Normalizing diffusion kernels with optimal transport

How to encode smoothness?

Nathan Kessler, Robin Magnet, Jean Feydy
HeKA team, Inria Paris, Inserm, Université Paris-Cité
PR[AI]RIE-PSAI Institute

Geometric Deep Learning and Generative Models for 3D Human, Lille

Wednesday, 26 November 2025

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).

HeKA: a translational research team for public health

Hospitals

Inria Inserm

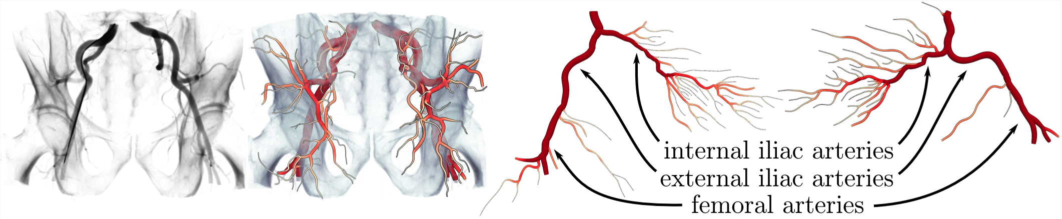
Universities

My main motivation

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

1. Speed up **geometric machine learning** on GPUs:
   $\Longrightarrow$ **pyKeOps** library for distance and kernel matrices, 900k+ downloads.
2. Scale up **pharmacovigilance** to the full French population:
   $\Longrightarrow$ **survivalGPU**, a fast re-implementation of the R survival package.
3. Ease access to modern statistical **shape analysis**:
   $\Longrightarrow$ **GeomLoss**, truly scalable optimal transport in Python.
   $\Longrightarrow$ **scikit-shapes**, beta release in January.

A vessel map that preserves vessel lengths and curvatures [HBAF25]



Our new visualization method, tailored to **endovascular** interventions.

A consistent theory of smoothness for diverse data structures

1. The **clean** method: **Laplacians** and heat **diffusions**

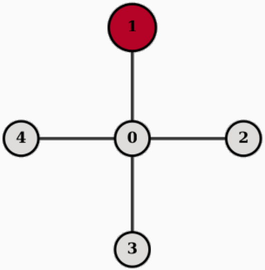
2. The **fast** method: **smoothing** with local averages

3. **Sinkhorn normalization** : **fast** smoothing $\mapsto$ **clean** diffusion

4. **Applications**

Laplacians and heat diffusions

The Laplace operator



Graph with 5 nodes.

Signals f are vectors

$(f_0, f_1, f_2, f_3, f_4)$.

$$\|f\|^2_{\text{smooth}} = (f_0 - f_1)^2 + \cdots + (f_0 - f_4)^2$$

$$\delta = \begin{pmatrix} +1 & -1 & & & \\ +1 & & -1 & & \\ +1 & & & -1 & \\ +1 & & & & -1 \end{pmatrix} \text{ is } n_{\text{edges}} \times n_{\text{points}}$$

$$\|f\|^2_{\text{smooth}} = \|\delta f\|^2 = f^\top \underbrace{\delta^\top \delta}_{\Delta} f = f^\top \Delta f$$

Properties of the Laplace operator

By construction, the **Laplacian** $\Delta = \delta^\top \delta$ is a $n_{\text{points}} \times n_{\text{points}}$ matrix that:

i)	$\Delta^\top = \Delta$	is **symmetric**
ii)	$\Delta \mathbf{1} = 0$	cancels **constant** signals
iii)	$f^\top \Delta f \geqslant 0$	is a **positive** operator
iv)	$i \neq j \Longrightarrow \Delta_{ij} \leqslant 0$	has non-positive **off-diagonal** coefficients

This can be **generalized** to weighted, continuous domains
with e.g. the **cotan** Laplacian on triangle meshes.

