Representing and Mapping 3D Imaging and Spatial-omics Data Simultaneously Across Scales with Image-Varifold

Benjamin Charlier (MIAT, INRAE)

Shape Seminar, Paris — September, 16th 2025.

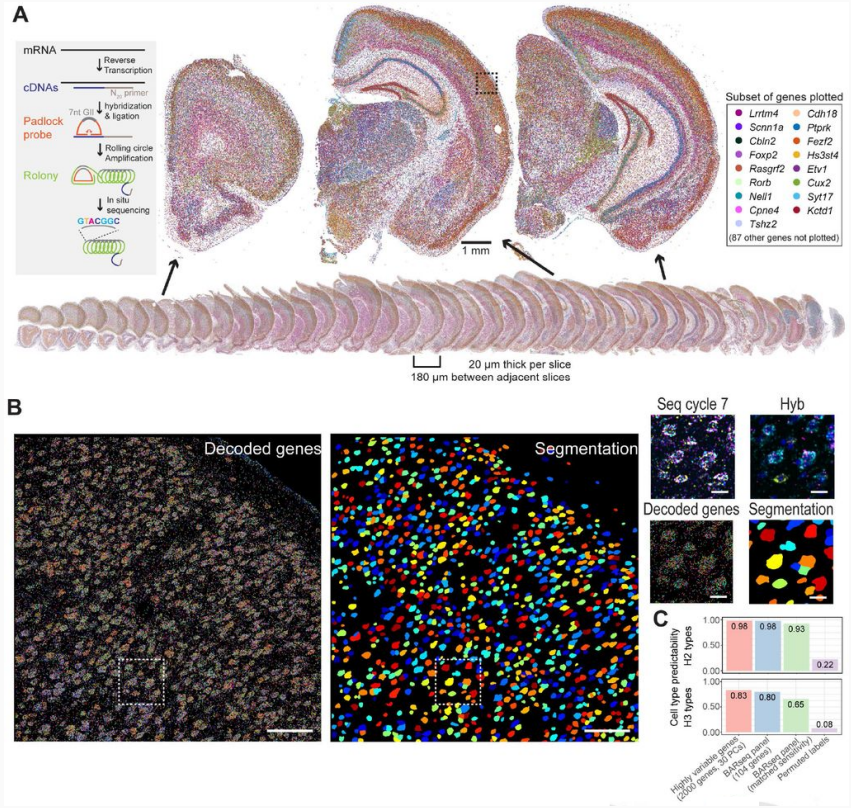
Introduction

- **Context:** Analysis of spatial transcriptomics data, characterized by multiple modalities and scales, high dimensionality, incomplete observations, and limited sample sizes.
- **Collaborators:** (1) *CIS, JHU (Baltimore):* M. Anant, J. Fan, M. Miller, **K. Stouffer**, L. Younès; (2) *ENS (Paris-Saclay), INRAE (Toulouse)):* B. C., A. Trouvé; (3) *Allen institute (Seattle):* X.Chen, M. Kunst, L. Ng, M. Rue, H. Zeng;
- **Topic:** Presentation of *cross-modality Mapping* implemented in the *cross-modality image-varifold LDDMM* (xIV-LDDMM) toolbox. `https://github.com/kstouff4/xIV-LDDMM-Particle`

Dataset: Spatial transcryptomics data

Data at molecular scale :

- **BARseq:** 32–40 coronal brain sections, identifying 39 or 52 cell types, based on raw expression of 104 genes. Hemibrain and full brain.



X. Chen et al. *Whole-cortex in situ sequencing reveals input-dependent area identity.* Nature, 2024.

- **MERFISH:** 56 coronal brain sections, profiling 500 genes.

Partial acquisitions (censored data). Feature space is denoted $\mathcal{F}$.

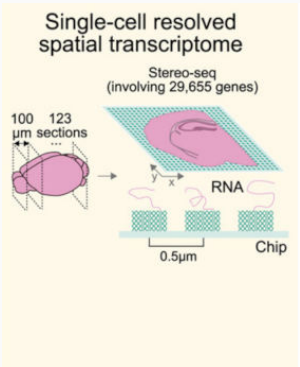
Dataset: Spatial transcryptomics data

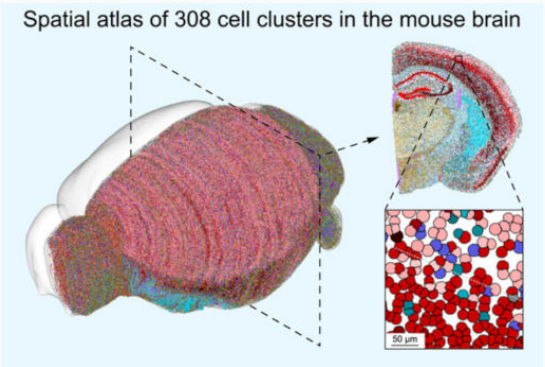
Data at molecular scale :

- **BARseq:** 32–40 coronal brain sections, identifying 39 or 52 cell types, based on raw expression of 104 genes. Hemibrain and full brain.
- **MERFISH:** 56 coronal brain sections, profiling 500 genes.

L. Han et al. *Single-cell spatial transcriptomic atlas of the whole mouse brain*. Neuron, 2025.

`https://mouse.digital-brain.cn/spatial-omics/singleCellData`

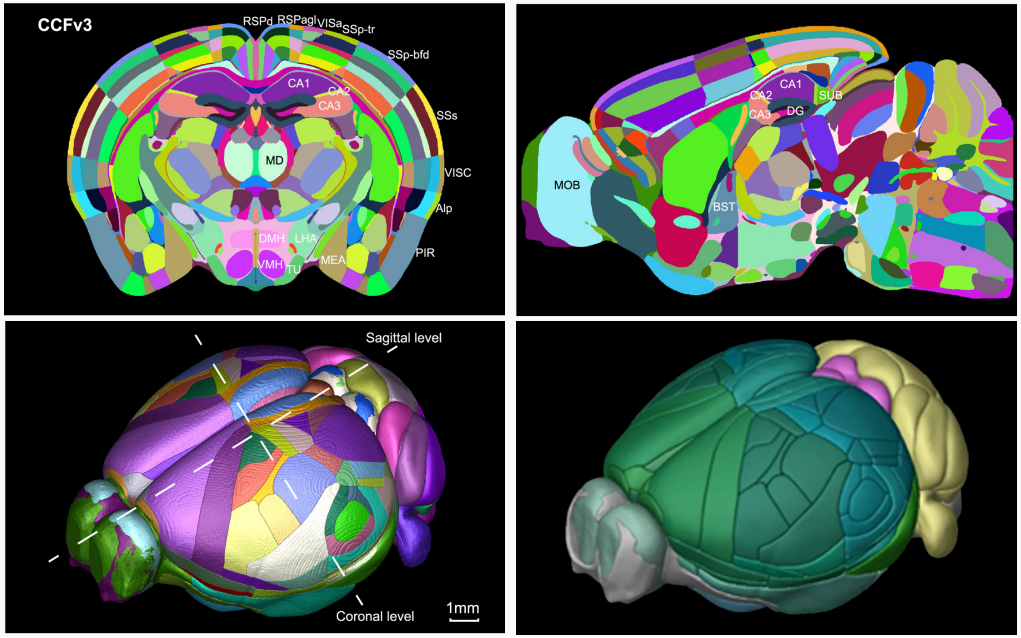




Partial acquisitions (censored data). Feature space is denoted $\mathcal{F}$.

Dataset : Brain Atlases

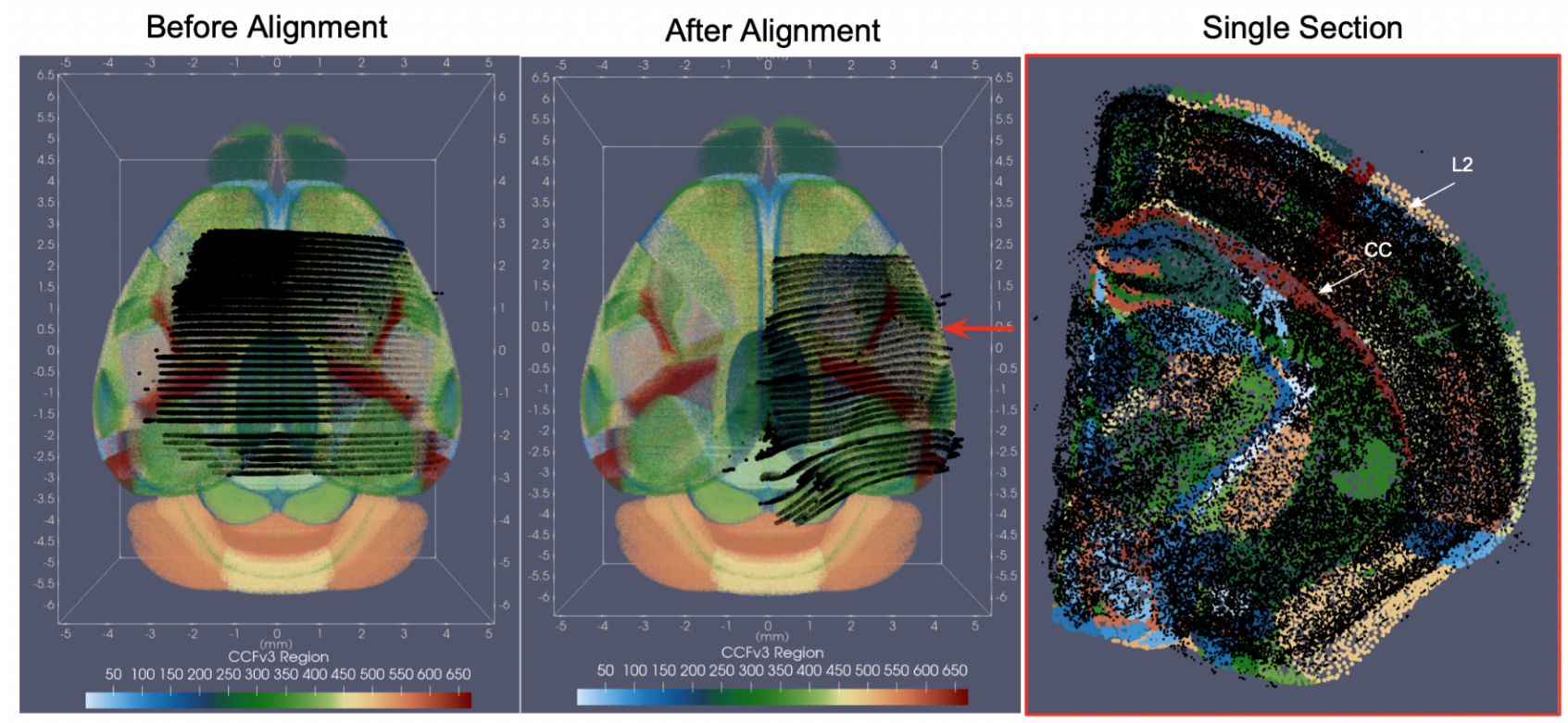
- **Data at tissue scale:** Allen Common Coordinate Framework (CCFv3), Franklin and Paxinos Atlas, etc...



