

# Geometric insights on matching problems

---

**Jean Feydy**

**HeKA team**, INRIA Paris, INSERM, Université Paris-Cité, PR[AI]RIE-PSAI

October 1st, 2026 – **MLCN workshop** @ MICCAI 2026, Strasbourg

# Who am I?

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

**2012–2016** ENS Paris, mathematics.

**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 team at INRIA Paris

# HeKA: a translational research team for public health

Hospitals  
Inria      Inserm  
Universities



# My main motivation

Develop **robust and efficient** software that **stimulates** other researchers:

1. Speed up **geometric machine learning** on GPUs:
  - ⇒ **pyKeOps** library for distance and kernel matrices, 1M+ downloads.
2. Scale up **pharmacovigilance** to the full French population:
  - ⇒ **survivalGPU**, a fast re-implementation of the R survival package.
3. Ease access to modern **shape analysis**:
  - ⇒ **GeomLoss**, scalable optimal transport in Python, 1M+ downloads.
  - ⇒ **scikit-shapes**, available soon.

# Today's talk – assuming that you would enjoy some theory

- **Matching problems?**

A fundamental tool in data sciences.

- **Geometric insights**

Matching  $\iff$  Affine registration in some feature space.

- **Practical consequences**

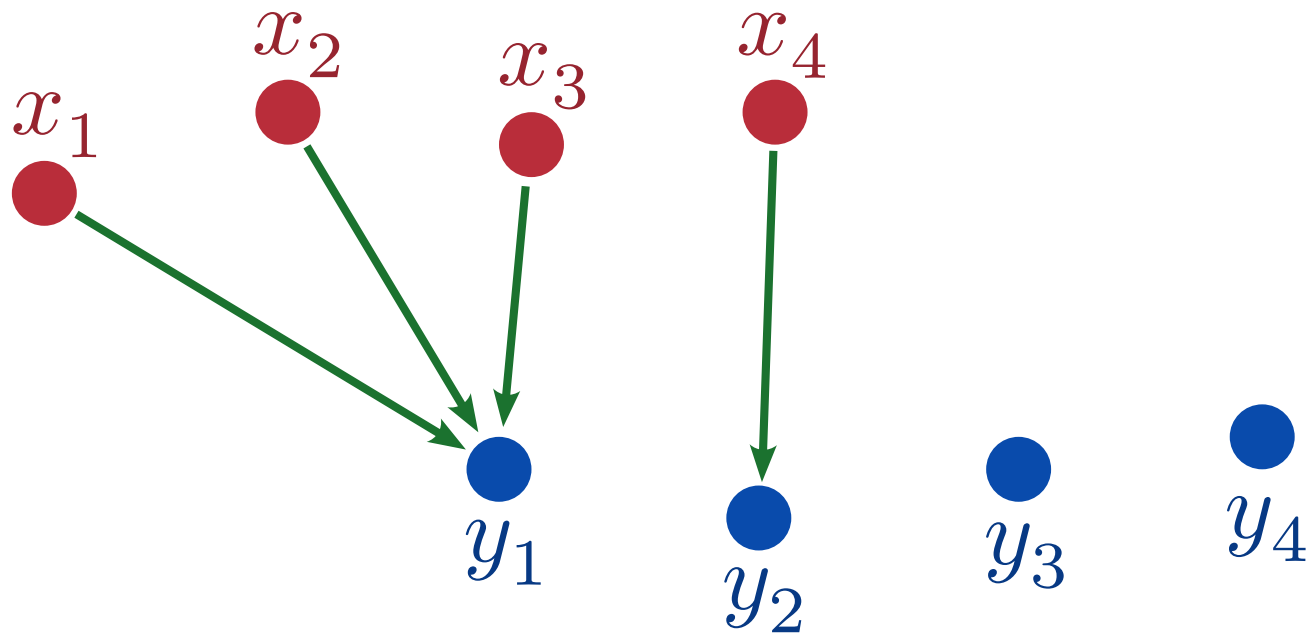
Scalable solvers on CPUs and GPUs.

Relevant for both learning-based and learning-free methods.

**Matching problems?**

---

# Nearest neighbor projection: a first example

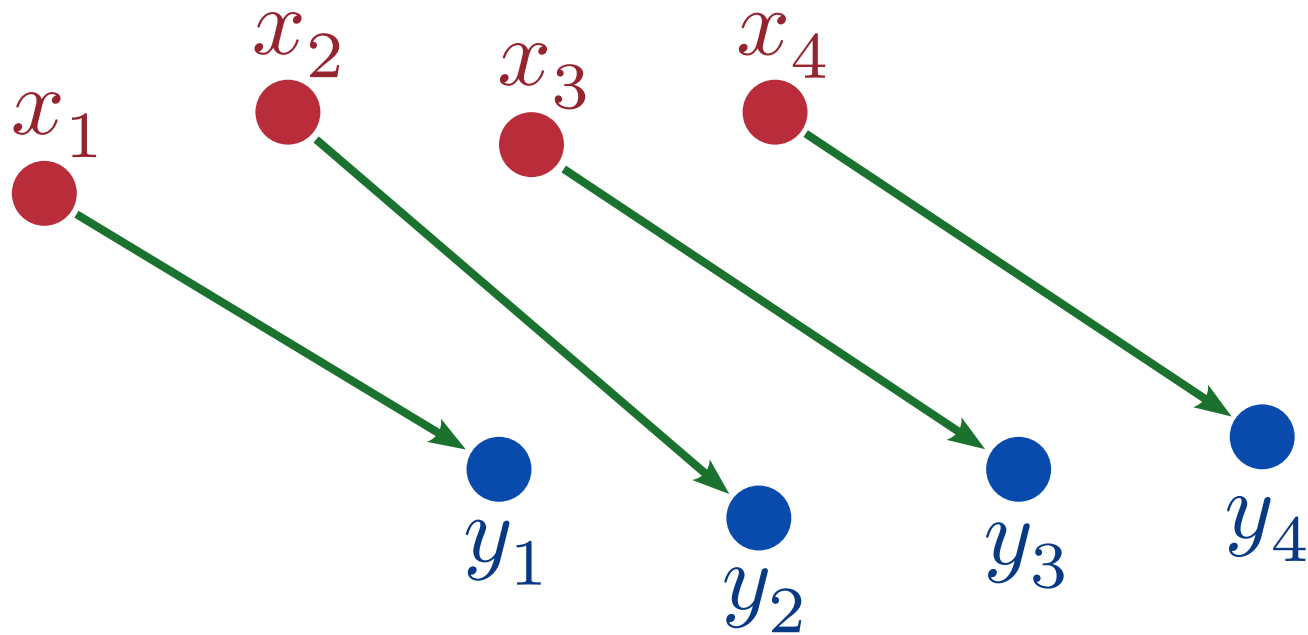


$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 1 | 1 | 2 |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \frac{1}{4} \sum_{i=1}^4 \|x_i - y_{\sigma(i)}\|^2$$

