

Generative Models and Optimal Transport

Marco Cuturi



Joint work with

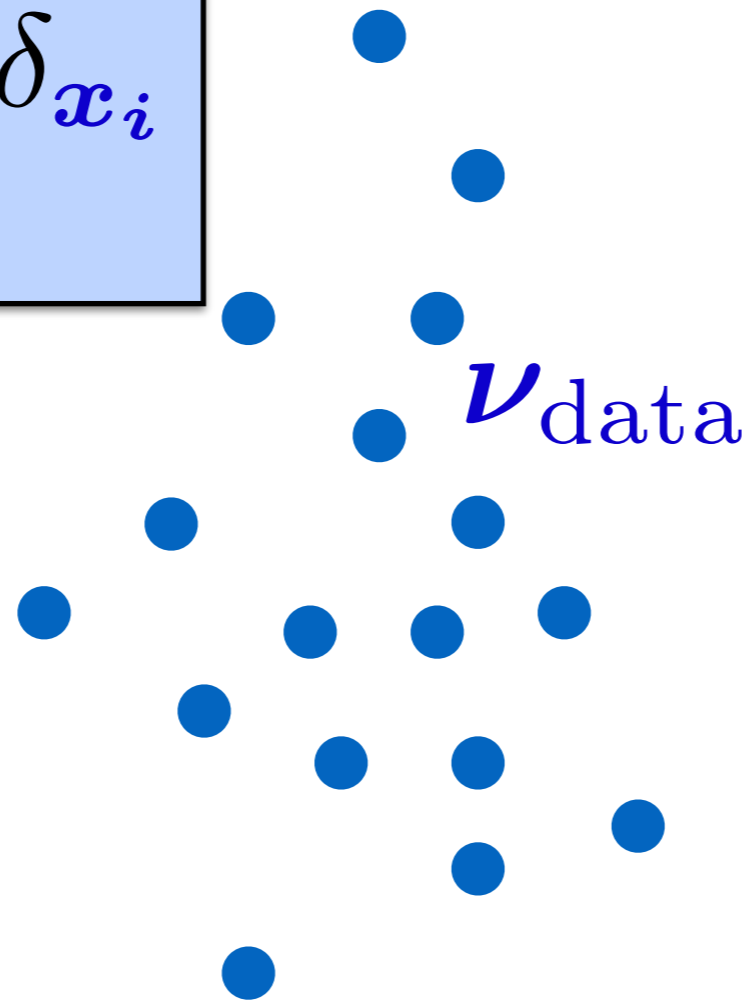
G. Peyré, A. Genevay (*ENS*)

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

Statistics 0.1 : Density Fitting

We collect data

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



Statistics 0.1 : Density Fitting

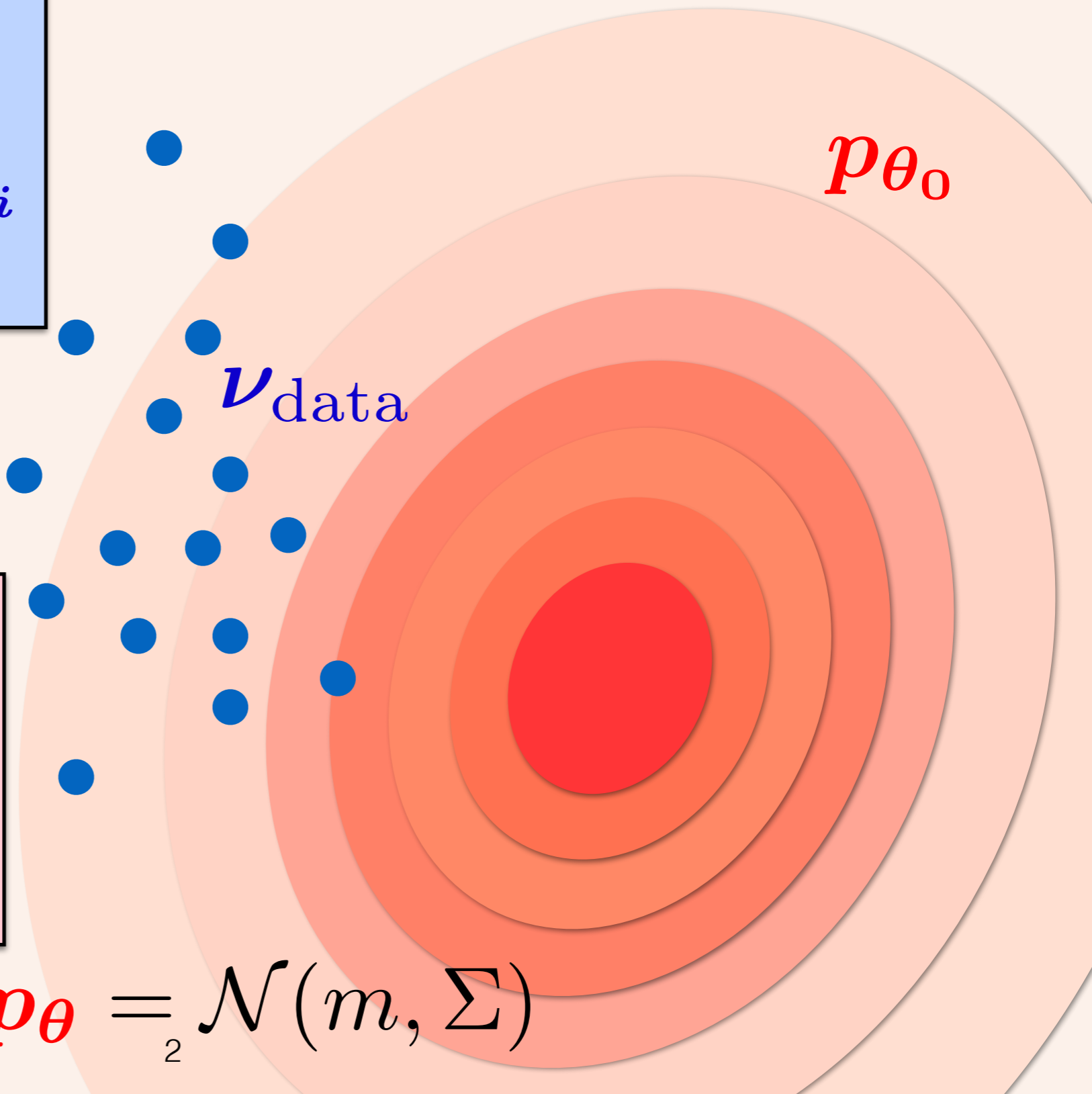
We collect data

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

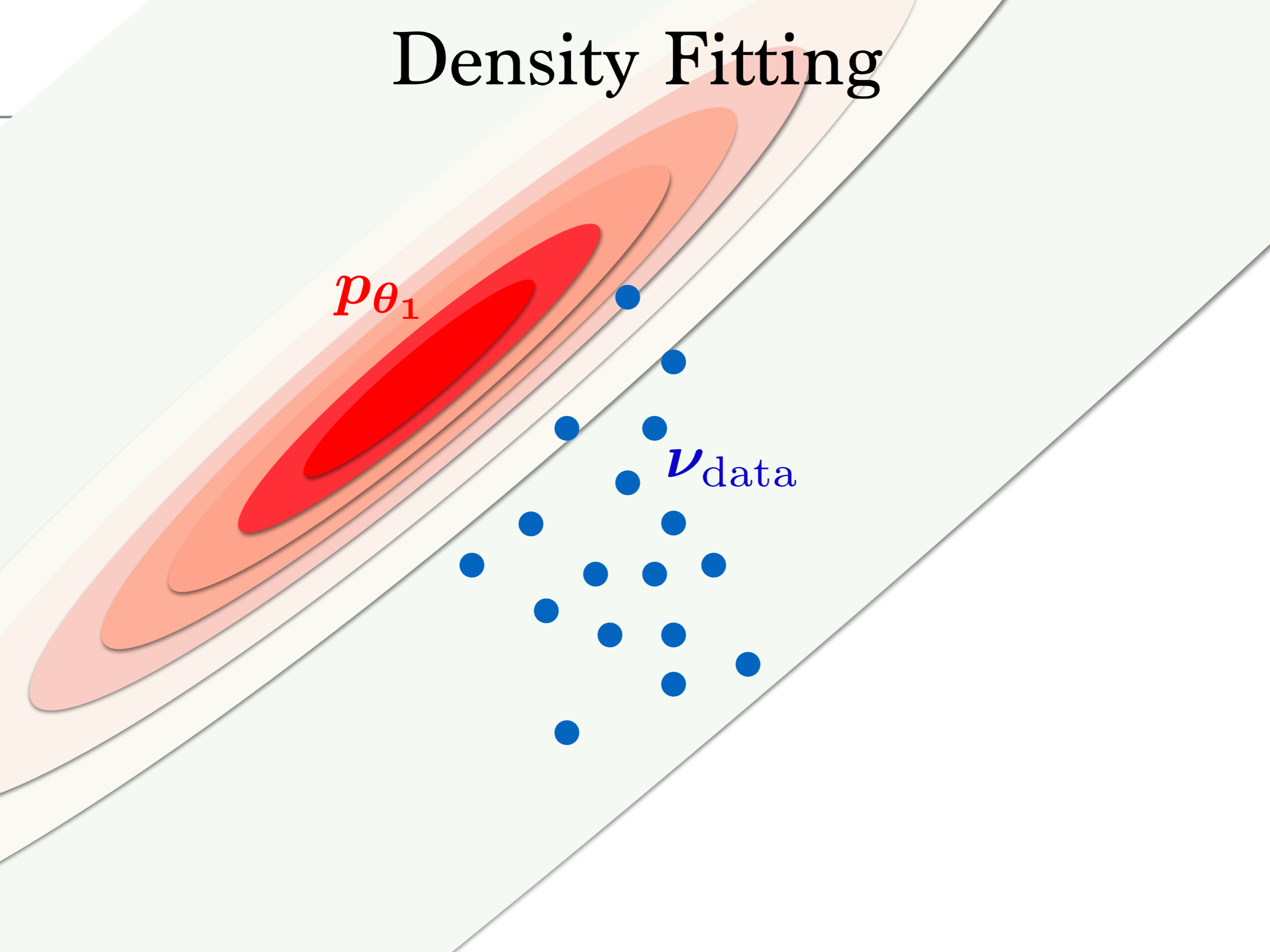
We fit a parametric family of densities

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

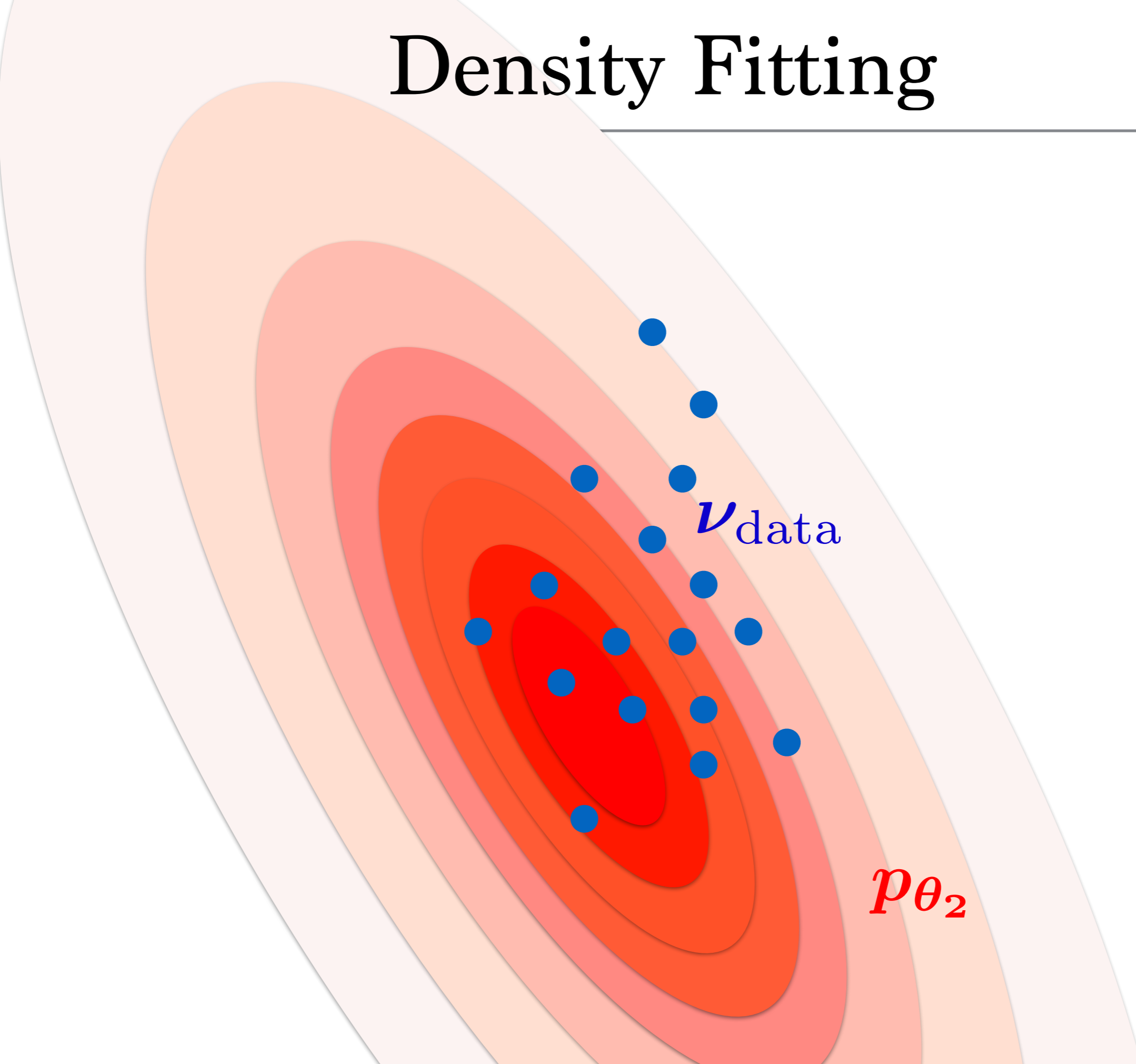
e.g. $\theta = (m, \Sigma); p_{\theta} = \mathcal{N}(m, \Sigma)$



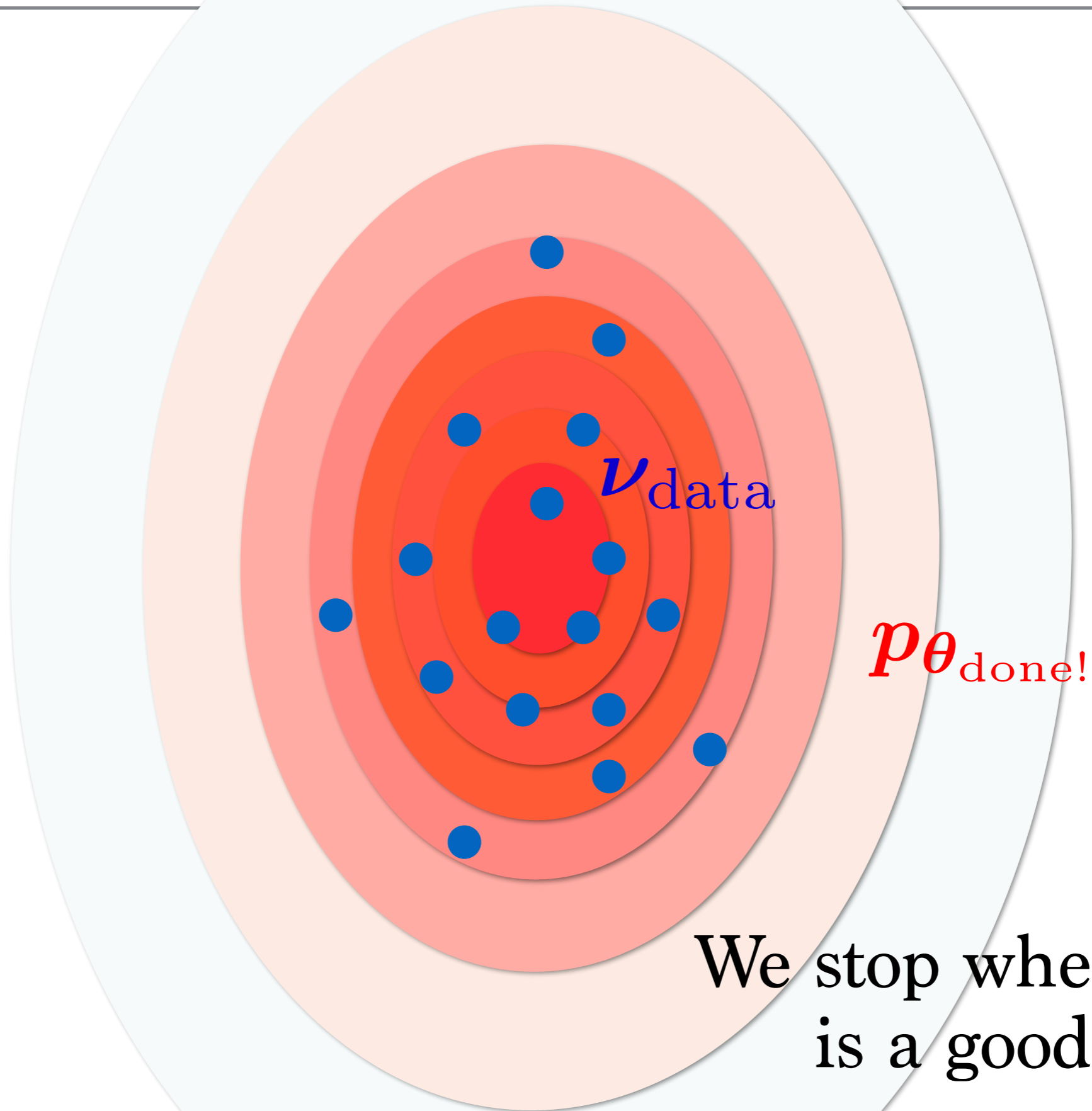
Density Fitting



Density Fitting



Density Fitting



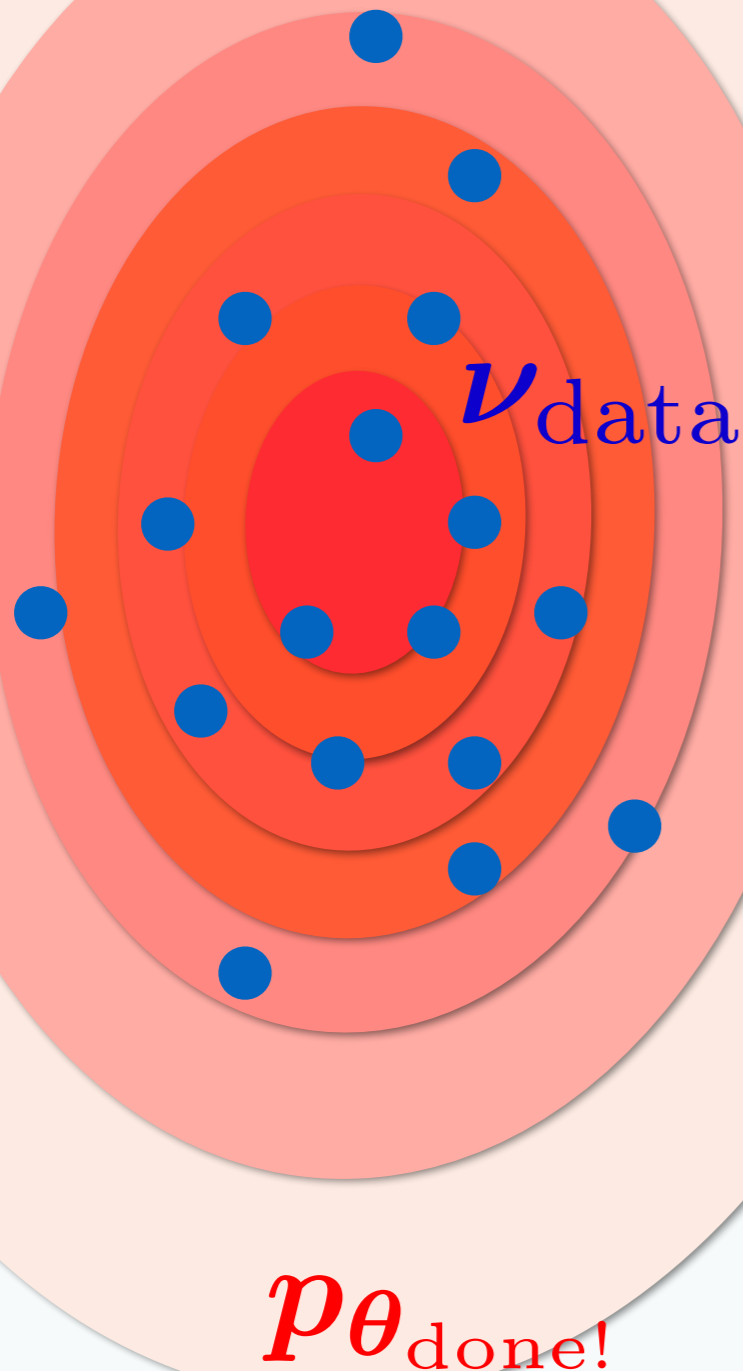
We stop when there is a good fit.

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



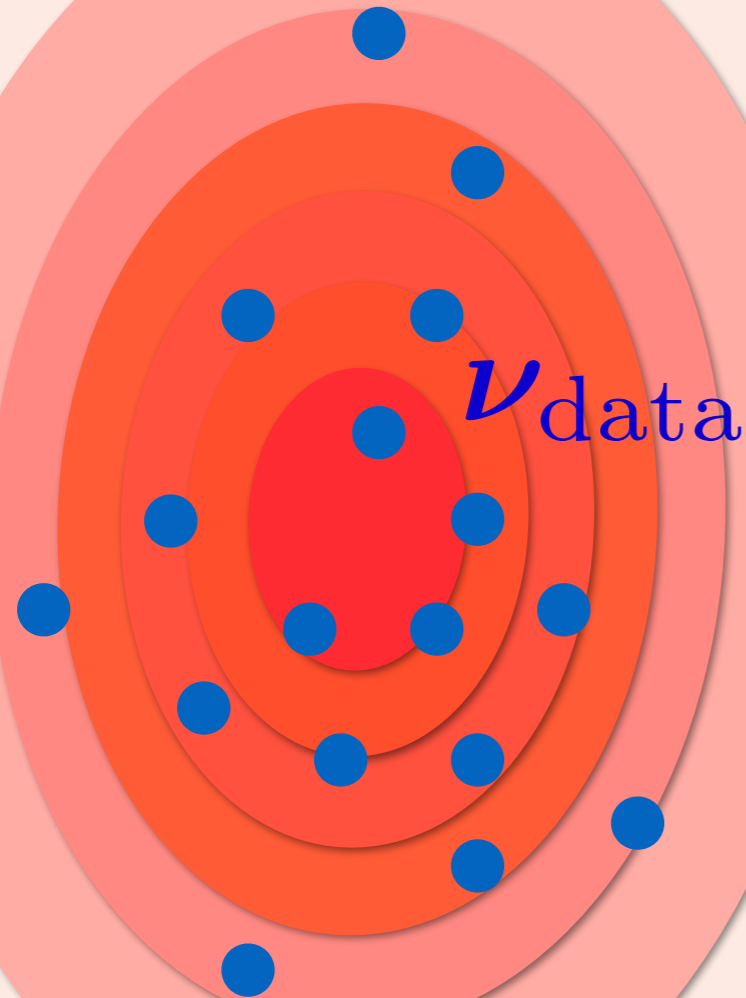
$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i)$$

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



p_{θ} done!

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

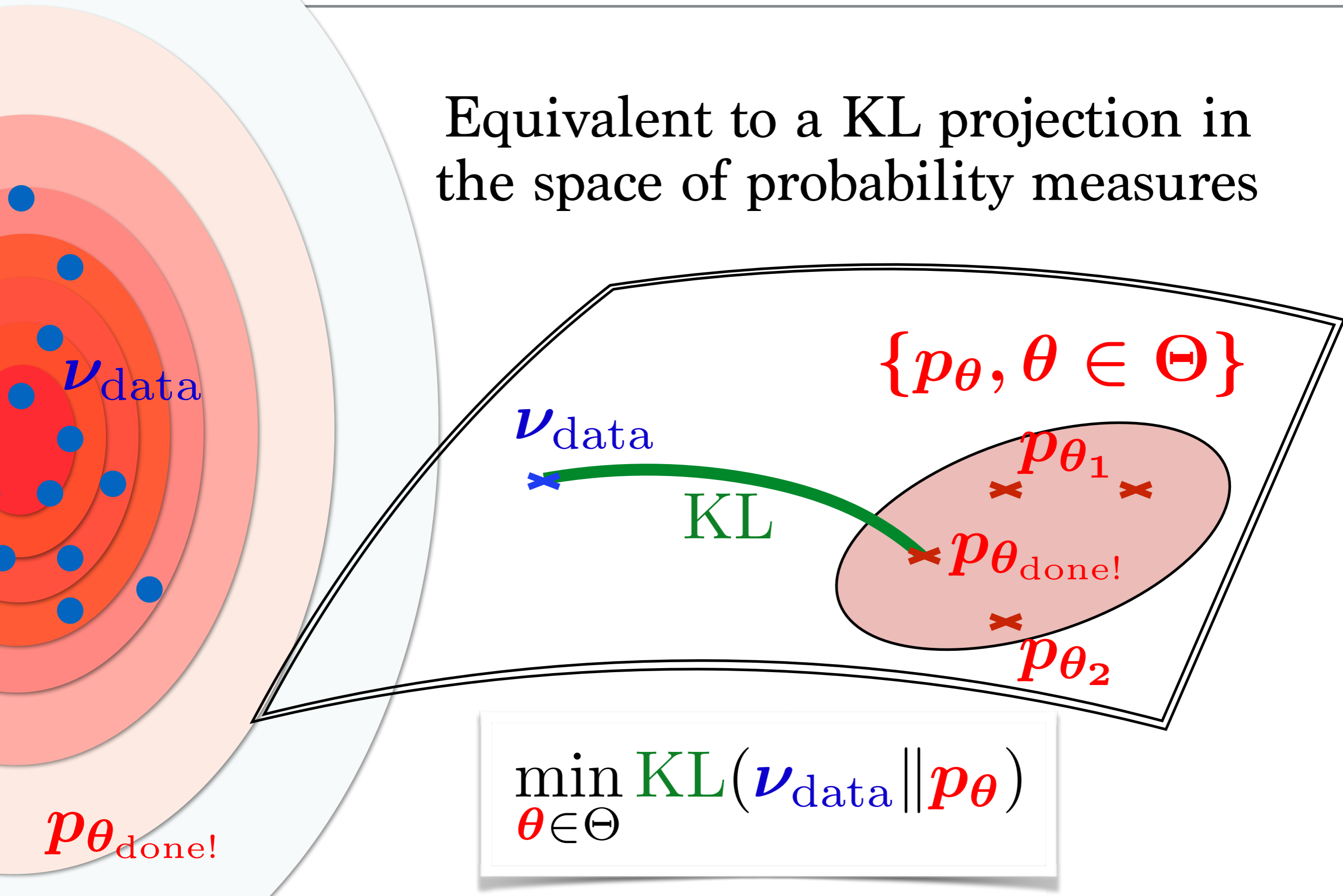


$\log 0 = -\infty$

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

Maximum Likelihood Estimation

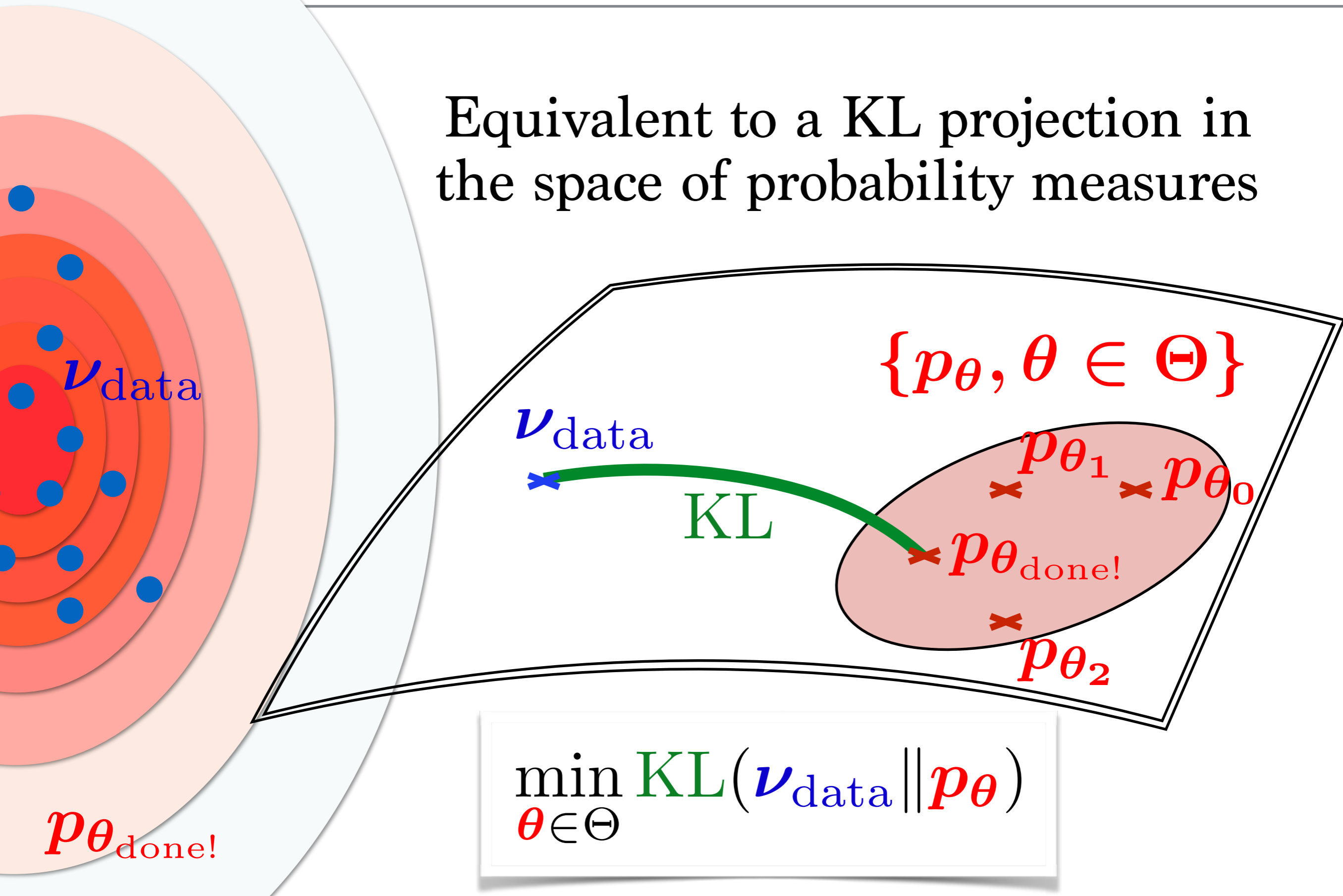
Equivalent to a KL projection in the space of probability measures



$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

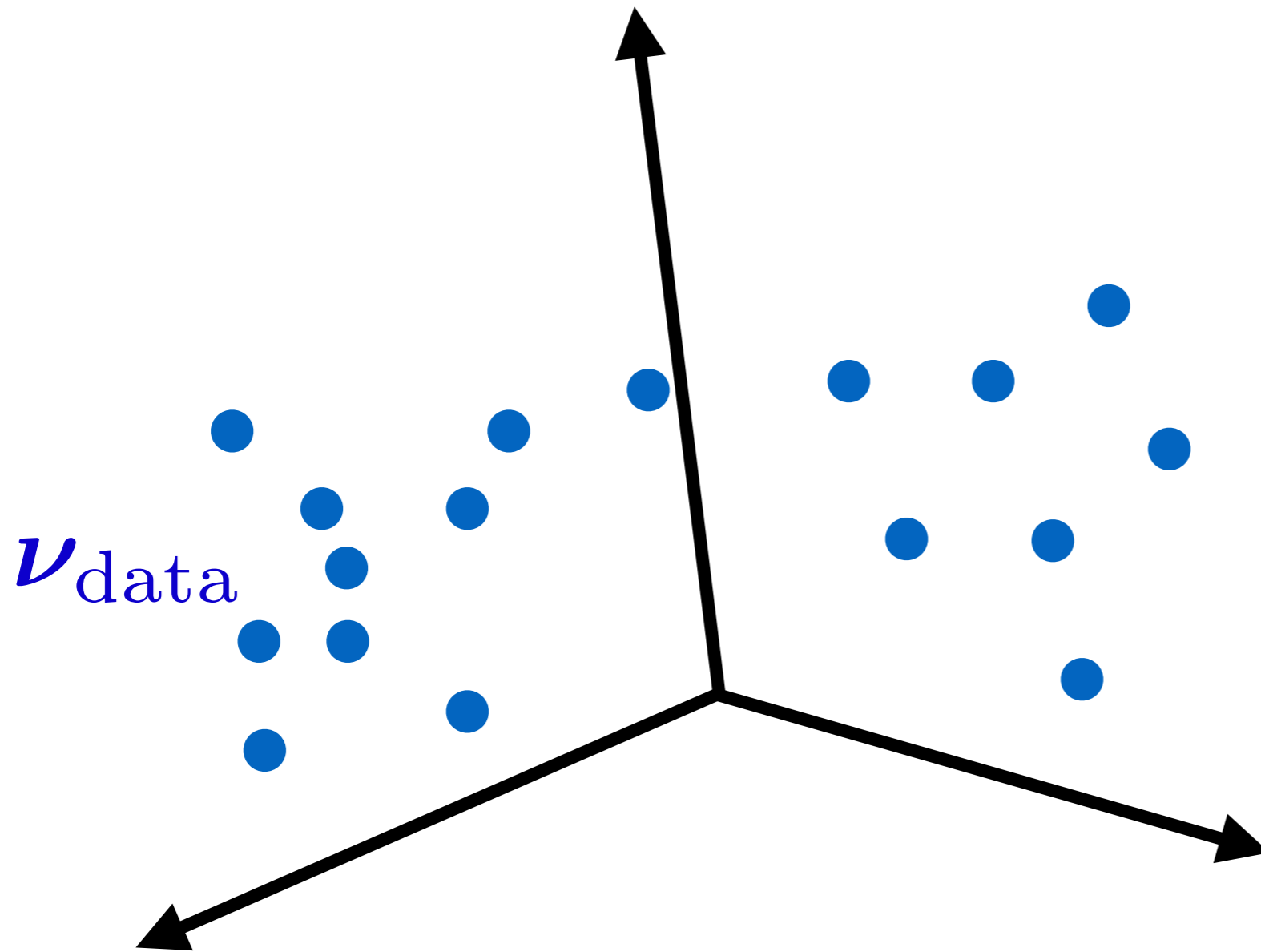
Maximum Likelihood Estimation

Equivalent to a KL projection in the space of probability measures



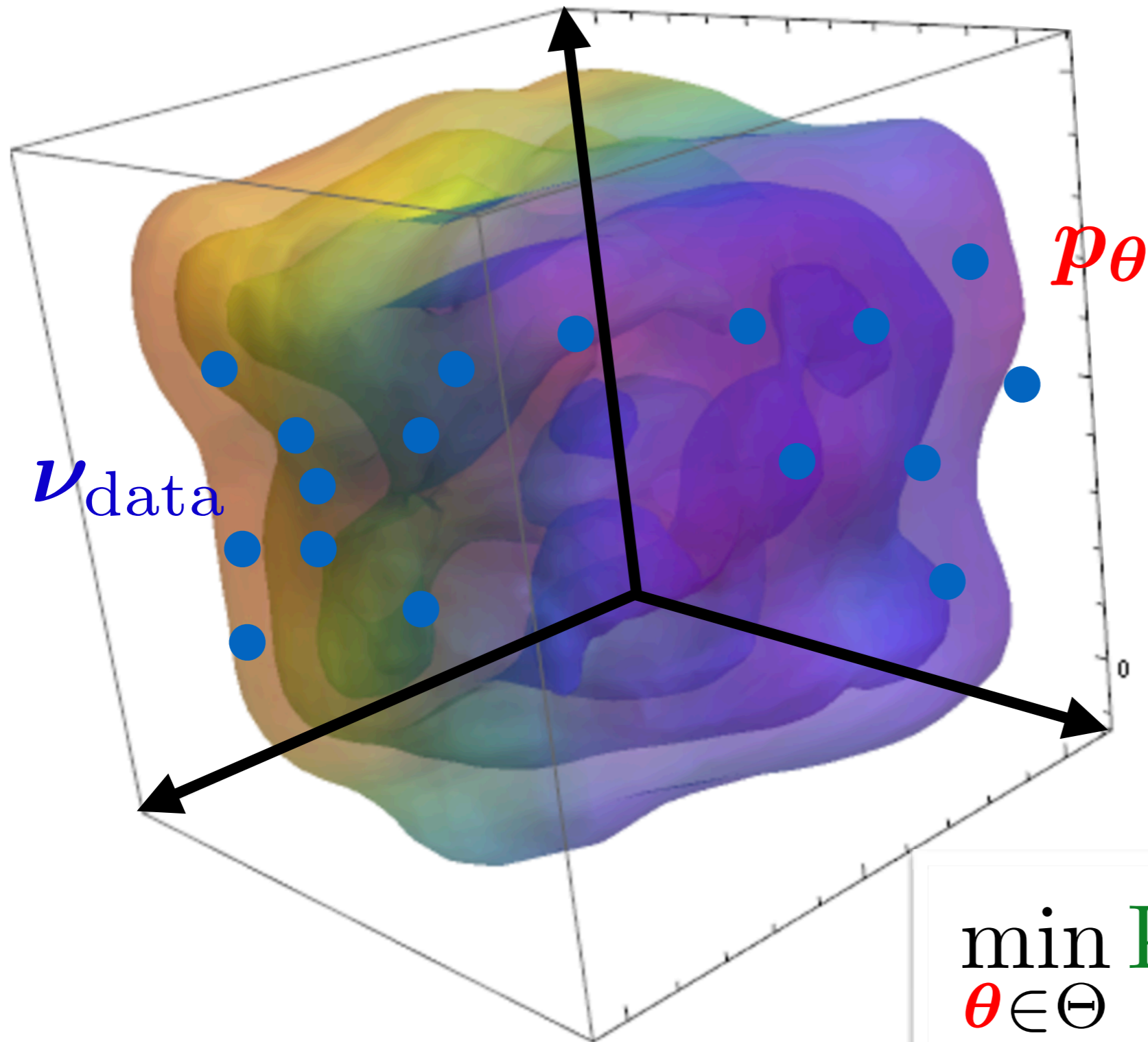
$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

In higher dimensional spaces...



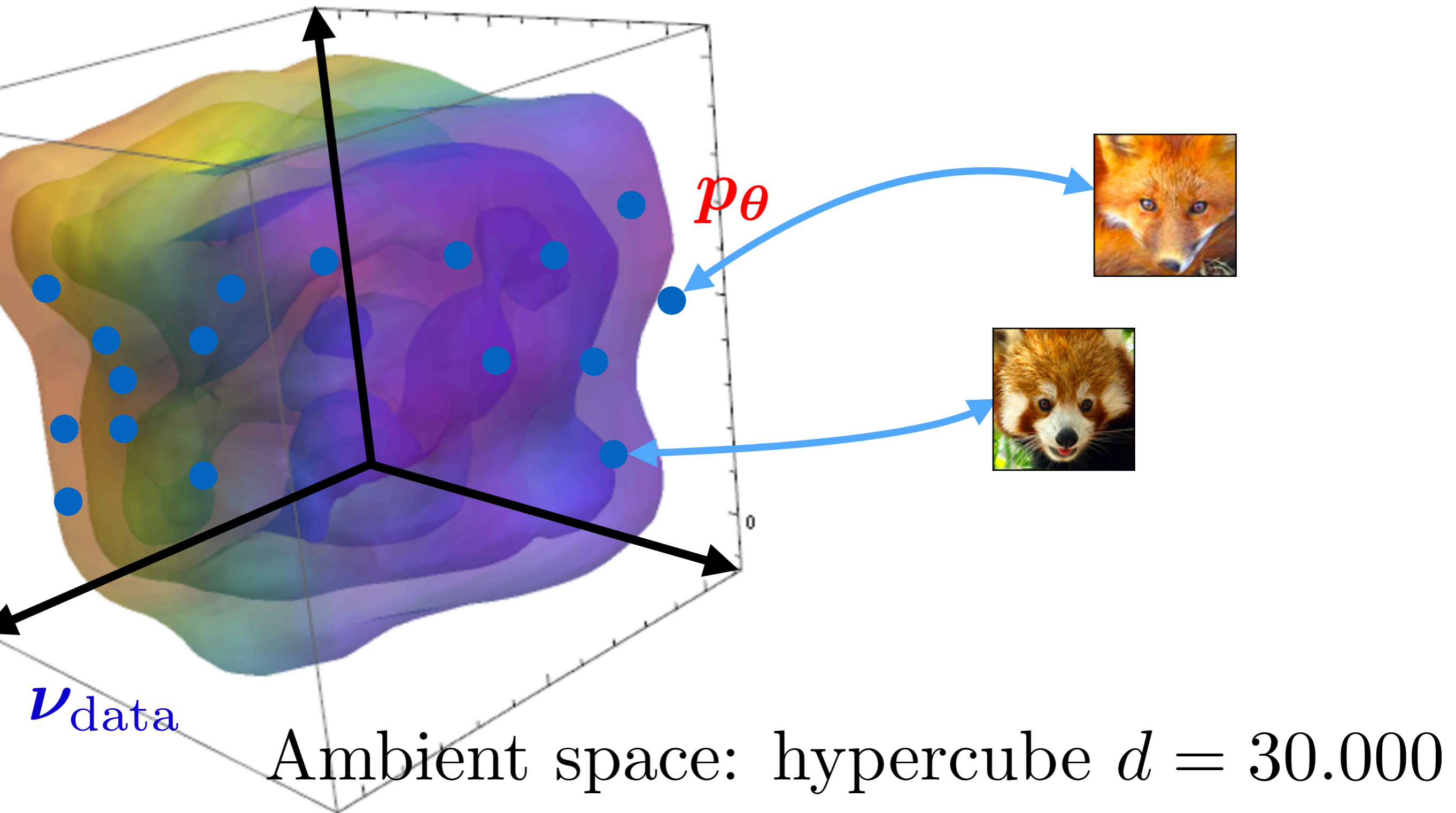
$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

In higher dimensional spaces...

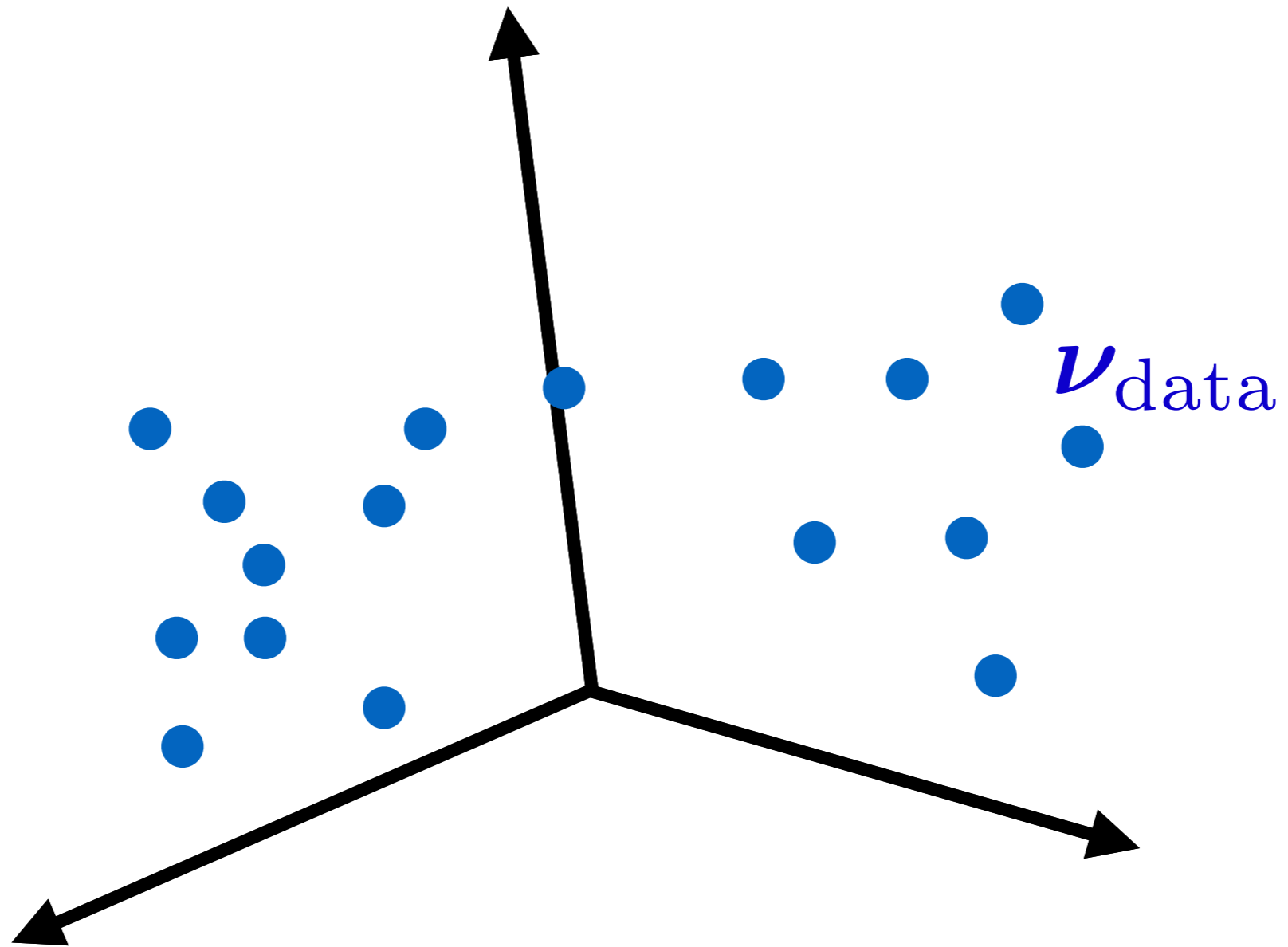


$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$

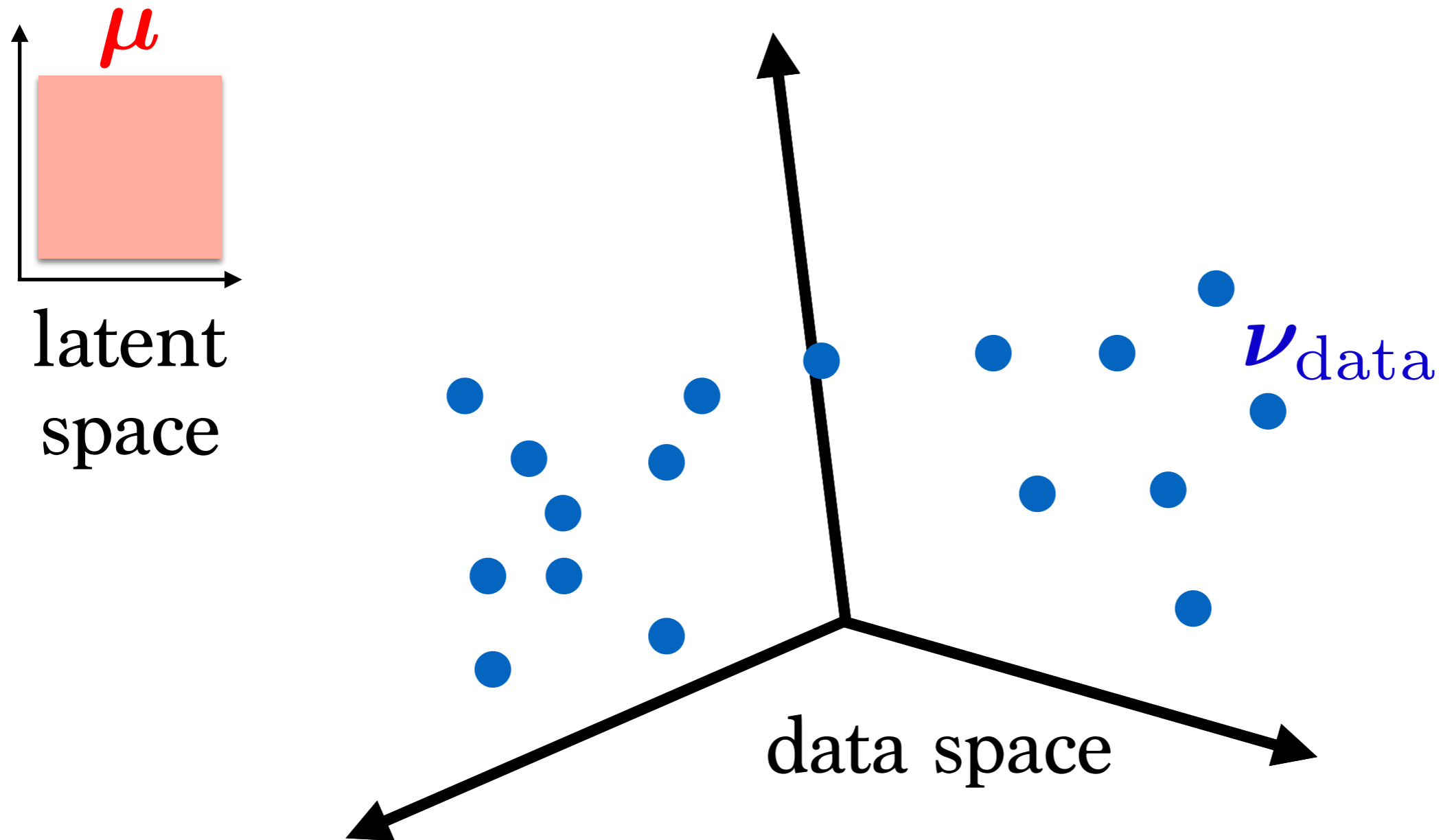
In higher dimensional spaces...



