Computational optimal transport: mature tools and open problems

Jean Feydy
HeKA team, Inria Paris
Inserm, Université Paris-Cité

24th of November, 2022
Measure-theoretic approaches and optimal transportation in statistics
Institut Henri Poincaré

Who am I?

Background in **mathematics** and **data sciences**:

2012–2016 ENS Paris, mathematics.

2014–2015 M2 mathematics, vision, learning at ENS Cachan.

2016–2019 PhD thesis in **medical imaging** with Alain Trouvé at ENS Cachan.

2019–2021 **Geometric deep learning** with Michael Bronstein at Imperial College.

2021+ **Medical data analysis** in the HeKA INRIA team (Paris).

Close ties with **healthcare**:

2015 Image denoising with **Siemens Healthcare** in Princeton.

2019+ MasterClass AI–Imaging, for **radiology interns** in the University of Paris.

2020+ Colloquium on **Medical imaging in the AI era** at the Paris Brain Institute.

My main motivation: speeding up core computations for healthcare

Computational anatomy. 3D medical scans are orders of magnitude heavier than natural 2D images:

- 100k triangles to represent a brain surface.
- 512x512x512 $\simeq$ 130M voxels for a typical 3D image.

Public health. Over the last decade, medical datasets have **blown up**:

- Clinical trials: **1k patients**, controlled environment.
- UK Biobank: **500k people**, curated data.
- French Health Data Hub: **70M people**, full social security data since ~2000.

Medical doctors, pharmacists and governments need scalable methods.

A field that is moving fast

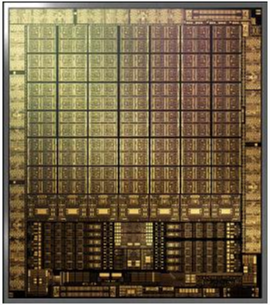
Target. Scale up models that combine medical **expertise** with modern **datasets**.

Context. The advent of **Graphics Processing Units** (GPU):

- Incredible **value for money**:

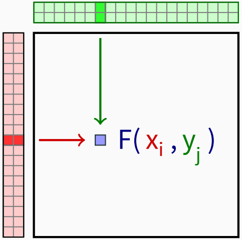
  1 000€ $\simeq$ 1 000 cores $\simeq 10^{12}$ operations/s.
- **Bottleneck**: constraints on **register** usage.

“User-friendly” Python ecosystem, consolidated around a **small number of key operations**.



7,000 cores in a single GPU.

The KeOps library: efficient support for symbolic matrices



Symbolic matrix

Formula + data

- Distances $d(x_i,y_j)$.
- Kernel $k(x_i,y_j)$.
- Numerous transforms.

Solution. KeOps – www.kernel-operations.io:

- For PyTorch, NumPy, Matlab and R, on **CPU and GPU**.
- **Automatic differentiation**.
- Just-in-time **compilation** of **optimized** C++ schemes, triggered for every new **reduction**: sum, min, etc.

If the formula "F" is simple ($\leqslant$ 100 arithmetic operations):

"100k $\times$ 100k" computation $\rightarrow$ 10ms – 100ms,

"1M $\times$ 1M" computation $\rightarrow$ 1s – 10s.

Hardware ceiling of 10^{12} operations/s.

$\times$**10 to** $\times$**100 speed-up** vs standard GPU implementations for a wide range of problems.

A long-term investment in the foundations of our field

Since 2016, I've been working on speeding up:

- Geometric **machine learning**: K-Nearest Neighbors, kernel methods.
- Geometric **statistics**: Gaussian processes, Maximum Mean Discrepancies.
- Geometric **deep learning**: point convolutions, attention layers.
- **Survival** analysis: CoxPH solvers, time-varying features.
- **Optimal transport**: our focus today!

Today's talk

1. My **motivations** to study discrete optimal transport.
2. **Computational** advances.
3. How do people use OT **today**?
4. **Open** problems.

Optimal transport?

Optimal transport (OT) generalizes sorting to spaces of dimension $D > 1$

If $A = (x_1, \dots, x_N)$ and $B = (y_1, \dots, y_N)$
are two clouds of N points in $\mathbb{R}^D$, we define:

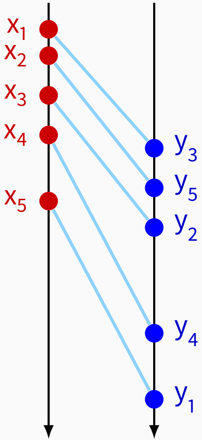
$$\mathrm{OT}(A, B) = \min_{\sigma \in \mathcal{S}_N} \frac{1}{2N} \sum_{i=1}^{N} \| x_i - y_{\sigma(i)} \|^2$$

Generalizes **sorting** to metric spaces.

Linear problem on the permutation matrix P:

$$\mathrm{OT}(A, B) = \min_{P \in \mathbb{R}^{N \times N}} \frac{1}{2N} \sum_{i,j=1}^{N} P_{i,j} \cdot \| x_i - y_j \|^2 ,$$

$$\text{s.t.} \quad P_{i,j} \geqslant 0 \qquad \underbrace{\textstyle\sum_j P_{i,j} = 1}_{\text{Each source point...}} \qquad \underbrace{\textstyle\sum_i P_{i,j} = 1}_{\text{is transported onto the target.}} .$$



assignment

$\sigma : [\![1, 5]\!] \to [\![1, 5]\!]$

Practical use

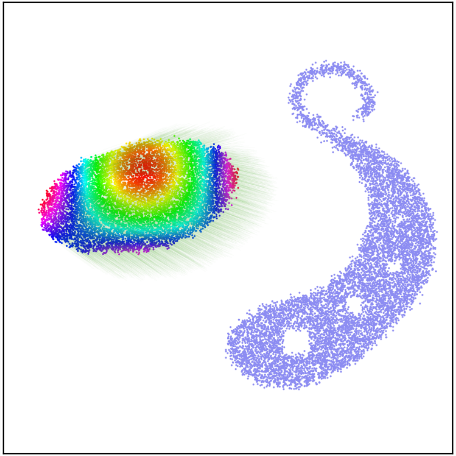
Alternatively, we understand OT as:

- Nearest neighbor **projection** + **incompressibility** constraint.
- Fundamental example of **linear optimization** over the transport plan $\mathrm{P}_{i,j}$.

