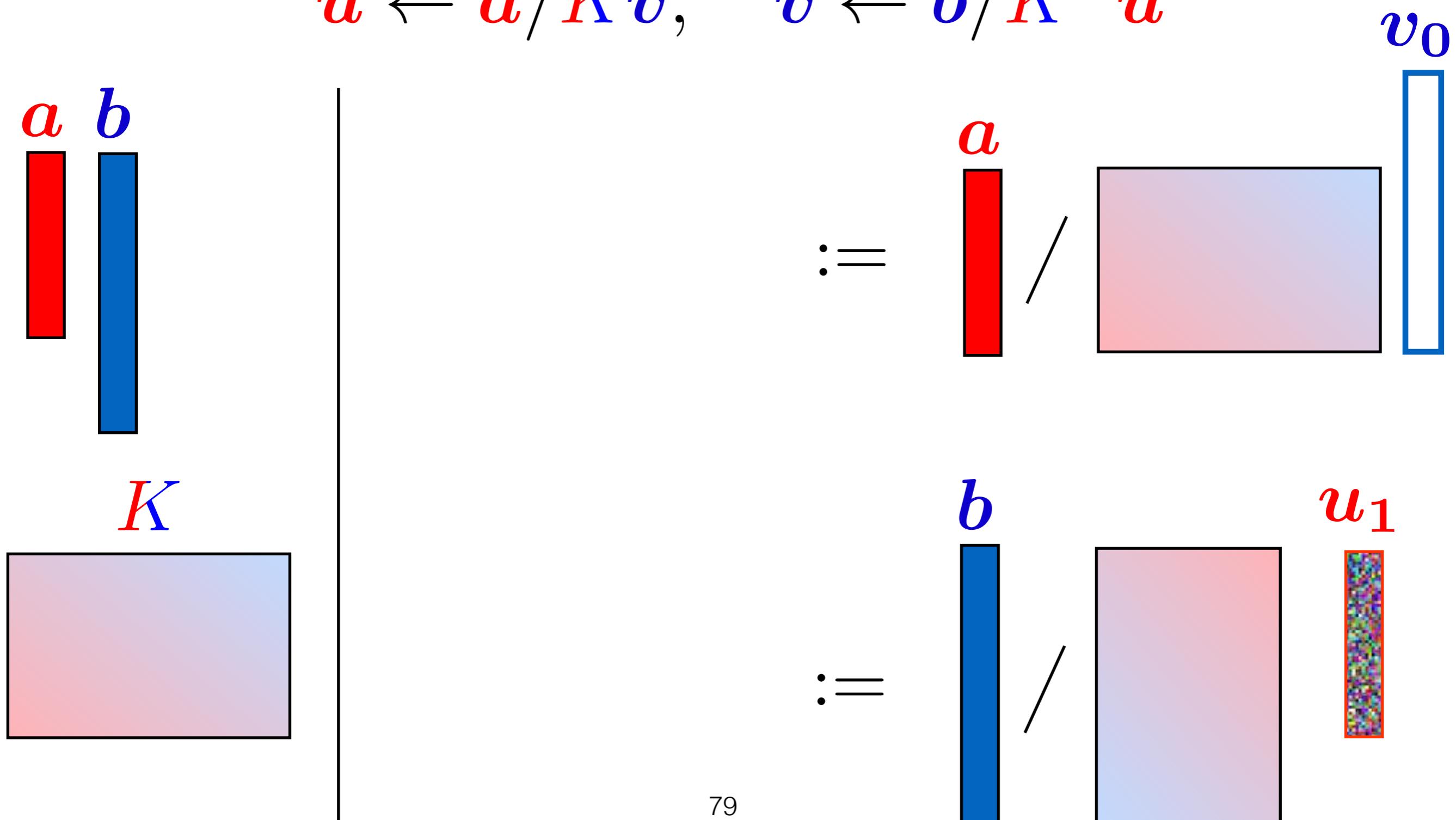
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

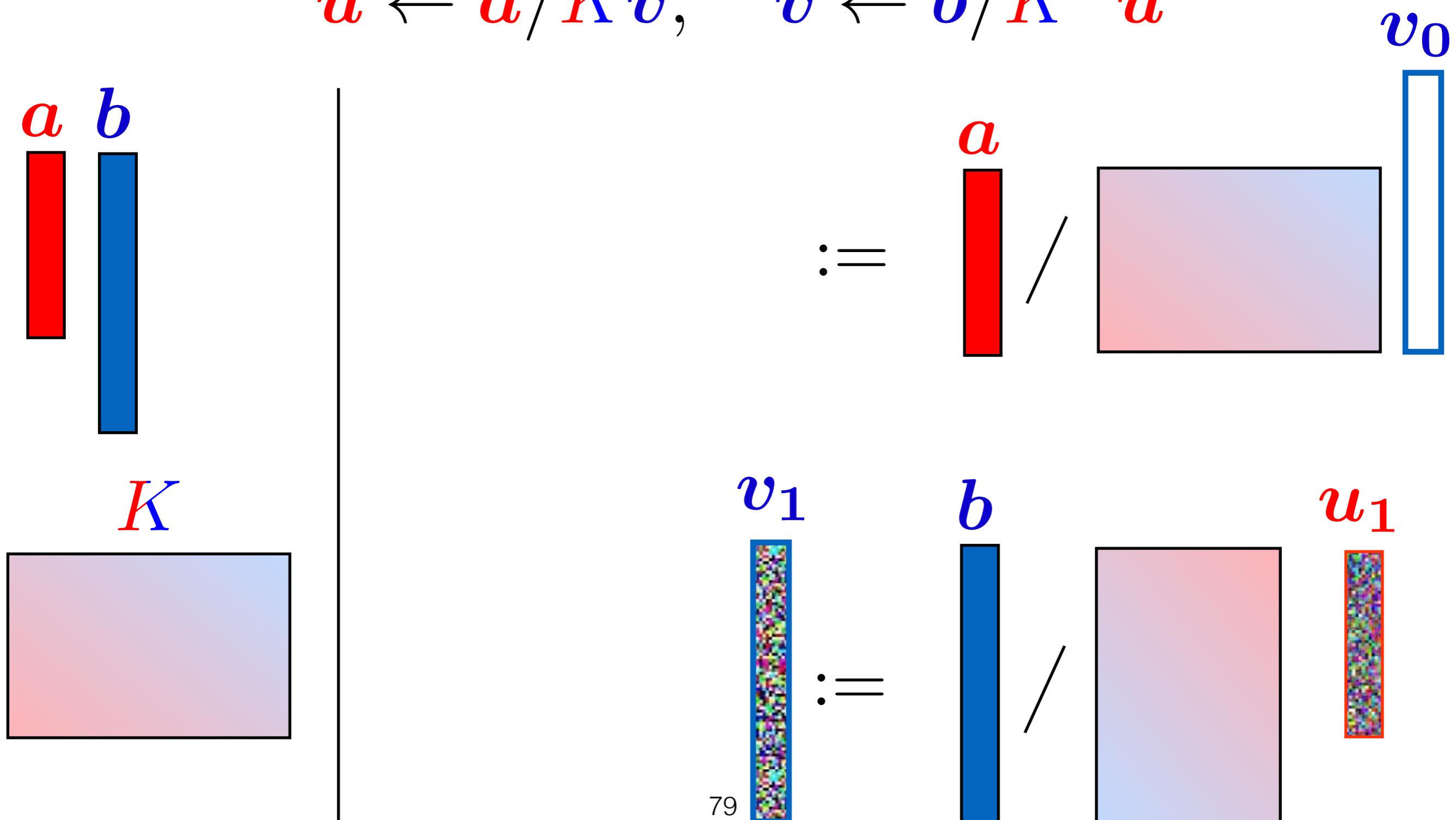
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

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

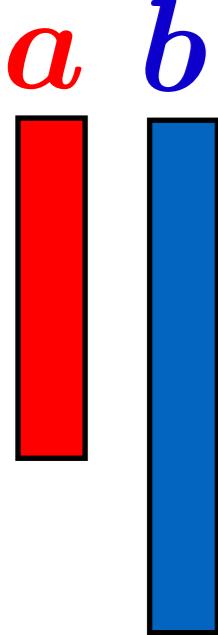


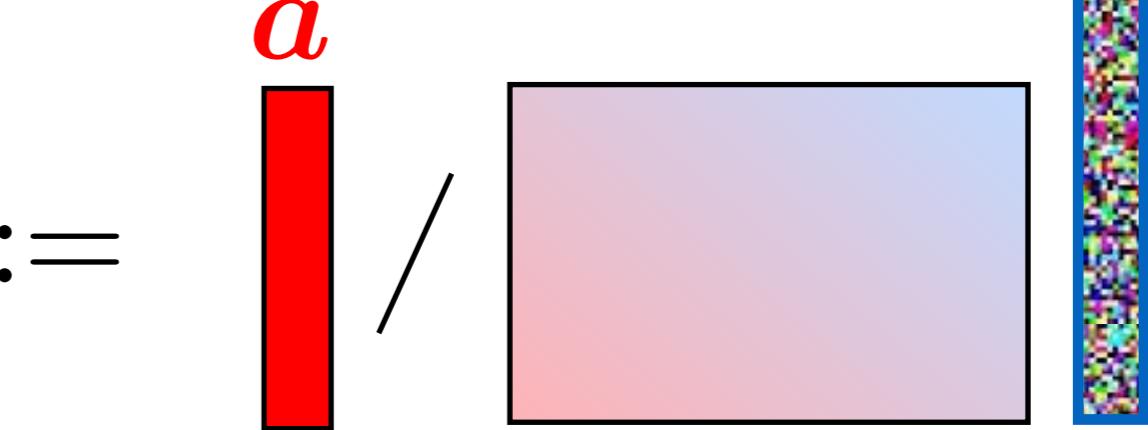
Fast & Scalable Algorithm

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

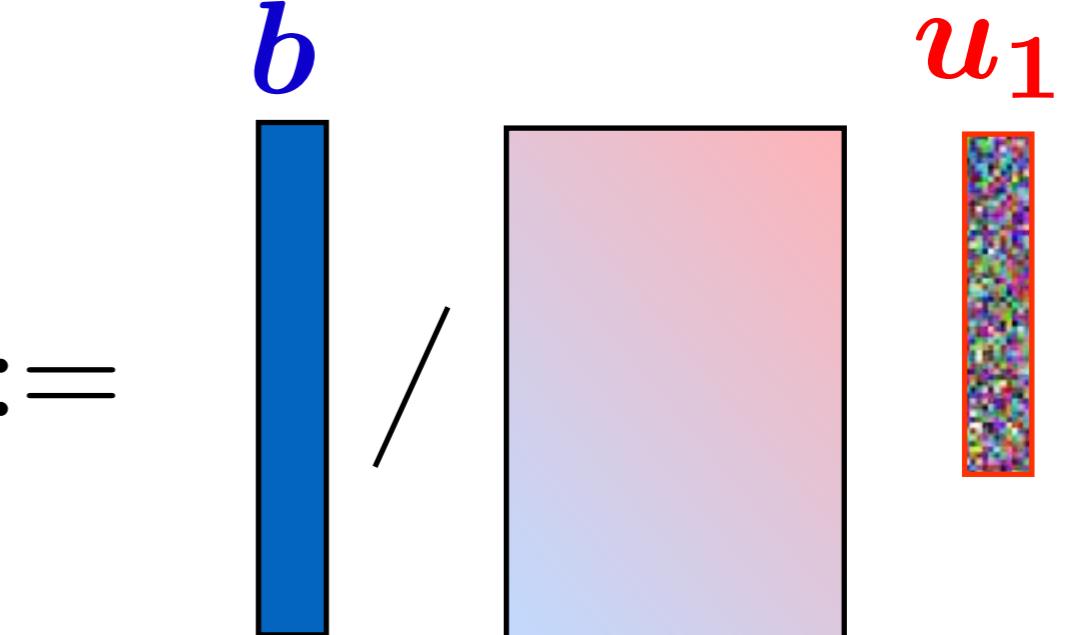
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

v_1

$$\begin{matrix} \mathbf{a} & \mathbf{b} \end{matrix}$$


$$:= \begin{matrix} \mathbf{a} \\ \mathbf{b} \end{matrix} / \begin{matrix} \text{pink gradient} \\ \text{blue gradient} \end{matrix} \quad v_1$$


$$\mathbf{K}$$

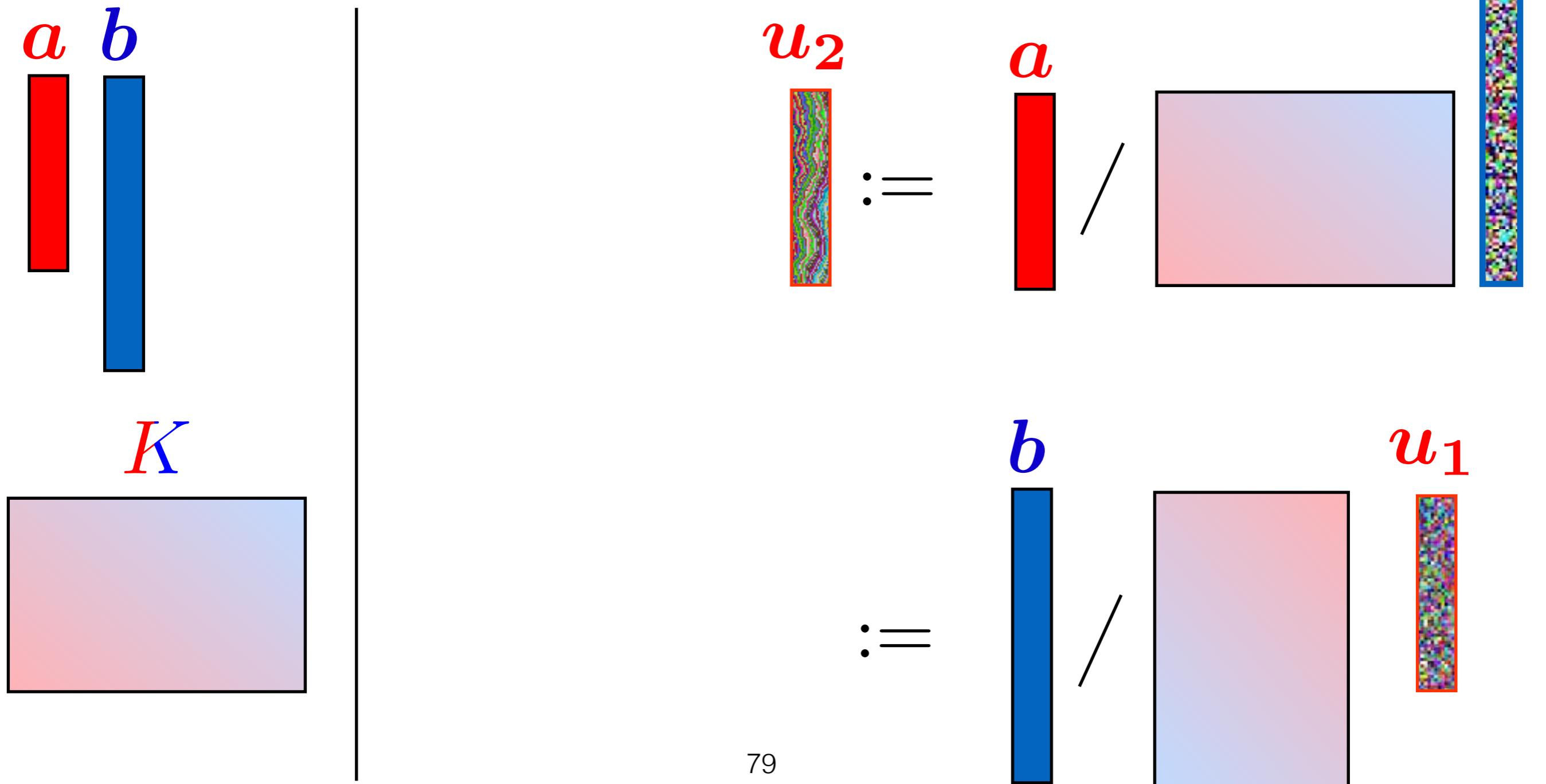

$$:= \begin{matrix} \mathbf{b} \\ \mathbf{a} \end{matrix} / \begin{matrix} \text{pink gradient} \\ \text{blue gradient} \end{matrix} \quad u_1$$


Fast & Scalable Algorithm

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

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

v_1

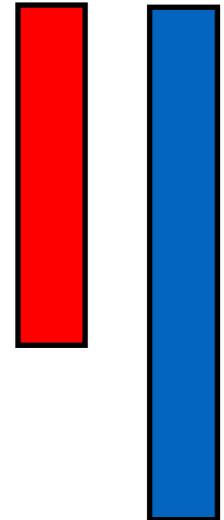


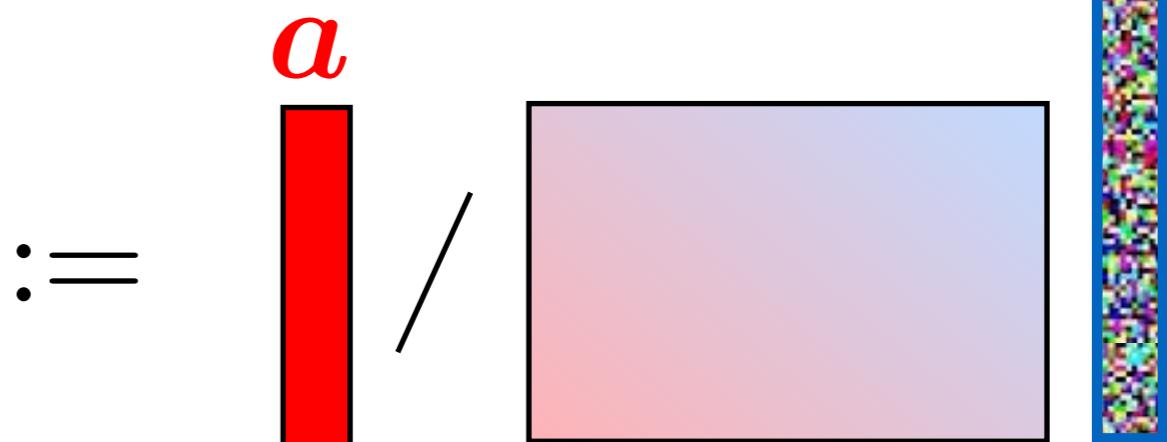
Fast & Scalable Algorithm

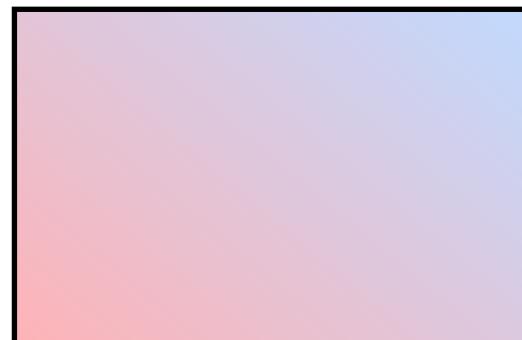
- [Sinkhorn'64] fixed-point iterations for (\mathbf{u}, \mathbf{v})

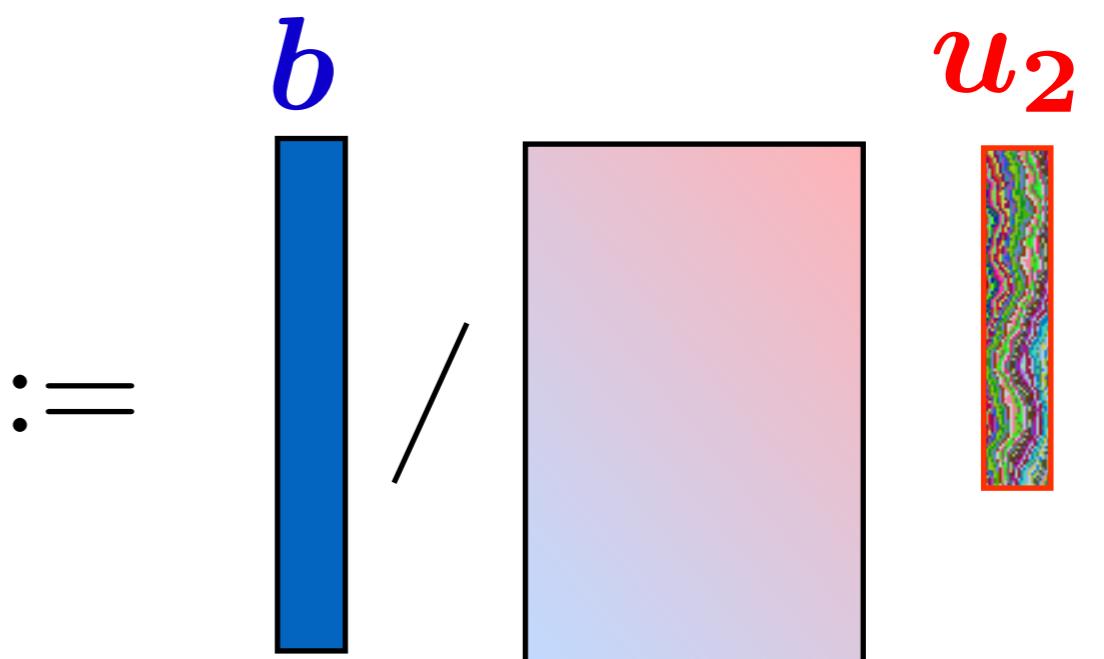
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

v_1

$$\begin{matrix} \mathbf{a} & \mathbf{b} \end{matrix}$$


$$:= \begin{matrix} \mathbf{a} \\ \mathbf{b} \end{matrix} / \begin{matrix} \text{pink gradient} \\ \text{blue gradient} \end{matrix} \quad v_1$$


$$\mathbf{K}$$


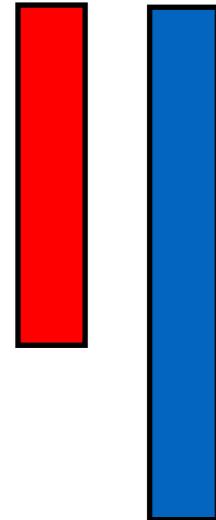
$$:= \begin{matrix} \mathbf{b} \\ \mathbf{a} \end{matrix} / \begin{matrix} \text{pink gradient} \\ \text{blue gradient} \end{matrix} \quad u_2$$


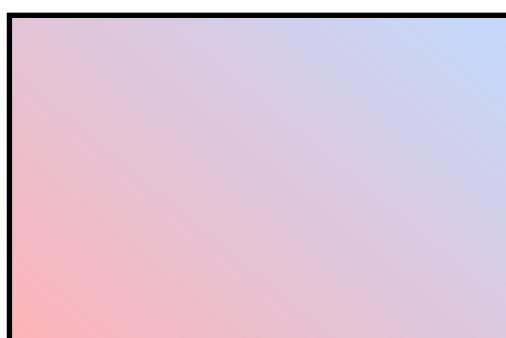
Fast & Scalable Algorithm

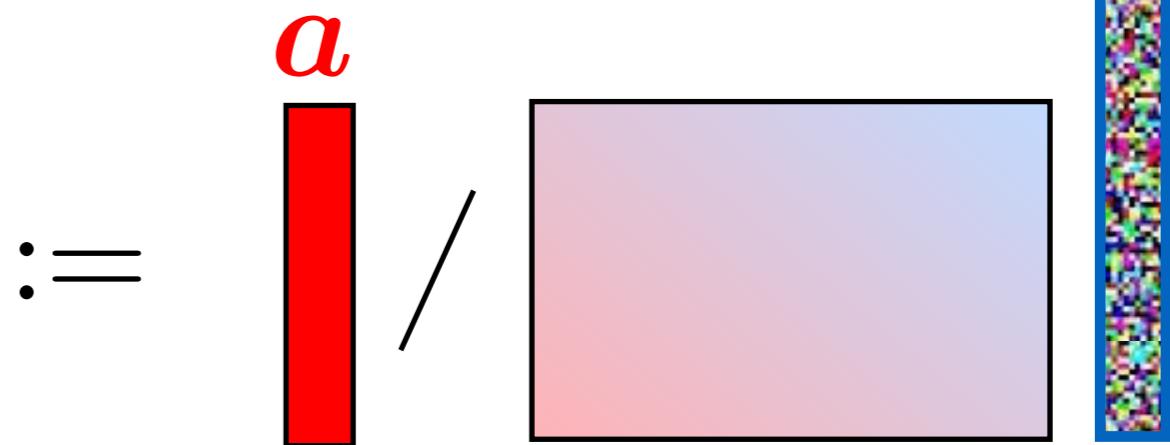
- [Sinkhorn'64] fixed-point iterations for (\mathbf{u}, \mathbf{v})

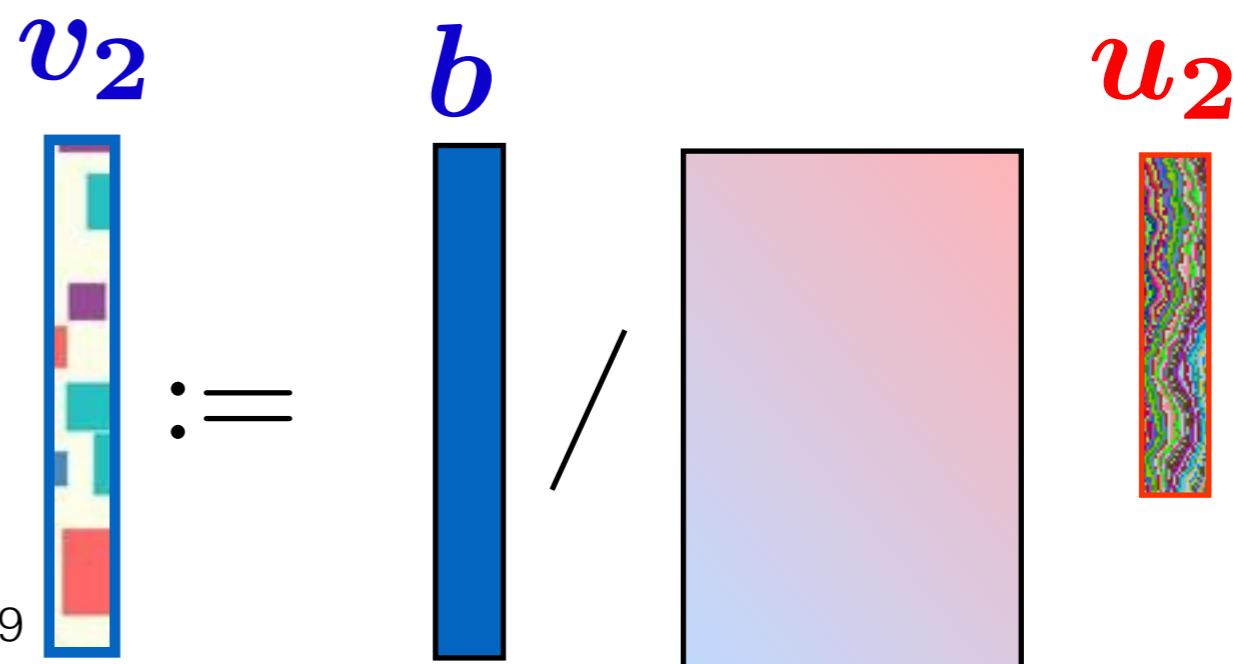
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

v_1

$$\begin{matrix} \mathbf{a} & \mathbf{b} \end{matrix}$$


$$K$$


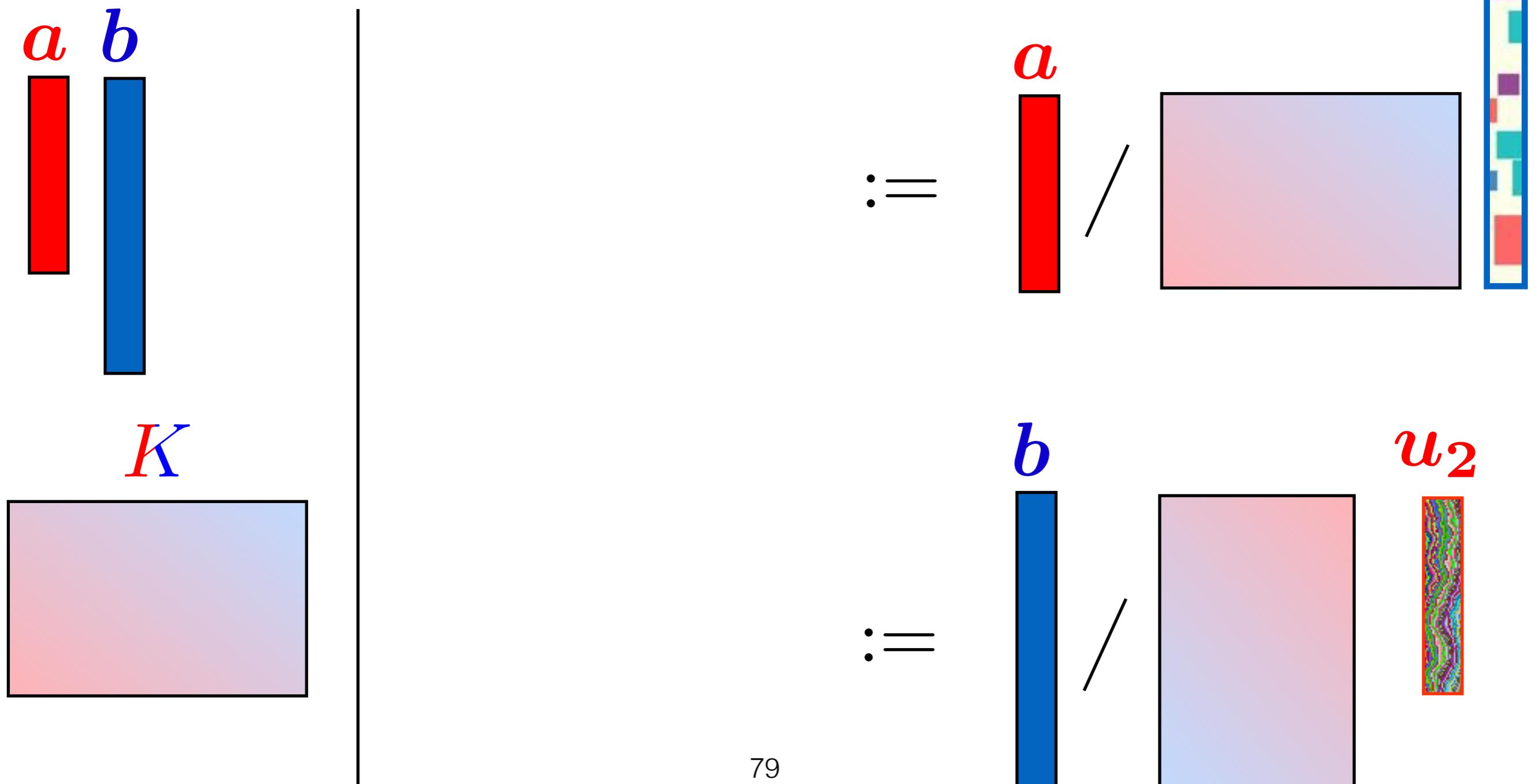
$$\mathbf{v}_1 := \mathbf{a} / \begin{matrix} \mathbf{b} \\ \text{---} \\ \mathbf{u}_1 \end{matrix}$$


$$\mathbf{v}_2 := \mathbf{b} / \begin{matrix} \mathbf{a} \\ \text{---} \\ \mathbf{u}_2 \end{matrix}$$


Fast & Scalable Algorithm

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

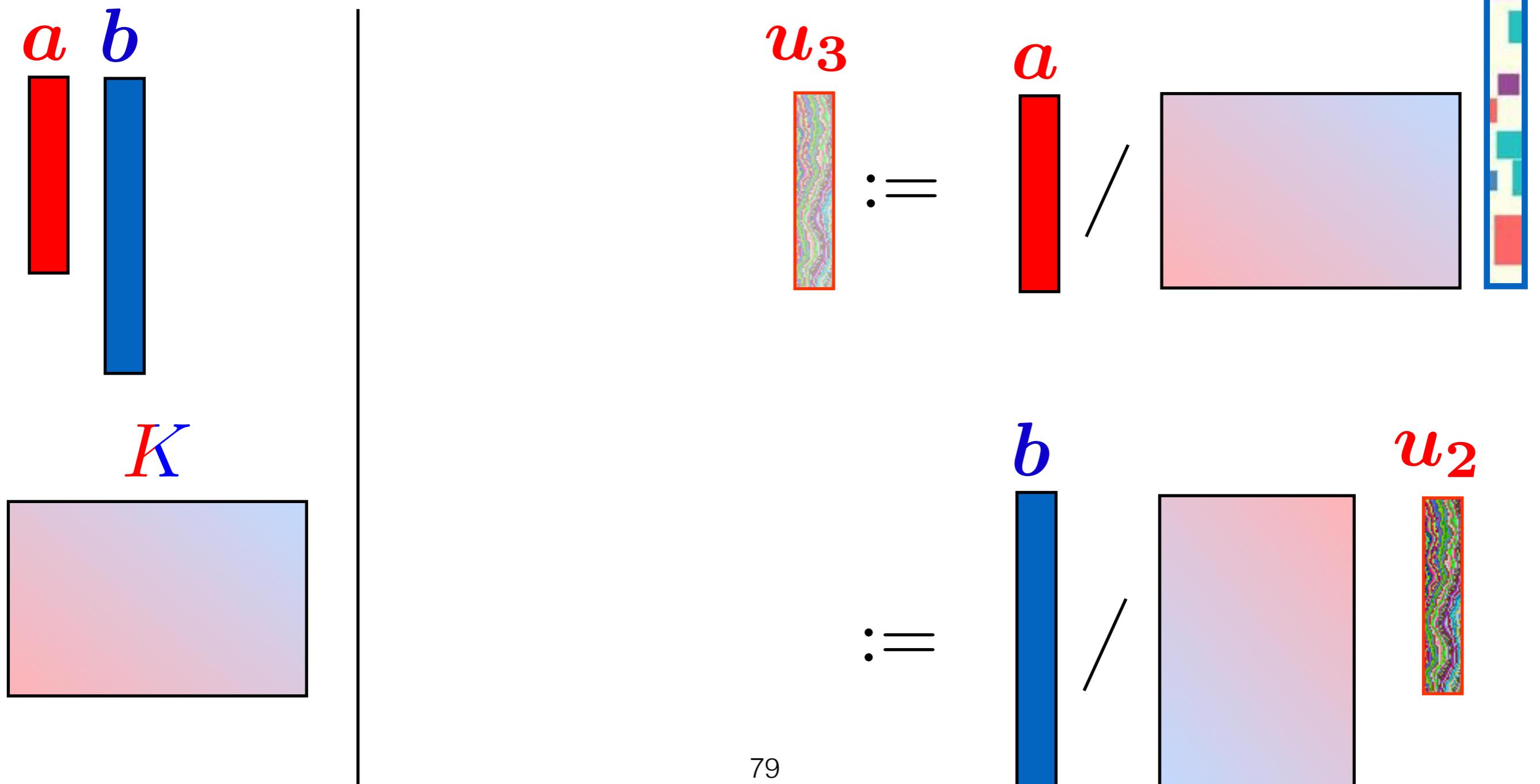
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

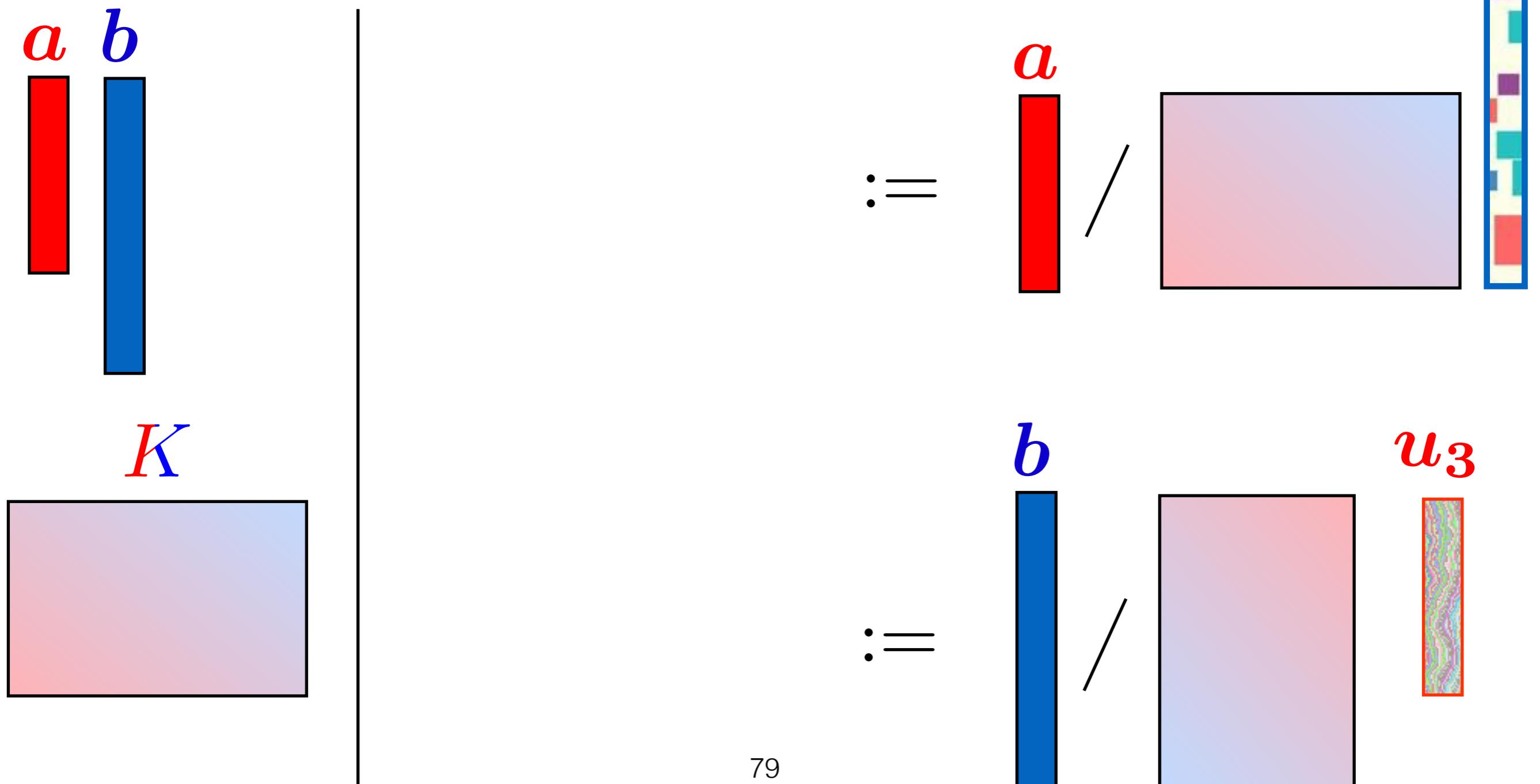
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

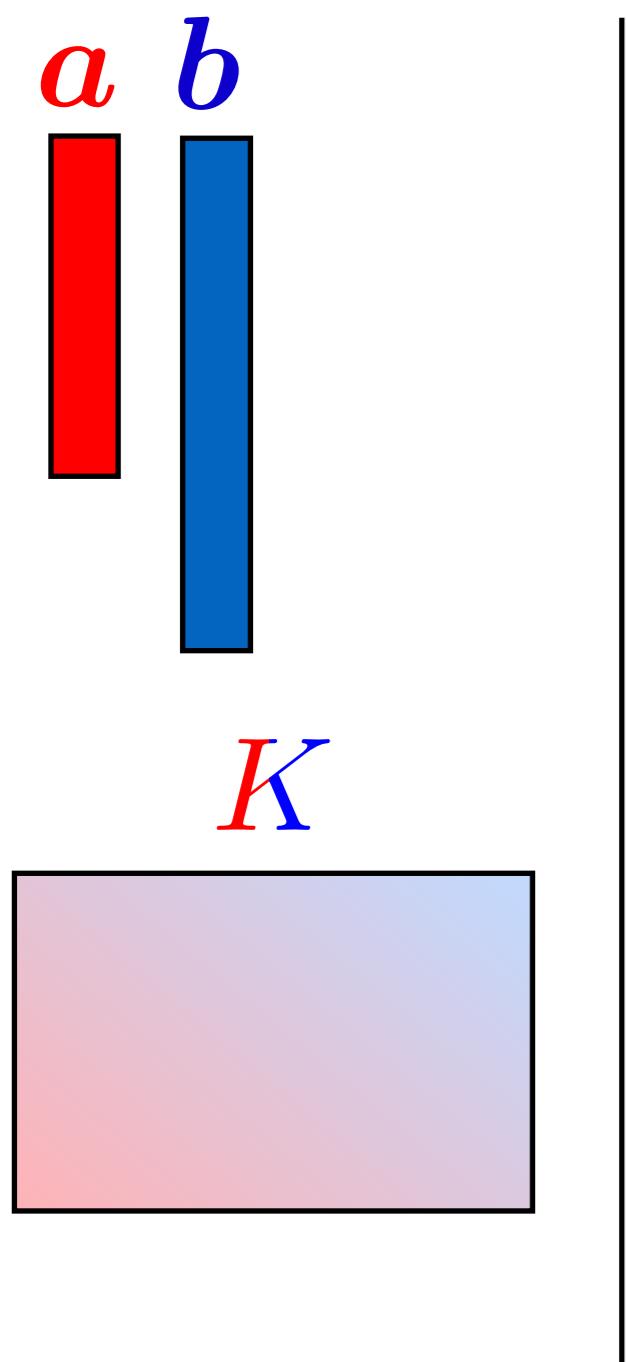
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



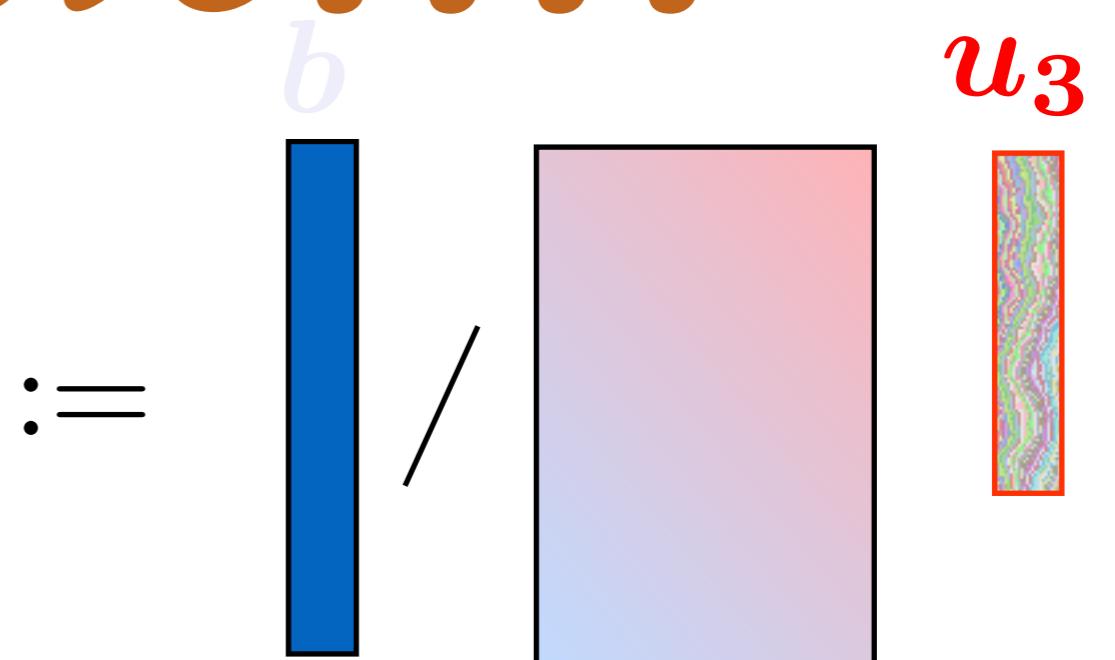
Fast & Scalable Algorithm

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

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

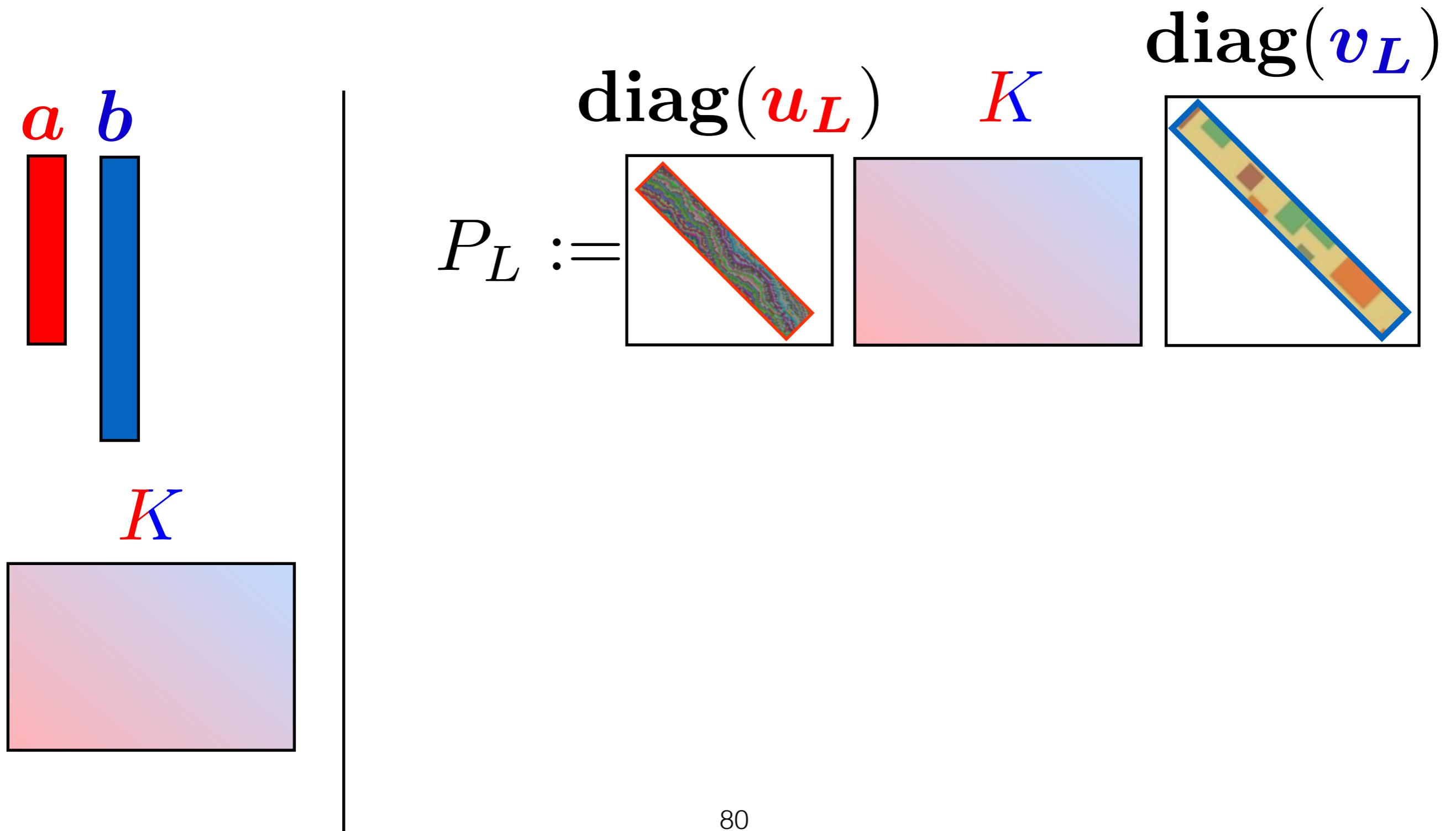


etc. . .



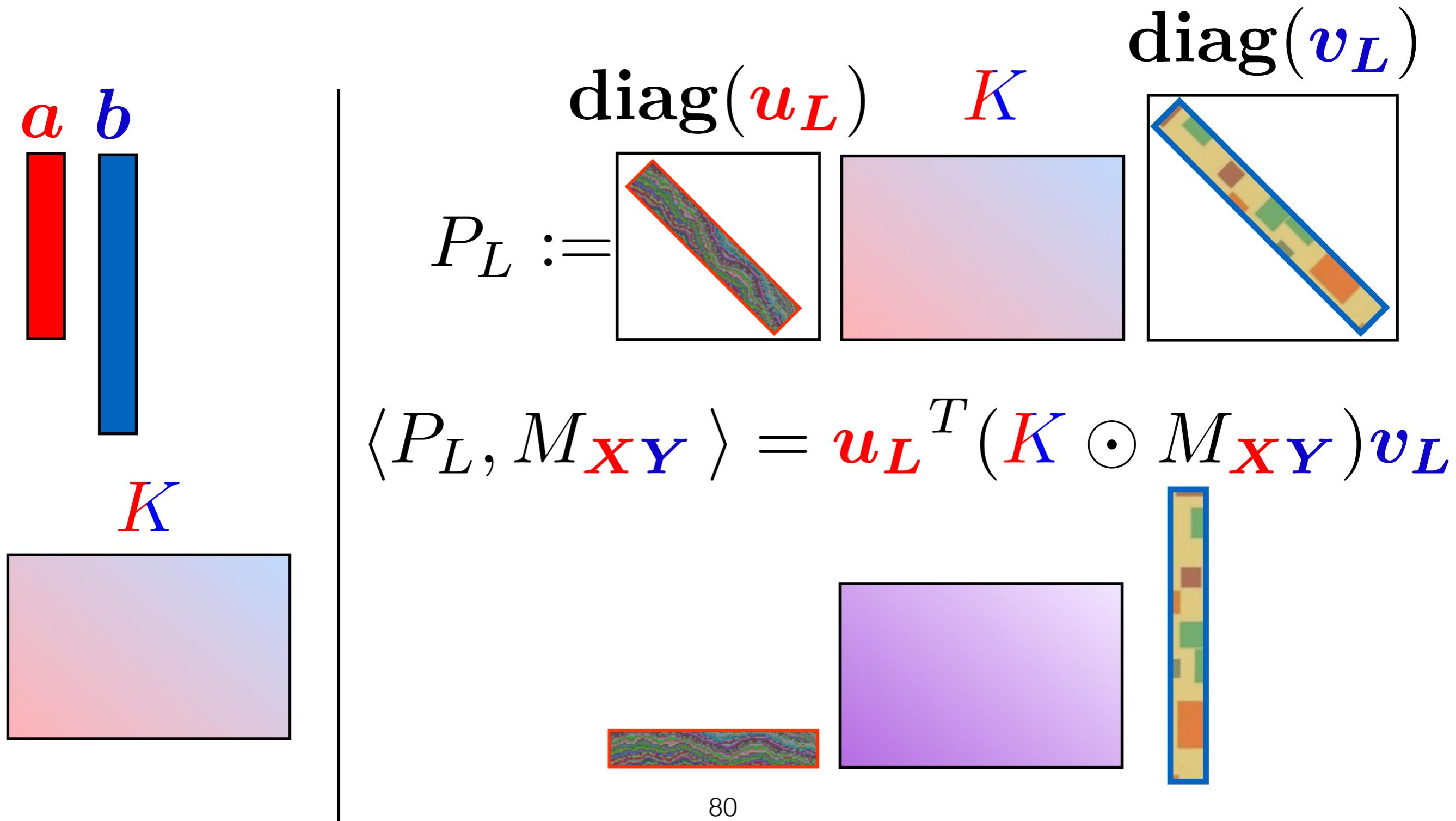
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



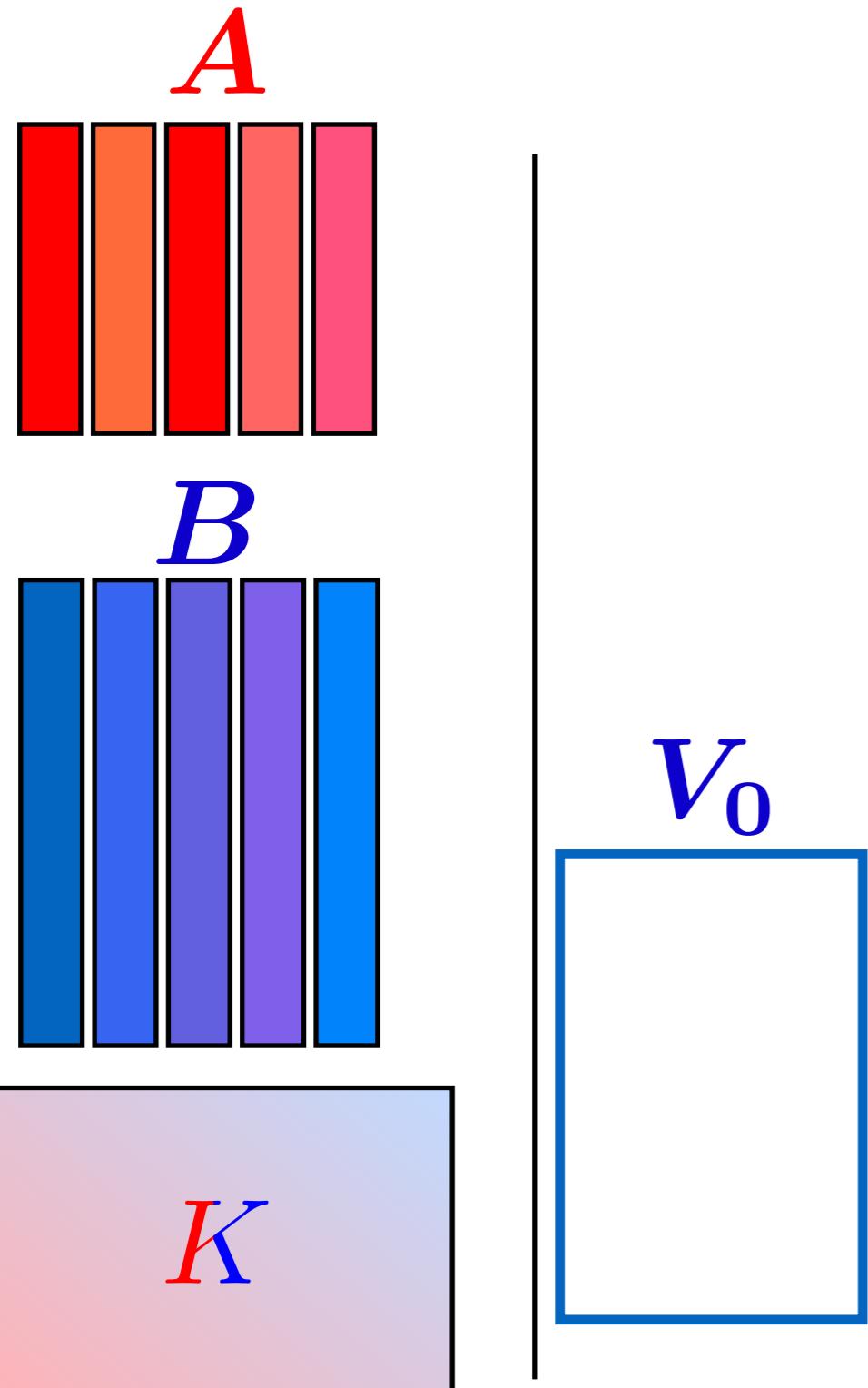
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



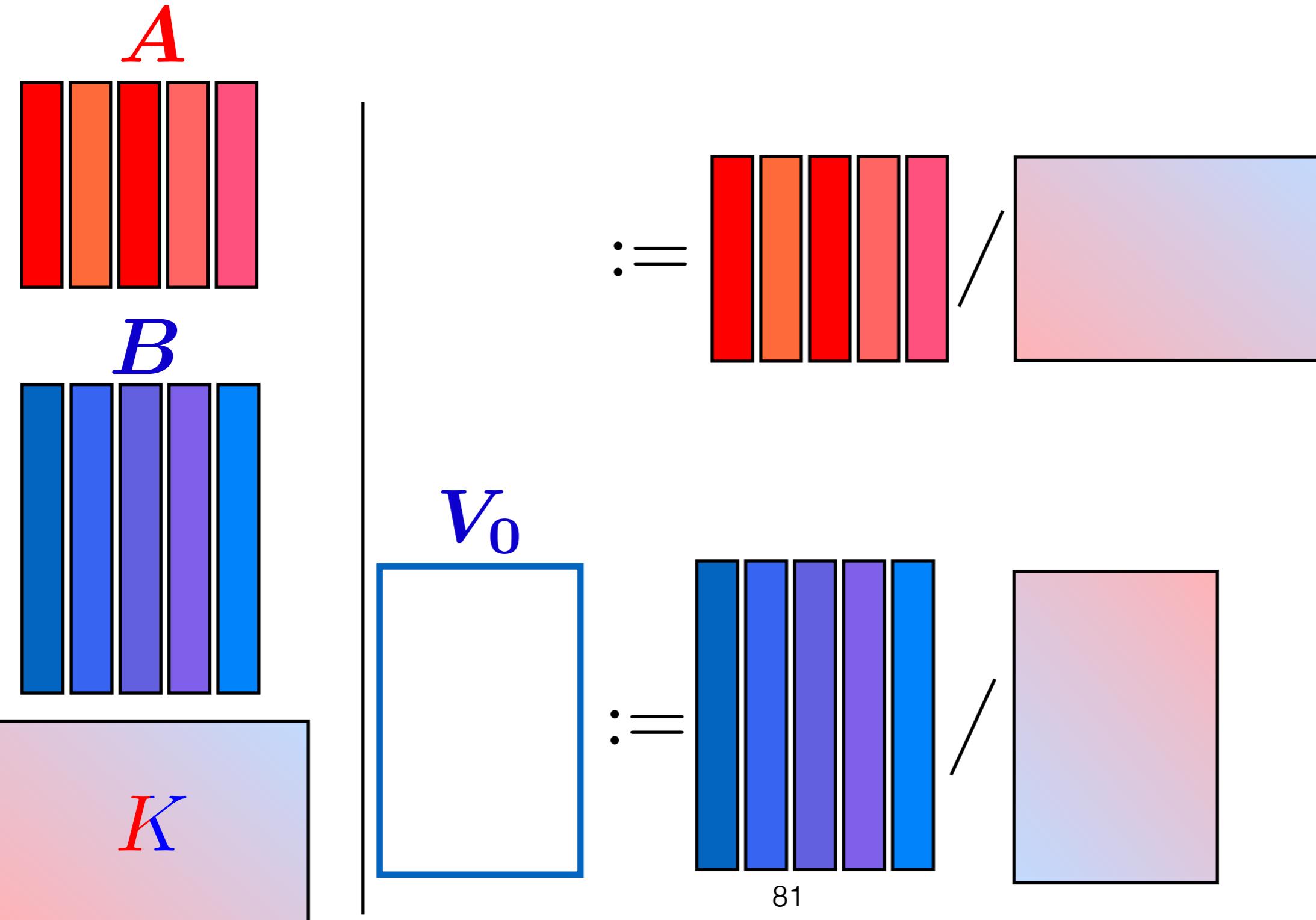
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



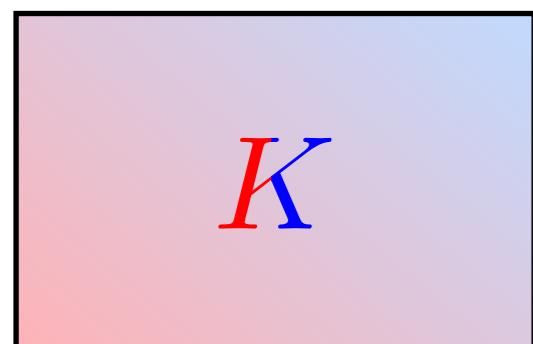
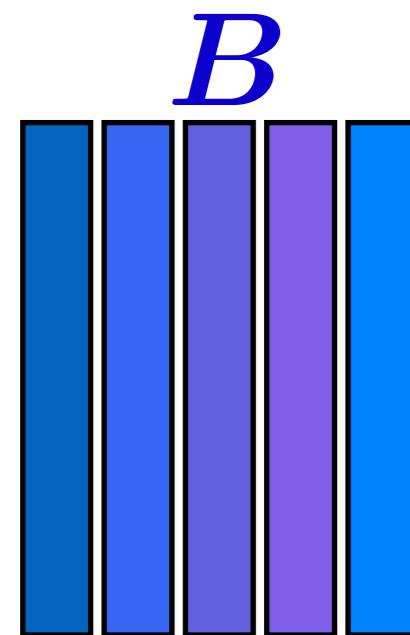
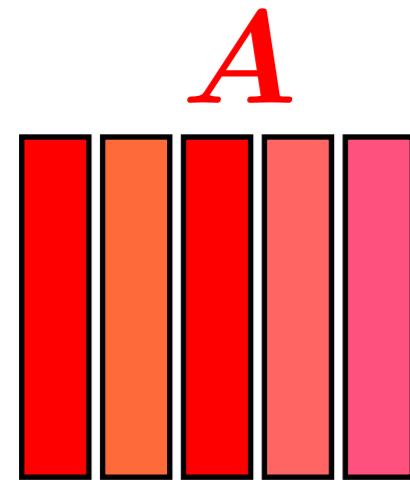
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



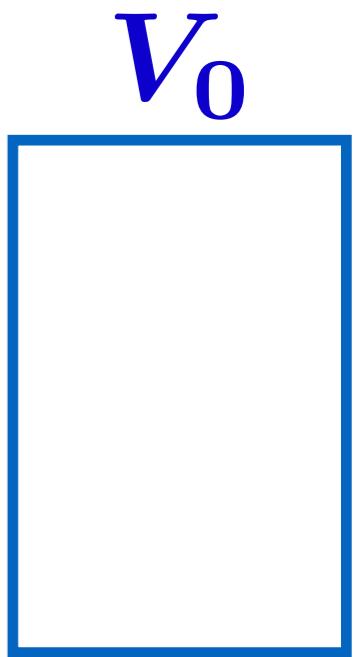
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



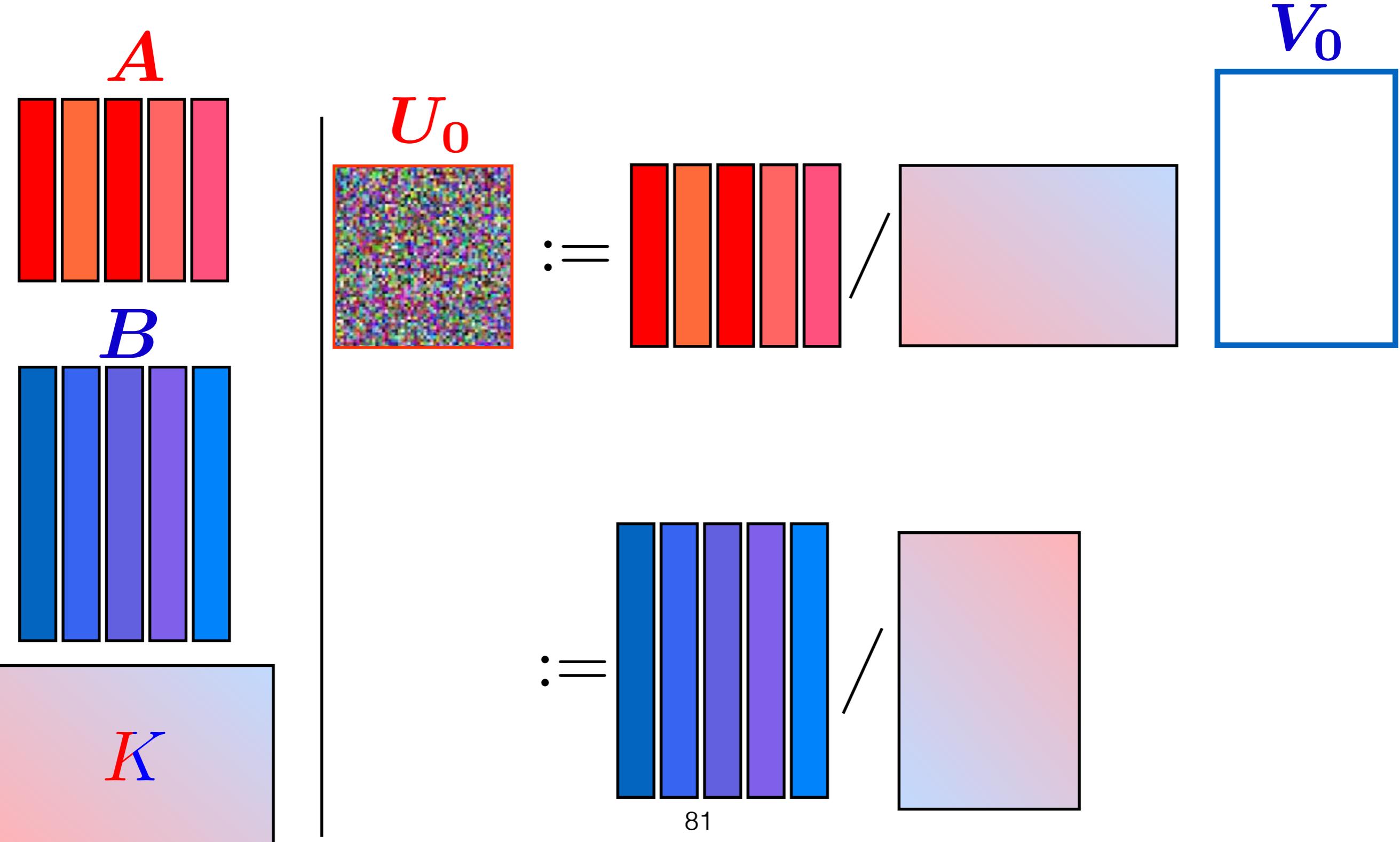
$$:= \begin{array}{c} | \\ \textcolor{red}{A} \\ | \\ \textcolor{blue}{B} \\ | \\ \textcolor{violet}{K} \end{array} \quad \begin{array}{c} | \\ \textcolor{red}{A} \\ | \\ \textcolor{blue}{B} \\ | \\ \textcolor{violet}{K} \end{array}$$

$$:= \begin{array}{c} | \\ \textcolor{red}{A} \\ | \\ \textcolor{blue}{B} \\ | \\ \textcolor{violet}{K} \end{array} \quad \begin{array}{c} | \\ \textcolor{red}{A} \\ | \\ \textcolor{blue}{B} \\ | \\ \textcolor{violet}{K} \end{array}$$



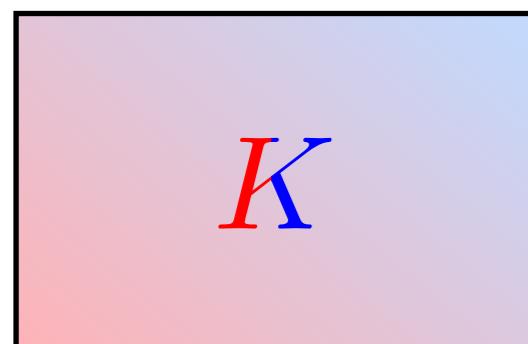
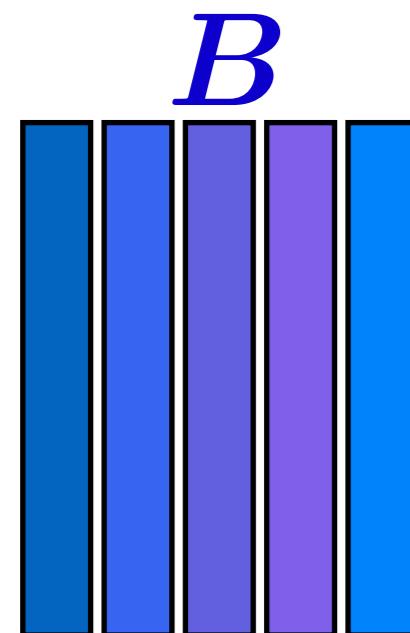
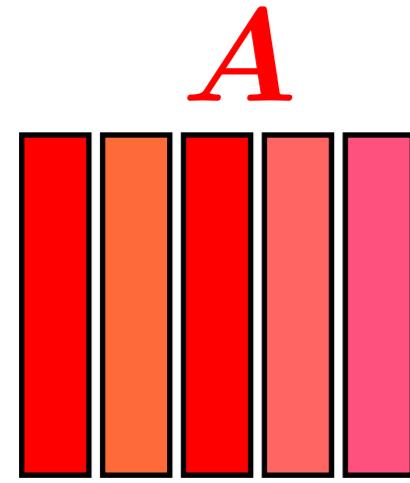
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



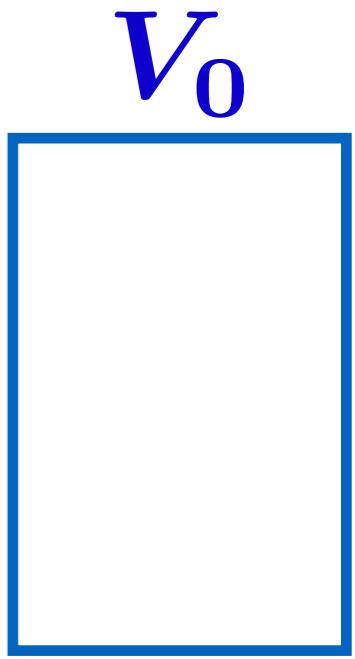
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

A vertical vector consisting of the same five colored bars as vector A .



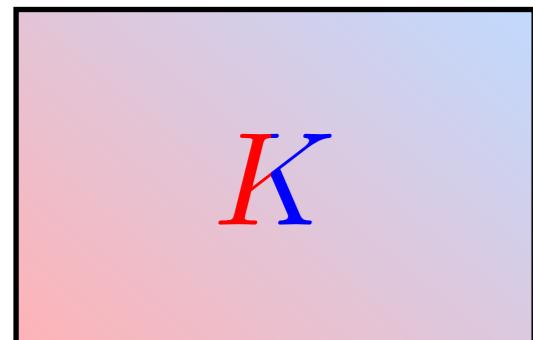
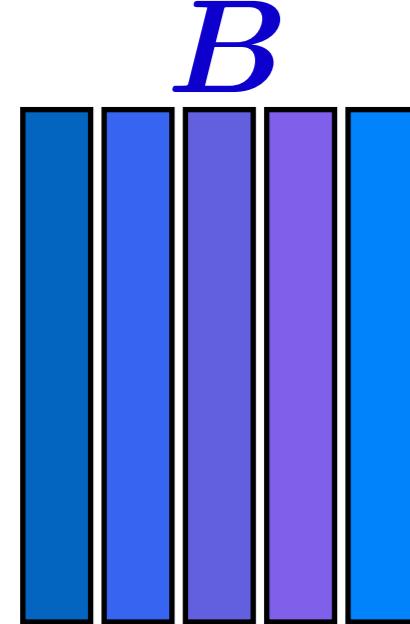
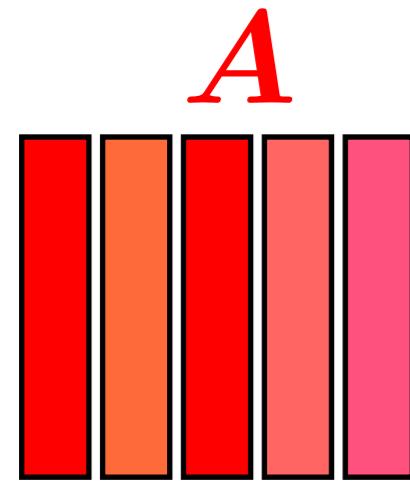
$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

A vertical vector consisting of the same five colored bars as vector B .

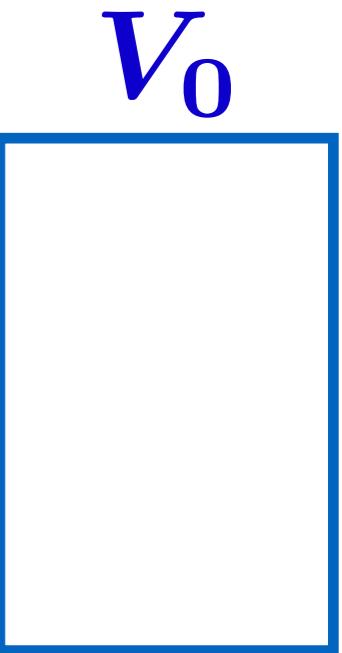


Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$$

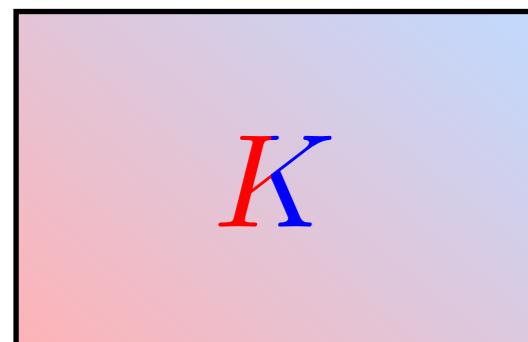
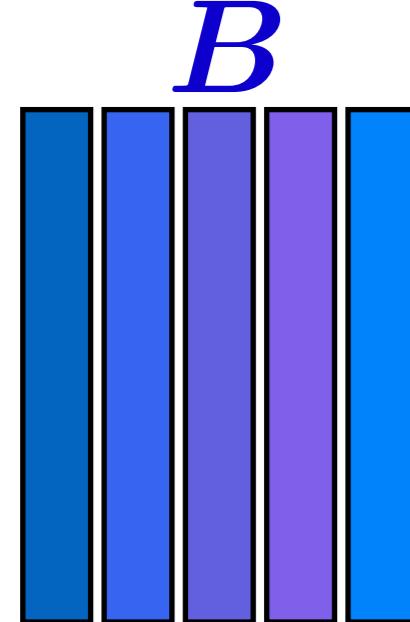
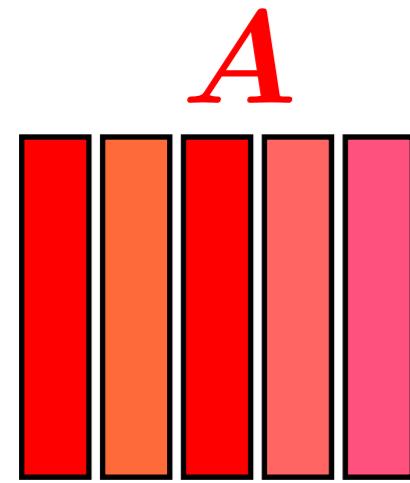


$$:= \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \quad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$$



Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ \text{red bar} \\ | \\ \text{orange bar} \\ | \\ \text{red bar} \\ | \\ \text{pink bar} \\ | \\ \text{red bar} \end{array}$$

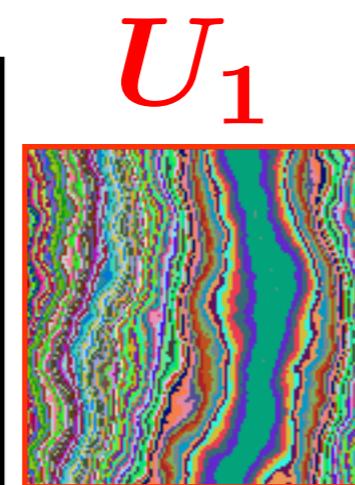
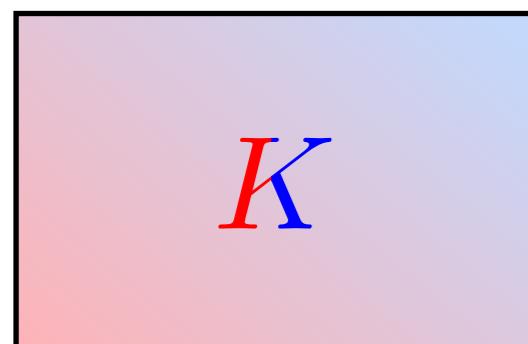
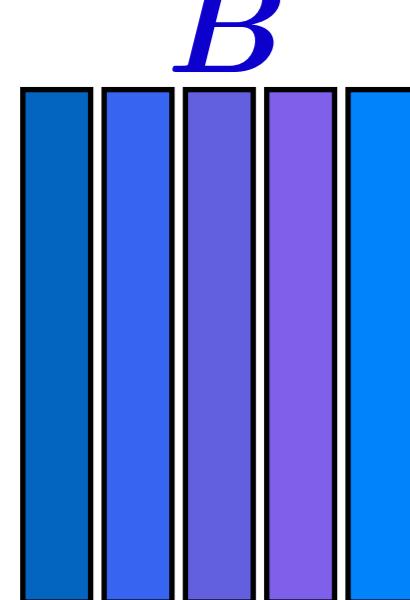
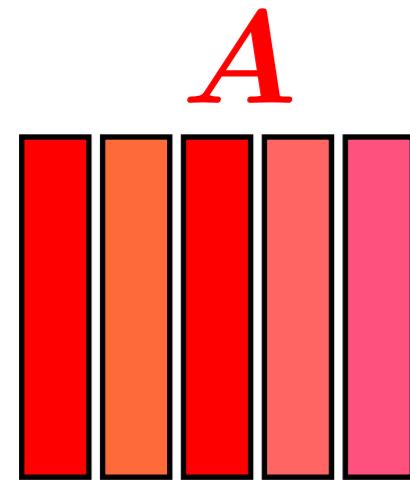


$$:= \begin{array}{c} | \\ \text{blue bar} \\ | \\ \text{blue bar} \\ | \\ \text{purple bar} \\ | \\ \text{purple bar} \\ | \\ \text{blue bar} \end{array}$$



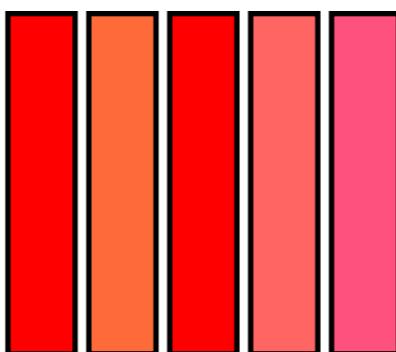
Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



U_1

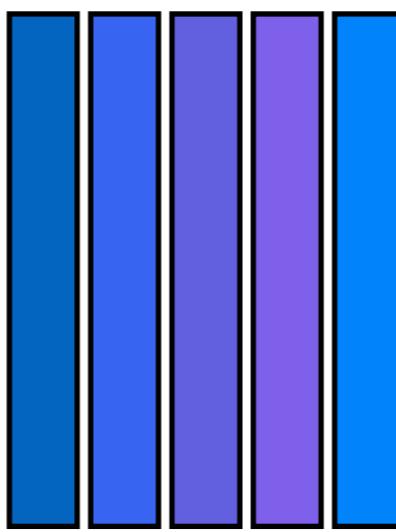
\coloneqq



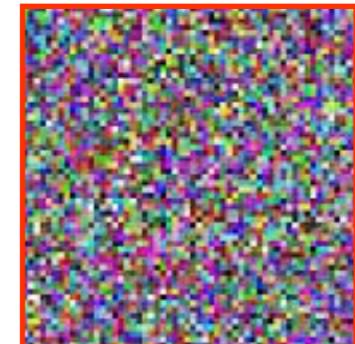
V_1



\coloneqq

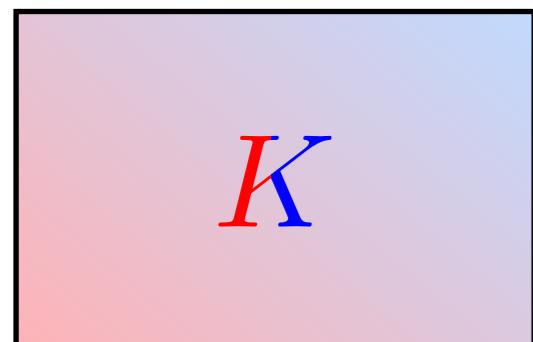
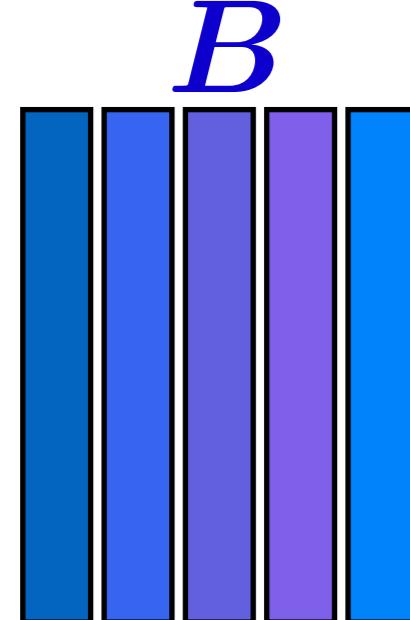
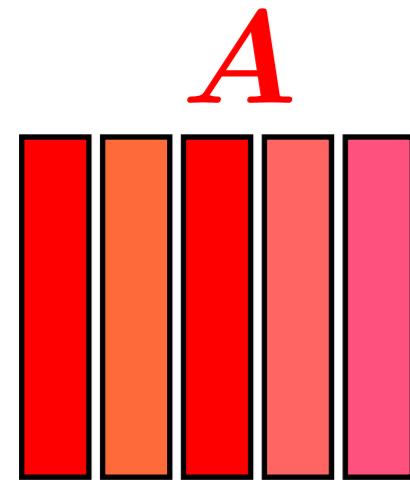


U_0



Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations

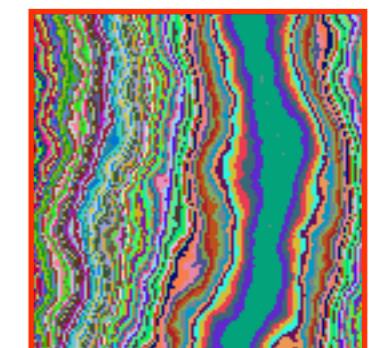


$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

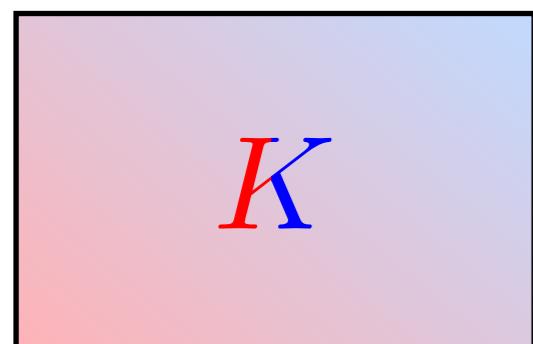
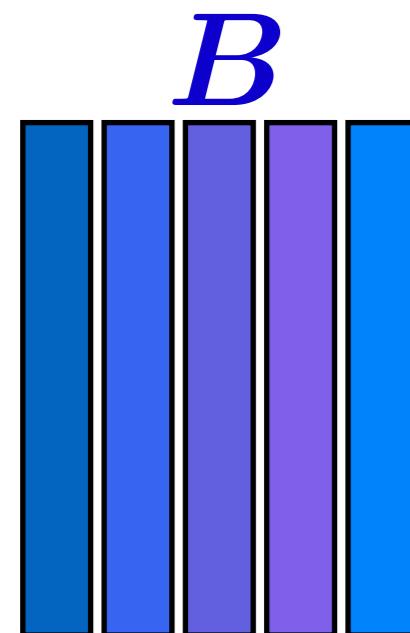
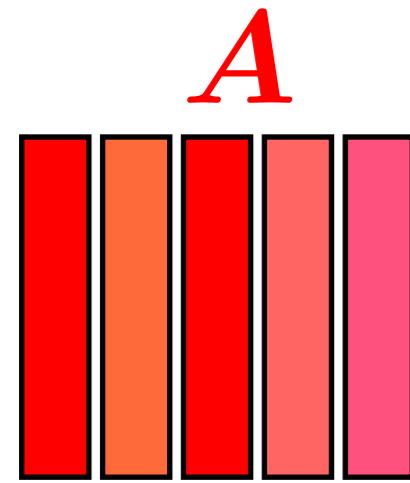


*V*₁



Also embarrassingly parallel

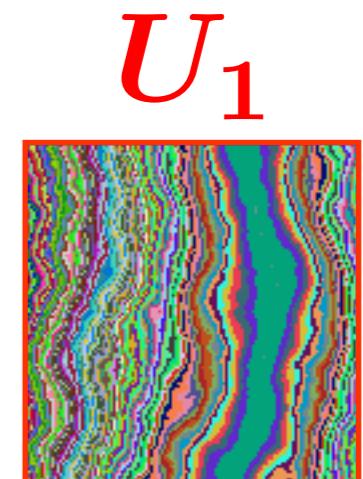
- [Sinkhorn'64] with *matrix* fixed-point iterations



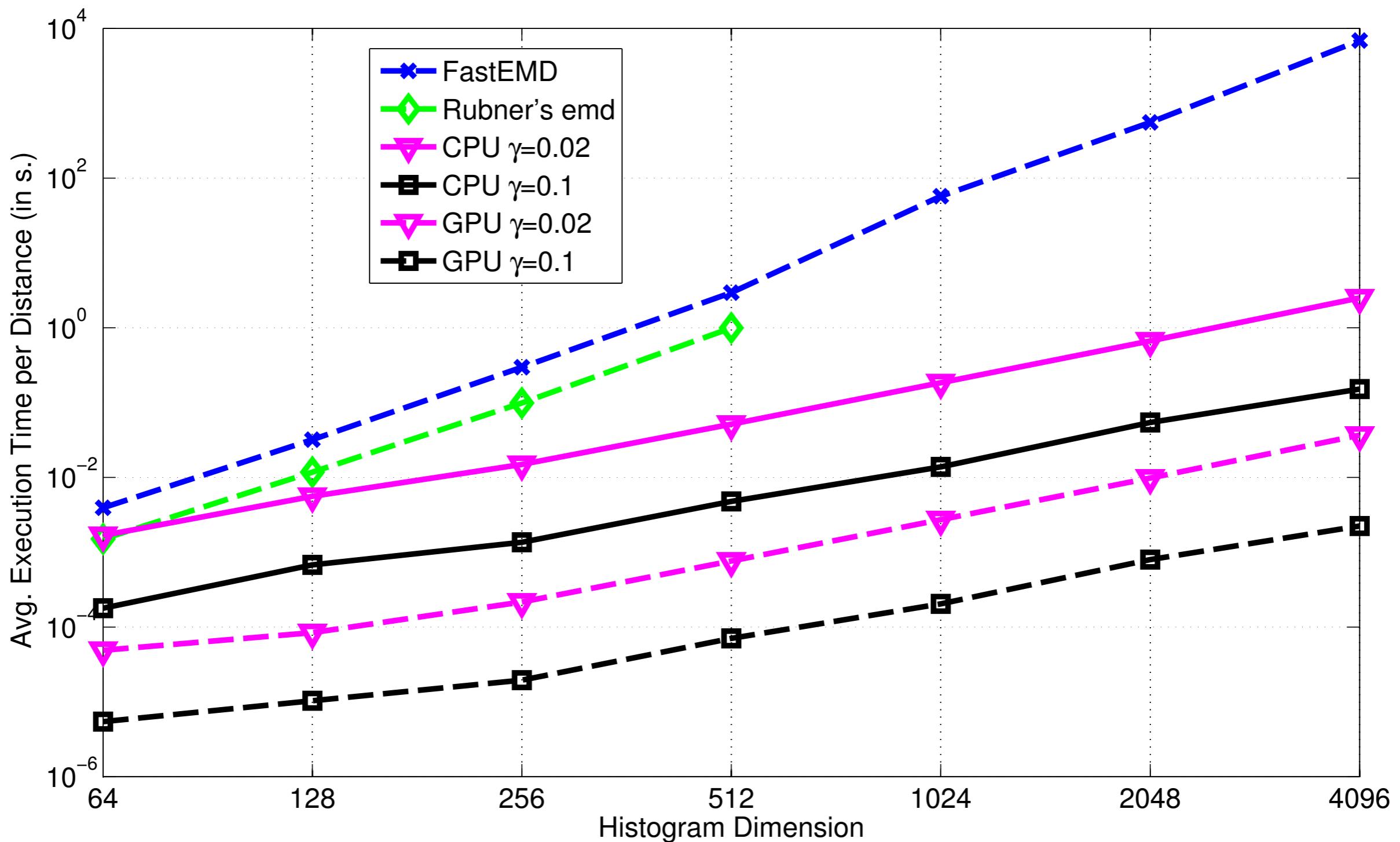
$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

etc. . . .

$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$



Very Fast EMD Approx. Solver

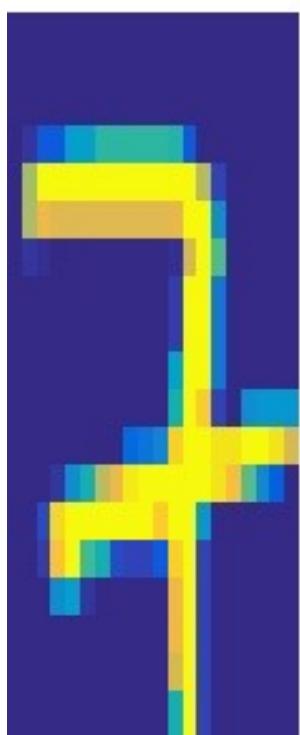


Note. (Ω, \mathbf{D}) is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance 10^{-2} .