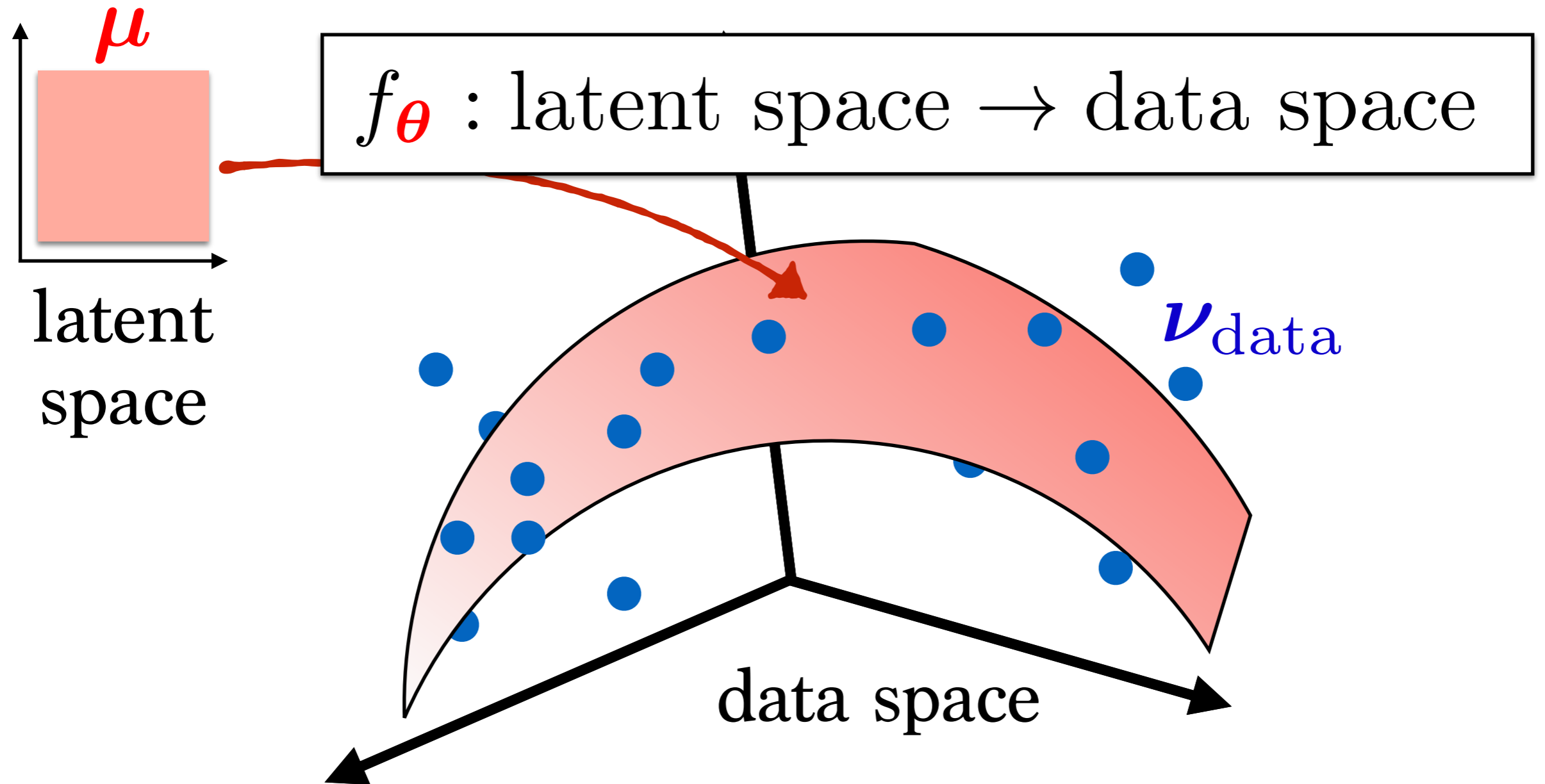
Generative Models



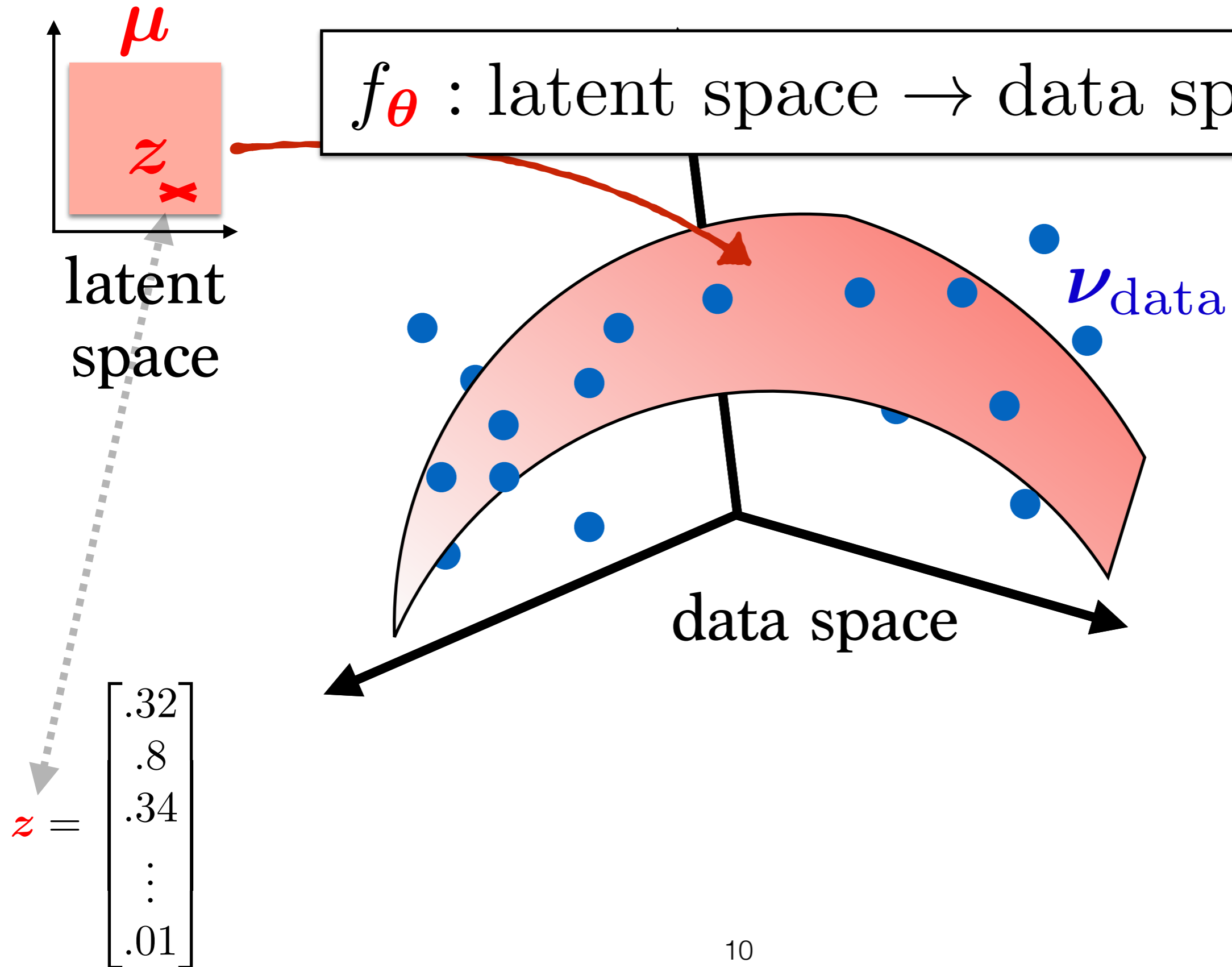
Generative Models



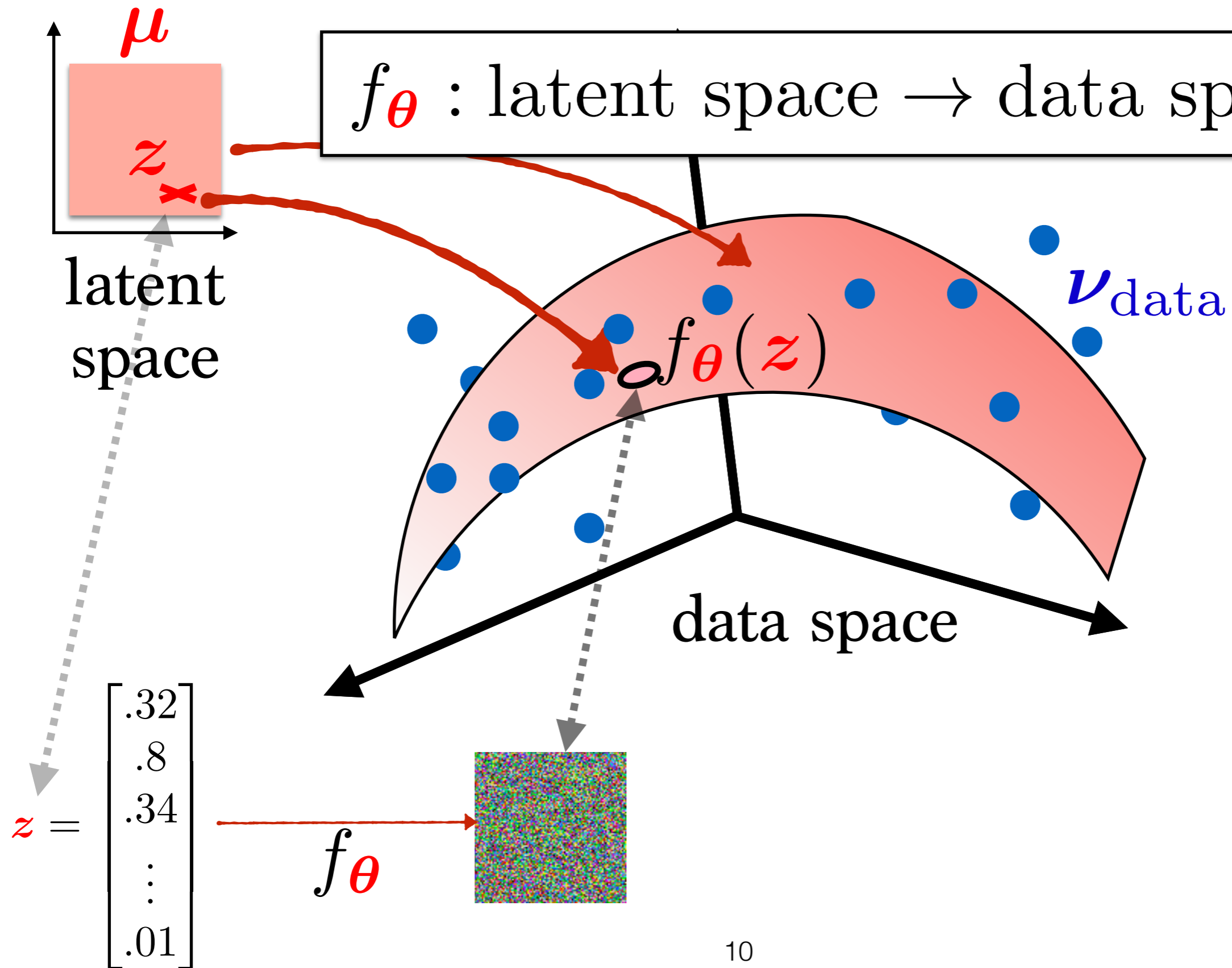
Generative Models



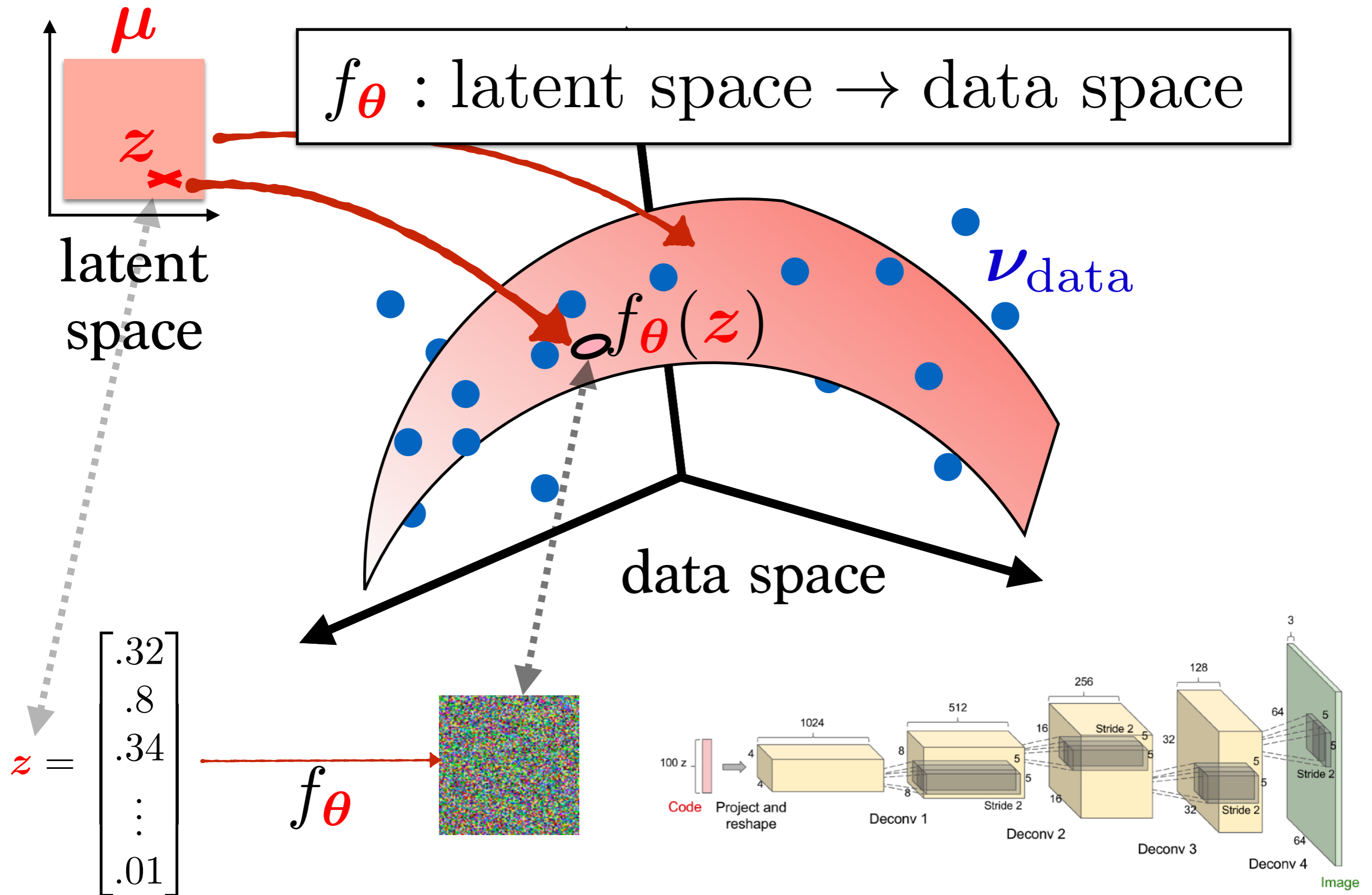
Generative Models



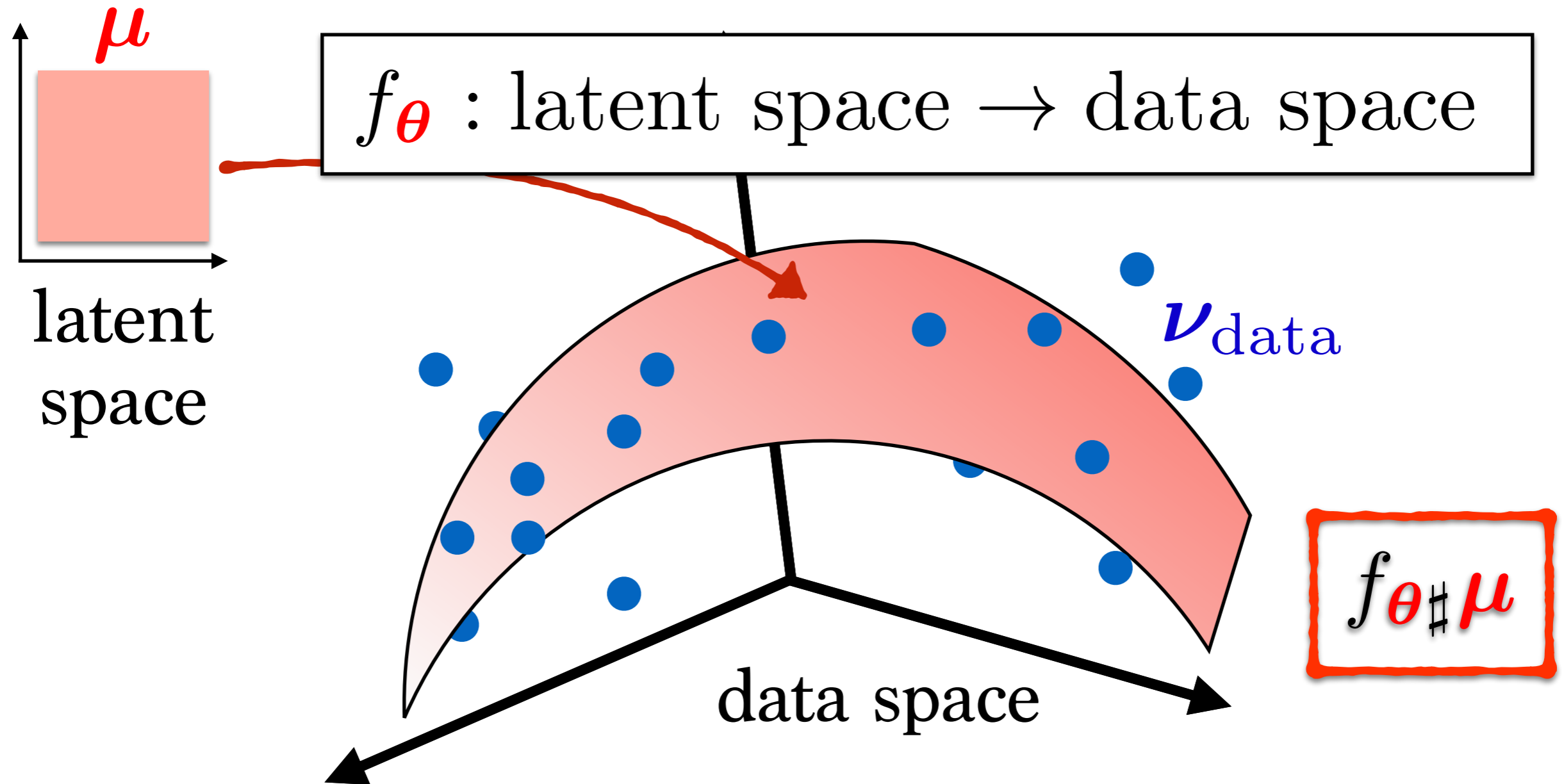
Generative Models



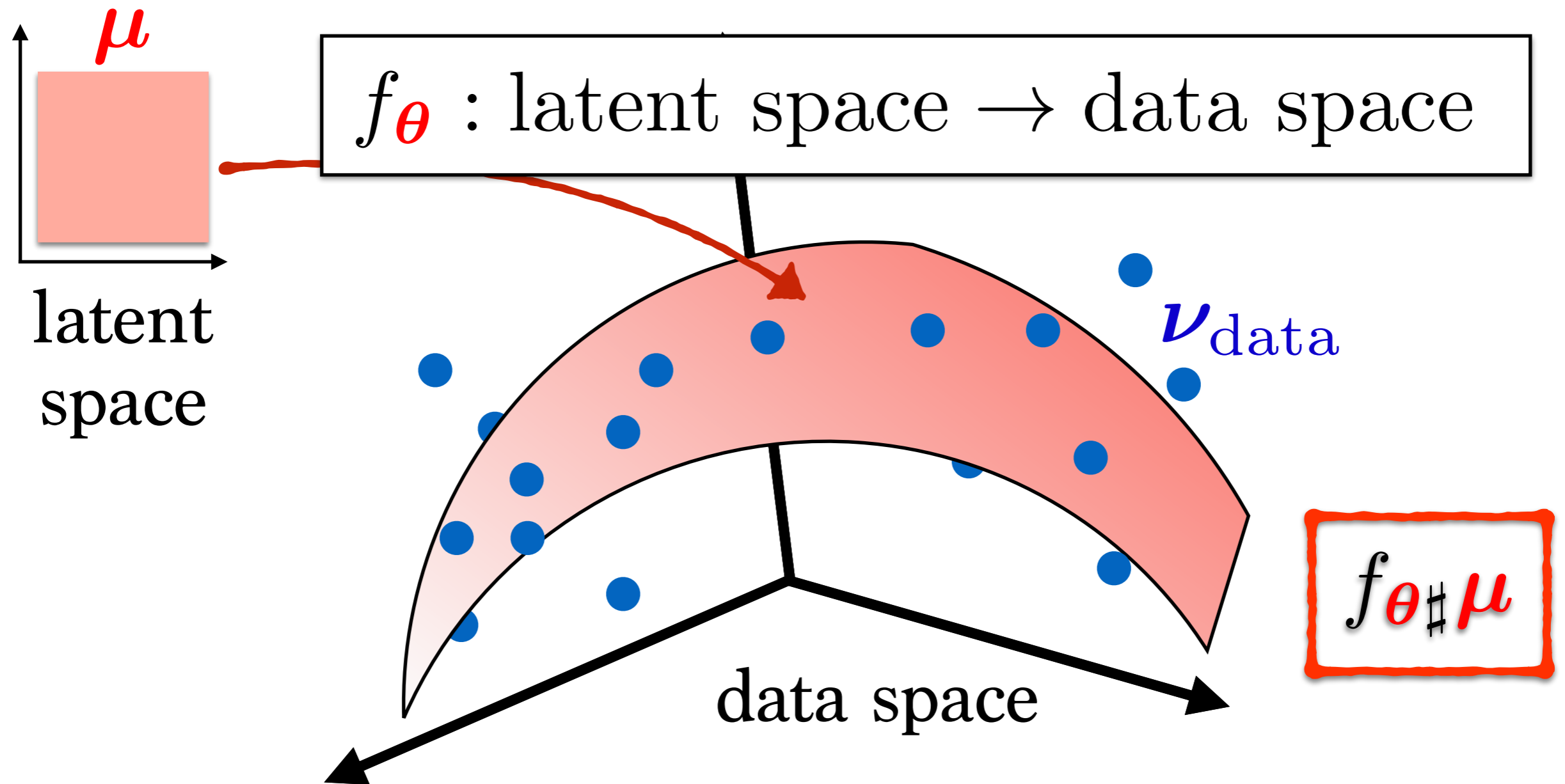
Generative Models



Generative Models

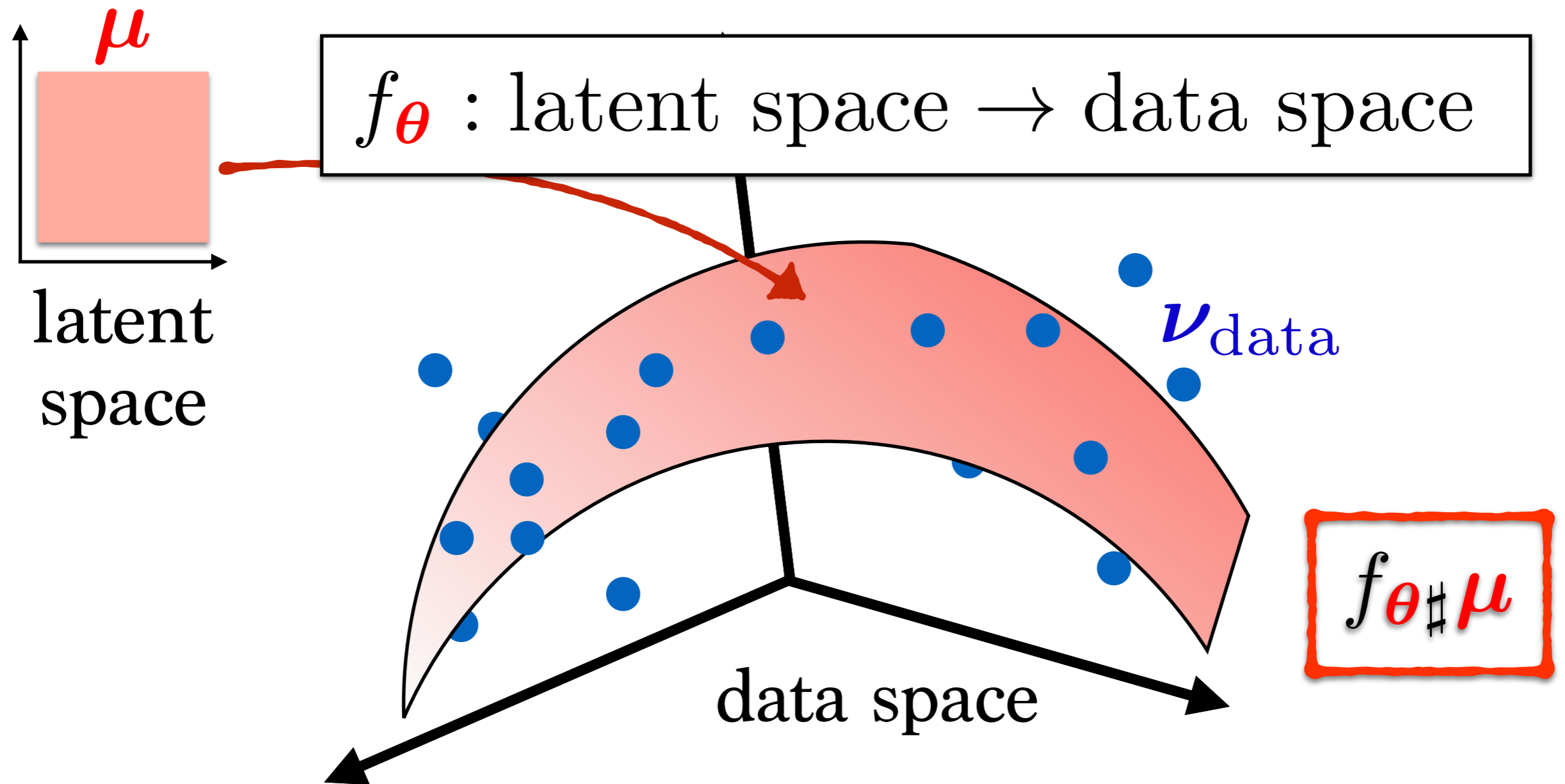


Generative Models



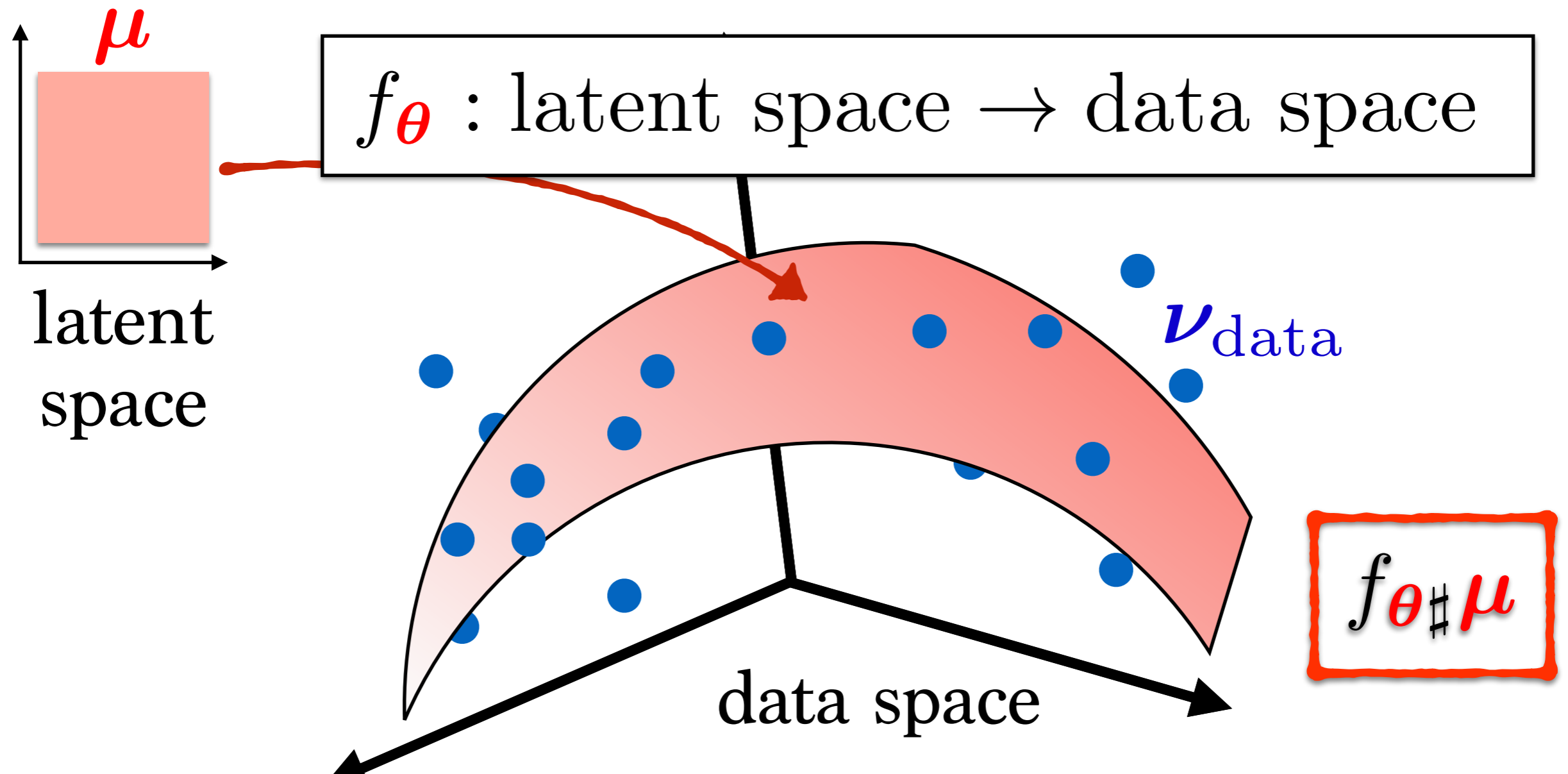
Goal: find θ such that $f_{\theta \# \mu}$ fits $\mathcal{V}_{\text{data}}$

Generative Models



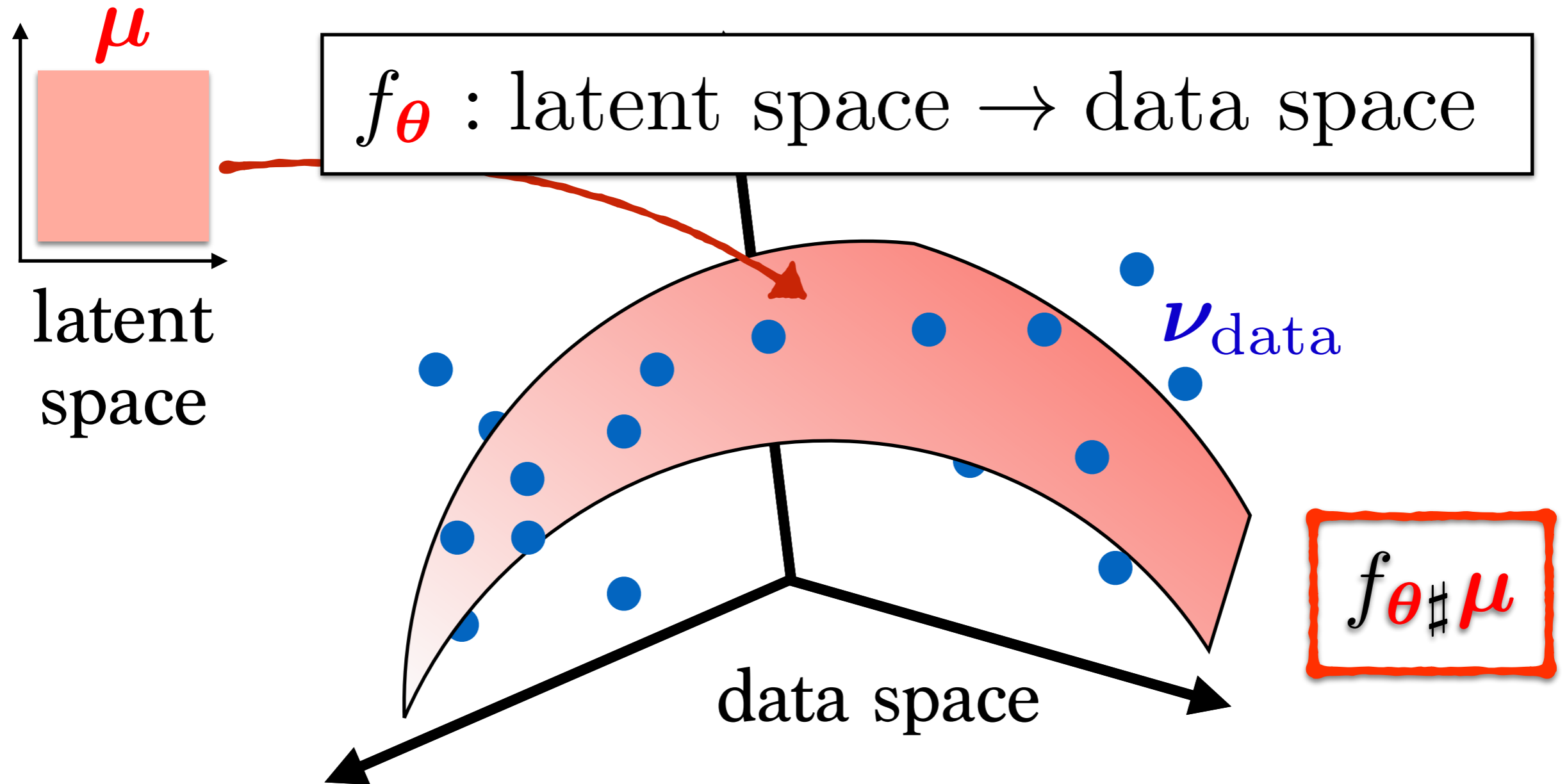
Goal: find θ such that $f_{\theta \# \mu}$ fits $\mathcal{V}_{\text{data}}$

Generative Models



Difference between fitting
 $f_{\theta \# \mu}$ vs. a density p_{θ} ?

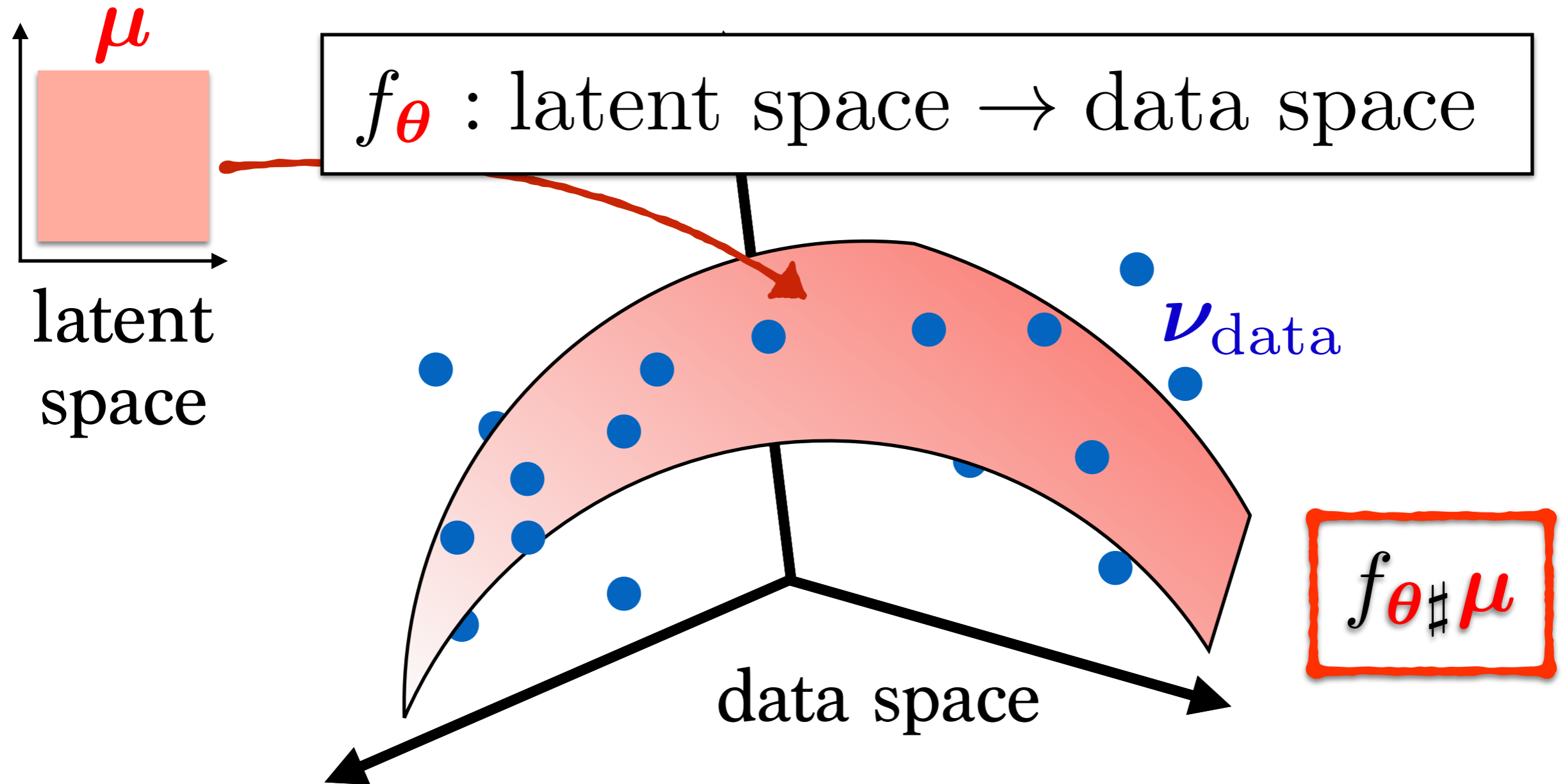
Generative Models



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

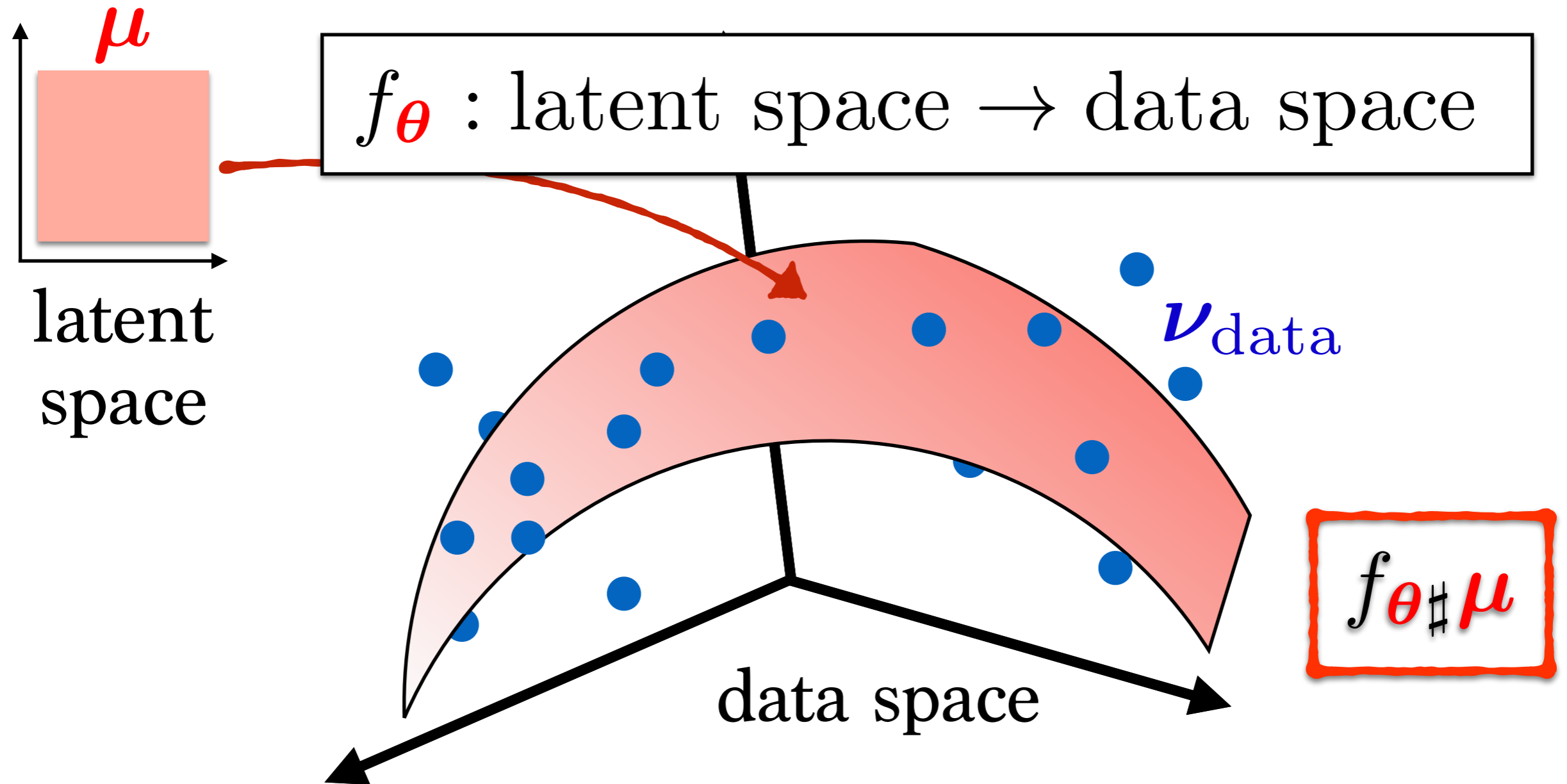


MLE

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

$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \| f_{\theta \# \mu})$$

Generative Models

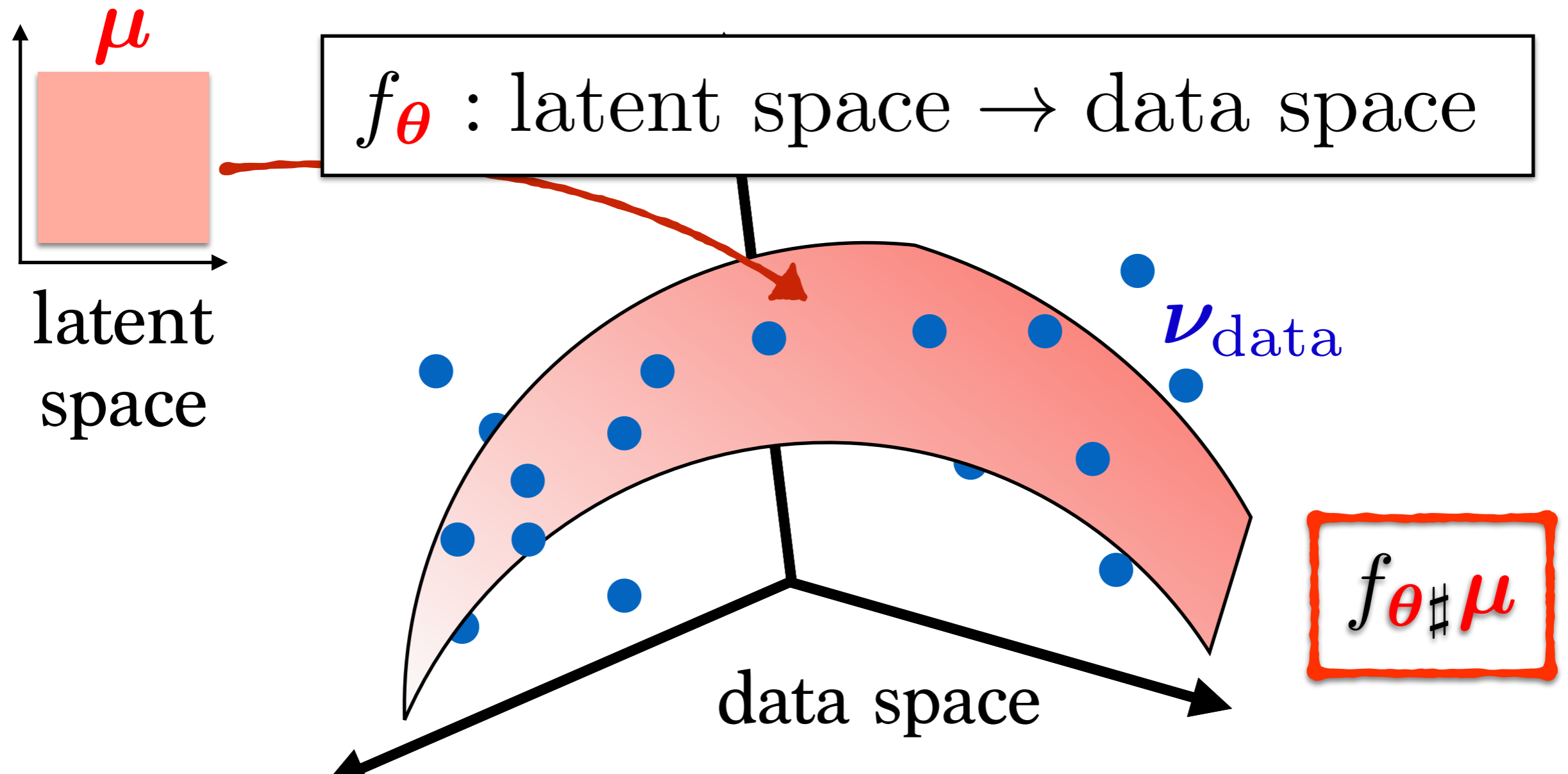


~~MLE~~

$$\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \log f_{\theta \# \mu}(x_i) \quad \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel f_{\theta \# \mu})$$

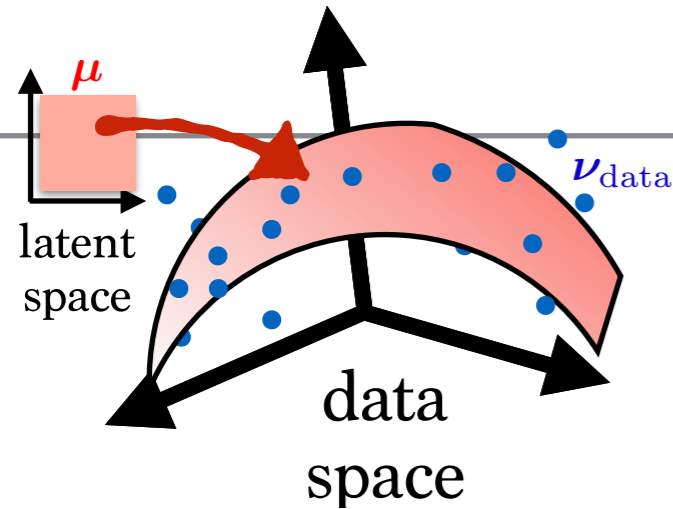


Generative Models



Need a more flexible **discrepancy function** to compare $\mathcal{V}_{\text{data}}$ and $f_{\theta \# \mu}$

Workarounds?



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

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

- 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 χ^2 CHI-SQUARE, NOT MAXIMUM LIKELIHOOD!

BY JOSEPH BERKSON

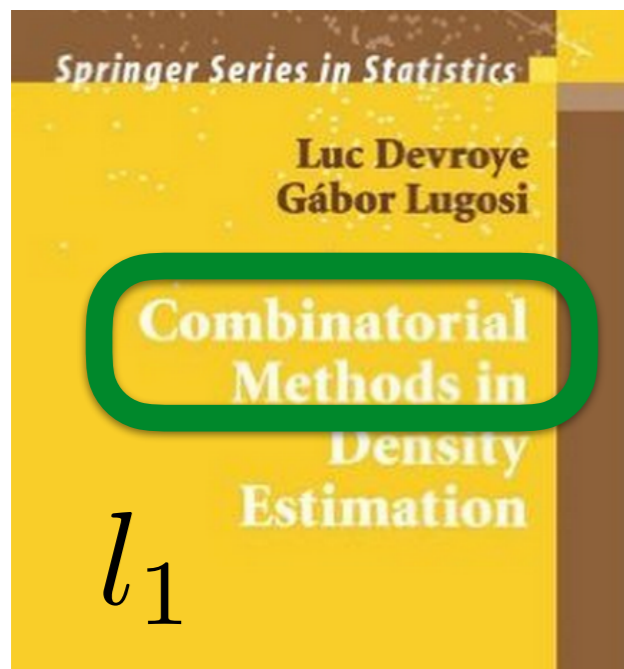
Mayo Clinic, Rochester, Minnesota



ELSEVIER

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®



ELSEVIER

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 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

**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
{chunli1,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

△ 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
{chunli1,wchang2,yiming,bapoczos}@cs.cmu.edu chengyu@us.ibm.com

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

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

Marco Cuturi
CREST, ENSAE, Université Paris-Saclay
marco.cuturi@ensae.fr

△ 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

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
{chunli1,wchang2,yiming,bapoczos}@cs.cmu.edu chengyu@us.ibm.com

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

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

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

Marco Cuturi
CREST, ENSAE, Université Paris-Saclay
marco.cuturi@ensae.fr

Improving GANs Using **Optimal Transport**

Minimum Kantorovich Estimation

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

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

- Optimal transport? fertile field in mathematics.



Monge



Kantorovich



Koopmans



Dantzig



Brenier



Otto



McCann



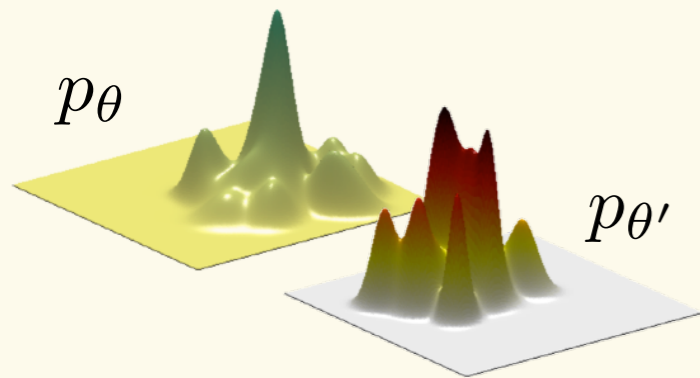
Villani

Nobel '75

Fields '10

What is Optimal Transport?

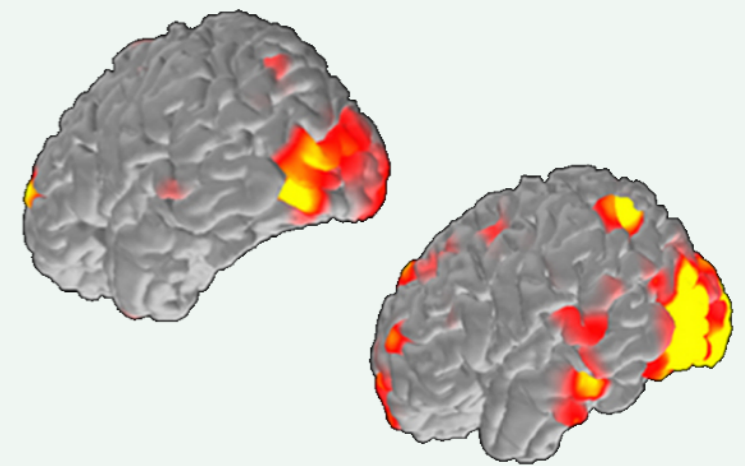
The natural geometry for **probability measures**



Statistical Models

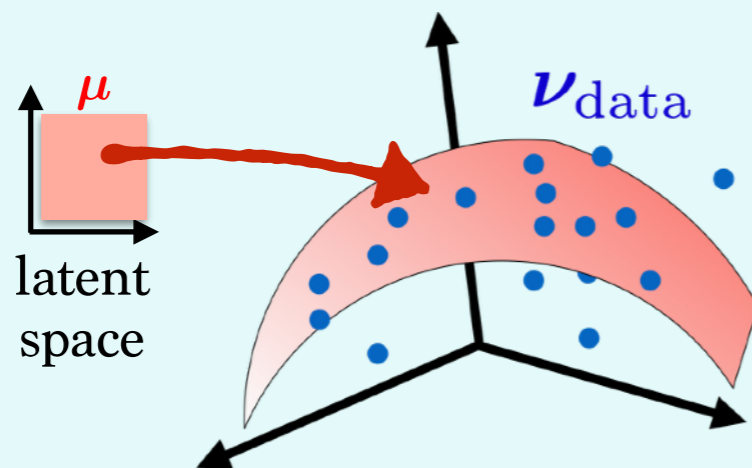


Bags of features



Brain Activation Maps

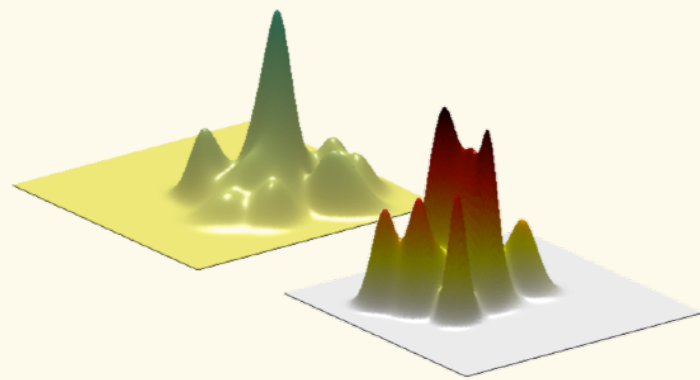
Generative Models vs. data



Color Histograms

What is Optimal Transport?

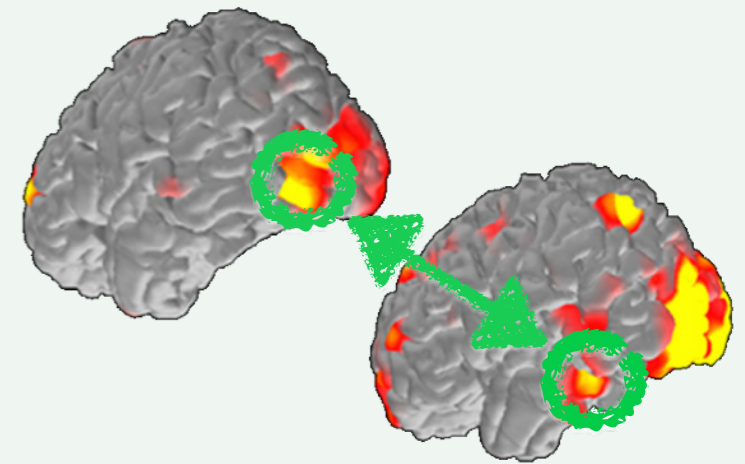
The natural geometry for **probability measures** supported on a geometric space.



