Optimal Transport

Marco Cuturi

MLSS NTU 2021

ENSAE

IP PARIS

Google AI
Brain Team

Optimal
Transport
Tools

First, a big thanks !

To the MLSS NTU 2021 organisers

To all my collaborators on these topics,
including here a recent subset

M. Seigal

L. Petrini

L. Papaxanthos

O. Teboul

JP Vert

J. Niles-Weed

C. Bunne

T. Lin

A. Genevay

M. Scetbon

A. Gramfort

J. Josse

H. Janati

F.P. Paty

G. Peyré

B. Muzellec

C. Boyer

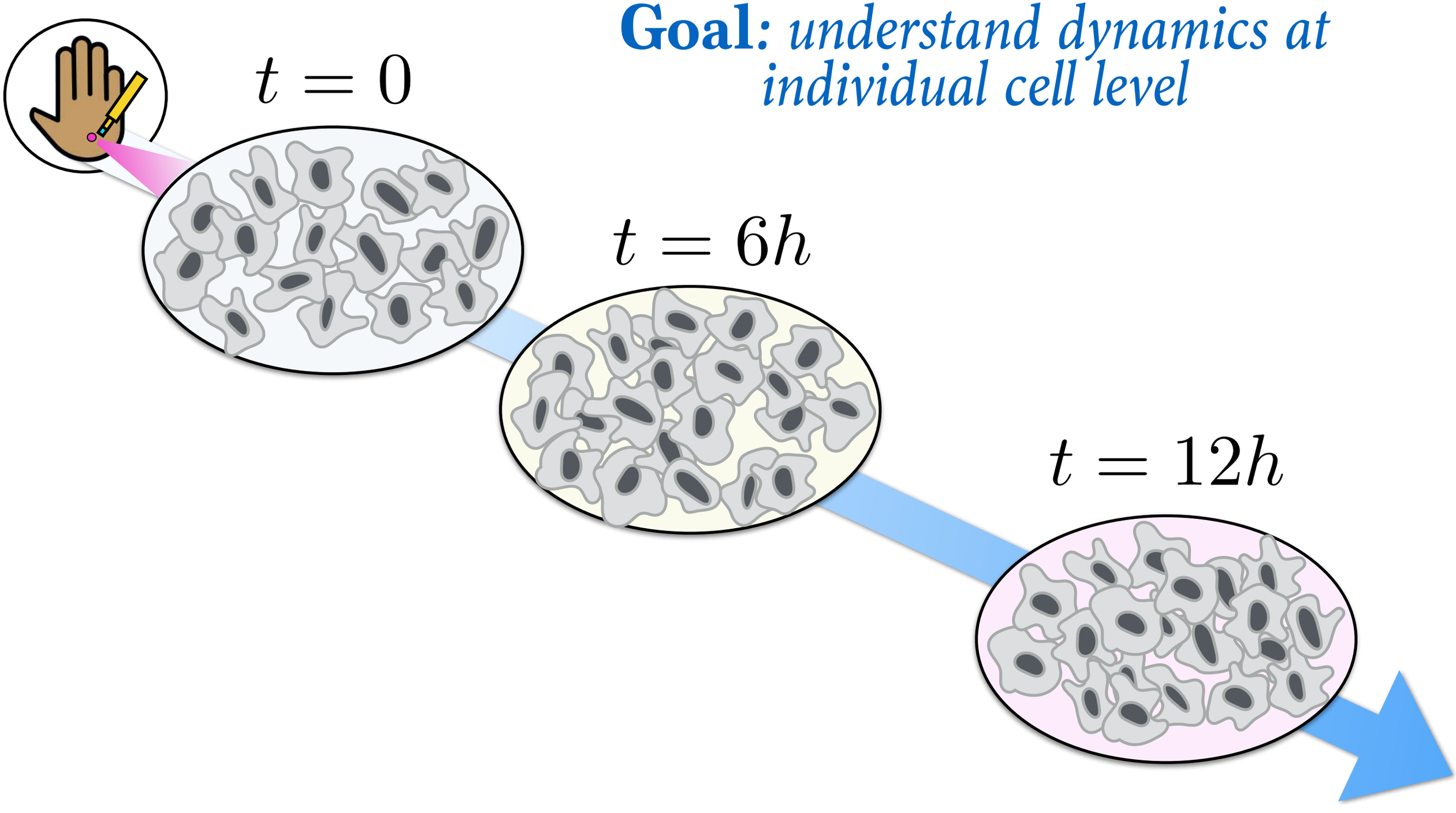
Google AI
Brain Team

ENSAE

IP PARIS

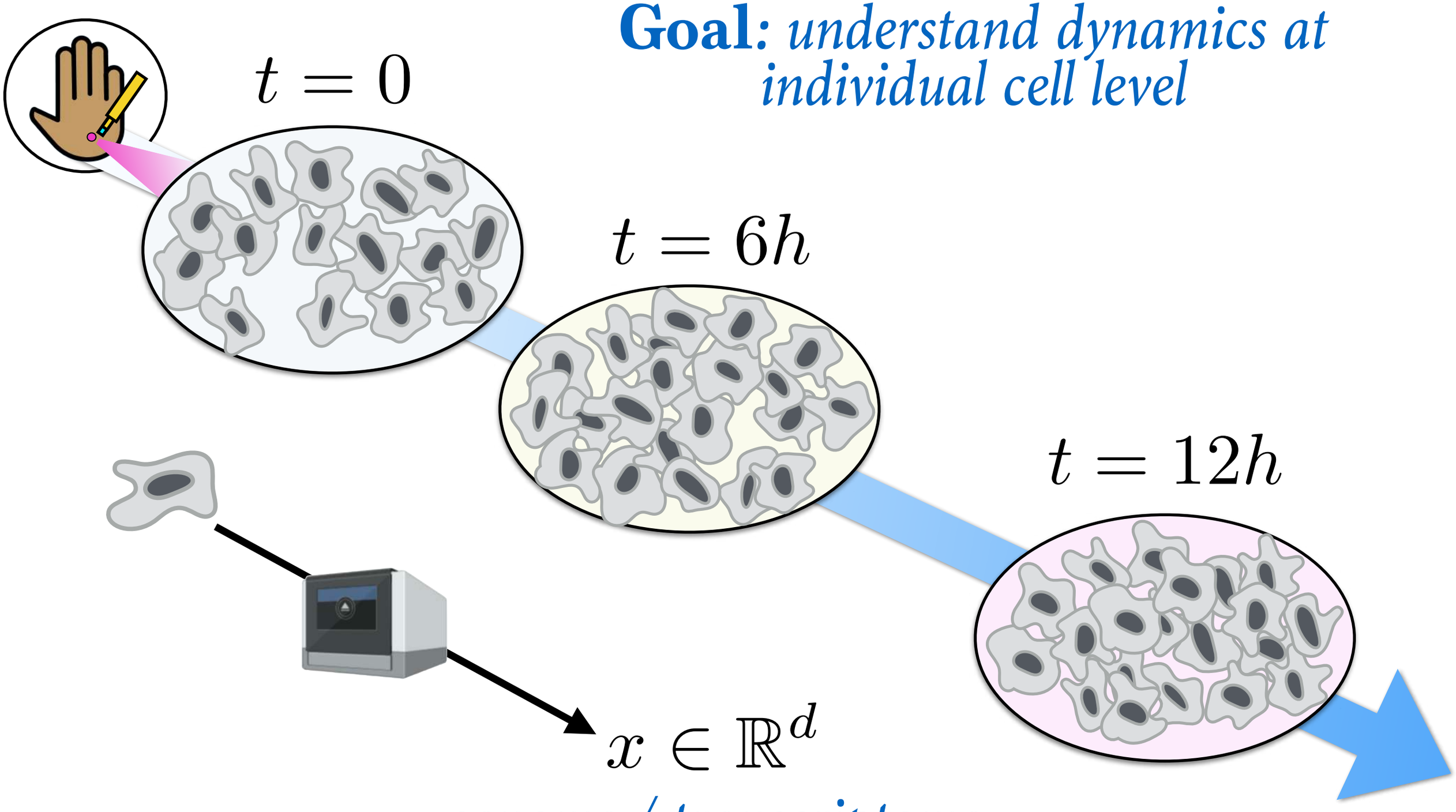
An Example from Single-Cell Genomics

Goal: *understand dynamics at individual cell level*



An Example from Single-Cell Genomics

Goal: *understand dynamics at individual cell level*

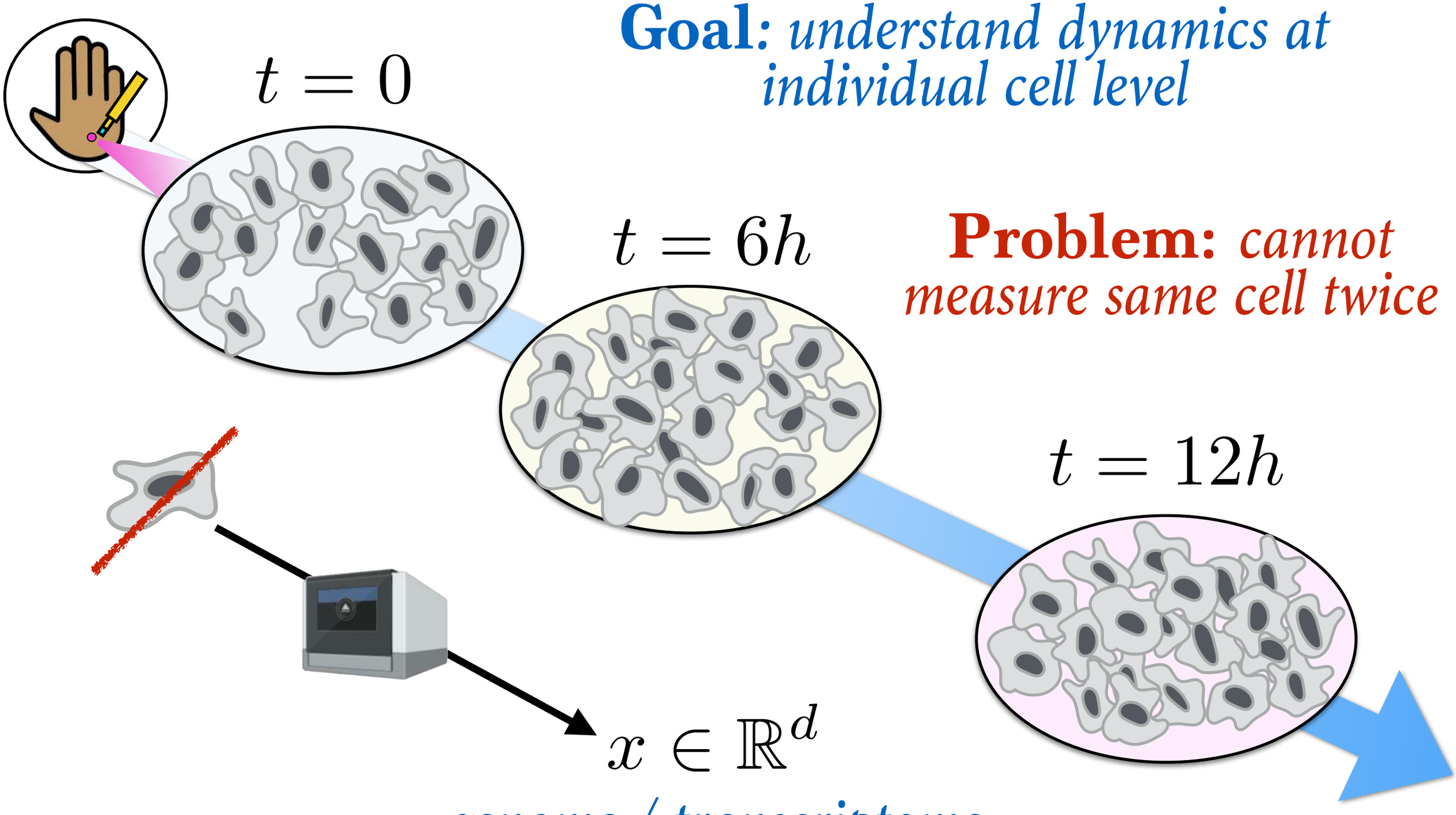


genome / transcriptome

An Example from Single-Cell Genomics

Goal: *understand dynamics at individual cell level*

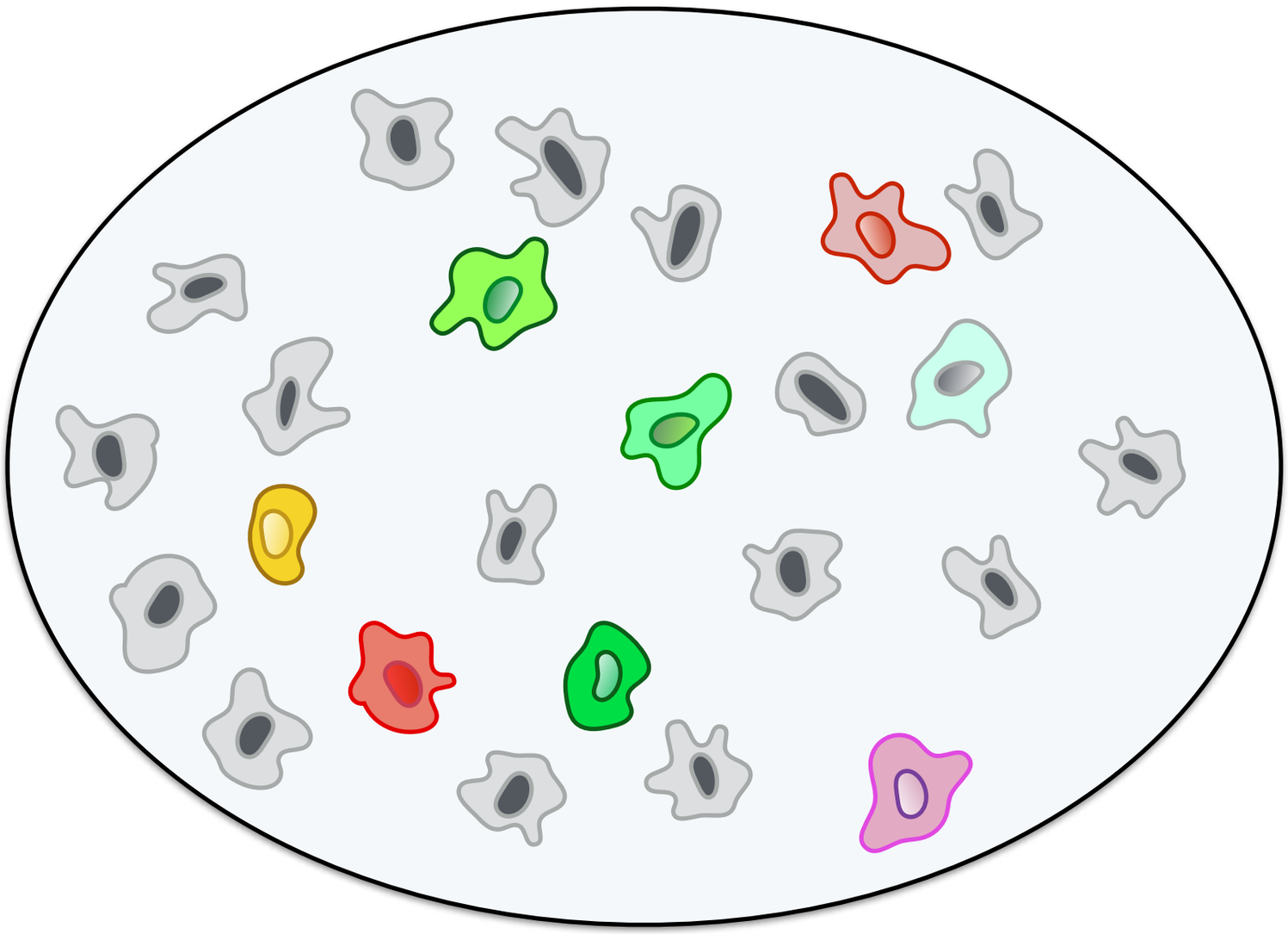
Problem: *cannot measure same cell twice*



genome / transcriptome

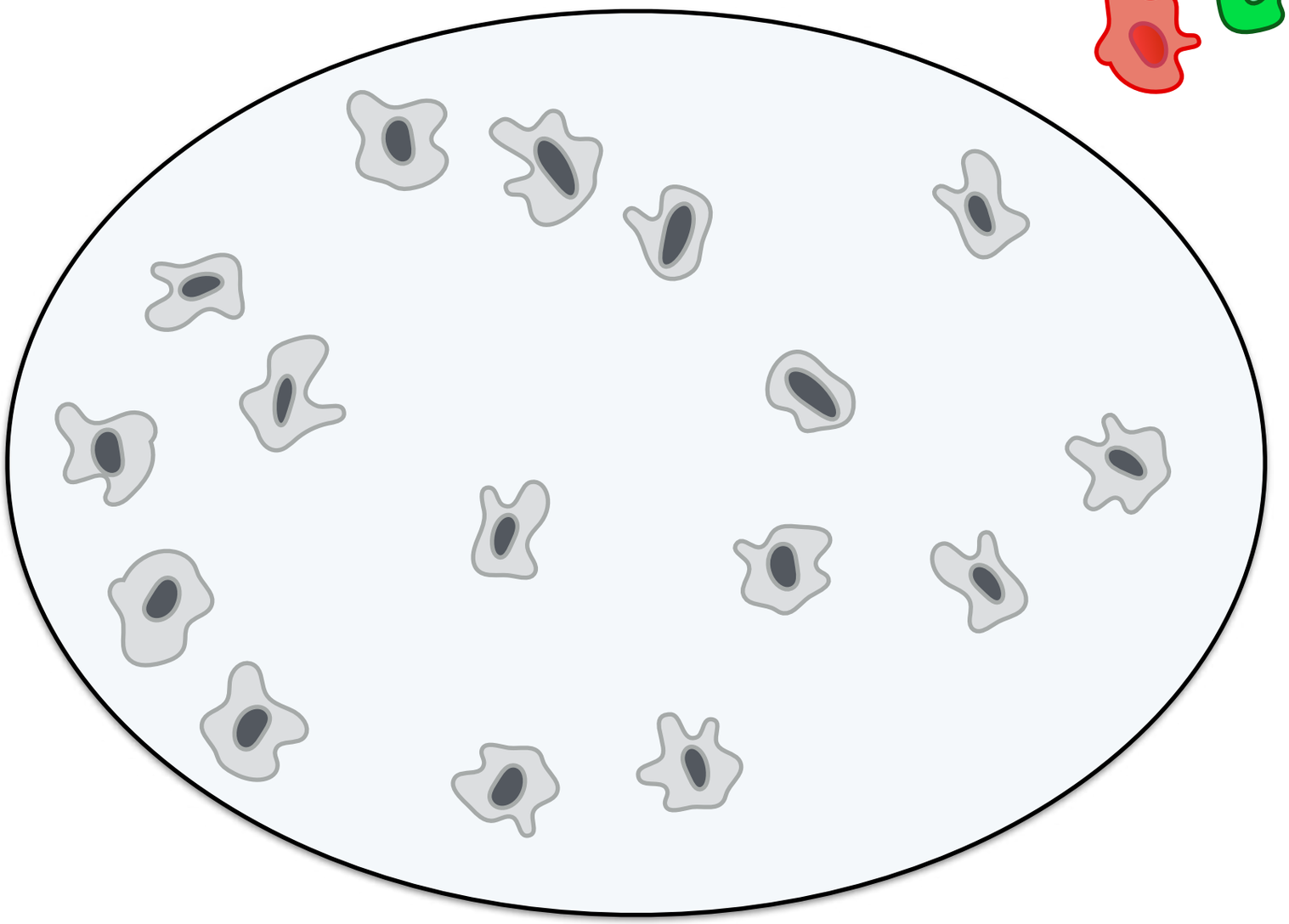
An Example from Single-Cell Genomics

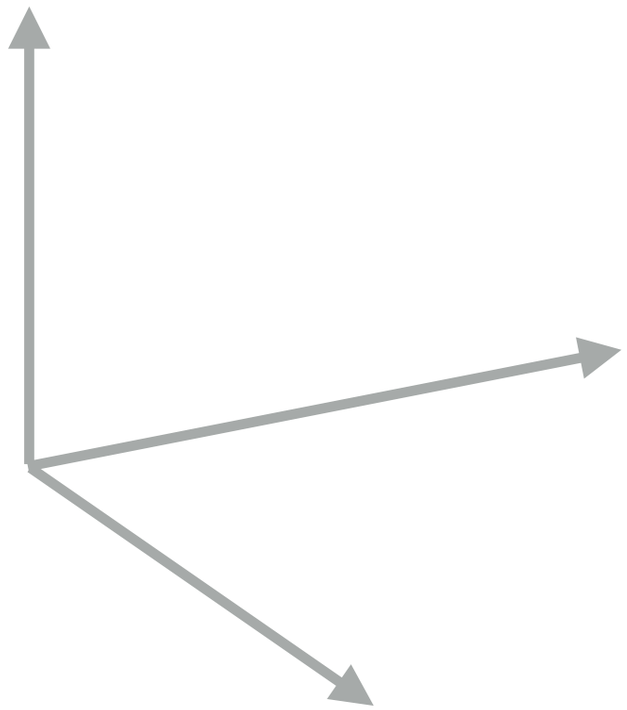
$t = 0$



An Example from Single-Cell Genomics

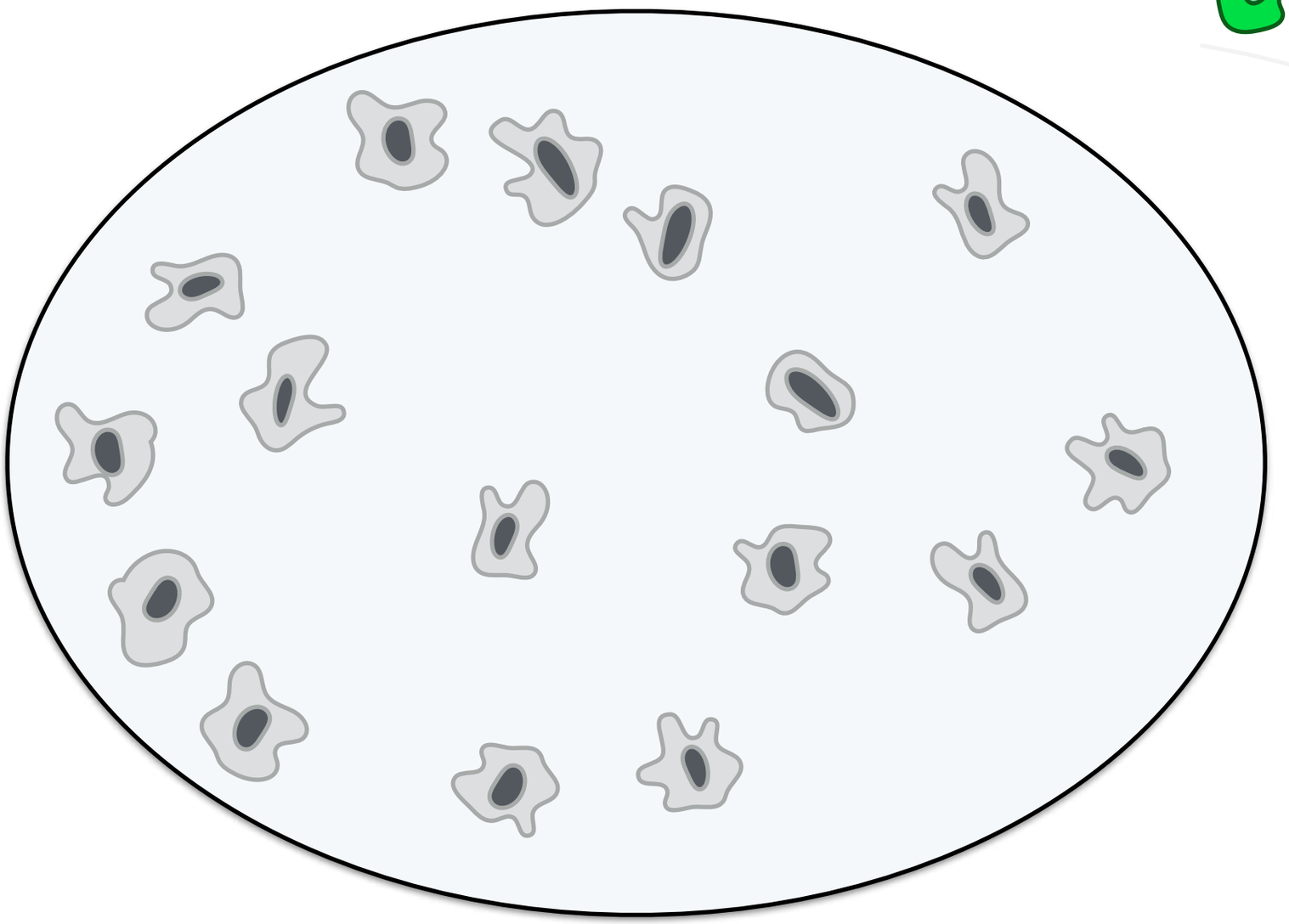
$t = 0$

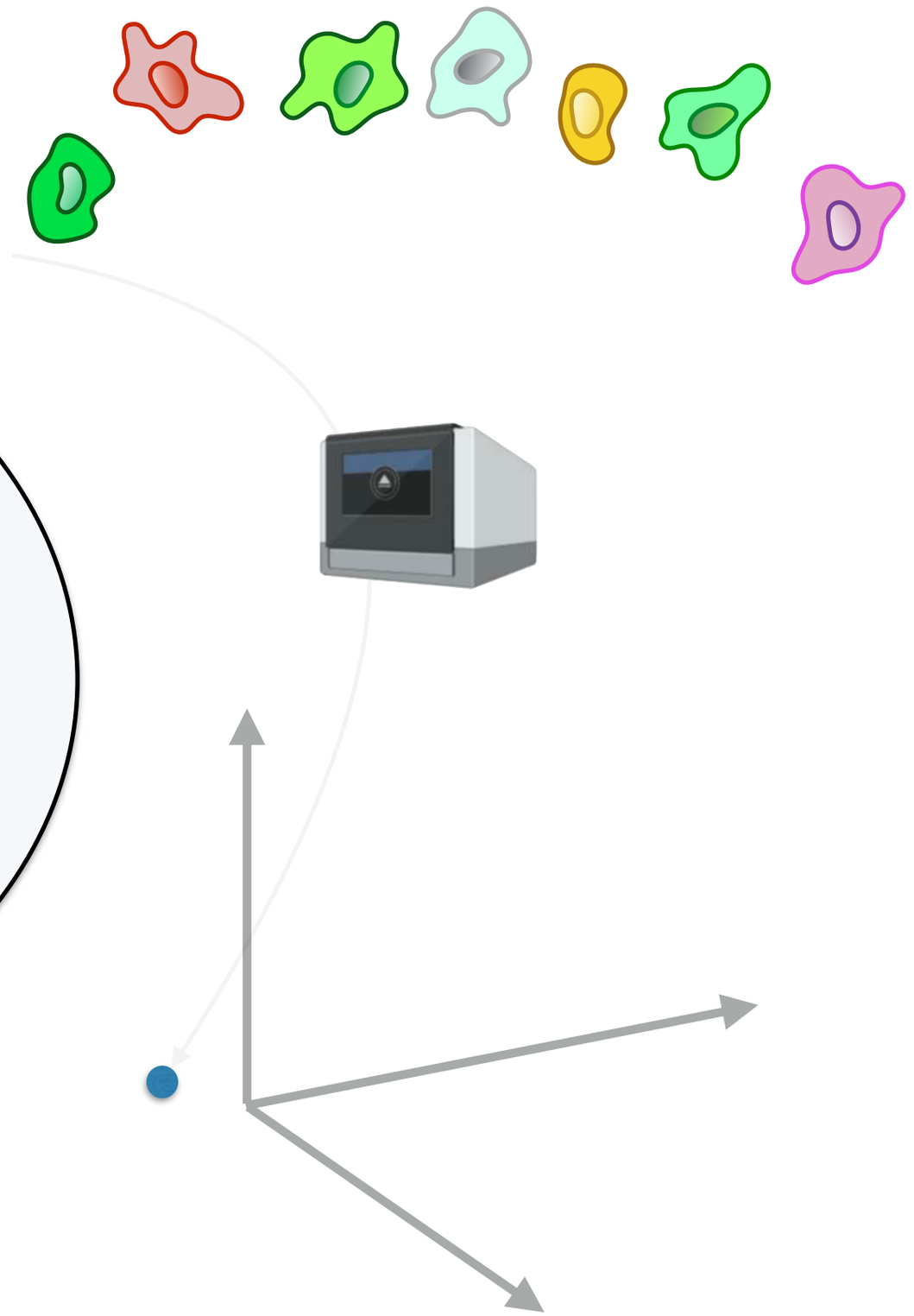




An Example from Single-Cell Genomics

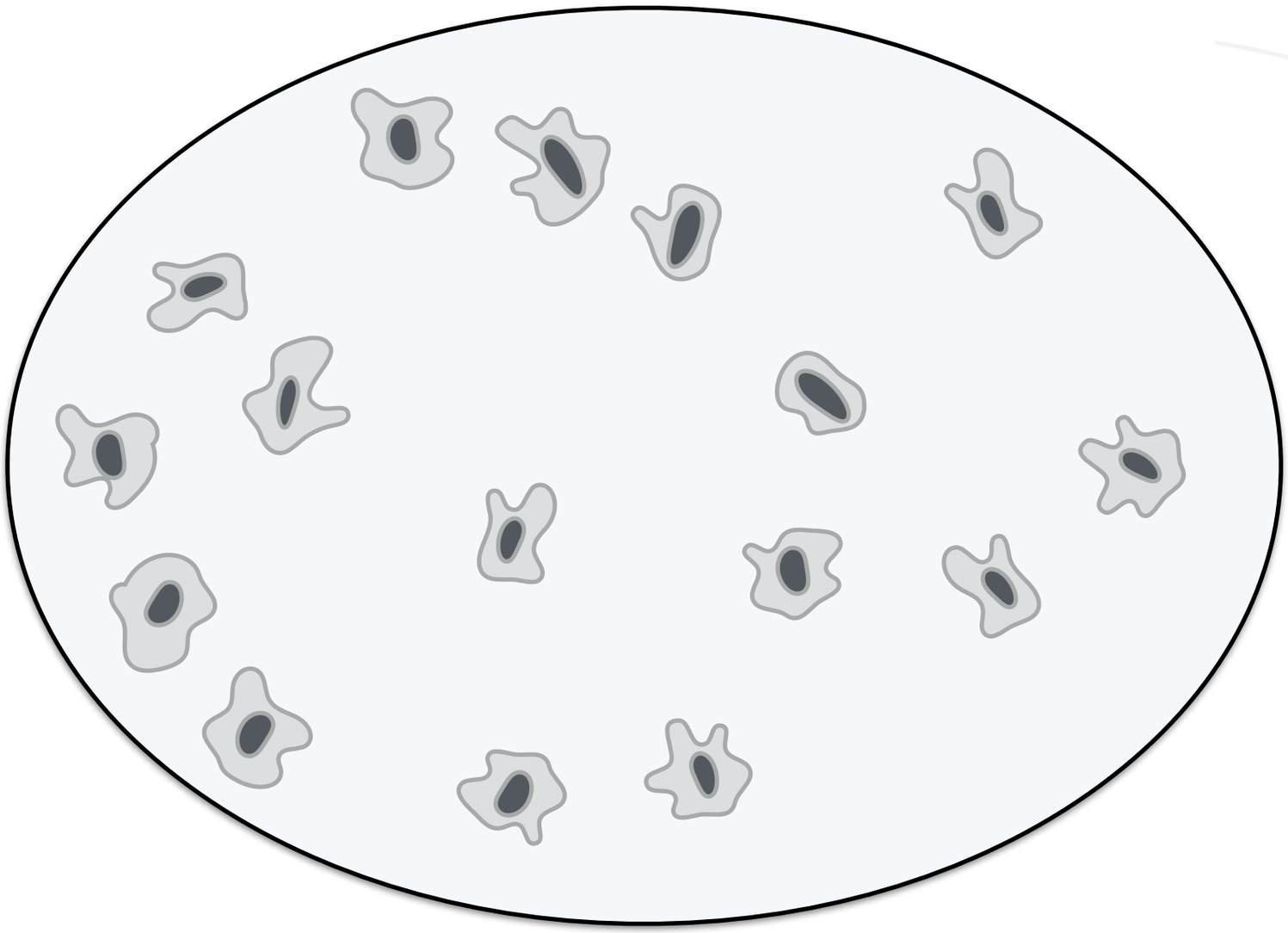
$t = 0$

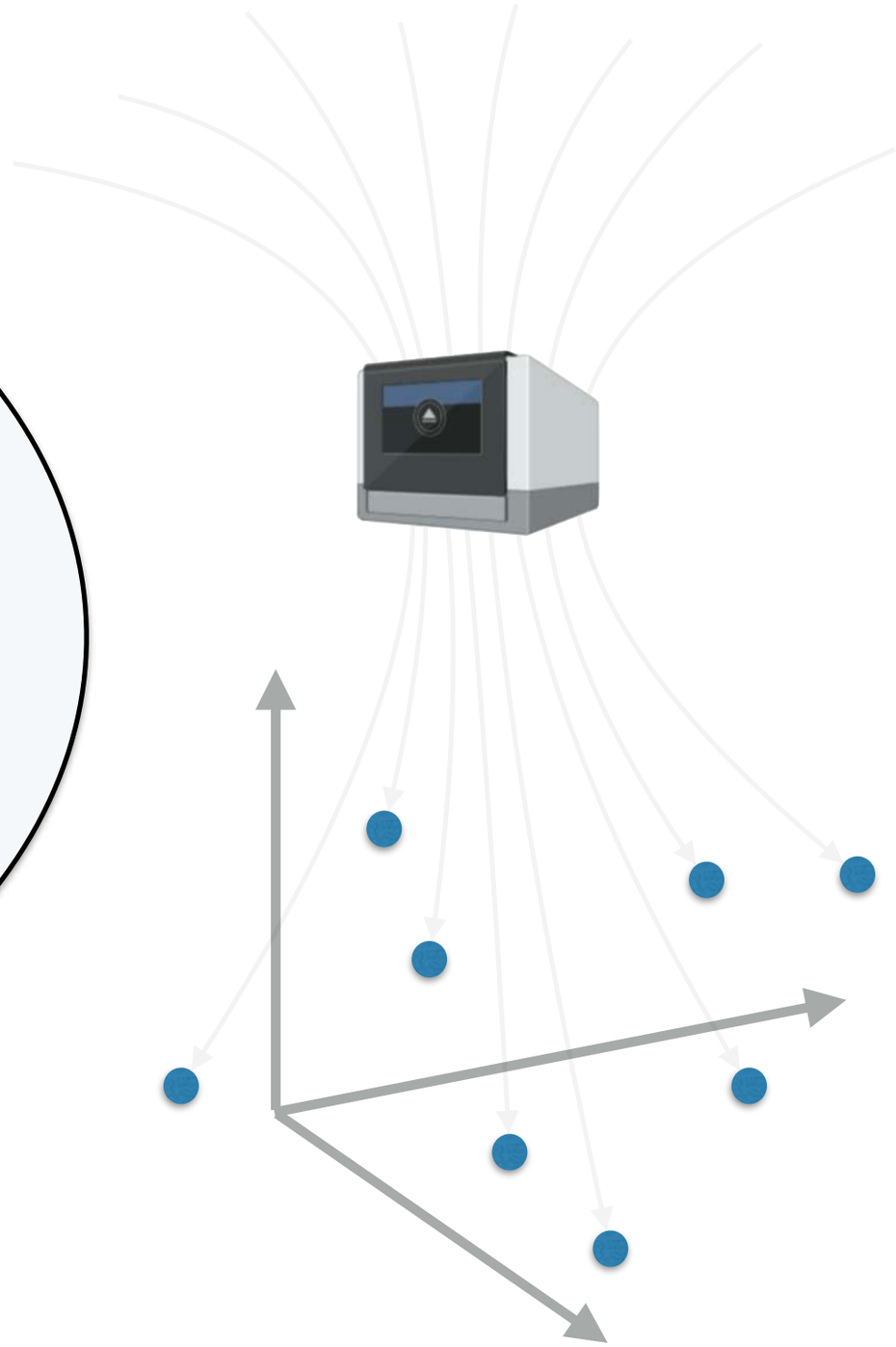




An Example from Single-Cell Genomics

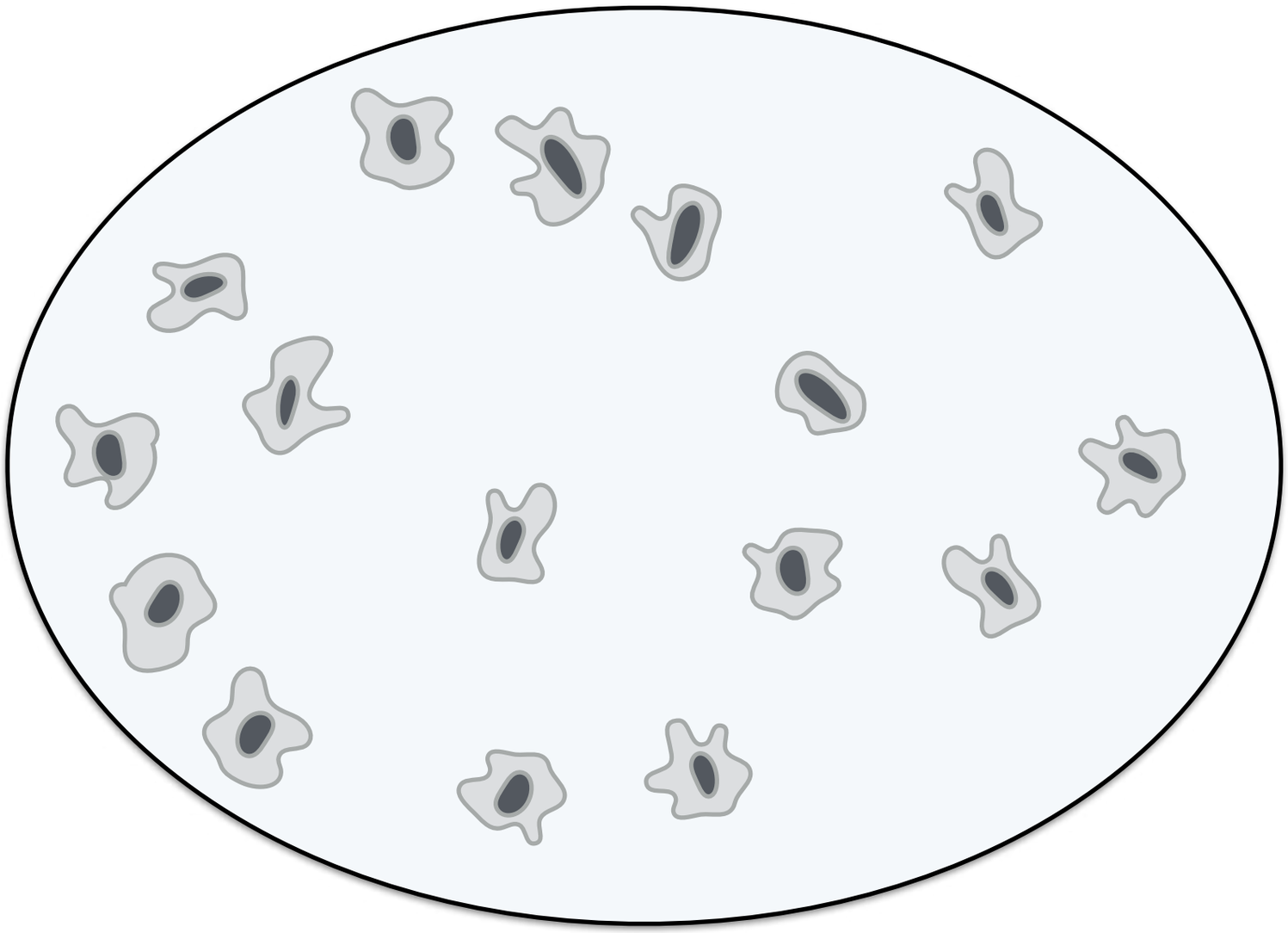
$t = 0$

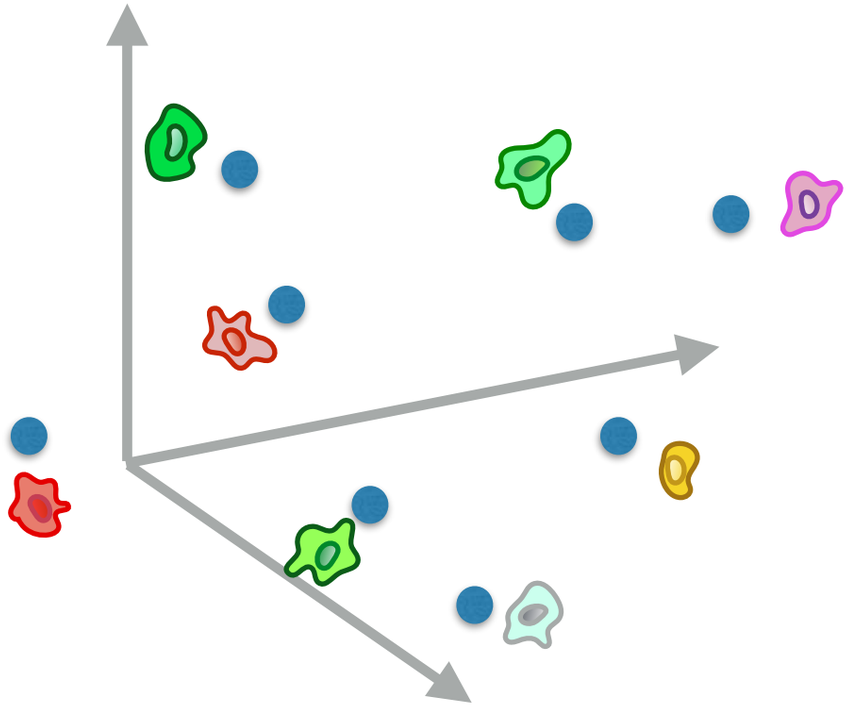




An Example from Single-Cell Genomics

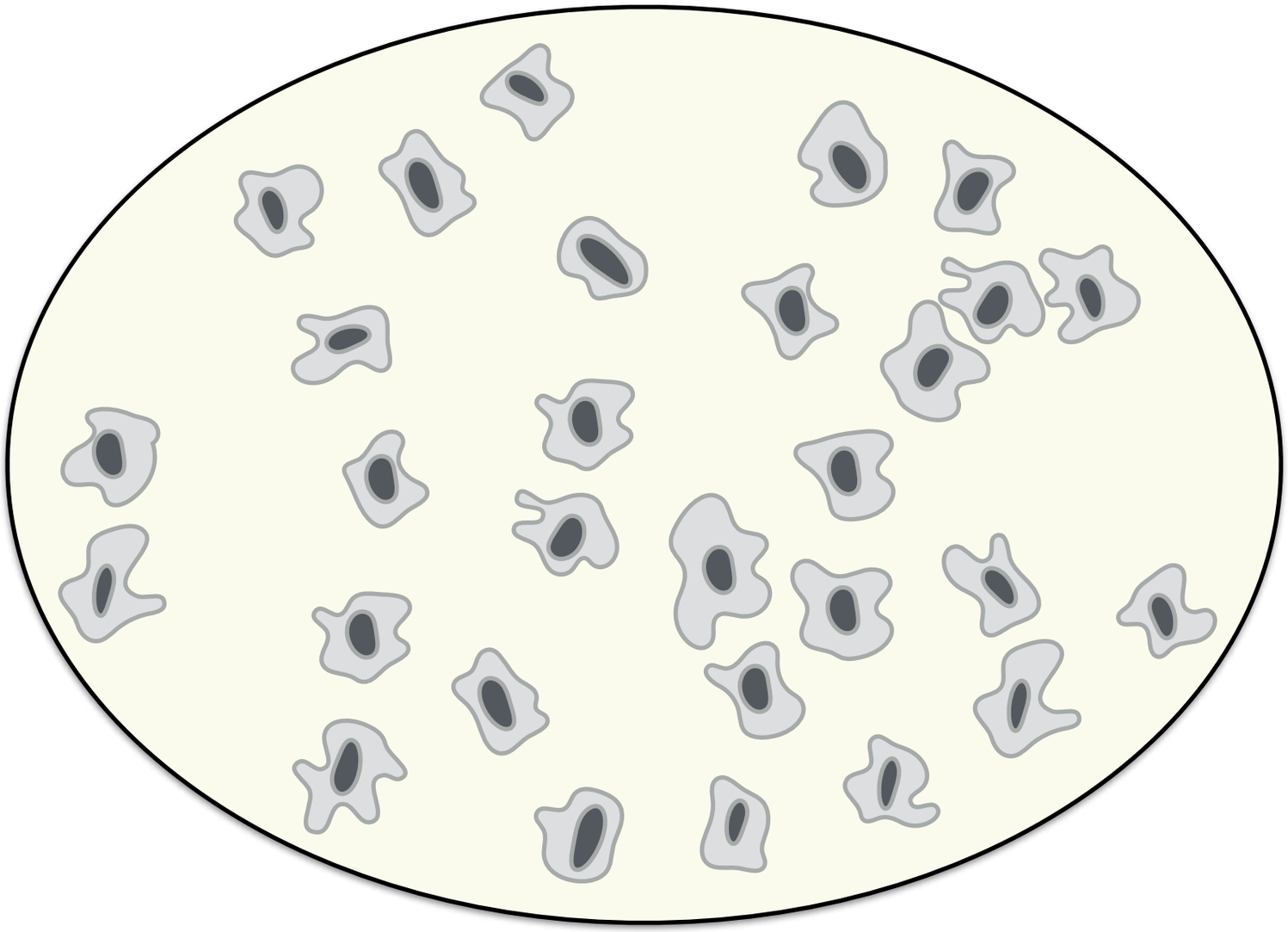
$t = 0$

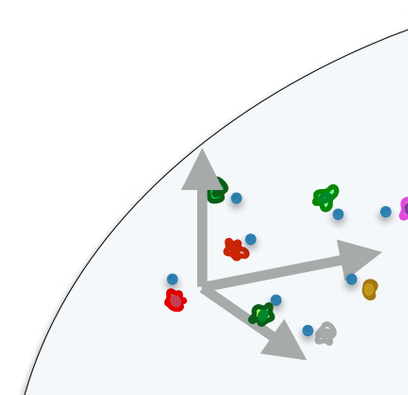




An Example from Single-Cell Genomics

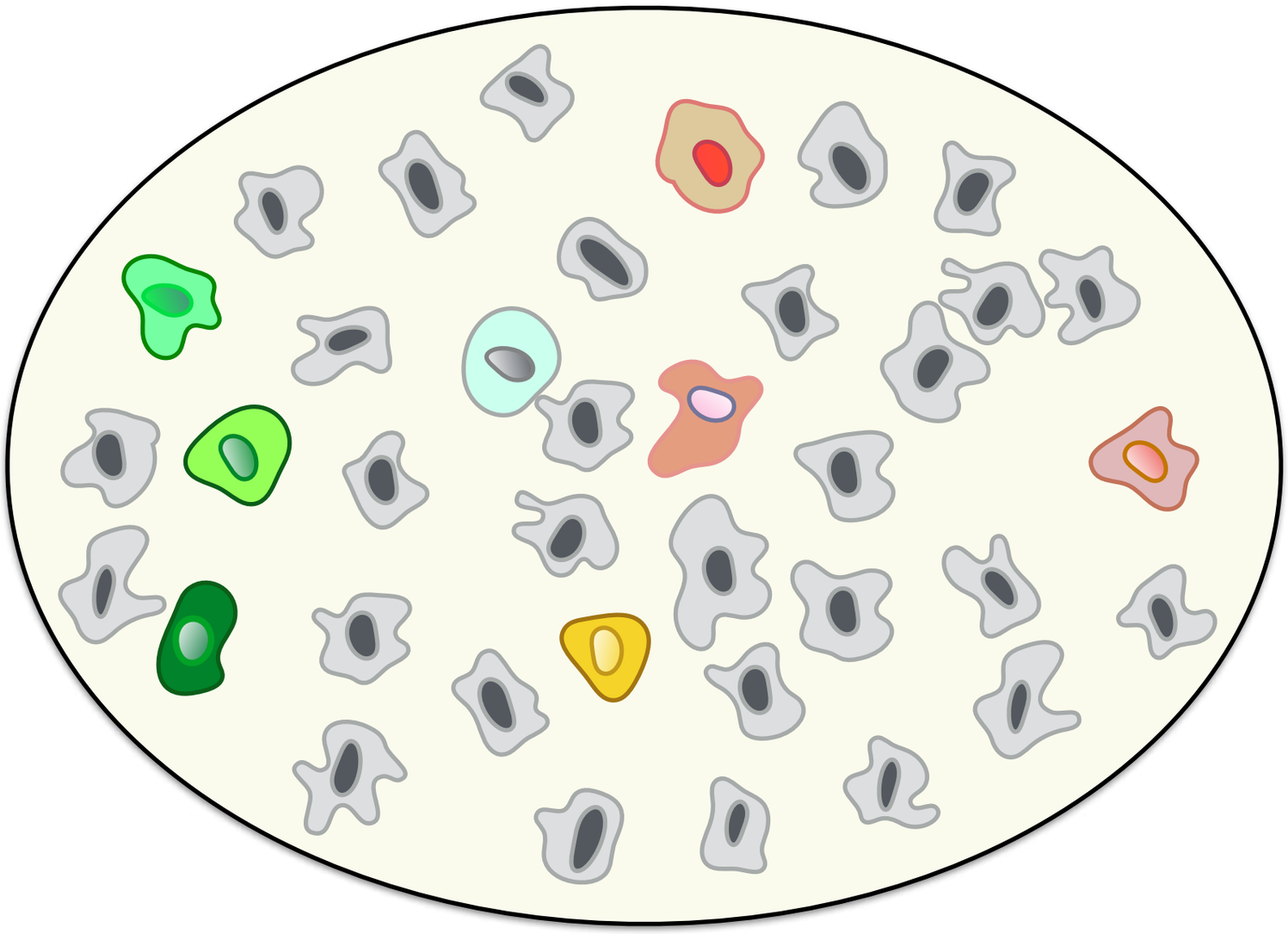
$$t = 6h$$

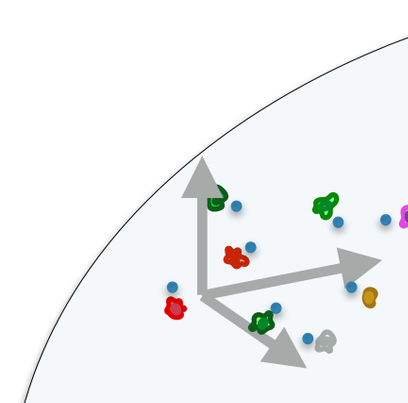




An Example from Single-Cell Genomics

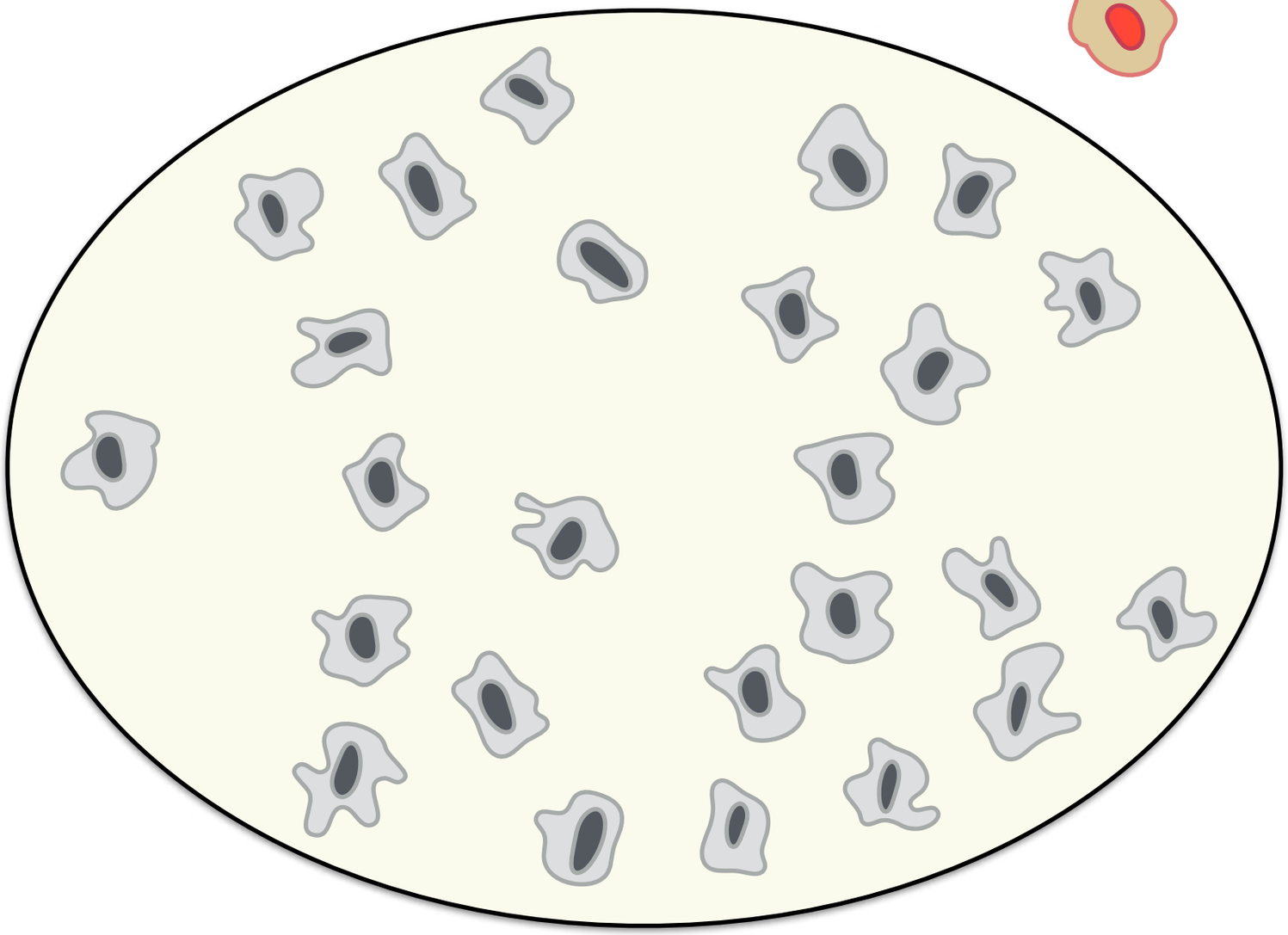
$$t = 6h$$

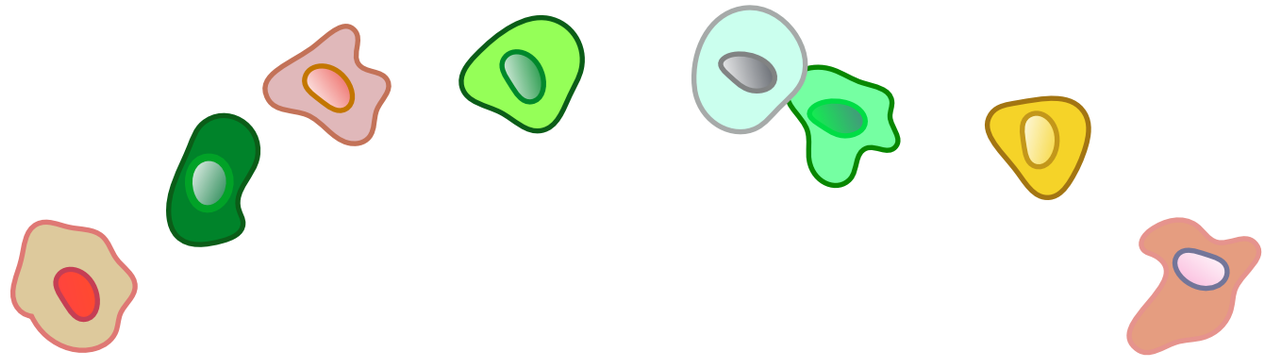


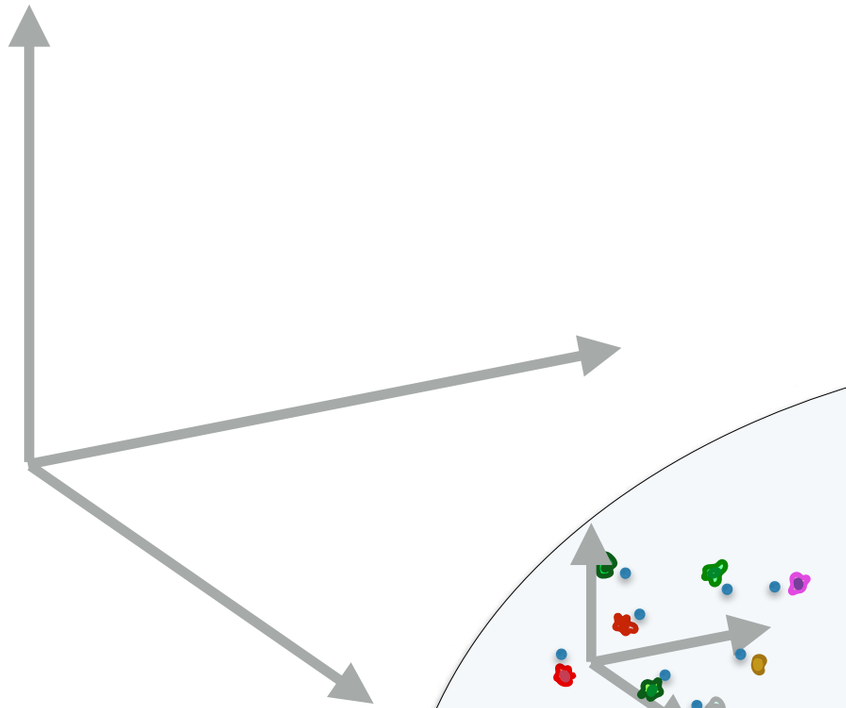


An Example from Single-Cell Genomics

$t = 6h$

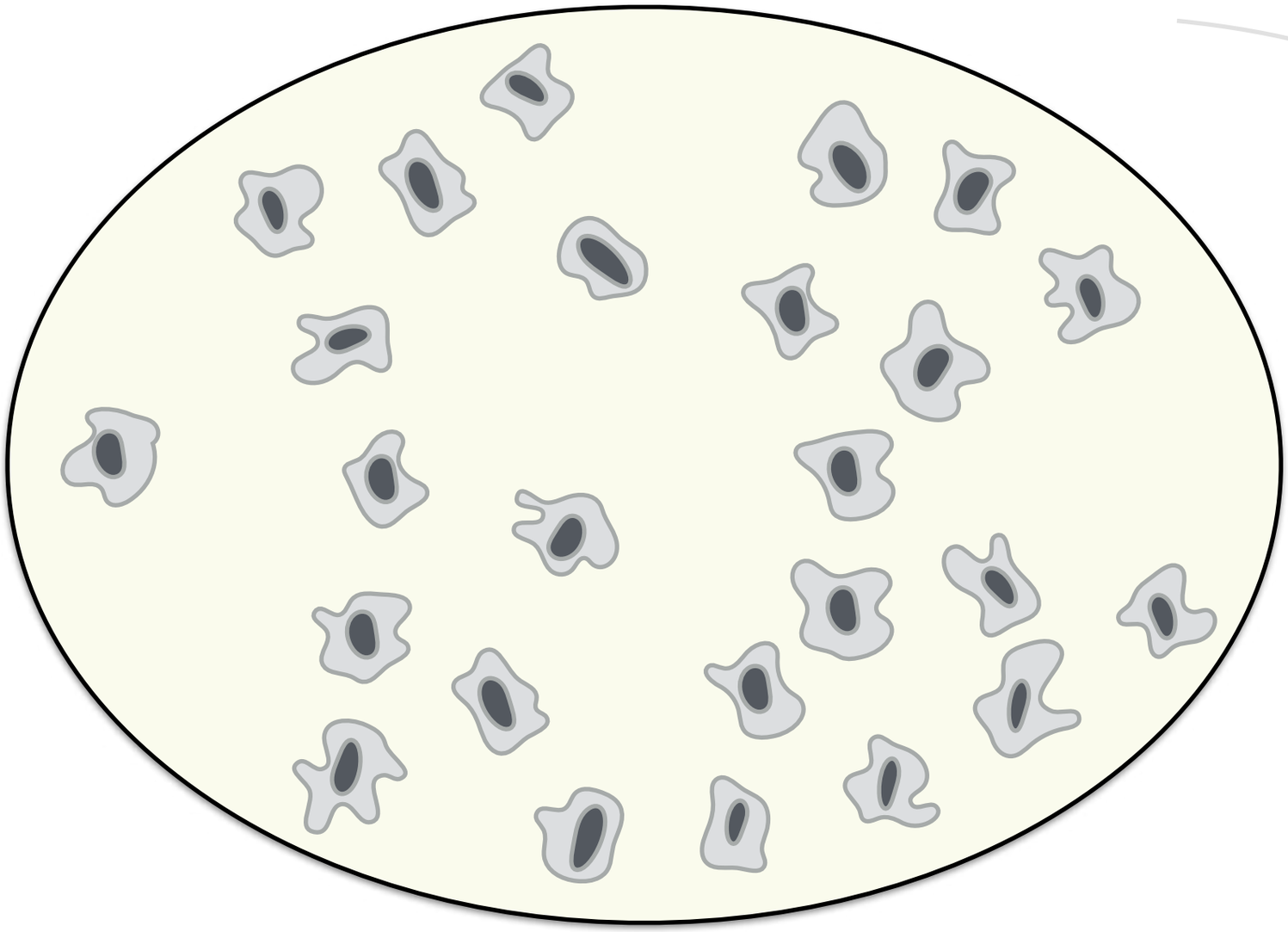


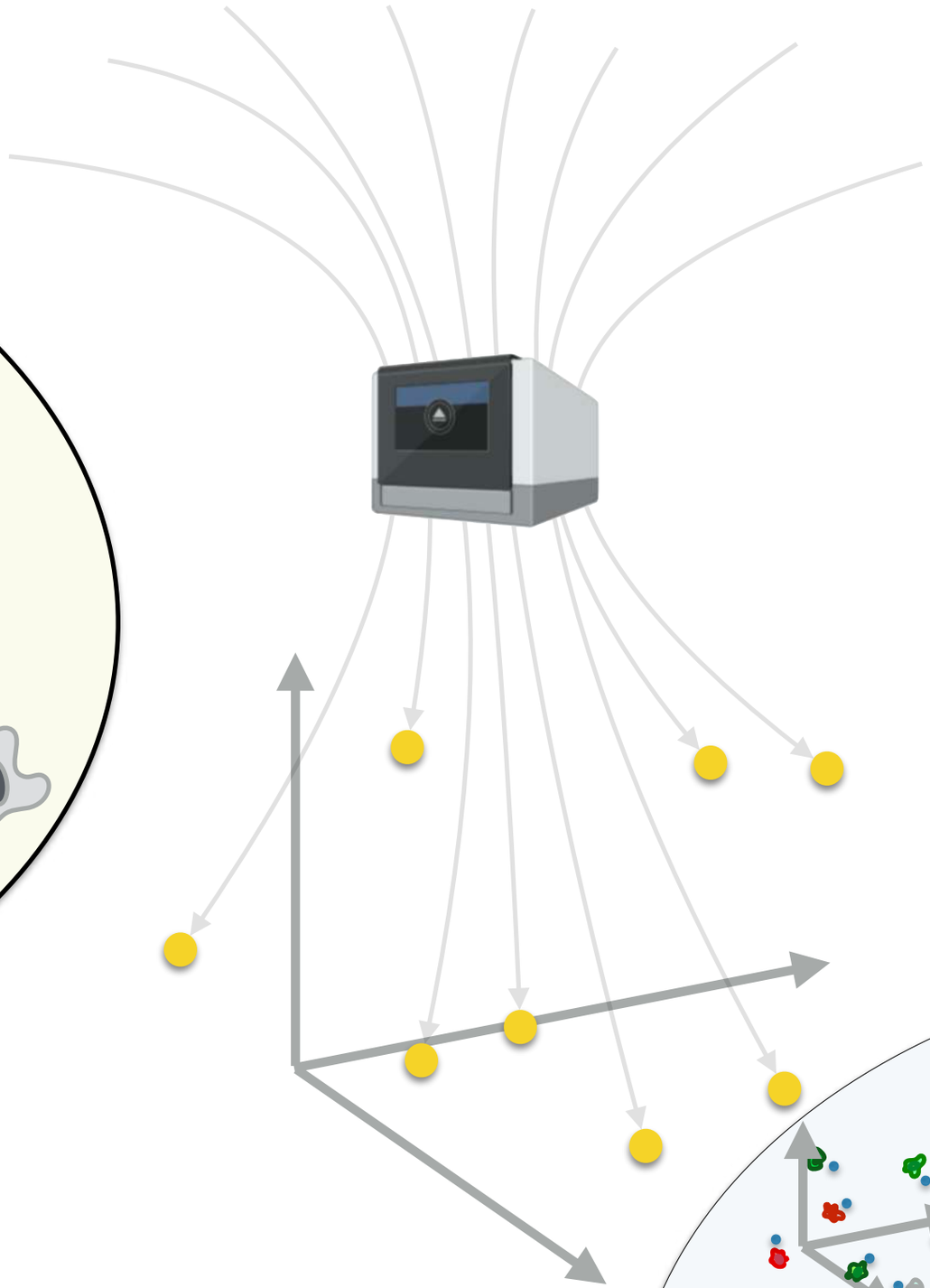




An Example from Single-Cell Genomics

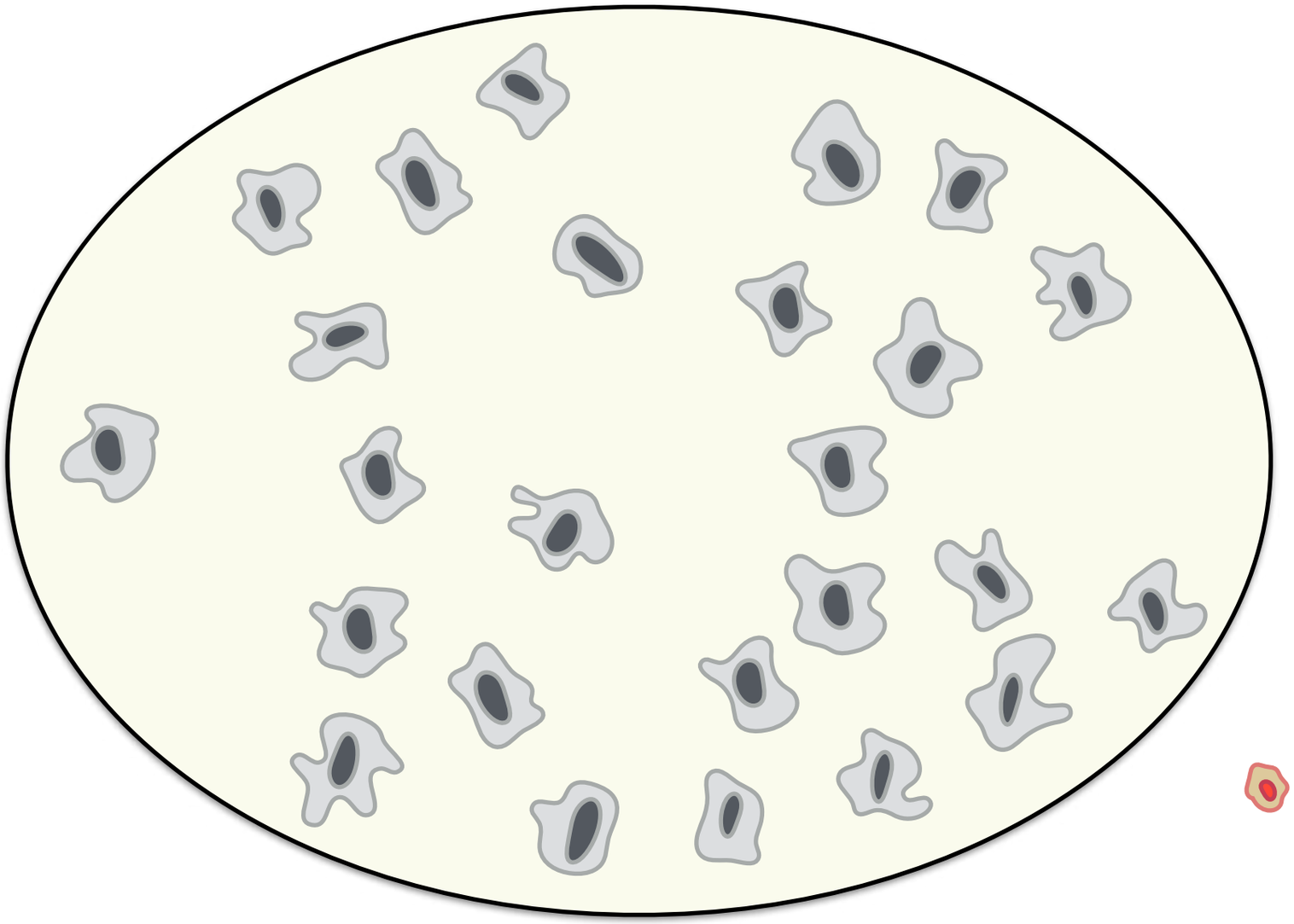
$$t = 6h$$

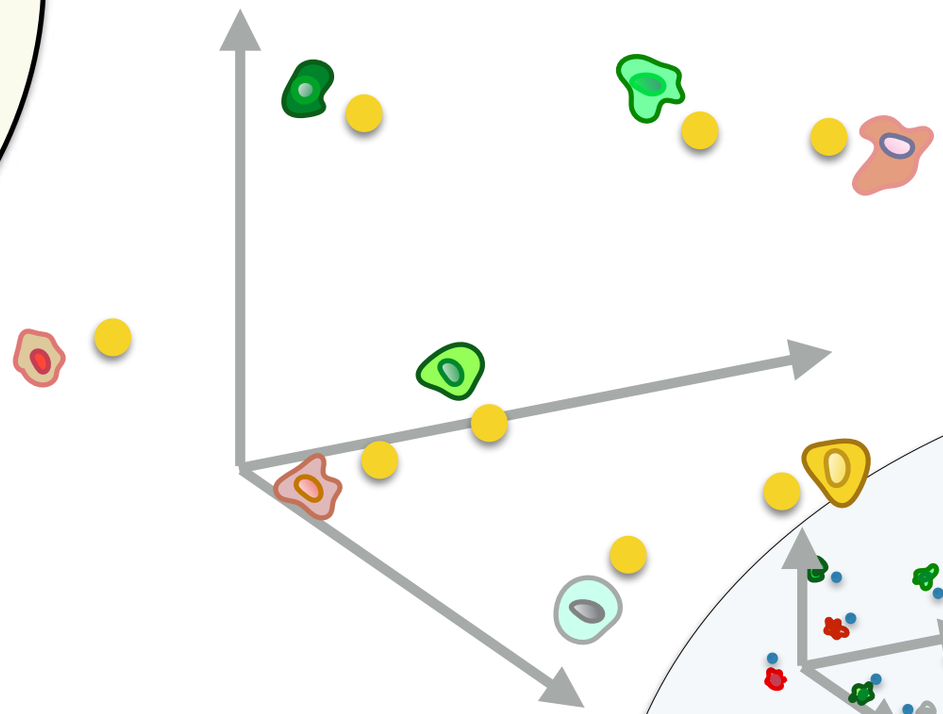




An Example from Single-Cell Genomics

$$t = 6h$$

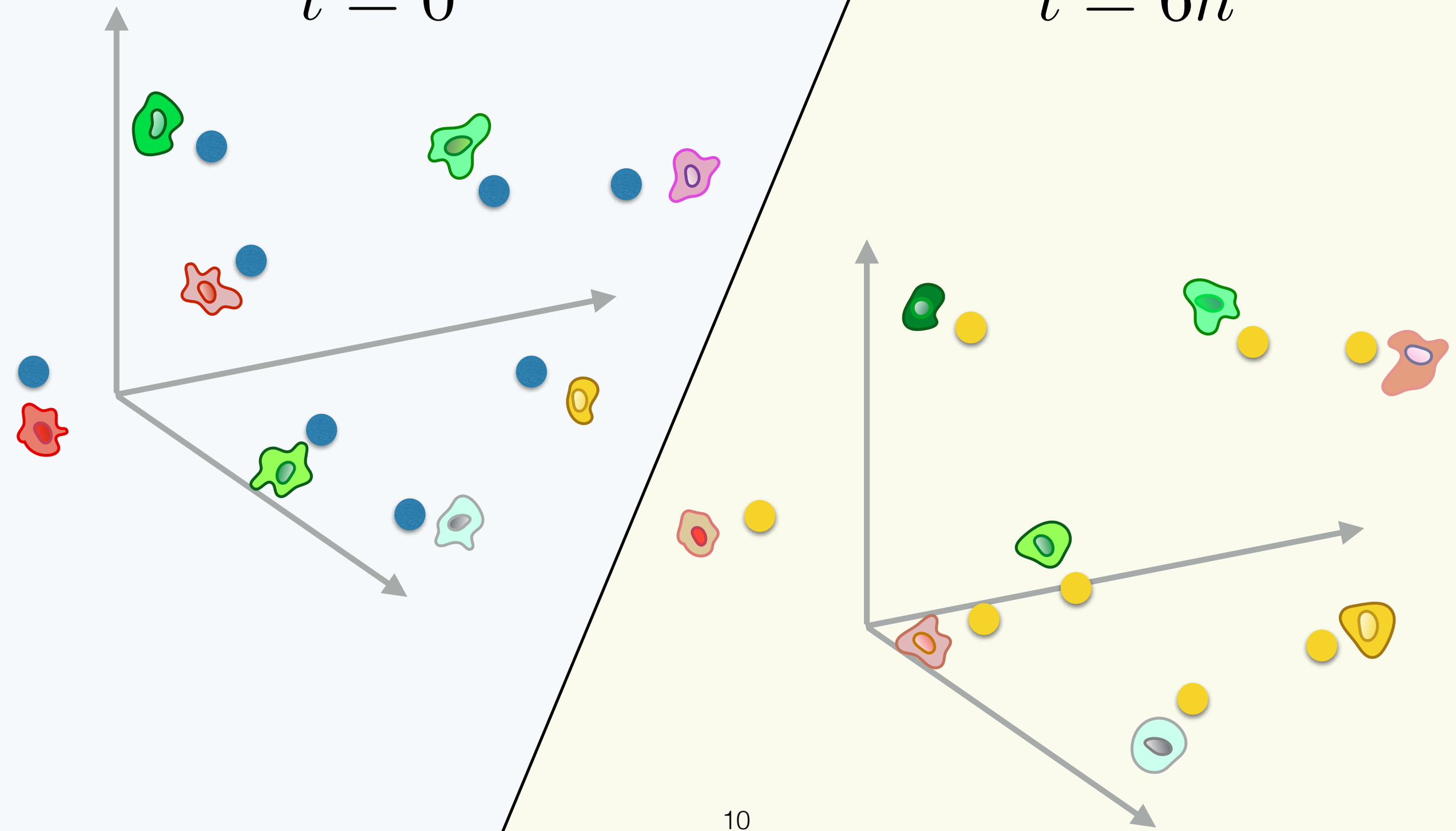




Comparing Two Populations

$t = 0$

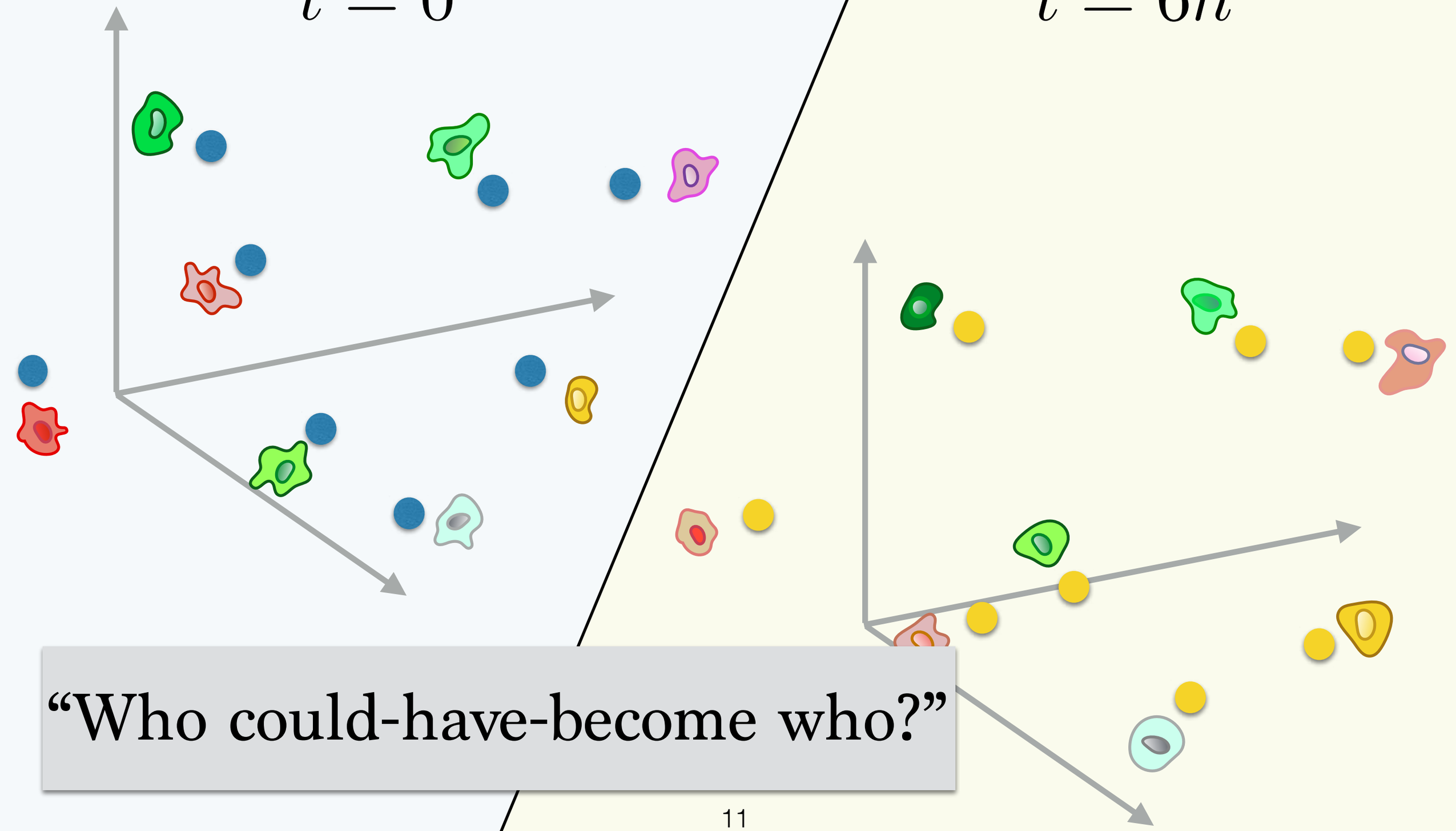
$t = 6h$



Comparing Two Populations

$t = 0$

$t = 6h$

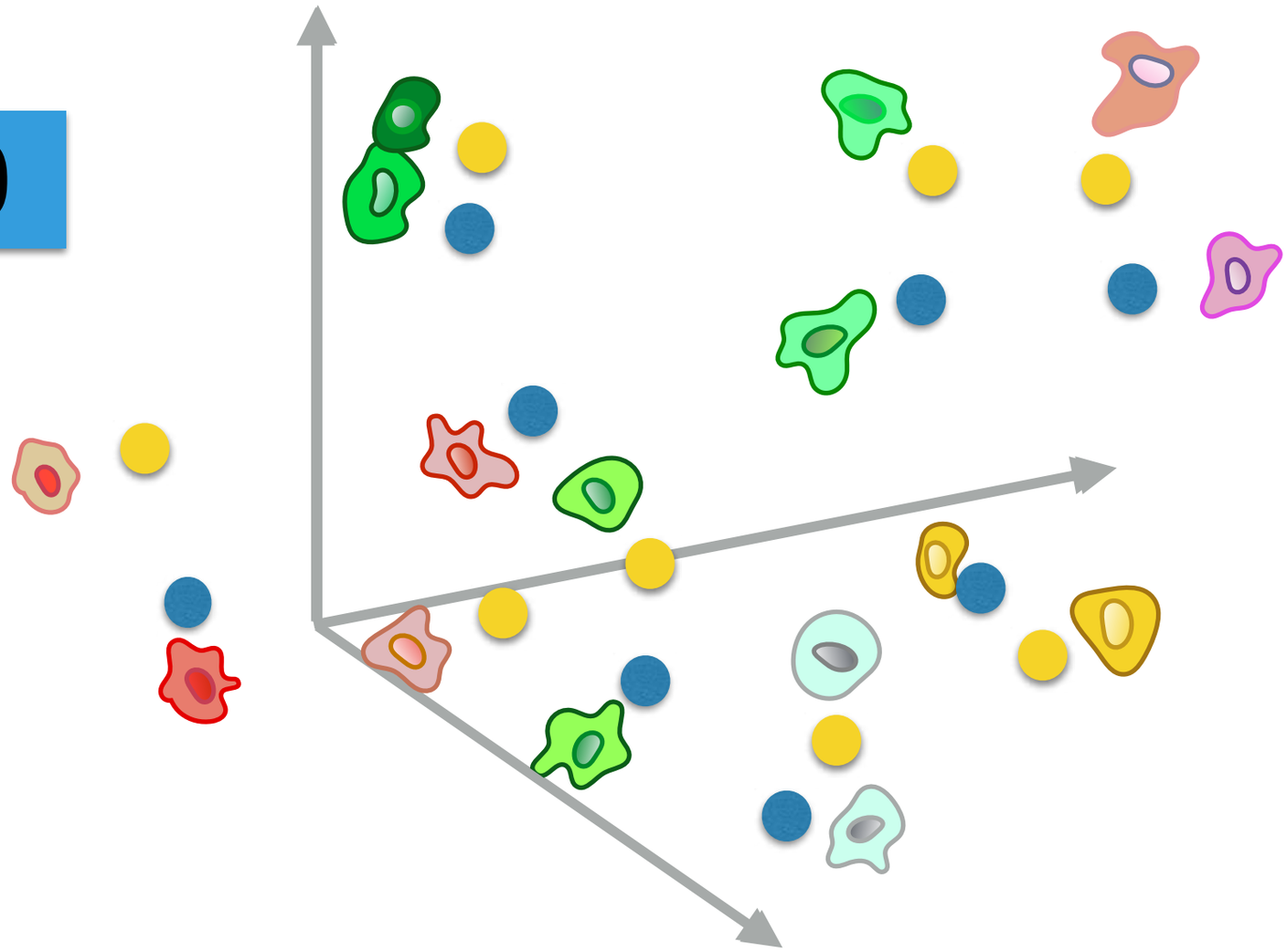


"Who could-have-become who?"

Matching Two Populations

to answer that question, use an **optimal matching***

$t = 0$

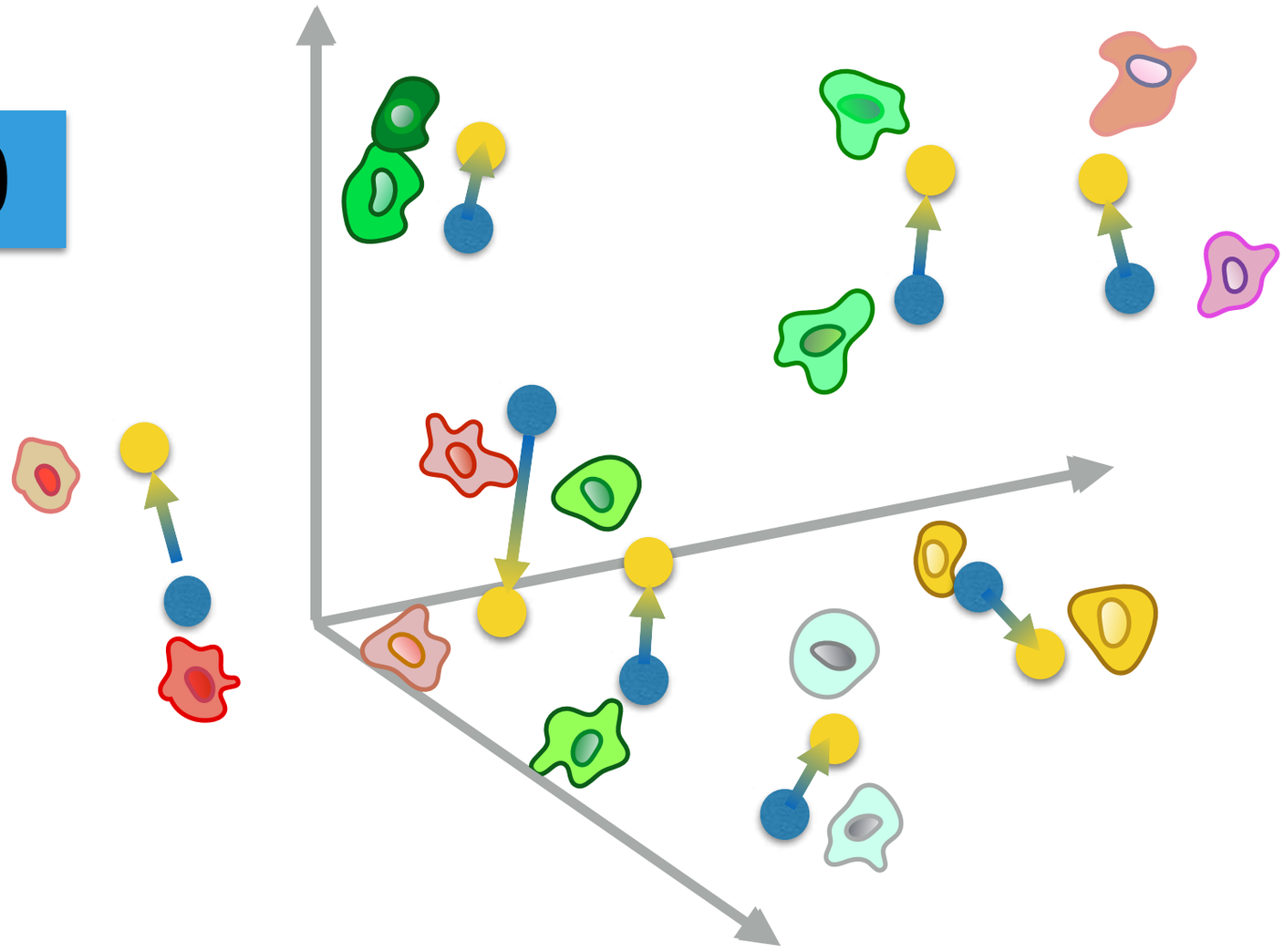


$t = 6h$

Matching Two Populations

to answer that question, use an **optimal matching***

$t = 0$

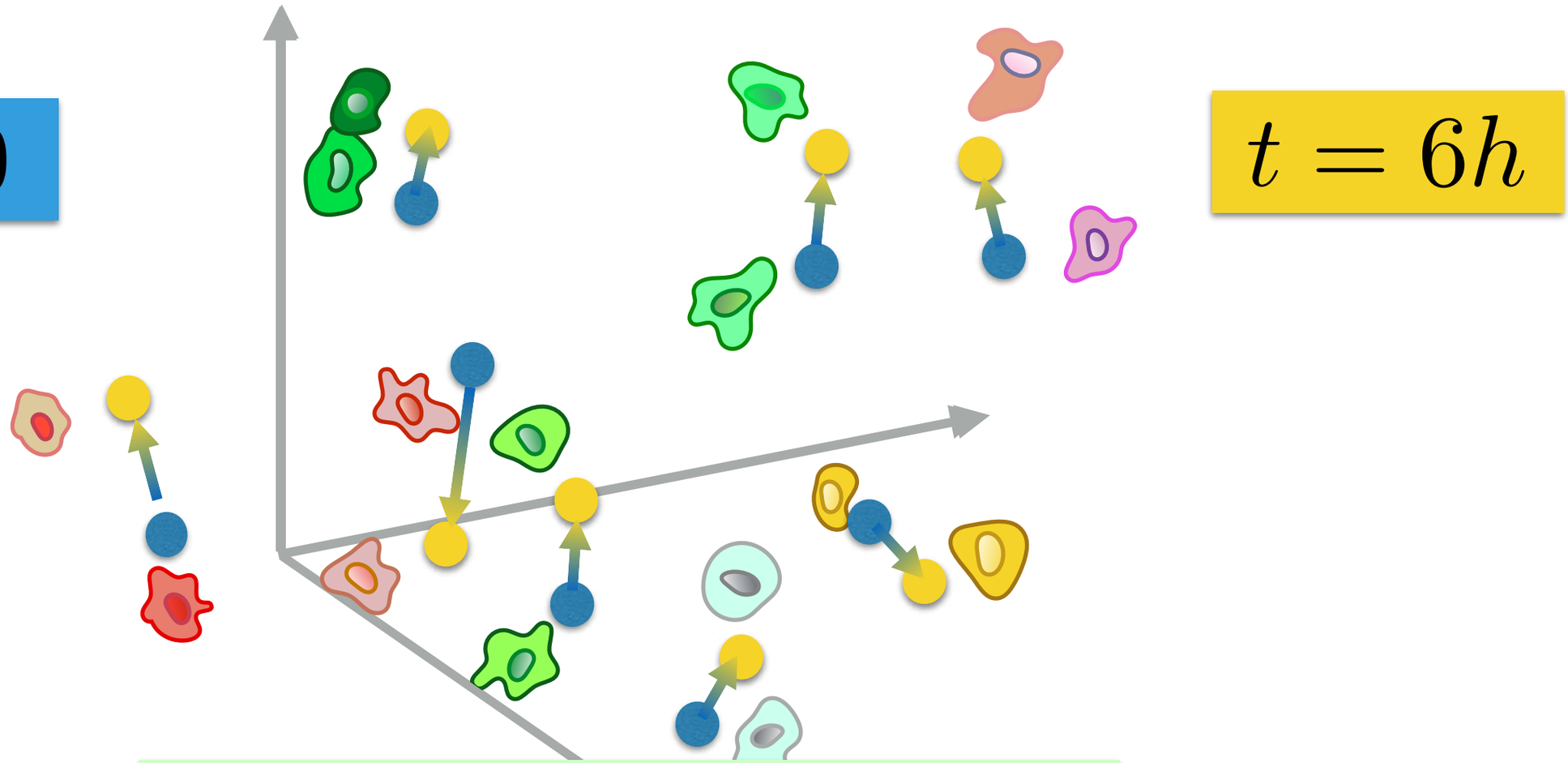


$t = 6h$

Matching Two Populations

to answer that question, use an **optimal matching***

$t = 0$



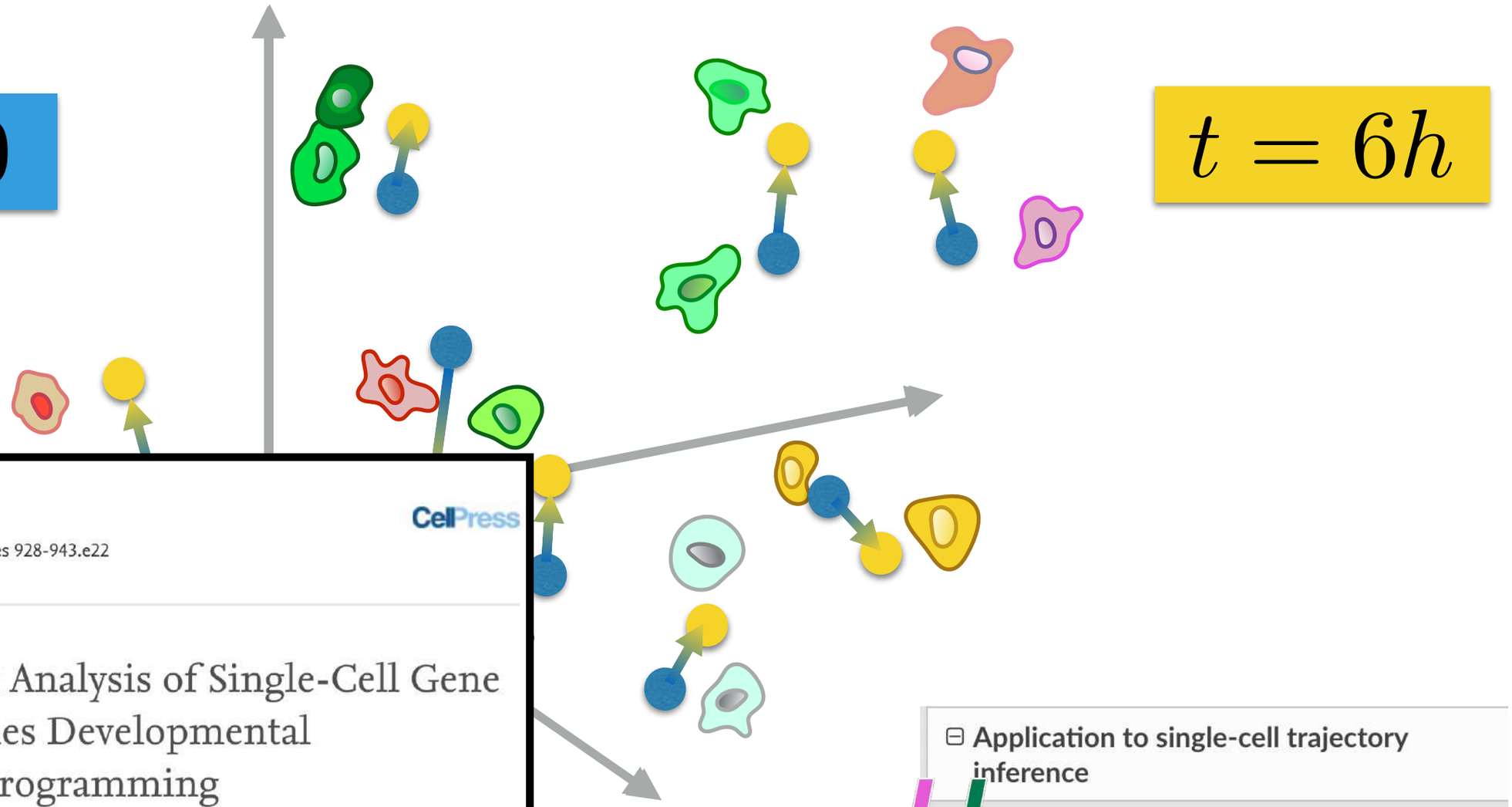
$t = 6h$

* $$\boldsymbol{\sigma}^\star = \underset{\boldsymbol{\sigma} \in \mathcal{S}(n)}{\operatorname{argmin}} \sum_{i=1}^{n} d(x_i, y_{\boldsymbol{\sigma}_i})$$

Matching Two Populations

to answer that question, use an **optimal matching***

$t = 0$



$t = 6h$

Cell

CellPress

Volume 176, Issue 4, 7 February 2019, Pages 928-943.e22

Resource

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

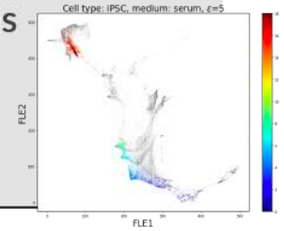
Geoffrey Schiebinger [1, 11, 16], Jian Shu [1, 2, 16] ✉, Marcin Tabaka [1, 16], Brian Cleary [1, 3, 16], Vidya Subramanian [1], Aryeh Solomon [1, 17], Joshua Gould [1], Siyan Liu [1, 15], Stacie Lin [1, 6], Peter Berube [1], Lia Lee [1], Jenny Chen [1, 4], Justin Brumbaugh [5, 7, 8, 9, 10], Philippe Rigollet [11, 12], Konrad Hochedlinger [7, 8, 9, 13], Rudolf Jaenisch [2, 3], Aviv Regev [1, 6, 13] ✉, Eric S. Lander [1, 6, 14, 18] ✉

Application to single-cell trajectory inference

Calculates optimal transport on gene expression point clouds

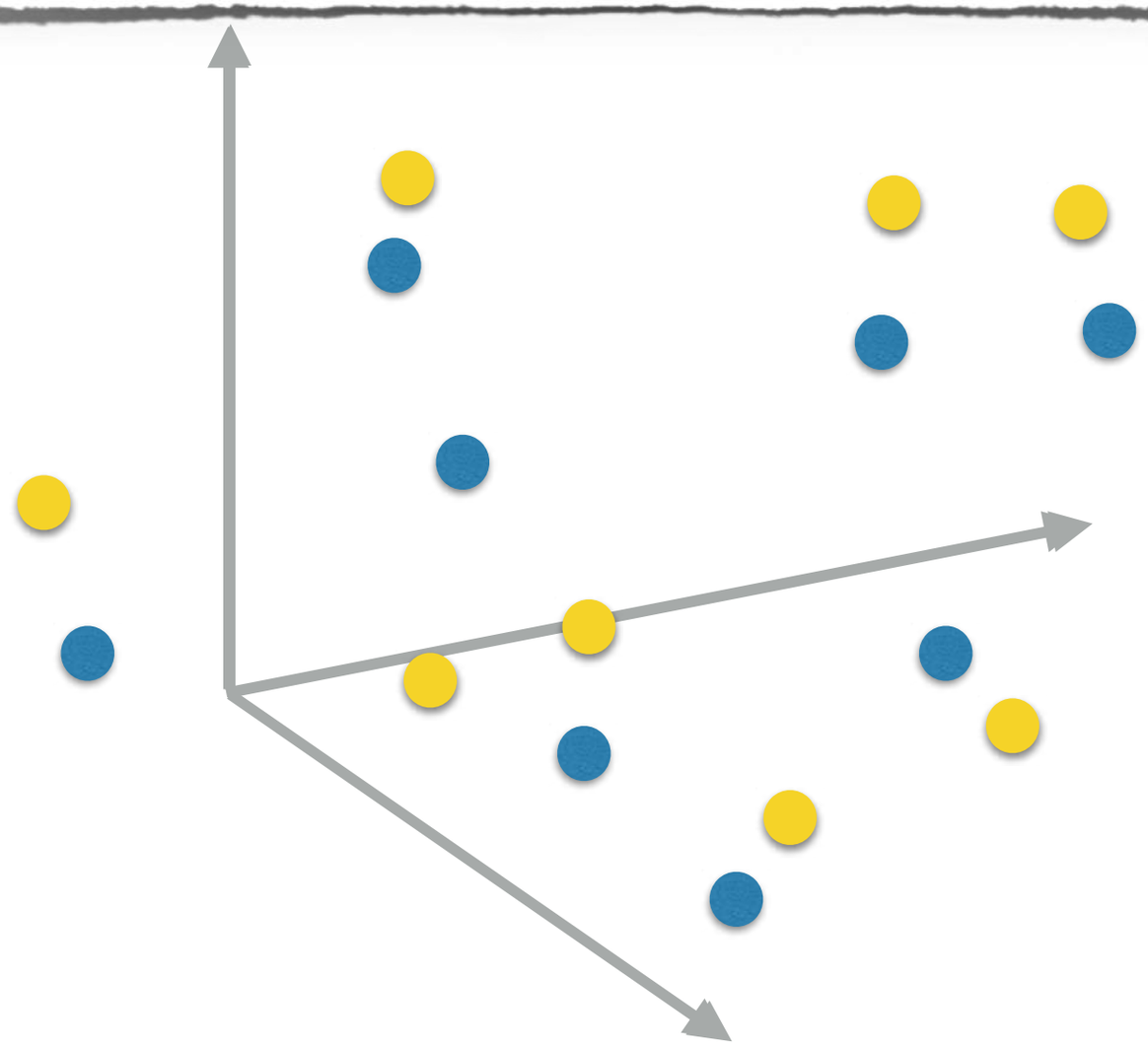
Infers cells' ancestors

Visualizes



From there, many generalisations

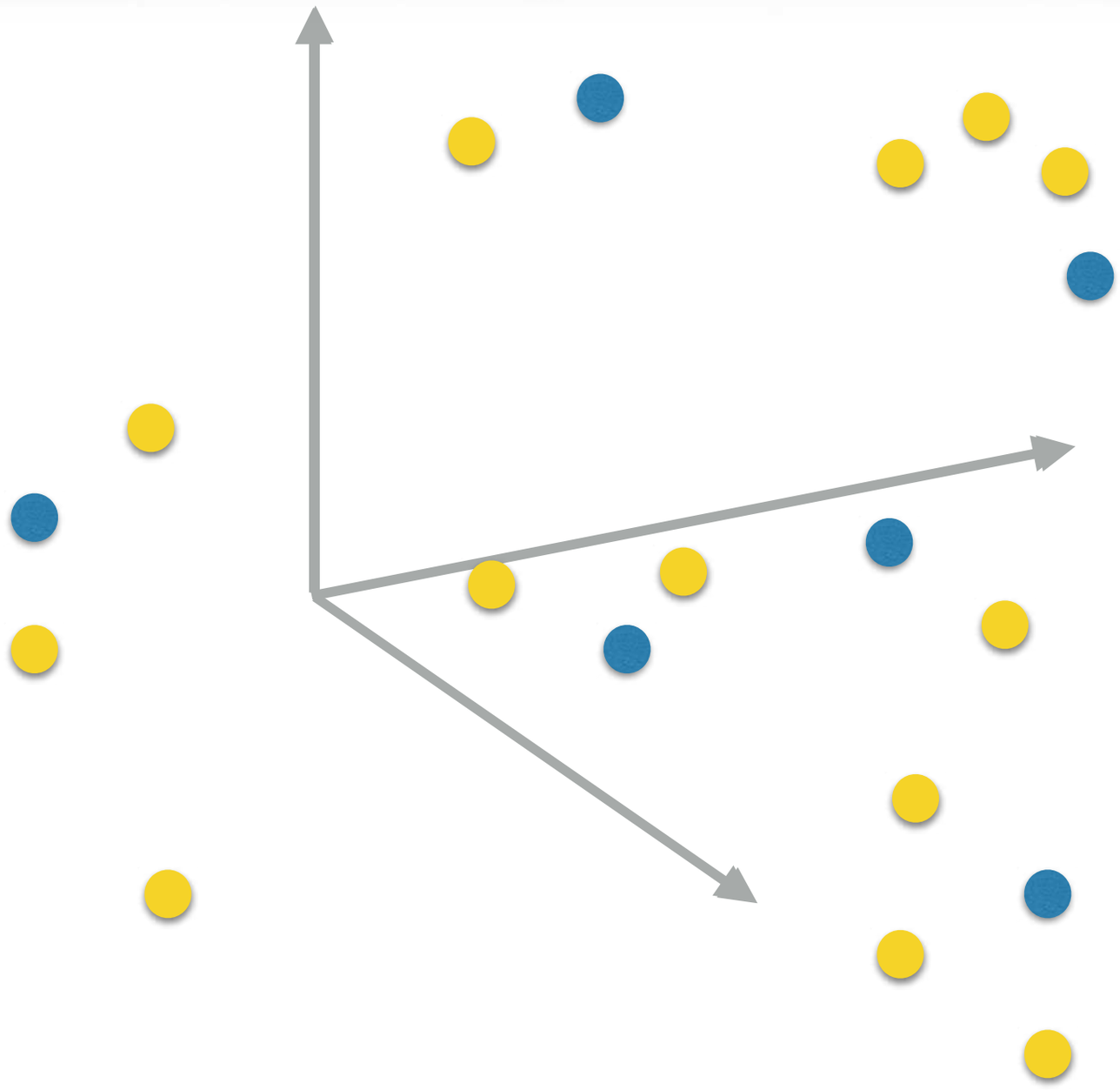
This story describes an ideal experimental setting, notably because this requires the same number of cells.



Challenges Down the Road

More likely to happen: *variable number of cells.*

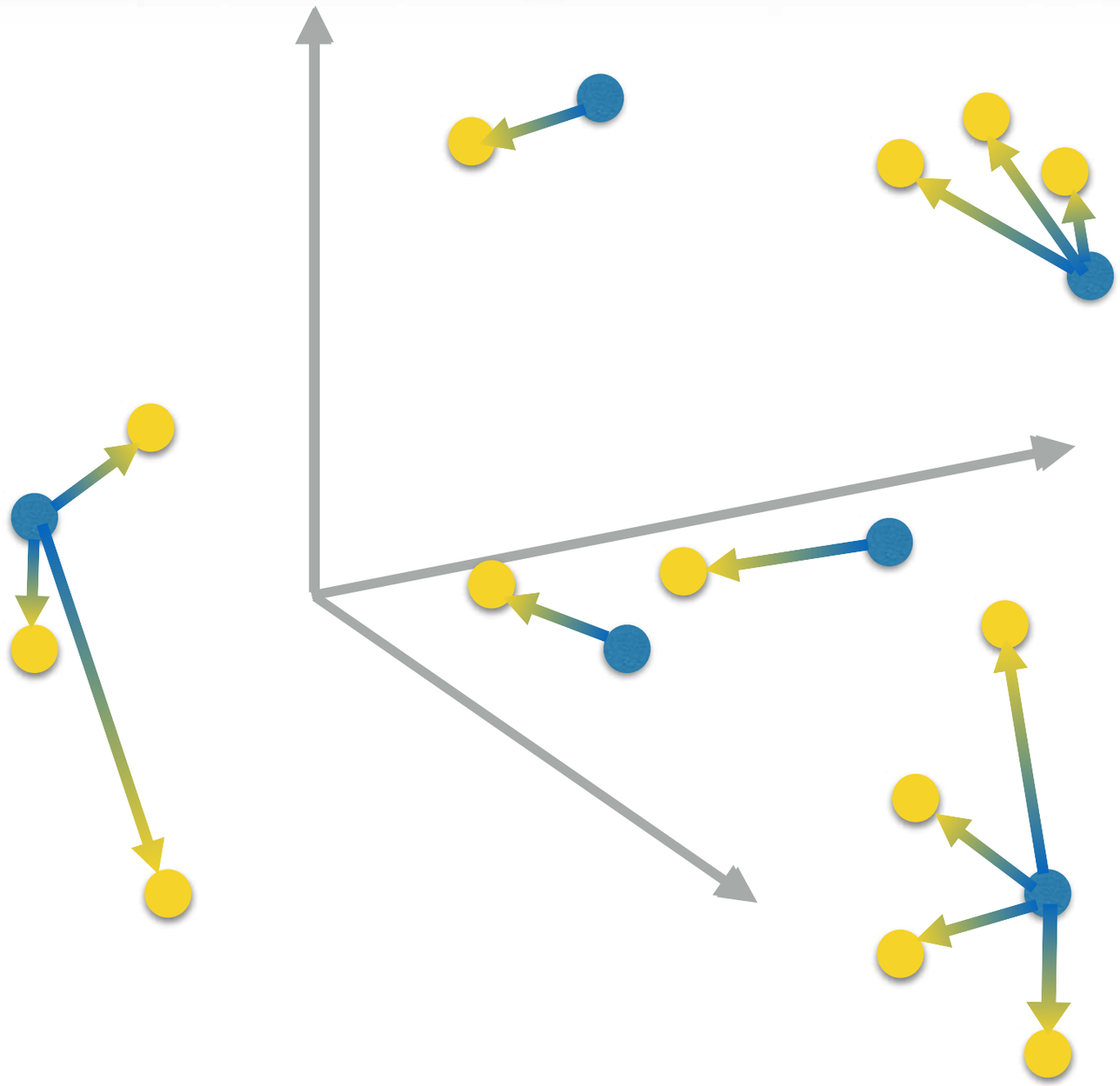
This requires a more versatile notion of matching,



Challenges Down the Road

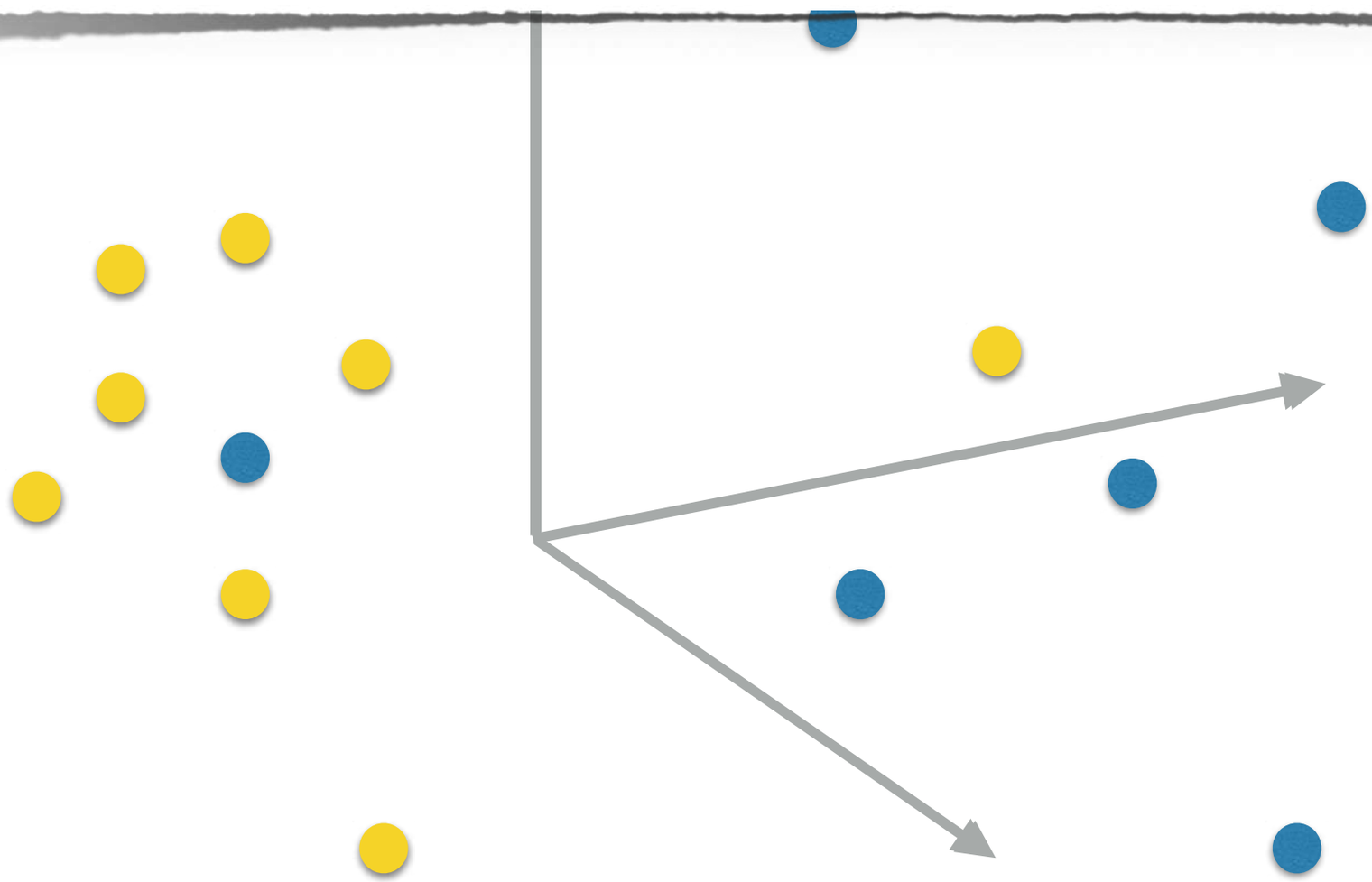
More likely to happen: *variable number of cells.*

This requires a more versatile notion of matching,



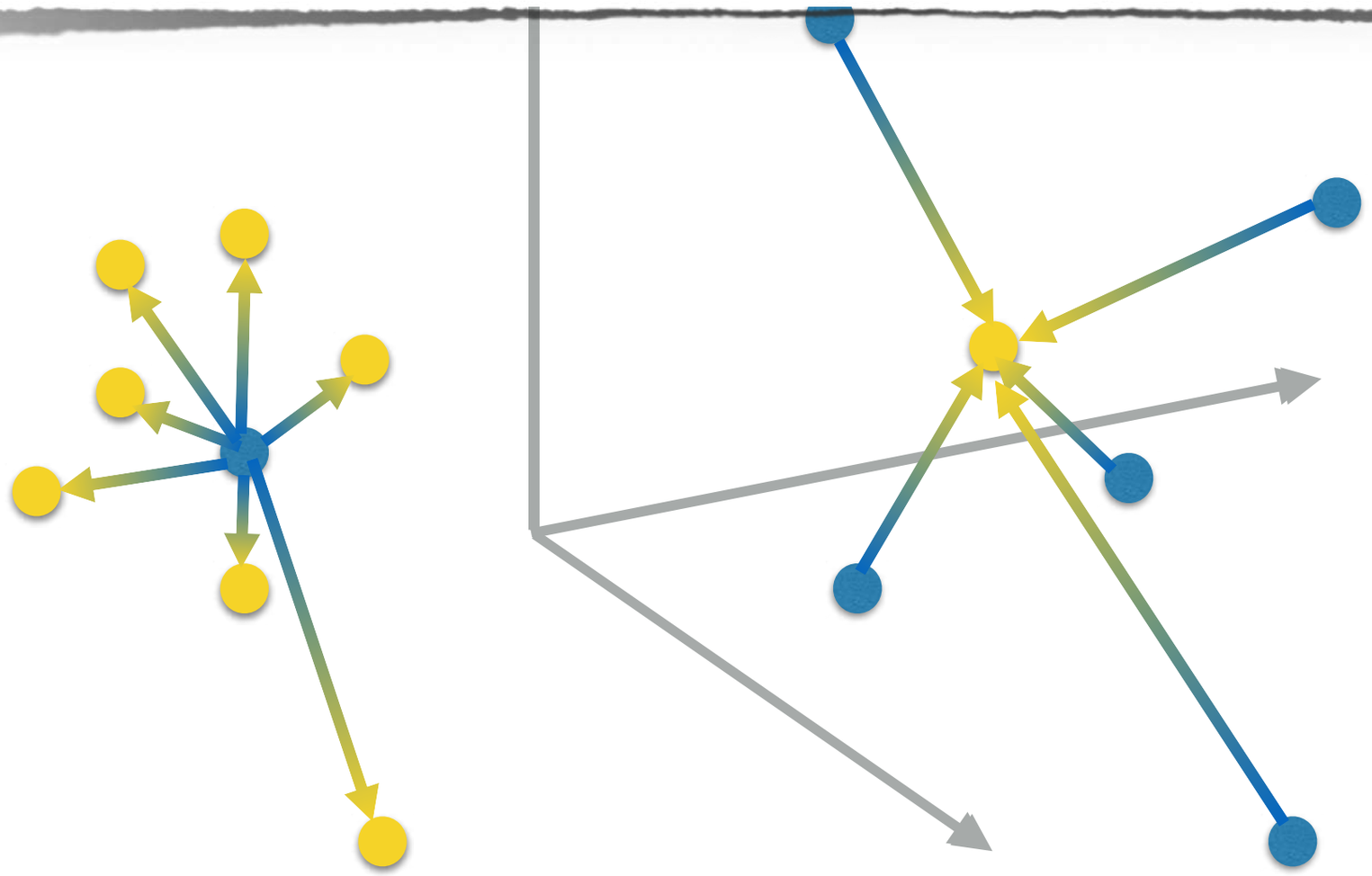
Challenges Down the Road

More likely to happen: *variable number of cells.*
This requires a more versatile notion of matching,
one that is more robust than simple nearest-neighbor.



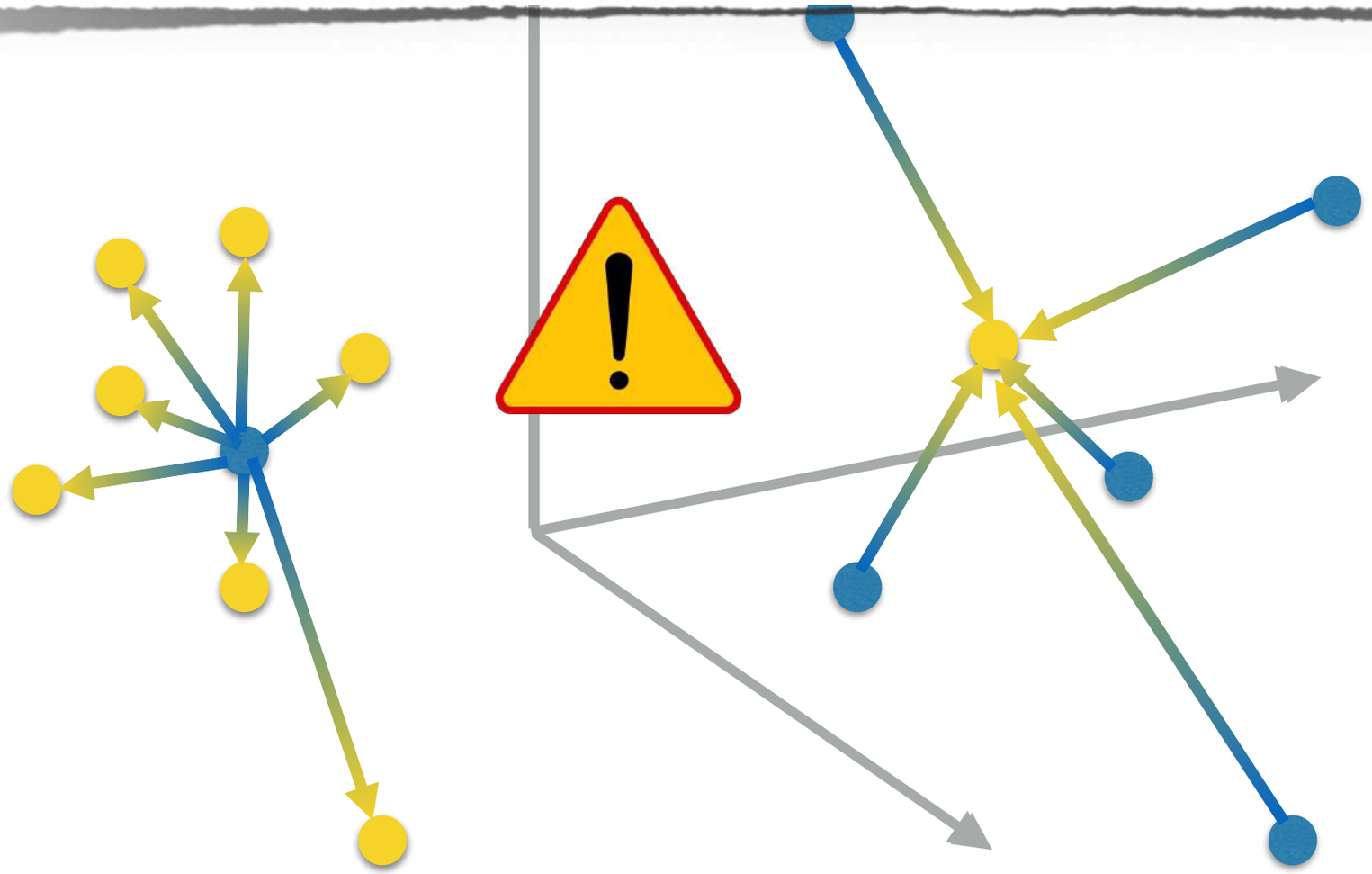
Challenges Down the Road

More likely to happen: *variable number of cells.*
This requires a more versatile notion of matching,
one that is more robust than simple nearest-neighbor.



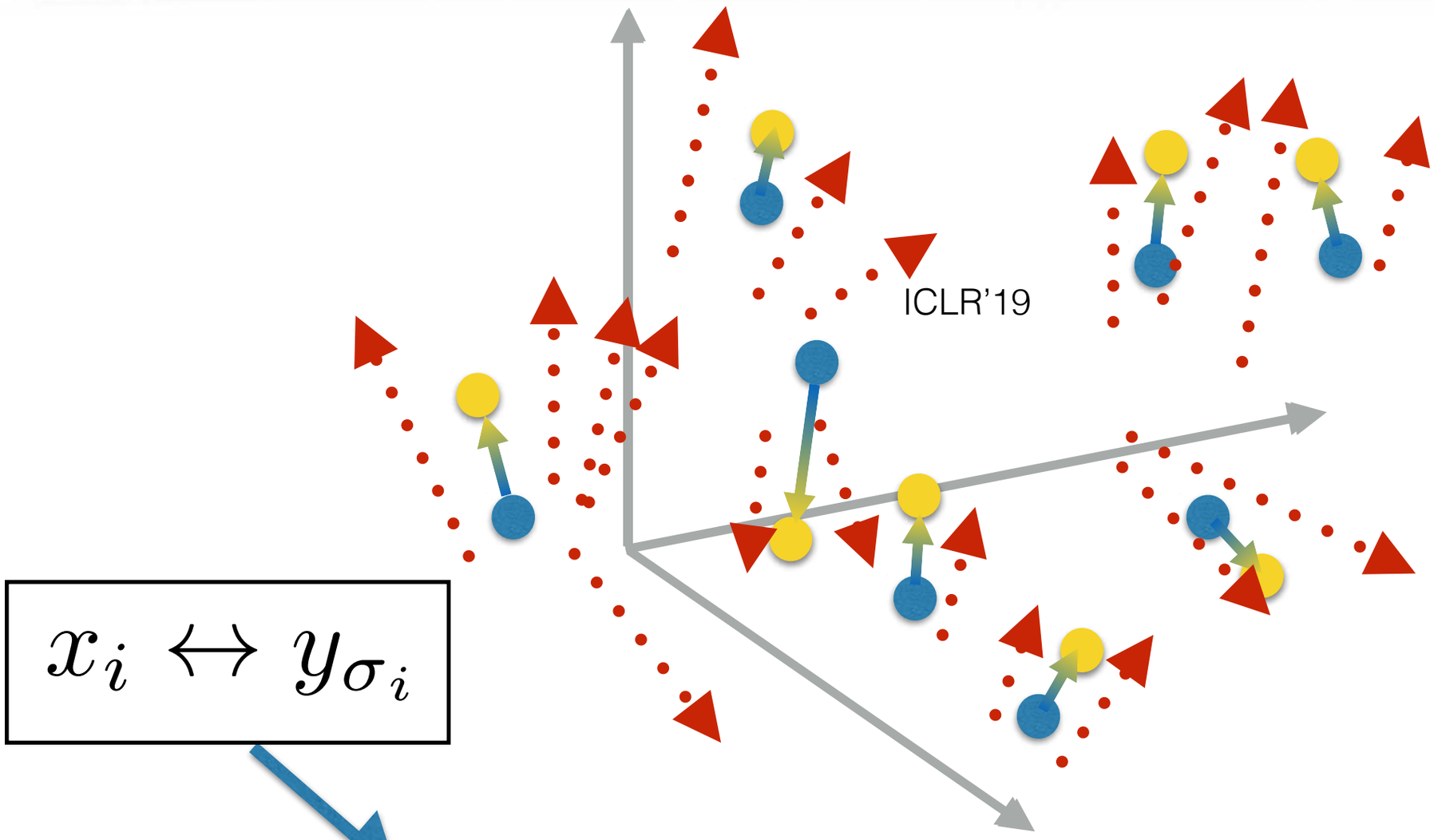
Challenges Down the Road

More likely to happen: *variable number of cells.*
This requires a more versatile notion of matching,
one that is more robust than simple nearest-neighbor.



Challenges Down the Road

A broader goal: *infer* functions *able to provide out of sample mappings, i.e. a continuous extension.*

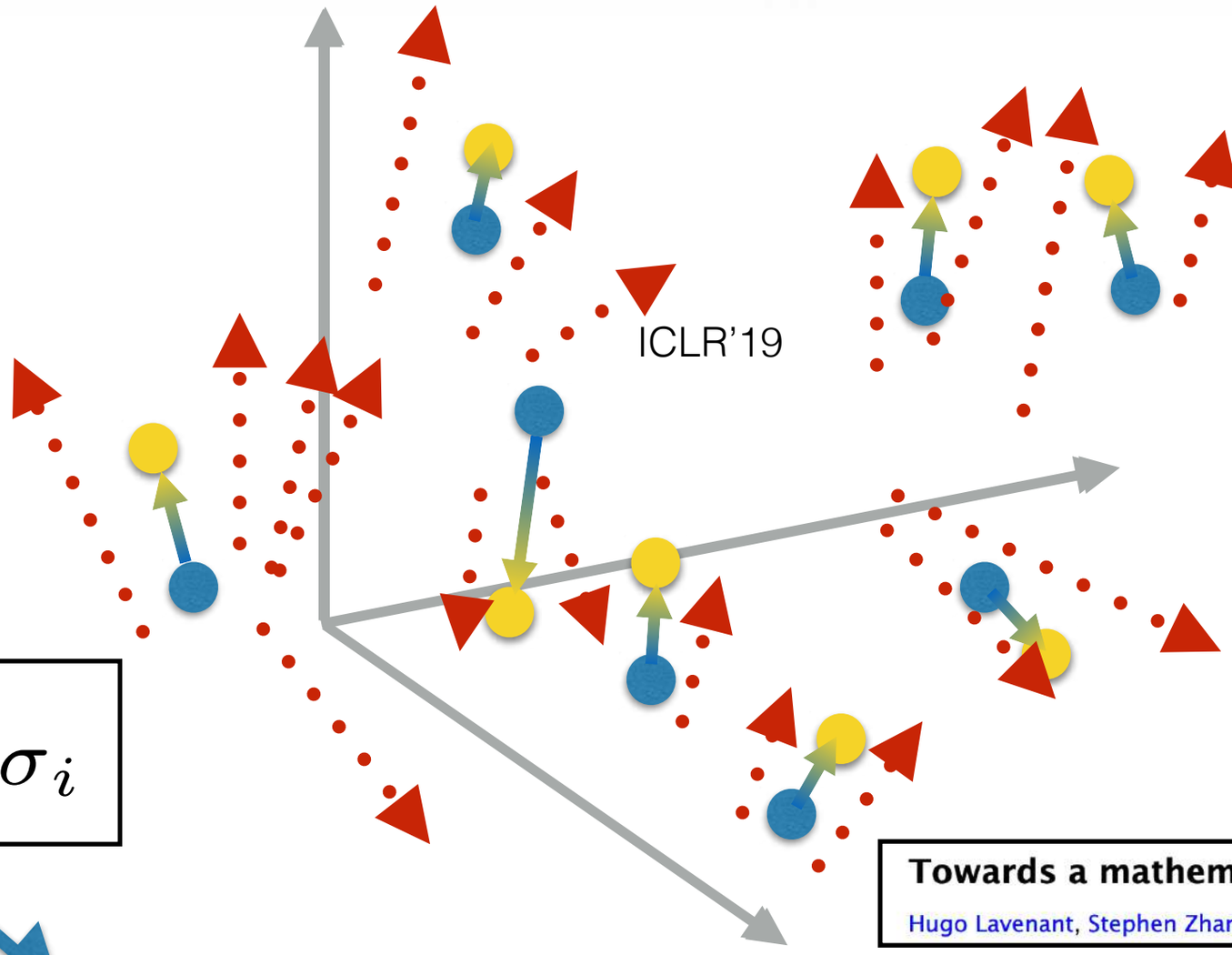


$$x_i \leftrightarrow y_{\sigma_i}$$

$$f_{\boldsymbol{\theta}}(x) \approx y?$$

Challenges Down the Road

A broader goal: *infer* functions *able to provide out of sample mappings, i.e. a continuous extension.*



$$x_i \leftrightarrow y_{\sigma_i}$$

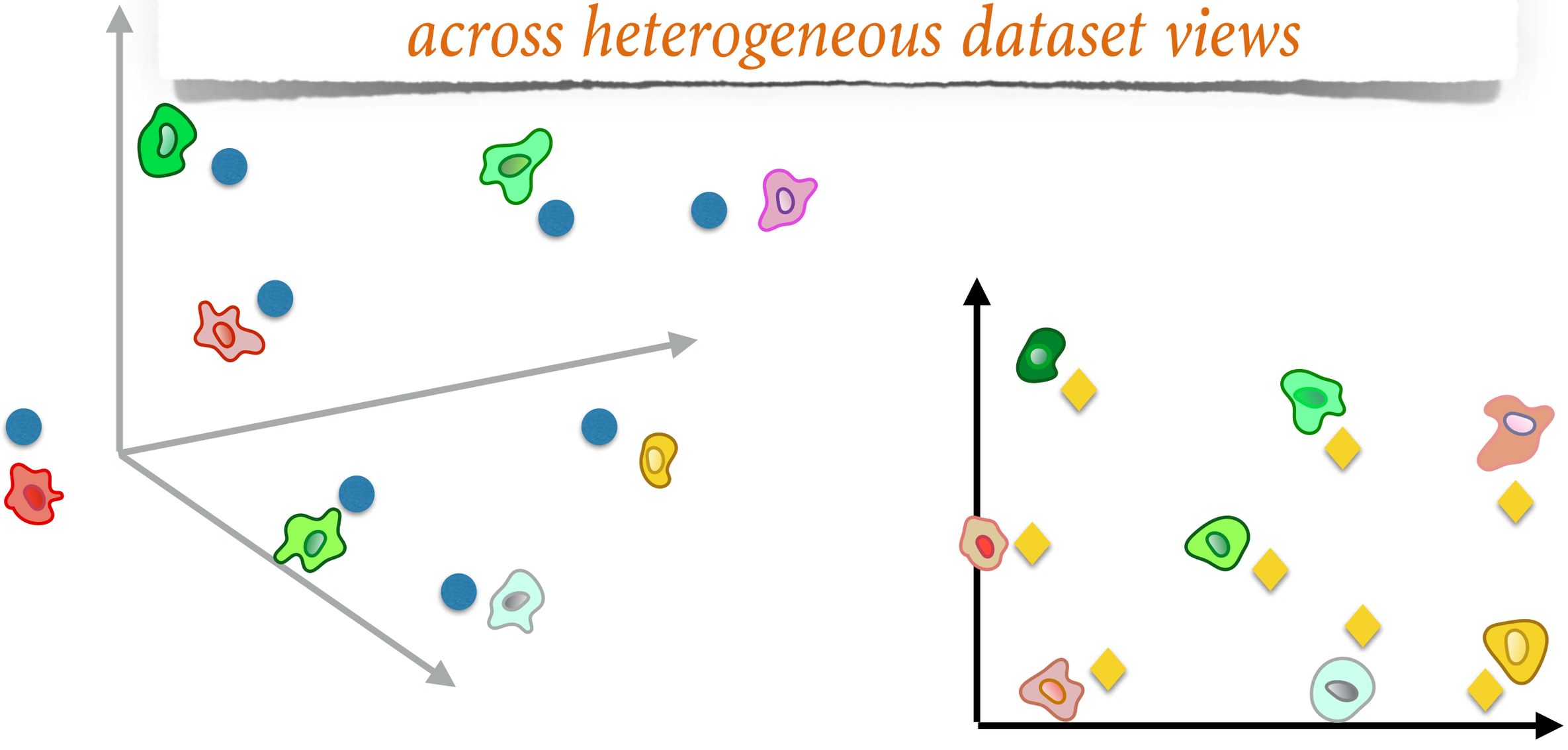
$$f_{\boldsymbol{\theta}}(x) \approx y?$$

Towards a mathematical theory of trajectory inference

Hugo Lavenant, Stephen Zhang, Young-Heon Kim, Geoffrey Schiebinger

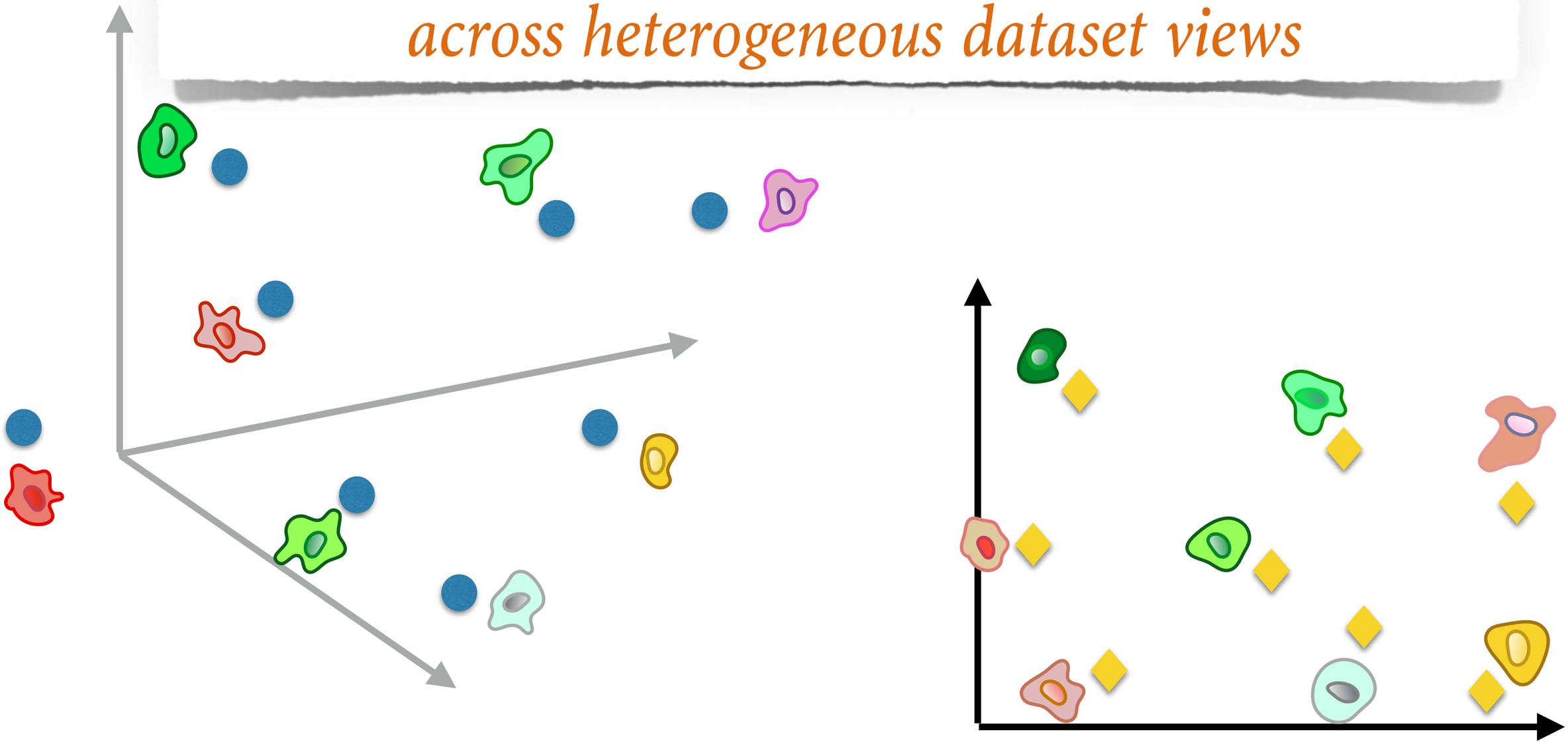
Challenges Down the Road

Add even more flexibility: *match points across heterogeneous dataset views*



Challenges Down the Road

Add even more flexibility: *match points across heterogeneous dataset views*



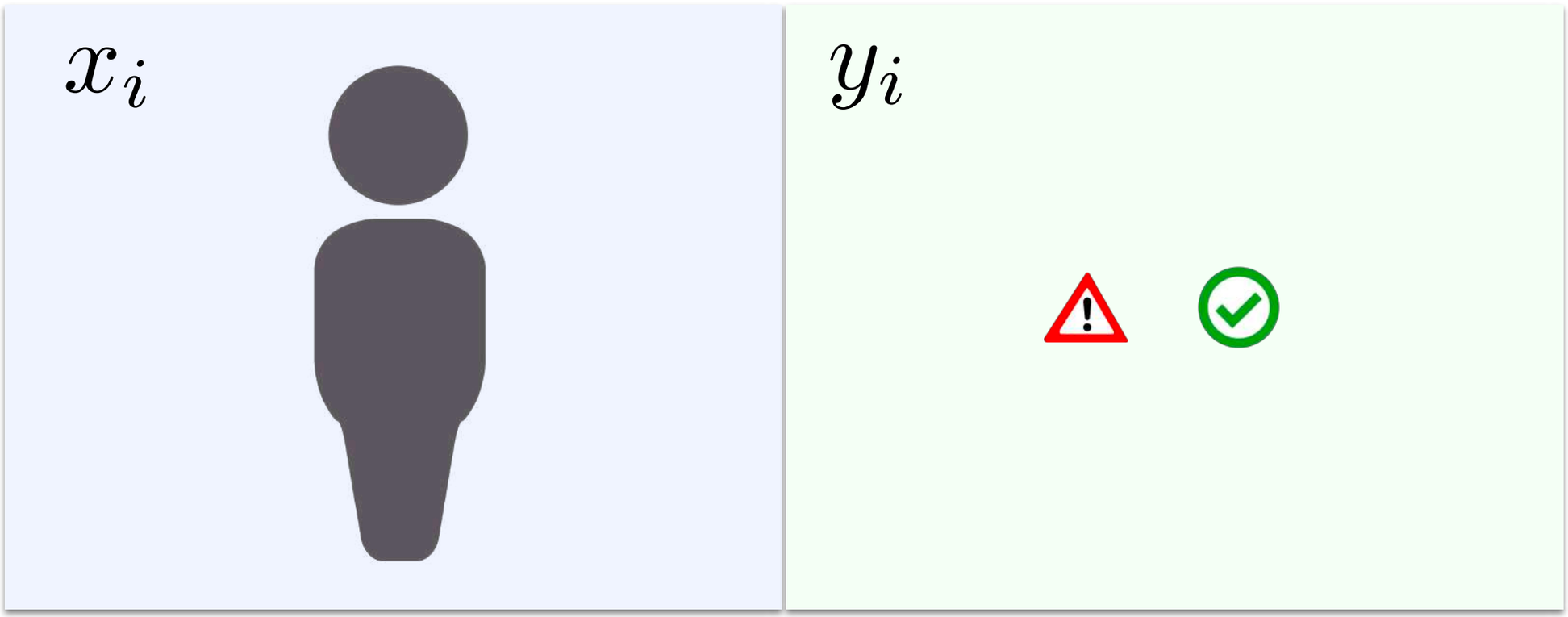
Gromov-Wasserstein optimal transport to align single-cell multi-omics data

Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, Ritambhara Singh

doi: https://doi.org/10.1101/2020.04.28.066787

An Example from Fairness

Person x_i, caracteristic y_i we wish to predict.