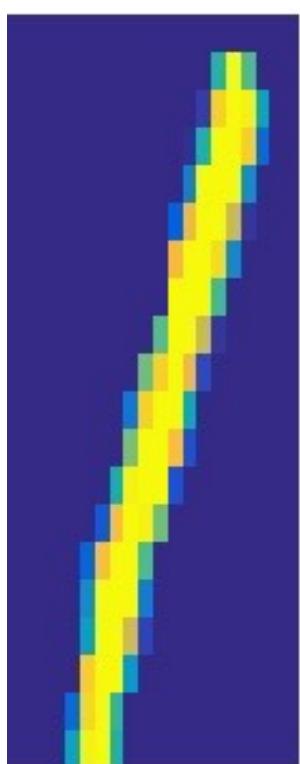
Very Fast EMD Approx. Solver

Very Fast EMD Approx. Solver

a

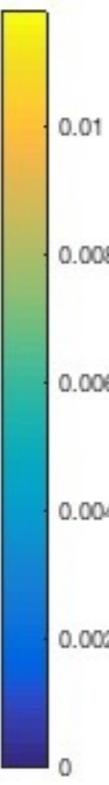
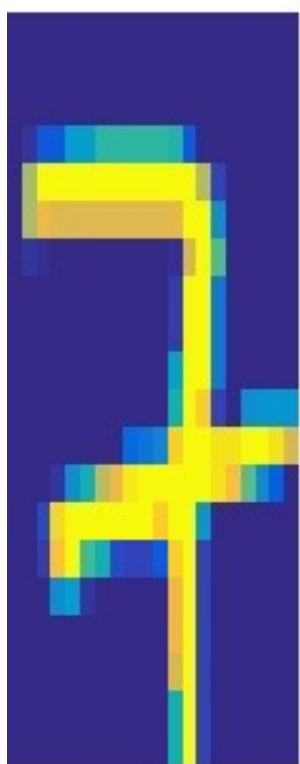


b

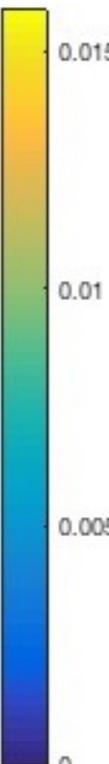
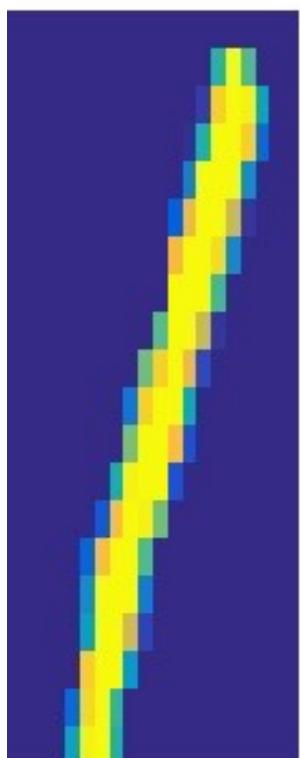


Very Fast EMD Approx. Solver

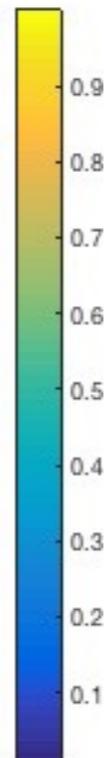
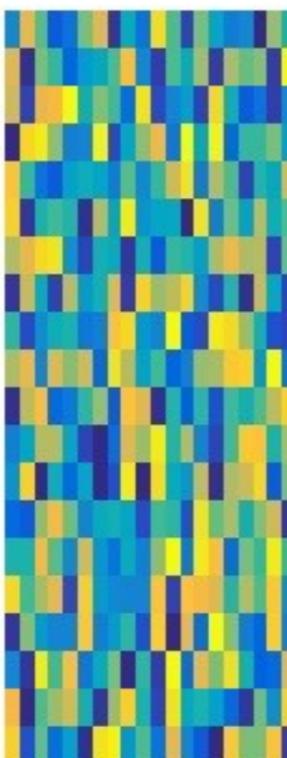
a



b

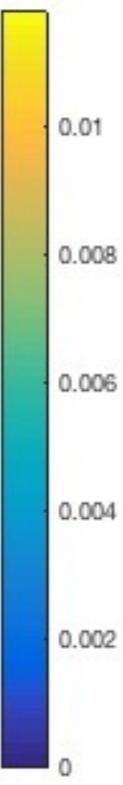
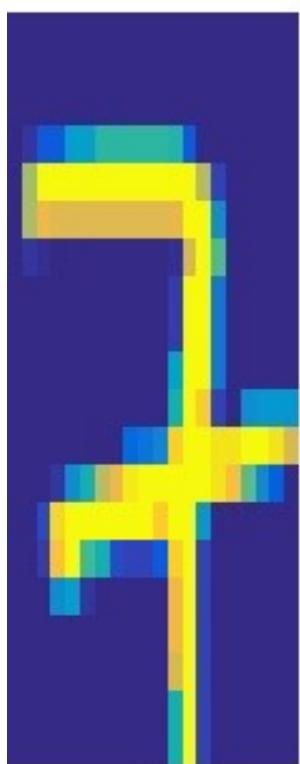


$v_1 \leftarrow \text{noise}$



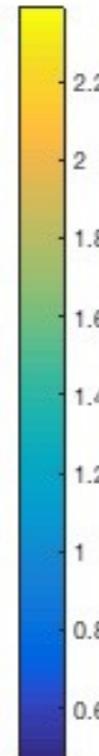
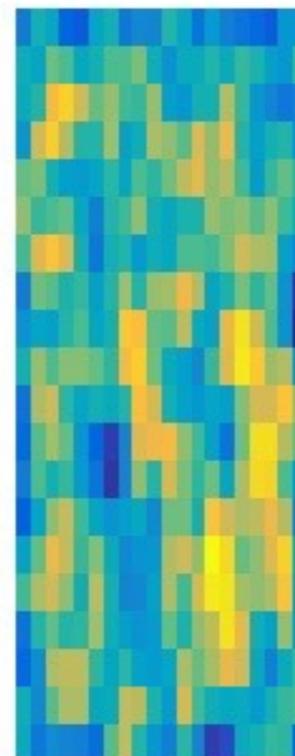
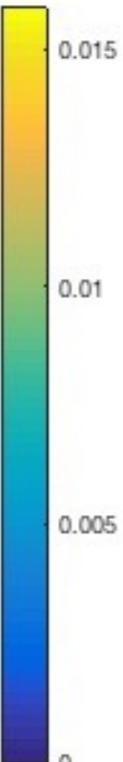
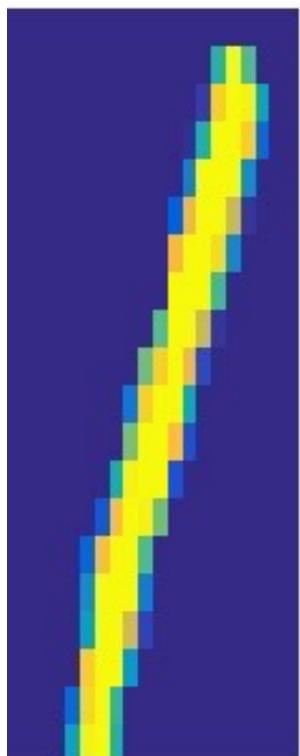
Very Fast EMD Approx. Solver

a



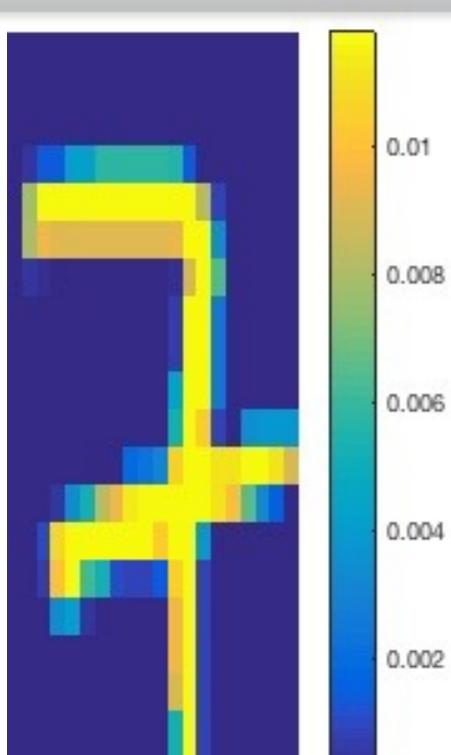
b

Kv_1

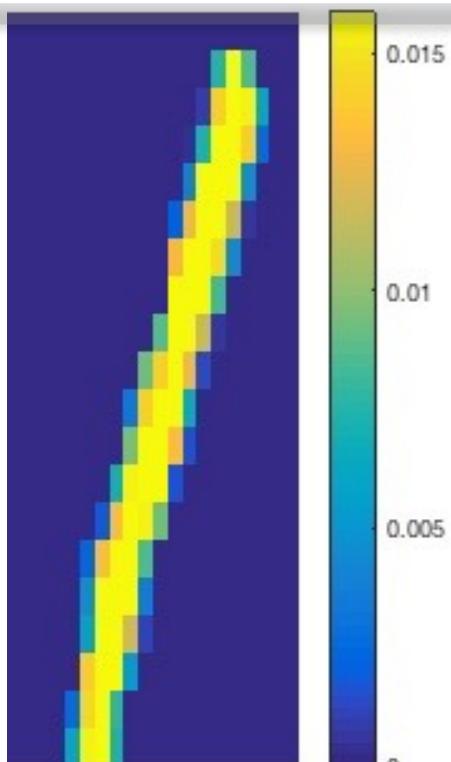


Very Fast EMD Approx. Solver

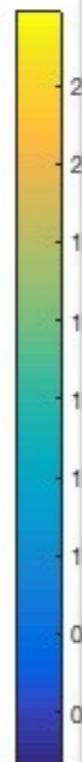
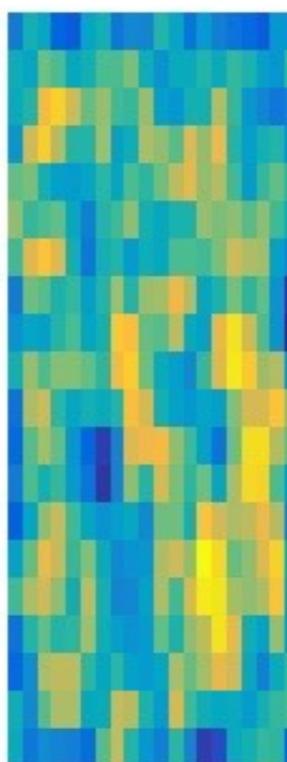
a



b

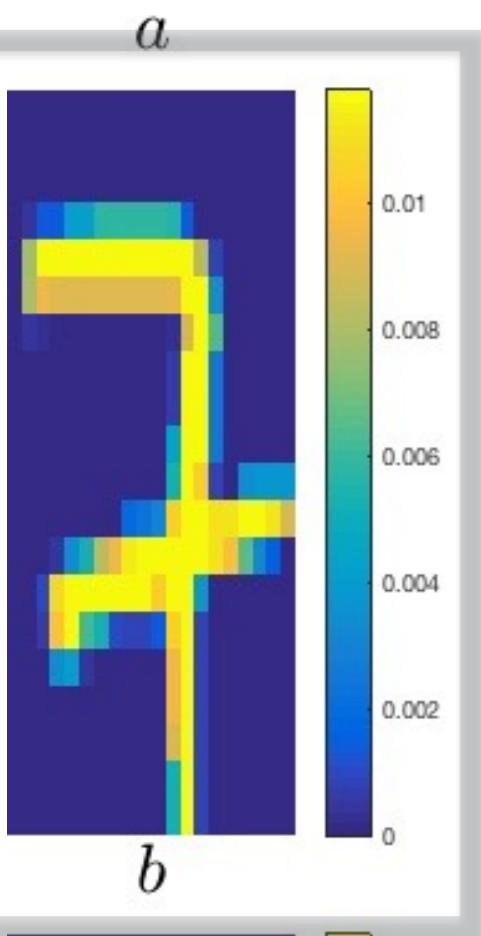


$K_{\alpha\beta}$

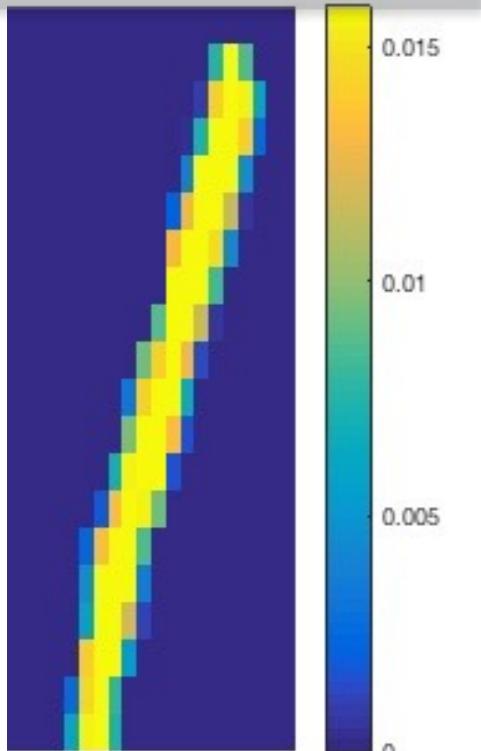


Very Fast EMD Approx. Solver

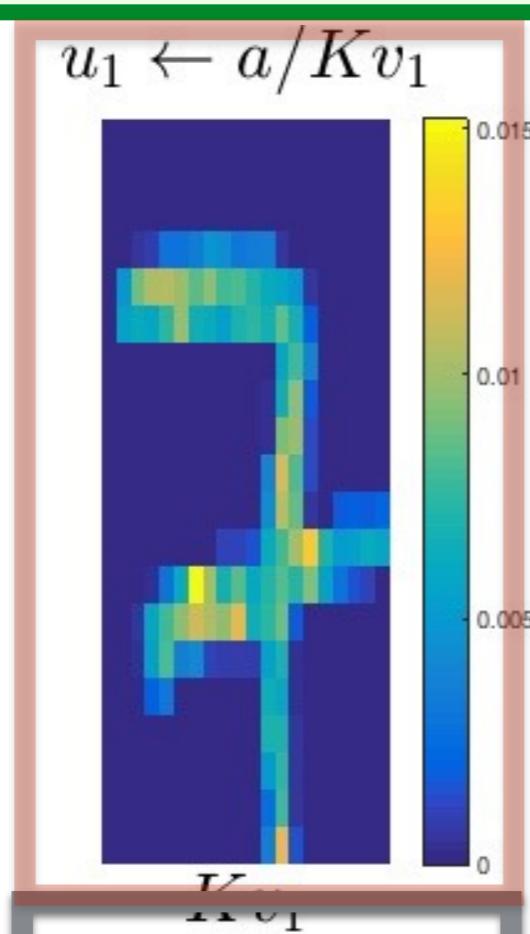
a



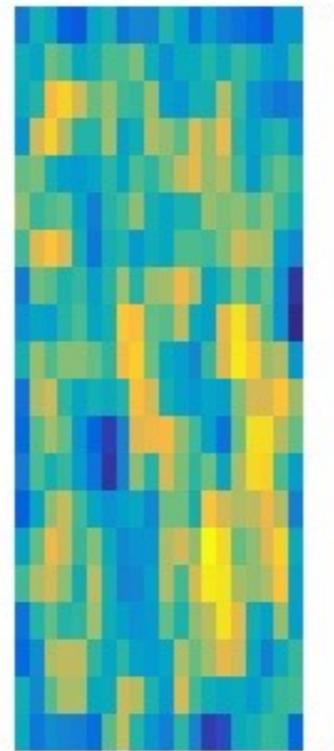
b



$u_1 \leftarrow a/Kv_1$

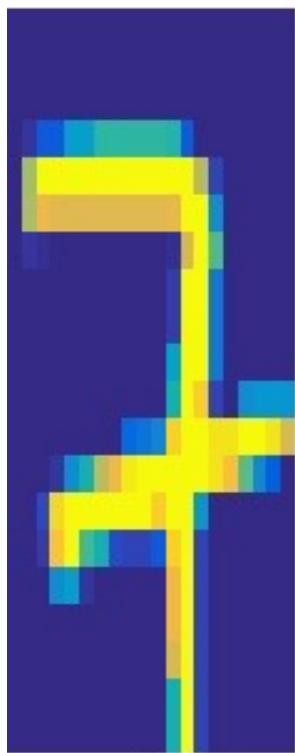


Kv_1

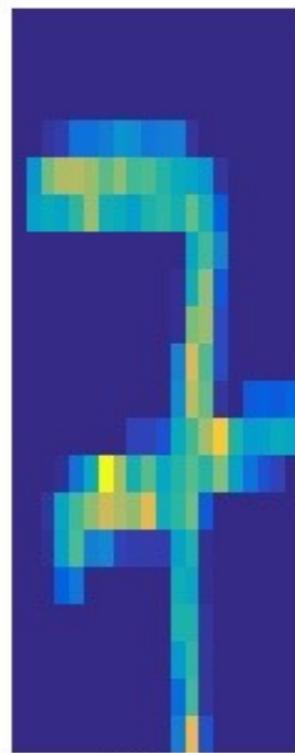


Very Fast EMD Approx. Solver

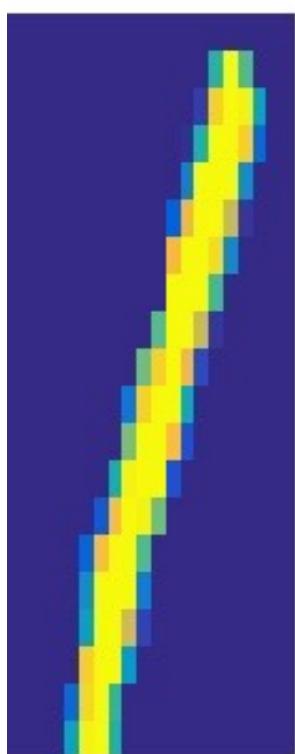
a



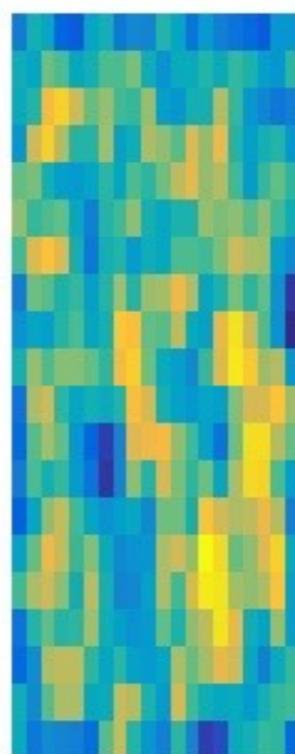
$u_1 \leftarrow a/Kv_1$



b



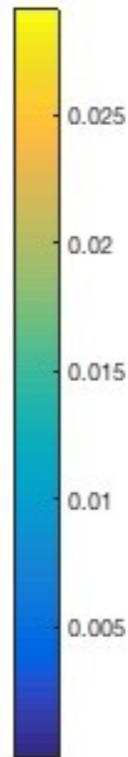
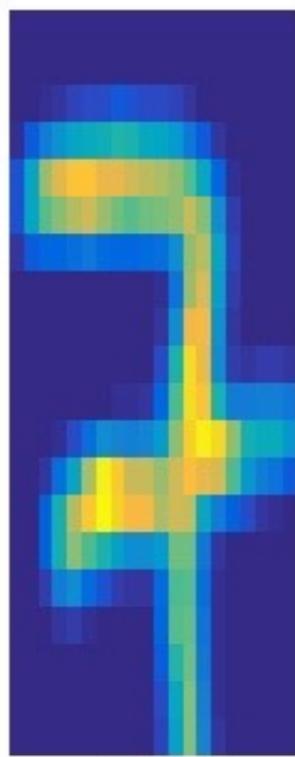
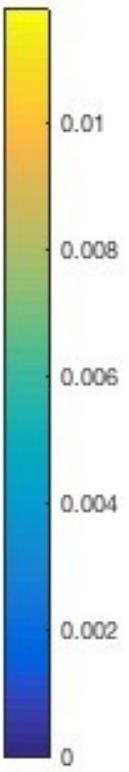
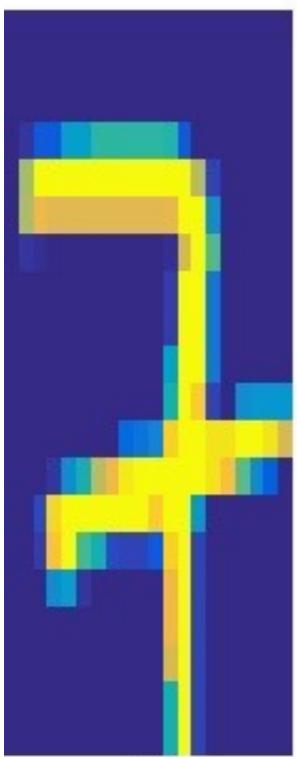
Kv_1



Very Fast EMD Approx. Solver

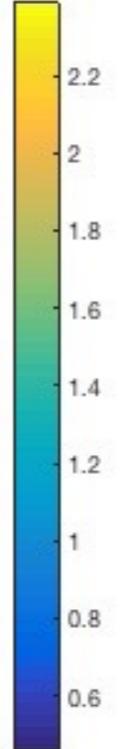
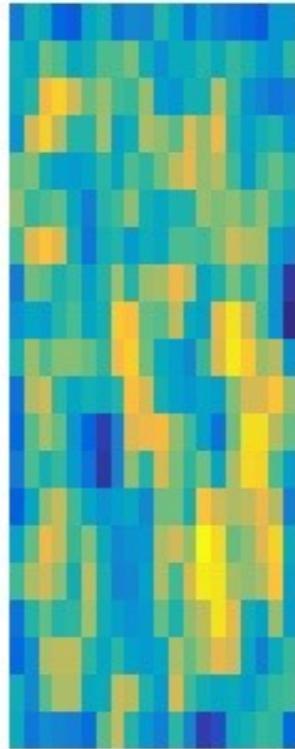
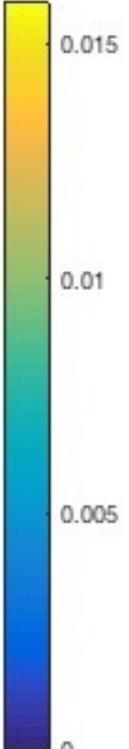
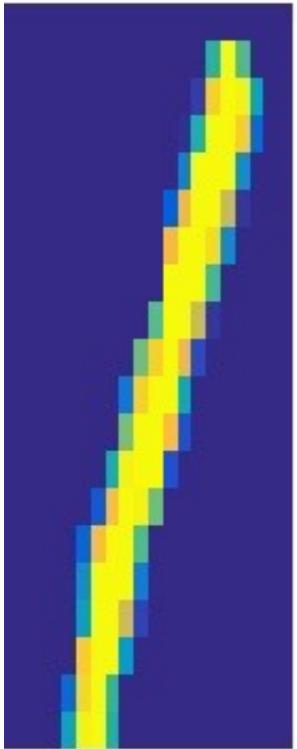
a

Ku_1



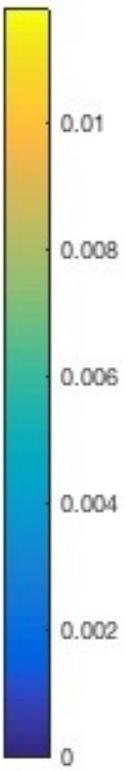
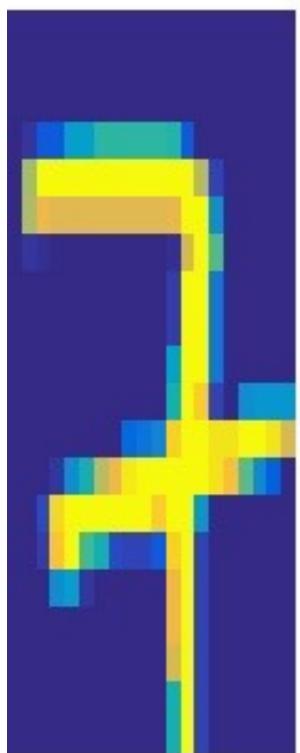
b

Kv_1

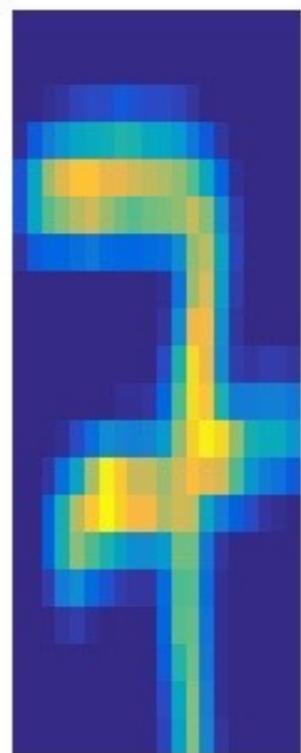


Very Fast EMD Approx. Solver

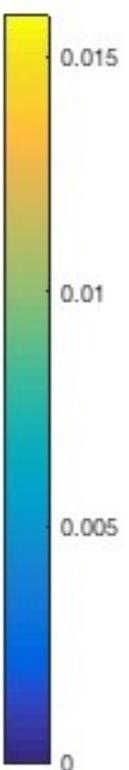
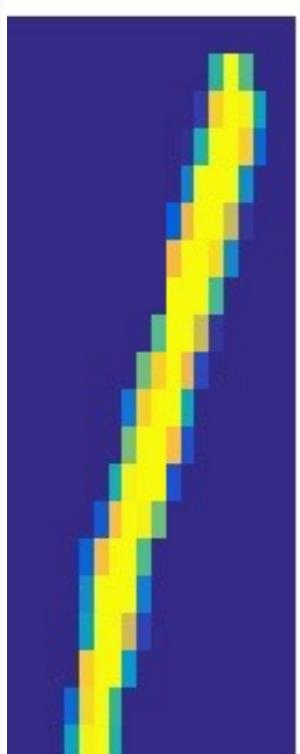
a



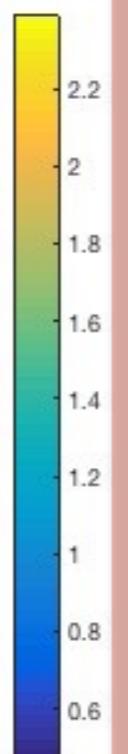
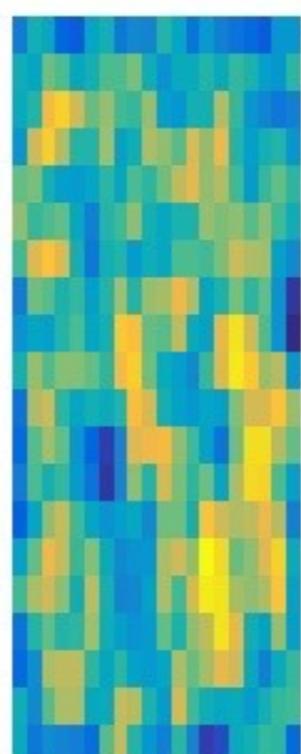
$K u_1$



b

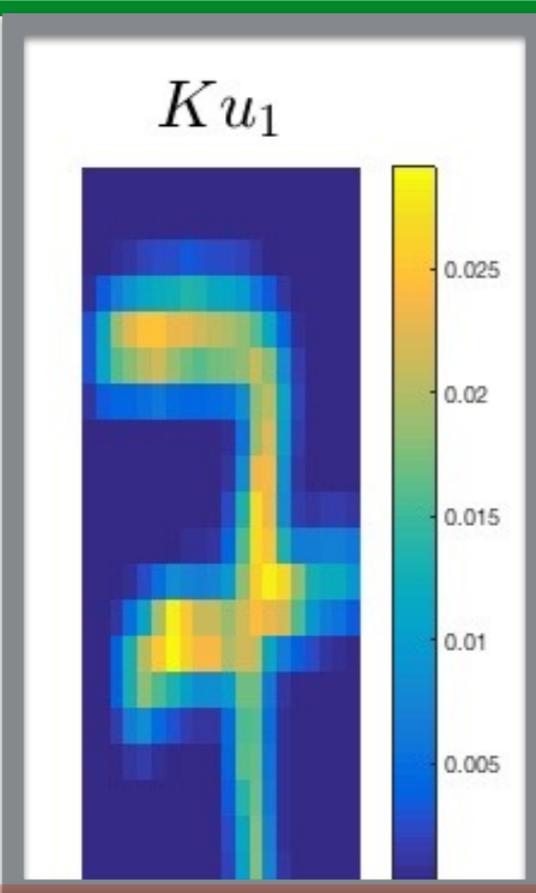
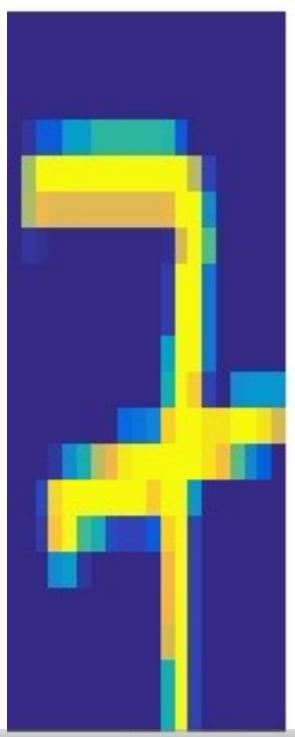


$K v_1$

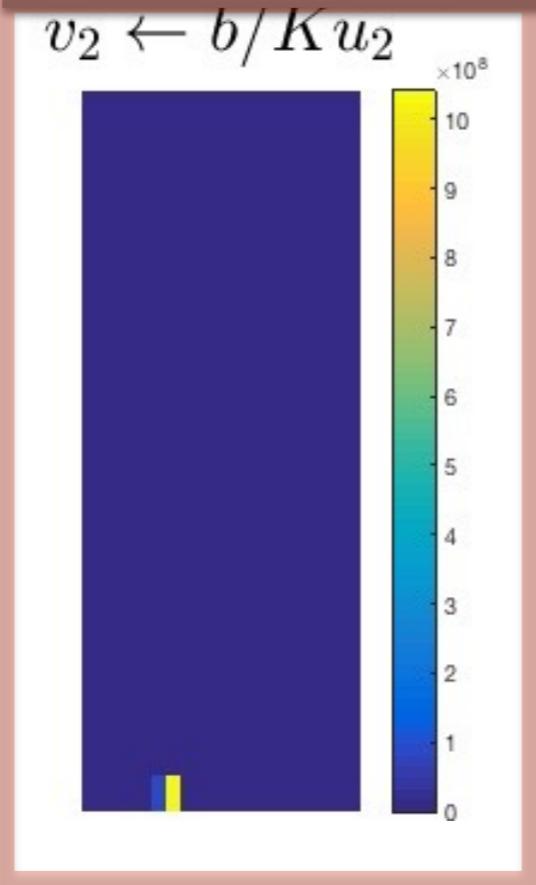
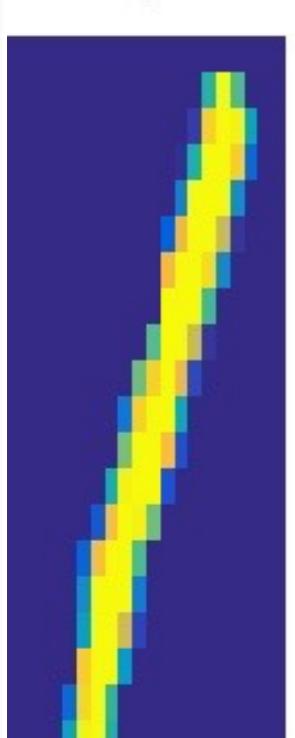


Very Fast EMD Approx. Solver

a

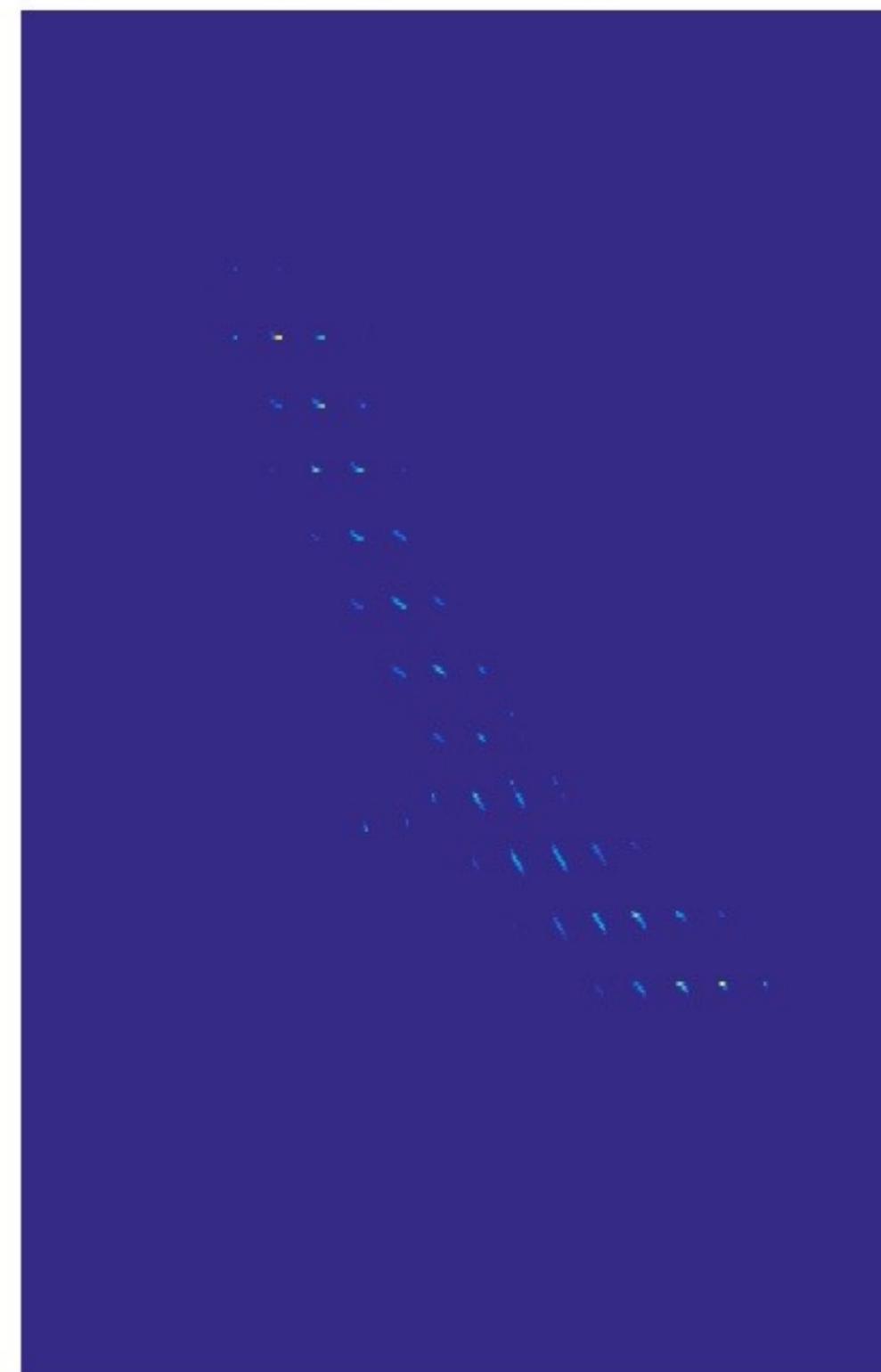


b



$$P_1 = D(u_1)KD(v_1)$$

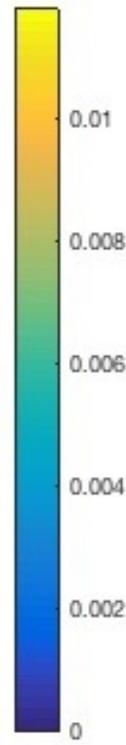
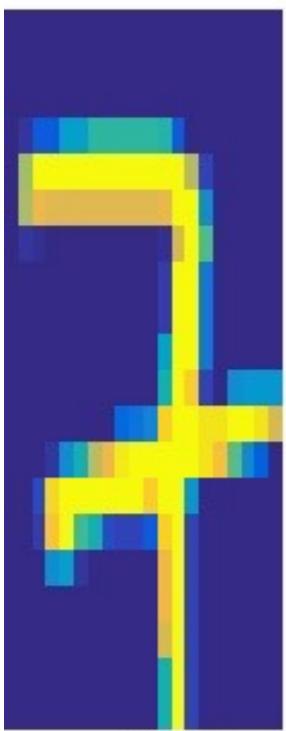
$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$



Very Fast EMD Approx. Solver

a

*Ku*₁



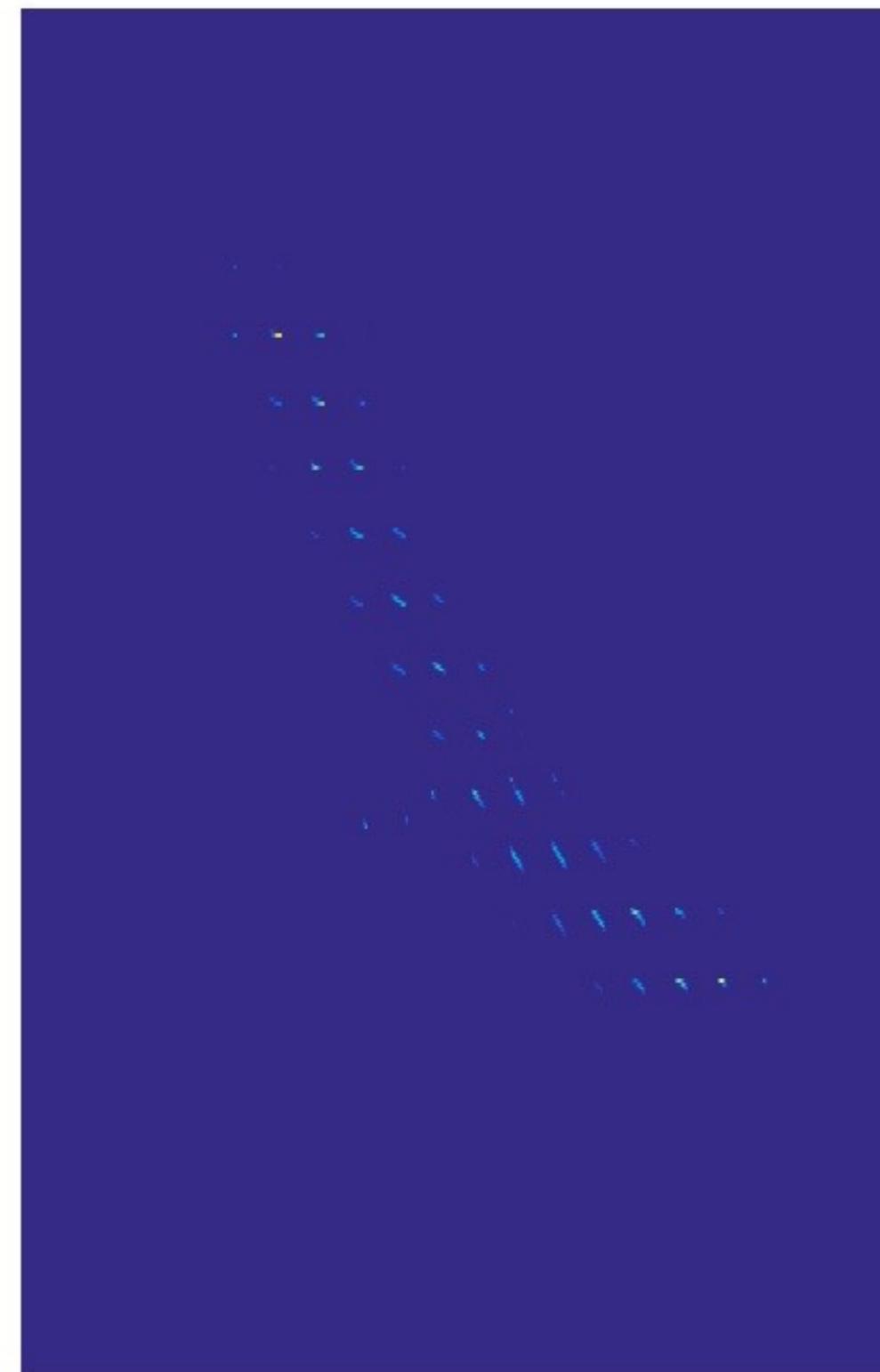
b

*v*₂ ← *b*/*Ku*₂



$$P_1 = D(u_1)K D(v_1)$$

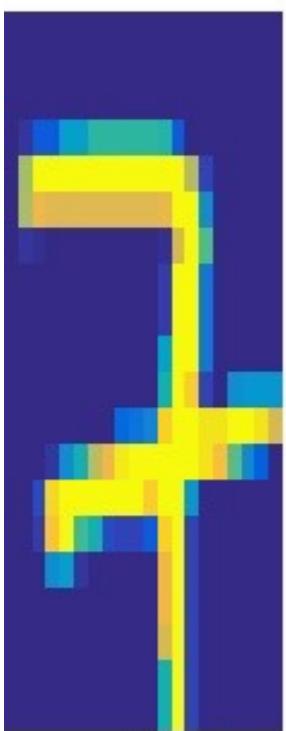
$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$



Very Fast EMD Approx. Solver

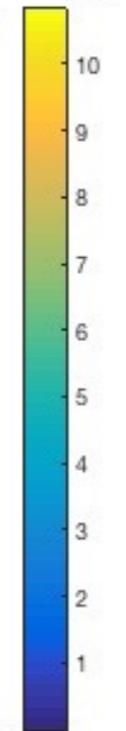
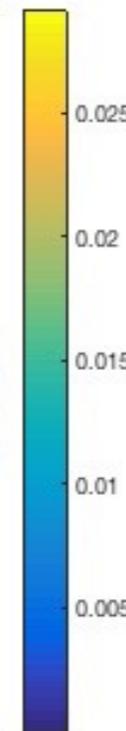
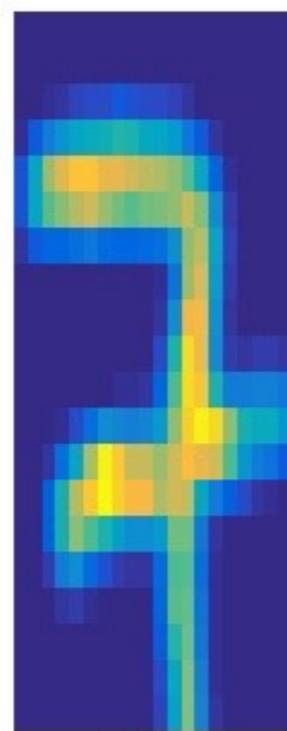
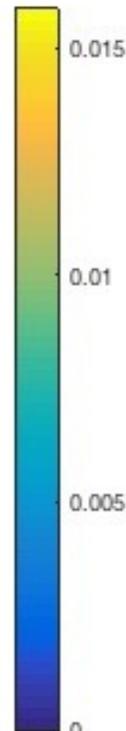
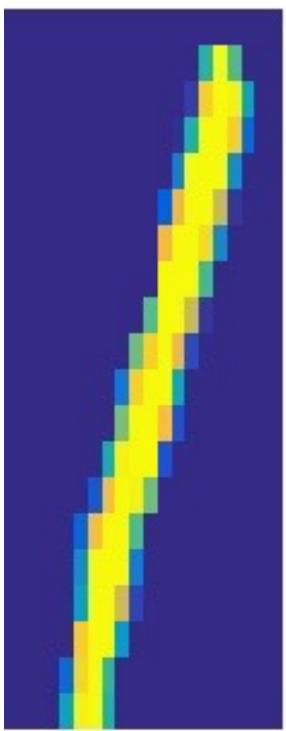
a

*Ku*₁



b

*Kv*₂



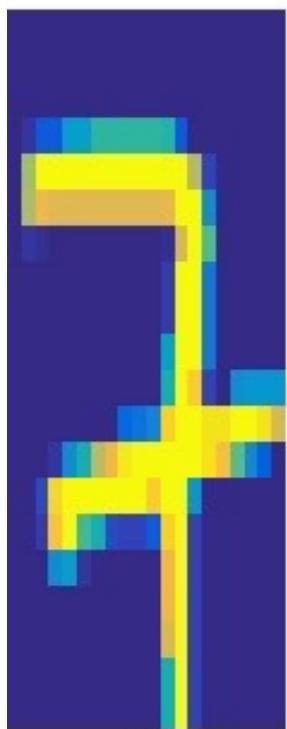
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$

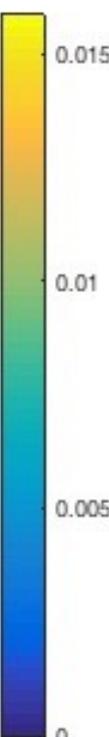


Very Fast EMD Approx. Solver

a



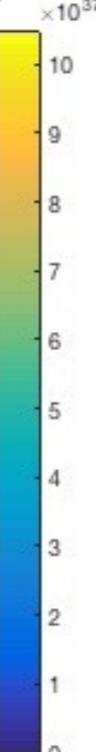
b



$$u_2 \leftarrow a/Kv_2$$

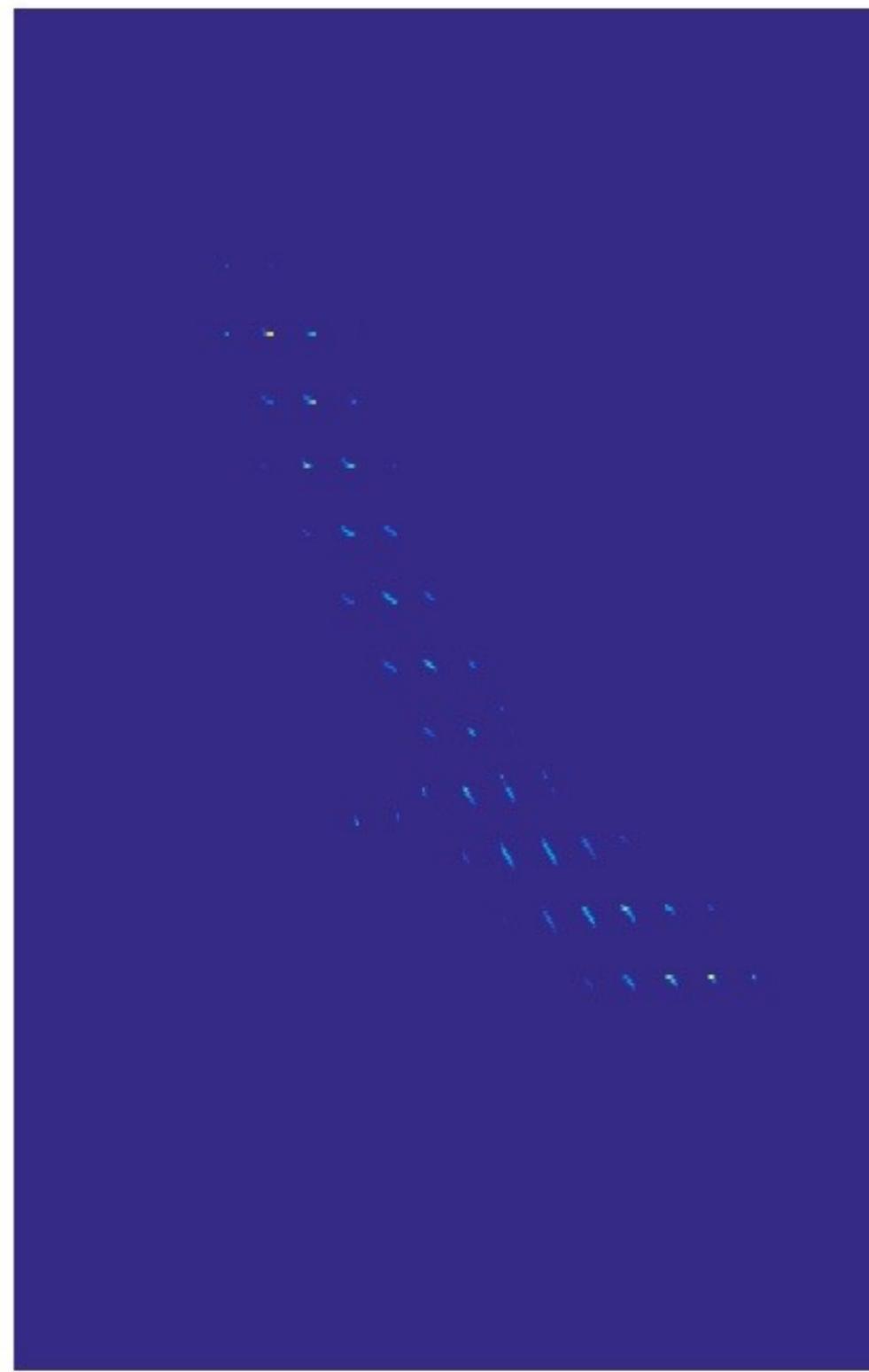


Kv_2



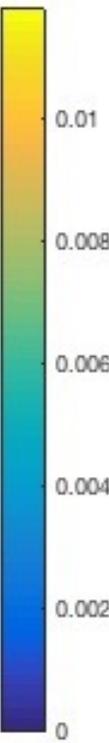
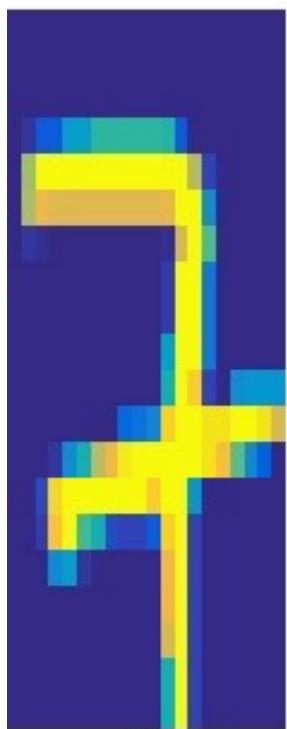
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$

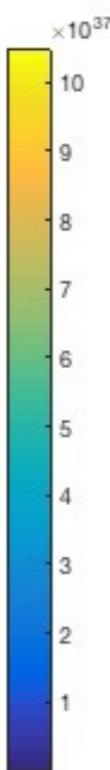


Very Fast EMD Approx. Solver

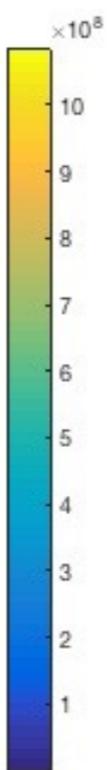
a



Ku_2



Kv_2



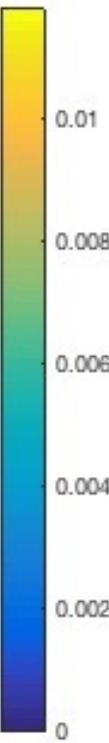
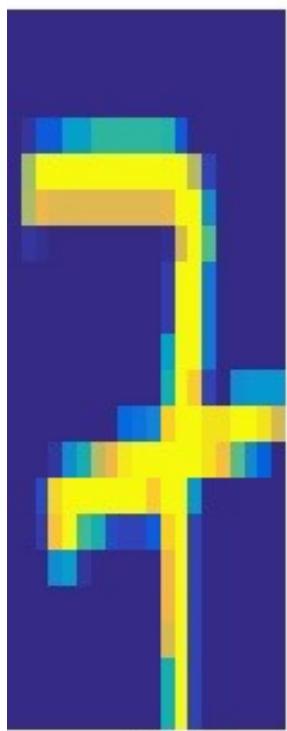
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$

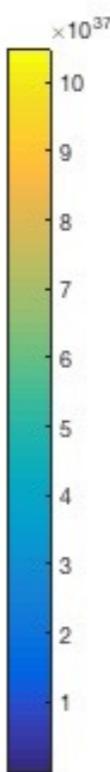


Very Fast EMD Approx. Solver

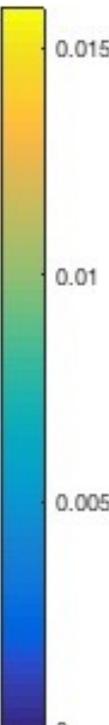
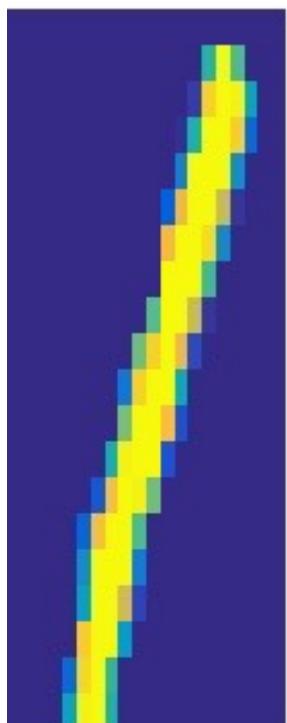
a



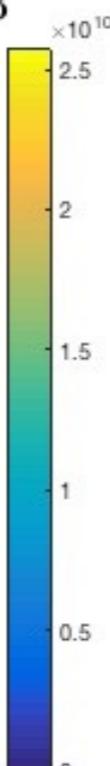
Ku₂



b

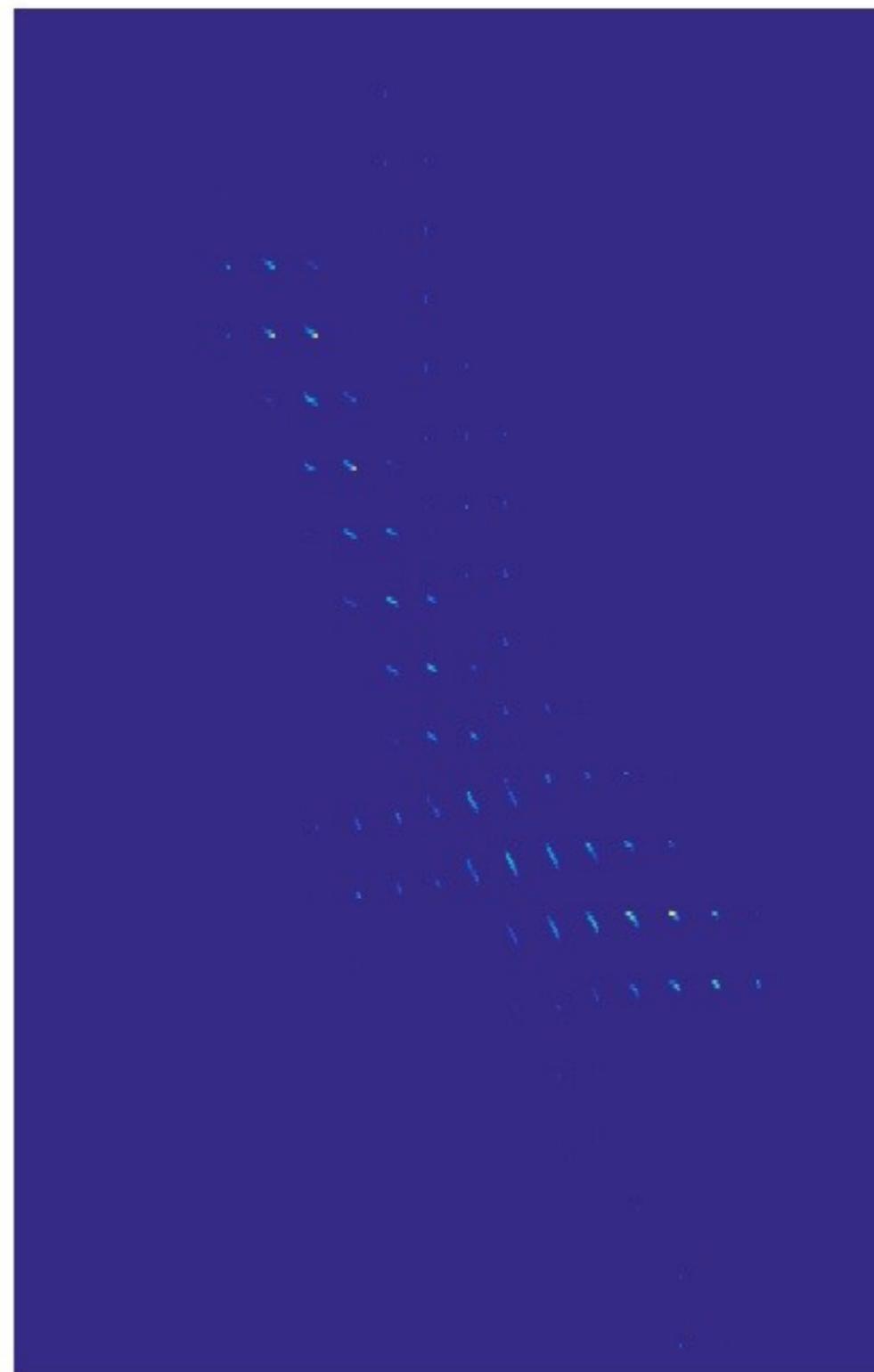


$v_3 \leftarrow b/Ku_3$



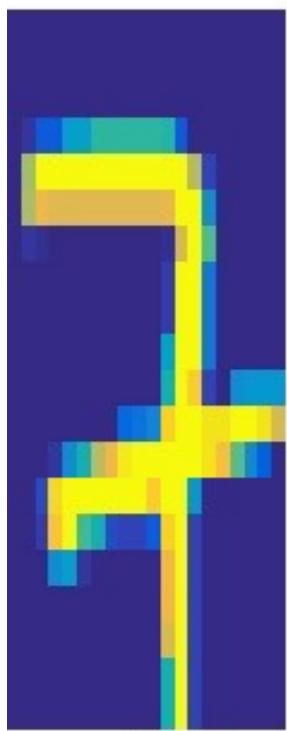
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$



Very Fast EMD Approx. Solver

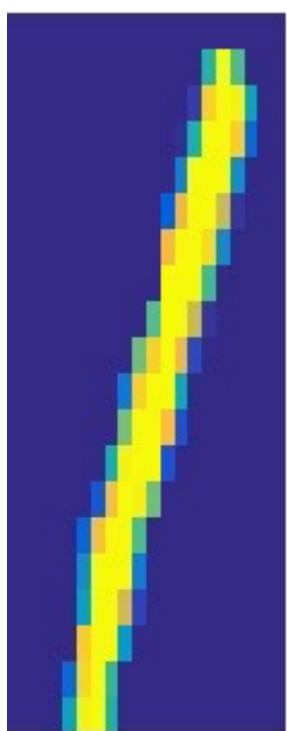
a



Ku_2



b

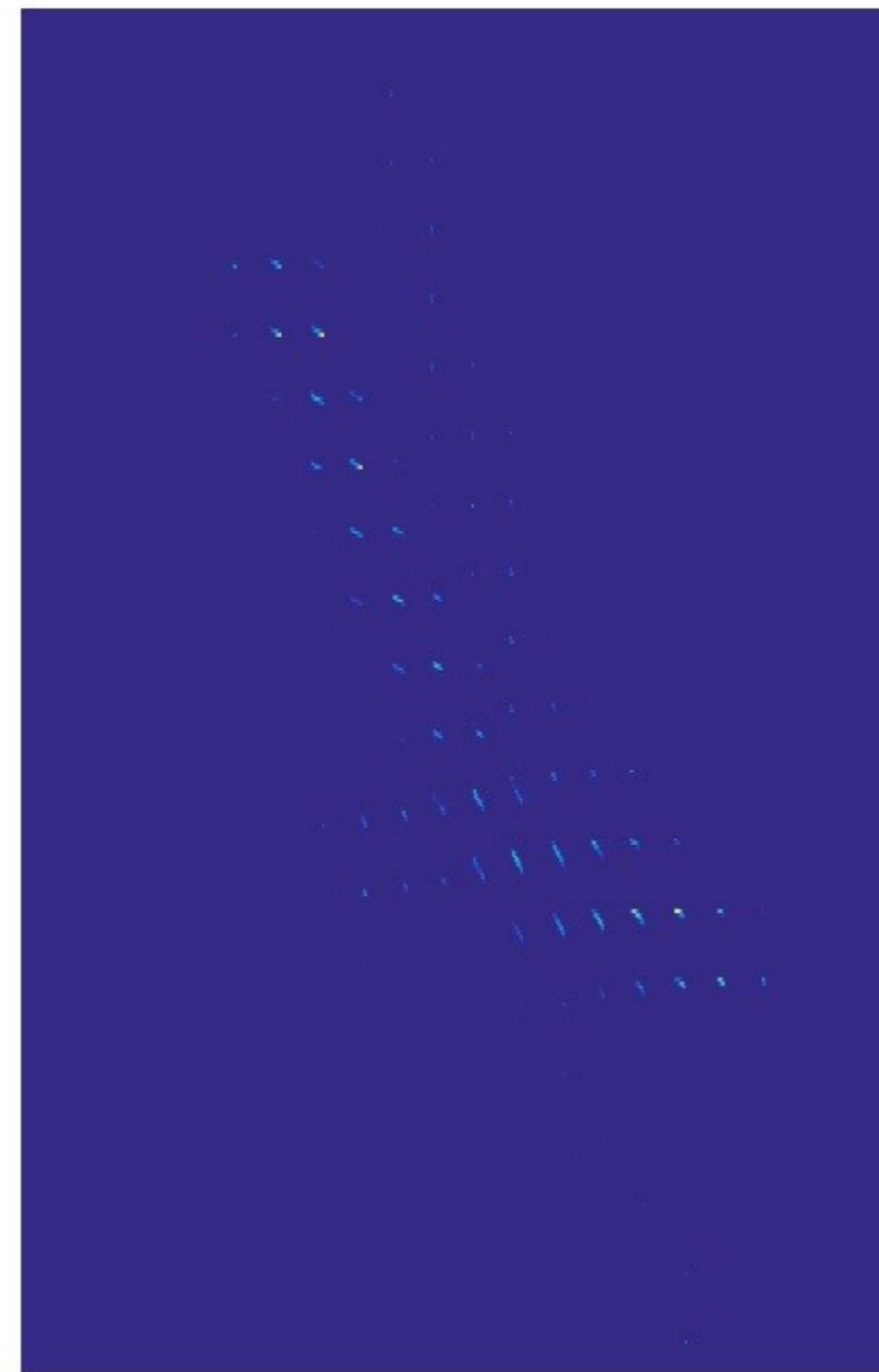


Kv_3



$$P_2 = D(u_2)KD(v_2)$$

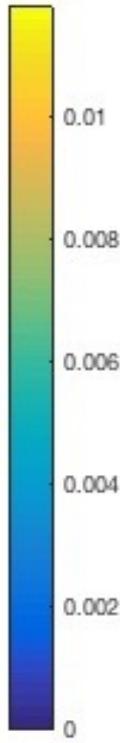
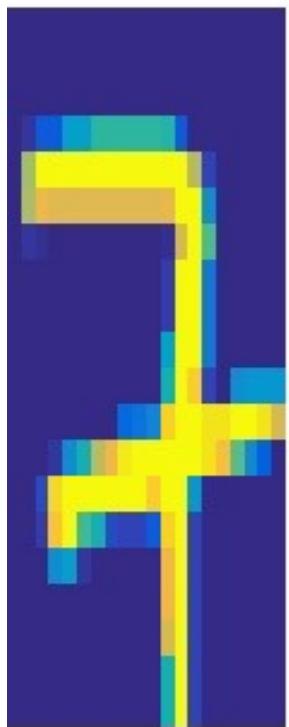
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$



Very Fast EMD Approx. Solver

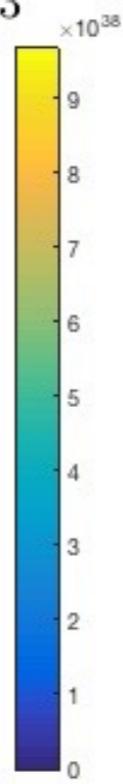
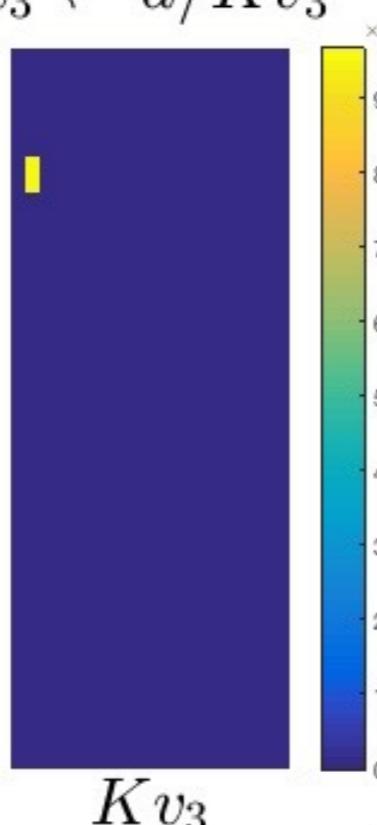
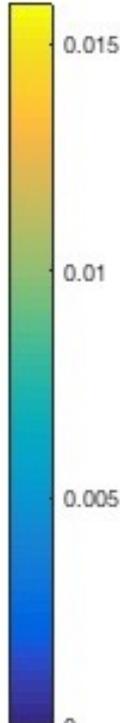
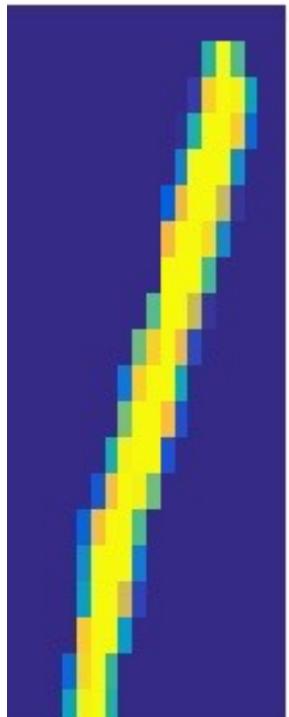
a

$$u_3 \leftarrow a/Kv_3$$



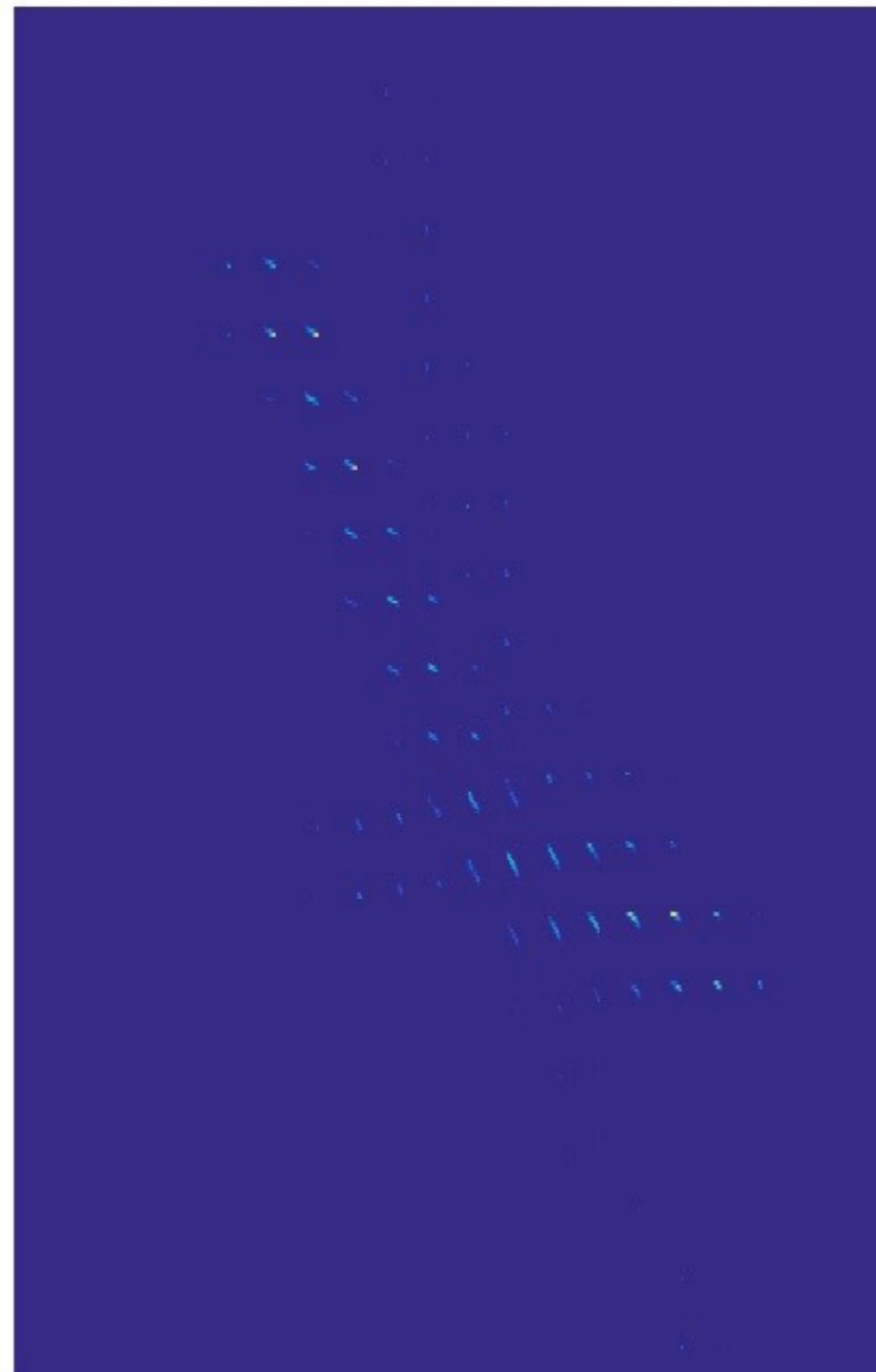
b

$$Kv_3$$



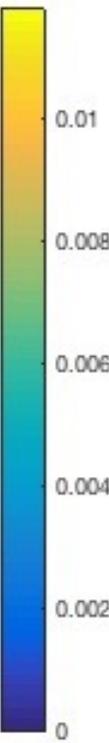
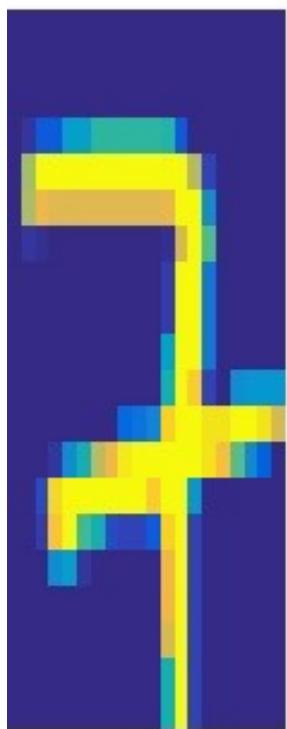
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

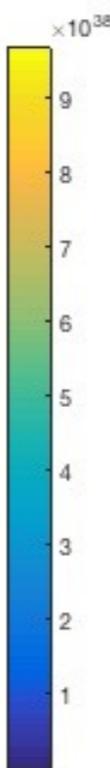


Very Fast EMD Approx. Solver

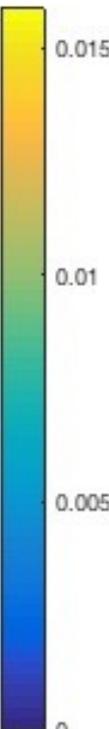
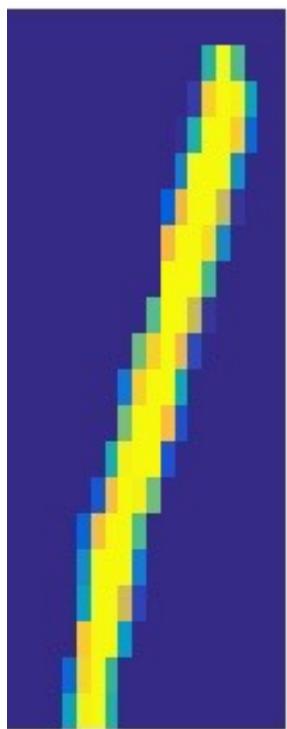
a



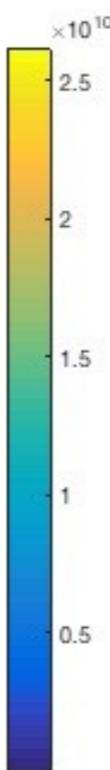
*Ku*₃



b

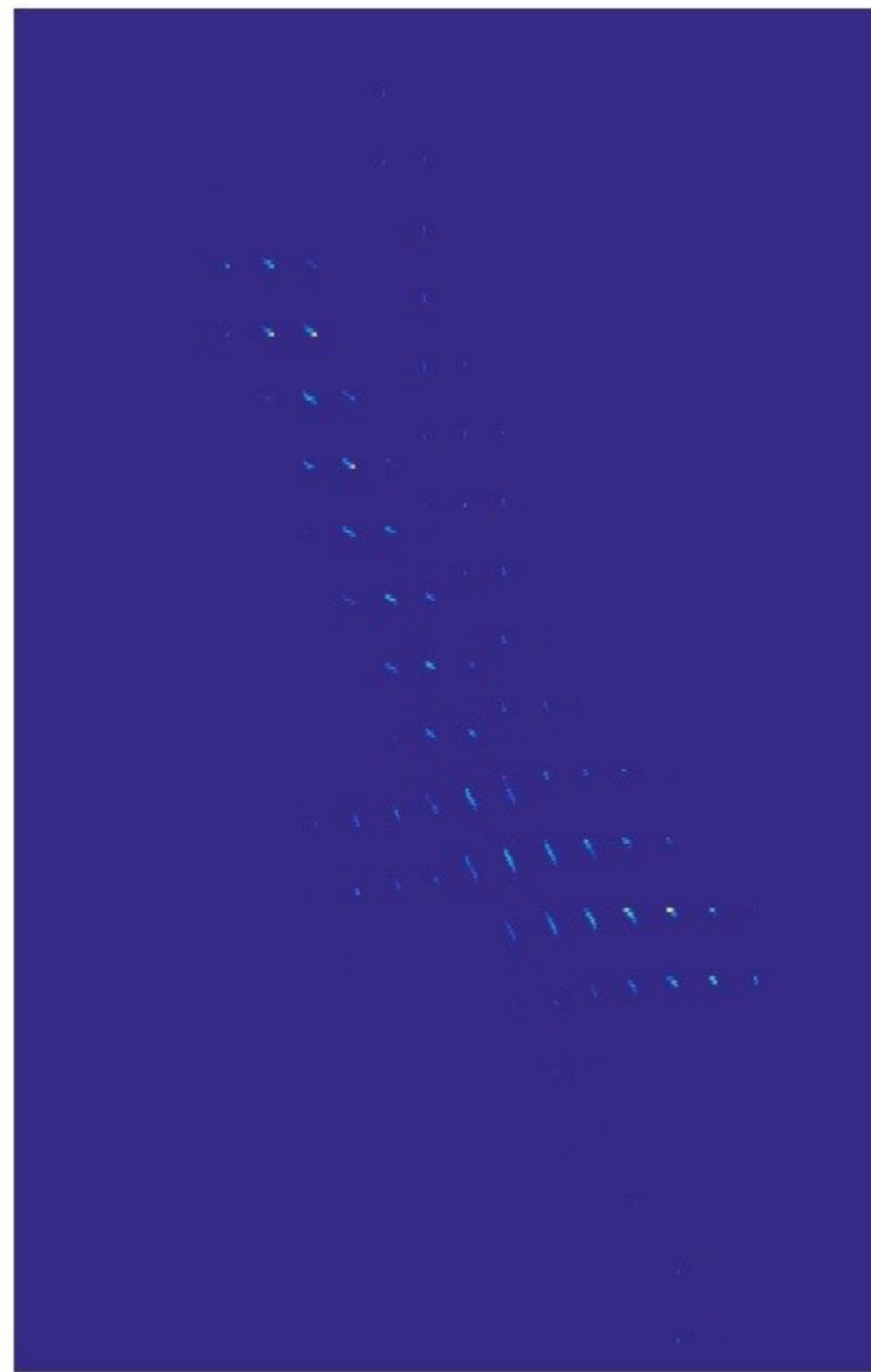


*Kv*₃



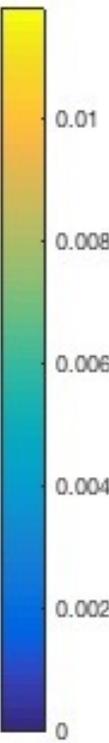
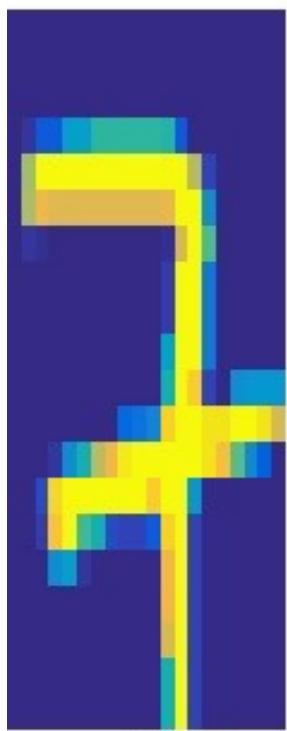
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

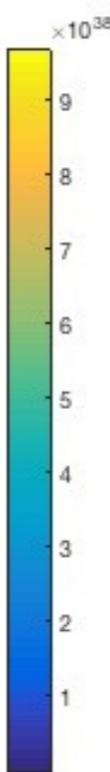


Very Fast EMD Approx. Solver

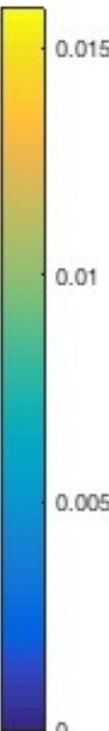
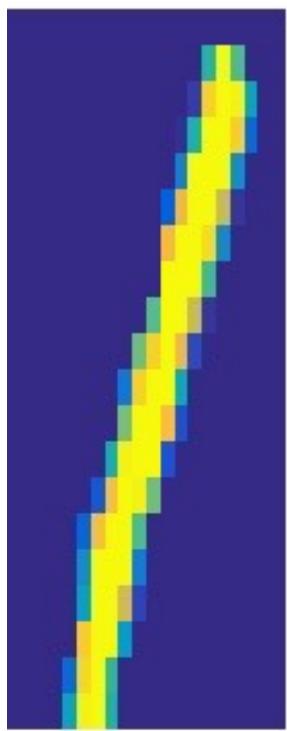
a



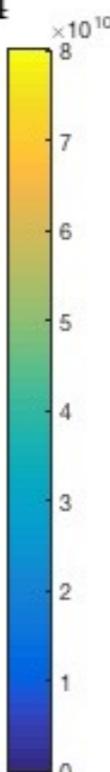
Ku₃



b

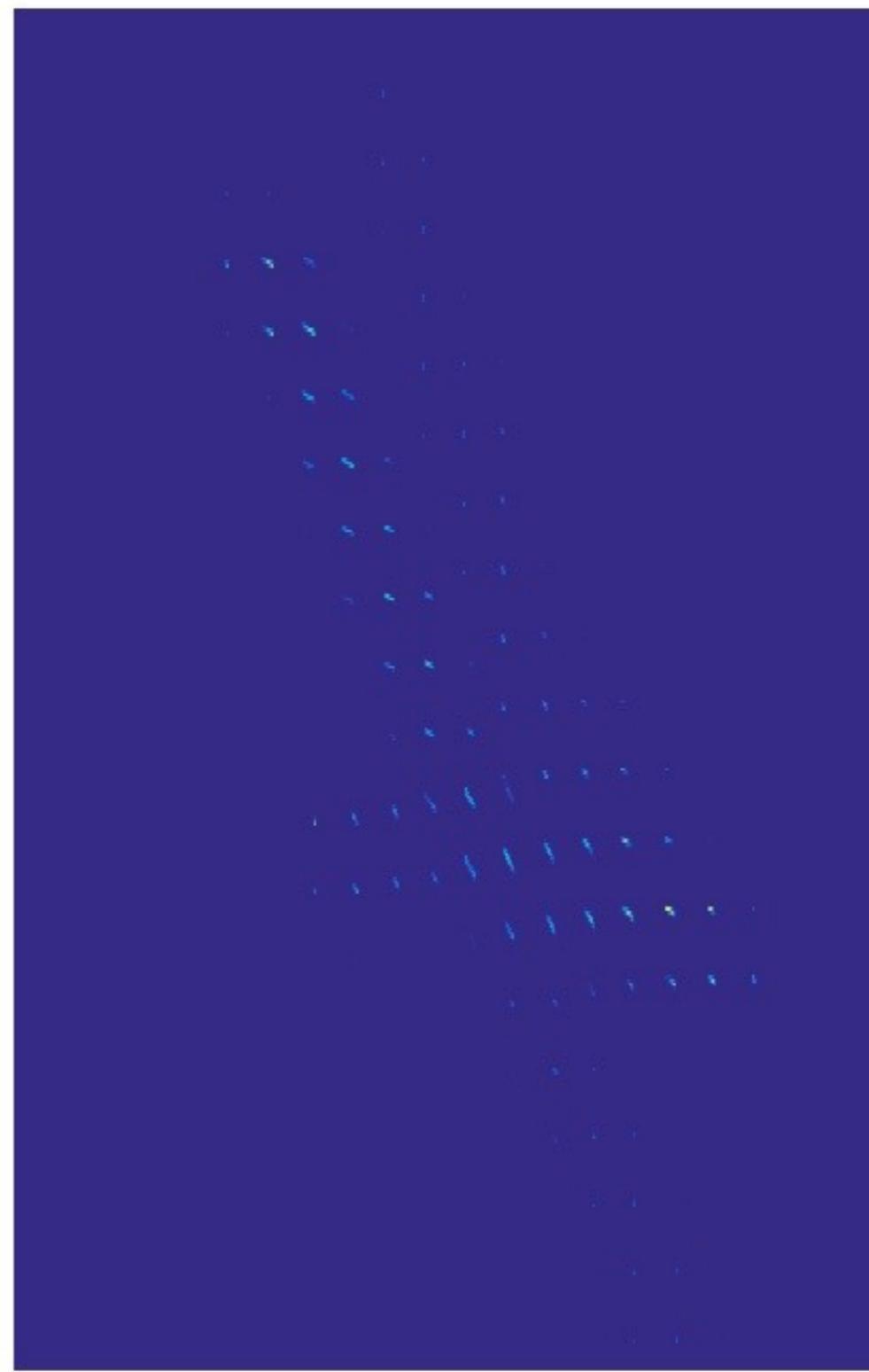


$v_4 \leftarrow b/Ku_4$



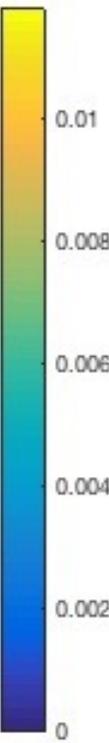
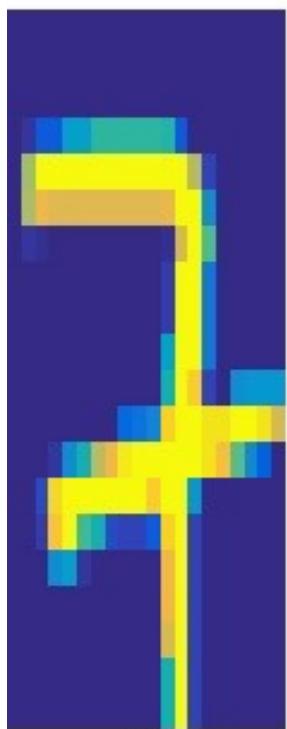
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

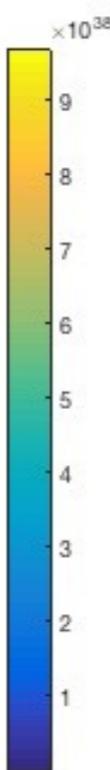


Very Fast EMD Approx. Solver

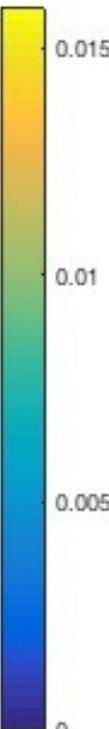
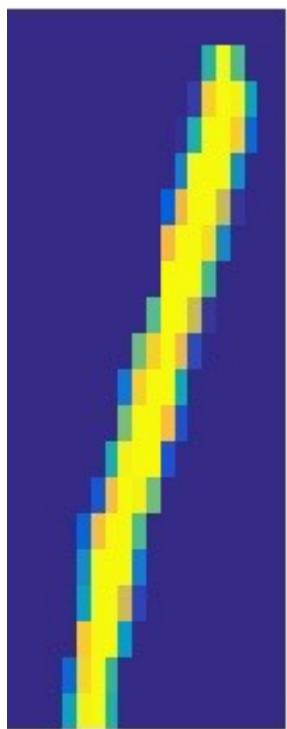
a



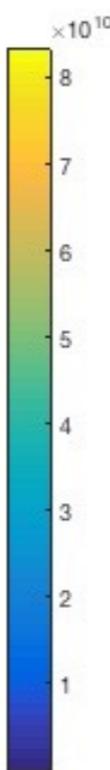
*Ku*₃



b

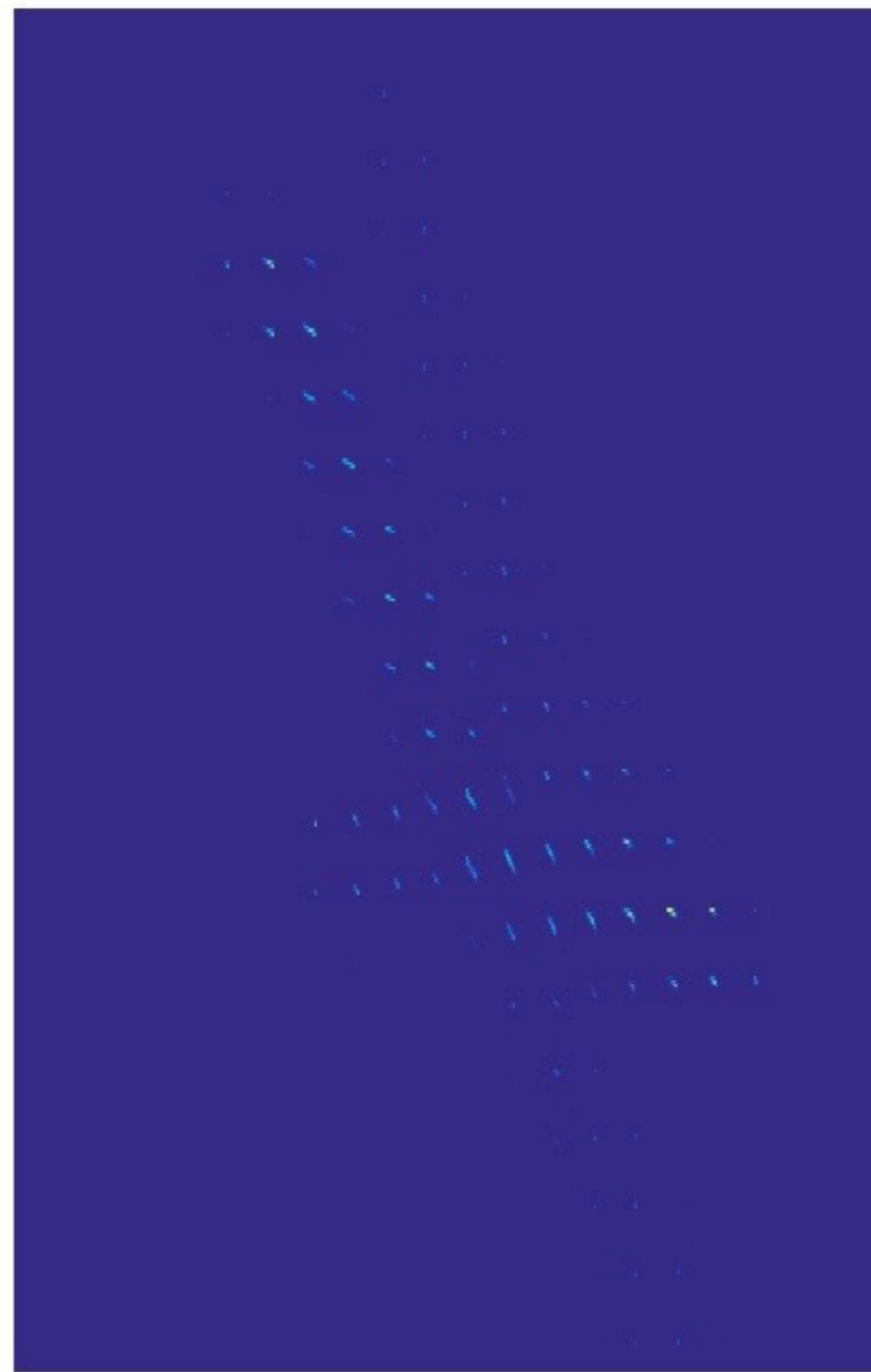


*Kv*₄



$$P_3 = D(u_3)KD(v_3)$$

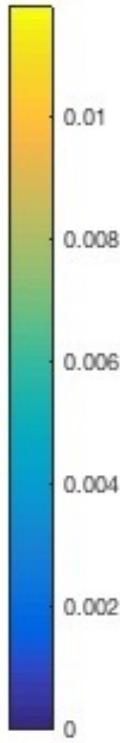
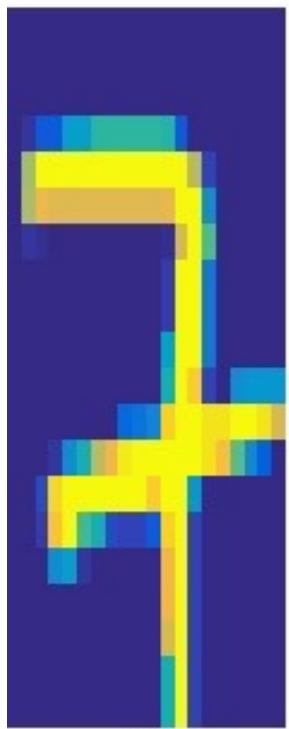
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$



Very Fast EMD Approx. Solver

a

$$u_4 \leftarrow a/Kv_4$$



b

$$Kv_4$$

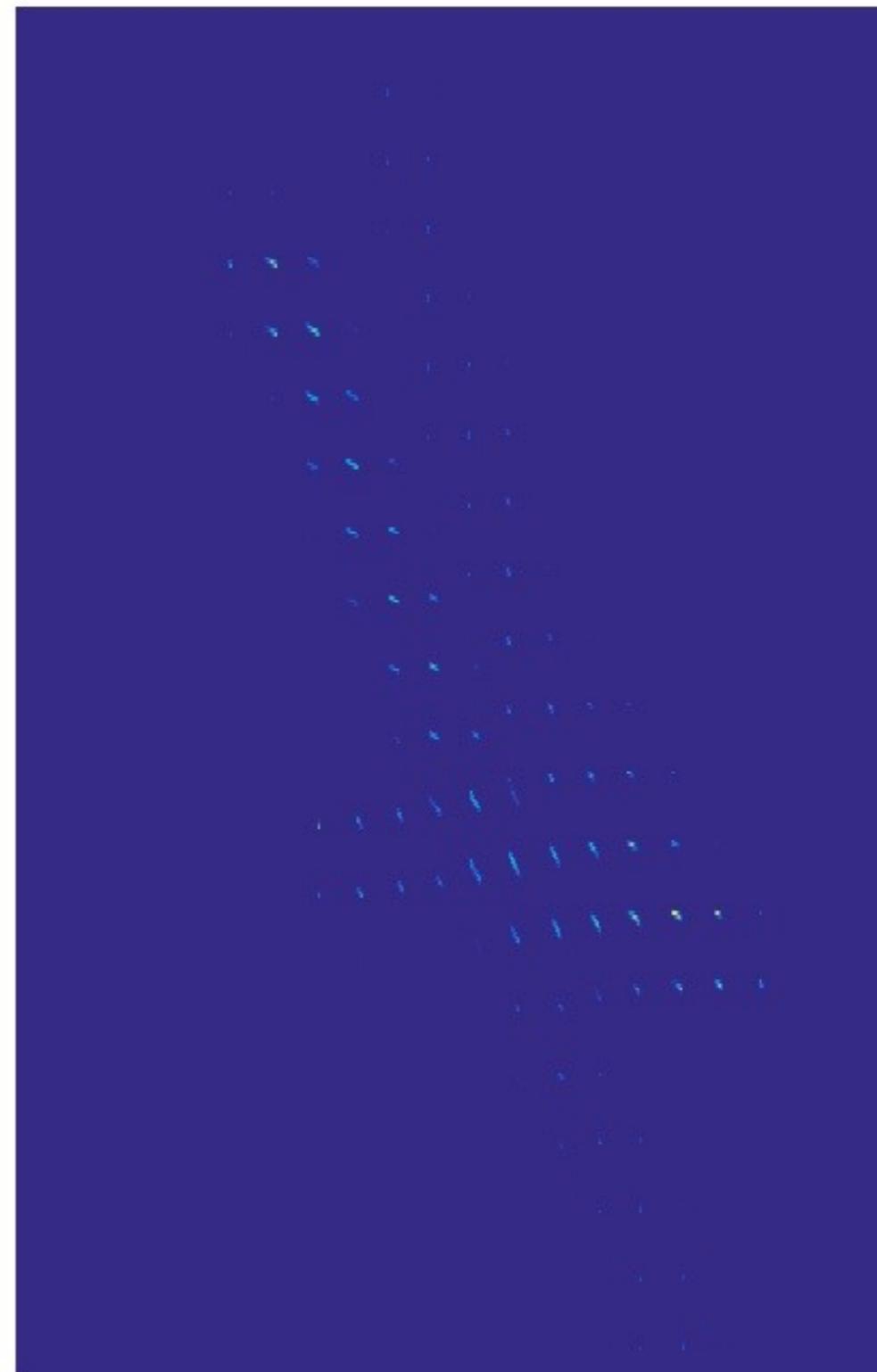


$\times 10^{39}$

$\times 10^{39}$

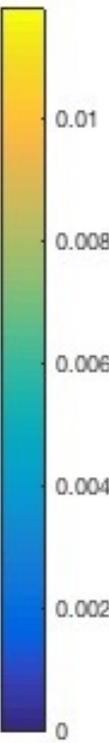
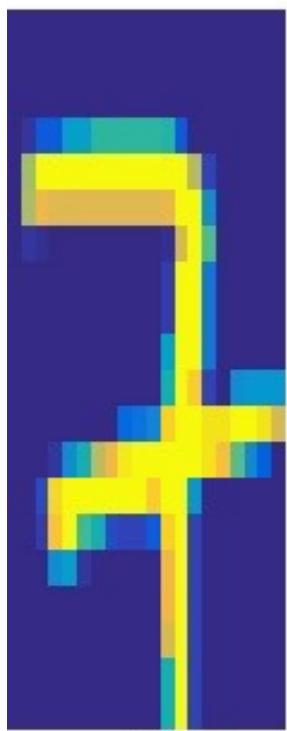
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

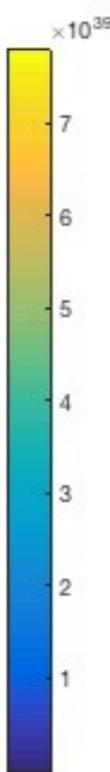


Very Fast EMD Approx. Solver

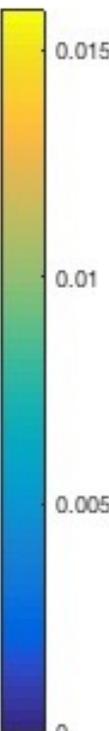
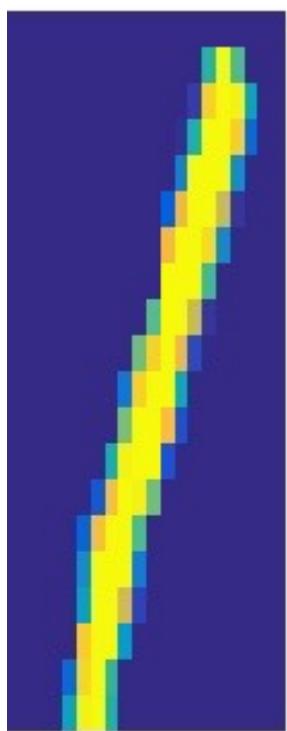
a



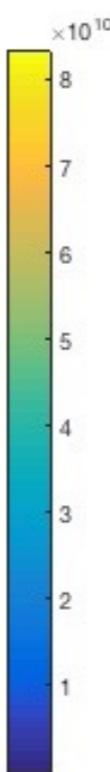
Ku_4



b

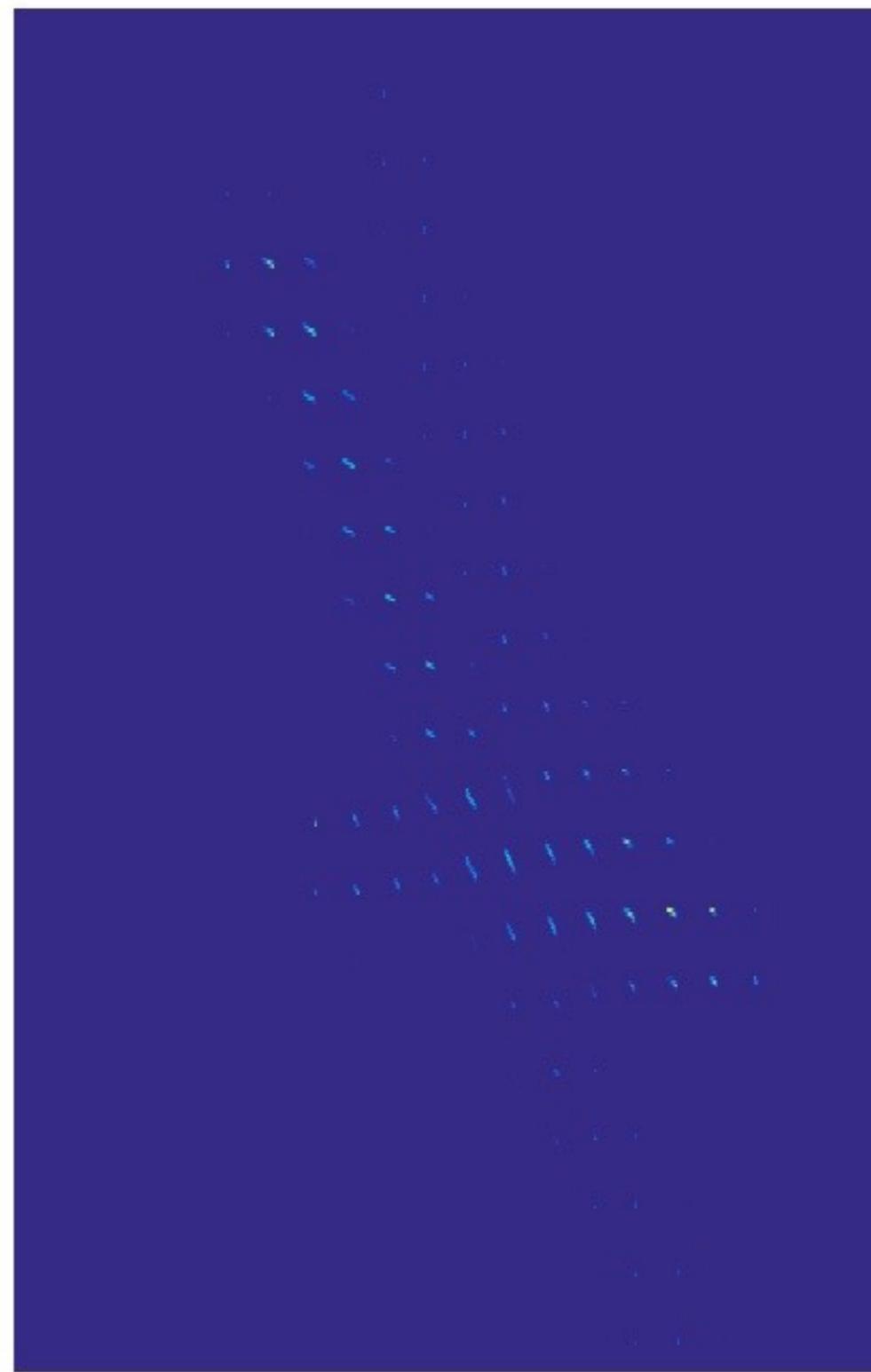


Kv_4



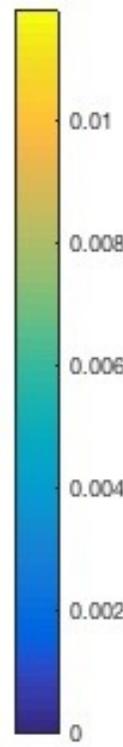
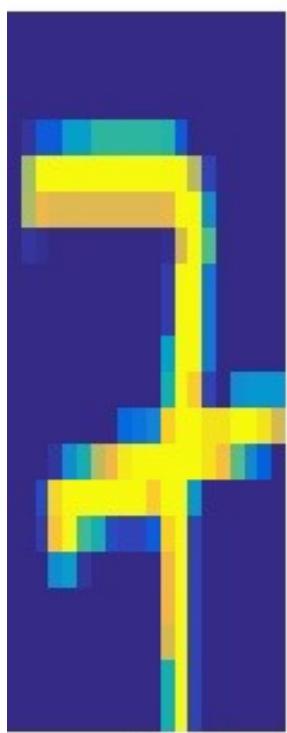
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

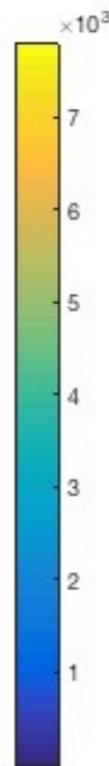


Very Fast EMD Approx. Solver

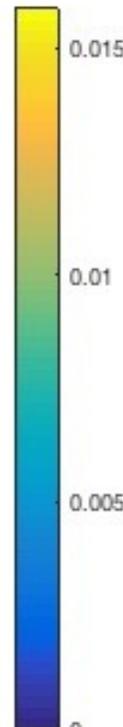
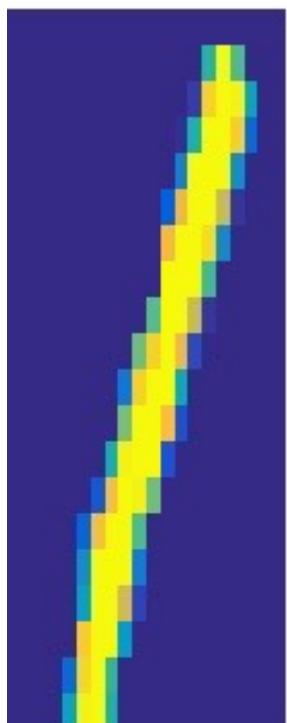
a



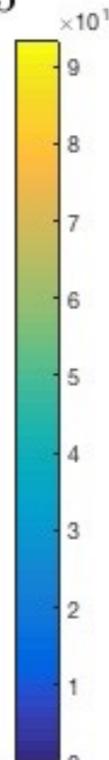
Ku_4



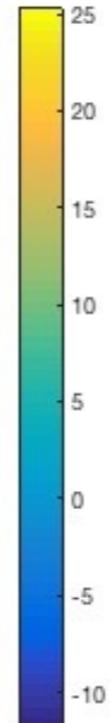
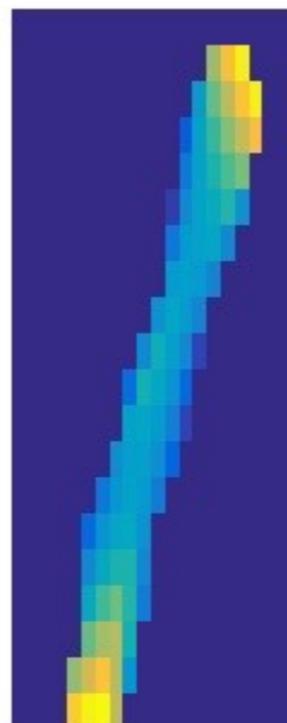
b



$v_5 \leftarrow b/Ku_5$

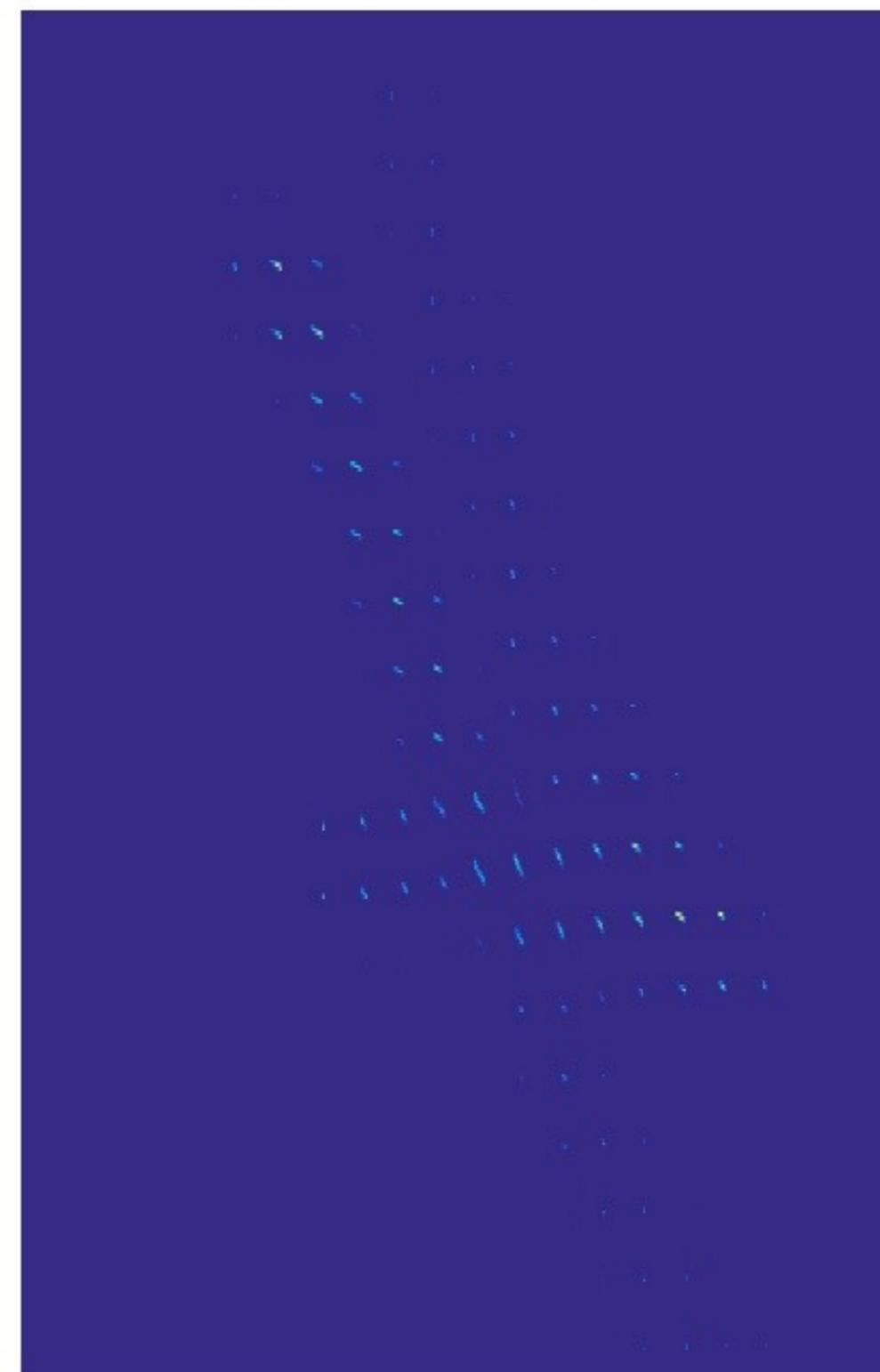


$\log(v_5)$



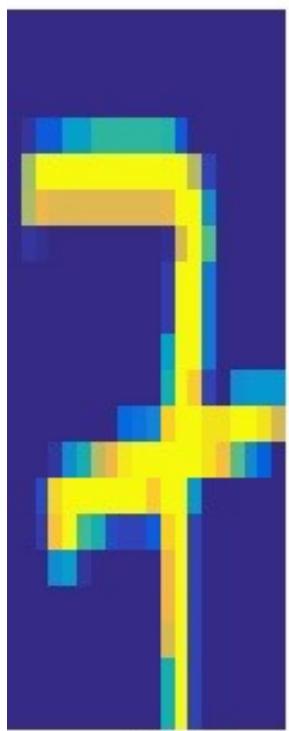
$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

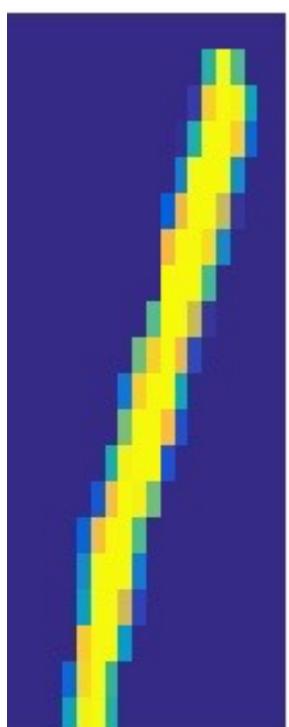


Very Fast EMD Approx. Solver

a



b



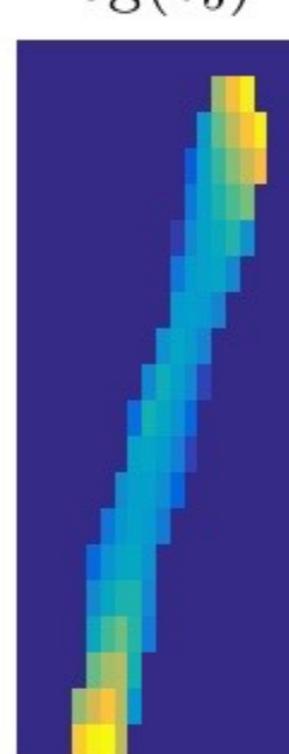
Ku_4



Kv_5

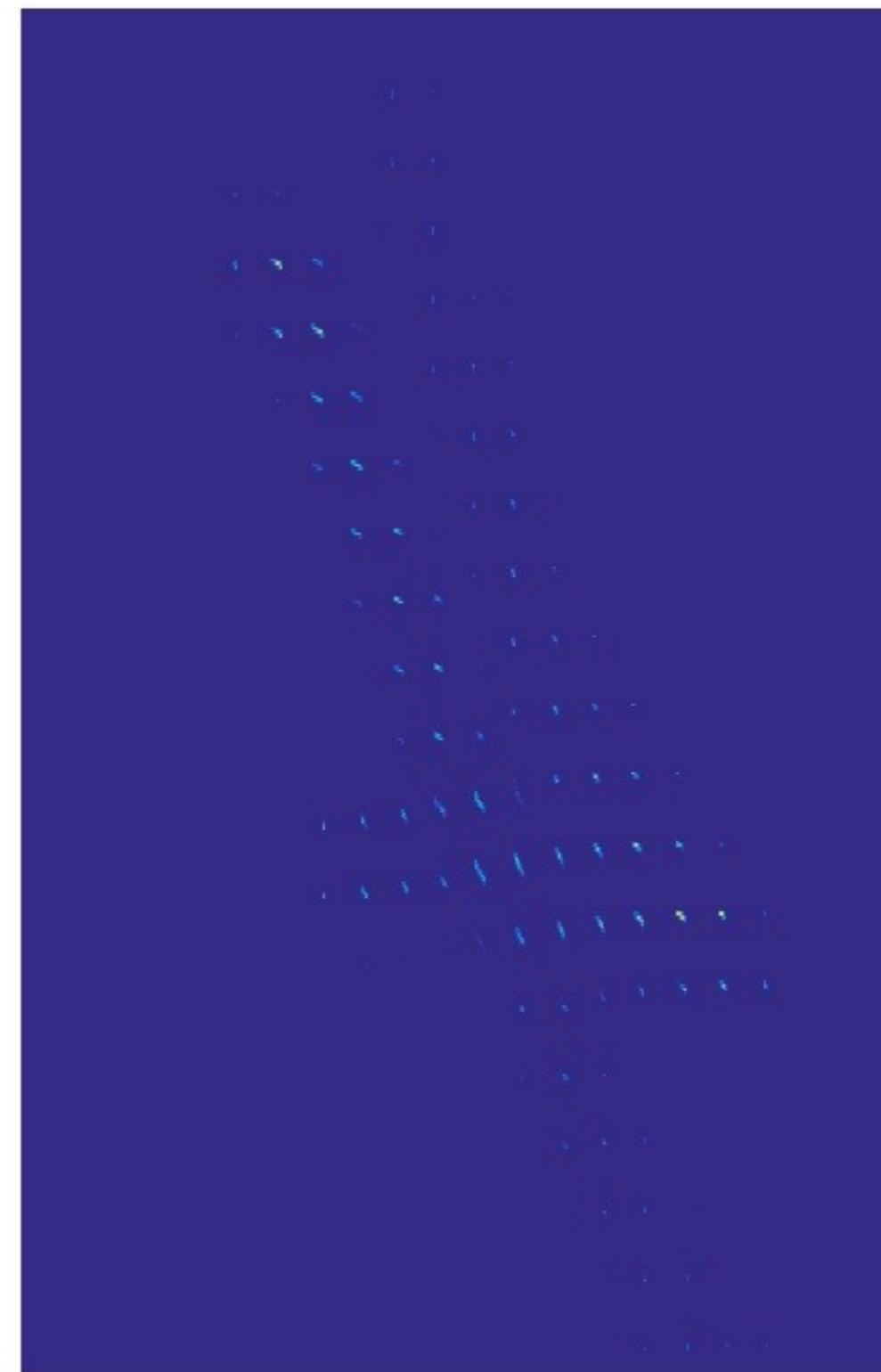


$\log(v_5)$



$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$



Very Fast EMD Approx. Solver

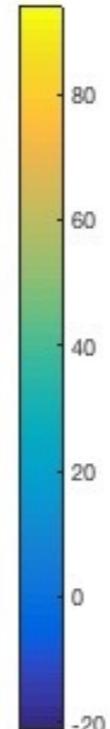
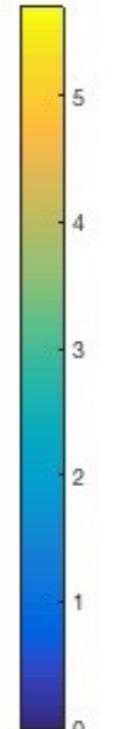
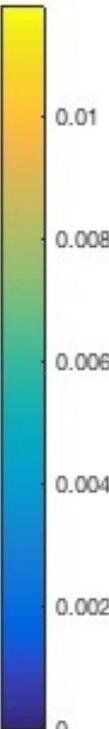
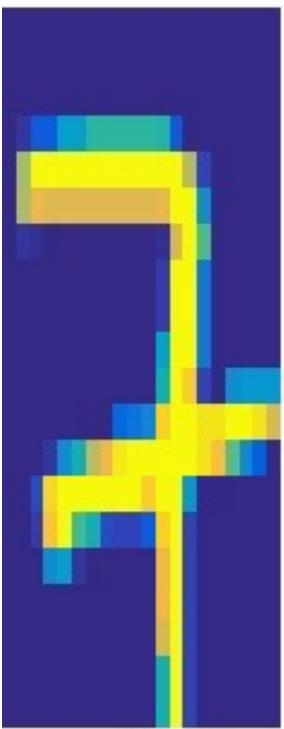
a

$$u_5 \leftarrow a/Kv_5$$

$$\log(u_5)$$

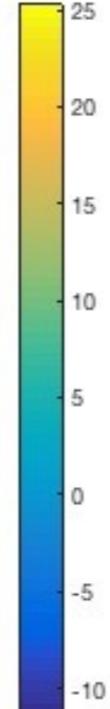
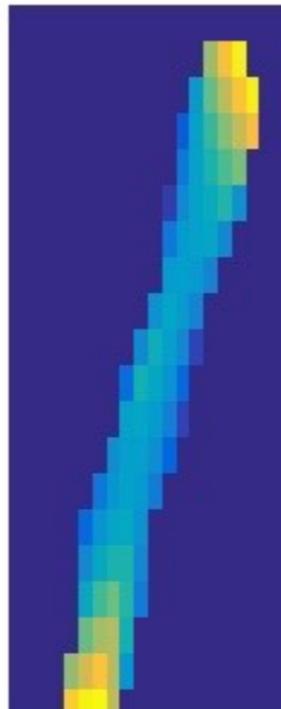
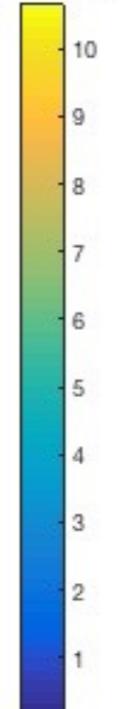
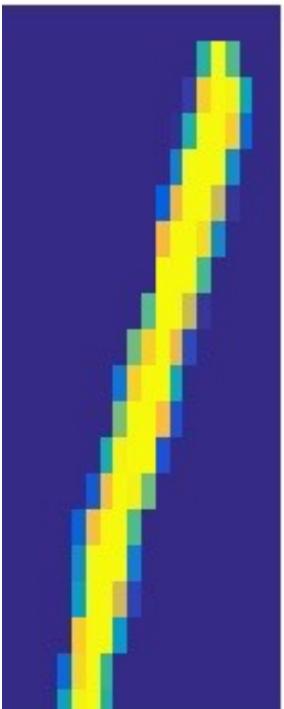
$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$



