Optimal Transport for Machine Learning: Distances, Algorithms, and Domain Adaptation

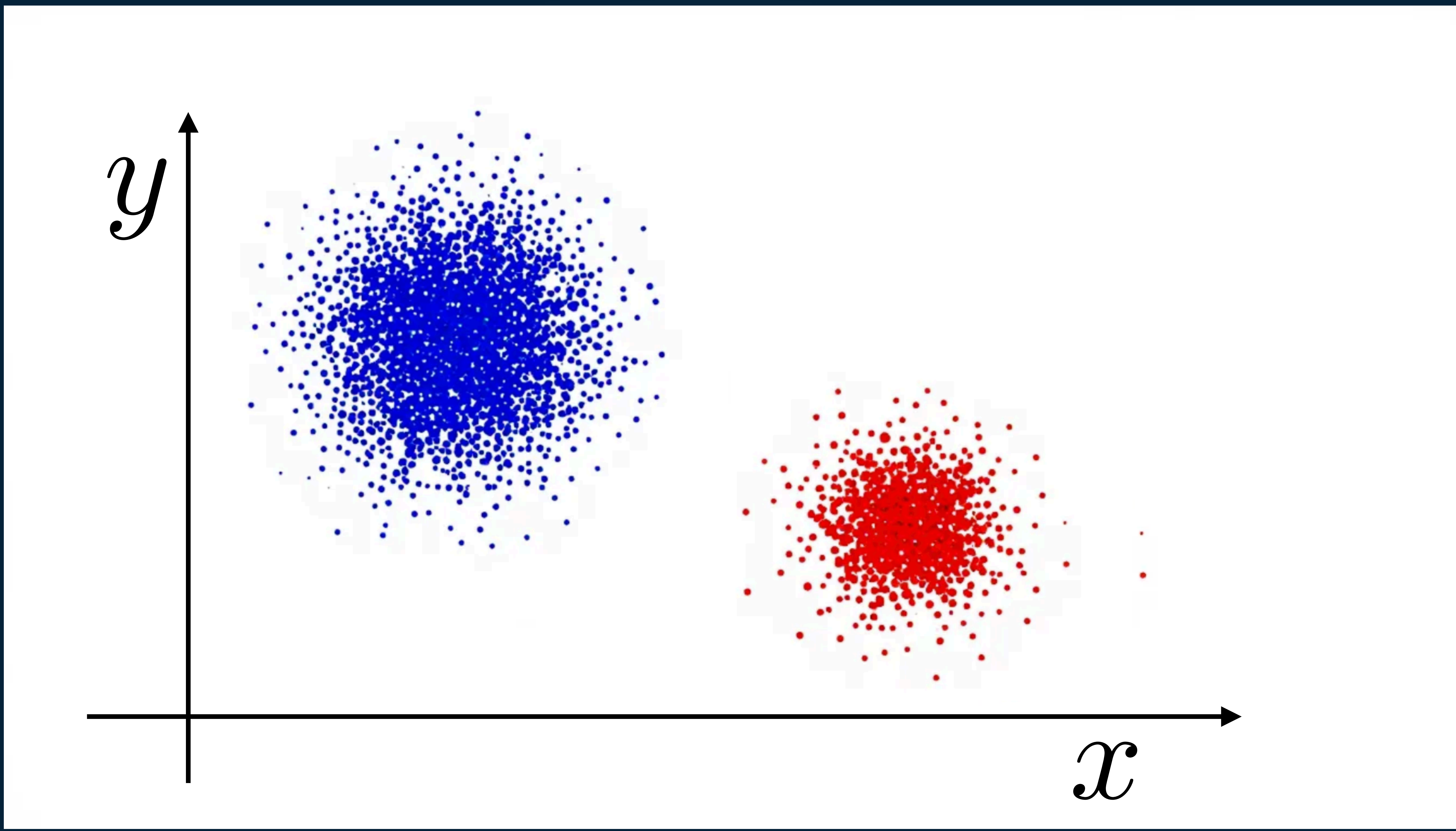
Mokhtar Z. Alaya

Séminaire LMAC, 2 février 2026

1. Motivations

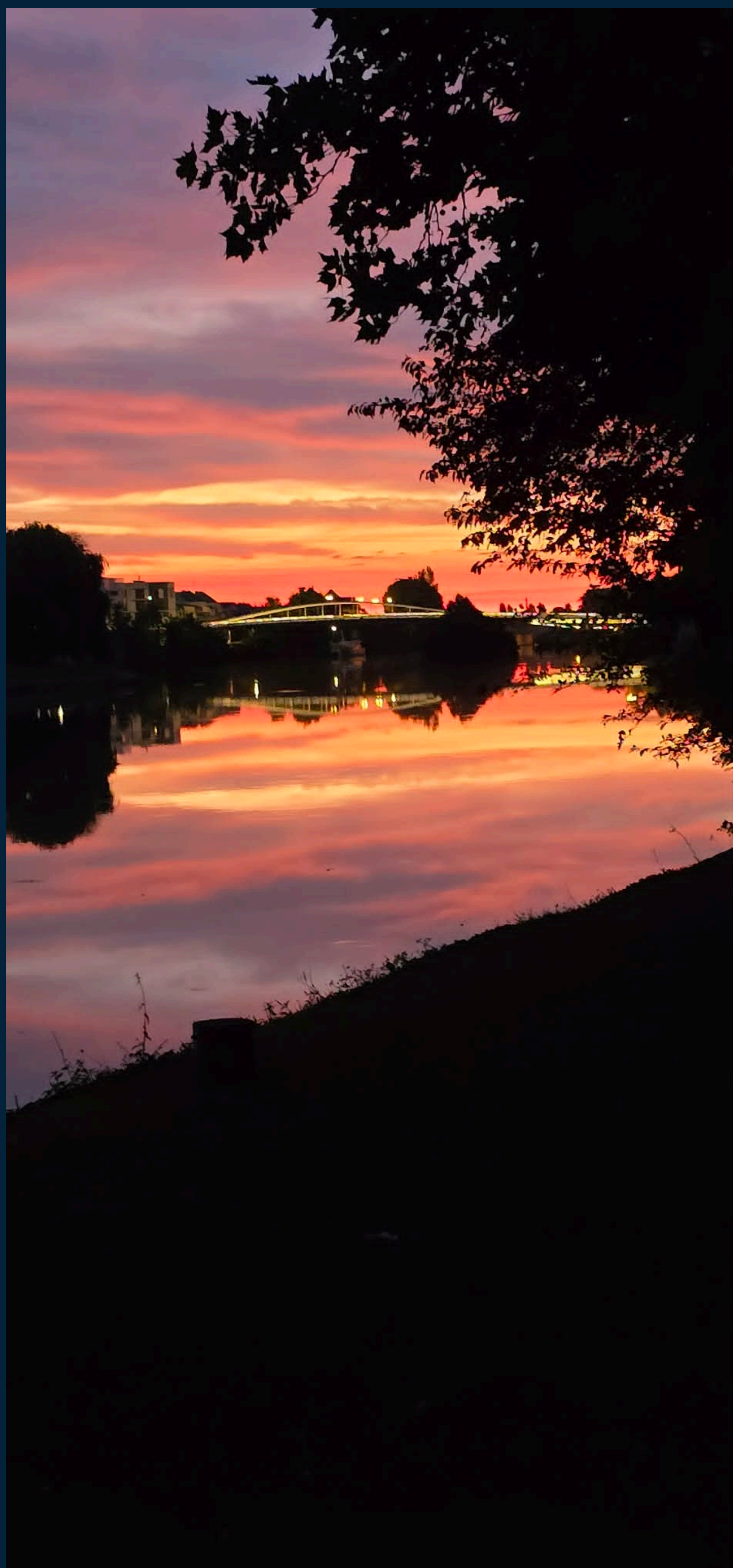
… in Cloud of points

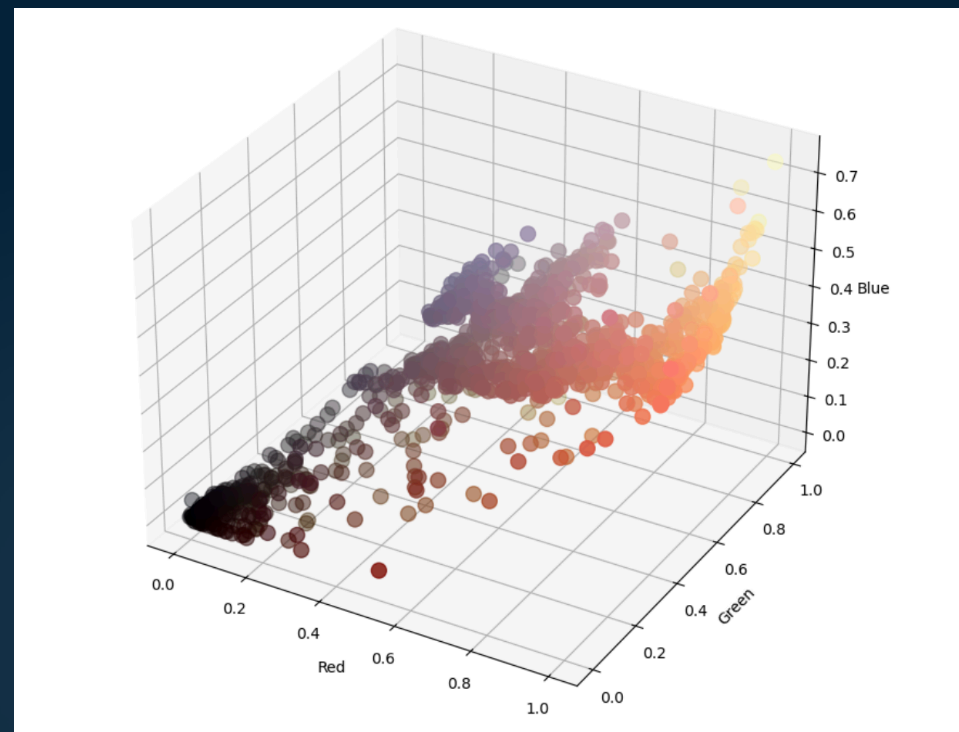
- How to compare two sets of clouds points?

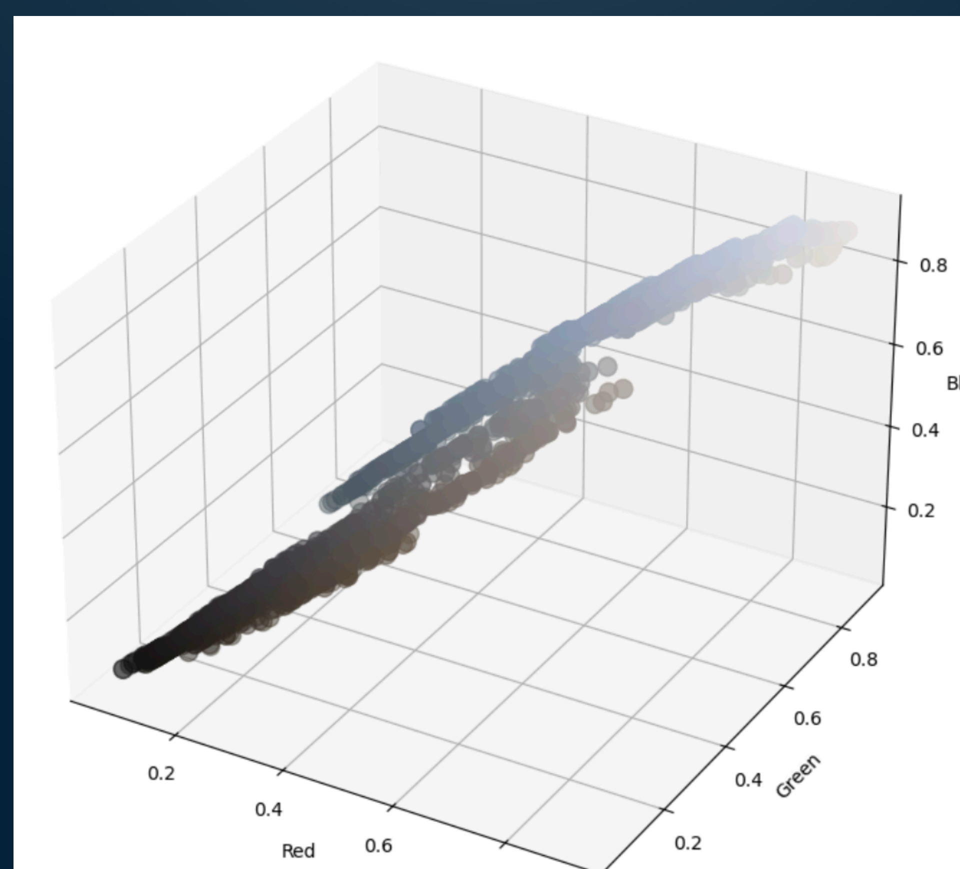


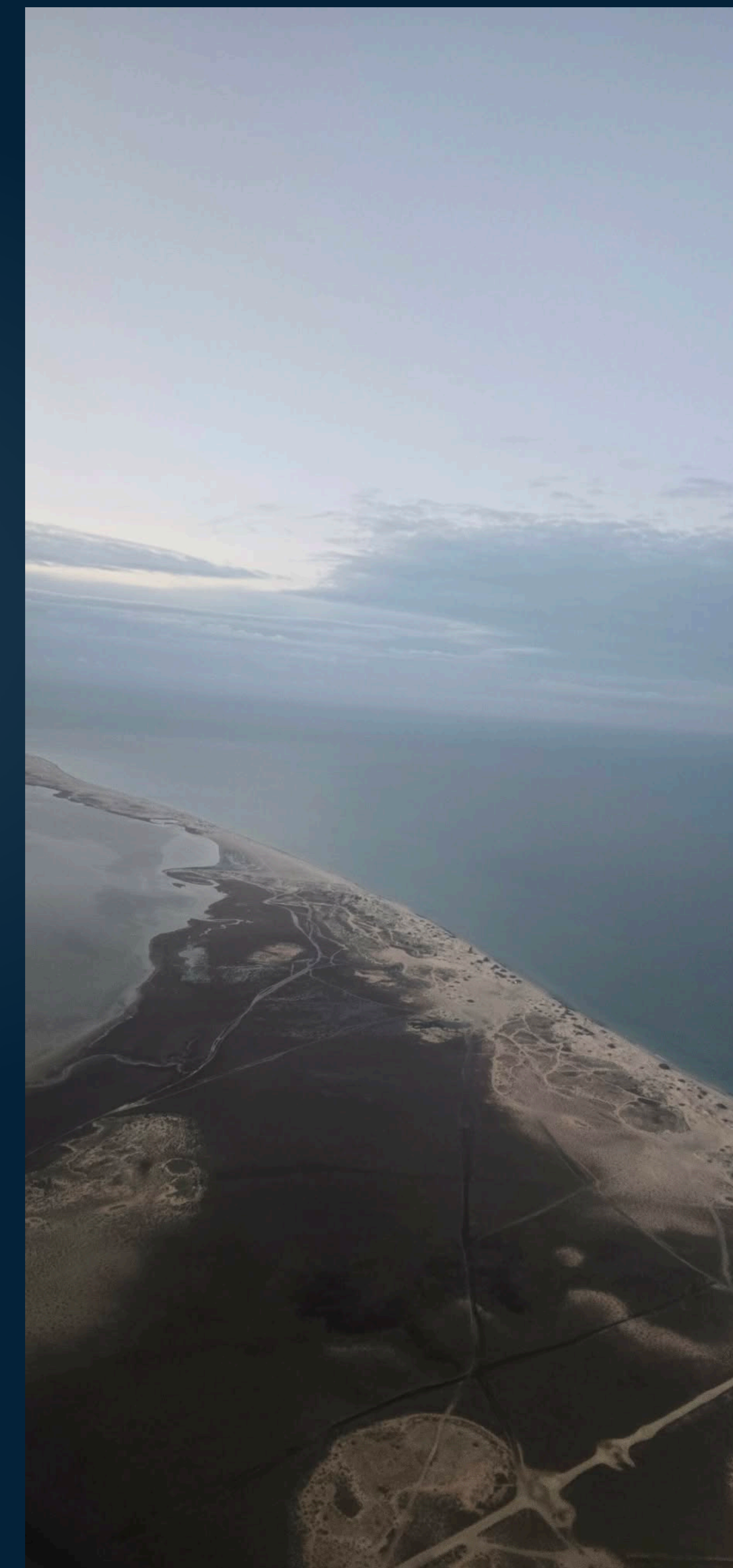
… in Image Processing

- How to measure the similarity between two images?



