

A Primer on Optimal Transport

Marco Cuturi



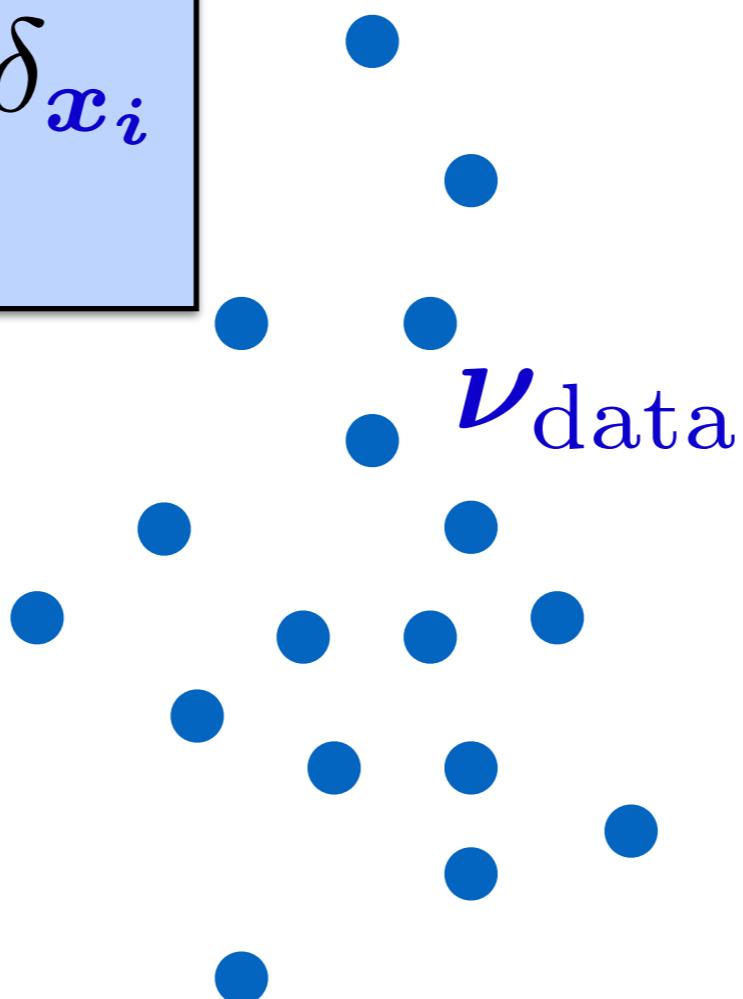
book with Gabriel Peyré (Ecole Normale Supérieure)

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

A Motivating Example

We collect data

$$\nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_i}$$



A Motivating Example

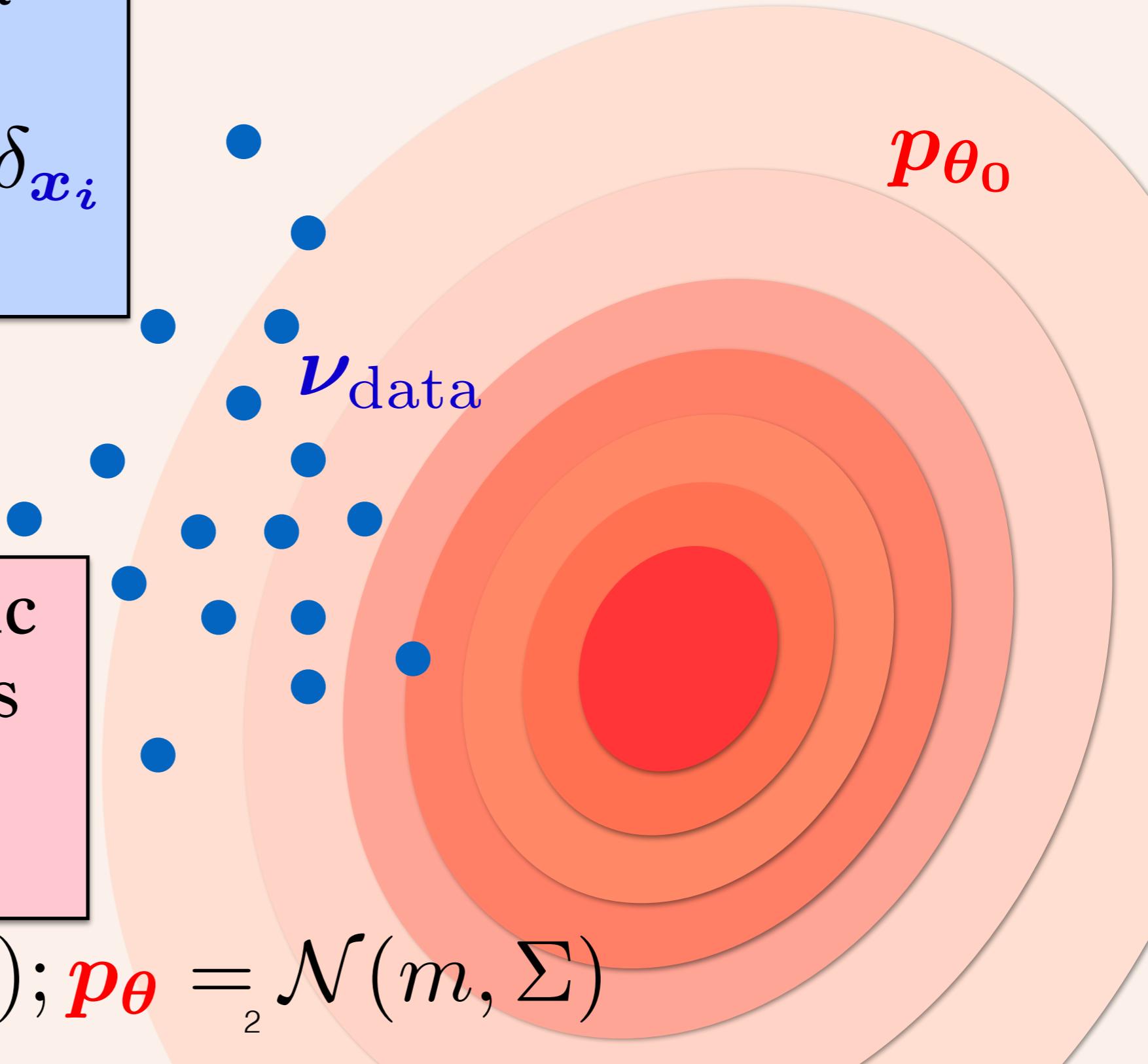
We collect data

$$\nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_i}$$

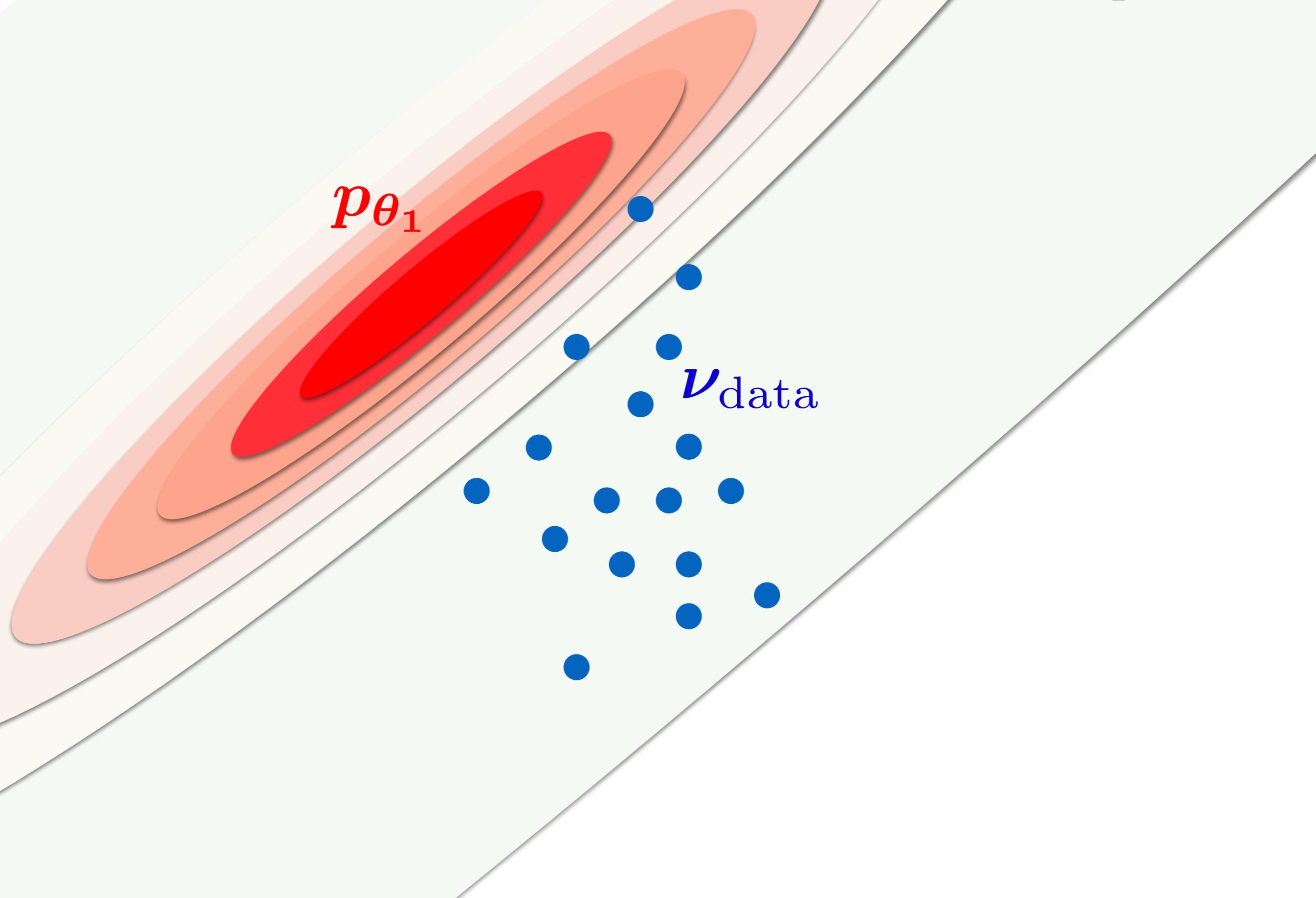
We fit a parametric family of densities

$$\{p_{\theta}, \theta \in \Theta\}$$

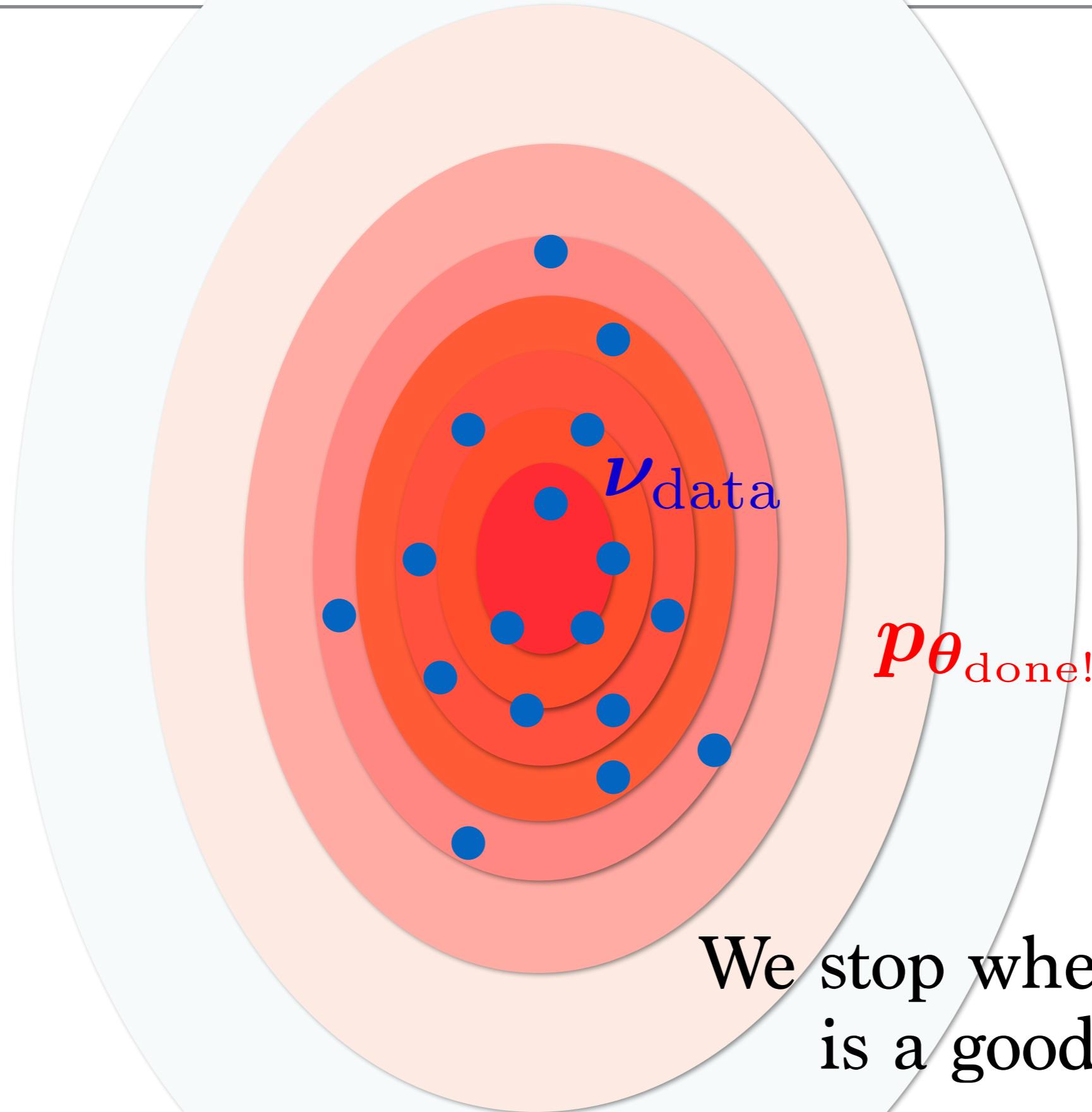
e.g. $\theta = (m, \Sigma)$; $p_{\theta} = \mathcal{N}_2(m, \Sigma)$



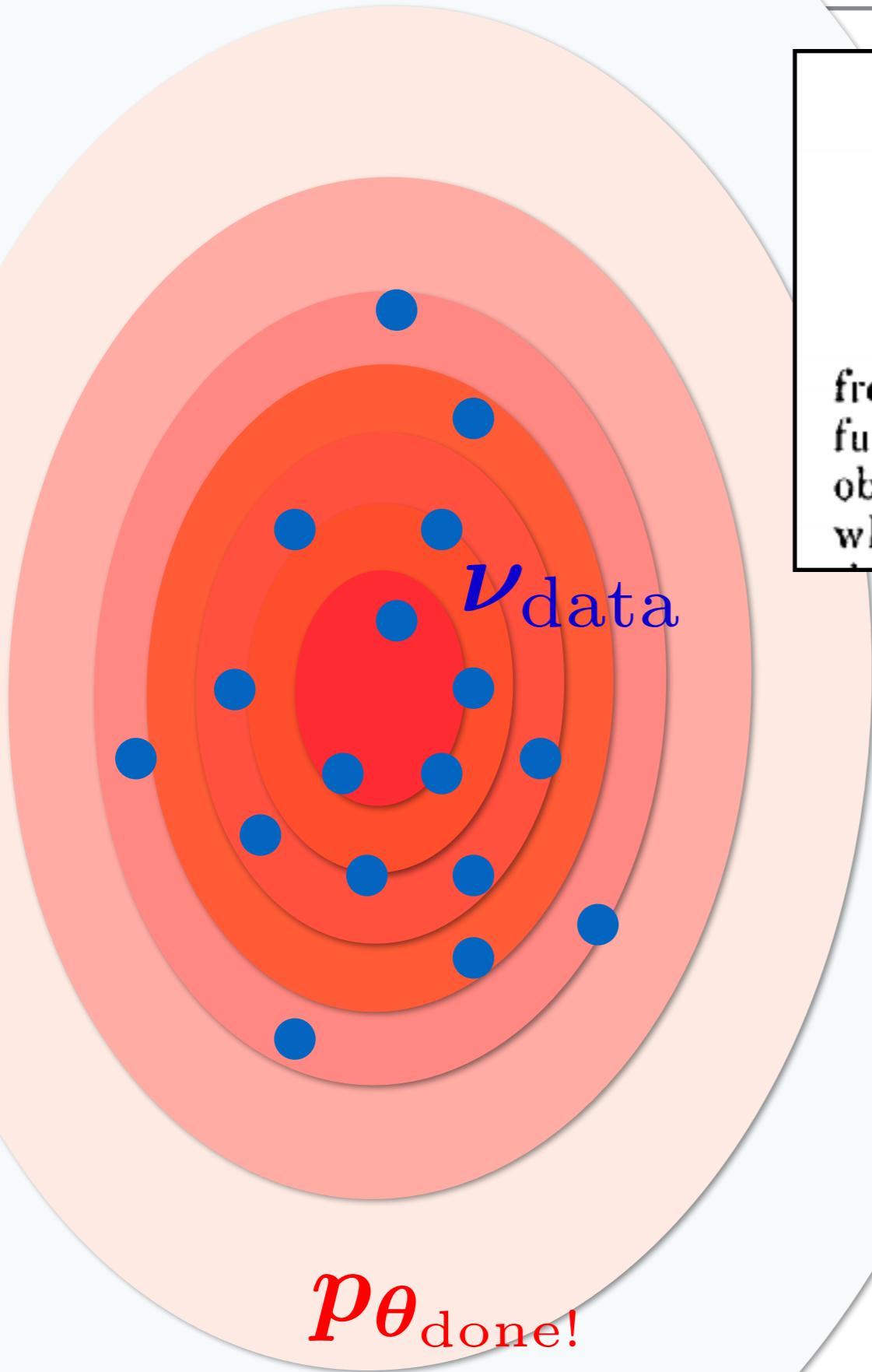
Statistics 0.1: Density Fitting



Statistics 0.1: Density Fitting



Maximum Likelihood Estimation



ON AN ABSOLUTE CRITERION
FOR FITTING FREQUENCY CURVES.

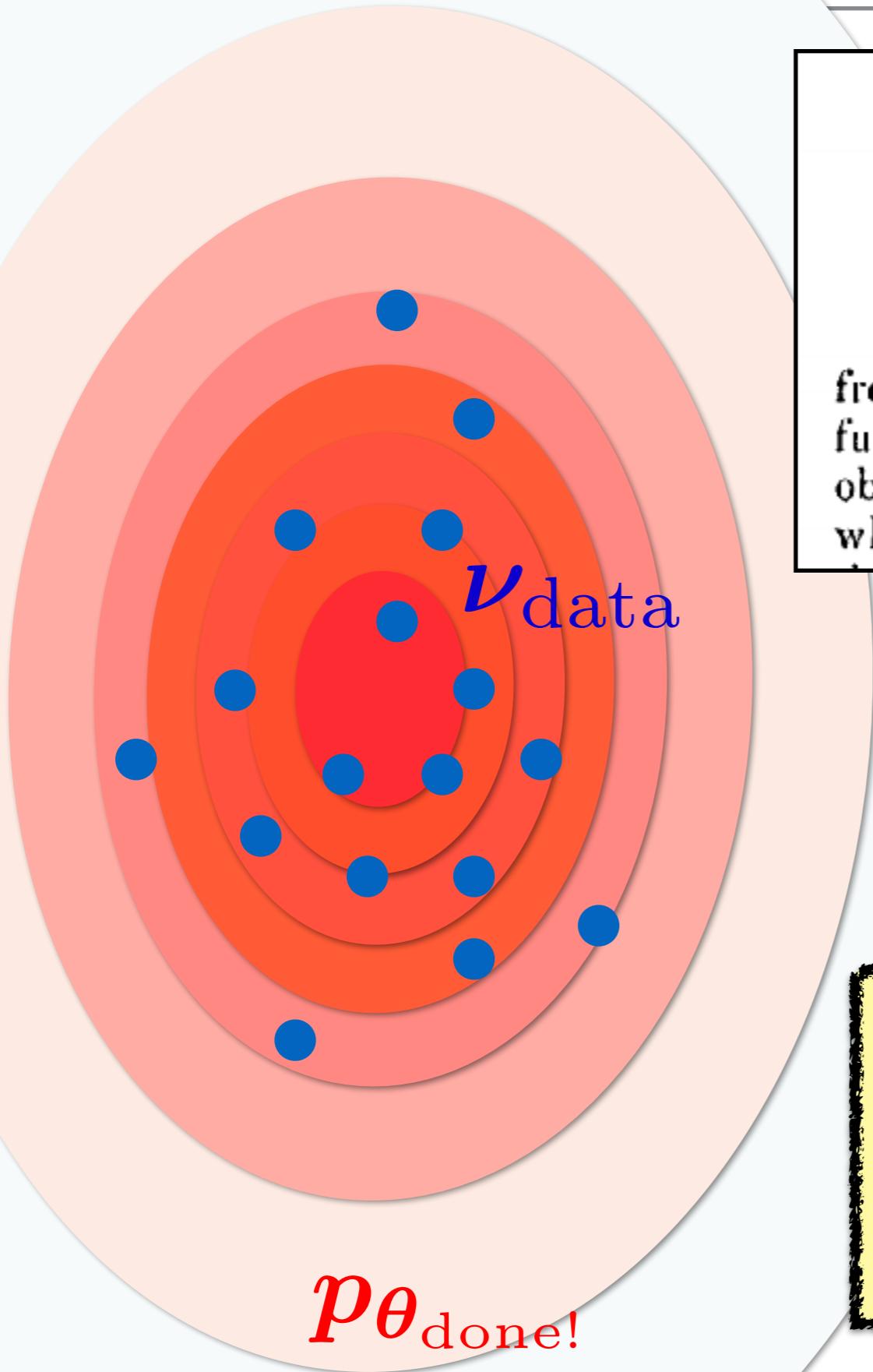
By *R. A. Fisher*, Gonville and Caius College, Cambridge.

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



$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i)$$

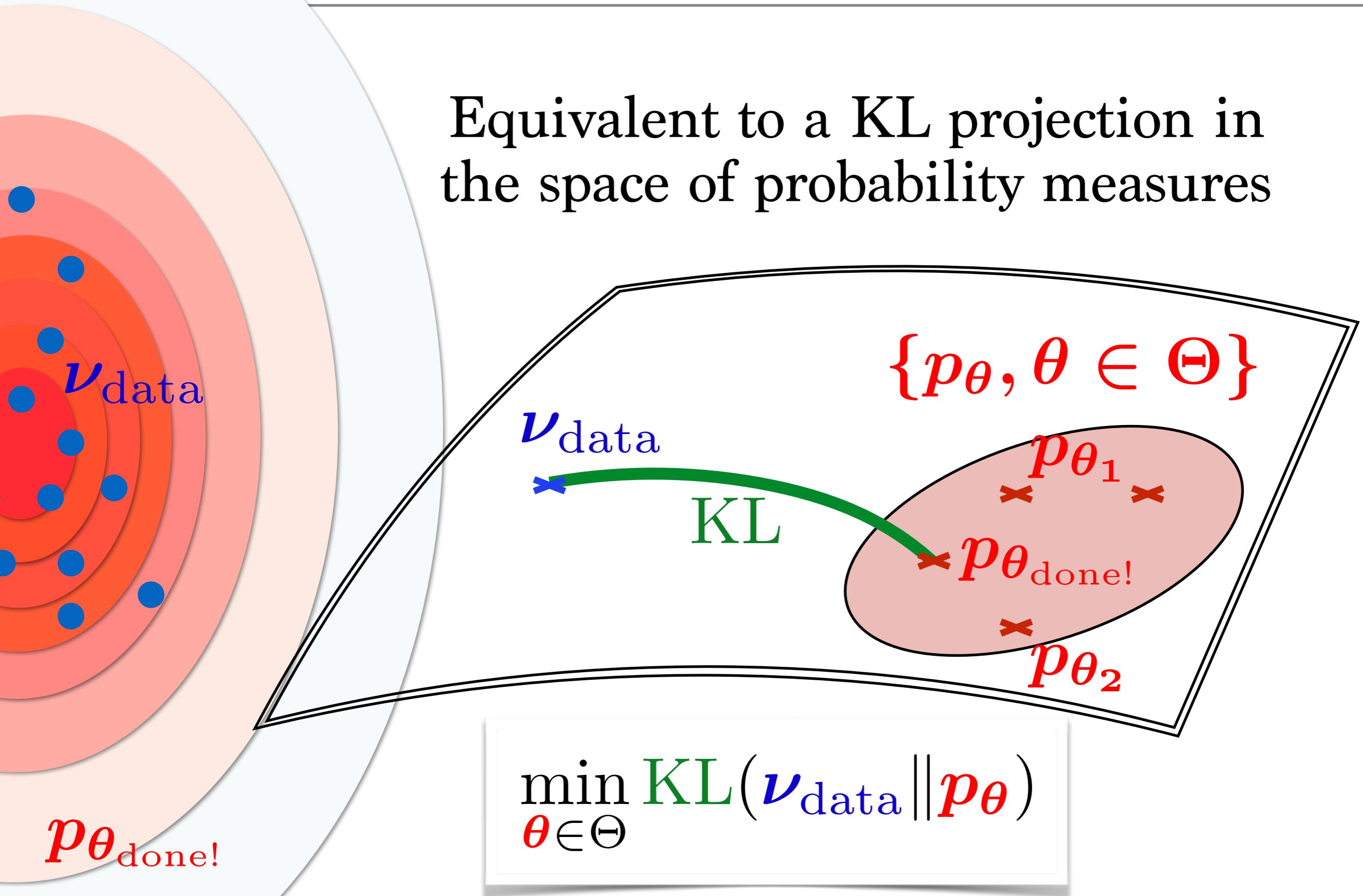


$$\log 0 = -\infty$$

$p_{\theta}(x_i)$ must be > 0

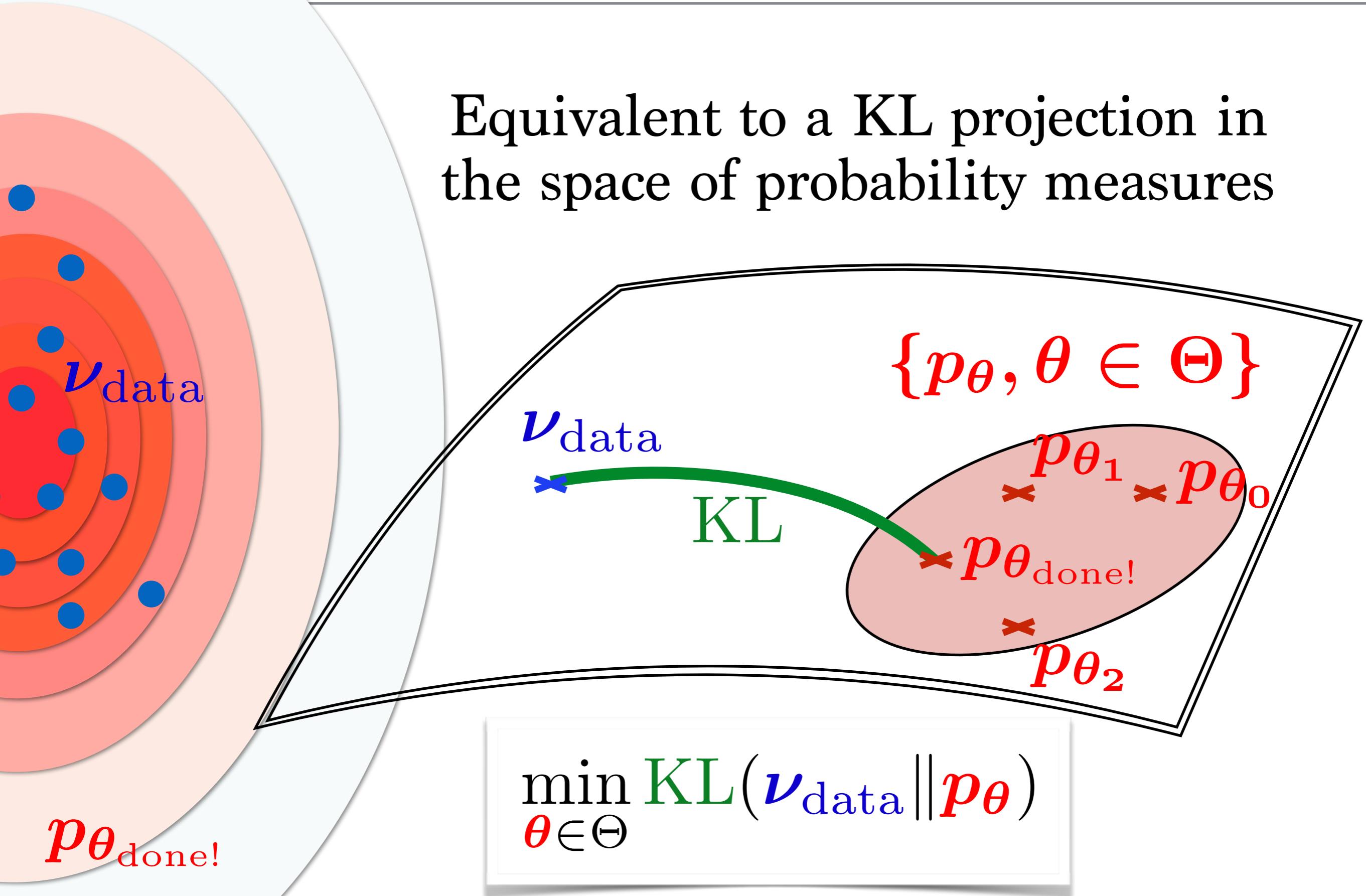
Maximum Likelihood Estimation

Equivalent to a KL projection in the space of probability measures

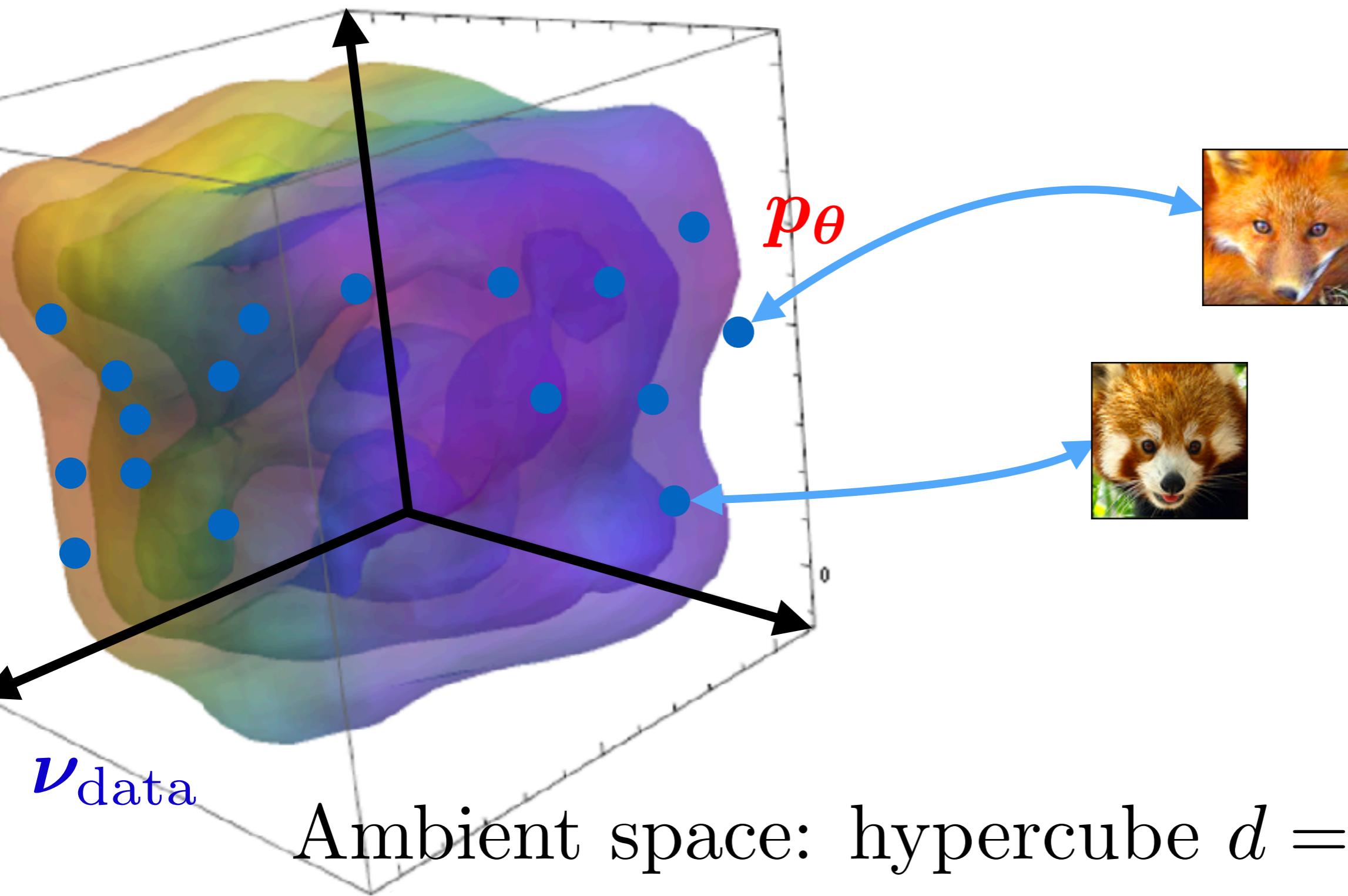


Maximum Likelihood Estimation

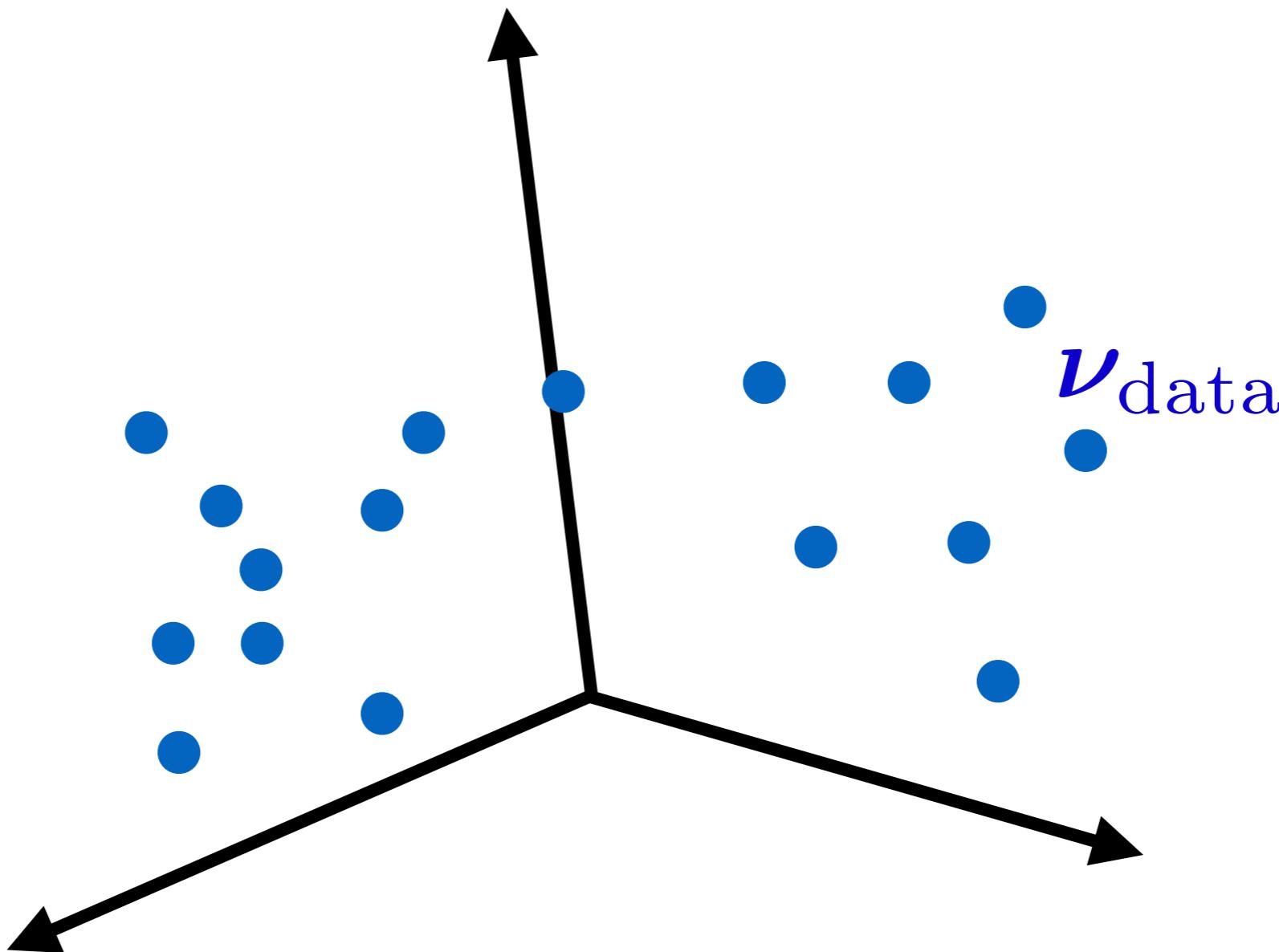
Equivalent to a KL projection in the space of probability measures



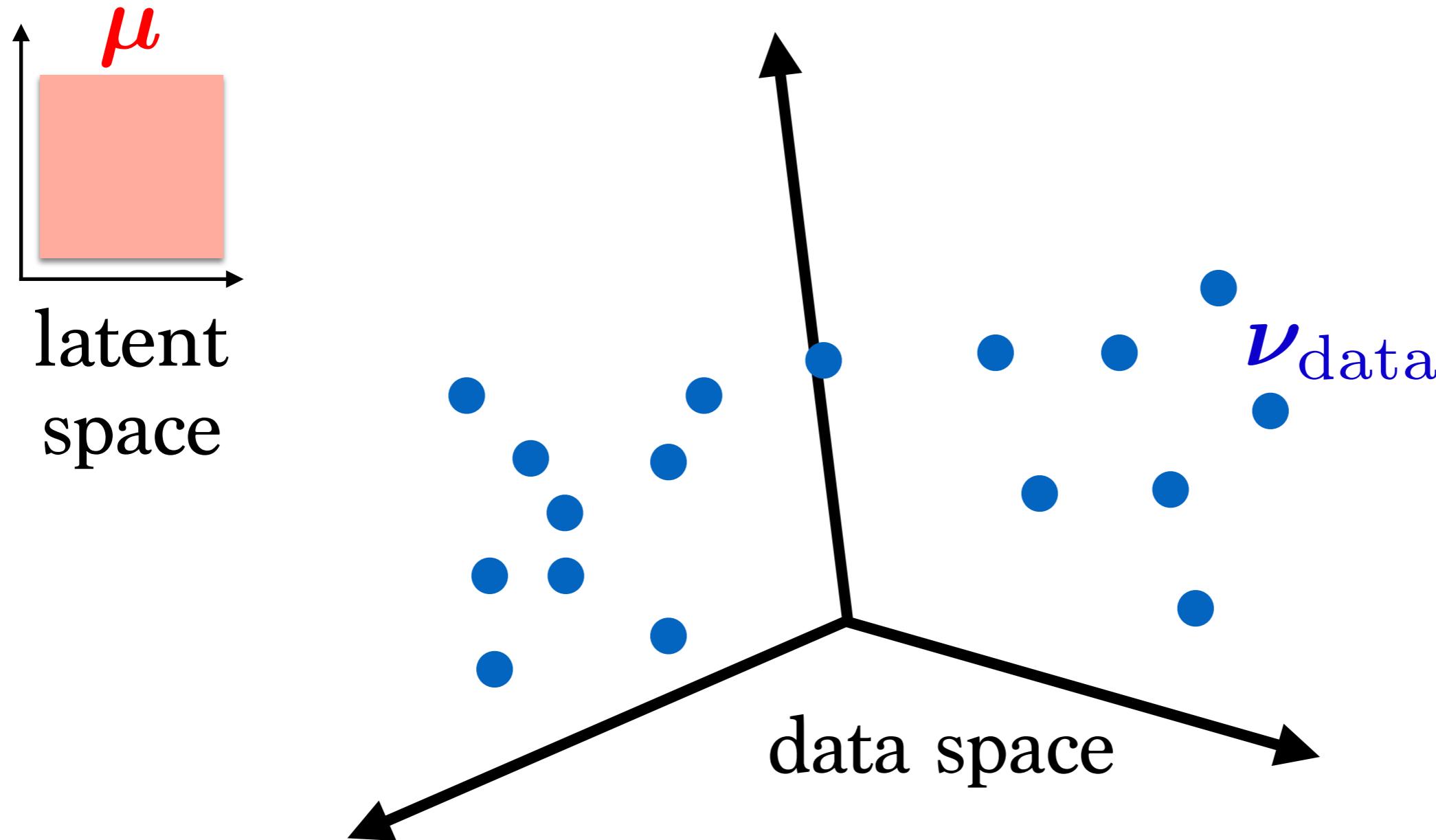
In higher dimensional spaces...



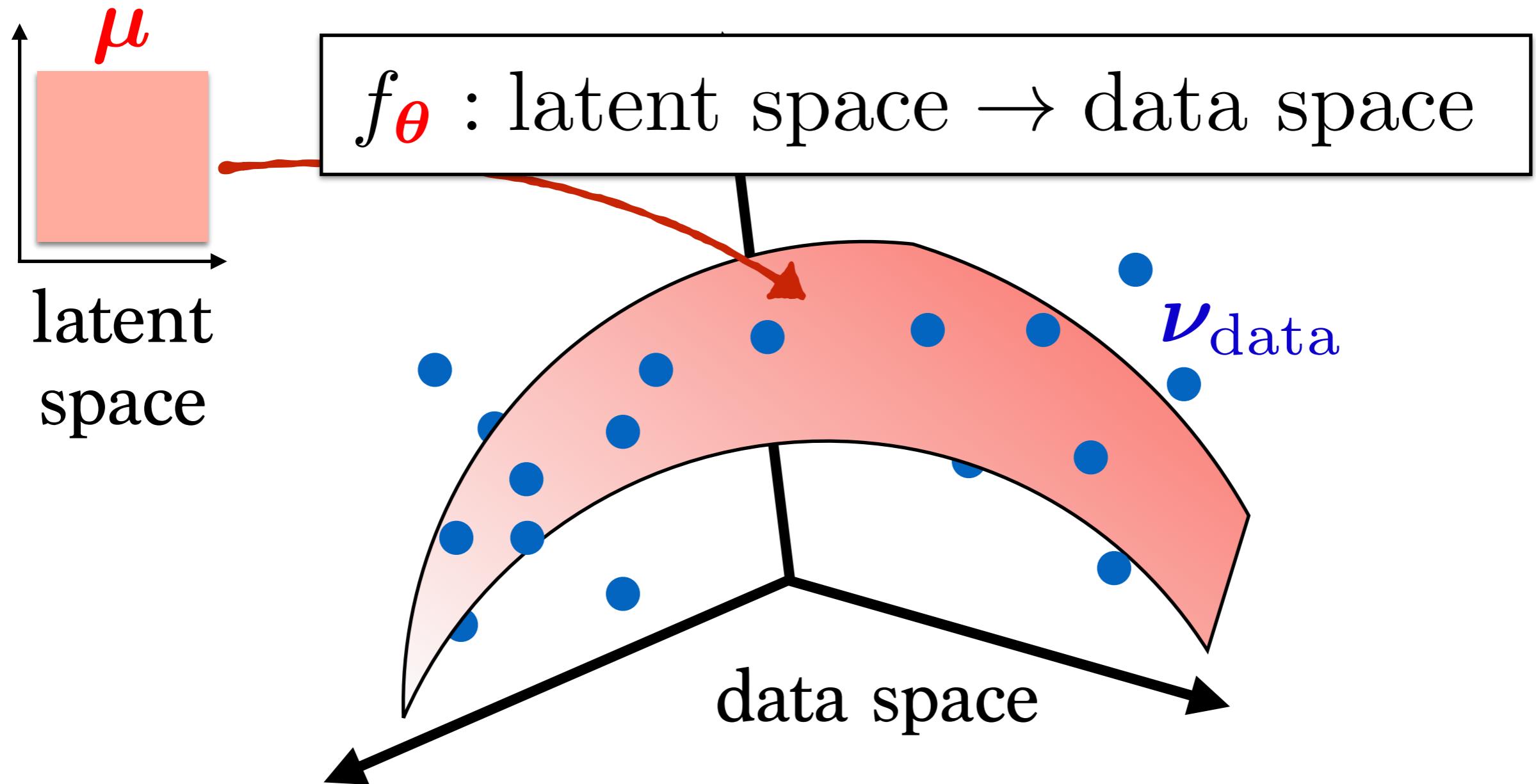
Generative Models



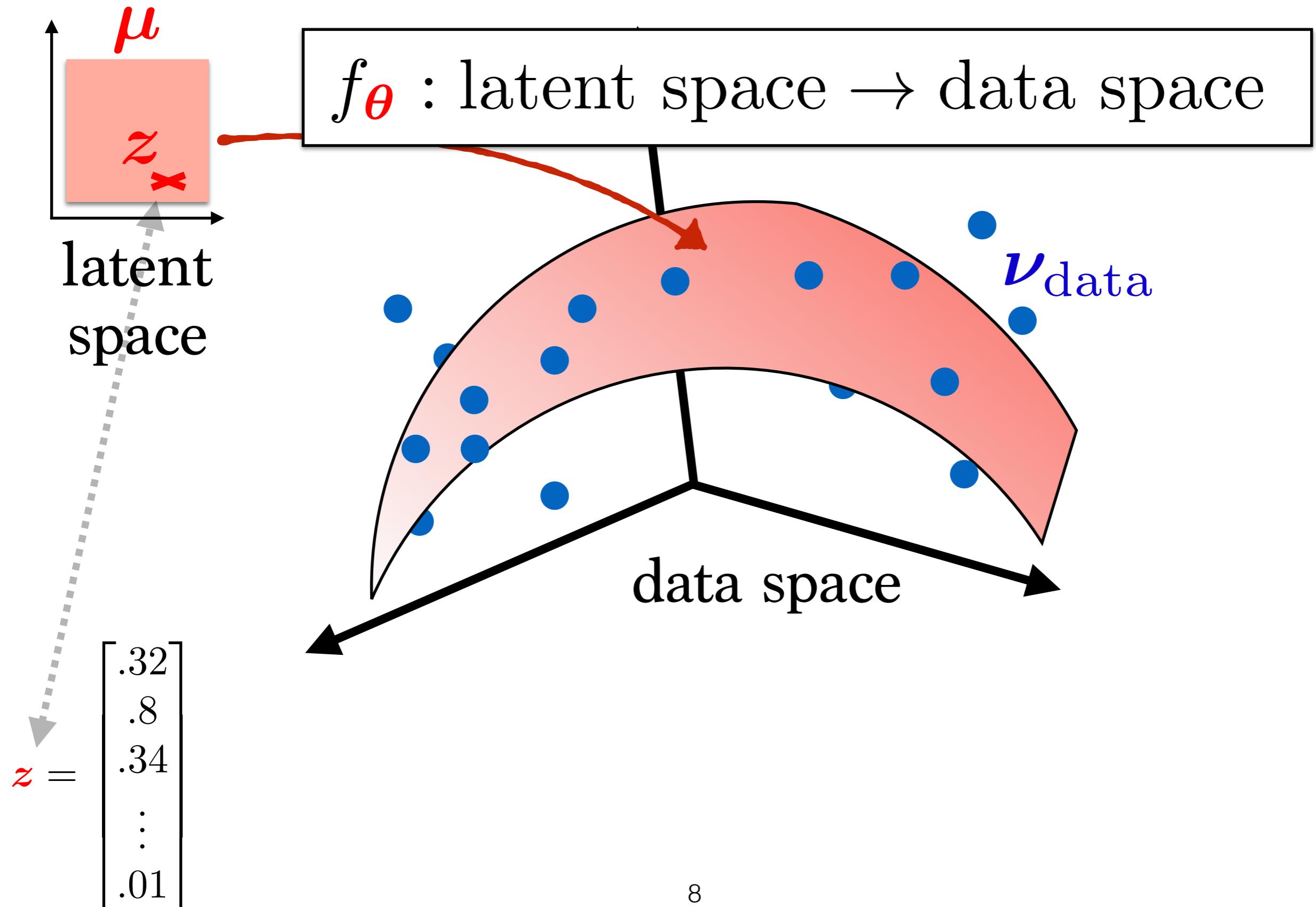
Generative Models



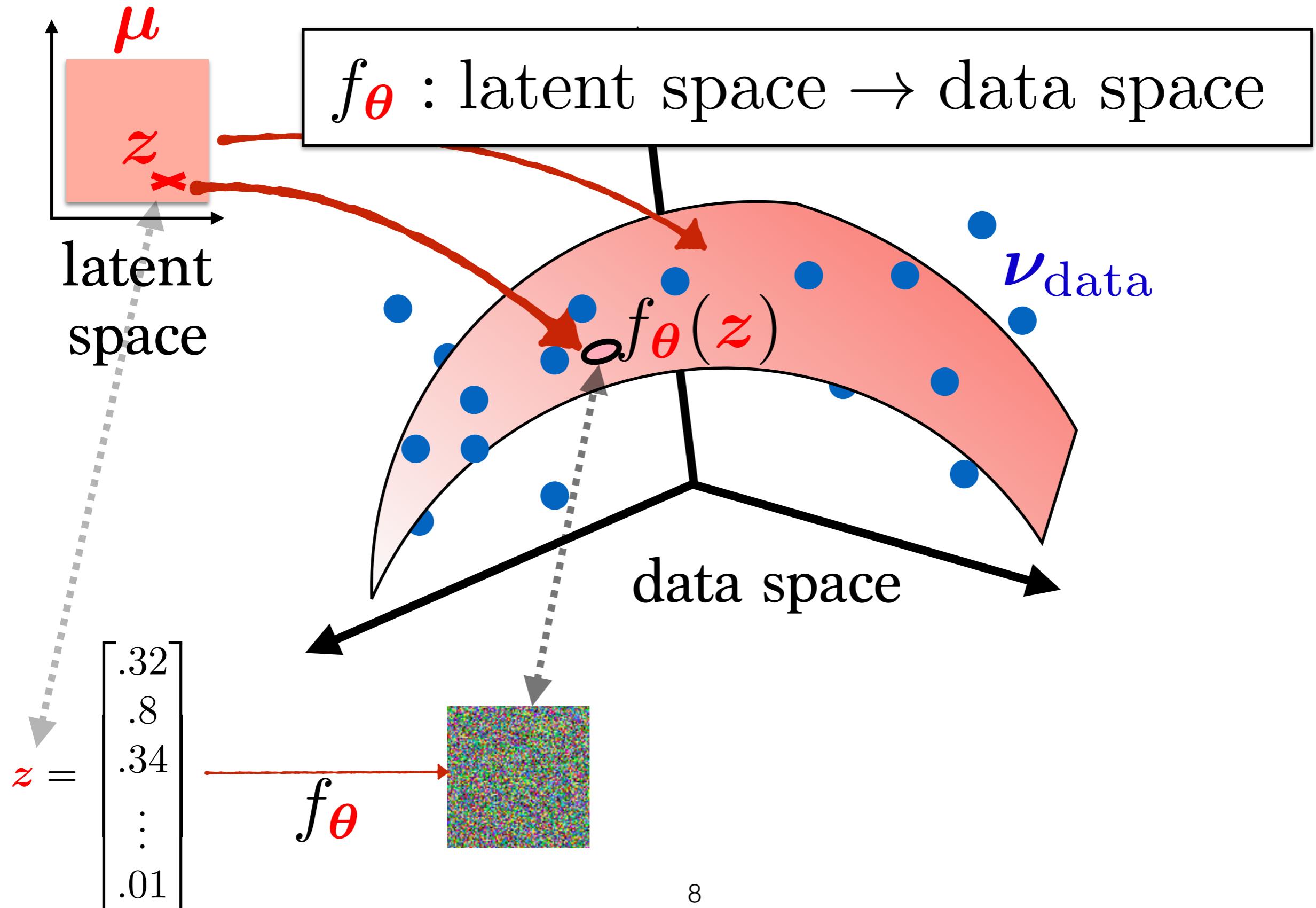
Generative Models



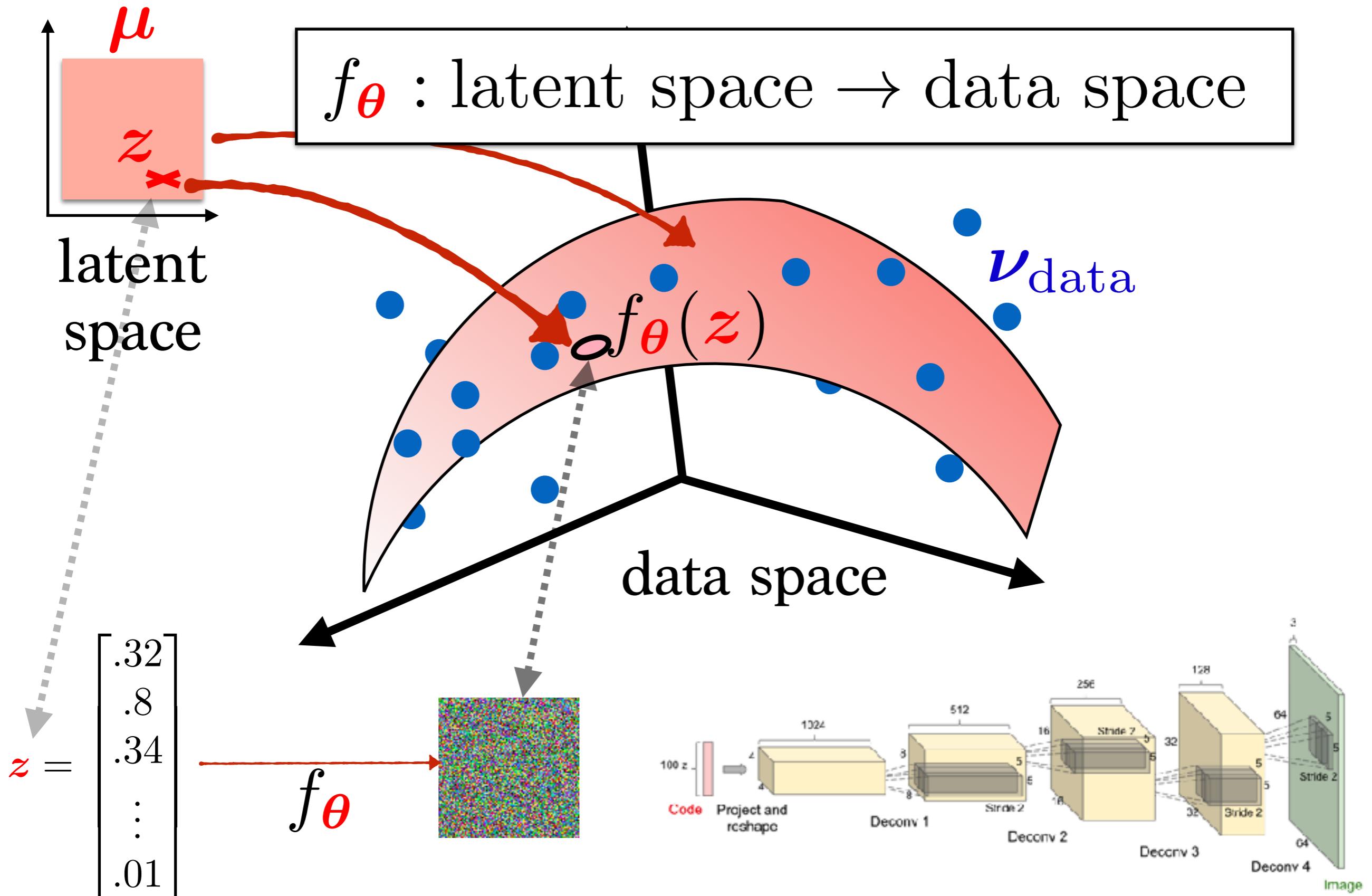
Generative Models



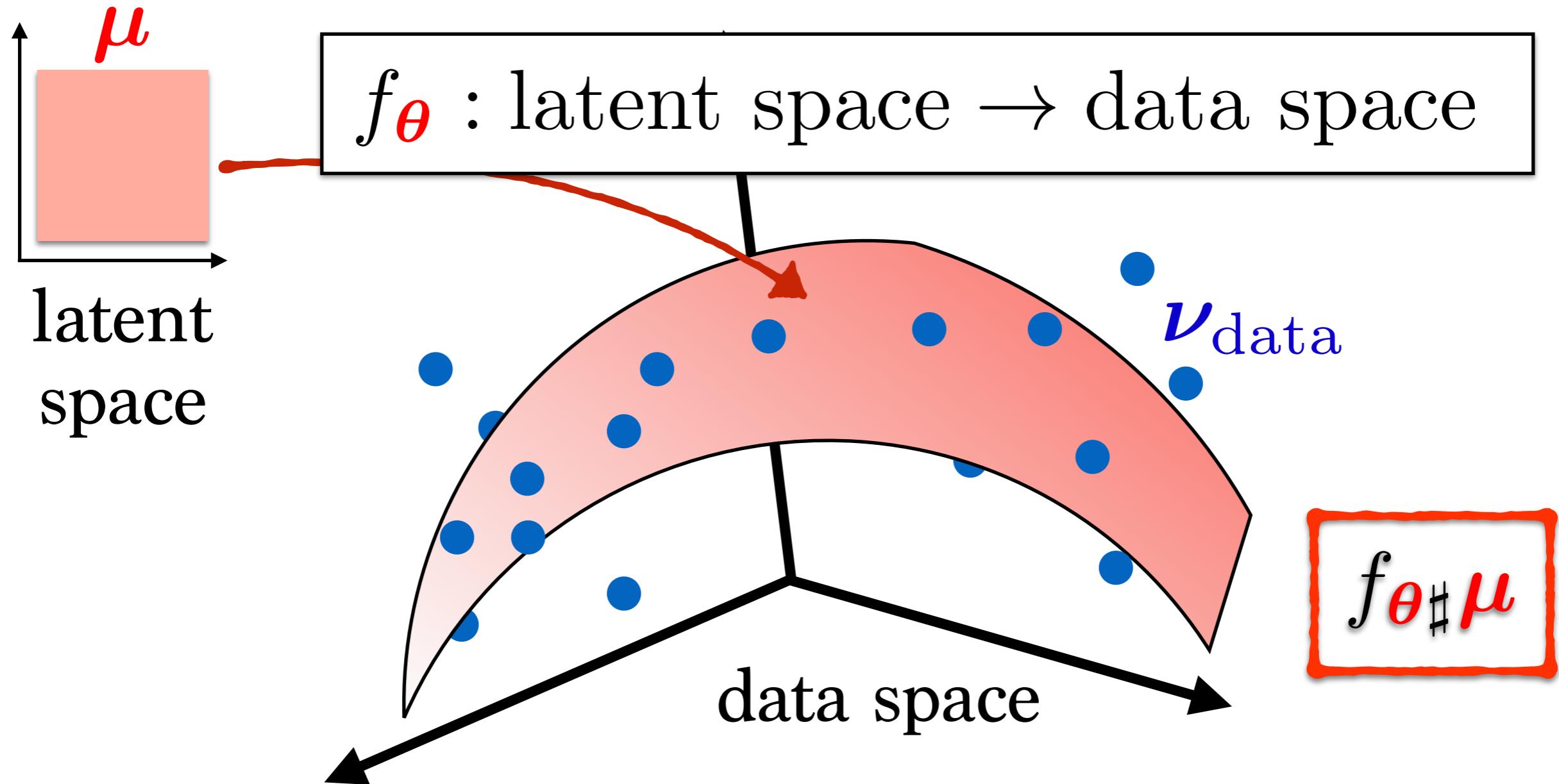
Generative Models



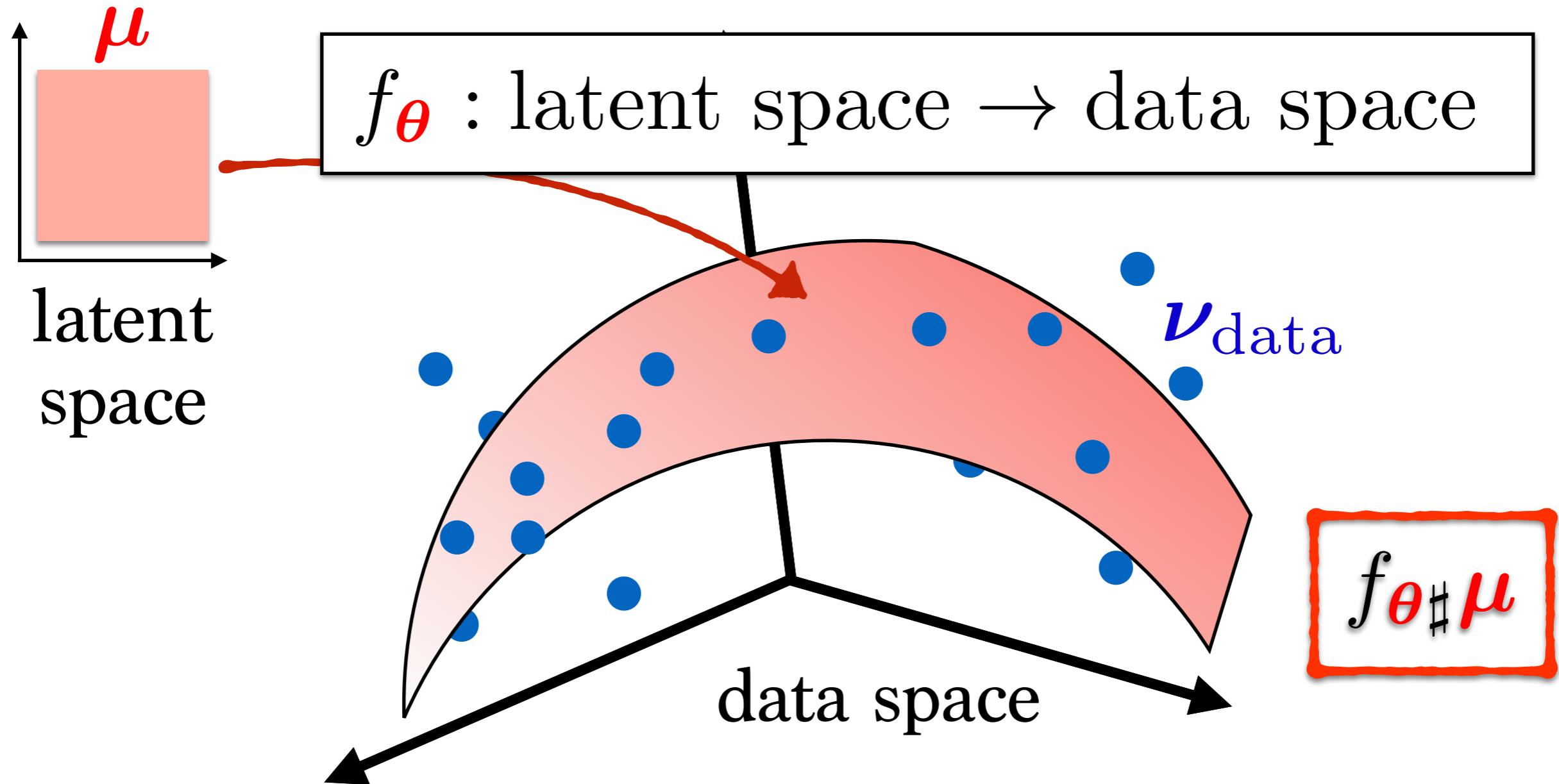
Generative Models



Generative Models

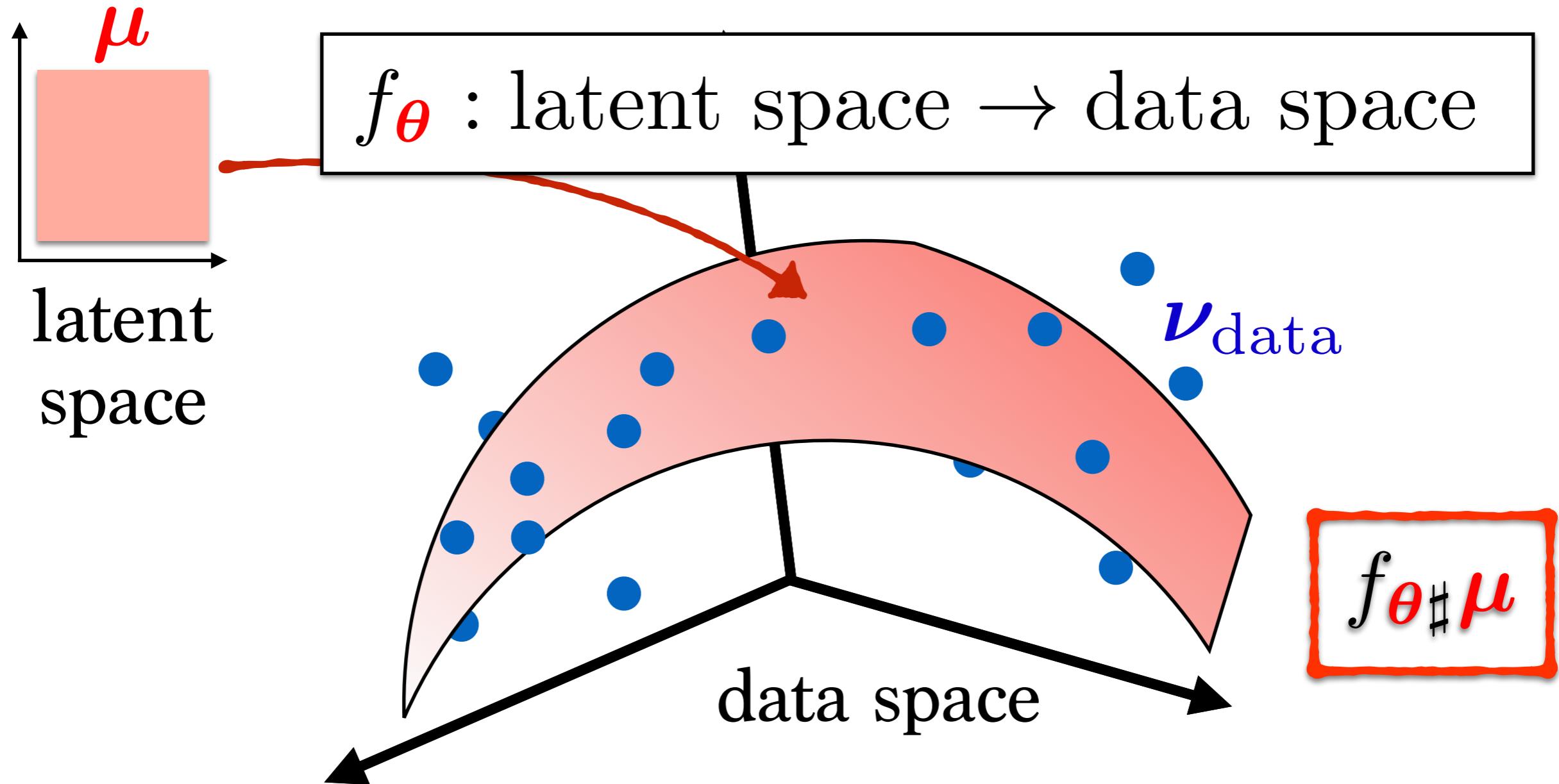


Generative Models



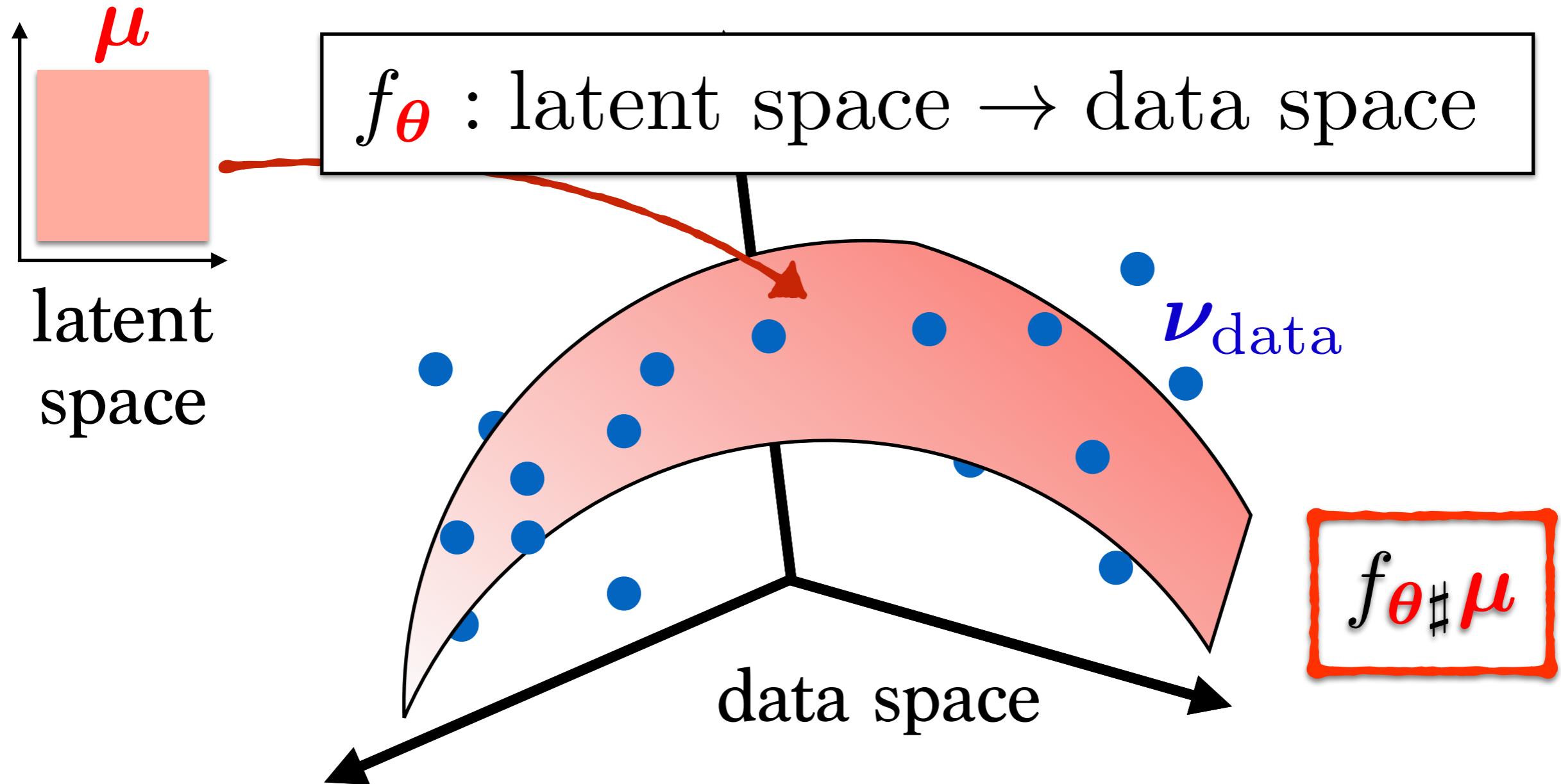
Goal: find θ such that $f_{\theta \sharp} \mu$ fits ν_{data}

Generative Models



Goal: find θ such that $f_{\theta \sharp} \mu$ fits ν_{data}

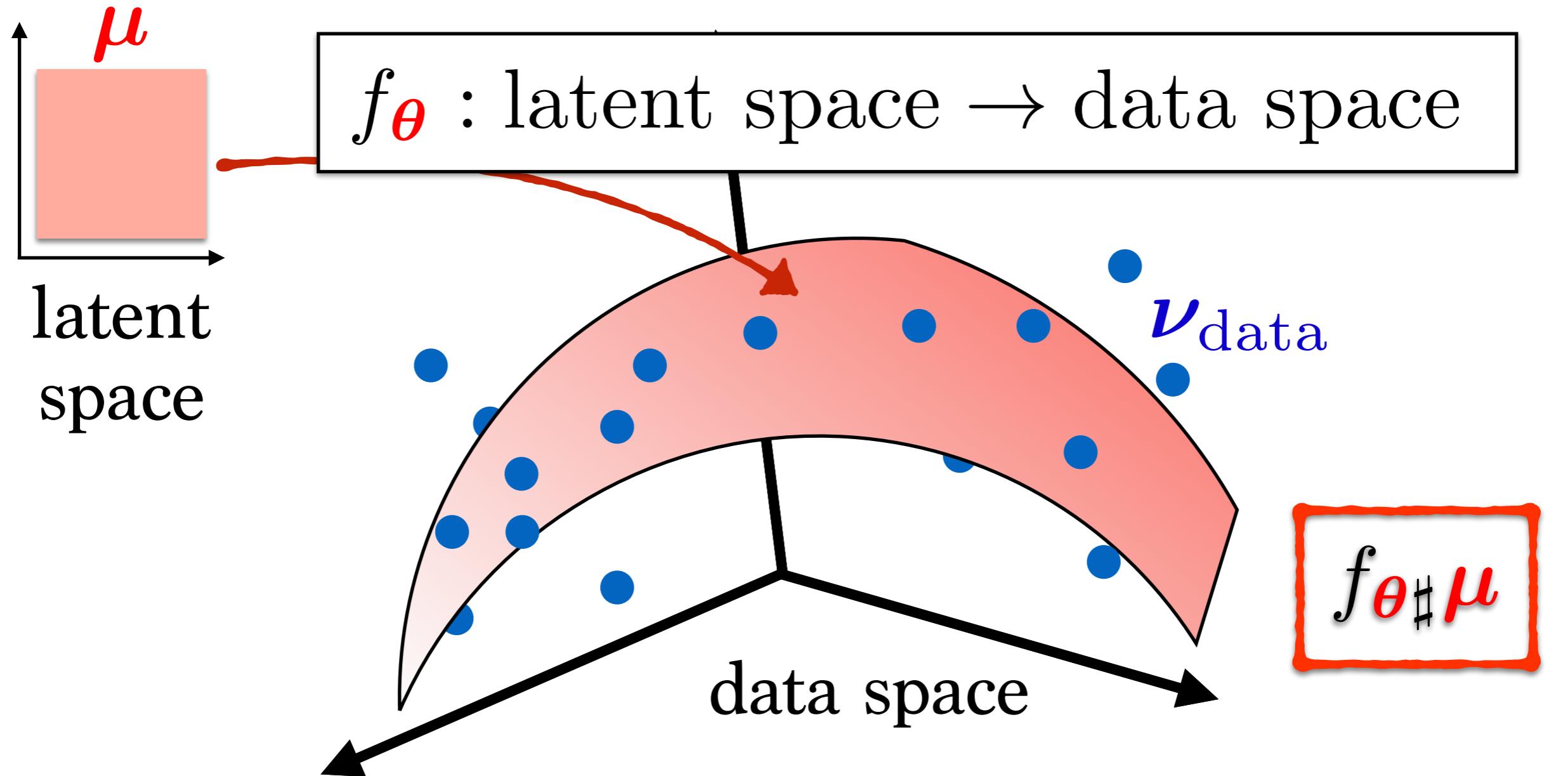
Generative Models



MLE

$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i) = \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

Generative Models

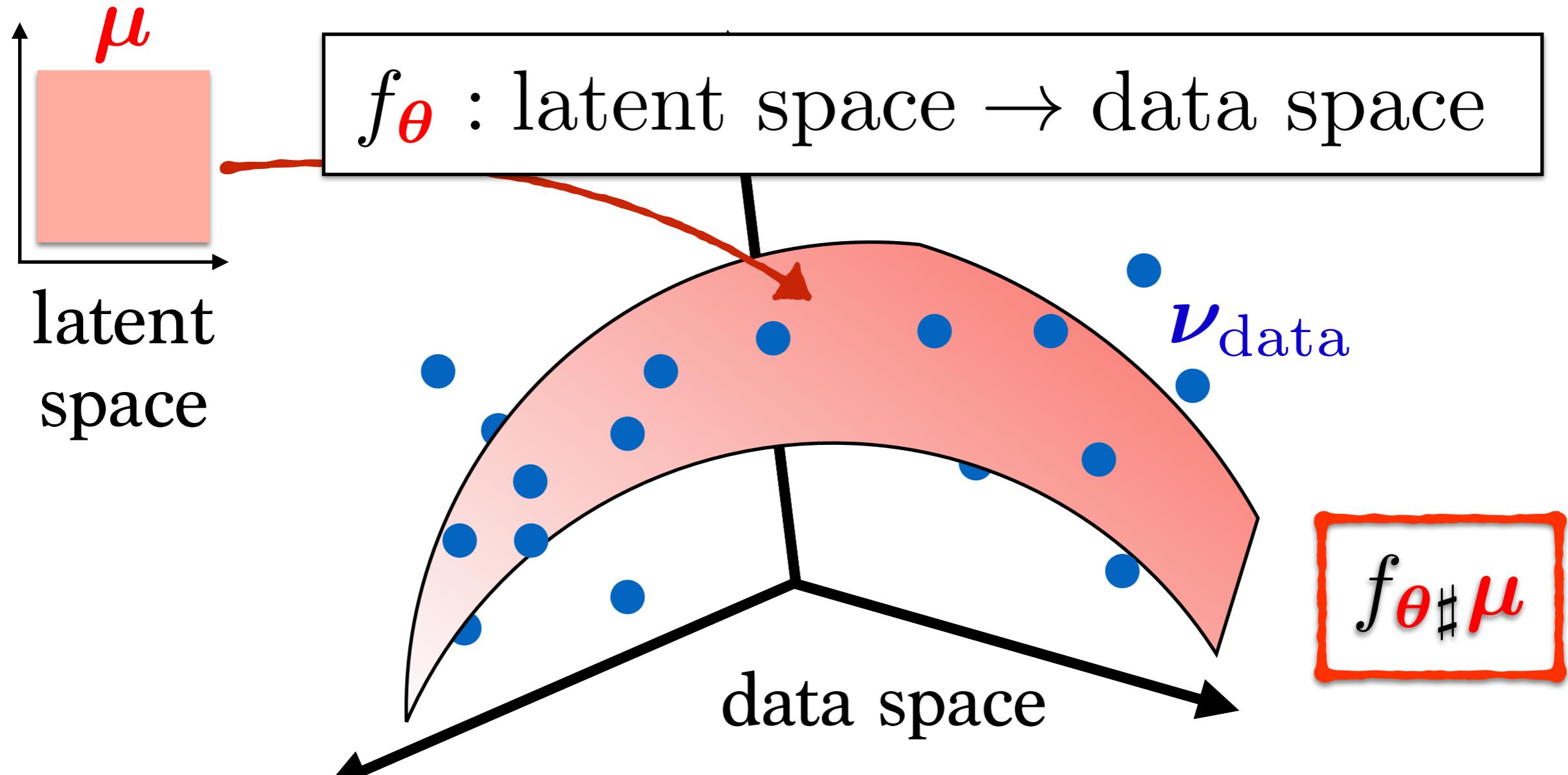


MLE

$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log \underline{f_{\theta \sharp} \mu}(x_i)$$

$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel \underline{f_{\theta \sharp} \mu})$$

Generative Models

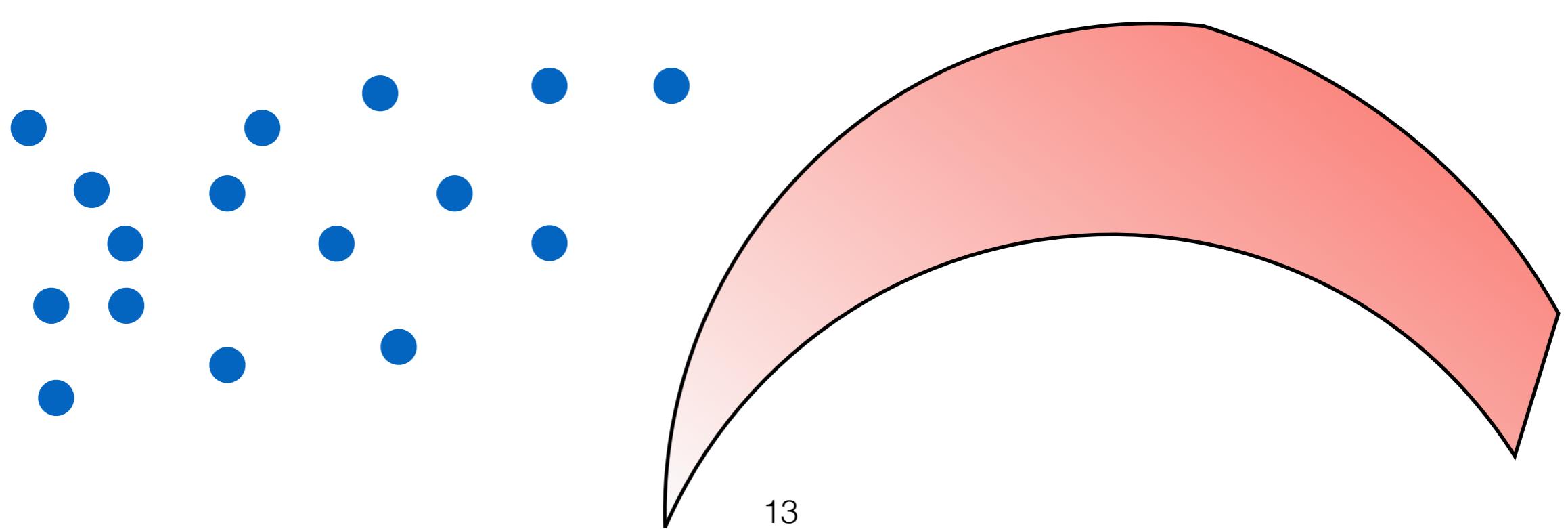


Need a more flexible discrepancy function to compare ν_{data} and $f_{\theta\sharp}\mu$

Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

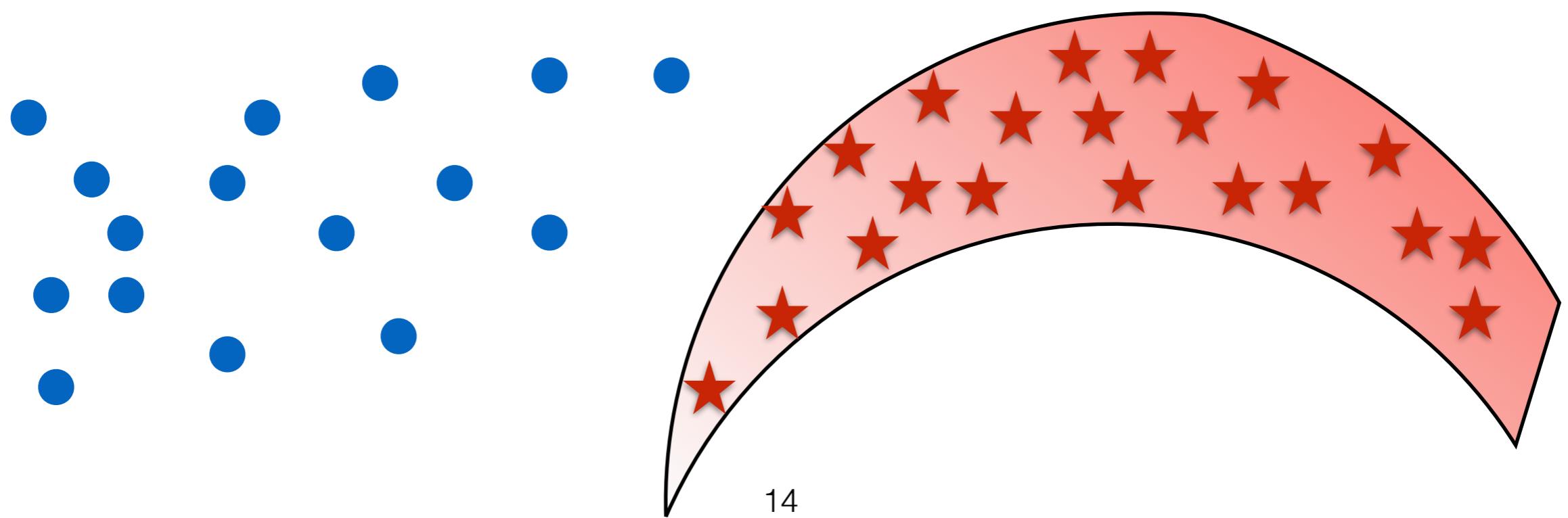
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

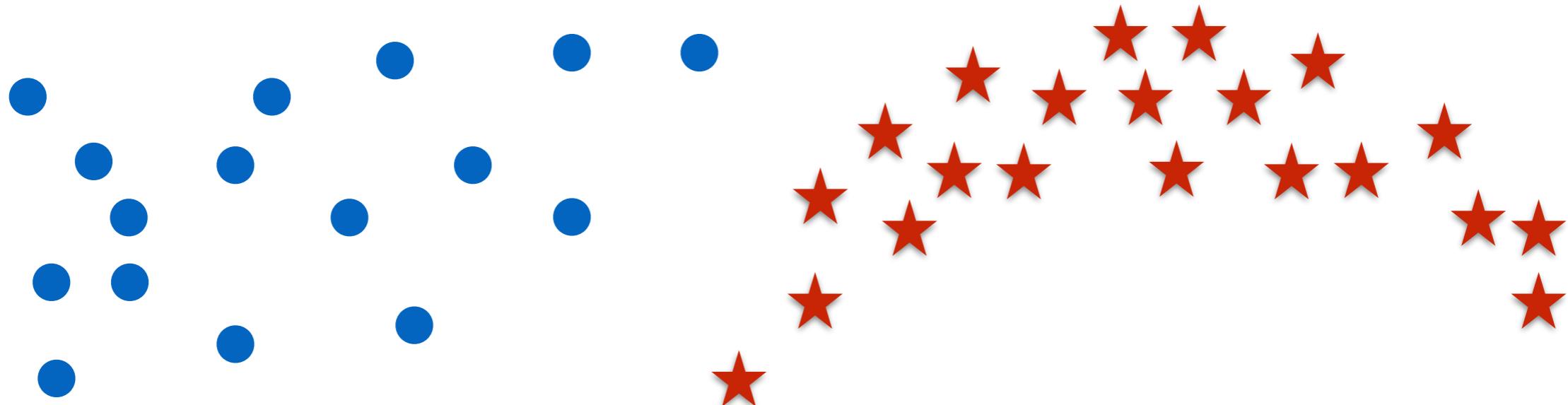
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

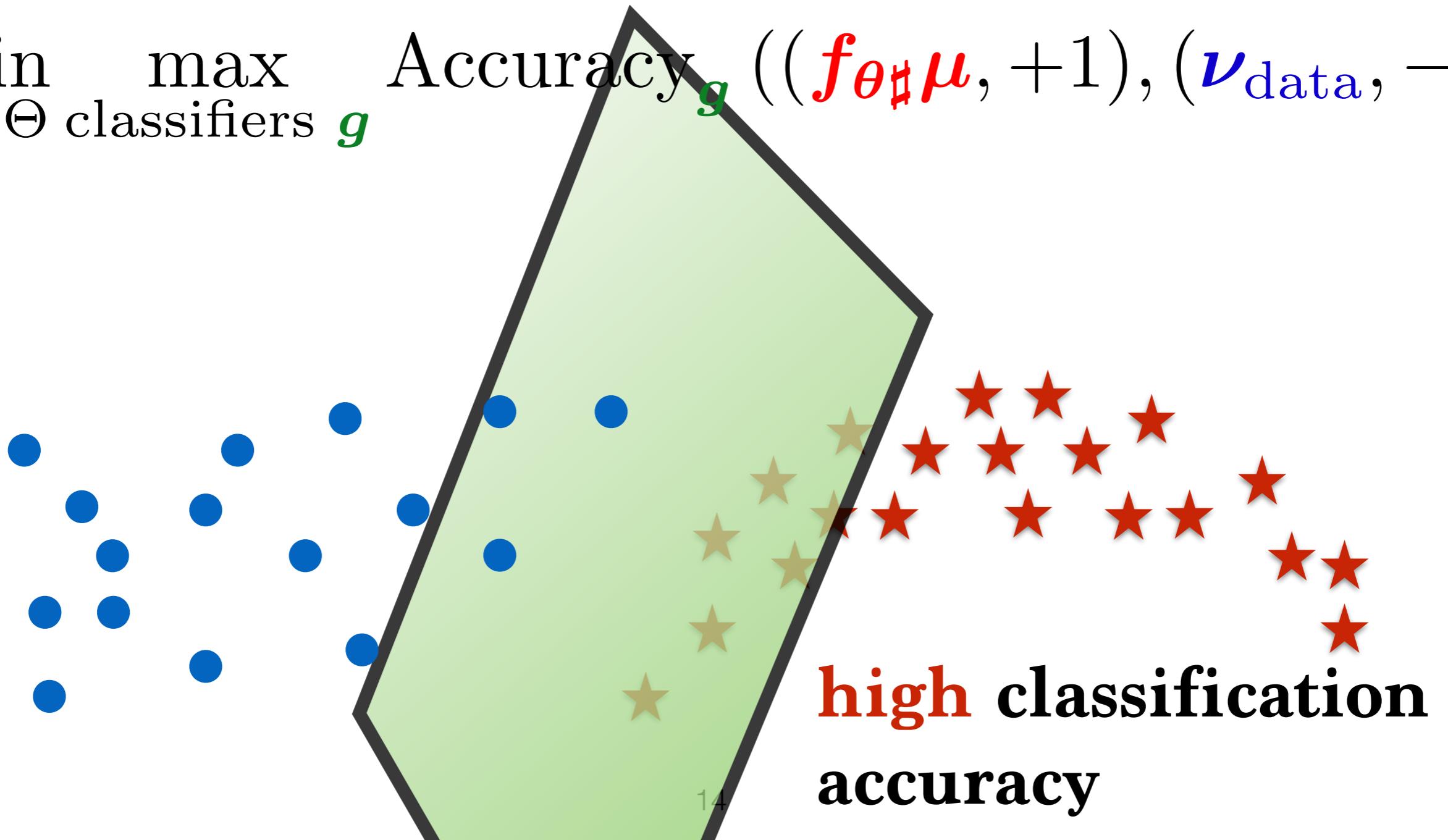
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

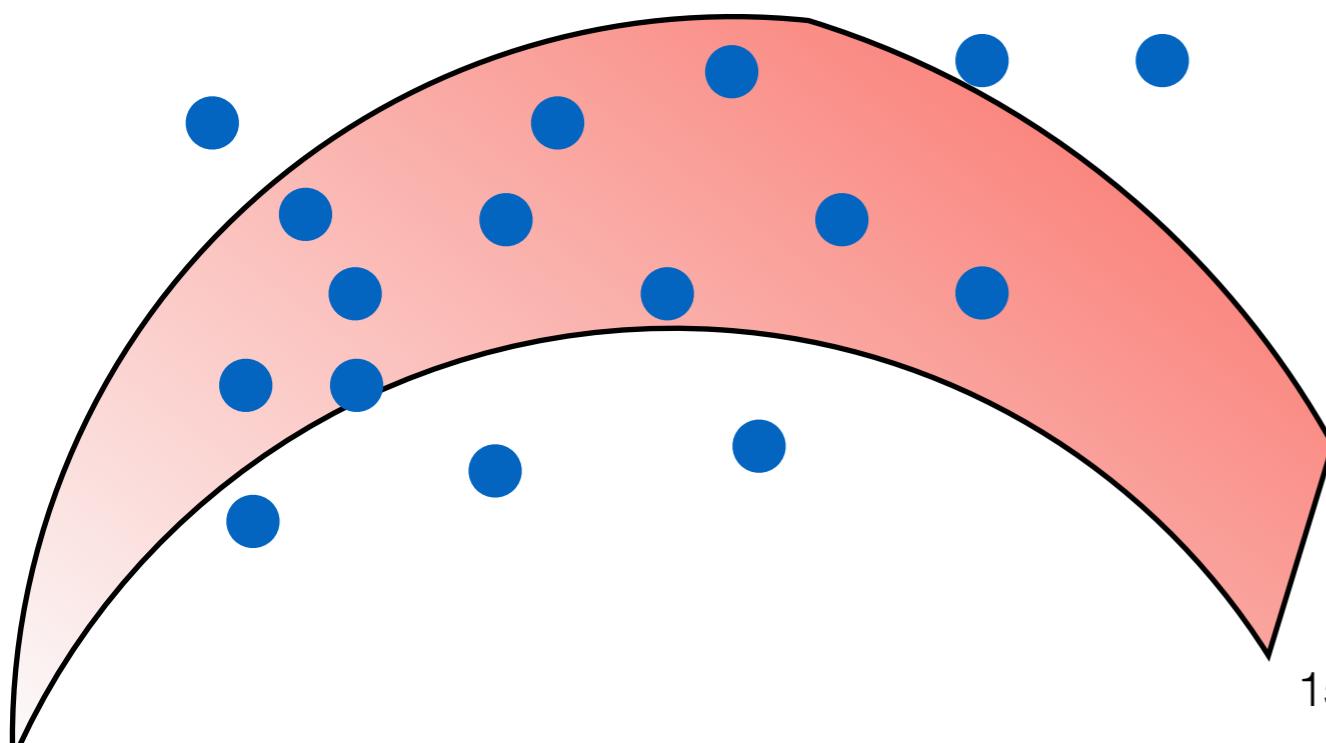
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1))$$



Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

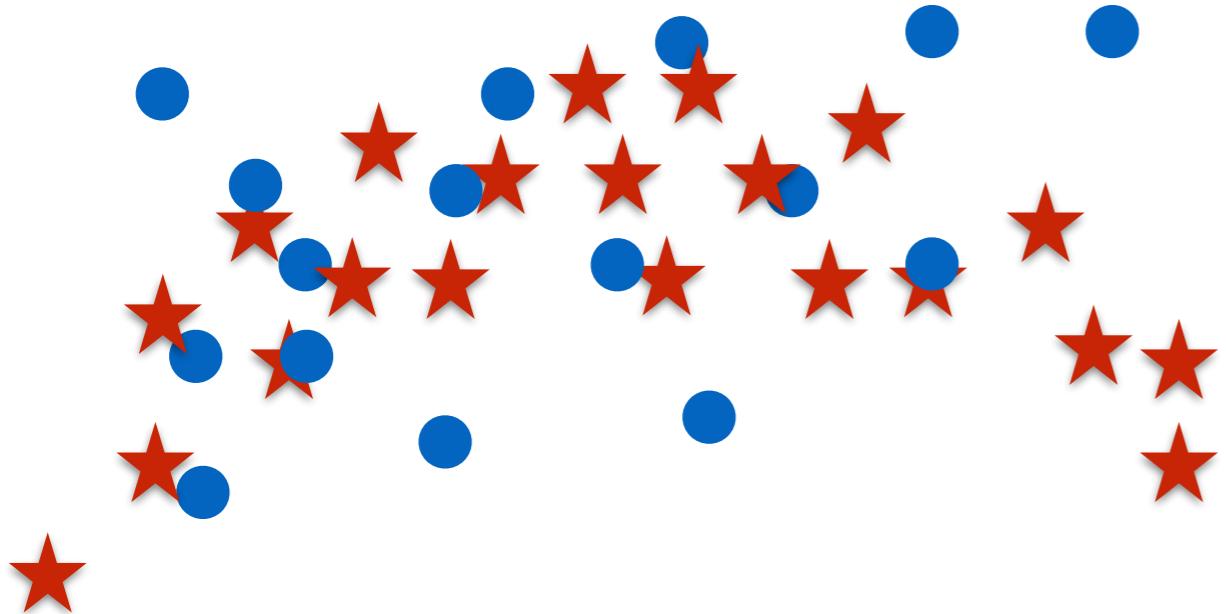
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



Workarounds?

- Formulation as **adversarial problem** [GPM...'14]

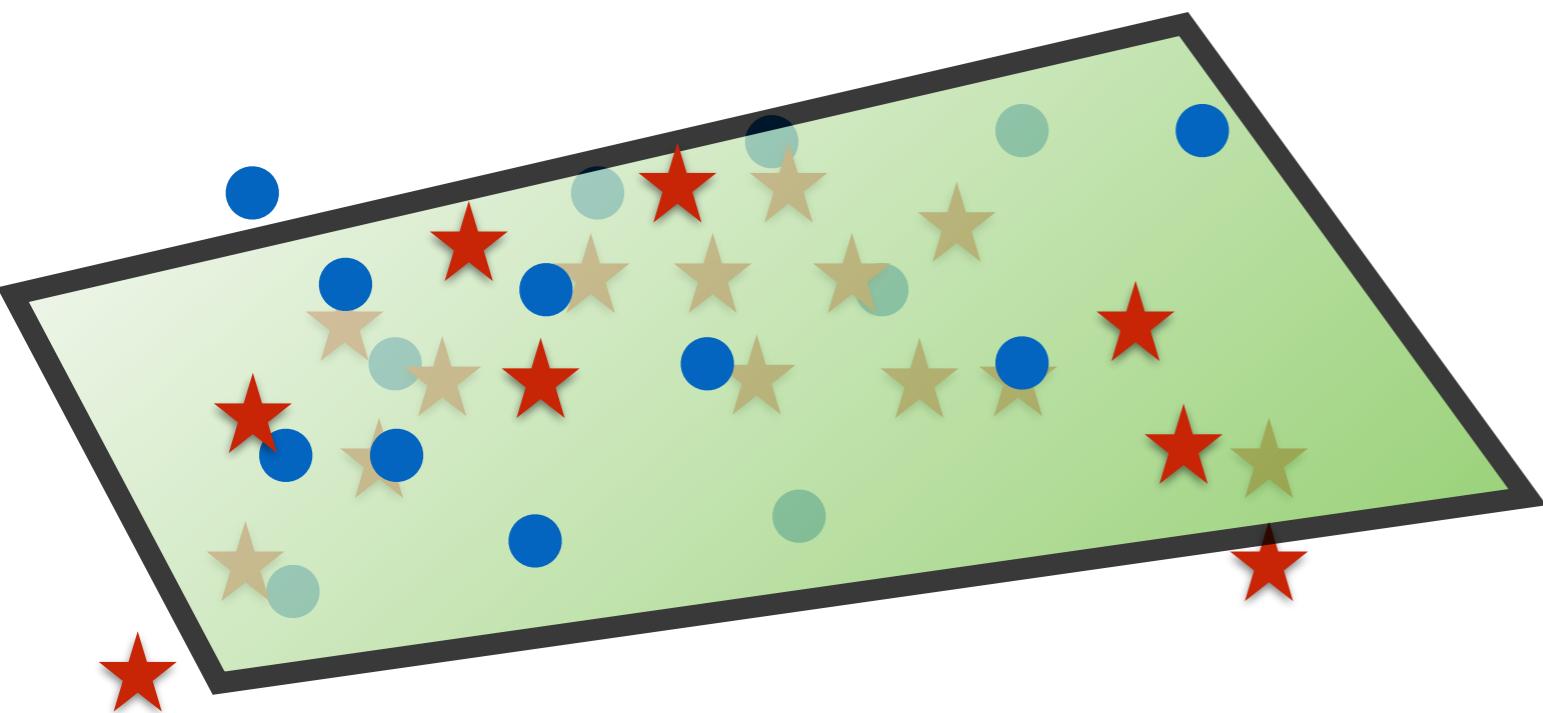
$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



Workarounds?

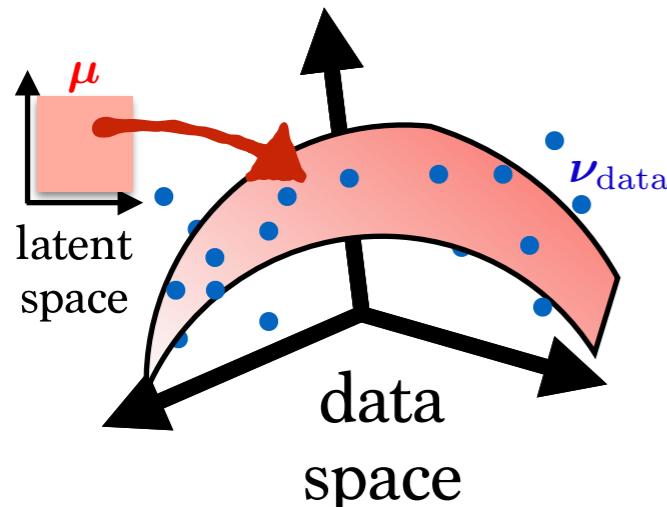
- Formulation as **adversarial problem** [GPM...'14]

$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left((f_{\theta \sharp} \mu, +1), (\nu_{\text{data}}, -1) \right)$$



**low classification accuracy...
is the goal.**

Another idea?