# Nearest neighbor projection + bijectivity = optimal transport

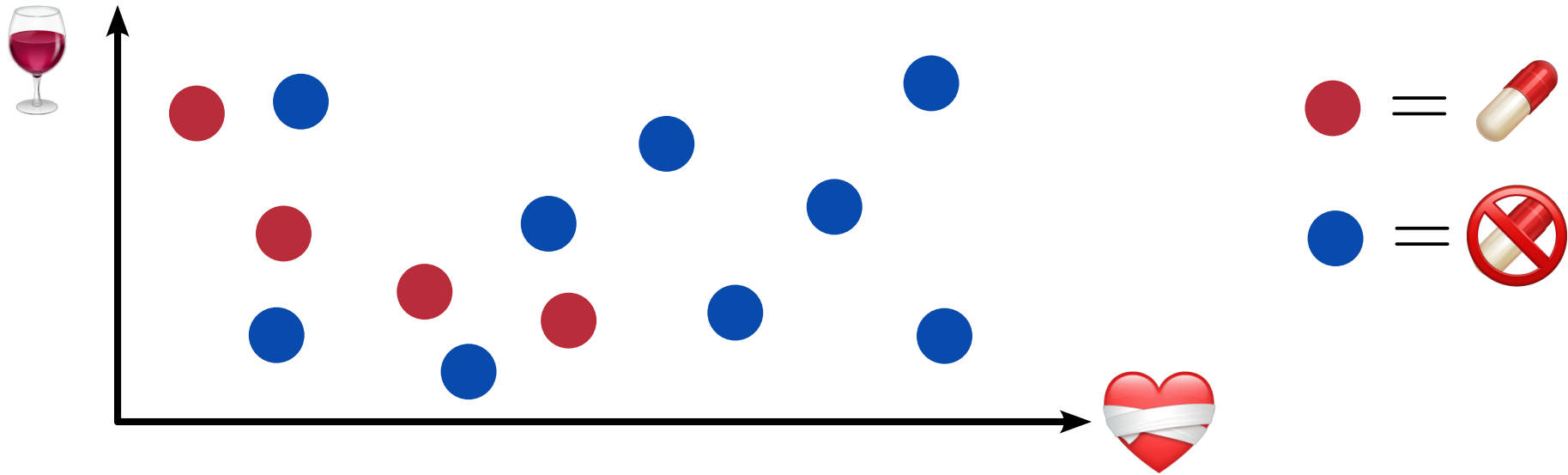


$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |

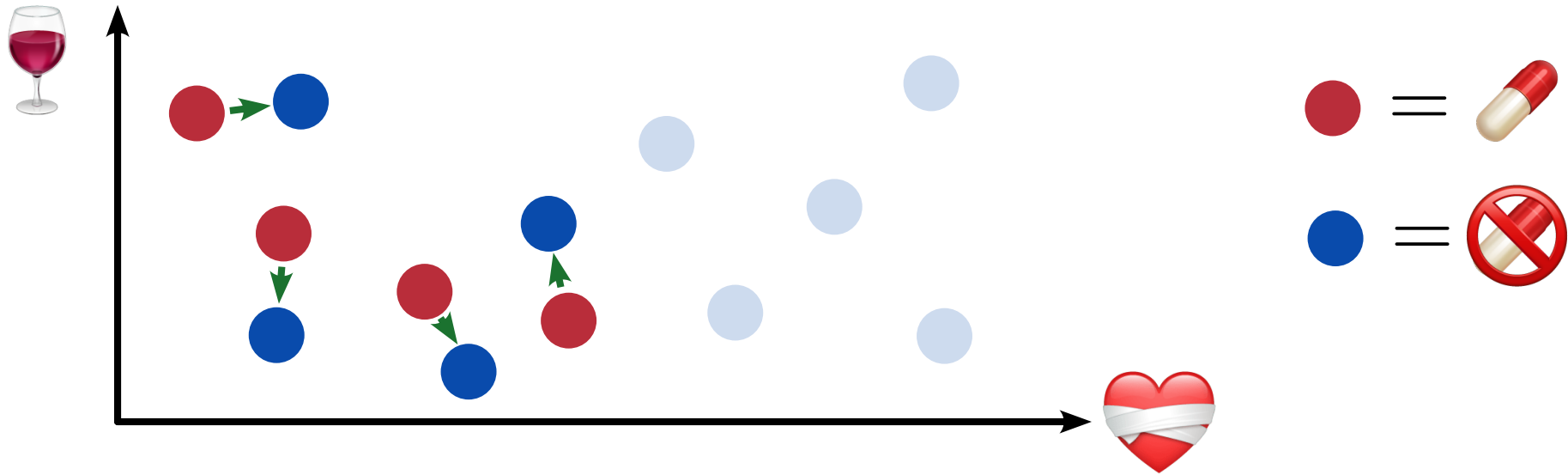
$$\min_{\sigma: [1,4] \rightarrow [1,4]} \frac{1}{4} \sum_{i=1}^4 \|x_i - y_{\sigma(i)}\|^2 + \infty \cdot \mathbb{1}(\sigma \text{ is not bijective})$$

# Typical application: causal inference in public health



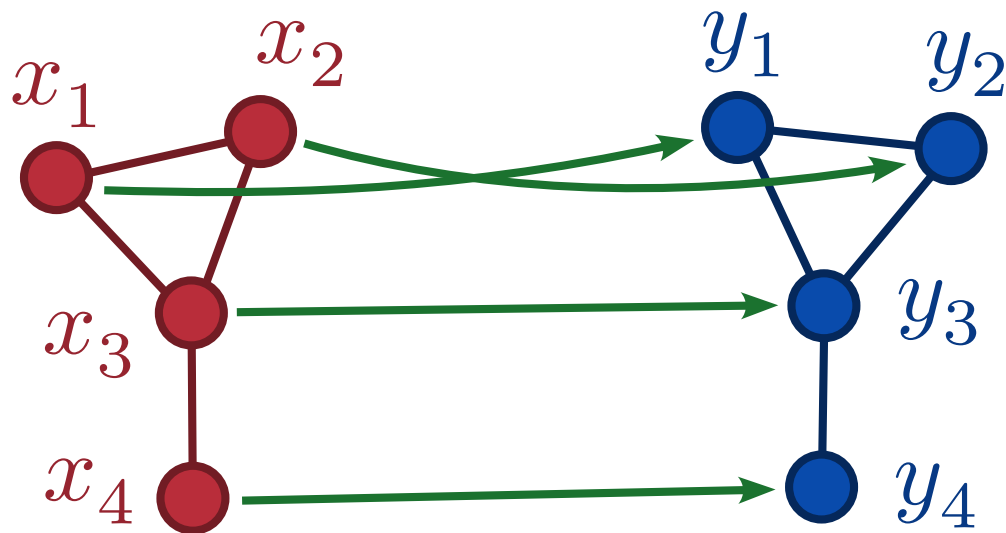
In order to compare a **treated cohort** with the **general population**, we must **adjust for confounders**: prior health condition, lifestyle...

# Typical application: causal inference in public health



In order to compare a **treated cohort** with the **general population**, we must **adjust for confounders**: prior health condition, lifestyle...  
Matching provides a non-parametric baseline.

# Graph matching

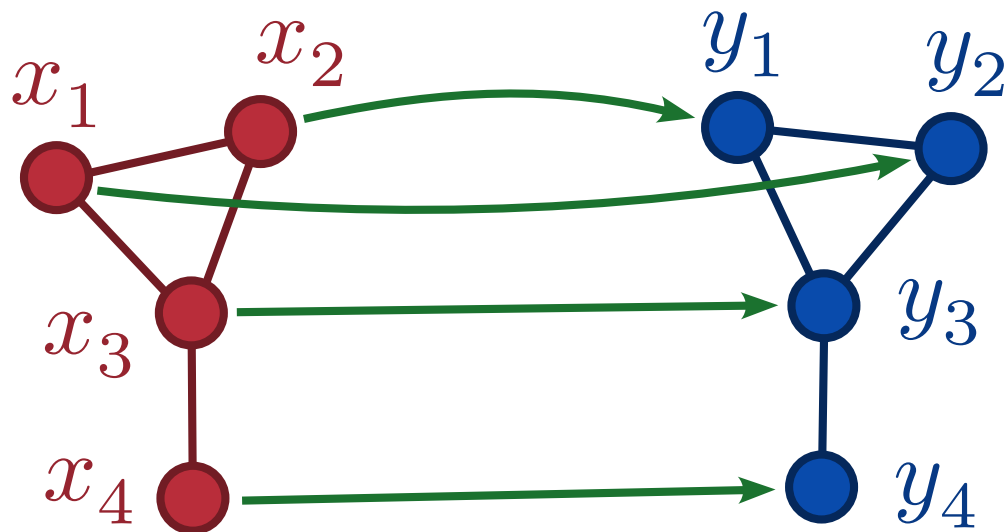


$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \sum_{i,j=1}^4 \underbrace{\| (x_i \leftrightarrow x_j) - (y_{\sigma(i)} \leftrightarrow y_{\sigma(j)}) \|^2}_{\sigma \text{ preserves the graph structure}} + \infty \cdot \mathbb{1}(\sigma \text{ is not bijective})$$

# Graph matching – beware of multiple minima!

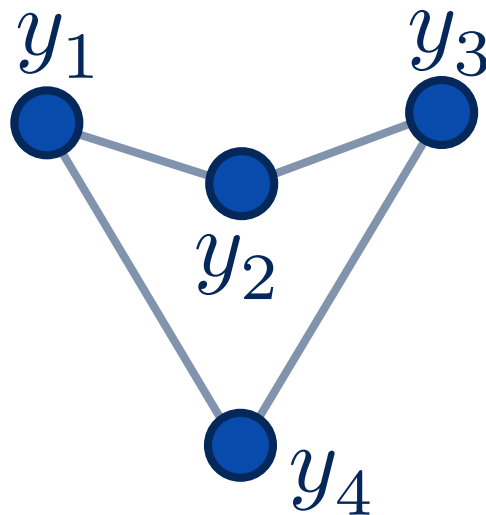
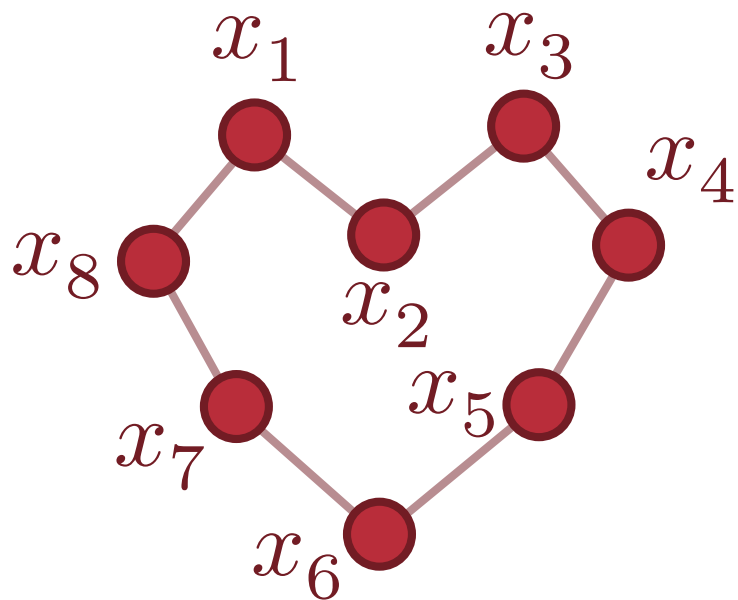