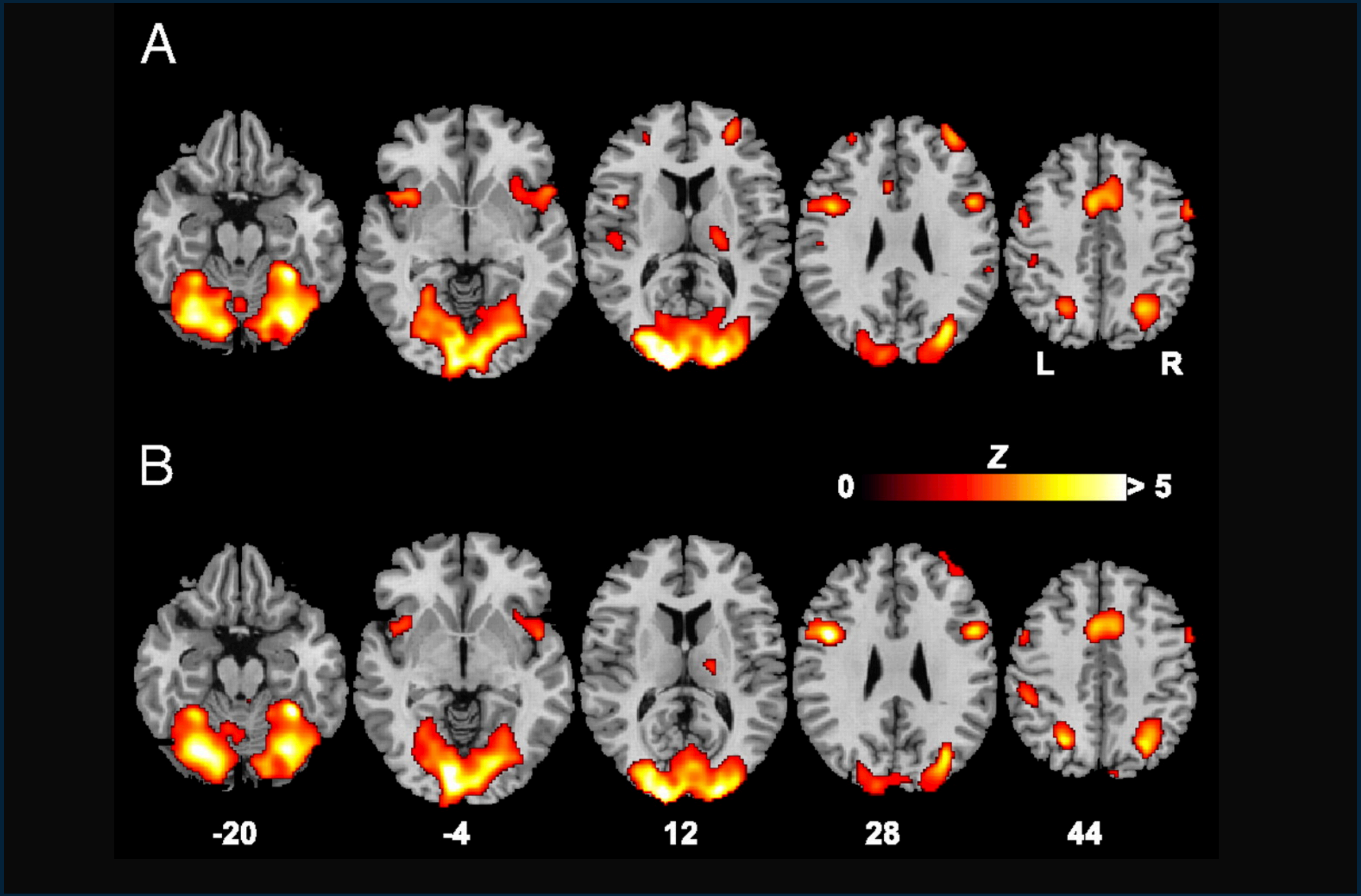






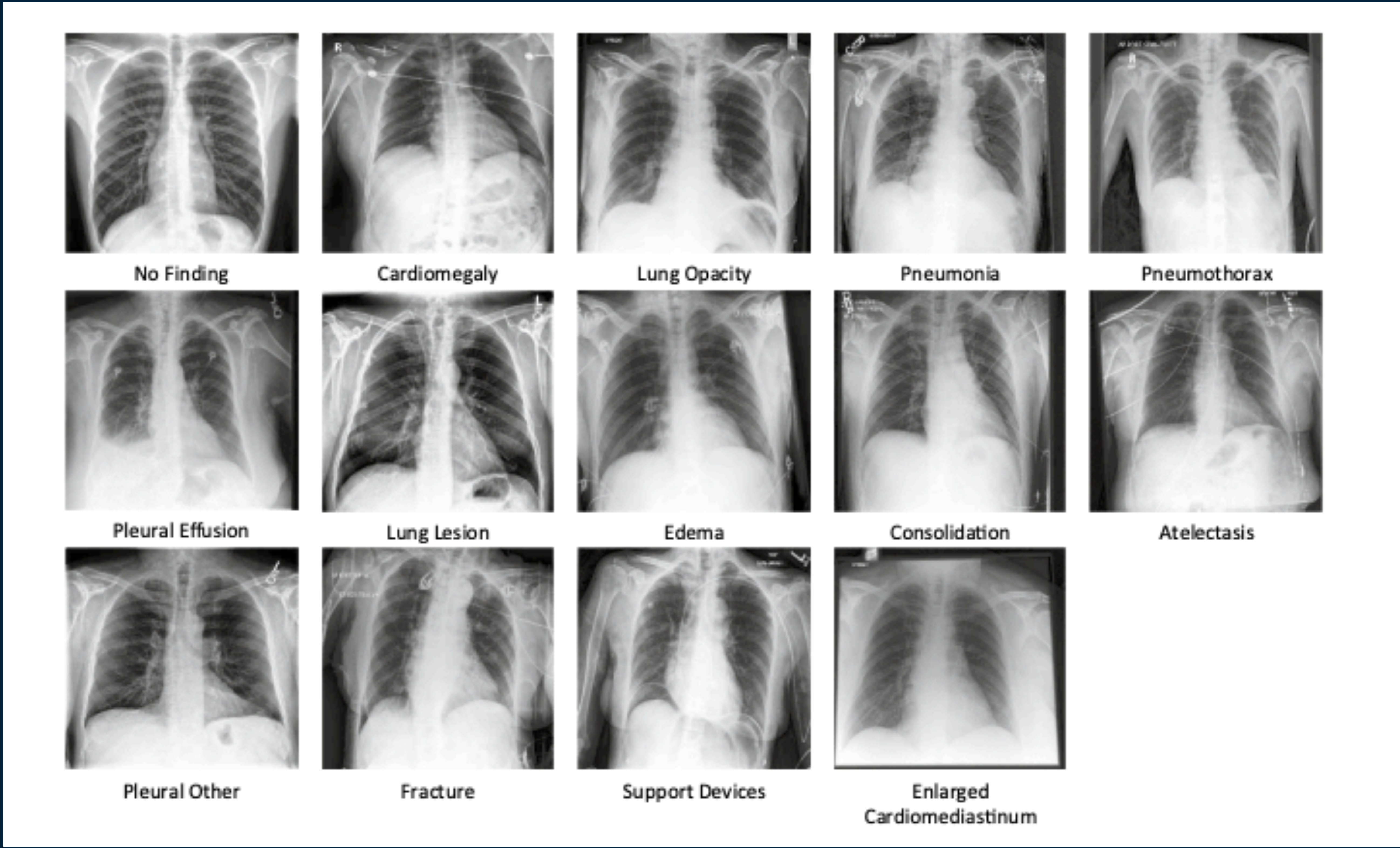
… in Neuroscience

- How to compare two brain activation maps?



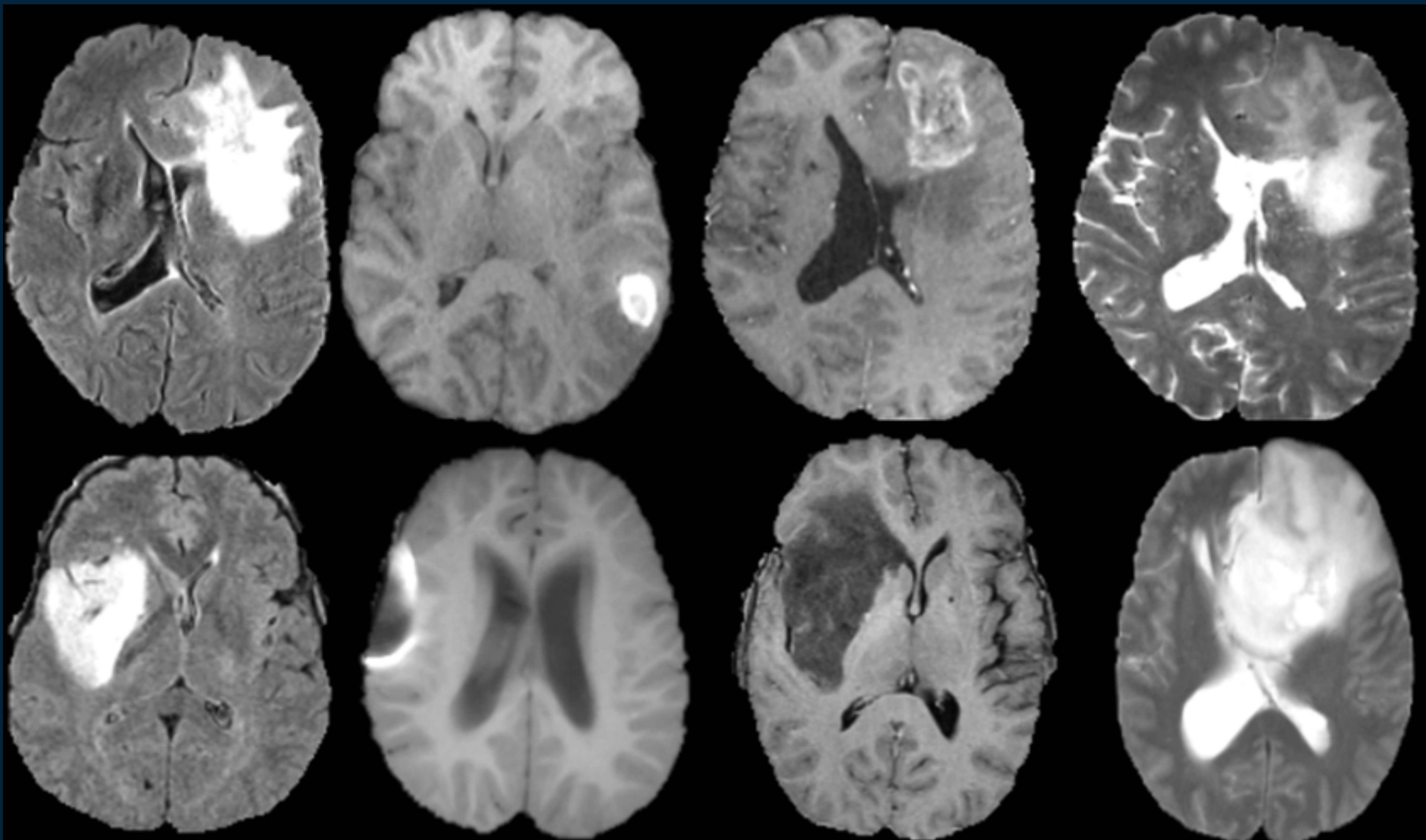
Source: Hu et al., 2008, Proceedings of the National Academy of Sciences

… in Classification: CheXpert data



Source: https://stanfordmlgroup.github.io/competitions/chexpert/

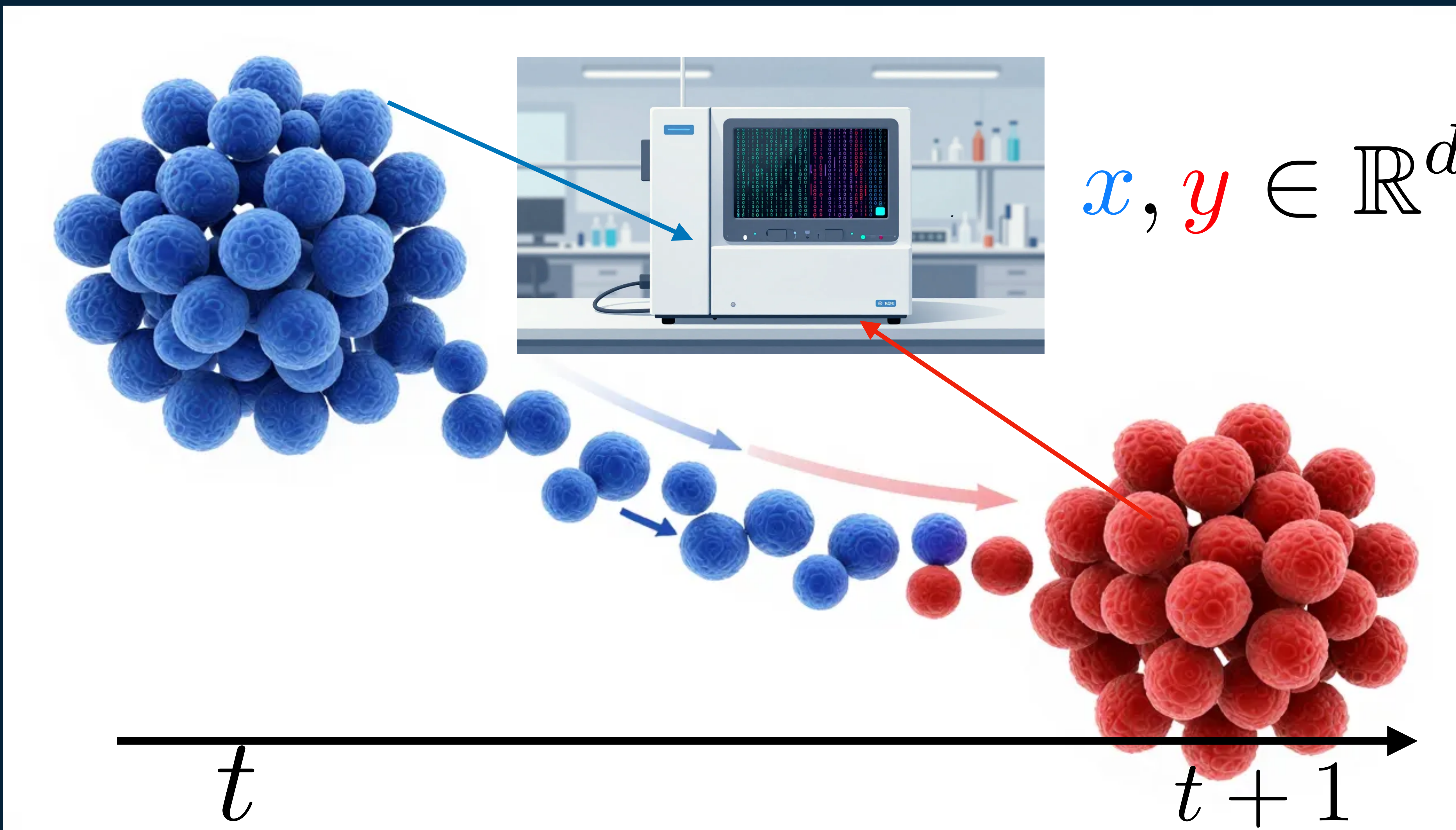
… in Anomaly detection: BraTS data



Source: https://www.med.upenn.edu/cbica/brats2020/registration.html

... in Genomics

- Understanding dynamics at individual cell level.



2. Machine Learning: Predictive Modeling

Training Set

$\mathcal{X}$ Feature Space

$\mathcal{Y}$ Label Space

$$(\vec{\boldsymbol{x}}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$$

$$\vec{\boldsymbol{x}}_i = (x_{i1}, \ldots, x_{id})^\top \in \mathbb{R}^d$$

Sampling Data

$$\mathscr{D}_n = \{(\vec{\boldsymbol{x}}_1, y_1), (\vec{\boldsymbol{x}}_2, y_2), \ldots, (\vec{\boldsymbol{x}}_n, y_n)\}$$

Learning Task

- The learning task consists of assuming that the labels have been calculated using a function.

$$h^{\star}: \mathcal{X} \to \mathcal{Y}$$

- Find a hypothesis that better approximates the target function. Build a predictor model, classifier, or regression function.

$$\hat{h} \equiv h_{\hat{\boldsymbol{\theta}}}: \mathcal{X} \to \mathcal{Y}$$

Learning Task: loss function

- A cost function, also known as a loss function, is a function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$$

used to quantify prediction quality

$$\ell(y, h(\vec{x}))$$

Real Risk

- Risk is defined as the expectation of a cost function (or expectation of prediction error), i.e.,

$$\mathcal{R}(h) = \mathbb{E}_{(\vec{\boldsymbol{x}},y)\sim\mathbb{P}}[\ell(h(\vec{\boldsymbol{x}}), y]$$

- Risk minimization is impossible in practice because the joint law $\mathbb{P}$ of observations is unknown.
- Idea: replace the theoretical probability measure $\mathbb{P}$ with the empirical probability measure $\mathbb{P}_n$:

$$d\mathbb{P}_n(x, y) = \frac{1}{n}\sum_{i=1}^{n}\delta_{(\vec{\boldsymbol{x}}_i, y_i)}(\vec{\boldsymbol{x}}, y)$$

Empirical Risk Minimisation

- Empirical risk assesses the cumulative effect of errors

$$\mathcal{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(\vec{\boldsymbol{x}}_i))$$

- Predictor by empirical risk minimization:

$$h_{\hat{\boldsymbol{\theta}}} = \arg\min_{h \in \mathcal{H}} \mathcal{R}_n(h)$$

3. Optimal Transport

Origin: Monge Problem (1781)

MÉMOIRE

SUR LA

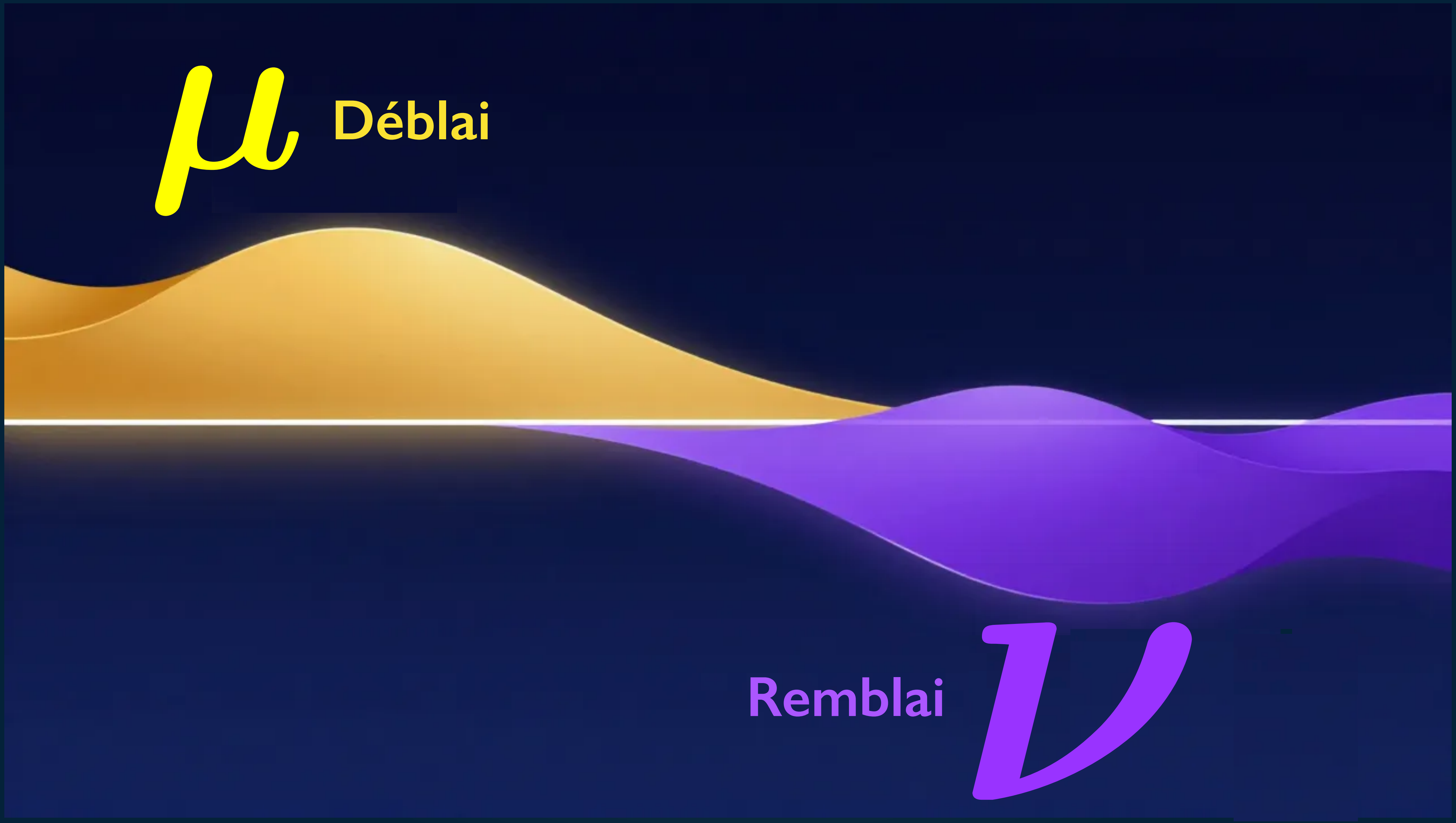
THÉORIE DES DÉBLAIS ET DES REMBLAIS.

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.