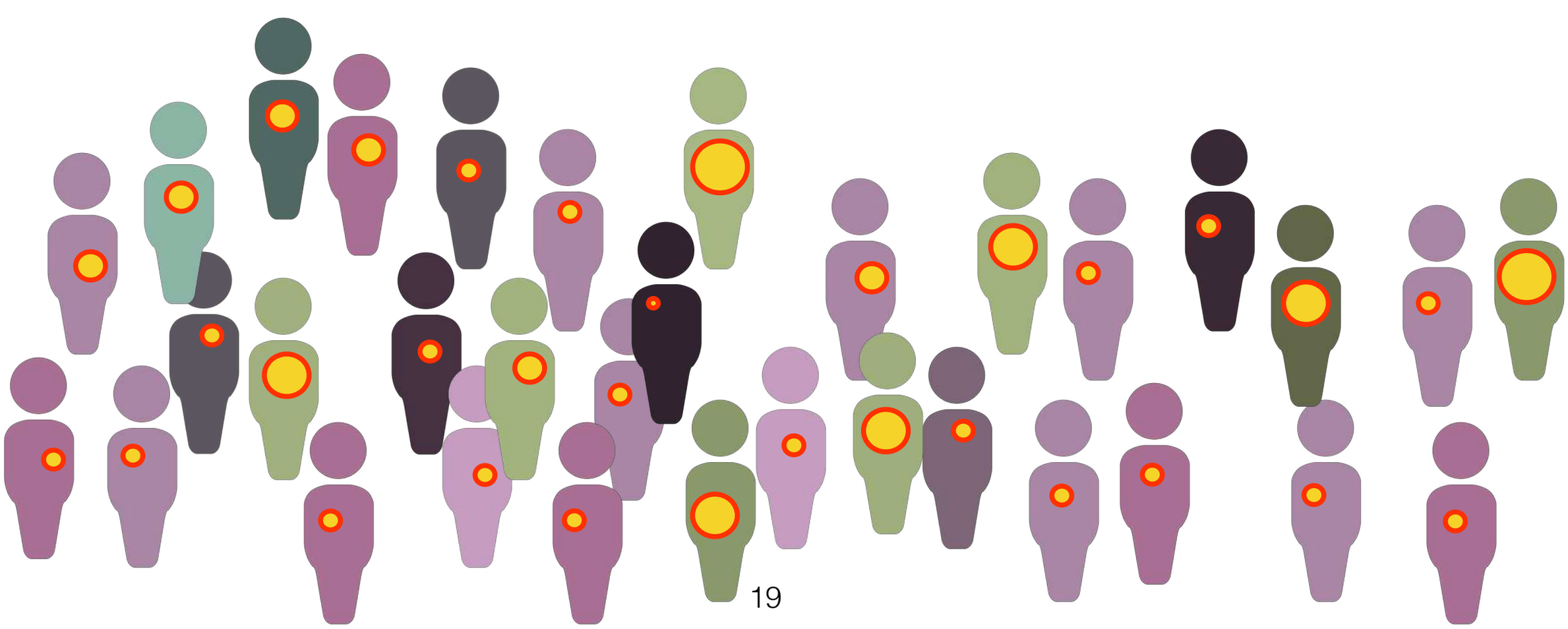
Find $\boldsymbol{\theta}$ such that $f_{\boldsymbol{\theta}}(x_i) \approx y_i$

typically $\min_{\boldsymbol{\theta}} \sum_i \ell(f_{\boldsymbol{\theta}}(x_i), y_i)$

An Example from Fairness

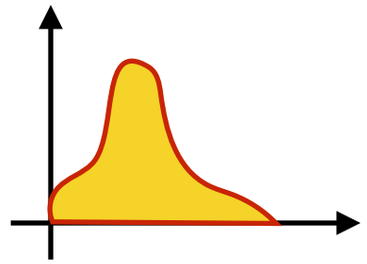
Found $\boldsymbol{\theta}^*$ such that $f_{\boldsymbol{\theta}^*}(x_i) \approx y_i$

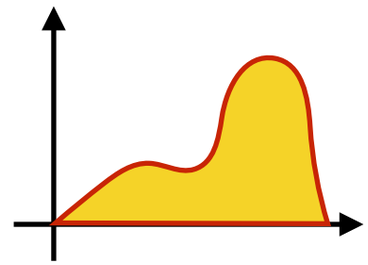
For each individual i, a different error $\ell(f_{\boldsymbol{\theta}^*}(x_i), y_i)$.

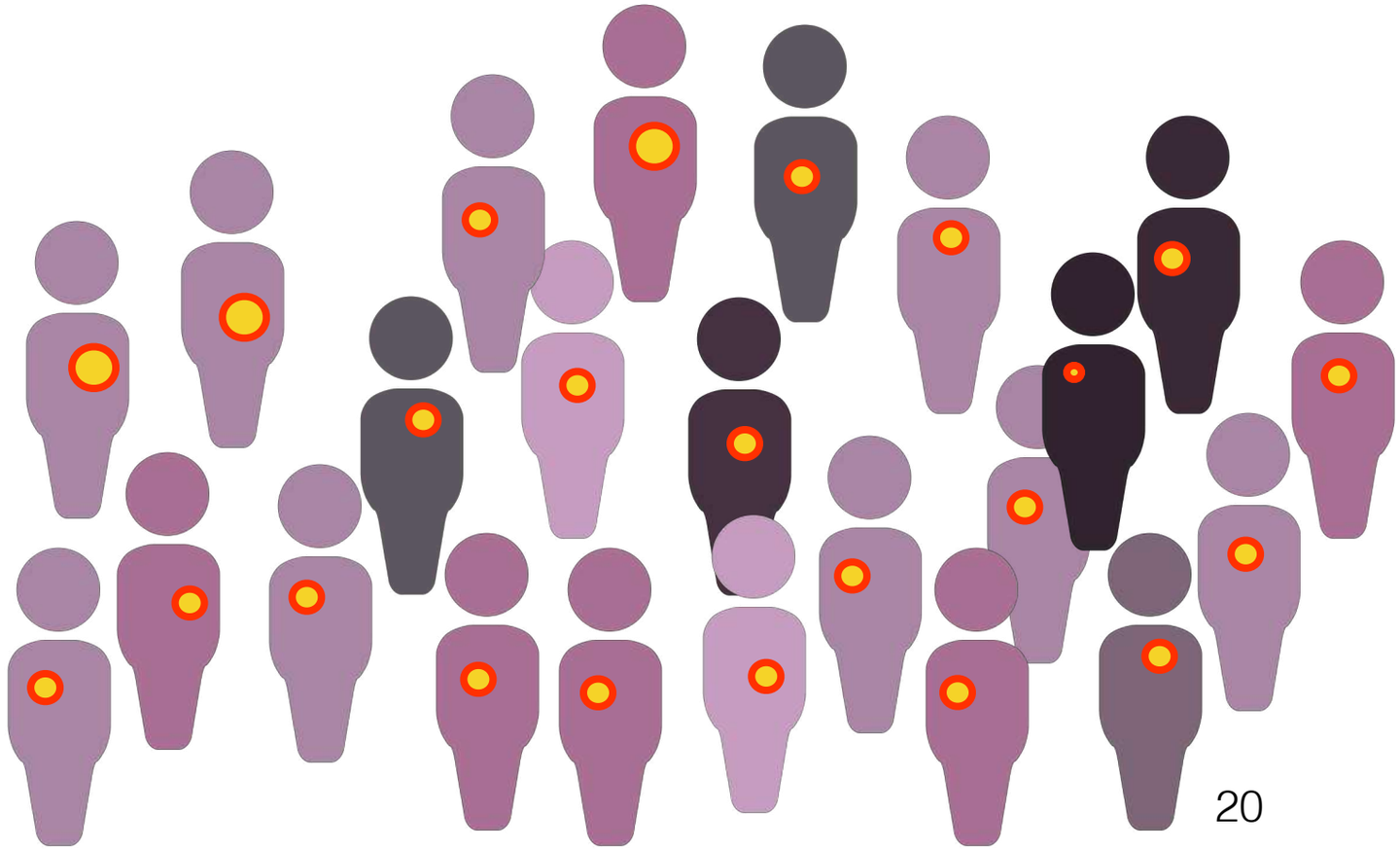


An Example from Fairness

Obviously, $\boldsymbol{\theta}^*$ achieved performance by skewing distribution of $\ell(f_{\boldsymbol{\theta}}(x_i), y_i)$

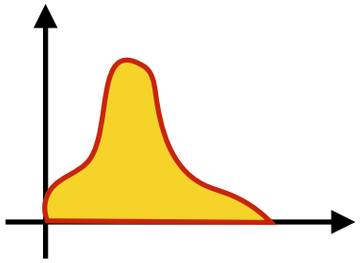


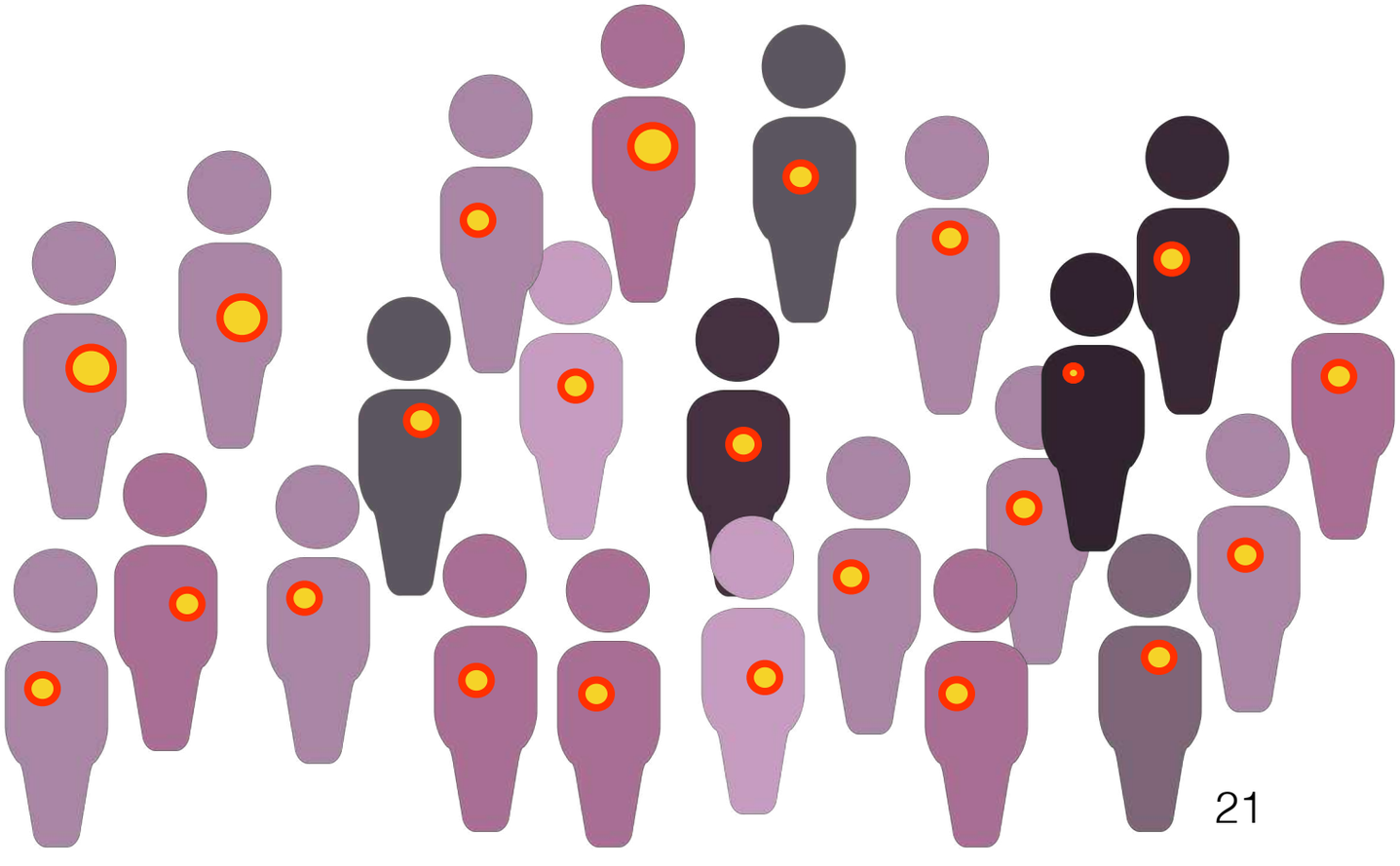


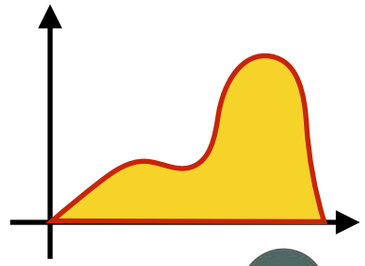


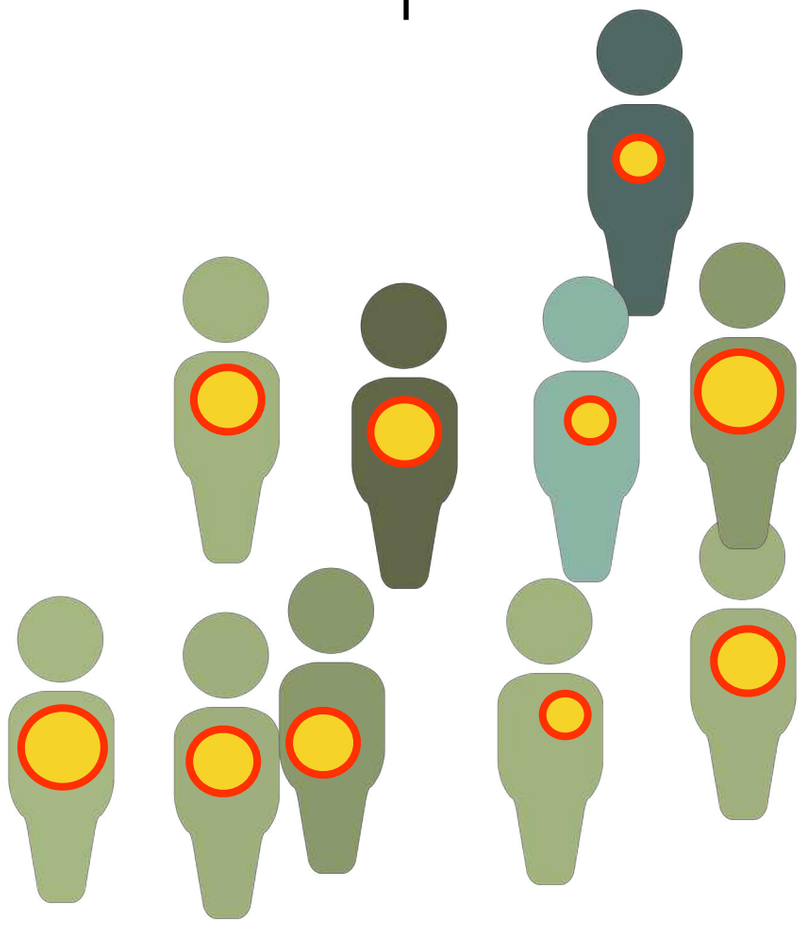
An Example from Fairness

Goal: *enforce fairness mechanisms.*





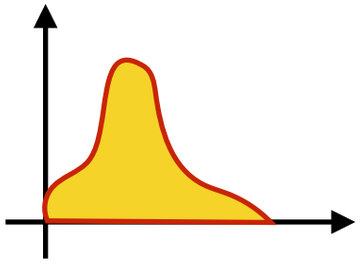


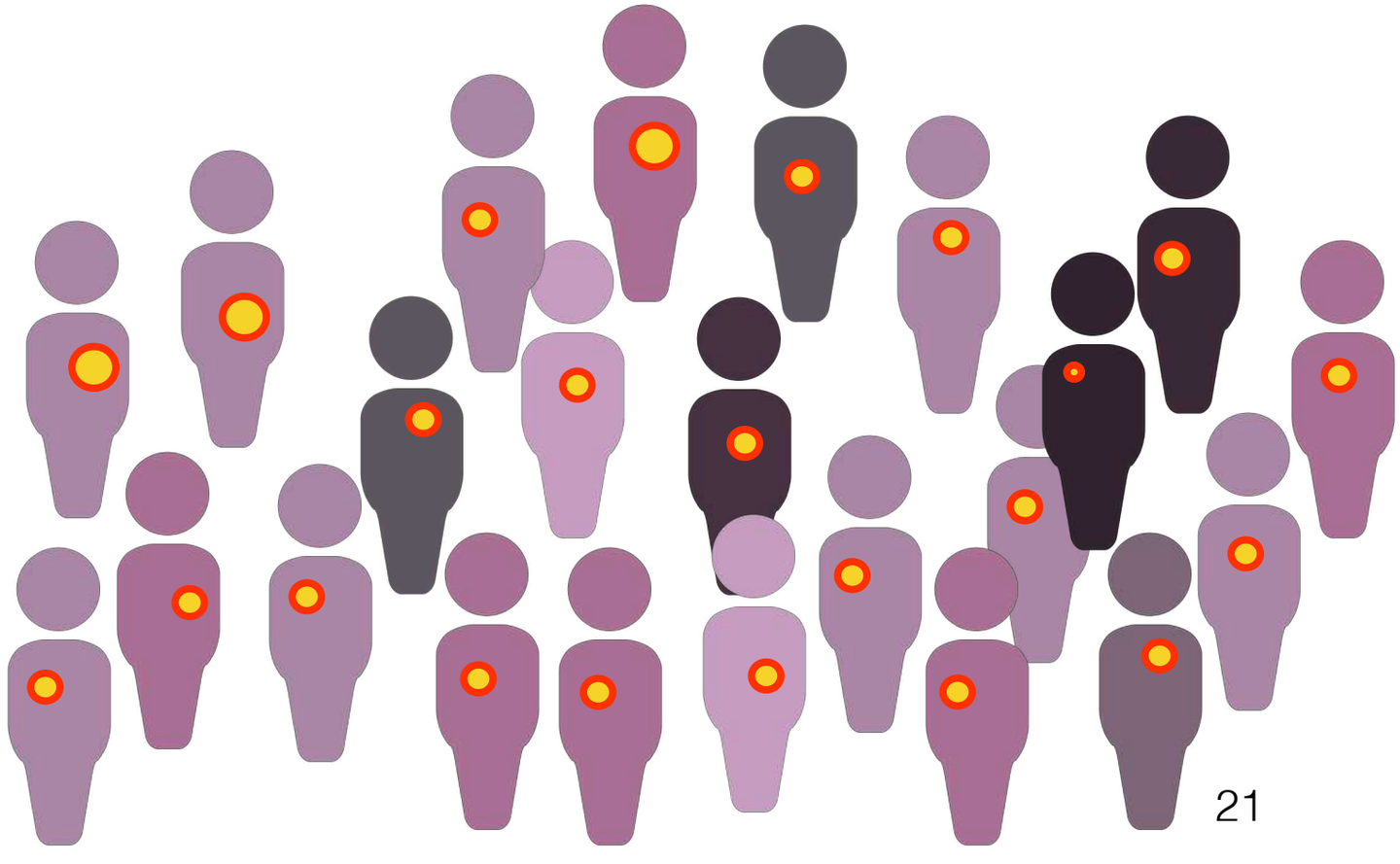


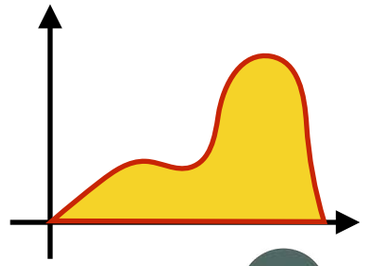
An Example from Fairness

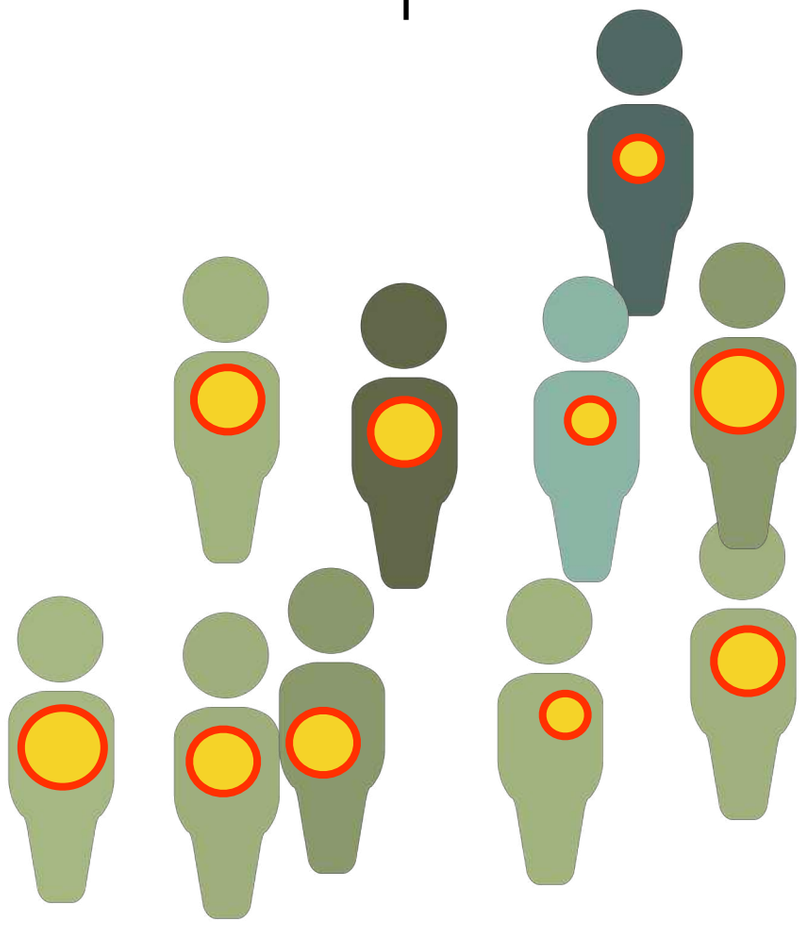
Goal: *enforce fairness mechanisms.*

Problem: *constraints are distributional, not individual*







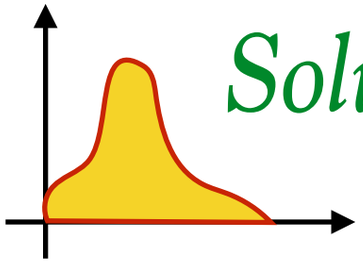


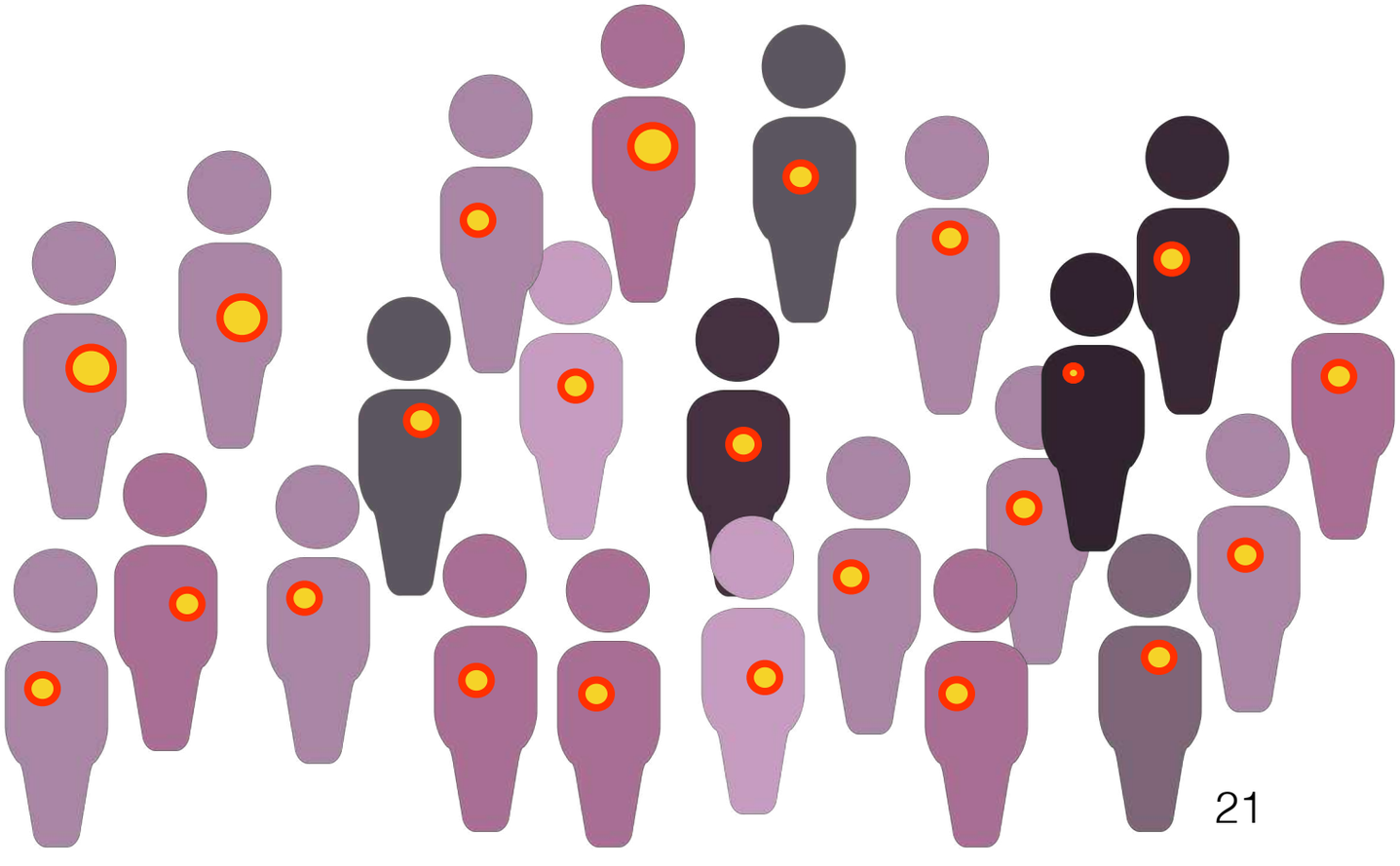
An Example from Fairness

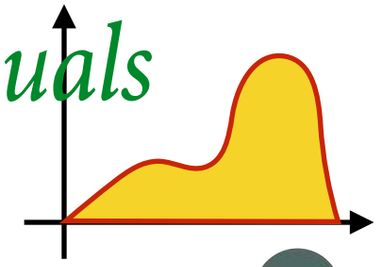
Goal: *enforce fairness mechanisms.*

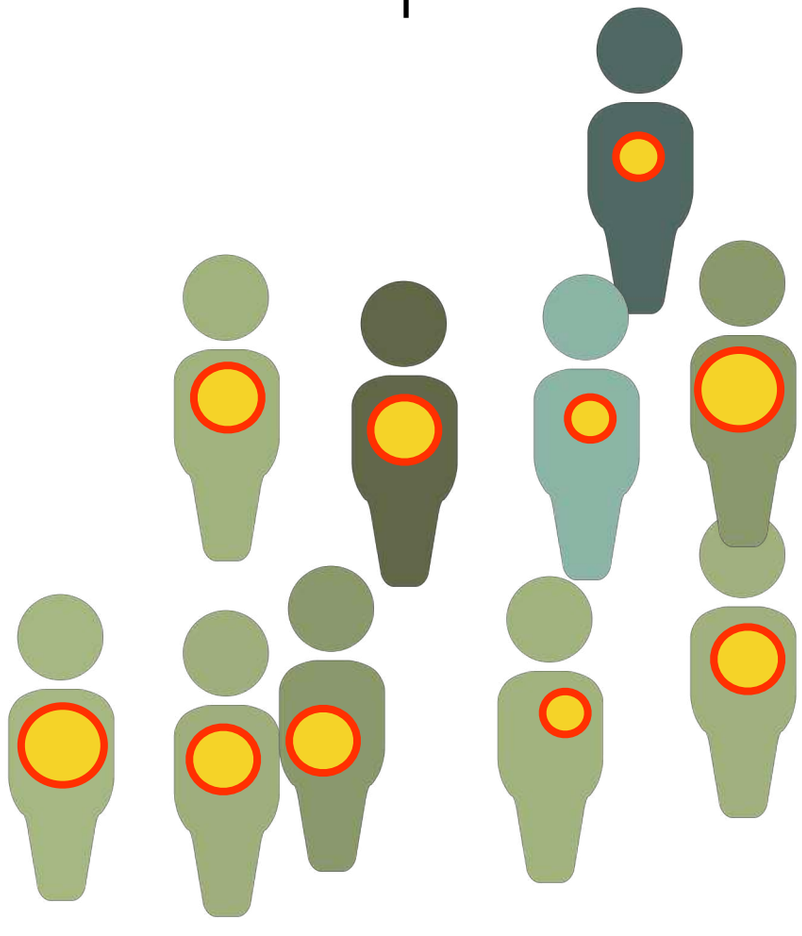
Problem: *constraints are distributional, not individual*

Solution: treat similarly **matched** *individuals*







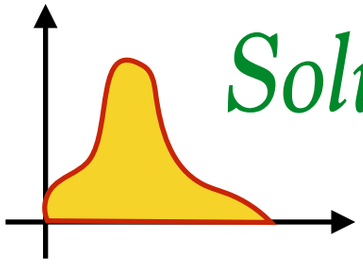


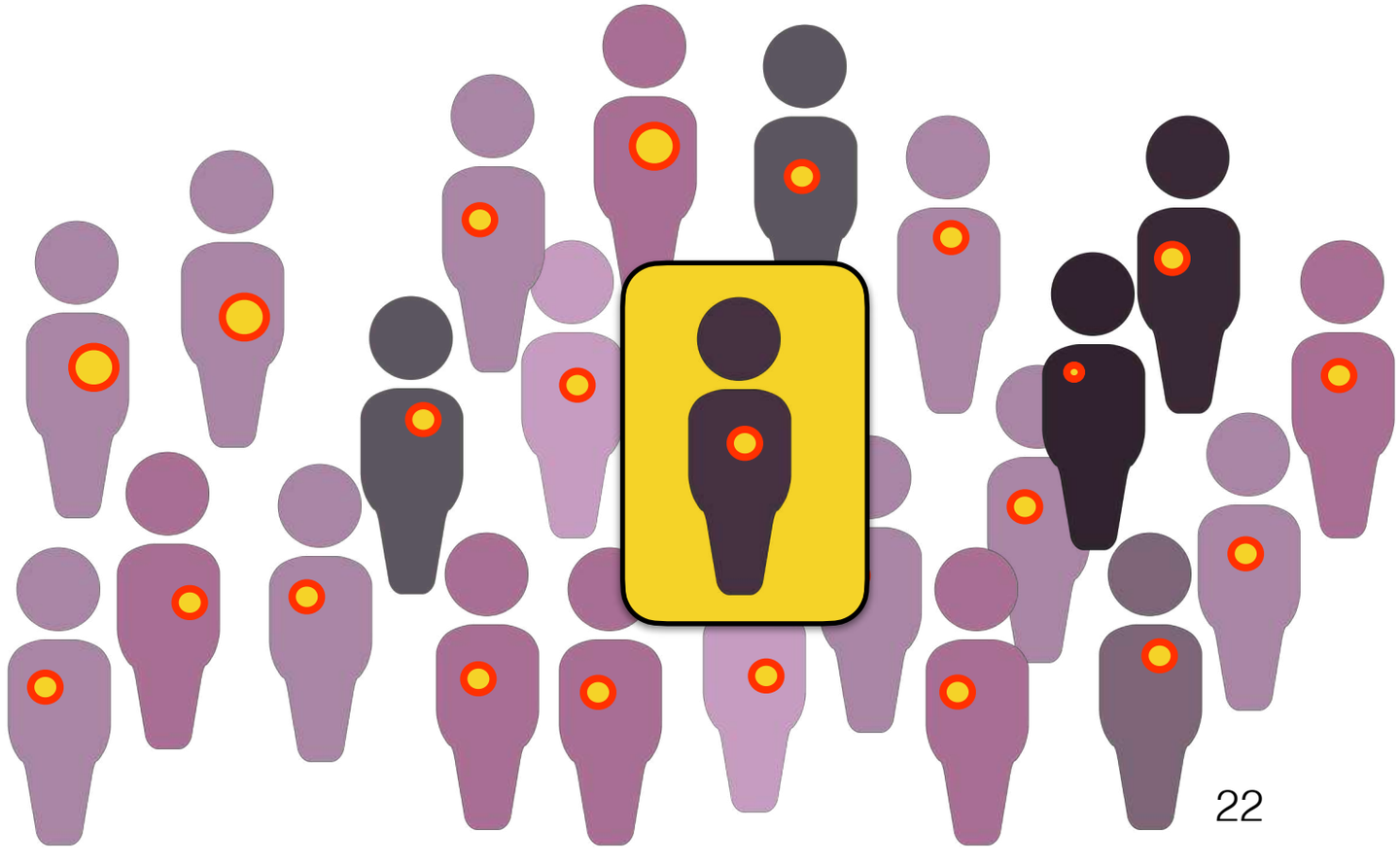
An Example from Fairness

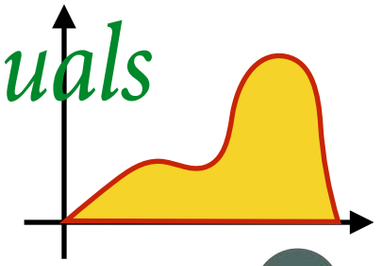
Goal: *enforce fairness mechanisms.*

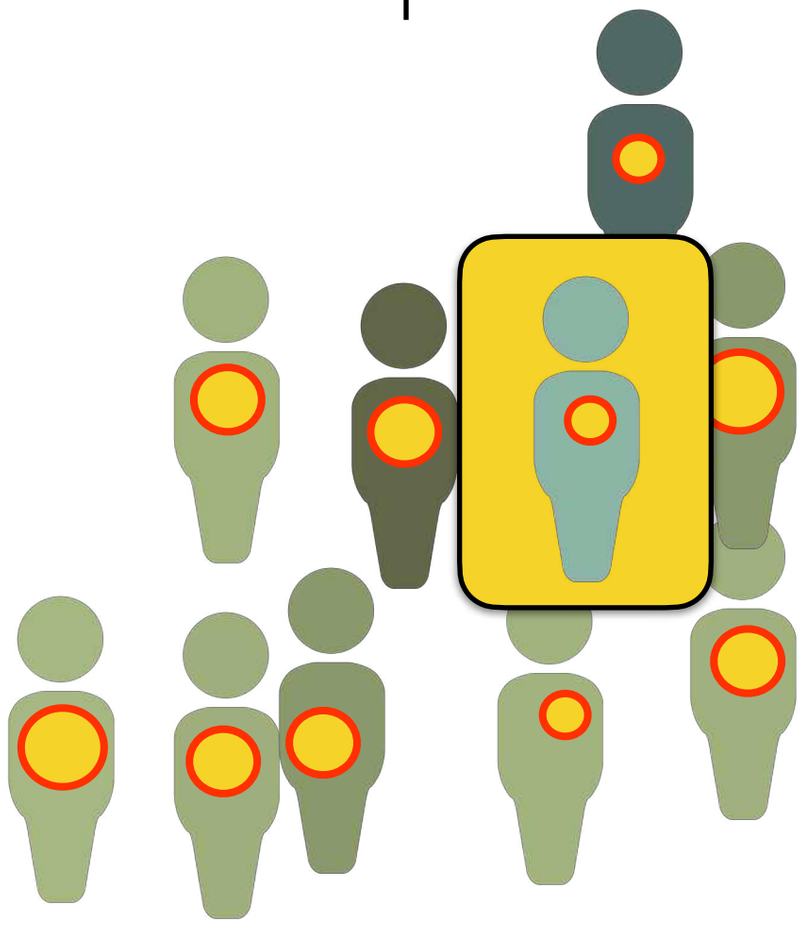
Problem: *constraints are distributional, not individual*

Solution: treat similarly **matched** *individuals*





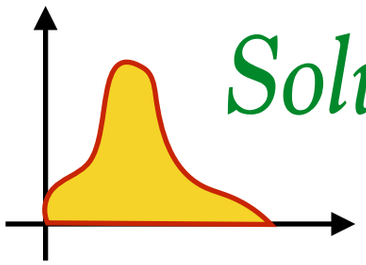




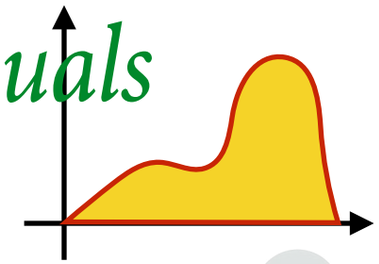
An Example from Fairness

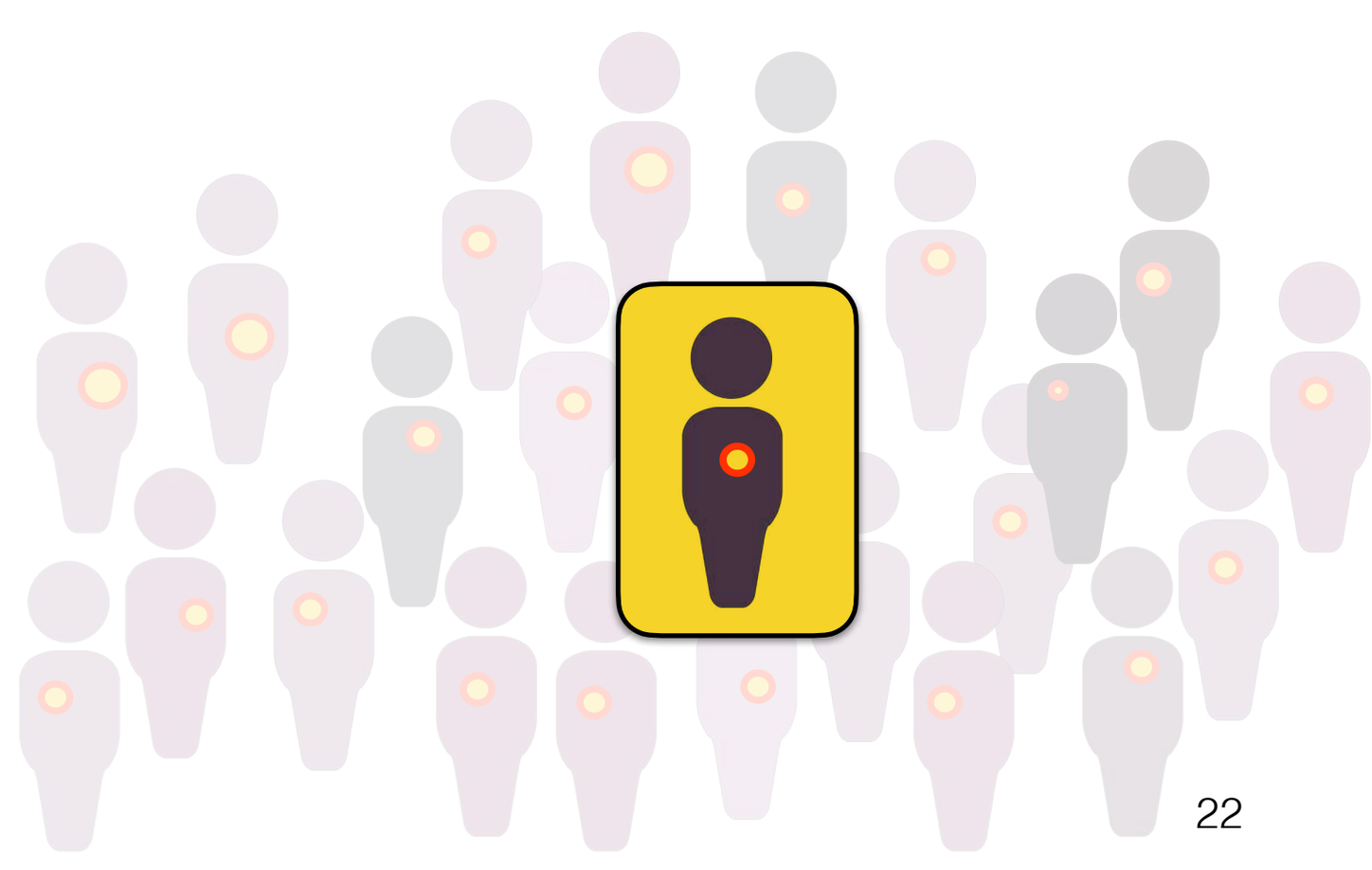
Goal: *enforce fairness mechanisms.*

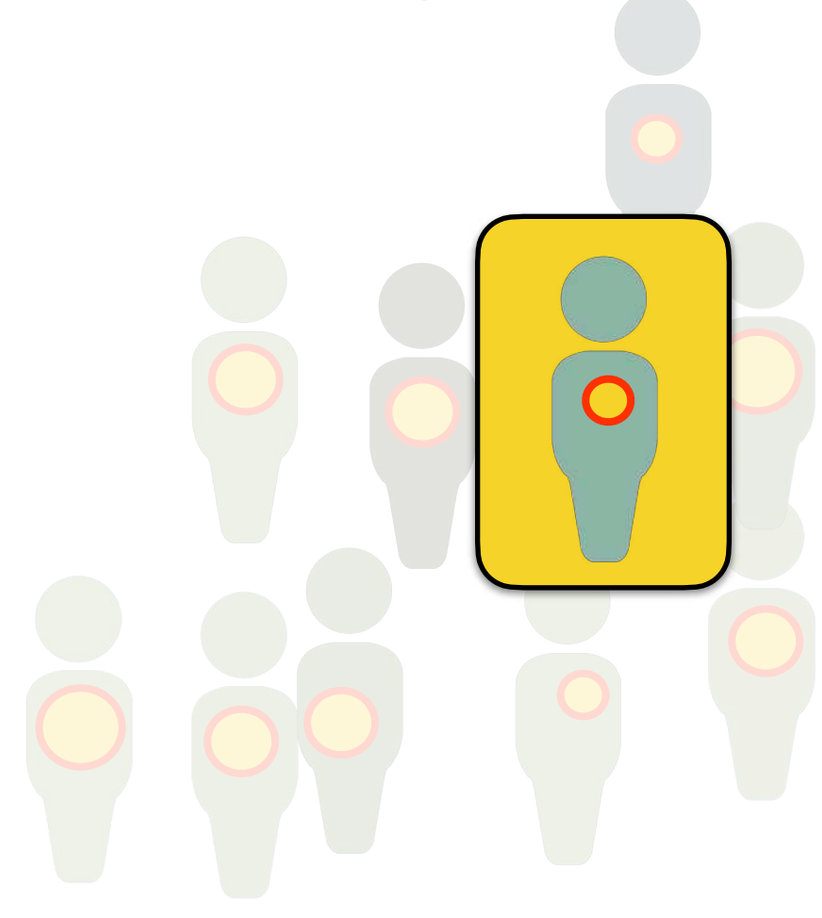
Problem: *constraints are distributional, not individual*



Solution: treat similarly **matched** *individuals*







An Example from Fairness

Goal: *enforce fairness*

fairness optimal transport

About 72,400 results (0.05 sec; Showing 50 m

hor name ○ Word matching ◉

Show the calculator

Obtaining **fairness** using **optimal transport** theory
P Gordaliza, E Del Barrio, G Fabrice - … on Machine Learning, 2019 - proceedings.mlr.press
In the fair classification setup, we recast the links between **fairness** and predictability in terms of probability metrics. We analyze repair methods based on mapping conditional distributions to the Wasserstein barycenter. We propose a Random Repair which yields a …
☆ 99 Cited by 55 Related articles »

Fliptest: **fairness** testing via **optimal transport**
E Black, S Yeom, M Fredrikson, Fairness - … of the 2020 Conference on Fairness …, 2020 - d
We present FlipTest, a black-box technique for uncovering discrimination in classifiers. FlipTest is motivated by the intuitive question: had an individual been of a different protected status, would the model have treated them differently? Rather than relying on causal …
☆ 99 Cited by 18 Related articles All 4 versions

Obtaining **fairness** using **optimal transport** theory
E Del Barrio, F Gamboa, P Gordaliza - arXiv preprint arXiv …, 2018 - arxiv.org
Statistical algorithms are usually helping in making decisions in many aspects of our lives. But, how do we know if these algorithms are biased and commit unfair discrimination of a particular group of people, typically a minority?\textit {**Fairness**} is generally studied in a …
☆ 99 Cited by 11 Related articles All 11 versions »

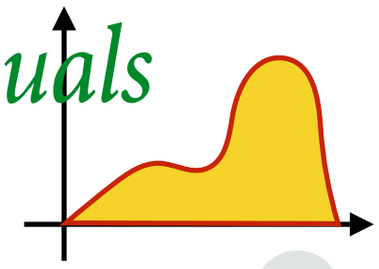
A general approach to **fairness** with **optimal transport**
C Silvia, J Ray, S Tom, P Aldo, J Heinrich - Proceedings of the AAAI …, 2020 - ojs.aaai.org
We propose a general approach to **fairness** based on transporting distributions corresponding to different sensitive attributes to a common distribution. We use **optimal transport** theory to derive target distributions and methods that allow us to achieve **fairness**
☆ 99 Cited by 7 Related articles All 2 versions »

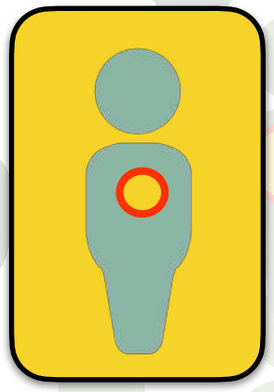
Problem: *constraints are distributional, not individual*

Fairness with Overlapping Groups

Forest Yang, Moustapha Cisse, Sanmi Koyejo

rly **matched** *individuals*





An Example from Fairness

Goal: *enforce fairness*

Problem: *constraints are distributional, not individual*

fairness optimal transport

About 72,400 results (0.05 sec; Showing 50 m

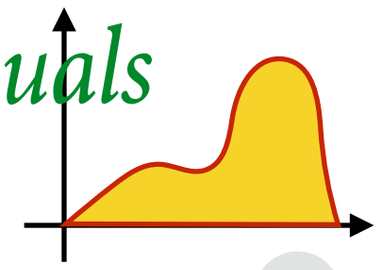
hor name ○ Word matching ◉

Show the calculator

Fairness with Overlapping Groups

Forest Yang, Moustapha Cisse, Sanmi Koyejo

rly **matched** *individuals*

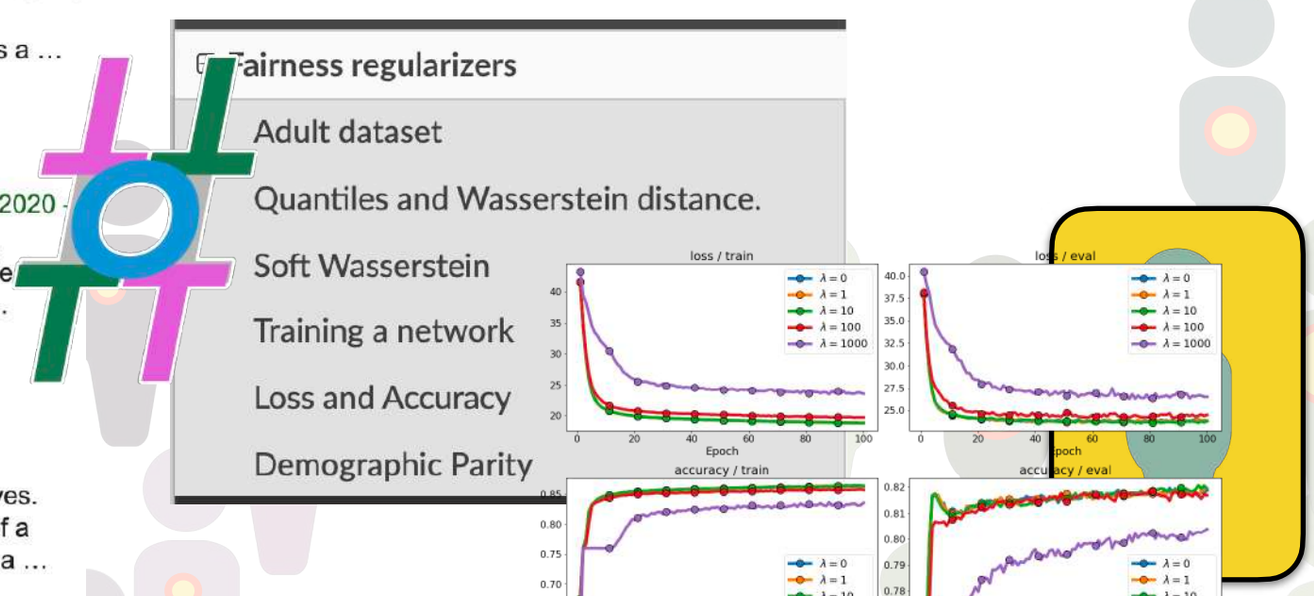


Obtaining **fairness** using **optimal transport** theory
P Gordaliza, E Del Barrio, G Fabrice - … on Machine Learning, 2019 - proceedings.mlr.press
In the fair classification setup, we recast the links between **fairness** and predictability in terms of probability metrics. We analyze repair methods based on mapping conditional distributions to the Wasserstein barycenter. We propose a Random Repair which yields a …
☆ 99 Cited by 55 Related articles »

Fliptest: **fairness** testing via **optimal transport**
E Black, S Yeom, M Fredrikson, Fairness - … of the 2020 Conference on Fairness …, 2020 -
We present FlipTest, a black-box technique for uncovering discrimination in classifiers. FlipTest is motivated by the intuitive question: had an individual been of a different prote status, would the model have treated them differently? Rather than relying on causal …
☆ 99 Cited by 18 Related articles All 4 versions

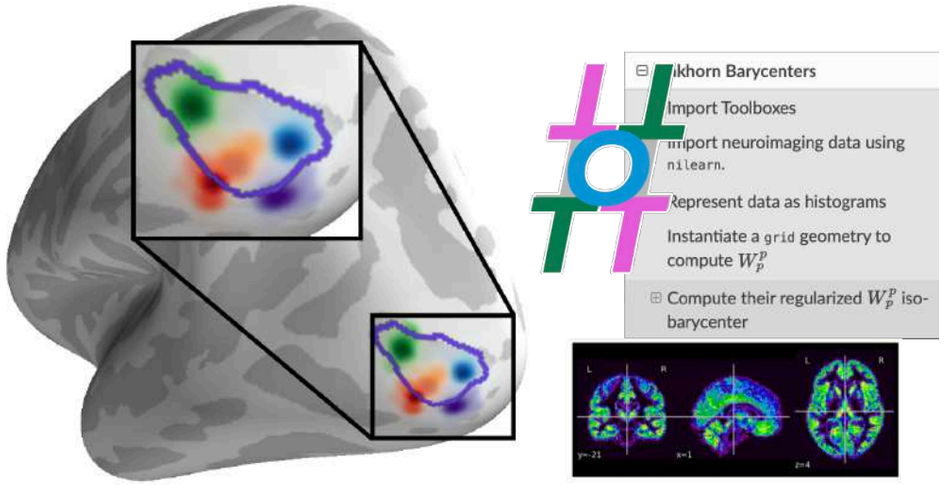
Obtaining **fairness** using **optimal transport** theory
E Del Barrio, F Gamboa, P Gordaliza - arXiv preprint arXiv …, 2018 - arxiv.org
Statistical algorithms are usually helping in making decisions in many aspects of our lives. But, how do we know if these algorithms are biased and commit unfair discrimination of a particular group of people, typically a minority?\textit {**Fairness**} is generally studied in a …
☆ 99 Cited by 11 Related articles All 11 versions »

A general approach to **fairness** with **optimal transport**
C Silvia, J Ray, S Tom, P Aldo, J Heinrich - Proceedings of the AAAI …, 2020 - ojs.aaai.org
We propose a general approach to **fairness** based on transporting distributions corresponding to different sensitive attributes to a common distribution. We use **optimal transport** theory to derive target distributions and methods that allow us to achieve **fairness**
☆ 99 Cited by 7 Related articles All 2 versions »



Optimal matchings are needed everywhere!

Goal: *predict from brain activation maps*



Problem: *no two persons have the same brain, need to match first*

ELSEVIER

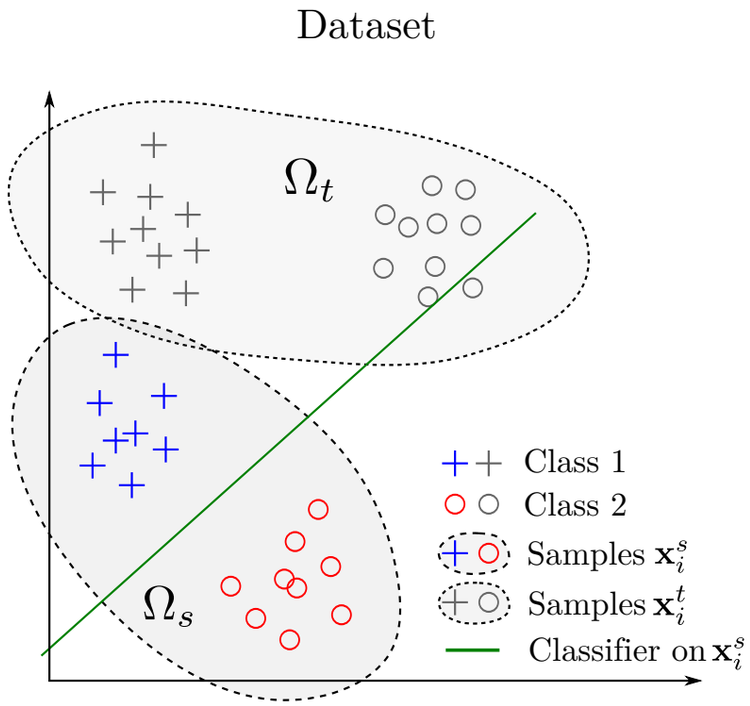
NeuroImage

Volume 220, 15 October 2020, 116847

Multi-subject MEG/EEG source imaging with sparse multi-task regression

Hicham Janati [a, c], Thomas Bazeille [a], Bertrand Thirion [a], Marco Cuturi [b, c], Alexandre Gramfort [a]

Goal: *domain adaptation*



Problem: *train and test data must be matched first*

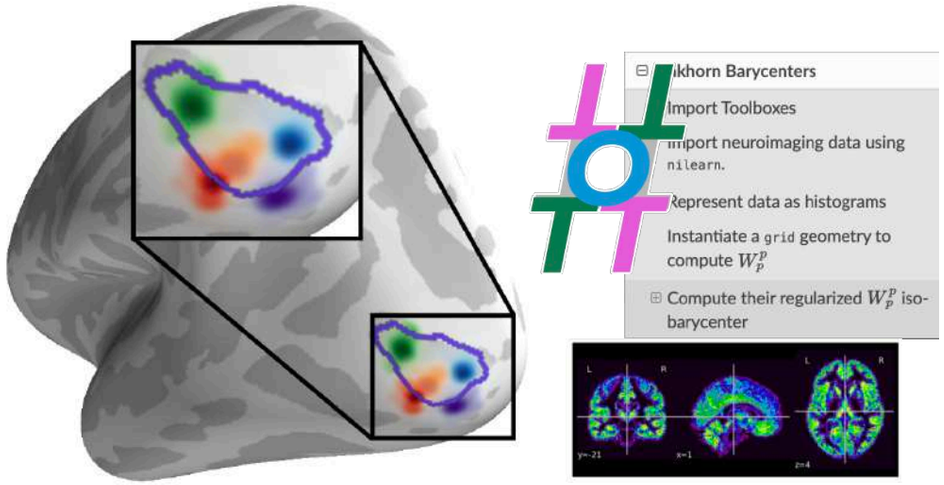
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. X, NO. X, JANUARY XX

Optimal Transport for Domain Adaptation

Nicolas Courty, Rémi Flamary, Devis Tuia, *Senior Member, IEEE*, Alain Rakotomamonjy, *Member, IEEE*

Optimal matchings are needed everywhere!

Goal: *predict from brain activation maps*

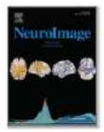


Problem: *no two persons have the same brain, need to match first*

ELSEVIER

NeuroImage

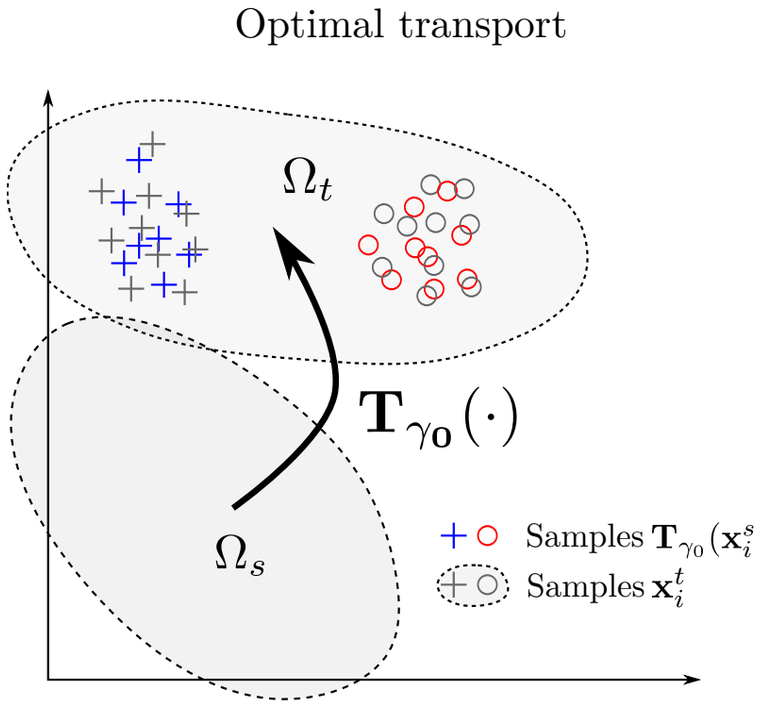
Volume 220, 15 October 2020, 116847



Multi-subject MEG/EEG source imaging with sparse multi-task regression

Hicham Janati [a, c], Thomas Bazeille [a], Bertrand Thirion [a], Marco Cuturi [b, c], Alexandre Gramfort [a]

Goal: *domain adaptation*



Problem: *train and test data must be matched first*

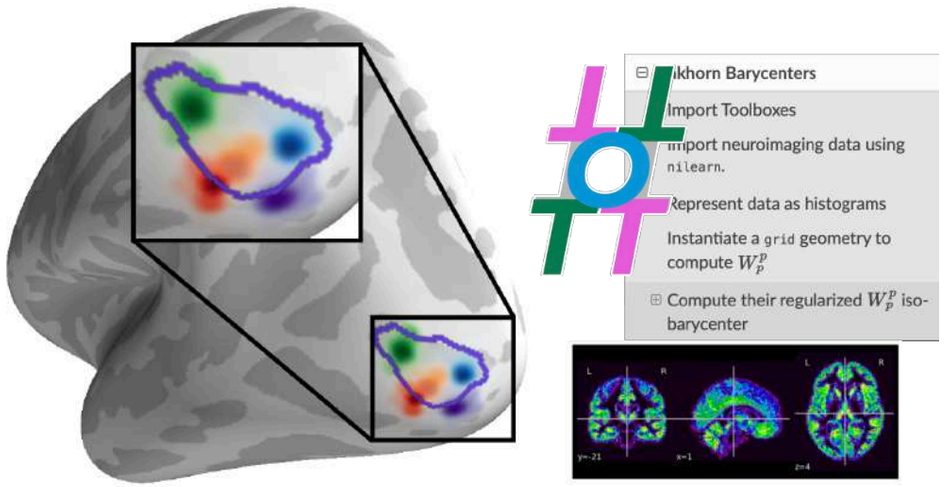
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. X, NO. X, JANUARY XX

Optimal Transport for Domain Adaptation

Nicolas Courty, Rémi Flamary, Devis Tuia, *Senior Member, IEEE*, Alain Rakotomamonjy, *Member, IEEE*

Optimal matchings are needed everywhere!

Goal: *predict from brain activation maps*

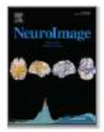


Problem: *no two persons have the same brain, need to match first*

ELSEVIER

NeuroImage

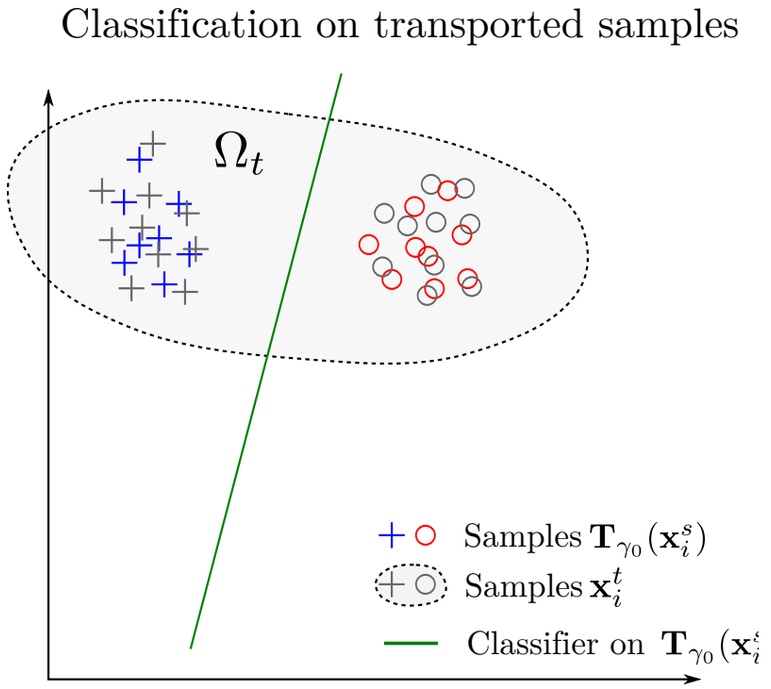
Volume 220, 15 October 2020, 116847



Multi-subject MEG/EEG source imaging with sparse multi-task regression

Hicham Janati [a, c], Thomas Bazeille [a], Bertrand Thirion [a], Marco Cuturi [b, c], Alexandre Gramfort [a]

Goal: *domain adaptation*



Problem: *train and test data must be matched first*

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. X, NO. X, JANUARY XX

Optimal Transport for Domain Adaptation

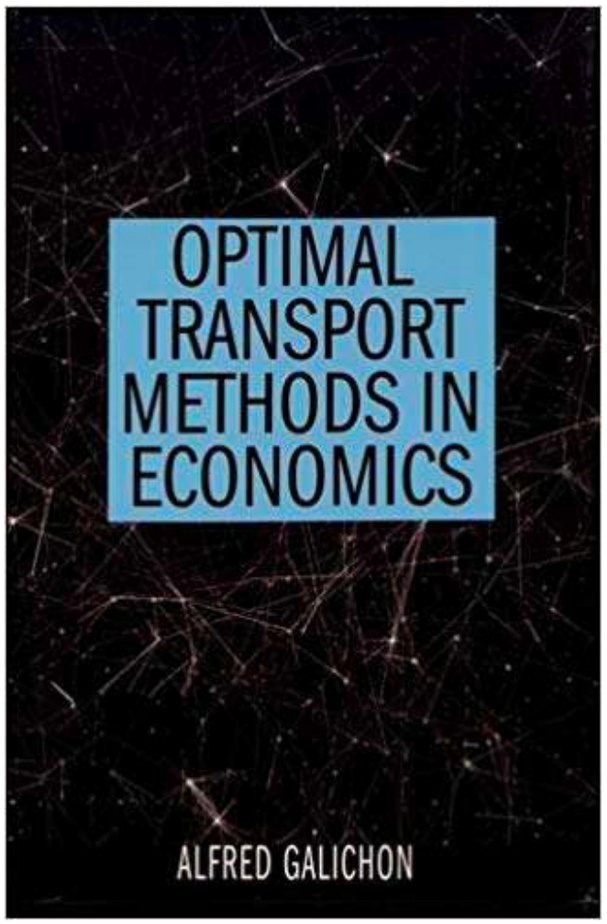
Nicolas Courty, Rémi Flamary, Devis Tuia, *Senior Member, IEEE*, Alain Rakotomamonjy, *Member, IEEE*

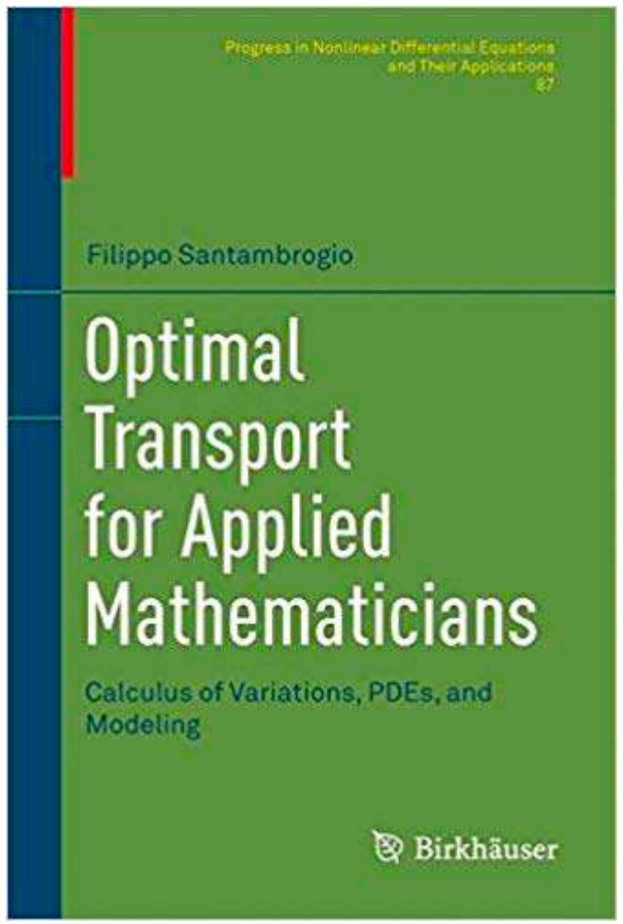
In all these problems…

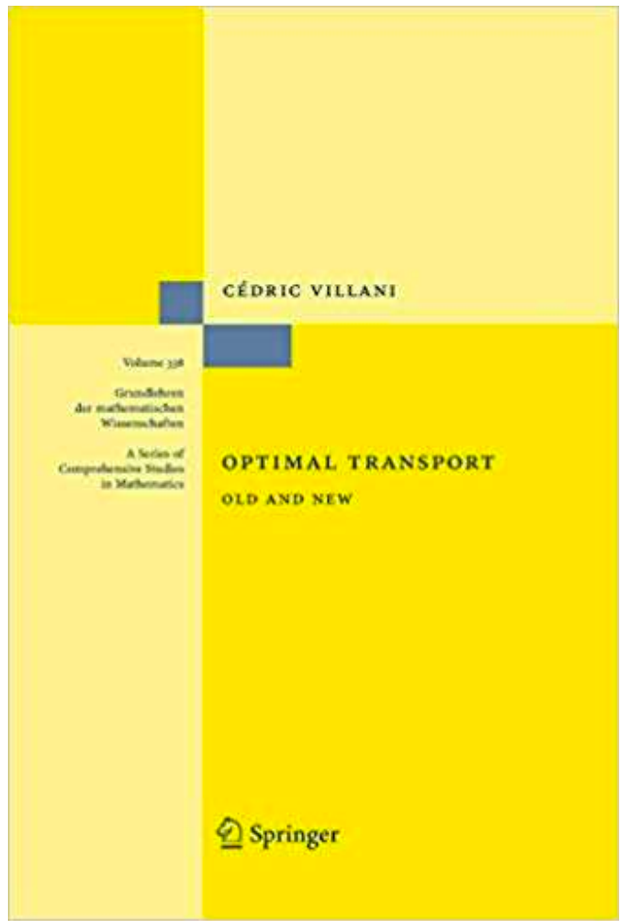
- An **intermediate** *matching/mapping* step is needed.
- That step is only **intermediate** in the sense that other *advanced learning procedures* build on top of it.
- Wishlist for a *matching* framework:
  - *guided by rich theory*, to inform new algorithms,
  - *versatile*, to accommodate several settings,
  - *scalable*, to handle large *dim*/*n* datesets,
  - *differentiable*, for end-to-end learning,
  - *robust*, to perform well in most settings.

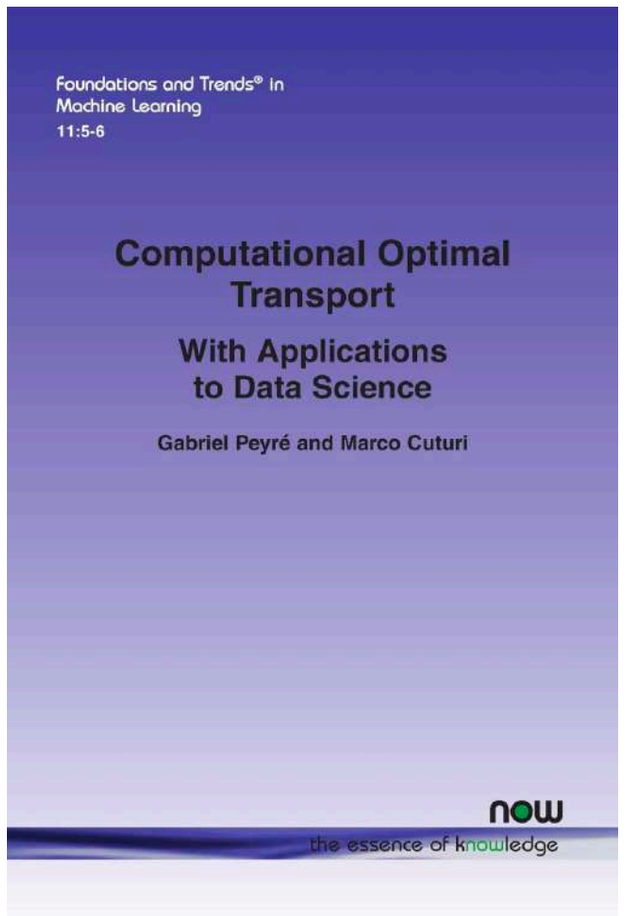
That theory is Optimal Transport

Matchings happen everywhere, so many people have (re)discovered it: economics, *applied* maths, *pure* maths, graphics, and, increasingly ML people!









Why is it popular across so many fields?

It has a bit of everything! **theory** provides theorems; **algorithms** translate them in tangible results; **applications** guide new developments.

Monge

Kantorovich

Koopmans

Nobel'75

Dantzig

McCann

Otto

Ambrosio

Figalli

Fields'18

Brenier

Caffarelli

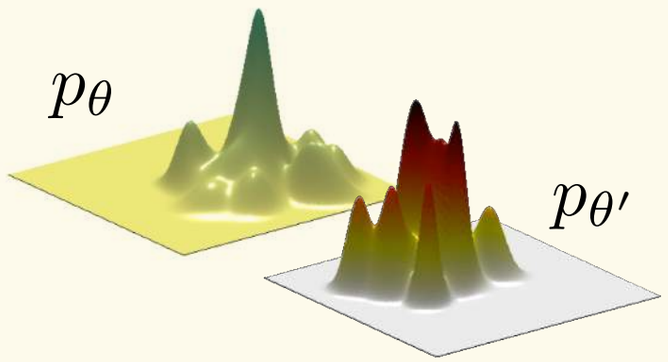
Villani

Fields'10

Gangbo

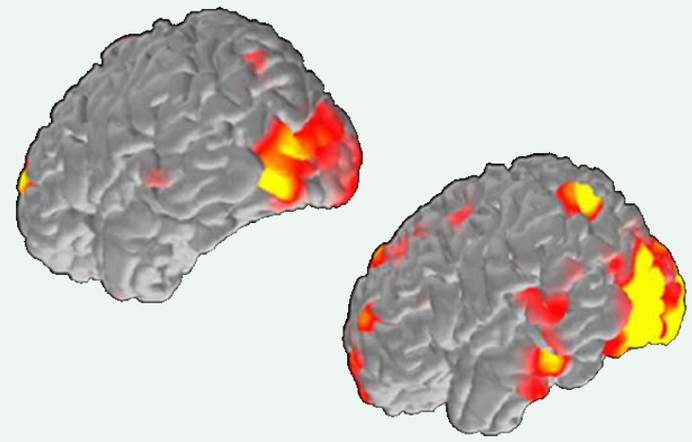
Optimal Transport in Data Sciences

Advertised as the natural way to define geometry, through matching, for probability measures, when supported on a geometric space.



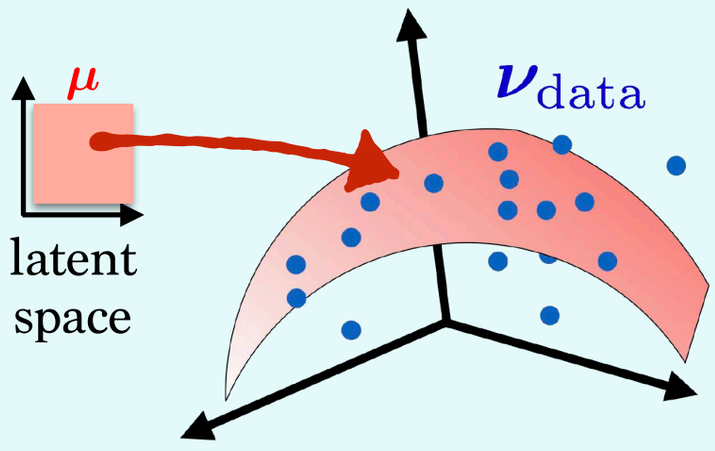
Statistical Models

Bags of features



Brain Activation Maps

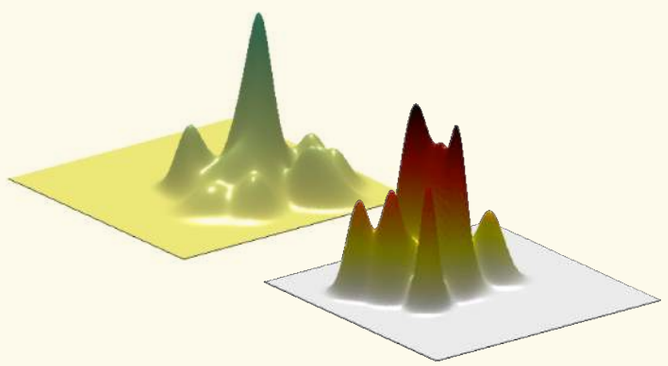
Generative Models vs. data



Color Histograms

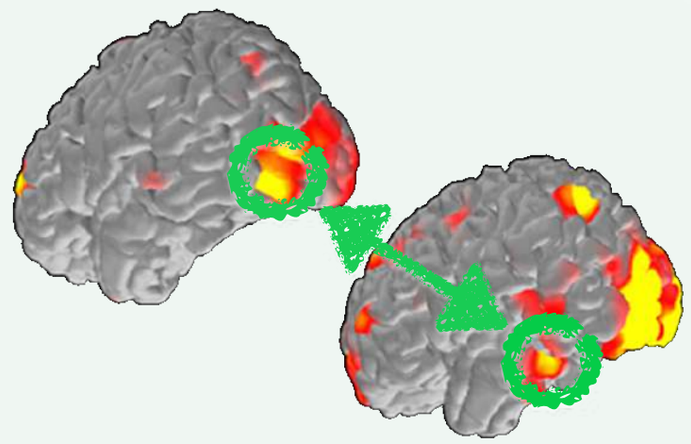
Optimal Transport in Data Sciences

Advertised as the natural way to define geometry, through matching, for probability measures, when supported on a geometric space.



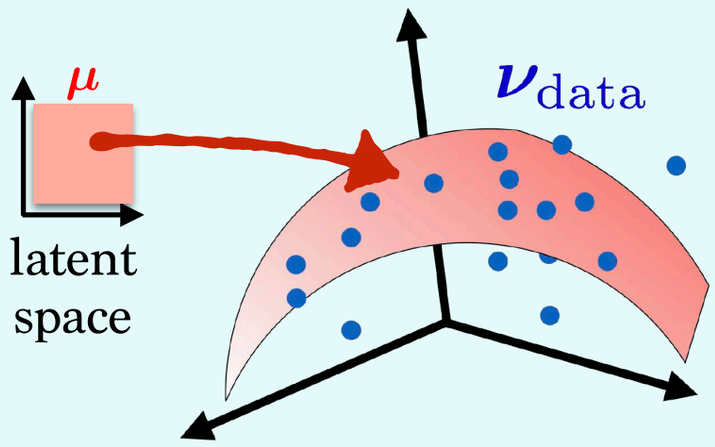
Statistical Models

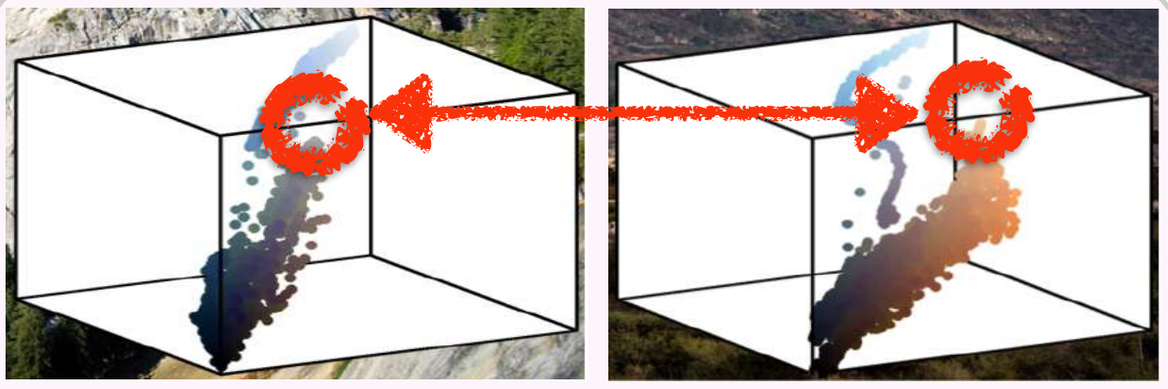
Bags of Features



Brain Activation Maps

Generative Models vs. Data

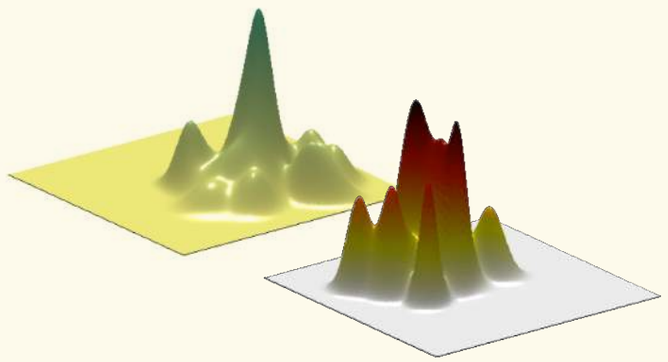




Color Histograms

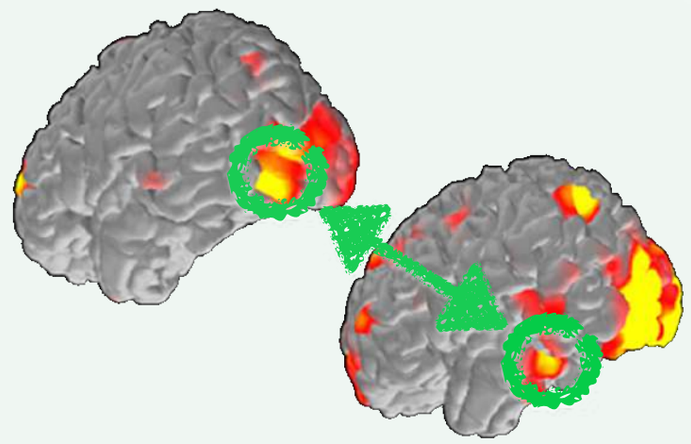
Optimal Transport in Data Sciences

Advertised as the natural way to define geometry, through matching, for probability measures, when supported on a geometric space.



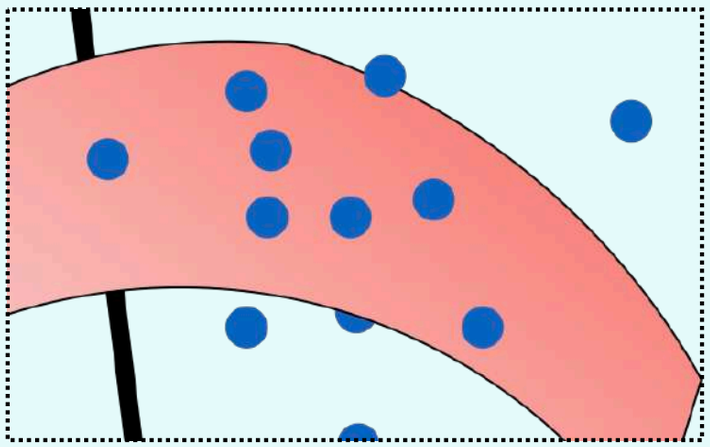
Statistical Models

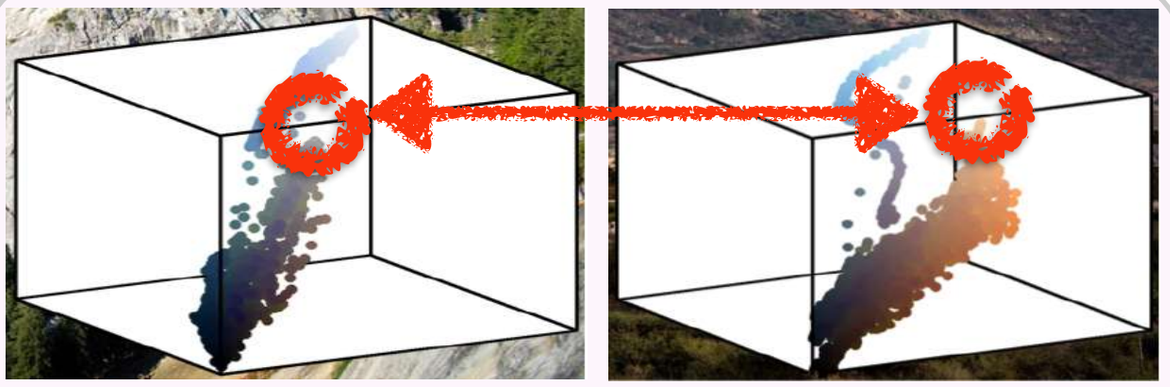
Bags of Features



Brain Activation Maps

Generative Models vs. Data

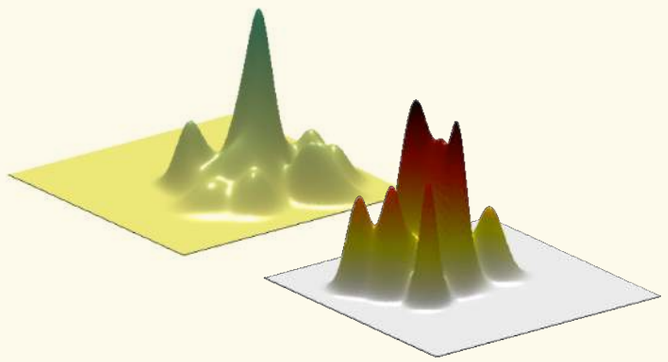




Color Histograms

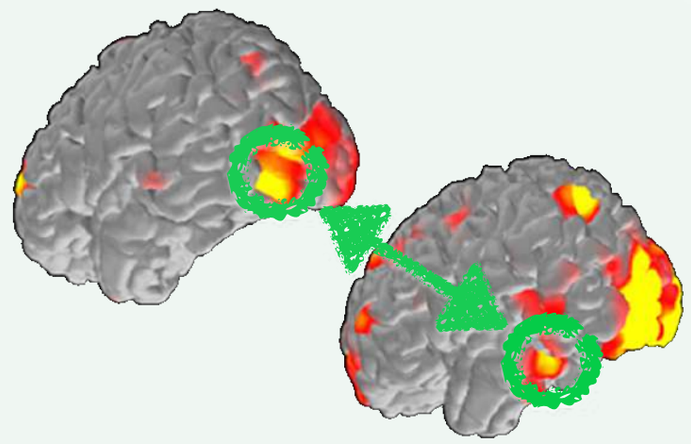
Optimal Transport in Data Sciences

Advertised as the natural way to define geometry, through matching, for probability measures, when supported on a geometric space.



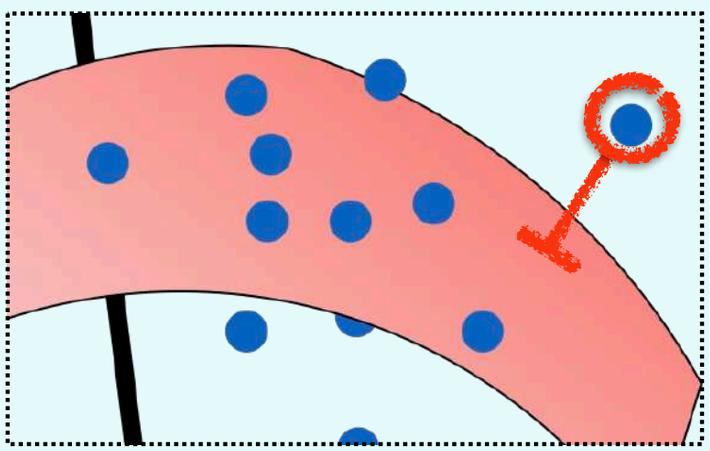
Statistical Models

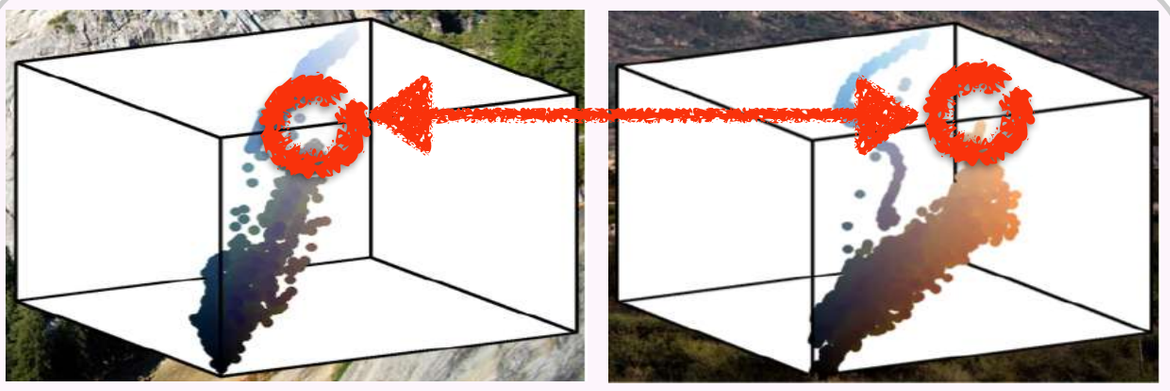
Bags of Features



Brain Activation Maps

Generative Models vs. Data





Color Histograms

Short Course Outline

1. Introduction to optimal transport
2. Computing OT exactly
3. Computing OT for data science

Introduction to OT

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

Origins: Monge Problem (1781)

Gaspard Monge (1746 - 1818)

Origins: Monge Problem (1781)

tripadvisor SOUTH AFRICA | Paris | Basket | Trips | Inbox | Profile | Join | Search

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

Europe › France › Ile-de-France › Paris › Things to do in Paris › Place Monge

Place Monge, Paris: Address, Place Monge Reviews: 4/5

Place Monge

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

Pl. Monge, Paris, France

Save | Share

Review Highlights

"Good local market.'

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

Reviewed 26 September 2018

johngl8492UH , Middle Park via mobile

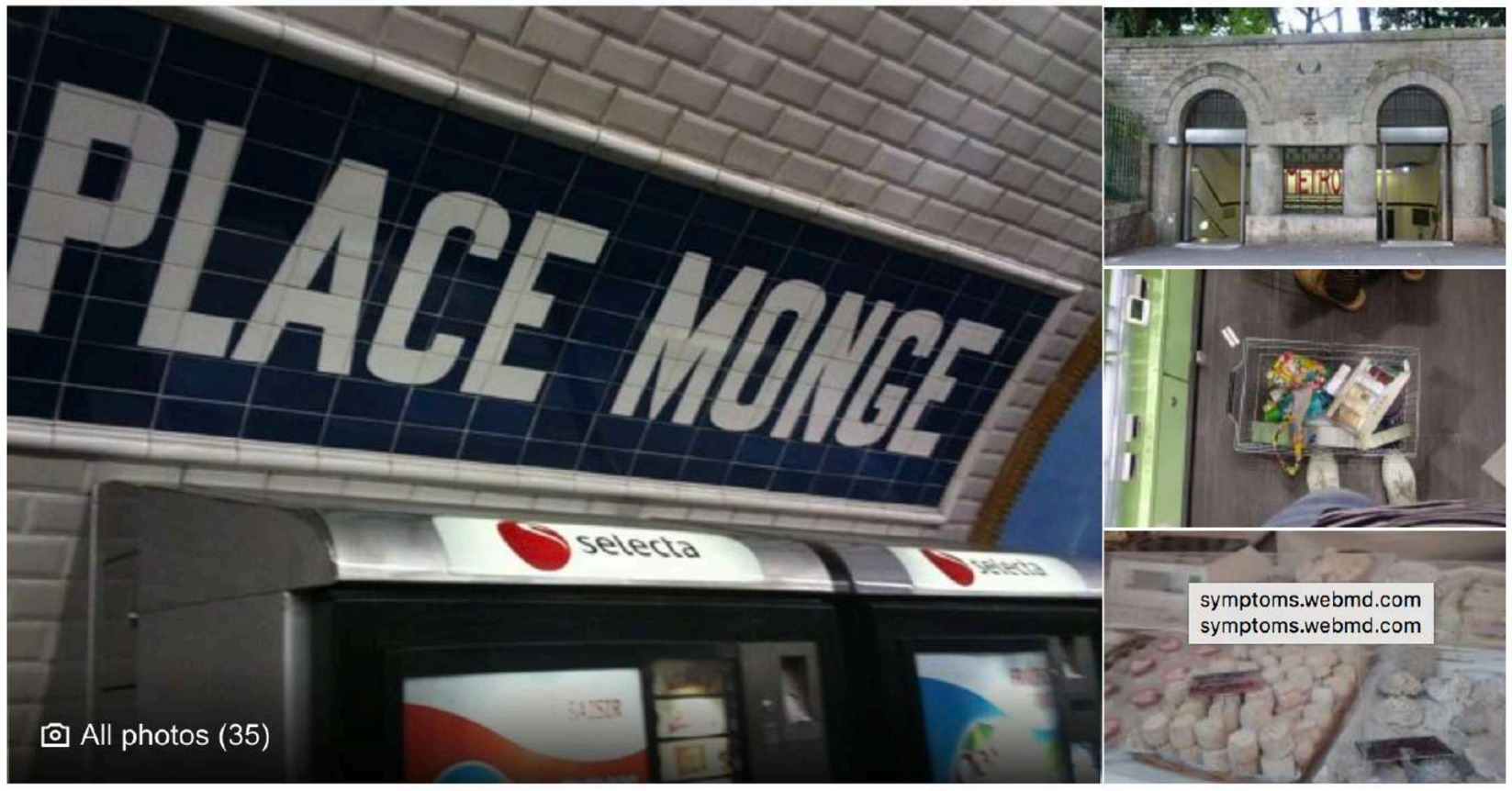
"One of the most amazing streets we have be…

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

Reviewed 11 October 2018

Steve L , Pacific Coast Australia, Australia via mobile

Read all 84 reviews



Origins: Monge Problem (1781)

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)

66 MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

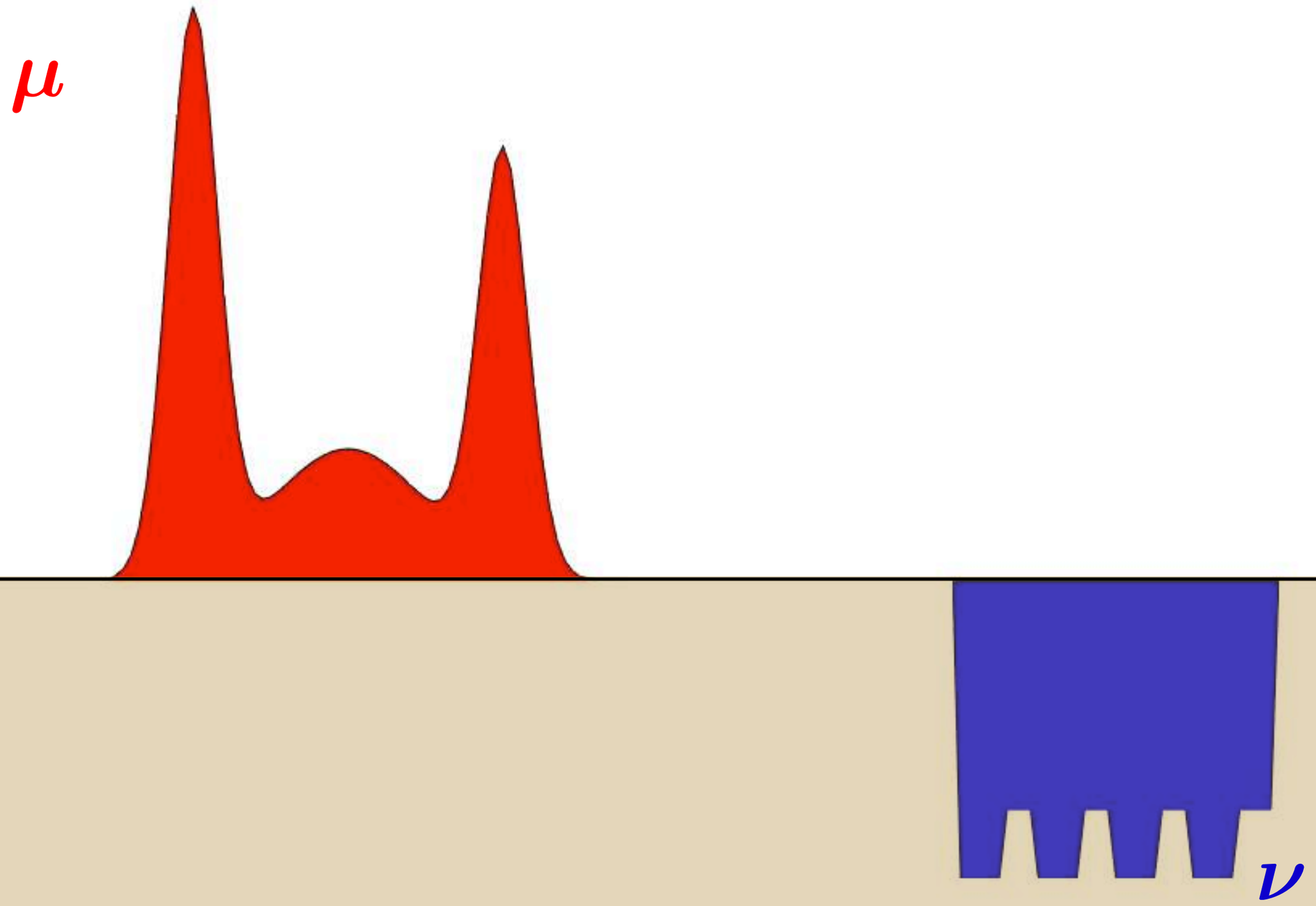
SUR LA

THÉORIE DES DÉBLAIS

When one has to bring earth from one place to another…

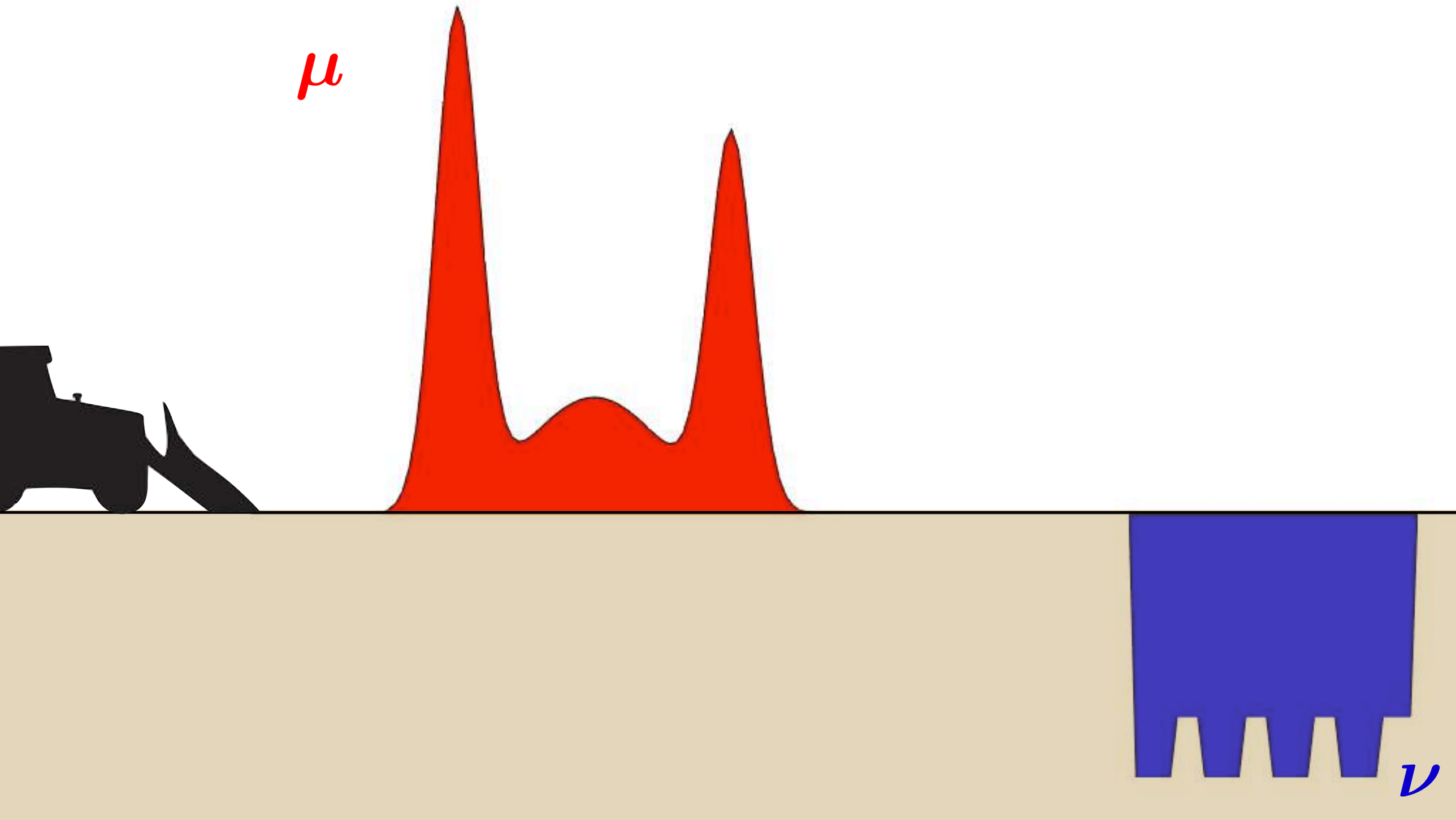
LORSQU'ON doit tranſporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit tranſporter, & le nom de *Remblai* à l'eſpace qu'elles doivent occuper après le tranſport.

Origins: Monge Problem



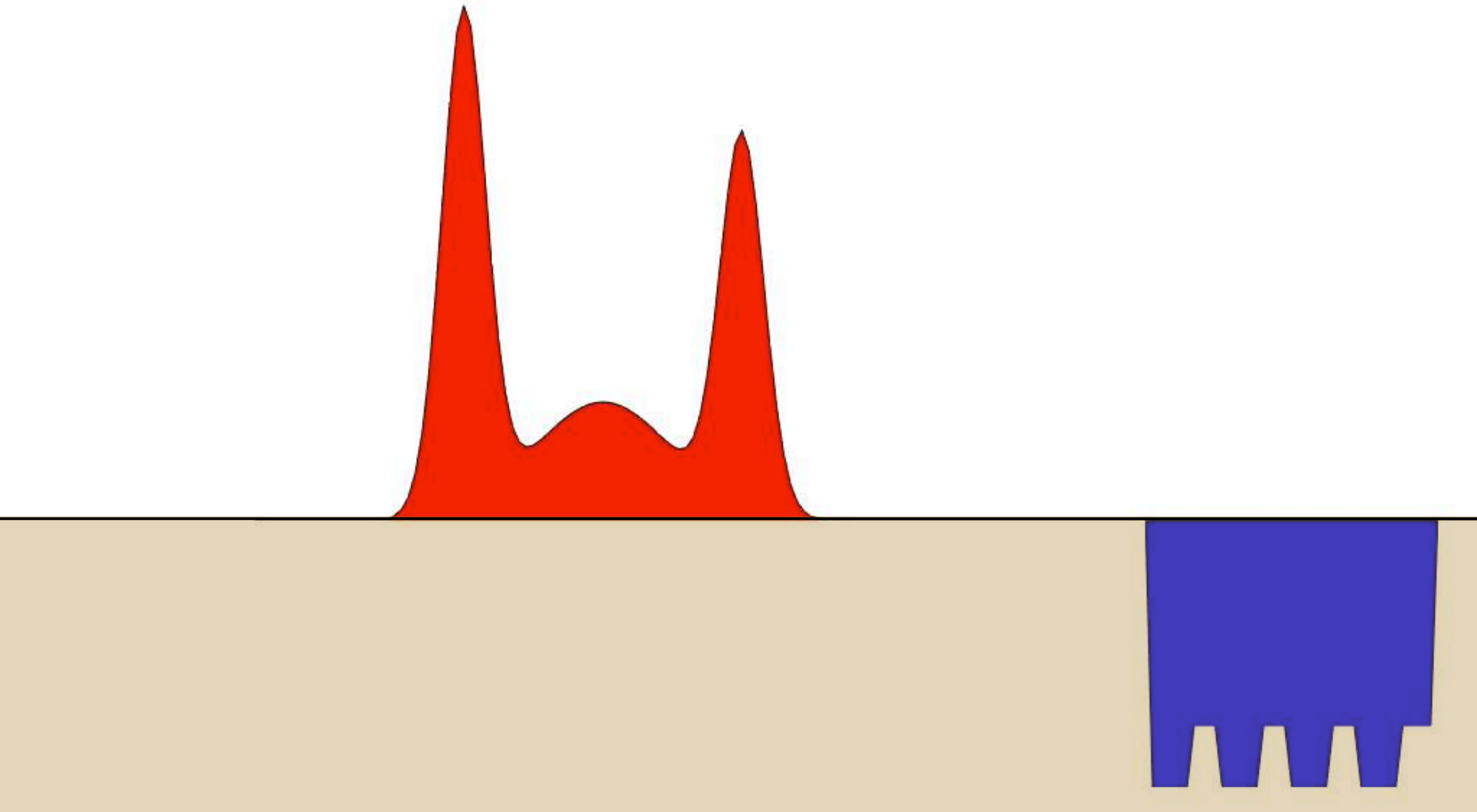
Origins: Monge Problem

In the 21st Century…



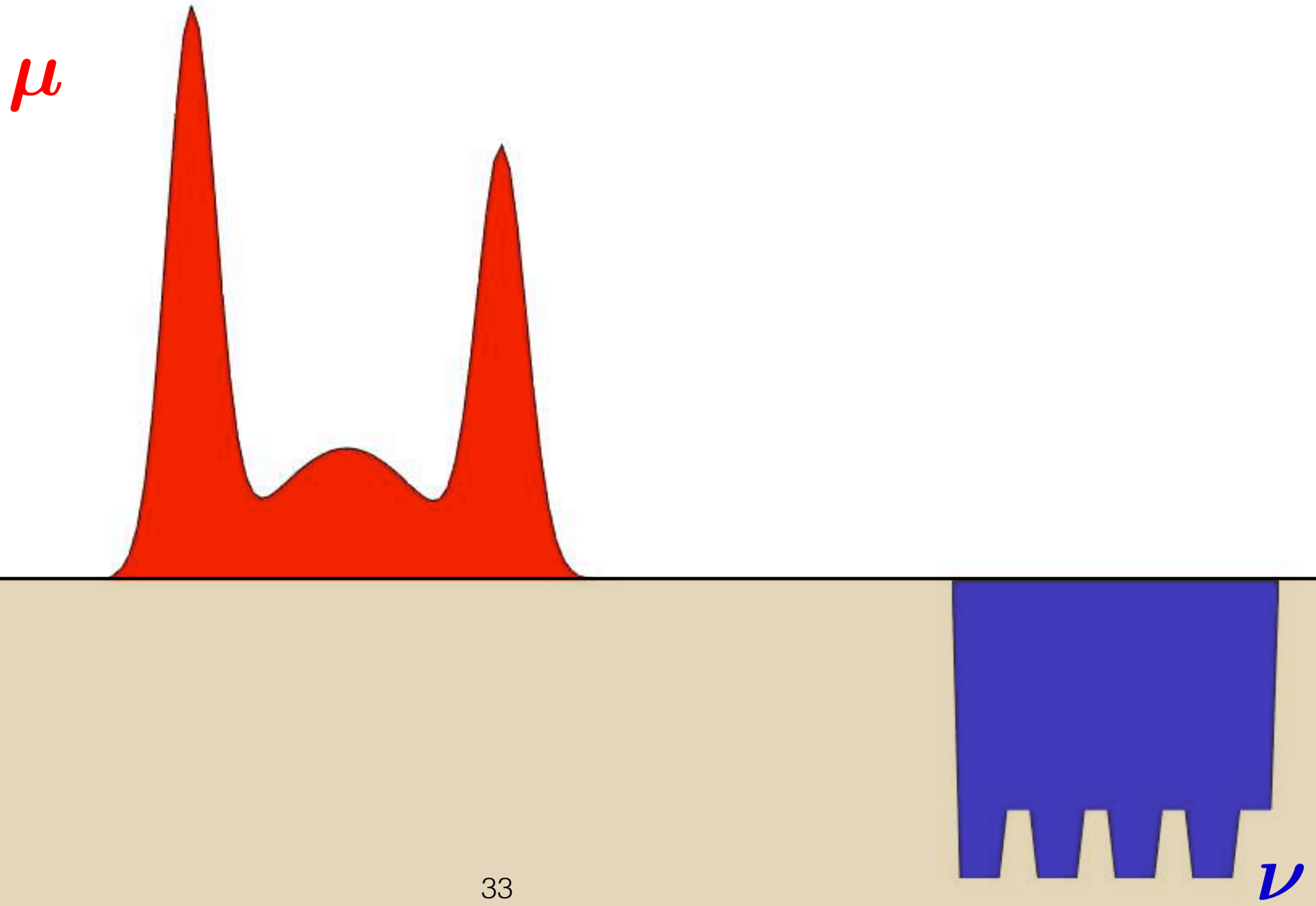
Origins: Monge Problem

In the 21st Century...



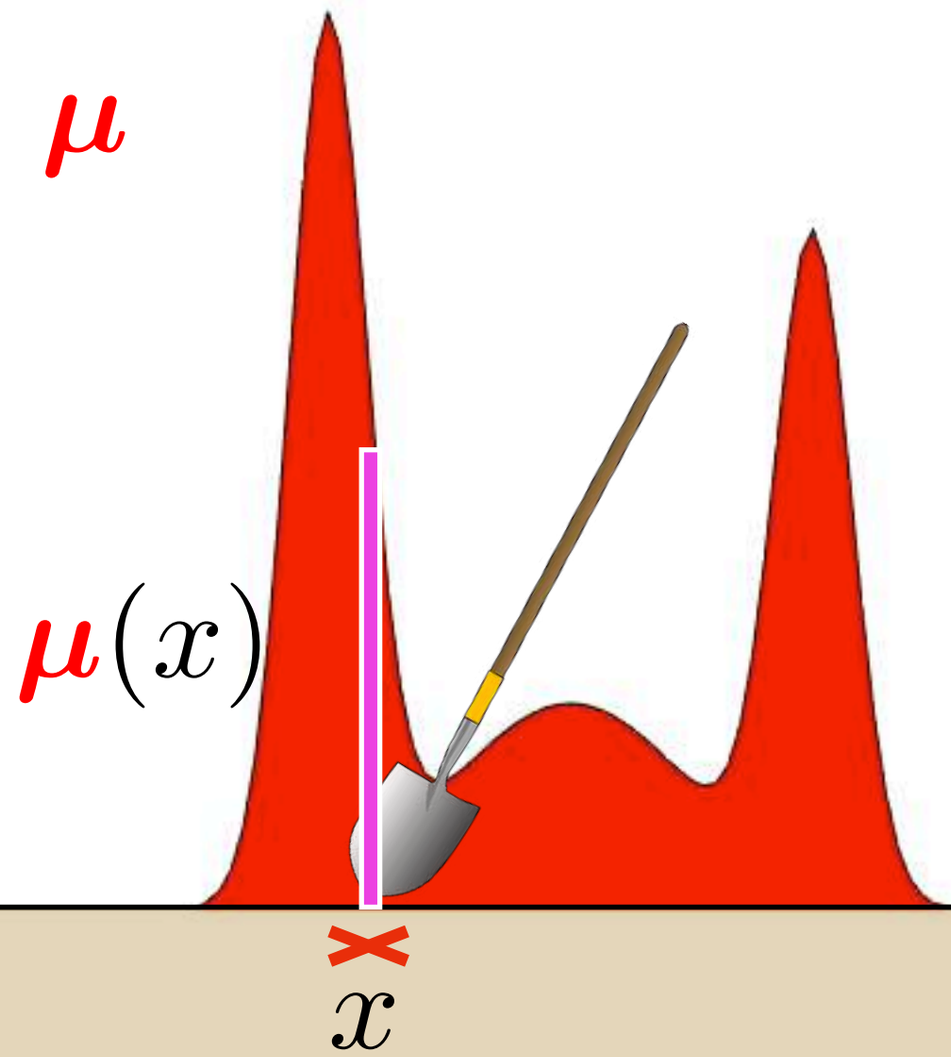
Origins: Monge's Problem

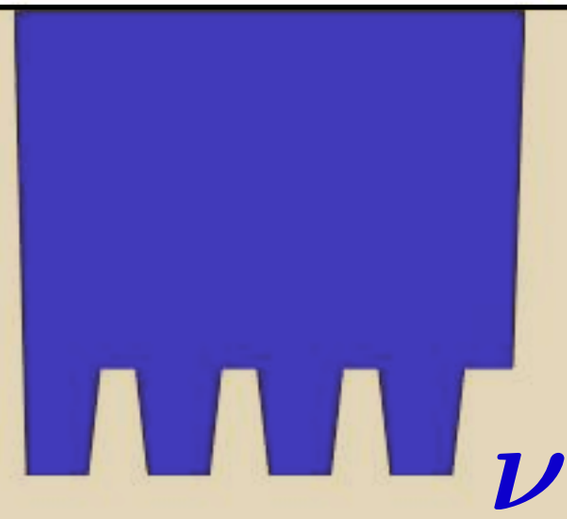
In 1781 however…



Origins: Monge's Problem

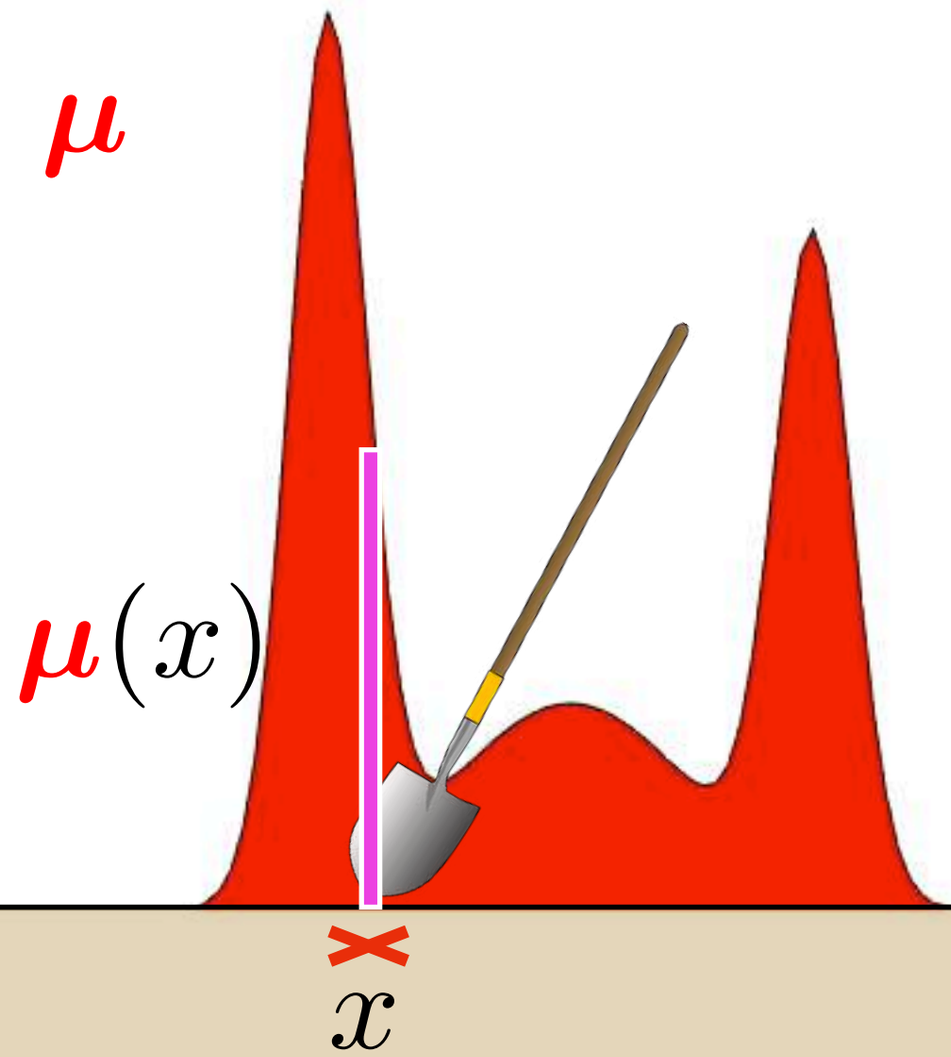
In 1781 however…



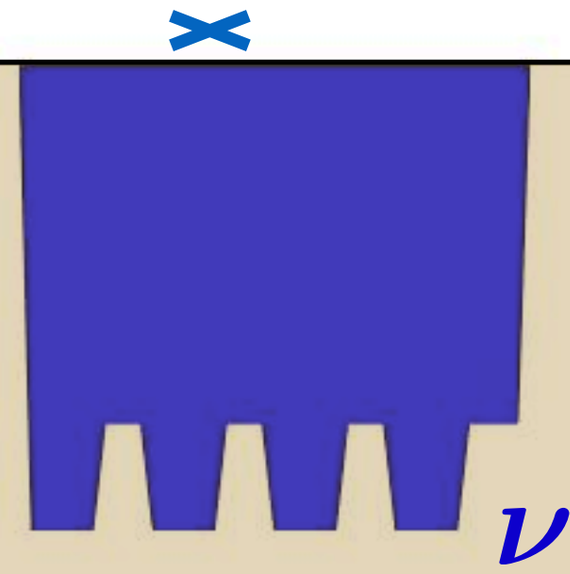


Origins: Monge's Problem

In 1781 however…

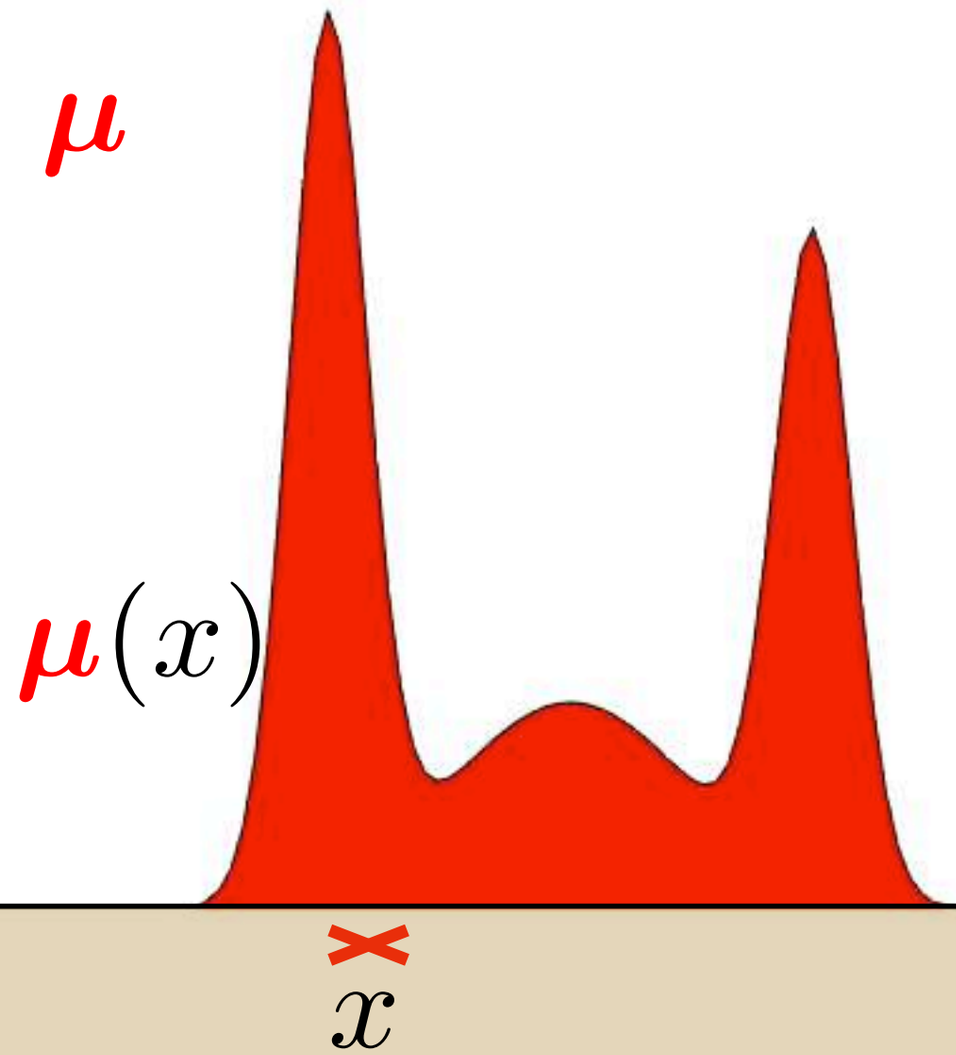


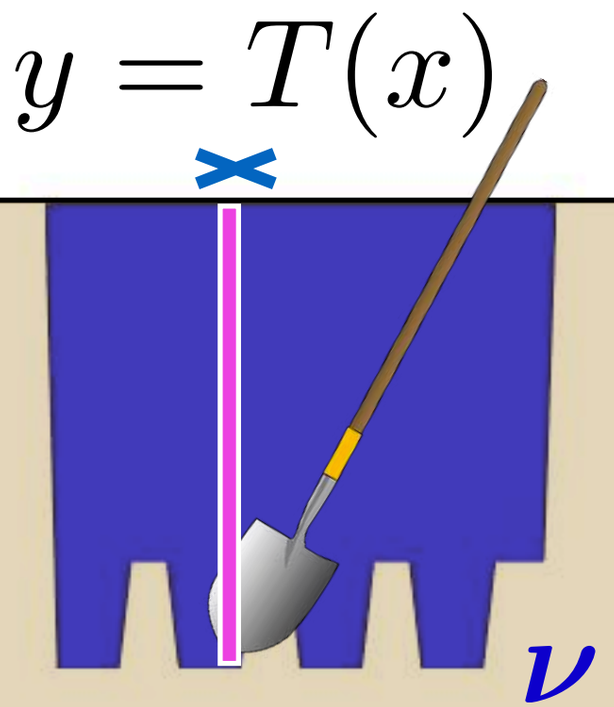
$$y = T(x)$$



Origins: Monge's Problem

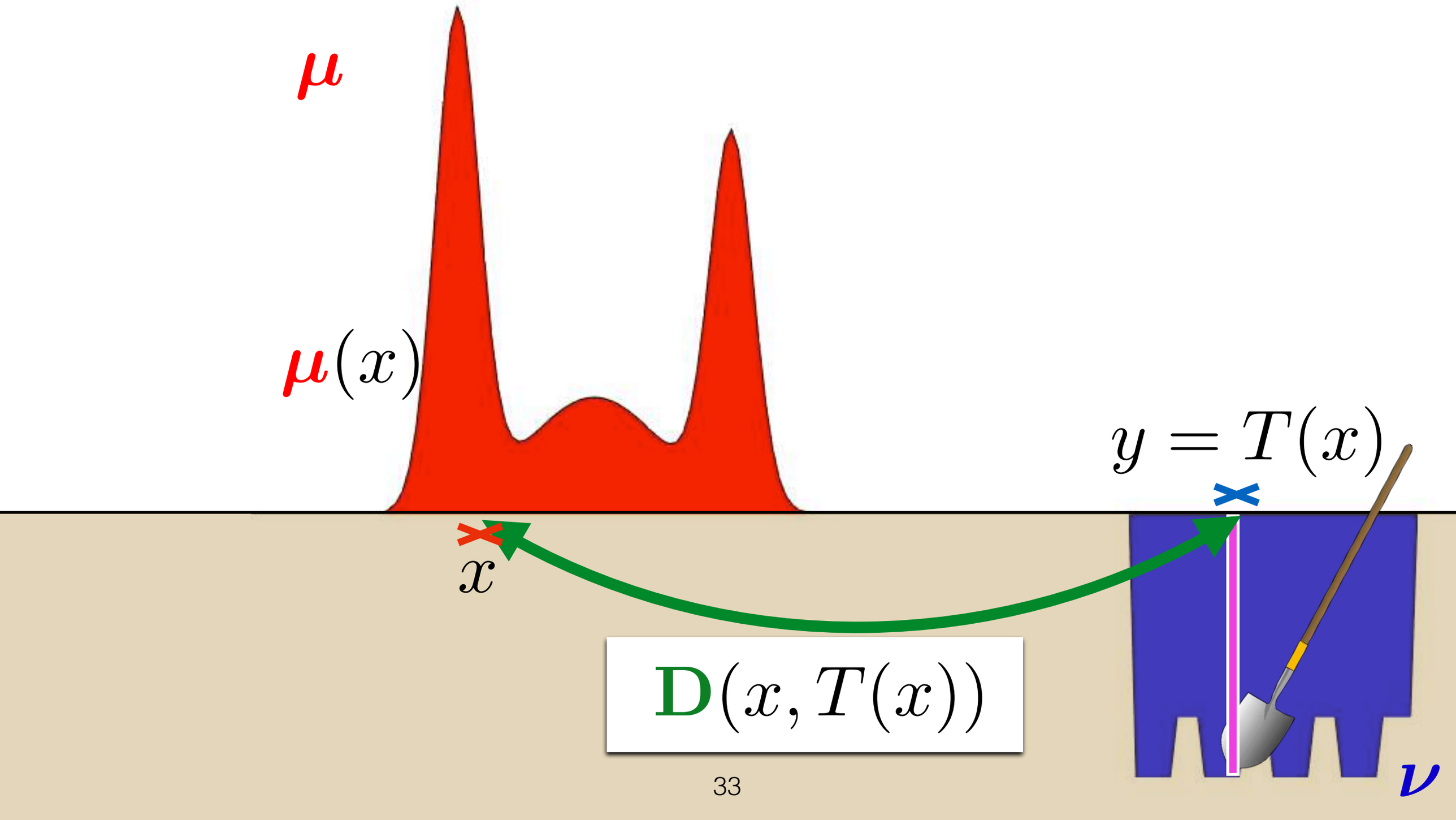
In 1781 however…





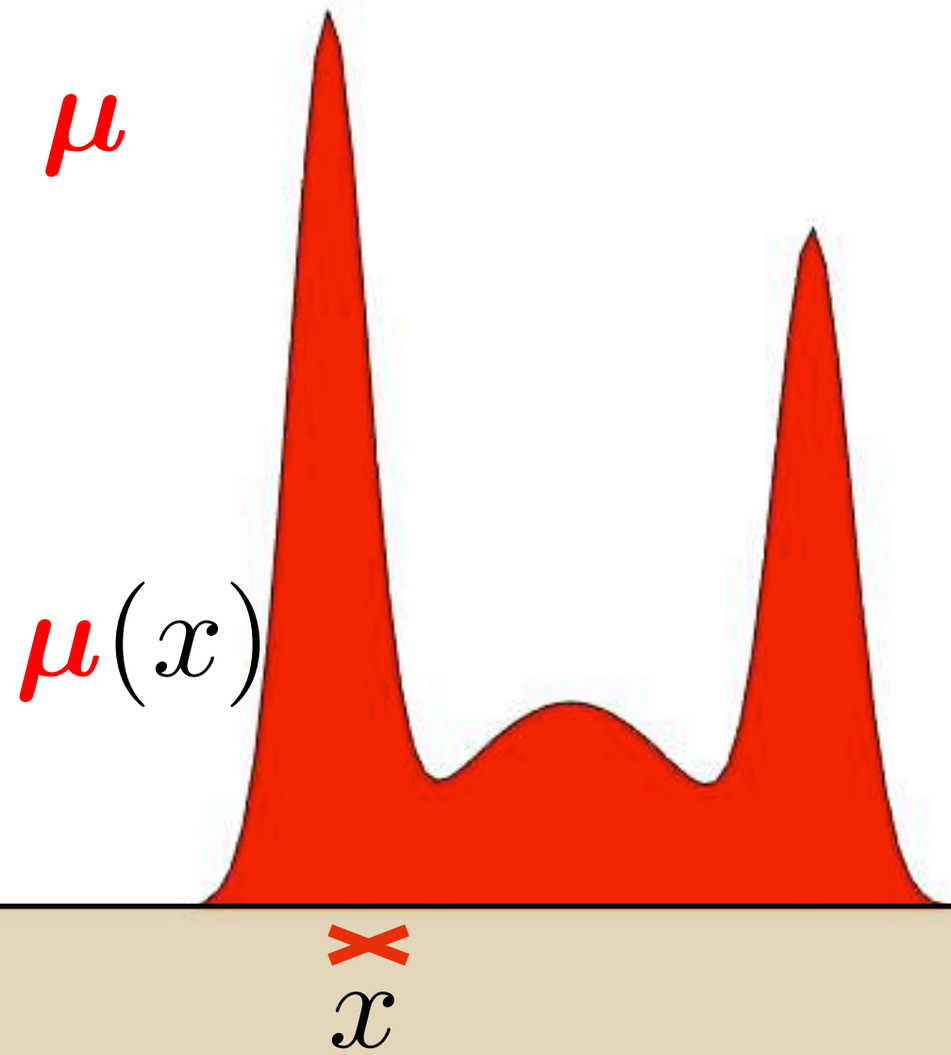
Origins: Monge's Problem

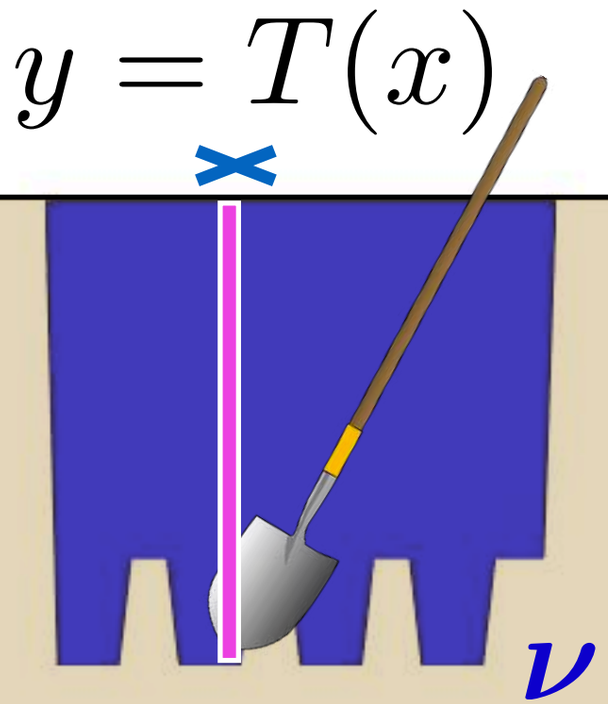
In 1781 however…



Origins: Monge's Problem

In 1781 however…

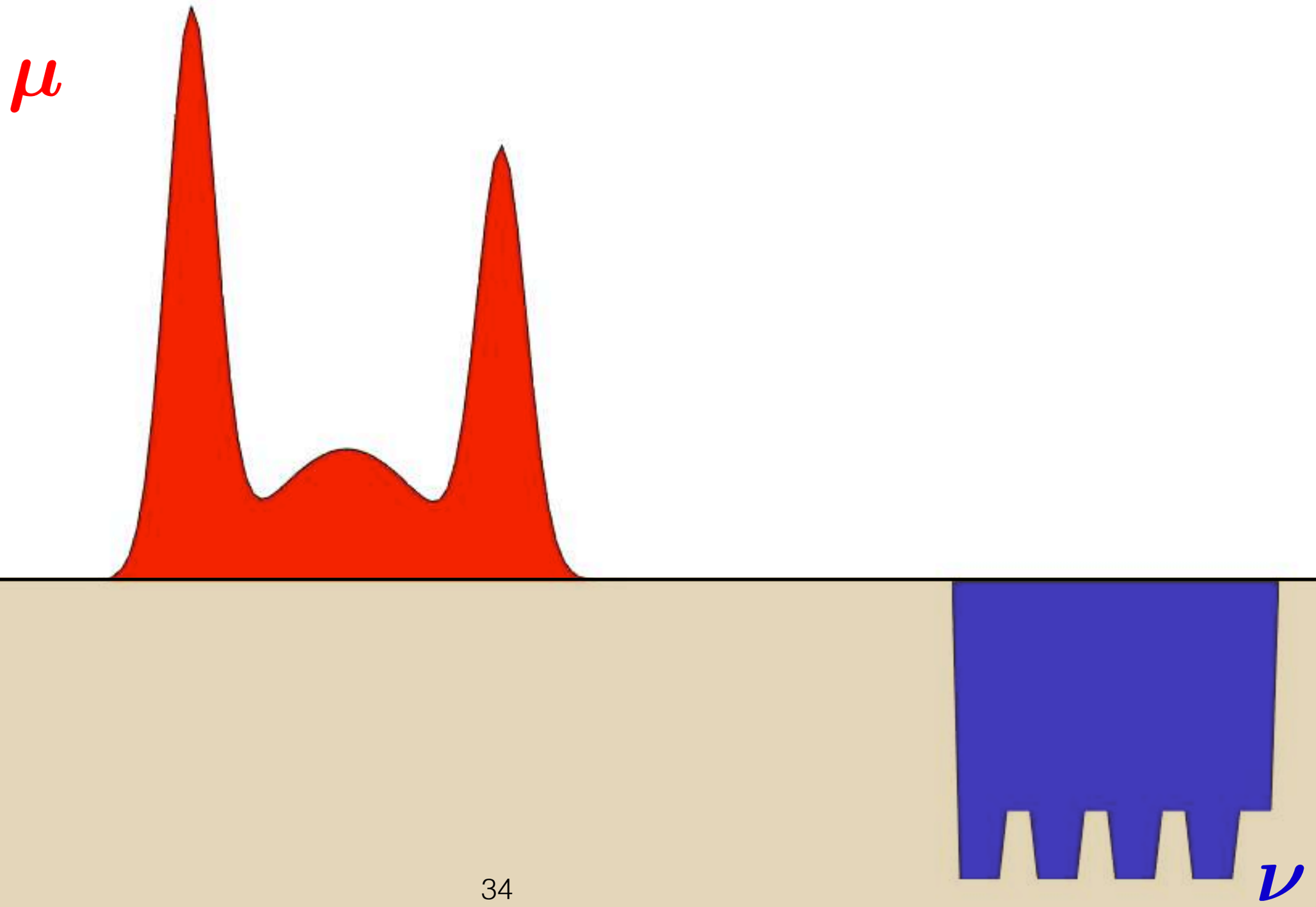




work: $\mu(x)D(x, T(x))$

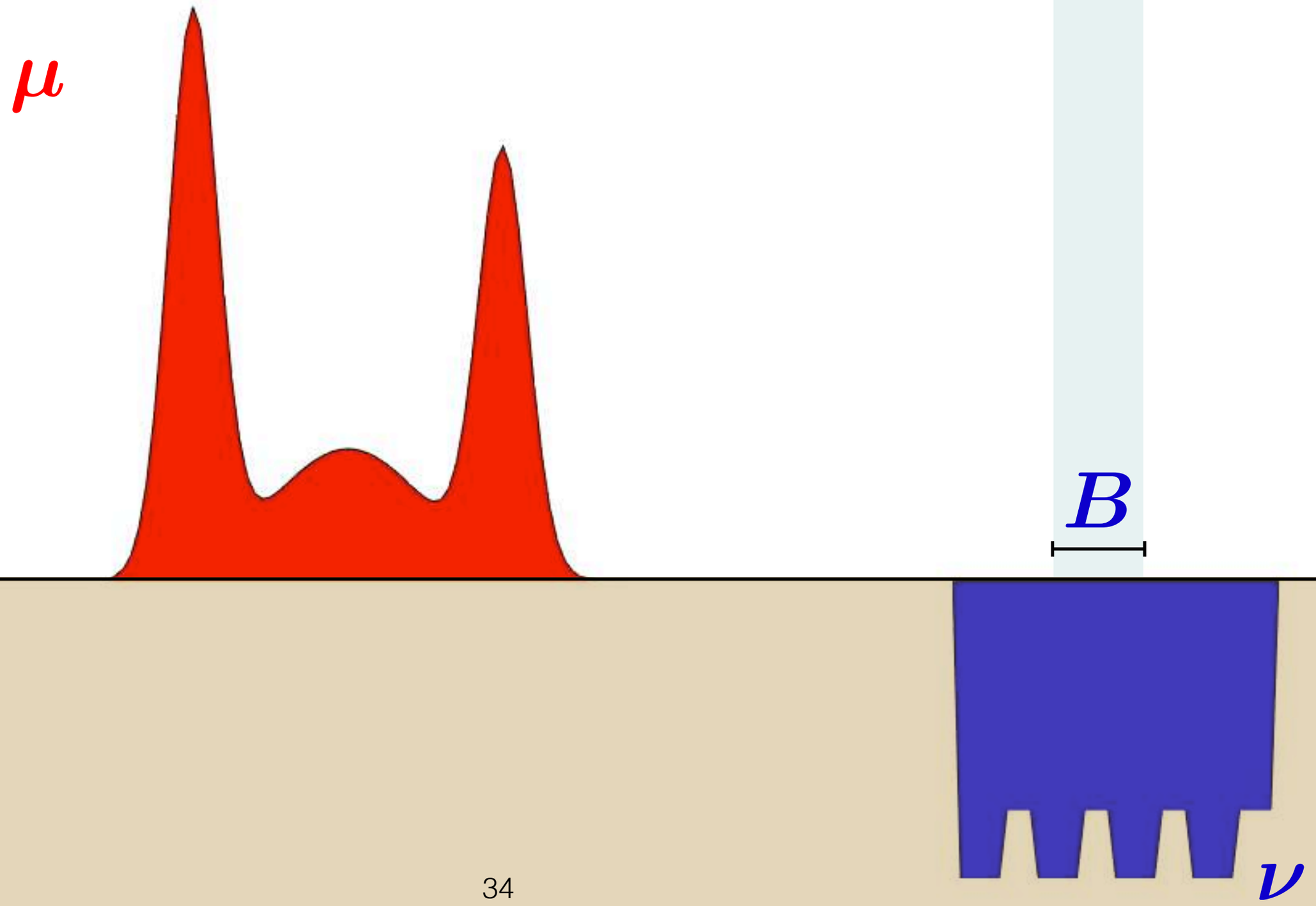
Origins: Monge's Problem

T must map red to blue.