- Use a **metric Δ** for probability measures, that can handle measures with non-overlapping supports:

$$\min_{\theta \in \Theta} \Delta(\nu_{\text{data}}, p_{\theta}), \quad \text{not } \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

Minimum Δ Estimation

The Annals of Statistics
1980, Vol. 8, No. 3, 457–487

MINIMUM **1 CHI-SQUARE, NOT MAXIMUM LIKELIHOOD!**

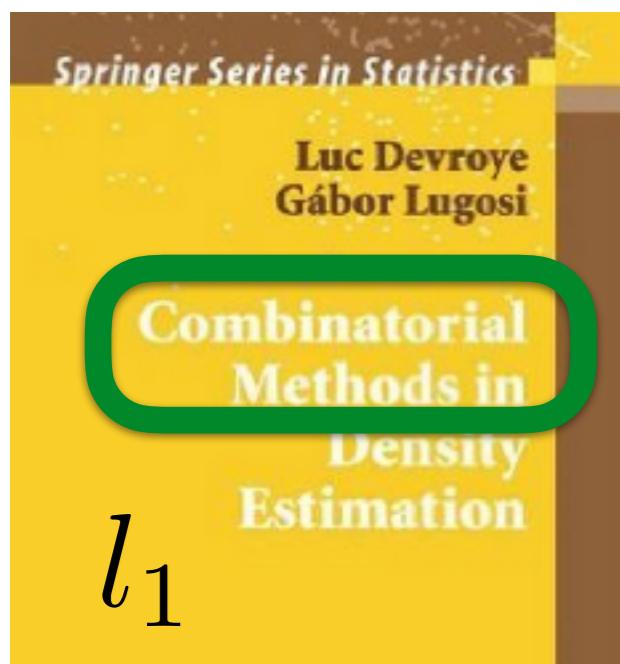
BY JOSEPH BERKSON

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

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,*}

△ Generative Model Estimation

Generative Moment Matching Networks

Yujia Li¹

Kevin Swersky¹

Richard Zemel^{1,2}

¹Department of Computer Science, University of Toronto, Toronto, ON, CANADA

²Canadian Institute for Advanced Research, Toronto, ON, CANADA

YUJIALI@CS.TORONTO.EDU

KSWERSKY@CS.TORONTO.EDU

ZEMEL@CS.TORONTO.EDU

Training generative neural networks via Maximum Mean Discrepancy optimization

Gintare Karolina Dziugaite
University of Cambridge

Daniel M. Roy
University of Toronto

Zoubin Ghahramani
University of Cambridge

MMD GAN: Towards Deeper Understanding of Moment Matching Network

Chun-Liang Li^{1,*} Wei-Cheng Chang^{1,*} Yu Cheng² Yiming Yang¹ Barnabás Póczos¹

¹ Carnegie Mellon University, ²IBM Research

{chunli, wchang2, yiming, bapoczos}@cs.cmu.edu chengyu@us.ibm.com

△ Generative Model Estimation

Generative Moment Matching Networks

Yujia Li¹
Kevin Swersky¹
Richard Zemel^{1,2}

YUJIALI@CS.TORONTO.EDU
KSWERSKY@CS.TORONTO.EDU
ZEMEL@CS.TORONTO.EDU

¹Department of Computer Science, University of Toronto, Toronto, ON, CANADA

²Canadian Institute for Advanced Research, Toronto, ON, CANADA

MMD GAN: Towards Deeper Understanding of Moment Matching Network

Chun-Liang Li^{1,*} Wei-Cheng Chang^{1,*} Yu Cheng² Yiming Yang¹ Barnabás Póczos¹
¹ Carnegie Mellon University, ²IBM Research
{chunli, wchang2, yiming, bapoczos}@cs.cmu.edu chengyu@us.ibm.com

Training generative neural networks via Maximum Mean Discrepancy optimization

Gintare Karolina Dziugaite
University of Cambridge

Daniel M. Roy
University of Toronto

Zoubin Ghahramani
University of Cambridge

Wasserstein Training of Restricted Boltzmann Machines

Grégoire Montavon
Technische Universität Berlin
gregoire.montavon@tu-berlin.de

Klaus-Robert Müller*
Technische Universität Berlin
klaus-robert.mueller@tu-berlin.de

Inference in generative models using the Wasserstein distance

Espen Bernton, Mathieu Gerber, Pierre E. Jacob, Christian P. Robert

Wasserstein GAN

Martin Arjovsky¹, Soumith Chintala², and Léon Bottou^{1,2}

¹Courant Institute of Mathematical Sciences

²Facebook AI Research

△ Generative Model Estimation

Generative Moment Matching Networks

Yujia Li¹
Kevin Swersky¹
Richard Zemel^{1,2}

YUJIALI@CS.TORONTO.EDU
KSWERSKY@CS.TORONTO.EDU
ZEMEL@CS.TORONTO.EDU

¹Department of Computer Science, University of Toronto, Toronto, ON, CANADA

²Canadian Institute for Advanced Research, Toronto, ON, CANADA

Training generative neural networks via Maximum Mean Discrepancy optimization

MMD GAN: Towards Deeper Understanding of Moment Matching Network

Chun-Liang Li^{1,*} Wei-Cheng Chang^{1,*} Yu Cheng² Yiming Yang¹ Barnabás Póczos¹
¹ Carnegie Mellon University, ²IBM Research
{chunli, wchang2, yiming, bapoczos}@cs.cmu.edu chengyu@us.ibm.com

Gintare Karolina Dziugaite
University of Cambridge

Daniel M. Roy
University of Toronto

Zoubin Ghahramani
University of Cambridge

Wasserstein Training of Restricted Boltzmann Machines

Grégoire Montavon
Technische Universität Berlin
gregoire.montavon@tu-berlin.de

Klaus-Robert Müller^{*}
Technische Universität Berlin
klaus-robert.mueller@tu-berlin.de

Inference in generative models using the Wasserstein distance

Espen Bernton, Mathieu Gerber, Pierre E. Jacob, Christian P. Robert

Wasserstein GAN

Martin Arjovsky¹, Soumith Chintala², and Léon Bottou^{1,2}

¹Courant Institute of Mathematical Sciences

²Facebook AI Research

Learning Generative Models with Sinkhorn Divergences

Improving GANs Using Optimal Transport

Aude Genevay
CEREMADE,
Université Paris-Dauphine

Gabriel Peyré
CNRS and DMA,
École Normale Supérieure

Marco Cuturi
ENSAE CREST
Université Paris-Saclay

Tim Salimans^{*}
OpenAI
tim@openai.com

Han Zhang[†]
Rutgers University
han.zhang@cs.rutgers.edu

Alec Radford
OpenAI
alec@openai.com

Dimitris Metaxas
Rutgers University
dnm@cs.rutgers.edu

Minimum Kantorovich Estimation

- Use optimal transport theory, namely *Wasserstein distances* to define discrepancy Δ .

$$\min_{\theta \in \Theta} W(\nu_{\text{data}}, f_{\theta \sharp} \mu)$$

- Optimal transport? fertile field in mathematics.



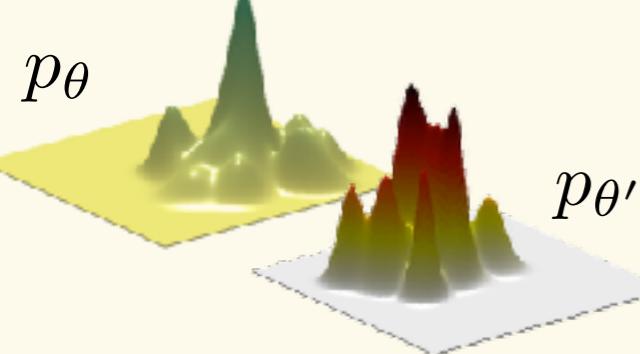
Monge Kantorovich Koopmans Dantzig Brenier Gangbo Otto McCann Villani

Nobel '75

Fields '10

What is Optimal Transport?

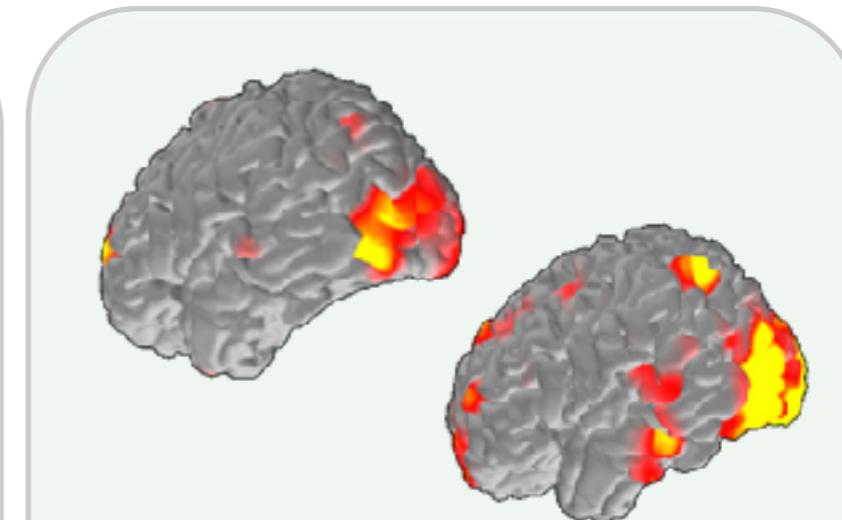
The natural geometry for probability measures



Statistical Models

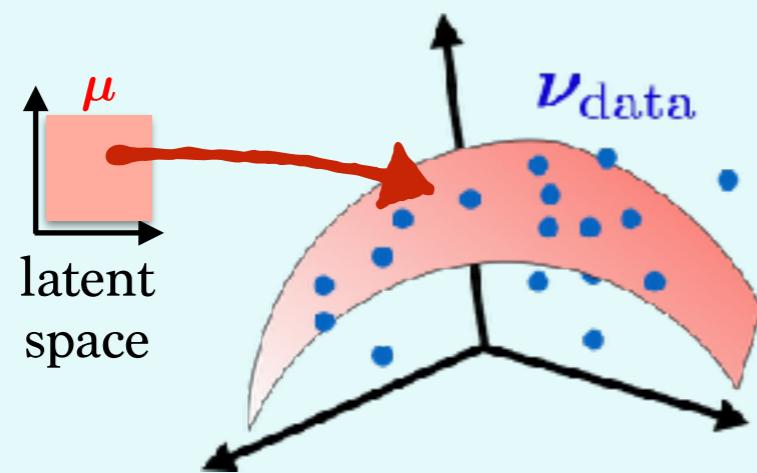


Bags
of features



Brain Activation Maps

Generative
Models
vs. data



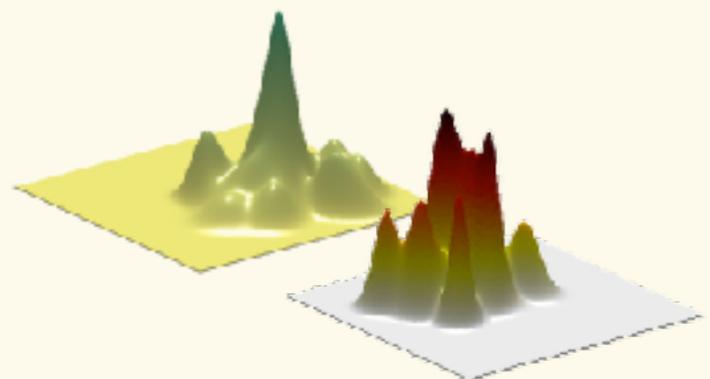
20



Color Histograms

What is Optimal Transport?

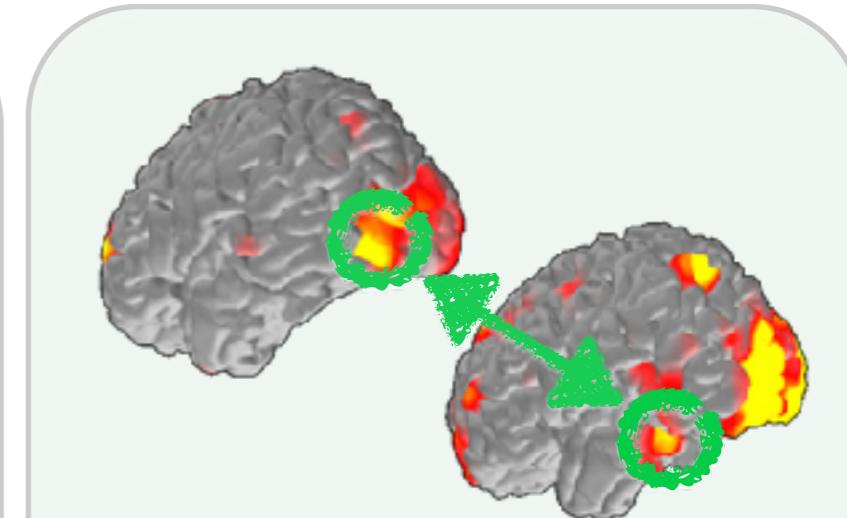
The natural geometry for **probability measures** supported on a metric space.



Statistical Models

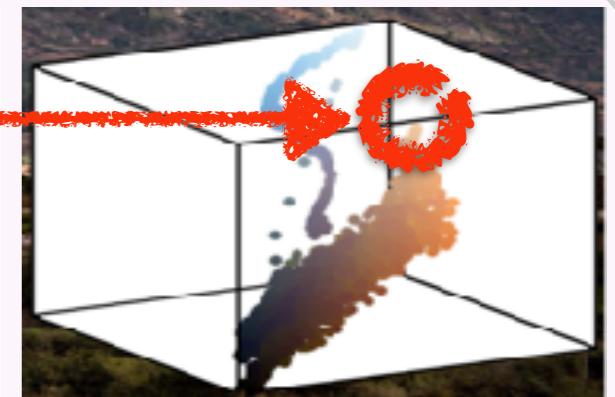
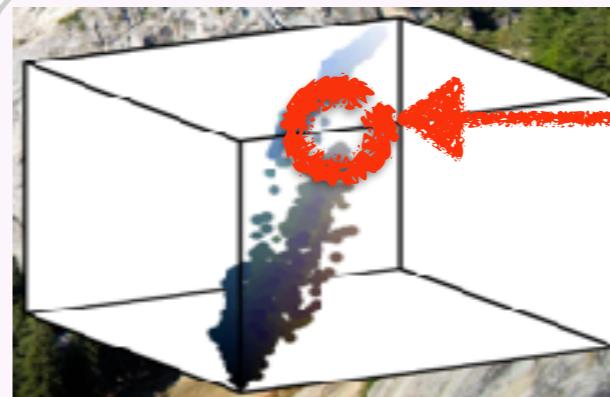
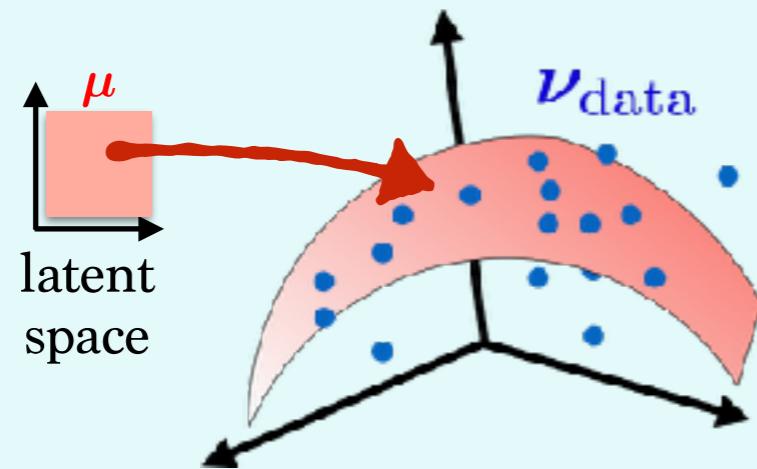


*Bags
of Features*



Brain Activation Maps

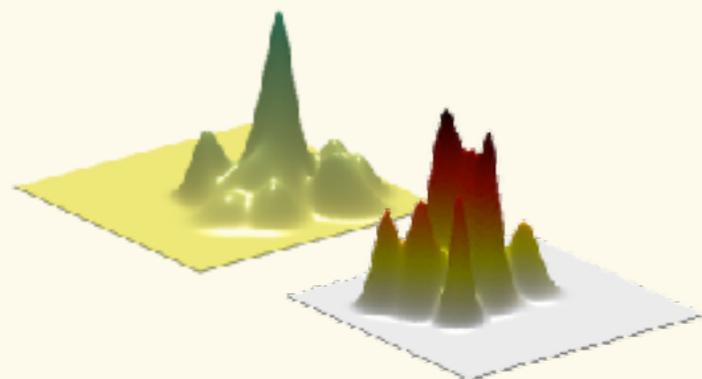
*Generative
Models
vs. Data*



Color Histograms

What is Optimal Transport?

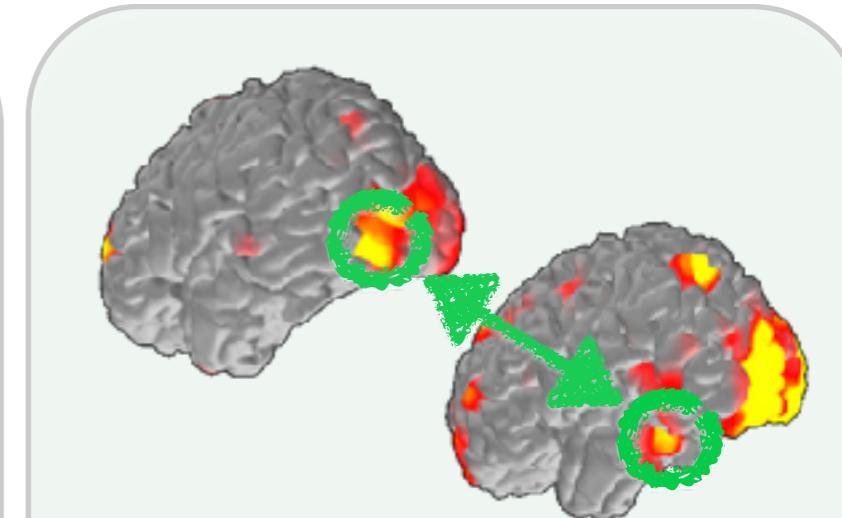
The natural geometry for **probability measures** supported on a metric space.



Statistical Models

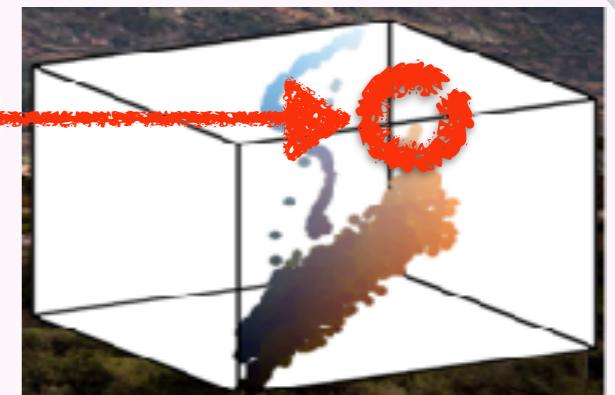
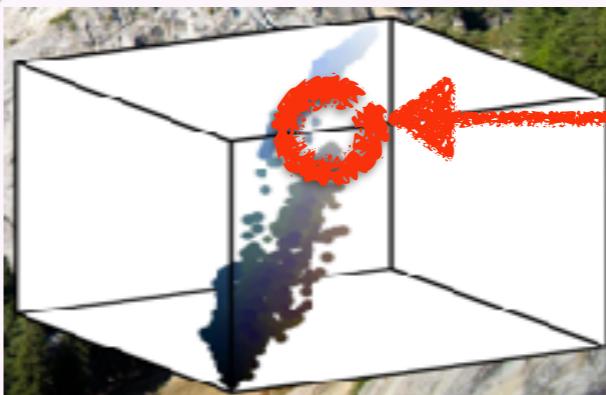
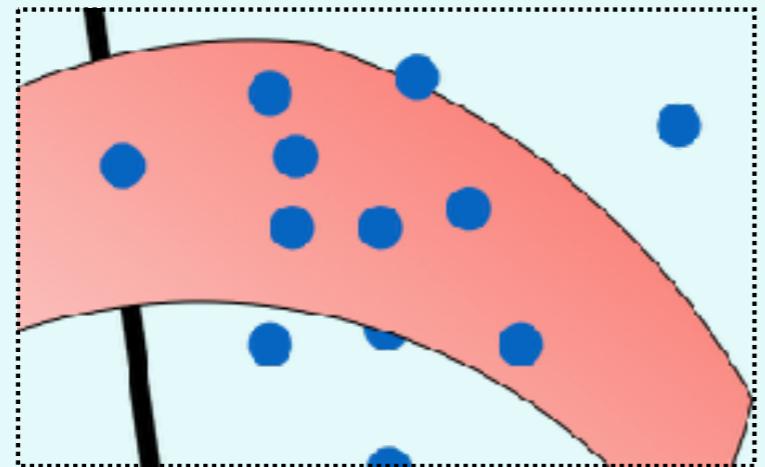


*Bags
of Features*



Brain Activation Maps

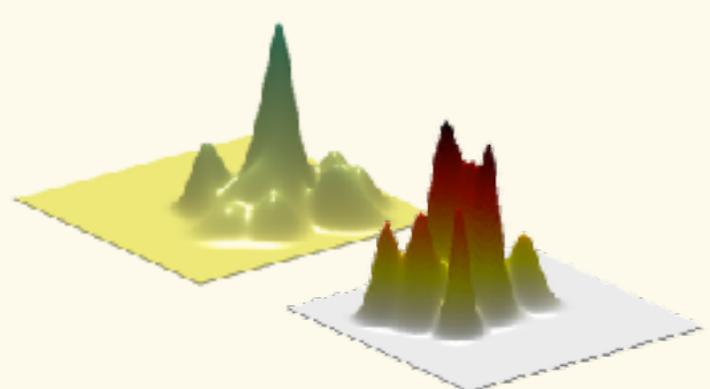
*Generative
Models
vs. Data*



Color Histograms

What is Optimal Transport?

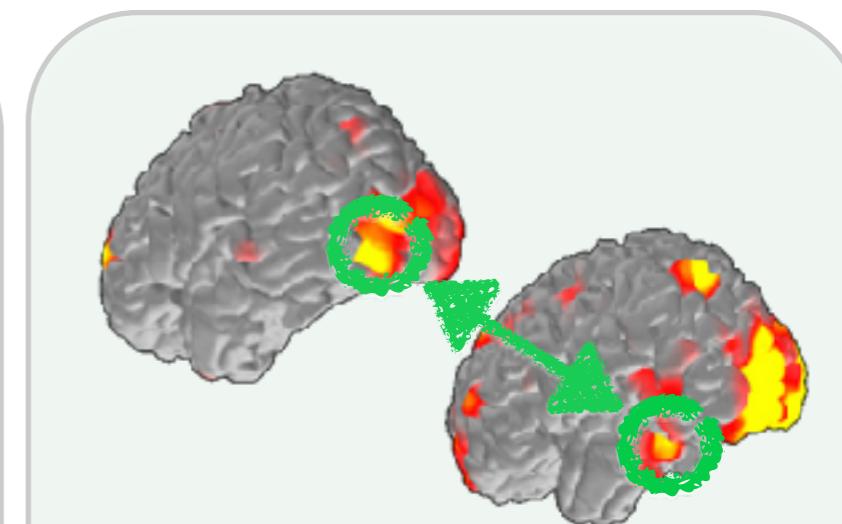
The natural geometry for **probability measures** supported on a metric space.



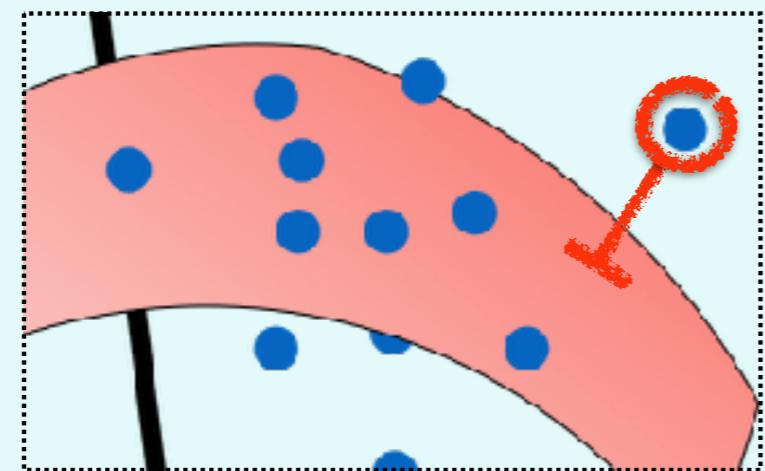
Statistical Models



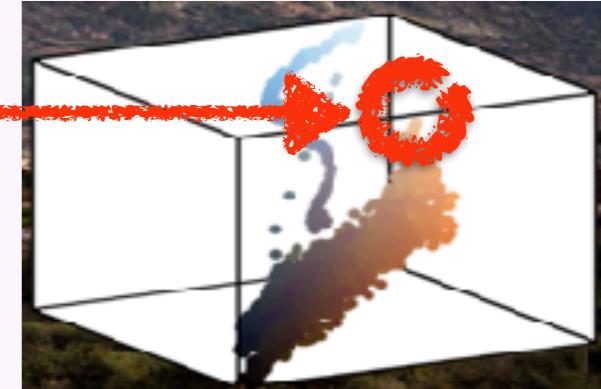
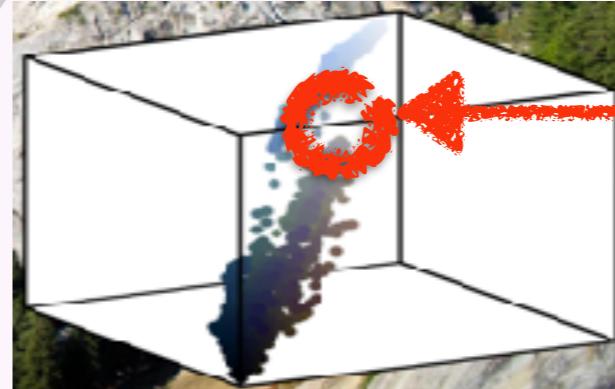
Bags of Features



Brain Activation Maps



Generative Models vs. Data



Color Histograms

Short Course Outline

1. Introduction to optimal transport
2. Optimal transport algorithms
3. Some Applications

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

Introduction to OT

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

Origins: Monge Problem (1781)



60 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

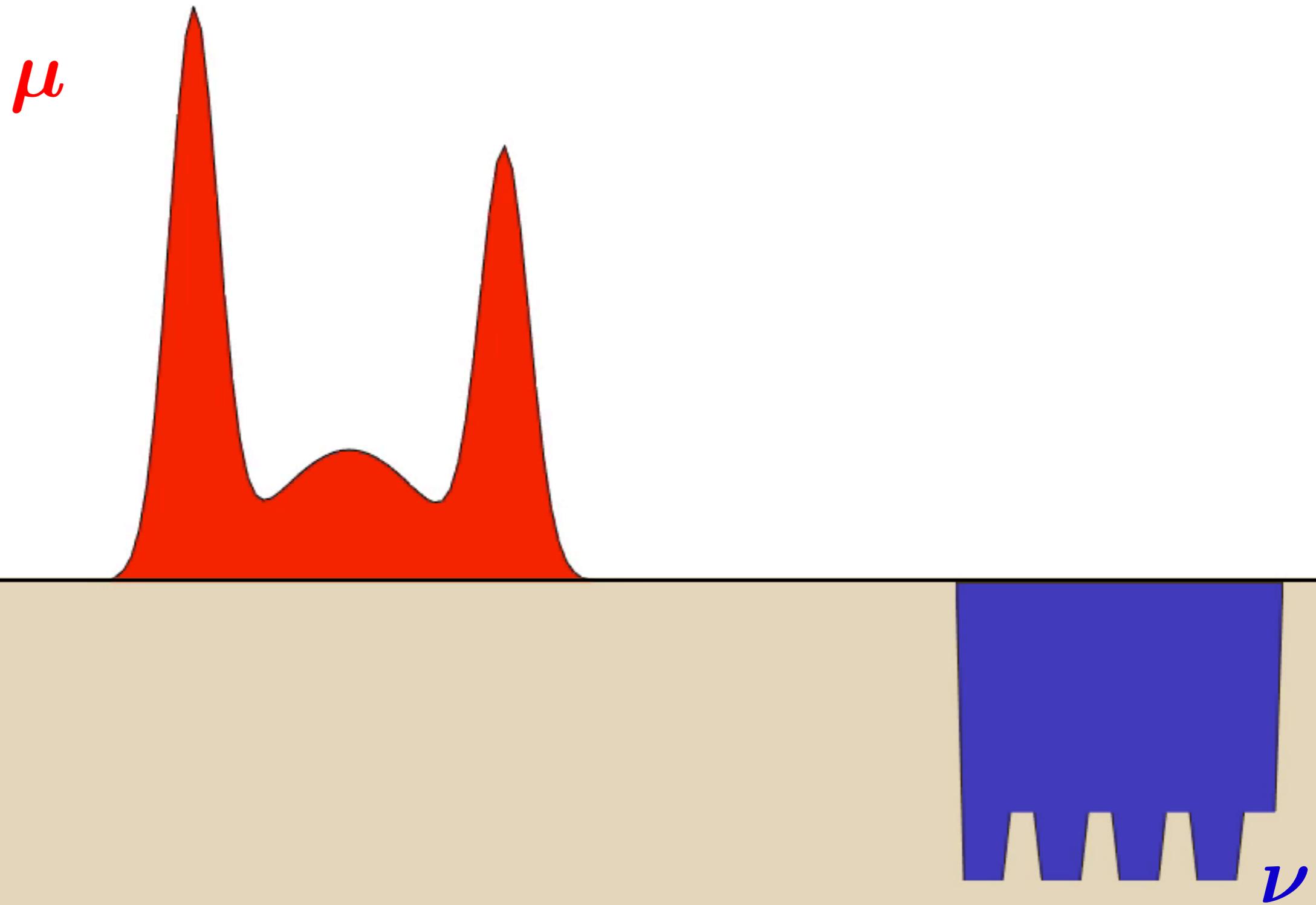
SUR LA

TRANSPORT DES TERRES

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

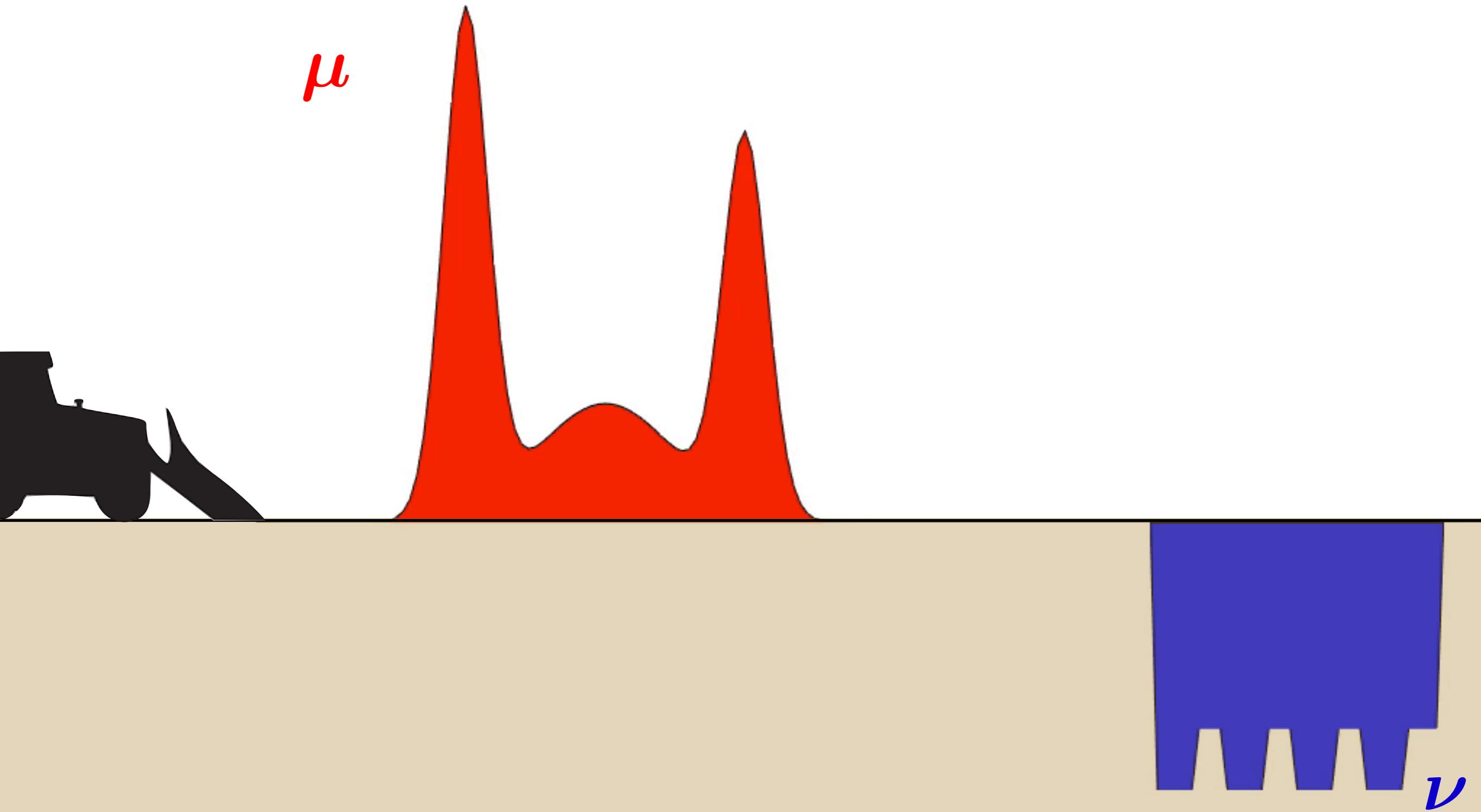
LORSQU'ON doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Origins: Monge Problem



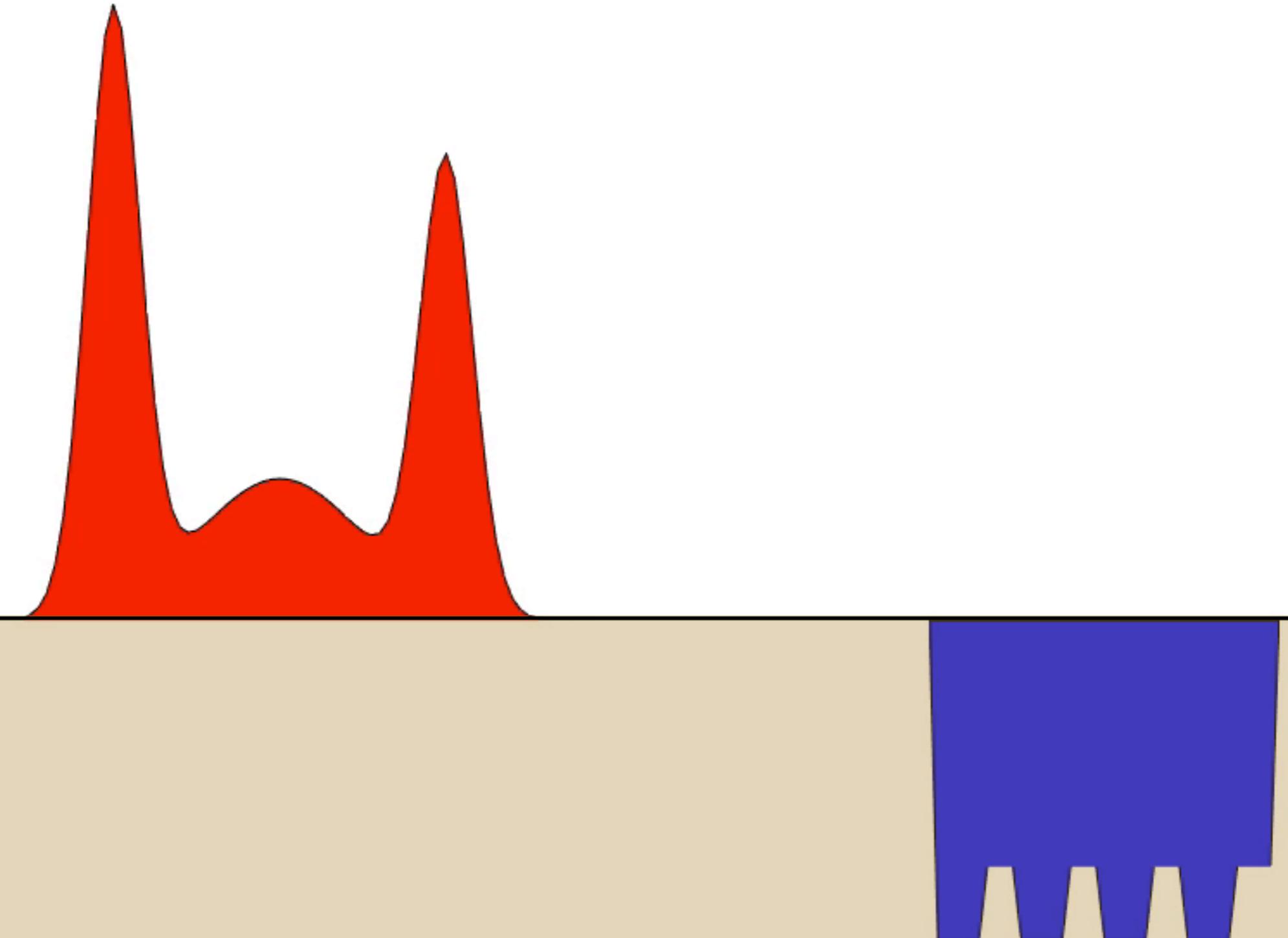
Origins: Monge Problem

In the 21st Century...



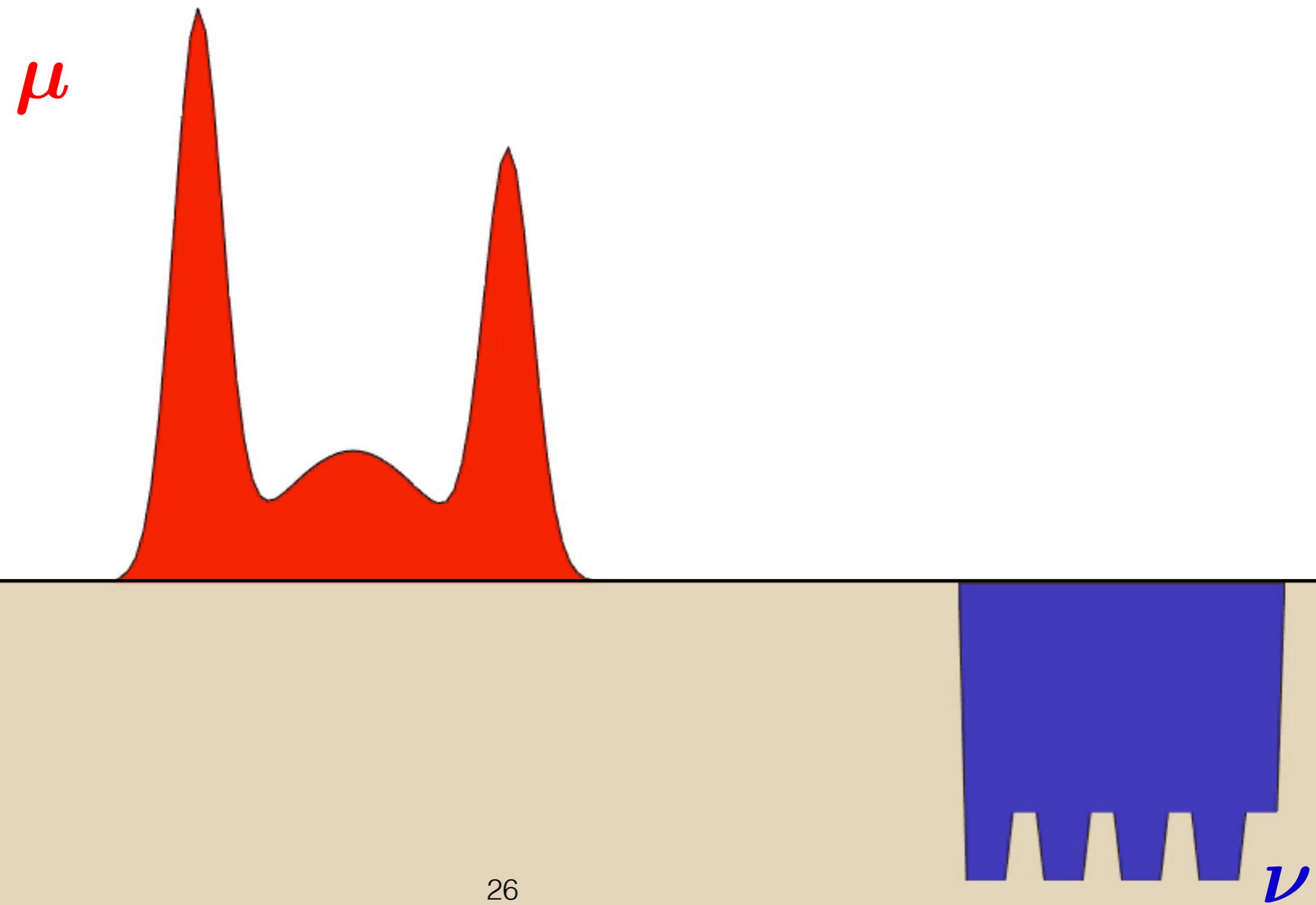
Origins: Monge Problem

In the 21st Century...



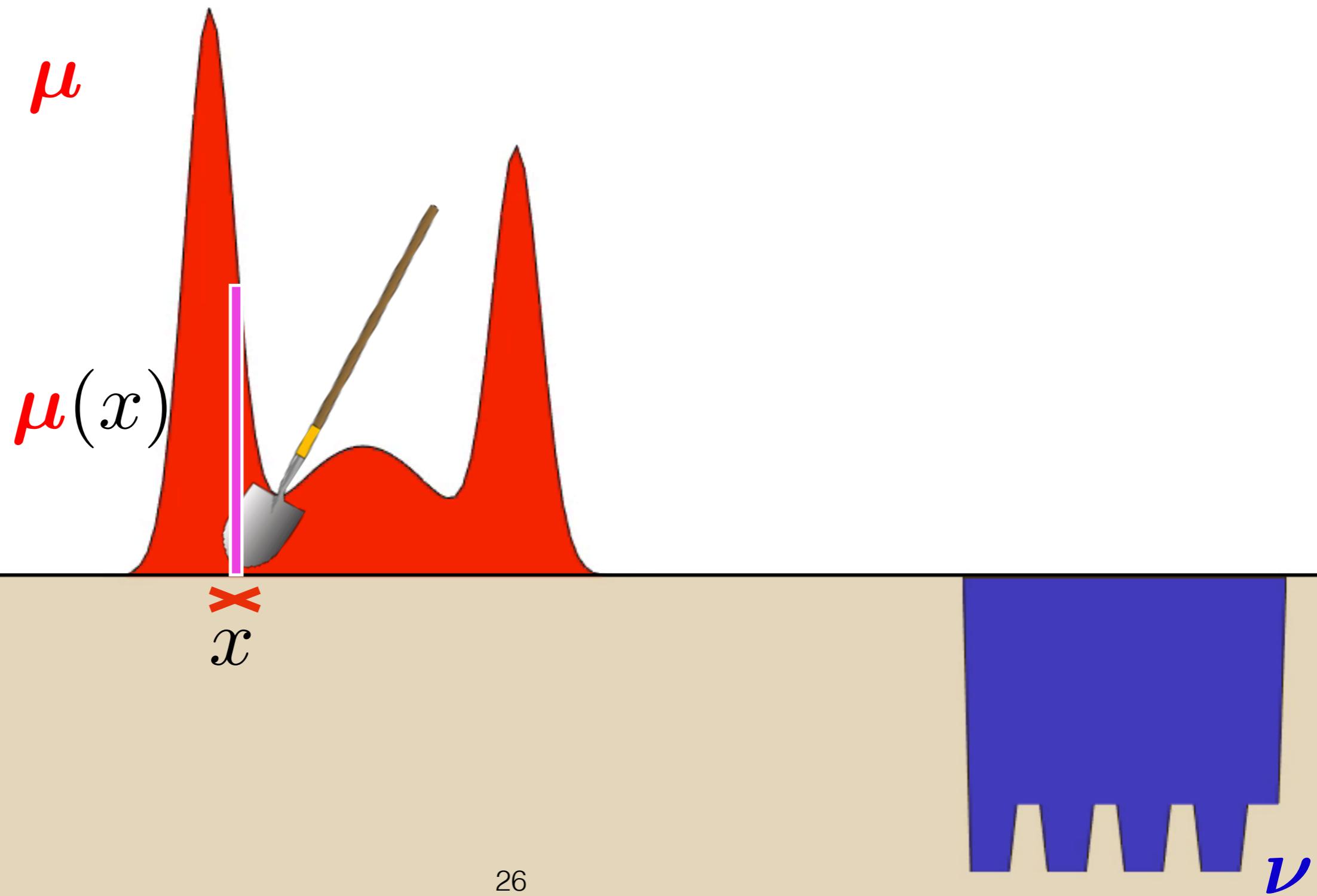
Origins: Monge's Problem

In 1781 however...



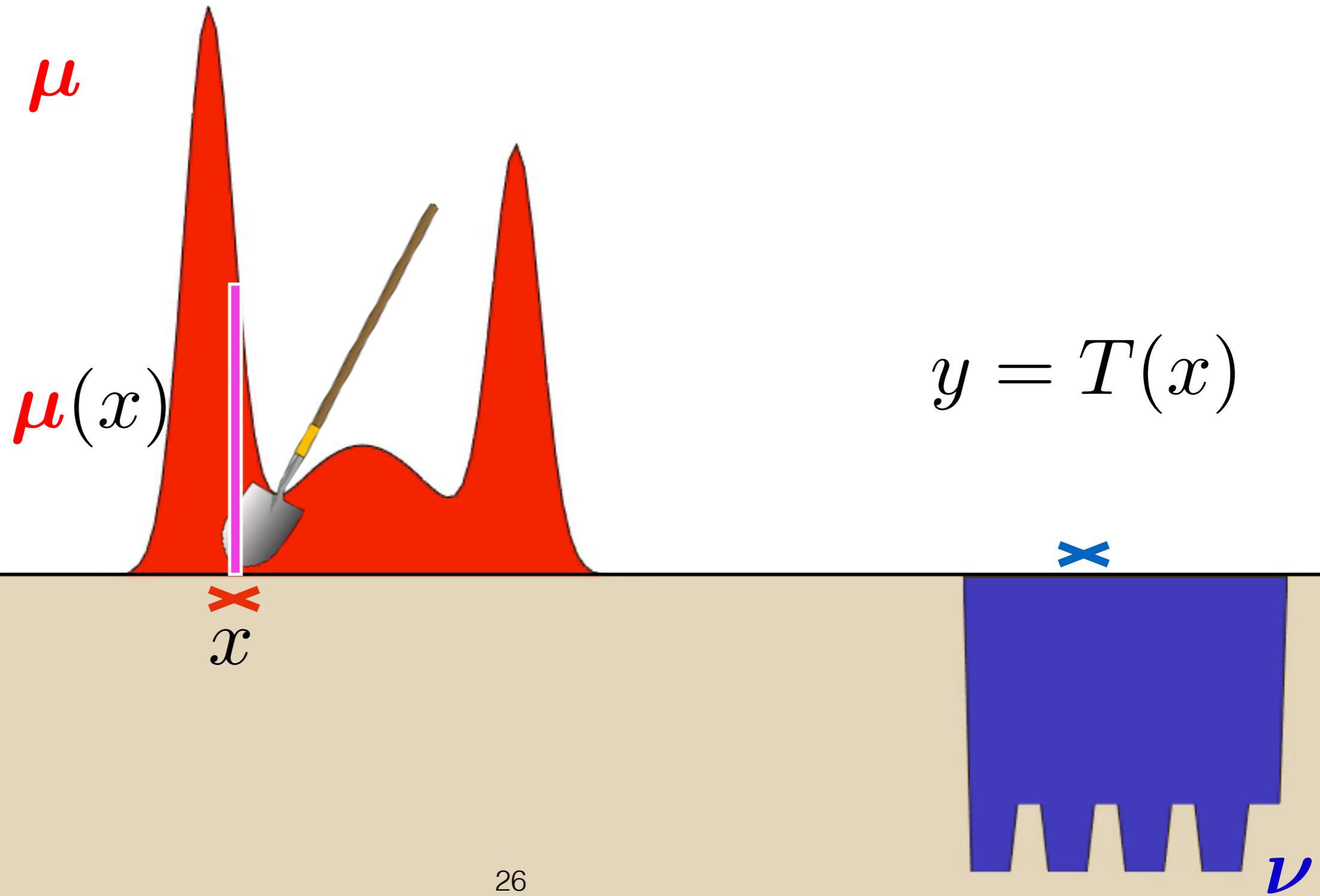
Origins: Monge's Problem

In 1781 however...



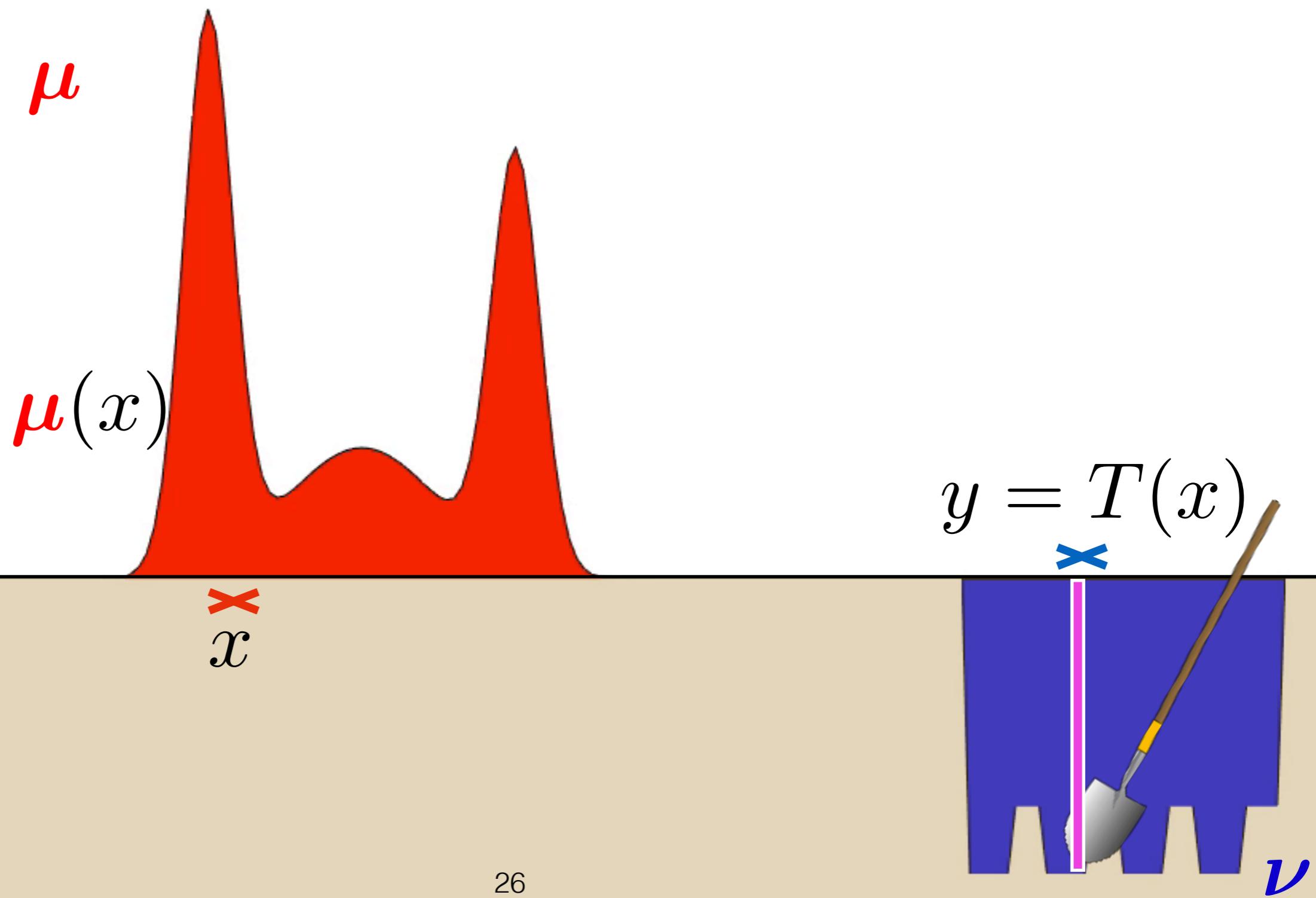
Origins: Monge's Problem

In 1781 however...



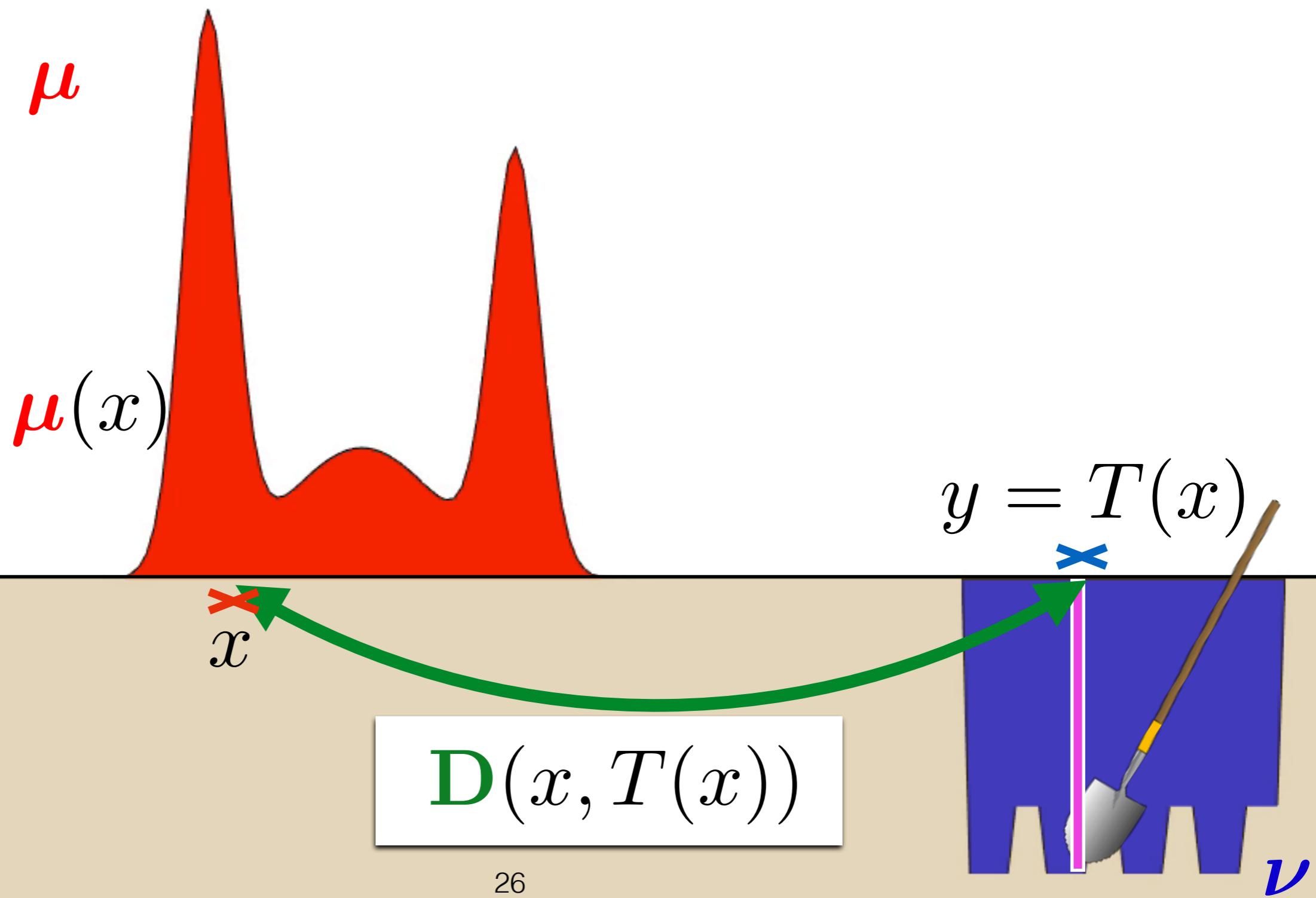
Origins: Monge's Problem

In 1781 however...



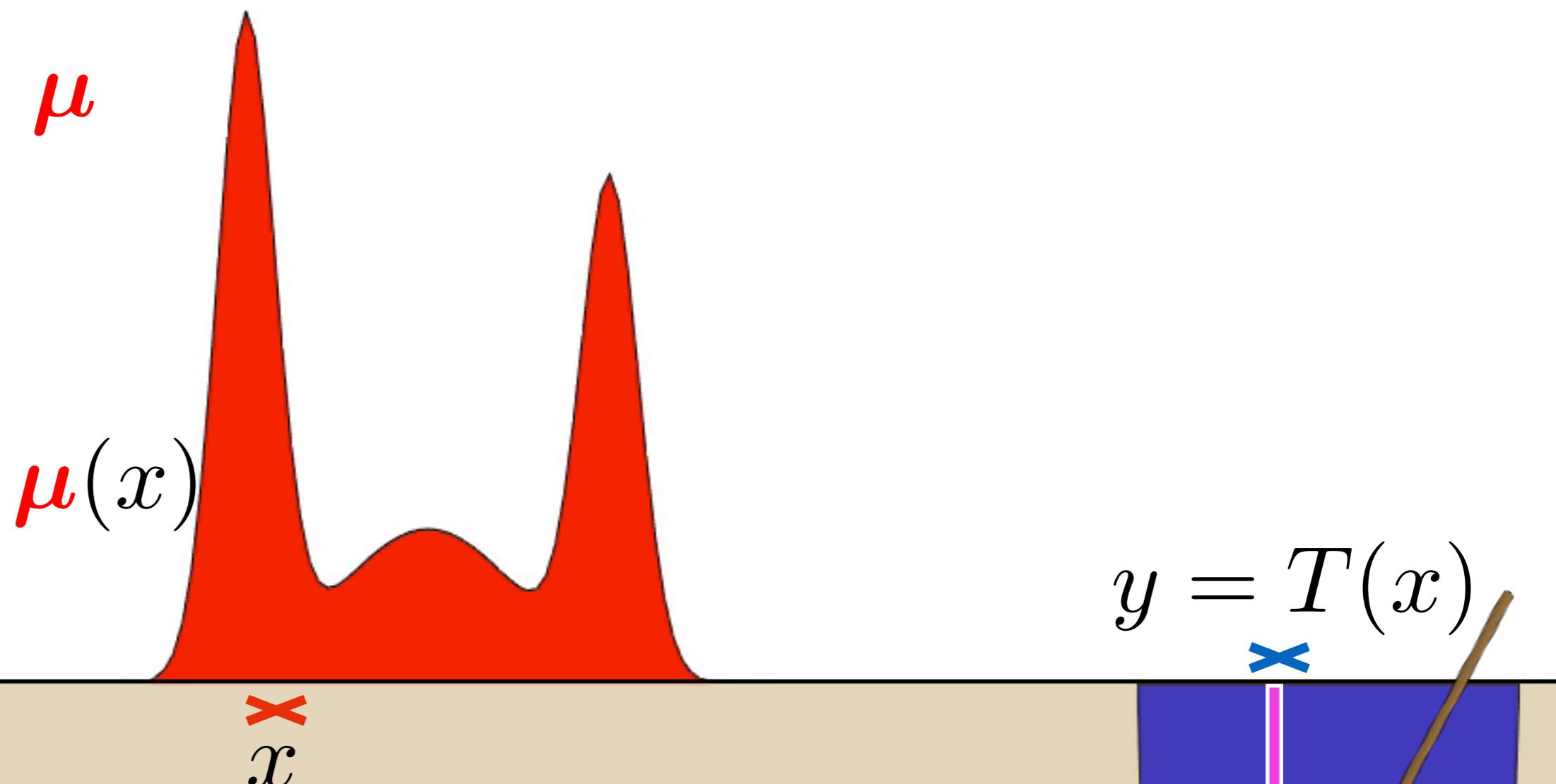
Origins: Monge's Problem

In 1781 however...



Origins: Monge's Problem

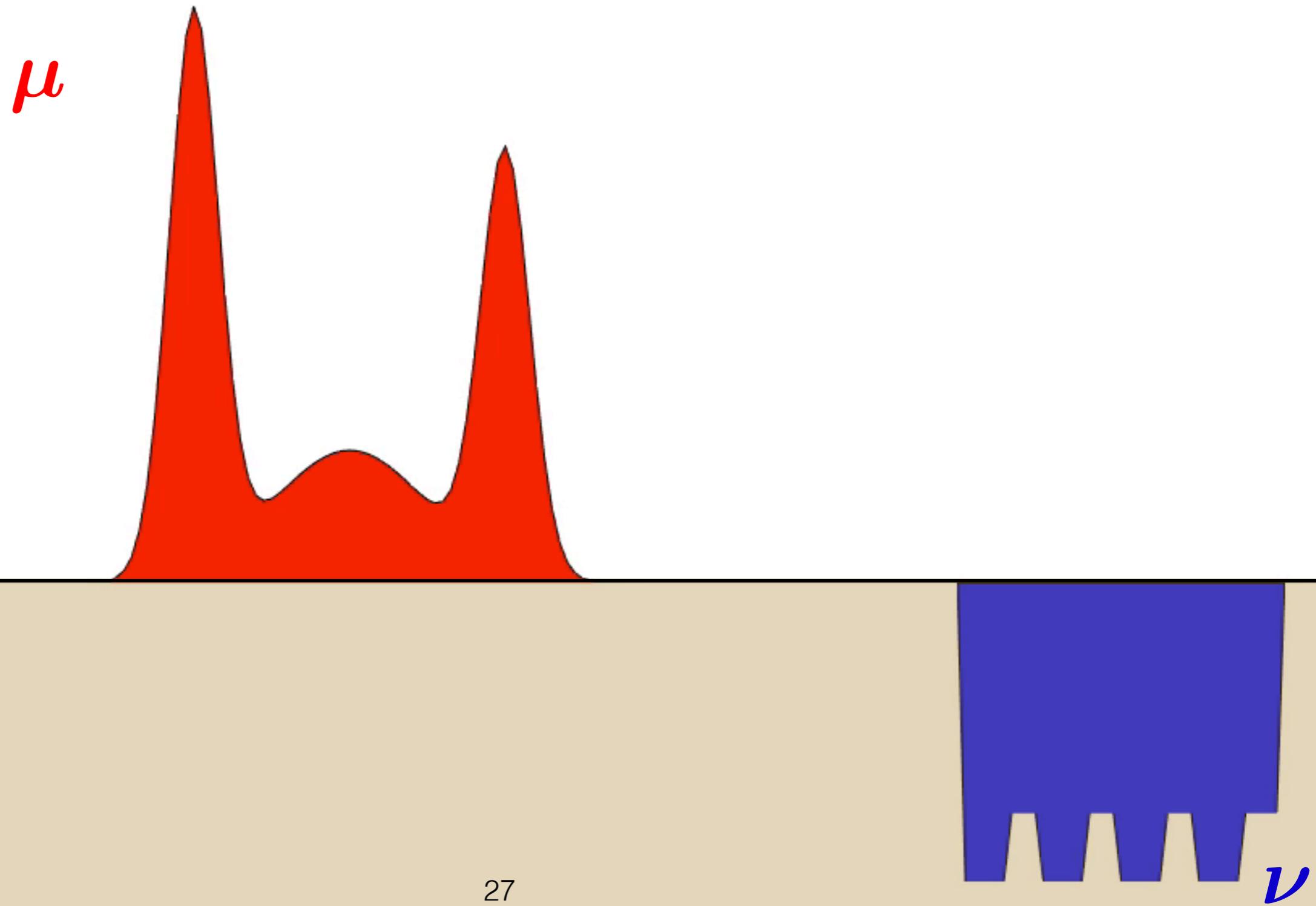
In 1781 however...



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

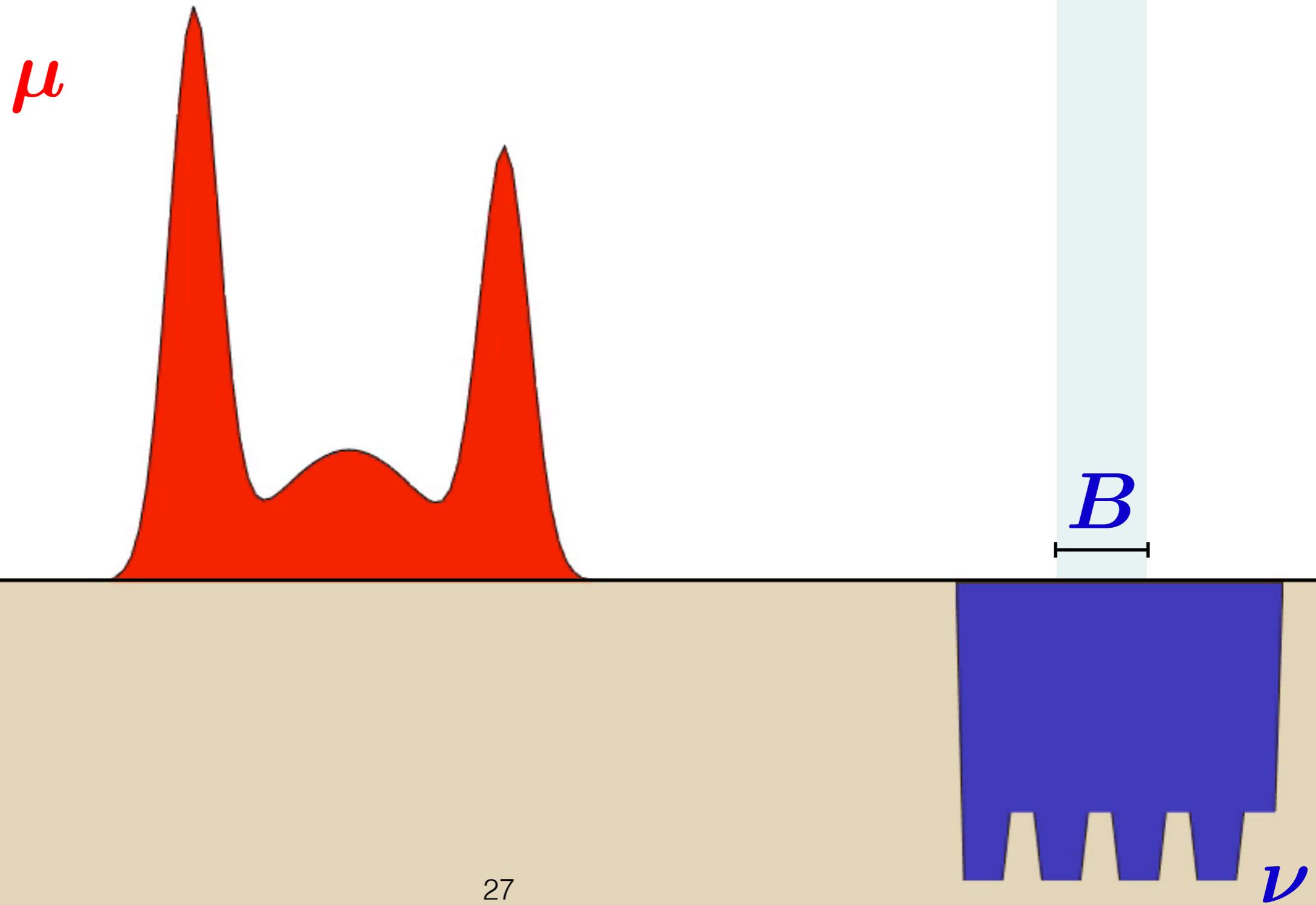
Origins: Monge's Problem

T must map red to blue.



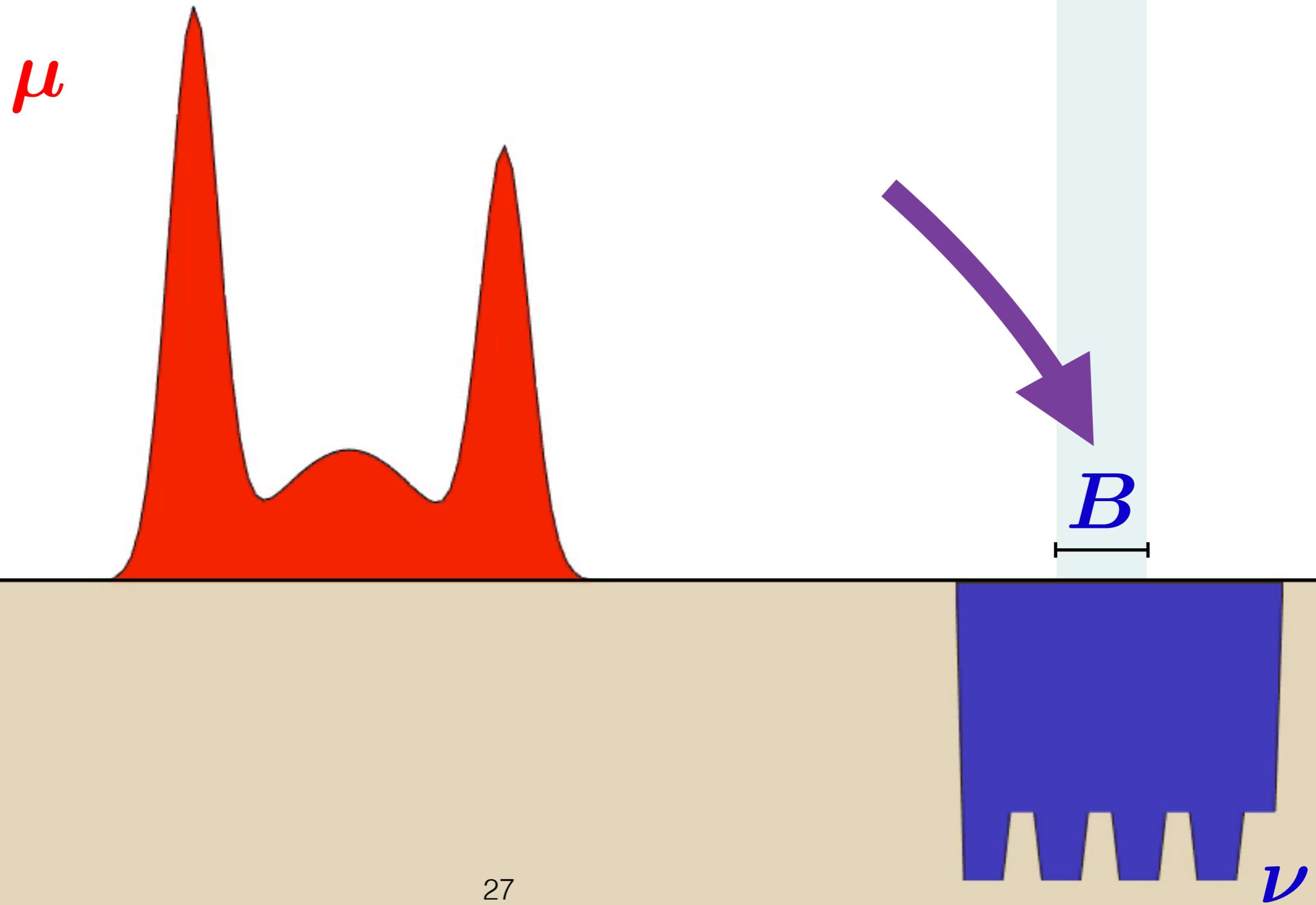
Origins: Monge's Problem

T must map red to blue.



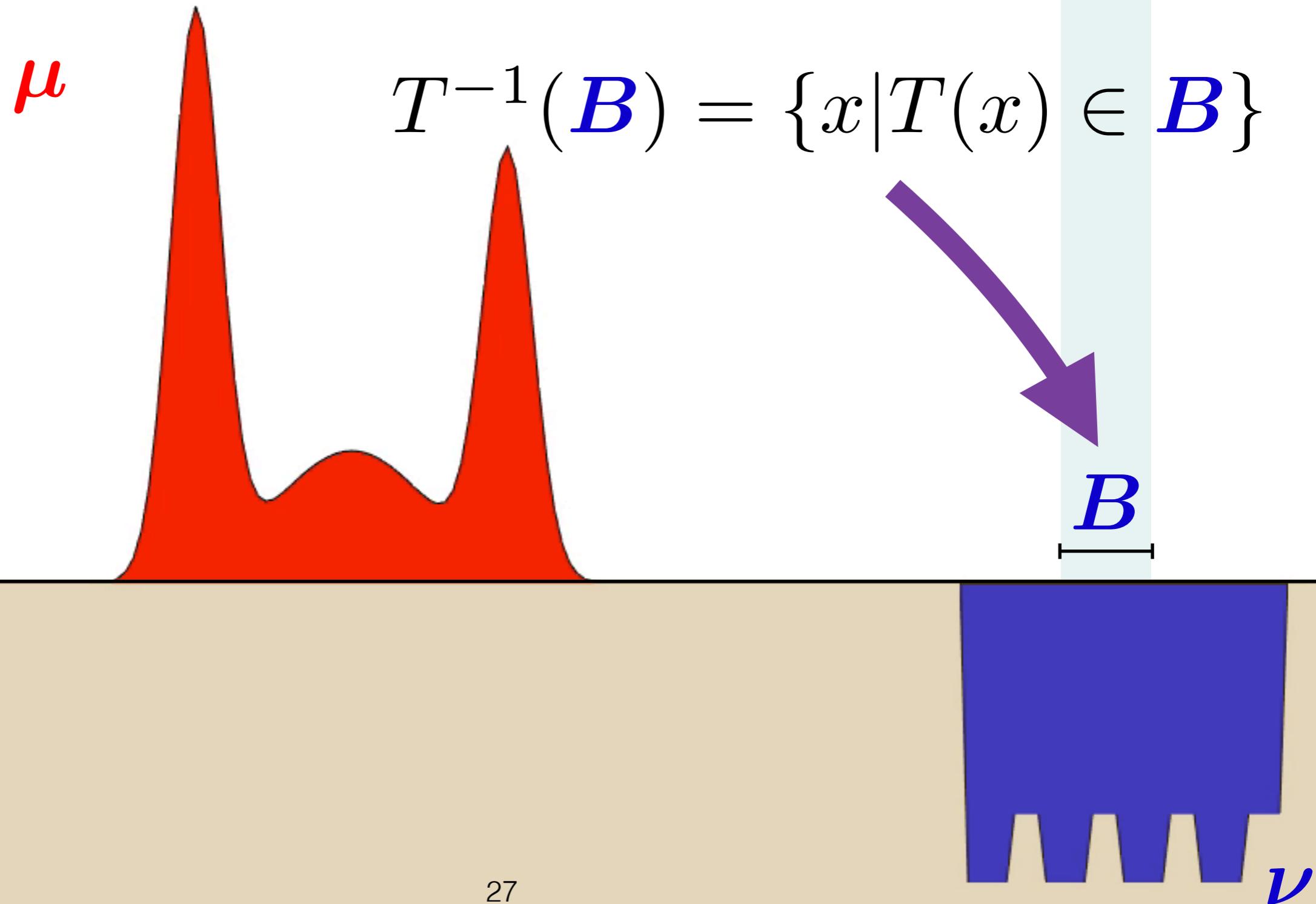
Origins: Monge's Problem

T must map red to blue.



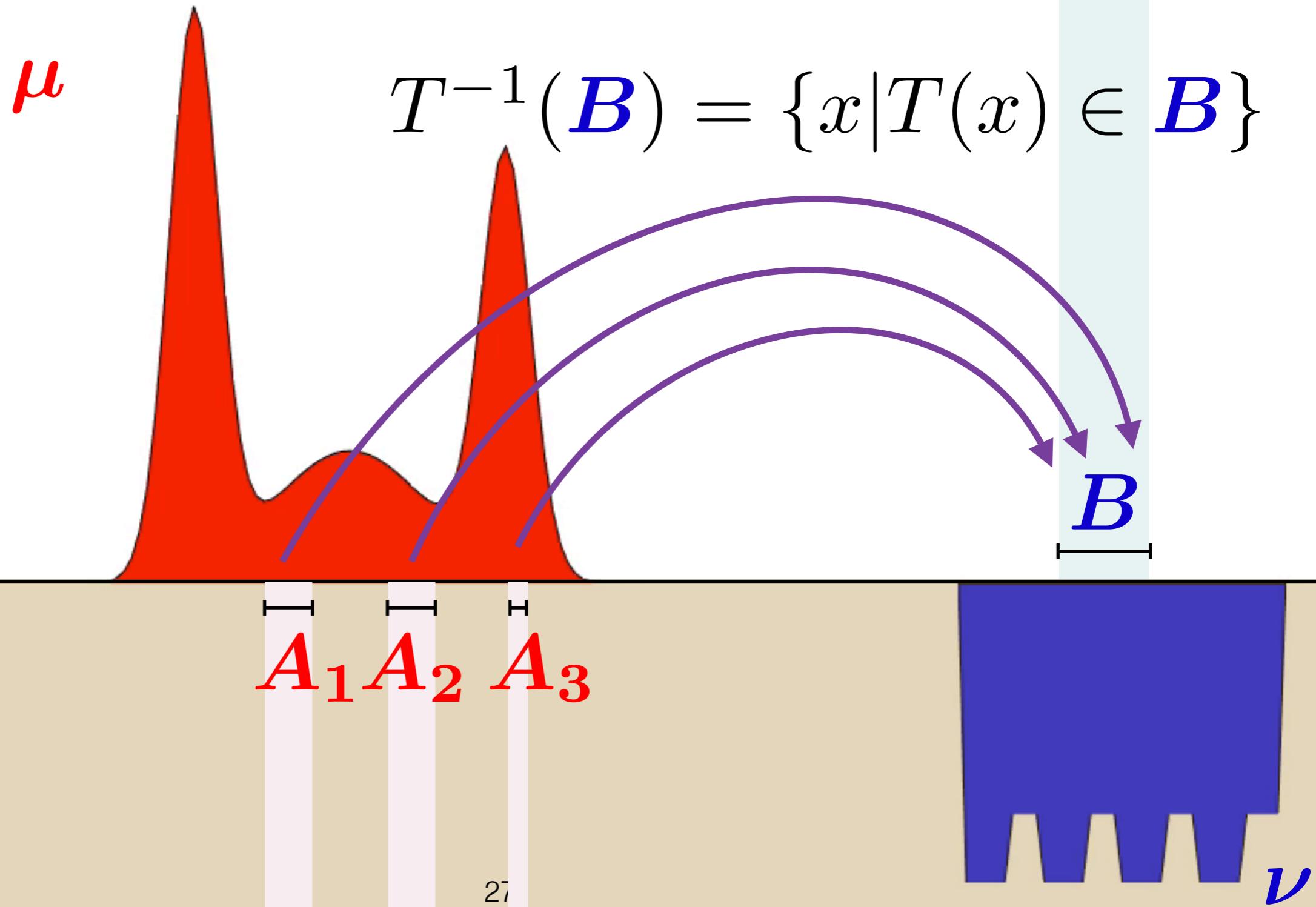
Origins: Monge's Problem

T must map red to blue.



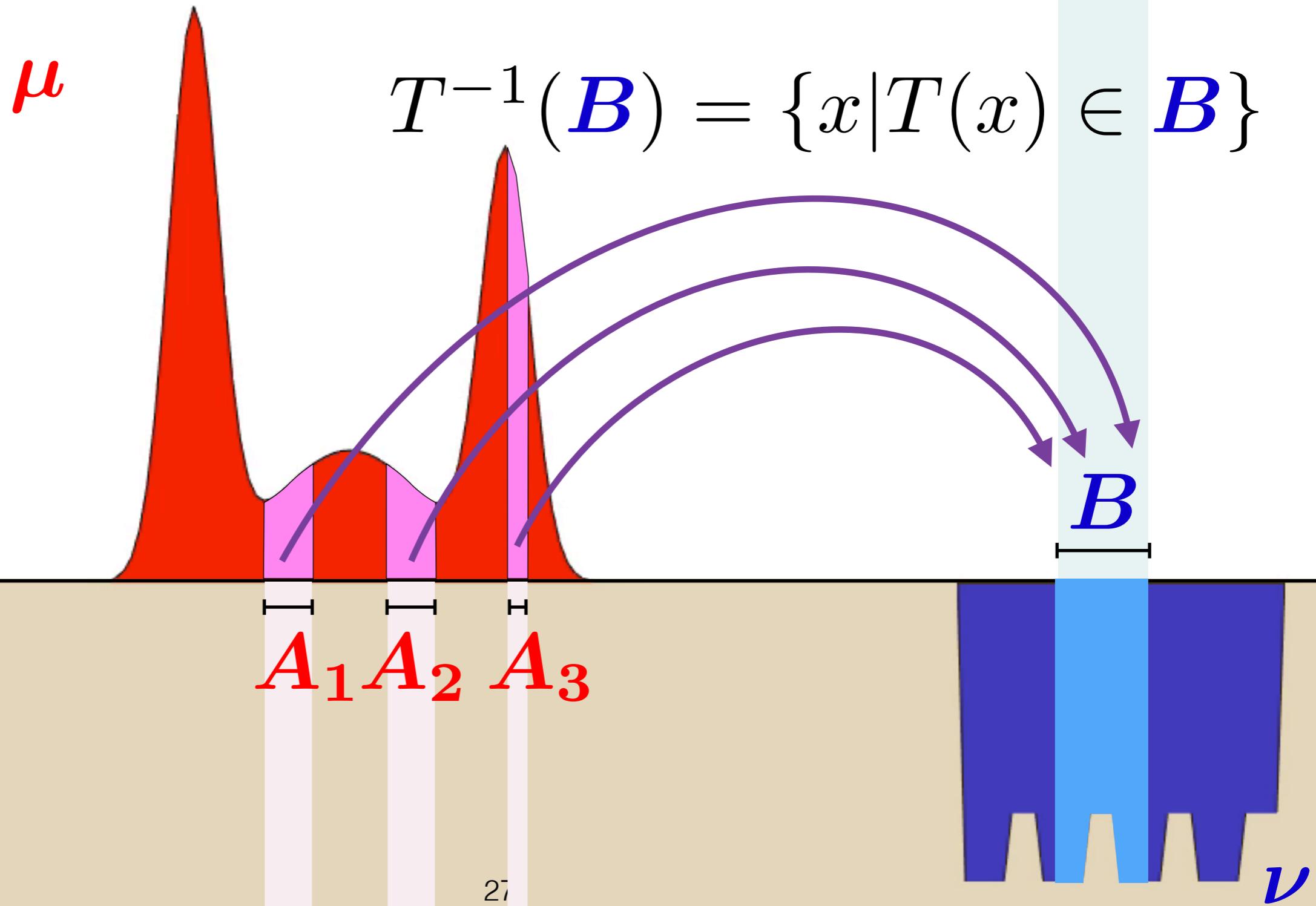
Origins: Monge's Problem

T must map red to blue.



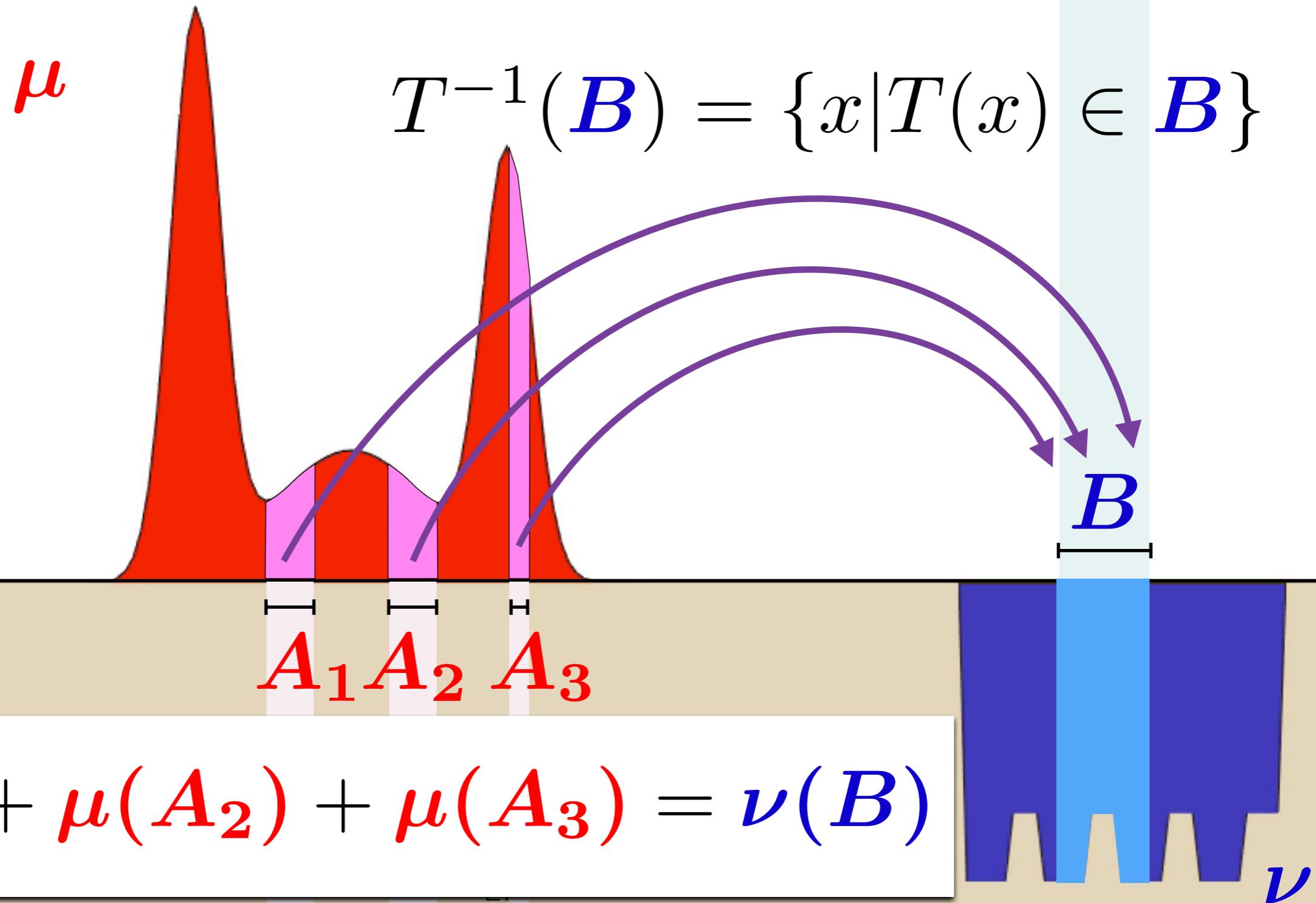
Origins: Monge's Problem

T must map red to blue.



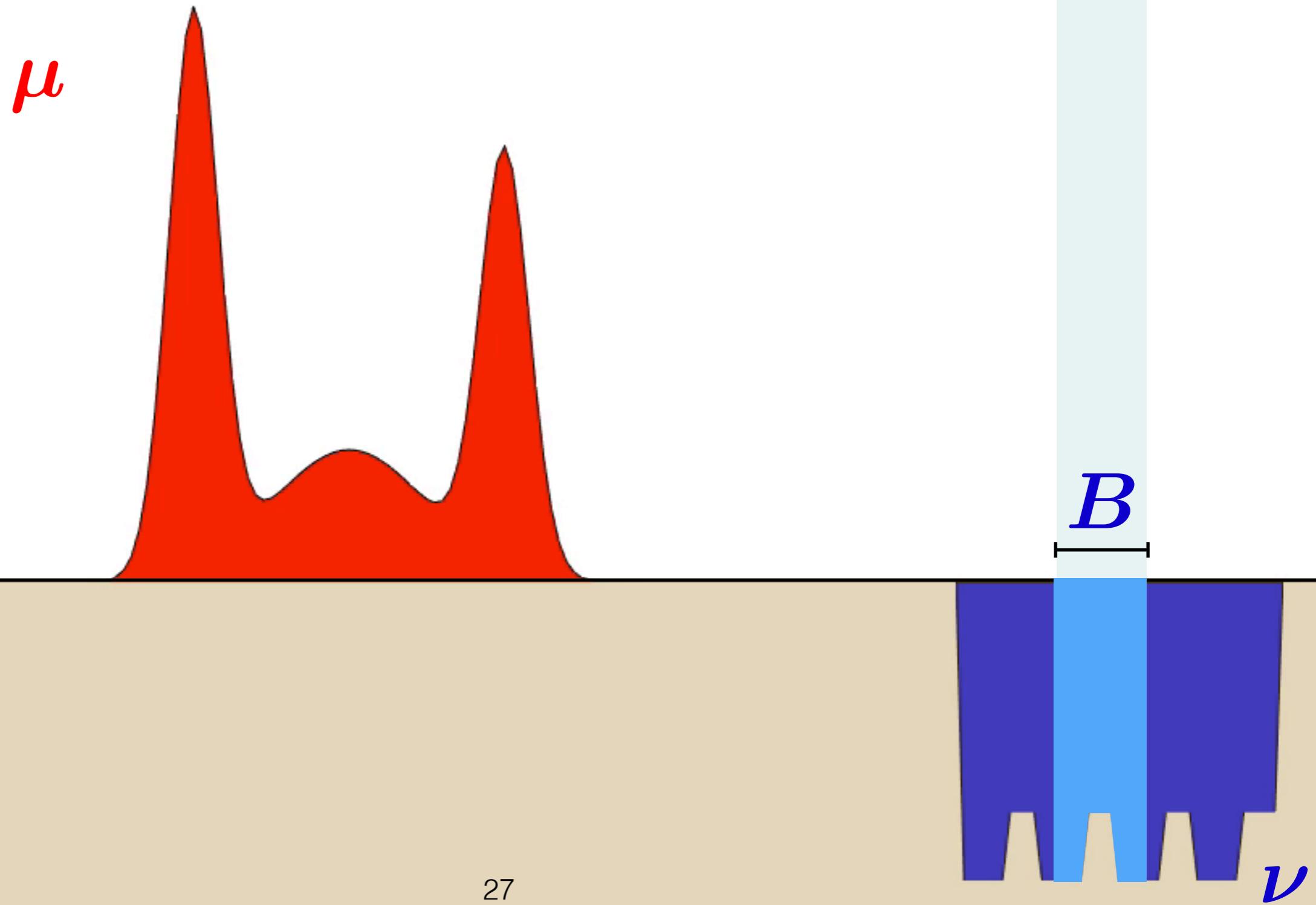
Origins: Monge's Problem

T must map red to blue.



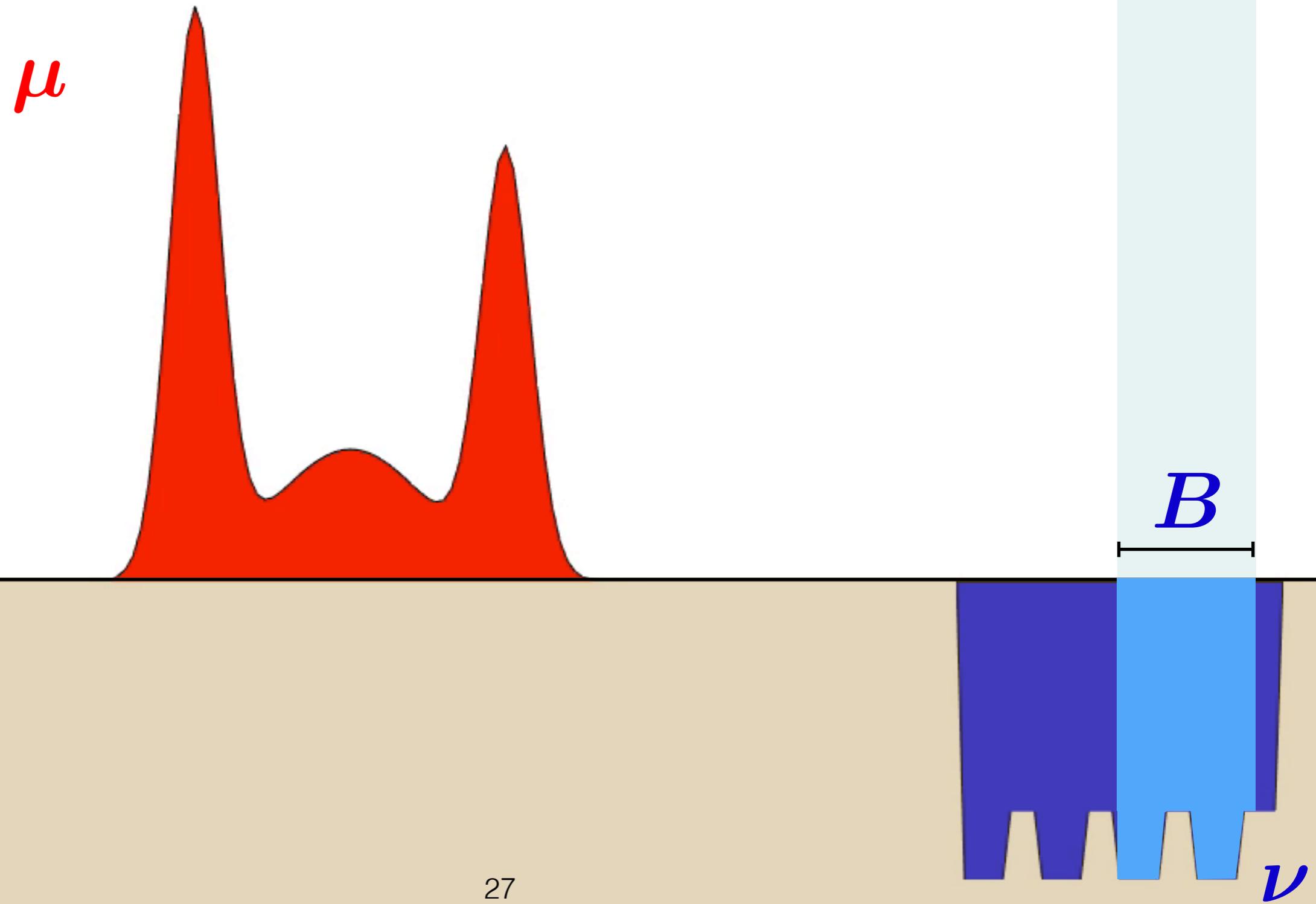
Origins: Monge's Problem

T must map red to blue.



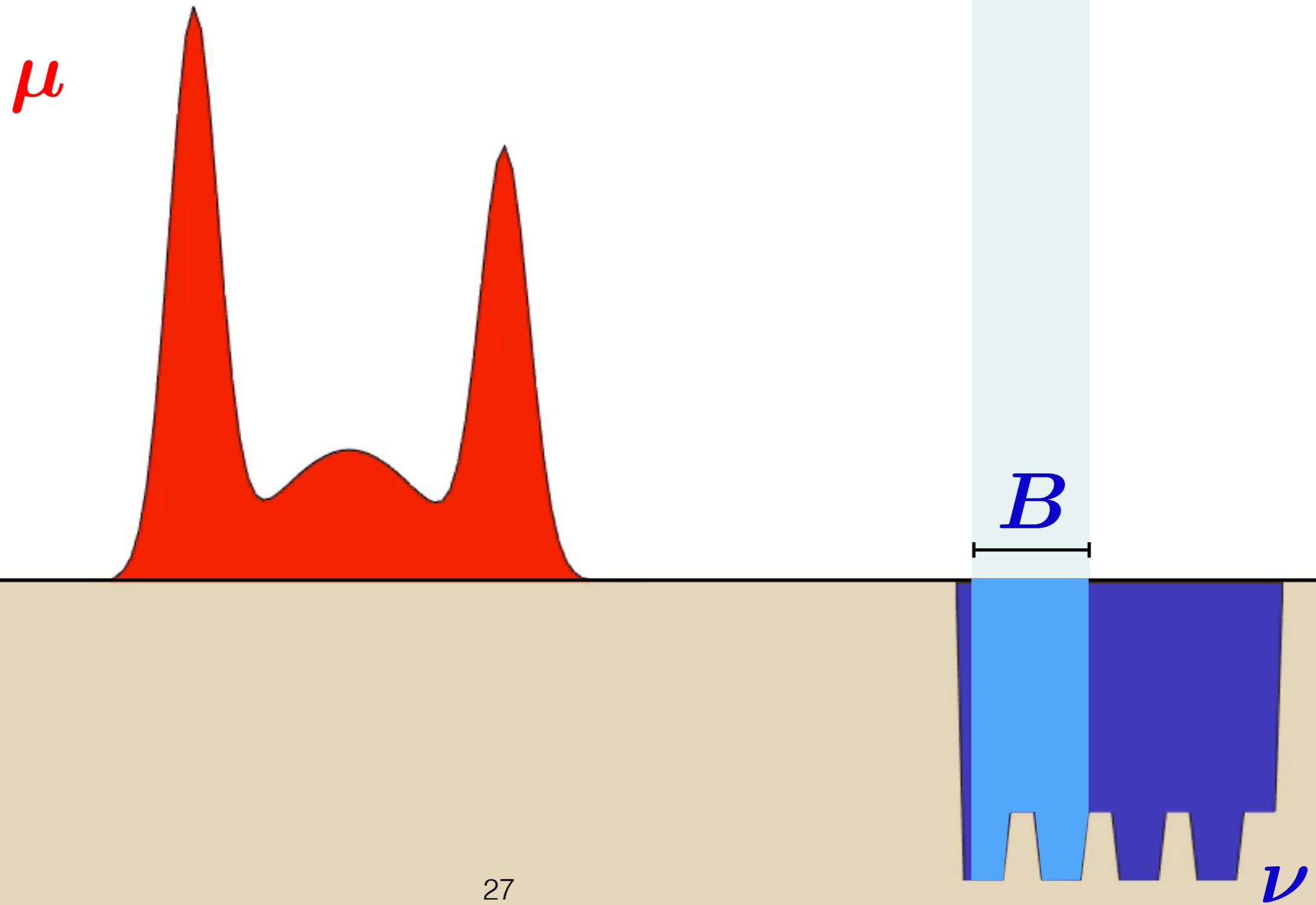
Origins: Monge's Problem

T must map red to blue.



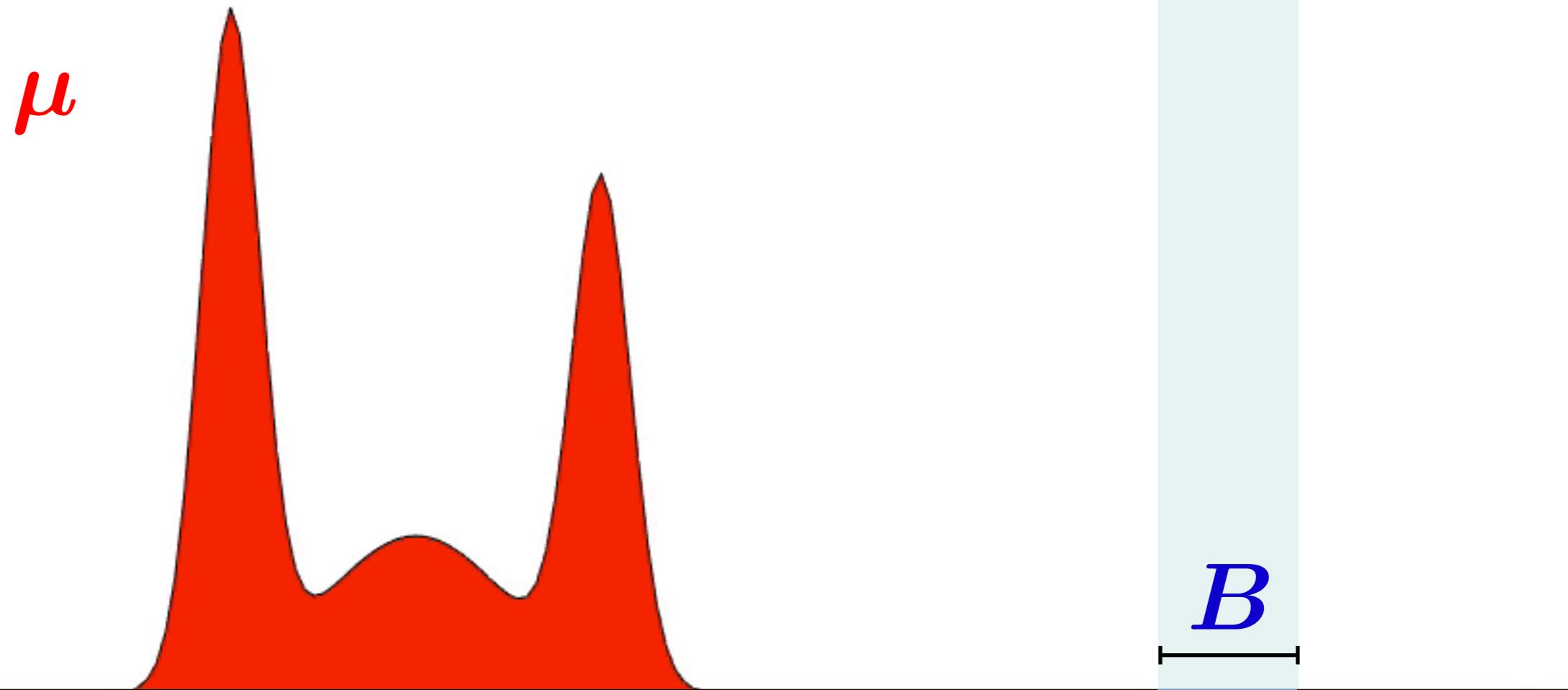
Origins: Monge's Problem

T must map red to blue.