Le prix du tranſport d'une molécule étant, toutes choſes d'ailleurs égales, proportionnel à ſon poids & à l'eſpace qu'on lui fait parcourir, & par conſéquent le prix du tranſport total devant être proportionnel à la ſomme des produits des molécules multipliées chacune par l'eſpace parcouru, il s'enſuit que le déblai & le remblai étant donnés de figure & de poſition, il n'eſt pas indifférent que telle molécule du déblai ſoit tranſportée dans tel ou tel autre endroit du remblai, mais qu'il y a une certaine diſtribution à faire des molécules du premier dans le ſecond, d'après laquelle la ſomme de ces produits ſera la moindre poſſible, & le prix du tranſport total ſera un *minimum*.

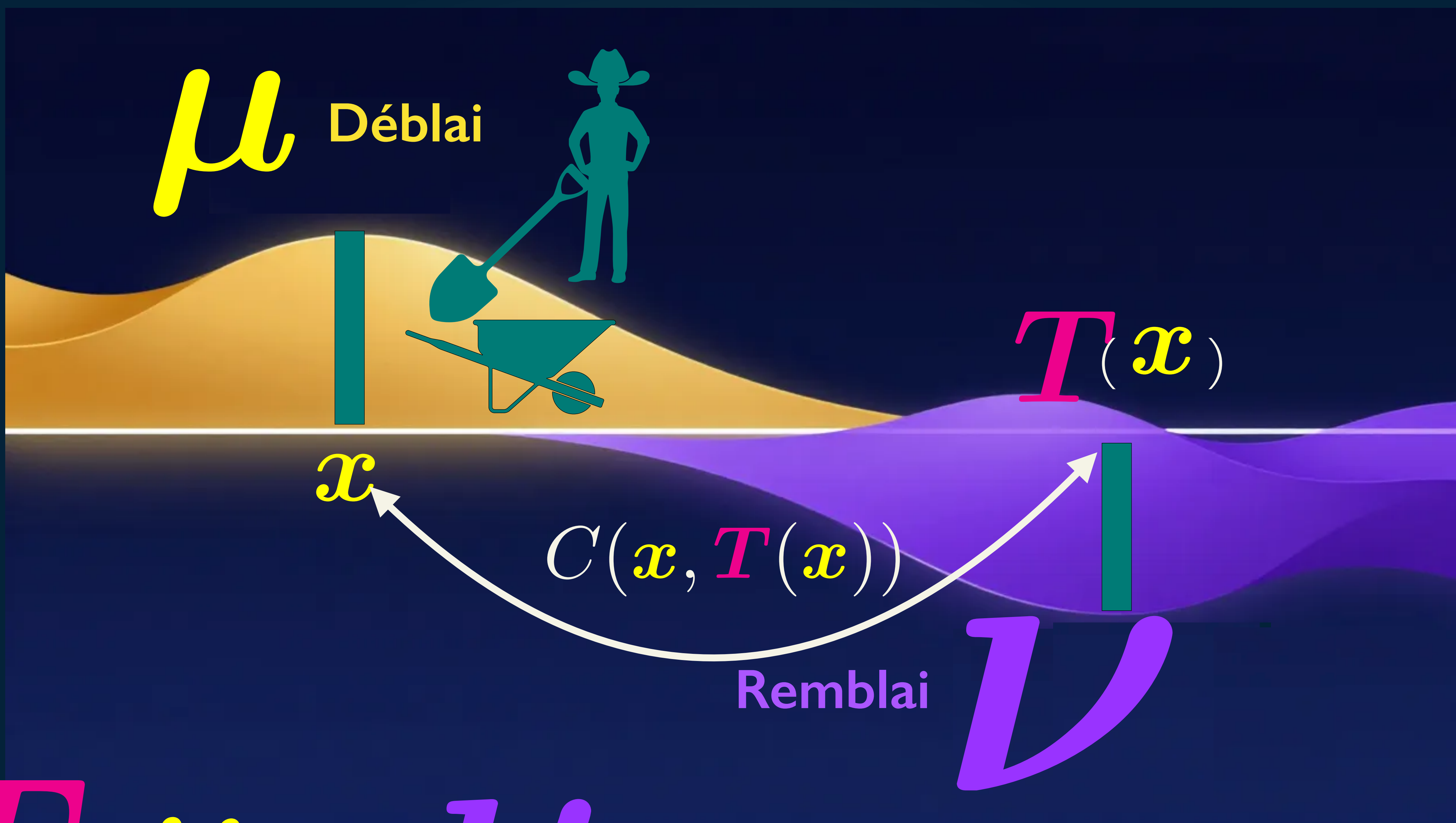
Monge Problem (1781)



- How to move dirt from one place (**déblai**) to another (**remblai**) while minimizing the effort?
- Find a mapping T between the two distributions of mass (**transport**).
- Optimize with respect to a displacement cost (**optimal**).

Monge Problem (1781)

- The mapping T must **push-forward** the **“déblai”** measure towards the **“remblai”**.



$$T_\# \mu = \nu$$

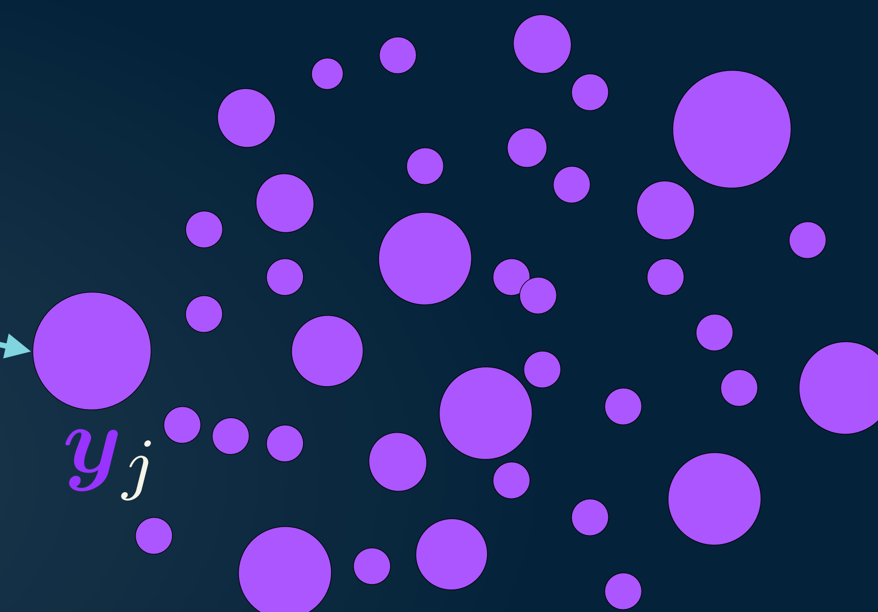
Monge Problem (1781)

- Monge formulation aim at finding a mapping $\boldsymbol{T}$ such that:

$$\inf_{\boldsymbol{T}_{\#}\mu=\nu} \int \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{T}(\boldsymbol{x}))\mu(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Mapping $\boldsymbol{T}$ does not exist in the general case.
- Brenier, 1991 proved existence and unicity of the Monge map for Euclidean cost and distributions with densities.

Discrete OT Framework



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

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

x_i, y_j

$$C_{ij} = (c(x_i, y_j))$$

C

Cost Matrix

n, m

$$C_{ij} = (c(x_i, y_j))$$

$$\min_{T_{\#}\mu = \nu} \sum_i C(x_i, T(x_i))\mu_i$$

$$\forall j \in \{1, \ldots, m\}, \nu_j = \sum_{i:T(x_i)=y_j} \mu_i$$

Discrete OT Framework: Monge's Formula

$$\min_{T_\# \mu = \nu} \sum_i C(\boldsymbol{x}_i, T(\boldsymbol{x}_i))\mu_i$$

$$\forall j \in \{1, \ldots, m\}, \nu_j = \sum_{i : T(\boldsymbol{x}_i) = \boldsymbol{y}_j} \mu_i$$

Strict: Deterministic Assignments

Discrete OT Framework: Monge's Formula

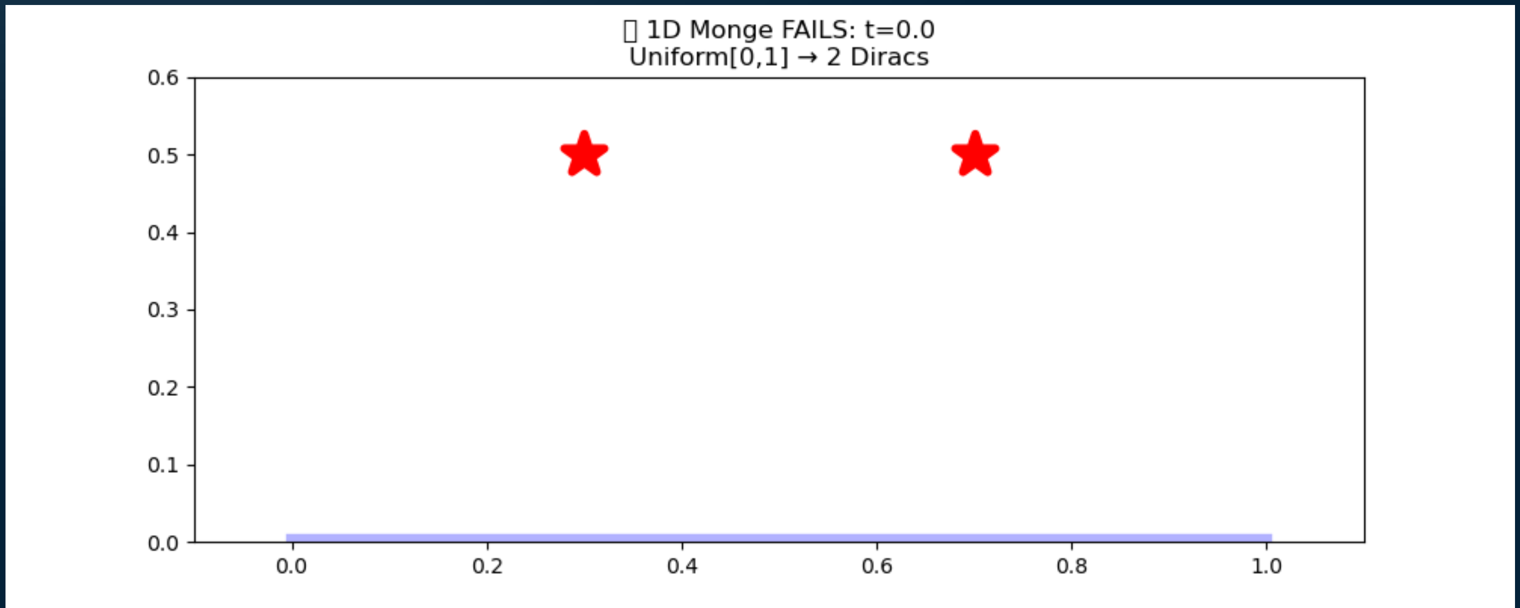
non-convex

combinatorial

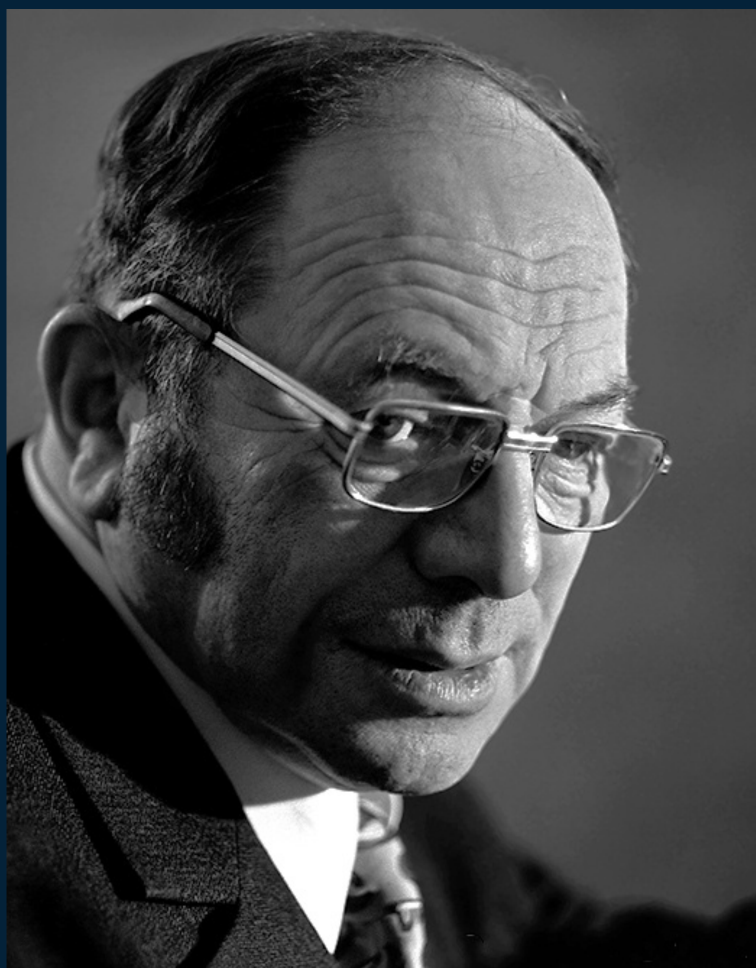
non-existant

Uniform weights

$$\min_{\sigma \in \mathfrak{S}_n} \sum_{i=1}^{n} C(\boldsymbol{x}_i, y_{\sigma(i)})$$

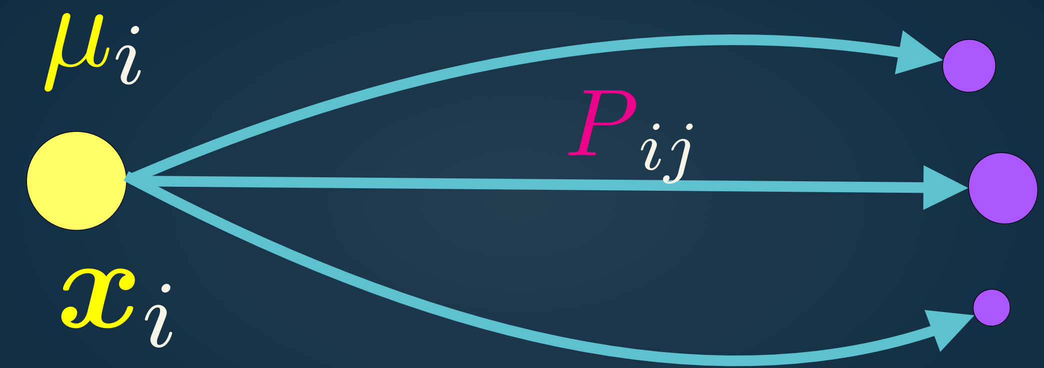


Discrete OT Framework: Kantorovich's Formula



Leonid Kantorovich
(1912-1986)

- Focus on where the mass goes, allow splitting.
- Applications mainly for resource allocation problems.



$$\mu_i = \sum_{i=1}^{m} P_{ij}$$

Relaxed: Fractional Assignments

Probabilistic couplings set (Transport Polytope)

$$\mathbf{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\boldsymbol{P} \in \mathbb{R}_+^{n \times m}, \boldsymbol{P}\mathbf{1}_m = \boldsymbol{\mu}, \boldsymbol{P}^\top \mathbf{1}_n = \boldsymbol{\nu}\}$$

Mass conservation constraints

Discrete OT Framework: Kantorovich's Formula

- Computing OT between μ and ν amounts to solving a linear problem:

Kantorovich 1942

$$S(\boldsymbol{\mu},\boldsymbol{\nu}) = \min_{\boldsymbol{P}\in\boldsymbol{\Pi}(\boldsymbol{\mu},\boldsymbol{\nu})} \left\{ \langle \boldsymbol{C}, \boldsymbol{P} \rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{C}_{ij}\boldsymbol{P}_{ij} \right\}$$

Distance

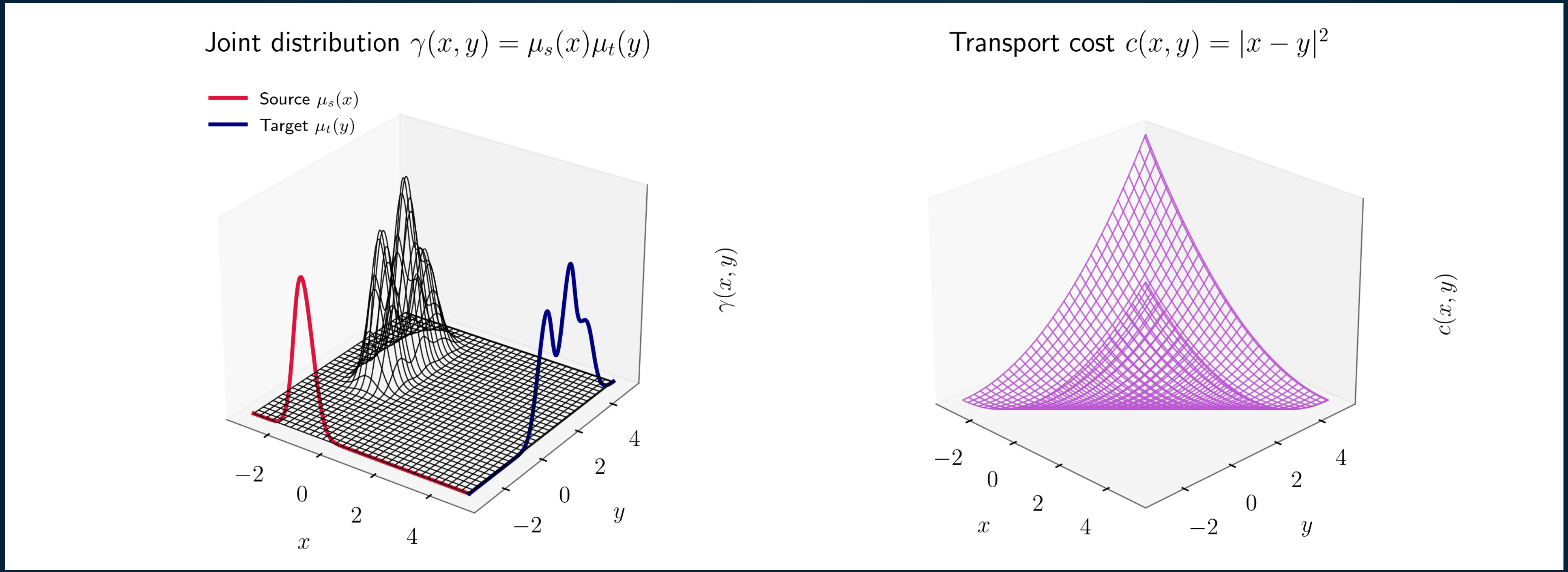
Monge-Kantorovich / Wasserstein Distance

Continuous OT Framework: Kantorovich's Formula

$$\min_{\gamma \in \Pi_{\text{con}}(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} C(x,y)\gamma(x,y)\mathrm{d}x\mathrm{d}y$$

Probabilistic couplings set (Transport Polytope)