Wang, Q., et al. *The allen mouse brain common coordinate framework: a 3d reference atlas.* Cell 181(4), 936–953 (2020)

- **Feature space:** atlas regions (ontology) denoted $\mathcal{L}$. Assume a spatial homogeneity inside each region: for each $\ell \in \mathcal{L}$, gene distributions (on set $\mathcal{F}$) are similar at every sites belonging to ℓ.

Global Alignment of Spatial Transcriptomics and Brain Atlas



- **Black dots:** BARseq spatial transcriptomics data (104 genes)
- **Colored regions:** Allen CCFv3 brain atlas (around 700 anatomical regions)

Stouffer KM, Trouvé A, Younès L, et al. Cross-modality mapping using image varifolds to align tissue-scale atlases to molecular-scale measures with application to 2D brain sections. Nat Commun. (2024)

Stouffer KM, Chen X, Zeng H, et al. xIV-LDDMM Toolkit: A Suite of Image-Varifold Based Technologies for Representing and Mapping 3D Imaging and Spatial-omics Data Simultaneously Across Scales. Prepint. (2025)

Keys ingredients

1. **RKHS and Non-rigid deformations:** Large Deformation Diffeomorphic Metric Mapping (LDDMM) for flexible geometric alignment.
2. **Data representation and distances:** Use of the (image) varifold framework to define geometry-aware similarity measures.
3. **Cross-modality data integration:** A registration formulation that accommodates differences in modality and spatial scale.

Non-Rigid Deformation with LDDMM

Varifold norms

Cross modalities Mapping Using Varifolds

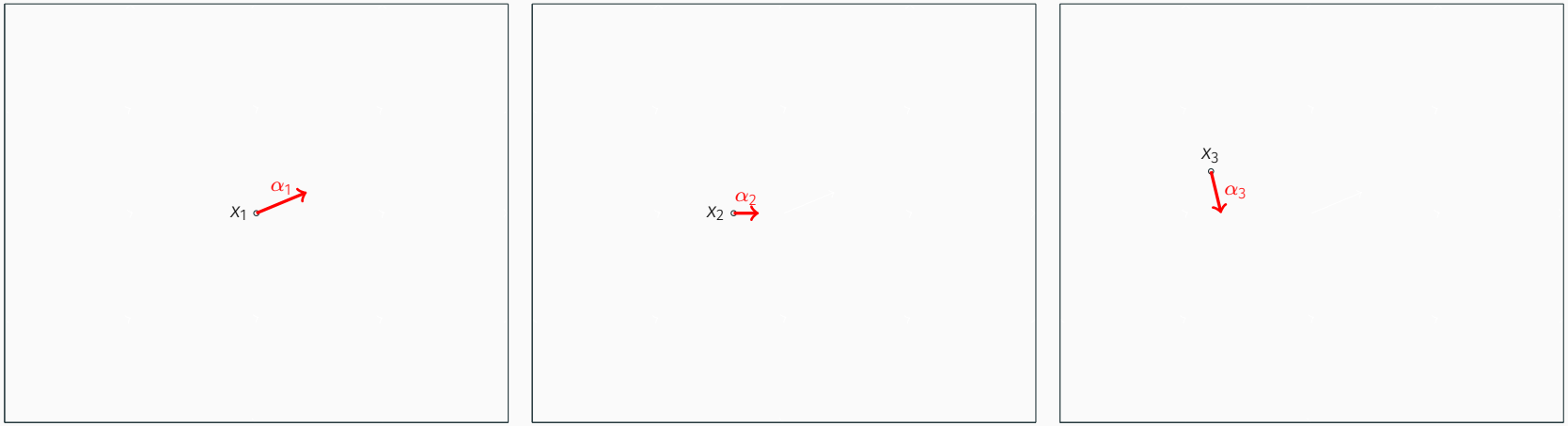
Geometrical deformations: RKHS of vectors fields

- **Space of vectors fields** V **:** an RKHS of vectors fields (smooth, vanishing at infinity). There exists a kernel $K_V : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{D\times D}$ such that

$$\text{Span}\{\delta_x^\alpha = K_V(x,\cdot)\alpha, x \in \mathbb{R}^D, \alpha \in \mathbb{R}^D\}$$

  is dense in V. In practice, $D = 2, 3$ and

$$K_V(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}} Id_D.$$



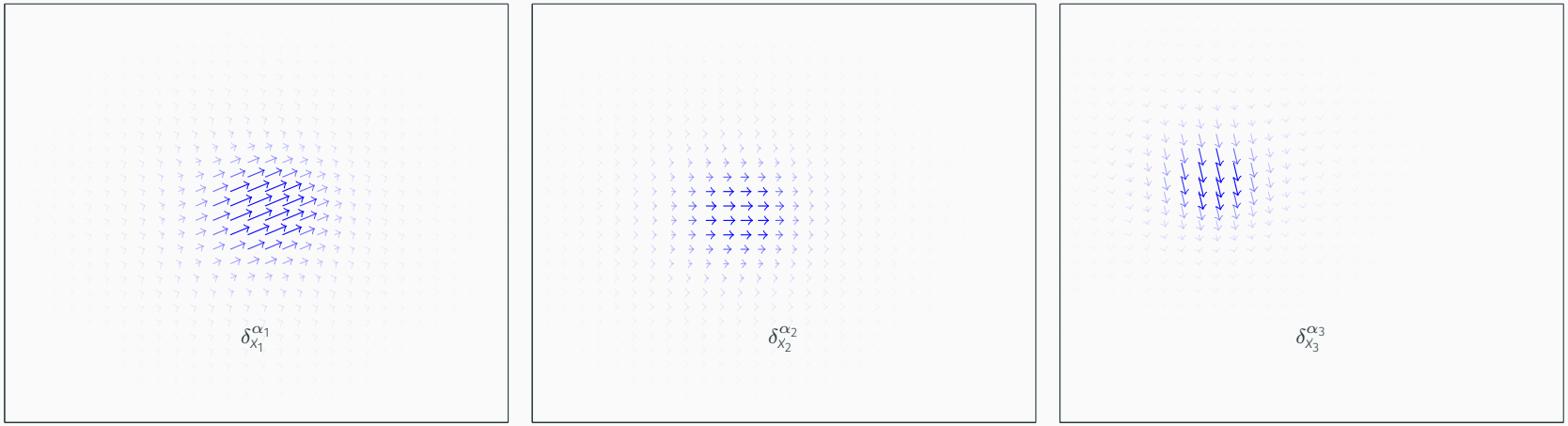
Geometrical deformations: RKHS of vectors fields