Statistical Models

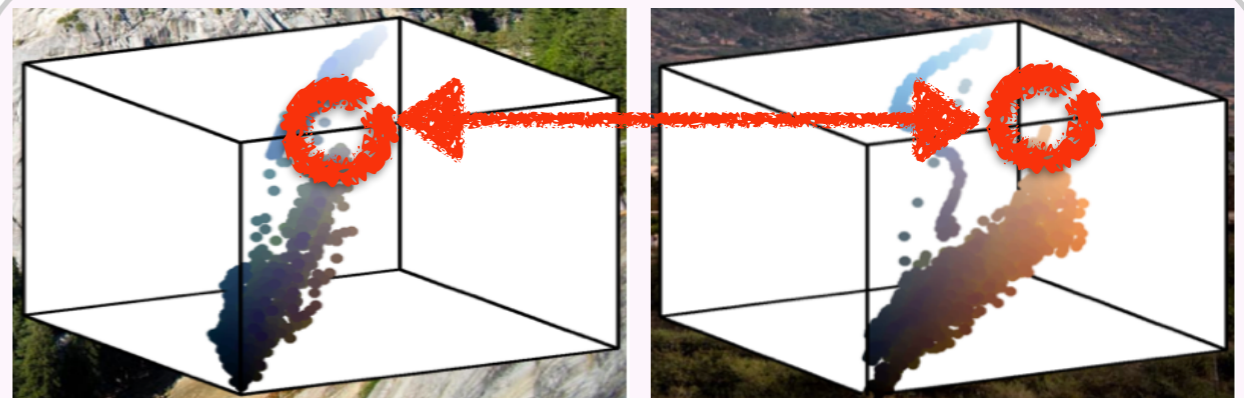
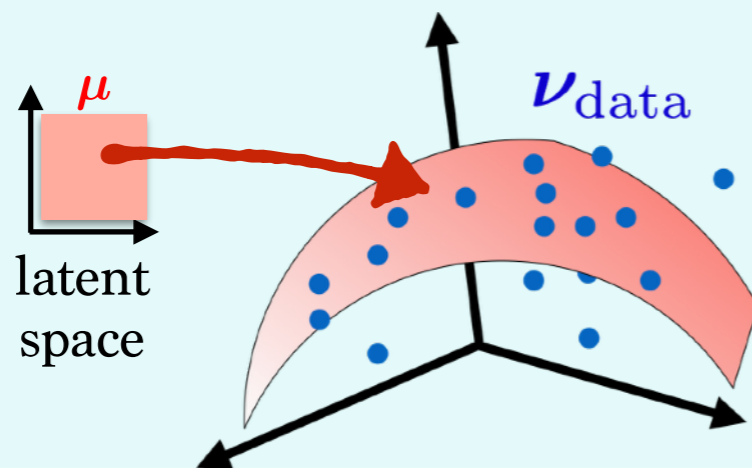


Bags of Features



Brain Activation Maps

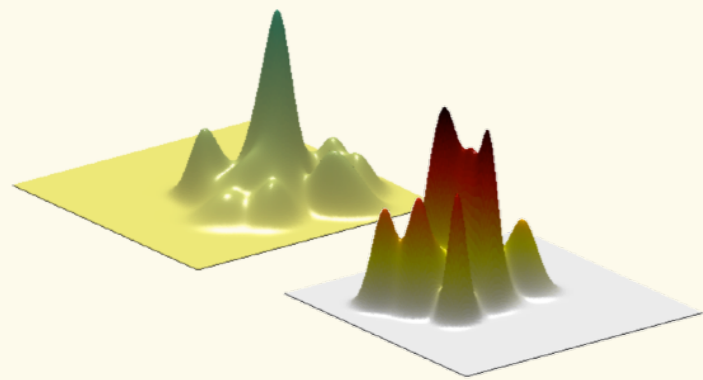
Generative Models vs. Data



Color Histograms

What is Optimal Transport?

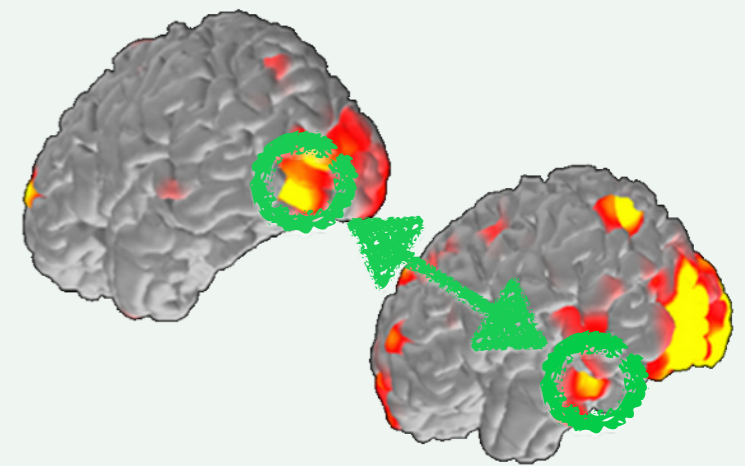
The natural geometry for **probability measures** supported on a geometric space.



Statistical Models

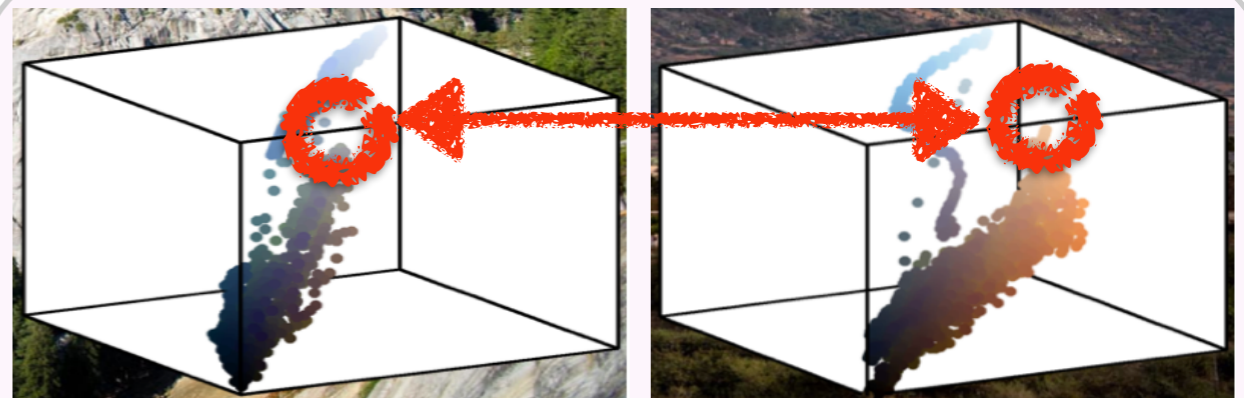
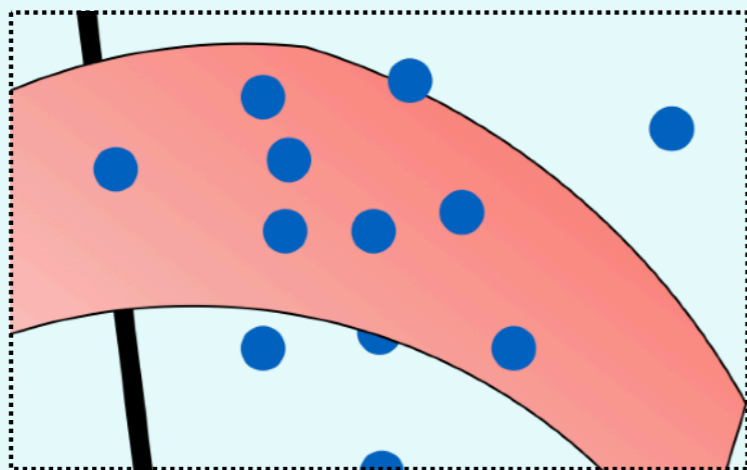


Bags of Features



Brain Activation Maps

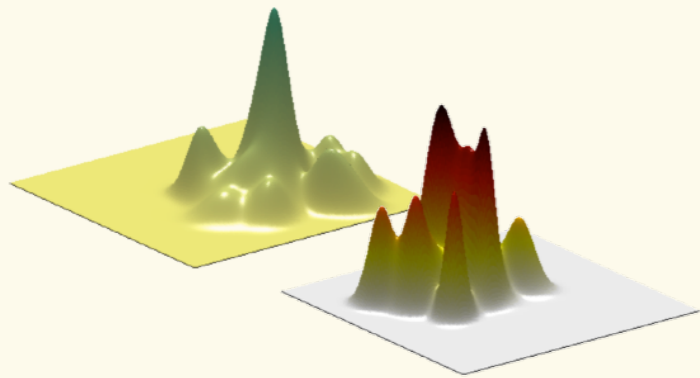
Generative Models vs. Data



Color Histograms

What is Optimal Transport?

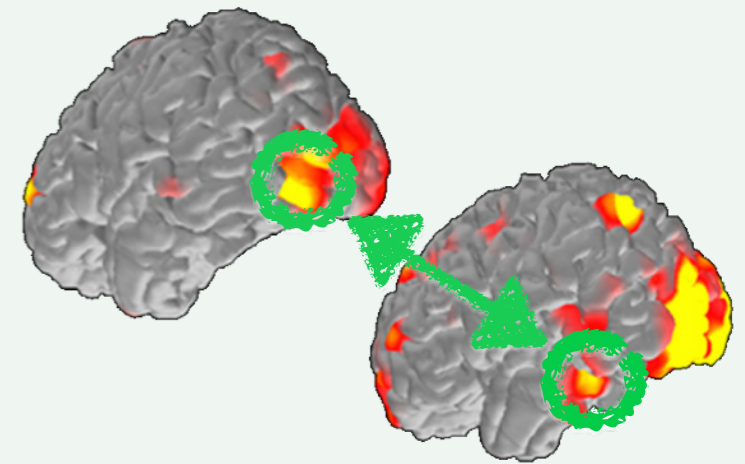
The natural geometry for **probability measures** supported on a geometric space.



Statistical Models

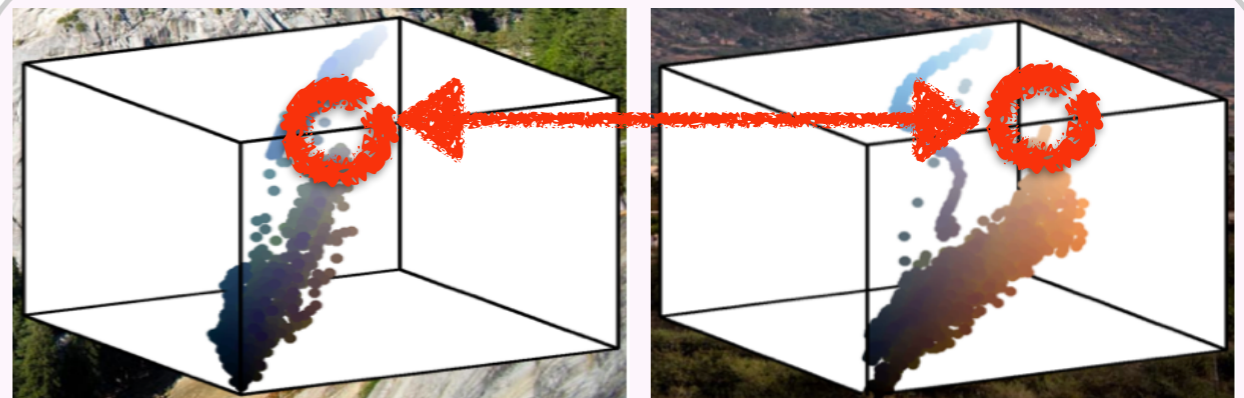
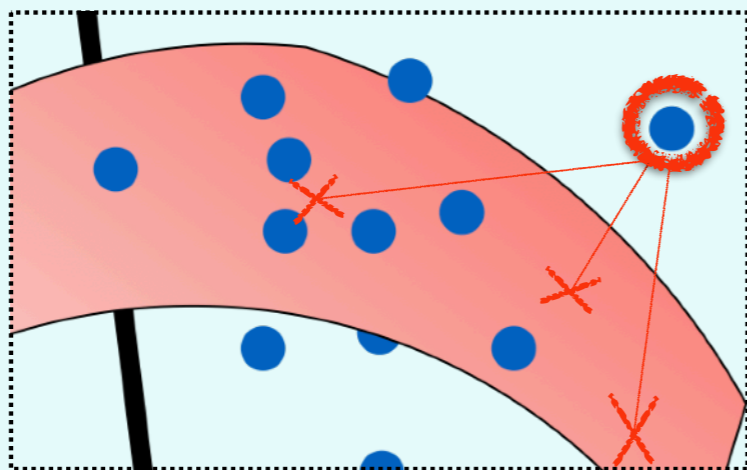


Bags of Features



Brain Activation Maps

Generative Models vs. Data



Color Histograms

Origins: Monge Problem (1781)



60 MÉMOIRES DE L'ACADÉMIE ROYALE

M É M O I R E

S U R L A

T H É O R I E D E S D É B L A I S

E T D E S R E M B L A I S.

Par M. M O N G E.

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'ACADÉMIE ROYALE

M É M O I R E

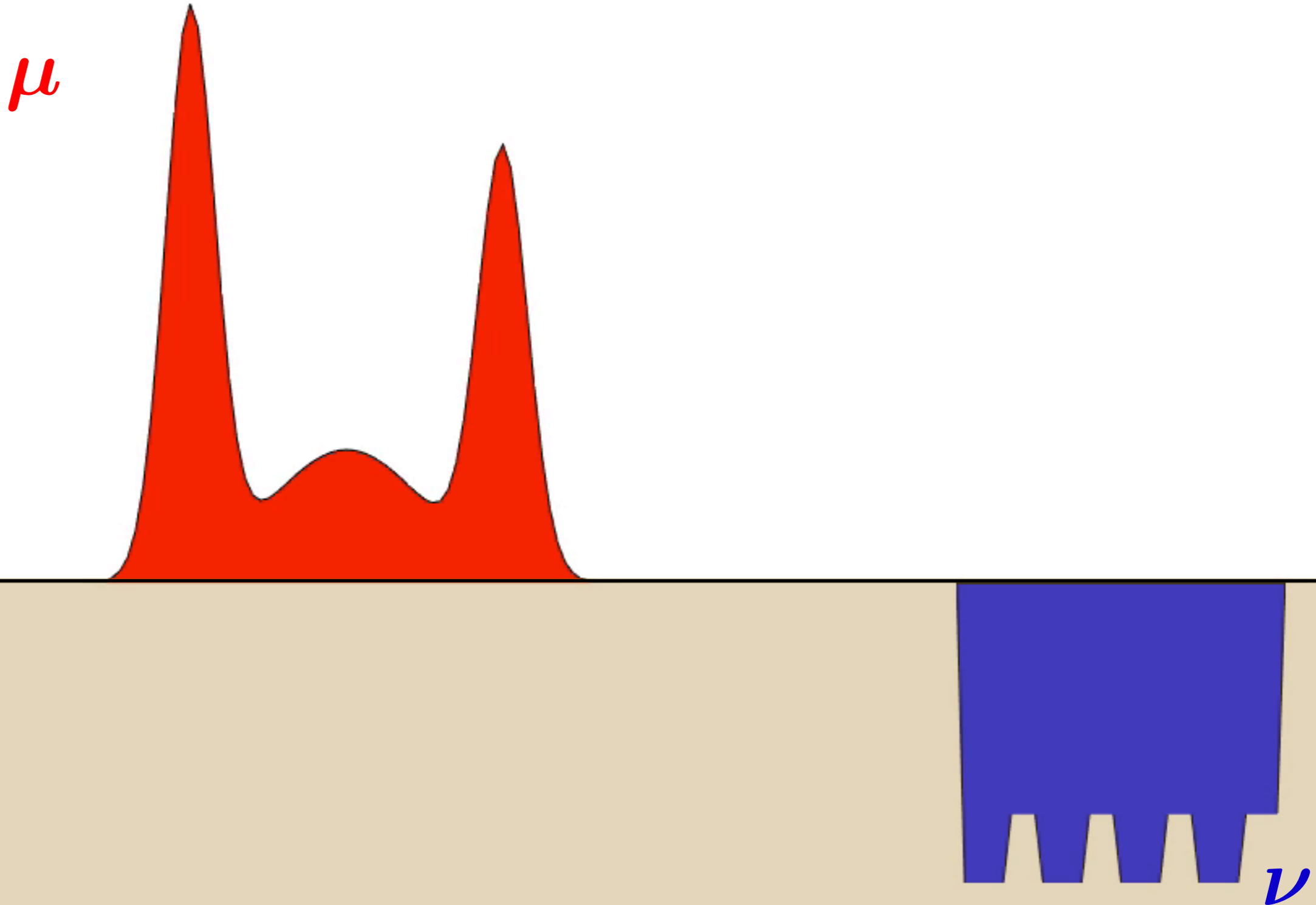
SUR LA

T H É O R I E D E S D É B L A I S

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

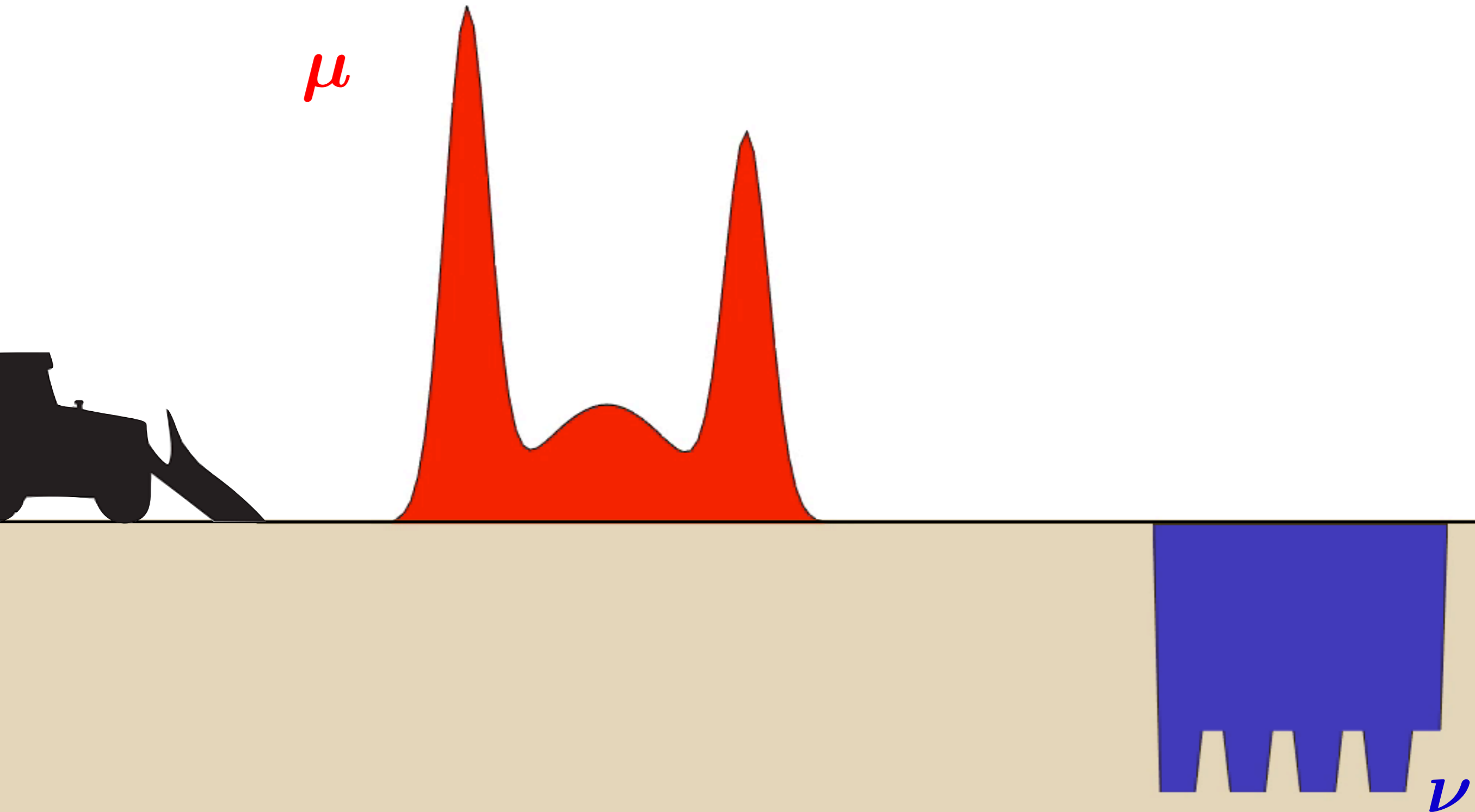
LORSQU'ON doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Origins: Monge Problem



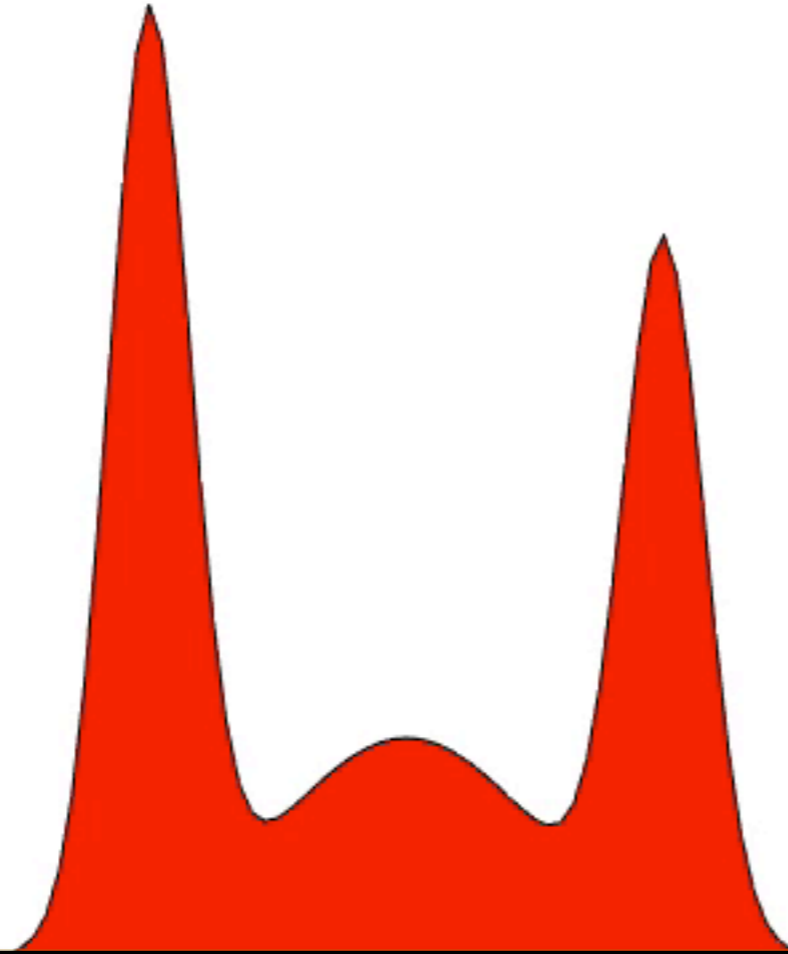
Origins: Monge Problem

In the 21st Century...



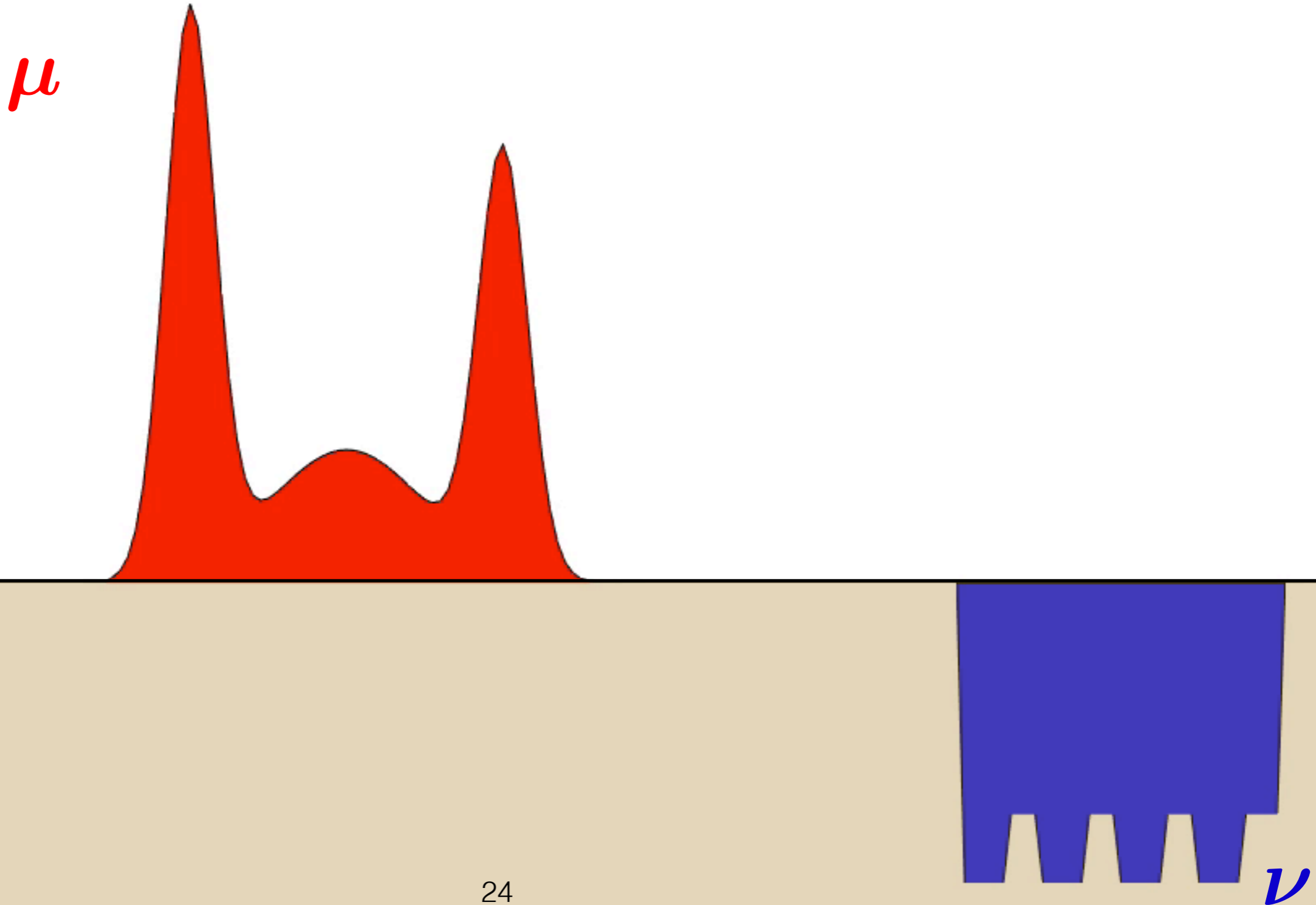
Origins: Monge Problem

In the 21st Century...



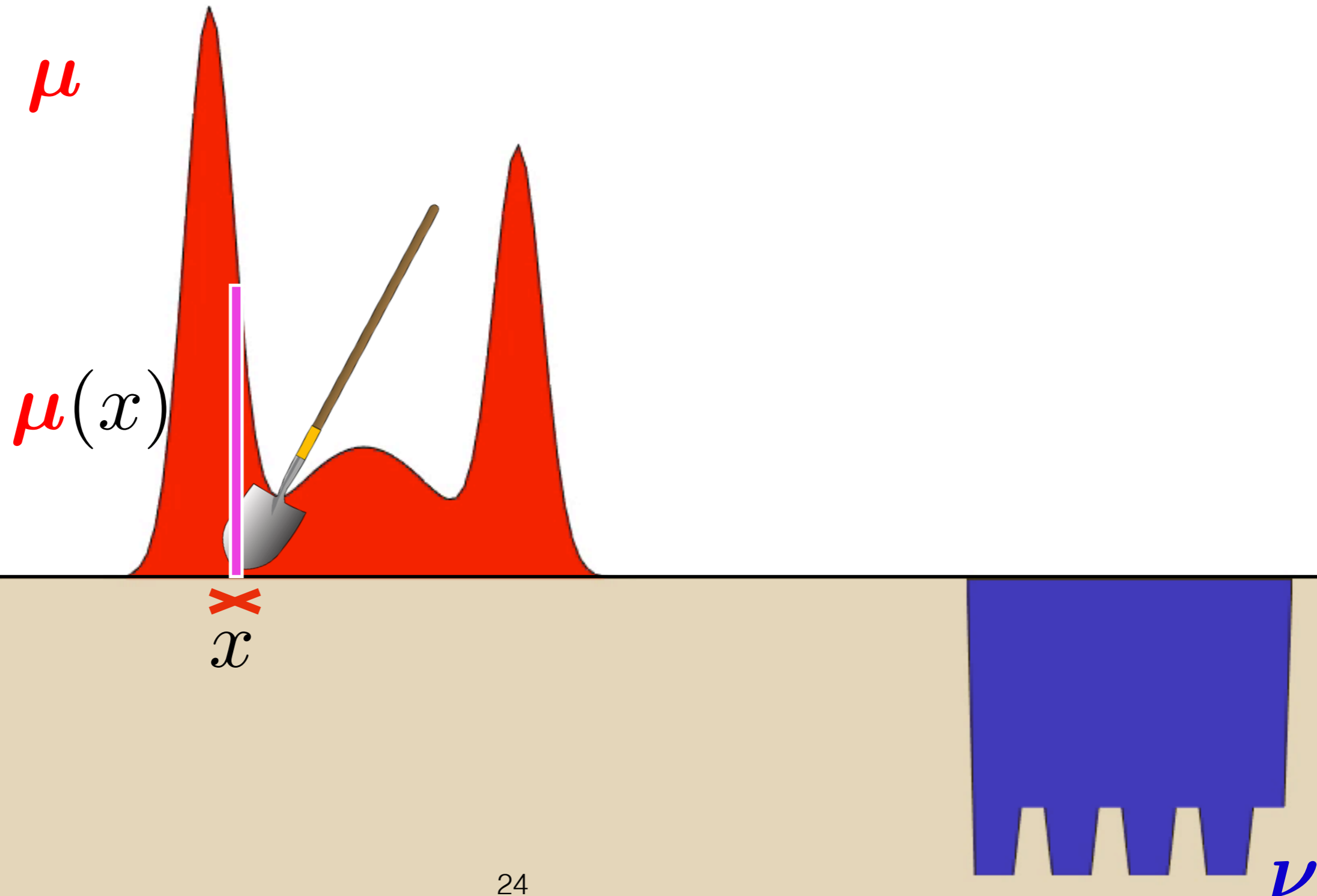
Origins: Monge's Problem

In 1781 however...



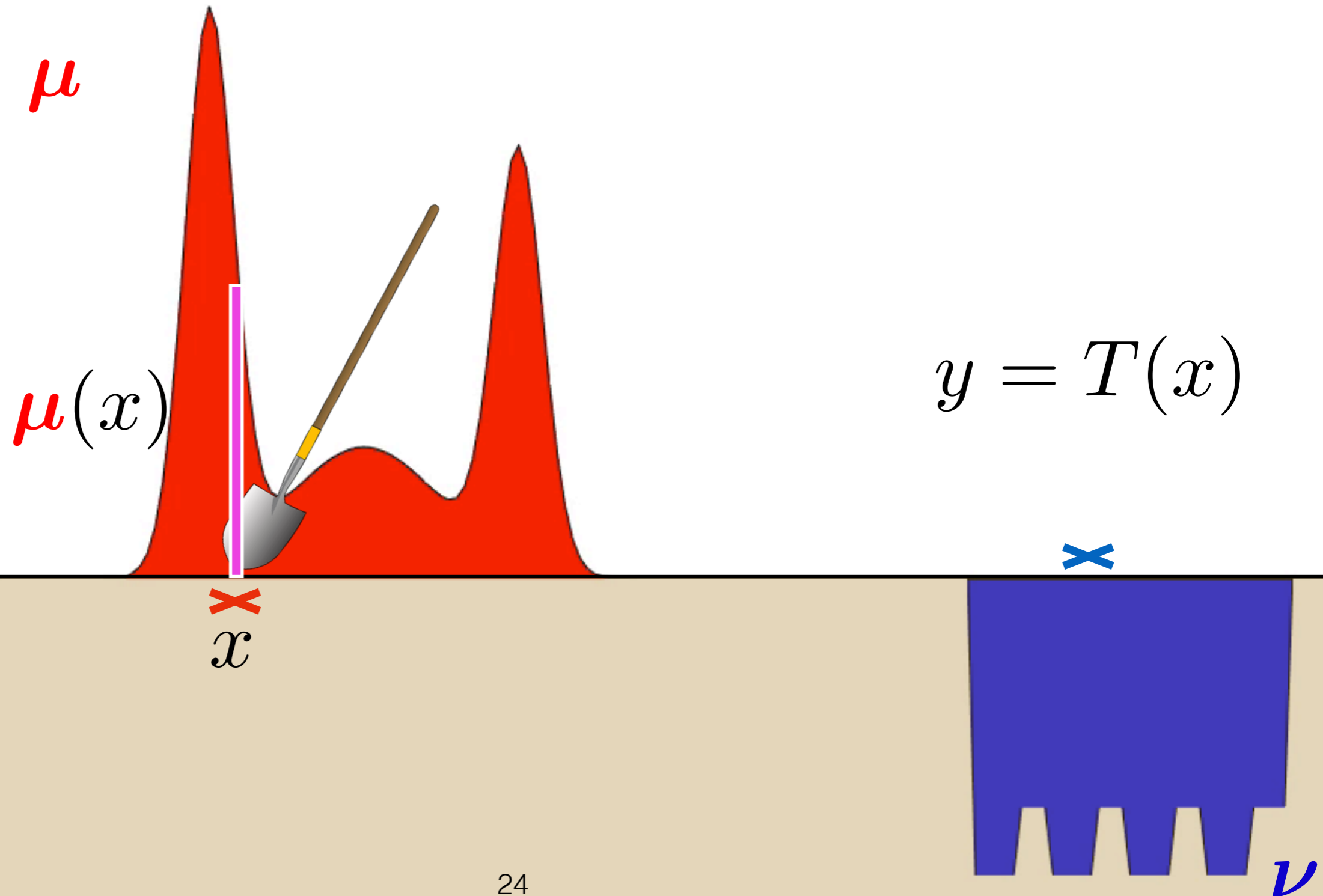
Origins: Monge's Problem

In 1781 however...



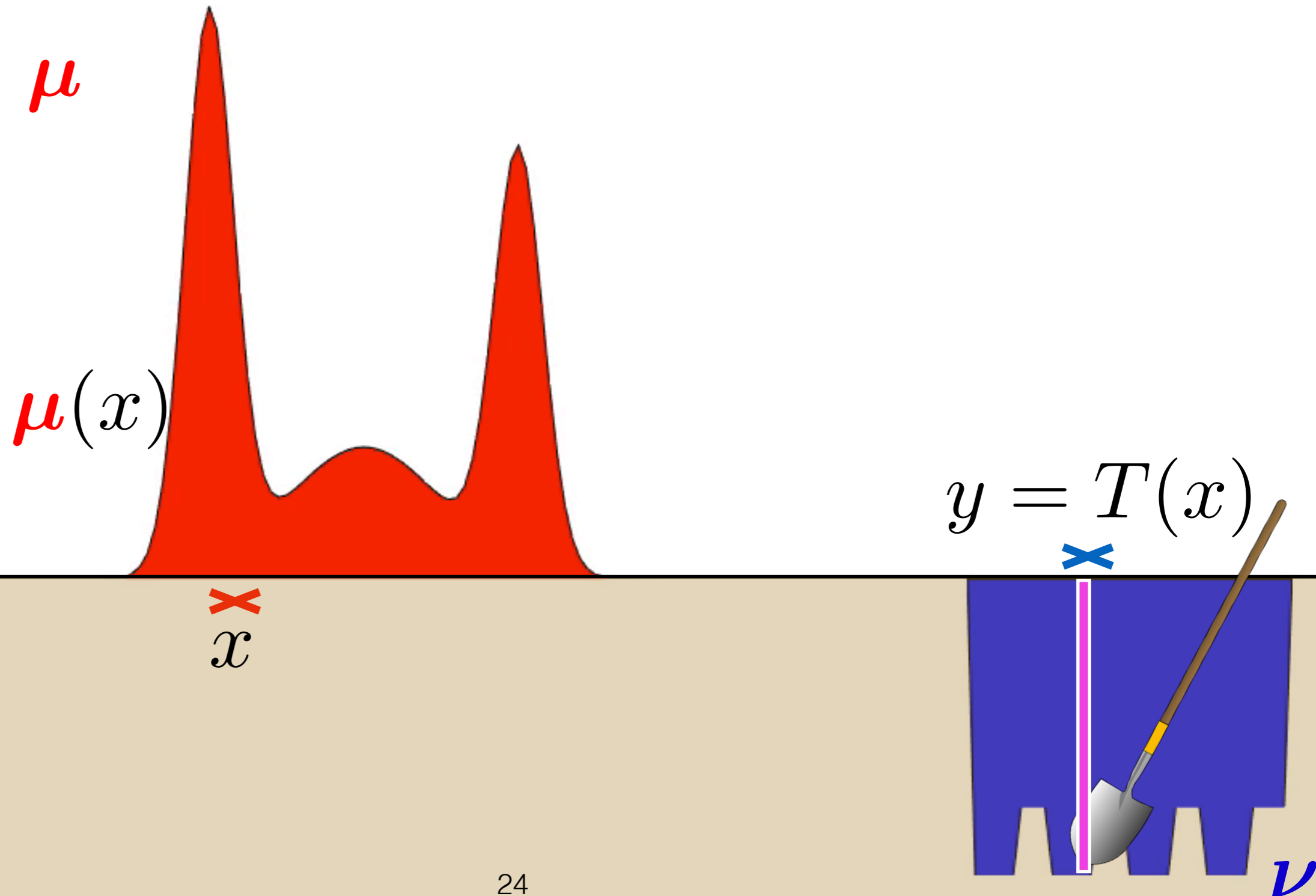
Origins: Monge's Problem

In 1781 however...



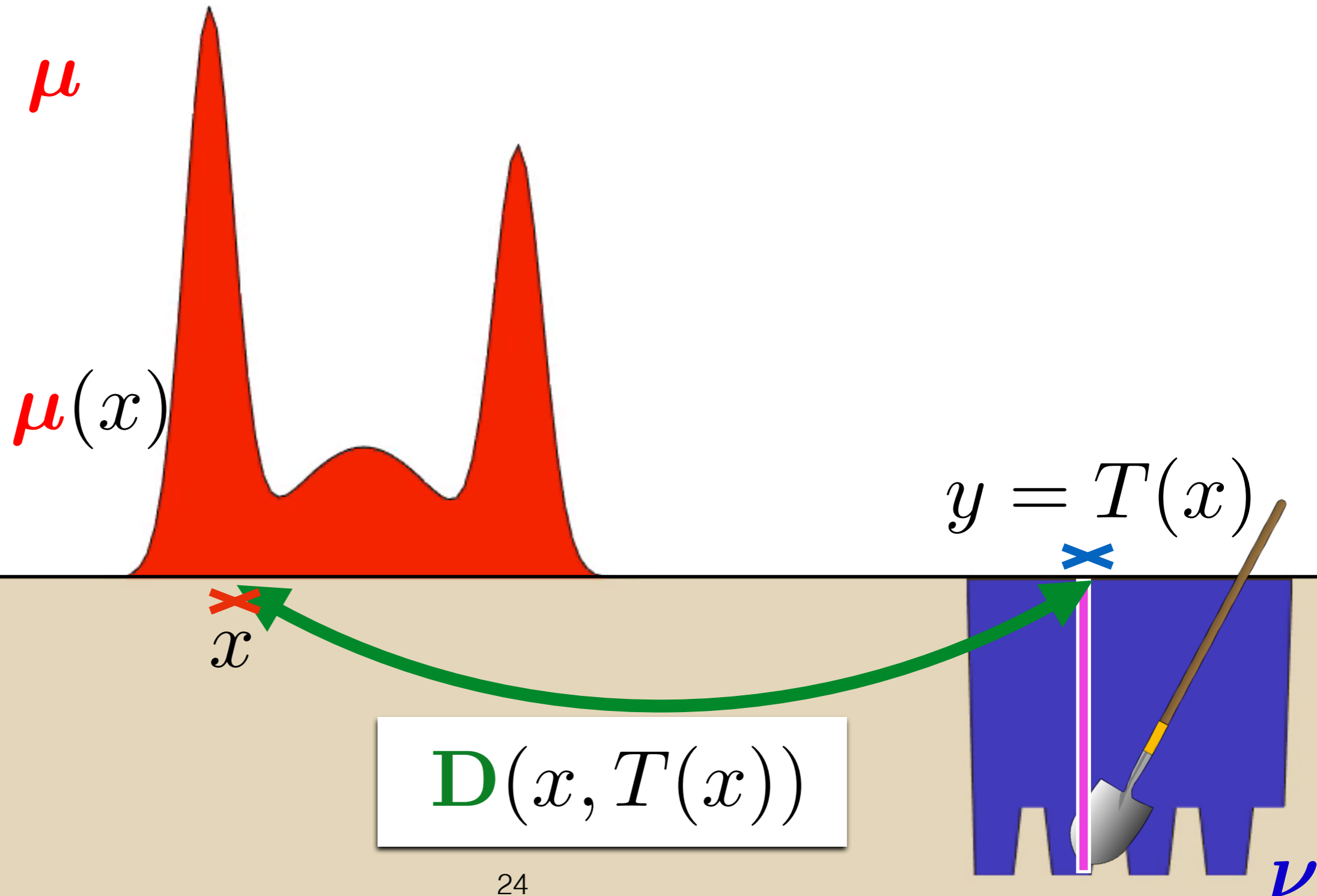
Origins: Monge's Problem

In 1781 however...



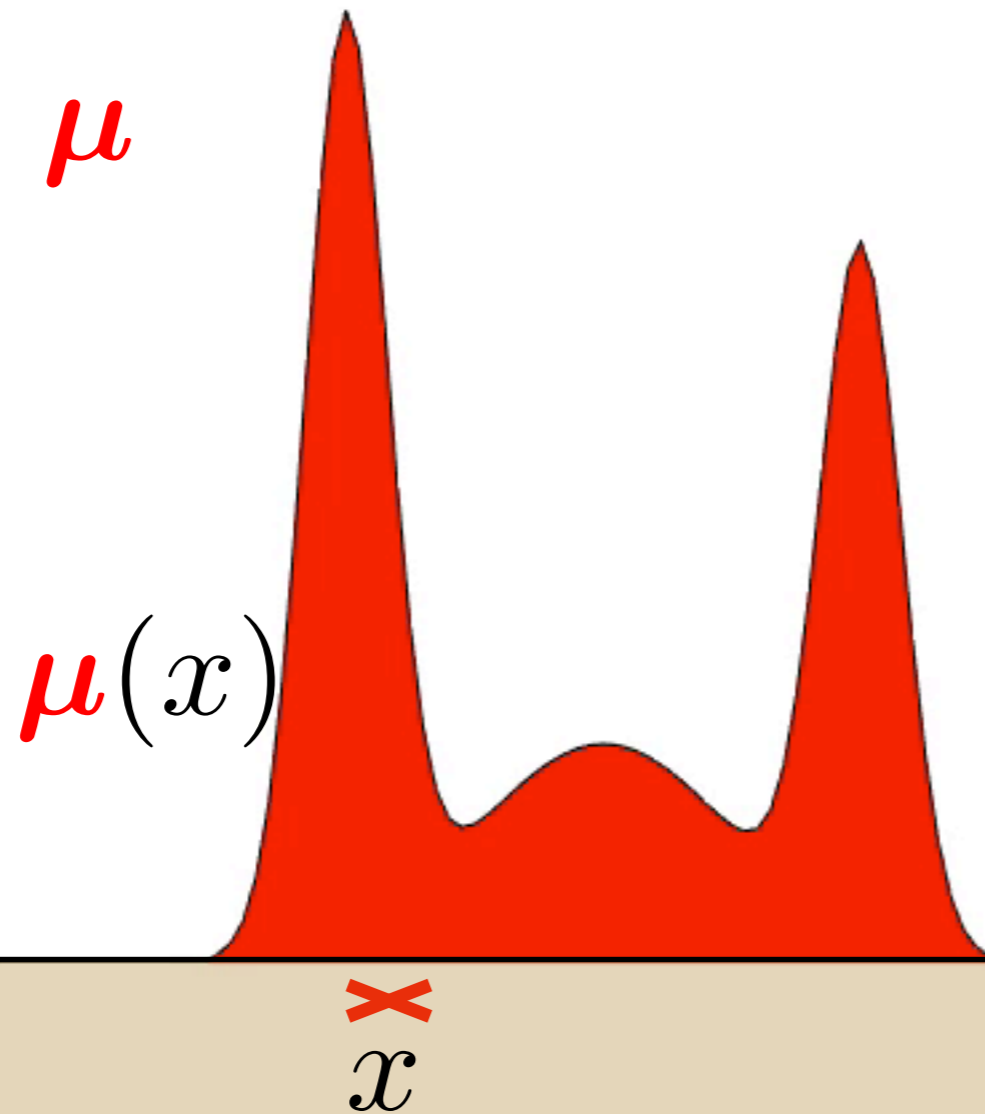
Origins: Monge's Problem

In 1781 however...

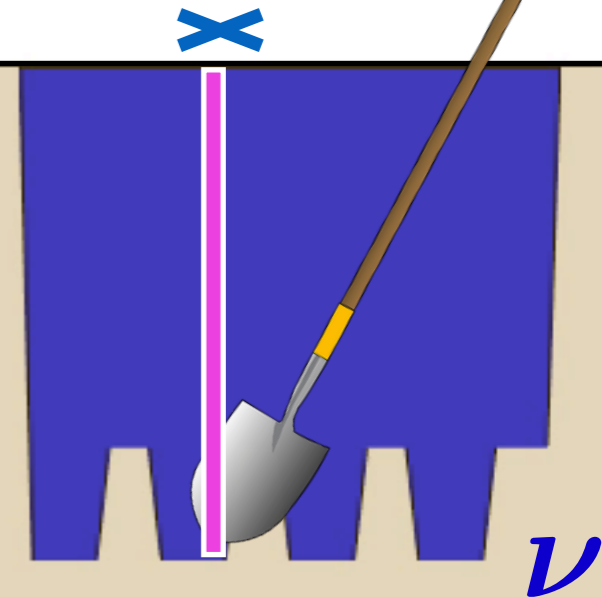


Origins: Monge's Problem

In 1781 however...



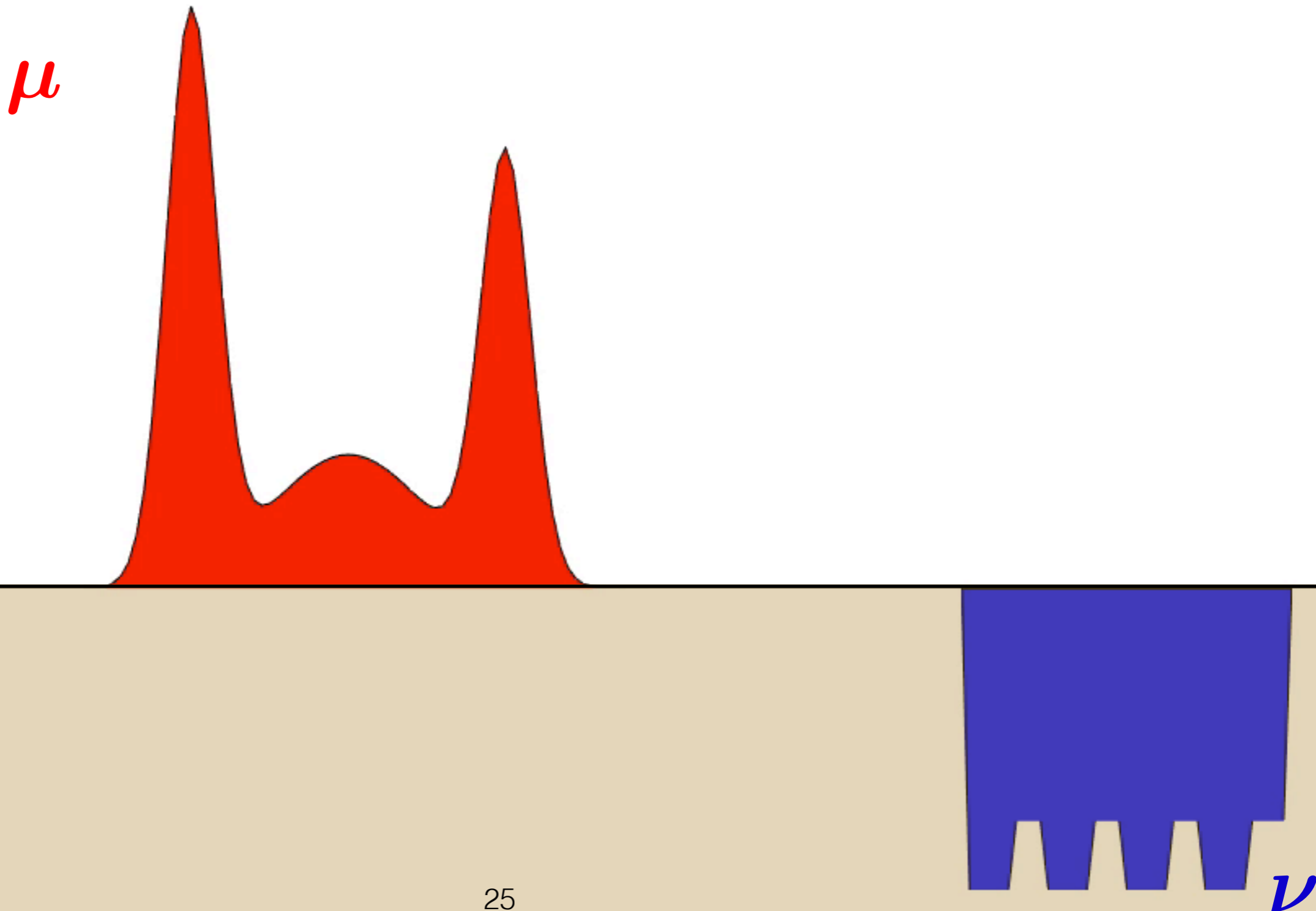
$$y = T(x)$$



work: $\mu(x) D(x, T(x))$

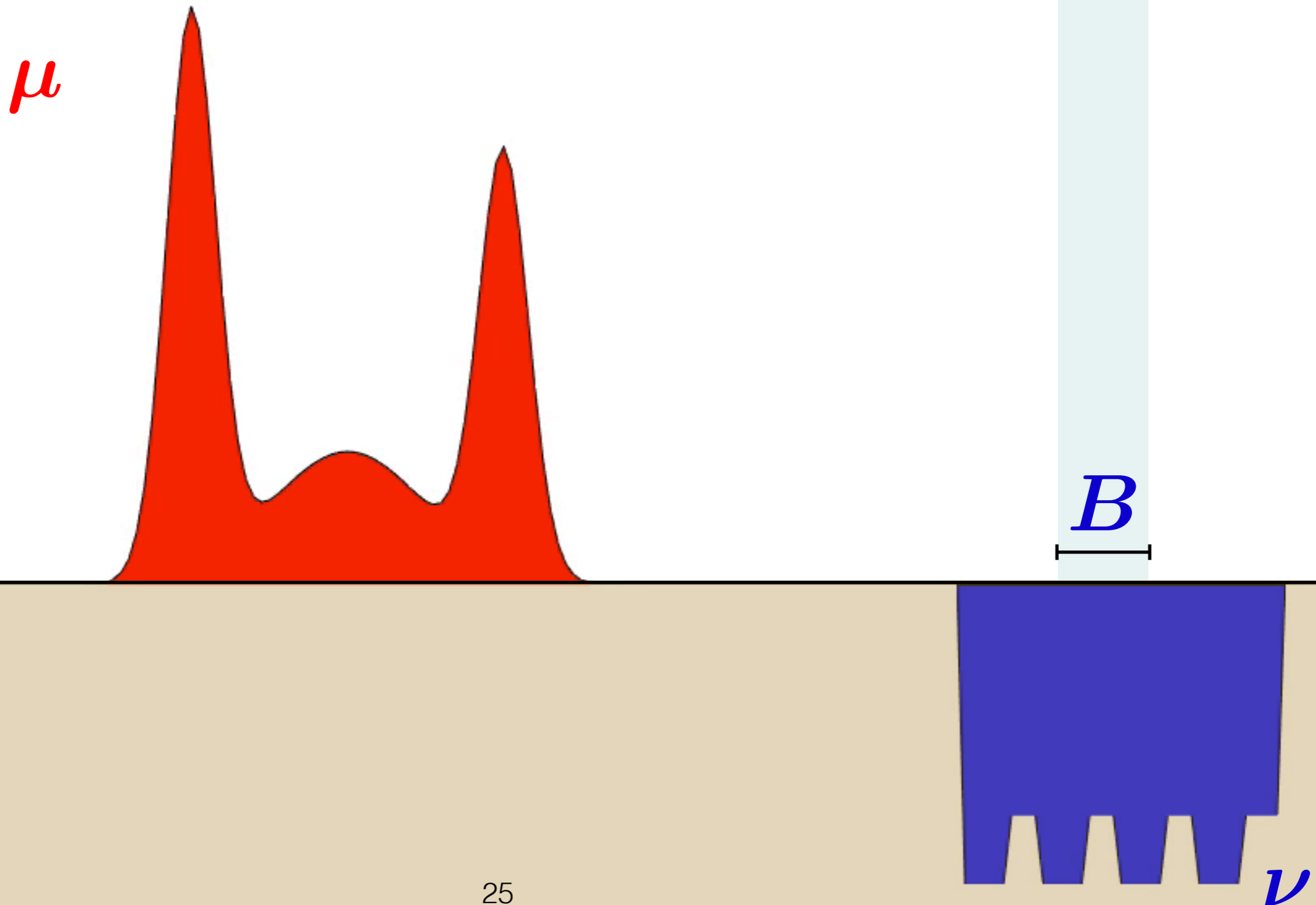
Origins: Monge's Problem

T must map red to blue.