Regularizing signals with heat diffusion

To **regularize** a signal f, we may compute:

$$\text{Regularize}(f) = \arg\min_g \|f - g\|^2 + g^\top \Delta g$$
$$= (I + \Delta)^{-1} f \;=\simeq e^{-\Delta} f$$

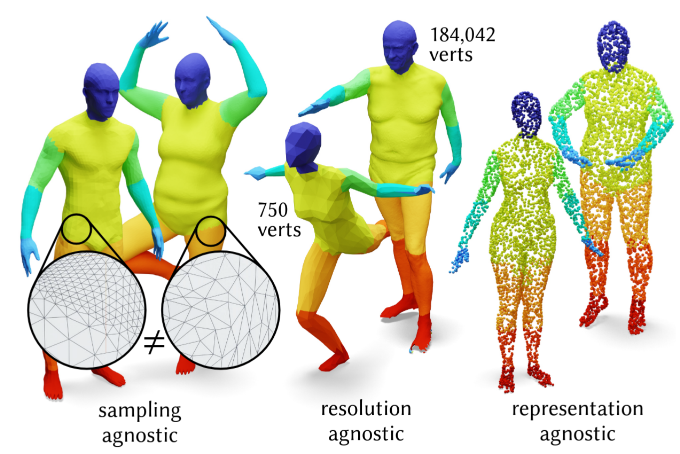
"Regularize" corresponds to a **linear diffusion operator** Q that:

i)	$Q^\top = Q$	is **symmetric**
ii)	$Q\mathbf{1} = \mathbf{1}$	preserves **constant** signals
iii)	$\text{Spectrum}(Q) \subset [0, 1]$	is a **damping** operator
iv)	$f \geqslant 0 \Longrightarrow Qf \geqslant 0$	has **non-negative** coefficients

Diffusion **preserves the mass** of input signals:

$$\langle \mathbf{1}, Qf \rangle = \langle Q\mathbf{1}, f \rangle = \langle \mathbf{1}, f \rangle$$

DiffusionNet [SACO22]

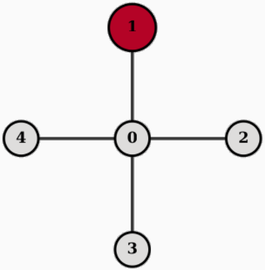


Super neat, but requires a **pre-factorization** of Δ.

Not GPU or real-time friendly.

Smoothing with local averages

Graph convolutions and smoothing with adjacency matrices



Graph with 5 nodes.

Signals f are vectors $(f_0, f_1, f_2, f_3, f_4)$.

$$S = \text{Degree} + \text{Adjacency} = \begin{pmatrix} 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & & & \\ 1 & & 1 & & \\ 1 & & & 1 & \\ 1 & & & & 1 \end{pmatrix}$$

S is a **smoothing** matrix that:

i) $S^\top = S$ is **symmetric**

ii) $f^\top S f \geqslant 0$ is a **positive** operator

iii) $f \geqslant 0 \Longrightarrow Sf \geqslant 0$ has **non-negative** coefficients

But S **does not really behave** like a diffusion: $S\mathbf{1} \neq \mathbf{1}$.

How to normalize our adjacency matrix S to recover a well-behaved diffusion Q?

1. **Row normalization.** Use $Q = (S\mathbf{1})^{-1} S$.
   This guarantees the **preservation of constants**: $Q\mathbf{1} = \mathbf{1}$.
   Problem: we lose symmetry, $Q^\top \neq Q$, $\langle \mathbf{1}, Qf \rangle \neq \langle \mathbf{1}, f \rangle$.

2. **Symmetric normalization.** Use $Q = (S\mathbf{1})^{-1/2}\, S\, (S\mathbf{1})^{-1/2}$.
   This guarantees **symmetry**: $Q^\top = Q$.
   Problem: we do not preserve constants, $Q\mathbf{1} \neq \mathbf{1}$.