- **Space of vectors fields** V **:** an RKHS of vectors fields (smooth, vanishing at infinity). There exists a kernel $K_V : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{D\times D}$ such that

$$\text{Span}\{\delta_x^\alpha = K_V(x,\cdot)\alpha, x \in \mathbb{R}^D, \alpha \in \mathbb{R}^D\}$$

  is dense in V. In practice, $D = 2, 3$ and

$$K_V(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}} Id_D.$$



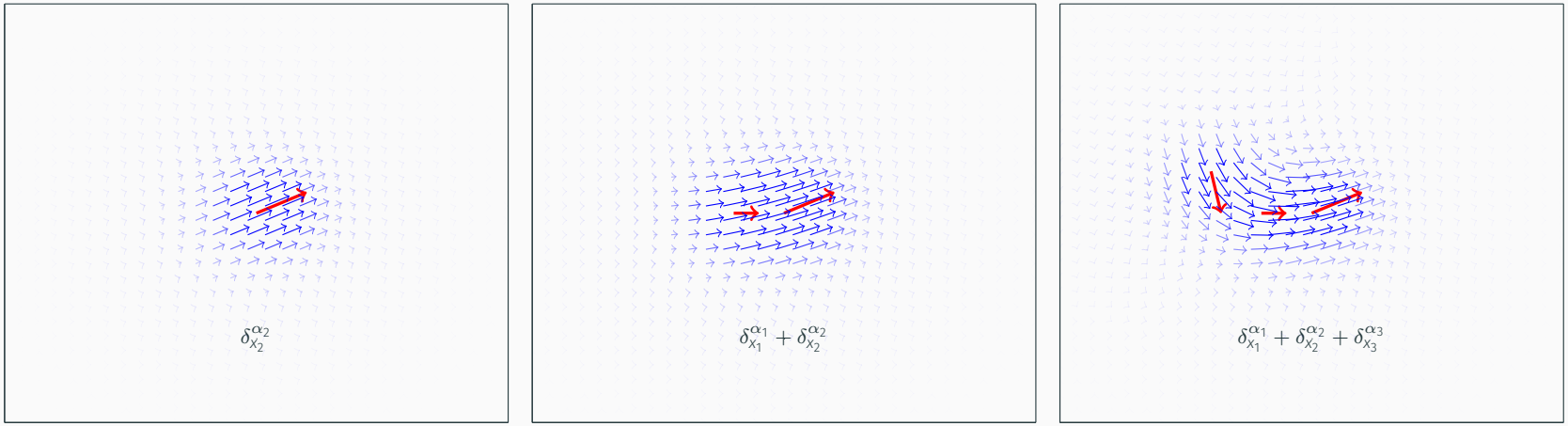
Geometrical deformations: RKHS of vectors fields

- **Space of vectors fields** V **:** an RKHS of vectors fields (smooth, vanishing at infinity). There exists a kernel $K_V : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{D\times D}$ such that

$$\mathbf{Span}\{\delta_x^\alpha = K_V(x,\cdot)\alpha, x \in \mathbb{R}^D, \alpha \in \mathbb{R}^D\}$$

  is dense in V. In practice, $D = 2, 3$ and

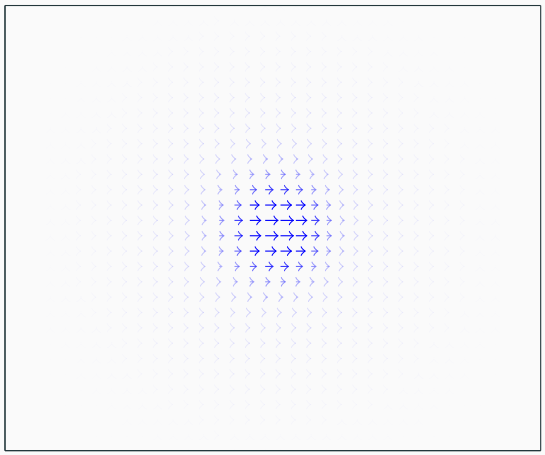
$$K_V(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}} Id_D.$$

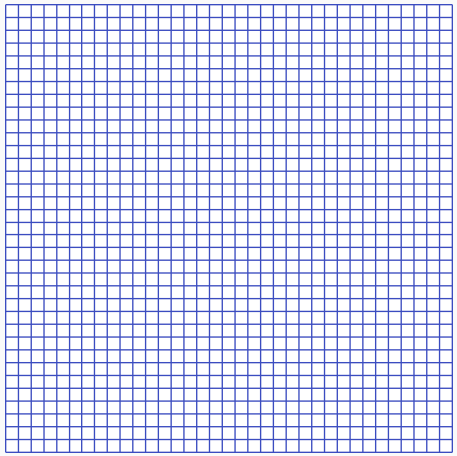


Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$





$t = 0$

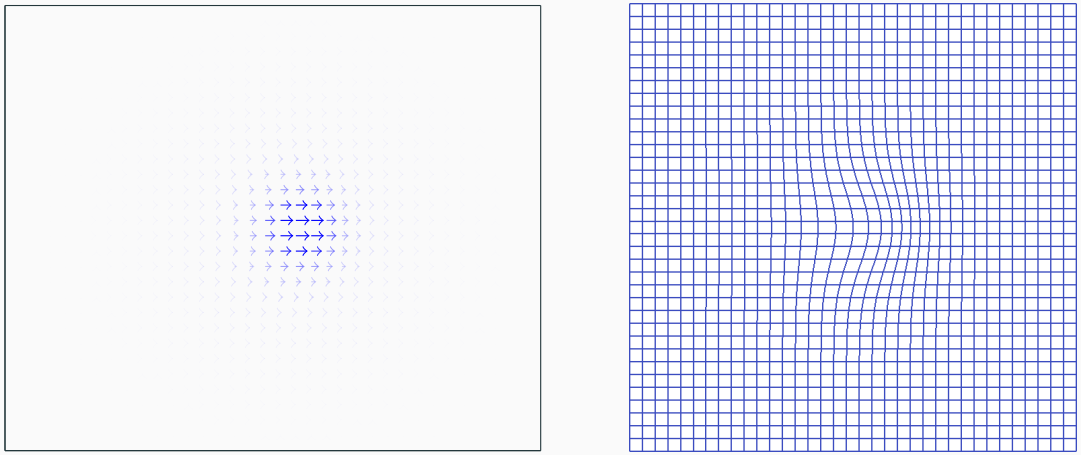
- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-difféomorphism of $\mathbb{R}^D$with distance

$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$



$t = 1/5$

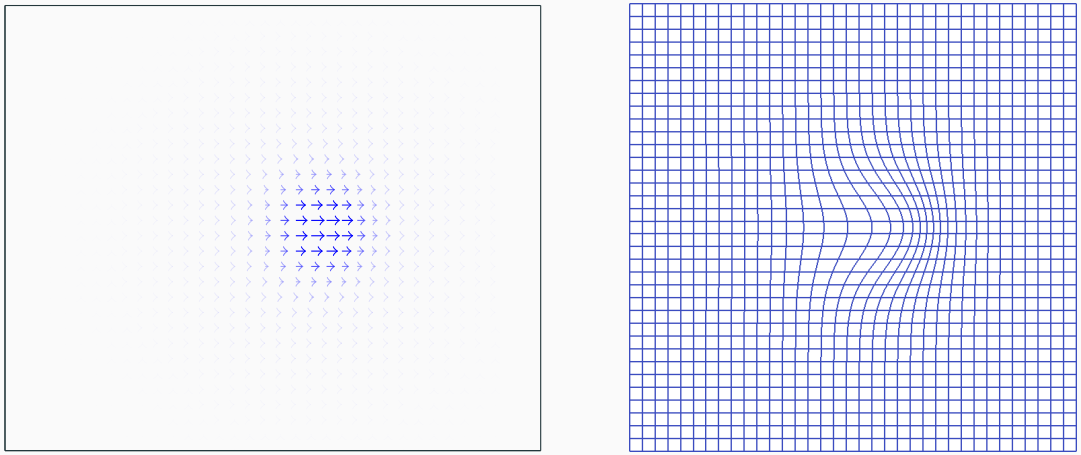
- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-difféomorphism of $\mathbb{R}^D$with distance

$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$



$t = 2/5$

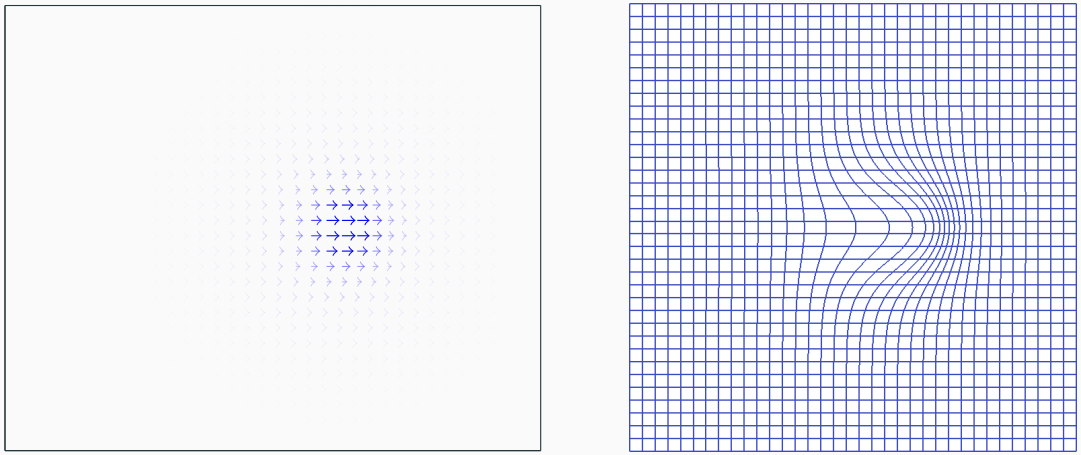
- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-difféomorphism of $\mathbb{R}^D$with distance