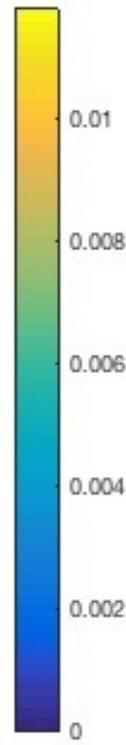
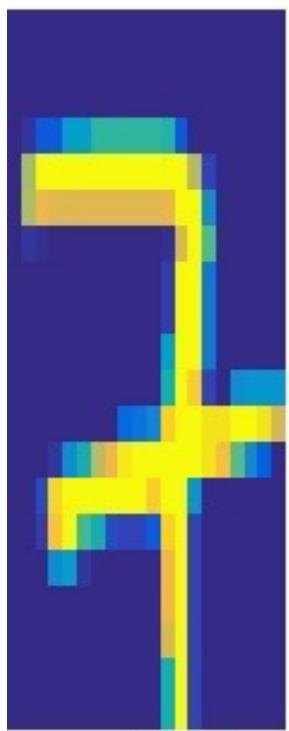
$$Kv_5$$

$$\log(v_5)$$

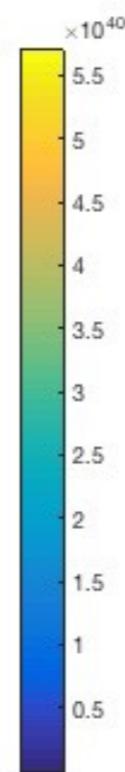


Very Fast EMD Approx. Solver

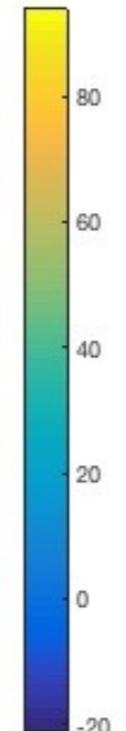
a



*Ku*₅



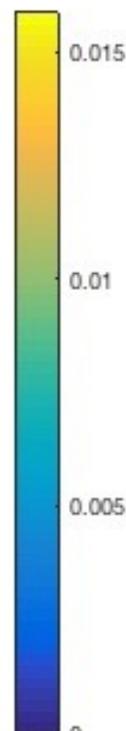
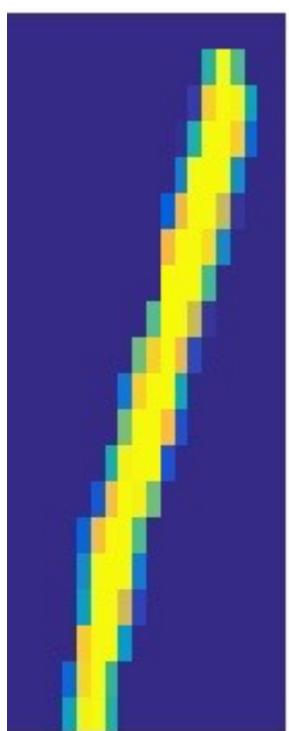
$\log(u_5)$



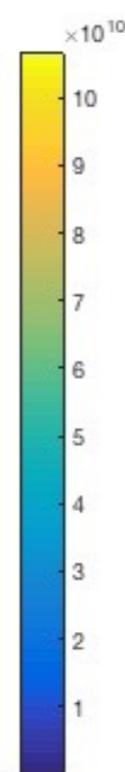
$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

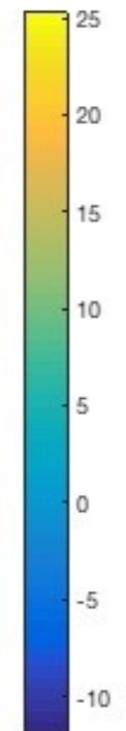
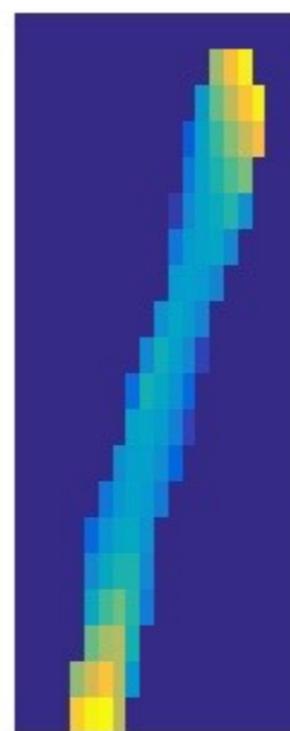
b



*Kv*₅

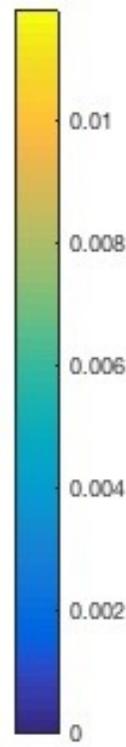
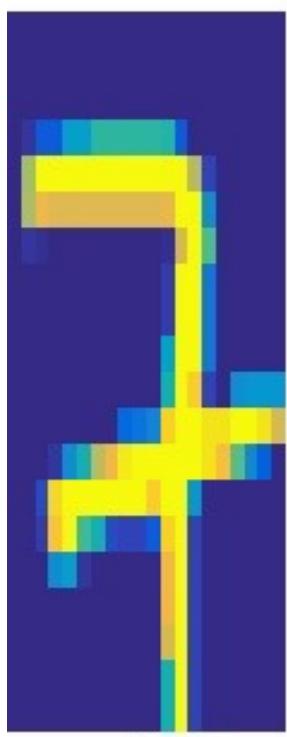


$\log(v_5)$

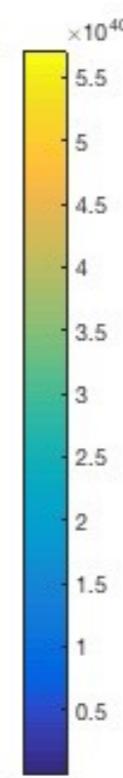


Very Fast EMD Approx. Solver

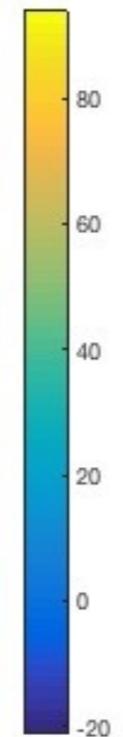
a



Ku_5



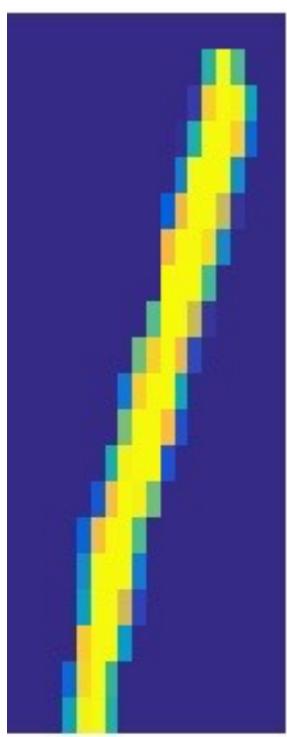
$\log(u_5)$



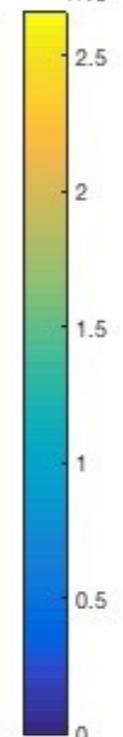
$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

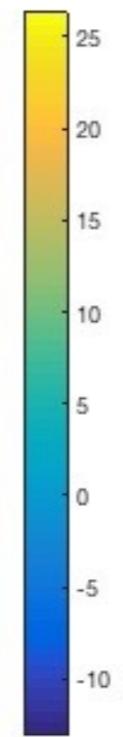
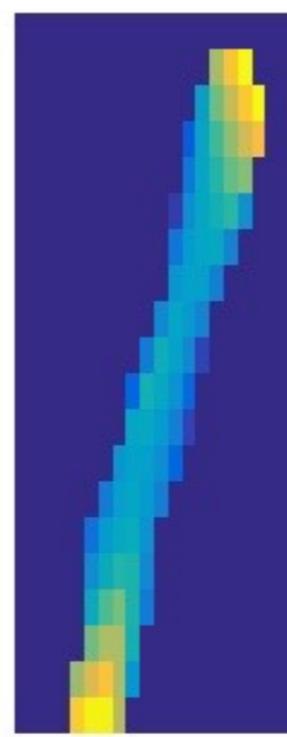
b



$v_6 \leftarrow b/Ku_6$

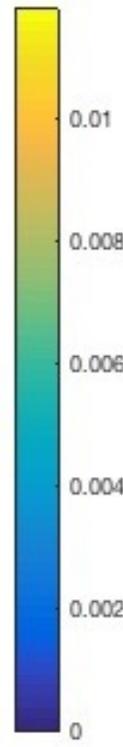
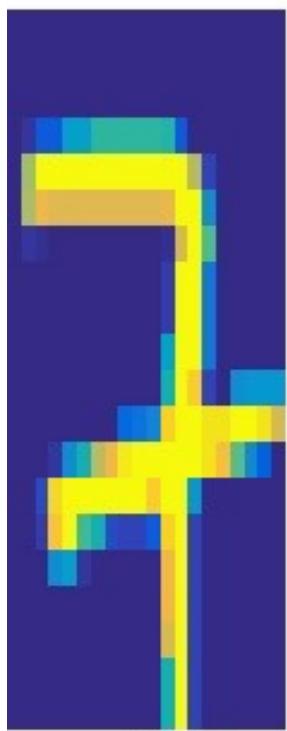


$\log(v_6)$

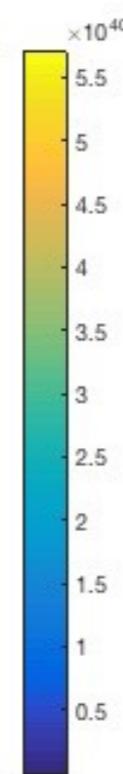


Very Fast EMD Approx. Solver

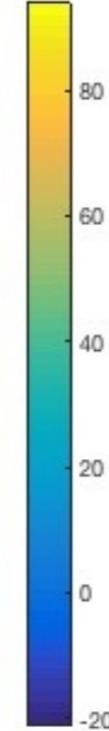
a



Ku_5



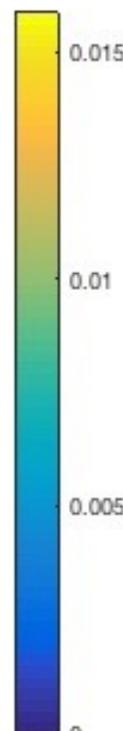
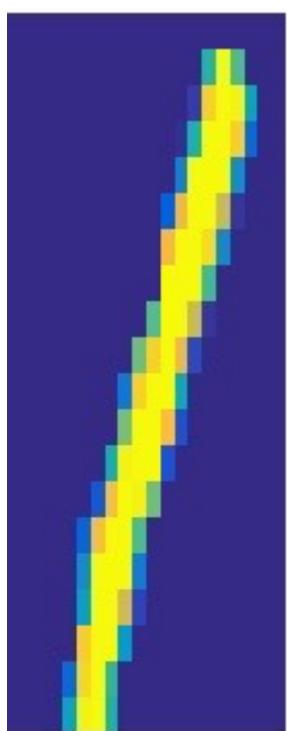
$\log(u_5)$



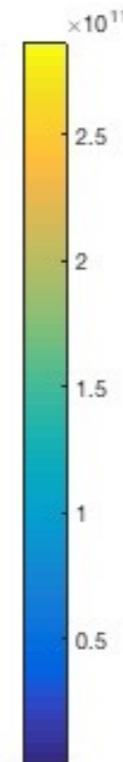
$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

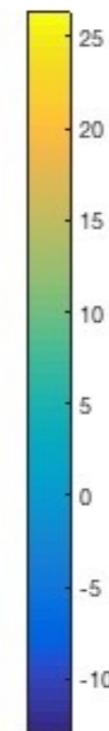
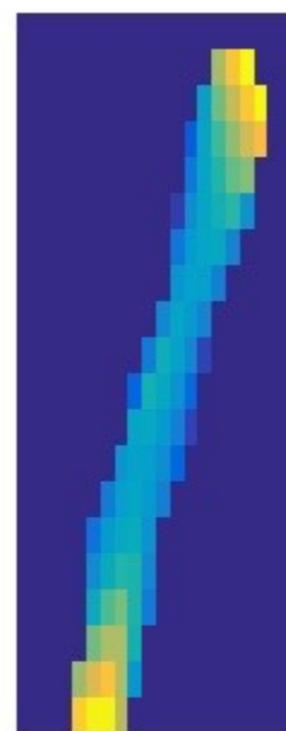
b



Kv_6



$\log(v_6)$



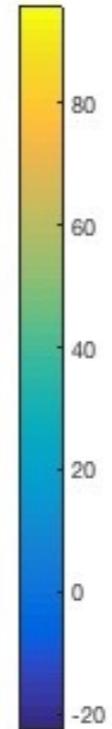
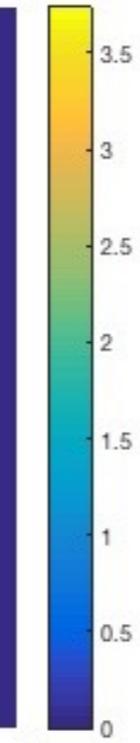
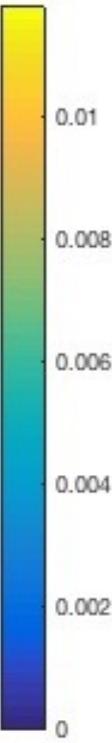
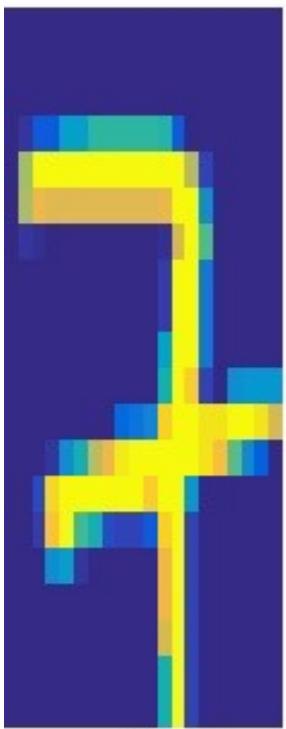
Very Fast EMD Approx. Solver

a

$$u_6 \leftarrow a/Kv_6$$

$$\log(u_6)$$

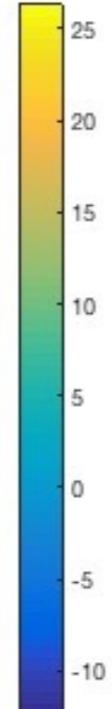
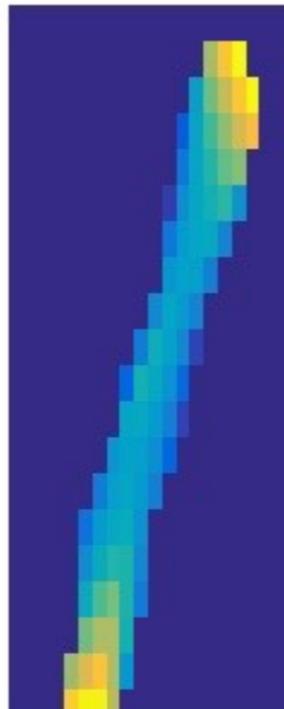
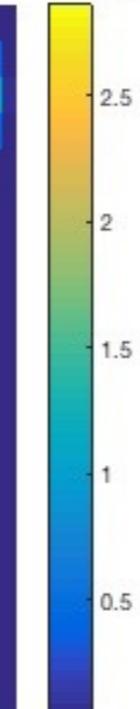
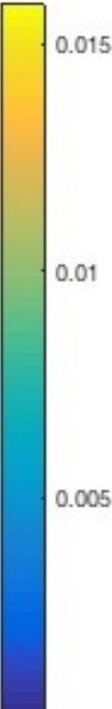
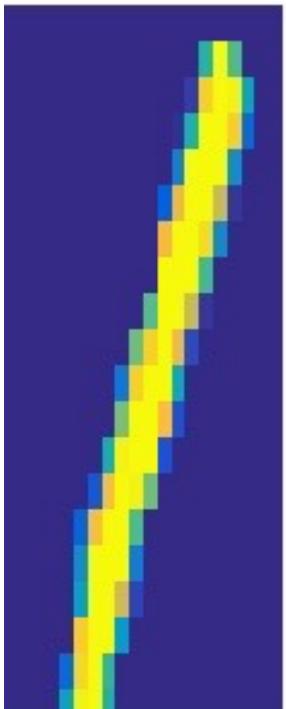
$$P_5 = D(u_5)KD(v_5)$$
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$



b

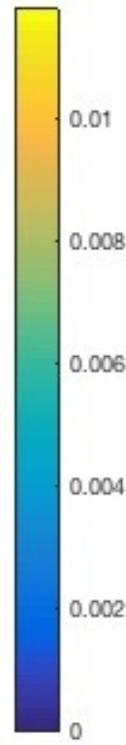
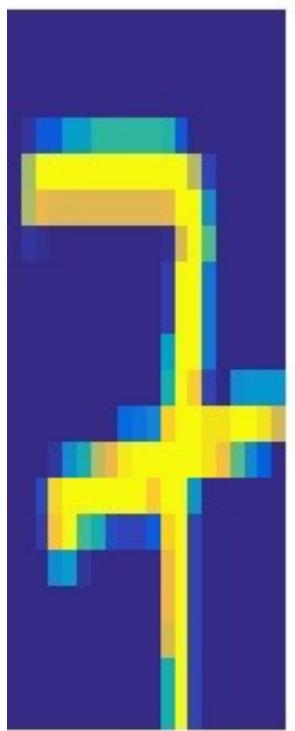
$$Kv_6$$

$$\log(v_6)$$

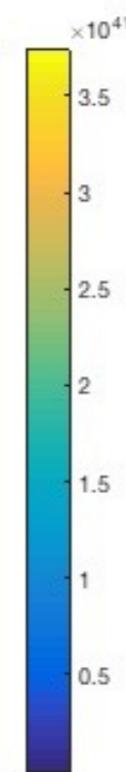
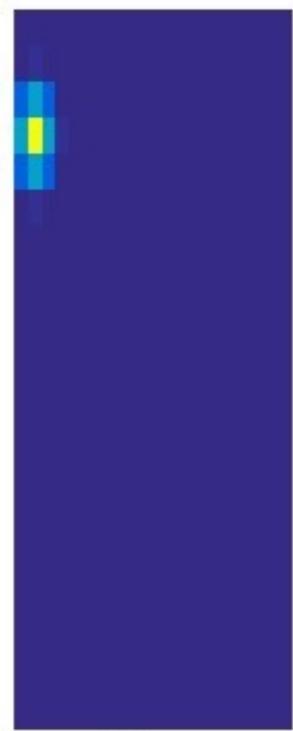


Very Fast EMD Approx. Solver

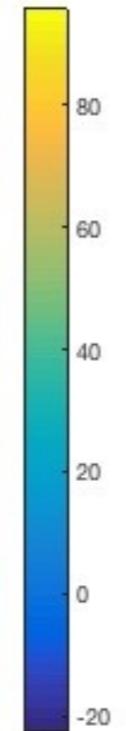
a



Ku_6



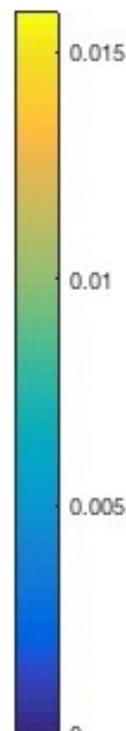
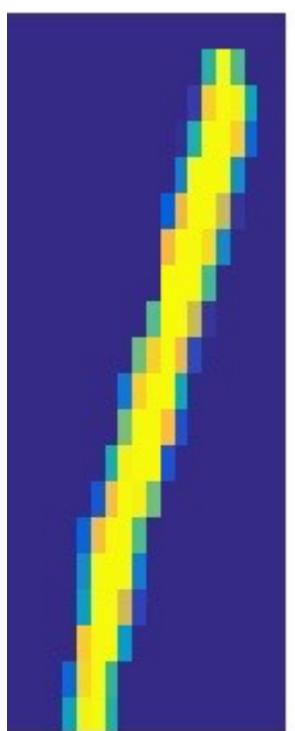
$\log(u_6)$



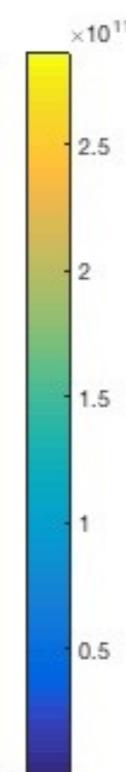
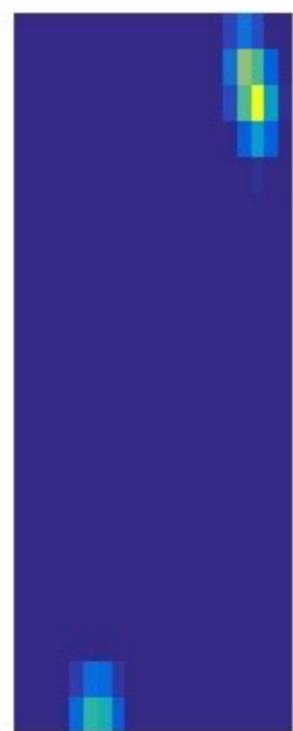
$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

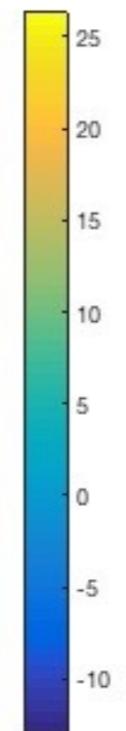
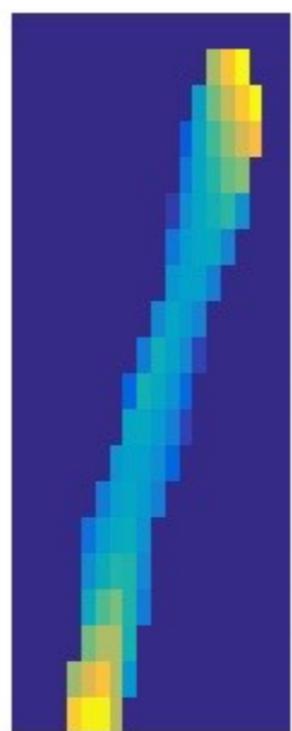
b



Kv_6

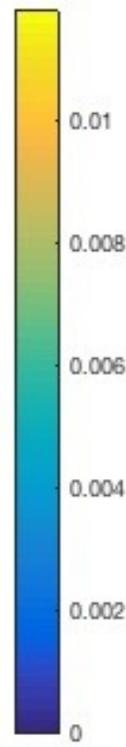
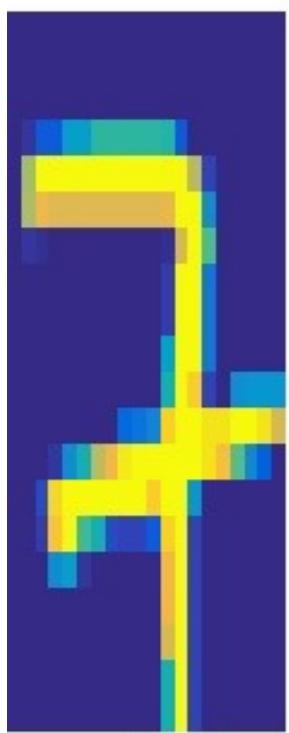


$\log(v_6)$

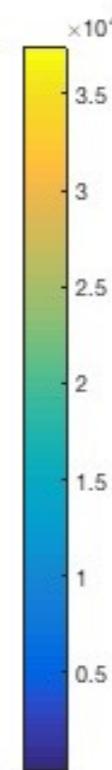


Very Fast EMD Approx. Solver

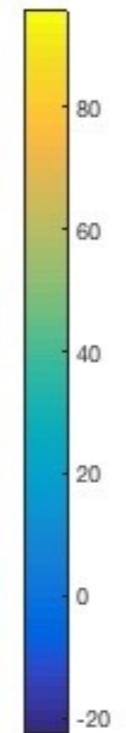
a



Ku_6



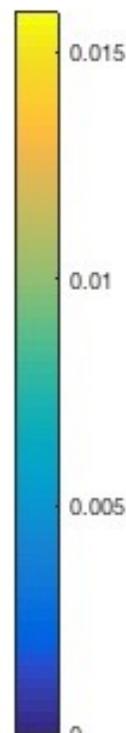
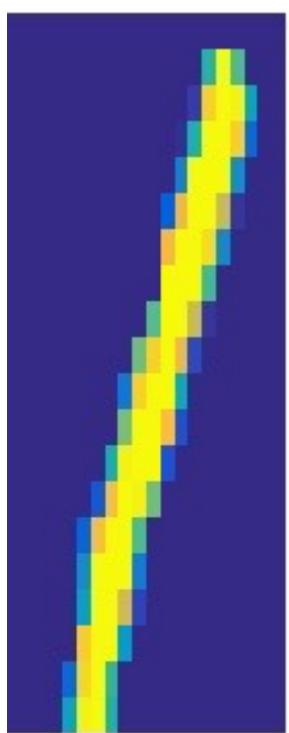
$\log(u_6)$



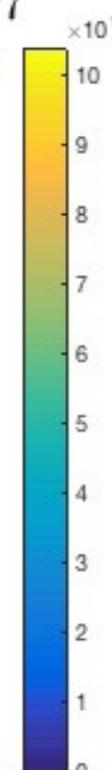
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

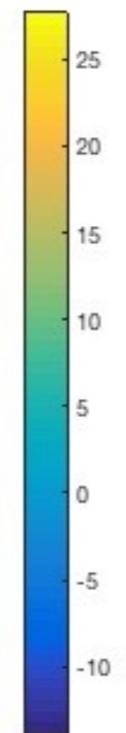
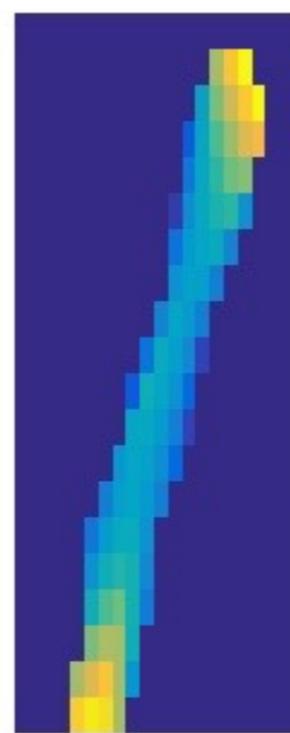
b



$v_7 \leftarrow b/Ku_7$

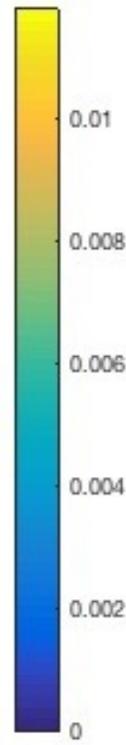
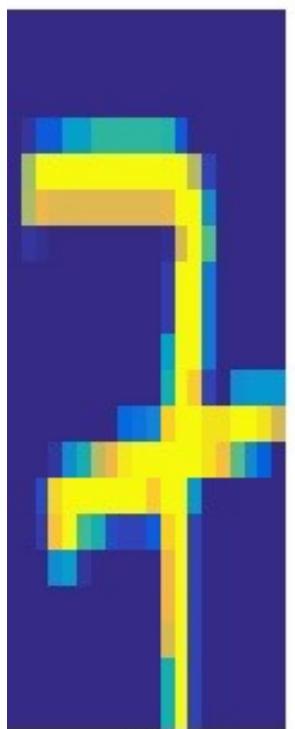


$\log(v_7)$

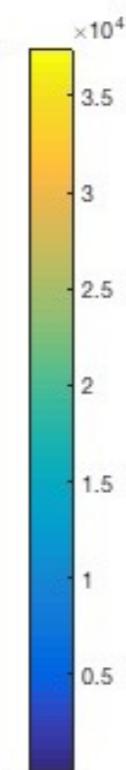


Very Fast EMD Approx. Solver

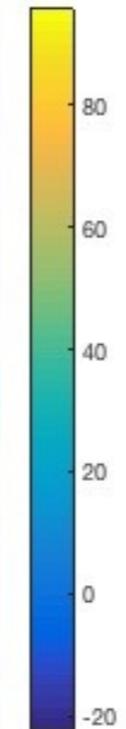
a



Ku_6



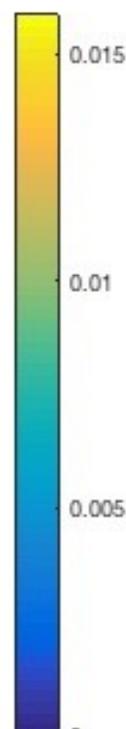
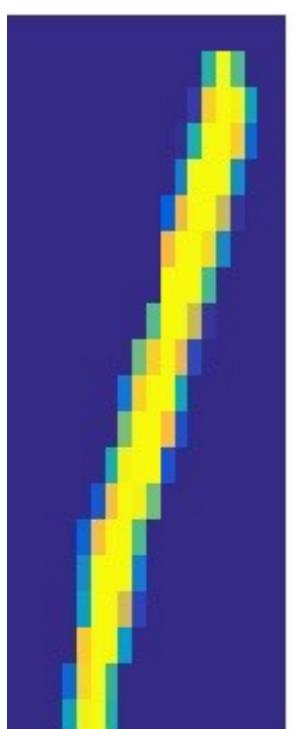
$\log(u_6)$



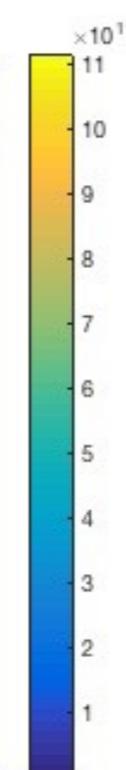
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

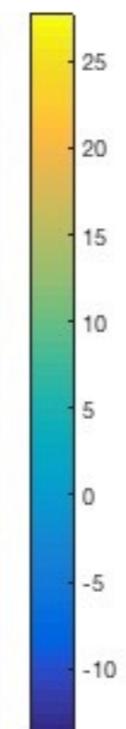
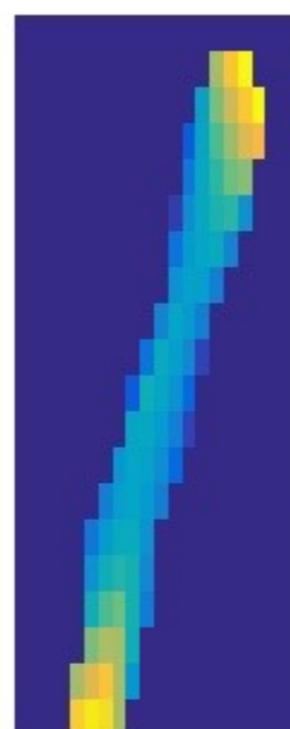
b



Kv_7

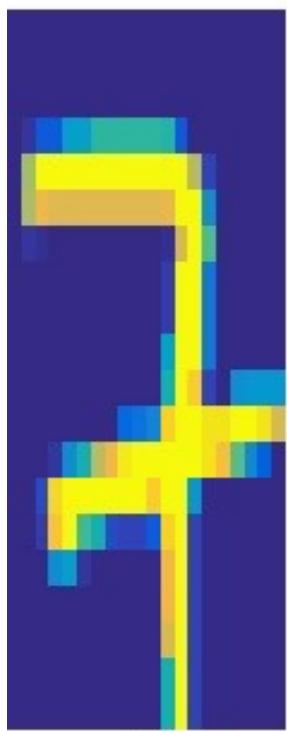


$\log(v_7)$

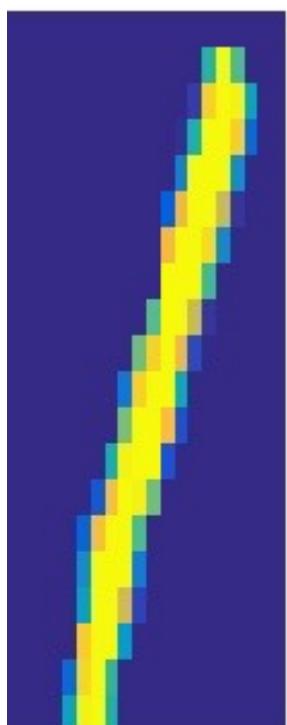


Very Fast EMD Approx. Solver

a



b



$$u_7 \leftarrow a/Kv_7$$



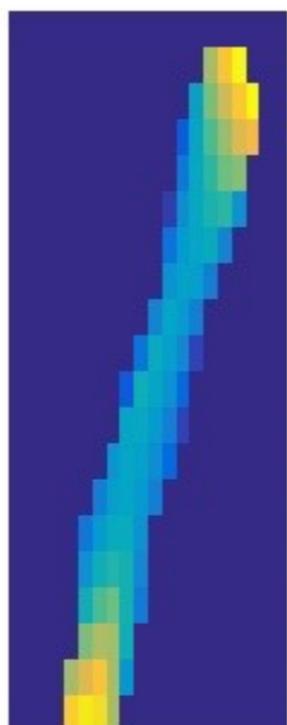
Kv_7



$$\log(u_7)$$

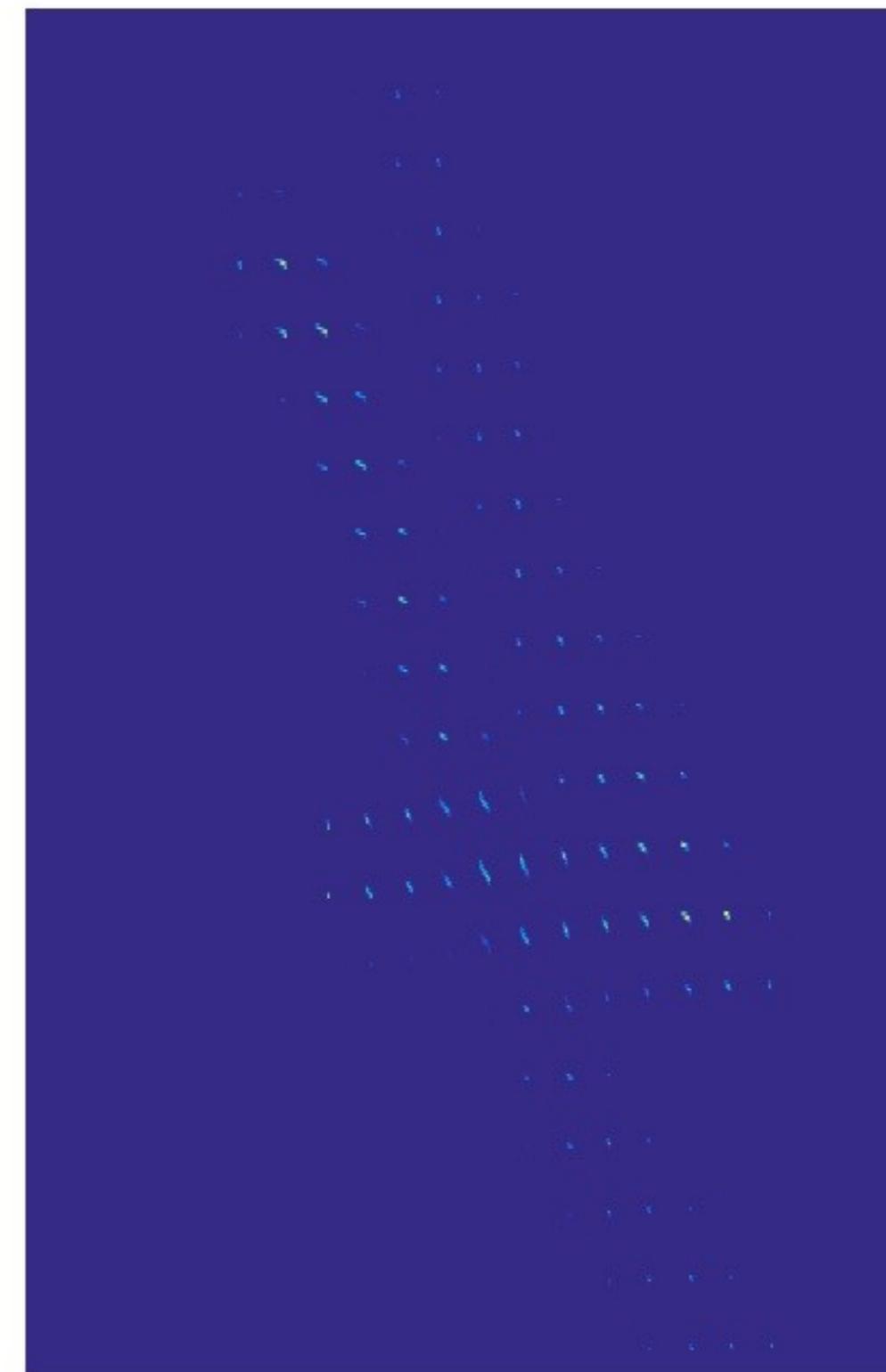


$\log(v_7)$



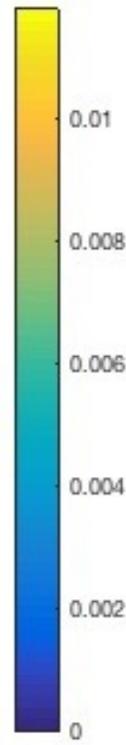
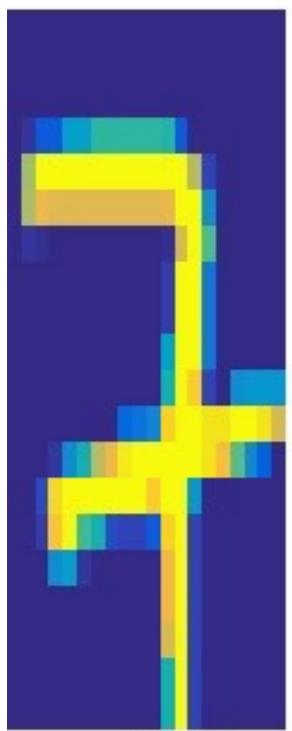
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

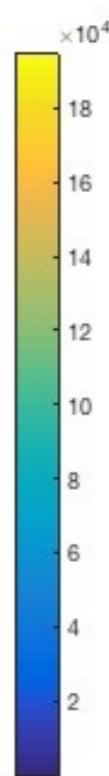
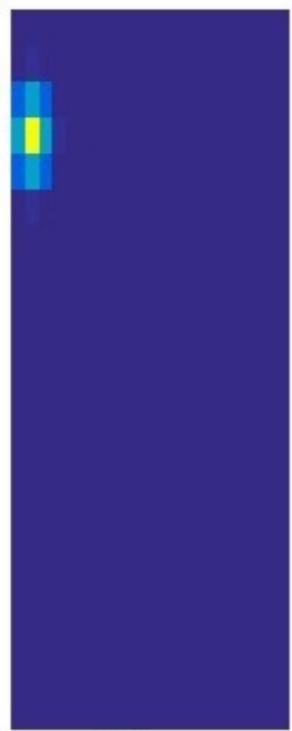


Very Fast EMD Approx. Solver

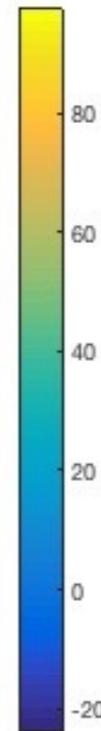
a



Ku_7



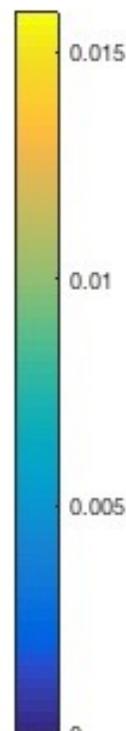
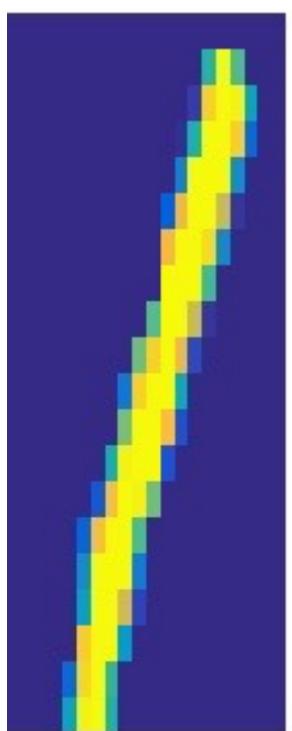
$\log(u_7)$



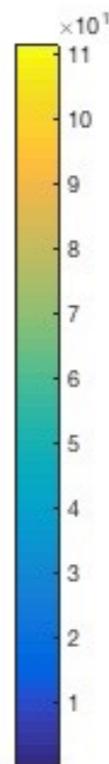
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

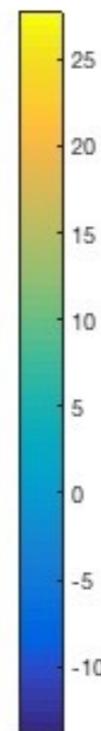
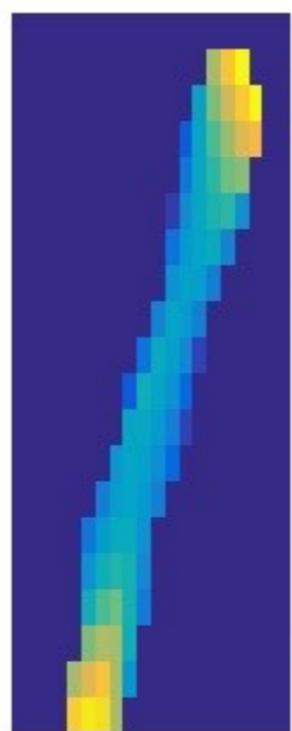
b



Kv_7

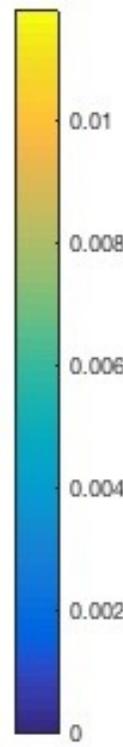
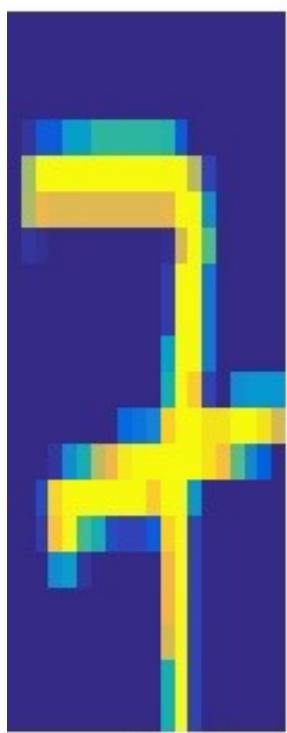


$\log(v_7)$

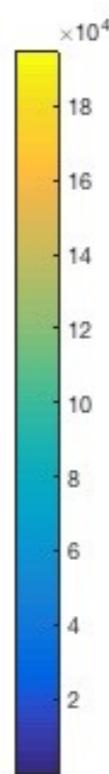


Very Fast EMD Approx. Solver

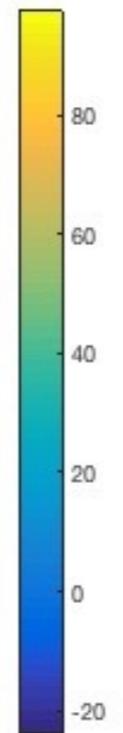
a



Ku₇



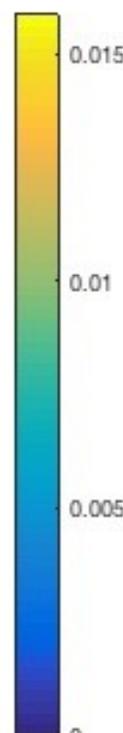
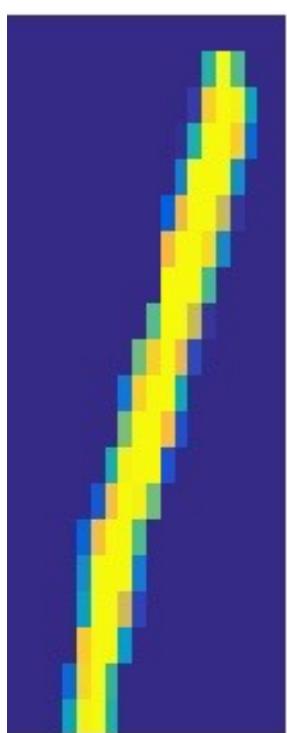
$\log(u_7)$



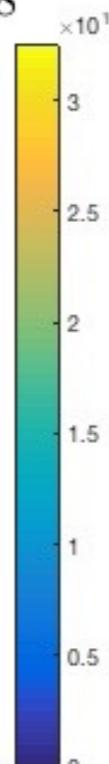
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

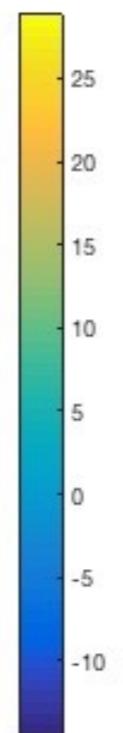
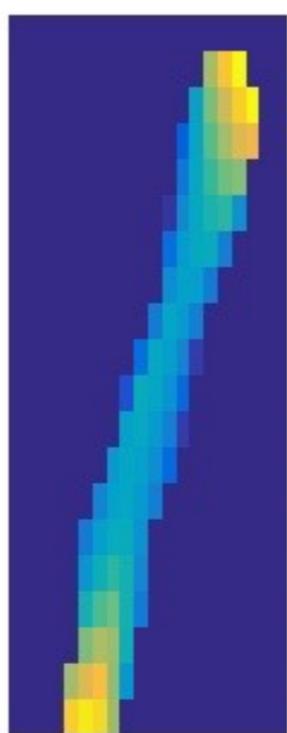
b



$v_8 \leftarrow b/Ku_8$

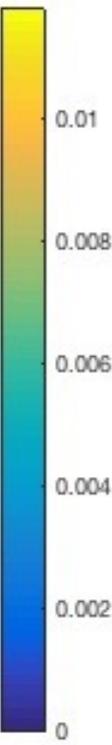
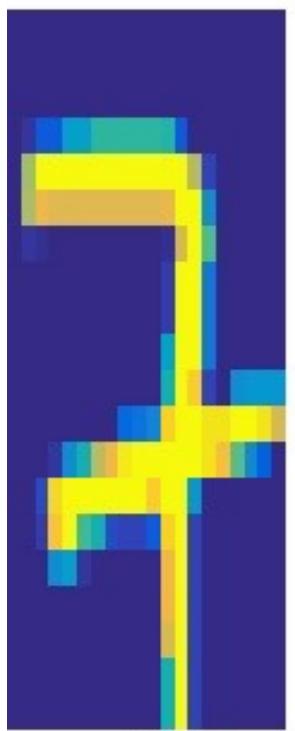


$\log(v_8)$

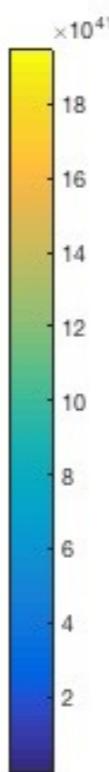
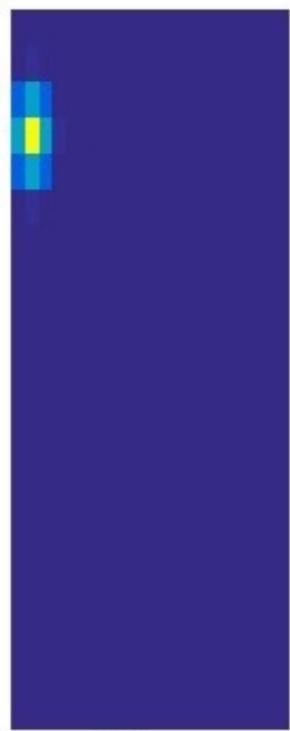


Very Fast EMD Approx. Solver

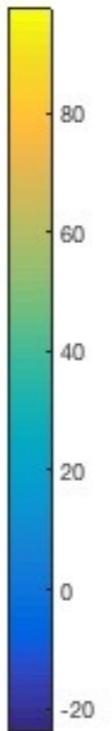
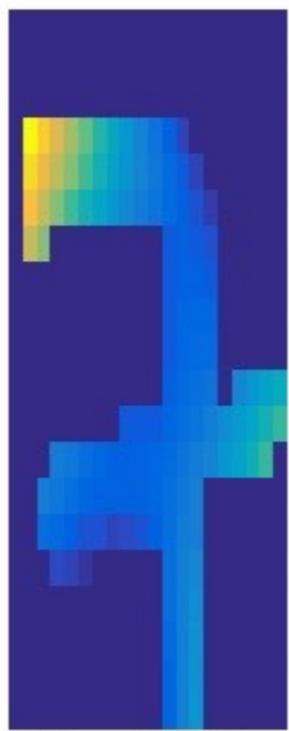
a



*Ku*₇



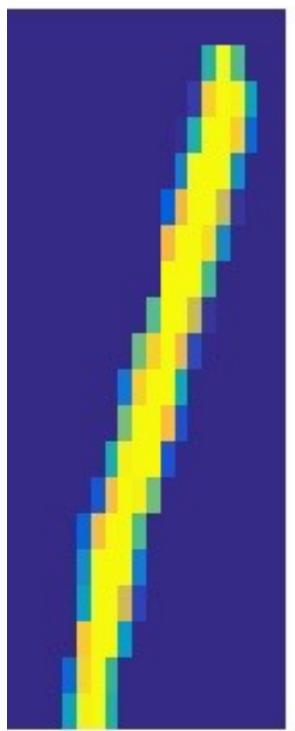
*log(u*₇)



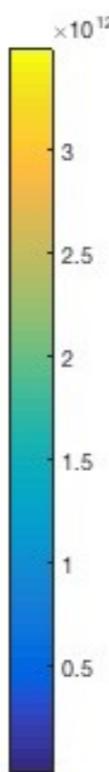
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

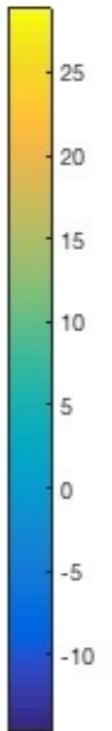
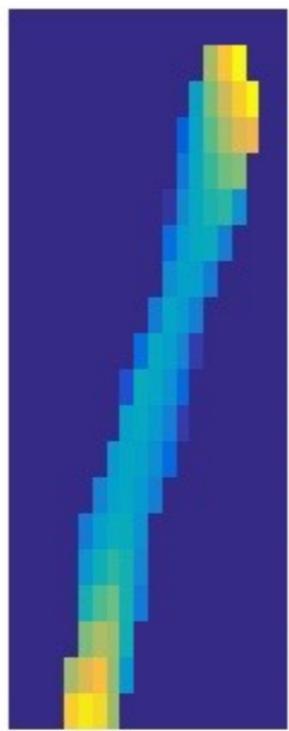
b



*Kv*₈



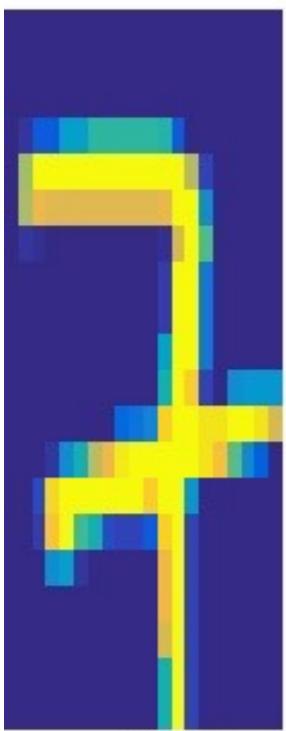
*log(v*₈)



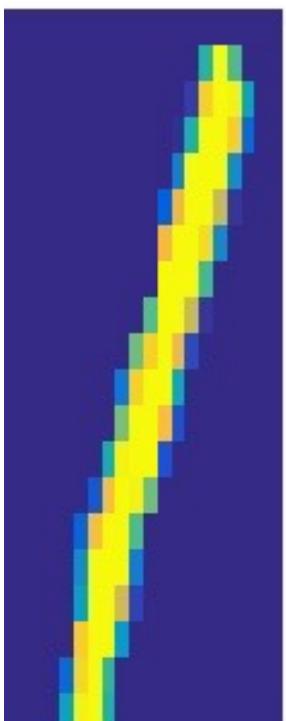
Very Fast EMD Approx. Solver

a

$$u_8 \leftarrow a/Kv_8$$



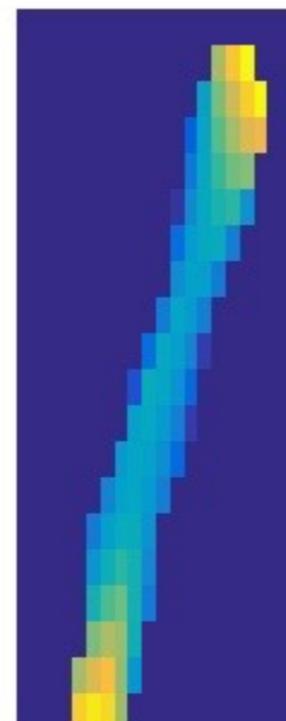
b



$$\log(u_8)$$

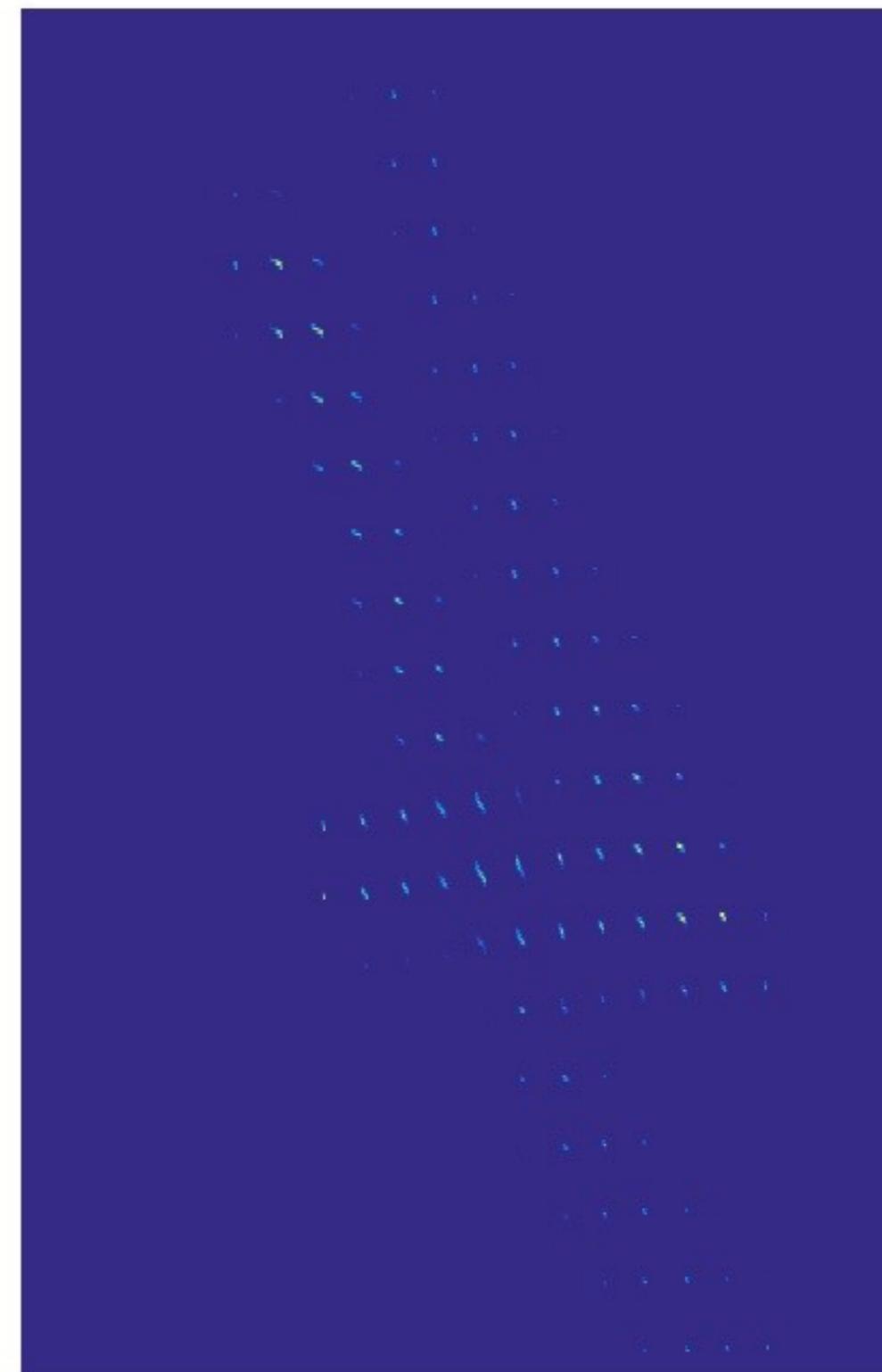


$$\log(v_8)$$



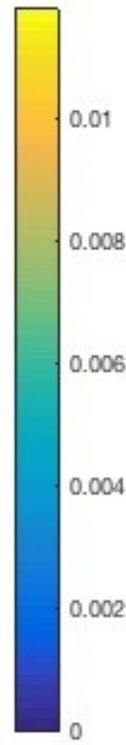
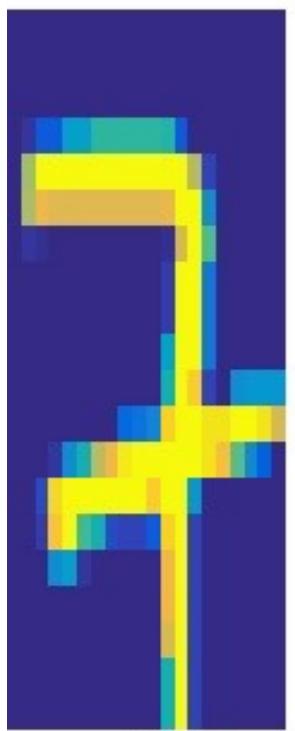
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

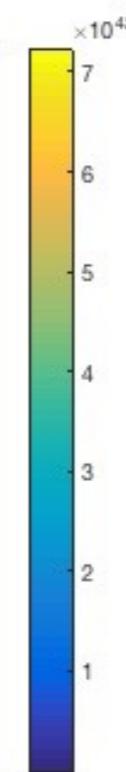


Very Fast EMD Approx. Solver

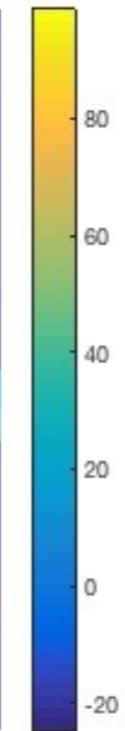
a



Ku_8



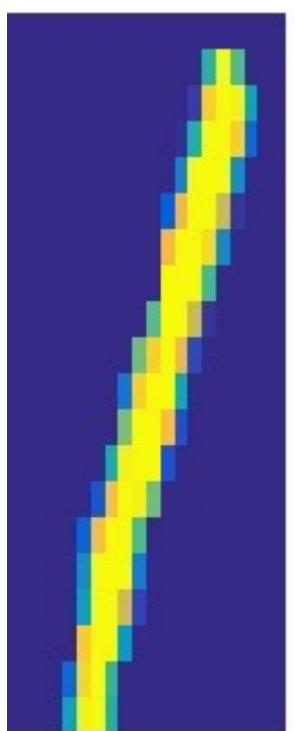
$\log(u_8)$



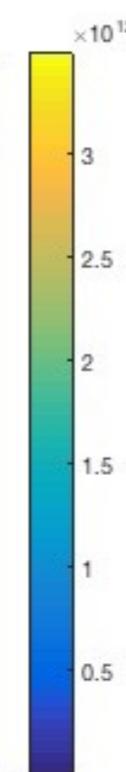
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

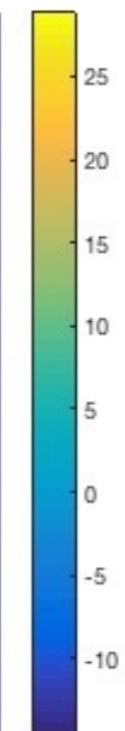
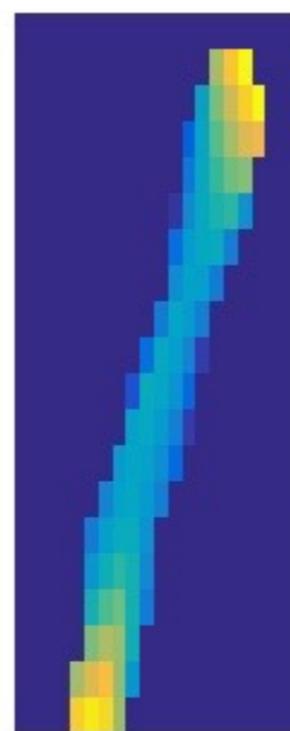
b



Kv_8

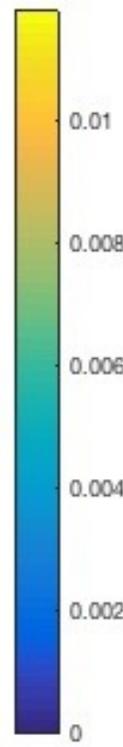
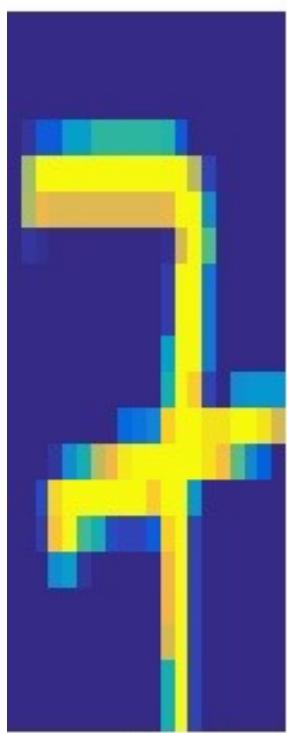


$\log(v_8)$

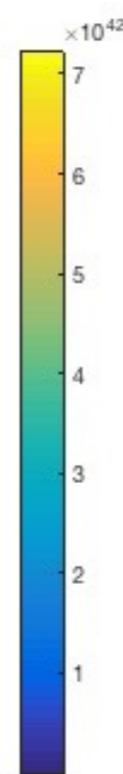


Very Fast EMD Approx. Solver

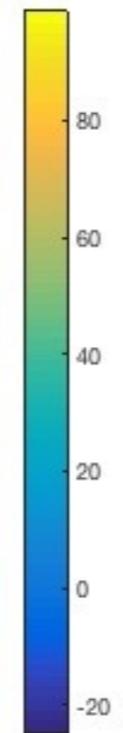
a



Ku₈



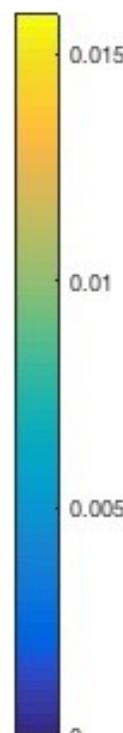
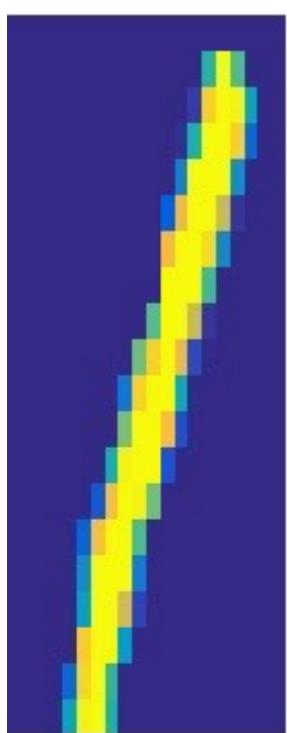
$\log(u_8)$



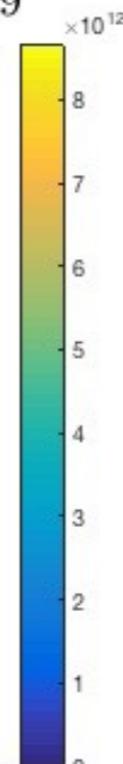
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

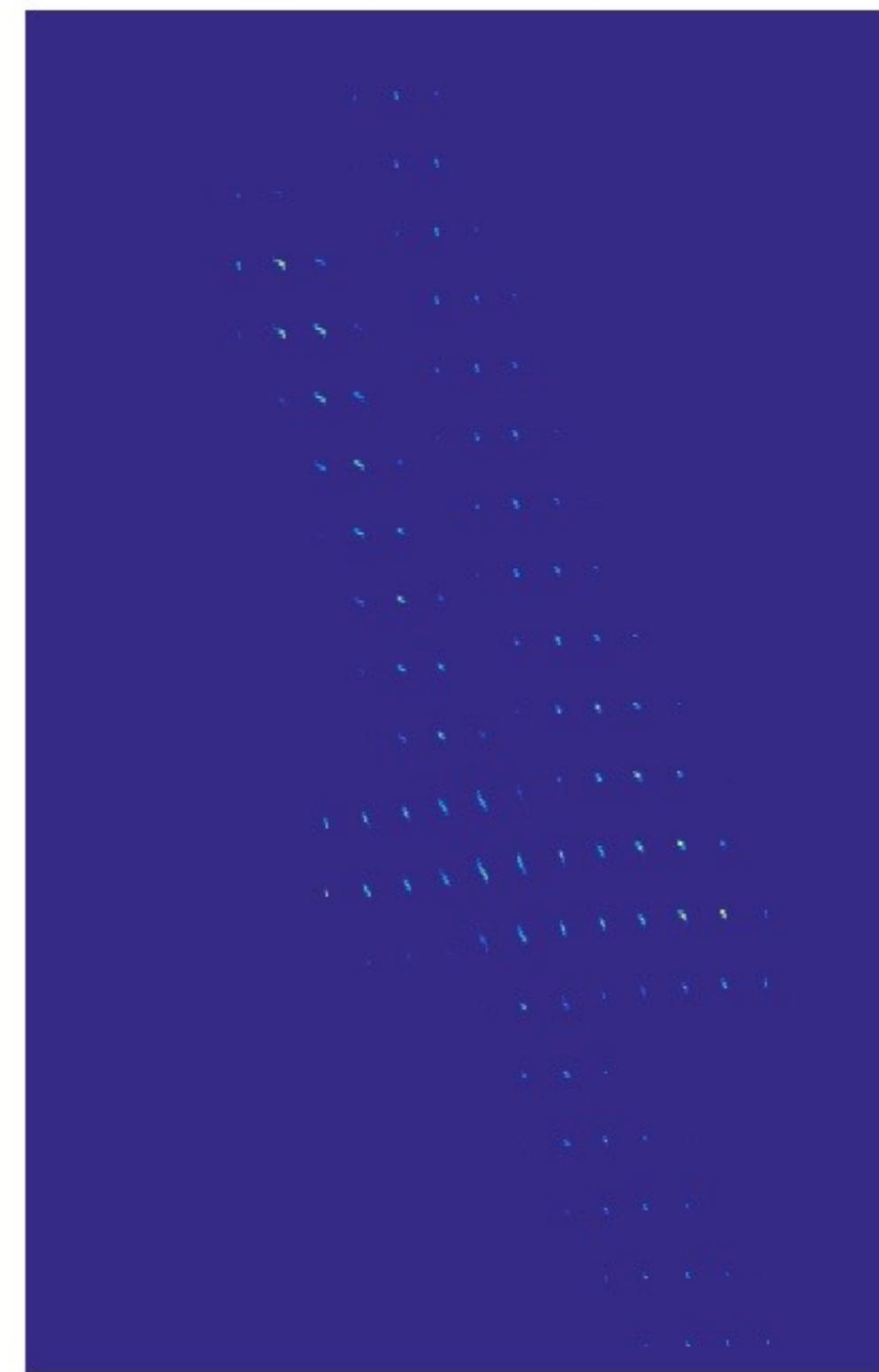
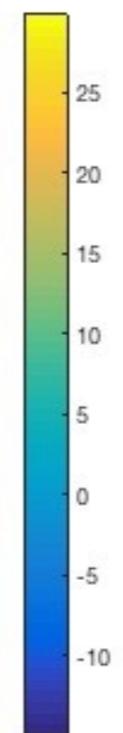
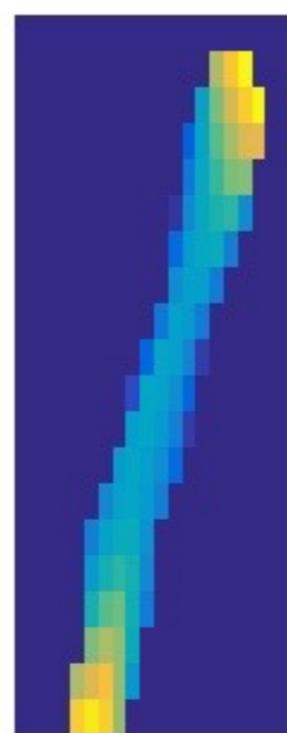
b



$v_9 \leftarrow b/Ku_9$

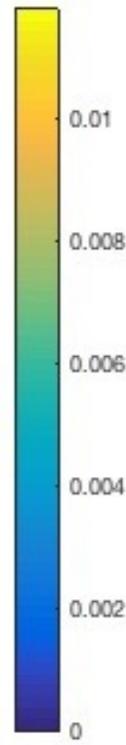
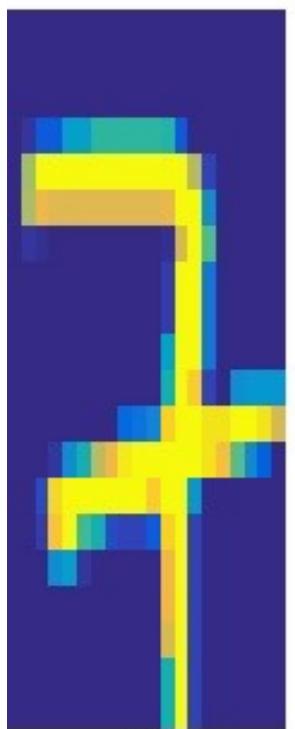


$\log(v_9)$

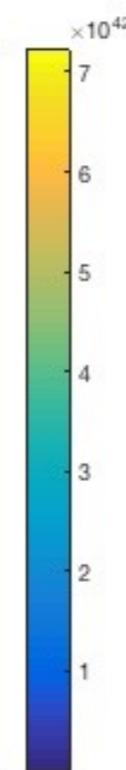
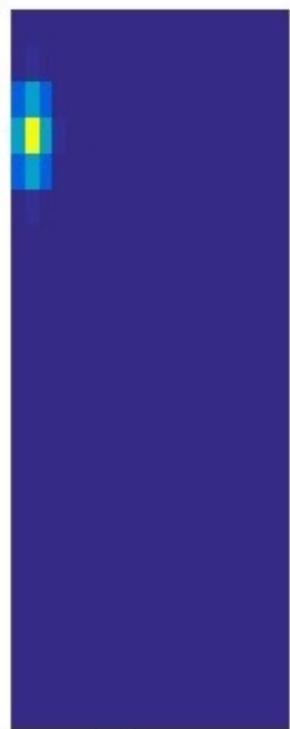


Very Fast EMD Approx. Solver

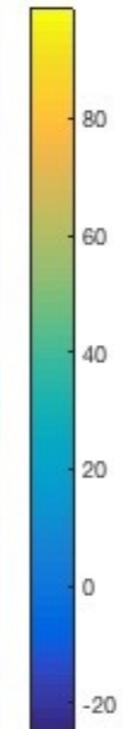
a



Ku_8



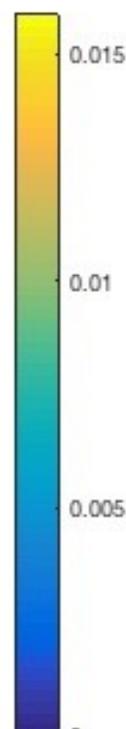
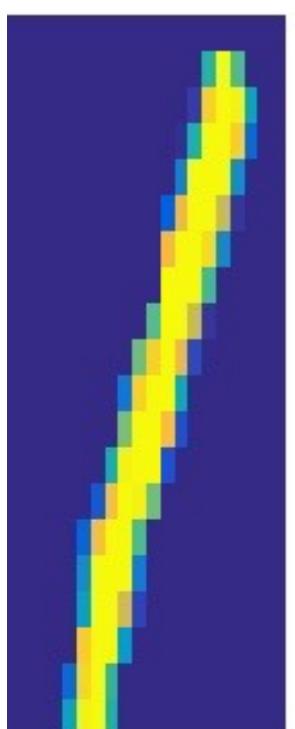
$\log(u_8)$



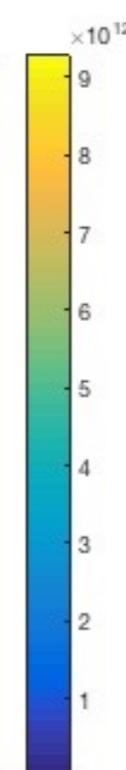
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

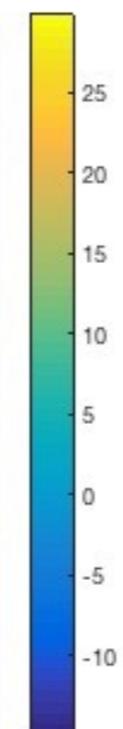
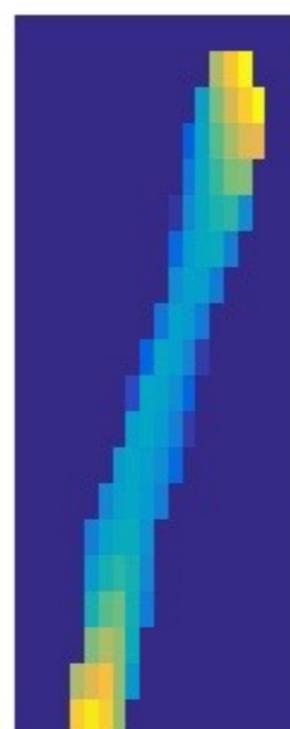
b



Kv_9

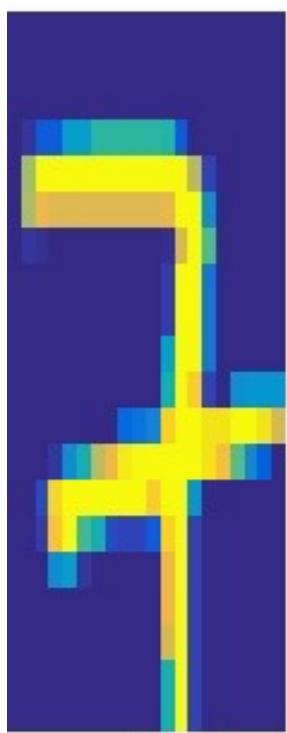


$\log(v_9)$



Very Fast EMD Approx. Solver

a



$$u_9 \leftarrow a/Kv_9$$

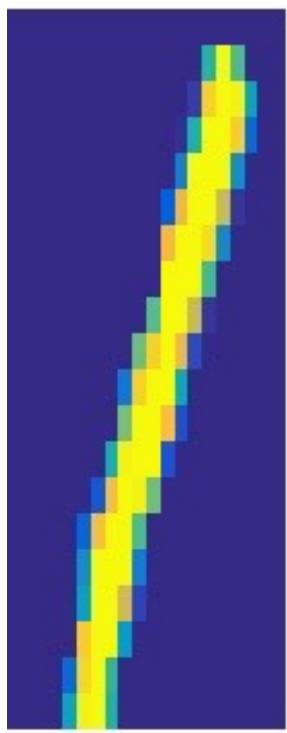


$$\log(u_9)$$



$$P_8 = D(u_8)KD(v_8)$$
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

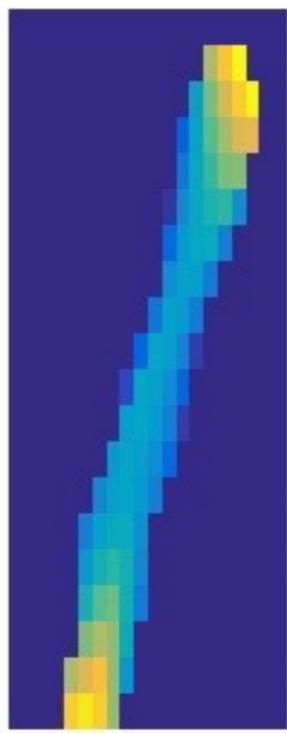
b



$$Kv_9$$

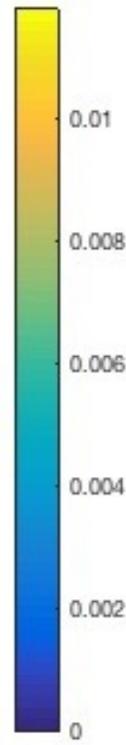
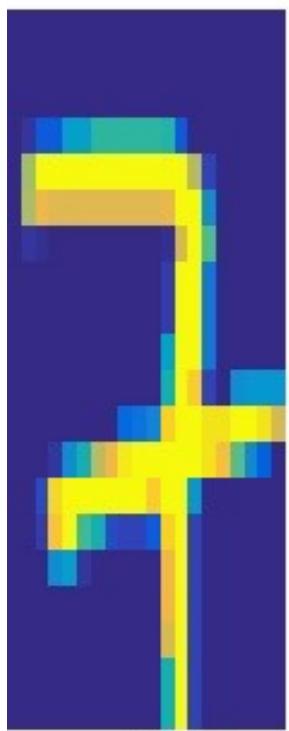


$$\log(v_9)$$

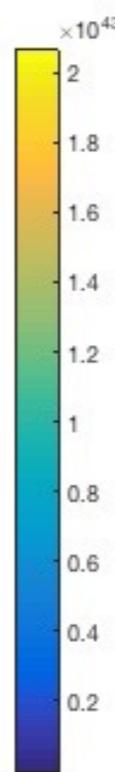


Very Fast EMD Approx. Solver

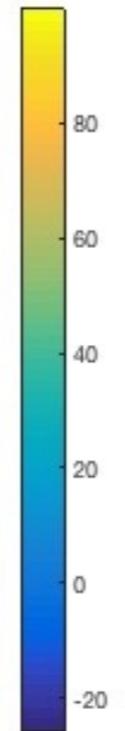
a



Ku_9



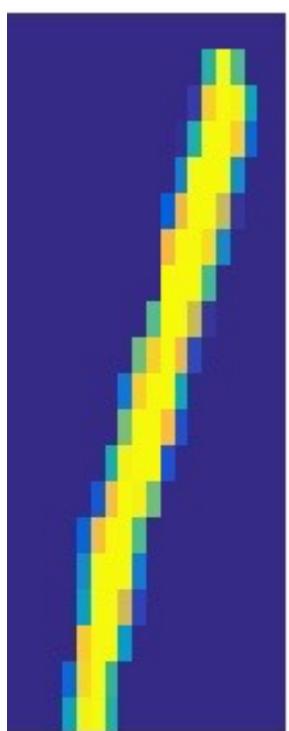
$\log(u_9)$



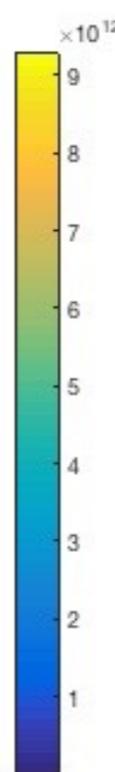
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

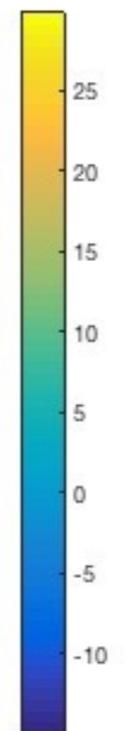
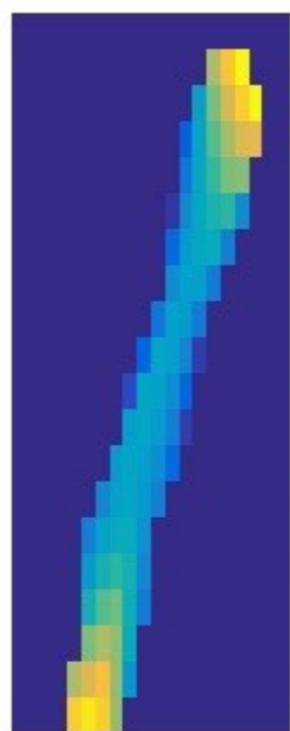
b



Kv_9

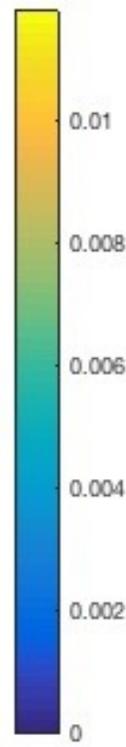
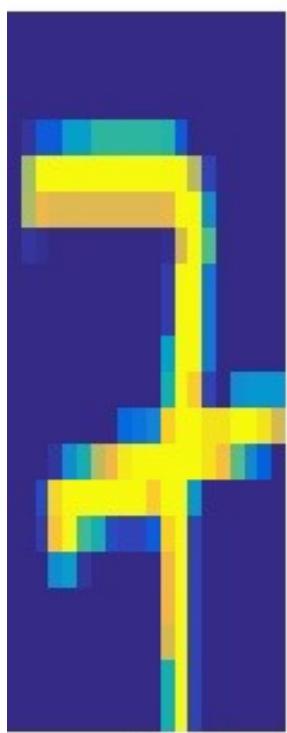


$\log(v_9)$

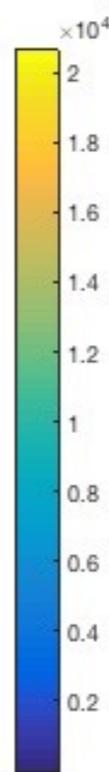


Very Fast EMD Approx. Solver

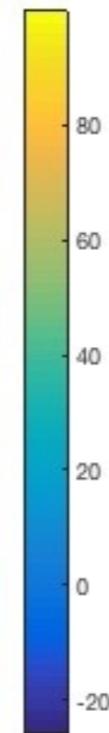
a



Ku₉



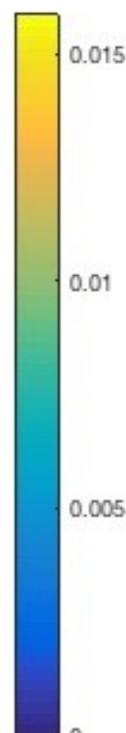
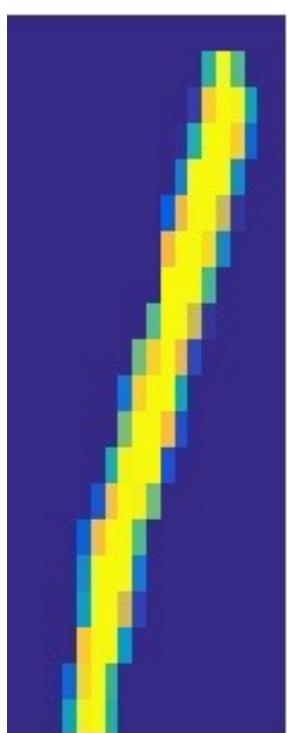
$\log(u_9)$



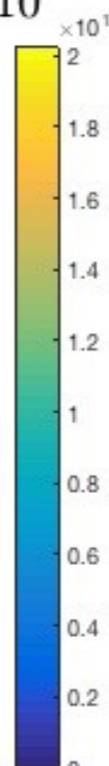
$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

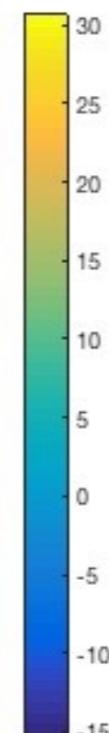
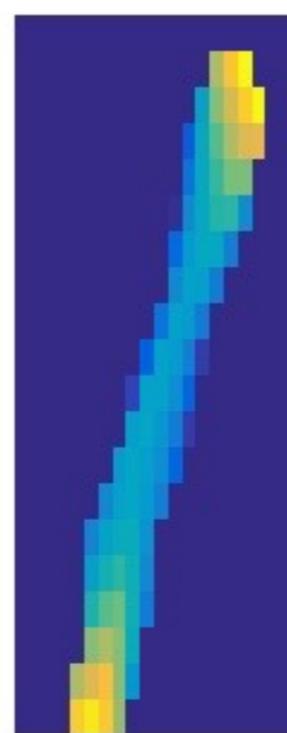
b



$v_{10} \leftarrow b/Ku_{10}$

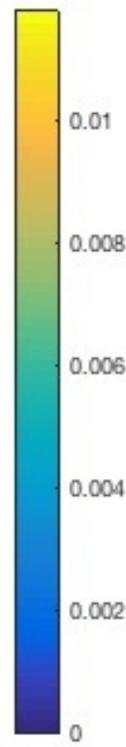
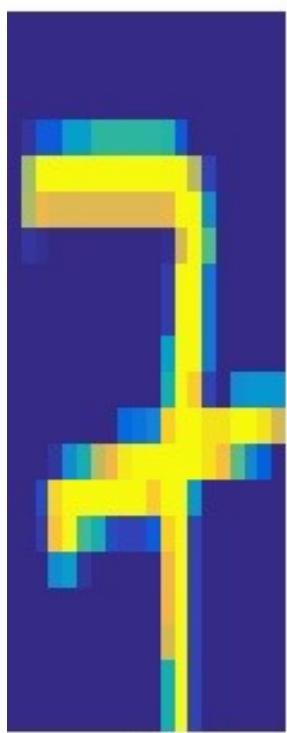


$\log(v_{10})$

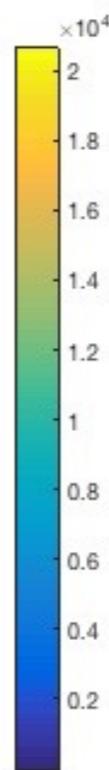


Very Fast EMD Approx. Solver

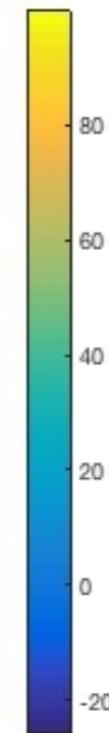
a



Ku_9



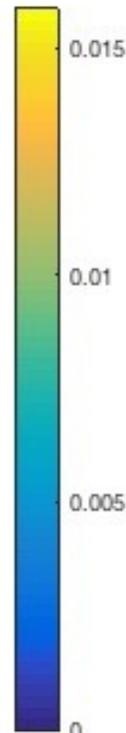
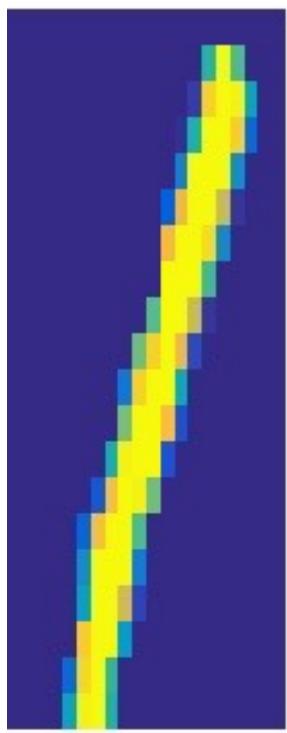
$\log(u_9)$



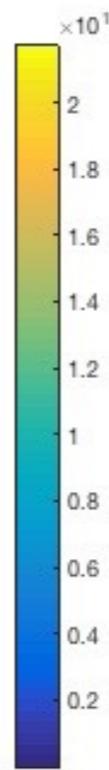
$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

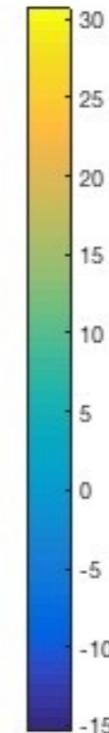
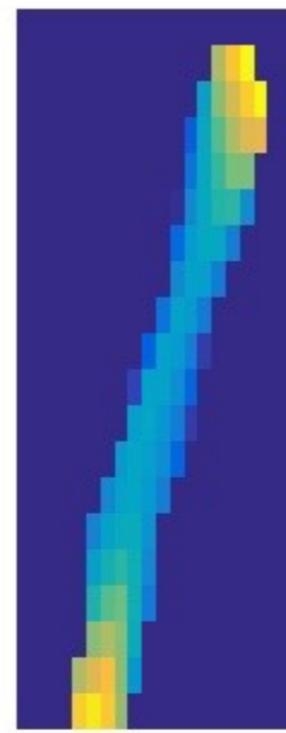
b



Kv_{10}



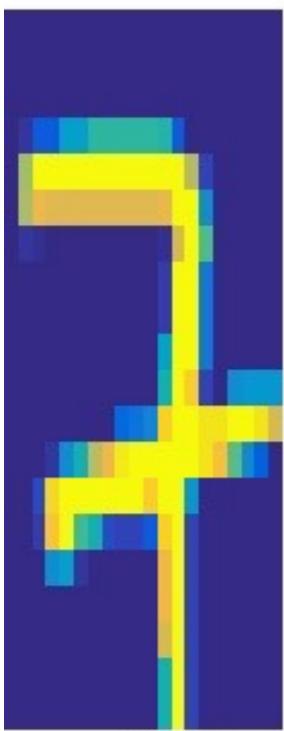
$\log(v_{10})$



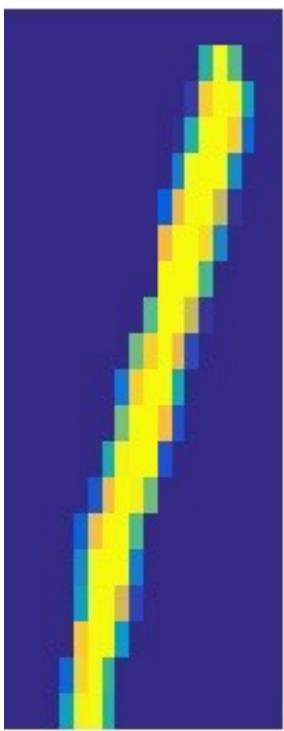
Very Fast EMD Approx. Solver

a

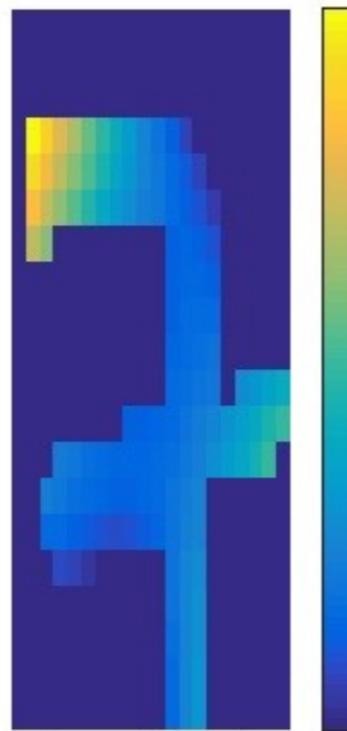
$$u_{10} \leftarrow a/Kv_{10}$$



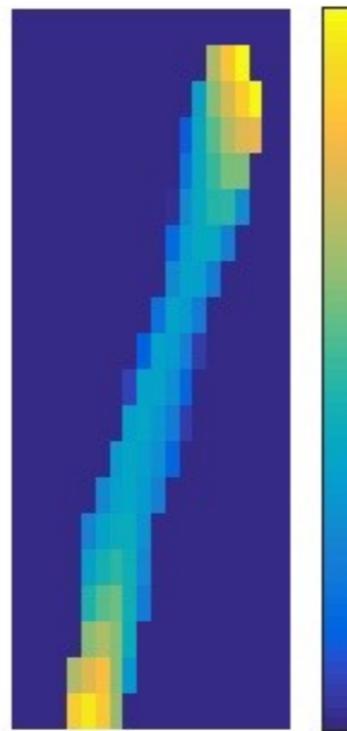
b



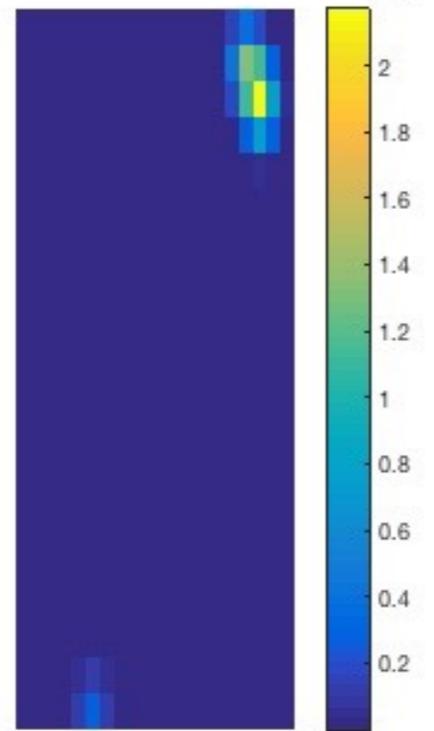
$$\log(u_{10})$$



$$\log(v_{10})$$

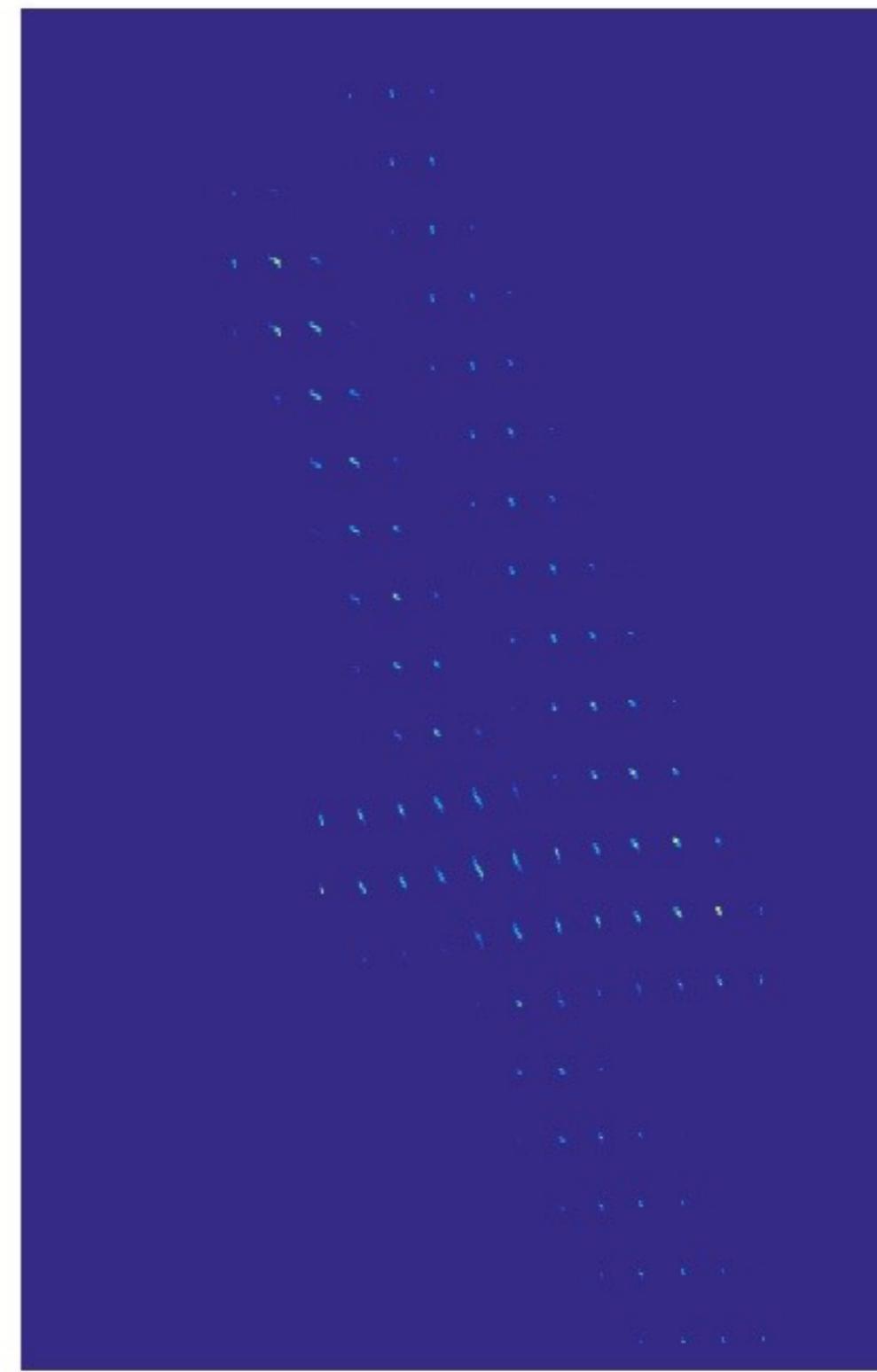


$$Kv_{10}$$



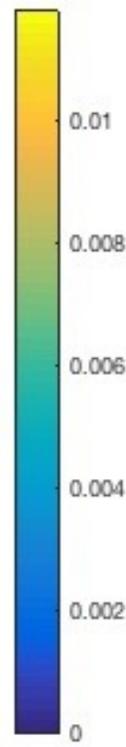
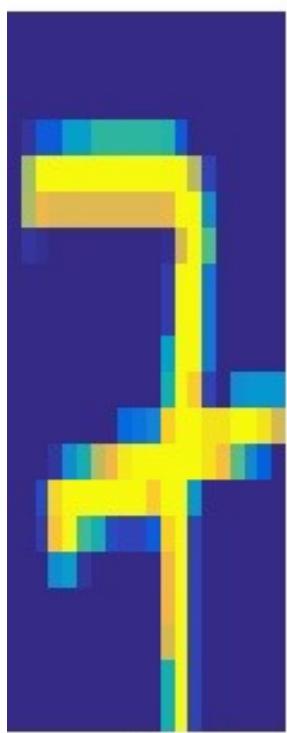
$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$



Very Fast EMD Approx. Solver

a



*Ku*₁₀



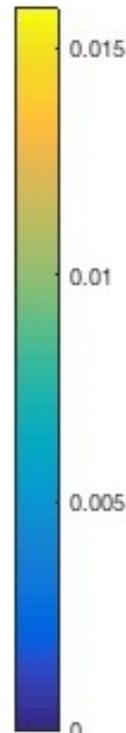
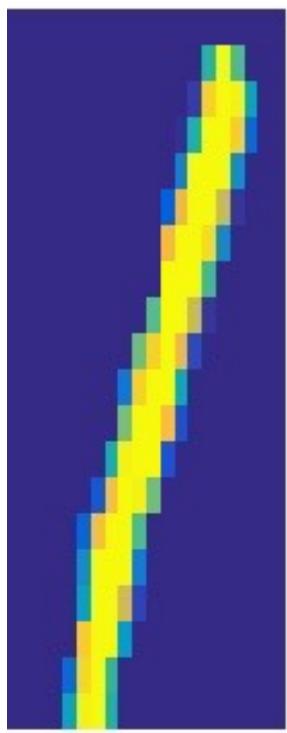
$\log(u_{10})$



$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

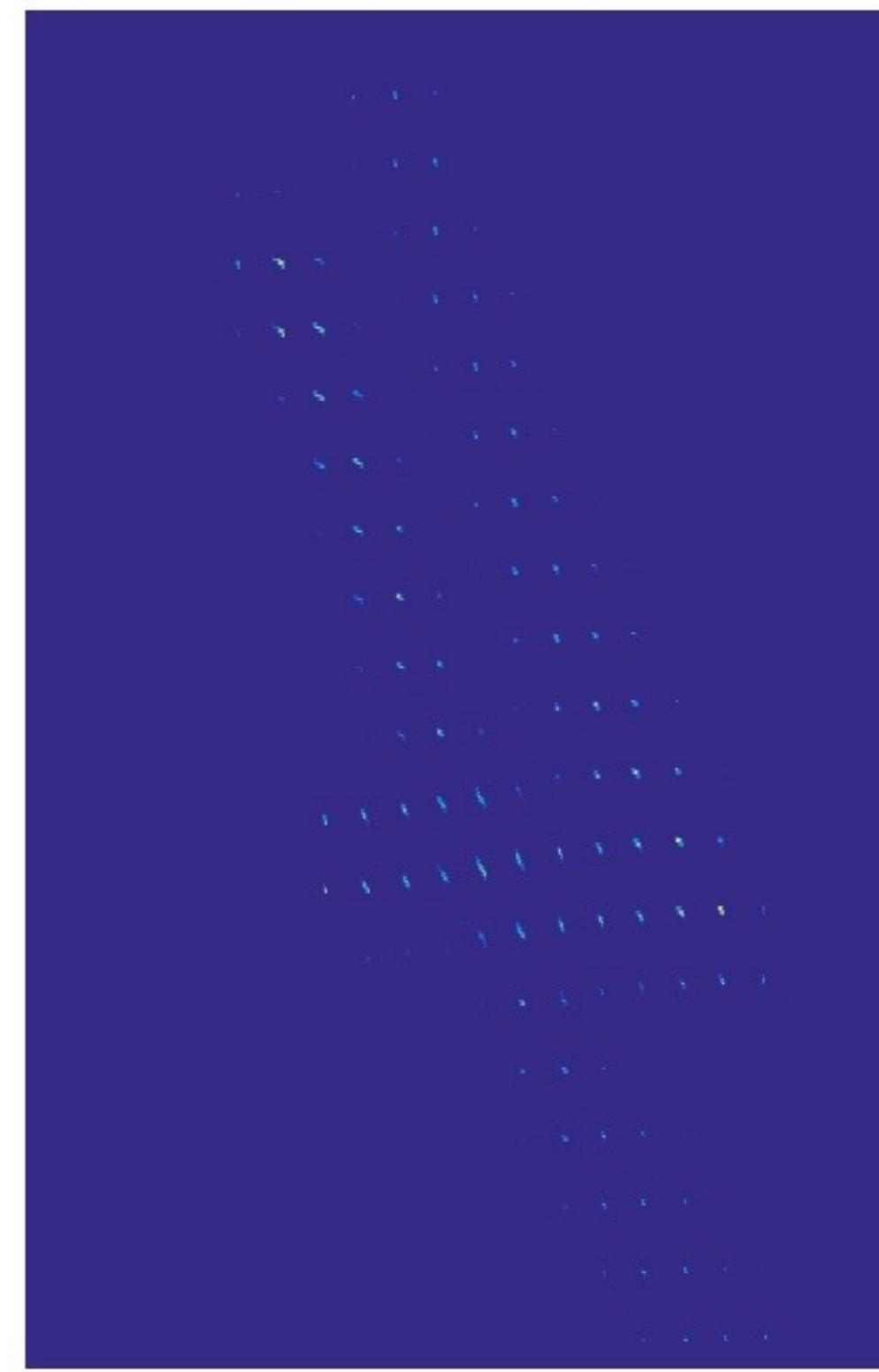
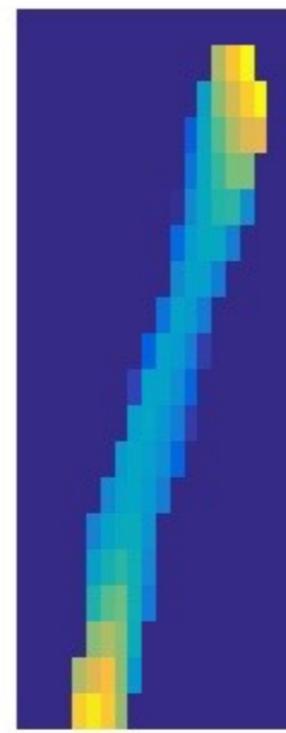
b



*Kv*₁₀

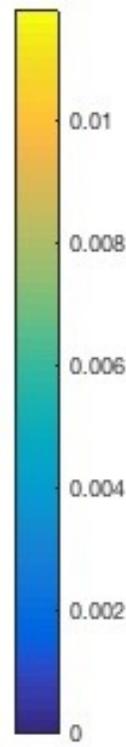
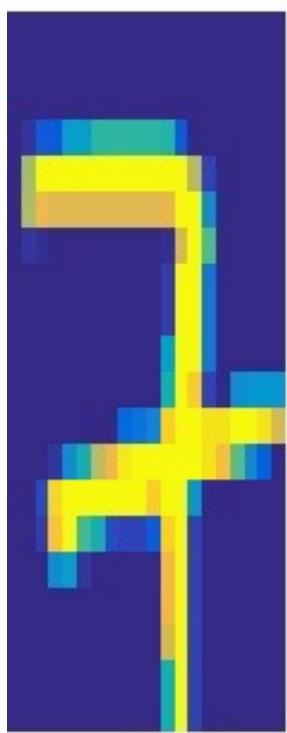