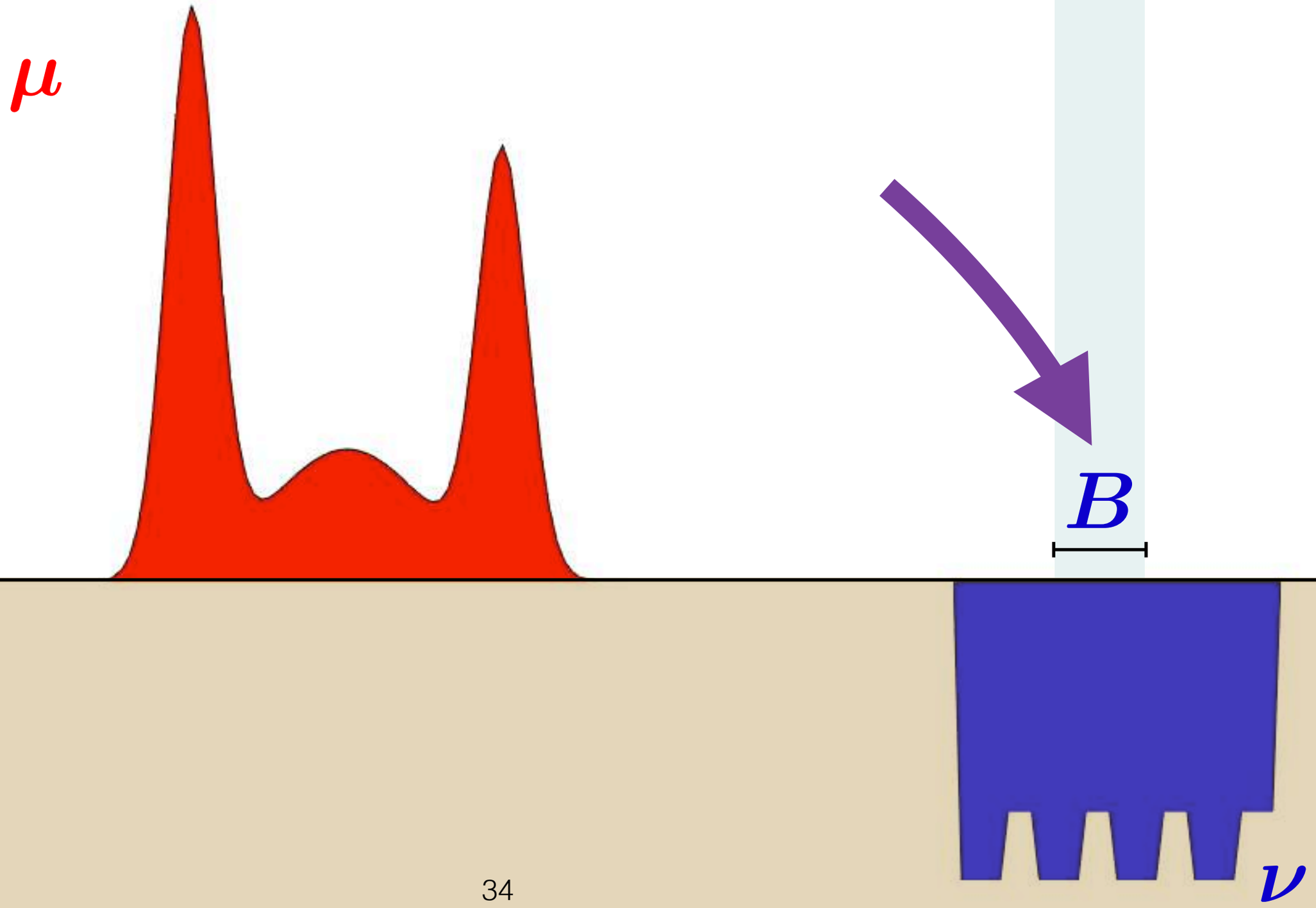
Origins: Monge's Problem

T must map red to blue.



Origins: Monge's Problem

T must map red to blue.

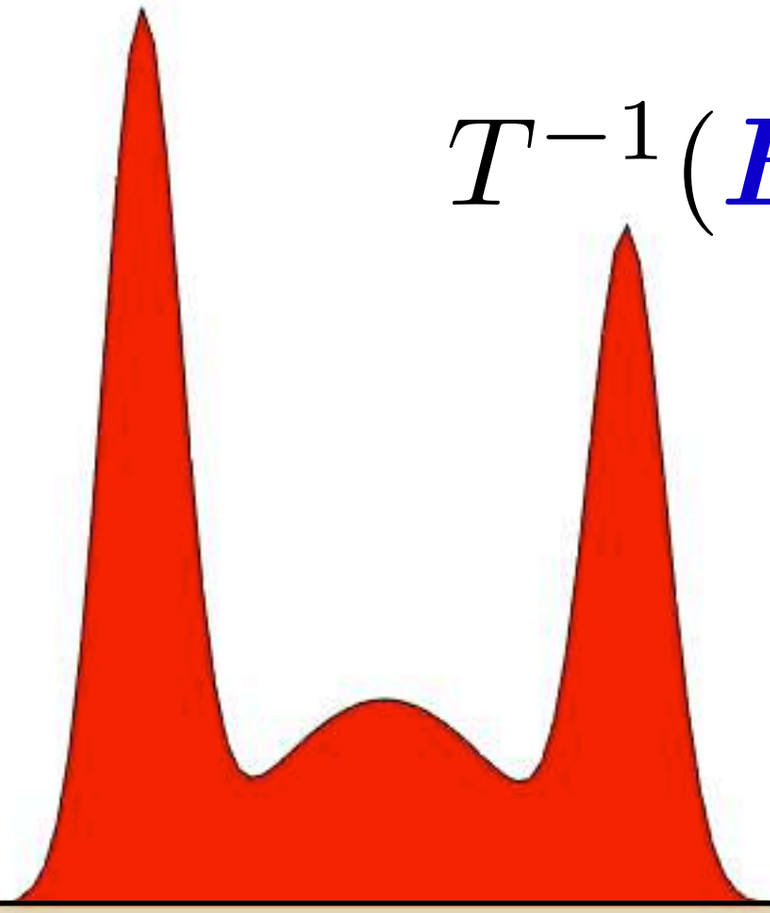


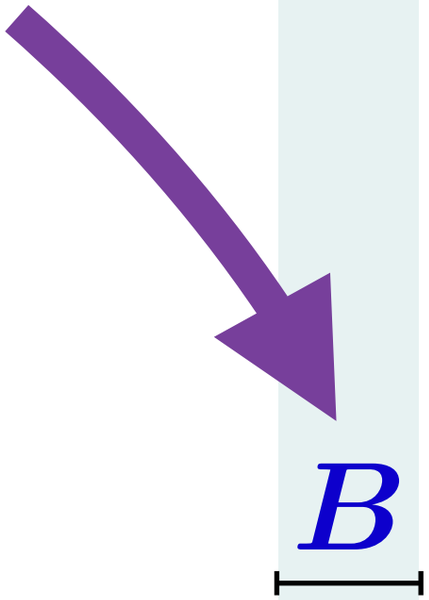
Origins: Monge's Problem

T must map red to blue.

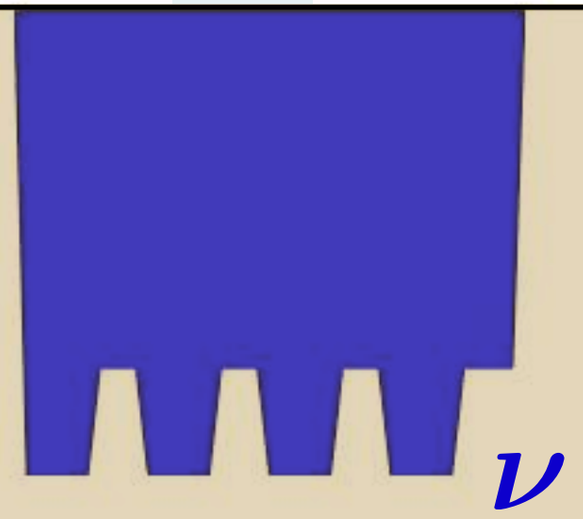
μ

$$T^{-1}(\boldsymbol{B}) = \{x | T(x) \in \boldsymbol{B}\}$$





B



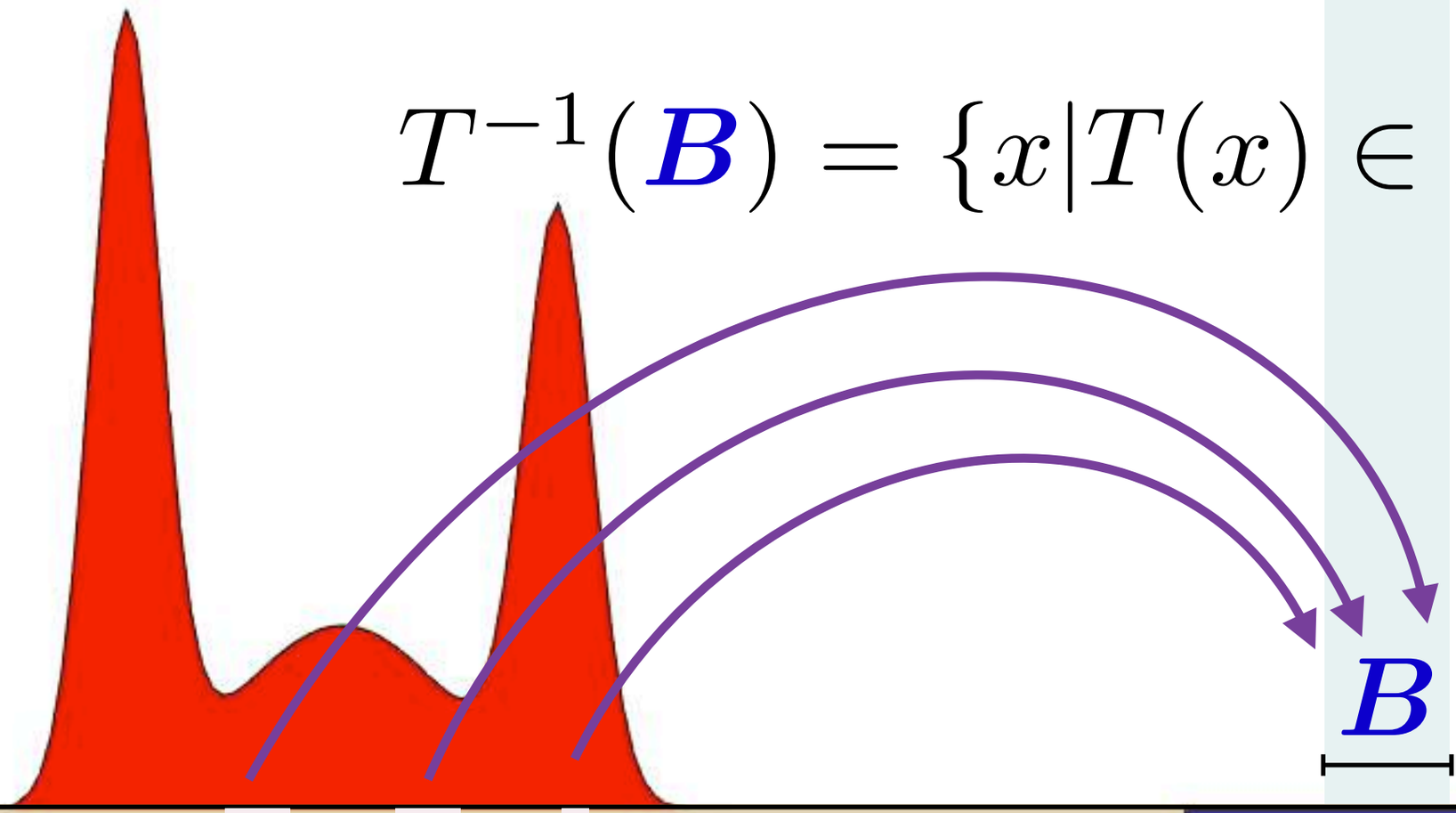
ν

Origins: Monge's Problem

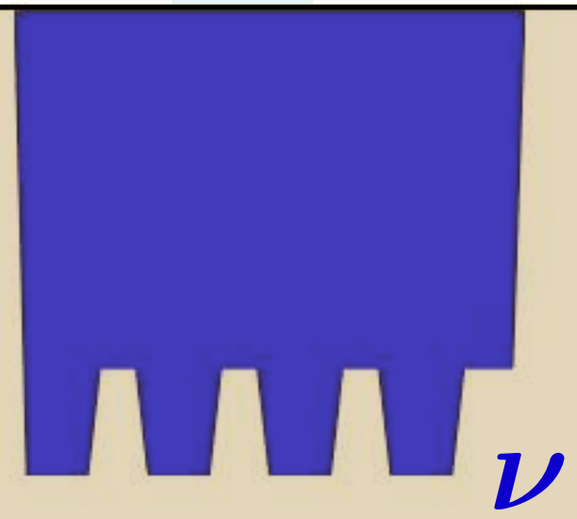
T must map red to blue.

μ

$$T^{-1}(\boldsymbol{B}) = \{x | T(x) \in \boldsymbol{B}\}$$

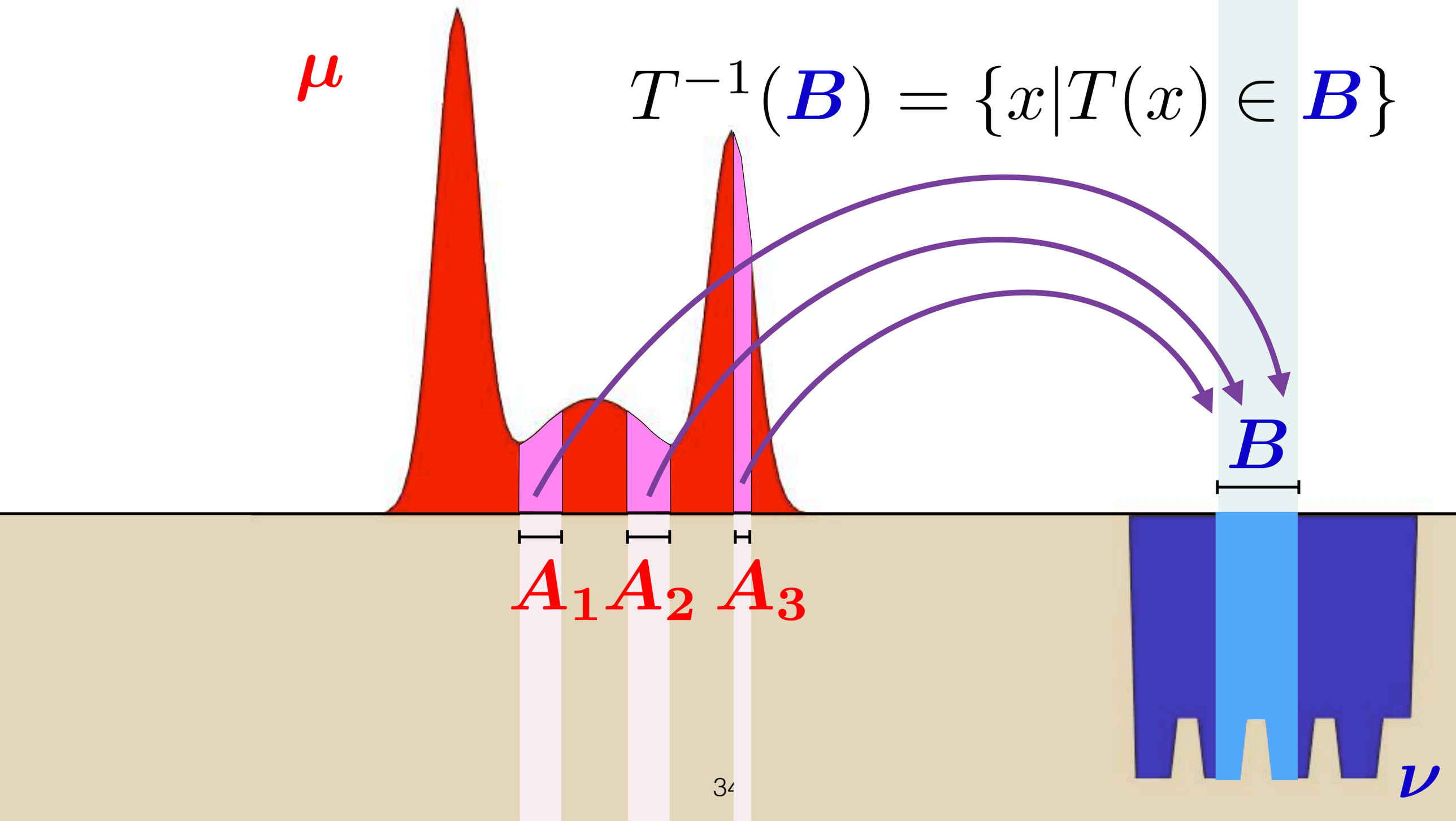


A_1 A_2 A_3



Origins: Monge's Problem

T must map red to blue.

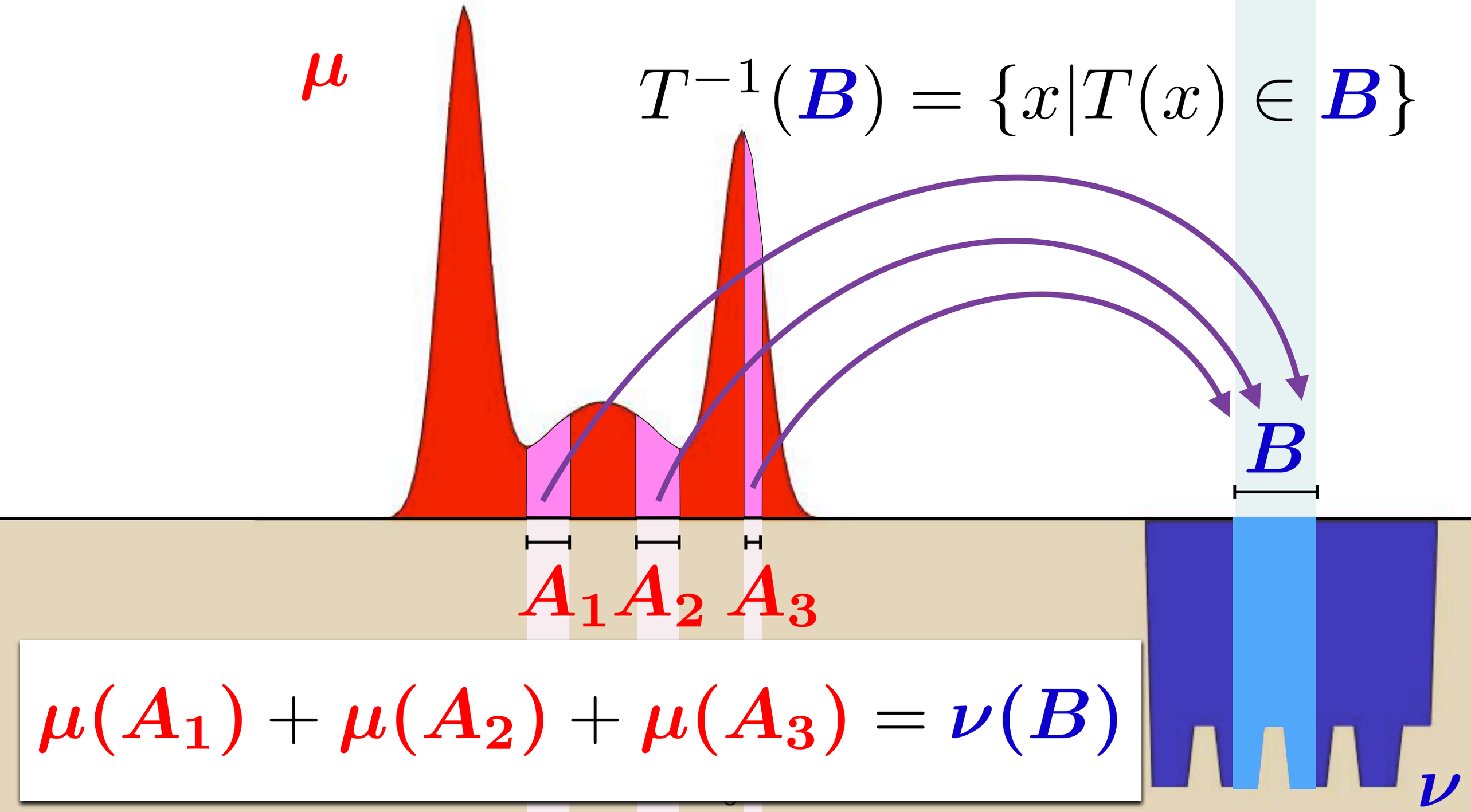


Origins: Monge's Problem

T must map red to blue.

μ

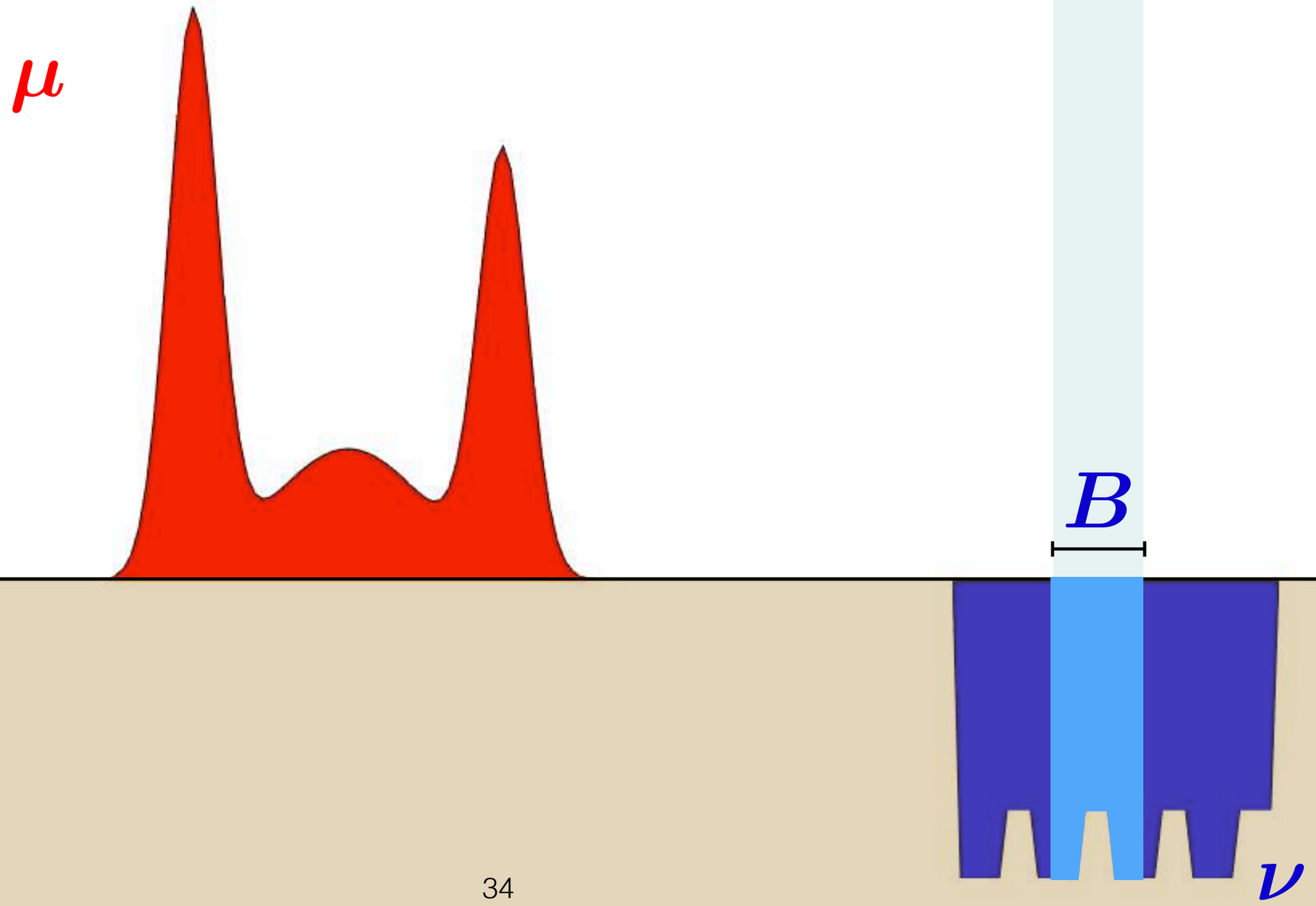
$$T^{-1}(\boldsymbol{B}) = \{x | T(x) \in \boldsymbol{B}\}$$



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

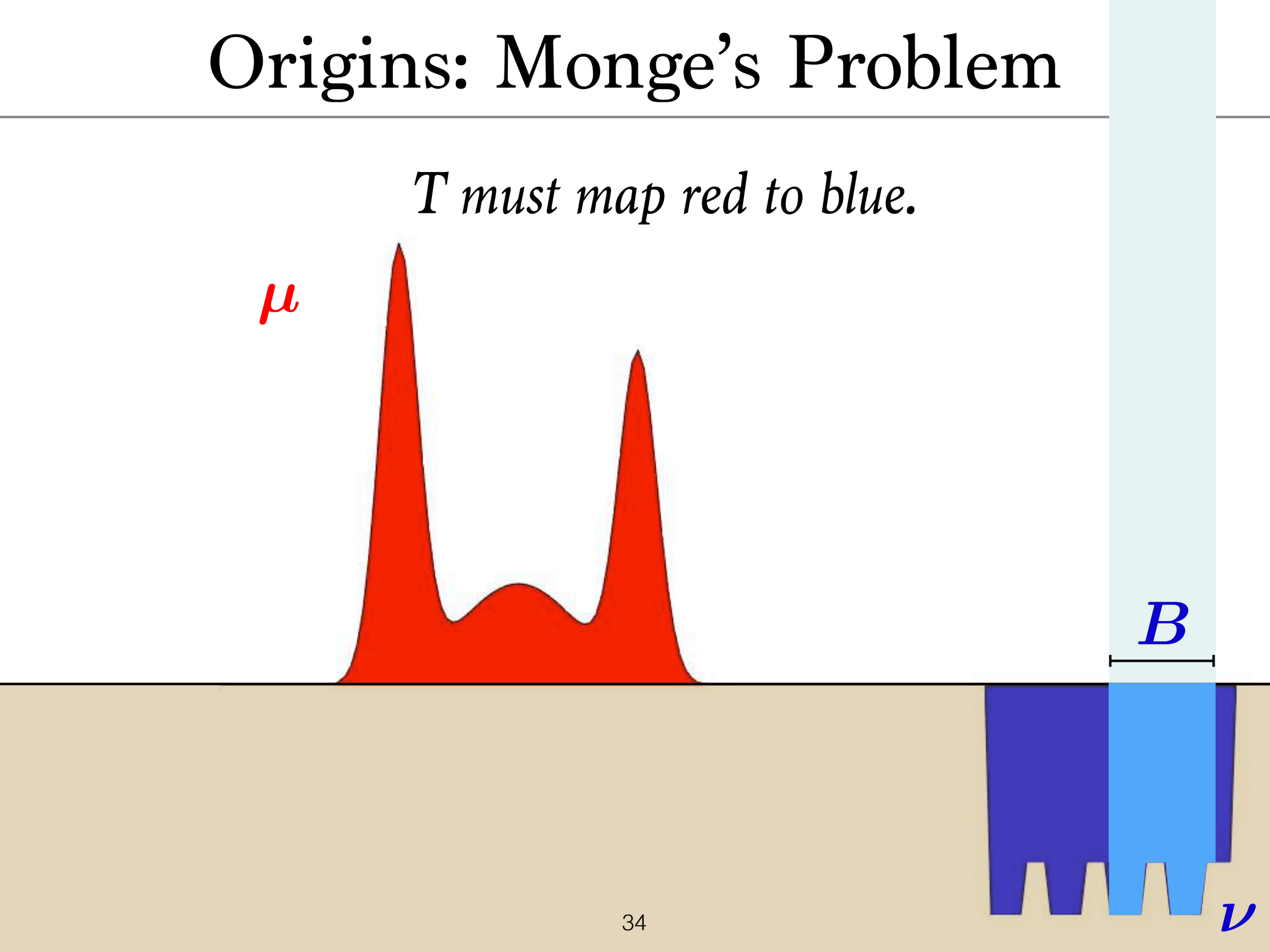
Origins: Monge's Problem

T must map red to blue.



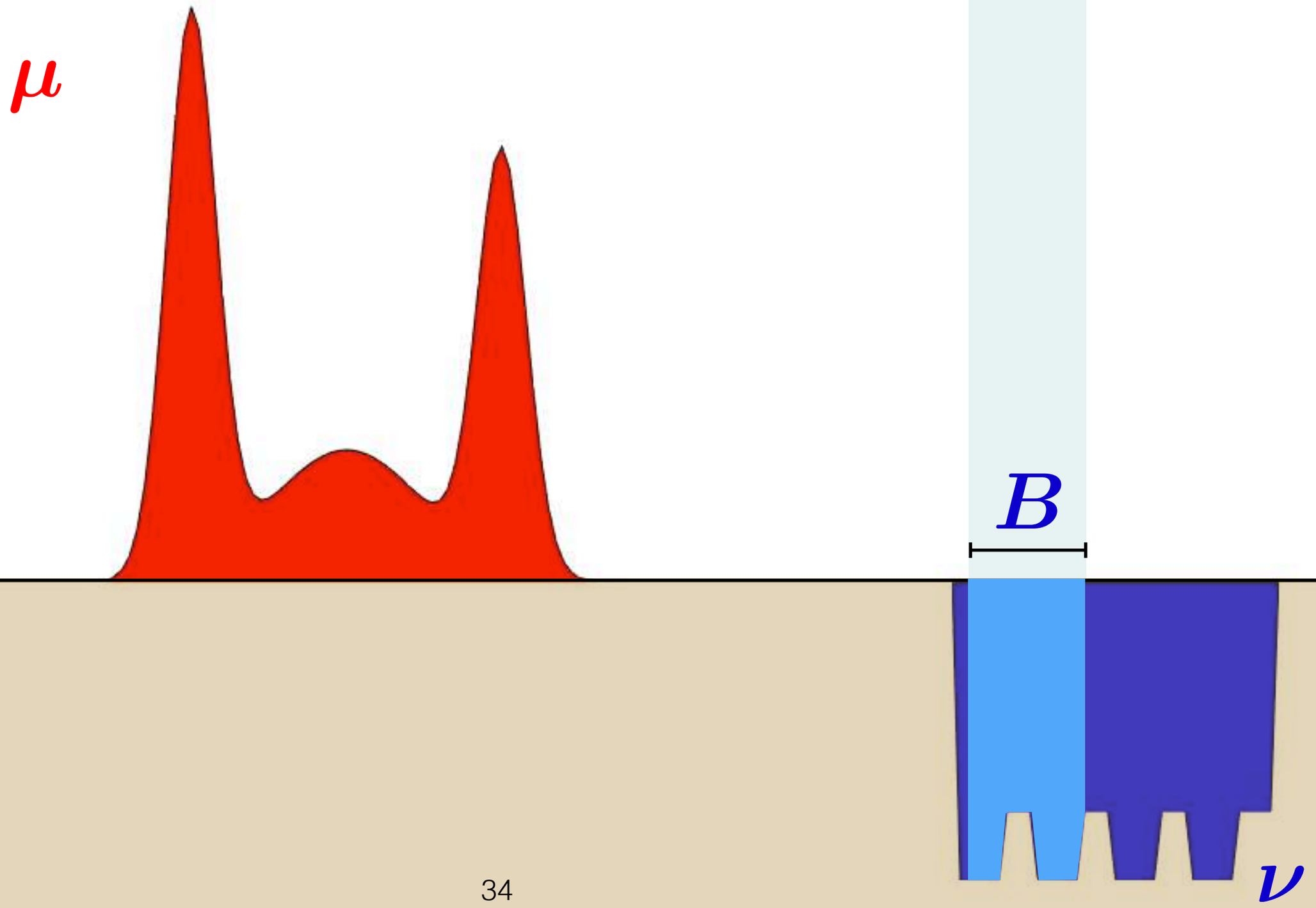
Origins: Monge's Problem

T must map red to blue.



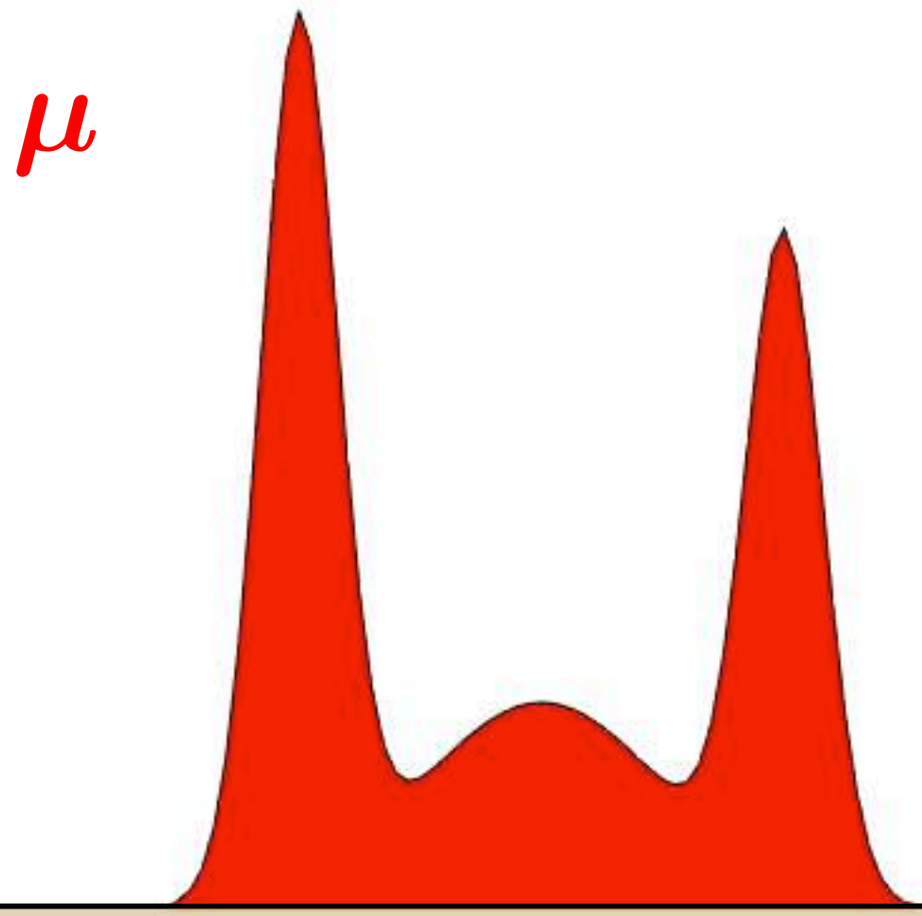
Origins: Monge's Proble

T must map red to blue.



Origins: Monge's Proble

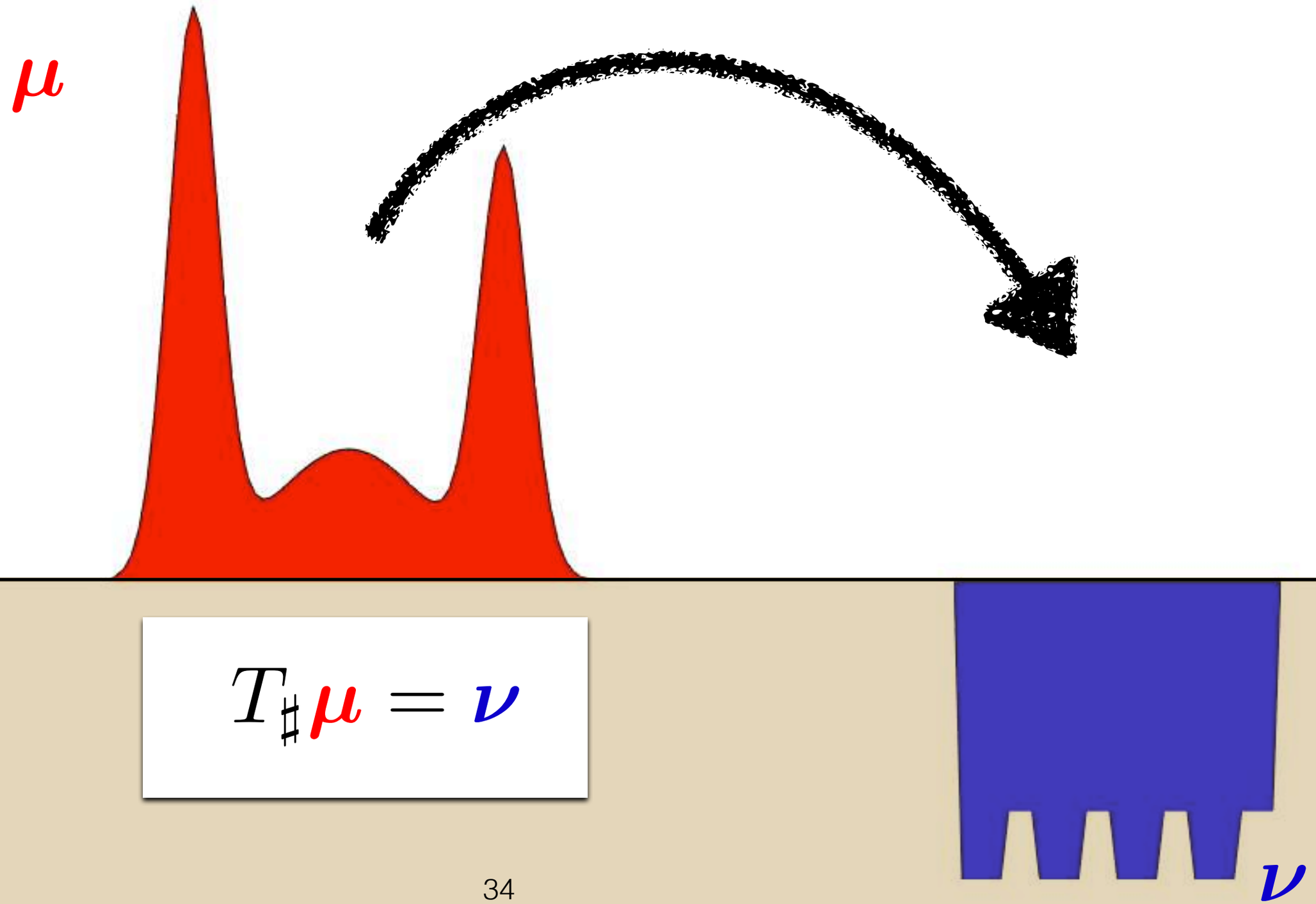
T must map red to blue.



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

Origins: Monge's Problem

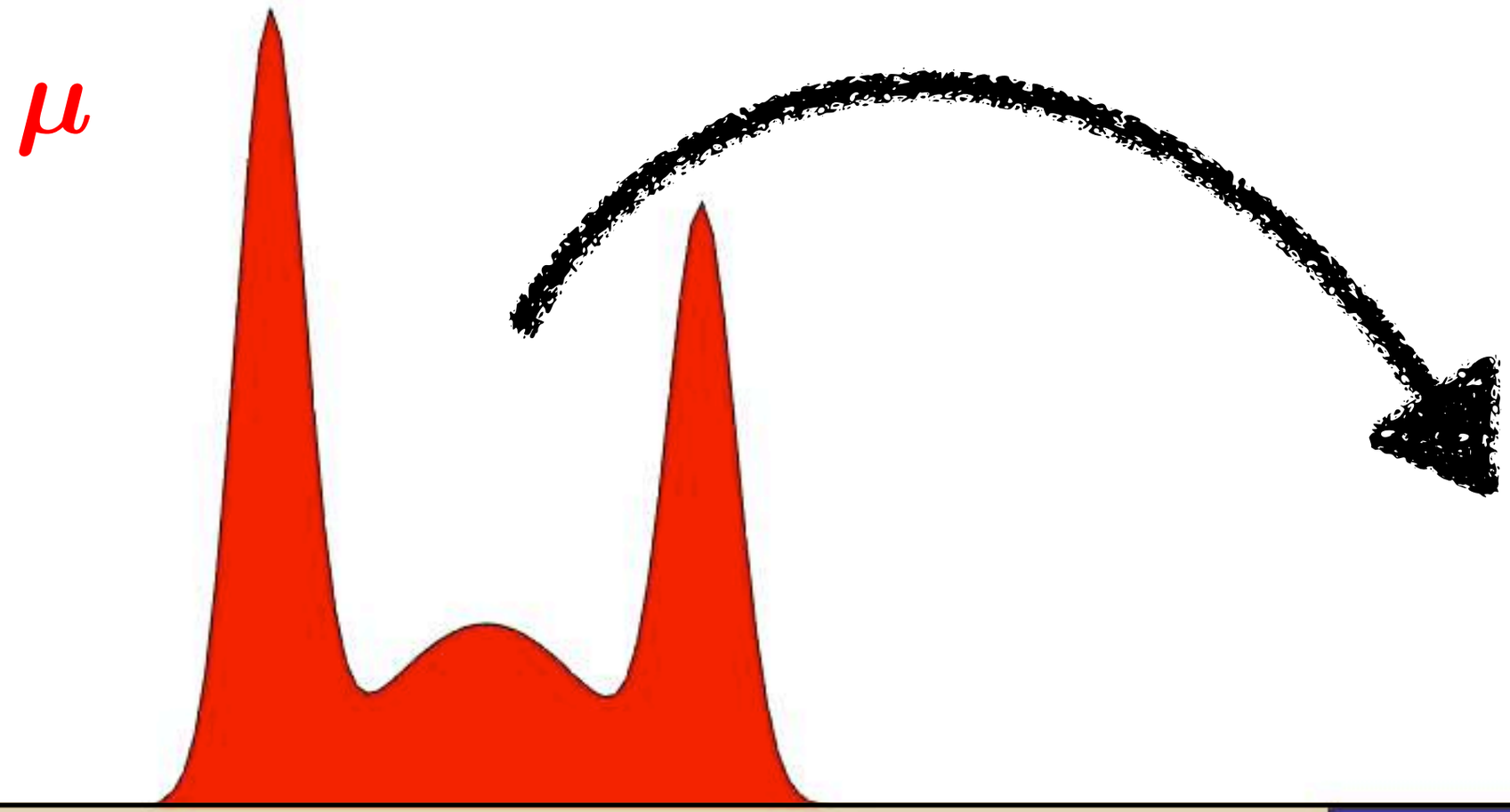
T must **push-forward** *the red measure towards the blue*



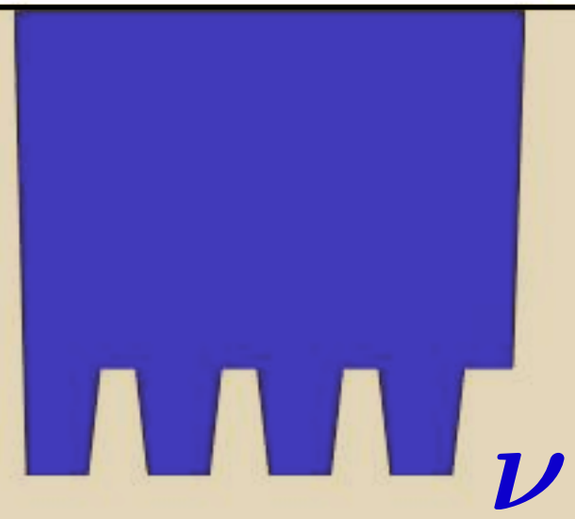
$$T_\sharp \mu = \nu$$

Origins: Monge's Problem

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



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



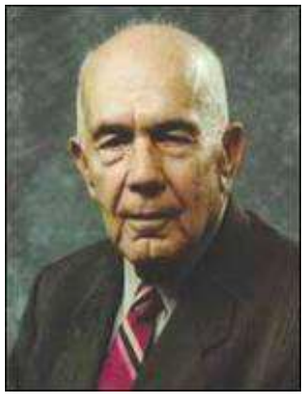
Kantorovich Problem

Kantorovich

1939

Tolstoi

1930



Hitchcock

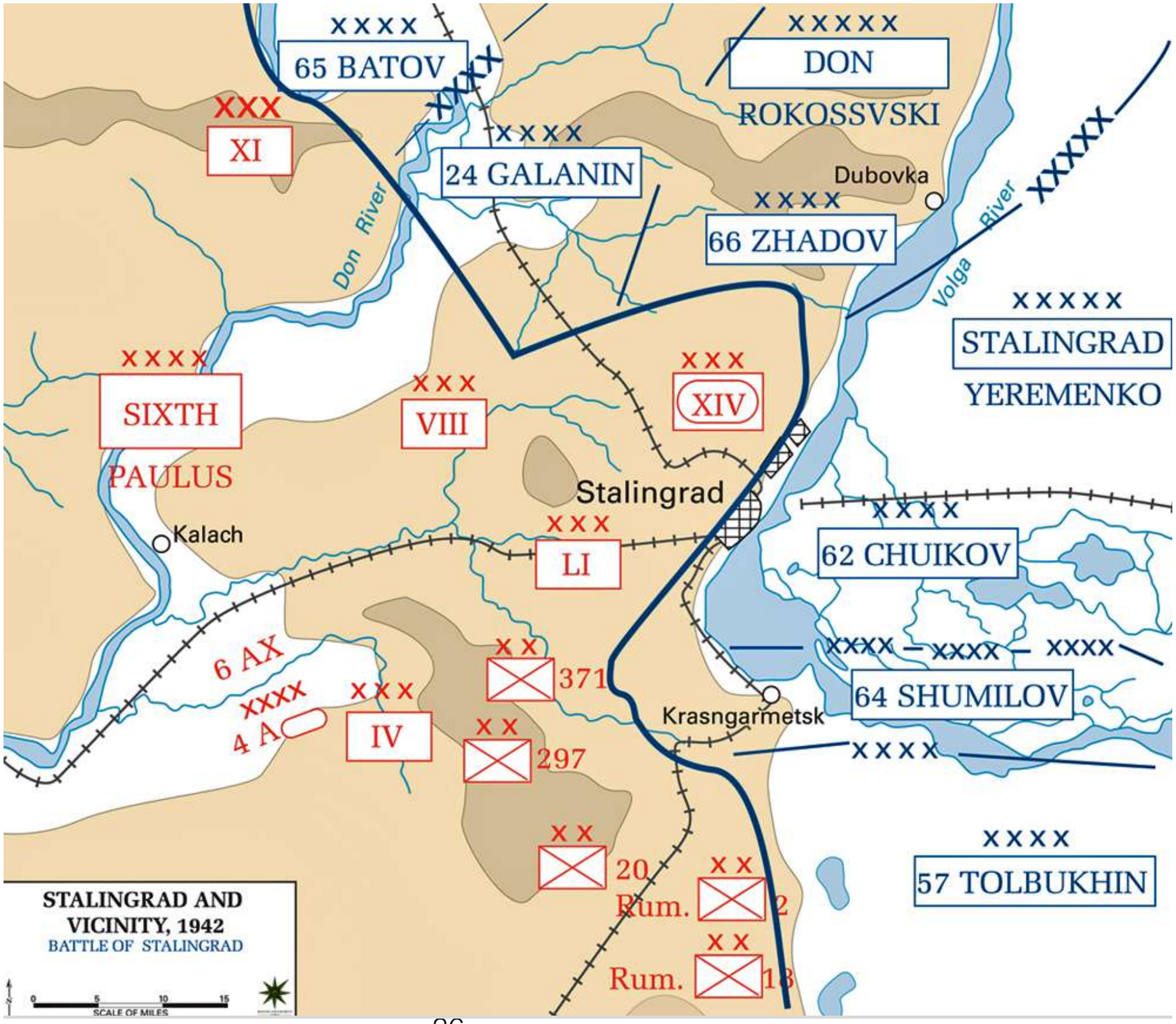
THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

By Frank L. Hitchcock

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

1941

Kantorovich Problem



Kantorovich Problem

60

90

120

90

150

90

Kantorovich Problem *à la française*

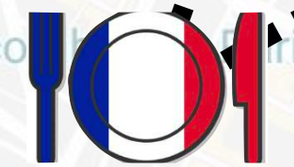
60

120

90

90

150



90

Kantorovich Problem

60

90

120

90

150

90

Kantorovich Problem

24 18 18

36 27 27

120

90

90

45 45 60

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

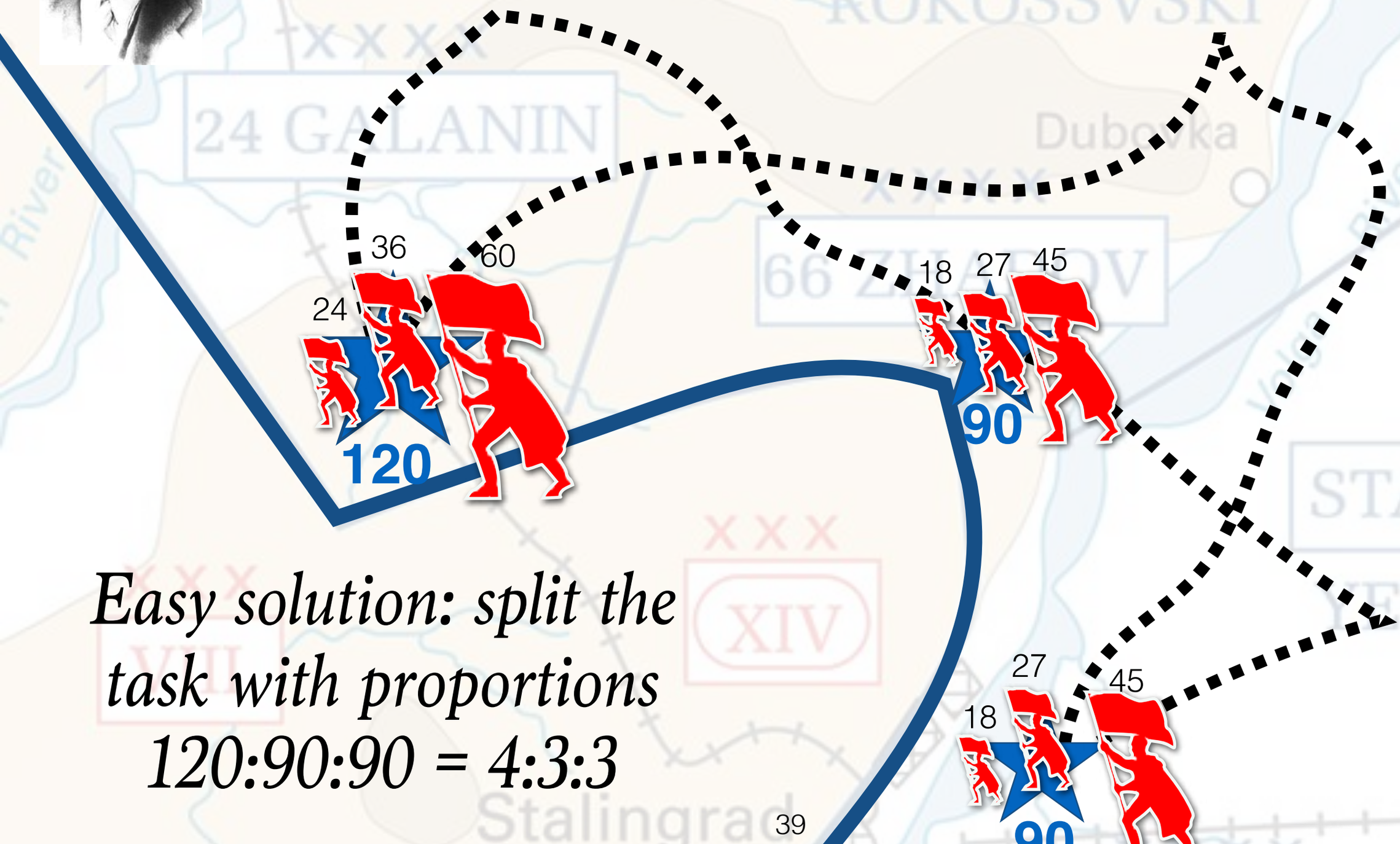
120:90:90 = 4:3:3

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

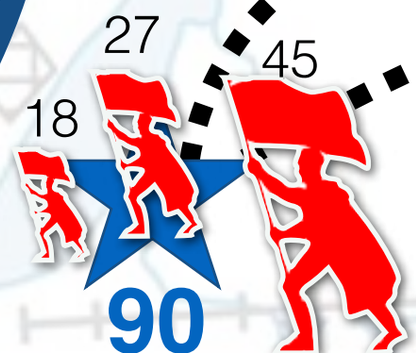
Kantorovich Problem

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

60

90

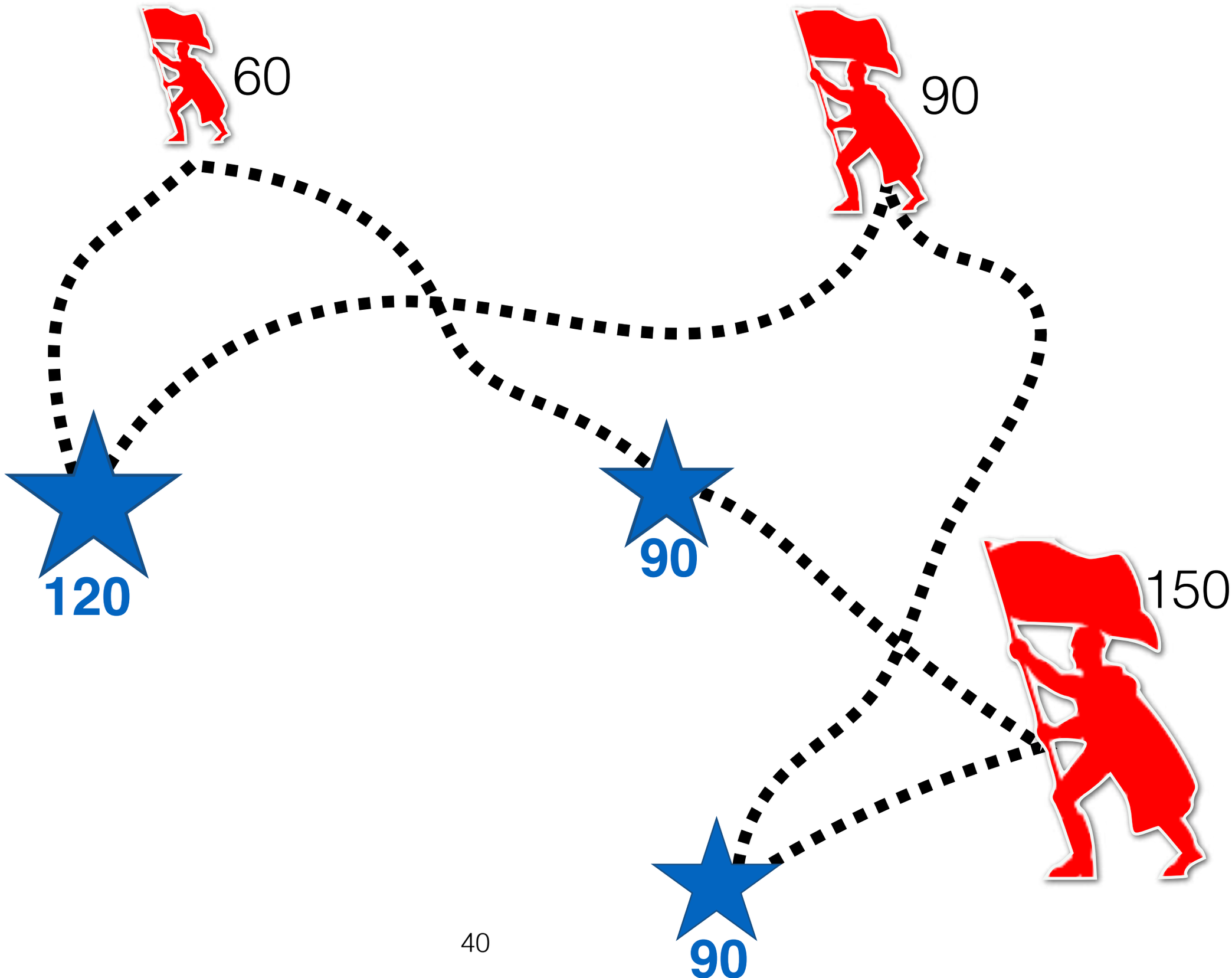
120

90

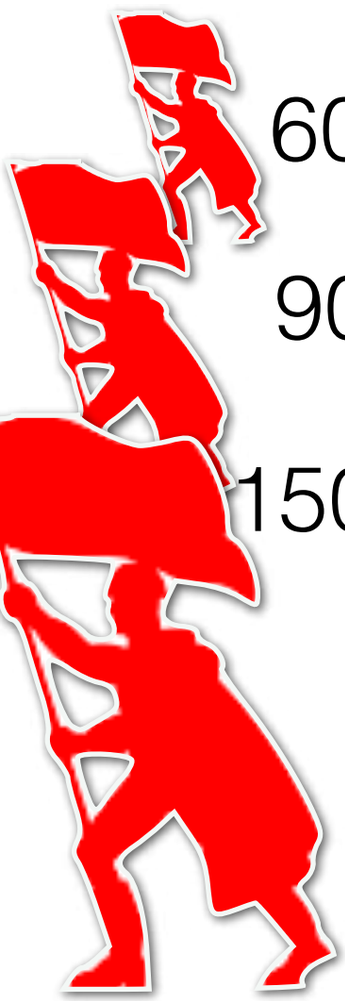
150

90

Kantorovich Problem



Kantorovich Problem

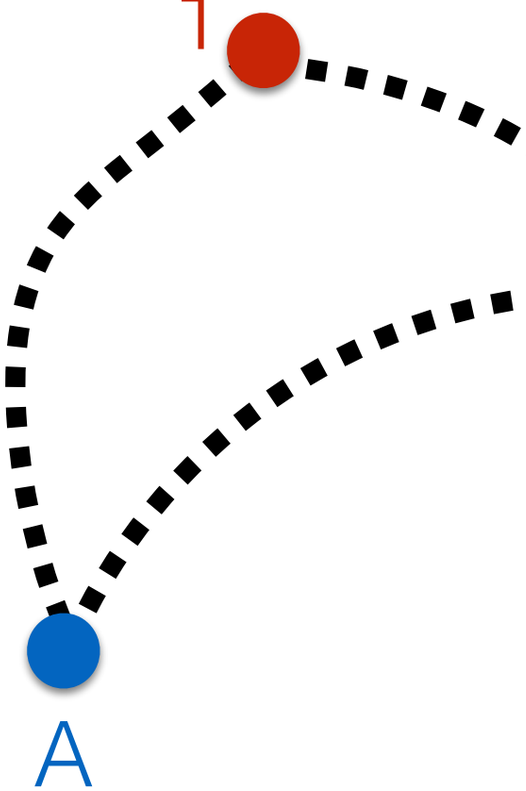


60

90

150

120 90 90



1

2

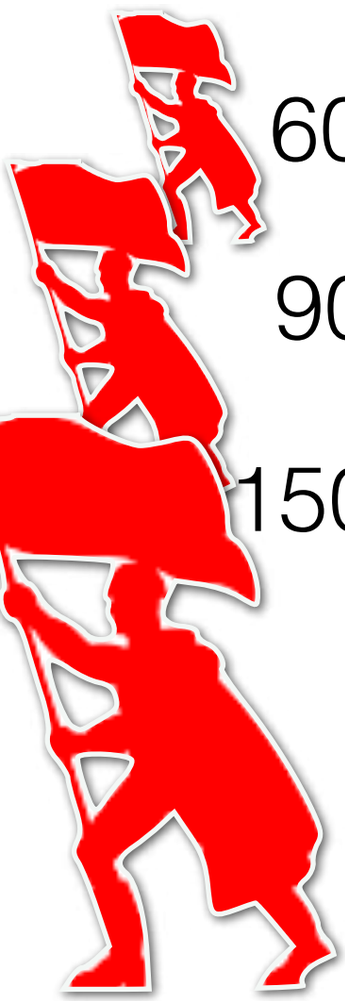
3

A

B

C

Kantorovich Problem

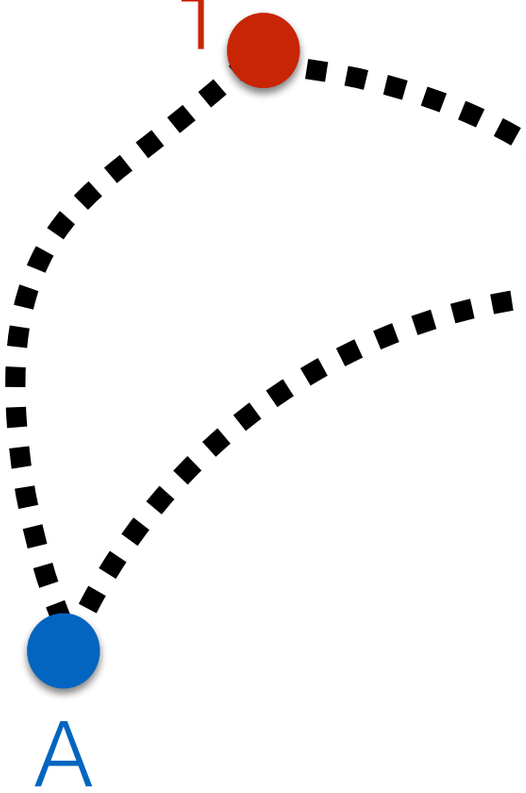


60

90

150

120 90 90



1

2

3

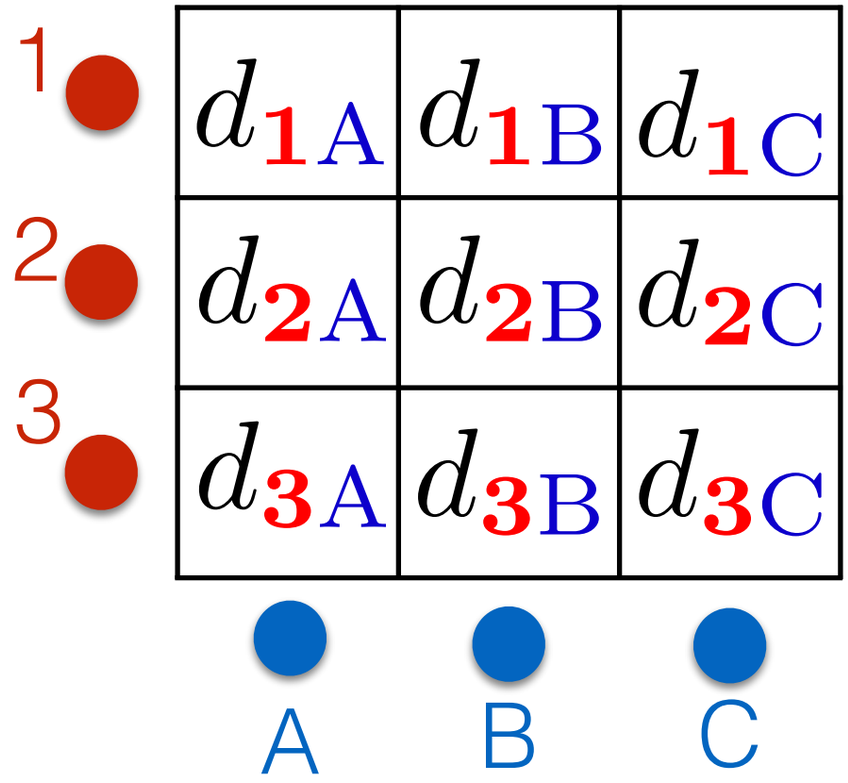
A

B

C

Kantorovich Problem

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



Kantorovich Problem

Transportation matrix

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

Distance matrix

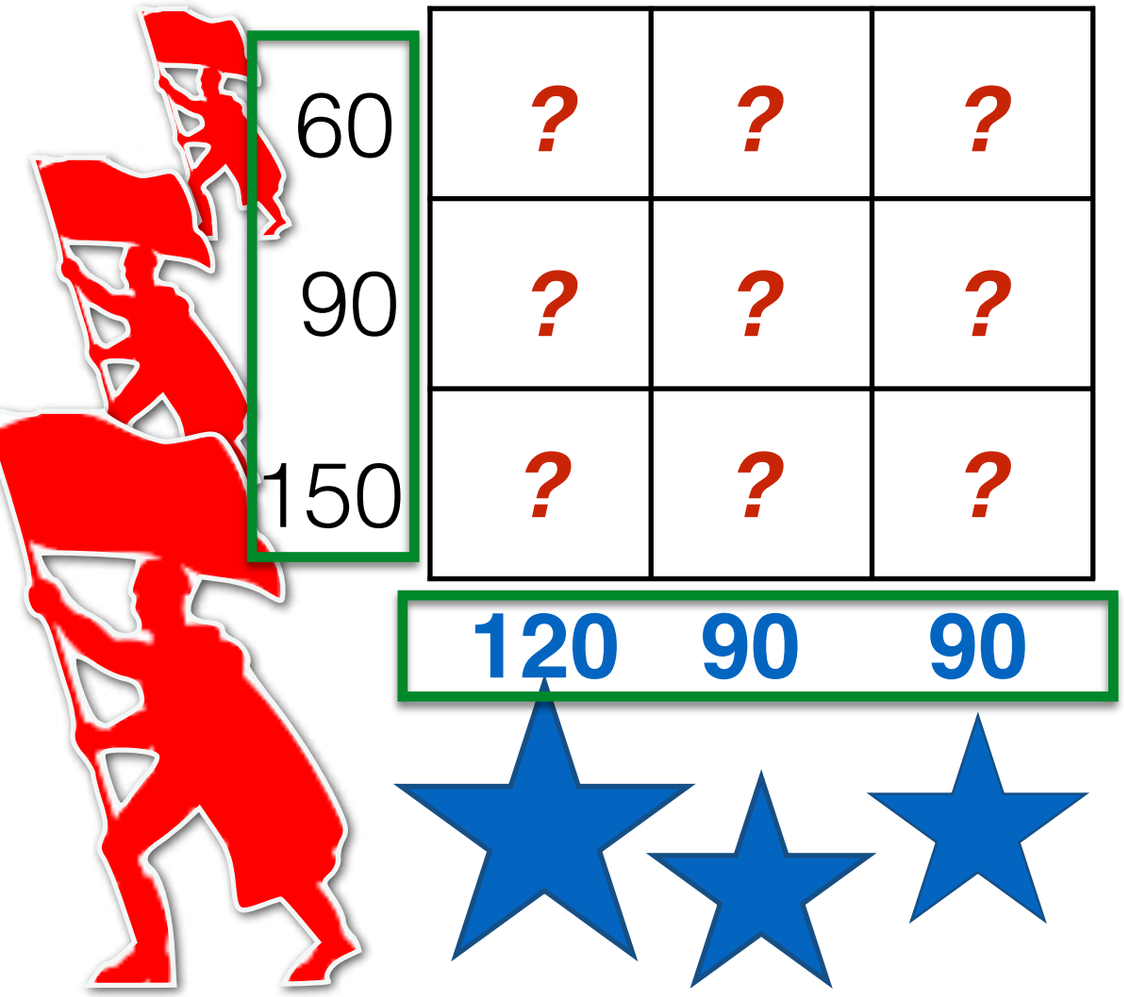
1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Kantorovich Problem

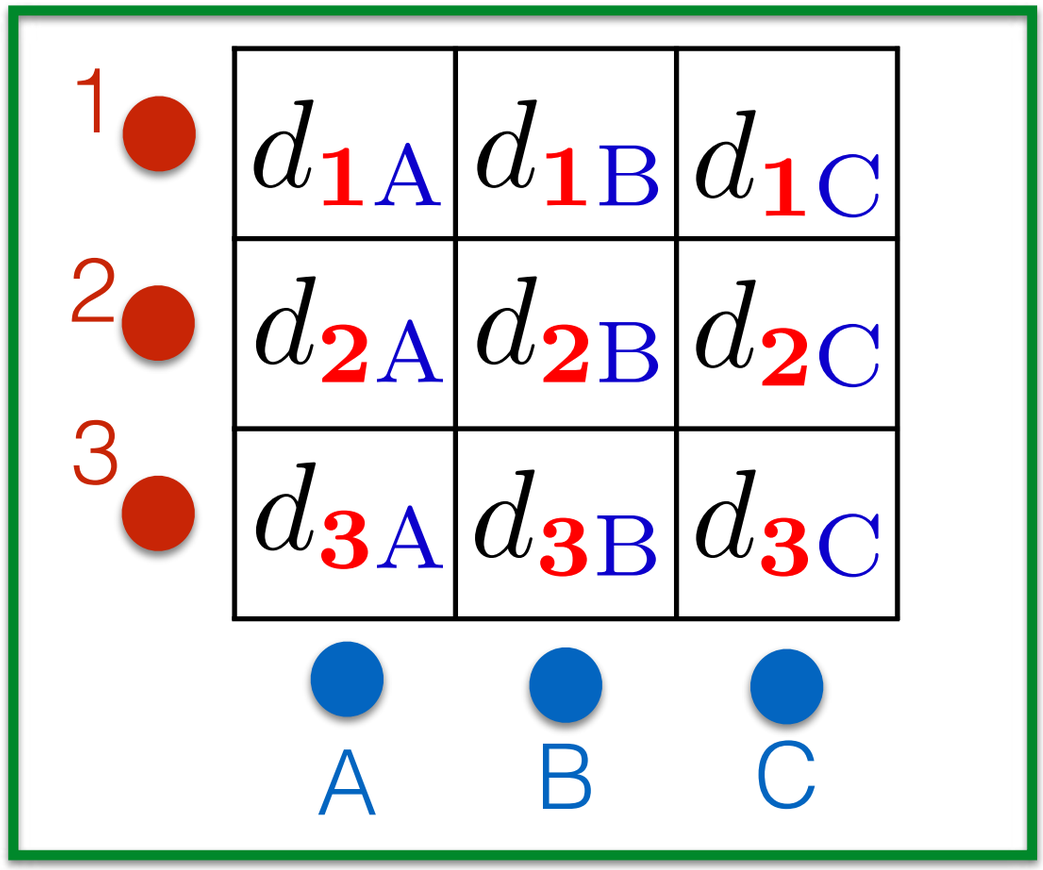
The problem is entirely described by **counts** *and a* **cost/distance matrix**

Transportation matrix

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



Distance matrix



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	$p_{1\mathrm{A}}$	$p_{1\mathrm{B}}$	$p_{1\mathrm{C}}$
a_2	$p_{2\mathrm{A}}$	$p_{2\mathrm{B}}$	$p_{2\mathrm{C}}$
a_3	$p_{3\mathrm{A}}$	$p_{3\mathrm{B}}$	$p_{3\mathrm{C}}$
	b_{A}	b_{B}	b_{C}

Distance matrix

1	$d_{1\mathrm{A}}$	$d_{1\mathrm{B}}$	$d_{1\mathrm{C}}$
2	$d_{2\mathrm{A}}$	$d_{2\mathrm{B}}$	$d_{2\mathrm{C}}$
3	$d_{3\mathrm{A}}$	$d_{3\mathrm{B}}$	$d_{3\mathrm{C}}$
	A	B	C

Kantorovich Problem

Transportation matrix

a_1	$p_{1\mathrm{A}}$	$p_{1\mathrm{B}}$	$p_{1\mathrm{C}}$
a_2	$p_{2\mathrm{A}}$	$p_{2\mathrm{B}}$	$p_{2\mathrm{C}}$
a_3	$p_{3\mathrm{A}}$	$p_{3\mathrm{B}}$	$p_{3\mathrm{C}}$
	b_{A}	b_{B}	b_{C}

Distance matrix

1	$d_{1\mathrm{A}}$	$d_{1\mathrm{B}}$	$d_{1\mathrm{C}}$
2	$d_{2\mathrm{A}}$	$d_{2\mathrm{B}}$	$d_{2\mathrm{C}}$
3	$d_{3\mathrm{A}}$	$d_{3\mathrm{B}}$	$d_{3\mathrm{C}}$
	A	B	C

Constraints

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

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

$$p_{ij} \geq 0$$

Kantorovich Problem

Transportation matrix

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

Distance matrix

1	$d_{1\mathrm{A}}$	$d_{1\mathrm{B}}$	$d_{1\mathrm{C}}$
2	$d_{2\mathrm{A}}$	$d_{2\mathrm{B}}$	$d_{2\mathrm{C}}$
3	$d_{3\mathrm{A}}$	$d_{3\mathrm{B}}$	$d_{3\mathrm{C}}$
	A	B	C

Constraints

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

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

$$p_{ij} \geq 0$$

Cost function

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

Kantorovich Problem

Transportation matrix

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

Distance matrix

1	$d_{1\mathrm{A}}$	$d_{1\mathrm{B}}$	$d_{1\mathrm{C}}$
2	$d_{2\mathrm{A}}$	$d_{2\mathrm{B}}$	$d_{2\mathrm{C}}$
3	$d_{3\mathrm{A}}$	$d_{3\mathrm{B}}$	$d_{3\mathrm{C}}$
	A	B	C

Constraints

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

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

$$p_{ij} \geq 0$$

Cost function

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

Problem

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

Kantorovich Problem

60

90

120

90

150

90

Kantorovich Problem

30 30

90

120

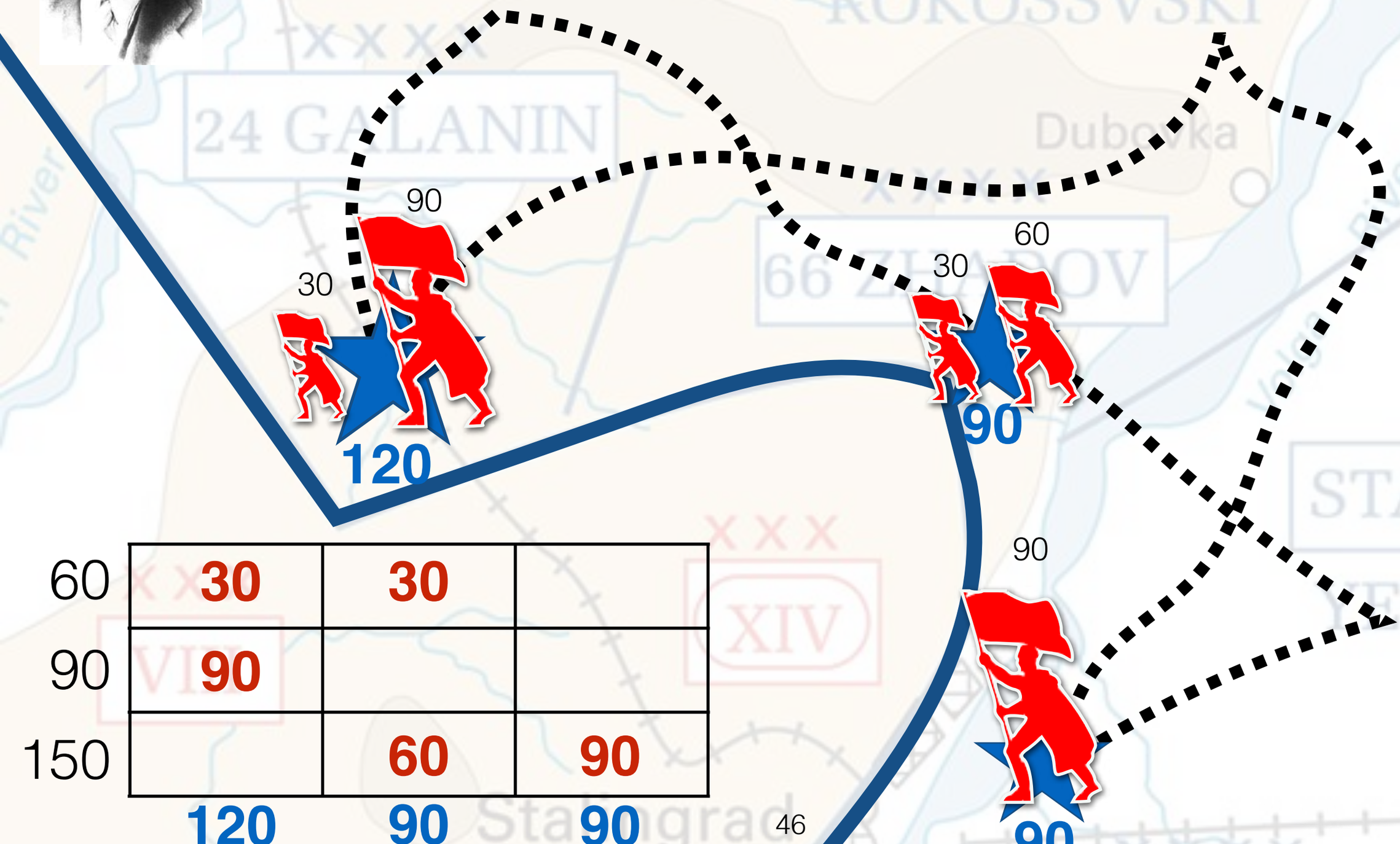
90

90

60 90

60	30	30	
90	90		
150		60	90
	120	90	90

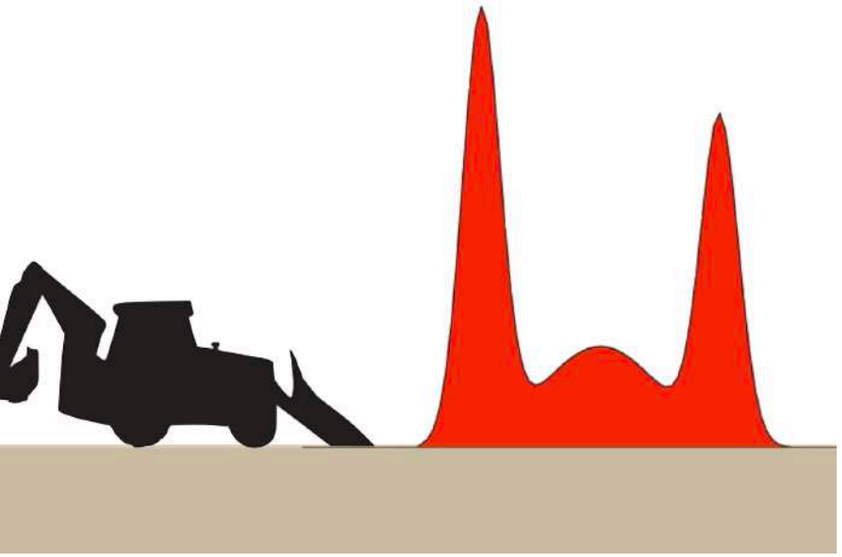
Kantorovich Problem



60	30	30	
90	90		
150		60	90
	120	90	90

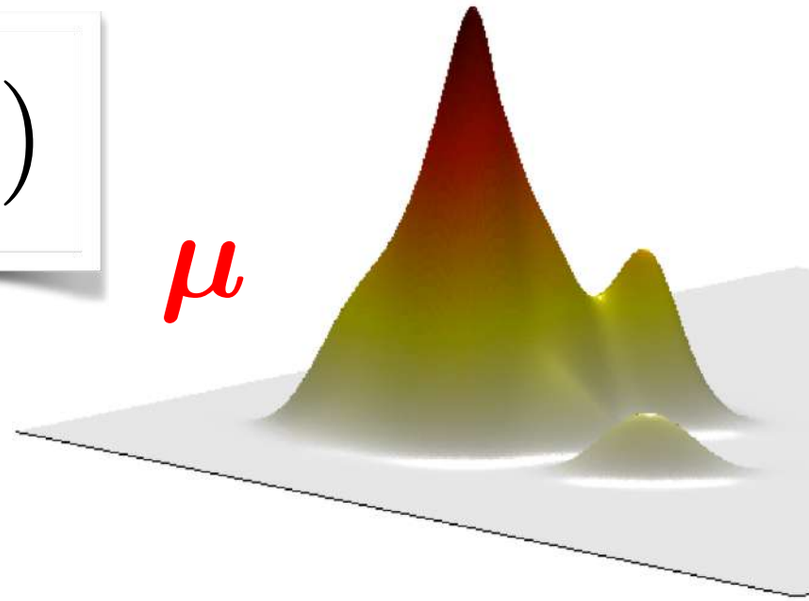
Mathematical Formalism

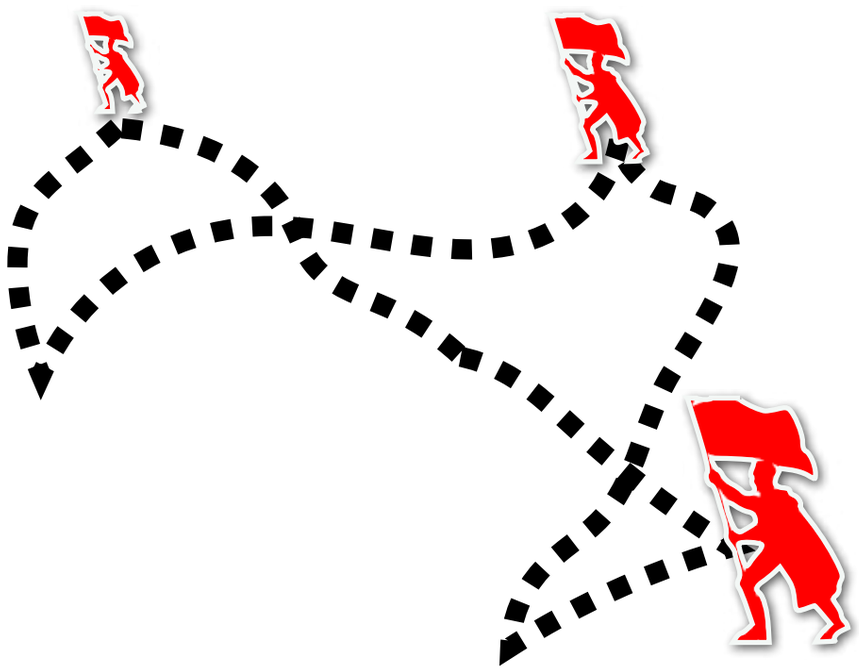
These problems involve discrete and continuous probability measures on a geometric space Ω



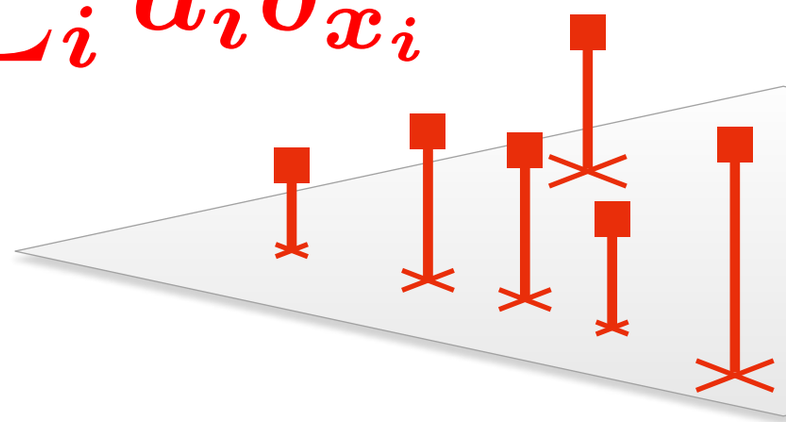
$\mathcal{P}(\Omega)$

μ

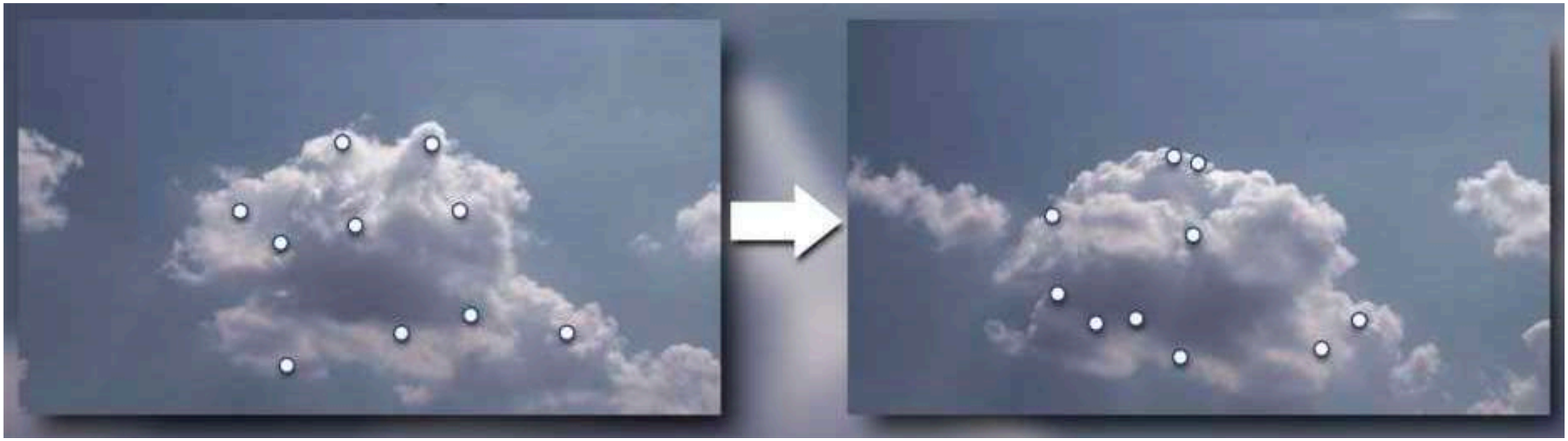




$\sum_i a_i \delta_{x_i}$



OT: Nature's way to move particles



screenshot video: ETH Zurich / Michael Steiner, vistory GmbH

Figalli

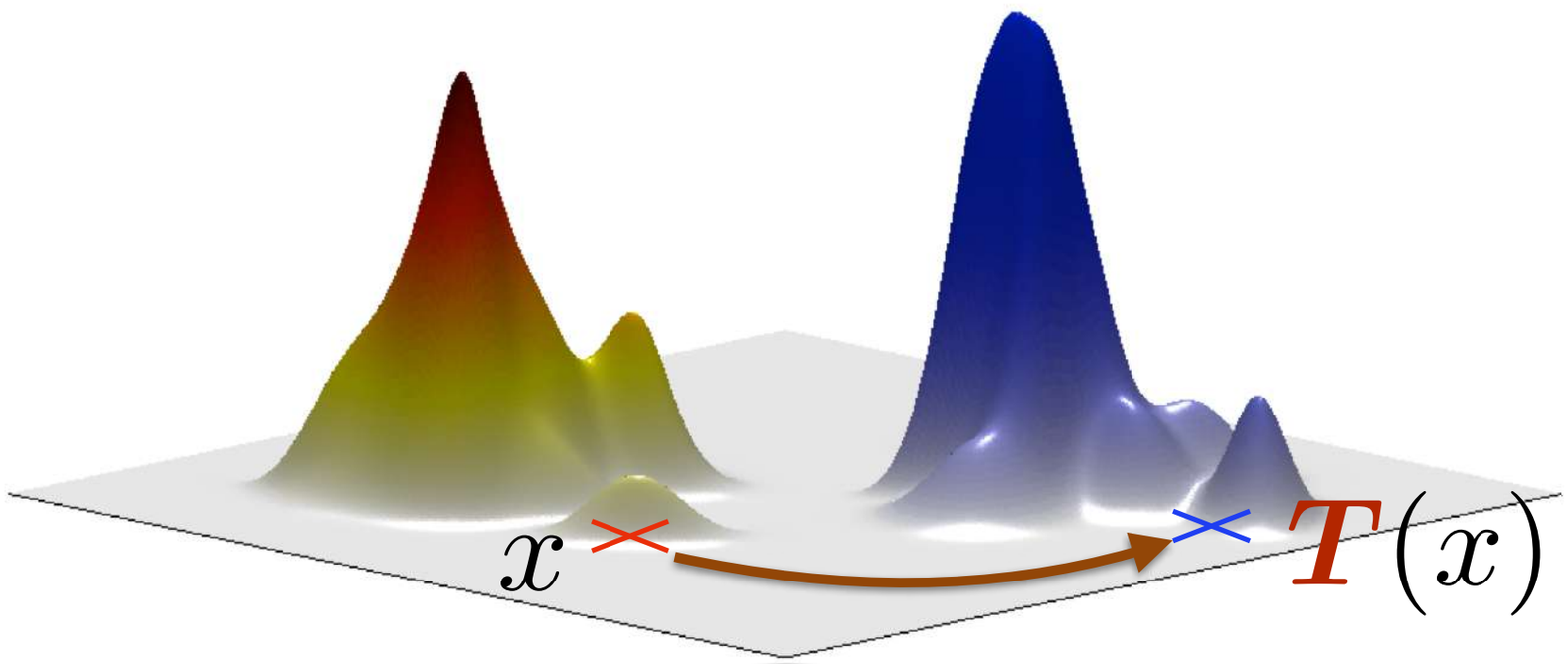
Fields'18

Monge Problem

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

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

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

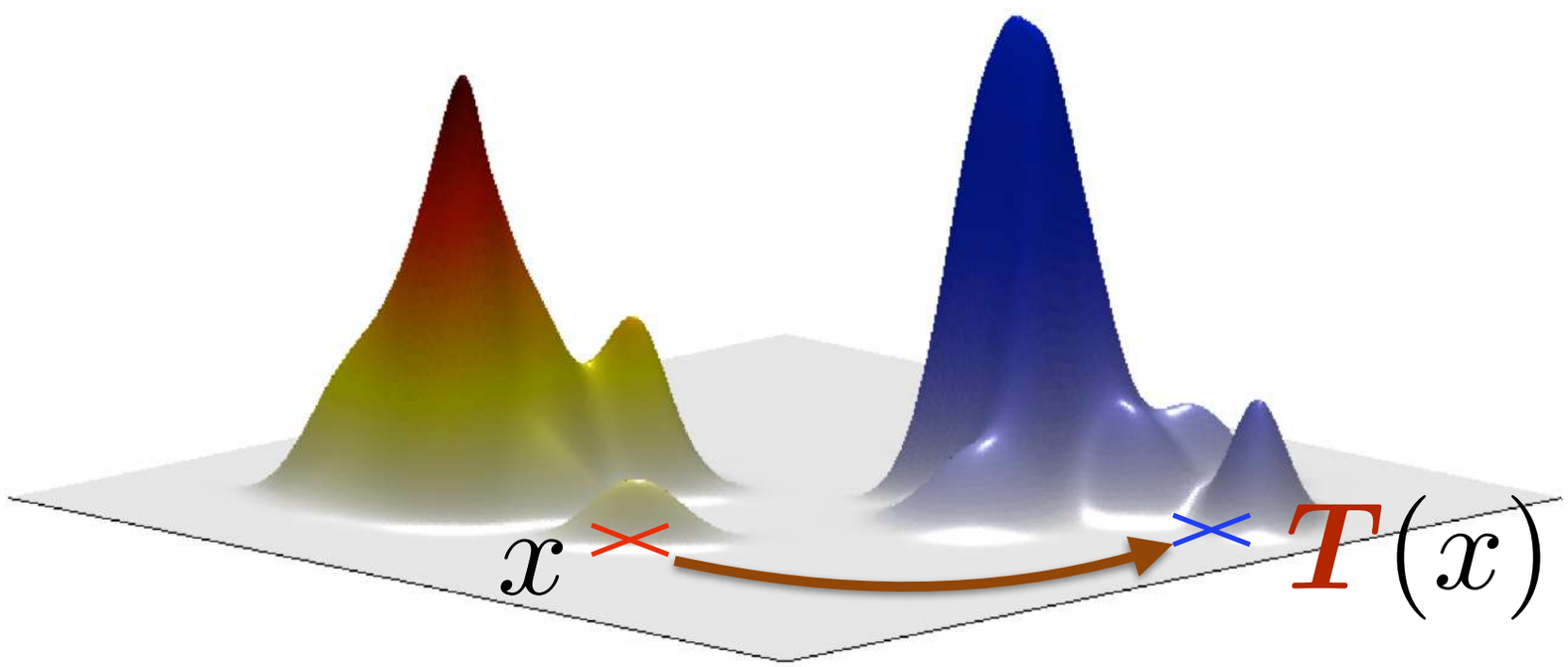


Monge Problem

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

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

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



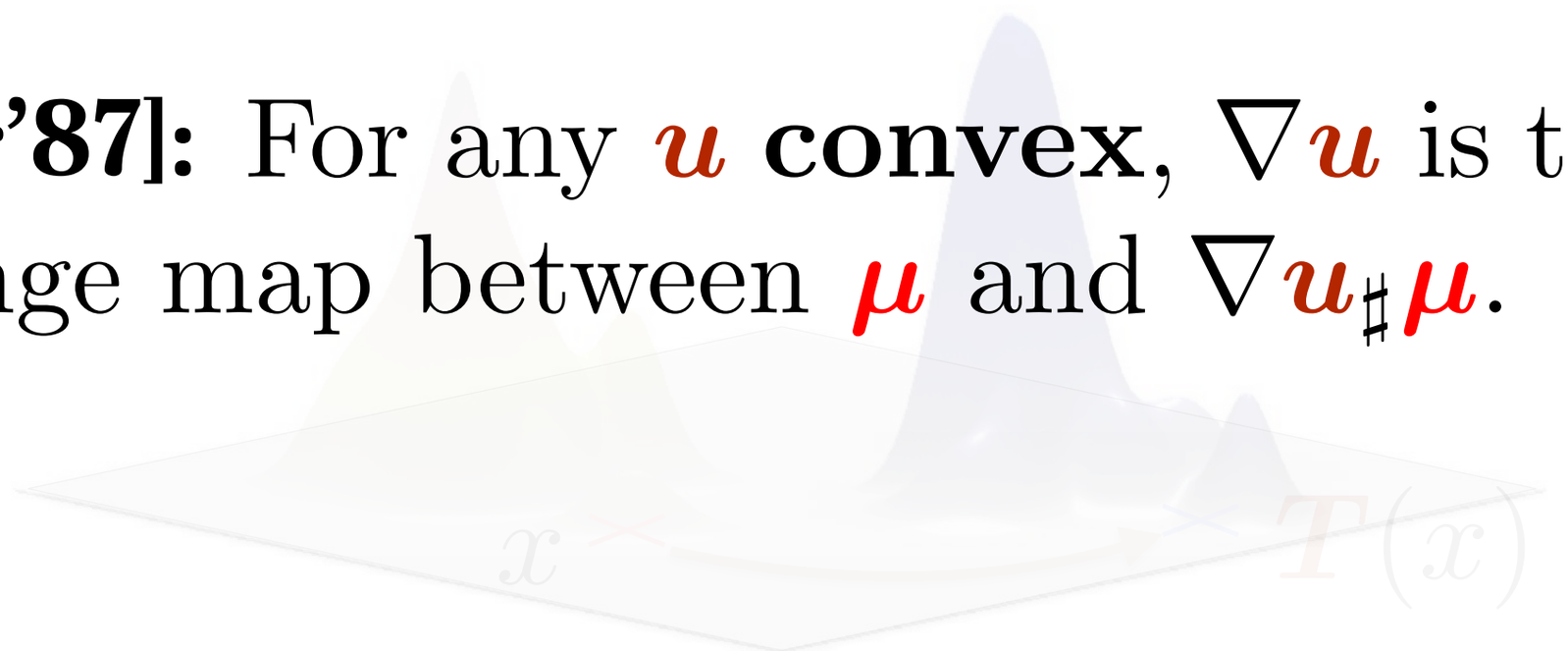
Monge Problem

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

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

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

[Brenier'87]: For any $\boldsymbol{u}$ **convex**, $\nabla \boldsymbol{u}$ is the OT Monge map between $\boldsymbol{\mu}$ and $\nabla \boldsymbol{u}_\sharp \boldsymbol{\mu}$.



Links: Monge-Ampère Equation

If $\Omega = \mathbb{R}^d$, $\boldsymbol{c} = \|\cdot - \cdot\|^2$, $\boldsymbol{\mu}, \boldsymbol{\nu}$ have densities $\boldsymbol{p}, \boldsymbol{q}$, then $\boldsymbol{T}_\sharp \boldsymbol{\mu} = \boldsymbol{\nu}$ is equivalent to

$$\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{q}(\boldsymbol{T}(\boldsymbol{x}))|\det J_{\boldsymbol{T}}(\boldsymbol{x})|$$

Monge-Ampère: find convex $\boldsymbol{f}$ such that

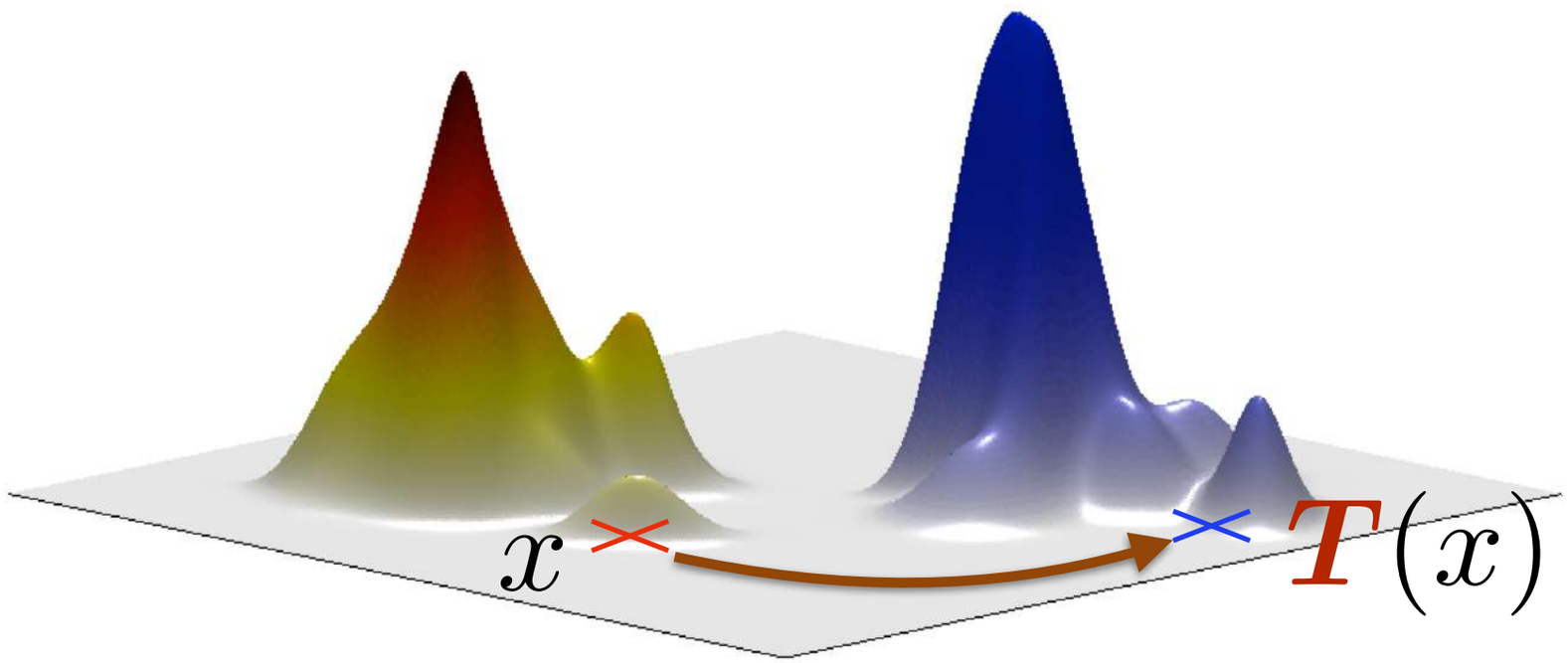
$$|\nabla^2 \boldsymbol{f}(\boldsymbol{x})| = \frac{\boldsymbol{p}(\boldsymbol{x})}{\boldsymbol{q}(\nabla \boldsymbol{f}(\boldsymbol{x}))}$$

Monge Problem

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

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

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

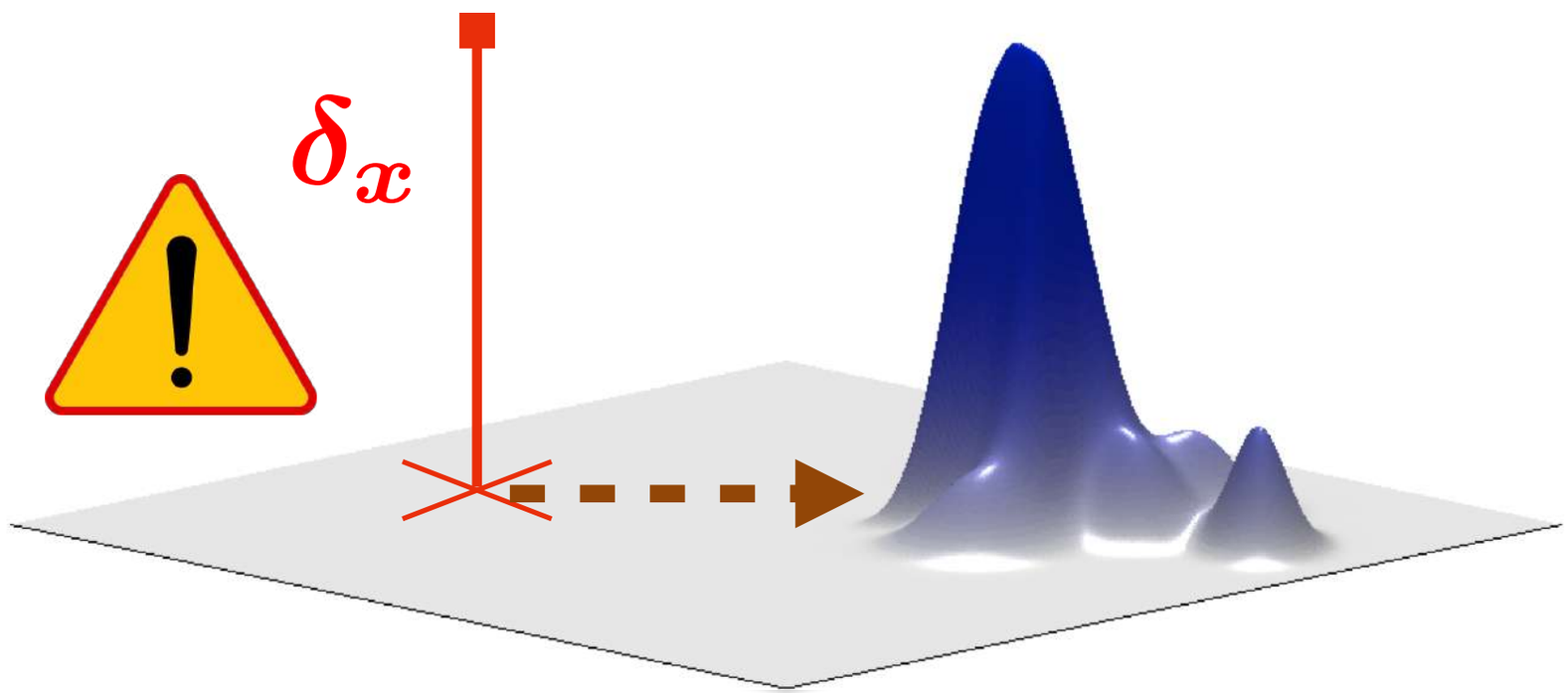


Monge Problem

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

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

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

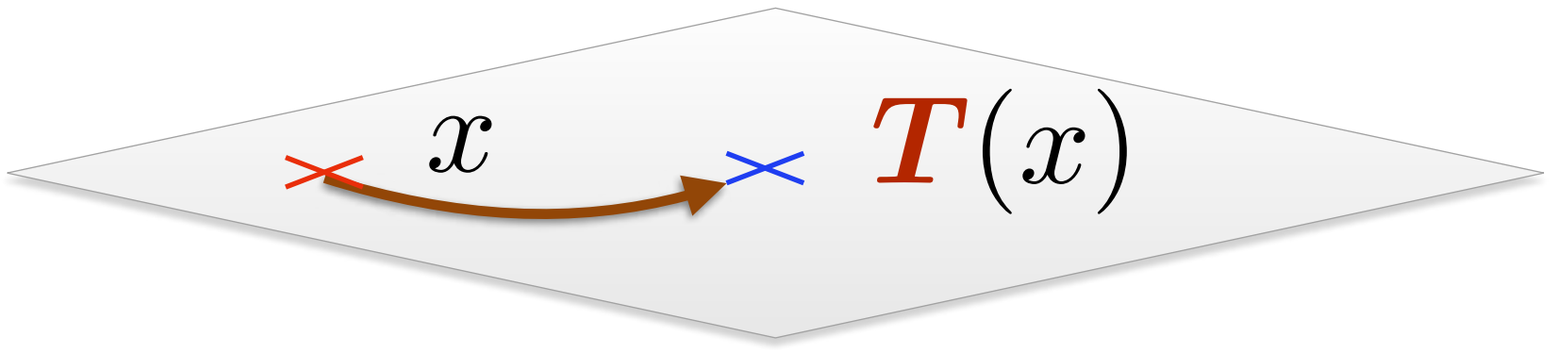


Kantorovich Relaxation

Instead of maps $\boldsymbol{T} : \Omega \to \Omega$,

consider probabilistic maps,

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

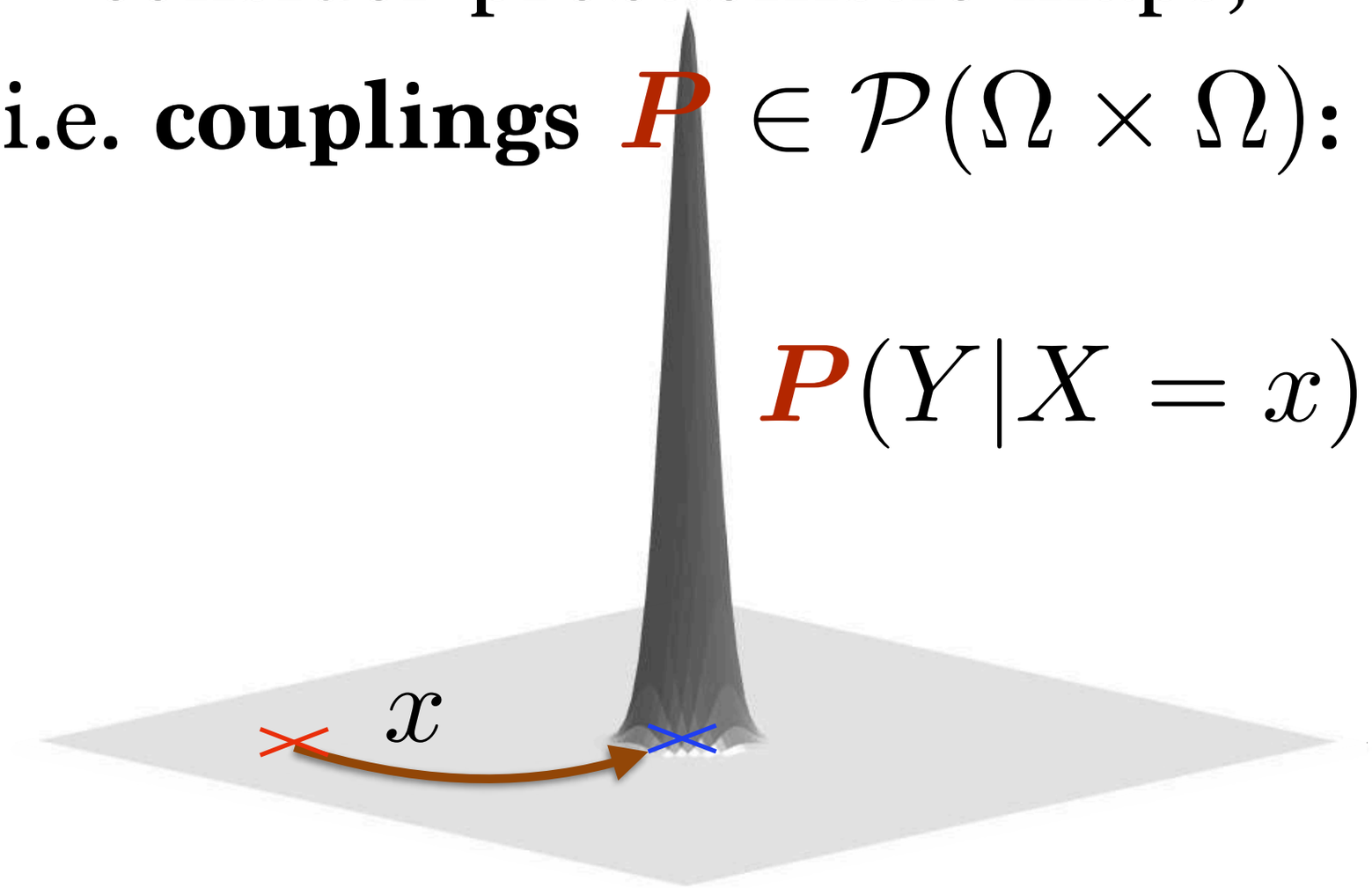


Kantorovich Relaxation

Instead of maps $\boldsymbol{T} : \Omega \to \Omega$,

consider probabilistic maps,

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



Kantorovich Relaxation

Instead of maps $\boldsymbol{T} : \Omega \to \Omega$,

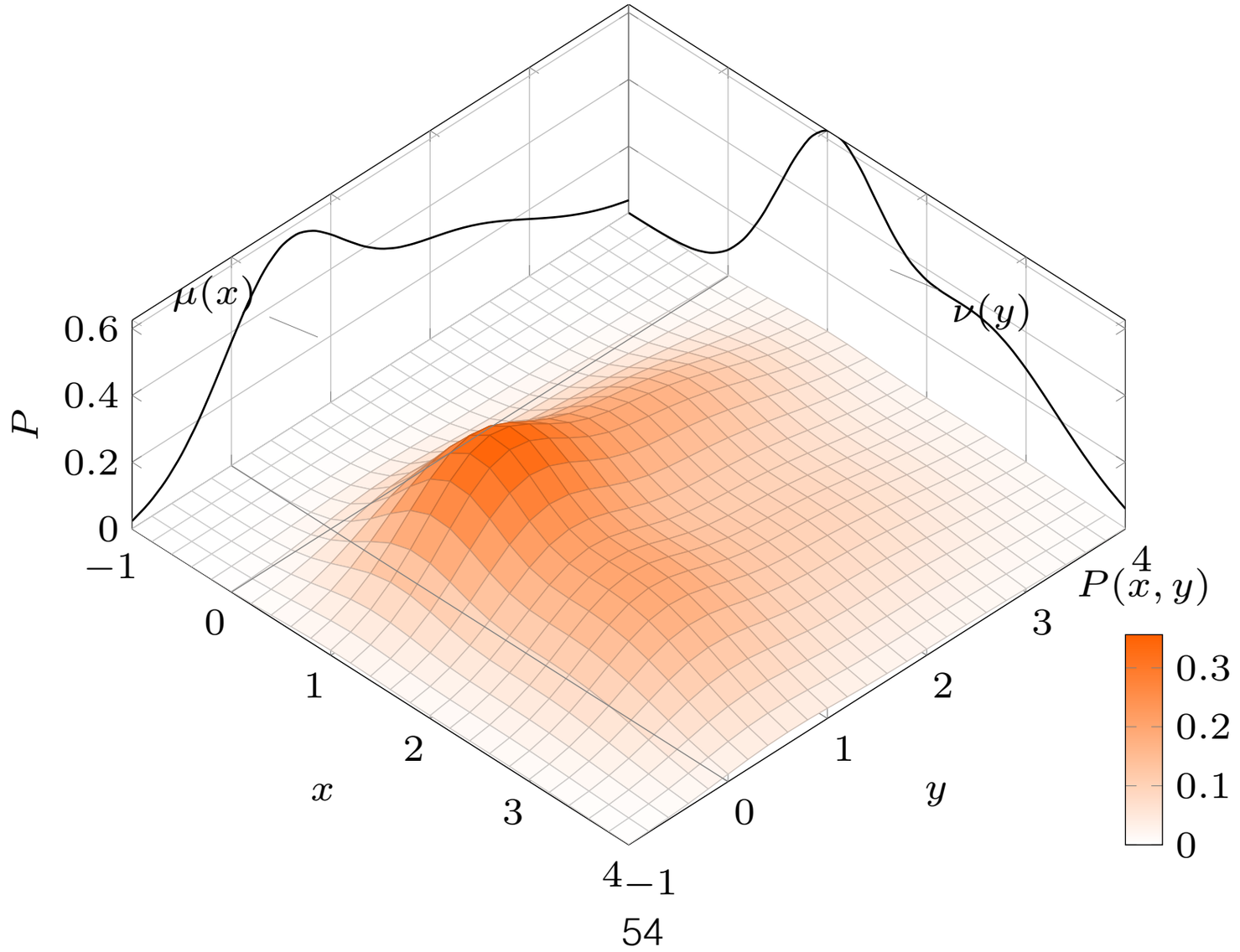
consider probabilistic maps,

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

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

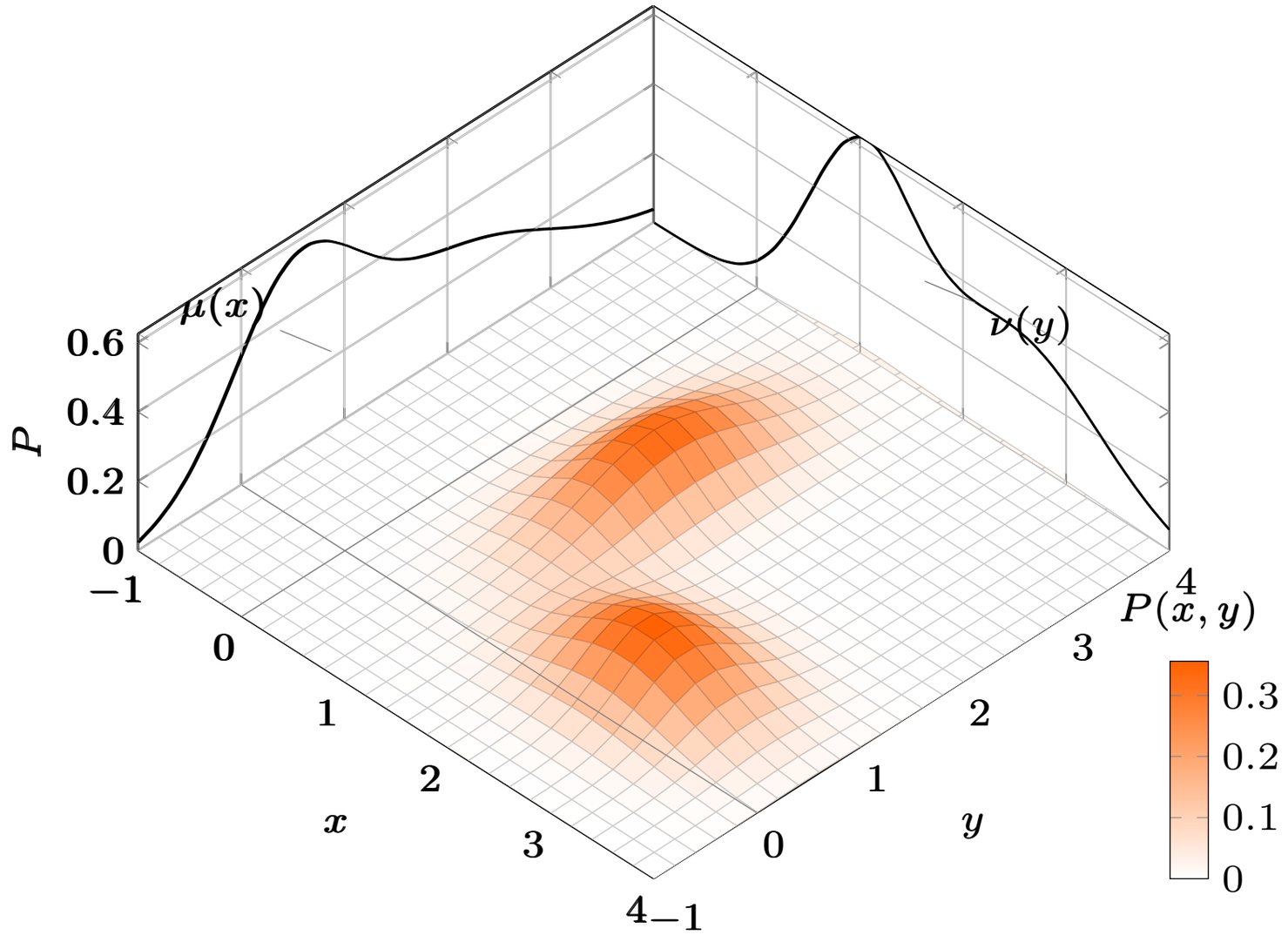
Kantorovich Relaxation

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



Kantorovich Relaxation

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



Kantorovich Problem

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

MONGE

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

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

PRIMAL

Kantorovich Problem

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

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

PRIMAL

Kantorovich Problem

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

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

PRIMAL

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

DUAL

Kantorovich Problem

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

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

PRIMAL

For two real-valued functions $\boldsymbol{\varphi}, \boldsymbol{\psi}$ on Ω,

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

Kantorovich Problem

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

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

PRIMAL

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

DUAL

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = ?$$

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = ?$$

$$\iint \boldsymbol{\varphi}(\boldsymbol{x}) d\boldsymbol{P}(\boldsymbol{x}, y) + \iint \boldsymbol{\psi}(\boldsymbol{y}) d\boldsymbol{P}(x, \boldsymbol{y})$$

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = ?$$

$$\iint \boldsymbol{\varphi}(\boldsymbol{x}) d\boldsymbol{P}(\boldsymbol{x}, y) + \iint \boldsymbol{\psi}(\boldsymbol{y}) d\boldsymbol{P}(x, \boldsymbol{y})$$

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = \mathbf{0}$$

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = \mathbf{0}$$

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

Deriving Kantorovich Duality

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} = \mathbf{0}$$

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

$$\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} =$$

$$\int \boldsymbol{\varphi} d \underbrace{(\boldsymbol{\mu} - \boldsymbol{P_X})}_{\neq 0} + \int \boldsymbol{\psi} d \underbrace{(\boldsymbol{\nu} - \boldsymbol{P_Y})}_{\neq 0}$$

and / or

Deriving Kantorovich Duality

$$\iota_\Pi(\boldsymbol{P}) = \sup_{\boldsymbol{\varphi},\boldsymbol{\psi}} \left[\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} \right]$$

$$= \begin{cases} 0 & \text{if } \boldsymbol{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu}), \\ +\infty & \text{otherwise.} \end{cases}$$

Deriving Kantorovich Duality

$$\iota_\Pi(\boldsymbol{P}) = \sup_{\boldsymbol{\varphi},\boldsymbol{\psi}} \left[\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} \right]$$

$$= \begin{cases} 0 & \text{if } \boldsymbol{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu}), \\ +\infty & \text{otherwise.} \end{cases}$$

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

Deriving Kantorovich Duality

$$\iota_\Pi(\boldsymbol{P}) = \sup_{\boldsymbol{\varphi},\boldsymbol{\psi}} \left[\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} \right]$$

$$= \begin{cases} 0 & \text{if } \boldsymbol{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu}), \\ +\infty & \text{otherwise.} \end{cases}$$

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

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

Deriving Kantorovich Duality

$$\iota_\Pi(\boldsymbol{P}) = \sup_{\boldsymbol{\varphi},\boldsymbol{\psi}} \left[\int \boldsymbol{\varphi} d\boldsymbol{\mu} + \int \boldsymbol{\psi} d\boldsymbol{\nu} - \iint \boldsymbol{\varphi} \oplus \boldsymbol{\psi} d\boldsymbol{P} \right]$$

$$= \begin{cases} 0 & \text{if } \boldsymbol{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu}), \\ +\infty & \text{otherwise.} \end{cases}$$

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

Deriving Kantorovich Duality

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

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

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

Deriving Kantorovich Duality

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

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

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

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

DUAL

Wasserstein Distances

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

Def. The p-**Wasserstein distance** between $\boldsymbol{\mu}, \boldsymbol{\nu}$ in $\mathcal{P}(\Omega)$ is

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

Wasserstein Distances

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

Def. The p-**Wasserstein distance** between $\boldsymbol{\mu}, \boldsymbol{\nu}$ in $\mathcal{P}(\Omega)$ is

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

Kantorovich Duality

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

DUAL

- Kantorovich duality is computationally easier: easier to store 2 functions than an entire coupling.
- *D* transforms: useful machinery to consider only **two** instead of **one** dual potential, notably when *p = 1*

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

W1

Links between Monge & Kantorovich

Prop. For "well behaved" costs $\boldsymbol{c}$, if $\boldsymbol{\mu}$ has a density then an *optimal* Monge map $T^\star$ between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ must exist.

Prop. In that case

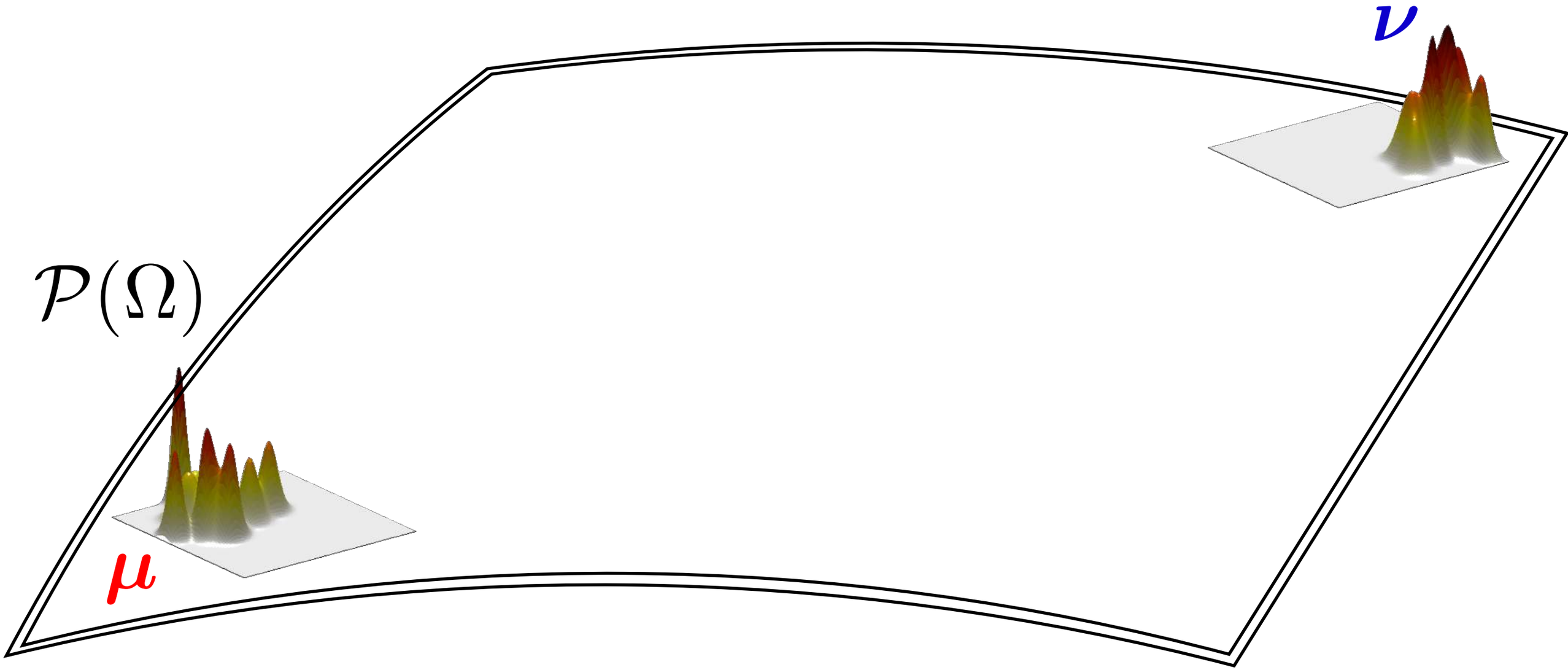
$$\boldsymbol{P}^\star := (\mathrm{Id}, T^\star)_\sharp \boldsymbol{\mu} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$$

is also *optimal* for the Kantorovich problem.