$\sigma \hookleftarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 1 | 3 | 4 |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \underbrace{\sum_{i,j=1}^4 \| (x_i \leftrightarrow x_j) - (y_{\sigma(i)} \leftrightarrow y_{\sigma(j)}) \|^2}_{\sigma \text{ preserves the graph structure}} + \infty \cdot \mathbb{1}(\sigma \text{ is not bijective})$$

# Shape registration



$\sigma \hookrightarrow$

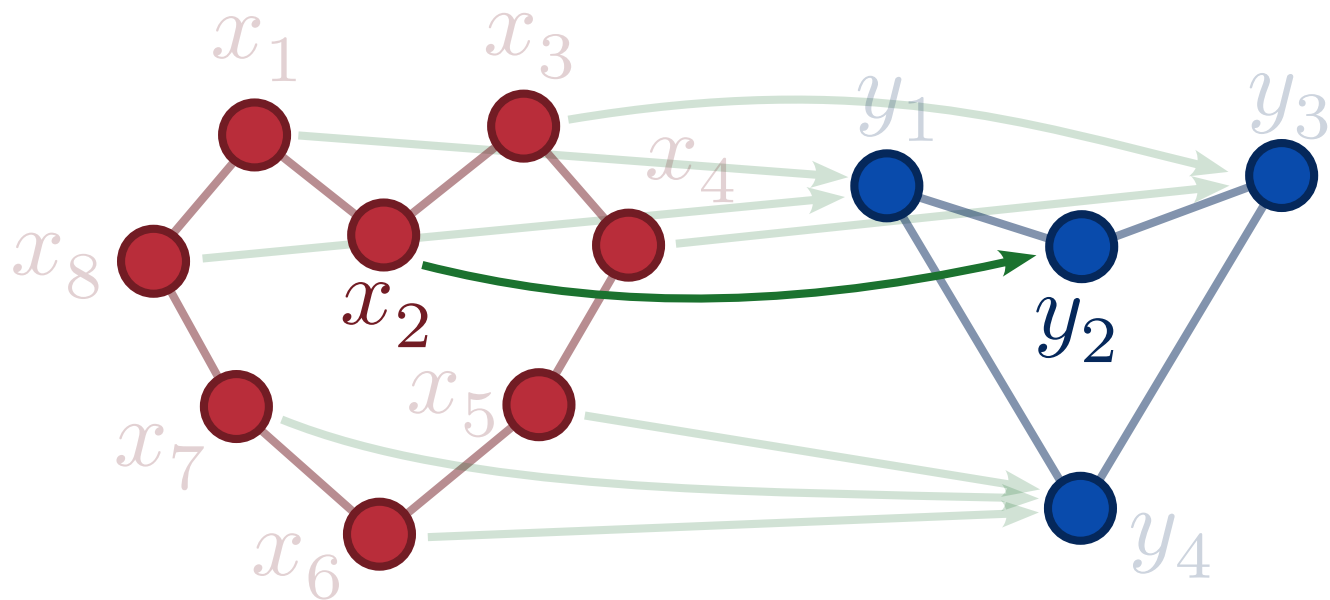
|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
|   |   |   |   |

$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 5 | 6 | 7 | 8 |
|   |   |   |   |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \underbrace{\sum_{x_i \leftrightarrow x_j} \|y_{\sigma(i)} - y_{\sigma(j)}\|^2}_{x_i \rightarrow y_{\sigma(i)} \text{ is a smooth vector field}} + \infty \cdot \mathbb{1}(\sigma \text{ is not surjective})$$

# Shape registration



$\sigma \hookleftarrow$

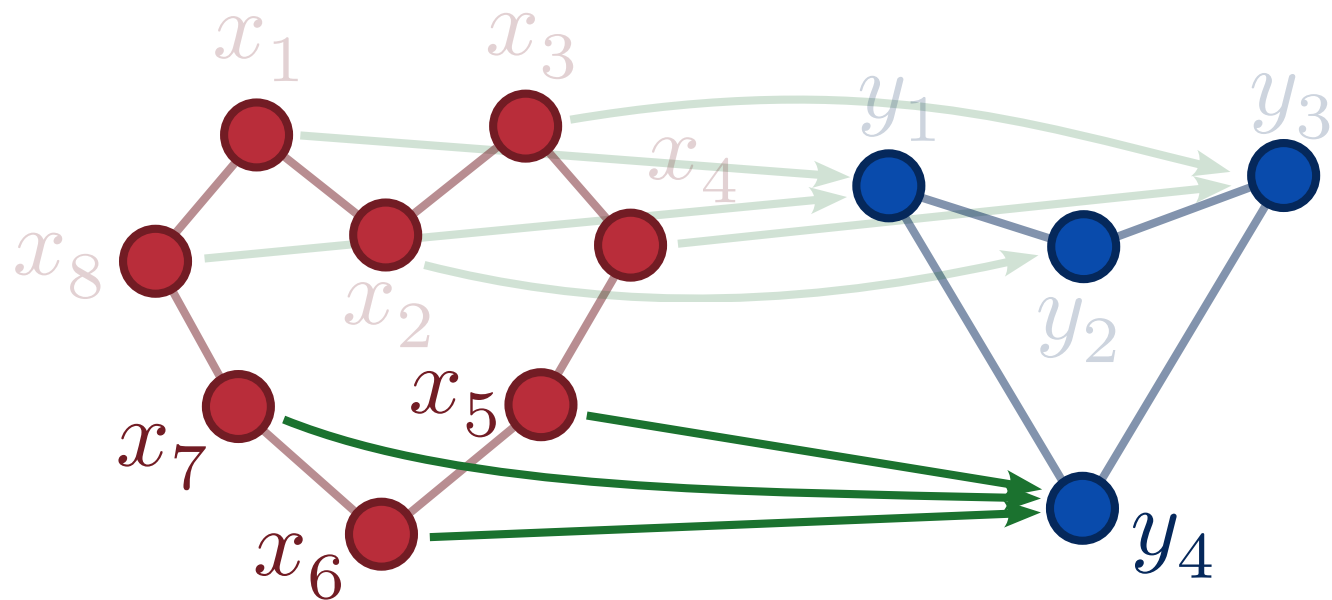
|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 3 |

$\sigma \hookleftarrow$

|   |   |   |   |
|---|---|---|---|
| 5 | 6 | 7 | 8 |
| 4 | 4 | 4 | 1 |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \underbrace{\sum_{x_i \leftrightarrow x_j} \|y_{\sigma(i)} - y_{\sigma(j)}\|^2}_{x_i \rightarrow y_{\sigma(i)} \text{ is a smooth vector field}} + \infty \cdot \mathbb{1}(\sigma \text{ is not surjective})$$

# Shape registration



$\sigma \hookleftarrow$

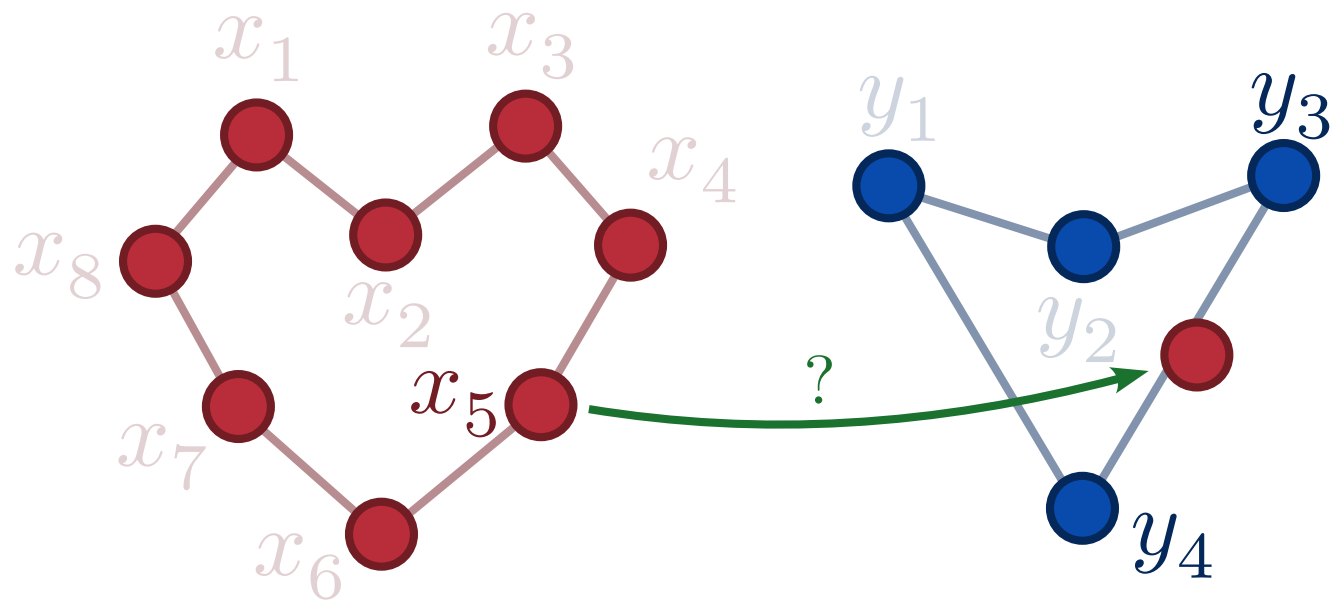
|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 3 |