$\log(v_{10})$

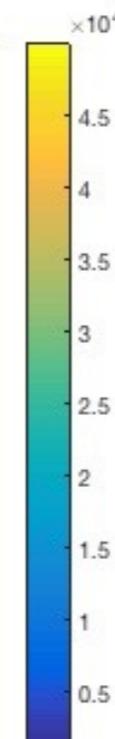


Very Fast EMD Approx. Solver

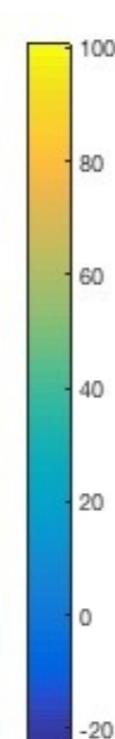
a



Ku_{10}



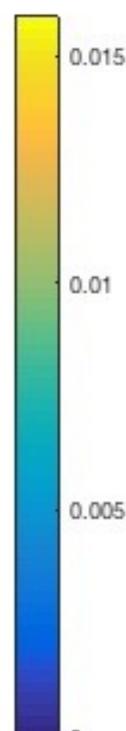
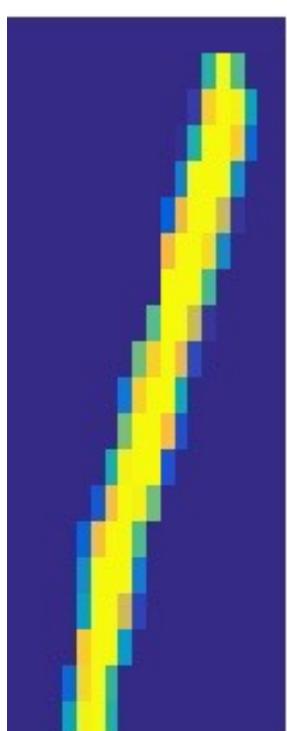
$\log(u_{10})$



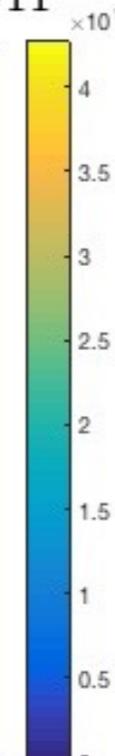
$$P_{10} = D(u_{10})KD(v_{10})$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.29009$$

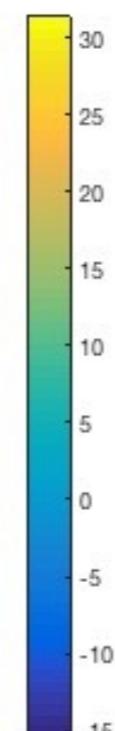
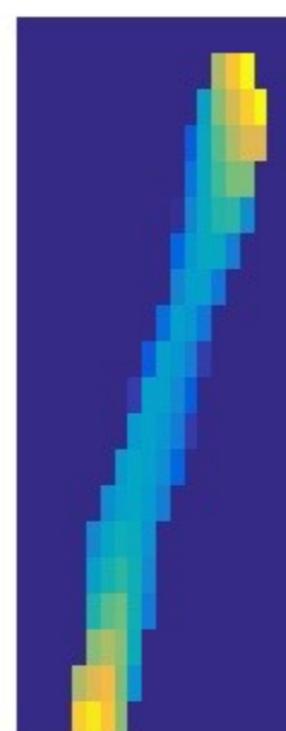
b



$v_{11} \leftarrow b/Ku_{11}$



$\log(v_{11})$



Sinkhorn as a Dual Algorithm

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

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

REGULARIZED DISCRETE PRIMAL

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

$$\text{where } \mathbf{K} = \left[e^{-\frac{\mathbf{D}^p(\mathbf{x}_i, \mathbf{y}_j)}{\gamma}} \right]_{ij}$$

REGULARIZED DISCRETE DUAL

Sinkhorn = *Block Coordinate Ascent* on Dual

Block Coordinate Ascent, a.k.a Sinkhorn

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

REGULARIZED DISCRETE DUAL

$$\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T \mathbf{a} + \boldsymbol{\beta}^T \mathbf{b} - \gamma(e^{\boldsymbol{\alpha}/\gamma})^T \mathbf{K} e^{\boldsymbol{\beta}/\gamma}$$

Block Coordinate Ascent, a.k.a Sinkhorn

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

REGULARIZED DISCRETE DUAL

$$\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T \mathbf{a} + \boldsymbol{\beta}^T \mathbf{b} - \gamma(e^{\boldsymbol{\alpha}/\gamma})^T \mathbf{K} e^{\boldsymbol{\beta}/\gamma}$$

$$\nabla_{\boldsymbol{\alpha}} \mathcal{E} = \mathbf{a} - e^{\boldsymbol{\alpha}/\gamma} \odot \mathbf{K} e^{\boldsymbol{\beta}/\gamma}$$

$$\nabla_{\boldsymbol{\beta}} \mathcal{E} = \mathbf{b} - e^{\boldsymbol{\beta}/\gamma} \odot \mathbf{K}^T e^{\boldsymbol{\alpha}/\gamma}$$

Block Coordinate Ascent, a.k.a Sinkhorn

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

REGULARIZED DISCRETE DUAL

$$\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T \mathbf{a} + \boldsymbol{\beta}^T \mathbf{b} - \gamma(e^{\boldsymbol{\alpha}/\gamma})^T \mathbf{K} e^{\boldsymbol{\beta}/\gamma}$$

$$\nabla_{\boldsymbol{\alpha}} \mathcal{E} = \mathbf{a} - e^{\boldsymbol{\alpha}/\gamma} \odot \mathbf{K} e^{\boldsymbol{\beta}/\gamma}$$

$$\boldsymbol{\alpha} \leftarrow \gamma \left(\log \mathbf{a} - \log \mathbf{K} e^{\boldsymbol{\beta}/\gamma} \right)$$

$$\nabla_{\boldsymbol{\beta}} \mathcal{E} = \mathbf{b} - e^{\boldsymbol{\beta}/\gamma} \odot \mathbf{K}^T e^{\boldsymbol{\alpha}/\gamma}$$

$$\boldsymbol{\beta} \leftarrow \gamma \left(\log \mathbf{b} - \log \mathbf{K}^T (e^{\boldsymbol{\alpha}/\gamma}) \right)$$

Block Coordinate Ascent, a.k.a Sinkhorn

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

REGULARIZED DISCRETE DUAL

Block Coordinate Ascent, a.k.a Sinkhorn

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

REGULARIZED DISCRETE DUAL

$$(\mathbf{u}, \mathbf{v}) \stackrel{\text{def}}{=} (e^{\boldsymbol{\alpha}/\gamma}, e^{\boldsymbol{\beta}/\gamma})$$

$$\boldsymbol{\alpha} \leftarrow \gamma \left(\log \mathbf{a} - \log \mathbf{K}(e^{\boldsymbol{\beta}/\gamma}) \right)$$

$$\mathbf{u} \leftarrow \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}}$$

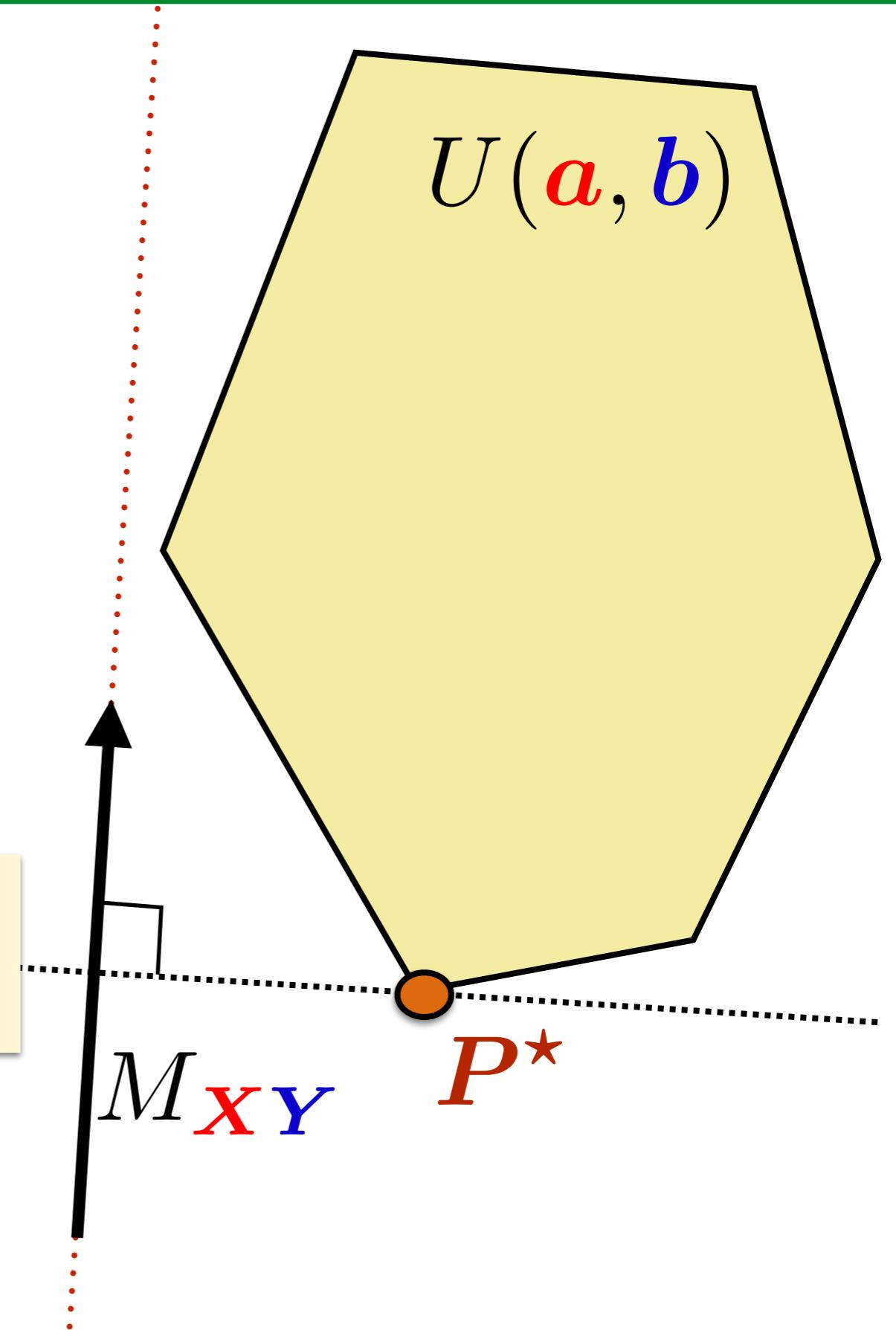
$$\boldsymbol{\beta} \leftarrow \gamma \left(\log \mathbf{b} - \log \mathbf{K}^T(e^{\boldsymbol{\alpha}/\gamma}) \right)$$

$$\mathbf{v} \leftarrow \frac{\mathbf{b}}{\mathbf{K}^T \mathbf{u}}$$

Sinkhorn in between W and MMD

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

$$W^p(\mu, \nu) = \langle P^\star, M_{\mathbf{XY}} \rangle$$

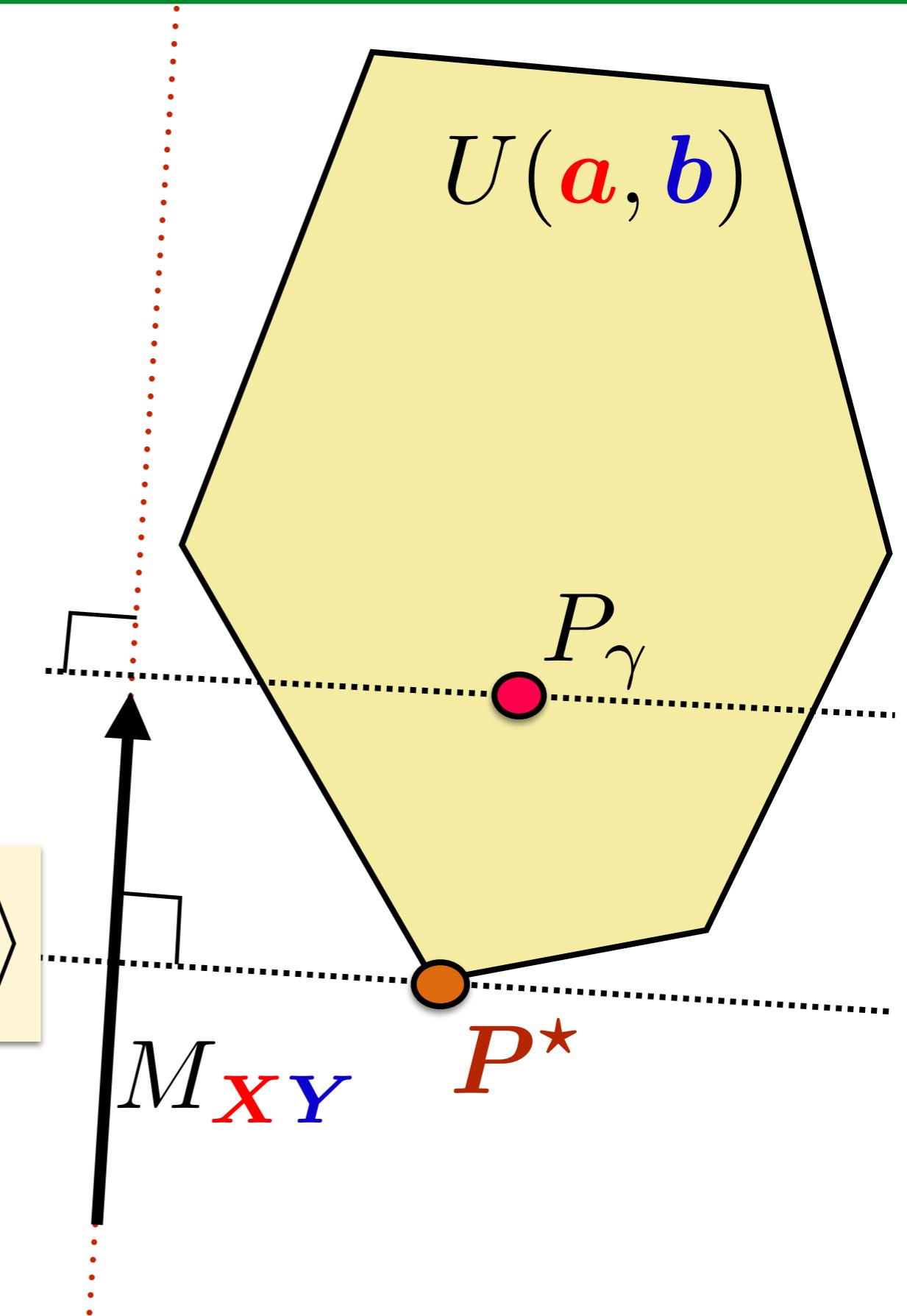


Sinkhorn in between W and MMD

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) = \langle P_\gamma, M_{\mathbf{X}\mathbf{Y}} \rangle$$

$$W^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \langle P^\star, M_{\mathbf{X}\mathbf{Y}} \rangle$$



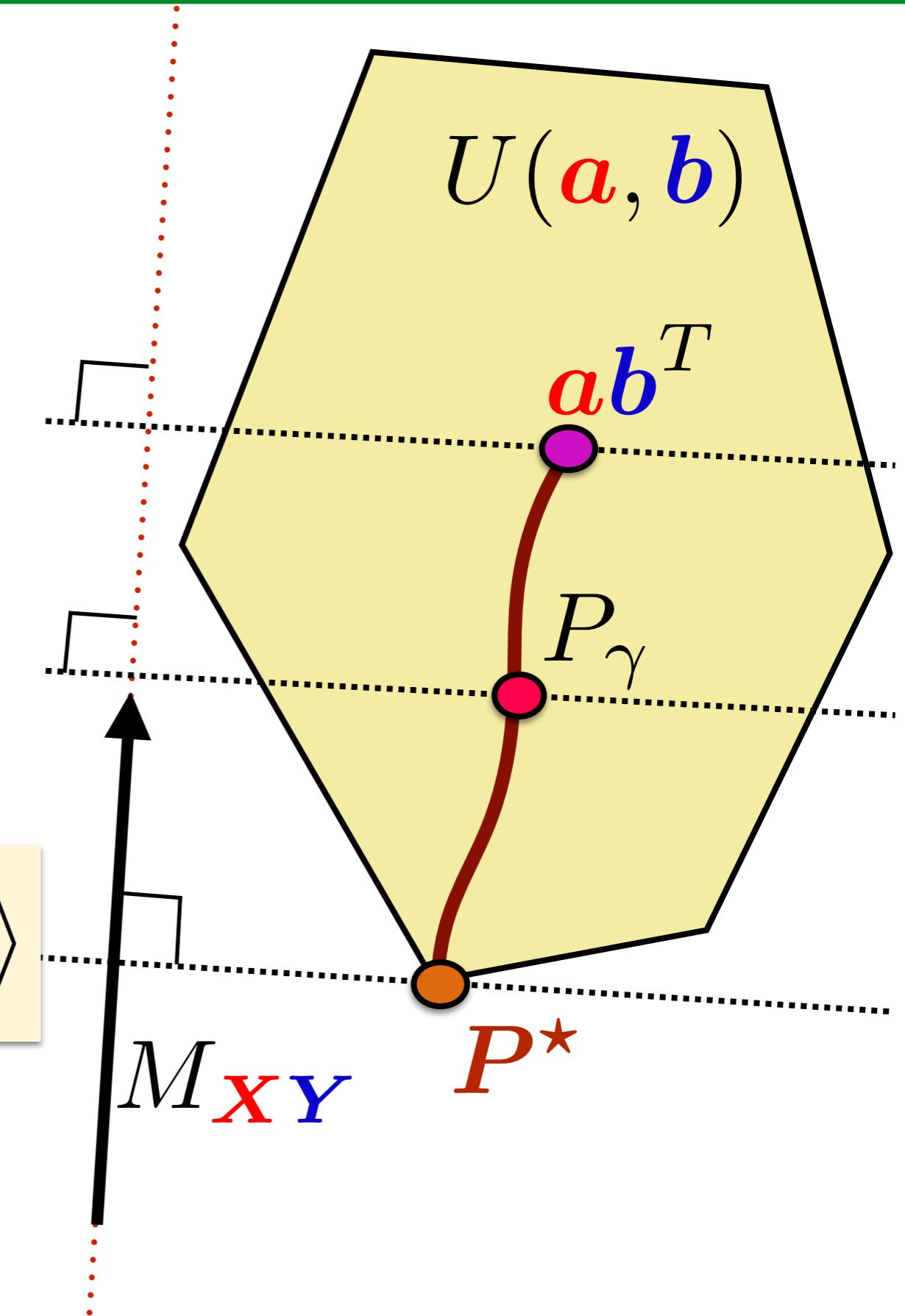
Sinkhorn in between W and MMD

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

$$\mathcal{E}(\mu, \nu) = \langle ab^T, M_{\mathbf{XY}} \rangle$$

$$W_\gamma(\mu, \nu) = \langle P_\gamma, M_{\mathbf{XY}} \rangle$$

$$W^p(\mu, \nu) = \langle P^\star, M_{\mathbf{XY}} \rangle$$



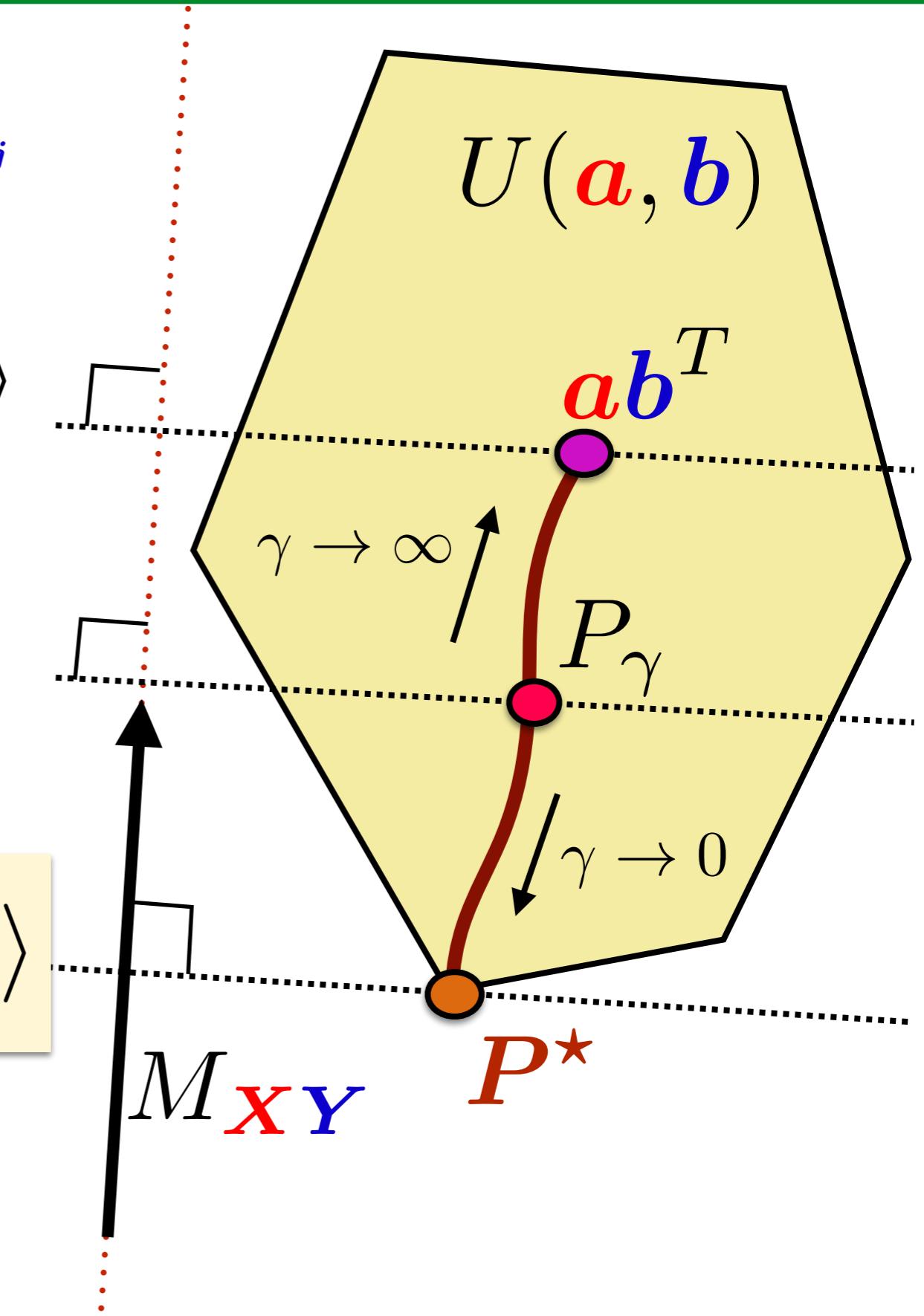
Sinkhorn in between W and MMD

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

$$\mathcal{E}(\mu, \nu) = \langle ab^T, M_{\mathbf{XY}} \rangle$$

$$W_\gamma(\mu, \nu) = \langle P_\gamma, M_{\mathbf{XY}} \rangle$$

$$W^p(\mu, \nu) = \langle P^\star, M_{\mathbf{XY}} \rangle$$



Sinkhorn in between W and MMD

$$\mathcal{E}(\mu, \nu) = \langle \textcolor{red}{ab}^T, M_{\mathbf{XY}} \rangle$$

$$\mathcal{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$$

$$W_\gamma(\mu, \nu) = \langle P_\gamma, M_{\mathbf{XY}} \rangle$$

$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$$W^p(\mu, \nu) = \langle \textcolor{brown}{P}^\star, M_{\mathbf{XY}} \rangle$$

Sinkhorn in between W and MMD

$$\mathcal{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$$

$\gamma \rightarrow \infty$



$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$\gamma \rightarrow 0$



$$W^p(\mu, \nu) = \langle P^\star, M_{XY} \rangle$$

How to compare them?

i.i.d samples $\textcolor{red}{x}_1, \dots, \textcolor{red}{x}_n \sim \mu$, $\textcolor{blue}{y}_1, \dots, \textcolor{blue}{y}_m \sim \nu$,

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\textcolor{red}{x}_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\textcolor{blue}{y}_j}$$

Computational properties

Effort to compute/approximate $\Delta(\hat{\mu}_n, \hat{\nu}_m)$?

Statistical properties

$$|\Delta(\mu, \nu) - \Delta(\hat{\mu}_n, \hat{\nu}_n)| \leq f(n)?$$

Sinkhorn in between W and MMD

$$\mathcal{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$$

$(n + m)^2$

$O(1/\sqrt{n})$

[see Arthur]

$$W^p(\mu, \nu) = \langle P^\star, M_{\mathbf{XY}} \rangle$$

$O((n + m)nm \log(n + m))$

$O(1/n^{1/d})$

Sinkhorn in between W and MMD

$$\mathcal{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$$

$(n + m)^2$

$O(1/\sqrt{n})$

[see Arthur]

$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$O((n + m)^2)$

$O\left(\frac{1}{\gamma^{d/2}\sqrt{n}}\right)$

[GCBP'18]

[FSVATP'18]

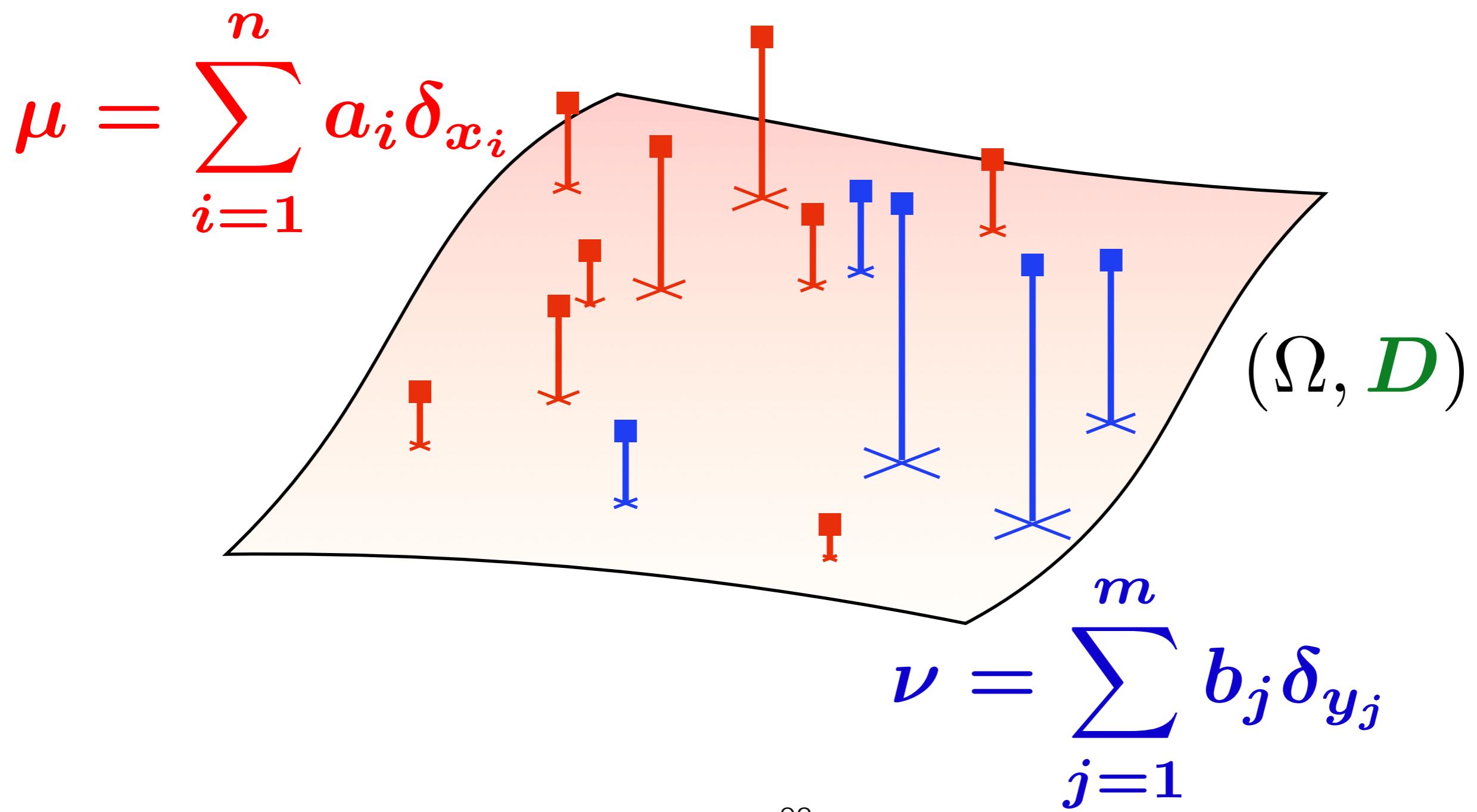
$$W^p(\mu, \nu) = \langle P^\star, M_{\mathbf{XY}} \rangle$$

$O((n + m)nm \log(n + m))$

$O(1/n^{1/d})$

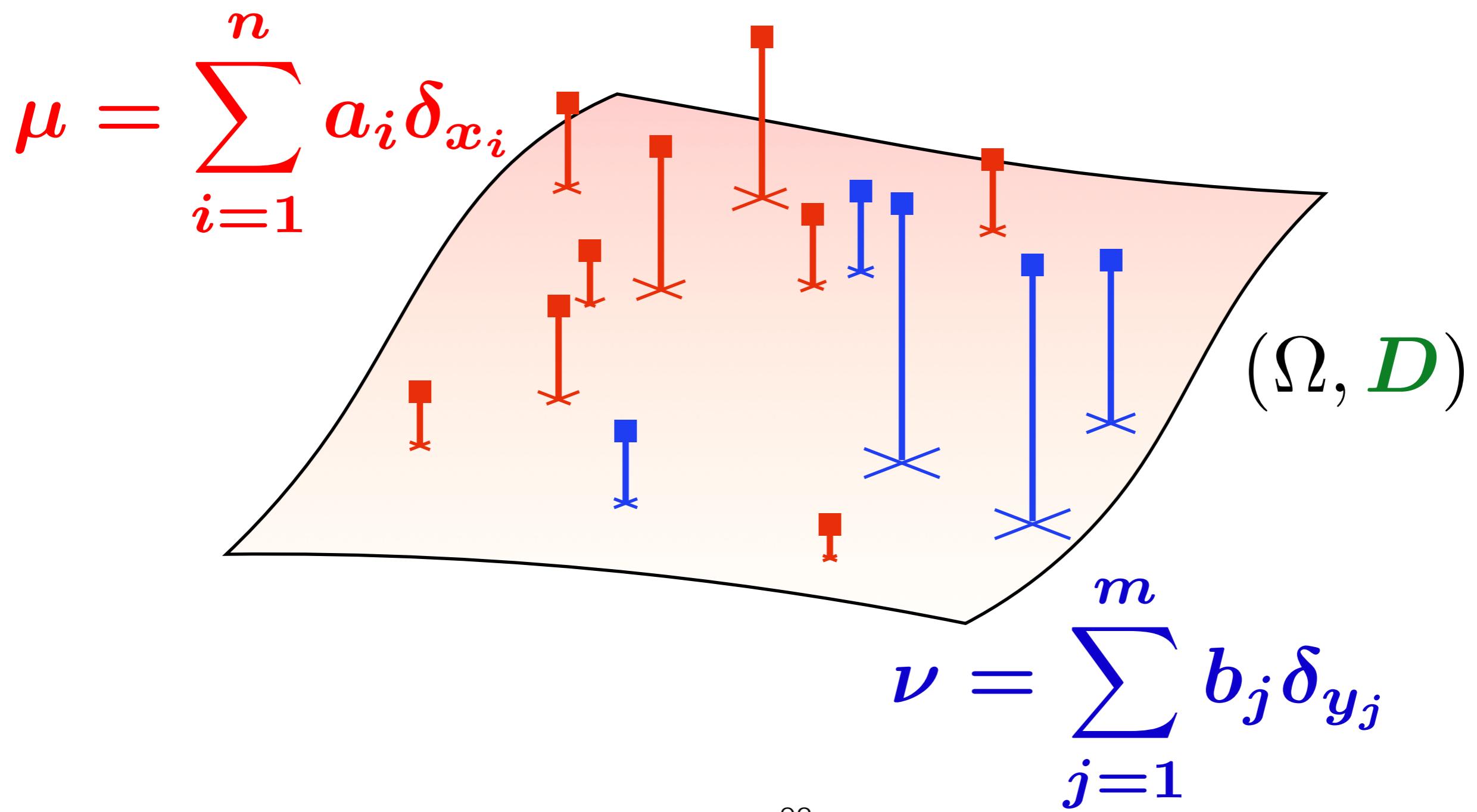
Differentiability of W

$$W((a, X), (b, Y))$$



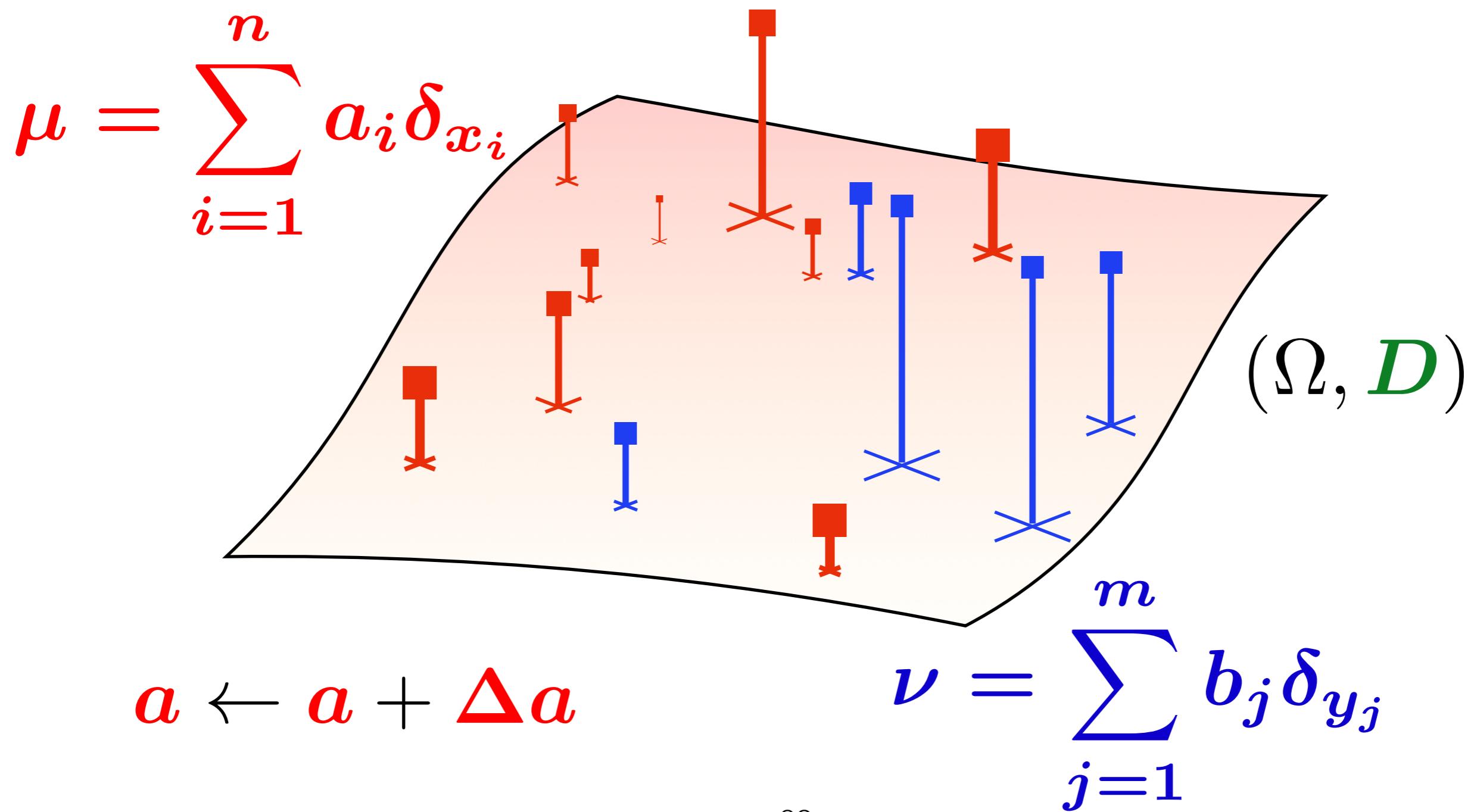
Differentiability of W

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



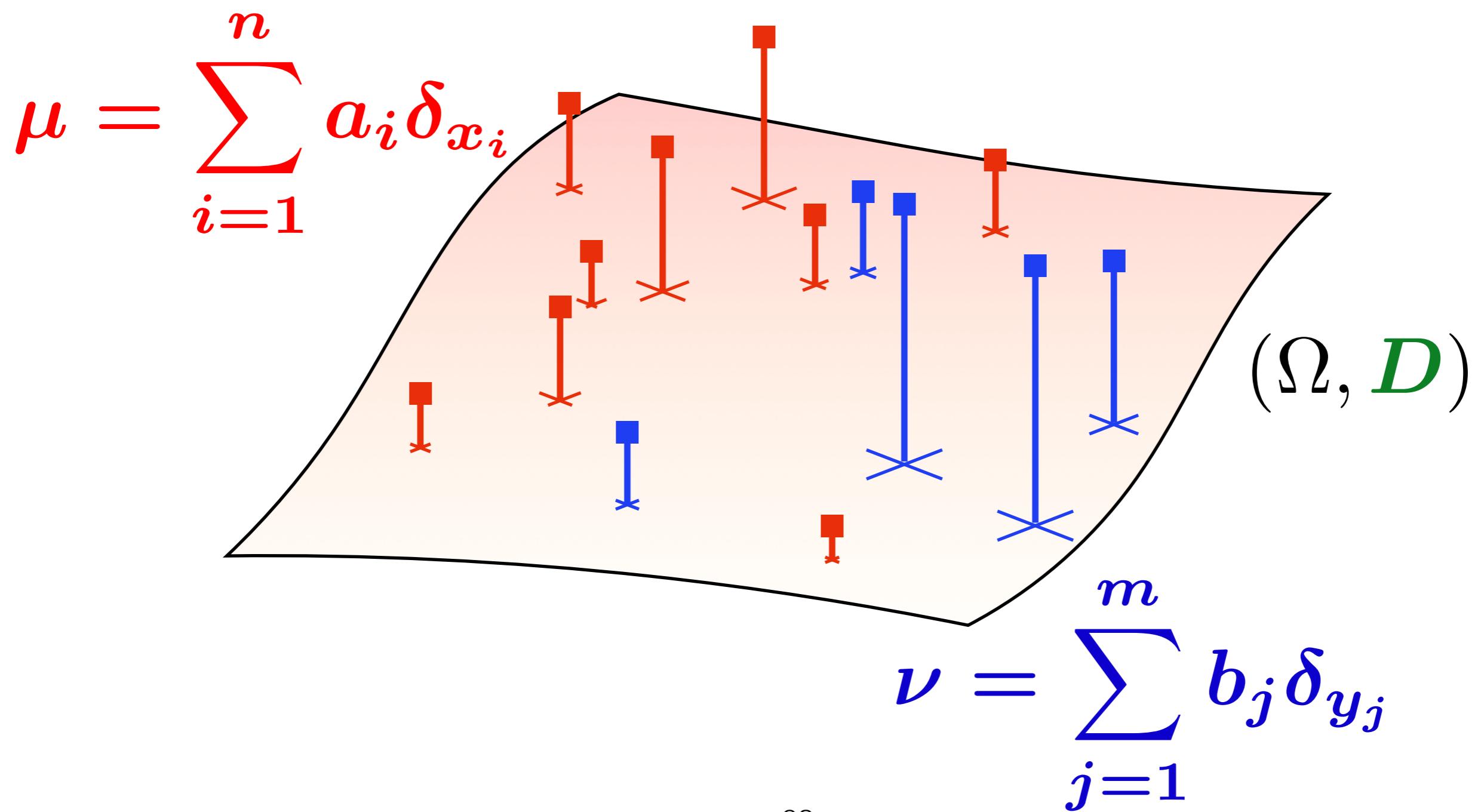
Differentiability of W

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



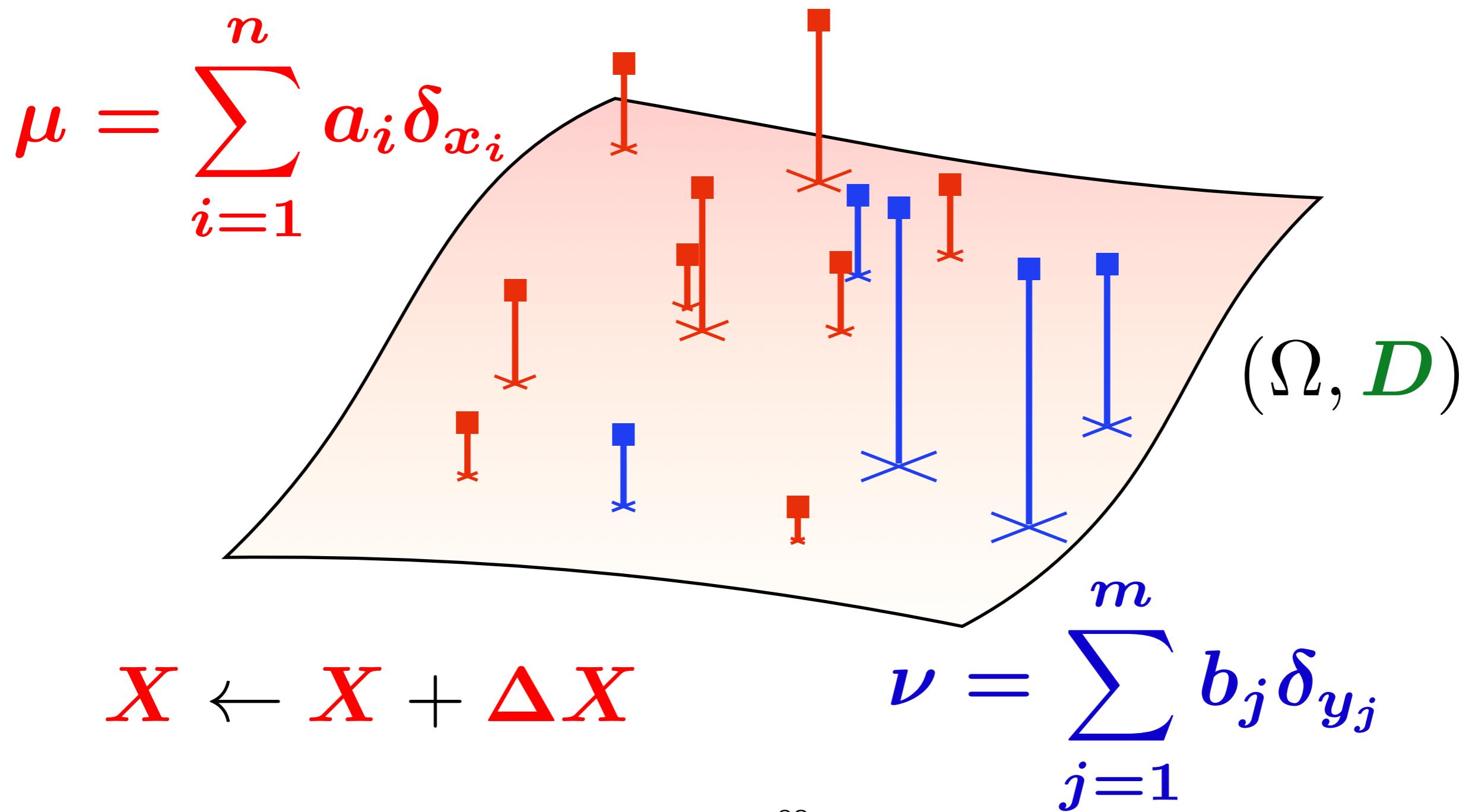
Sinkhorn \longrightarrow Differentiability

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



Sinkhorn \longrightarrow Differentiability

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



How to decrease W ? change weights

$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^m \\ \boldsymbol{\alpha} \oplus \boldsymbol{\beta} \leq M_{\mathbf{XY}}}} \boldsymbol{\alpha}^T \mathbf{a} + \boldsymbol{\beta}^T \mathbf{b}.$$

DUAL

Prop. $W(\boldsymbol{\mu}, \boldsymbol{\nu})$ is convex w.r.t. \mathbf{a} ,

$$\partial_{\mathbf{a}} W = \arg_{\boldsymbol{\alpha}} \max_{\substack{\boldsymbol{\alpha} \oplus \boldsymbol{\beta} \leq M_{\mathbf{XY}}}} \boldsymbol{\alpha}^T \mathbf{a} + \boldsymbol{\beta}^T \mathbf{b}.$$

Prop. $W_{\gamma}(\boldsymbol{\mu}, \boldsymbol{\nu})$ is convex and differentiable w.r.t. \mathbf{a} , $\nabla_{\mathbf{a}} W_{\gamma} = \boldsymbol{\alpha}_{\gamma}^{\star} = \gamma \log \mathbf{u}$

How to decrease W ? change locations

$$W_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\substack{\mathbf{P} \in \mathbb{R}_+^{n \times m} \\ \mathbf{P}\mathbf{1}_m = \mathbf{a}, \mathbf{P}^T\mathbf{1}_n = \mathbf{b}}} \langle \mathbf{P}, \mathbf{1}_n \mathbf{1}_d^T \mathbf{X}^2 + \mathbf{Y}^{2T} \mathbf{1}_d \mathbf{1}_m - 2\mathbf{X}^T \mathbf{Y} \rangle$$

PRIMAL

Prop. $p = 2, \Omega = \mathbb{R}^d$. $W(\boldsymbol{\mu}, \boldsymbol{\nu})$ decreases if
 $\mathbf{X} \leftarrow \mathbf{Y} P^{\star T} \mathbf{D}(\mathbf{a}^{-1})$.

Prop. $p = 2, \Omega = \mathbb{R}^d$. $W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu})$ is differentiable w.r.t. \mathbf{X} , with

$$\nabla_{\mathbf{X}} W_\gamma = \mathbf{X} - \mathbf{Y} P_\gamma^T \mathbf{D}(\mathbf{a}^{-1}).$$

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

$$W_L(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \langle \mathbf{P}_L, M_{\mathbf{X}\mathbf{Y}} \rangle,$$

where $\mathbf{P}_L \stackrel{\text{def}}{=} \text{diag}(\mathbf{u}_L) \mathbf{K} \text{diag}(\mathbf{v}_L)$,

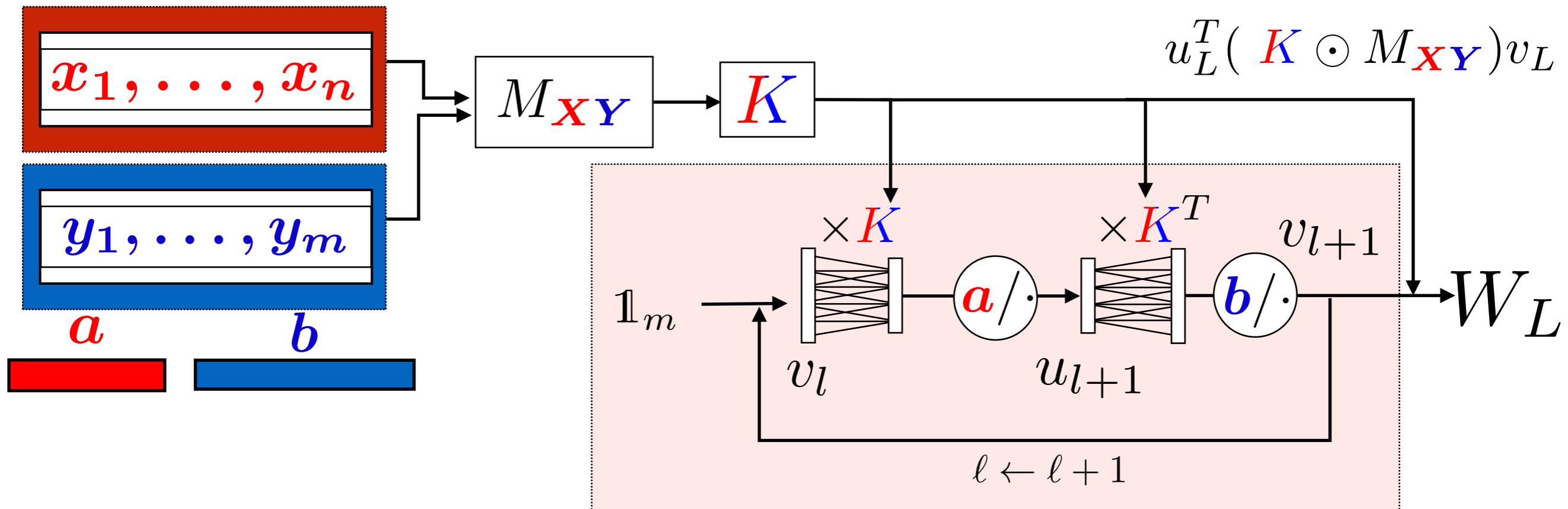
$\mathbf{v}_0 = \mathbf{1}_m$; $l \geq 0$, $\mathbf{u}_l \stackrel{\text{def}}{=} \mathbf{a}/K\mathbf{v}_l$, $\mathbf{v}_{l+1} \stackrel{\text{def}}{=} \mathbf{b}/K^T \mathbf{u}_l$.

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

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

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



Sinkhorn $\ell = 1, \dots, L - 1$

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

$$W_L(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \langle \mathbf{P}_L, M_{\mathbf{X}\mathbf{Y}} \rangle,$$

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

[Hashimoto'16] [Bonneel'16][Shalit'16]

Projecting to Regularize

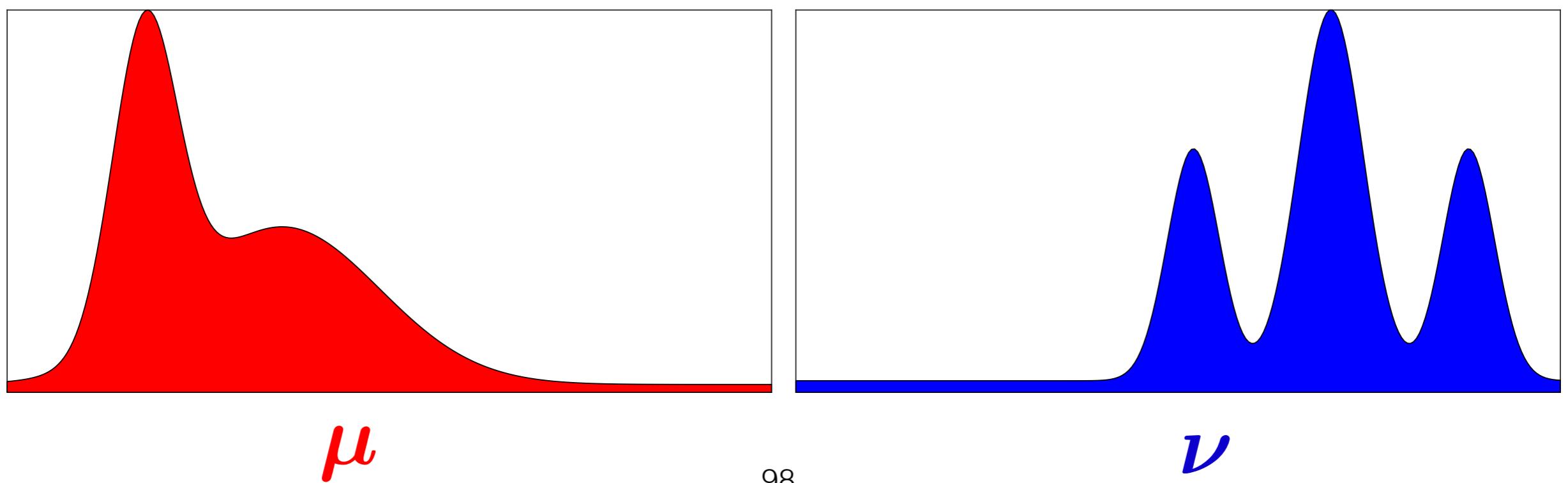
Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$

Projecting to Regularize

Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

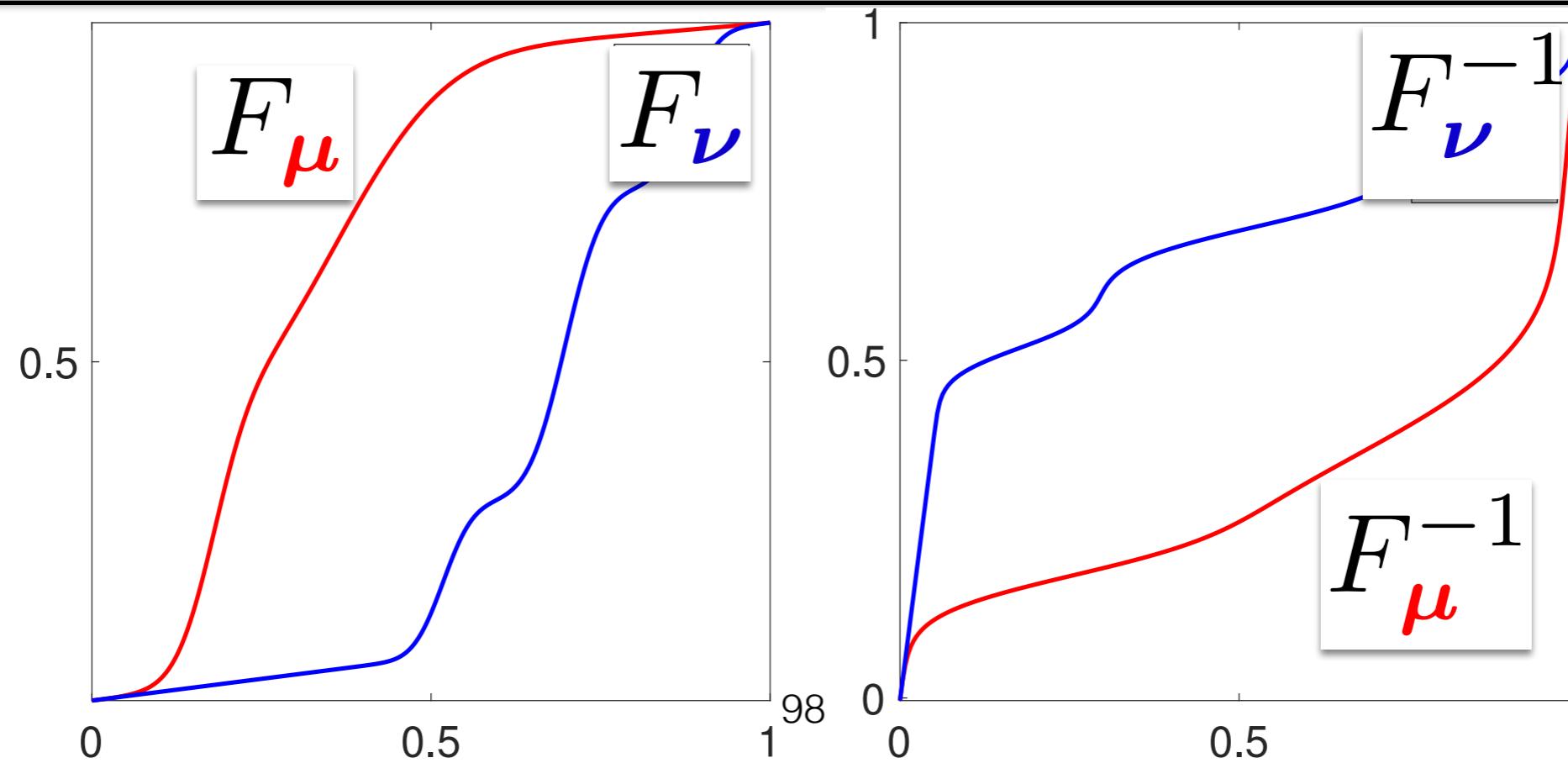
$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Projecting to Regularize

Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

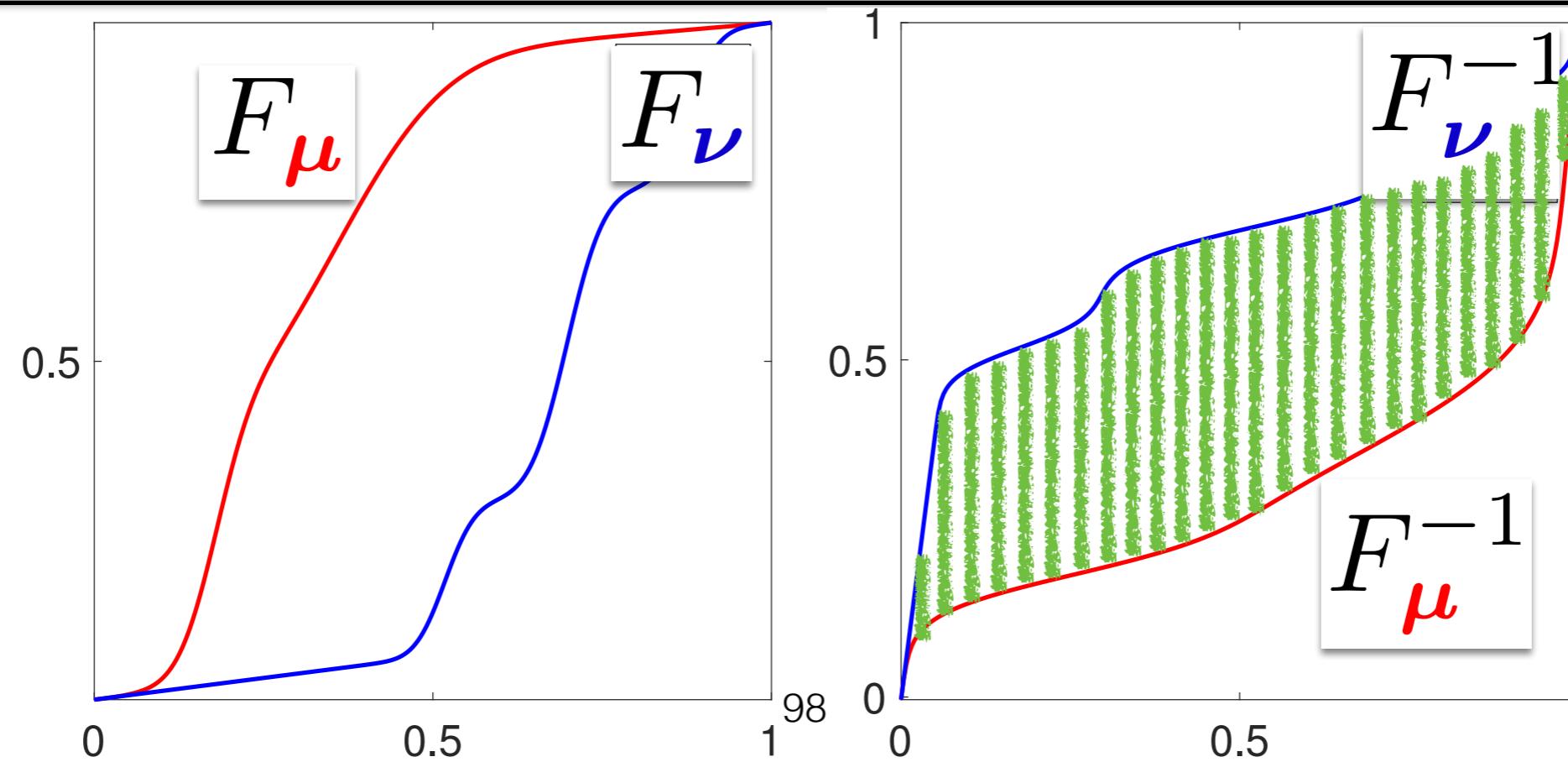
$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Projecting to Regularize

Remark. If $\Omega = \mathbb{R}$, $c(x, y) = c(|x - y|)$,
 c convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 c(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Projecting to Regularize

Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$

Sliced Wasserstein Distance [Rabin+'11]

$$SW(\mu, \nu) = \mathbb{E}_{\theta \sim \mathcal{S}^{d-1}} \left[\int_0^1 \textcolor{green}{c}(|F_{\theta_{\sharp}^T \mu}^{-1}(x) - F_{\theta_{\sharp}^T \nu}^{-1}(x)|) dx \right]$$

Recall: For Univariate measures

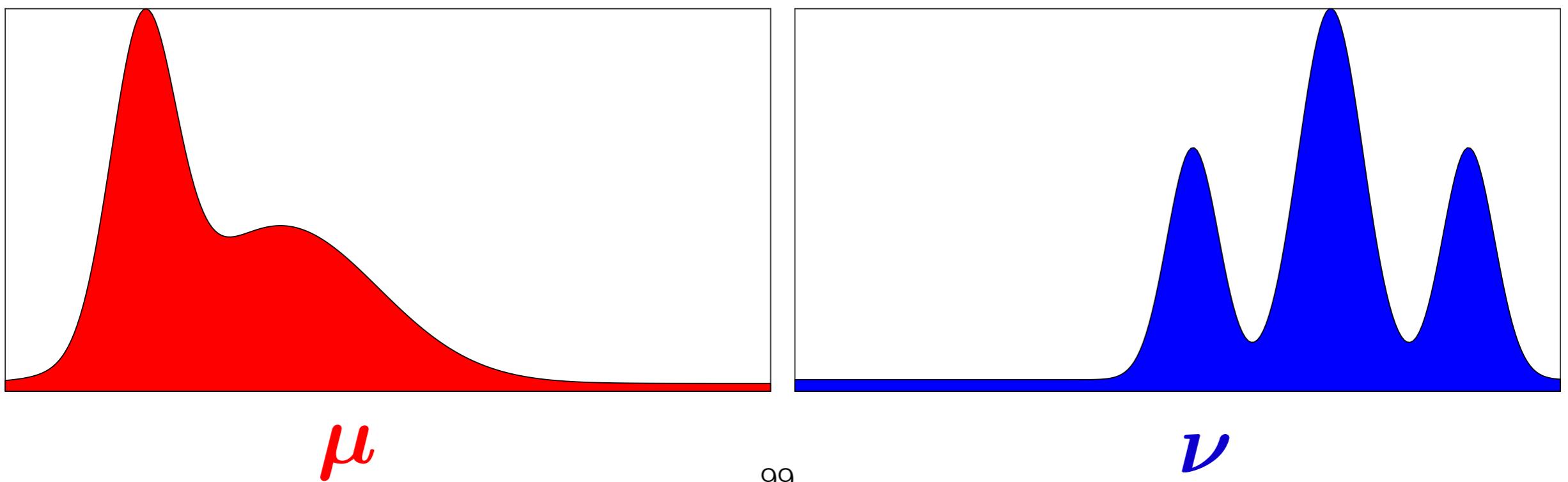
Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$

Recall: For Univariate measures

Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

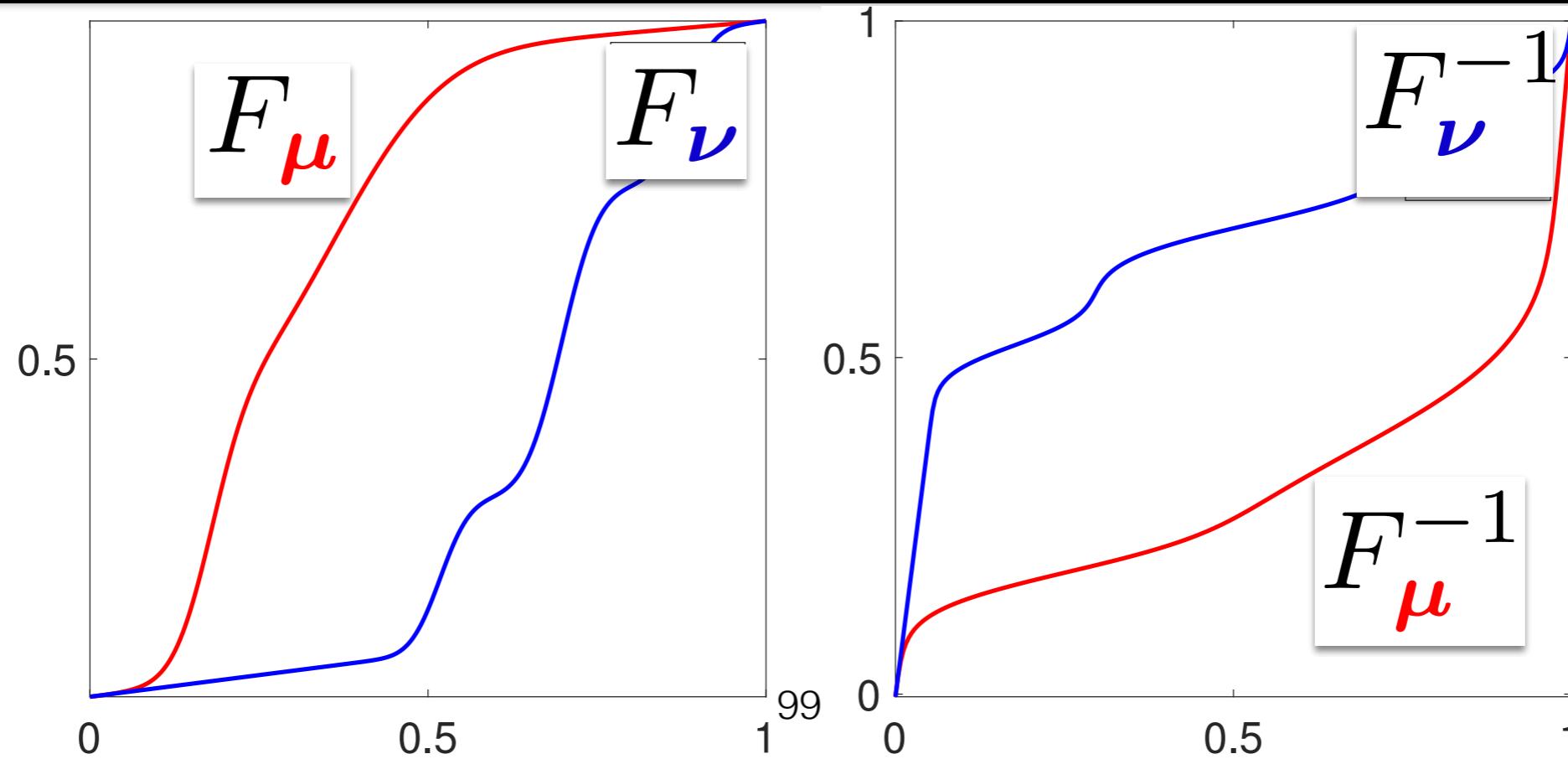
$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Recall: For Univariate measures

Remark. If $\Omega = \mathbb{R}$, $\textcolor{red}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

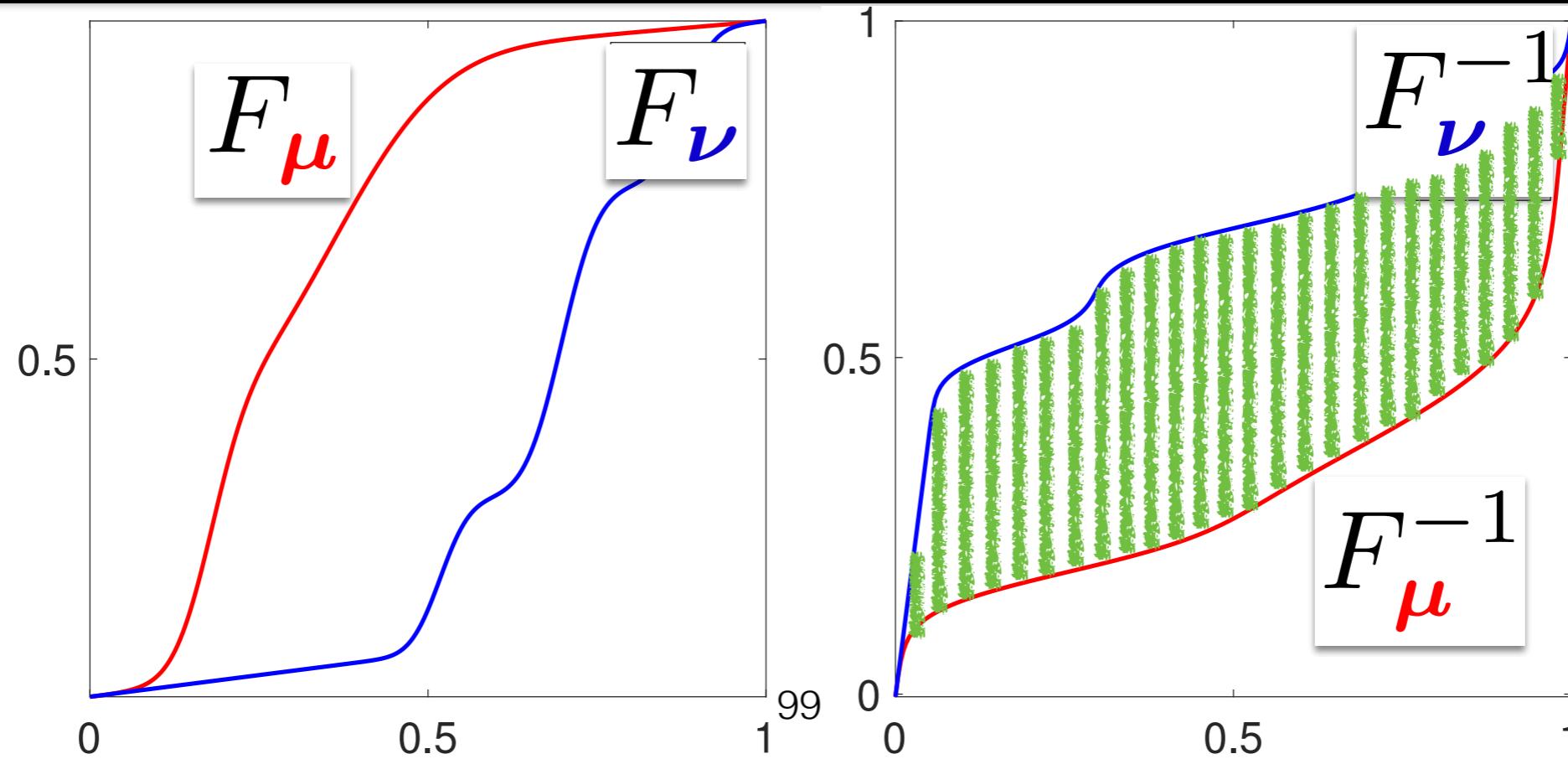
$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Recall: For Univariate measures

Remark. If $\Omega = \mathbb{R}$, $\textcolor{green}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\mu}^{-1}, F_{\nu}^{-1}$ quantile functions,

$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Trending in ML

- Dodges the high-dimensionality curse, although it's not clear what it computes, certainly not OT.
- Works very well in practice, fast and easy.

Trending in ML

Sliced Wasserstein Distance [Rabin+'11]

$$SW(\boldsymbol{\mu}, \boldsymbol{\nu}) = \mathbb{E}_{\theta \sim \mathcal{S}^{d-1}} \left[\int_0^1 \textcolor{green}{c}(|F_{\theta_{\sharp}^T \boldsymbol{\mu}}^{-1}(x) - F_{\theta_{\sharp}^T \boldsymbol{\nu}}^{-1}(x)|) dx \right]$$

- Dodges the high-dimensionality curse, although it's not clear what it computes, certainly not OT.
- Works very well in practice, fast and easy.

Project first on PCA, Wasserstein next

To handle high-dimensionality, PCA is often used to project data first, compute W next.

$$\text{var}\left(\frac{\boldsymbol{\mu}+\boldsymbol{\nu}}{2}\right) = E\Lambda E^T$$

$$\hat{\mathbf{L}}_k := \begin{bmatrix} \vdots & \vdots & \vdots \\ e_1 & \cdots & e_k \\ \vdots & \vdots & \vdots \end{bmatrix} \Lambda_k^{-\frac{1}{2}} \begin{bmatrix} \vdots & \vdots & \vdots \\ e_1 & \cdots & e_k \\ \vdots & \vdots & \vdots \end{bmatrix}^T$$

$$\tilde{W}_{p,k}(\boldsymbol{\mu}, \boldsymbol{\nu}) = W_p(\hat{\mathbf{L}}_{k\sharp}^T \boldsymbol{\mu}, \hat{\mathbf{L}}_{k\sharp}^T \boldsymbol{\nu})$$

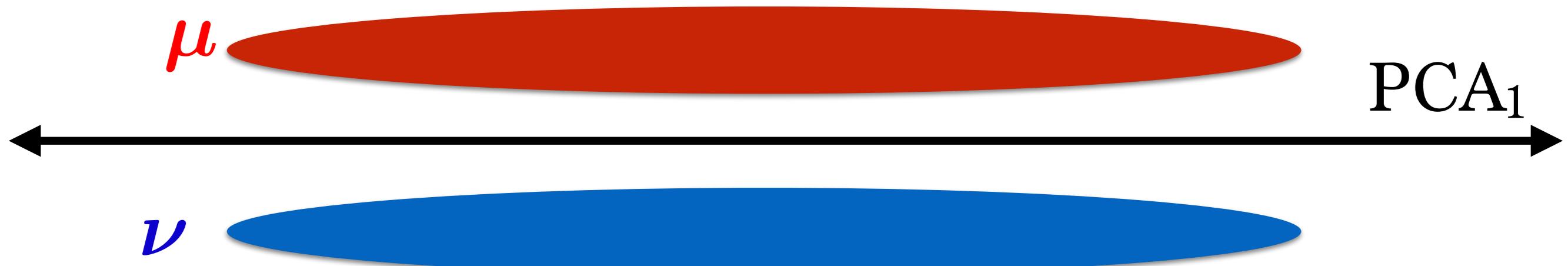
Project first on PCA, Wasserstein next

- Problem: when computed first, PCA may lose important information that matters in OT.



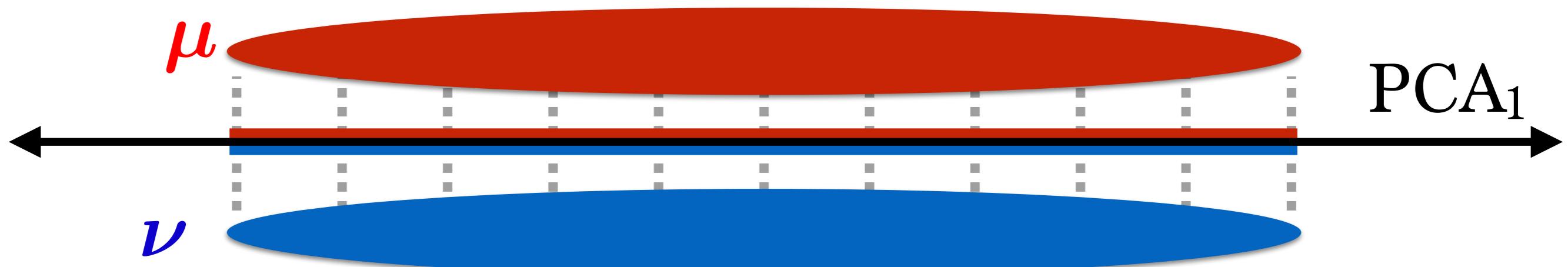
Project first on PCA, Wasserstein next

- Problem: when computed first, PCA may lose important information that matters in OT.



Project first on PCA, Wasserstein next

- Problem: when computed first, PCA may lose important information that matters in OT.



$$W = 0!$$

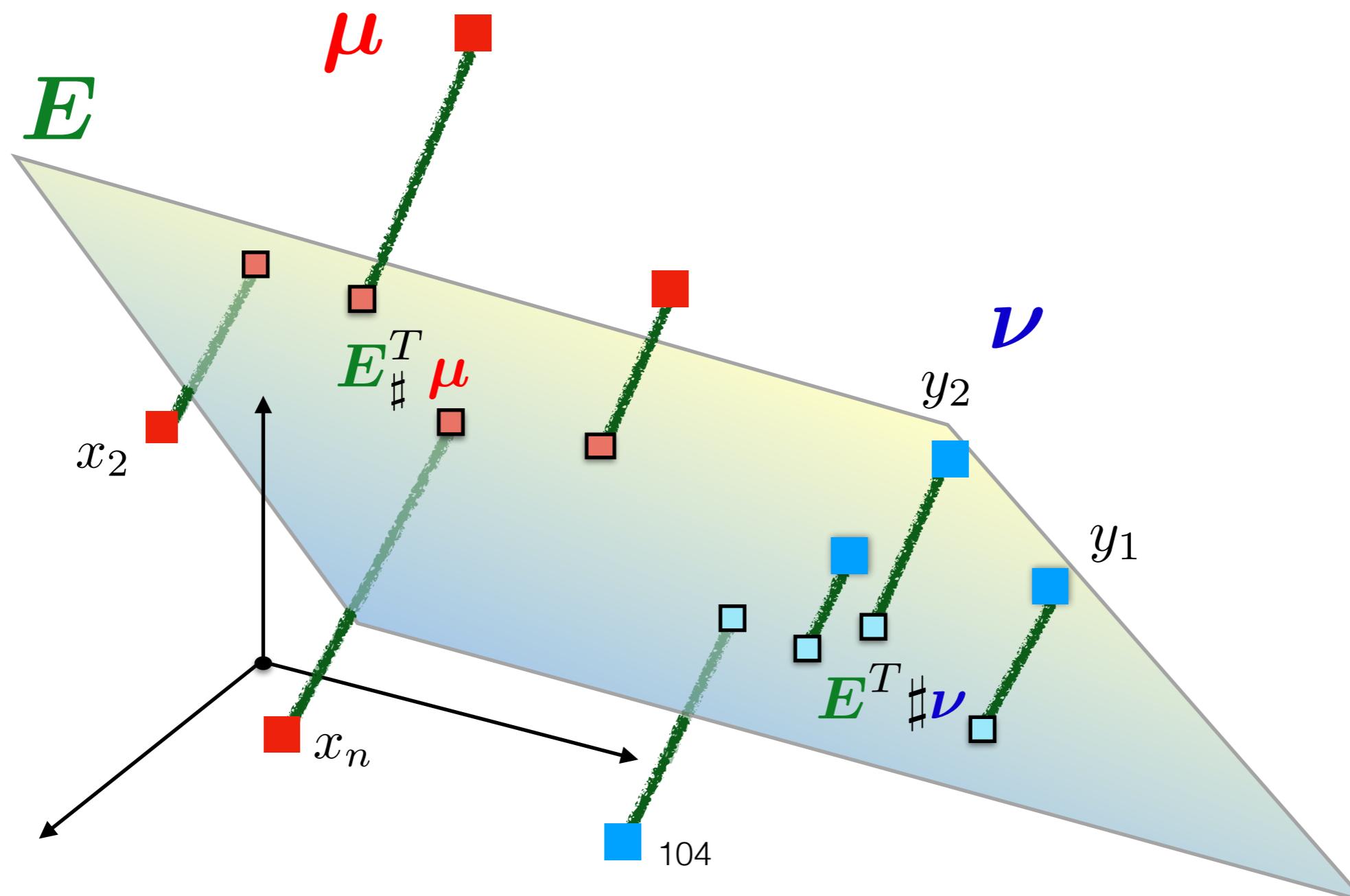
k -dim (robust) projection

Look instead for “worst” possible projection.

$$\mathcal{W}_{p,k}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} W_p(\mathbf{E}_\sharp^T \boldsymbol{\mu}, \mathbf{E}_\sharp^T \boldsymbol{\nu})$$

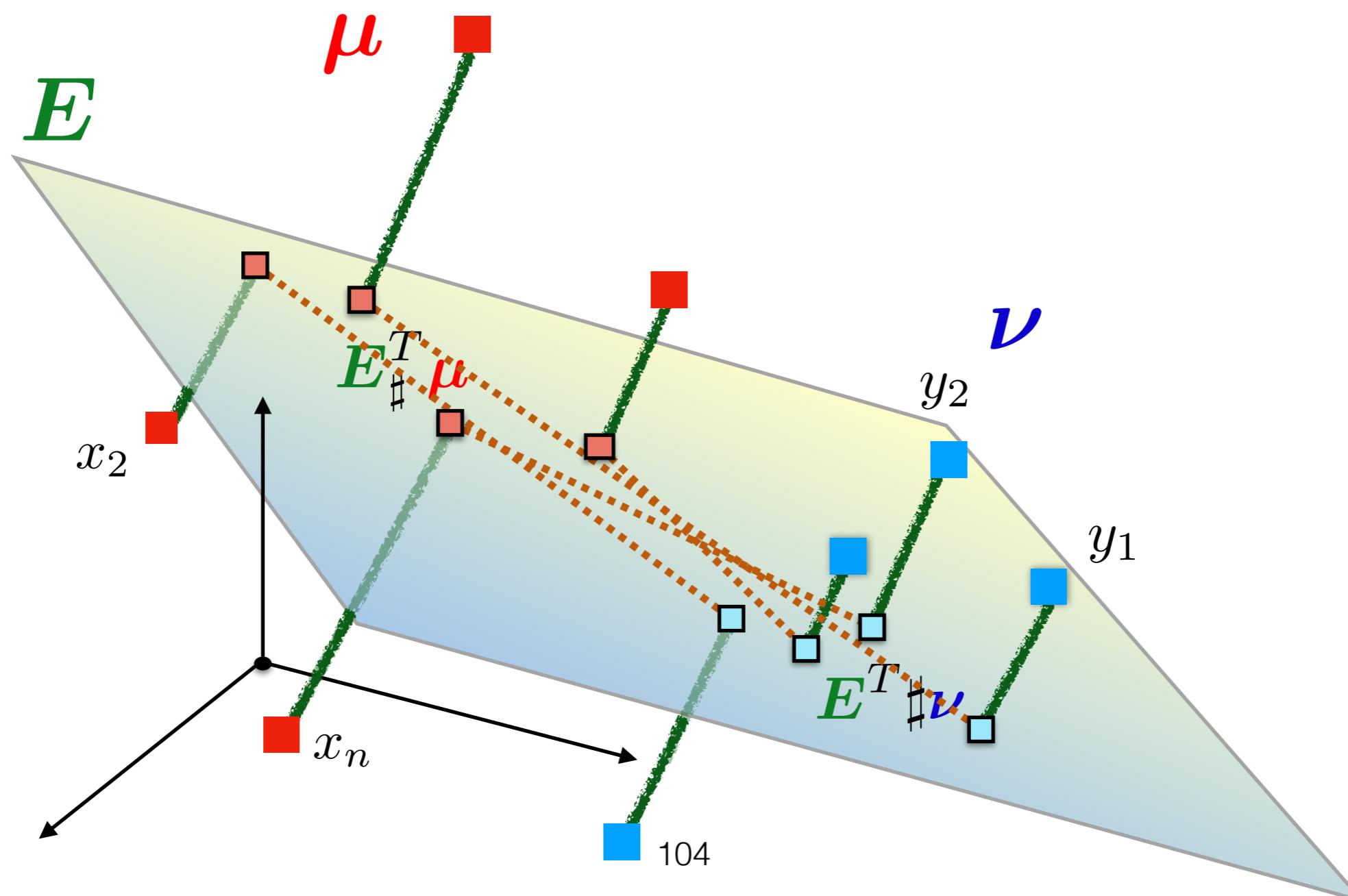
max-Wasserstein

$$\mathcal{W}_{p,k}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} W_p(\mathbf{E}_{\sharp}^T \boldsymbol{\mu}, \mathbf{E}_{\sharp}^T \boldsymbol{\nu})$$



max-Wasserstein

$$\mathcal{W}_{p,k}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} W_p(\mathbf{E}_\sharp^\top \boldsymbol{\mu}, \mathbf{E}_\sharp^\top \boldsymbol{\nu})$$



On k -dim (robust) projections

- Look instead for “worst” possible projection.

$$\mathcal{W}_{p,k}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} W_p(\mathbf{E}_\sharp^T \boldsymbol{\mu}, \mathbf{E}_\sharp^T \boldsymbol{\nu})$$

- [Weed+'19]: convergence for “well behaved” measures effectively supported on k dimensional subspace, would recover estimator with $O(n^{-1/k})$

Not convex! min/max?

On k -dim (robust) projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = \mathbf{I}_k}} \iint \|\mathbf{E}^T x - \mathbf{E}^T y\|_2^2 \mathbf{P}(dx, dy)$$

On k -dim (robust) projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = \mathbf{I}_k}} \iint \|\mathbf{E}^T x - \mathbf{E}^T y\|_2^2 \mathbf{P}(dx, dy)$$

$$\|\mathbf{E}^T x - \mathbf{E}^T y\|_2^2 = \langle \mathbf{E} \mathbf{E}^T, (x - y)(x - y)^T \rangle$$

On k -dim (robust) projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = \mathbf{I}_k}} \iint \|\mathbf{E}^T x - \mathbf{E}^T y\|_2^2 \mathbf{P}(dx, dy)$$

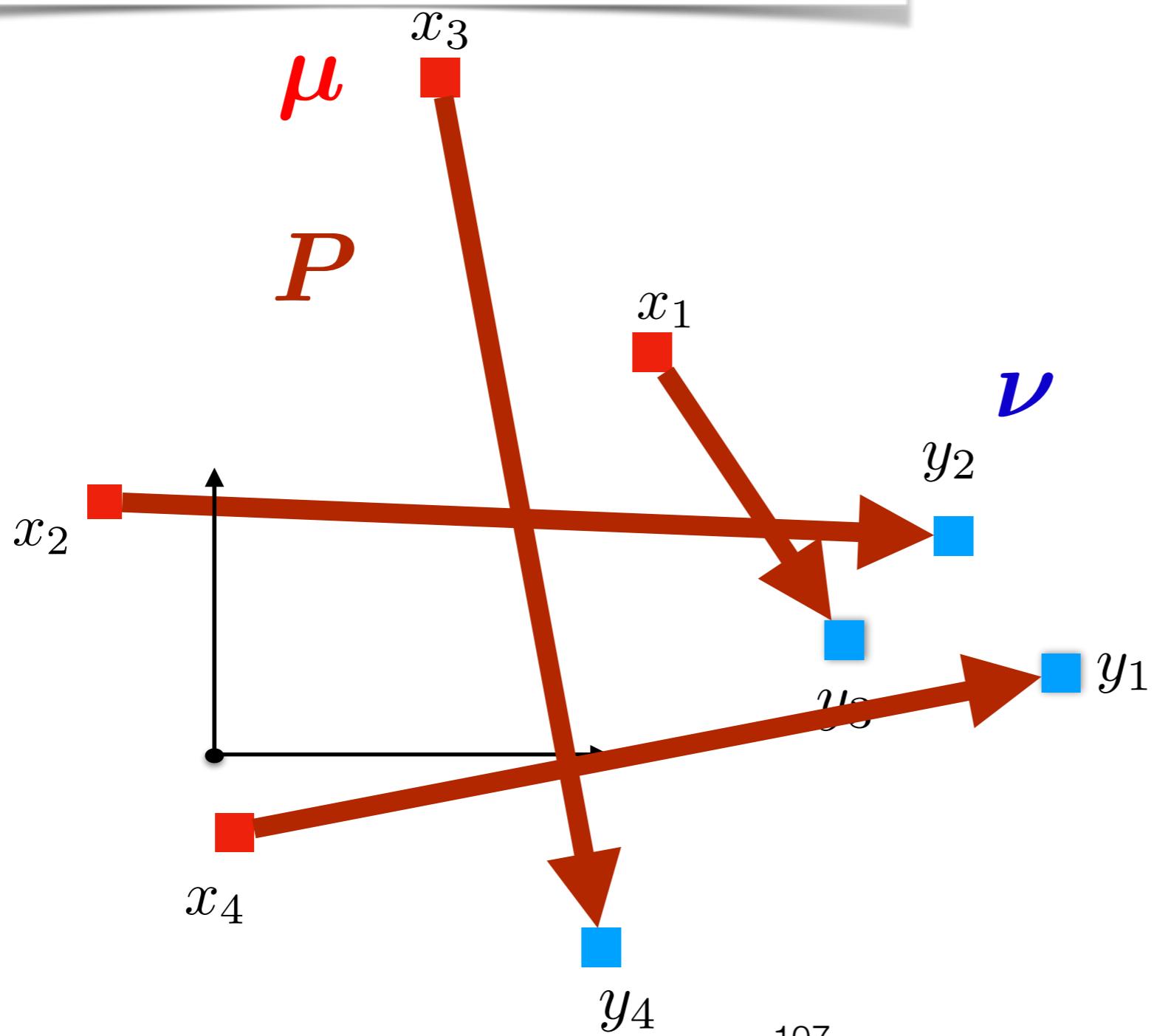
$$\|\mathbf{E}^T x - \mathbf{E}^T y\|_2^2 = \langle \mathbf{E} \mathbf{E}^T, (x - y)(x - y)^T \rangle$$

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = \mathbf{I}_k}} \langle \mathbf{E} \mathbf{E}^T, V_{\mathbf{P}} \rangle$$

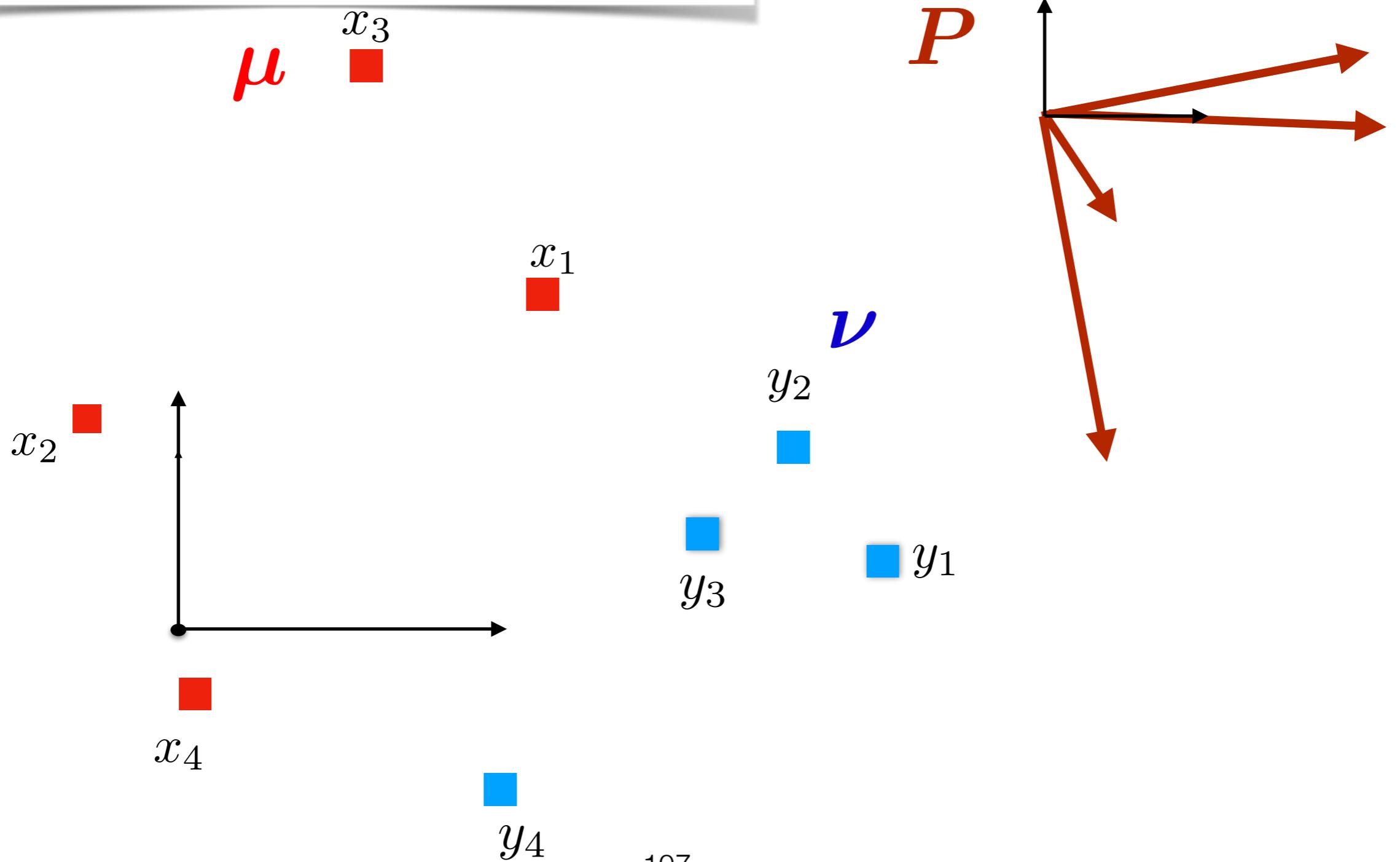
The $V_{\mathbf{P}}$ Moment Matrix

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



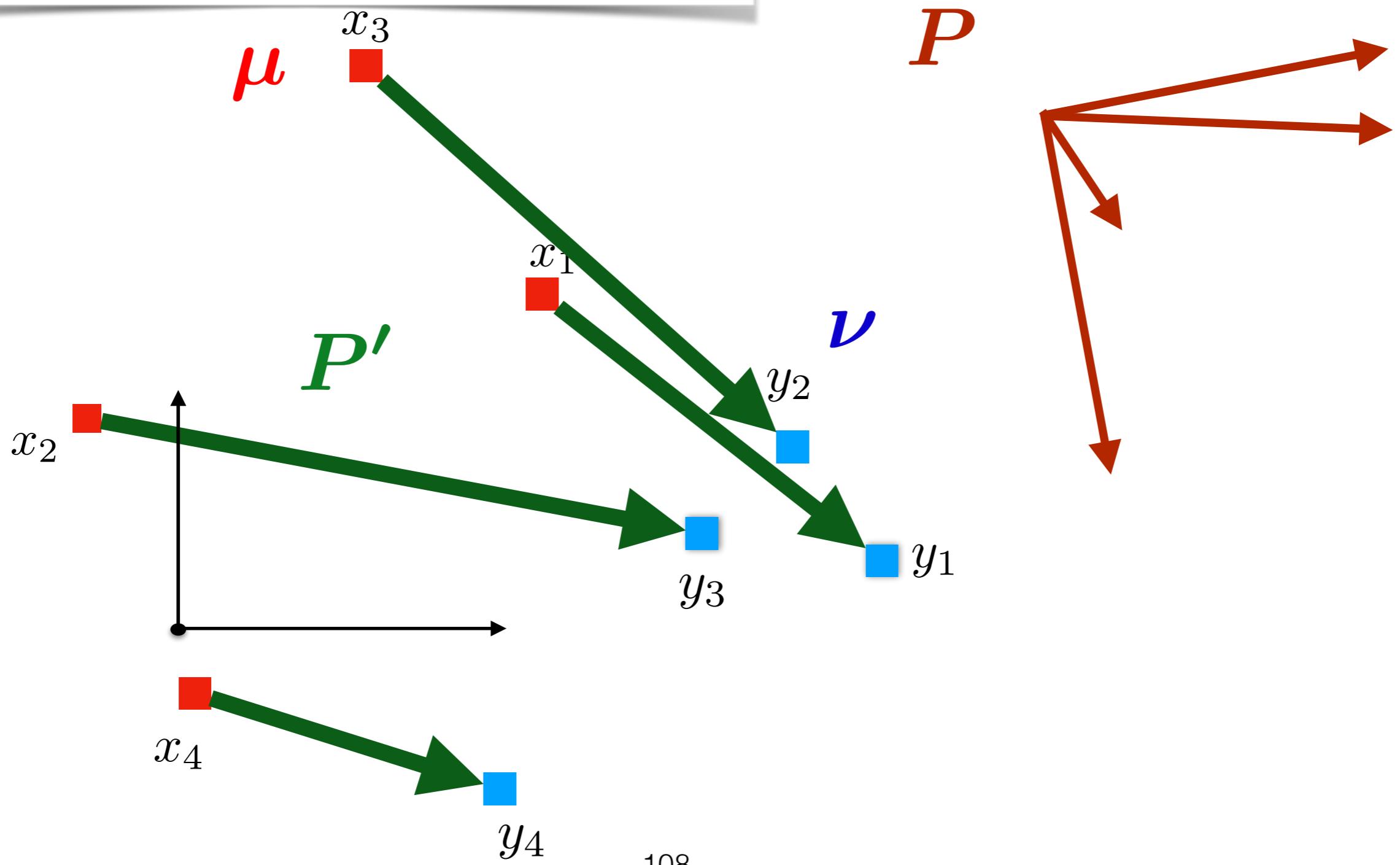
The $V_{\mathbf{P}}$ Moment Matrix

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



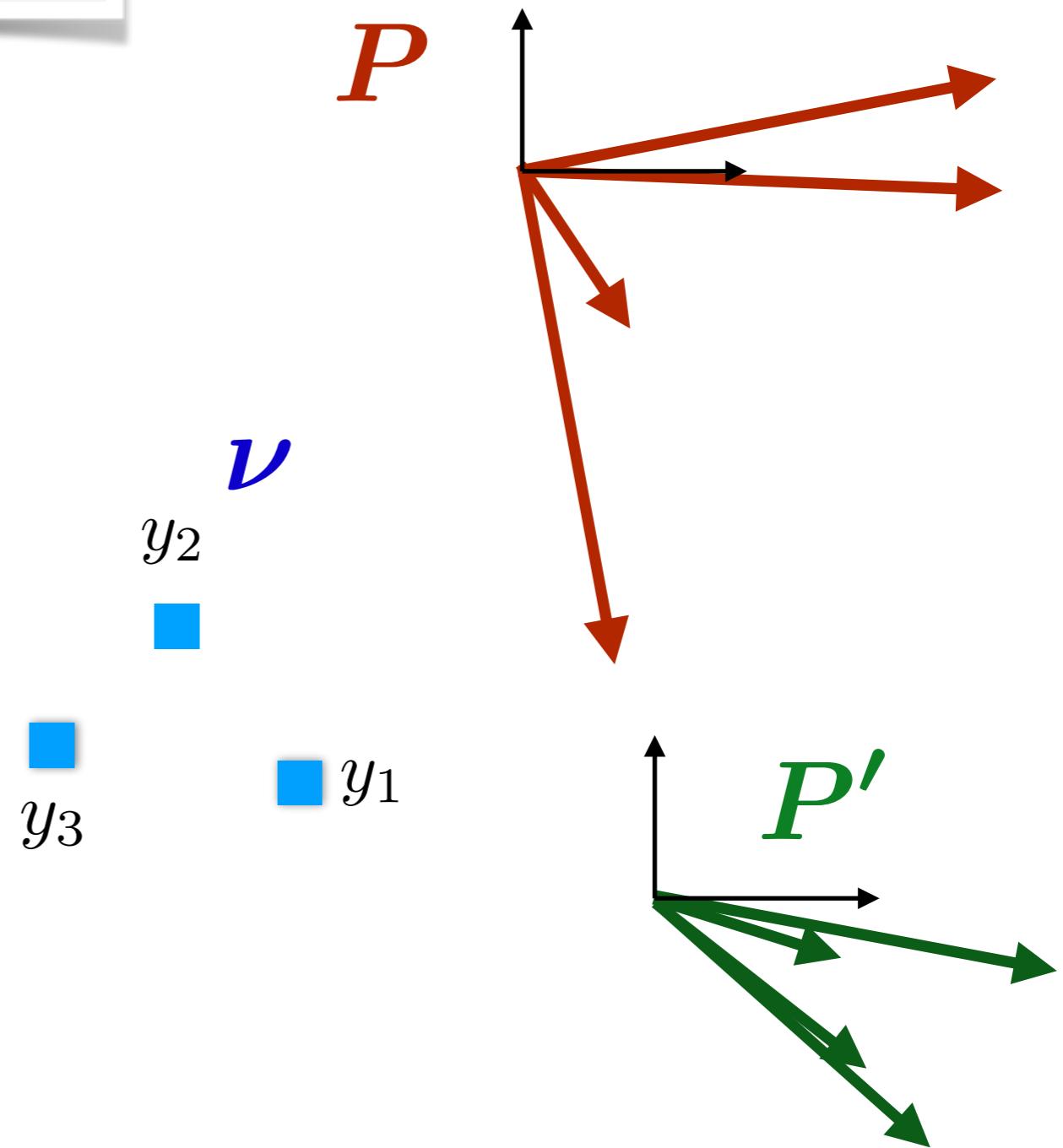
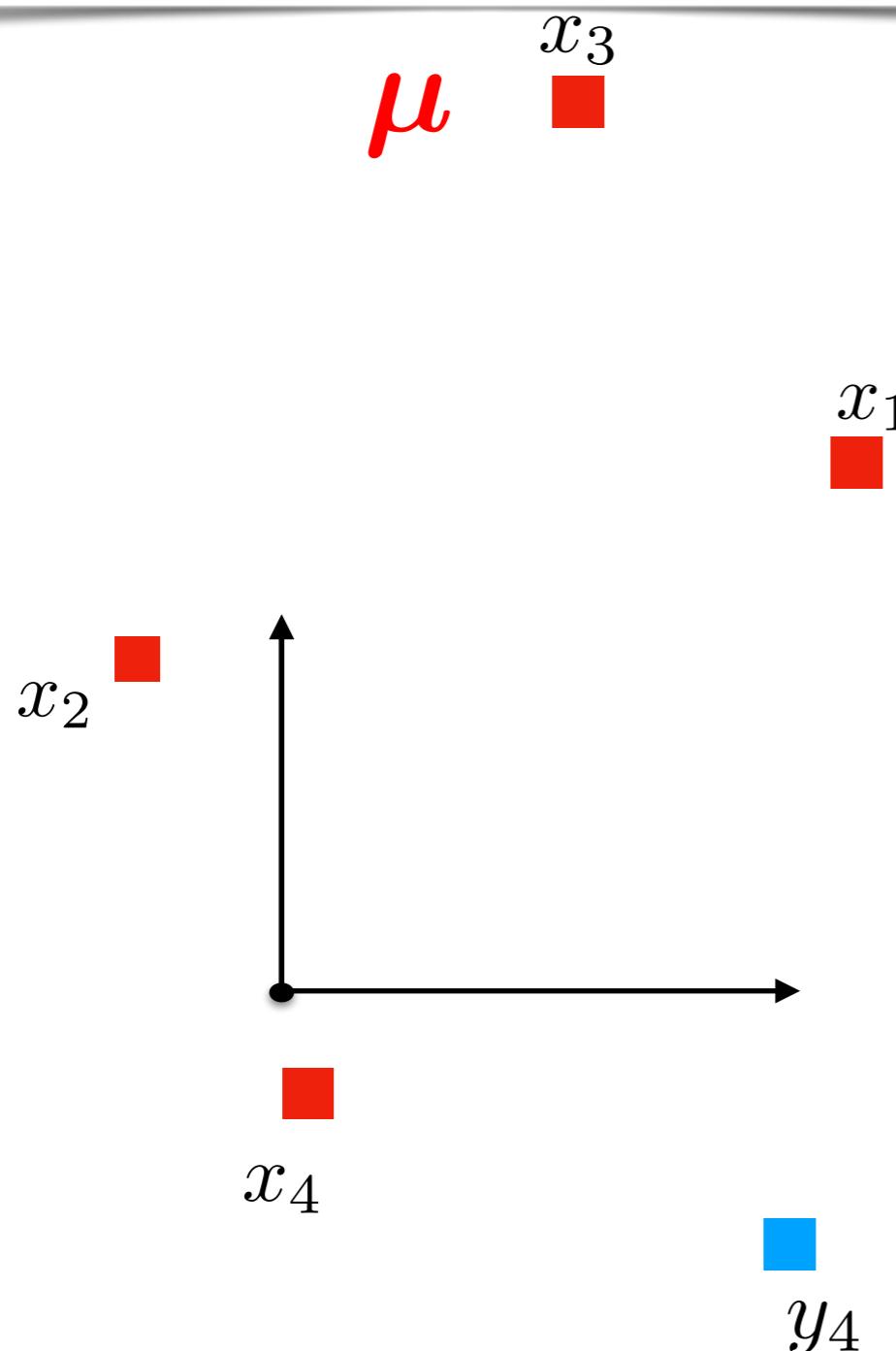
The $V_{\mathbf{P}}$ Moment Matrix

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



The $V_{\mathbf{P}}$ Moment Matrix

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



On k -dim (robust) projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} \langle \mathbf{E} \mathbf{E}^T, V_{\mathbf{P}} \rangle$$

eigenvalue problem

$$\max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} \langle \mathbf{E} \mathbf{E}^T, V_{\mathbf{P}} \rangle = \sum_{i=1}^k \lambda_i(V_{\mathbf{P}})$$

On k -dim (robust) projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} \langle \mathbf{E} \mathbf{E}^T, V_{\mathbf{P}} \rangle$$

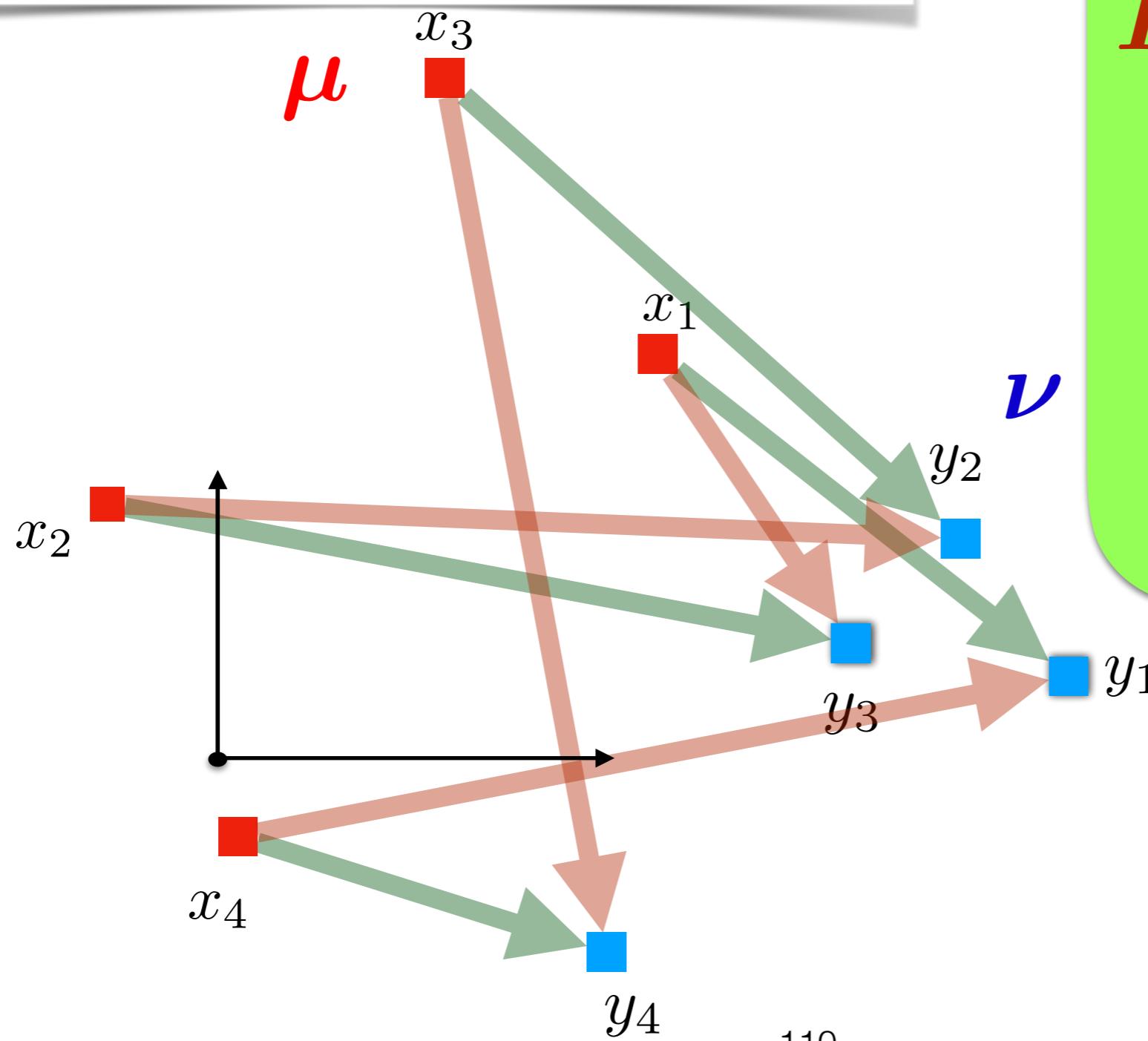
eigenvalue problem

$$\max_{\substack{\mathbf{E} \in \mathbb{R}^{d \times k} \\ \mathbf{E}^T \mathbf{E} = I_k}} \langle \mathbf{E} \mathbf{E}^T, V_{\mathbf{P}} \rangle = \sum_{i=1}^k \lambda_i(V_{\mathbf{P}})$$

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^k \lambda_i(V_{\mathbf{P}})$$

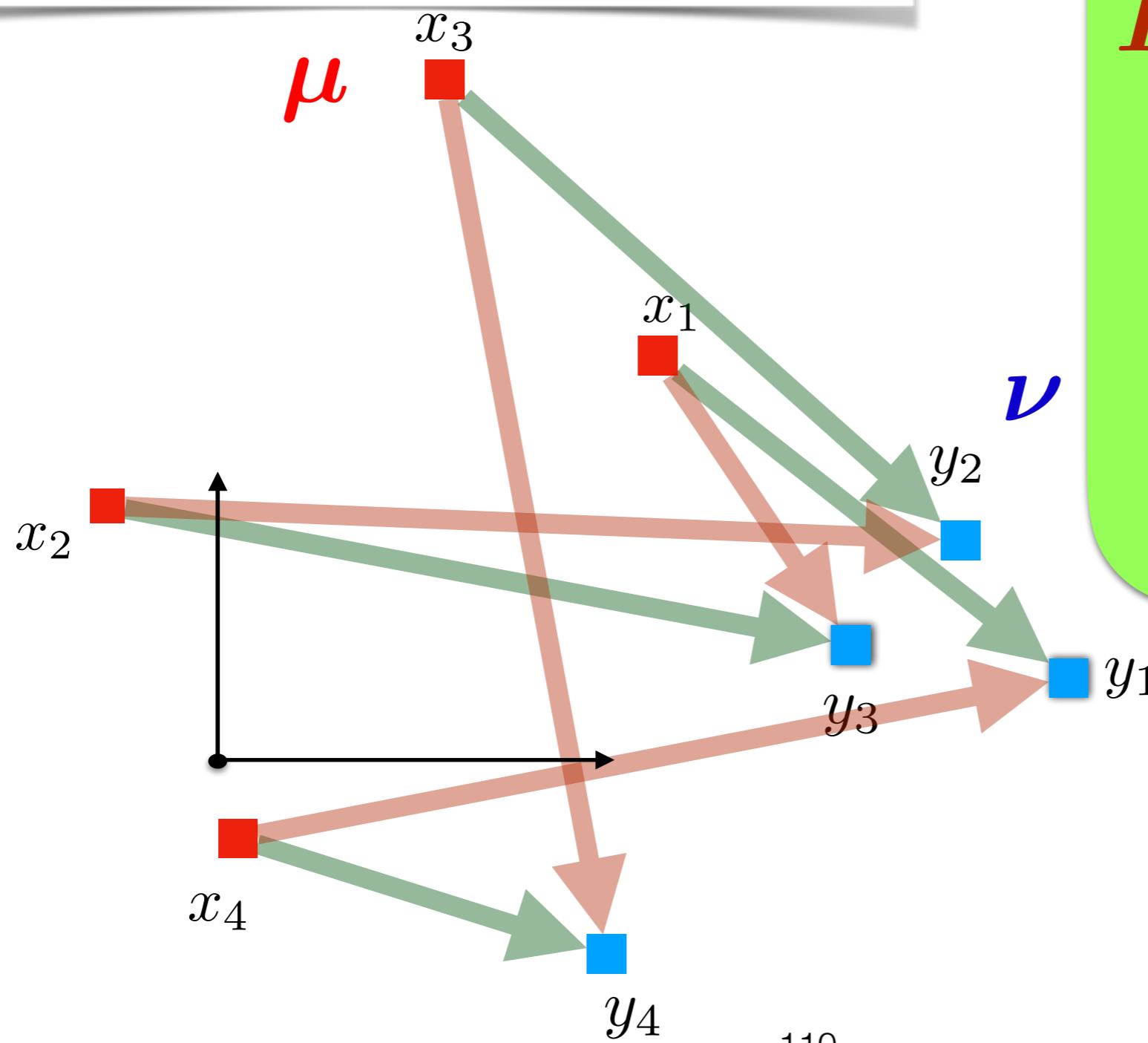
On k -dim (robust) projections

$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



On k -dim (robust) projections

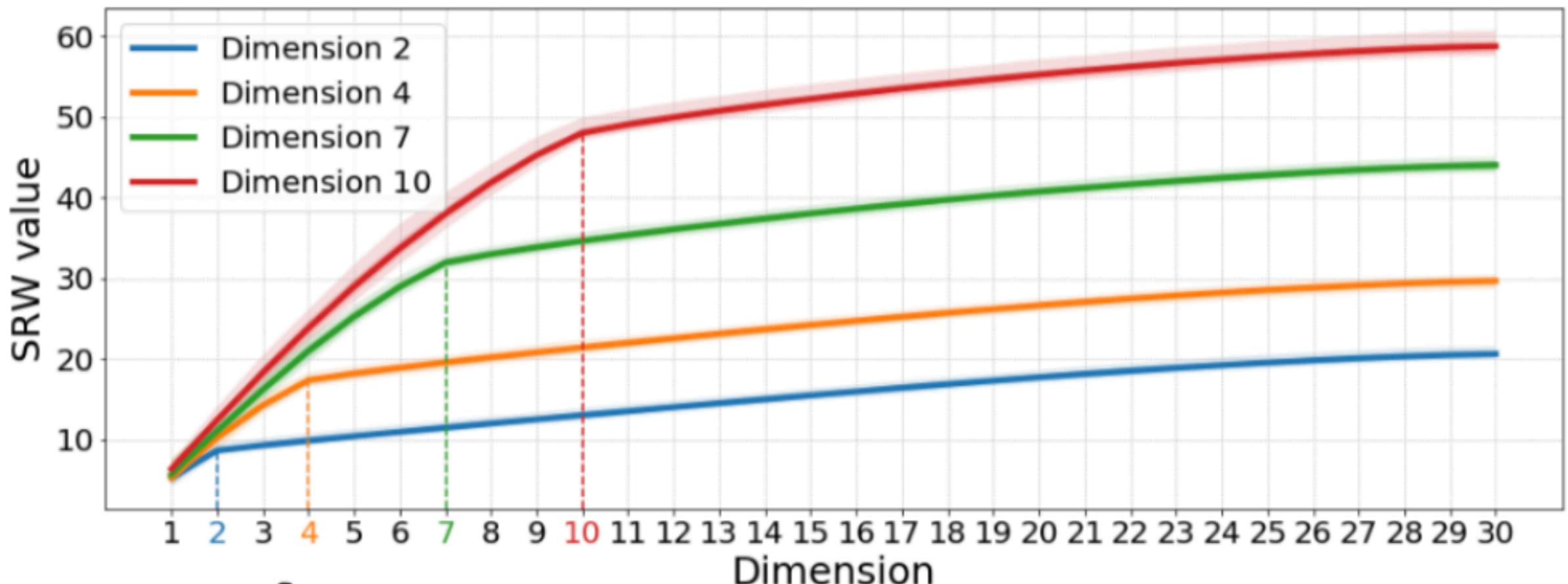
$$V_{\mathbf{P}} := \iint (x - y)(x - y)^T \mathbf{P}(dx, dy)$$



On k -dim robust projections

$$\mathcal{S}_{2,k} = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^k \lambda_i(V_{\mathbf{P}})$$

Solved using FW and eigenvalue decomposition
and entropic regularization to approximate \mathbf{P}



4. Applications

- Wasserstein distances for retrieval
- Wasserstein barycenters
- W for unsupervised learning
- W inverse problems
- W to learn parameters and generative models

The Earth Mover's Distance



The Earth Mover's Distance



The Earth Mover's Distance



[Rubner'98] $\text{dist}(I_1, I_2) = W_1(\mu, \nu)$

The Word Mover's Distance



[Kusner'15]

$$\text{dist}(D_1, D_2) = W_2(\mu, \nu)$$

Recall that our goal is...