Origins: Monge's Problem

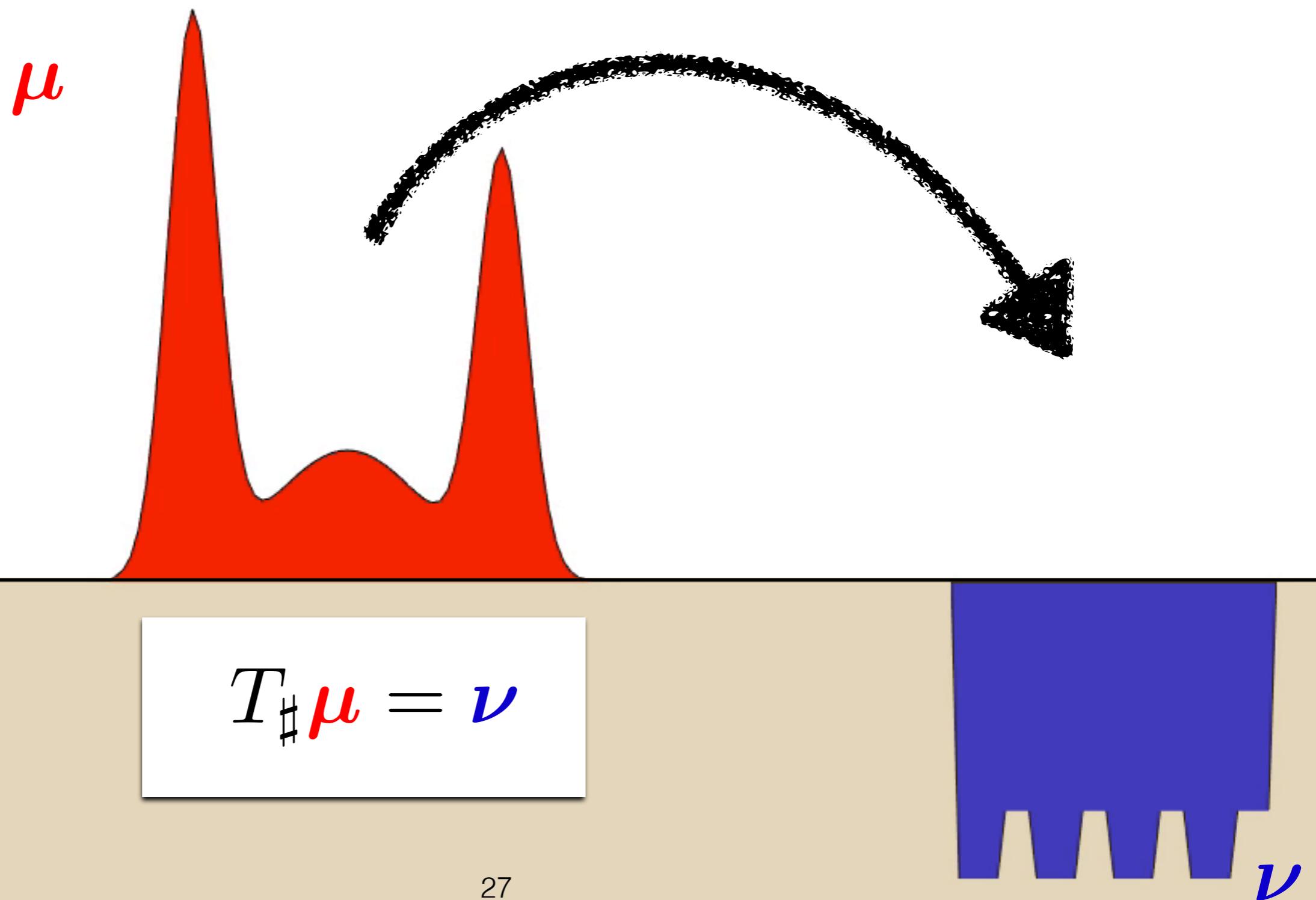
T must map red to blue.



$$\forall \mathbf{B}, \mu(T^{-1}(\mathbf{B})) = \nu(\mathbf{B})$$

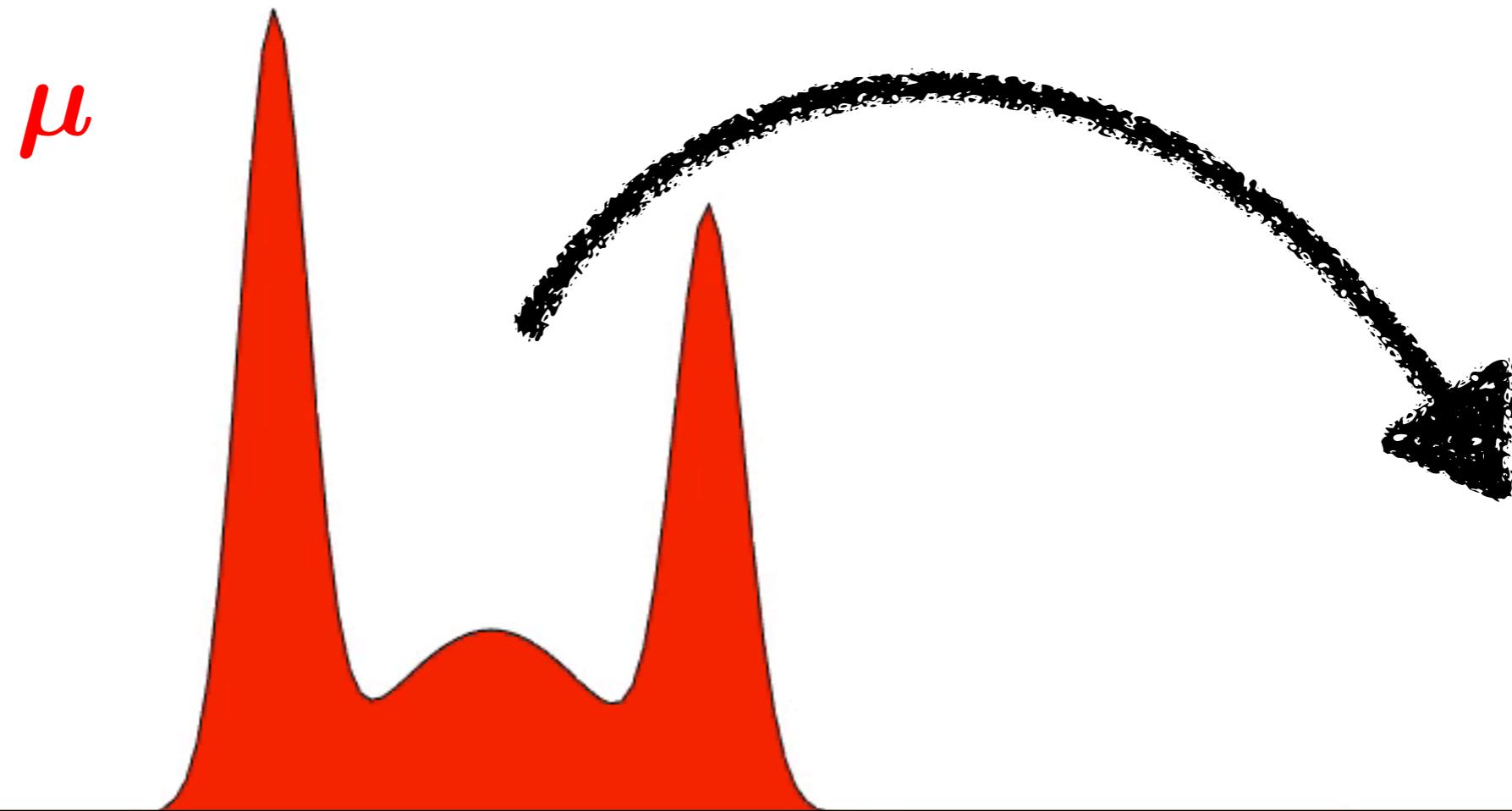
Origins: Monge's Problem

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. $T_{\#}\mu = \nu$
minimizes $\int D(x, T(x))\mu(dx)$?

Kantorovich Problem



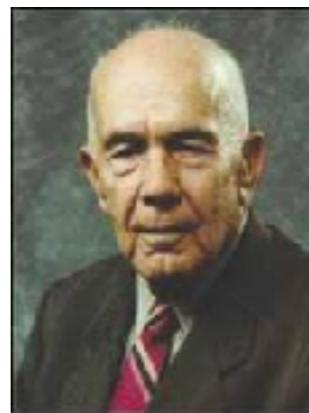
Kantorovich



1939



Tolstoi
1930



Hitchcock

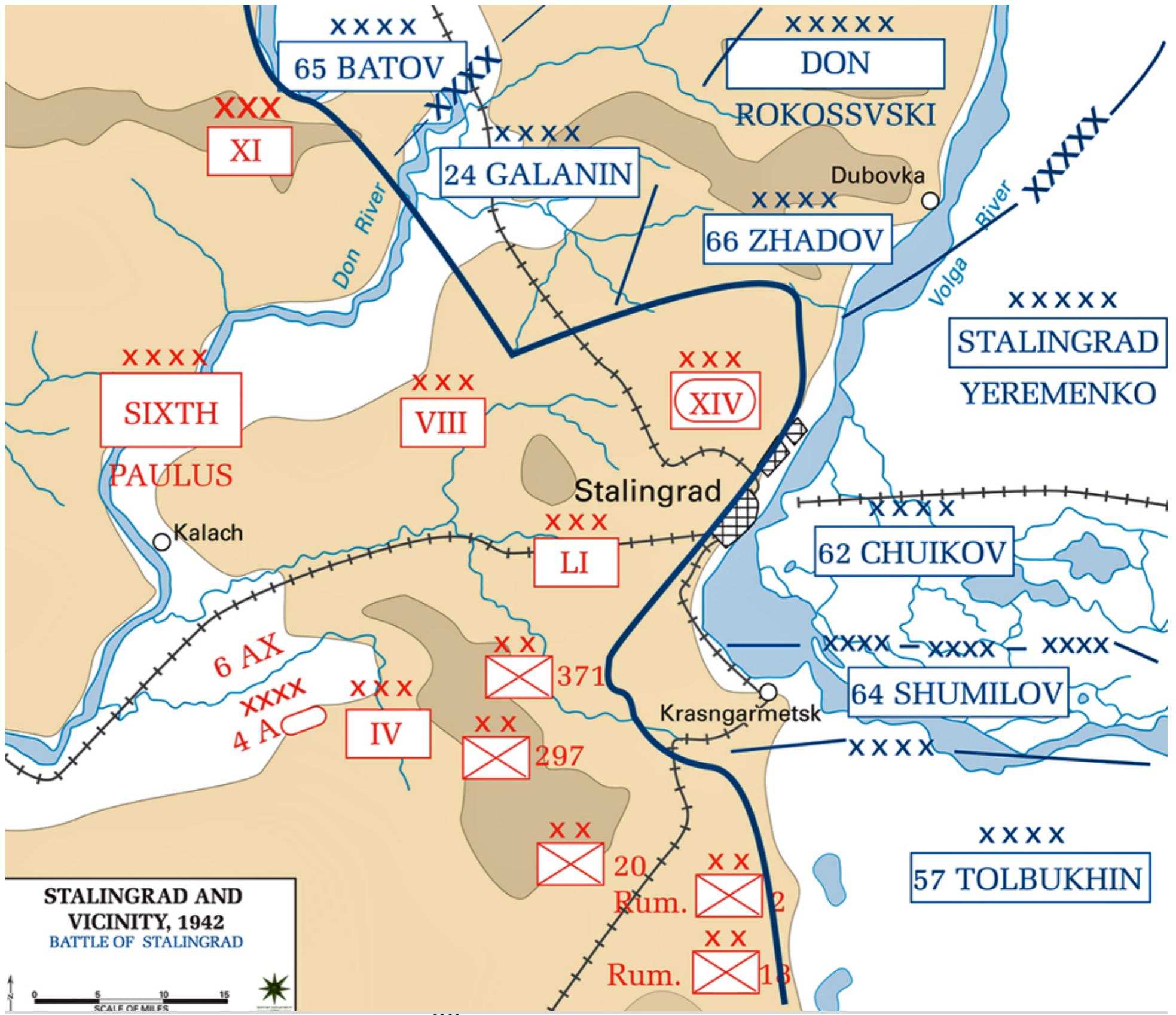
THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

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

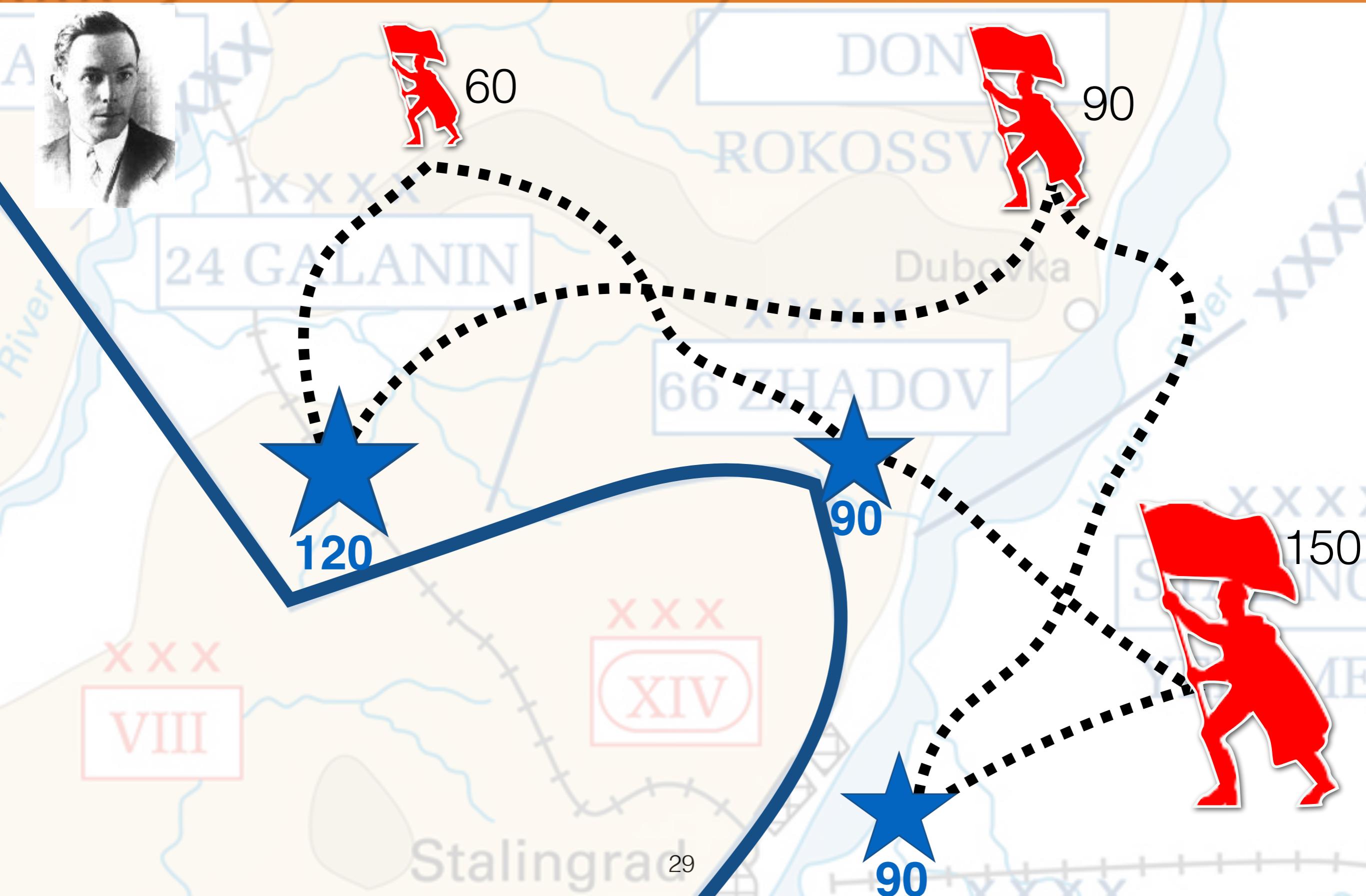
Kantorovich Problem



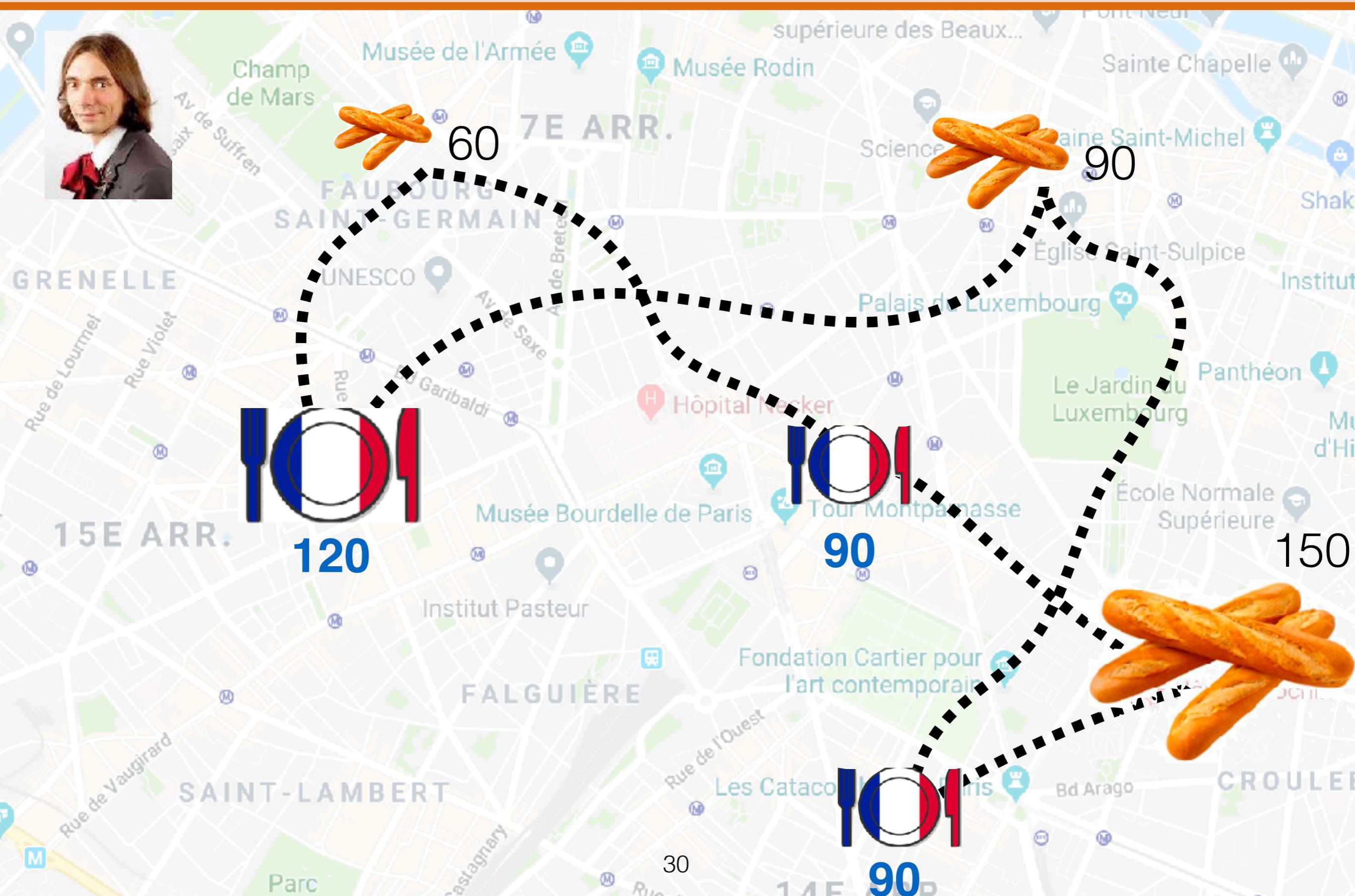
STALINGRAD AND
VICINITY, 1942
BATTLE OF STALINGRAD

0 5 10 15
SCALE OF MILES

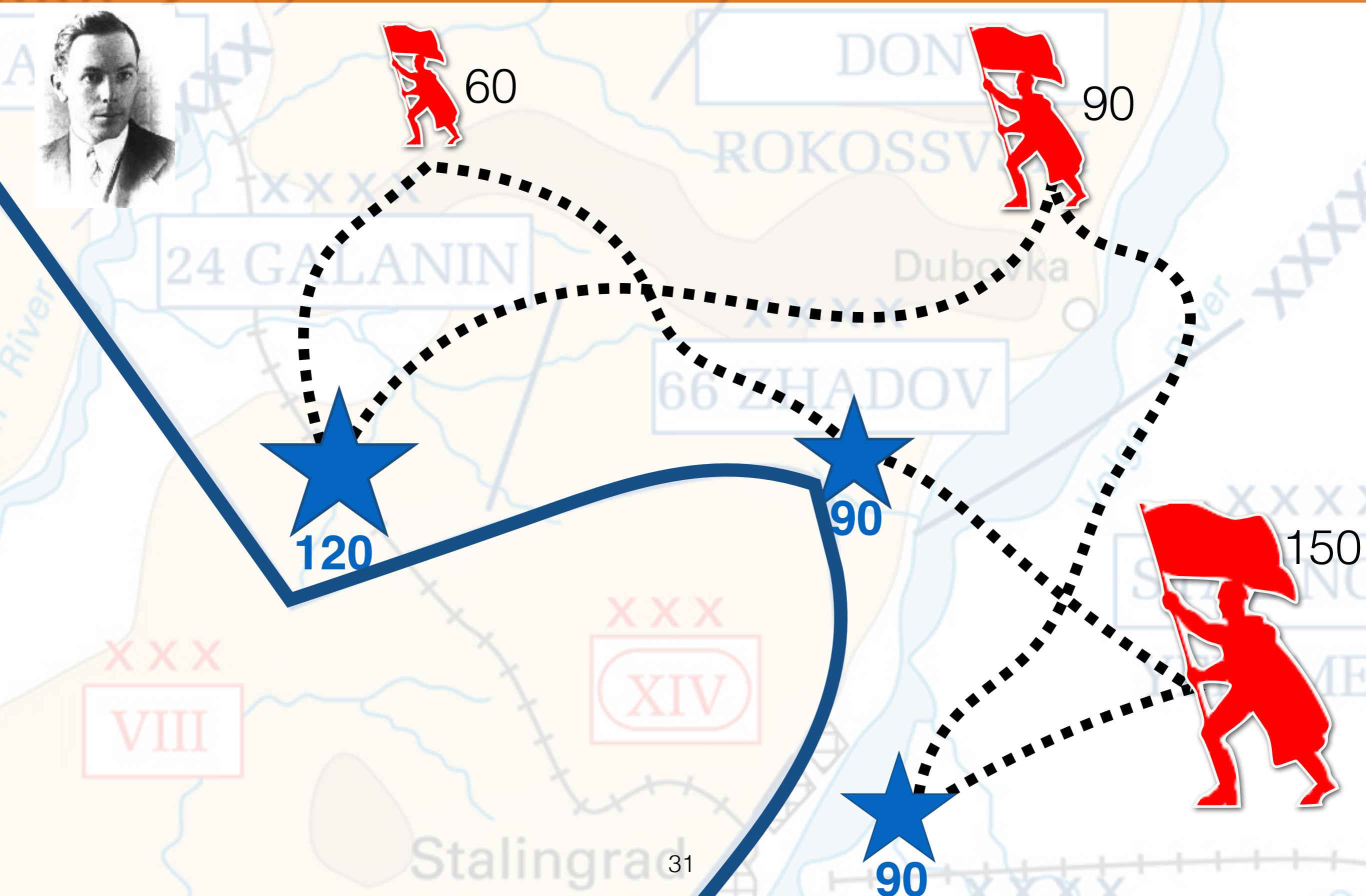
Kantorovich Problem



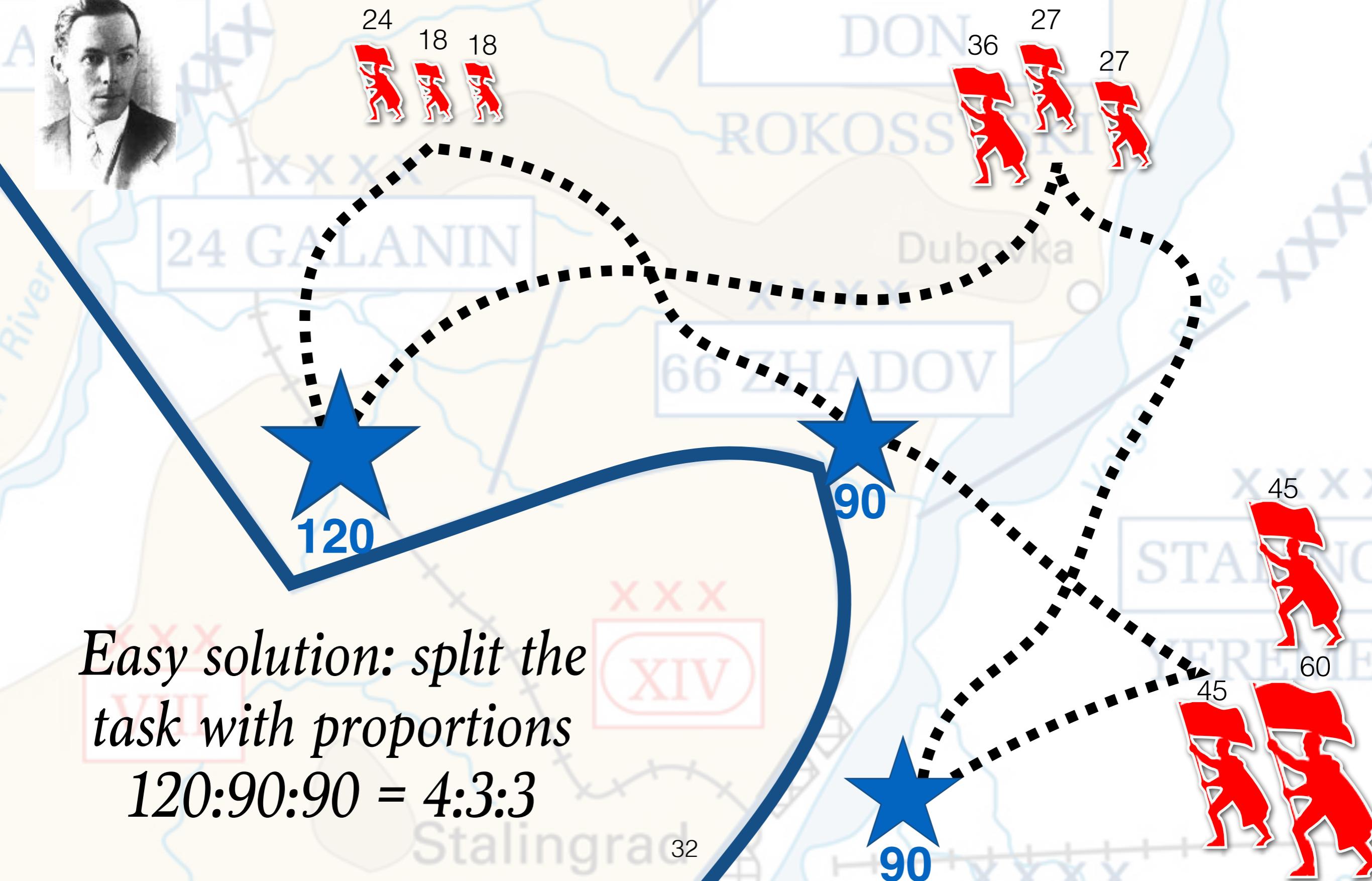
Kantorovich Problem à la française



Kantorovich Problem



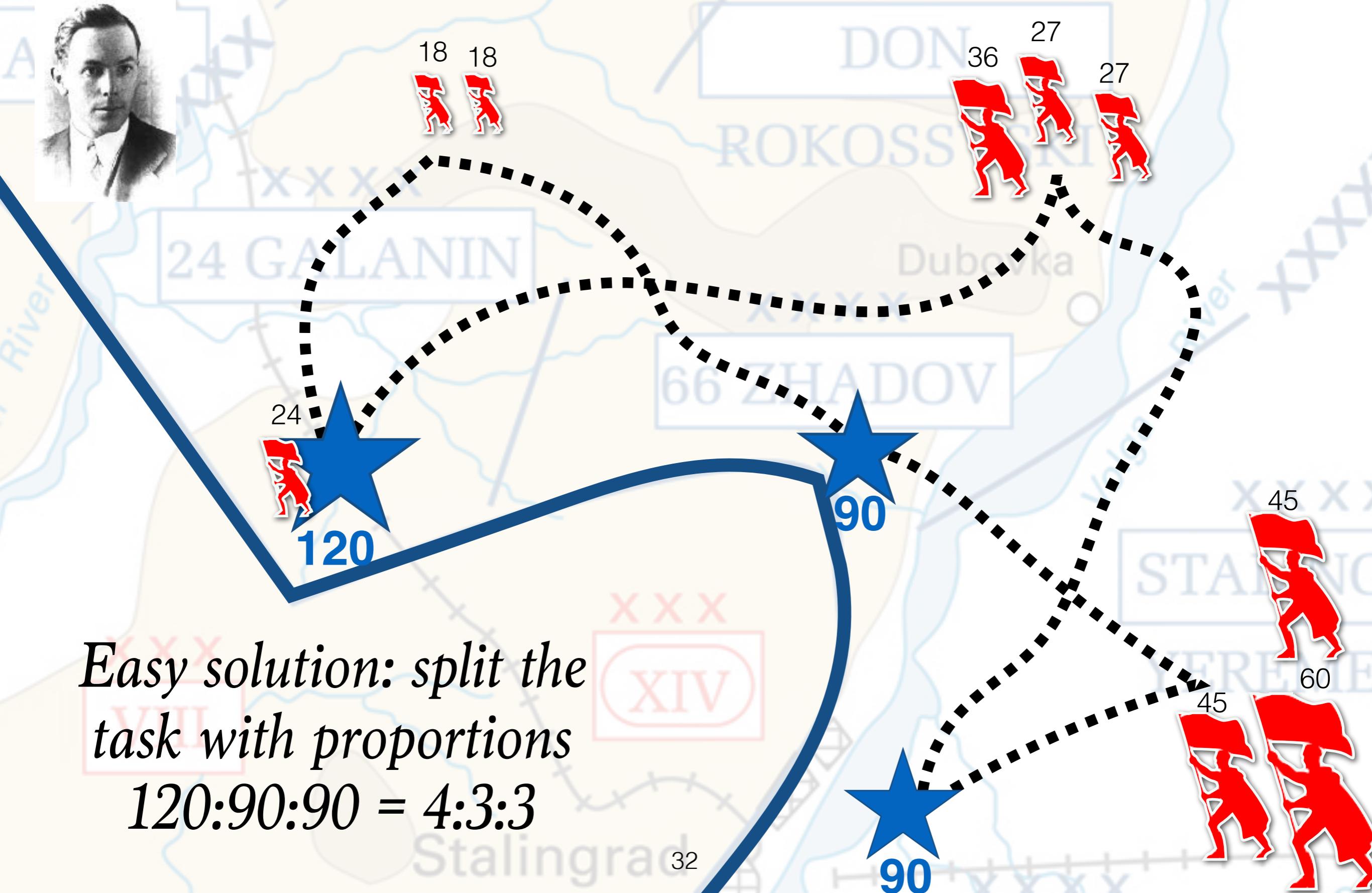
Kantorovich Problem



Easy solution: split the task with proportions

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

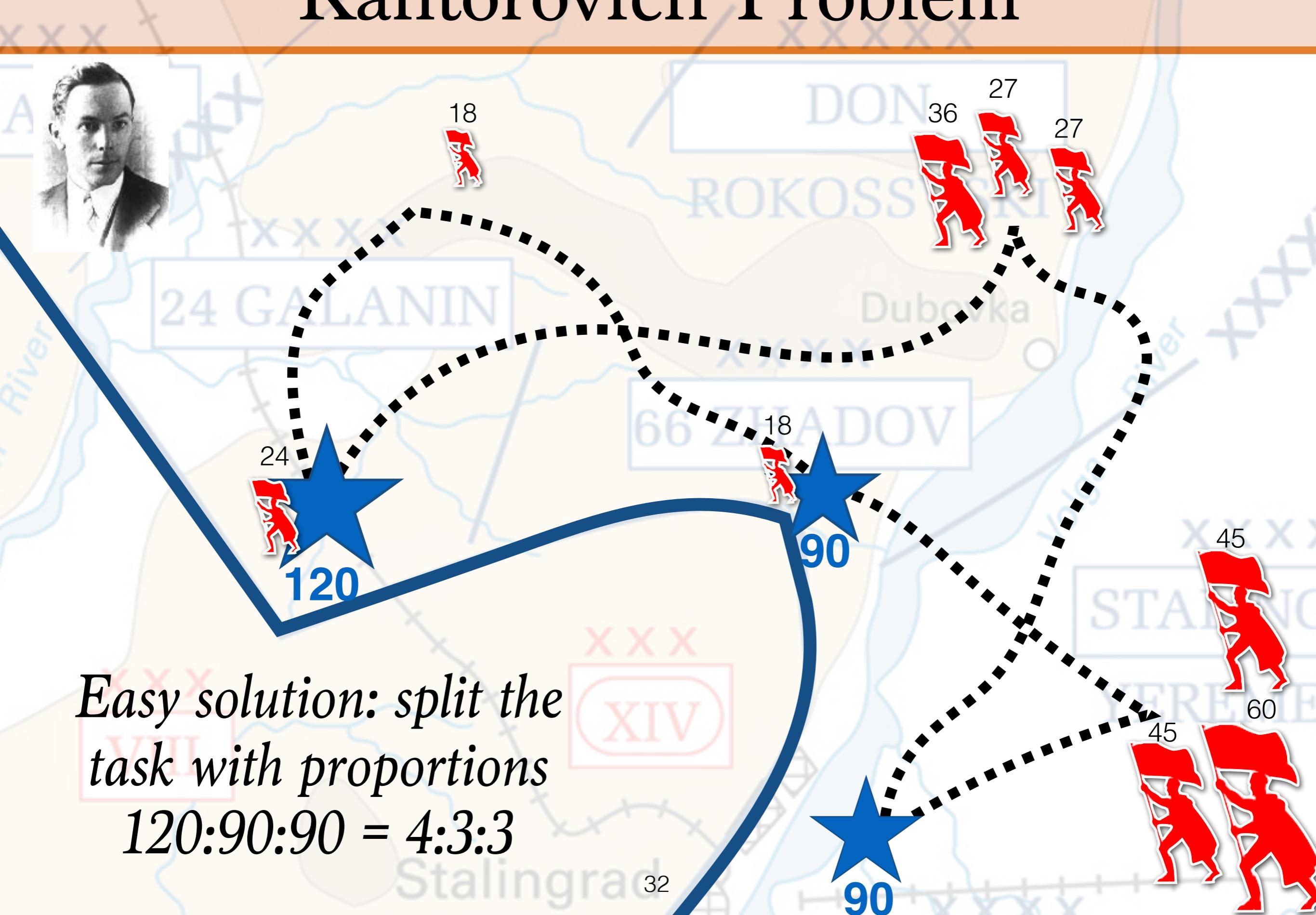
Kantorovich Problem



Easy solution: split the task with proportions

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

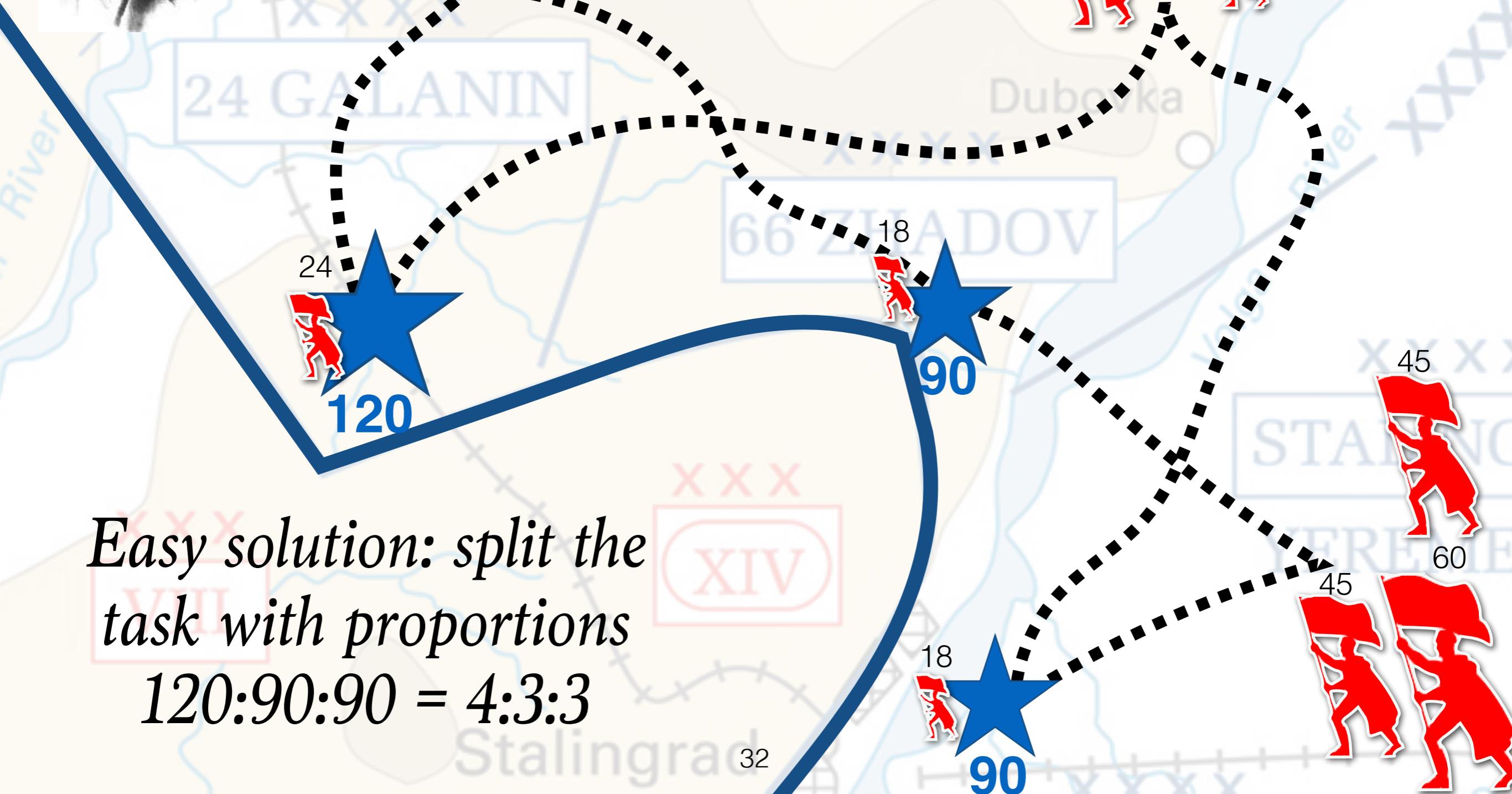
Kantorovich Problem



Easy solution: split the task with proportions

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

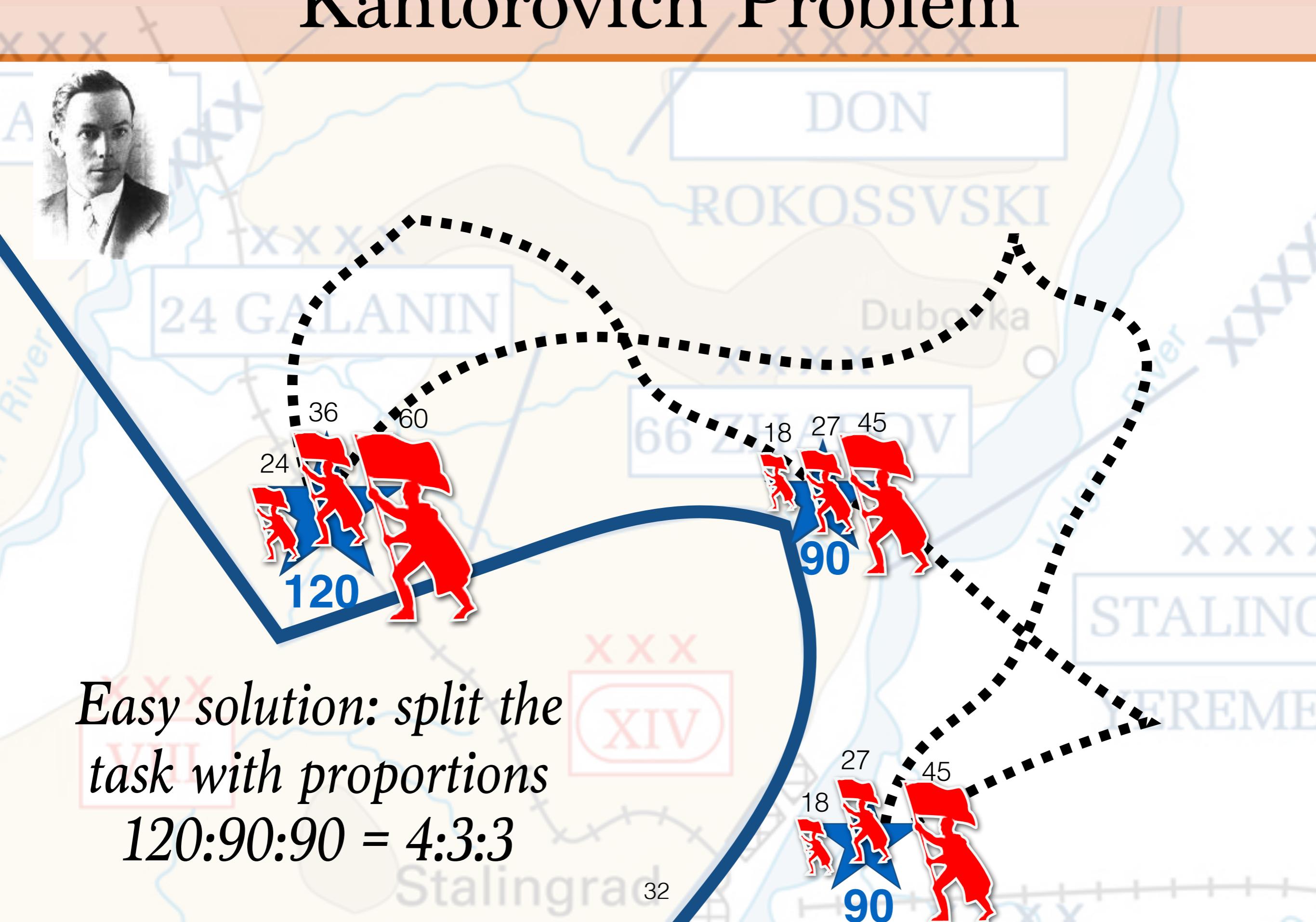
Kantorovich Problem



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



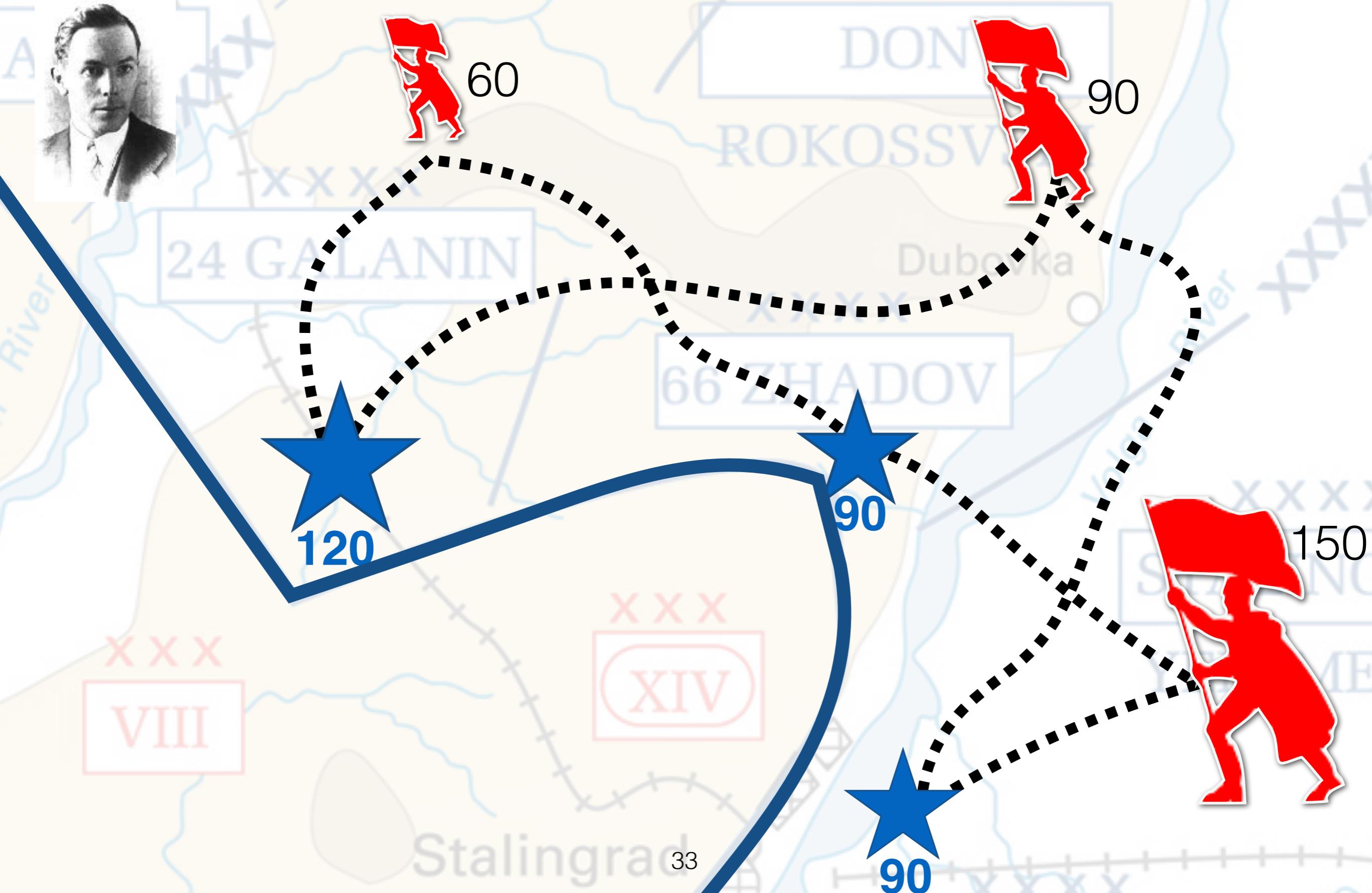
Naive approach results in
many displacements...

Can we find a cheaper
alternative?

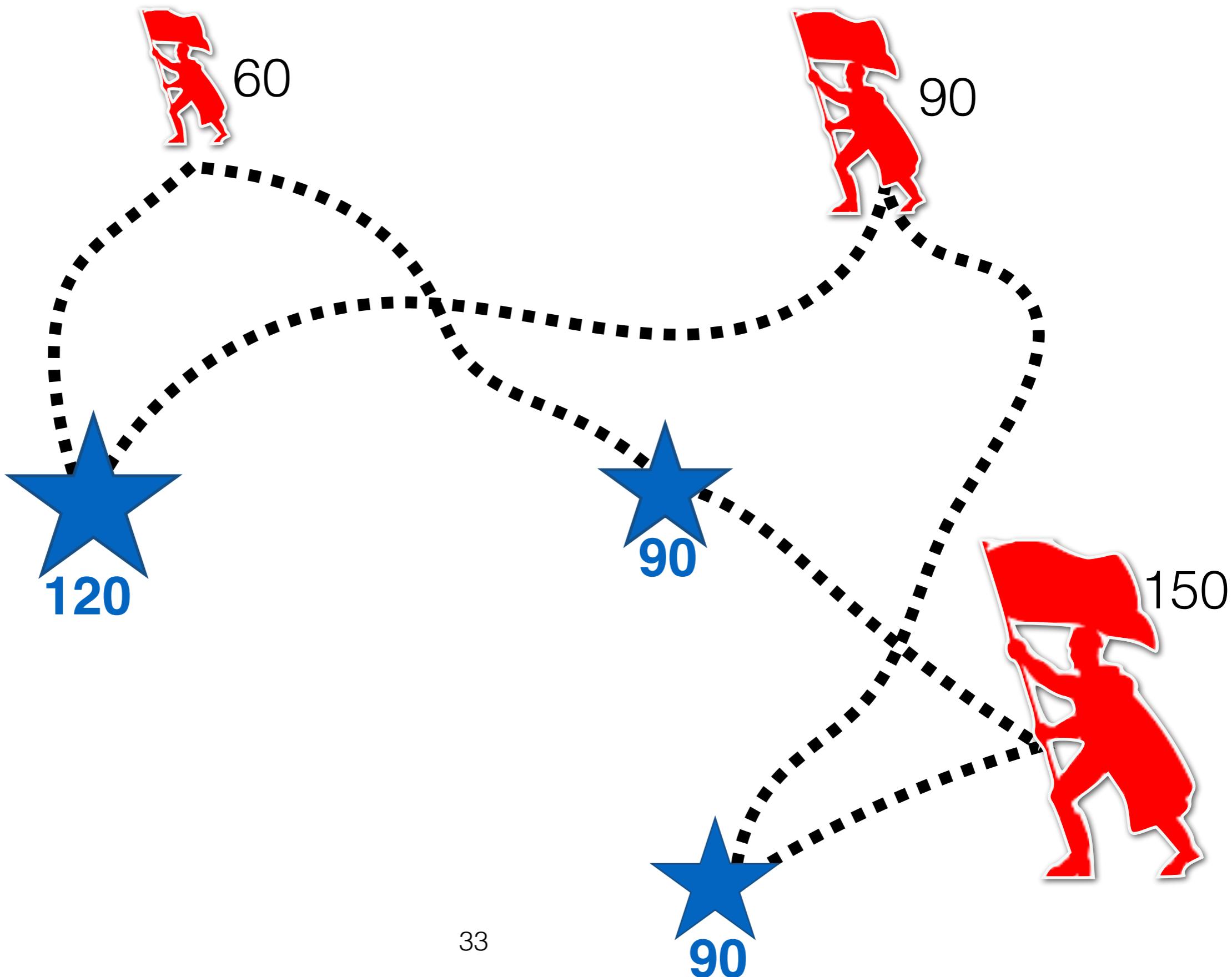
*Easy solution: split the
task with proportions*

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

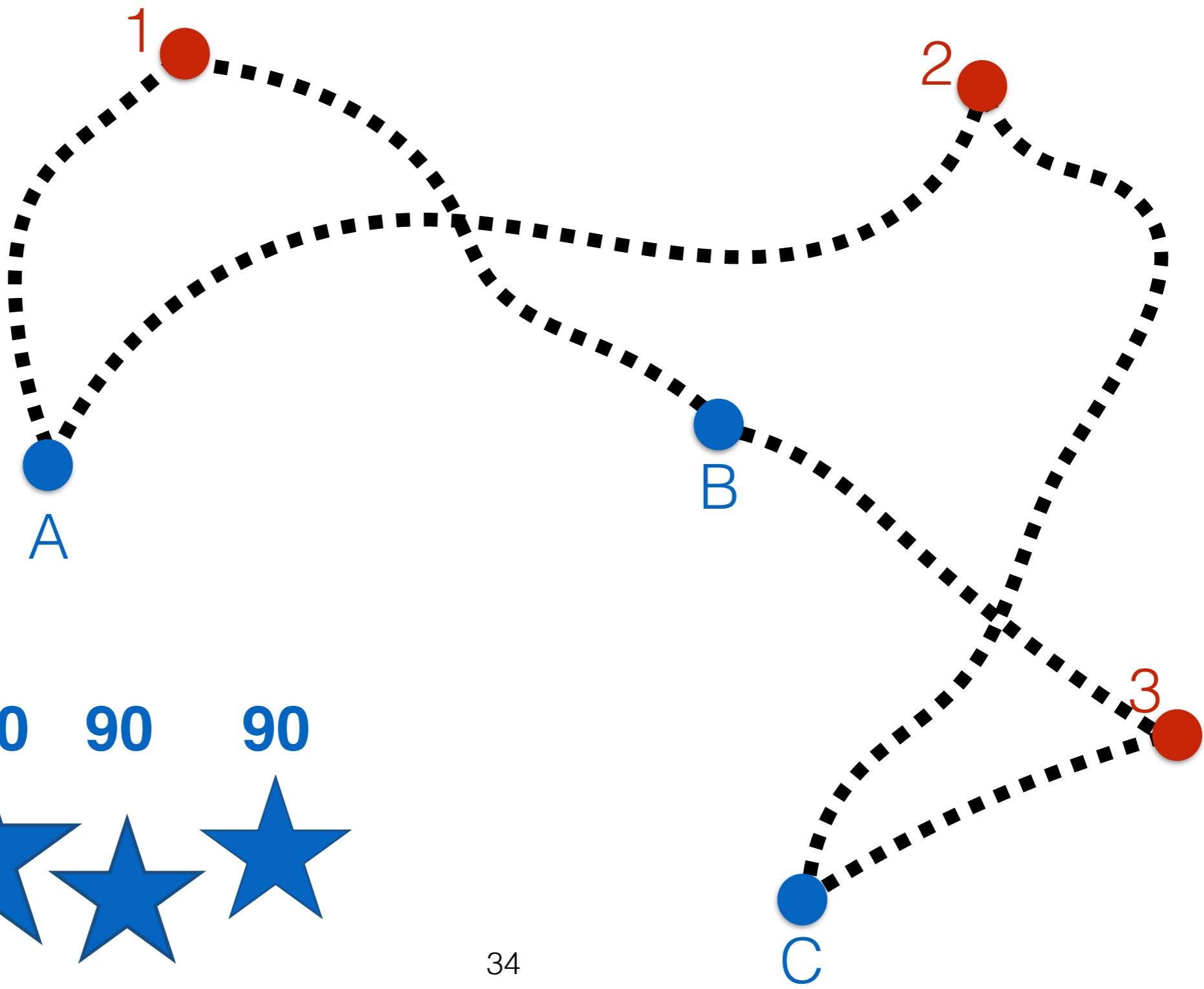
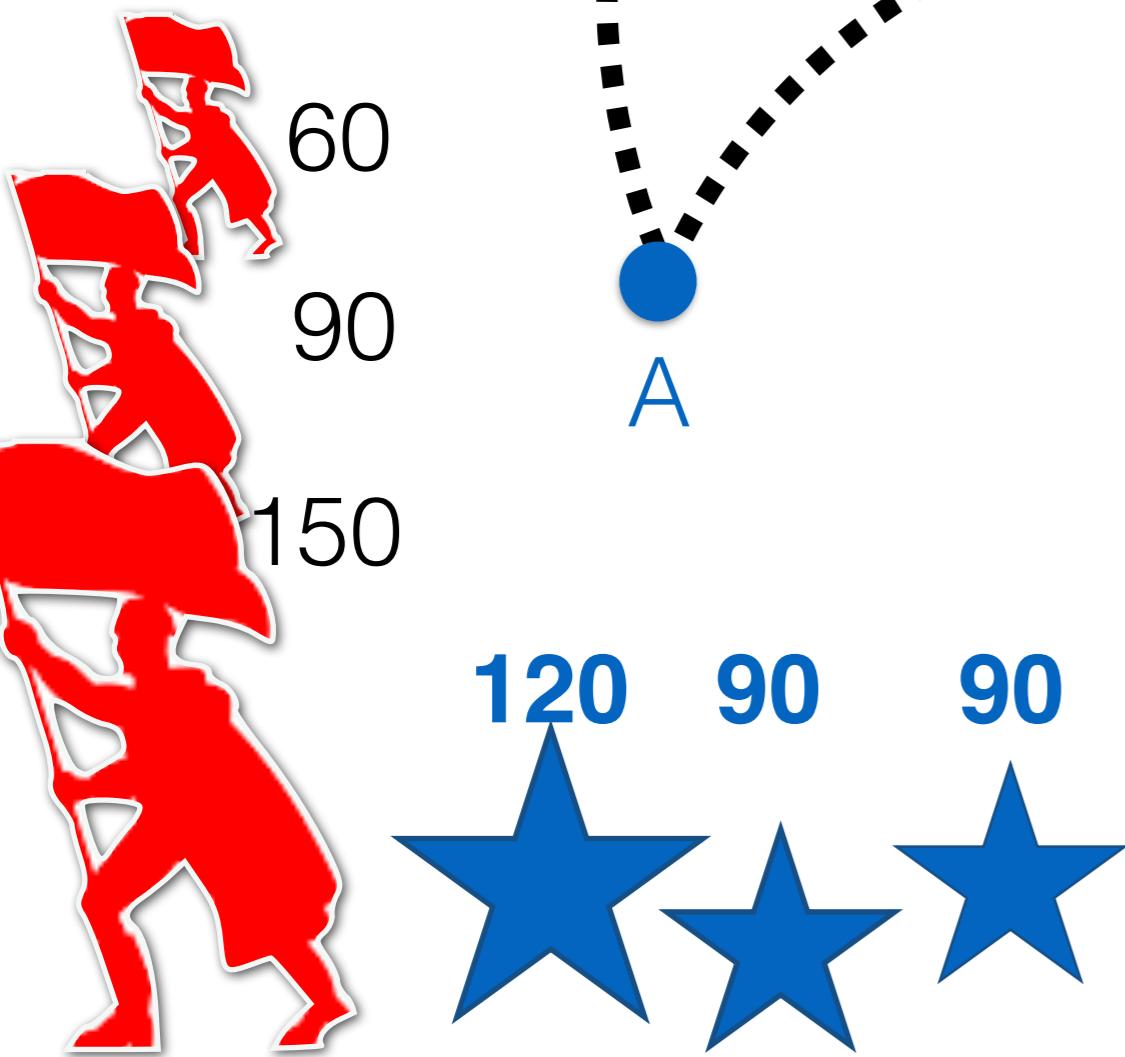
Kantorovich Problem



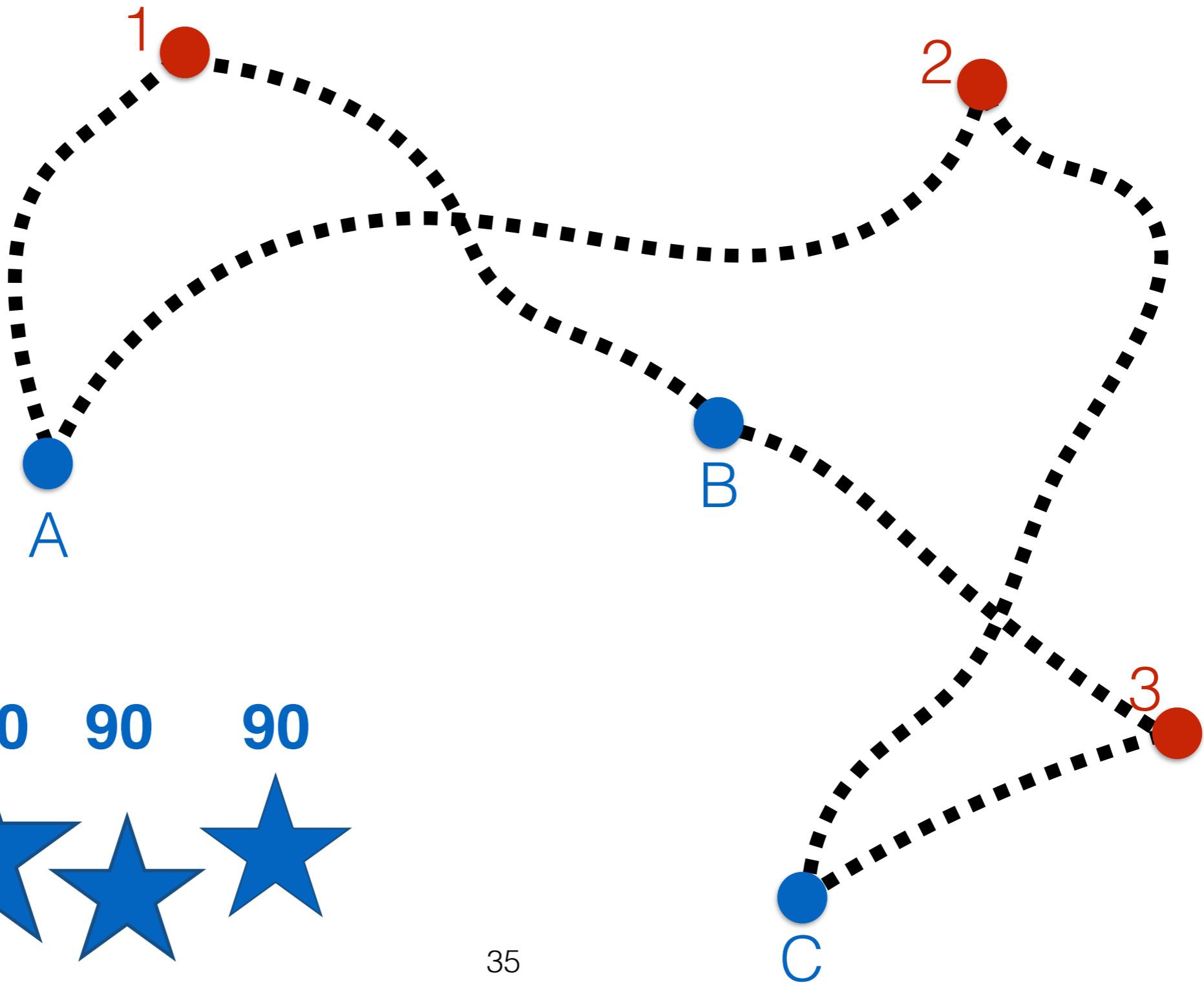
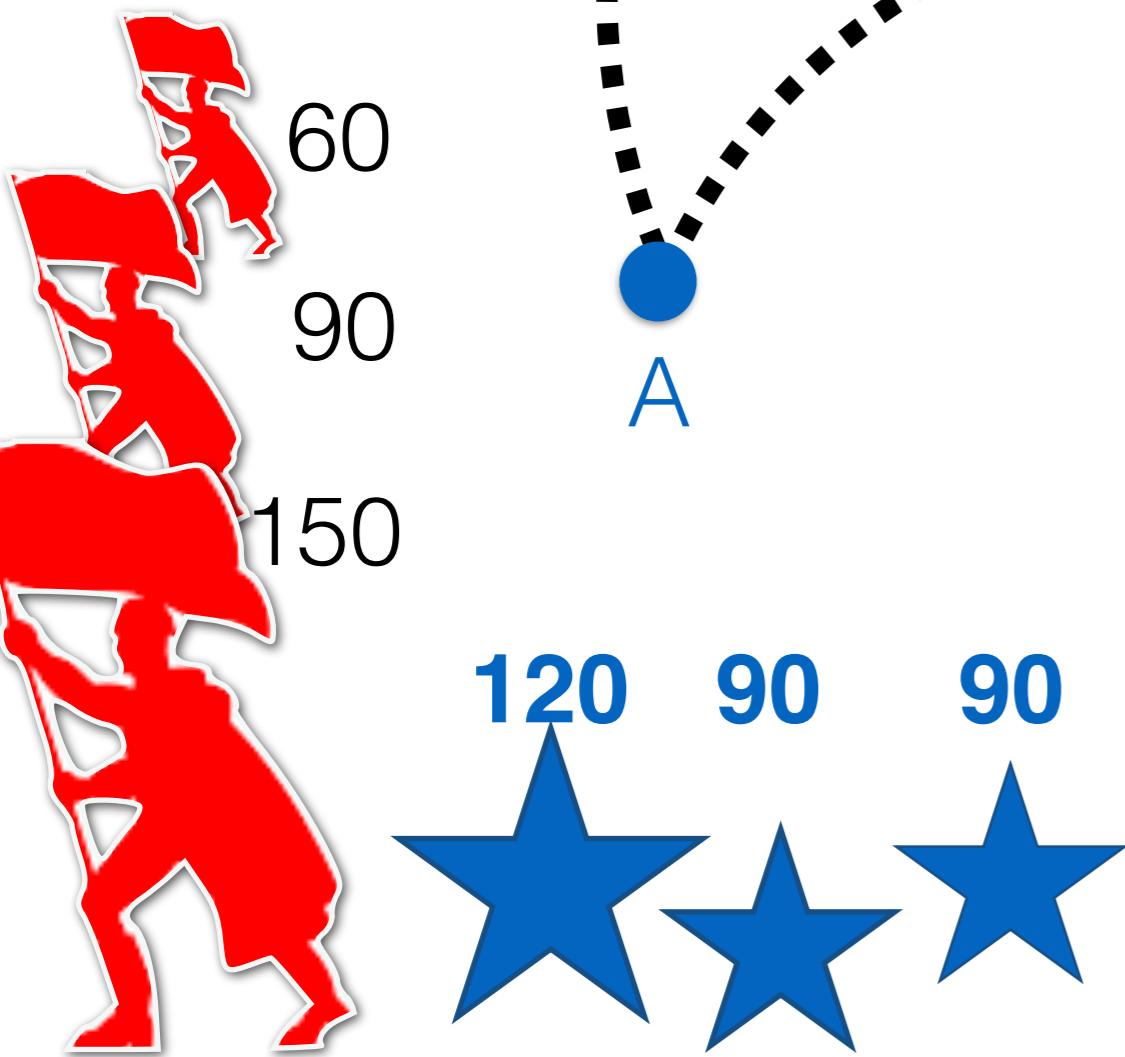
Kantorovich Problem



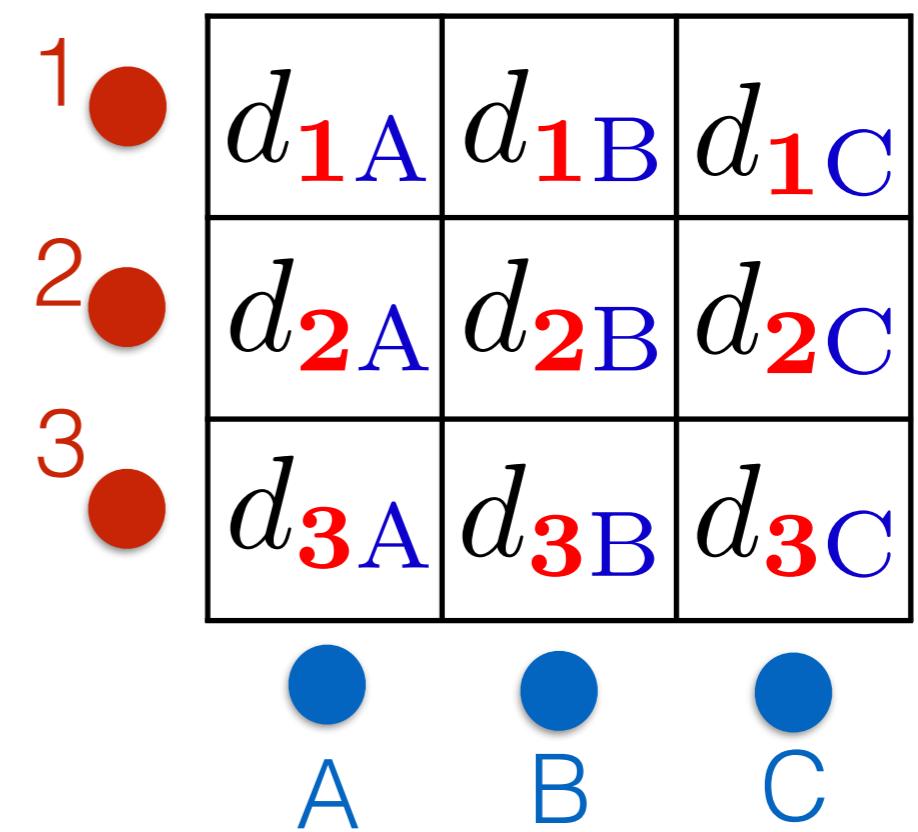
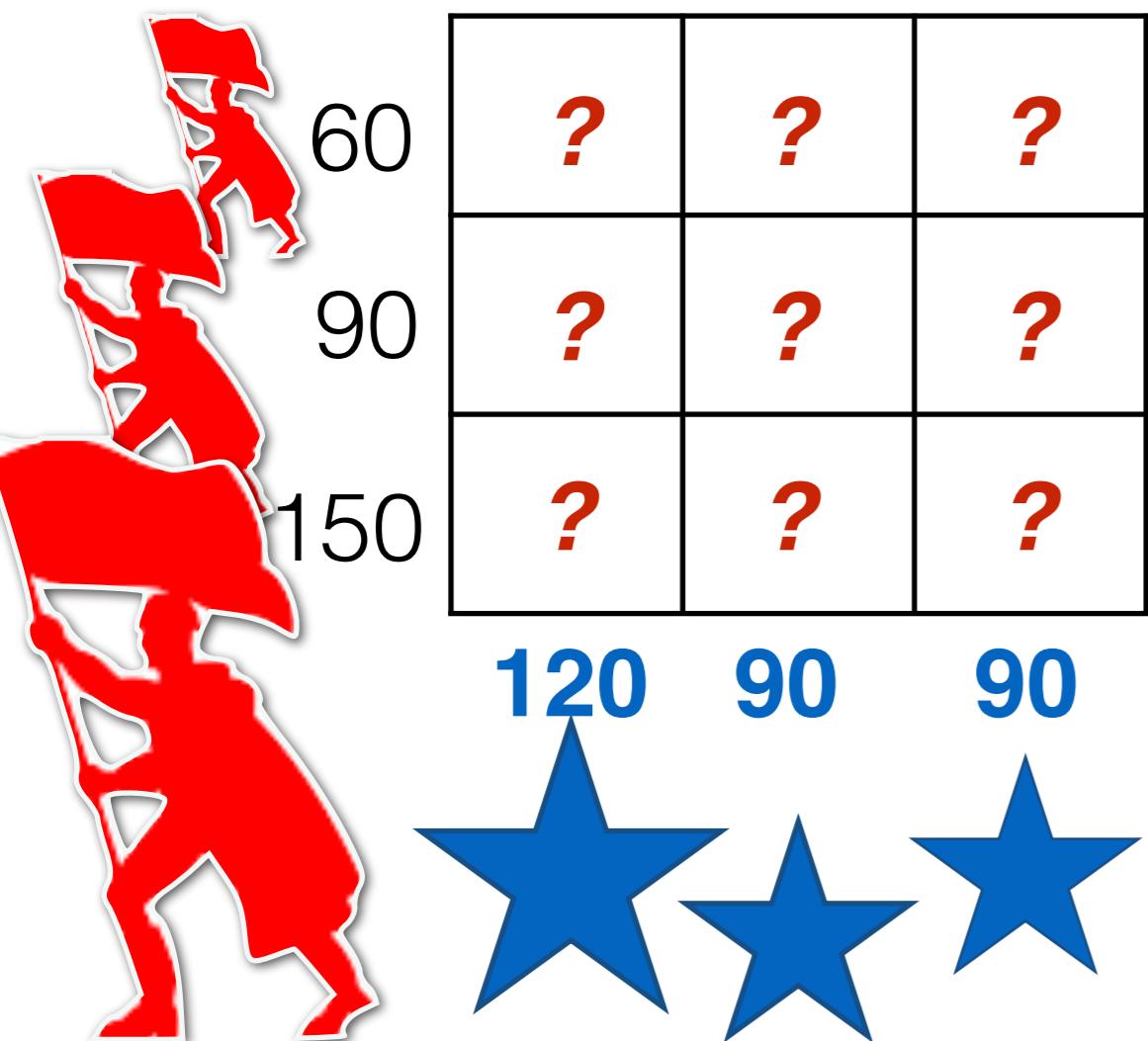
Kantorovich Problem



Kantorovich Problem



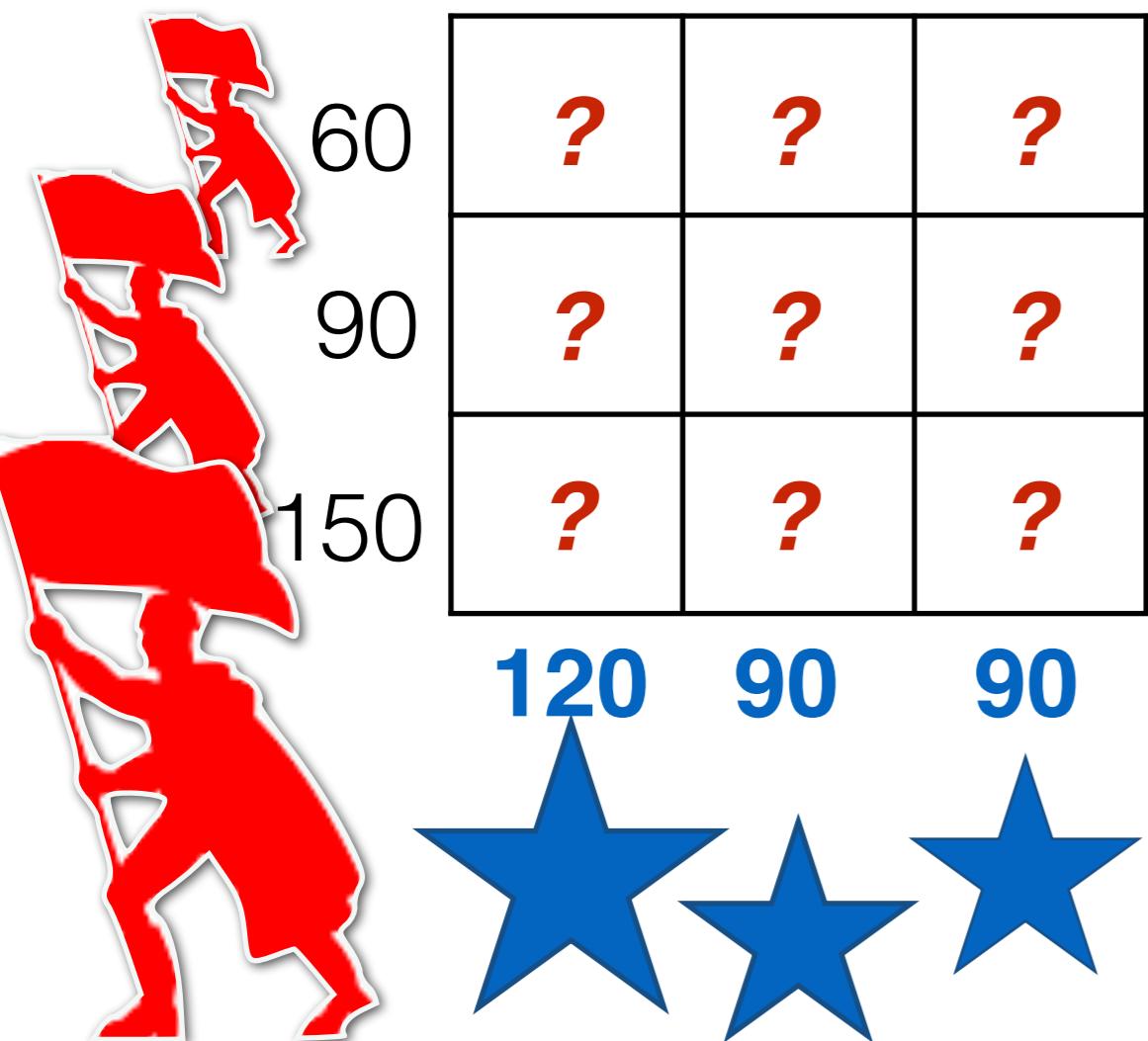
Kantorovich Problem



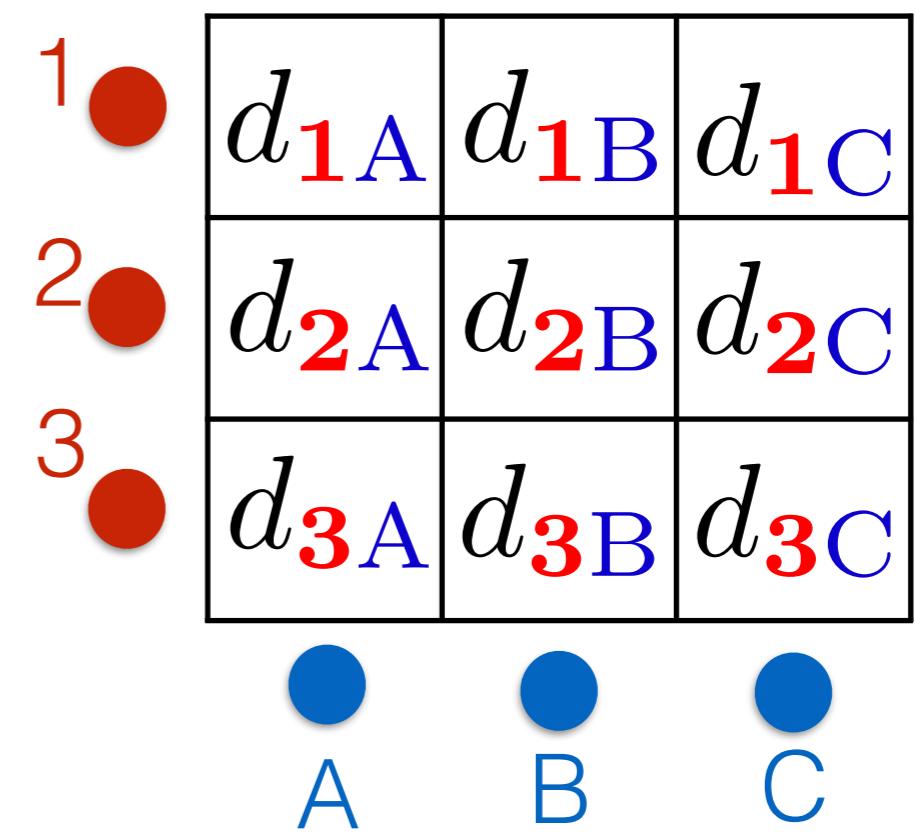
Kantorovich Problem



Transportation matrix



Distance matrix

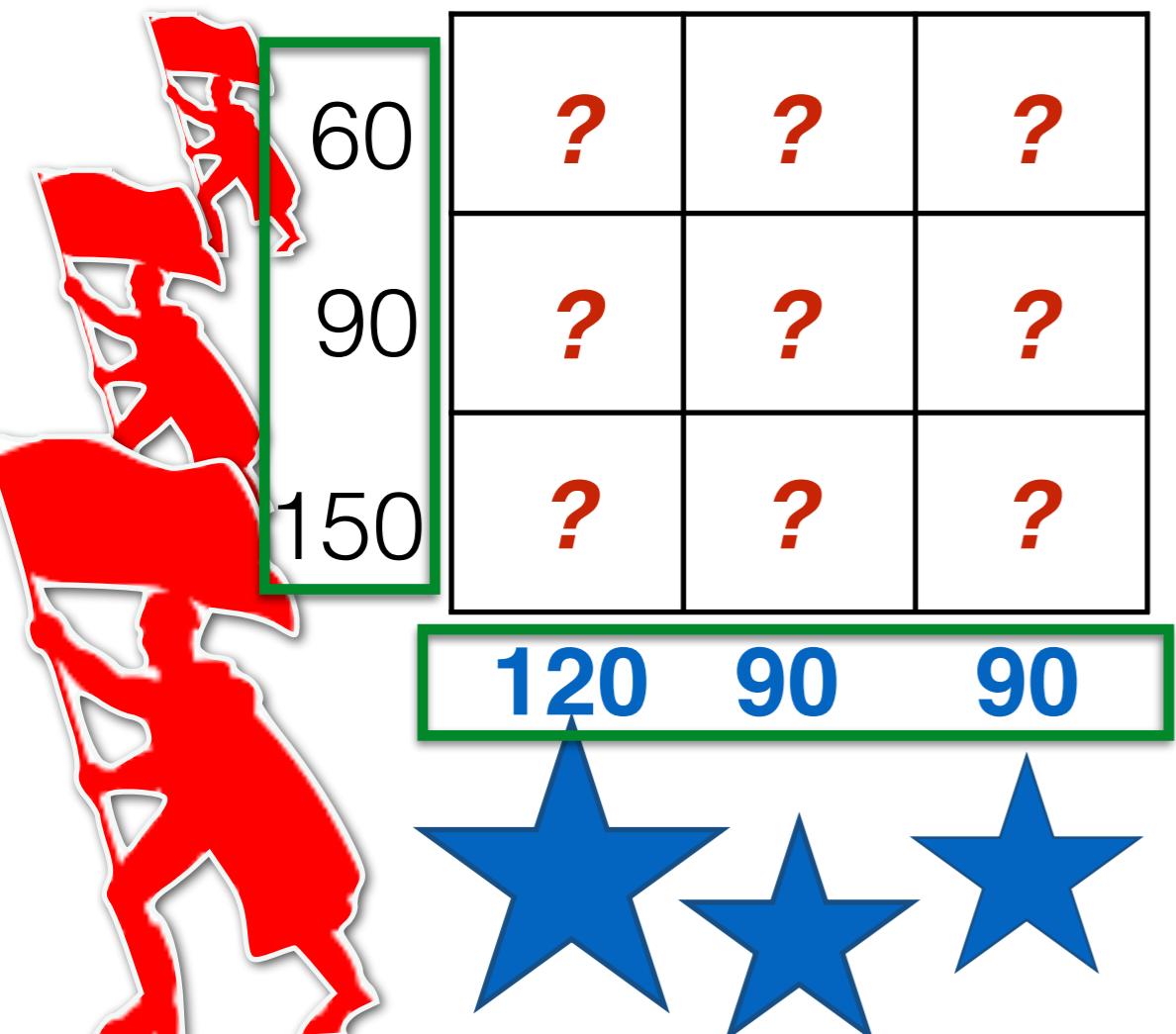


Kantorovich Problem



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

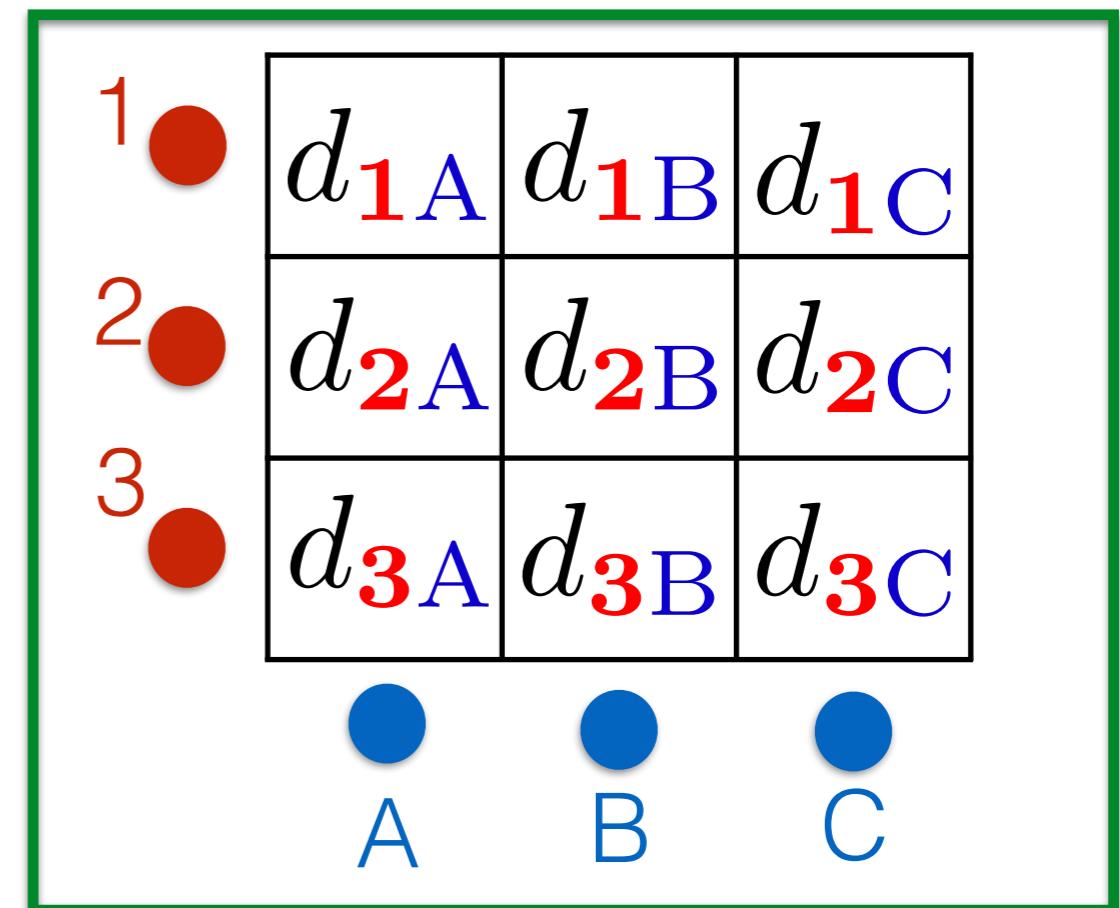
Transportation matrix

A large, semi-transparent watermark of a red elephant is visible on the left side of the slide.

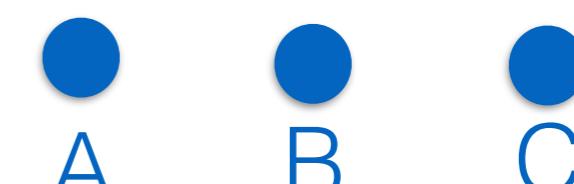
60	?	?	?
90	?	?	?
150	?	?	?
	120	90	90

Three blue five-pointed stars are arranged horizontally at the bottom of the matrix.

Distance matrix

A green rectangular border surrounds the distance matrix and its labels.

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

Three blue circles are arranged horizontally at the bottom, corresponding to the labels A, B, and C.