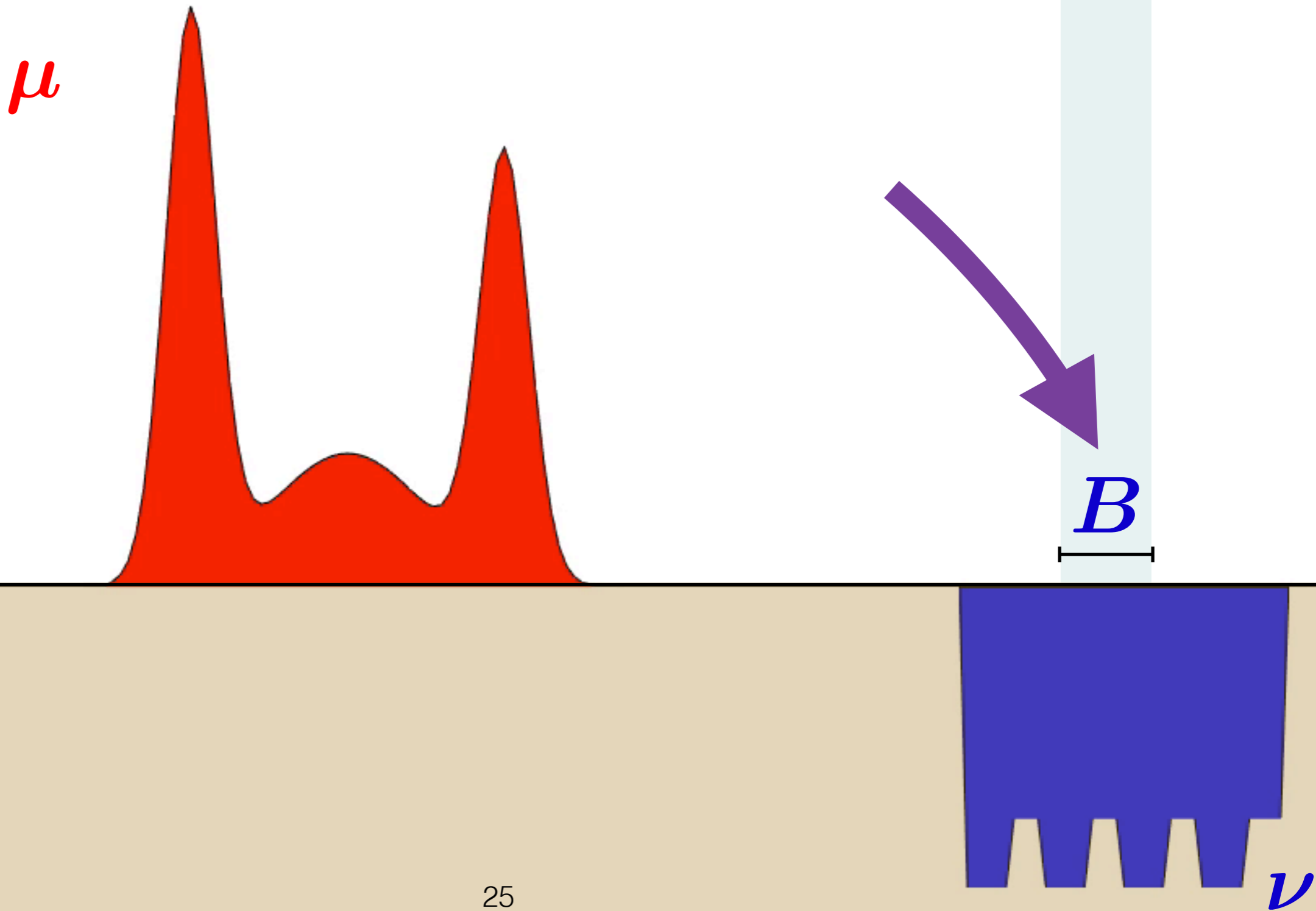
Origins: Monge's Problem

T must map red to blue.



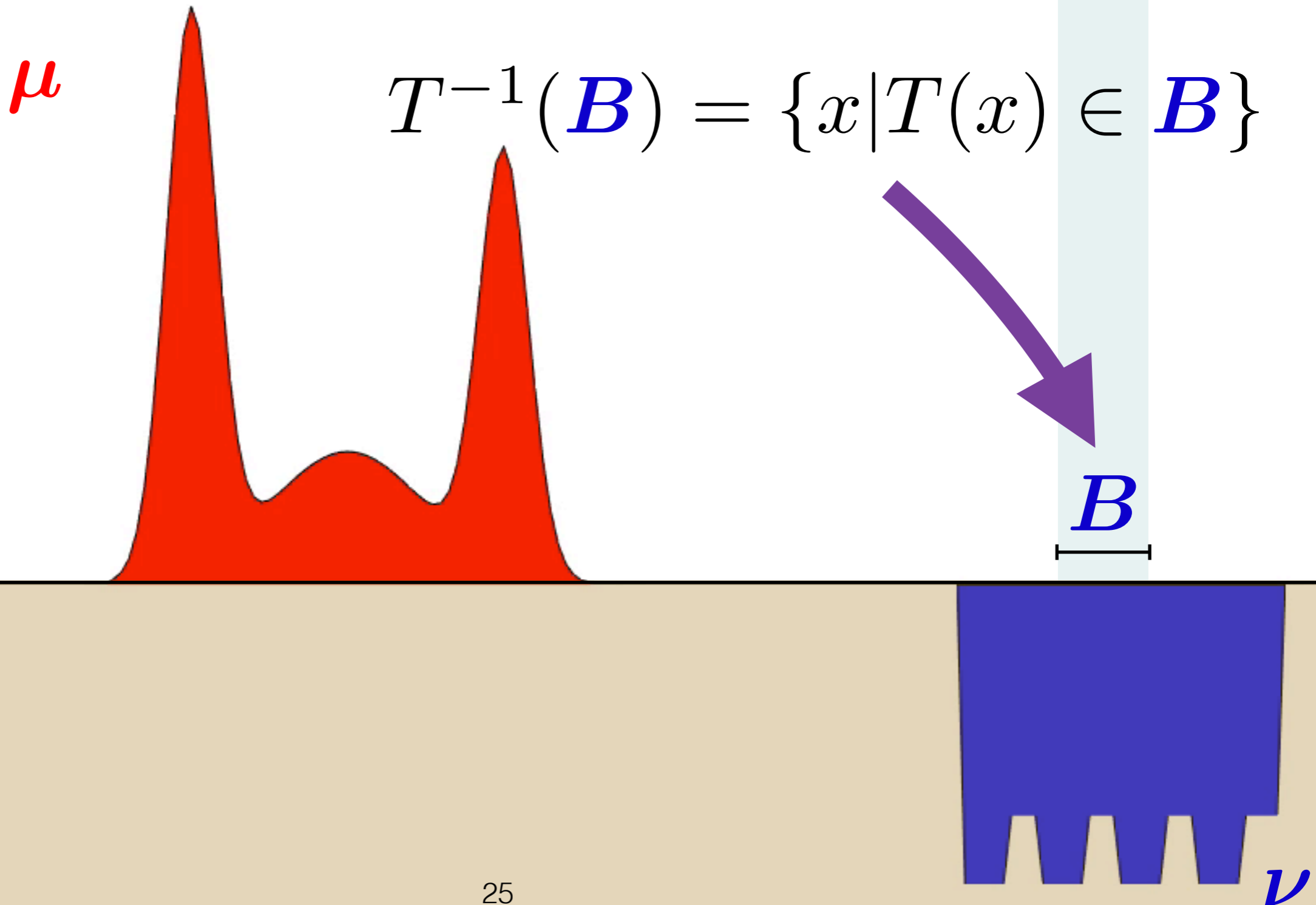
Origins: Monge's Problem

T must map red to blue.



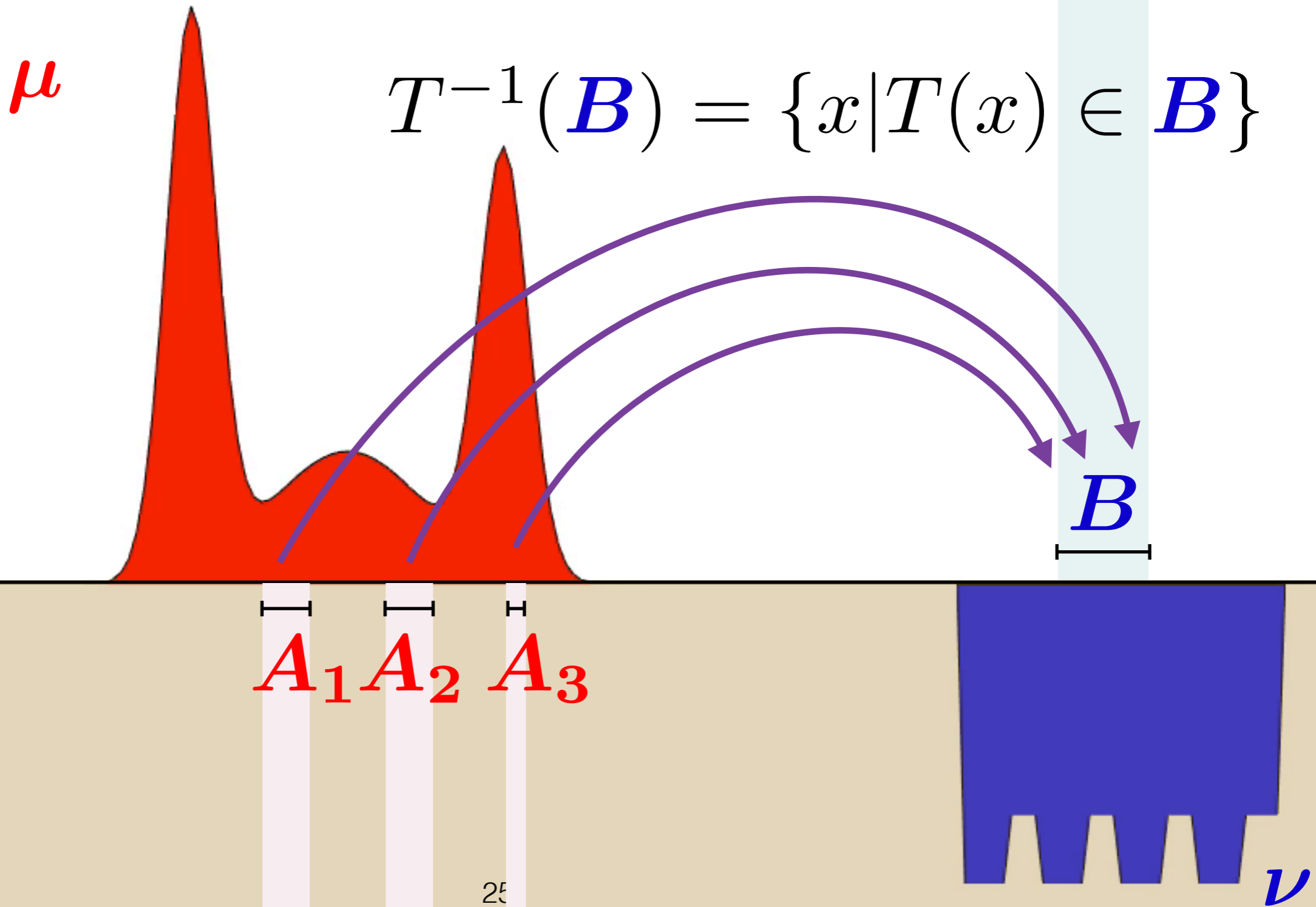
Origins: Monge's Problem

T must map red to blue.



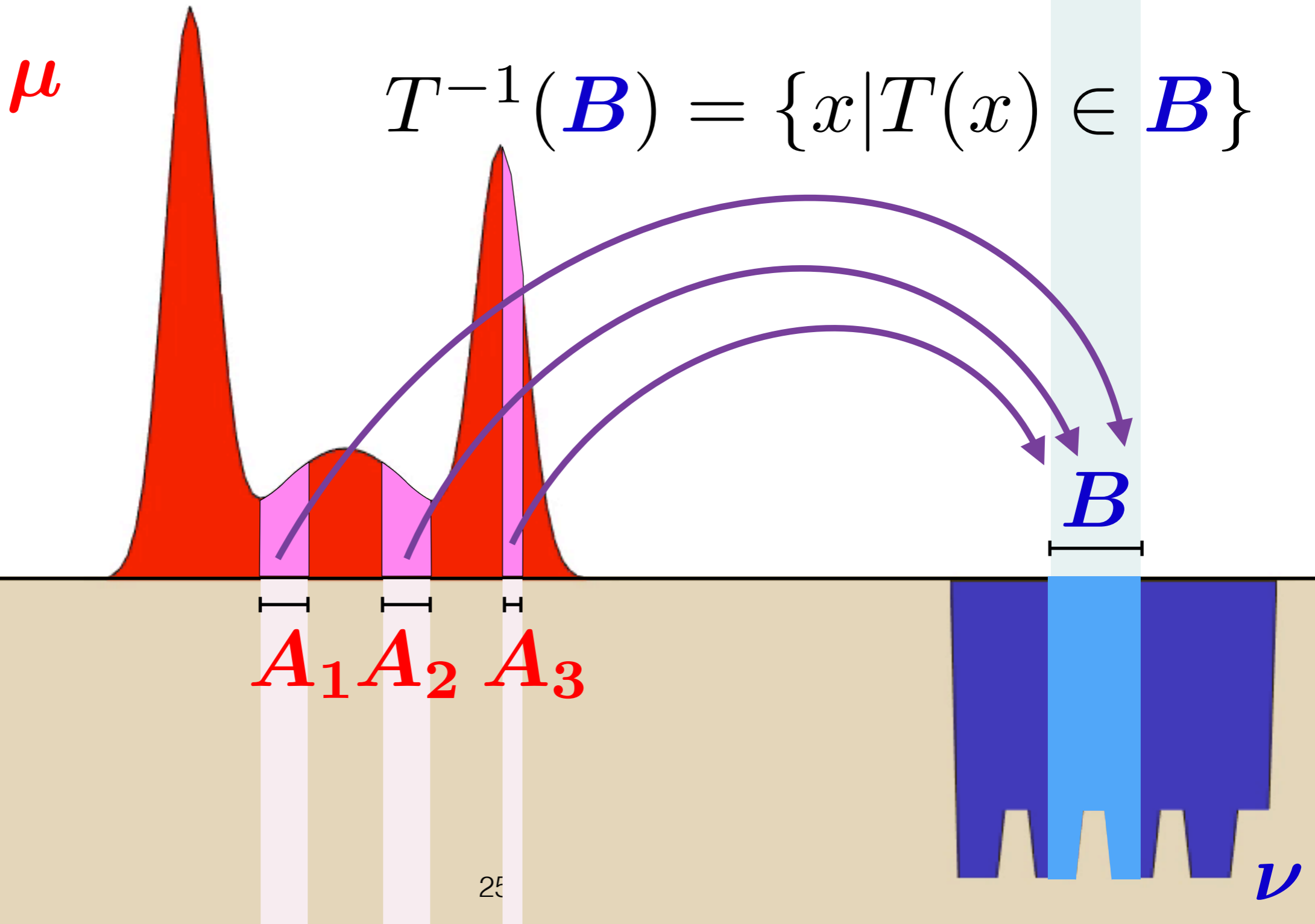
Origins: Monge's Problem

T must map red to blue.



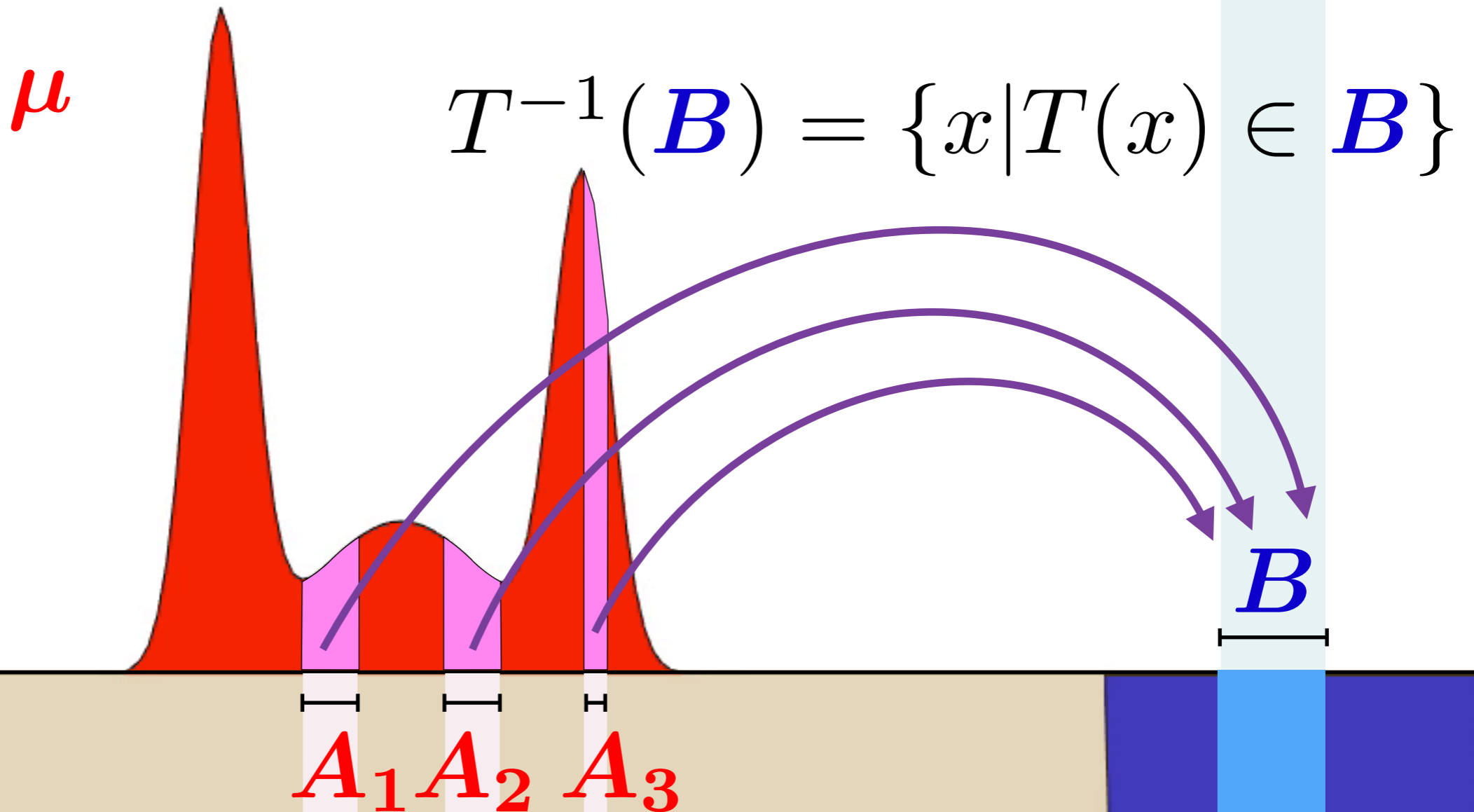
Origins: Monge's Problem

T must map red to blue.



Origins: Monge's Problem

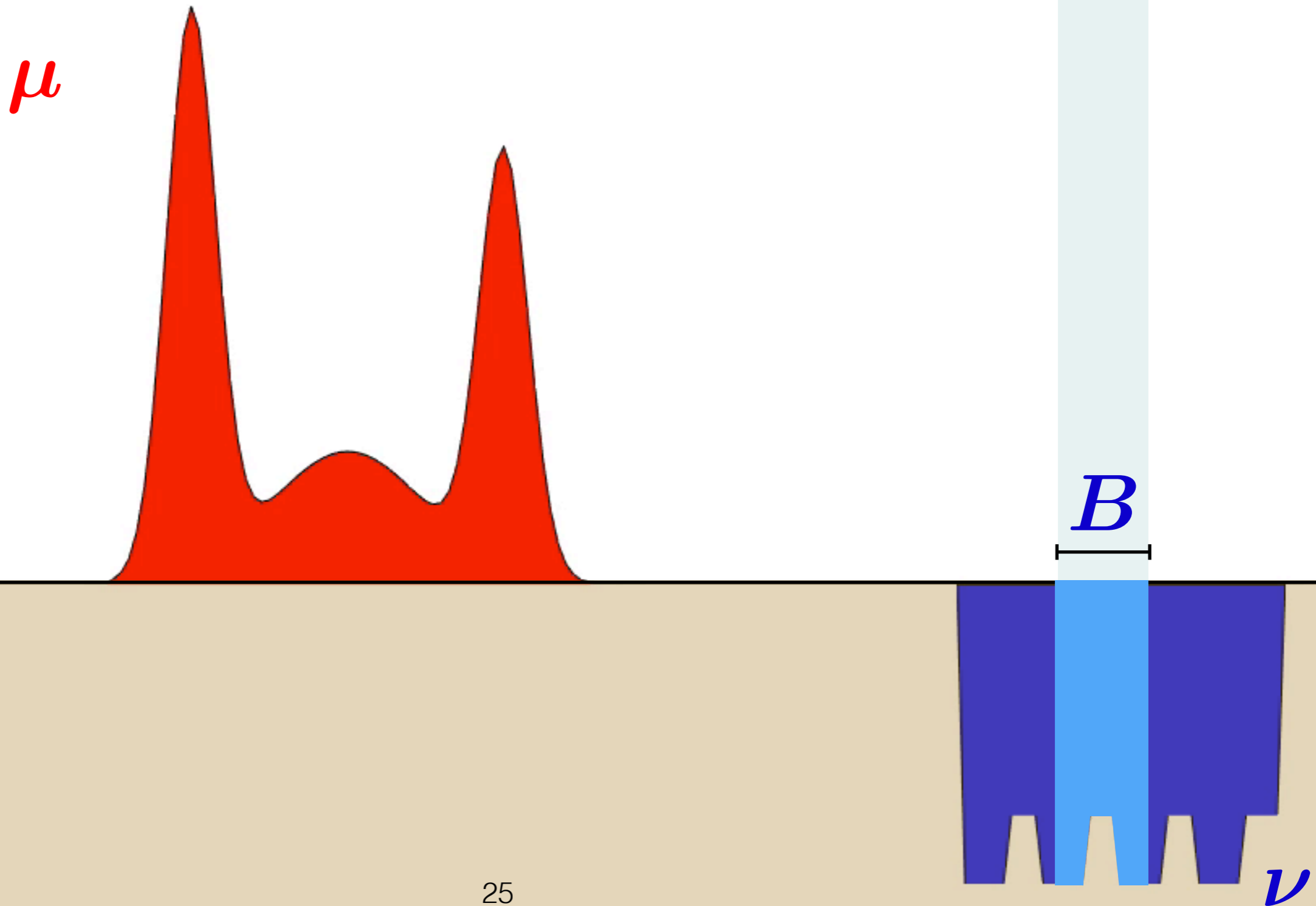
T must map red to blue.



$$\mu(A_1) + \mu(A_2) + \mu(A_3) = \nu(B)$$

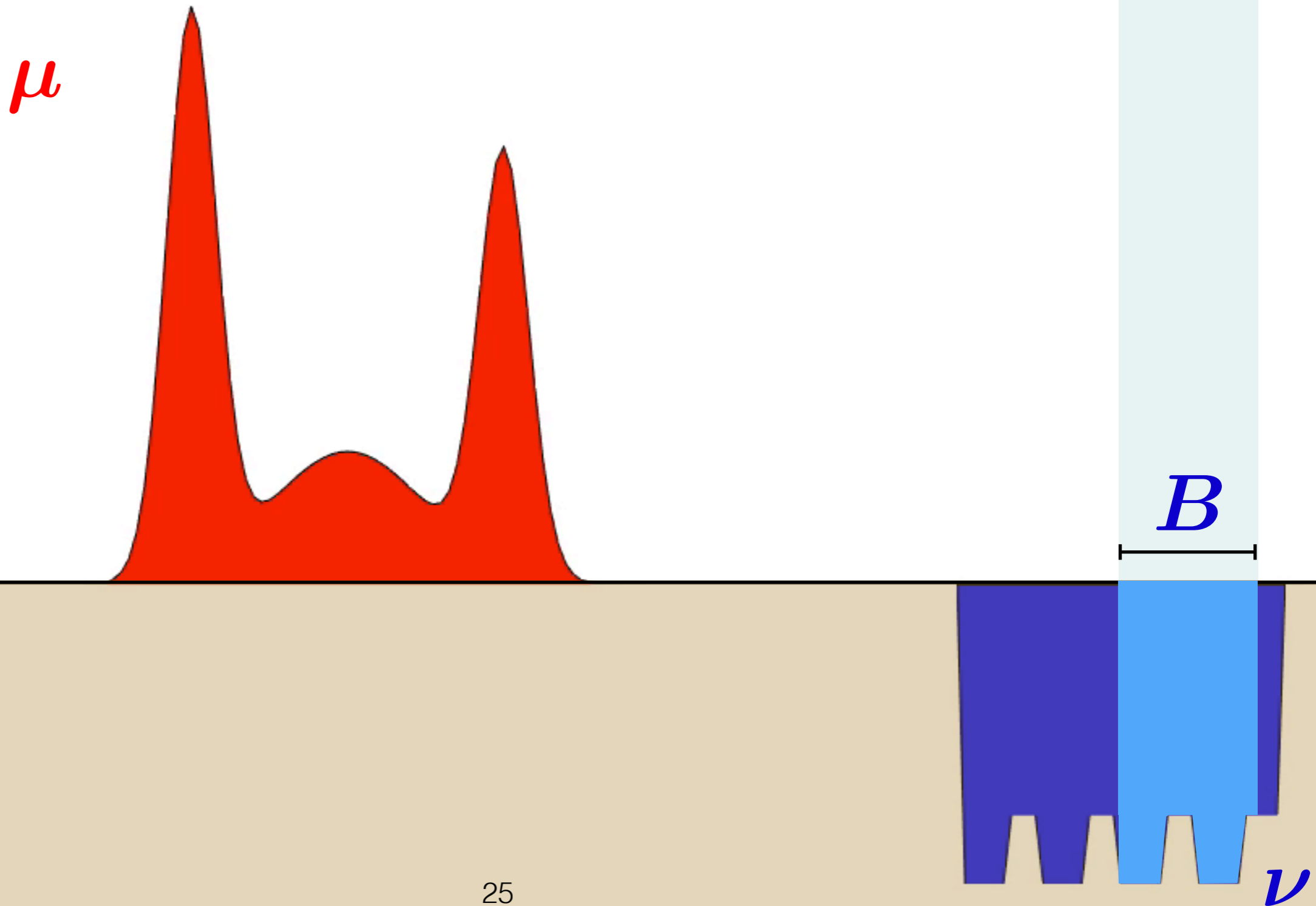
Origins: Monge's Problem

T must map red to blue.



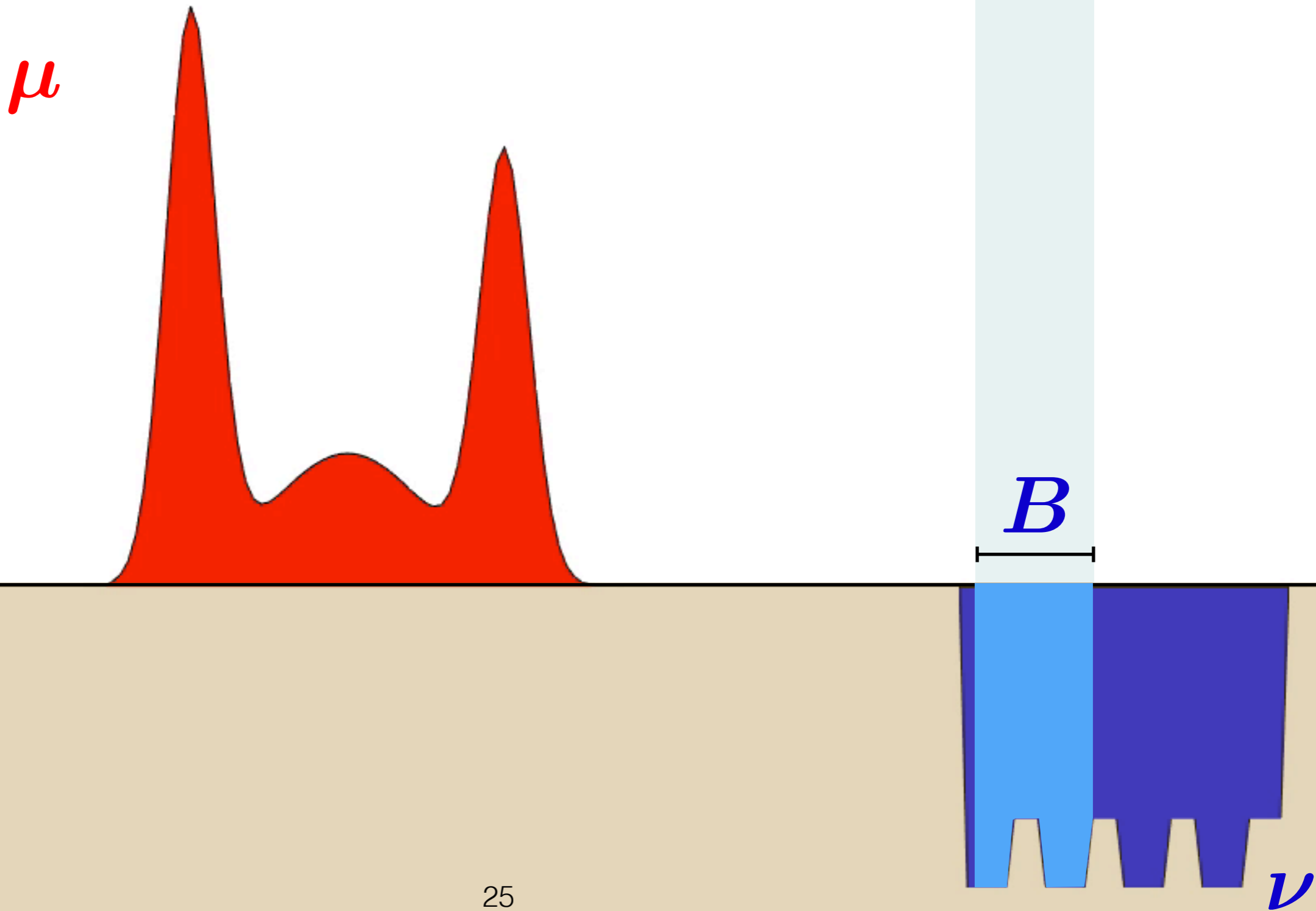
Origins: Monge's Problem

T must map red to blue.



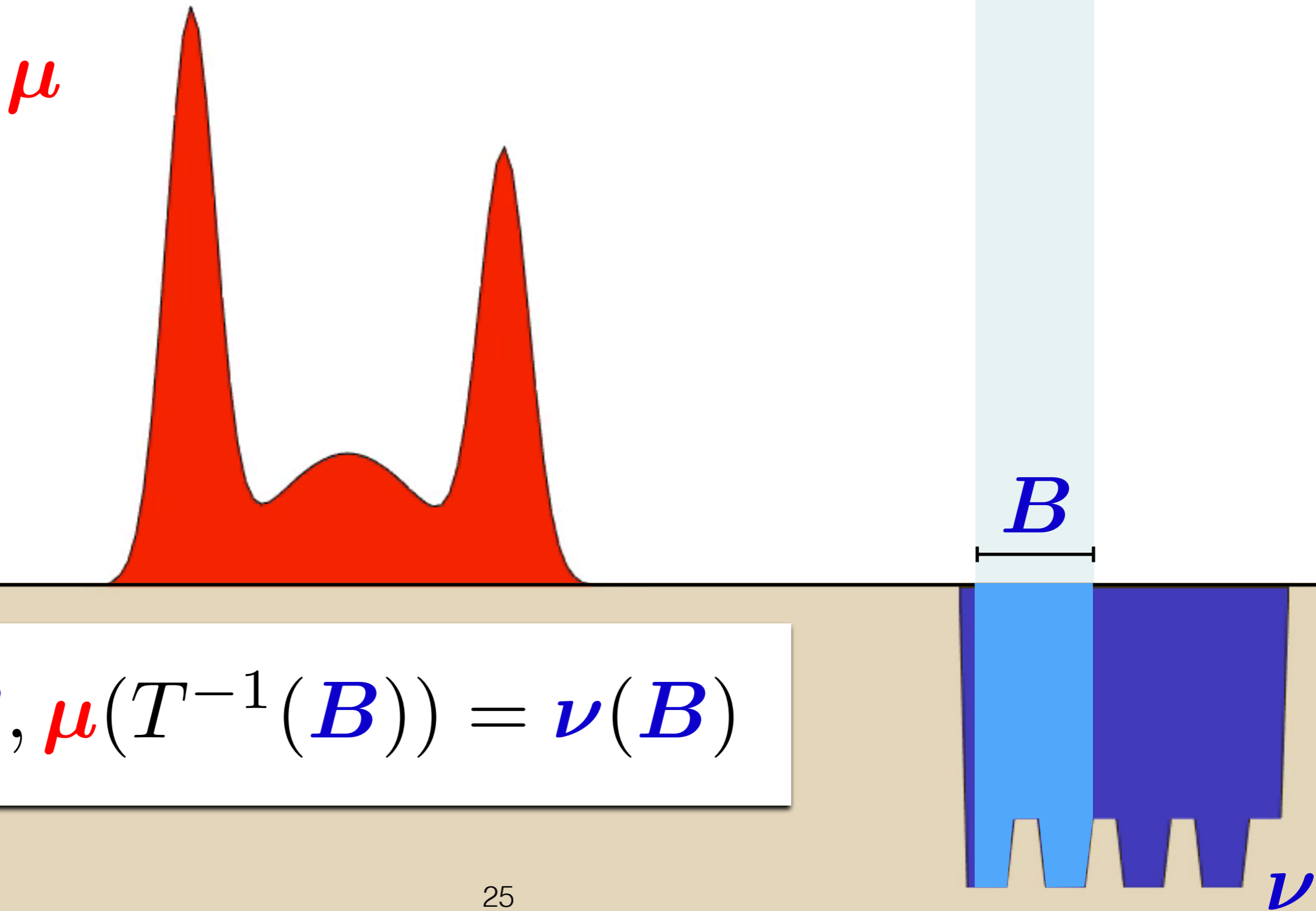
Origins: Monge's Problem

T must map red to blue.



Origins: Monge's Problem

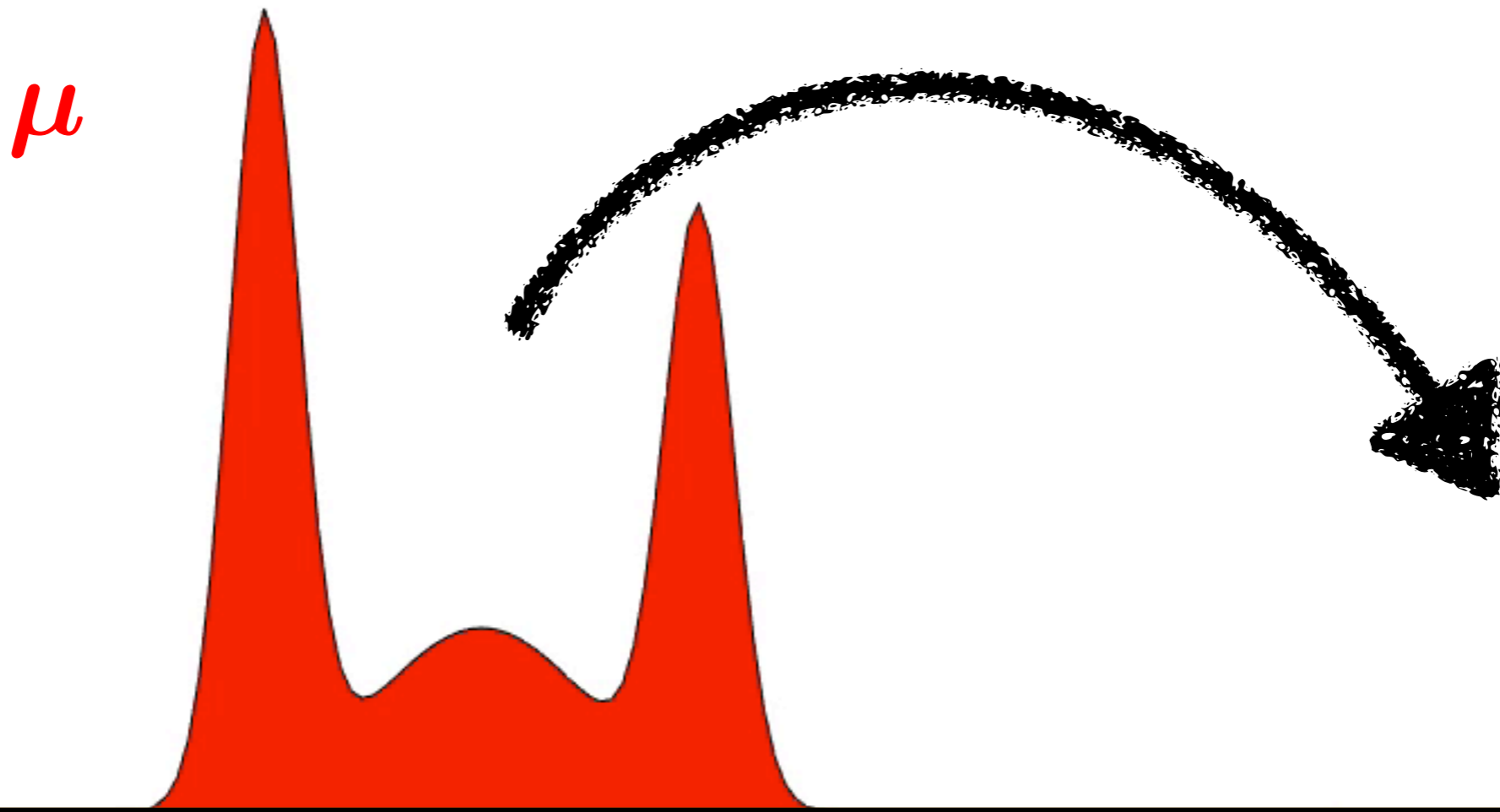
T must map red to blue.



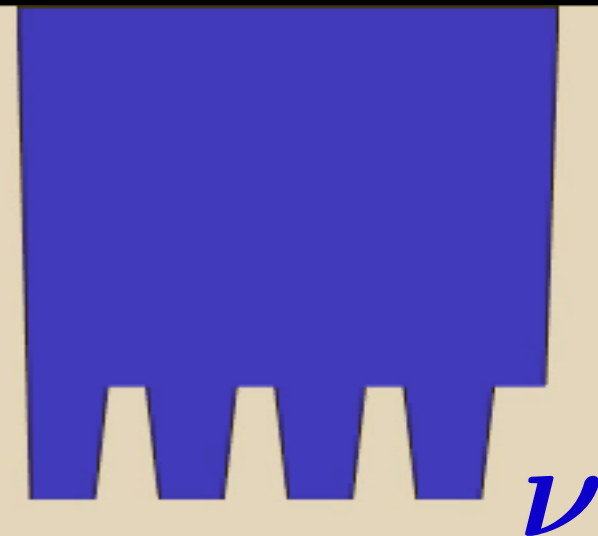
$$\forall B, \mu(T^{-1}(B)) = \nu(B)$$

Origins: Monge's Problem

T must **push-forward** the red measure towards the blue

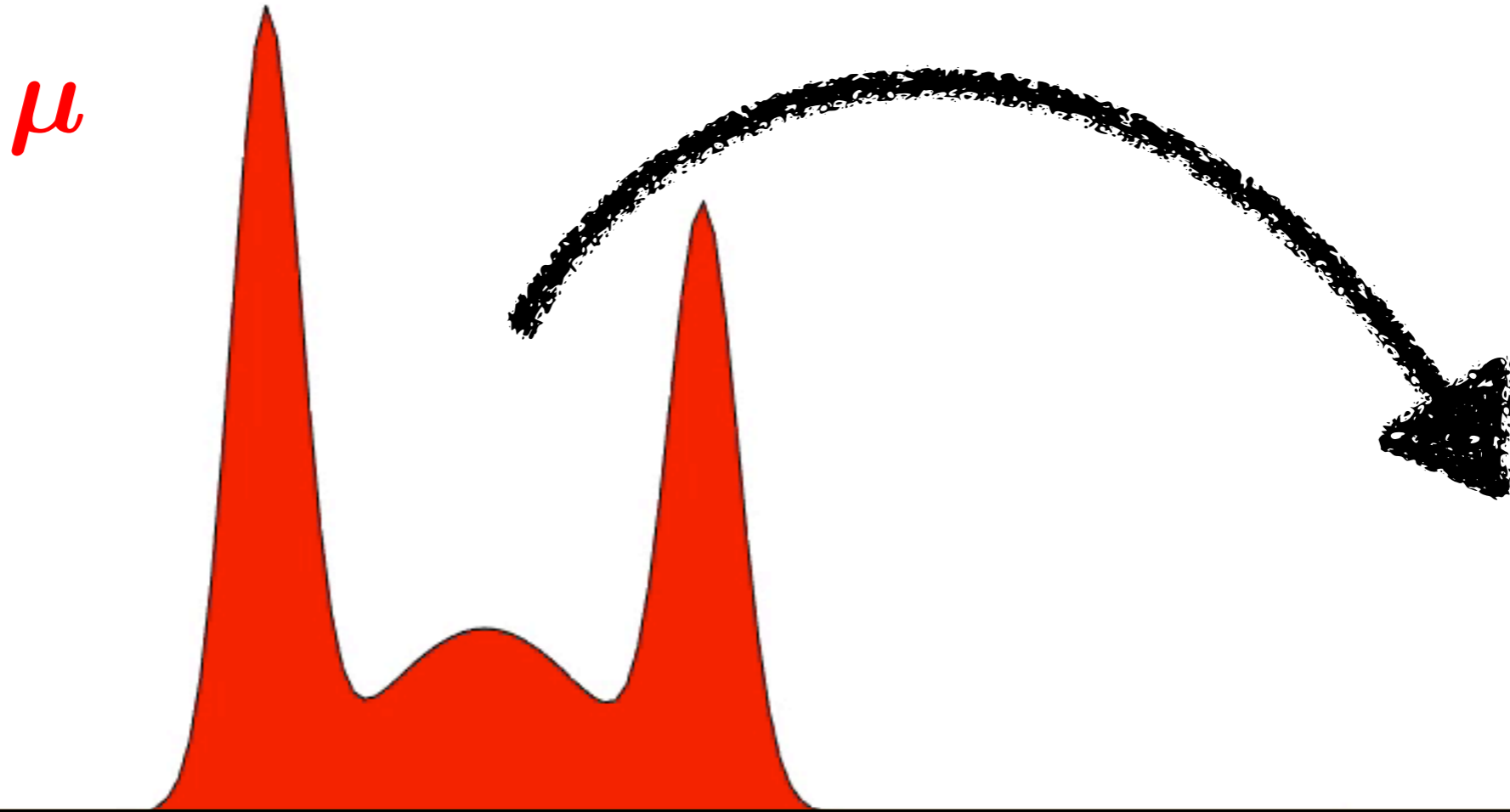


$$T_{\#} \mu = \nu$$



Origins: Monge's Problem

T must **push-forward** the red measure towards the blue



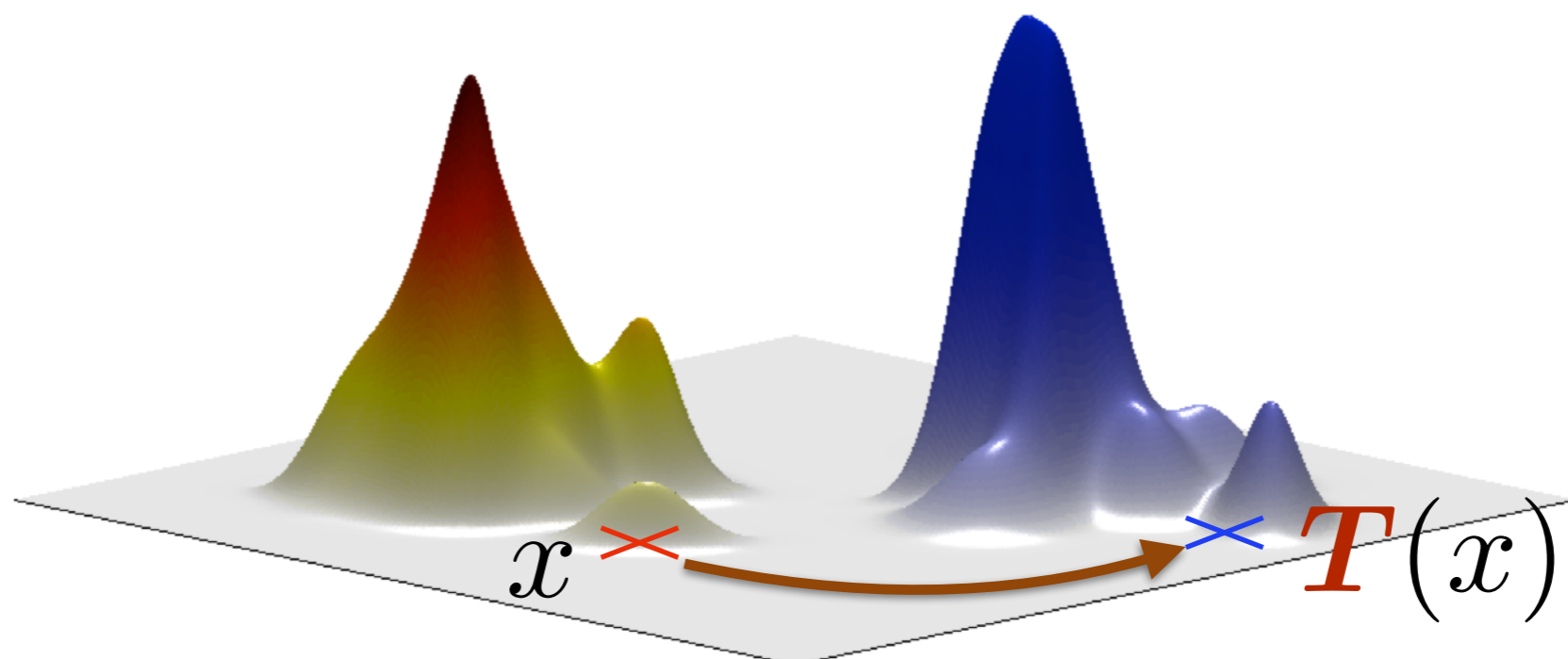
What T s.t. $T_{\#}\mu = \nu$
minimizes $\int D(x, T(x)) \mu(dx)$?