Up to 2010: OT solvers
used mostly for retrieval
in databases of histograms

$$W_p(\mu, \nu) = ?$$

$$W_p(\mu, \nu) \leq \dots ?$$

OT is now used as a **loss** or **fidelity** term:

$$\operatorname{argmin}_{\mu \in \mathcal{P}(\Omega)} F(W_p(\mu, \nu_1), W_p(\mu, \nu_2), \dots, \mu) = ?$$

“ ∇_{μ} ” $W_p(\mu, \nu) = ?$

[Jordan Kinderlehrer Otto'98]

[Ambrosio Gigli Savaré'05]

Wassersteinization

[wos-ur-stahyn-ahy-sey-shuh-n]

noun.

**Introduction of optimal transport
into an optimization or learning
problem.**

**cf. least-squarification, L₁ification, deep-netification,
kernelization**

“Wasserstein + Data” Problems

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

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

- [McCann'95] Interpolant

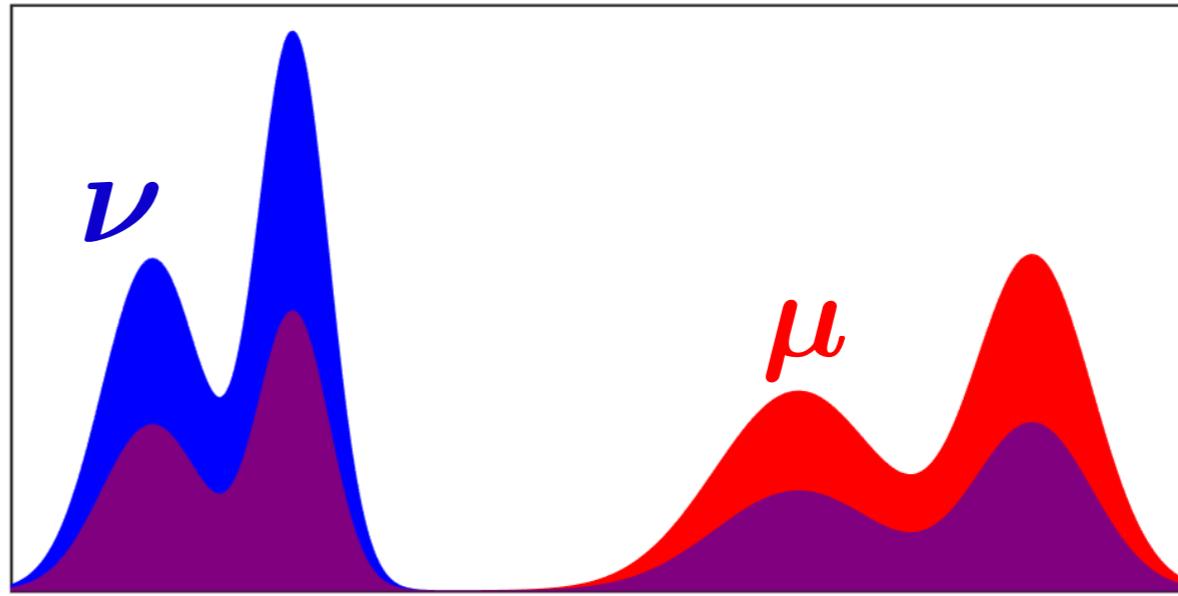
$$\min_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} (1-t)W_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}_1) + tW_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}_2)$$

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

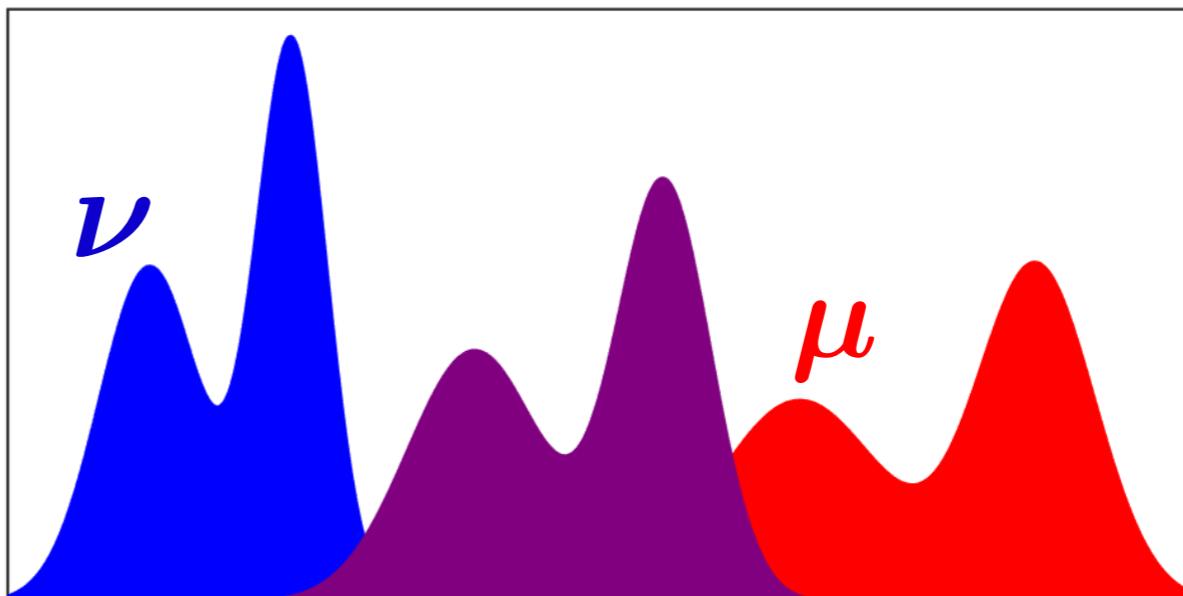
$$\mu_{t+1} = \operatorname*{argmin}_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} J(\boldsymbol{\mu}) + \lambda_t W_p^p(\boldsymbol{\mu}, \mu_t)$$

Averaging Measures

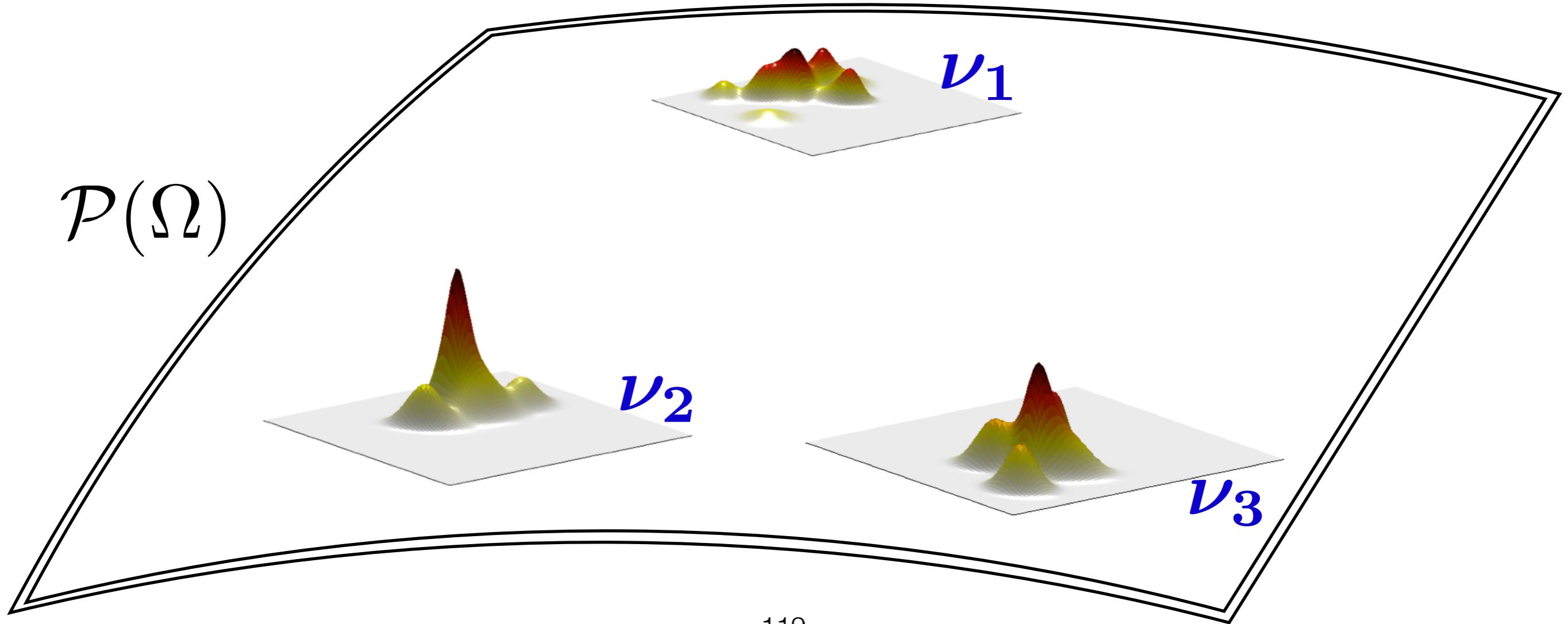
L_2 average



W average



Barycenter for Measures?

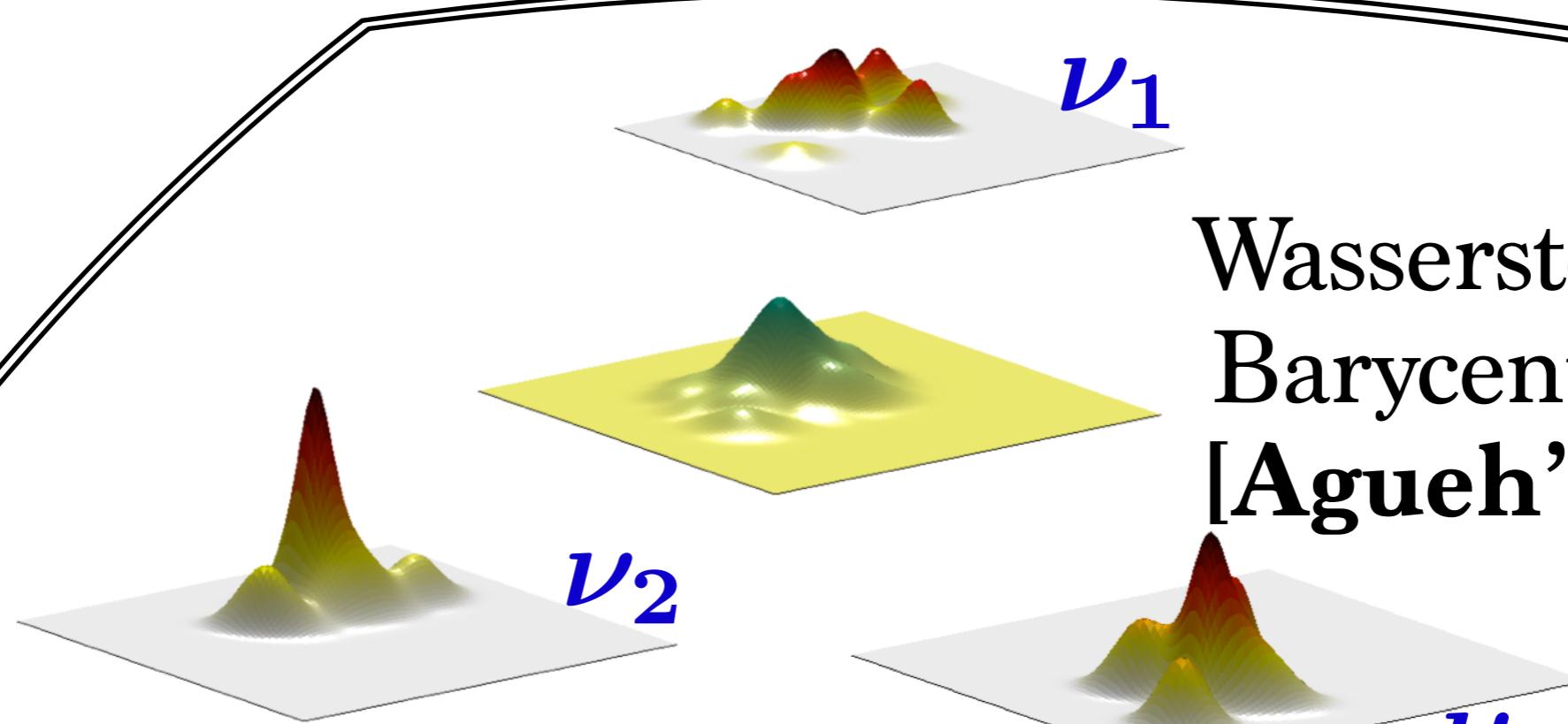


Barycenter for Measures?

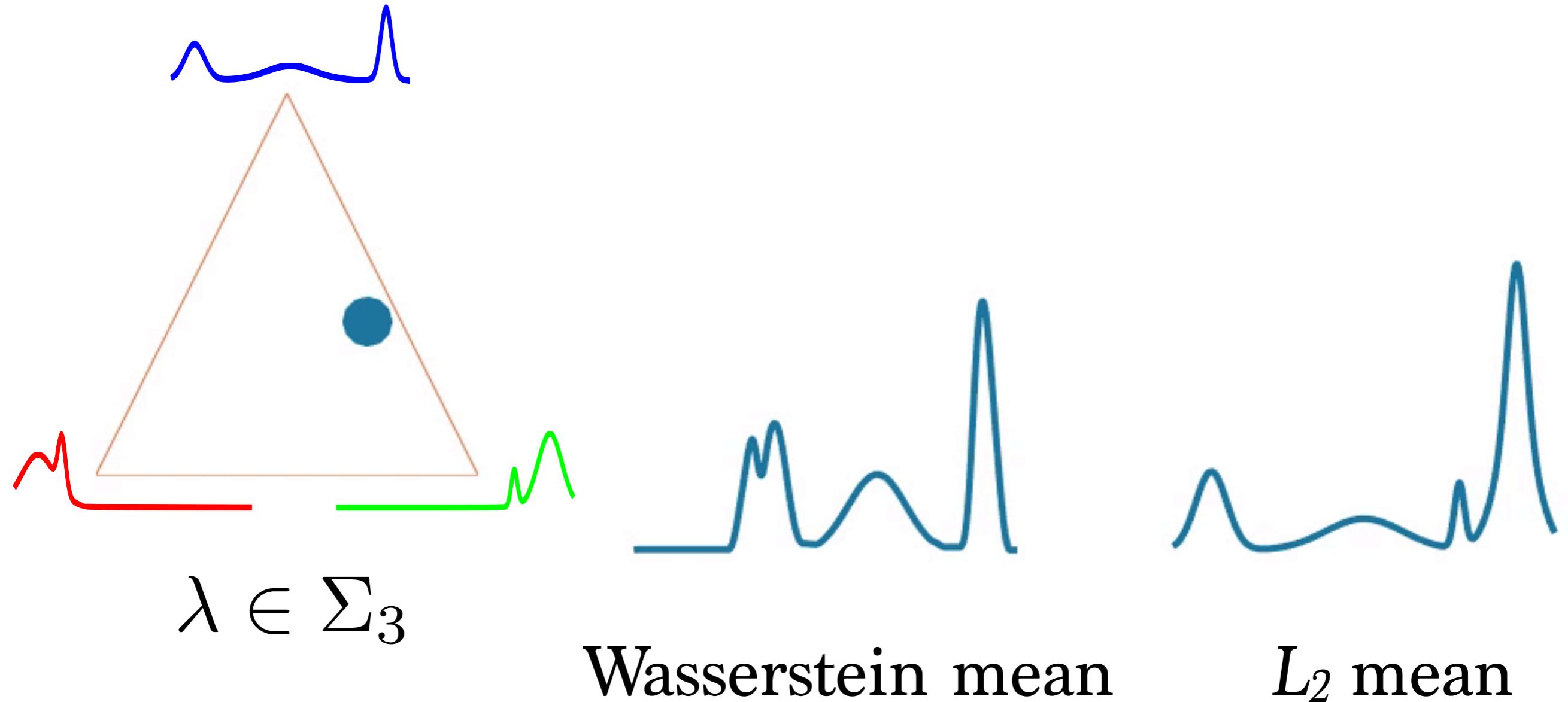
$$\min_{\mu \in \mathcal{P}(\Omega)} \sum_{i=1}^N \lambda_i W_p^p(\mu, \nu_i)$$

$\mathcal{P}(\Omega)$

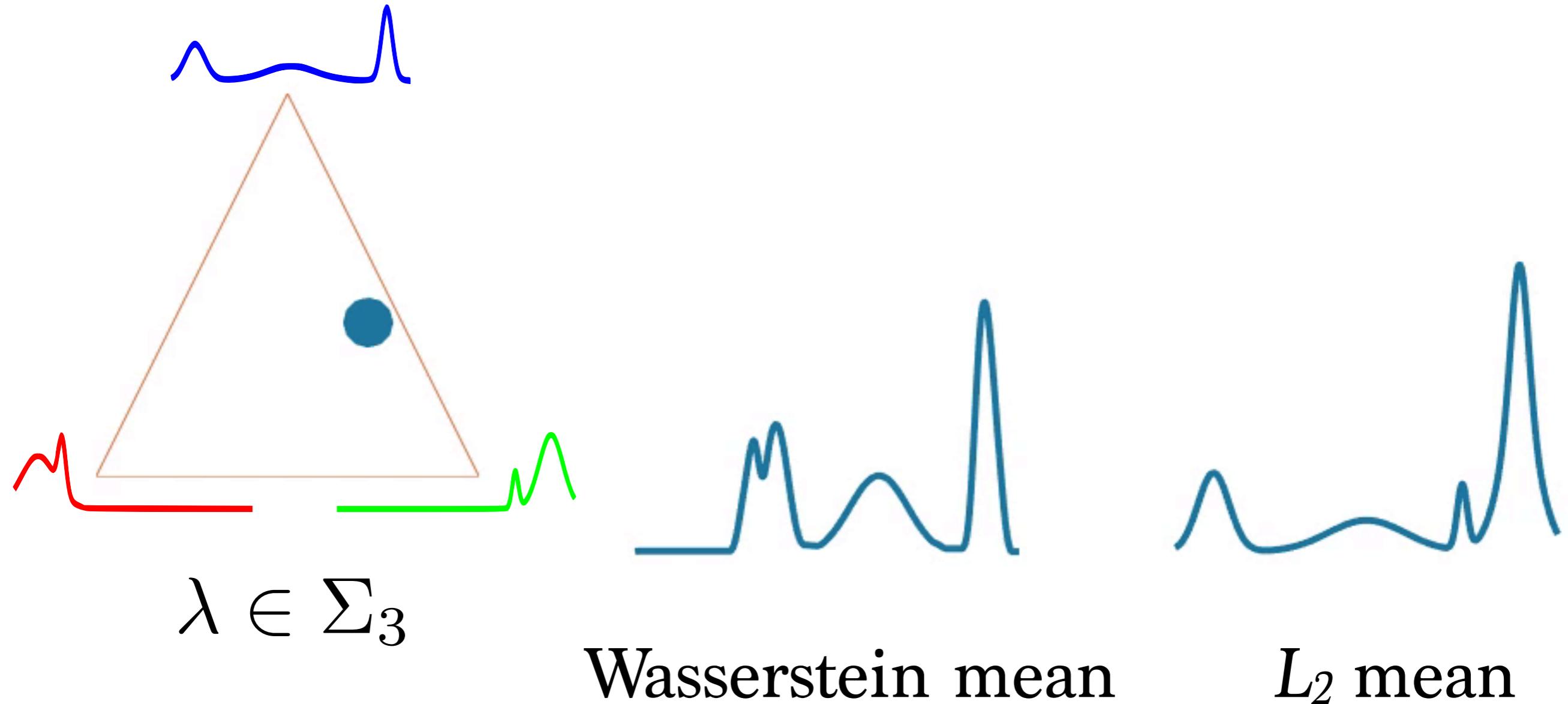
Wasserstein
Barycenter
[Aguech'11]



Barycenter for Measures?

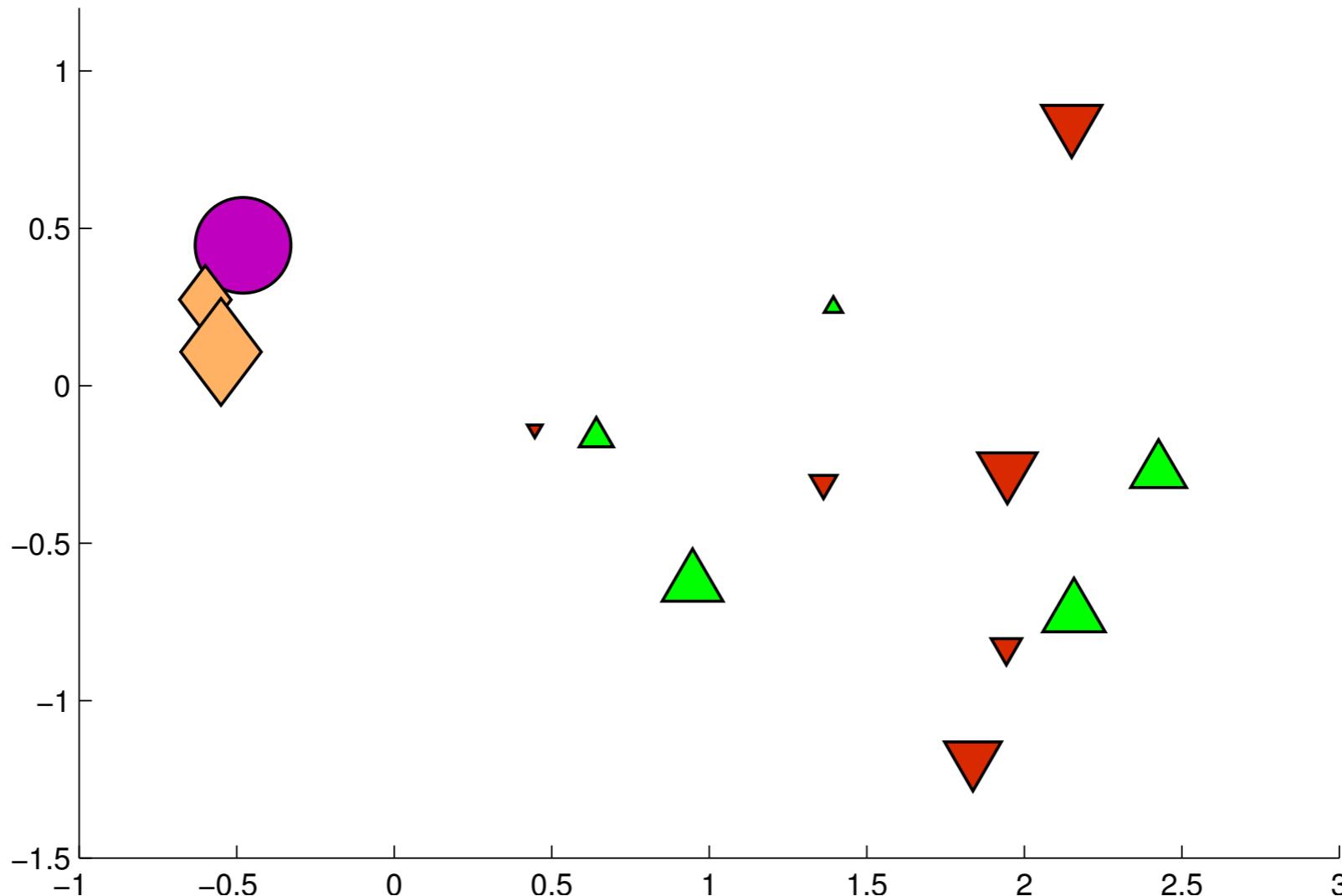


Barycenter for Measures?



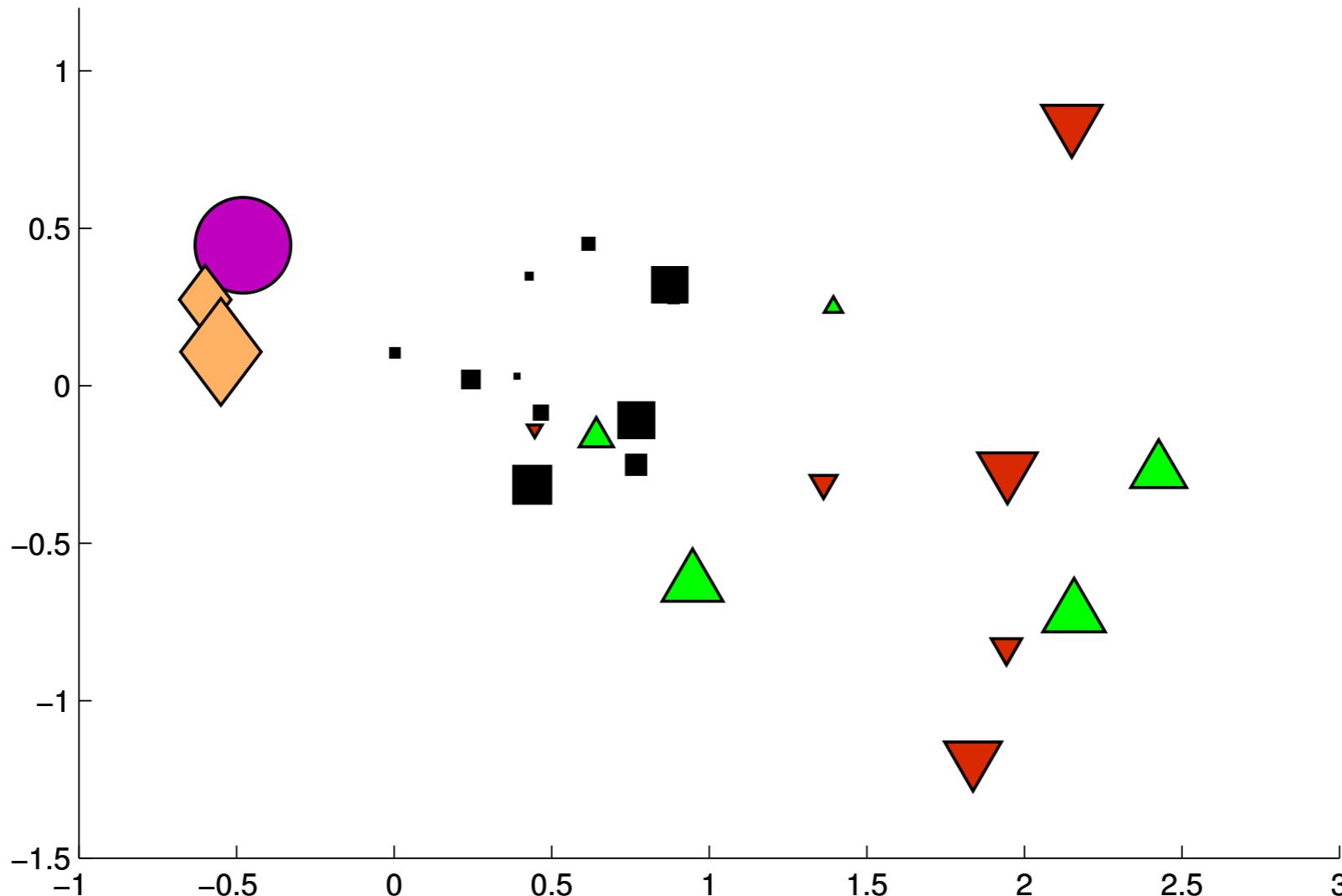
Multimarginal Formulation

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



Multimarginal Formulation

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



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

Averaging Histograms is a LP

When Ω is a finite metric space defined by M .

$$\min_{\mathbf{a} \in \Sigma_n} \sum_i \lambda_i W_M(\mathbf{a}, \mathbf{b}_i)$$

Averaging Histograms is a LP

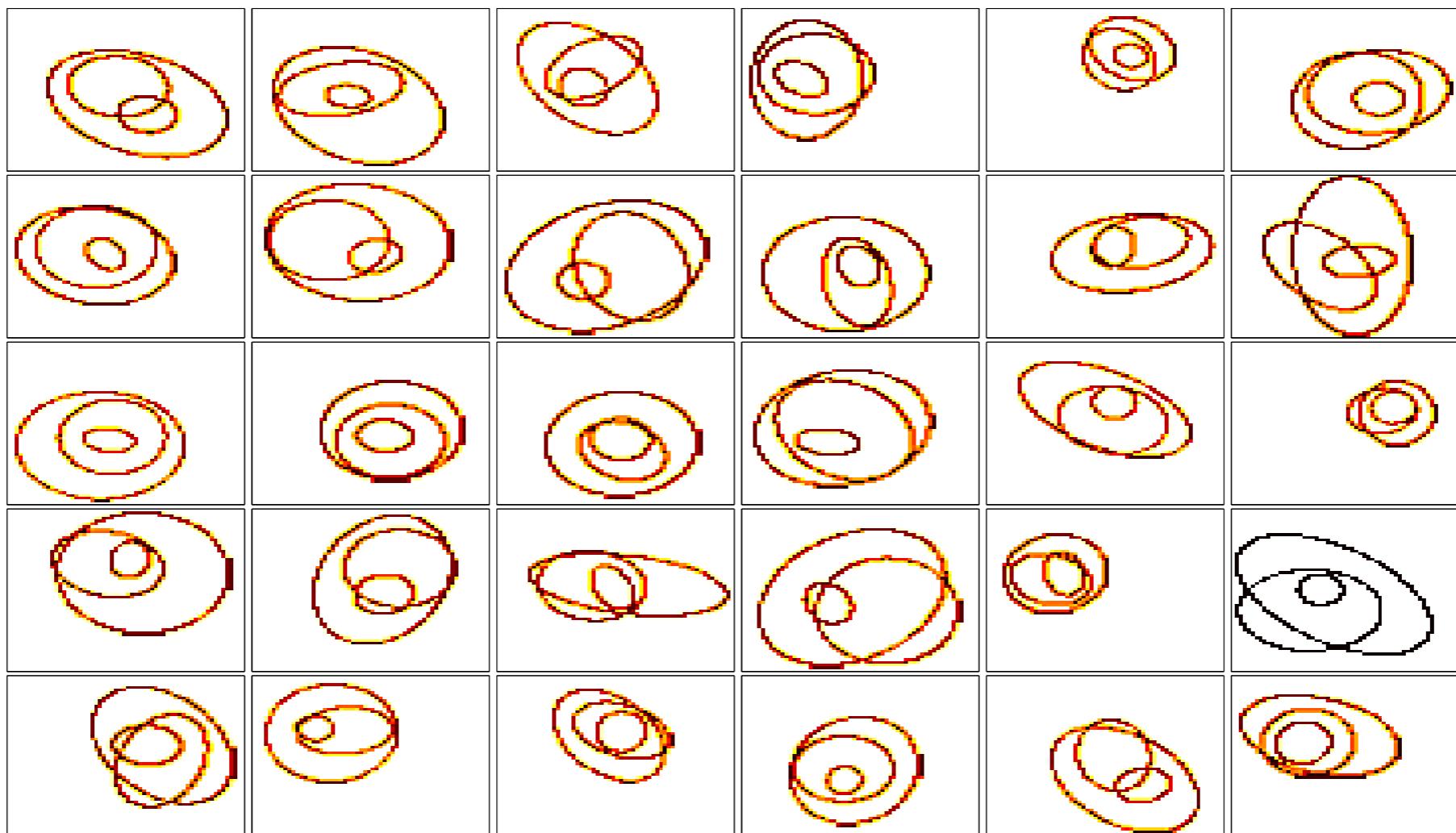
When Ω is a finite metric space defined by M .

$$\begin{aligned} & \min_{P_1, \dots, P_N, \mathbf{a}} \sum_{i=1}^N \lambda_i \langle \mathbf{P}_i, M \rangle \\ \text{s.t. } & \mathbf{P}_i^T \mathbf{1}_n = \mathbf{b}_i, \forall i \leq N, \\ & \mathbf{P}_1 \mathbf{1}_n = \dots = \mathbf{P}_N \mathbf{1}_d = \mathbf{a}. \end{aligned}$$

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

Primal Descent on Regularized W

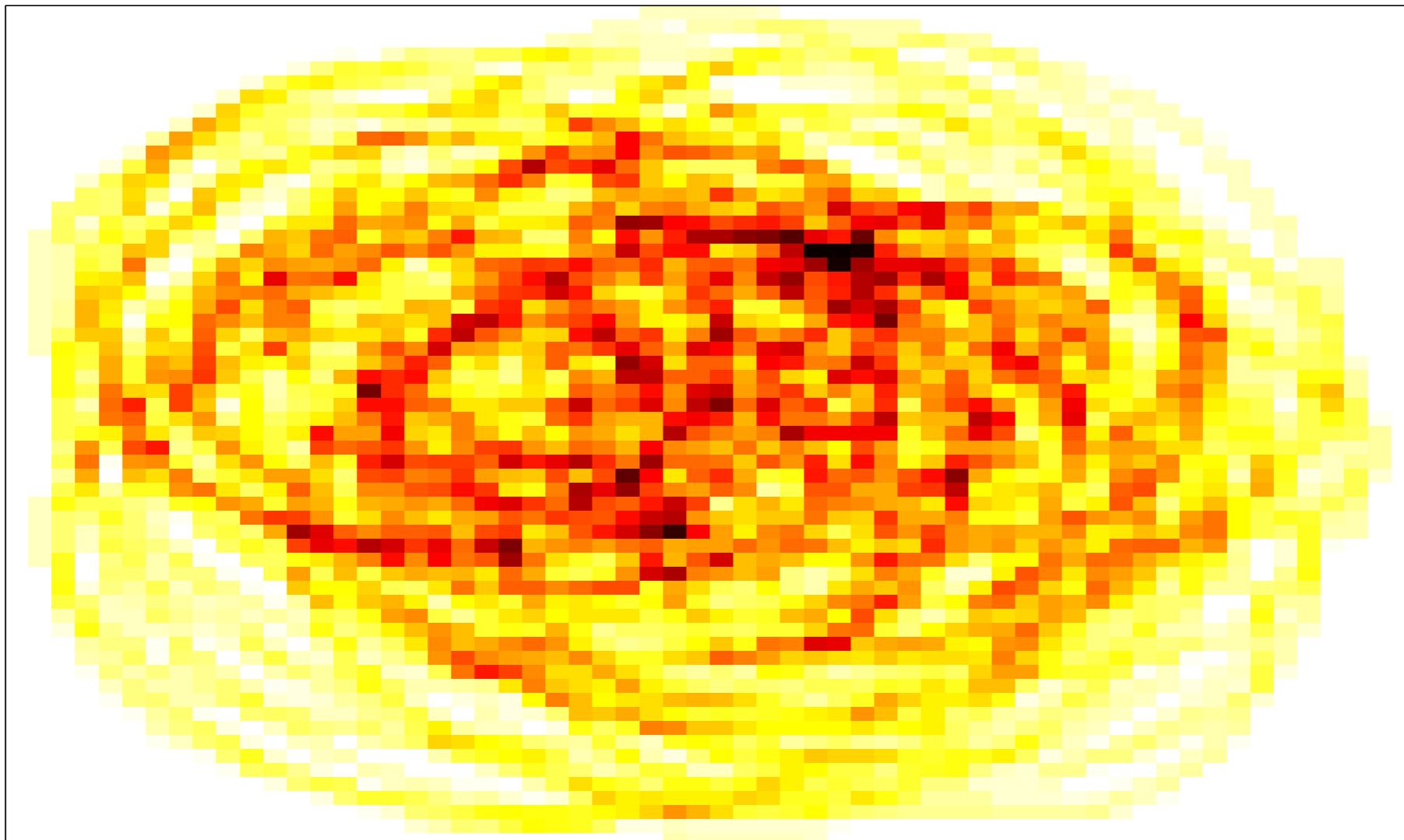
$$\min_{\mathbf{a} \in \Sigma_{h \times h}} \sum_{i=1}^N \lambda_i W_{\gamma}(\mathbf{a}, \mathbf{b}_i)$$



[Cuturi'14]

Primal Descent on Regularized W

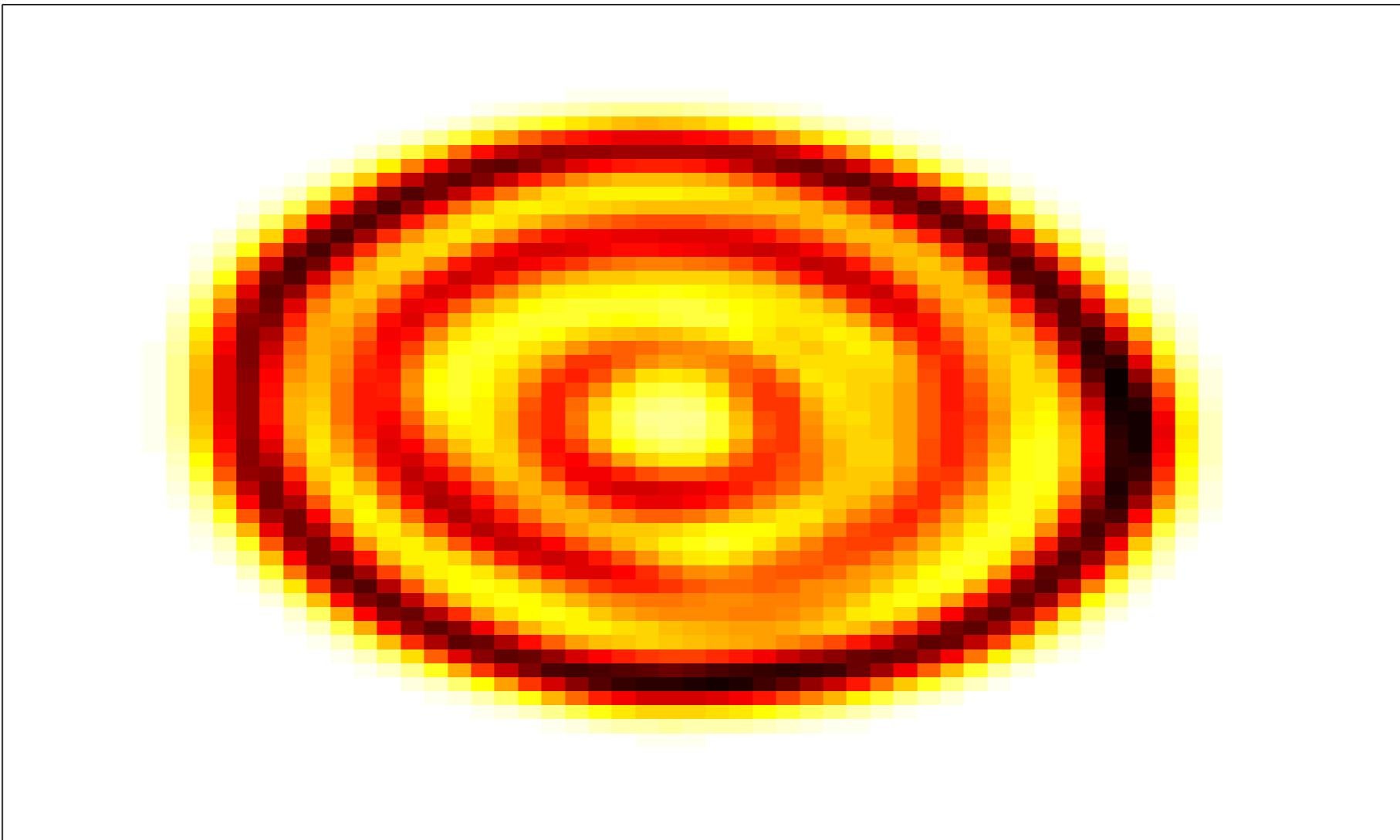
$$\min_{\mathbf{a} \in \Sigma_{h \times h}} \sum_{i=1}^N \lambda_i W_{\gamma}(\mathbf{a}, \mathbf{b}_i)$$



[Cuturi'14]

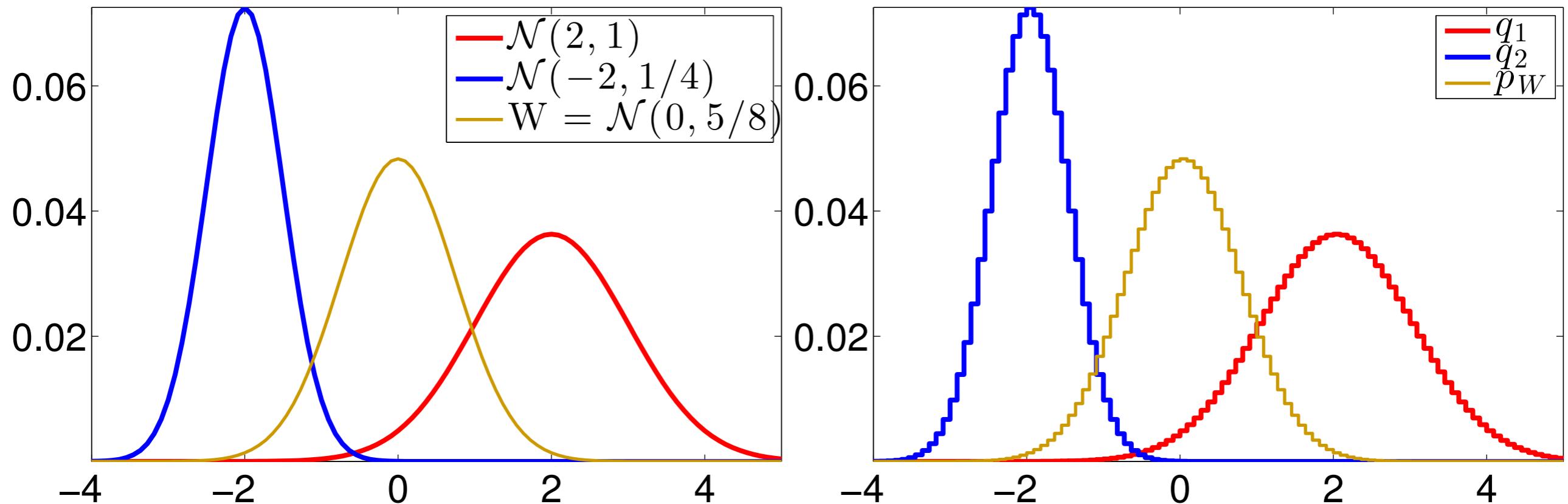
Primal Descent on Regularized W

$$\min_{\mathbf{a} \in \Sigma_{h \times h}} \sum_{i=1}^N \lambda_i W_{\gamma}(\mathbf{a}, \mathbf{b}_i)$$

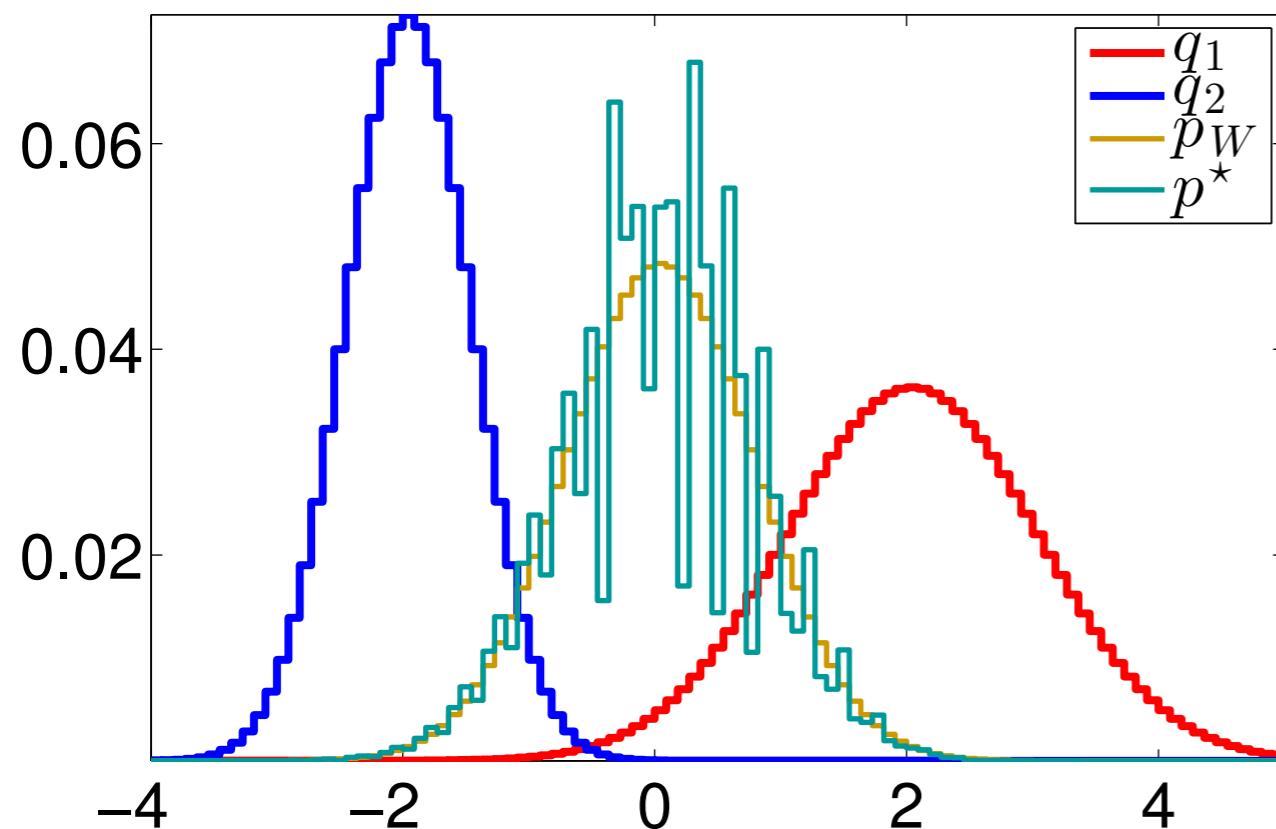


[Cuturi'14]

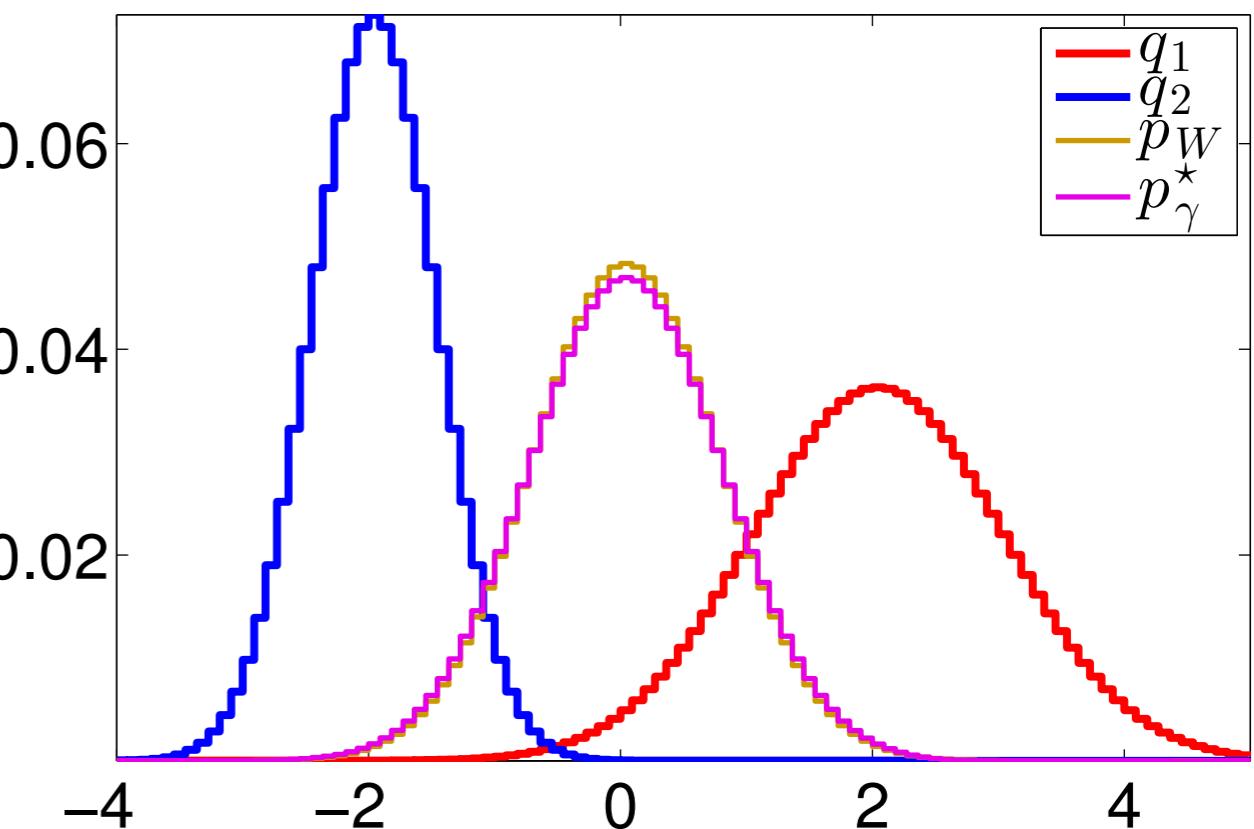
On Regularizing or Not



On Regularizing or Not

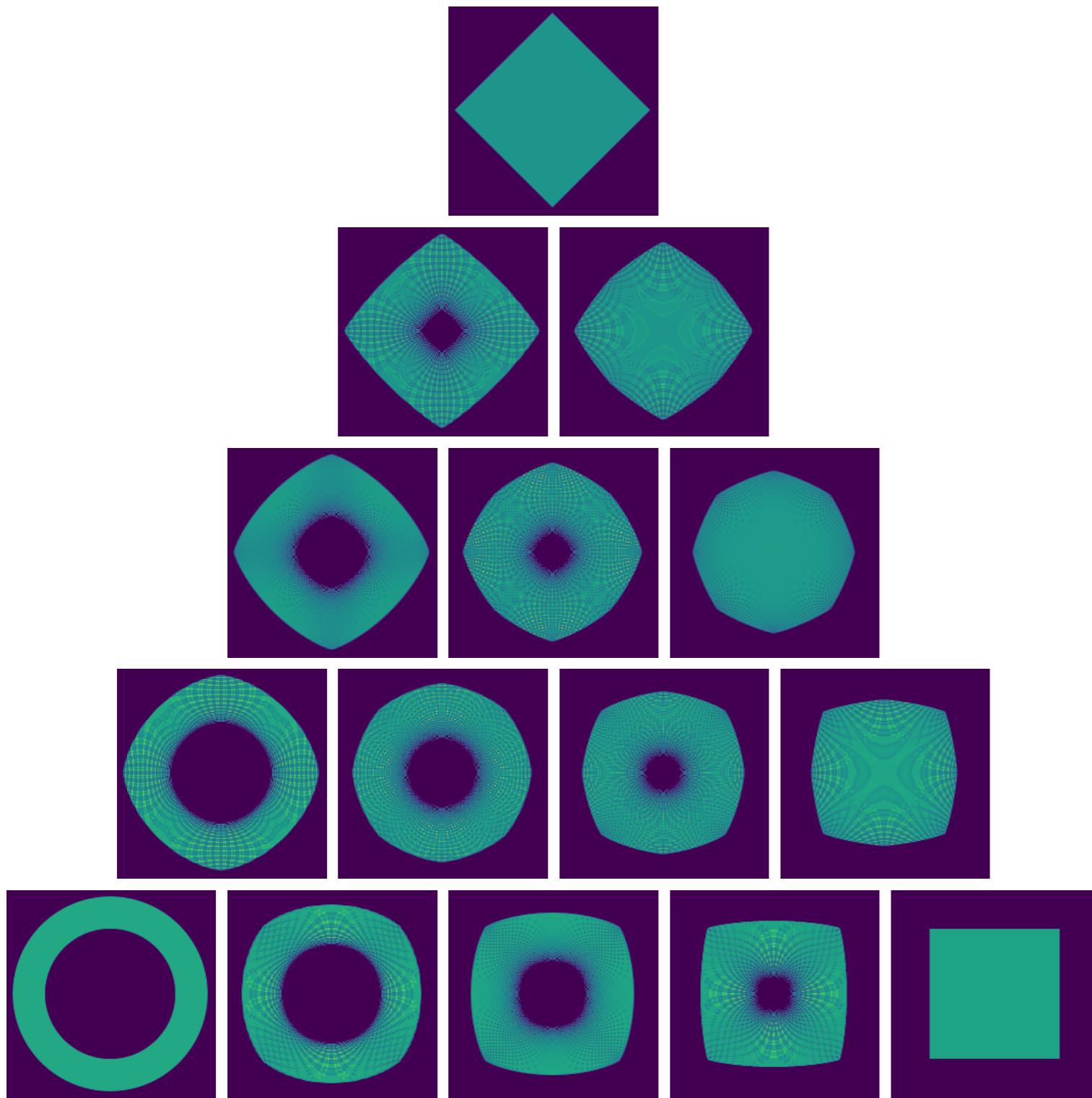


True barycenter



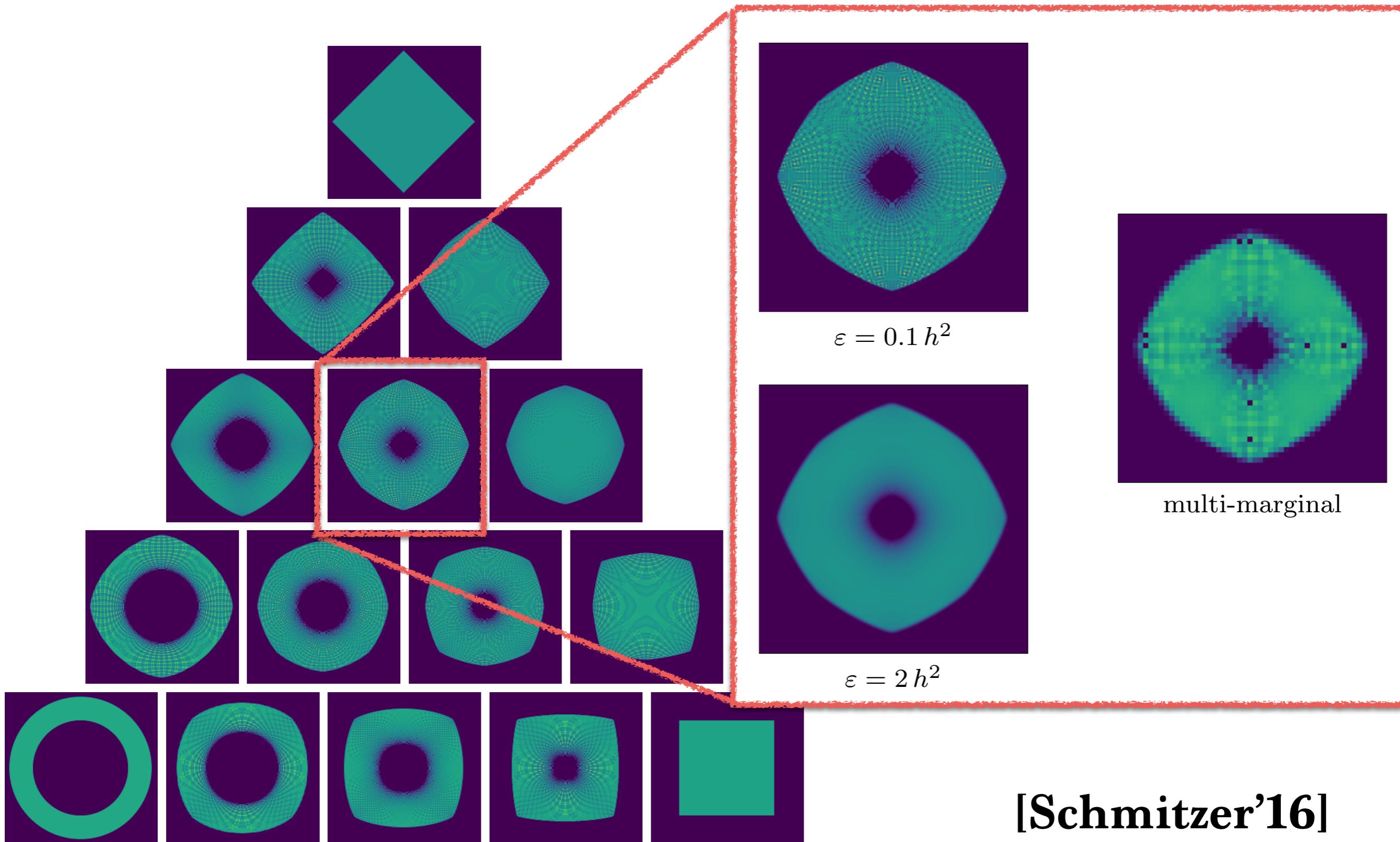
*Barycenter using
regularized OT*

On Regularizing or Not

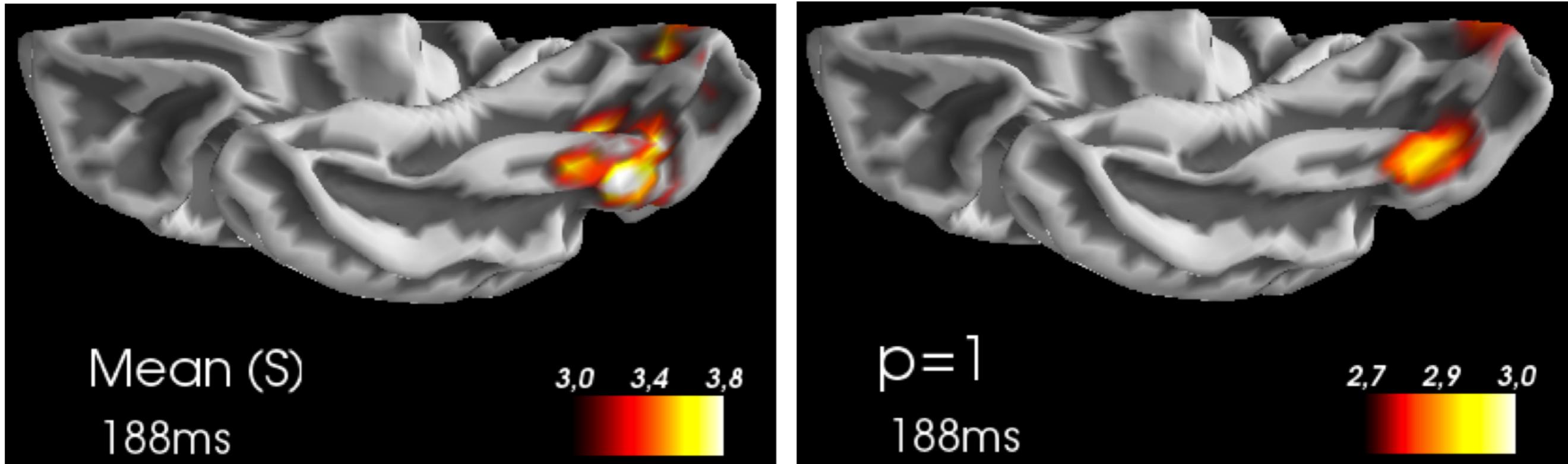


[Schmitzer'16]

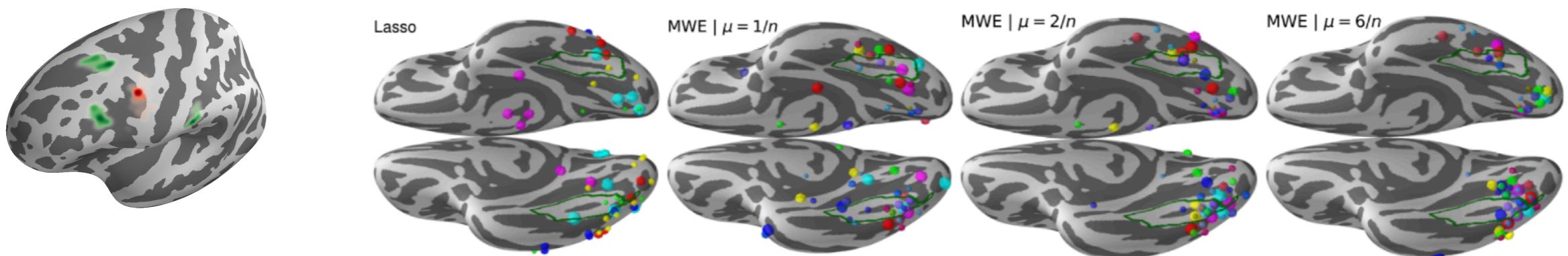
On Regularizing or Not



Applications: Brain Imaging



[Gramfort+'14][Janati'+19a,b]



Inverse Wasserstein Problems

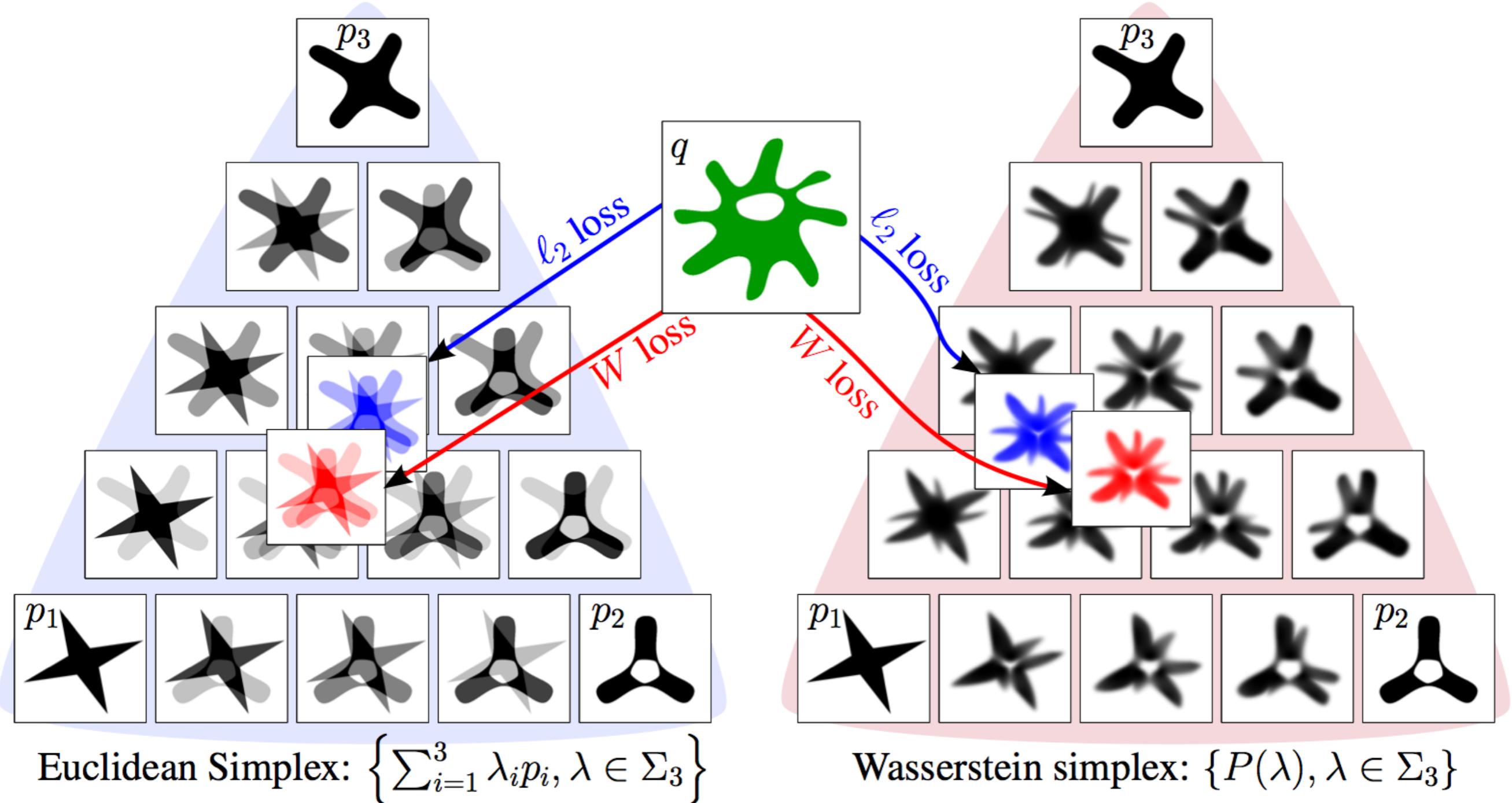
- consider Barycenter operator:

$$\mathbf{b}(\lambda) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\mathbf{a}} \sum_{i=1}^N \lambda_i W_\gamma(\mathbf{a}, \mathbf{b}_i)$$

- address now **Wasserstein inverse problems**:

Given \mathbf{a} , find $\operatorname*{argmin}_{\lambda \in \Sigma_N} \mathcal{E}(\lambda) \stackrel{\text{def}}{=} \text{Loss}(\mathbf{a}, \mathbf{b}(\lambda))$

Wasserstein Inverse Problems



Barycenters = Fixed Points

Prop. [BCCNP'15] Consider $\mathbf{B} \in \Sigma_d^N$ and let $\mathbf{U}_0 = \mathbf{1}_{d \times N}$, and then for $l \geq 0$:

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

Using Truncated Barycenters

- instead of using the exact barycenter

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

- use instead the L-iterate barycenter

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

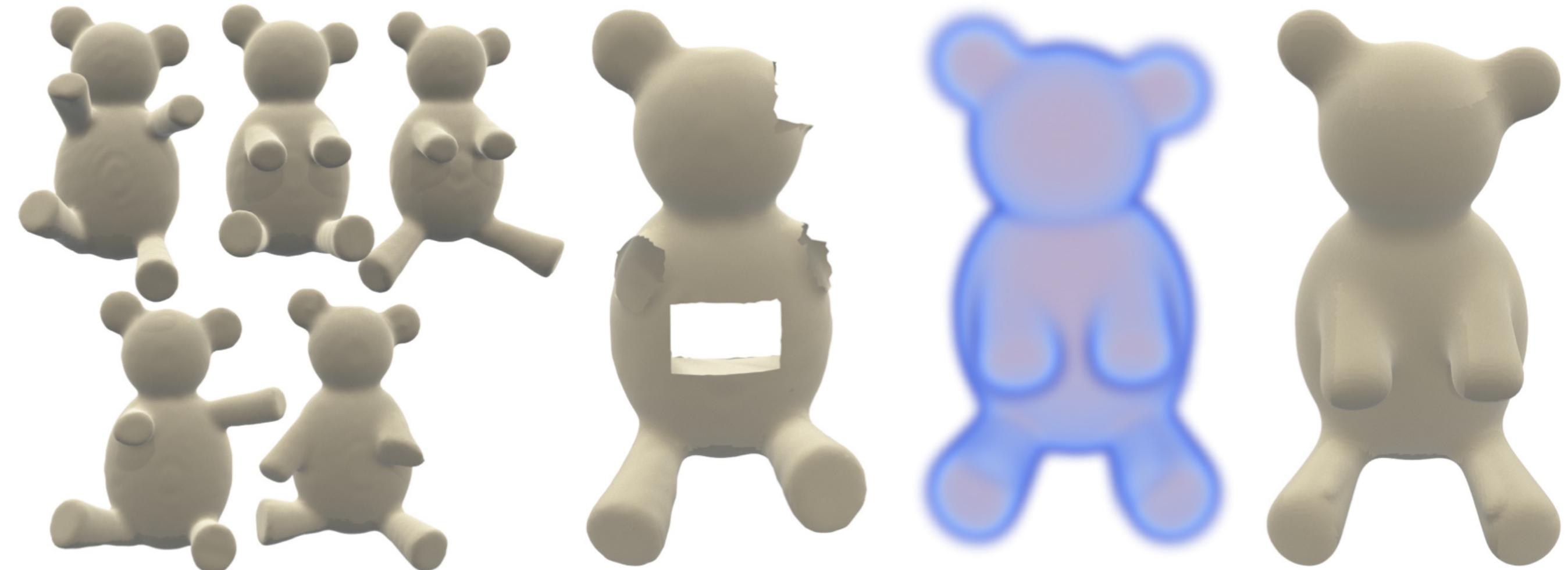
- Different using **the chain rule.**

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

Gradient / Barycenter Computation

```
function SINKHORN-DIFFERENTIATE( $(p_s)_{s=1}^S, q, \lambda$ )
     $\forall s, b_s^{(0)} \leftarrow \mathbf{1}$ 
     $(w, r) \leftarrow (0^S, 0^{S \times N})$ 
    for  $\ell = 1, 2, \dots, L$  // Sinkhorn loop
         $\forall s, \varphi_s^{(\ell)} \leftarrow K^\top \frac{p_s}{Kb_s^{(\ell-1)}}$ 
         $p \leftarrow \prod_s \left( \varphi_s^{(\ell)} \right)^{\lambda_s}$ 
         $\forall s, b_s^{(\ell)} \leftarrow \frac{p}{\varphi_s^{(\ell)}}$ 
         $g \leftarrow \nabla \mathcal{L}(p, q) \odot p$ 
    for  $\ell = L, L-1, \dots, 1$  // Reverse loop
         $\forall s, w_s \leftarrow w_s + \langle \log \varphi_s^{(\ell)}, g \rangle$ 
         $\forall s, r_s \leftarrow -K^\top \left( K \left( \frac{\lambda_s g - r_s}{\varphi_s^{(\ell)}} \right) \odot \frac{p_s}{(Kb_s^{(\ell-1)})^2} \right) \odot b_s^{(\ell-1)}$ 
         $g \leftarrow \sum_s r_s$ 
    return  $P^{(L)}(\lambda) \leftarrow p, \nabla \mathcal{E}_L(\lambda) \leftarrow w$ 
```

Application: Volume Reconstruction



Shape database
 (p_1, \dots, p_5)

Input shape q

Projection
 $P(\lambda)$

Iso-surface

[Bonneel'16]

Application: Color Grading



Application: Color Grading



$$\lambda_0 = 0.03$$



$$\lambda_1 = 0.12$$



$$\lambda_2 = 0.40$$

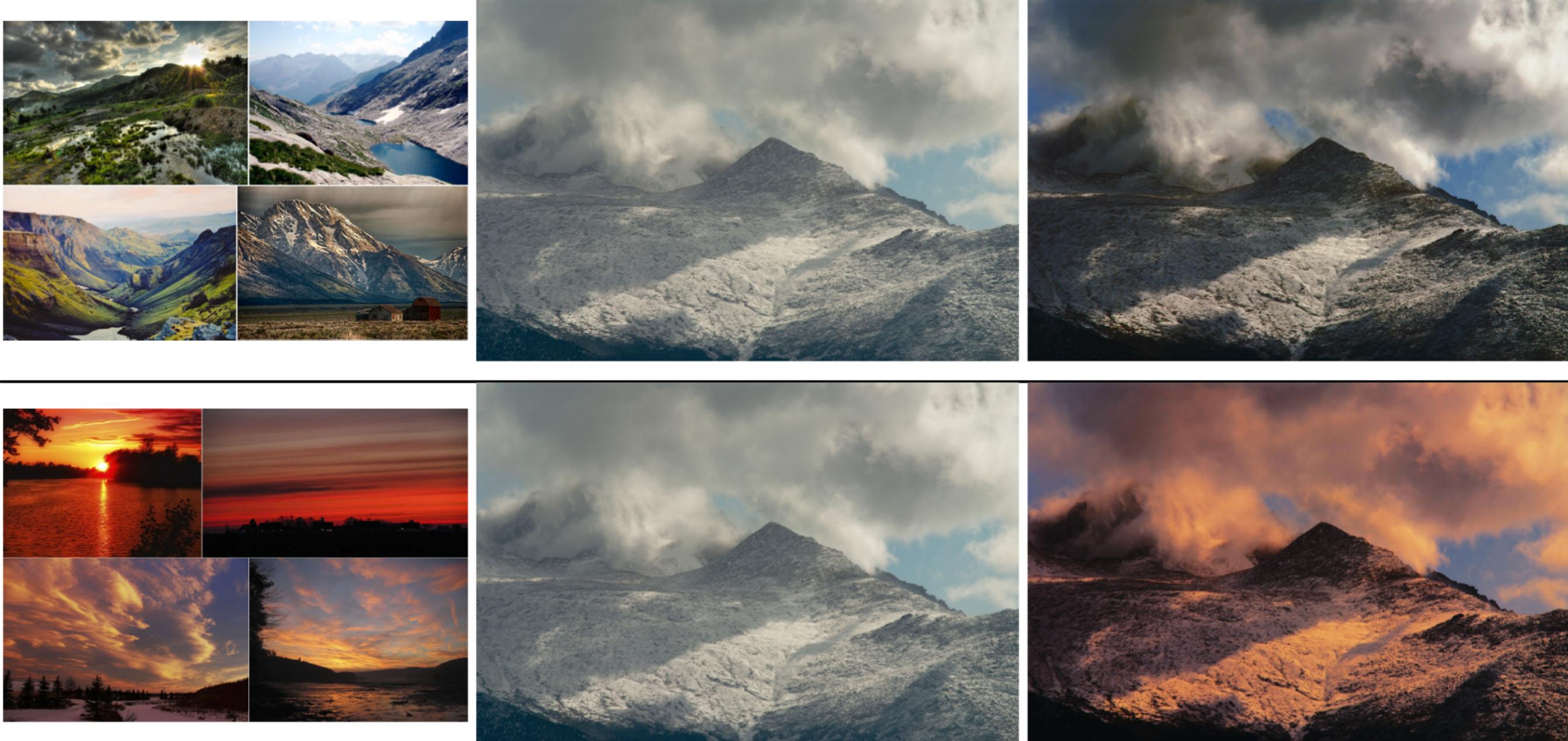


$$\lambda_3 = 0.43$$

Application: Color Grading



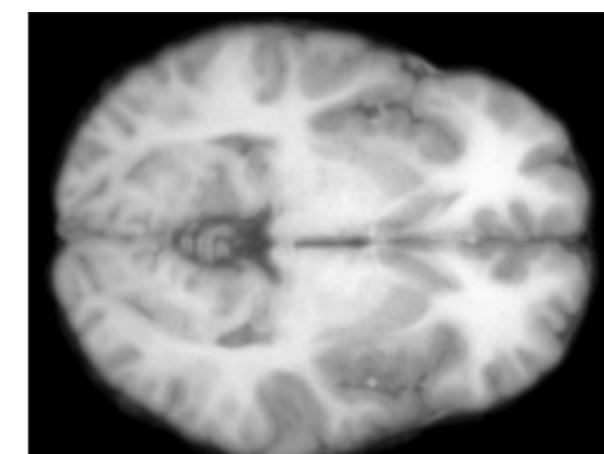
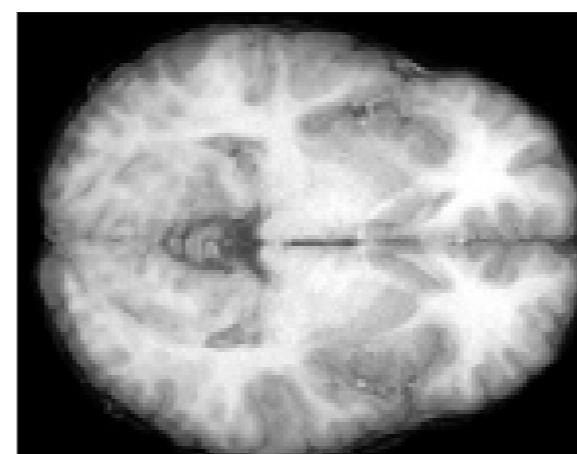
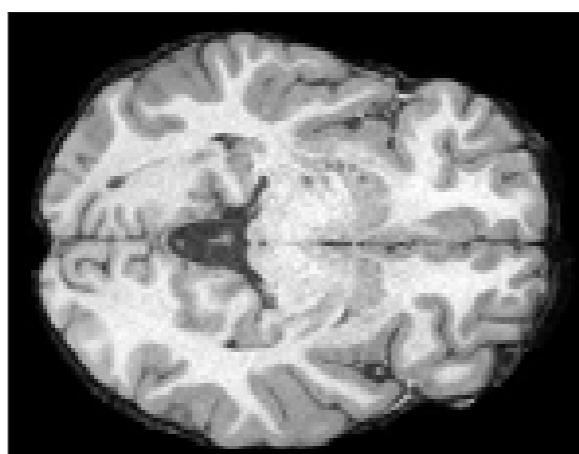
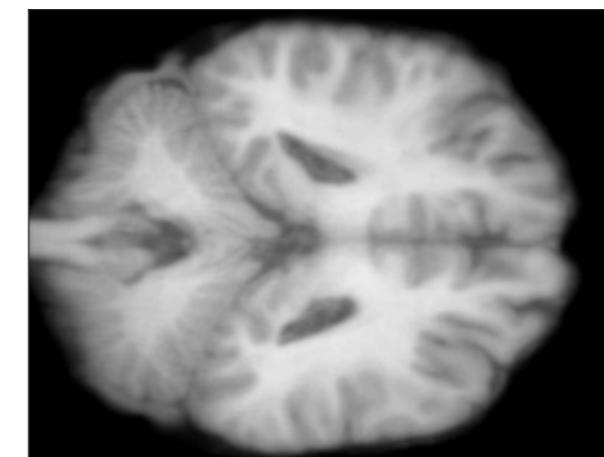
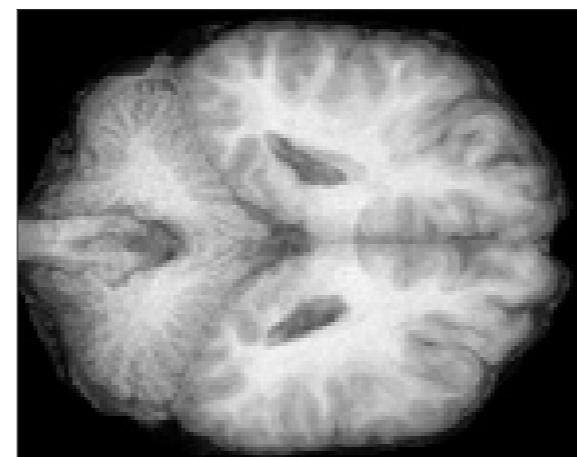
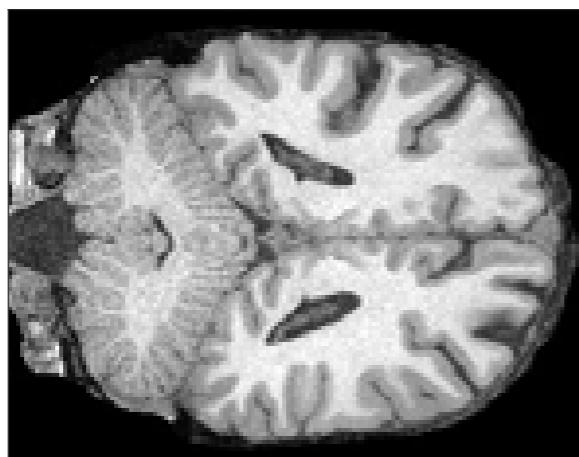
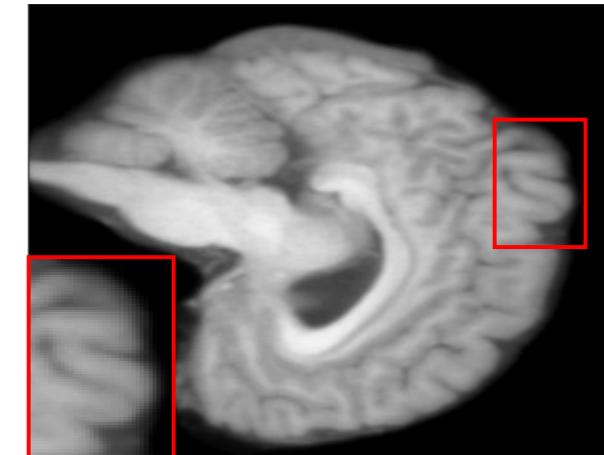
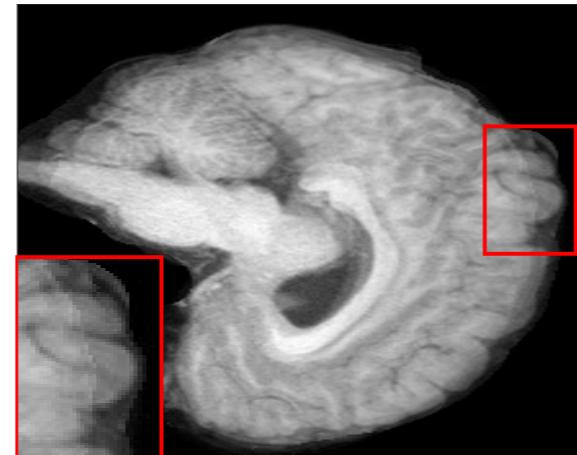
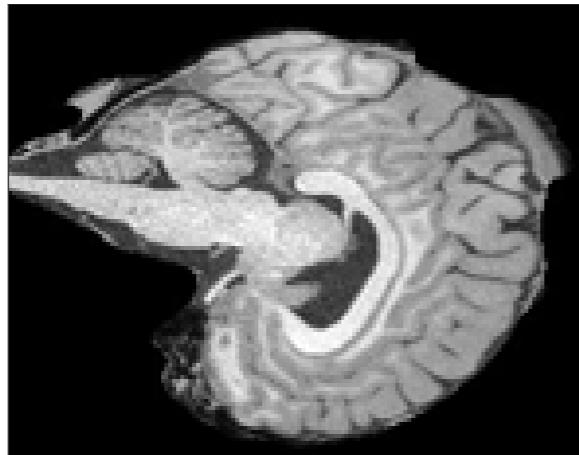
Application: Color Grading



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

[BPC'16]

Application: Brain Mapping

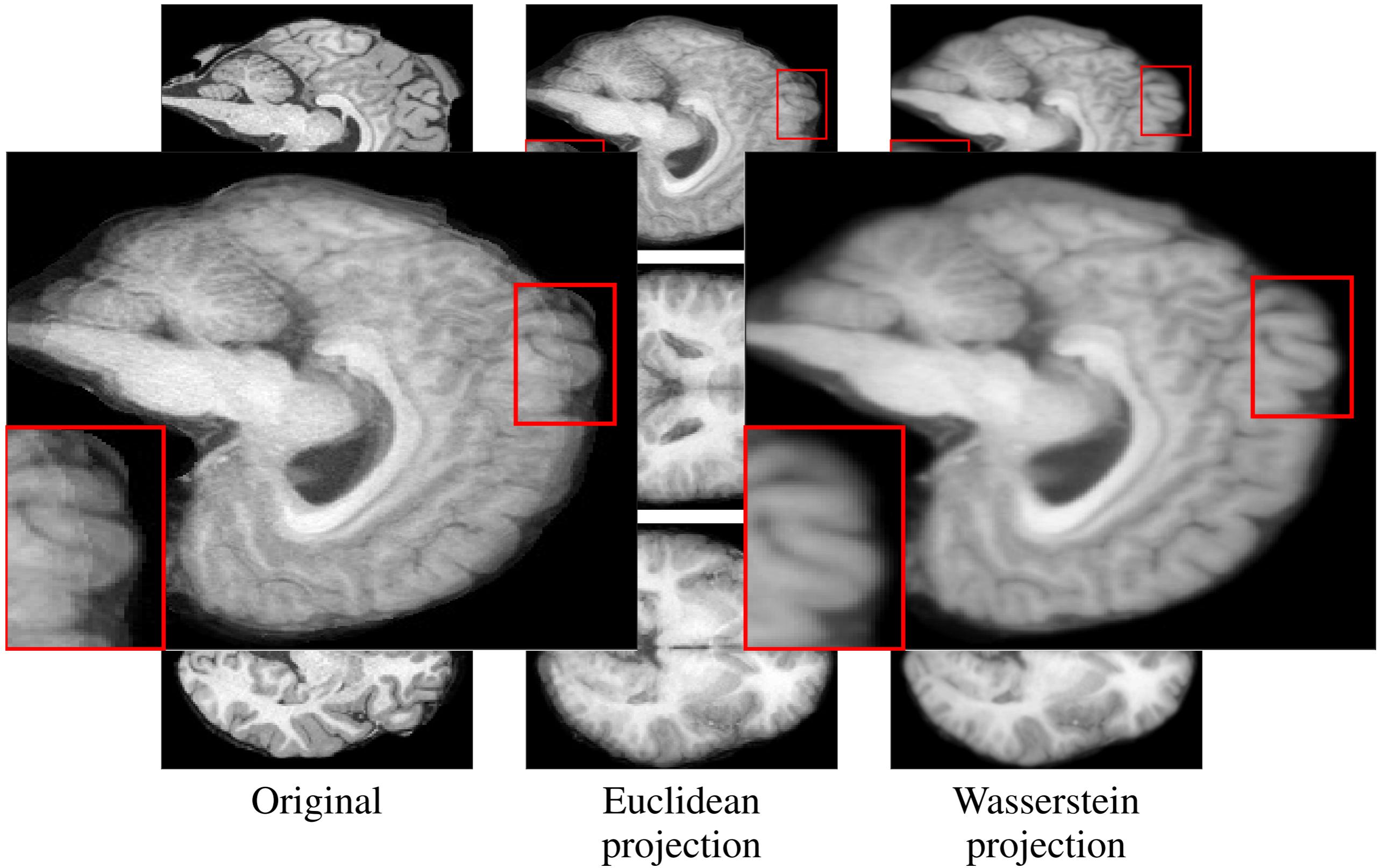


Original

Euclidean
projection

Wasserstein
projection

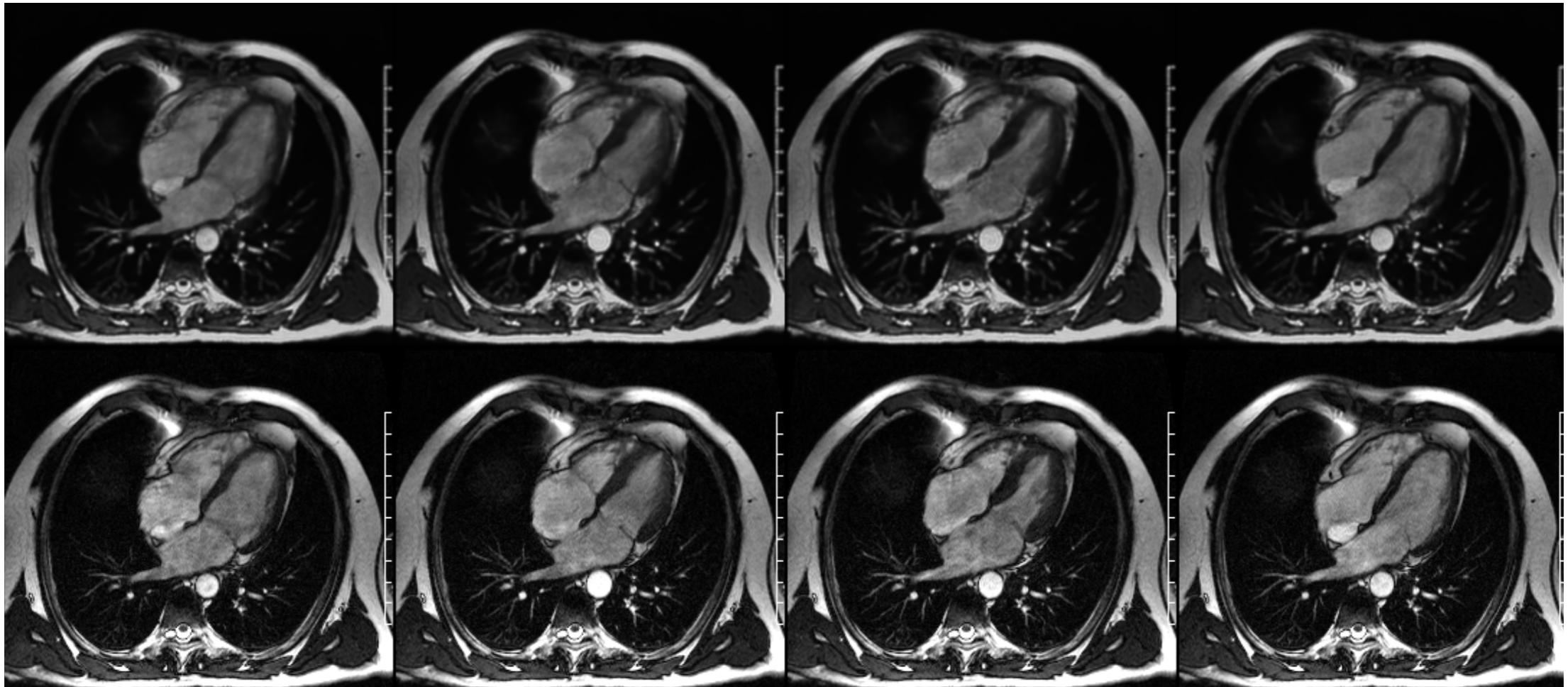
Application: Brain Mapping



end-to-end W Dictionary Learning

$$\min_{\textcolor{red}{A} \in (\Sigma_n)^K \textcolor{green}{\Lambda} \in (\Sigma_K)^N} \sum_{i=1}^N \mathcal{L}(\textcolor{blue}{b}_i, \textcolor{red}{a}(\textcolor{green}{\lambda}_i))$$

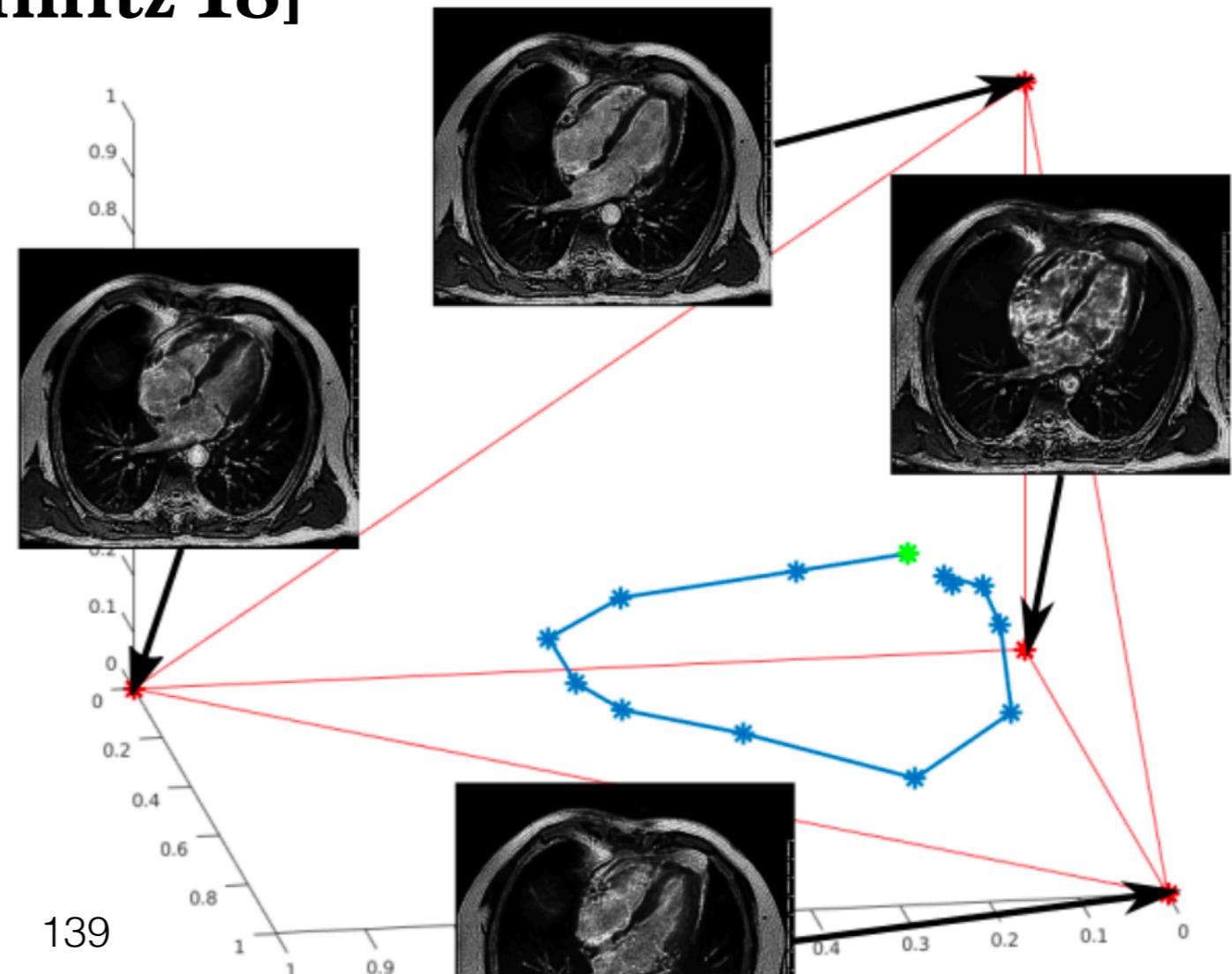
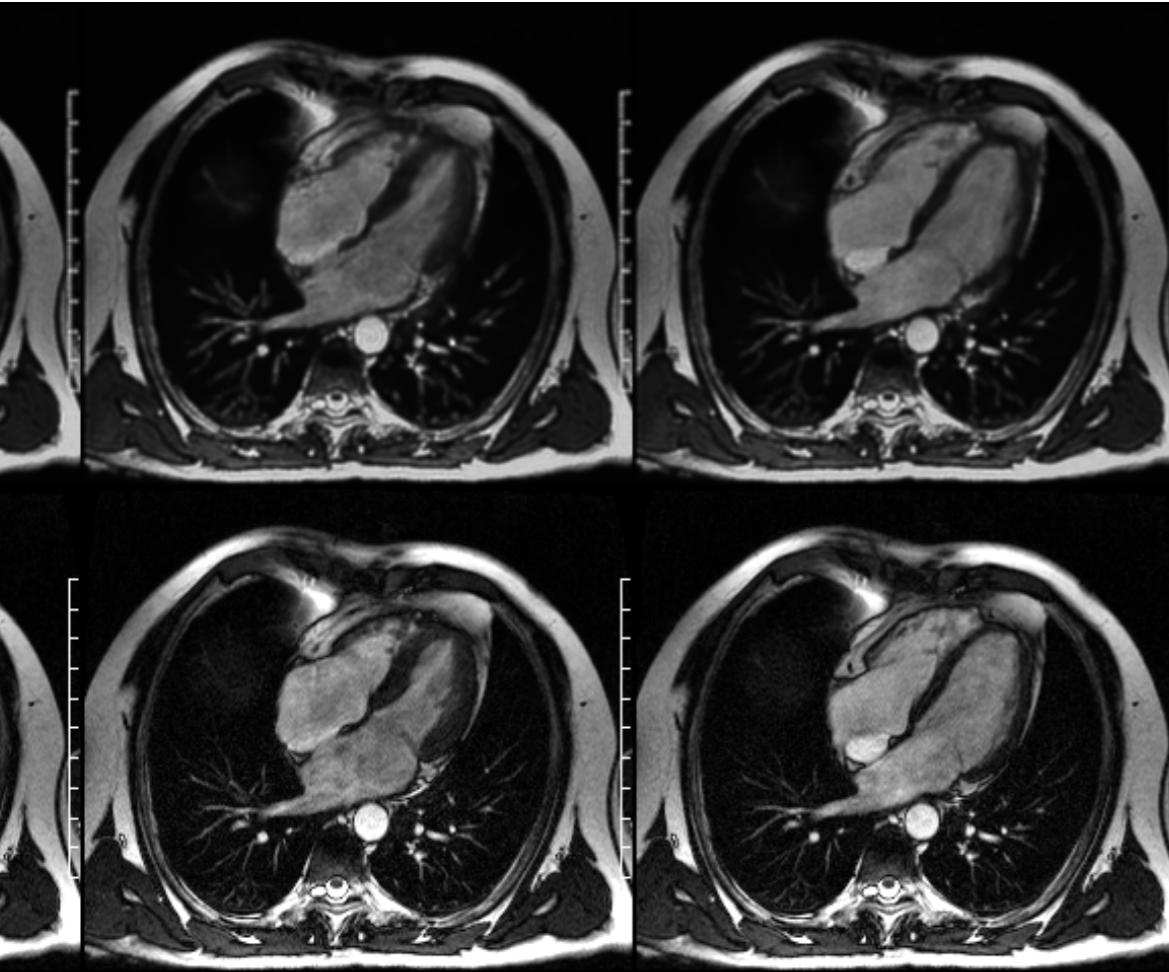
[Schmitz'18]



end-to-end W Dictionary Learning

$$\min_{\mathbf{A} \in (\Sigma_n)^K, \mathbf{\Lambda} \in (\Sigma_K)^N} \sum_{i=1}^N \mathcal{L}(\mathbf{b}_i, \mathbf{a}(\boldsymbol{\lambda}_i))$$

[Schmitz'18]



Distributionally Robust Optimization

$$\nu_{\text{data}} = \frac{1}{n} \sum_{i=1}^N \delta_{(x_i, y_i)}$$

Supervised learning

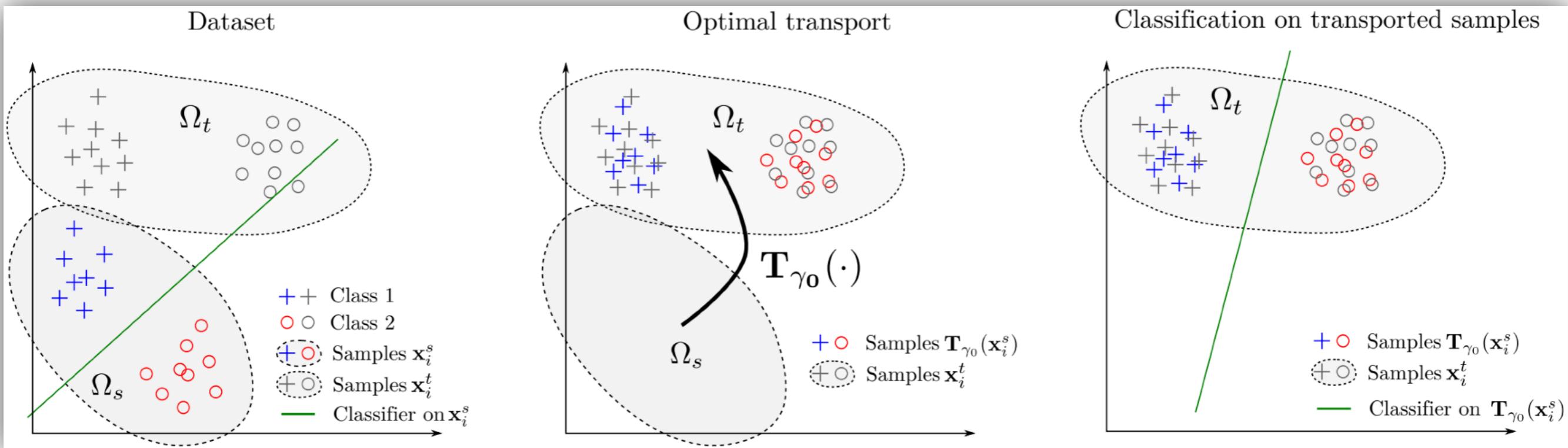
$$\inf_{\theta \in \Theta} \mathbb{E}_{\nu_{\text{data}}} [\mathcal{L}(f_\theta(X), Y)]$$

Learning with Wasserstein Ambiguity

$$\inf_{\theta \in \Theta} \sup_{\mu: W_p(\nu_{\text{data}}, \mu) < \varepsilon} \mathbb{E}_{\mu} [\mathcal{L}(f_\theta(X), Y)]$$

[Esvahani'17]

Domain Adaptation



1. Estimate transport map
2. Transport labeled samples to new domain
3. Train classifier on transported labeled samples

[Courty'16]

Learning with a Wasserstein Loss

Dataset $\{(x_i, y_i)\}$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}_+^n$



x_i

husky
snow
sled
slope
men

y_i

Goal is to find f_{θ} : Images \mapsto Labels

Learning with a Wasserstein Loss

$$\min_{\theta \in \Theta} \sum_{i=1}^N \mathcal{L}(f_{\theta}(x_i), y_i)$$



x_i

husky
snow
sled
slope
men

y_i

Which loss \mathcal{L} could we use?