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

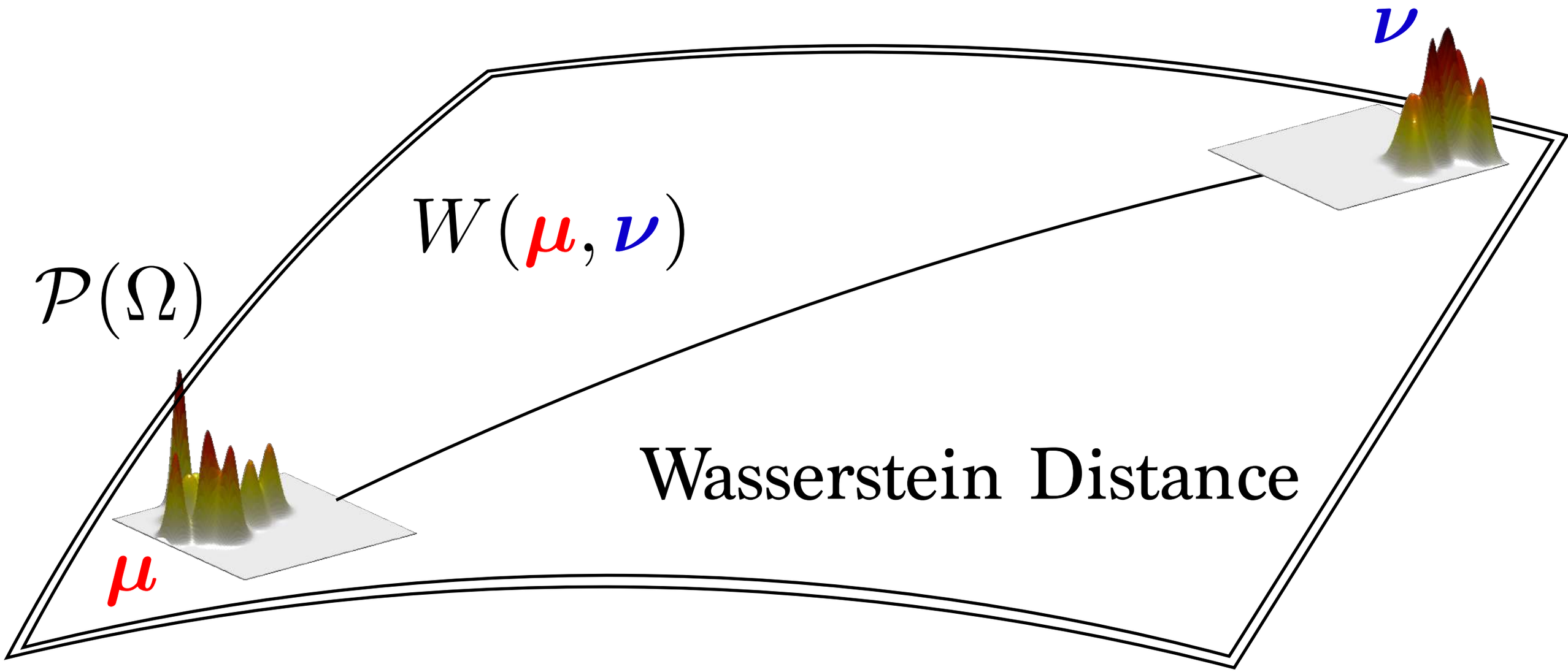
Optimal Transport Geometry

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



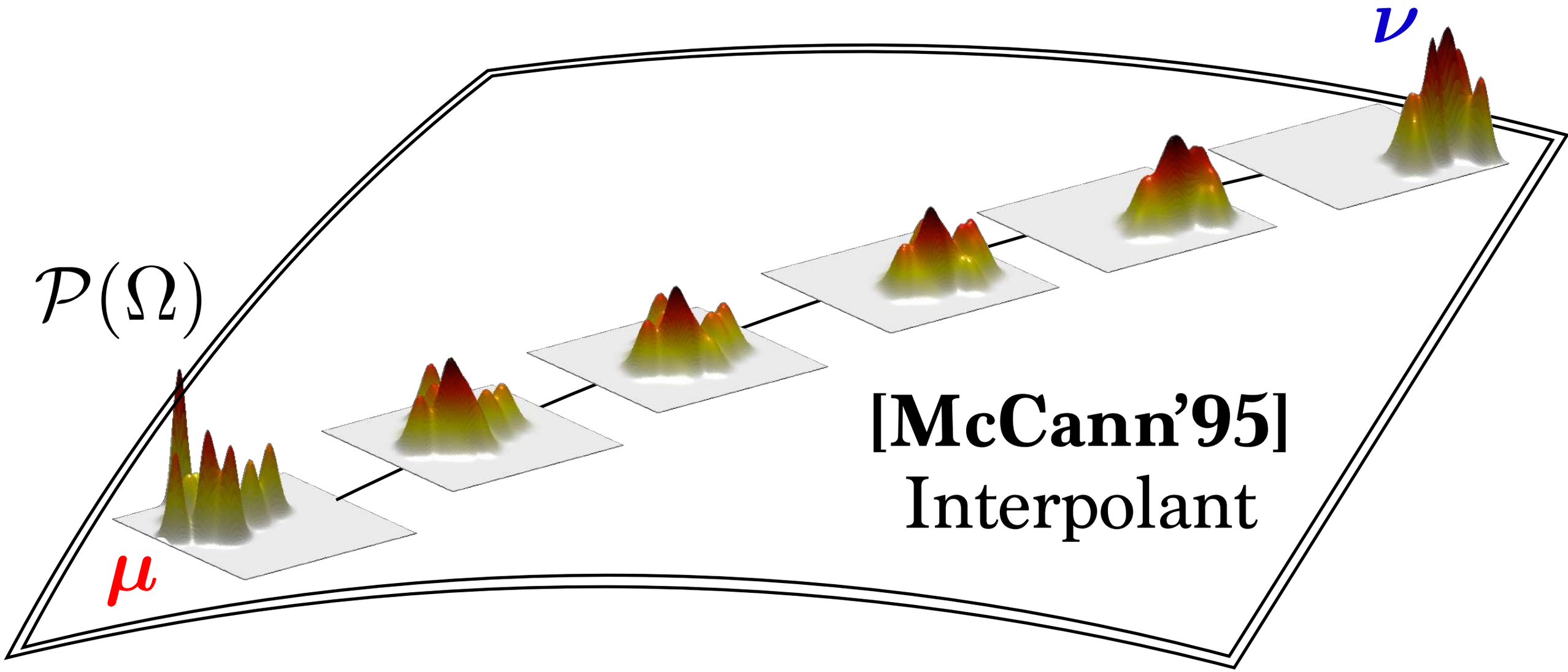
Optimal Transport Geometry

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



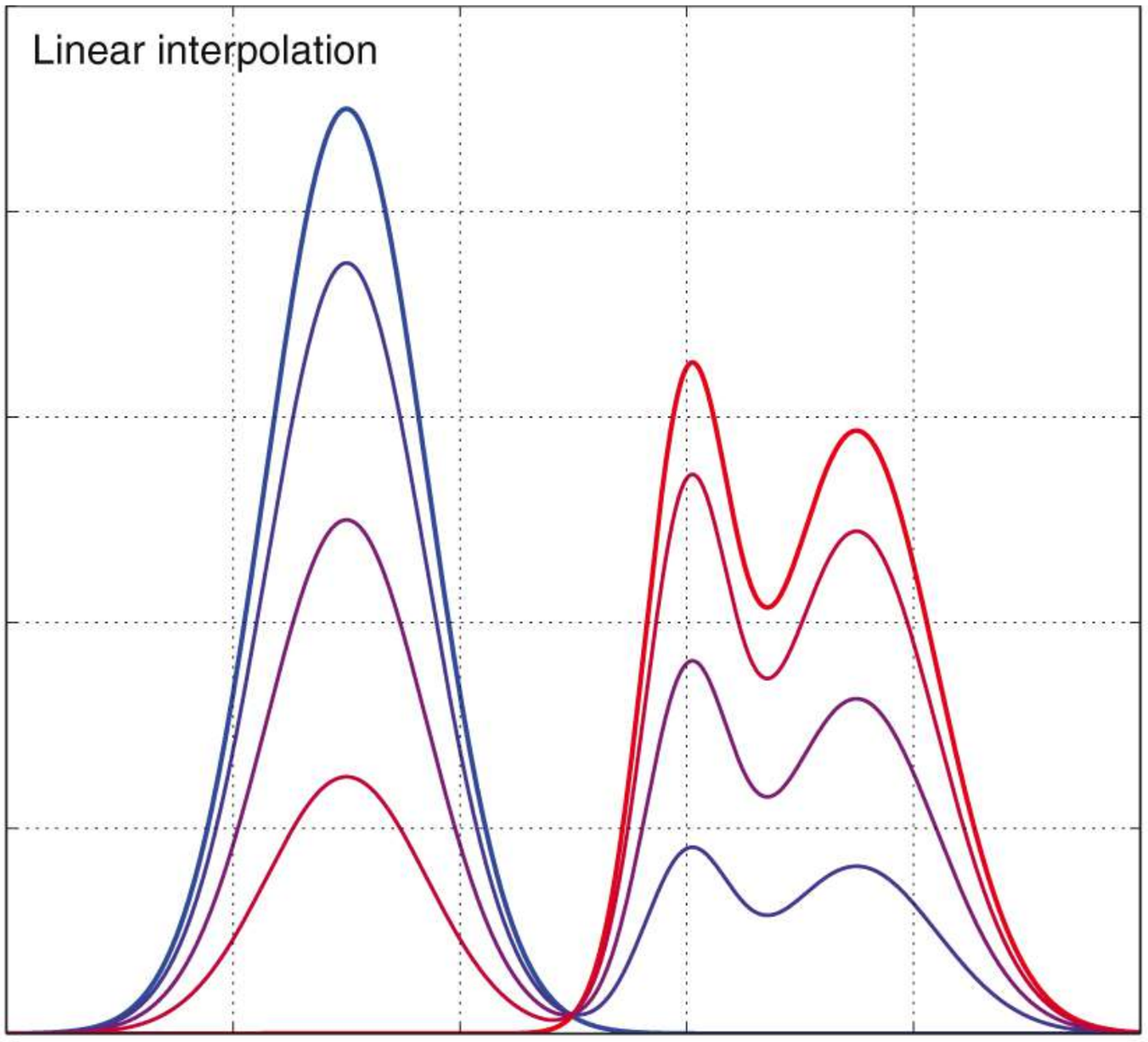
Optimal Transport Geometry

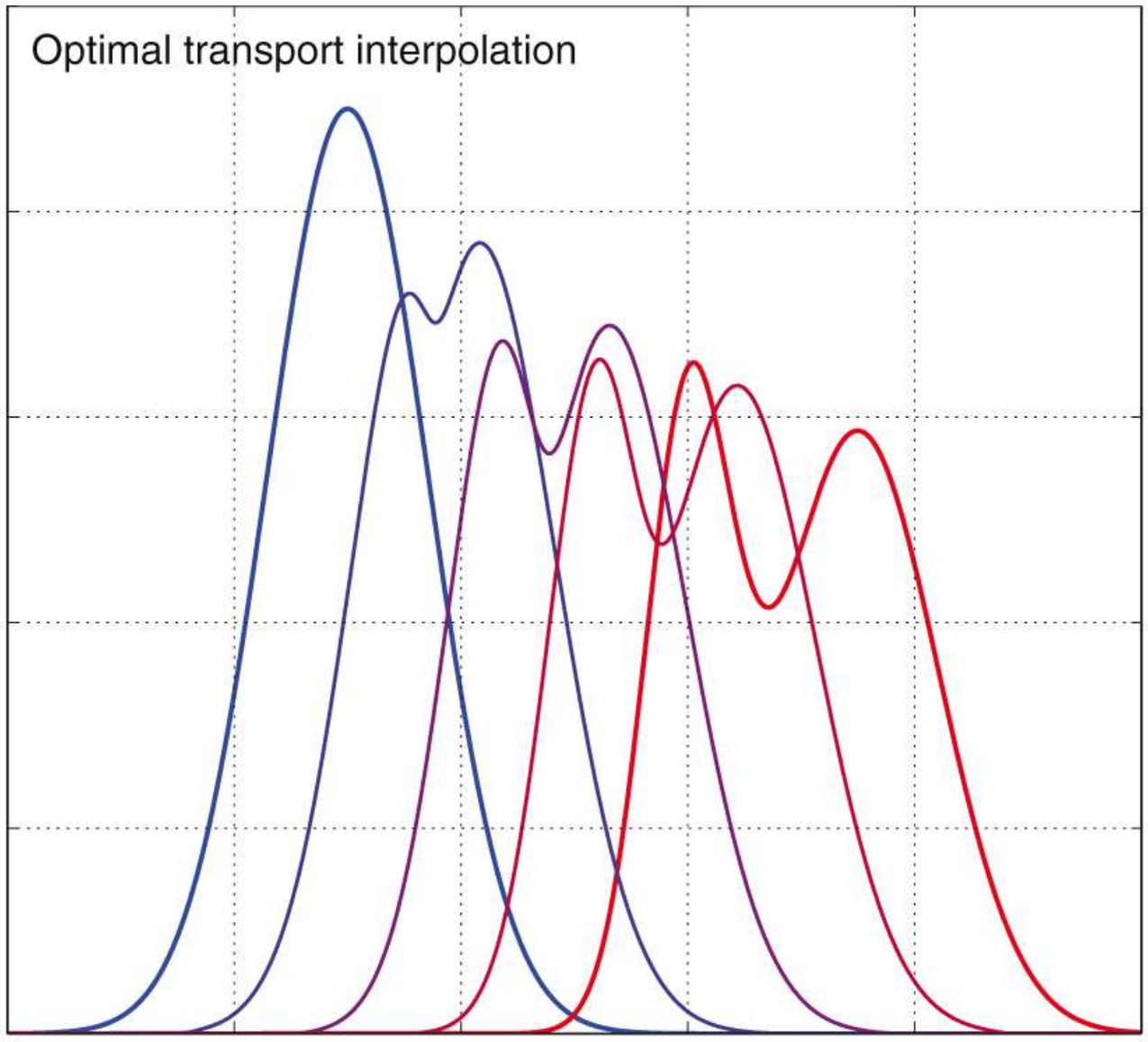
Very different geometry than standard information divergences (*KL*, Euclidean)



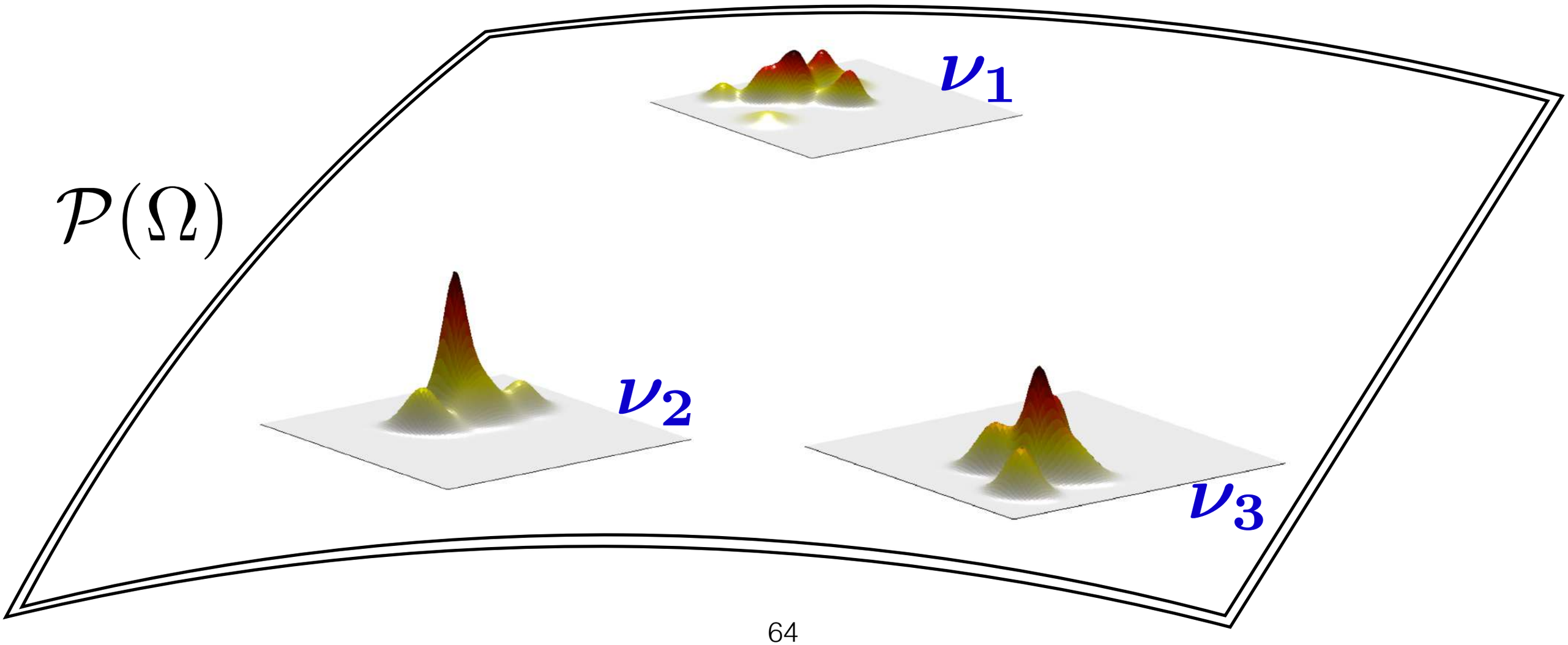
Optimal Transport Geometry

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



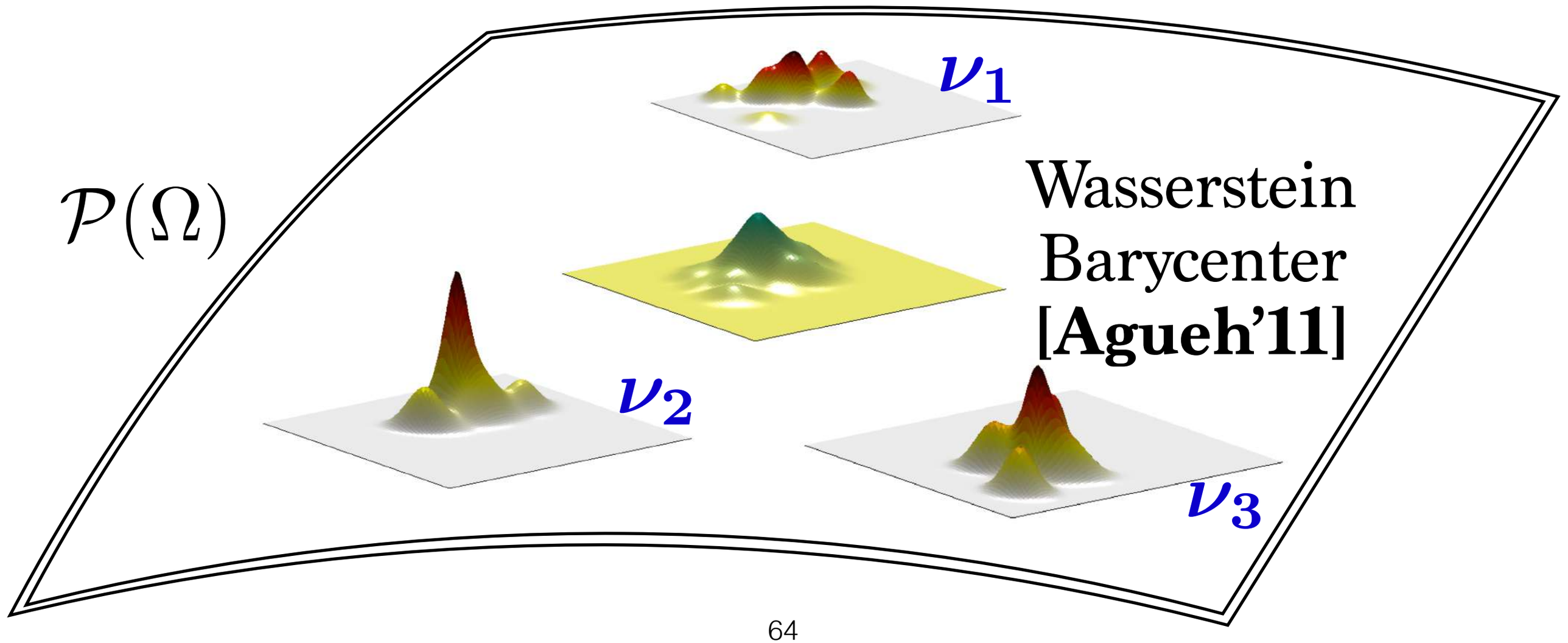


Optimal Transport Geometry



Optimal Transport Geometry

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



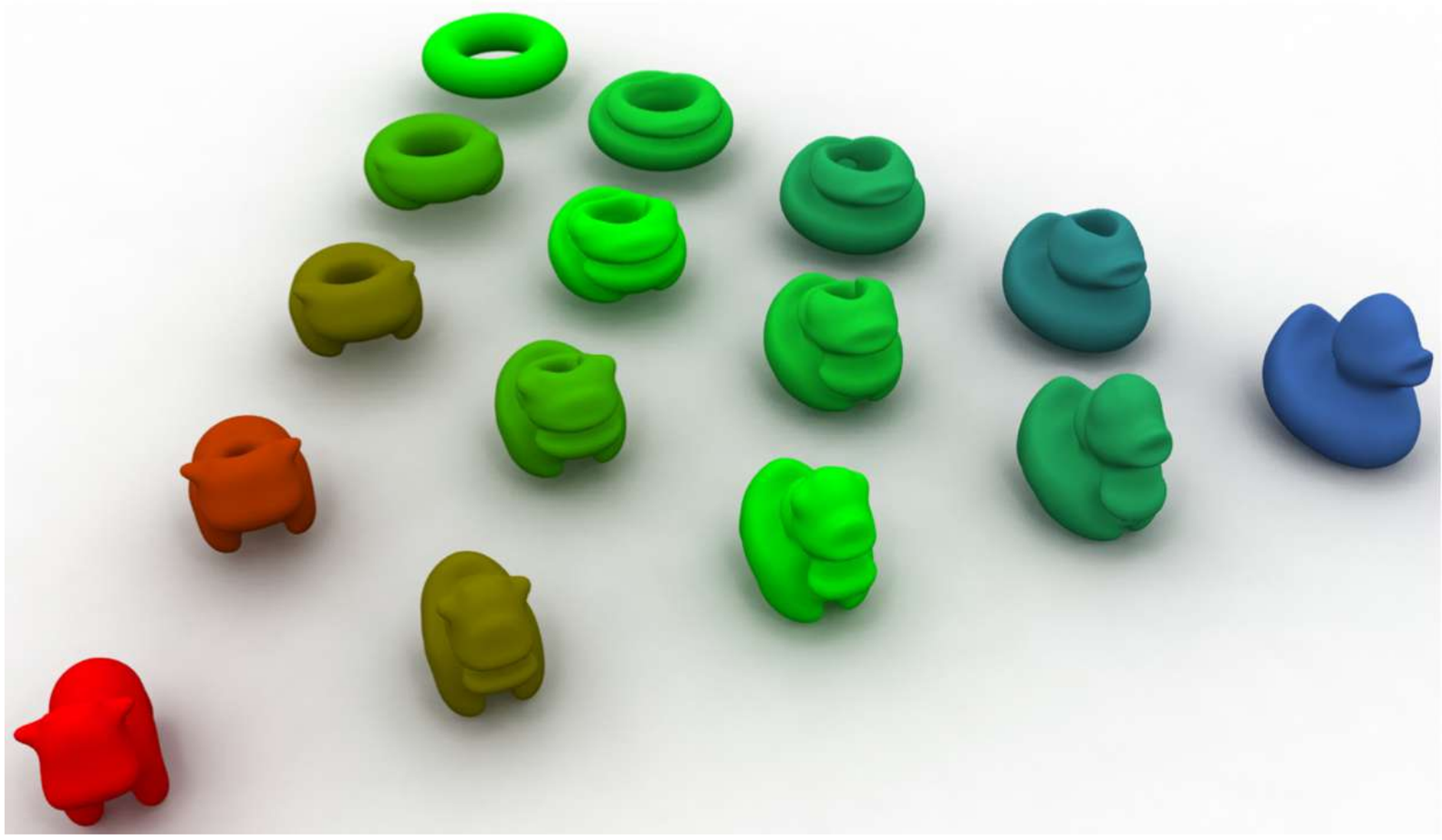
Optimal Transport Geometry

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

[SDPC..'15]

Optimal Transport Geometry

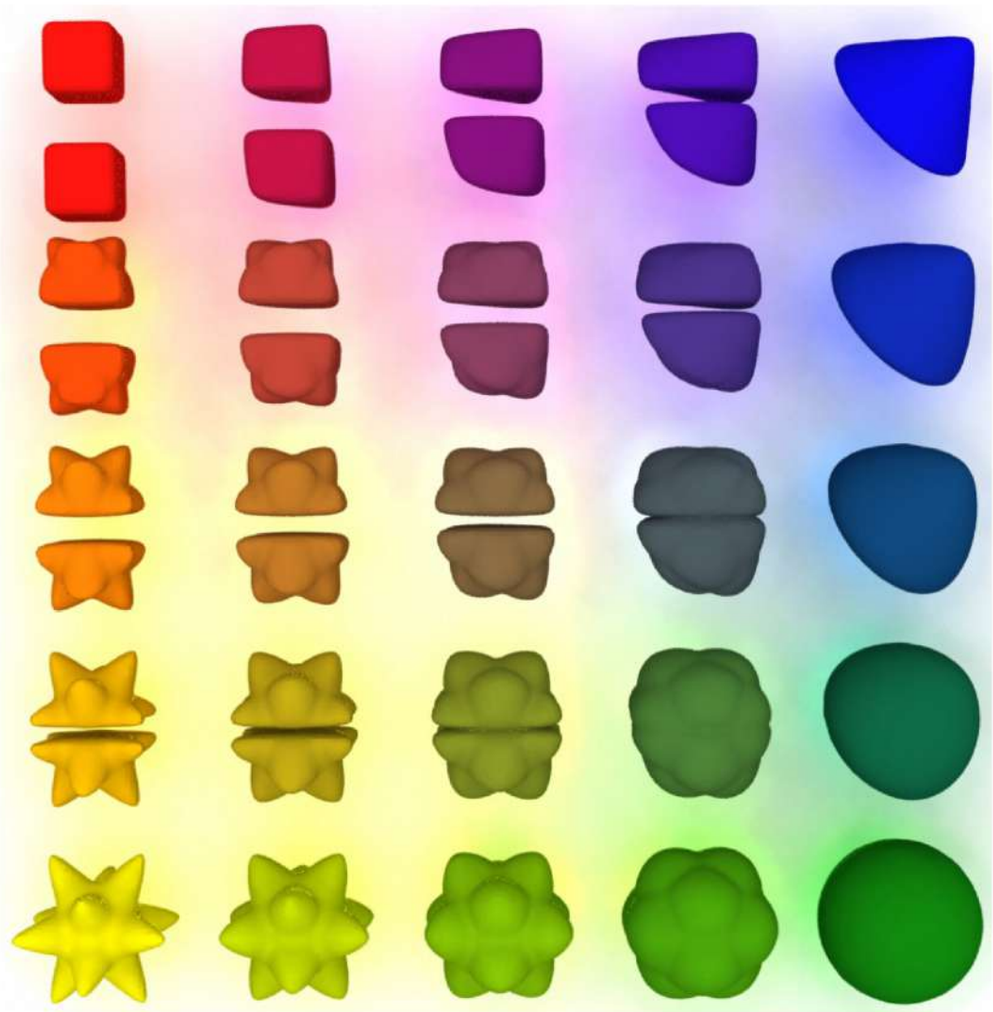
Very different geometry than standard information divergences (*KL*, Euclidean)



[SDPC..'15]

Optimal Transport Geometry

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



[SDPC..'15]

Variational OT Problems in ML

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

$$W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = ?$$

$$W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) \leq \cdots ?$$

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

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

[Jordan Kinderlehrer Otto'98]
[Ambrosio Gigli Savaré'05]

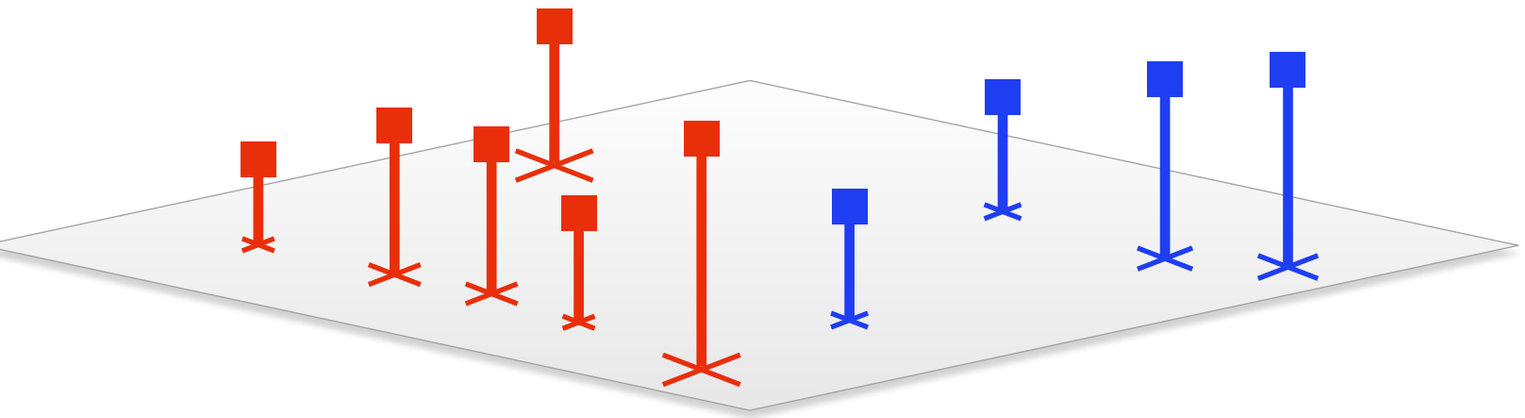
$$\text{“}\nabla_{\boldsymbol{\mu}}\text{”} W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = ?$$

2. Computing OT exactly

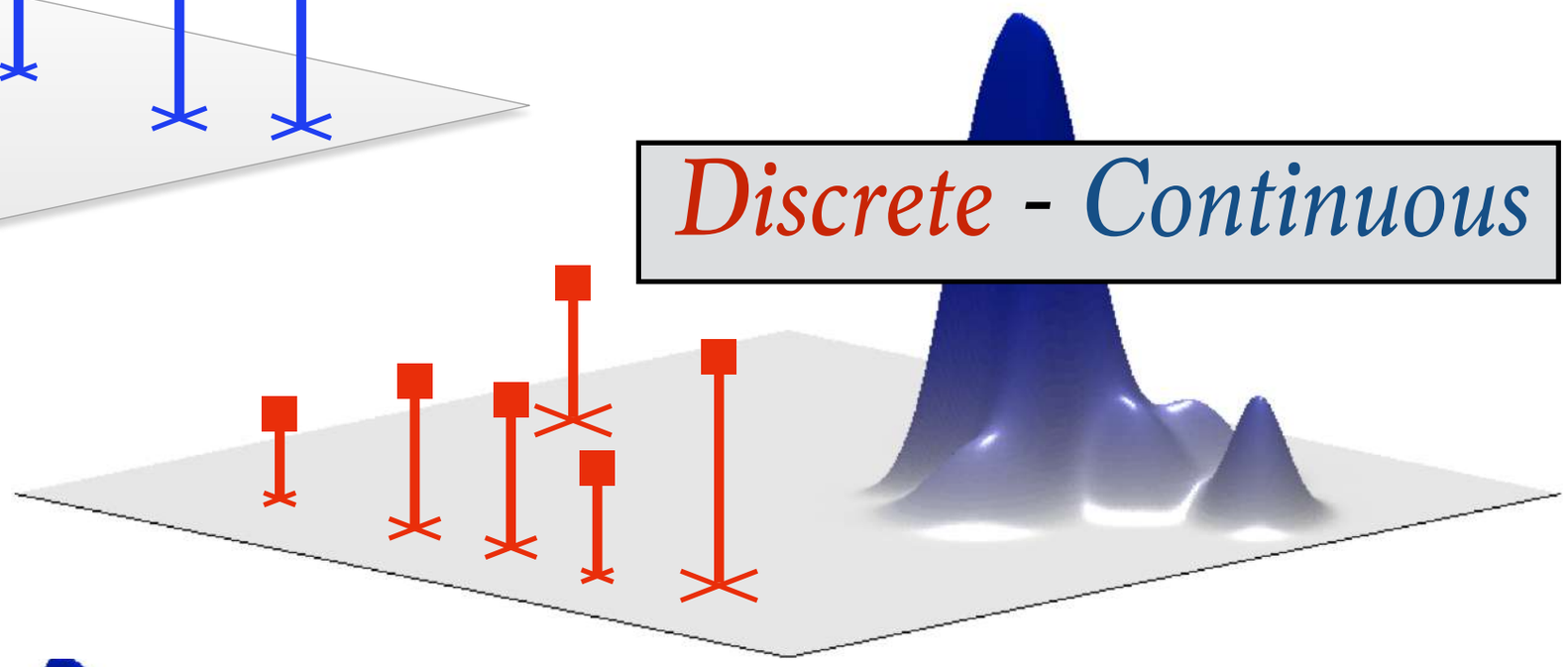
- Typology: discrete/continuous problems
- Easy cases and exact solvers for discrete measures.

When can we compute OT?

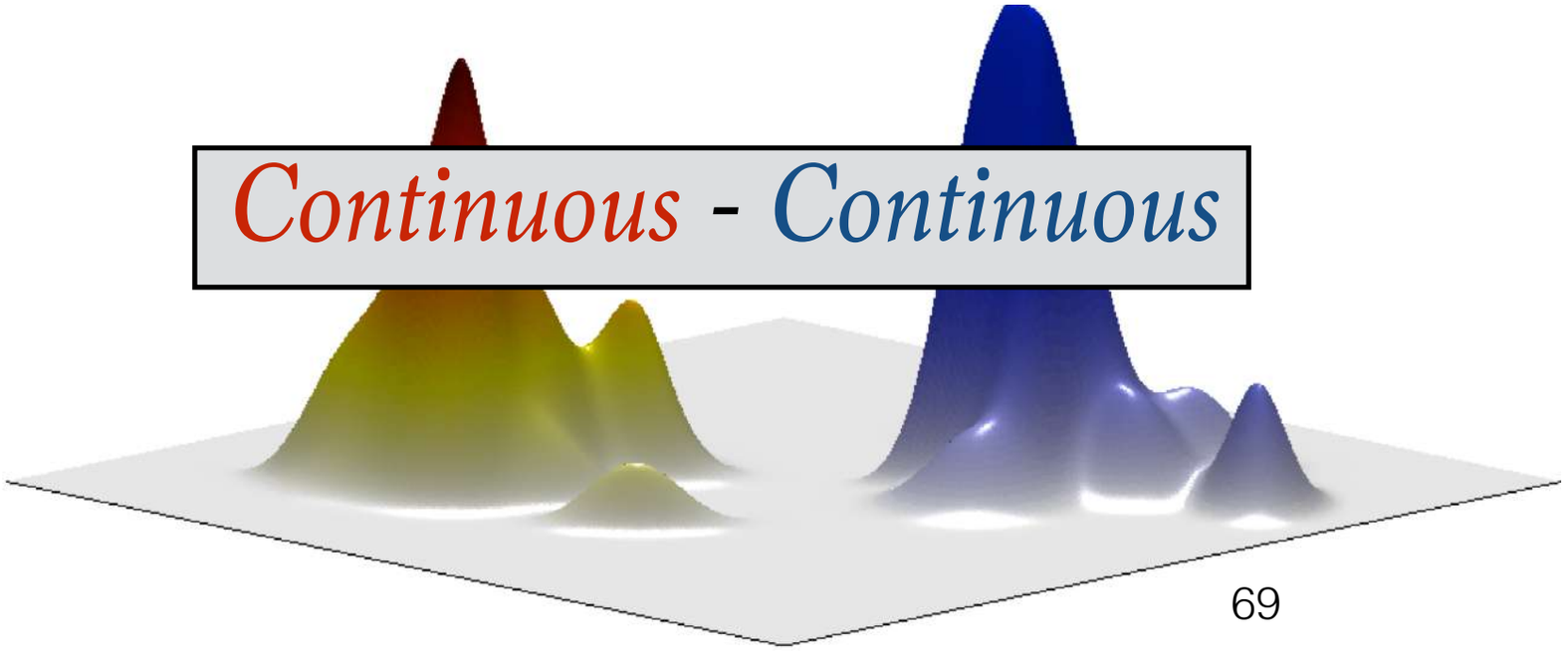
Discrete - Discrete



Discrete - Continuous

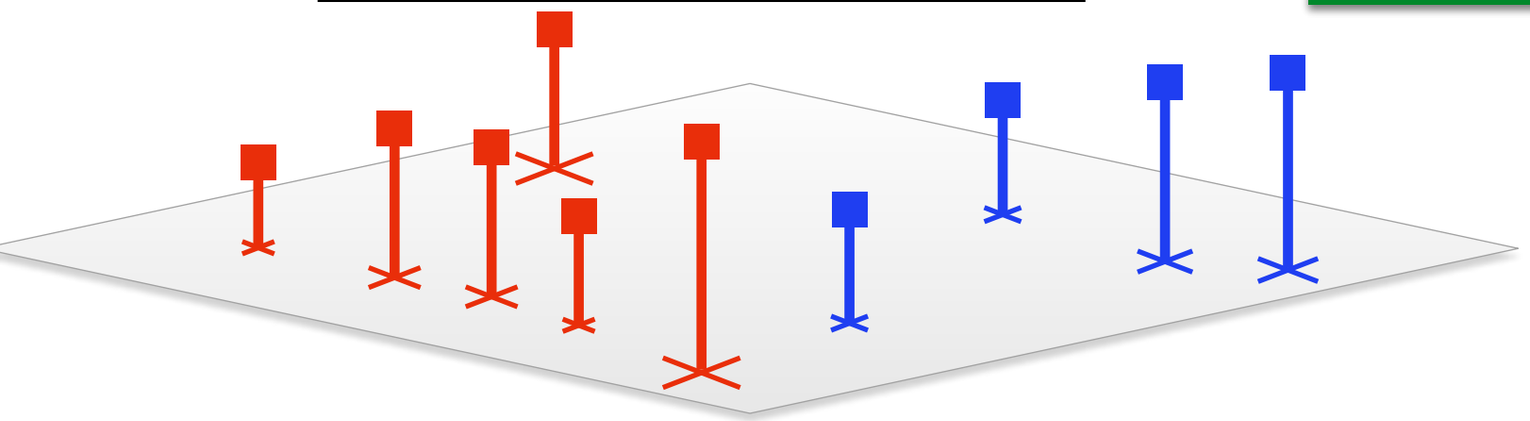


Continuous - Continuous



When can we compute OT?

Discrete - *Discrete*

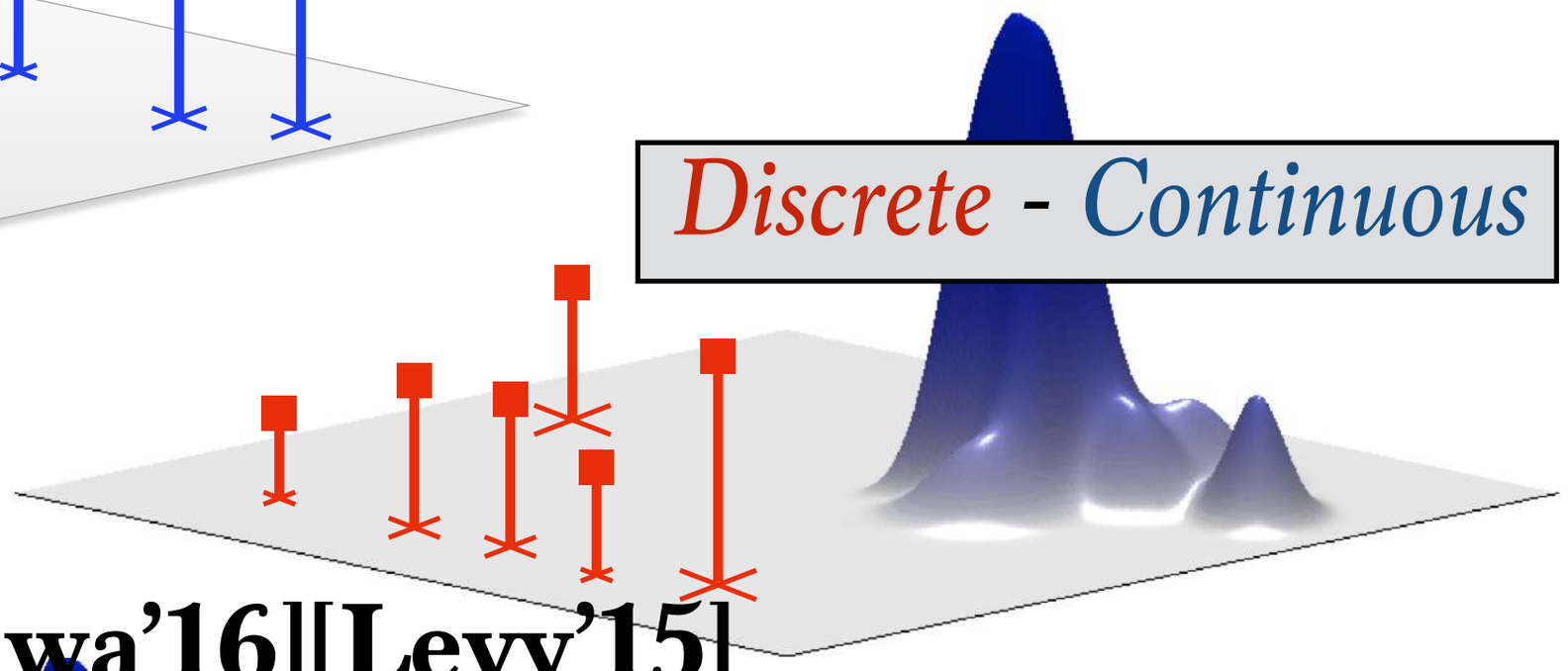


Network flow solvers

low dim.

[Mérigot'11][Kitagawa'16][Levy'15]

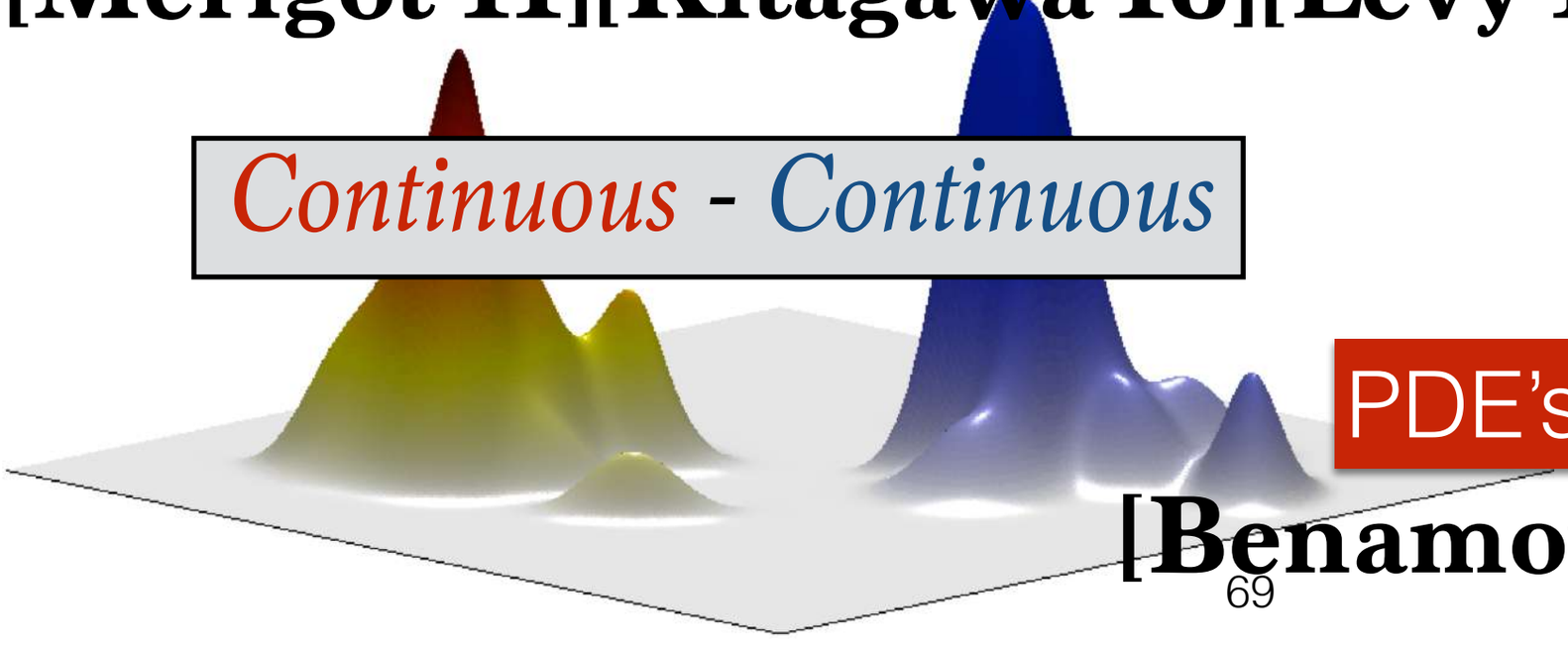
Discrete - *Continuous*



Stochastic Optimization

[Genevay'16]
[Arjovsky'17]

Continuous - *Continuous*



PDE's

[Benamou'98]

Easy (1): Univariate Measures

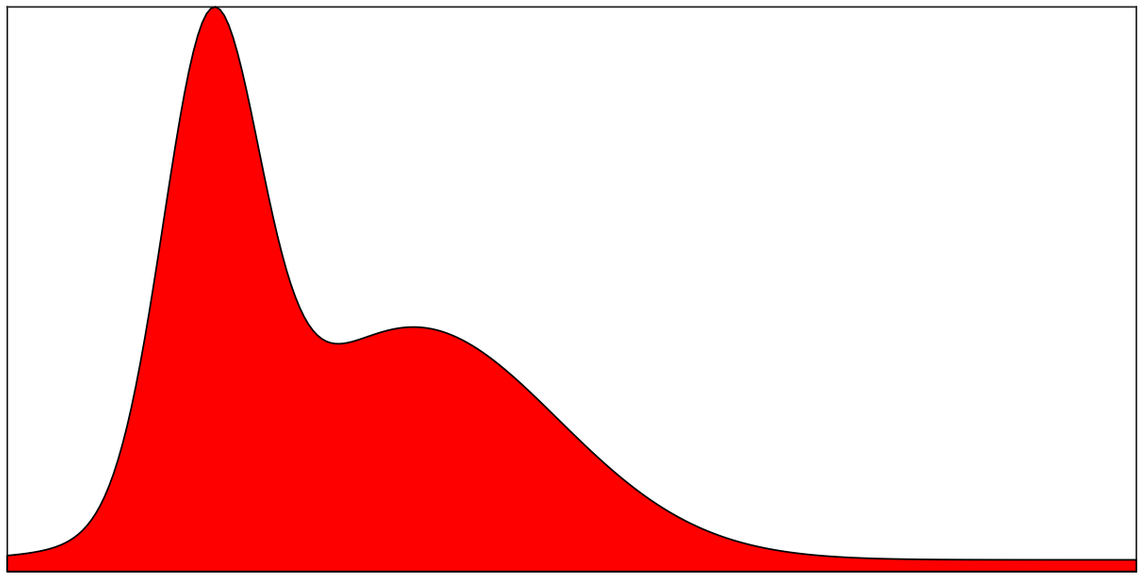
Remark. If $\Omega = \mathbb{R}$, $\boldsymbol{c}(x,y) = \boldsymbol{c}(|x-y|)$, $\boldsymbol{c}$ convex, $F_{\boldsymbol{\mu}}^{-1}, F_{\boldsymbol{\nu}}^{-1}$ quantile functions,

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

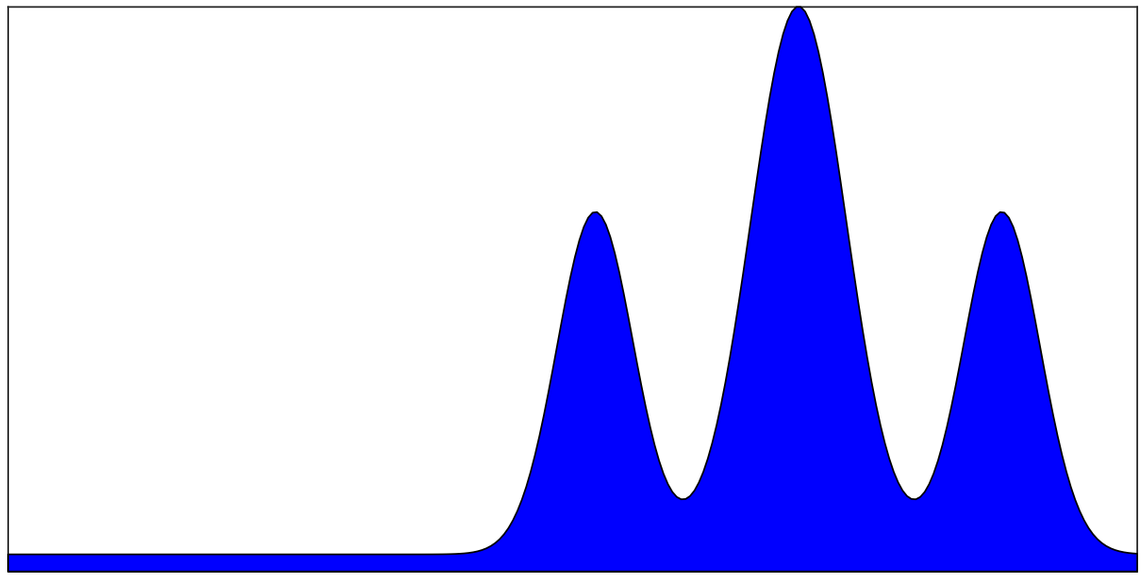
Easy (1): Univariate Measures

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

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



$\boldsymbol{\mu}$

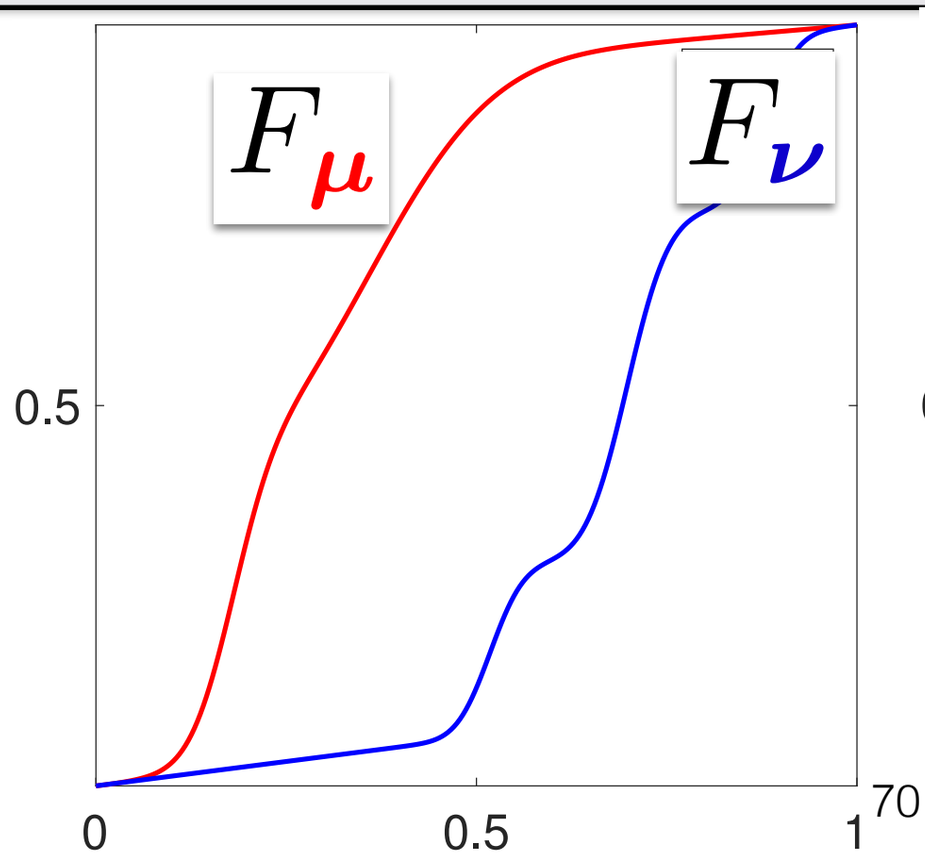


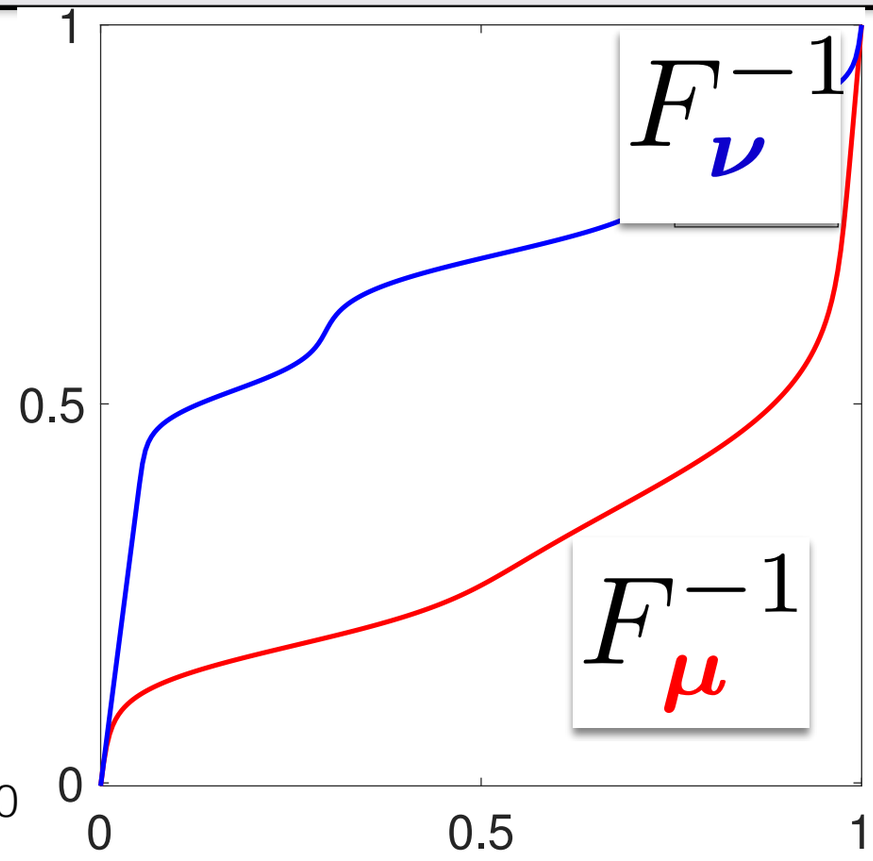
$\boldsymbol{\nu}$

Easy (1): Univariate Measures

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

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

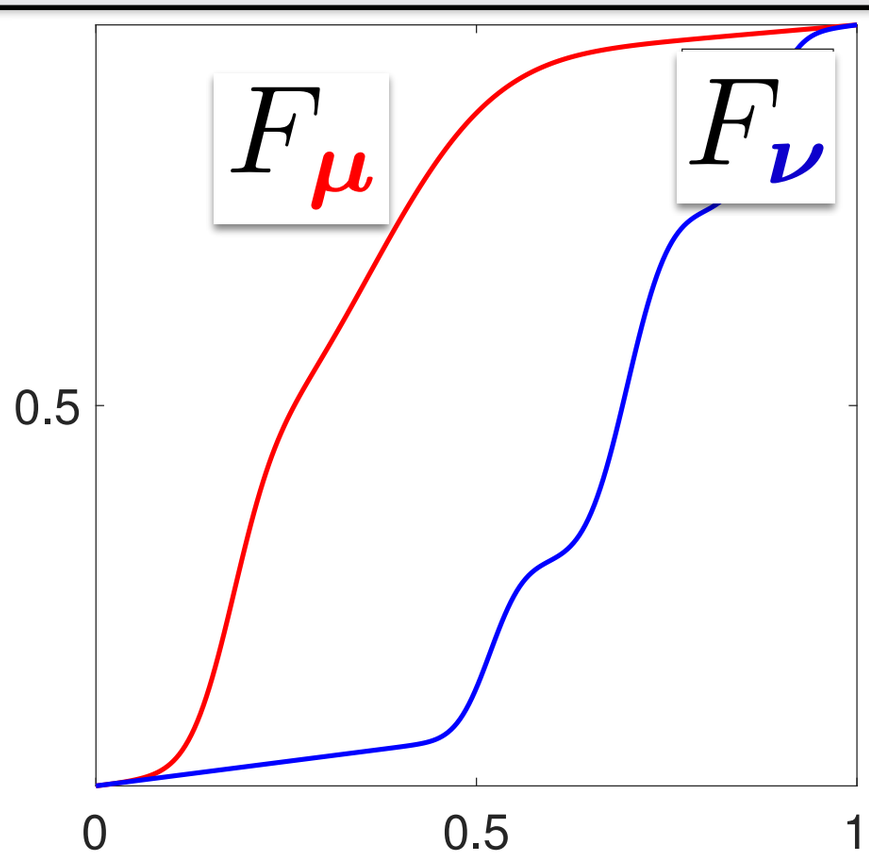


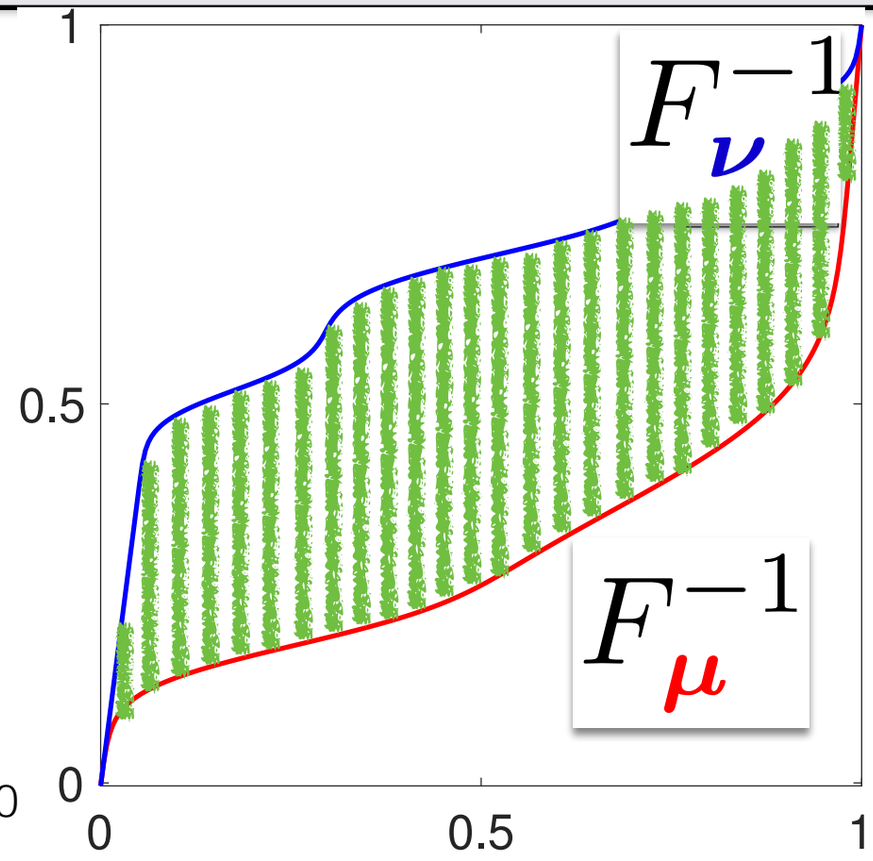


Easy (1): Univariate Measures

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

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





Easy (2): Gaussian Measures

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

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

where B is the Bures metric

$$B(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}})^2 = \operatorname{trace}(\boldsymbol{\Sigma}_{\boldsymbol{\mu}} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} - 2(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2} \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2})^{1/2}).$$

Easy (2): Gaussian Measures

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

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

where B is the Bures metric

$$B(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}})^2 = \operatorname{trace}(\boldsymbol{\Sigma}_{\boldsymbol{\mu}} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} - 2(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2} \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2})^{1/2}).$$

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

Easy (2): Gaussian Measures

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

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

where B is the Bures

if $X \sim \mathcal{N}(m, \boldsymbol{\Sigma})$ then $Y := CX + b \sim \mathcal{N}(Cm + b, C\boldsymbol{\Sigma}C^T)$.

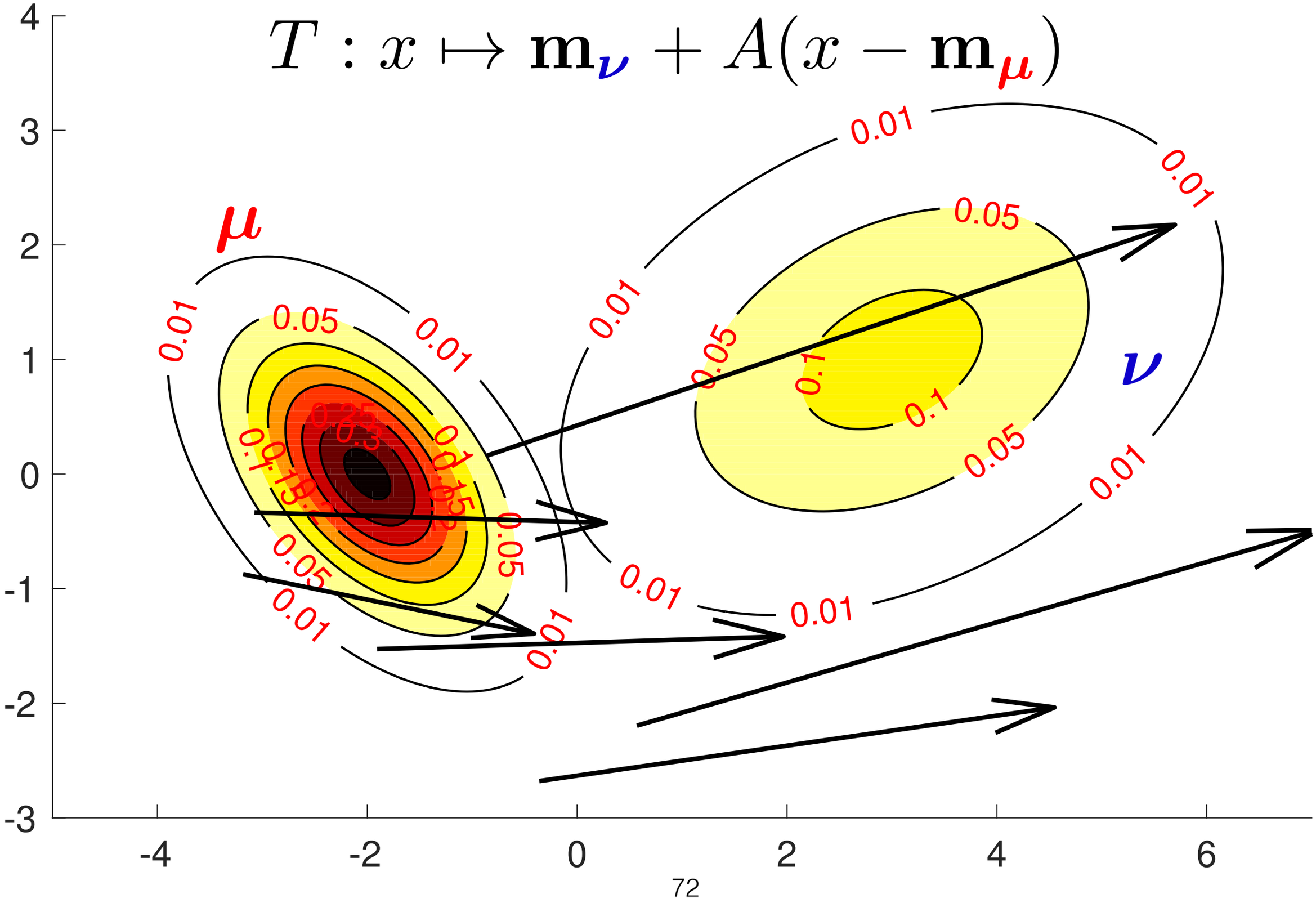
$\text{trace}(\boldsymbol{\Sigma}_{\boldsymbol{\mu}} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} - 2(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2} \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{1/2})^{1/2})$.

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

where $A = \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-\frac{1}{2}} \left(\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{\frac{1}{2}} \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{\frac{1}{2}} \right)^{\frac{1}{2}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-\frac{1}{2}}$.

Easy (2): Gaussian Measures

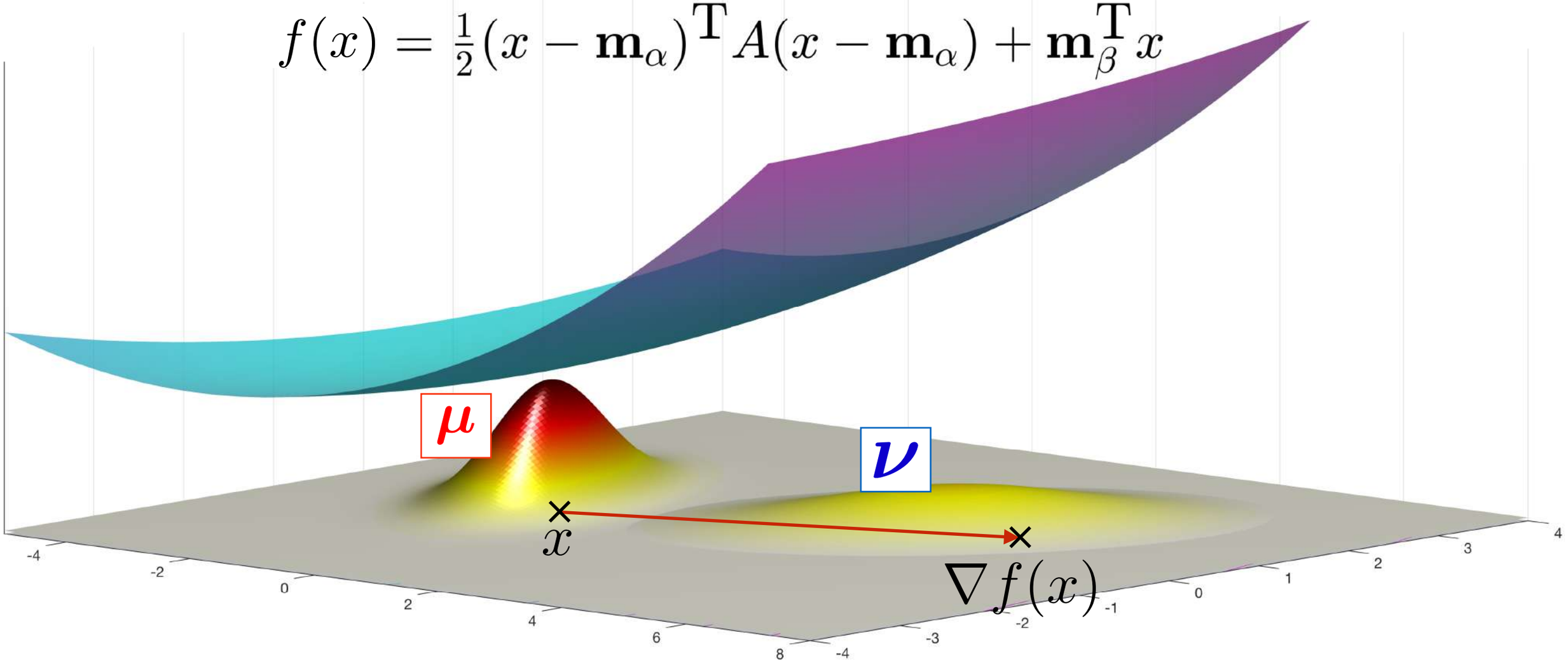
$$T : x \mapsto \mathbf{m}_{\boldsymbol{\nu}} + A(x - \mathbf{m}_{\boldsymbol{\mu}})$$



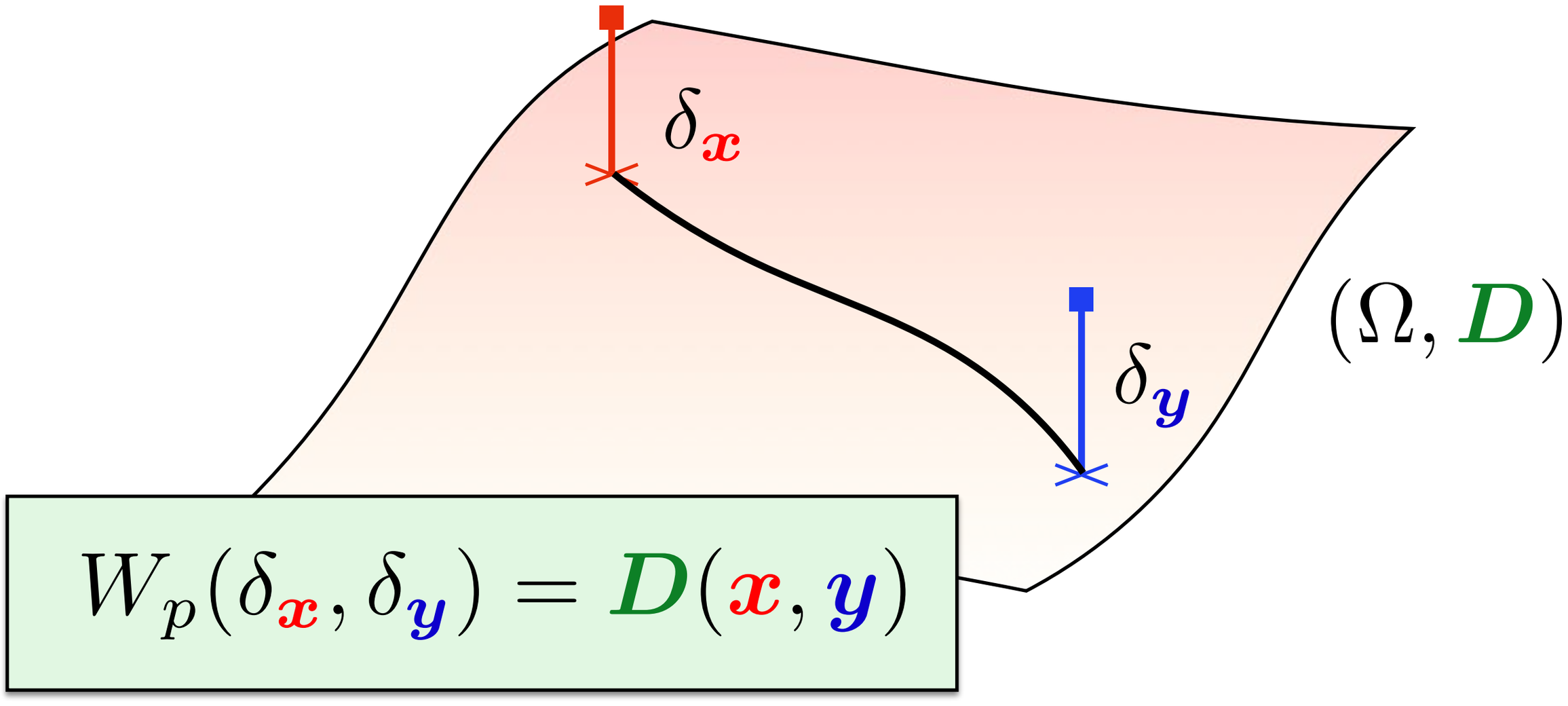
Easy (2): Gaussian Measures

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

$$f(x) = \tfrac{1}{2}(x - \mathbf{m}_\alpha)^{\mathrm{T}} A(x - \mathbf{m}_\alpha) + \mathbf{m}_\beta^{\mathrm{T}} x$$



Wasserstein Between Two Diracs

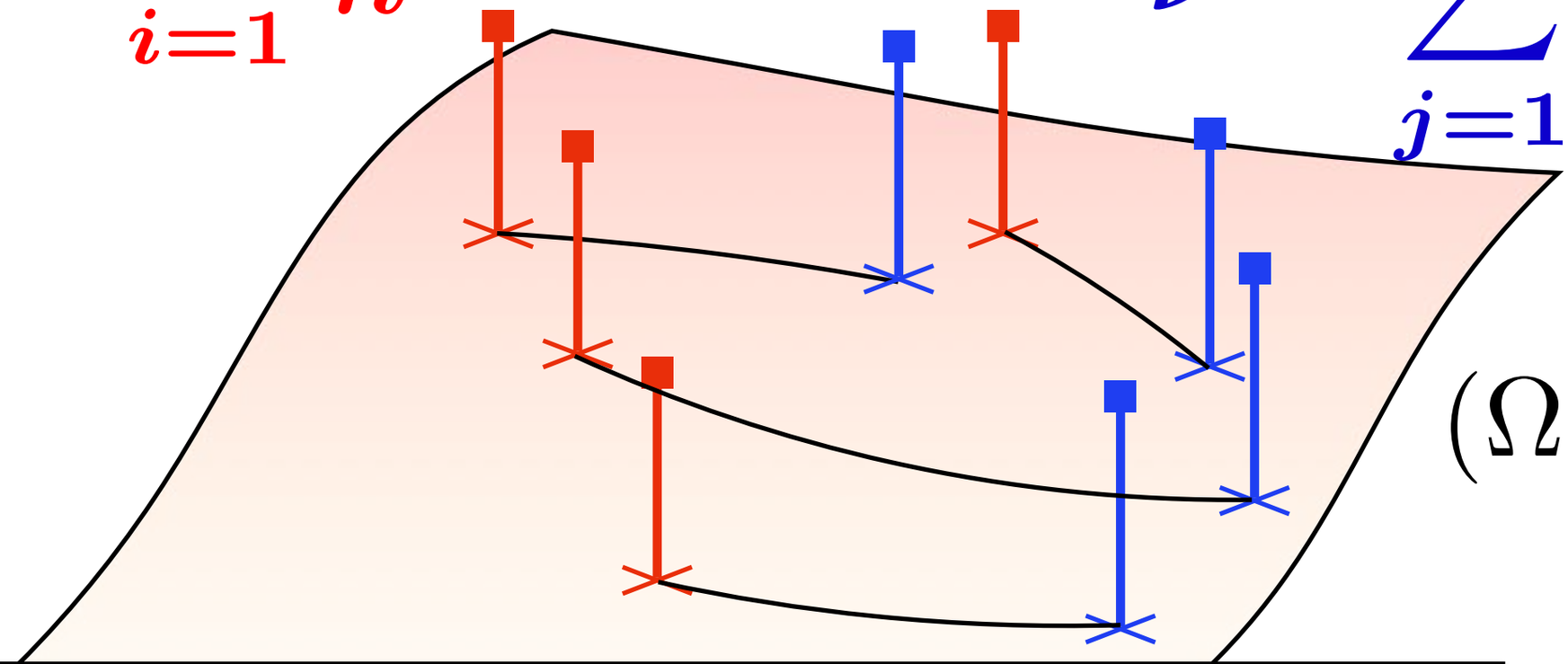


$$W_p(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}}) = \boldsymbol{D}(\boldsymbol{x}, \boldsymbol{y})$$

Linear Assignment $\subset$ Wasserstein

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

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



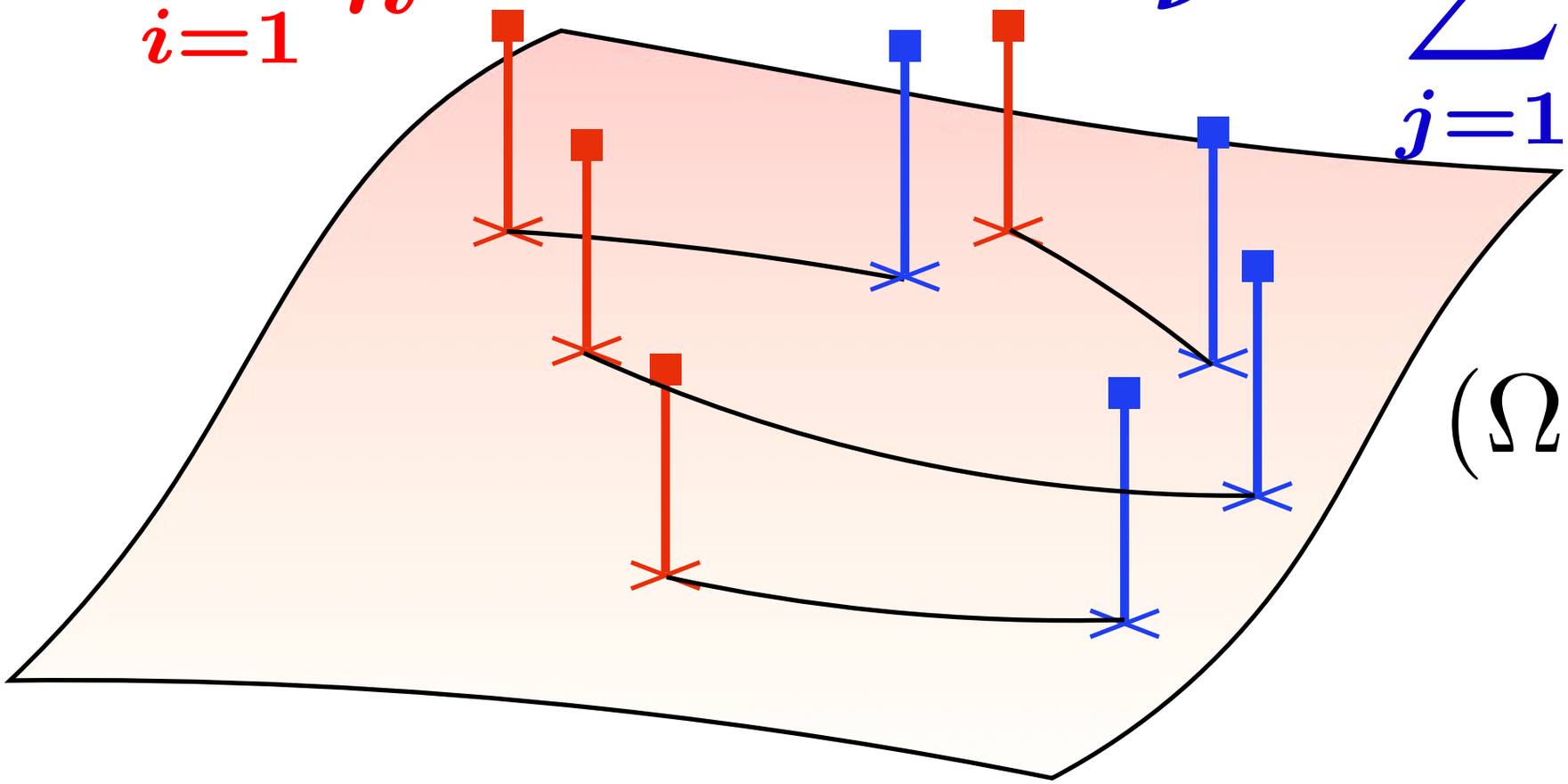
(Ω, D)

$$W_p^p(\mu, \nu) = \min_{\sigma \in S_n} \frac{1}{n} \sum_{i=1}^{n} D(x_i, y_{\sigma_i})^p$$

Linear Assignment $\subset$ Wasserstein

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

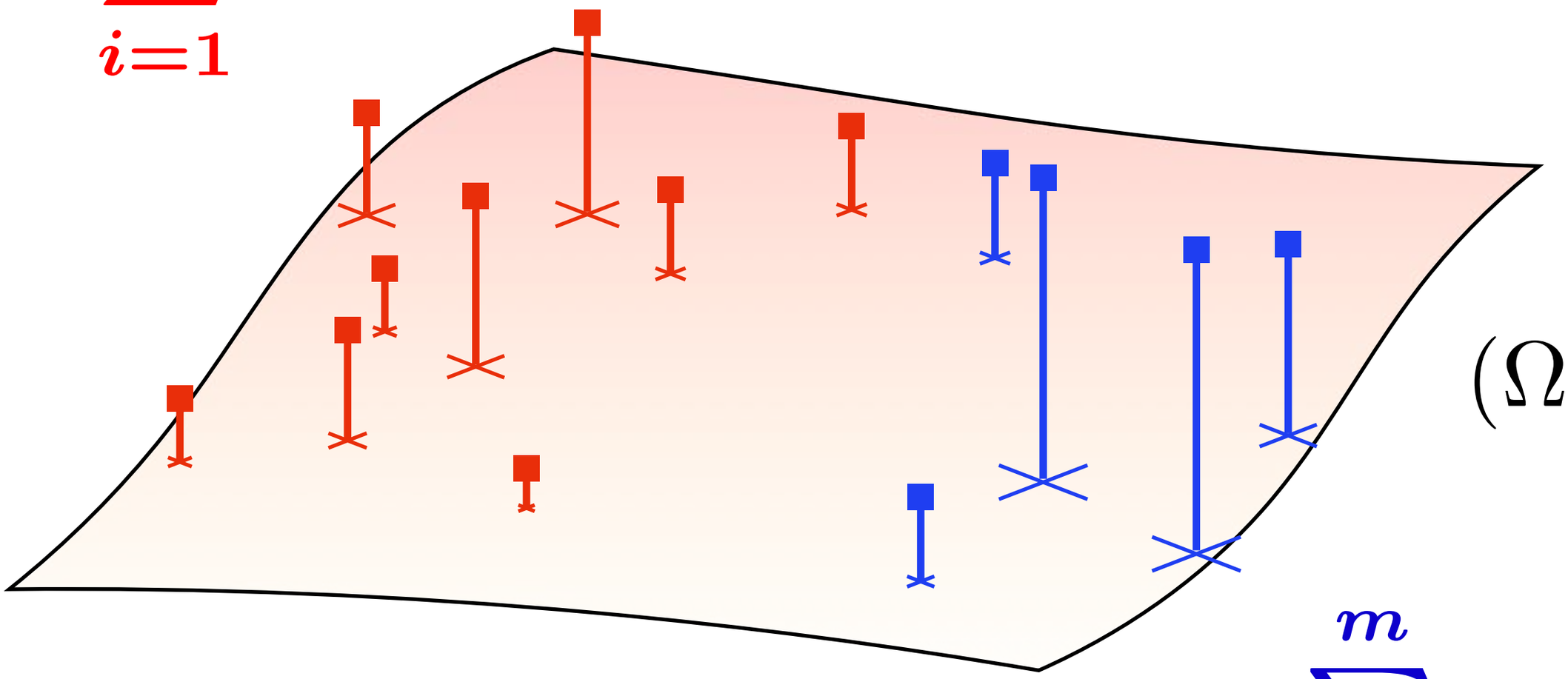
$$\nu = \sum_{j=1}^{n} \frac{1}{n} \delta_{y_j}$$



$$(\Omega, D)$$

OT on Two Empirical Measures

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

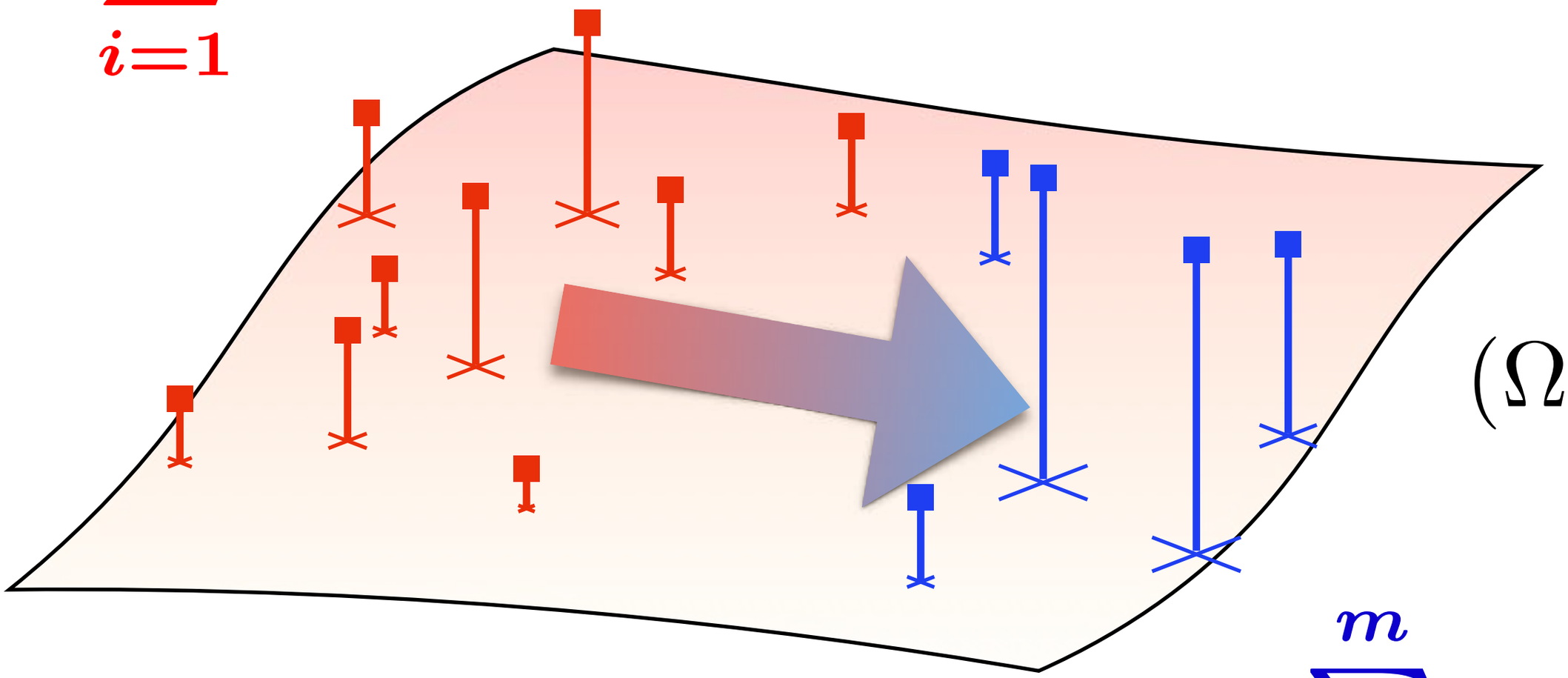


(Ω, D)

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

OT on Two Empirical Measures

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



(Ω, D)

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

Wasserstein on Empirical Measures

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

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

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

Def. Optimal Transport Problem

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

Dual Kantorovich Problem

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

Dual Kantorovich Problem

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

Def. Dual OT problem

$$W_p^p(\boldsymbol{\mu},\boldsymbol{\nu}) = \max_{\boldsymbol{\alpha}\in\mathbb{R}^n,\boldsymbol{\beta}\in\mathbb{R}^m} \alpha^T\boldsymbol{a}+\beta^T\boldsymbol{b}$$

$$\boldsymbol{\alpha}_i+\boldsymbol{\beta}_j\leq D(\boldsymbol{x}_i,\boldsymbol{y}_j)^p$$

Dual Kantorovich Problem

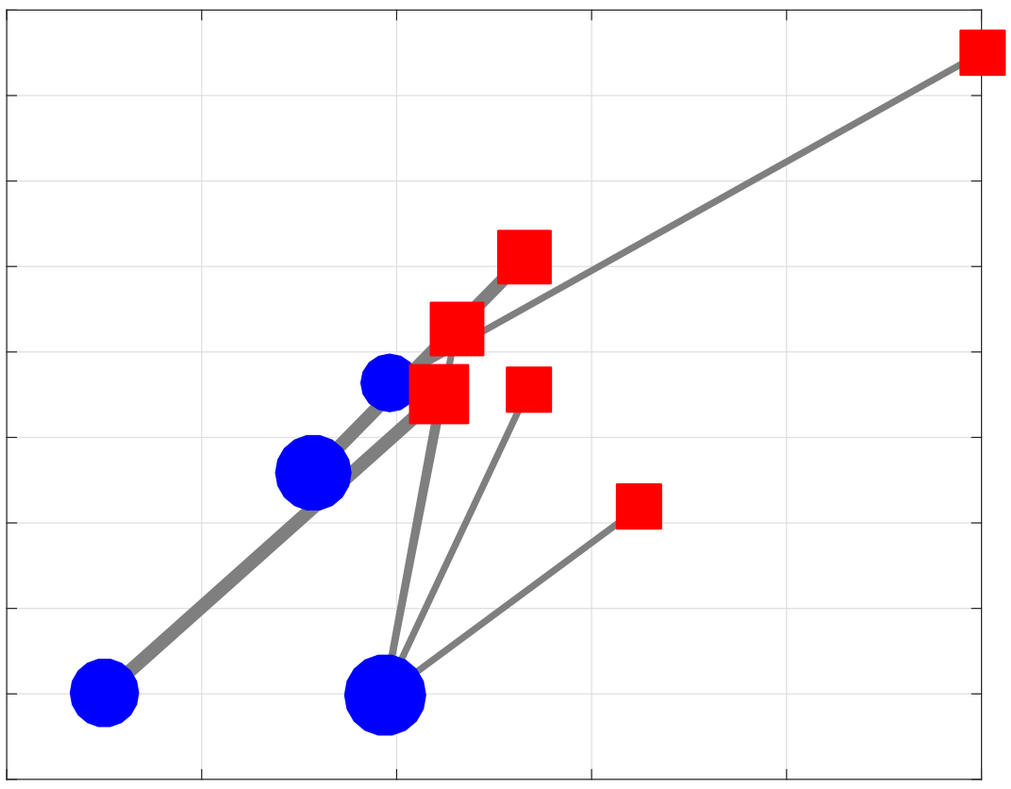
$$W_p^p(\boldsymbol{\mu},\boldsymbol{\nu}) = \min_{\boldsymbol{P}\in\mathbb{R}_+^{n\times m}} \langle \boldsymbol{P}, M_{\boldsymbol{XY}} \rangle$$

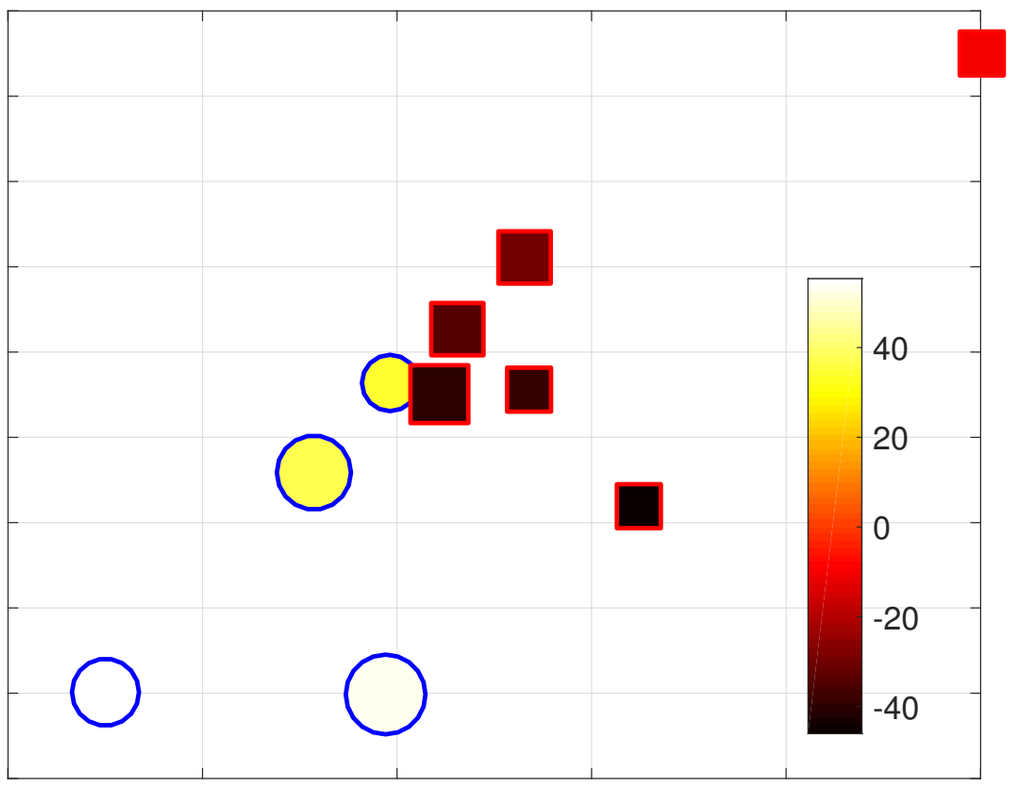
$$\boldsymbol{P}\mathbf{1}_m=\boldsymbol{a}, \boldsymbol{P}^T\mathbf{1}_n=\boldsymbol{b}$$

Def. Dual OT problem

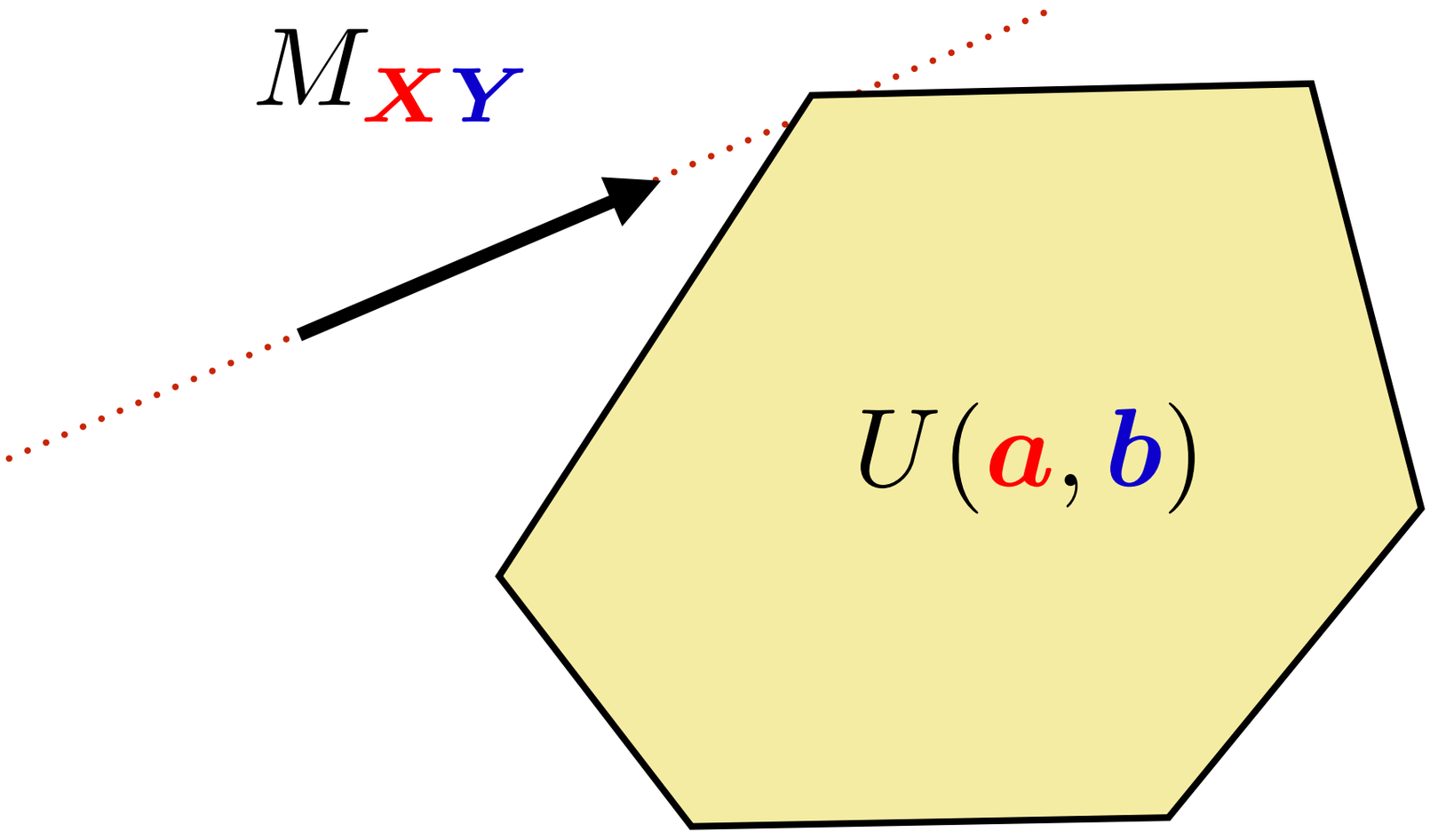
$$W_p^p(\boldsymbol{\mu},\boldsymbol{\nu}) = \max_{\boldsymbol{\alpha}\in\mathbb{R}^n,\boldsymbol{\beta}\in\mathbb{R}^m} \alpha^T\boldsymbol{a}+\beta^T\boldsymbol{b}$$

$$\boldsymbol{\alpha}_i+\boldsymbol{\beta}_j\leq D(\boldsymbol{x}_i,\boldsymbol{y}_j)^p$$





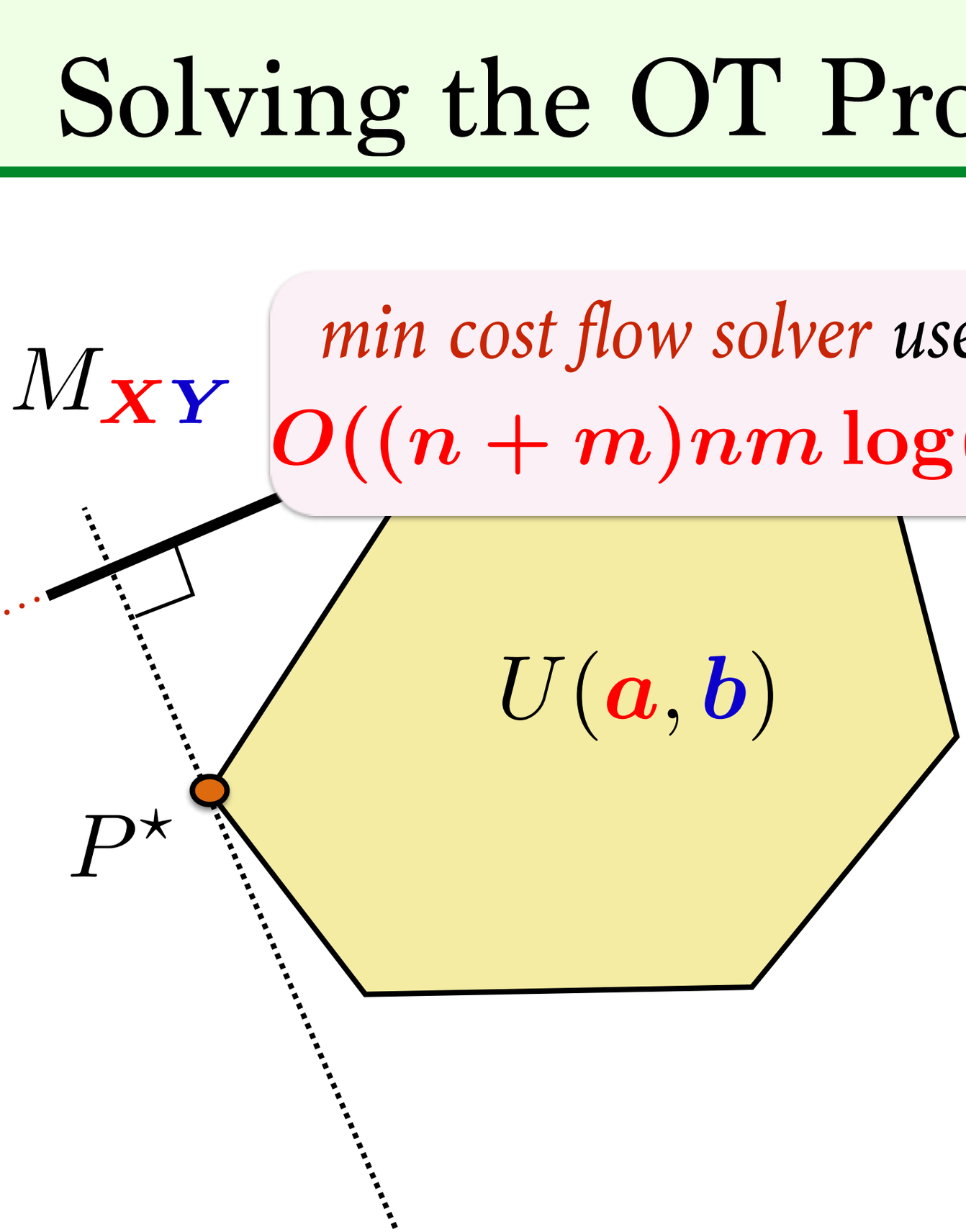
Solving the OT Problem



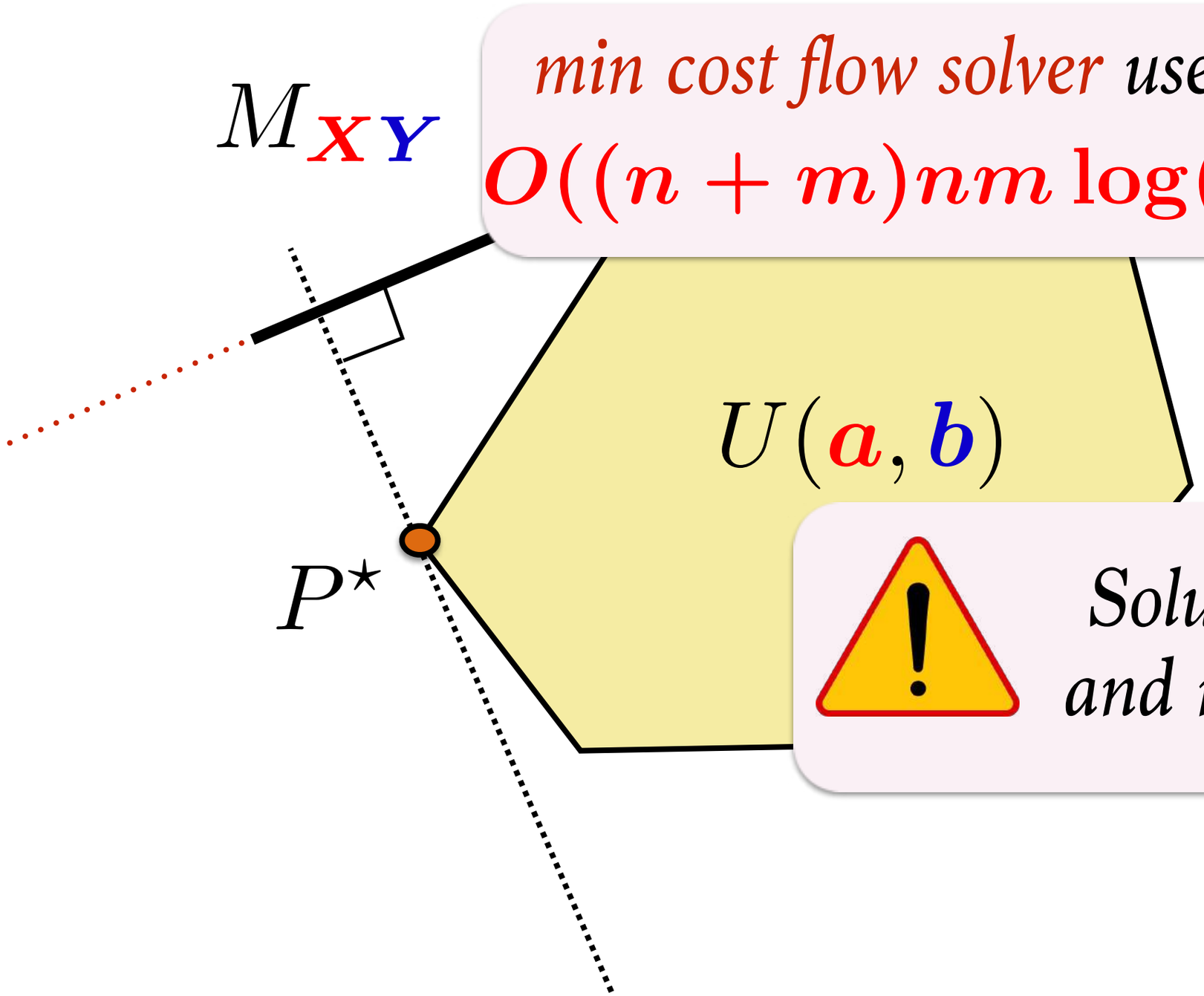
Solving the OT Problem

min cost flow solver used in practice.

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



Solving the OT Problem

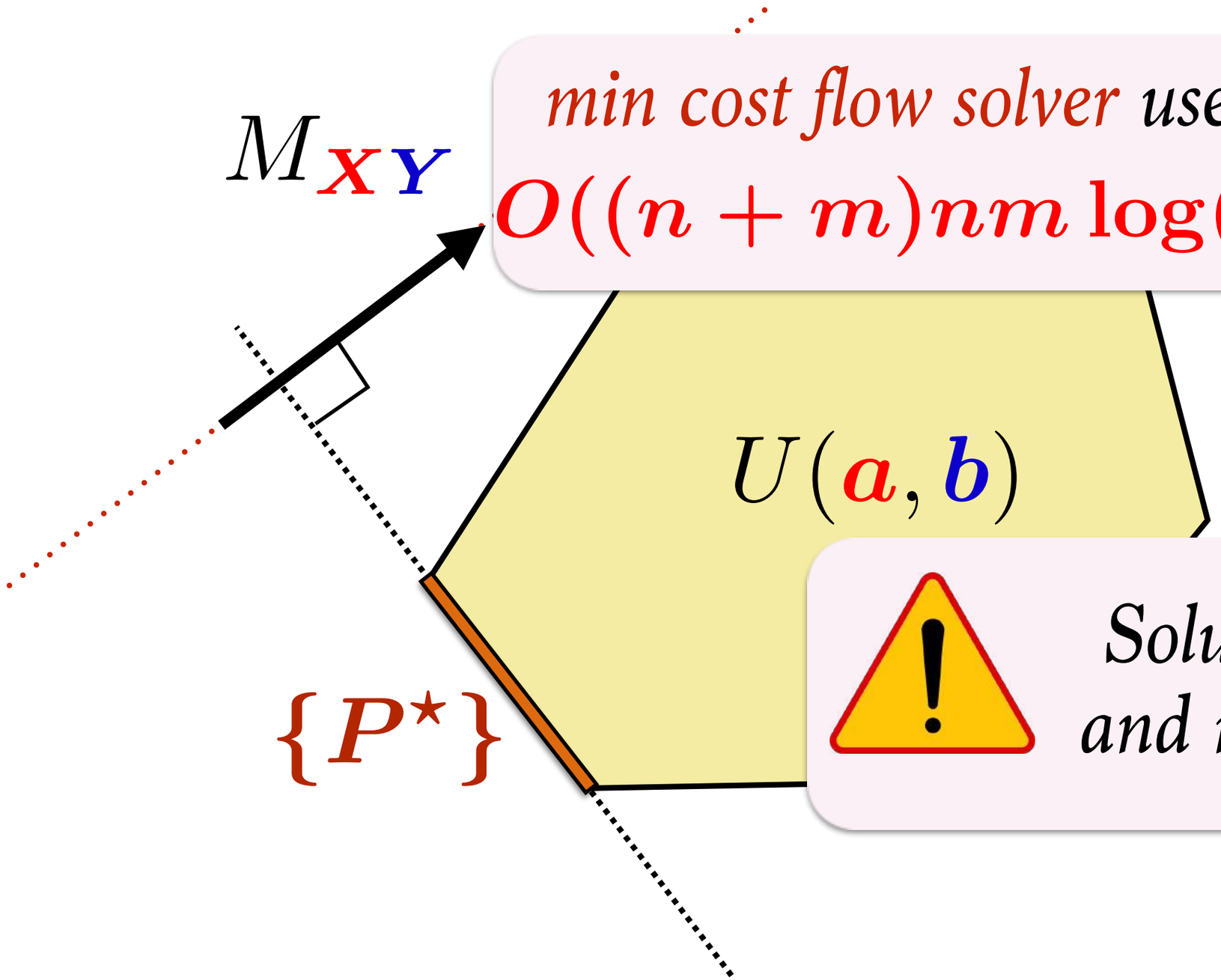


min cost flow solver used in practice.

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

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

Solving the OT Problem

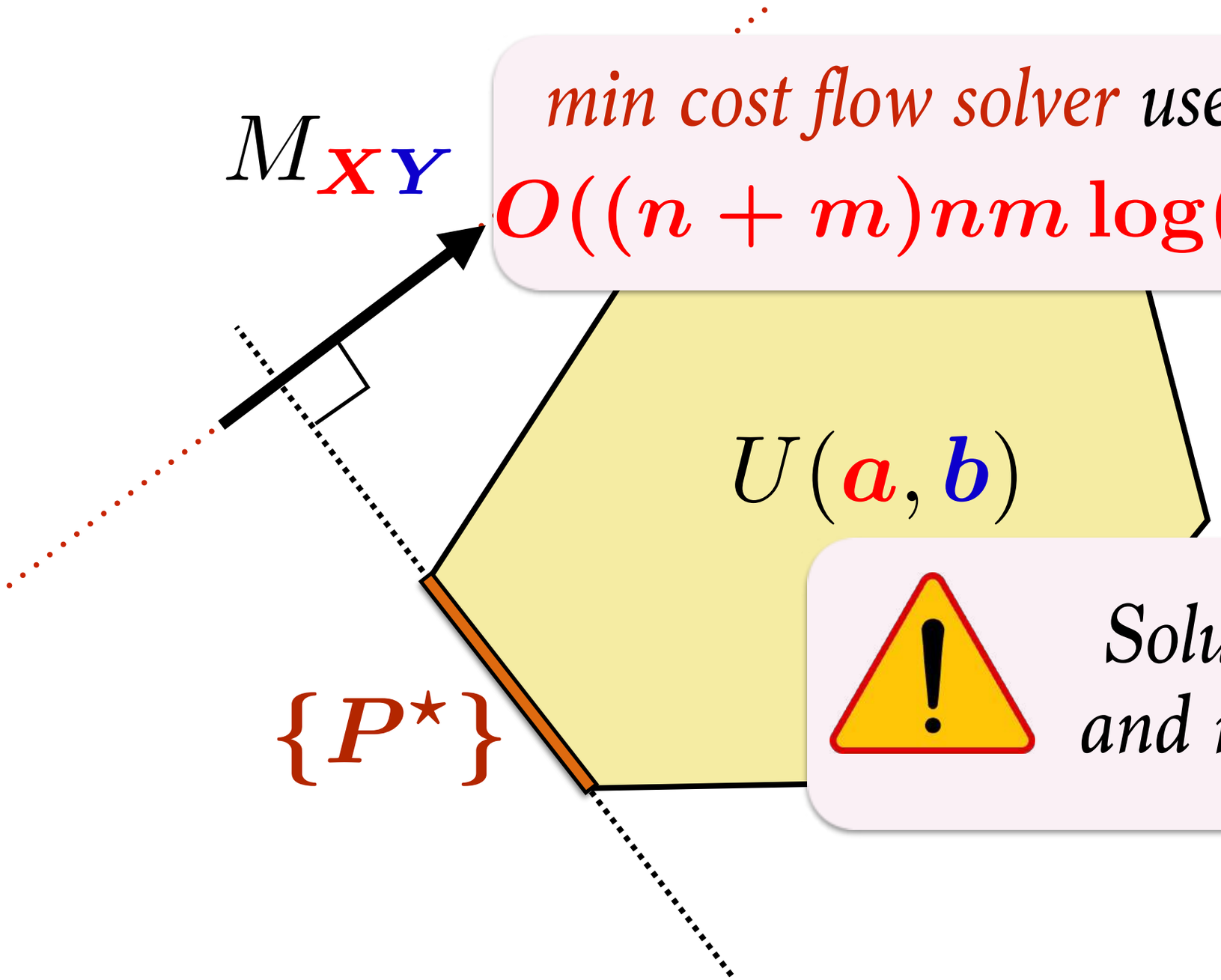


min cost flow solver used in practice.

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

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

Solving the OT Problem

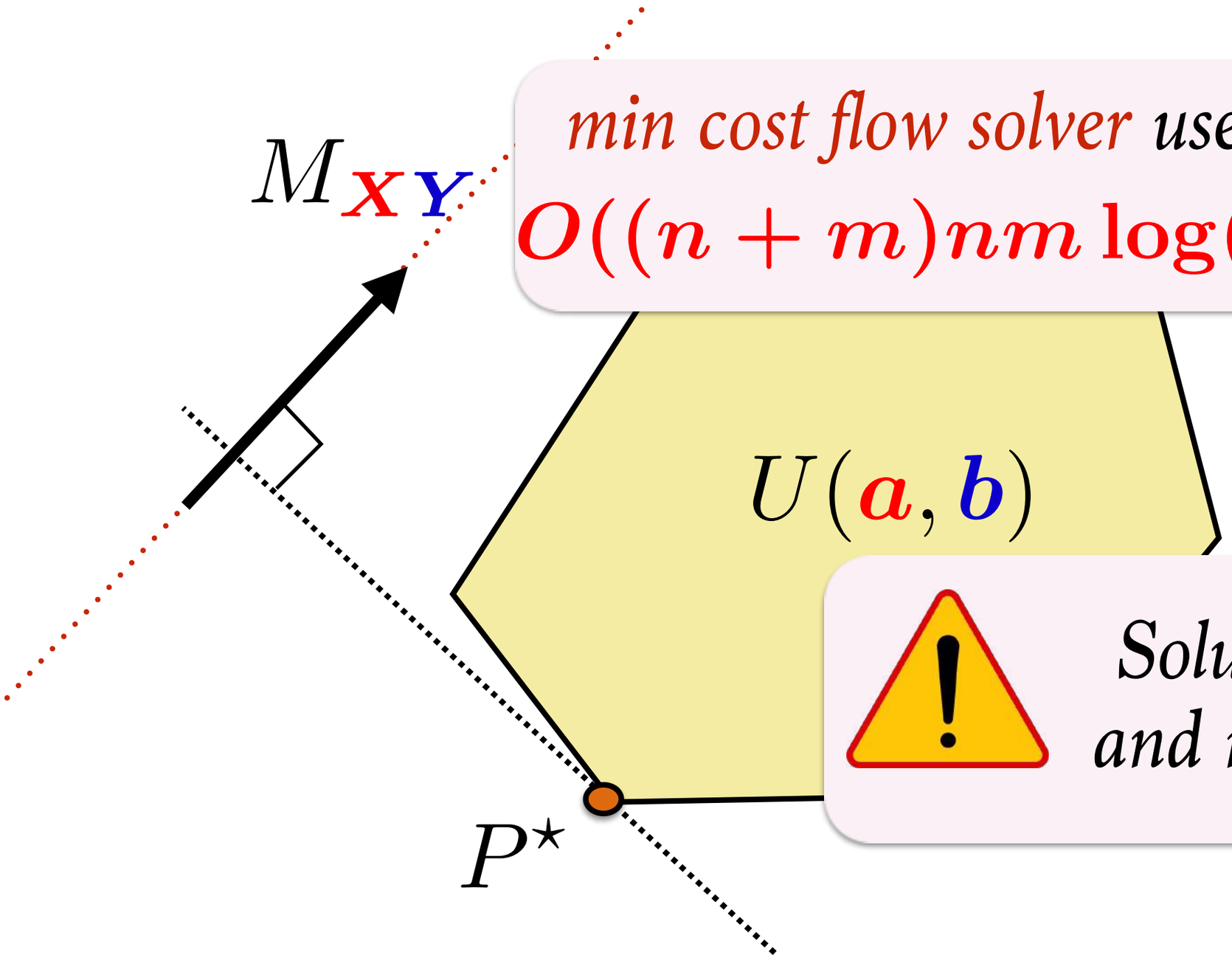


min cost flow solver used in practice.

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

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

Solving the OT Problem

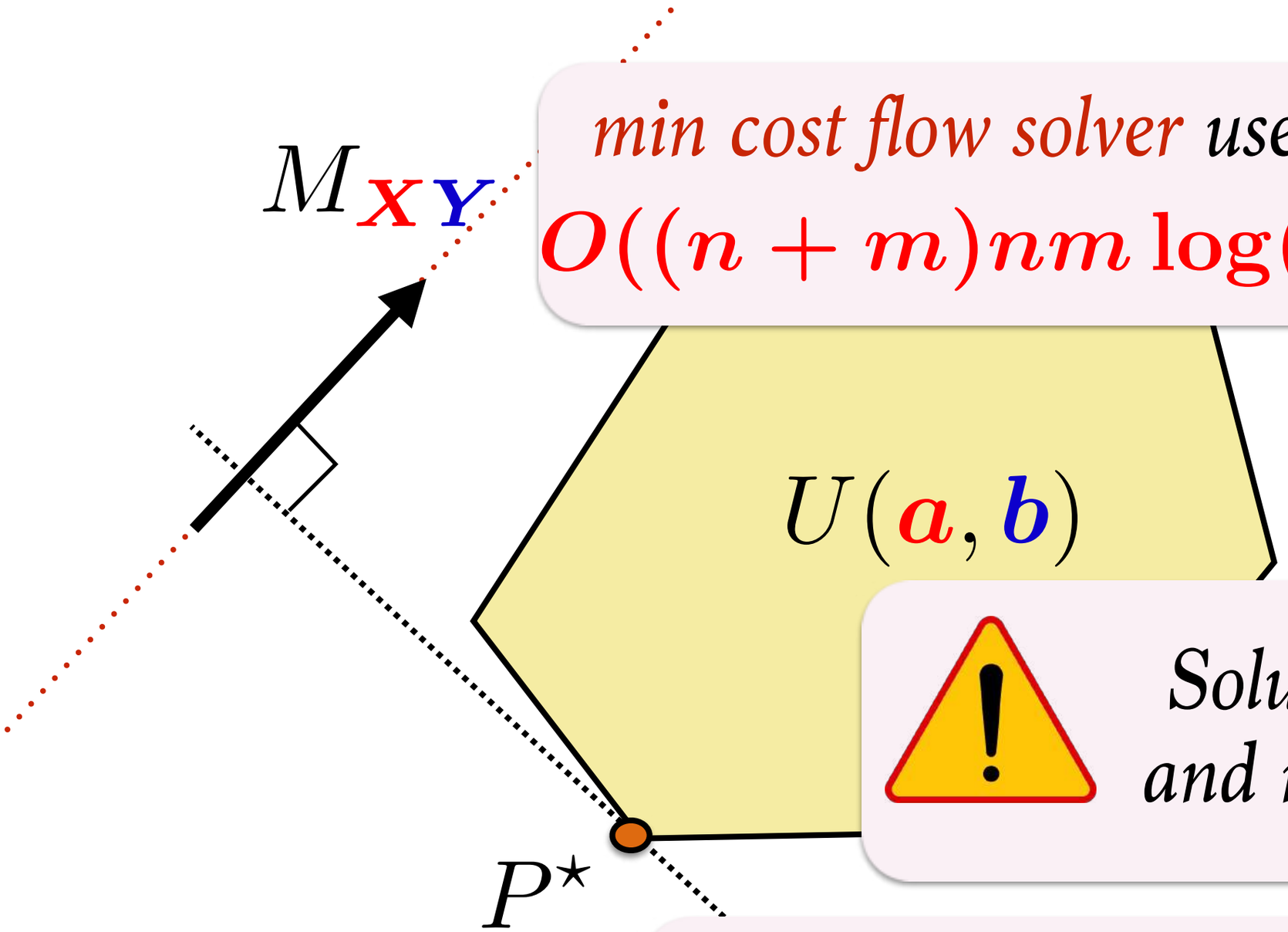


min cost flow solver used in practice.

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

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

Solving the OT Problem



min cost flow solver used in practice.

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

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

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

Discrete OT Problem

emd.c

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

```c
/*
    emd.c

    Last update: 3/14/98

    An implementation of the Earth Movers Distance.
    Based of the solution for the Transportation problem as described in
    "Introduction to Mathematical Programming" by F. S. Hillier and
    G. J. Lieberman, McGraw-Hill, 1990.

    Copyright (C) 1998 Yossi Rubner
    Computer Science Department, Stanford University
    E-Mail: rubner@cs.stanford.edu    URL: http://vision.stanford.edu/~rubner
*/

/*#include <stdio.h>
#include <stdlib.h>*/
#include <math.h>

#include "emd.h"

#define DEBUG_LEVEL 0
/*
 DEBUG_LEVEL:
   0 = NO MESSAGES
   1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
   2 = PRINT THE RESULT AFTER EVERY ITERATION
   3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
   4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
*/


#define MAX_SIG_SIZE1  (MAX_SIG_SIZE+1)  /* FOR THE POSIBLE DUMMY FEATURE */

/* NEW TYPES DEFINITION */

/* node1_t IS USED FOR SINGLE-LINKED LISTS */
typedef struct node1_t {
  int i;
  double val;
  struct node1_t *Next;
} node1_t;

/* node1_t IS USED FOR DOUBLE-LINKED LISTS */
typedef struct node2_t {
  int i, j;
  double val;
  struct node2_t *NextC;                /* NEXT COLUMN */
  struct node2_t *NextR;                /* NEXT ROW */
} node2_t;


/* GLOBAL VARIABLE DECLARATION */
static int _n1, _n2;                              /* SIGNATURES SIZES */
static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1];/* THE COST MATRIX */
static node2_t _X[MAX_SIG_SIZE1*2];               /* THE BASIC VARIABLES VECTOR */
```

ice.

))

Discrete OT Problem

emd.c

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

```c
/*
    emd.c

    Last update: 3/14/98

    An implementation of the Earth Movers Distance.
    Based of the solution for the Transportation problem as described in
    "Introduction to Mathematical Programming" by F. S. Hillier and
    G. J. Lieberman, McGraw-Hill, 1990.

    Copyright (C) 1998 Yossi Rubner
    Computer Science Department, Stanford University
    E-Mail: rubner@cs.stanford.edu    URL: http://vision.stanford.edu/~rubner
*/

/*#include <stdio.h>
#include <stdlib.h>*/
#include <math.h>

#include "emd.h"

#define DEBUG_LEVEL 0
/*
 DEBUG_LEVEL:
   0 = NO MESSAGES
   1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
   2 = PRINT THE RESULT AFTER EVERY ITERATION
   3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
   4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
*/


#define MAX_SIG_SIZE1  (MAX_SIG_SIZE+1)  /* FOR THE POSIBLE DUMMY FEATURE */

/* NEW TYPES DEFINITION */

/* node1_t IS USED FOR SINGLE-LINKED LISTS */
typedef struct node1_t {
  int i;
  double val;
  struct node1_t *Next;
} node1_t;

/* node1_t IS USED FOR DOUBLE-LINKED LISTS */
typedef struct node2_t {
  int i, j;
  double val;
  struct node2_t *NextC;                /* NEXT COLUMN */
  struct node2_t *NextR;                /* NEXT ROW */
} node2_t;


/* GLOBAL VARIABLE DECLARATION */
static int _n1, _n2;                              /* SIGNATURES SIZES */
static float _C[MAX_SIG_SIZE1][MAX_SIG_SIZE1];/* THE COST MATRIX */
static node2_t _X[MAX_SIG_SIZE1*2];               /* THE BASIC VARIABLES VECTOR */
```

ice.

))

3. Computing OT for data sciences

- The need for regularization
- Entropic regularization
- Differentiation

What matters for practitioners?

i.i.d samples $\boldsymbol{x_1, \dots, x_n \sim \mu}$, $\boldsymbol{y_1, \dots, y_m \sim \nu}$,

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

Computational properties

Compute/approximate $\boldsymbol{W_p(\hat{\mu}_n, \hat{\nu}_m)}$?

Statistical properties

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

What matters for practitioners?

i.i.d samples $\boldsymbol{x_1, \dots, x_n \sim \mu}$, $\boldsymbol{y_1, \dots, y_m \sim \nu}$,

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

Computational properties

$$\boldsymbol{O((n+m)nm\log(n+m))}$$

Statistical properties

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

Sample Complexity

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

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

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

From theory to practice ?

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

Computational properties

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

Statistical properties

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

From theory to practice ?

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

Computational properties

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

Statistical properties

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

For data sciences, we must regularize the problem to improve on either/both aspects.

Many ways to regularize (dual) OT

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

Many ways to regularize (dual) OT

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

- RKHS for potentials **[GCBP'16]**, dualize/smooth indicator constraint, **[VMVRB'21]**

Many ways to regularize (dual) OT

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

- RKHS for potentials **[GCBP'16]**, dualize/smooth indicator constraint, **[VMVRB'21]**

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

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

Many ways to regularize (primal) OT

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

Many ways to regularize (primal) OT

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

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

Many ways to regularize (primal) OT

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

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

- Quantize measures first **[Canas+'12]**; use Gaussians **[Gelbrich'92]**[**Chen+17**]; projections **[Rabin+'11]** & k-dimensional subspaces **[Paty+'19]**[**Weed+'19**].

Many ways to regularize (primal) OT

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

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

- Quantize measures first **[Canas+'12]**; use Gaussians **[Gelbrich'92]**[**Chen+17**]; projections **[Rabin+'11]** & k-dimensional subspaces **[Paty+'19]**[**Weed+'19**].

- Add regularization on coupling **[C'13]**[**GP'16**] **[GCBP'16]** **[GCBCP'19]** **[DPR'16]** **[BSR'18]**

A Different Route: Projection

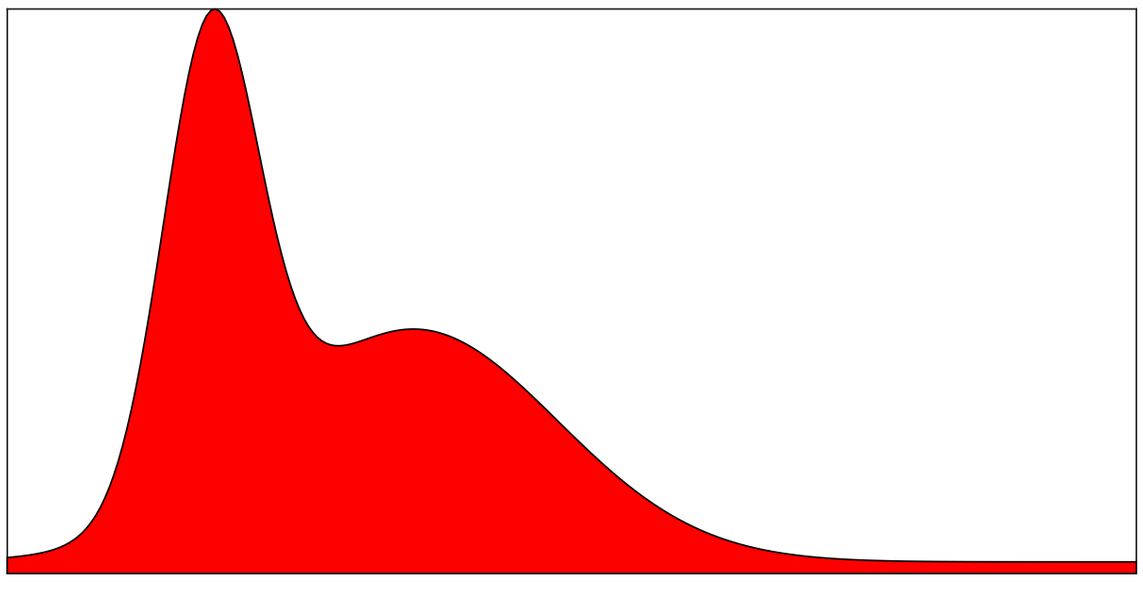
Remark. If $\Omega = \mathbb{R}$, $\boldsymbol{c}(x,y) = \boldsymbol{c}(|x-y|)$, $\boldsymbol{c}$ convex, $F_{\boldsymbol{\mu}}^{-1}, F_{\boldsymbol{\nu}}^{-1}$ quantile functions,

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

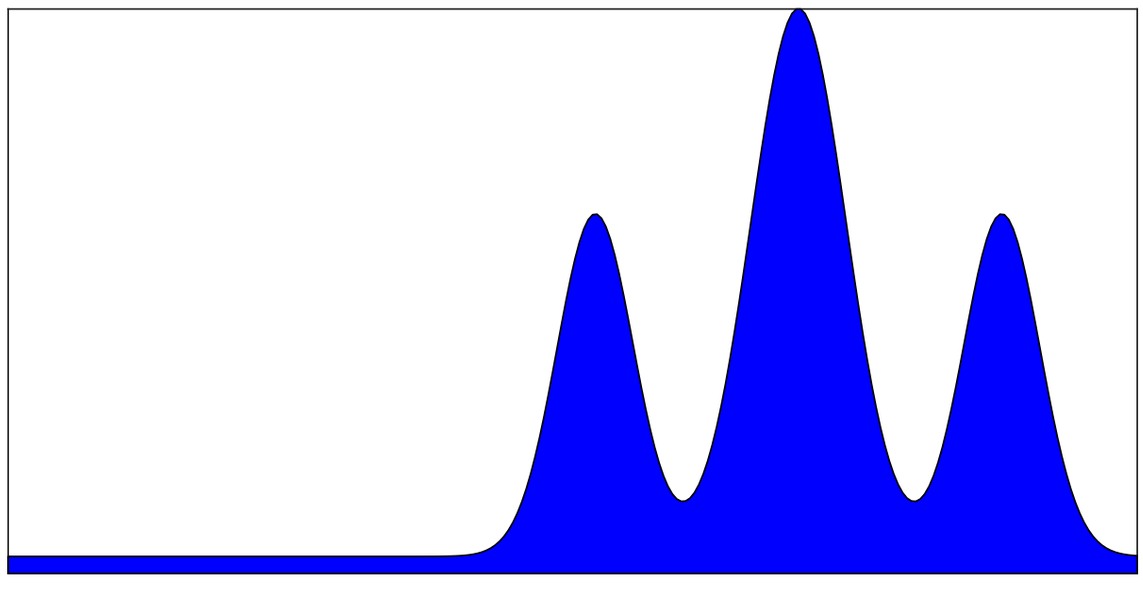
A Different Route: Projection

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

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



$\boldsymbol{\mu}$

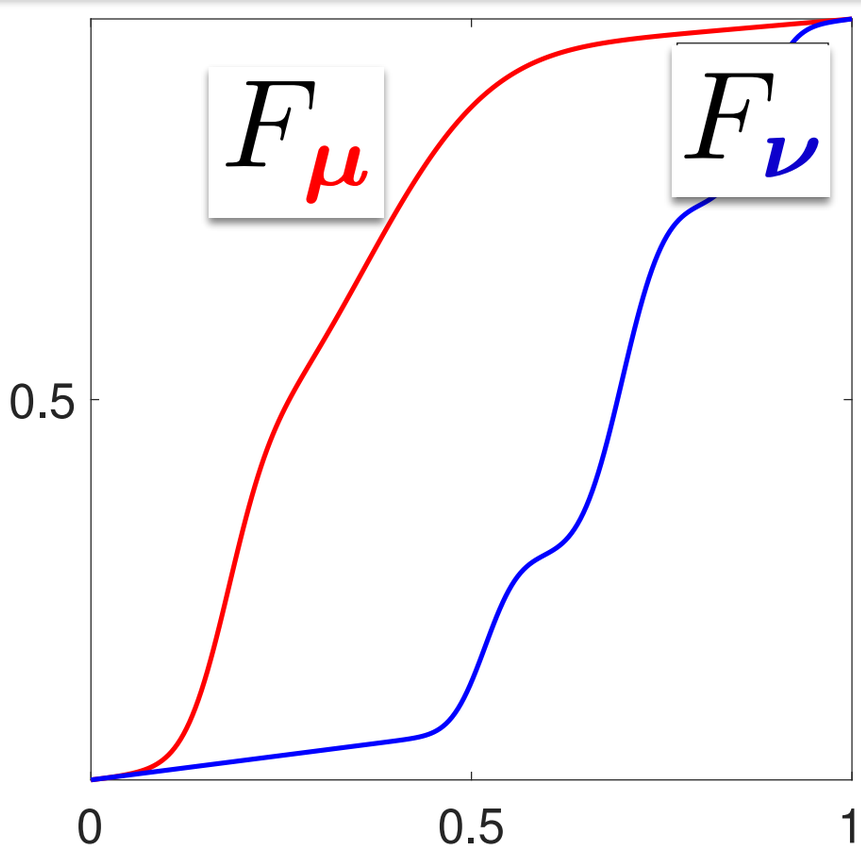


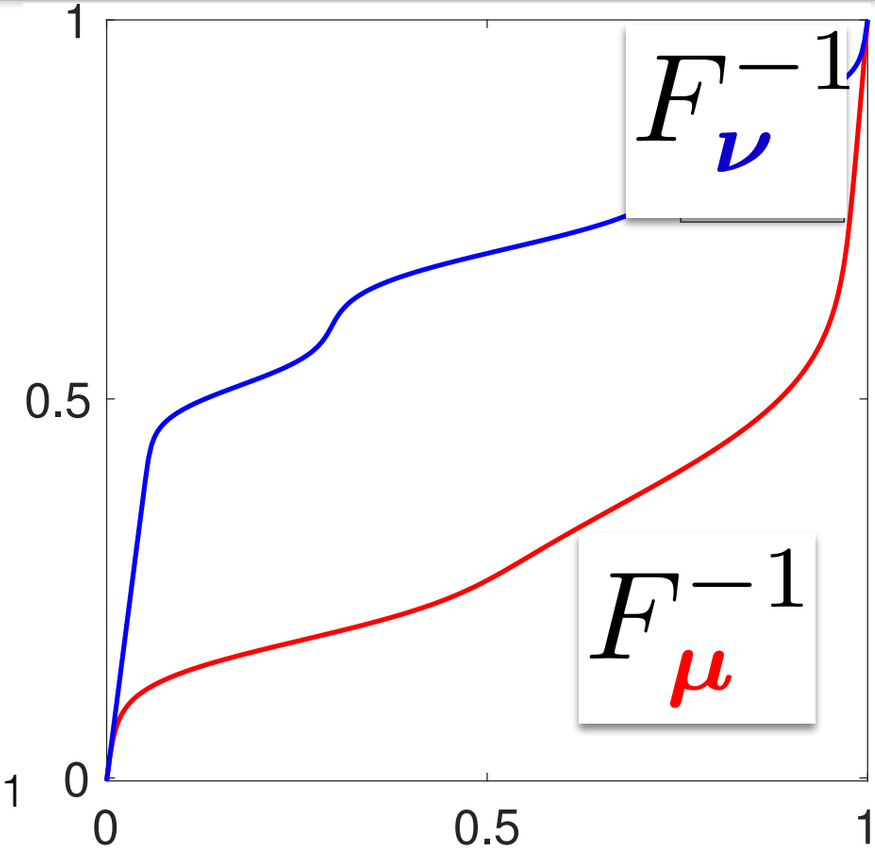
$\boldsymbol{\nu}$

A Different Route: Projection

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

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

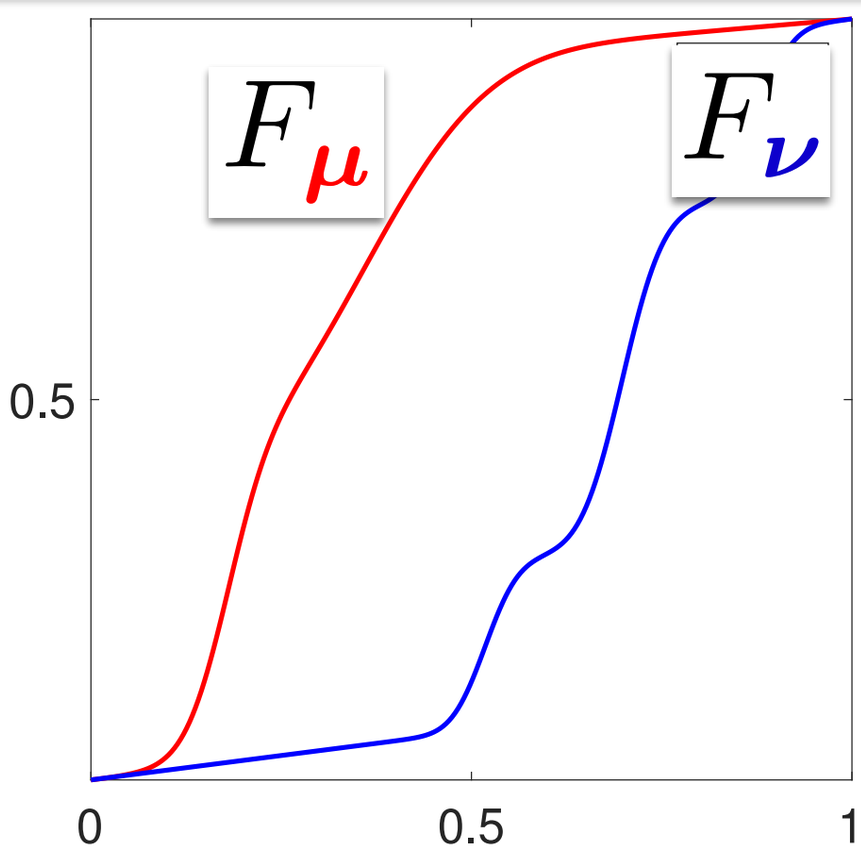


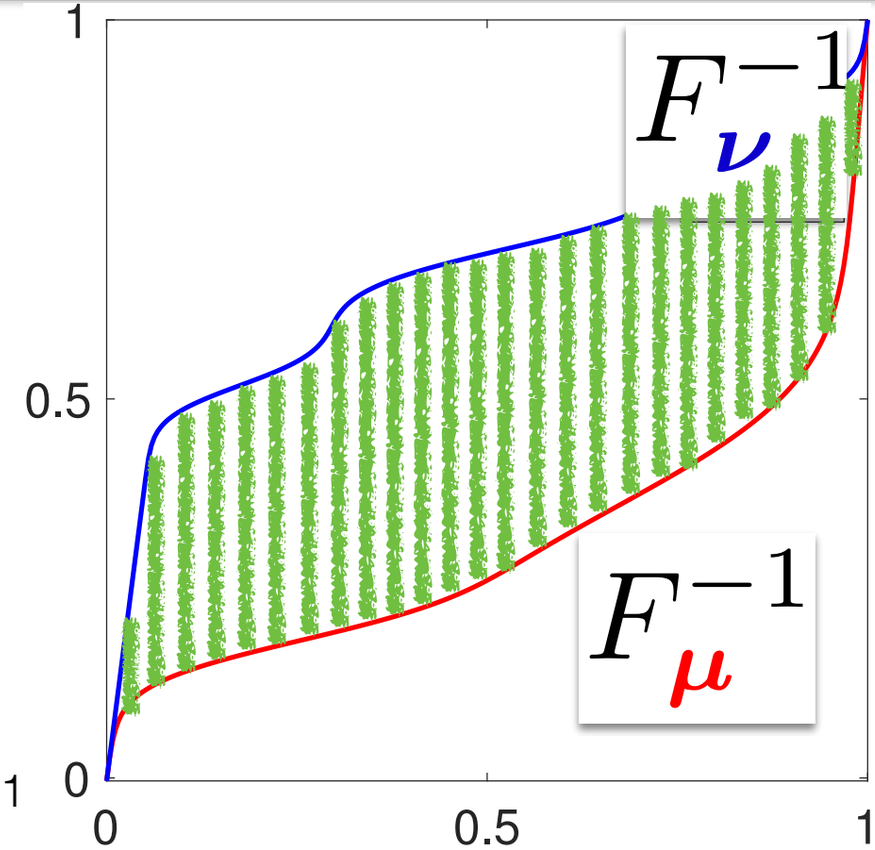


A Different Route: Projection

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

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





A simple baseline

- Dodges the high-dimensionality curse, by simplifying considerably the measures.
- Effective in practice, fast and easy, but far from OT. Induced matchings are extremely blurry.