$\sigma \hookleftarrow$

|   |   |   |   |
|---|---|---|---|
| 5 | 6 | 7 | 8 |
| 4 | 4 | 4 | 1 |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \underbrace{\sum_{x_i \leftrightarrow x_j} \|y_{\sigma(i)} - y_{\sigma(j)}\|^2}_{x_i \rightarrow y_{\sigma(i)} \text{ is a smooth vector field}} + \infty \cdot \mathbb{1}(\sigma \text{ is not surjective})$$

# Shape registration



$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
|   |   |   |   |

$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 5 | 6 | 7 | 8 |
| ? |   |   |   |

$$\min_{\sigma: [1,4] \rightarrow [1,4]} \underbrace{\sum_{x_i \leftrightarrow x_j} \|y_{\sigma(i)} - y_{\sigma(j)}\|^2}_{x_i \rightarrow y_{\sigma(i)} \text{ is a smooth vector field}} + \infty \cdot \mathbb{1}(\sigma \text{ is not surjective})$$

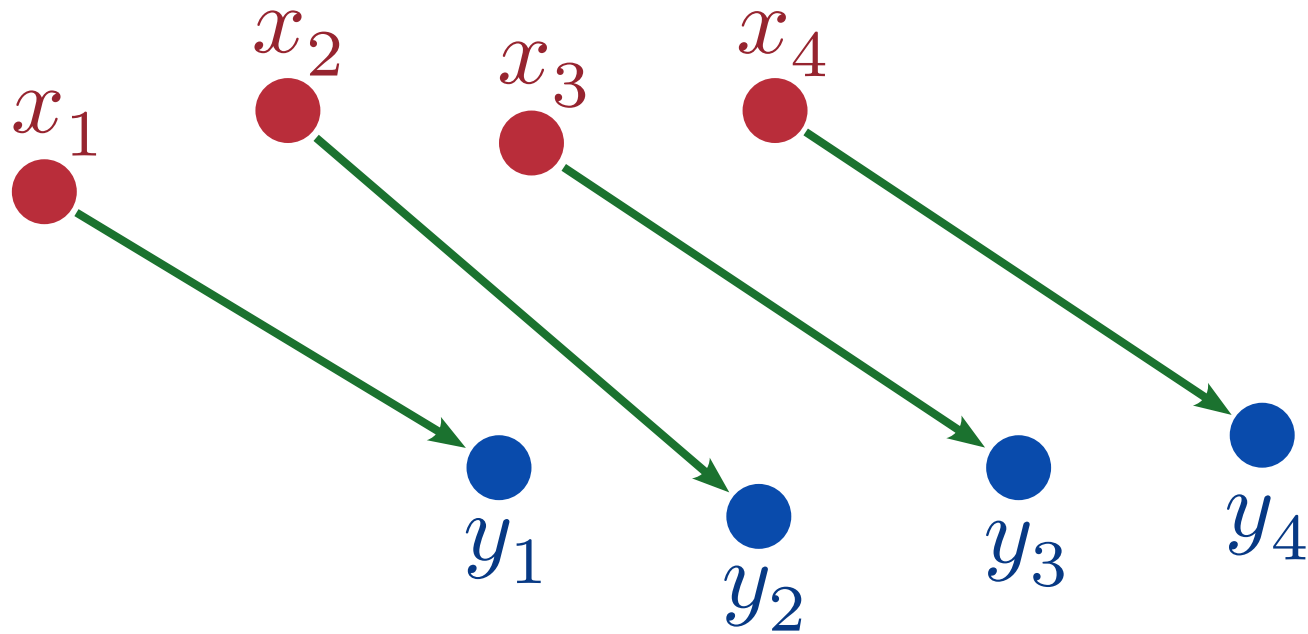
# Matching algorithms

- **Defining** the objective is **easy**.
  - **Optimizing** it is **hard**:
    - **Local** minima.
    - **Combinatorial** constraints.
  - **Machine learning** turns **inference** costs into **training** costs, but does not magically solve those issues.
- ⇒ Strong motivation to **understand** the optimization **landscape**.

## Geometric insights

---

# From permutations $\sigma$ to transport plans $P$

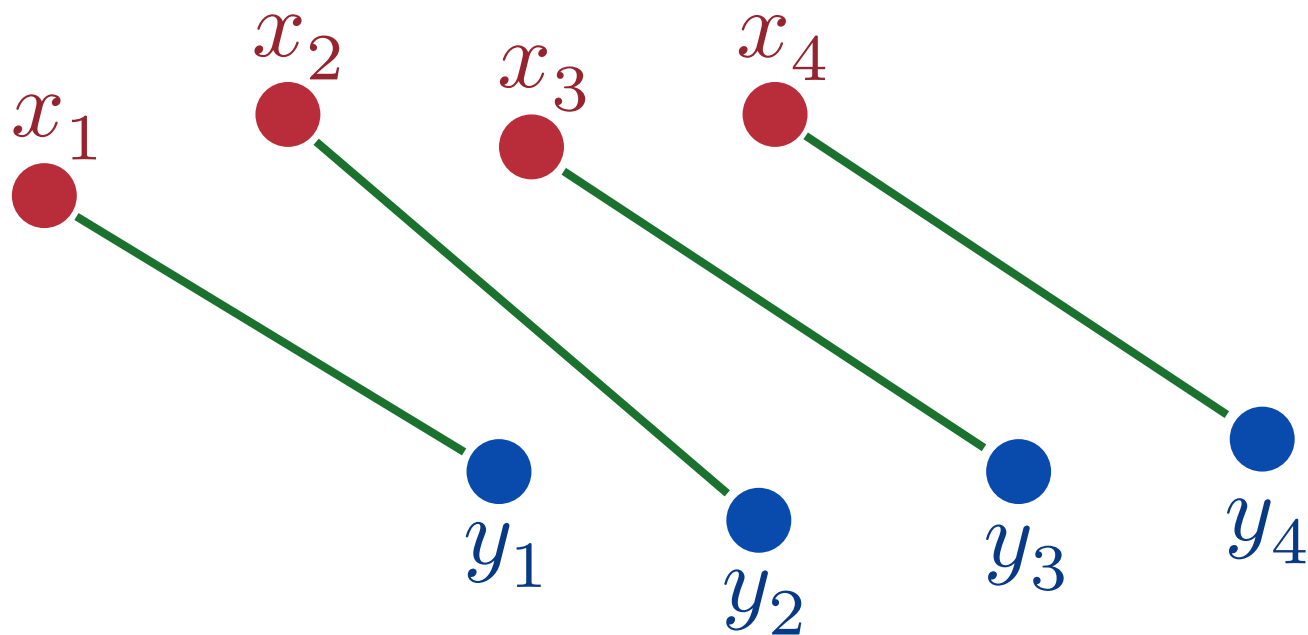


$\sigma \hookrightarrow$

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |

**Permutations**  $\sigma : \llbracket 1, 4 \rrbracket \rightarrow \llbracket 1, 4 \rrbracket$  have their **limitations**.