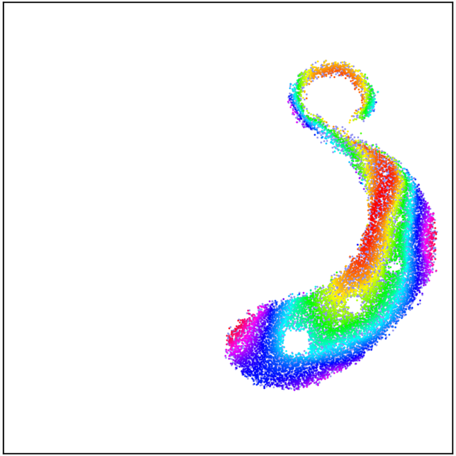
This theory induces two main quantities:

- The transport plan $\mathrm{P}_{i,j} \simeq$ the optimal mapping $x_i \mapsto y_{\sigma(i)}$.
- The "Wasserstein" distance $\sqrt{\mathrm{OT}(\mathrm{A}, \mathrm{B})}$.

The optimal transport plan

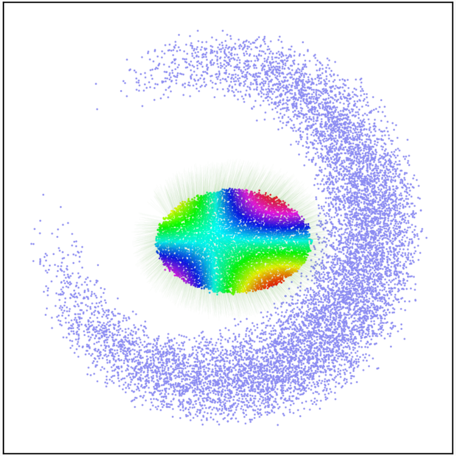


Before

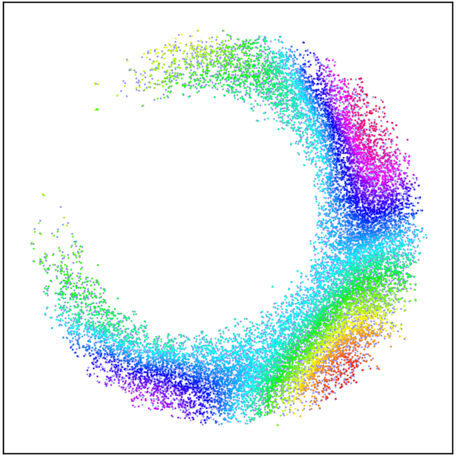


After

The optimal transport plan

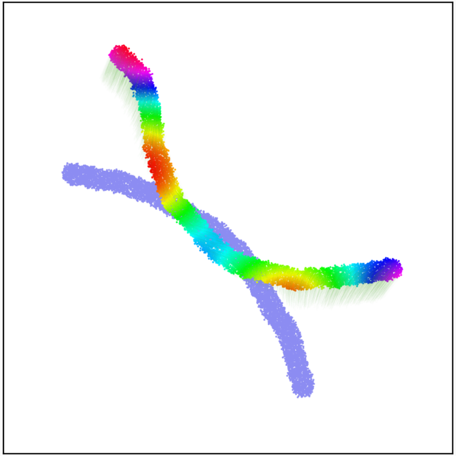


Before



After

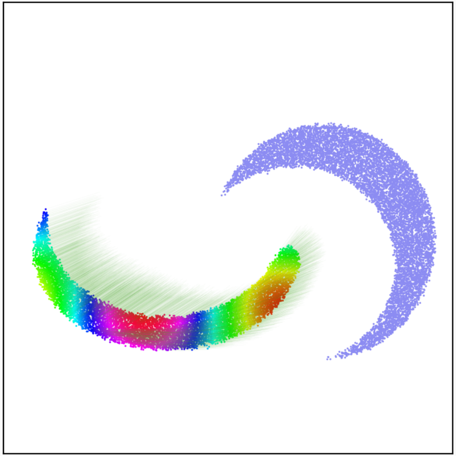
The optimal transport plan



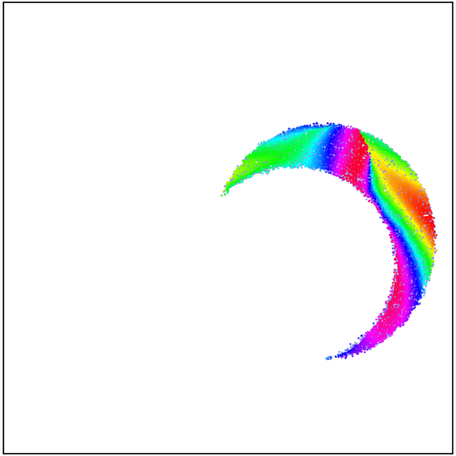
Before

After

The optimal transport plan



Before



After

Key properties of the OT distance

The Wasserstein distance $\sqrt{\text{OT}}(A, B)$ is:

- **Symmetric**: $$\text{OT}(A, B) = \text{OT}(B, A)\,.$$
- **Positive**: $$\text{OT}(A, B) \geqslant 0\,.$$
- **Definite**: $$\text{OT}(A, B) = 0 \iff A = B\,.$$
- **Translation-aware**: $\text{OT}(A,\ \text{Translate}_{\vec{v}}(A)\,) = \frac{1}{2}\|\,\vec{v}\,\|^2\,.$
- More generally, OT retrieves the unique **gradient of a convex function** $T = \nabla\phi$ that maps A onto B:

In dimension 1, $$(x_i - x_j) \cdot (y_{\sigma(i)} - y_{\sigma(j)}) \geqslant 0$$

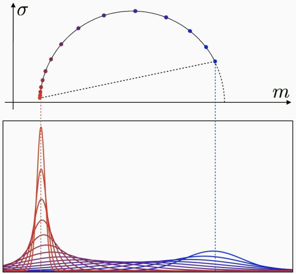
In dimension D, $$\langle\, x_i - x_j\ ,\ T(x_i) - T(x_j)\,\rangle_{\mathbb{R}^D} \geqslant 0\,.$$

$\Longrightarrow$ Appealing generalization of an **increasing mapping**.

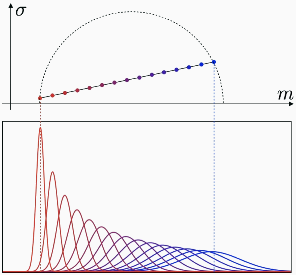
OT induces a geometry-aware distance between probability distributions [PC18]

Gauss map $\quad \mathcal{N} : (m, \sigma) \in \mathbb{R} \times \mathbb{R}_{\geqslant 0} \quad \mapsto \quad \mathcal{N}(m, \sigma) \in \mathbb{P}(\mathbb{R}).$

If the space of **probability distributions** $\mathbb{P}(\mathbb{R})$ is endowed with a given metric, what is the “pull-back” geometry on the space of **parameters** (m, σ)?



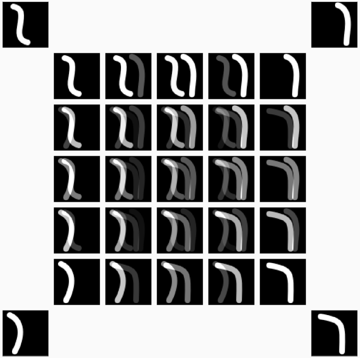
Fisher-Rao ($\simeq$ relative entropy) on $\mathcal{N}(m, \sigma)$
$\rightarrow$ Hyperbolic **Poincaré** metric on (m, σ).



OT on $\mathcal{N}(m, \sigma)$
$\rightarrow$ Flat **Euclidean** metric on (m, σ).

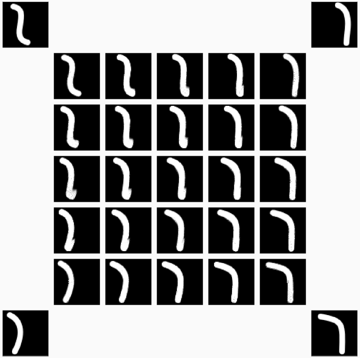
Geometric solutions to least square problems [AC11]

$$\text{Barycenter } \mathsf{A}^* = \arg\min_{\mathsf{A}} \sum_{i=1}^{4} \lambda_i \, \mathsf{Loss}(\mathsf{A}, \mathsf{B}_i) \, .$$



Euclidean barycenters.

$$\mathsf{Loss}(\mathsf{A}, \mathsf{B}) = \|\mathsf{A} - \mathsf{B}\|^2_{L^2}$$



Wasserstein barycenters.

$$\mathsf{Loss}(\mathsf{A}, \mathsf{B}) = \mathsf{OT}(\mathsf{A}, \mathsf{B})$$

How should we solve the OT problem?

Flash-back: the primal OT problem

If $A = (x_1, \dots, x_N)$ and $B = (y_1, \dots, y_N)$
are two clouds of N points in $\mathbb{R}^D$, we define:

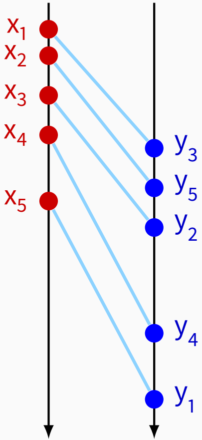
$$\mathrm{OT}(A, B) = \min_{\sigma \in \mathcal{S}_N} \frac{1}{2N} \sum_{i=1}^{N} \| x_i - y_{\sigma(i)} \|^2$$

Generalizes **sorting** to metric spaces.

Linear problem on the permutation matrix P:

$$\mathrm{OT}(A, B) = \min_{P \in \mathbb{R}^{N \times N}} \frac{1}{2N} \sum_{i,j=1}^{N} P_{i,j} \cdot \| x_i - y_j \|^2 ,$$

$$\text{s.t.} \quad P_{i,j} \geqslant 0 \qquad \underbrace{\textstyle\sum_j P_{i,j} = 1}_{\text{Each source point...}} \qquad \underbrace{\textstyle\sum_i P_{i,j} = 1}_{\text{is transported onto the target.}} .$$



assignment
$\sigma : [\![1, 5]\!] \to [\![1, 5]\!]$

A fundamental problem in applied mathematics

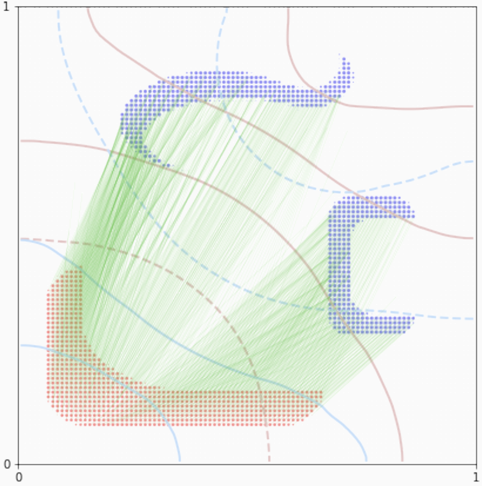
Key dates for discrete optimal transport with N points:

- [Kan42]: **Dual** problem of Kantorovitch.
- [Kuh55]: **Hungarian** methods in $O(\mathrm{N}^3)$.
- [Ber79]: **Auction** algorithm in $O(\mathrm{N}^2)$.
- [KY94]: **SoftAssign** = Sinkhorn + simulated annealing, in $O(\mathrm{N}^2)$.
- [GRL$^+$98, CR00]: **Robust Point Matching** = Sinkhorn as a loss.
- [Cut13]: Start of the **GPU era.**
- [Mér11, Lév15, Sch19]: **multi-scale** solvers in $O(\mathrm{N} \log \mathrm{N})$.

- **Solution,** today: **Multiscale Sinkhorn algorithm, on the GPU**.

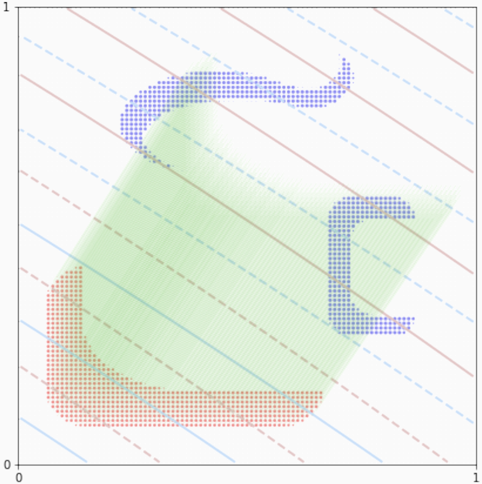
$\Longrightarrow$ Generalized **QuickSort** algorithm.

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\textbf{OT}(\alpha, \beta)$



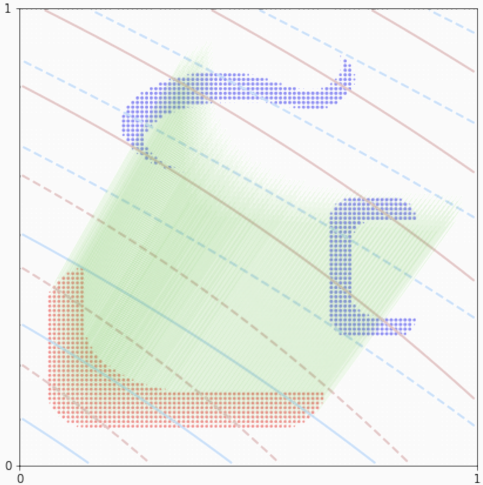
OT plan in 2D.

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



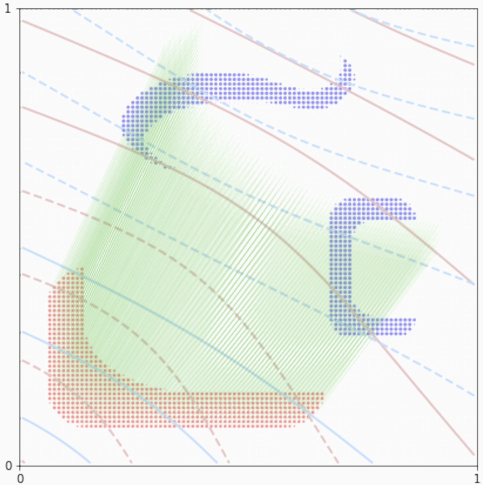
Iteration 0, blur $\sigma = 2^0$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



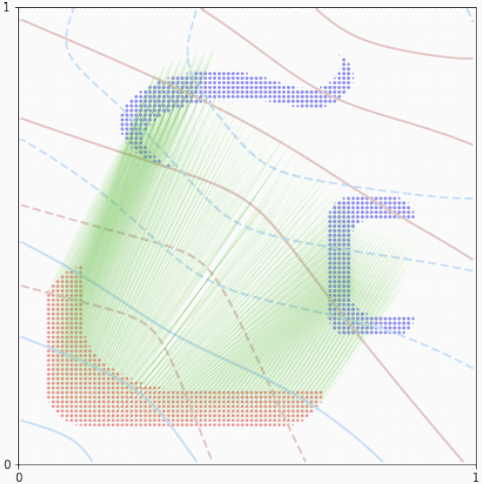
Iteration 1, blur $\sigma = 2^{-1}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}$ OT(α, β)



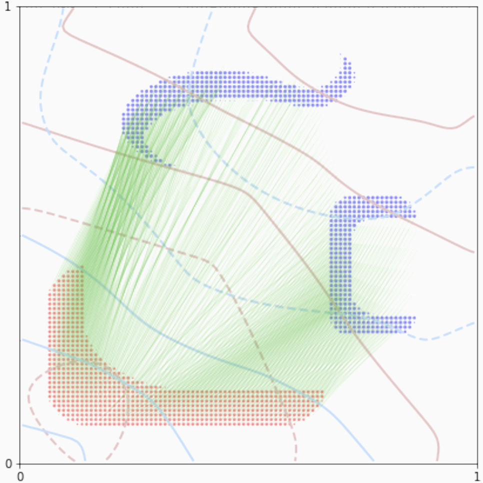
Iteration 2, blur $\sigma = 2^{-2}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha,\beta)$