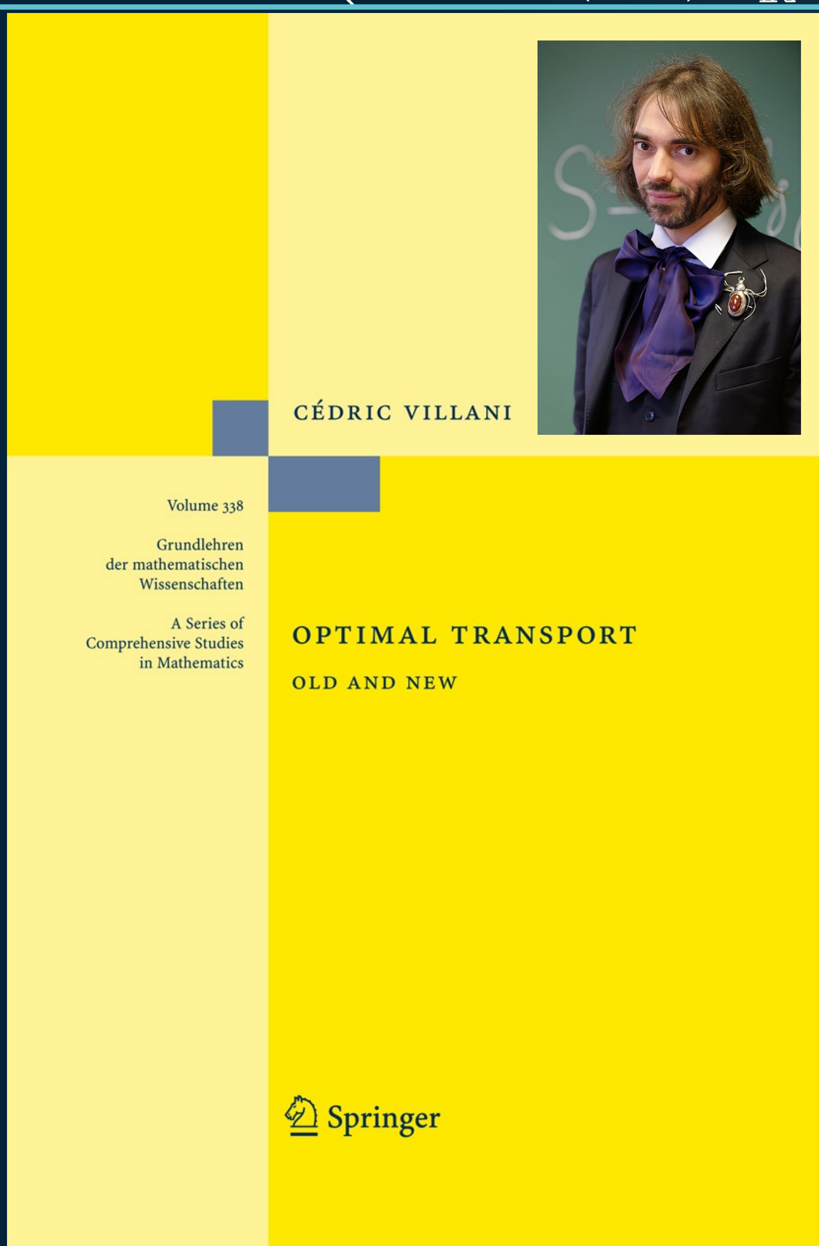
$$\Pi_{\text{con}}(\mu,\nu) = \{\gamma \geq 0, \int_{\mathbb{R}^d} \gamma(x,y)\mathrm{d}y = \mu, \int_{\mathbb{R}^d} \gamma(x,y)\mathrm{d}x = \nu\}$$

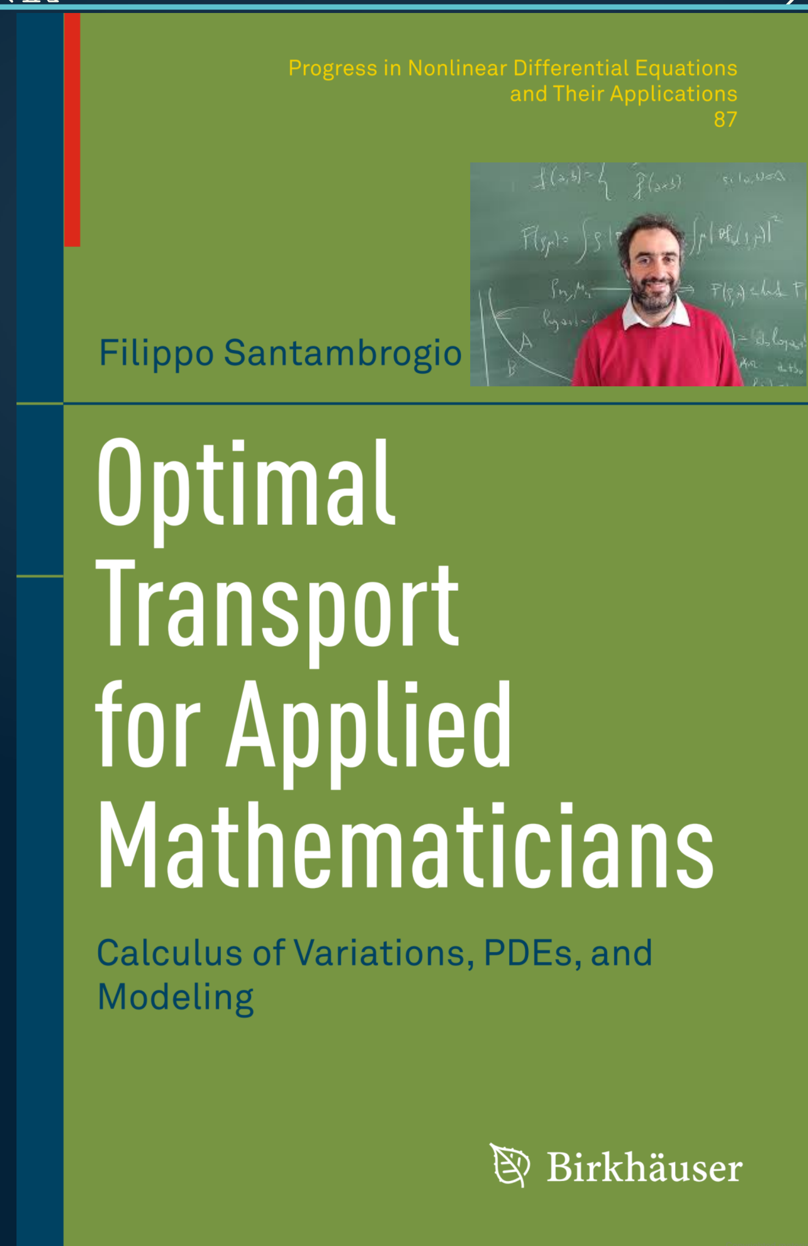


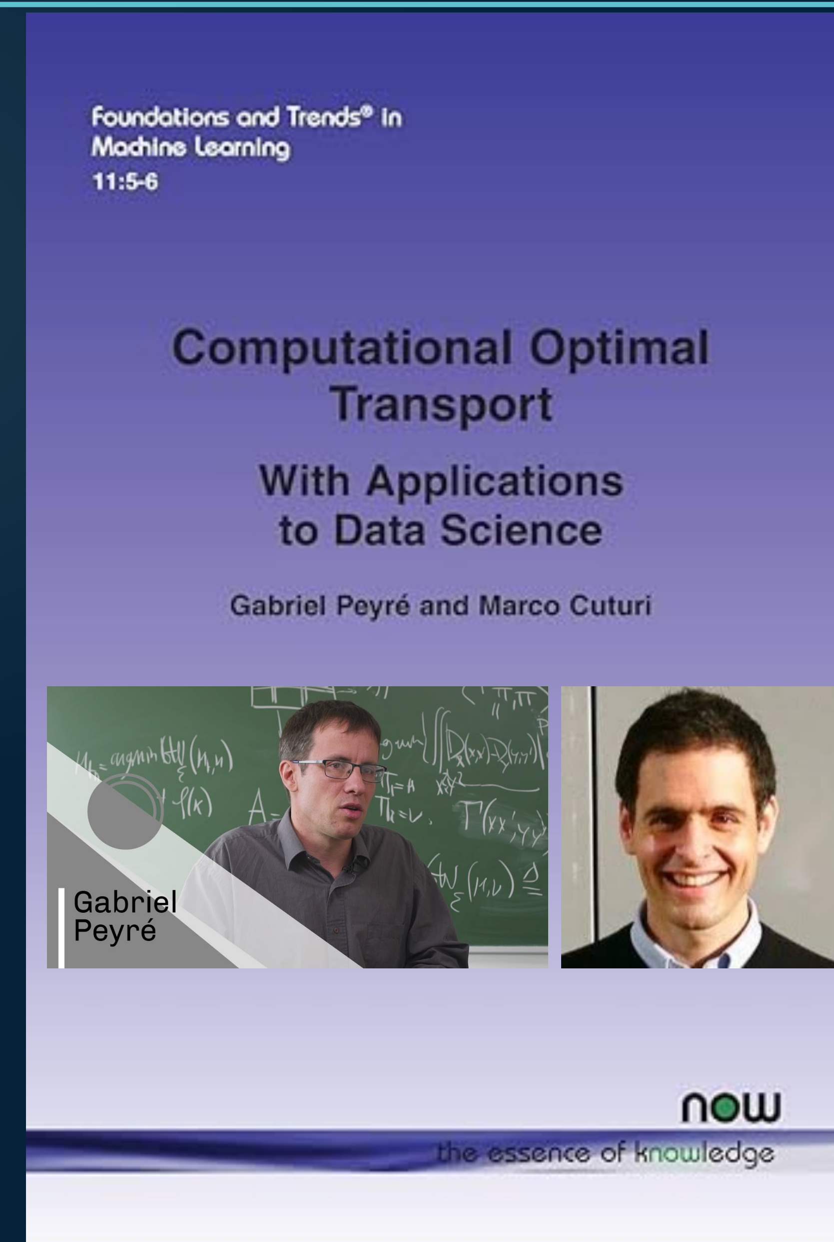
Continuous OT Framework: Wasserstein Distance

Wasserstein Distance

$$\mathrm{W}_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left(\inf_{\gamma \in \boldsymbol{\Pi}_{\mathrm{con}}(\boldsymbol{\mu}, \boldsymbol{\nu})} \int_{\mathbb{R}^d \times \mathbb{R}^d} D^p(\boldsymbol{x}, \boldsymbol{y}) \gamma(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} \right)^{1/p} = \left(\mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \gamma} [D^p(\boldsymbol{x}, \boldsymbol{y})] \right)^{1/p}$$