3. **Sinkhorn normalization.** Iterate step 2!
   This converges quickly to a diagonal matrix $\Lambda > 0$ such that $Q\mathbf{1} = \Lambda S \Lambda \mathbf{1} = \mathbf{1}$.
   We guarantee **both symmetry** and the **preservation of constants**.

Sinkhorn turns **any smoothing matrix** $S > 0$ into a **genuine diffusion operator**.

A simple and versatile algorithm

Algorithm 1 Symmetric Sinkhorn Normalization

Require: Smoothing matrix $S \in \mathbb{R}^{N \times N}$

Initialize $\Lambda \leftarrow I_N$ ▷ Λ is a diagonal matrix, stored as a vector of size N.
while $\sum_i |\Lambda_{ii} \sum_j S_{ij} \Lambda_{jj} - 1|$ is larger than a tolerance parameter **do**
  $d_i \leftarrow \sum_j S_{ij} \Lambda_{jj}$ ▷ Matrix-vector product with S.
  $\Lambda_{ii} \leftarrow \sqrt{\Lambda_{ii} / d_i}$ ▷ Coordinate-wise update on a vector of size N.
end while
return $Q = \Lambda S \Lambda$ ▷ The diffusion Q is a positive scaling of S.

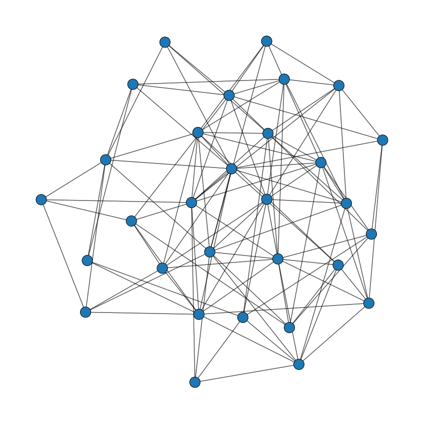
This is **much cheaper** and more **general** than using:

- $Q = (I + \Delta)^{-1}$ via a direct solver or a sparse Cholesky decomposition.
- $Q = e^{-\Delta}$ via a truncated eigendecomposition.

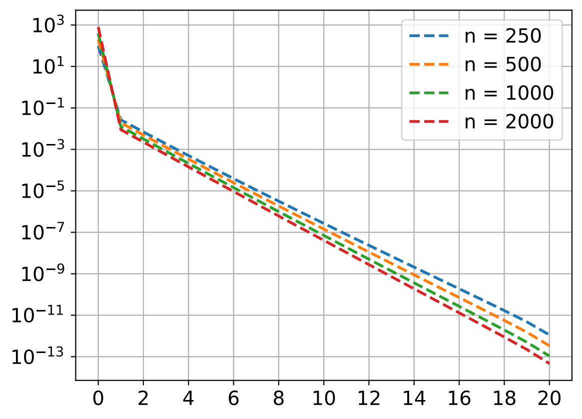
N	GPU Sinkhorn	CPU LU
10,000	3	65
50,000	21	393
100,000	89	1,030
250,000	448	3,510
500,000	1,817	9,100
1,000,000	6,789	23,600

Sinkhorn normalization

Fast convergence: monitoring the average value of $|Q1 - 1|$

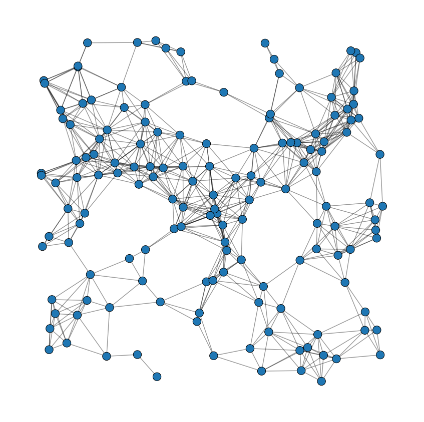


Random graph with n nodes.