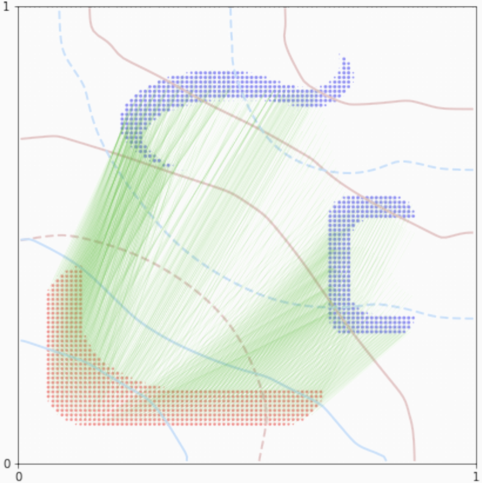
Iteration 3, blur $\sigma = 2^{-3}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



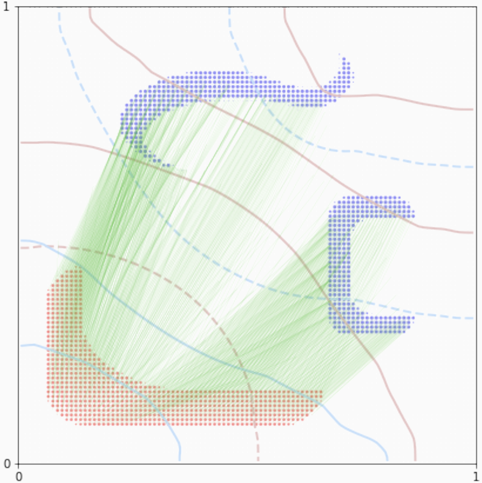
Iteration 4, blur $\sigma = 2^{-4}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}$ OT(α, β)



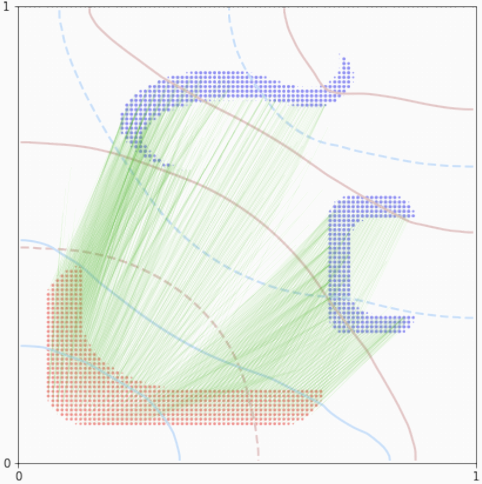
Iteration 5, blur $\sigma = 2^{-5}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



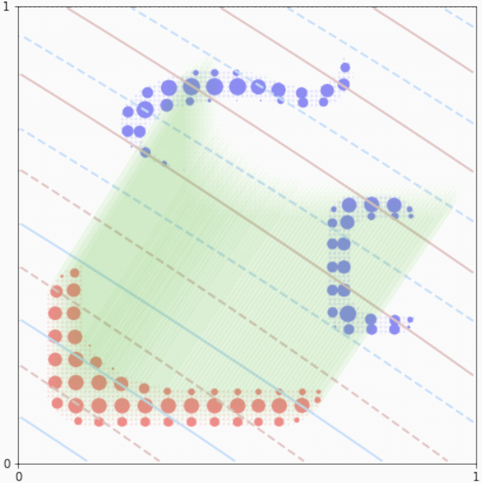
Iteration 6, blur $\sigma = 2^{-6}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathbf{OT}(\alpha, \beta)$



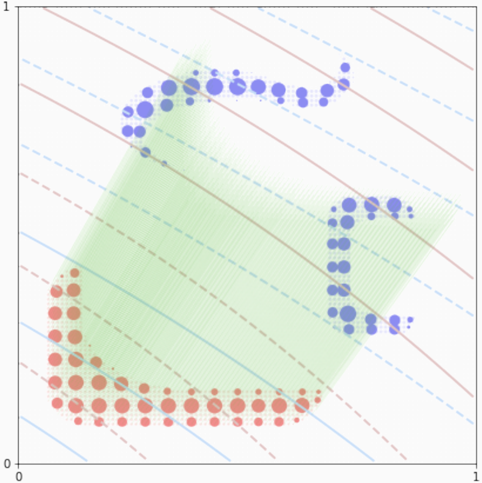
Iteration 7, blur σ = `.01`

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathbf{OT}(\alpha, \beta)$



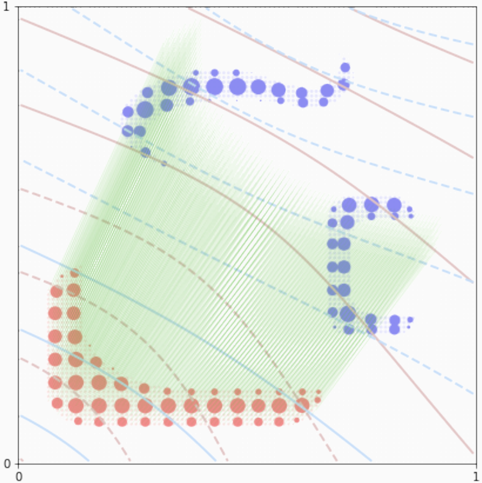
Iteration 0, blur $\sigma = 2^0$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



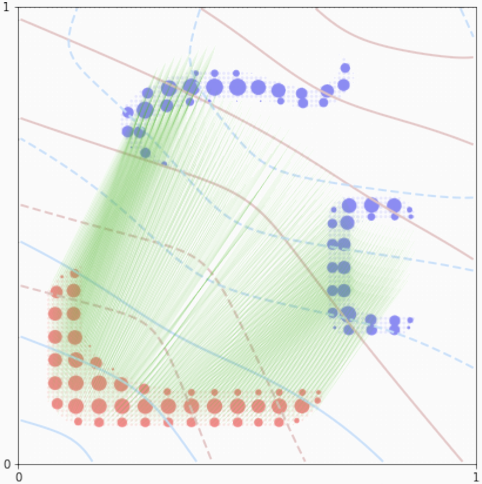
Iteration 1, blur $\sigma = 2^{-1}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha,\beta)$