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}

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

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

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

$$p_{ij} \geq 0$$

Cost function

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

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

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

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

$$p_{ij} \geq 0$$

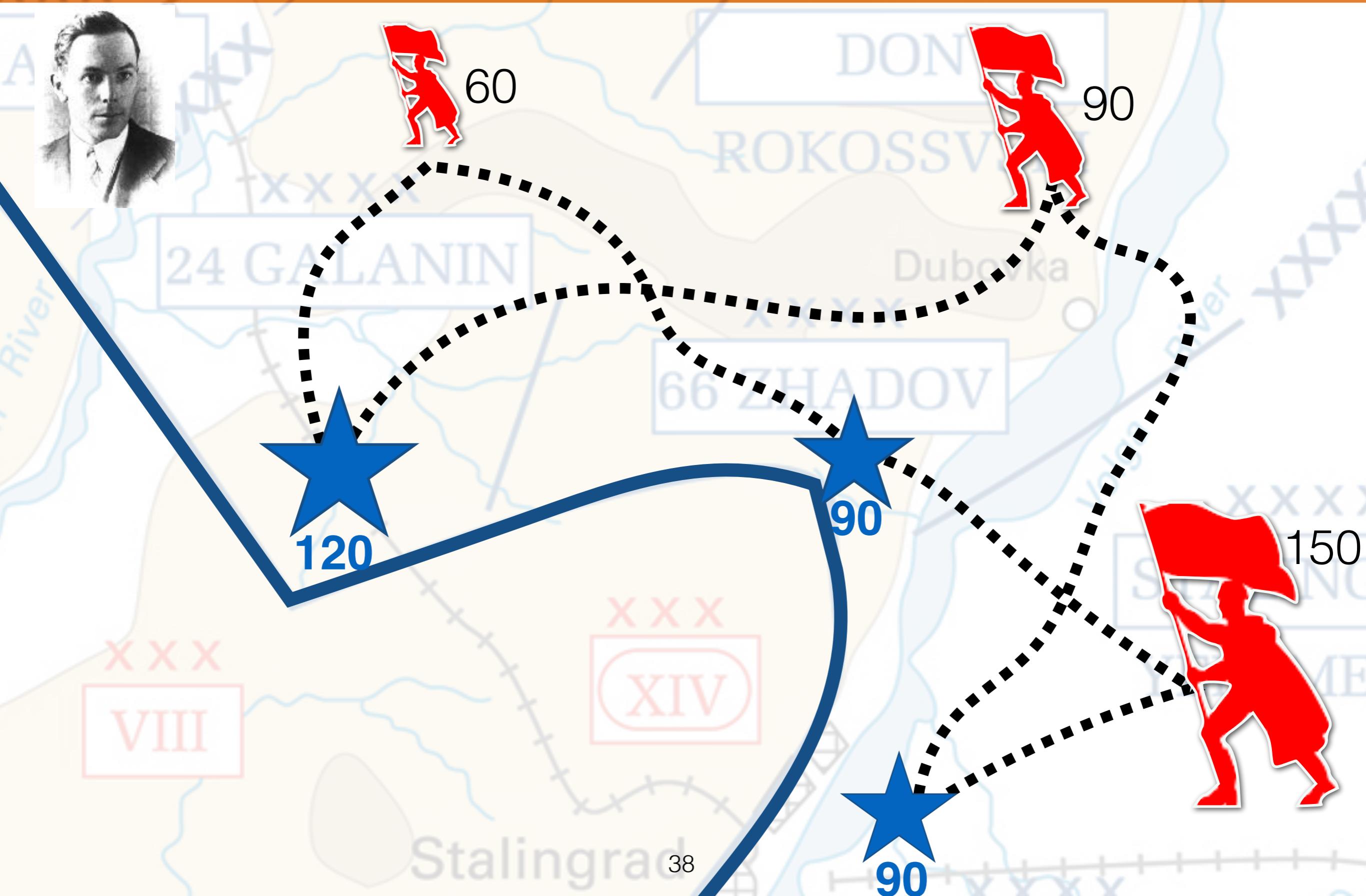
Cost function

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

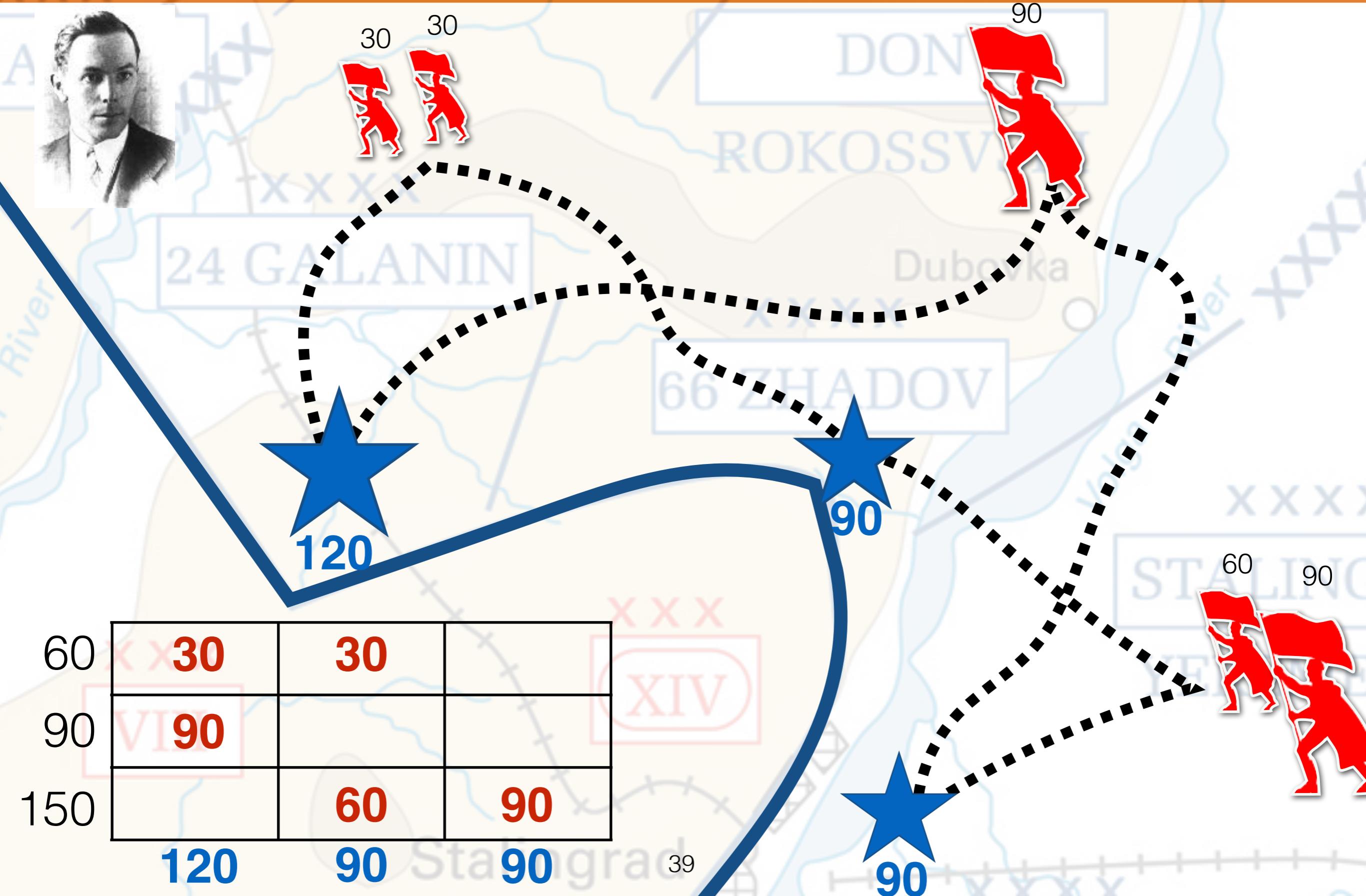
Problem

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

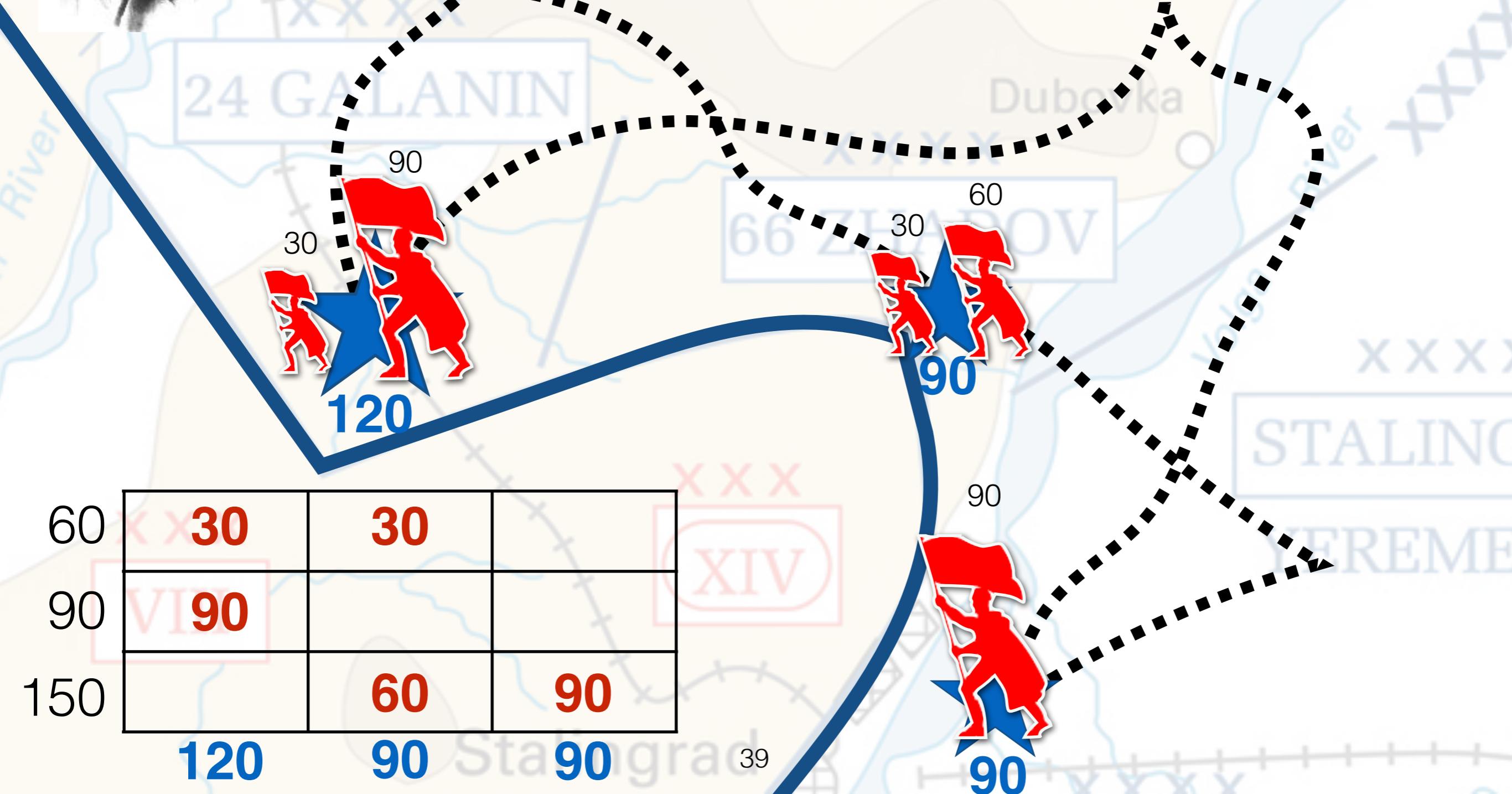
Kantorovich Problem



Kantorovich Problem



Kantorovich Problem

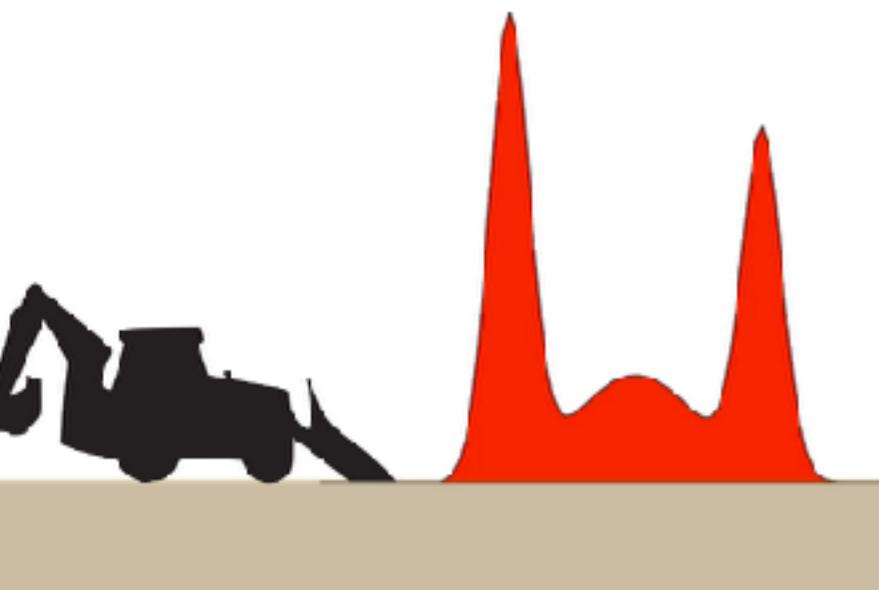


30	30	
90		
	60	90

120 90 90

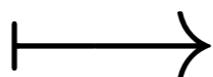
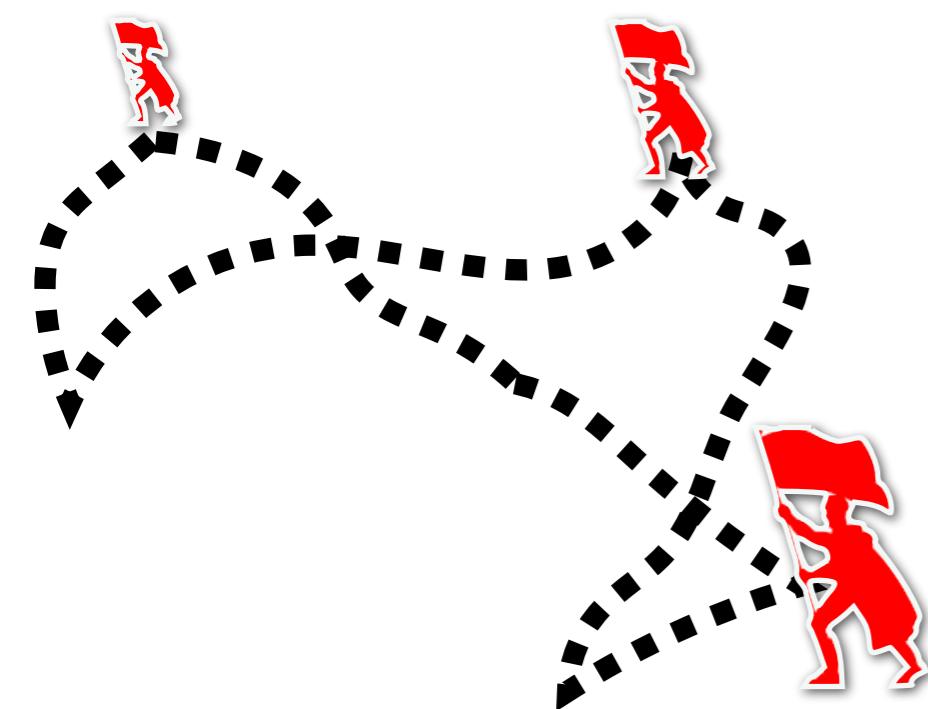
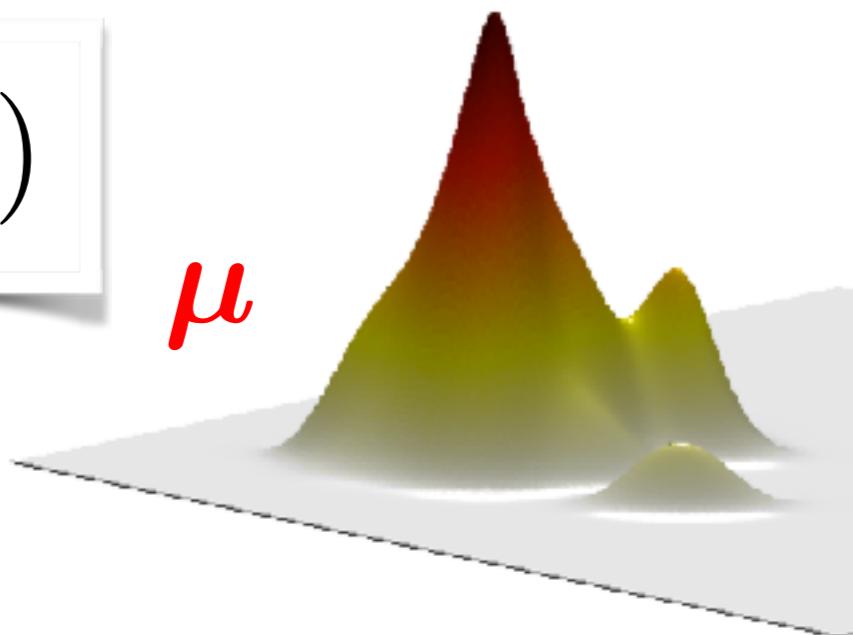
Mathematical Formalism

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

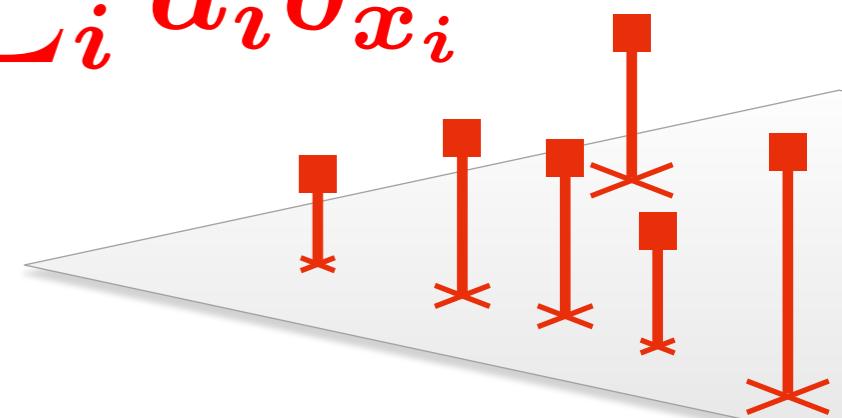


$$\mathcal{P}(\Omega)$$

$$\mu$$



$$\sum_i a_i \delta_{x_i}$$

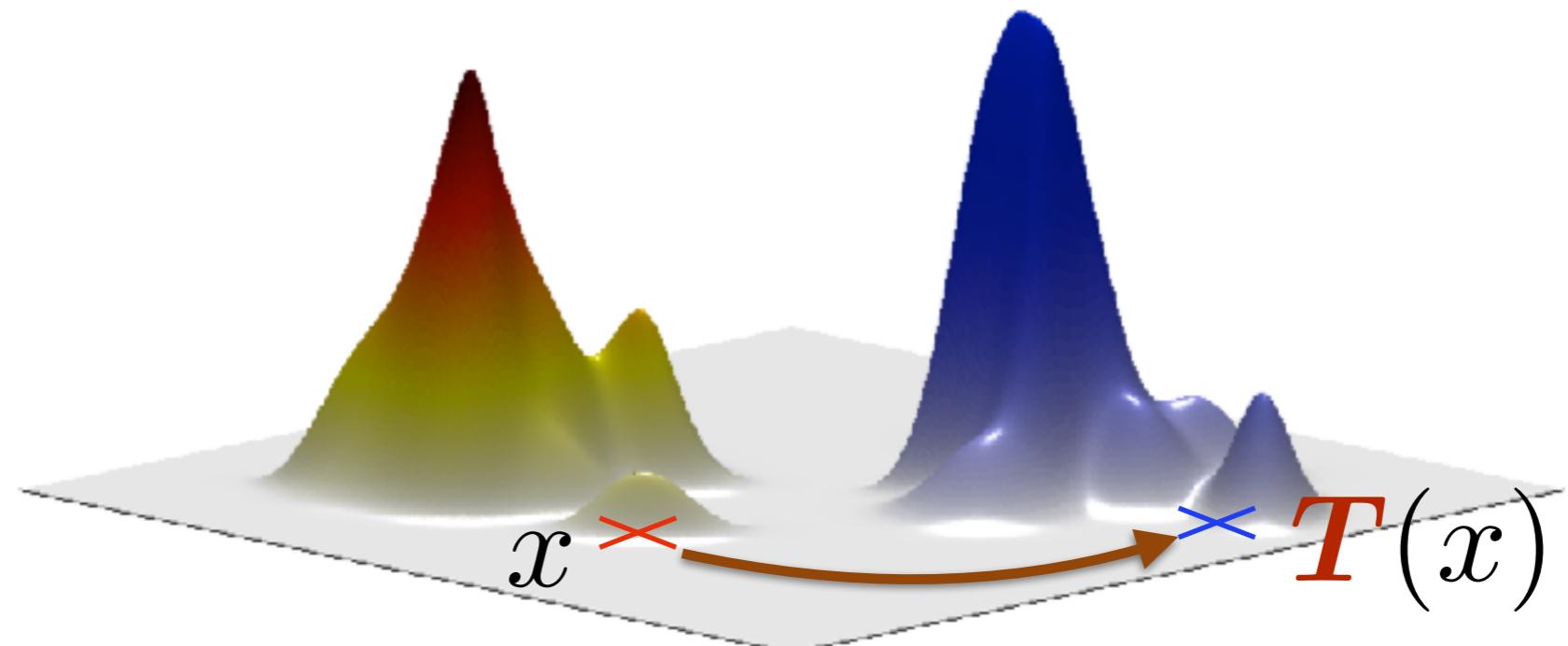


Monge Problem

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

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

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



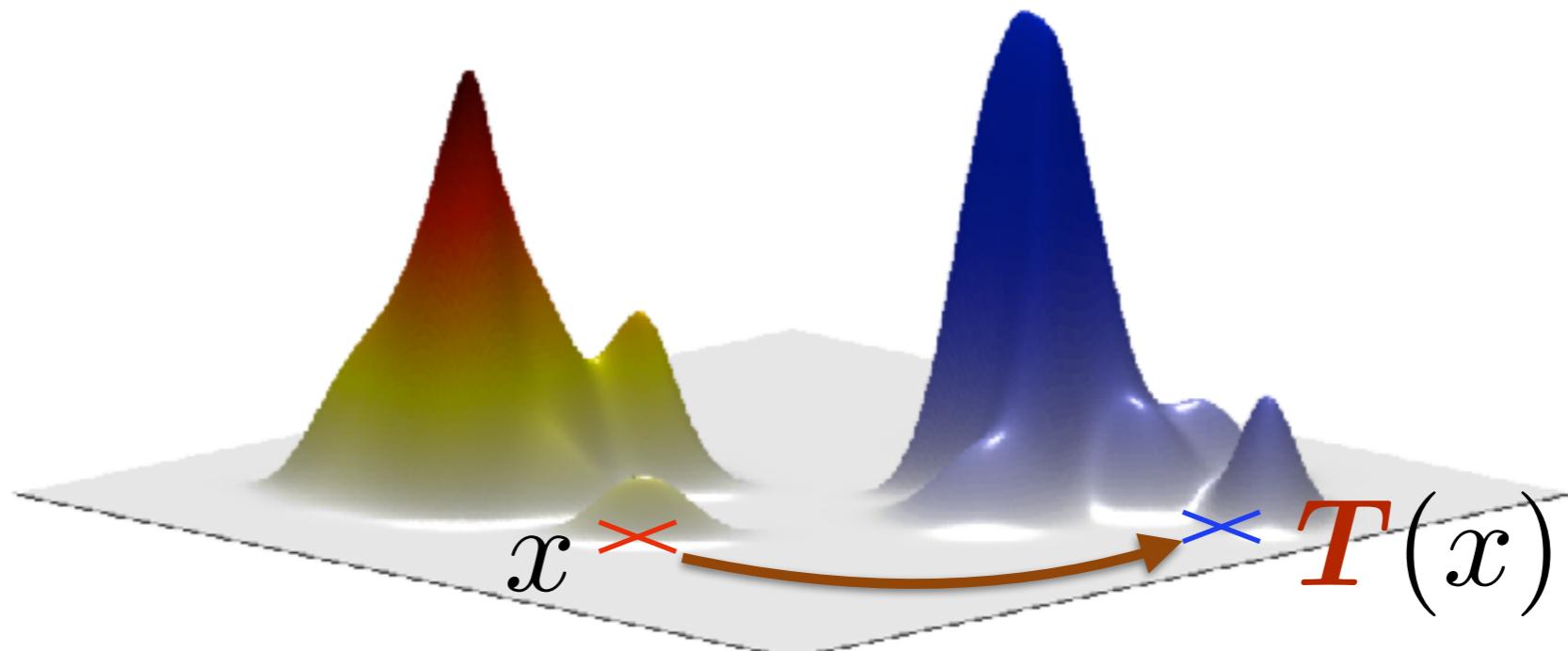
Monge Problem

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

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

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

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

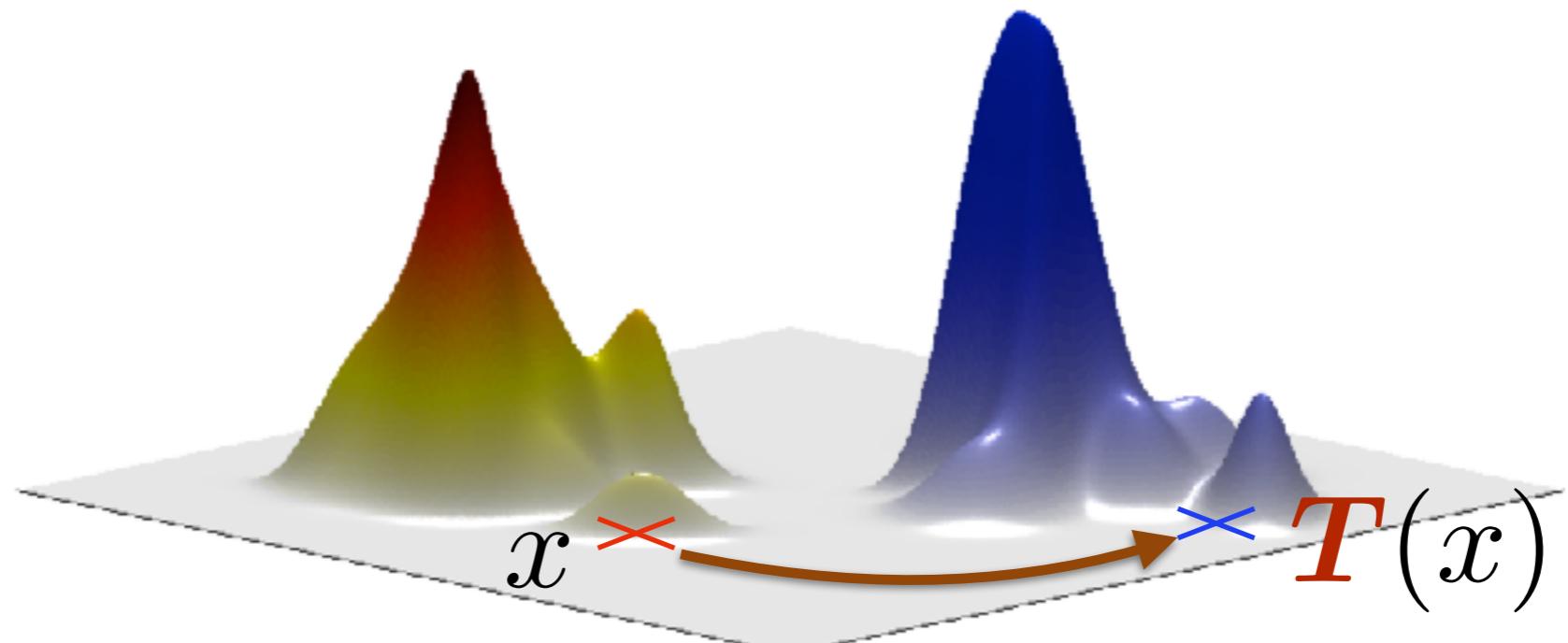


Monge Problem

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

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

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

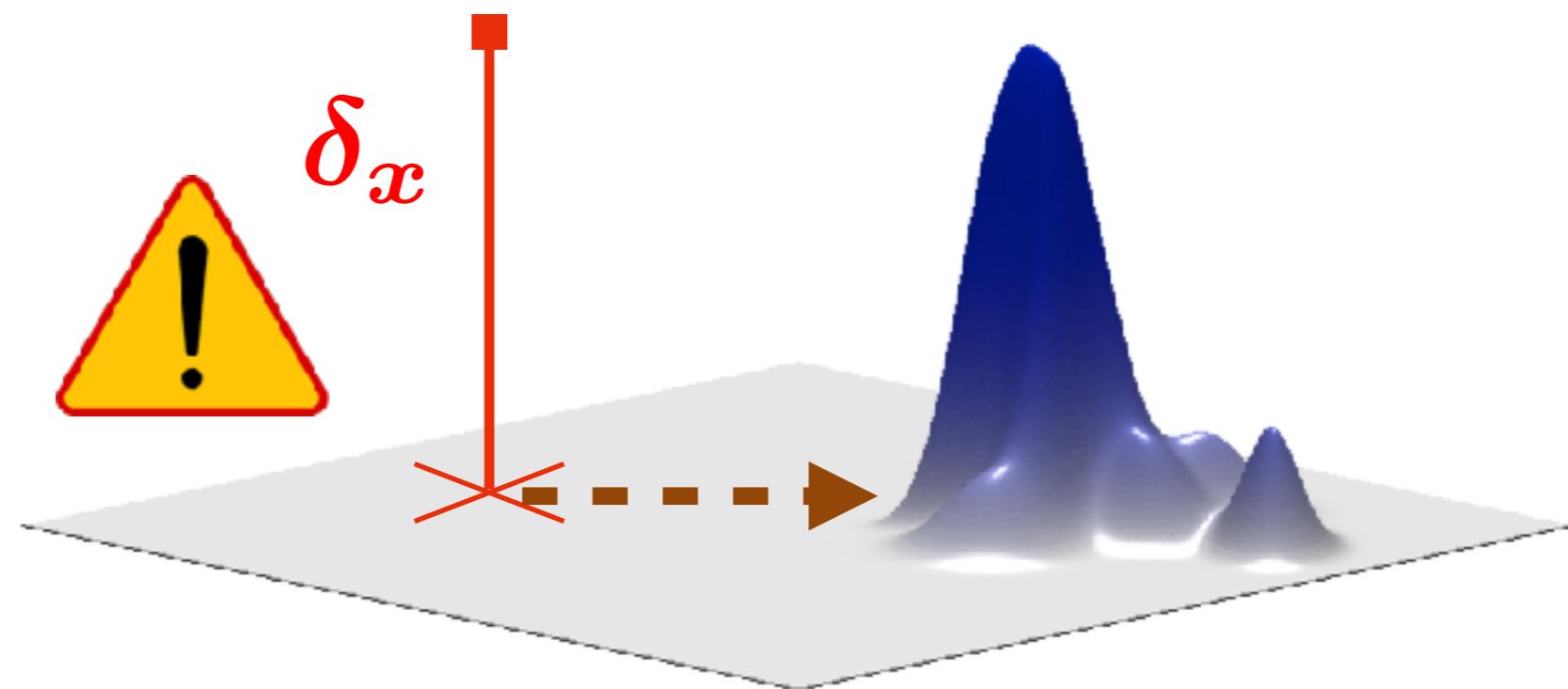


Monge Problem

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

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

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

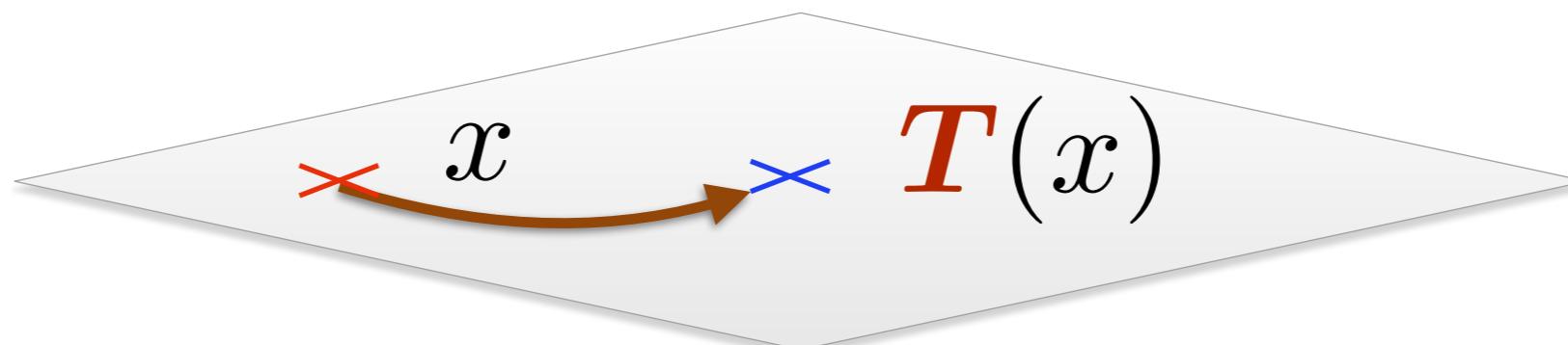


Kantorovich Relaxation

Instead of maps $\mathbf{\mathcal{T}} : \Omega \rightarrow \Omega$,

consider probabilistic maps,

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



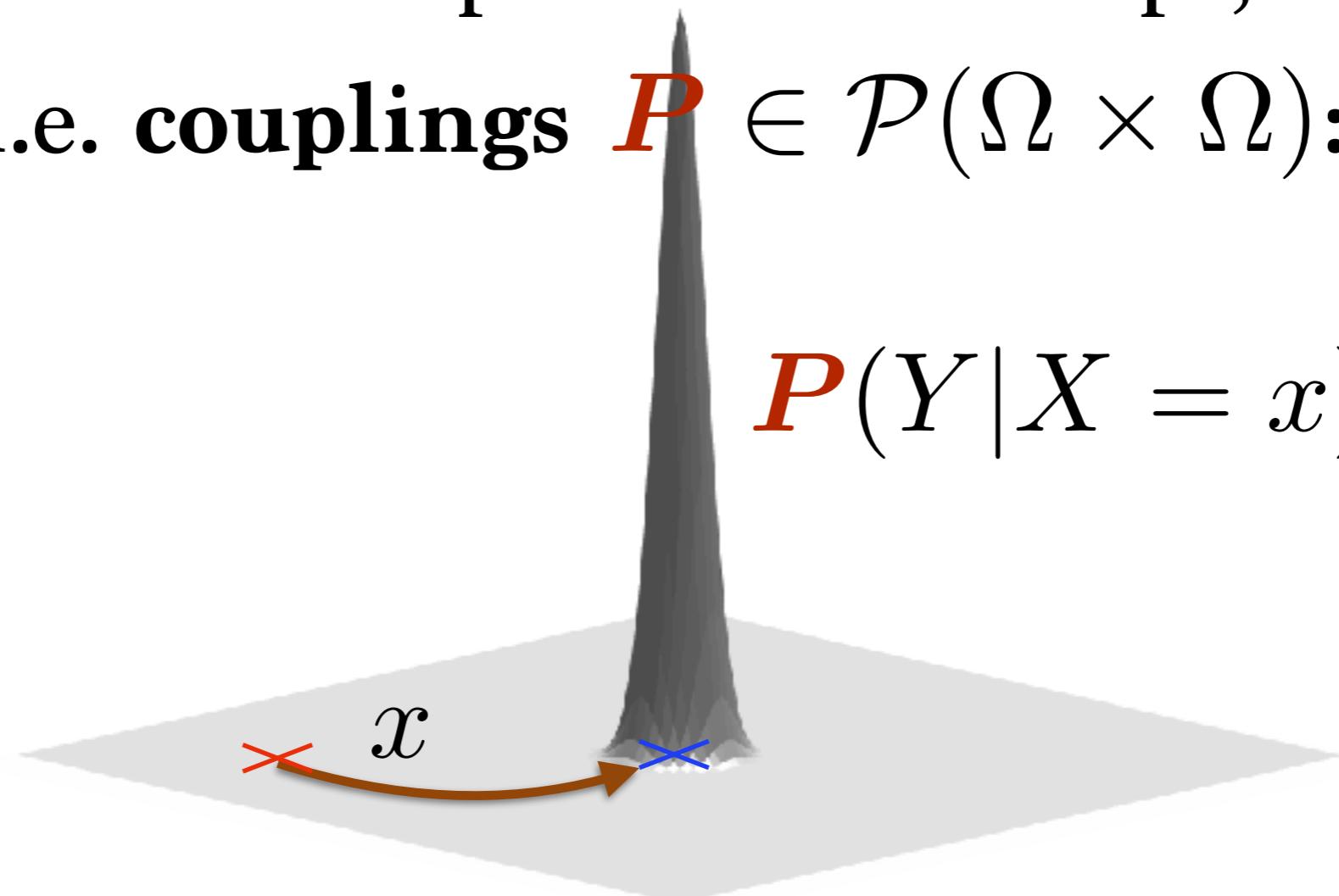
Kantorovich Relaxation

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

consider probabilistic maps,

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

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



Kantorovich Relaxation

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

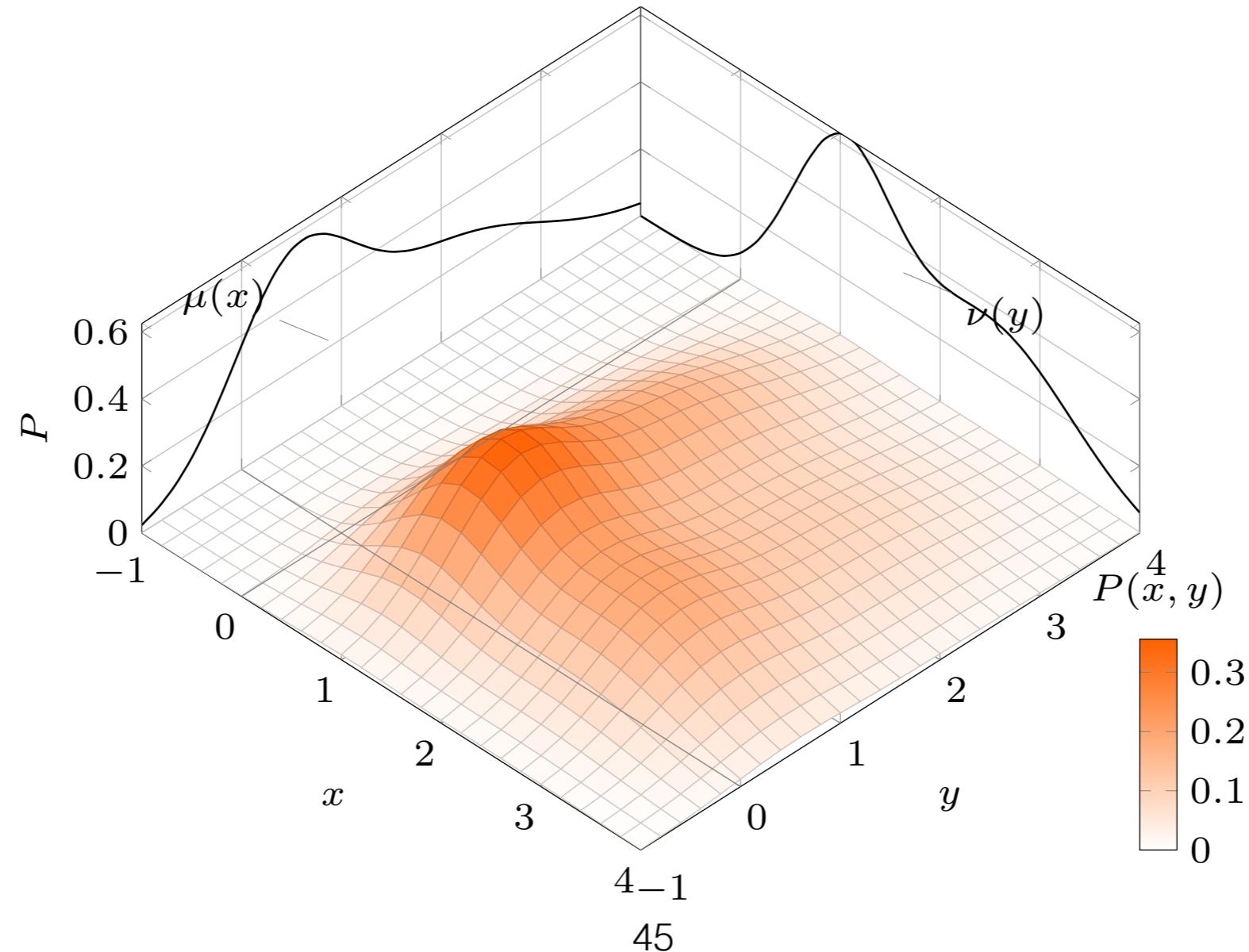
consider probabilistic maps,

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

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

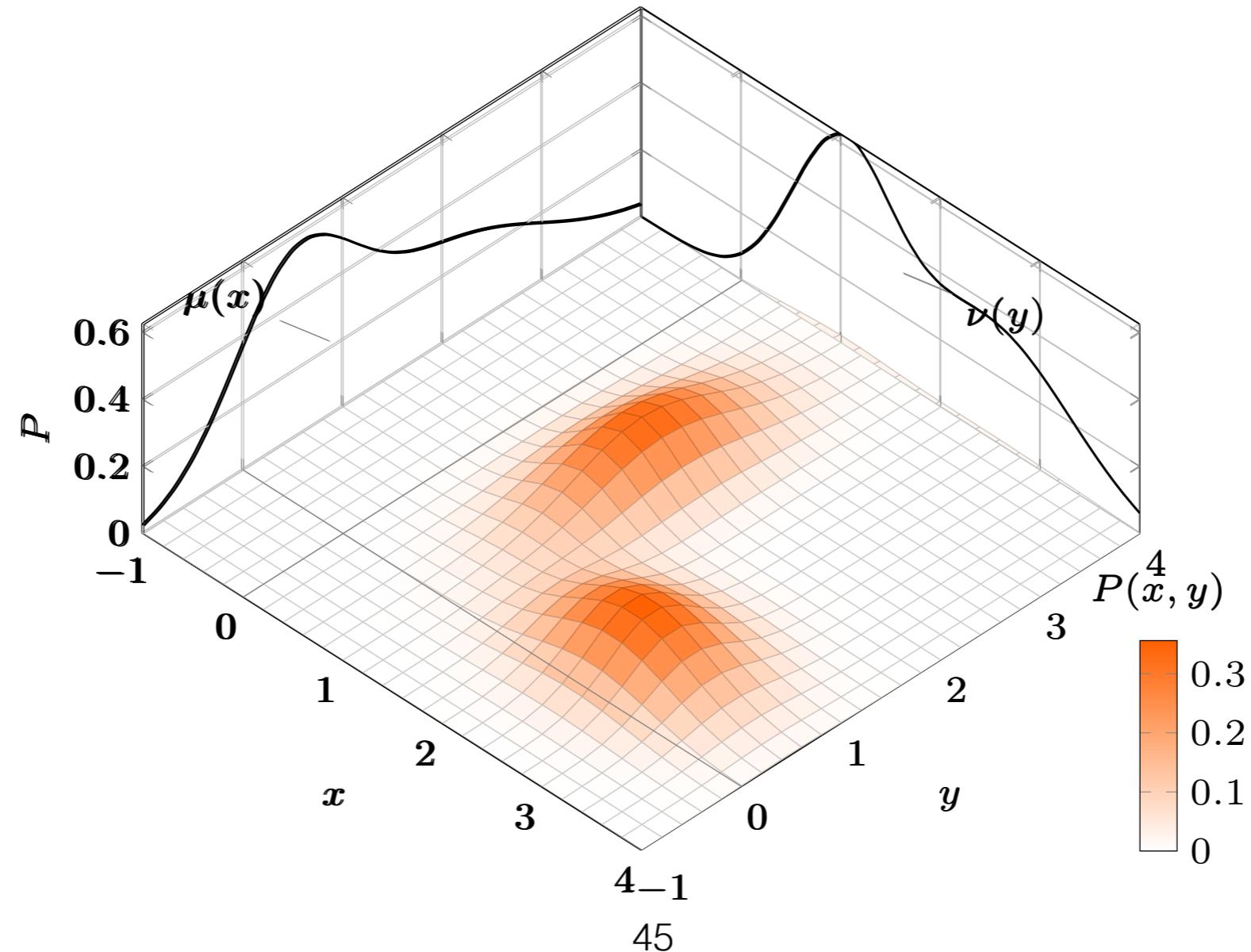
Kantorovich Relaxation

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



Kantorovich 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 \mathbf{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

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

PRIMAL

Kantorovich Problem

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

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

PRIMAL

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

DUAL

Kantorovich Problem

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

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

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

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

DUAL

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

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

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

Deriving Kantorovich Duality

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

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

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

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

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

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

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

DUAL

- 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

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

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

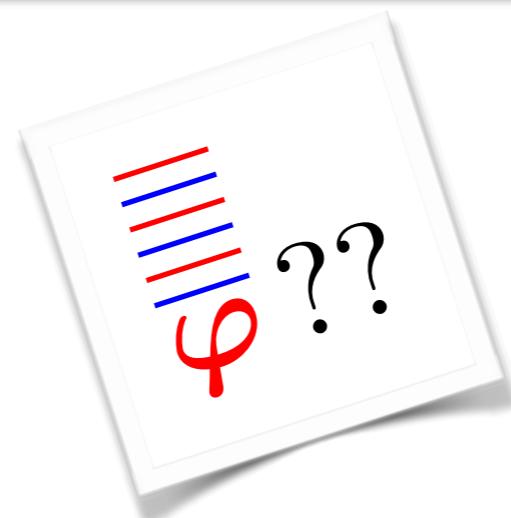
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\varphi} \int \overline{\varphi} d\boldsymbol{\mu} + \int \overline{\varphi} d\boldsymbol{\nu}.$$

D transforms

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

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

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



D transforms

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

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

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

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

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

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

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

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

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.

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.

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

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.

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

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

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

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

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

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

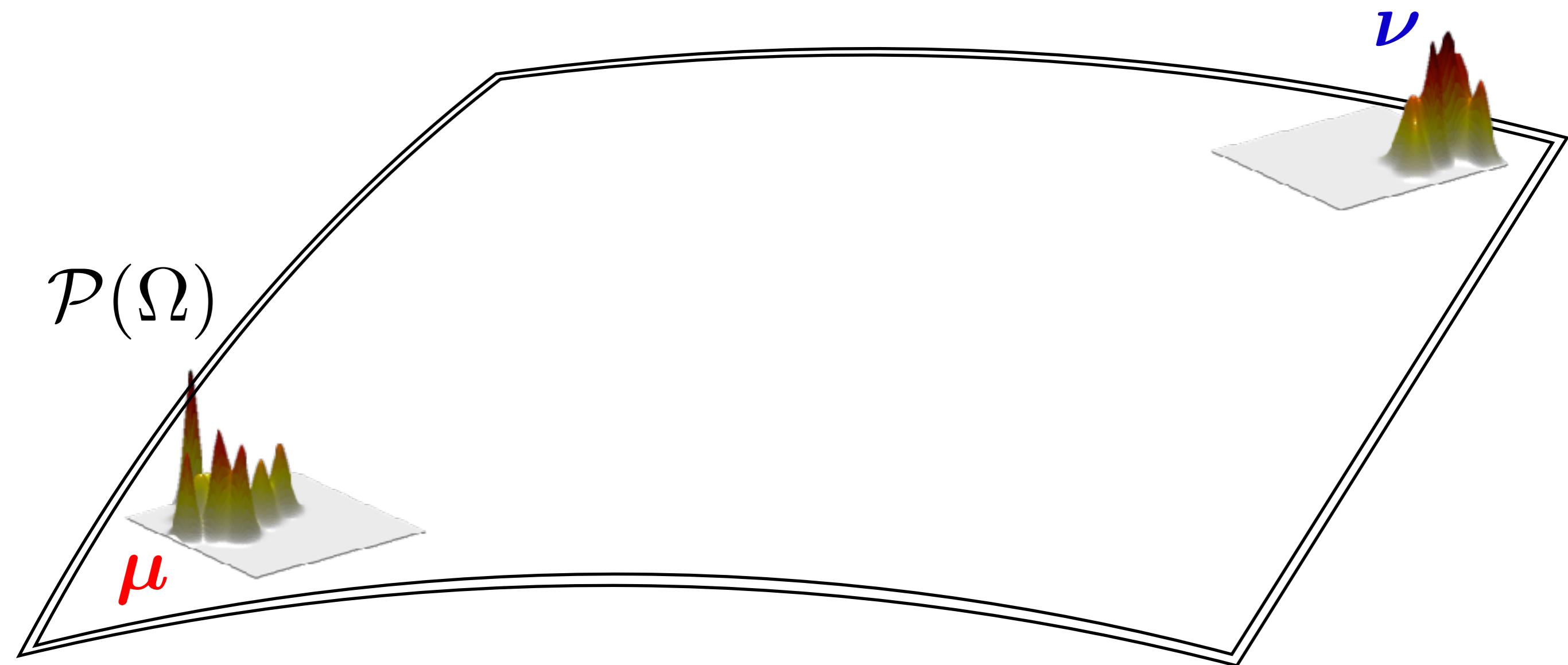
$$P^* := (\text{Id}, T^*)_{\sharp} \mu \in \Pi(\mu, \nu)$$

is also *optimal* for the Kantorovich problem.

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

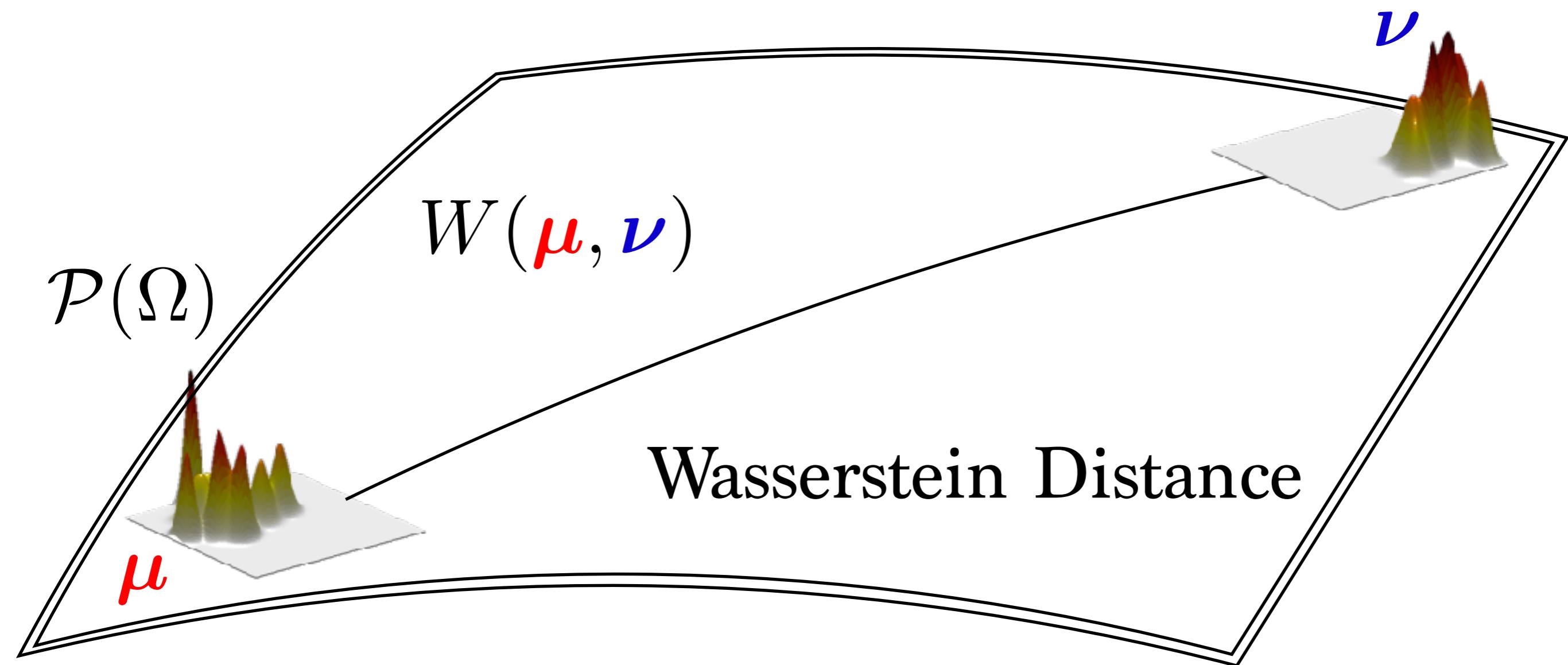
Optimal Transport Geometry

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



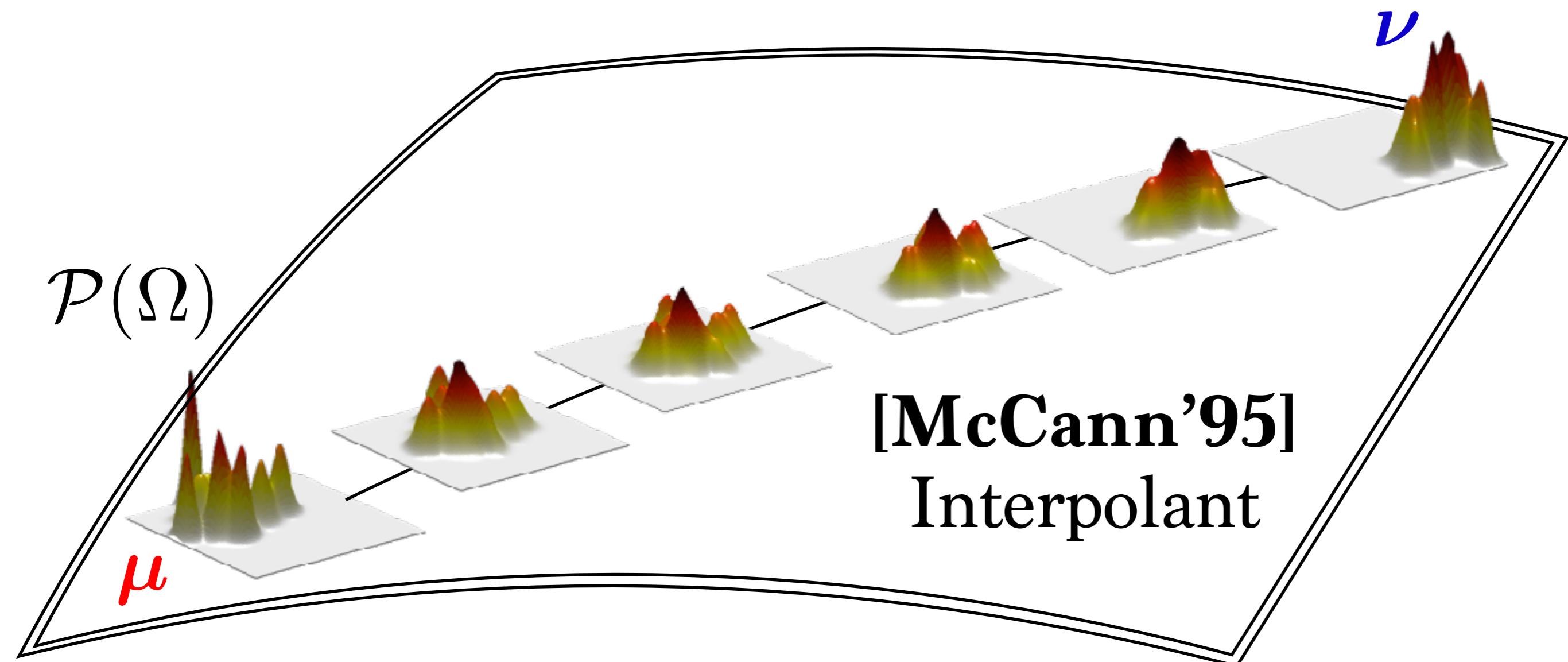
Optimal Transport Geometry

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



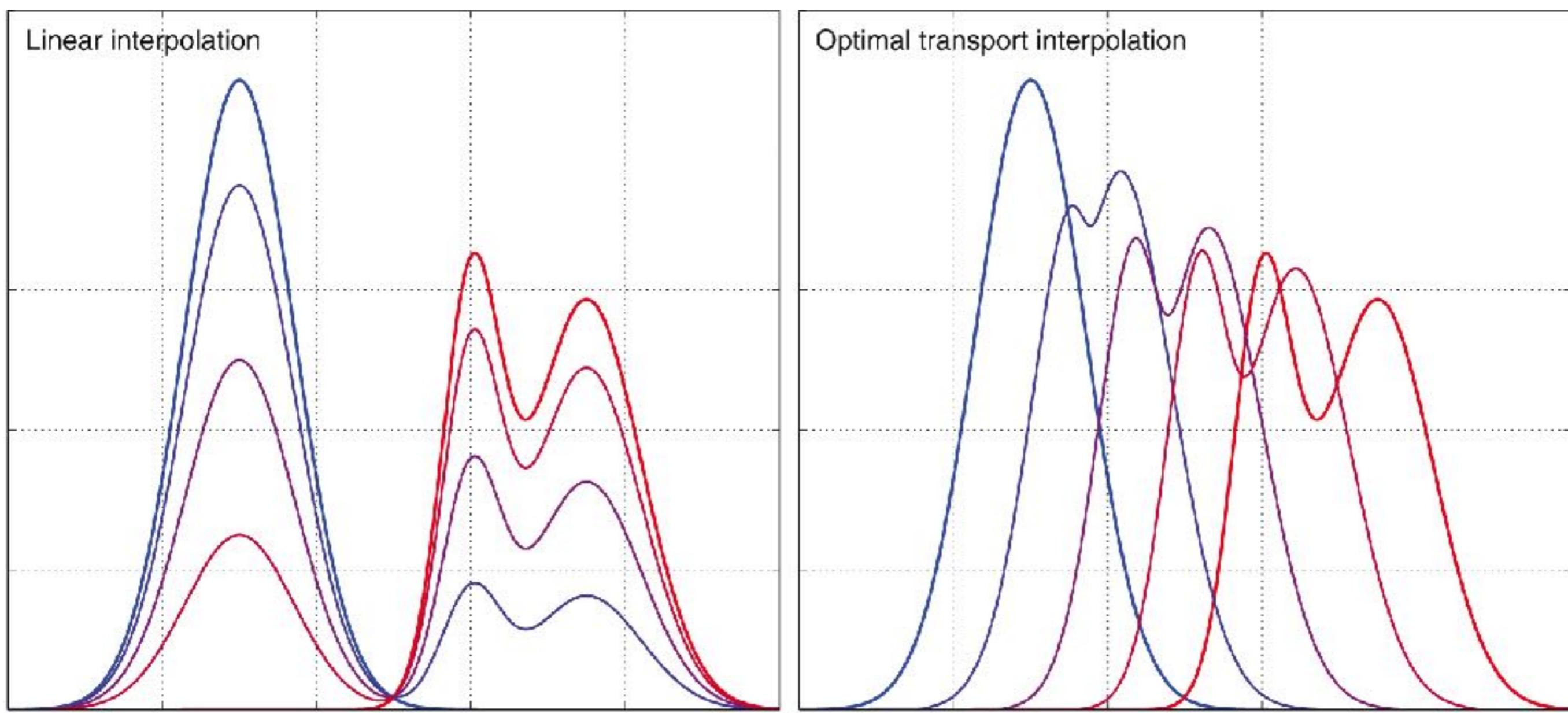
Optimal Transport Geometry

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



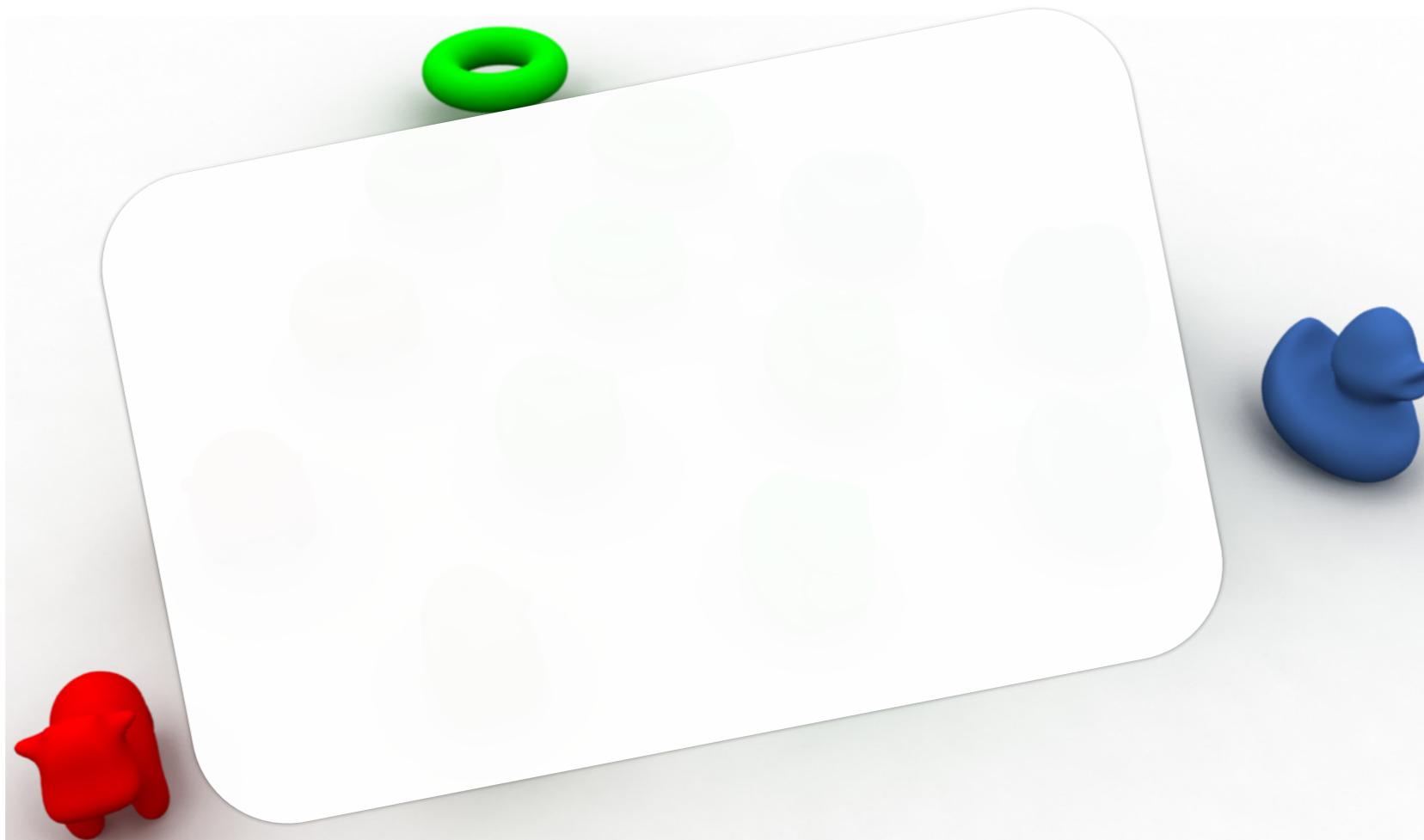
Optimal Transport Geometry

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



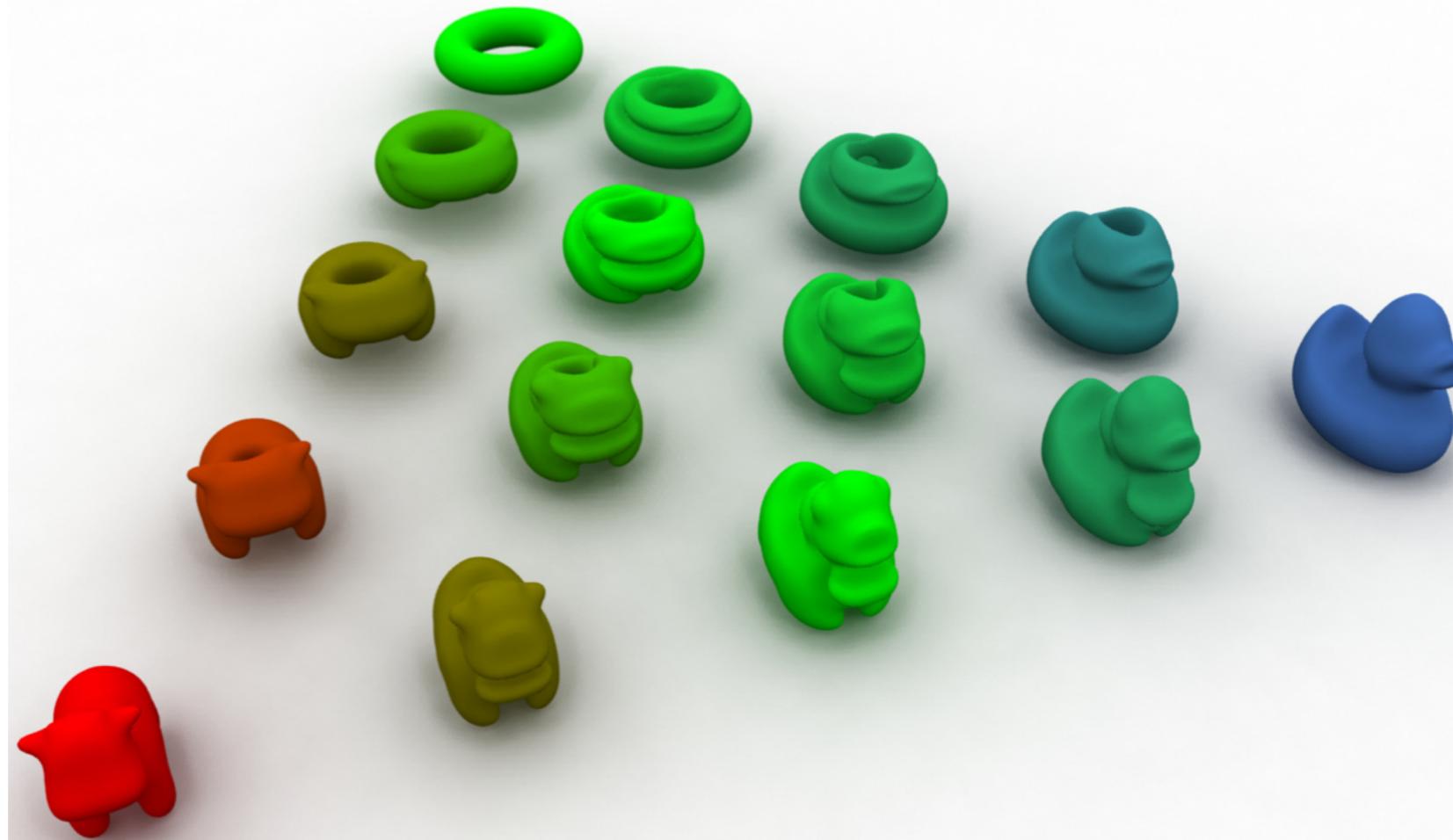
Optimal Transport Geometry

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



Optimal Transport Geometry

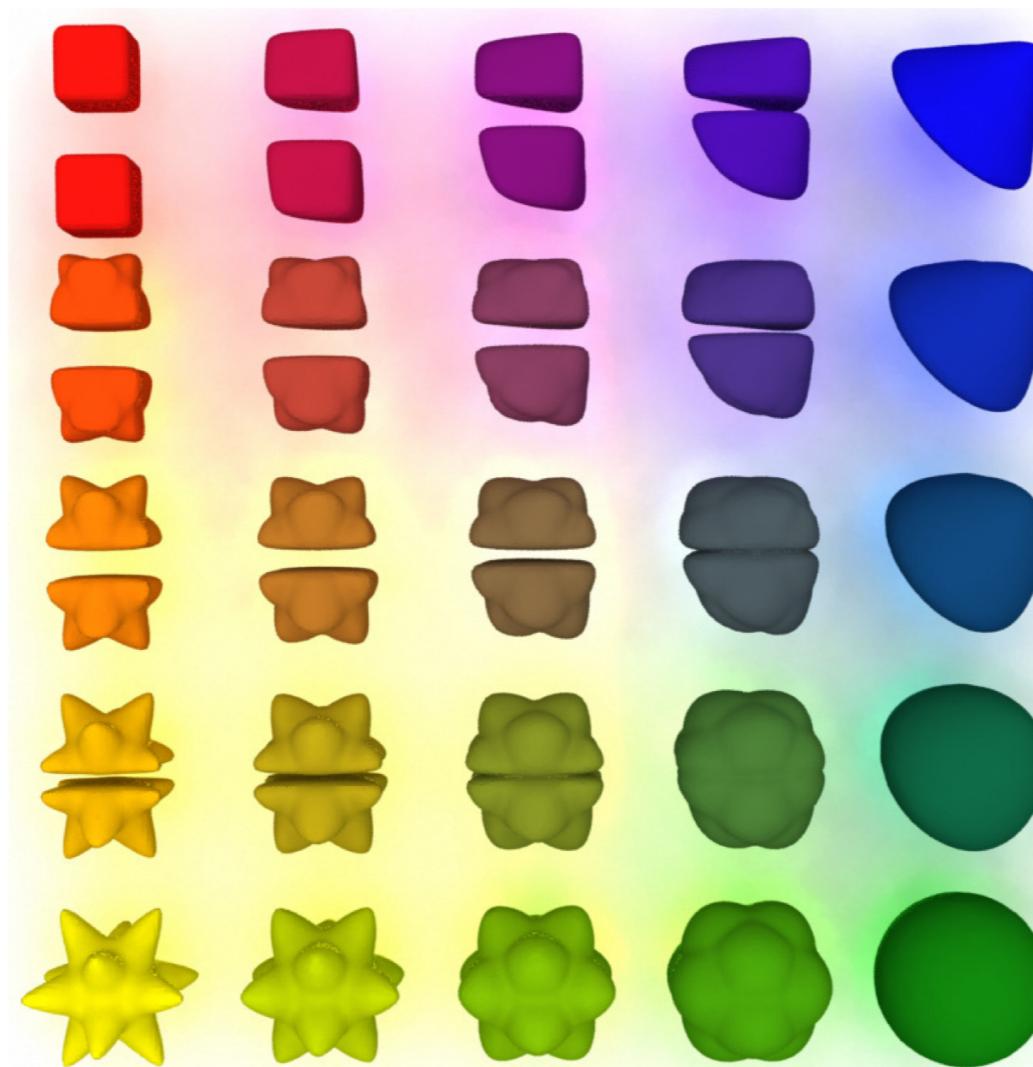
Very different geometry than standard information divergences (KL, Euclidean)



[SDPC..'15]

Optimal Transport Geometry

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



[SDPC..'15]

Computational OT

Up to 2010: OT solvers $W_p(\mu, \nu) = ?$

Goal now: use OT as a **loss or fidelity term**

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

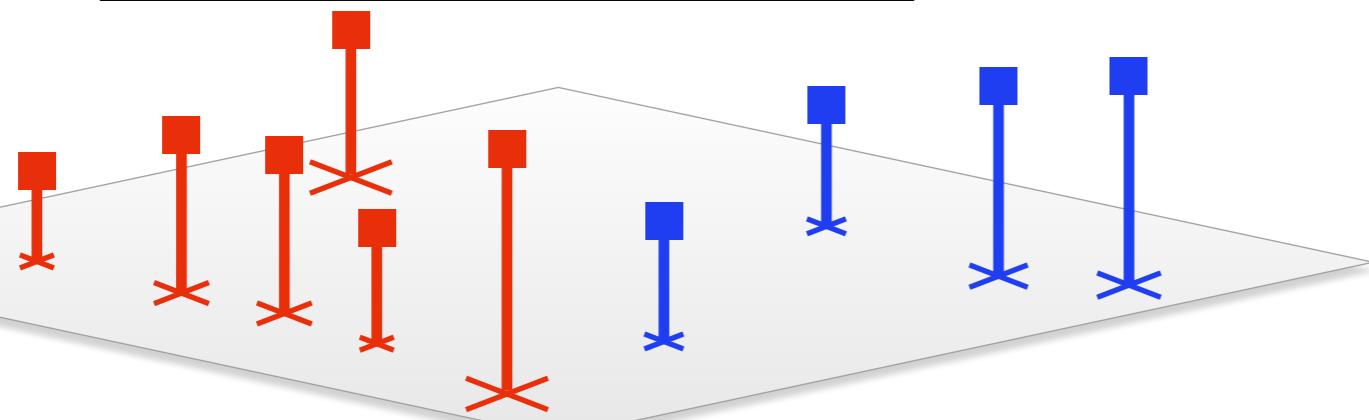
$\nabla_{\mu} W_p(\mu, \nu_1) = ?$

2. How to compute OT

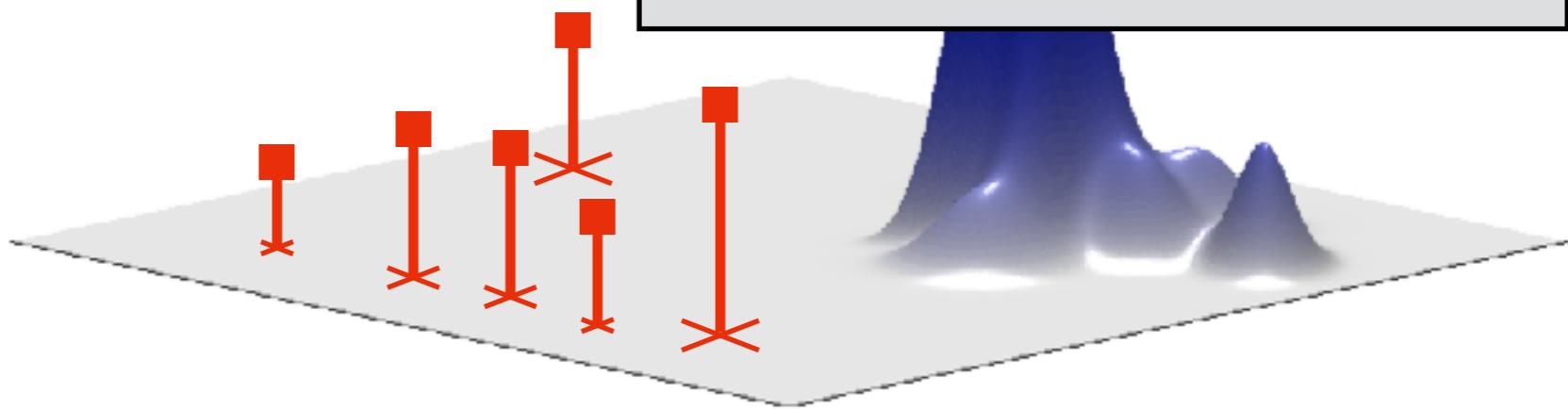
- Typology: discrete/continuous problems
- Easy cases, zoo of solvers
- Entropic regularization
- Differentiability of the W distance

How can we compute OT ?

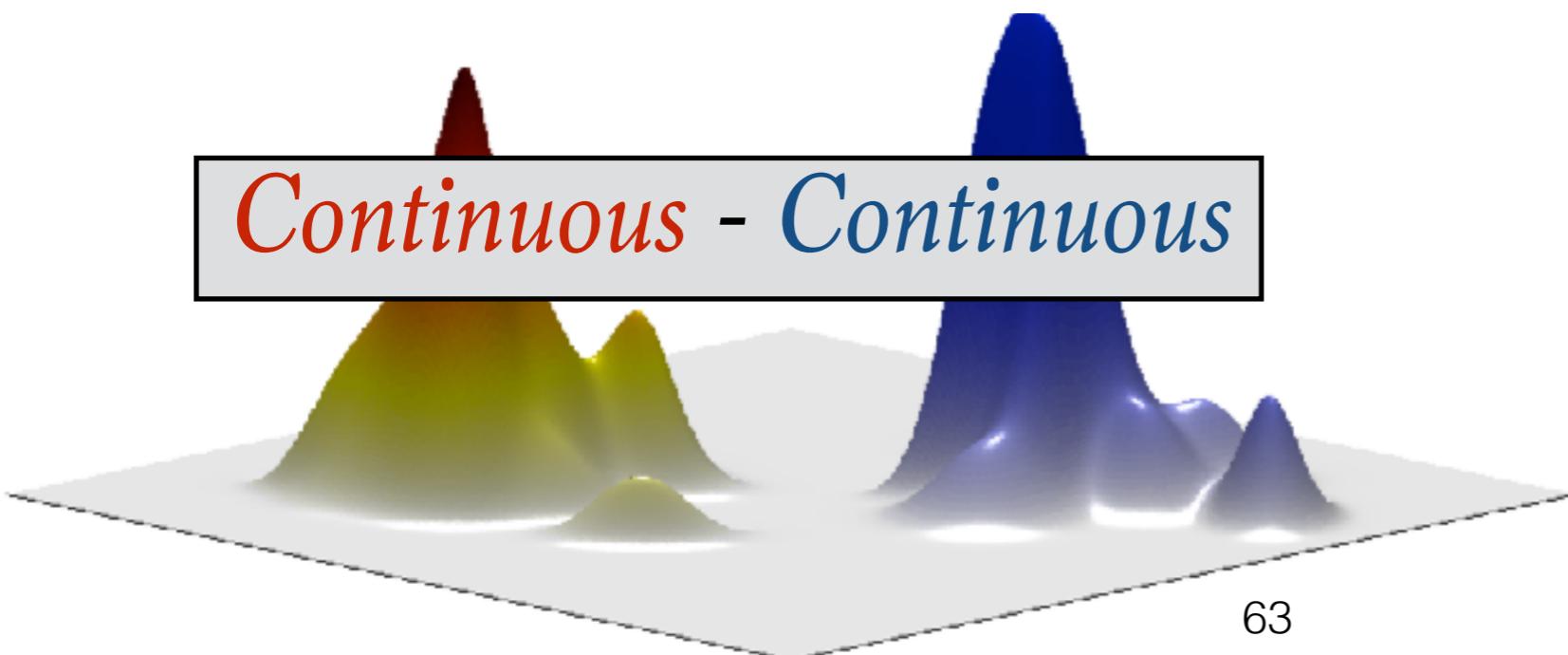
Discrete - Discrete



Discrete - Continuous



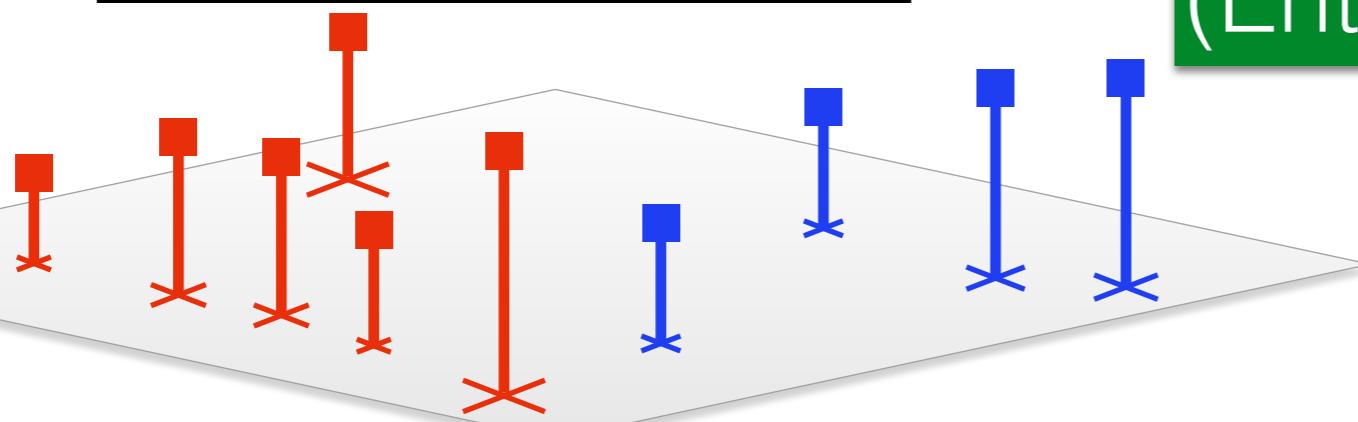
Continuous - Continuous



How can we compute OT ?

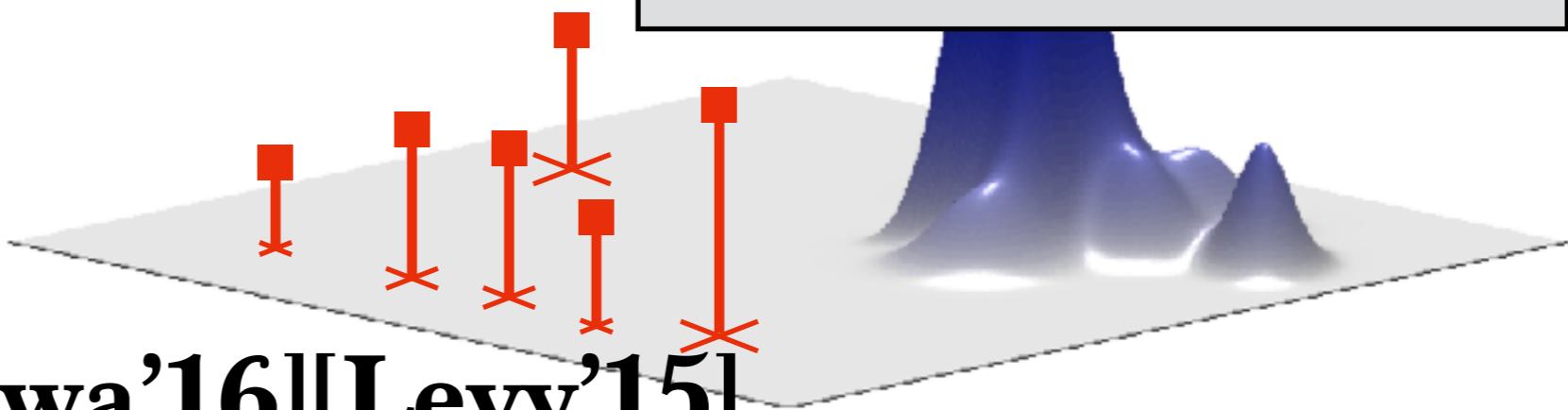
Discrete - Discrete

Network flow solvers
(Entropic) regularization



low dim.

Discrete - Continuous



[Mérigot'11][Kitagawa'16][Levy'15]

Continuous - Continuous

Stochastic
Optimization

PDE's

[Genevay'16]

[Benamou'98]

Easy (1): Univariate Measures

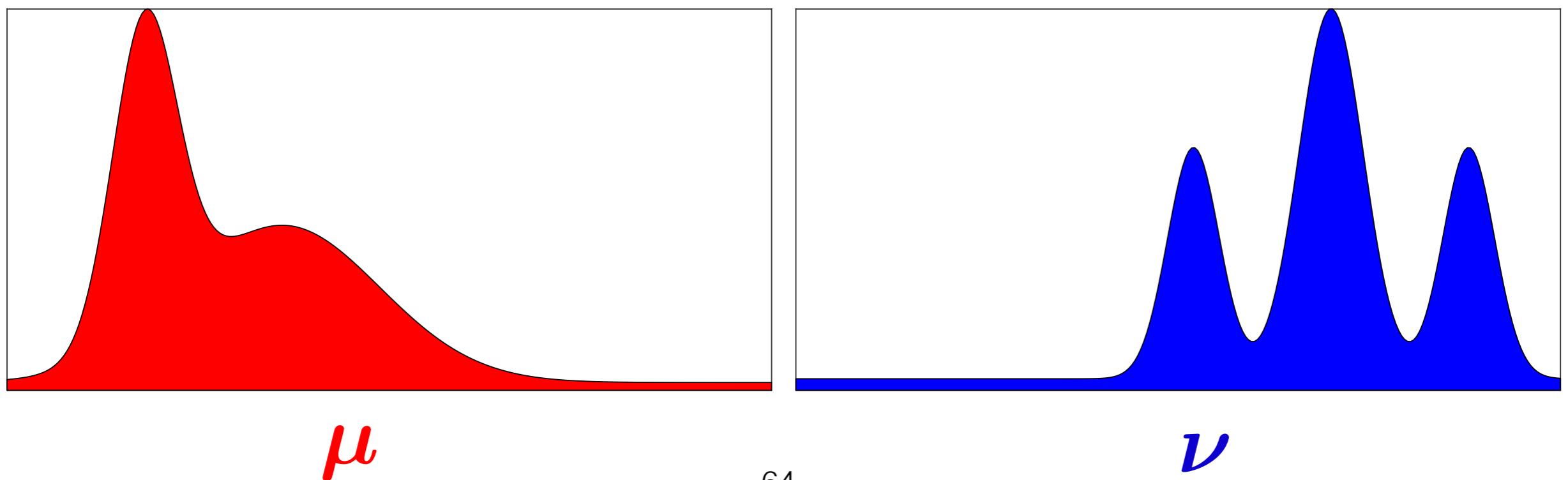
Remark. If $\Omega = \mathbb{R}$, $\textcolor{red}{c}(x, y) = \textcolor{green}{c}(|x - y|)$,
 $\textcolor{green}{c}$ convex, $F_{\boldsymbol{\mu}}^{-1}, F_{\boldsymbol{\nu}}^{-1}$ quantile functions,

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

Easy (1): Univariate Measures

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

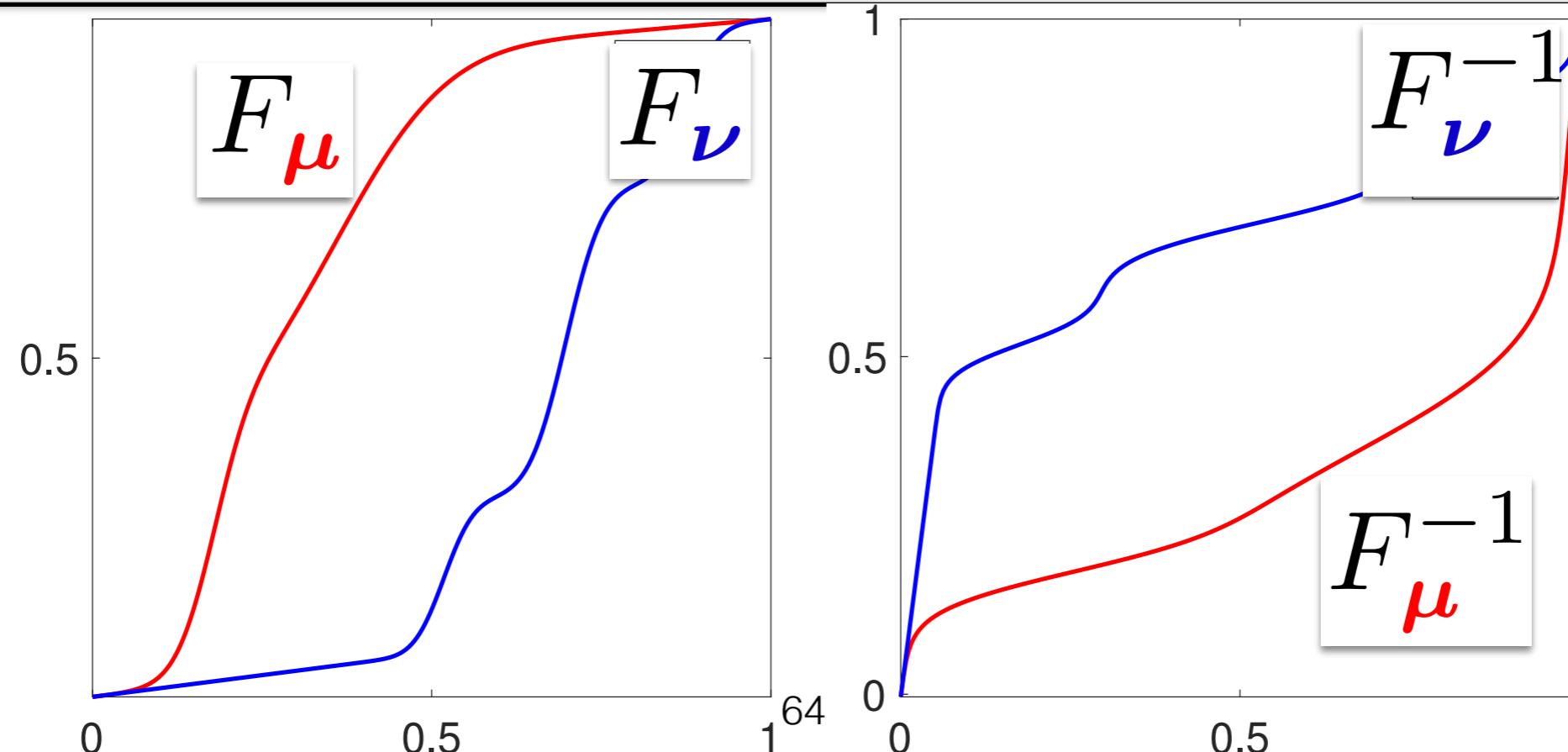
$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \int_0^1 \textcolor{green}{c}(|F_{\boldsymbol{\mu}}^{-1}(x) - F_{\boldsymbol{\nu}}^{-1}(x)|) dx$$



Easy (1): Univariate Measures

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

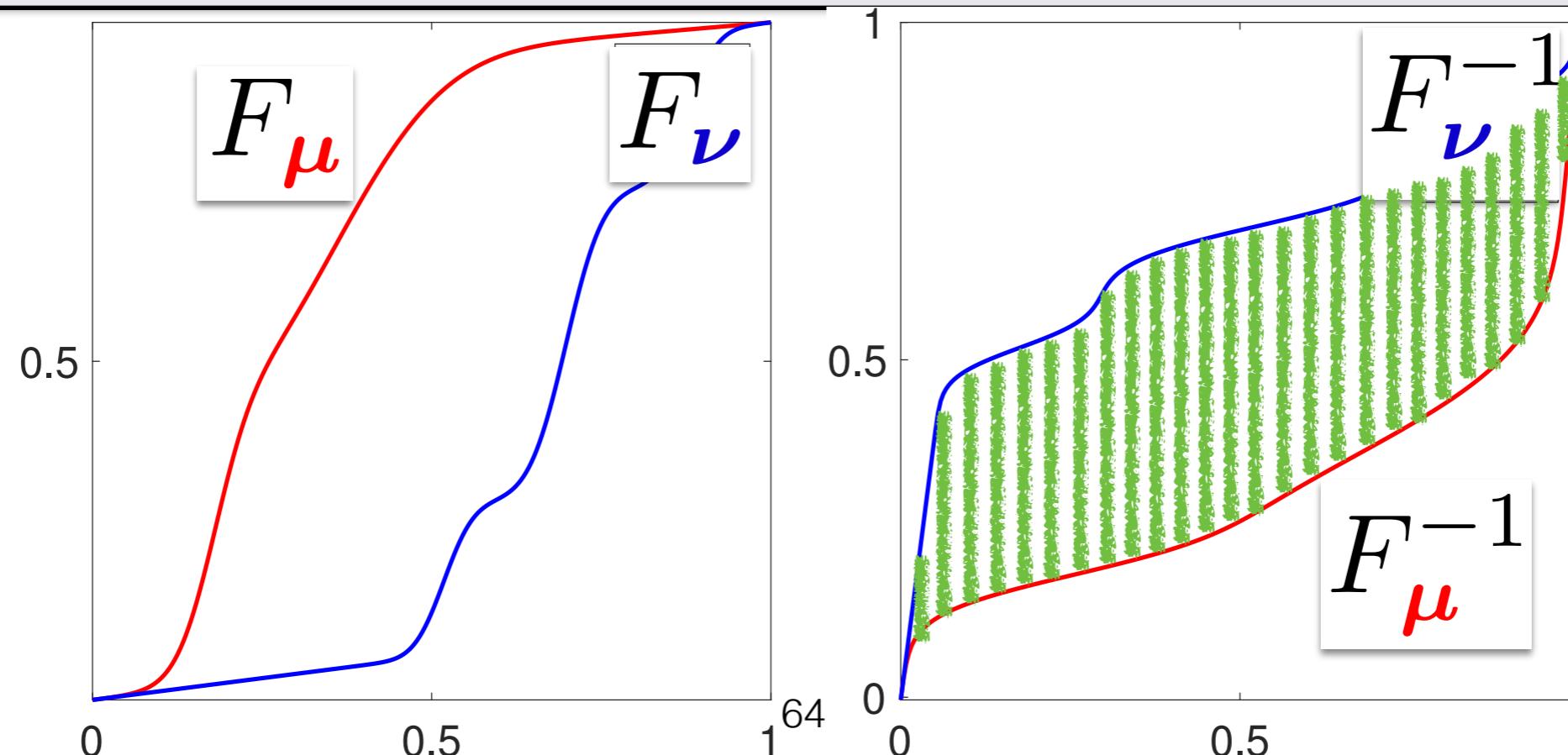
$$W(\mu, \nu) = \int_0^1 \textcolor{green}{c}(|F_{\mu}^{-1}(x) - F_{\nu}^{-1}(x)|) dx$$



Easy (1): 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,

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



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

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

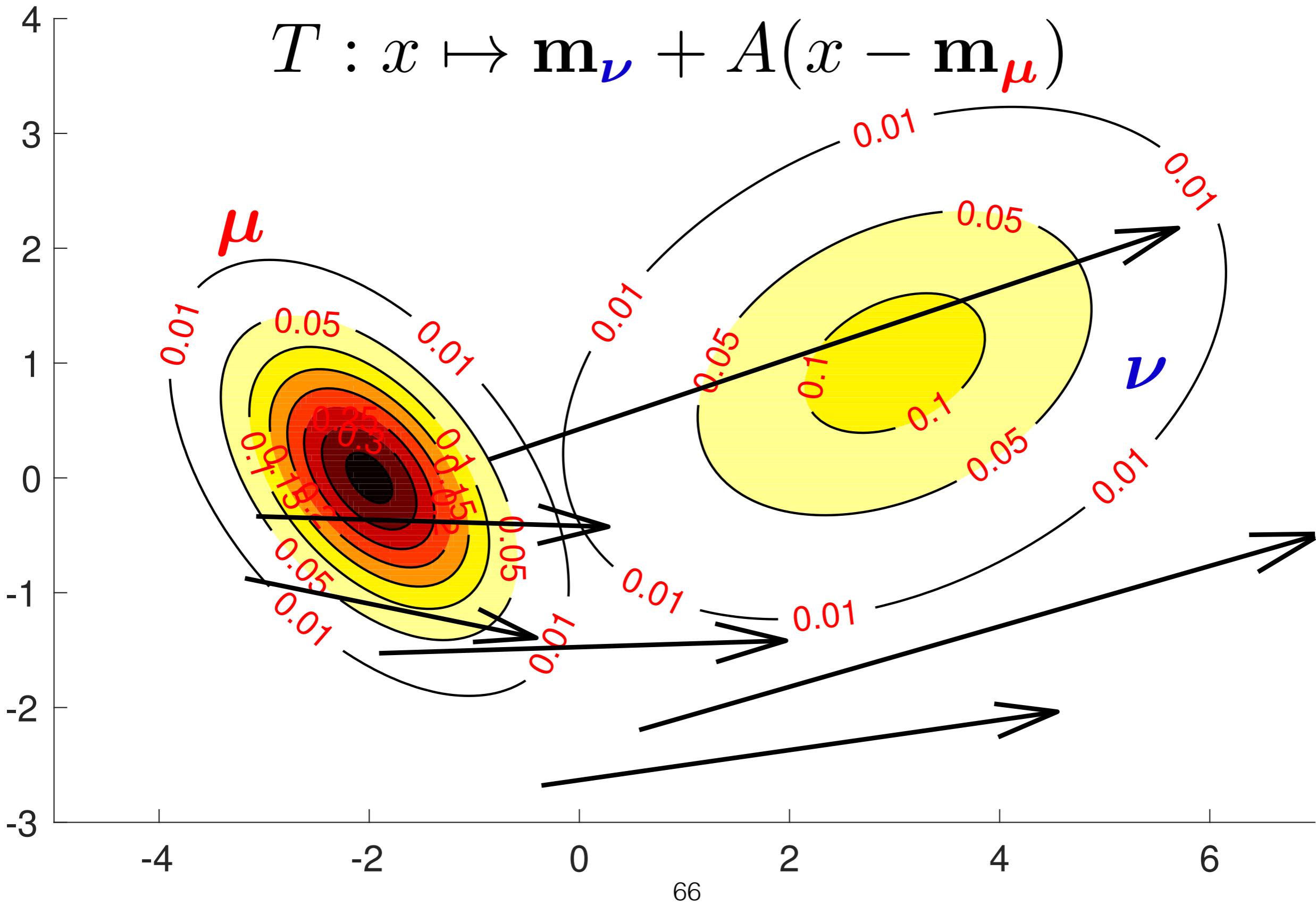
where B is the Bures metric

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

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

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

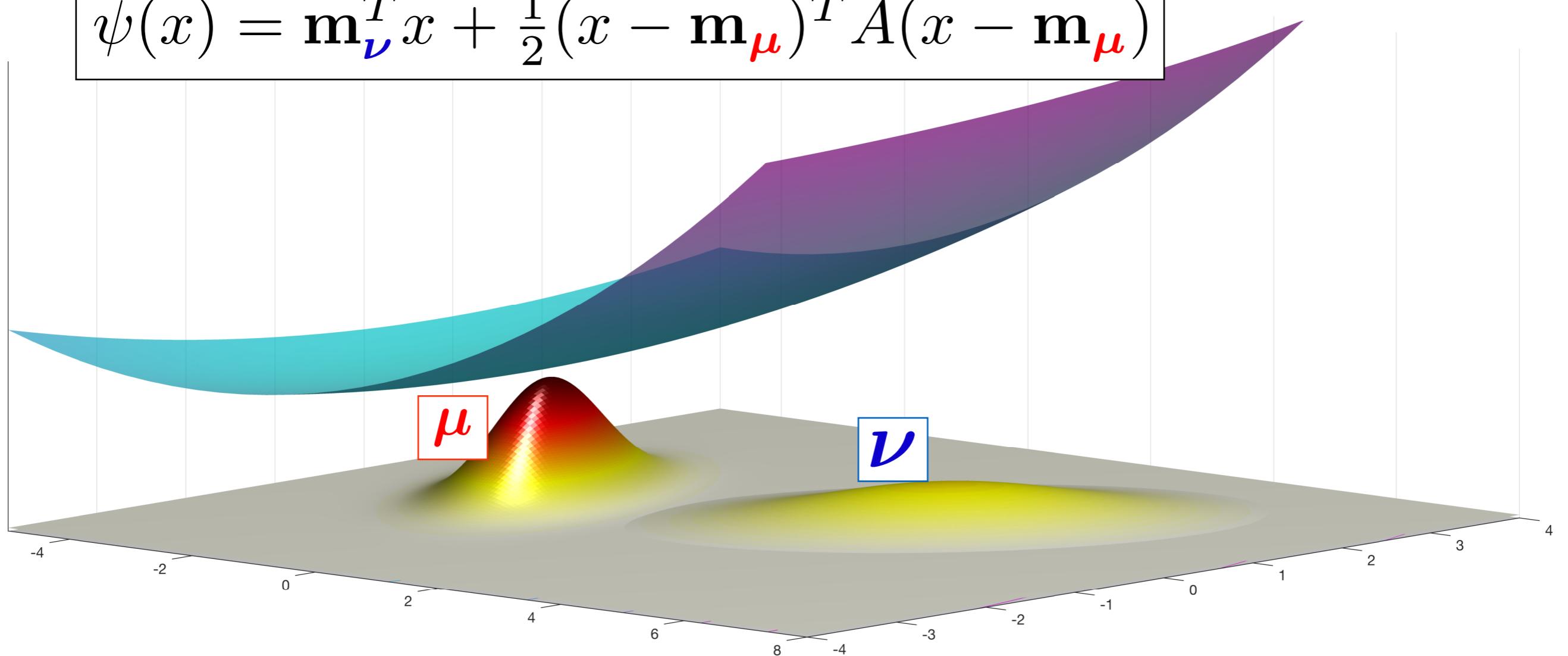
Easy (2): Gaussian Measures



Easy (2): Gaussian Measures

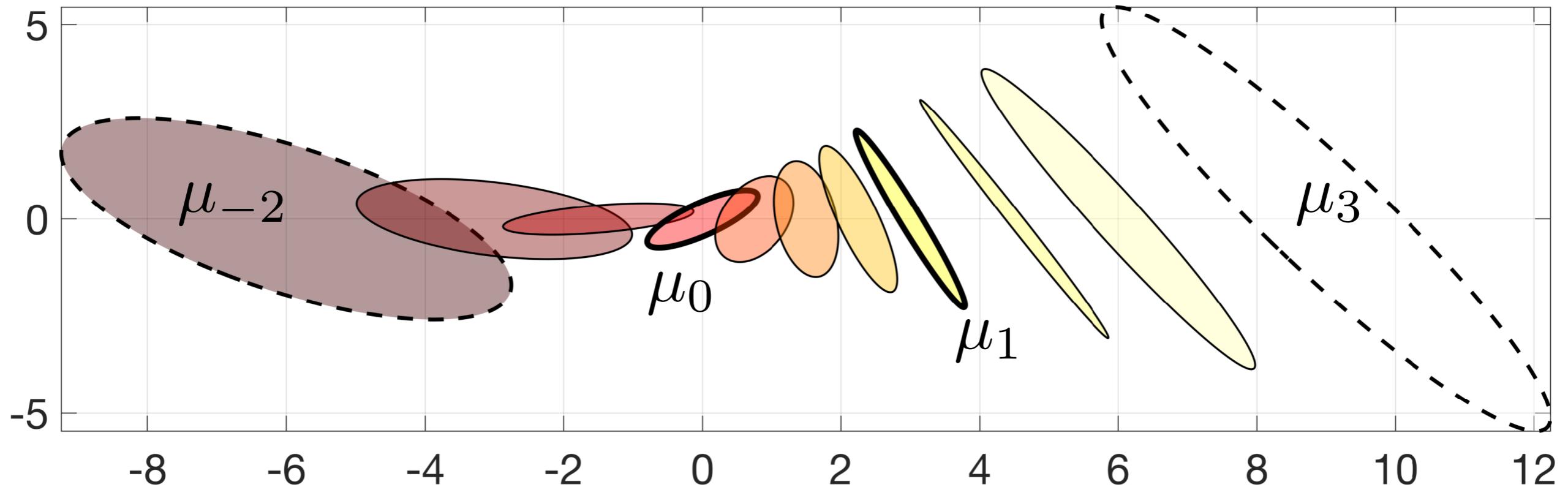
$$T = \nabla \psi : x \mapsto \mathbf{m}_{\nu} + A(x - \mathbf{m}_{\mu})$$

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



Easy (2): Gaussian Measures

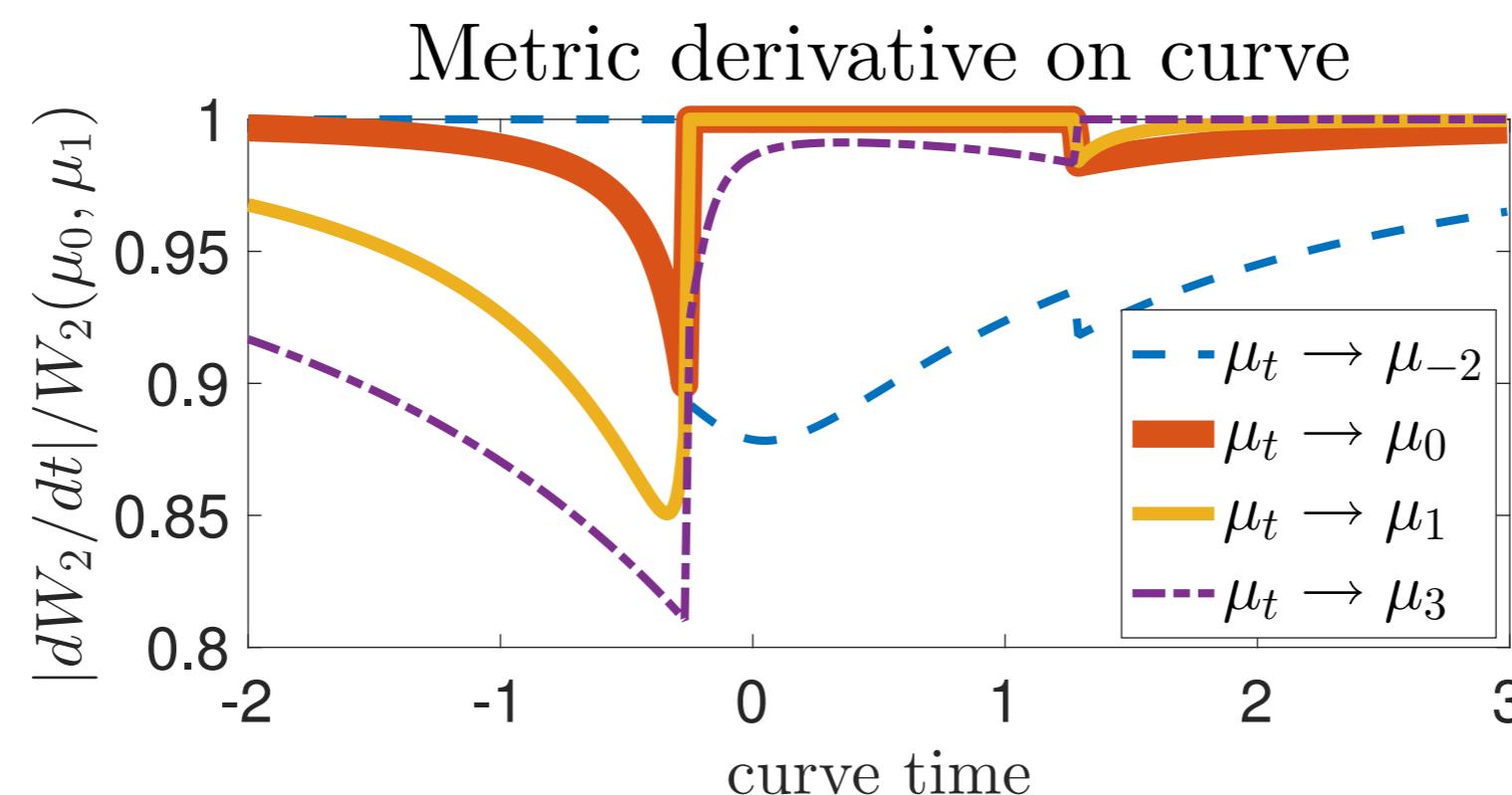
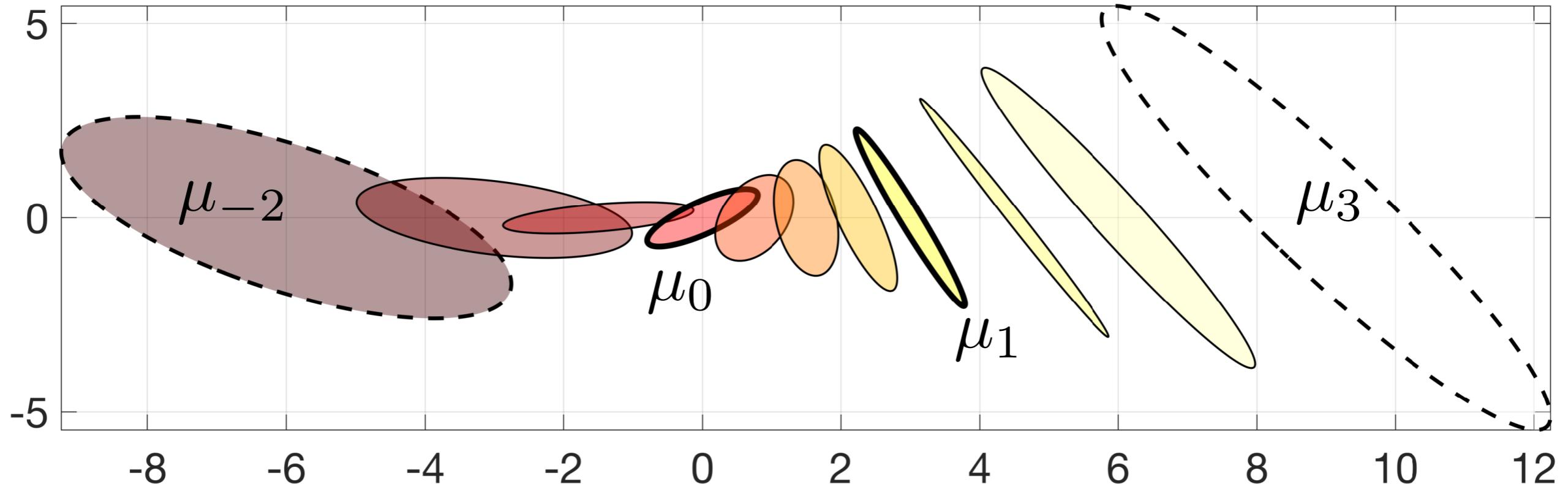
W_2 geodesic $(\mu_t)_t$ from μ_0 to μ_1 ($t \in [0, 1]$) and extrapolation



$$\Sigma_{\textcolor{green}{t}} = ((1 - t) I + tA) \Sigma_{\textcolor{red}{\mu}} ((1 - t) I + tA)$$

Easy (2): Gaussian Measures

W_2 geodesic $(\mu_t)_t$ from μ_0 to μ_1 ($t \in [0, 1]$) and extrapolation



Easy (3): Elliptical Distributions

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

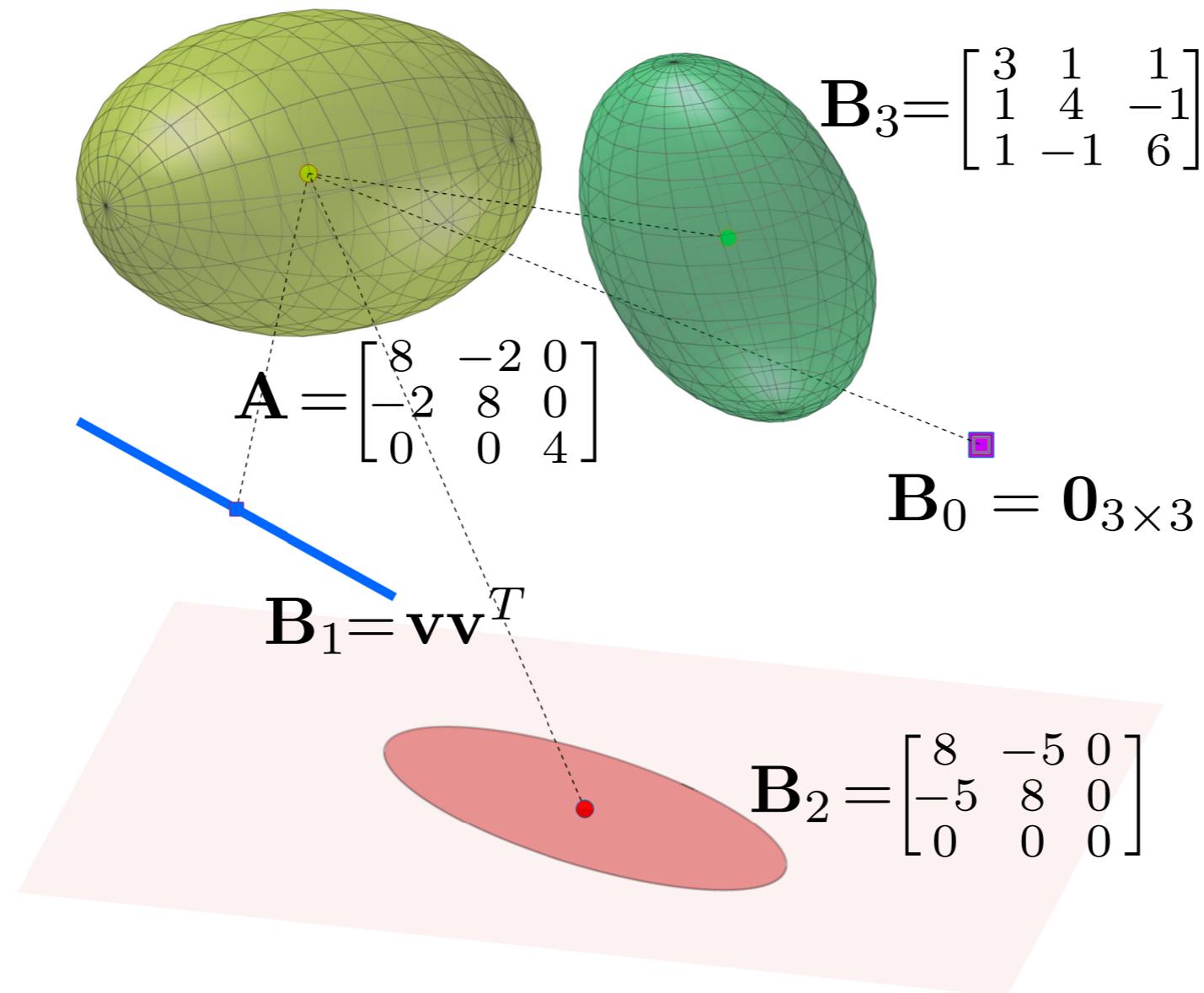
[Gelbrich'92] shows that the linear map T is also **optimal** for elliptically contoured distributions, *i.e.* distributions whose MGF are

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

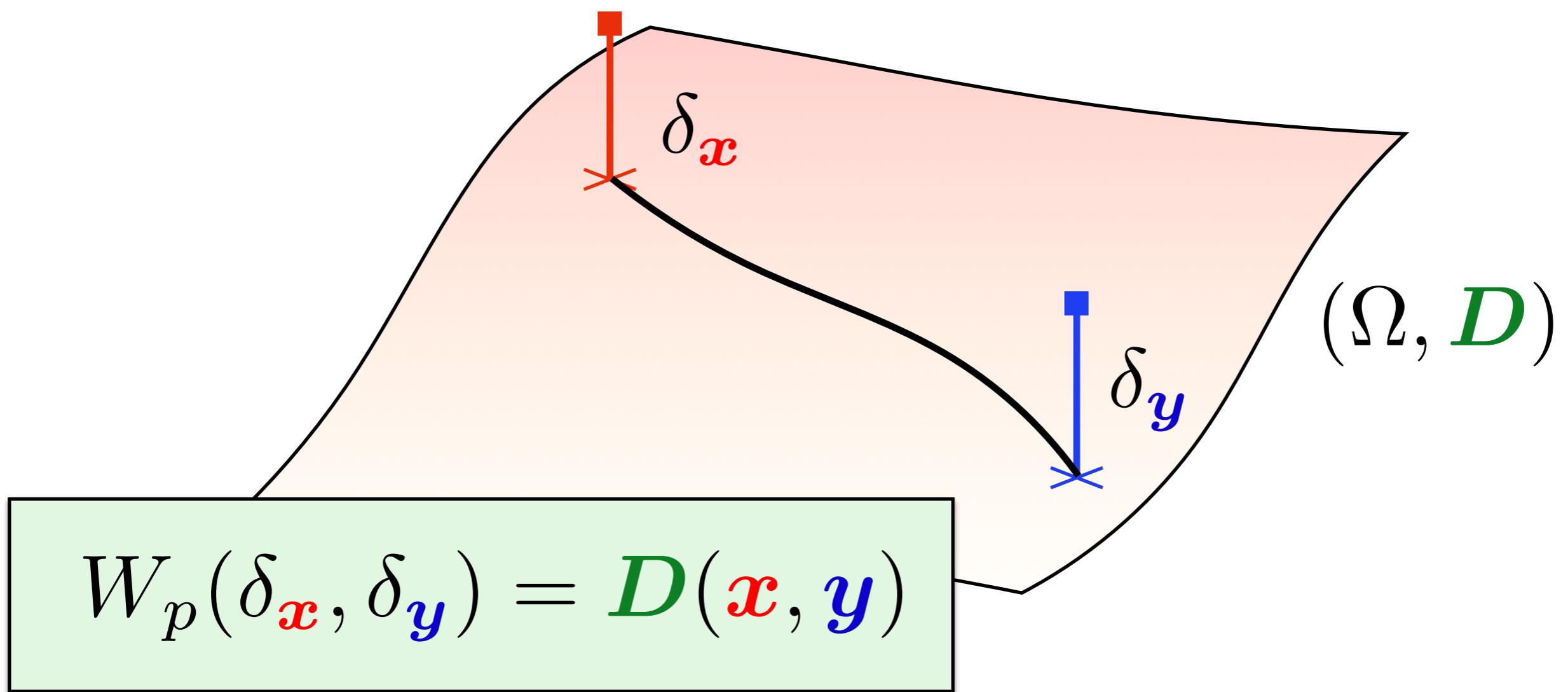
g of positive type.