$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$



$t = 3/5$

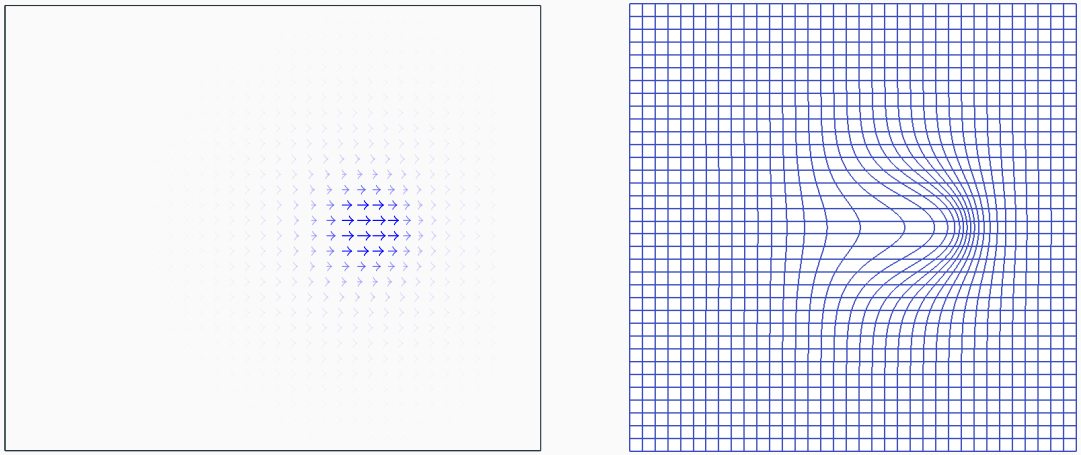
- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-difféomorphism of $\mathbb{R}^D$with distance

$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$



$t = 4/5$

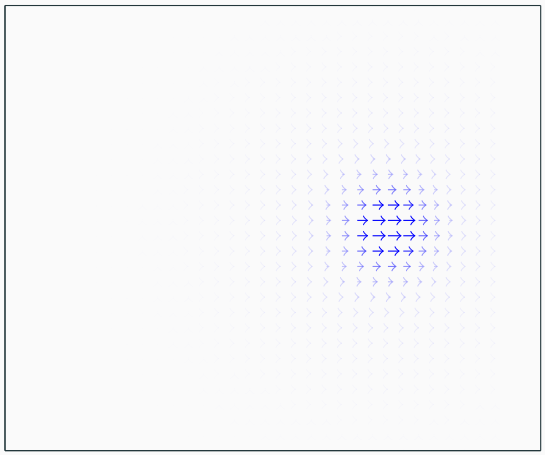
- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-diffféomorphism of $\mathbb{R}^D$with distance

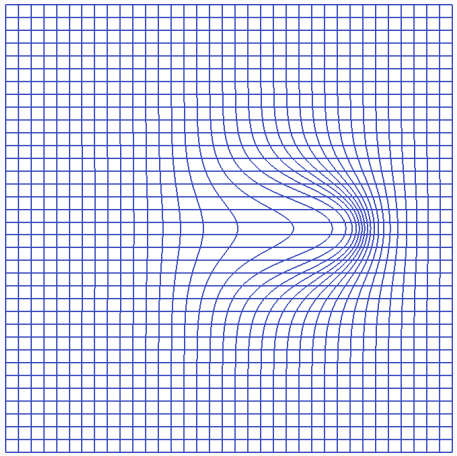
$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Geometrical deformations: flow of time varying smooth vector field

- **Flow:** let $v = (v_t)_{t\in[0,1]} \in V$ be a time dependant vectors field of $\mathbb{R}^D$. Let $\varphi : [0,1] \times \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{cases} \dot{\varphi}_t(x) = v_t(\varphi_t(x)) \\ \varphi_0(x) = x. \end{cases} \qquad t \in [0,1] \text{ and } x \in \mathbb{R}^D$$





$t = 1$

- **Group action :** $G_V = \{\varphi_1^v : \mathbb{R}^D \to \mathbb{R}^D, v \in L^2([0,1], V)\}$ are $\mathcal{C}^1$-diffèomorphism of $\mathbb{R}^D$with distance

$$d^2(\mathrm{Id}, \varphi) = \inf\{\|v\|^2_{L^2_V} \doteq \int_0^1 \|v_t\|^2_V dt, \dot{\varphi} = v \circ \varphi, \varphi_1 = \varphi\}$$

Advertisement

Generating deformations requires:

- **Fast convolutions** with kernels (e.g., Gaussian, Matérn, etc.)
- Their corresponding **derivatives**

Advertisement

Generating deformations requires:

- **Fast convolutions** with kernels (e.g., Gaussian, Matérn, etc.)
- Their corresponding **derivatives**

Since 2017, with J. Glaunès, J. Feydy we are developing  KeOps (kernels on GPU with CUDA)

- developped for Deep Learning framework (NeurIPS 2020)
- autodiff with kernels for optimisation (JMLR 2021)

- **Downloads:** 800k
- **Github stars:** 1k
- **Dependency:** 400
- **Citations:** 170

Prix science ouverte 2023

Ouvrir la science !

```
$ pip install pykeops
```

```
> remotes::install_github("getkeops/keops", subdir = "rkeops")
```

Offline Scale-Space Resampling

The full resolution acquisition is $\mu = \sum_{i \in I} \delta_{x_i} \otimes w_i p_i$.

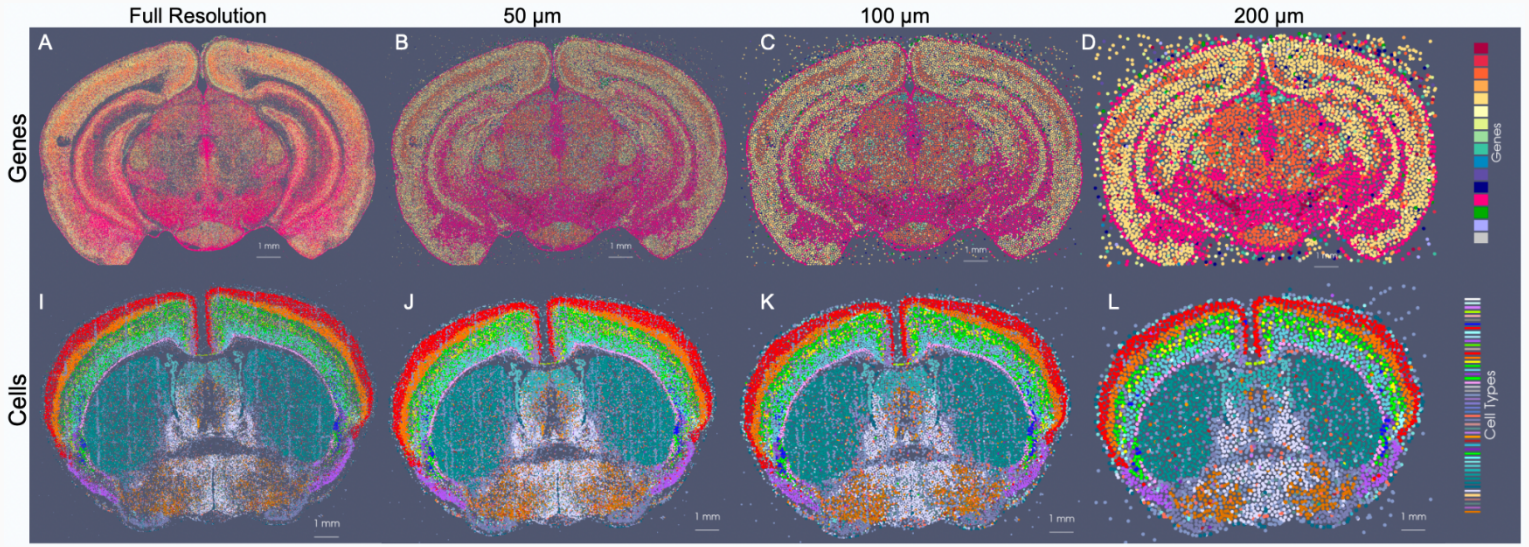
- **Series of scales:** Let $\sigma_1 = 200\,\mu m > \sigma_2 = 100\,\mu m > \sigma_3 = 50\,\mu m > \dots$ and

$$\mu_\sigma = \sum_{i \in I_\sigma} \delta_{x_i} \otimes w_i p_i, \ \{x_i, i \in I_\sigma\}, \text{ for } \sigma = \sigma_1, \sigma_2, \dots$$

- **Closest approximation** in varifold norm. Each μ_σ is defined by

$$\min_{x_i, w_i, p_i, i \in I_\sigma} \|\mu_\sigma - \mu\|_M$$

- **Practical problem:** μ do not fit in GPU memory (tiled optimization procedure).



Multi Gpu: by hands

```
import torch

# 1. Create a random tensor on CPU (to avoid OOM before splitting)
x = torch.randn(8192, 8192)
```

Multi Gpu: by hands

```python
import torch

# 1. Create a random tensor on CPU (to avoid OOM before splitting)
x = torch.randn(8192, 8192)
```

```python
# 1. Split into 8 chunks along the first dimension
chunks = torch.chunk(x, 8, dim=0)

# 2. Process each chunk on a different GPU
results = []
for i, chunk in enumerate(chunks):
    device = torch.device(f"cuda:{i}")
    # Move chunk to GPU
    chunk = chunk.to(device)
    # Apply sine
    chunk = torch.sin(chunk)
    # Move back to CPU for reaggregation
    chunk = chunk.to("cpu")
    results.append(chunk)

# 3. Concatenate results back into a single tensor
final_tensor = torch.cat(results, dim=0)
```

Calling are sequential.

Multi Gpu: Data Parallel

```python
import torch.nn as nn

# 1. Define a simple module that applies sine
class SineModule(nn.Module):
    def forward(self, x):
        return torch.sin(x)

# 2. Wrap it with DataParallel across all 8 GPUs
device = torch.device("cuda:0")  # main device
model = SineModule()
model = nn.DataParallel(model, device_ids=list(range(8)))
model.to(device)

# 3. Apply the model (DataParallel will split, scatter, gather automatically)
y = model(x)
```

Work on a single node.