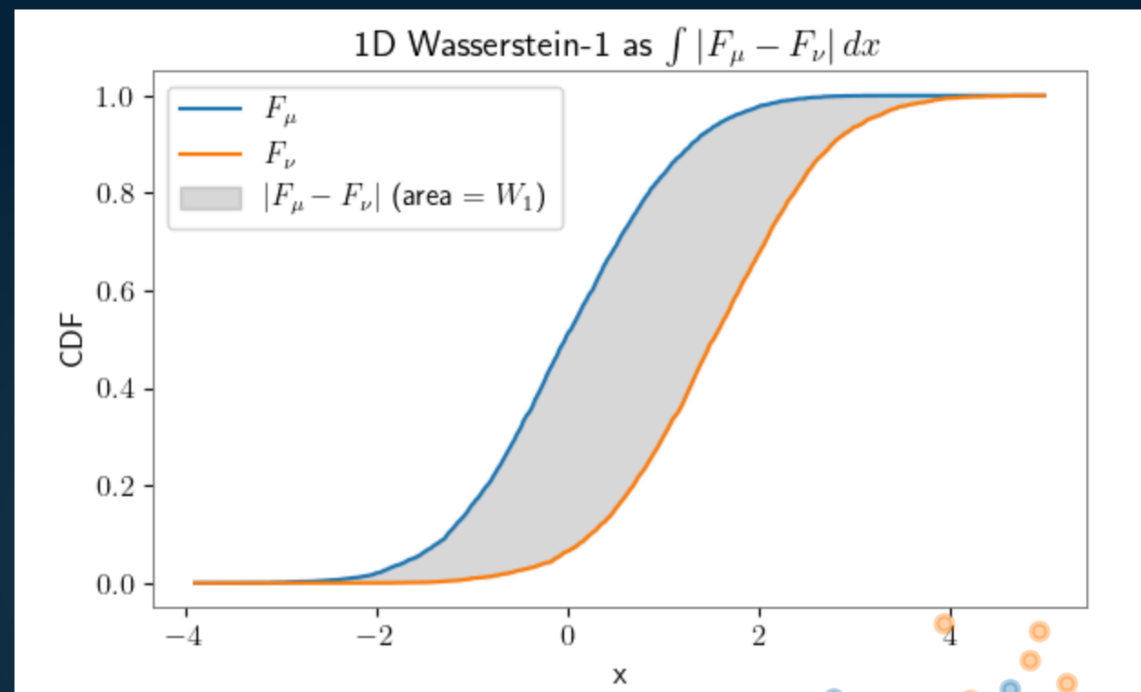


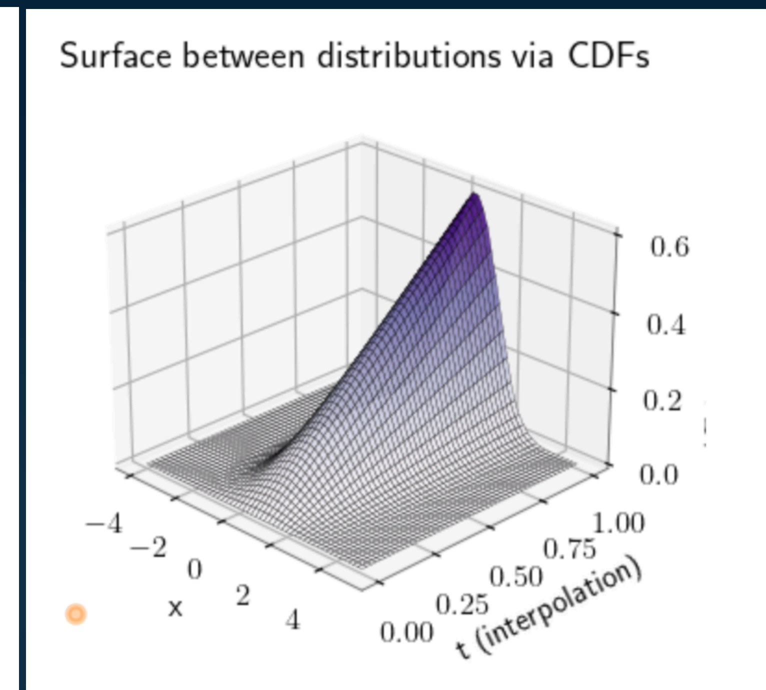




Continuous OT Framework: Wasserstein Distance,

$\mathrm{W}_1(\boldsymbol{\mu}, \boldsymbol{\nu})$

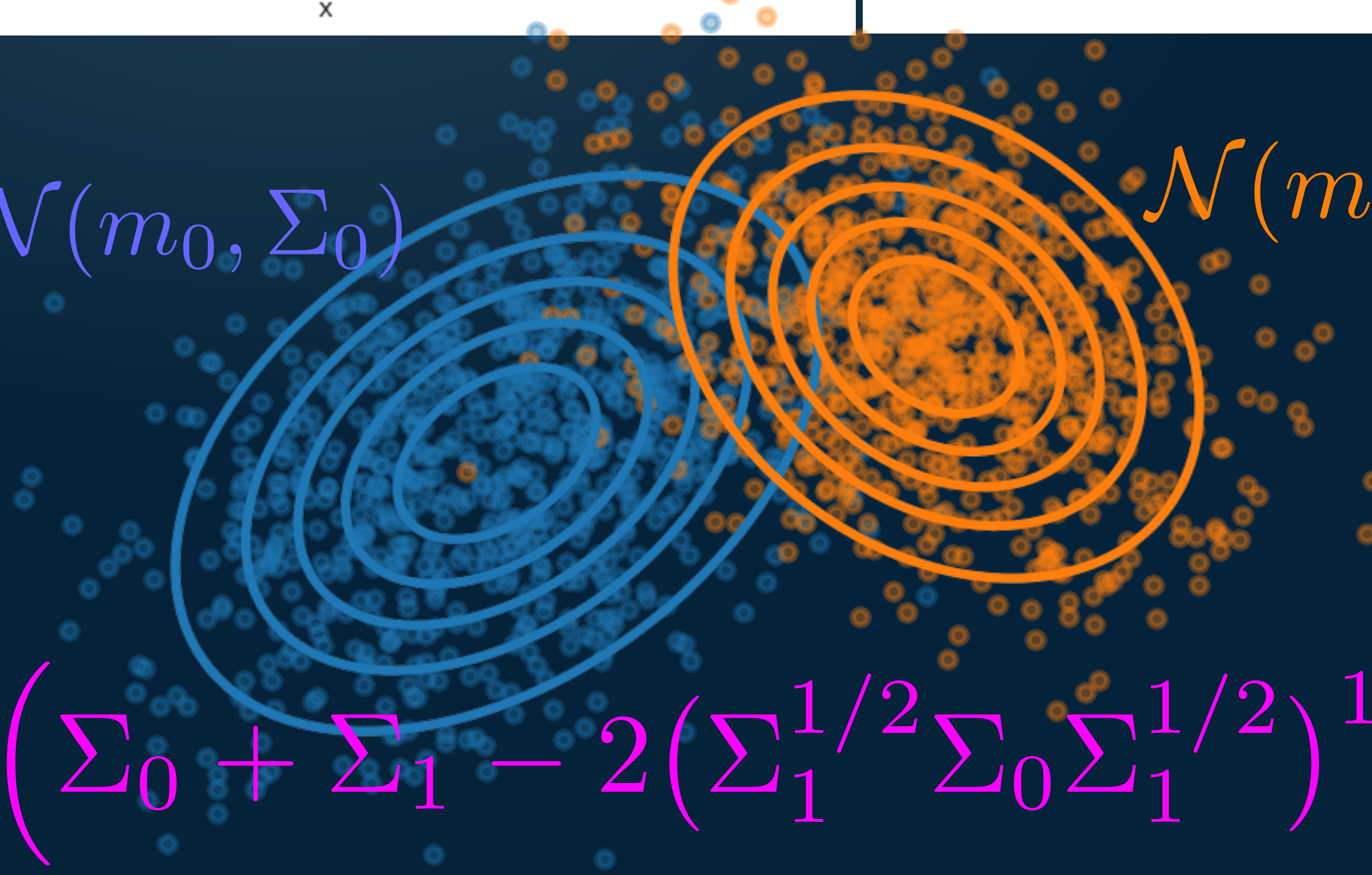




Bures Distance

$\mathcal{N}(m_0, \Sigma_0)$

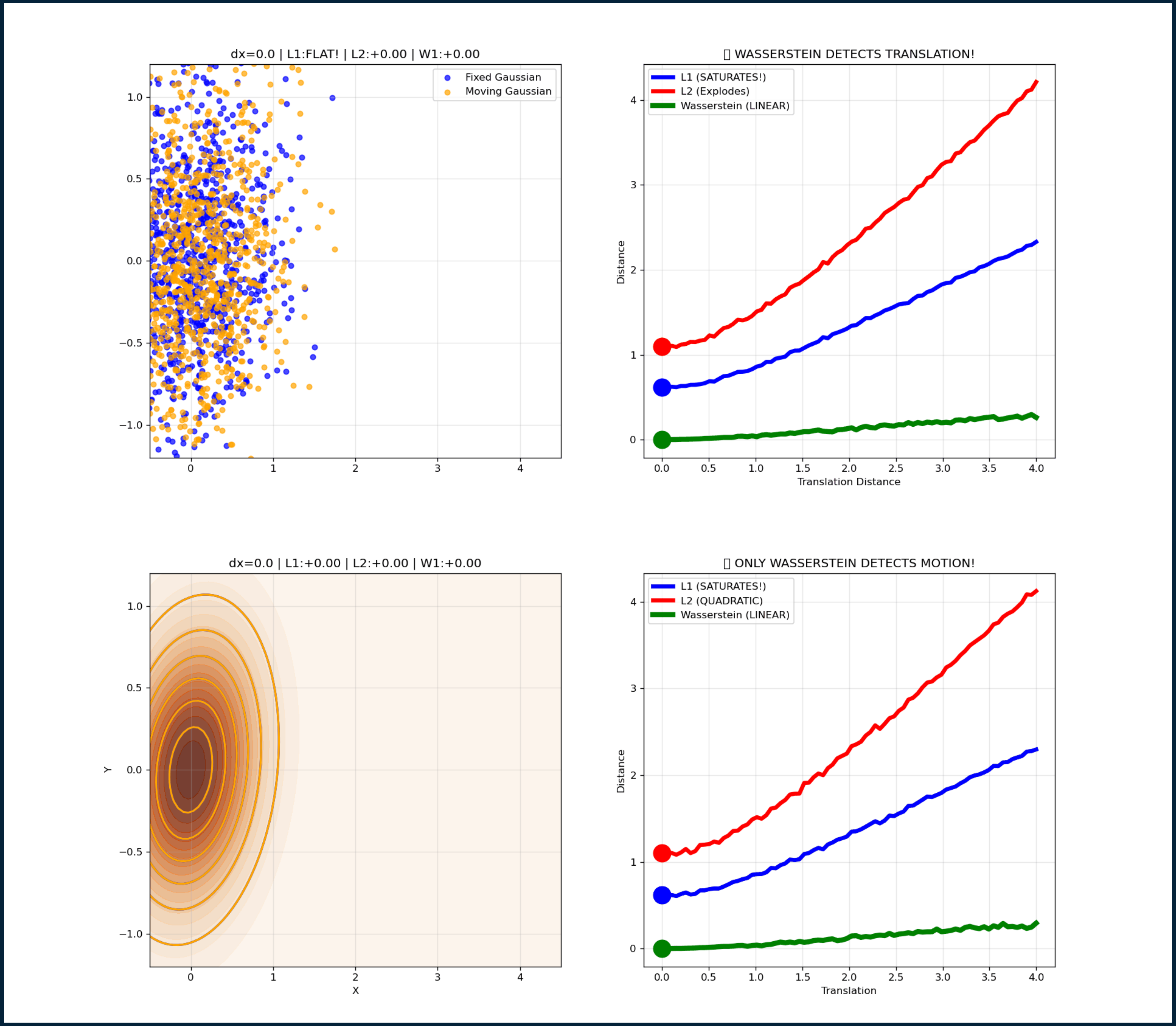
$\mathcal{N}(m_1, \Sigma_1)$



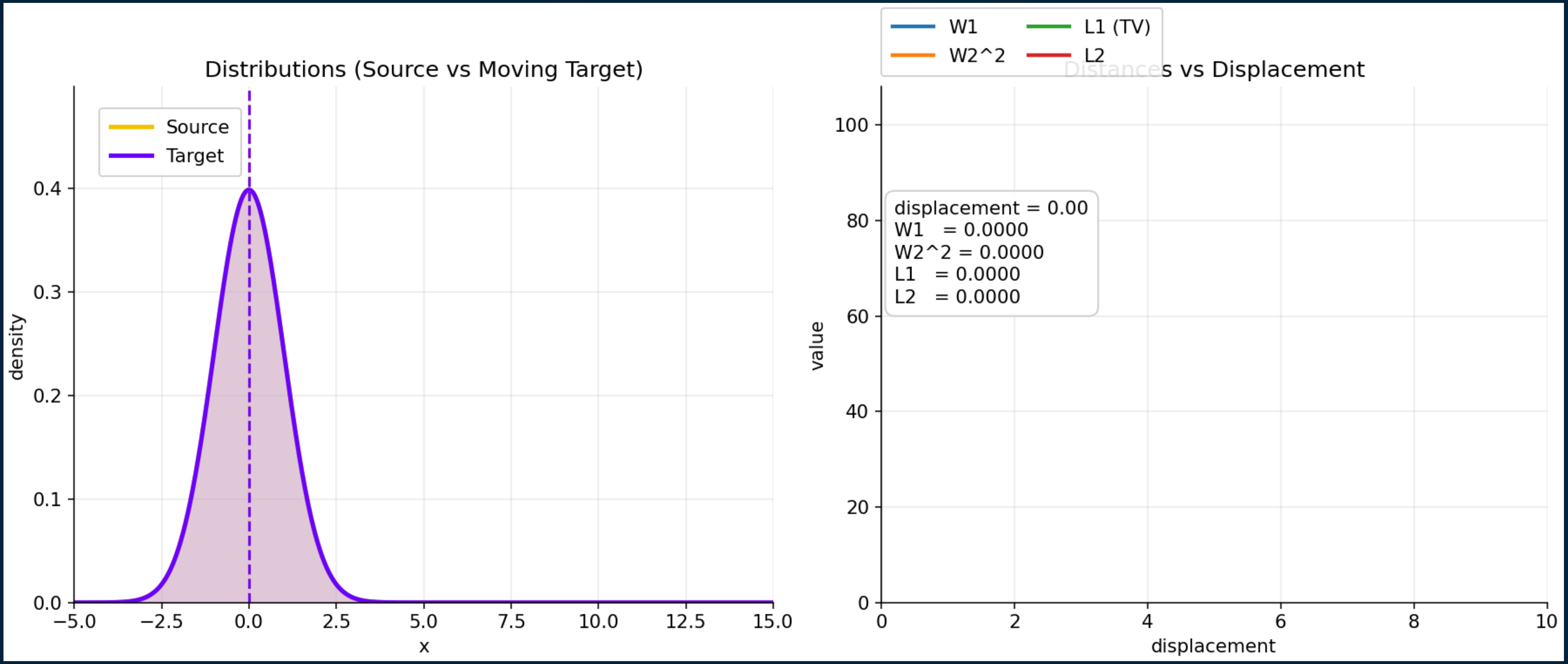
$$\mathrm{W}_2^2(\boldsymbol{\mu}, \boldsymbol{\nu})$$

$$= \|m_0 - m_1\|^2 + \mathrm{Tr}\Big(\Sigma_0 + \Sigma_1 - 2\big(\Sigma_1^{1/2}\Sigma_0\Sigma_1^{1/2}\big)^{1/2}\Big).$$

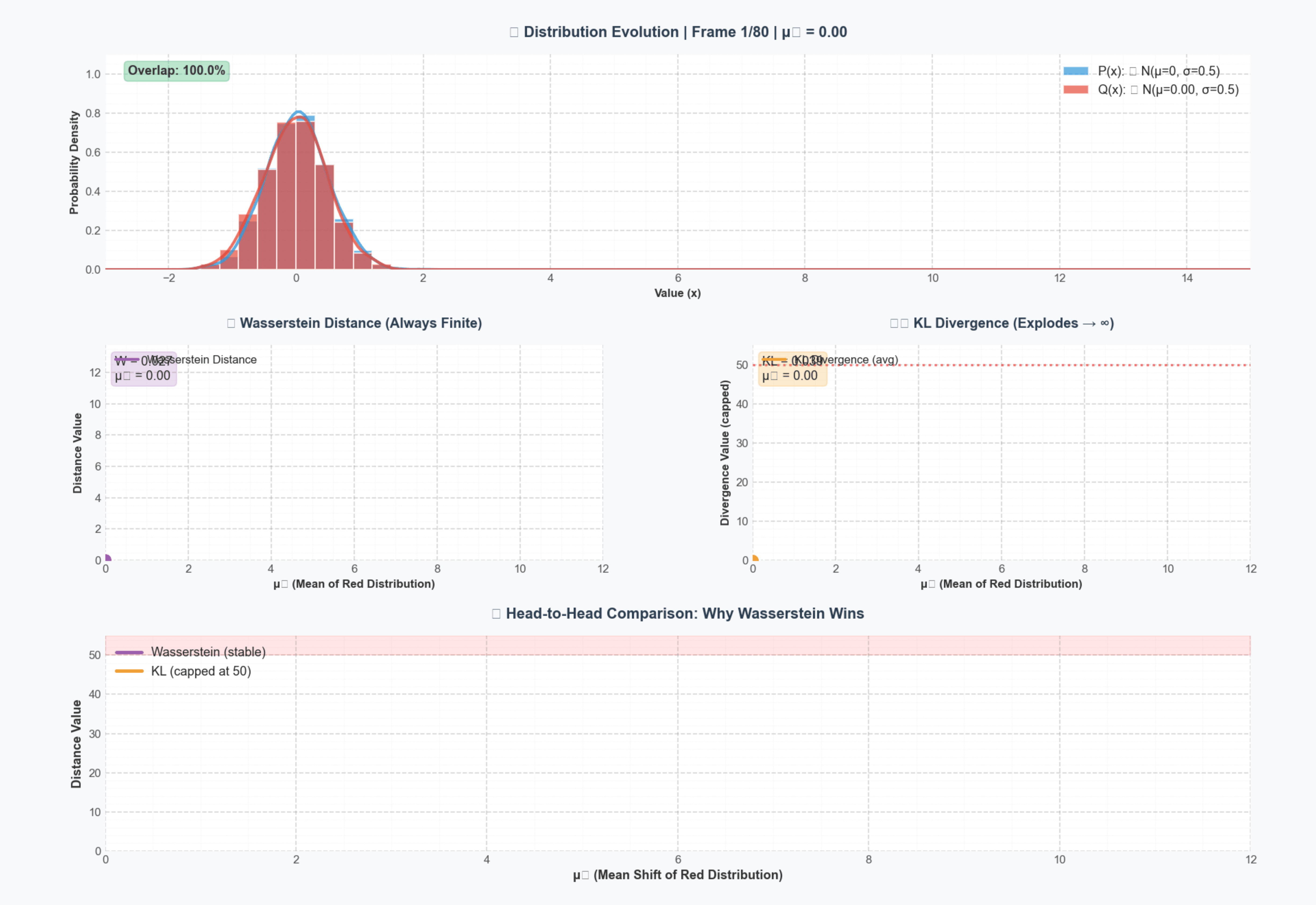
Benefit of Wasserstein Distance (1)



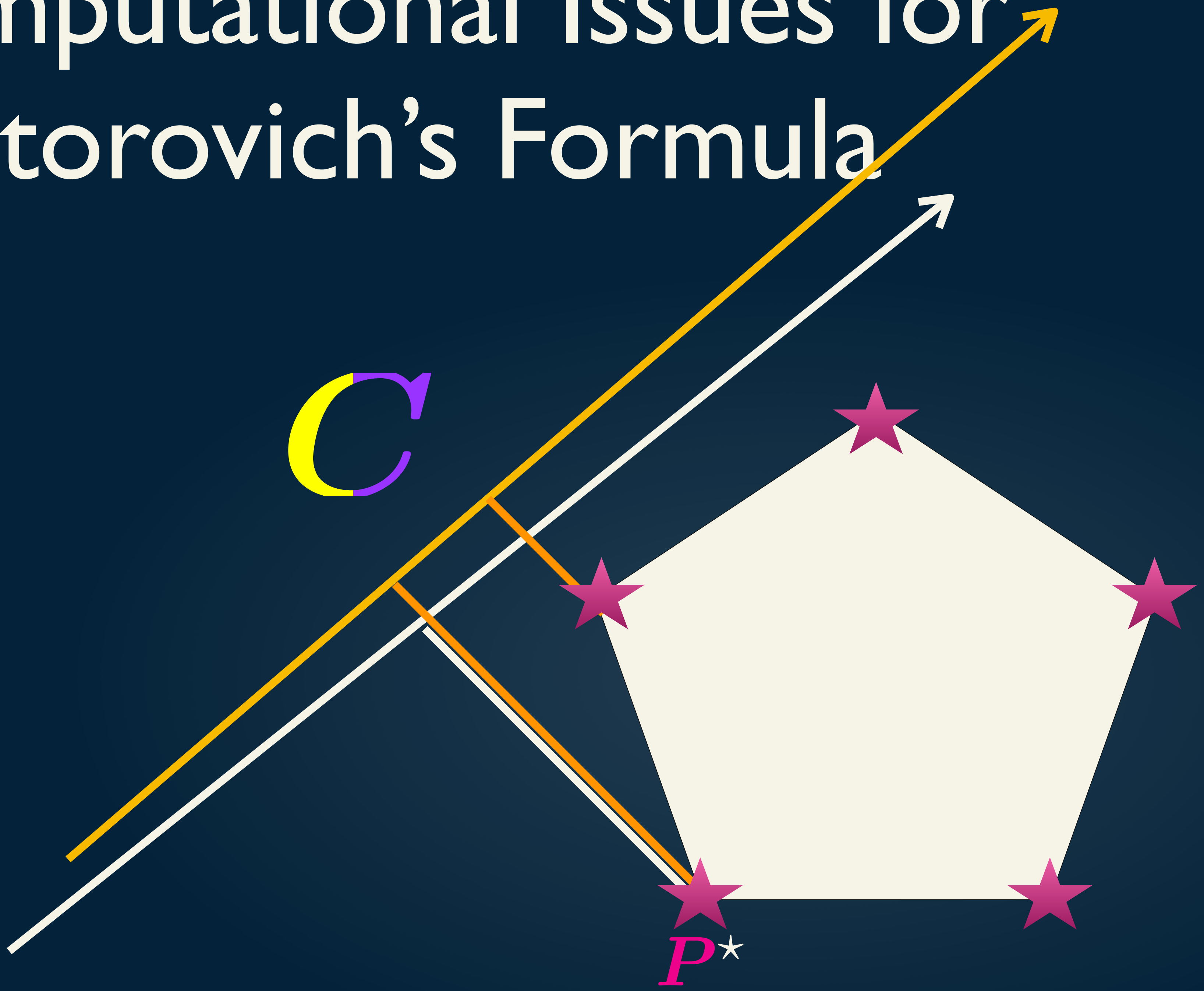
Benefit of Wasserstein Distance (2)



Benefit of Wasserstein Distance (3)



Computational Issues for Kantorovich's Formula



$\Pi(\mu, \nu)$ is the Birkhoff polytope

- No unique solution in some cases, numerical instabilities.
- Linear programming problem that requires generally $\mathcal{O}(n^3 \log(n)^2)$ arithmetic operations.

Regularized Discrete OT Framework: Sinkhorn Divergence

- Entropic regularization of OT distances relies on the addition of a penalty term as follows:

Regularisation parameter

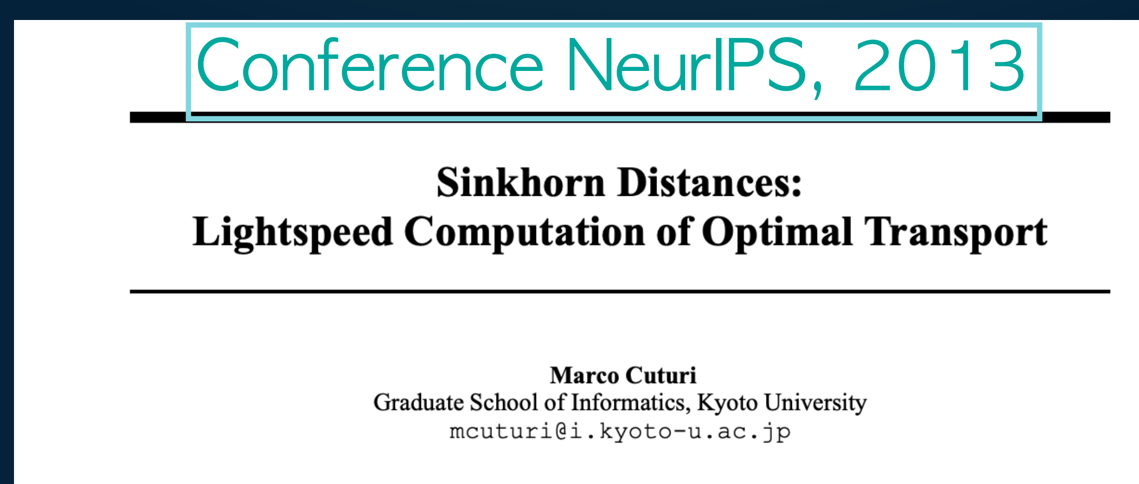
$$\mathcal{S}_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{P} \in \boldsymbol{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu})} \{ \langle \boldsymbol{C}, \boldsymbol{P} \rangle - \eta H(\boldsymbol{P}) \}$$

Sinkhorn divergence

Negative entropy

$$H(\boldsymbol{P}) = - \sum_{i,j} \boldsymbol{P}_{ij} \log(\boldsymbol{P}_{ij})$$

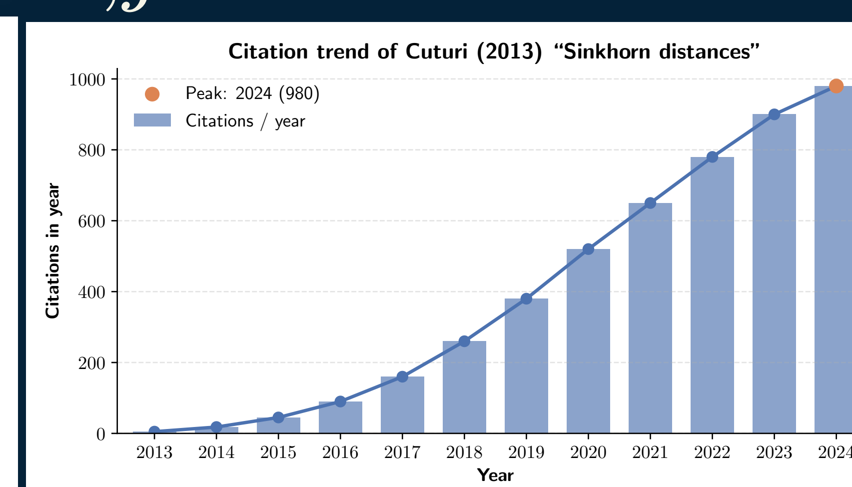
Marco Cuturi



Conference NeurIPS, 2013

Sinkhorn Distances: Lightspeed Computation of Optimal Transport

Marco Cuturi
Graduate School of Informatics, Kyoto University
mcuturi@i.kyoto-u.ac.jp



Regularized Discrete OT Framework: Dual of $\mathcal{S}_\eta(\boldsymbol{\mu}, \boldsymbol{\nu})$

Dual of Sinkhorn divergence

$$\mathcal{S}_\eta^{\mathrm{d}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{v}\in\mathbb{R}^m} \left\{ \Psi(\boldsymbol{u}, \boldsymbol{v}) := \mathbf{1}_n^\top B(\boldsymbol{u}, \boldsymbol{v}) \mathbf{1}_m - \langle \boldsymbol{u}, \boldsymbol{\mu} \rangle - \langle \boldsymbol{v}, \boldsymbol{\nu} \rangle \right\}$$

where

$$B(\boldsymbol{u}, \boldsymbol{v}) := \mathrm{diag}(e^{\boldsymbol{u}})\, \boldsymbol{K} \,\mathrm{diag}(e^{\boldsymbol{v}}) \qquad \boldsymbol{K} = e^{-\boldsymbol{C}/\eta}$$

Gibbs Kernel

- The primal optimal solution $\boldsymbol{P}^\star$ takes the form:

$$\boldsymbol{P}^\star = \mathrm{diag}(e^{\boldsymbol{u}^\star})\, \boldsymbol{K} \,\mathrm{diag}(e^{\boldsymbol{v}^\star})$$

Optimal Transportation Plan

with $$(\boldsymbol{u}^\star, \boldsymbol{v}^\star) = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{v}\in\mathbb{R}^m} \{\Psi(\boldsymbol{u}, \boldsymbol{v})\}$$

Dual Optimal Variables

Regularized Discrete OT Framework: Sinkhorn Algorithm

- $\boldsymbol{P}^\star$ can be solved efficiently by Sinkhorn iterations (near- $\mathcal{O}(n^2)$ complexity [Altschuler et al., 2017]).