Monge Problem

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

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

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

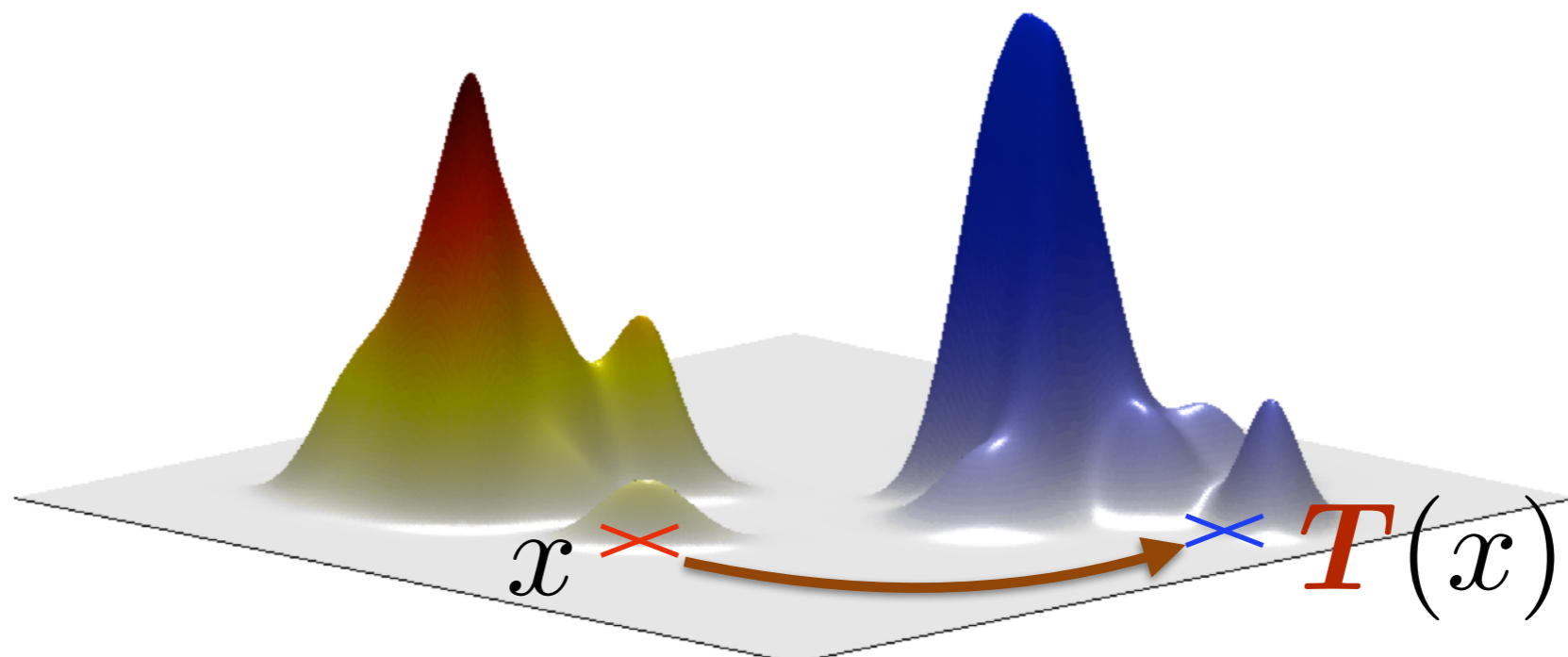


Monge Problem

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

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

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

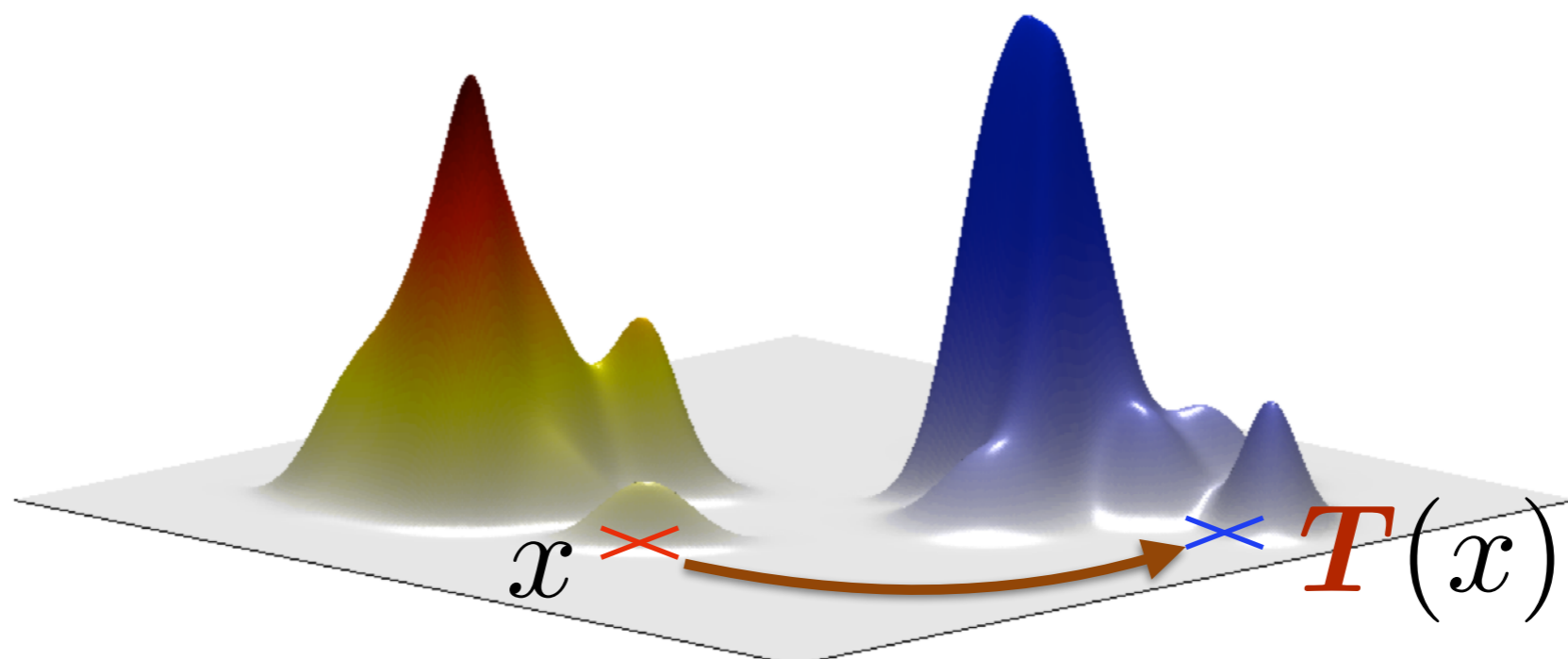


Monge Problem

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

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

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

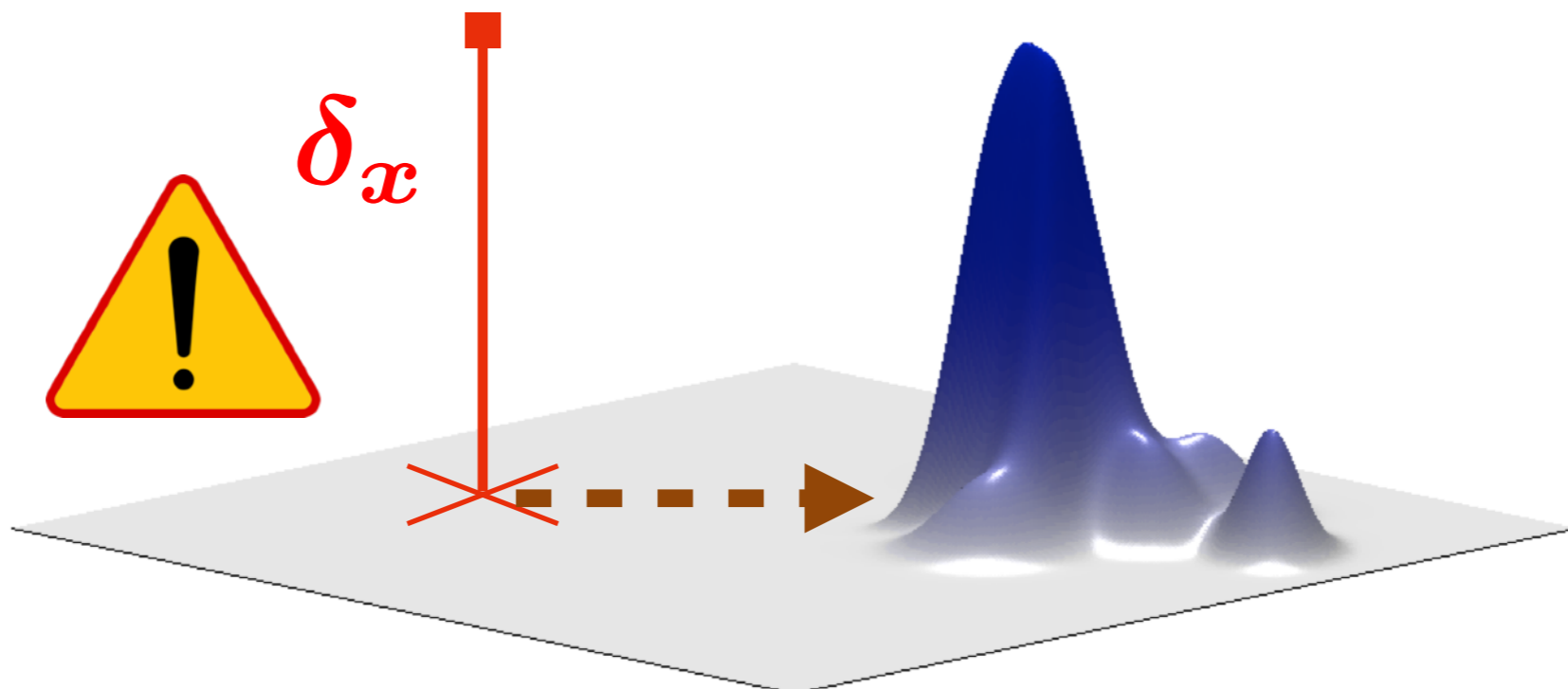


Monge Problem

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

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

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



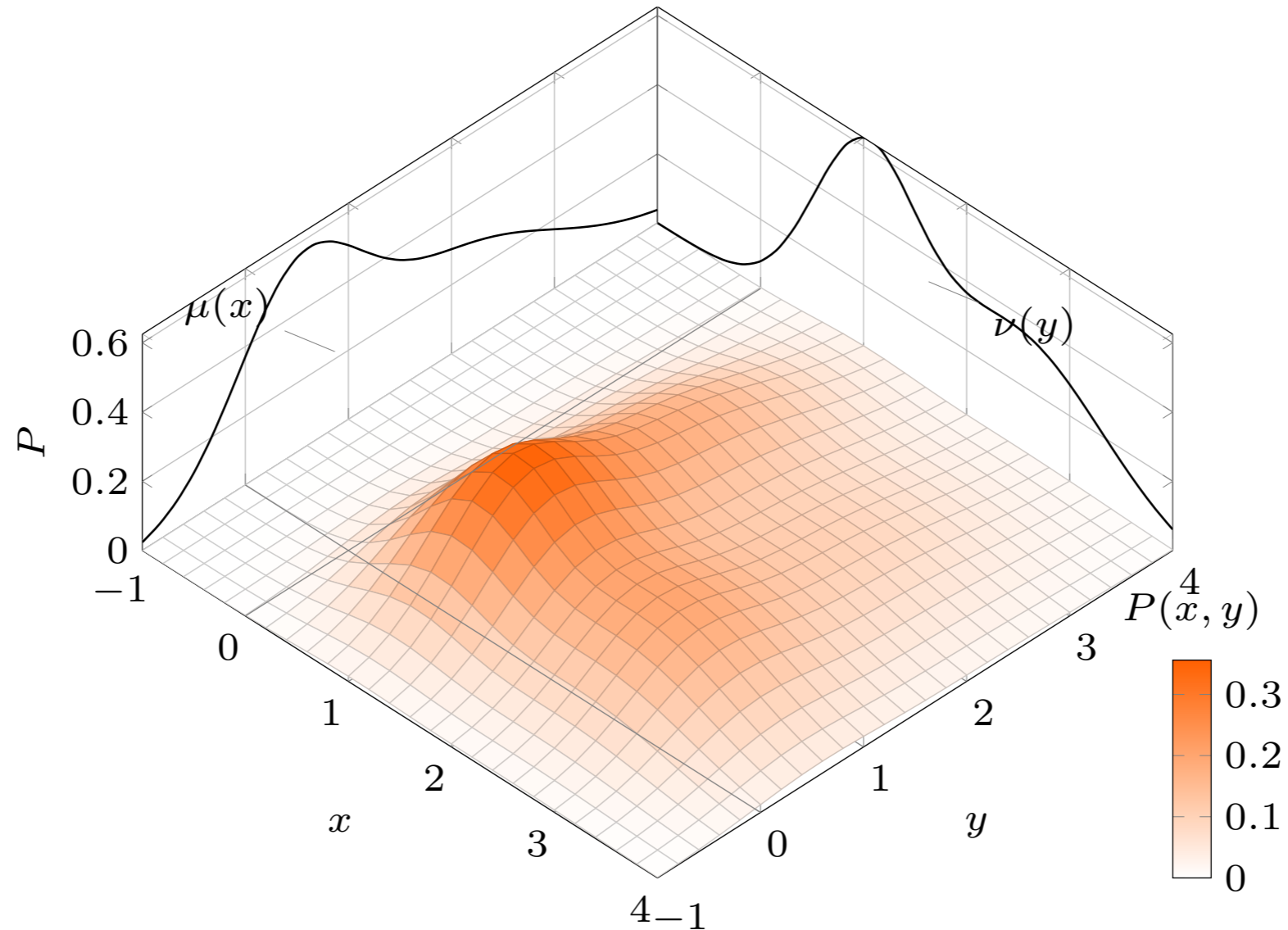
Kantorovich Relaxation

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

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

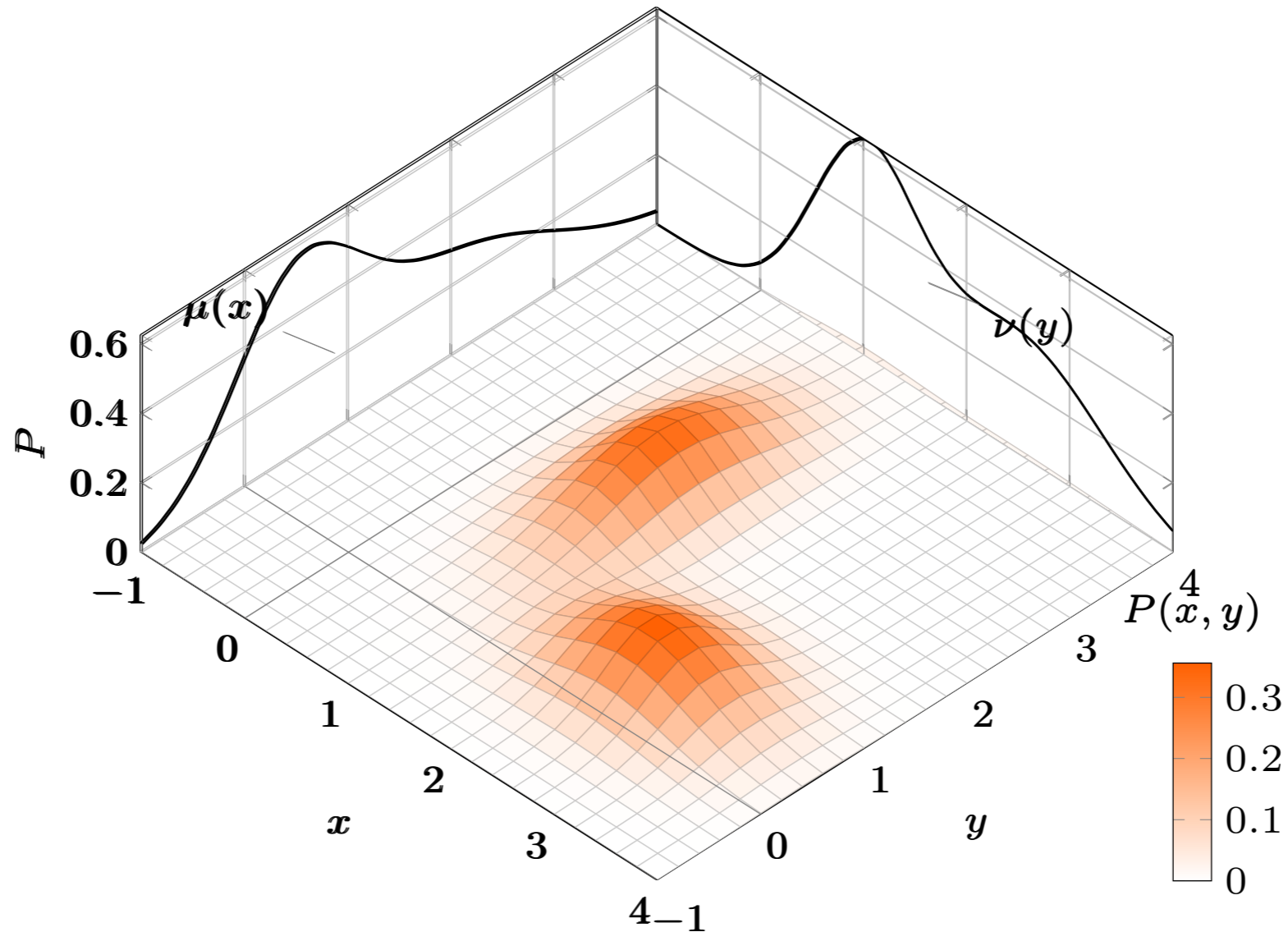
[Kantorovich'42] Relaxation

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



[Kantorovich'42] Relaxation

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



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) Wasserstein Distances

Let $p \geq 1$.

Let $c := D$, a metric.

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

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

(Kantorovich) Wasserstein Distances

Let $p \geq 1$.

Let $c := D$, a metric.

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

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

Computational OT

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

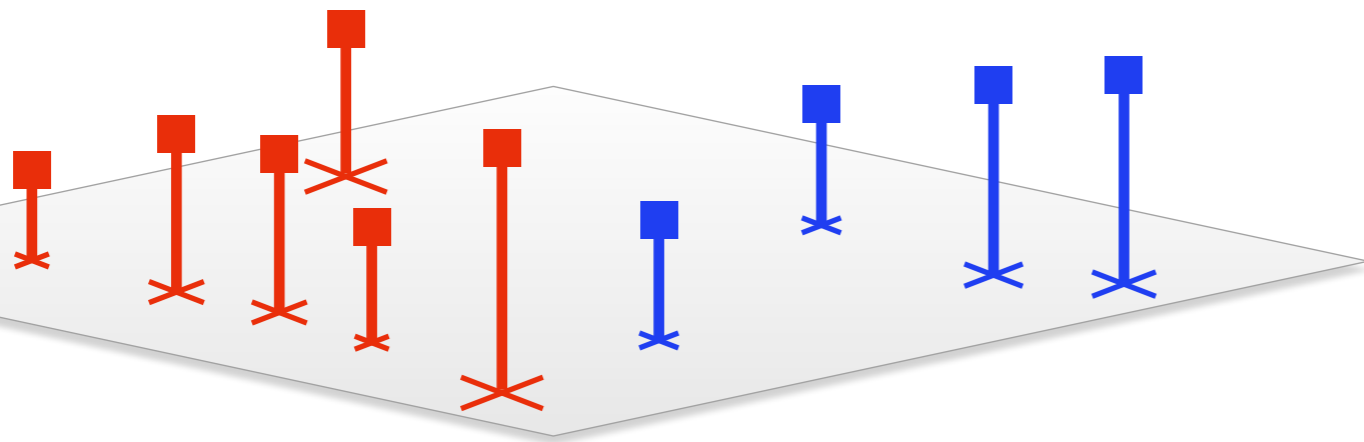
Goal now: use OT as a **loss or fidelity** term

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

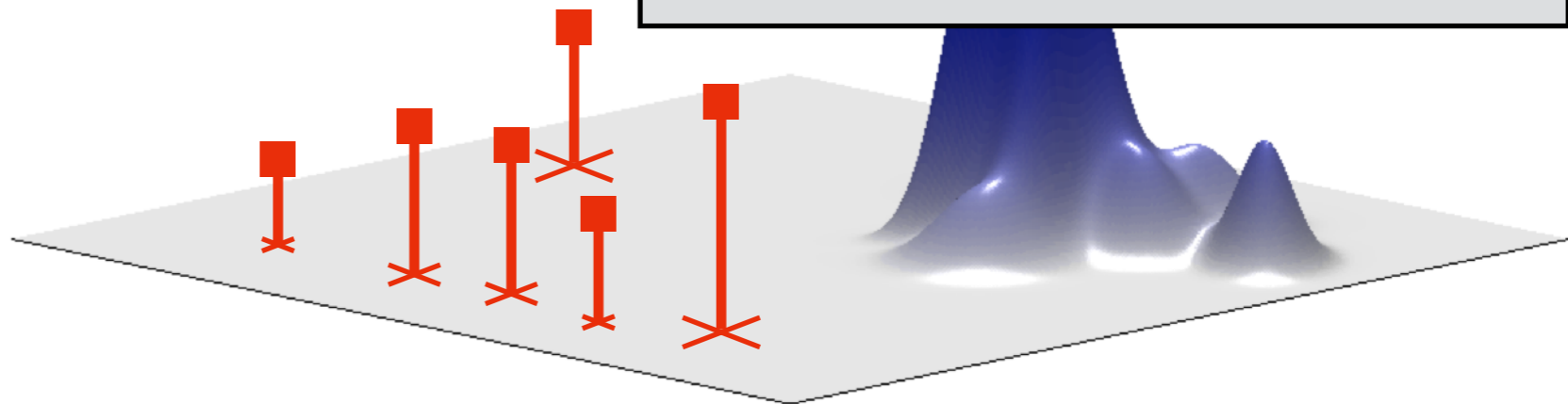
$\nabla_{\mu} W_p(\mu, \nu_1) = ?$

How can we compute OT?

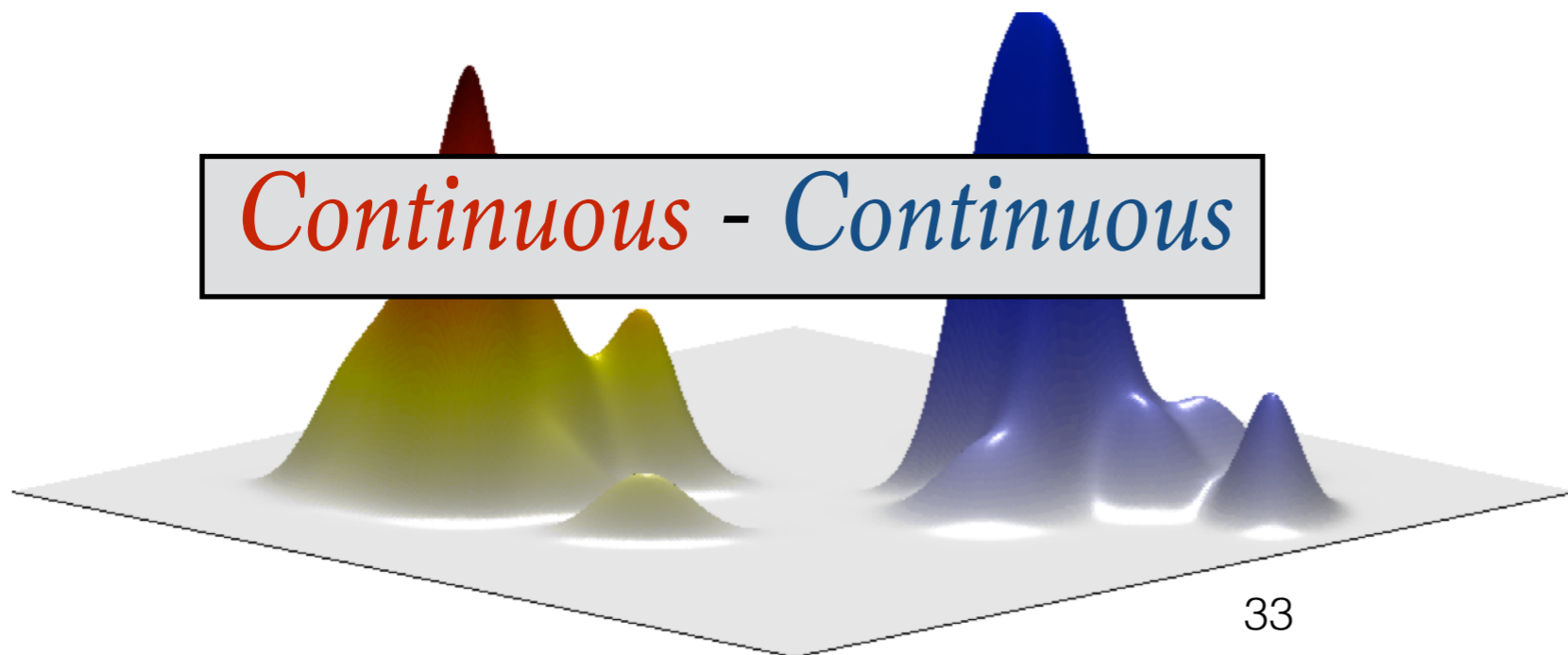
Discrete - Discrete



Discrete - Continuous



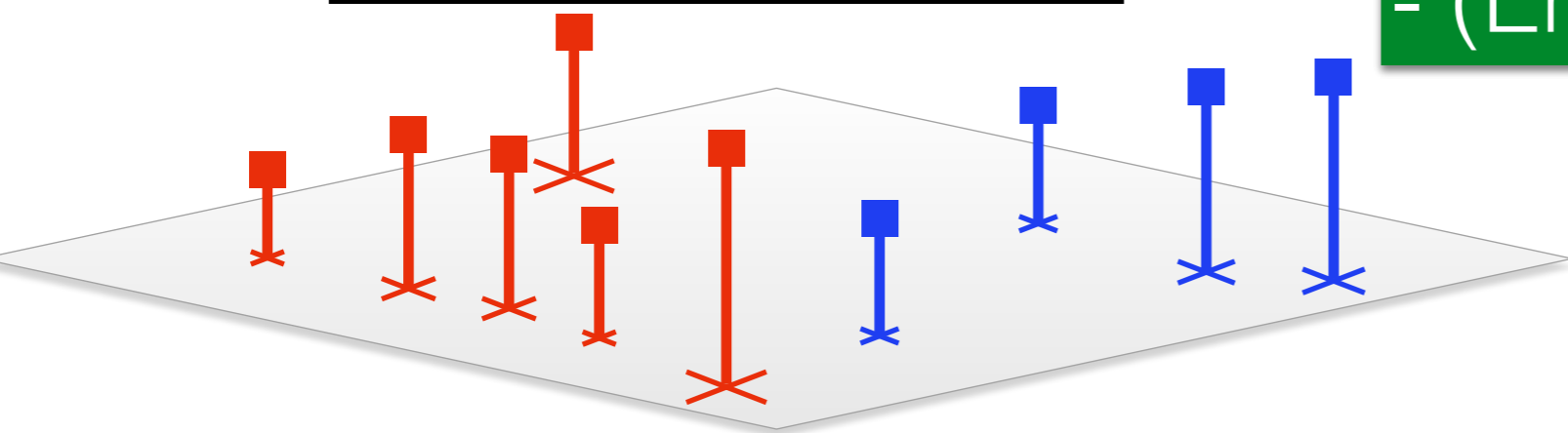
Continuous - Continuous



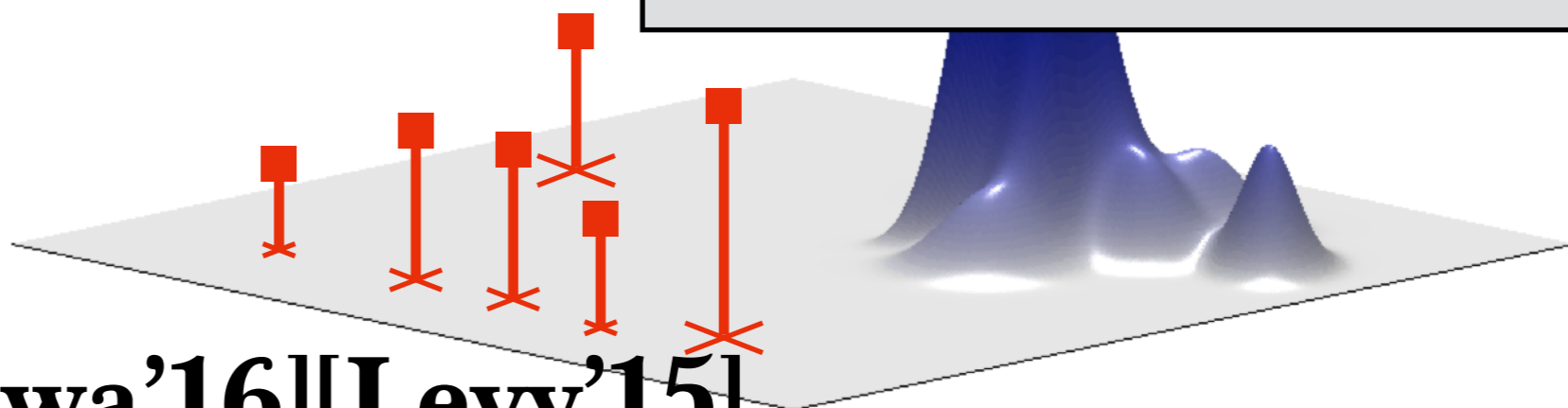
How can we compute OT?

Discrete - Discrete

- Network flow solvers
- (Entropic) regularization



Discrete - Continuous



low dim.

[Mérigot'11][Kitagawa'16][Levy'15]

Continuous - Continuous

Stochastic
Optimization

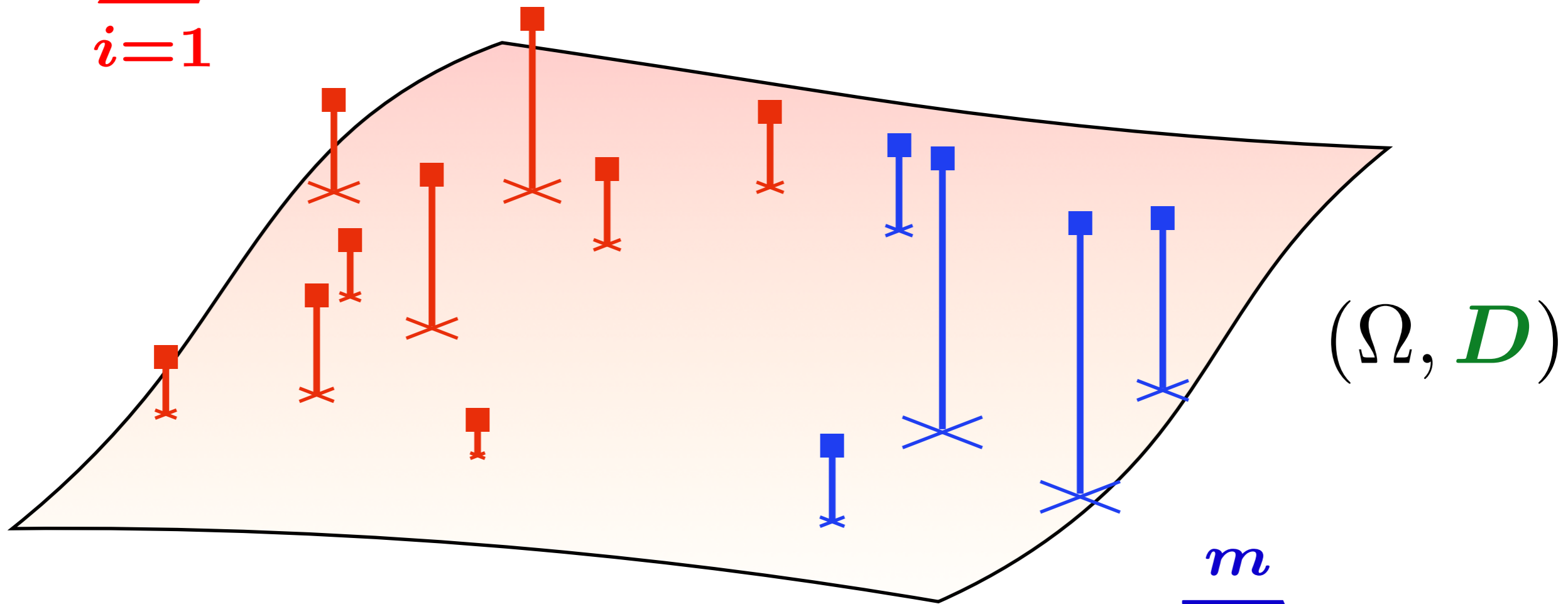
[Genevay'16]

PDE's

[Benamou'98]

OT on Two Empirical Measures

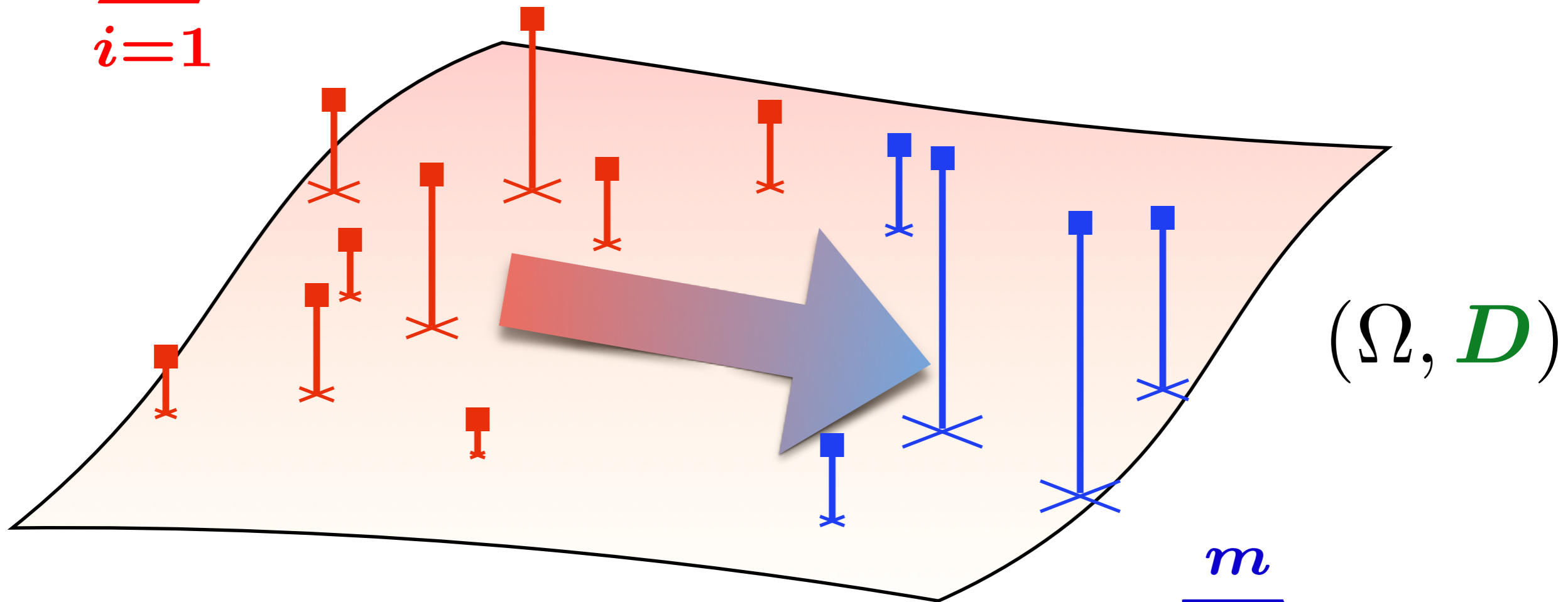
$$\mu = \sum_{i=1}^n a_i \delta_{x_i}$$



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

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

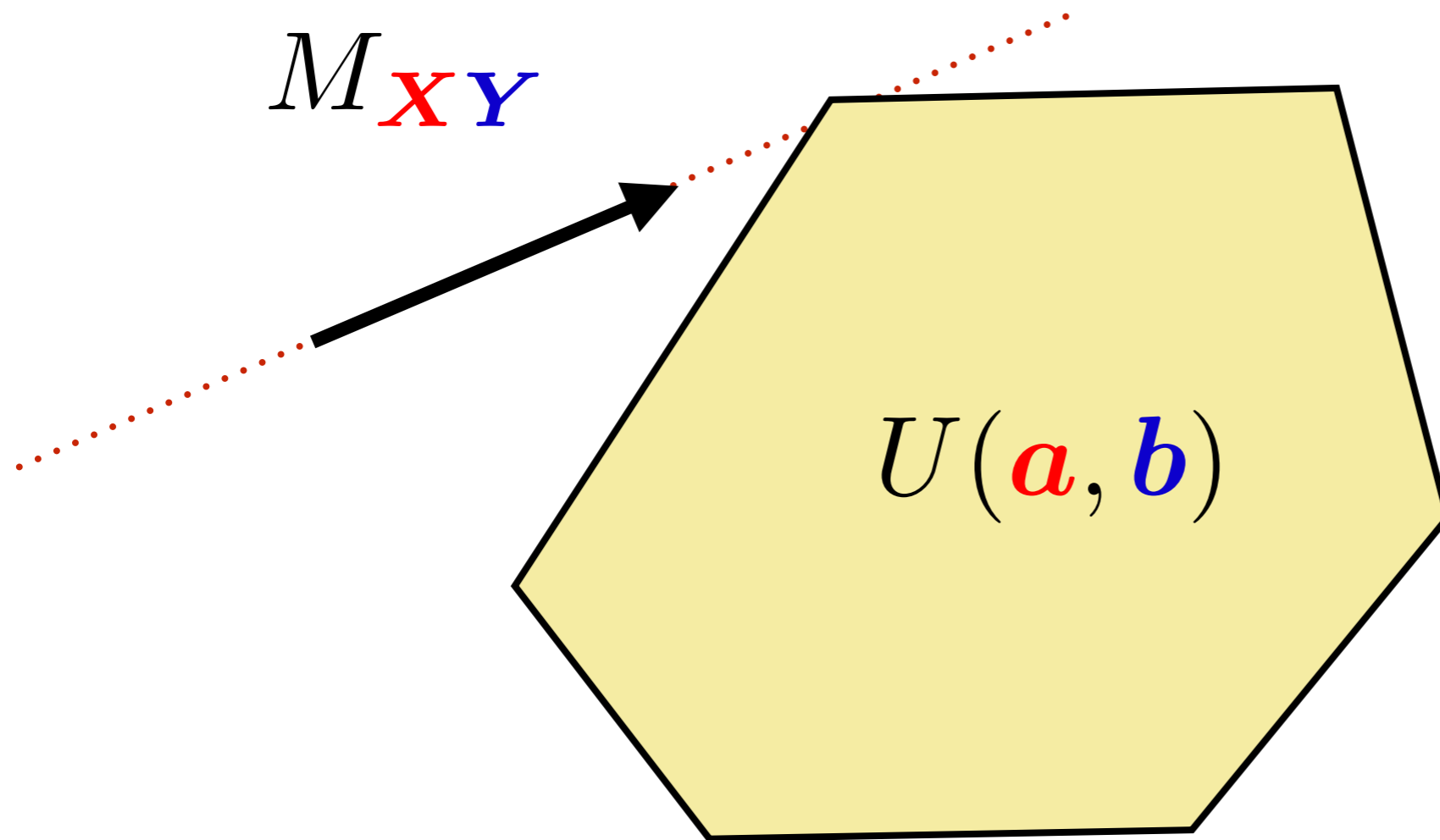
$$M_{\mathbf{X}\mathbf{Y}} \stackrel{\text{def}}{=} [D(\mathbf{x}_i, \mathbf{y}_j)^p]_{ij}$$

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

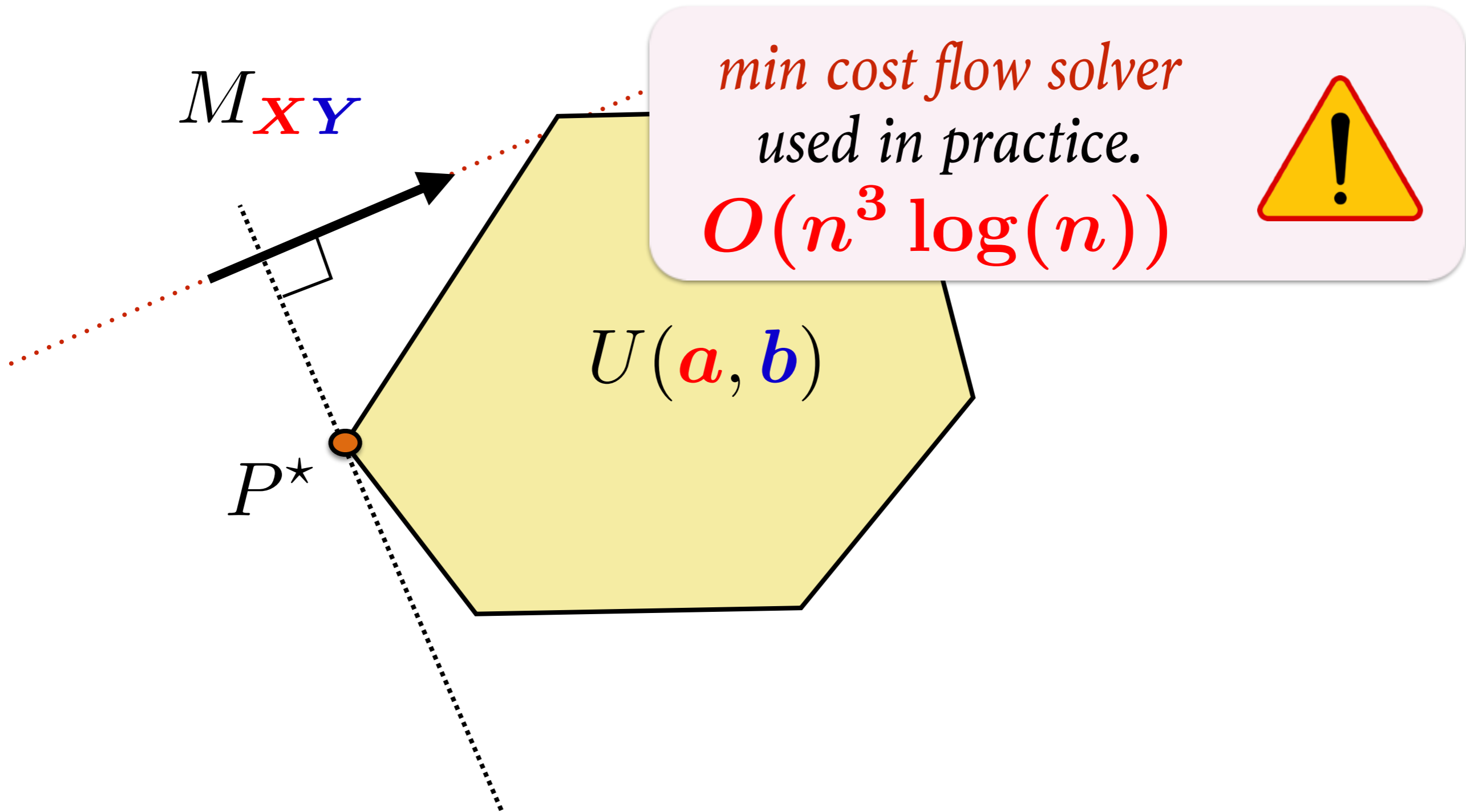
Def. Optimal Transport Problem

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

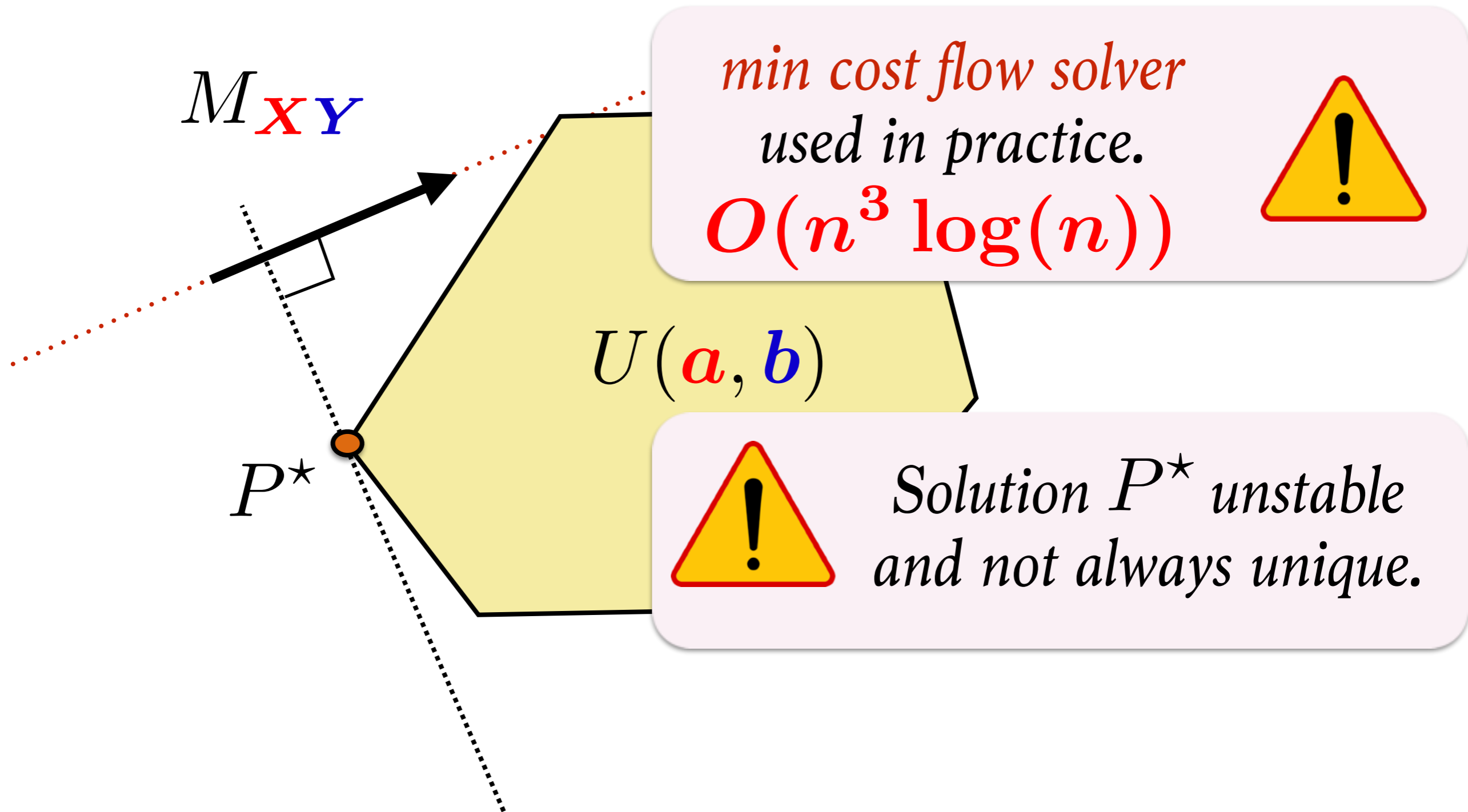
Solving the OT Problem



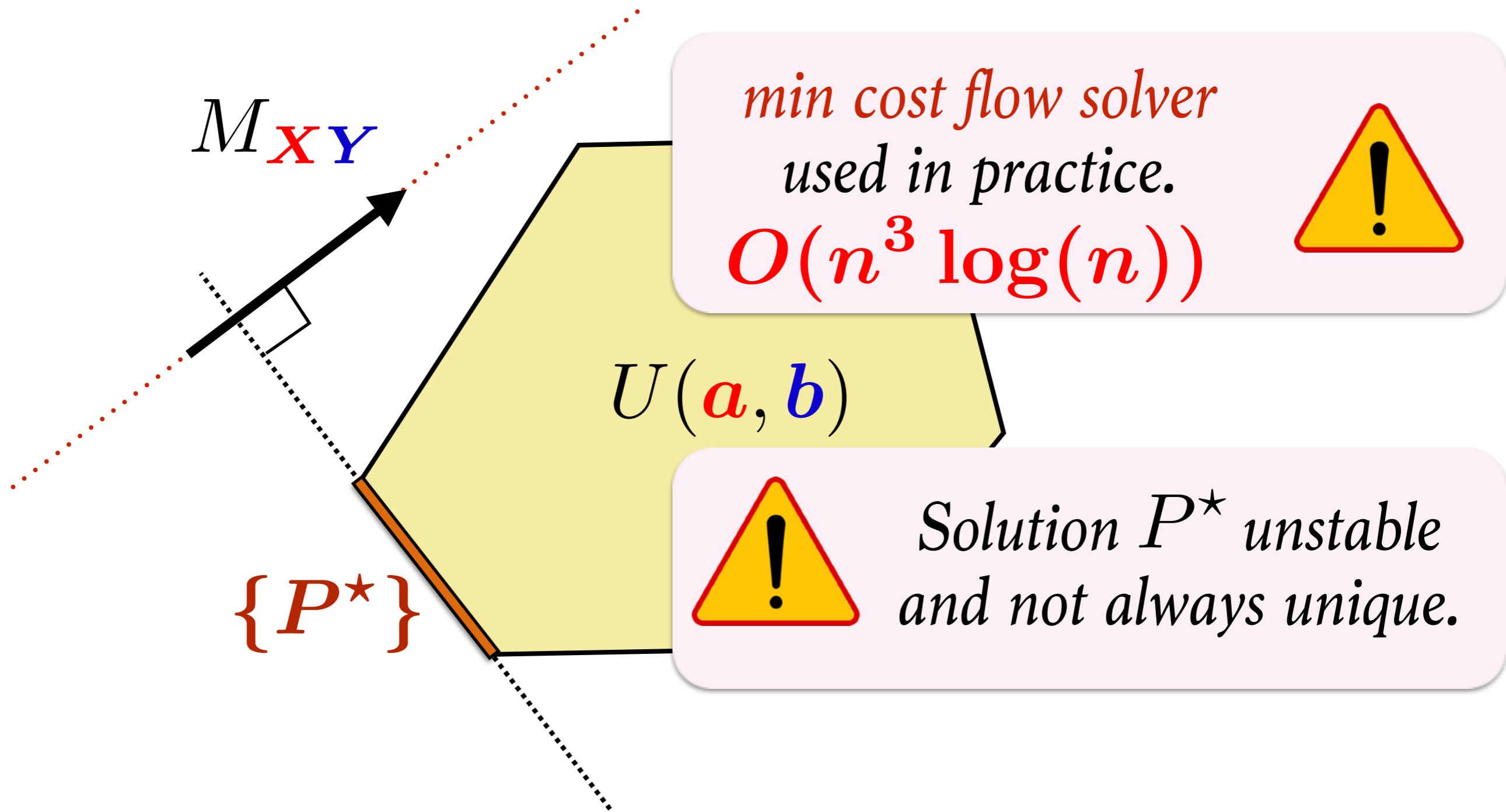
Solving the OT Problem



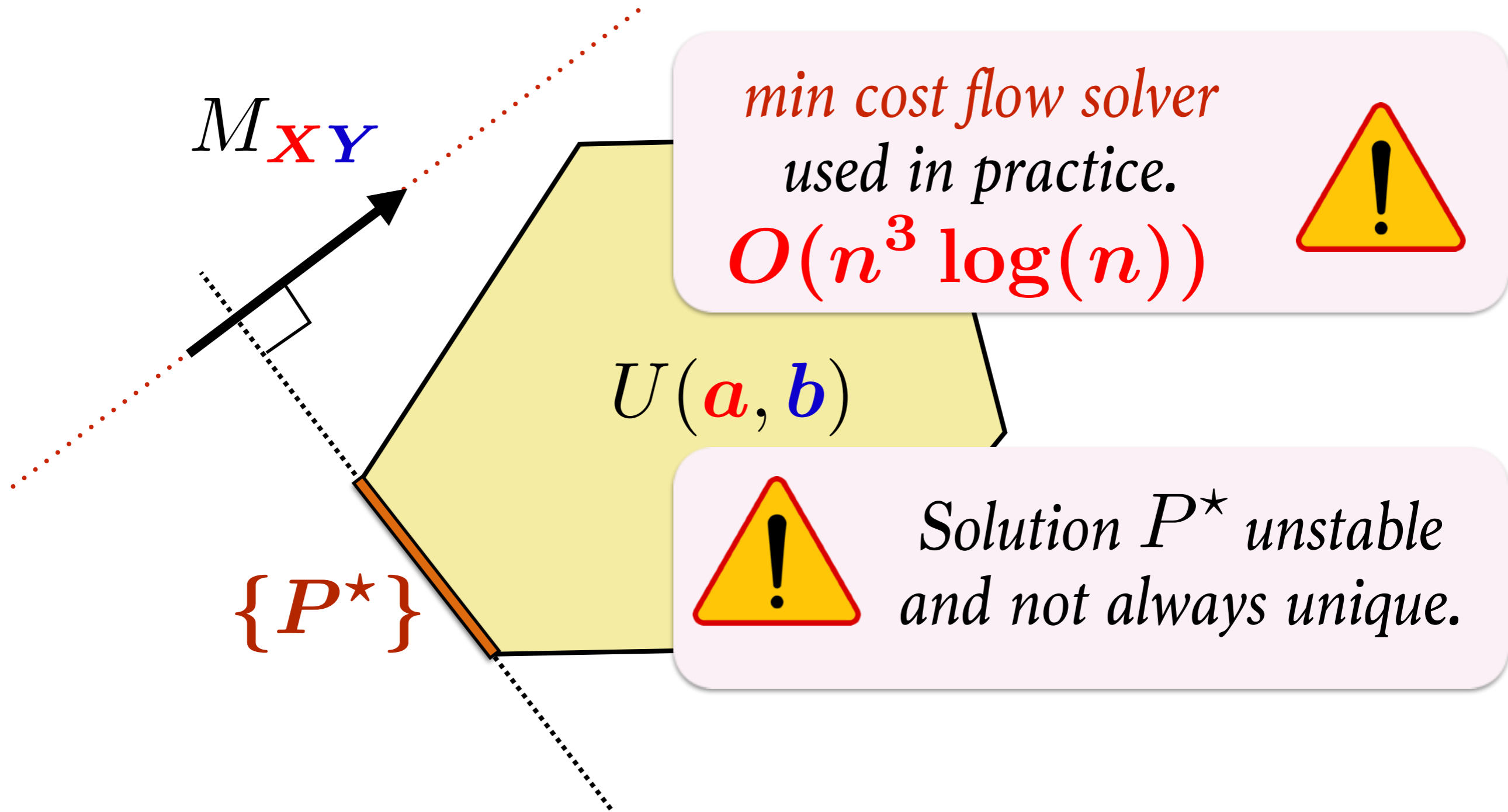
Solving the OT Problem



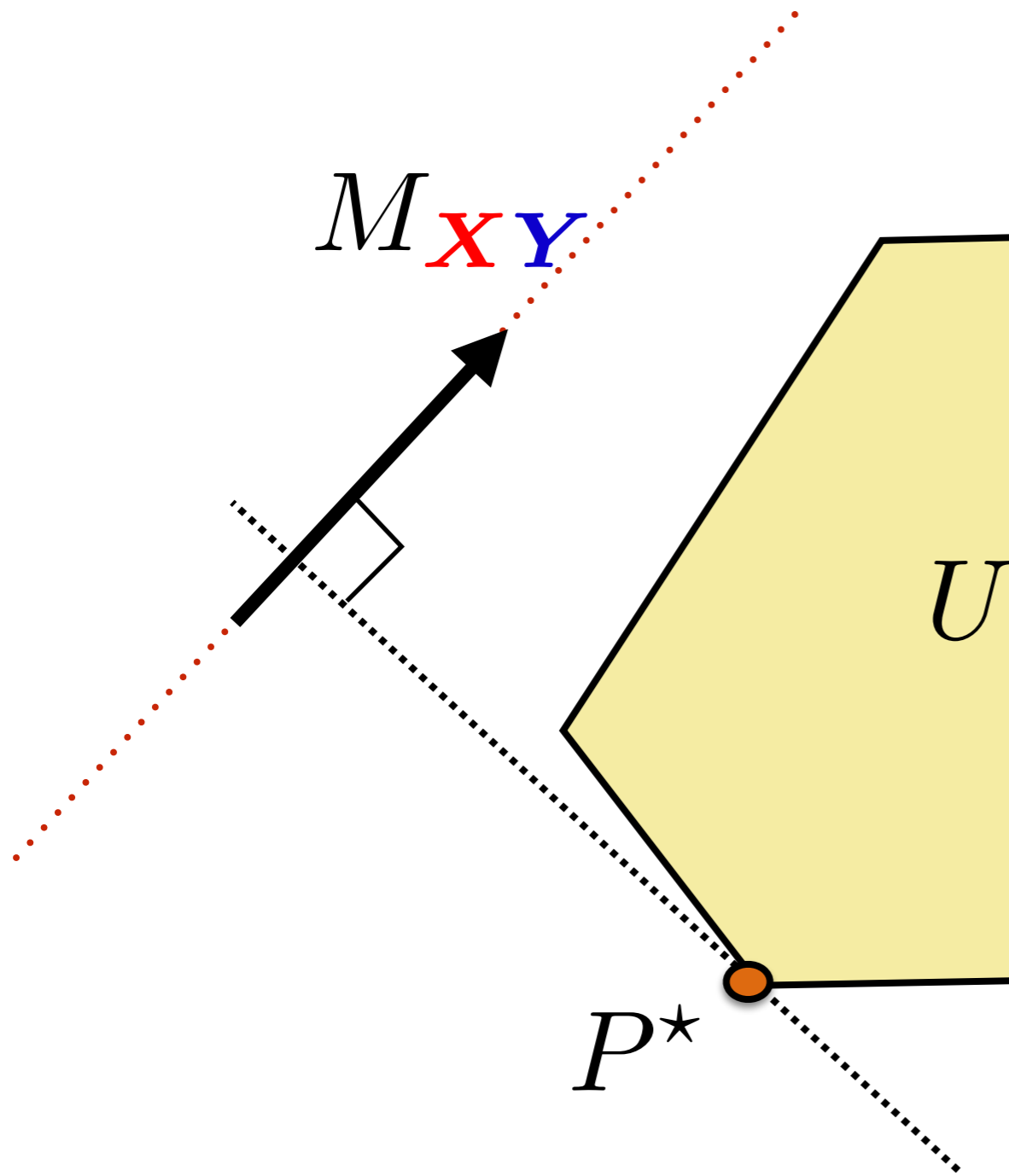
Solving the OT Problem



Solving the OT Problem



Solving the OT Problem

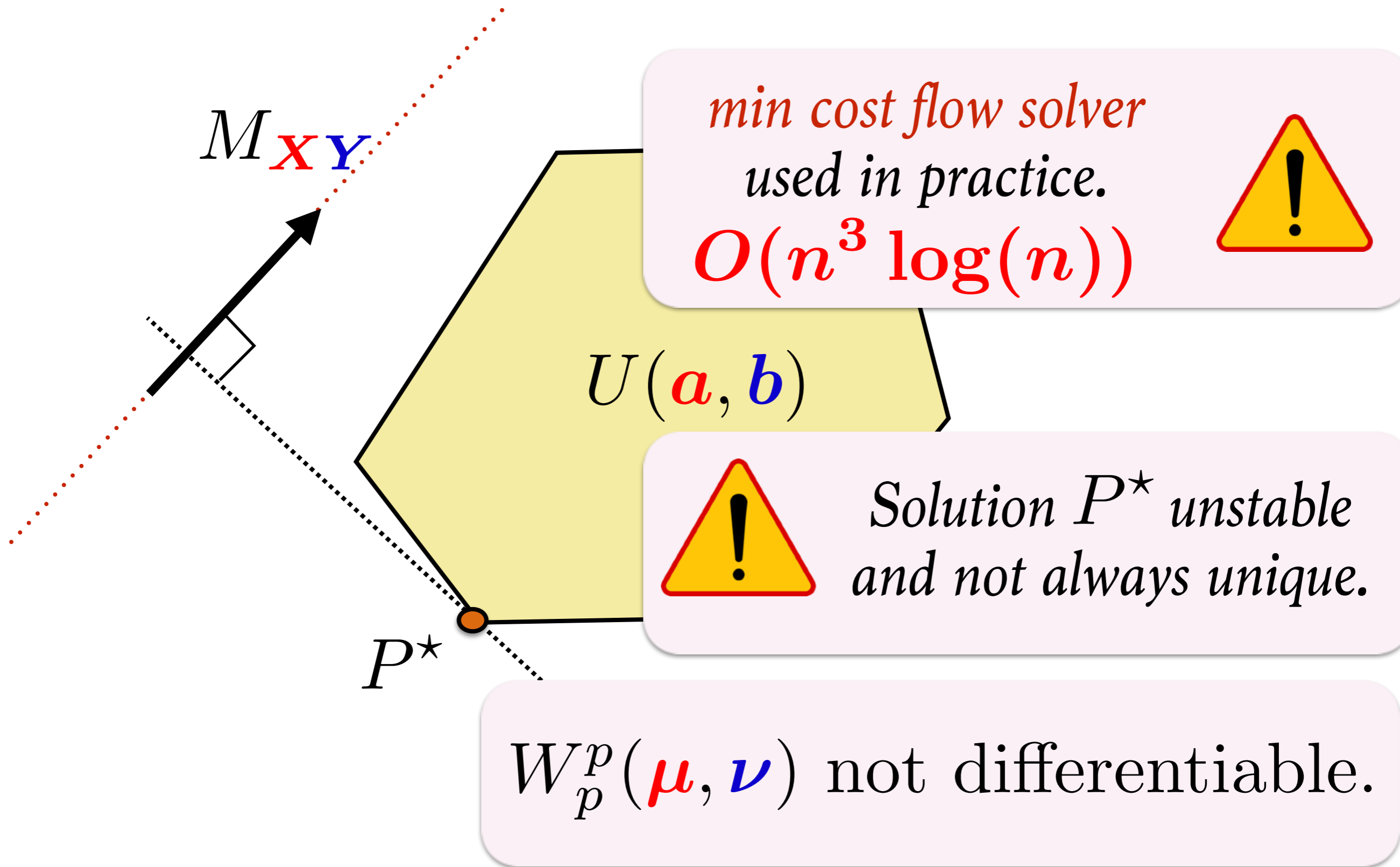


*min cost flow solver
used in practice.
 $O(n^3 \log(n))$*

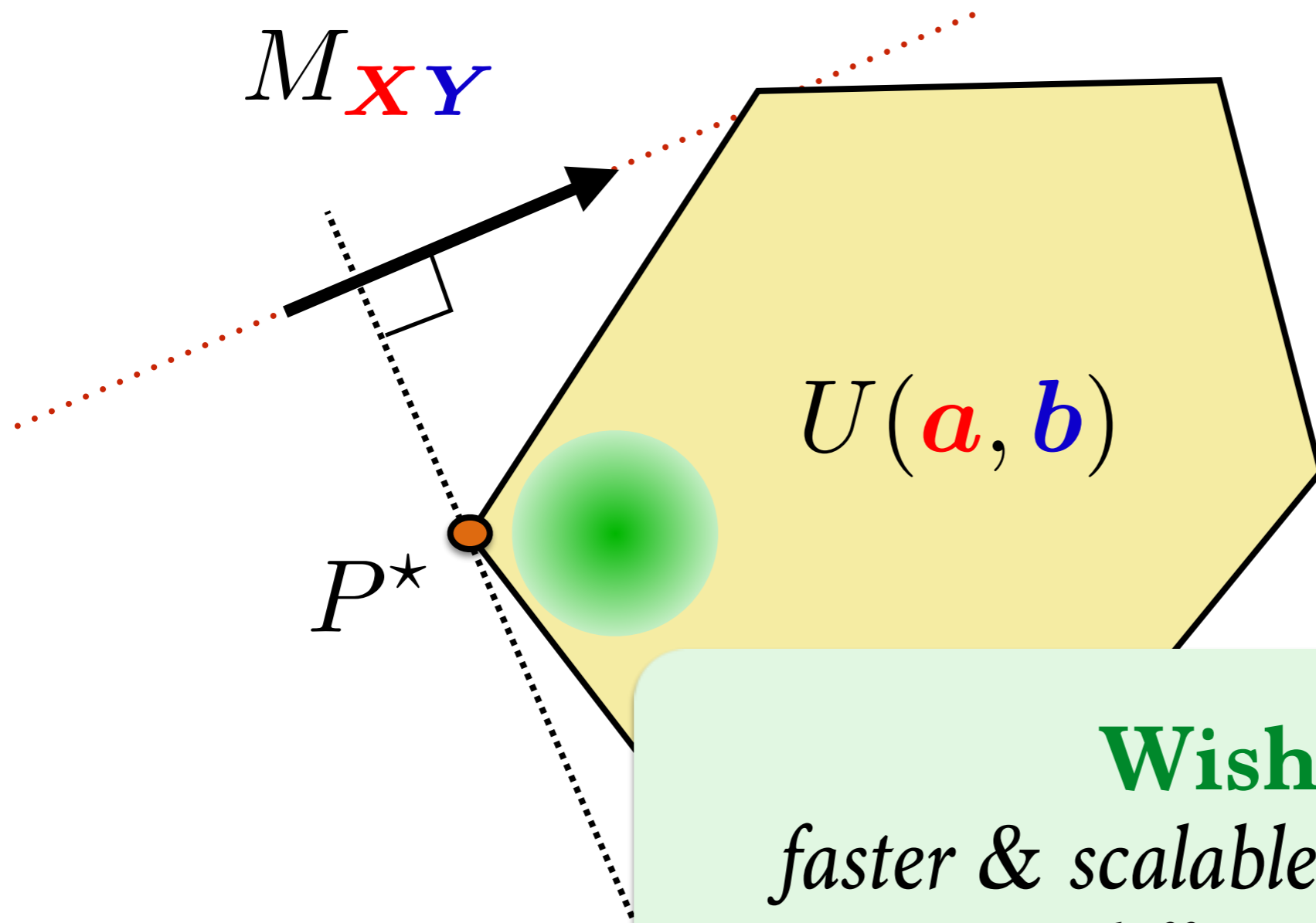


Solution P^ unstable
and not always unique.*

Solving the OT Problem



Solution: Regularization



Wishlist:
*faster & scalable, more stable,
differentiable*

Entropic Regularization [Wilson'62]