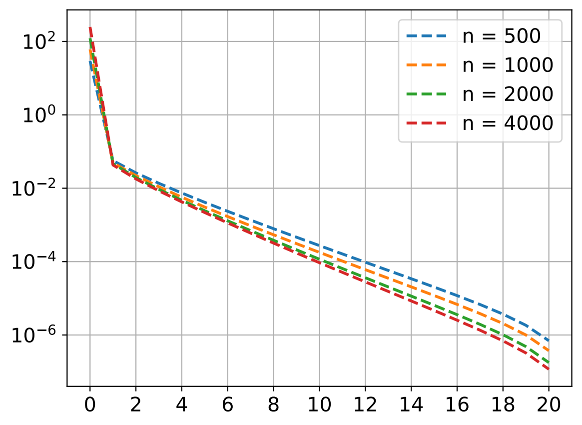


After 5 iterations: 0.01% error.

Fast convergence: monitoring the average value of $|Q1 - 1|$



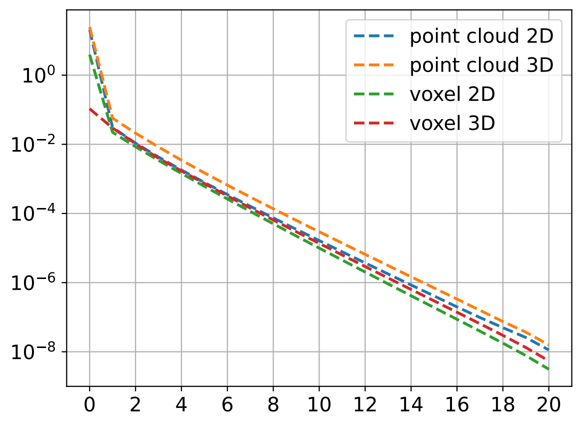
Geometric graph with n nodes.



After 5 iterations: 0.5% error.

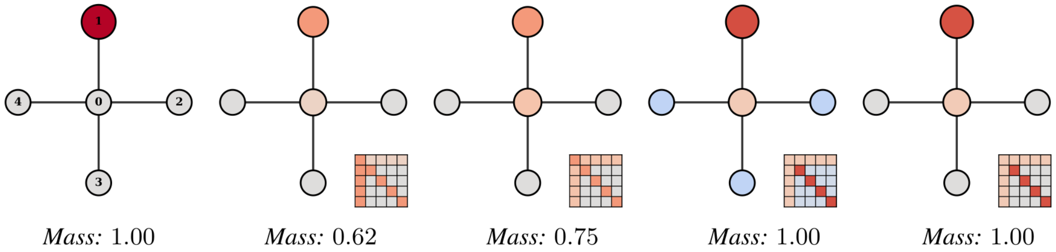
Fast convergence: monitoring the average value of $|Q1 - 1|$

Armadillo surface – 5,000 points.



After 5 iterations: 0.1% error.

Fixing the “central node bias”



(a) Input Dirac (b) Row (c) Symmetric (d) Spectral (e) Sinkhorn

Well-posedness

Focus on **convolution** with a Gaussian or exponential kernel on $\mathbb{R}^d$:

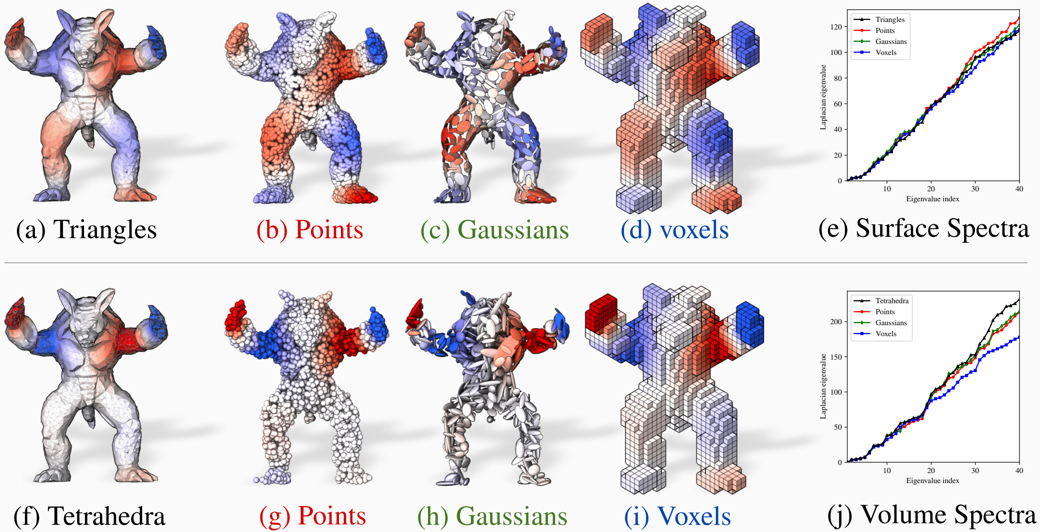
$$S_\mu f \,:\, x_i \mapsto \sum_j k(x_i, x_j) m_j f(x_j)$$

We can interpret the diagonal **scaling** matrix Λ for $Q_\mu = \Lambda S_\mu \Lambda$
as pointwise **multiplication** with a positive, **continuous** function λ.

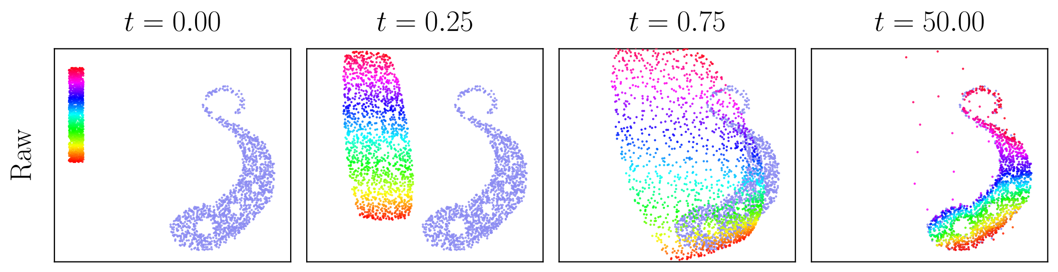
Using standard lemmas from optimal transport theory, we show that:

$$\mu^t \rightharpoonup \mu \implies \lambda^t \xrightarrow{\|\cdot\|_\infty} \lambda \implies Q_\mu f = \lambda^t S_\mu \lambda^t f \xrightarrow{\|\cdot\|_\infty} \lambda S \lambda f = Q f\,.$$