$\textsc{Sinkhorn}(\boldsymbol{C}, \boldsymbol{\mu}, \boldsymbol{\nu})$

Matrix-Scaling Problem

1. $\boldsymbol{a}^{(0)} \leftarrow \mathbf{1}_n/n, \boldsymbol{b}^{(0)} \leftarrow \mathbf{1}_m/m;$
2. $\boldsymbol{K} \leftarrow e^{-\boldsymbol{C}/\eta};$
3. For $k = 1, 2, 3, \ldots$

   $\boldsymbol{a}^{(k)} \leftarrow \boldsymbol{\mu} \oslash \boldsymbol{K}\boldsymbol{b}^{(k-1)};$

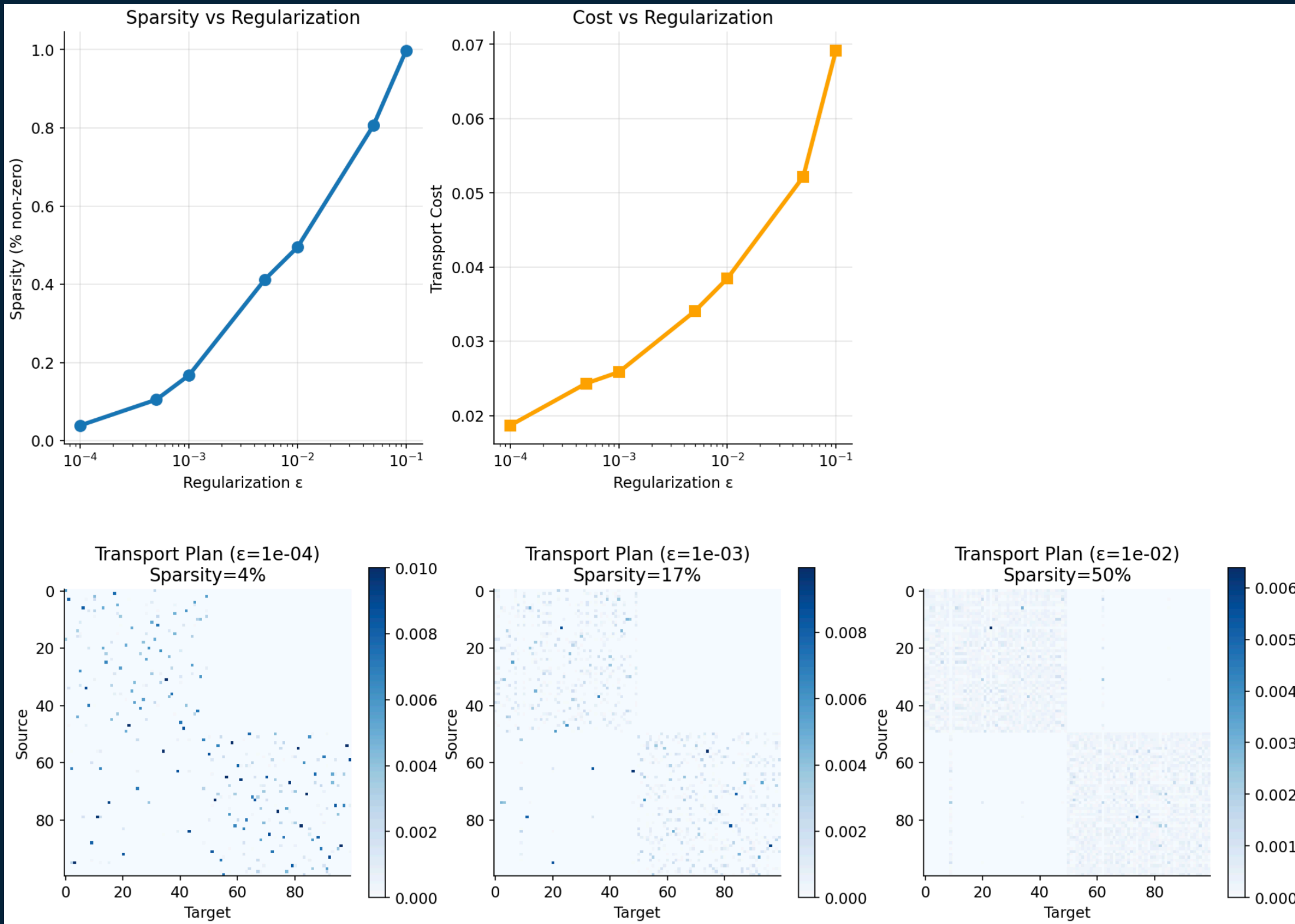
   $\boldsymbol{b}^{(k)} \leftarrow \boldsymbol{\nu} \oslash \boldsymbol{K}^\top \boldsymbol{a}^{(k-1)};$
4. Return $\mathrm{diag}(\boldsymbol{a}^{(k)})\ \boldsymbol{K}\,\mathrm{diag}(\boldsymbol{b}^{(k)})$

python POT

(Flamary et al. 2017)

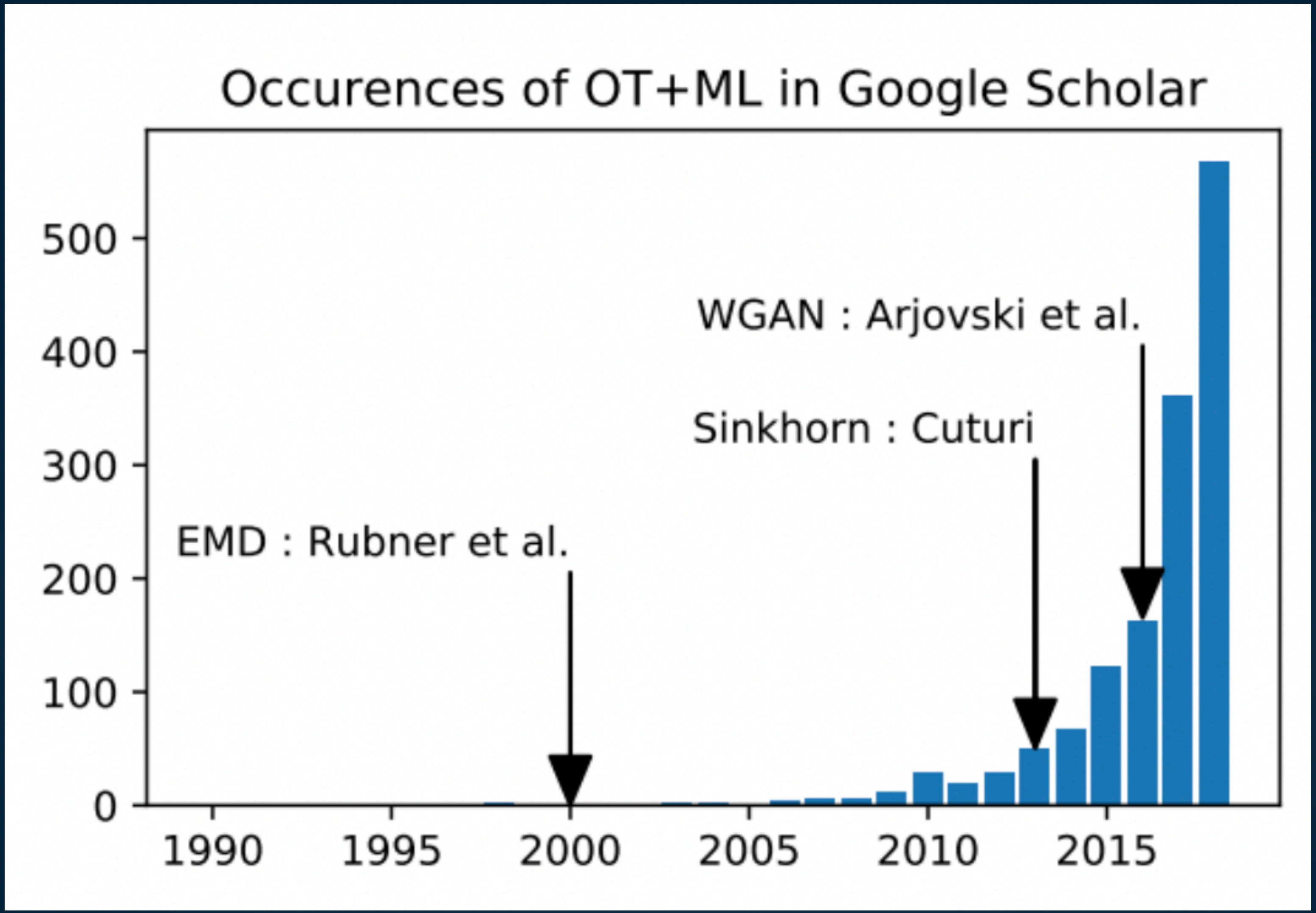
```
from ot import sinkhorn
P_star = sinkhorn(mu, nu, C, eta)
```

Sparsity of Transport Plan



4. How can it be used in data science?

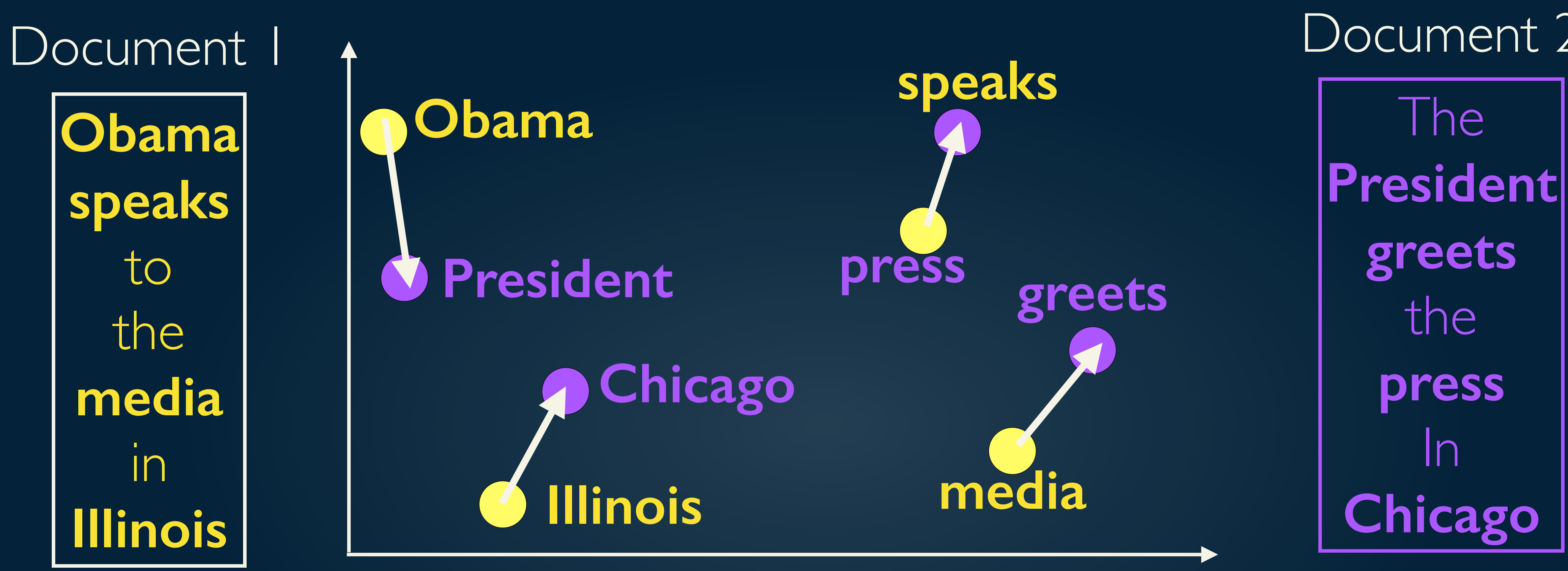
History of OT for machine learning



[R. Flamary, 2019 (HDR)]

- Recently introduced to ML (well know in image processing since 2000).
- Computational OT allows numerous applications (regularization).
- Deep learning boost (numerical optimisation and GAN).

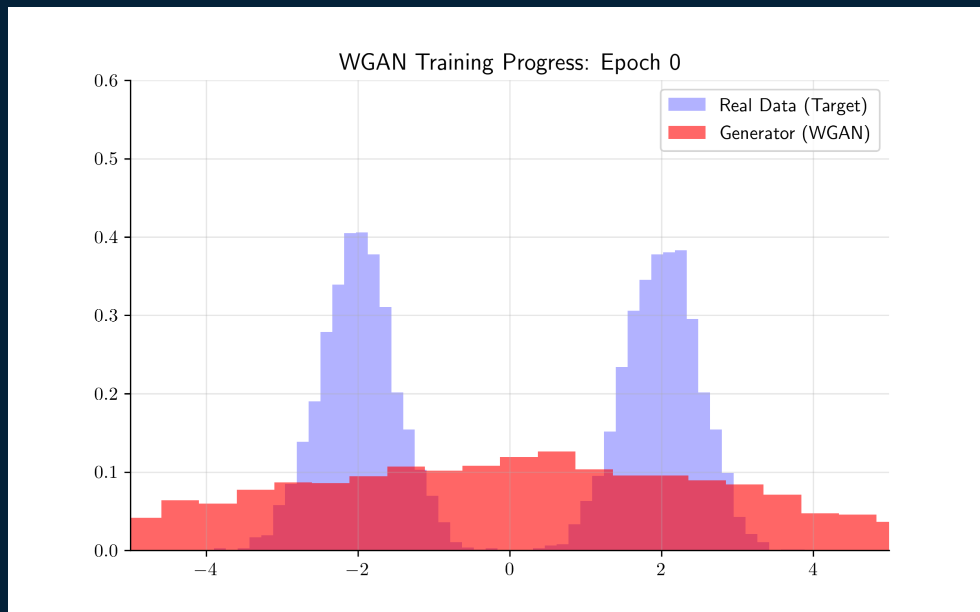
Matching words embeddings

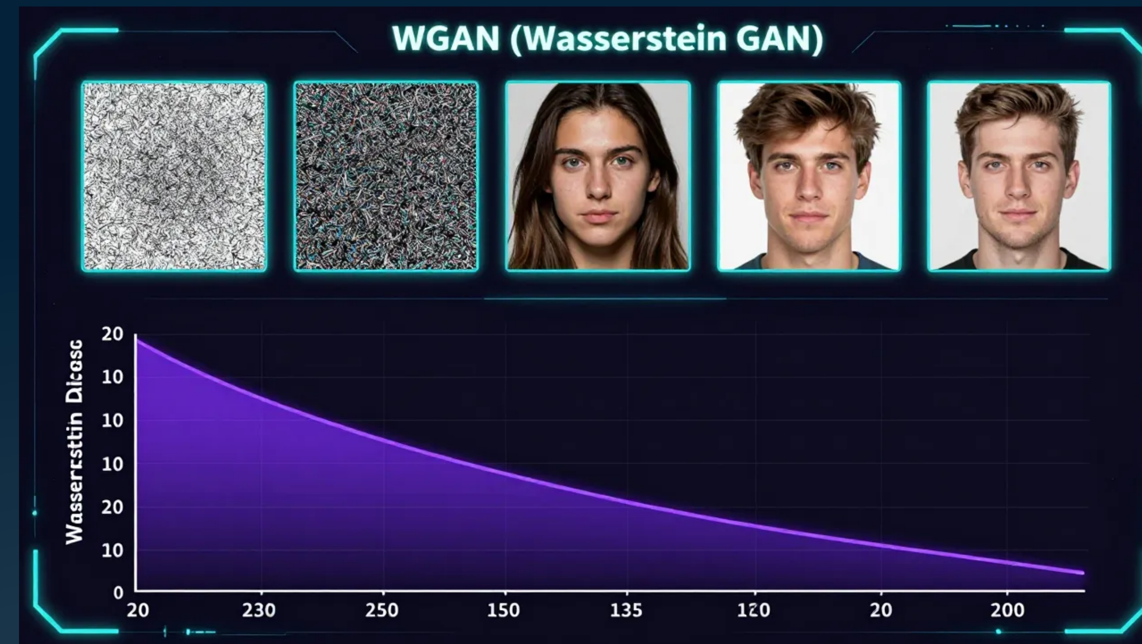


Word Mover's Distance avec Word2vec embeddings
[Kusner et al, 2015 (ICML)]

- Words are embedded in a high-dimensional space with deep neural networks.
- Matching two documents in an OT problem, with the Euclidean distance in the embedded space.