Multi Gpu: Distributed Data Parallel

```python
def setup(rank, world_size):
    os.environ["MASTER_ADDR"] = "localhost"
    os.environ["MASTER_PORT"] = "12355"
    dist.init_process_group("nccl", rank=rank, world_size=world_size)

def cleanup():
    dist.destroy_process_group()

def run_worker(rank, world_size, x, return_dict):
    setup(rank, world_size)

    # Slice the pre-created tensor (each rank gets a row block)
    local_chunk = x.chunk(world_size, dim=0)[rank].to(f"cuda:{rank}")

    # Compute sine locally
    local_result = torch.sin(local_chunk)

    # Gather results back to rank 0
    ...

    cleanup()
```

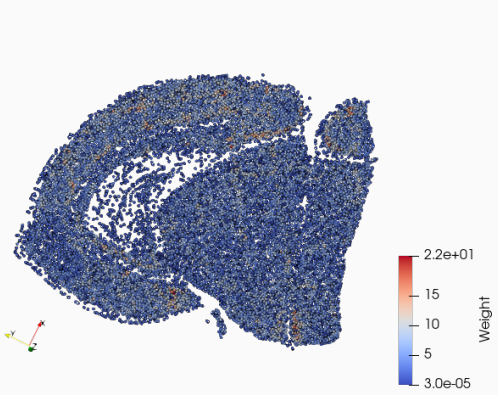
Work on multi-nodes. Beware of blocking barrier when transferring data.

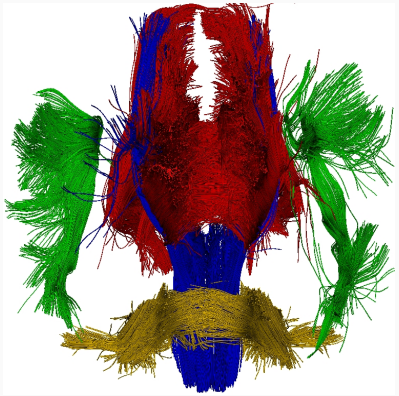
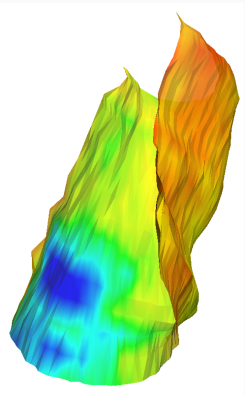
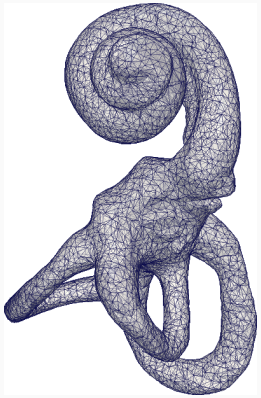
Non-Rigid Deformation with LDDMM

Varifold norms

Cross modalities Mapping Using Varifolds

Geometric measure theory to compare shapes



Data with **geometrical** information with (possibly) a **feature** attached (genes mix, label, orientation, etc.)

Glaunès, Vaillant. Surface Matching via currents. (2006)

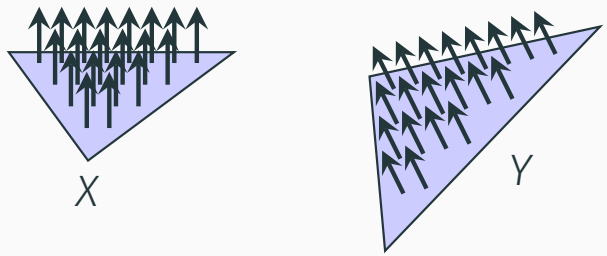
Charon, Trouvé. The Varifold representation of non-oriented shapes for diffeomorphic registration. (2013)

Kaltenmark et al. A general framework for curve and surface comparison and registration with oriented varifolds. (2017)

Geometric objects as measures

X is curve or surface in $\mathbb{R}^3$.

- A *varifold* μ_X is a **distribution** on $\mathbb{R}^3 \times S^2$, *i.e.* (position $\times$ tangent space orientation).



- A Dirac $\delta_{(x,t)}$ is a singular mass located at position $x \in \mathbb{R}^3$ in the direction of $t \in S^2$.

Remark: invariance to parametrization. Use kernel based distances.

Varifolds

Definition

A *varifold* on $\mathbb{R}^d$ is a distribution (or measure) on the space

$$\mathbb{R}^d \times G_k(\mathbb{R}^d),$$

where $G_k(\mathbb{R}^d)$ is the set of k-dimensional subspaces of $\mathbb{R}^d$ (Grassmannian).

Varifolds

Definition

A *varifold* on $\mathbb{R}^d$ is a distribution (or measure) on the space

$$\mathbb{R}^d \times G_k(\mathbb{R}^d),$$

where $G_k(\mathbb{R}^d)$ is the set of k-dimensional subspaces of $\mathbb{R}^d$ (Grassmannian).

RKHS: Let W generated by kernel $k_{pos} \otimes k_{or} : (\mathbb{R}^3 \times S^2)^2 \to \mathbb{R}$ induces scalar product on shapes:

$$\begin{aligned}&\langle \mu_X, \mu_Y \rangle_{W'} \\ &= \int_{X \times Y} k_{pos}(x, y) k_{or}(T_x X, T_y Y) d\mathcal{H}^2(x) d\mathcal{H}^2(y)\end{aligned}$$

Remark: if the chosen kernels are smooth...varifold norms are differentiable.

Varifolds

Definition

A *varifold* on $\mathbb{R}^d$ is a distribution (or measure) on the space

$$\mathbb{R}^d \times G_k(\mathbb{R}^d),$$

where $G_k(\mathbb{R}^d)$ is the set of k-dimensional subspaces of $\mathbb{R}^d$ (Grassmannian).

RKHS: Let W generated by kernel $k_{pos} \otimes k_{or} : (\mathbb{R}^3 \times S^2)^2 \to \mathbb{R}$ induces scalar product on shapes:

$$\begin{aligned}
&\langle \mu_X, \mu_Y \rangle_{W'} \\
&= \int_{X \times Y} k_{pos}(x, y) k_{or}(T_x X, T_y Y) d\mathcal{H}^2(x) d\mathcal{H}^2(y) \\
&\approx \sum_i \sum_j k_{pos}(x_i, y_j) k_{or}(\overrightarrow{n}_i, \overrightarrow{m}_j)
\end{aligned}$$

Remark: if the chosen kernels are smooth...varifold norms are differentiable.

Varifolds

Definition

A *varifold* on $\mathbb{R}^d$ is a distribution (or measure) on the space

$$\mathbb{R}^d \times G_k(\mathbb{R}^d),$$

where $G_k(\mathbb{R}^d)$ is the set of k-dimensional subspaces of $\mathbb{R}^d$ (Grassmannian).

RKHS: Let W generated by kernel $k_{pos} \otimes k_{or} : (\mathbb{R}^3 \times S^2)^2 \to \mathbb{R}$ induces scalar product on shapes:

$$\begin{aligned}
&\langle \mu_X, \mu_Y \rangle_{W'} \\
&= \int_{X \times Y} k_{pos}(x, y) k_{or}(T_x X, T_y Y) d\mathcal{H}^2(x) d\mathcal{H}^2(y) \\
&\approx \sum_i \sum_j k_{pos}(x_i, y_j) k_{or}(\overrightarrow{n}_i, \overrightarrow{m}_j)
\end{aligned}$$

Distance: $\|\mu_X - \mu_Y\|^2_{W'} = \langle \mu_X, \mu_X \rangle_{W'} + \langle \mu_Y, \mu_Y \rangle_{W'} - 2 \langle \mu_X, \mu_Y \rangle_{W'}$.

Remark: if the chosen kernels are smooth...varifold norms are differentiable.

Implementation with KeOps

Gaussian-Linear kernel $(K(x, y, u, v)b)_i = \sum_j \exp(-\sigma \|x_i - y_j\|^2) \langle u_i, v_j \rangle b_j$

```
from pykeops.torch import Vi, Vj

def GaussLinKernel(sigma):
    x, y, u, v, b = Vi(0, 3), Vj(1, 3), Vi(2, 3), Vj(3, 3), Vj(4, 1)
    gamma = 1 / (sigma * sigma)
    D2 = x.sqdist(y)
    K = (-D2 * gamma).exp() * (u * v).sum()
    return (K * b).sum_reduction(axis=1)
```

Implementation with KeOps

Gaussian-Linear kernel $(K(x,y,u,v)b)_i = \sum_j \exp(-\sigma\|x_i - y_j\|^2)\langle u_i, v_j\rangle b_j$

```
from pykeops.torch import Vi, Vj

def GaussLinKernel(sigma):
    x, y, u, v, b = Vi(0, 3), Vj(1, 3), Vi(2, 3), Vj(3, 3), Vj(4, 1)
    gamma = 1 / (sigma * sigma)
    D2 = x.sqdist(y)
    K = (-D2 * gamma).exp() * (u * v).sum()
    return (K * b).sum_reduction(axis=1)
```

Varifold data attachment loss for surfaces

```
def lossVarifoldSurf(VS, FS, VT, FT, K=GaussLinKernel):
    """VS, VT: vertices coordinates of target surface,
       FS, FT: face connectivity of source and target surfaces
       K: kernel"""
    CT, LT, NTn = get_center_length_normal(FT, VT)
    CS, LS, NSn = get_center_length_normal(FS, VS)
    return ( (LT * K(CT, CT, NTn, NTn, LT)).sum()
             + (LS * K(CS, CS, NSn, NSn, LS)).sum()
             - 2 * (LS * K(CS, CT, NSn, NTn, LT)).sum() )
```