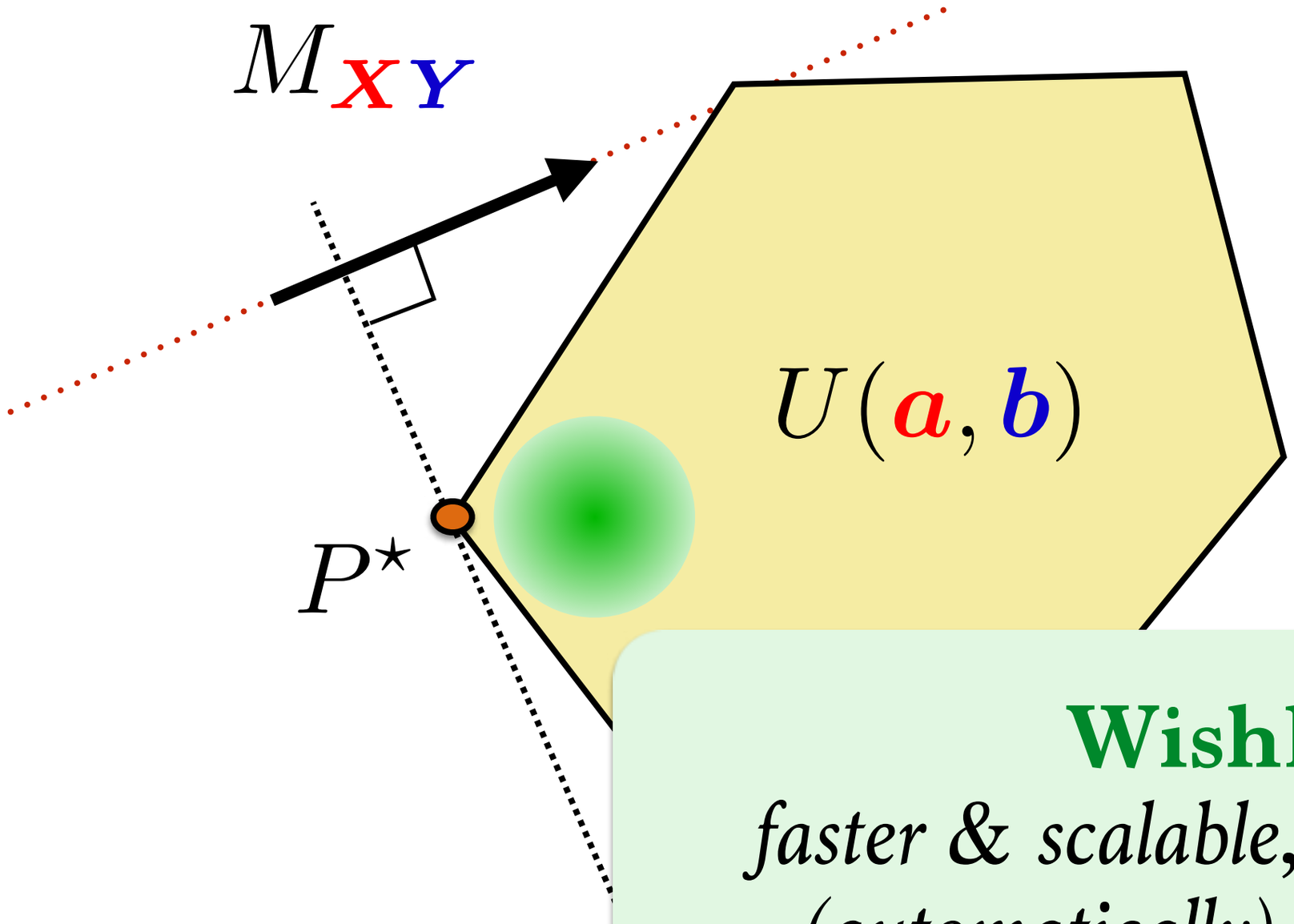
A simple baseline

Sliced Wasserstein Distance [Rabin+'11]

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

- Dodges the high-dimensionality curse, by simplifying considerably the measures.
- Effective in practice, fast and easy, but far from OT. Induced matchings are extremely blurry.

Regularization on the Primal



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

Entropic Regularization [Wilson'62]

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

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

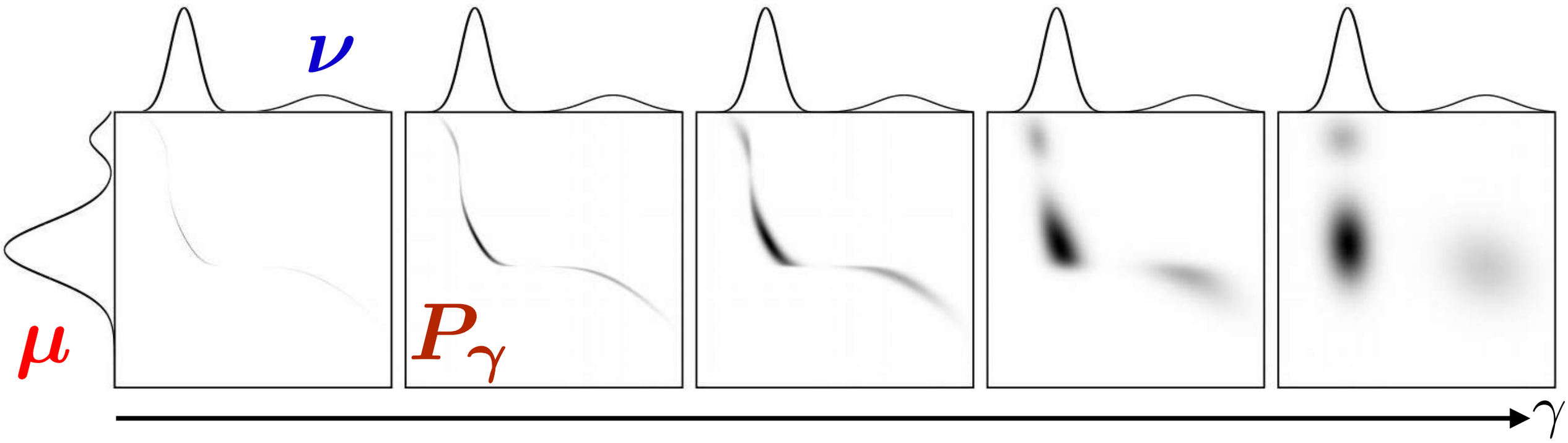
$$E(P) \stackrel{\text{def}}{=} -\sum_{i,j=1}^{nm} P_{ij}(\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

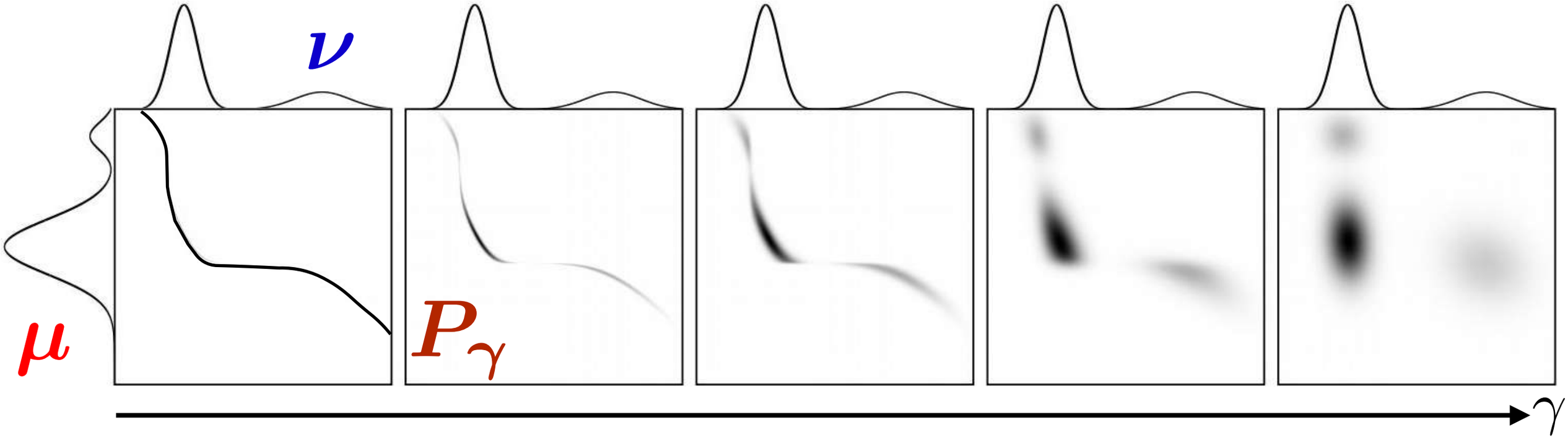


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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

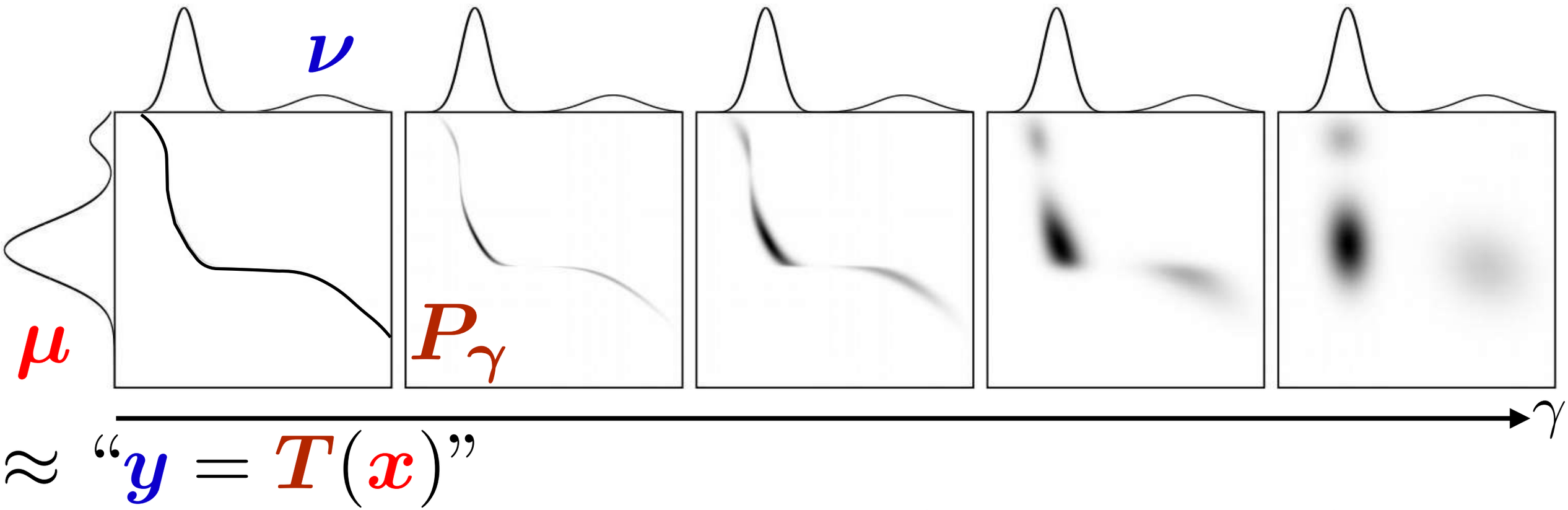


Note: Unique optimal solution because of strong concavity of entropy

Entropic Regularization [Wilson'62]

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

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



$\approx$ "$\boldsymbol{y} = \boldsymbol{T}(\boldsymbol{x})$"

Note: Unique optimal solution because of strong concavity of entropy

Fast & Scalable Algorithm

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

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

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

Fast & Scalable Algorithm

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

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

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

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

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

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

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

Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

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

Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

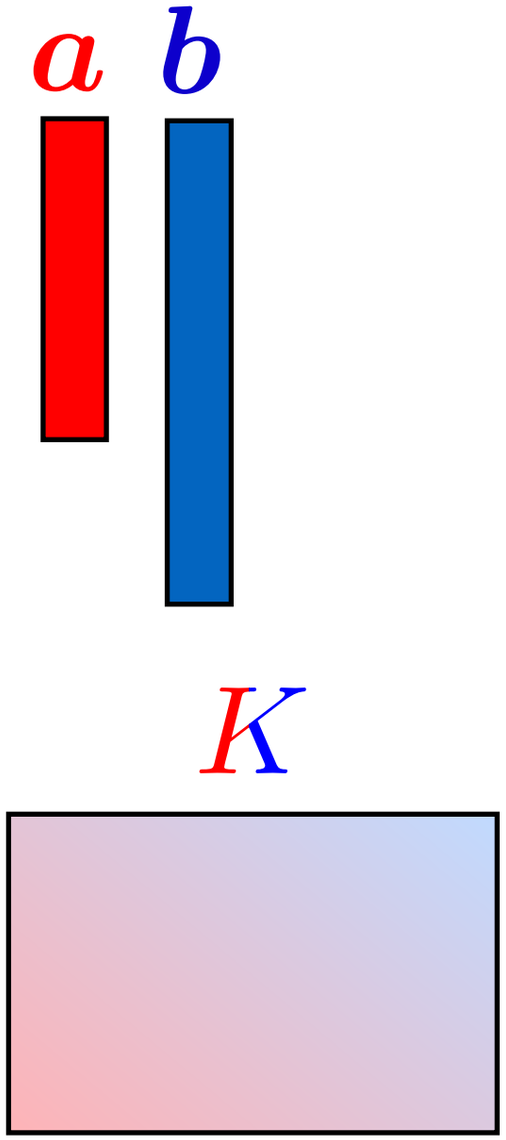
1. $\boldsymbol{u} = \boldsymbol{a} / K\boldsymbol{v}$
2. $\boldsymbol{v} = \boldsymbol{b} / K^T\boldsymbol{u}$

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

Fast & Scalable Algorithm

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

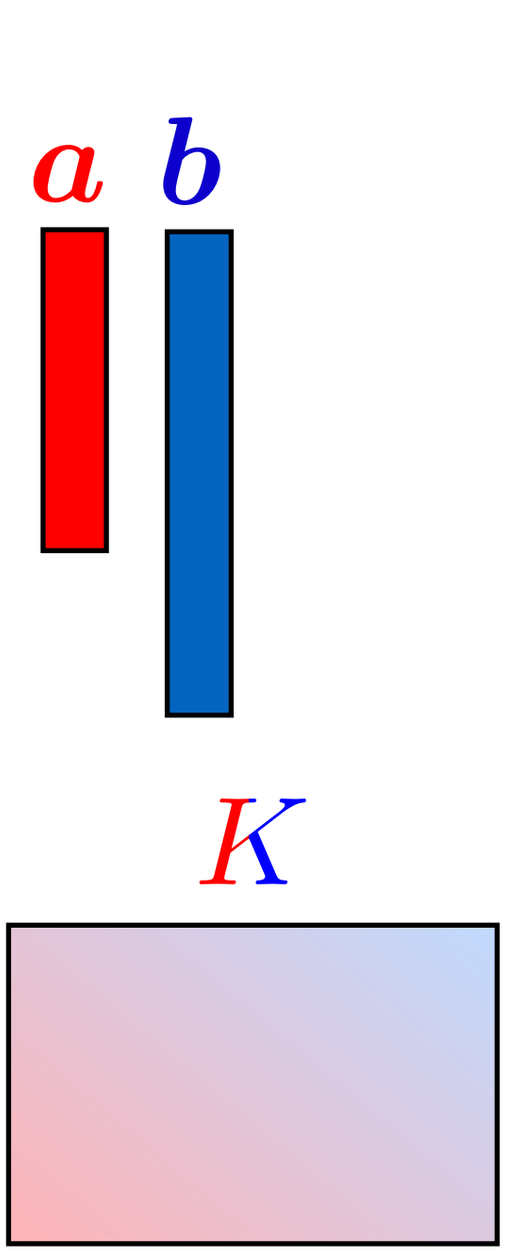
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$



Fast & Scalable Algorithm

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

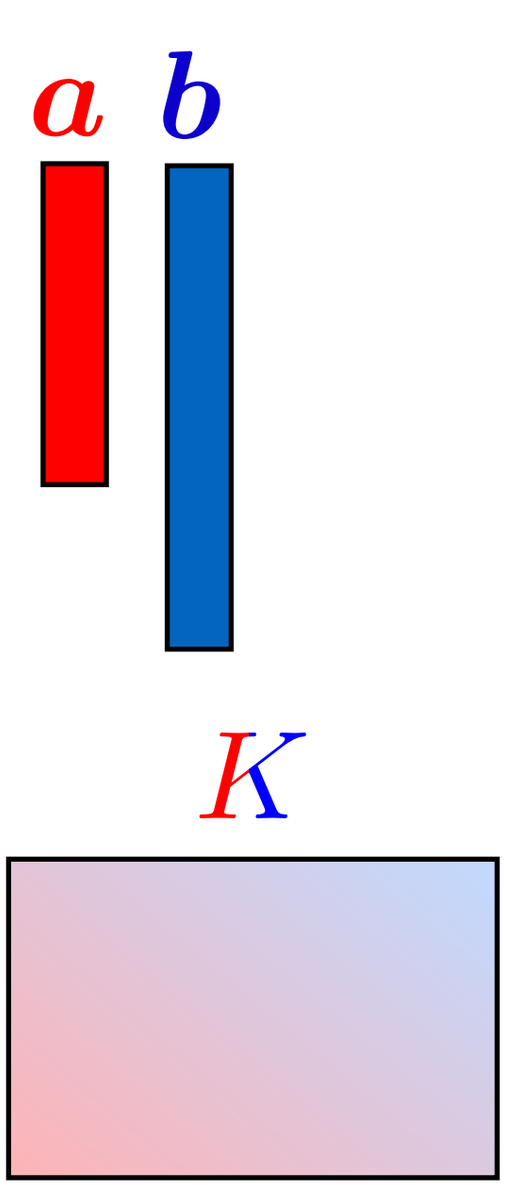
$$\boldsymbol{u} \leftarrow \boldsymbol{a}/K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b}/K^T\boldsymbol{u}$$

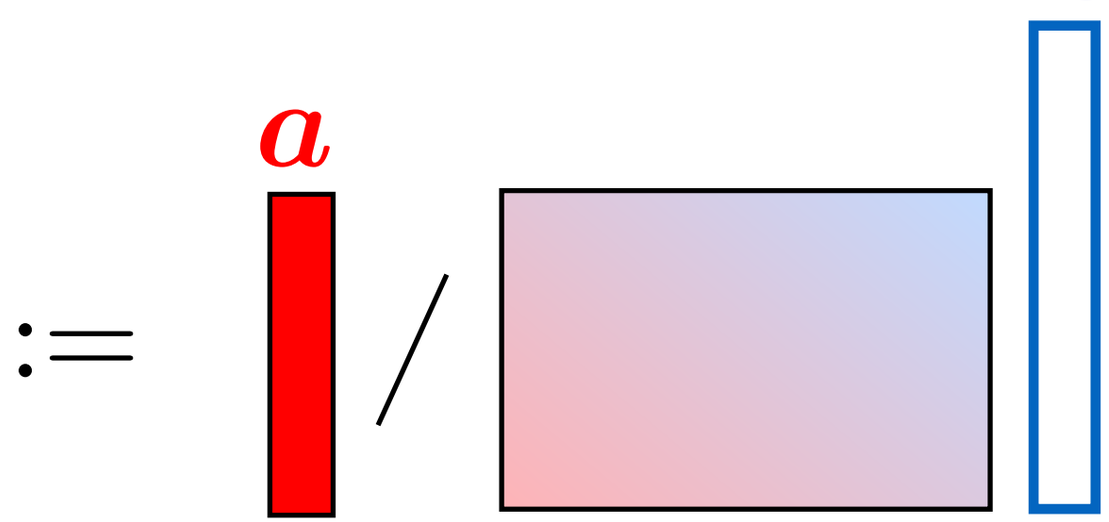


Fast & Scalable Algorithm

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

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

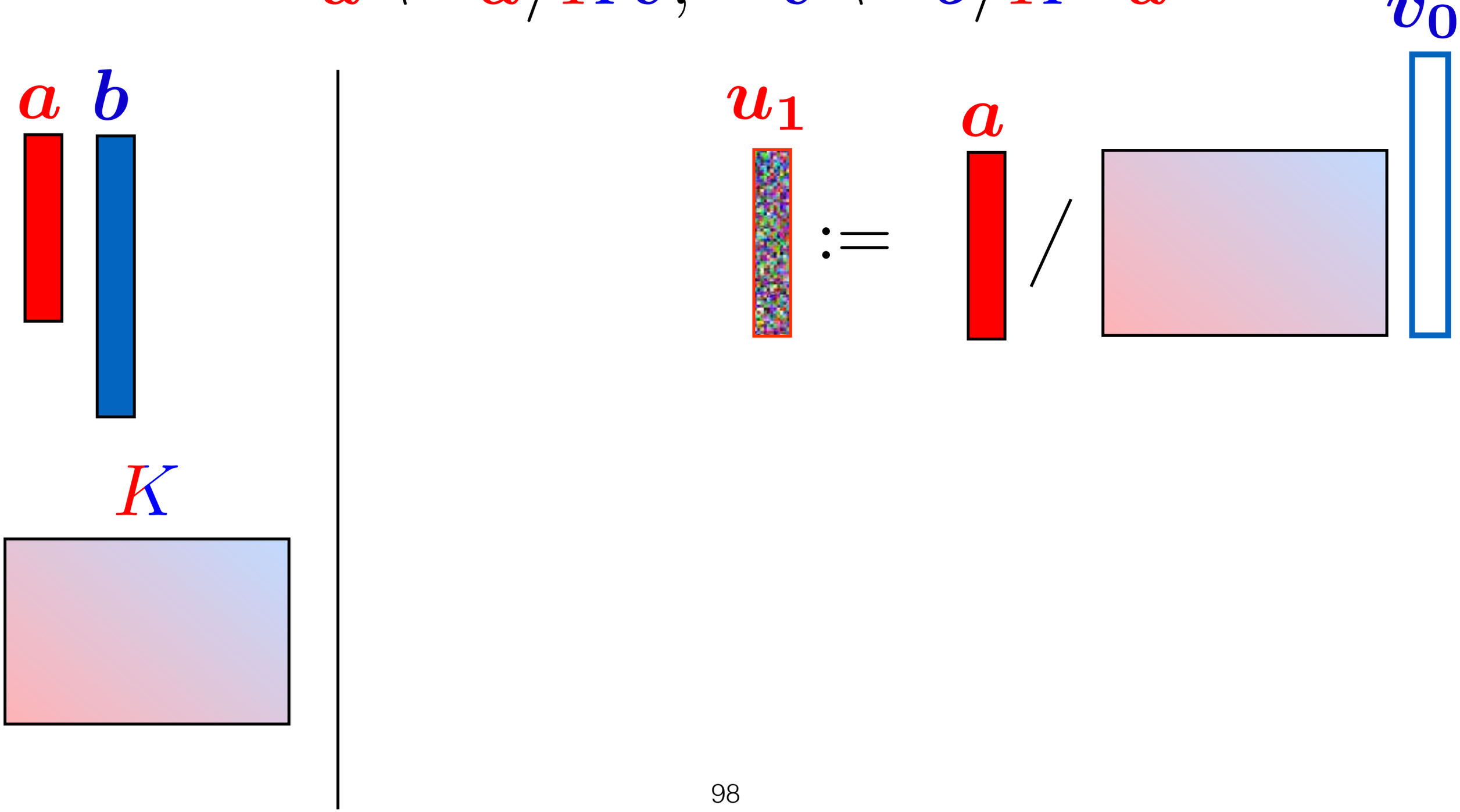




Fast & Scalable Algorithm

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

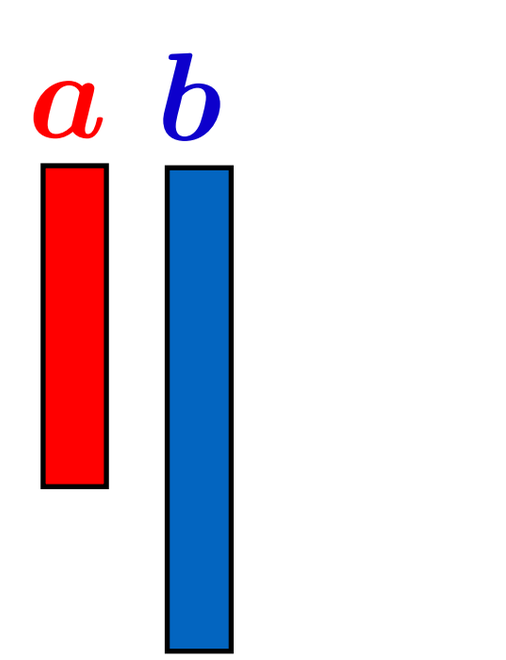
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$

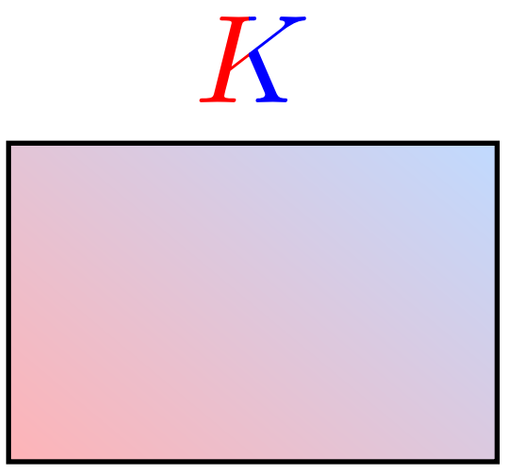


Fast & Scalable Algorithm

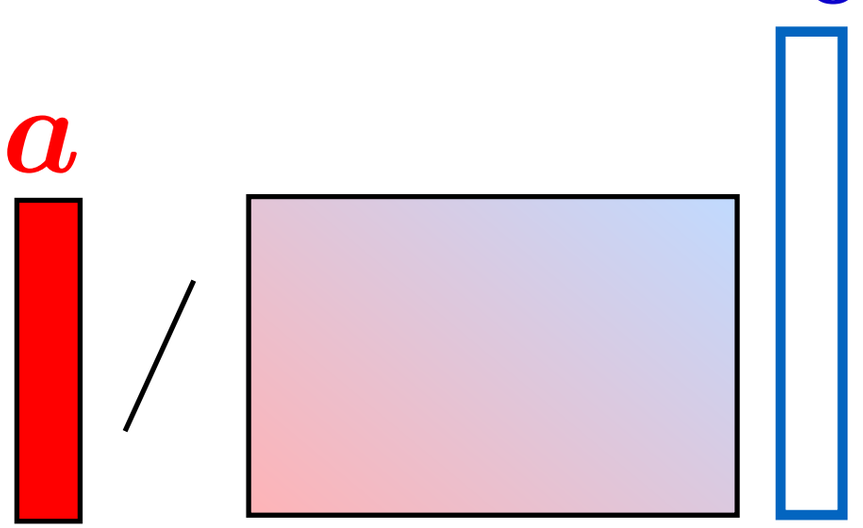
- **[Sinkhorn'64]** fixed-point iterations for $(\boldsymbol{u}, \boldsymbol{v})$

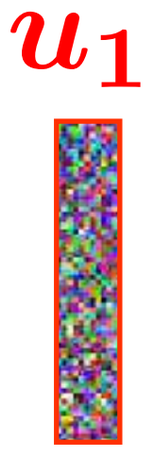
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$





$:=$

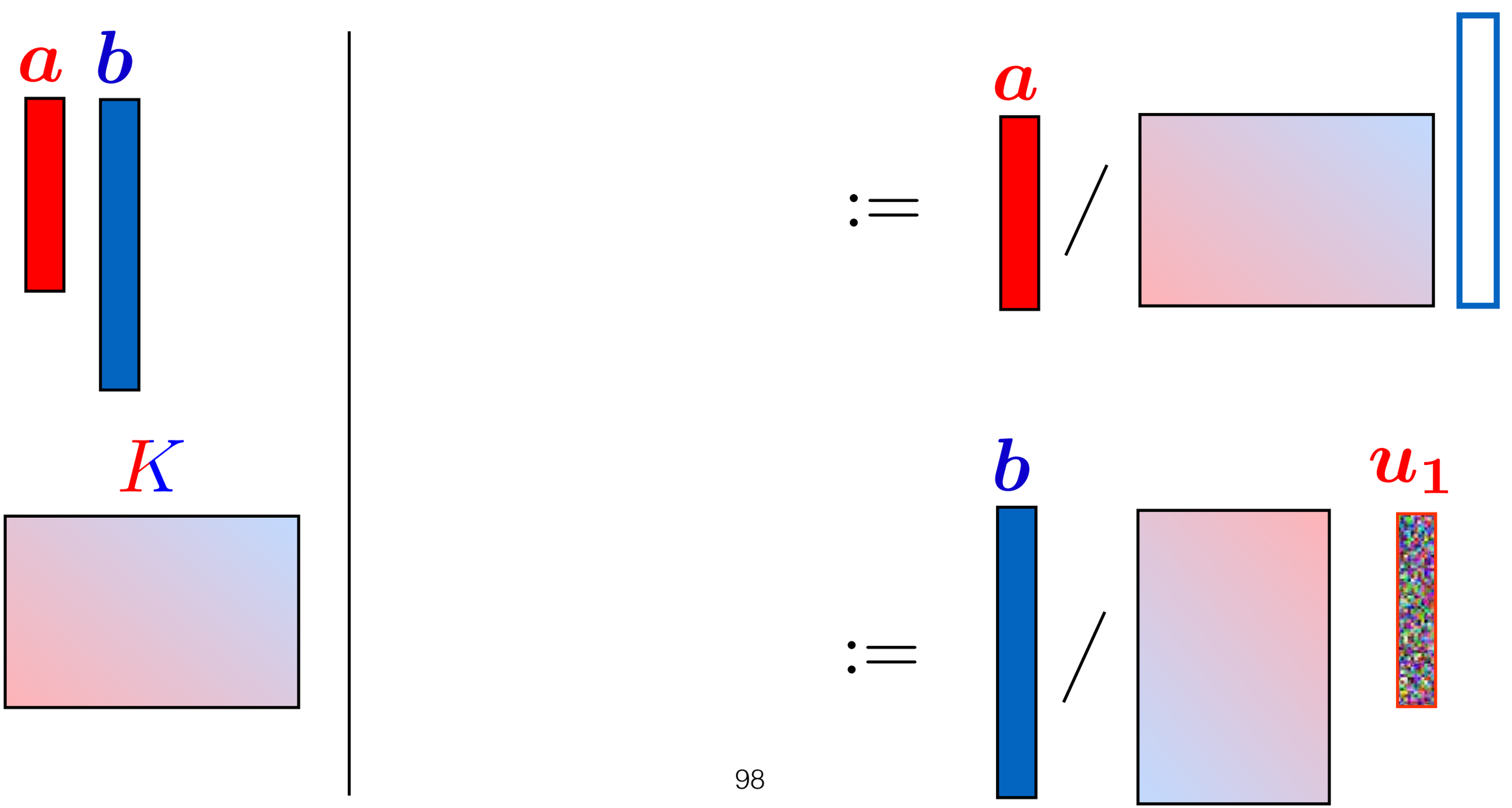




Fast & Scalable Algorithm

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

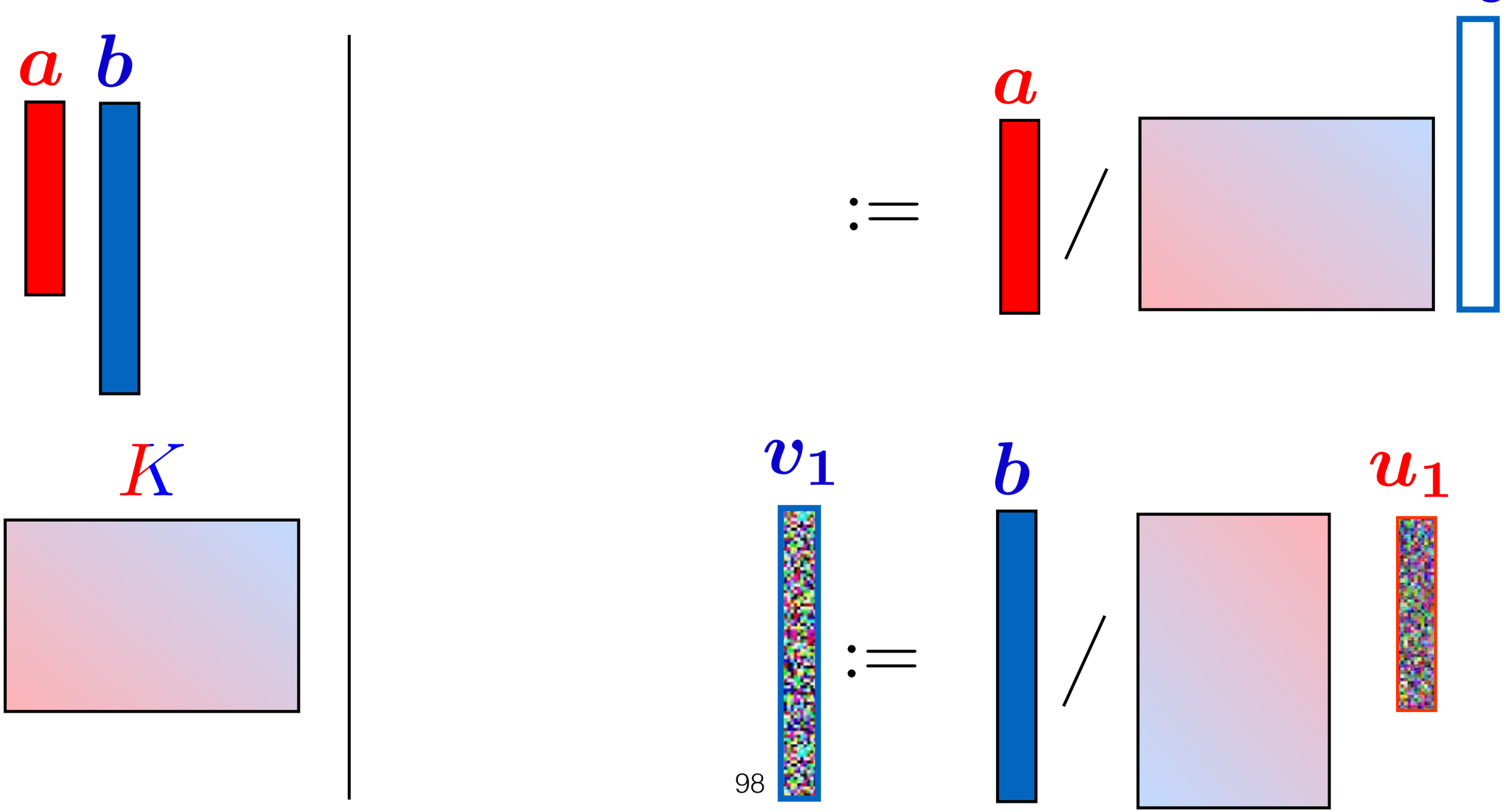
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

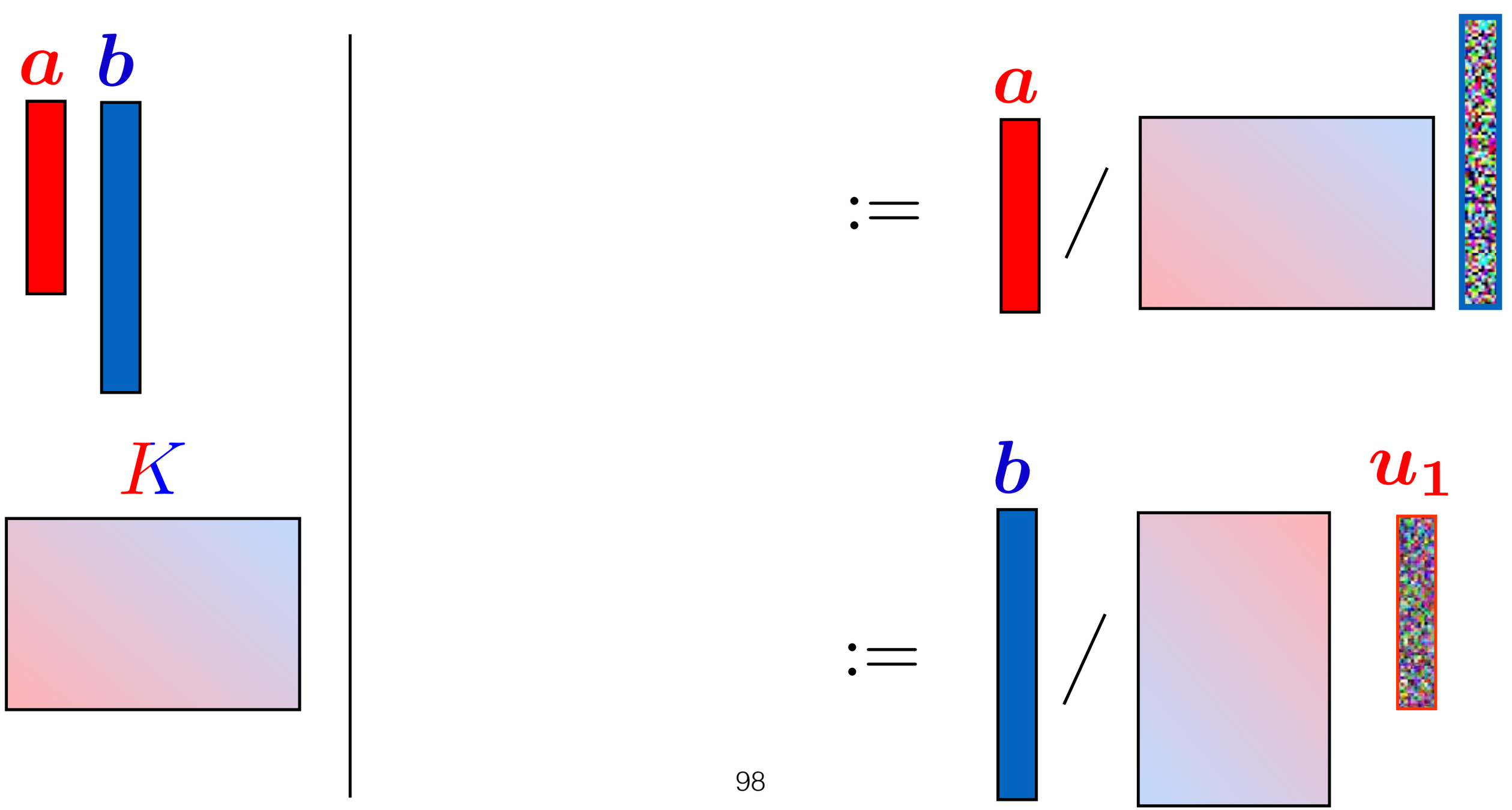
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

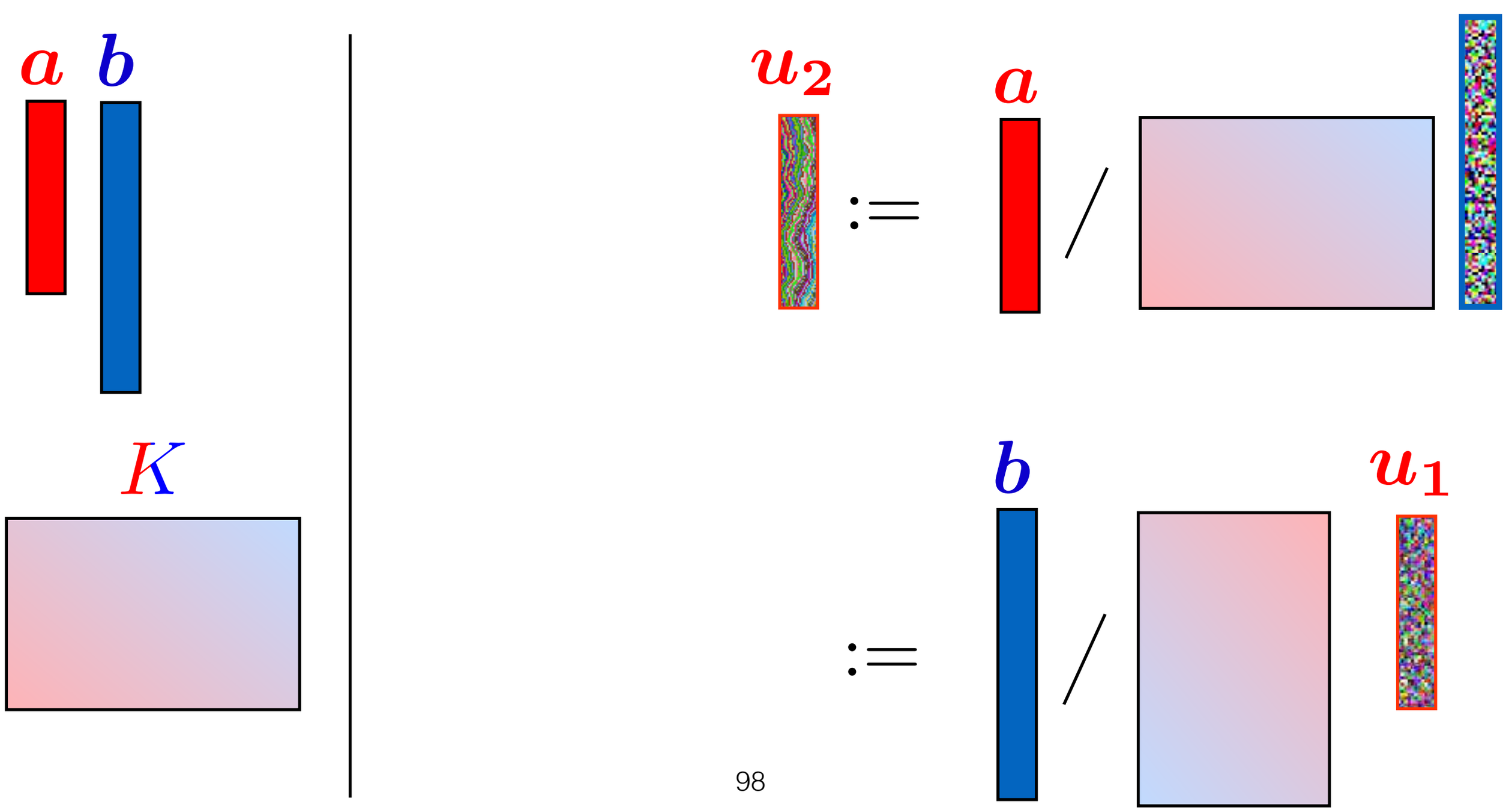
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

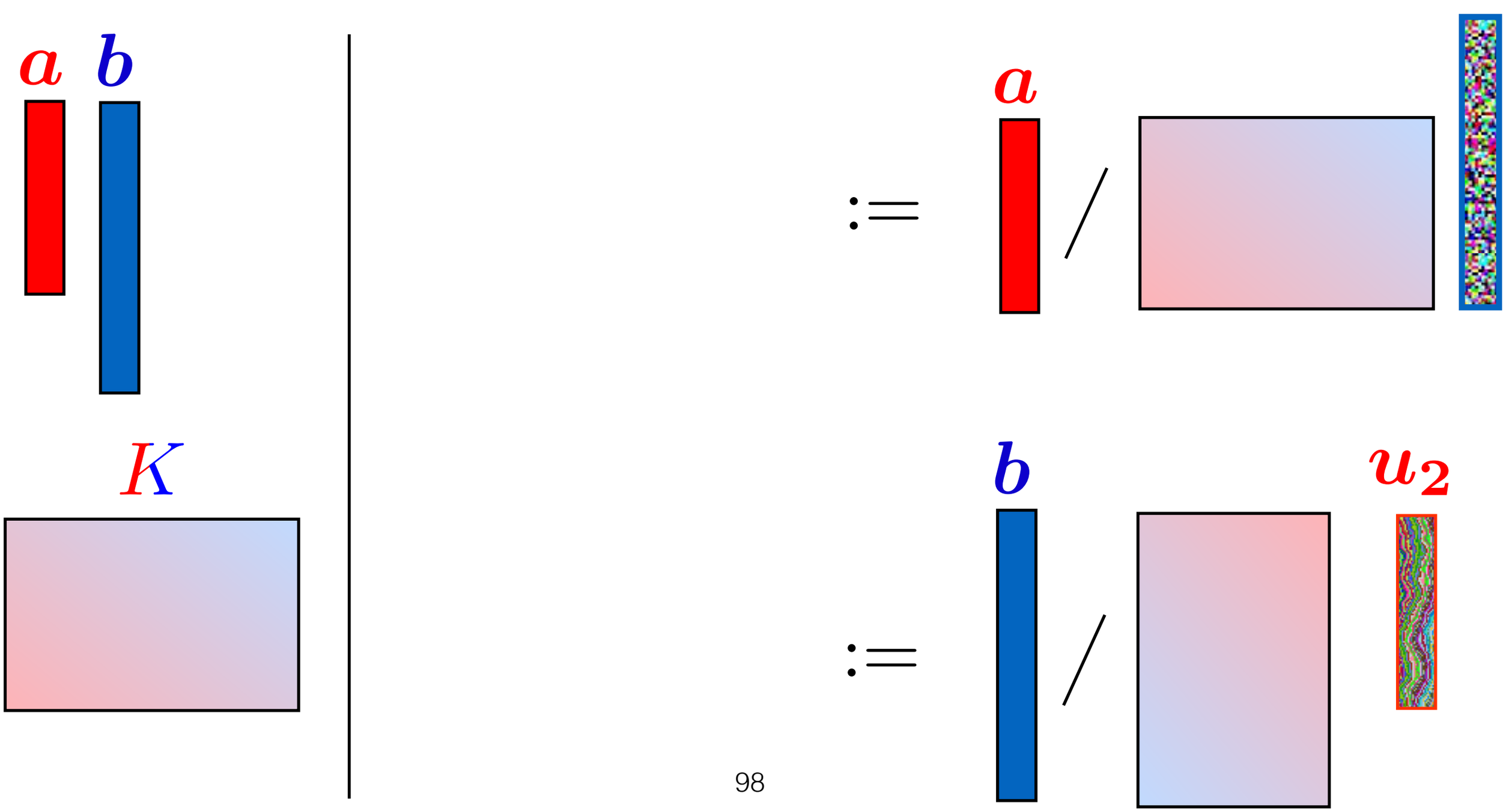
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

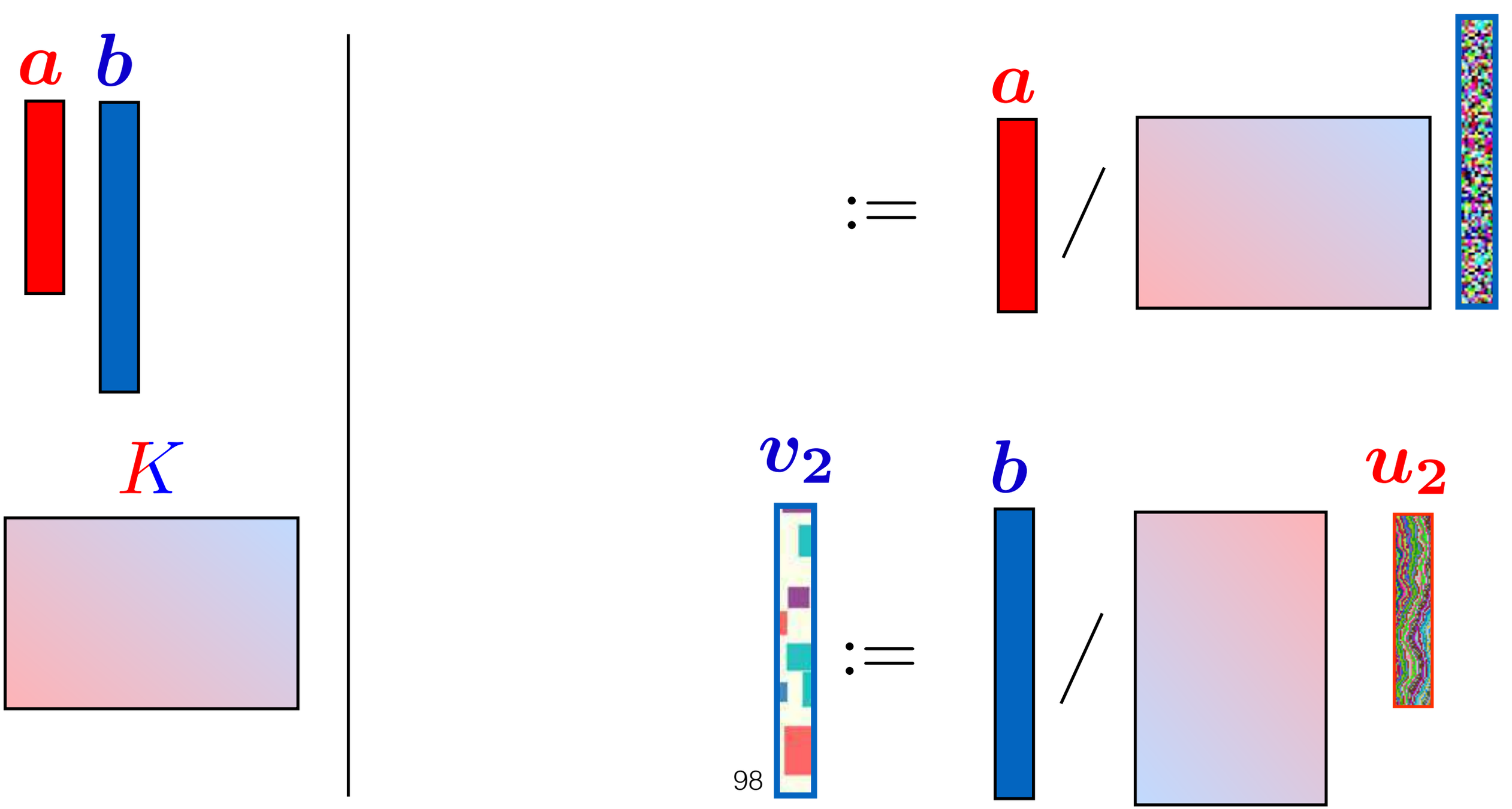
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

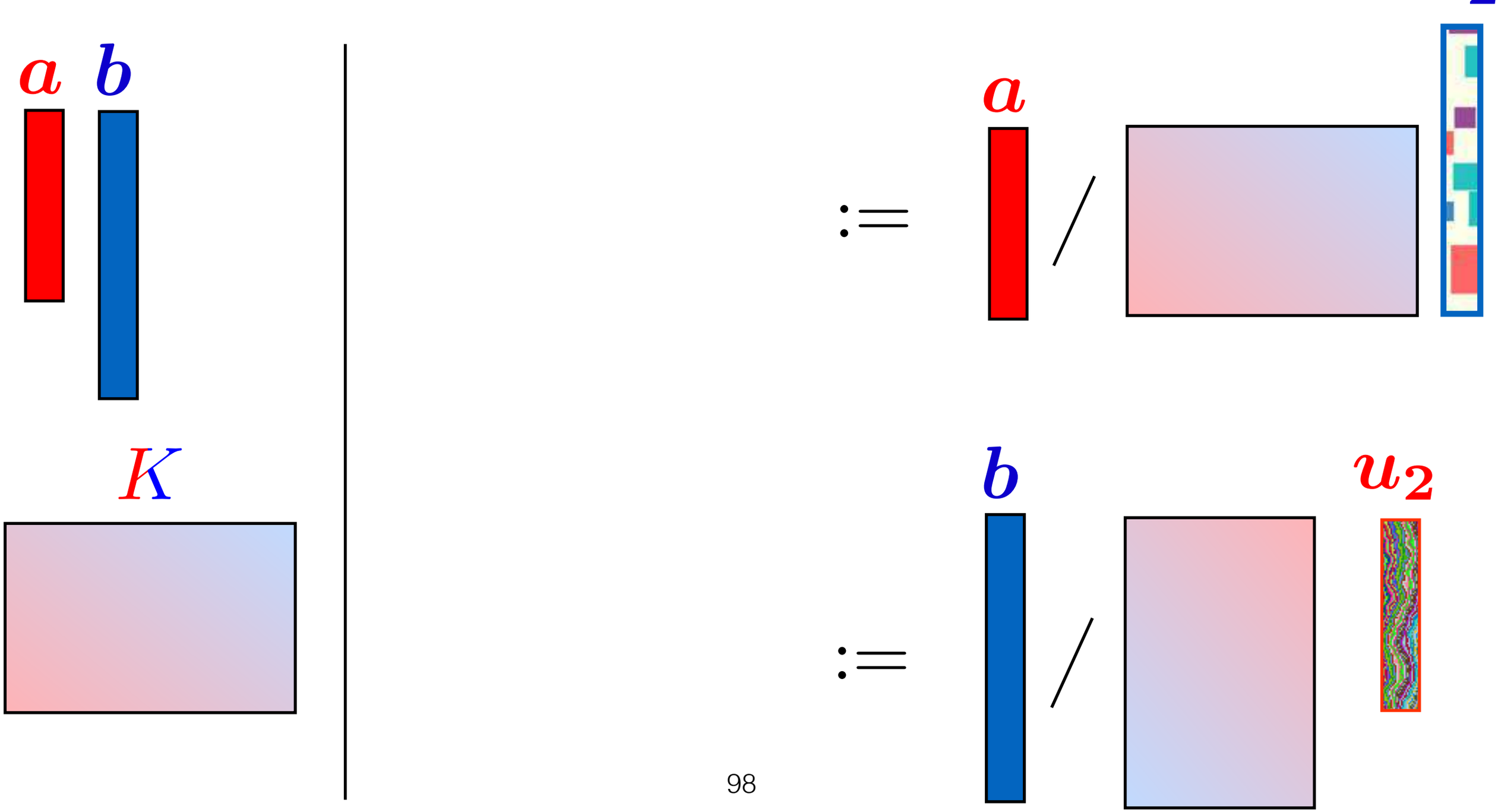
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

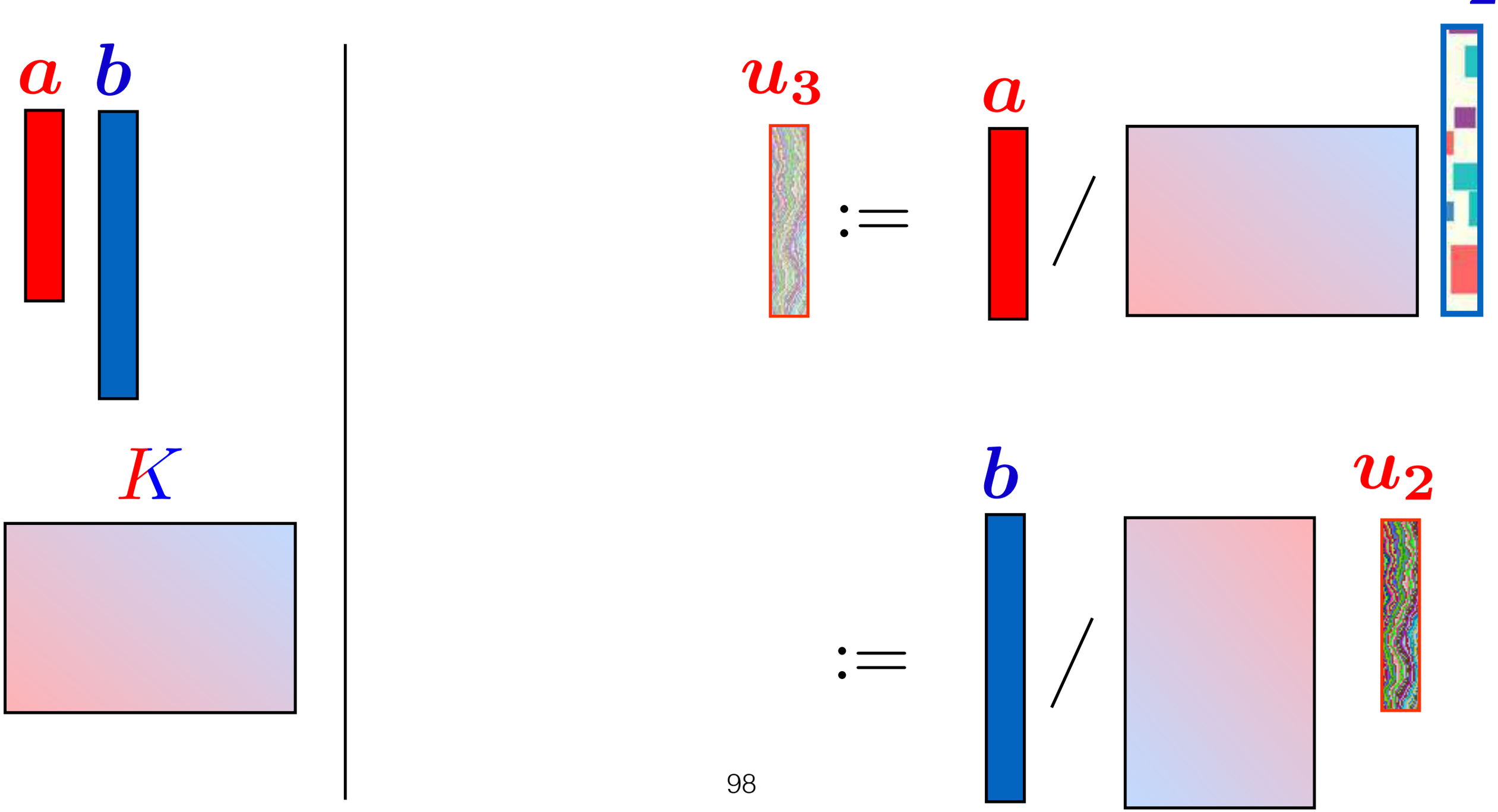
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

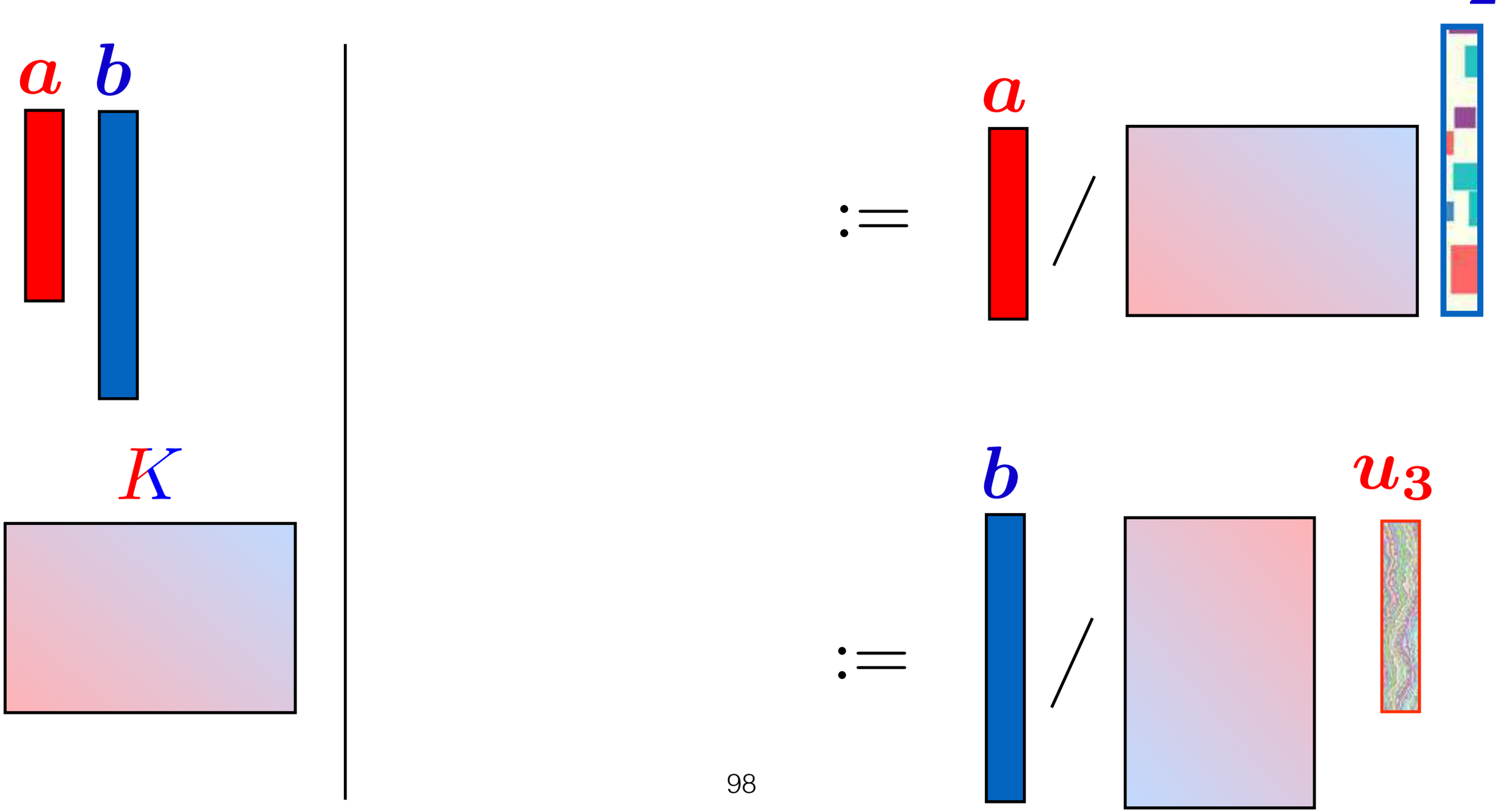
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$



Fast & Scalable Algorithm

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

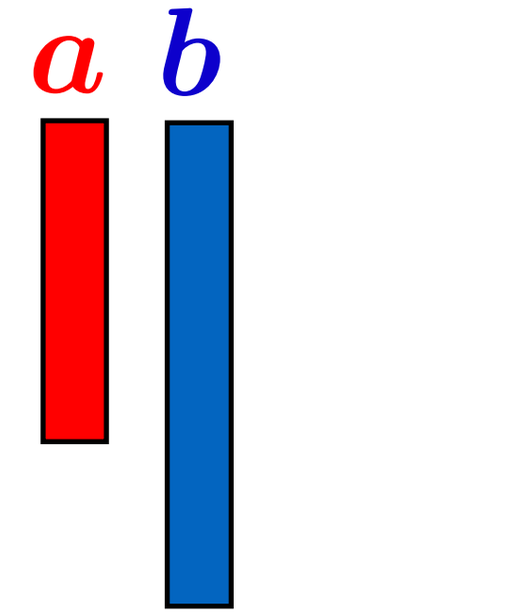
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$

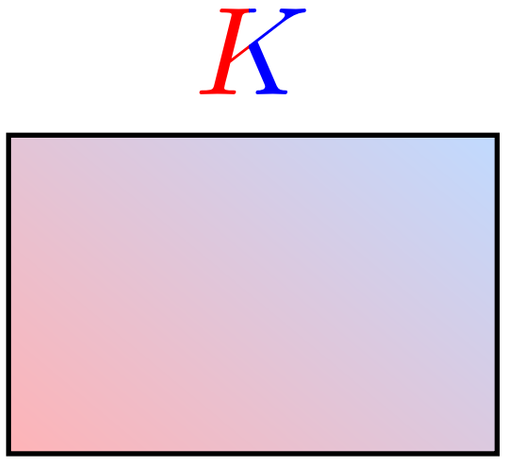


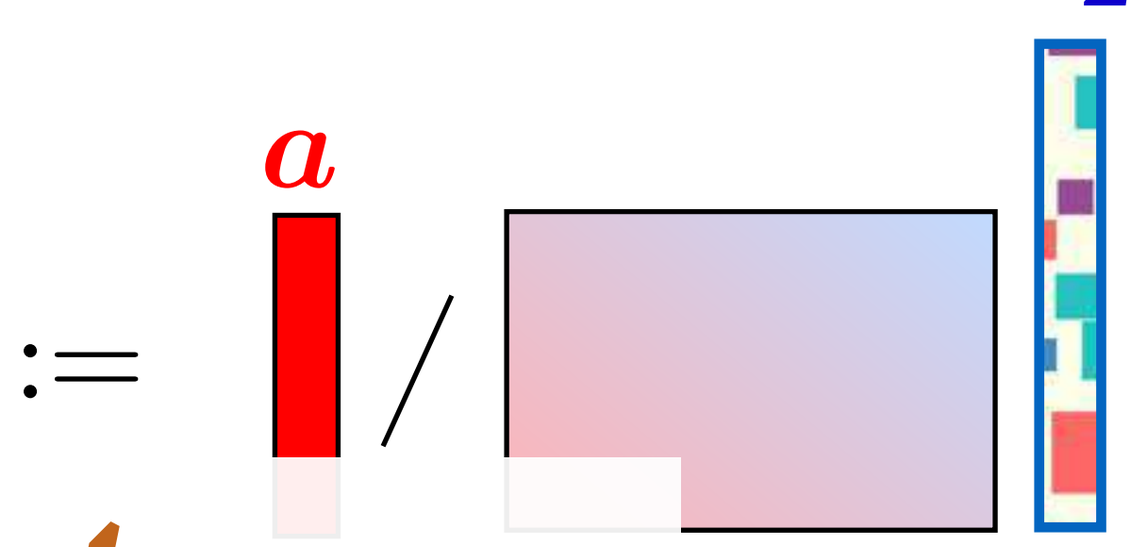
Fast & Scalable Algorithm

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

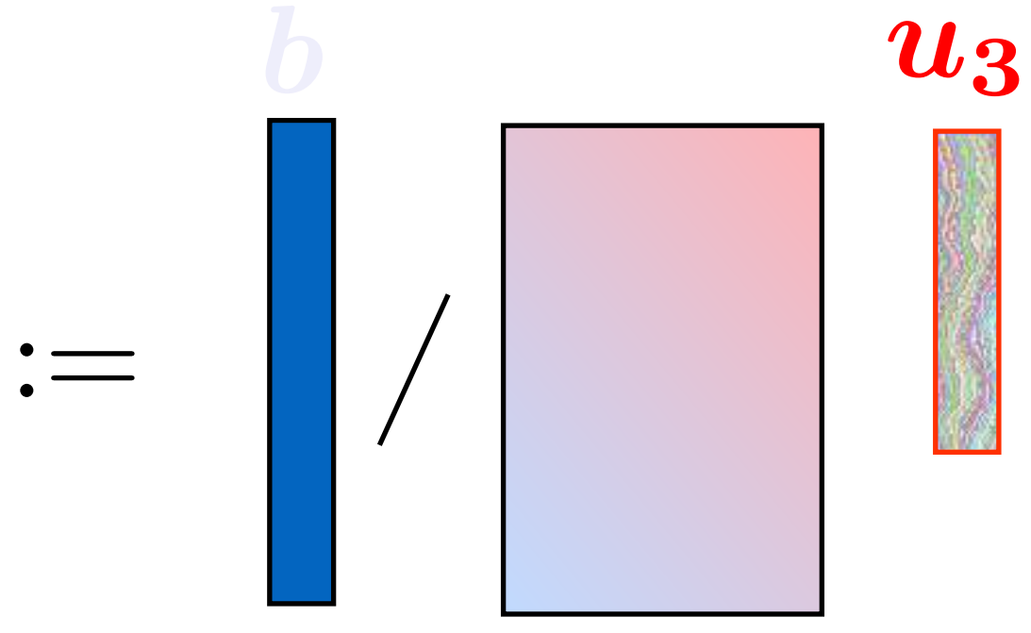
$$\boldsymbol{u} \leftarrow \boldsymbol{a} / K\boldsymbol{v}, \quad \boldsymbol{v} \leftarrow \boldsymbol{b} / K^T \boldsymbol{u}$$





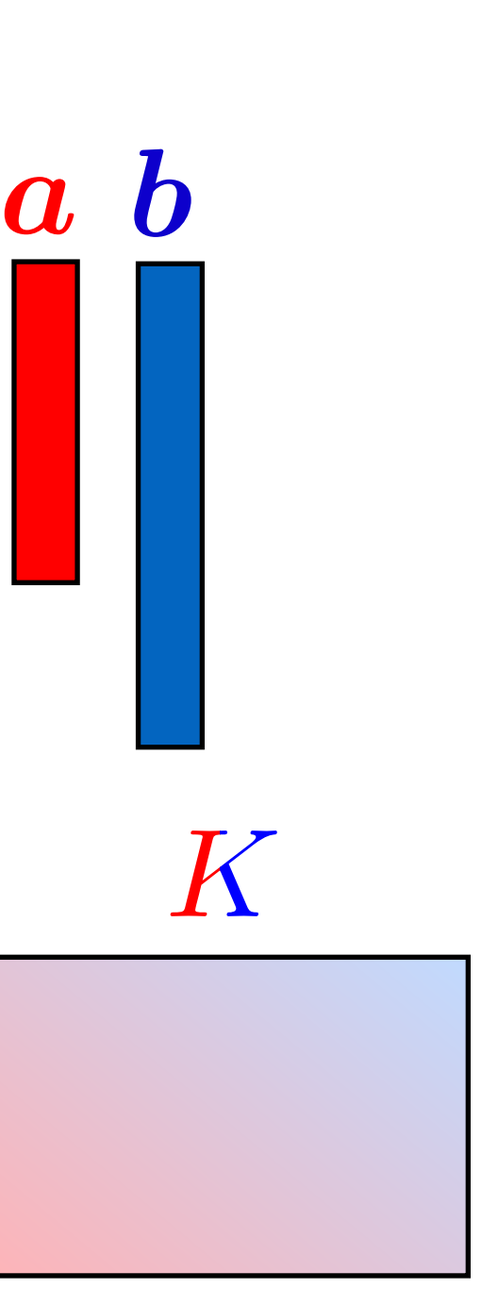


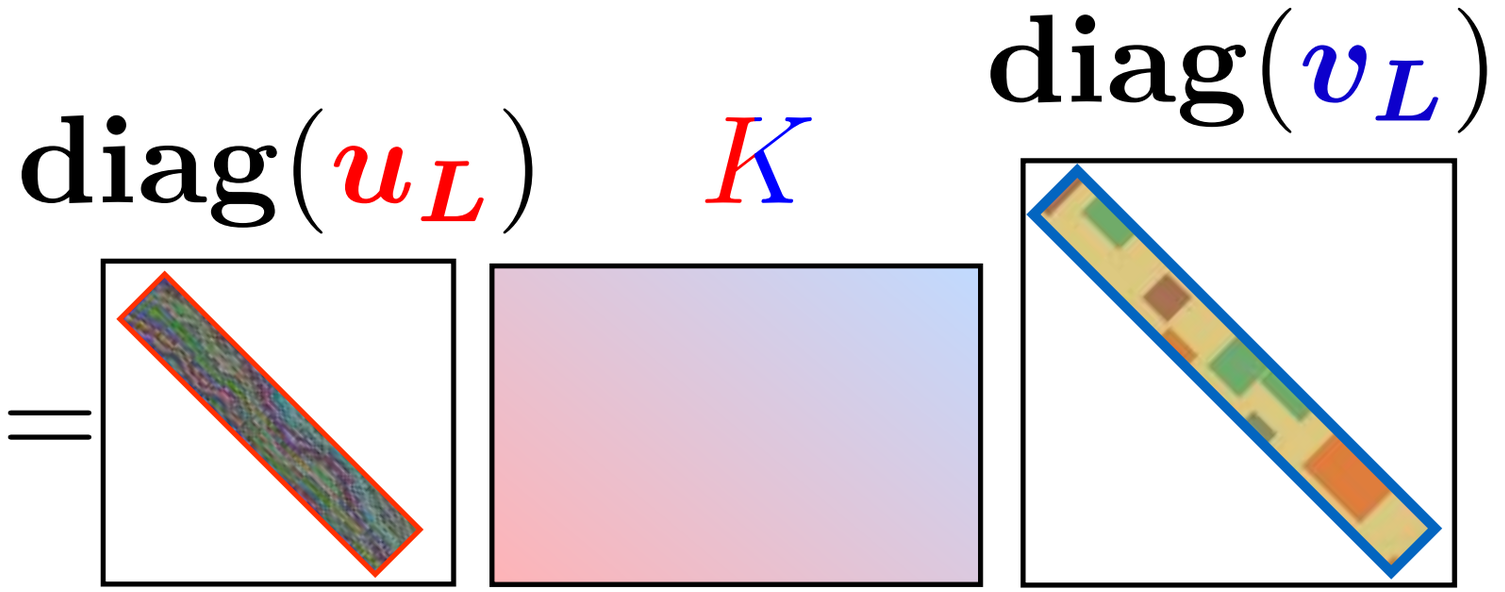
etc....



Fast & Scalable Algorithm

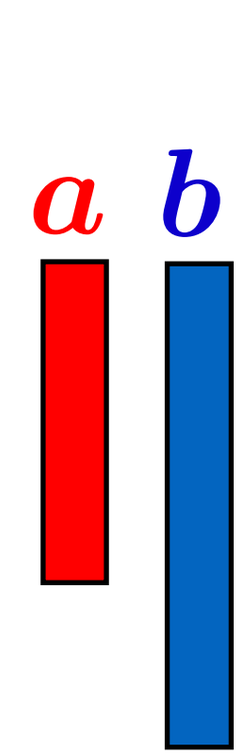
- **[Sinkhorn'64]** fixed-point iterations.

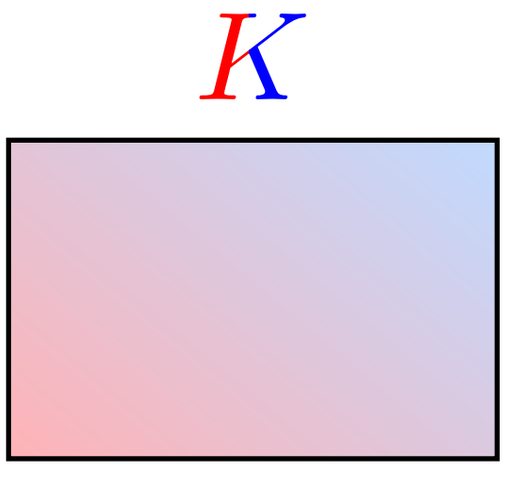




Fast & Scalable Algorithm

- **[Sinkhorn'64]** fixed-point iterations.



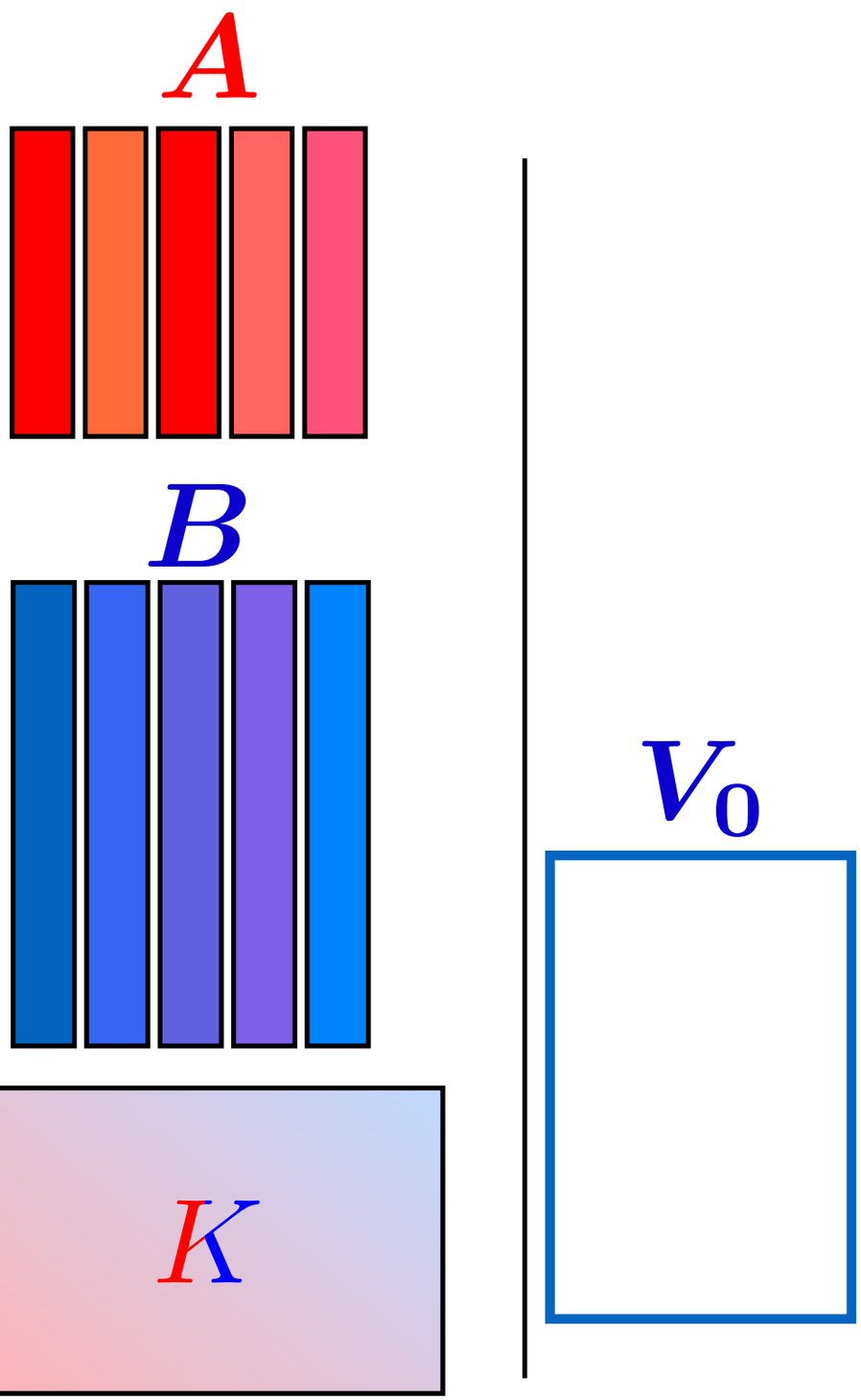


$$P_L := \mathrm{diag}(u_L)\, K\, \mathrm{diag}(v_L)$$

$$\langle P_L, M_{XY} \rangle = u_L{}^T (K \odot M_{XY}) v_L$$

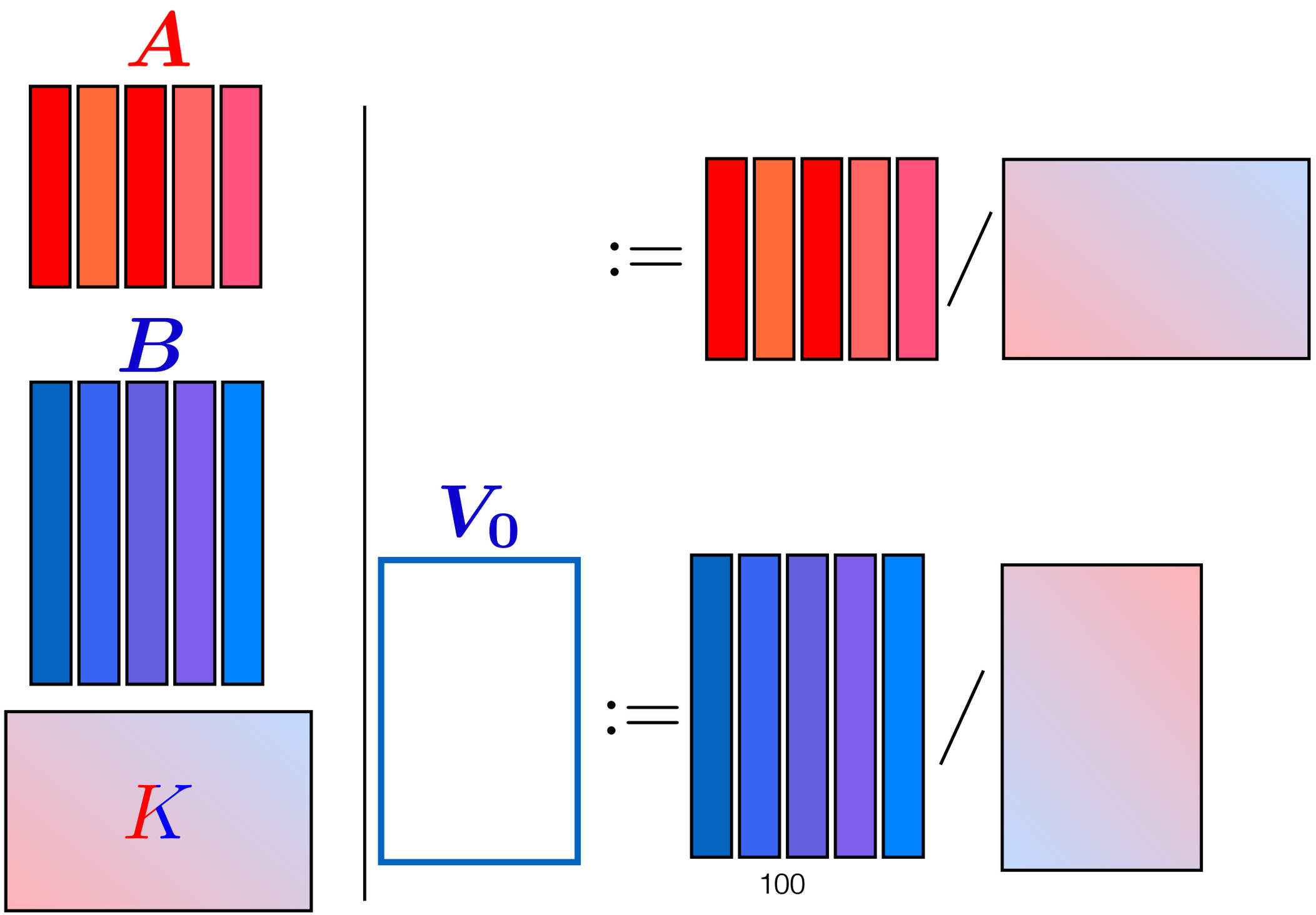
Also embarrassingly parallel

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



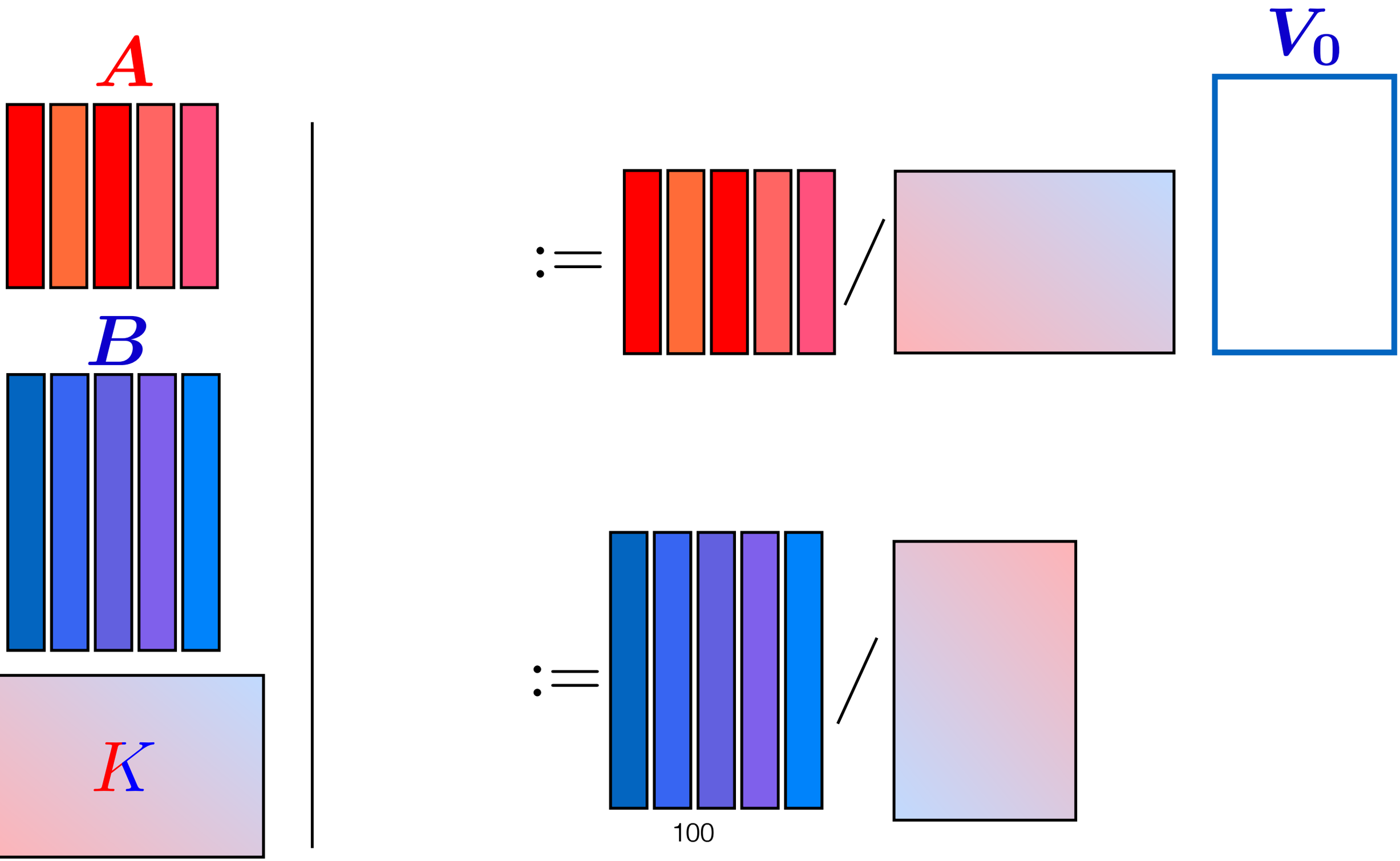
Also embarrassingly parallel

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



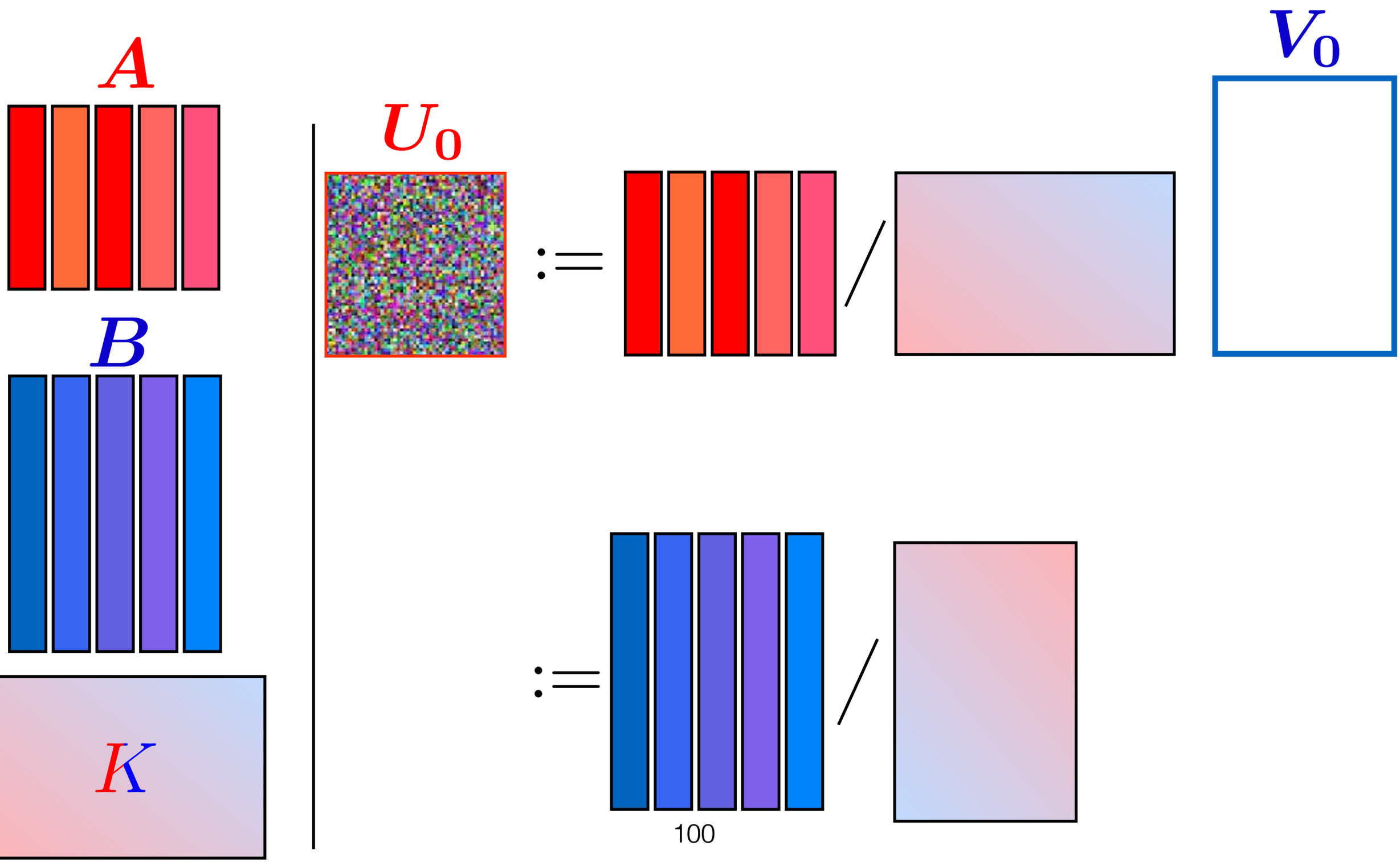
Also embarrassingly parallel

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



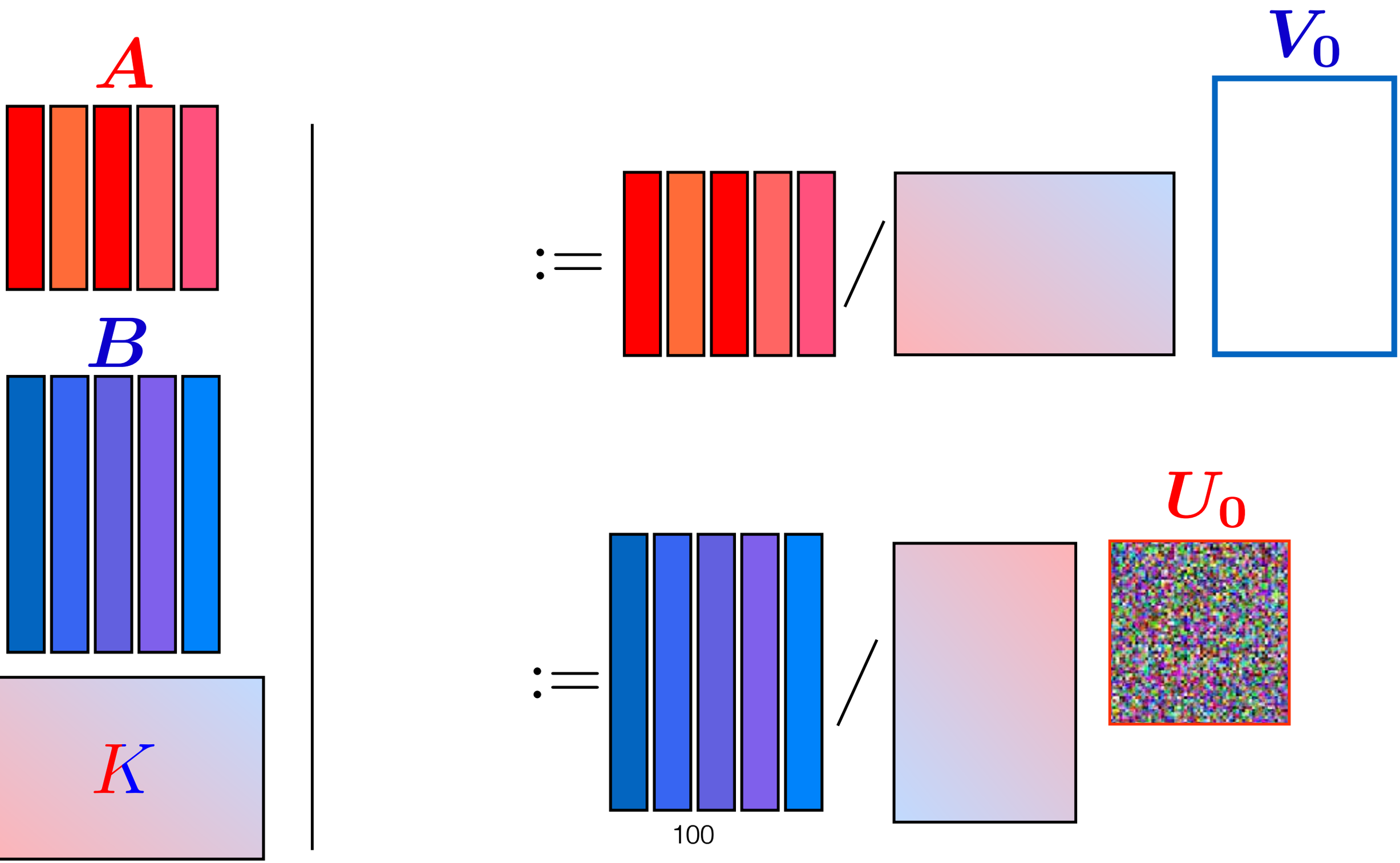
Also embarrassingly parallel

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



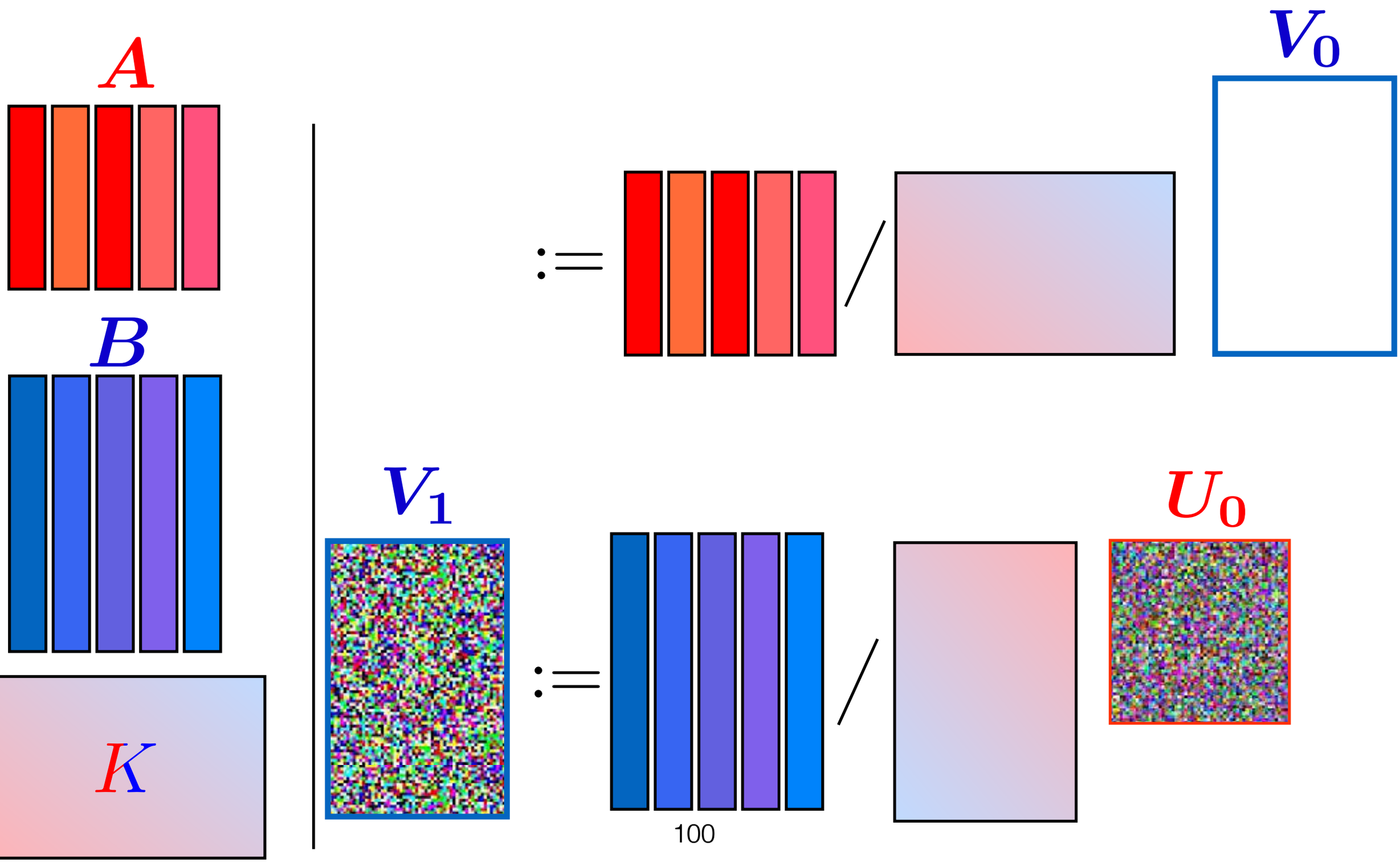
Also embarrassingly parallel

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



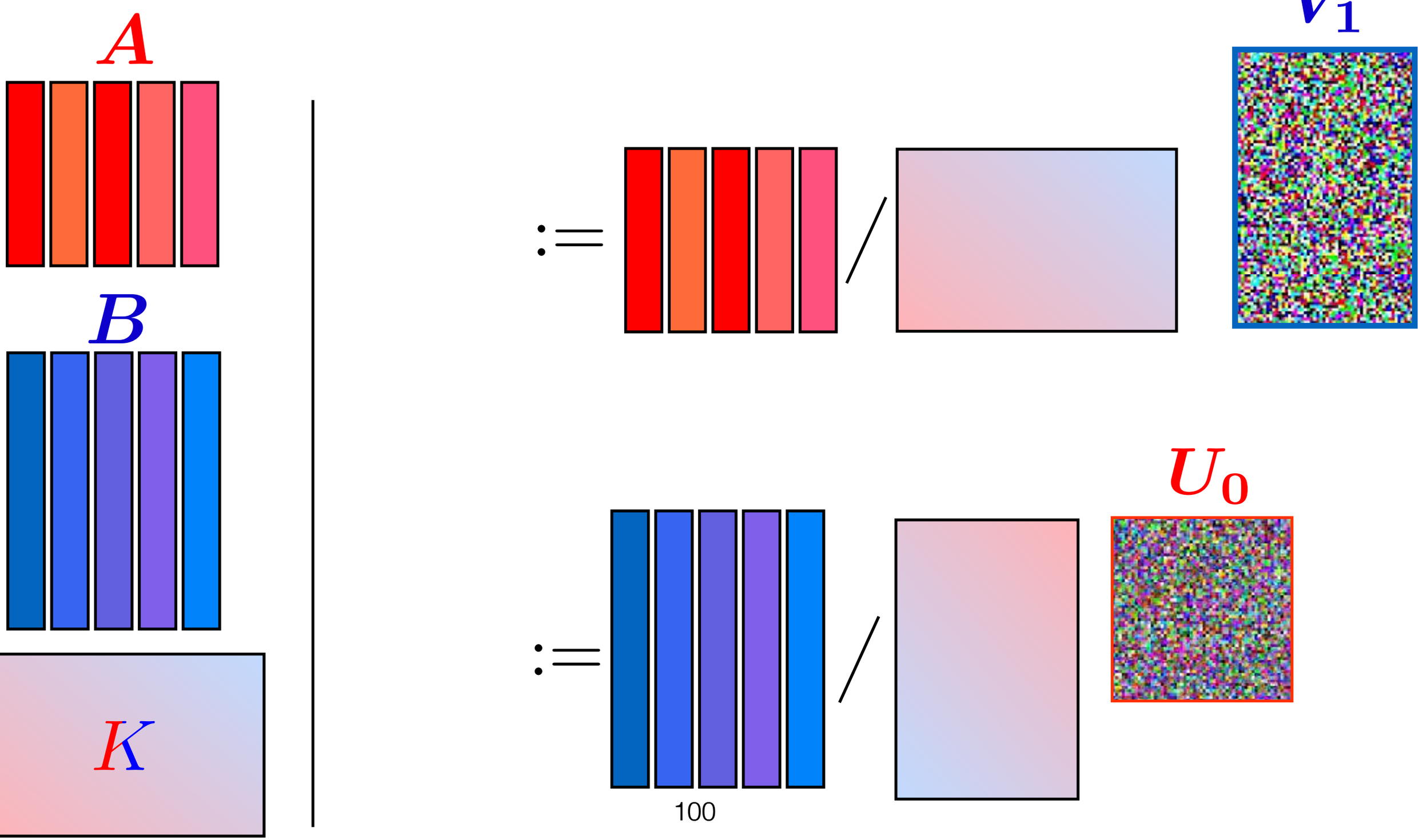
Also embarrassingly parallel

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



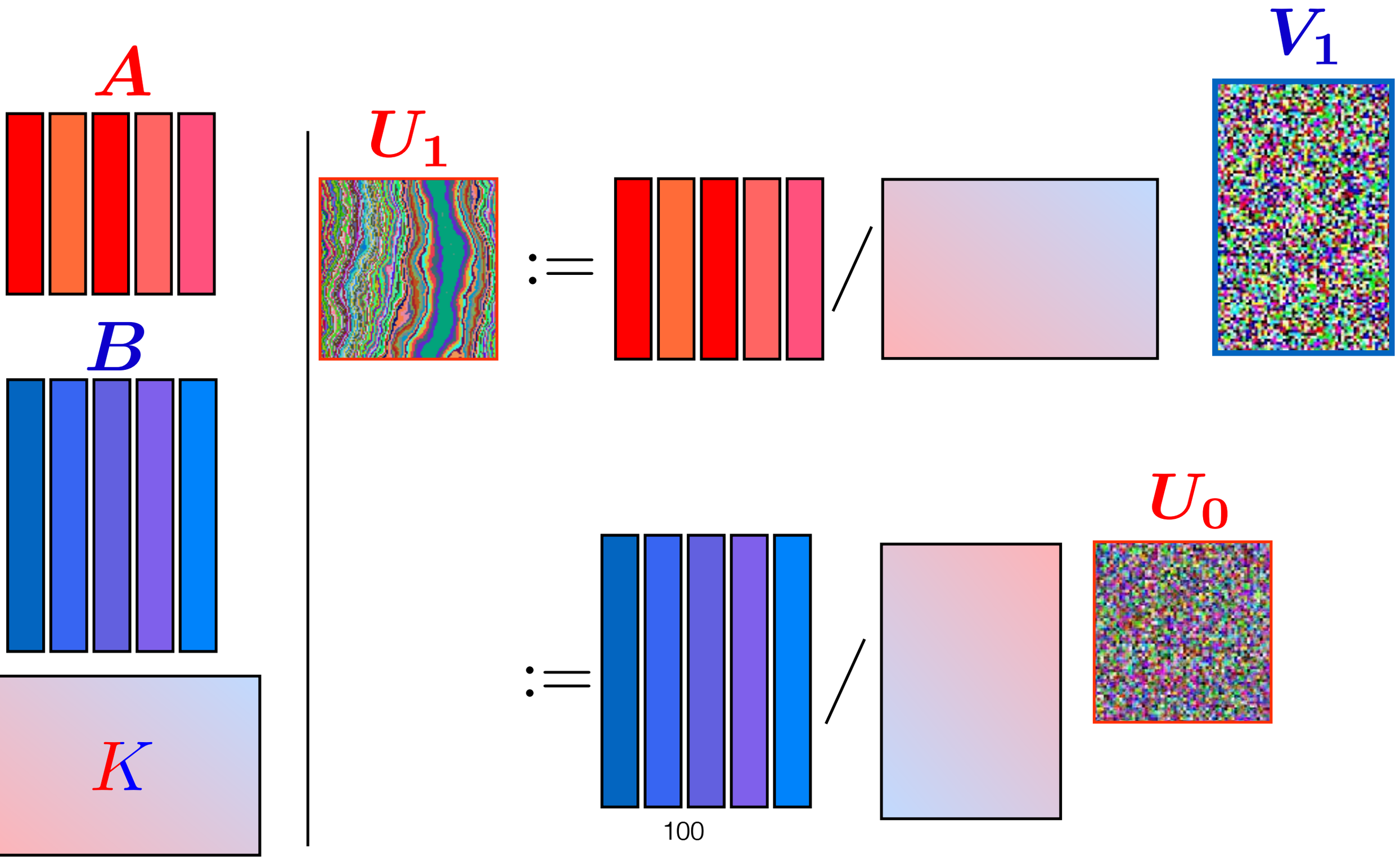
Also embarrassingly parallel

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



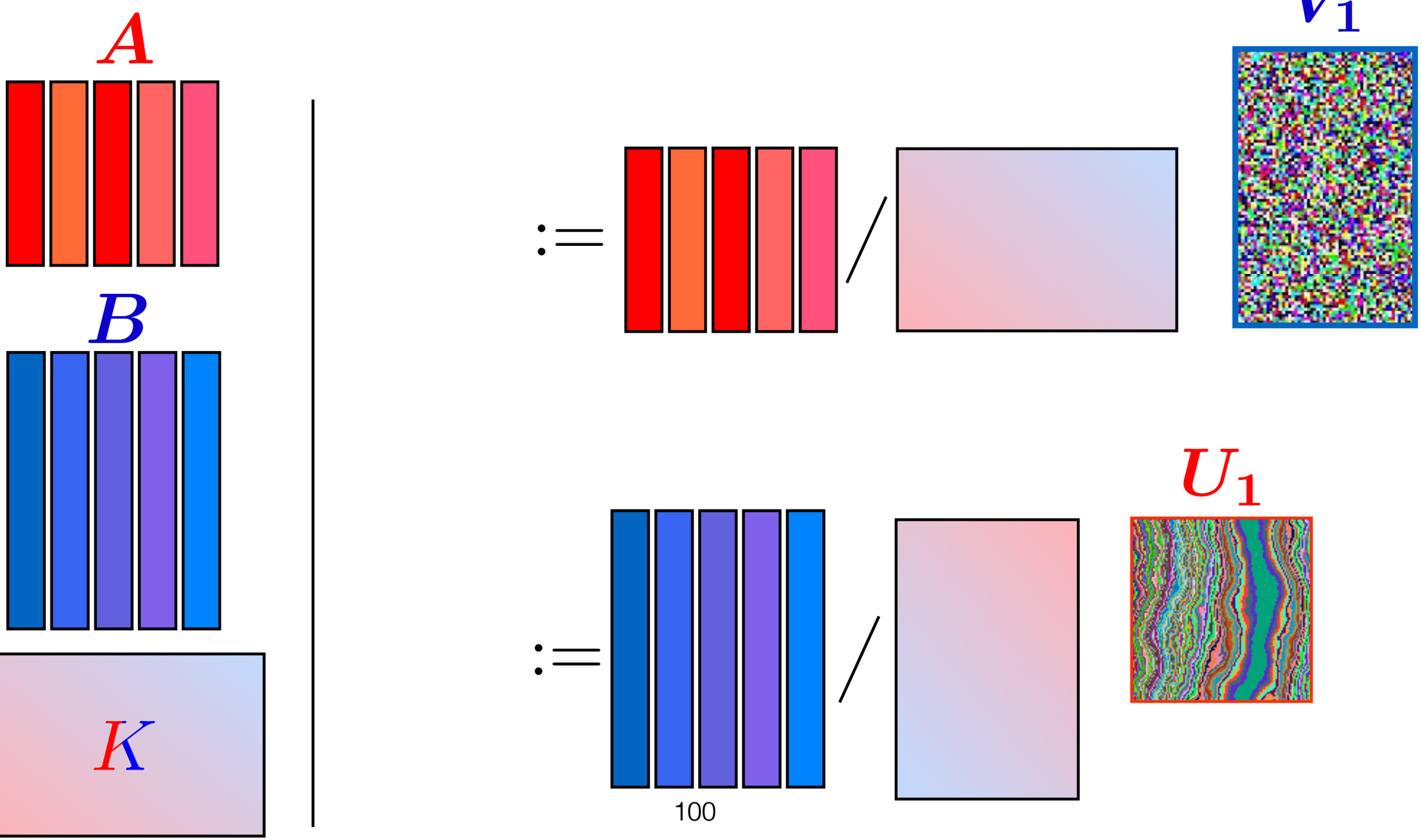
Also embarrassingly parallel

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



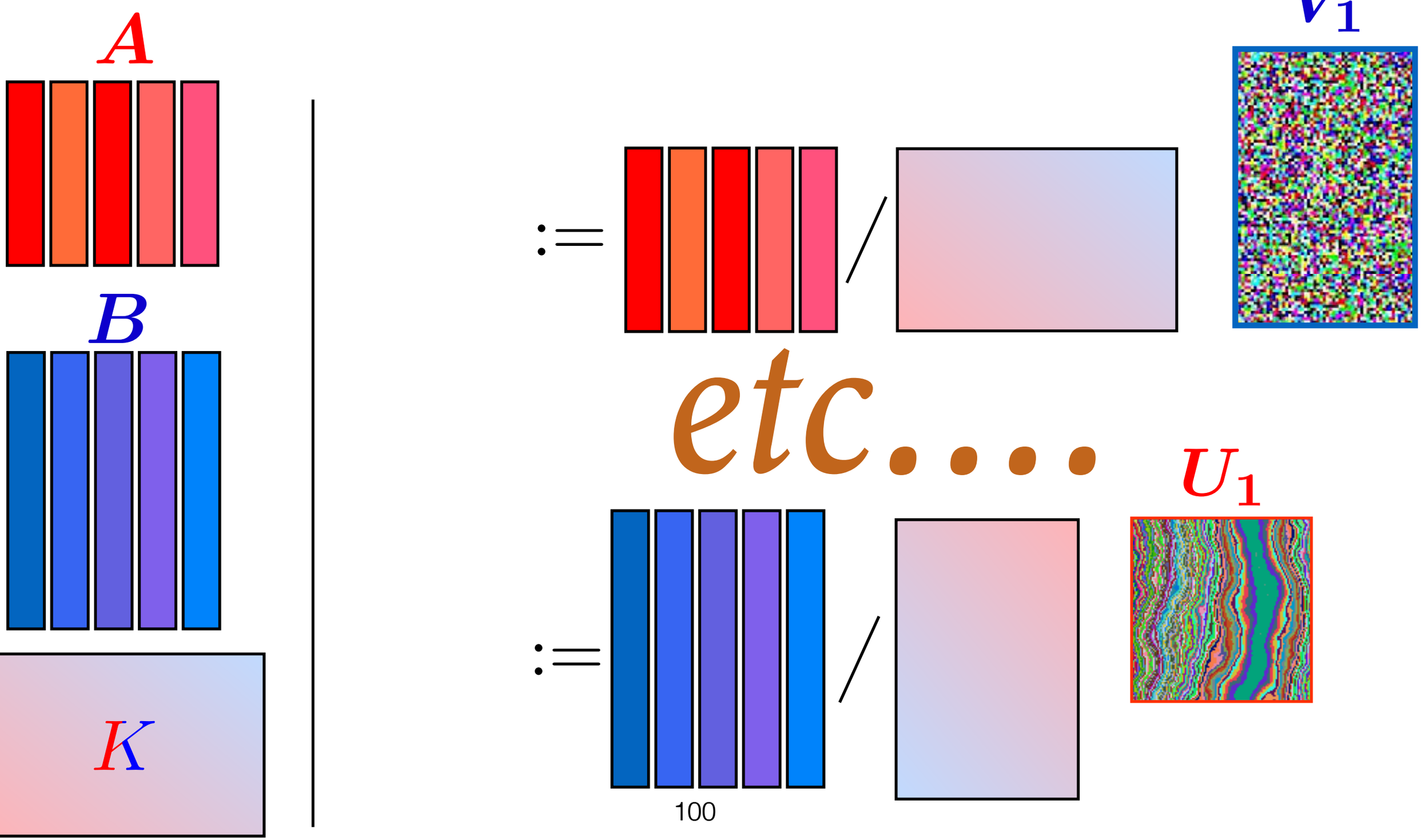
Also embarrassingly parallel

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

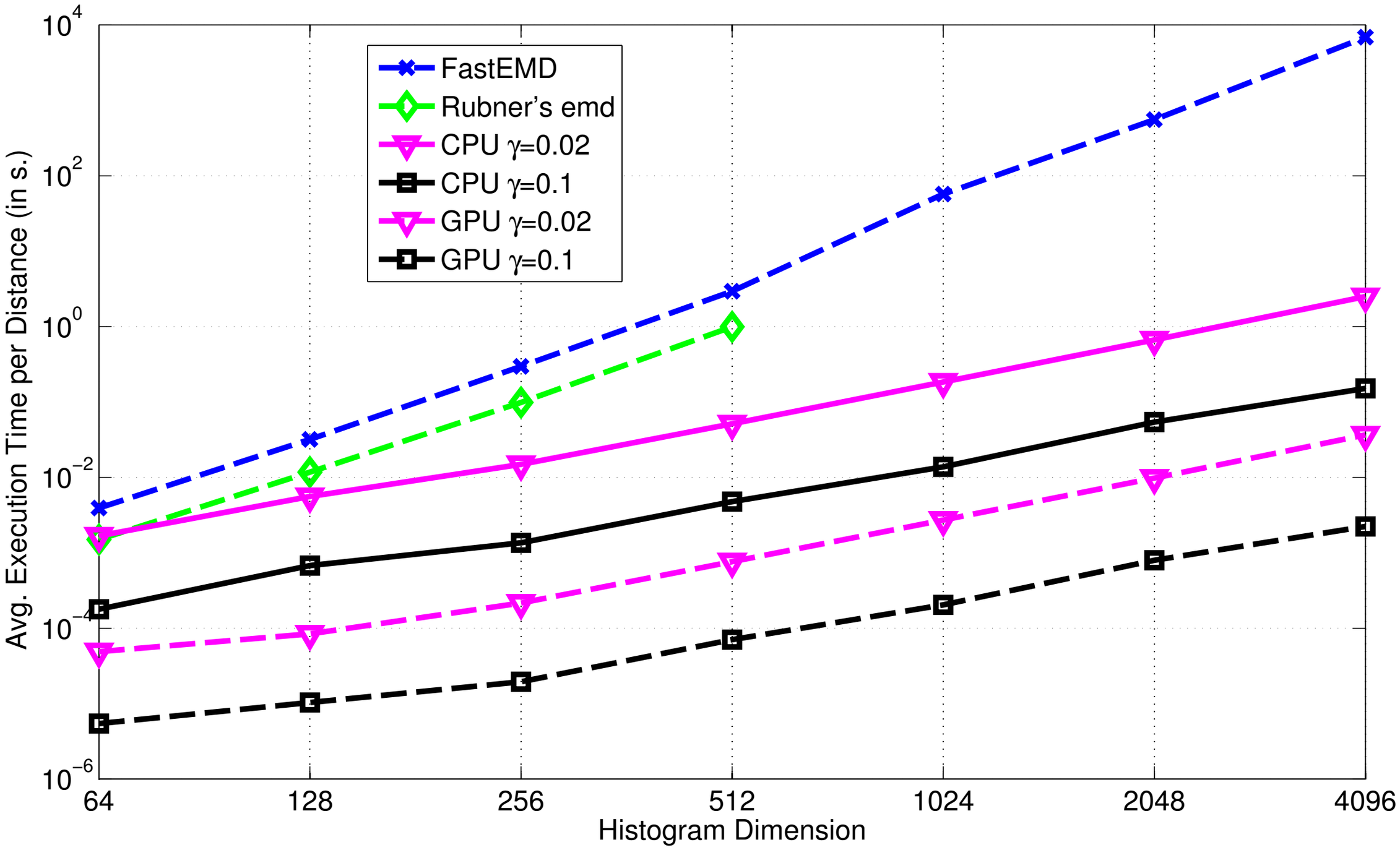


Also embarrassingly parallel

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



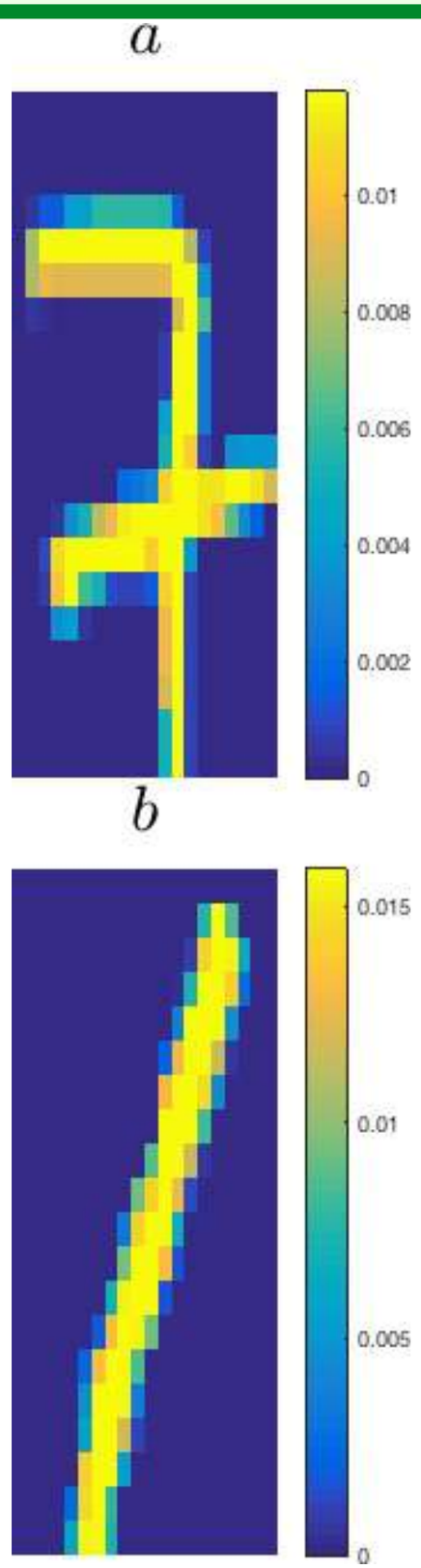
Very Fast EMD Approx. Solver



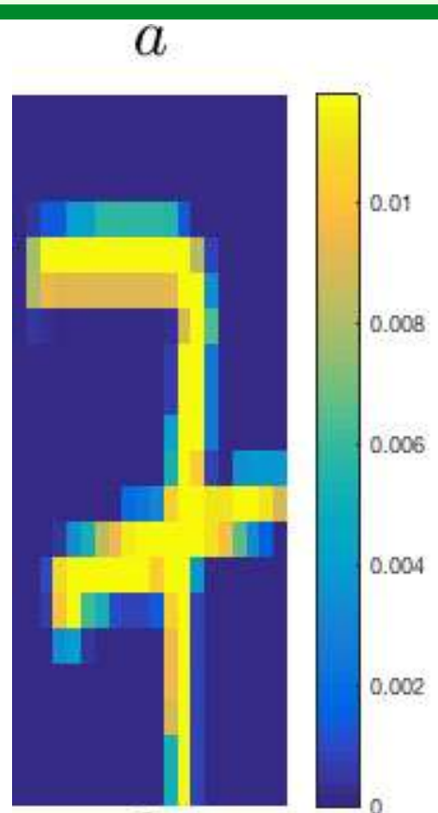
Setup. $(\Omega, \boldsymbol{D})$ is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance 10^{-2}.