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

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

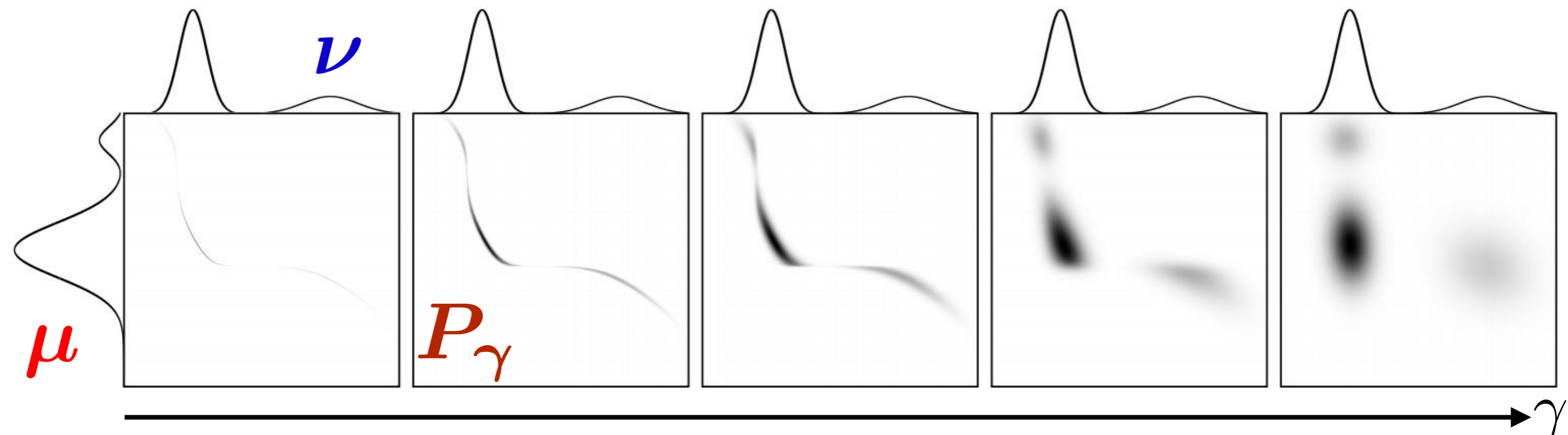
$$E(P) \stackrel{\text{def}}{=} - \sum_{i,j=1}^{nm} P_{ij} (\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

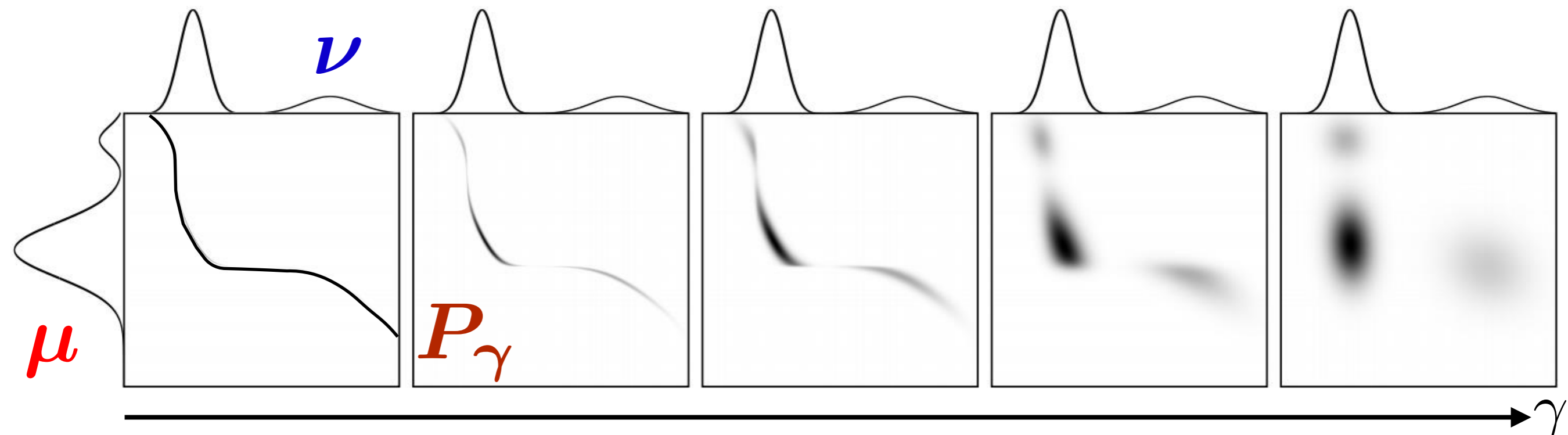


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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



Note: Unique optimal solution because of strong concavity of entropy

Fast & Scalable Algorithm

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

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

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

Fast & Scalable Algorithm

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

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

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

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

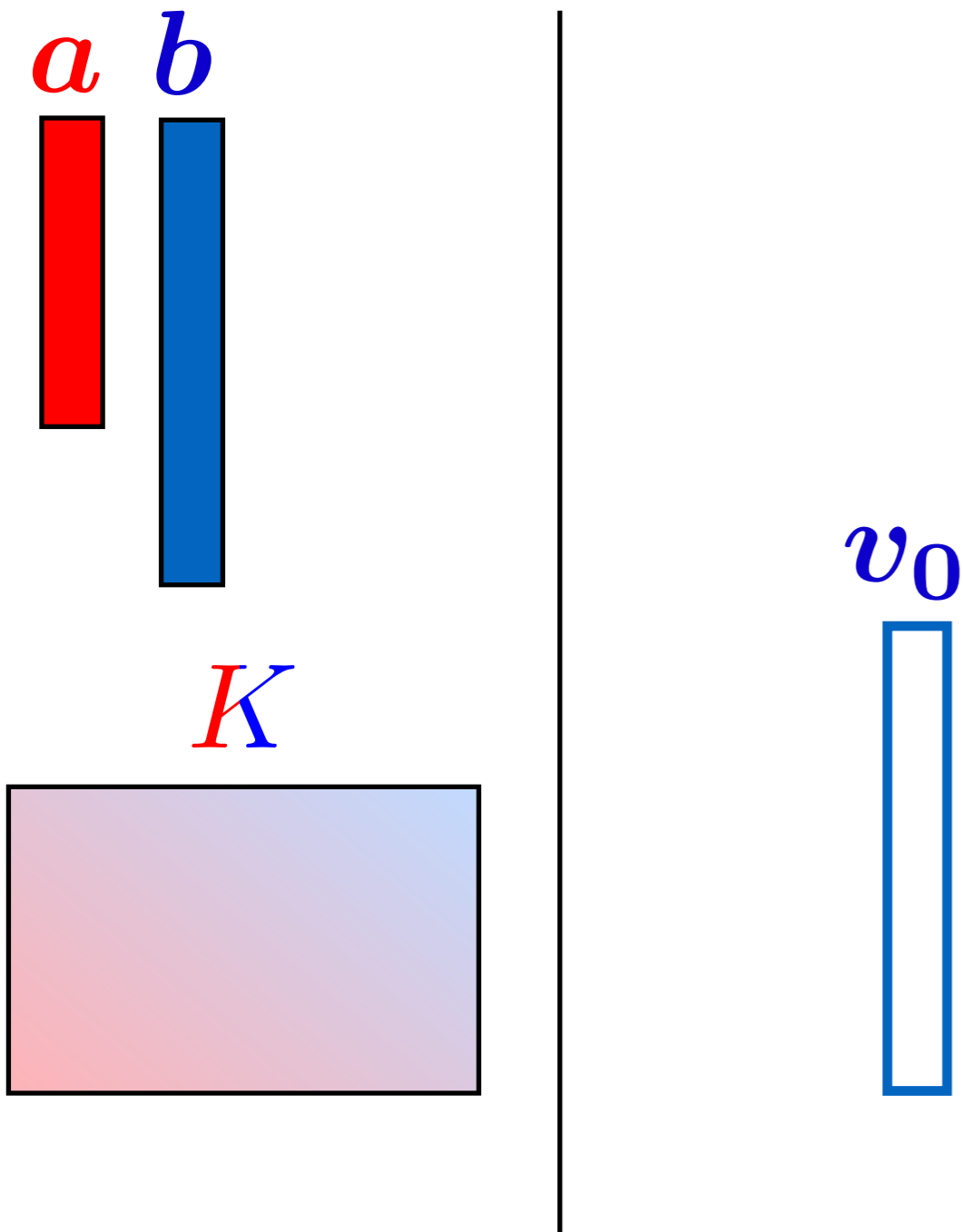
$$1. \quad \mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v}$$

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

- [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

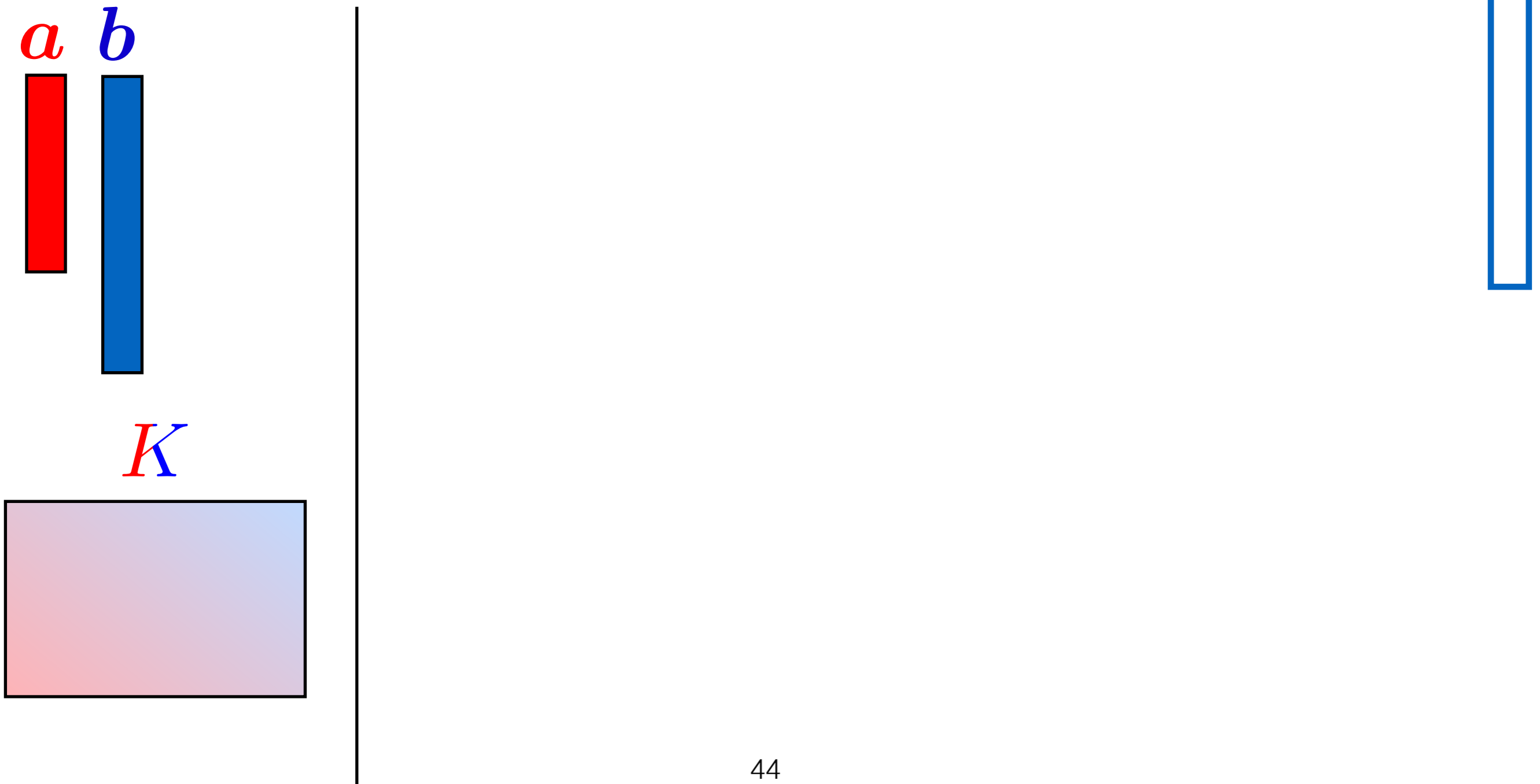
- [Sinkhorn'64] fixed-point iterations for (\mathbf{u}, \mathbf{v})
$$\mathbf{u} \leftarrow \mathbf{a} / K \mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b} / K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

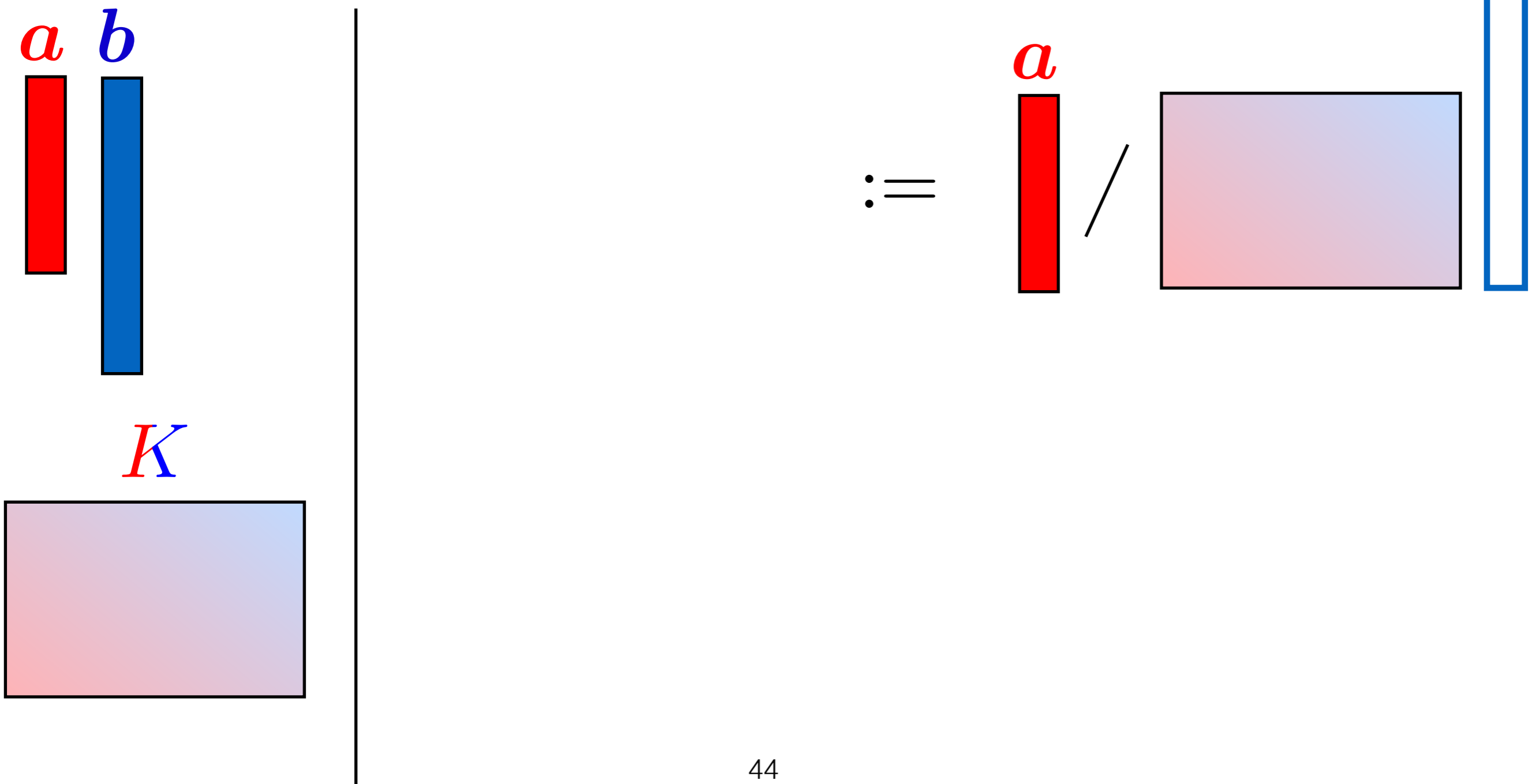
$$\mathbf{u} \leftarrow \mathbf{a} / K \mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b} / K^T \mathbf{u}$$



Fast & Scalable Algorithm

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

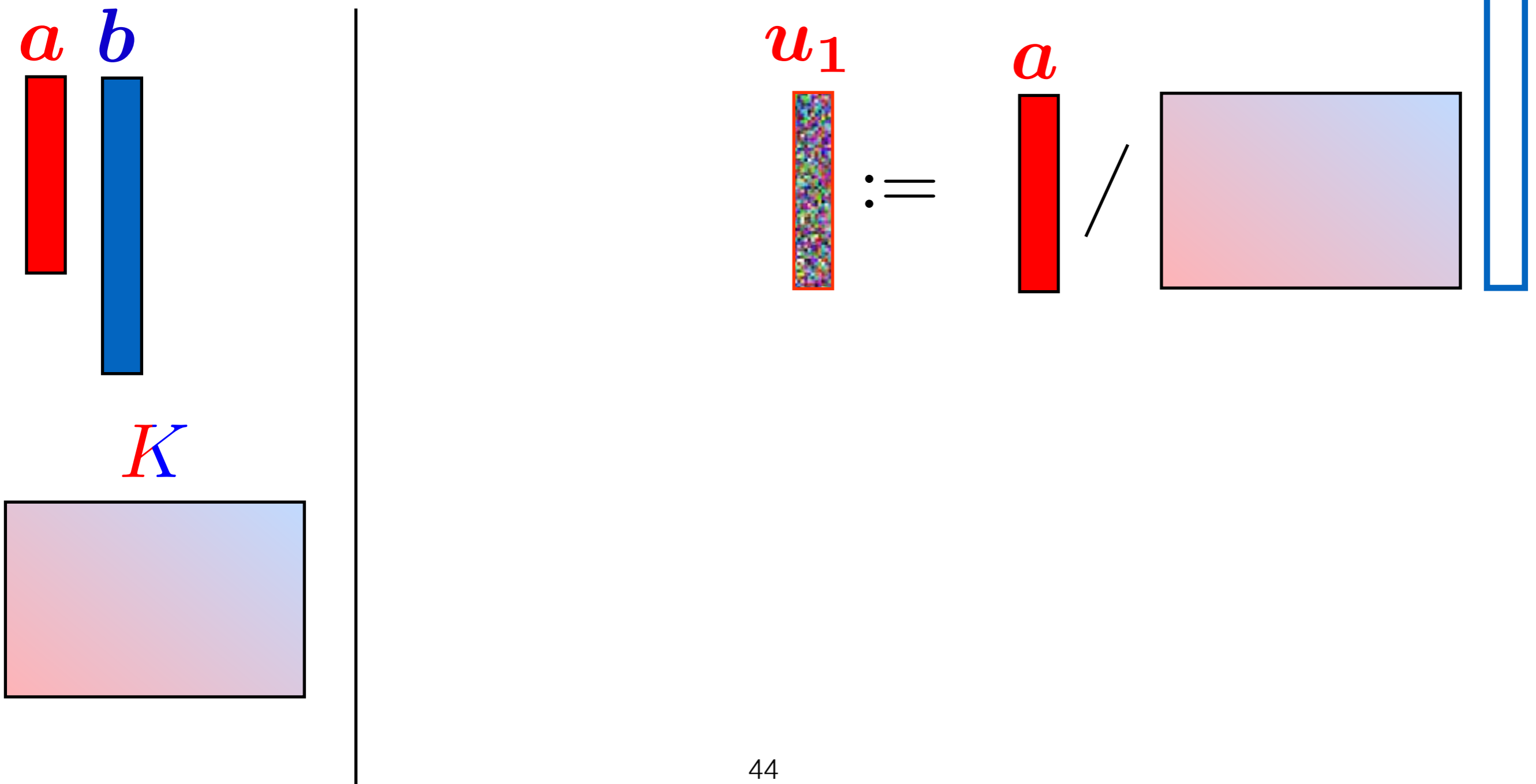
$$\mathbf{u} \leftarrow \mathbf{a} / K \mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b} / K^T \mathbf{u}$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

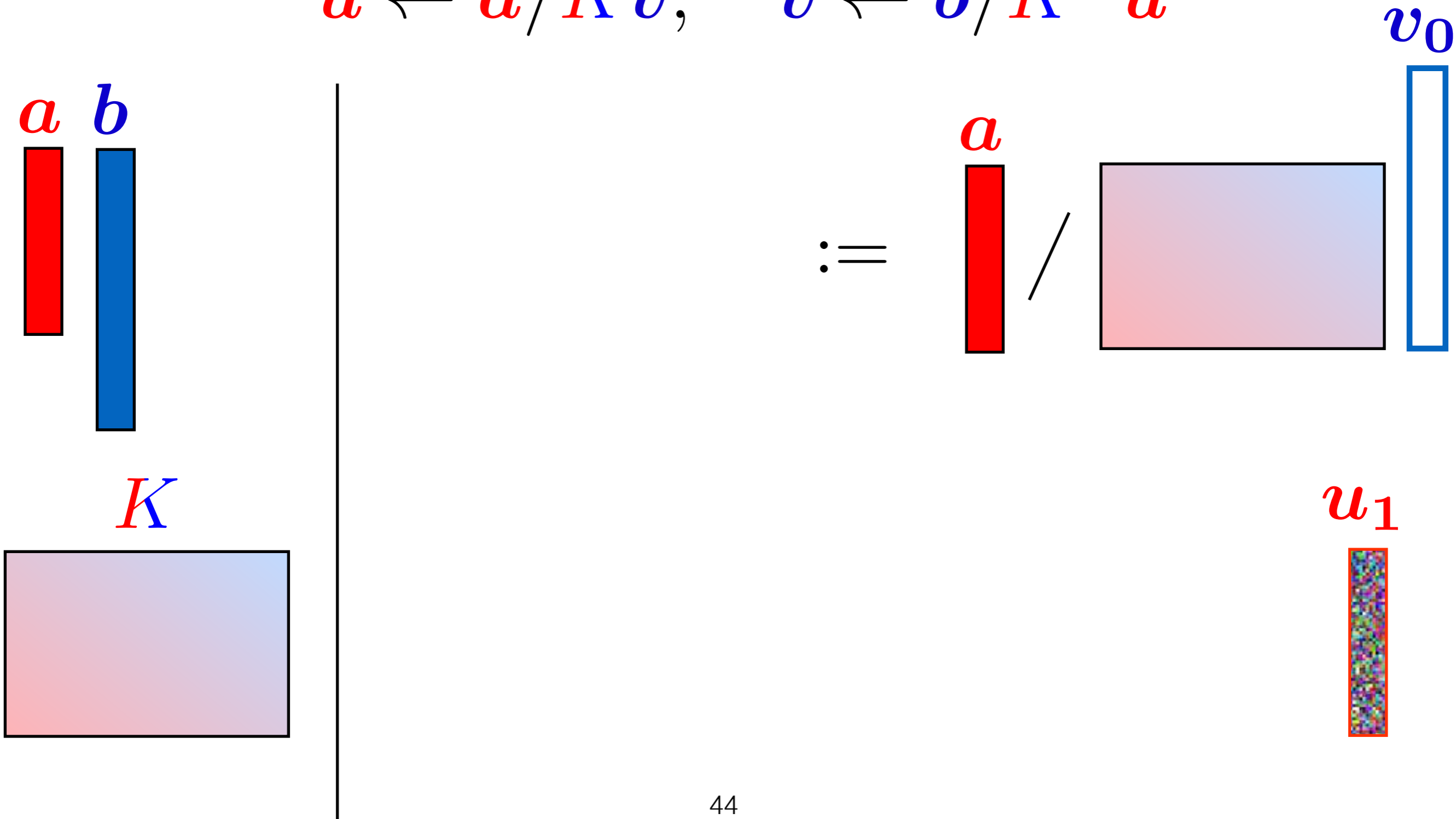
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

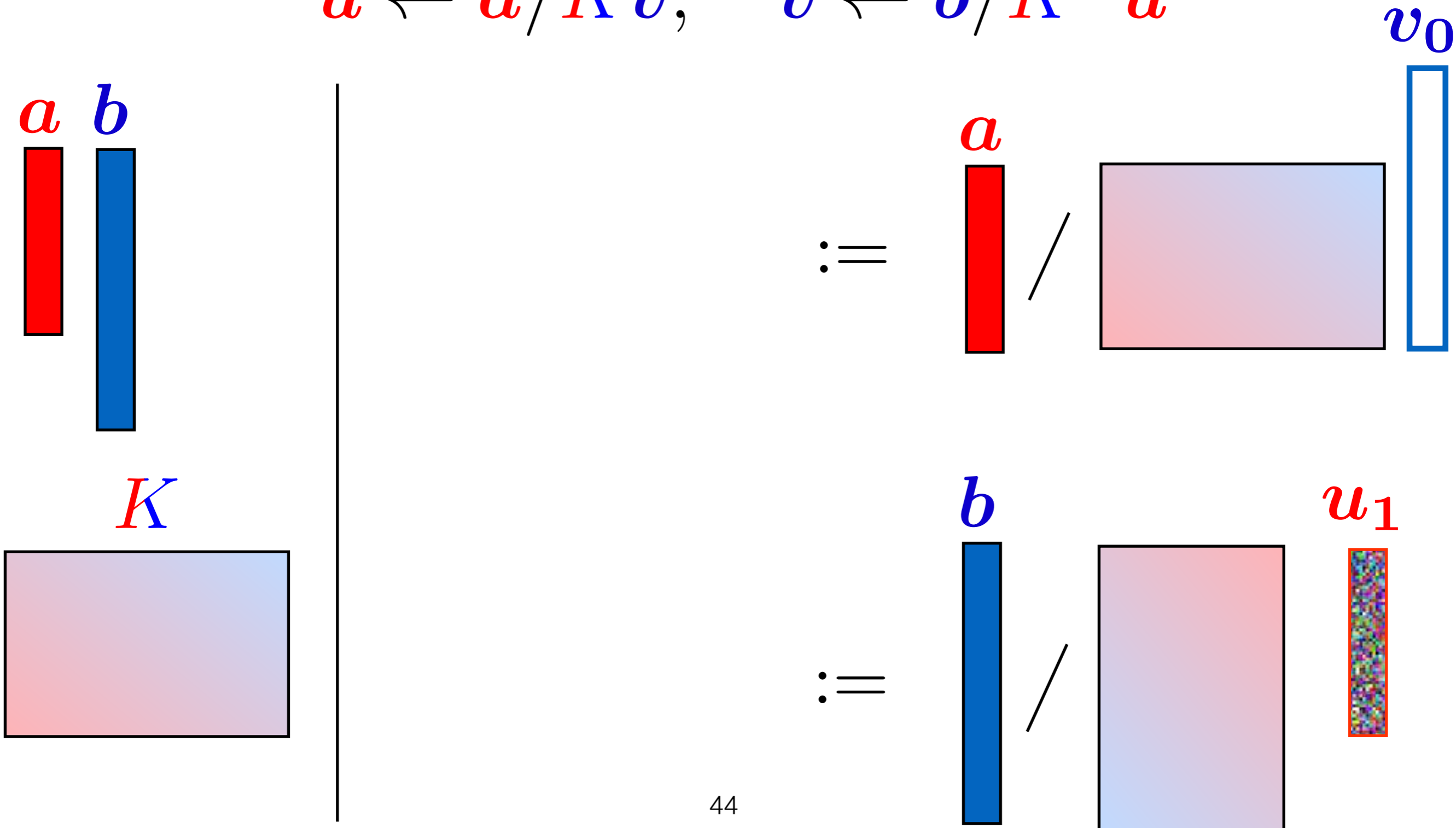
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

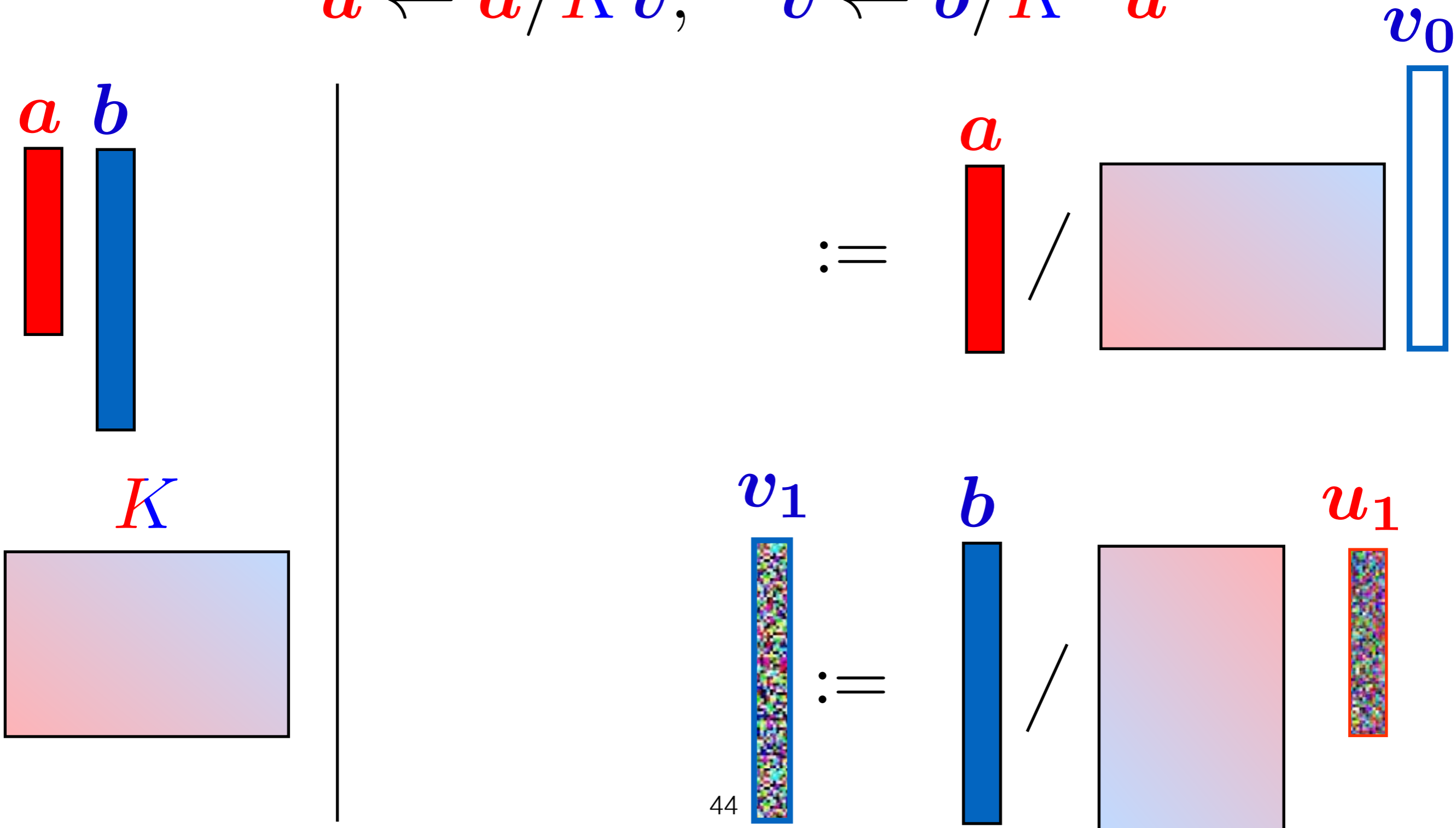
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

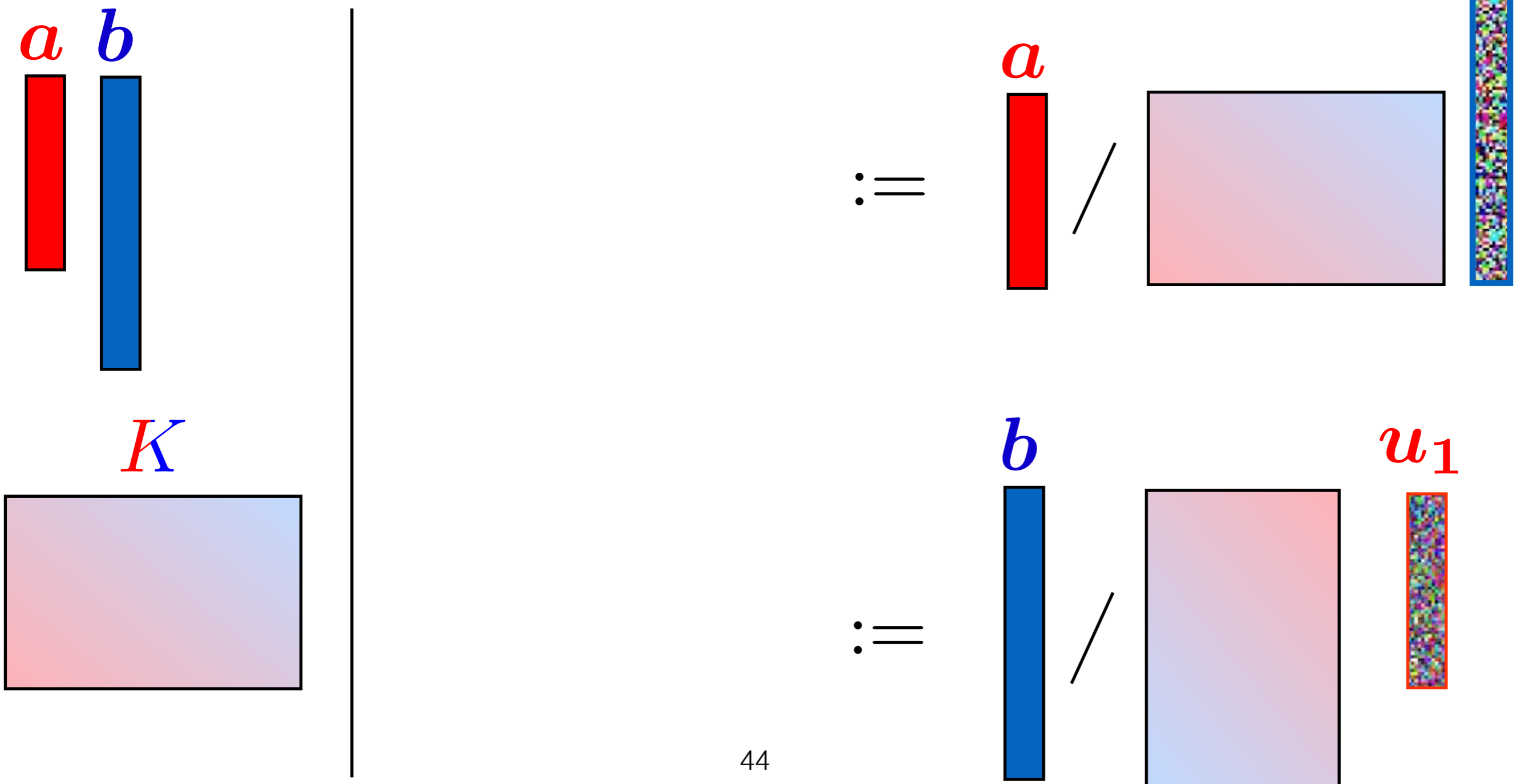
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

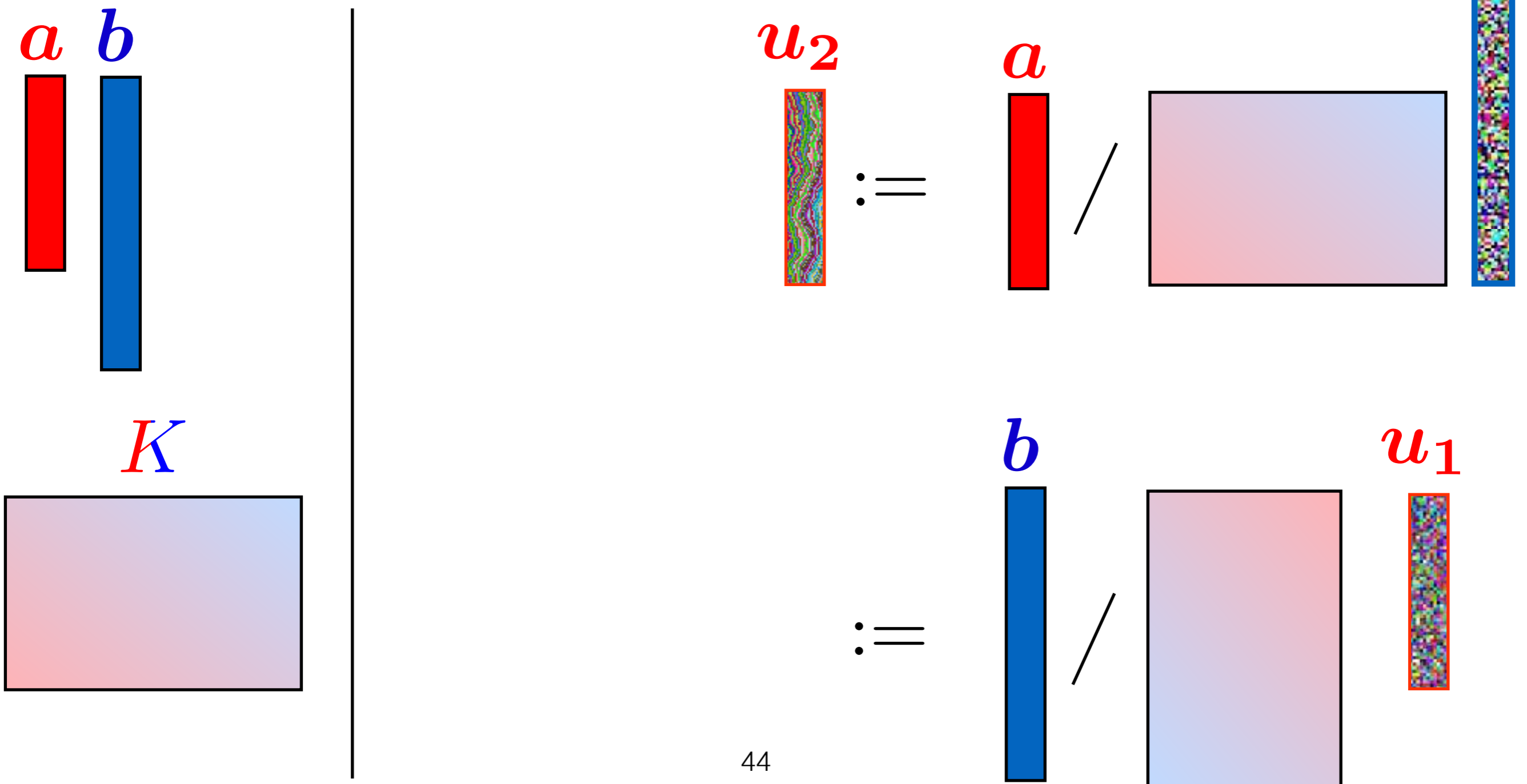
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

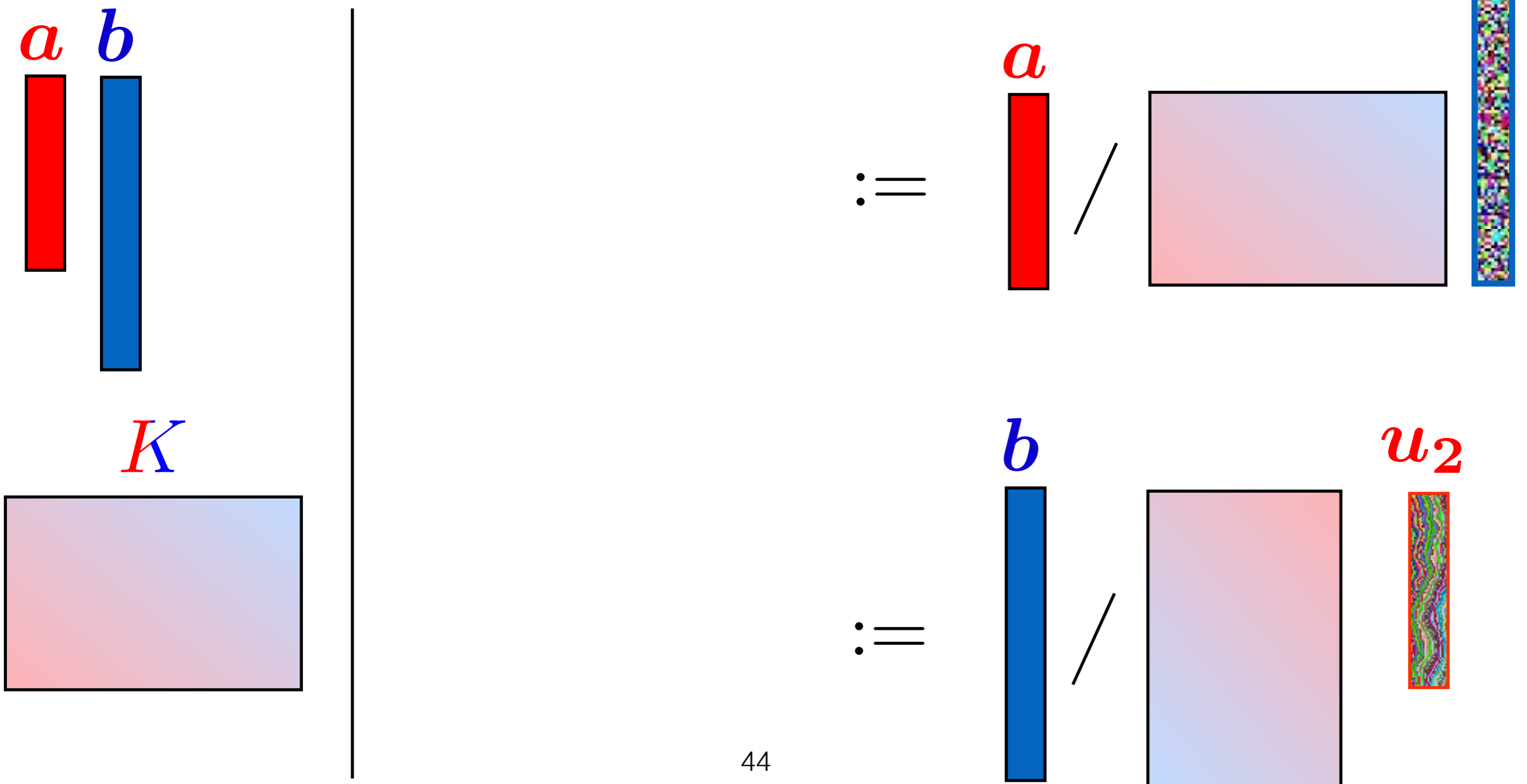
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

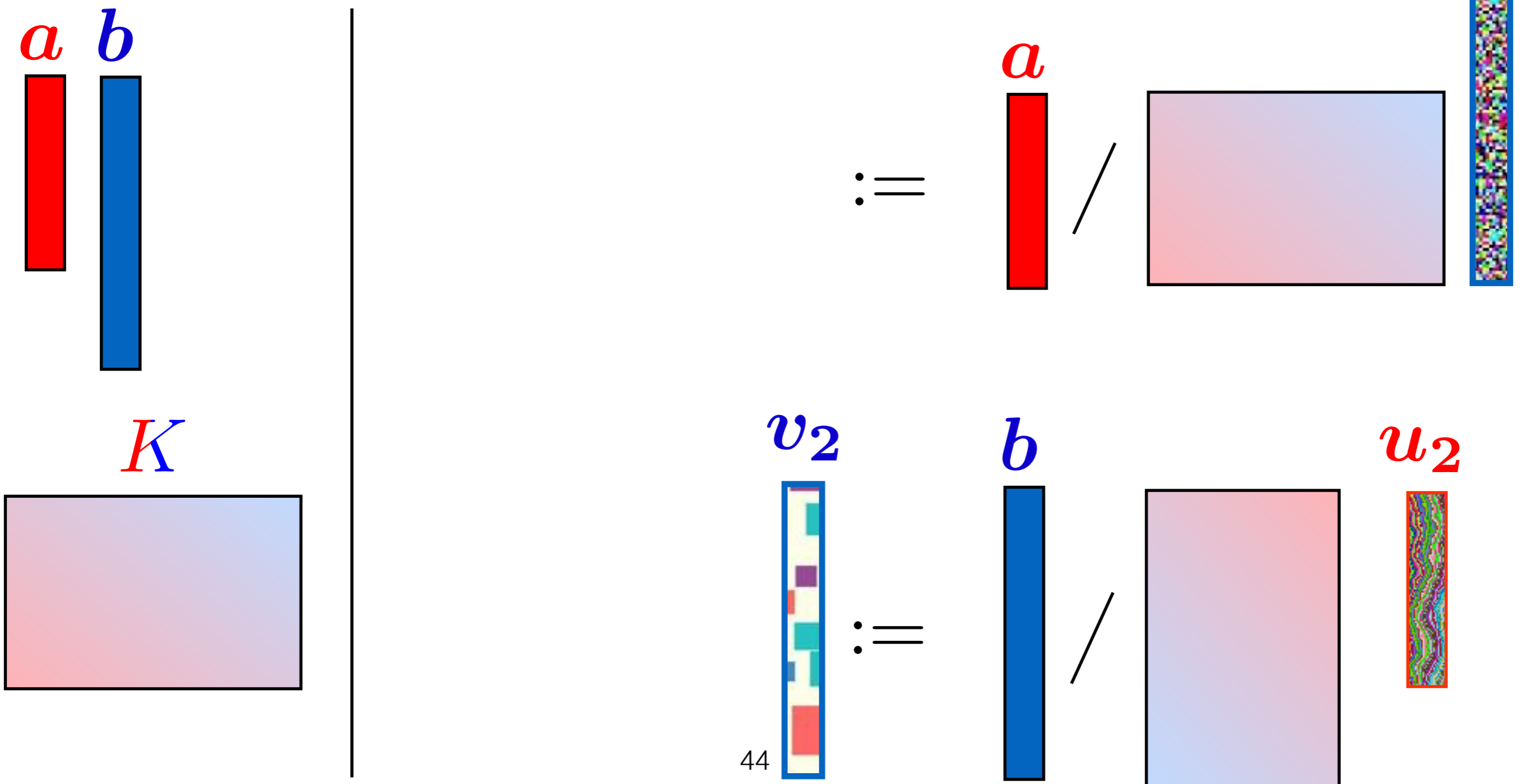
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