Implementation with KeOps

Gaussian-Linear kernel $(K(x,y,u,v)b)_i = \sum_j \exp(-\sigma\|x_i - y_j\|^2)\langle u_i, v_j\rangle b_j$

```
from pykeops.torch import Vi, Vj

def GaussLinKernel(sigma):
    x, y, u, v, b = Vi(0, 3), Vj(1, 3), Vi(2, 3), Vj(3, 3), Vj(4, 1)
    gamma = 1 / (sigma * sigma)
    D2 = x.sqdist(y)
    K = (-D2 * gamma).exp() * (u * v).sum()
    return (K * b).sum_reduction(axis=1)
```

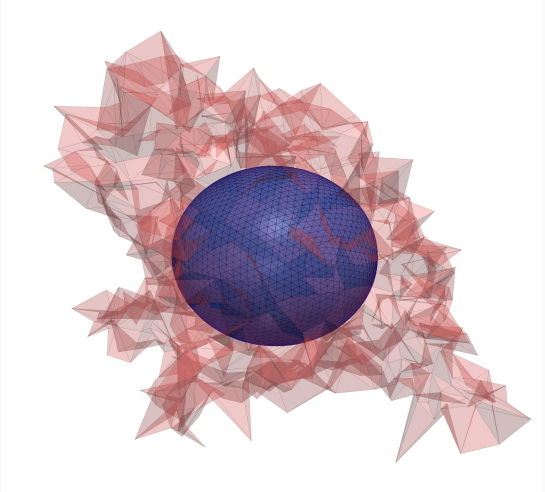
Varifold data attachment loss for surfaces

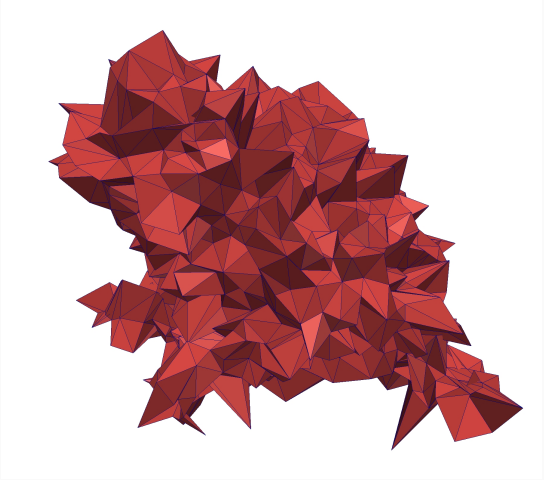
```
def lossVarifoldSurf(VS, FS, VT, FT, K=GaussLinKernel):
    """VS, VT: vertices coordinates of target surface,
       FS, FT: face connectivity of source and target surfaces
       K: kernel"""
    CT, LT, NTn = get_center_length_normal(FT, VT)
    CS, LS, NSn = get_center_length_normal(FS, VS)
    return ( (LT * K(CT, CT, NTn, NTn, LT)).sum()
             + (LS * K(CS, CS, NSn, NSn, LS)).sum()
             - 2 * (LS * K(CS, CT, NSn, NTn, LT)).sum() )
```

Compatible with torch autodiff:

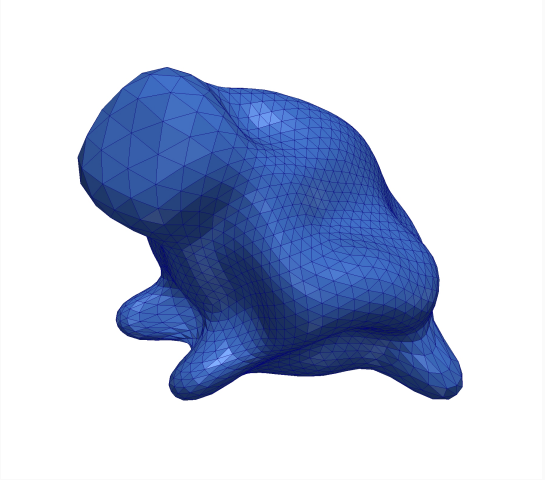
```
L = lossVarifoldSurf(q0, FS, VT, FT, GaussLinKernel(sigma))
L.backward()
```