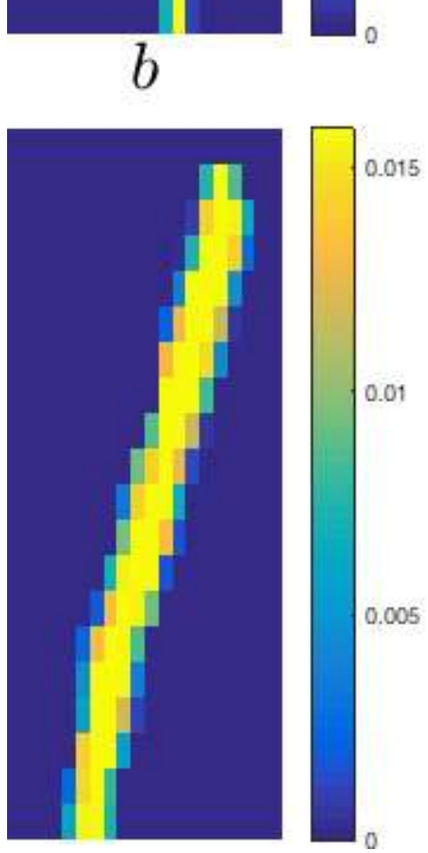
Very Fast EMD Approx. Solver

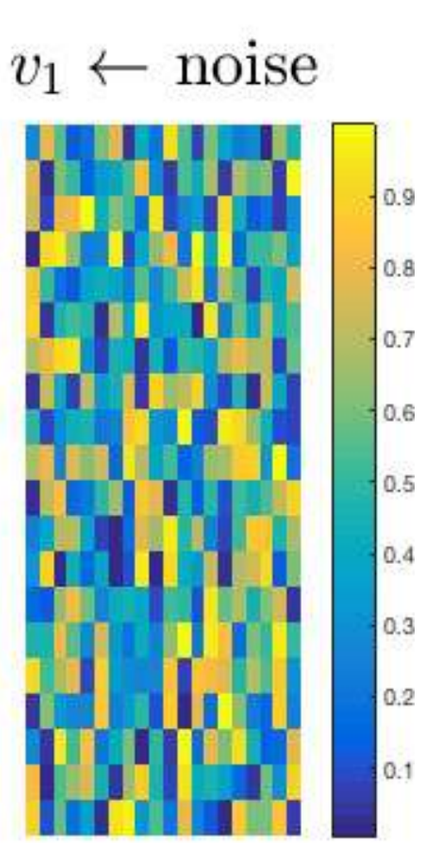
Very Fast EMD Approx. Solver



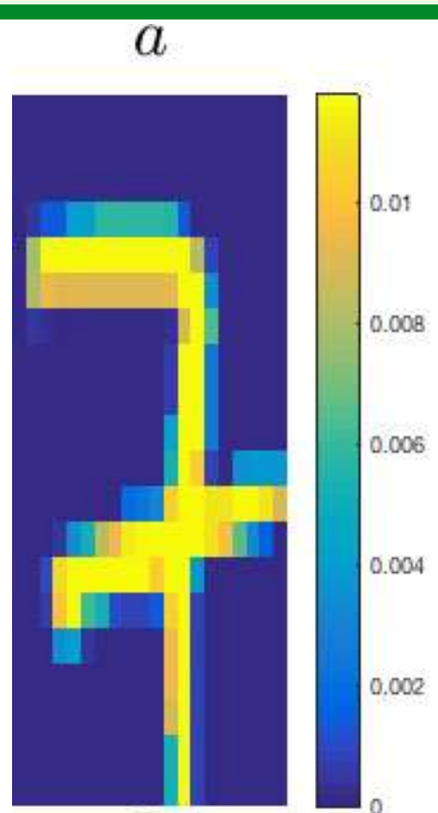
Very Fast EMD Approx. Solver

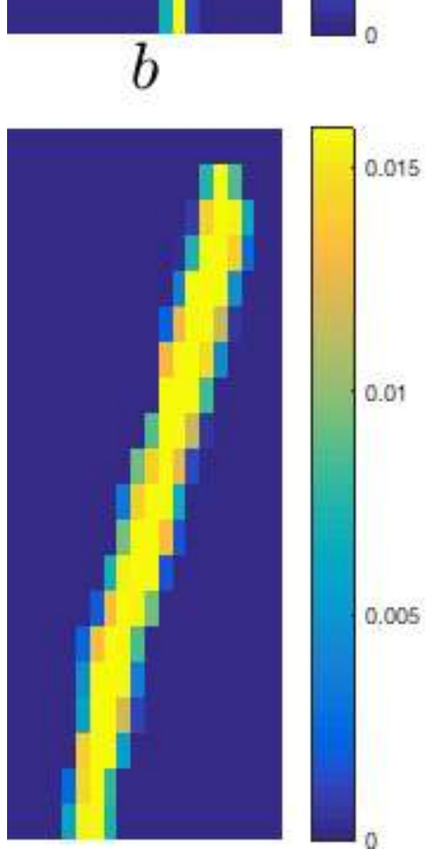


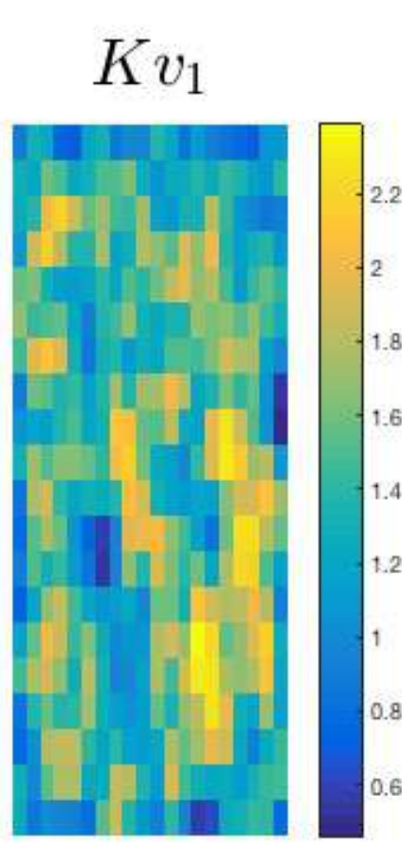




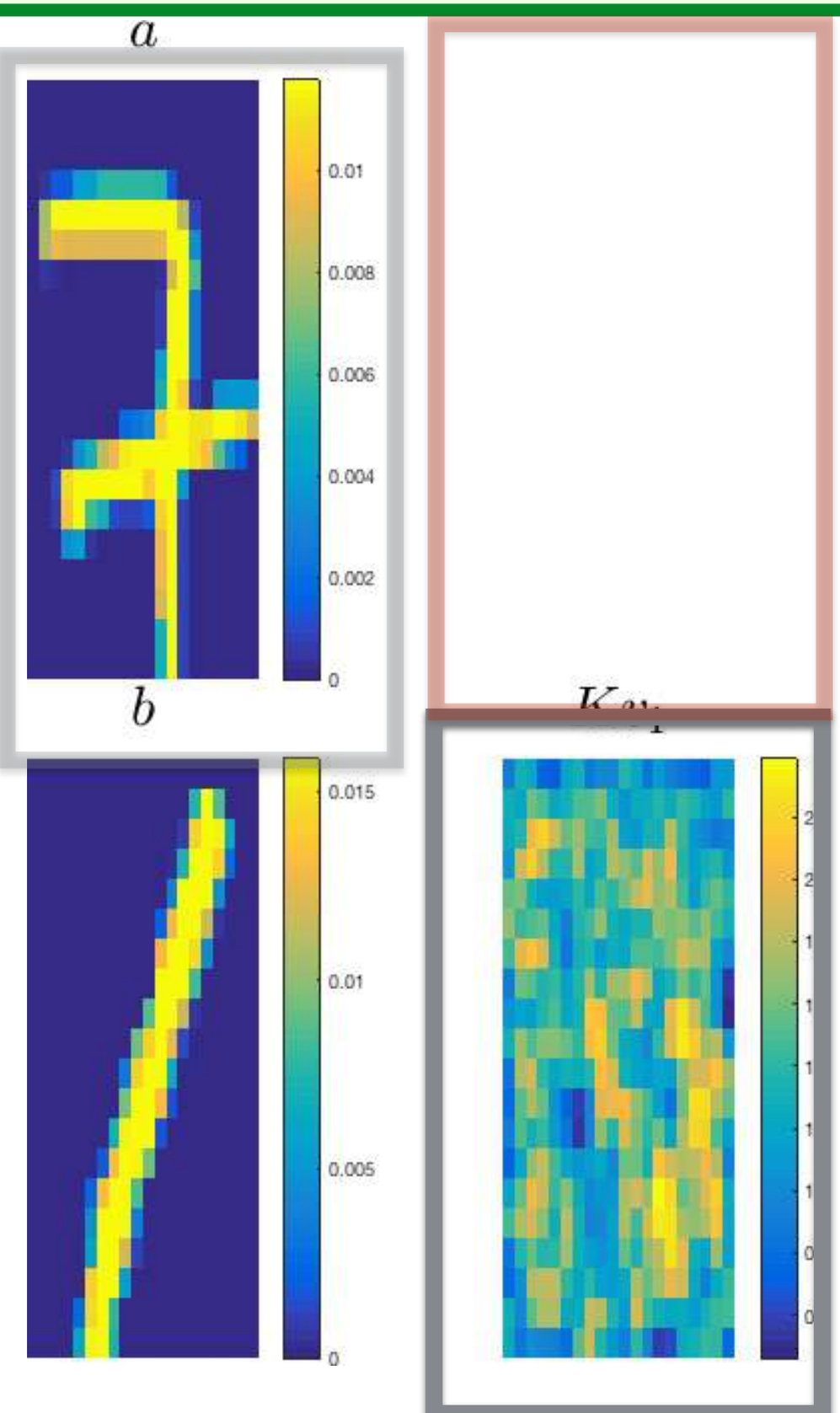
Very Fast EMD Approx. Solver



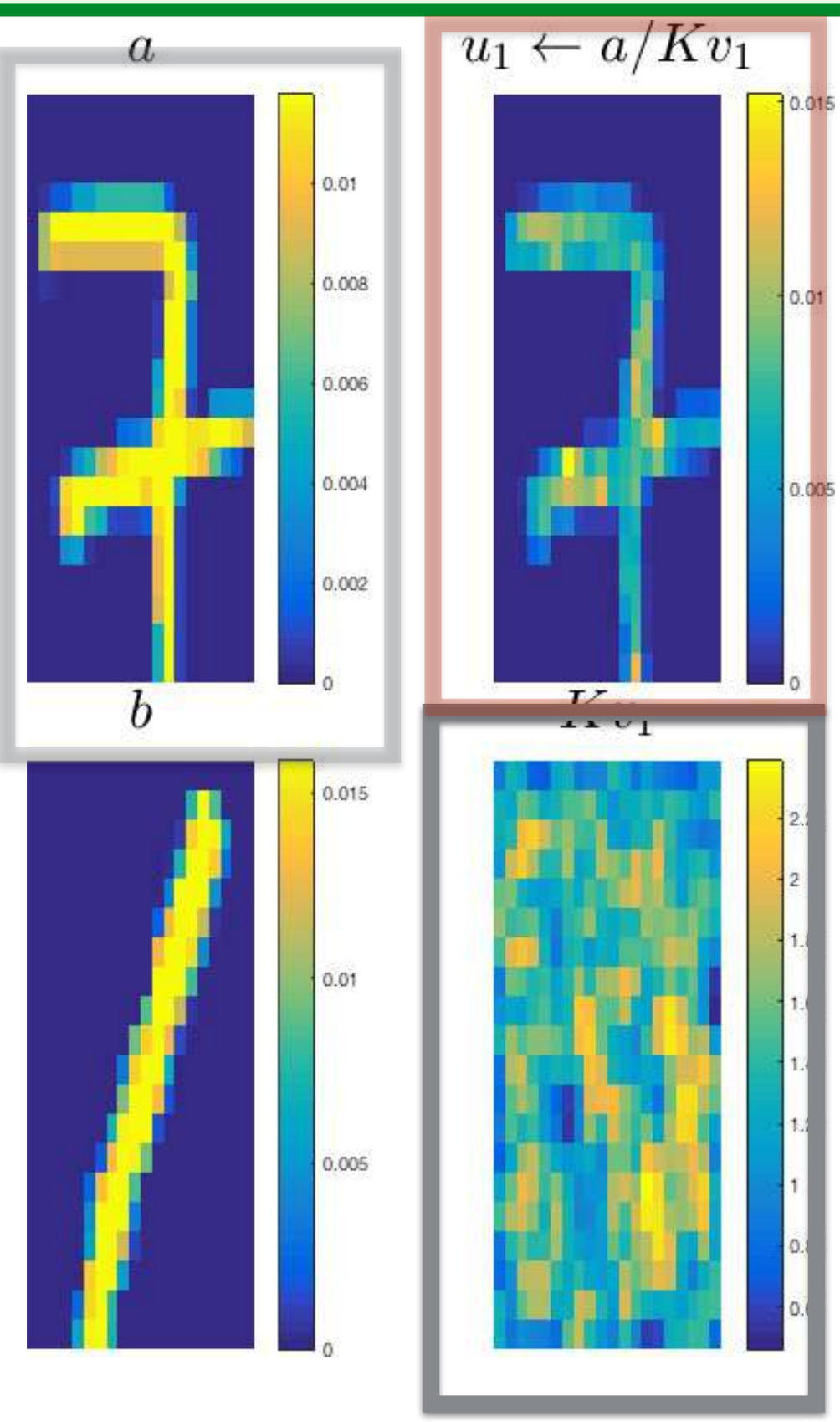




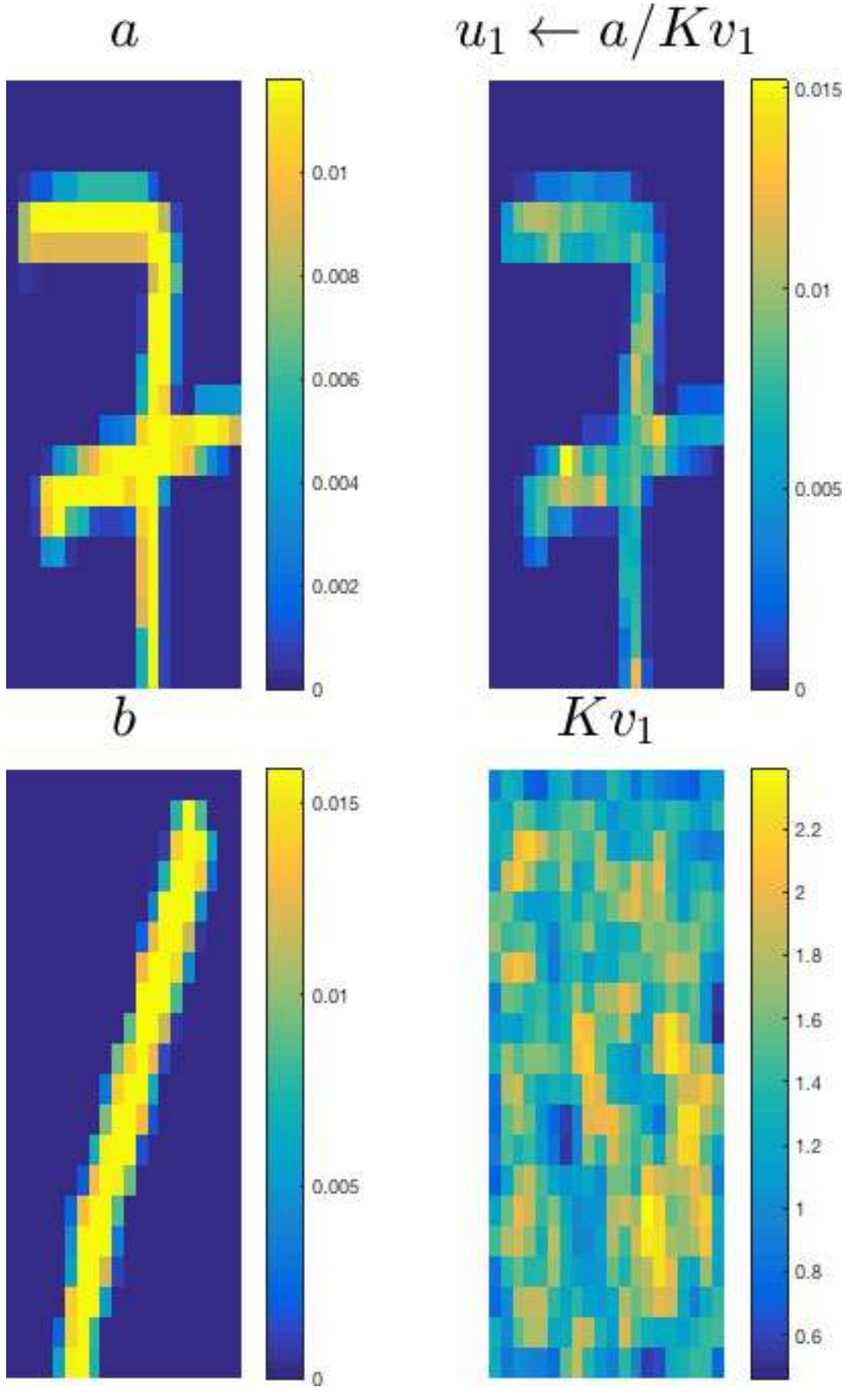
Very Fast EMD Approx. Solver



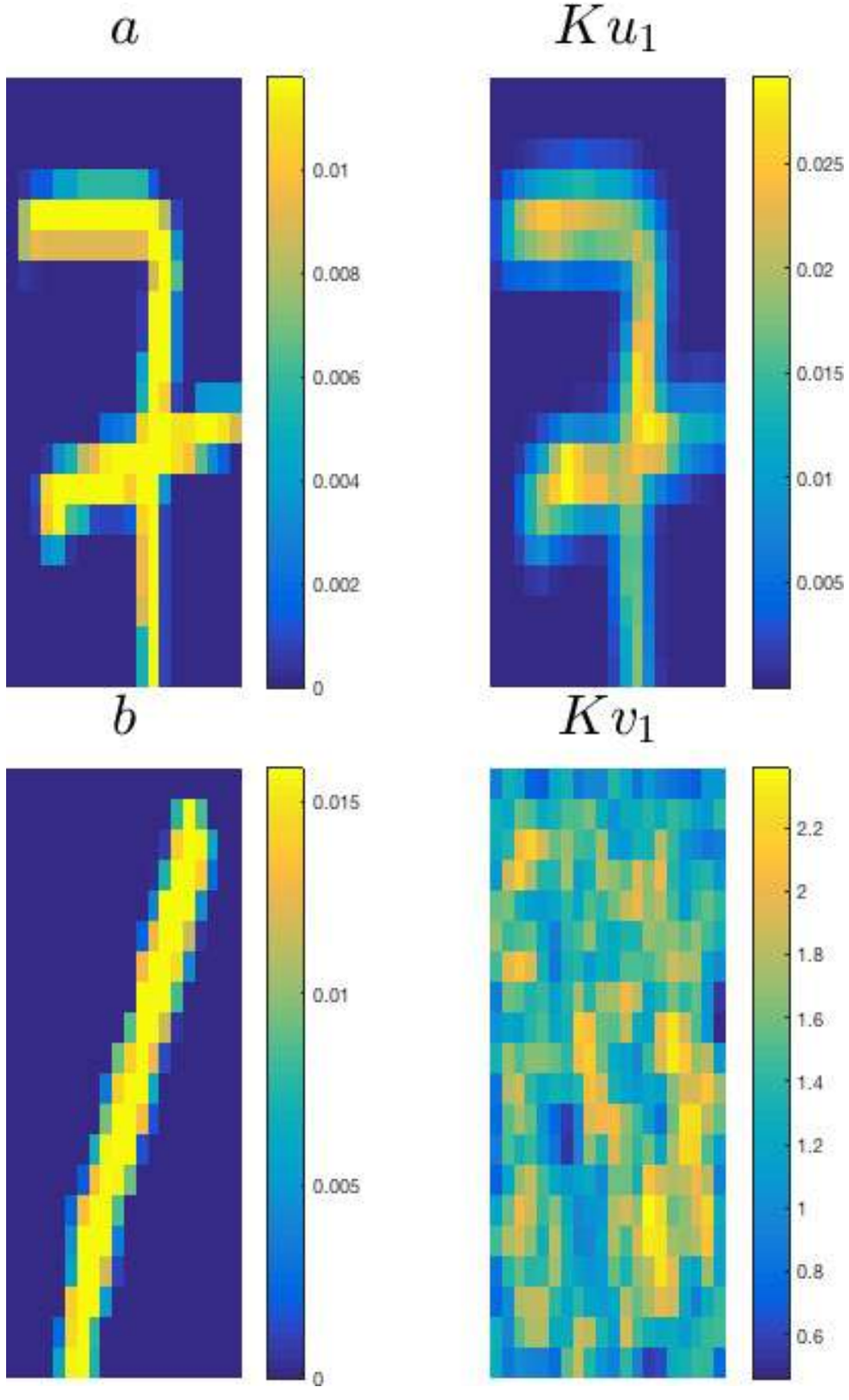
Very Fast EMD Approx. Solver



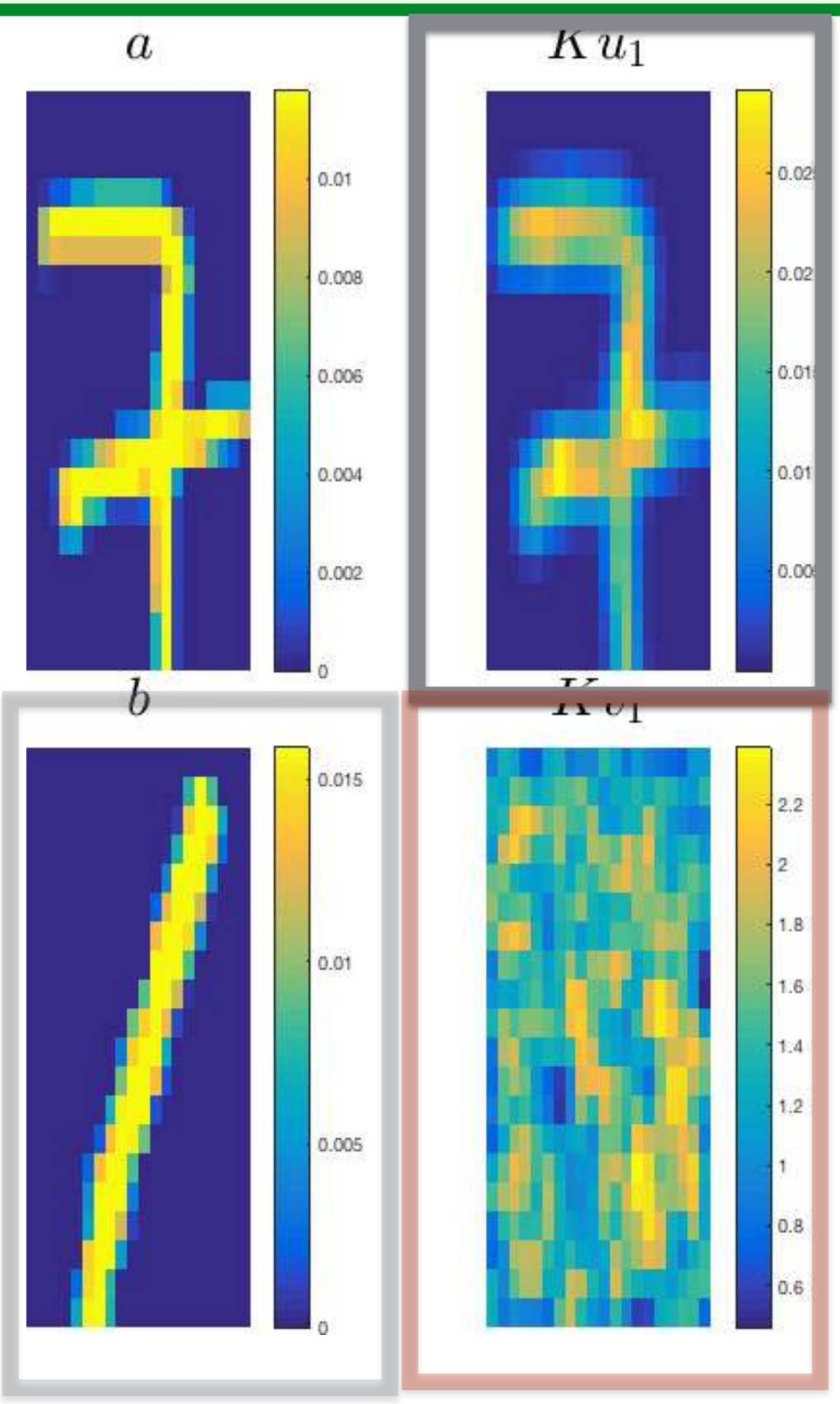
Very Fast EMD Approx. Solver



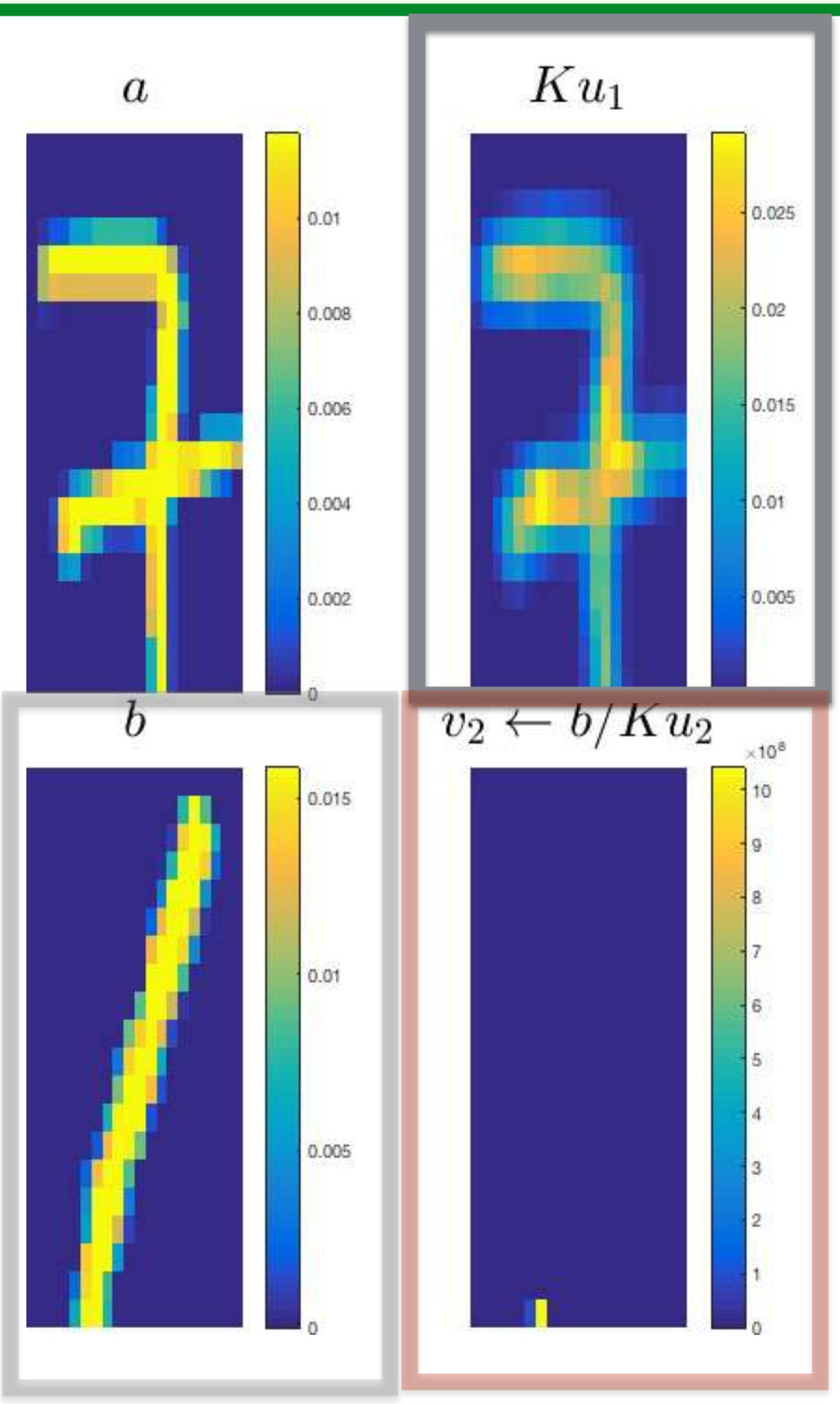
Very Fast EMD Approx. Solver



Very Fast EMD Approx. Solver

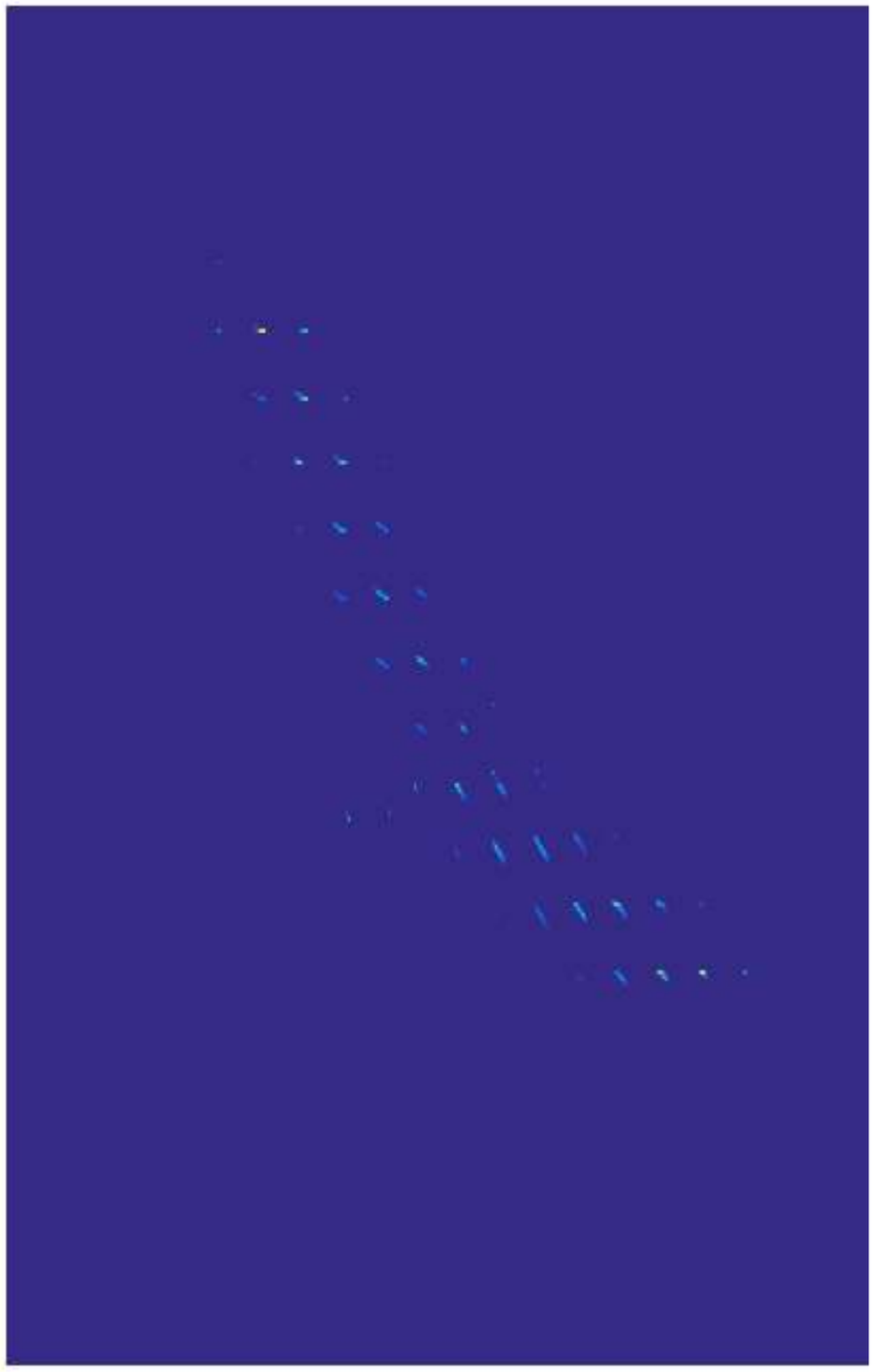


Very Fast EMD Approx. Solver



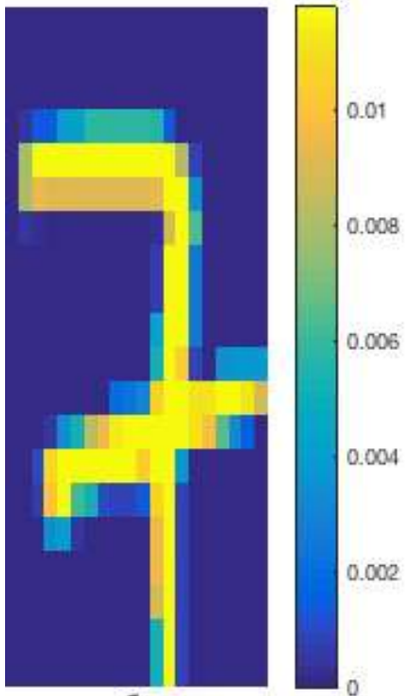
$P_1 = D(u_1)KD(v_1)$

$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

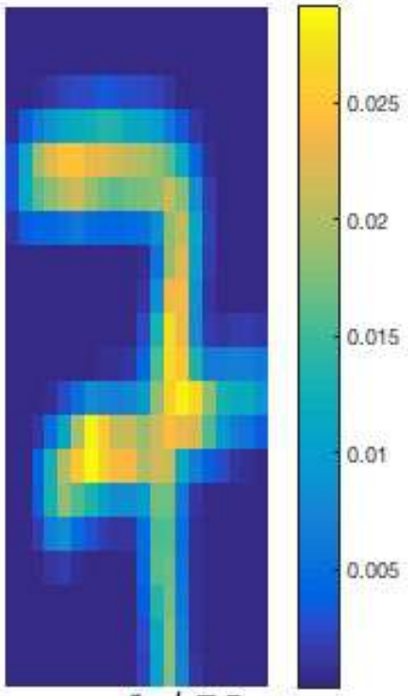


Very Fast EMD Approx. Solver

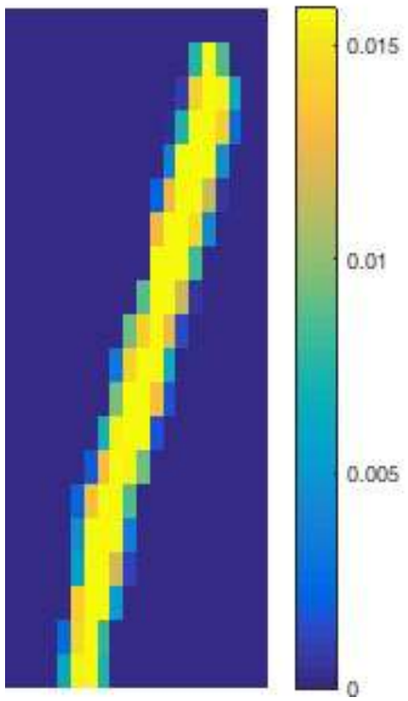
a



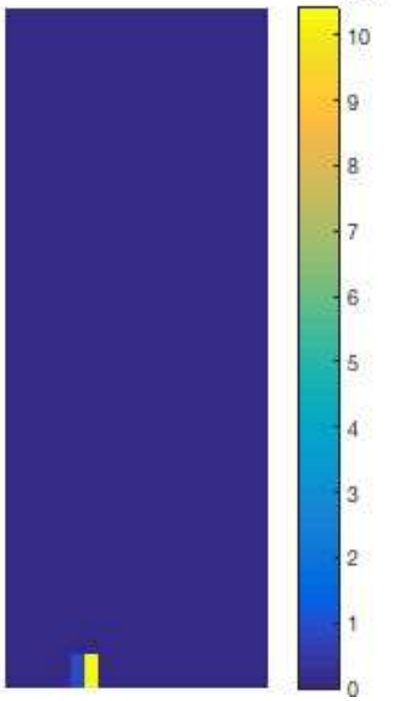
Ku_1



b

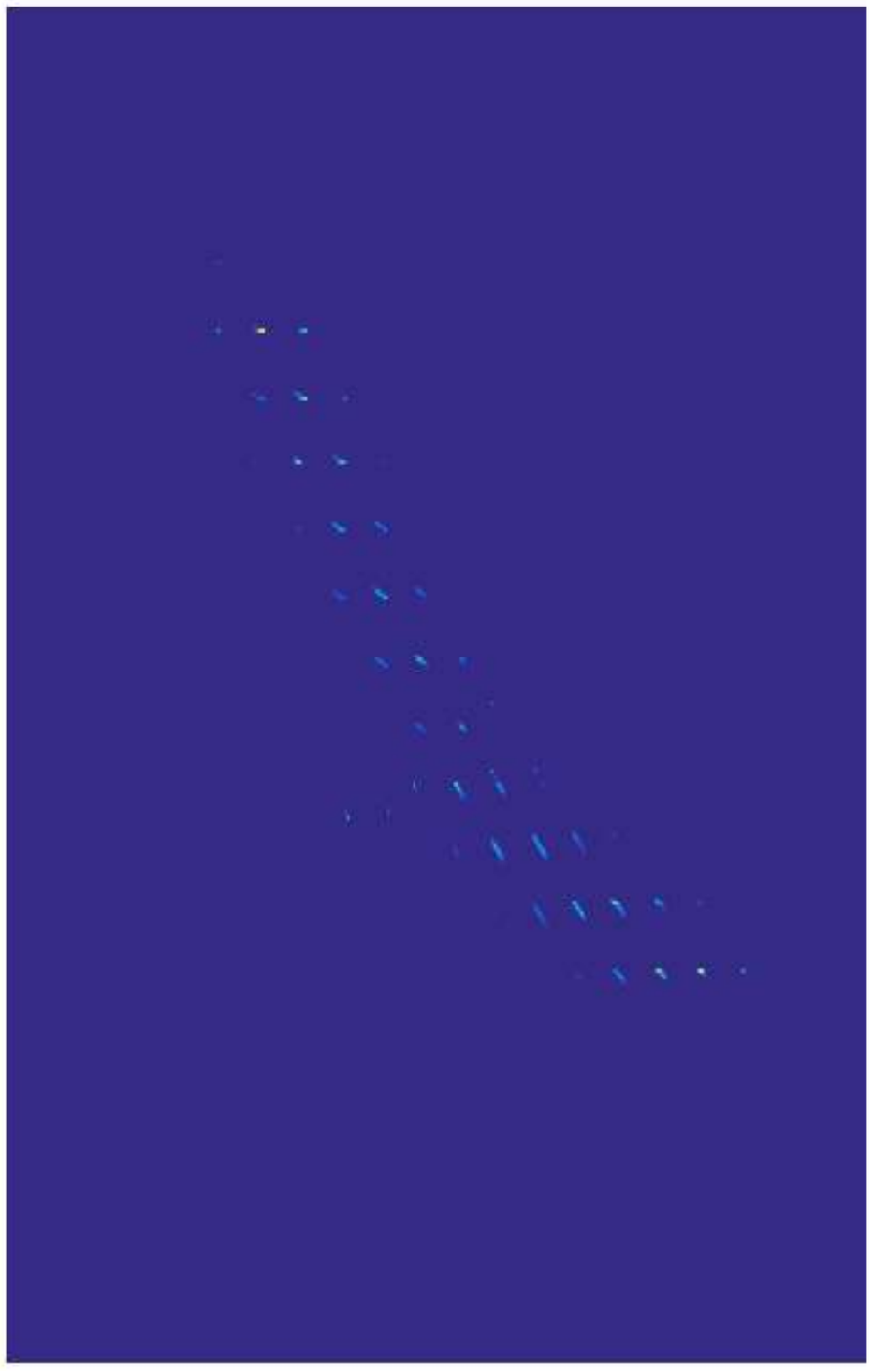


$v_2 \leftarrow b/Ku_2$



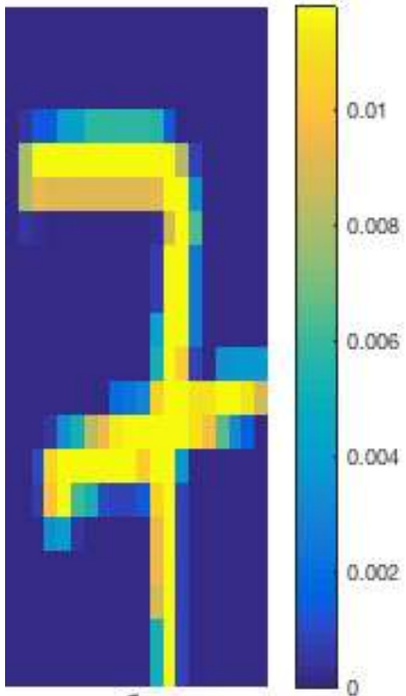
$P_1 = D(u_1)KD(v_1)$

$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

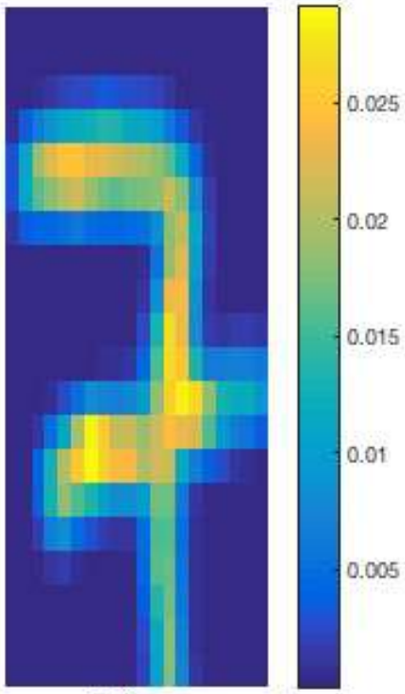


Very Fast EMD Approx. Solver

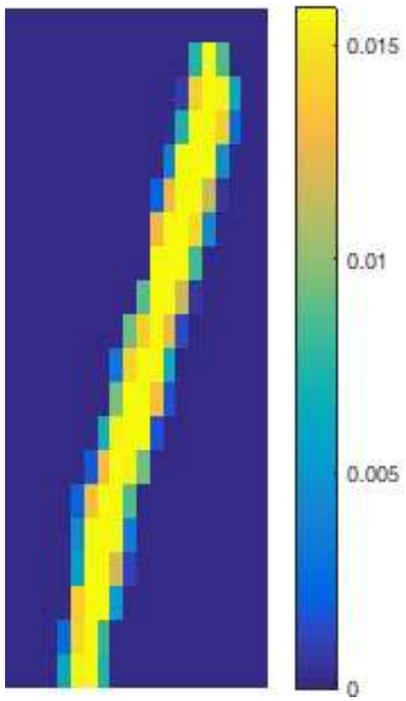
a



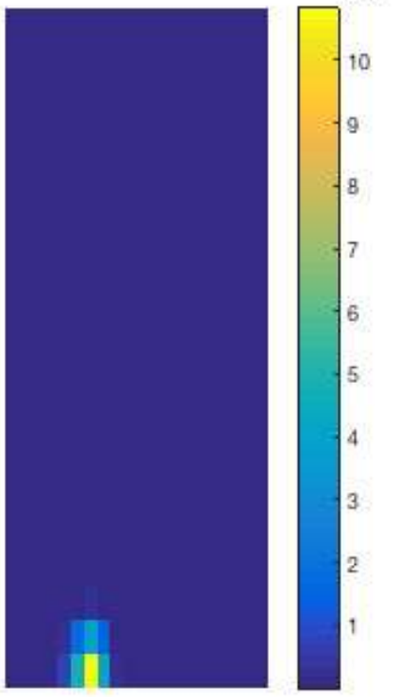
Ku_1



b

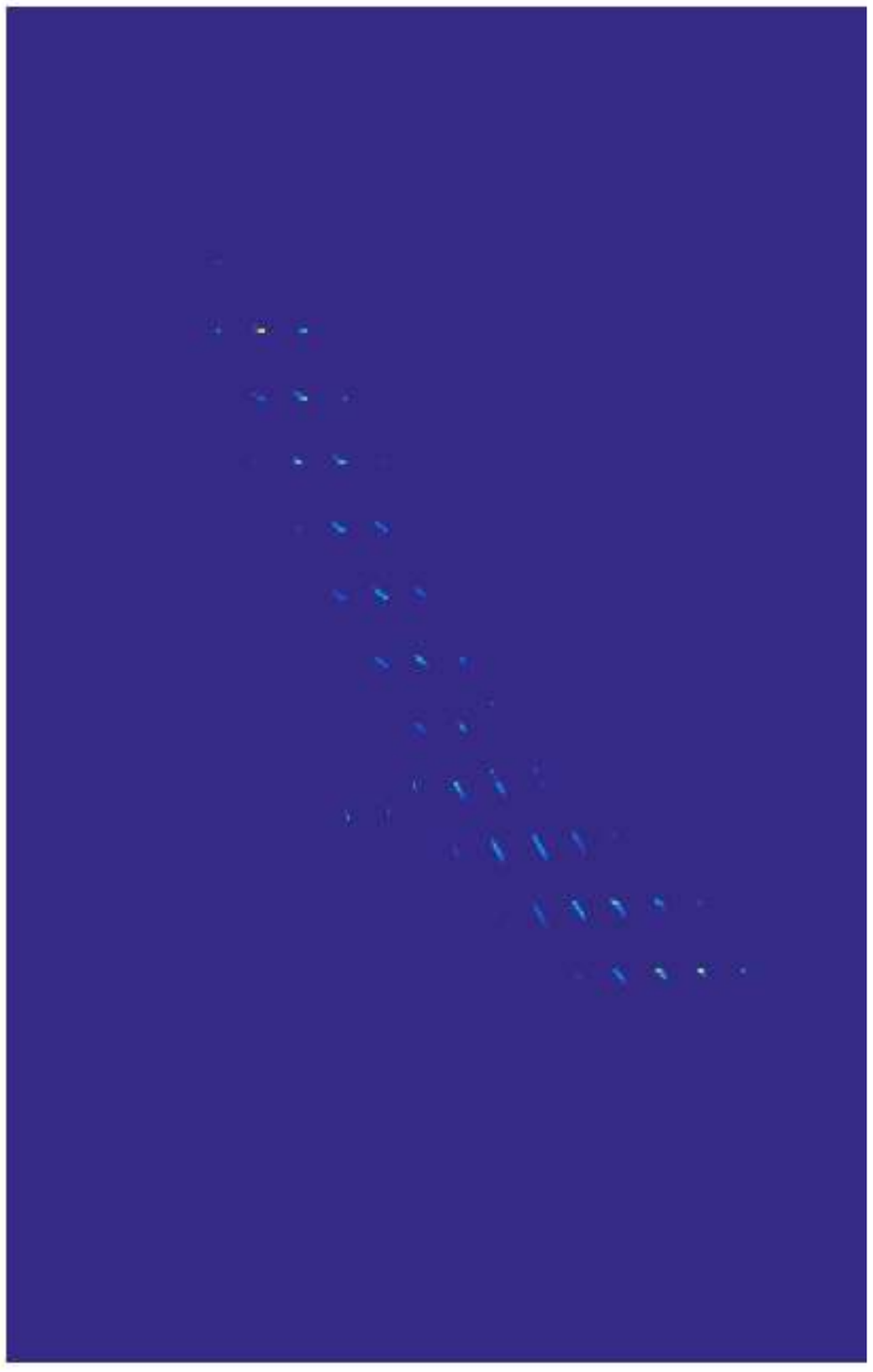


Kv_2



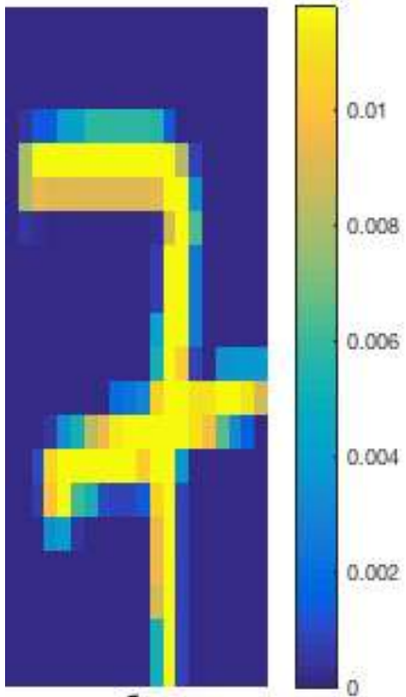
$P_1 = D(u_1)KD(v_1)$

$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

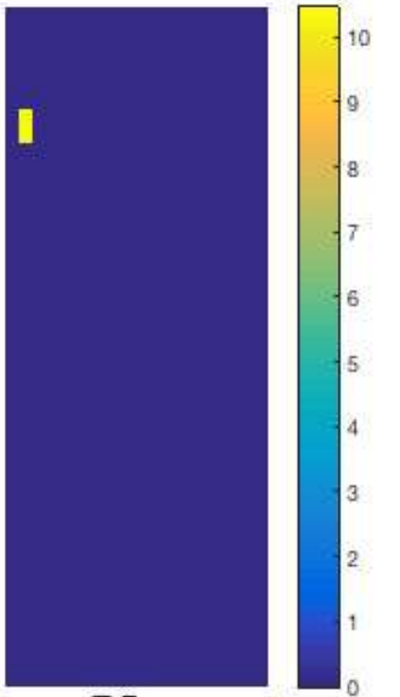


Very Fast EMD Approx. Solver

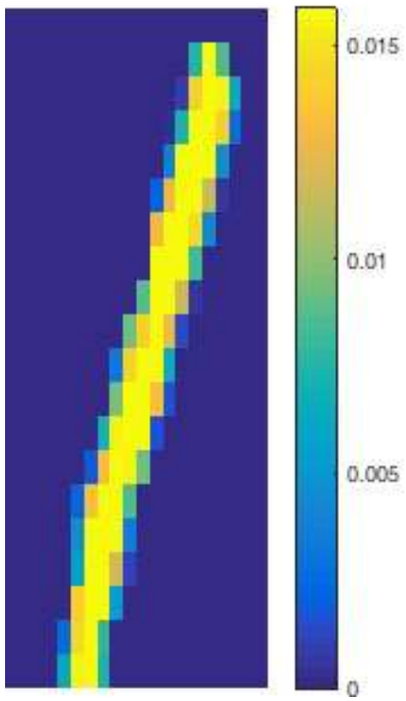
a



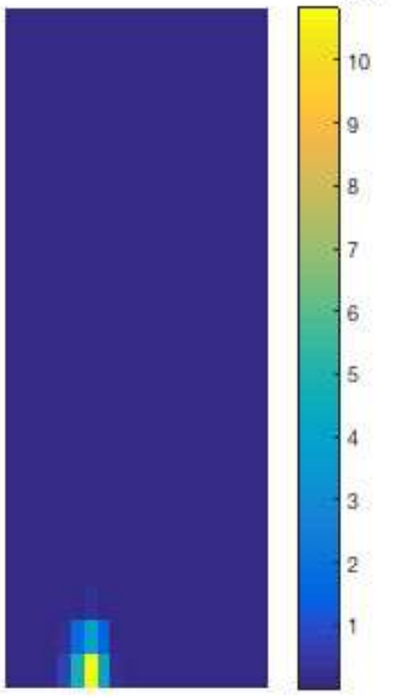
$u_2 \leftarrow a/Kv_2$



b

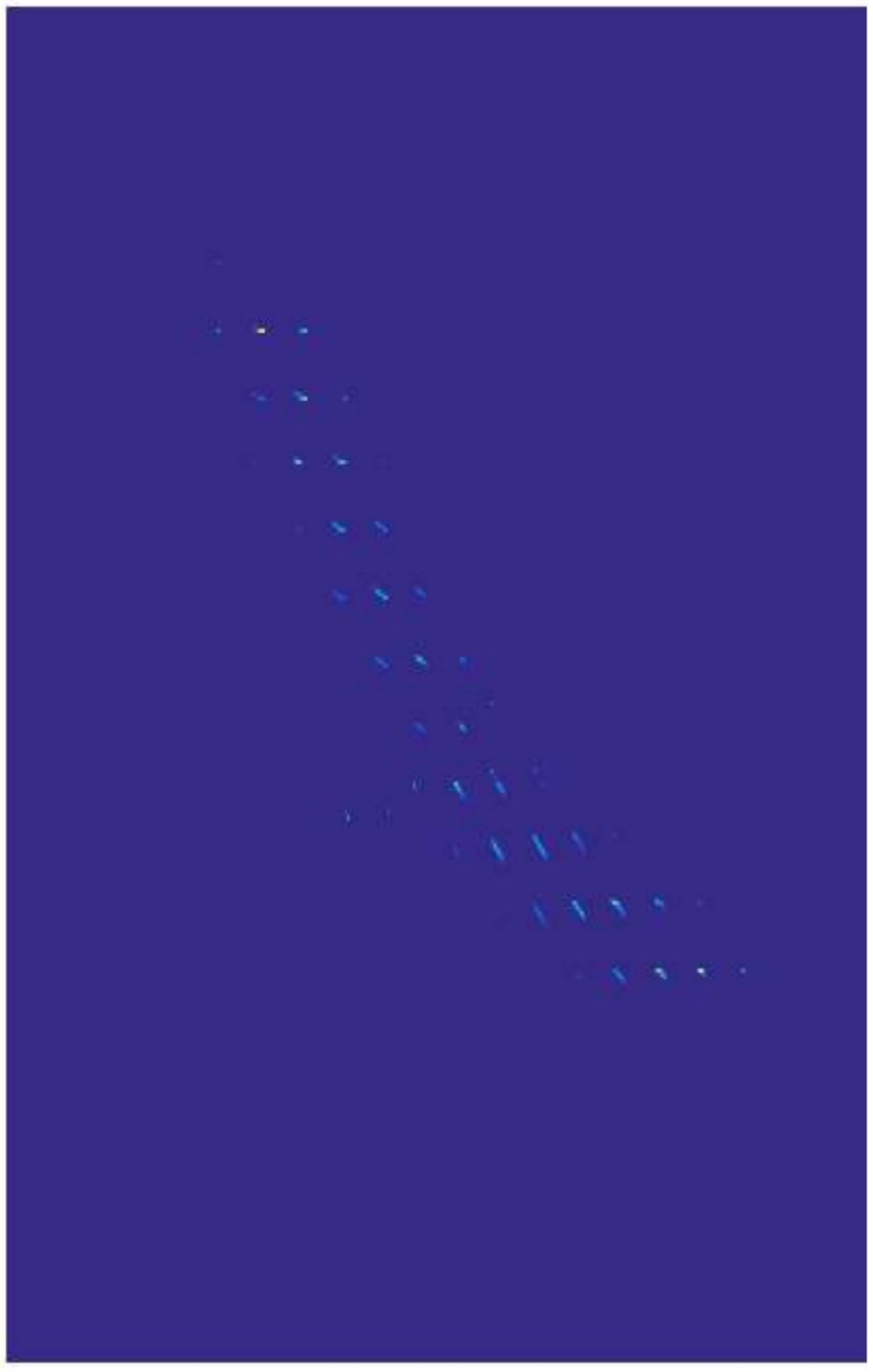


Kv_2

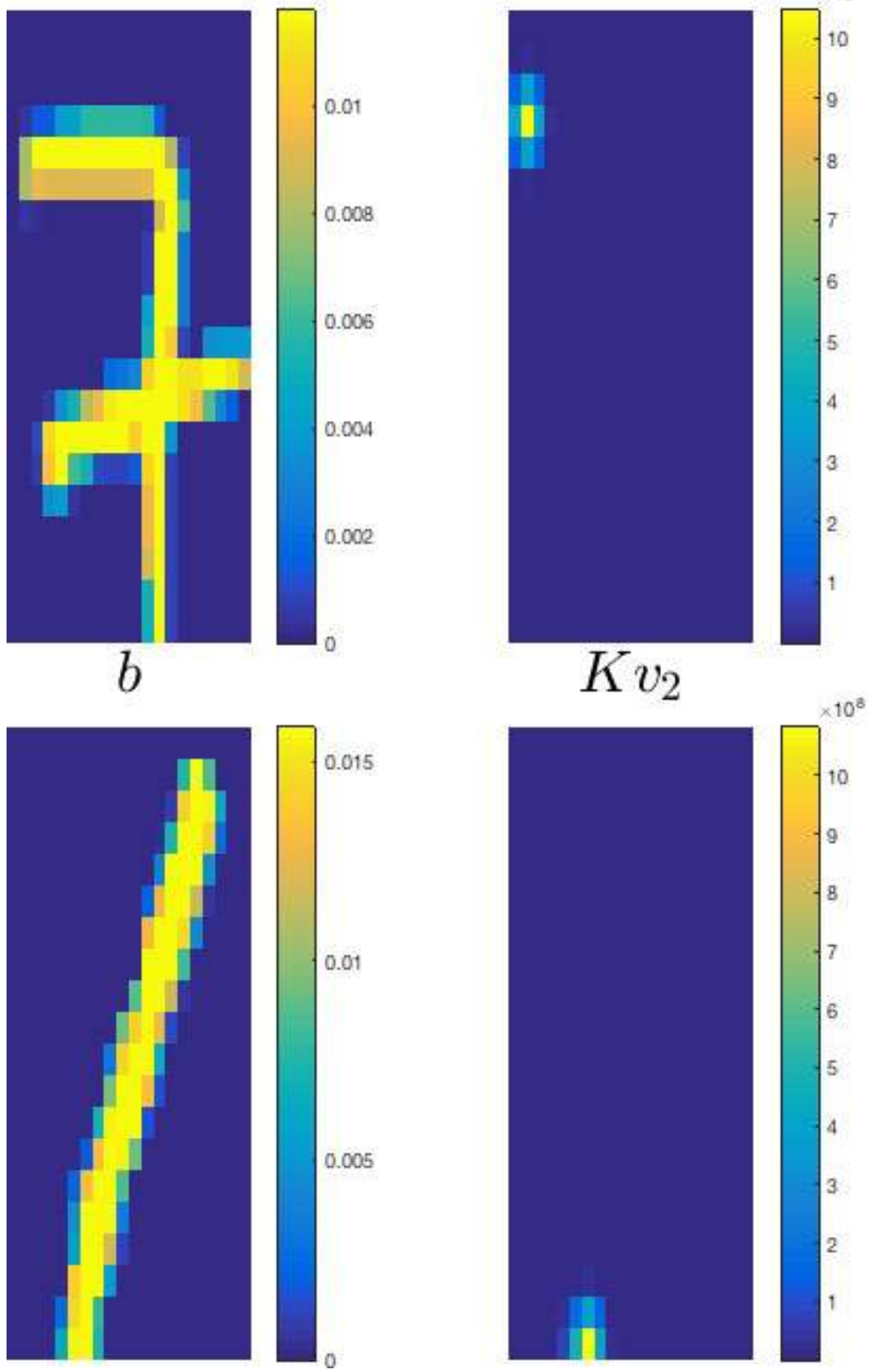


$P_1 = D(u_1)KD(v_1)$

$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

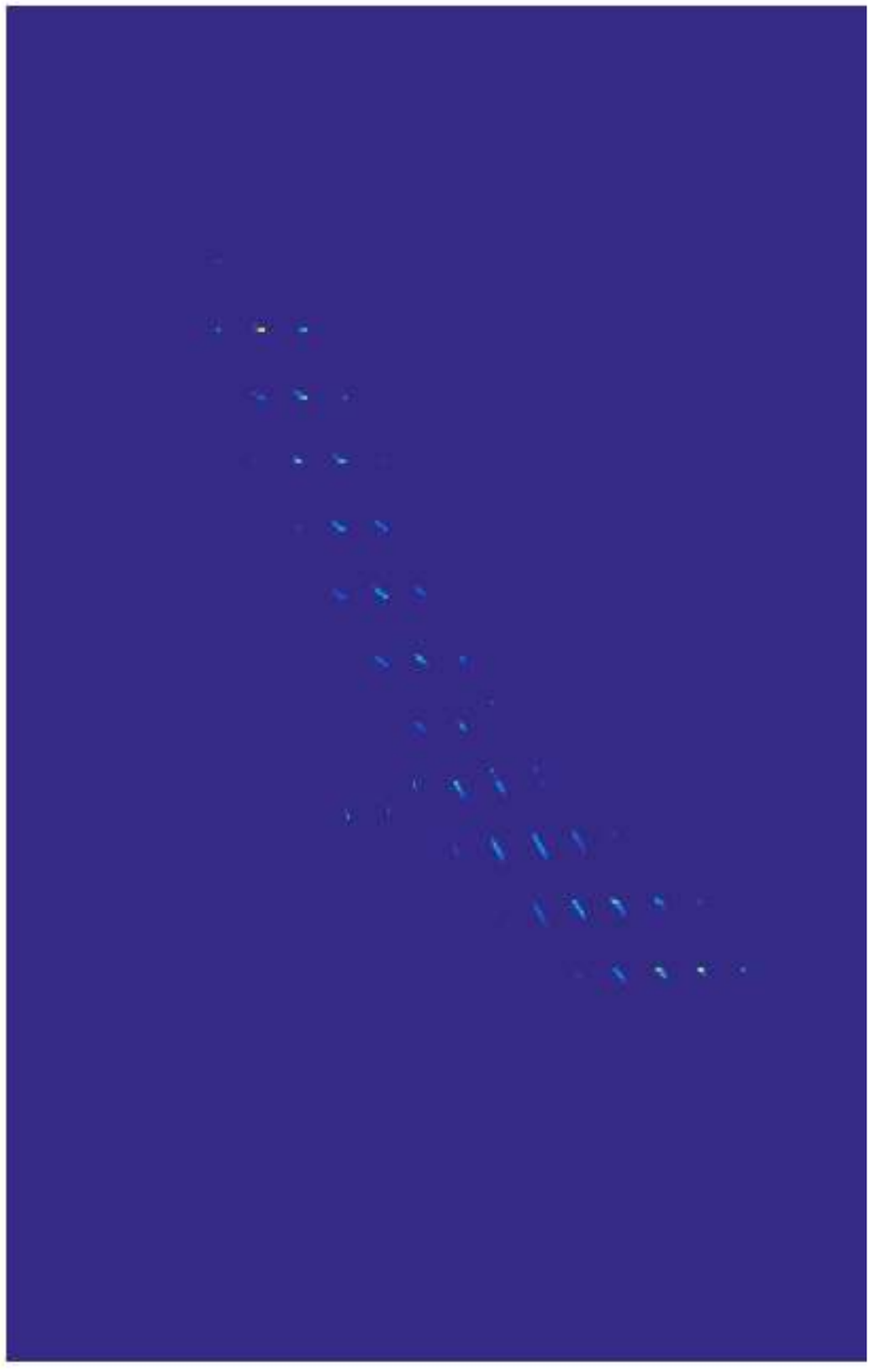


Very Fast EMD Approx. Solver

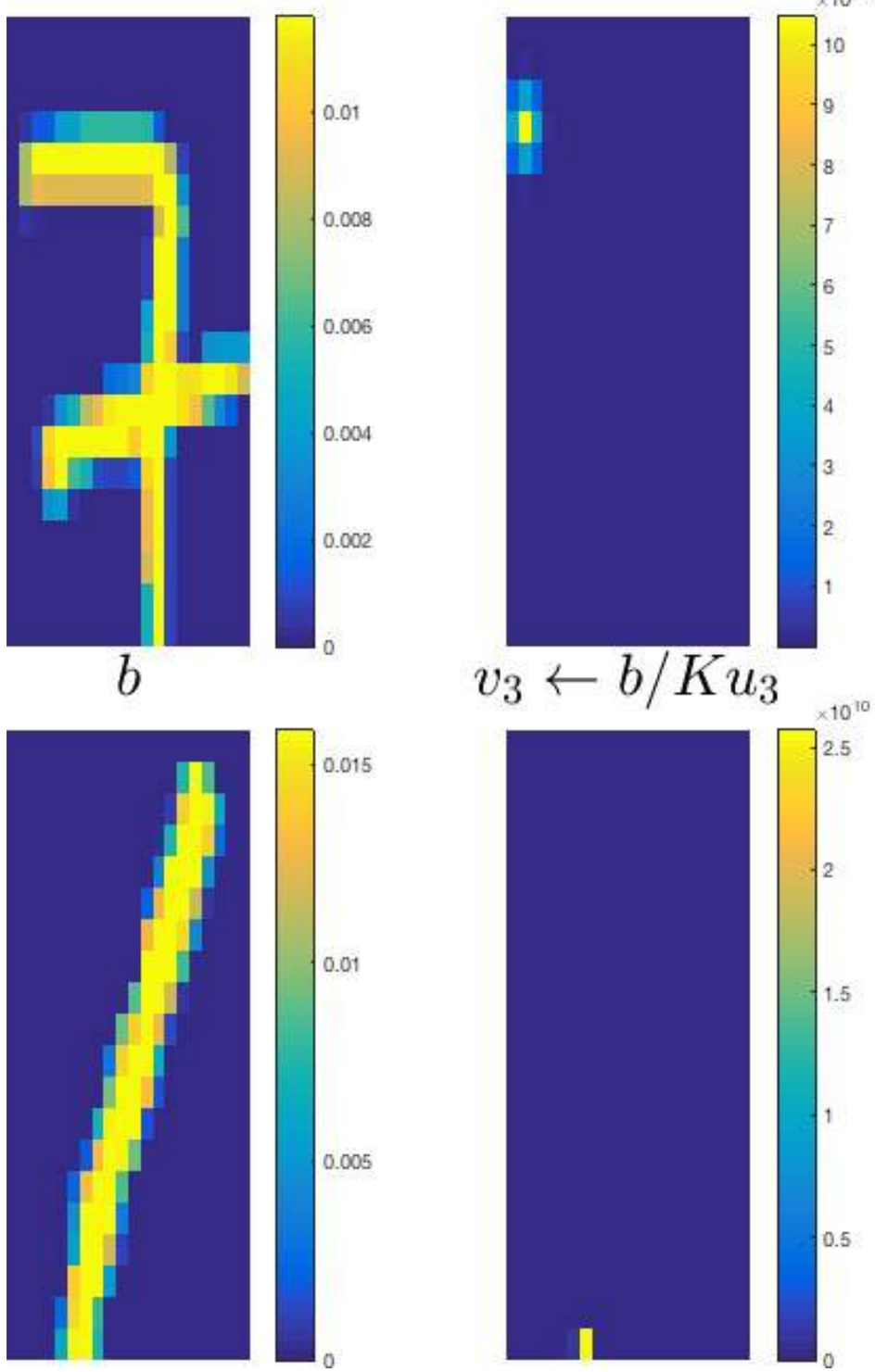


$P_1 = D(u_1)KD(v_1)$

$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$

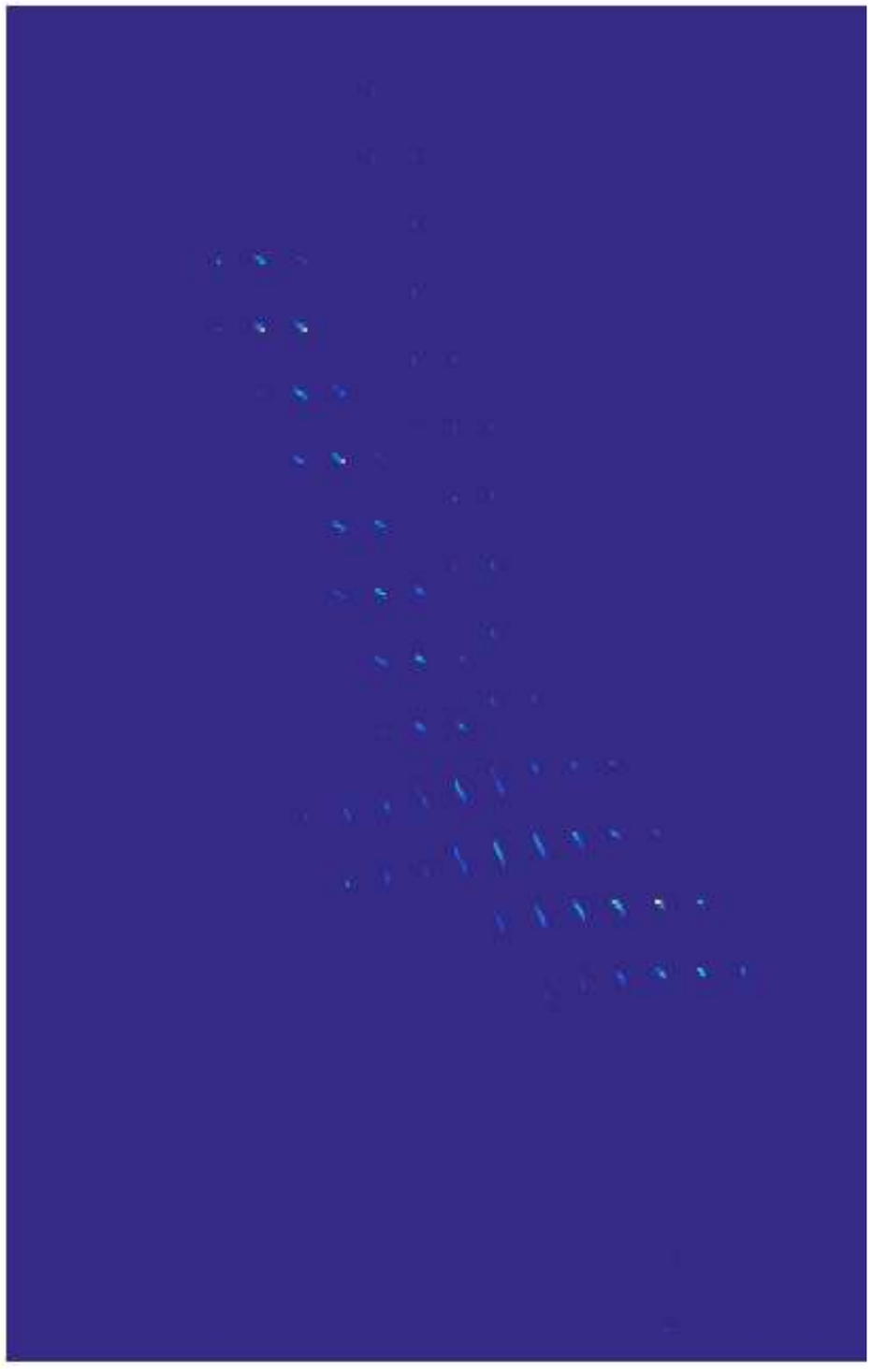


Very Fast EMD Approx. Solver

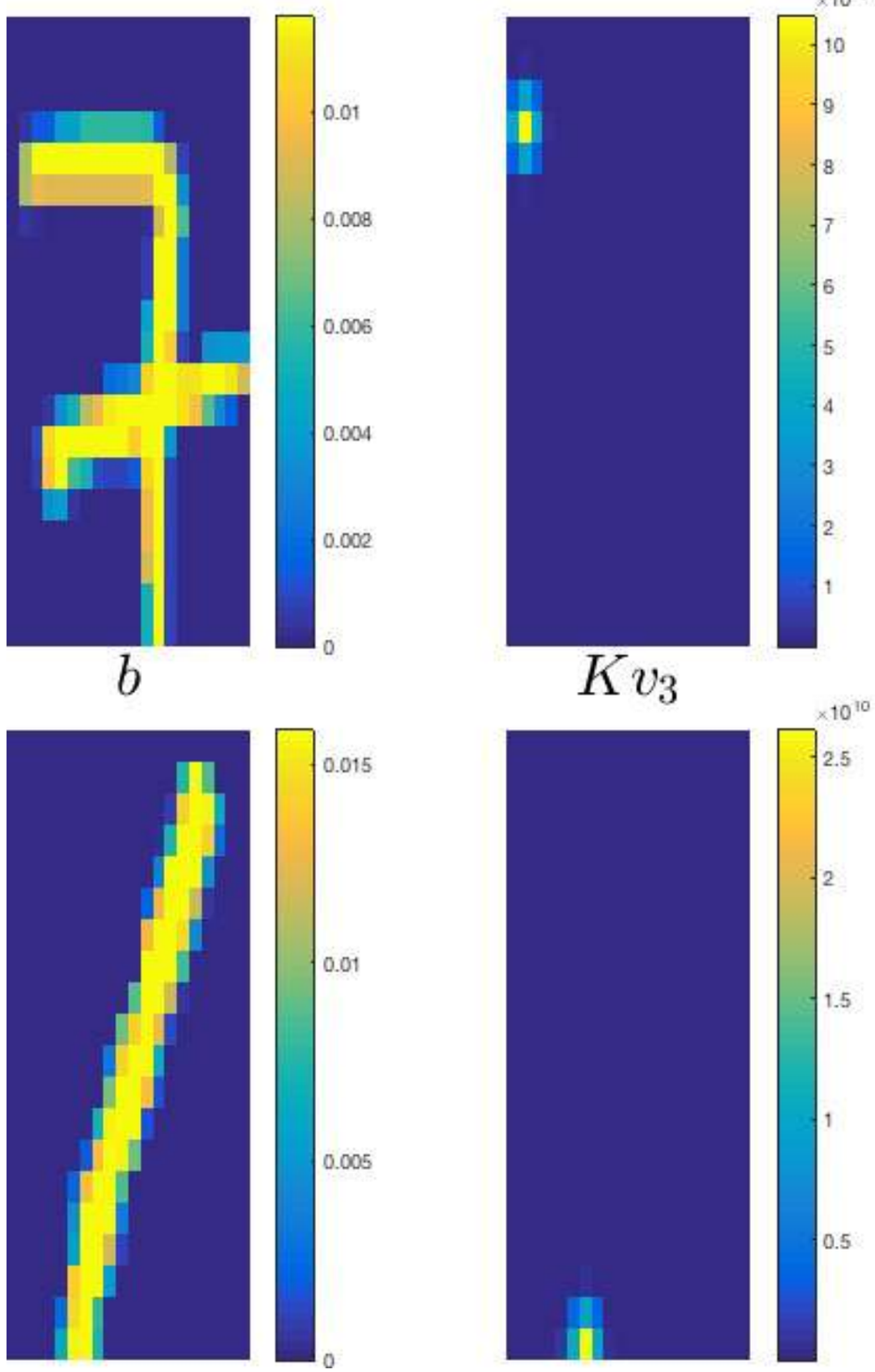


$P_2 = D(u_2)KD(v_2)$

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

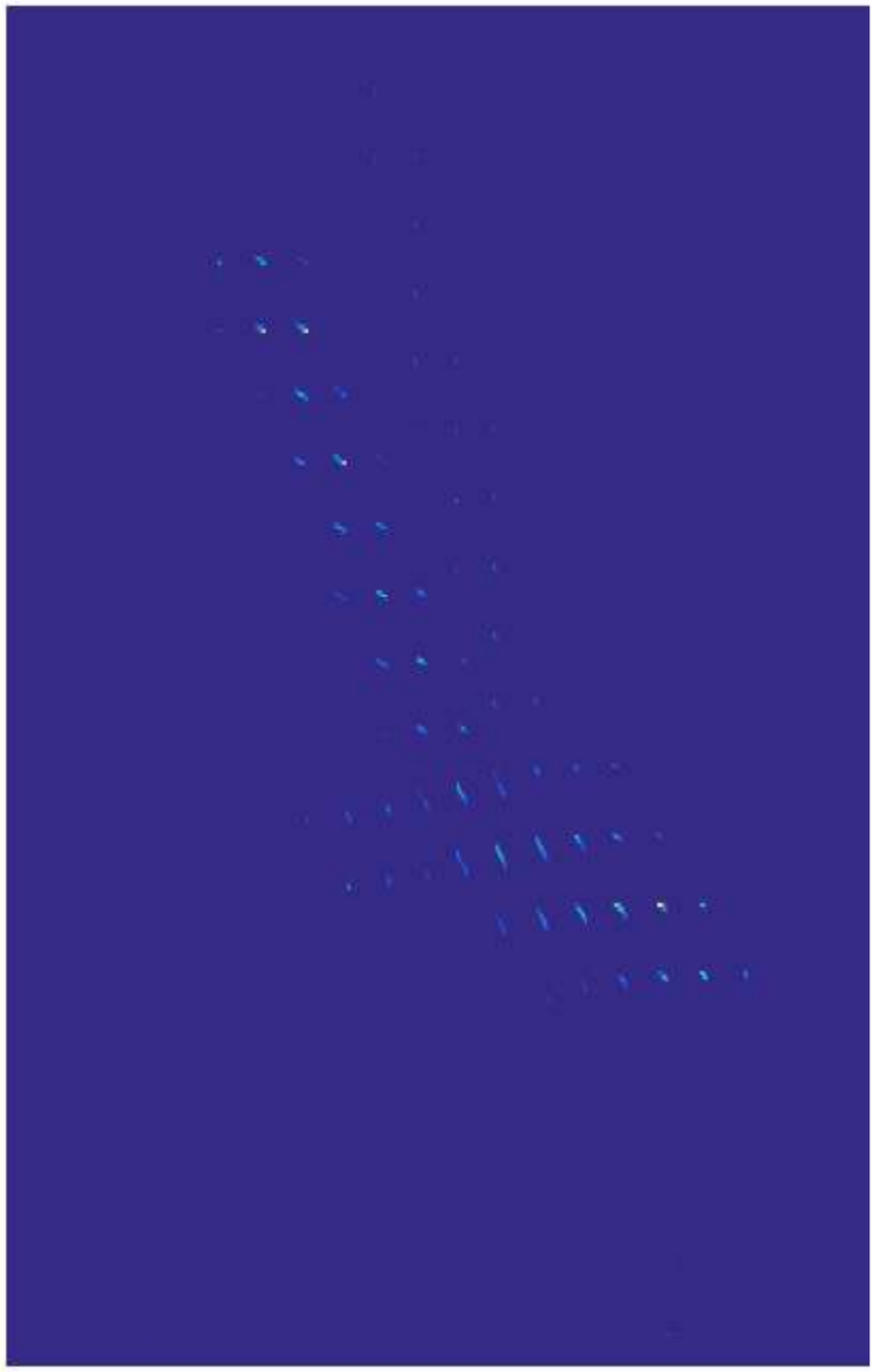


Very Fast EMD Approx. Solver



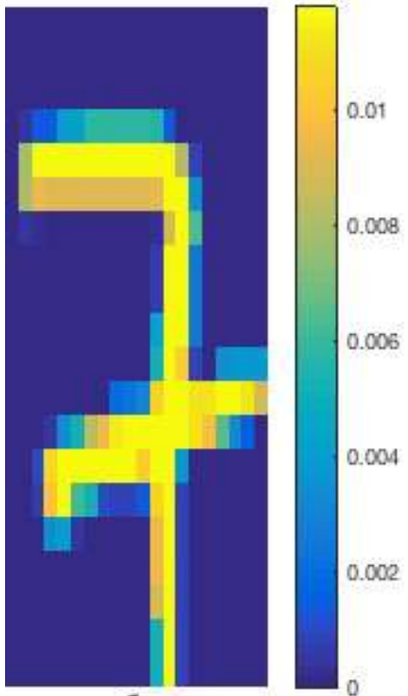
$P_2 = D(u_2)KD(v_2)$

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

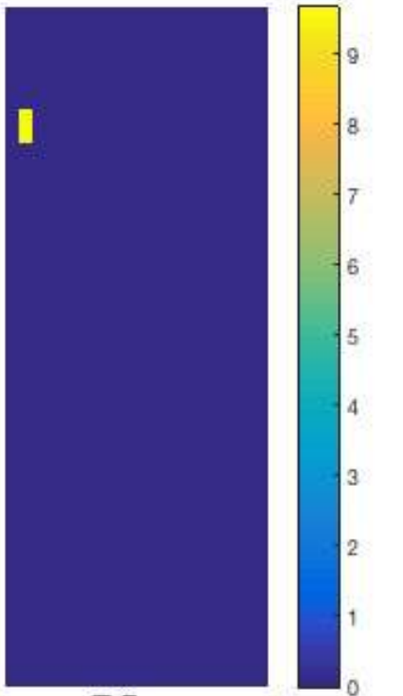


Very Fast EMD Approx. Solver

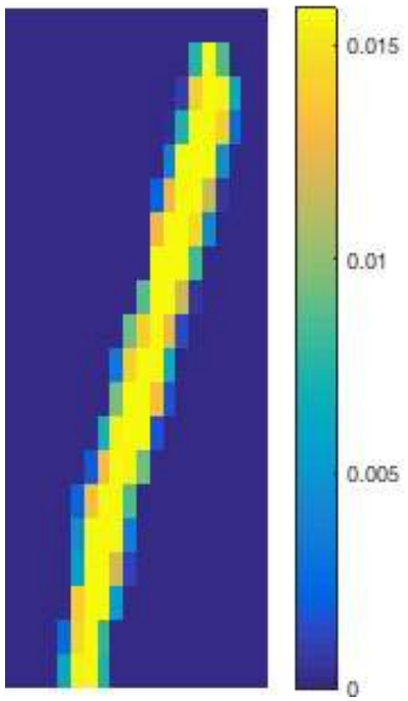
a



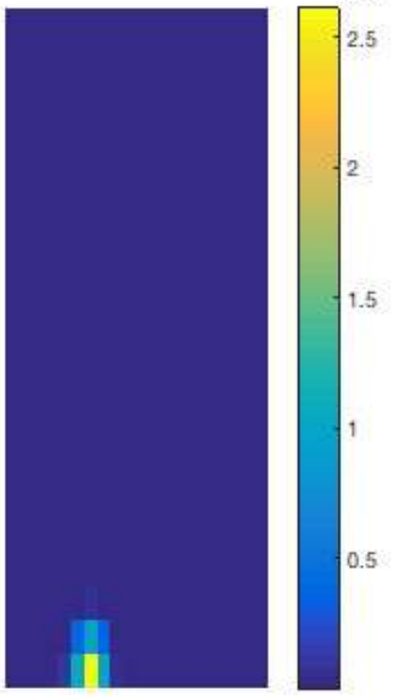
$u_3 \leftarrow a/Kv_3$



b

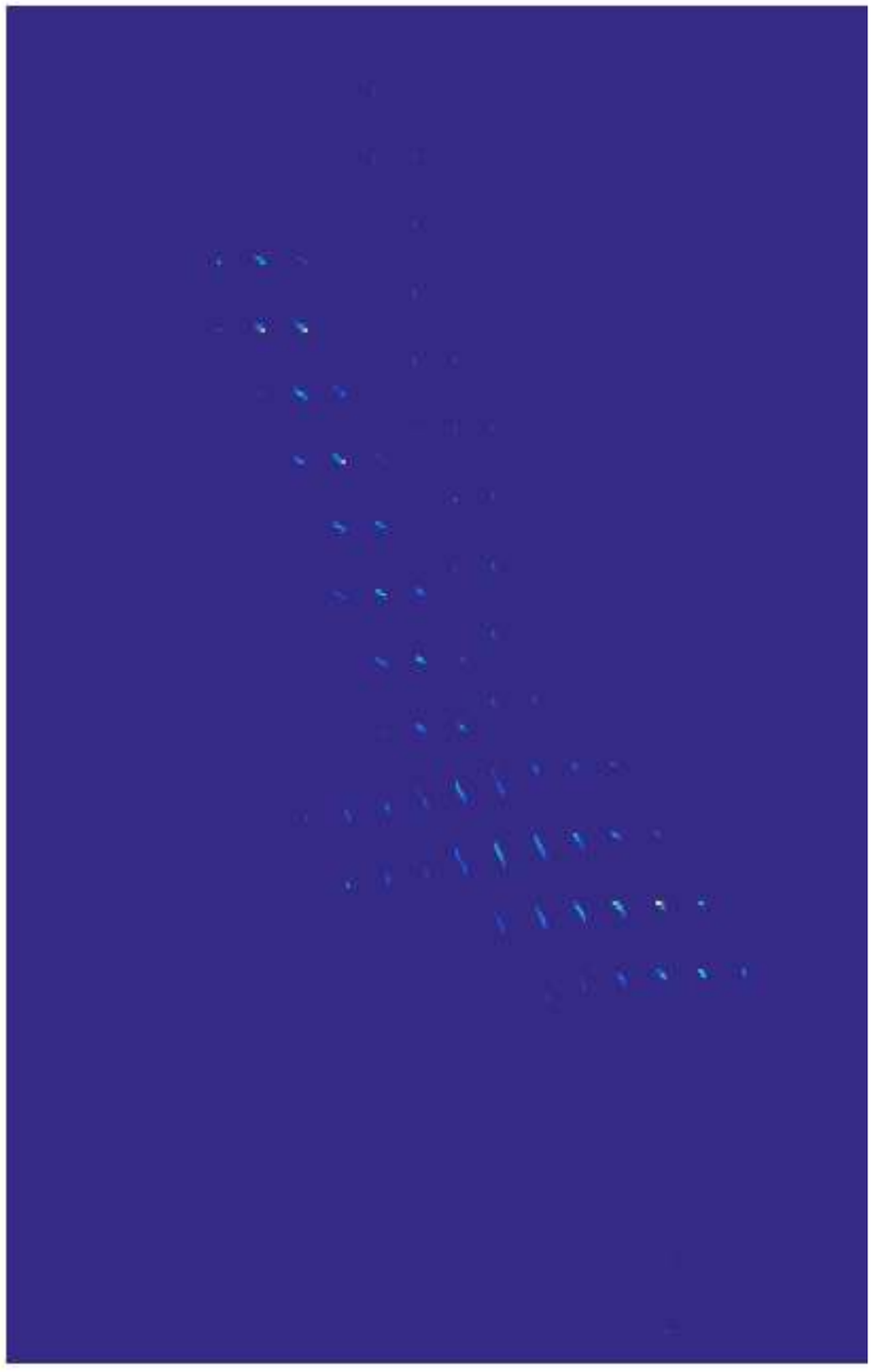


Kv_3

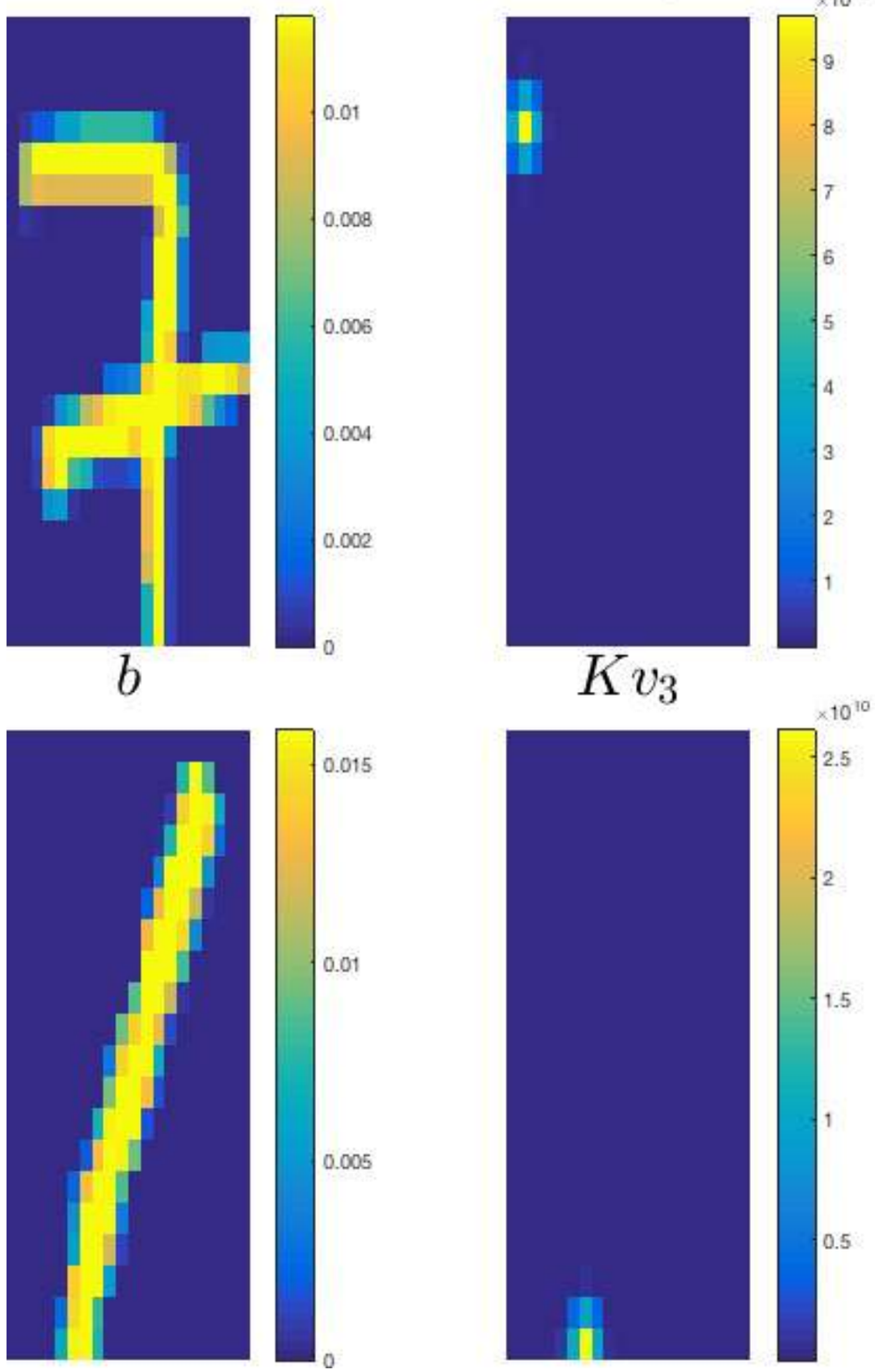


$P_2 = D(u_2)KD(v_2)$

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

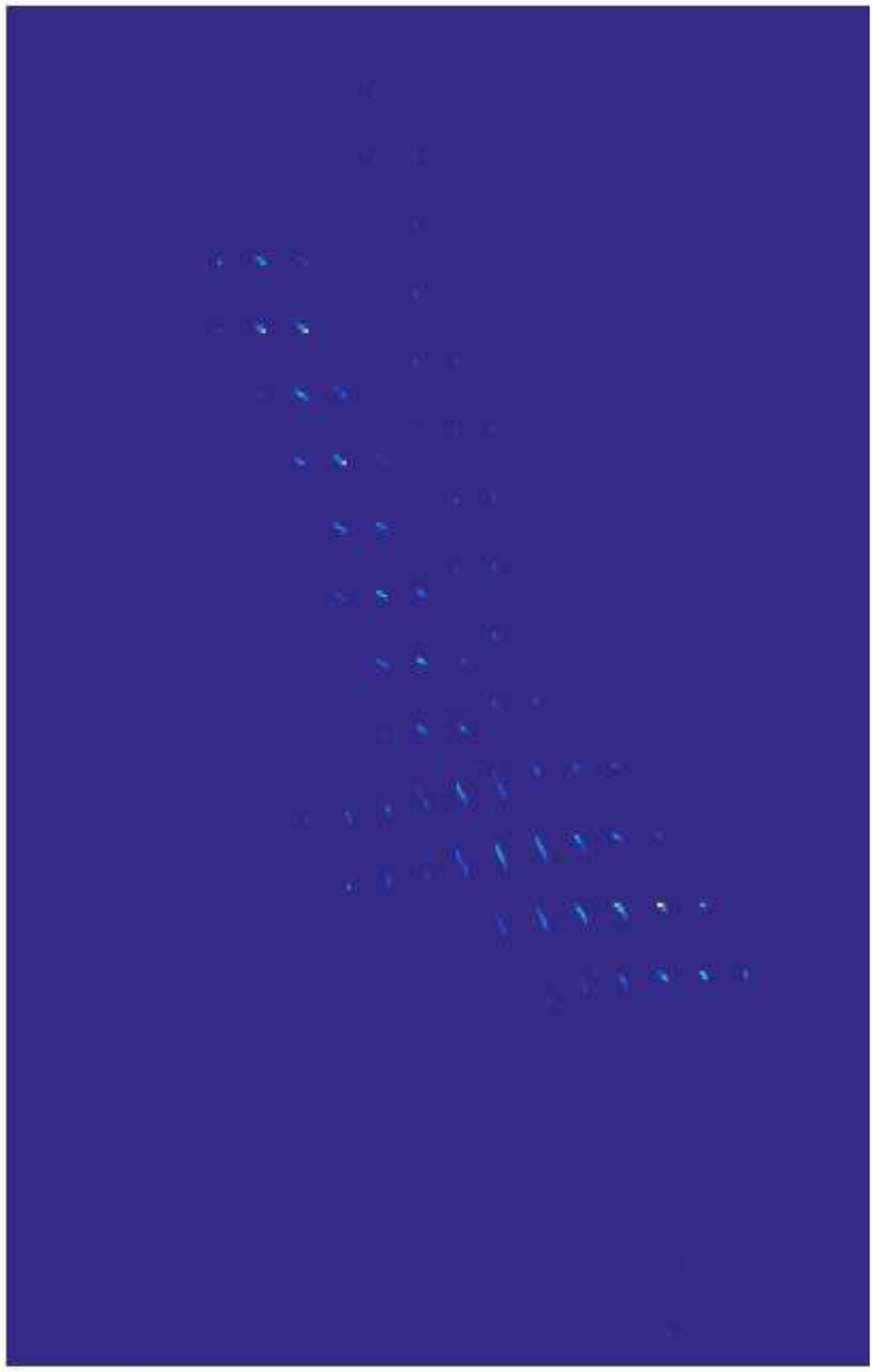


Very Fast EMD Approx. Solver



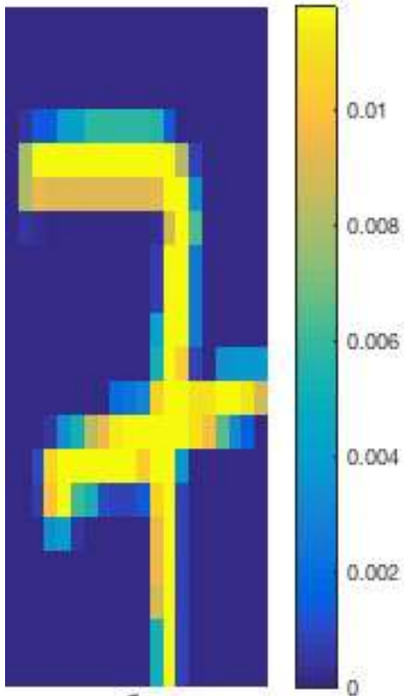
$P_2 = D(u_2)KD(v_2)$

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

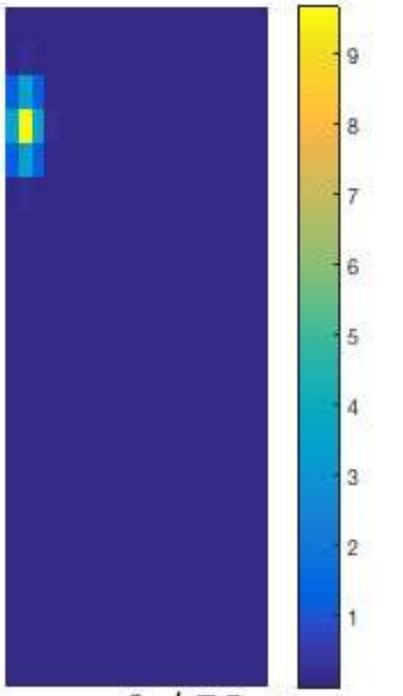


Very Fast EMD Approx. Solver

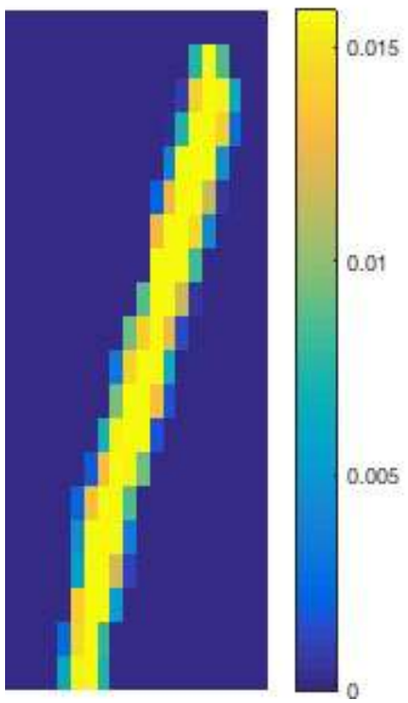
a



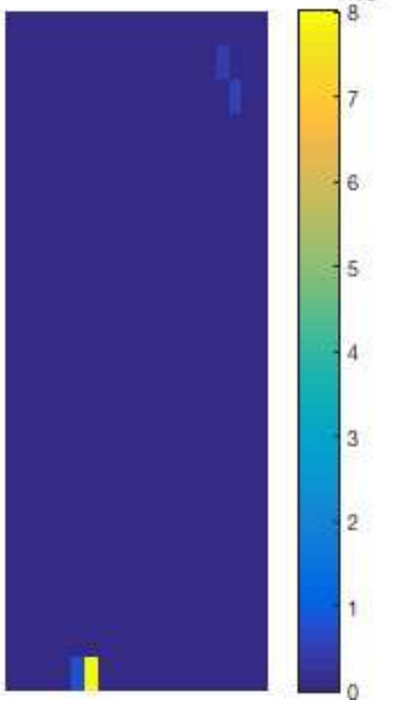
Ku_3



b

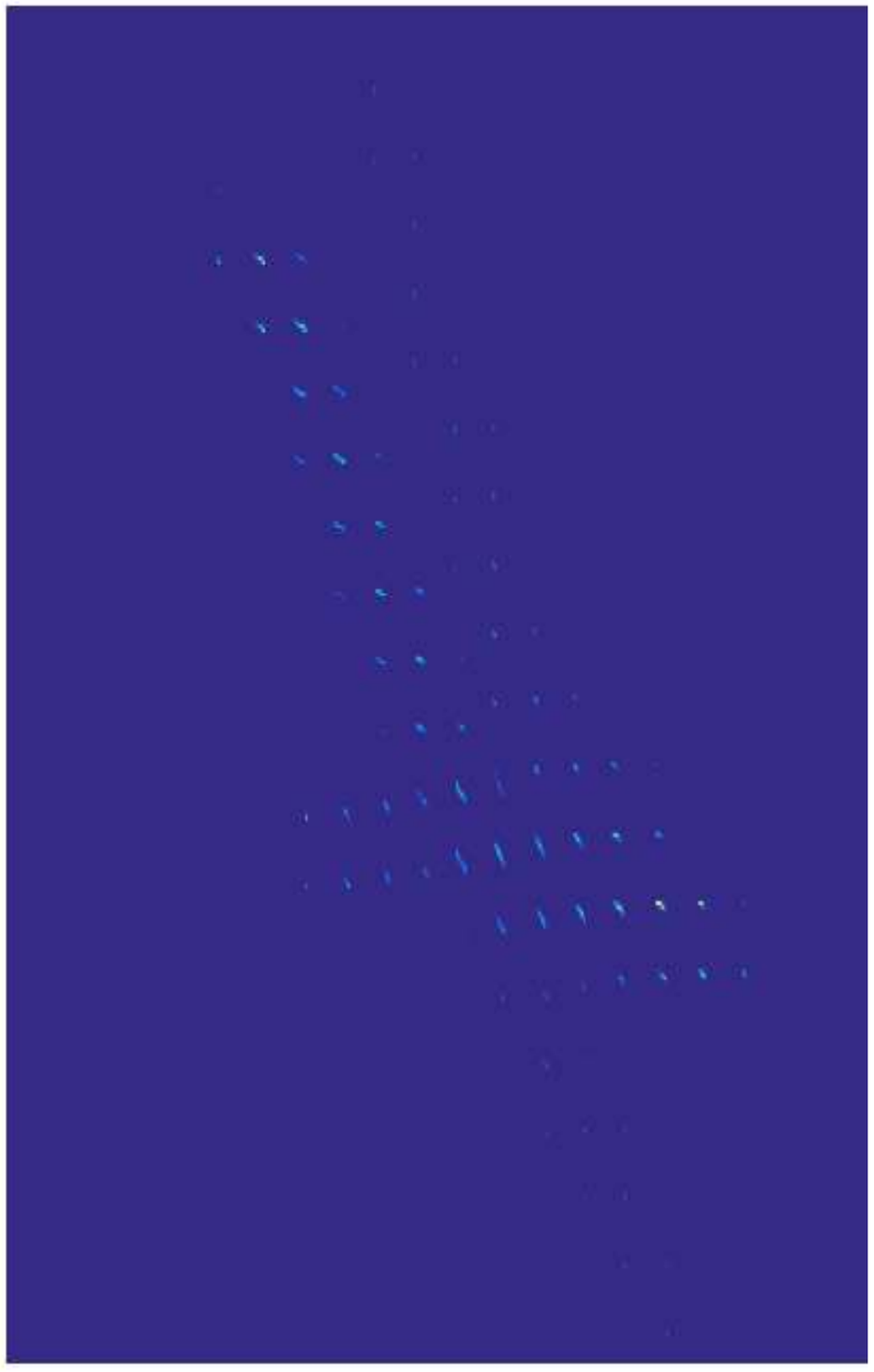


$v_4 \leftarrow b/Ku_4$

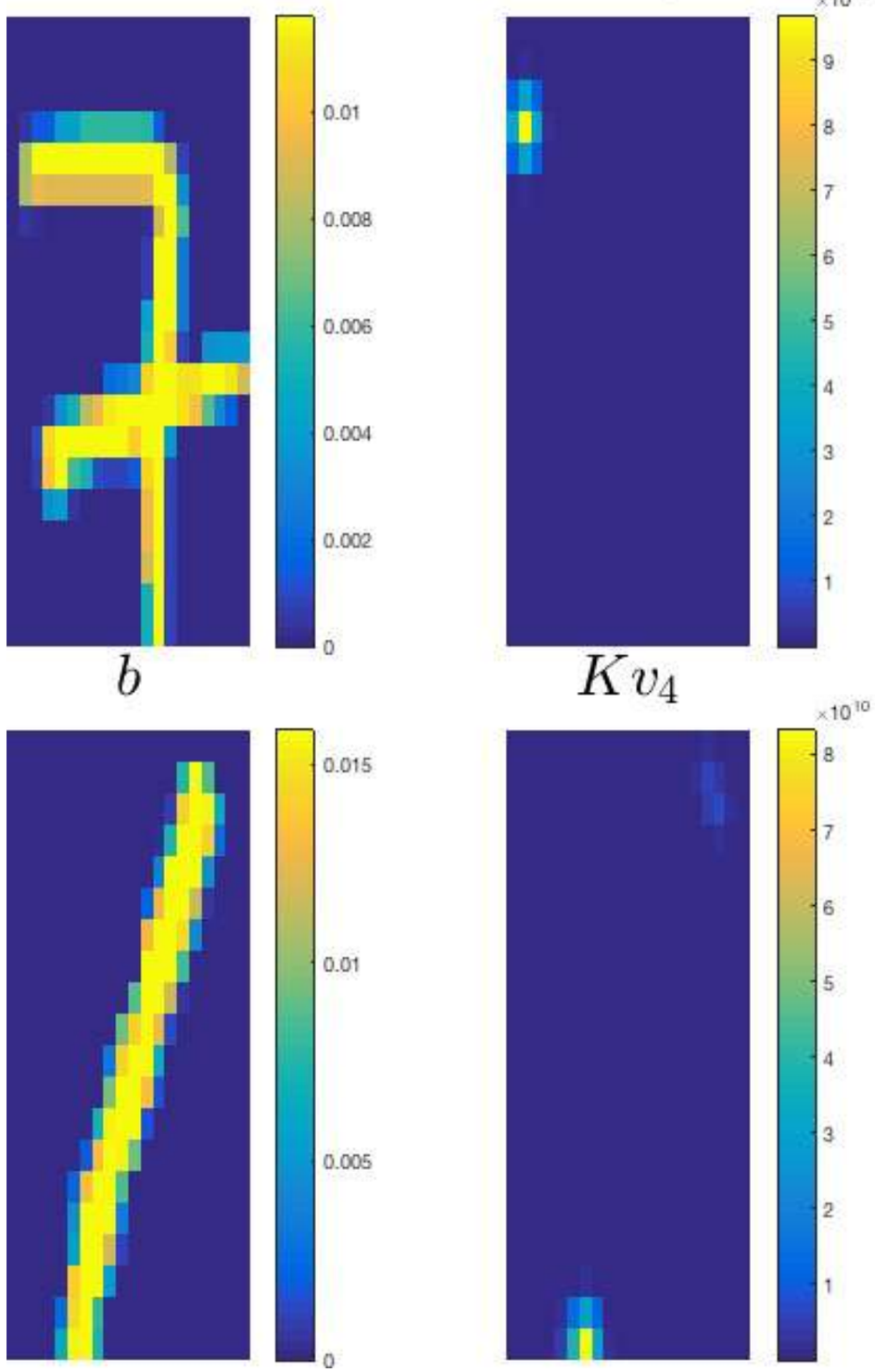


$P_3 = D(u_3)KD(v_3)$

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

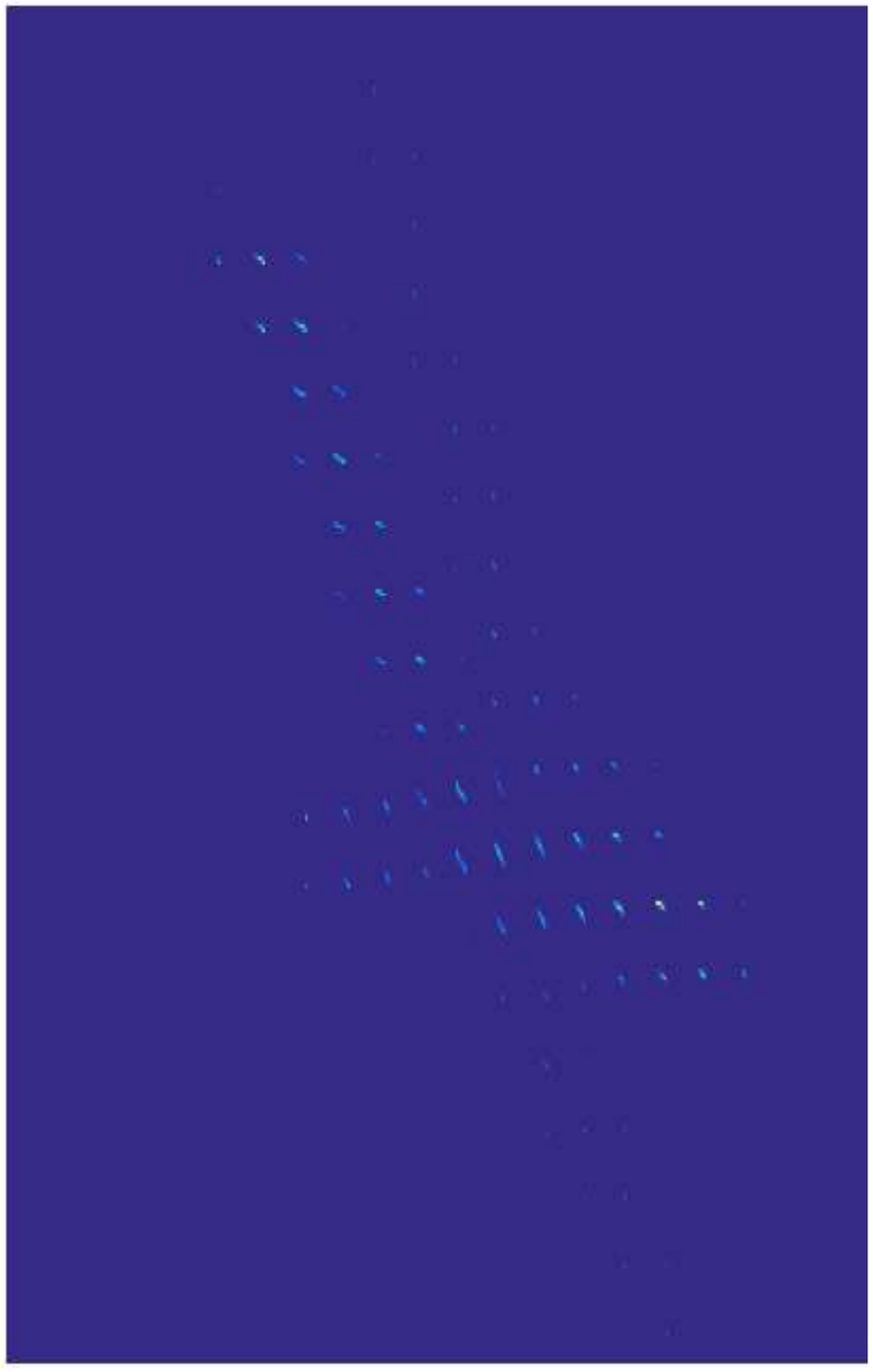


Very Fast EMD Approx. Solver



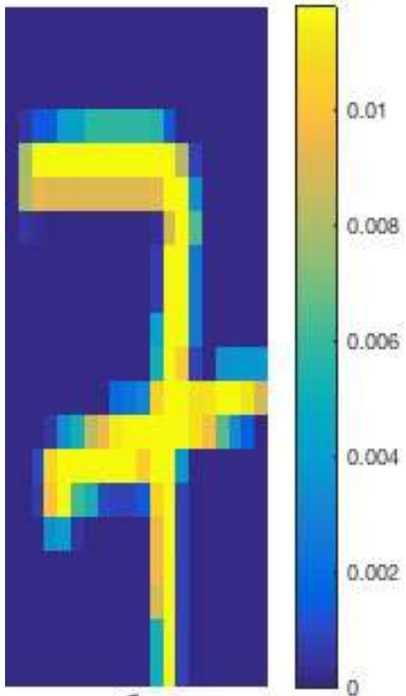
$P_3 = D(u_3)KD(v_3)$

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

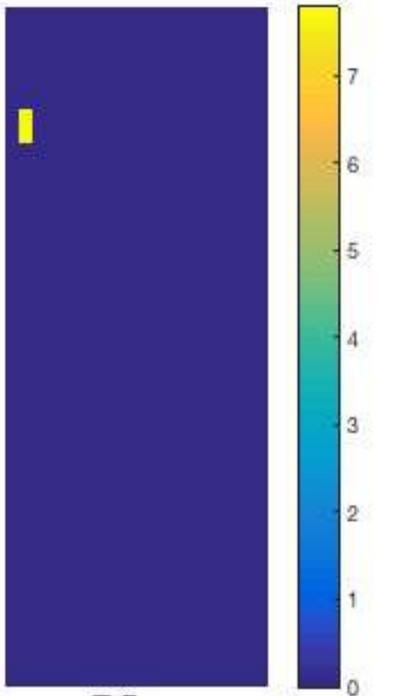


Very Fast EMD Approx. Solver

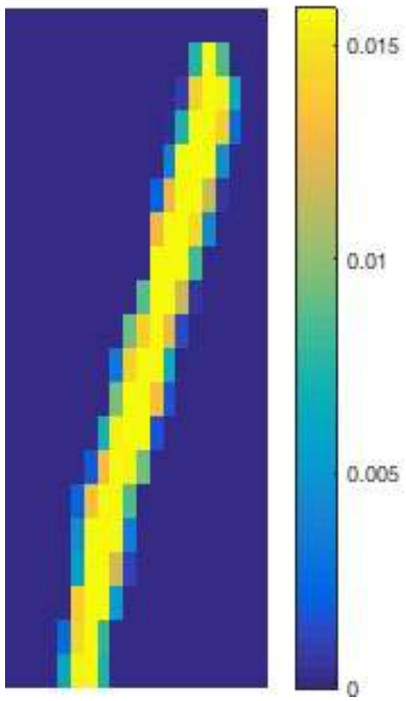
a



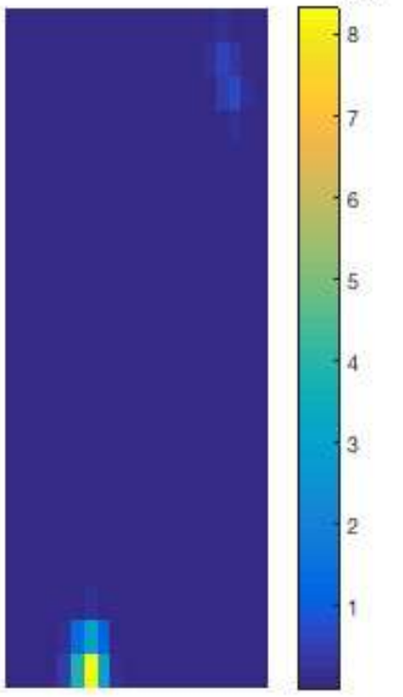
$u_4 \leftarrow a/Kv_4$



b

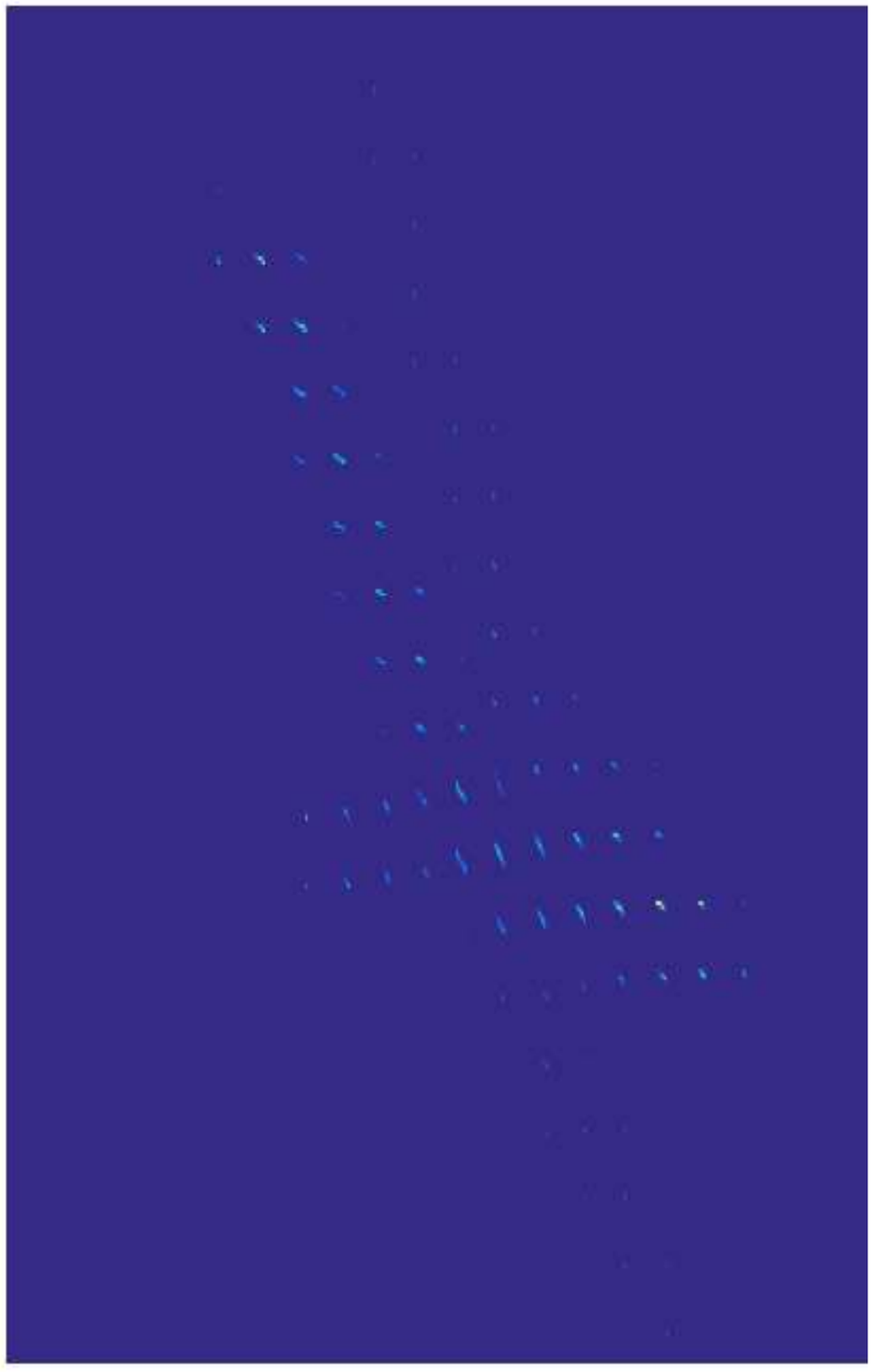


Kv_4

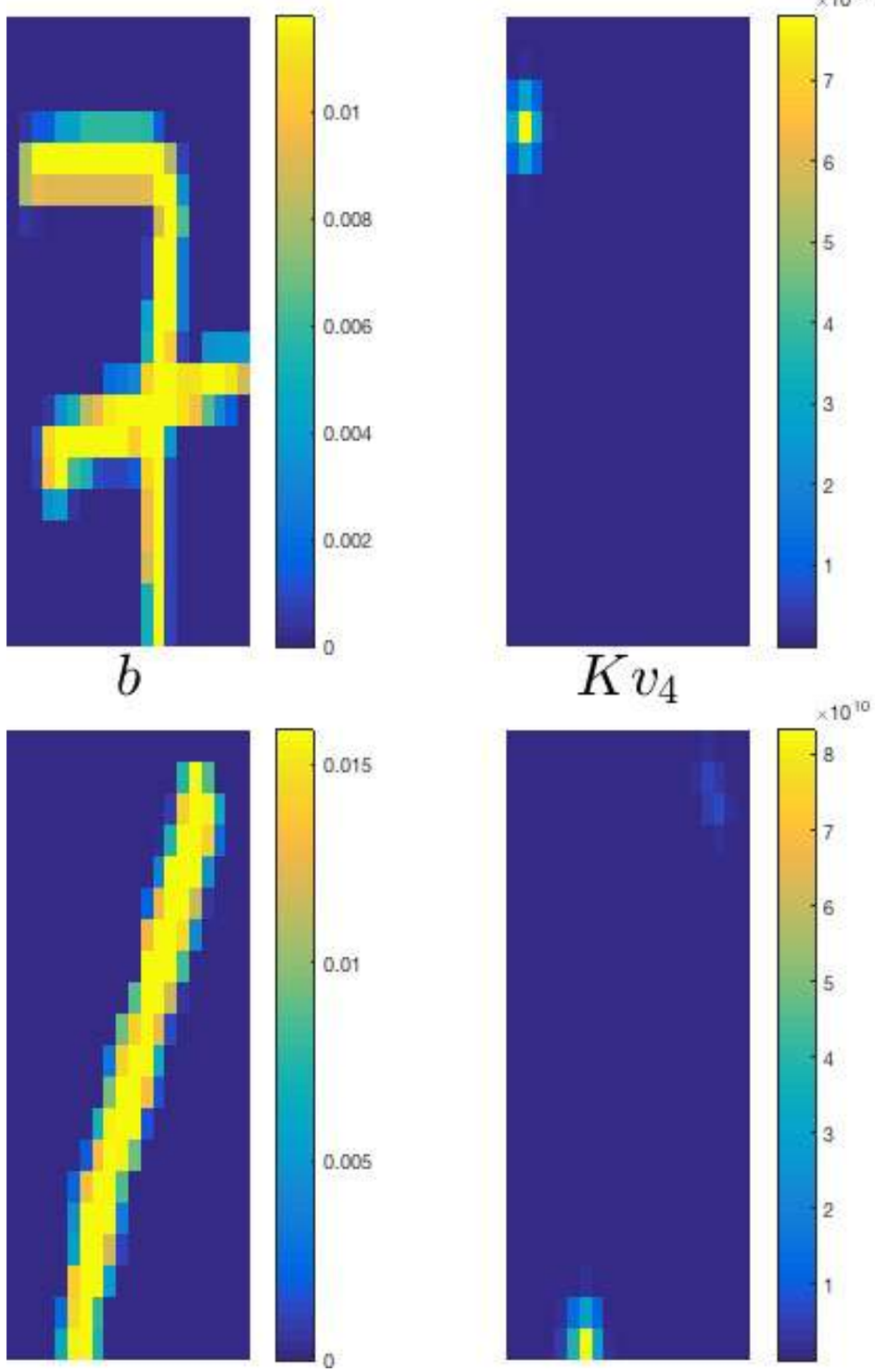


$P_3 = D(u_3)KD(v_3)$

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

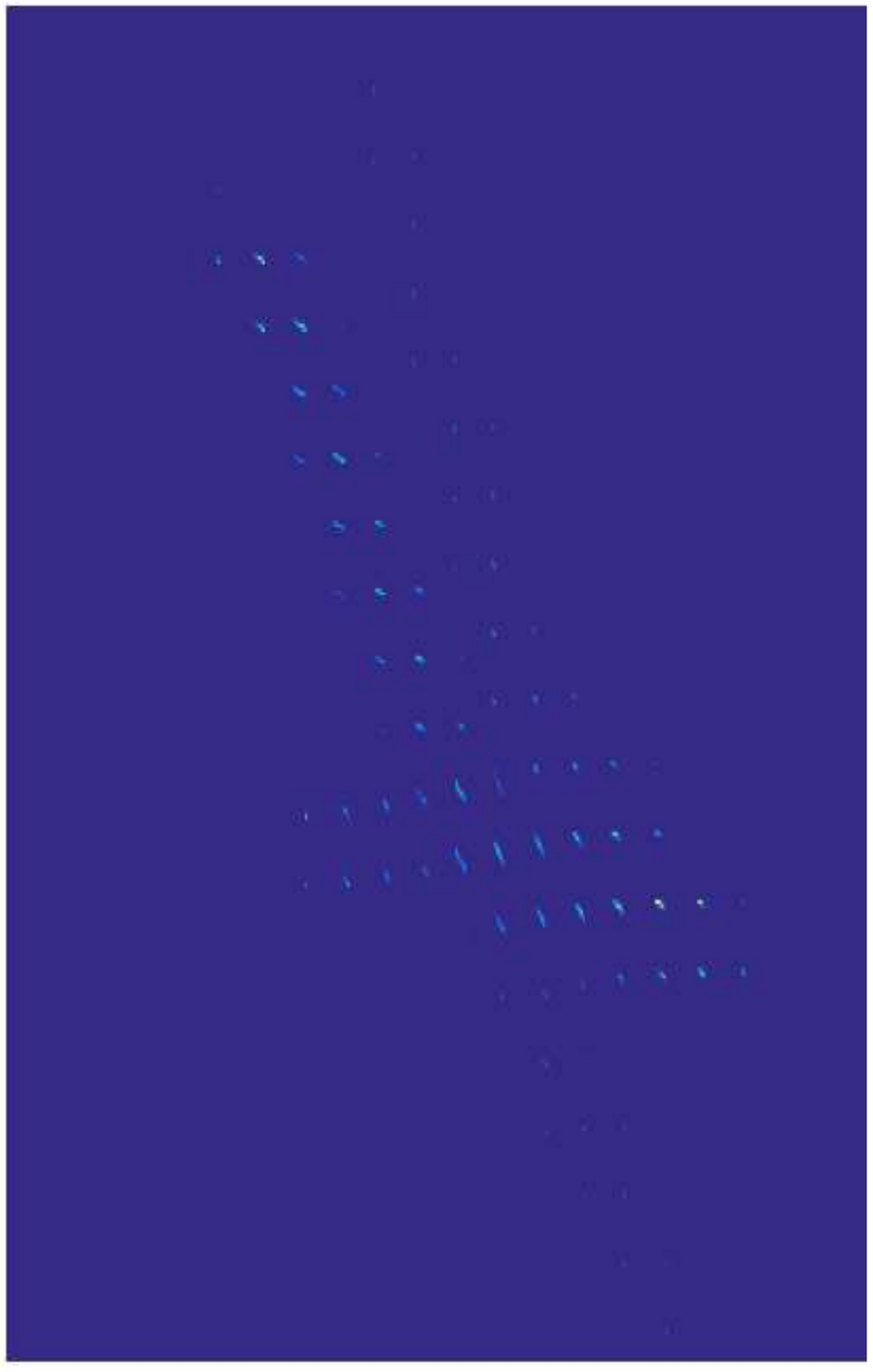


Very Fast EMD Approx. Solver



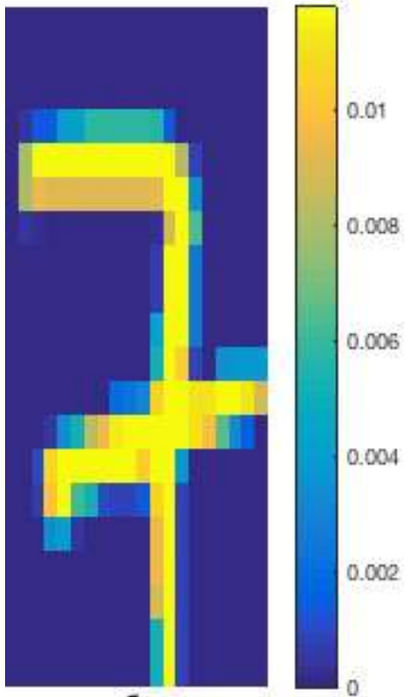
$P_3 = D(u_3)KD(v_3)$

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

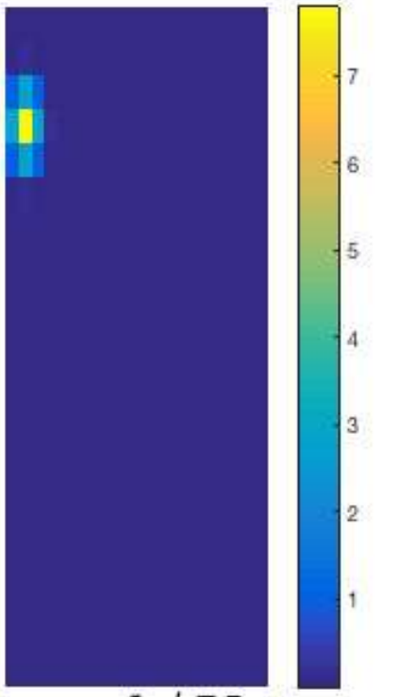


Very Fast EMD Approx. Solver

a

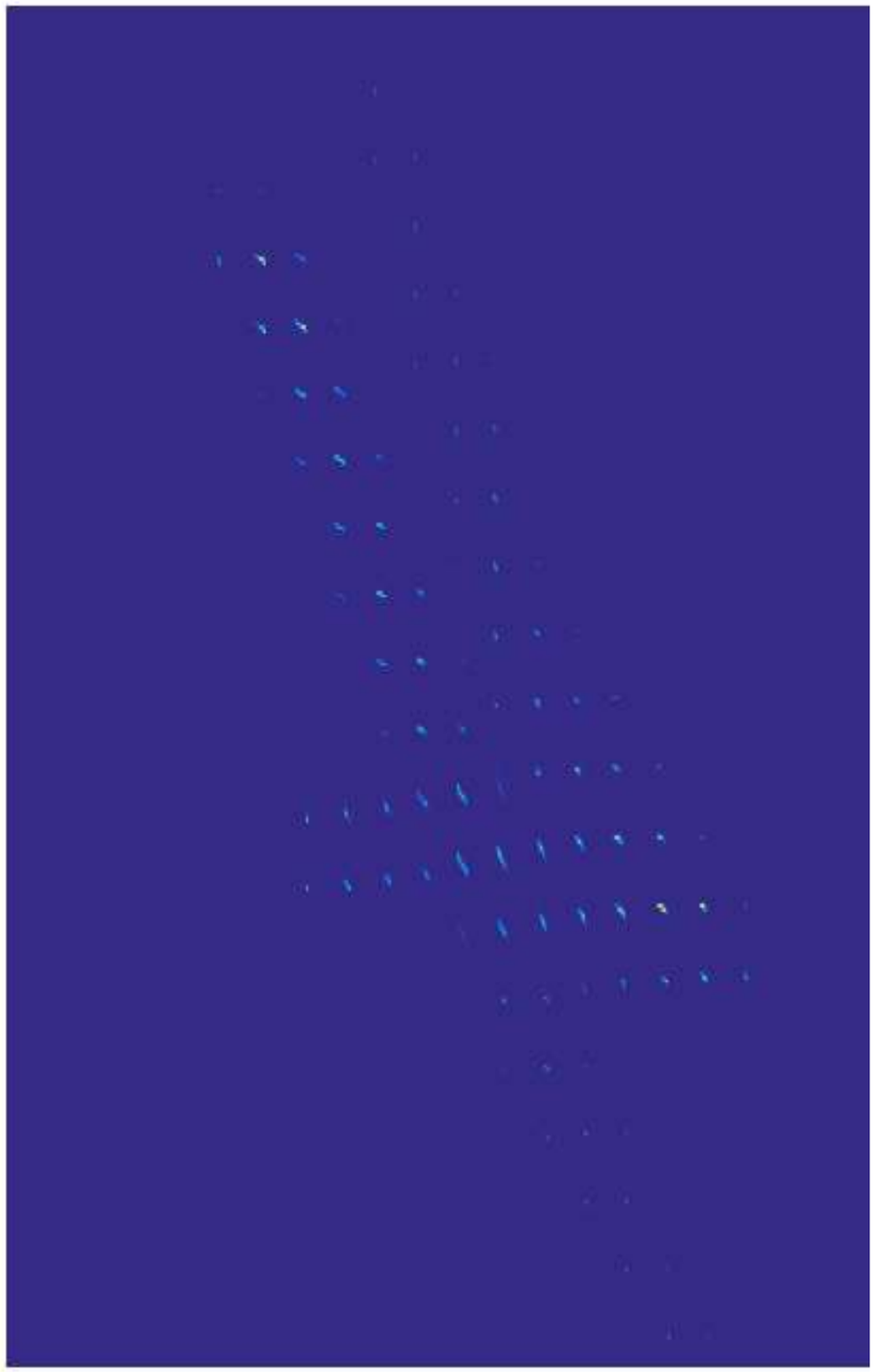


Ku_4

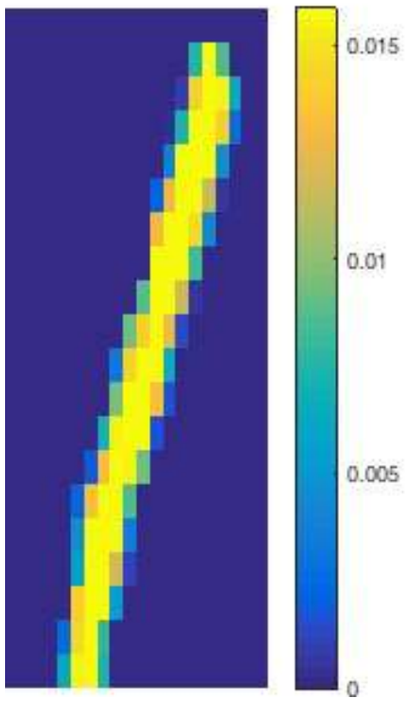


$P_4 = D(u_4)KD(v_4)$

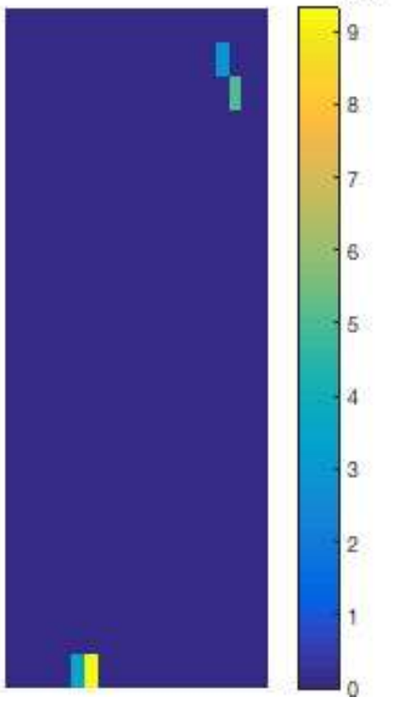
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$



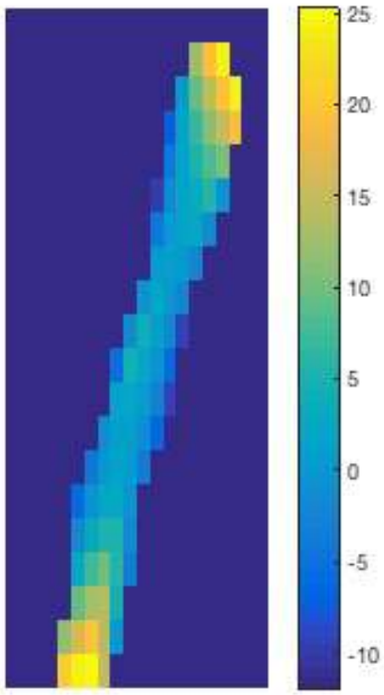
b



$v_5 \leftarrow b/Ku_5$

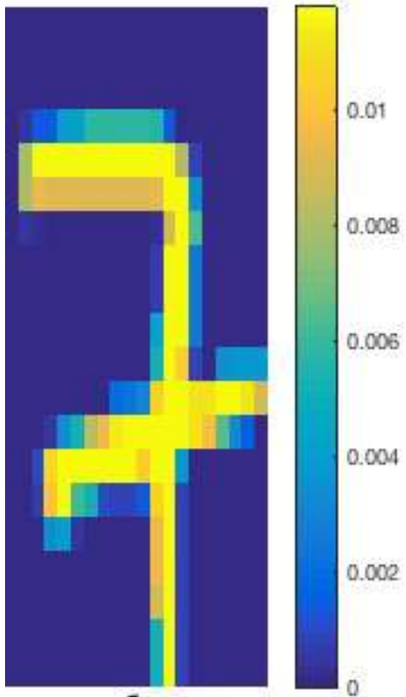


$\log(v_5)$

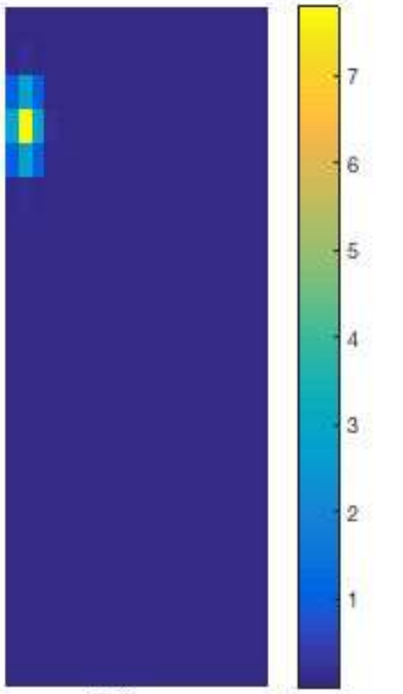


Very Fast EMD Approx. Solver

a

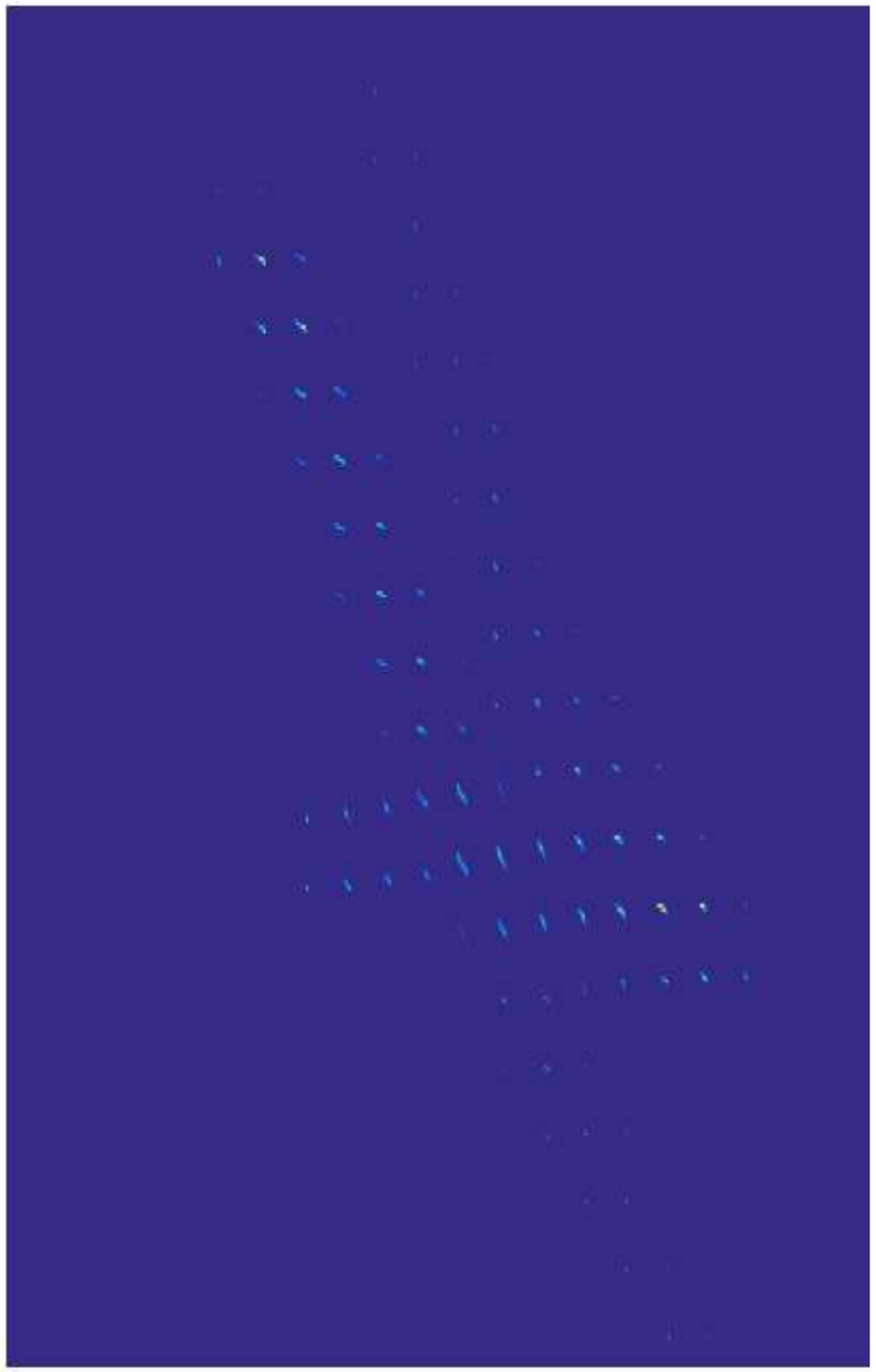


Ku_4

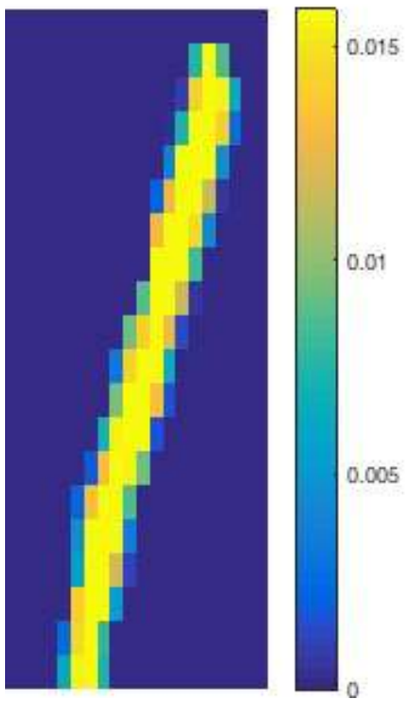


$P_4 = D(u_4)KD(v_4)$

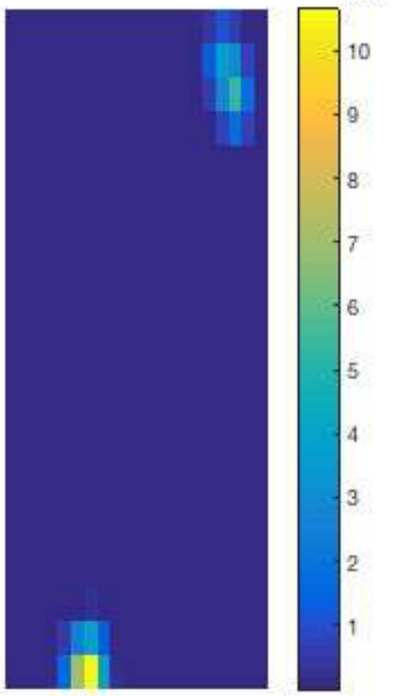
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$



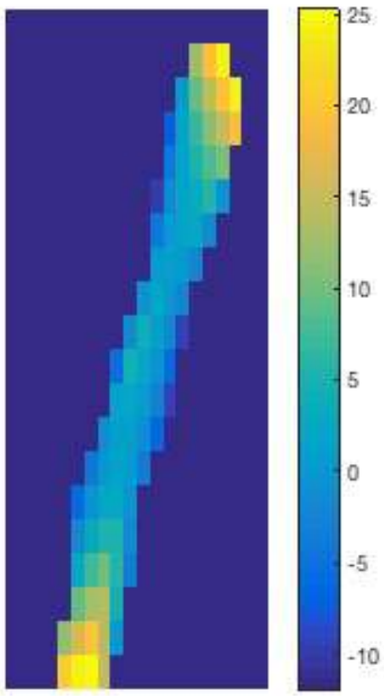
b



Kv_5

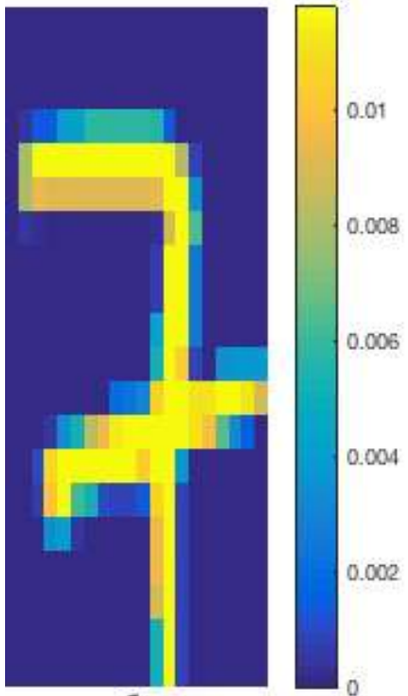


$\log(v_5)$

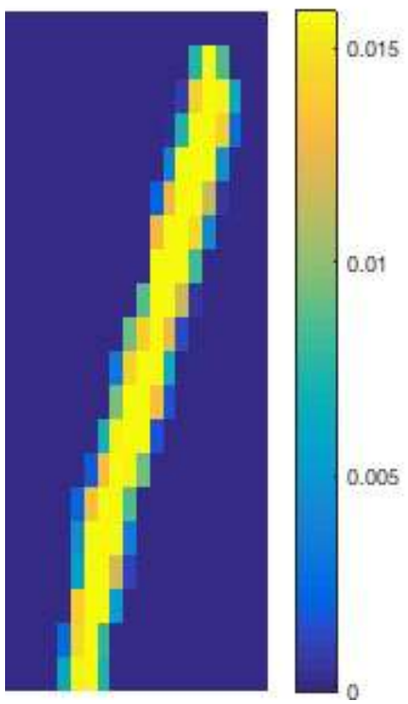


Very Fast EMD Approx. Solver

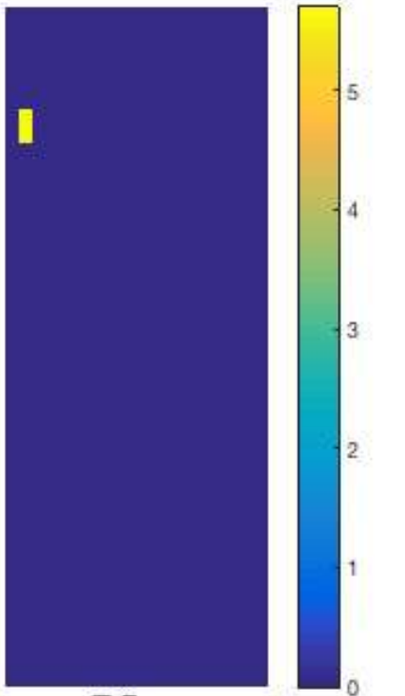
a



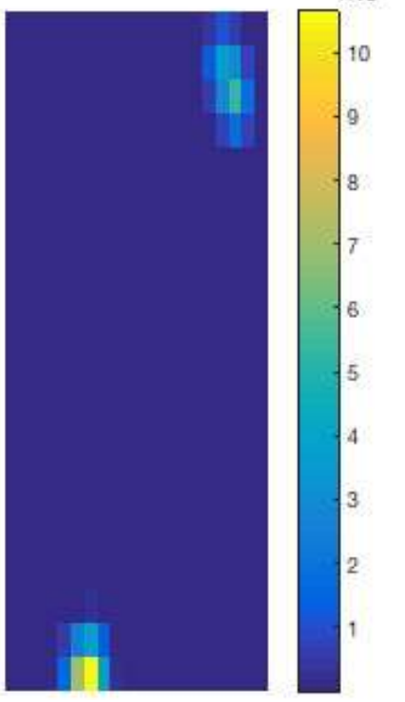
b



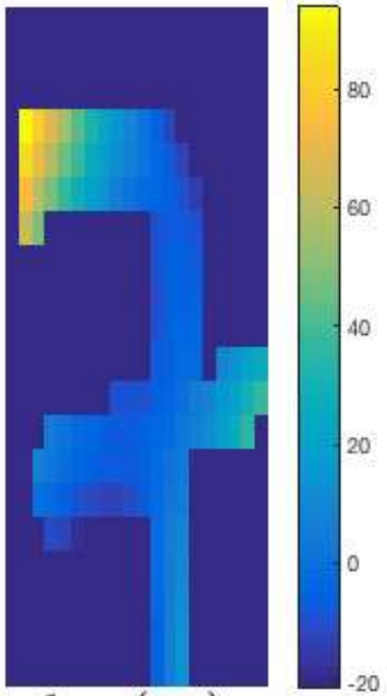
$u_5 \leftarrow a/Kv_5$



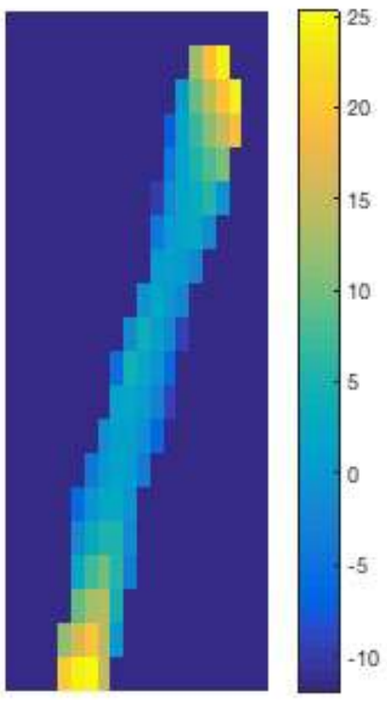
Kv_5



$\log(u_5)$

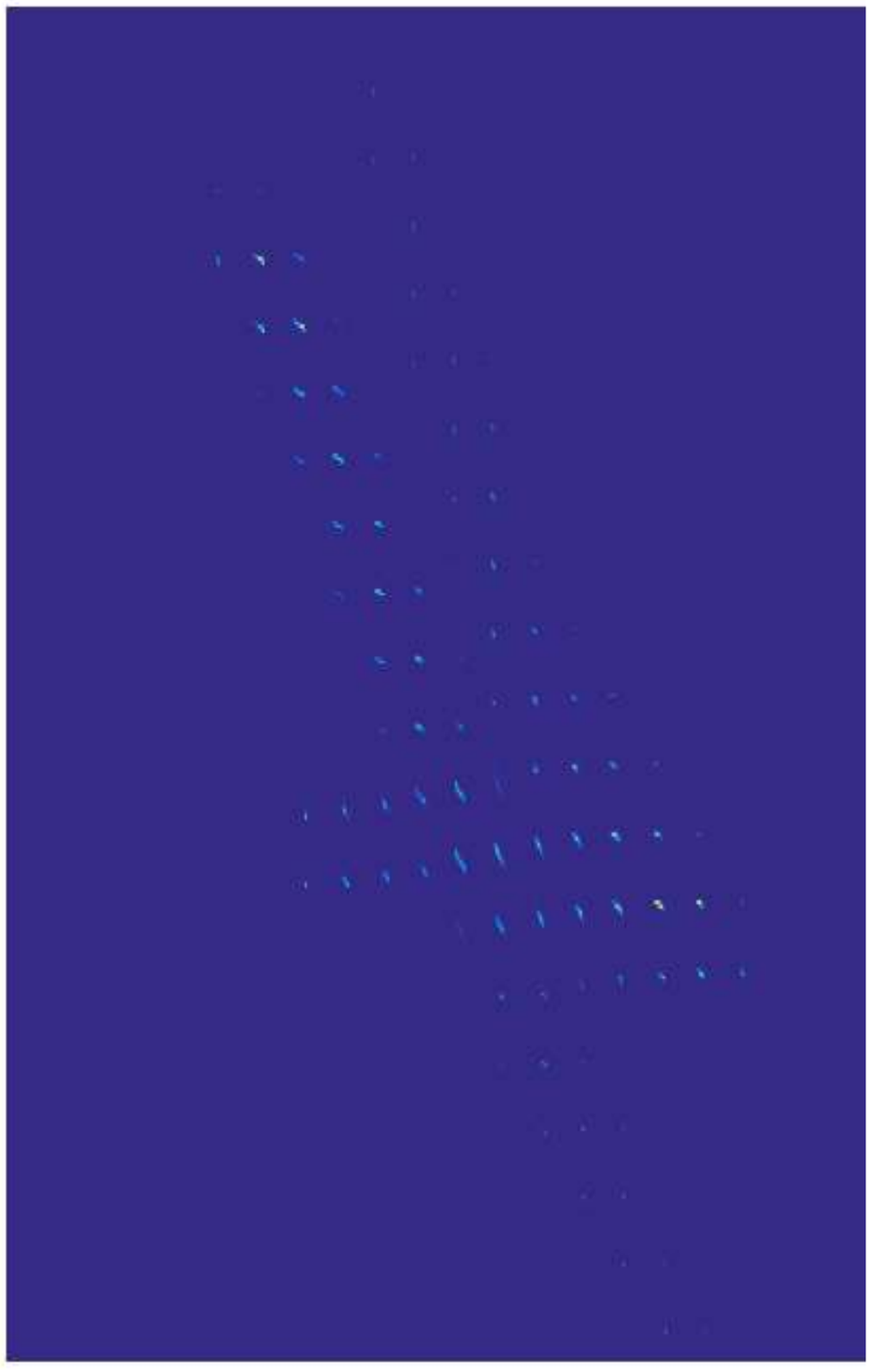


$\log(v_5)$

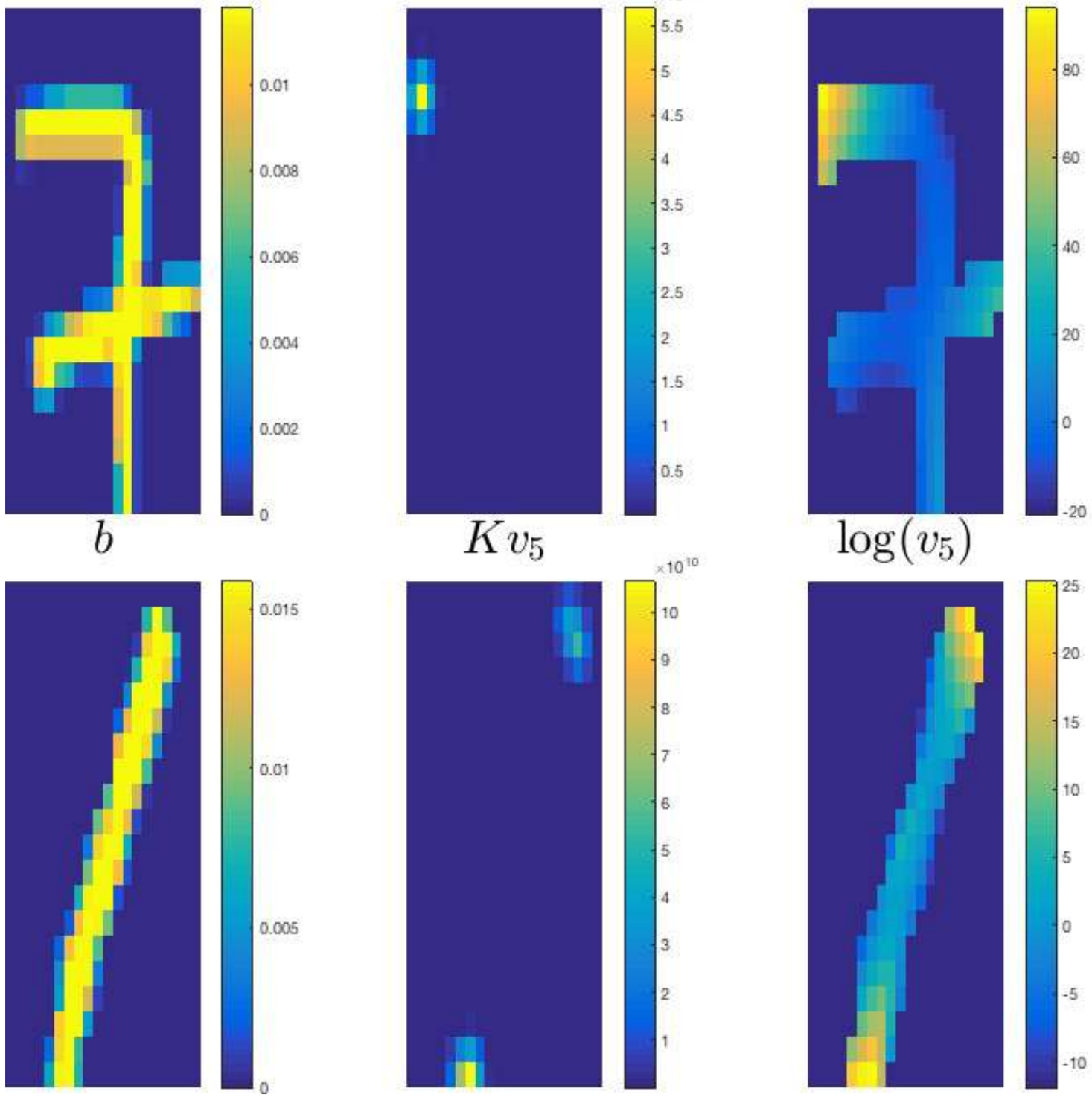


$P_4 = D(u_4)KD(v_4)$

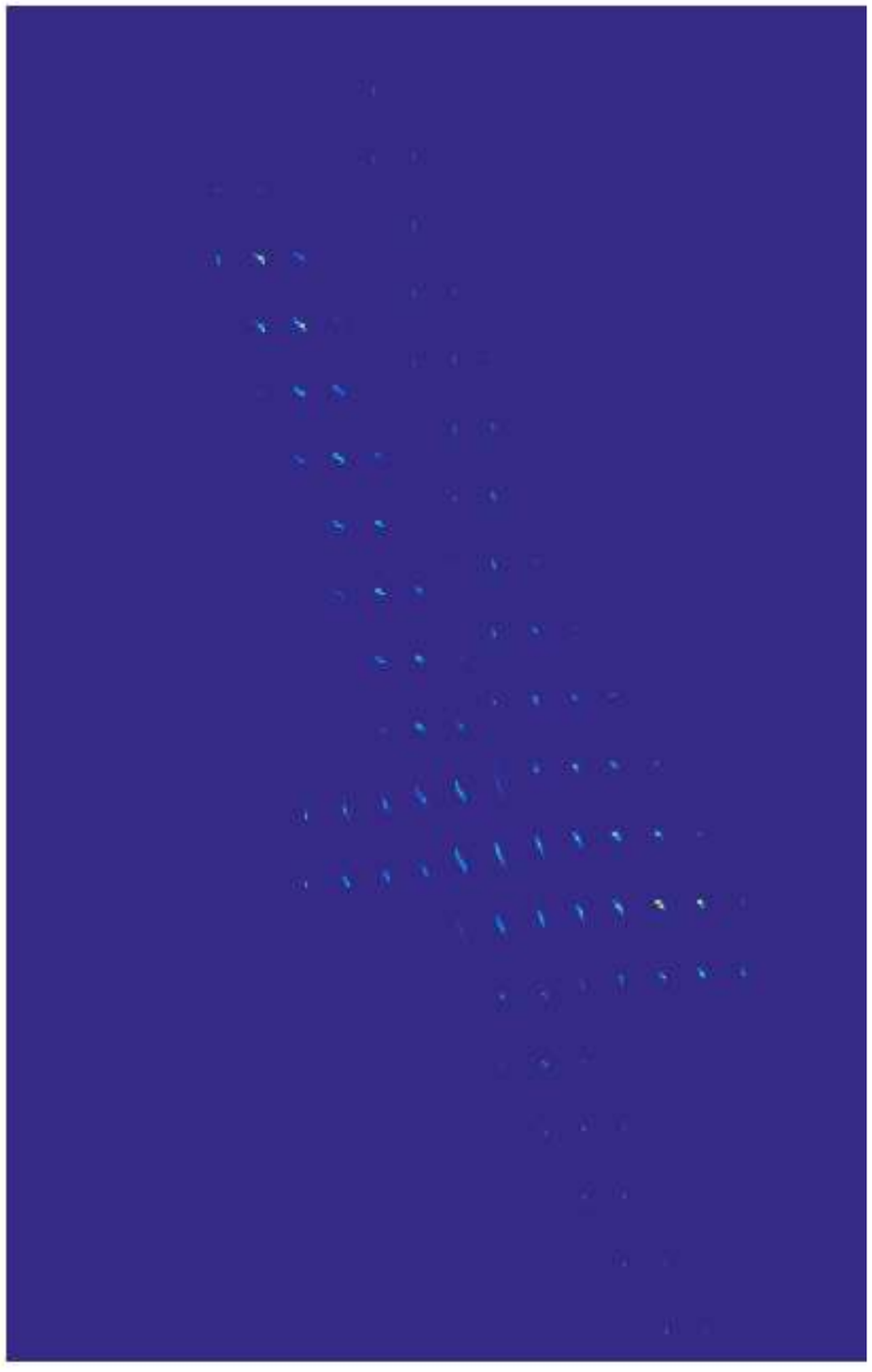
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$