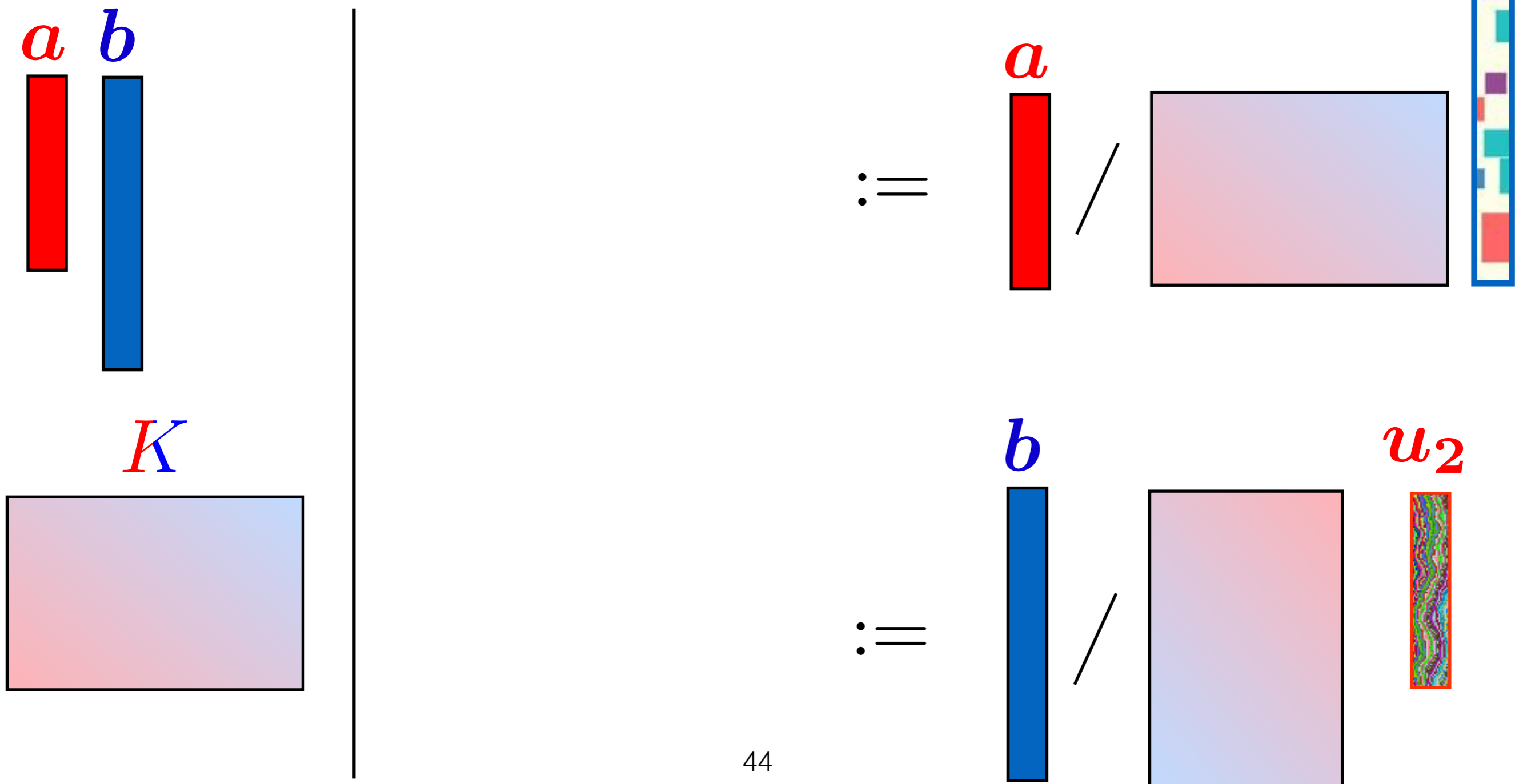
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

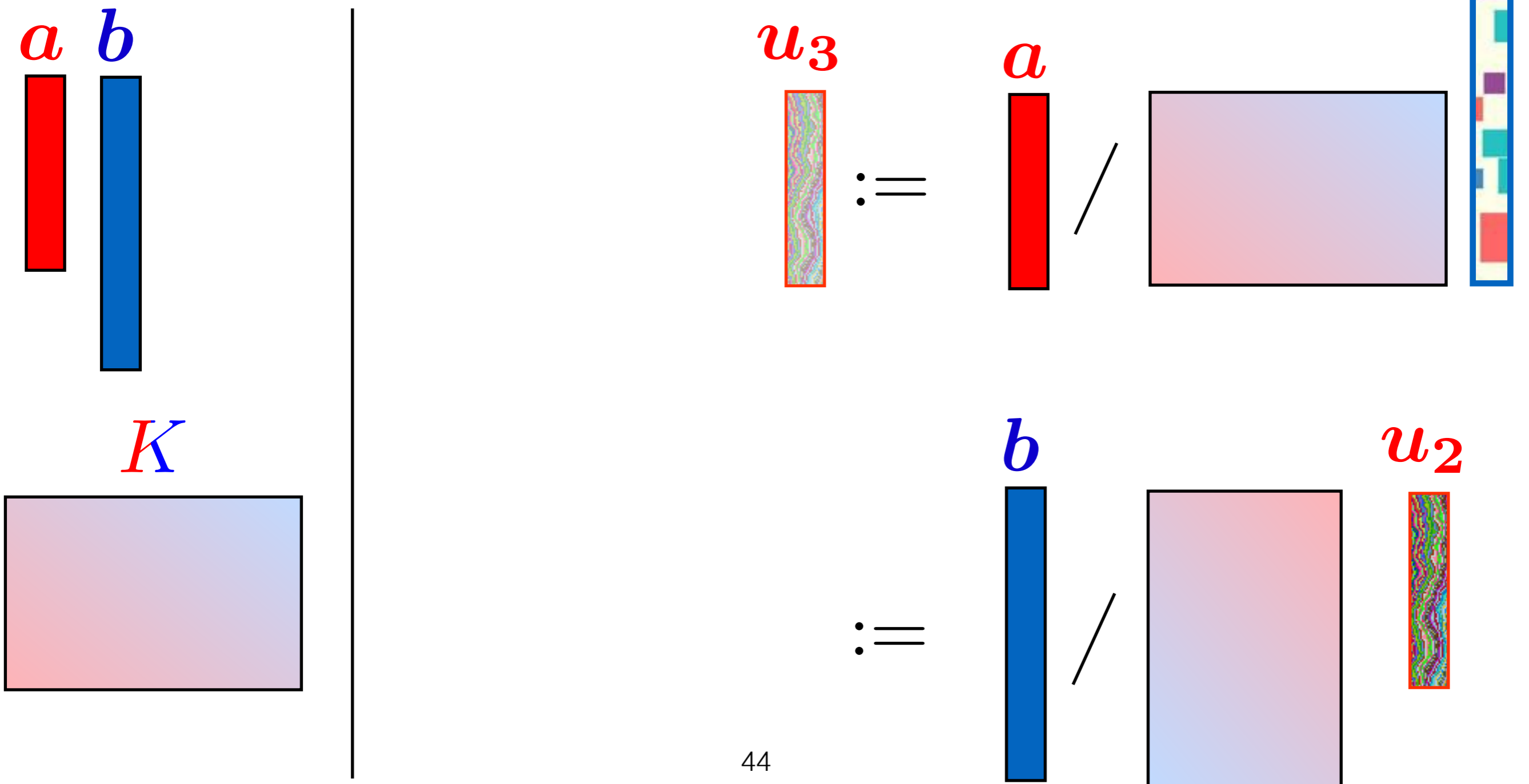
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

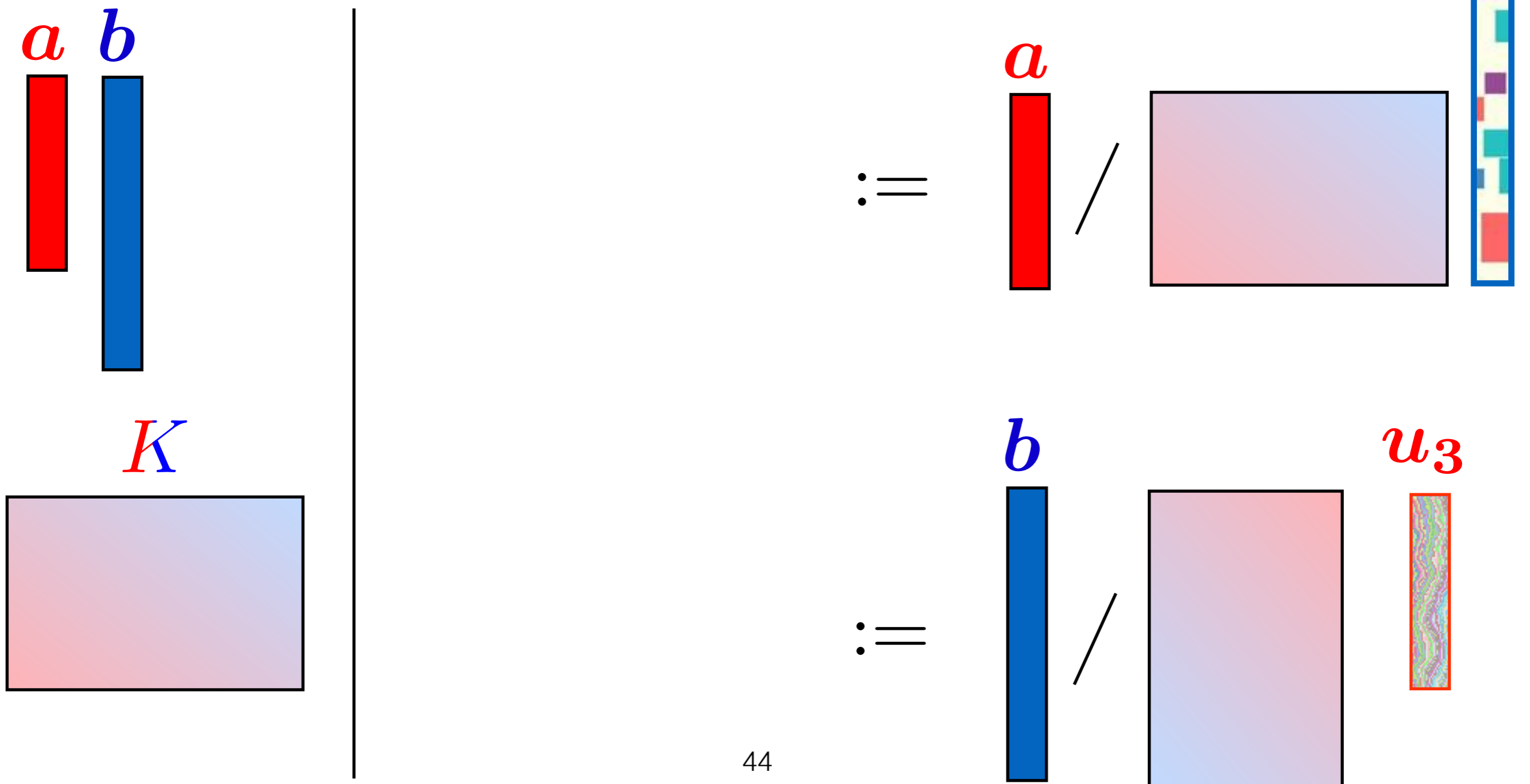
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for (u, v)

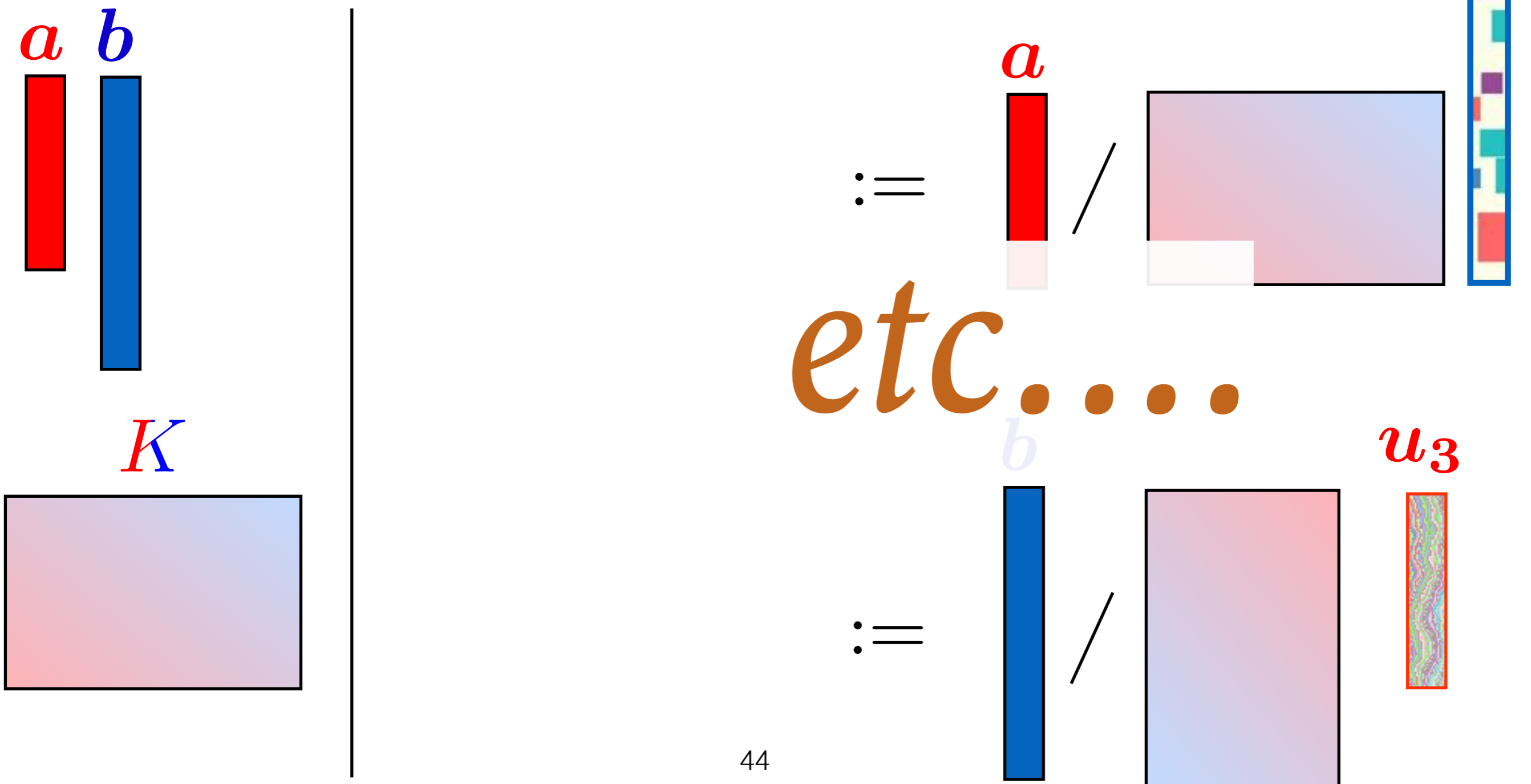
$$u \leftarrow a / K v, \quad v \leftarrow b / K^T u$$



Fast & Scalable Algorithm

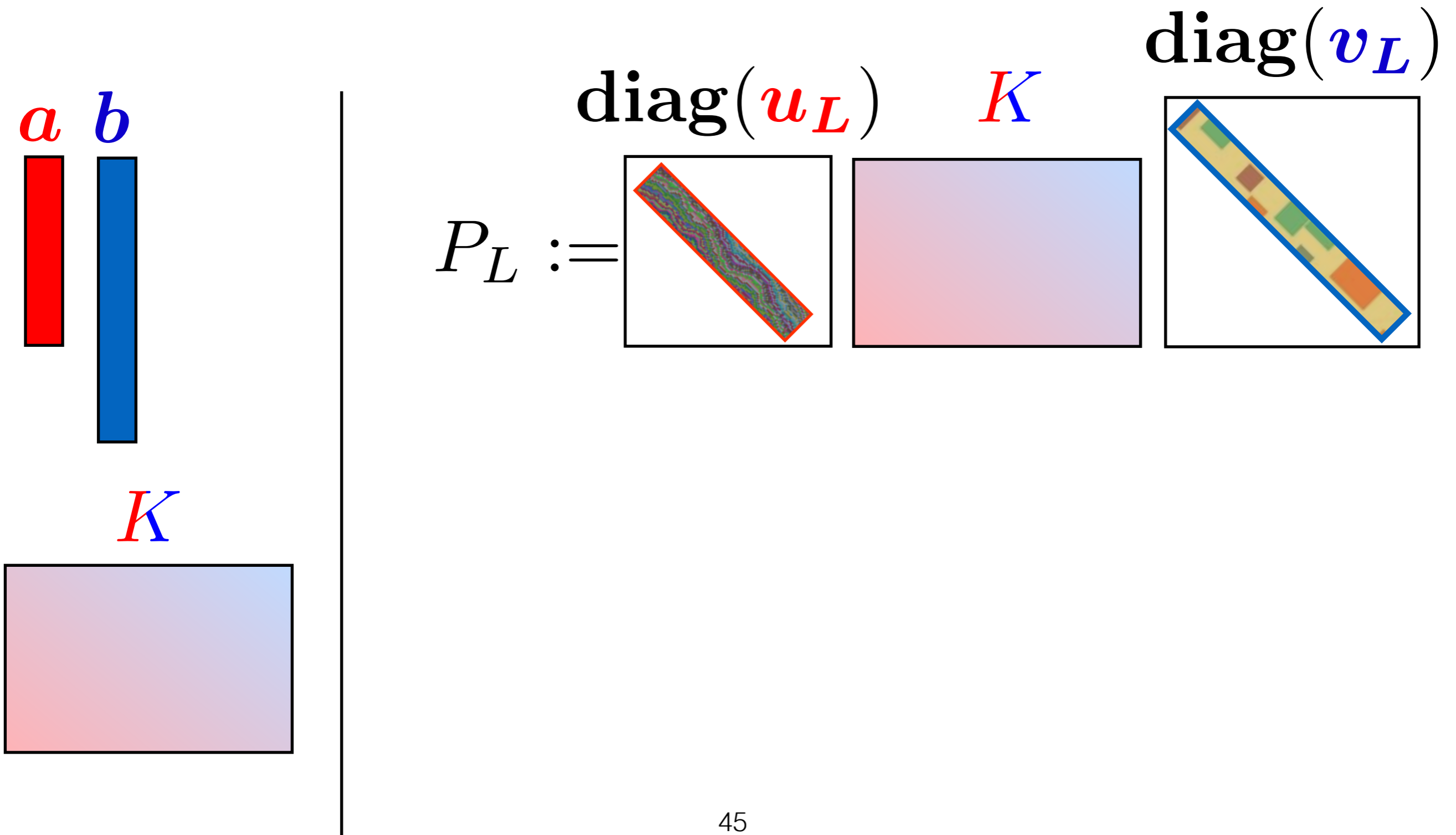
- [Sinkhorn'64] fixed-point iterations for (u, v)

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



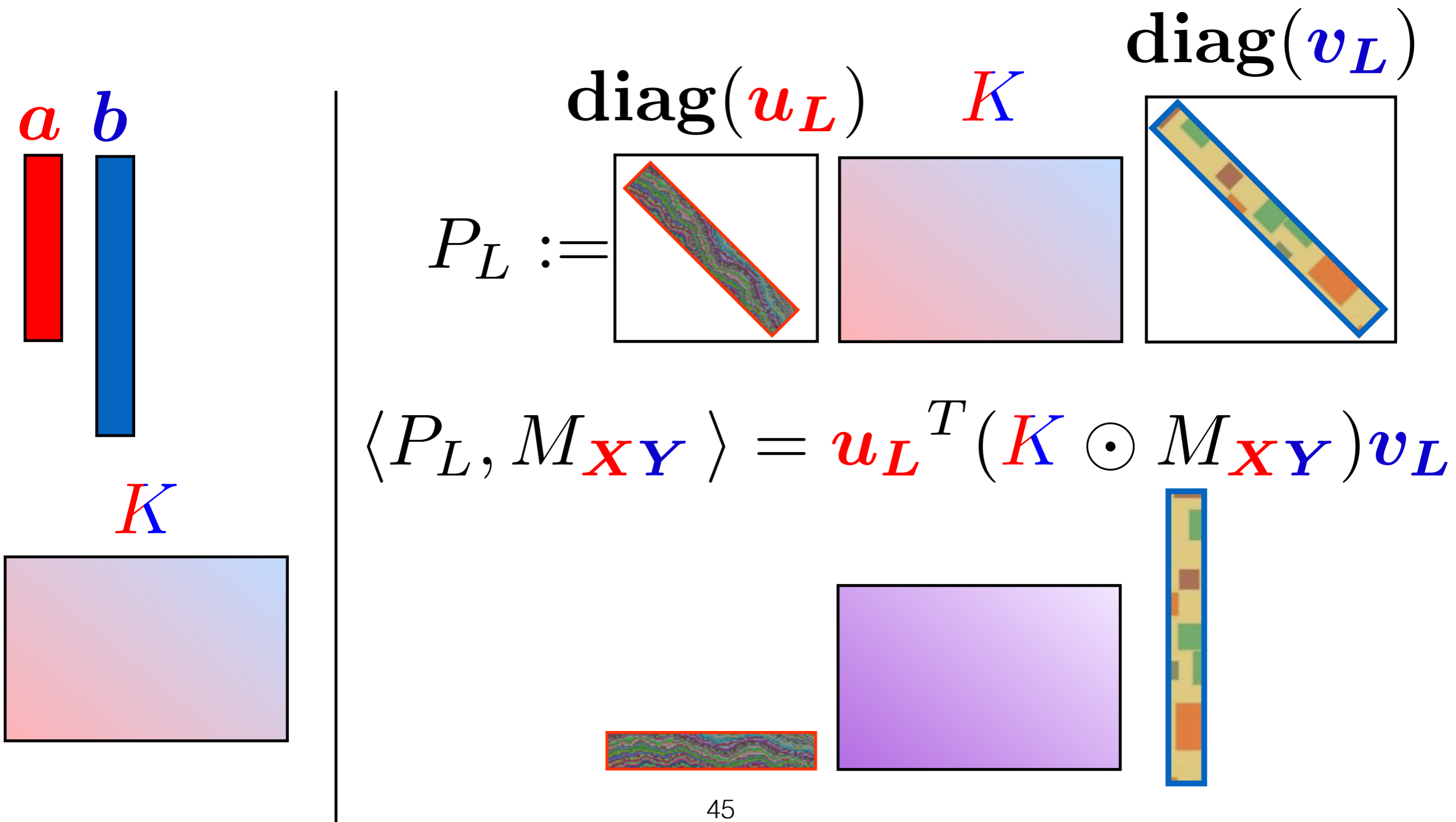
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



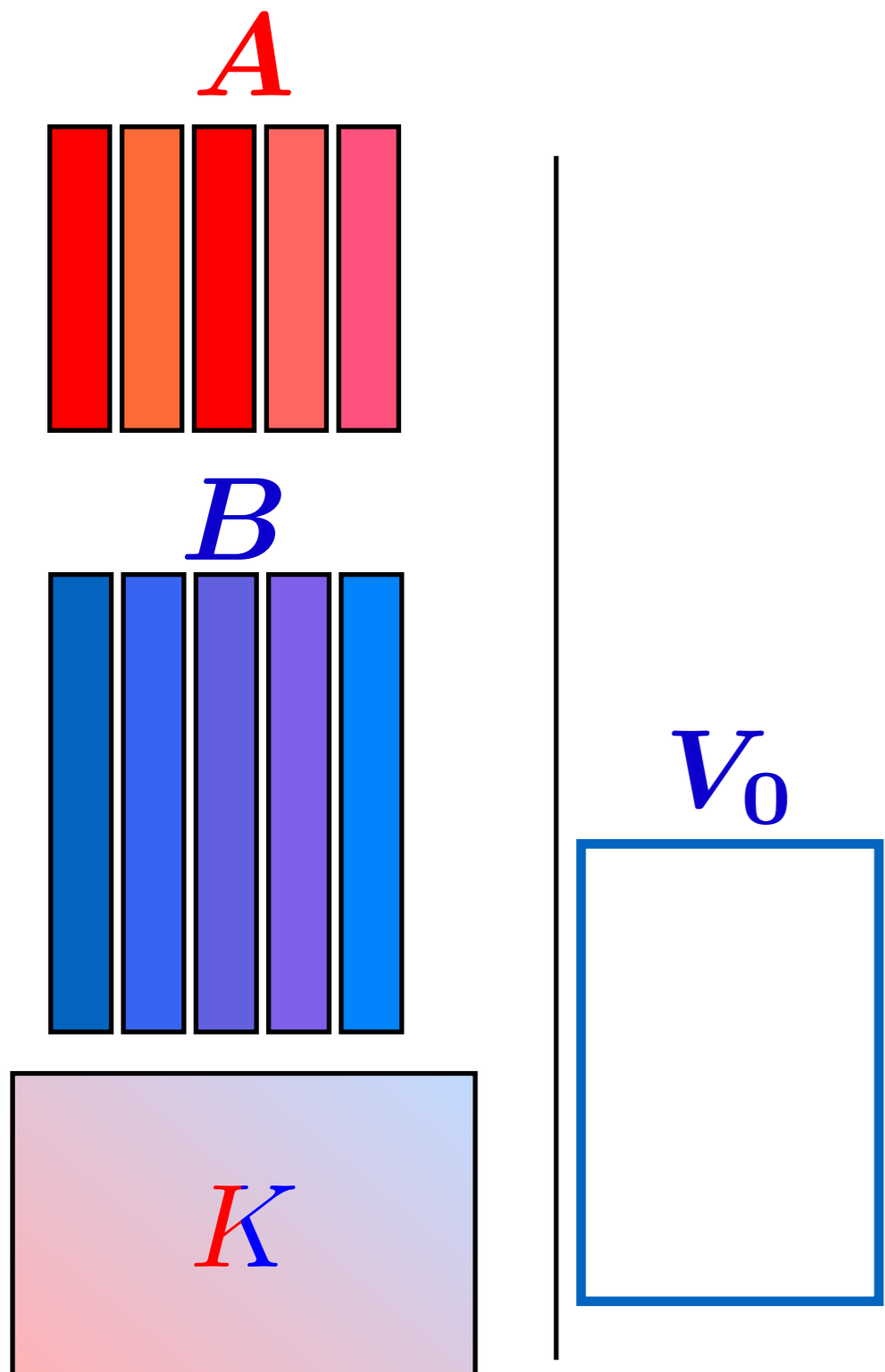
Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