Varifold norm are robust to noise

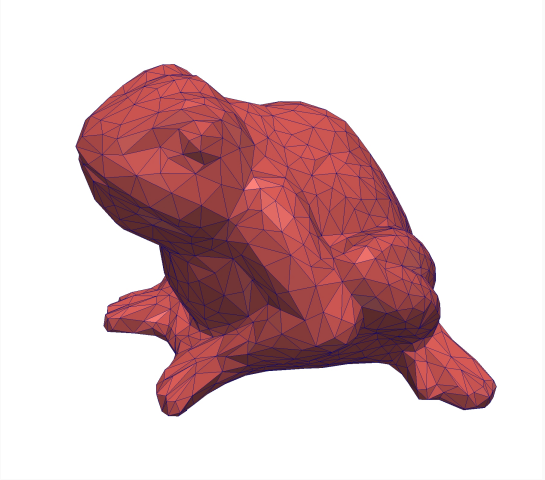




Observation

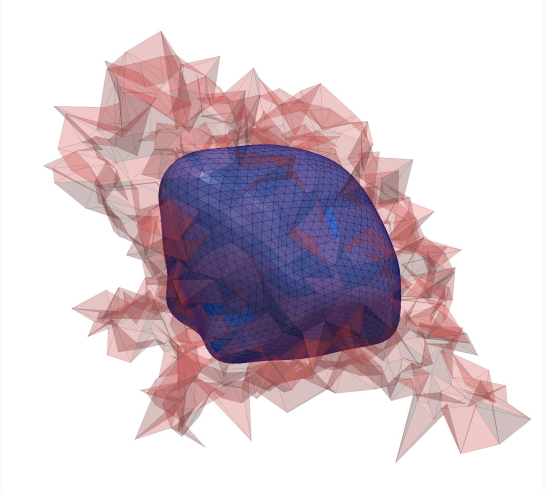


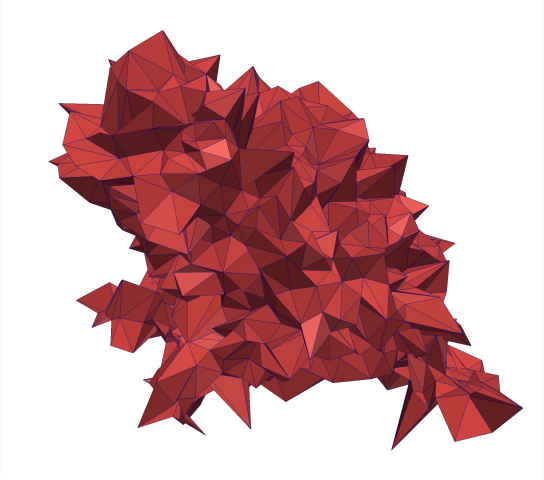
Reconstruction



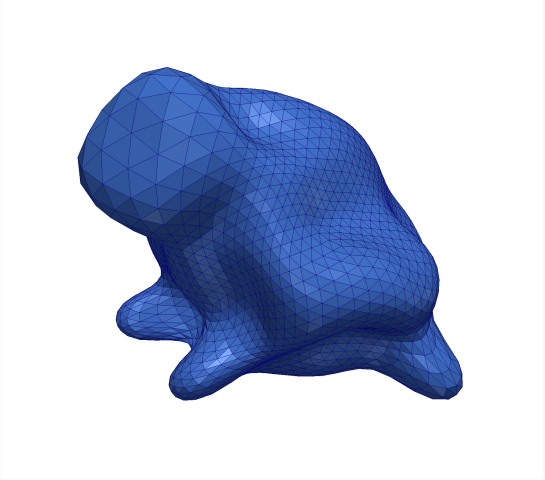
True shape

Varifold norm are robust to noise

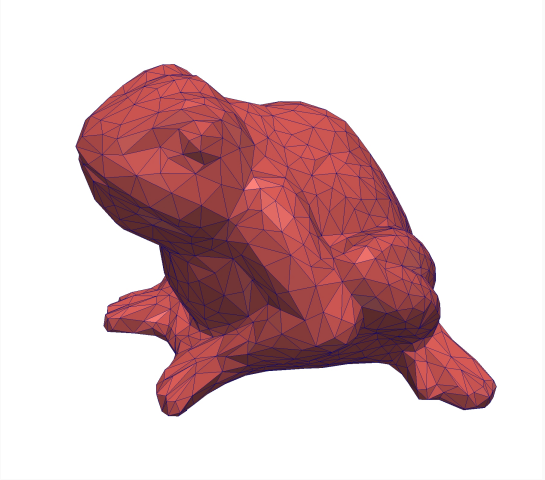




Observation

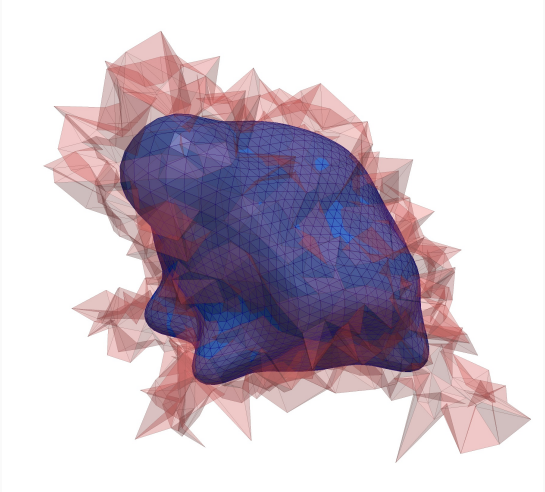


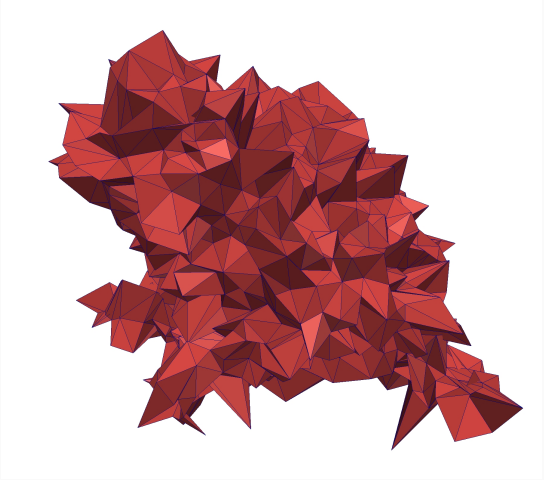
Reconstruction



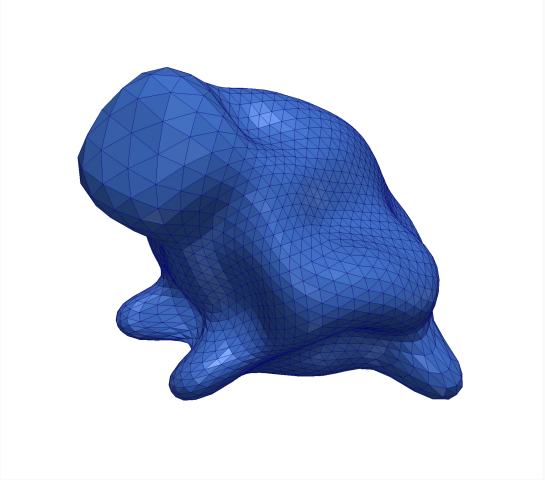
True shape

Varifold norm are robust to noise

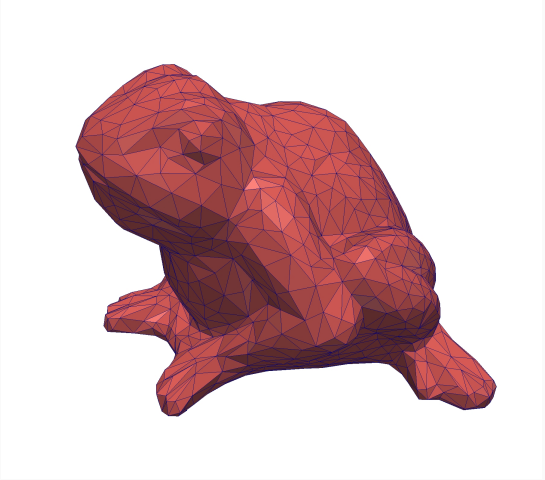




Observation

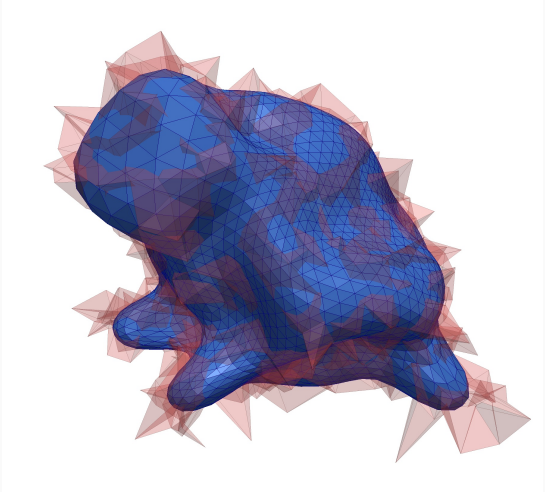


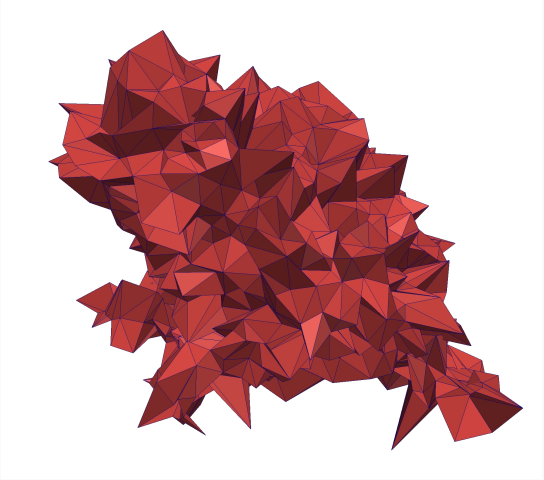
Reconstruction



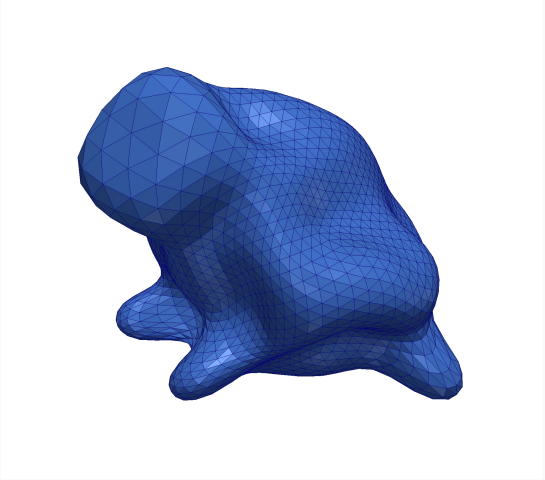
True shape

Varifold norm are robust to noise

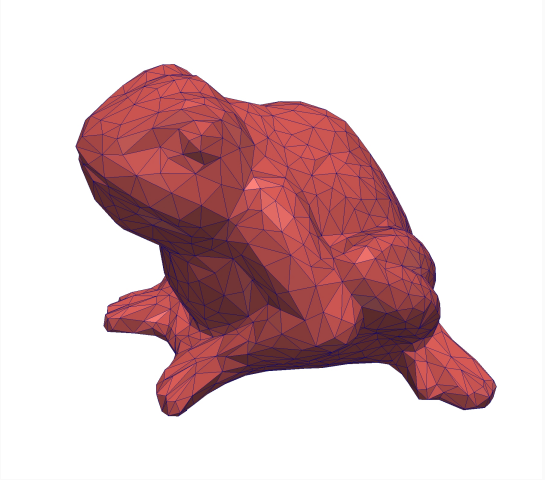




Observation

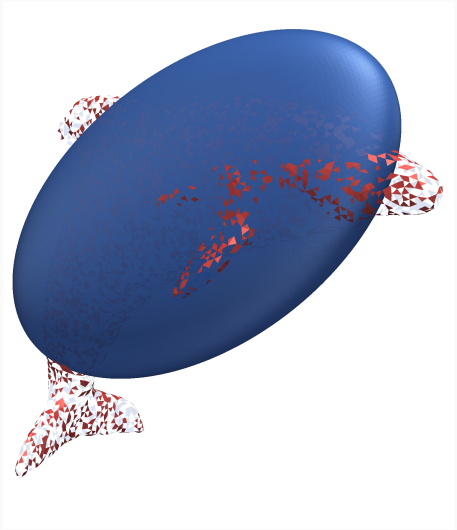


Reconstruction

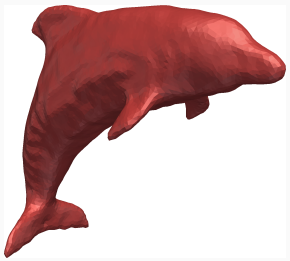


True shape

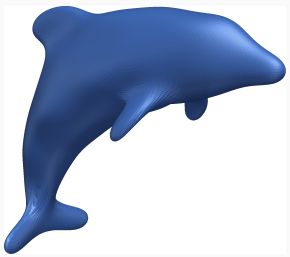
Varifold norm are robust to missing data