Very Fast EMD Approx. Solver

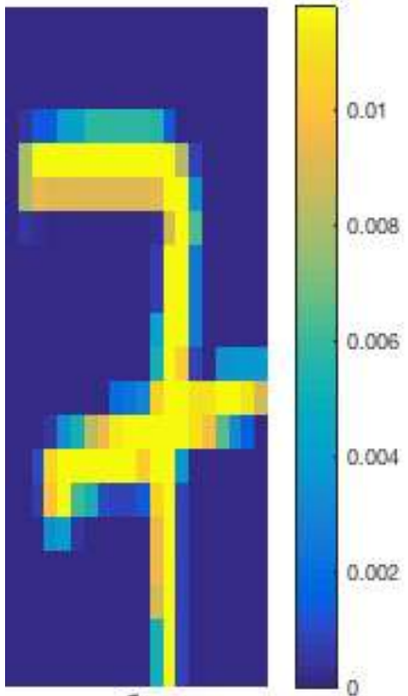


$$P_4 = D(u_4)KD(v_4)$$
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

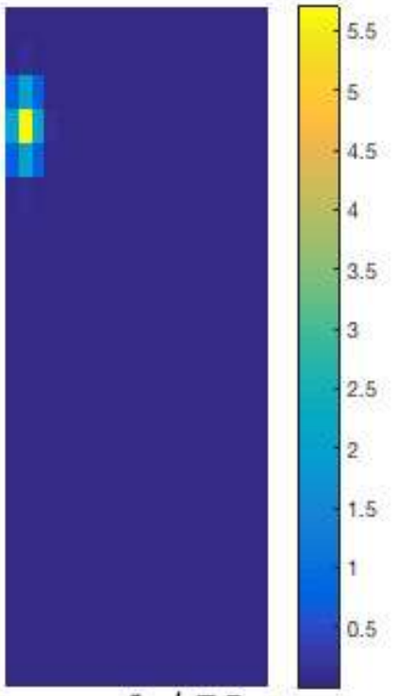


Very Fast EMD Approx. Solver

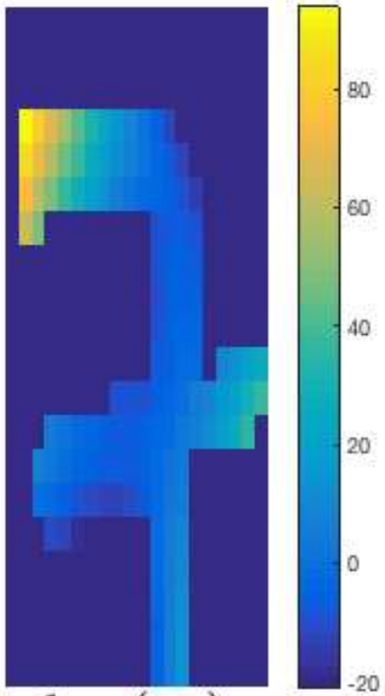
a



Ku_5

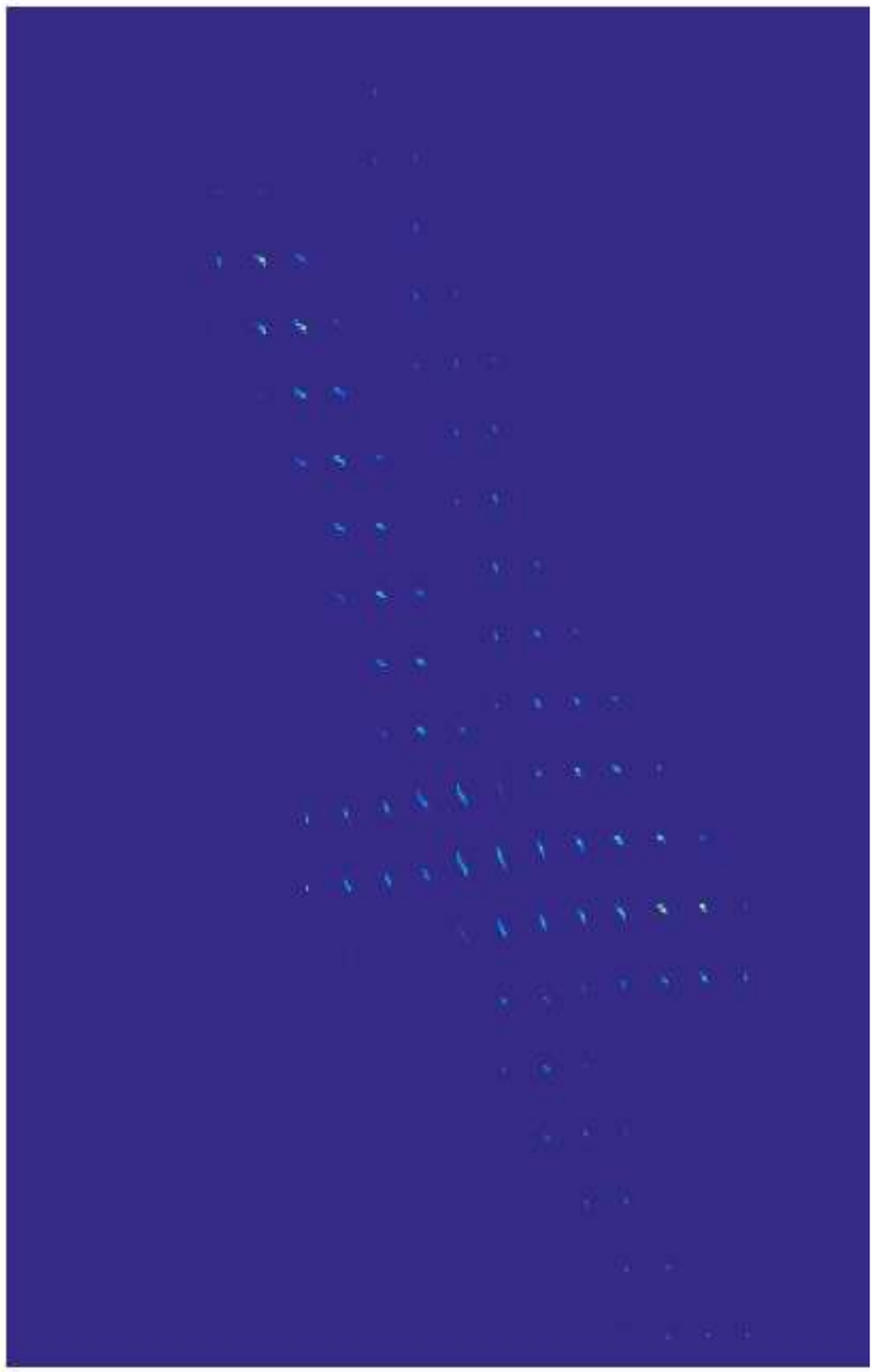


$\log(u_5)$

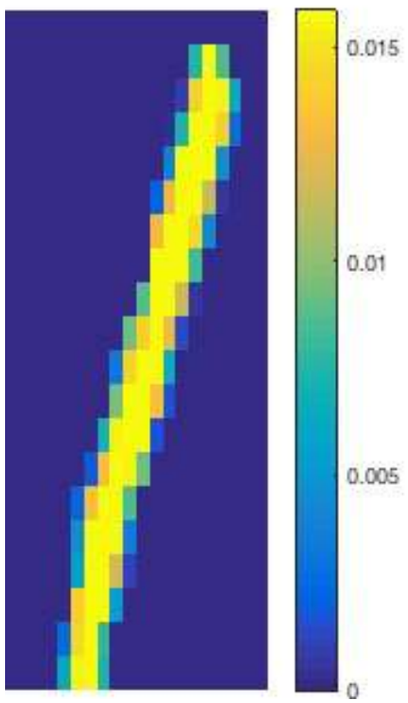


$P_5 = D(u_5)KD(v_5)$

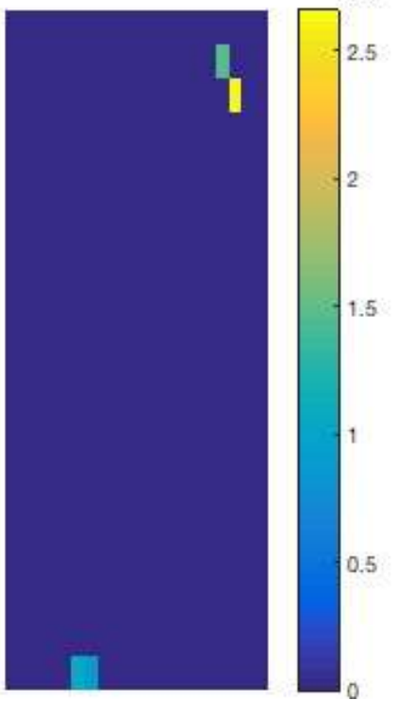
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$



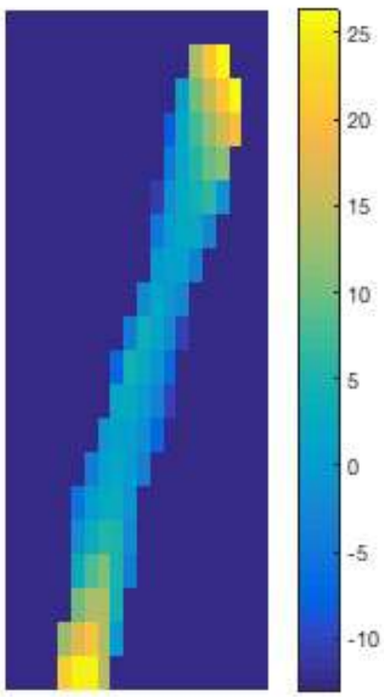
b



$v_6 \leftarrow b/Ku_6$

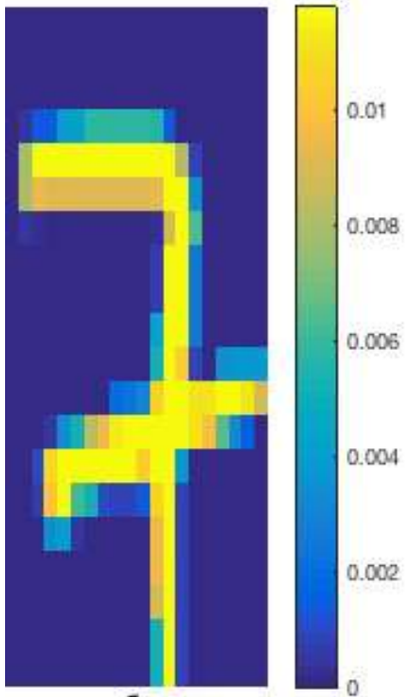


$\log(v_6)$

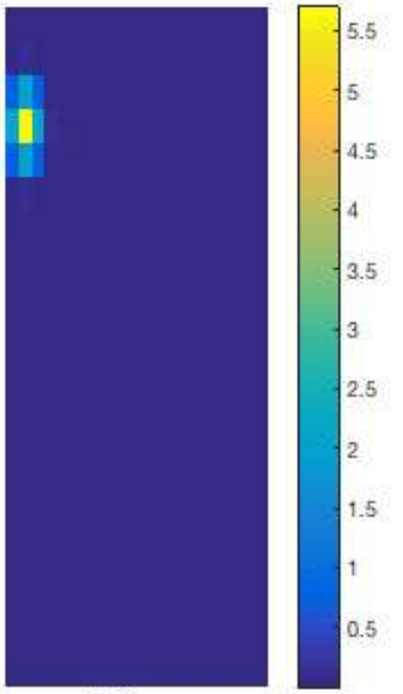


Very Fast EMD Approx. Solver

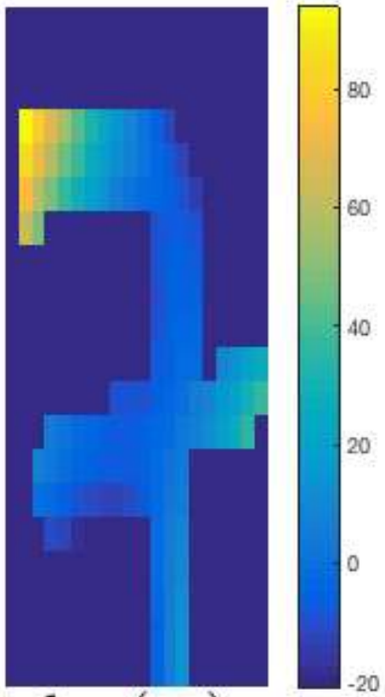
a



Ku_5

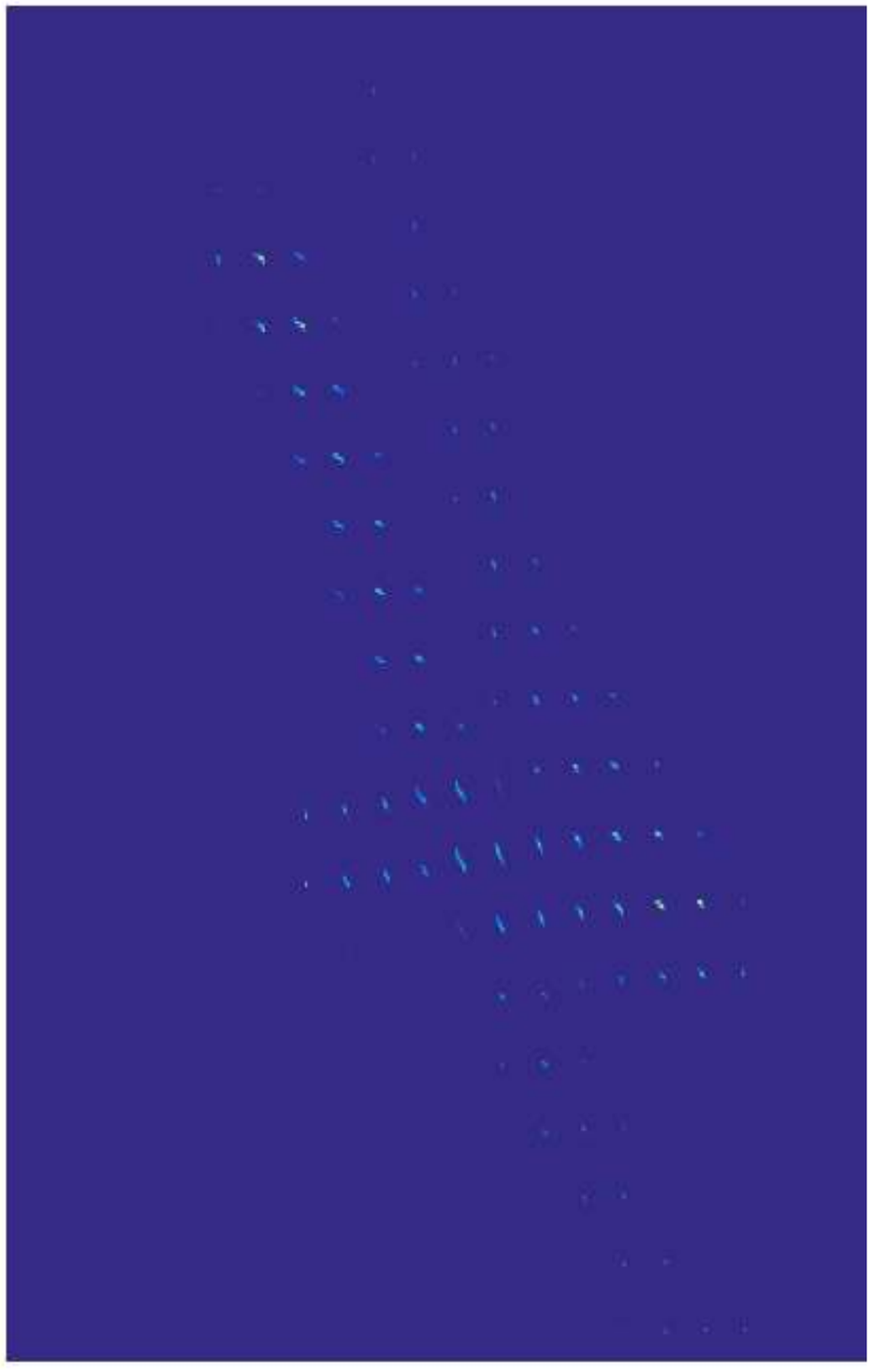


$\log(u_5)$

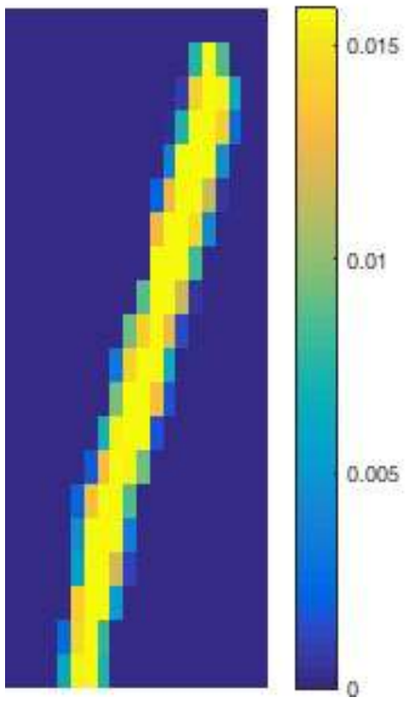


$P_5 = D(u_5)KD(v_5)$

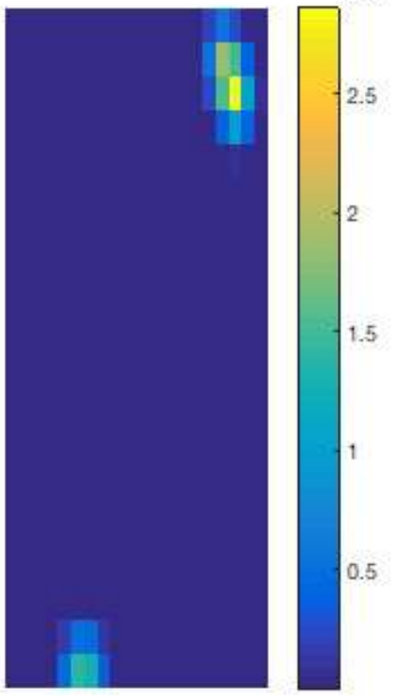
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$



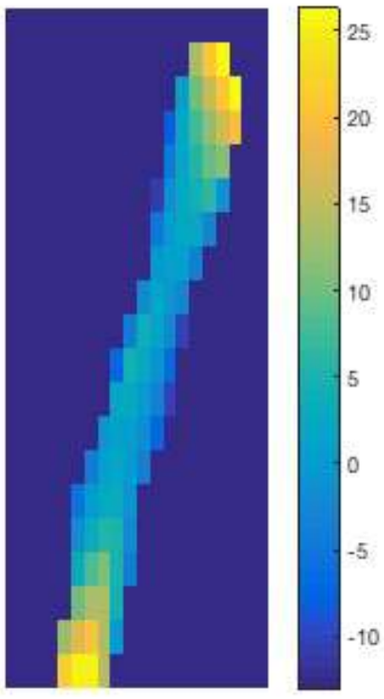
b



Kv_6

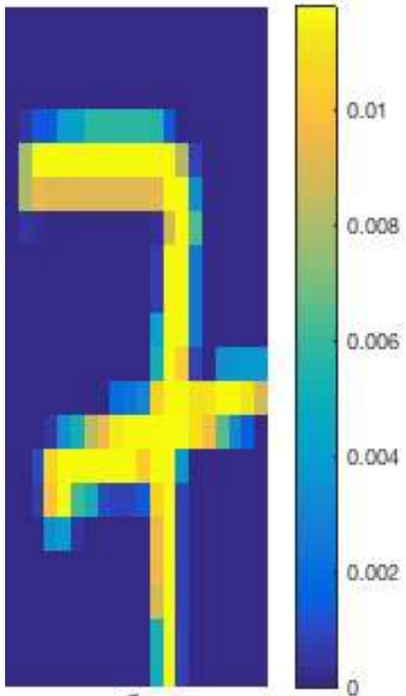


$\log(v_6)$

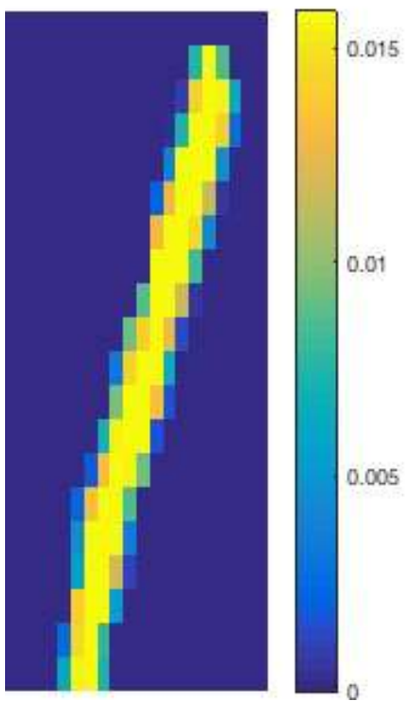


Very Fast EMD Approx. Solver

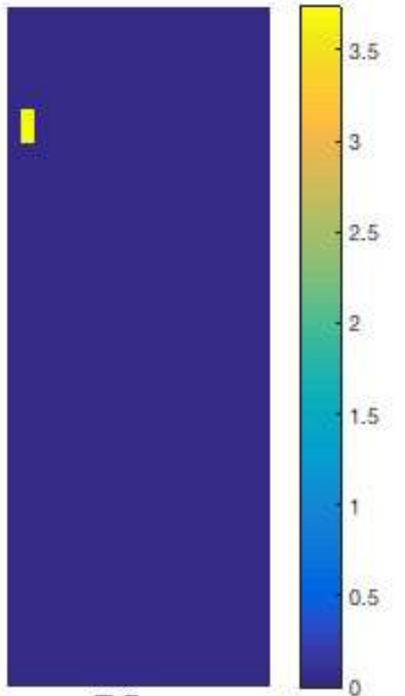
a



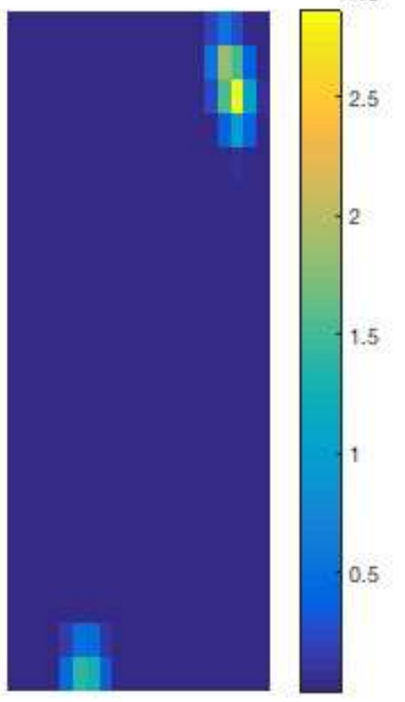
b



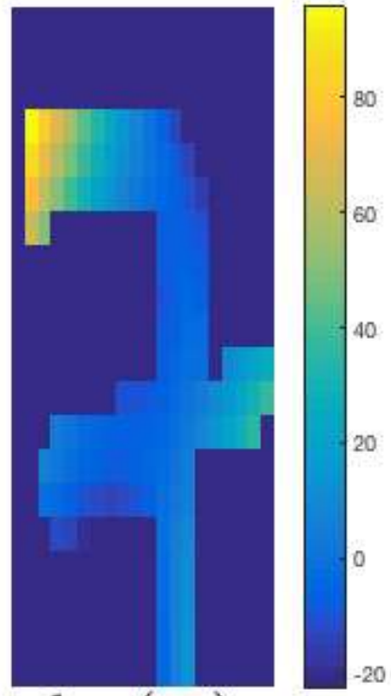
$u_6 \leftarrow a/Kv_6$



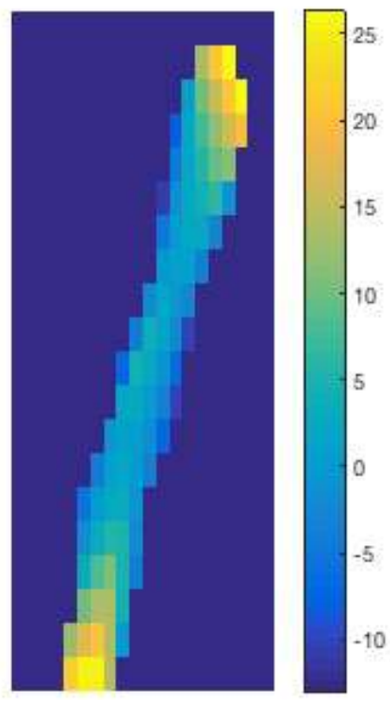
Kv_6



$\log(u_6)$

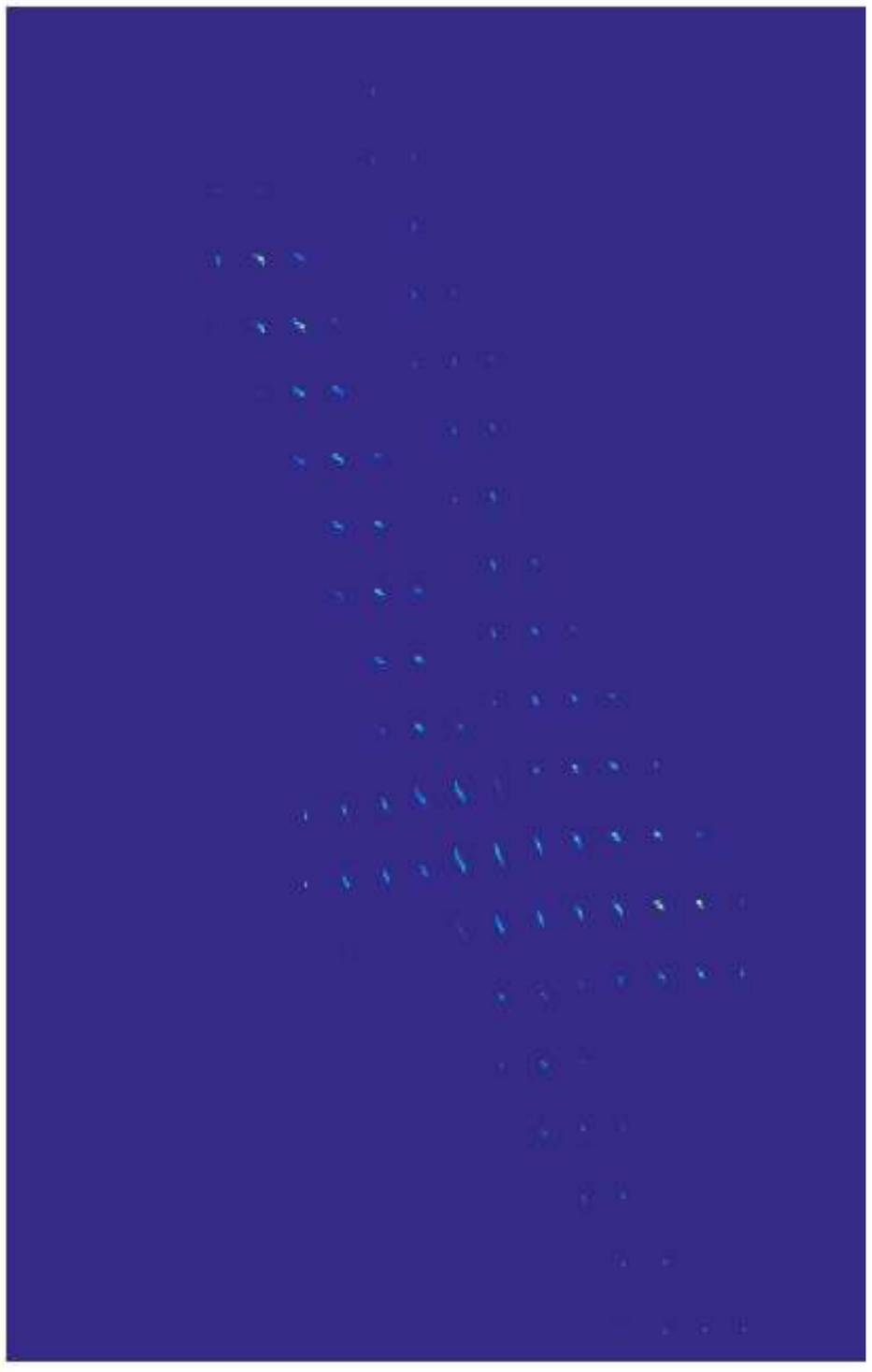


$\log(v_6)$



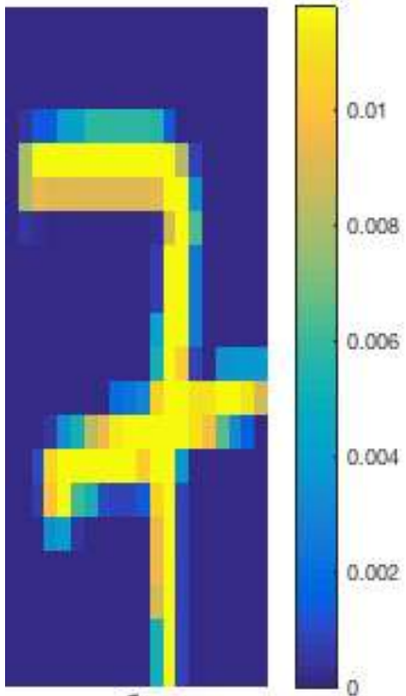
$P_5 = D(u_5)KD(v_5)$

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

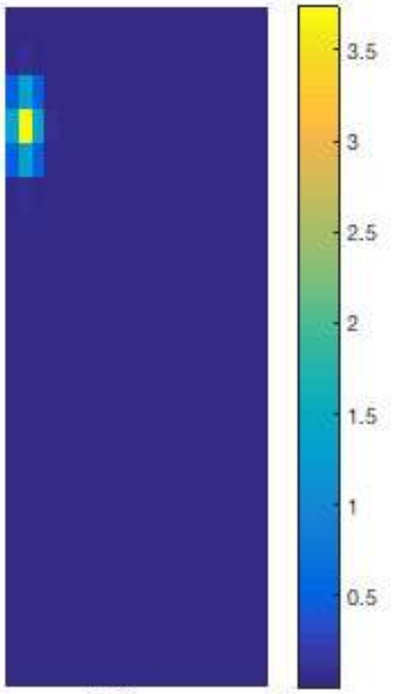


Very Fast EMD Approx. Solver

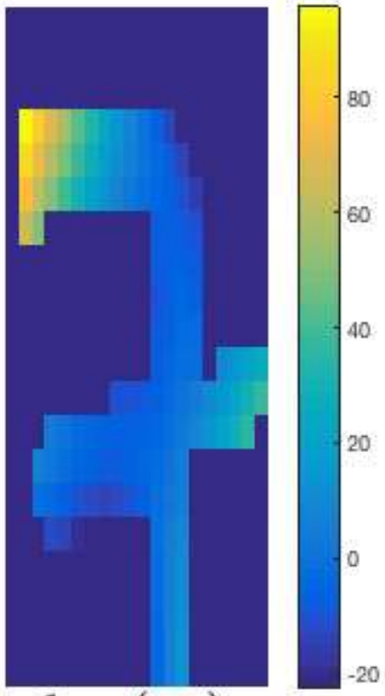
a



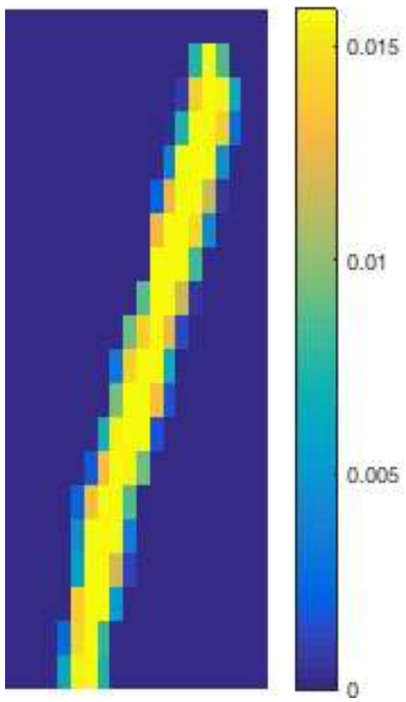
Ku_6



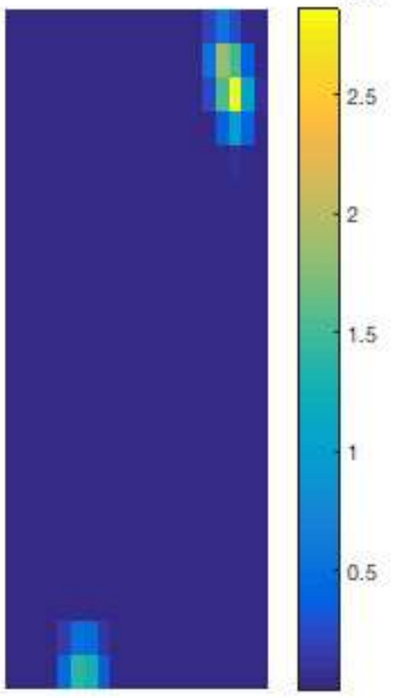
$\log(u_6)$



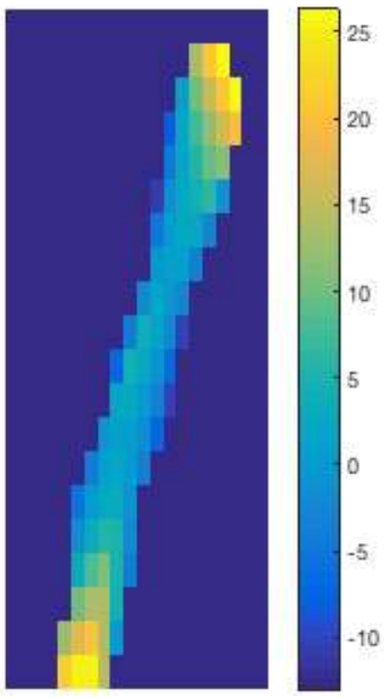
b



Kv_6

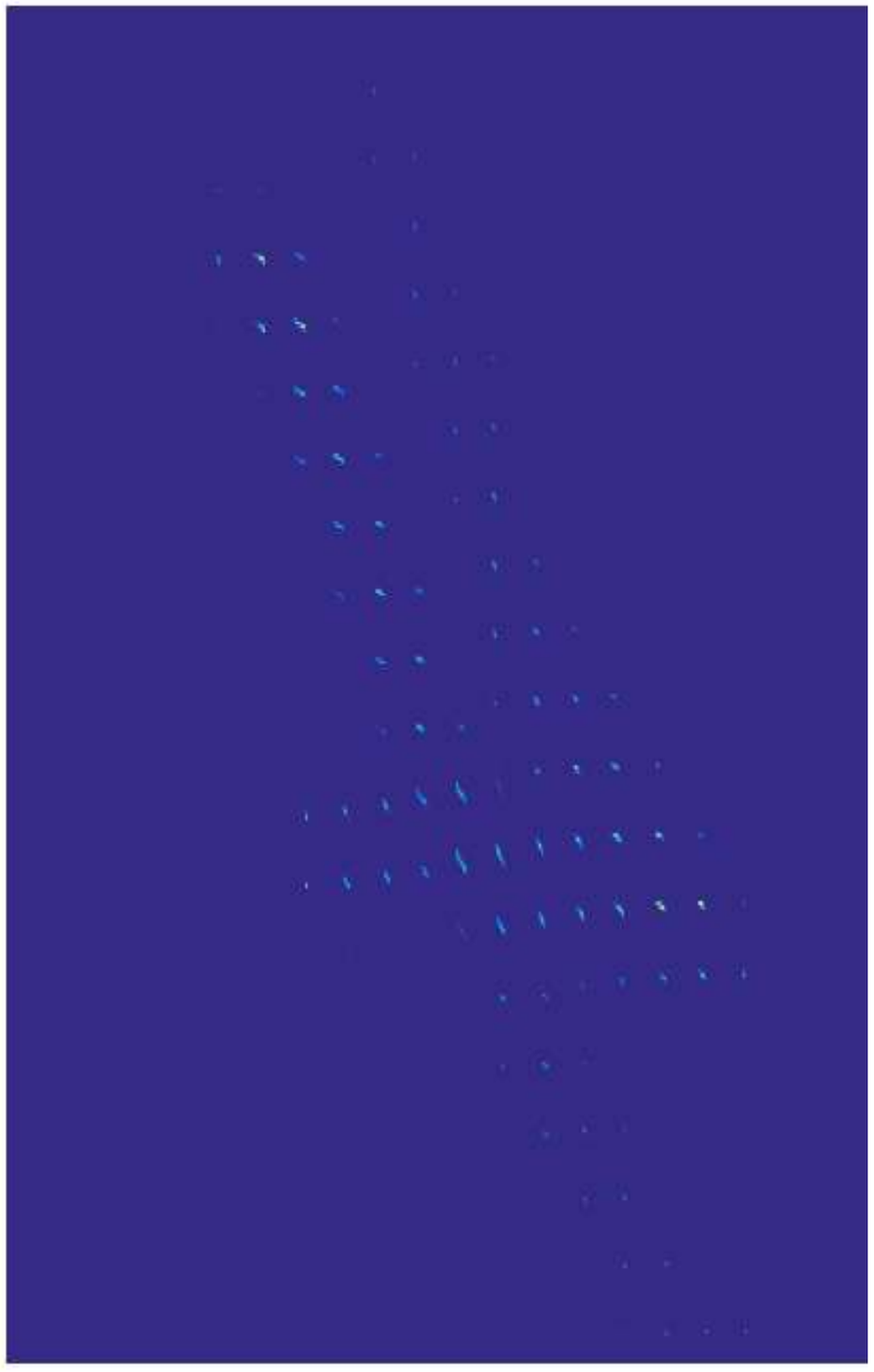


$\log(v_6)$

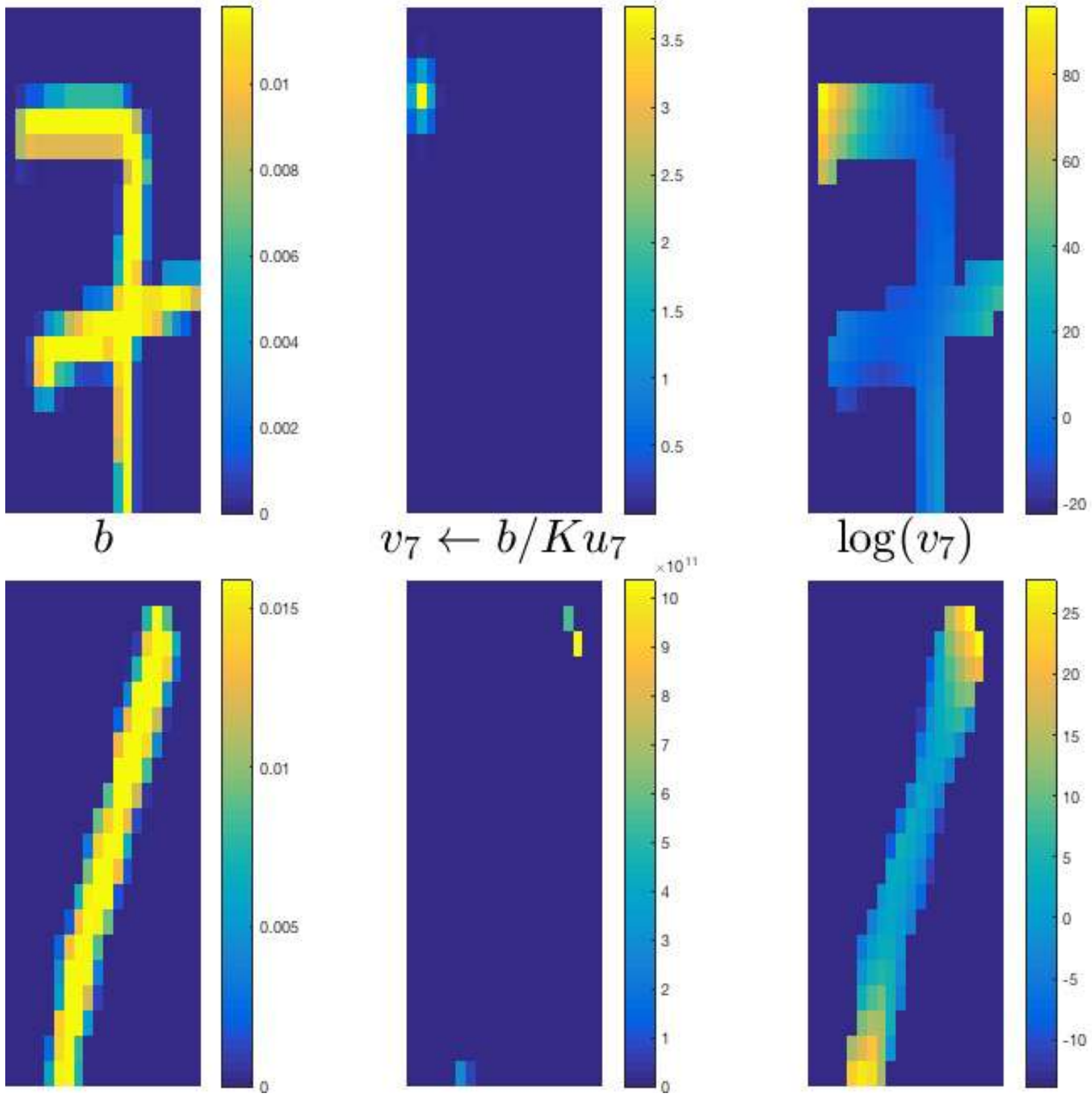


$P_5 = D(u_5)KD(v_5)$

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

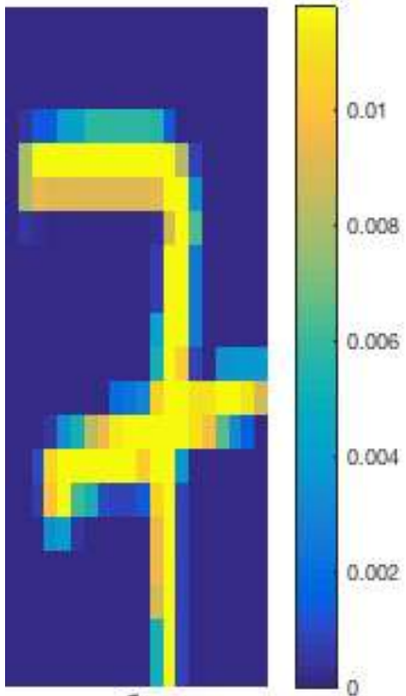


Very Fast EMD Approx. Solver

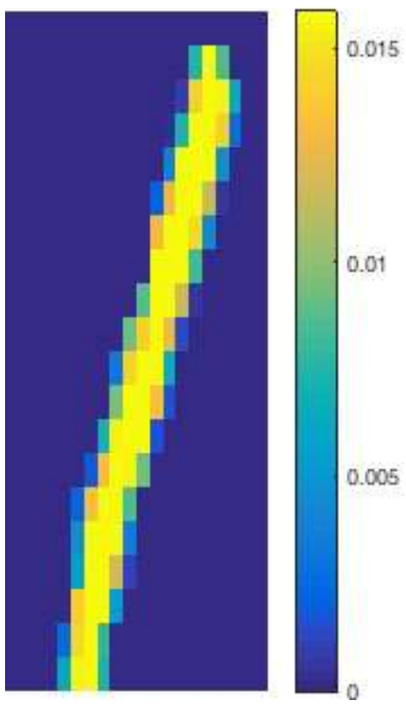


Very Fast EMD Approx. Solver

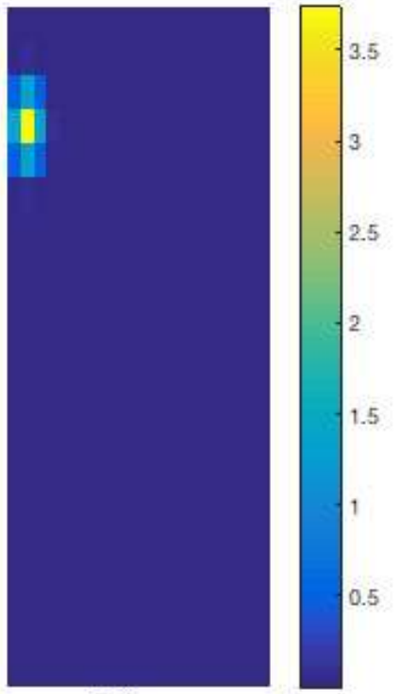
a



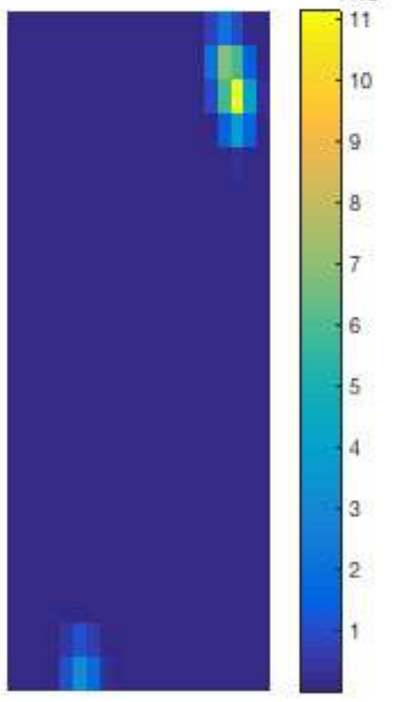
b



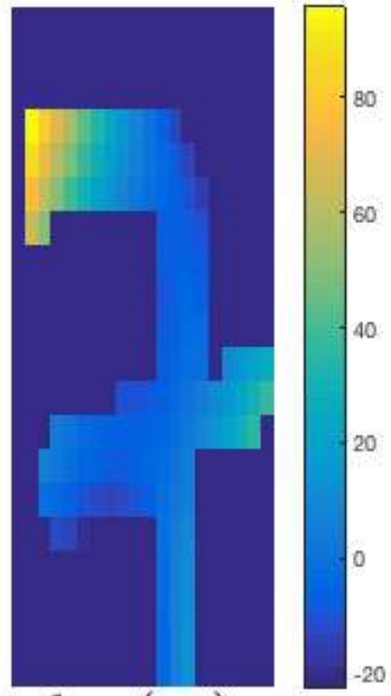
Ku_6



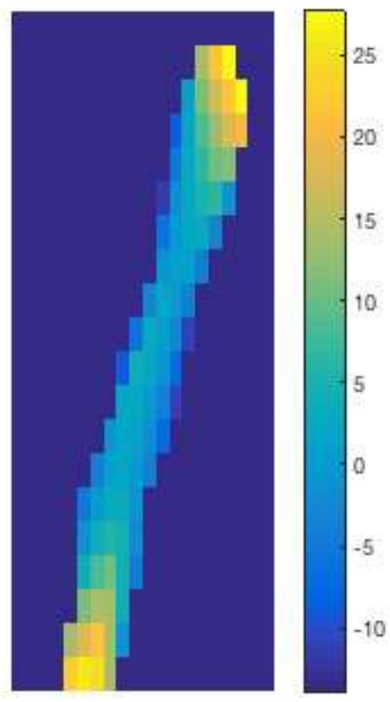
Kv_7



$\log(u_6)$

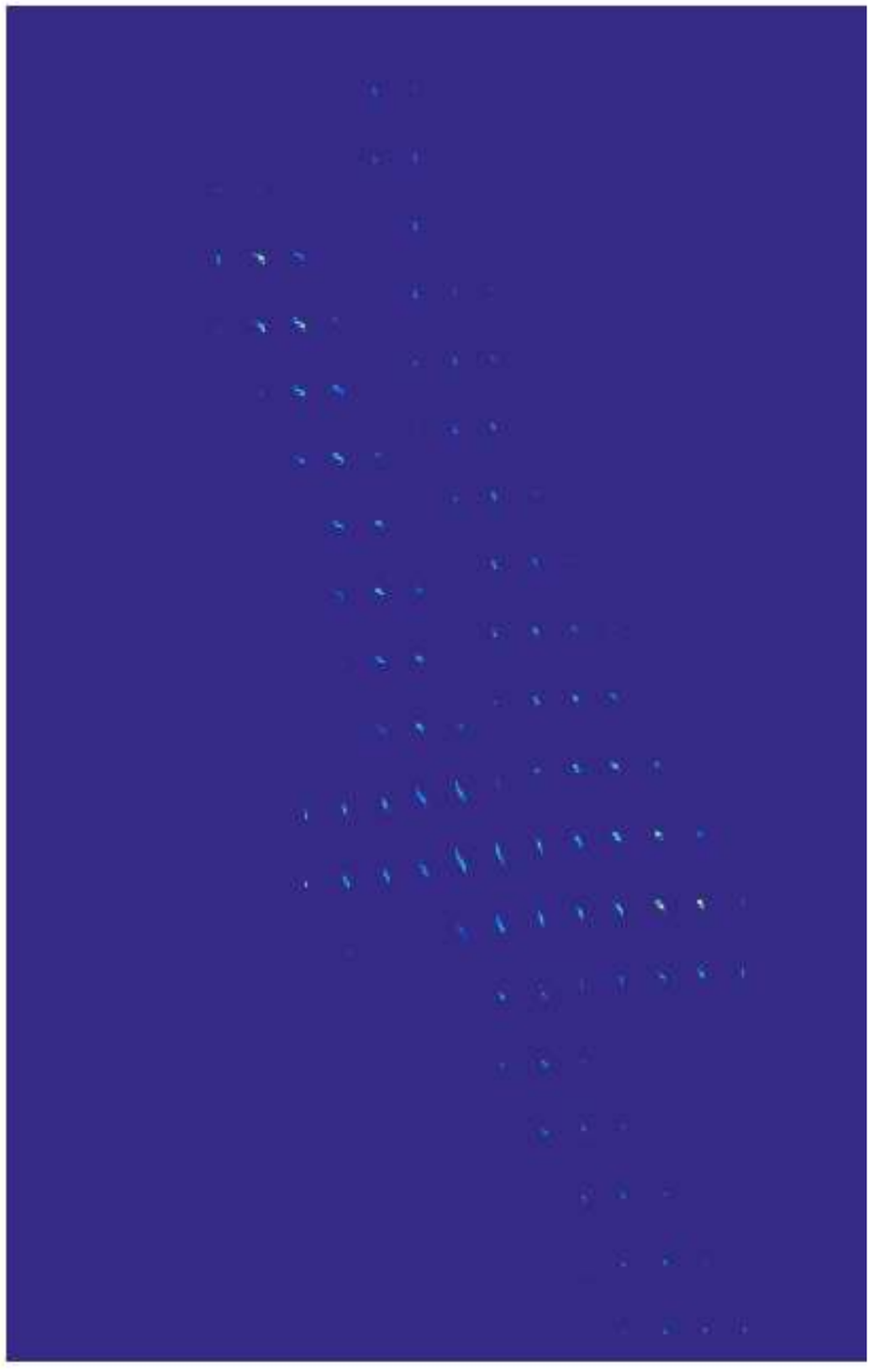


$\log(v_7)$



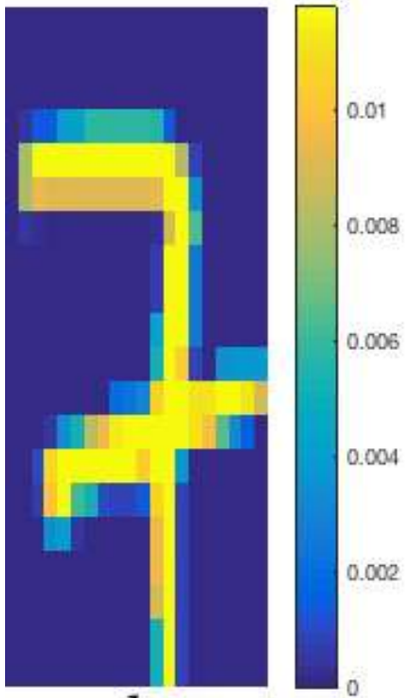
$P_6 = D(u_6)KD(v_6)$

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

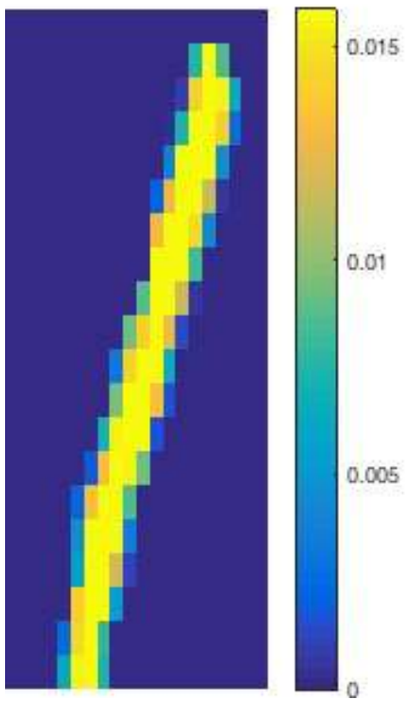


Very Fast EMD Approx. Solver

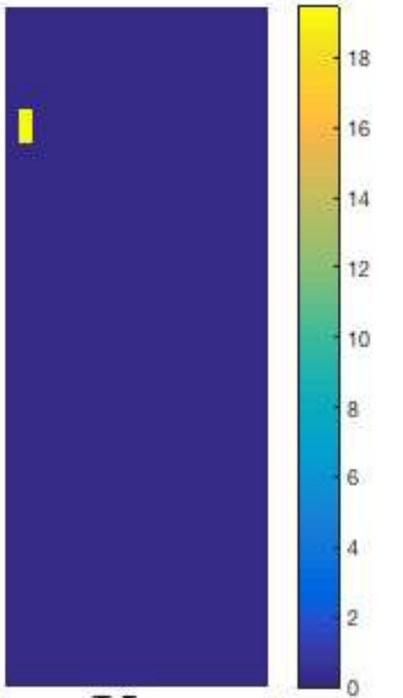
a



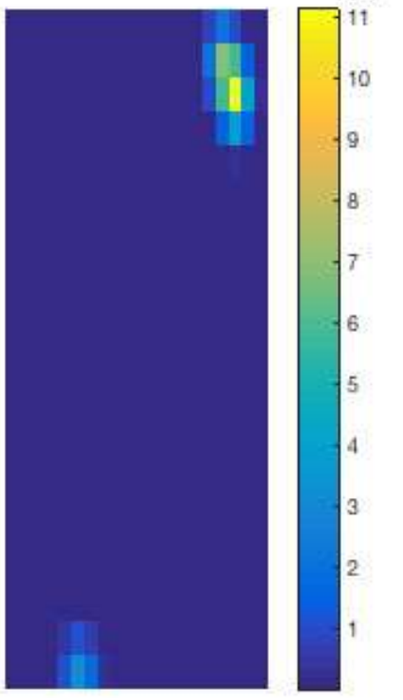
b



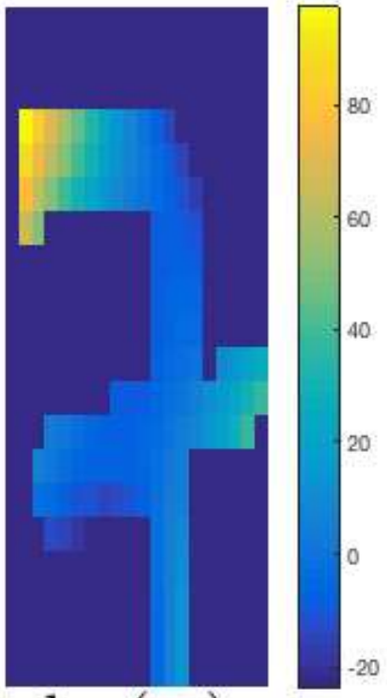
$u_7 \leftarrow a/Kv_7$



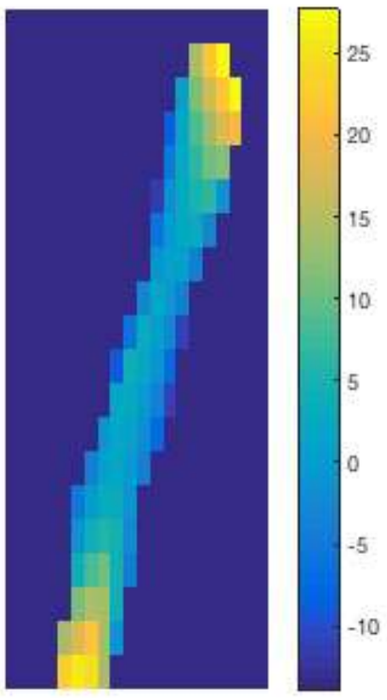
Kv_7



$\log(u_7)$

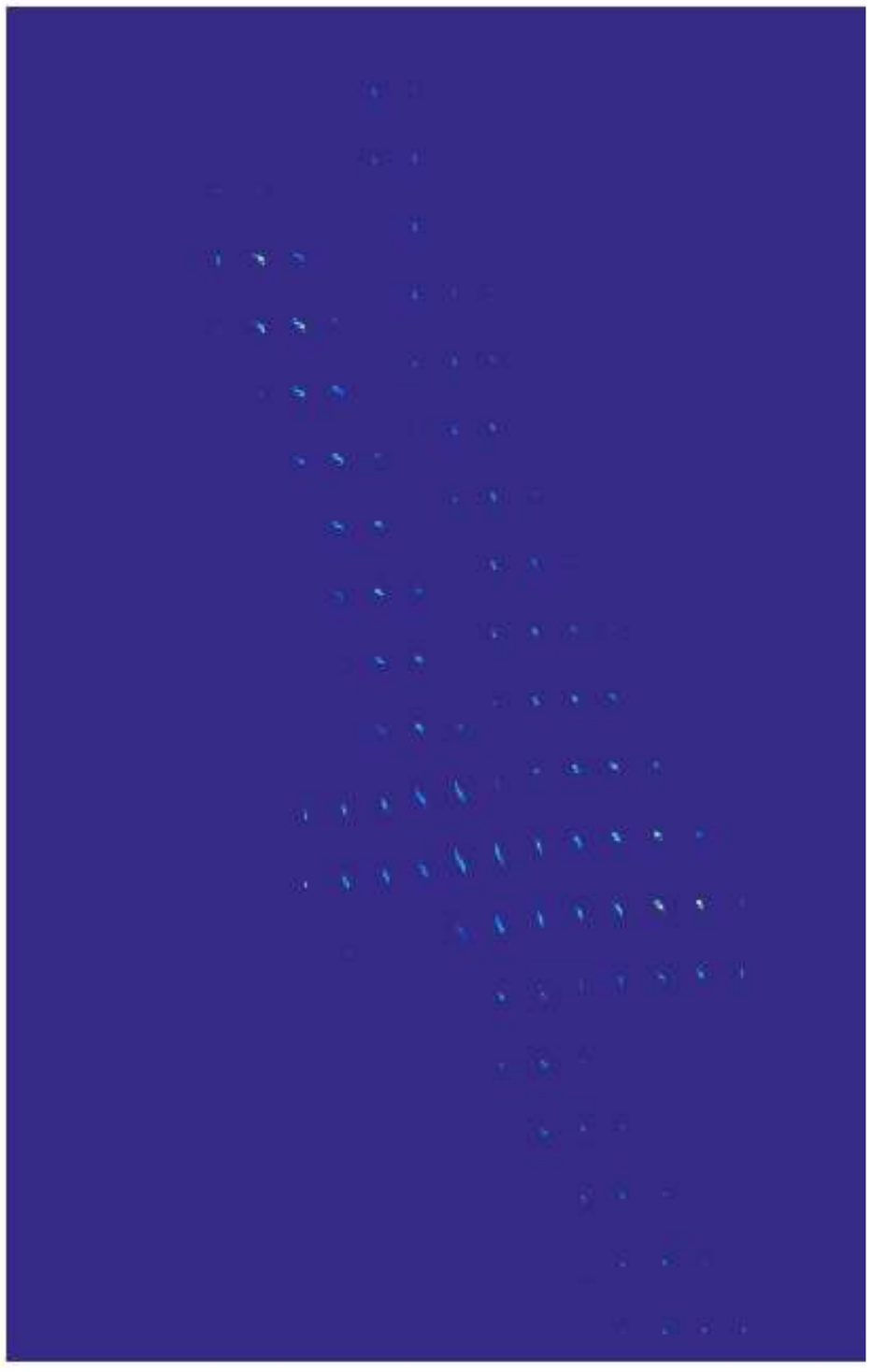


$\log(v_7)$



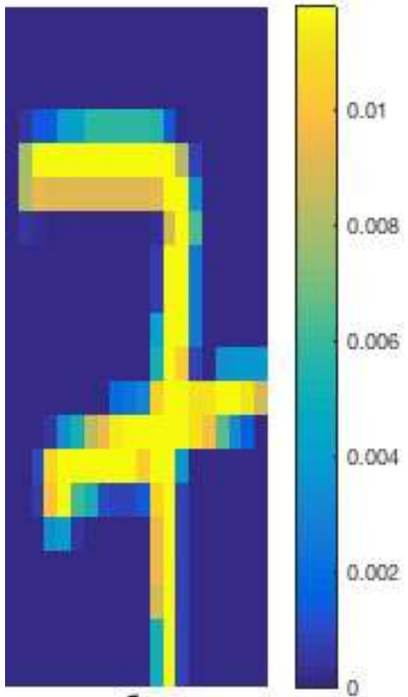
$P_6 = D(u_6)KD(v_6)$

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

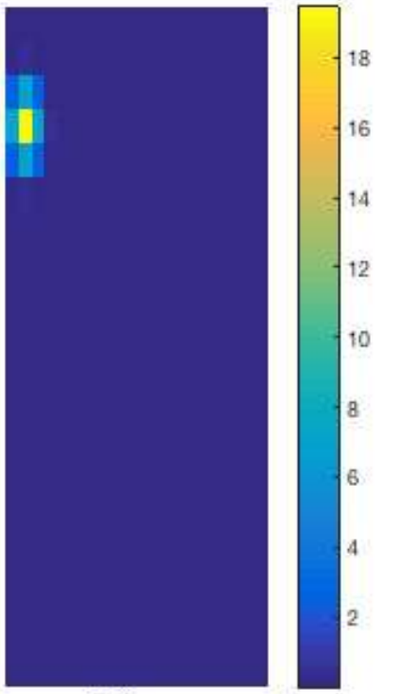


Very Fast EMD Approx. Solver

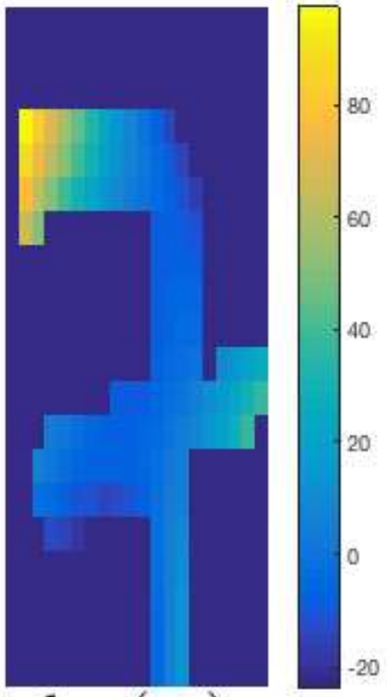
a



Ku_7

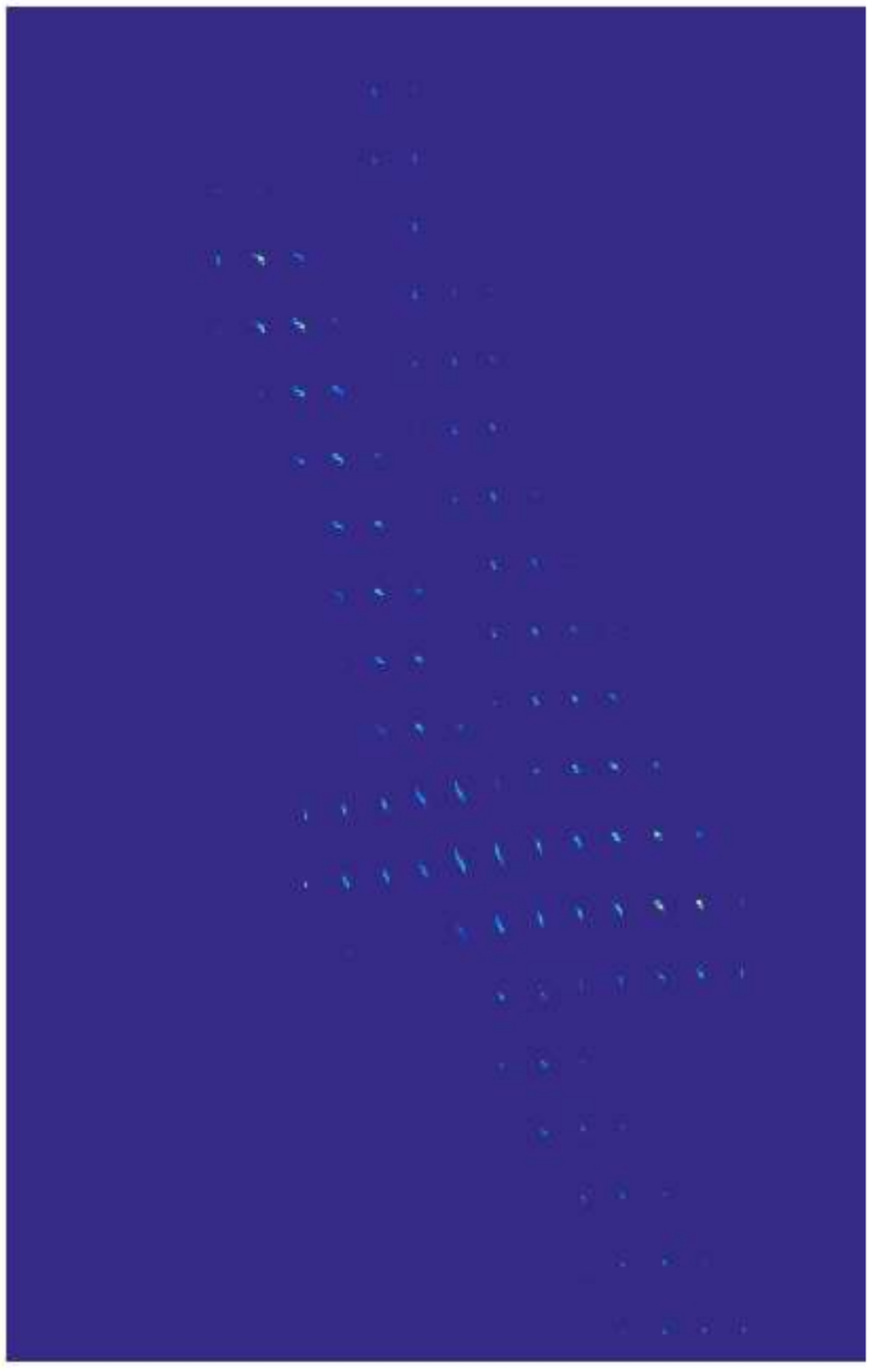


$\log(u_7)$

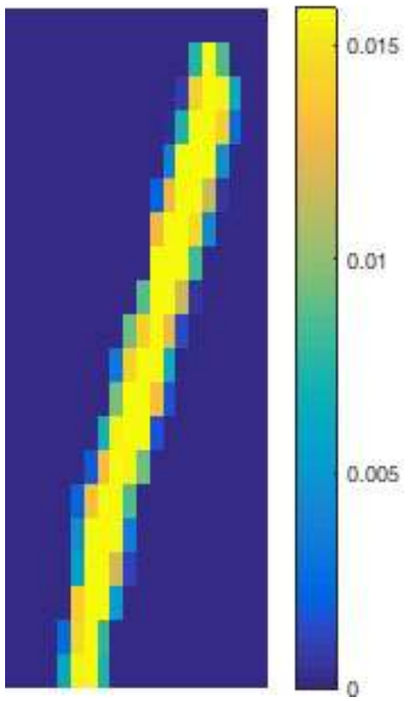


$P_6 = D(u_6)KD(v_6)$

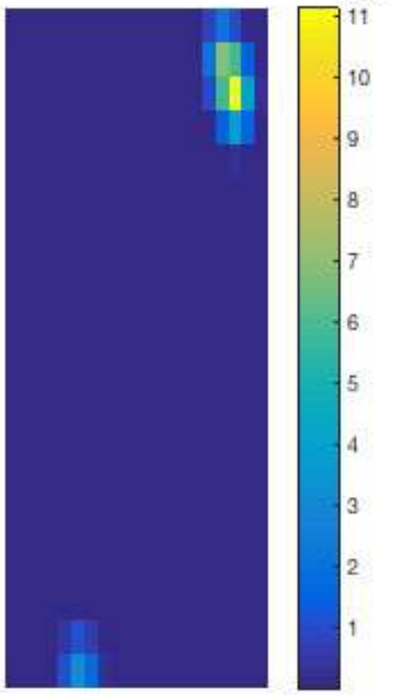
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$



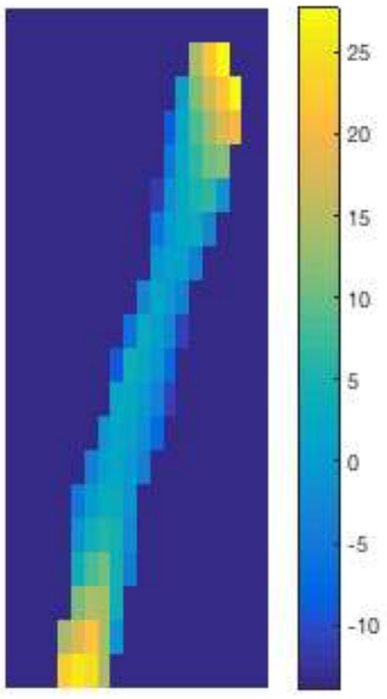
b



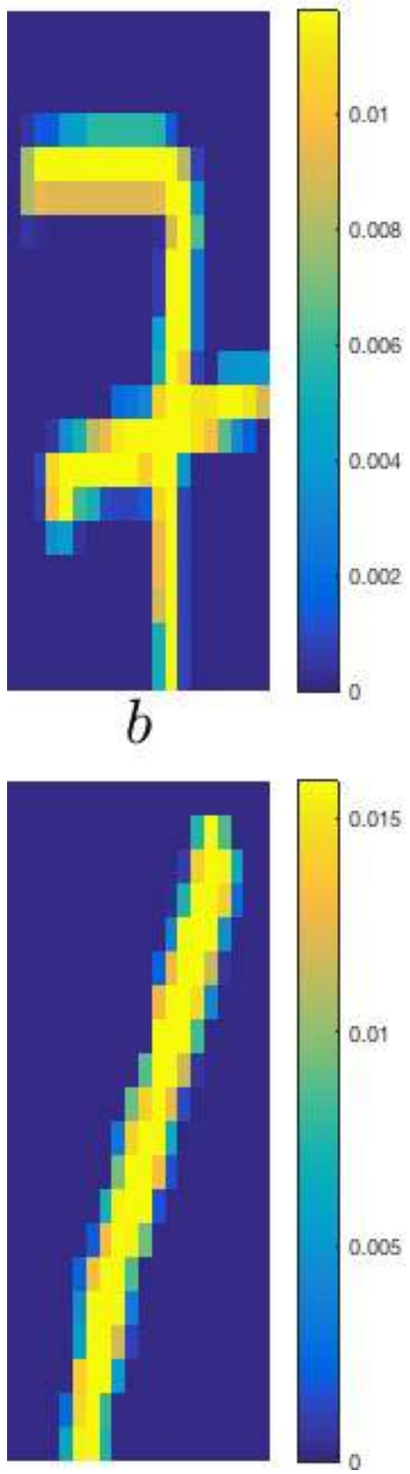
Kv_7

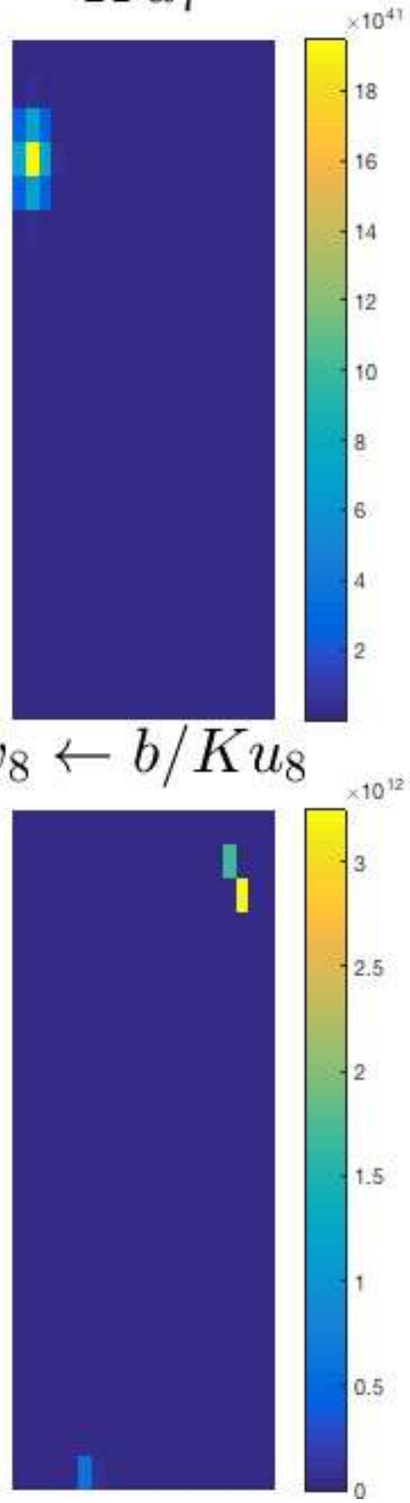


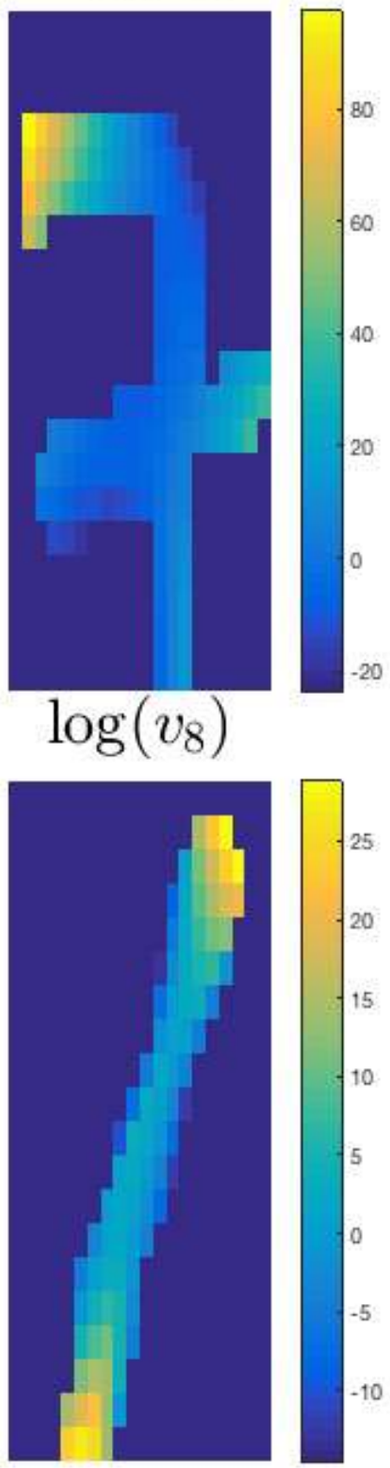
$\log(v_7)$

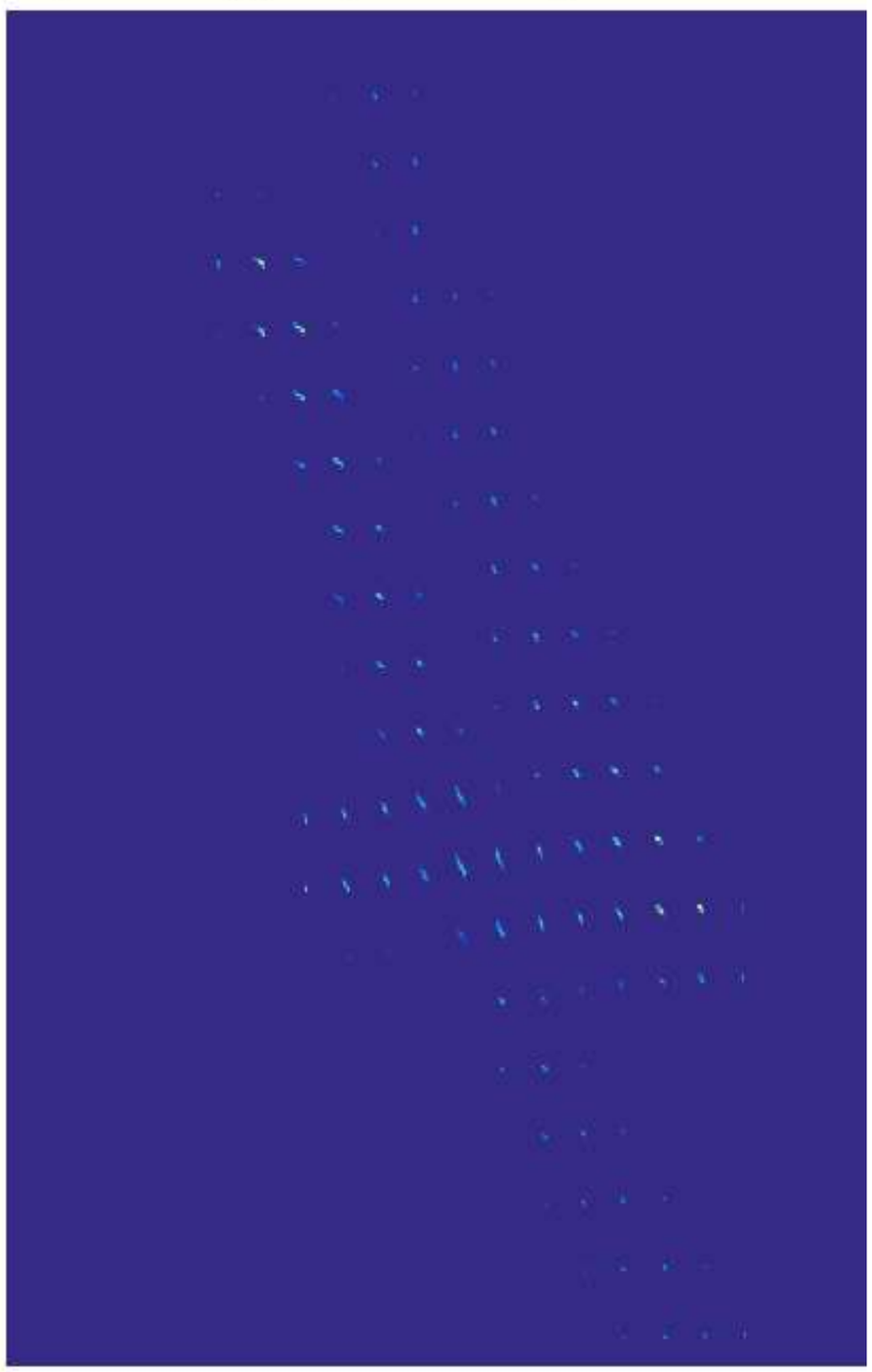


Very Fast EMD Approx. Solver



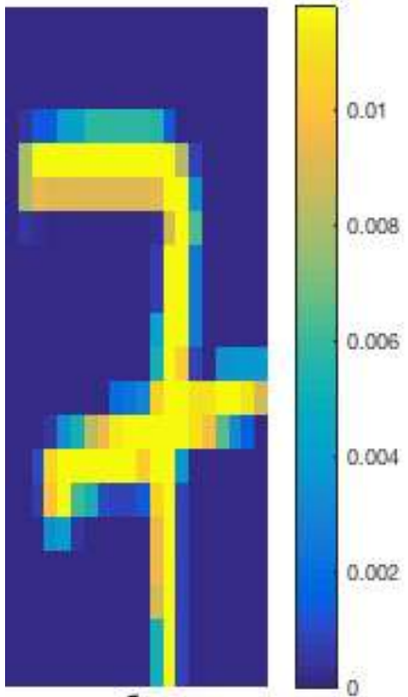




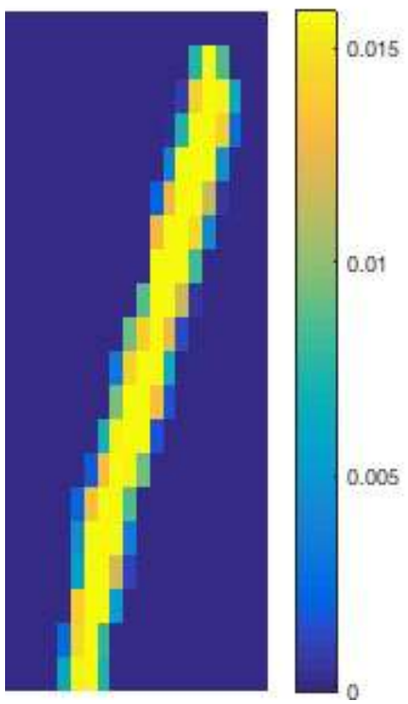


Very Fast EMD Approx. Solver

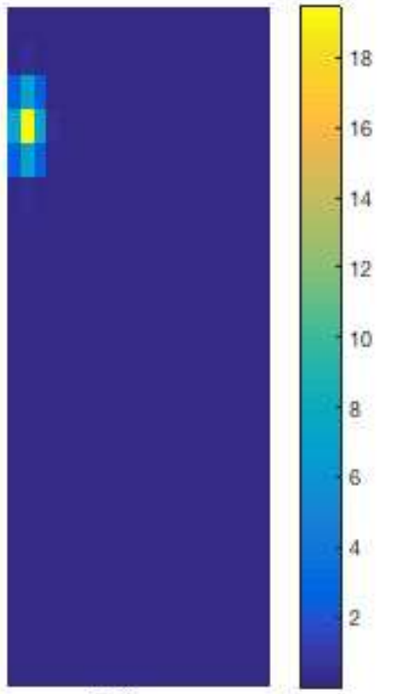
a



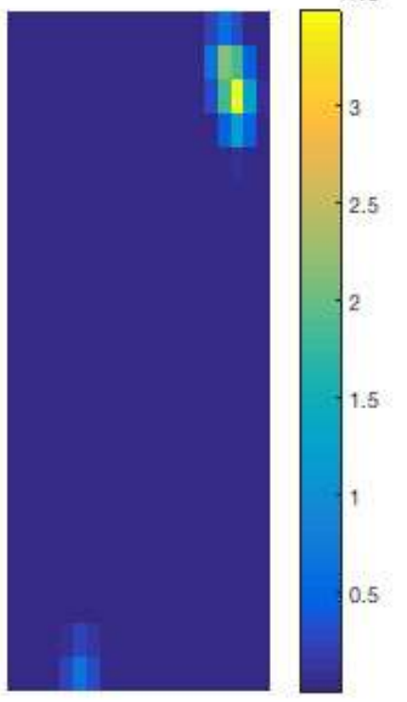
b



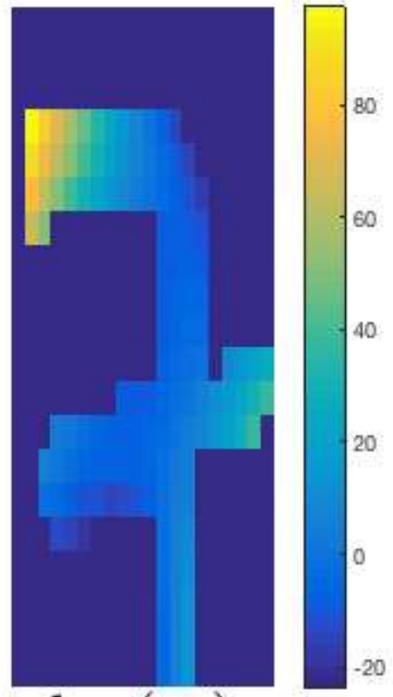
Ku_7



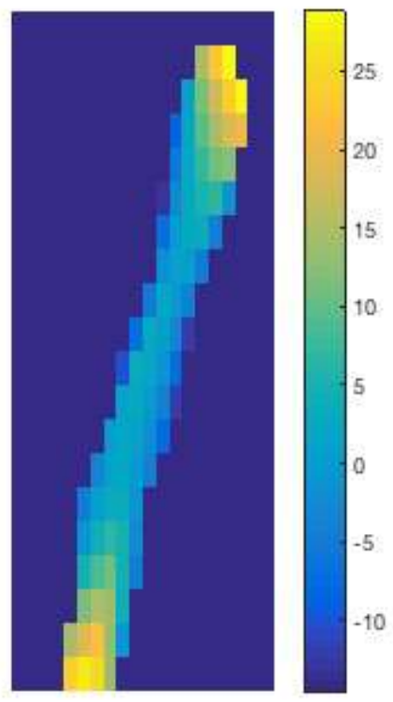
Kv_8



$\log(u_7)$

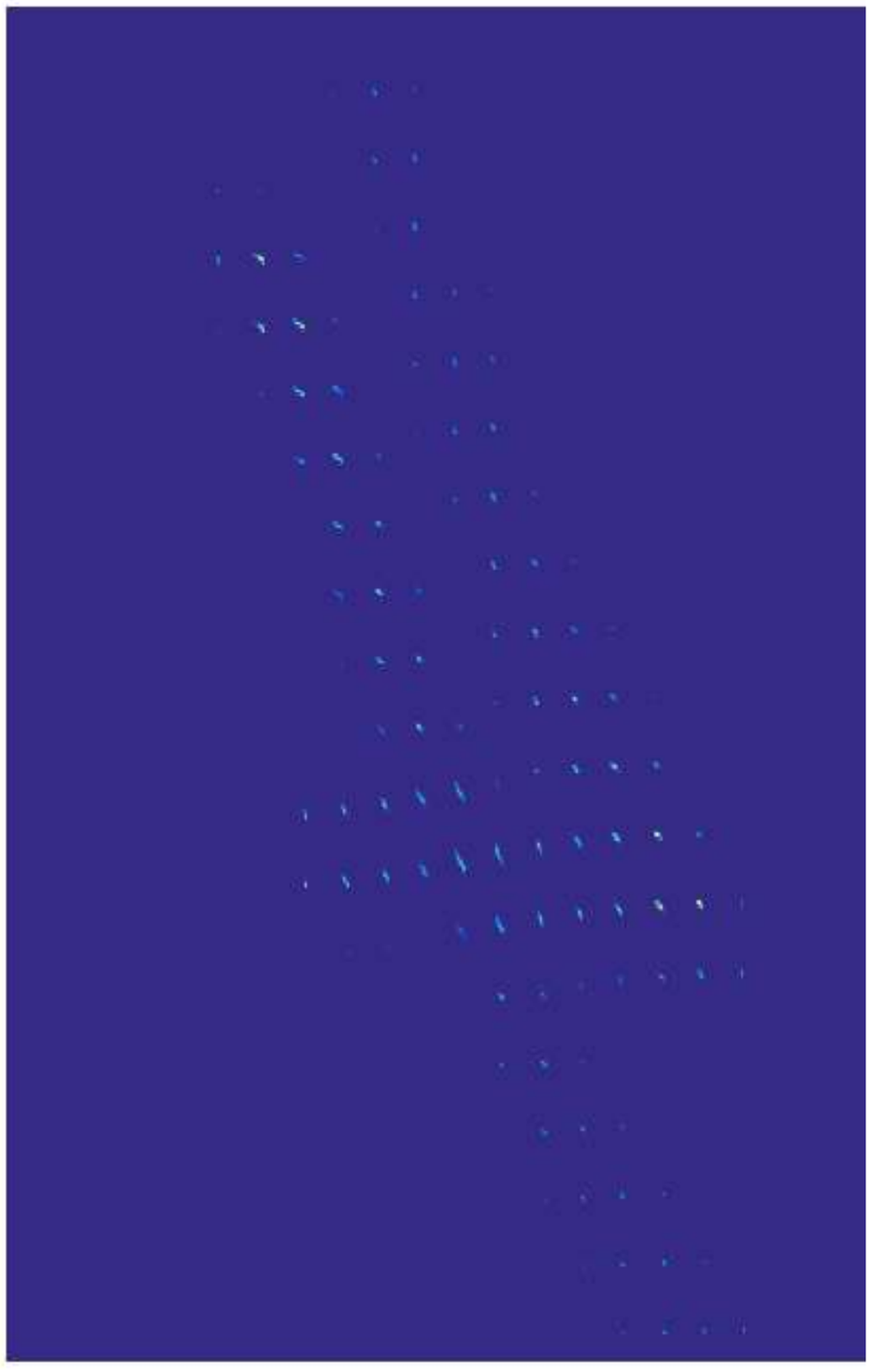


$\log(v_8)$



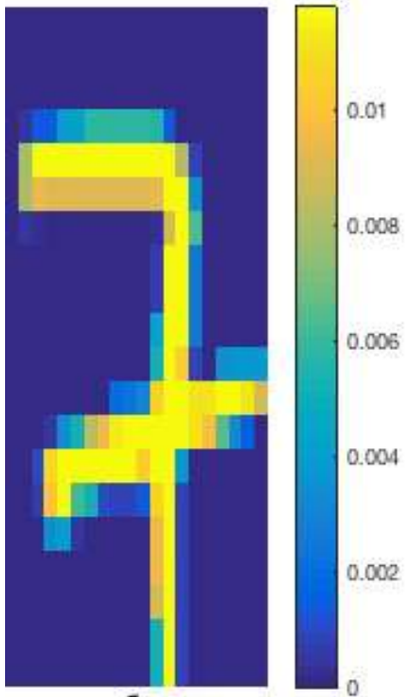
$P_7 = D(u_7)KD(v_7)$

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

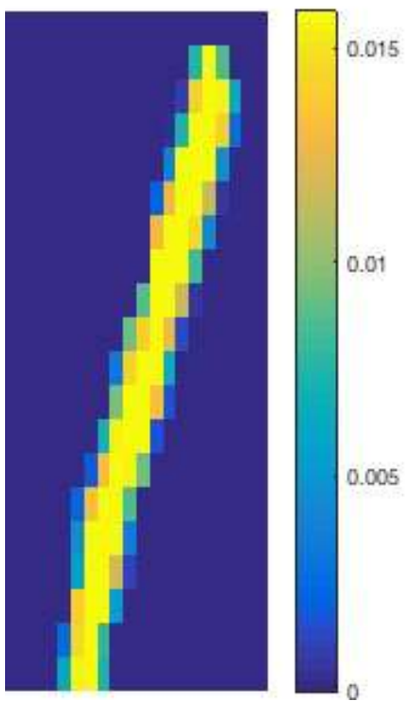


Very Fast EMD Approx. Solver

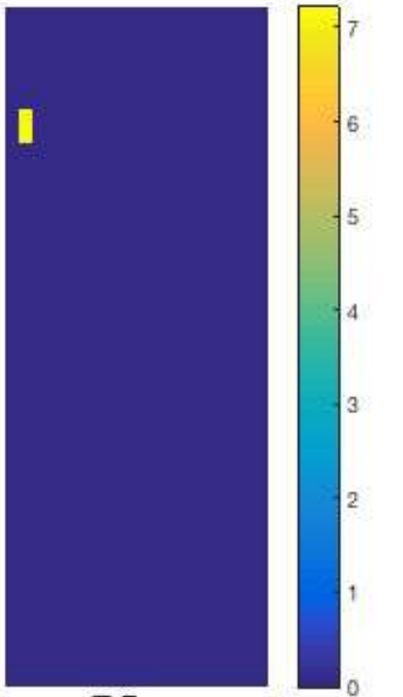
a



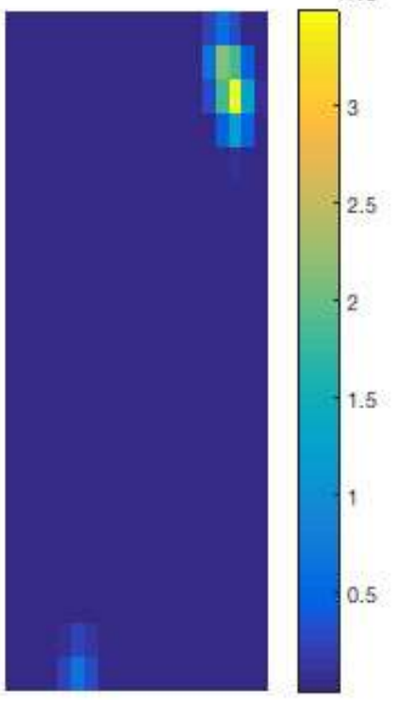
b



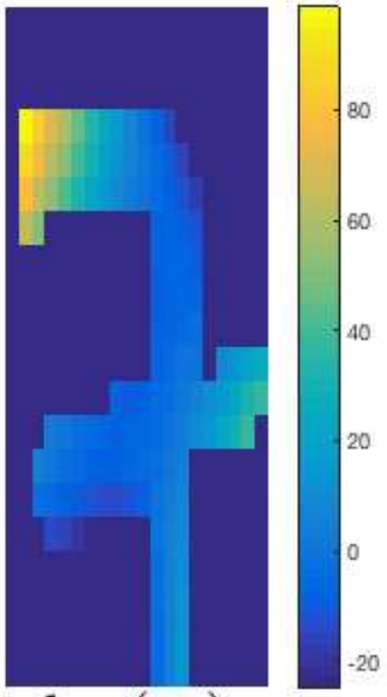
$u_8 \leftarrow a/Kv_8$



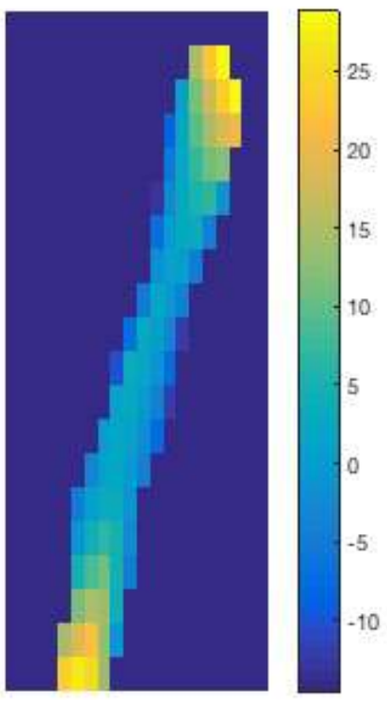
Kv_8



$\log(u_8)$

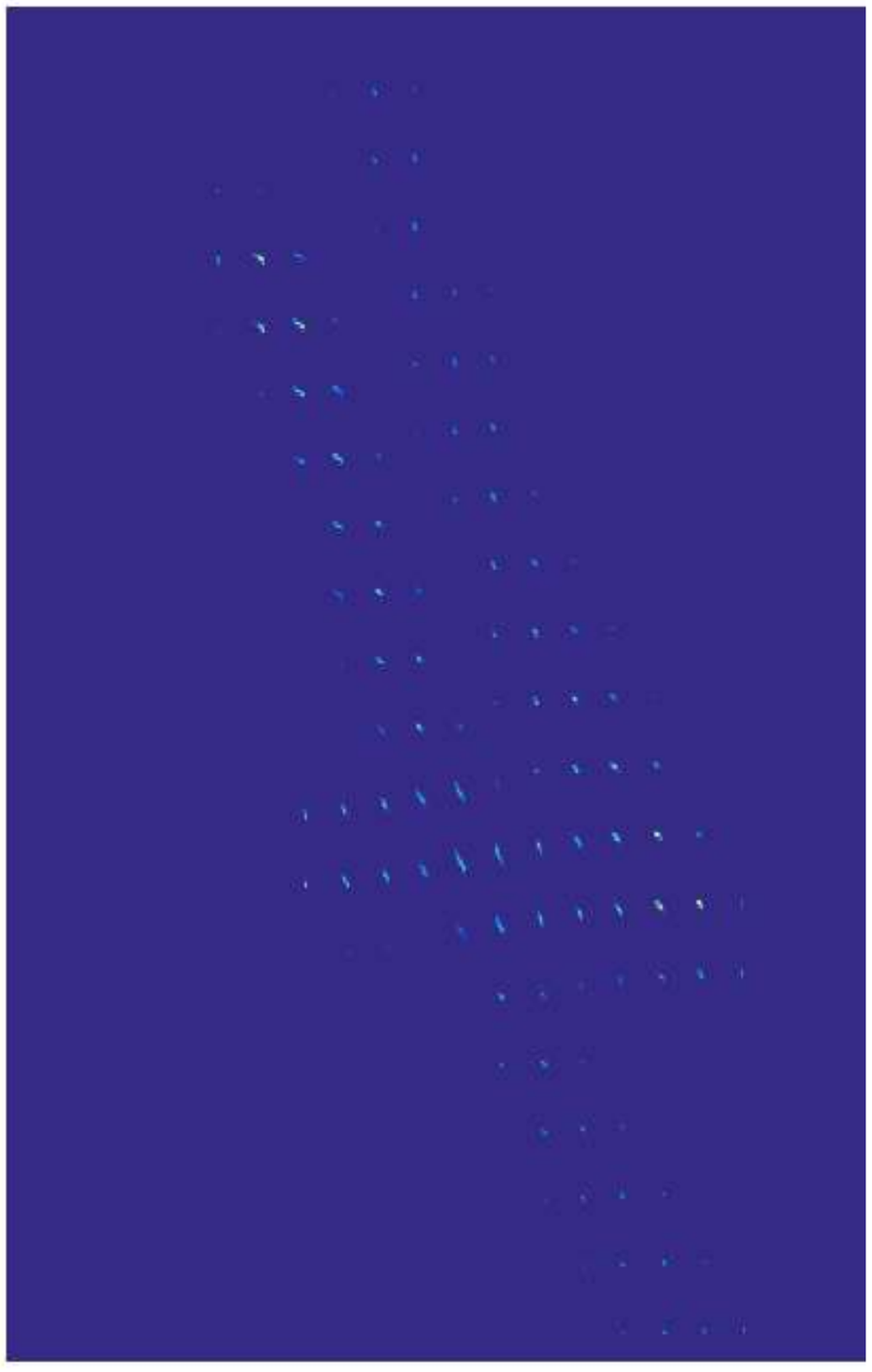


$\log(v_8)$



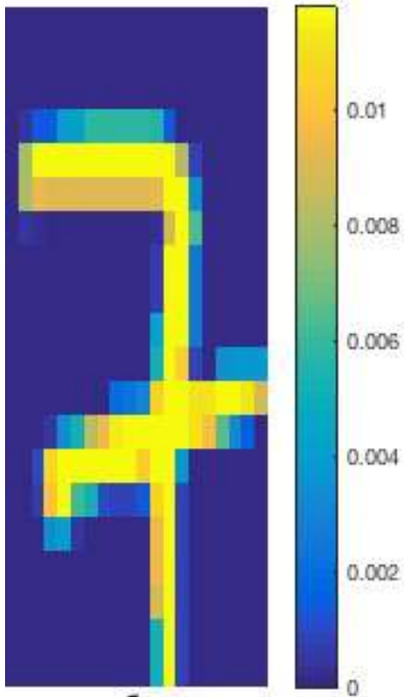
$P_7 = D(u_7)KD(v_7)$

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

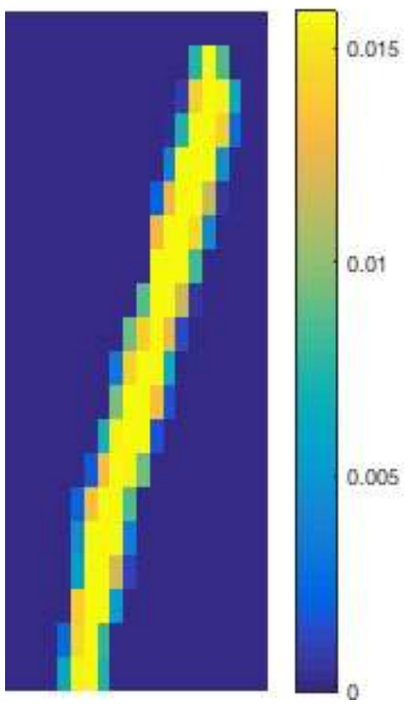


Very Fast EMD Approx. Solver

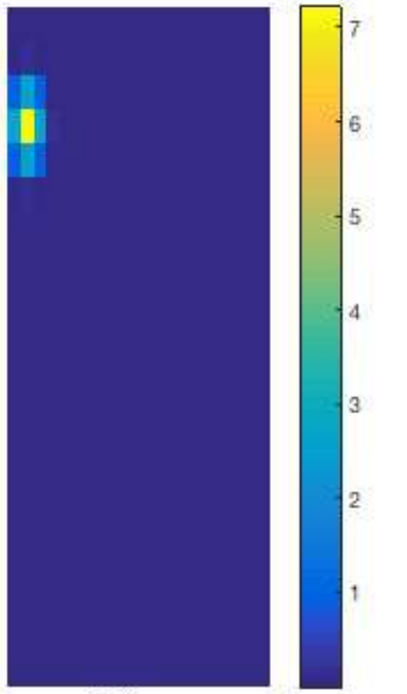
a



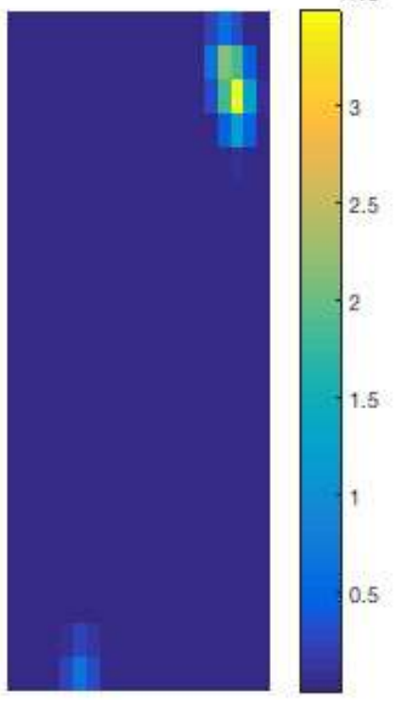
b



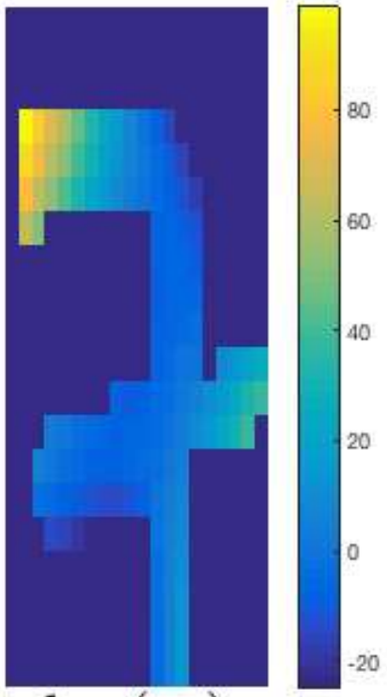
Ku_8



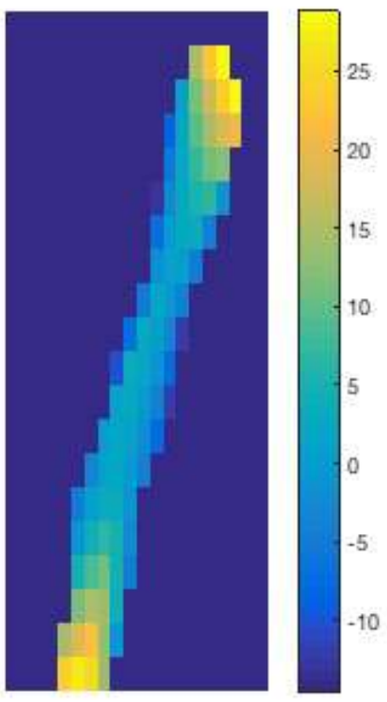
Kv_8



$\log(u_8)$

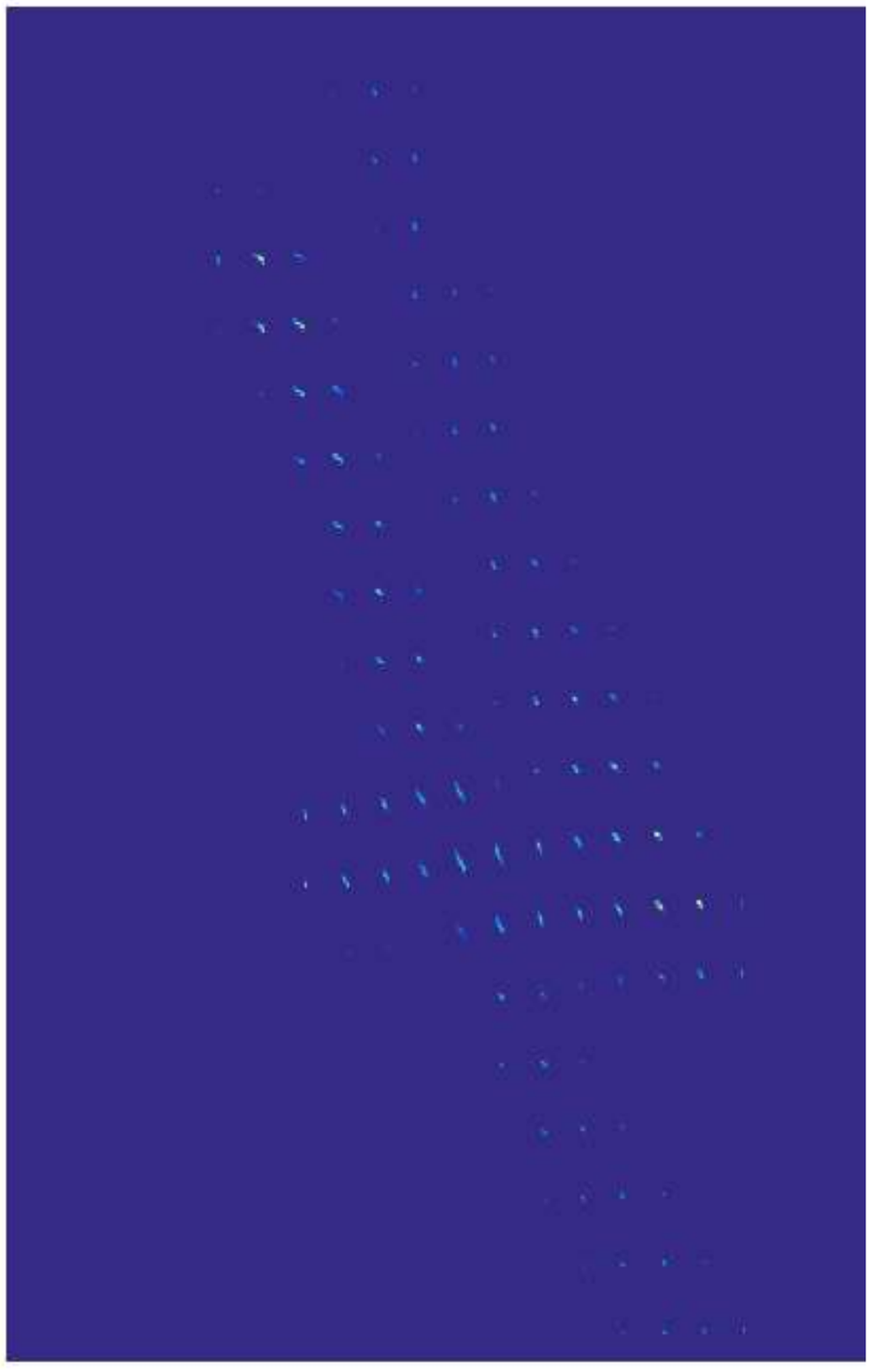


$\log(v_8)$

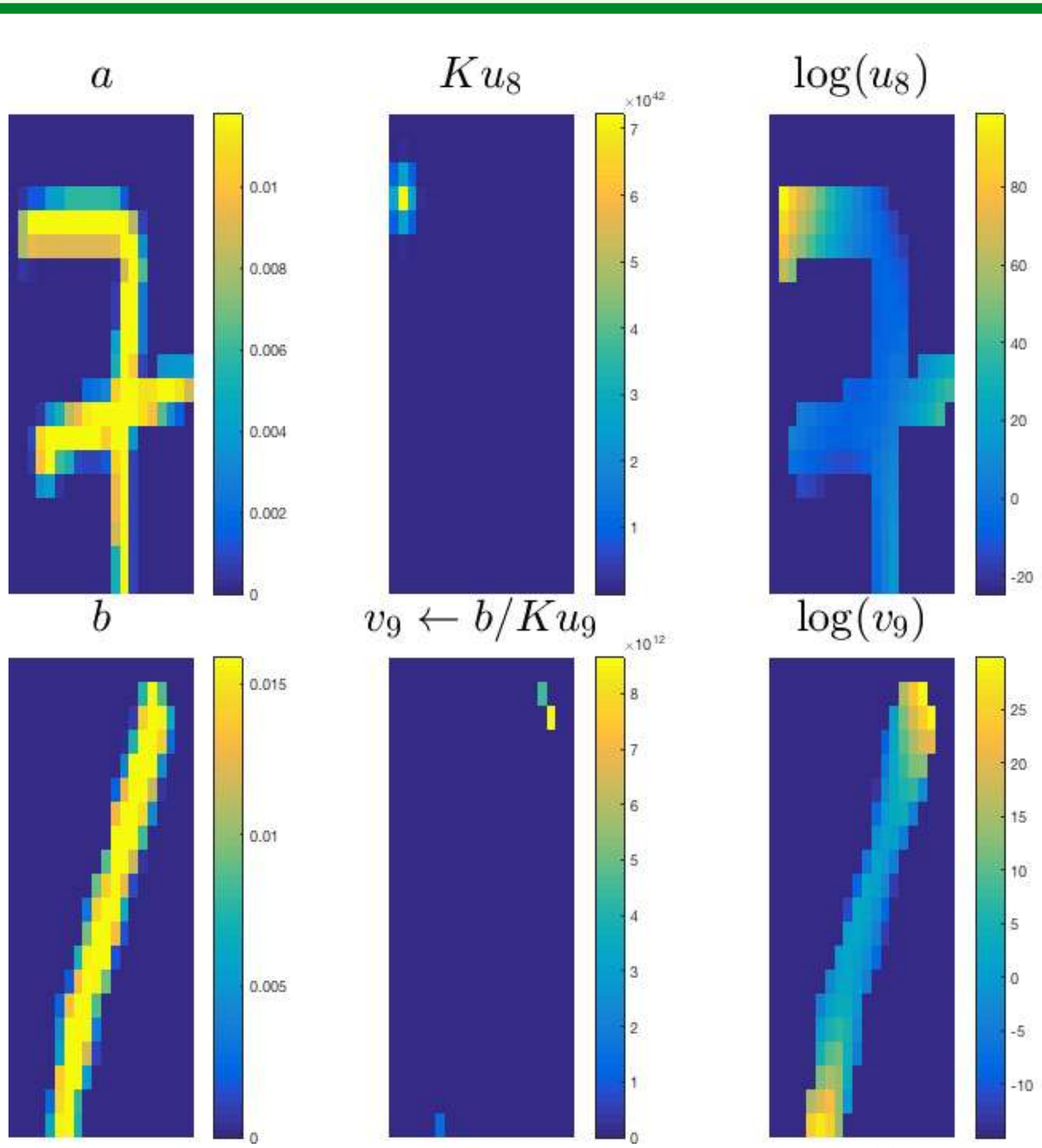


$P_7 = D(u_7)KD(v_7)$

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

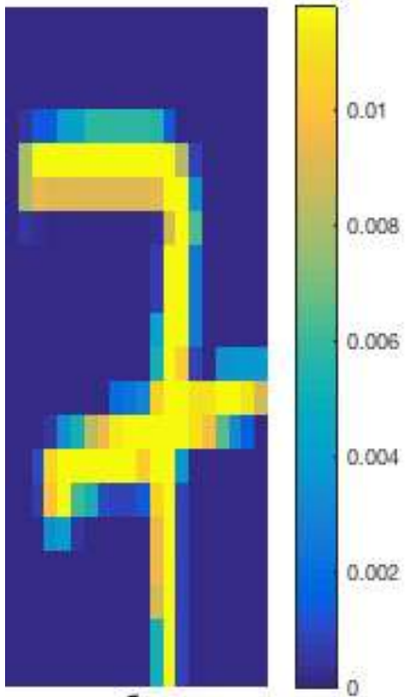


Very Fast EMD Approx. Solver

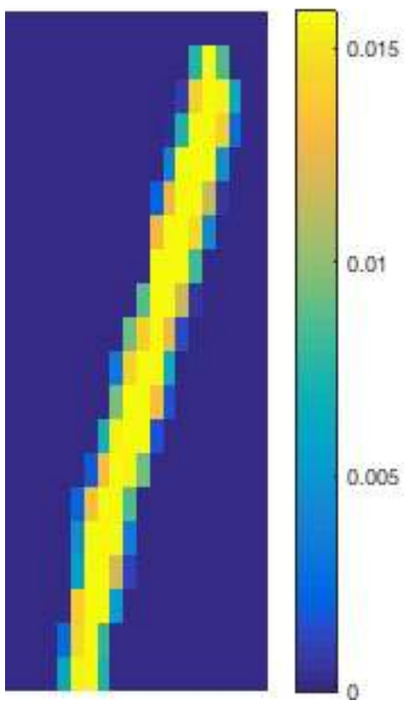


Very Fast EMD Approx. Solver

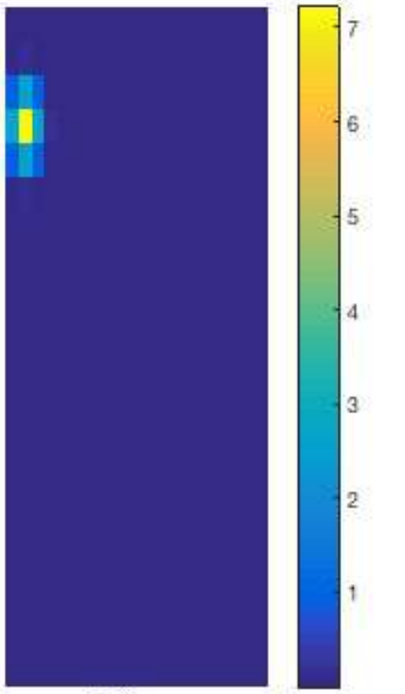
a



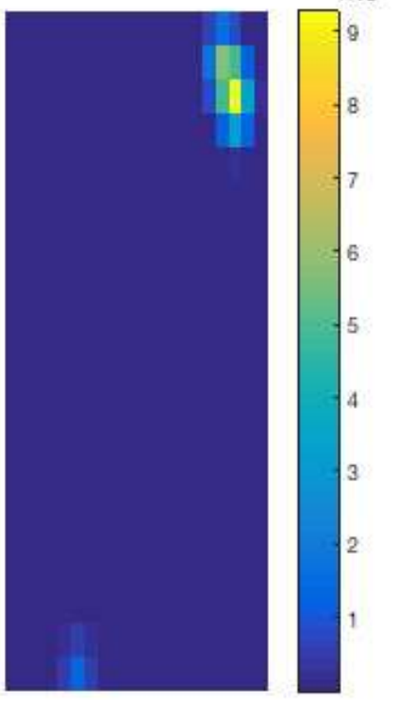
b



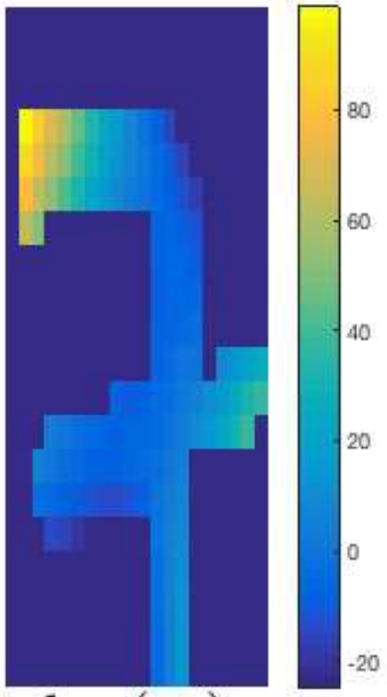
Ku_8



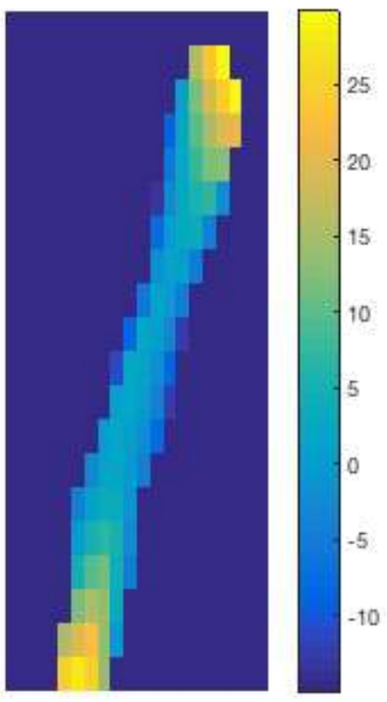
Kv_9



$\log(u_8)$

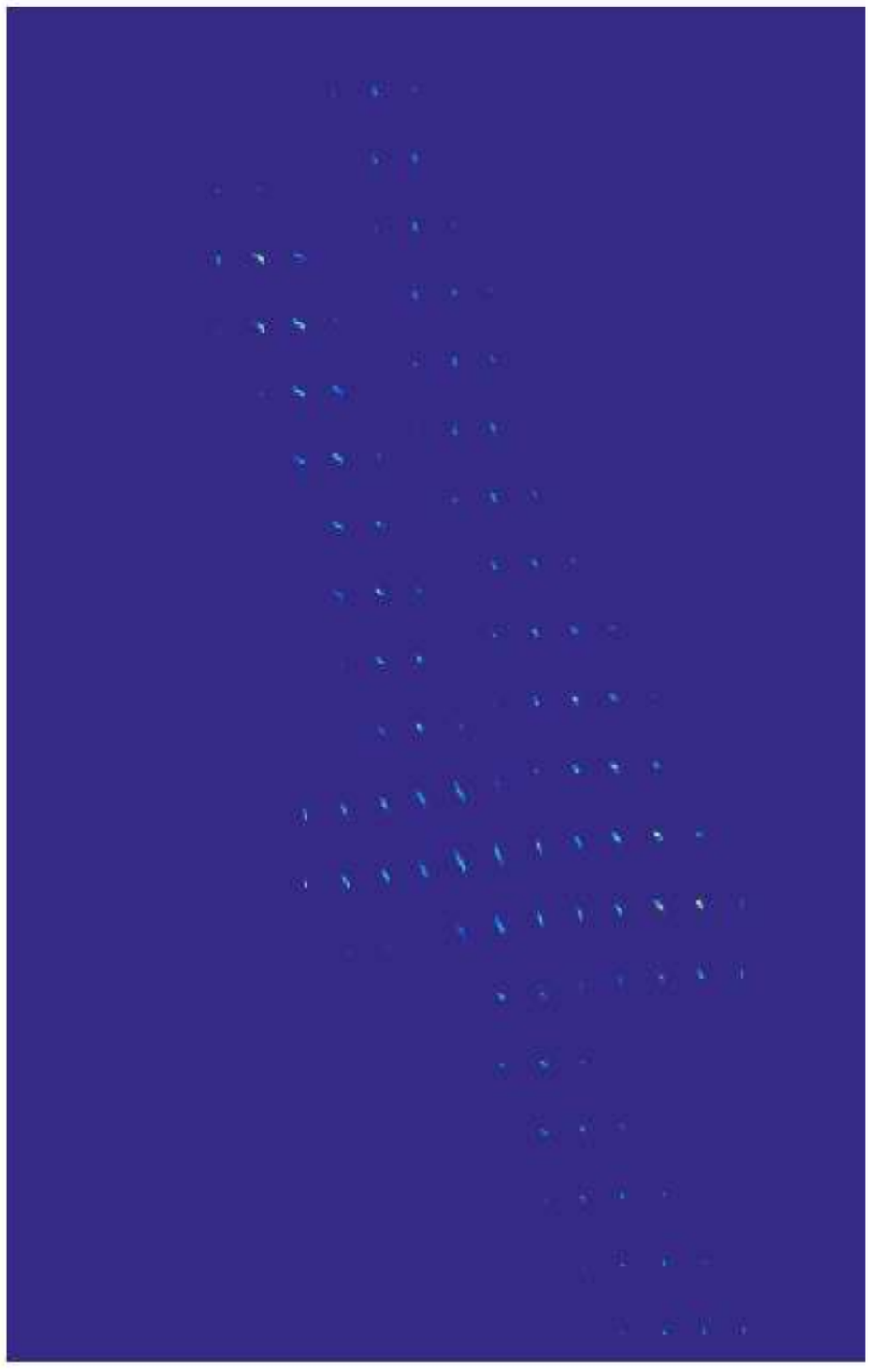


$\log(v_9)$



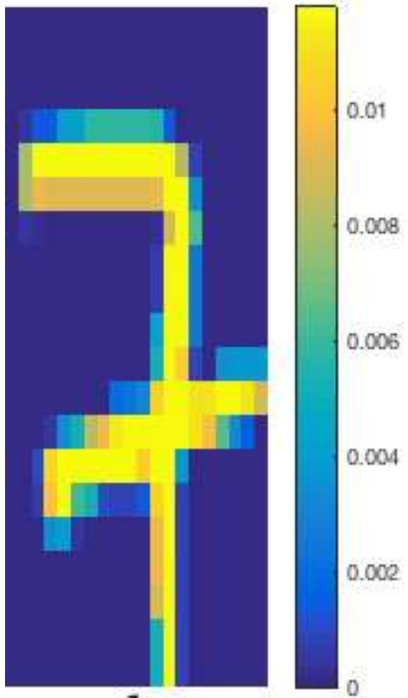
$P_8 = D(u_8)KD(v_8)$

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

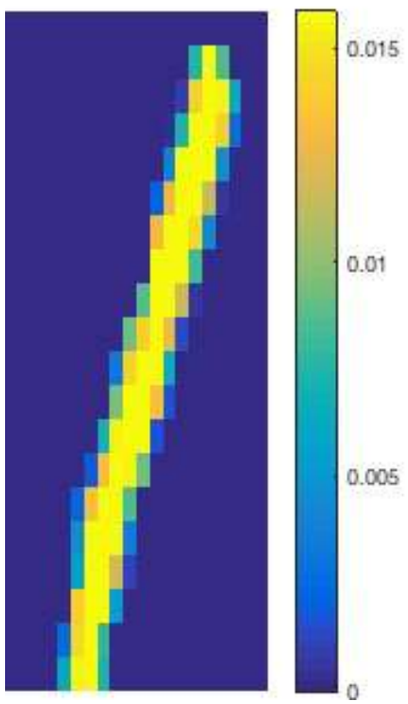


Very Fast EMD Approx. Solver

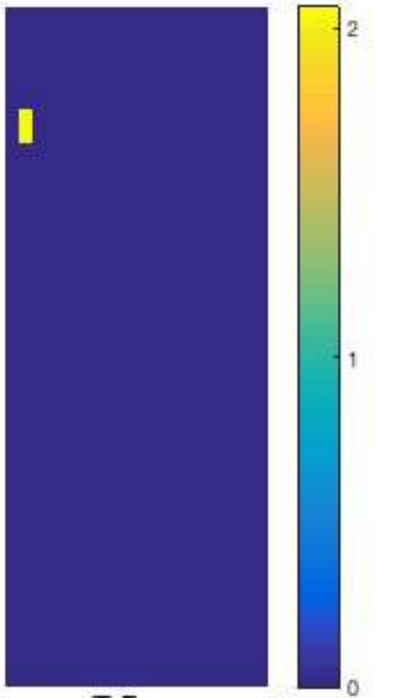
a



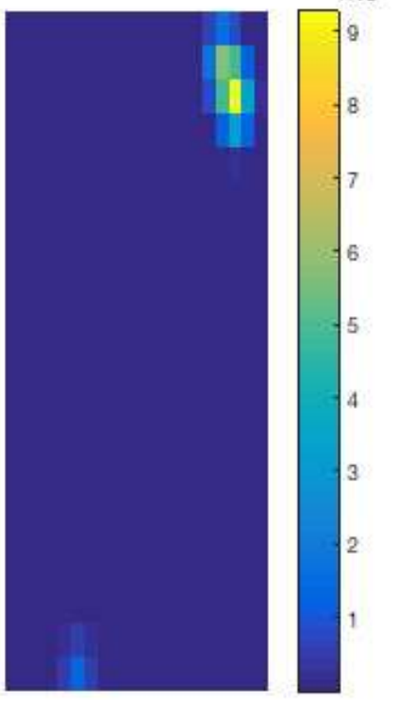
b



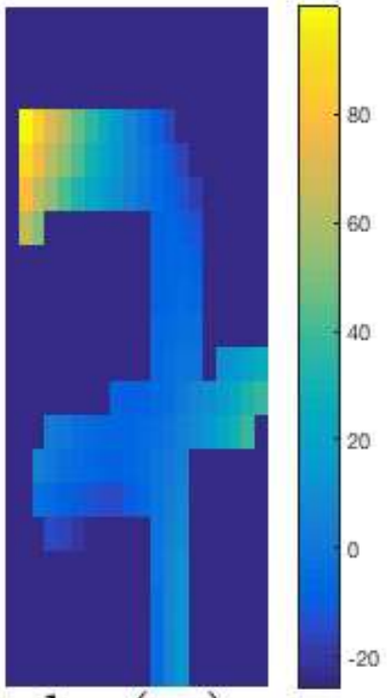
$u_9 \leftarrow a/Kv_9$



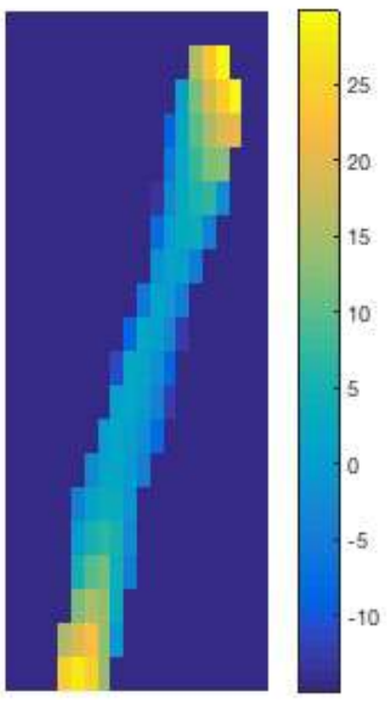
Kv_9



$\log(u_9)$

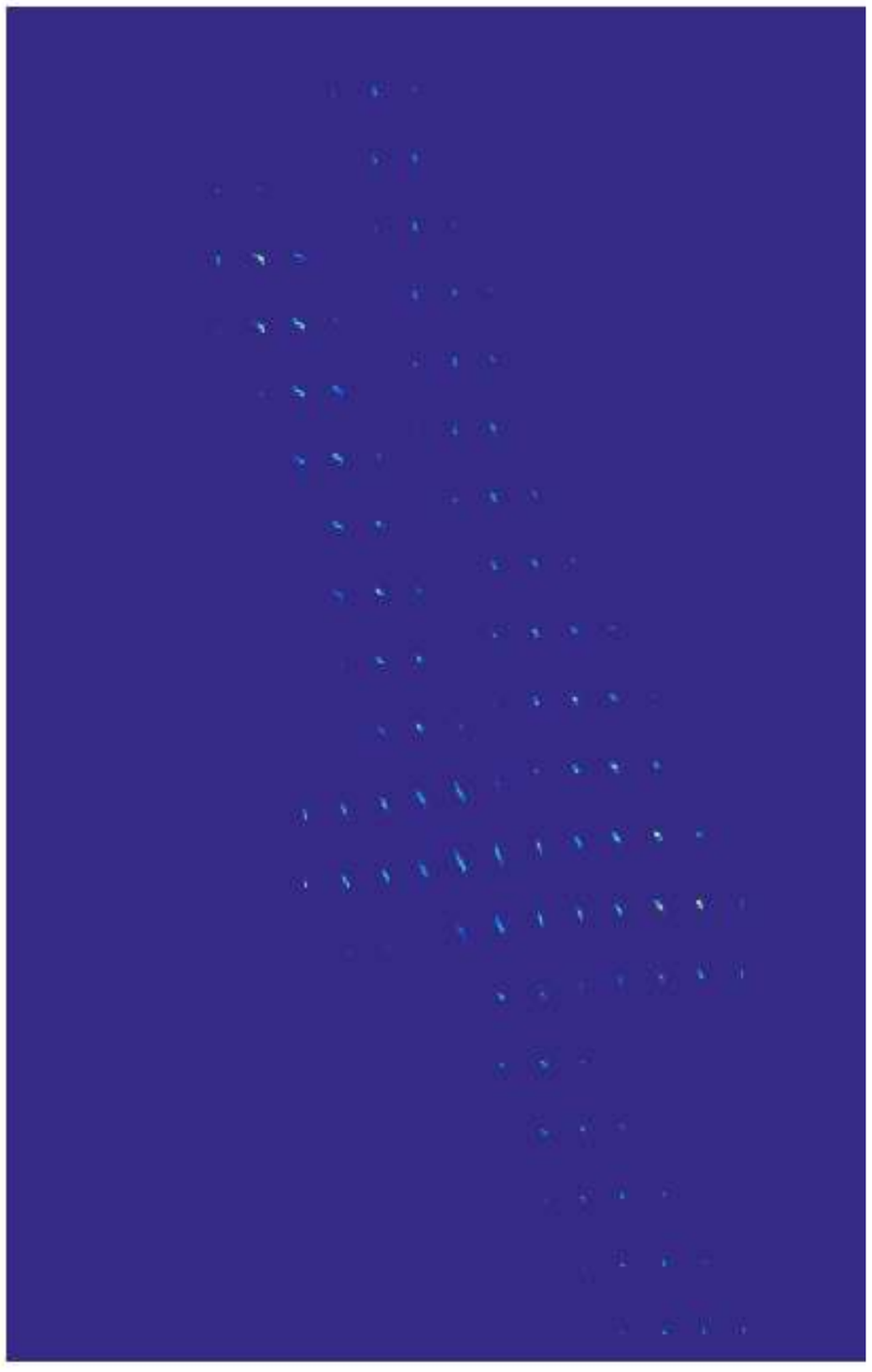


$\log(v_9)$



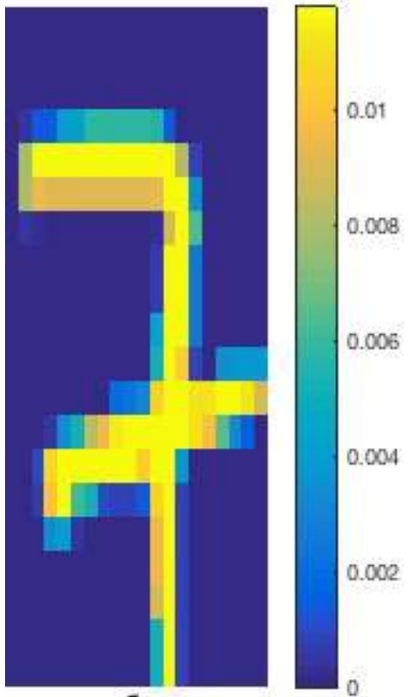
$P_8 = D(u_8)KD(v_8)$

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

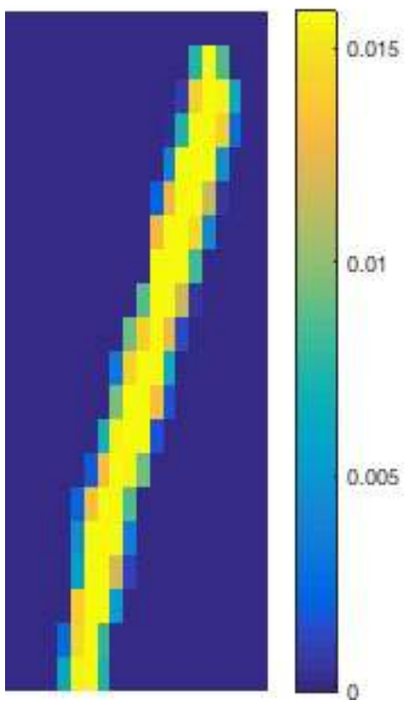


Very Fast EMD Approx. Solver

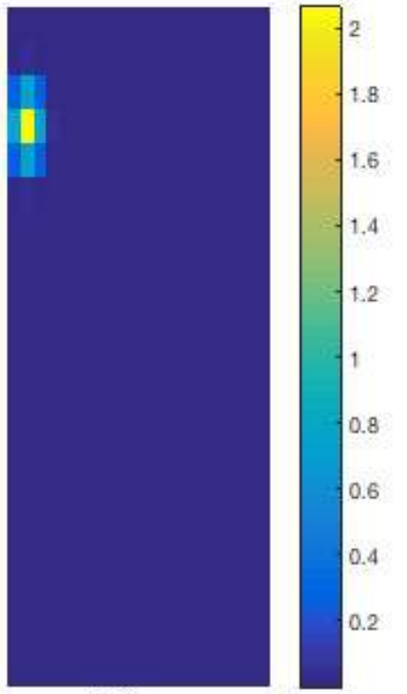
a



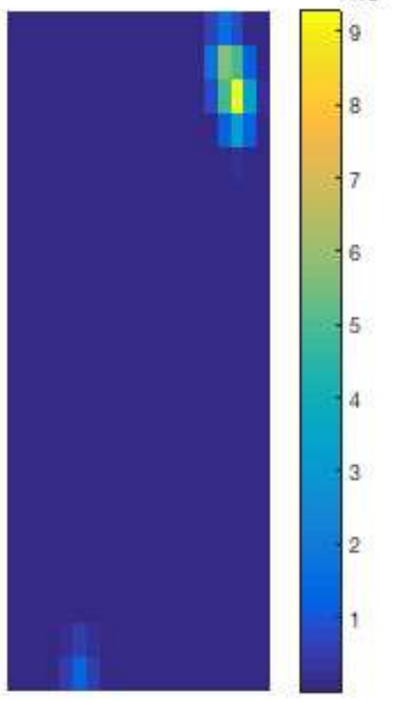
b



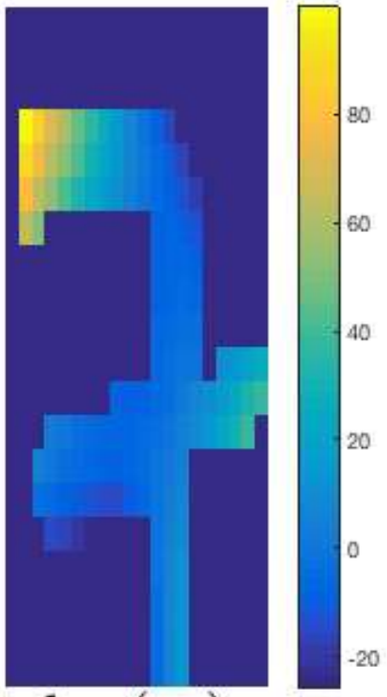
Ku_9



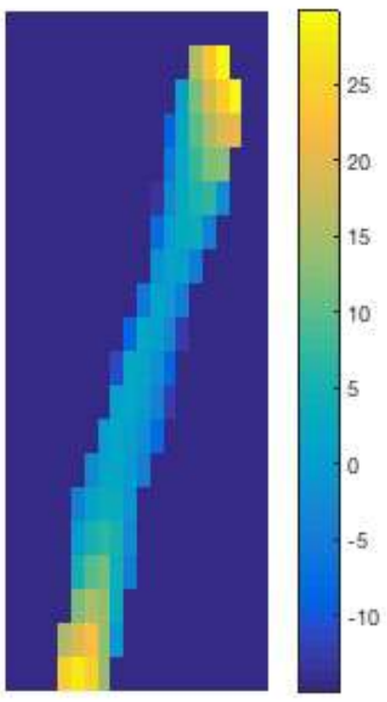
Kv_9



$\log(u_9)$

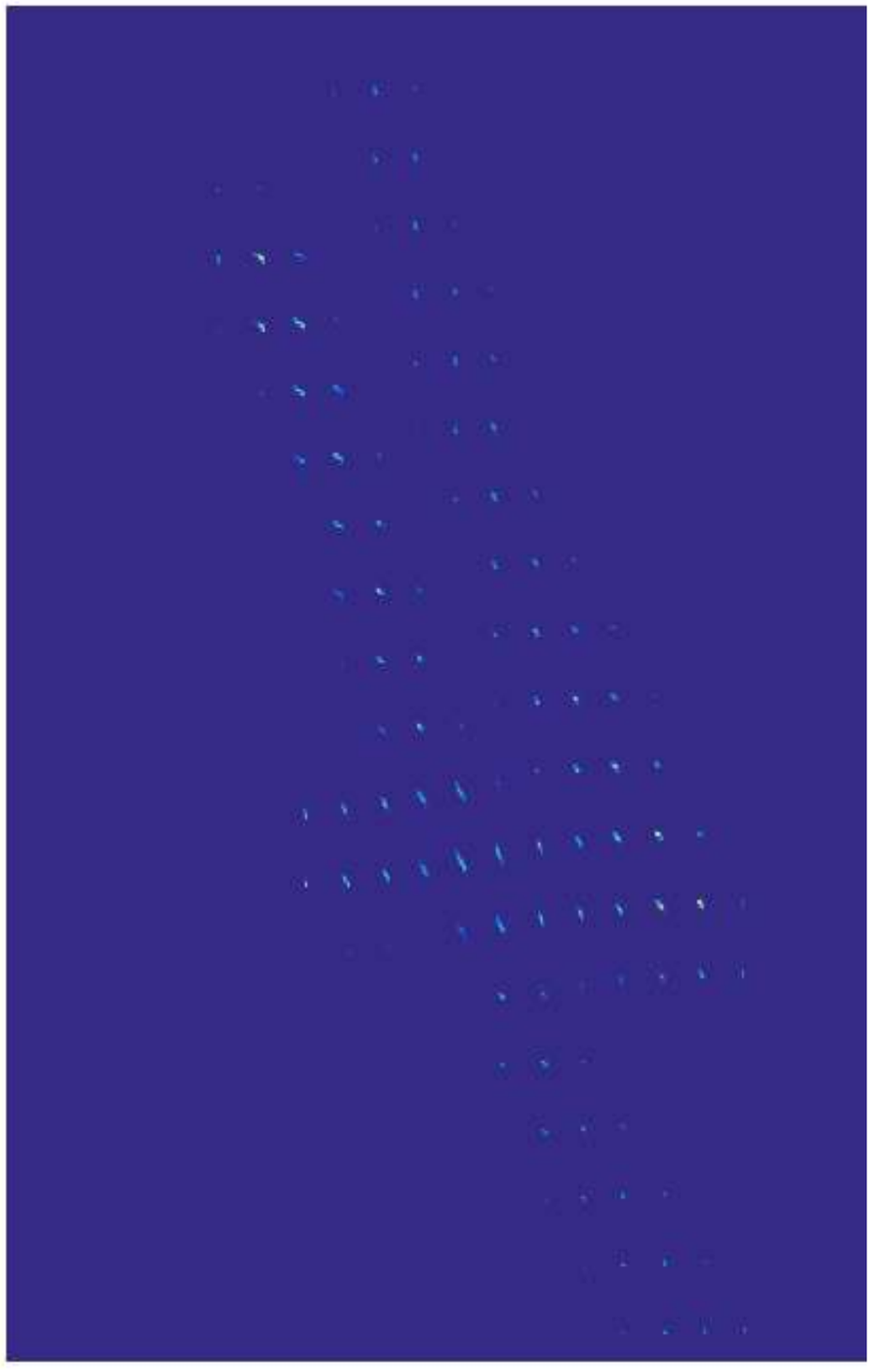


$\log(v_9)$



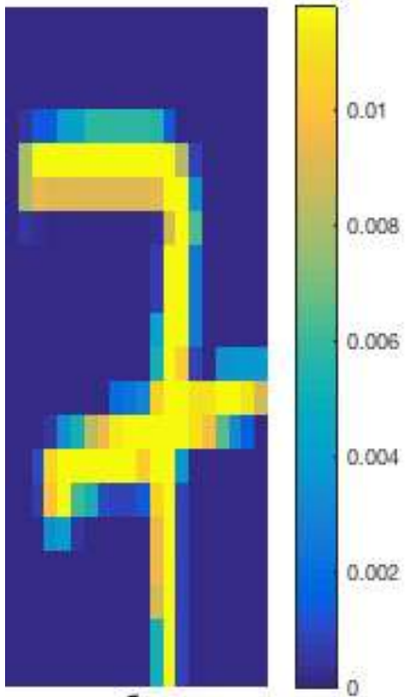
$P_8 = D(u_8)KD(v_8)$

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

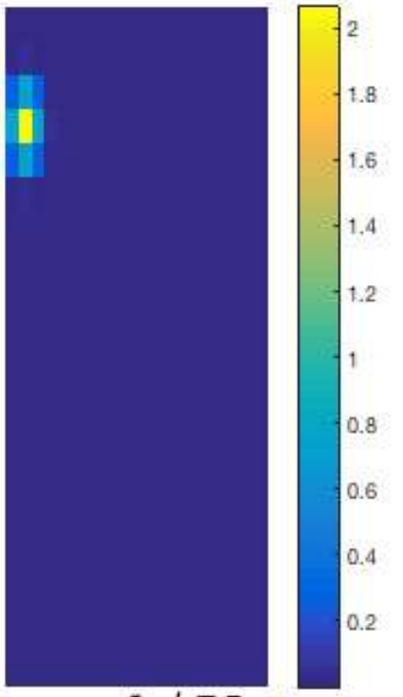


Very Fast EMD Approx. Solver

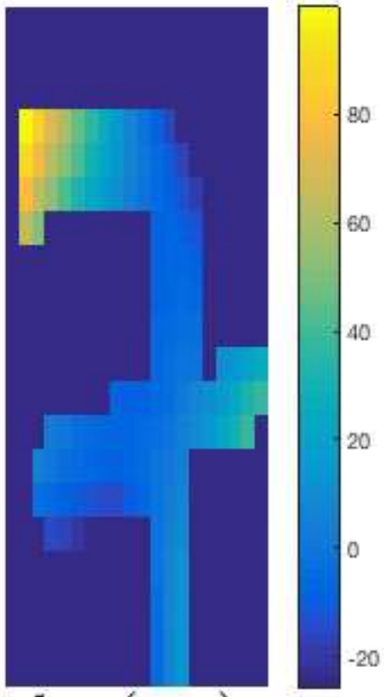
a



Ku_9

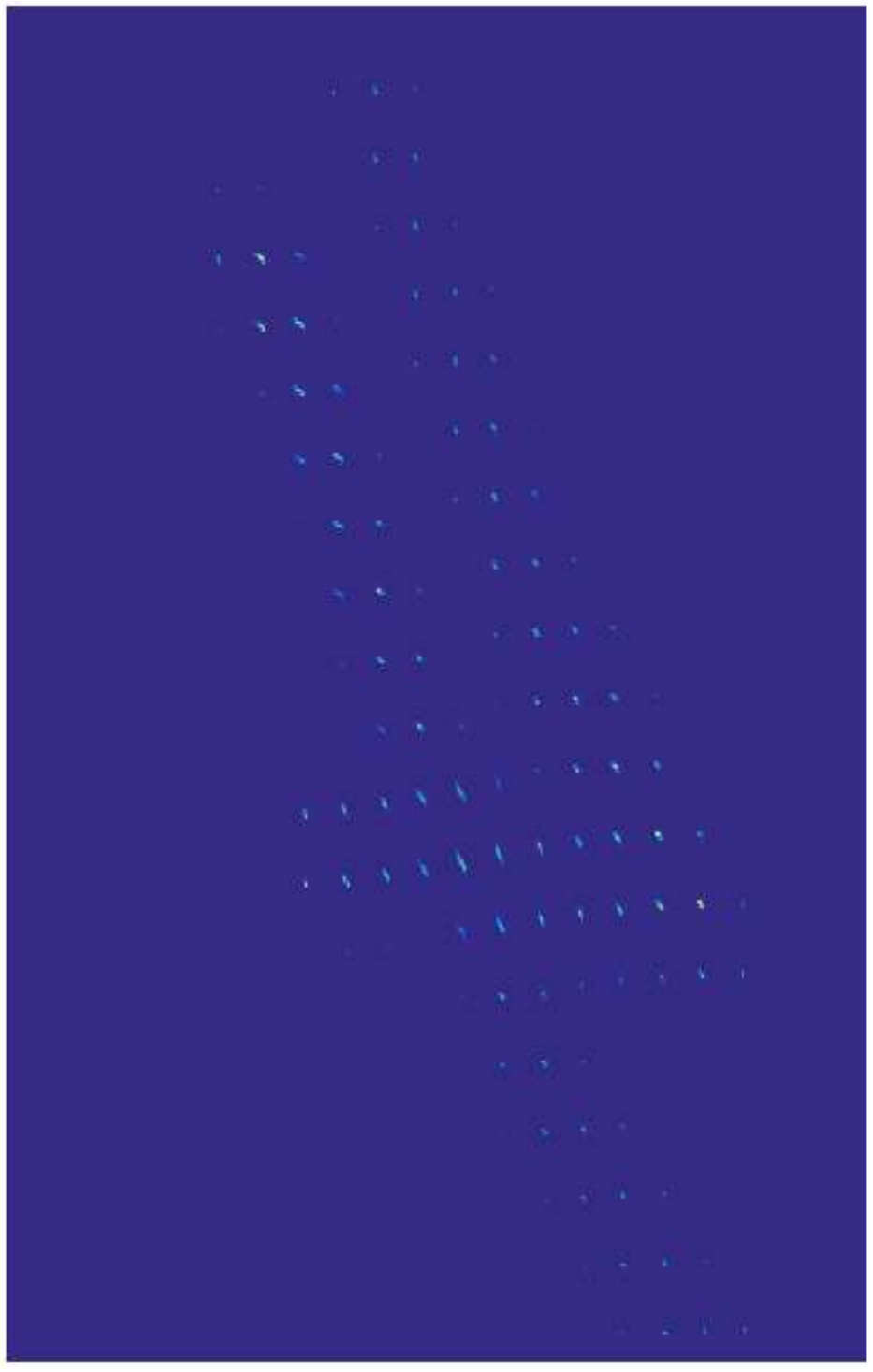


$\log(u_9)$

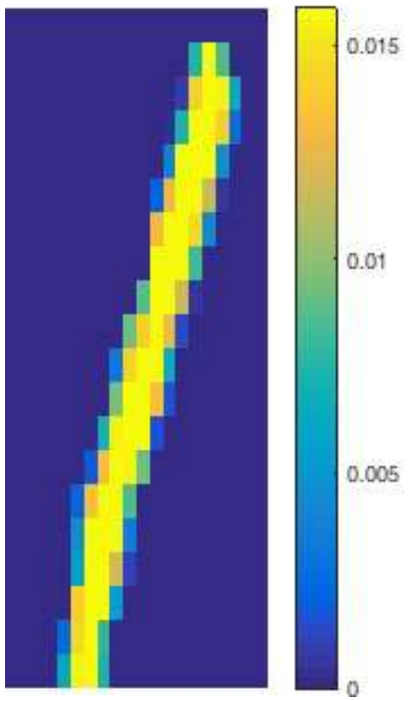


$P_9 = D(u_9)KD(v_9)$

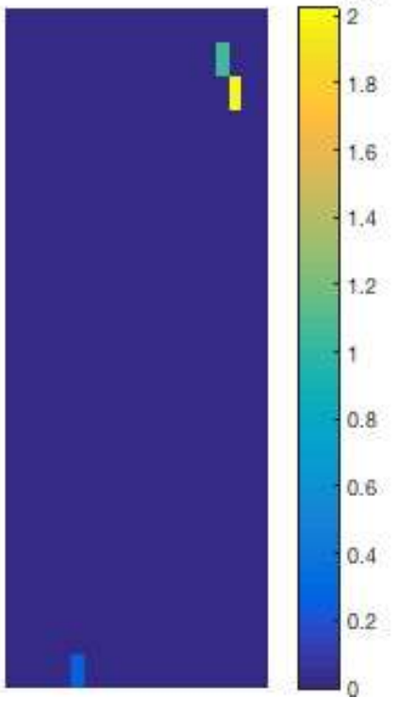
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$