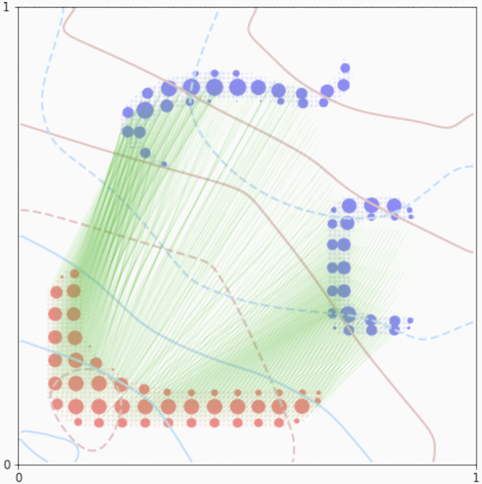
Iteration 2, blur $\sigma = 2^{-2}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



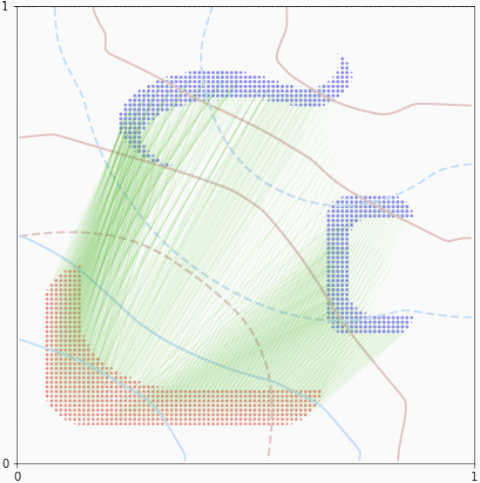
Iteration 3, blur $\sigma = 2^{-3}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathbf{OT}(\alpha, \beta)$



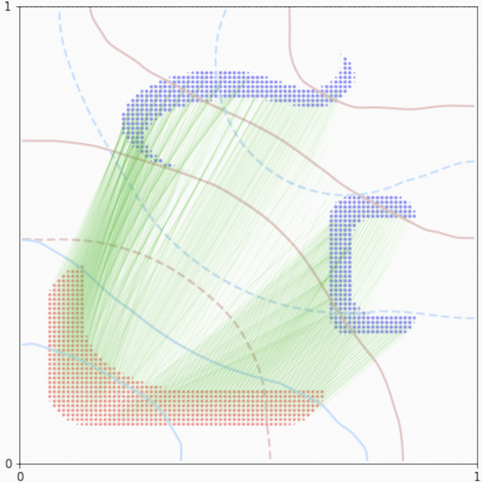
Iteration 4, blur $\sigma = 2^{-4}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathbf{OT}(\alpha, \beta)$



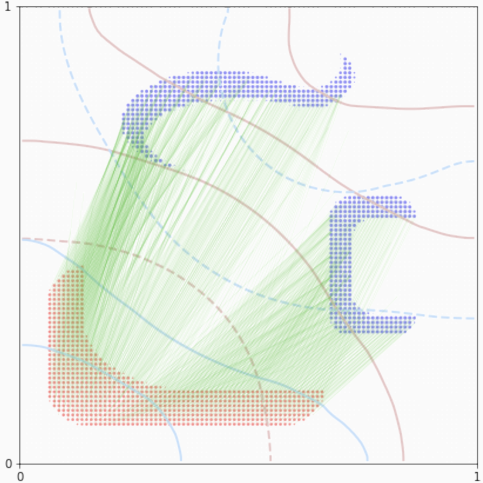
Iteration 5, blur $\sigma = 2^{-5}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}$ OT(α, β)



Iteration 6, blur $\sigma = 2^{-6}$

Visualizing F, G and the Brenier map $\nabla F(x_i) = -\frac{1}{\alpha_i}\partial_{x_i}\mathrm{OT}(\alpha, \beta)$



Iteration 7, blur σ = `.01`

Scaling up optimal transport to anatomical data

Progresses of the last decade add up to a $\times$**100** - $\times$**1000** acceleration:

$$\text{Sinkhorn GPU} \xrightarrow{\times 10} + \text{KeOps} \xrightarrow{\times 10} + \text{Annealing} \xrightarrow{\times 10} + \text{Multi-scale}$$

With a precision of 1%, on a modern gaming GPU:

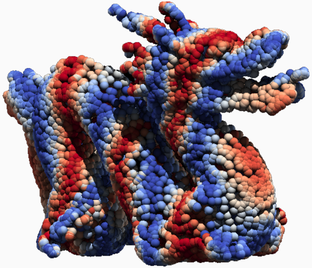
`pip install geomloss`

+

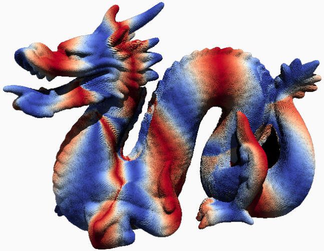
modern GPU

(1 000 €)

$\Longrightarrow$



10k points in 30-50ms



100k points in 100-200ms

How do people use OT in 2022?