Wasserstein loss for generative modelling





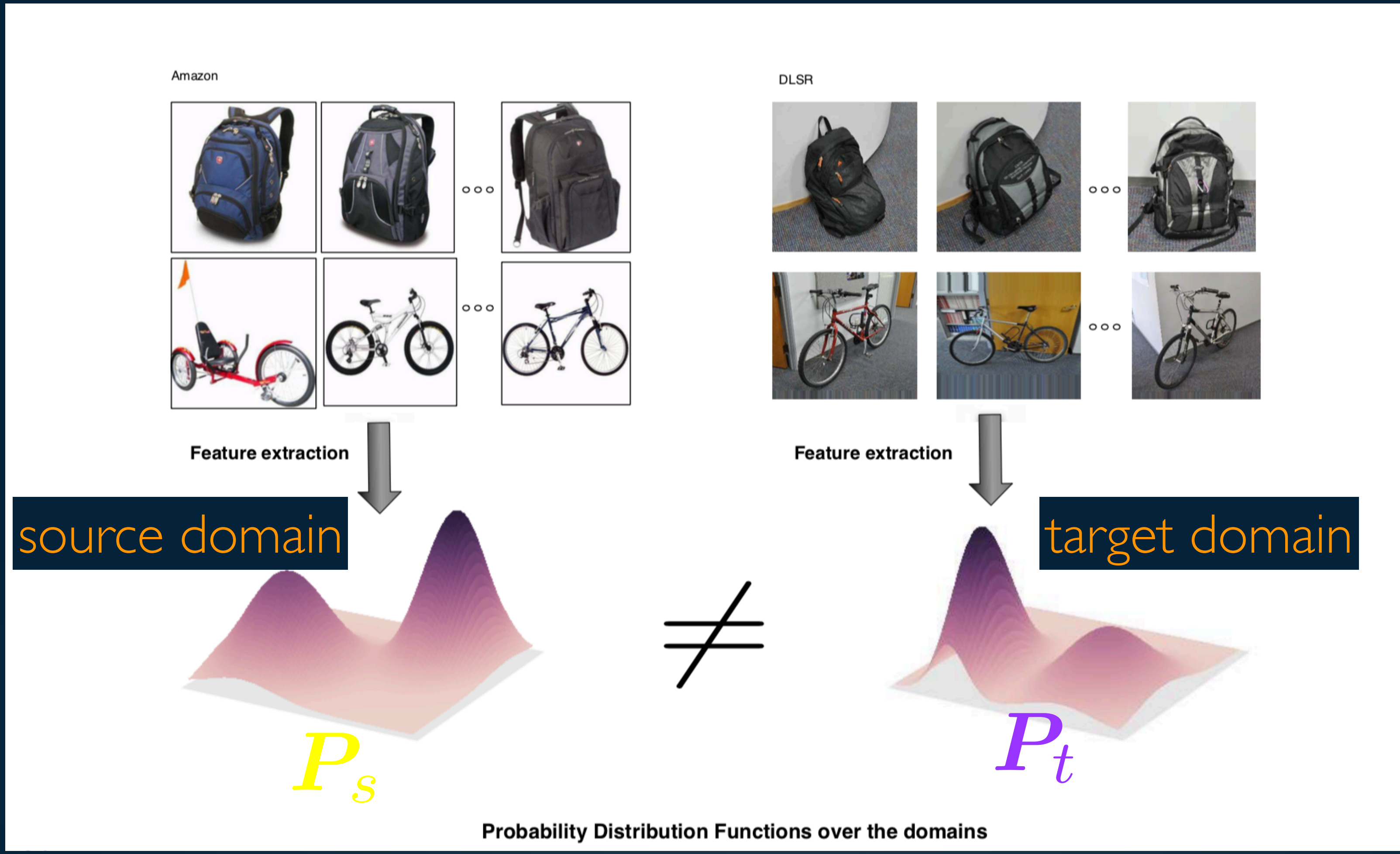
Generative modelling as a matching distribution problem

- Learn a model that maps random vector to target space.
- Distribution of the model is targeted to be similar to the learning samples.
- Similarity as Wasserstein sense [Arjovsky et al. 2017, Deshpande et al. 2018, Nguyen et al. 2020].

$$\min_{f_{\boldsymbol{\theta}}} W_p^p\Big(\{f_{\boldsymbol{\theta}}(\boldsymbol{z}_i)\}_{i=1}^K, \{\boldsymbol{x}_j\}_{j=1}^K\Big)$$

$\{\boldsymbol{z}_i\}$ some random vectors, $\{\boldsymbol{x}_j\}$ some samples from the target distribution.

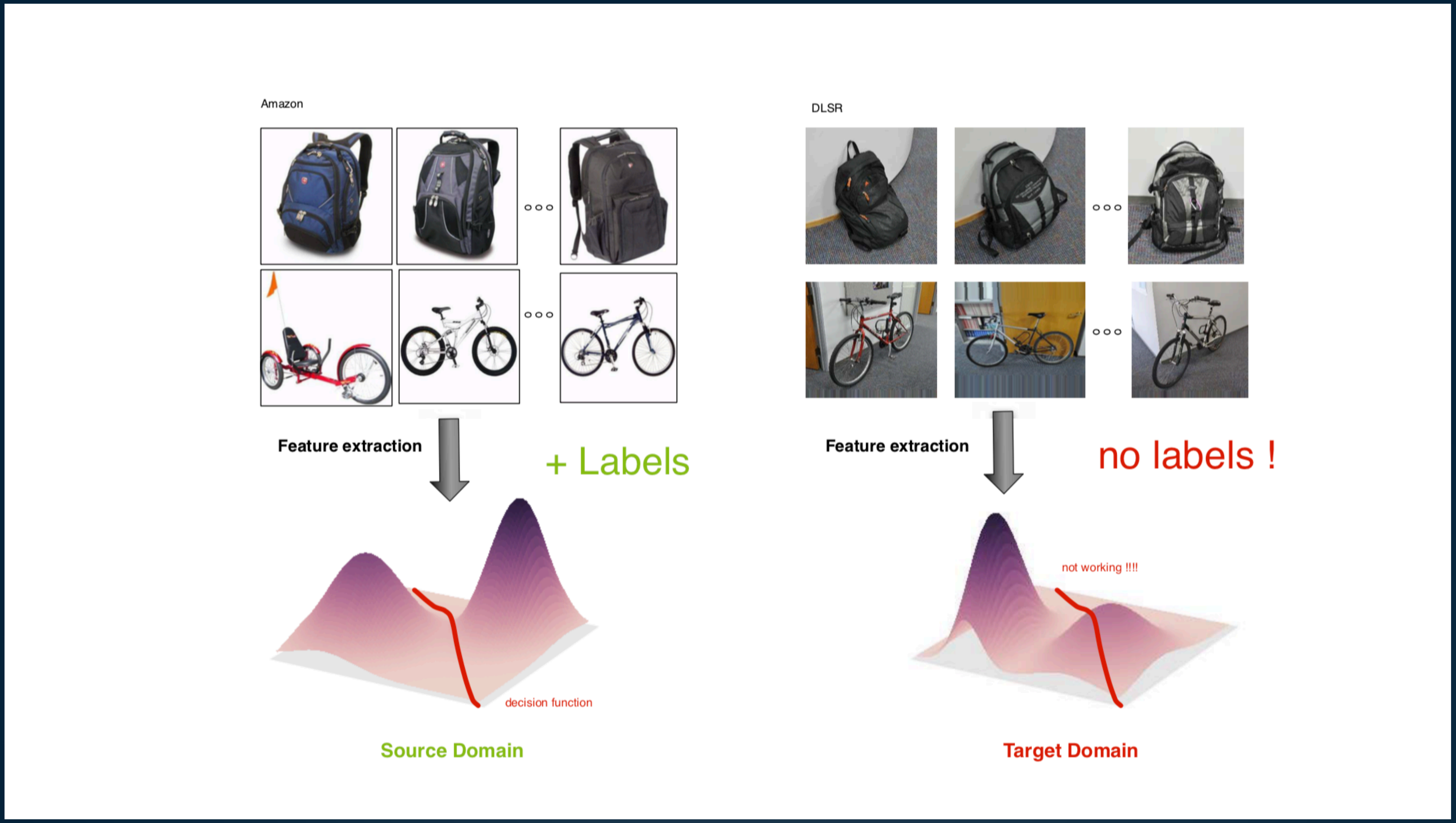
Unsupervised Domain Adaptation



- Traditional machine learning hypothesis:
  - We have access to training data. Probability distribution of the training set and the testing are the same.
  - We want to learn a classifier that generalizes to new data.

(Credit image: N. Courty)

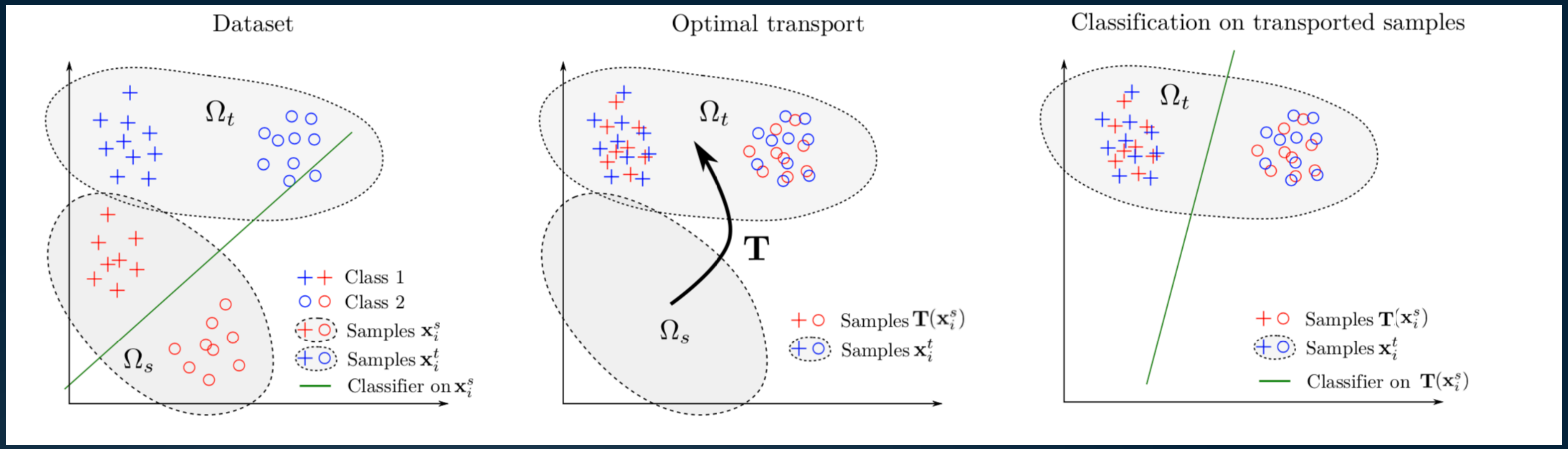
Unsupervised Domain Adaptation



- Domain adaptation: classification problem with data coming from different sources (domains).
- Labels only available in the source domain, and classification is conducted in the target domain.
- Classifier trained on the source domain data performs badly in the target domain.

(Credit image: N. Courty)

OTDA: Optimal Transport Domain Adaptation



Assumptions

- There exist a transport $\boldsymbol{T}$ in the feature space between the two domains.
- The transport preserves the conditional distributions:

$$\boldsymbol{P}_s[\boldsymbol{y}|\boldsymbol{x}_s] = \boldsymbol{P}_t[\boldsymbol{y}|\boldsymbol{T}(\boldsymbol{x}_s)]$$

3-step strategy

1. Estimate optimal transport between distributions.
2. Transport the training samples onto the target distribution using barycentric mapping (Ferradans et al., 2013)
3. Learn a classifier on the transported training samples.

(Source image: Courty et al., 2017)

5. (Smoothed, Sliced) OT

Wasserstein Distance: curse of dimensionality

$$\mathrm{W}_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left(\inf_{\boldsymbol{\gamma} \in \boldsymbol{\Pi}_{\mathrm{con}}(\boldsymbol{\mu}, \boldsymbol{\nu})} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{x} - \boldsymbol{y}\|^p \boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} \right)^{1/p}$$

- The Wasserstein distance is often estimated from samples.

$$\hat{\boldsymbol{\mu}}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{\boldsymbol{x}_i}$$

- The error of these empirical estimates suffers from an exponential dependence on dimension d that presents an obstacle to sample-efficient bounds.

$$\mathbb{E}[\mathrm{W}_p(\hat{\boldsymbol{\mu}}_n, \boldsymbol{\mu})] \lesssim n^{-1/d}$$

[Altschuler et al., 2017, Weed & Bach, 2019, Lei, 2020]

Smoothed-Wasserstein Distance

- To alleviate the curse of dimensionality of empirical W_p, **Gaussian smoothing** was recently introduced.
  [Goldfeld et al., 2020; Goldfeld & Greenewald, 2020]

- The σ-smooth p-Wasserstein distance between probability measures is defined as

$$\mathrm{G}_\sigma \mathrm{W}_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \mathrm{W}_p(\boldsymbol{\mu} * \mathcal{N}_\sigma, \boldsymbol{\nu} * \mathcal{N}_\sigma)$$

- Fast rate of convergence [Nietertet al., 2021]

$$\mathbb{E}[\mathrm{G}_\sigma \mathrm{W}_p(\hat{\boldsymbol{\mu}}_n, \boldsymbol{\mu})] \lesssim n^{-1/2}$$

Smoothed-Wasserstein Distance

- When $d = 1$, the Wasserstein distance can be calculated in closed-form owing to the cumulative distributions of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$.

- In practice for empirical distributions, the closed-form solution requires only the sorting of the samples, which makes it very efficient.

- To derive a metric for high-dimensional distributions based on 1D Wasserstein distance.

- The main idea is to project high-dimensional probability distributions onto a random one-dimensional space and then to compute the Wasserstein distance.

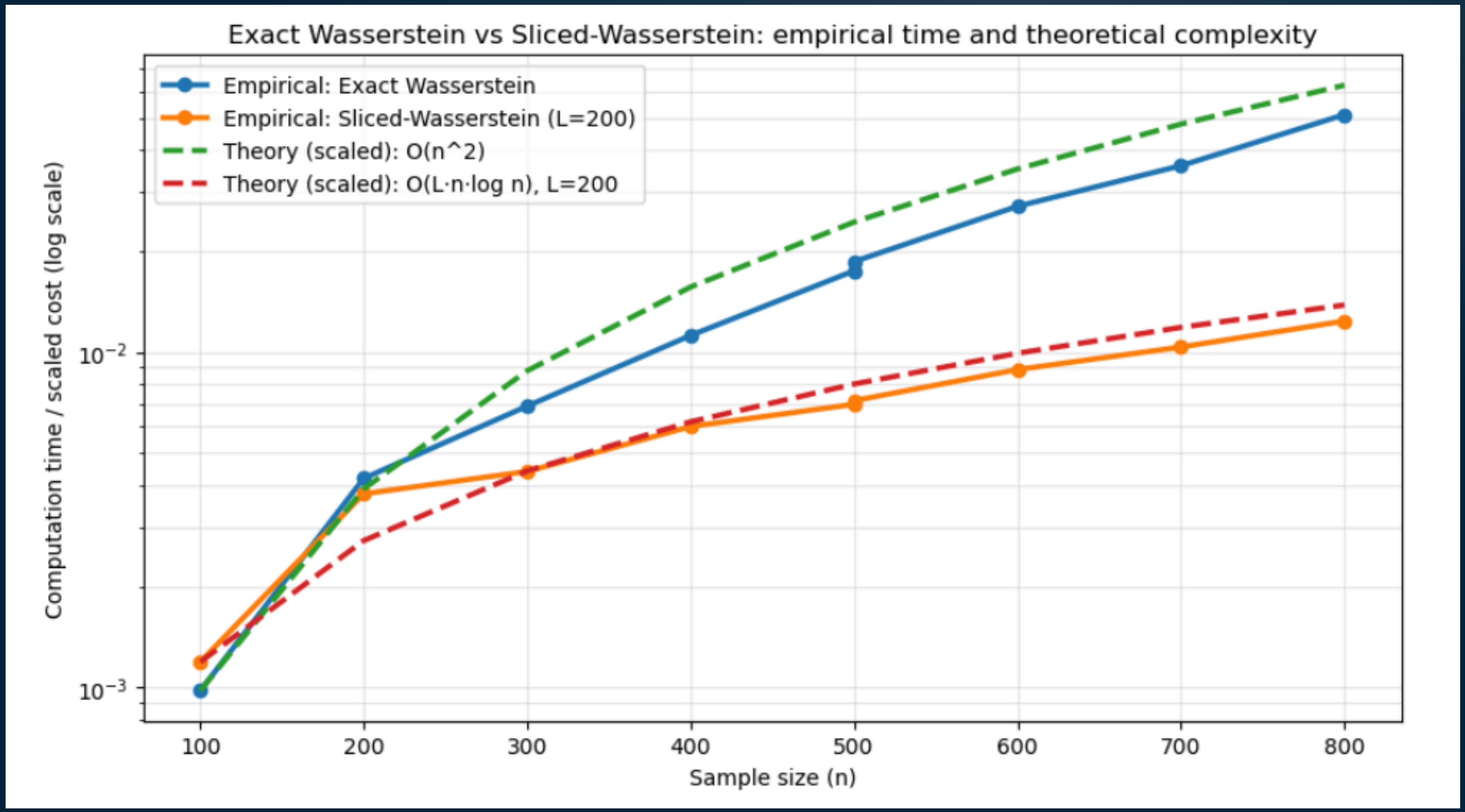
Sliced-Wasserstein Distance

- The sliced Wasserstein distance (SW) reads as

$$\mathrm{SW}_p(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left(\int_{\mathbb{S}^{d-1}} \mathrm{W}_p^p(\mathcal{R}_{\mathbf{u}}\boldsymbol{\mu}, \mathcal{R}_{\mathbf{u}}\boldsymbol{\nu}) u_d(\mathbf{u}) \mathrm{d}\mathbf{u} \right)^{1/p}.$$

Where $\mathcal{R}_{\mathbf{u}}$ the Radon transform of a probability distribution, i.e.,

$$\mathcal{R}_{\mathbf{u}}\boldsymbol{\mu}(\cdot) = \int_{\mathbb{R}^d} \boldsymbol{\mu}(\mathbf{s}) \delta(\cdot - \mathbf{s}^\top \mathbf{u}) \mathrm{d}\mathbf{s}$$



Complexity of SW

$O(Ldn + Ln \log n)$

Published in Transactions on Machine Learning Research (11/2024)

TRANSACTIONS tmlr on ML RESEARCH

Gaussian-Smoothed Sliced Probability Divergences

Mokhtar Z. Alaya *alayaelm@utc.fr*
Université de Technologie de Compiègne,
LMAC (Laboratoire de Mathématiques Appliquées de Compiègne), CS 60 319 - 60 203 Compiègne Cedex

Alain Rakotomamonjy *a.rakotomamonjy@criteo.com*
Criteo AI Lab, Paris, France,

Maxime Berar *maxime.berar@univ-rouen.fr*
Univ Rouen Normandie, INSA Rouen Normandie, Universite Le Havre Normandie
Normandie Univ, LITIS UR4108, Rouen, France

Gilles Gasso *gilles.gasso@insa-rouen.fr*
INSA Rouen Normandie, Univ Rouen Normandie, Universite Le Havre Normandie,
Normandie Univ, LITIS UR4108, Rouen, France

Reviewed on OpenReview: *https://openreview.net/forum?id=weuALLWUV2*

Smoothed-Sliced-Wasserstein Divergence

Gaussian-smoothed sliced divergence

The σ-Gaussian-smooth p-Sliced Divergence between probability measures is defined as

$$\mathrm{G}_\sigma\mathrm{SD}_p(\boldsymbol{\mu},\boldsymbol{\nu}) = \left(\int_{\mathbb{S}^{d-1}} \mathrm{D}^p(\mathcal{R}_\mathbf{u}\boldsymbol{\mu} * \mathcal{N}_\sigma, \mathcal{R}_\mathbf{u}\boldsymbol{\nu} * \mathcal{N}_\sigma) u_d(\mathbf{u})\mathrm{d}\mathbf{u}\right)^{1/p}.$$

- Typical relevant divergences : Sinkhorn divergence or maximum mean discrepancy (MMD)

$$\mathrm{MMD}^2(\boldsymbol{\mu},\boldsymbol{\nu}) = \|\Phi_k(\boldsymbol{\mu}) - \Phi_k(\boldsymbol{\nu})\|^2_{\mathcal{H}_k}$$

$$= \mathbb{E}_{T,T'\sim\boldsymbol{\mu}}[k(T,T')] - 2\mathbb{E}_{T\sim\boldsymbol{\mu},R\sim\boldsymbol{\nu}}[k(T,R)] + \mathbb{E}_{R,R'\sim\boldsymbol{\nu}}[k(R,R')]$$