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

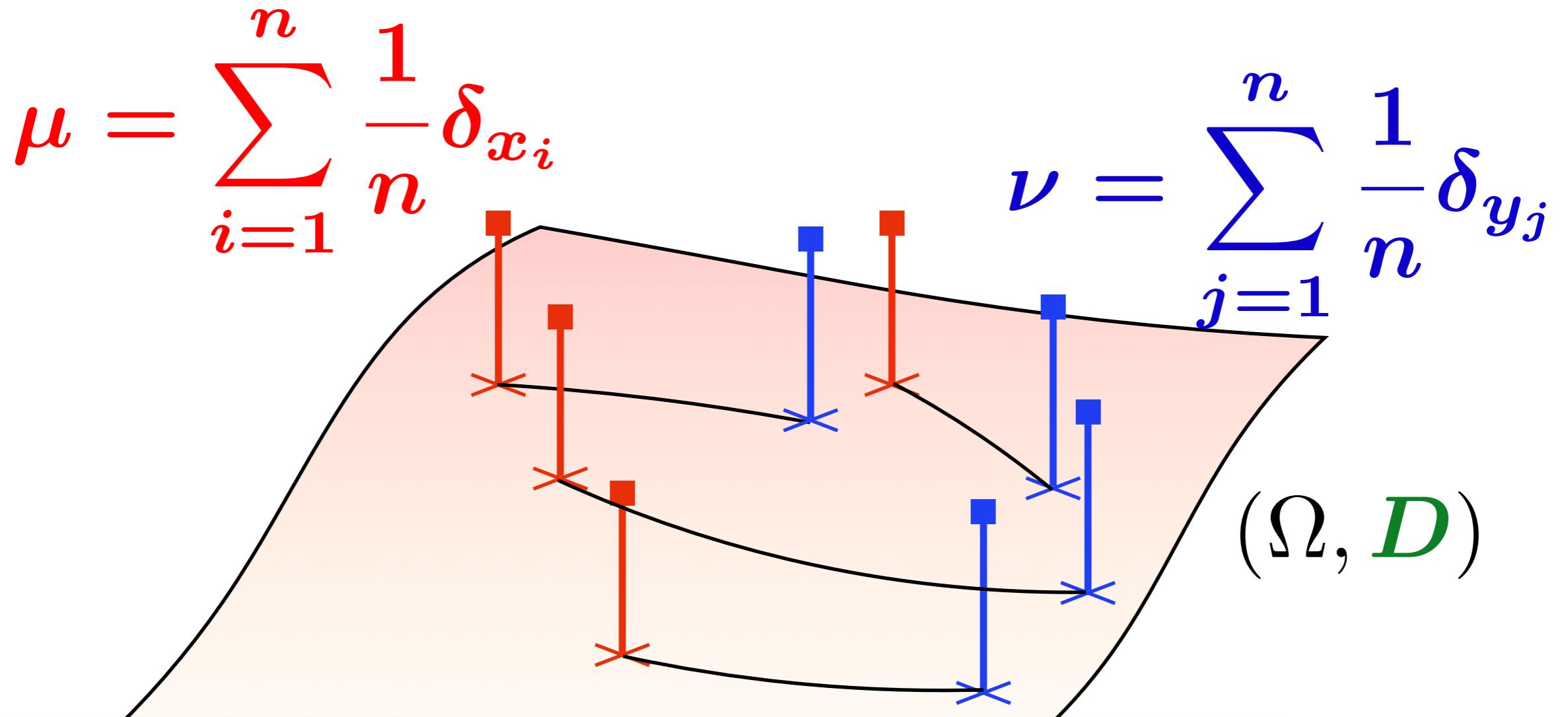
Easy (3): Uniform Ellipses



Wasserstein Between Two Diracs



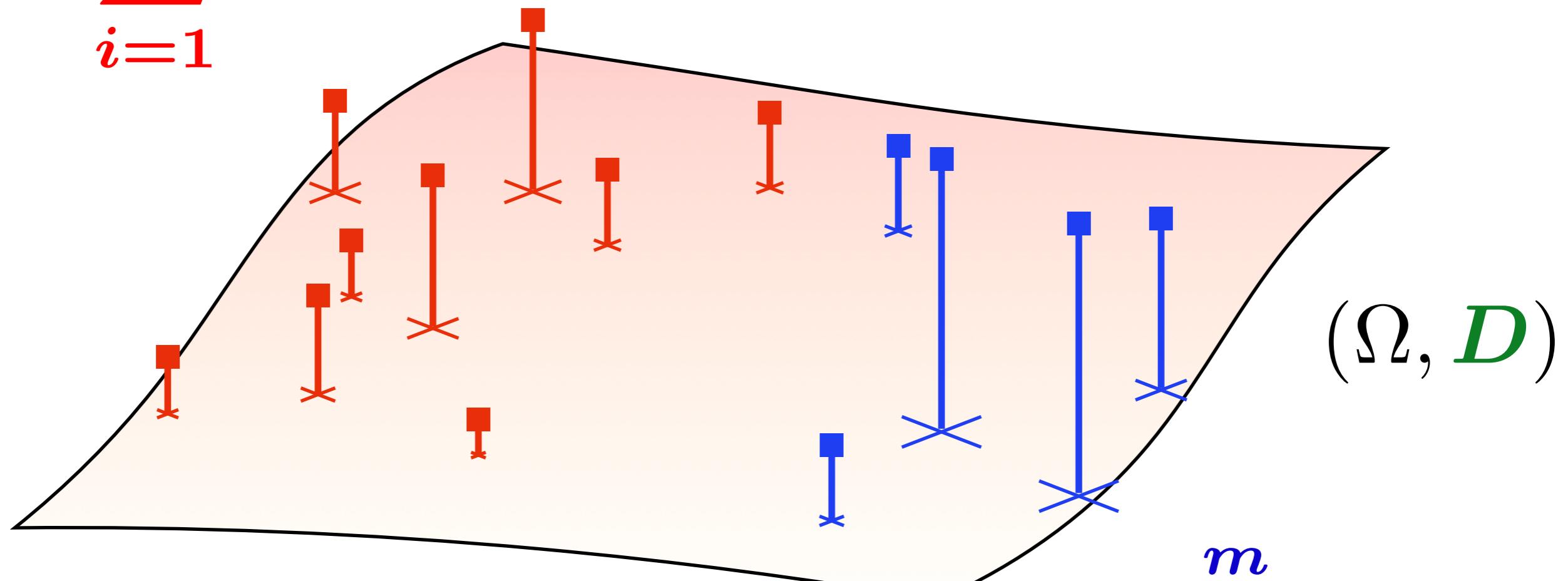
Linear Assignment \subset Wasserstein



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

OT on Two Empirical Measures

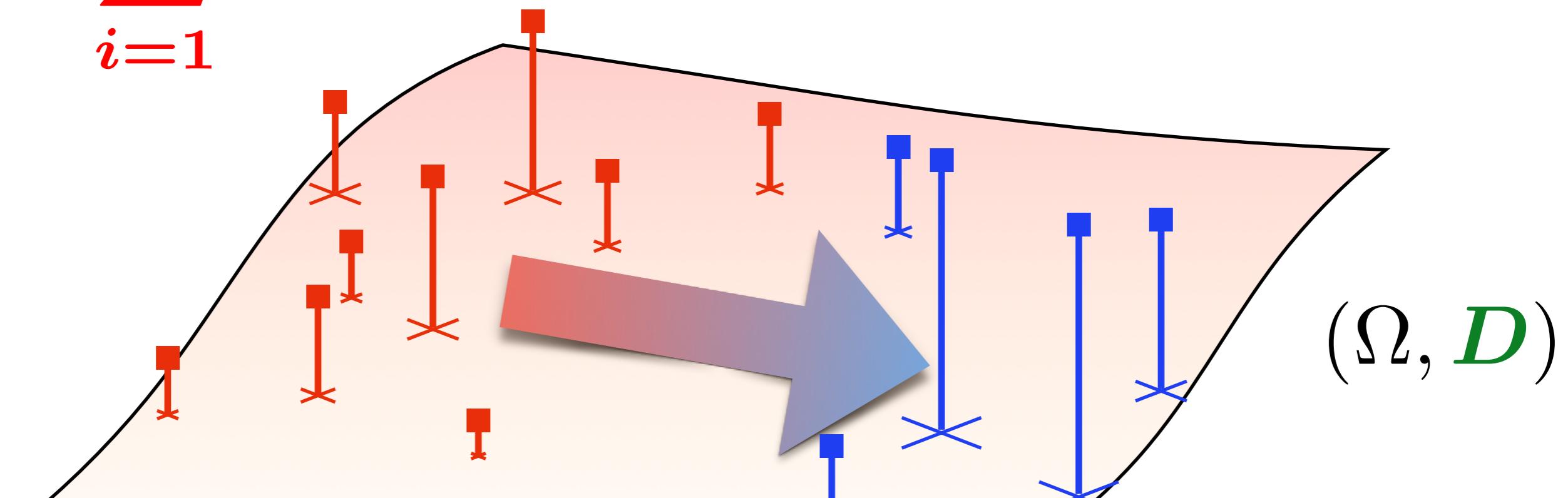
$$\mu = \sum_{i=1}^n a_i \delta_{x_i}$$



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

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

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

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

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

Def. Dual OT problem

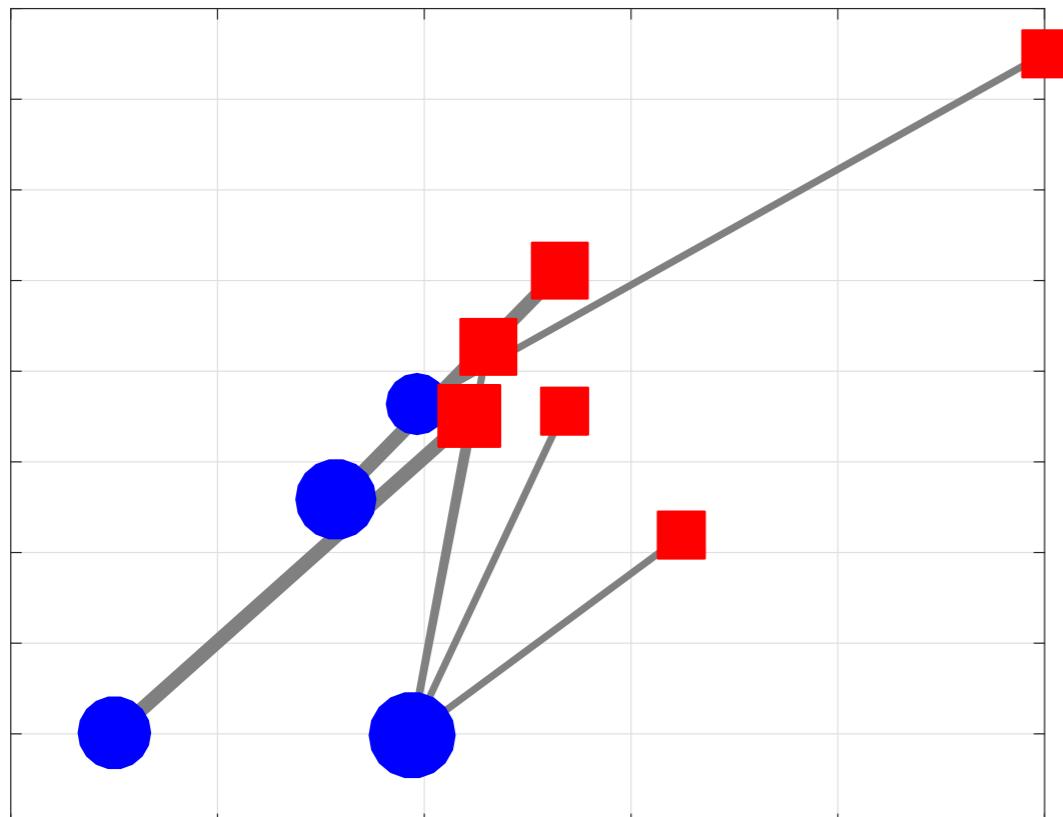
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(\mathbf{x}_i, \mathbf{y}_j)^p}} \alpha^T \boldsymbol{a} + \beta^T \boldsymbol{b}$$

Dual Kantorovich Problem

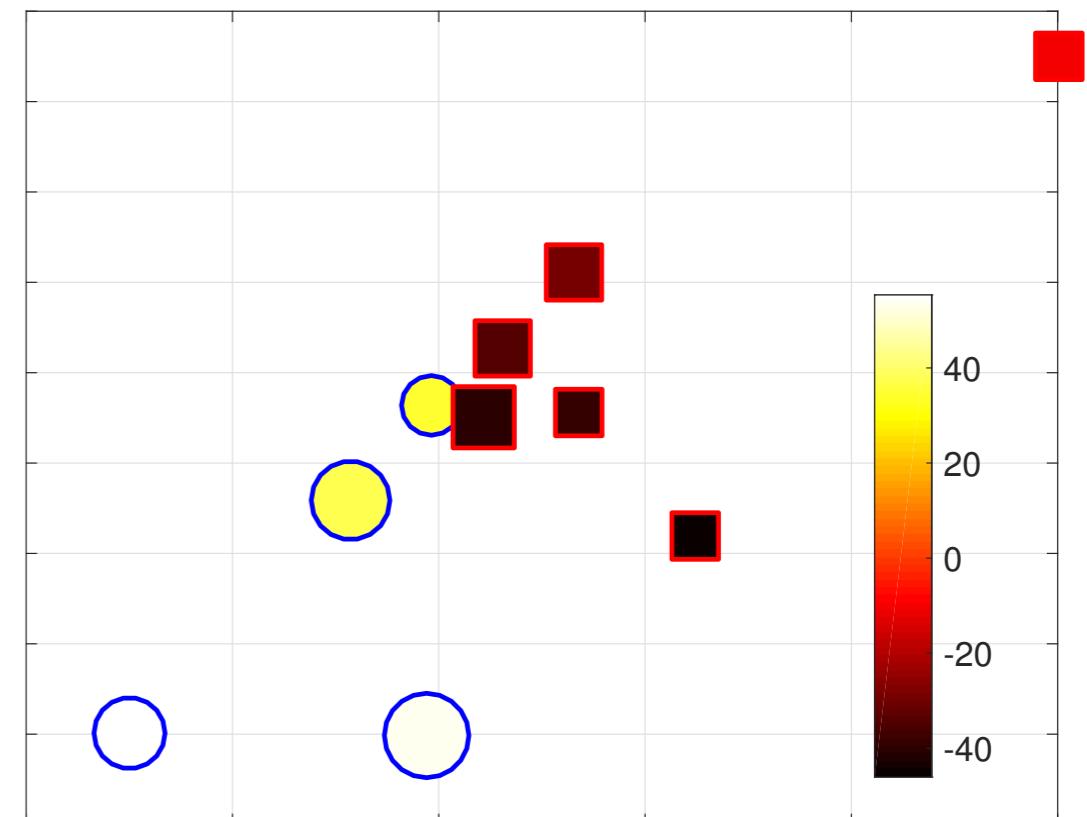
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\substack{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} \\ \boldsymbol{P} \mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T \mathbf{1}_n = \boldsymbol{b}}} \langle \boldsymbol{P}, M_{\mathbf{XY}} \rangle$$

Def. Dual OT problem

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



75

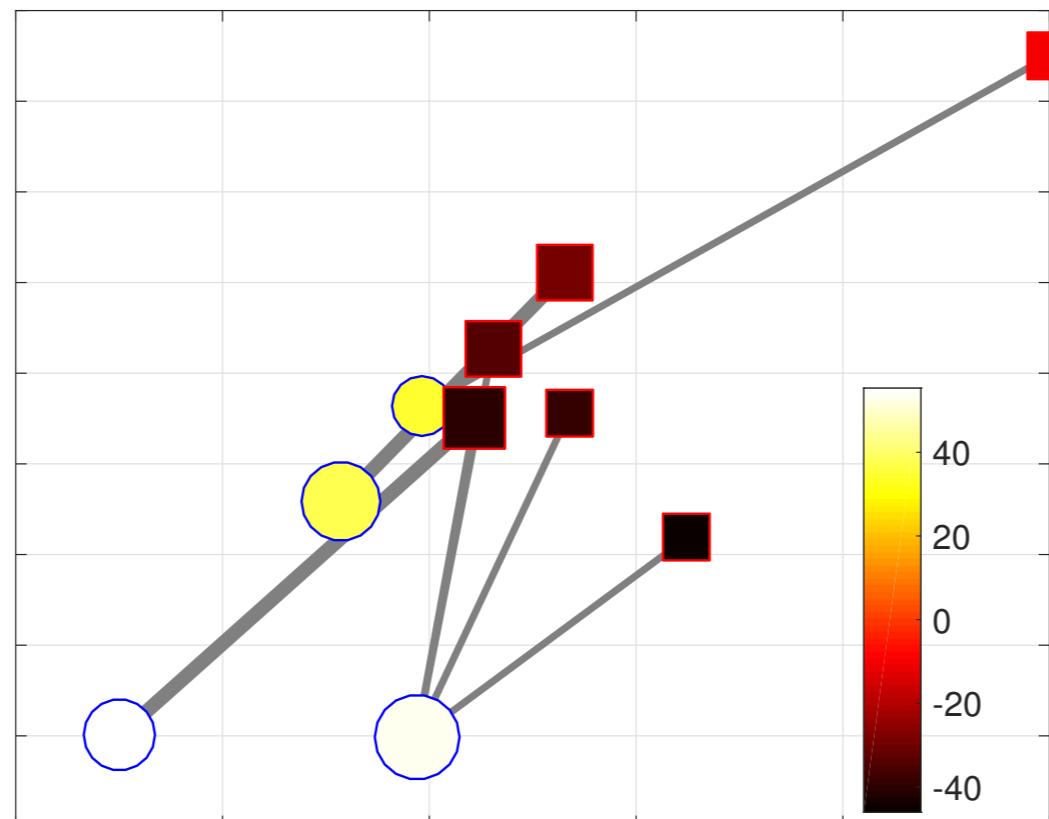


Dual Kantorovich Problem

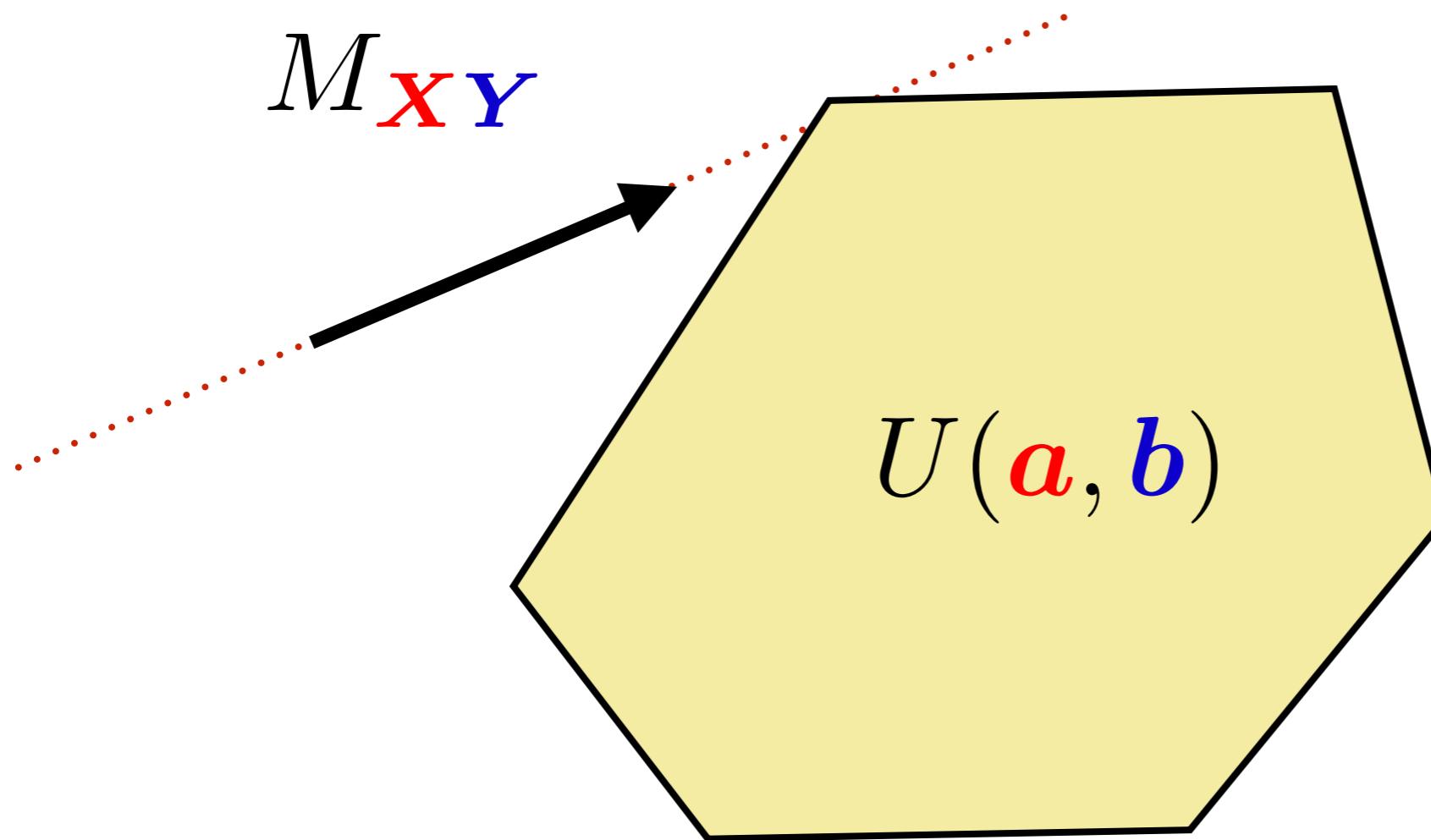
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\substack{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} \\ \boldsymbol{P} \mathbf{1}_m = \boldsymbol{a}, \boldsymbol{P}^T \mathbf{1}_n = \boldsymbol{b}}} \langle \boldsymbol{P}, M_{\mathbf{XY}} \rangle$$

Def. Dual OT problem

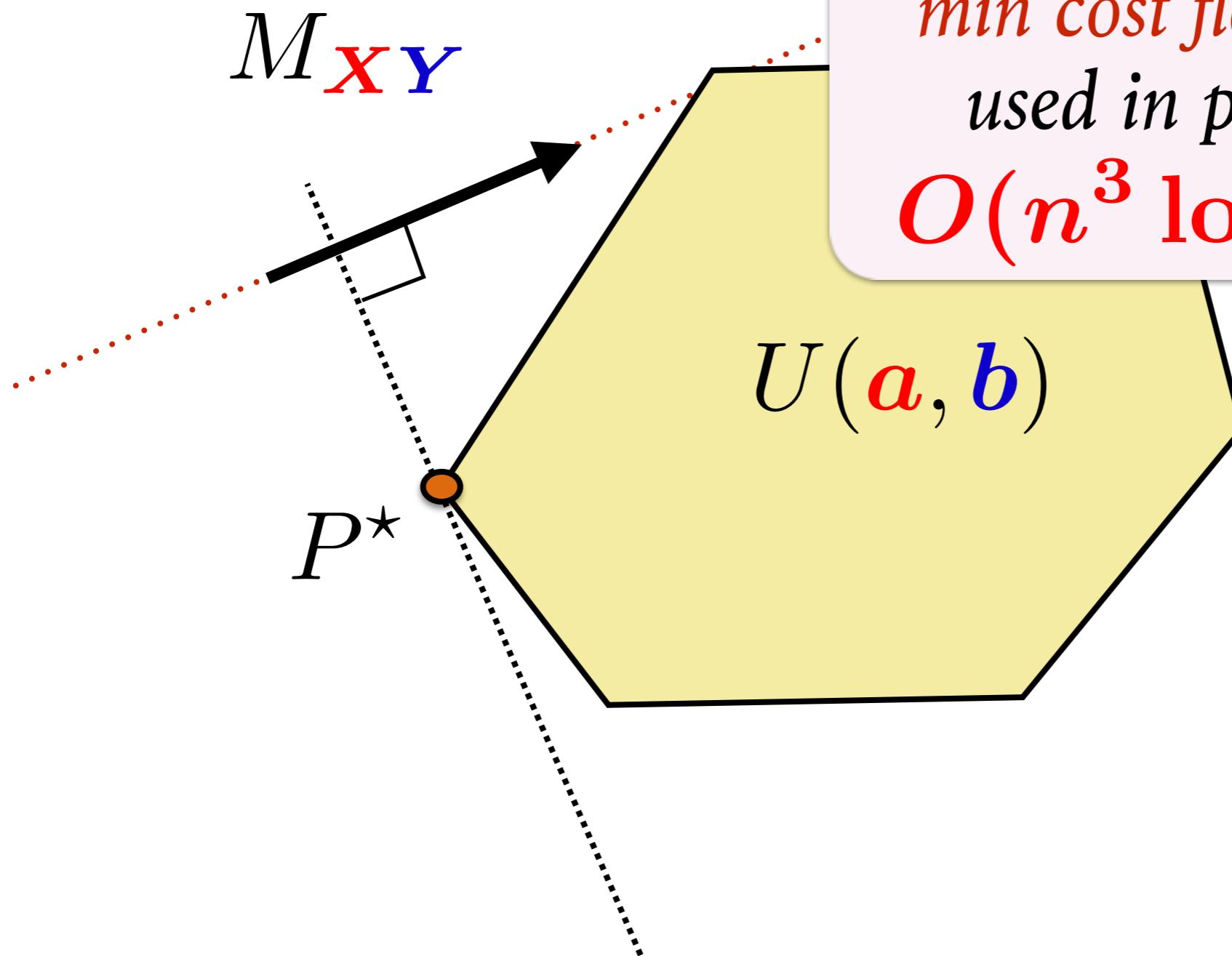
$$W_p^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^m \\ \alpha_i + \beta_j \leq D(\mathbf{x}_i, \mathbf{y}_j)^p}} \alpha^T \boldsymbol{a} + \beta^T \boldsymbol{b}$$



Solving the OT Problem



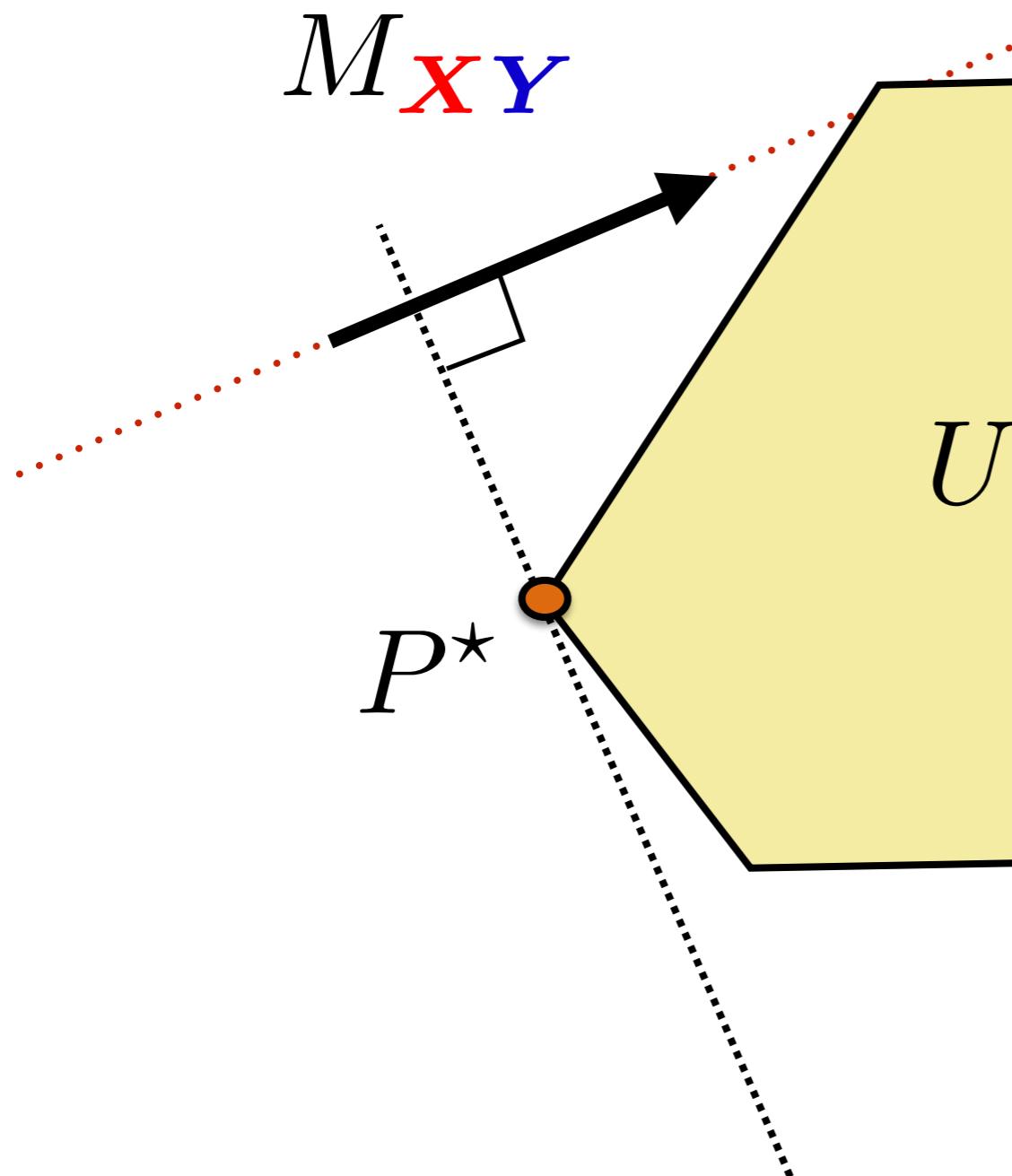
Solving the OT Problem



*min cost flow solver
used in practice.
 $O(n^3 \log(n))$*



Solving the OT Problem

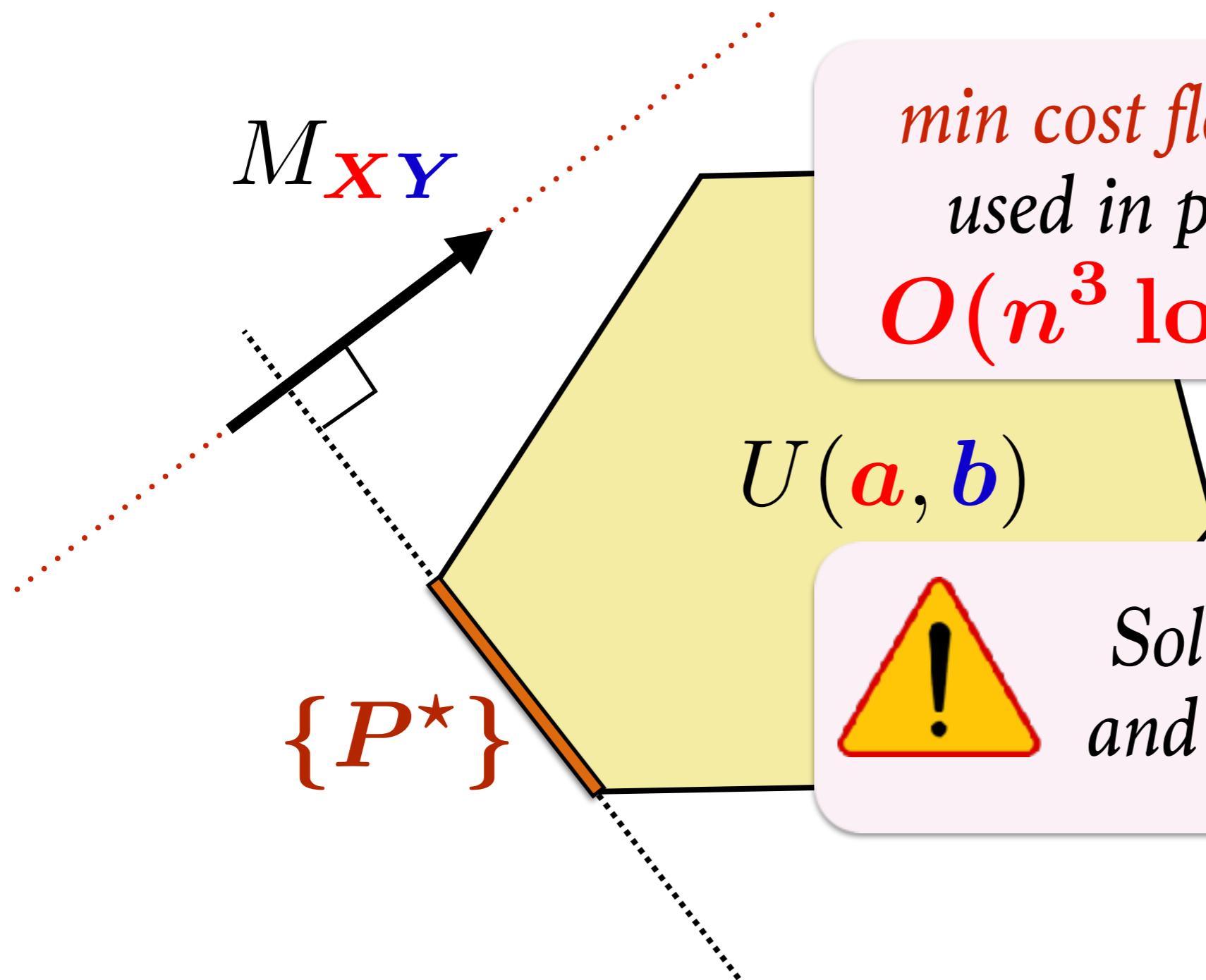


*min cost flow solver
used in practice.
 $O(n^3 \log(n))$*



Solution P^ unstable
and not always unique.*

Solving the OT Problem

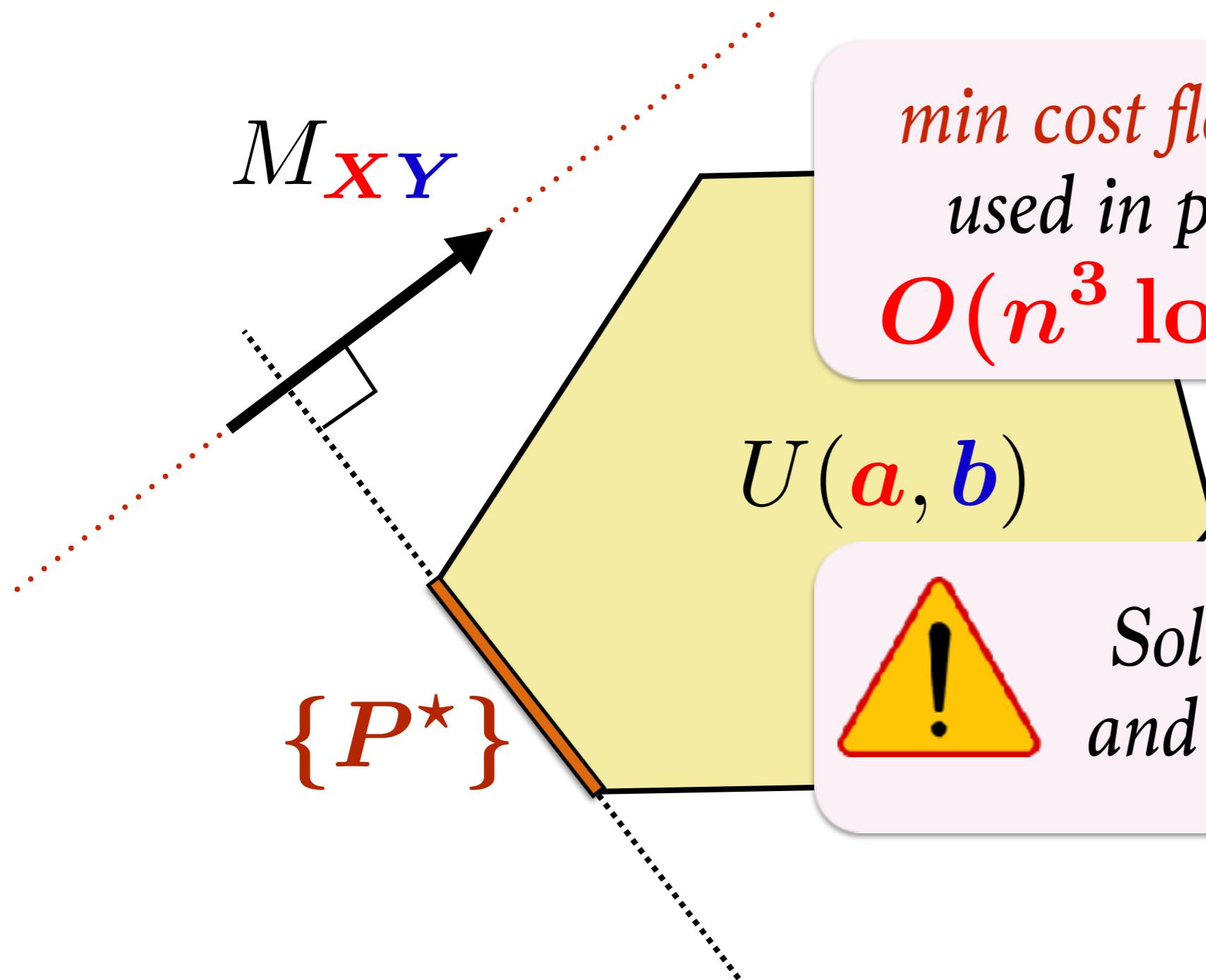


*min cost flow solver
used in practice.
 $O(n^3 \log(n))$*

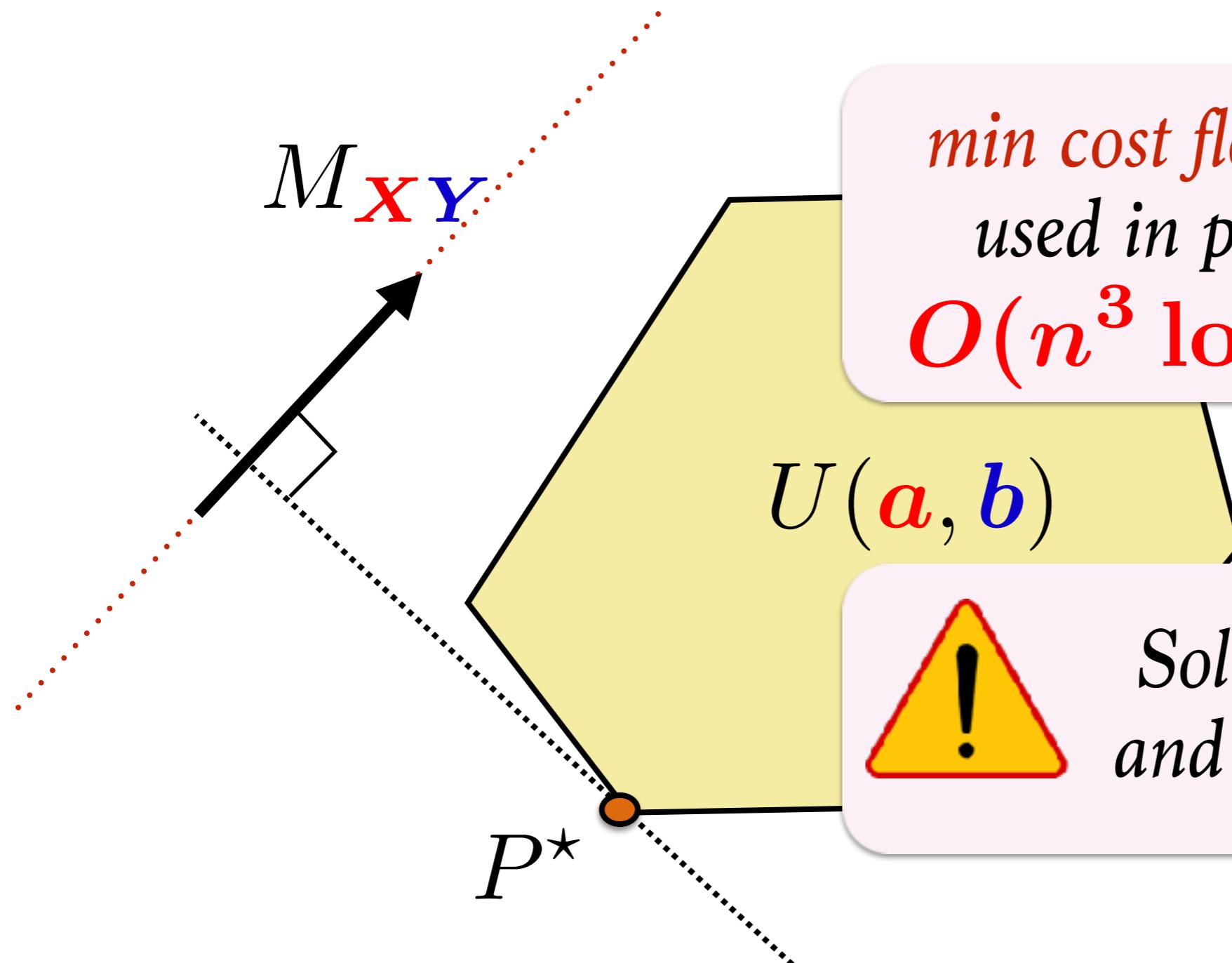


Solution P^ unstable
and not always unique.*

Solving the OT Problem



Solving the OT Problem



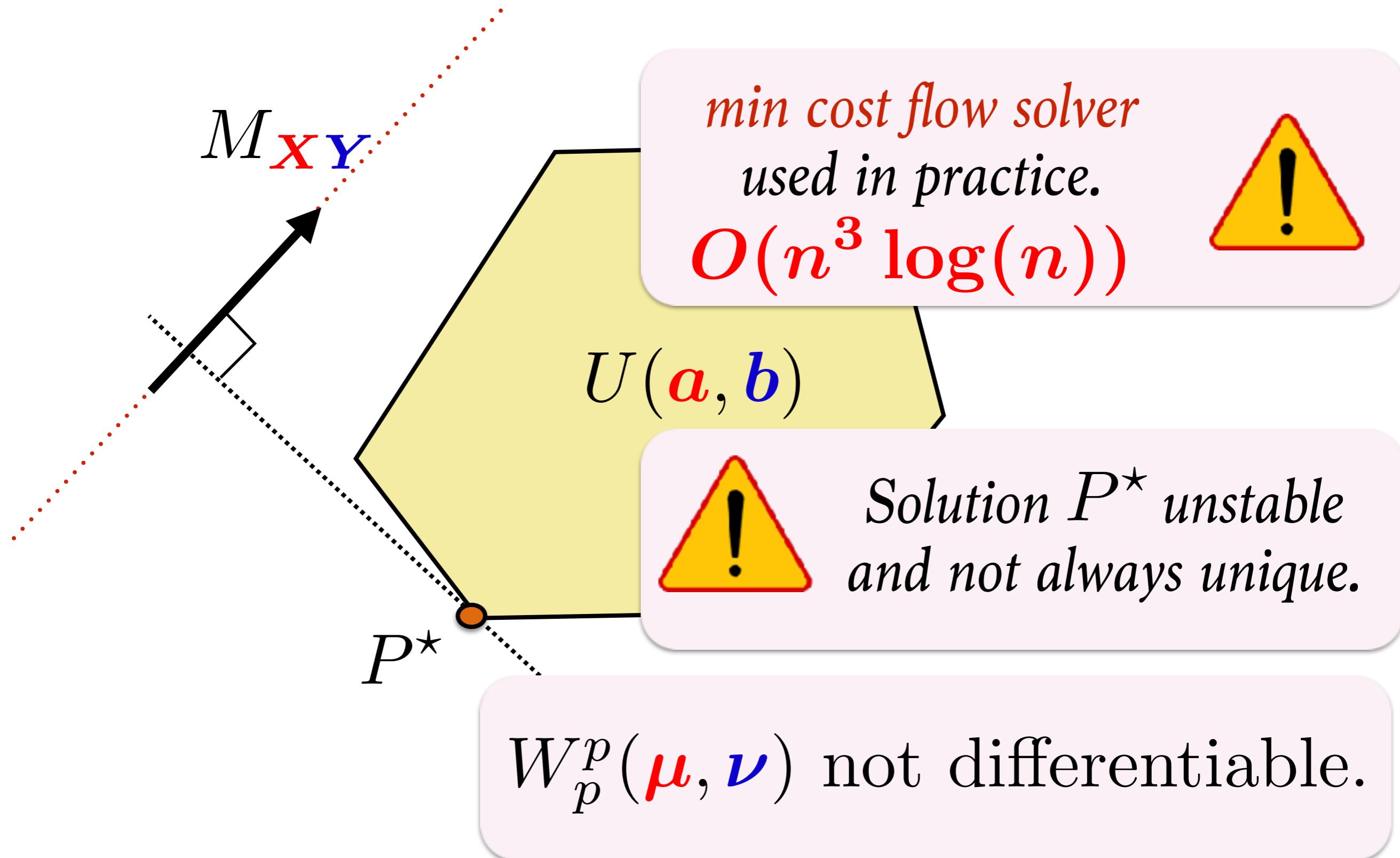
*min cost flow solver
used in practice.
 $O(n^3 \log(n))$*



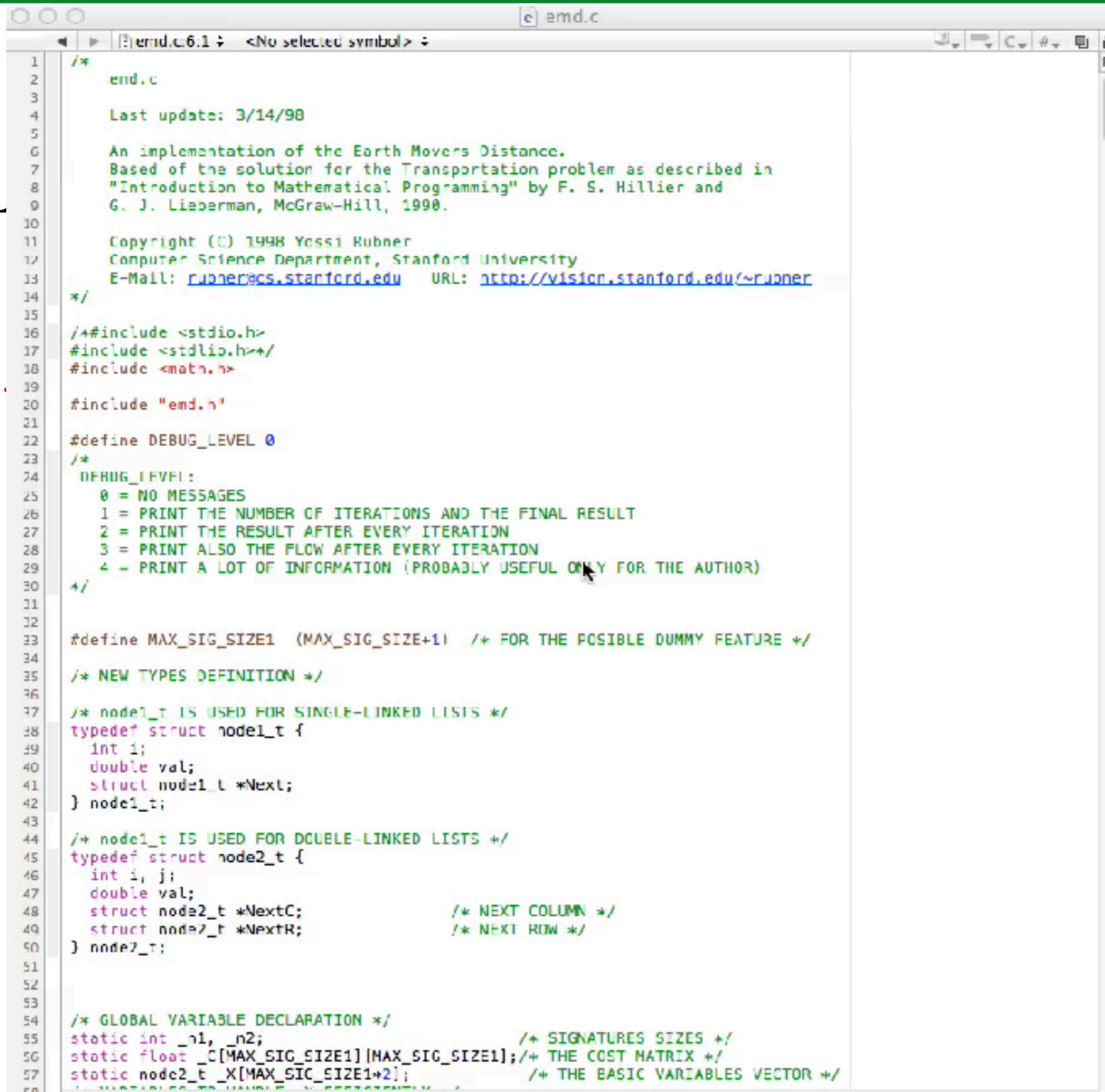
Solution P^ unstable
and not always unique.*



Solving the OT Problem



Discrete OT Problem



```
emd.c:6:1 <No selected symbol>
```

```
1  /*
2   * end.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
```

Discrete OT Problem

```
emdc6:1 <No selected symbol>
```

```
1  /*
2   * end.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  /*
25  * DEBUG_LEVEL:
26  * 0 = NO MESSAGES
27  * 1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
28  * 2 = PRINT THE RESULT AFTER EVERY ITERATION
29  * 3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
30  * 4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
31  */
32  /*
33  * #define MAX_SIG_SIZE1 (MAX_SIG_SIZE+1) /* FOR THE POSSIBLE DUMMY FEATURE */
34  */
35  /*
36  * NEW TYPES DEFINITION */
37  /*
38  * node1_t IS USED FOR SINGLE-LINKED LISTS */
39  typedef struct node1_t {
40  int i;
41  double val;
42  struct node1_t *Next;
43  } node1_t;
44  /*
45  * node2_t IS USED FOR DOUBLE-LINKED LISTS */
46  typedef struct node2_t {
47  int i, j;
48  double val;
49  struct node2_t *NextC; /* NEXT COLUMN */
50  struct node2_t *NextR; /* NEXT ROW */
51  } node2_t;
52  /*
53  */
54  /*
55  * GLOBAL VARIABLE DECLARATION */
56  static int _n1, _n2; /* SIGNATURES SIZES */
57  static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1]; /* THE COST MATRIX */
58  static node2_t _X[MAX_SIG_SIZE1+2]; /* THE BASIC VARIABLES VECTOR */
59  /*
60  * INITIALIZATION OF THE COST MATRIX
61  */
```

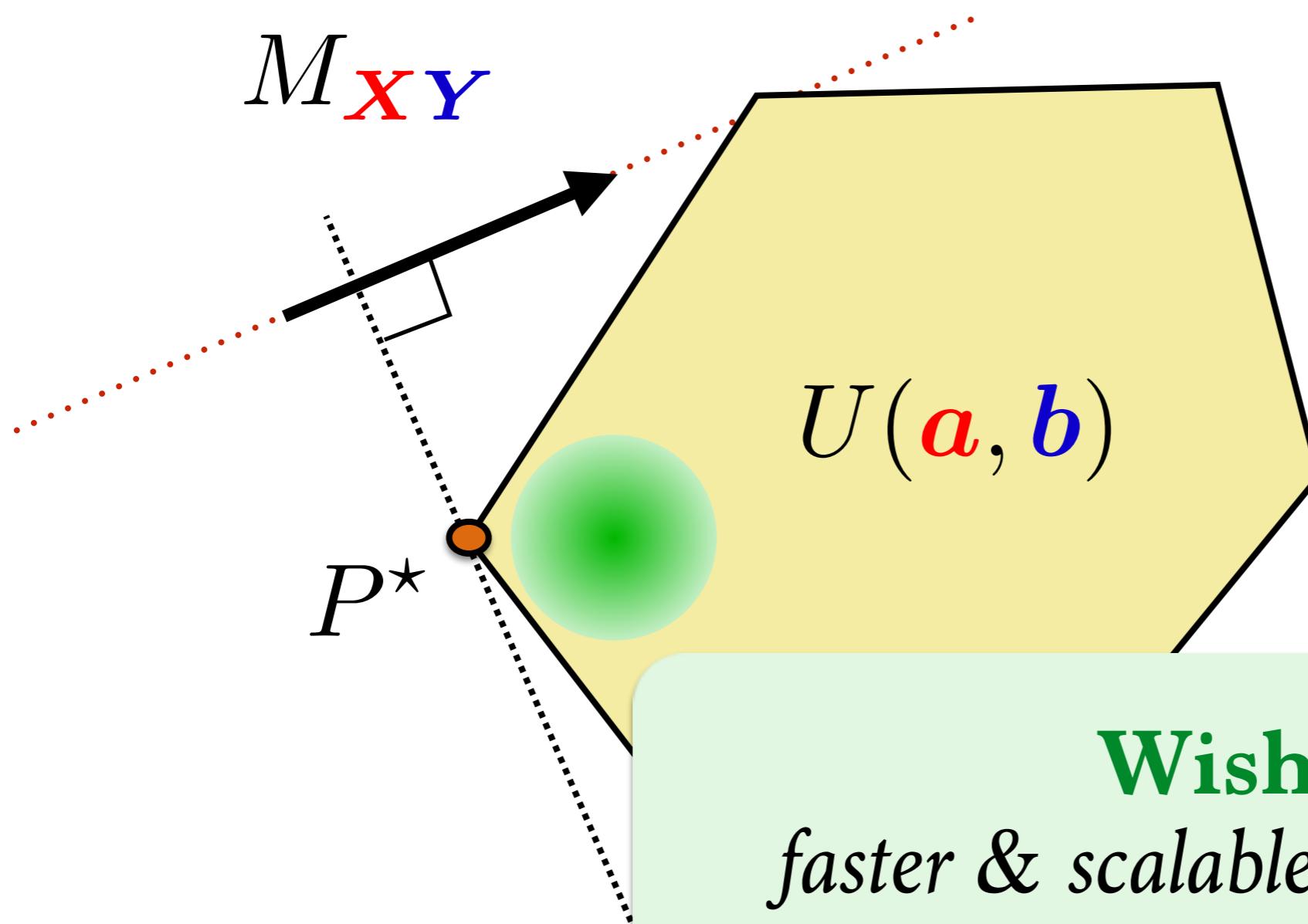


Discrete OT Problem

```
1  /*  
2   end.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 /* node1_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 /* VARIABLES TO MAKE IT CONSISTENT */  
59
```



Solution: Regularization



Wishlist:

*faster & scalable, more stable,
differentiable*

Entropic Regularization [Wilson'62]

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

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

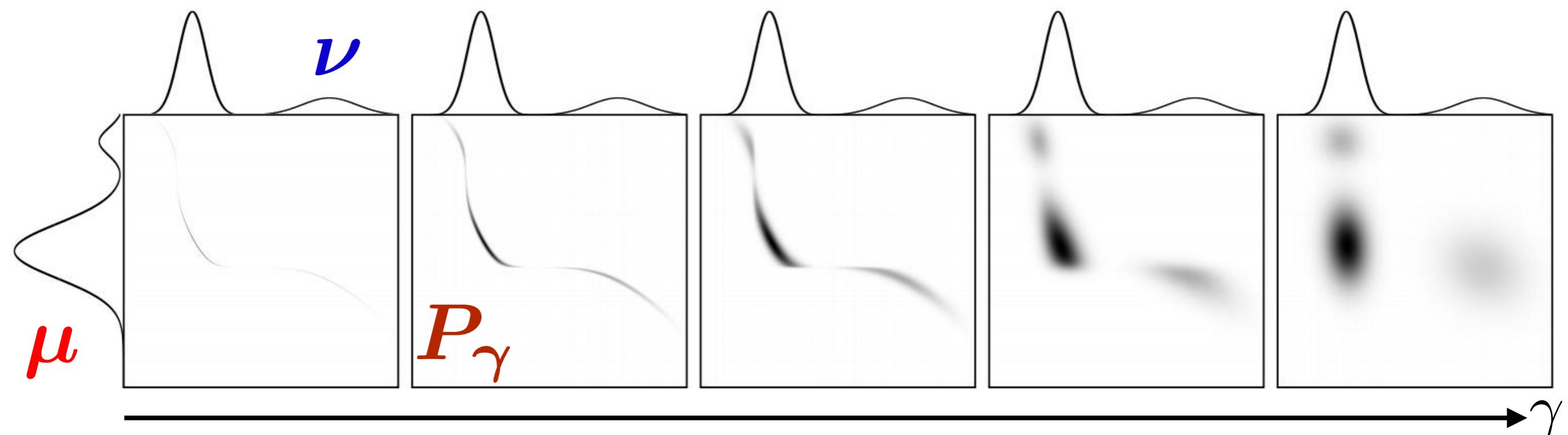
$$E(P) \stackrel{\text{def}}{=} - \sum_{i,j=1}^{nm} P_{ij} (\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

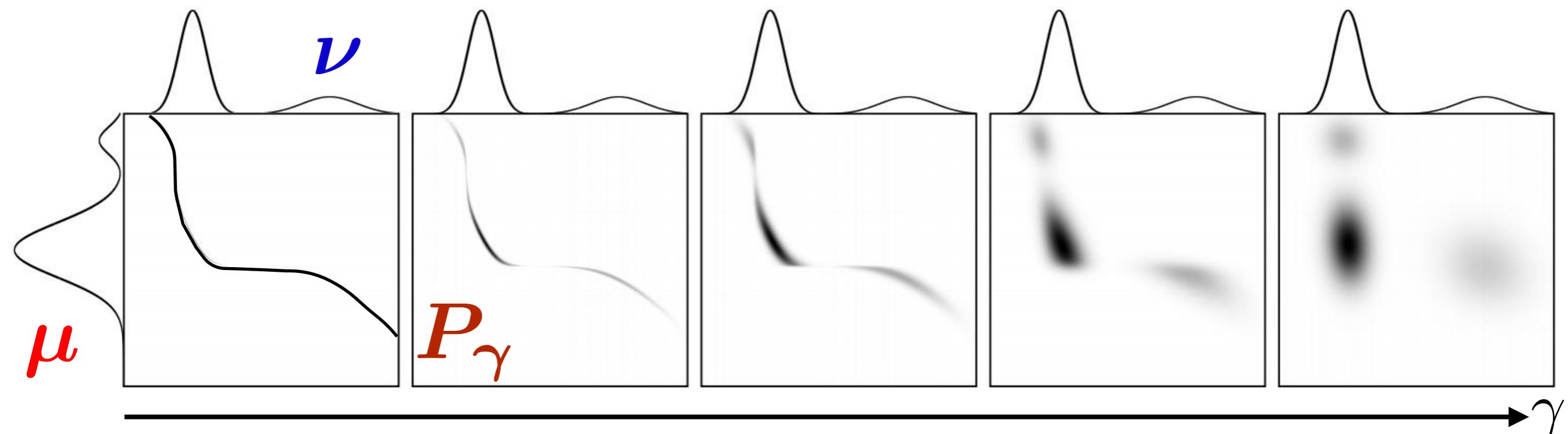


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

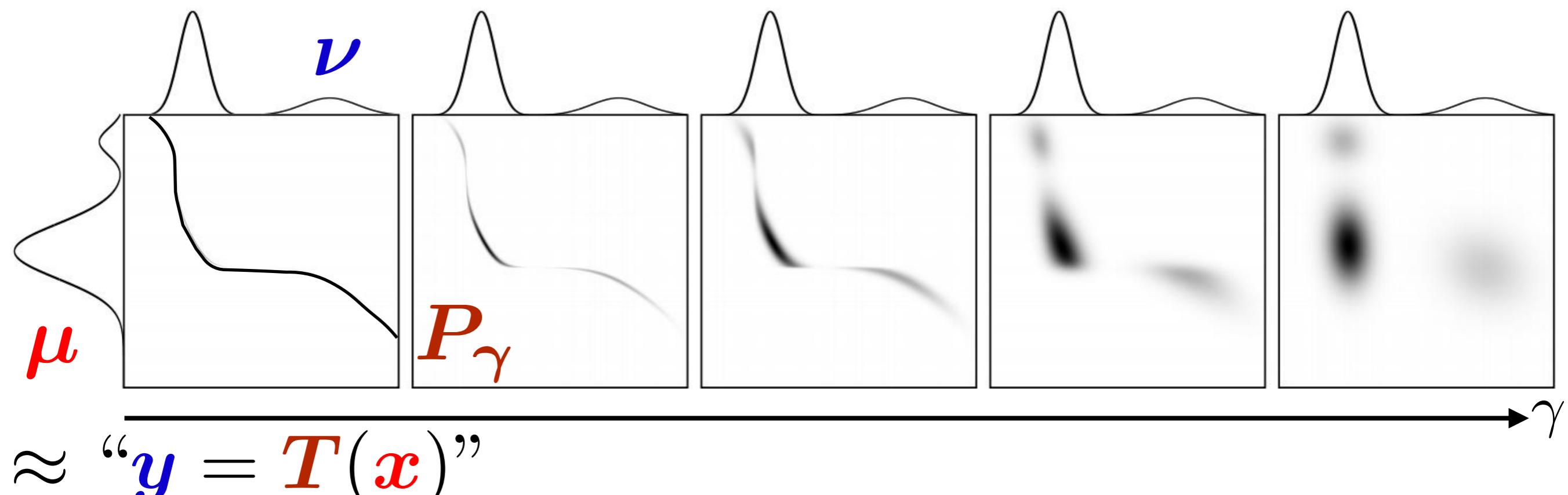


Note: Unique optimal solution because of strong concavity of entropy

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

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

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

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

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

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

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

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

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

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

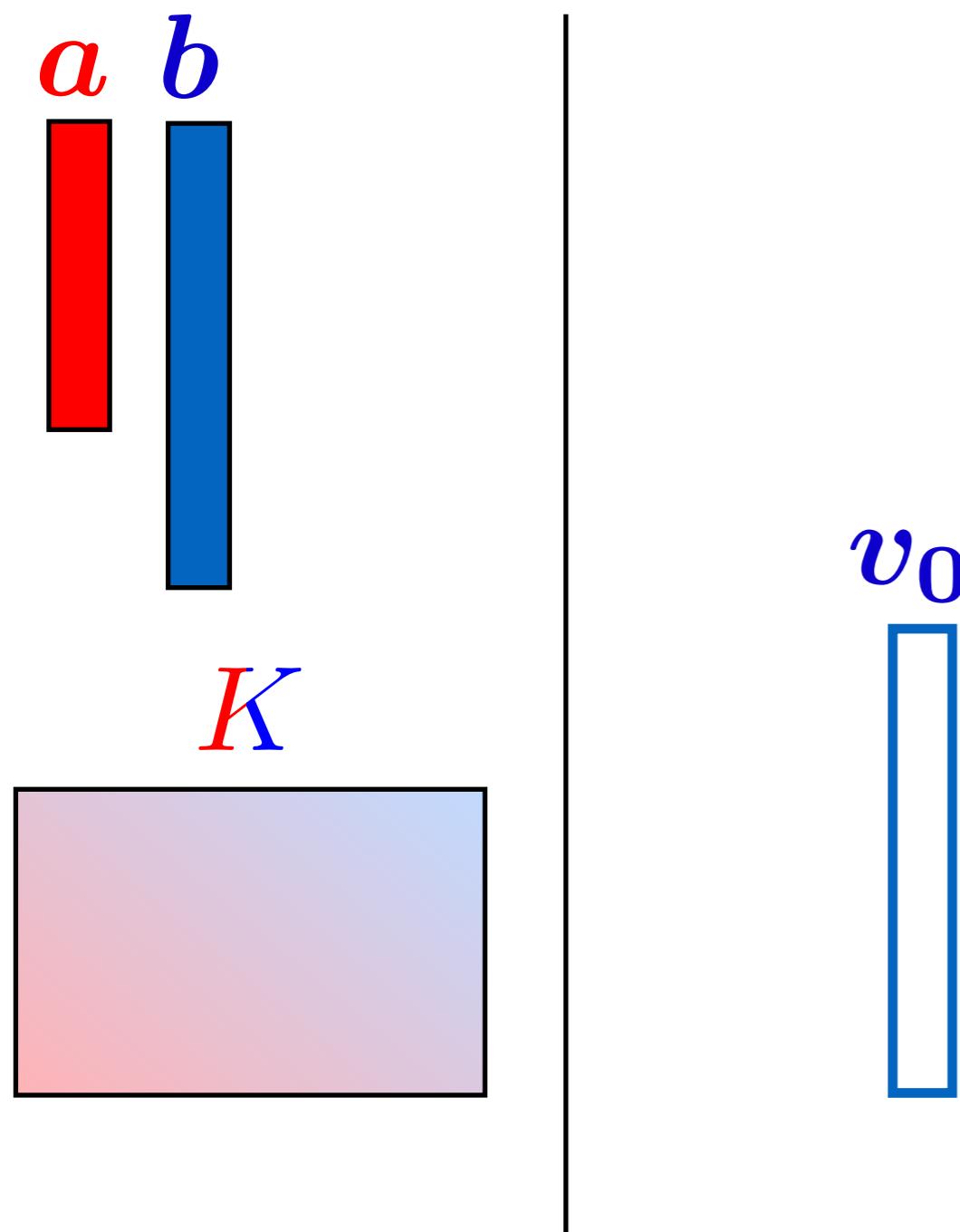
- [Sinkhorn'64] proved convergence for the first time.
- [Lorenz'89] linear convergence, see [Altschuler'17]
- $O(nm)$ complexity, GPGPU parallel [Cuturi'13].
- $O(n \log n)$ on gridded spaces using convolutions.

[Solomon'15]

Fast & Scalable Algorithm

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

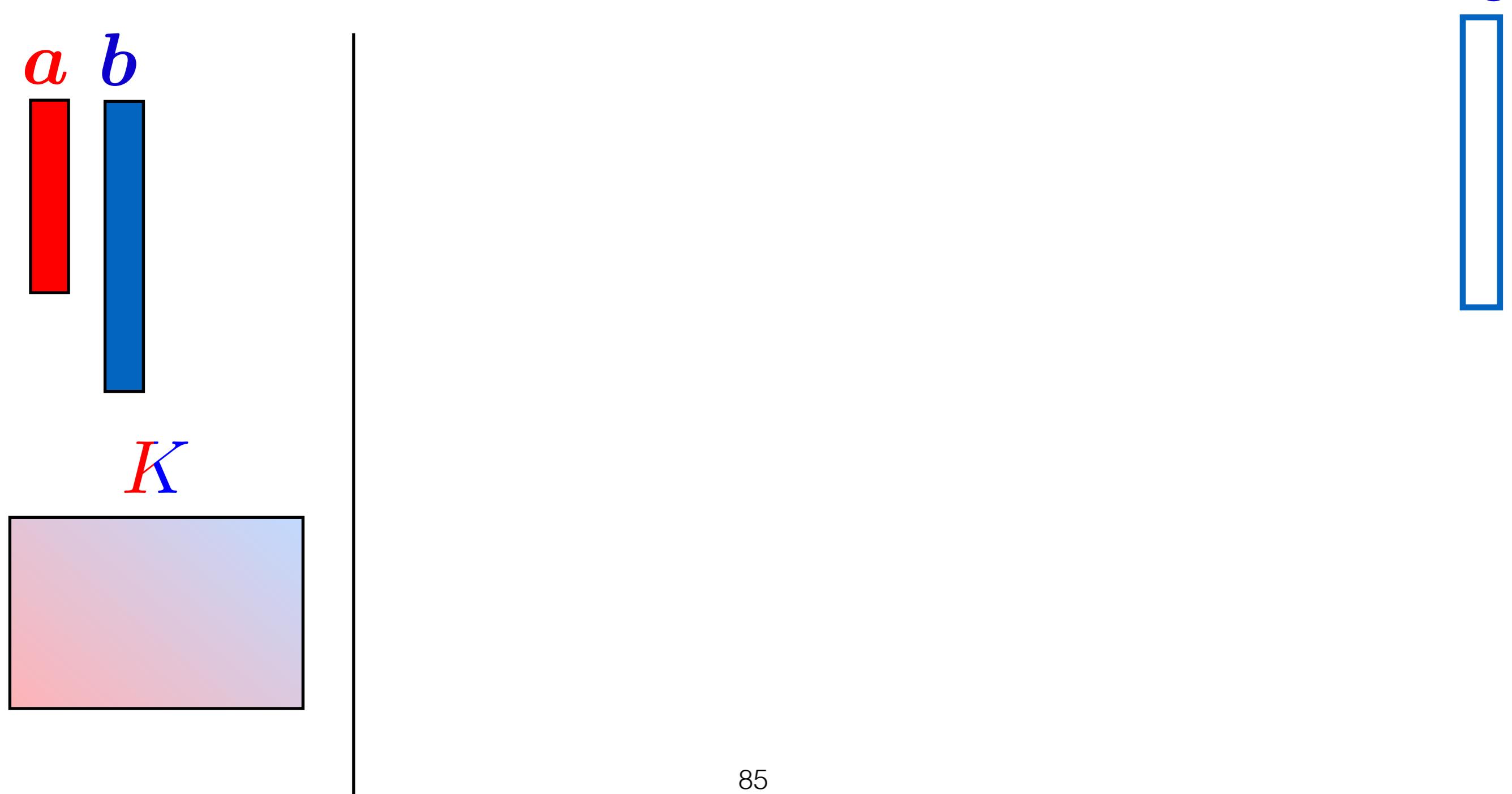
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

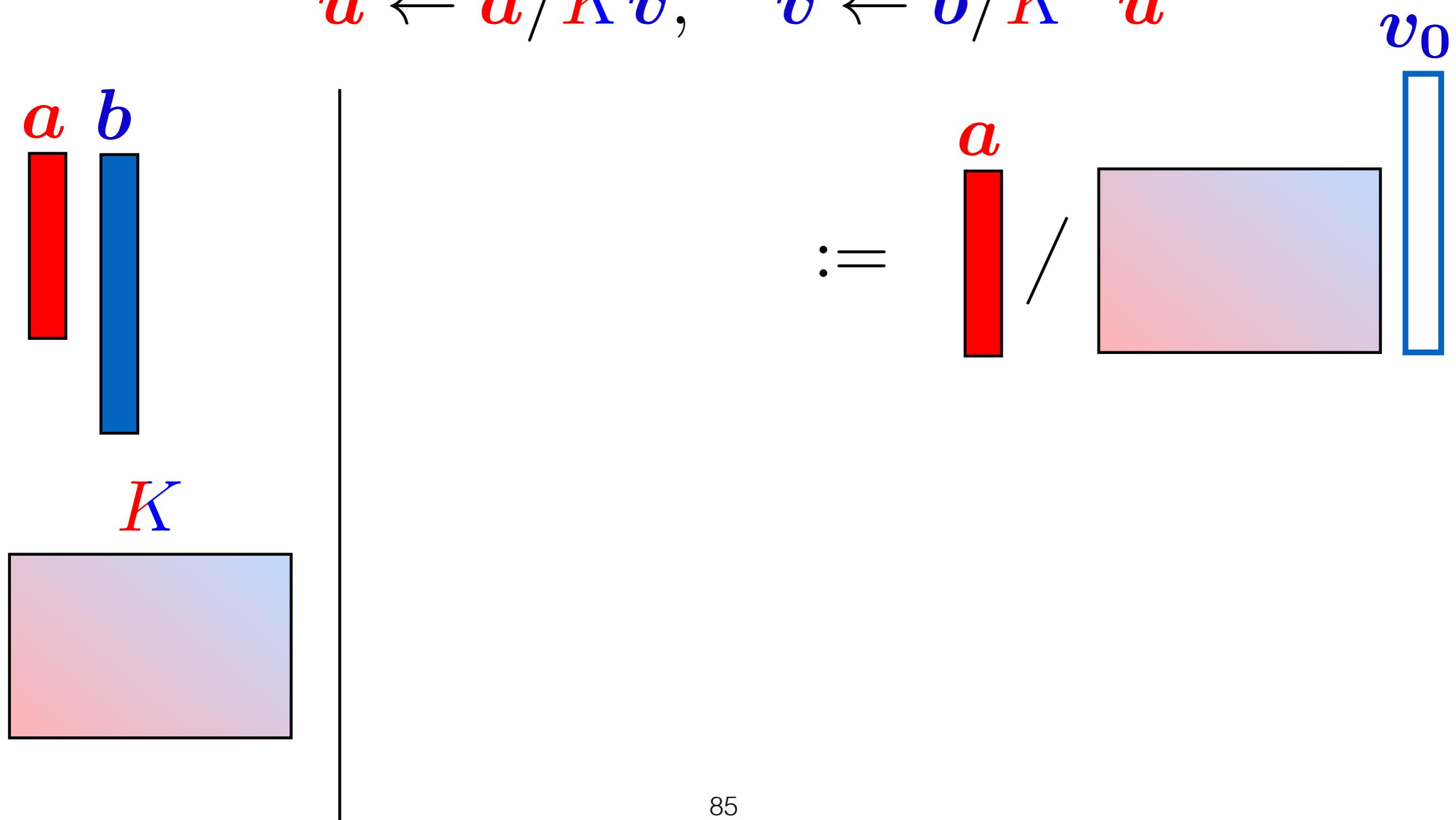
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

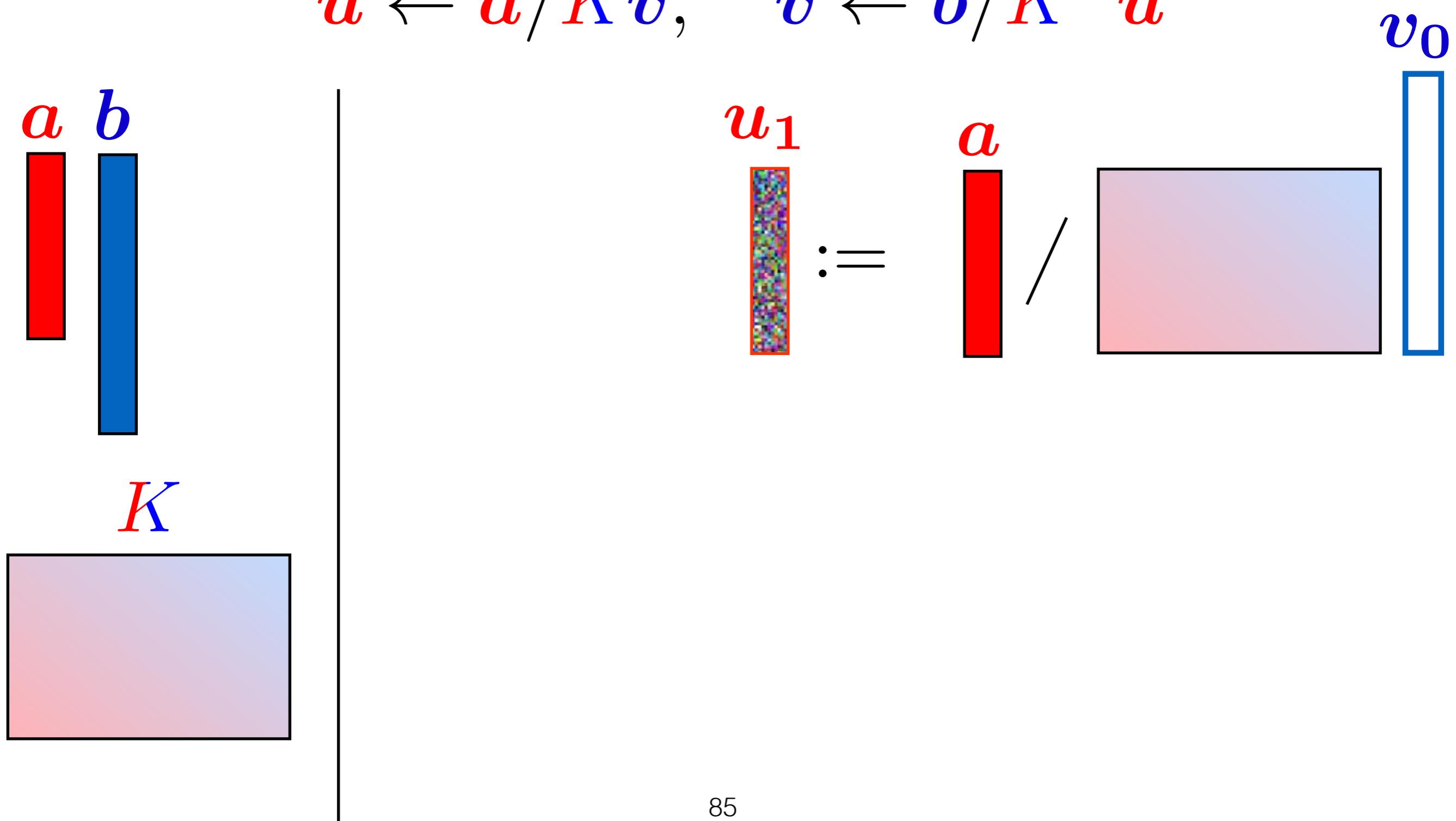
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

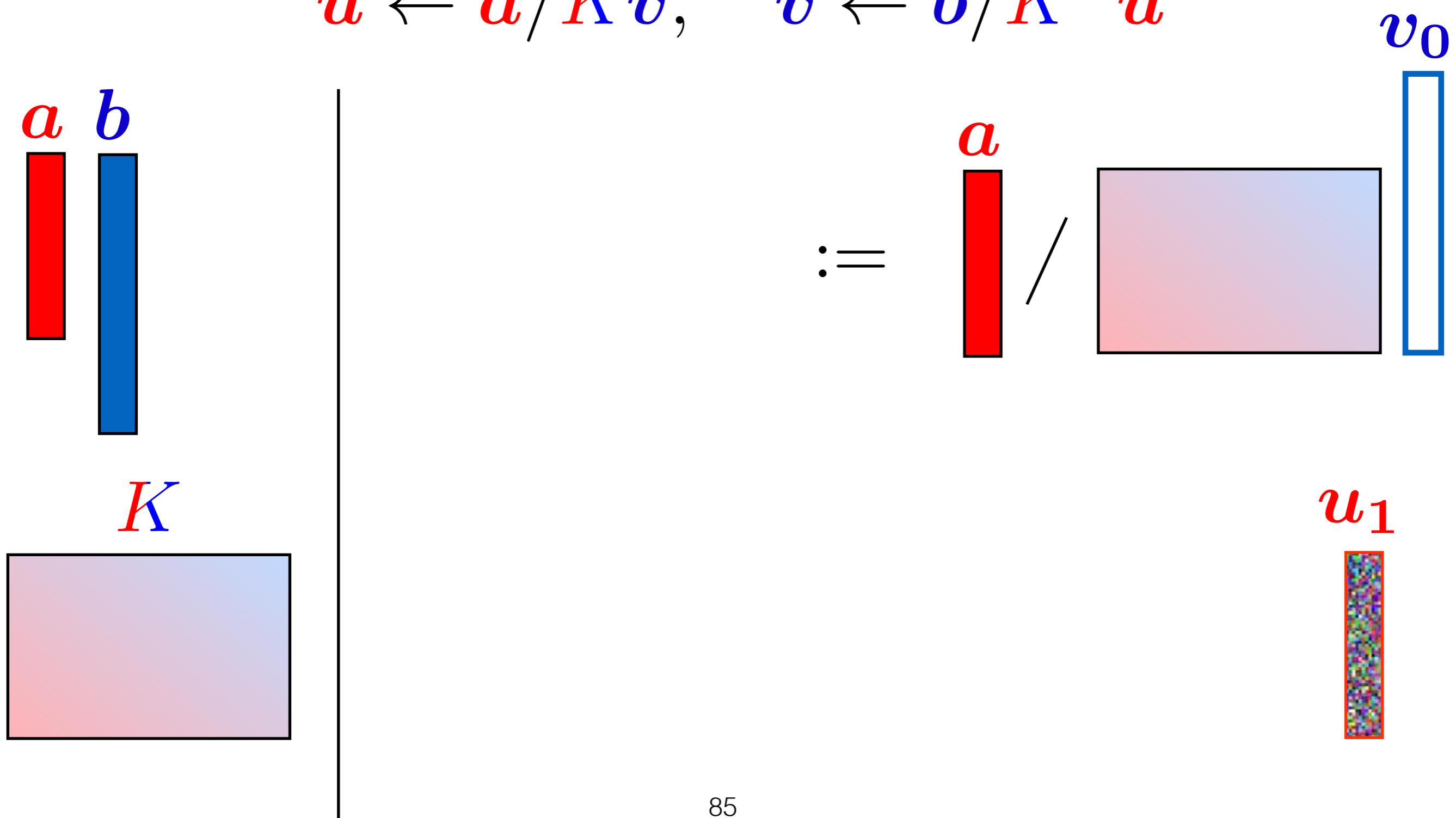
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

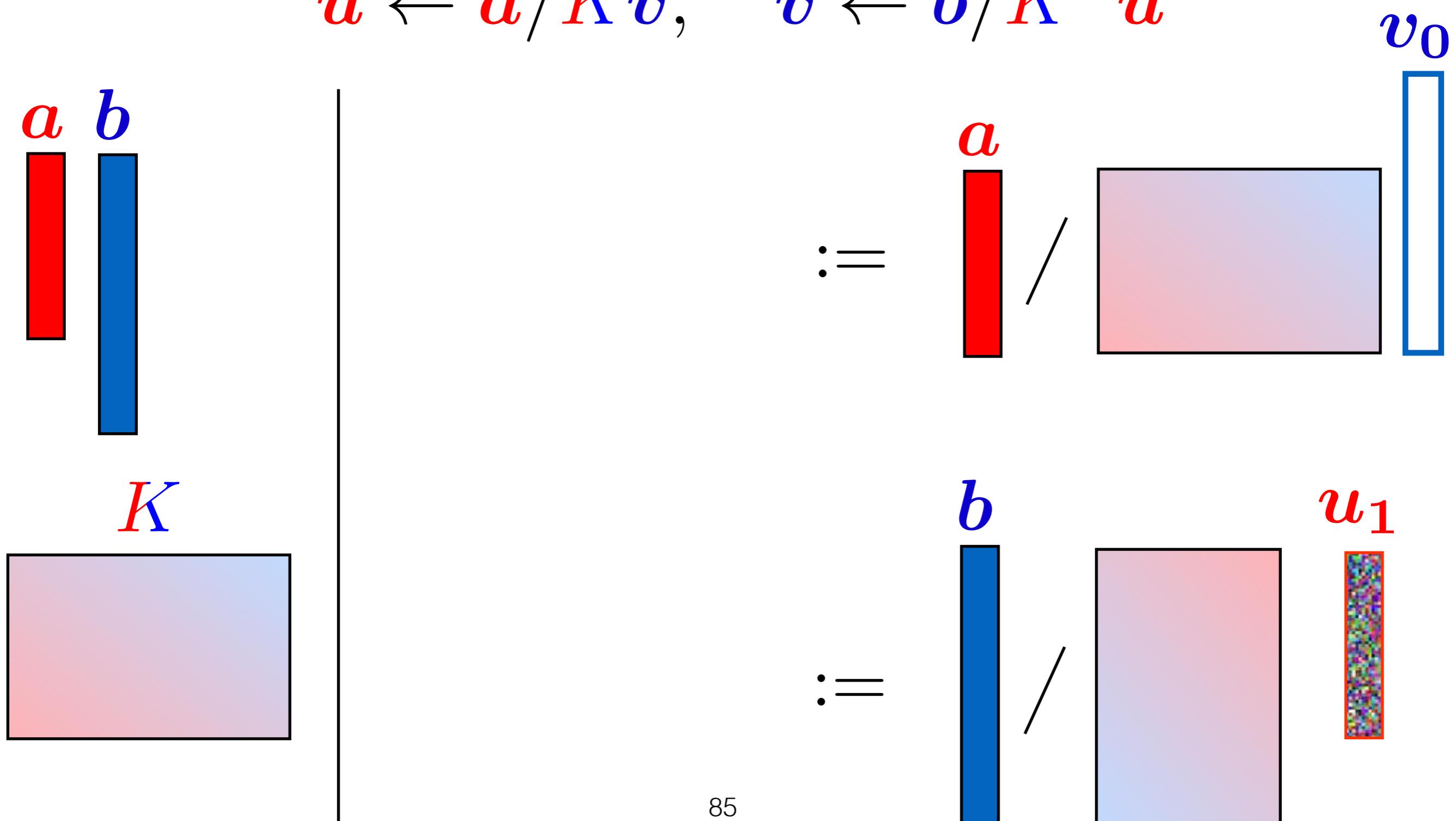
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

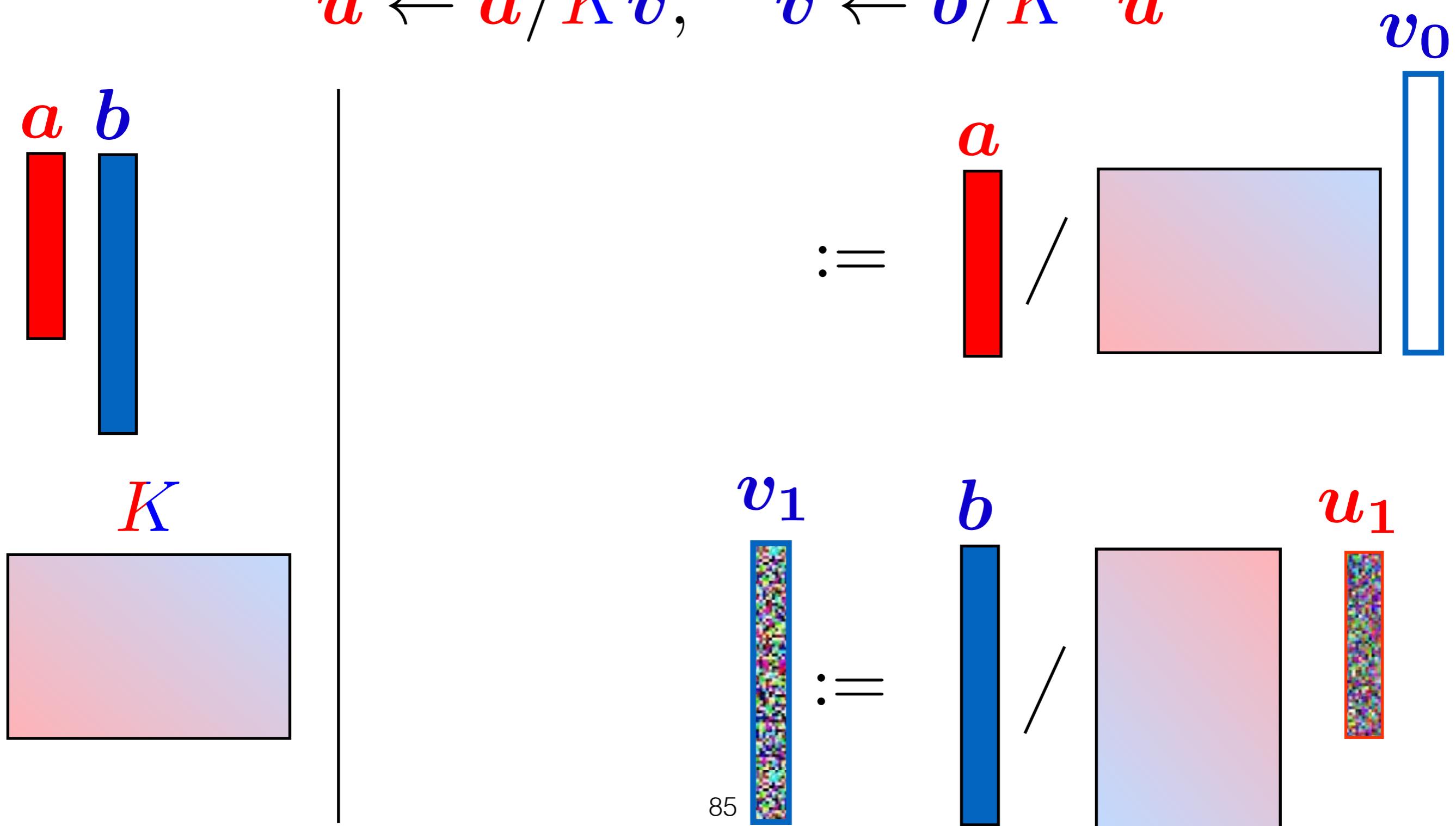
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

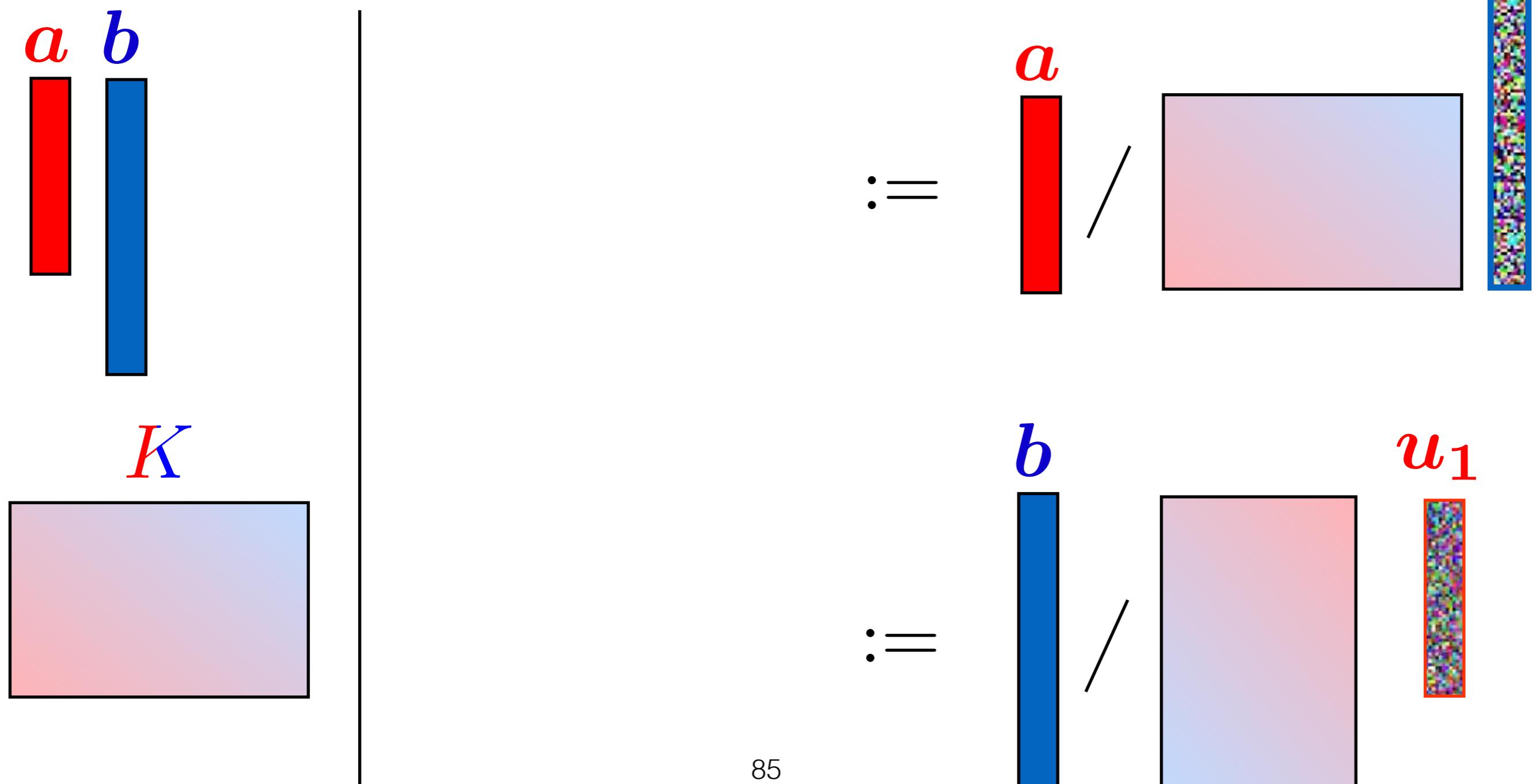
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

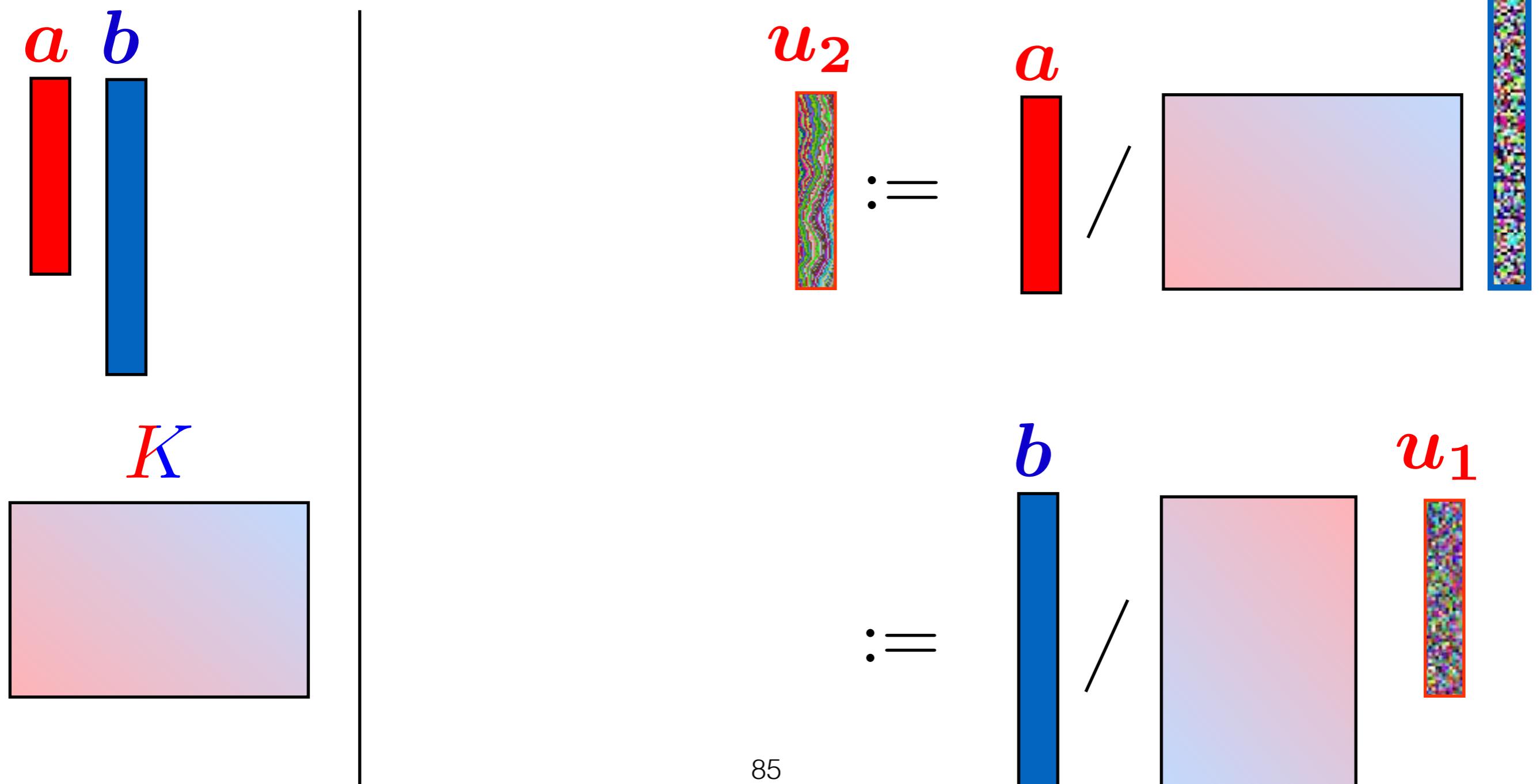
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

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

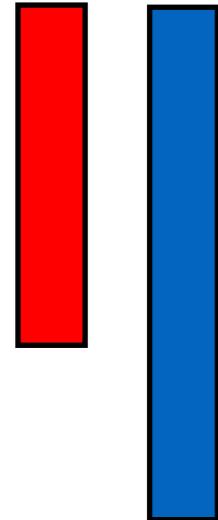


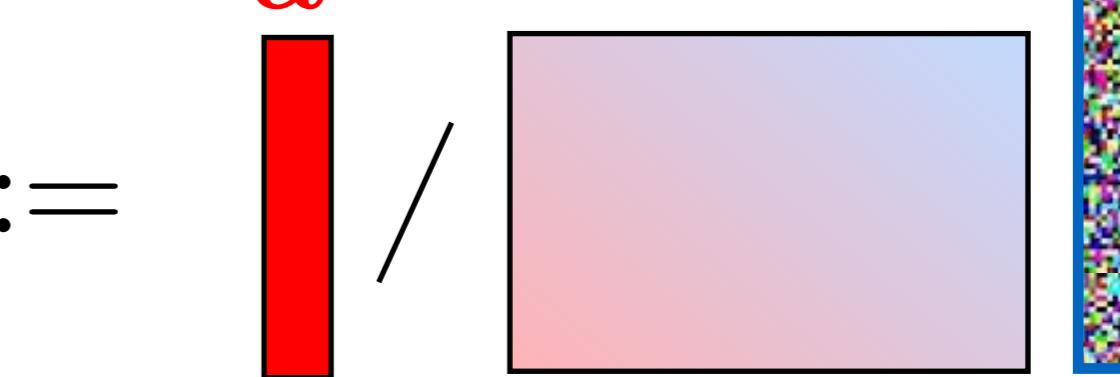
Fast & Scalable Algorithm

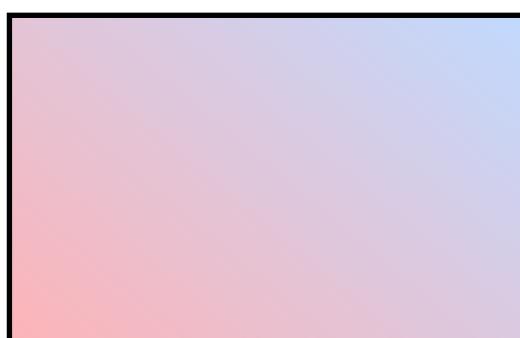
- [Sinkhorn'64] fixed-point iterations for (\mathbf{u}, \mathbf{v})

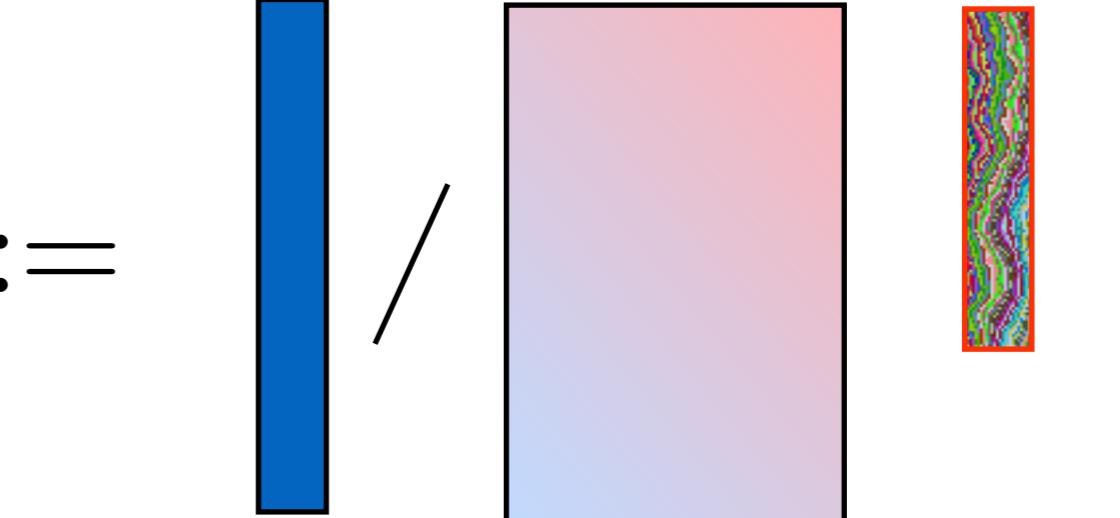
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

v_1

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


$$:= \begin{matrix} \mathbf{a} \\ \mathbf{b} \end{matrix} / \begin{matrix} \mathbf{K} \end{matrix}$$