Learning with a Wasserstein Loss

$$\min_{\theta \in \Theta} \sum_{i=1}^N \mathcal{L}(f_{\theta}(x_i), y_i)$$

dog
driver
winter
ice

$f_{\theta}(x_i)$

husky
snow
sled
slope
men

y_i

Which loss \mathcal{L} could we use?

Learning with a Wasserstein Loss

$$\min_{\theta \in \Theta} \sum_{i=1}^N \mathcal{L}(f_{\theta}(x_i), y_i)$$

$$\begin{aligned} \mathcal{L}(\mathbf{a}, \mathbf{b}) = & \min_{\mathbf{P} \in \mathbb{R}^{nm}} \langle \mathbf{P}, M \rangle + \varepsilon \text{KL}(\mathbf{P} \mathbf{1}, \mathbf{a}) \\ & + \varepsilon \text{KL}(\mathbf{P}^T \mathbf{1}, \mathbf{b}) - \gamma E(\mathbf{P}) \end{aligned}$$

1. Generalizes Word Mover's to label clouds
2. Sinkhorn algorithm can be generalized

[Frogner'15] [Chizat'15][Chizat'16]

Life Sciences

- Biology to infer evolution of cells

[Hashimoto+’16] [.... Rigollet ...’19 : *Waddington OT*]

Cell

Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming

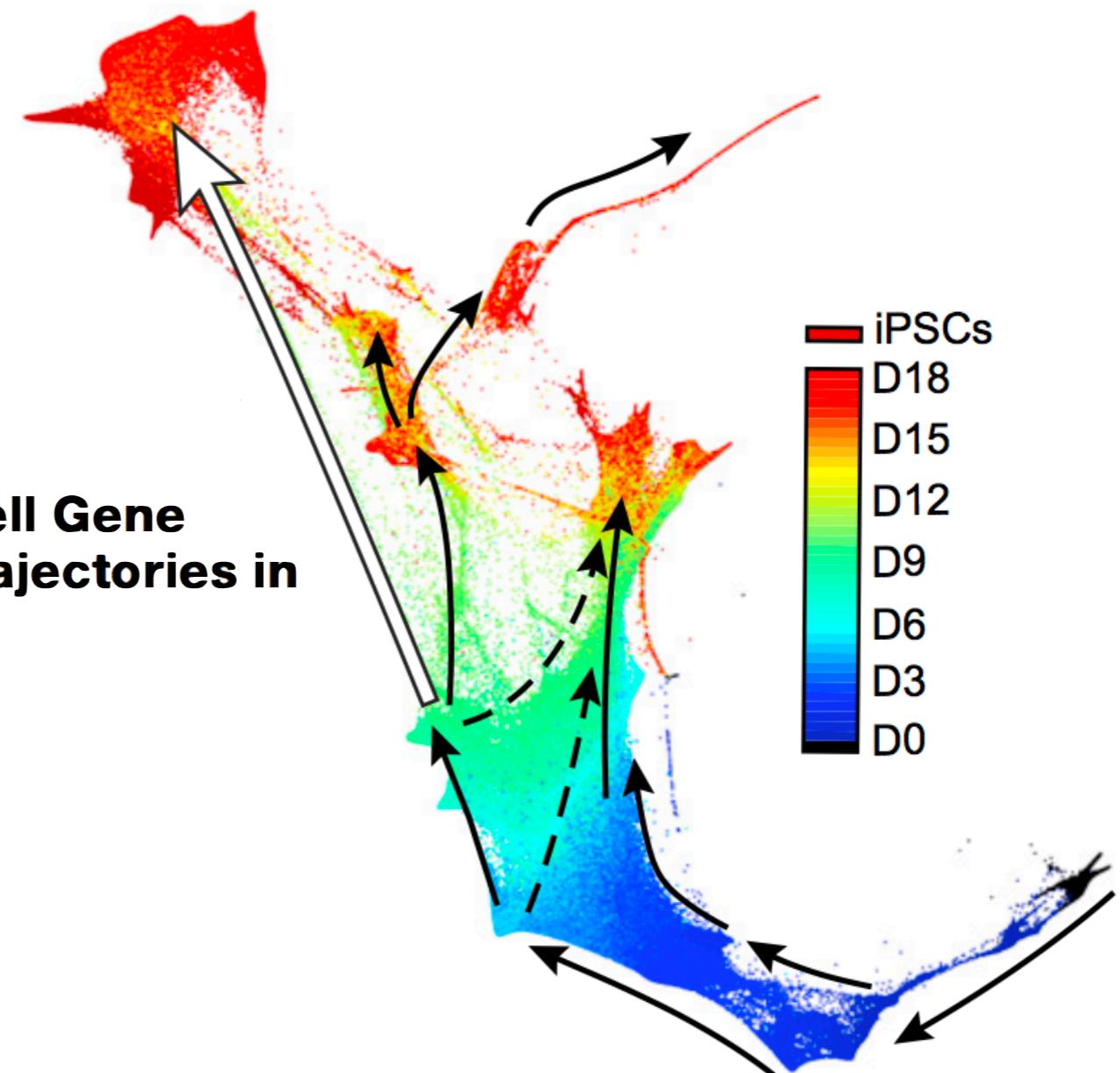
Life Sciences

- Biology to infer evolution of cells

[Hashimoto+’16] [.... Rigollet ...’19 : *Waddington OT*]

Cell

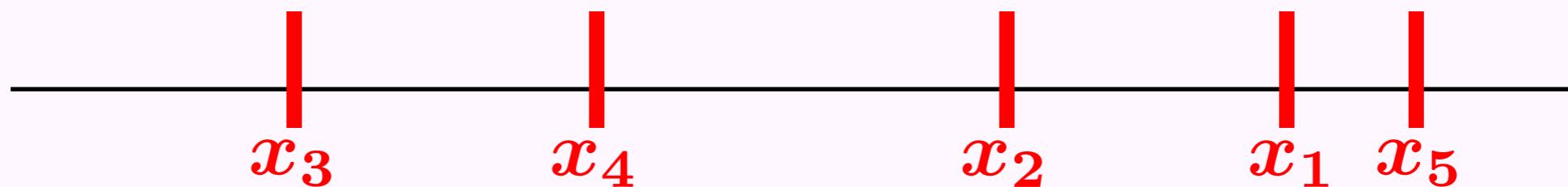
Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming



Ranks and Sort operator

Sorting permutation

$$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$$



$$\sigma(\mathbf{x}) = (3, 4, 2, 1, 5)$$

Ranks and Sort operator

Sorting permutation

$$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$$



$$\sigma(\mathbf{x}) = (3, 4, 2, 1, 5)$$

permutation matrix

$$\Pi_{\boldsymbol{\sigma}} = \text{sp1}((i, \boldsymbol{\sigma}_i)_i)$$

$$\Pi_{\sigma^{-1}} = \Pi_{\sigma}^T$$

$$\Pi_{\sigma(\mathbf{x})} = \begin{bmatrix} & & 1 & & \\ & & 1 & & \\ & & & 1 & \\ & & & 1 & \\ 1 & & & & \\ & & & & 1 \end{bmatrix}$$

Ranks and Sort operator

Ranking / Sorting

$$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$$



$$\sigma(\mathbf{x}) = (3, 4, 2, 1, 5)$$

$$R \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 2 \\ 5 \end{bmatrix} \quad S \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} x_3 \\ x_4 \\ x_2 \\ x_1 \\ x_5 \end{bmatrix}$$

$\hat{=} \sigma(\mathbf{x})^{-1}$ $\hat{=} \mathbf{x}_{\sigma(\mathbf{x})}$

Sorting as a Subroutine in ML

Classifiers

select top- k activations

k -NN

- (1) select neighbours
- (2) majority vote

Least Quantile regression

minimize empirical quantile of loss (not mean)

MoM

estimators

Ranking / Sorting

$O(n \log n)$

Learning to rank

NDCG loss and others

Rank-based statistics

data viewed as ranks

Descriptive statistics

Empirical distribution function
quantile normalization

Non-differentiability of ranks and sort

Ranking / Sorting

$$\mathbf{x} = (x_1, x_2, x_3, x_4 + \tau, x_5)$$

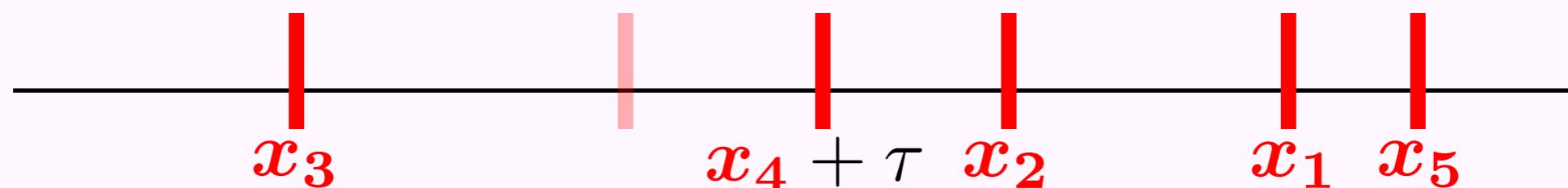


$$R \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 + \tau \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 2 \\ 5 \end{bmatrix}$$
$$S \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 + \tau \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} x_3 \\ x_4 + \tau \\ x_2 \\ x_1 \\ x_5 \end{bmatrix}$$

Non-differentiability of ranks and sort

Ranking / Sorting

$$\mathbf{x} = (x_1, x_2, x_3, x_4 + \tau, x_5)$$

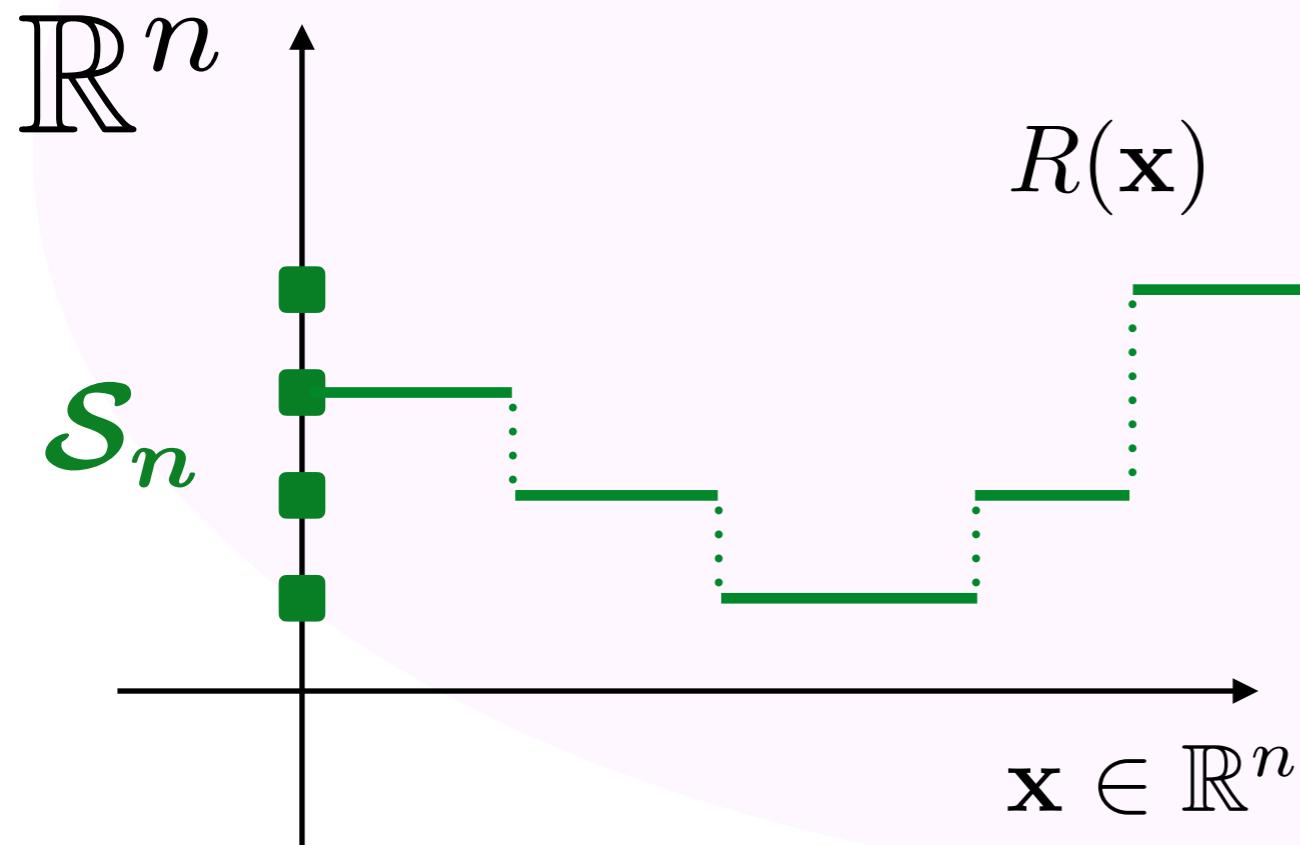


$$R \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 + \tau \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 2 \\ 5 \end{bmatrix}$$
$$S \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 + \tau \\ x_5 \end{bmatrix} \right) = \begin{bmatrix} x_3 \\ x_4 + \tau \\ x_2 \\ x_1 \\ x_5 \end{bmatrix}$$

Non-differentiability of ranks and sort

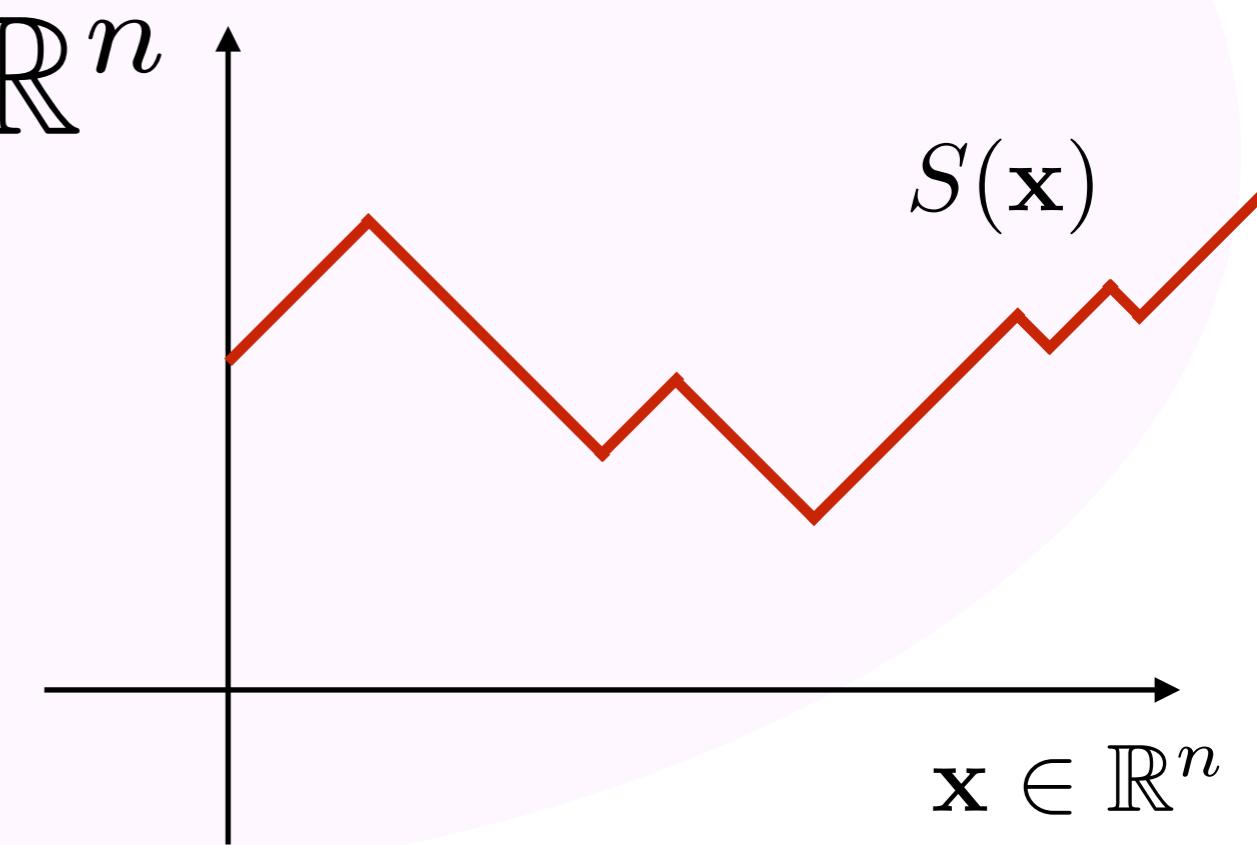
Ranking / Sorting

piecewise constant



$$JR(\mathbf{x}) \in \{0, \pm\infty\}^{n \times n}$$

piecewise linear



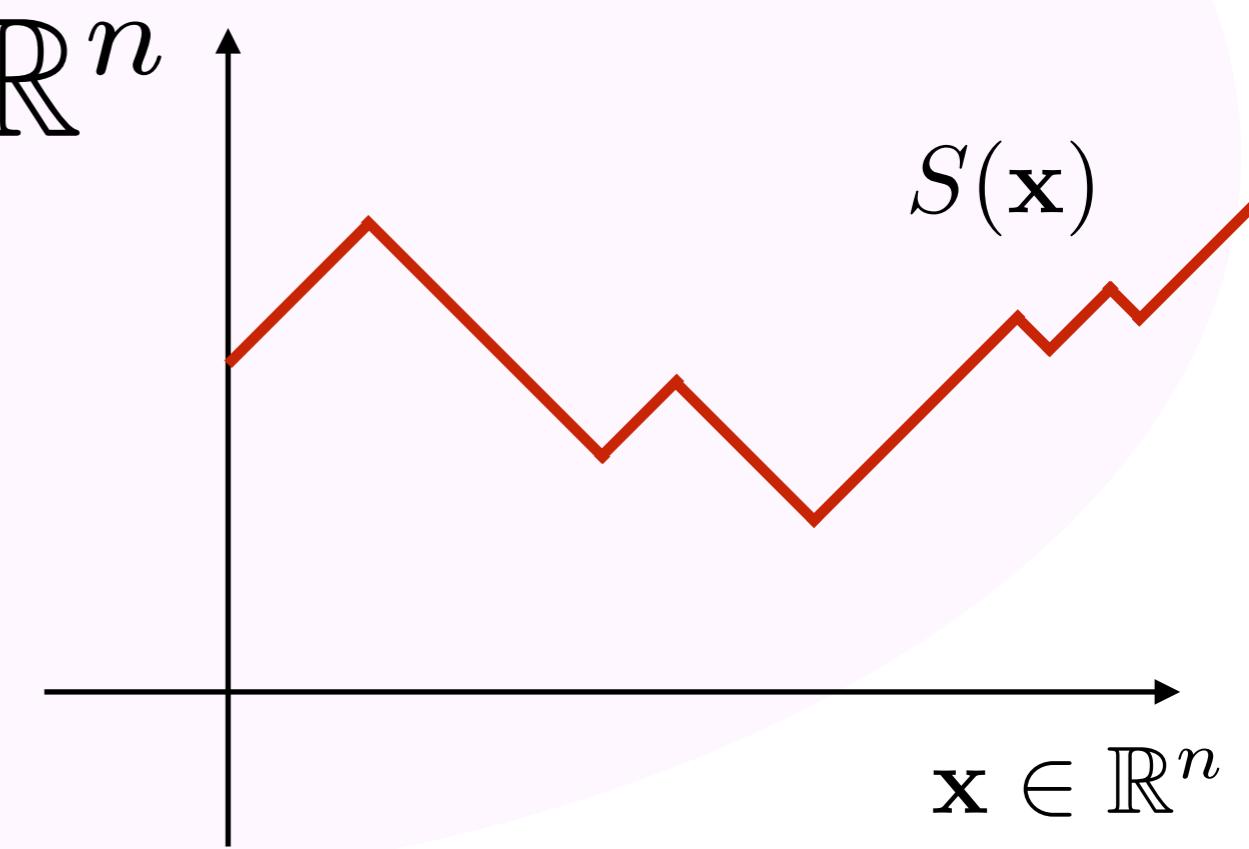
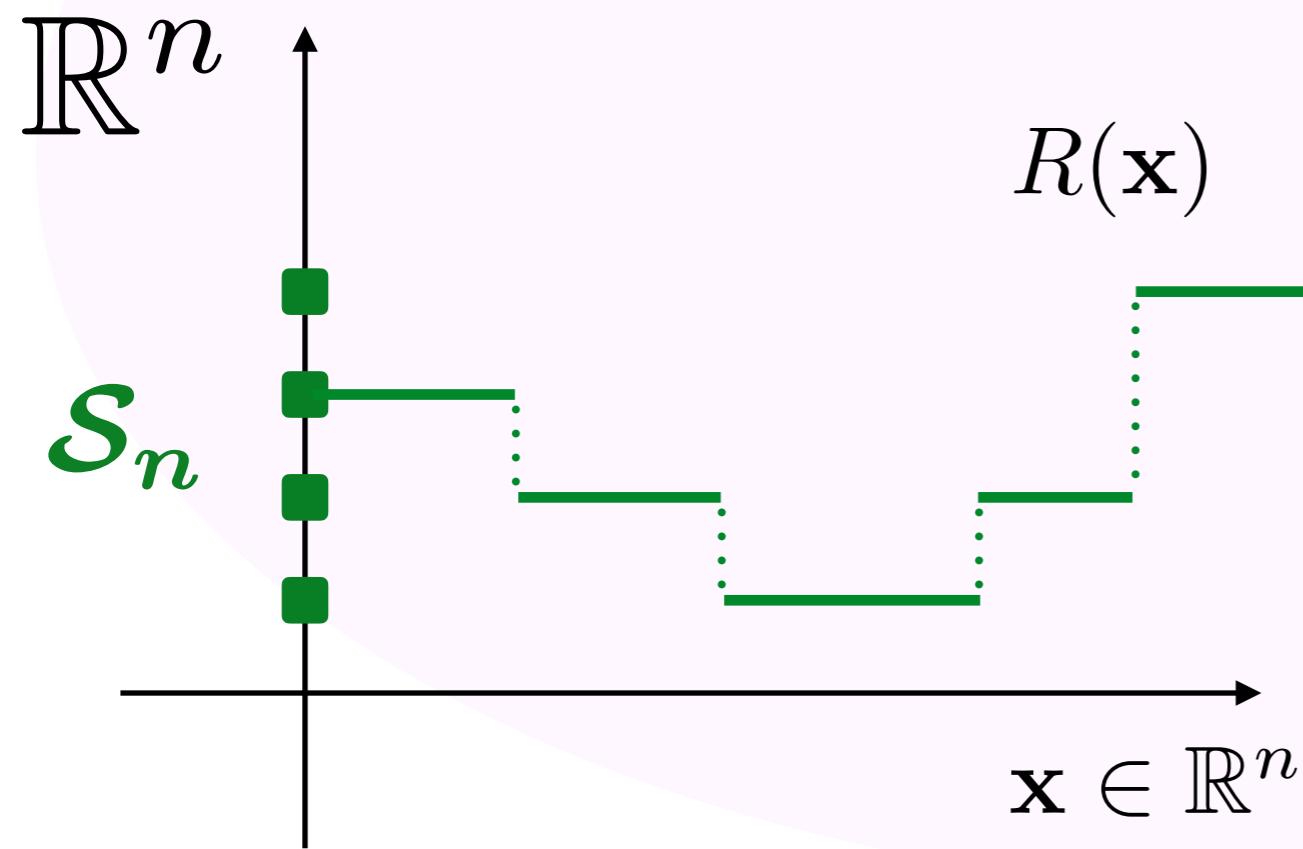
$$JS(\mathbf{x}) = \Pi_{\sigma(\mathbf{x})}^T \text{ a.e.}$$

Non-smoothness of ranks/sorts

- **k -NN** LMNN [Weinberger & Saul'06] Neural [Plötz & Roth'18]
- ***Learning to rank***
 - **Pairwise losses:** Ranknet[Burges+'05], LambdaRank[Burges+'07], Rankboost [Freund+'03], BoltzRank [Volkovs & Zemel'09]
 - **Smoothed NDCG:** SoftRanks [Taylor+'08][Chapelle & Wu'09']
- ***Multiclass classification*** XE on Softmax activations, smoothed top- k losses [Boyd+'12][Berrada+'18] Focal loss [Lin+'17]
- ***Trimmed Regression*** Combinatorial approaches

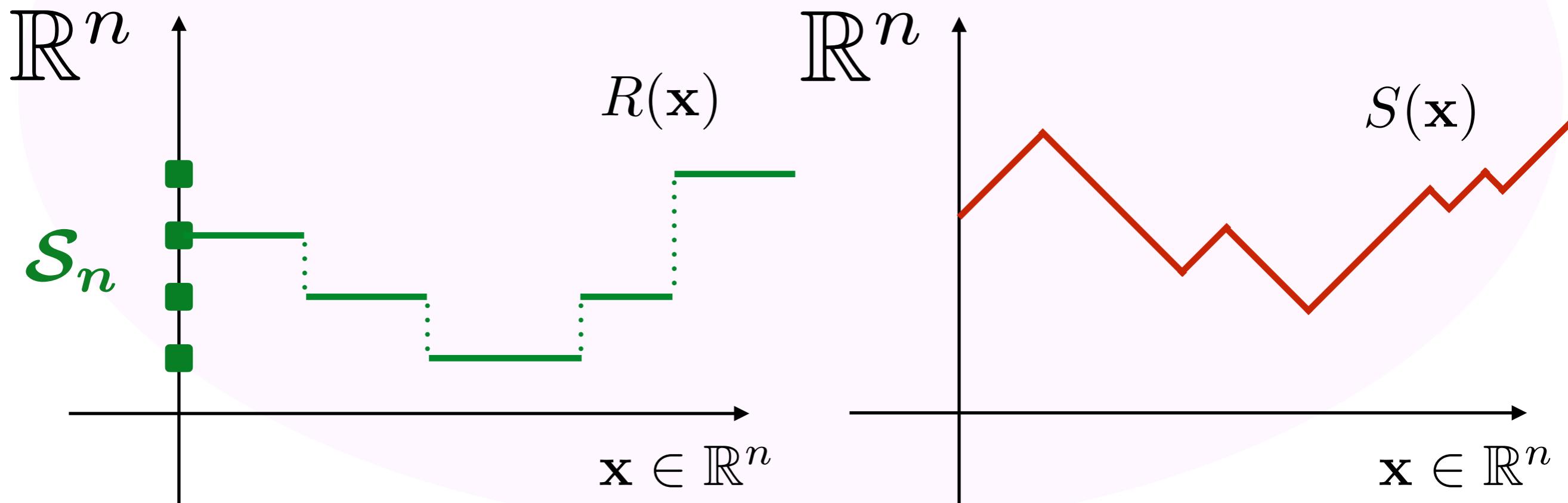
Soft Ranks and Sort Operators

Ranking / Sorting



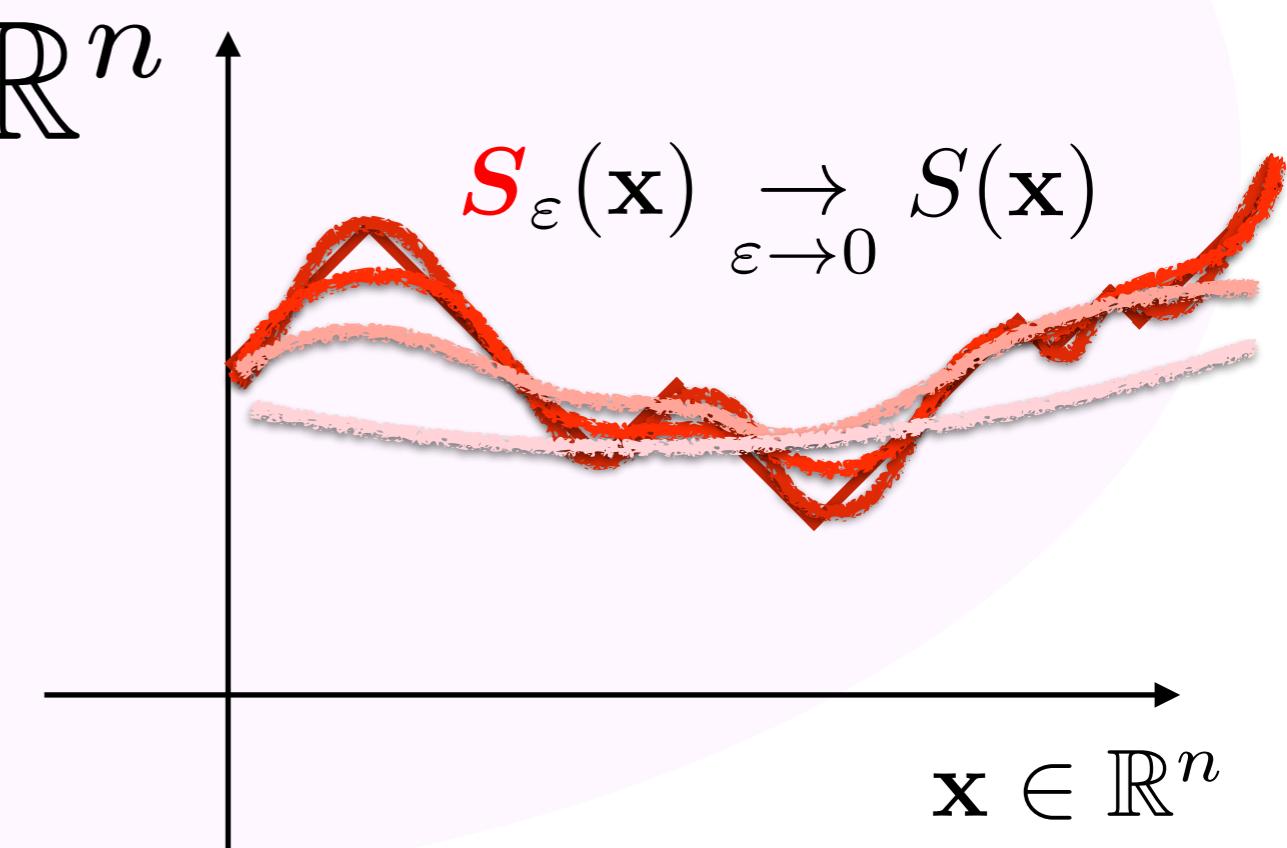
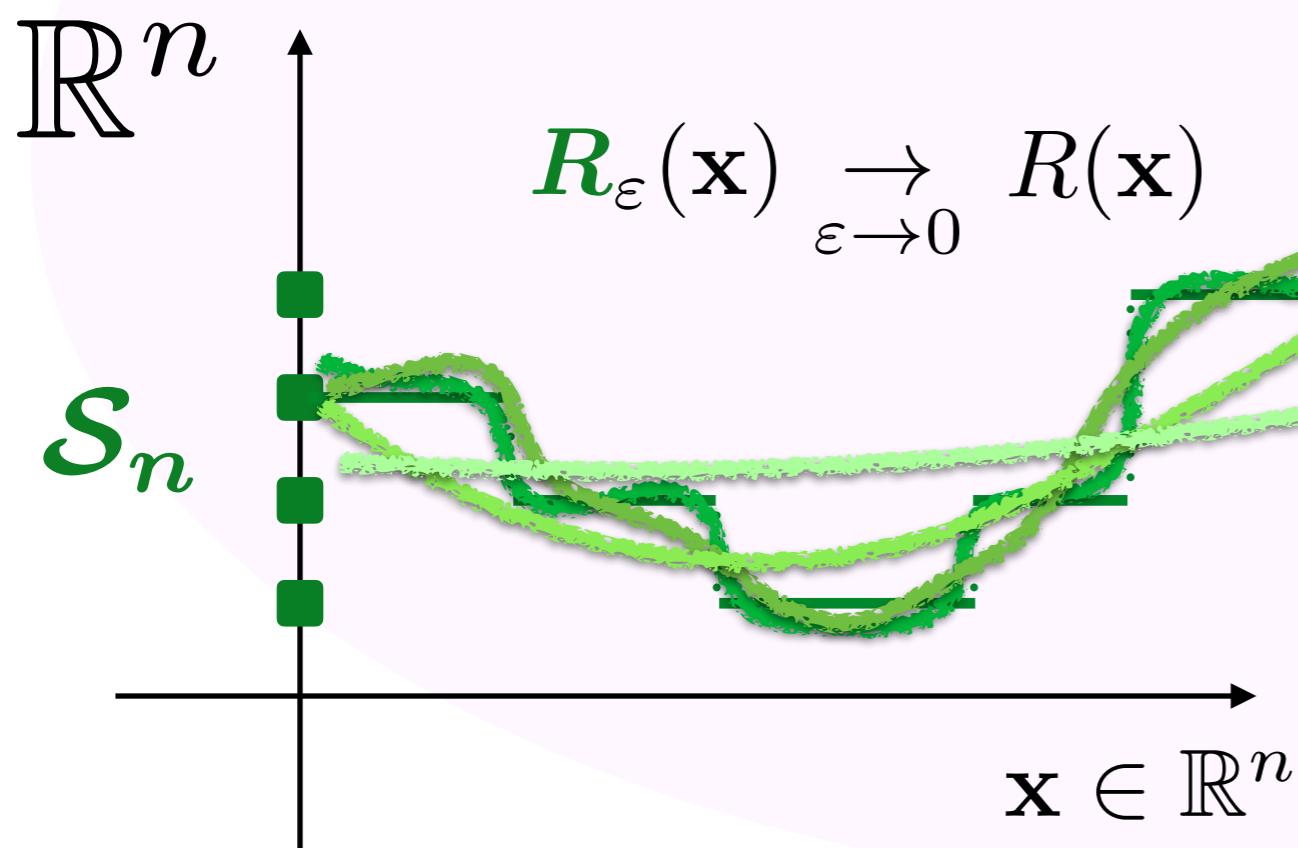
Soft Ranks and Sort Operators

Goal: define (programmatically) **everywhere differentiable** approximation functions for R/S , **arbitrarily close** to the true R/S vector outputs



Soft Ranks and Sort Operators

Goal: define (programmatically) **everywhere differentiable** approximation functions for R/S , **arbitrarily close** to the true R/S vector outputs

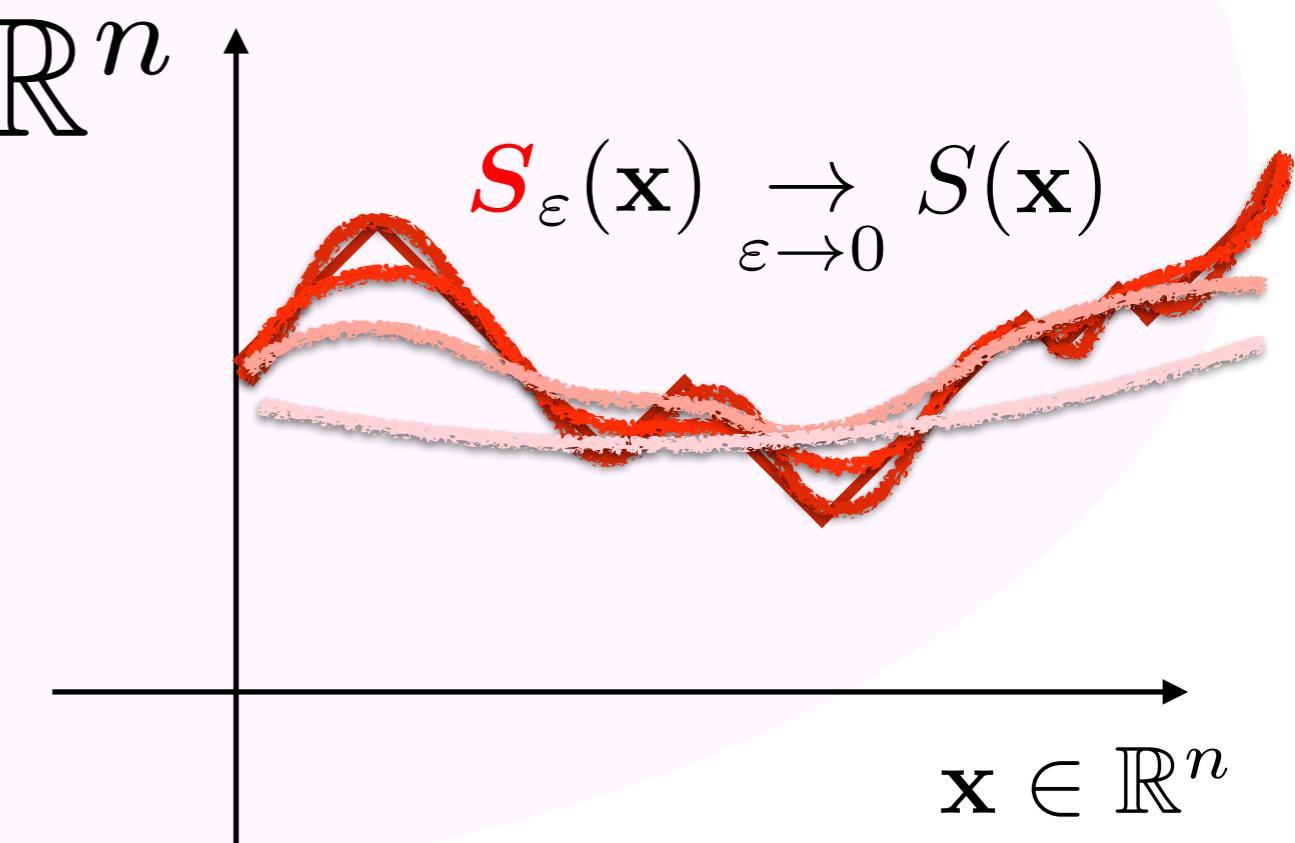
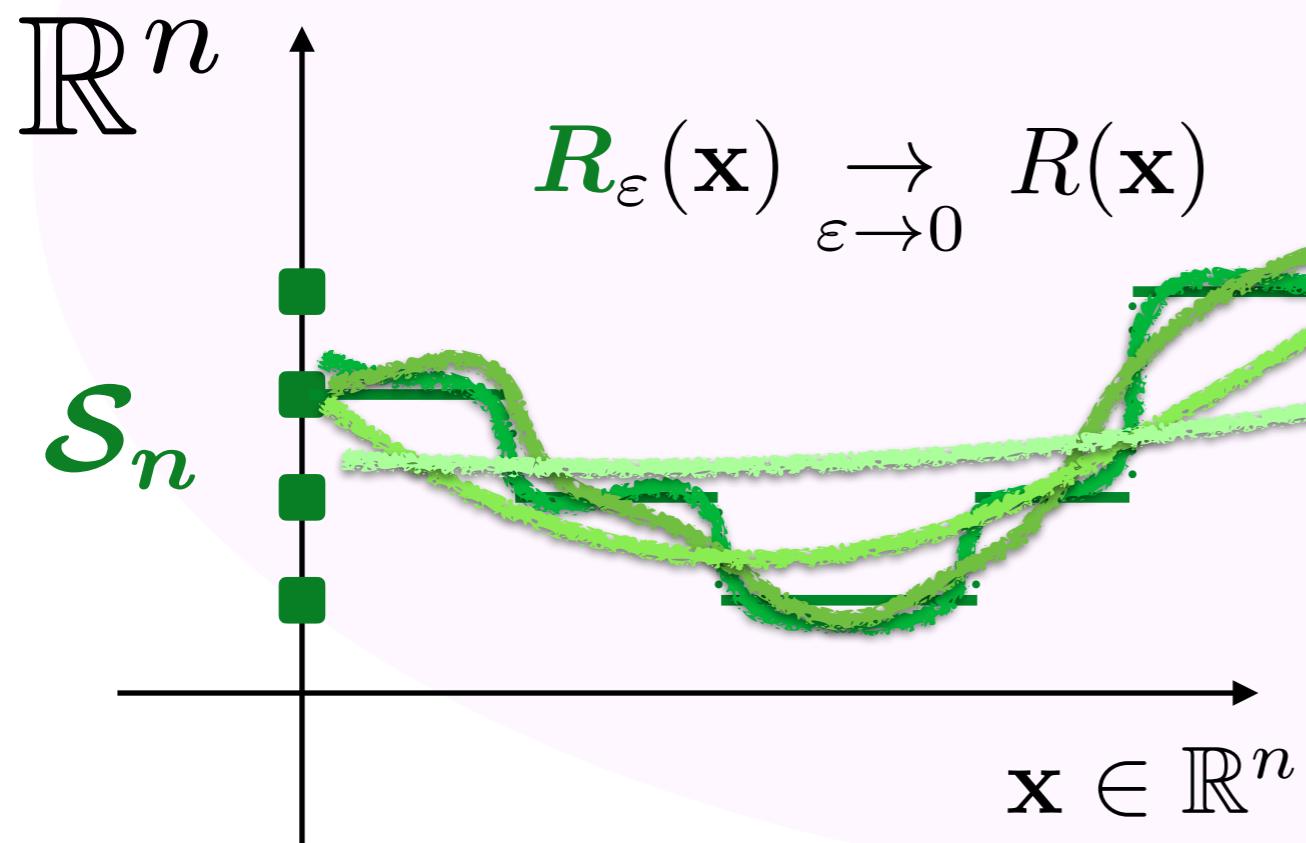


Soft Ranks and Sort Operators

Why?

Now Train the way you test + ε

Next Constraints in real life are *relative* (fairness)



Related work

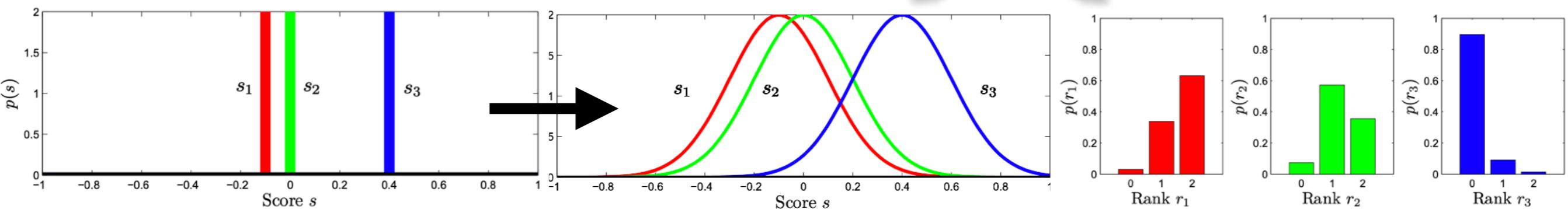
SoftRank: Optimising Non-Smooth Rank Metrics

Mike Taylor, John Guiver, Stephen Robertson, Tom Minka

WSDM 2008 | February 2008

$O(n^3)$

Approximation with
documented **pathologies**

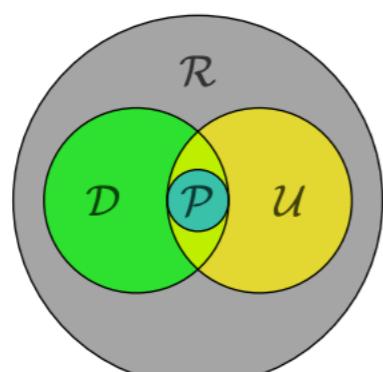


Stochastic Optimization of Sorting Networks via Continuous Relaxations

Aditya Grover, Eric Wang, Aaron Zweig, Stefano Ermon

Published as a conference paper at ICLR 2019

$$\begin{pmatrix} 0 & 1/2 & 1/2 \\ 7/16 & 3/16 & 3/8 \\ 9/16 & 5/16 & 1/8 \end{pmatrix}$$



Doubly stochastic

$$\begin{pmatrix} 3/8 & 1/8 & 1/2 \\ 3/4 & 1/4 & 0 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}$$

Unimodal row matrices

$$A_{\mathbf{s}}[i, j] = |s_i - s_j|.$$

$$\hat{P}_{\text{sort}(\mathbf{s})}[i, :](\tau) = \text{softmax} [((n+1-2i)\mathbf{s} - A_{\mathbf{s}}\mathbf{1})/\tau]$$

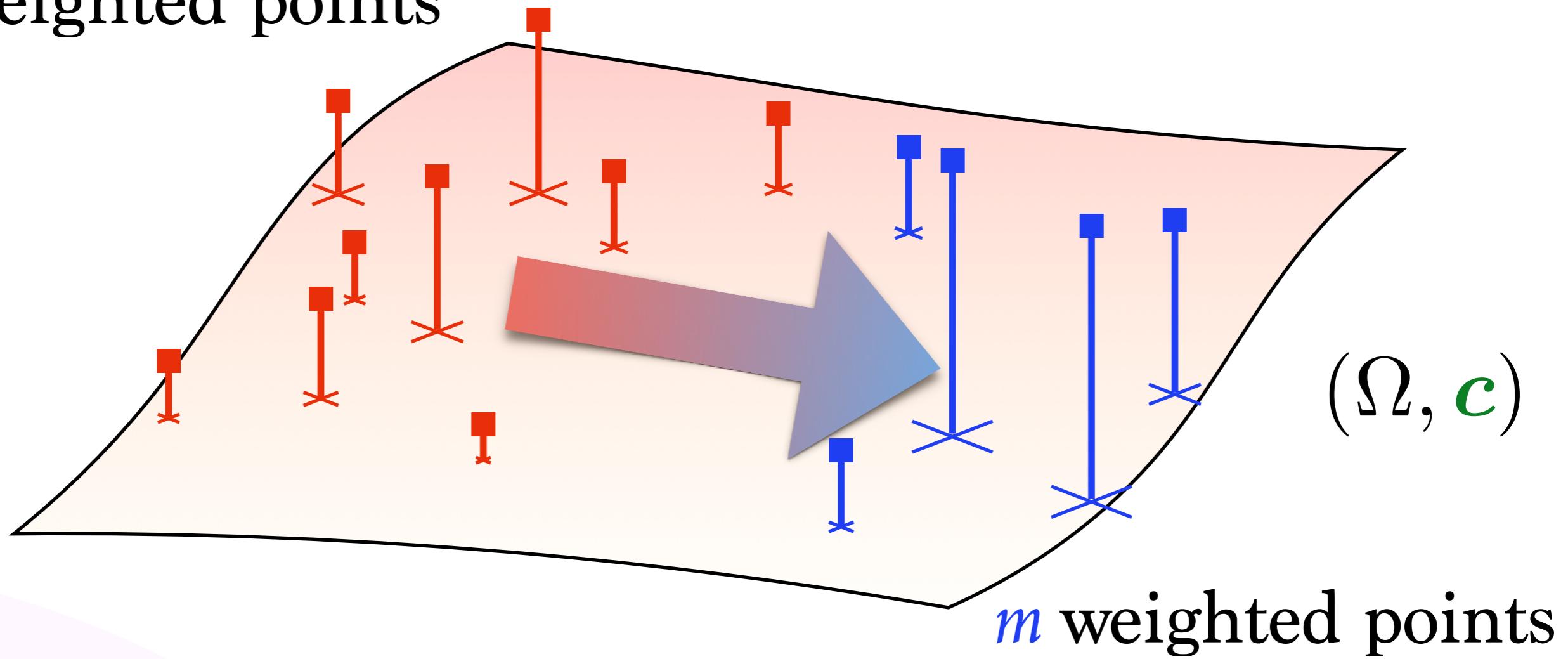
$O(n^2)$

Our Approach builds on OT

Optimal Transport

$$O((n+m)nm \log(n+m))$$

n weighted points



OT in 1D needs sorting

Optimal Transport

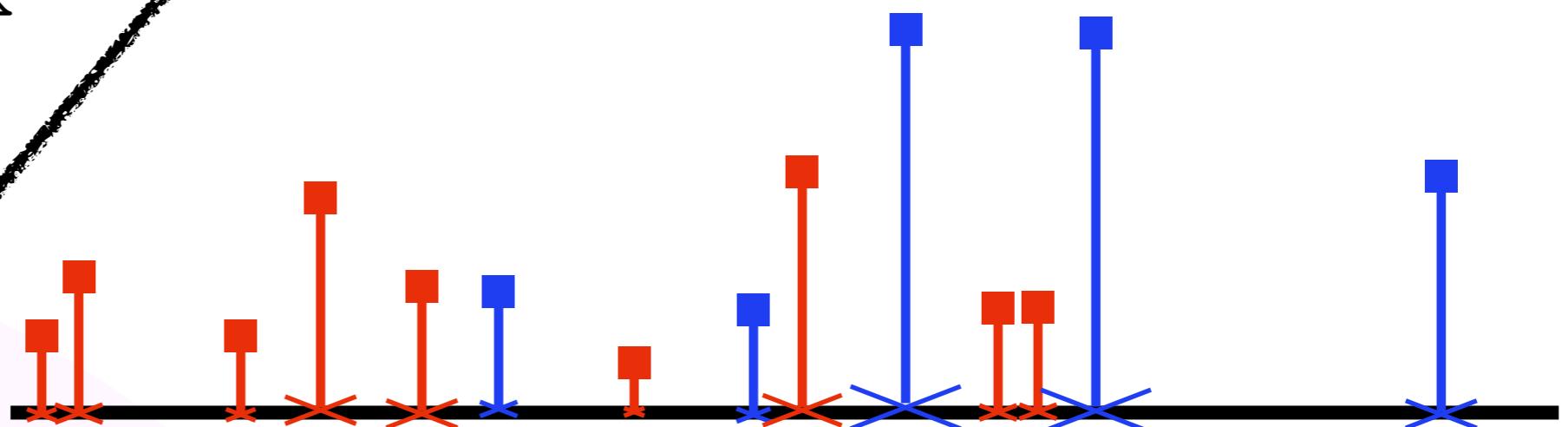
$$O((n+m)nm \log(n+m))$$

*important to solve
OT when $\Omega = \mathbb{R}$*

Sorting / Sorting

$$O(n \log n)$$

$$+ O(m \log m)$$



$$\Omega = \mathbb{R}$$

Our idea in one slide

Optimal Transport

$$O((n+m)nm \log(n+m))$$

*important to solve
OT when $\Omega = \mathbb{R}$*



g / Sorting

$$O(n \log n)$$

$$+ O(m \log m)$$

Our idea in one slide

Optimal Transport

generalize both $O((n+m)nm \log(n+m))$

using OT
(overkill!!)



g / Sorting

$O(n \log n)$

$O(m^+ \log m)$

Our idea in one slide

Optimal Transport

generalize both $O((n+m)nm \log(n+m))$

using OT
(overkill!!)



g / Sorting

$O(n \log n)$

$+ O(m \log m)$

Regularized OT
Sinkhorn Algorithm

$O(nm)$

Our idea in one slide

Optimal Transport

generalize both $O((n+m)nm \log(n+m))$

using OT
(overkill!!)

*faster and
differentiable
variant.*

Differentiable
sorting in $O(nm)$

Sorting / Sorting
 $O(n \log n)$

$+ O(m \log m)$

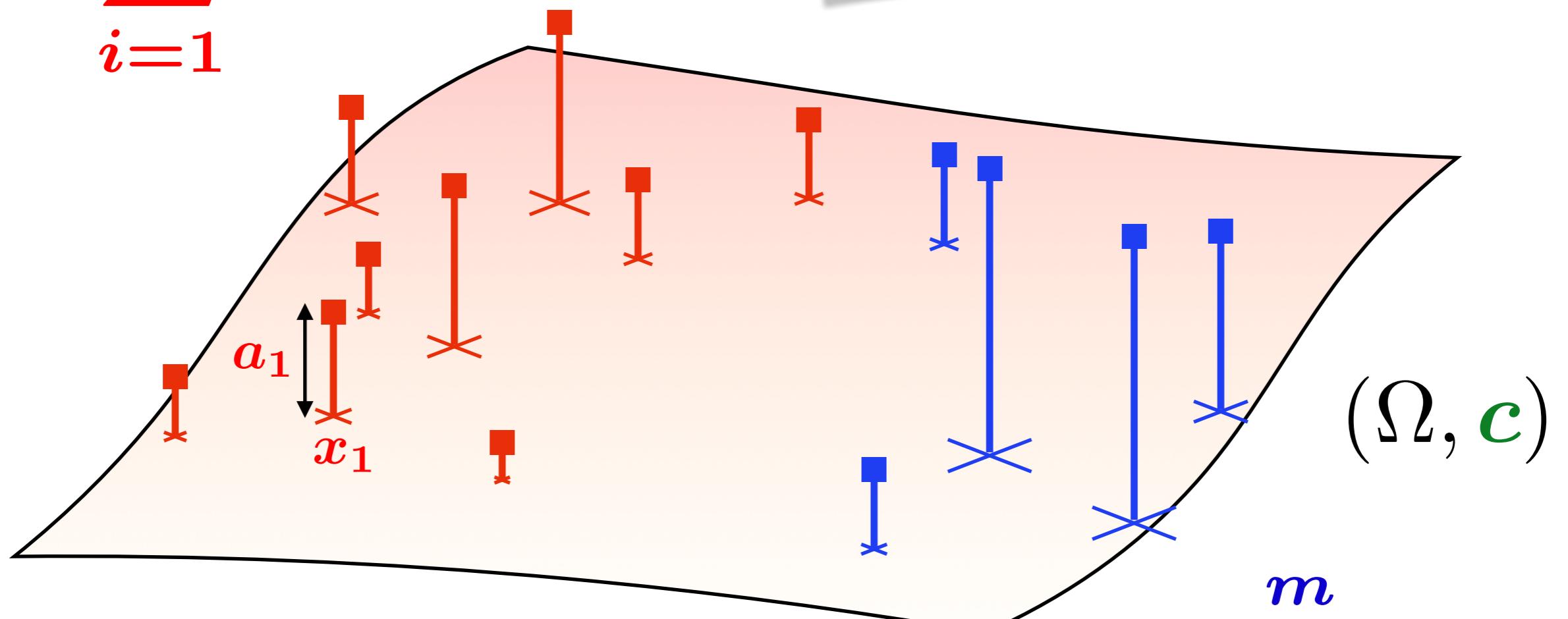
Regularized OT
Sinkhorn Algorithm

$O(nm)$

Discrete OT Problem

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

$$\sum_i a_i = \sum_j b_j = 1$$

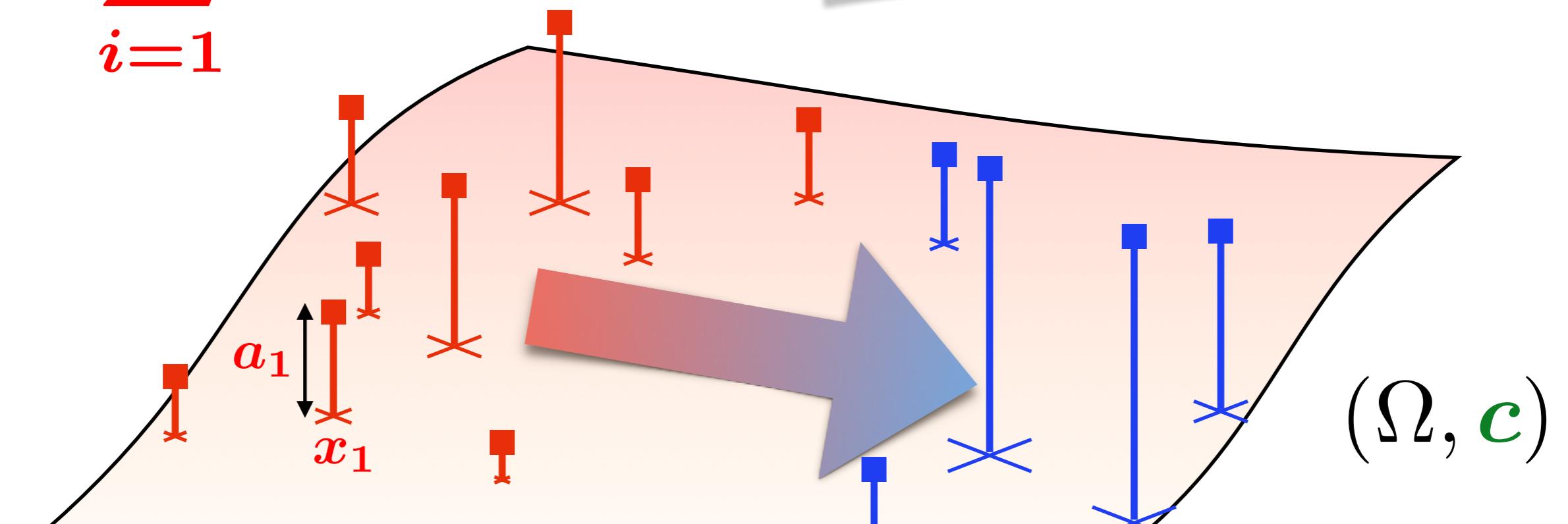


$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$

Discrete OT Problem

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

$$\sum_i a_i = \sum_j b_j = 1$$



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

LP Formulation

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

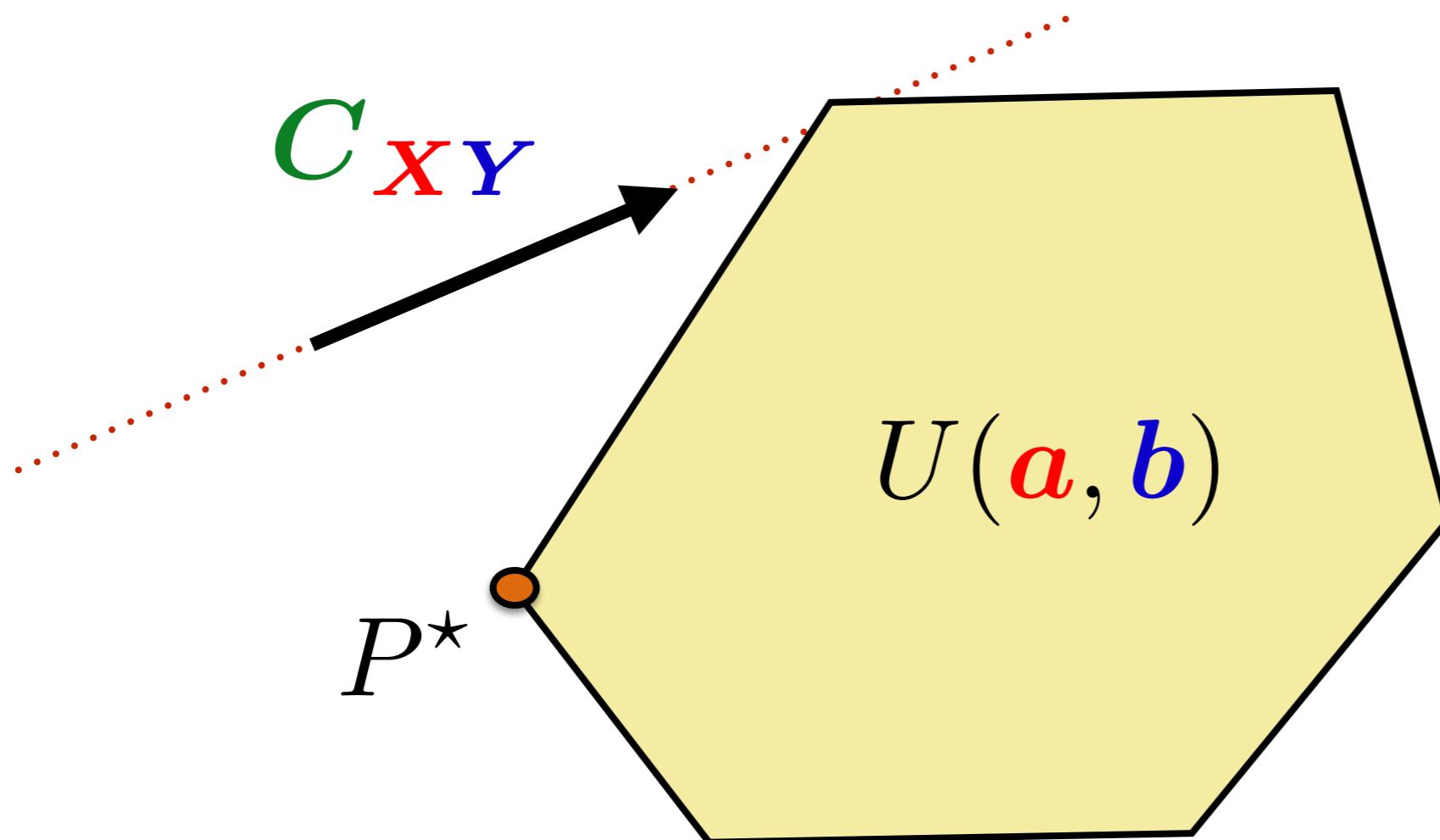
$$C_{XY} \stackrel{\text{def}}{=} [c(x_i, y_j)]_{ij}$$

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

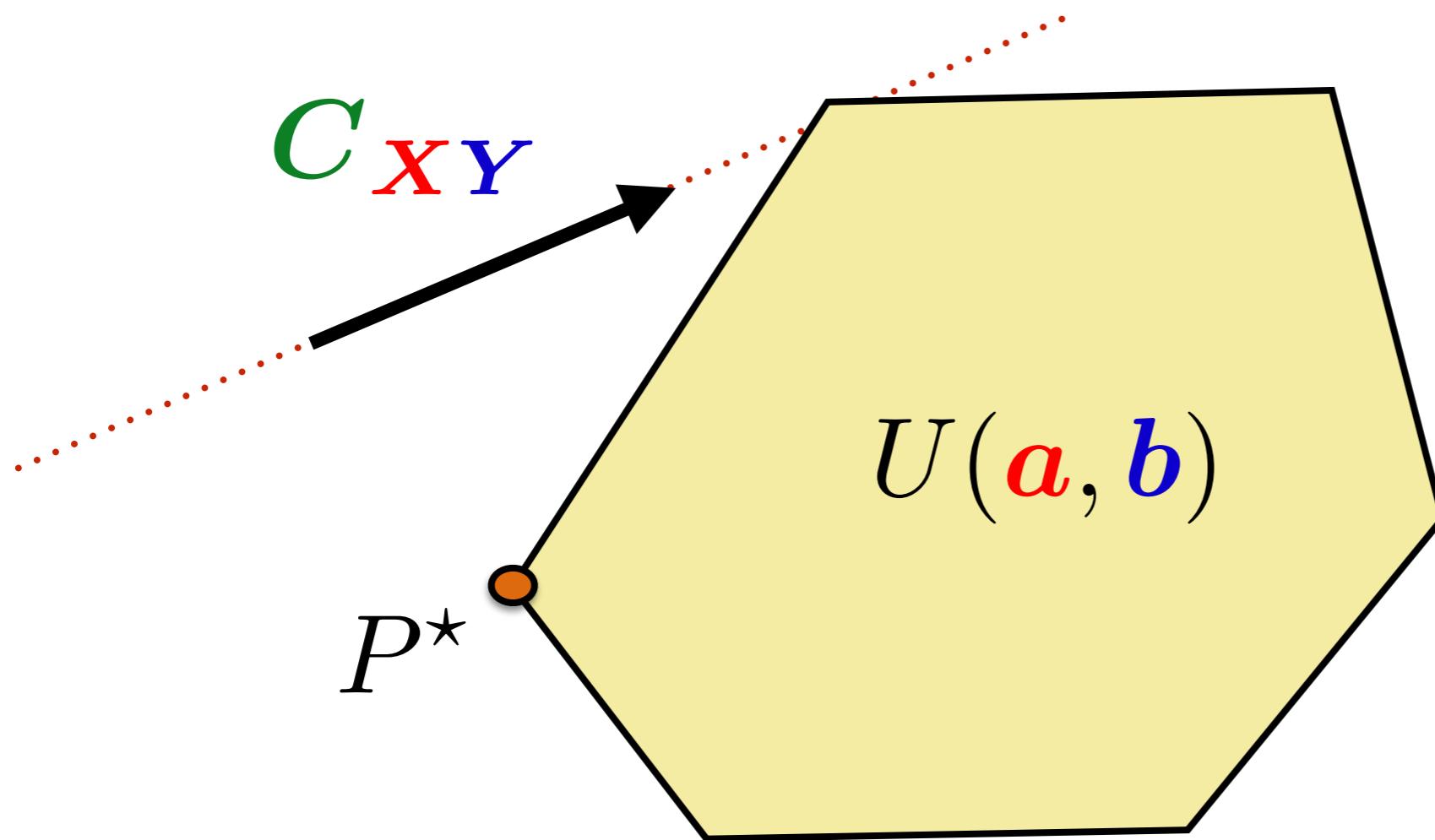
Def. Optimal Transport Problem

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle P, C_{XY} \rangle$$

Computations



Computations



min cost flow solver used in practice.

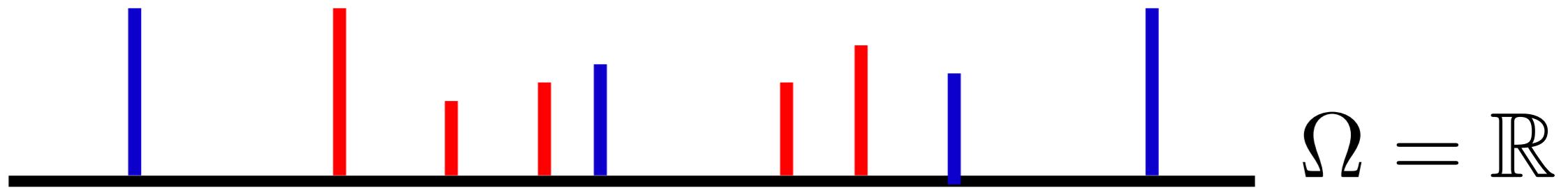
$O((n + m)nm \log(n + m))$



Optimal Transport in 1D

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

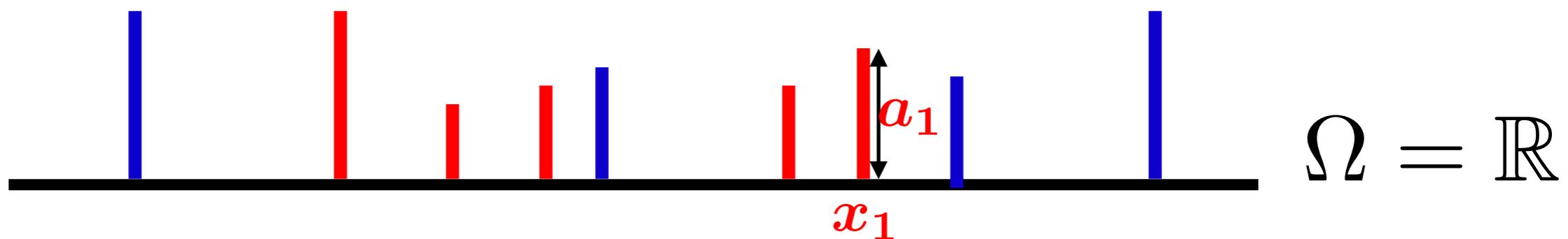
$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$



Optimal Transport in 1D

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

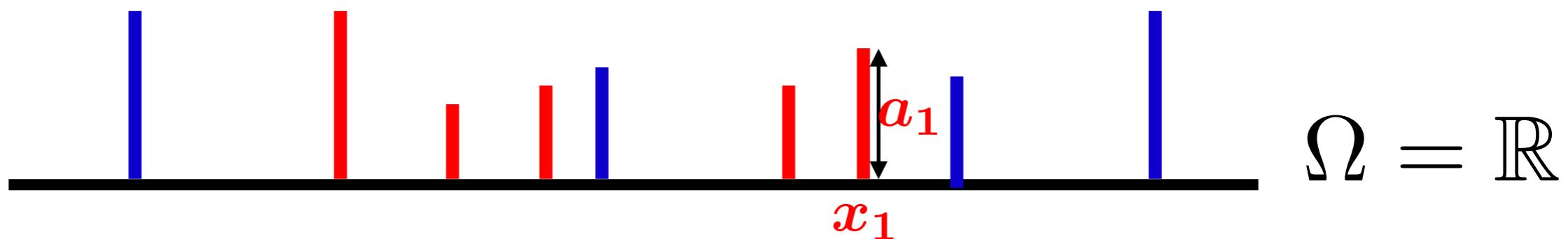
$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$



Optimal Transport in 1D

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

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



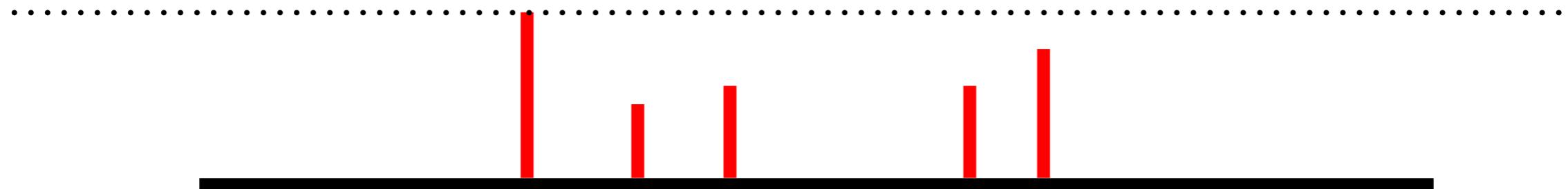
Assume...

$c(x, y) := h(y - x)$, h is convex

...then OT boils down to sorting

Building Emp. Distribution Functions

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

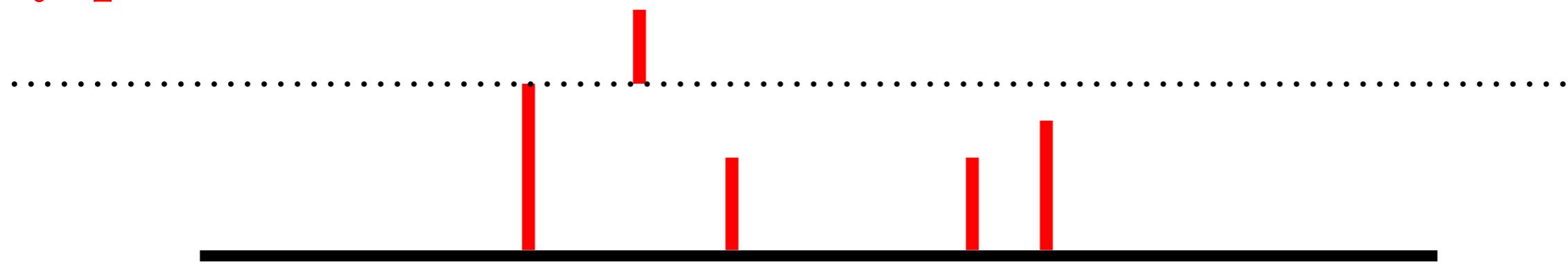


$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$

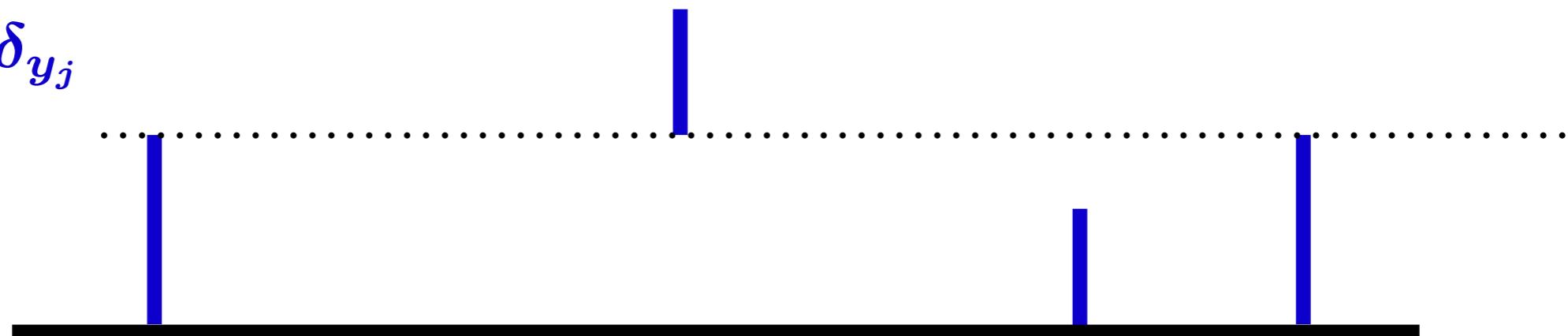


Building Emp. Distribution Functions

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

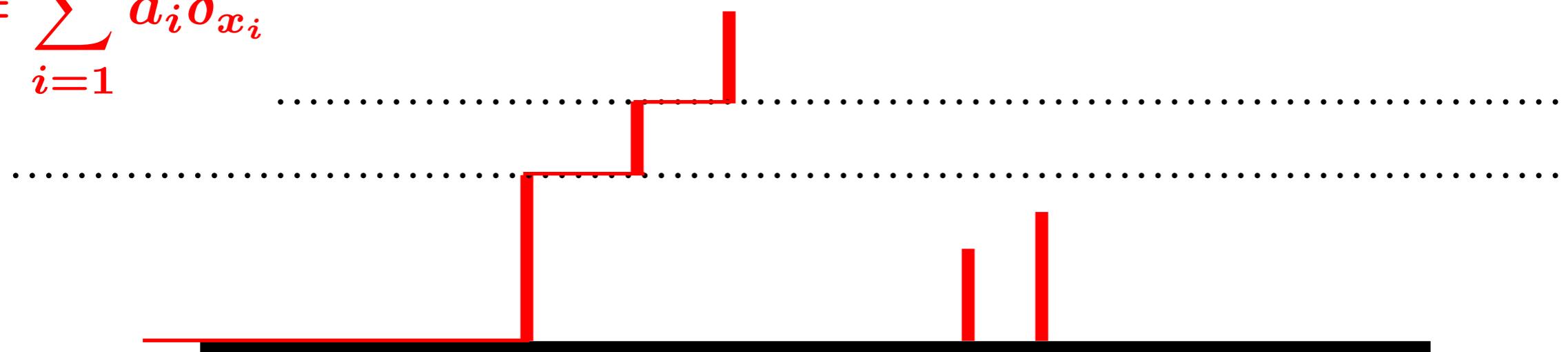


$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$

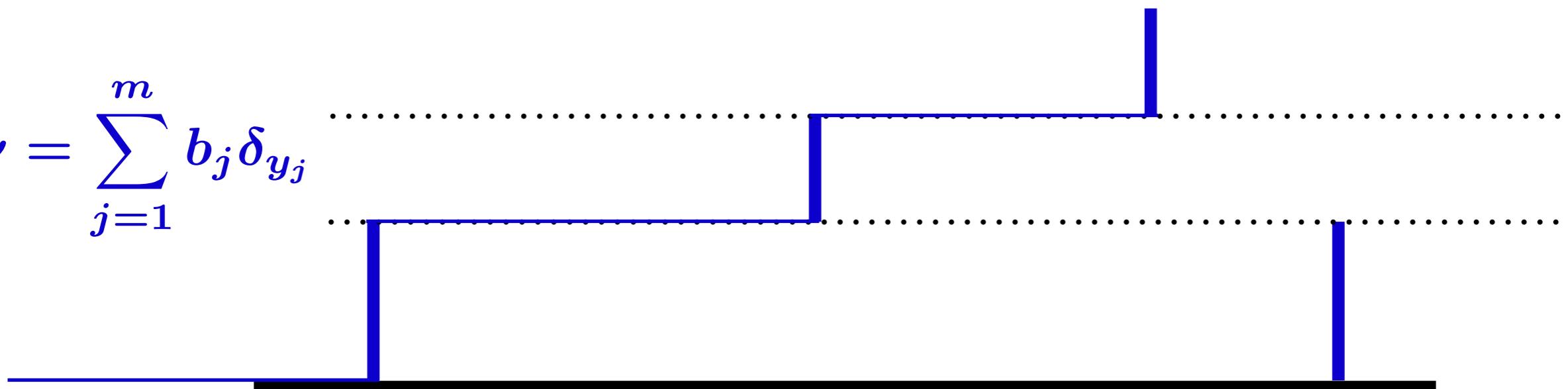


Building Emp. Distribution Functions

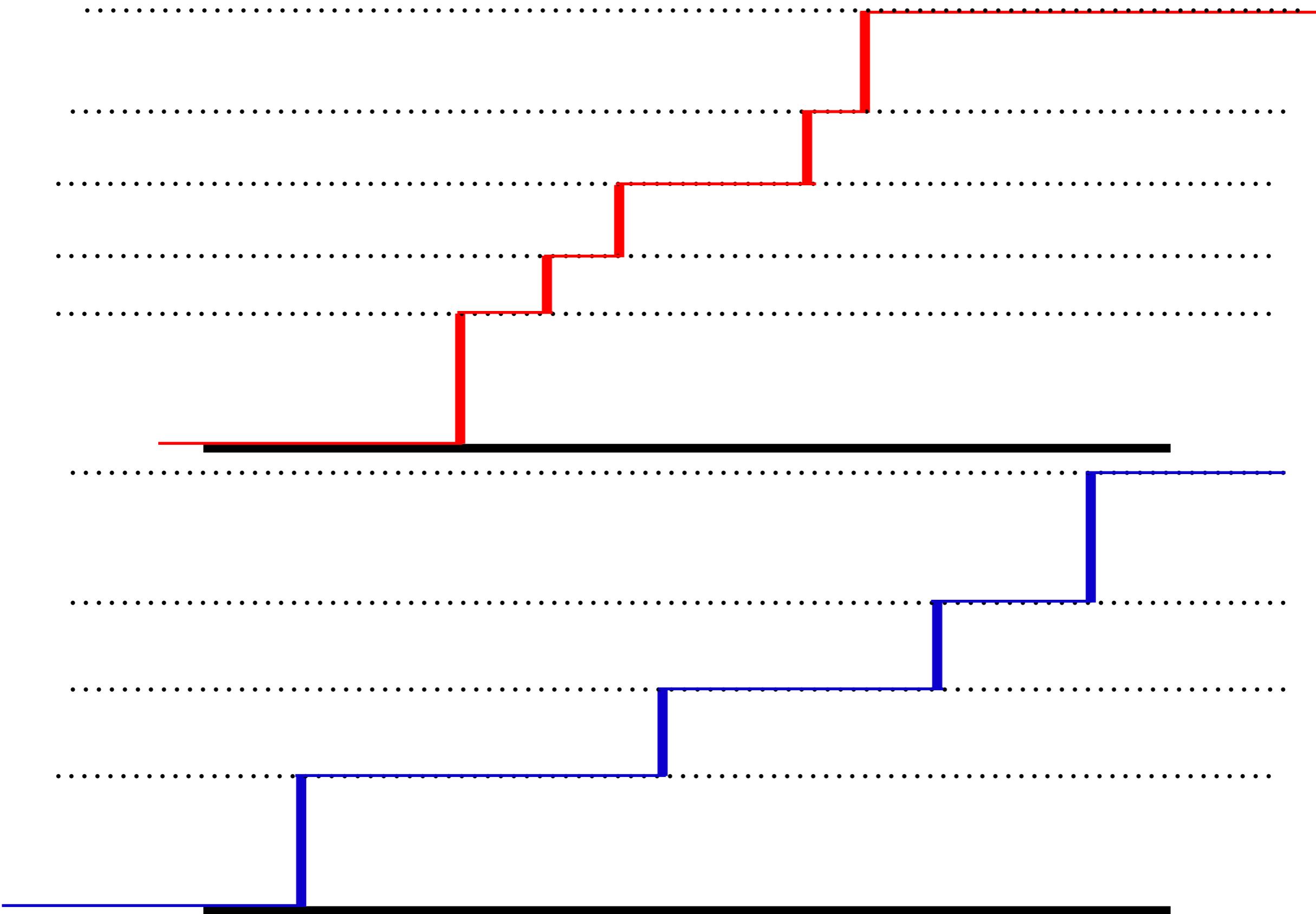
$$\mu = \sum_{i=1}^n a_i \delta_{x_i}$$



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

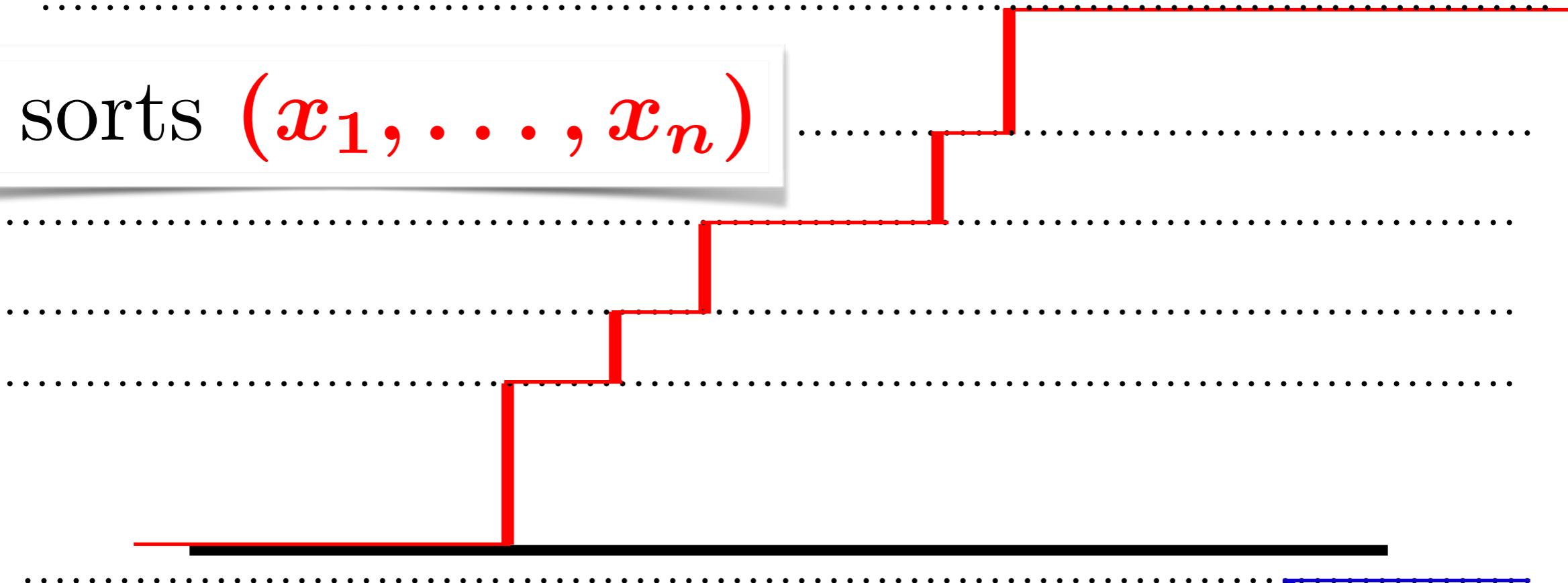


Building Emp. Distribution Functions

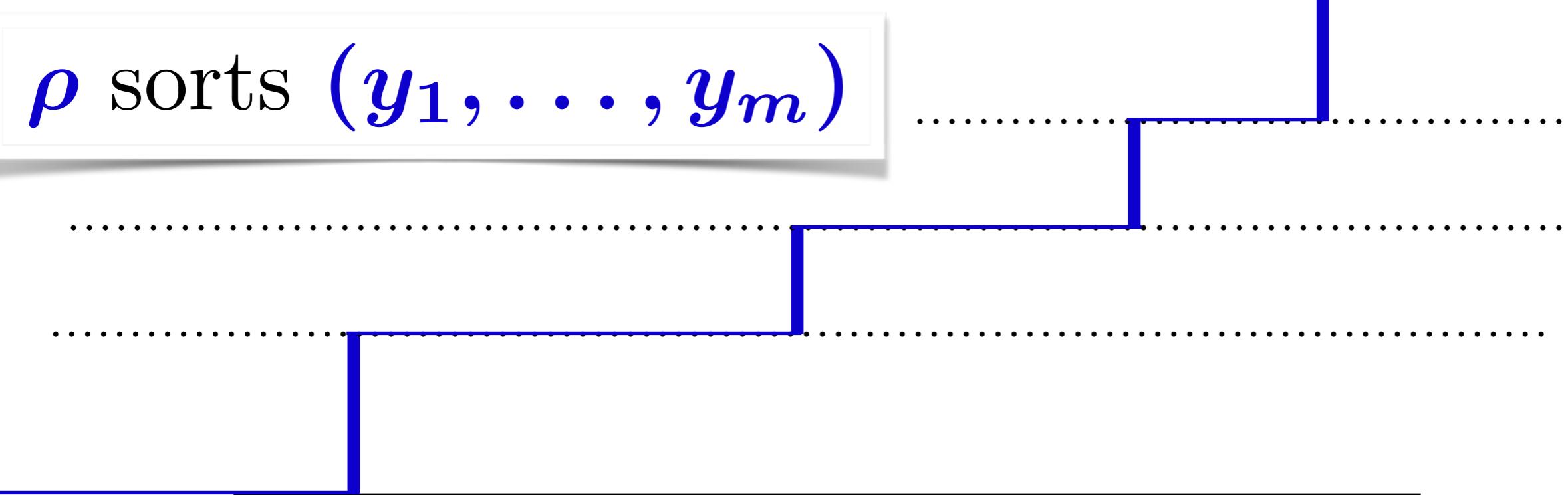


Building Emp. Distribution Functions

σ sorts (x_1, \dots, x_n)

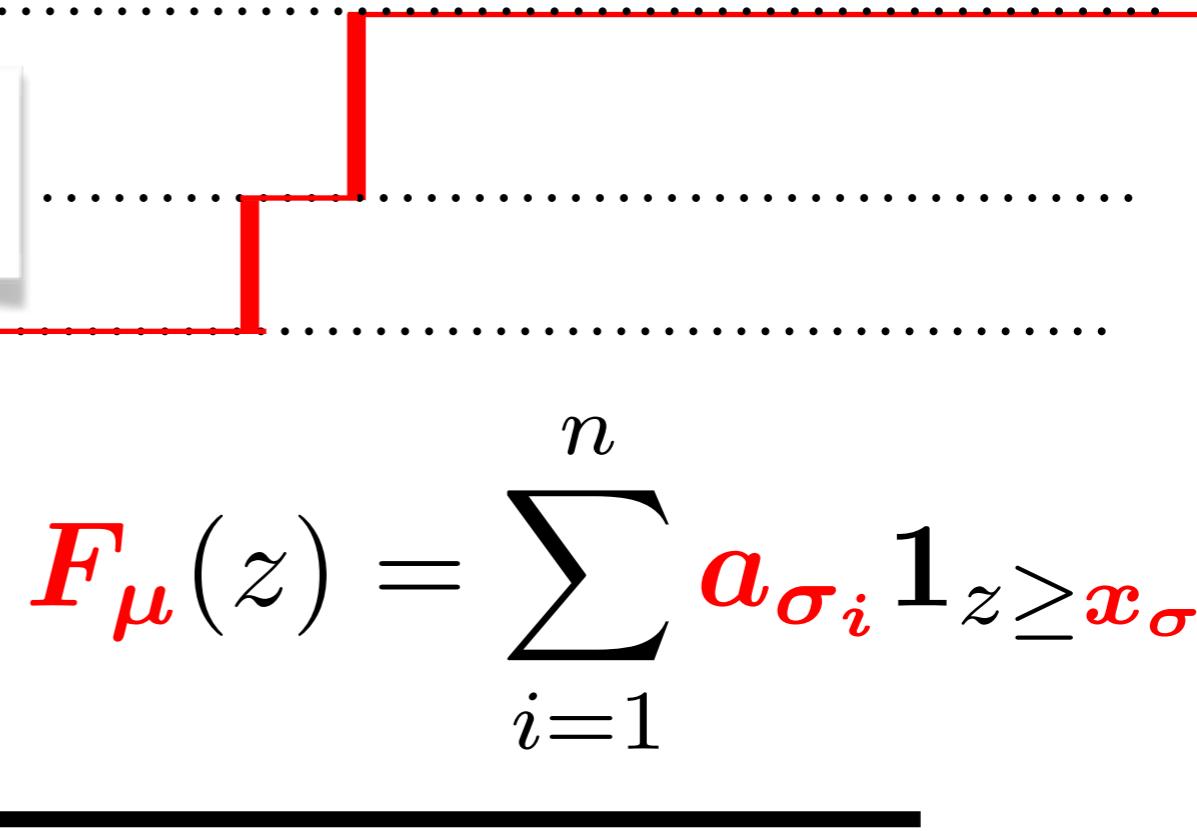


ρ sorts (y_1, \dots, y_m)

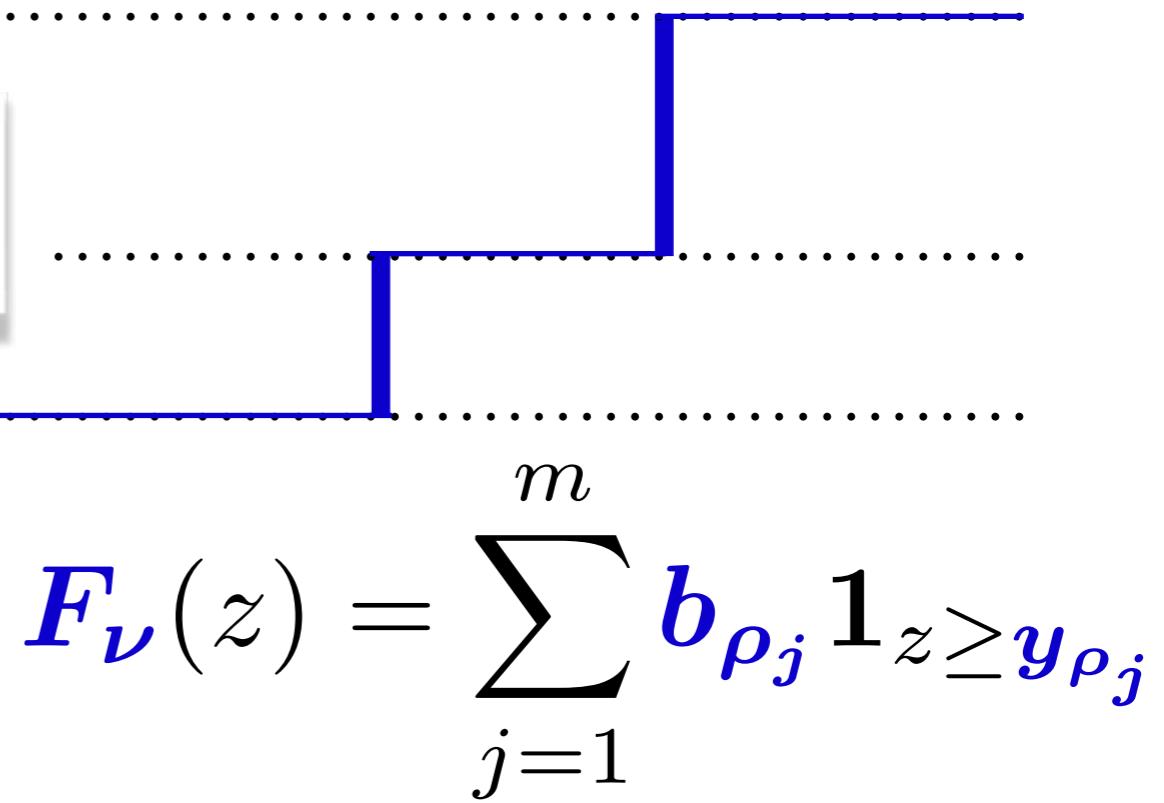


Building Emp. Distribution Functions

σ sorts (x_1, \dots, x_n)

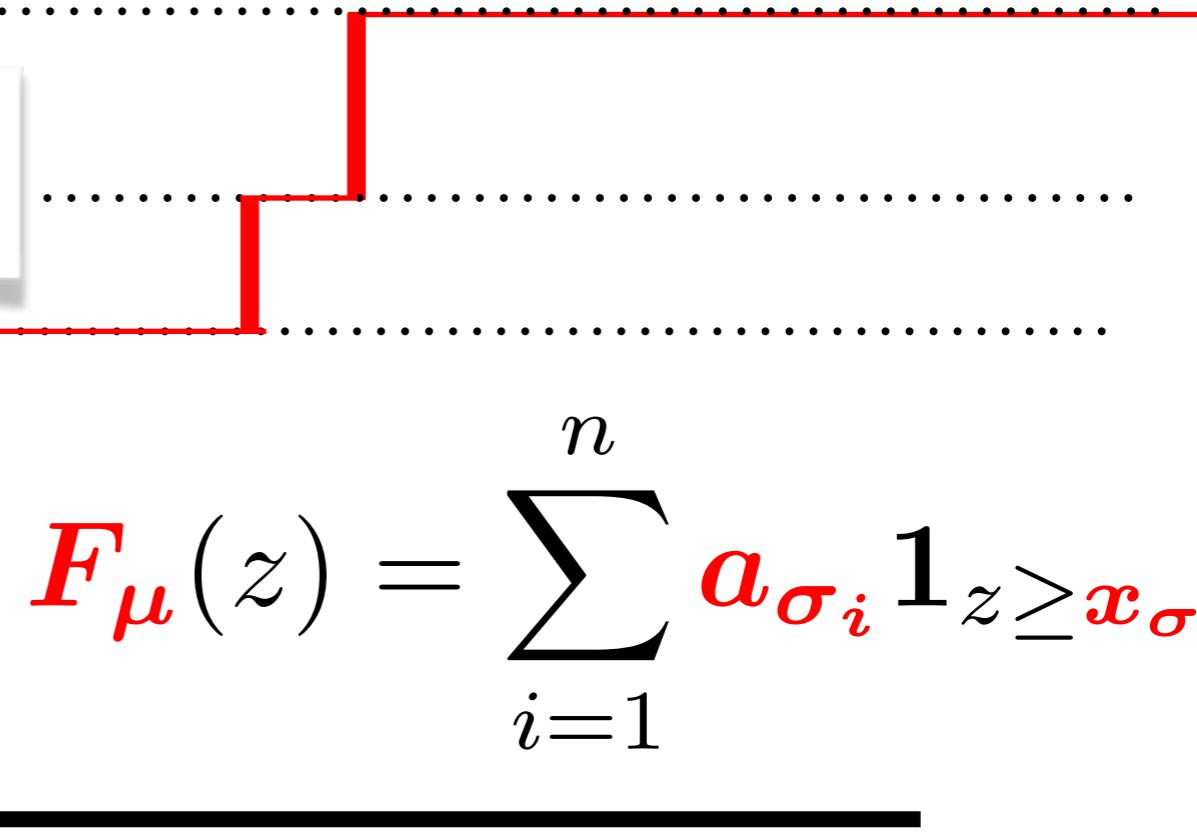


ρ sorts (y_1, \dots, y_m)