# From permutations $\sigma$ to transport plans $P$

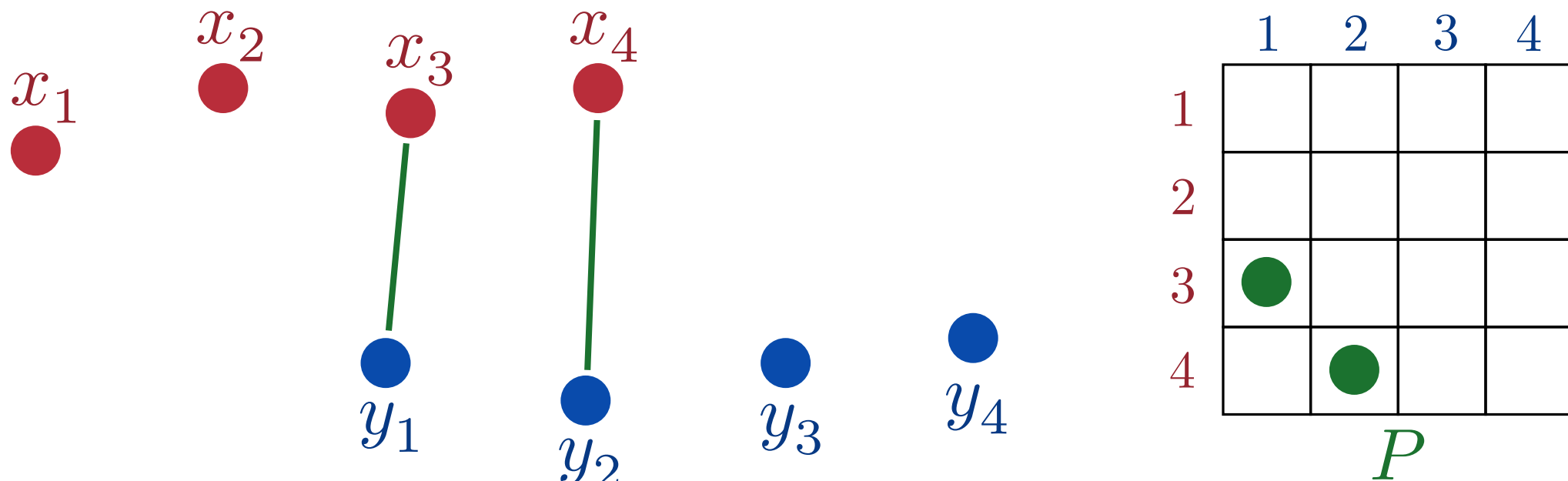


|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | ● |   |   |   |
| 2 |   | ● |   |   |
| 3 |   |   | ● |   |
| 4 |   |   |   | ● |

$P$

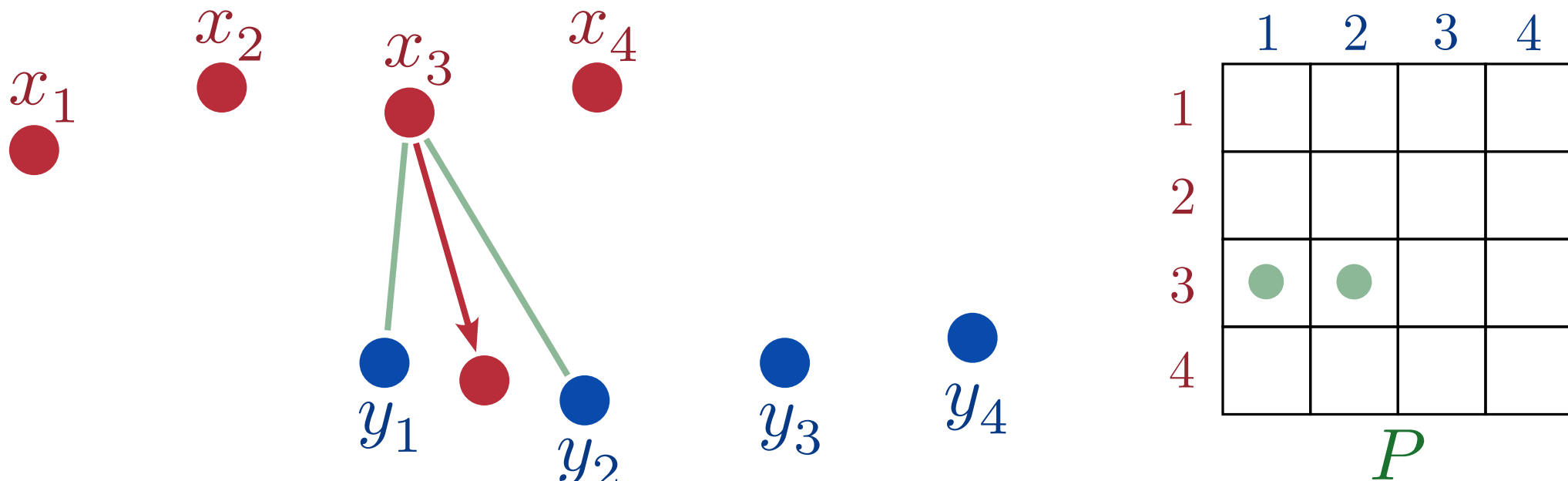
We prefer to use **transport plans**  $P_{ij} \simeq \text{coupling}(x_i \leftrightarrow y_j)$ ,  
 $4 \times 4$  matrices that respect the **symmetry** of the problem.

# From permutations $\sigma$ to transport plans $P$



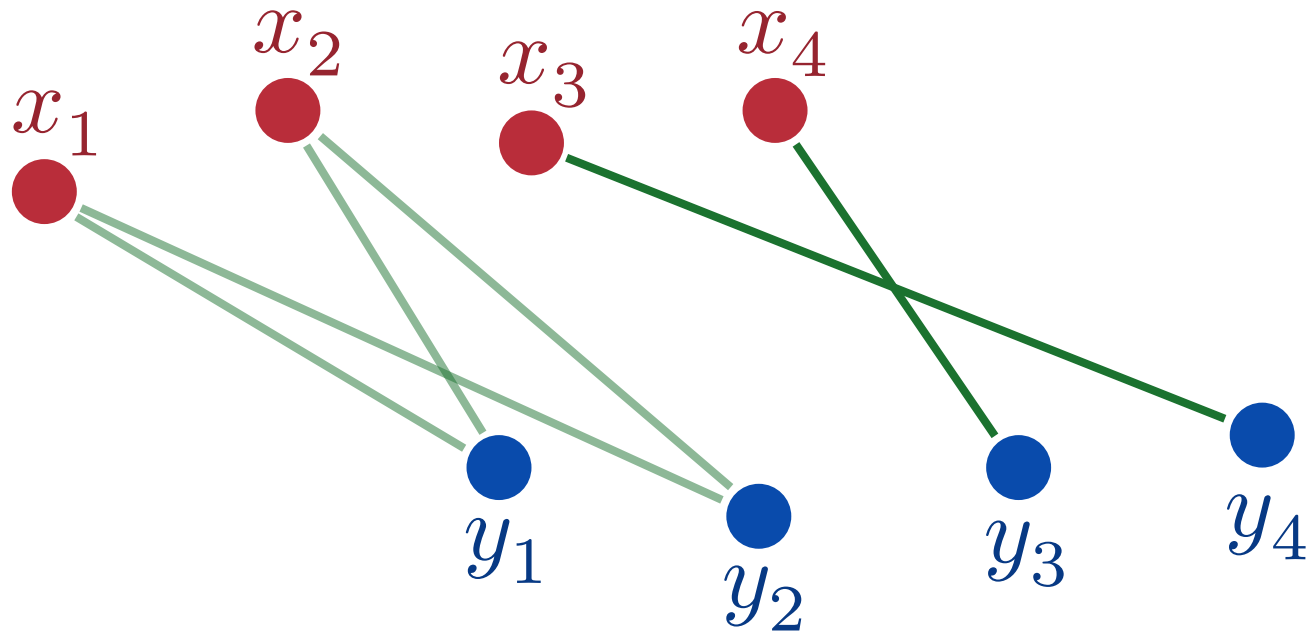
Transport plans let us encode **partial** matchings.

# From permutations $\sigma$ to transport plans $P$



Using **fractional weights**, we can also **interpolate** between the source and target points.