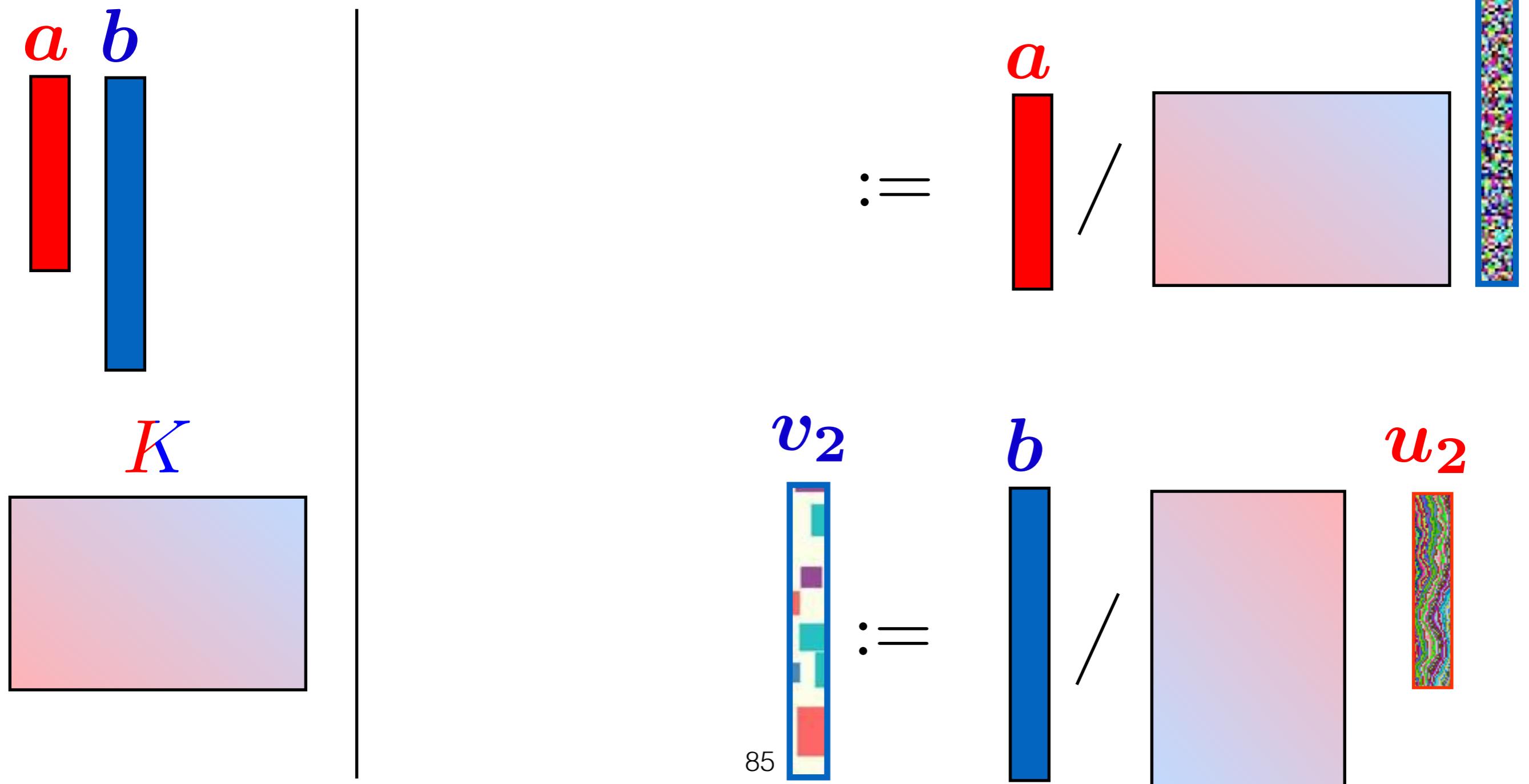
$$\mathbf{K}$$


$$:= \begin{matrix} \mathbf{b} \\ \mathbf{K} \end{matrix} / \begin{matrix} \mathbf{u}_2 \end{matrix}$$


Fast & Scalable Algorithm

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

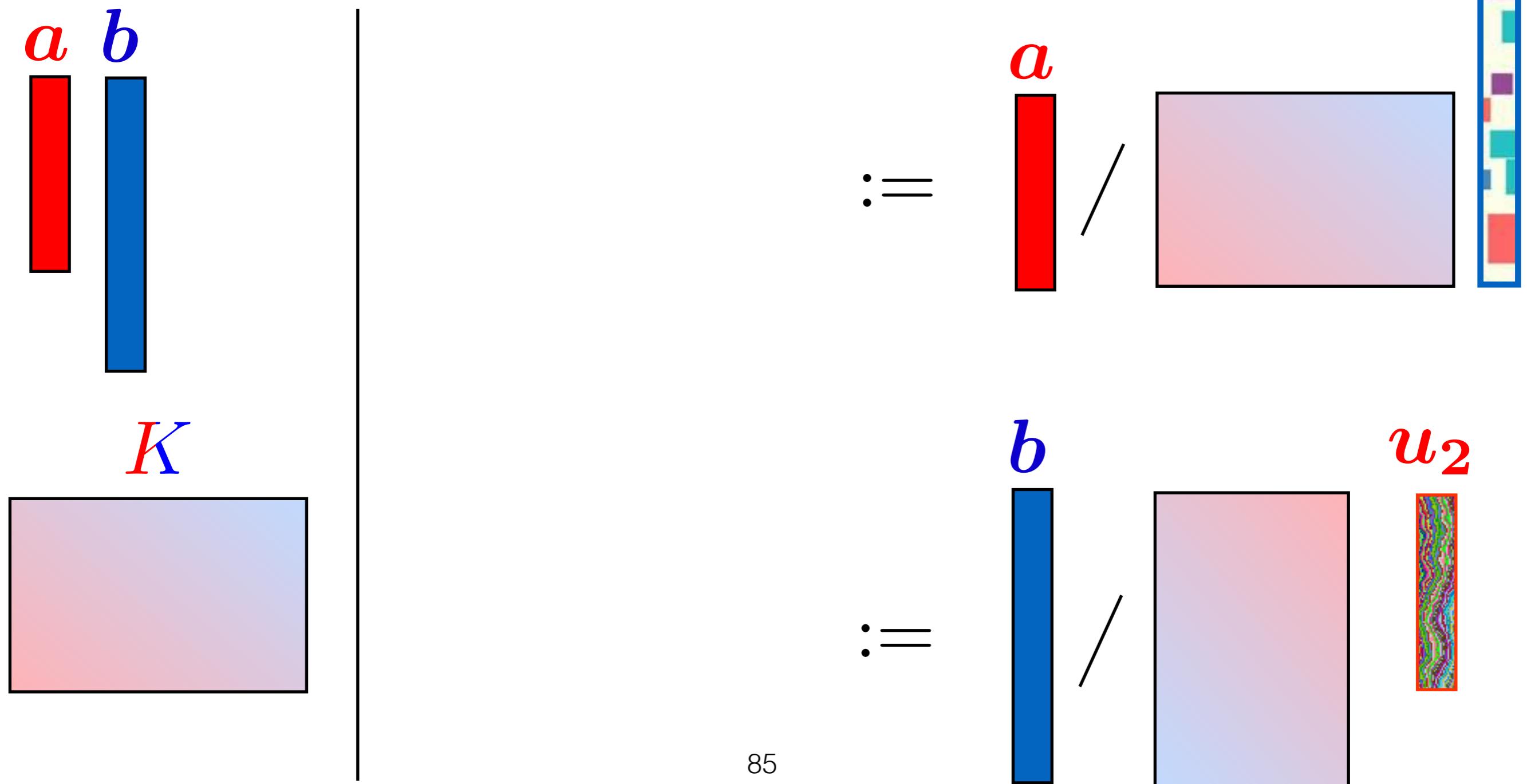
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

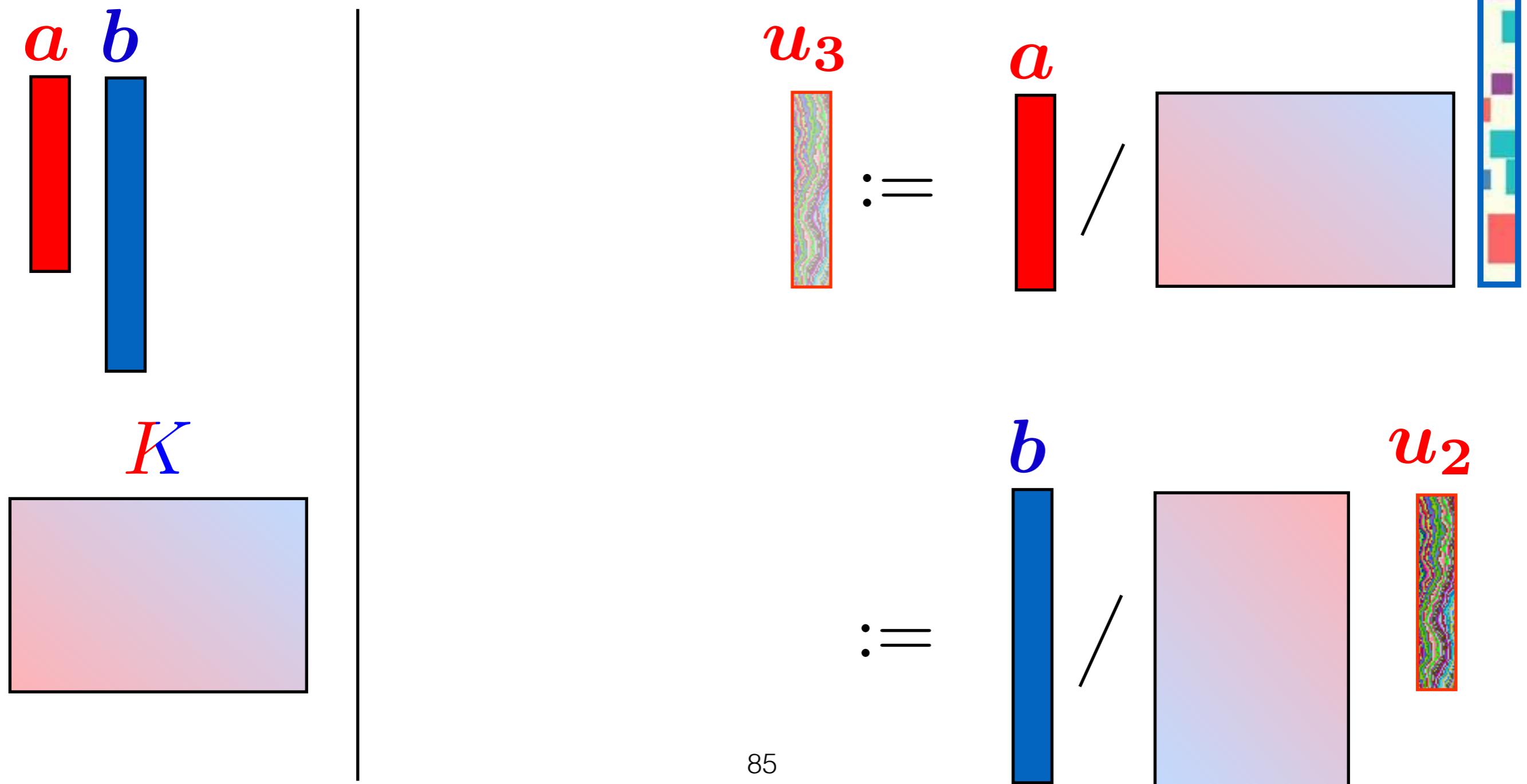
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

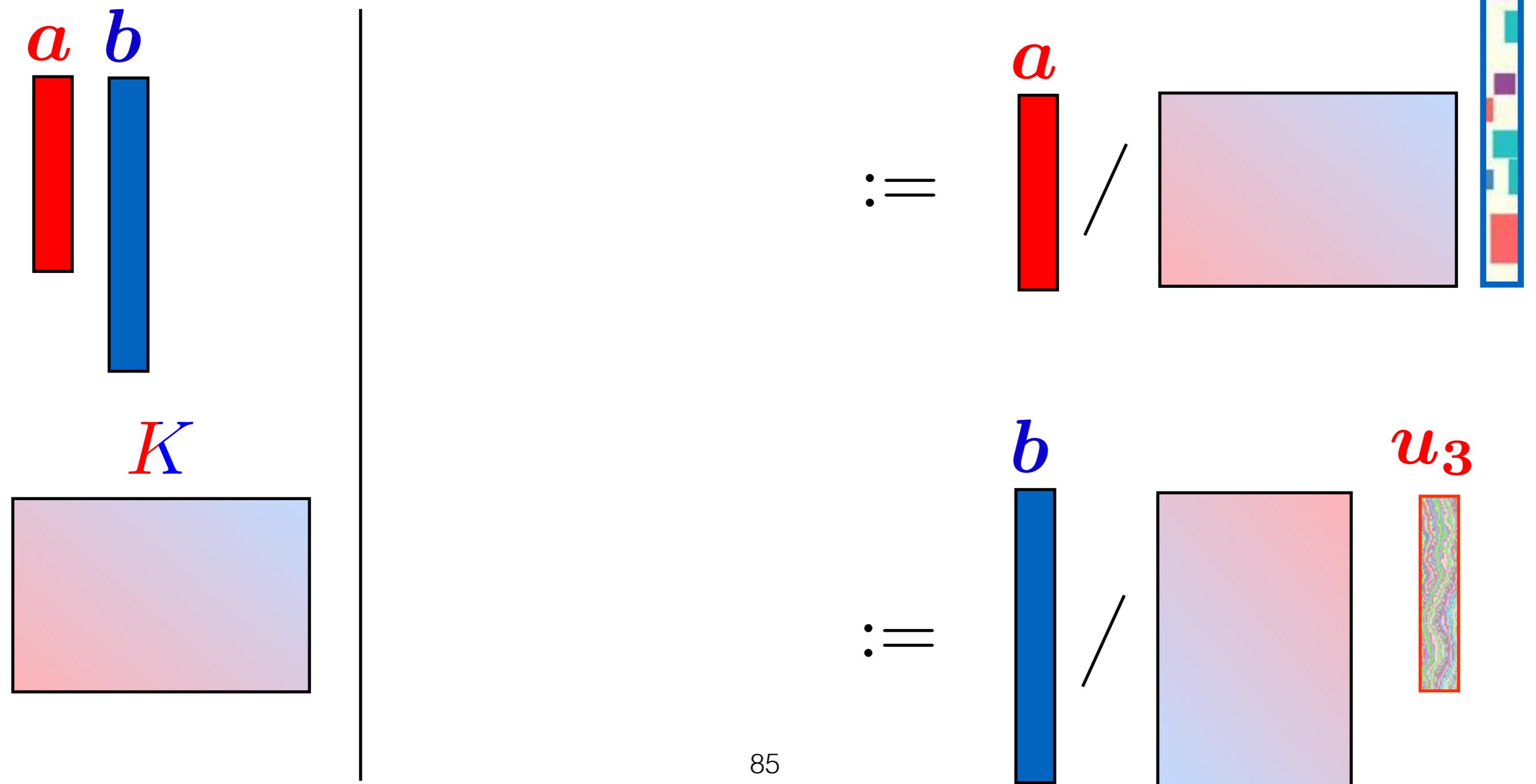
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

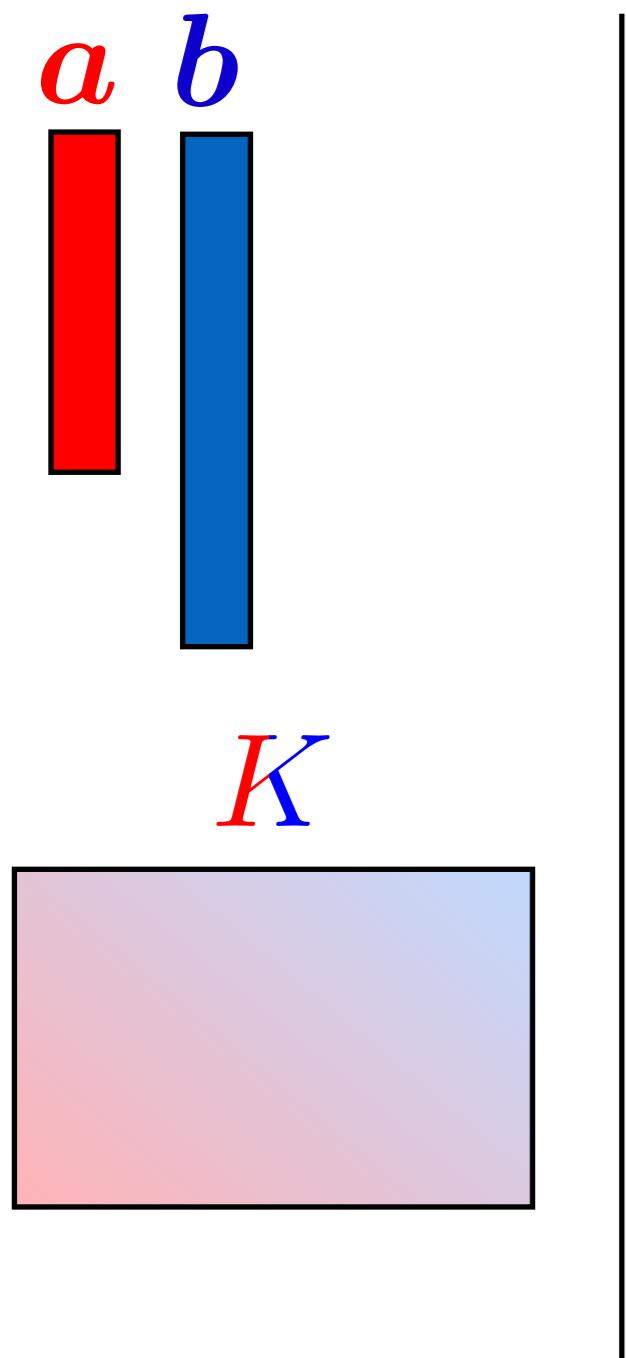
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

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



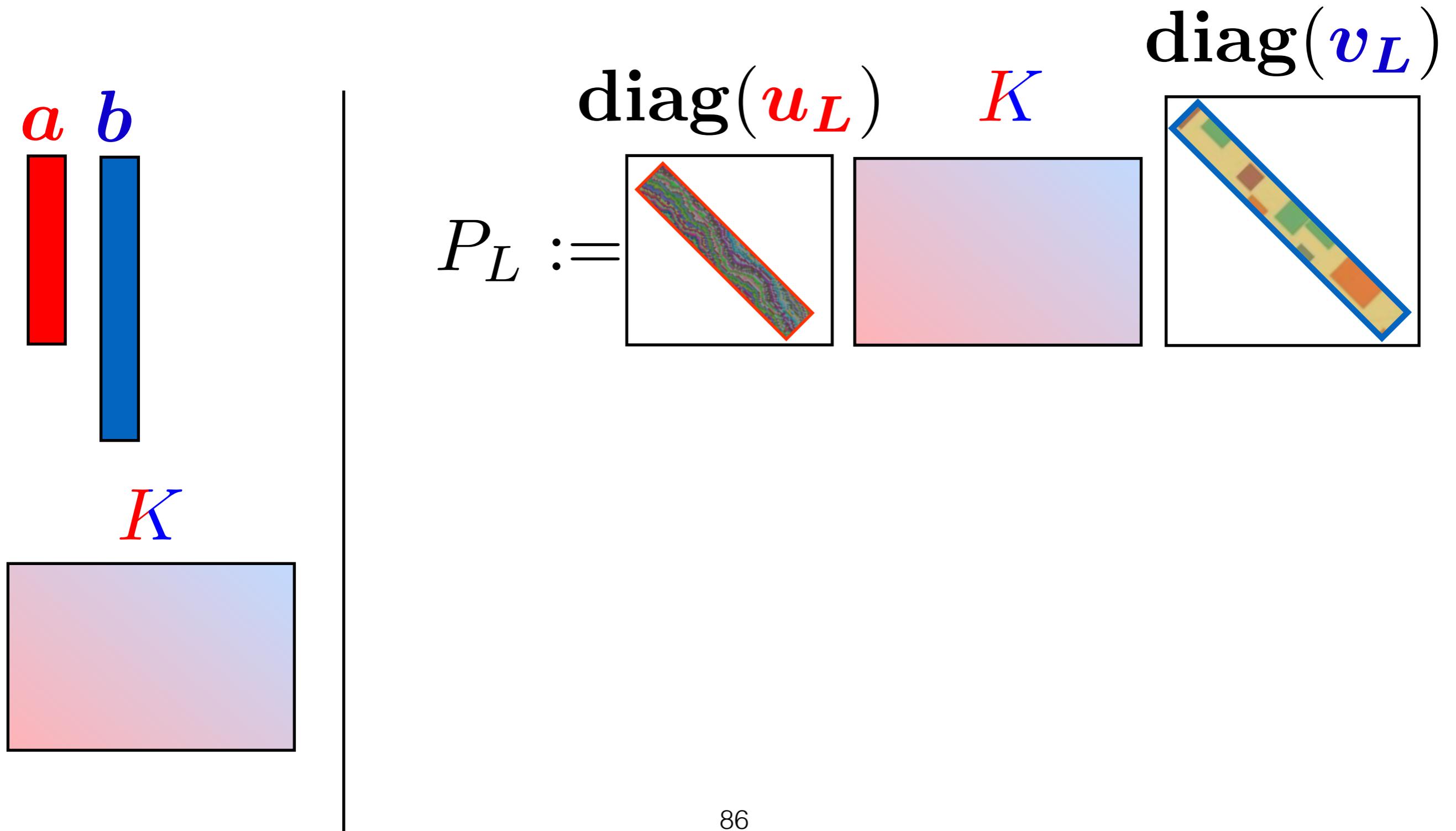
$$:= \mathbf{a} / \begin{array}{c} \text{---} \\ \text{---} \end{array} \mathbf{v}_2$$

etc. . .

$$:= \mathbf{b} / \begin{array}{c} \text{---} \\ \text{---} \end{array} \mathbf{u}_3$$

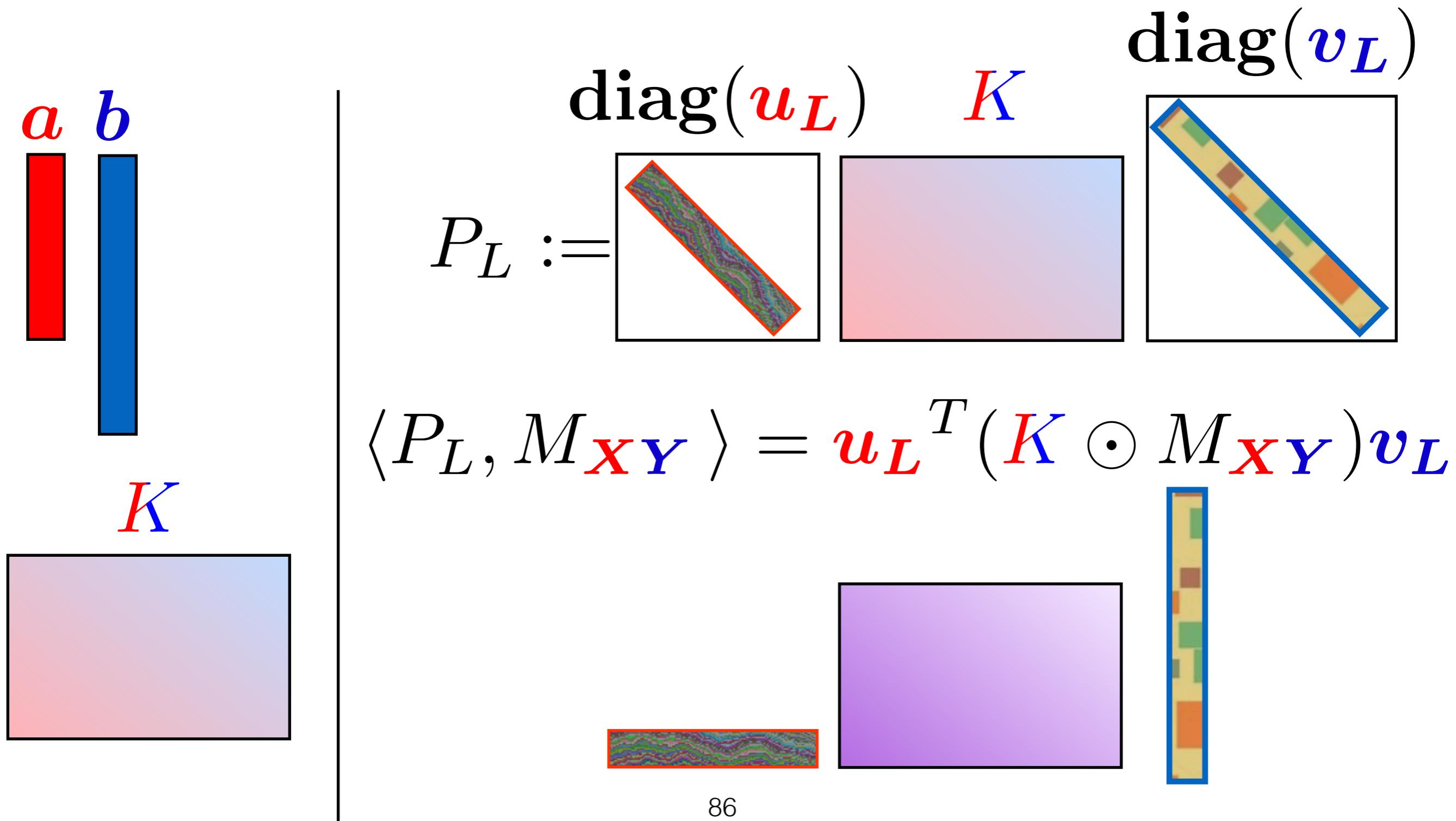
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



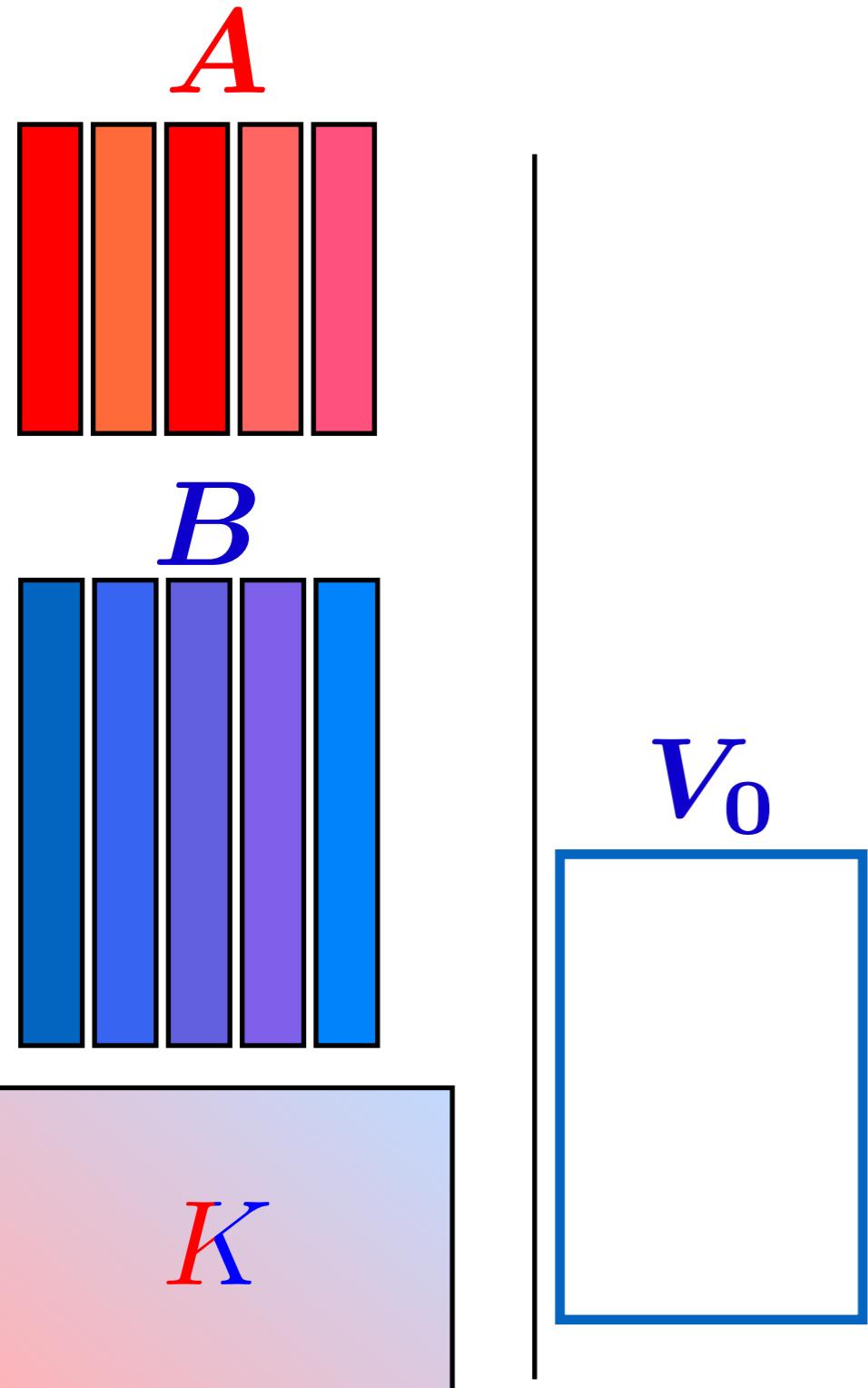
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



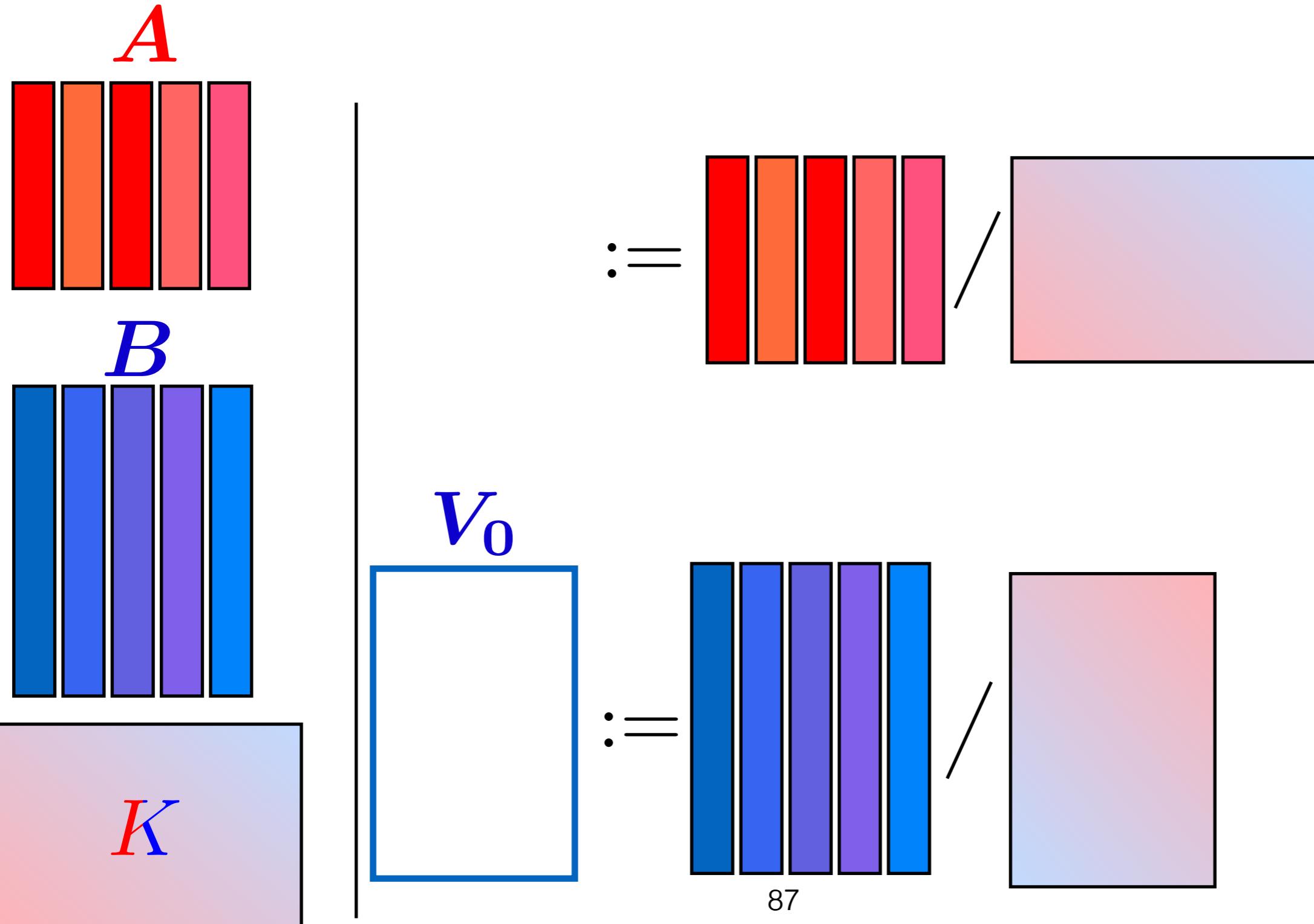
Also embarrassingly parallel

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



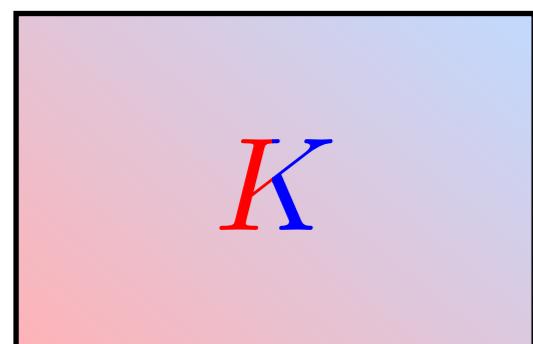
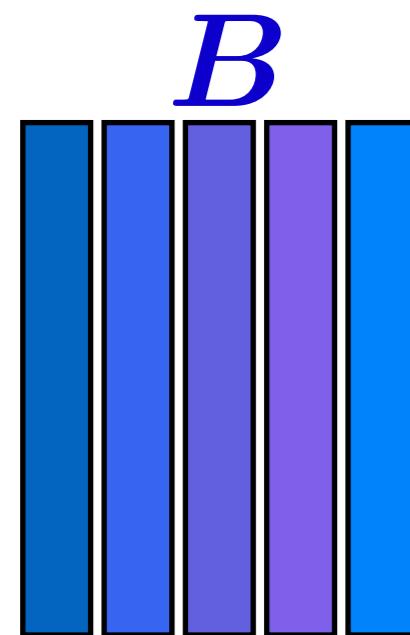
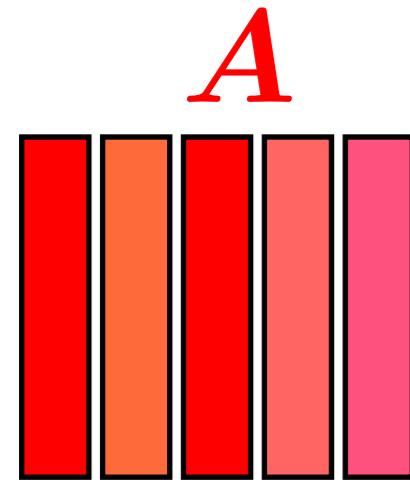
Also embarrassingly parallel

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



Also embarrassingly parallel

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

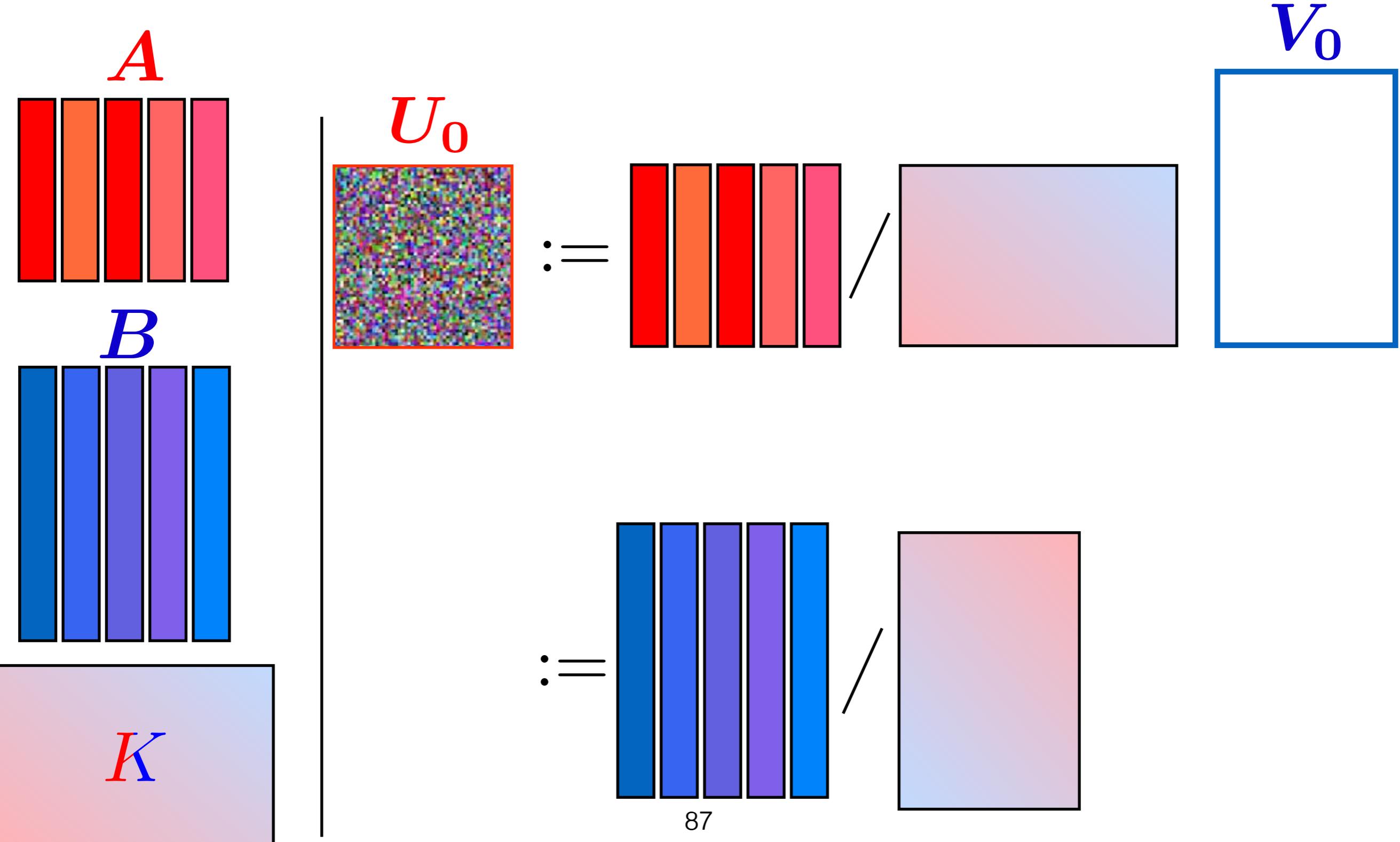


$$V_0 := \begin{bmatrix} \text{red} \\ \text{orange} \\ \text{red} \\ \text{pink} \\ \text{red} \end{bmatrix} \begin{bmatrix} \text{pink} & \text{blue} \\ \text{blue} & \text{pink} \end{bmatrix}$$

$$V_1 := \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{purple} \\ \text{purple} \\ \text{blue} \end{bmatrix} \begin{bmatrix} \text{pink} & \text{blue} \\ \text{blue} & \text{pink} \end{bmatrix}$$

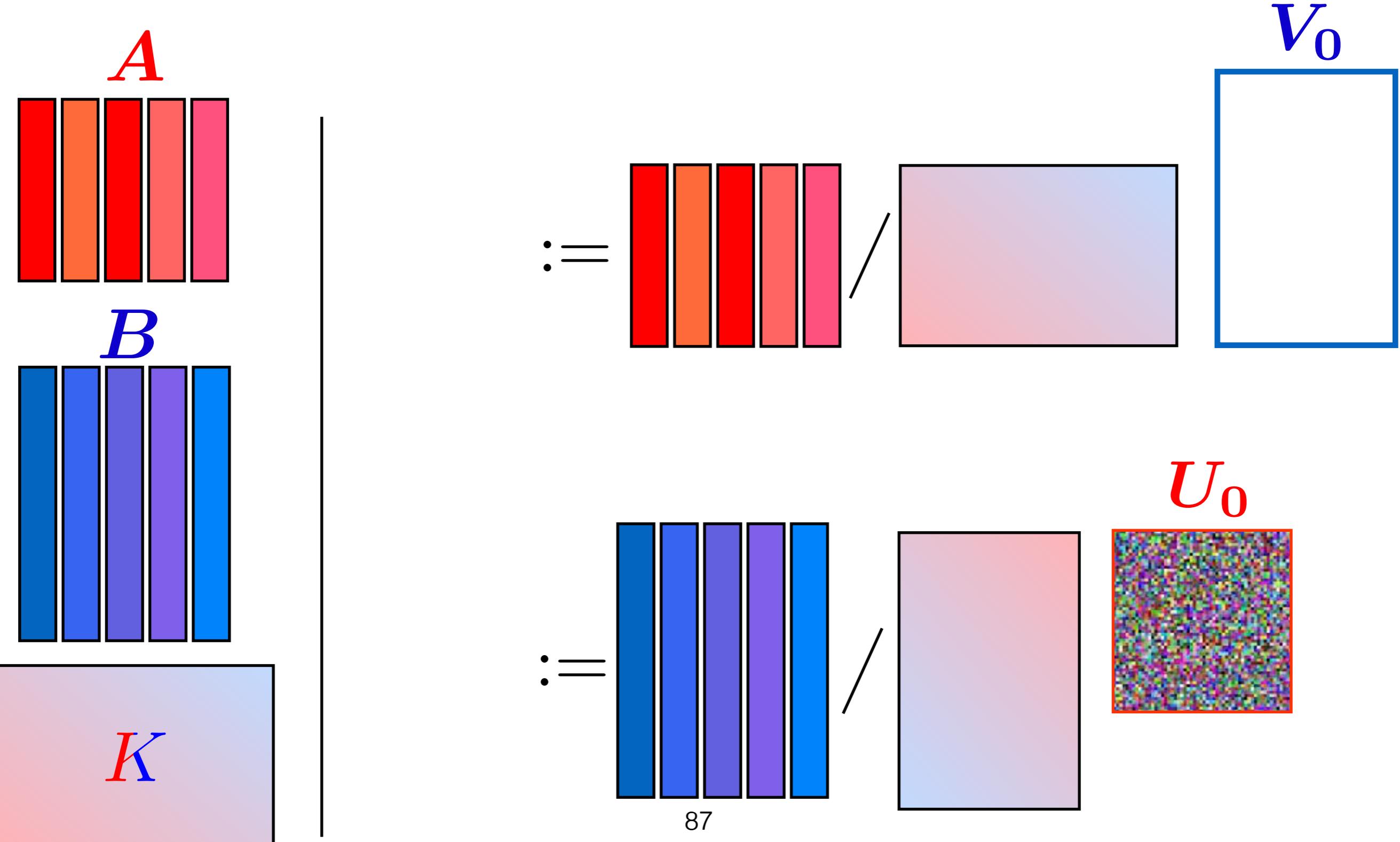
Also embarrassingly parallel

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



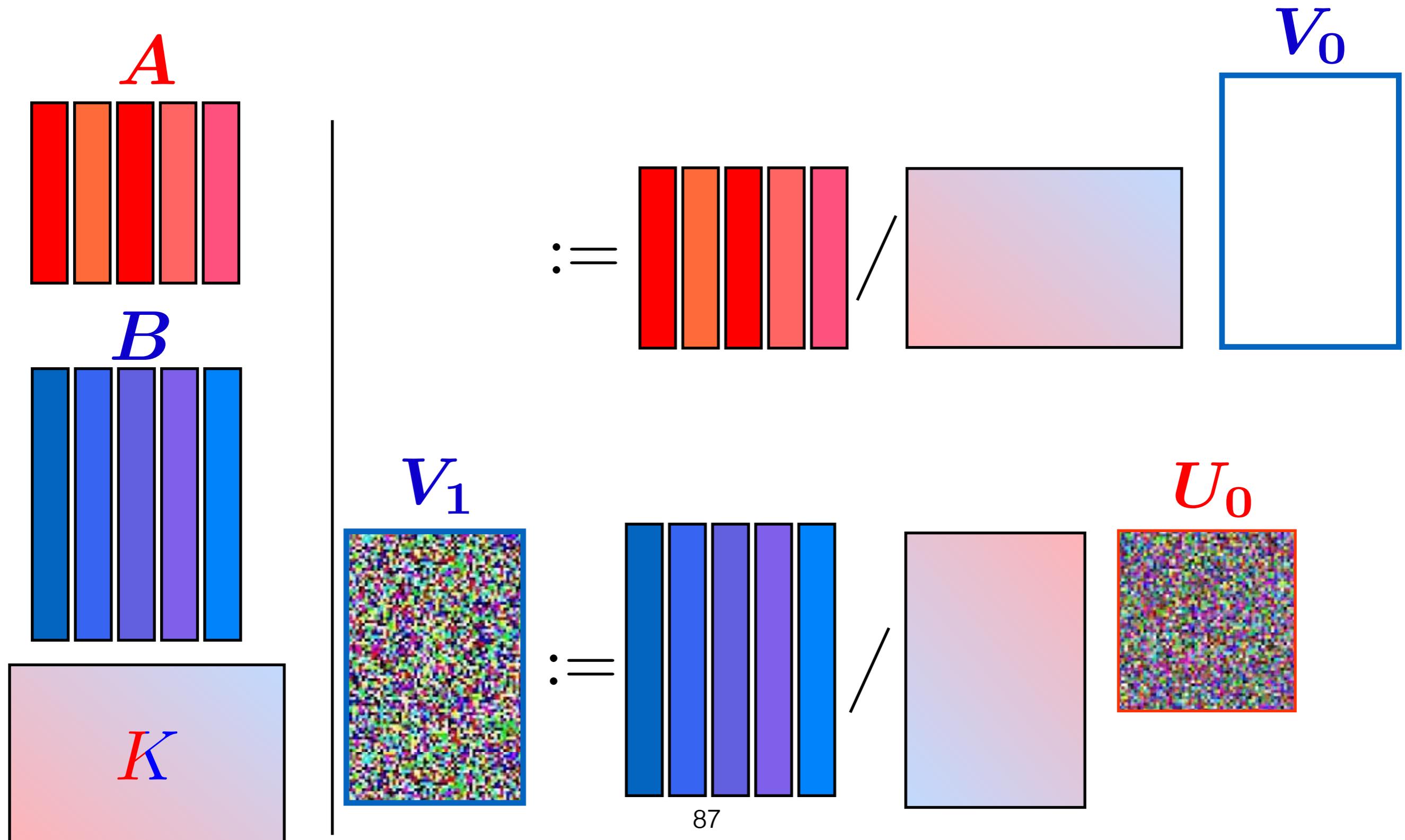
Also embarrassingly parallel

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



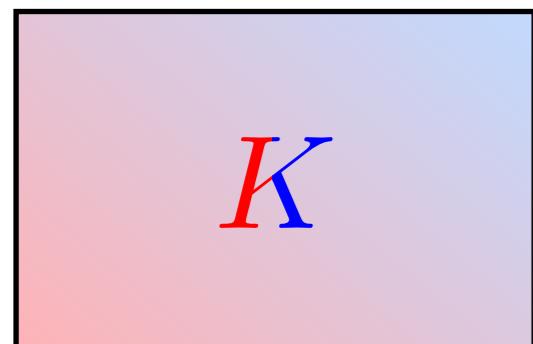
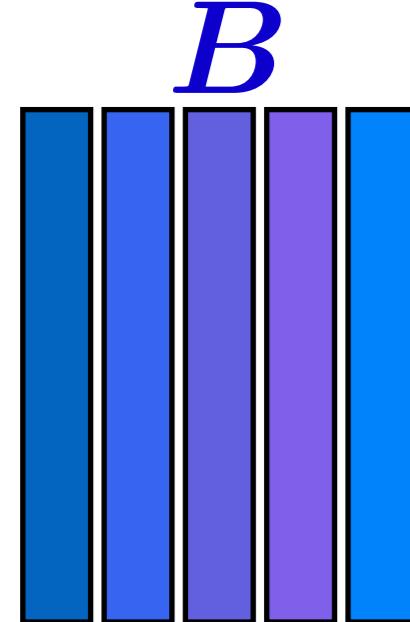
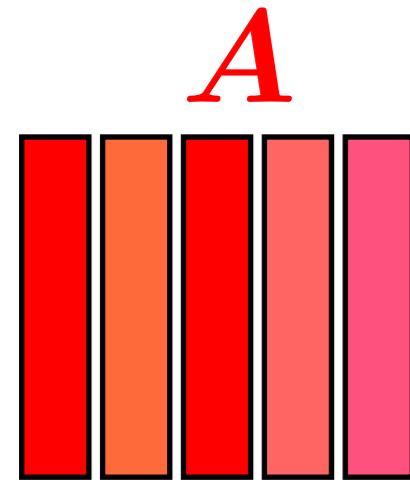
Also embarrassingly parallel

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



Also embarrassingly parallel

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



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

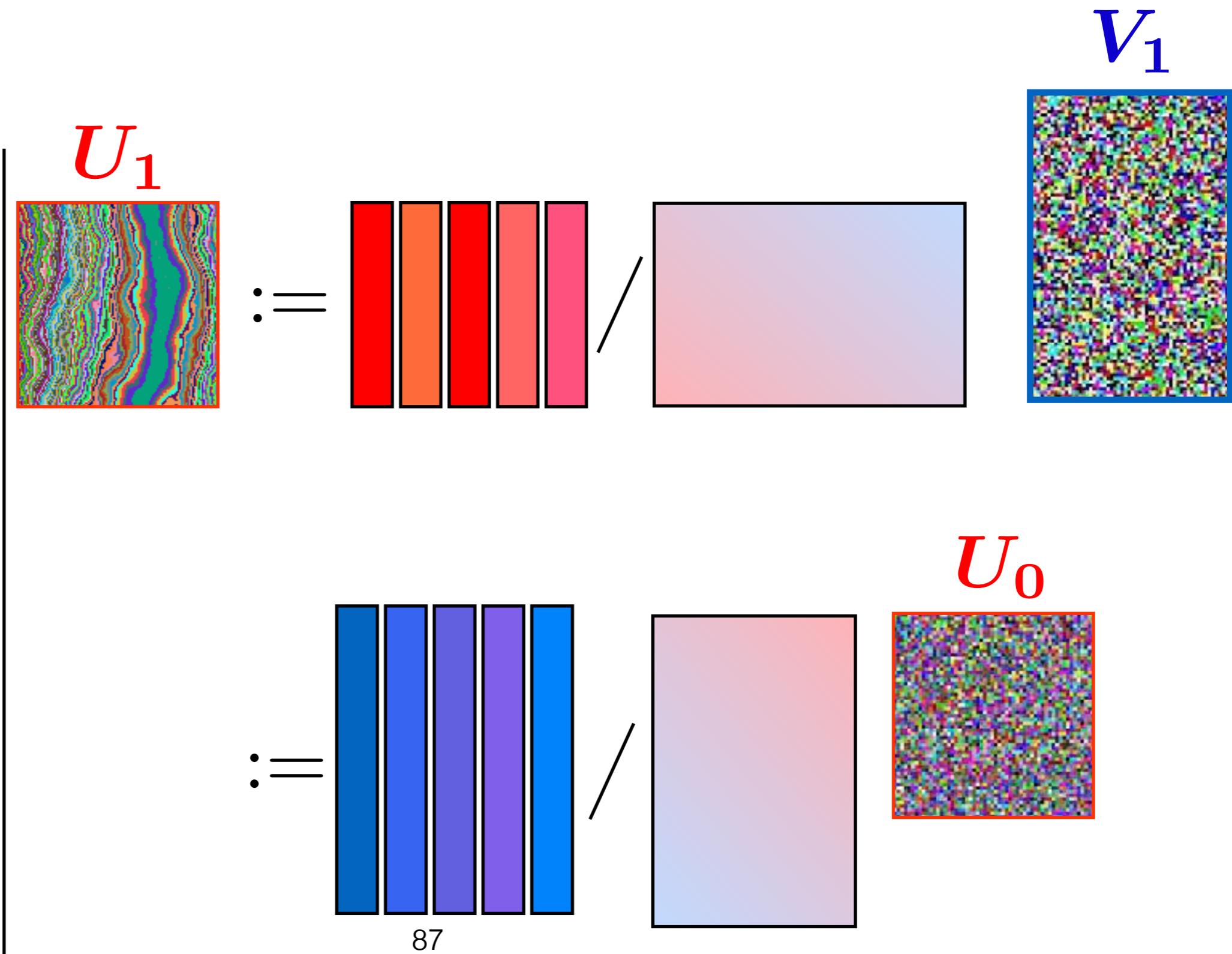
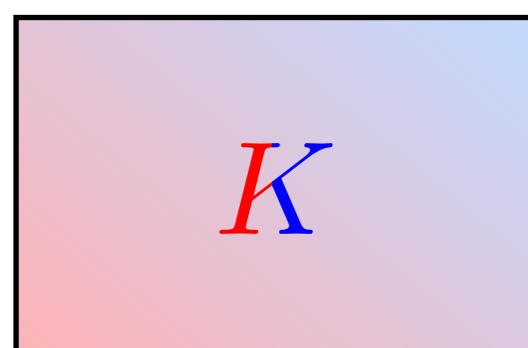
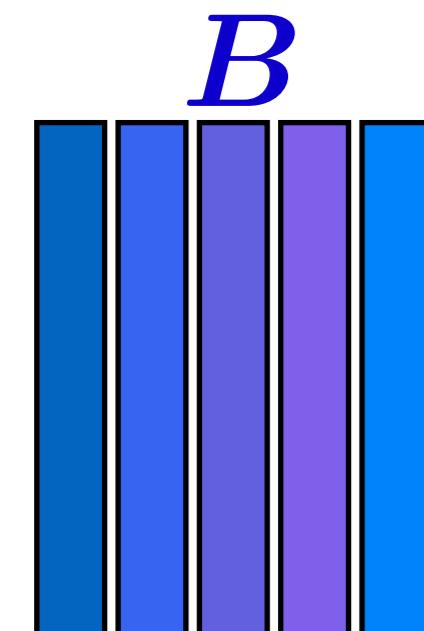
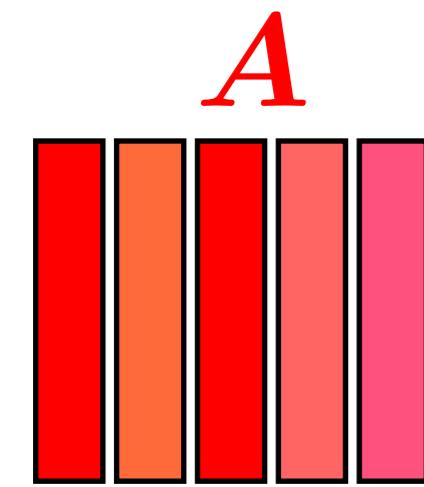


$$:= \begin{array}{c} \text{blue bar} \\ \text{blue bar} \\ \text{purple bar} \\ \text{purple bar} \\ \text{blue bar} \end{array}$$



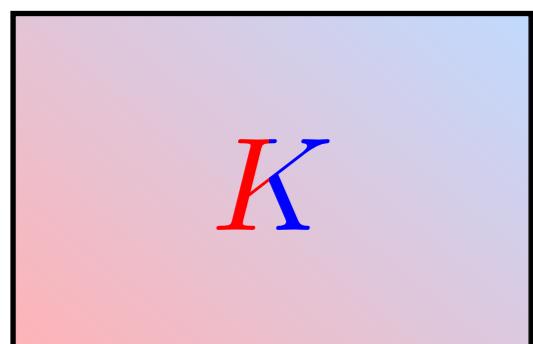
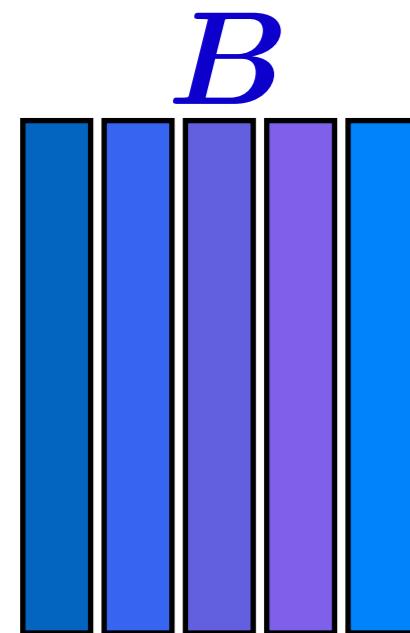
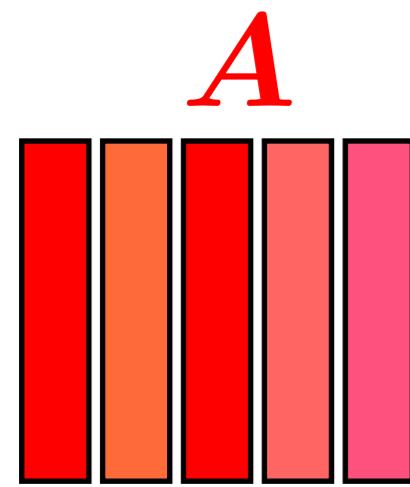
Also embarrassingly parallel

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

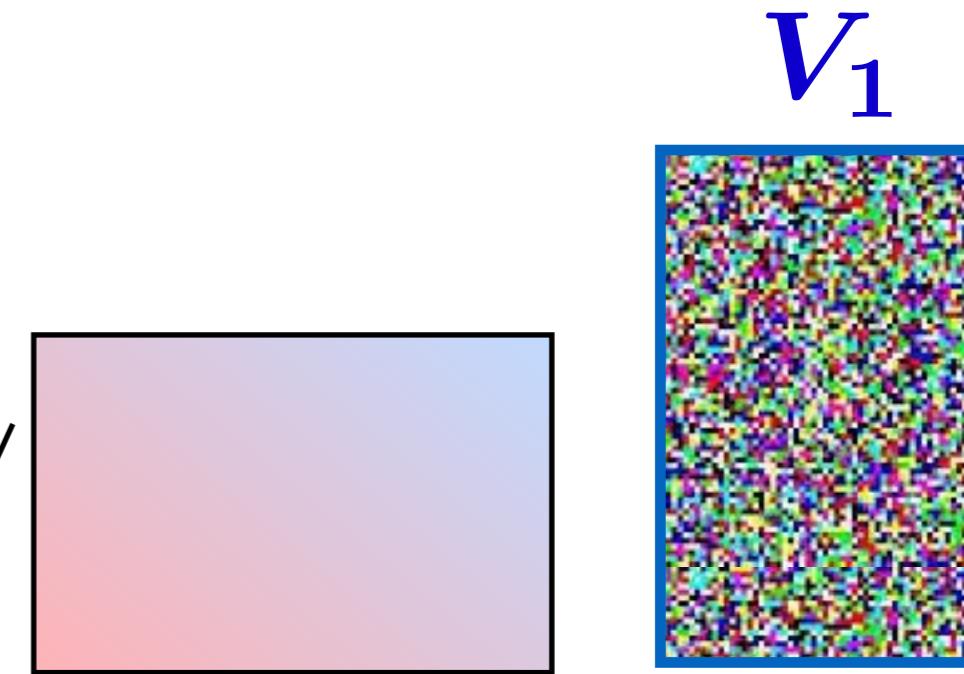


Also embarrassingly parallel

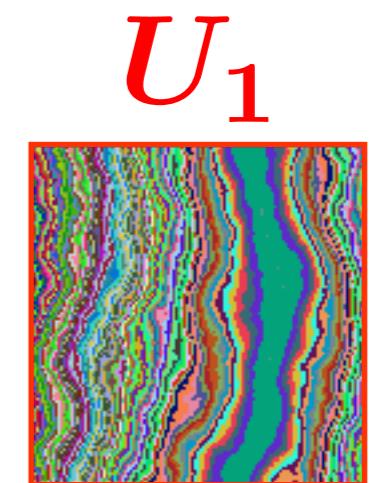
- [Sinkhorn'64] with *matrix* fixed-point iterations



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

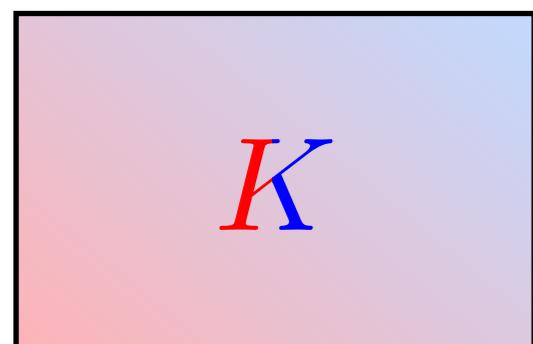
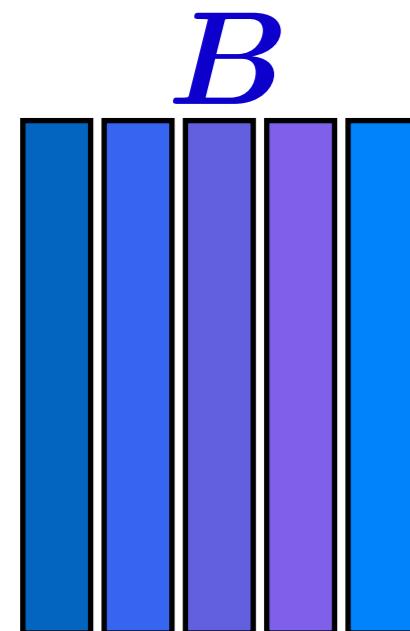
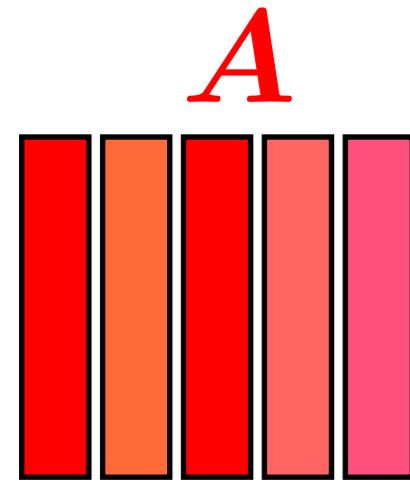


$$:= \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c}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



Also embarrassingly parallel

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



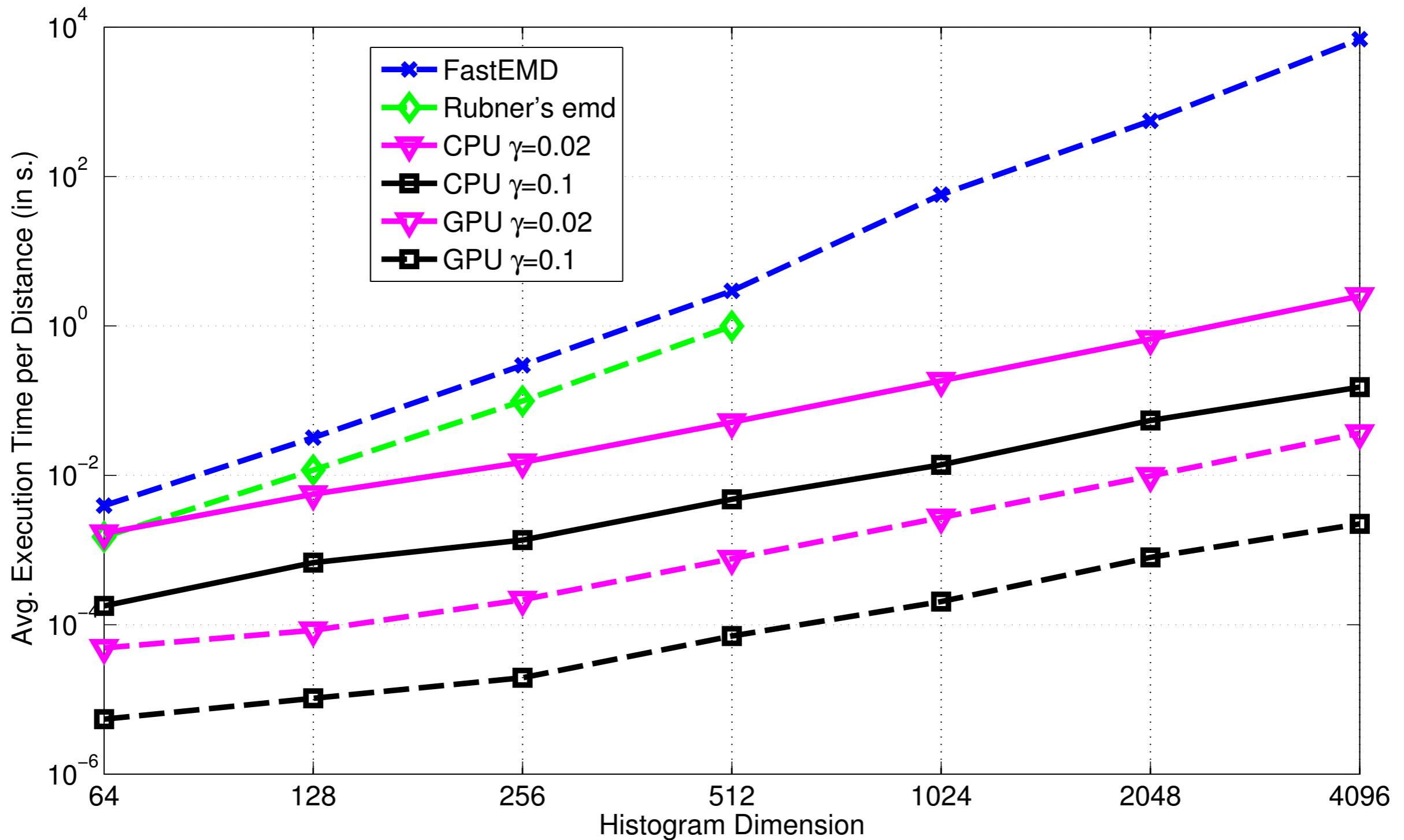
$$:= \begin{array}{c} \text{red bar} \\ \text{orange bar} \\ \text{red bar} \\ \text{pink bar} \\ \text{red bar} \end{array} \quad \begin{array}{c} \text{pink gradient} \\ \text{blue gradient} \end{array}$$

etc. . . .

$$:= \begin{array}{c} \text{blue bar} \\ \text{blue bar} \\ \text{purple bar} \\ \text{purple bar} \\ \text{blue bar} \end{array} \quad \begin{array}{c} \text{pink gradient} \\ \text{blue gradient} \end{array}$$

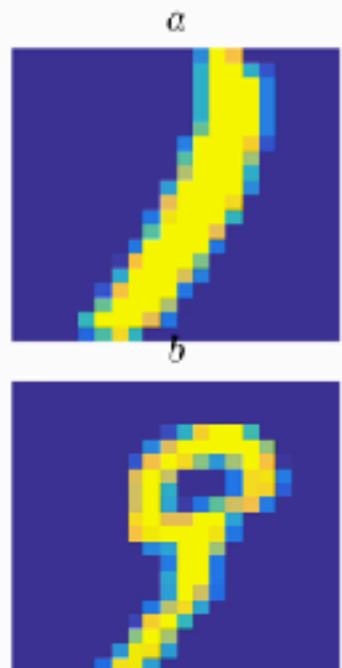


Very Fast EMD Approx. Solver

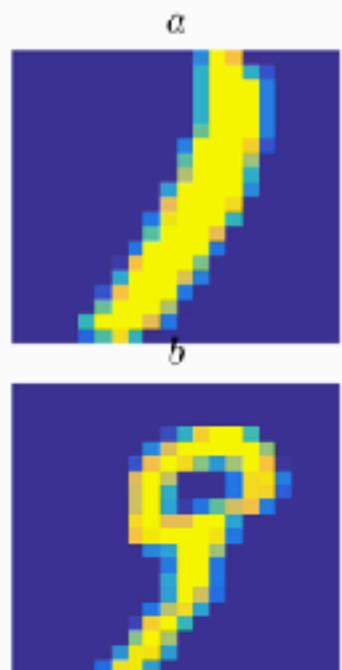


Note. (Ω, \mathcal{D}) is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance 10^{-2} .

Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver



Sinkhorn as a Dual Algorithm

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

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

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

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

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

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

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

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

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

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

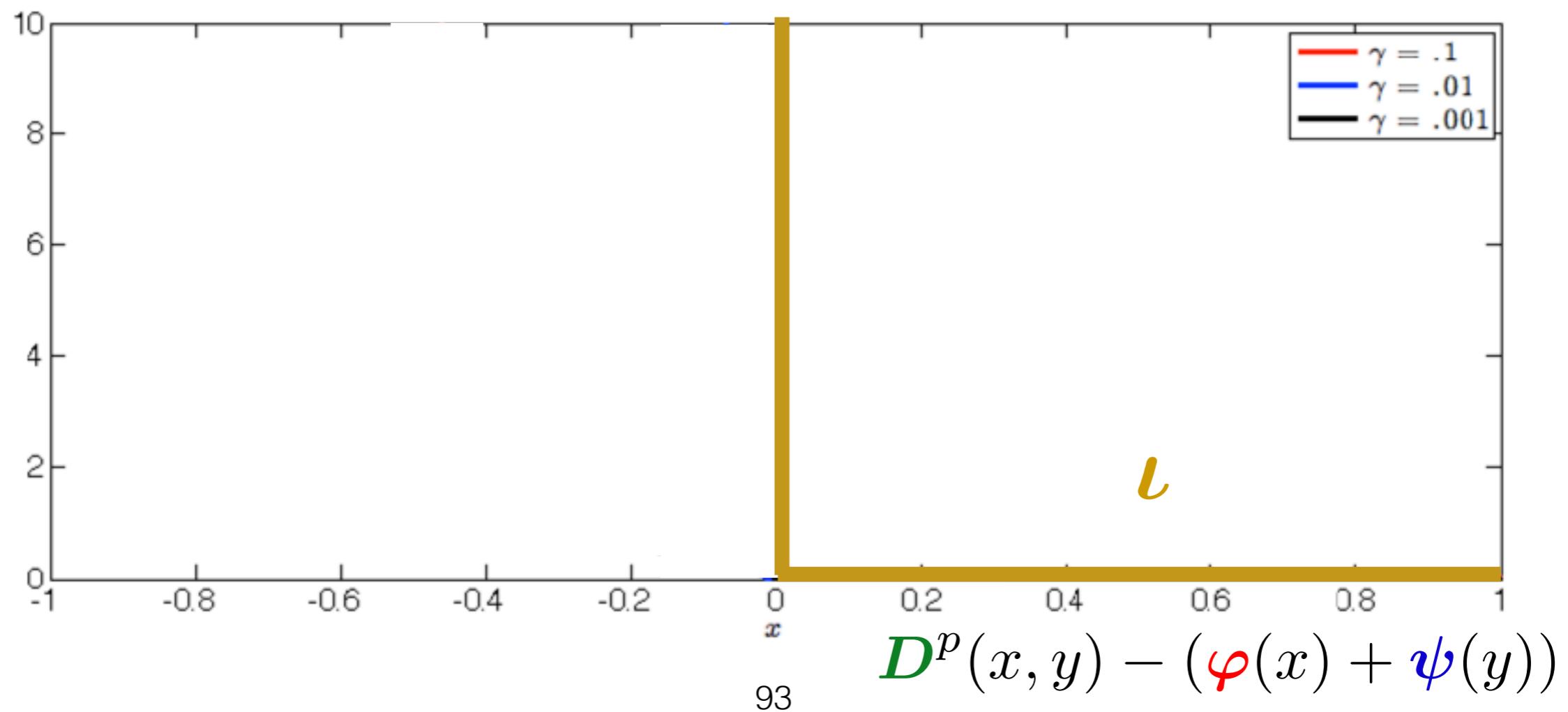
$$\boldsymbol{\beta} \leftarrow \gamma \left(\log \mathbf{b} - \log \mathbf{K}^T (e^{\boldsymbol{\alpha}/\gamma}) \right)$$

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

Stochastic Formulation

$$W_p^p(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C(\varphi, \psi)$$
$$C = \{(\varphi, \psi) \mid \forall x, y, \varphi(x) + \psi(y) \leq D(x, y)^p\}$$

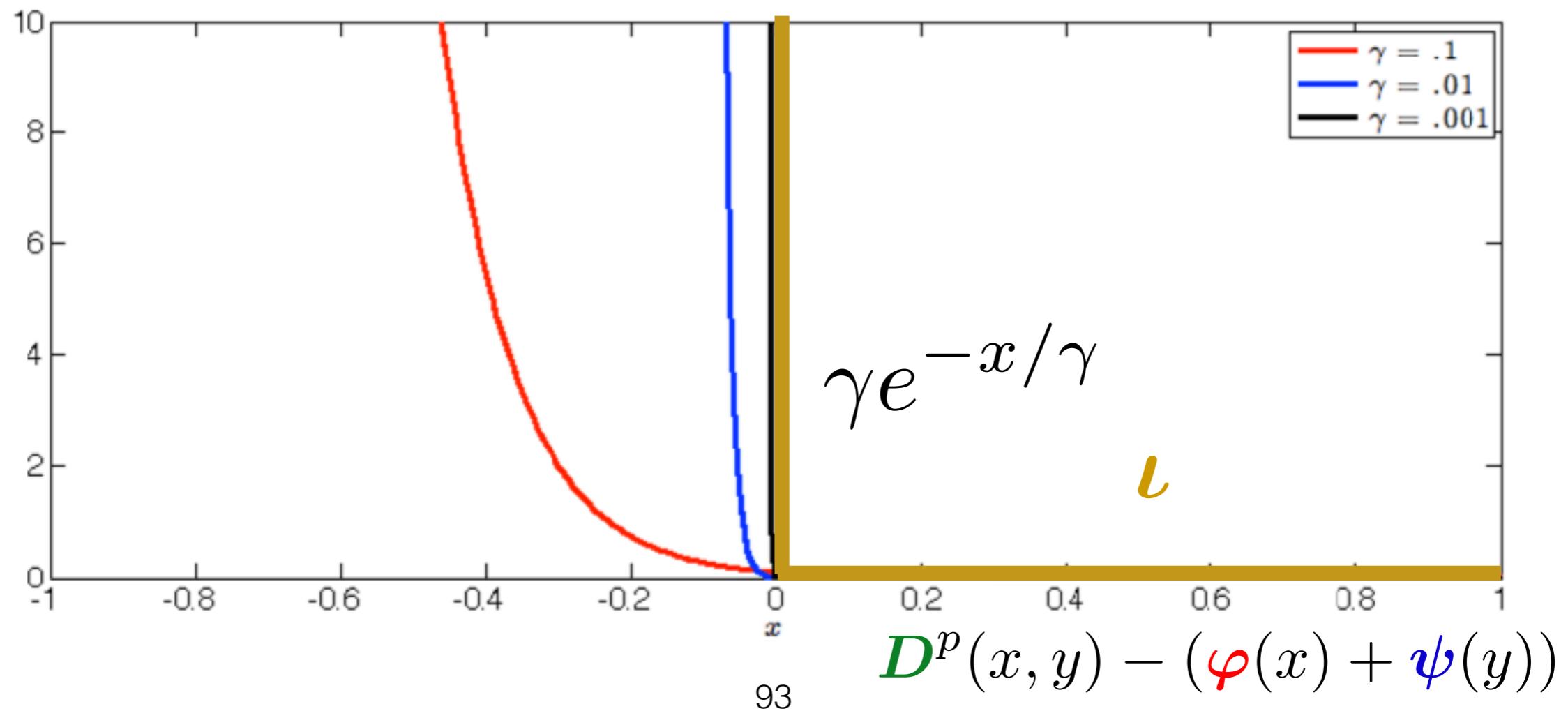
DUAL



Stochastic Formulation

$$W_p^p(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C(\varphi, \psi)$$
$$C = \{(\varphi, \psi) \mid \forall x, y, \varphi(x) + \psi(y) \leq D(x, y)^p\}$$

DUAL



Stochastic Formulation

$$W_p^p(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C(\varphi, \psi)$$

$$C = \{(\varphi, \psi) \mid \forall x, y, \varphi(x) + \psi(y) \leq D(x, y)^p\}$$

DUAL

regularizing dual  constraints $\gamma > 0$

$$W_\gamma(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C^\gamma(\varphi, \psi)$$

$$\iota_C^\gamma(\varphi, \psi) = \gamma \iint e^{(\varphi \oplus \psi - D^p)/\gamma} d\mu d\nu$$

REGULARIZED DUAL

Stochastic Formulation

$$W_p^p(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C(\varphi, \psi)$$

$$C = \{(\varphi, \psi) \mid \forall x, y, \varphi(x) + \psi(y) \leq D(x, y)^p\}$$

DUAL

regularizing dual  constraints $\gamma > 0$

$$W_{\gamma}(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi d\mu + \int \psi d\nu - \iota_C^{\gamma}(\varphi, \psi)$$

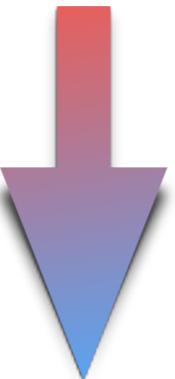
$$\iota_C^{\gamma}(\varphi, \psi) = \gamma \iint e^{(\varphi \oplus \psi - D^p)/\gamma} d\mu d\nu$$

REGULARIZED DUAL

Smoothed D transforms

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

SEMI-DUAL



$$\gamma > 0$$

$$W_\gamma(\mu, \nu) = \sup_{\varphi} \int \varphi d\mu + \int \varphi^{D, \gamma} d\nu.$$

$$\varphi^{D, \gamma} = -\gamma \log \int e^{\frac{\varphi(x) - D(x, \cdot)^p}{\gamma}} d\mu(x)$$

REGULARIZED SEMI-DUAL

Regularized Semidual Wasserstein

$$W_\gamma(\mu, \nu) = \sup_{\varphi} \int \varphi d\mu + \int \varphi^{D,\gamma} d\nu.$$
$$\varphi^{D,\gamma} = -\gamma \log \int e^{\frac{\varphi(x) - D(x, \cdot)^p}{\gamma}} d\mu(x)$$

REGULARIZED SEMI-DUAL

substituting

$$\sup_{\varphi} \int_y \left[\int_x \varphi(x) d\mu(x) - \gamma \log \int_x e^{\frac{\varphi(x) - D(x, y)^p}{\gamma}} d\mu(x) \right] d\nu(y).$$

REGULARIZED SEMI-DUAL

Stochastic Regularized Semidual

$$\sup_{\varphi} \int_y \left[\int_x \varphi(x) d\mu(x) - \gamma \log \int_x e^{\frac{\varphi(x) - D(x, y)^p}{\gamma}} d\mu(x) \right] d\nu(y).$$

REGULARIZED SEMI-DUAL

Stochastic Regularized Semidual

$$\sup_{\varphi} \int_y \left[\int_x \varphi(x) d\mu(x) - \gamma \log \int_x e^{\frac{\varphi(x) - D(x, y)^p}{\gamma}} d\mu(x) \right] d\nu(y).$$

REGULARIZED SEMI-DUAL

What if μ is a discrete measure?

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

$\varphi \in L_1(\mu)$ is now just a vector $\alpha \in \mathbb{R}^n$!

Stochastic Regularized Semidual

$$\sup_{\varphi} \int_y \left[\int_x \varphi(x) d\mu(x) - \gamma \log \int_x e^{\frac{\varphi(x) - D(x, y)^p}{\gamma}} d\mu(x) \right] d\nu(y).$$

REGULARIZED SEMI-DUAL

What if μ is a discrete measure?

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

$\varphi \in L_1(\mu)$ is now just a vector $\alpha \in \mathbb{R}^n$!

$$\sup_{\alpha \in \mathbb{R}^n} \int_y \left[\sum_{i=1}^n \alpha_i a_i - \gamma \log \sum_{i=1}^n e^{\frac{\alpha_i - D(x_i, y)^p}{\gamma}} a_i \right] d\nu(y)$$

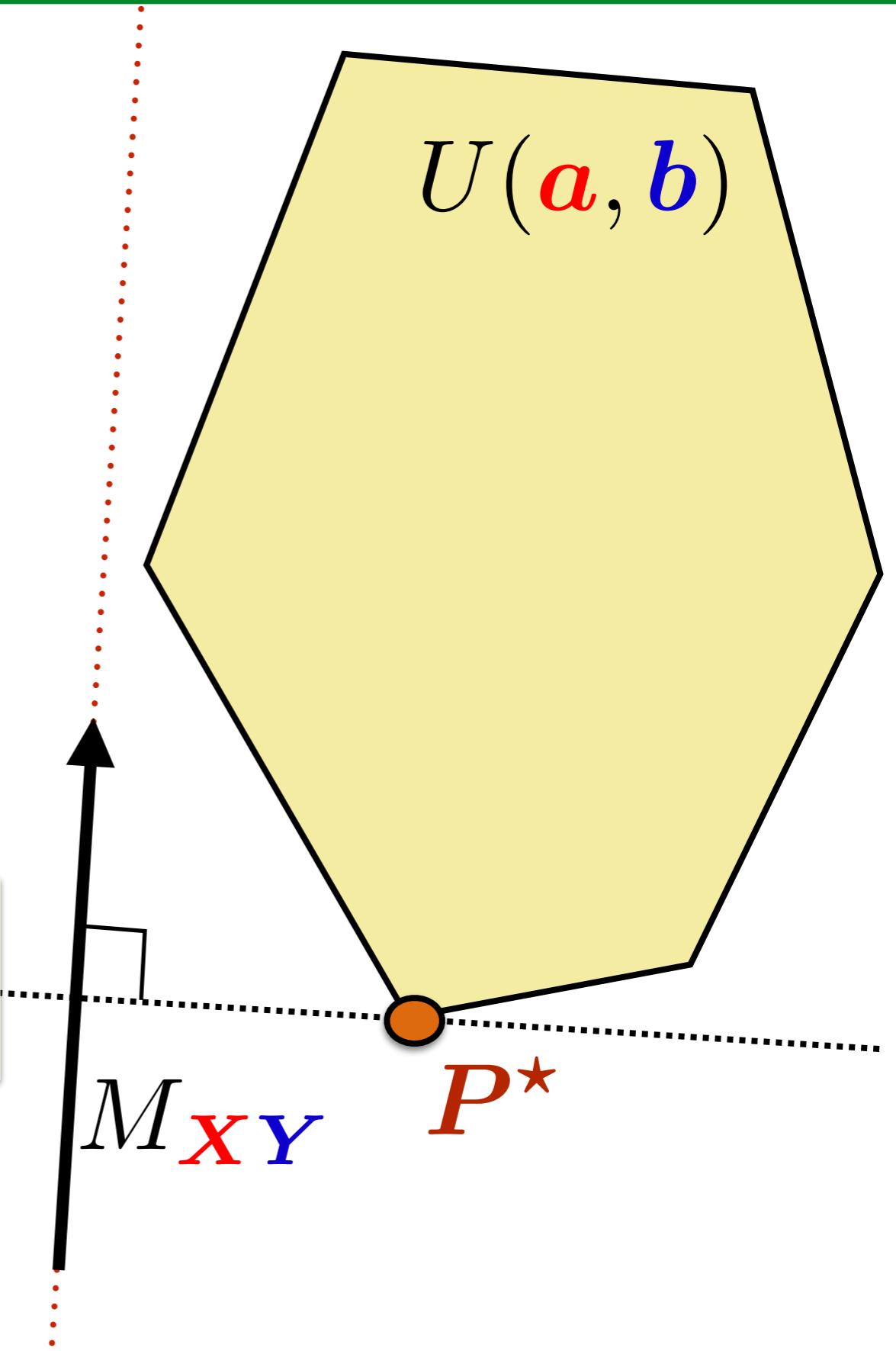
$$= \sup_{\alpha \in \mathbb{R}^n} \mathbb{E}_{\nu}[f(\alpha, y)]$$

STOCHASTIC REGULARIZED SEMI-DUAL

Sinkhorn in between W and MMD

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

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

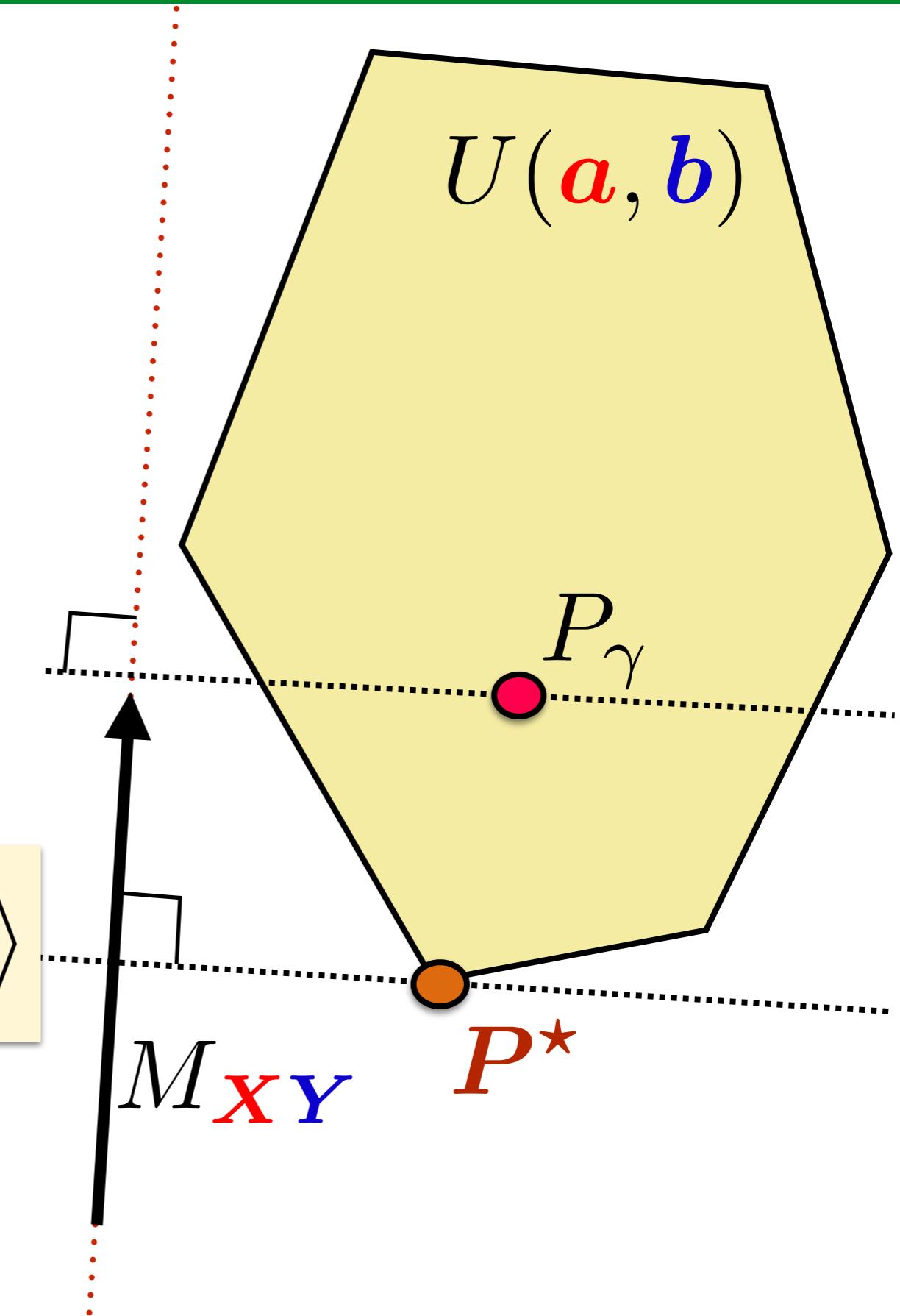


Sinkhorn in between W and MMD

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

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

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



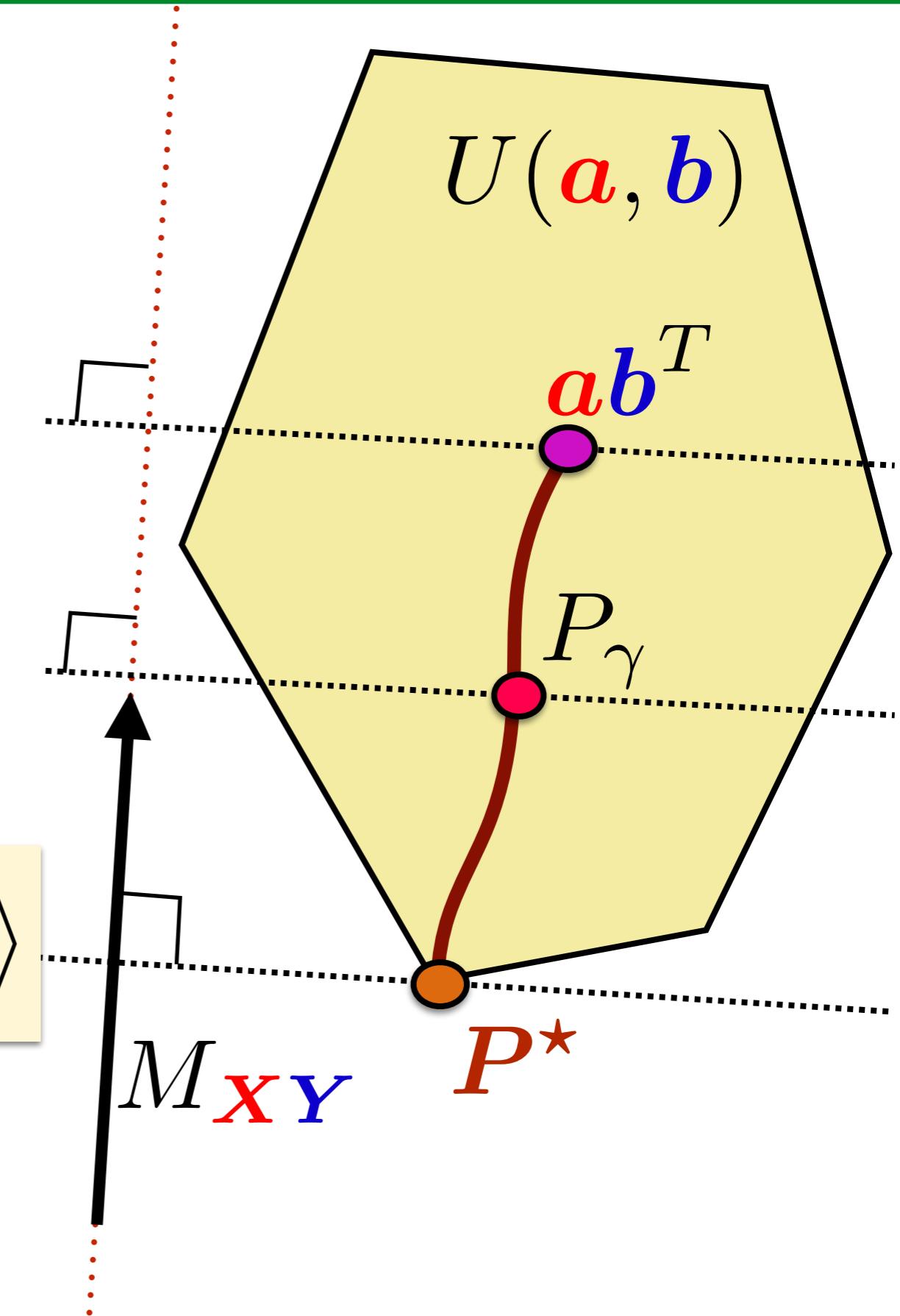
Sinkhorn in between W and MMD

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

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

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

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



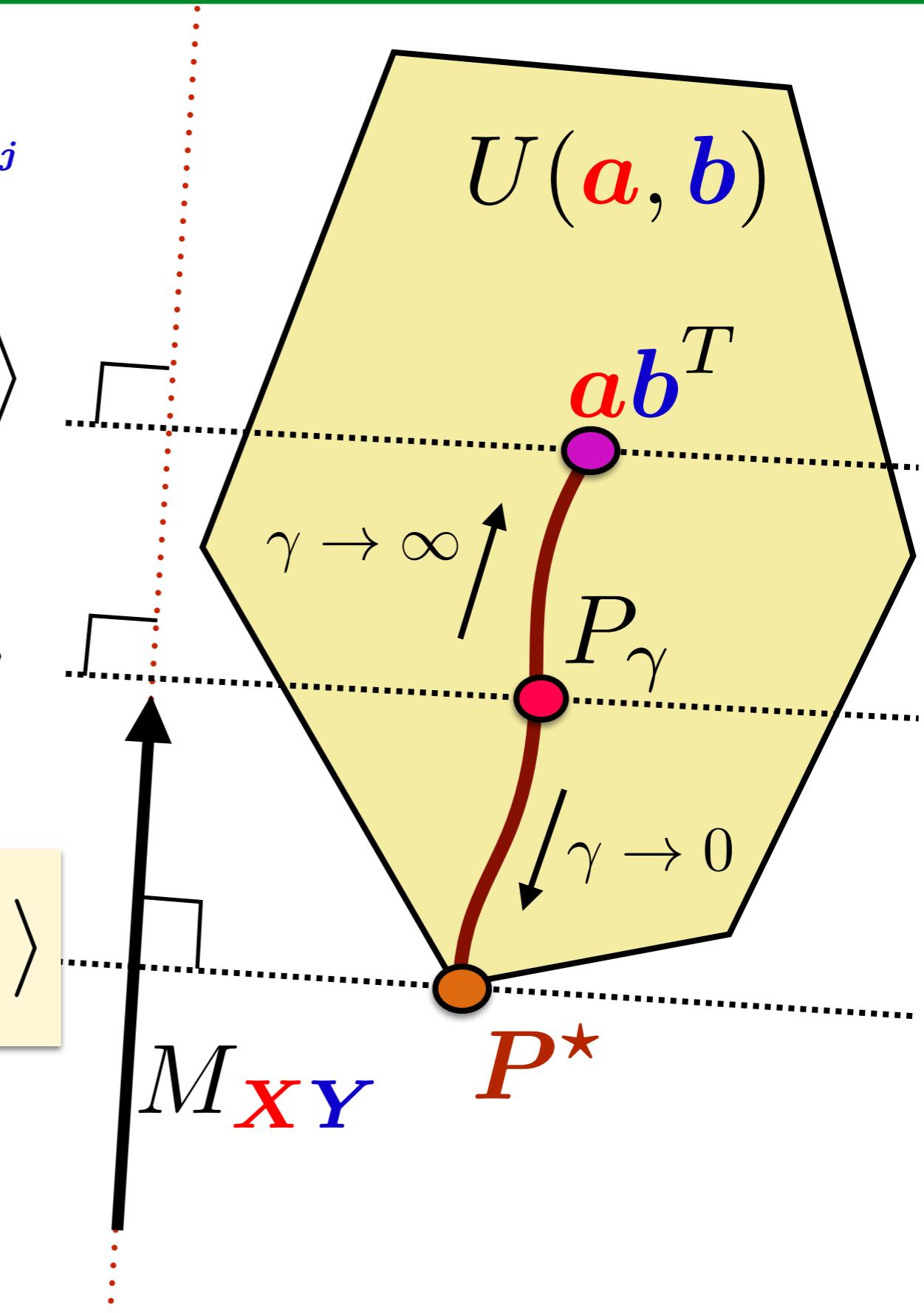
Sinkhorn in between W and MMD

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

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

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

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



Sinkhorn in between W and MMD

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

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

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

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

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

Sinkhorn in between W and MMD

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

$$\gamma \rightarrow \infty$$

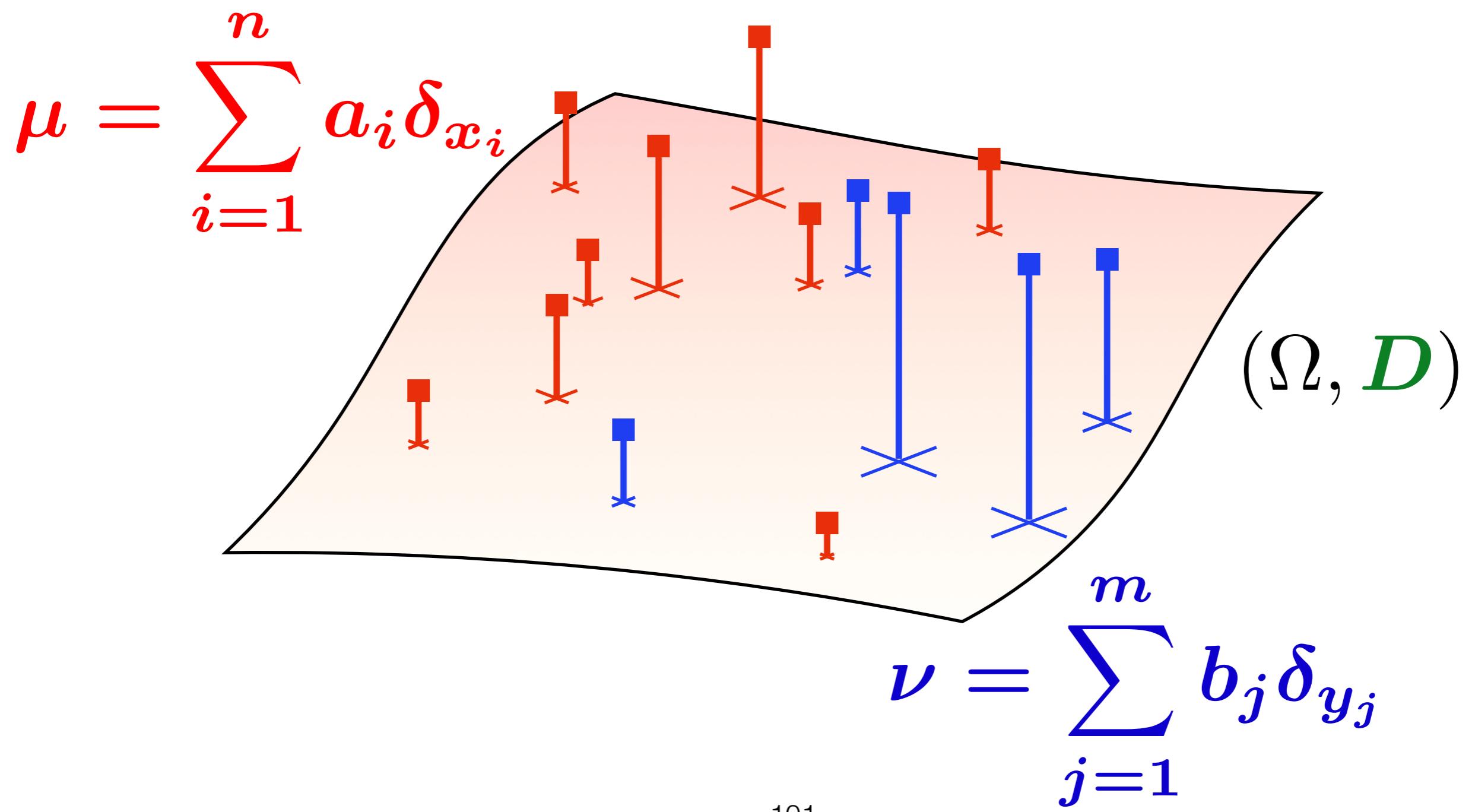

$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$$\gamma \rightarrow 0$$


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

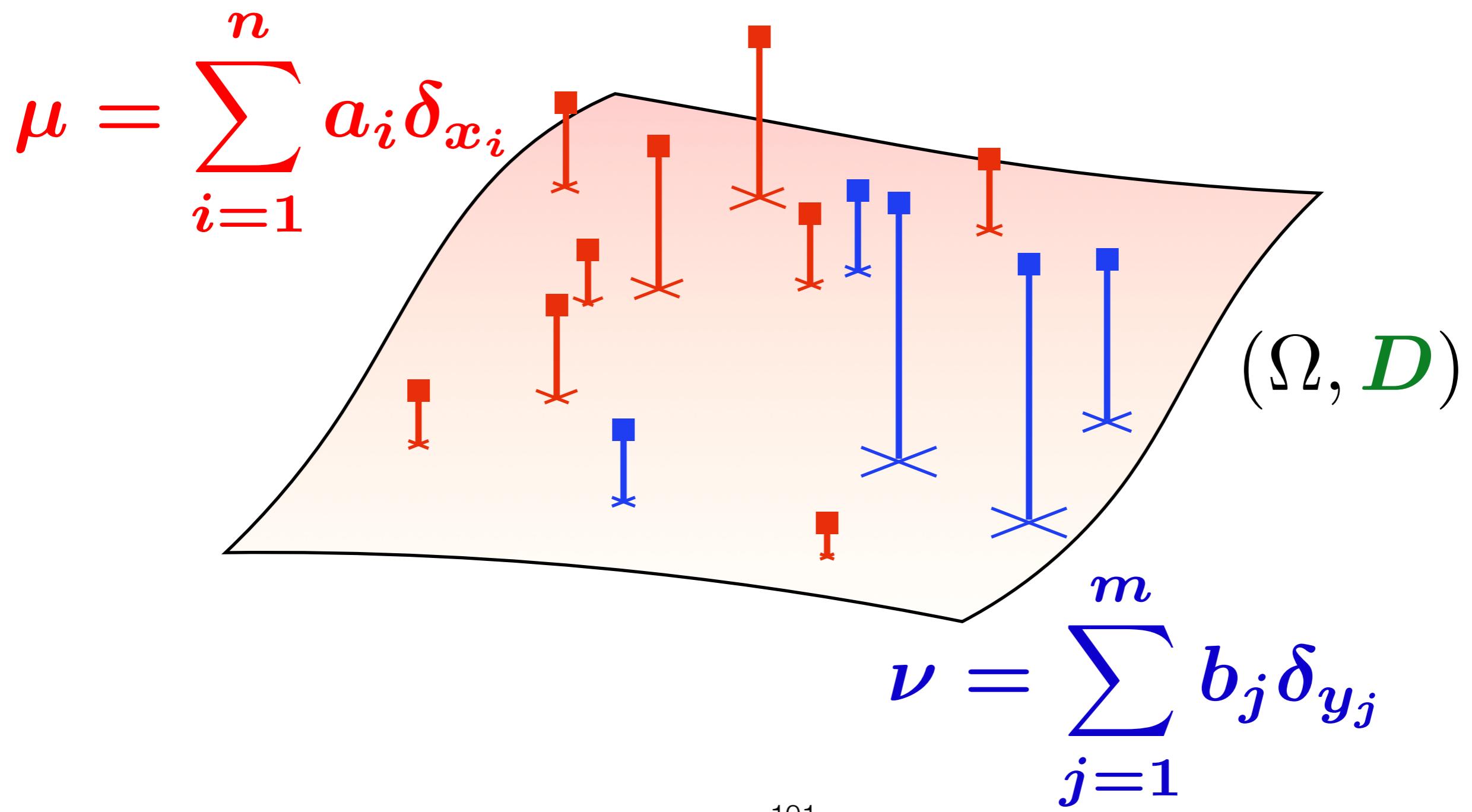
Differentiability of W

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



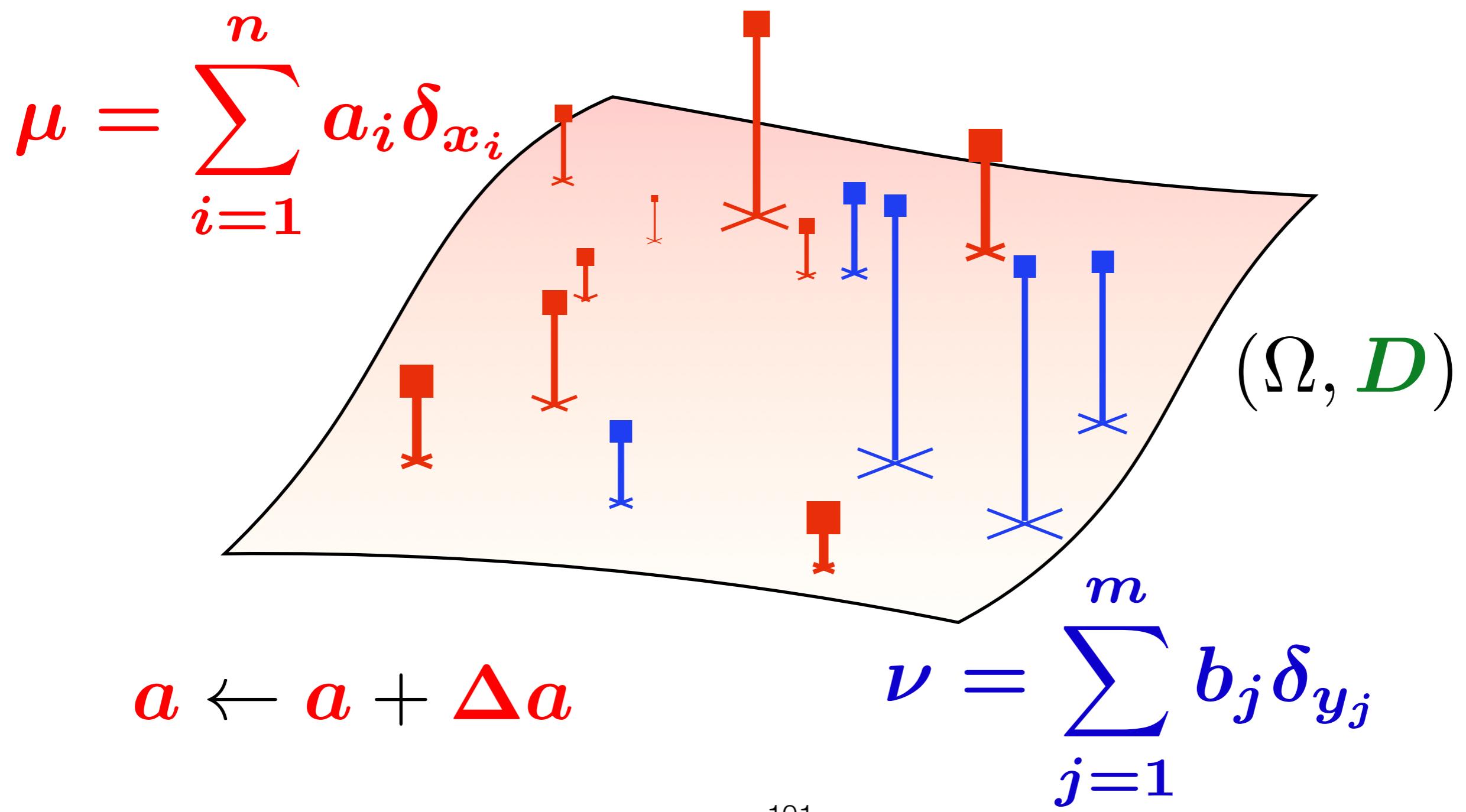
Differentiability of W

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



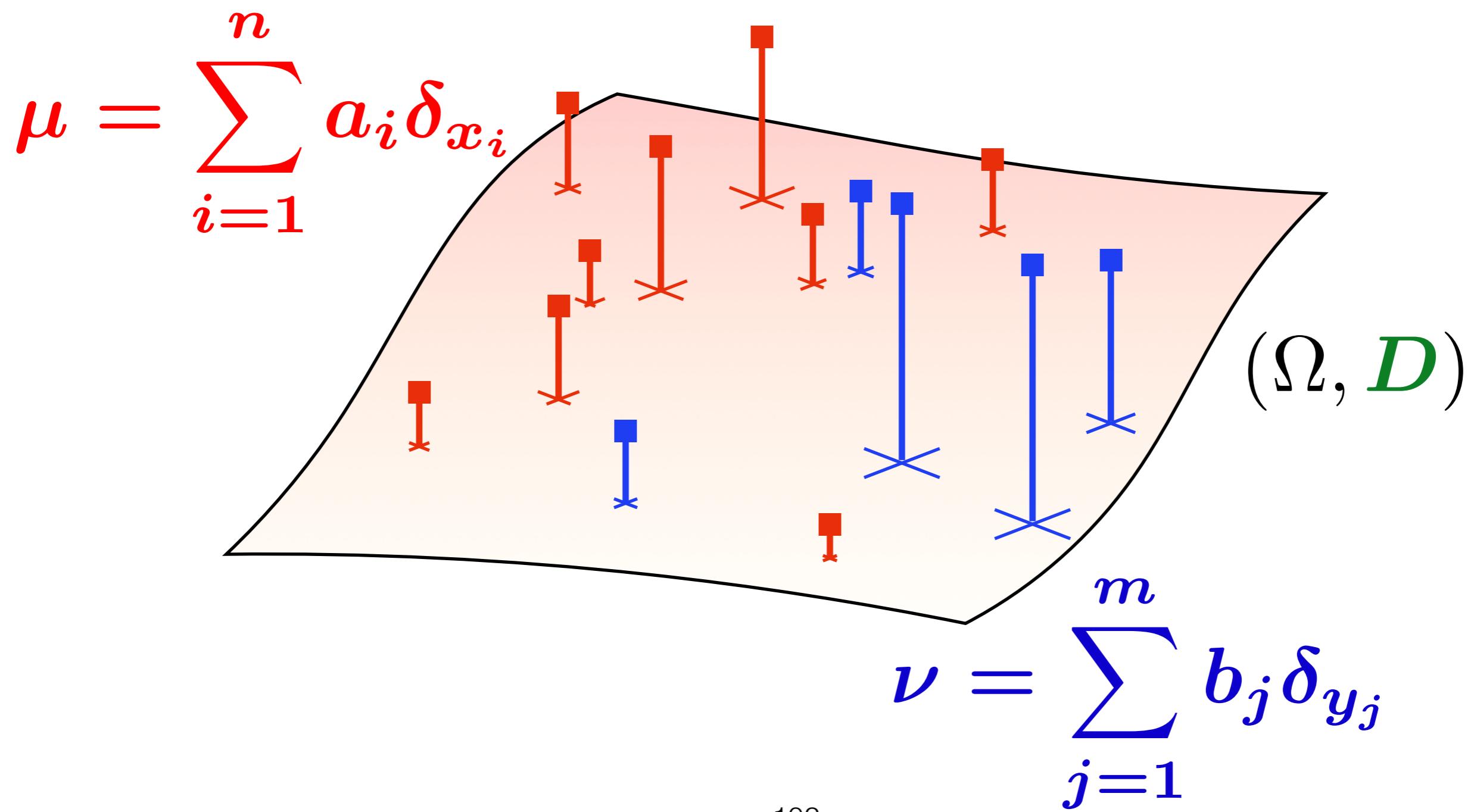
Differentiability of W

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



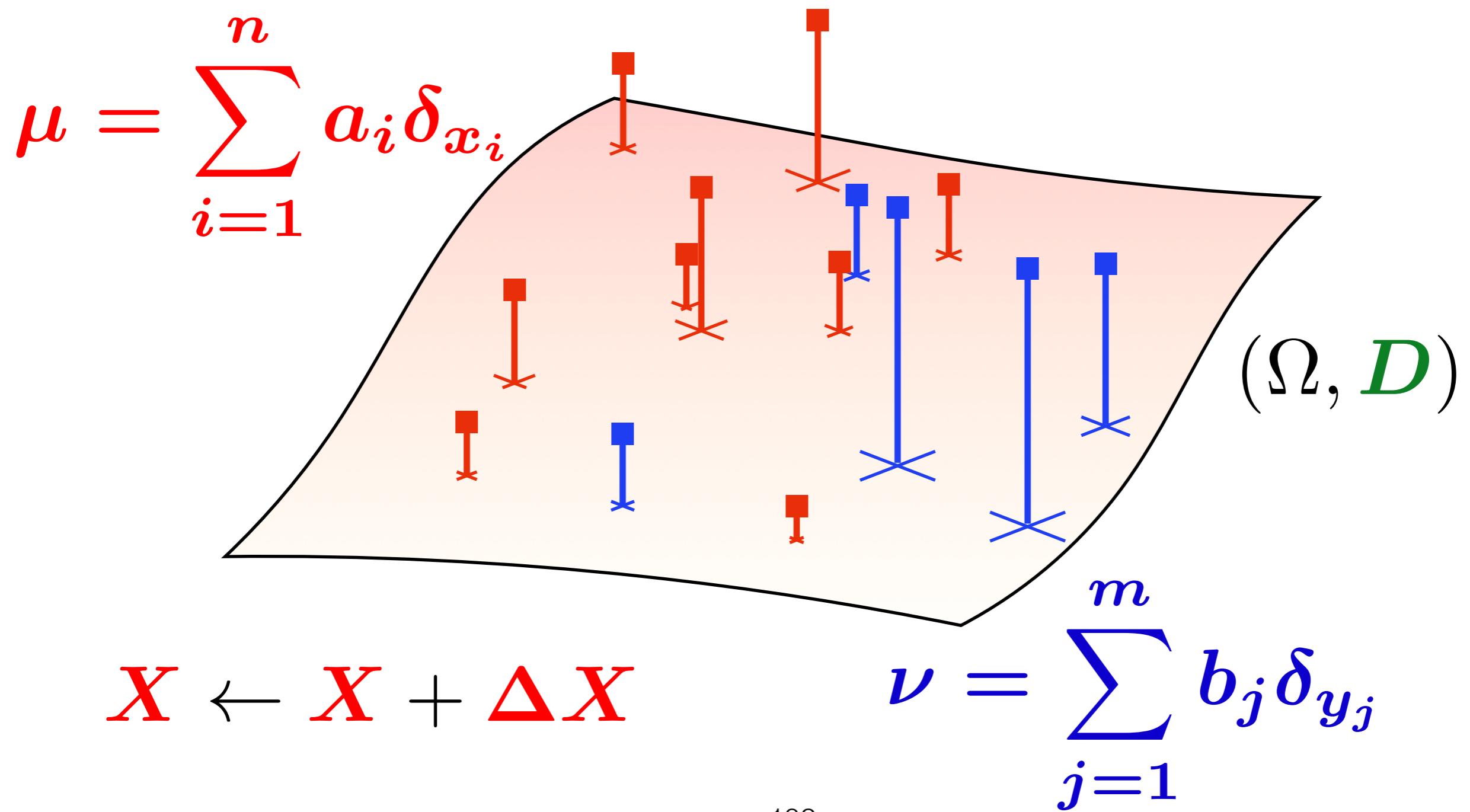
Sinkhorn \rightarrowtail Differentiability