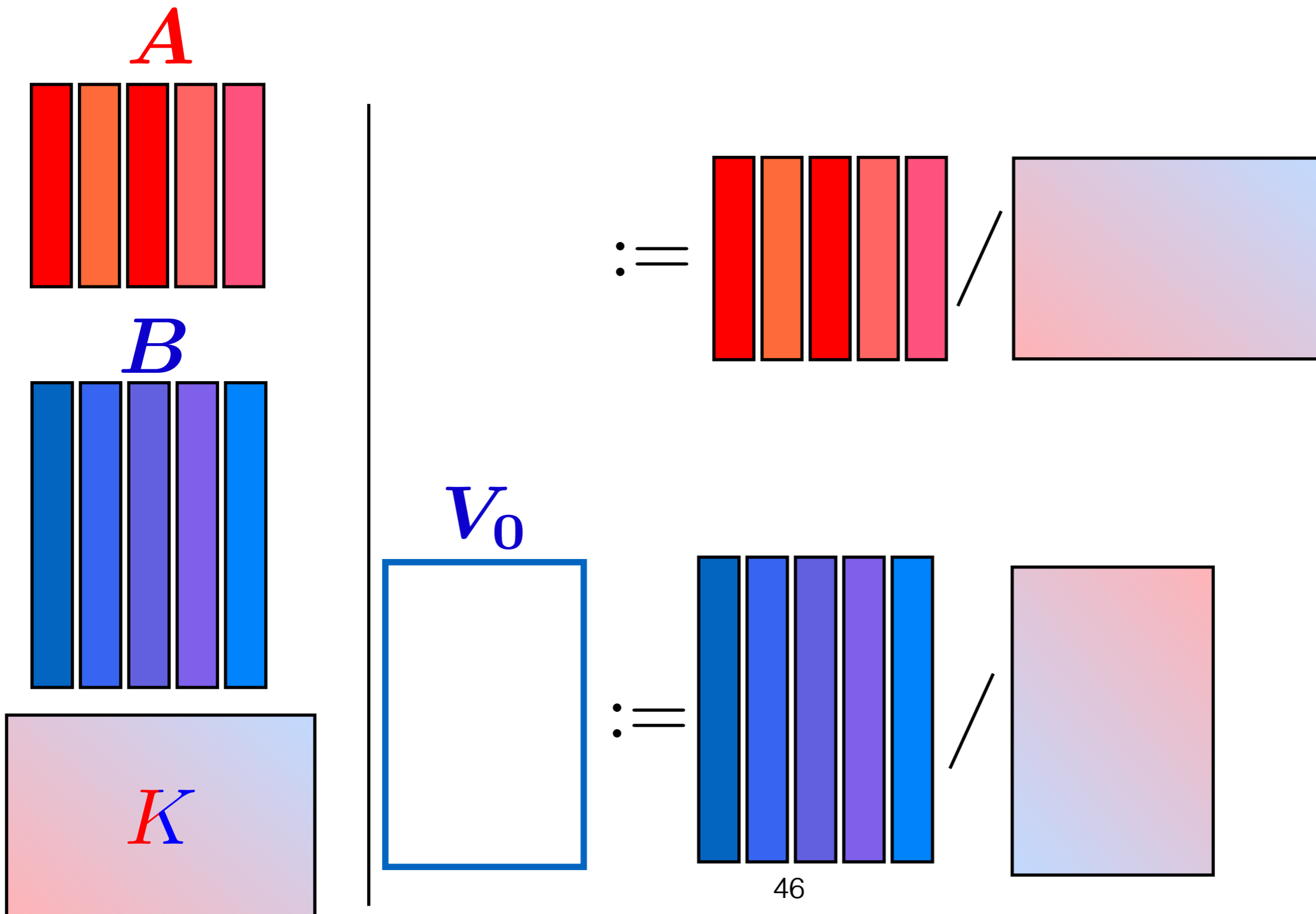
Also embarrassingly parallel

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



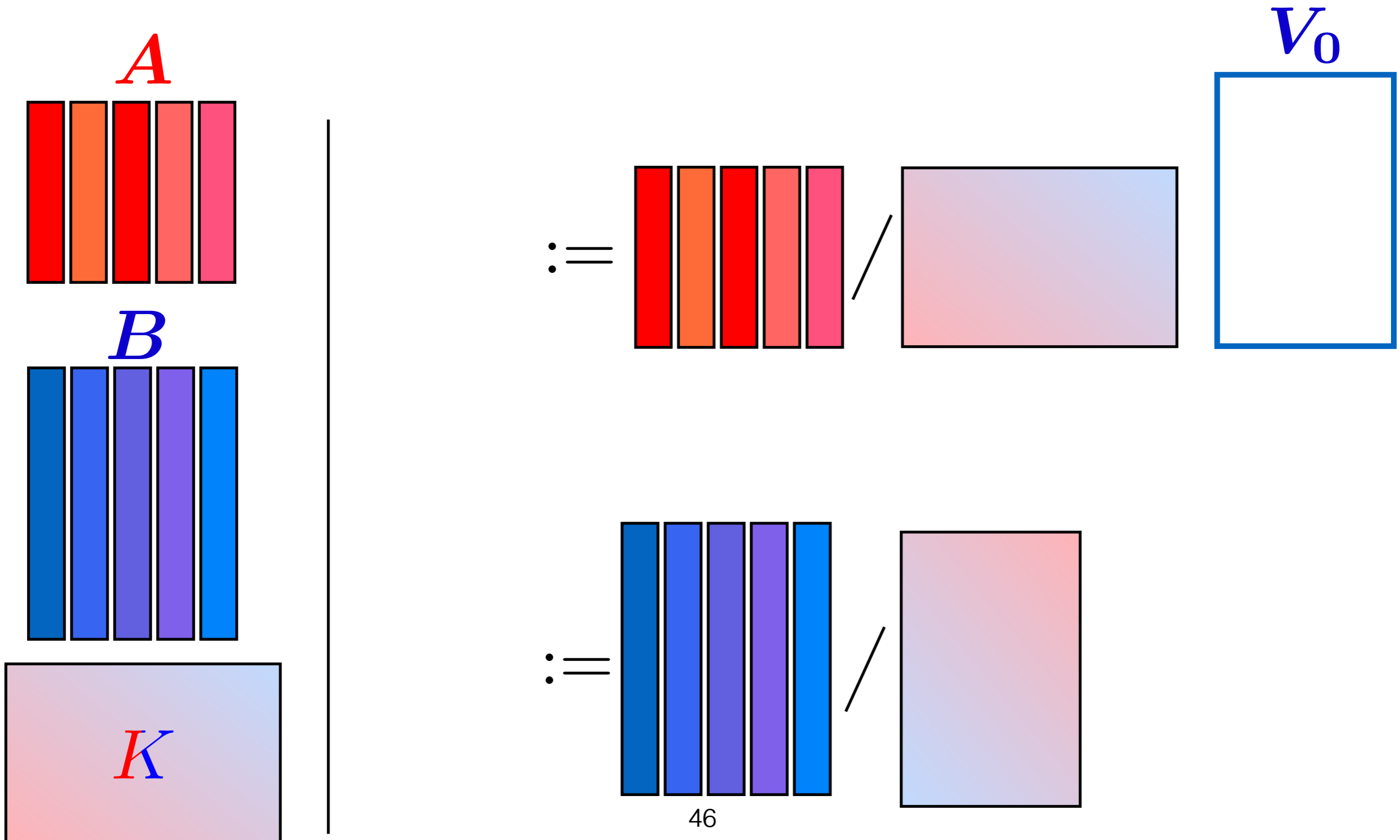
Also embarrassingly parallel

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



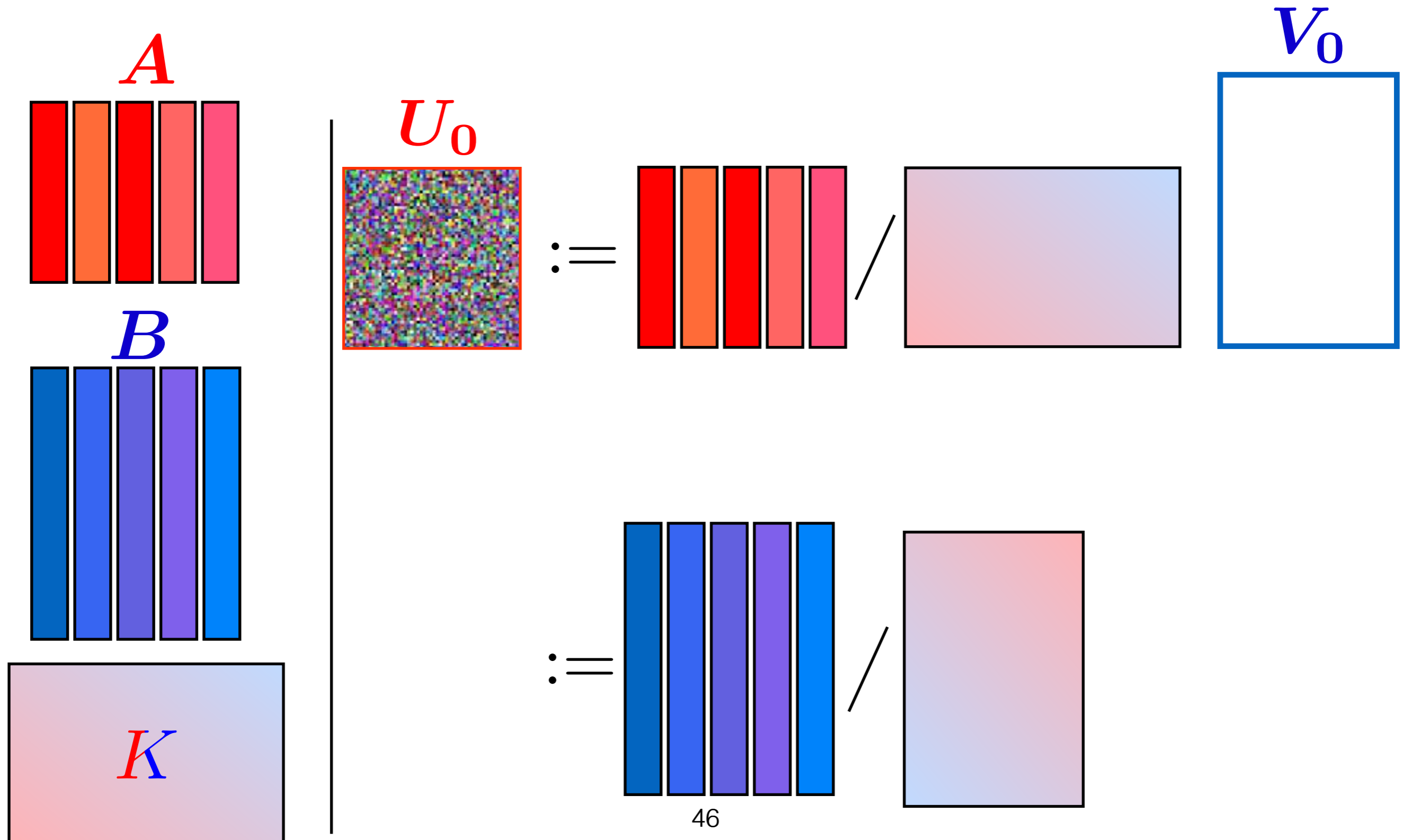
Also embarrassingly parallel

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



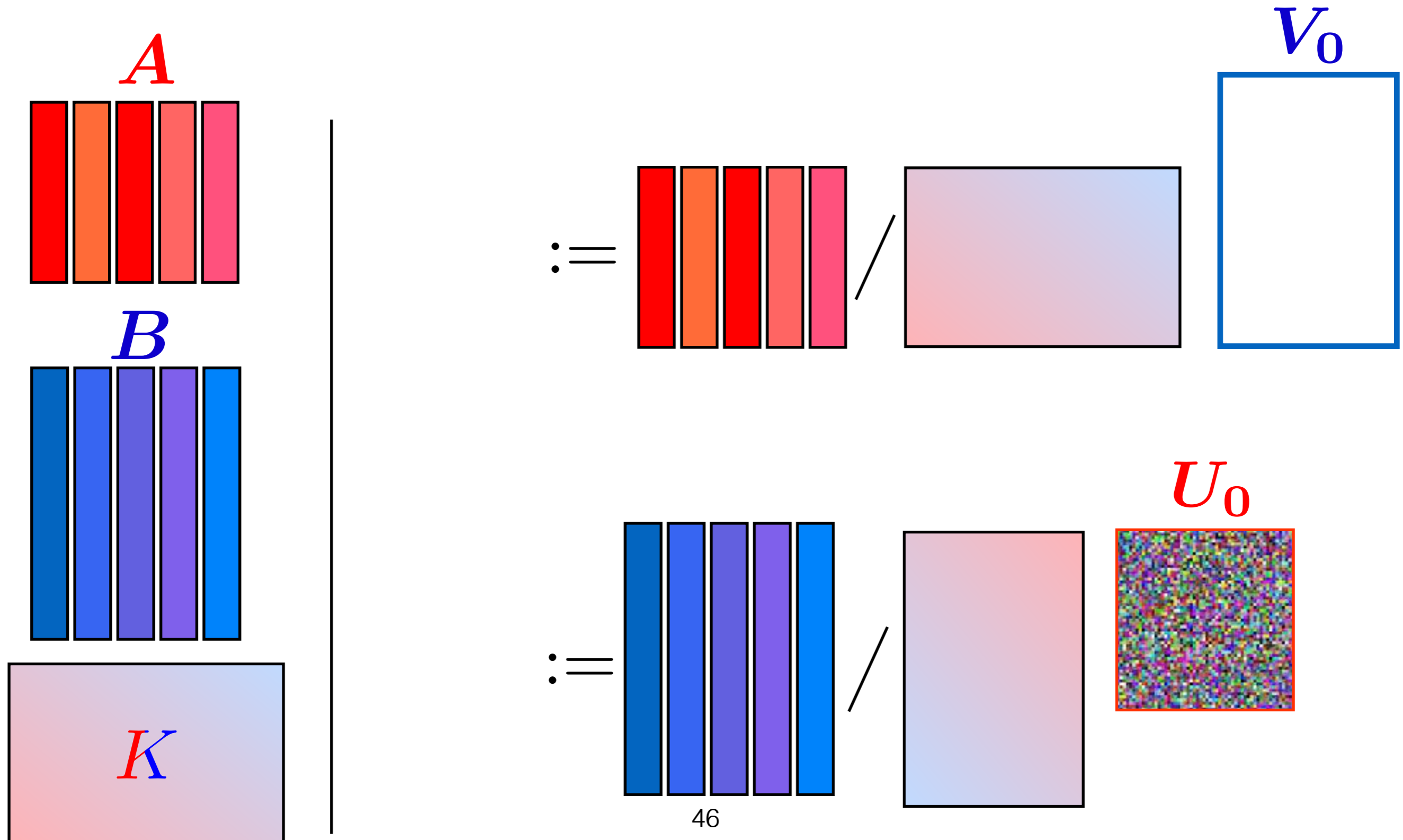
Also embarrassingly parallel

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



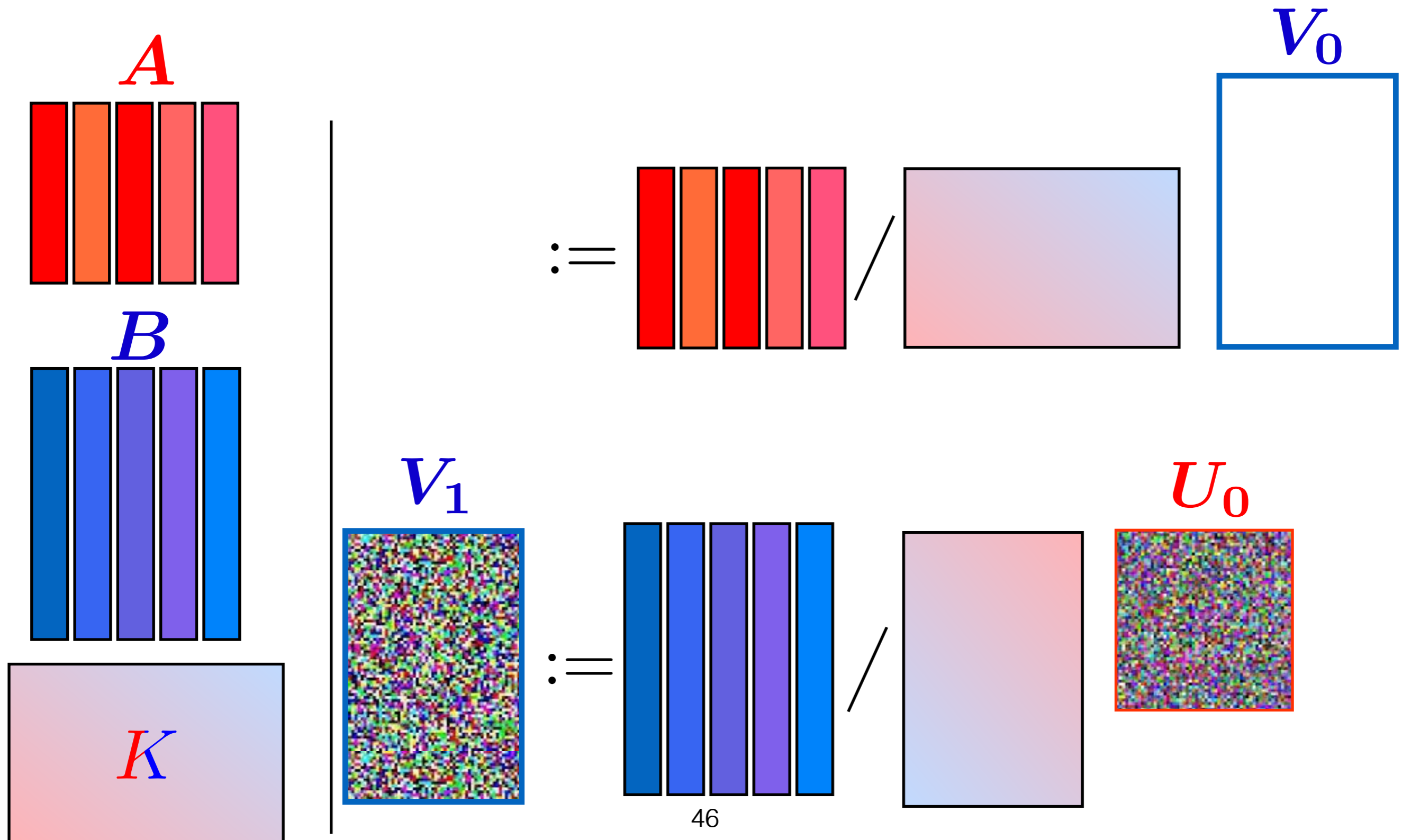
Also embarrassingly parallel

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



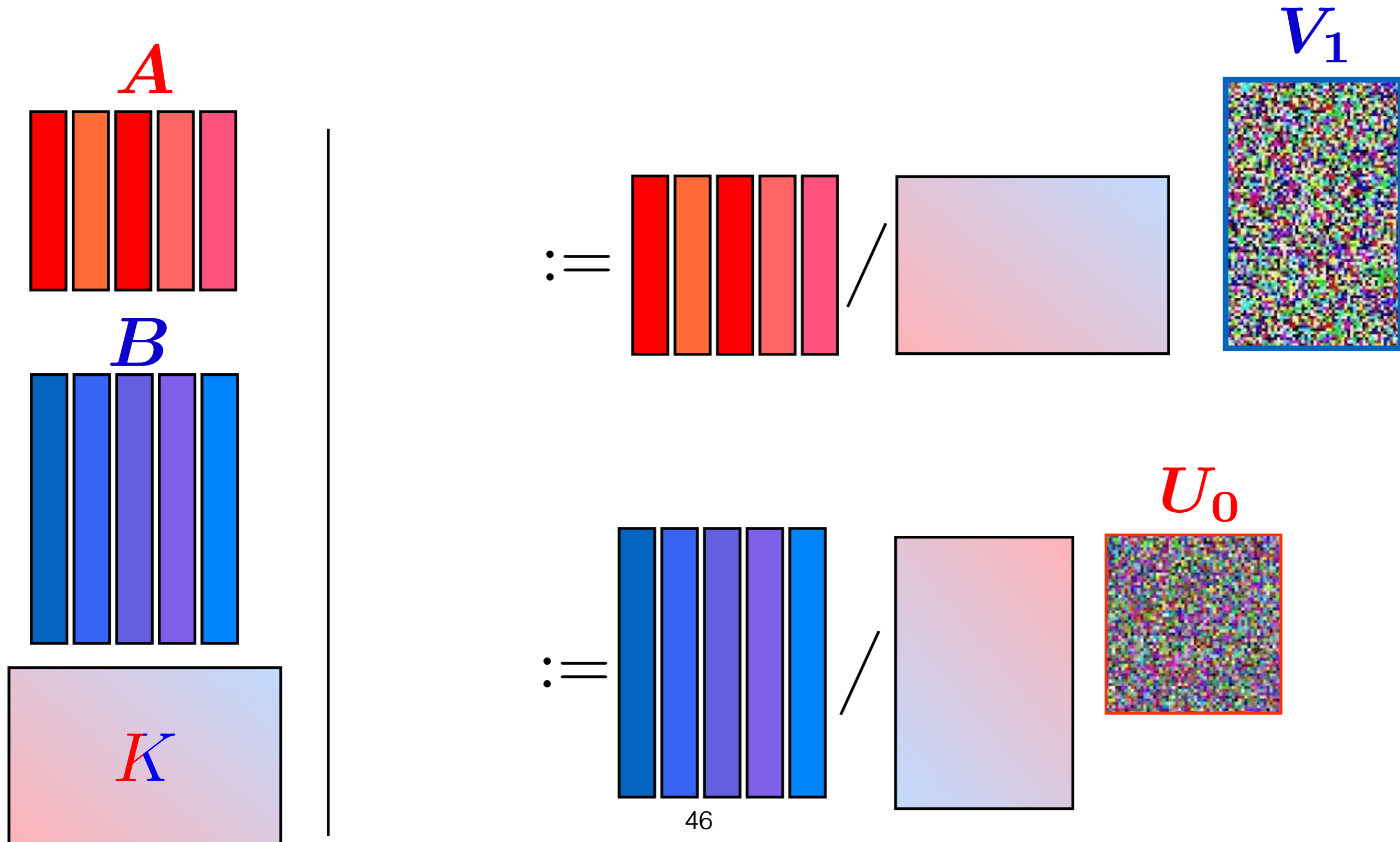
Also embarrassingly parallel

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



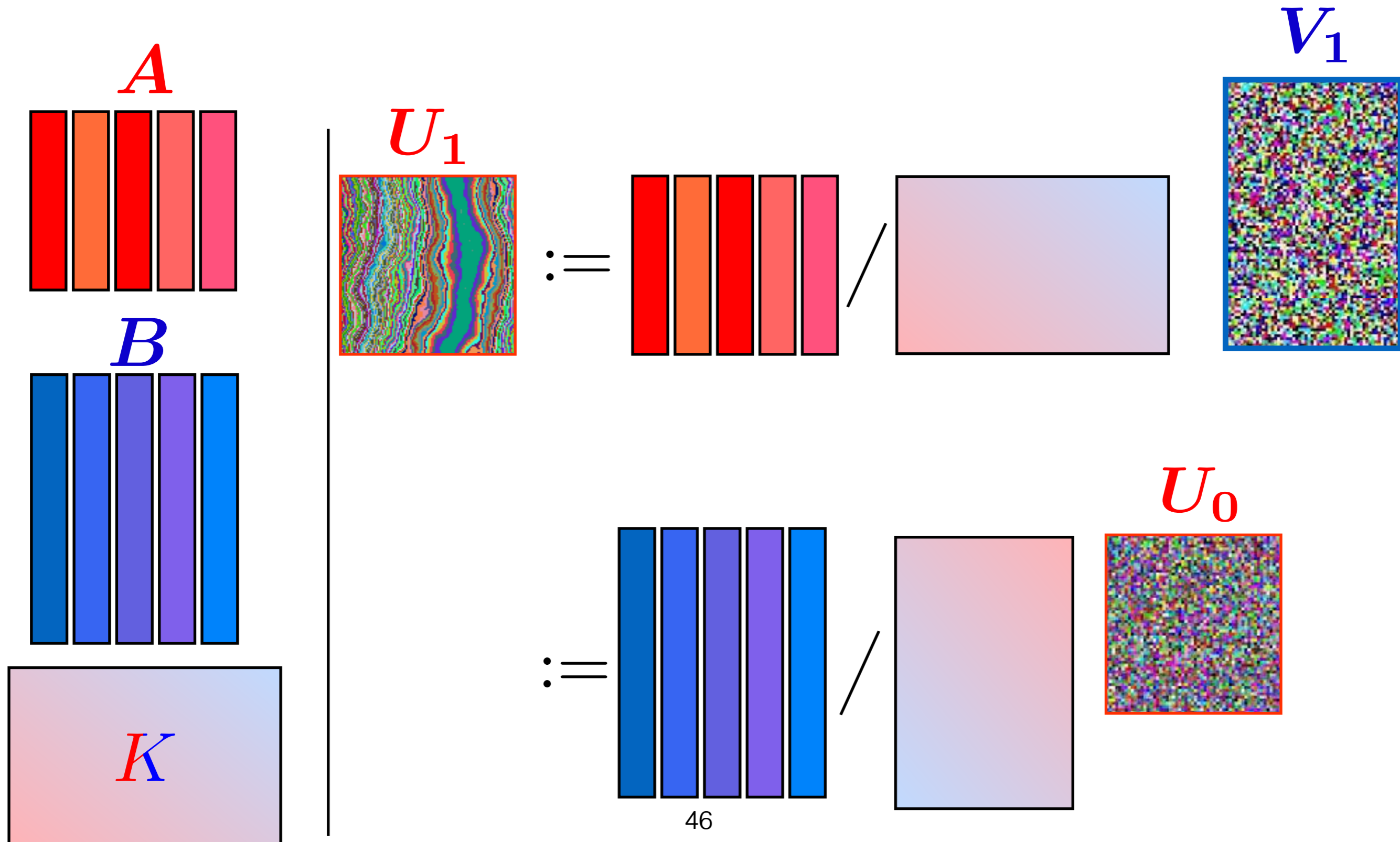
Also embarrassingly parallel

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



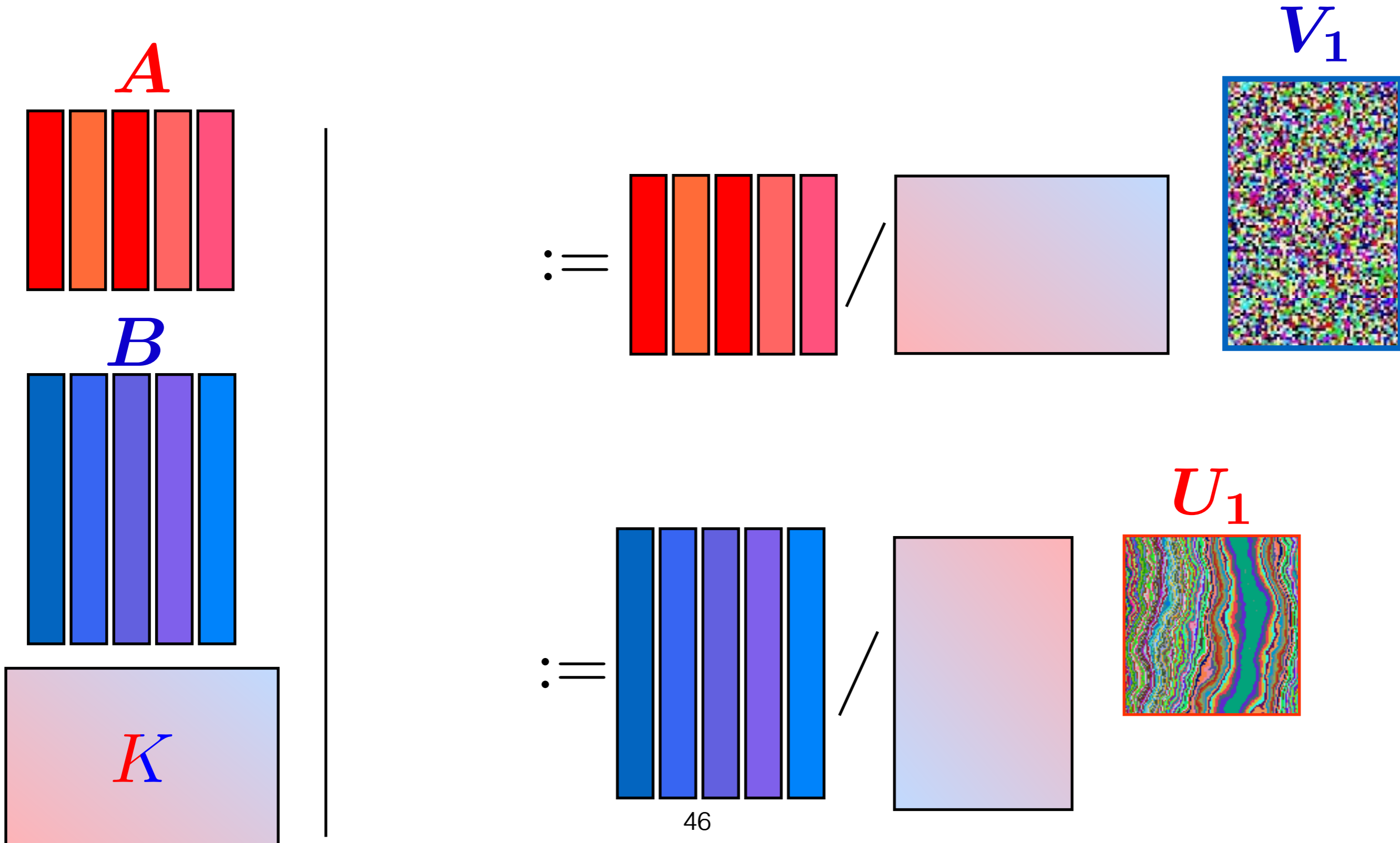
Also embarrassingly parallel

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



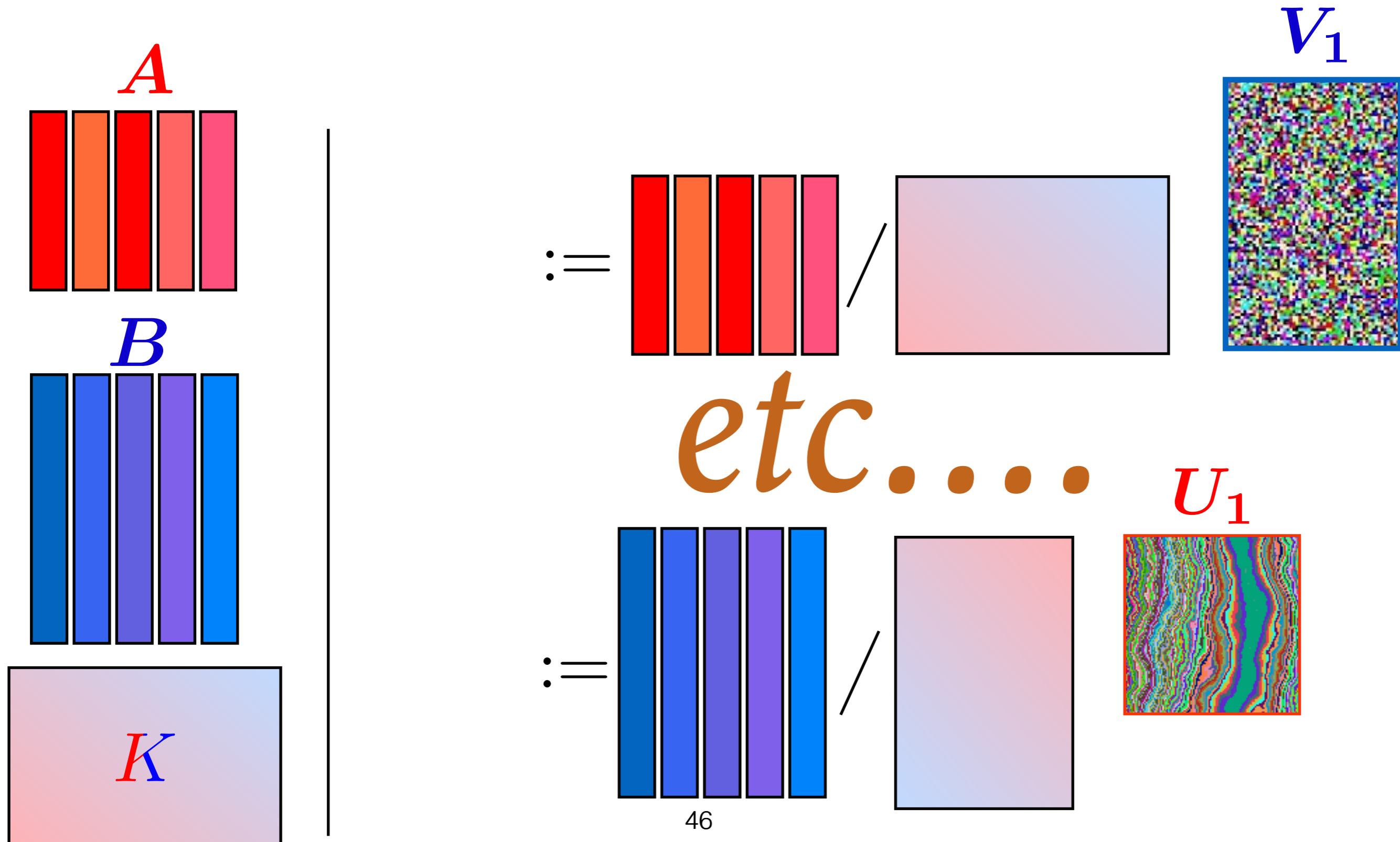
Also embarrassingly parallel

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



Also embarrassingly parallel

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



Sinkhorn as a Dual Algorithm

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

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

REGULARIZED DISCRETE PRIMAL

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

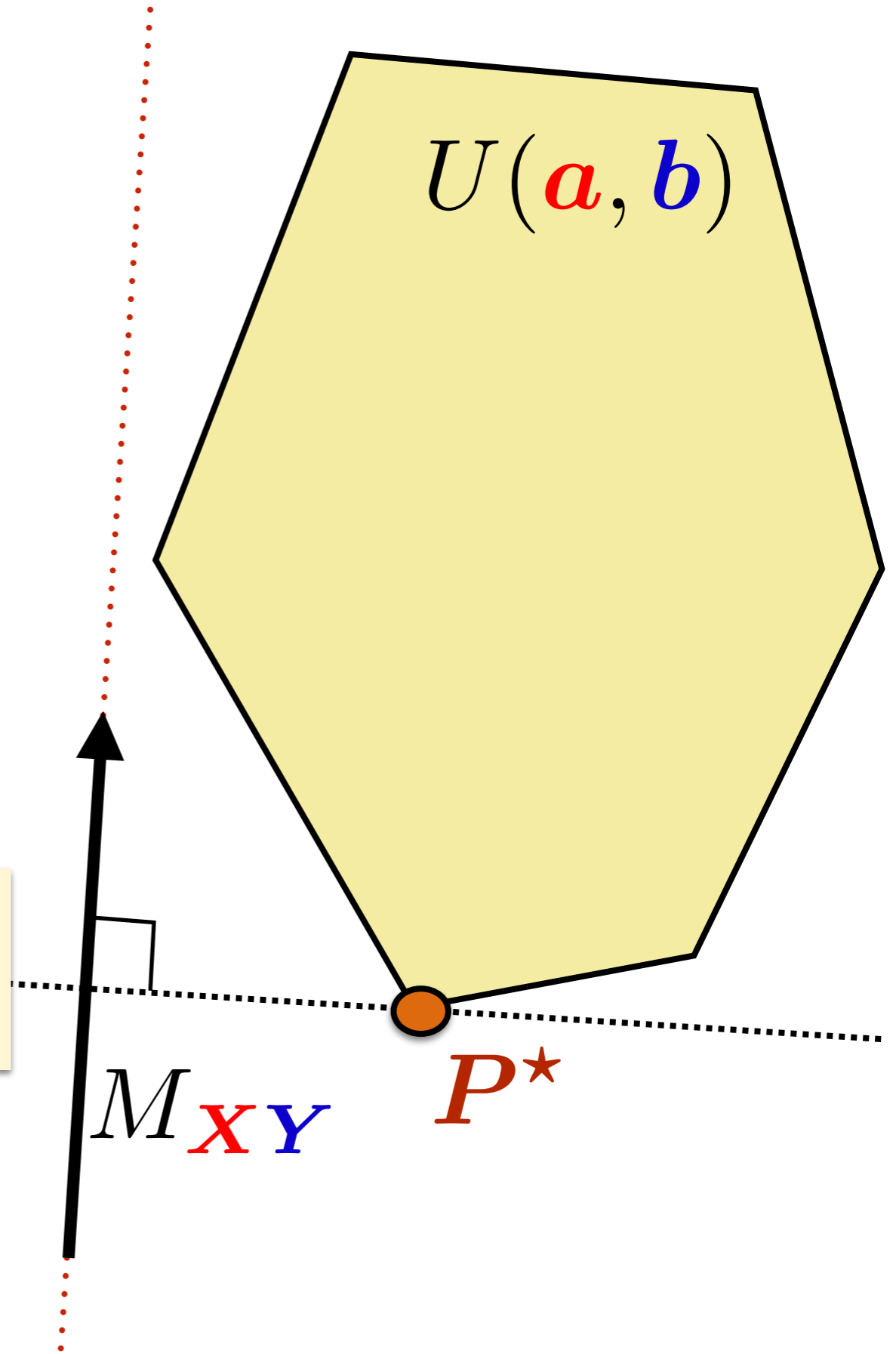
REGULARIZED DISCRETE DUAL

Sinkhorn = *Block Coordinate Ascent* on Dual

Sinkhorn in between W and MMD

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

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

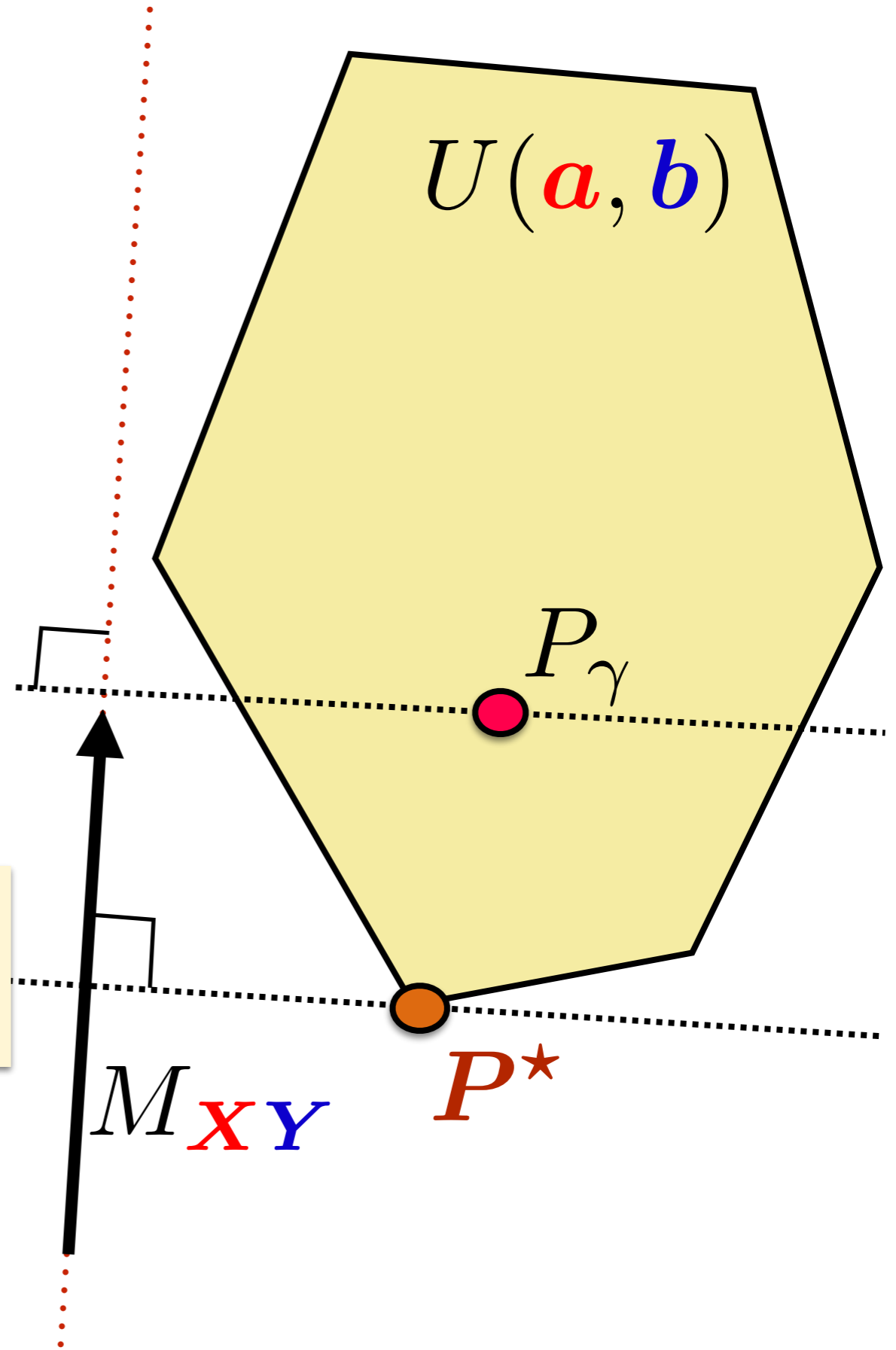


Sinkhorn in between W and MMD

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

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

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



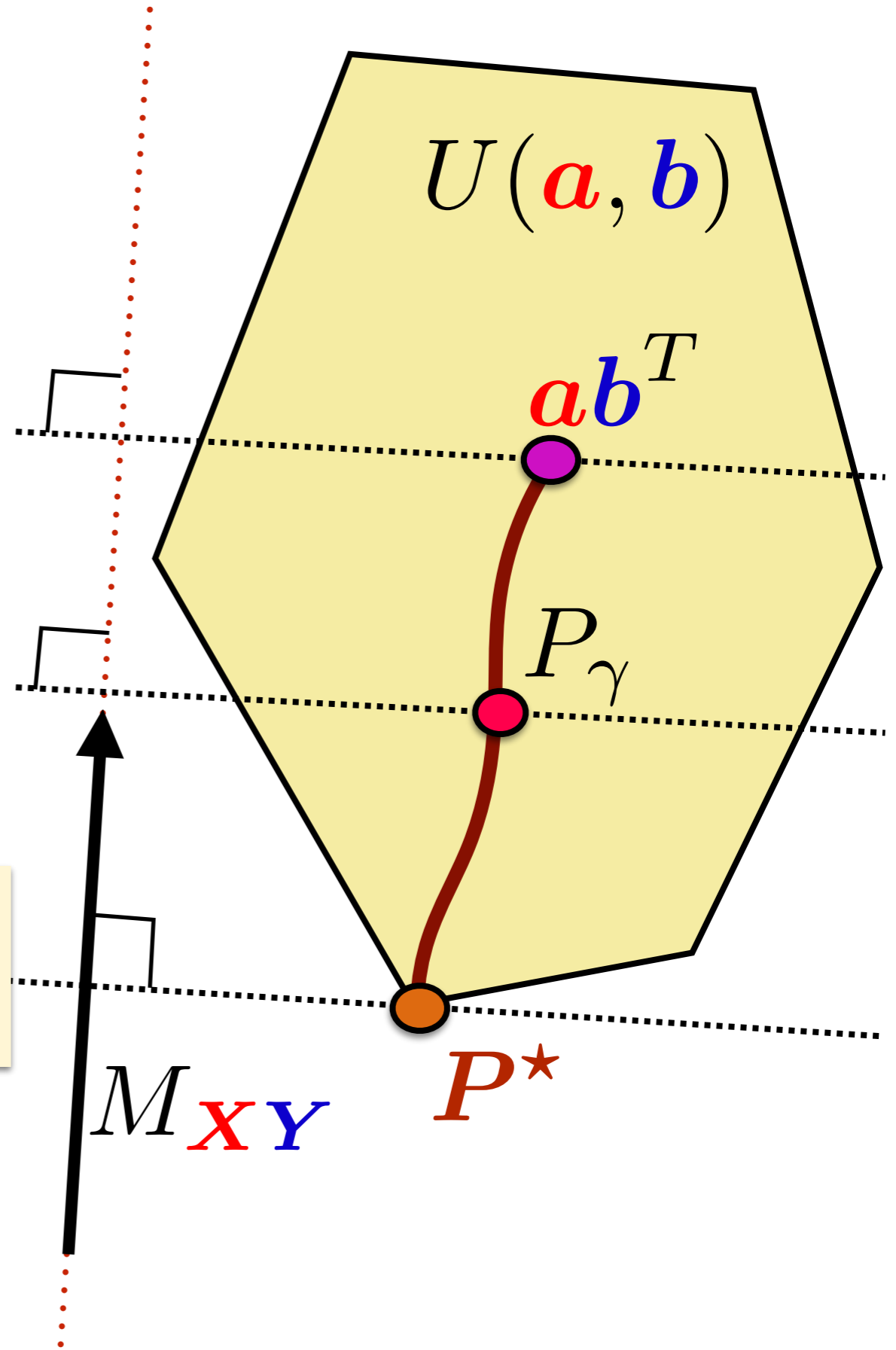
Sinkhorn in between W and MMD

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

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

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

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



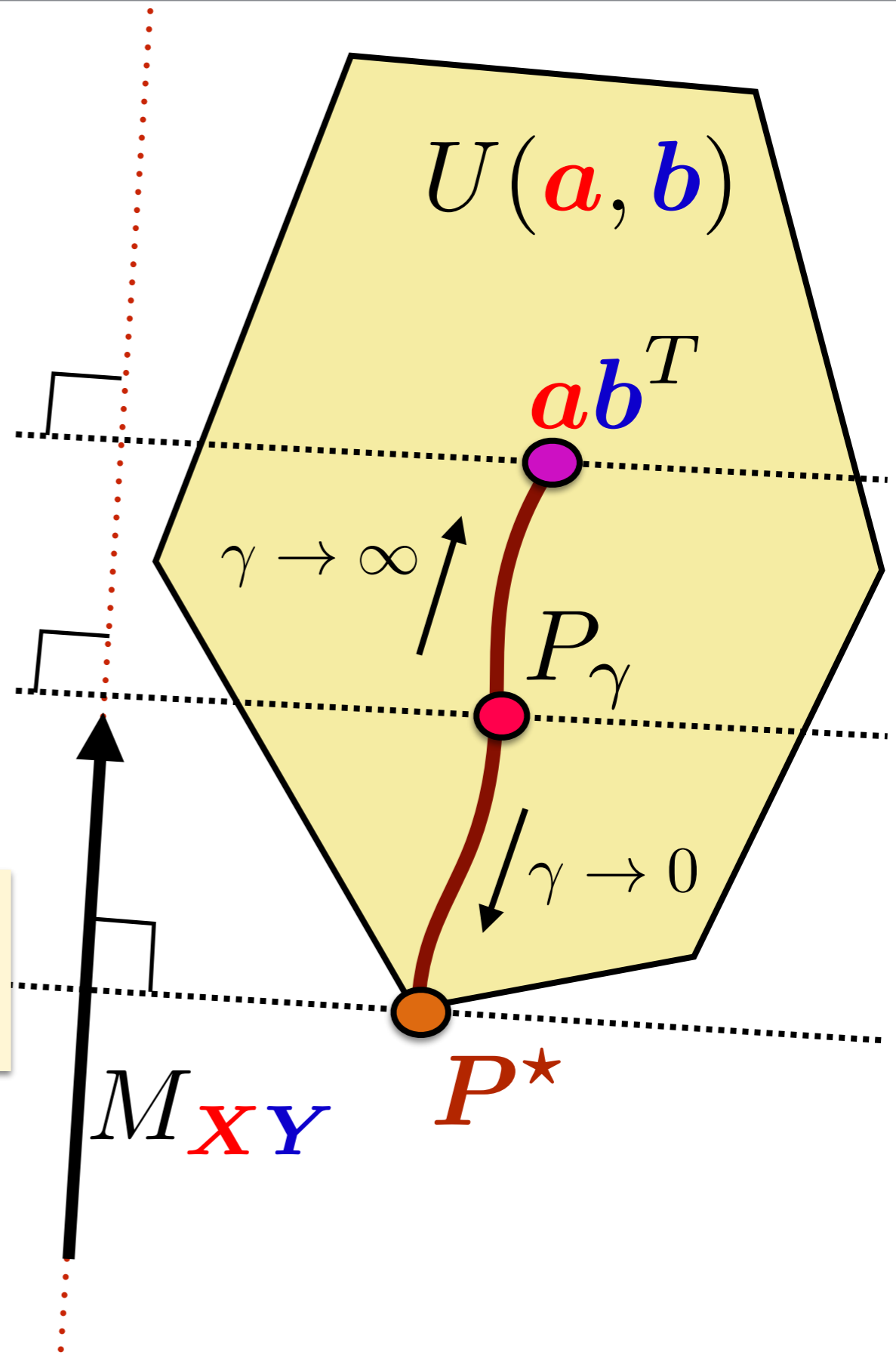
Sinkhorn in between W and MMD

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

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

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

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



Sinkhorn in between W and MMD

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

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

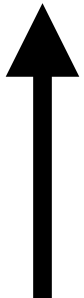
$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) = \langle P_\gamma, M_{\mathbf{XY}} \rangle$$

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

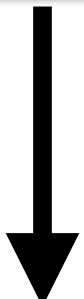
$$W^p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \langle \mathbf{P}^*, M_{\mathbf{XY}} \rangle$$

Sinkhorn in between W and MMD

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

$\gamma \rightarrow \infty$ 

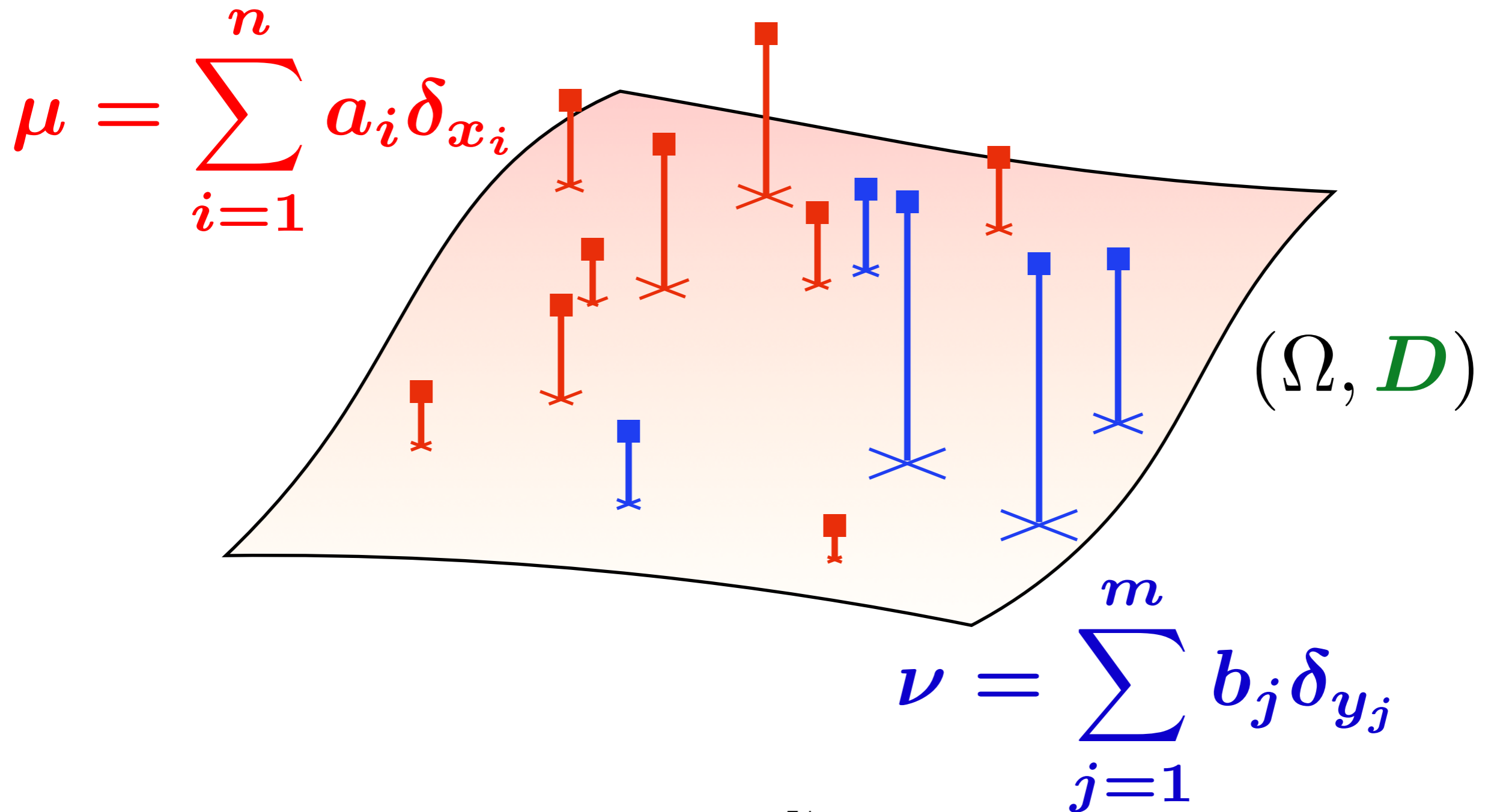
$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$\gamma \rightarrow 0$ 

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

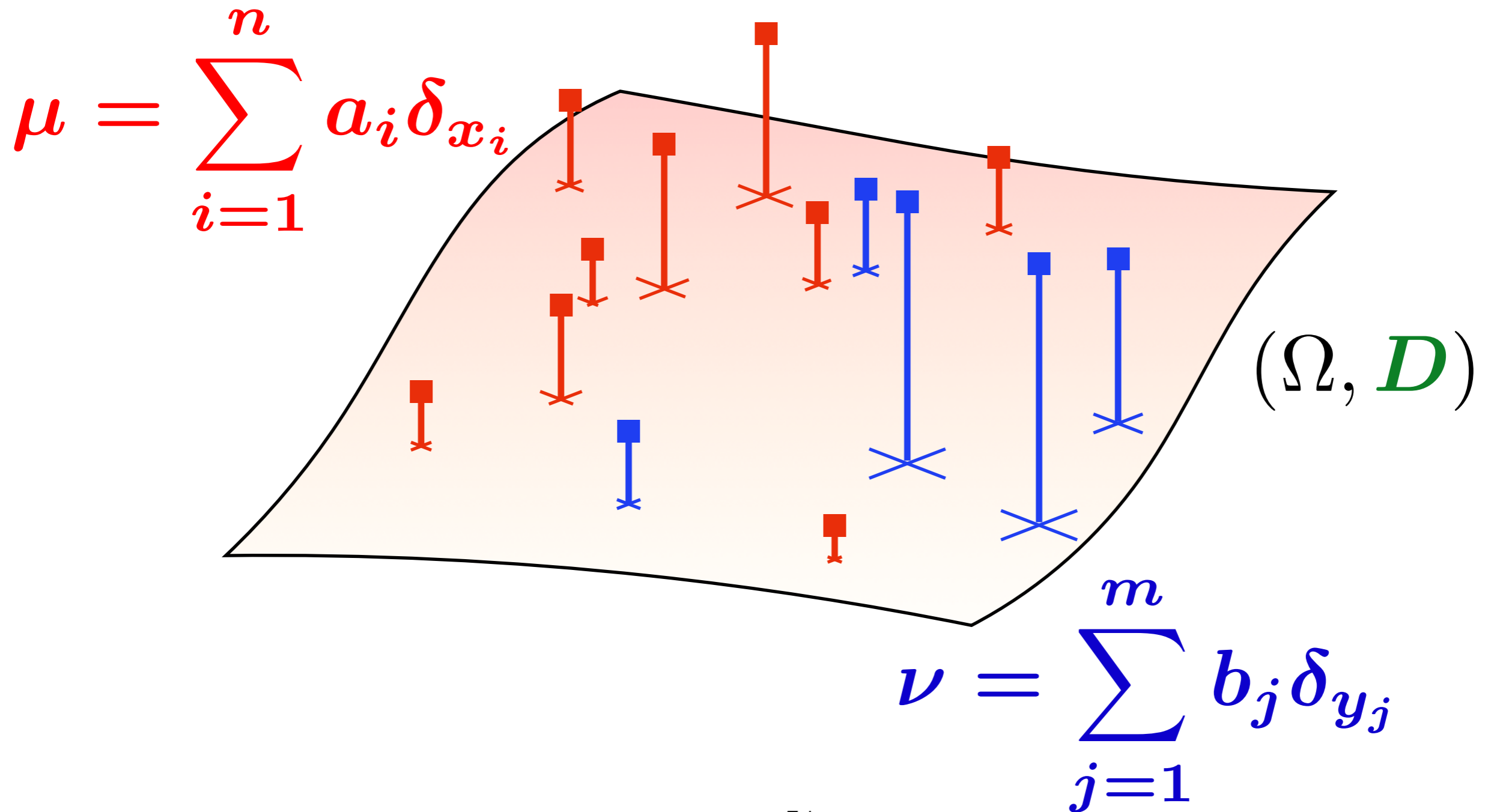
Differentiability of W

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



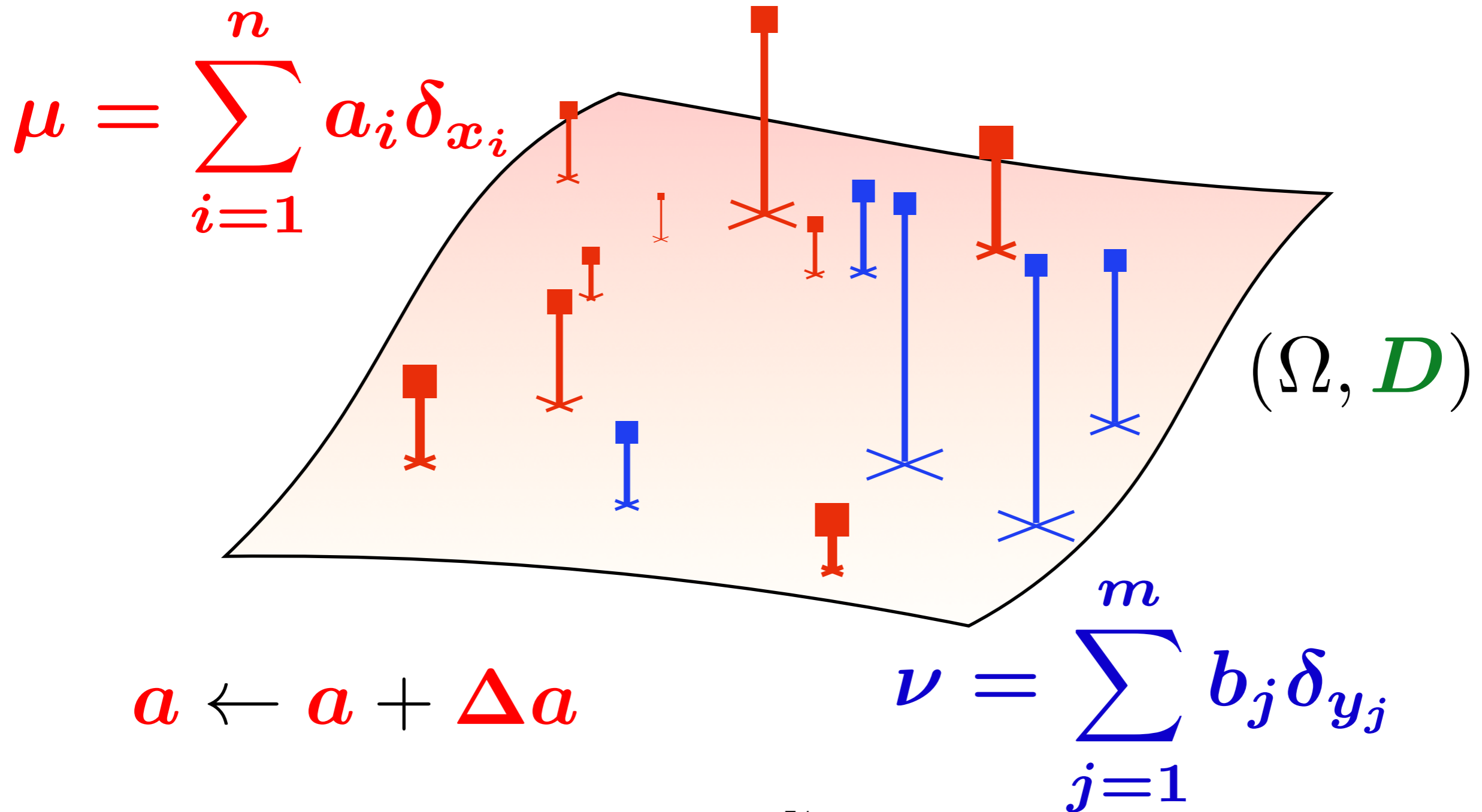
Differentiability of W

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



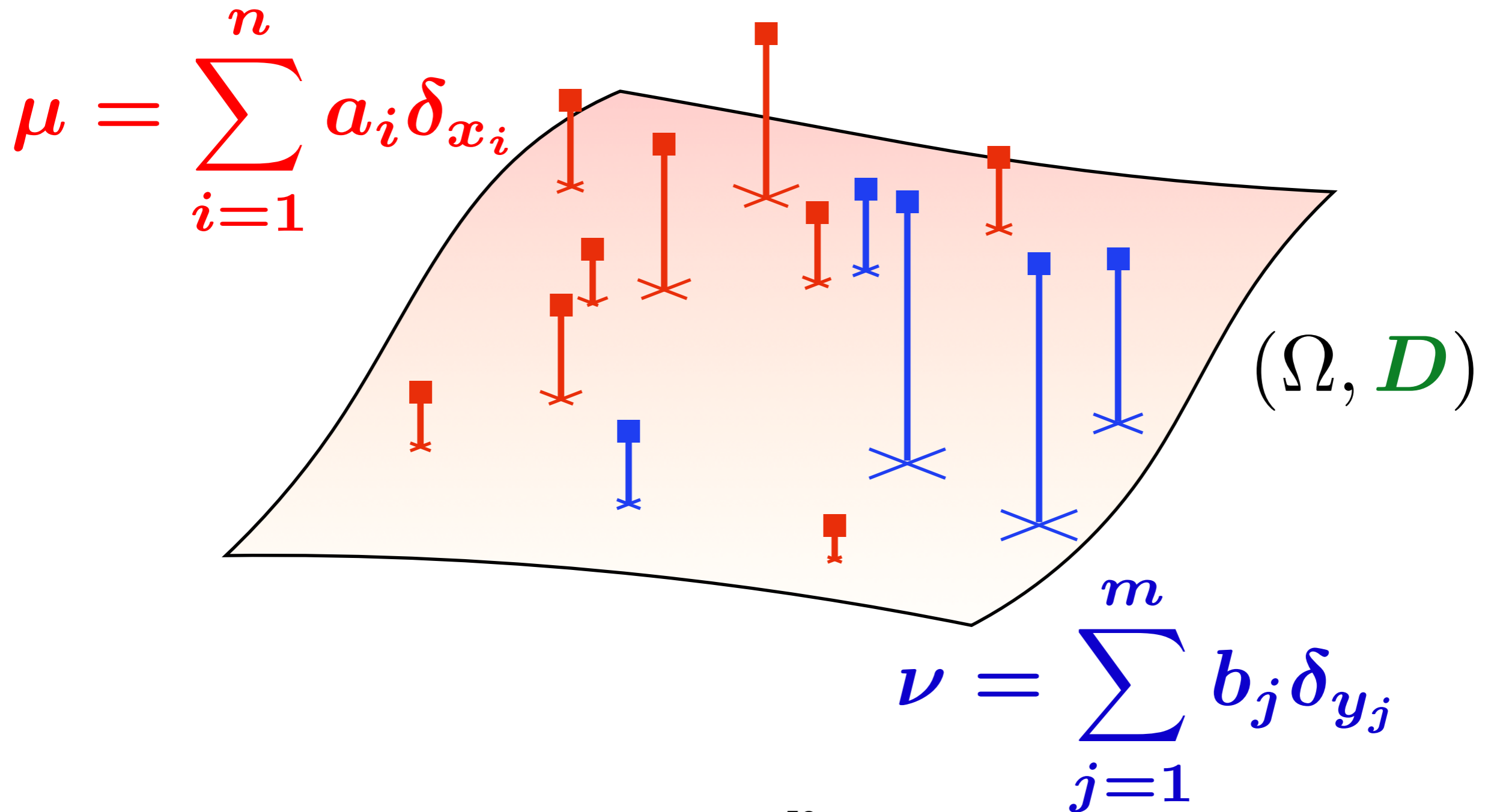
Differentiability of W

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



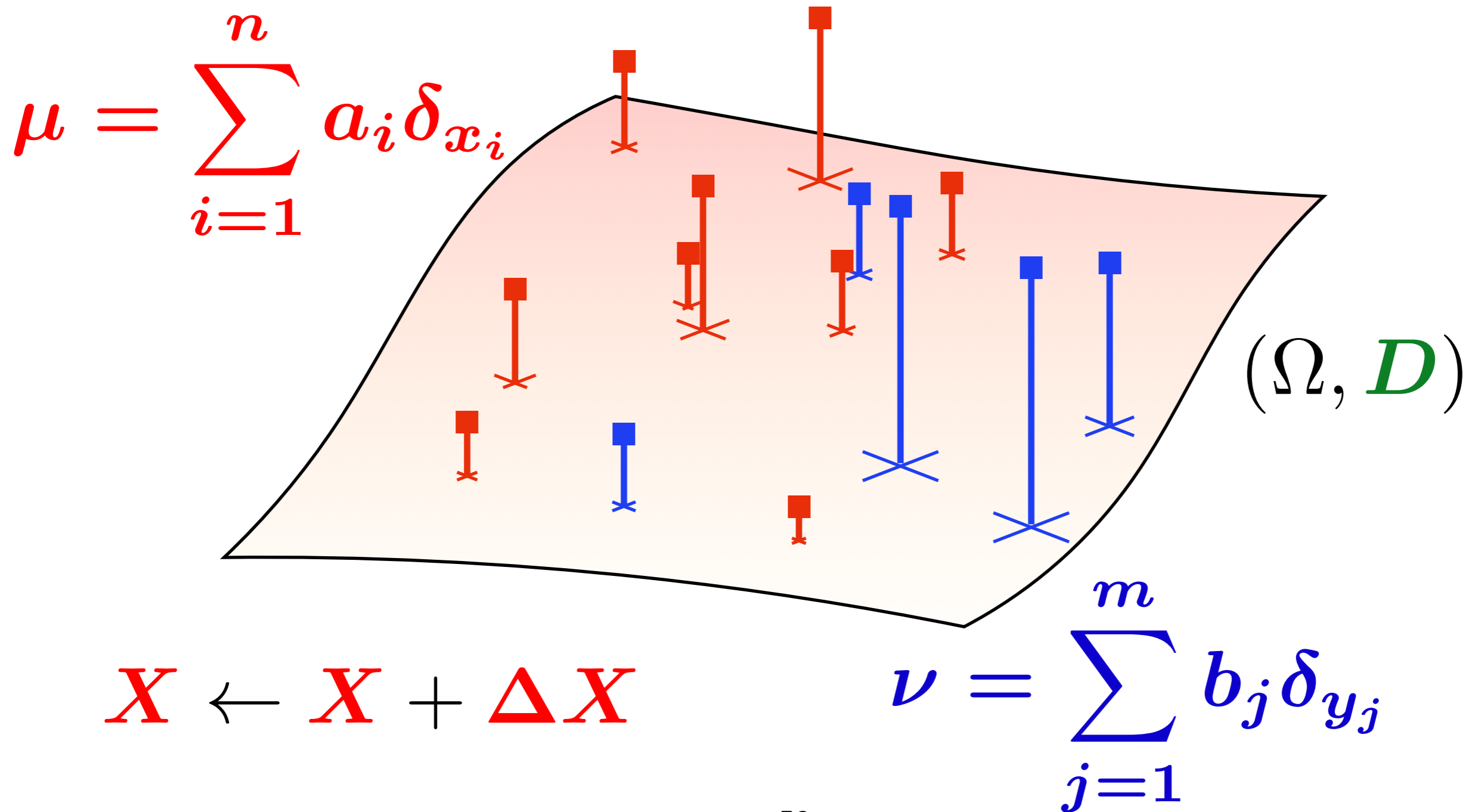
Sinkhorn \rightsquigarrow *Differentiability*

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



Sinkhorn \rightsquigarrow *Differentiability*

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



How to decrease W ? change weights

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

DUAL

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

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

How to decrease W ? change locations

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

PRIMAL

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

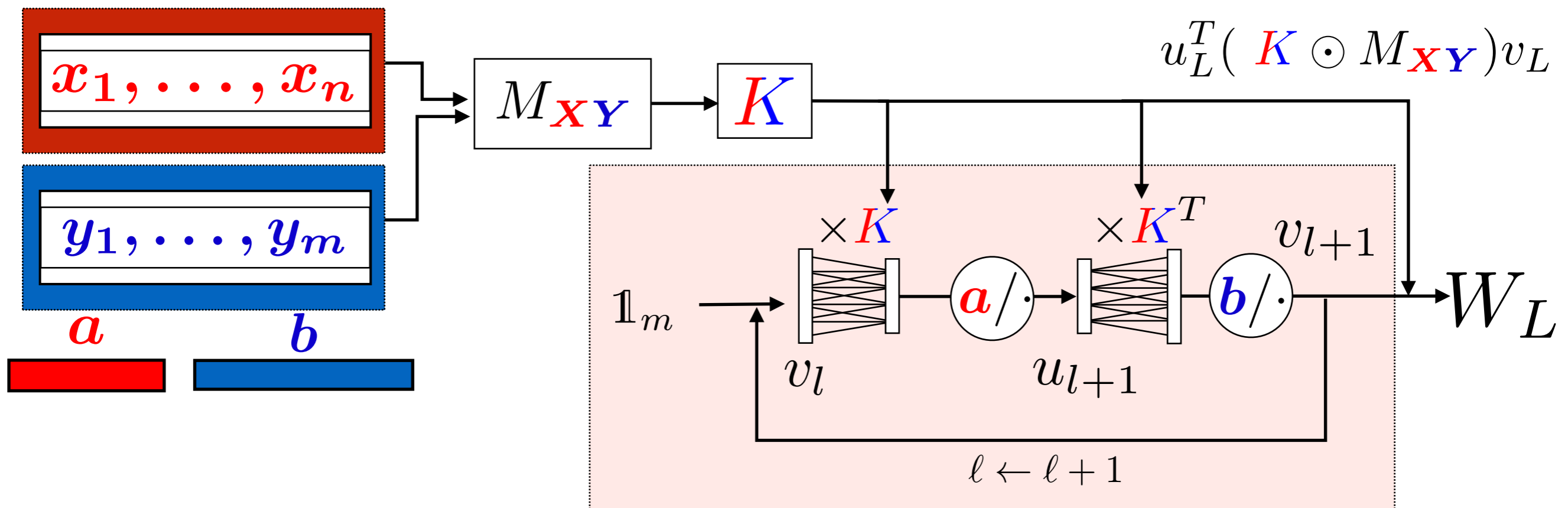
Prop. $p = 2, \Omega = \mathbb{R}^d$. $W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu})$ is differentiable w.r.t. X , with

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

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

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



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

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

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

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

[Hashimoto'16][Bonnel'16][Shalit'16][Flammery'16]

Minimum Kantorovich Estimators

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

[Bassetti'06] 1st reference discussing this approach.

Challenge: $\nabla_{\theta} W(\nu_{\text{data}}, f_{\theta\#} \mu)$?

[Montavon'16] use regularized OT in a finite setting.

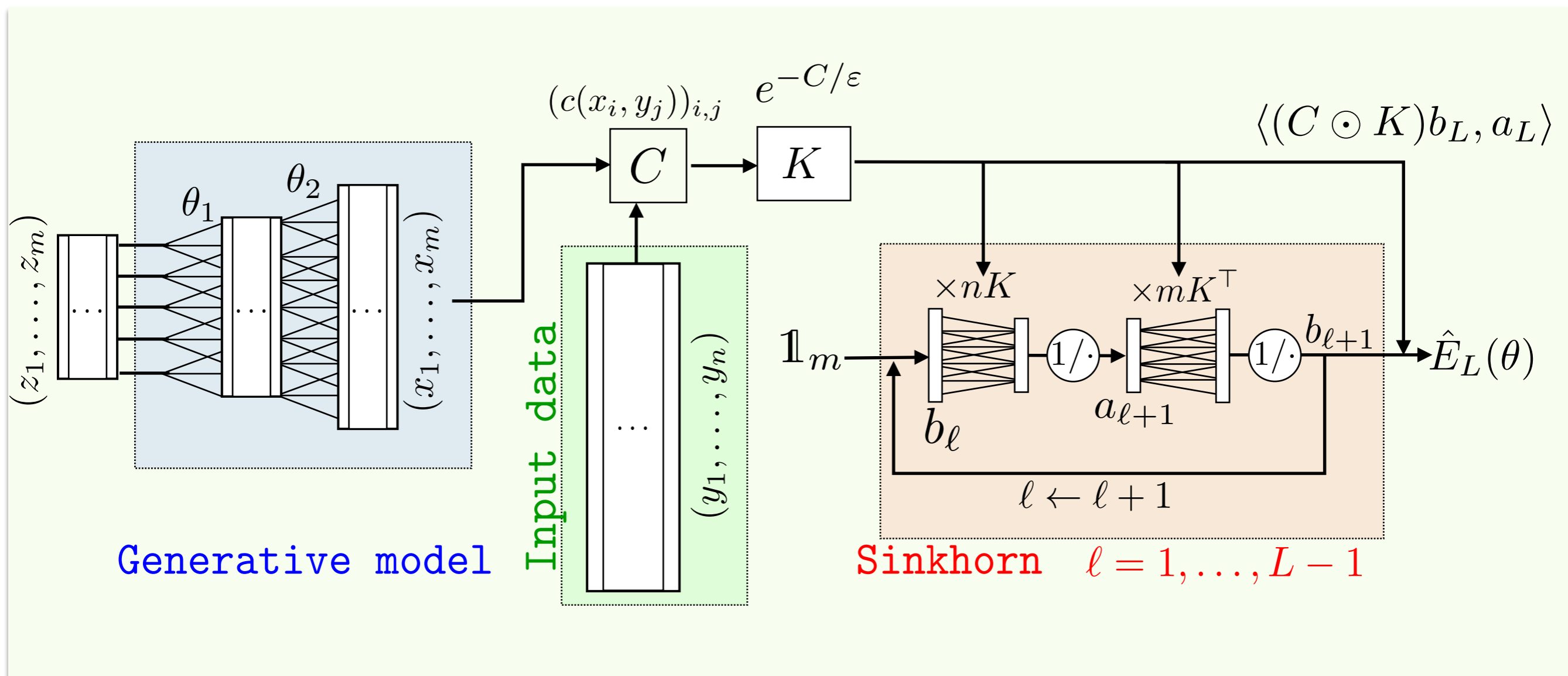
[Arjovsky'17] (WGAN) uses a NN to approximate dual solutions and recover gradient w.r.t. parameter

[Bernton'17] reject mechanism $W(\text{sample}, \text{data})$

[Genevay'17, Salimans'17] (Sinkhorn approach)

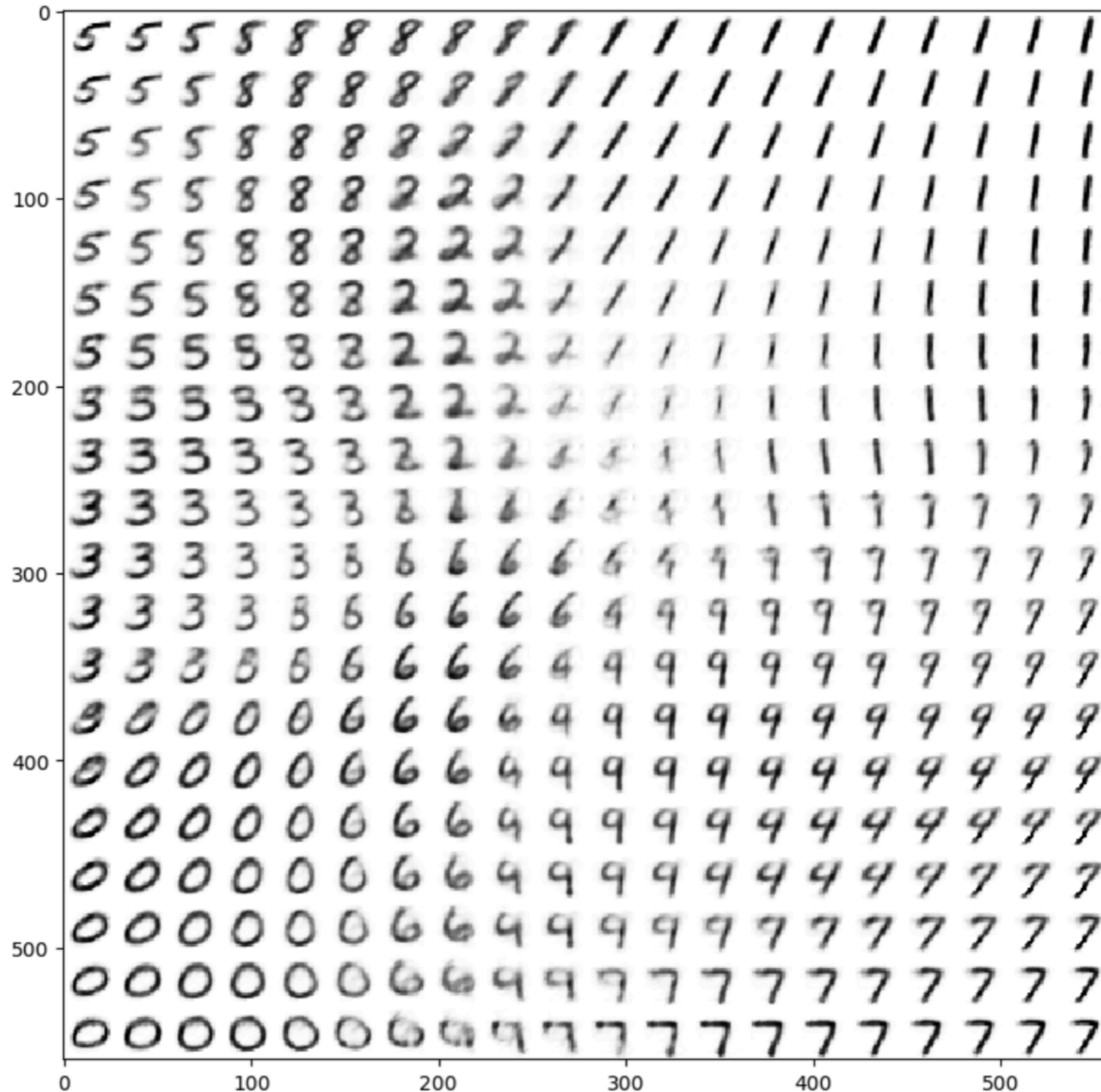
Proposal: Autodiff OT using Sinkhorn

Approximate W loss by the transport cost \bar{W}_L after L Sinkhorn iterations.



[GPC'17]

Example: MNIST, Learning f_{θ}



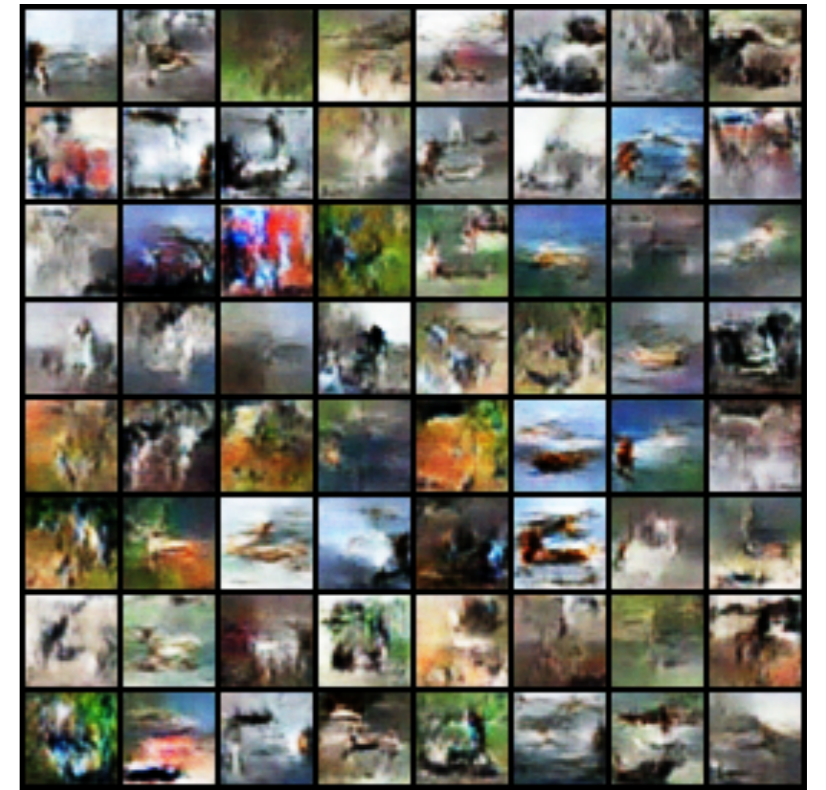
Example: Generation of Images



MMD-GAN



$\tau = 1000$



$\tau = 10$

- Learning with CIFAR-10 images
- In these examples the cost function is also learned adversarially, as a NN mapping onto feature vectors.

Concluding Remarks

- *Regularized OT* is much faster than OT.
- *Regularized OT* can interpolate between W and the *MMD / Energy distance* metrics.
- The solution of *regularized OT* is “*auto-differentiable*”.
- **Many open problems remain!**