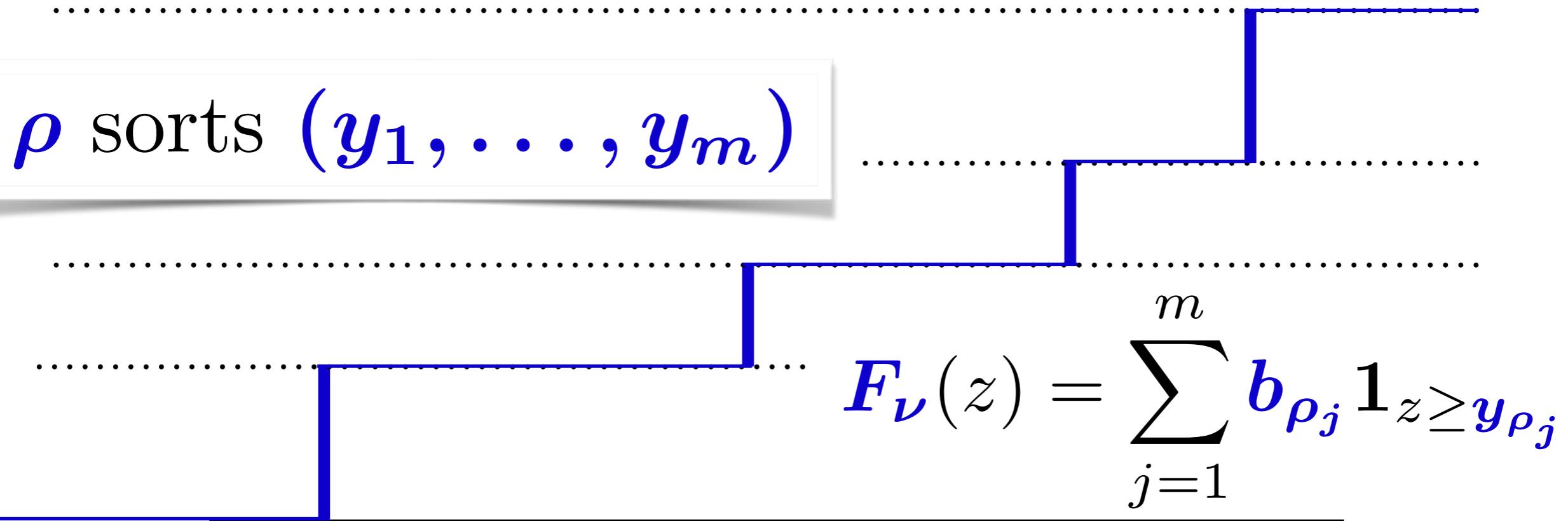


Emp. Distributions to Quantiles

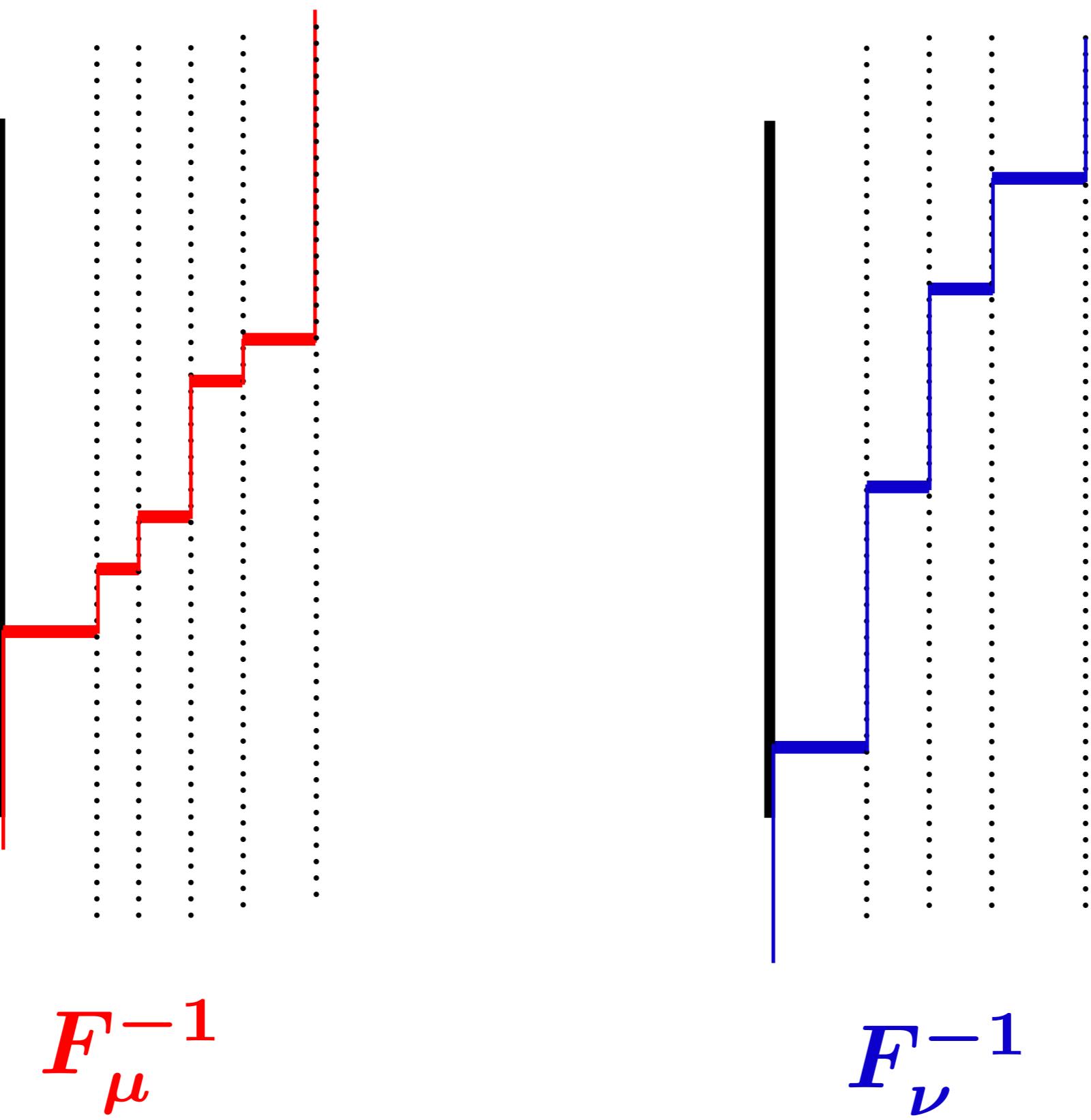
σ sorts (x_1, \dots, x_n)



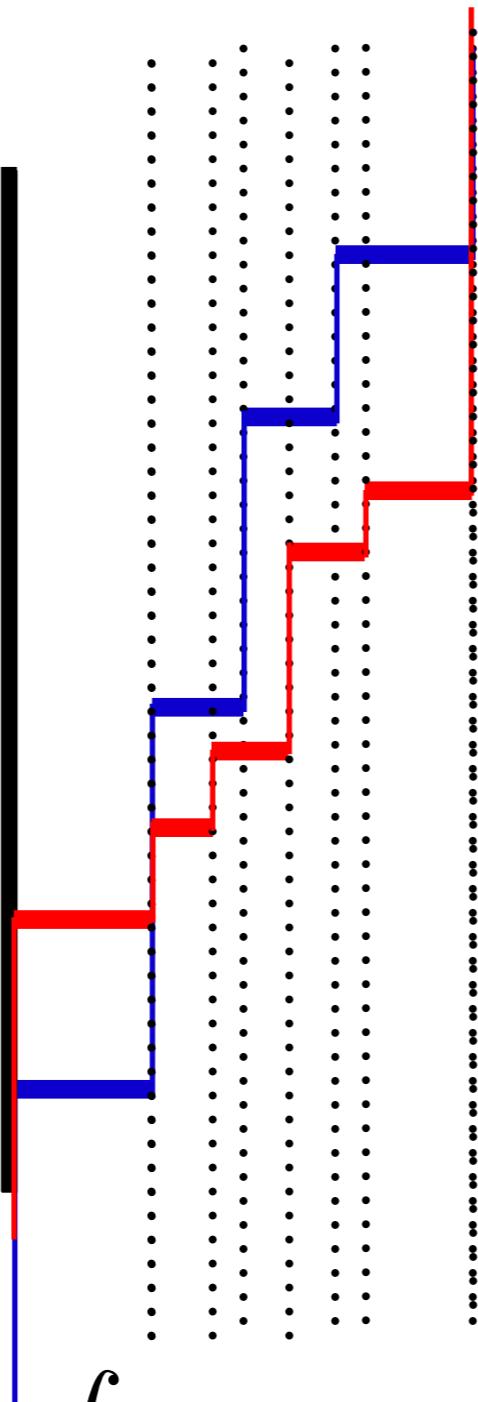
ρ sorts (y_1, \dots, y_m)



Emp. Distributions to Quantiles

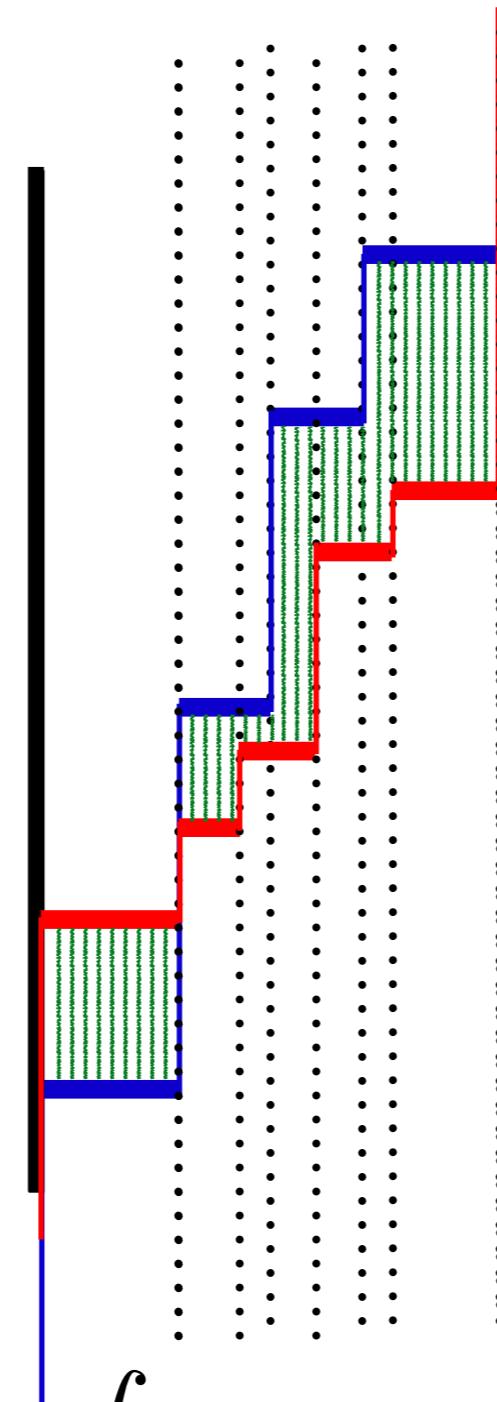


OT = Comparing Quantiles



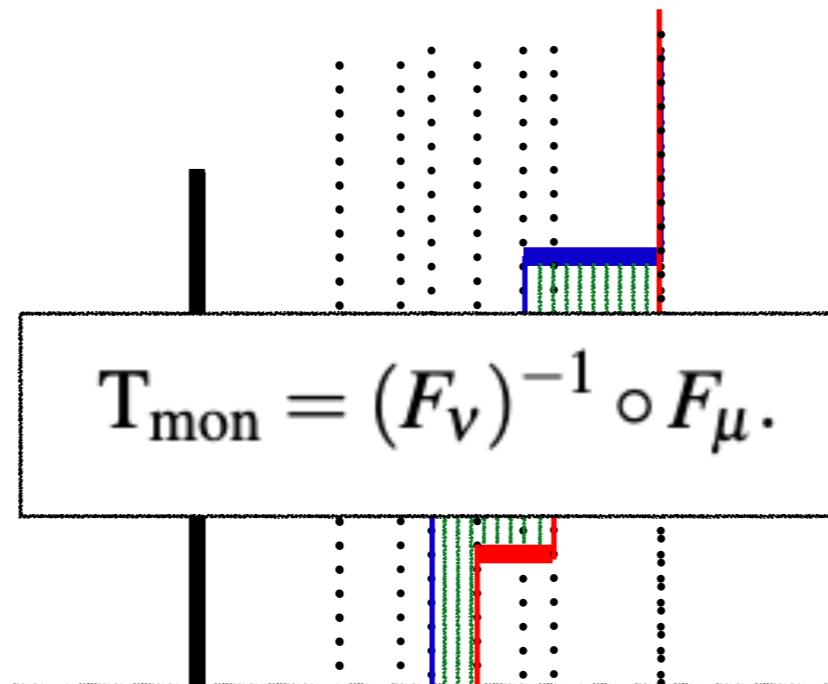
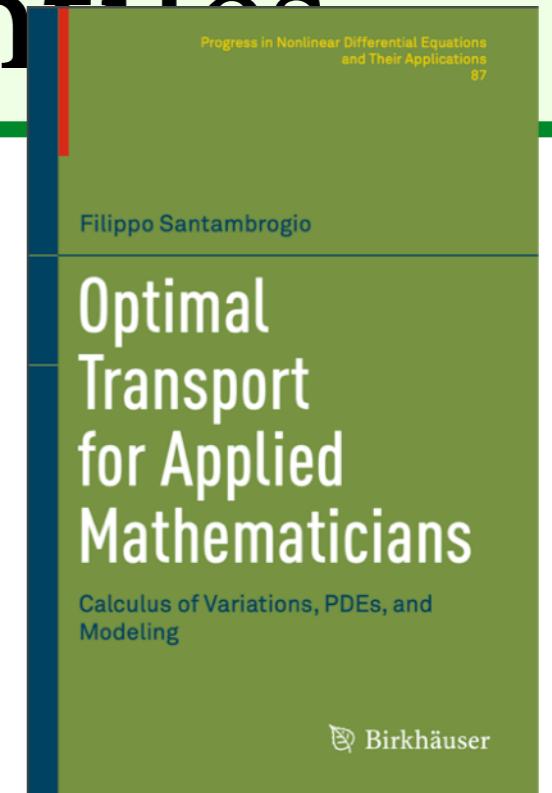
$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C}_{\mathbf{X}\mathbf{Y}} \rangle = \int_{u \in [0,1]} \mathbf{h}(F_\nu^{-1}(u) - F_\mu^{-1}(u)) du$$

OT = Comparing Quantiles



$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C}_{\mathbf{XY}} \rangle = \int_{u \in [0,1]} \mathbf{h}(F_\nu^{-1}(u) - F_\mu^{-1}(u)) du$$

OT = Comparing Quantiles



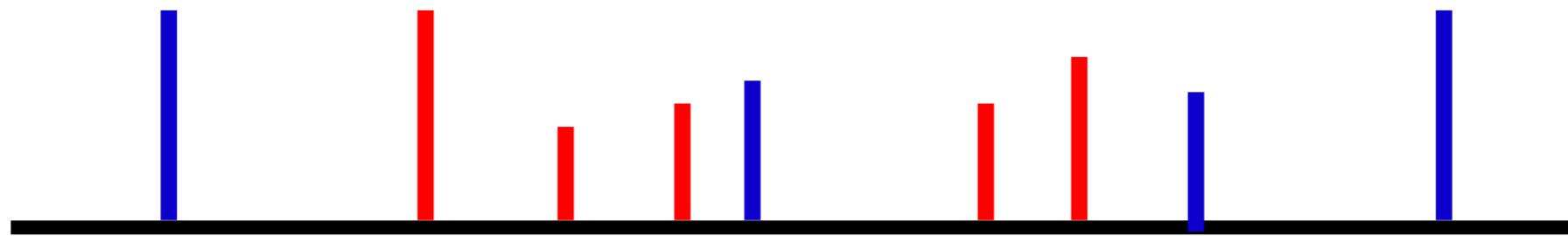
Theorem 2.9. Let $h : \mathbb{R} \rightarrow \mathbb{R}_+$ be a strictly convex function, and $\mu, \nu \in \mathcal{P}(\mathbb{R})$ be probability measures. Consider the cost $c(x, y) = h(y - x)$ and suppose that (KP) has a finite value. Then, (KP) has a unique solution, which is given by γ_{mon} . In the case where μ is atomless, this optimal plan is induced by the map T_{mon} .

$$\min_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C}_{\mathbf{XY}} \rangle = \int_{u \in [0,1]} h(F_\nu^{-1}(u) - F_\mu^{-1}(u)) du$$

More Explicit Link to Sorting

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

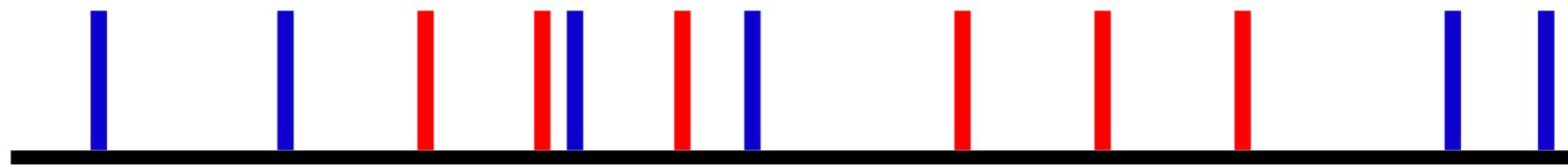
$$\nu = \sum_{j=1}^m b_j \delta_{y_j}$$



Link to Sorting: Uniform Measures

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

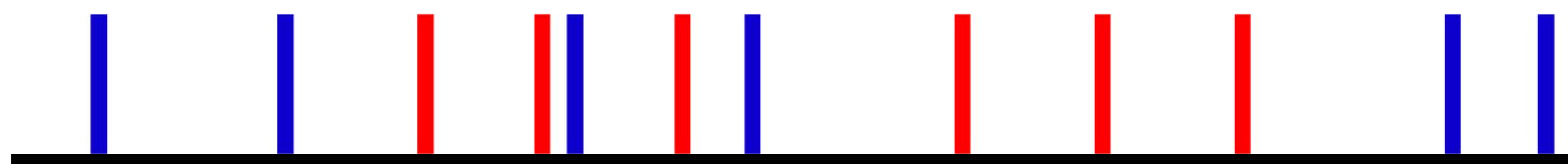
$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



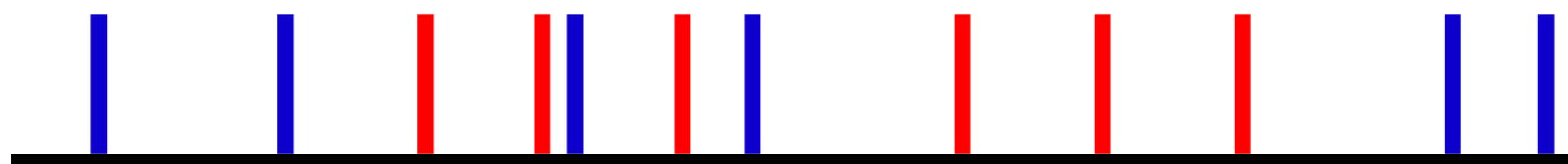
σ sorts (x_1, \dots, x_n)

ρ sorts (y_1, \dots, y_m)

Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



σ sorts (x_1, \dots, x_n)

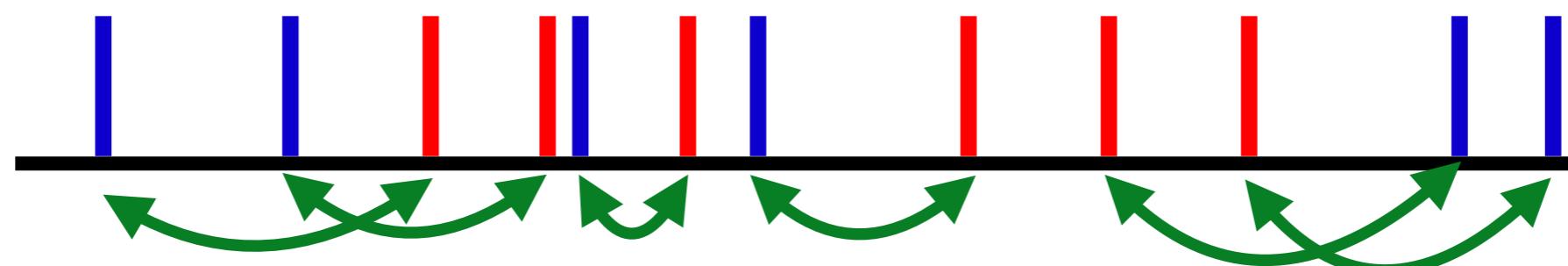
ρ sorts (y_1, \dots, y_m)

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle P, C_{\mathbf{X}\mathbf{Y}} \rangle = \sum_{i=1}^n h(y_{\rho_i} - x_{\sigma_i})$$

Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



σ sorts (x_1, \dots, x_n)

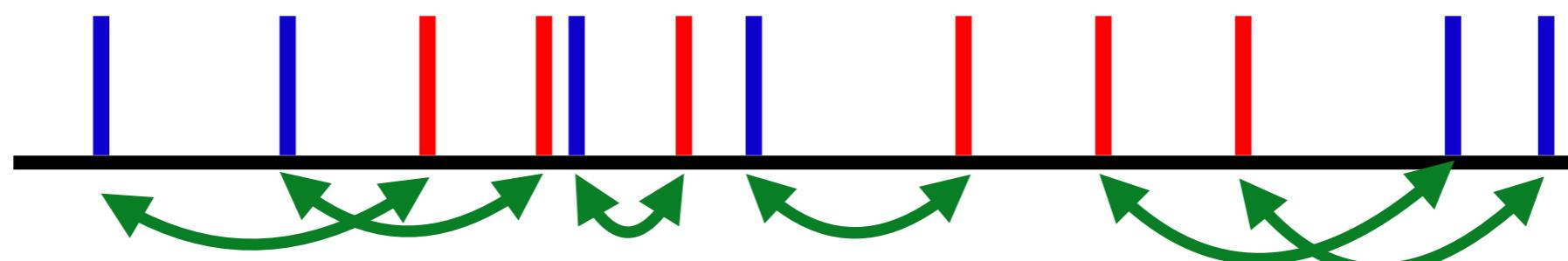
ρ sorts (y_1, \dots, y_m)

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle P, C_{\mathbf{XY}} \rangle = \sum_{i=1}^n h(y_{\rho_i} - x_{\sigma_i})$$

Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



σ sorts (x_1, \dots, x_n)

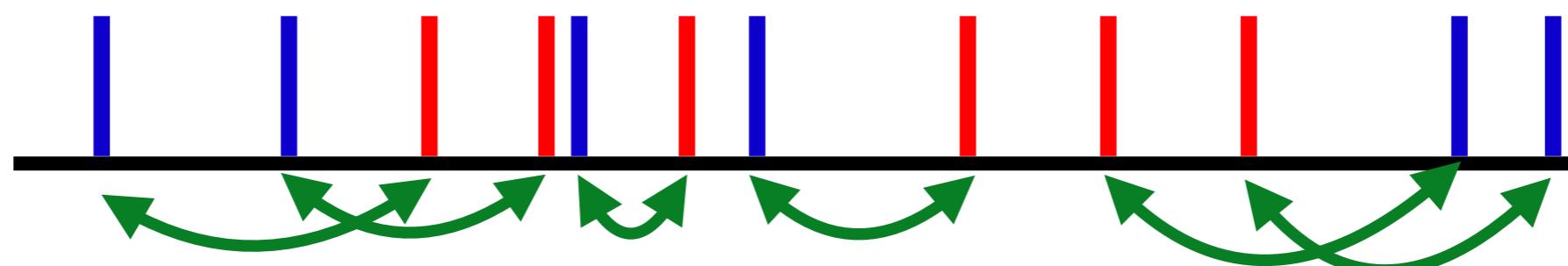
ρ sorts (y_1, \dots, y_m)

$$P^\star = \frac{1}{n} \text{sp1} ((\sigma_i, \rho_i)_i)$$

Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



σ sorts (x_1, \dots, x_n)

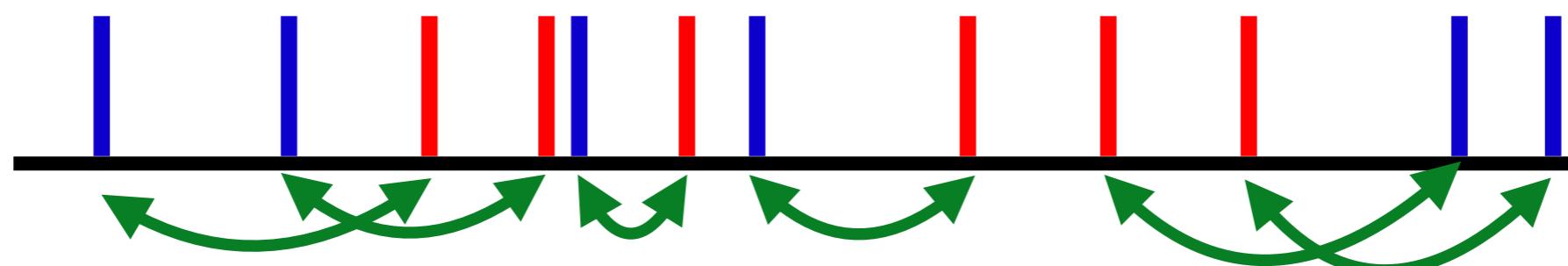
ρ sorts (y_1, \dots, y_m)

$$P^\star = \frac{1}{n} \text{sp1} ((\sigma_i, \rho_i)_i)$$

Link to Sorting: Uniform Measures

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_{y_i}$$



σ sorts (x_1, \dots, x_n)

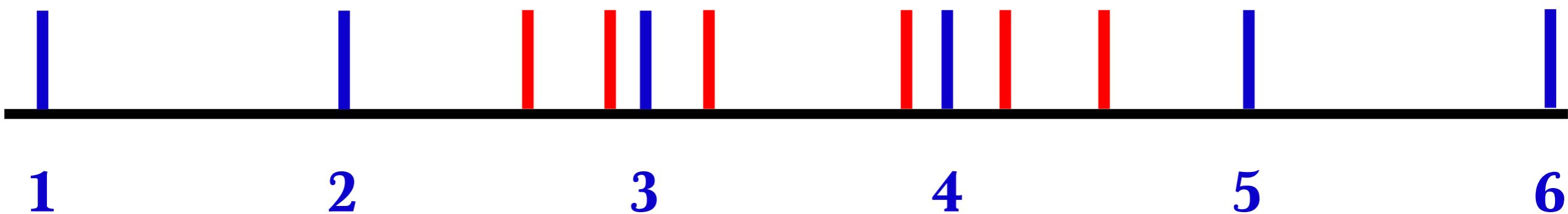
ρ sorts (y_1, \dots, y_m)

$$P^\star = \frac{1}{n} \text{sp1} \left((\sigma_i, \rho_i)_i \right)$$

OT towards a *sorted* sequence

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_i$$

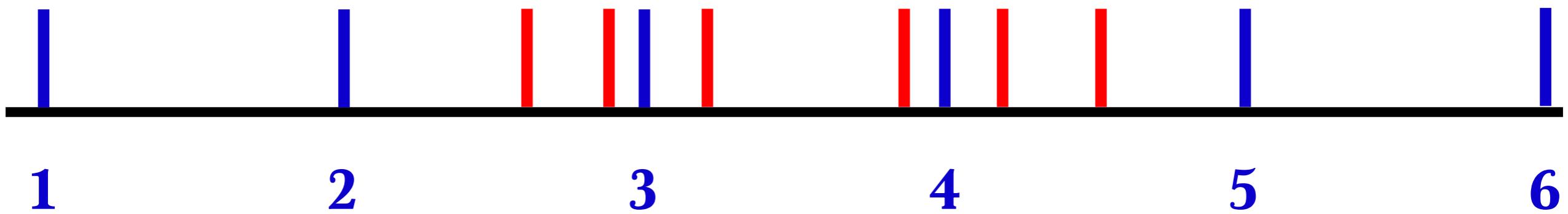


σ sorts (x_1, \dots, x_n)

OT towards a *sorted* sequence

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_i$$



σ sorts (x_1, \dots, x_n)

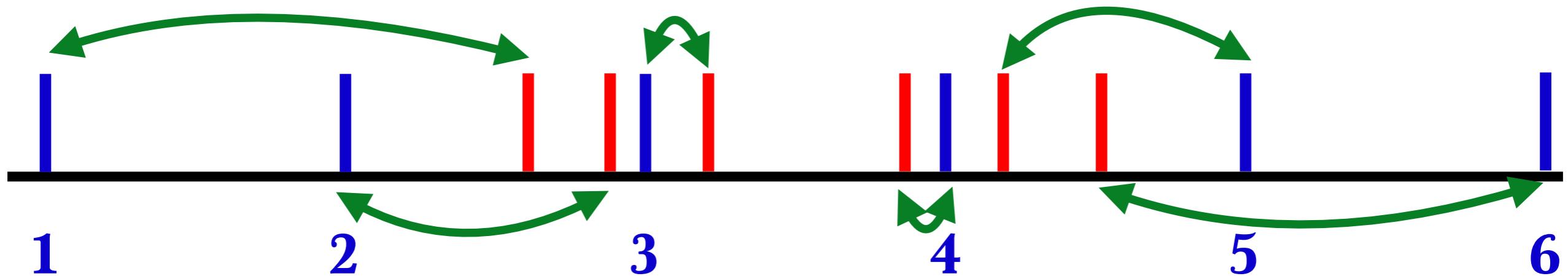
$\rho = \text{Id}$

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle P, C_{\mathbf{XY}} \rangle = \sum_{i=1}^n h(i - x_{\sigma_i})$$

OT towards a sorted sequence

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_i$$



σ sorts (x_1, \dots, x_n)

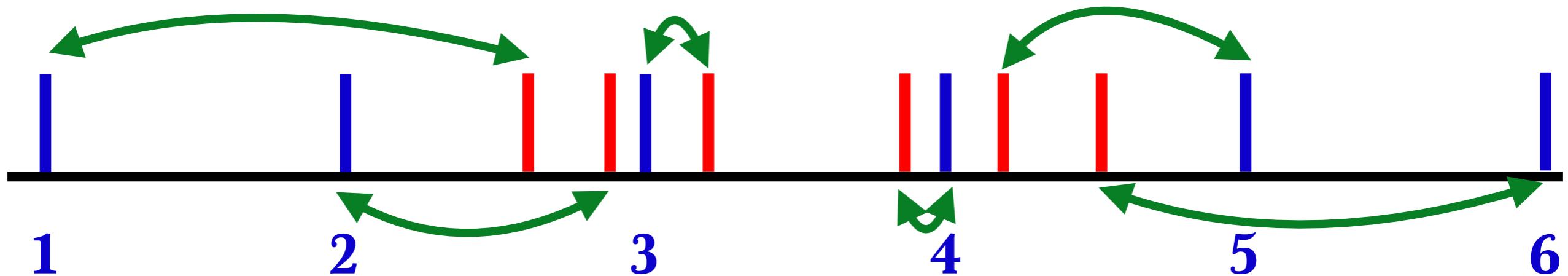
$\rho = \text{Id}$

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \langle P, C_{\mathbf{XY}} \rangle = \sum_{i=1}^n h(i - x_{\sigma_i})$$

OT towards a *sorted* sequence

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_i$$



σ sorts (x_1, \dots, x_n)

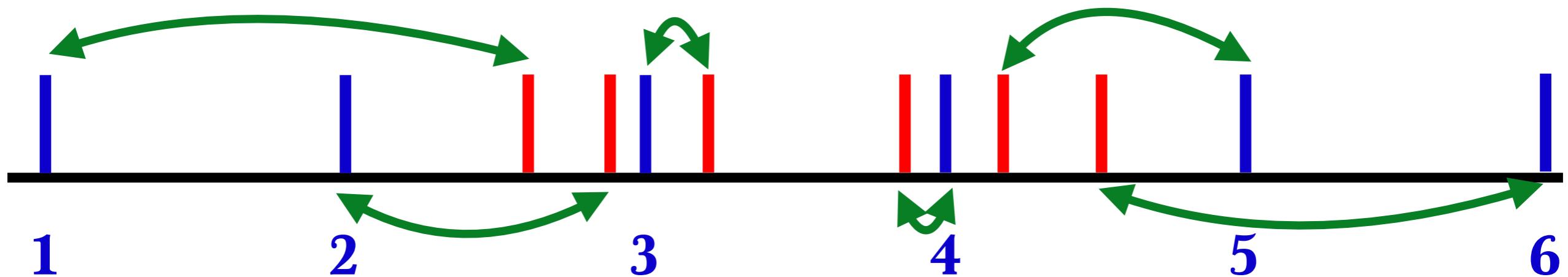
$\rho = \text{Id}$

$$P^\star = \frac{1}{n} \text{sp1} ((\sigma_i, i)_i)$$

OT towards a sorted sequence

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

$$\nu = \frac{1}{n} \sum_{i=1}^n \delta_i$$

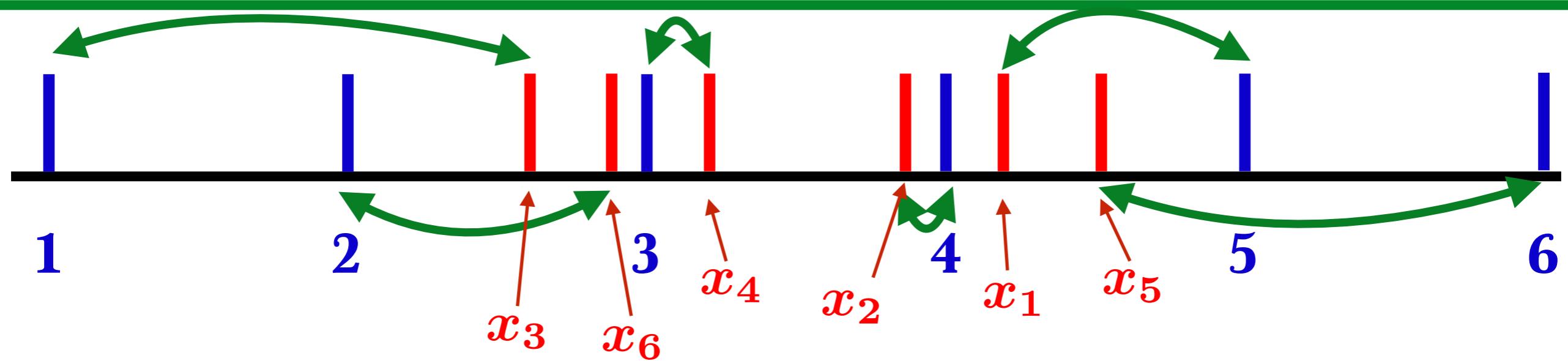


σ sorts (x_1, \dots, x_n)

$\rho = \text{Id}$

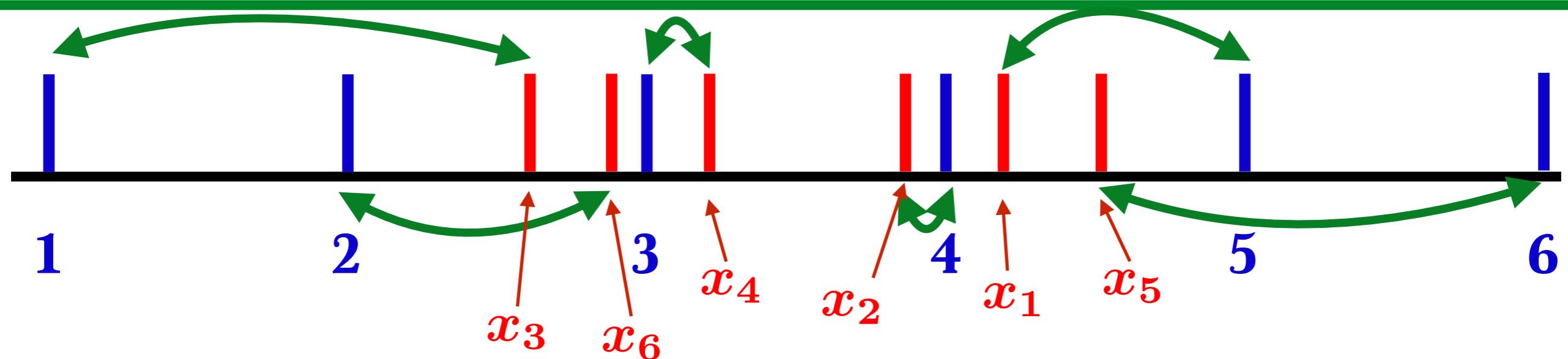
$$P^\star = \frac{1}{n} \text{sp1} ((\sigma_i, i)_i) = \frac{1}{n} \Pi_\sigma^T$$

OT towards a sorted sequence



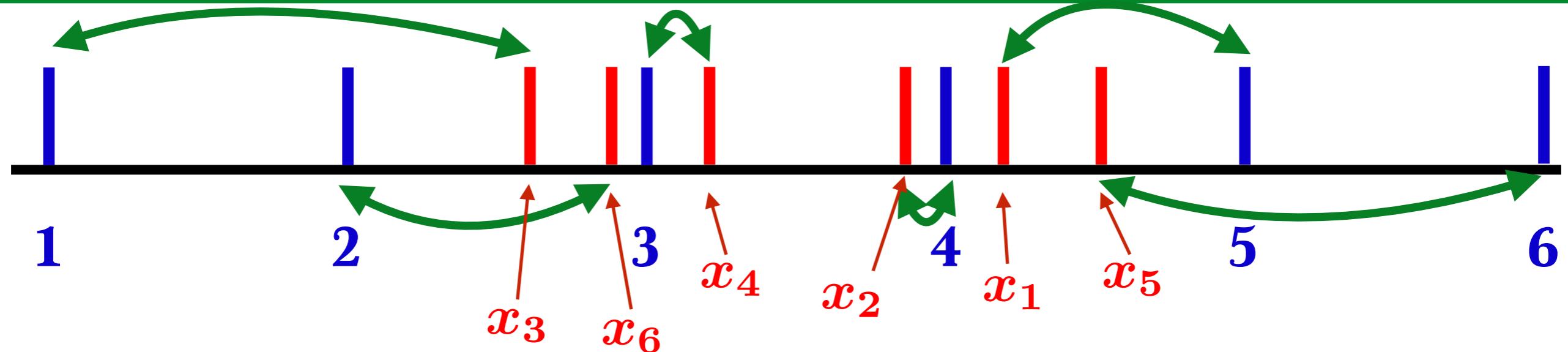
$$P^{\star} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix} = \frac{1}{n} \Pi_{\sigma(\mathbf{x})}^T = \frac{1}{n} \Pi_{\sigma(\mathbf{x})}^{-1}$$

OT towards a *sorted* sequence = sorting



$$P^{\star} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \left[\begin{array}{cccccc} & & & & & \frac{1}{n} \\ & & & & \frac{1}{n} & \\ \frac{1}{n} & & & & & \\ & & & & & \frac{1}{n} \\ & & \frac{1}{n} & & & \\ & & & & & \frac{1}{n} \\ & & & & & \end{array} \right] \end{matrix}$$

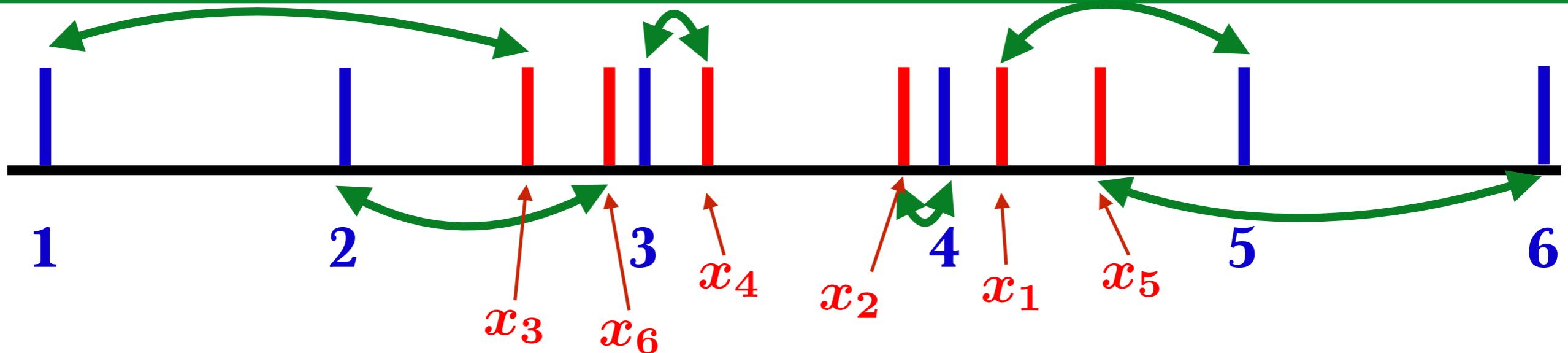
OT towards a sorted sequence = sorting



$$P^{\star} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{matrix} & \left[\begin{array}{cccccc} & & & & & \frac{1}{n} \\ \frac{1}{n} & & & & & \\ & & & & & \frac{1}{n} \\ & & & & & \\ & & & & & \frac{1}{n} \\ & & & & & \end{array} \right] \end{matrix}$$

$$R(\mathbf{x}) = \sigma(\mathbf{x})^{-1} = n^2 P^{\star} \begin{bmatrix} 1/6 \\ 2/6 \\ 3/6 \\ 4/6 \\ 5/6 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 1 \\ 3 \\ 6 \\ 2 \end{bmatrix}$$

OT towards a sorted sequence = sorting



$$\begin{aligned}
 P^* &= \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ x_1 & x_2 & x_3 & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\ x_2 & x_3 & x_4 & & & \\ x_3 & x_4 & x_5 & & & \\ x_4 & x_5 & x_6 & & & \\ x_5 & x_6 & & & & \end{matrix} \\
 R(\mathbf{x}) = \sigma(\mathbf{x})^{-1} &= n^2 P^* \begin{bmatrix} 1/6 \\ 2/6 \\ 3/6 \\ 4/6 \\ 5/6 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 1 \\ 3 \\ 6 \\ 2 \end{bmatrix} \\
 S(\mathbf{x}) &= n(P^*)^T \mathbf{x} = \begin{bmatrix} x_3 \\ x_6 \\ x_4 \\ x_2 \\ x_1 \\ x_5 \end{bmatrix}
 \end{aligned}$$

Two ways to rank and sort



Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,

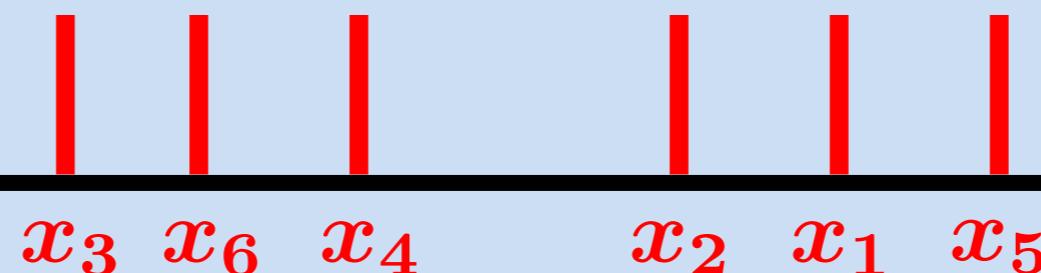


Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,



$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$



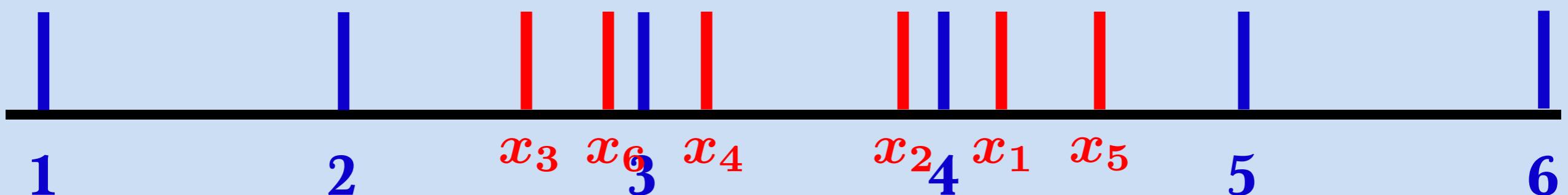
Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,

$$x_3 \xrightarrow{} x_6 \xrightarrow{} x_4 \xrightarrow{} x_2 \xrightarrow{} x_1 \xrightarrow{} x_5$$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$

... Set first Milestones in the race ...

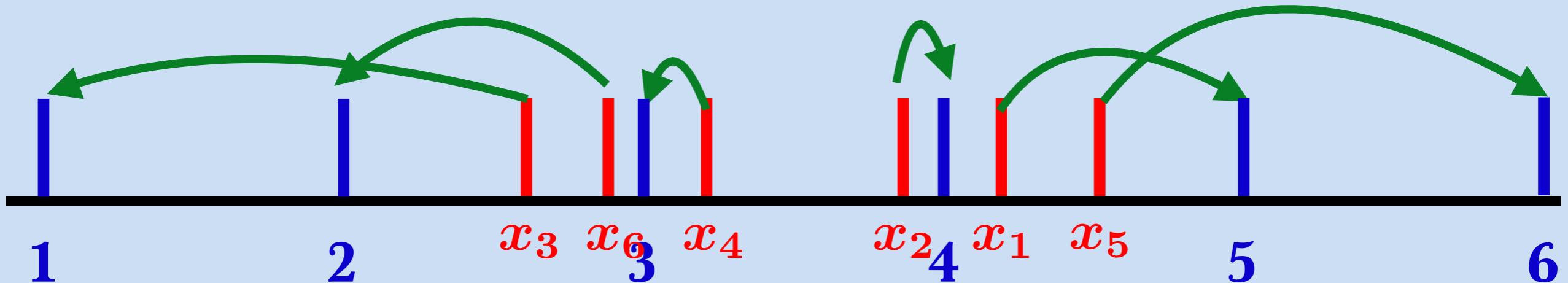


Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,

$$x_3 \xleftarrow{} x_6 \xleftarrow{} x_4 \xleftarrow{} x_2 \xrightarrow{} x_1 \xrightarrow{} x_5$$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$



Compute OT solution P^* ,

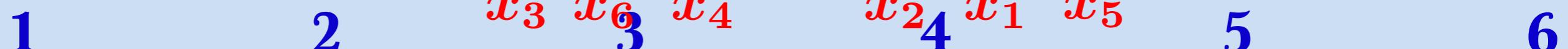
Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,

$$x_3 \xleftarrow{} x_6 \xleftarrow{} x_4 \xleftarrow{} x_2 \xrightarrow{} x_1 \xrightarrow{} x_5$$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$

Compute OT solution P^* ,



$$R(\mathbf{x}) = n^2 P^* \begin{bmatrix} 1/6 \\ \vdots \\ 6/6 \end{bmatrix}, \quad S(\mathbf{x}) = n(P^*)^T \mathbf{x}$$

Two ways to rank and sort

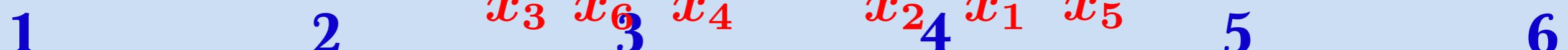
Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,



$O(n \log n)$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$

Compute OT solution \mathbf{P}^* ,



$$R(\mathbf{x}) = n^2 \mathbf{P}^* \begin{bmatrix} 1/6 \\ \vdots \\ 6/6 \end{bmatrix}, S(\mathbf{x}) = n (\mathbf{P}^*)^T \mathbf{x}$$

Two ways to rank and sort

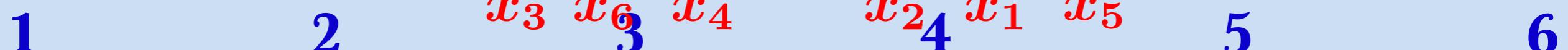
Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,



$O(n \log n)$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$

Compute OT solution P^* ,



$R(\mathbf{x}) = n^2 P^* \begin{bmatrix} 1/6 \\ \vdots \\ 6/6 \end{bmatrix}$, $S(\mathbf{x}) = n(P^*)^T \mathbf{x}$



$O(n^3 \log n)$

Two ways to rank and sort

Compute $\sigma(\mathbf{x}) = (3, 6, 4, 2, 1, 5)$,



$O(n \log n)$

$R(\mathbf{x}) = (5, 4, 1, 3, 6, 2)$, $S(\mathbf{x}) = (x_3, x_6, x_4, x_2, x_1, x_5)$

Compute OT solution P^* ,



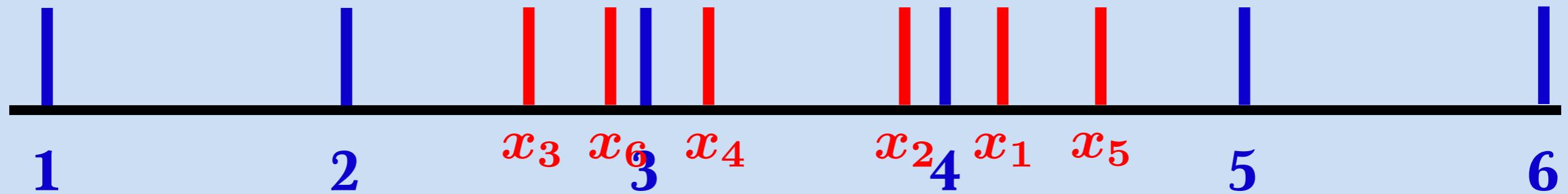
$R(\mathbf{x}) = n^2 P^*$, $S(\mathbf{x}) = n(P^*)^T \mathbf{x}$

$$\begin{bmatrix} 1/6 \\ \vdots \\ 6/6 \end{bmatrix}$$



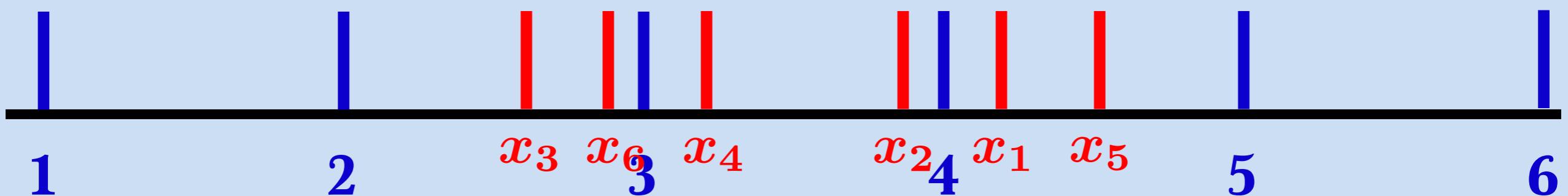
$O(n^3 \log n)$

Generalized Sorting and Ranking

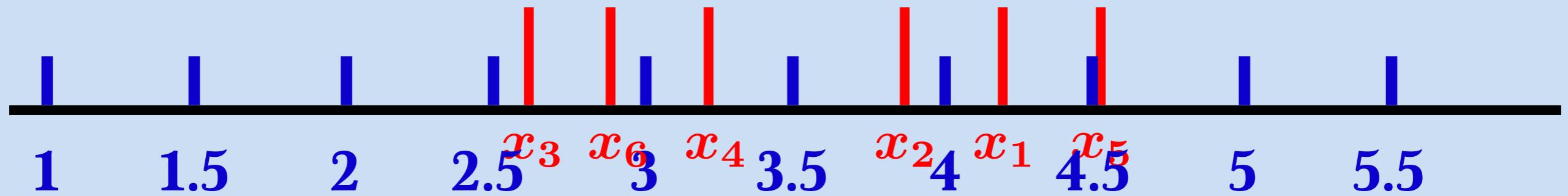


Generalized Sorting and Ranking

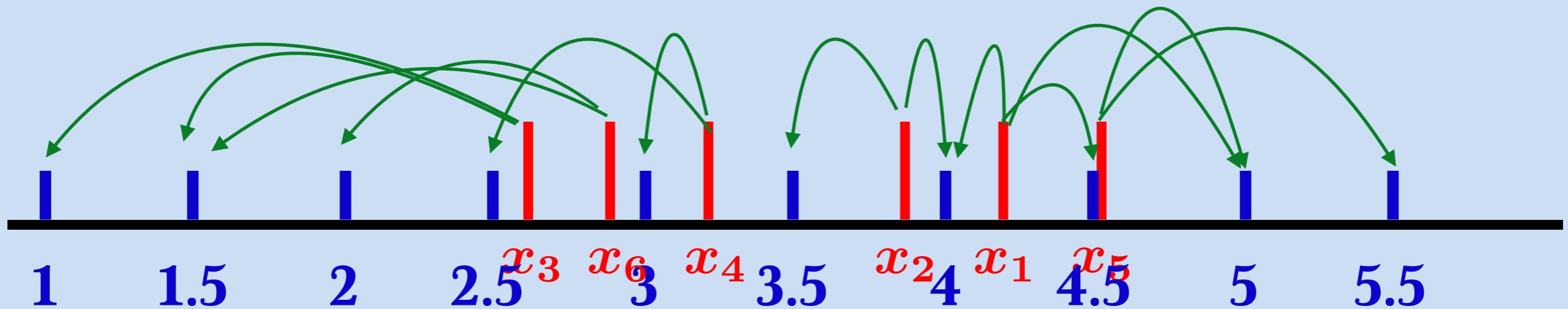
Interesting possibility: set a different number m of Milestones



Generalized Sorting and Ranking



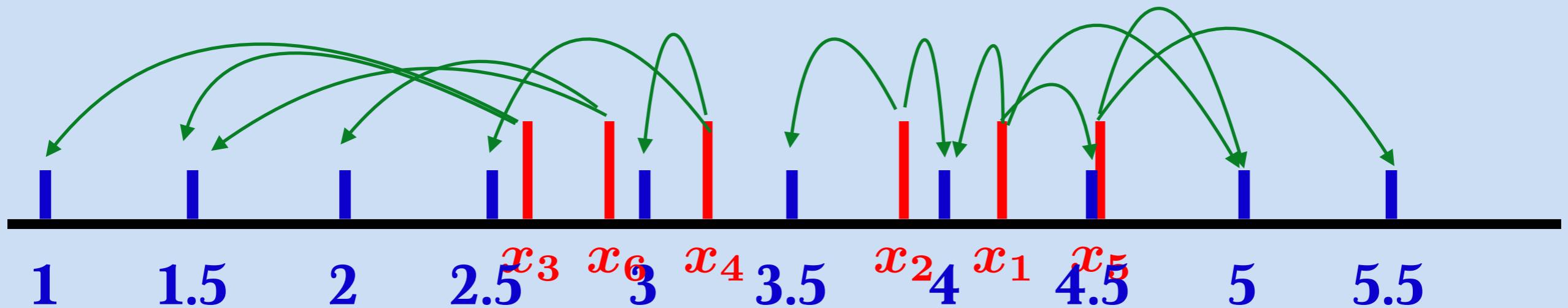
Generalized Sorting and Ranking



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \begin{bmatrix} 1/m \\ \vdots \\ m/m \end{bmatrix}, \quad S(\mathbf{x}) = m (\mathbf{P}^*)^T \mathbf{x}$$

Generalized Sorting and Ranking

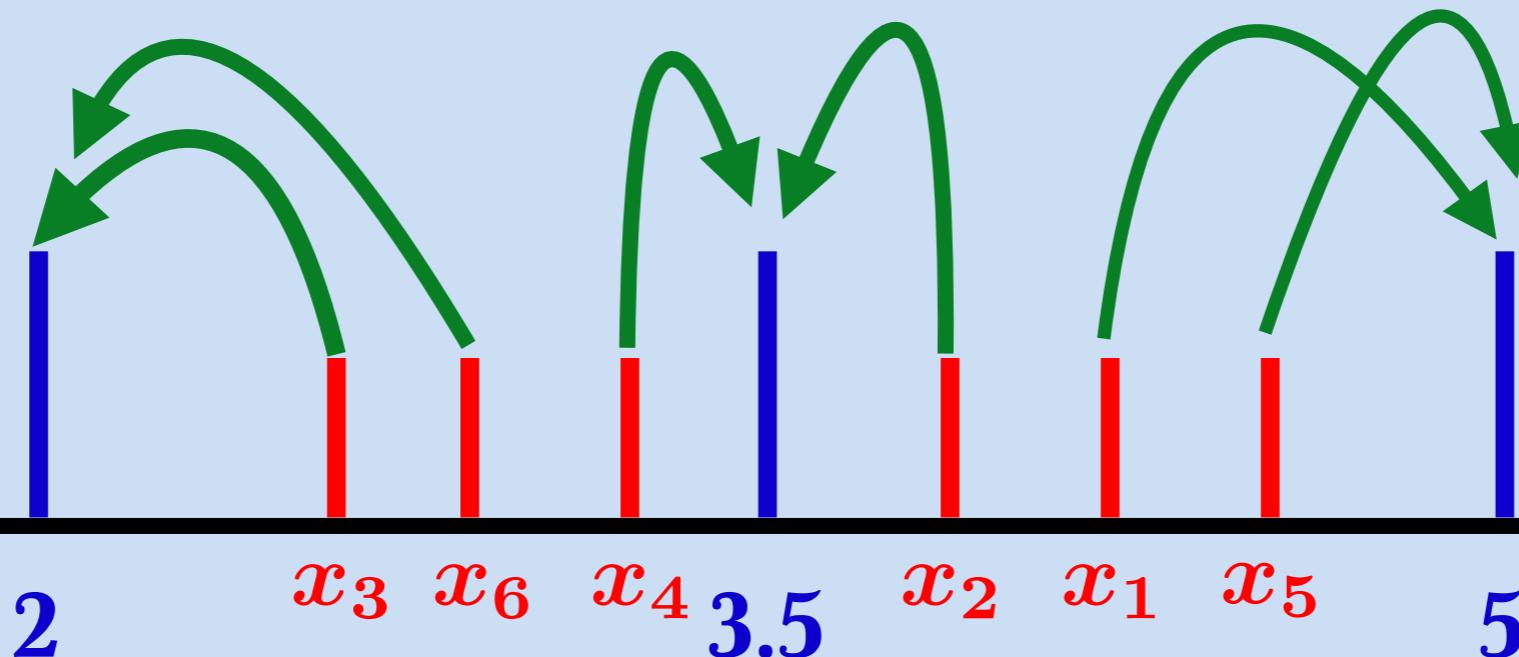


Compute $n \times m$ OT solution P^* ,

$$R(\mathbf{x}) = n^2 P^* \begin{bmatrix} 1/m \\ \vdots \\ m/m \end{bmatrix}, \quad S(\mathbf{x}) = m(P^*)^T \mathbf{x}$$

$\in \mathbb{R}^n$ $\in \mathbb{R}^m$

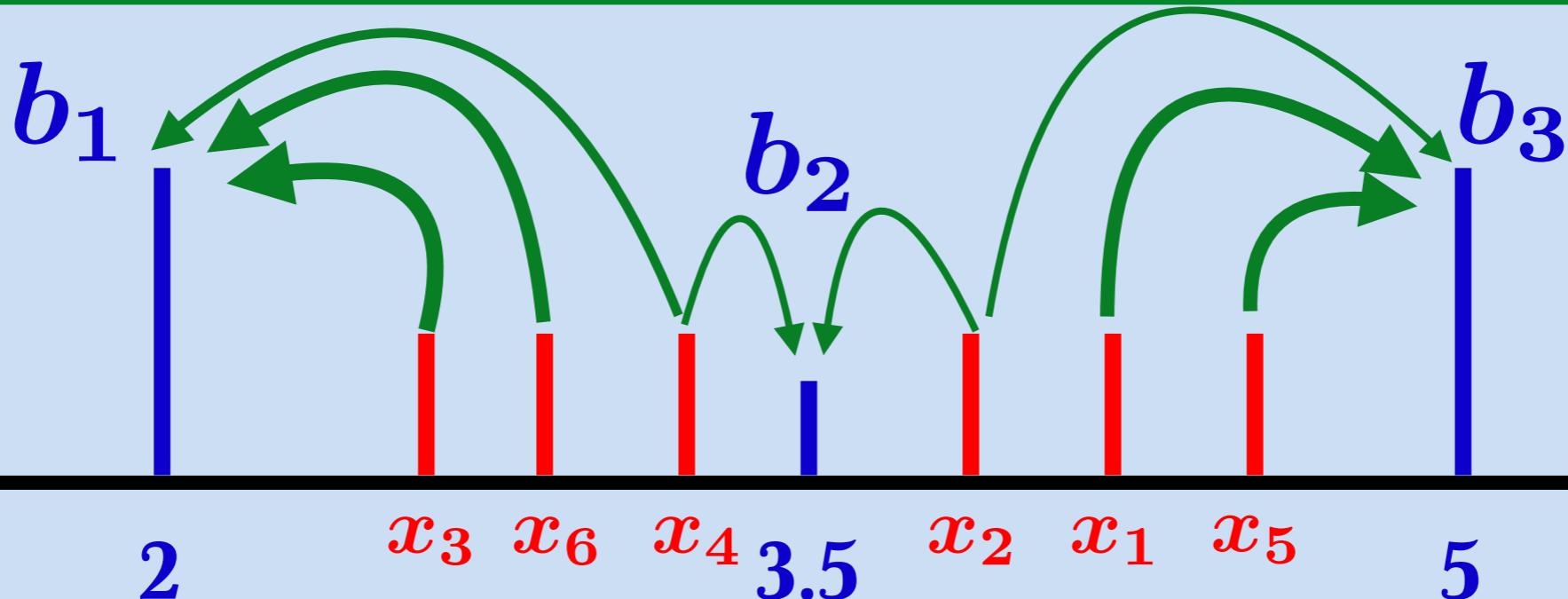
Less Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \begin{bmatrix} 1/m \\ \vdots \\ m/m \end{bmatrix}, \quad S(\mathbf{x}) = \mathbf{m} (\mathbf{P}^*)^T \mathbf{x}$$

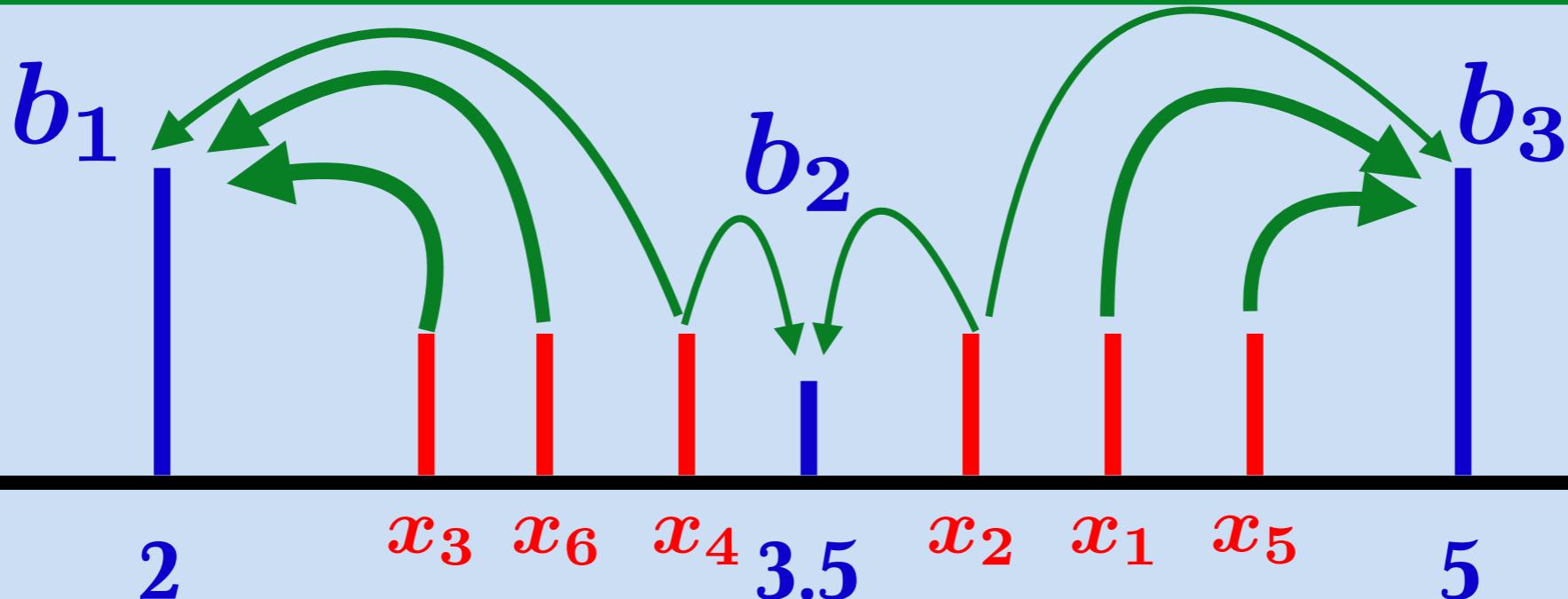
Weighted Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \begin{bmatrix} 1/m \\ \vdots \\ m/m \end{bmatrix}, \quad S(\mathbf{x}) = \mathbf{m} (\mathbf{P}^*)^T \mathbf{x}$$

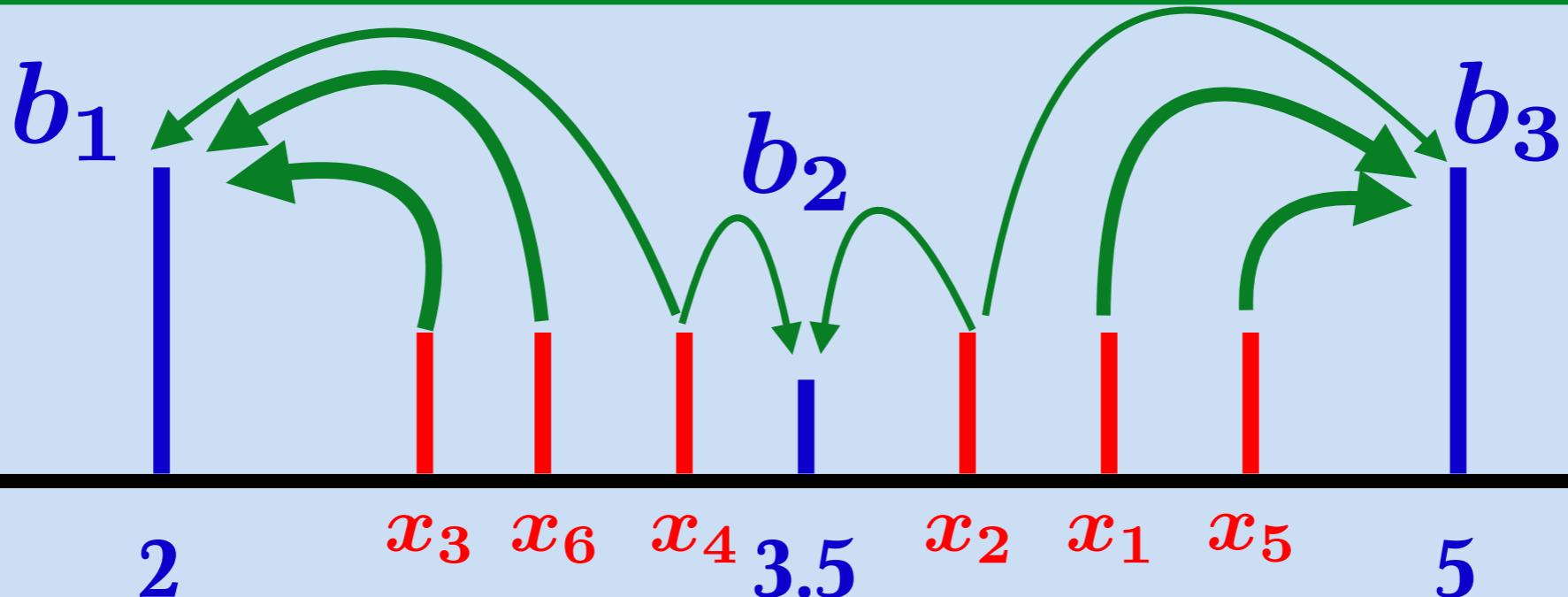
Weighted Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \begin{bmatrix} 1/m \\ \vdots \\ m/m \end{bmatrix}, \quad S(\mathbf{x}) = \mathbf{m} (\mathbf{P}^*)^T \mathbf{x}$$

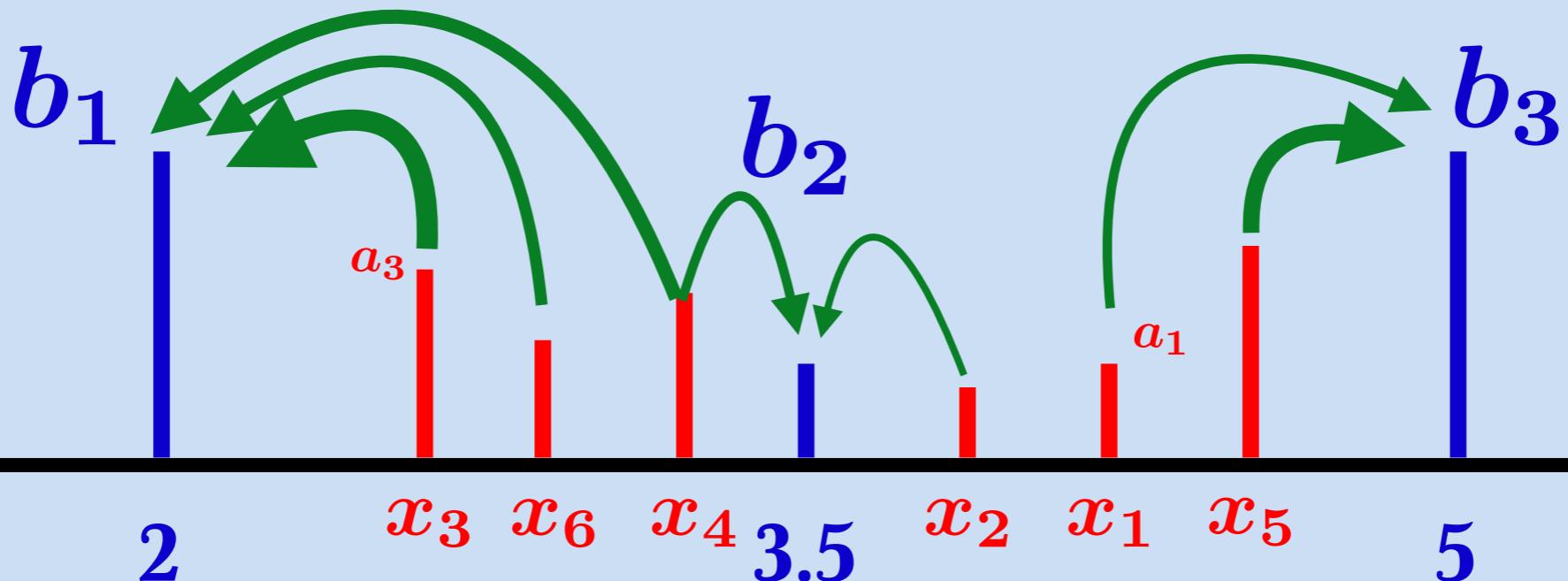
Weighted Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \text{cs}(\mathbf{b}), \quad S(\mathbf{x}) = \mathbf{b}^{-1} \circ (\mathbf{P}^*)^T \mathbf{x}$$

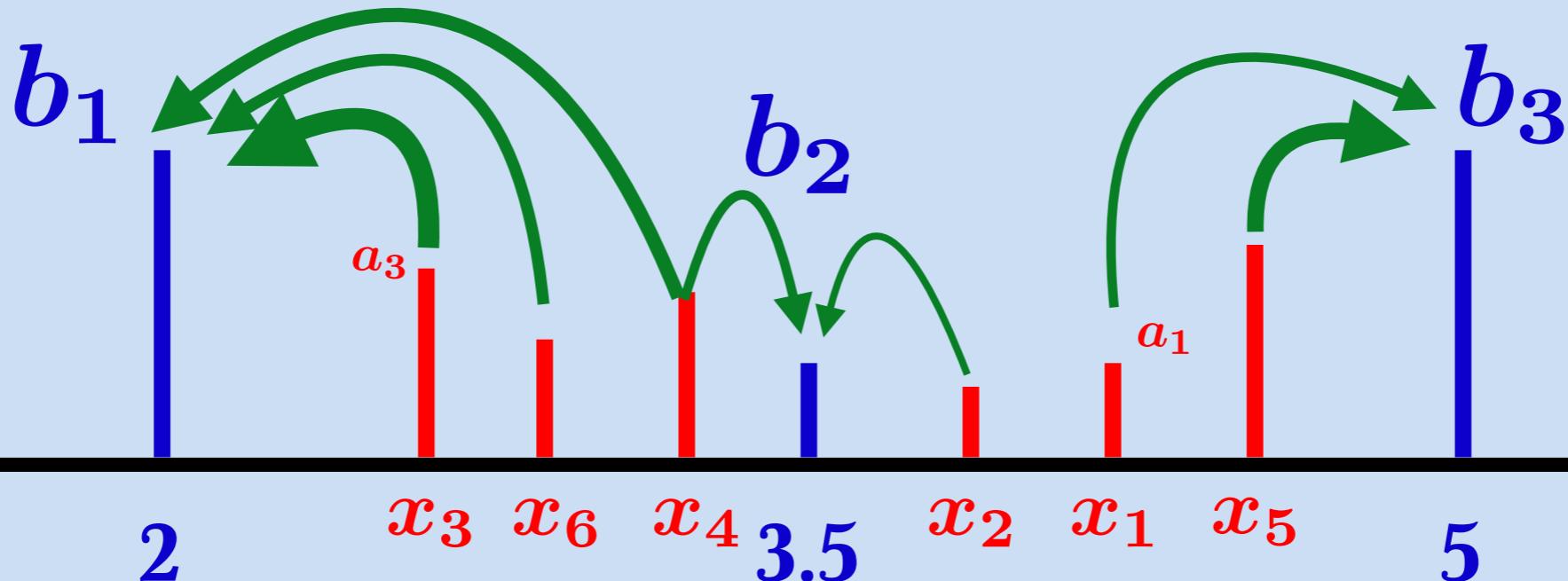
Weighted Inputs and Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \text{cs}(\mathbf{b}), S(\mathbf{x}) = \mathbf{b}^{-1} \circ (\mathbf{P}^*)^T \mathbf{x}$$

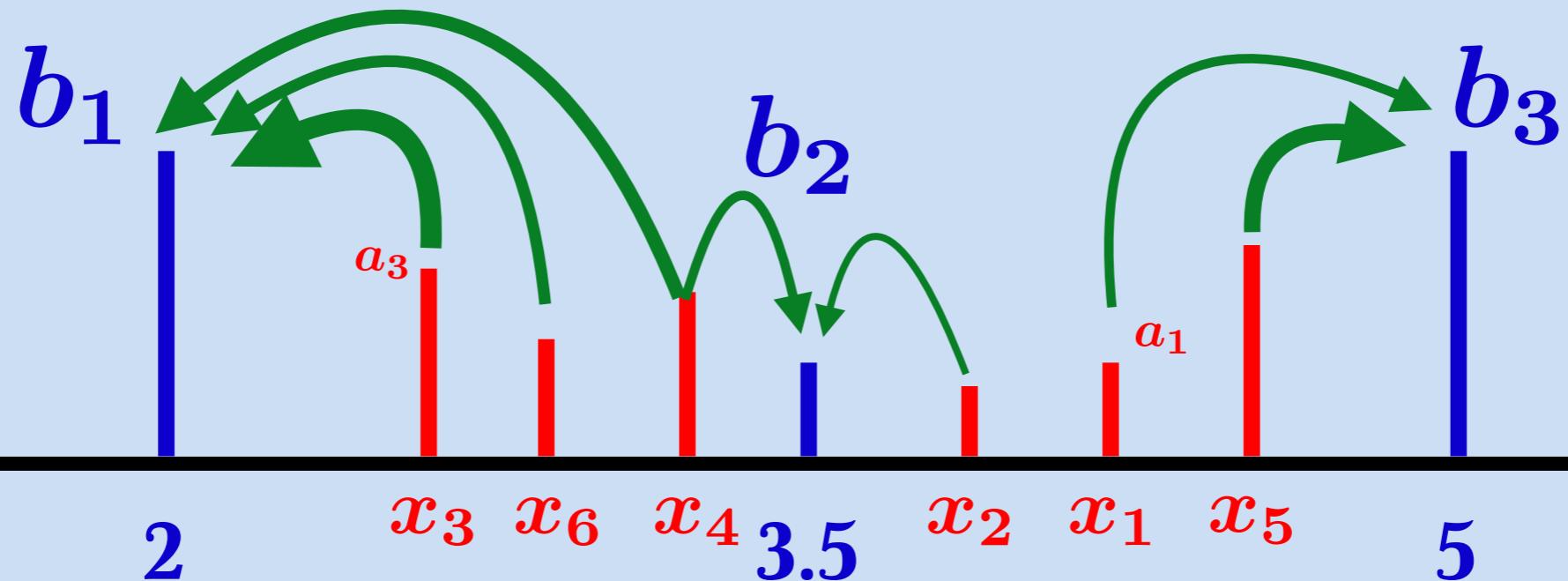
Weighted Inputs and Milestones



Compute $n \times m$ OT solution \mathbf{P}^* ,

$$R(\mathbf{x}) = n^2 \mathbf{P}^* \text{cs}(\mathbf{b}), \quad S(\mathbf{x}) = \mathbf{b}^{-1} \circ (\mathbf{P}^*)^T \mathbf{x}$$

Weighted Inputs and Milestones



Compute $n \times m$ OT solution P^* ,

$$R(\mathbf{x}) = n \mathbf{a}^{-1} \circ P^* \text{cs}(\mathbf{b}), \quad S(\mathbf{x}) = \mathbf{b}^{-1} \circ (P^*)^T \mathbf{x}$$

Weighted Inputs and Milestones

Compute $\textcolor{red}{n} \times \textcolor{blue}{m}$ OT solution $\textcolor{red}{P}^*$,

$$R(\textcolor{red}{x}) = \textcolor{red}{n} \textcolor{red}{a}^{-1} \circ \textcolor{red}{P}^* \text{cs}(\textcolor{blue}{b}), S(\textcolor{red}{x}) = \textcolor{blue}{b}^{-1} \circ (\textcolor{red}{P}^*)^T \textcolor{red}{x}$$

Weighted Inputs and Milestones

Compute $\textcolor{red}{n} \times \textcolor{blue}{m}$ OT solution $\textcolor{red}{P}^*$,

$$R(\textcolor{red}{x}) = \textcolor{red}{n} \textcolor{black}{a}^{-1} \circ \textcolor{red}{P}^* \text{cs}(\textcolor{blue}{b}), S(\textcolor{red}{x}) = \textcolor{blue}{b}^{-1} \circ (\textcolor{red}{P}^*)^T \textcolor{red}{x}$$

All issues do remain however !

- (1) Still not differentiable *w.r.t* inputs
- (2) Still very costly generalisation

Weighted Inputs and Milestones

Compute $n \times m$ OT solution P^* ,

$$R(\mathbf{x}) = n a^{-1} \circ P^* \text{cs}(\mathbf{b}), S(\mathbf{x}) = \mathbf{b}^{-1} \circ (P^*)^T \mathbf{x}$$

All issues do remain however !

- (1) Still not differentiable w.r.t inputs
- (2) Still very costly generalisation

Optimal Transport

generalize both using OT
(overkill!!)

$O((n+m)nm \log(n+m))$

Ranking / Sorting

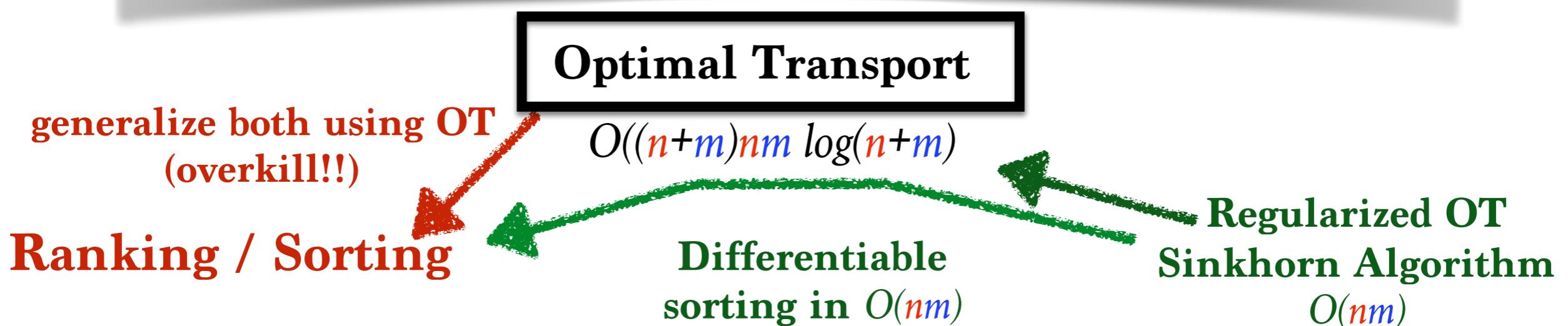
Weighted Inputs and Milestones

Compute $n \times m$ OT solution P^* ,

$$R(\mathbf{x}) = n a^{-1} \circ P^* \text{cs}(\mathbf{b}), S(\mathbf{x}) = \mathbf{b}^{-1} \circ (P^*)^T \mathbf{x}$$

All issues do remain however !

- (1) Still not differentiable w.r.t inputs
- (2) Still very costly generalisation



Sinkhorn Ranks and Sorts

Compute $\textcolor{red}{n} \times \textcolor{blue}{m}$ OT solution $\textcolor{red}{P}^*$,

$$R(\textcolor{red}{x}) = \textcolor{red}{n} \textcolor{red}{a}^{-1} \circ \textcolor{red}{P}^* \text{cs}(\textcolor{blue}{b}), S(\textcolor{red}{x}) = \textcolor{blue}{b}^{-1} \circ (\textcolor{red}{P}^*)^T \textcolor{red}{x}$$

Sinkhorn Ranks and Sorts

Compute $\textcolor{red}{n} \times \textcolor{blue}{m}$ OT solution $\textcolor{red}{P}^*$,

$$R(\mathbf{x}) = \textcolor{red}{n} \mathbf{a}^{-1} \circ \textcolor{red}{P}^* \text{cs}(\mathbf{b}), S(\mathbf{x}) = \mathbf{b}^{-1} \circ (\textcolor{red}{P}^*)^T \mathbf{x}$$

Set ε , compute $\mathbf{u}_L, \mathbf{v}_L$ in $\textcolor{red}{n} \textcolor{blue}{m} L$ ops

$$R(\mathbf{x}) = \textcolor{red}{n} \mathbf{a}^{-1} \circ \mathbf{u}_L \circ \textcolor{red}{K} \mathbf{v}_L \circ \text{cs}(\mathbf{b})$$

$$S(\mathbf{x}) = \mathbf{b}^{-1} \circ \mathbf{v}_L \circ \textcolor{blue}{K}^T \mathbf{u}_L \circ \mathbf{x}$$

Sinkhorn Ranks and Sorts

Compute $n \times m$ OT solution P^* ,

$$R(\mathbf{x}) = n a^{-1} \circ P^* \text{cs}(\mathbf{b}), S(\mathbf{x}) = \mathbf{b}^{-1} \circ (P^*)^T \mathbf{x}$$

Set ε , compute u_L, v_L in nmL ops

$$R(\mathbf{x}) = n a^{-1} \circ u_L \circ K v_L \circ \text{cs}(\mathbf{b})$$

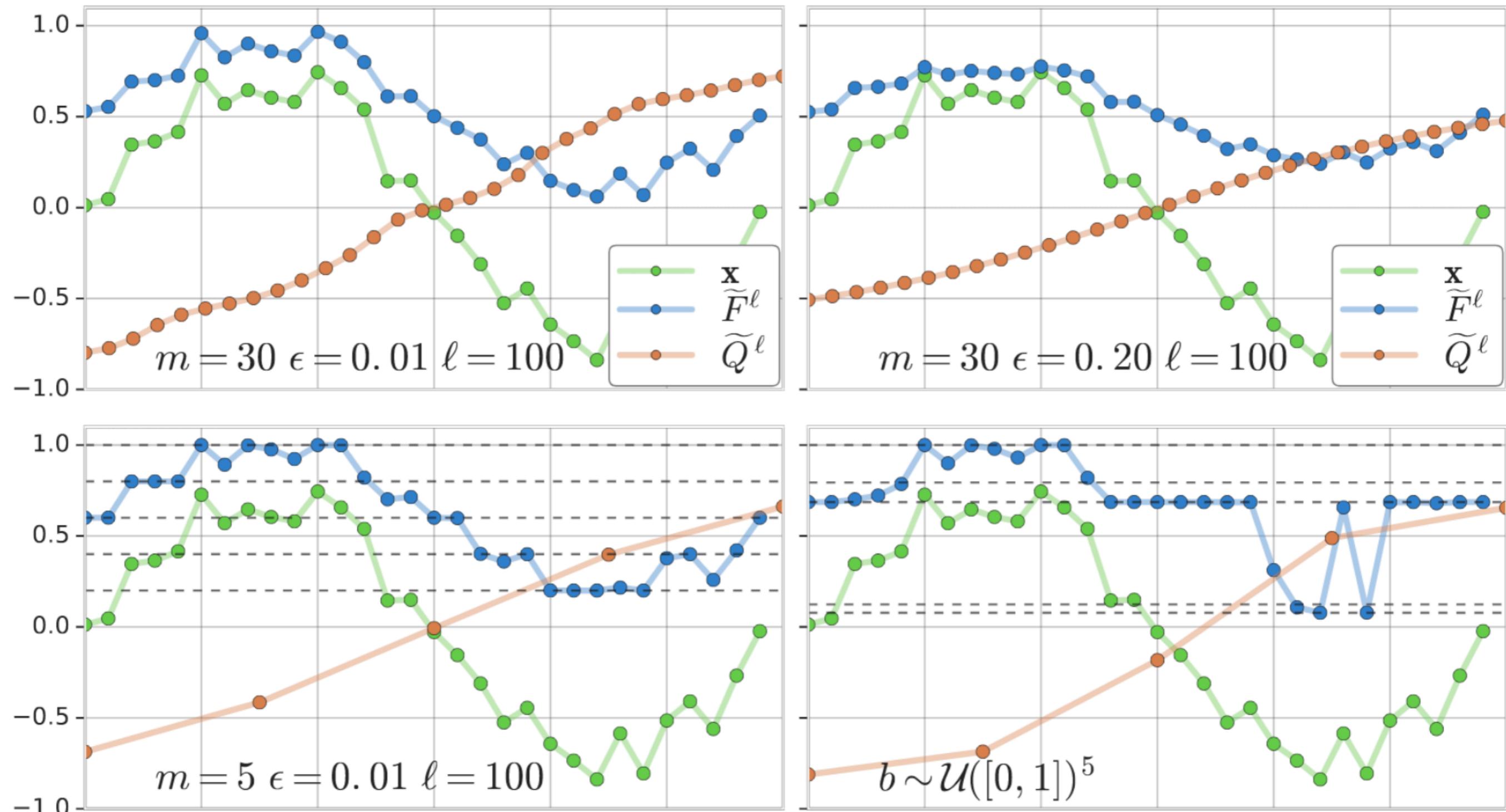
$$S(\mathbf{x}) = \mathbf{b}^{-1} \circ v_L \circ K^T u_L \circ \mathbf{x}$$

Ranking / Sorting

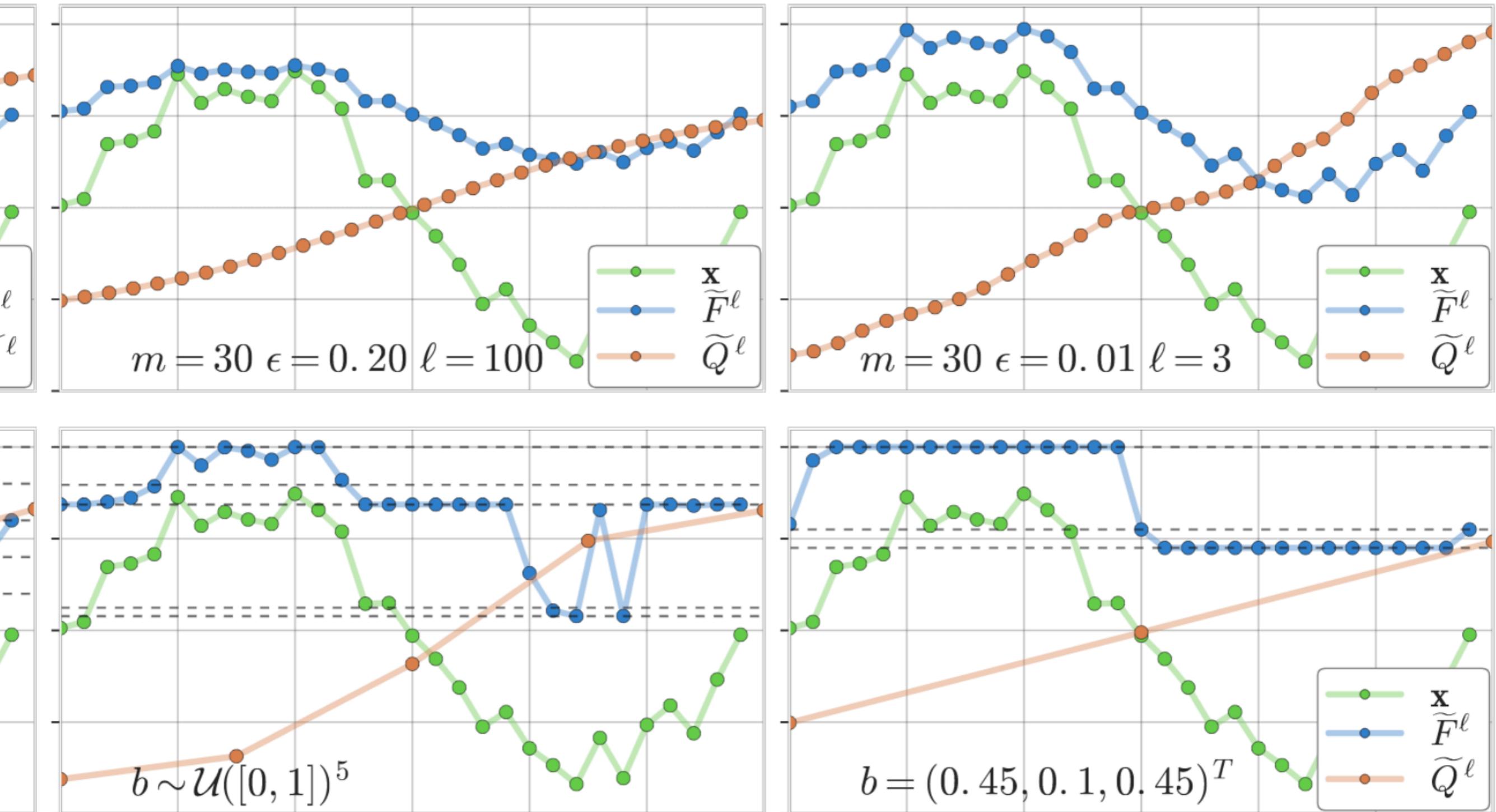
Differentiable
sorting in $O(nm)$

Regularized OT
Sinkhorn Algorithm

Sinkhorn Sort, Ranks, Quantiles



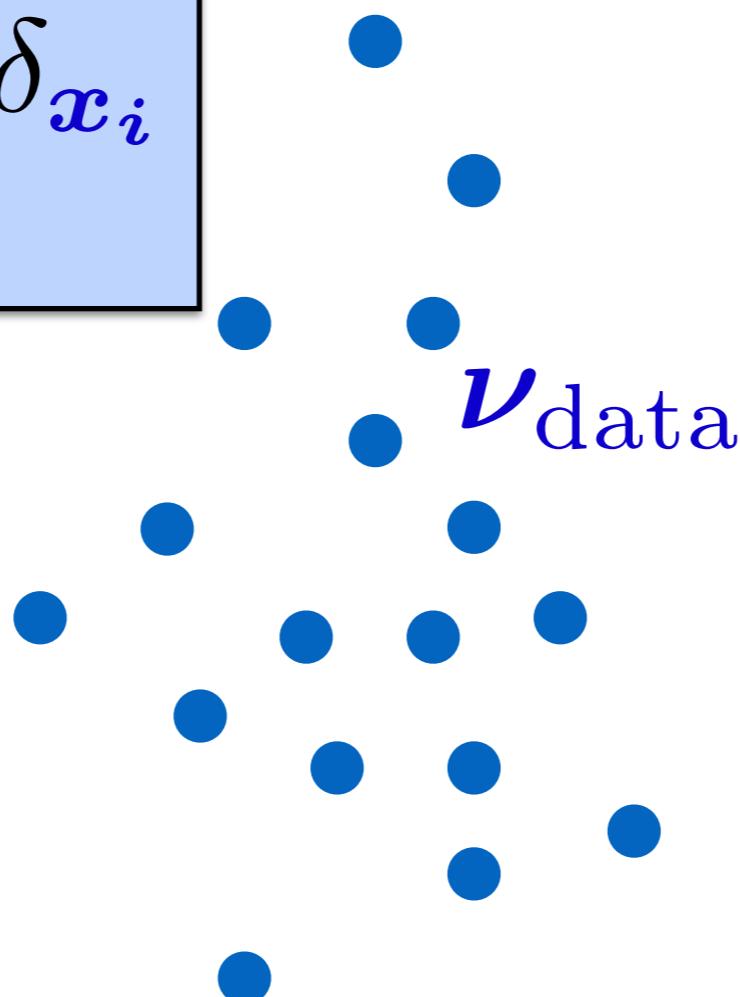
Sinkhorn Sort, Ranks, Quantiles



Generative Models

We collect data

$$\nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_i}$$



Generative Models

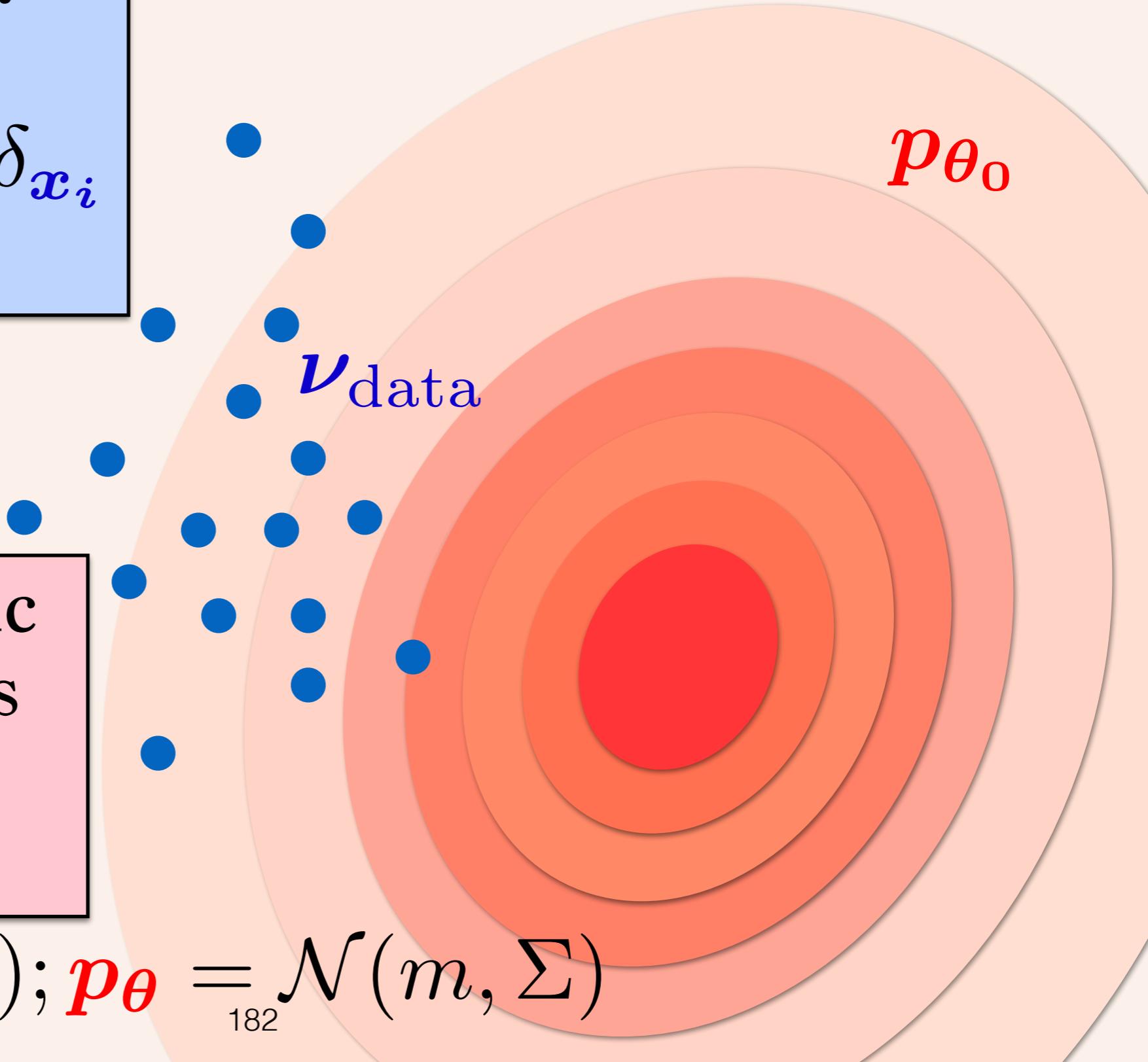
We collect data

$$\nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_i}$$

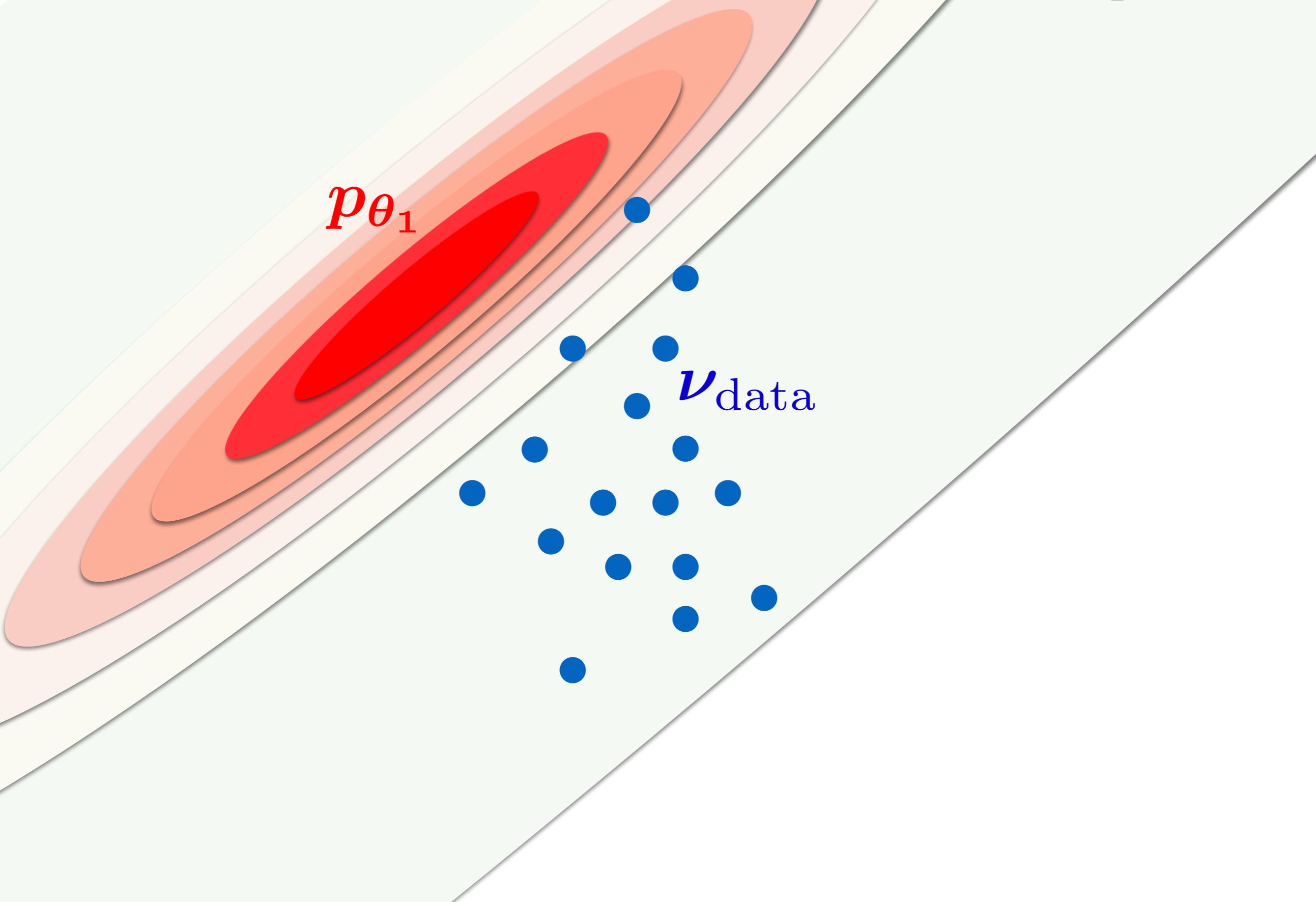
We fit a parametric family of densities

$$\{p_\theta, \theta \in \Theta\}$$

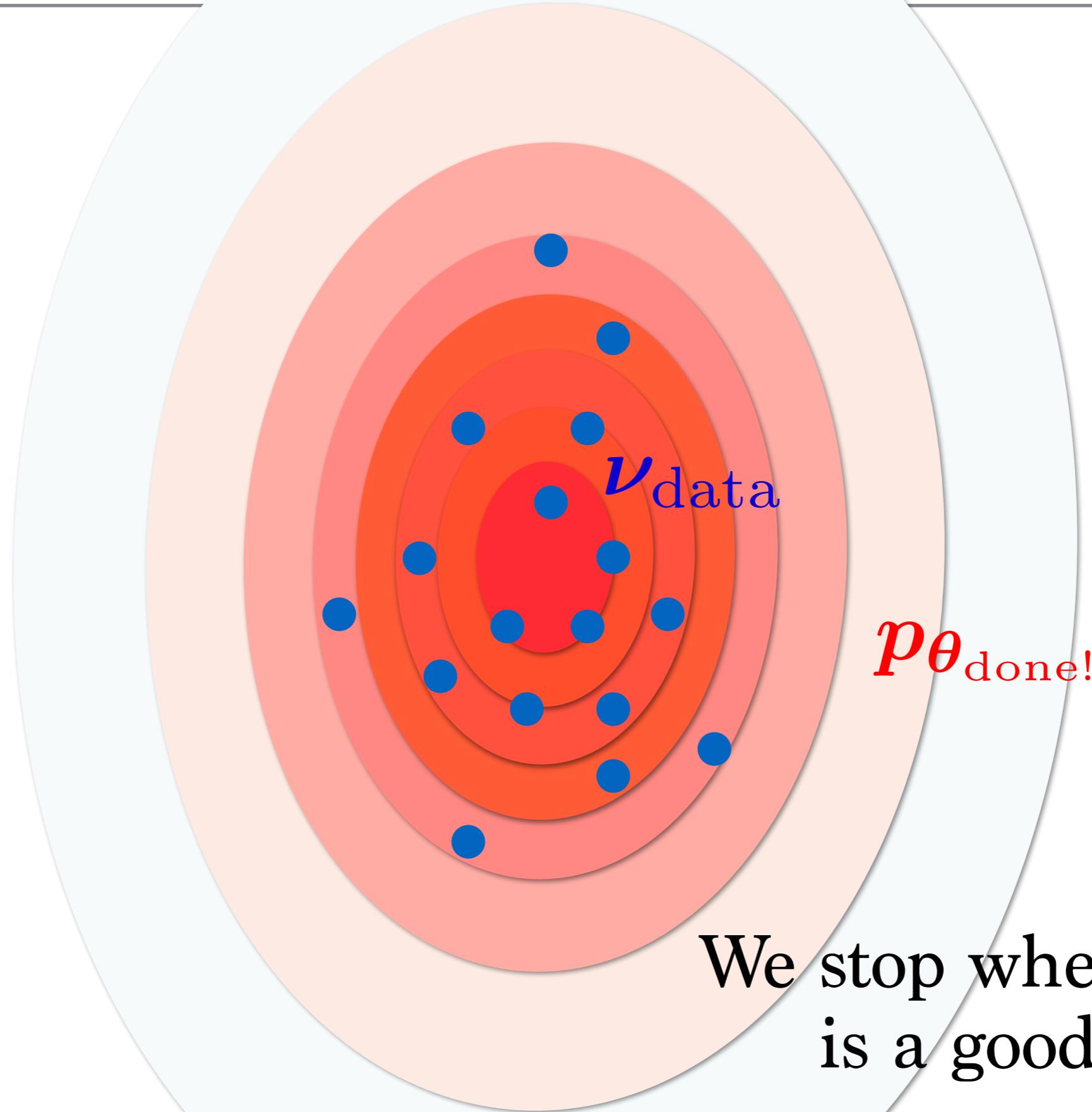
e.g. $\theta = (m, \Sigma)$; $p_\theta = \mathcal{N}(m, \Sigma)$



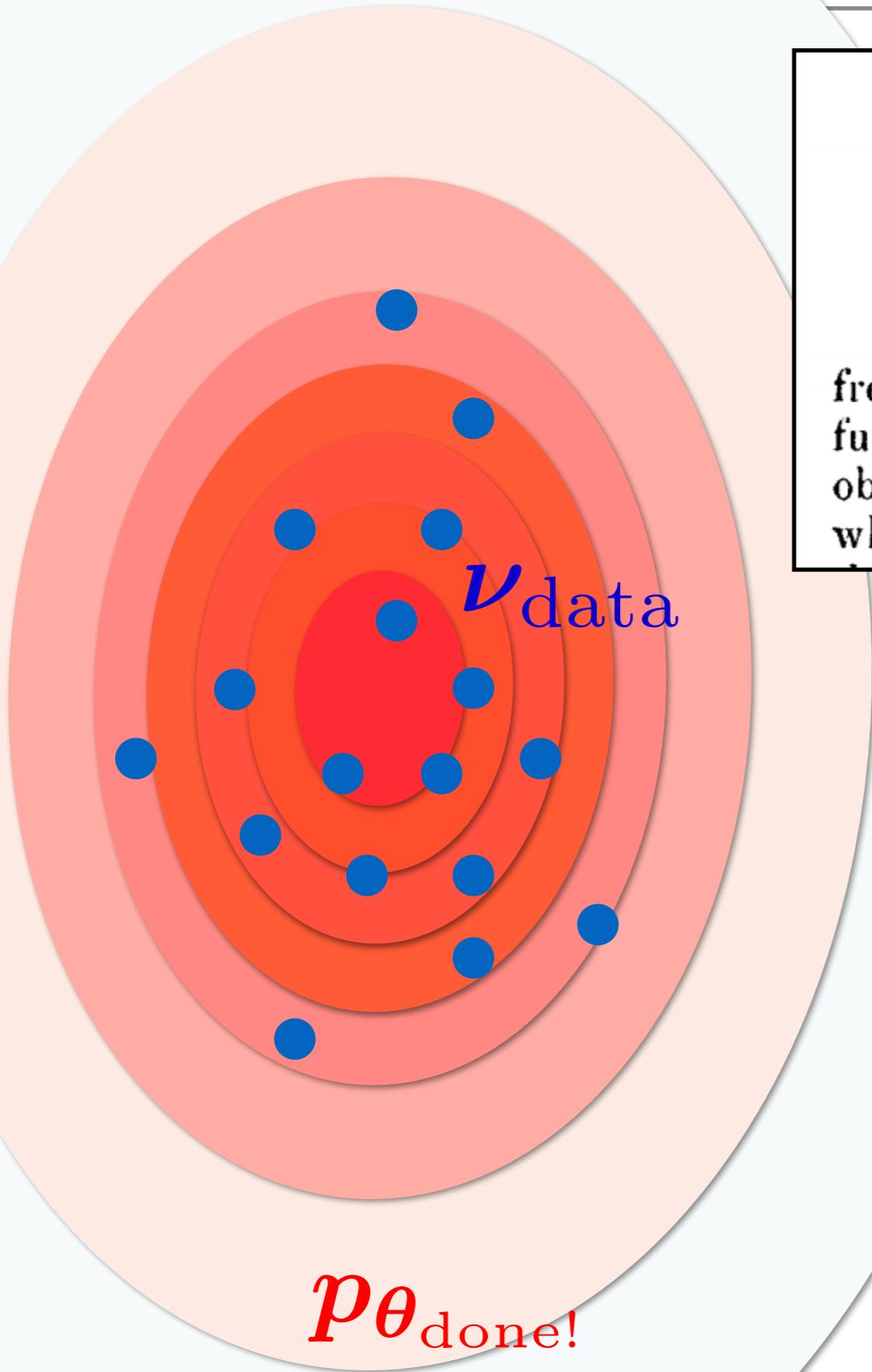
Statistics 0.1: Density Fitting



Statistics 0.1: Density Fitting



Maximum Likelihood Estimation



ON AN ABSOLUTE CRITERION
FOR FITTING FREQUENCY CURVES.