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



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

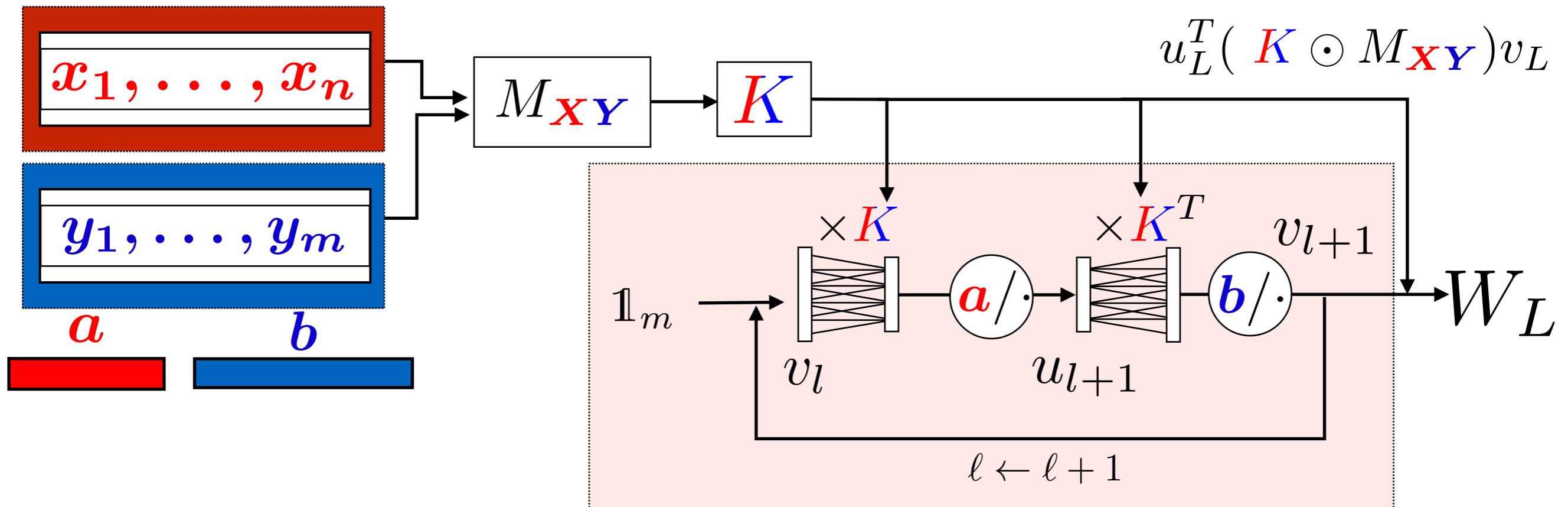
$\mathbf{v}_0 = \mathbf{1}_m$; $l \geq 0$, $\mathbf{u}_l \stackrel{\text{def}}{=} \mathbf{a}/\mathbf{K}\mathbf{v}_l$, $\mathbf{v}_{l+1} \stackrel{\text{def}}{=} \mathbf{b}/\mathbf{K}^T \mathbf{u}_l$.

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

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



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]

3. 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

Up to 2010: OT solvers $W_p(\mu, \nu) = ?$

Goal now: use OT as a **loss or fidelity term**

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

$\nabla_{\mu} W_p(\mu, \nu_1) = ?$

Wassersteinization

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

noun.

Introduction of optimal transport
into an optimization or learning
problem.

cf. least-squarification, L_1 ification, 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

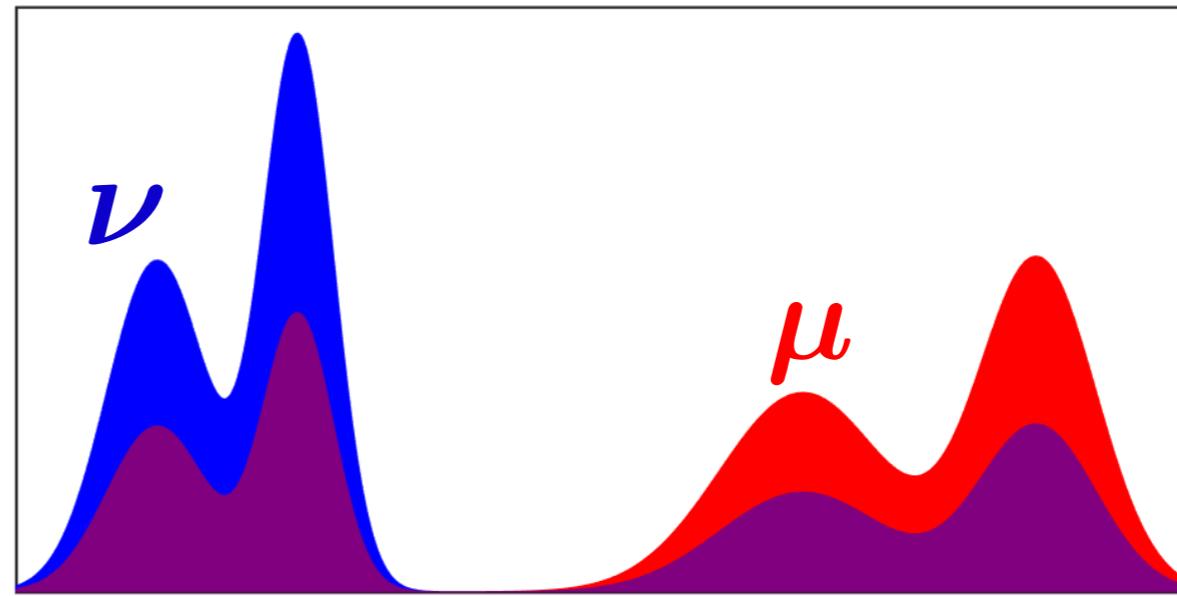
$$\min_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} (1-t)W_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}_1) + tW_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}_2)$$

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

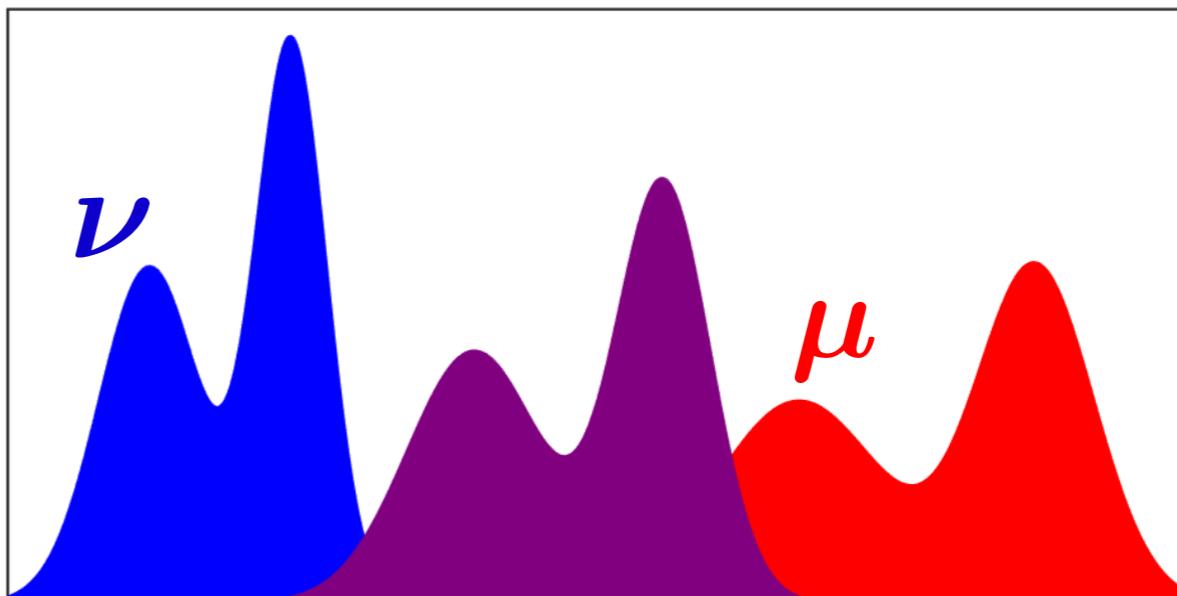
$$\mu_{t+1} = \operatorname*{argmin}_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} J(\boldsymbol{\mu}) + \lambda_t W_p^p(\boldsymbol{\mu}, \mu_t)$$

Averaging Measures

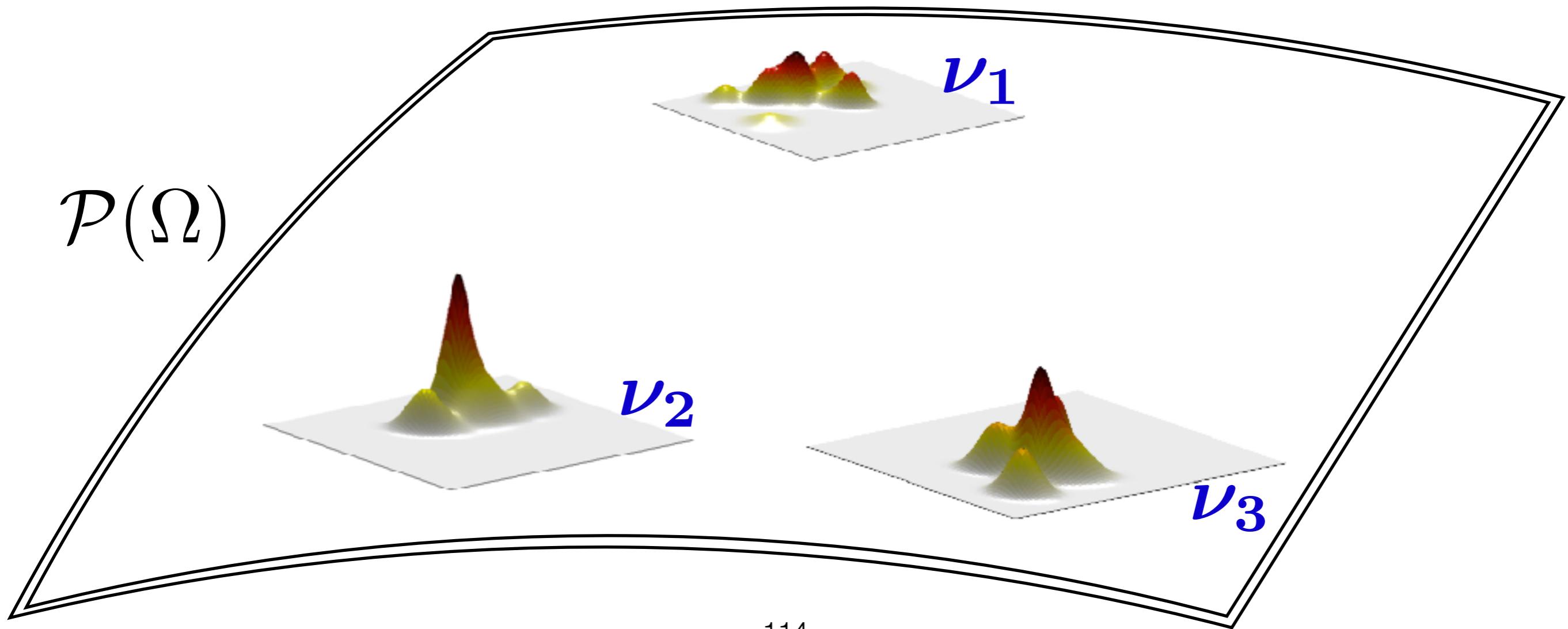
L_2 average



W average

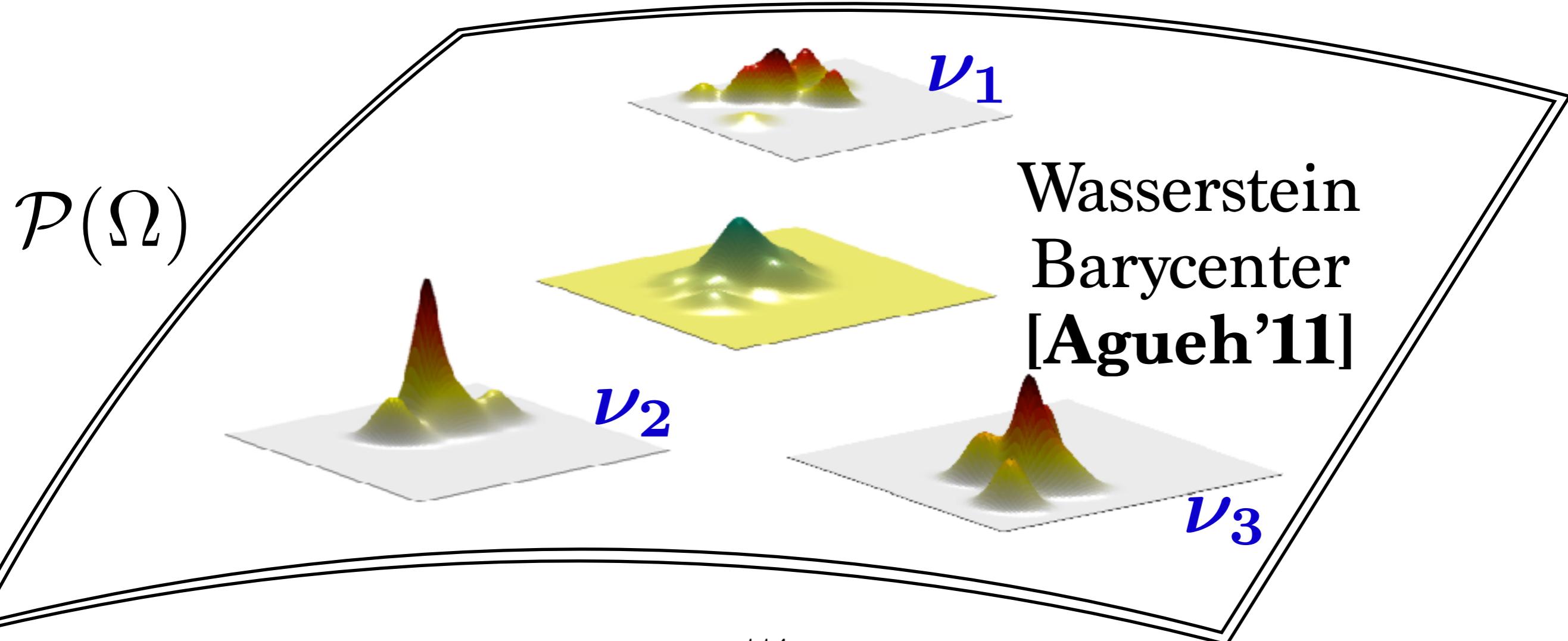


Barycenter for Measures?

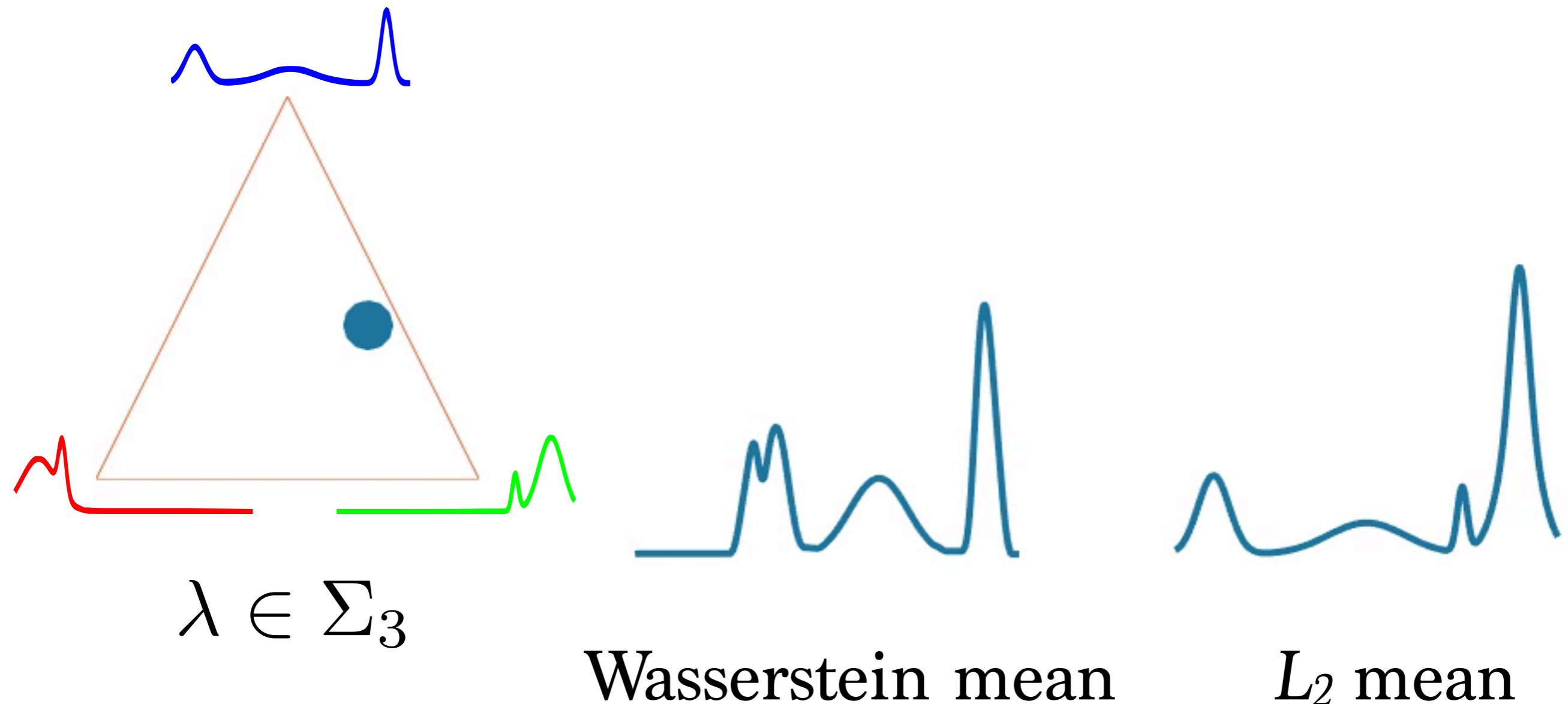


Barycenter for Measures?

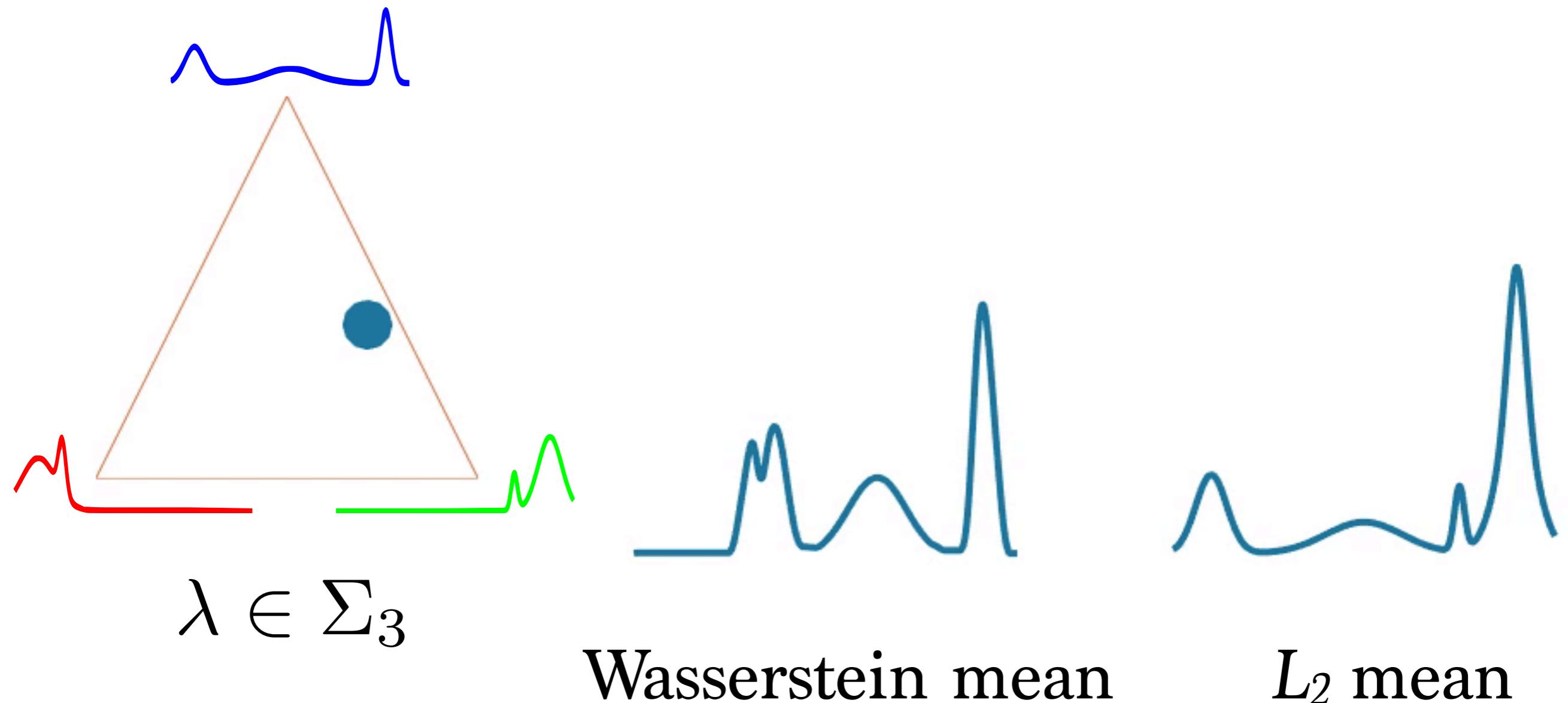
$$\min_{\mu \in \mathcal{P}(\Omega)} \sum_{i=1}^N \lambda_i W_p^p(\mu, \nu_i)$$



Barycenter for Measures?

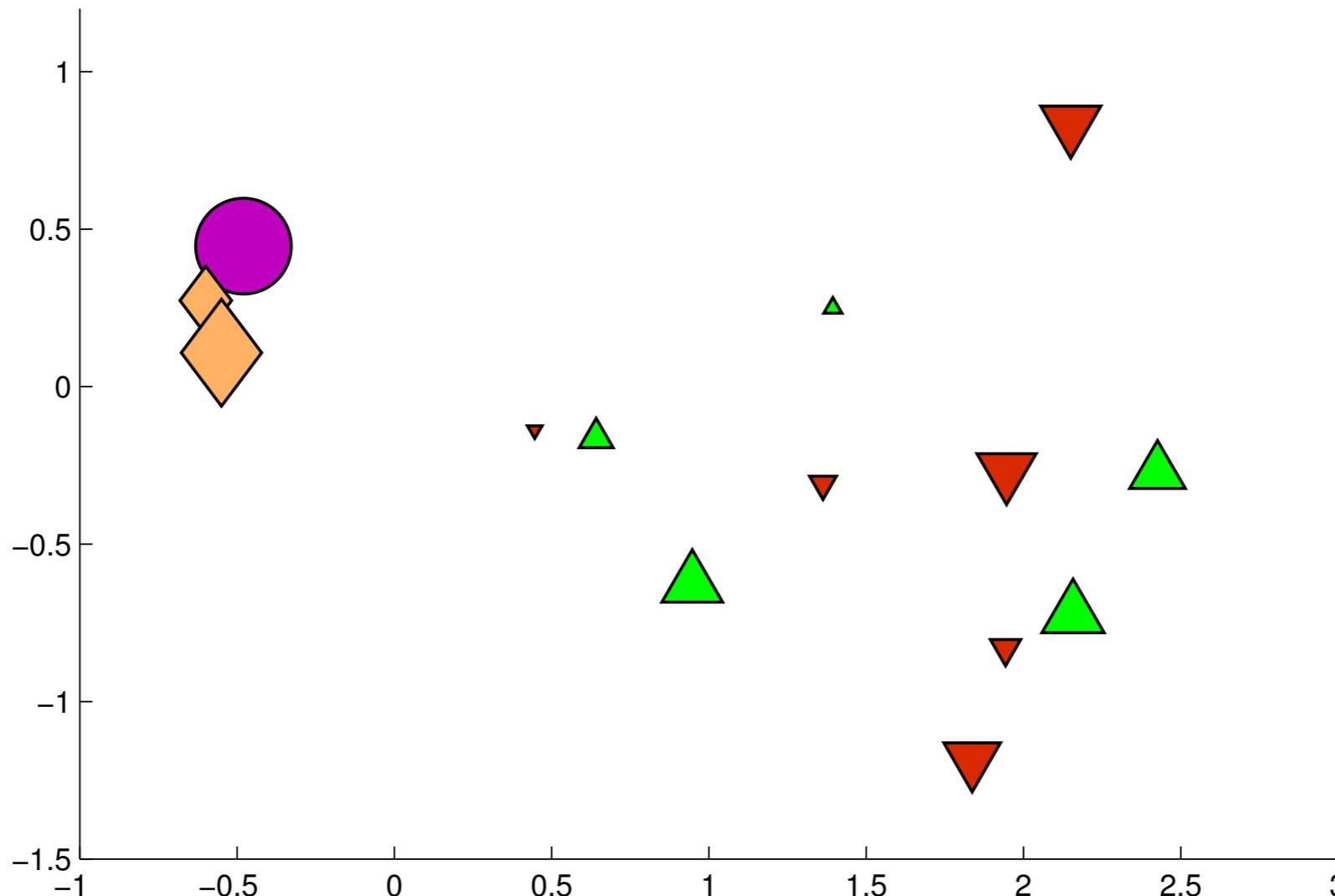


Barycenter for Measures?



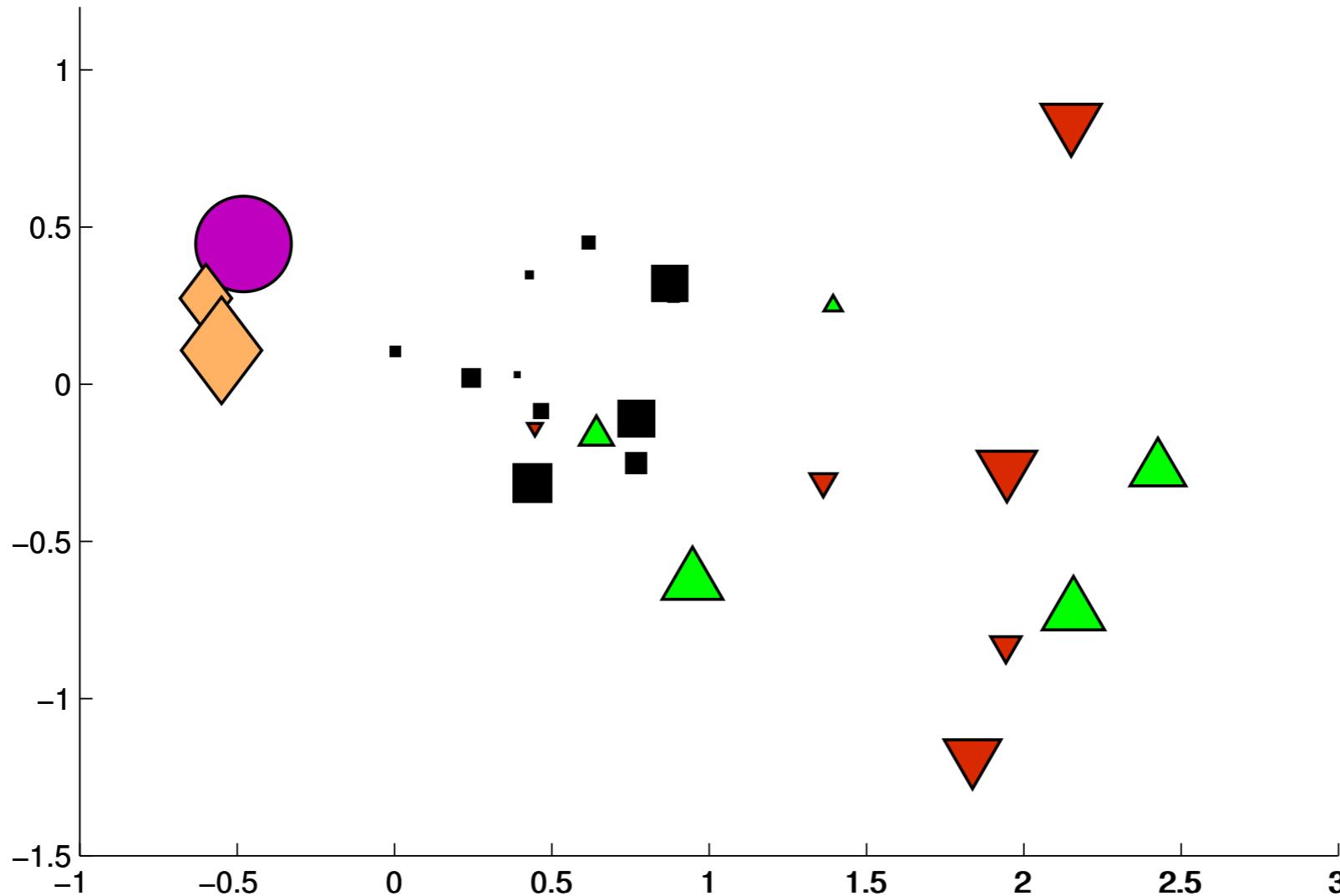
Multimarginal Formulation

- 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

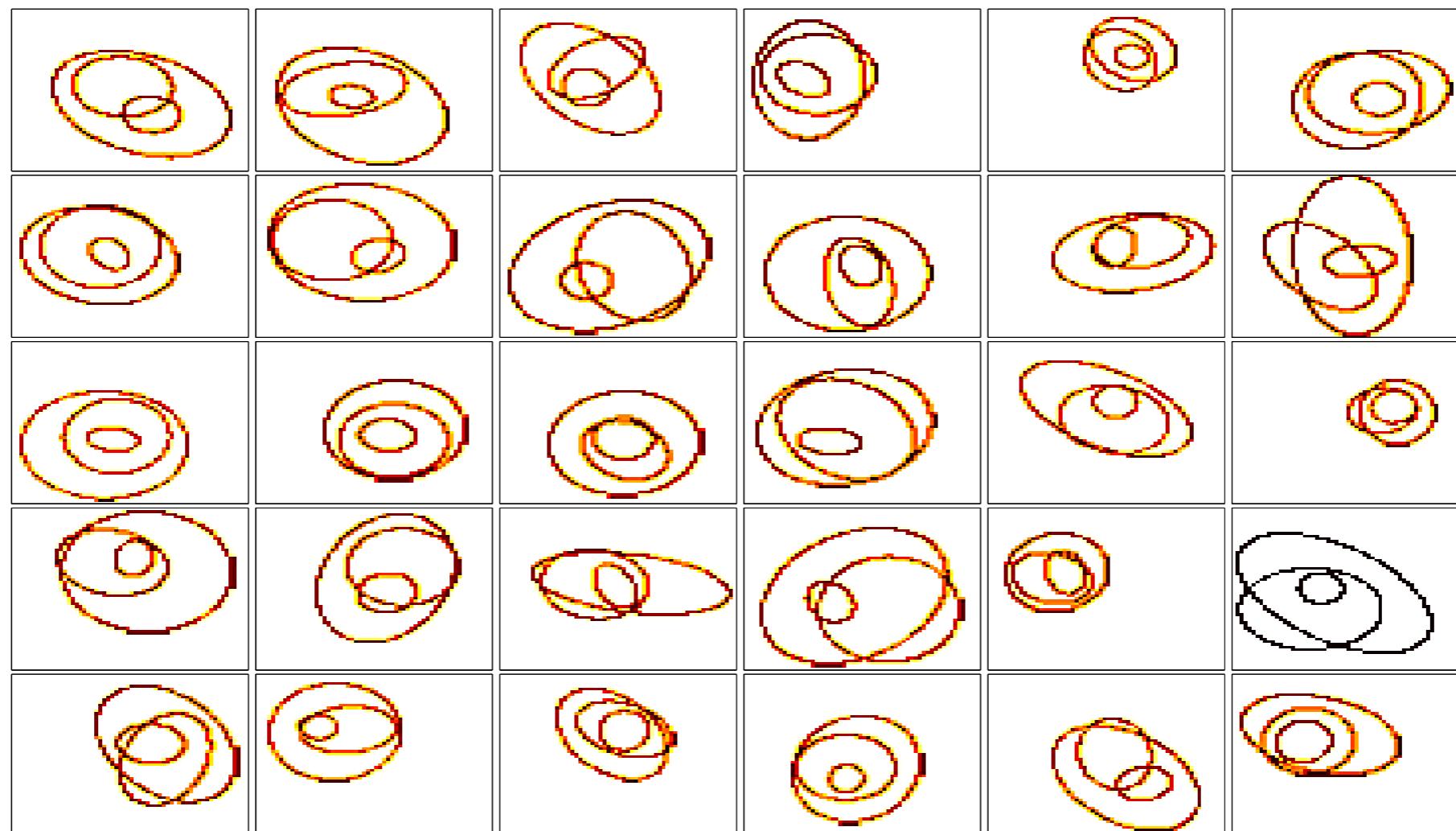
When Ω is a finite metric space defined by M .

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

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

Primal Descent on Regularized W

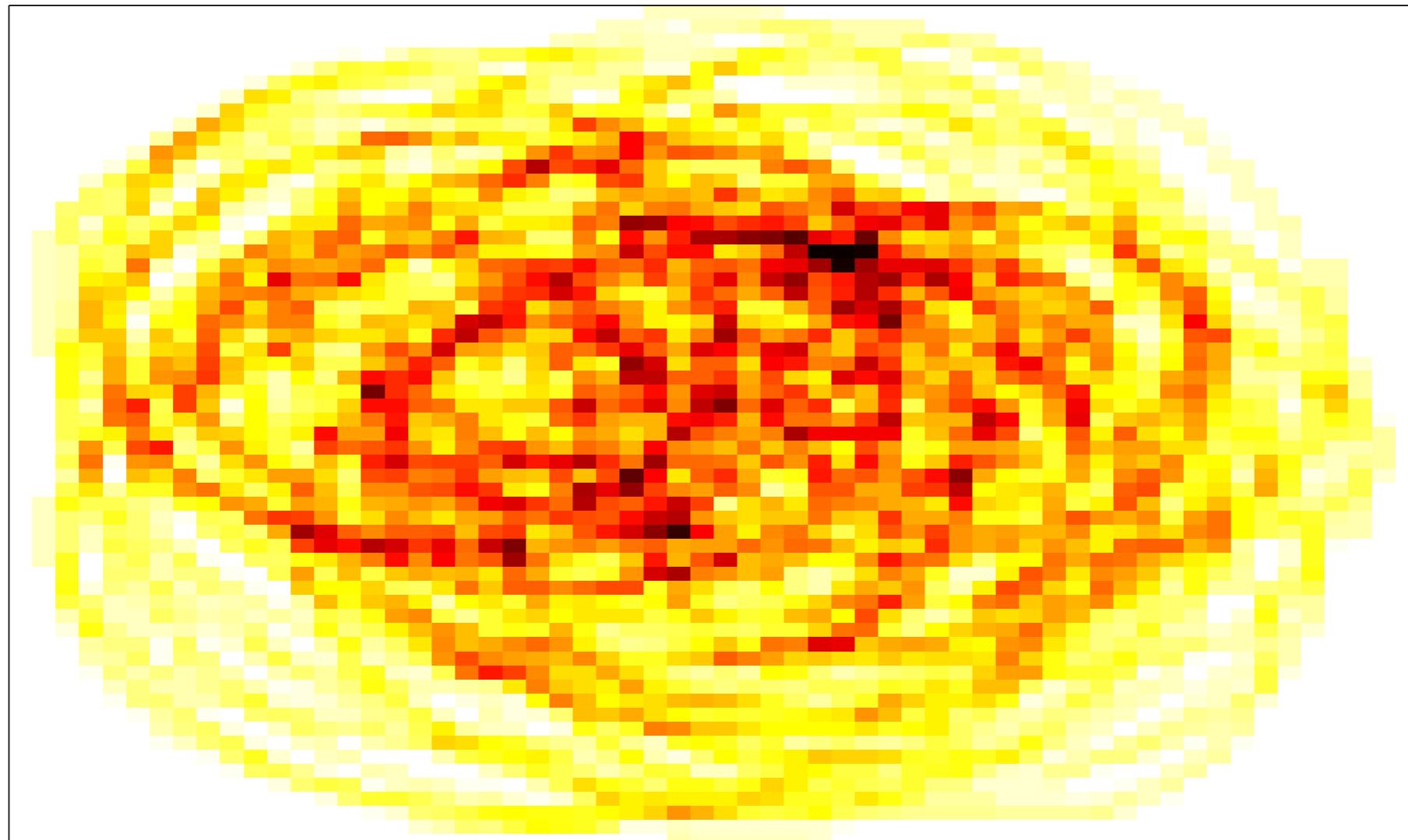
$$\min_{\mathbf{a} \in \Sigma_{h \times h}} \sum_{i=1}^N \lambda_i W_{\gamma}(\mathbf{a}, \mathbf{b}_i)$$



[Cuturi'14]

Primal Descent on Regularized W

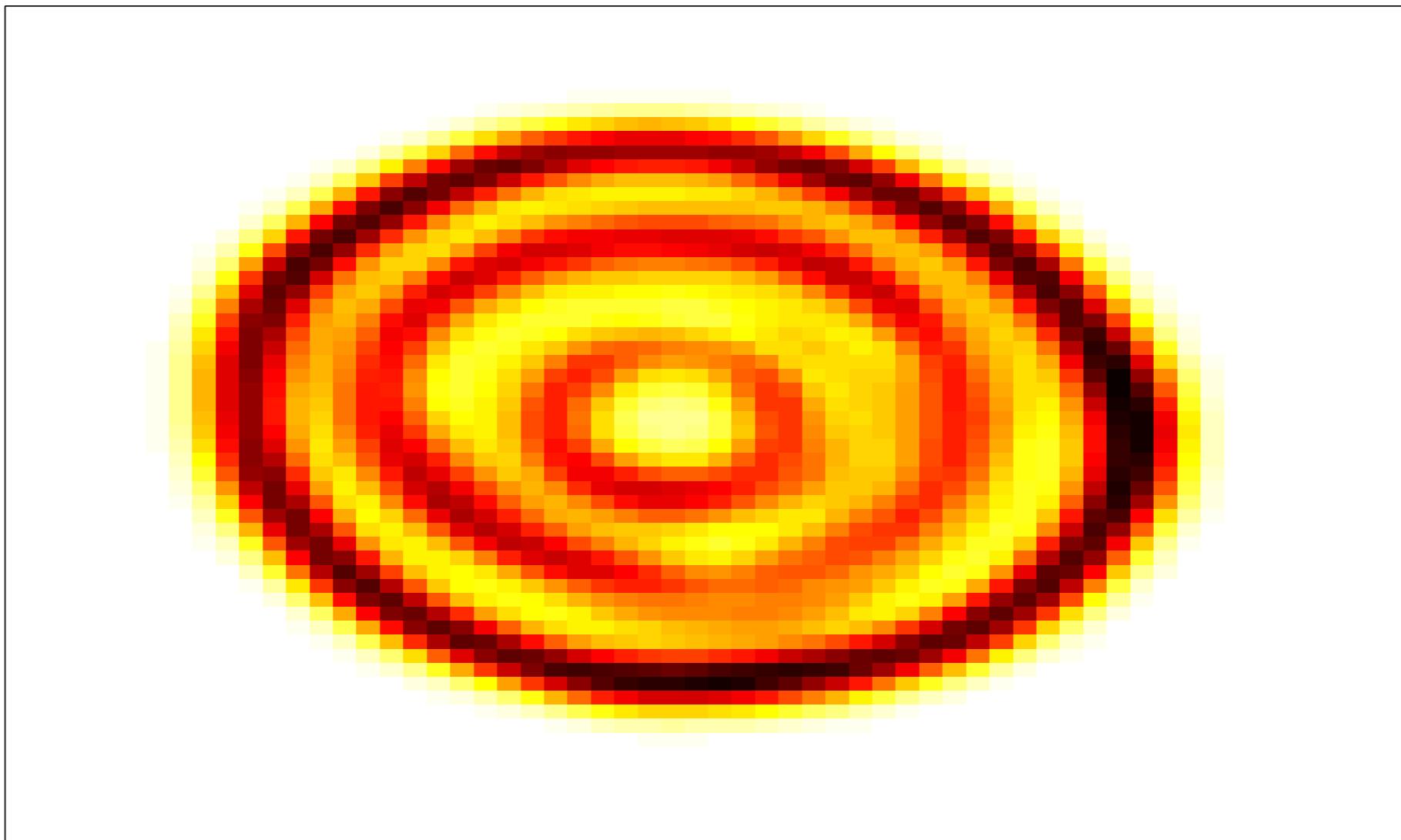
$$\min_{\mathbf{a} \in \Sigma_{h \times h}} \sum_{i=1}^N \lambda_i W_{\gamma}(\mathbf{a}, \mathbf{b}_i)$$



[Cuturi'14]

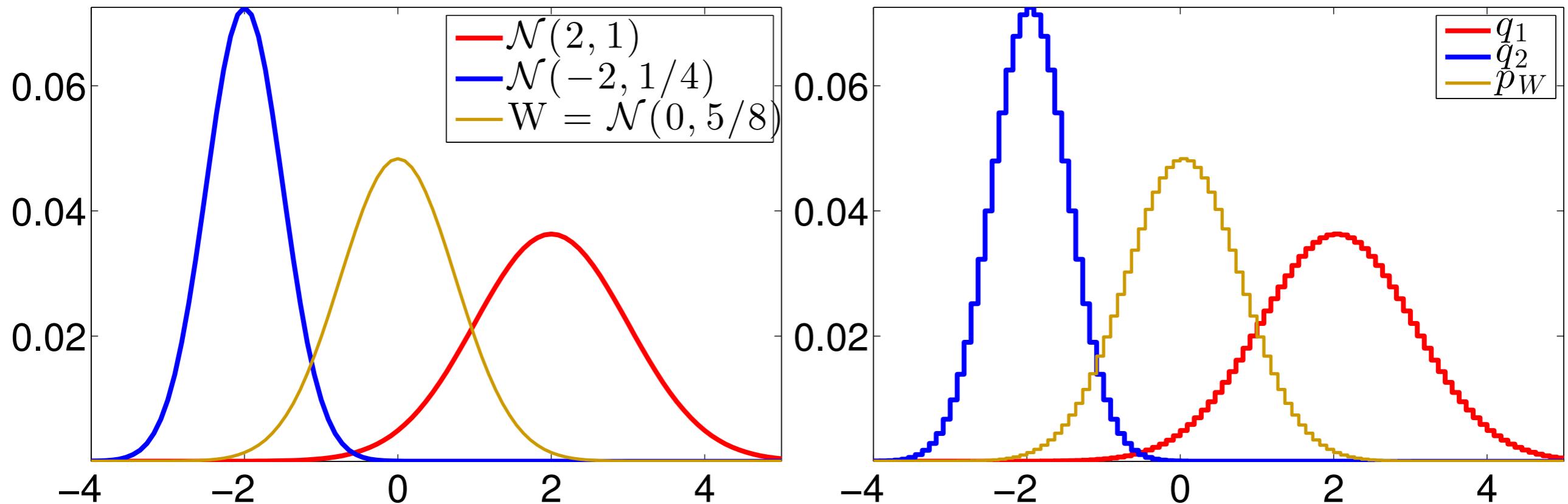
Primal Descent on Regularized W

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

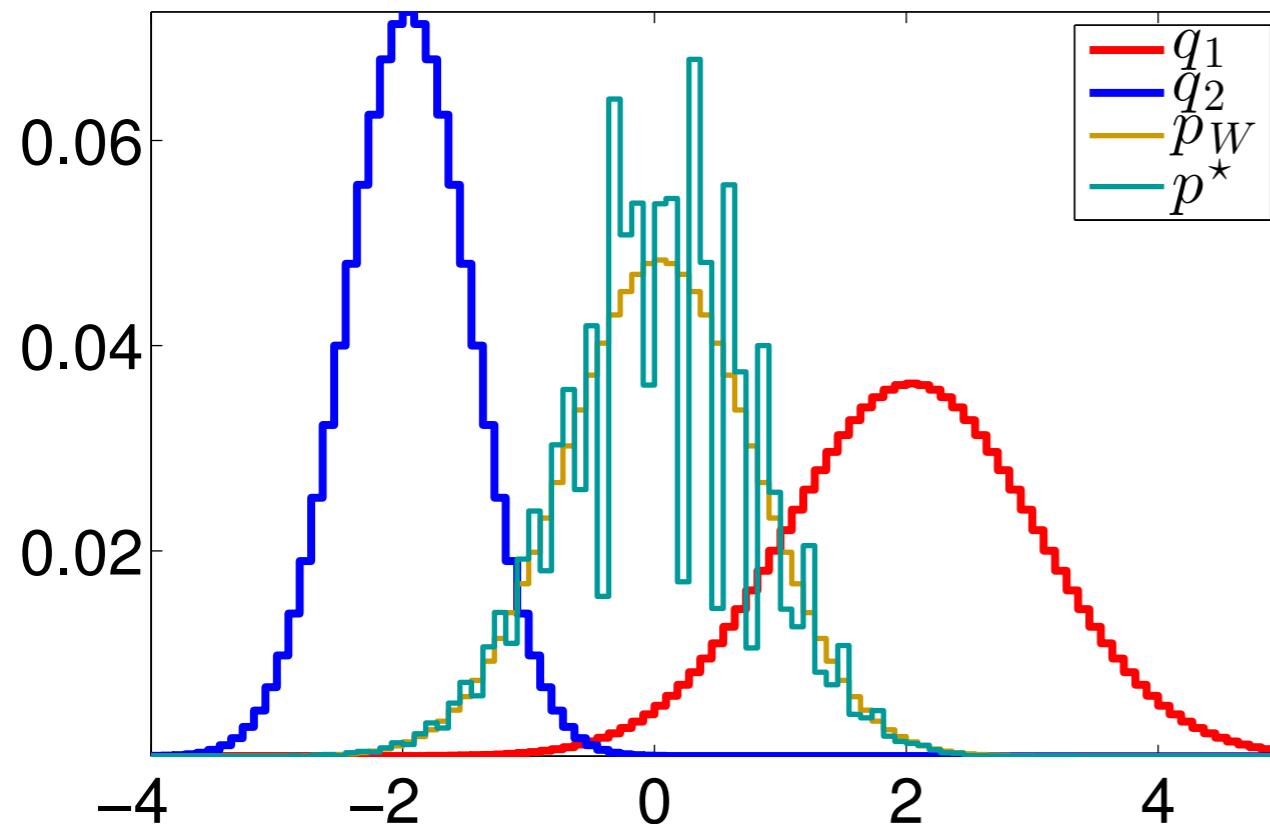


[Cuturi'14]

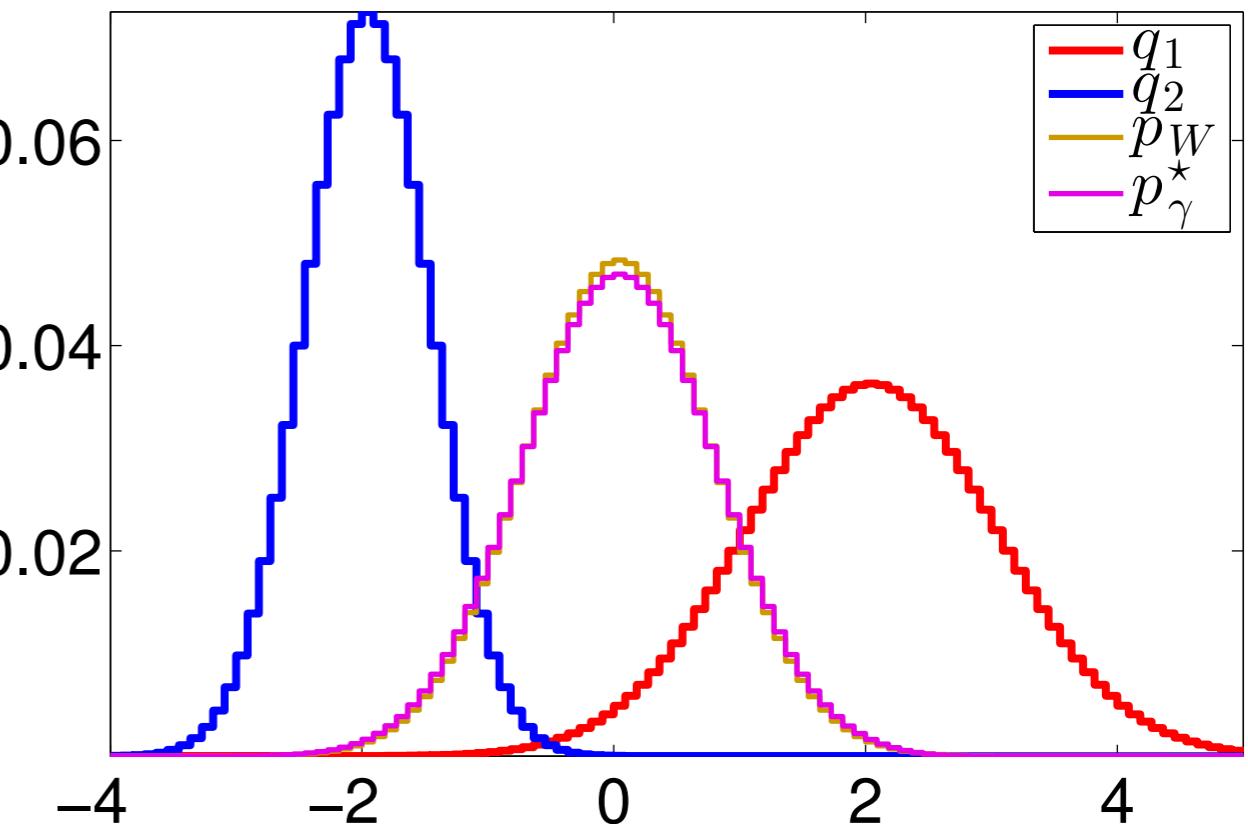
On Regularizing or Not



On Regularizing or Not

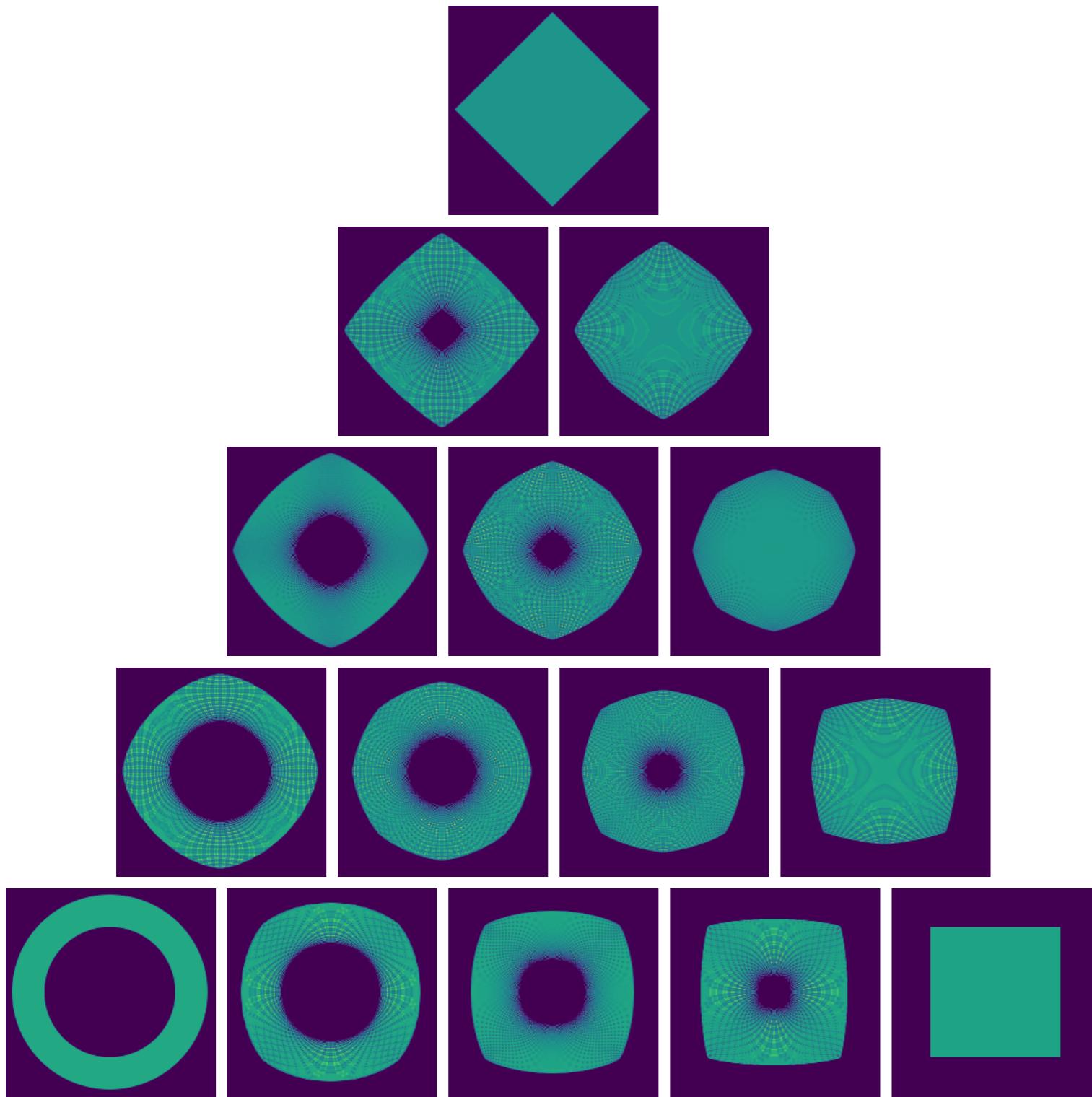


True barycenter



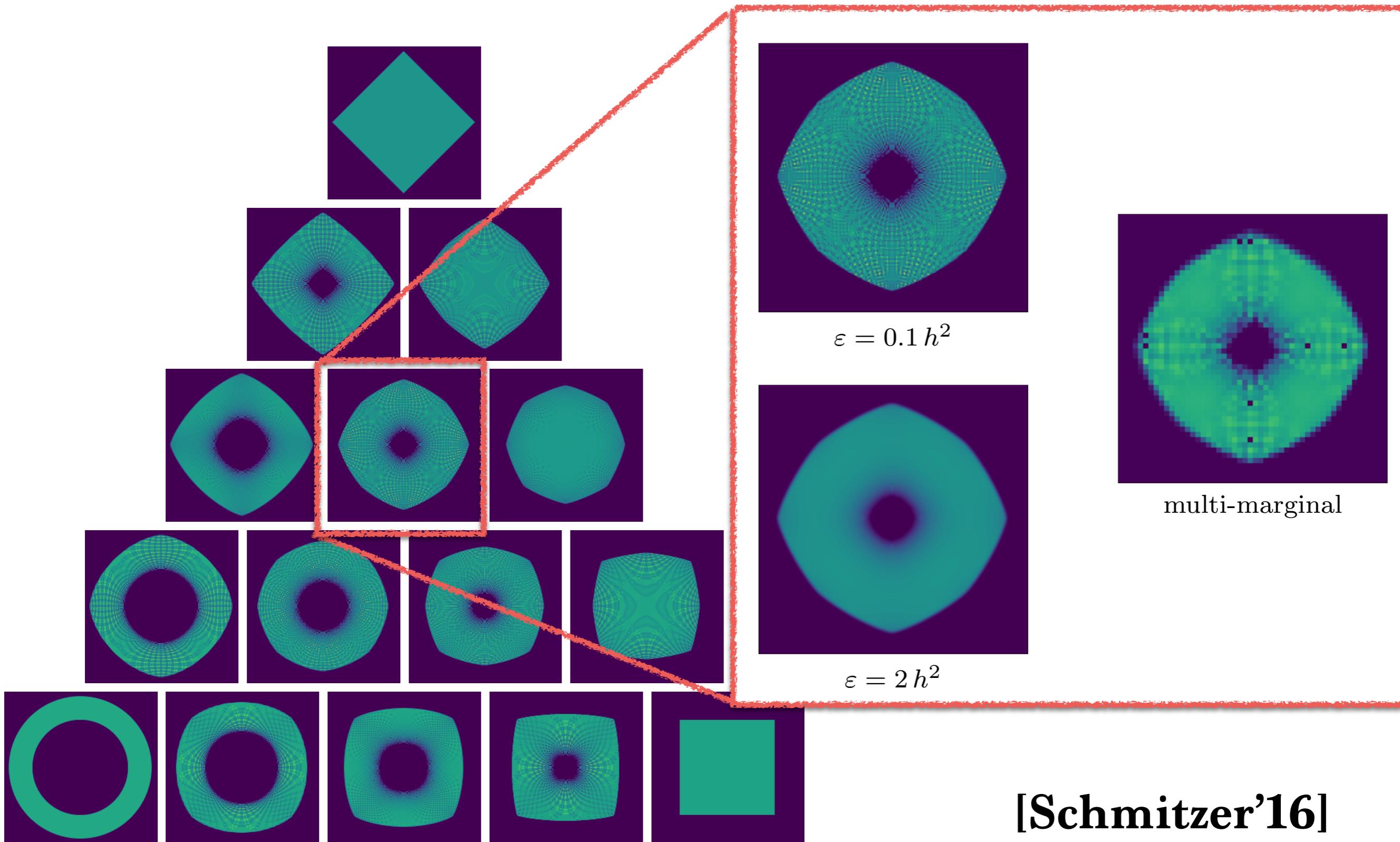
*Barycenter using
regularized OT*

On Regularizing or Not



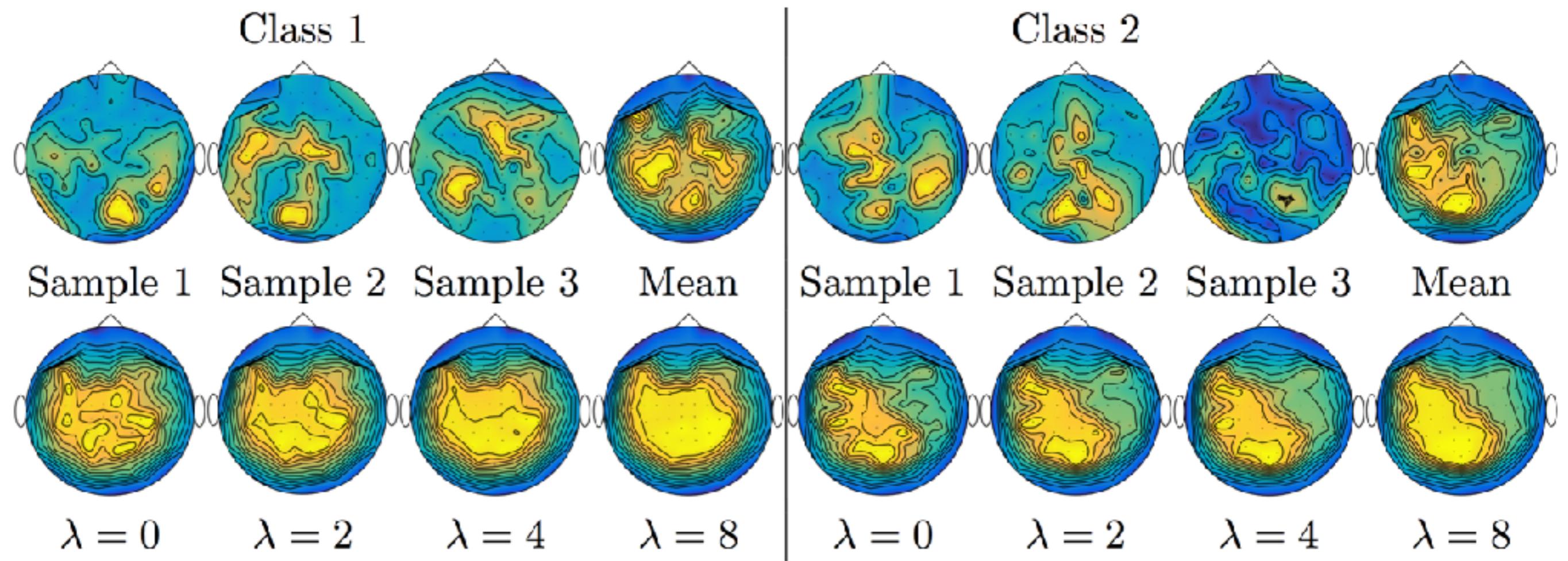
[Schmitzer'16]

On Regularizing or Not



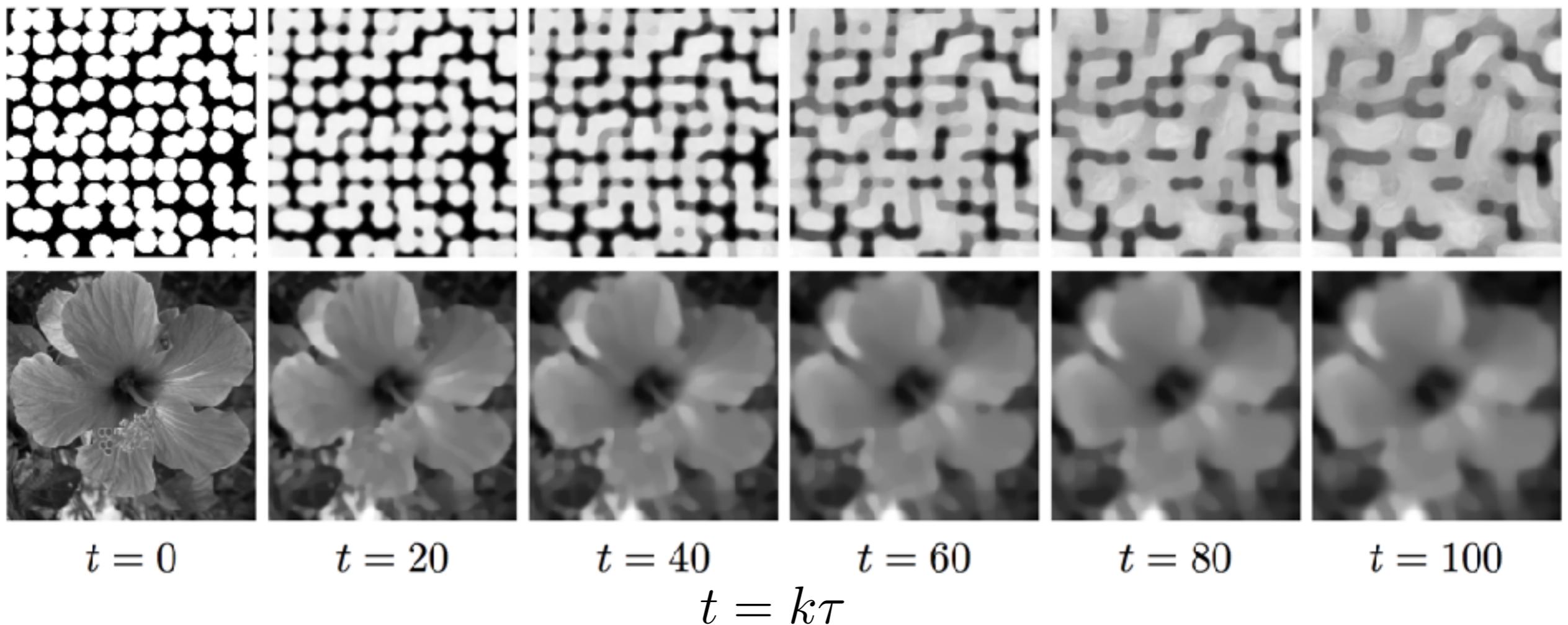
Duality: Regularized Barycenters

$$\min_{\boldsymbol{\mu} \in \mathcal{P}(\Omega)} \sum_{i=1}^N W_{\gamma}(\boldsymbol{\mu}, \boldsymbol{\nu}_i) + \lambda \text{TV}(\boldsymbol{\mu})$$



Duality: TV Gradient Flow

$$\mu_{k+1} = \operatorname{argmin}_{\mu \in \mathcal{P}(\Omega)} W_\gamma(\mu, \mu_k) + \tau \operatorname{TV}(\mu)$$



Regularized OT as KL Projection

$$\mathbf{KL}(P \mid \textcolor{red}{K}) = \sum_{ij} P_{ij} \log \left(P_{ij} / \textcolor{red}{K}_{ij} \right)$$
$$\langle P, M_{\textcolor{red}{X} \textcolor{blue}{Y}} \rangle - \gamma E(P) = \gamma \mathbf{KL}(P \mid \textcolor{red}{K})$$

Prop. $P_\gamma = \text{Proj}_{C_{\textcolor{red}{a}} \cap C'_{\textcolor{blue}{b}}}(\textcolor{red}{K})$

$$C_{\textcolor{red}{a}} = \{P \mid P \mathbf{1}_m = \textcolor{red}{a}\}, \quad C'_{\textcolor{blue}{b}} = \{P \mid P^T \mathbf{1}_n = \textcolor{blue}{b}\}$$

Regularized OT as KL Projection

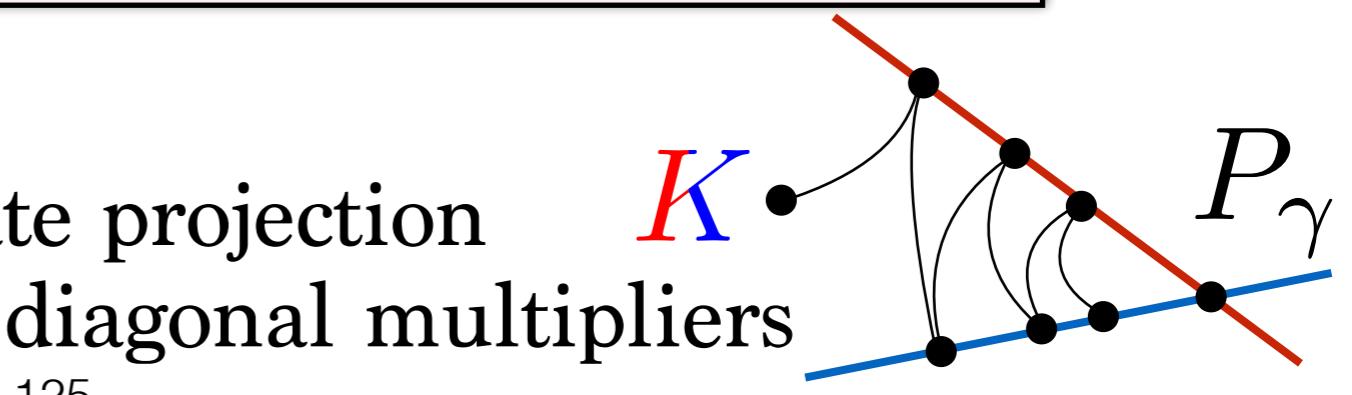
Prop. $P_\gamma = \text{Proj}_{C_{\mathbf{a}} \cap C'_{\mathbf{b}}}(\mathbf{K})$

$C_{\mathbf{a}} = \{P \mid P\mathbf{1}_m = \mathbf{a}\}, \quad C'_{\mathbf{b}} = \{P \mid P^T \mathbf{1}_n = \mathbf{b}\}$

$$\text{Proj}_{C_{\mathbf{a}}}(P) = \mathbf{D} \left(\frac{\mathbf{a}}{P\mathbf{1}_m} \right) P,$$

$$\text{Proj}_{C'_{\mathbf{b}}}(P) = P \mathbf{D} \left(\frac{\mathbf{b}}{P^T \mathbf{1}_n} \right).$$

1. Sinkhorn = Dykstra's alternate projection
2. Only need to store & update diagonal multipliers



Wasserstein Barycenter = KL Projections

$$\langle P, M_{\mathbf{X} \mathbf{Y}} \rangle - \gamma E(P) = \gamma \mathbf{KL}(P \mid \textcolor{red}{K})$$

$$\min_{\textcolor{red}{a}} \sum_{i=1}^N \lambda_i W_\gamma(\textcolor{red}{a}, \textcolor{blue}{b}_i) = \min_{\substack{\mathbf{P} = [P_1, \dots, P_N] \\ \mathbf{P} \in \textcolor{red}{C}_1 \cap \textcolor{blue}{C}_2}} \sum_{i=1}^N \lambda_i \mathbf{KL}(\textcolor{red}{P}_i \mid \textcolor{red}{K})$$

$$\textcolor{red}{C}_1 = \{\mathbf{P} \mid \exists \textcolor{red}{a}, \forall i, P_i \mathbf{1}_m = \textcolor{red}{a}\}$$

$$\textcolor{blue}{C}_2 = \{\mathbf{P} \mid \forall i, P_i^T \mathbf{1}_n = \textcolor{blue}{b}_i\}$$

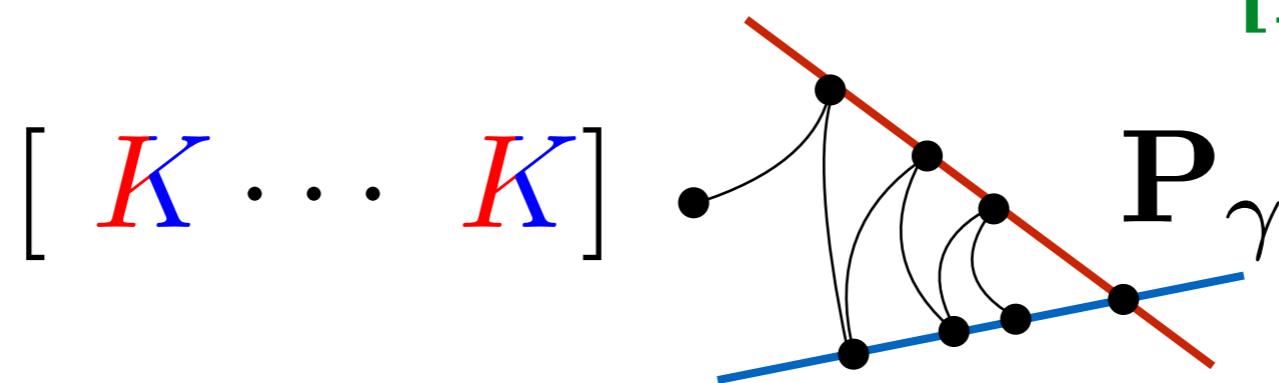
Wasserstein Barycenter = KL Projections

$$\min_{\mathbf{a}} \sum_{i=1}^N \lambda_i W_\gamma(\mathbf{a}, \mathbf{b}_i) = \min_{\substack{\mathbf{P} = [\mathbf{P}_1, \dots, \mathbf{P}_N] \\ \mathbf{P} \in \mathbf{C}_1 \cap \mathbf{C}_2}} \sum_{i=1}^N \lambda_i \text{KL}(\mathbf{P}_i | \mathbf{K})$$

$$\mathbf{C}_1 = \{\mathbf{P} \mid \exists \mathbf{a}, \forall i, \mathbf{P}_i \mathbf{1}_m = \mathbf{a}\}$$

$$\mathbf{C}_2 = \{\mathbf{P} \mid \forall i, \mathbf{P}_i^T \mathbf{1}_n = \mathbf{b}_i\}$$

[BCCNP'15]



Wasserstein Barycenter = KL Projections

$$\min_{\mathbf{a}} \sum_{i=1}^N \lambda_i W_\gamma(\mathbf{a}, \mathbf{b}_i) = \min_{\substack{\mathbf{P} = [\mathbf{P}_1, \dots, \mathbf{P}_N] \\ \mathbf{P} \in \mathbf{C}_1 \cap \mathbf{C}_2}} \sum_{i=1}^N \lambda_i \text{KL}(\mathbf{P}_i | \mathbf{K})$$

$$\mathbf{C}_1 = \{\mathbf{P} \mid \exists \mathbf{a}, \forall i, \mathbf{P}_i \mathbf{1}_m = \mathbf{a}\}$$

$$\mathbf{C}_2 = \{\mathbf{P} \mid \forall i, \mathbf{P}_i^T \mathbf{1}_n = \mathbf{b}_i\}$$

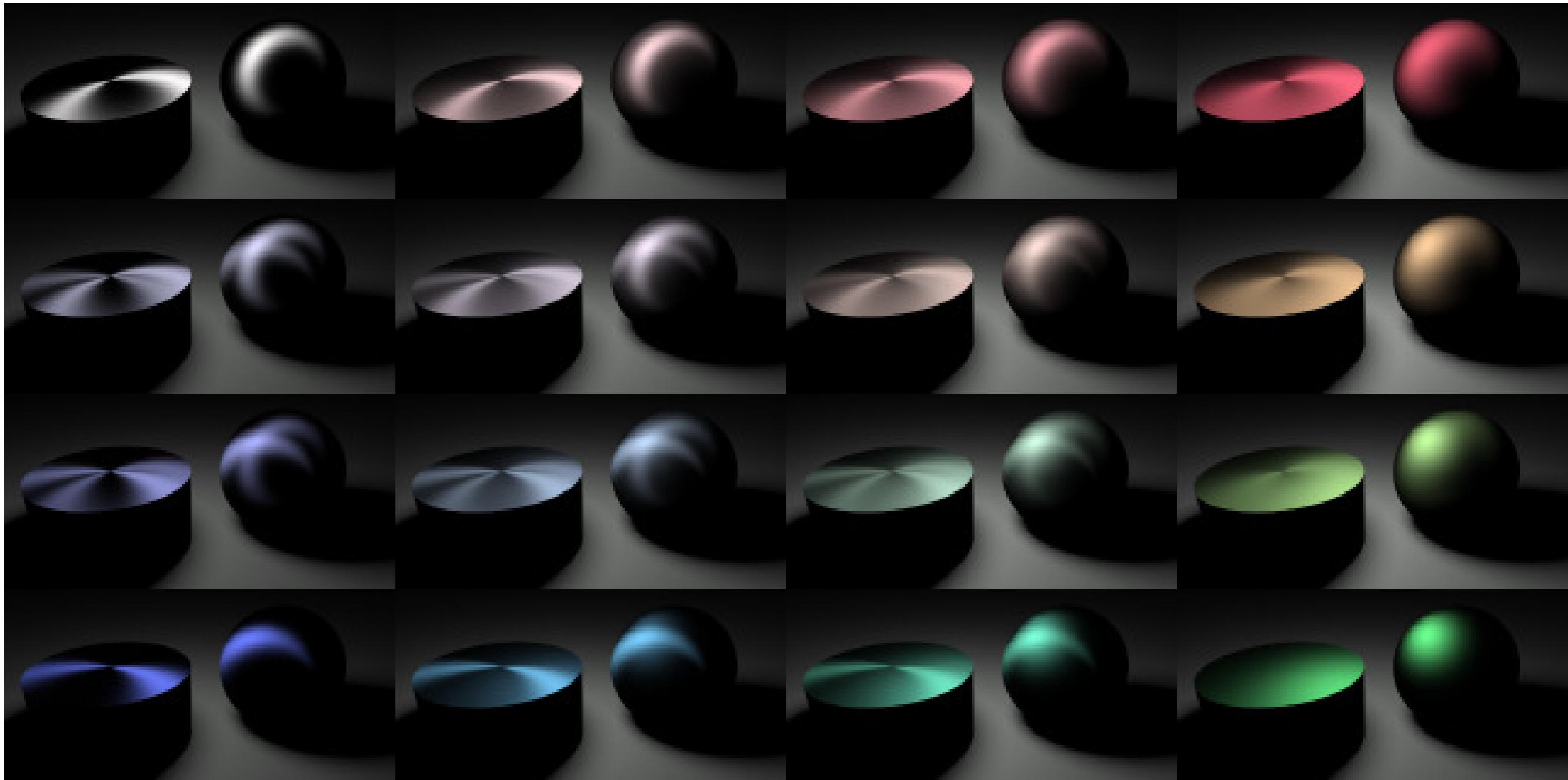
```
u=ones(size(B)); % d x N matrix
while not converged
```

```
    v=u.*(K'*(B./(K*u))); % 2(Nd^2) cost
    u=bsxfun(@times,u,exp(log(v)*weights))./v;
end
a=mean(v,2);
```

[BCCNP'15]

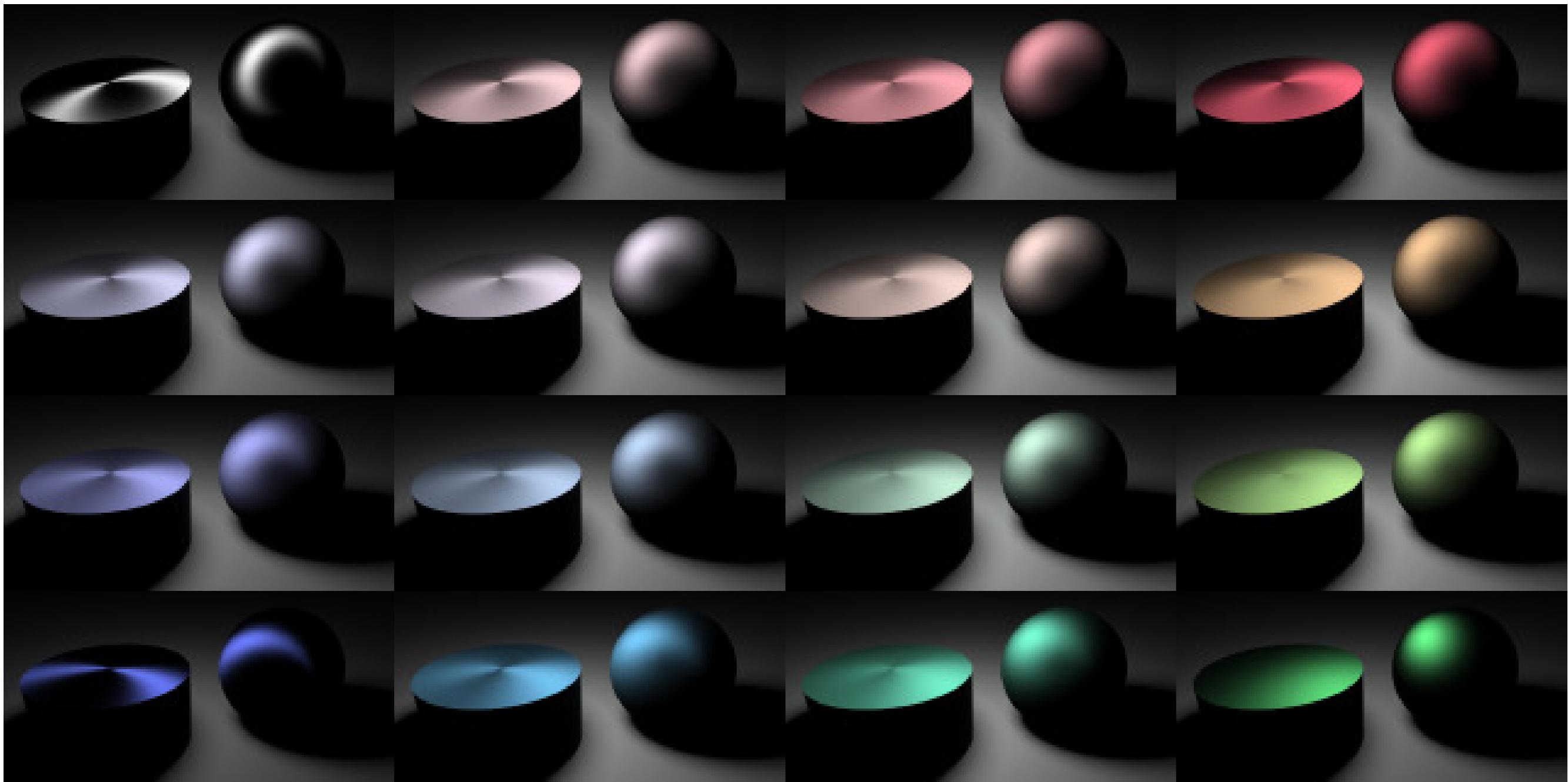
Iterative Bregman Projections for
Regularized Transportation Problems
SIAM J. on Sci. Comp. 2015

Applications in Imaging



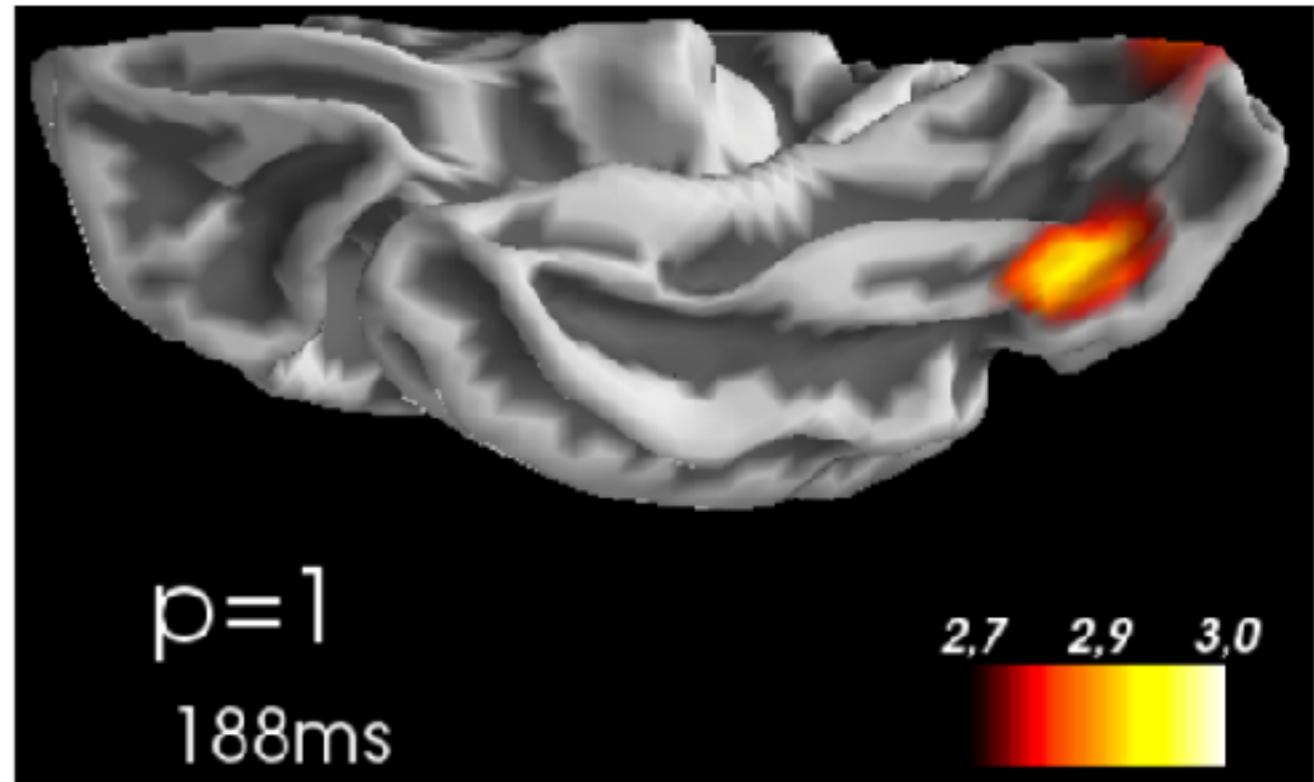
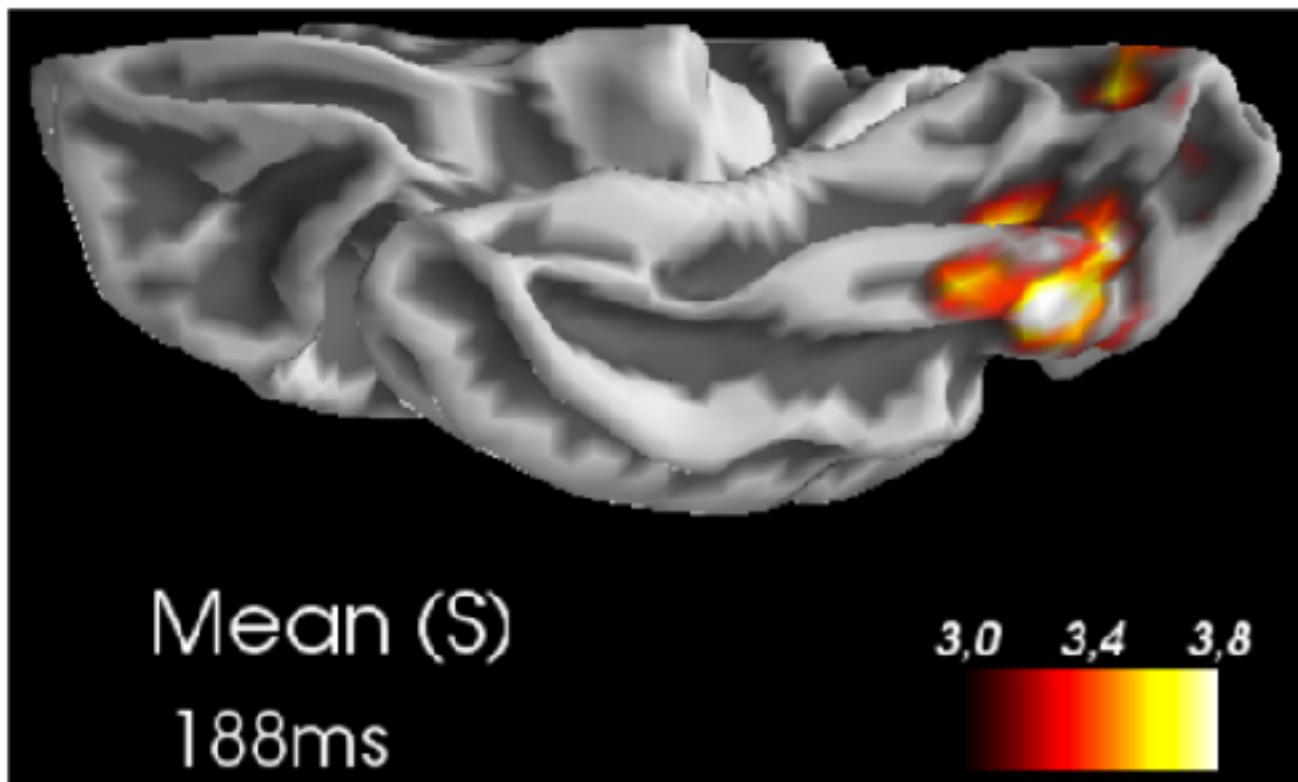
[Solomon'15]

Applications in Imaging



[Solomon'15]

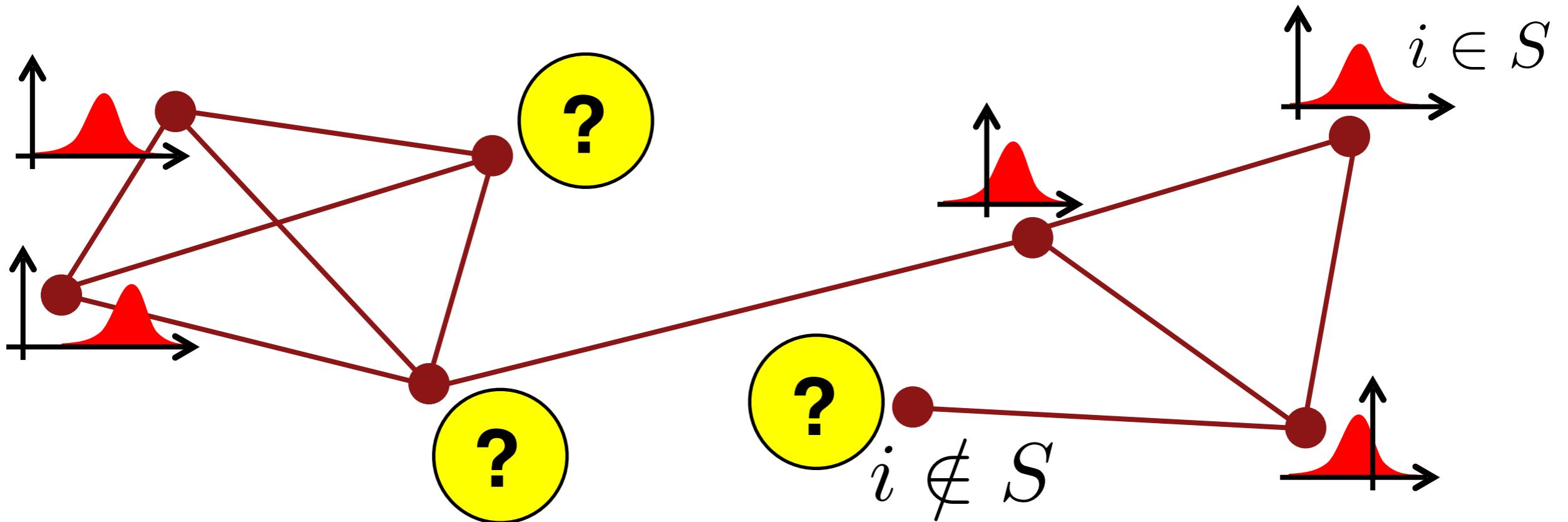
Applications: Brain Imaging



Extension to **non-normalized** data!
Applied to MEG and fMRI.