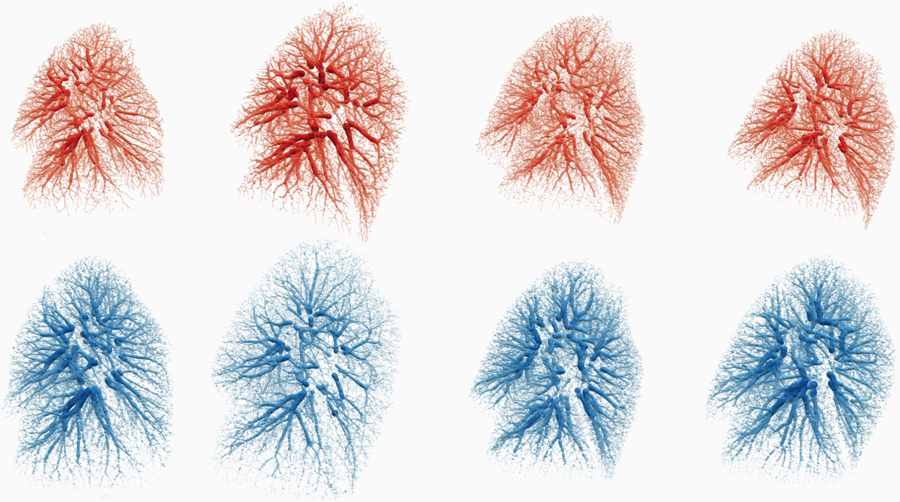
1. Physics and simulation of Partial Differential Equations

Since the 1990s, OT is an essential tool to deal with flows:

- Fundamental models have an **appealing form** when seen through the OT lense:
  the incompressible **Euler flow** is a **geodesic** trajectory,
  heat diffusion is a gradient **descent**…
- This framework allows mathematicians to design and study new models **effectively**.
- **Implementations** in 2D and 3D are now becoming mature.
- Lots of cool simulations of **crowds**, **water** or the **early universe**!

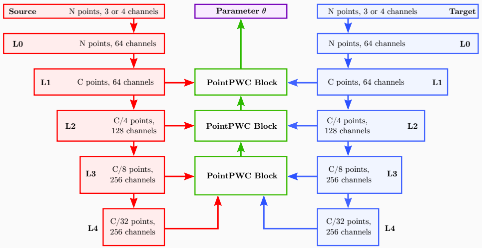
Pointers: MoKaPlan Inria team, Bruno Lévy, Quentin Mérigot,
Filippo Santambrogio, Yann Brenier, Felix Otto…

2. A typical example in shape analysis: lung registration “Exhale – Inhale”



Complex deformations, high **resolution** (50k–300k points), high **accuracy** (< 1mm).

State-of-the-art networks – and their limitations



Multi-scale convolutional point neural network.

Point neural nets, **in practice**:

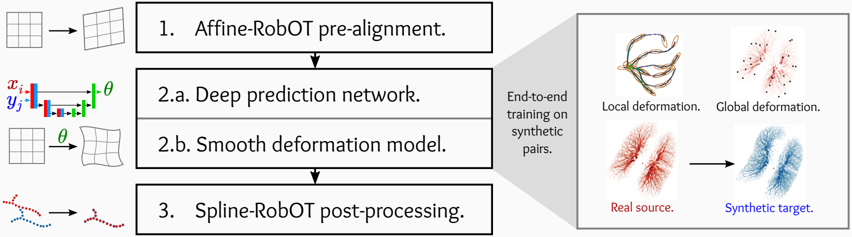
- Compute **descriptors** at all scales.
- **Match** them using geometric layers.
- Train on **synthetic** deformations.

Strengths and weaknesses:

- Good at **pairing** branches.
- Hard to train to high **accuracy**.

$\Longrightarrow$ **Complementary** to OT.

Three-steps registration



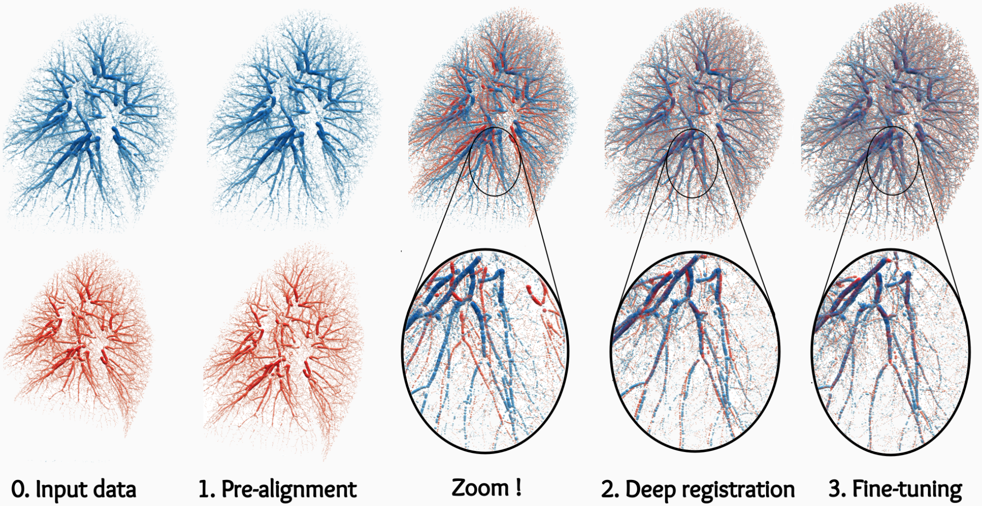
This **pragmatic** method:

- Is **easy to train** on synthetic data.
- Scales up to high-resolution: 100k points in 1s.
- Excellent results: **KITTI** (outdoors scans) and **DirLab** (lungs).

__Accurate__ point cloud registration with __robust__ optimal transport,

Shen, Feydy et al., NeurIPS 2021.

Three-steps registration



3. An intriguing tool in machine learning

OT **lifts to probability distributions** the geometry of the sample space $\|x_i - y_j\|$.

This is relevant at the intersection between geometry and statistics in order to:

- Design **2-sample tests** : do these two samples come from the same distribution?
- Quantify the **discrepancy** between a synthetic sample and the data distribution.
- Study the convergence of **particle-based optimization** schemes, from simple neural networks to MCMC samplers.