# From permutations $\sigma$ to transport plans $P$

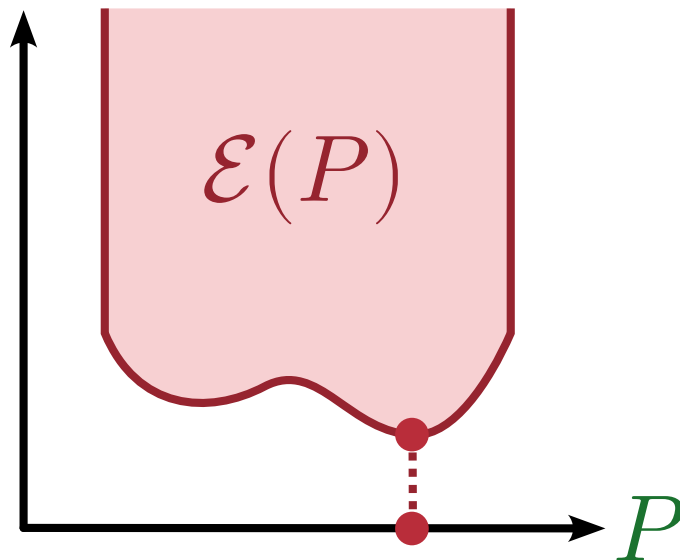


|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | ● | ● |   |   |
| 2 | ● | ● |   |   |
| 3 |   |   |   | ● |
| 4 |   |   | ● |   |

$P$

**Permutation-like** transport plans have  
**rows** and **columns** that sum up to 1.

# Why bother with optimization problems?

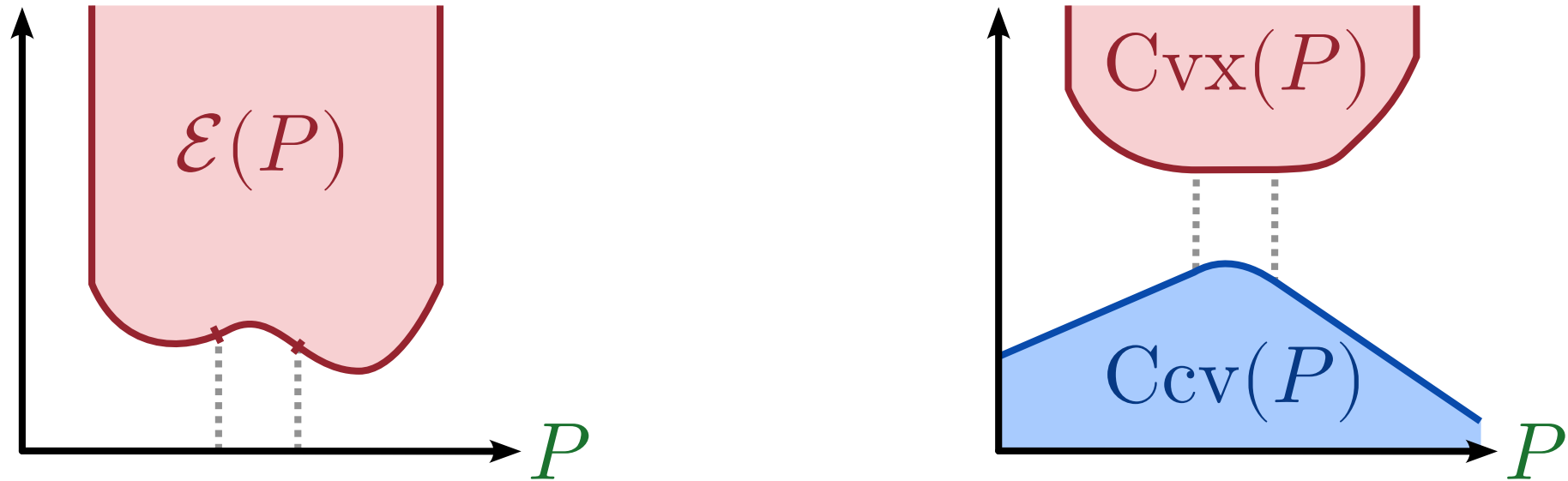


Algorithm : (source  $A$ , target  $B$ )  $\mapsto$  plan  $P^* = \arg \min_P \mathcal{E}(P)$

This formulation **decouples**:

- **Modeling choices** that have an impact on the energy  $\mathcal{E}(P)$  and the matching  $P^*$ .
- **Optimization tricks** that speed up the algorithm.

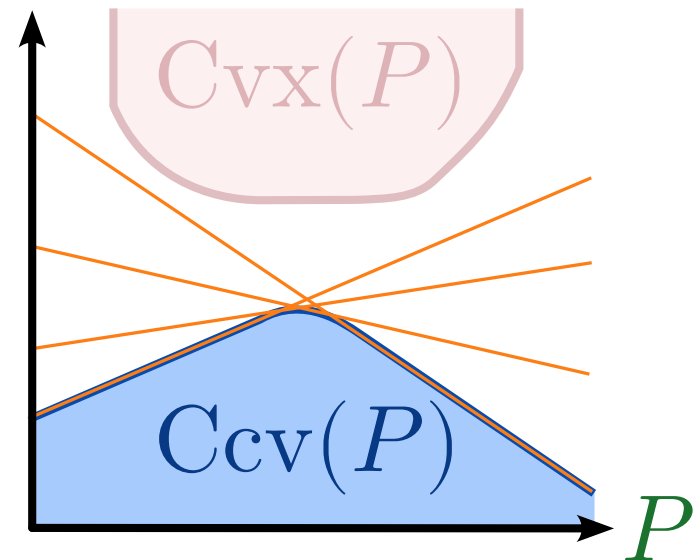
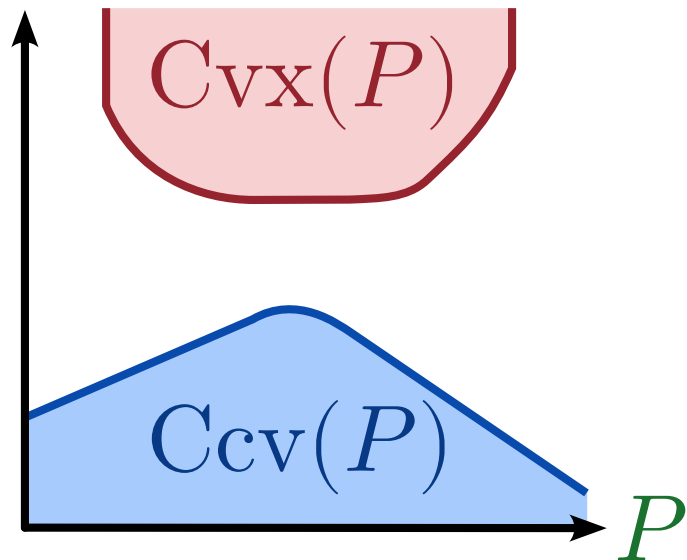
# We can “always” split the energy as a difference of convex functions



$$\mathcal{E}(P) = \underbrace{\text{Convex}(P)}_{\text{easy}} + \underbrace{\text{Concave}(P)}_{\text{hard}}$$

The **convex** term encodes the **constraints**.  
The **concave** term encodes the **basins of attractions**.

And rewrite the concave part as a minimum of linear functions of  $P$



$$\underbrace{\text{Convex}(P)}_{\text{easy}} + \underbrace{\text{Concave}(P)}_{\text{hard}} = \underbrace{\text{Convex}(P)}_{\text{easy}} + \min_S \underbrace{\langle P, S \rangle + \text{Convex}'(S)}_{\text{trivial wrt. } P, \text{ easy wrt. } S}$$

The dual variable  $S$  encodes the **slopes** of the **tangents** to the **concave** term.  
 Their **offsets** are given by  $\text{Convex}'(S)$ .

## In practice, looking at popular energies $\mathcal{E}(P)$

- $\text{Convex}(P)$  encodes **combinatorial constraints** (soft or hard bijectivity...) and a **sharpness or fuzziness** prior on the final result  $P^\star$ .  
**Easy to optimize** with a nearest neighbor projection or an OT solver.

- The auxiliary variable  $S$  is a **similarity matrix**. We often have:

$$S_{ij} = \langle x_i, \Gamma y_j \rangle = x_i^\top \Gamma y_j \in \mathbb{R}$$

for some **point features**  $x_i, y_j$  and a **matrix**  $\Gamma$  to optimize.

- $\text{Convex}'(S) \simeq \text{Convex}'(\Gamma)$  encodes a **smoothness** prior on the deformation.  
**Easy to optimize** with linear algebra routines (SVD...).

# A robust and popular method

We are looking for:

$$\begin{aligned} P^* &= \arg \min_P \mathcal{E}(P) \\ &= \arg \min_P \min_S \text{Convex}(P) + \langle P, S \rangle + \text{Convex}'(S) \end{aligned}$$

Starting from an arbitrary similarity  $S$ , we **alternate** the updates:

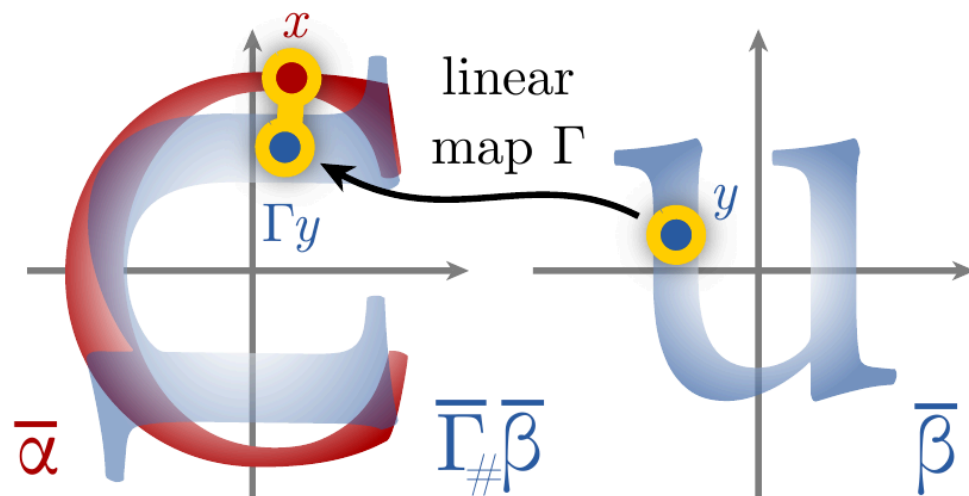
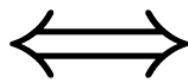
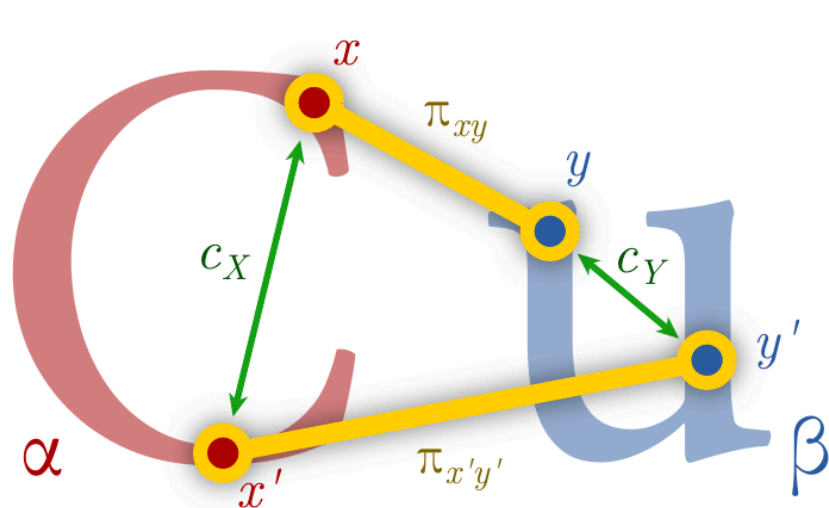
**Matching**  $P \leftarrow \arg \min_P \langle P, S \rangle + \text{Convex}(P)$

**Deformation**  $S \leftarrow \arg \min_S \langle P, S \rangle + \text{Convex}'(S)$

Converges towards a **local minimum** of  $\mathcal{E}(P)$ .

Use multiple seeds to explore the symmetries of the problem.

# Example: Gromov-Wasserstein with squared Euclidean costs



**Quadratic** assignment problem



**Affine** registration in feature spaces

**Complexity** of the problem



**Dimensions** of the feature spaces

Local **minima** of the problem



**Symmetries** of the embeddings

## **Practical consequences**

---

# Unified and modular structure $\implies$ better software support

In the **literature**, one method = one solver = one implementation.

Users have to choose between:

- **Reference** implementations: **stable**, well-tested, but may be CPU-only.
- **Custom** re-implementations: less reliable but **fit in the workflow**.

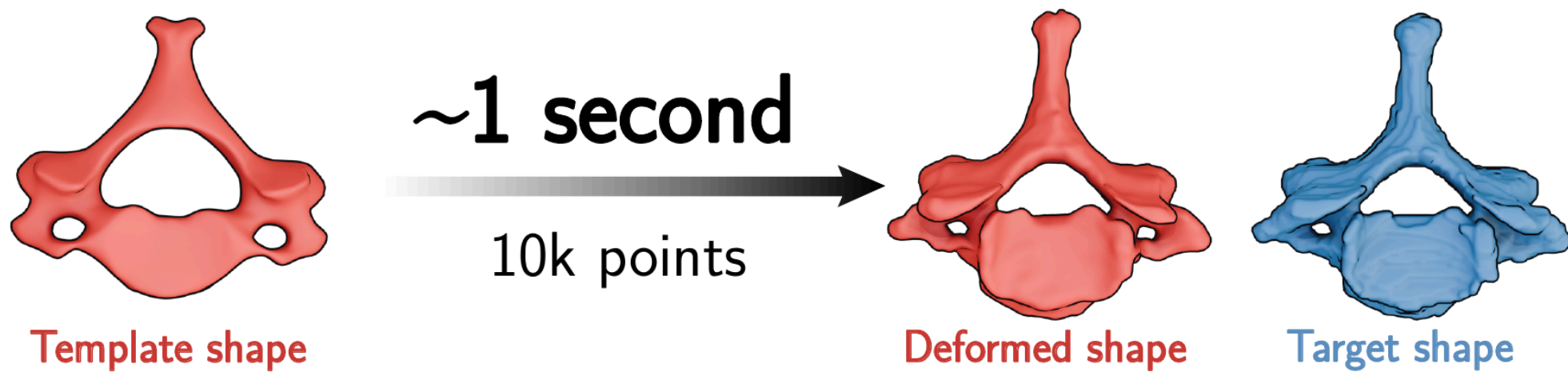
We can now optimize **all popular energies** with the same modules:

- **Features computation**: problem-specific.
- **Matching step**: nearest neighbor projection or optimal transport.
- **Deformation step**: standard linear algebra.

Reduces the **maintenance** load, easier to **deploy** on all platforms.

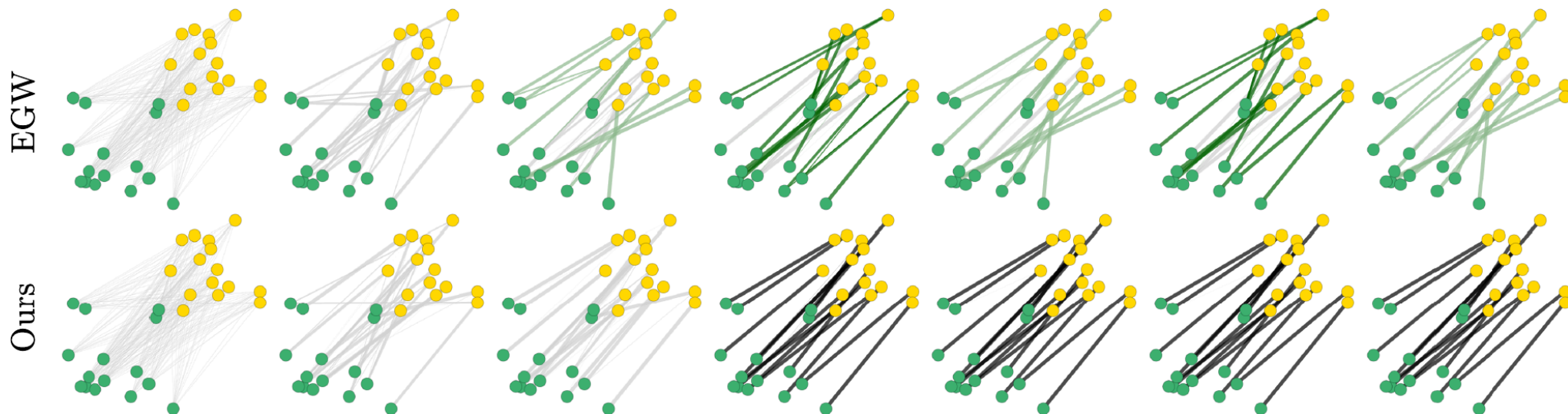
$\implies$  The **scikit-shapes** project (WIP).

# Faster algorithms, beyond gradient descent



**Optimal steps** for fast diffeomorphic shape registration:  
oral at MICCAI 2026, 40x faster than Deformetrica.

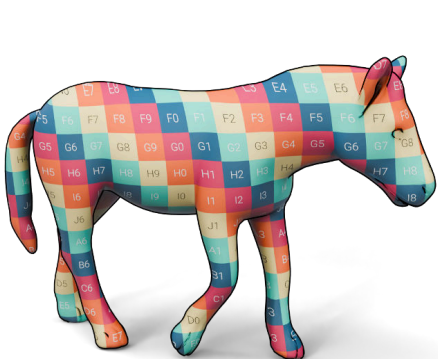
# Convergence guarantees



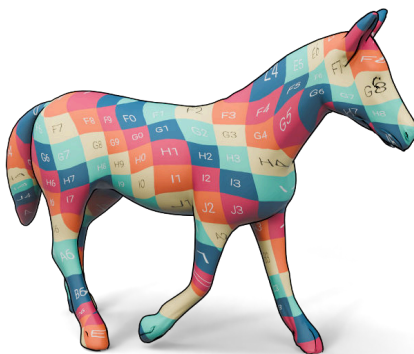
Trying to find an **optimal coupling** for  $\mathcal{E}(P) \sim (\|x - x'\|^6 - \|y - y'\|^6)^2$ , the standard EGW solver of Peyré et al. (2016) **oscillates** between **two sub-optimal matchings**.

Our method **converges** cleanly.

# Sharper, cleaner results



(a) Source



(b) Ours



(c) LR-GW, rank = 100



(d) LR-GW, rank = 500

**Scalable methods** from the ML community such as LR-GW (Scetbon et al. 2022) often rely on **low-rank approximations**.