[Gramfort'16]

Wasserstein Propagation



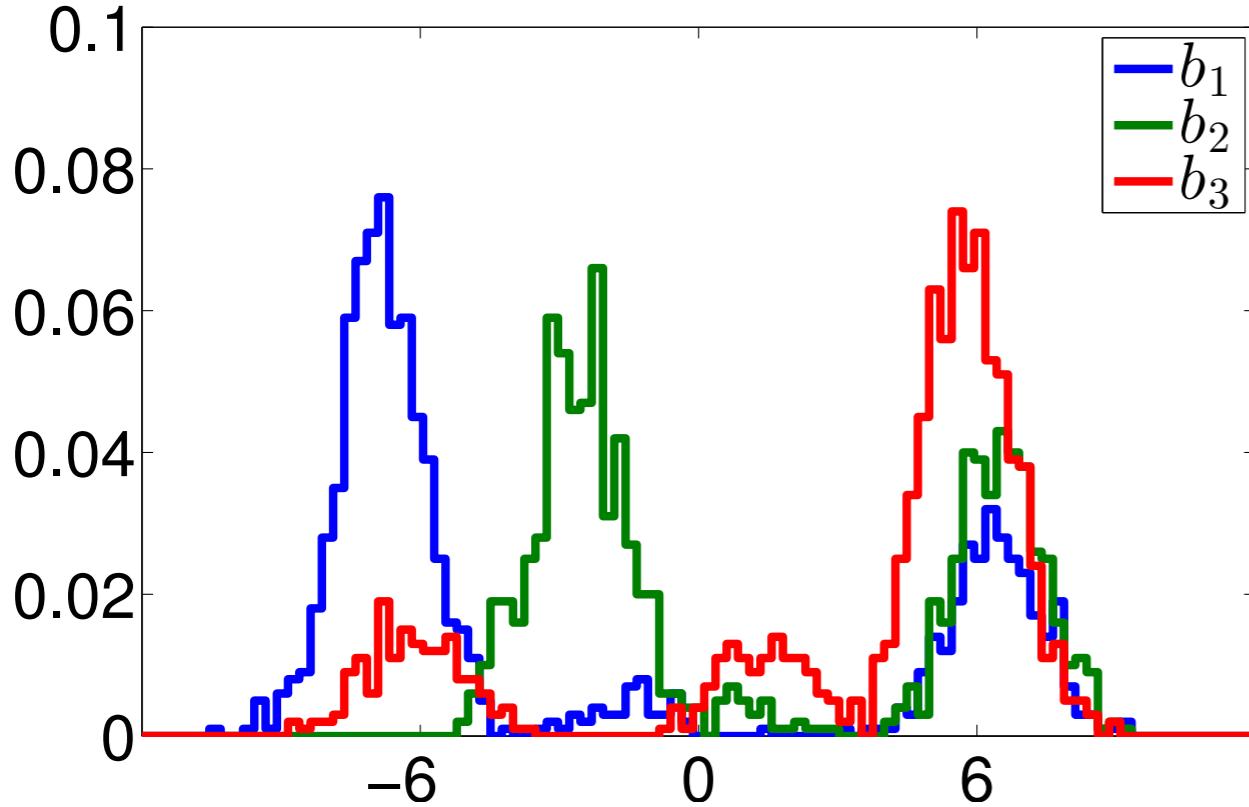
$$\min_{\substack{\mu_i \in \mathcal{P}(\Omega) \\ \mu_i \text{ fixed for } i \in S}} \sum_{(e_1, e_2) \in E} W_2^2(\mu_{e_1}, \mu_{e_2})$$

[Solomon'14]

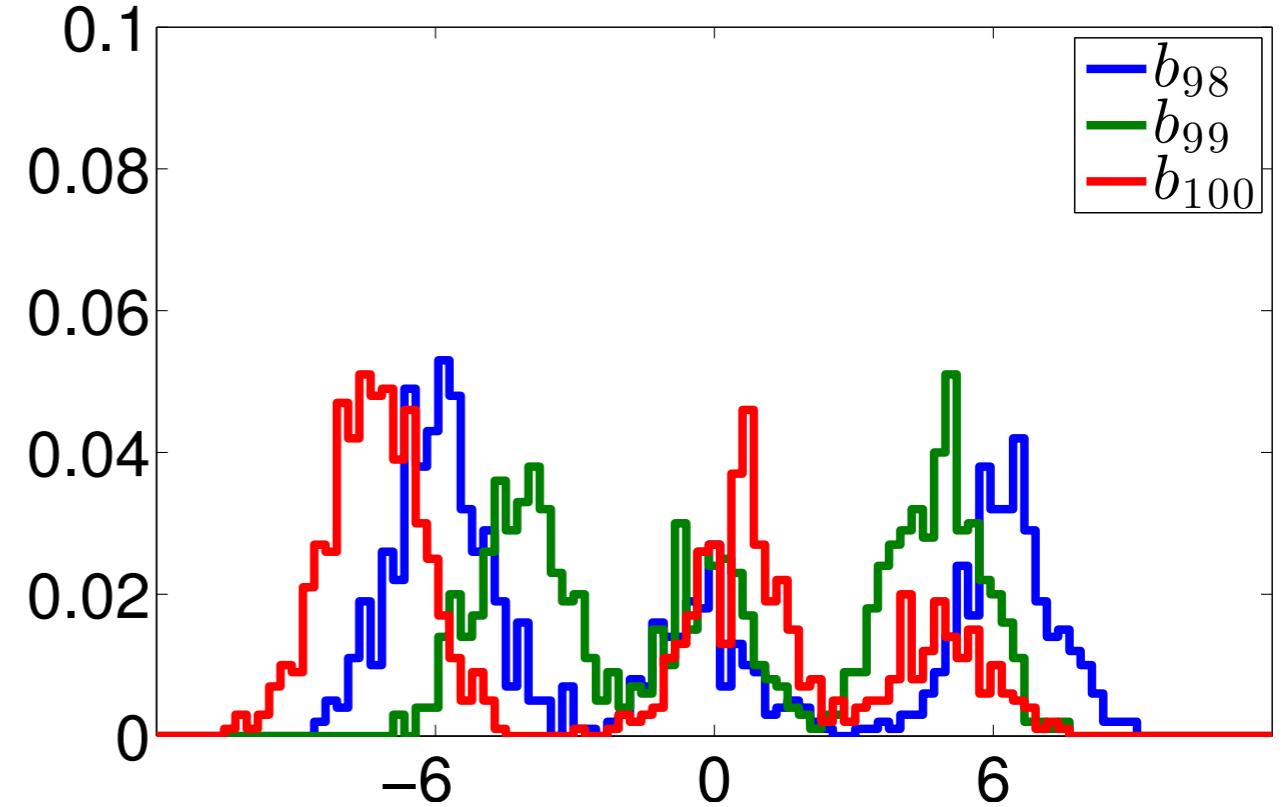
Dictionary Learning

$$\min_{\mathbf{A} \in (\Sigma_n)^K, \mathbf{\Lambda} \in (\Sigma_K)^N} \sum_{i=1}^N W \left(\mathbf{b}_i, \sum_{k=1}^K \mathbf{\Lambda}_k^i \mathbf{a}_k \right)$$

Data samples



Data samples

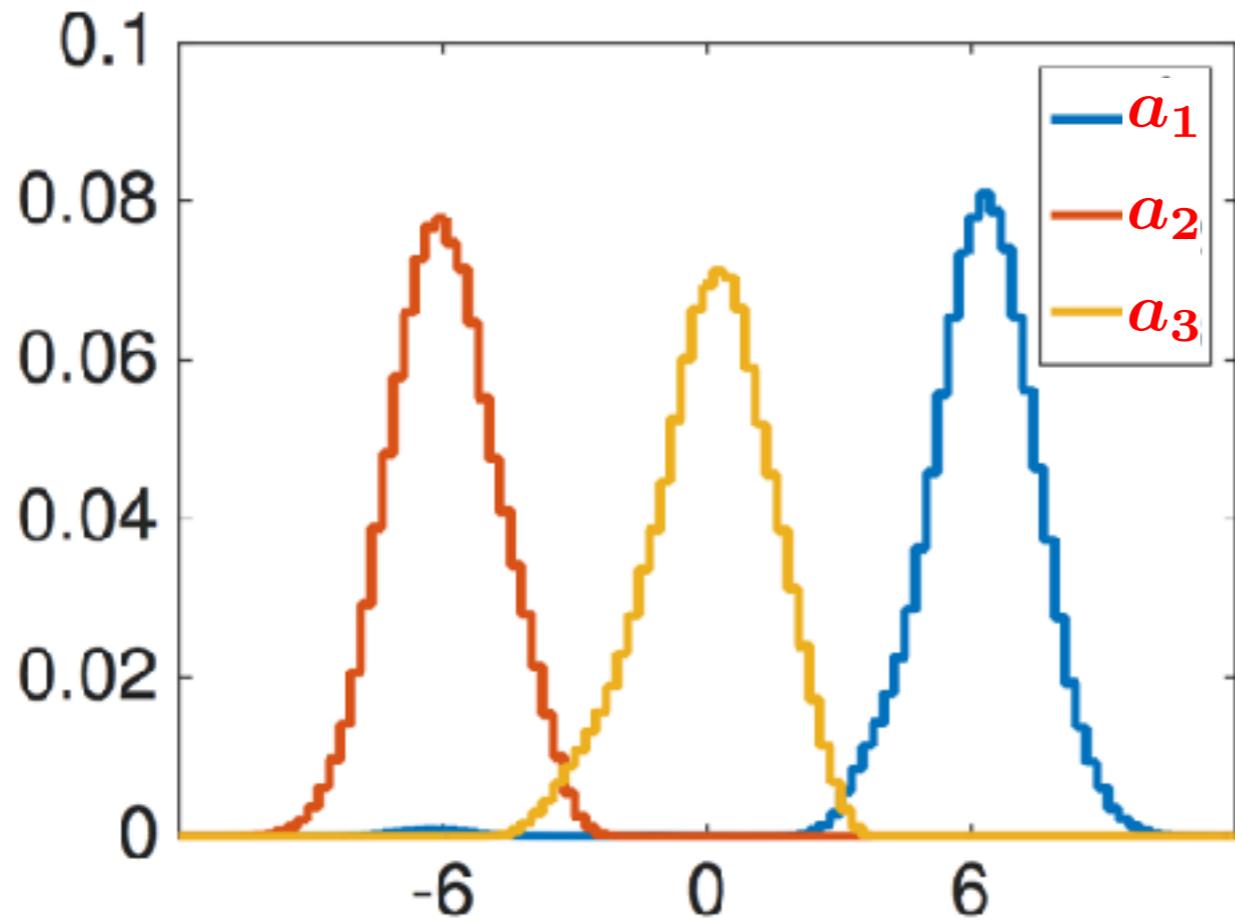


[Sandler'11] [Zen'14] [Rolet'16]

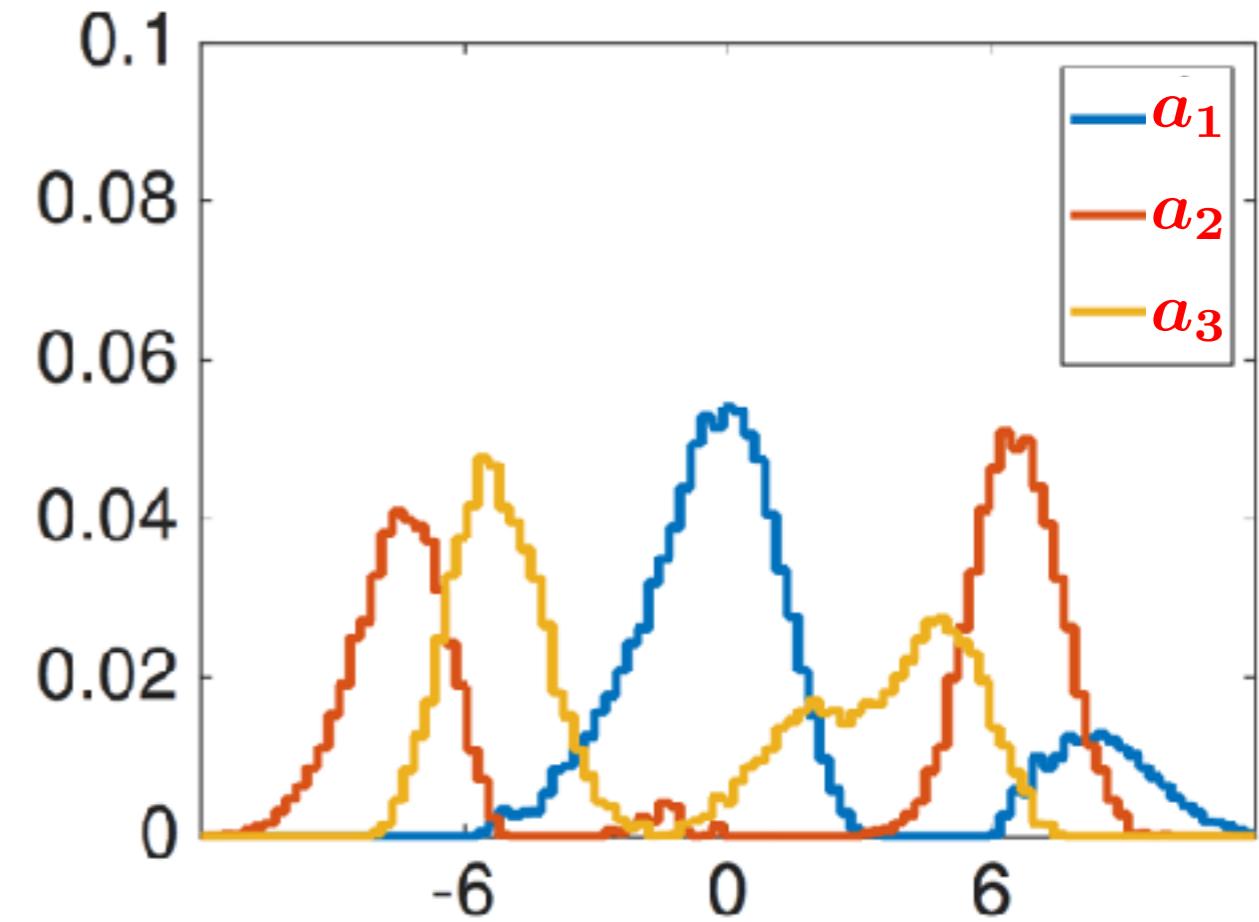
Dictionary Learning

$$\min_{\mathbf{A} \in (\Sigma_n)^K, \mathbf{\Lambda} \in (\Sigma_K)^N} \sum_{i=1}^N W \left(\mathbf{b}_i, \sum_{k=1}^K \mathbf{\Lambda}_k^i \mathbf{a}_k \right)$$

Wasserstein NMF



KL NMF



[Sandler'11] [Zen'14] [Rolet'16]

OT Dictionary Learning

- [Hoffman'98] proposed to learn dictionaries (topics) for text, seen as histograms-of-words.

$$\Omega = \{\text{words}\}, \quad |\Omega| \approx 13,000$$

- Vector embeddings for words [Mikolov'13] [Pennington'14] defines geometry:
$$\textcolor{violet}{D}(\text{public}, \text{car}) = \|x_{\text{public}} - x_{\text{car}}\|^2$$
- Data: 7,034 Reuters, 737 BBC sports news articles

Topic Models

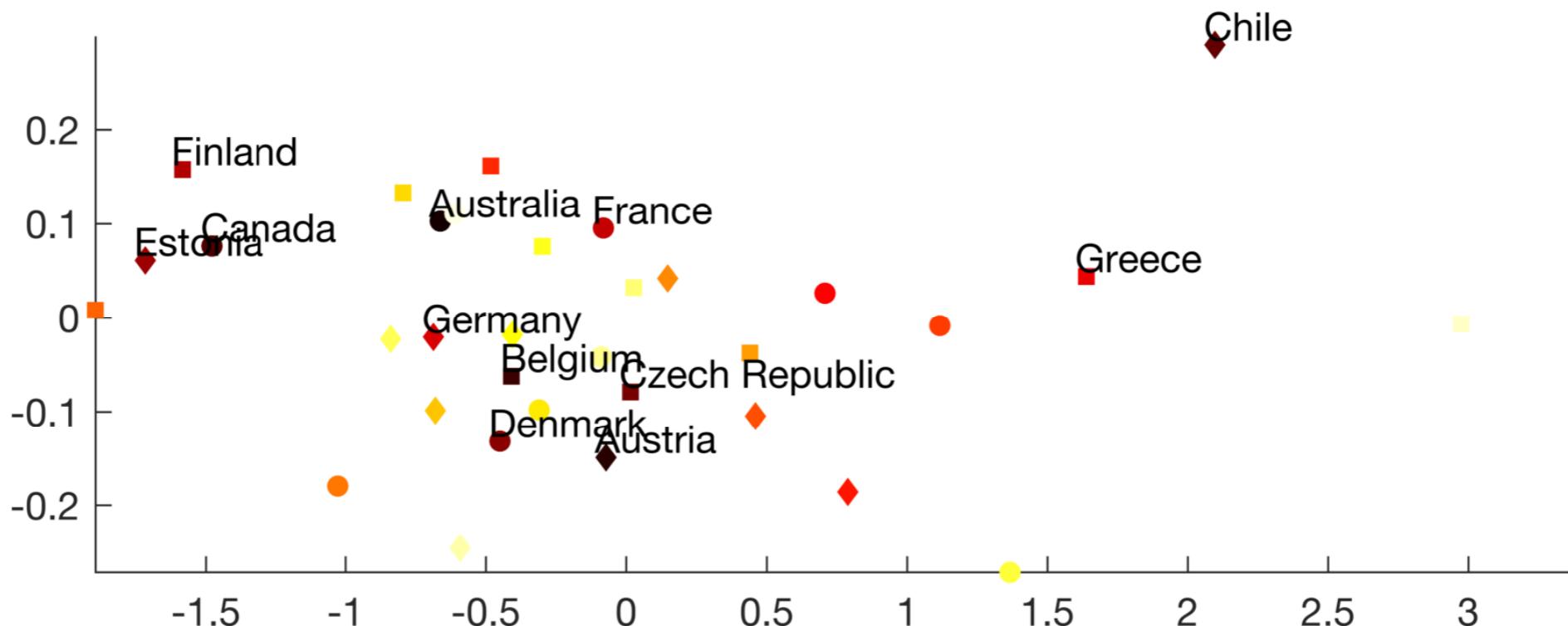


[Rolet'16]

Elliptical Embeddings

Multidimensional Scaling [MDS]

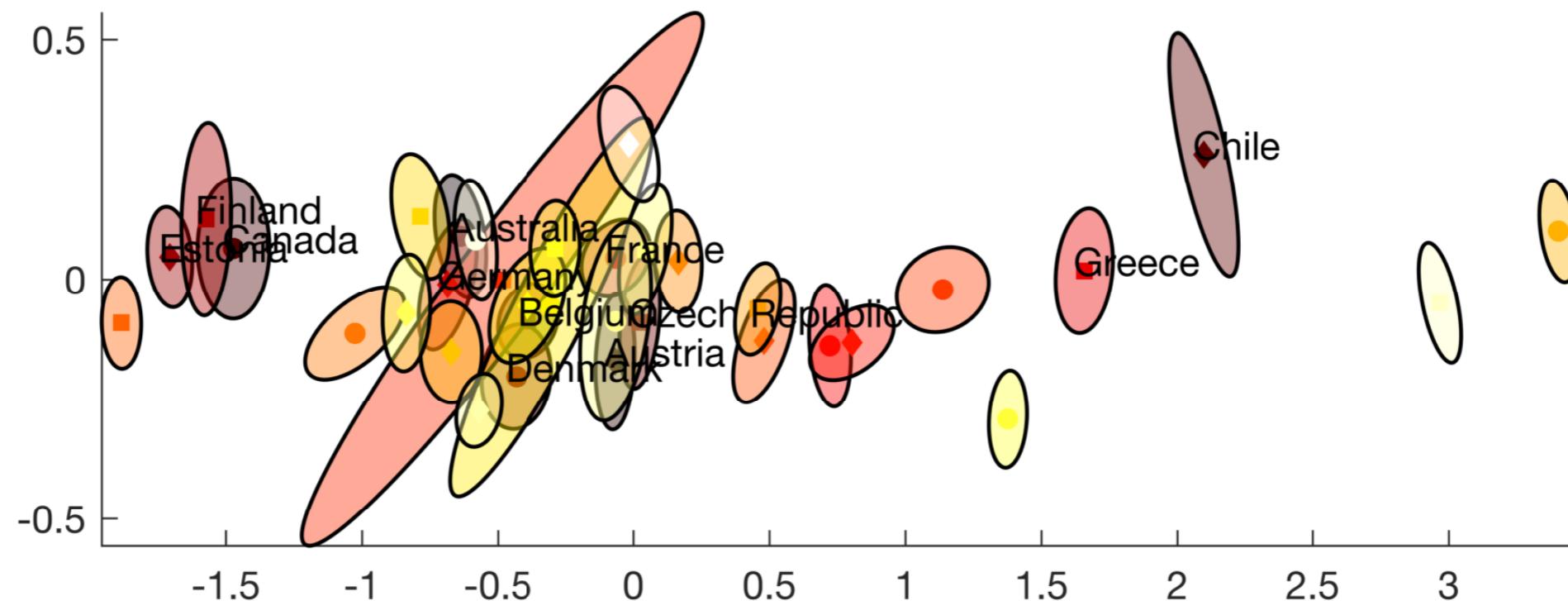
embed a metric space in R^2



Elliptical Embeddings

Multidimensional Scaling [MDS]

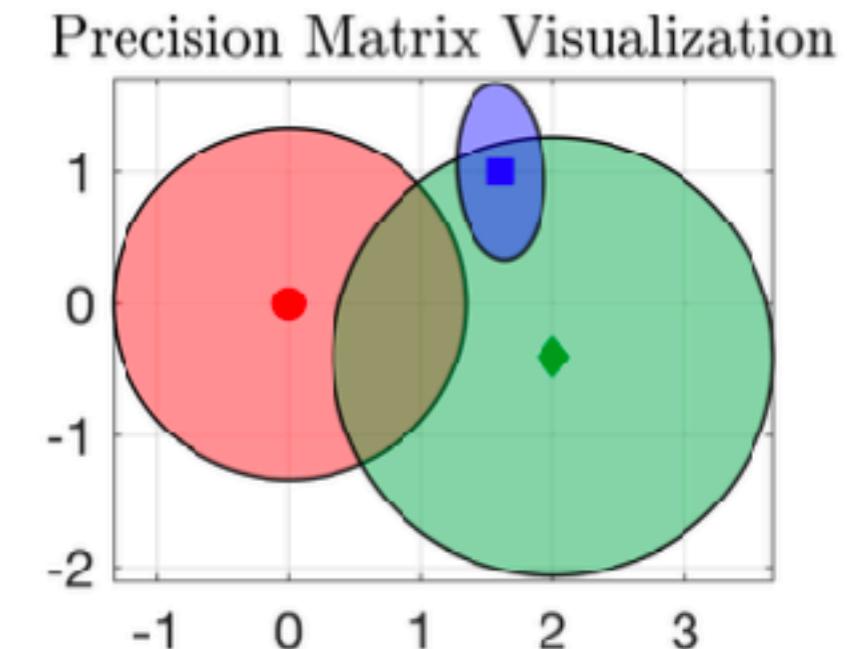
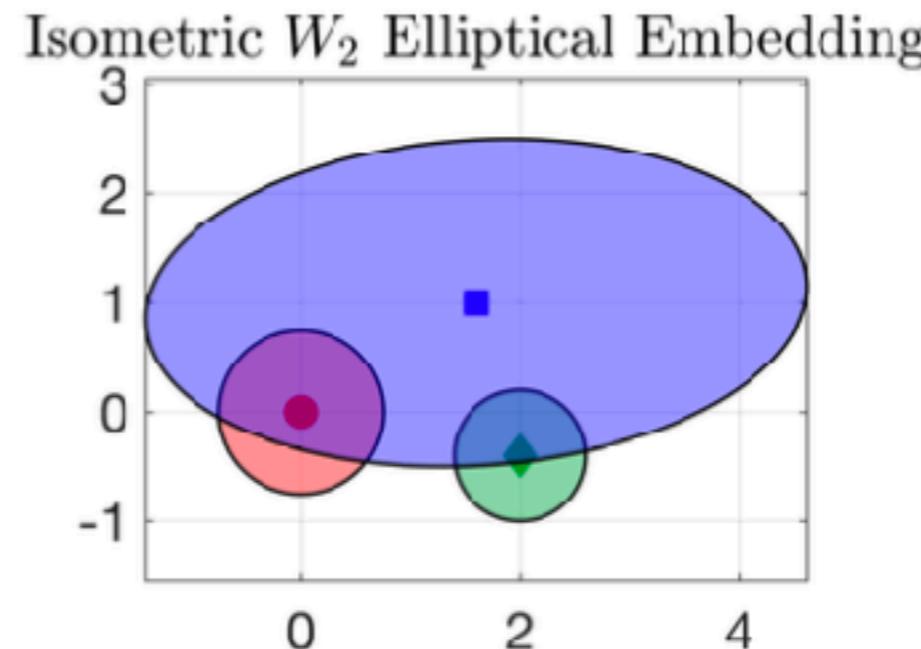
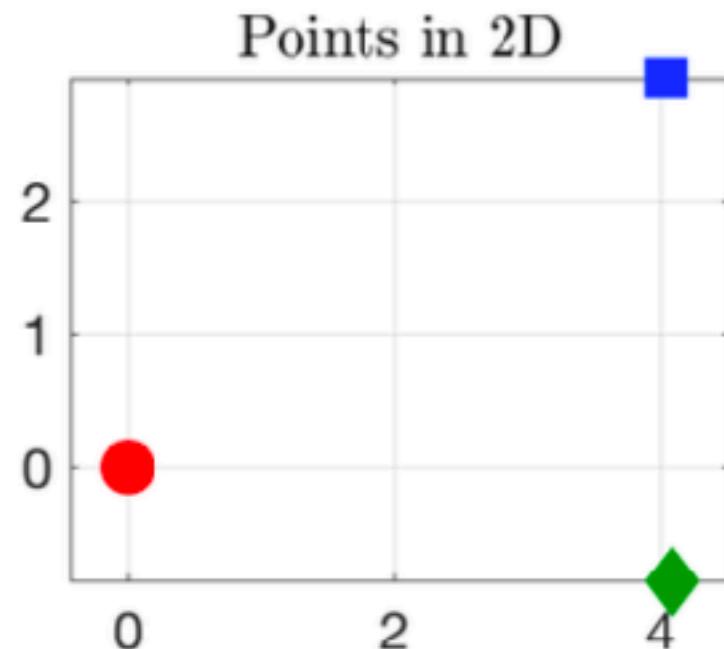
embed a metric space in elliptical distributions in $P(\mathbb{R}^2)$, W_2



Elliptical Embeddings

Visualization issue

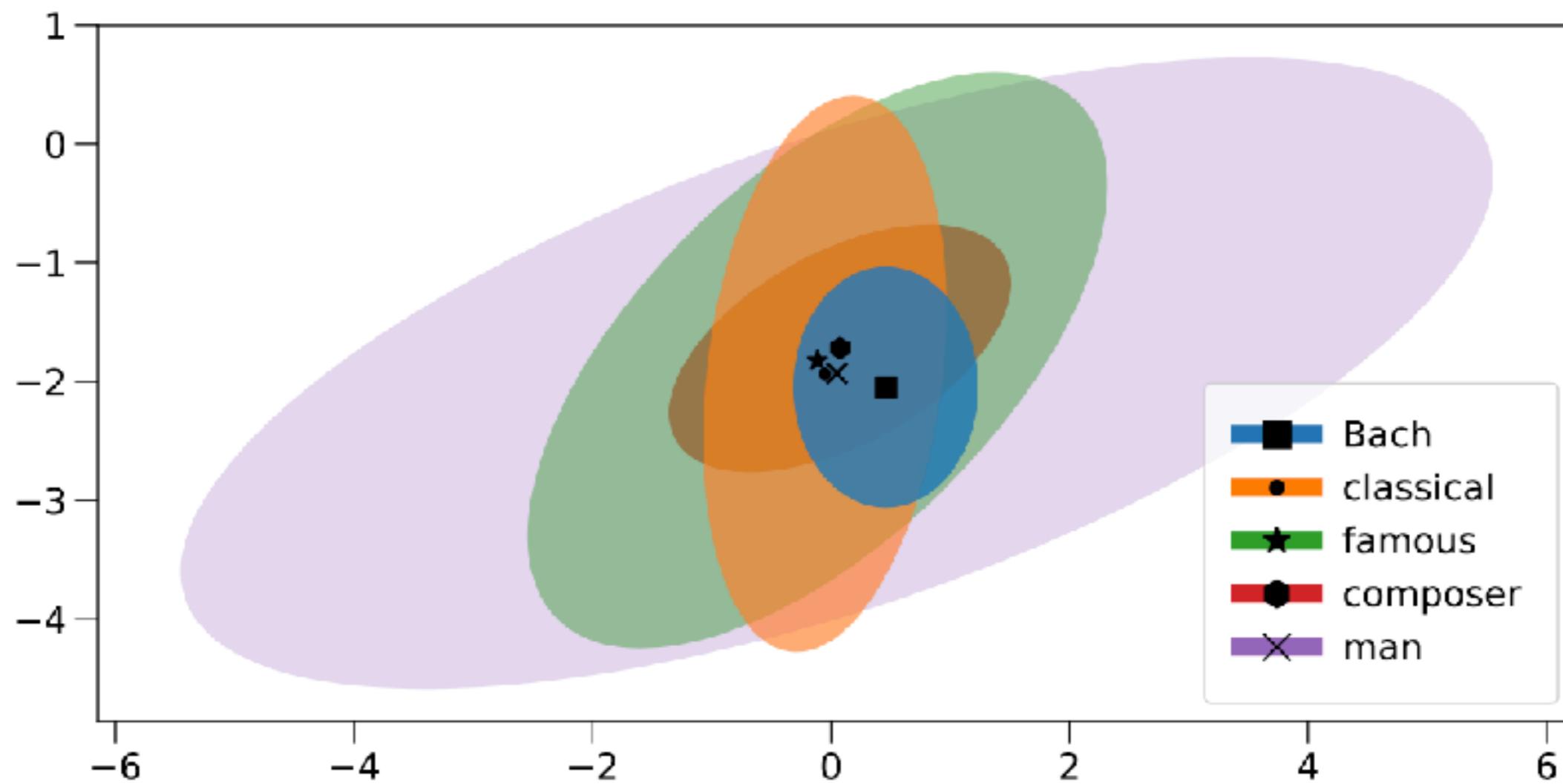
need to shift to precision matrix to recover intuition



Elliptical Embeddings

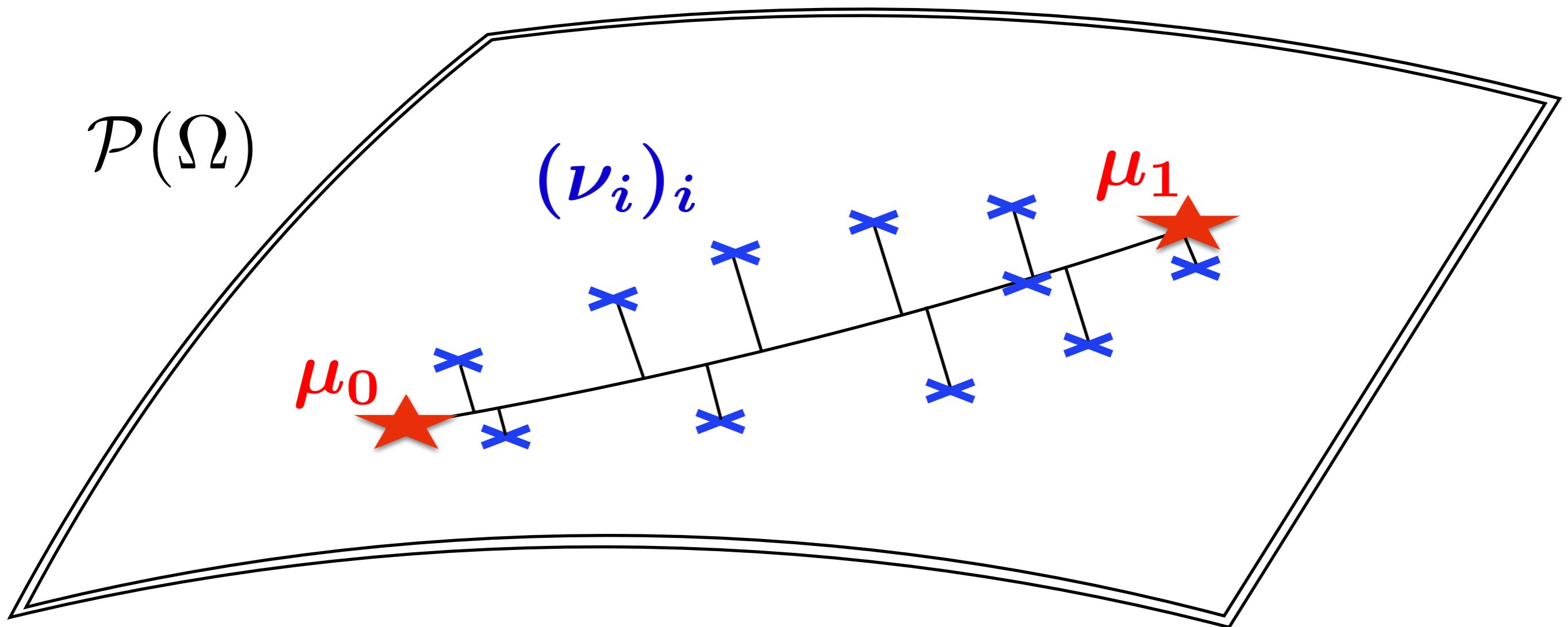
Word Embeddings

Compute elliptical distribution representations for Words

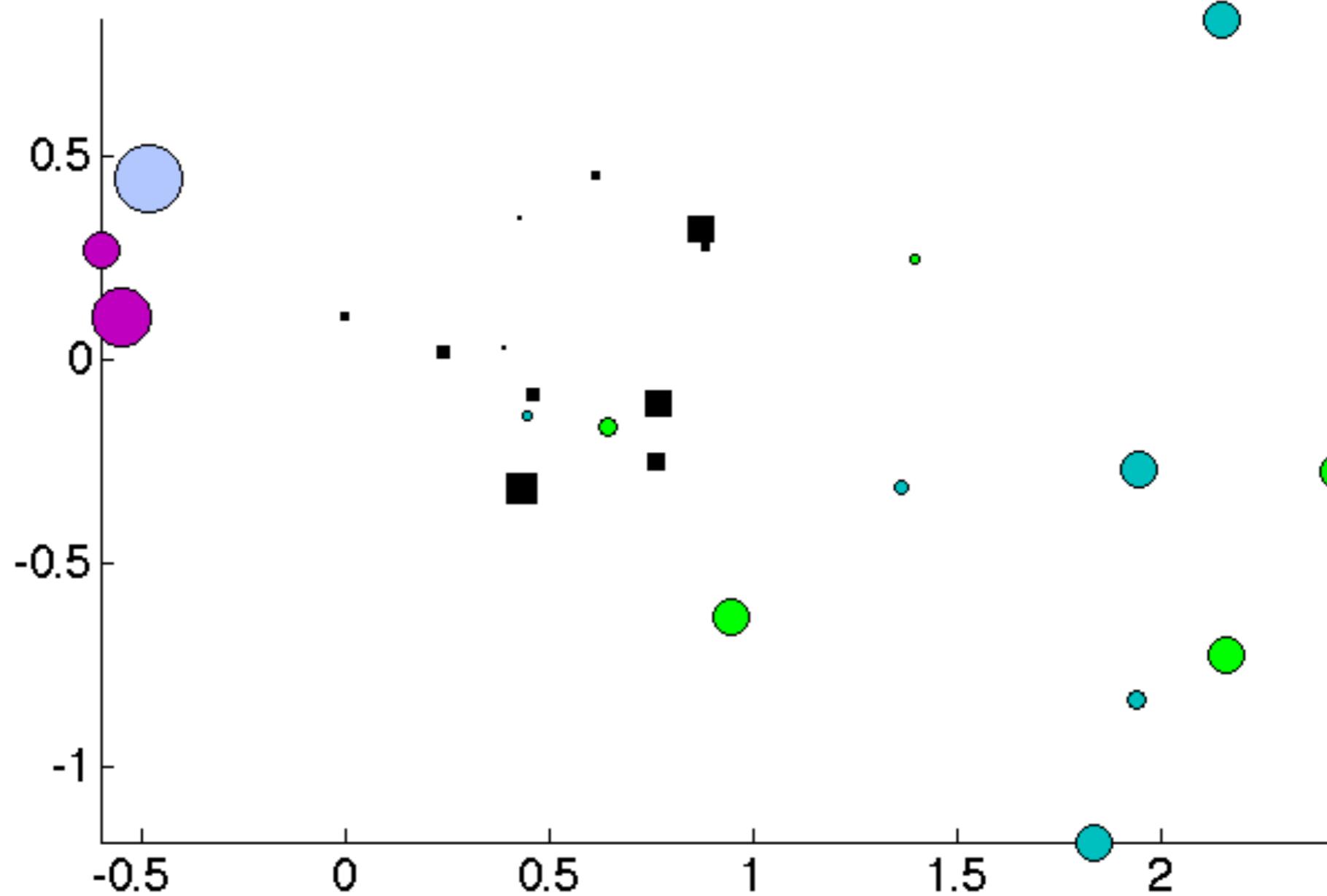


Wasserstein PCA

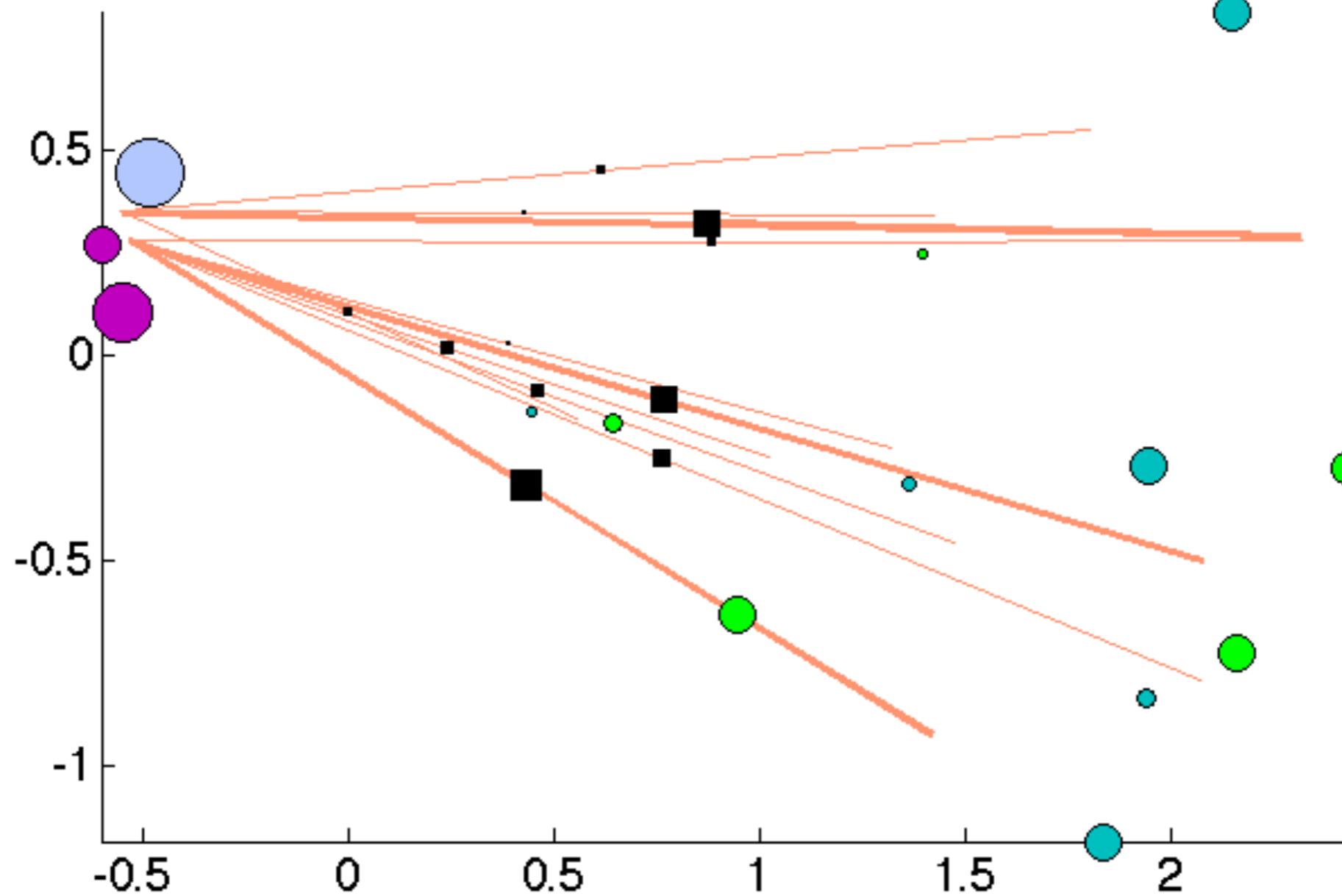
$$\min_{\mu_0, \mu_1} \sum_{i=1}^N \min_t W_2^2(\rho_{\mu_0 \rightarrow \mu_1}^t, \nu_i)$$



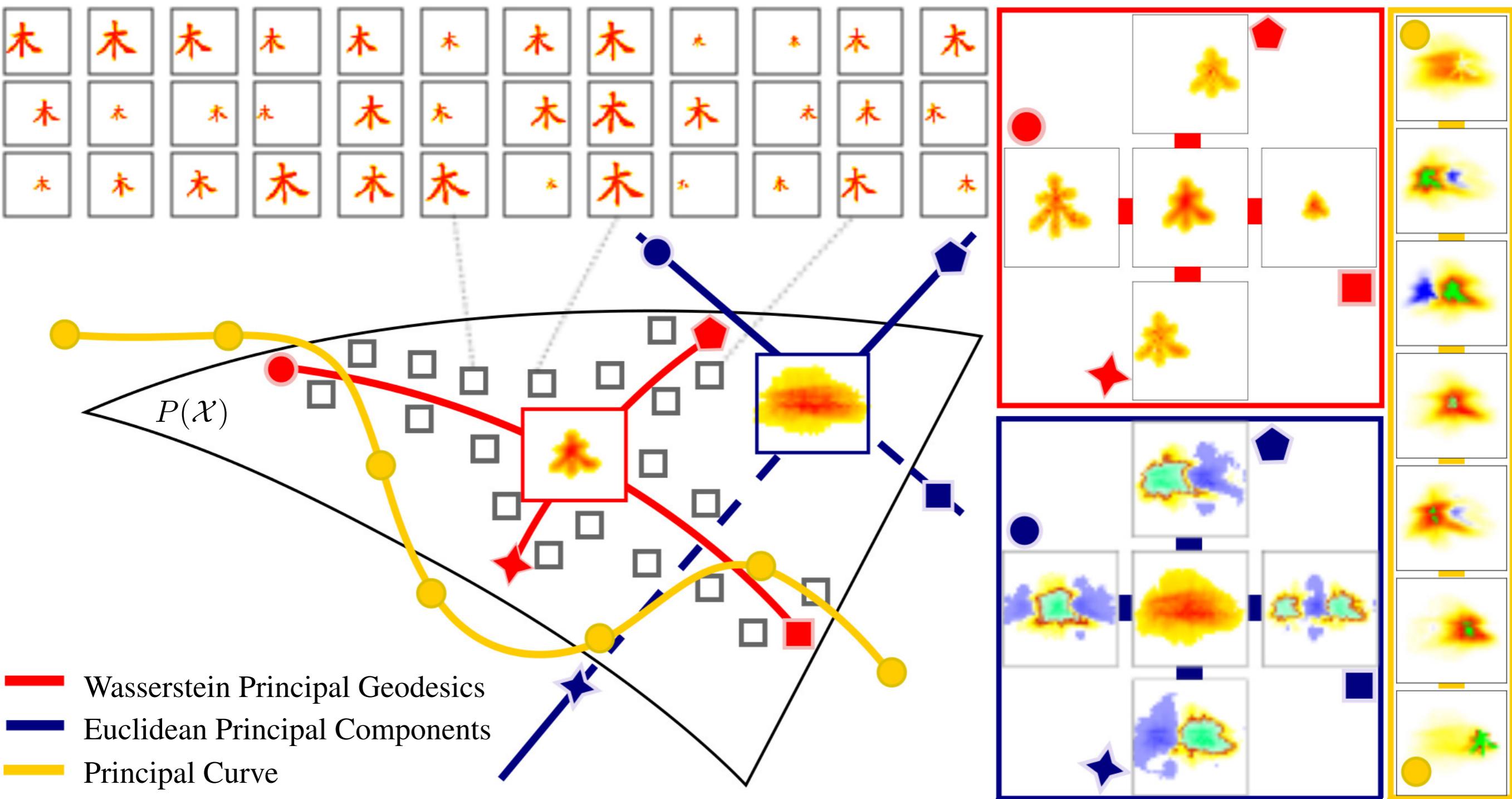
Wasserstein PCA



On Empirical Measures



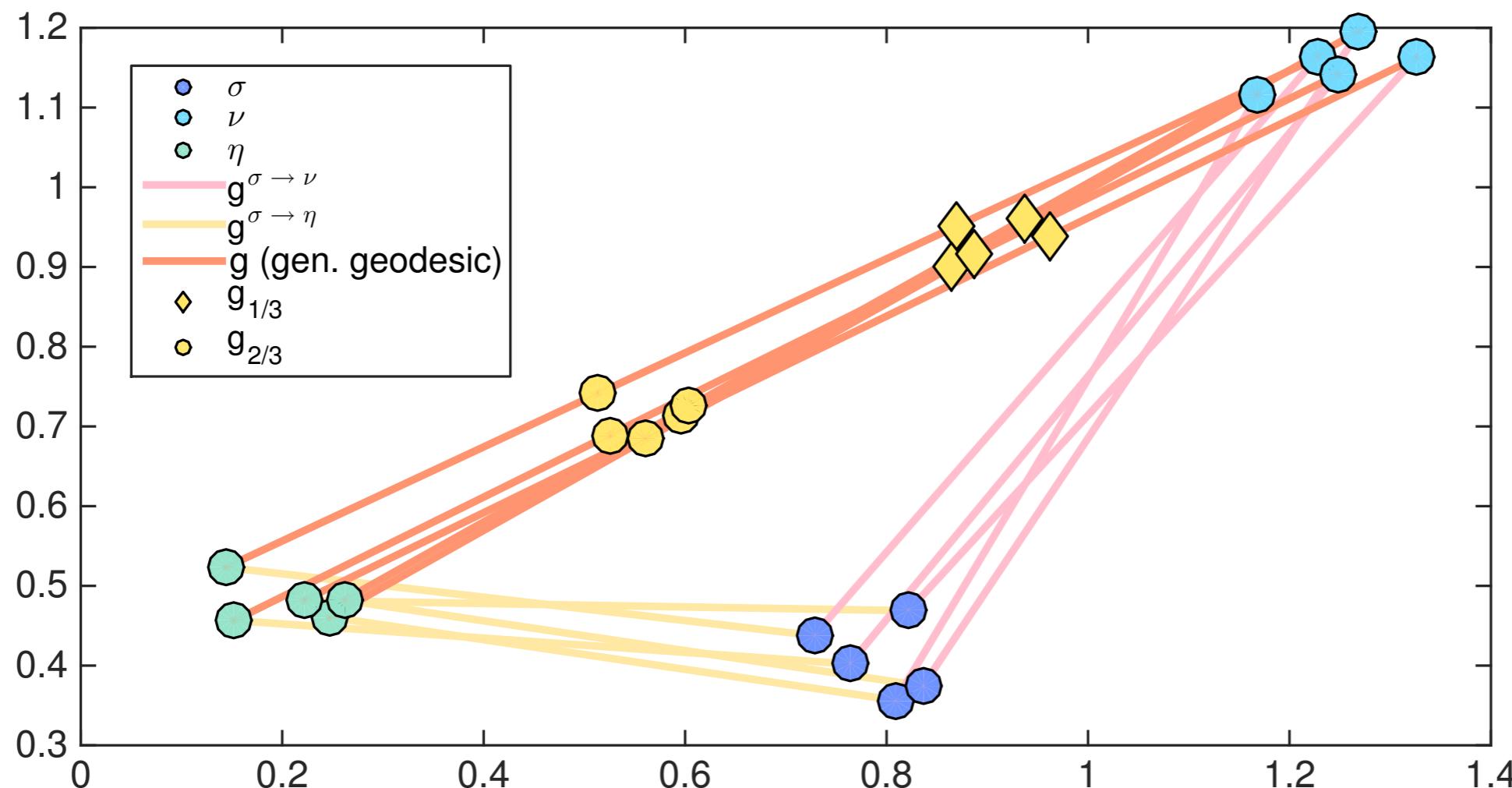
Wasserstein PCA vs. Euclidean PCA



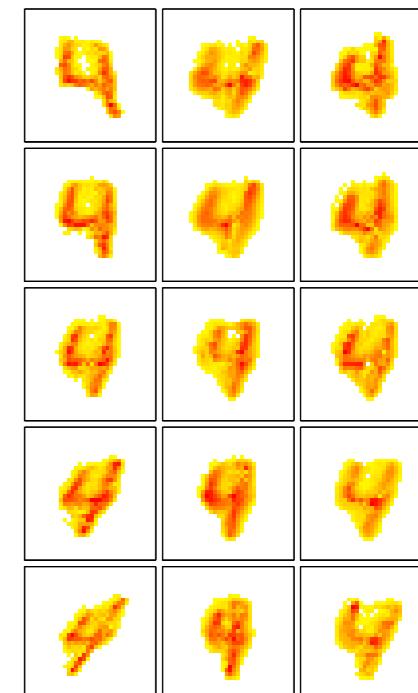
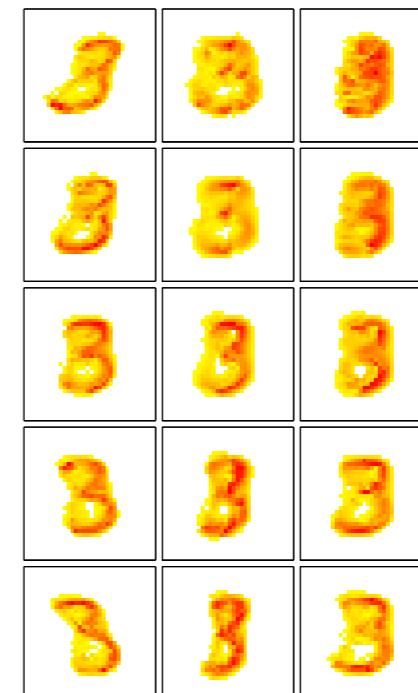
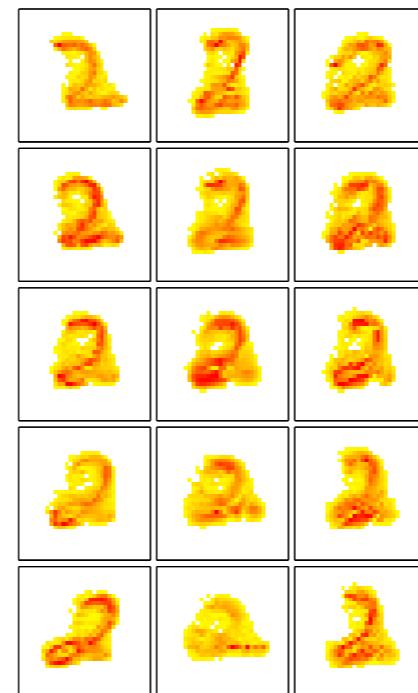
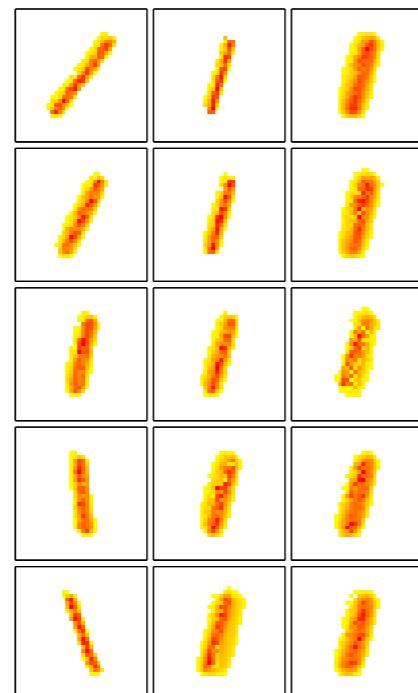
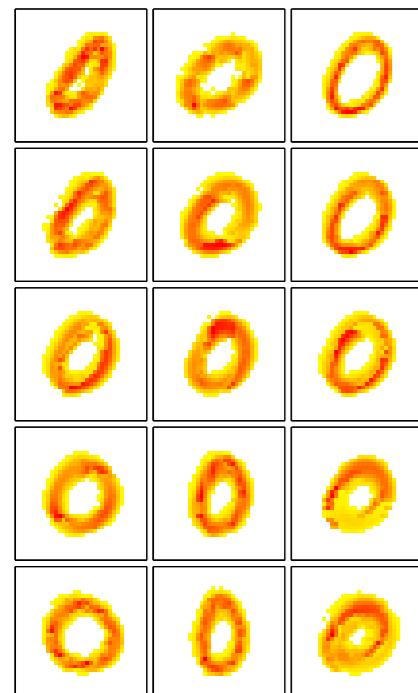
[Ambrosio'06] Generalized Geodesics

$$\min_{\mathbf{v}_1, \mathbf{v}_2 \in L^2(\bar{\nu}, \Omega)} \sum_{i=1}^N \min_{t \in [0,1]} W_2^2(g_t(\mathbf{v}_1, \mathbf{v}_2), \nu_i) + \lambda R(\mathbf{v}_1, \mathbf{v}_2),$$

subject to $\begin{cases} g_t(\mathbf{v}_1, \mathbf{v}_2) = (\text{Id} - \mathbf{v}_1 + t(\mathbf{v}_1 + \mathbf{v}_2)) \# \bar{\nu} \\ \text{Id} - \mathbf{v}_1 \text{ and } \text{Id} + \mathbf{v}_2 \text{ are Monge maps from } \bar{\nu} \end{cases}$



Generalized Principal Geodesics



For each digit, 1,000 MNIST images

[Seguy'15

Inverse Wasserstein Problems

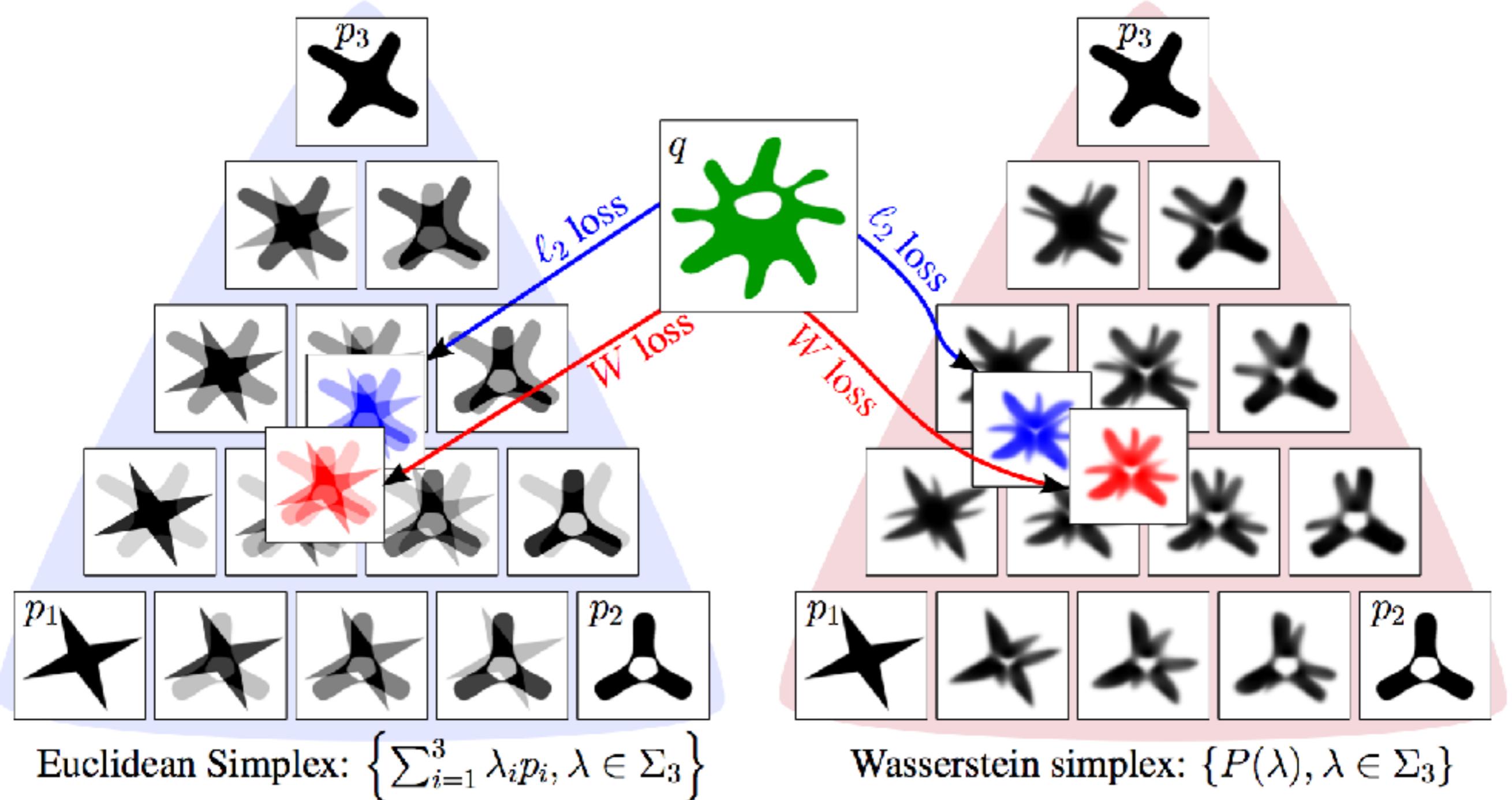
- consider Barycenter operator:

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

- 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

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

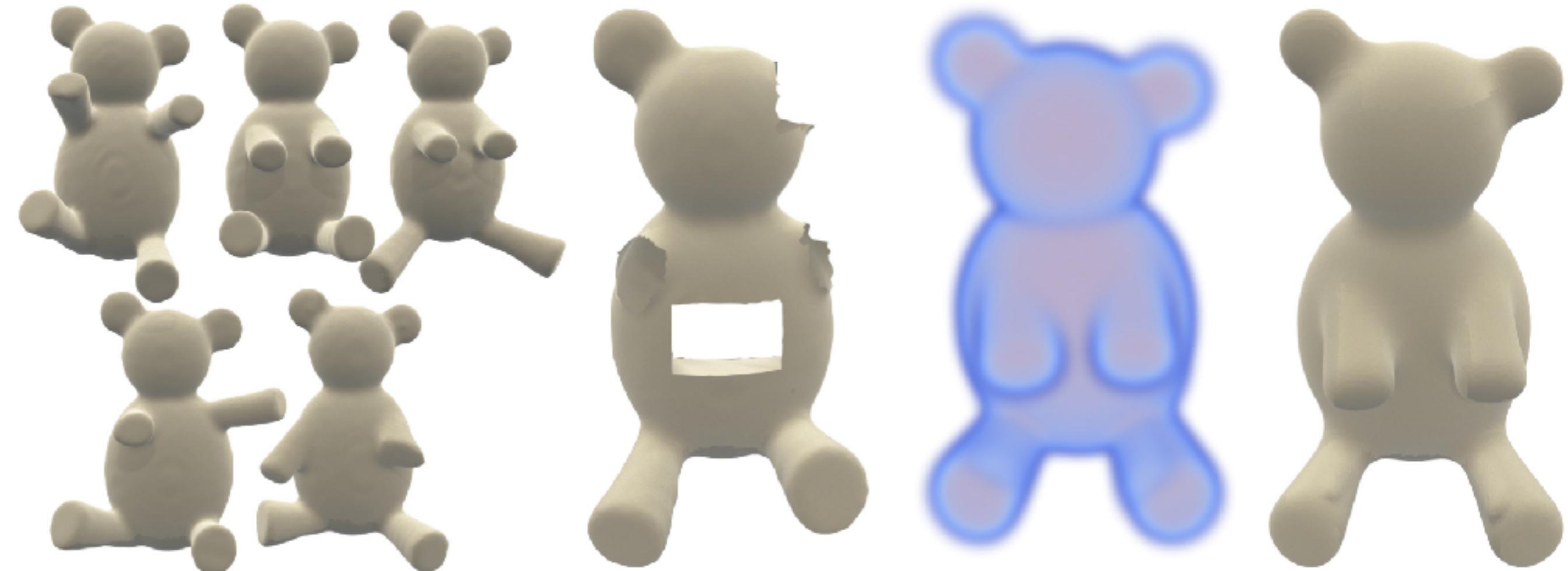
- Different using **the chain rule.**

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

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



$\lambda_0 = 0.03$



$\lambda_1 = 0.12$



$\lambda_2 = 0.40$

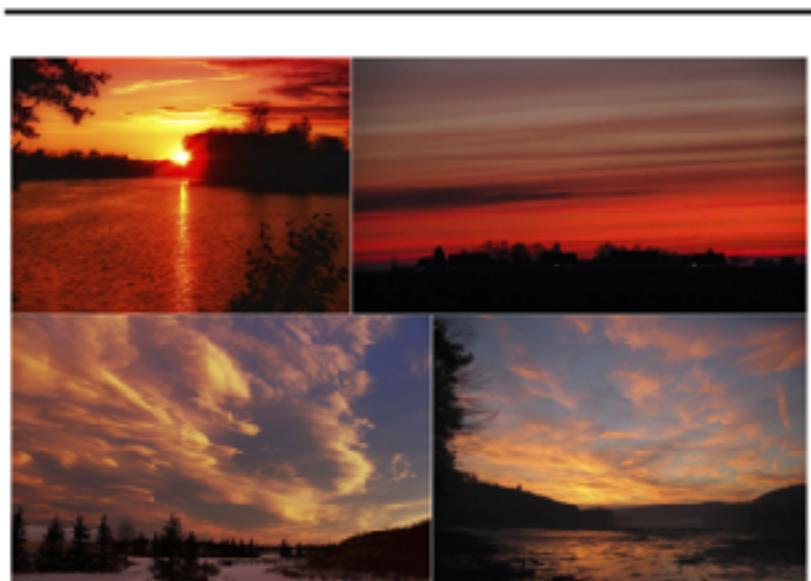
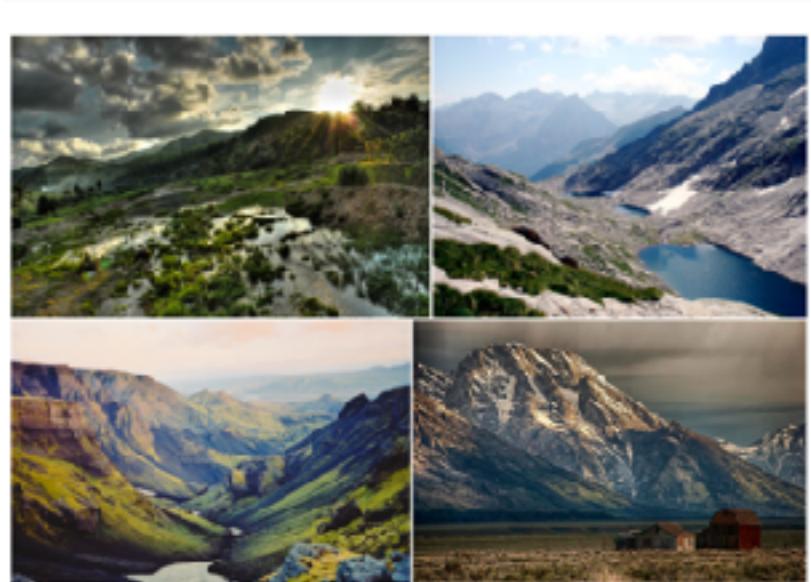


$\lambda_3 = 0.43$

Application: Color Grading



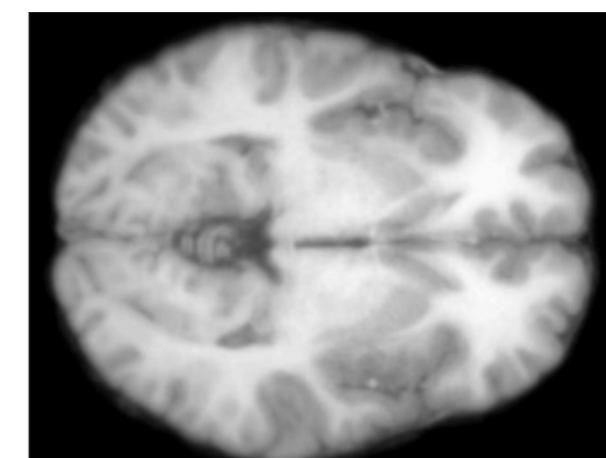
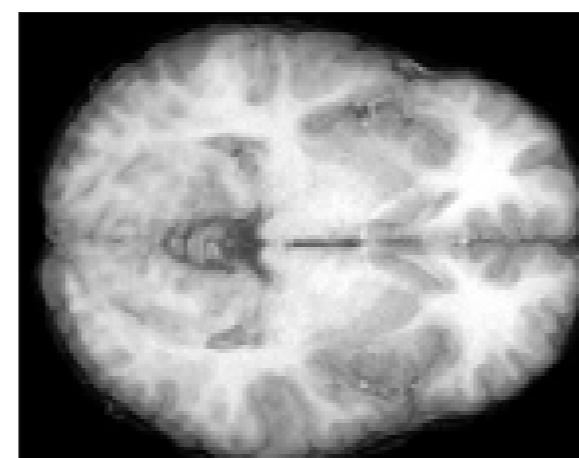
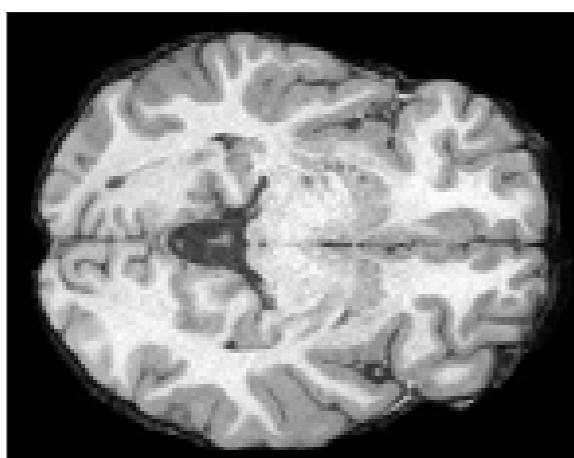
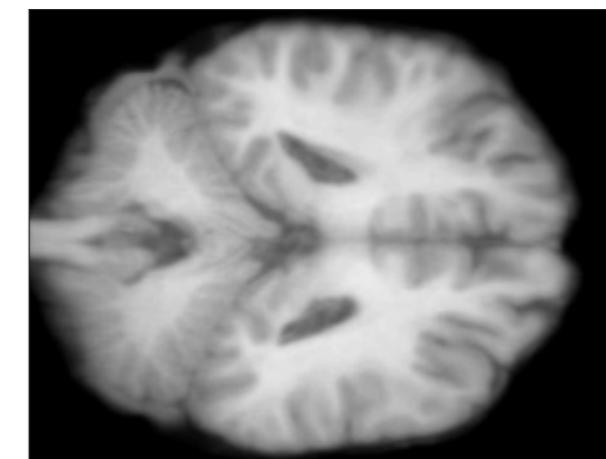
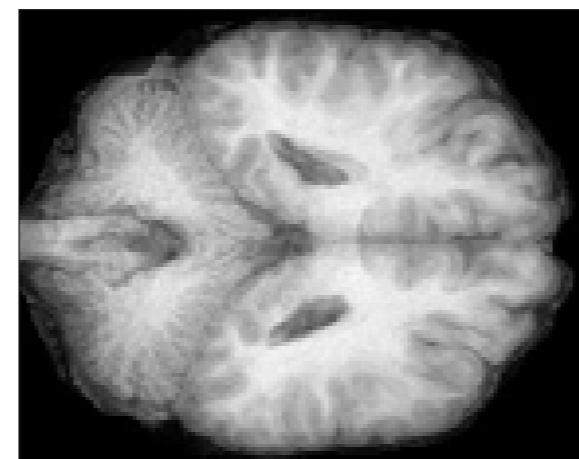
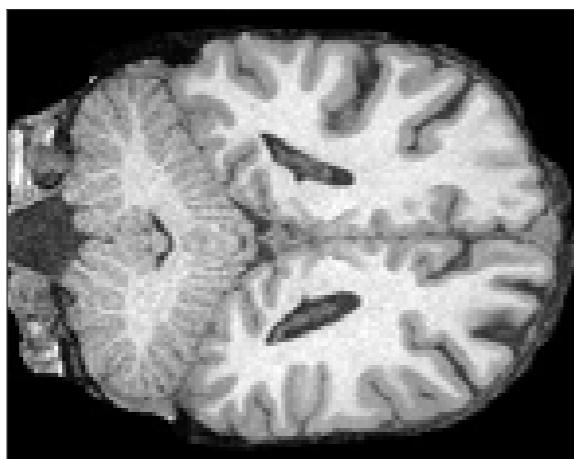
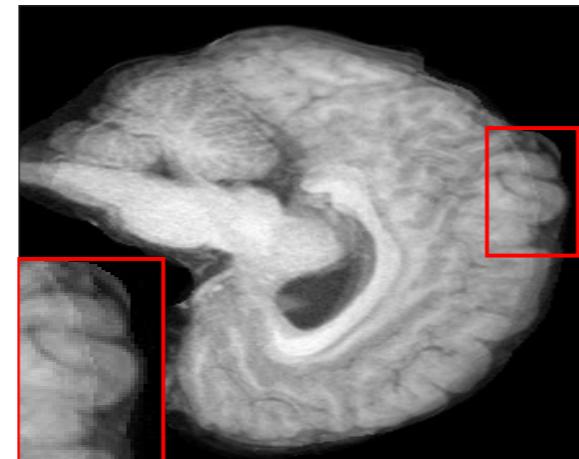
Application: Color Grading



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

[BPC'16]

Application: Brain Mapping

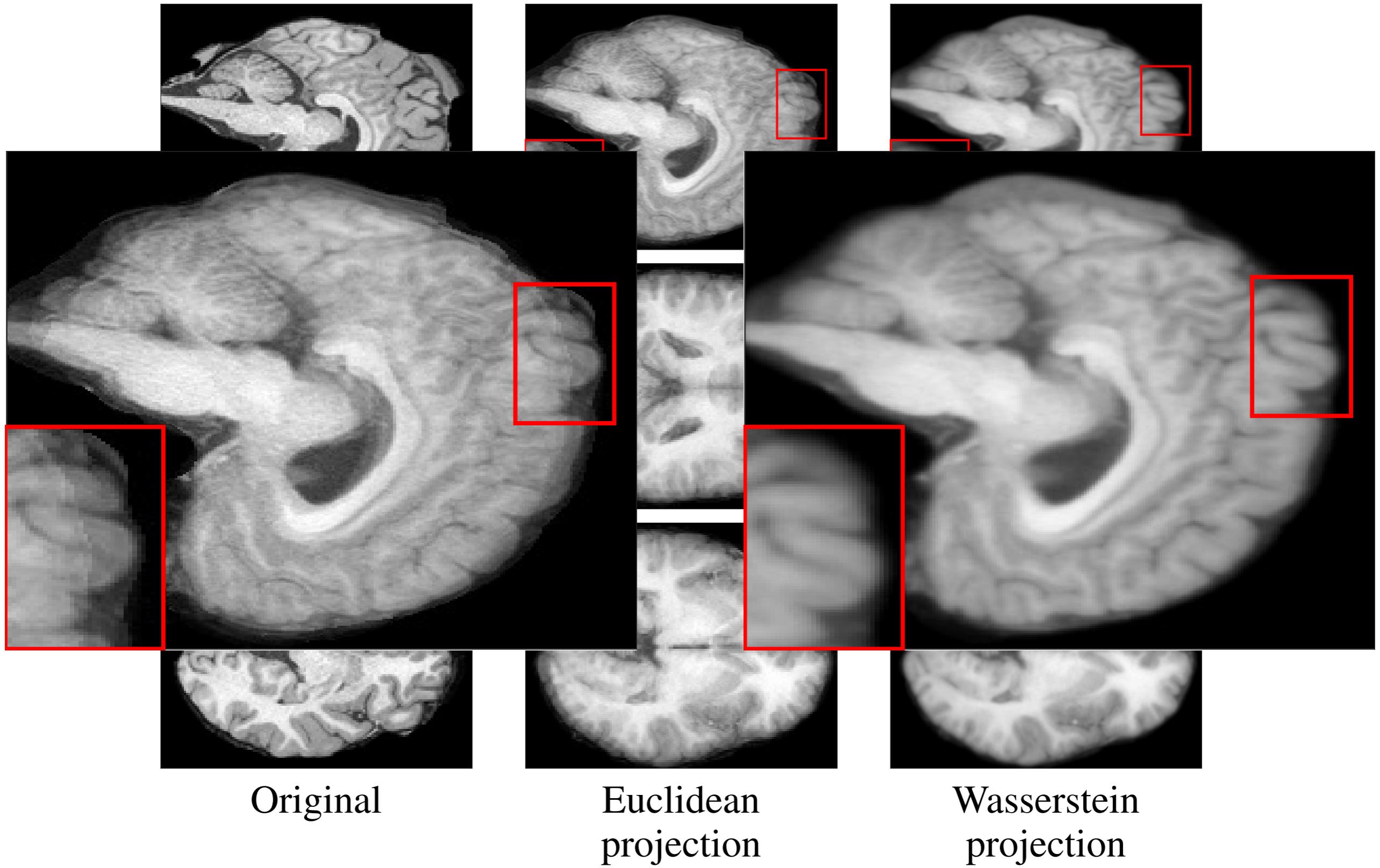


Original

Euclidean
projection

Wasserstein
projection

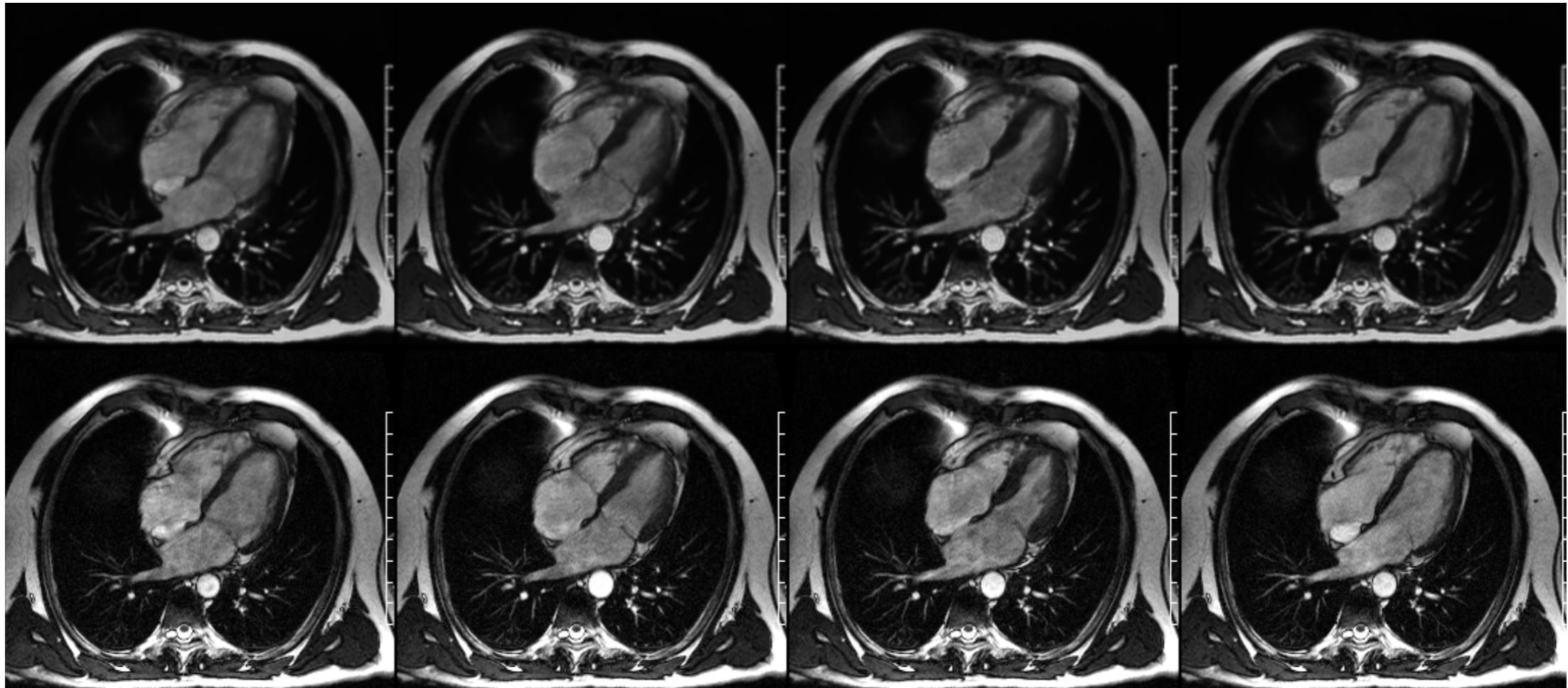
Application: Brain Mapping



end-to-end W Dictionary Learning

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

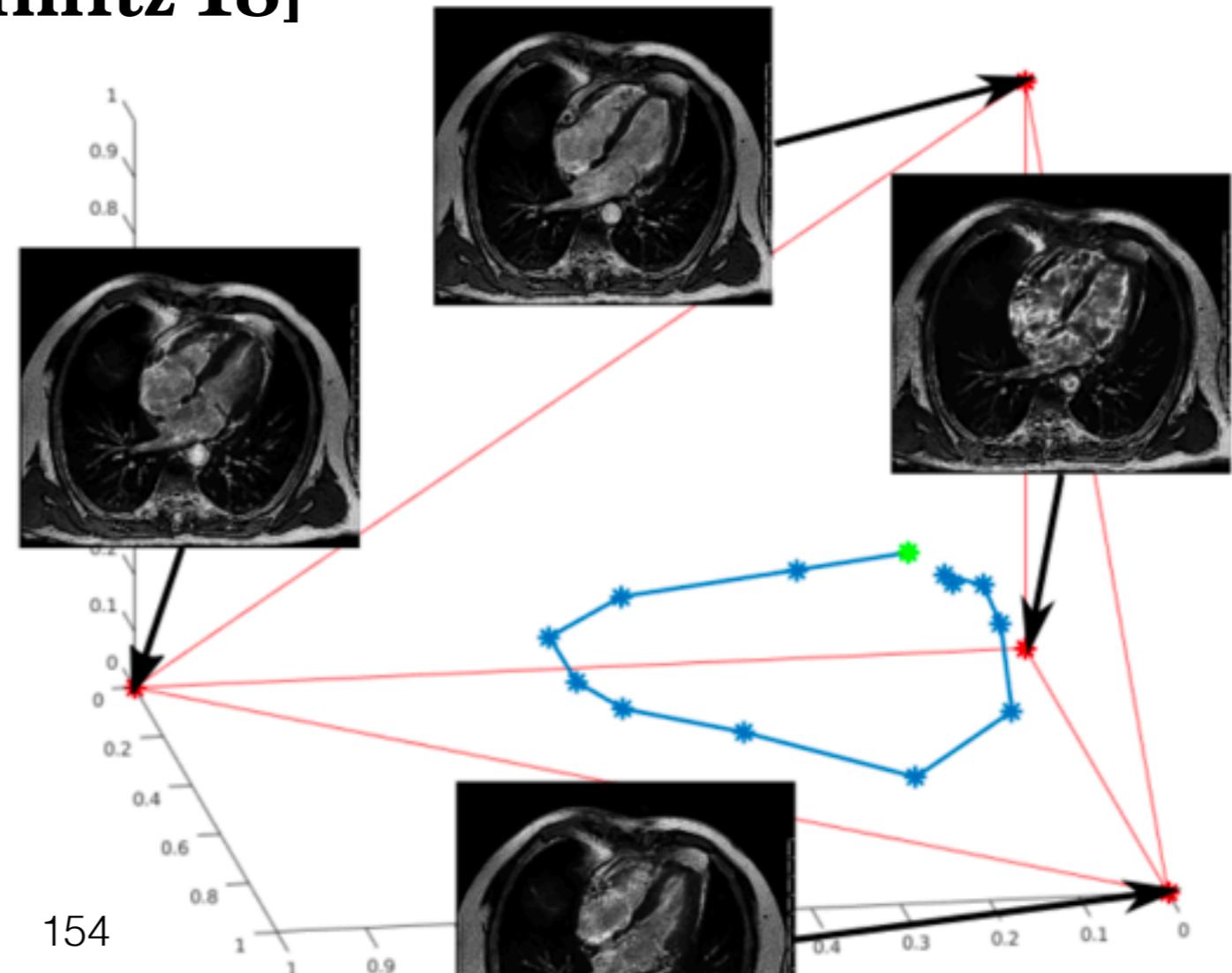
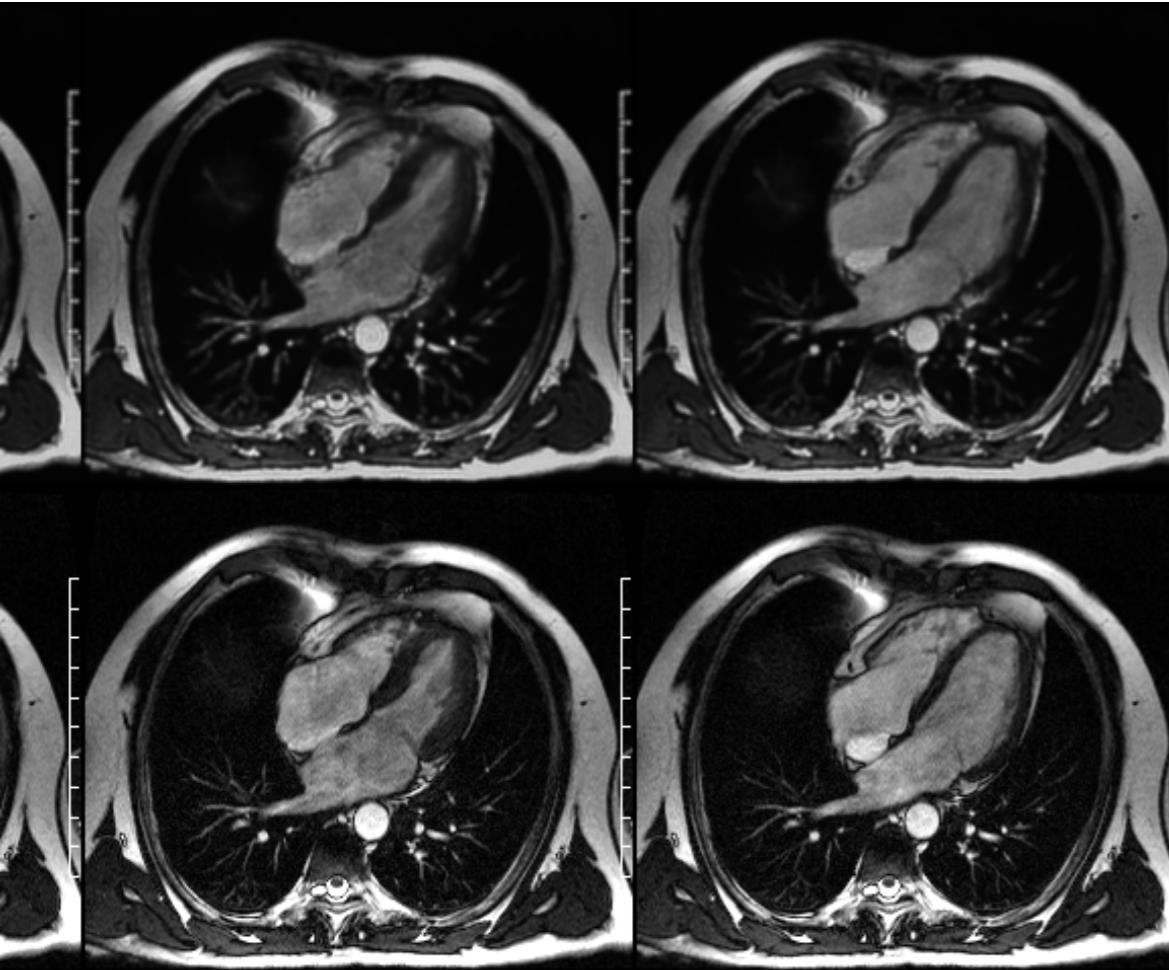
[Schmitz'18]



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]

Distributionally Robust Learning

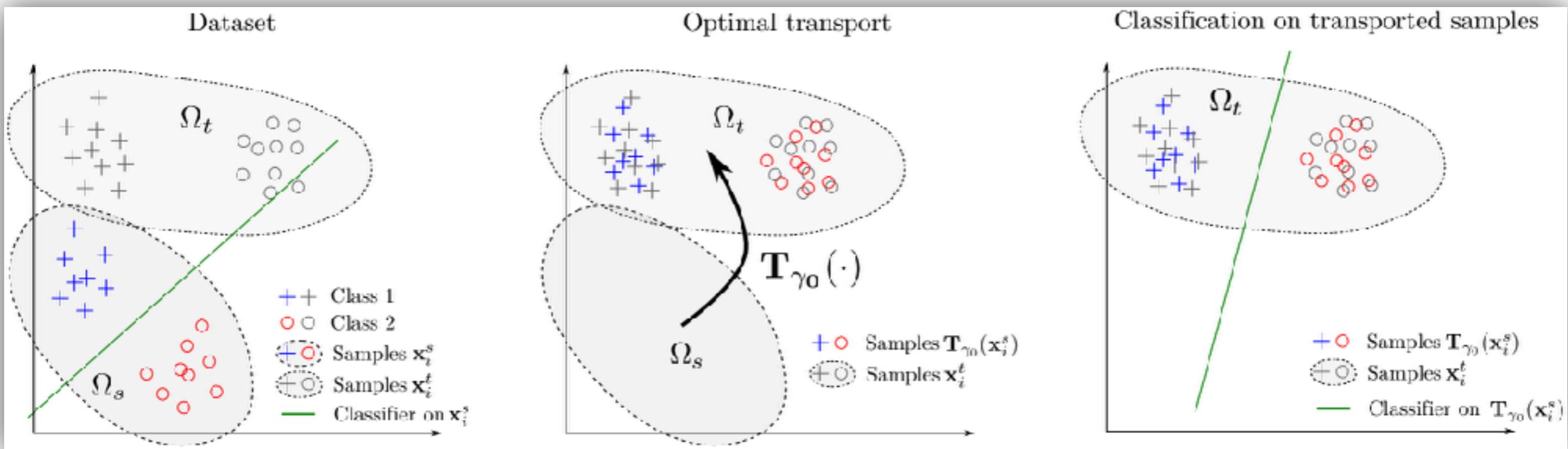
Learning with Wasserstein Ambiguity

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

Advantages:

- Bound on out-of-sample performance
- Converges as size of dataset increases
- Often reduces to a finite convex program (e.g. when f is element-wise max over elementary concave functions)

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]

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.

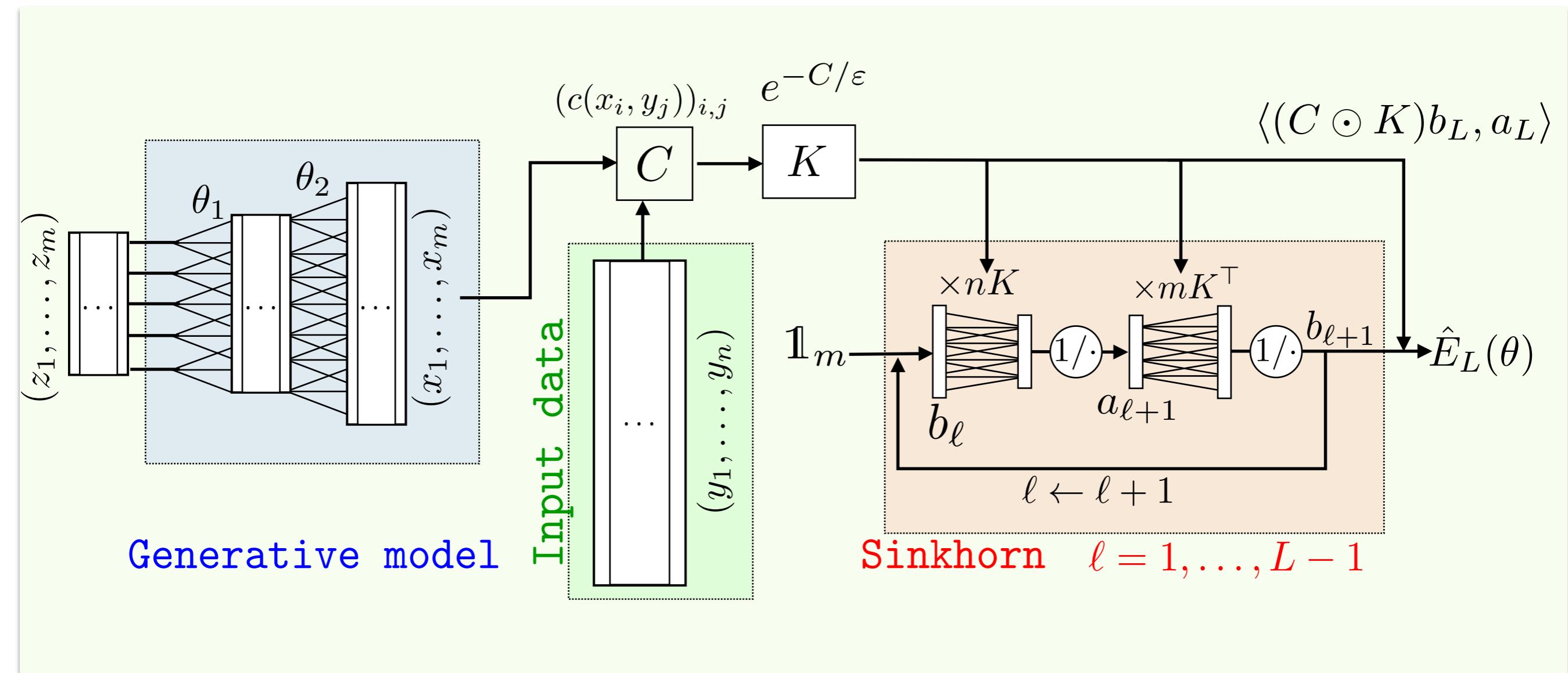
[Arjovsky'17] (WGAN) uses a NN to approximate dual solutions and recover gradient w.r.t. parameter

[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.

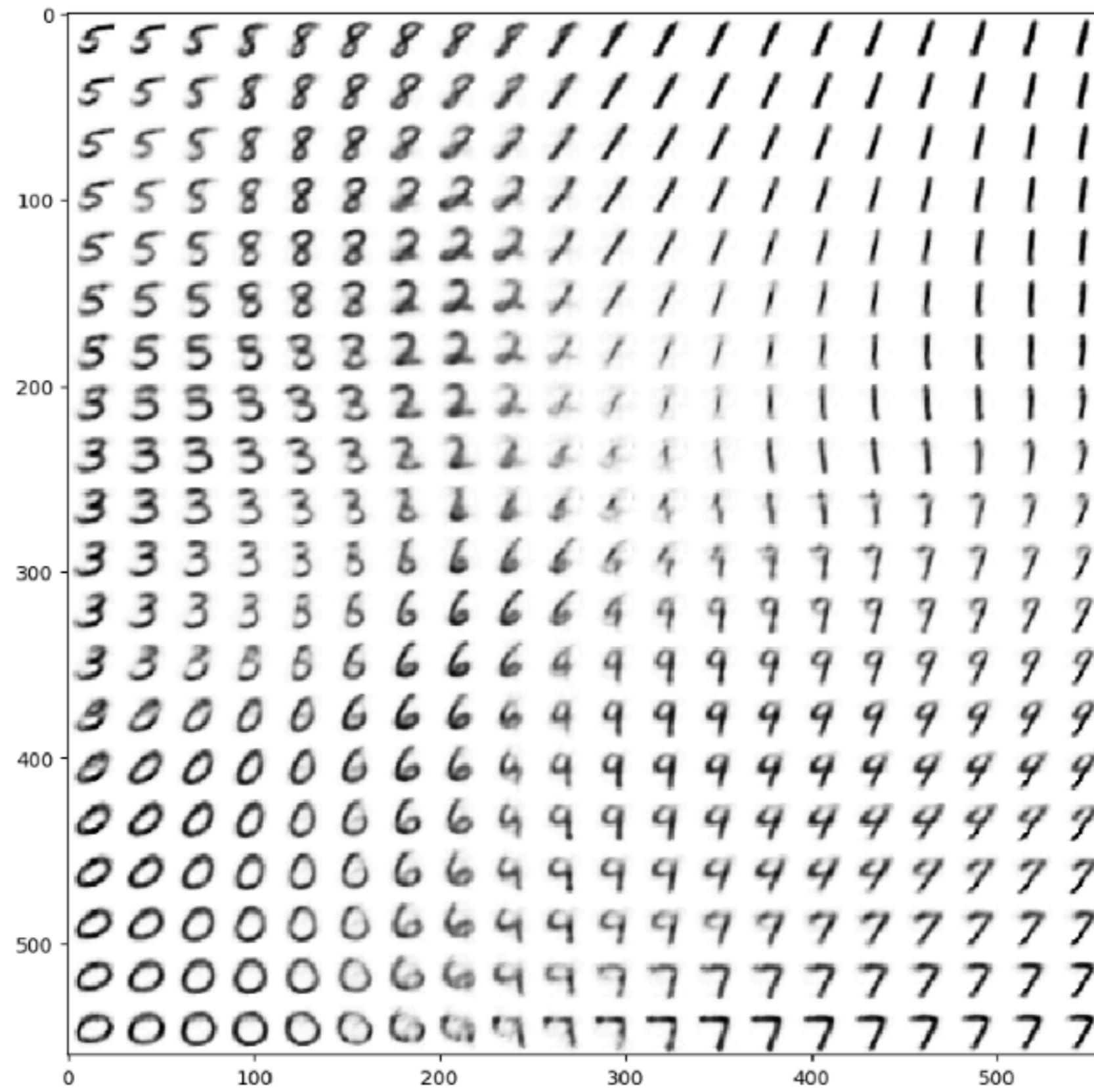


Example: MNIST, Learning f_θ

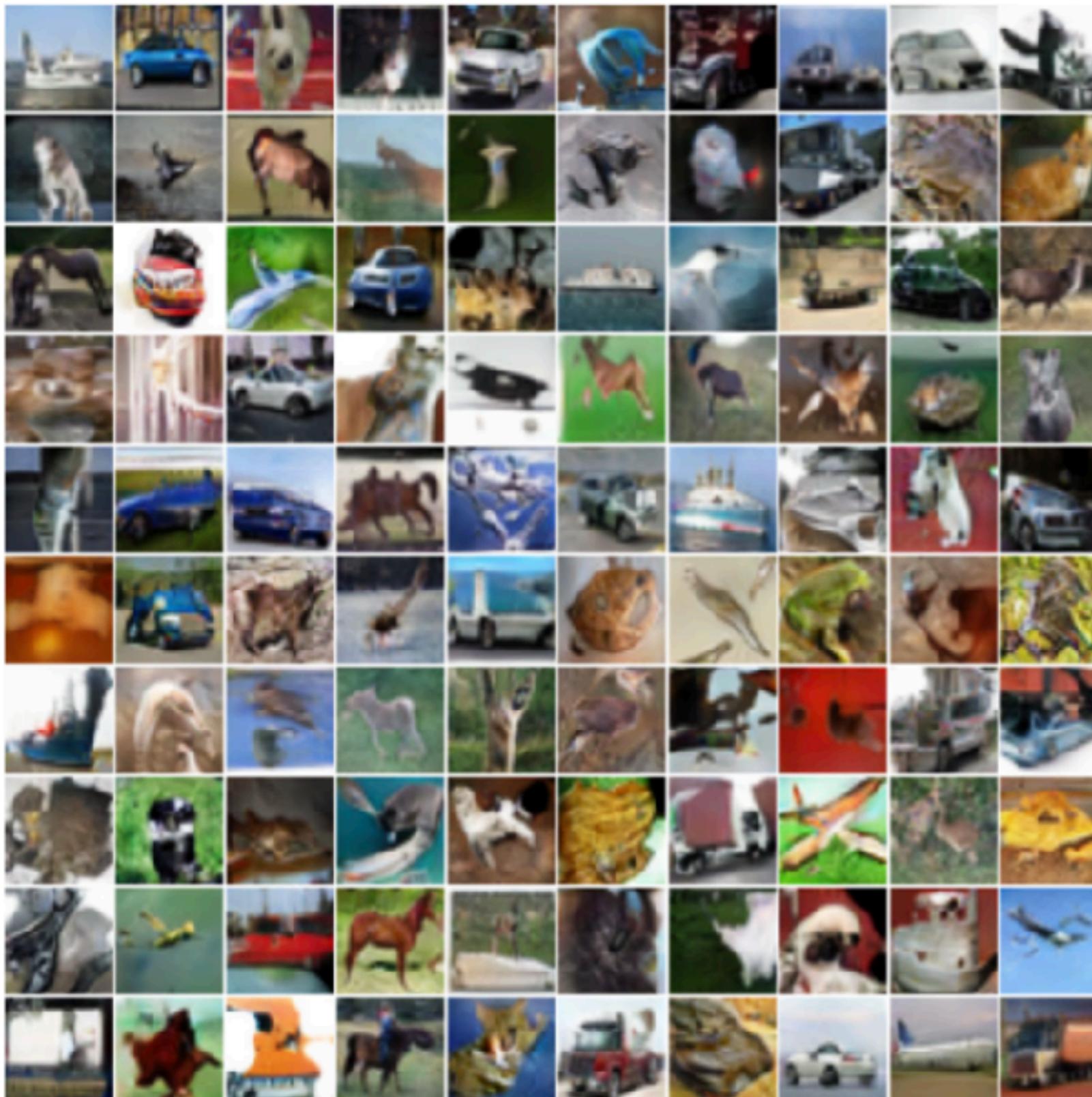
0000000000000000
1111111111111111
22222222222222223
33333333333333333
4444444444444444
55555555555555555
66666666666666666
77777777777777777
88888888888888888
99999999999999999

Example: MNIST, Learning f_{θ}

Latent
space
 $[0, 1]^2$



Example: Generation of Images



Example: Generation of Images



Concluding Remarks

- *Regularized OT* is much faster than OT.
- *Regularized OT* can interpolate between W and the MMD / *Energy distance* (MMD) metrics.
- The solution of *regularized OT* is “*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
- Applications to sampling.

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