Pointers: Python Optimal Transport (Flamary, Courty et al.),
Computational Optimal Transport (Peyré and Cuturi),
Jonathan Weed, Justin Solomon, Philippe Rigollet, Lenaïc Chizat, Anna Korba…

Open problems

1. Learning in the space of probability distributions

Can we generalize standard ML algorithms for:

- population visualization
- regression
- classification

from **vector spaces** to a (non-linear) space of **probability** distributions?

Thanks to **fast and reliable solvers** for the Wasserstein **barycenter** problem,
this now seems realistic in dimensions 2 and 3,
with applications to PDE solvers and shape analysis.

2. Going beyond the (squared) Euclidean distance

Most results and heuristics only hold for simple cost functions ($\|x_i - y_j\|$, $\|x_i - y_j\|^2$, etc.):

- What about **concave** costs, e.g. $\sqrt{\|x_i - y_j\|}$?
- What about distances that cannot be written in closed form, e.g. geodesic distances on **graphs**?
- Can we guarantee (some) **smoothness** for the transport map while keeping super-fast solvers?

3. OT as a source of inspiration in high-dimensional scenarios

Standard OT is hardly relevant when dealing with **high-dimensional** data samples (collections of images, text documents, electronic health records…).

This is a direct consequence of the **curse of dimensionality**:
OT cannot extract information out of a meaningless matrix of distances $\|x_i - y_j\|$.

However, we can still **build upon** the geometric ideas of OT theory to design interesting, domain-specific distances **between distributions**.

This is the key idea behind "Wasserstein" GANs, metric learning…
Can we build other **fruitful analogies**?

My job: create tools for a new generation of researchers

1. **Secure** a permanent position.
   $\rightarrow$ Inria researcher since Dec. 2021.

2. Shore up the **GPU foundations** of the field.
   $\rightarrow$ KeOps v2.0 released in March 2022, now seamless to install.

3. **Re-write GeomLoss** with a better interface and full support for 2D/3D images.
   $\rightarrow$ WIP with the Python Optimal Transport devs.

4. Maintain an **open benchmarking platform** for the community,
   following the example of www.ann-benchmarks.com for nearest neighbor search.
   $\rightarrow$ WIP.

Conclusion

Genuine team work


Benjamin Charlier


Joan Glaunès


Thibault Séjourné


F.-X. Vialard


Gabriel Peyré


Alain Trouvé


Marc Niethammer


Shen Zhengyang


Olga Mula


Hieu Do

Key points

- Optimal Transport = **generalized sorting** :

  $\longrightarrow$ Super-fast solvers on simple domains (esp. 2D/3D spaces).

  $\longrightarrow$ Simple registration for shapes that are close to each other.

  $\longrightarrow$ Fundamental tool at the intersection of geometry and statistics.

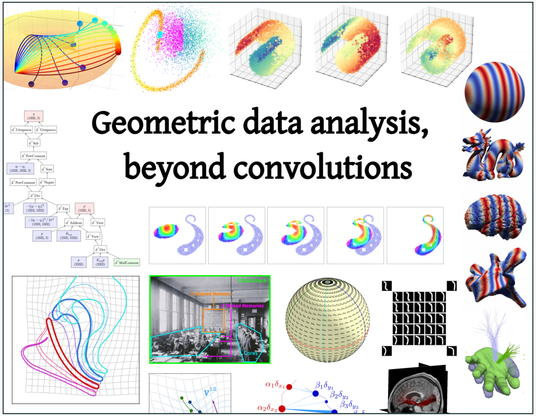
  $\longrightarrow$ Open geometric questions with a genuine application.

- GPUs are more **versatile** than you think.

  $\longrightarrow$ Ongoing work to provide **fast GPU backends** to researchers, going beyond what Google and Facebook are ready to pay for.

Documentation and tutorials are available online

⟹ www.kernel-operations.io ⟸



www.jeanfeydy.com/geometric_data_analysis.pdf

References

References i

M. Agueh and G. Carlier.

Barycenters in the Wasserstein space.

SIAM Journal on Mathematical Analysis, 43(2):904–924, 2011.

Dimitri P Bertsekas.

A distributed algorithm for the assignment problem.

Lab. for Information and Decision Systems Working Paper, M.I.T., Cambridge, MA, 1979.

Haili Chui and Anand Rangarajan.

A new algorithm for non-rigid point matching.

In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 44–51. IEEE, 2000.

Marco Cuturi.

Sinkhorn distances: Lightspeed computation of optimal transport.

In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness.

New algorithms for 2d and 3d point matching: Pose estimation and correspondence.

Pattern recognition, 31(8):1019–1031, 1998.

Leonid V Kantorovich.

On the translocation of masses.

In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942.

Harold W Kuhn.

The Hungarian method for the assignment problem.

Naval research logistics quarterly, 2(1-2):83–97, 1955.

Jeffrey J Kosowsky and Alan L Yuille.

The invisible hand algorithm: Solving the assignment problem with statistical physics.

Neural networks, 7(3):477–490, 1994.

References v

Bruno Lévy.

A numerical algorithm for l2 semi-discrete optimal transport in 3d.

ESAIM: Mathematical Modelling and Numerical Analysis, 49(6):1693–1715, 2015.

Quentin Mérigot.

A multiscale approach to optimal transport.

In *Computer Graphics Forum*, volume 30, pages 1583–1592. Wiley Online Library, 2011.

Gabriel Peyré and Marco Cuturi.

Computational optimal transport.

arXiv preprint arXiv:1803.00567, 2018.

Bernhard Schmitzer.

Stabilized sparse scaling algorithms for entropy regularized transport problems.

SIAM Journal on Scientific Computing, 41(3):A1443–A1481, 2019.