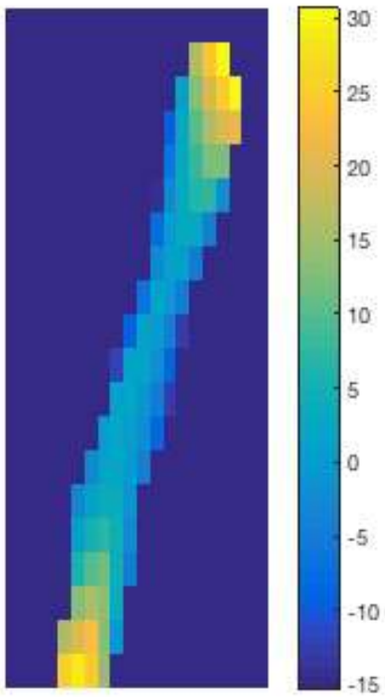
b



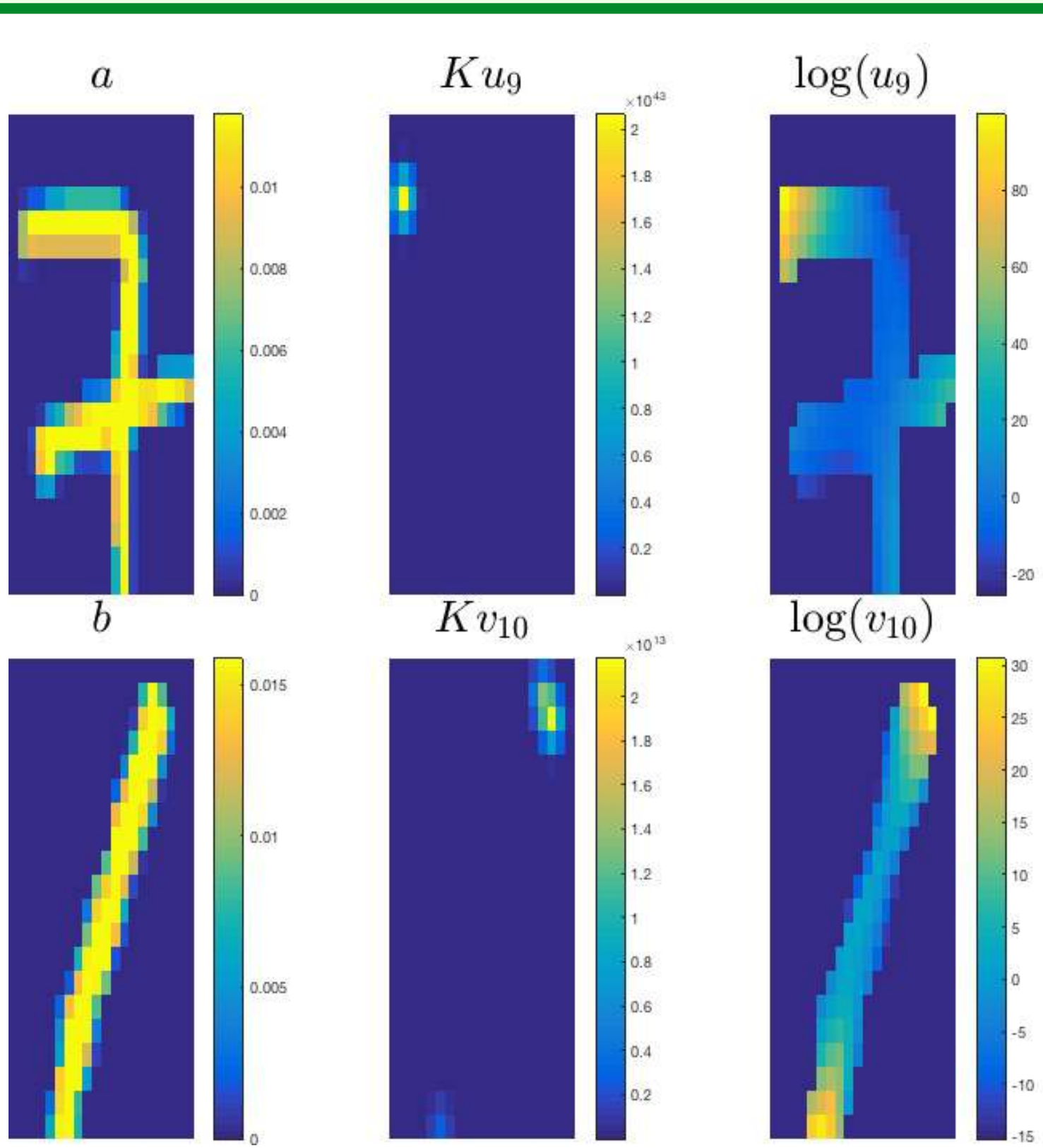
$v_{10} \leftarrow b/Ku_{10}$



$\log(v_{10})$

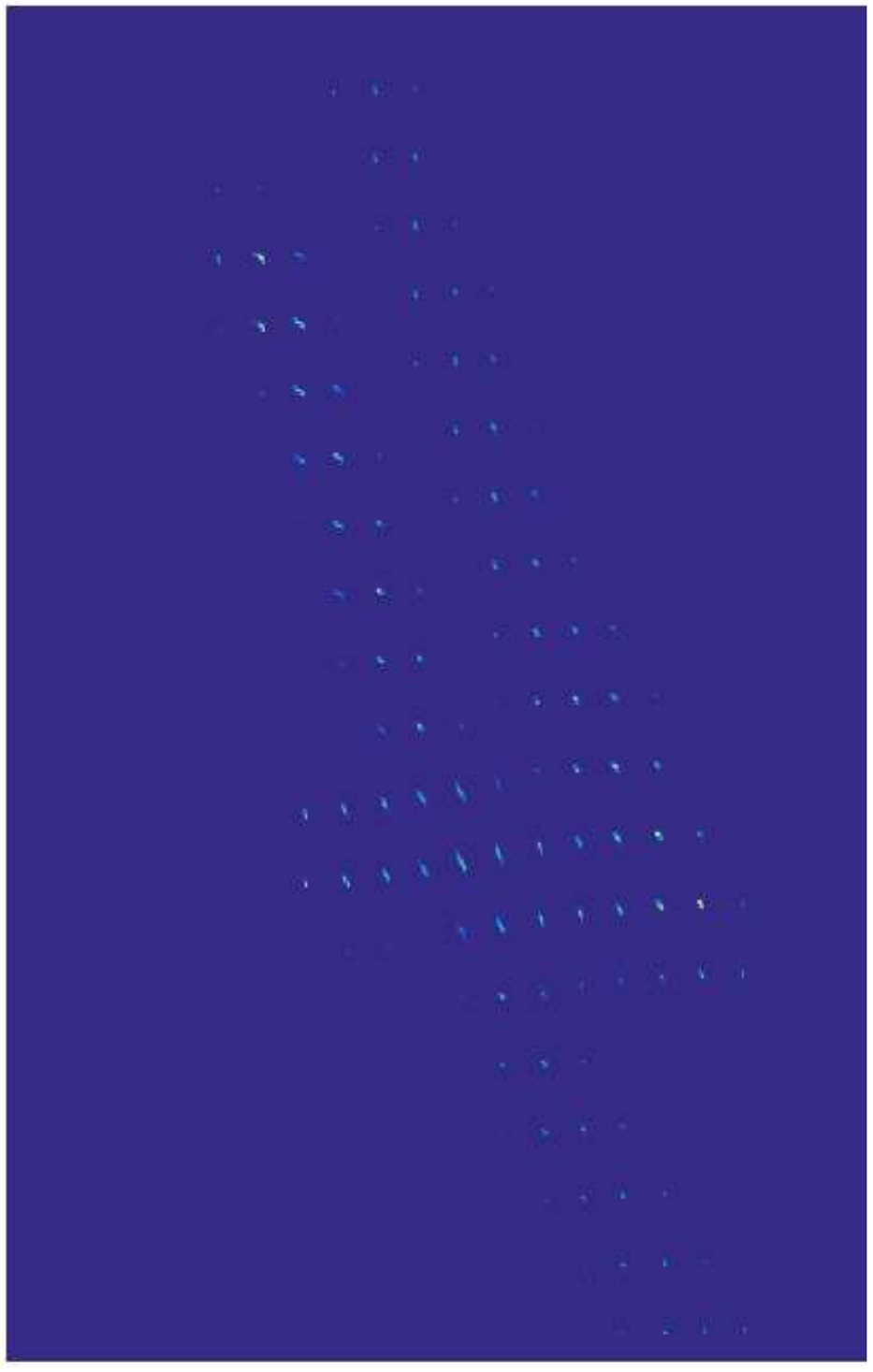


Very Fast EMD Approx. Solver



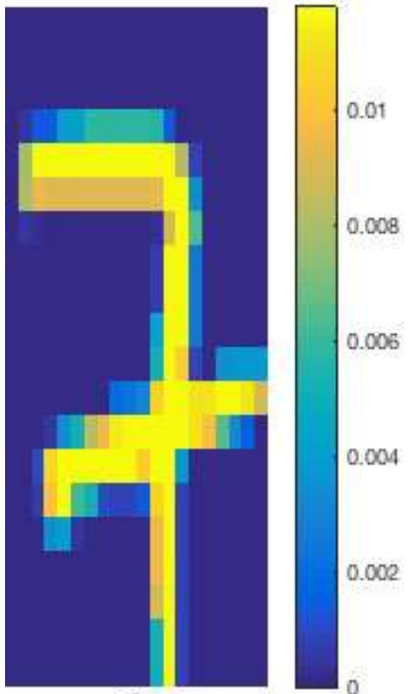
$P_9 = D(u_9)KD(v_9)$

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

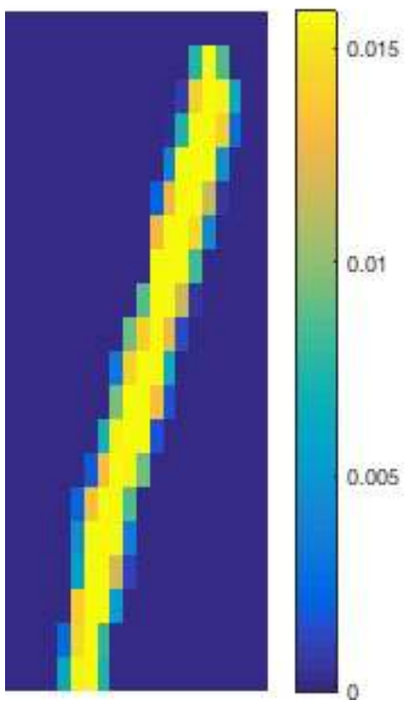


Very Fast EMD Approx. Solver

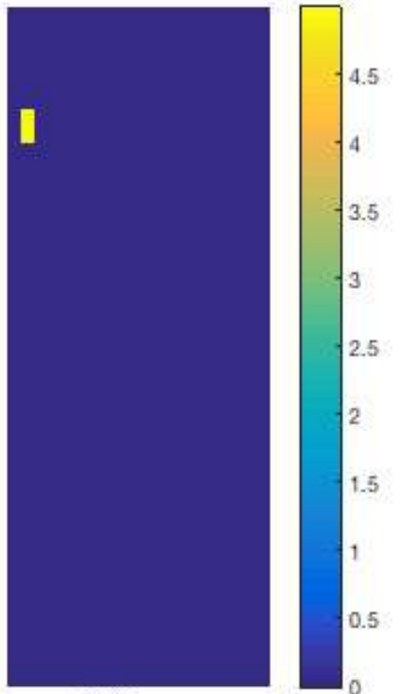
a



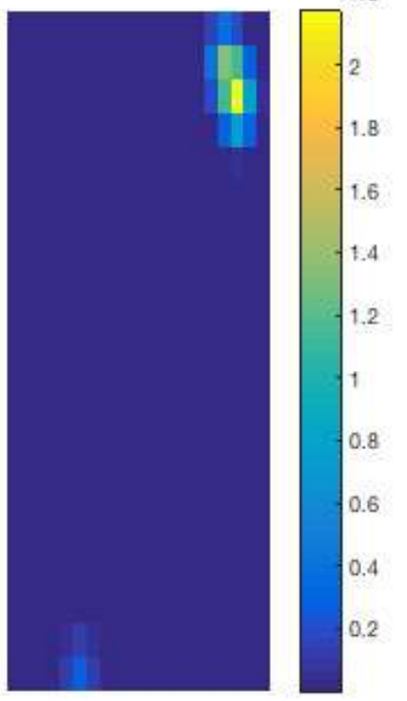
b



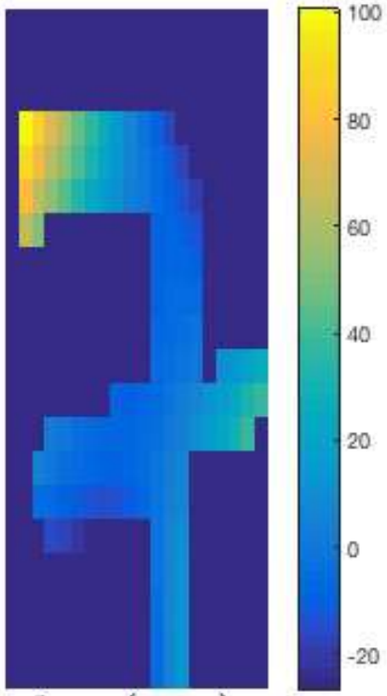
$u_{10} \leftarrow a/Kv_{10}$



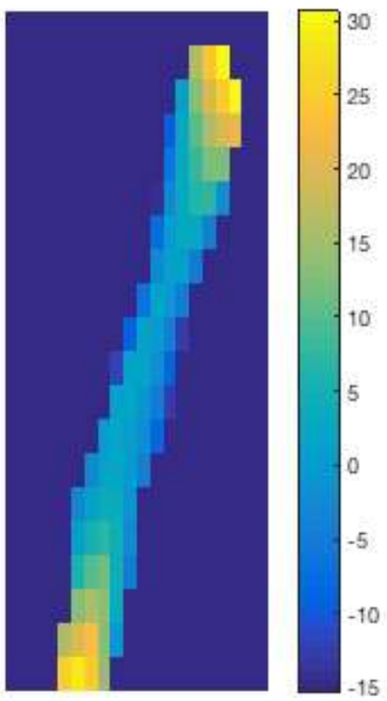
Kv_{10}



$\log(u_{10})$

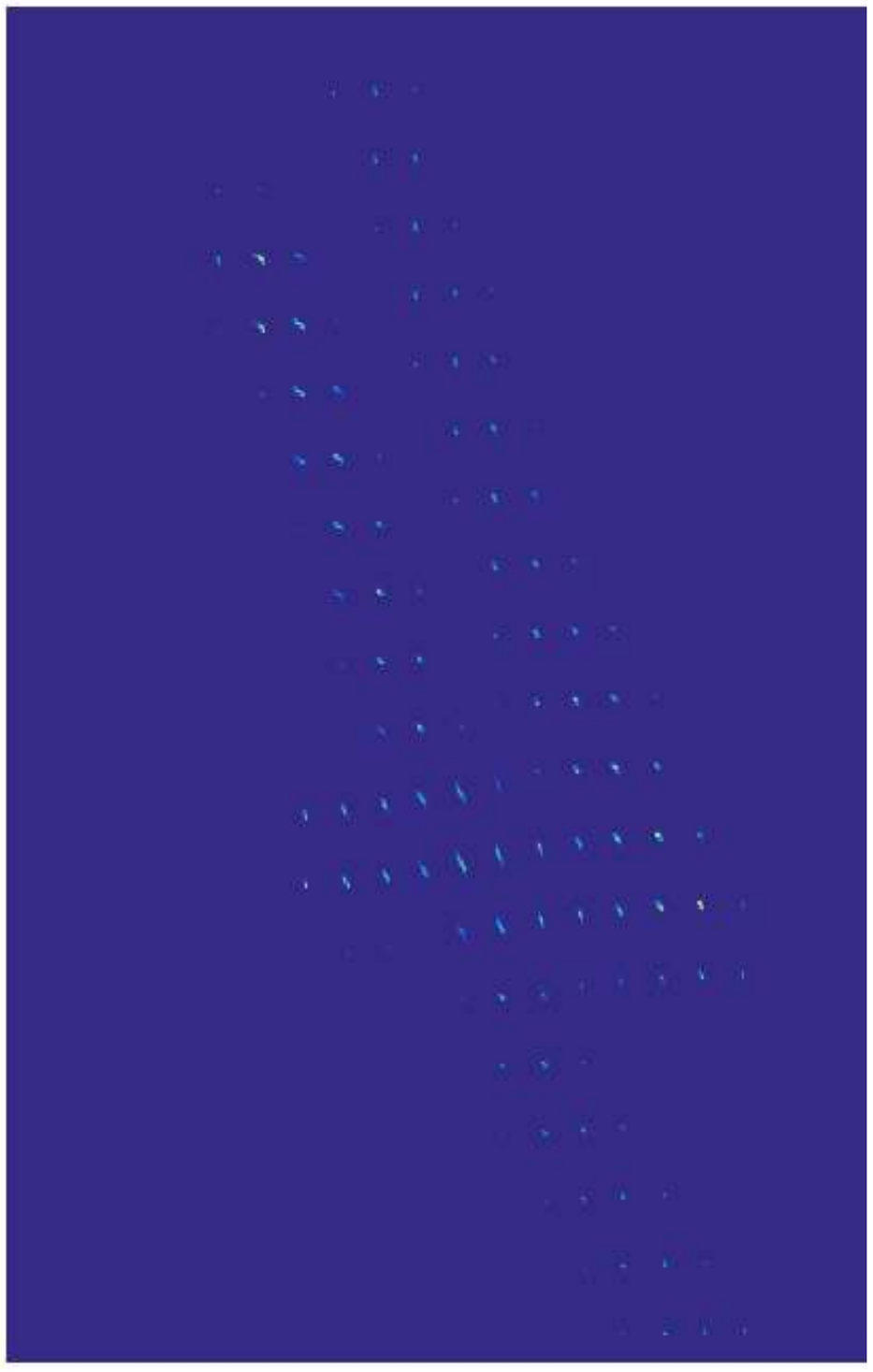


$\log(v_{10})$



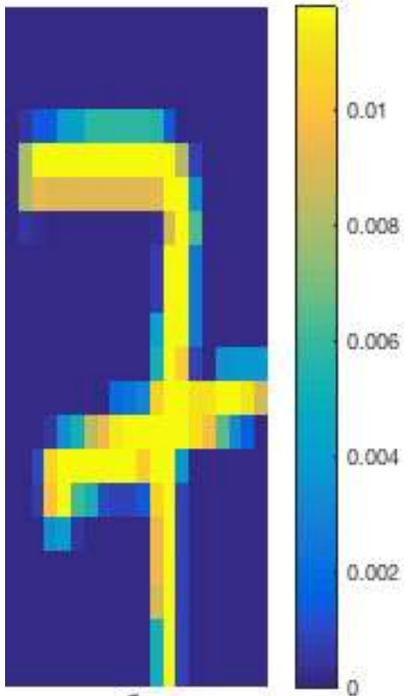
$P_9 = D(u_9)KD(v_9)$

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

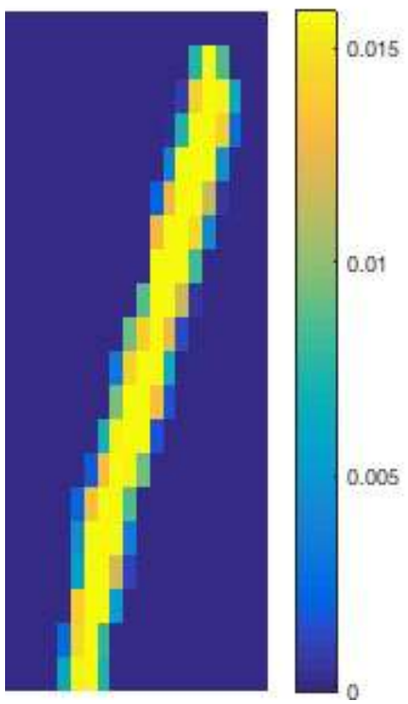


Very Fast EMD Approx. Solver

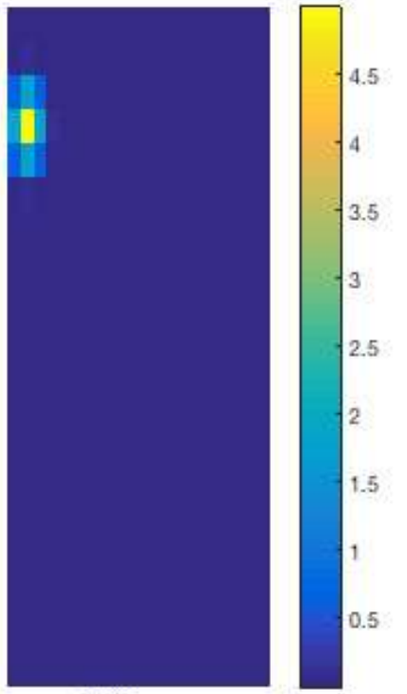
a



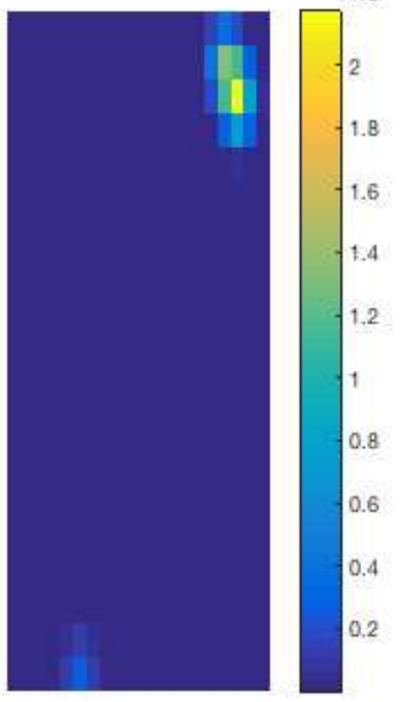
b



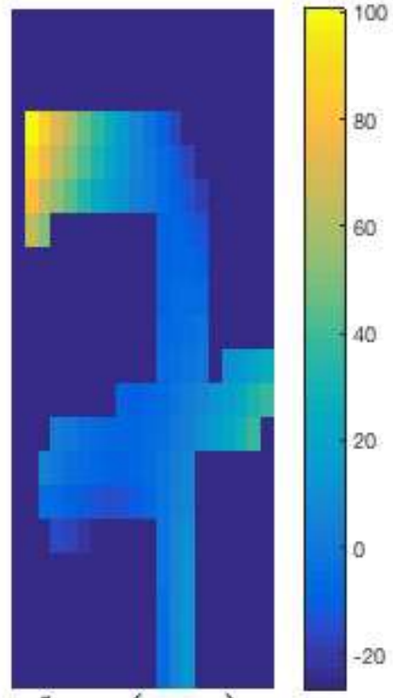
Ku_{10}



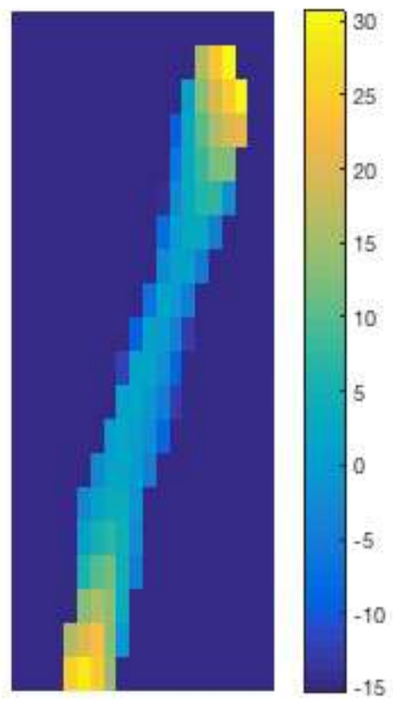
Kv_{10}



$\log(u_{10})$

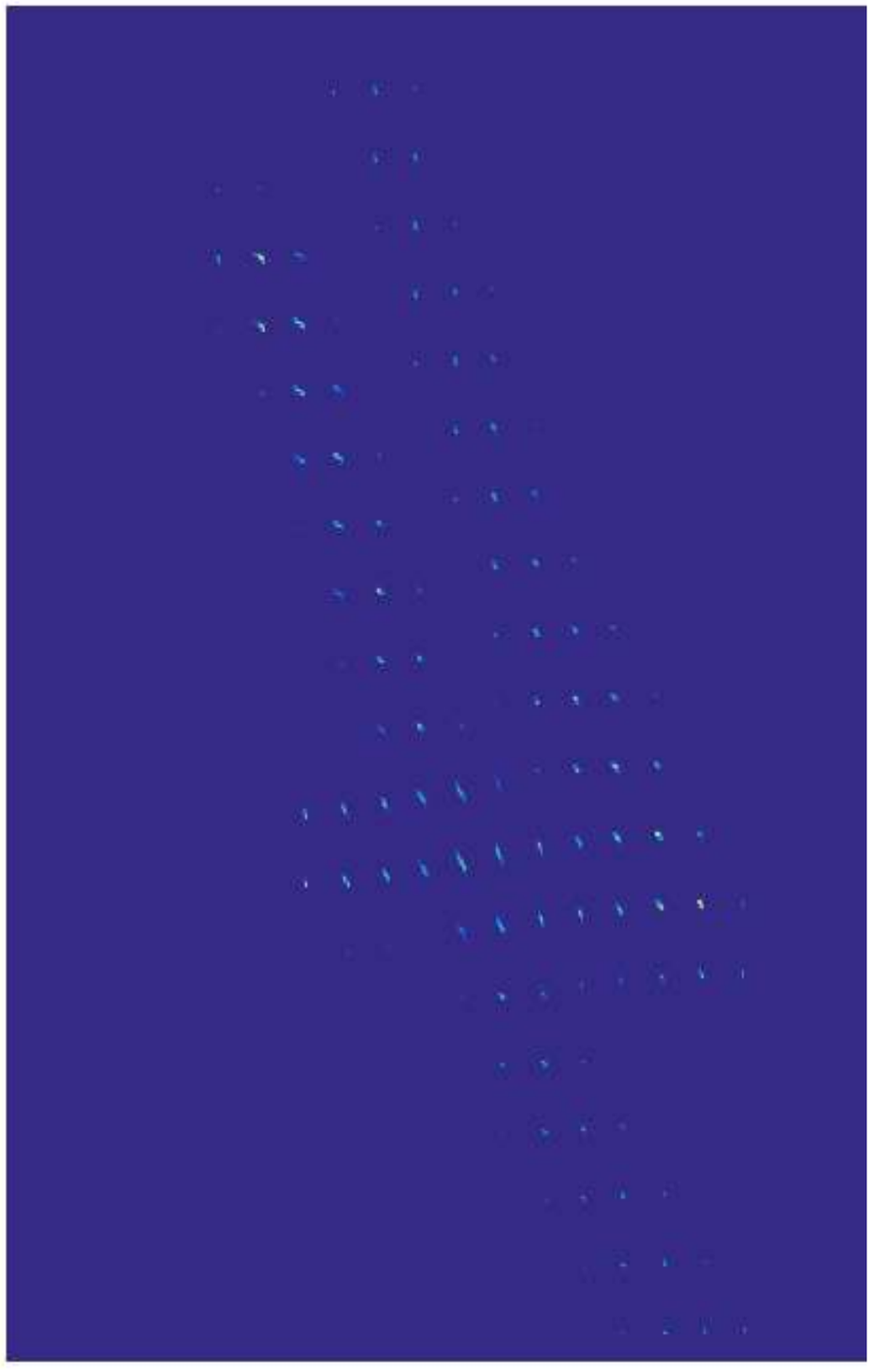


$\log(v_{10})$



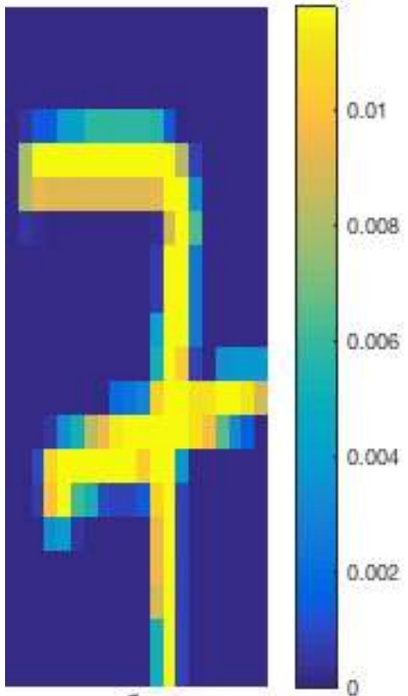
$P_9 = D(u_9)KD(v_9)$

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

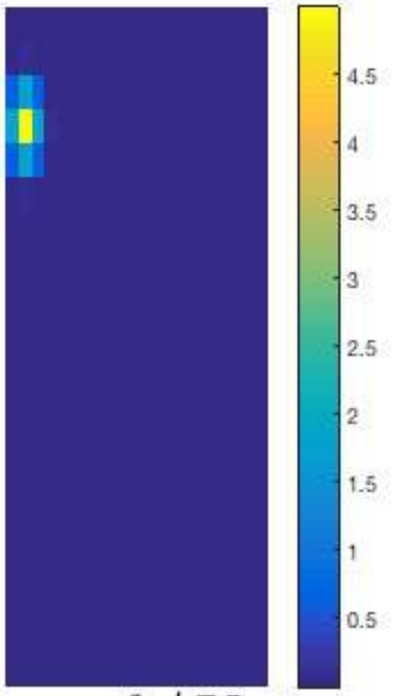


Very Fast EMD Approx. Solver

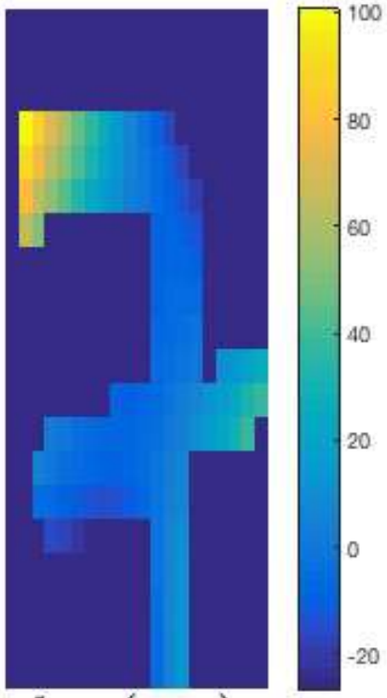
a



Ku_{10}

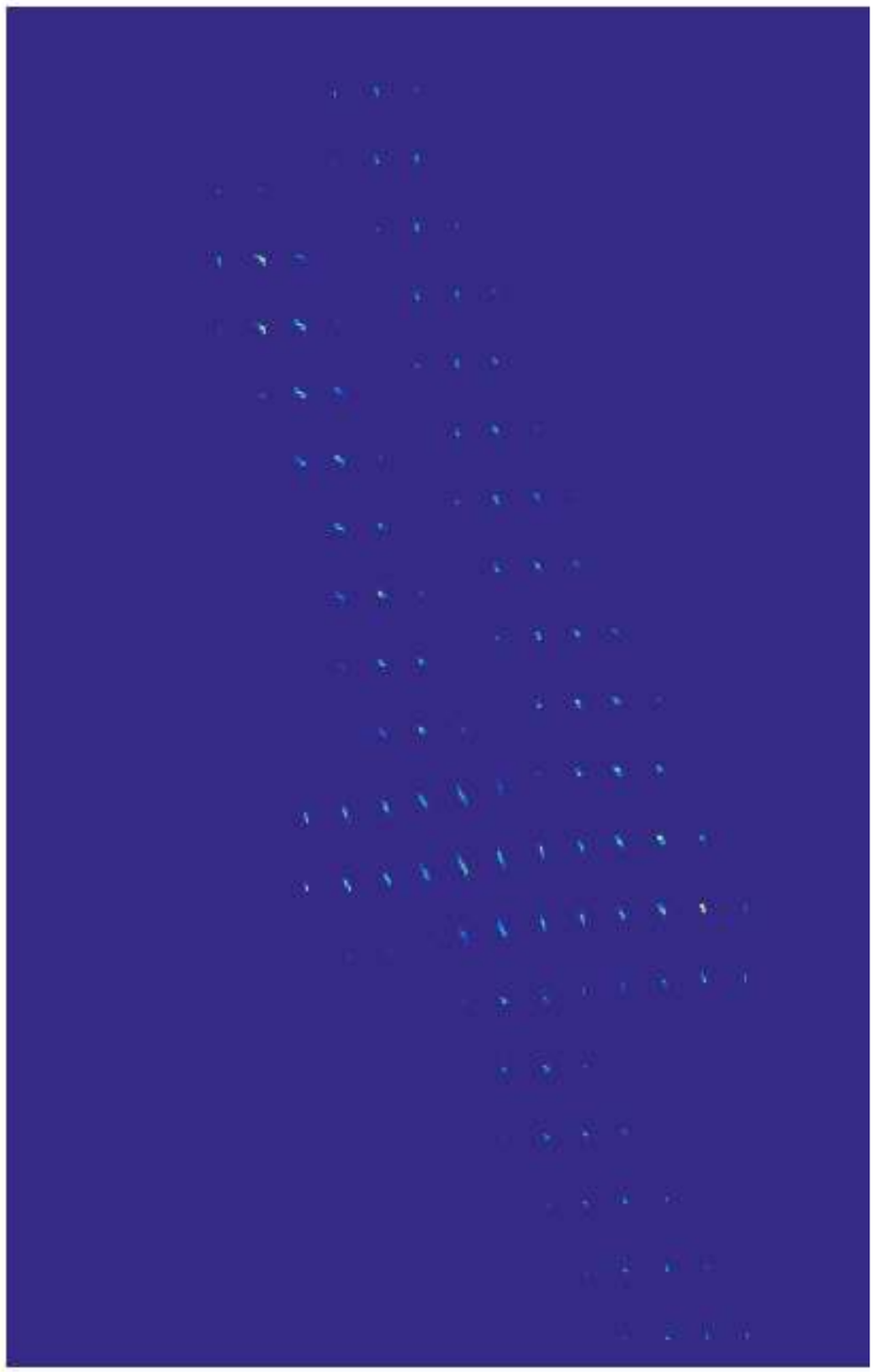


$\log(u_{10})$

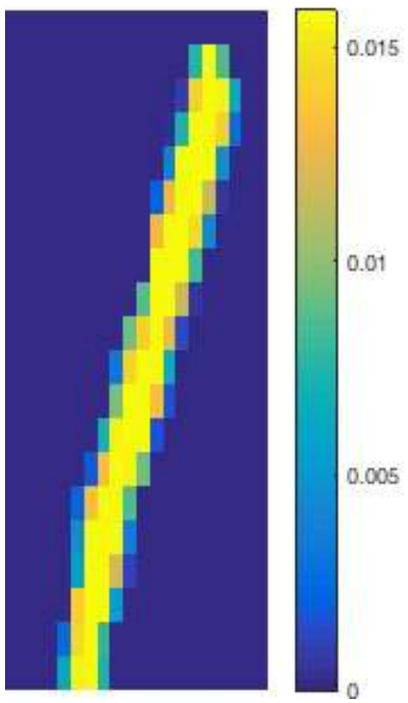


$P_{10} = D(u_{10})KD(v_{10})$

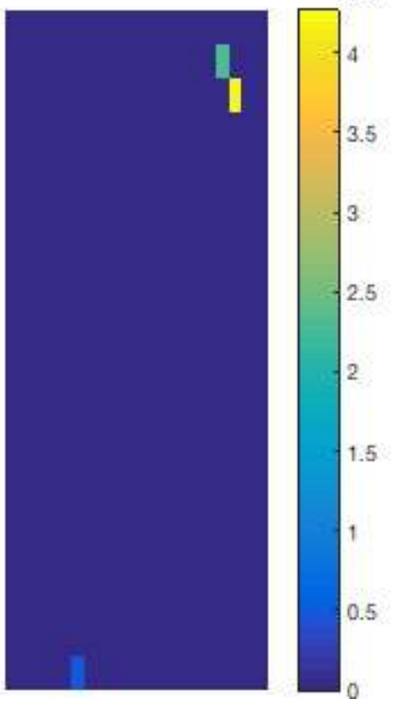
$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.29009$



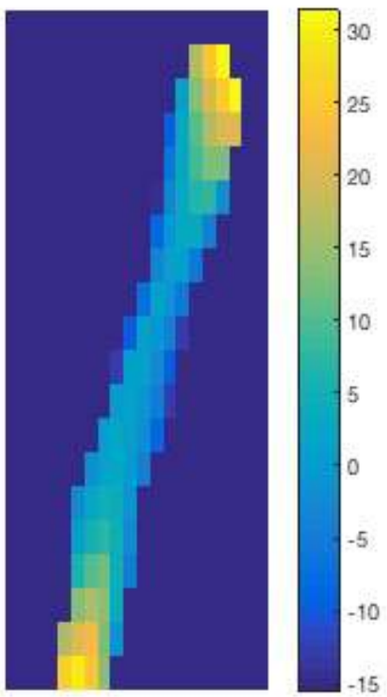
b



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



$\log(v_{11})$



Sinkhorn as a Dual Algorithm

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

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

REGULARIZED DISCRETE PRIMAL

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

$$\text{where} \quad K = \left[e^{-\frac{\boldsymbol{D}^p(\boldsymbol{x_i}, \boldsymbol{y_j})}{\gamma}} \right]_{ij}$$

REGULARIZED DISCRETE DUAL

Sinkhorn = *Block Coordinate Ascent* on Dual

Block Coordinate Ascent, *a.k.a Sinkhorn*

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

REGULARIZED DISCRETE DUAL

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

Block Coordinate Ascent, *a.k.a Sinkhorn*

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

REGULARIZED DISCRETE DUAL

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

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

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

Block Coordinate Ascent, *a.k.a Sinkhorn*

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

REGULARIZED DISCRETE DUAL

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

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

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

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

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

Block Coordinate Ascent, *a.k.a Sinkhorn*

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

REGULARIZED DISCRETE DUAL

Block Coordinate Ascent, *a.k.a Sinkhorn*

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

REGULARIZED DISCRETE DUAL

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

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

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

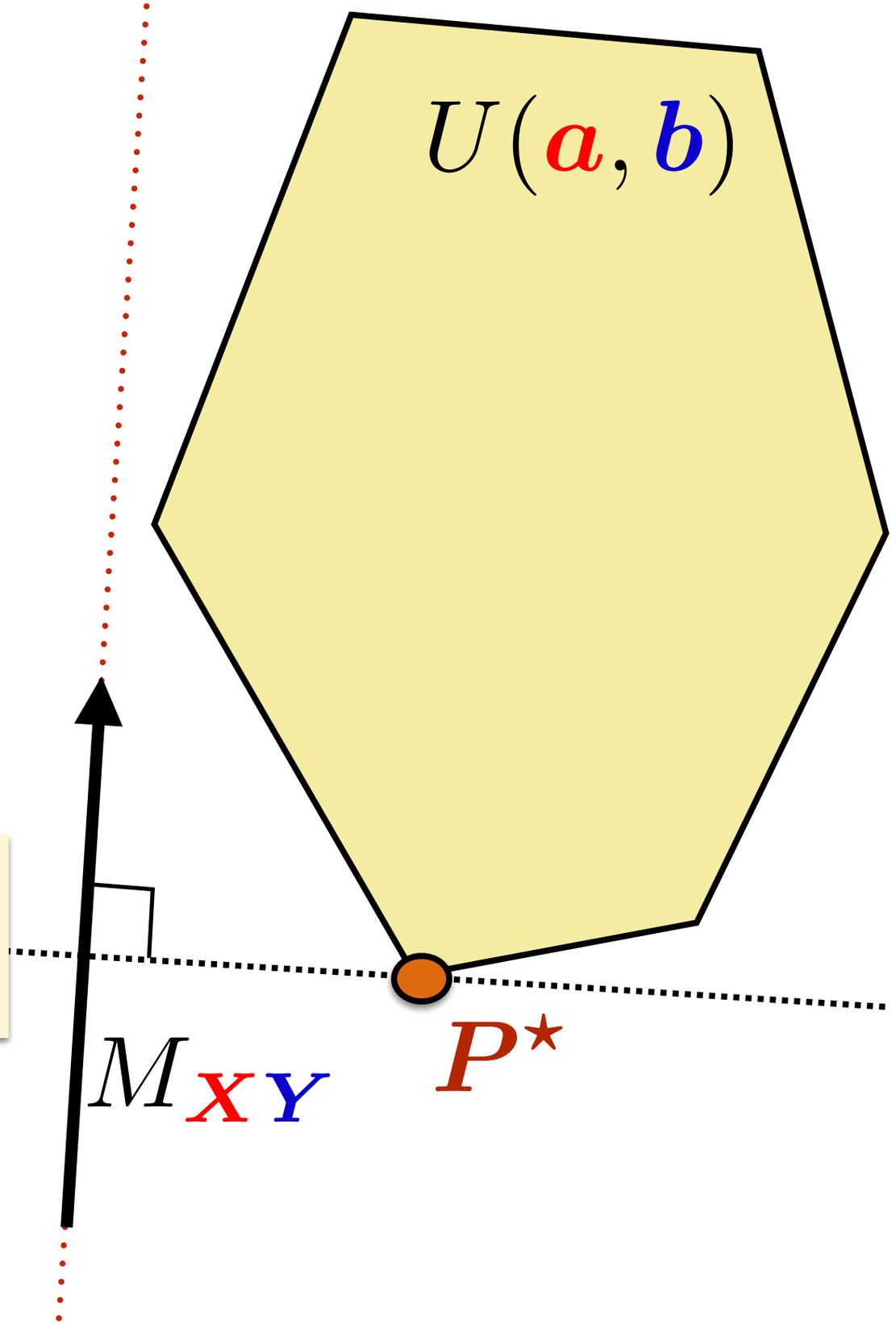
$$\boldsymbol{\beta} \leftarrow \gamma \left(\log \boldsymbol{b} - \log \ K^T (e^{\boldsymbol{\alpha}/\gamma}) \right)$$

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

Sinkhorn, W and *Energy Distance*

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

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

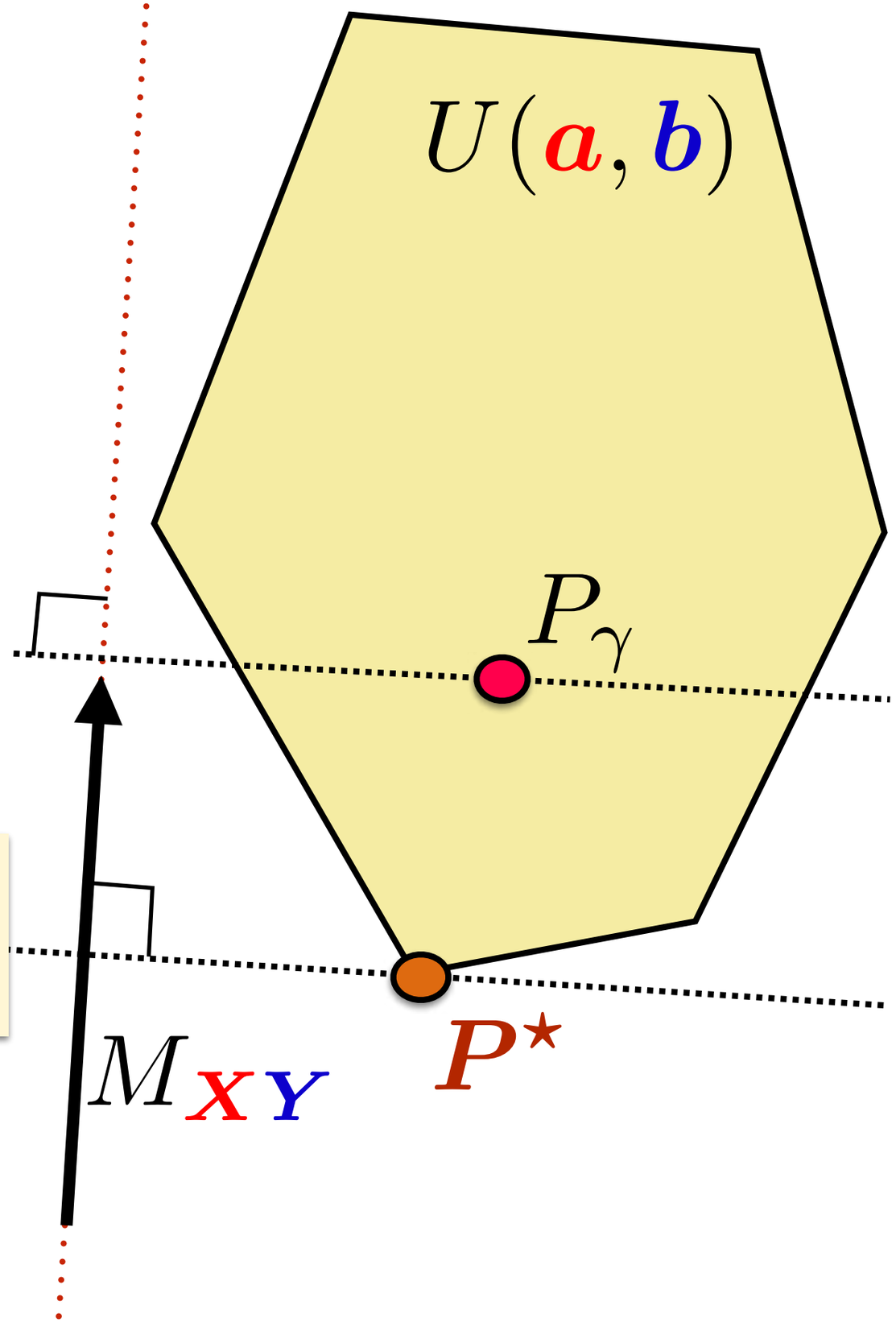


Sinkhorn, W and *Energy Distance*

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

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

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



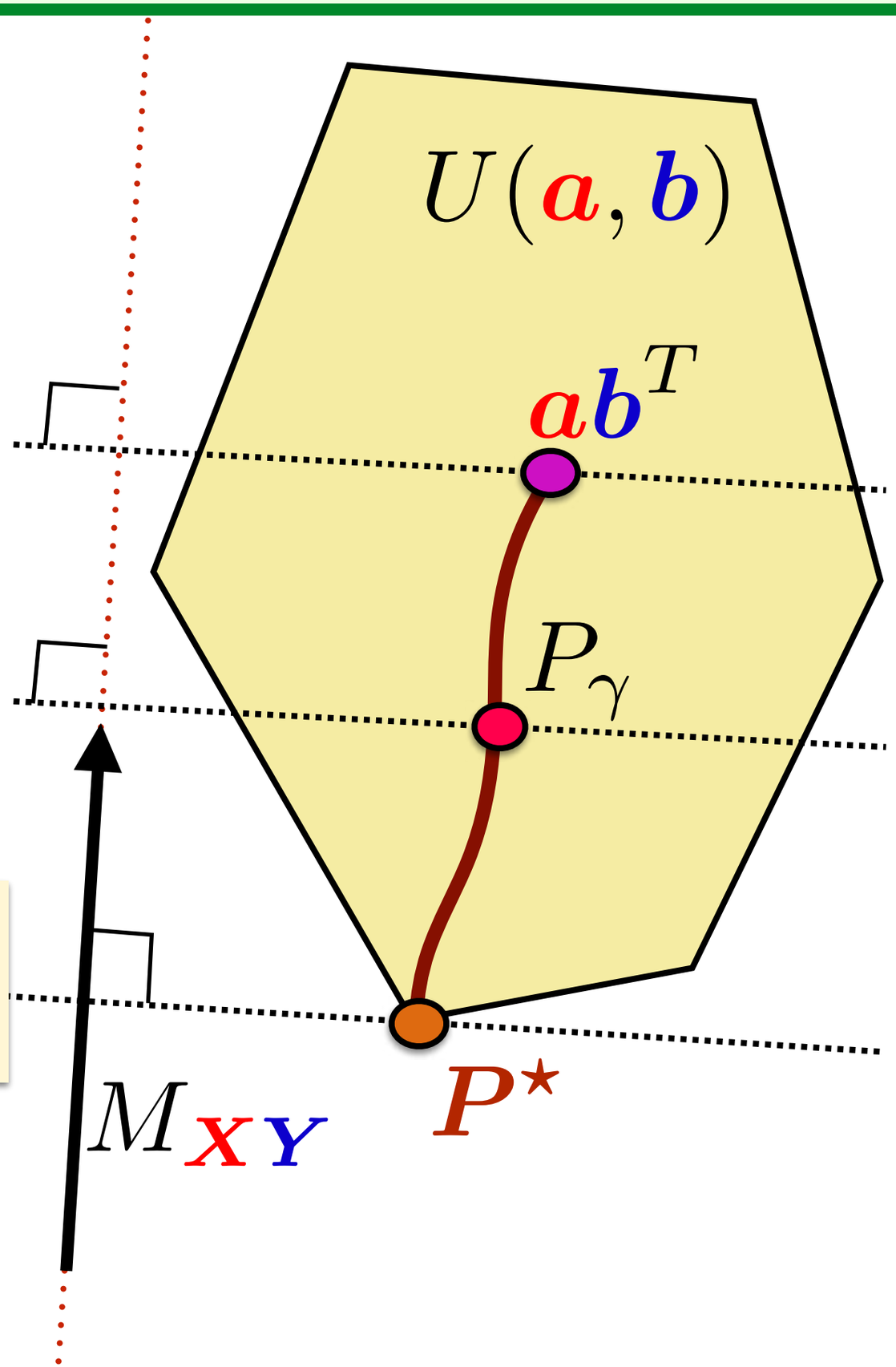
Sinkhorn, W and *Energy Distance*

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

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

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

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



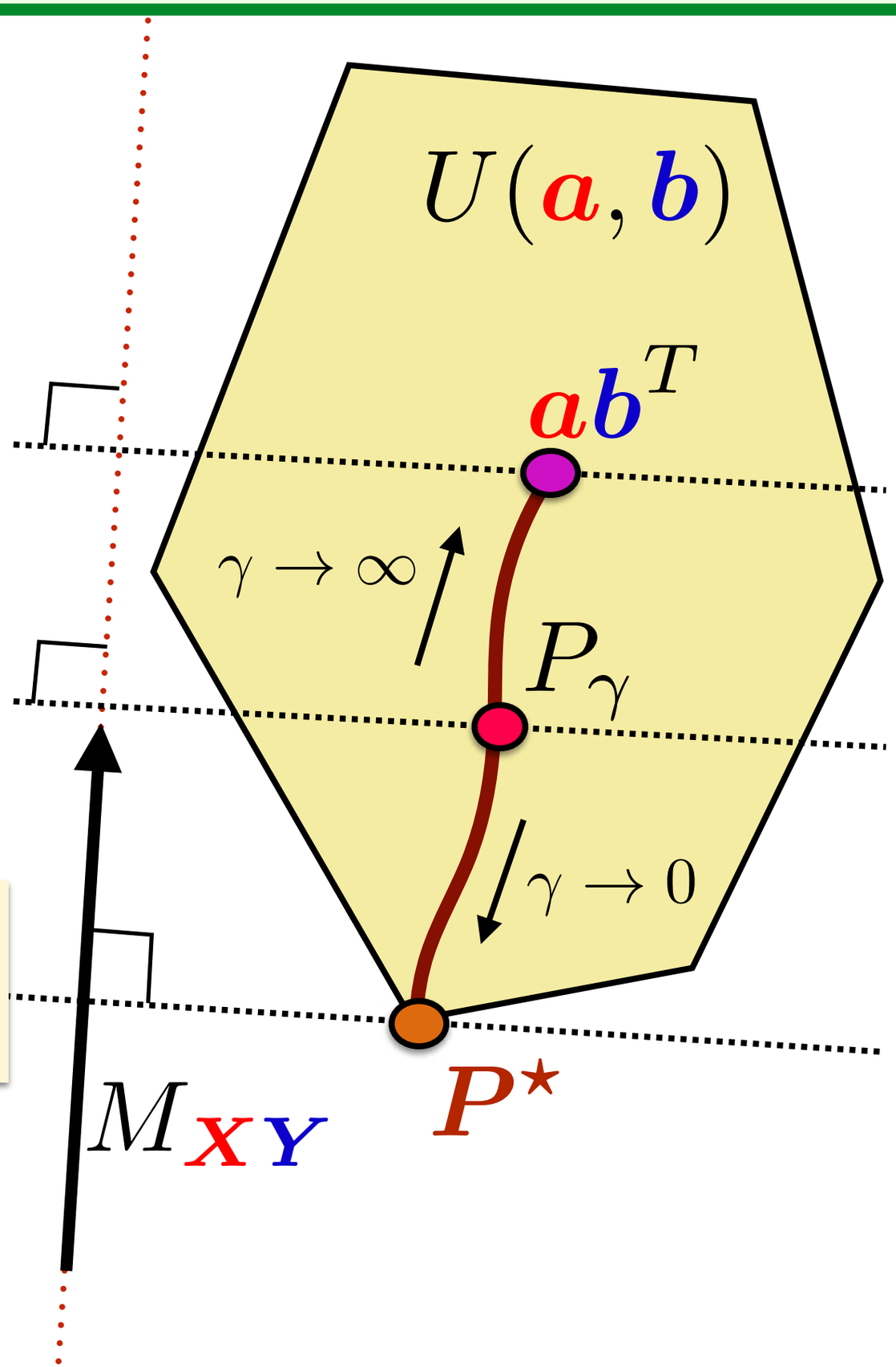
Sinkhorn, W and *Energy Distance*

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

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

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

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



Sinkhorn, W and *Energy Distance*

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

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

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

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

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

Sinkhorn, W and *Energy Distance*

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

$\gamma \to \infty$ ↑

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

$\gamma \to 0$ ↓

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

How to compare them?

i.i.d samples $\boldsymbol{x_1, \ldots, x_n \sim \mu}$, $\boldsymbol{y_1, \ldots, y_m \sim \nu}$,

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

Computational properties

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

Statistical properties

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

Sinkhorn in between W and MMD

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

$(n+m)^2$

$O(1/\sqrt{n})$ **[see Arthur]**

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

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

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

Sinkhorn in between W and MMD

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

$(n+m)^2$

$O(1/\sqrt{n})$ **[see Arthur]**

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

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

$O\left(\frac{1}{\gamma^{d/2}\sqrt{n}}\right)$ **[GCBCP'18]**

[FSVATP'18]

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

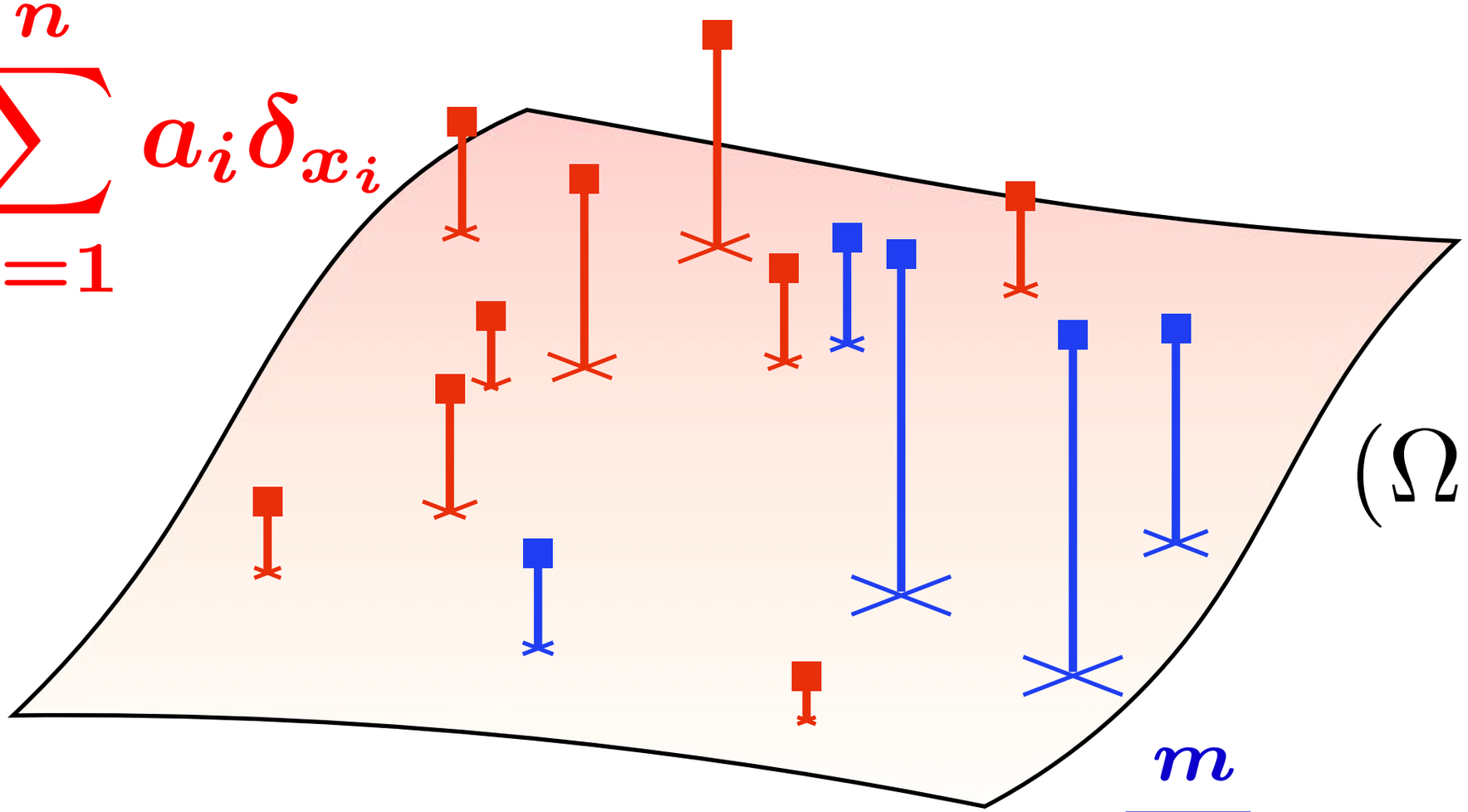
$O((n+m)nm\log(n+m))$

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

Differentiability of W

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

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



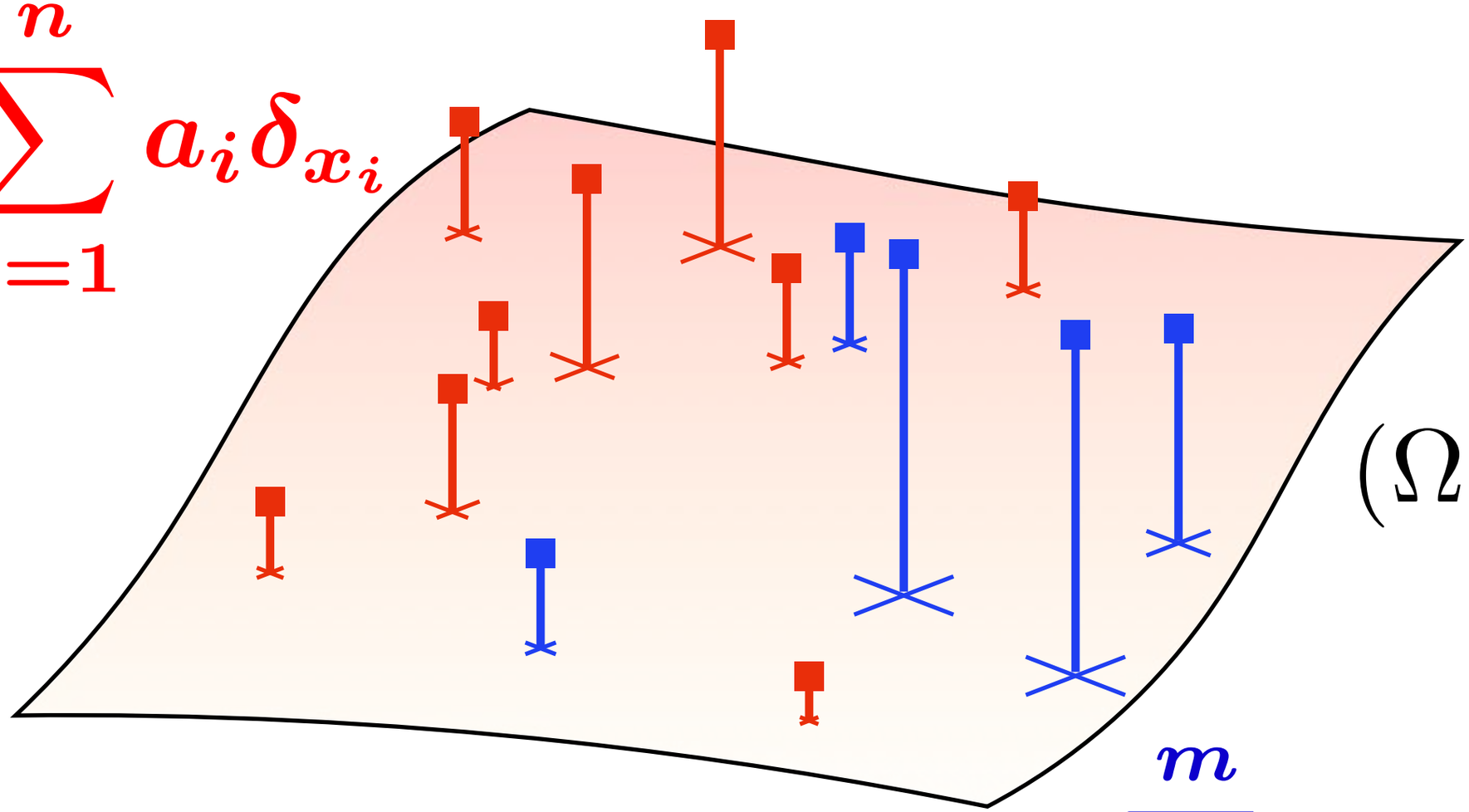
$$(\Omega, D)$$

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

Differentiability of W

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

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



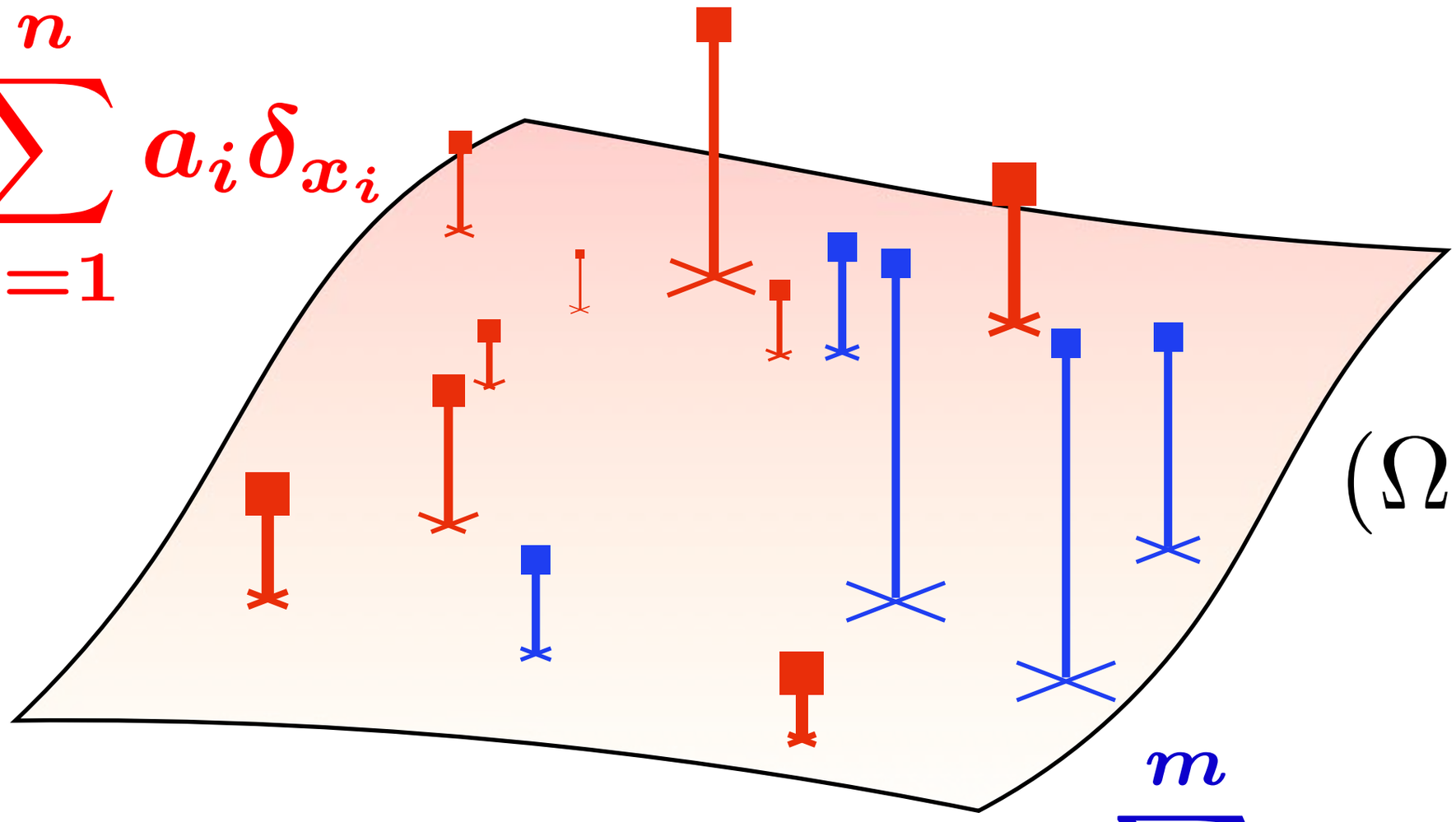
(Ω, D)

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

Differentiability of W

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

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



(Ω, D)

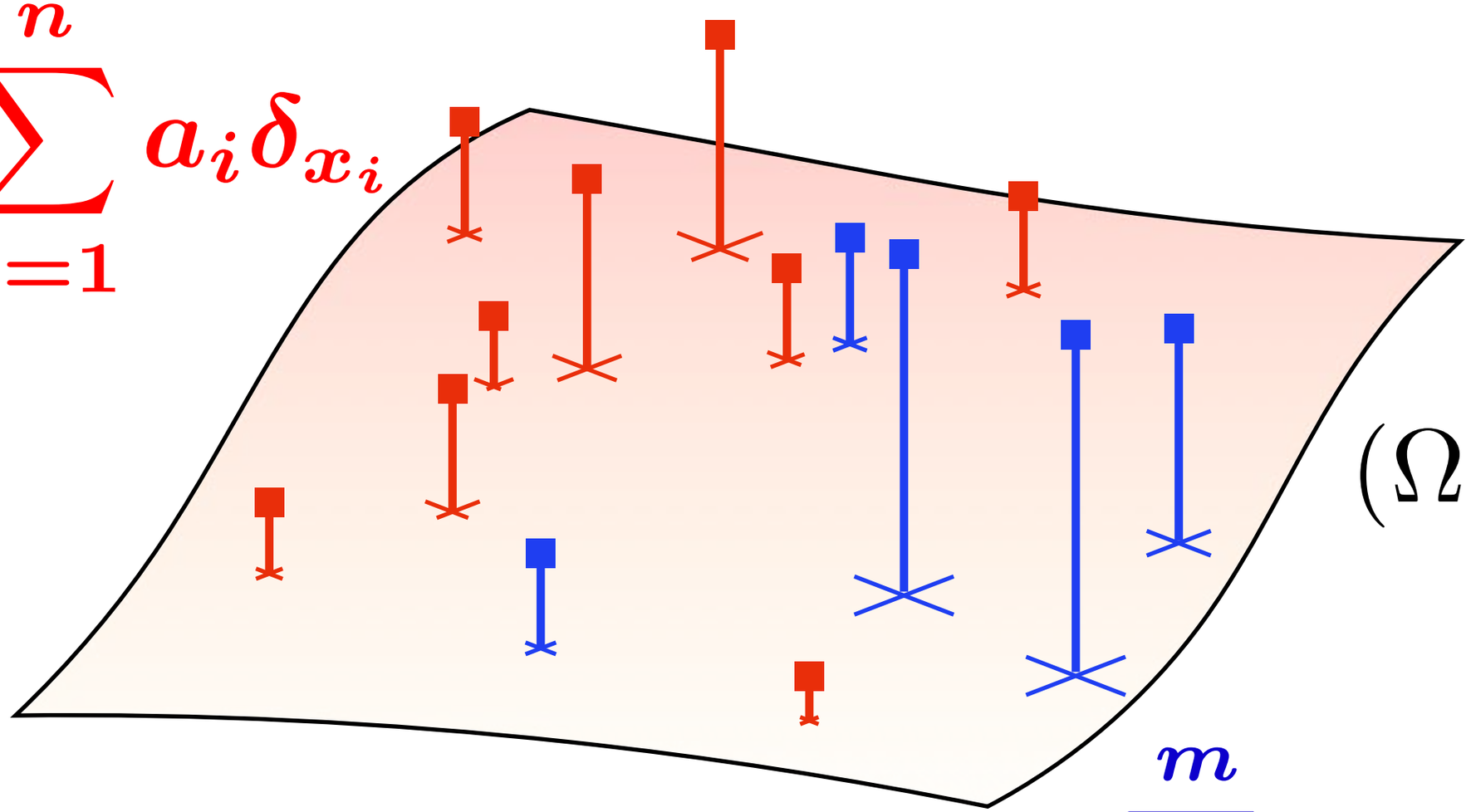
$$a \leftarrow a + \Delta a$$

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

Sinkhorn ⇢ *Differentiability*

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

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



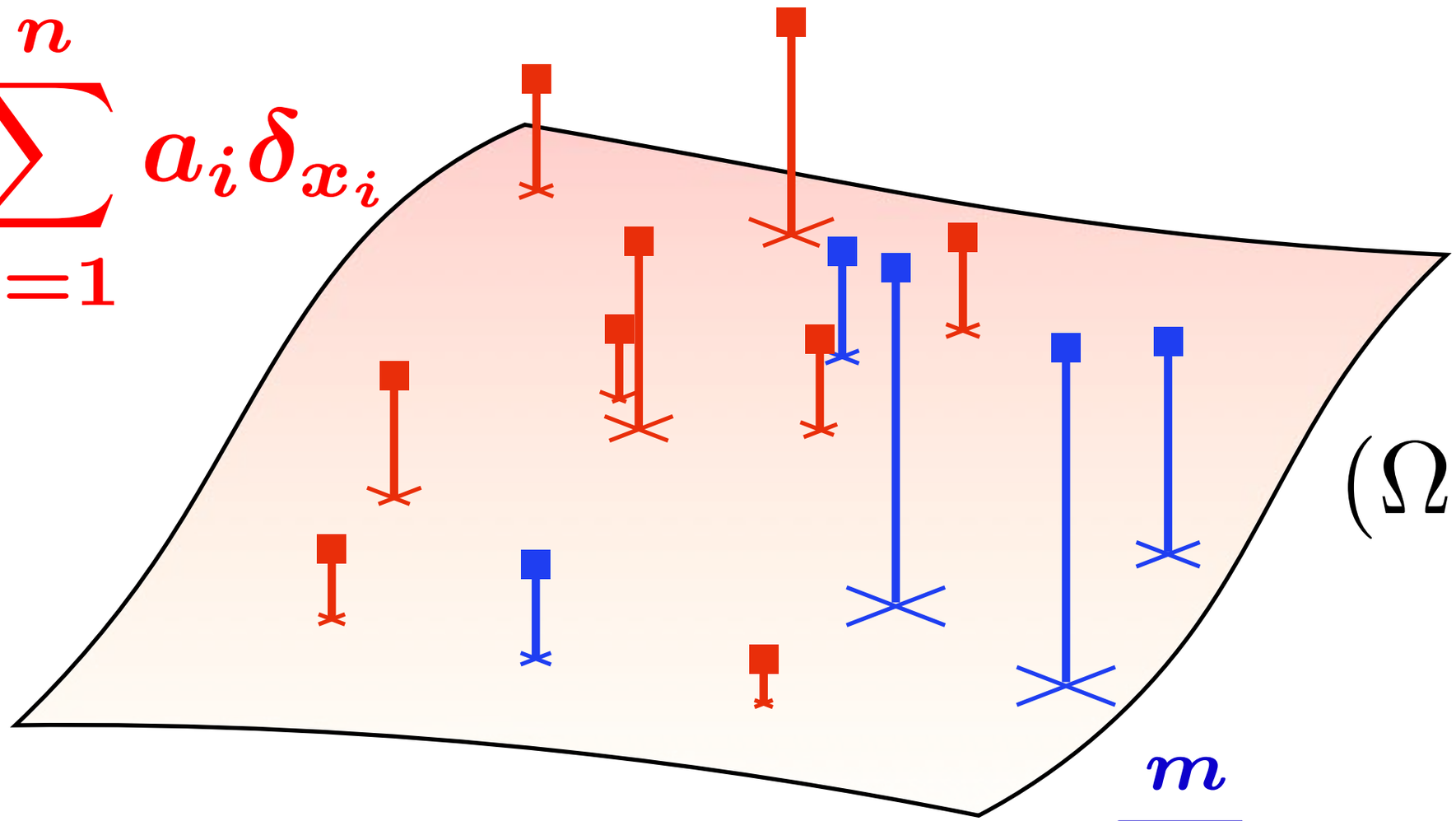
$$(\Omega, \boldsymbol{D})$$

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

Sinkhorn ⇢ *Differentiability*

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

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



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

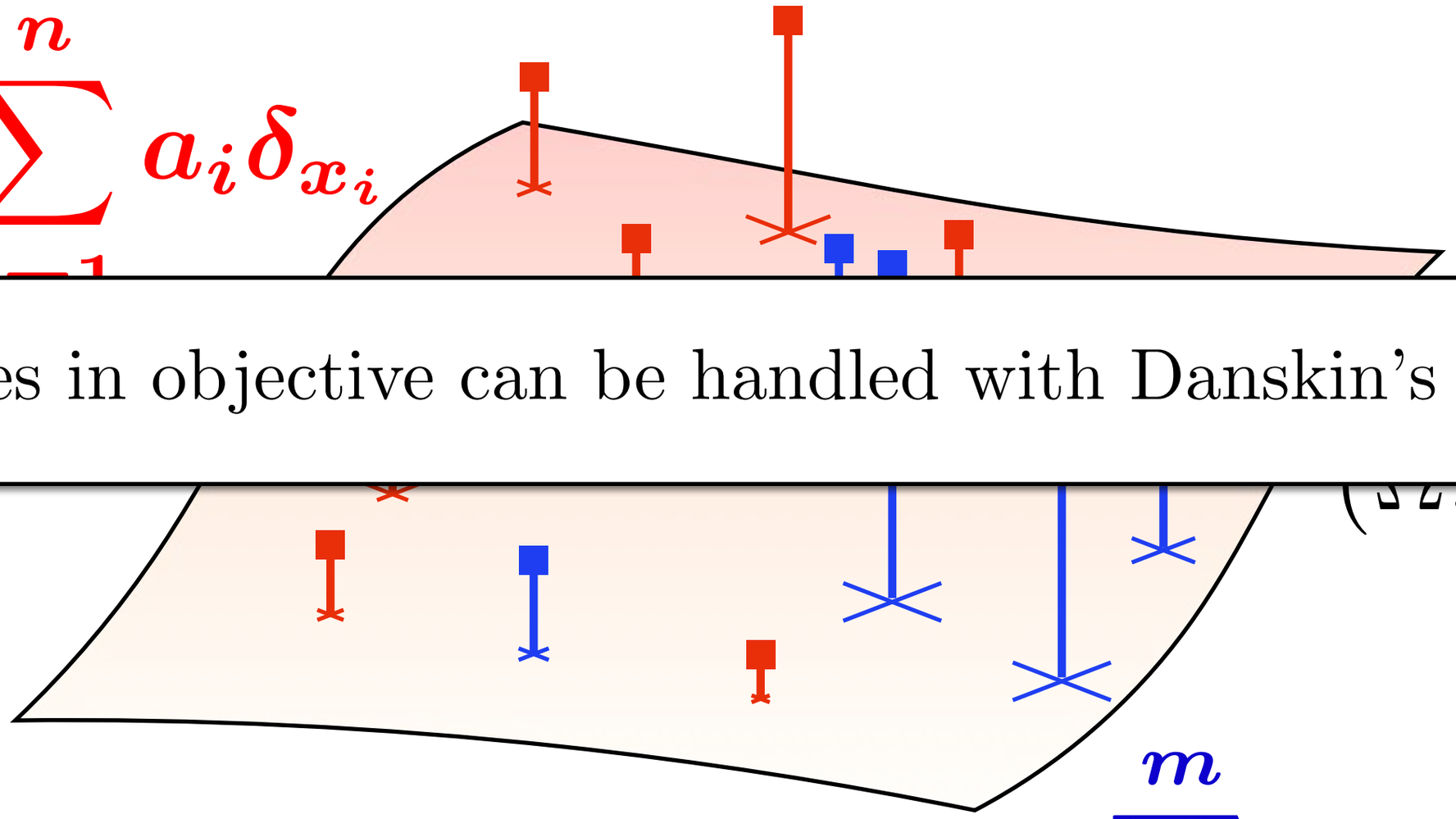
$$\boldsymbol{X} \leftarrow \boldsymbol{X} + \boldsymbol{\Delta X}$$

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

Sinkhorn ⇢ *Differentiability*

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

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



Changes in objective can be handled with Danskin's theorem.

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

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

How to decrease W ? change weights

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

DUAL

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

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

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

How to decrease W ? change locations

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

PRIMAL

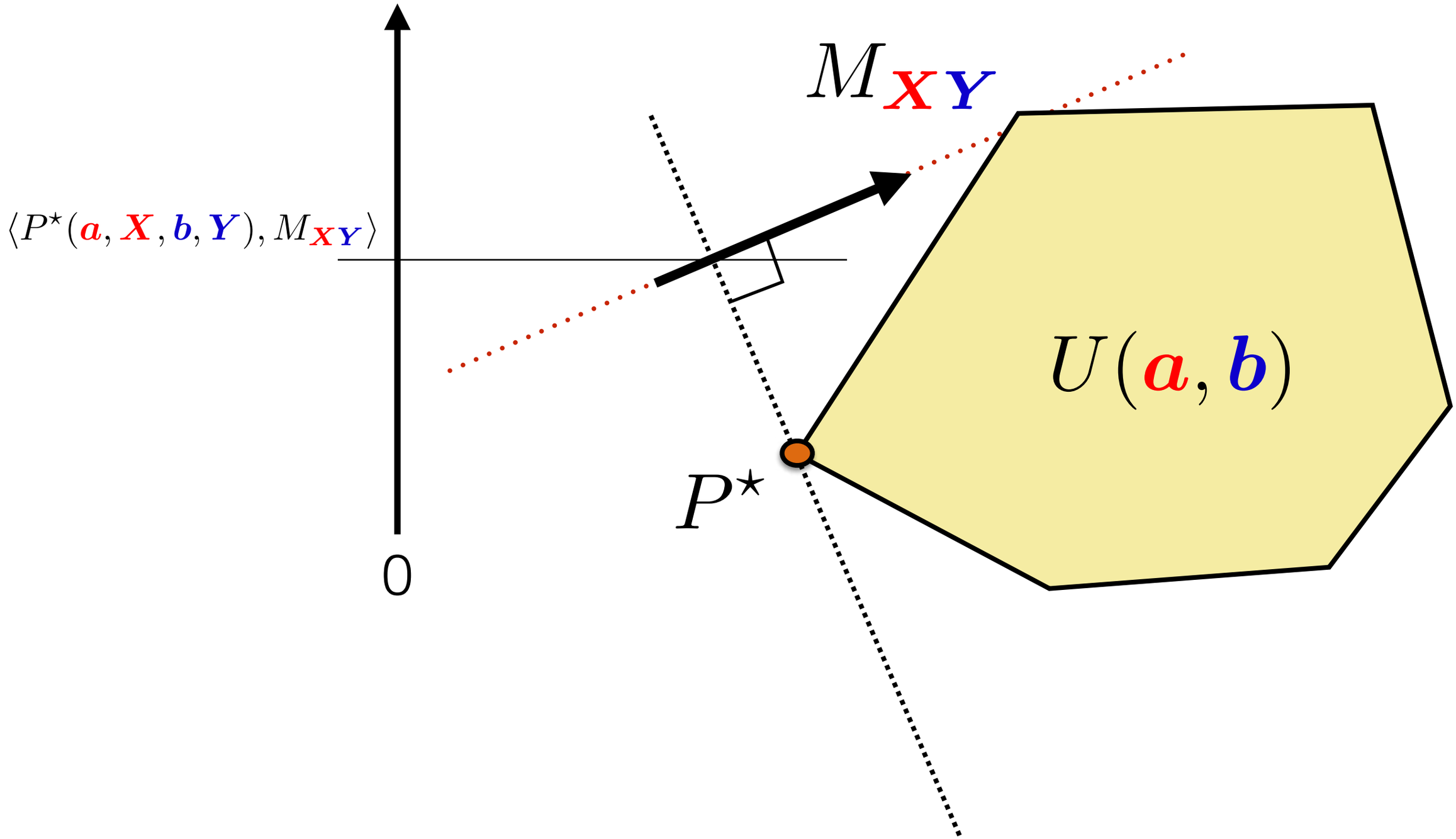
Prop. $p = 2, \Omega = \mathbb{R}^d$. $W(\boldsymbol{\mu}, \boldsymbol{\nu})$ decreases if

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

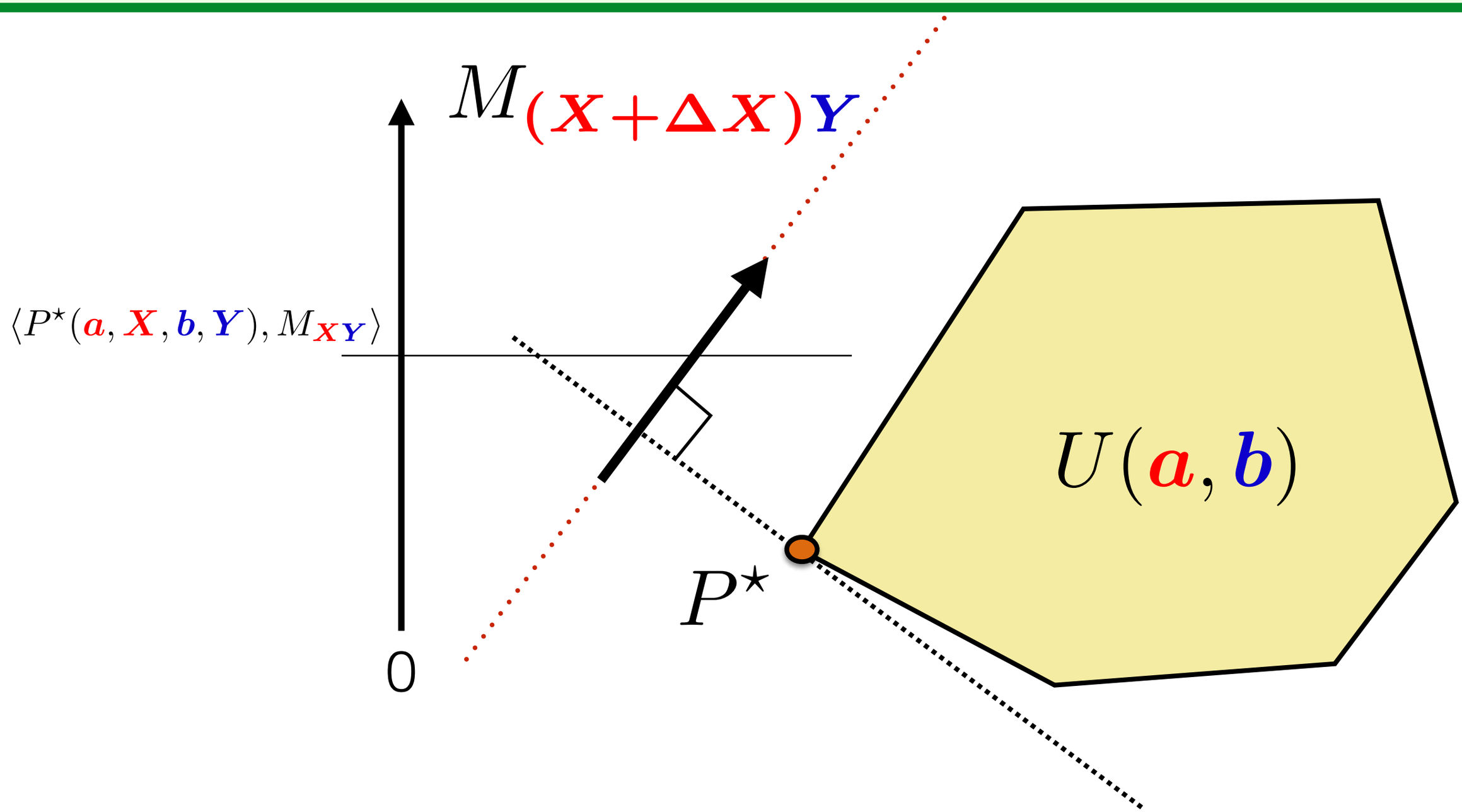
Prop. $p = 2, \Omega = \mathbb{R}^d$. $W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu})$ is **differentiable** w.r.t. $\boldsymbol{X}$, with

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

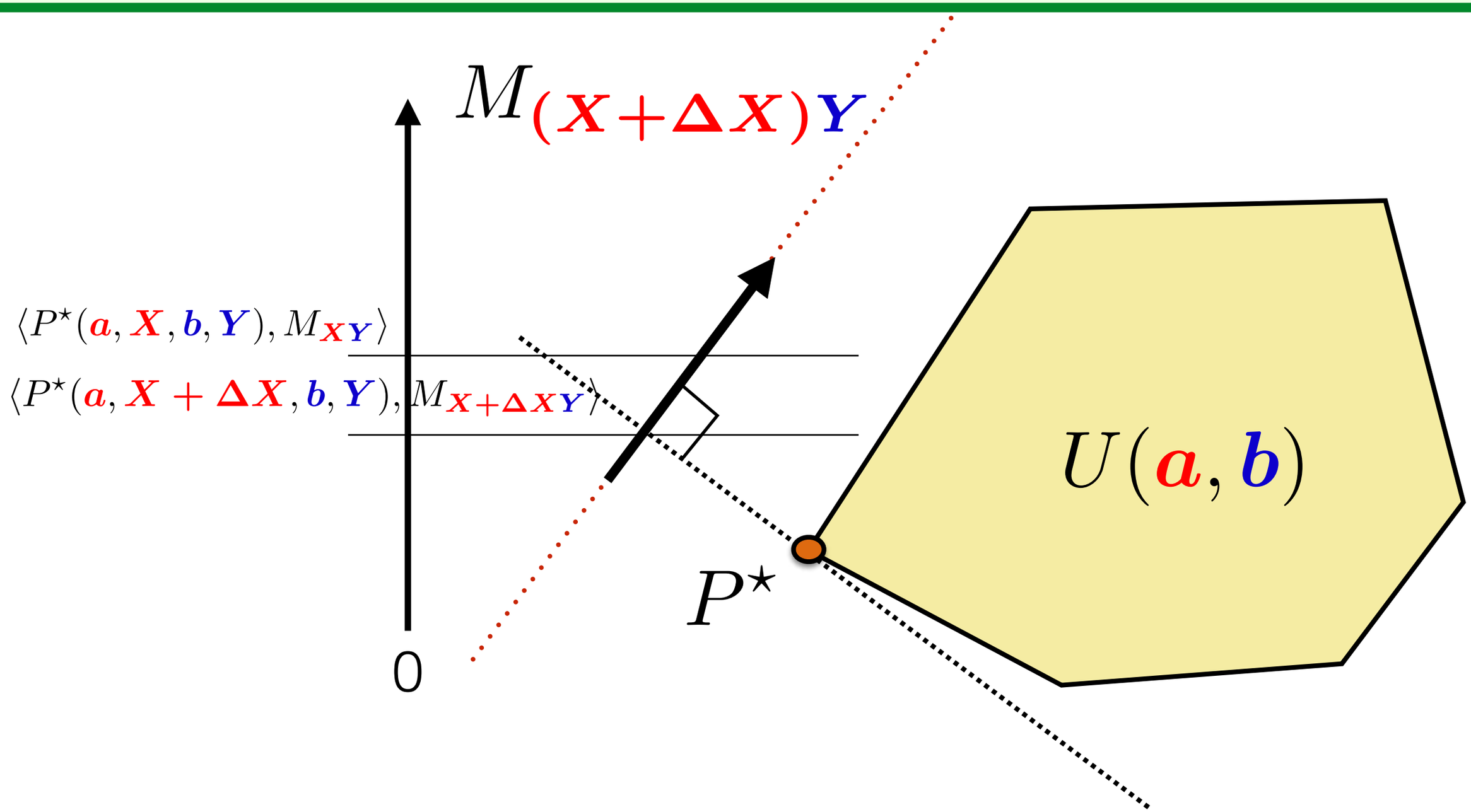
Solving the OT Problem



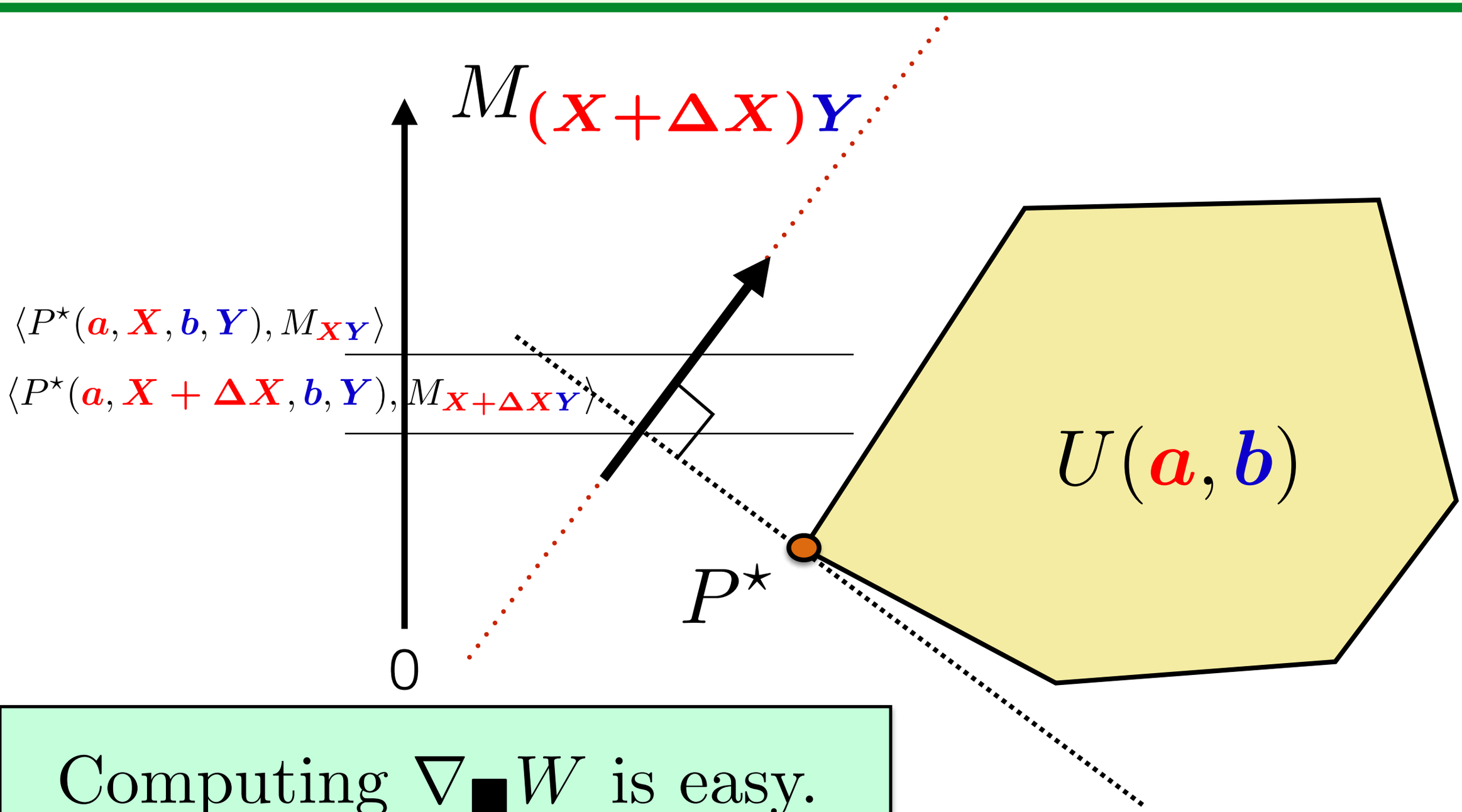
Solving the OT Problem



Solving the OT Problem



Solving the OT Problem

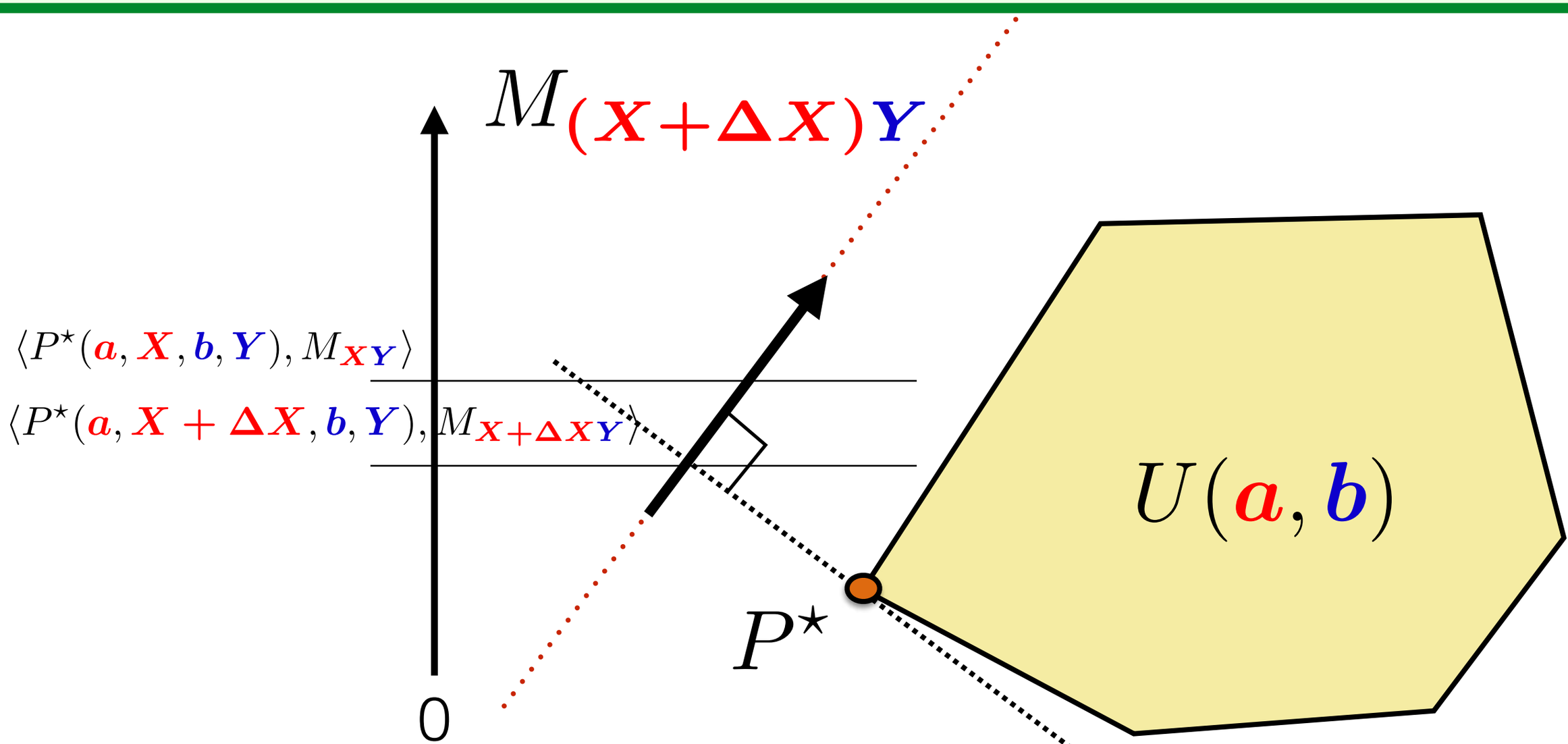


Computing $\nabla_{\blacksquare} W$ is easy.

here $\blacksquare$ can be anything,

e.g. a or X, parameter θ for cost c_θ

Solving the OT Problem



Computing $\nabla_{\blacksquare} W$ is easy.

here $\blacksquare$ can be anything,

e.g. $\boldsymbol{a}$ or $\boldsymbol{X}$, parameter $\boldsymbol{\theta}$ for cost $\boldsymbol{c_\theta}$

Computing $\dfrac{\partial P^\star}{\partial \blacksquare}$ is harder.

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

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

where

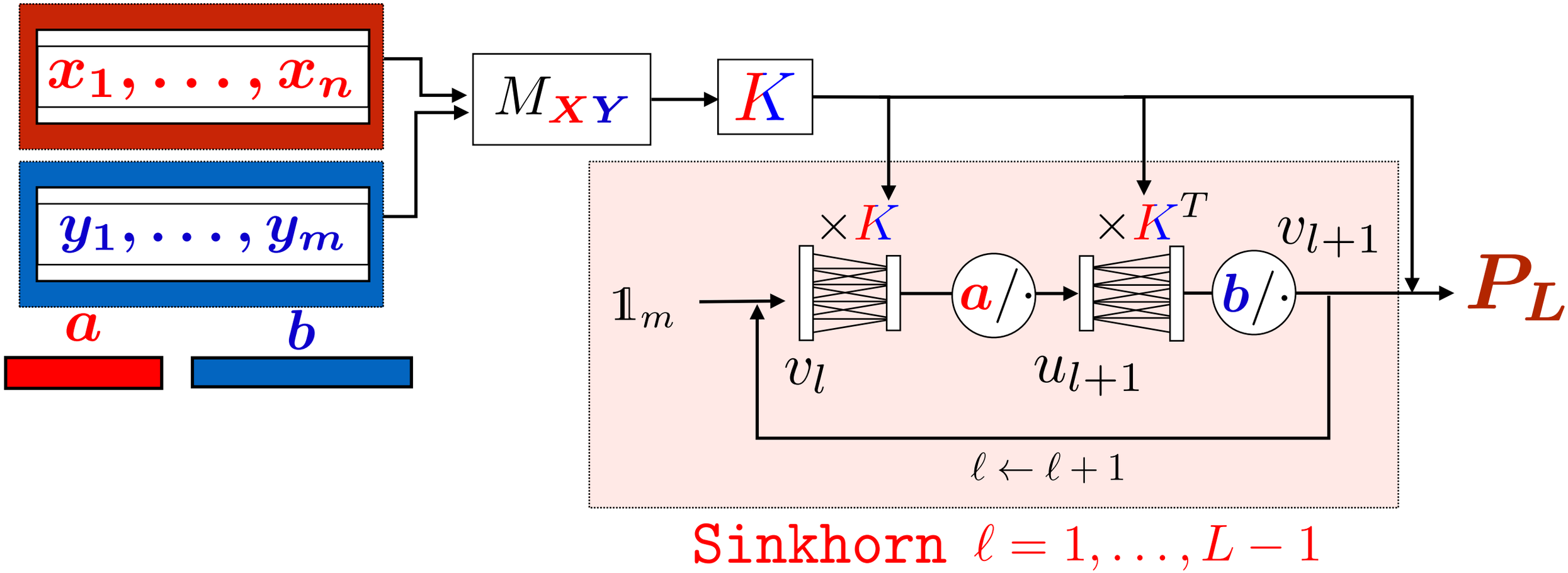
$$\boldsymbol{v_0} = \mathbf{1}_m; l \geq 0, \boldsymbol{u_l} \stackrel{\text{def}}{=} \boldsymbol{a}/K\boldsymbol{v_l}, \boldsymbol{v_{l+1}} \stackrel{\text{def}}{=} \boldsymbol{b}/K^T\boldsymbol{u_l}.$$

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

Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

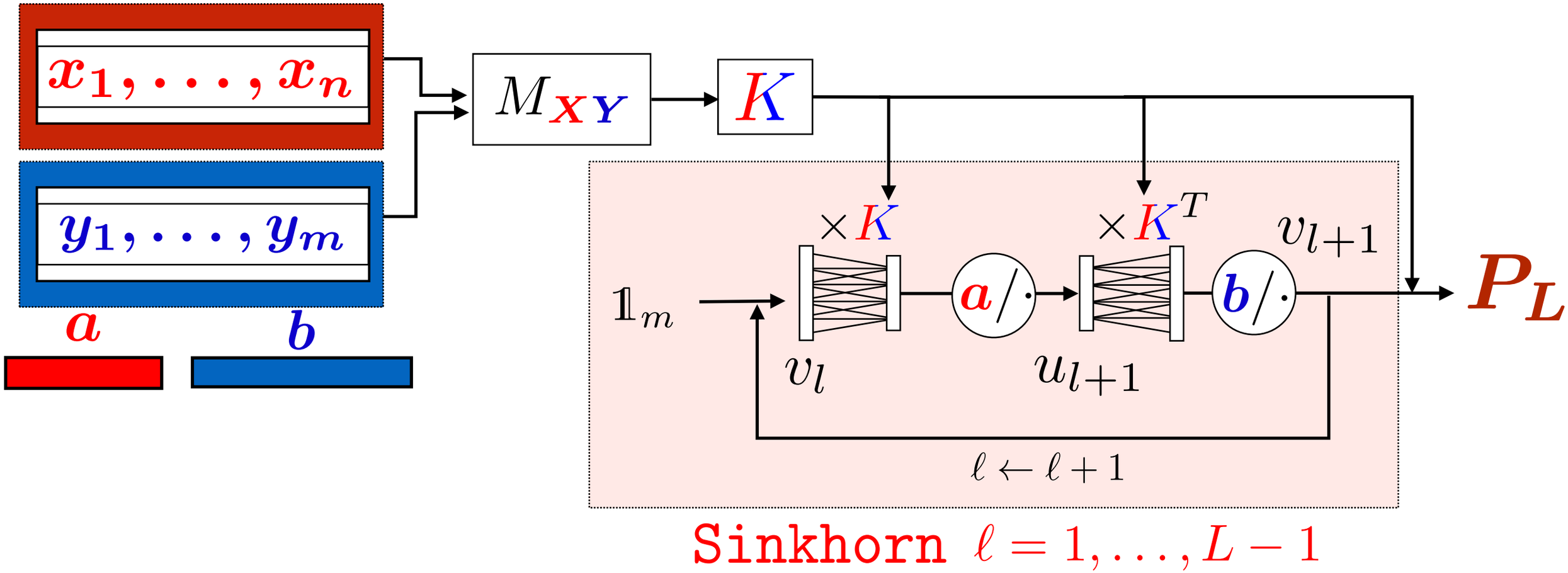
$$W_L(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \langle \boldsymbol{P_L}, M_{\boldsymbol{XY}} \rangle,$$



Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

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



[Adams'11] [Hashimoto'16] [Bonneel'16][Shalit'16]

Sinkhorn: A Mathematician View

$$F : \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon \rightarrow (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$$

$$H(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon; \boldsymbol{\alpha}, \boldsymbol{\beta}) := \begin{bmatrix} e^{\frac{\boldsymbol{\alpha} \oplus \boldsymbol{\beta} - \boldsymbol{C}}{\varepsilon}} 1_m - \boldsymbol{a} \\ e^{\frac{\boldsymbol{\beta} \oplus \boldsymbol{\alpha} - \boldsymbol{C}^T}{\varepsilon}} 1_n - \boldsymbol{b} \end{bmatrix} = 0$$

At the optimum,

$$H(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon; F(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon)) = \mathbf{0}$$

Sinkhorn: A Mathematician View

$$F : \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon \rightarrow (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$$

$$H(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{c}, \varepsilon; \boldsymbol{\alpha}, \boldsymbol{\beta}) := \begin{bmatrix} e^{\frac{\boldsymbol{\alpha} \oplus \boldsymbol{\beta} - \boldsymbol{C}}{\varepsilon}} \mathbf{1}_m = \boldsymbol{a} \\ e^{\frac{\boldsymbol{\beta} \oplus \boldsymbol{\alpha} - \boldsymbol{C}^T}{\varepsilon}} \mathbf{1}_n = \boldsymbol{b} \end{bmatrix} = 0$$

Using the implicit function theorem

$$J_{F,\blacksquare} = -\left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare, (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\right)^{-1} J_{H,\blacksquare}(\blacksquare, (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$$

$$J^T_{F,\blacksquare} = -J_{H,\blacksquare}(\blacksquare, (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)^T \left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare, (\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*))\right)^{-T}$$

The best of both worlds

When computing the gradient of a loss whose evaluation depends on the optimal transport solution, backward mode differentiation is more efficient. This involves evaluating:

$$J_{F,\blacksquare}^{T}\,\mathbf{z}$$

$$-J_{H,\blacksquare}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)^T\left(\left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\right)^T\right)^{-1}\mathbf{z}$$

The best of both worlds

When computing the gradient of a loss whose evaluation depends on the optimal transport solution, backward mode differentiation is more efficient. This involves evaluating:

$$J_{F,\blacksquare}^{T}\,\mathbf{z}$$

$$-J_{H,\blacksquare}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)^T\left(\underbrace{\left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\right)^T}_{\texttt{jax.vjp}}\right)^{-1}\mathbf{z}$$

The best of both worlds

When computing the gradient of a loss whose evaluation depends on the optimal transport solution, backward mode differentiation is more efficient. This involves evaluating:

$$J_{F,\blacksquare}^{T}\,\mathbf{z}$$

$$-J_{H,\blacksquare}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)^T\left(\left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\right)^T\right)^{-1}\mathbf{z}$$

`jax.vjp`

`jax.linalg.sparse.cg`

The best of both worlds

When computing the gradient of a loss whose evaluation depends on the optimal transport solution, backward mode differentiation is more efficient. This involves evaluating:

$$J_{F,\blacksquare}^{T}\,\mathbf{z}$$

$$-J_{H,\blacksquare}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)^T\left(\left(J_{H,(\boldsymbol{\alpha},\boldsymbol{\beta})}(\blacksquare,(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\right)^T\right)^{-1}\mathbf{z}$$

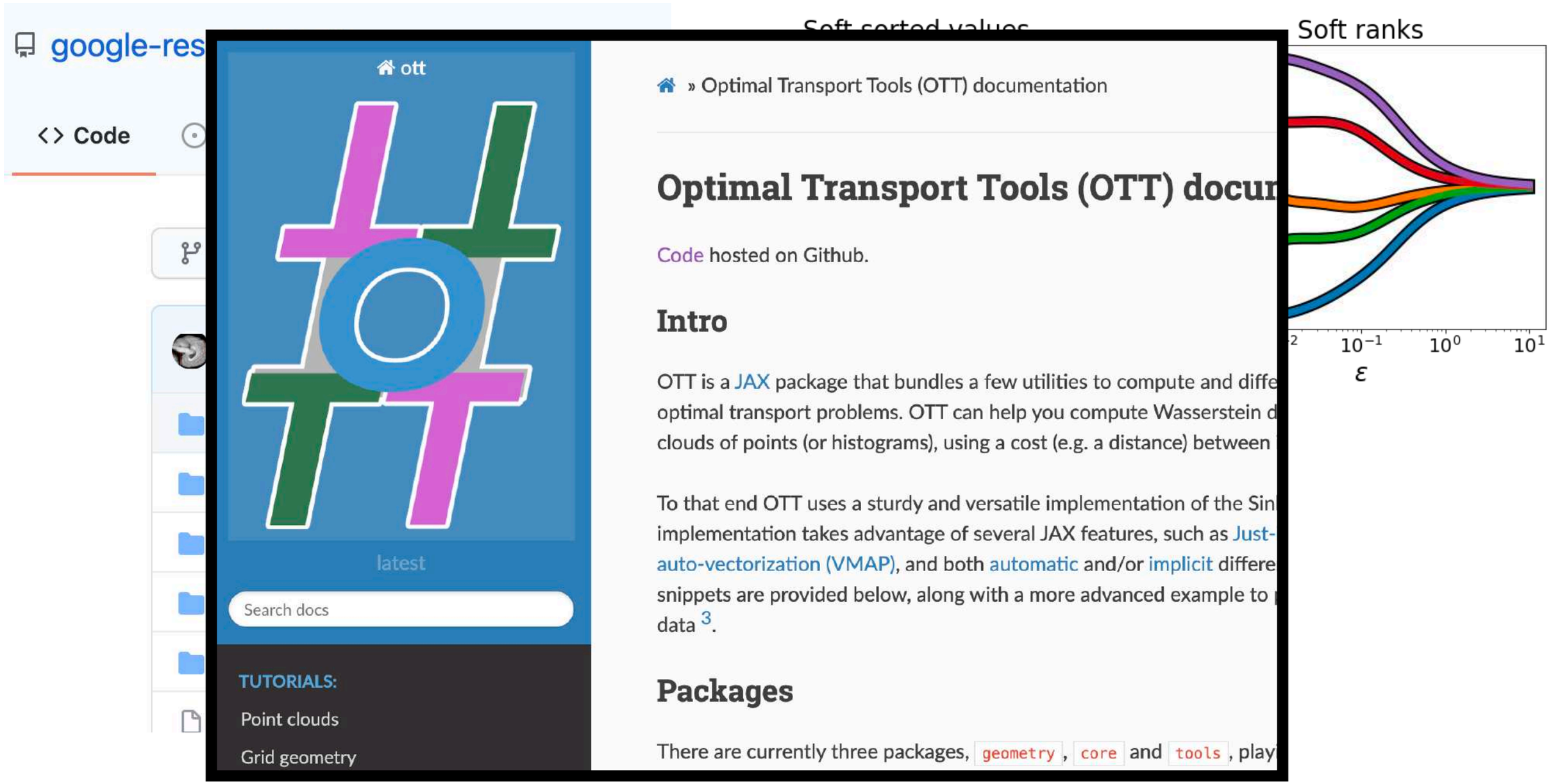
`jax.vjp`

`jax.vjp`

`jax.linalg.sparse.cg`

For more details

`https://github.com/google-research/ott`

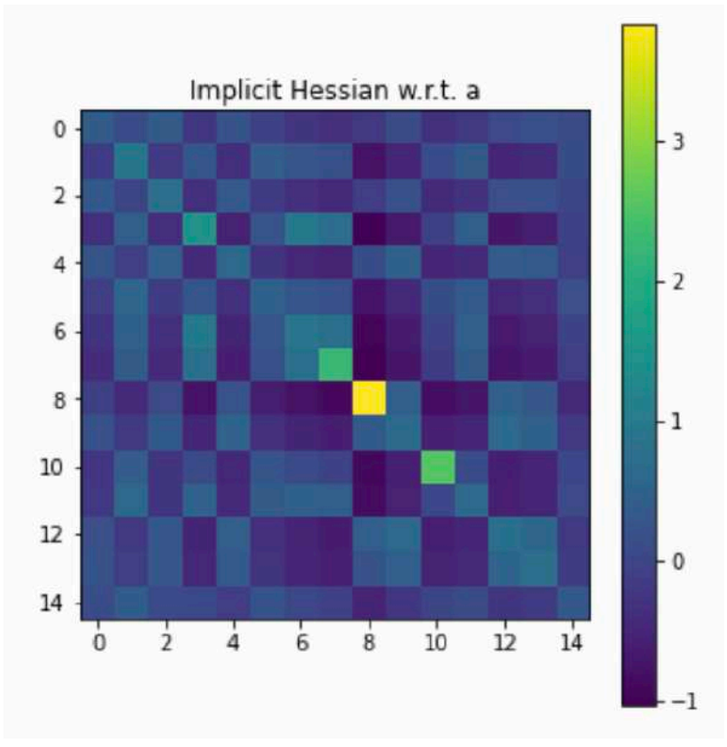


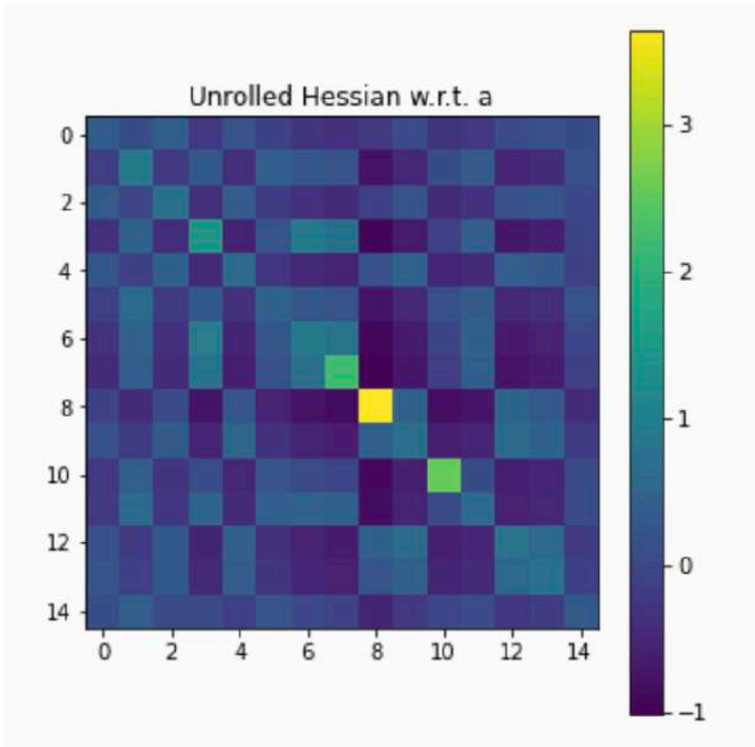
For instance...computing Hessians.

https://ott-jax.readthedocs.io/en/stable/notebooks/Hessians.html

[9]:
```
def loss(a, x, implicit):
  return sinkhorn_divergence.sinkhorn_divergence(
      pointcloud.PointCloud, x, y,  # this part defines geometry
      a=a, b=b,  # this sets weights
      sinkhorn_kwargs={'implicit_differentiation': implicit, 'use_danskin': False}  # to be
      ).divergence
```

[17]:
```
for arg in [0,1]:
  # Compute Hessians using either unrolling or implicit differentiation.
  hess_loss_imp = jax.jit(jax.hessian(lambda a, x: loss(a, x, True),
                                      argnums=arg))
```





To conclude

To sum up

- **Optimal matchings** used everywhere in ML, to
  - Compute a distance (W) between measures.
  - register/map/match points, to disambiguate.
- **Regularization** is needed to:
  - Scale up/robustify these tools;
  - Ensure gradients/jacobians exist and are not 0.
- **To differentiate** through regularised OT, either
  - Automatic differentiation (unrolling)
  - Implicit differentiation

A few research topics around OT

- **Sinkhorn fixed-point iterations** speed
  - Using convolutions **[Solomon+15]**
  - Faster kernel multiplications *(many refs!)*
  - Accelerations (Anderson **[Chizat+20]**)
  - Improved convergence proofs

- Generalize Sinkhorn to **continuous spaces**? (Schrödinger bridges) **[Chen+14]** **[Bortoli+21]**
- **Other** regularizers (*L2*) / constraints (low-rank)

A few research topics around OT

- **Unbalanced OT formulations**
  - Penalized **[FZMAP15][Chizat+15]**, approaches can handle (with Sinkhorn) different marginals
- **Brenier / Monge topics**
  - Estimate Monge maps using ICNN **[VM+19]**
  - Links with Normalizing Flows **[Huang+20]**
- **Proximal optimization** in spaces of measures.
  - **[JKO'98]** starting to make an impact in practice, see for instance **[Bunne+21]**

A few research topics around OT

- **OT for Heterogeneous spaces**
  - Use quadratic assignment problem to compute machines (GW, **[Memoli'11]**)
  - Computational challenges, stats unknown.

- **Exploit matching differentiability**
  - soft-sorting **[CTV19]** and -quantiles **[CTNV'20]**
  - to impute missing values **[Muzellec+20]**