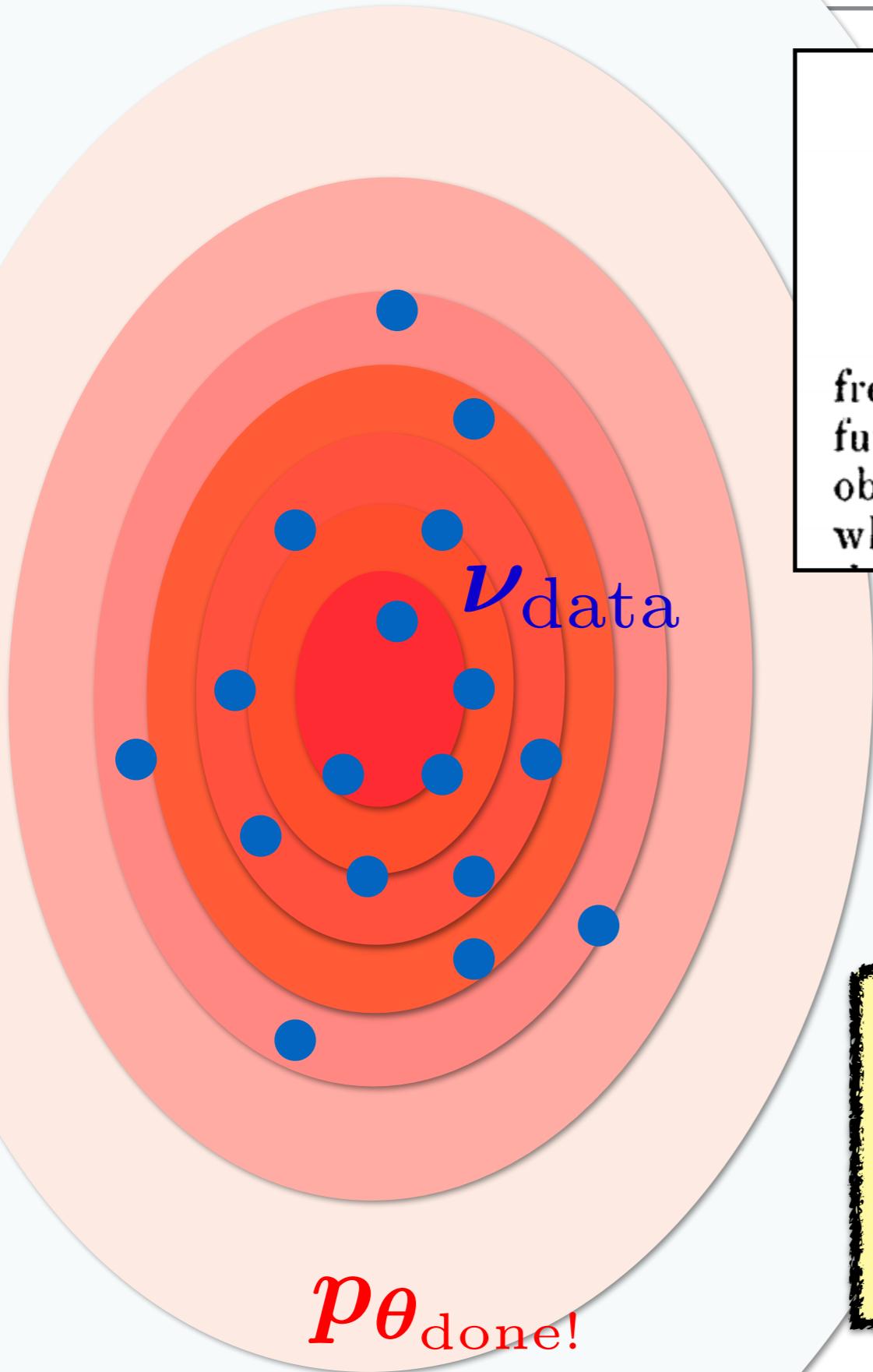
By *R. A. Fisher*, Gonville and Caius College, Cambridge.

1. IF we set ourselves the problem, in its frequent occurrence, of finding the arbitrary function of known form, which best suit a observations, we are met at the outset by an which appears to invalidate any results we ma



$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i)$$

Maximum Likelihood Estimation



ON AN ABSOLUTE CRITERION
FOR FITTING FREQUENCY CURVES.

By R. A. Fisher, Gonville and Caius College, Cambridge.

1. IF we set ourselves the problem, in its frequent occurrence, of finding the arbitrary function of known form, which best suit a observations, we are met at the outset by an which appears to invalidate any results we ma



$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i)$$

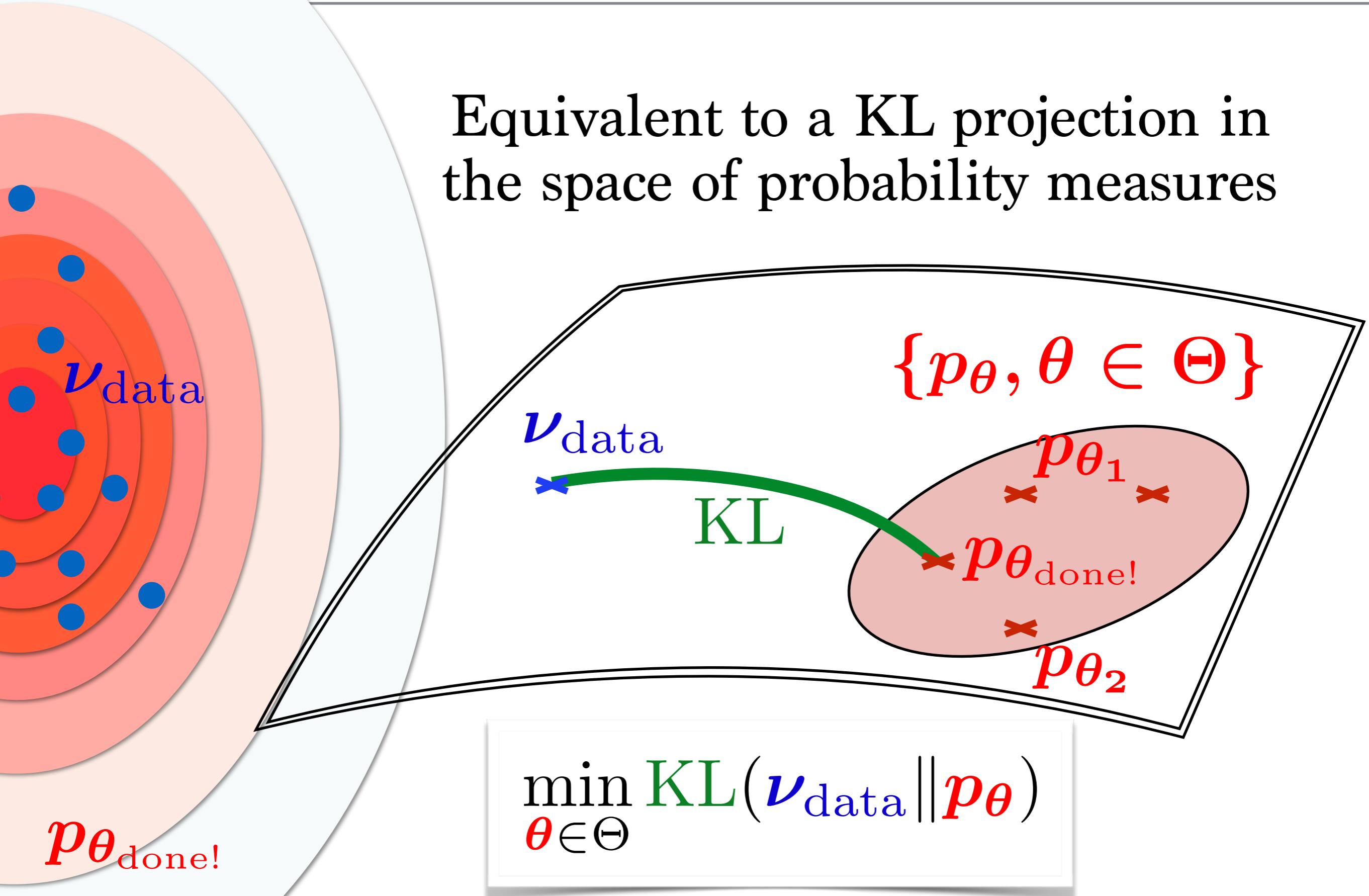


$$\log 0 = -\infty$$

$p_{\theta}(x_i)$ must be > 0

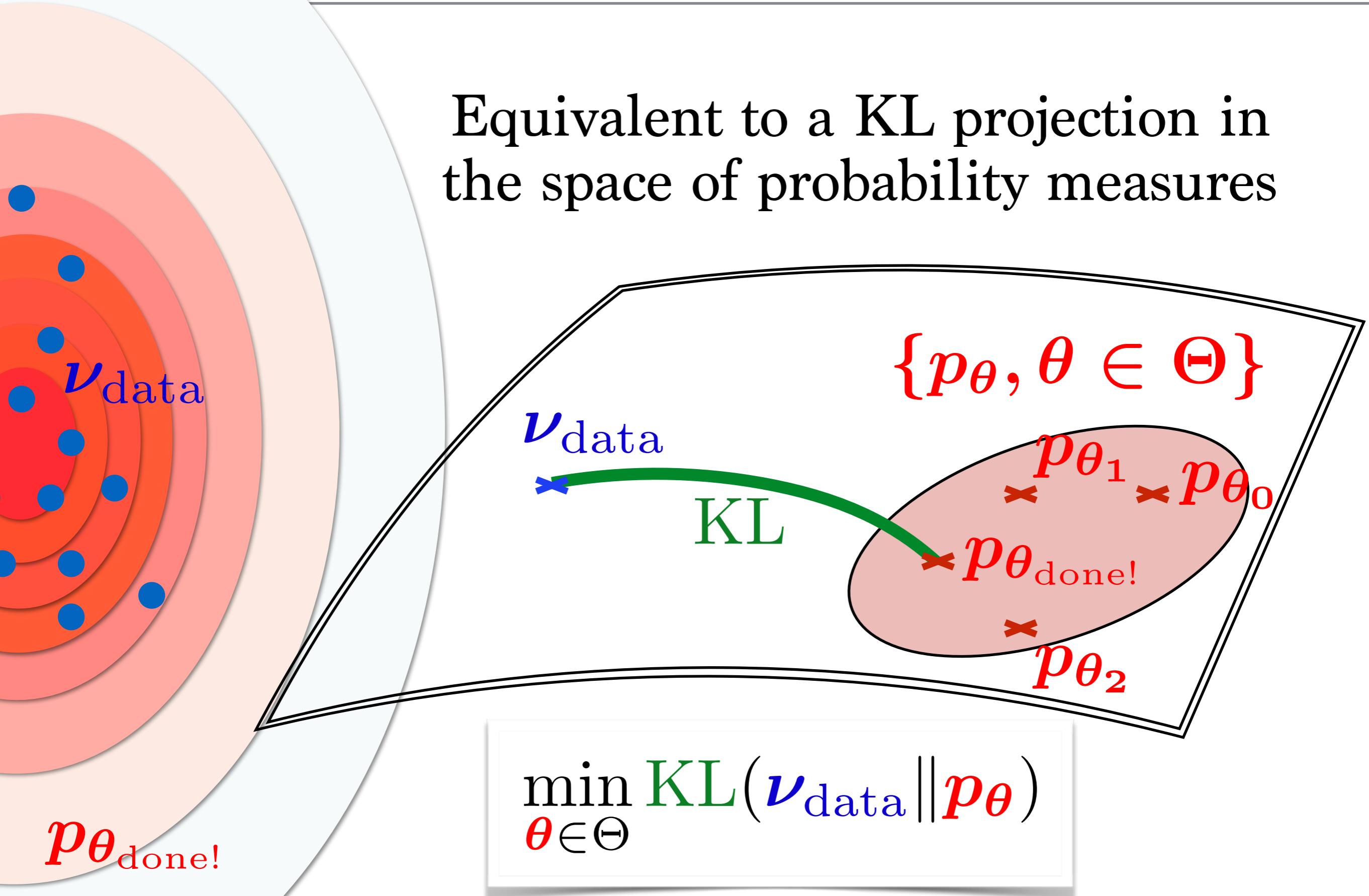
Maximum Likelihood Estimation

Equivalent to a KL projection in the space of probability measures

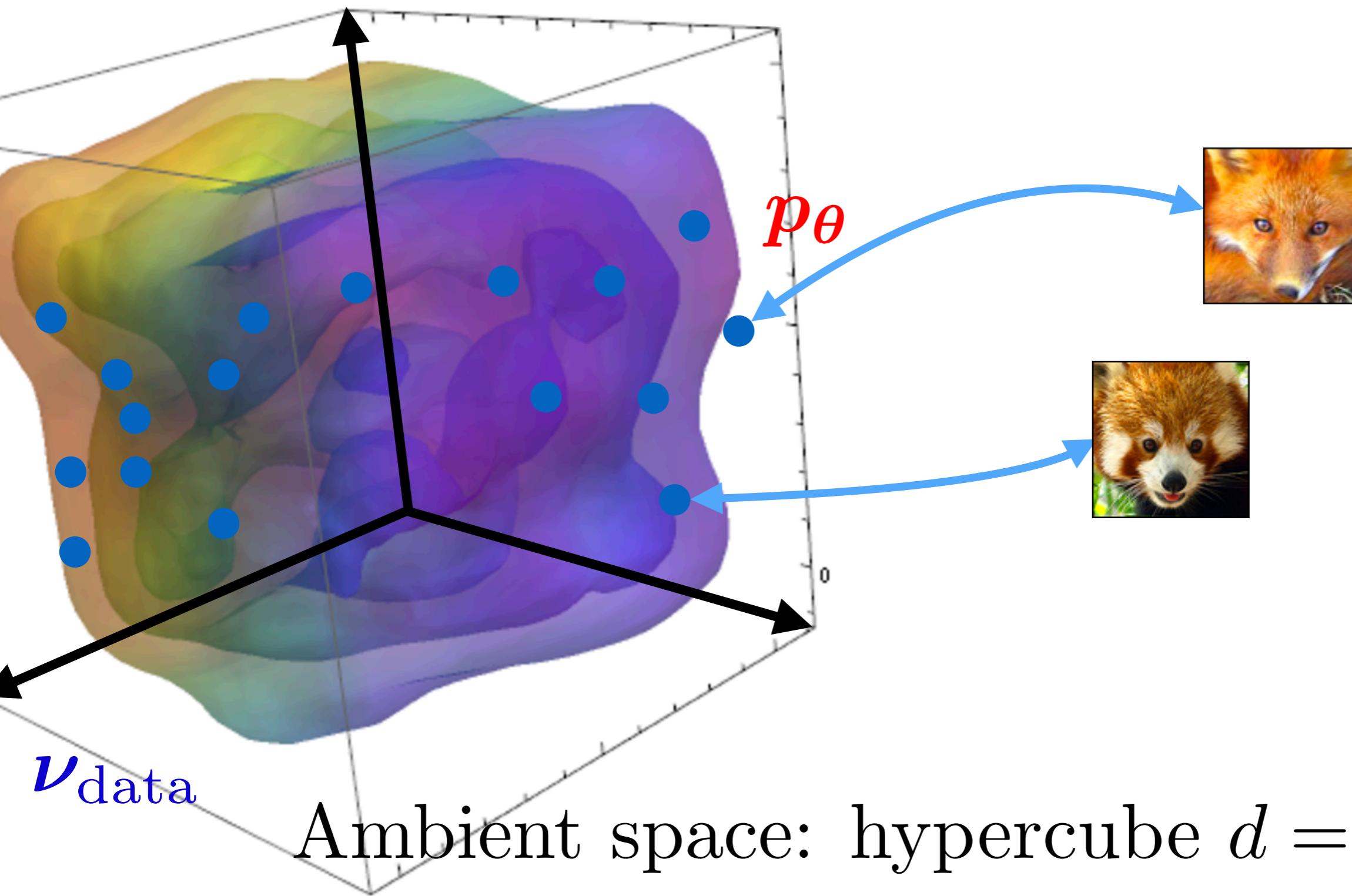


Maximum Likelihood Estimation

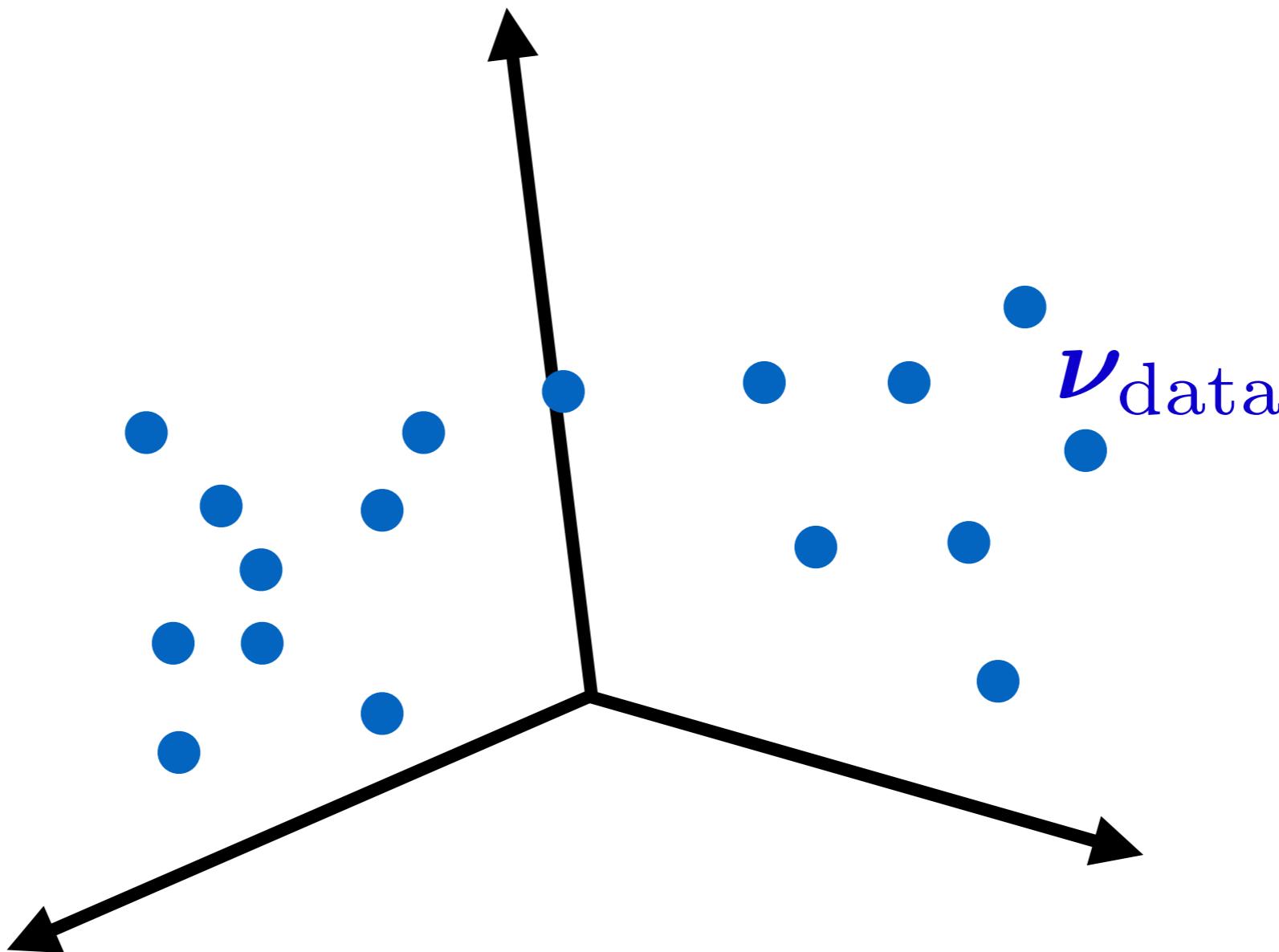
Equivalent to a KL projection in the space of probability measures



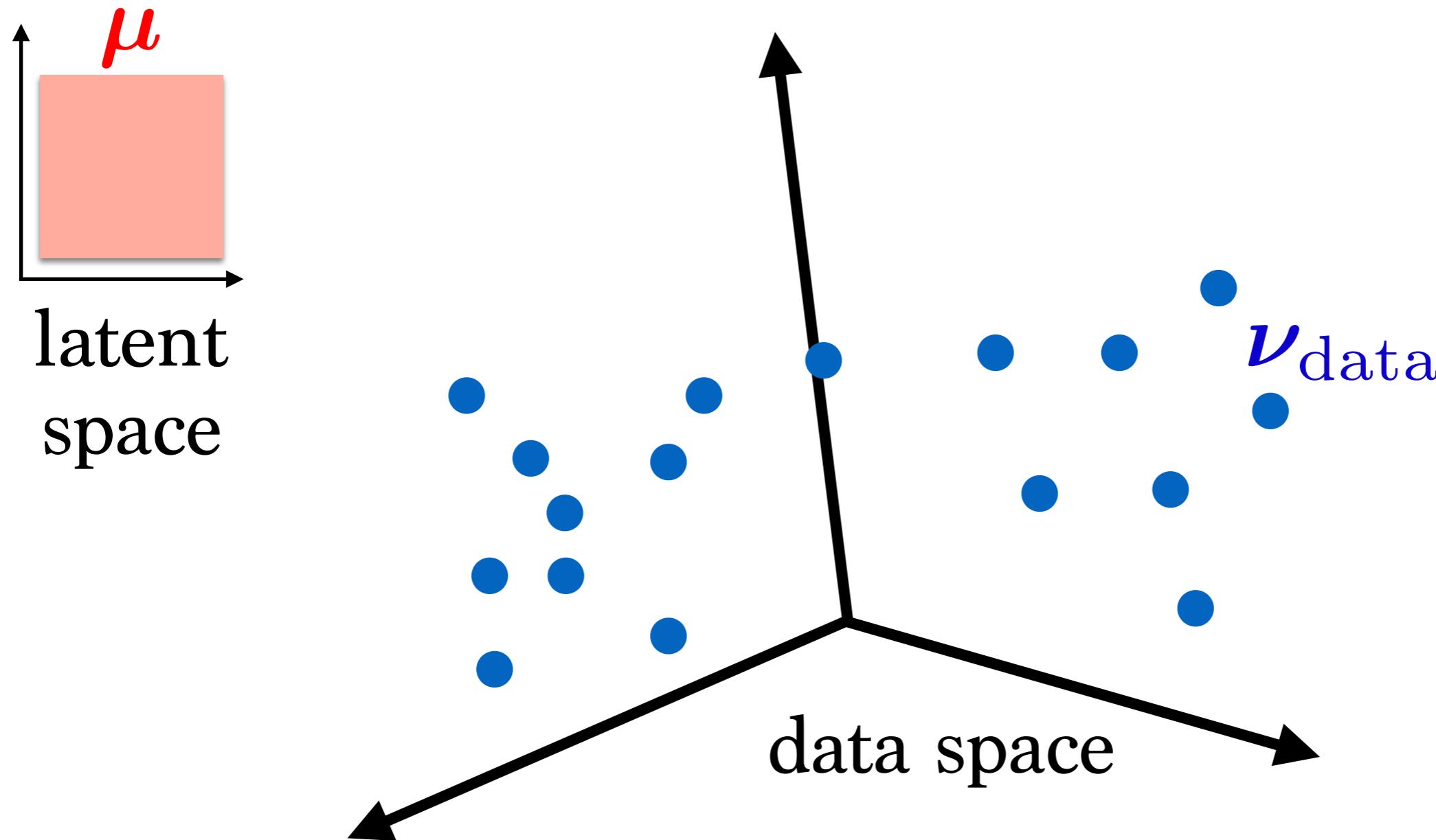
In higher dimensional spaces...



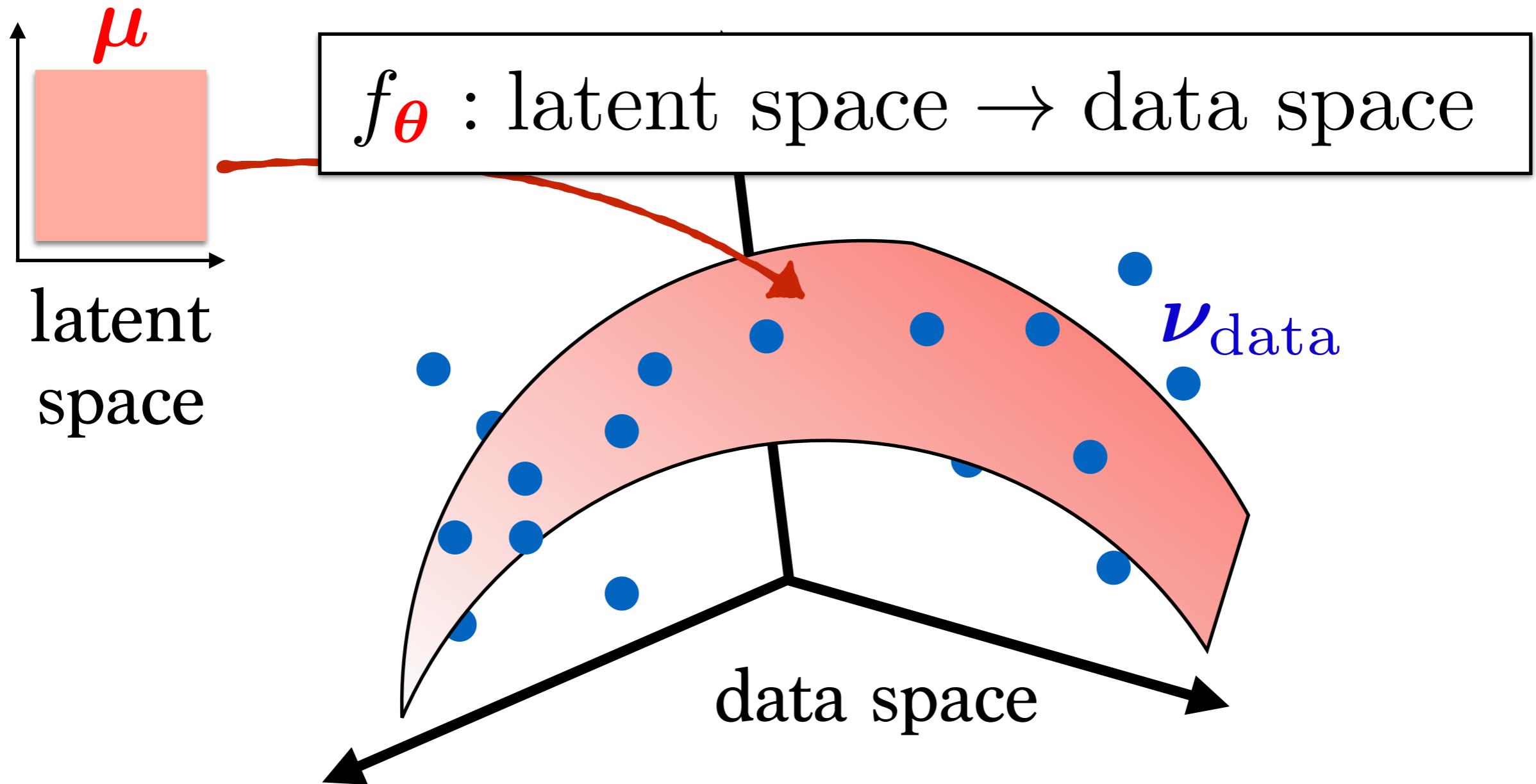
Generative Models



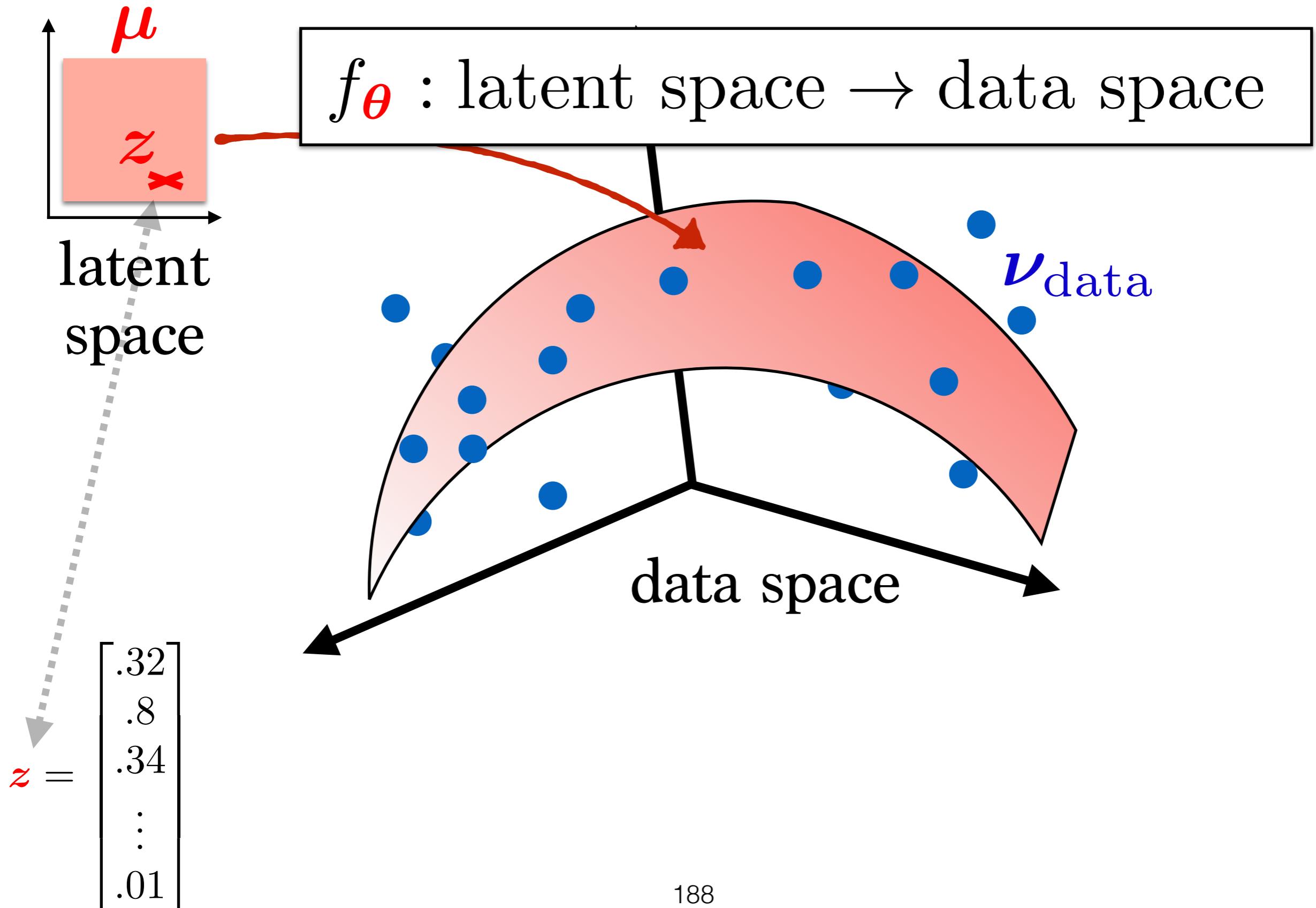
Generative Models



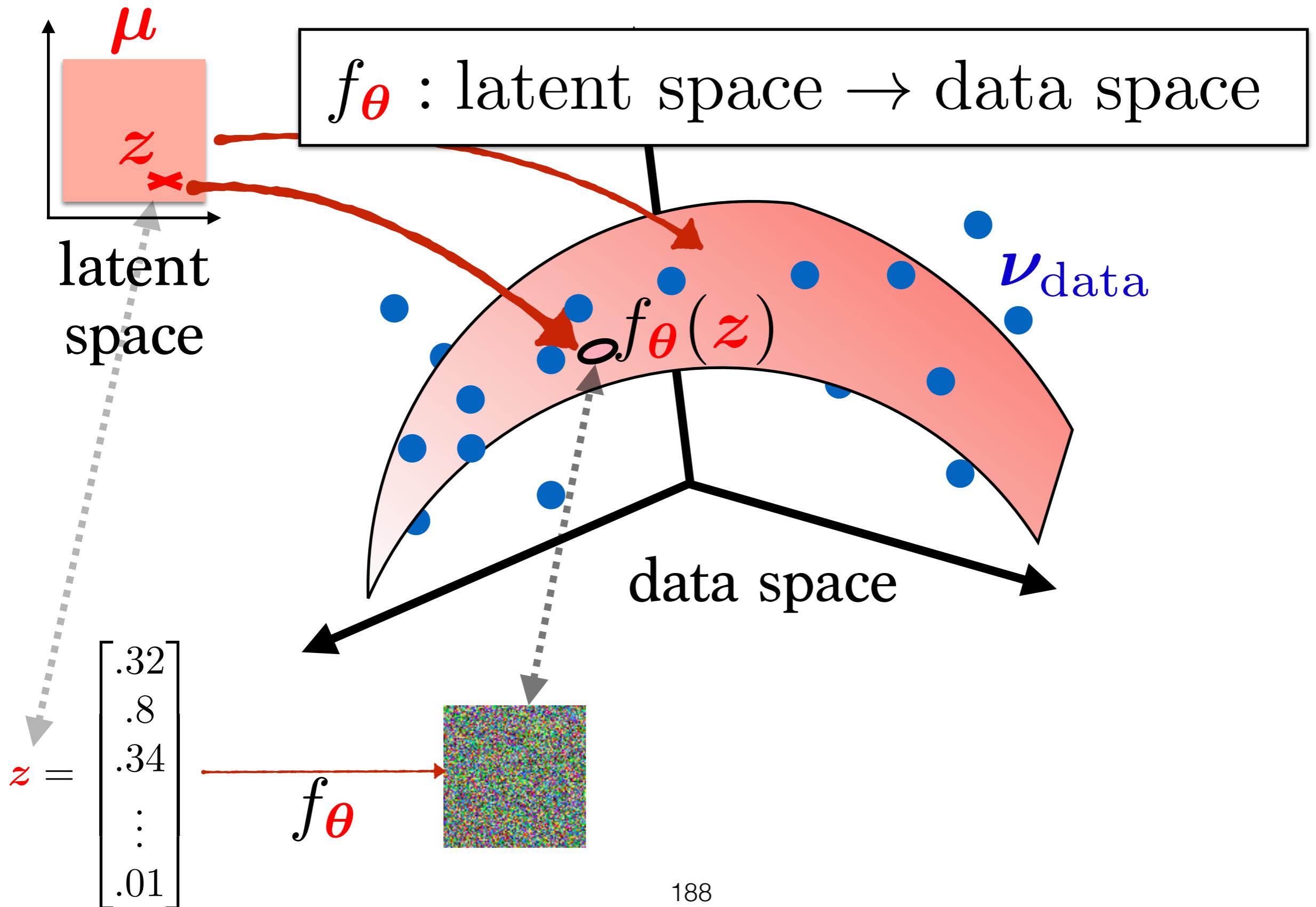
Generative Models



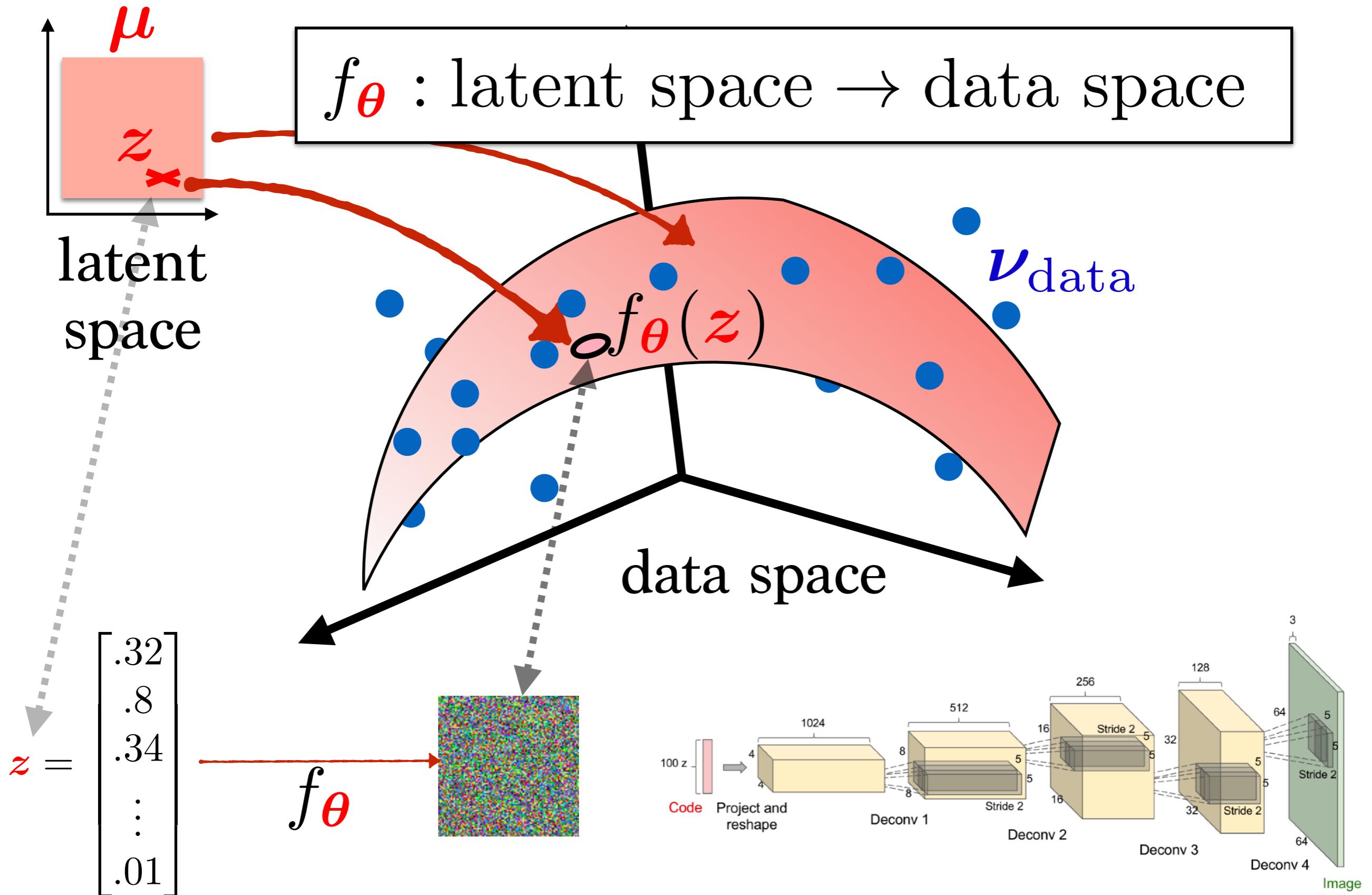
Generative Models



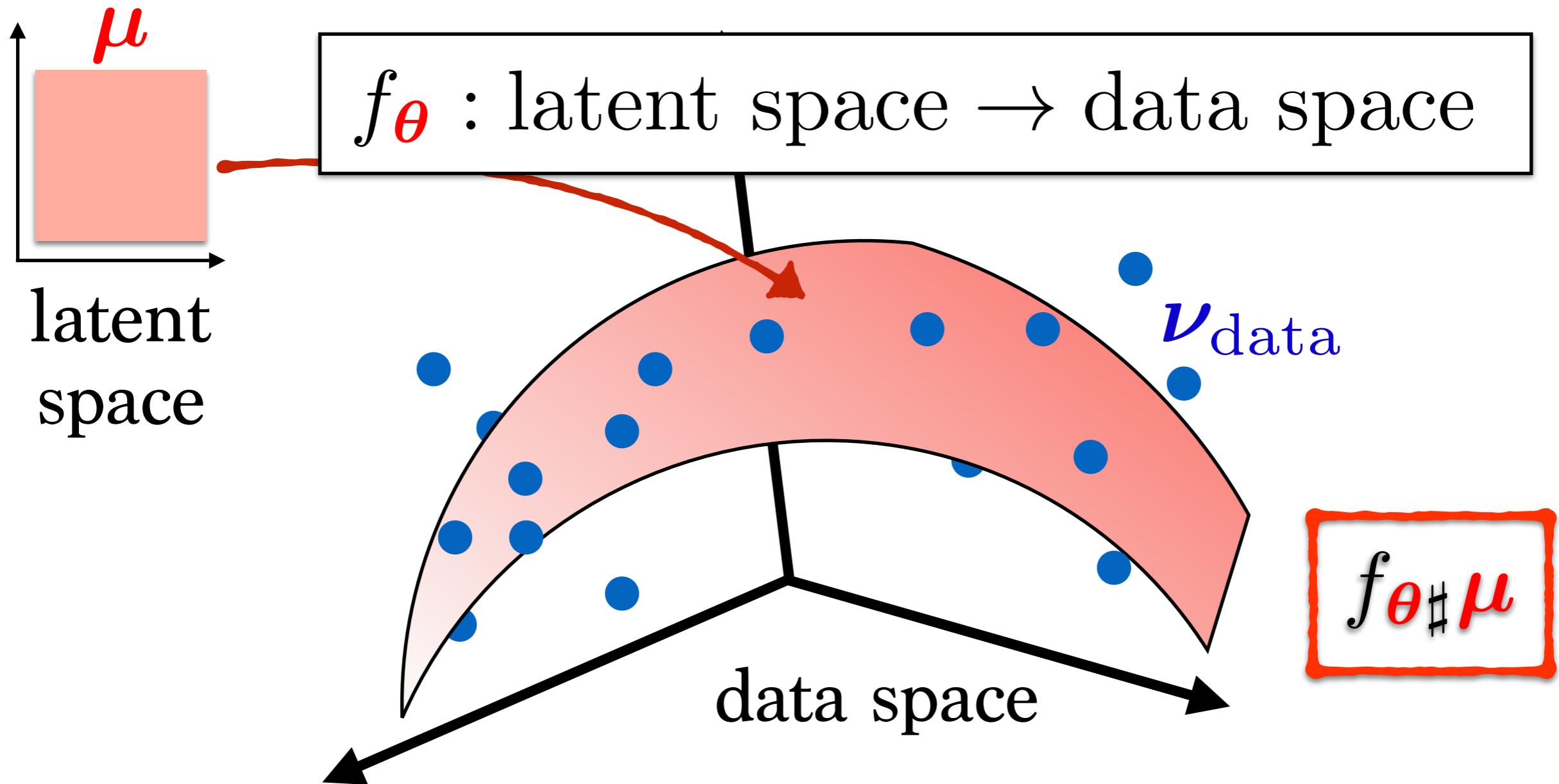
Generative Models



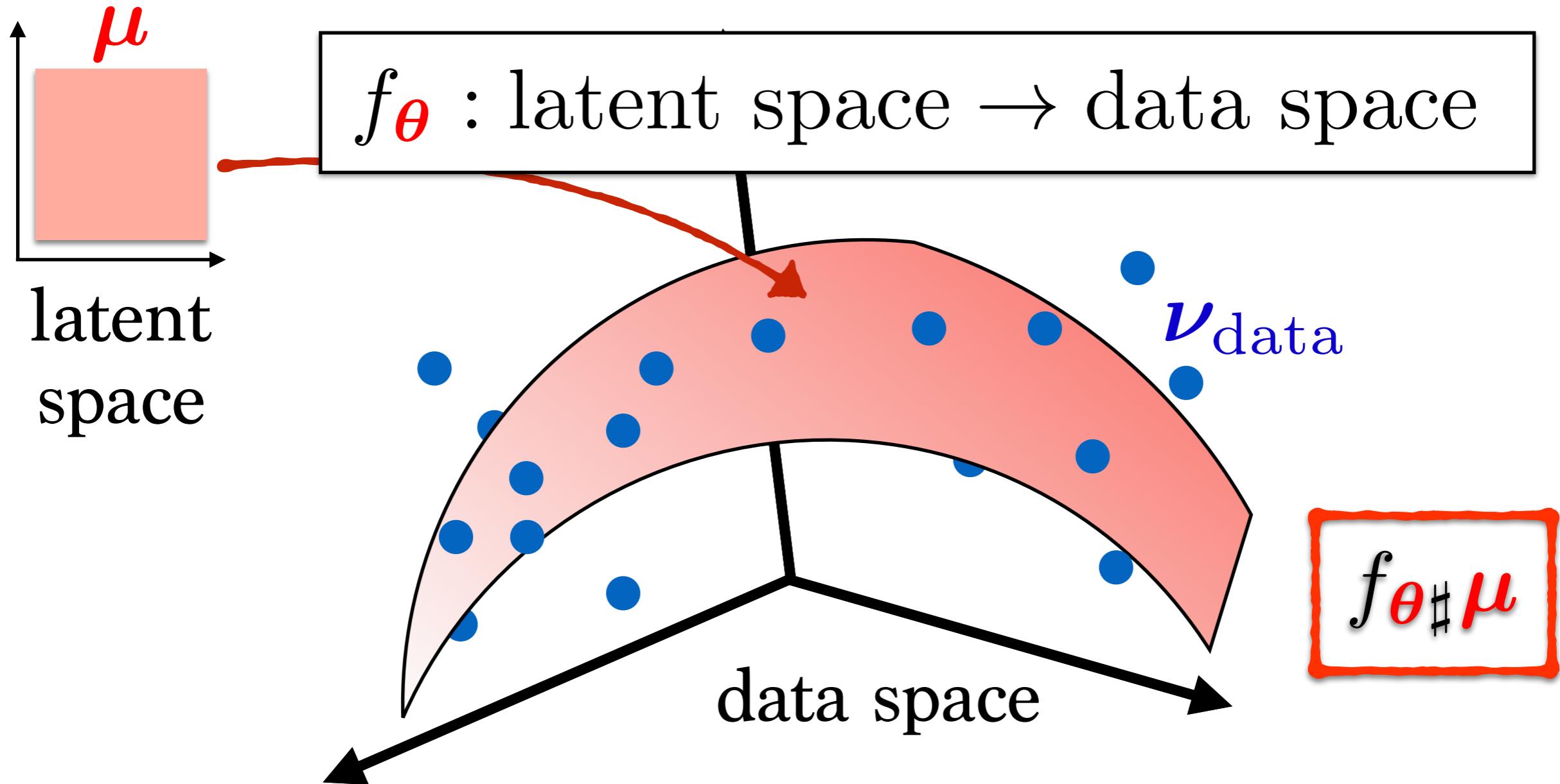
Generative Models



Generative Models

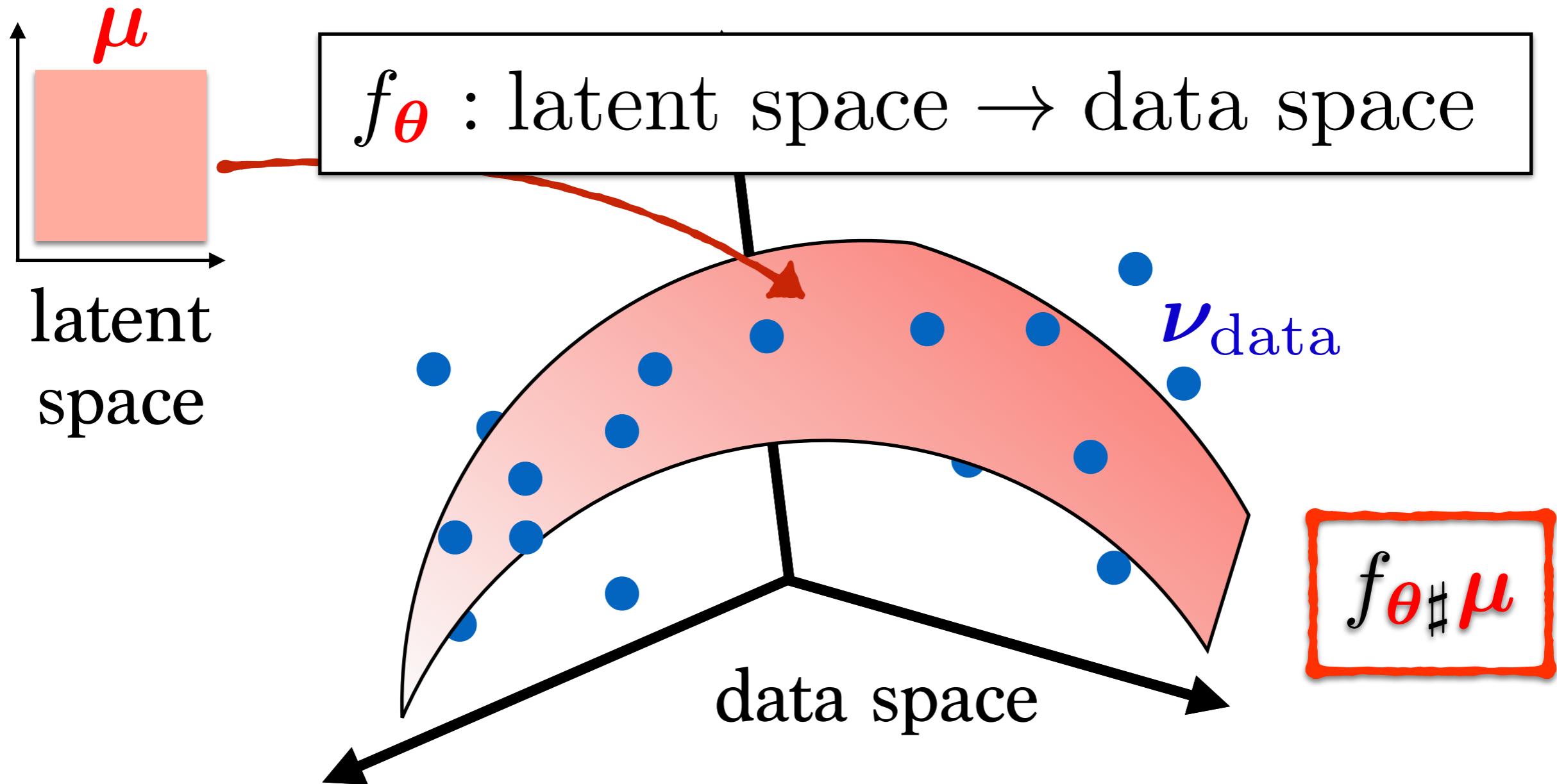


Generative Models



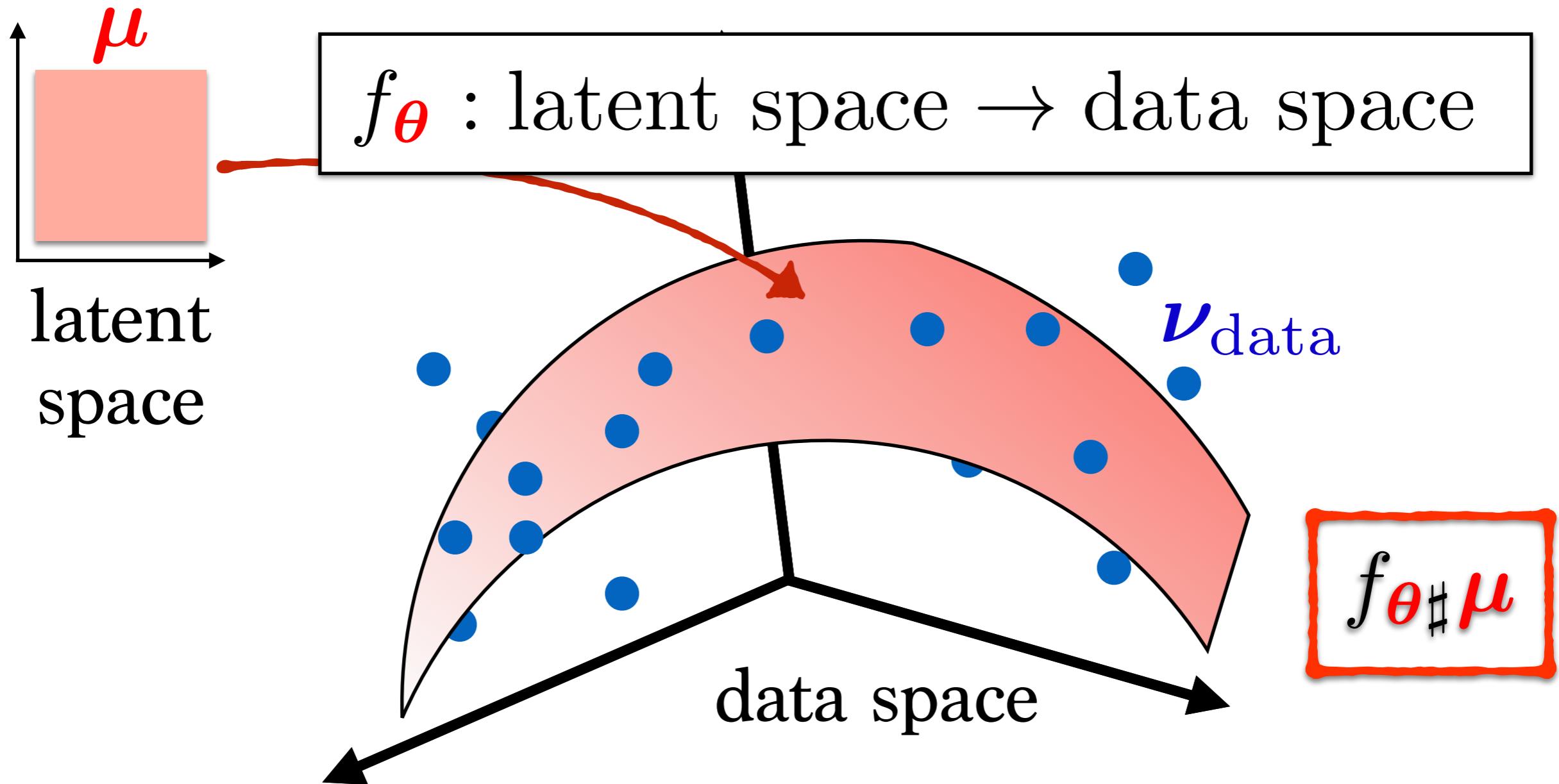
Goal: find θ such that $f_{\theta \sharp} \mu$ fits ν_{data}

Generative Models



Goal: find θ such that $f_{\theta \sharp} \mu$ fits ν_{data}

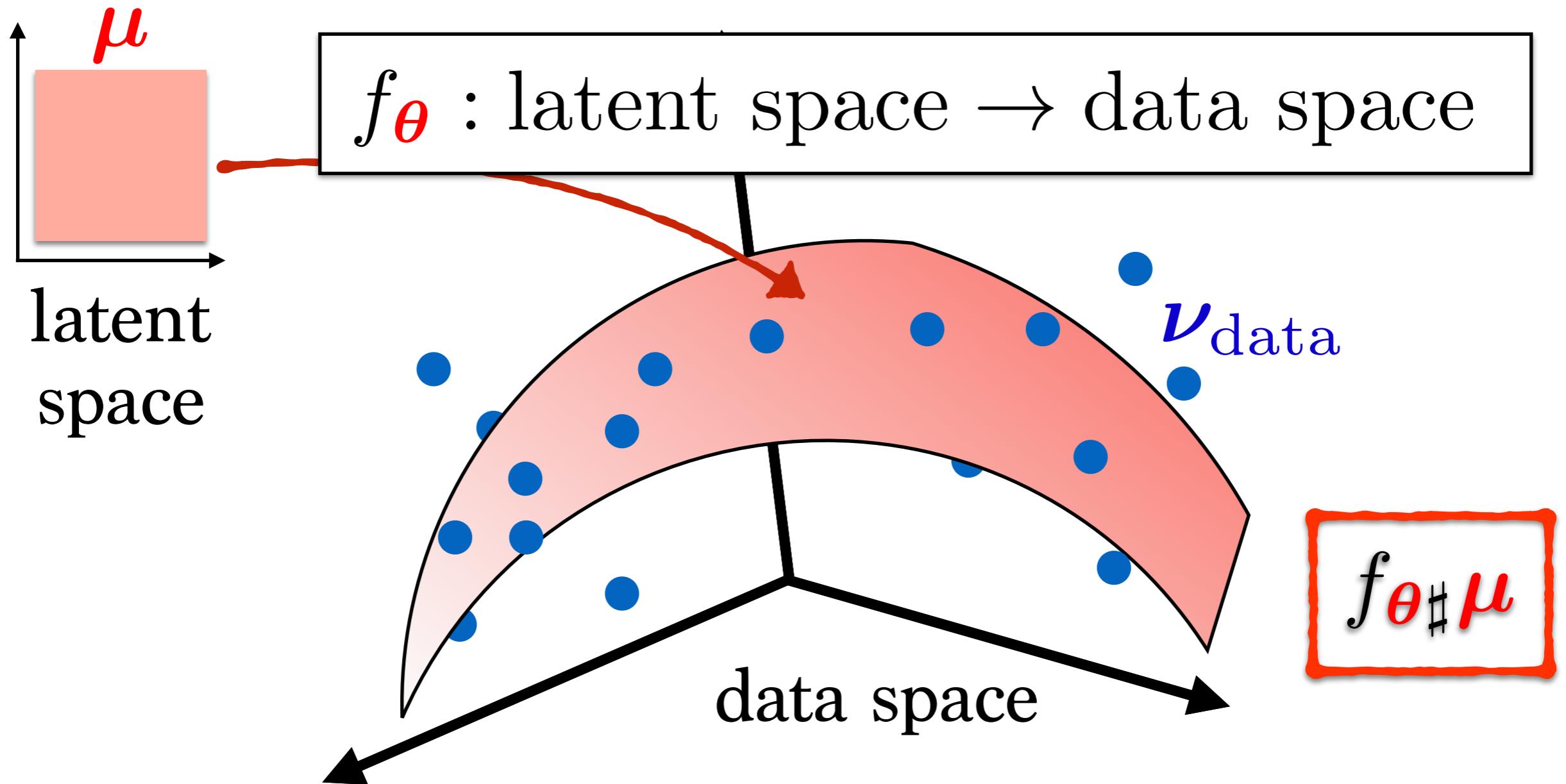
Generative Models



MLE

$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_\theta(\mathbf{x}_i) = \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \| p_\theta)$$

Generative Models



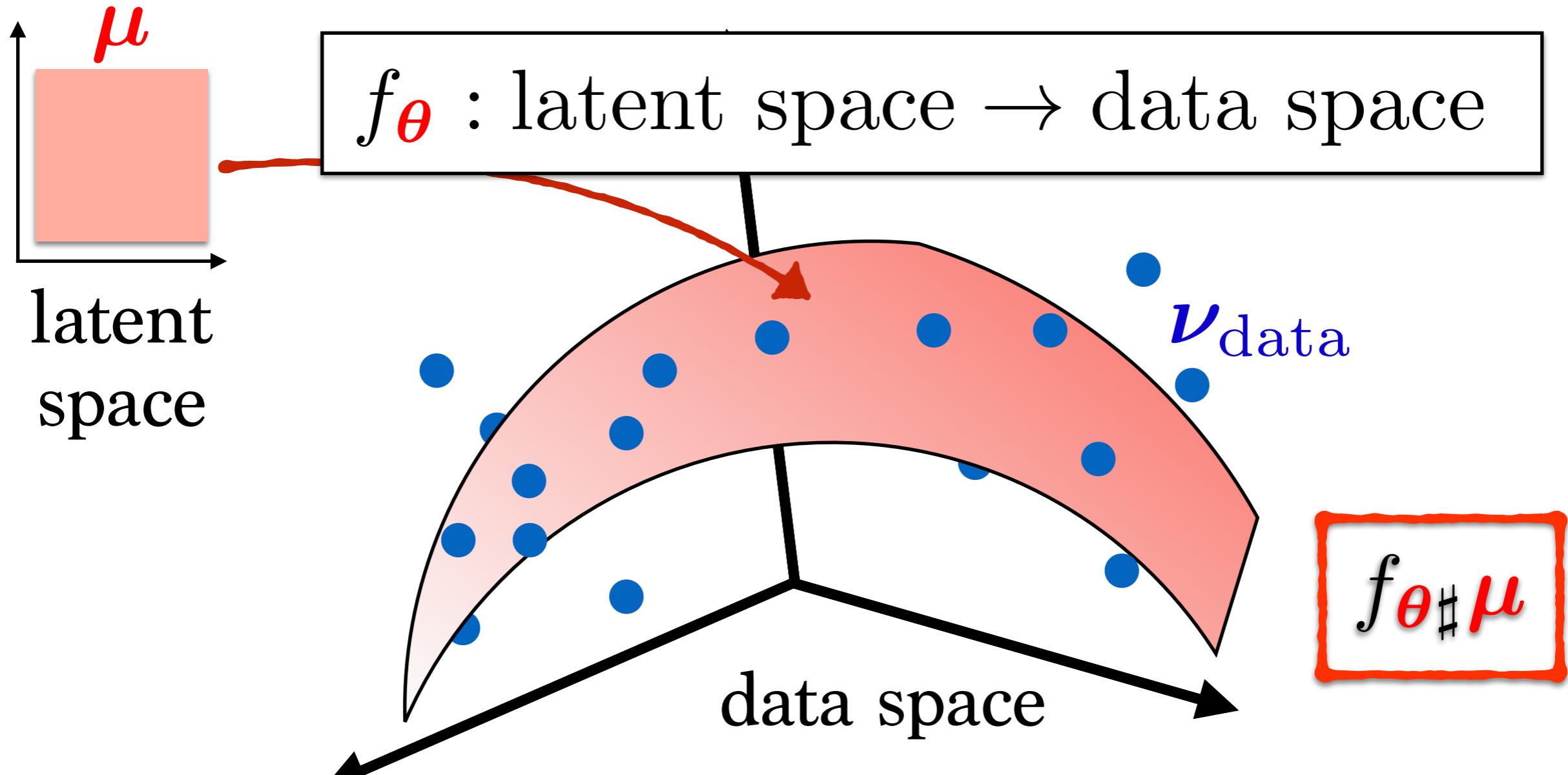
MLE

$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log \underline{f_{\theta \sharp} \mu}(x_i)$$

$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel \underline{f_{\theta \sharp} \mu})$$



Generative Models

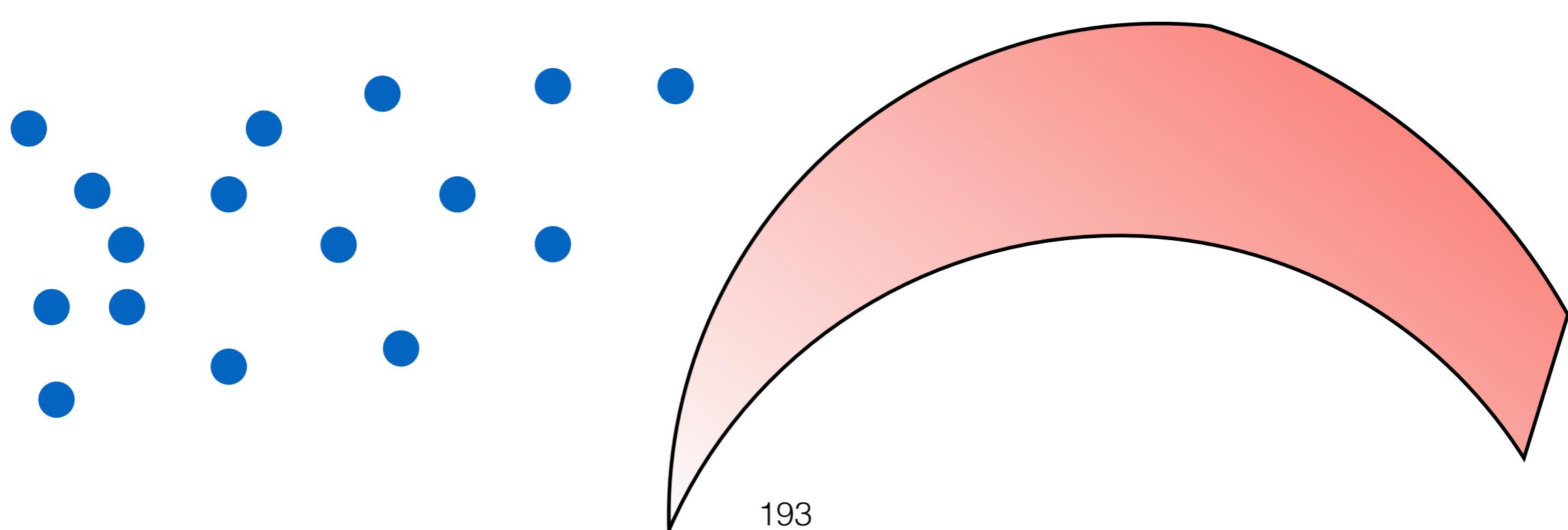


Need a more flexible discrepancy function to compare ν_{data} and $f_{\theta\sharp}\mu$

Workarounds?

- Formulation as adversarial problem [GPM...'14]

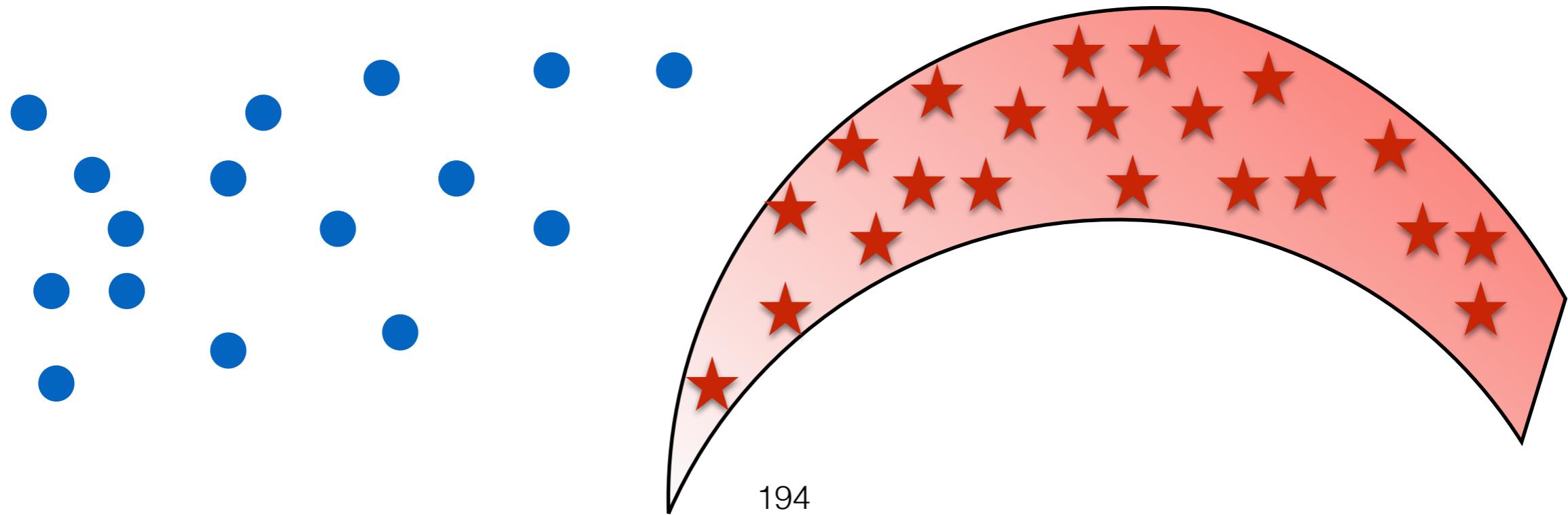
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as adversarial problem [GPM...'14]

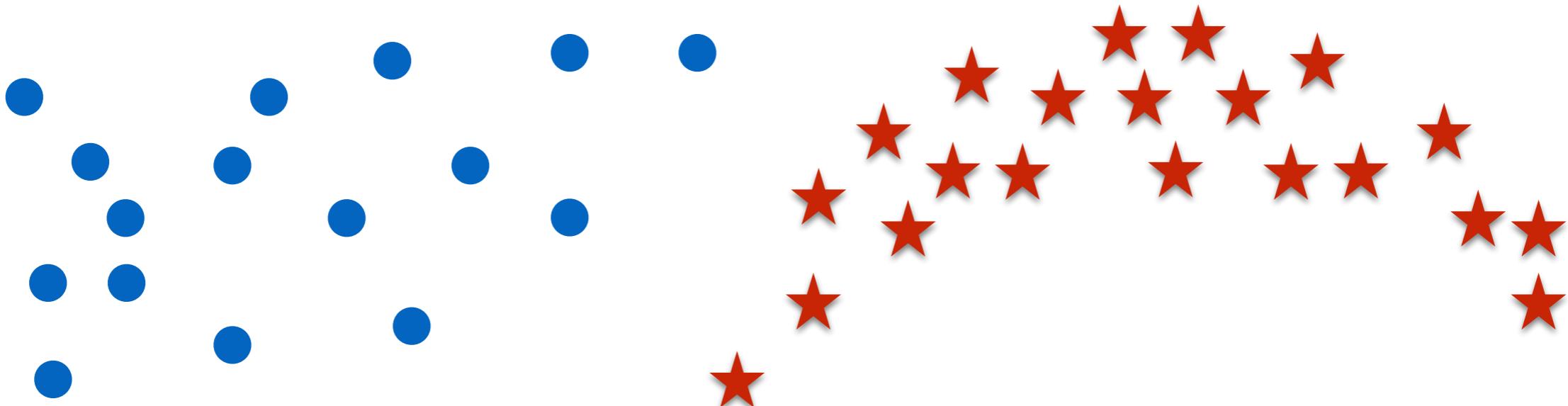
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as adversarial problem [GPM...'14]

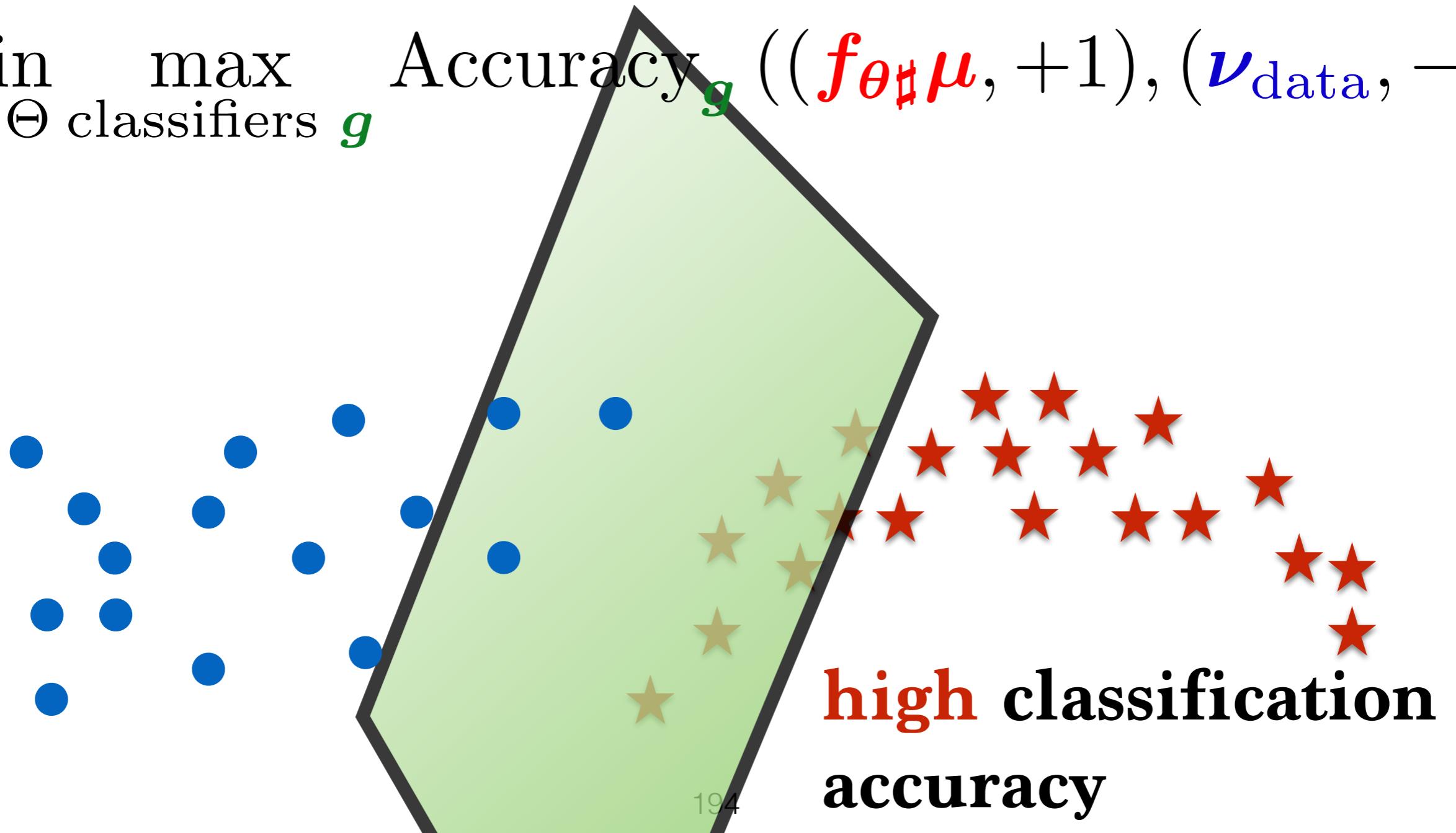
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as adversarial problem [GPM...'14]

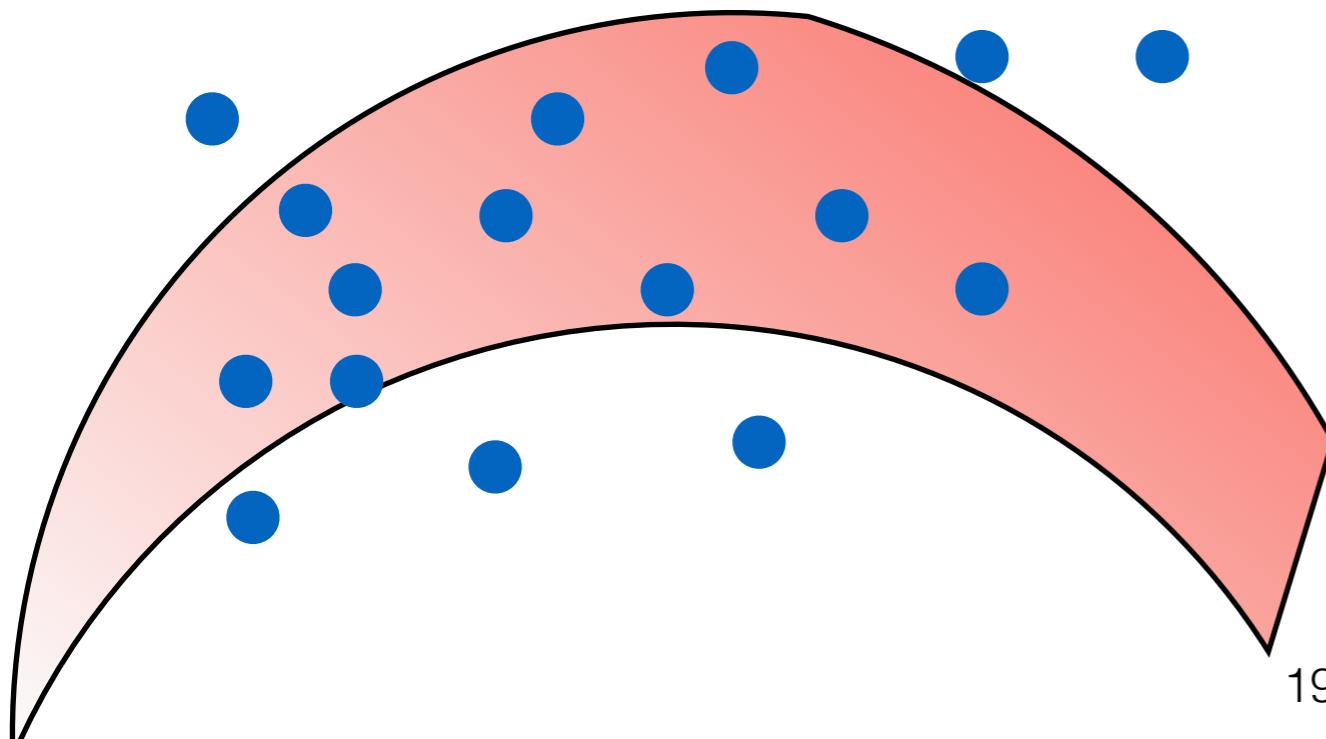
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as adversarial problem [GPM...'14]

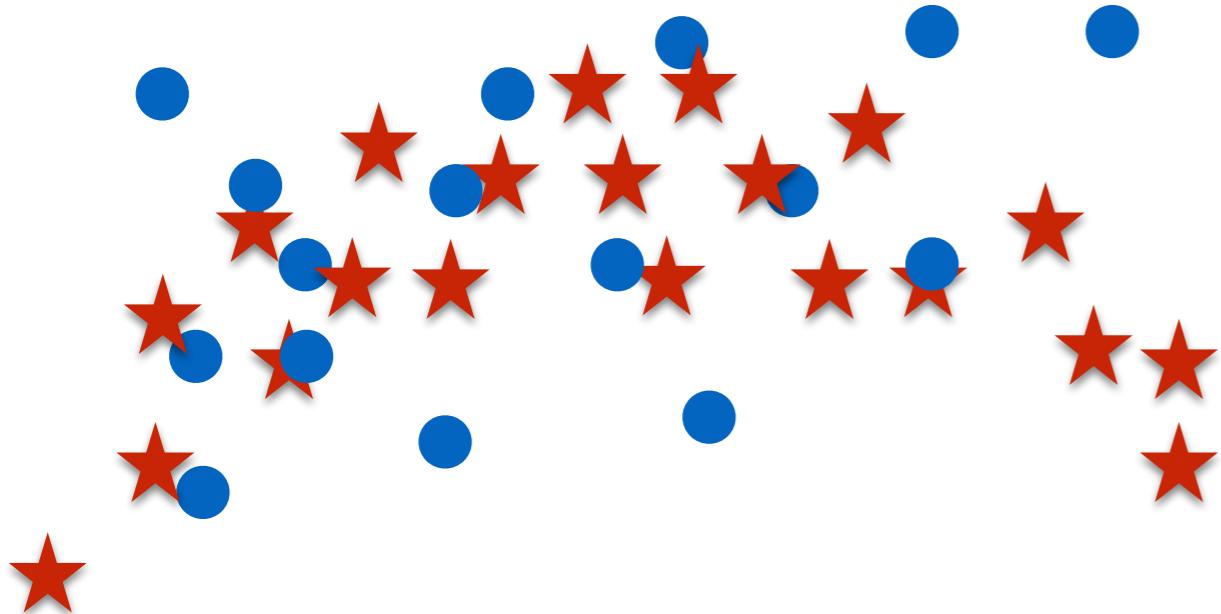
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as adversarial problem [GPM...'14]

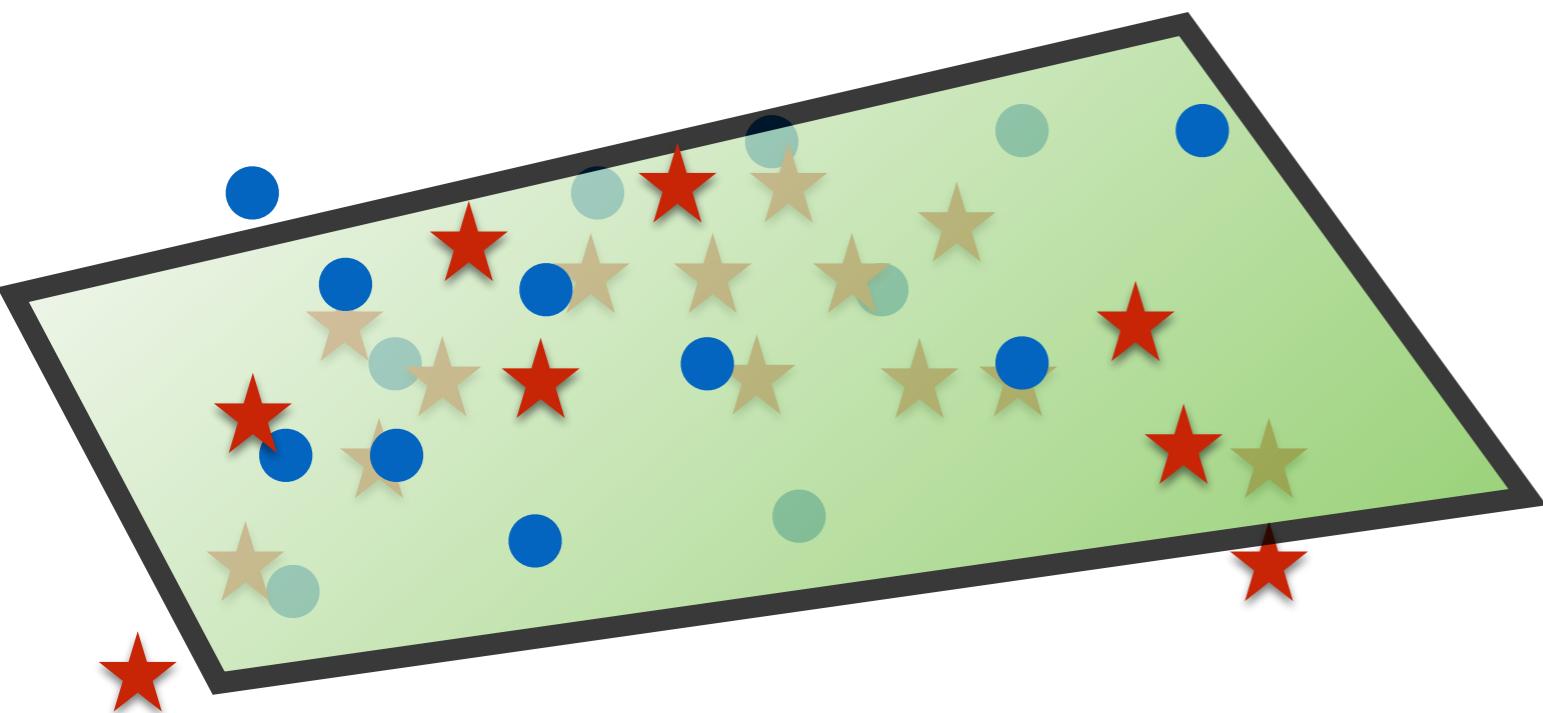
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

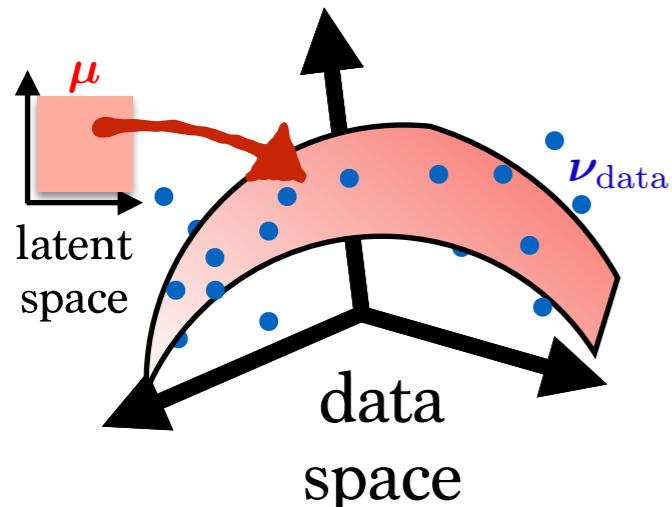
- Formulation as adversarial problem [GPM...'14]

$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



**low classification
accuracy...
is the goal.**

Another idea?



- Use a **metric** Δ for probability measures, that can handle measures with non-overlapping supports:

$$\min_{\theta \in \Theta} \Delta(\nu_{\text{data}}, p_{\theta}), \quad \text{not } \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \| p_{\theta})$$

Minimum Δ Estimation

The Annals of Statistics
1980, Vol. 8, No. 3, 457–487

MINIMUM CHI-SQUARE, NOT MAXIMUM LIKELIHOOD!

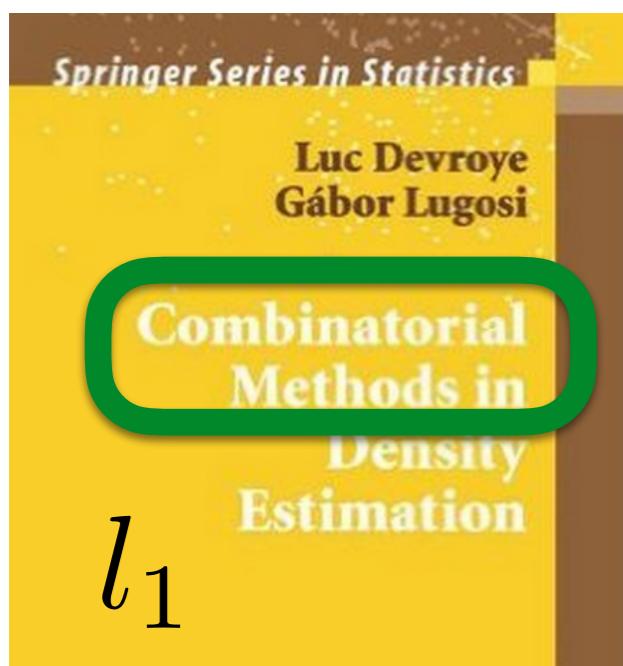
BY JOSEPH BERKSON

Mayo Clinic, Rochester, Minnesota



Computational Statistics & Data Analysis 29 (1998) 81–103

COMPUTATIONAL
STATISTICS
& DATA ANALYSIS



Minimum Hellinger distance
estimation for Poisson mixtures

Dimitris Karlis, Evdokia Xekalaki*

Department of Statistics, Athens University of Economics and Business, 76 Patission Str., 104 34 Athens, Greece

Available online at www.sciencedirect.com



Statistics & Probability Letters 76 (2006) 1298–1302

STATISTICS &
PROBABILITY
LETTERS

www.elsevier.com/locate/stapro

On minimum Kantorovich distance estimators

Federico Bassetti^a, Antonella Bodini^b, Eugenio Regazzini^{a,*}

Minimum Kantorovich Estimation



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Statistics & Probability Letters 76 (2006) 1298–1302

STATISTICS &
PROBABILITY
LETTERS

www.elsevier.com/locate/stapro

On minimum Kantorovich distance estimators

Federico Bassetti^a, Antonella Bodini^b, Eugenio Regazzini^{a,*}

Use *Wasserstein distances* to define a loss
between data and model.

$$\min_{\theta \in \Theta} W(\nu_{\text{data}}, p_{\theta})$$

Minimum Kantorovich Estimators

$$\min_{\theta \in \Theta} W(\nu_{\text{data}}, f_{\theta \sharp} \mu)$$

[Bassetti'06] 1st reference discussing this approach.

Challenge: $\nabla_{\theta} W(\nu_{\text{data}}, f_{\theta \sharp} \mu)$?

[Montavon'16] use regularized OT in a finite setting.

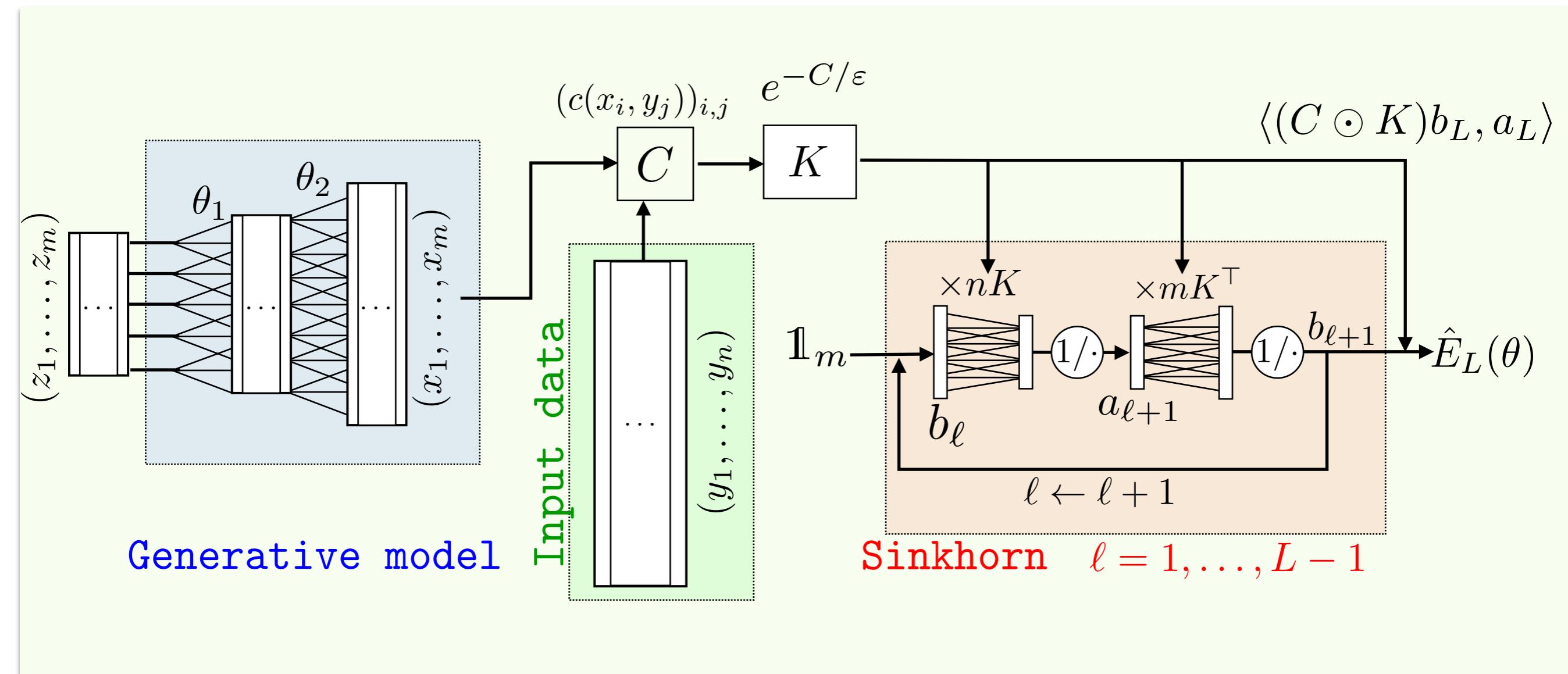
[Arjovsky'17] (WGAN) uses a NN to approximate dual solutions and recover gradient w.r.t. parameter

[Bernton'17] (*Wasserstein ABC*)

[Genevay'17, Salimans'17] (*Sinkhorn approach*)

Proposal: Autodiff OT using Sinkhorn

Approximate W loss by the transport cost \bar{W}_L after L Sinkhorn iterations.



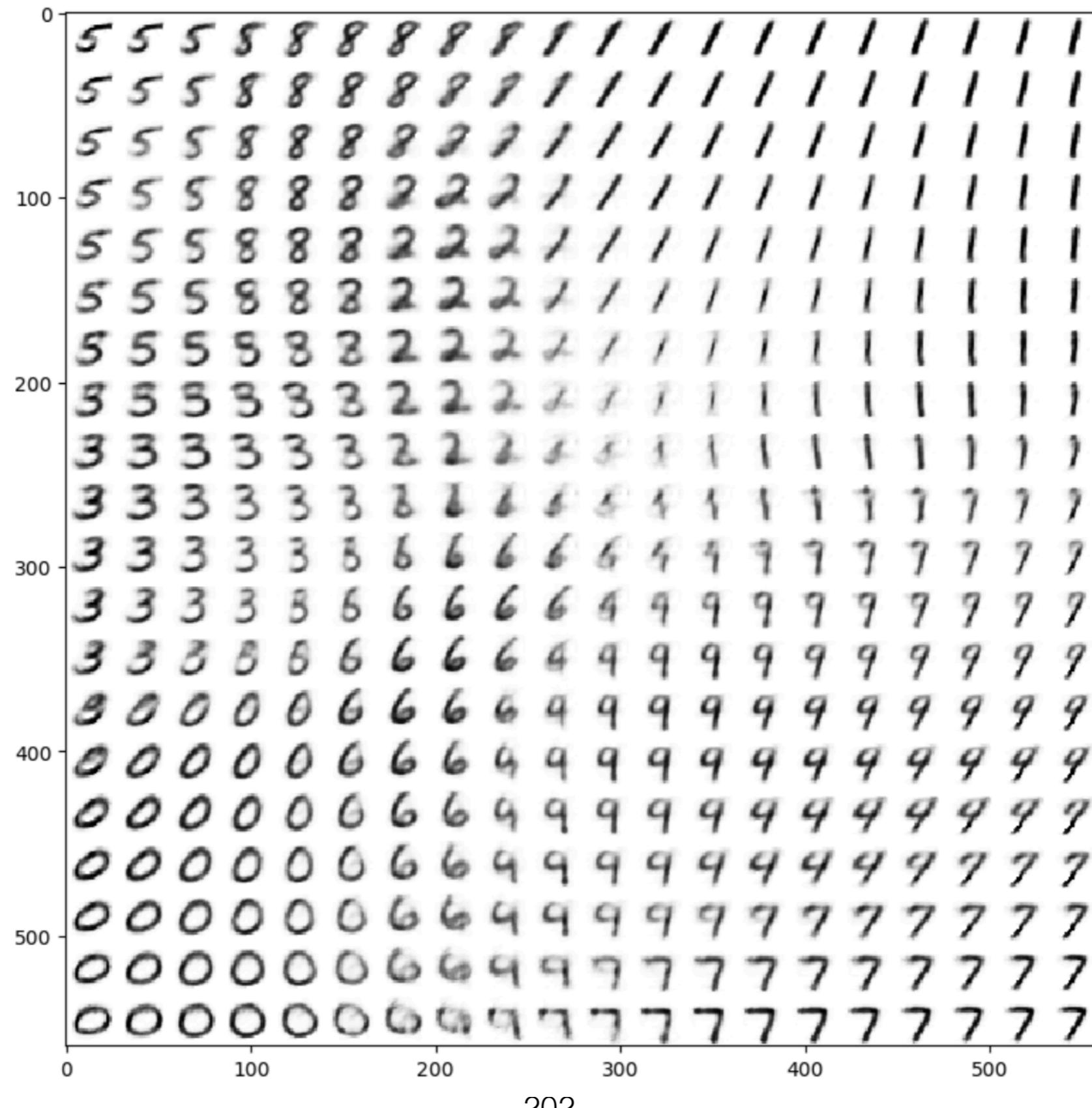
Example: MNIST, Learning f_{θ}

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

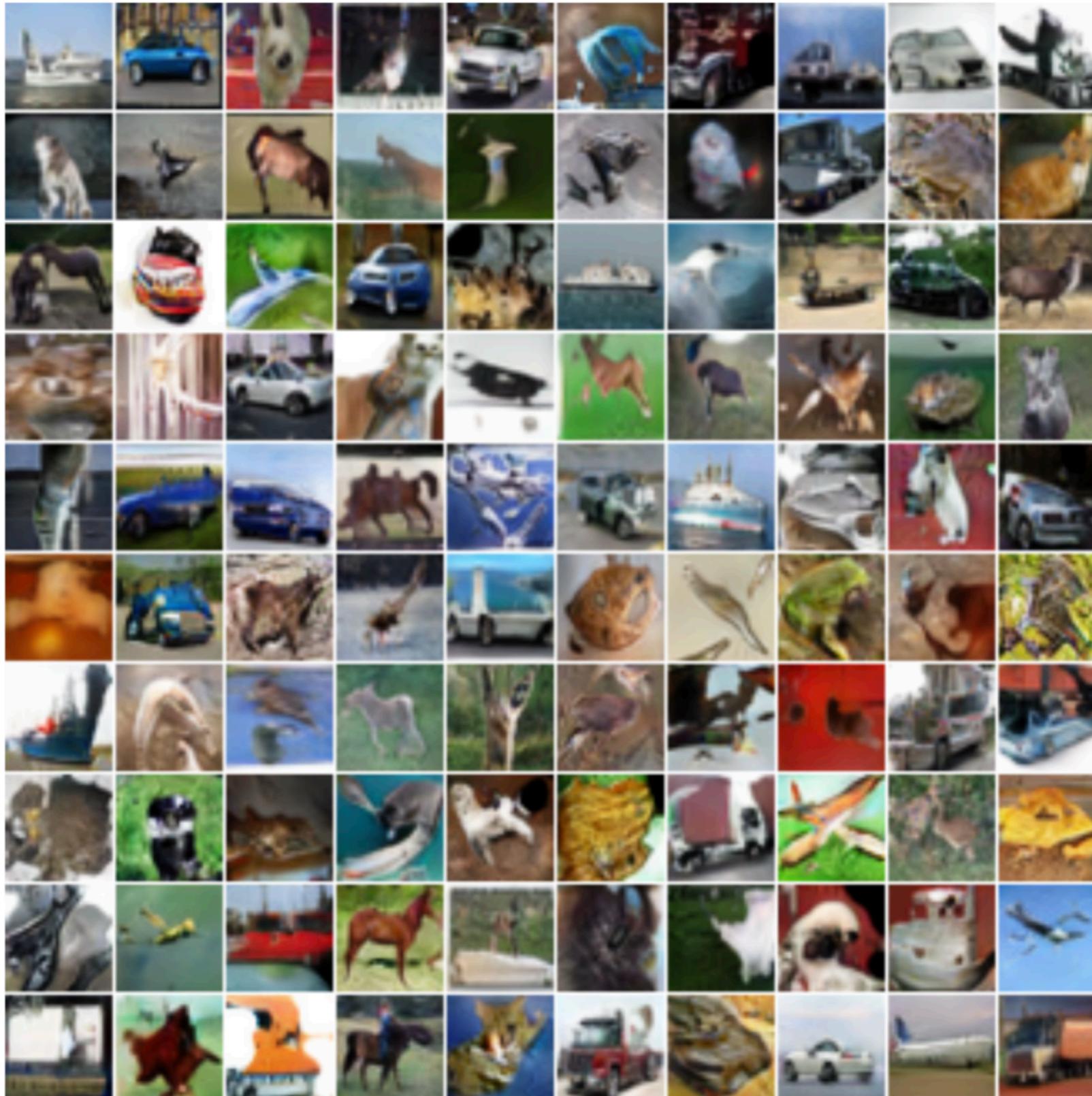
Example: MNIST, Learning f_{θ}

Latent
space

$[0, 1]^2$



Example: Generation of Images



Example: Generation of Images



Concluding Remarks

- *Regularization* is required for OT to work on data.
- If one expects that these tools become widely adopted, they must be “*auto-differentiable*”.
- **Many open problems remain!**

What I could not talk about...

- Very large supply of **maths**...
- **Statistical** challenges to compute W .
- If **linear assignment** = Wasserstein, then
quadratic assignment = Gromov-Wasserstein.
- Wasserstein gradient flows (a.k.a. **JKO** flow).
- **Dynamical** aspects of optimal transport
- Transporting vectors and matrices