Our geometric formulation **avoids** the creation of **undesirable artifacts**.

# Ease of use and modularity



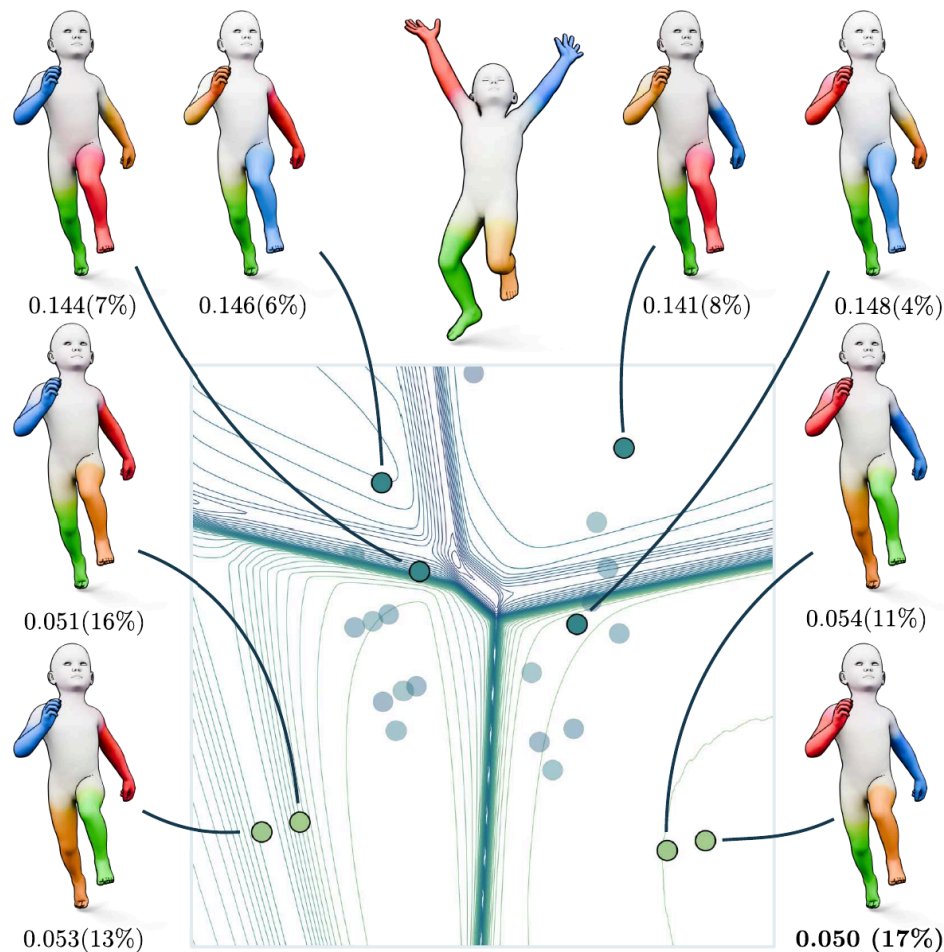
Matching the left-most surface with five **partial acquisitions**.  
We can now **relax bijectivity constraints** for any deformation model.

# Exploration of the energy landscape

$A = \text{jumping} \rightarrow B = \text{running}$ ,  
 $\mathcal{E}(P) = \text{preserve geodesic distances}$   
with features of dimension 20.

Starting from 500 **random matrices**  $\Gamma$ ,  
we display the **8 best local minima**  
in a 2D PCA plane.

In the bottom-right corner, the **global**  
minimum ( $\mathcal{E}(P) = 0.050$ ) corresponds to  
the **correct permutation** of the limbs.  
It captured **17% of the random seeds**.



# Conclusion

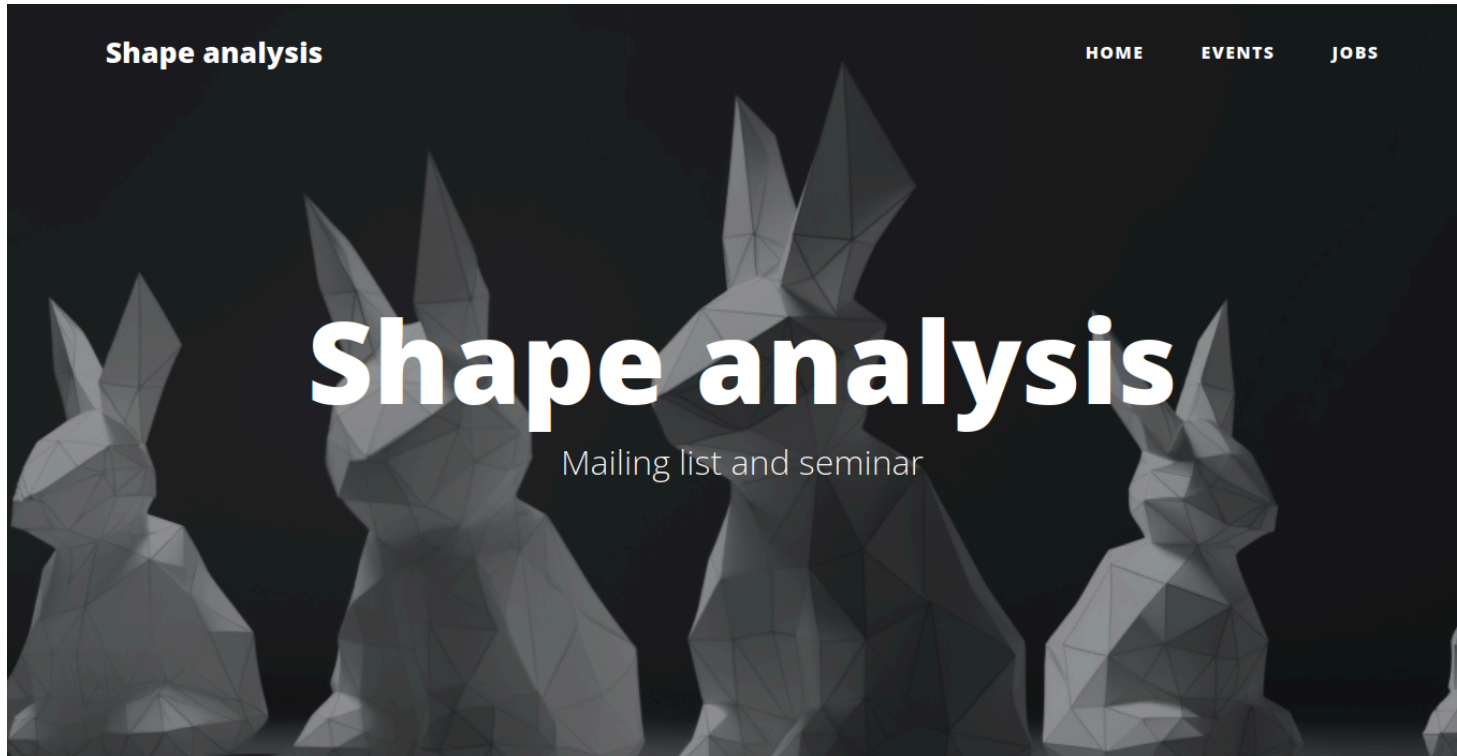
- **Matching** generalizes **sorting** to spaces of dimension  $D > 1$ .  
Studied for machine learning, biostatistics, geometry processing...
- **Defining** a relevant objective  $\mathcal{E}(\sigma)$  is **not enough**.  
Naive methods require 100's of iterations, may fail to converge.
- Recent **theoretical insights**:
  - Unlock **scalable** and **robust** solvers.
  - Are relevant for both **learning-based** and **learning-free** approaches.

# References

- **Optimal steps** for fast diffeomorphic shape registration,  
Hadrien Bigo-Balland, Tom Boeken, Jean Feydy, MICCAI 2026.
- **Gromov-Wasserstein** at scale, beyond squared norms,  
Guillaume Houry, Jean Feydy, François-Xavier Vialard, ICML 2026.
- **A unified dual method** for matching problems,  
Guillaume Houry, Ferdinand Genans, Jean Feydy, François-Xavier Vialard, Arxiv.

# Documentation and tutorials are available online

⇒ [shape-analysis.github.io](https://shape-analysis.github.io) ⇐



Monthly seminar, videos on YouTube.

Any questions?

**Main content** goes here. aaab lkbe lkjljka  
lkjj

**Sidebar content** goes here. lkjlkj

# A Heading

Some text here.

Just normal text. Notice how this now starts at the exact same height as the heading! alkjlkj m

alkj

# Forced Top Alignment

Content for the left side.

Content for the right side.

akjl

# Vertically Centered Columns

Short text on the left.

A much taller block of text on the right.

Because of `align: horizon`, the short text on the left will float perfectly in the vertical middle of this taller block.

A slide with equation:

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

## Second Section

---