[Gretton et al. (2012)]

Smoothed-Sliced-Wasserstein Divergence: topology properties

Theorem 3.1. *For any $\sigma > 0, p \geq 1$, the following properties hold:*

1. *if $\mathrm{D}(\cdot,\cdot)$ is non-negative (or symmetric), then $\mathrm{G}_\sigma\mathrm{SD}_p(\cdot,\cdot)$ is non-negative (or symmetric);*
2. *if $\mathrm{D}(\cdot,\cdot)$ satisfies the identity of indiscernibles, i.e. for $\mu', \nu' \in \mathcal{P}(\mathbb{R})$, $\mathrm{D}(\mu',\nu') = 0$ if and only if $\mu' = \nu'$, then this identity also holds for $\mathrm{G}_\sigma\mathrm{SD}_p(\cdot,\cdot)$ for any $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$;*
3. *if $\mathrm{D}(\cdot,\cdot)$ satisfies the triangle inequality then $\mathrm{G}_\sigma\mathrm{SD}_p(\cdot,\cdot)$ satisfies the triangle inequality.*

Theorem 3.2. *Let $\sigma > 0, p \geq 1$, $\mu \in \mathcal{P}_p(\mathbb{R}^d)$, and $\{\mu_k \in \mathcal{P}_p(\mathbb{R}^d)\}_{k\in\mathbb{N}}$ a sequence of distributions. Assume that the divergence D is bounded and metrizes the weak topology on $\mathcal{P}(\mathbb{R})$. Then, $\lim_{k\to\infty} \mathrm{G}_\sigma\mathrm{SD}_p(\mu_k,\mu) = 0$ if and only if $\mu_k \Rightarrow \mu$.*

Proposition 3.3. *Let $\sigma > 0, p \geq 1$ and assume that the base divergence D is lower semi-continuous w.r.t. the weak topology in $\mathcal{P}(\mathbb{R})$. Then, $\mathrm{G}_\sigma\mathrm{SD}_p$ is lower semi-continuous with respect to the weak topology in $\mathcal{P}_p(\mathbb{R}^d)$.*

Smoothed-Sliced-Wasserstein Divergence: statistical properties (1)

- The smoothed Gaussian sliced divergence between the empirical probability measures $\hat{\mu}_n$ and $\hat{\nu}_n$

$$\mathrm{G}_\sigma\mathrm{SD}_p(\hat{\mu}_n, \hat{\nu}_n) = \left(\int_{\mathbb{S}^{d-1}} \mathrm{D}^p(\mathcal{R}_{\mathbf{u}}\hat{\mu}_n * \mathcal{N}_\sigma, \mathcal{R}_{\mathbf{u}}\hat{\nu}_n * \mathcal{N}_\sigma) u_d(\mathbf{u}) \mathrm{d}\mathbf{u} \right)^{1/p}.$$

Remark 3.4. Remark that for a fixed $\mathbf{u} \in \mathbb{S}^{d-1}$, the distributions $\mathcal{R}_{\mathbf{u}}\hat{\mu}_n * \mathcal{N}_\sigma$ and $\mathcal{R}_{\mathbf{u}}\hat{\nu}_n * \mathcal{N}_\sigma$ are *continuous*, in particular they are a mixture of Gaussian distributions centered on the projected samples with variance σ^2.

Lemma 3.5. *Conditionally on the samples* $\{X_i\}_{i=1,\ldots,n}$ *and* $\{Y_i\}_{i=1,\ldots,n}$, *one has:* $\mathcal{R}_{\mathbf{u}}\hat{\mu}_n * \mathcal{N}_\sigma = \frac{1}{n}\sum_{i=1}^n \mathcal{N}(\mathbf{u}^\top X_i, \sigma^2)$ *and* $\mathcal{R}_{\mathbf{u}}\hat{\nu}_n * \mathcal{N}_\sigma = \frac{1}{n}\sum_{i=1}^n \mathcal{N}(\mathbf{u}^\top Y_i, \sigma^2)$.

Smoothed-Sliced-Wasserstein Divergence: statistical properties (II)

Let $T_1^x, \ldots, T_n^x$ and $T_1^y, \ldots, T_n^y$ be i.i.d. observations of $\mathcal{R}_{\mathbf{u}}\hat{\mu}_n * \mathcal{N}_\sigma$ and $\mathcal{R}_{\mathbf{u}}\hat{\nu}_n * \mathcal{N}_\sigma$, respectively. Sampling i.i.d. $\{T_i^x\}_{i=1,\ldots,n}$ is given by the following scheme: for $i = 1, \ldots, n$, we first choose the component $\mathcal{N}(\mathbf{u}^\top X_i, \sigma^2)$ from the mixture $\frac{1}{n}\sum_{i=1}^n \mathcal{N}(\mathbf{u}^\top X_i, \sigma^2)$ then we generate $T_i^x = \mathbf{u}^\top X_i + Z_i^x$, where $Z_i^x \sim \mathcal{N}_\sigma$. Hence, we set, for a given $\mathbf{u}$

$$\hat{\hat{\mu}}_n = \frac{1}{n}\sum_{i=1}^n \delta_{T_i^x} = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{u}^\top X_i + Z_i^x} \text{ and } \hat{\hat{\nu}}_n = \frac{1}{n}\sum_{i=1}^n \delta_{T_i^y} = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{u}^\top Y_i + Z_i^y}.$$

Double empirical divergence

The double empirical smoothed Gaussian sliced divergence reads as

$$\hat{\mathrm{G}}_\sigma\mathrm{SD}_p(\hat{\hat{\mu}}_n, \hat{\hat{\nu}}_n) = \left(\int_{\mathbb{S}^{d-1}} \mathrm{D}^p(\hat{\hat{\mu}}_n, \hat{\hat{\nu}}_n) u_d(\mathbf{u}) \mathrm{d}\mathbf{u}\right)^{1/p}.$$

Smoothed-Sliced-Wasserstein Divergence: statistical properties (III)

Proposition 3.8. *Fix $\sigma > 0, p \geq 1$ and $\vartheta > \sqrt{2}$. For $X \sim \mu$, assume that $\int_0^\infty e^{\frac{2\xi^2}{\sigma^2\vartheta^2}} \mathbf{P}\left[\|X\| > \xi\right] d\xi < \infty$. Then,*

$$\mathbf{E}_{\mu^{\otimes n}|\mathcal{N}_\sigma^{\otimes n}}[\hat{\mathrm{G}}_\sigma \mathrm{SW}_p(\hat{\mu}_n, \mu)] \leq \Xi_{p,\sigma,\vartheta} \frac{1}{n^{1/2p}} + \Upsilon_{p,\sigma,\mu} \frac{(\log n)^{1/p}}{n^{1/p}},$$

where $\Xi_{p,\sigma,\vartheta} = \frac{2^{\frac{5}{2}-\frac{5}{4p}}}{\pi^{1/2p}} \sigma^{1-\frac{1}{4p}} \vartheta^{1+\frac{1}{p}} \big(\Gamma(p+\frac{1}{2})\big(\sqrt{\frac{4\pi\sigma^2\vartheta^2}{\vartheta^2-2}} + 4\int_0^\infty e^{\frac{2\xi^2}{\sigma^2\vartheta^2}} \mathbf{P}[\|X\| > \xi] d\xi\big)\big)^{1/2p}$ *and* $\Upsilon_{p,\sigma,\mu} = \frac{2^{2-\frac{1}{2p}} C_p}{\pi^{1/2p}} \sigma^2 \big(\Gamma(p+\frac{1}{2}) \sum_{k=0}^\infty \frac{(-p)_k}{(\frac{1}{2})_k} \frac{(-1)^k}{(2\sigma^2)^k k!} M_{2k}(\mu)\big)^{1/p}$ *with C_p is a positive constant depending only on p.*

Double empirical divergence

The double empirical smoothed Gaussian sliced divergence reads as

$$\hat{\mathrm{G}}_\sigma \mathrm{SD}_p(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\nu}}_n) = \left(\int_{\mathbb{S}^{d-1}} \mathrm{D}^p(\hat{\hat{\boldsymbol{\mu}}}_n, \hat{\hat{\boldsymbol{\nu}}}_n) u_d(\mathbf{u}) \mathrm{d}\mathbf{u}\right)^{1/p}.$$

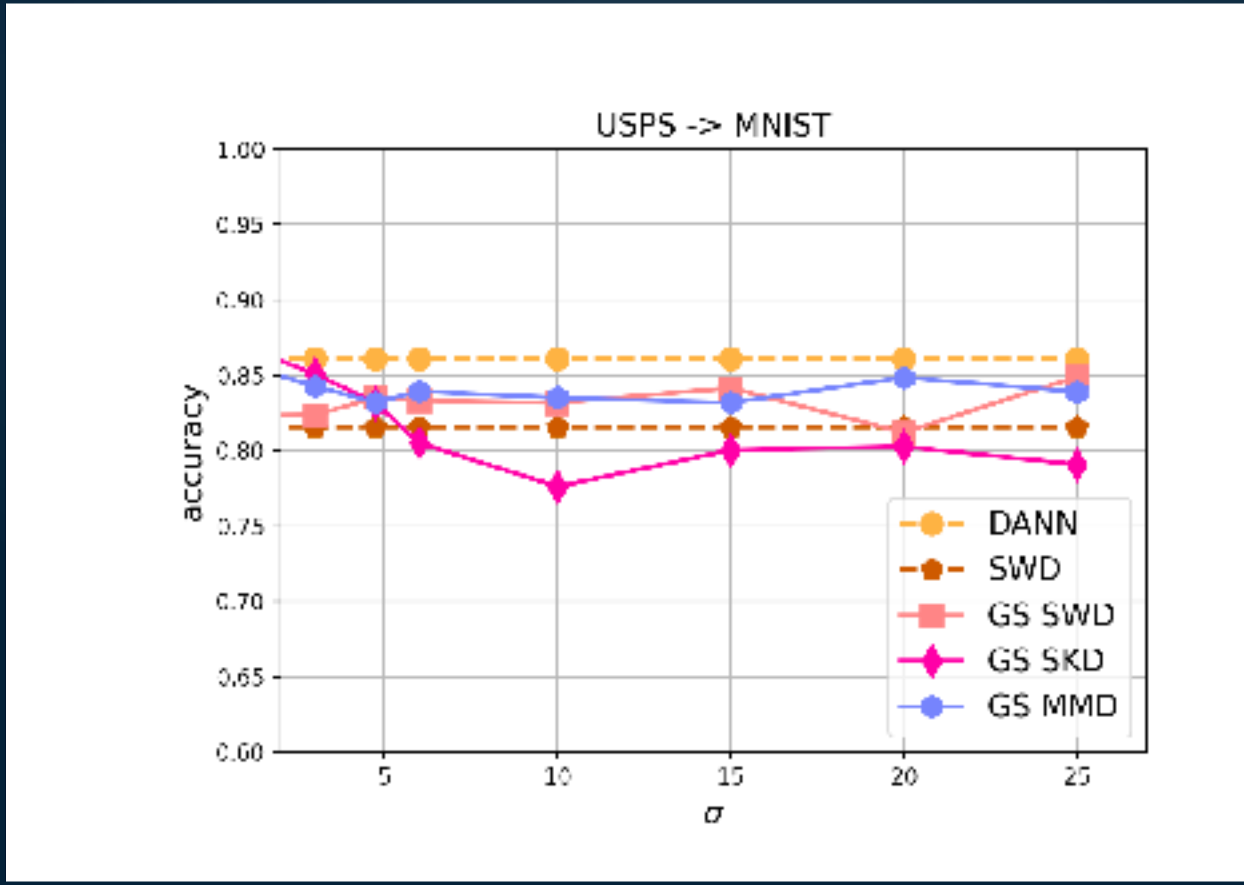
Domain adaptation with $\mathrm{G}_\sigma\mathrm{SW}$: MNIST(source) to USPS (target)

MNIST

USPS

Domain adaptation with $\mathrm{G}_\sigma\mathrm{SW}$: MNIST(source) to USPS (target)

$$\min_{g,h} \left\{ \mathcal{L}_c(h(g(\mathbf{X}_s)), \mathbf{y}_s) + \mathcal{D}(g(\mathbf{X}_s), g(\mathbf{X}_t)) \right\}$$



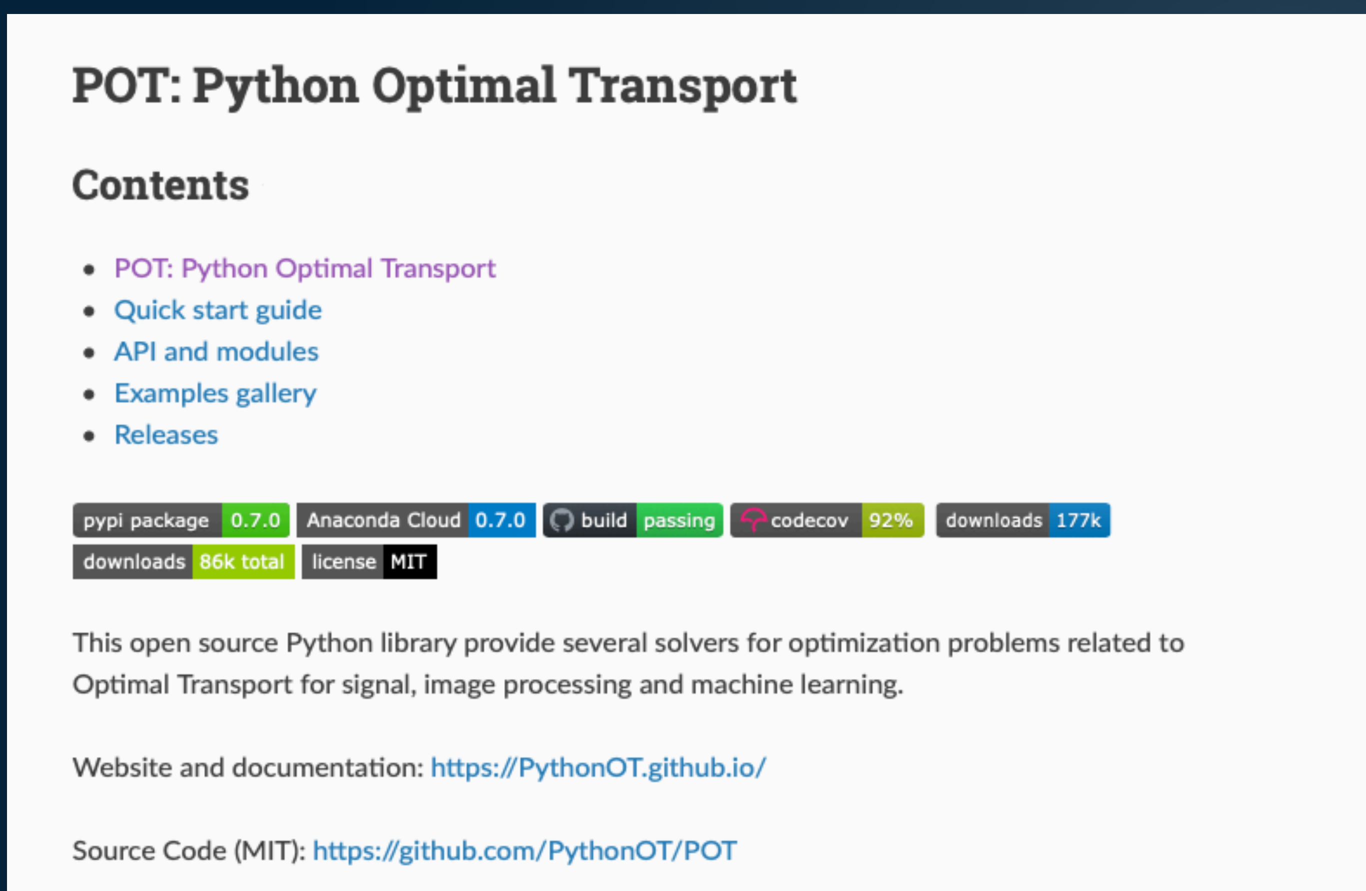
Domain adaptation performances using different divergences on distributions with respect to the Gaussian smoothing.

Take Home Message

- A powerful tool, well theoretically grounded, for manipulating distributions in machine learning.
- Despite its initial computational complexity, a lot of applications, even in large scale/deep learning settings.
- Others OT aspects (out the scope of the presentation): Gromov-Wasserstein distance (working with structured data), Sliced Wasserstein, Multimarginal Optimal Transport (MOT) and many more !

Some References

- G. Peyré and M. Cuturi,
  Computational Optimal Transport with Applications to Data Sciences, 2019
- N. country, R. Flamary, D. Tuia and A. Rakotomamonjy.
  Optimal Transport for Domain Adaptation, *PAMI 2017*
- R. Flamary et al. POT: Python Optimal Transport Library, 2017.



POT: Python Optimal Transport

Contents

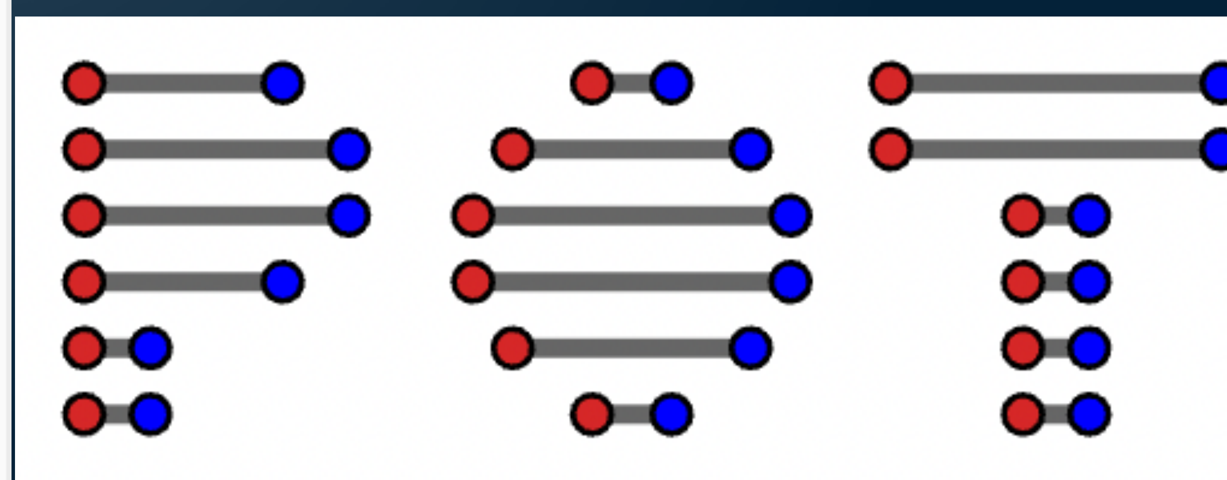
- POT: Python Optimal Transport
- Quick start guide
- API and modules
- Examples gallery
- Releases

pypi package 0.7.0 | Anaconda Cloud 0.7.0 | build passing | codecov 92% | downloads 177k | downloads 86k total | license MIT

This open source Python library provide several solvers for optimization problems related to Optimal Transport for signal, image processing and machine learning.

Website and documentation: https://PythonOT.github.io/

Source Code (MIT): https://github.com/PythonOT/POT



Thank You for your attention!