Spectral convergence – 10th eigenvector on the Armadillo

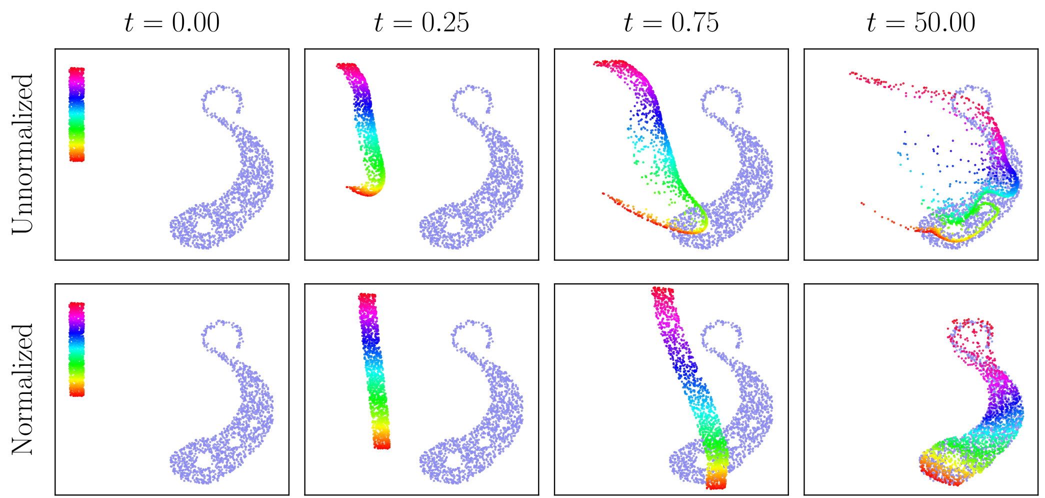


Gradient flows – without regularization

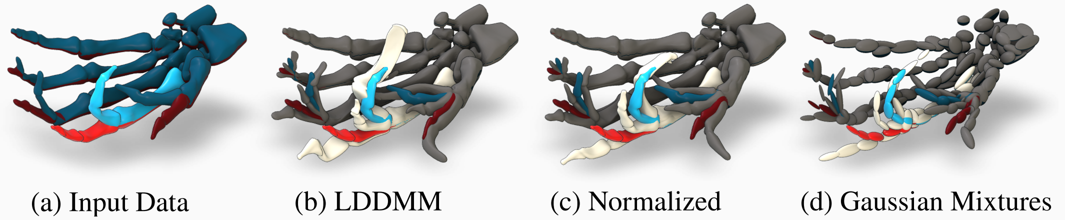


Wasserstein gradient flow of the Energy Distance.

Gradient flows – with a Gaussian regularization at scale $\sigma = 0.07$



Shape metrics – geodesic interpolation and extrapolation



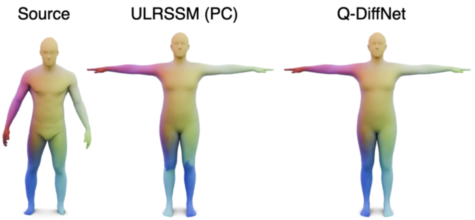
Normalizing LDDMM kernel metrics fixes the “**exploding geodesics**” problem.

We obtain a versatile and topology-preserving metric for shape analysis.

Use a normalized Gaussian convolution instead of a pre-factored Laplacian

	Method	FAUST	SCAPE	S19
Mesh	DiffNet	**1.6**	2.2	4.5
	ULRSSM	**1.6**	**2.1**	4.6
Points	DiffNet (PC)	3.0	2.5	7.5
	ULRSSM (PC)	2.3	2.4	5.1
	Q-DiffNet (QFM)	2.5	3.1	4.1
	Q-DiffNet	2.1	2.4	**3.5**

(a) Mean Geodesic Error



(b) Visualization of Correspondences

Figure 6: Evaluation of Q-DiffNet for shape matching. (a) Mean geodesic error comparison. "PC" denotes retraining on point clouds. (b) Predicted correspondences on SHREC19 [59] visualized via color transfer. ULRSSM trained on point clouds confuses symmetric parts (e.g., legs).

Conclusion

We used to face **dilemmas**:

- **Smooth** with **Laplacians** (expensive) or with **local averages** (biased).
- **Normalize** operators with **row-wise** or **symmetric** scaling.

A simple trick – **iterate** the symmetric scaling update:

- Cheap and **versatile**.
- Turn **convolutions** into genuine **diffusion** operators.
- Fix the **“central node bias”**.

Non-intrusive method to enforce theoretical axioms.

Ideally suited to modern parametric models.

References

References i

Guillaume Houry, Tom Boeken, Stéphanie Allassonnière, and Jean Feydy.
Untangling vascular trees for surgery and interventional radiology.
In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, page to appear. Springer, 2025.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov.
Diffusionnet: Discretization agnostic learning on surfaces.
ACM Transactions on Graphics (TOG), 41(3):1–16, 2022.