Optimal transport: mature tools and open problems

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

25th of July, 2022
Frontiers in Design Representation Summer School
University of Maryland
Who am I?

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

- **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.
- $512 \times 512 \times 512 \approx 130M$ 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.
**Target.** Scale up models that combine medical expertise with modern datasets.

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

- Incredible **value for money**:
  
  $1000\,\€ \approx 1000$ cores $\approx 10^{12}$ operations/s.

- **Bottleneck**: constraints on register usage.

“User-friendly” Python ecosystem, consolidated around a **small number of key operations**.
The KeOps library: efficient support for symbolic matrices

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 (≤ 100 arithmetic operations):
- “100k × 100k” computation → 10ms – 100ms,
- “1M × 1M” computation → 1s – 10s.

Hardware ceiling of \(10^{12}\) operations/s.
×10 to ×100 speed-up vs standard GPU implementations for a wide range of problems.
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!
1. What is Optimal Transport, and why does it matter?

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

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

Generalizes **sorting** to metric spaces.

**Linear problem** on the permutation matrix $P$:

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

s.t. $P_{i,j} \geq 0$ \quad $\sum_j P_{i,j} = 1$ \quad $\sum_i P_{i,j} = 1$.

Each source point is transported onto the target.

$\sigma : [1, 5] \to [1, 5]$
Alternatively, we understand OT as:

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

This theory induces two main quantities:

- The transport plan $P_{i,j} \approx$ the optimal mapping $x_i \mapsto y_{\sigma(i)}$.
- The “Wasserstein” distance $\sqrt{\text{OT}(A, B)}$. 
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 $\sqrt{\mathcal{O}T}(A, B)$ is:

- **Symmetric**: $\mathcal{O}T(A, B) = \mathcal{O}T(B, A)$.
- **Positive**: $\mathcal{O}T(A, B) \geq 0$.
- **Definite**: $\mathcal{O}T(A, B) = 0 \iff A = B$.
- **Translation-aware**: $\mathcal{O}T(A, \text{Translate}_{\vec{v}}(A)) = \frac{1}{2}\|\vec{v}\|^2$.

More generally, OT retrieves the unique gradient of a convex function $T = \nabla \phi$ that maps $A$ onto $B$:

- In dimension 1, $(x_i - x_j) \cdot (y_{\sigma(i)} - y_{\sigma(j)}) \geq 0$.
- In dimension $D$, $\langle x_i - x_j, T(x_i) - T(x_j) \rangle_{\mathbb{R}^D} \geq 0$.

$\implies$ Appealing generalization of an increasing mapping.
OT induces a geometry-aware distance between probability distributions

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

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** \((m, \sigma)\)?

- **Fisher-Rao** (\( \cong \) relative entropy) on \( \mathcal{N}(m, \sigma) \)
  \( \Rightarrow \) Hyperbolic Poincaré metric on \((m, \sigma)\).

- **OT** on \( \mathcal{N}(m, \sigma) \)
  \( \Rightarrow \) Flat Euclidean metric on \((m, \sigma)\).
Geometric solutions to least square problems \[\text{[AC11]}\]

Barycenter \( A^* = \arg\min_A \sum_{i=1}^{4} \lambda_i \text{Loss}(A, B_i) \).

**Euclidean** barycenters.

\[ \text{Loss}(A, B) = \| A - B \|_{L^2}^2 \]

**Wasserstein** barycenters.

\[ \text{Loss}(A, B) = \text{OT}(A, B) \]
How should we solve the OT problem?
Flash-back: the primal OT problem

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

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

Generalizes sorting to metric spaces.

**Linear problem** on the permutation matrix $P$:

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

s.t. $P_{i,j} \geq 0$  \[\sum_j P_{i,j} = 1\]  \[\sum_i P_{i,j} = 1.\]

Each source point… 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 $O(N^3)$.
- [Ber79]: **Auction** algorithm in $O(N^2)$.
- [KY94]: **SoftAssign** = Sinkhorn + simulated annealing, in $O(N^2)$.
- [GRL98, CR00]: **Robust Point Matching** = Sinkhorn as a loss.
- [Cut13]: Start of the **GPU era**.
- [Mér11, Lév15, Sch19]: **multi-scale** solvers in $O(N \log N)$.

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

  $\implies$ Generalized **QuickSort** algorithm.
Scaling up optimal transport to anatomical data

Progresses of the last decade add up to a $\times 100 - \times 1,000$ acceleration:

Sinkhorn GPU $\times 10$ + KeOps $\times 10$ + Annealing $\times 10$ + Multi-scale

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

```
pip install geomloss
+ modern GPU (1 000 €)
```

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 has been 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

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**.

**Multi-scale** convolutional point neural network.

⟹ **Complementary** to OT.
Three-steps registration

1. Affine-RobOT pre-alignment.
2.a. Deep prediction network.
2.b. Smooth deformation model.

End-to-end training on synthetic pairs.

Real source. Synthetic target.

Local deformation. Global deformation.

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,**
Three-steps registration

0. Input data  
1. Pre-alignment  
Zoom !  
2. Deep registration  
3. Fine-tuning
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
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.
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: pave the way for a new generation of researchers

1. **Secure** a permanent position.

2. Shore up the **GPU foundations** of the field.
   → 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.
   → WIP with the Python Optimal Transport devs, first release after the Summer.

4. Maintain an **open benchmarking platform** for the community,
   following the example of www.ann-benchmarks.com for nearest neighbor search.
   → WIP, release this Fall.
Conclusion
Key points

• **Symbolic** matrices are to **geometric** ML what **sparse** matrices are to **graph** processing:
  ➔ KeOps: **x30 speed-up** vs. PyTorch, TF et JAX.
  ➔ Useful in a wide range of settings.

• Optimal Transport = **generalized sorting** = **incompressibility** prior:
  ➔ Super-fast solvers on simple domains (especially 2D/3D spaces).
  ➔ Fundamental tool at the intersection of geometry and statistics.

• GPUs are more **versatile** than you think.
  ➔ 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
M. Agueh and G. Carlier.

**Barycenters in the Wasserstein space.**


Dimitri P Bertsekas.

**A distributed algorithm for the assignment problem.**

Haili Chui and Anand Rangarajan.

A new algorithm for non-rigid point matching.


Marco Cuturi.

Sinkhorn distances: Lightspeed computation of optimal transport.

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

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


Leonid V Kantorovich.

On the translocation of masses.

Harold W Kuhn.

The Hungarian method for the assignment problem.


Jeffrey J Kosowsky and Alan L Yuille.

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

Bruno Lévy.

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


Quentin Mérigot.

**A multiscale approach to optimal transport.**

Bernhard Schmitzer.

**Stabilized sparse scaling algorithms for entropy regularized transport problems.**