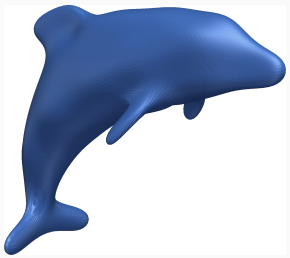


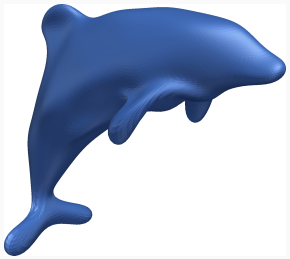
Observation



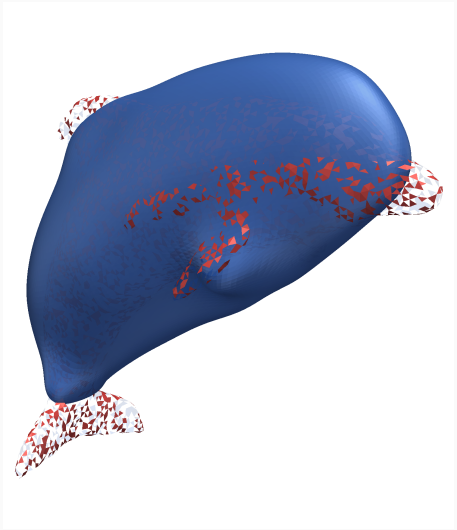
Reconstruction



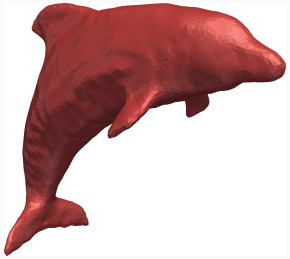


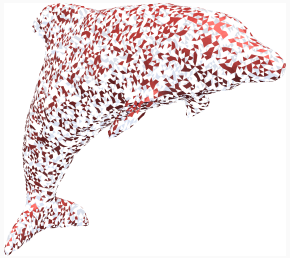


Varifold norm are robust to missing data

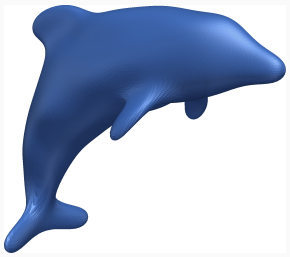


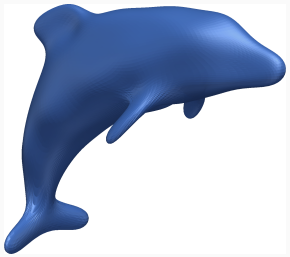
Observation

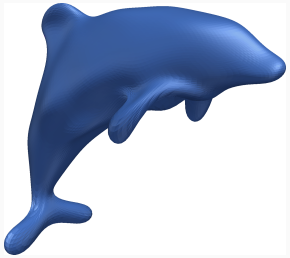




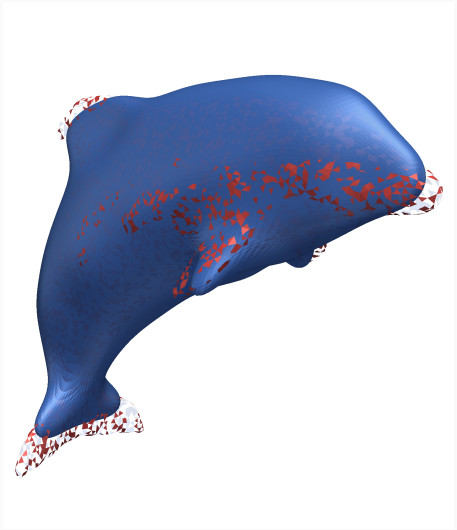
Reconstruction



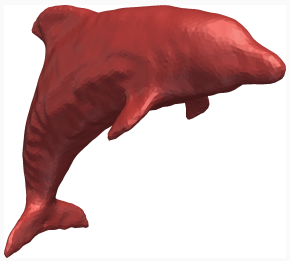




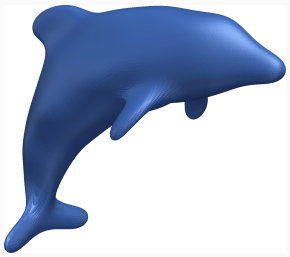
Varifold norm are robust to missing data

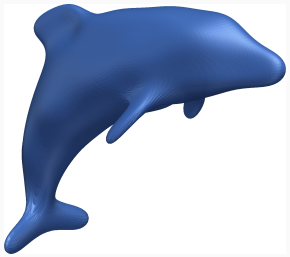


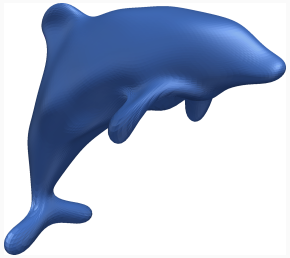
Observation



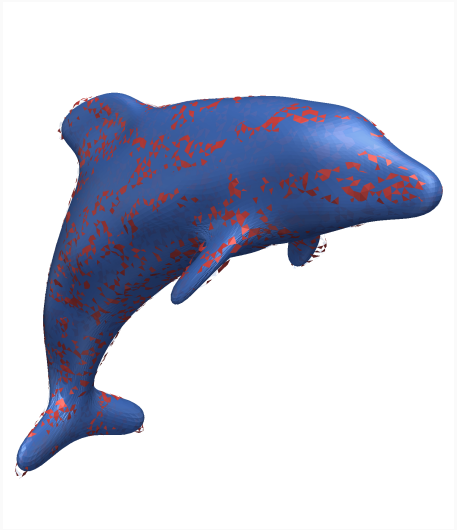
Reconstruction



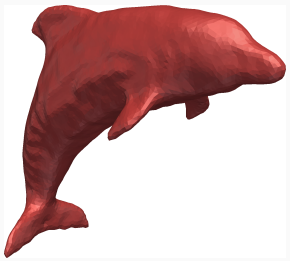




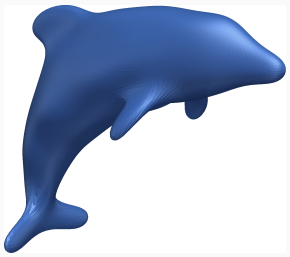
Varifold norm are robust to missing data

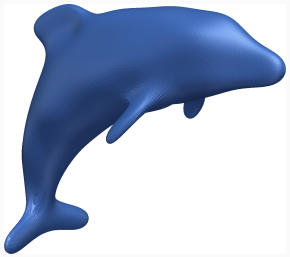


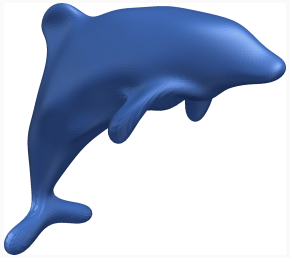
Observation



Reconstruction





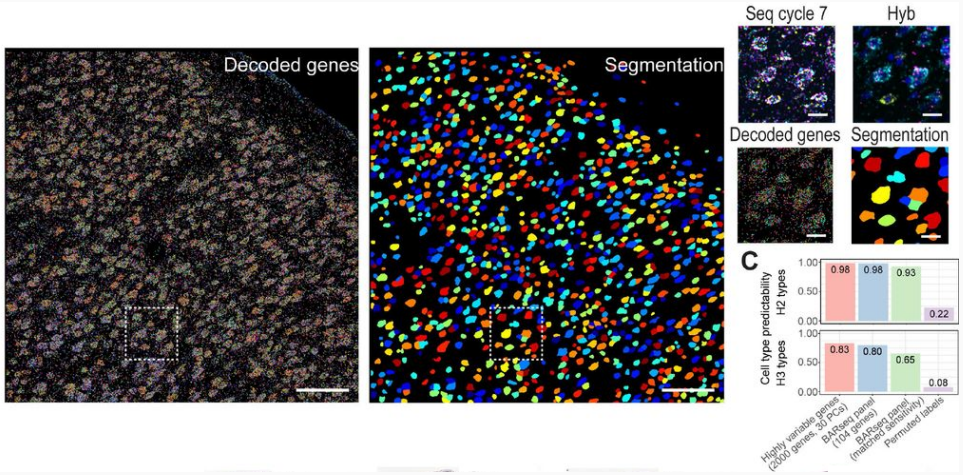


Generic varifold framework for transcriptomic data

A **single read** is a Dirac mass in a product space (location, feature) at:

- location: $x \in \mathbb{R}^d$. Typically $d = 2, 3$
- feature distribution: $wp \in \mathcal{M}(F)$, where $w \geq 0$ is a weight and p is a probability distribution over feature space F. Typically $F = \mathcal{F}, \mathcal{L}$.

- **BARseq**: $\mathcal{F}$ is the set of cell type ($|\mathcal{F}| \sim 30$)
  - w is total cells at location x
  - $p \in \mathcal{M}(\mathcal{F})$ is the probability distribution on cell type.

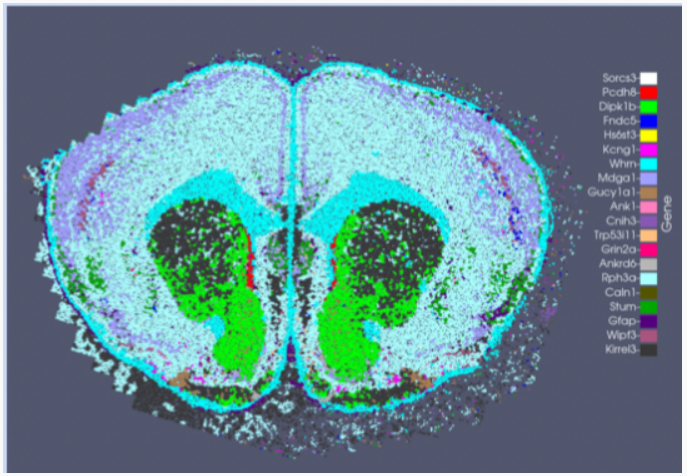


Generic varifold framework for transcriptomic data

A **single read** is a Dirac mass in a product space (location, feature) at:

- location: $x \in \mathbb{R}^d$. Typically $d = 2, 3$
- feature distribution: $wp \in \mathcal{M}(F)$, where $w \geq 0$ is a weight and p is a probability distribution over feature space F. Typically $F = \mathcal{F}, \mathcal{L}$.

- **MERFISH**: $\mathcal{F}$ is the set of gene type ($|\mathcal{F}| \sim 700$)
  - w is total mRNA at location x,
  - $p \in \mathcal{M}(\mathcal{F})$ is the probability distribution on gene.



Generic varifold framework for transcriptomic data

A **single read** is a Dirac mass in a product space (location, feature) at:

- location: $x \in \mathbb{R}^d$. Typically $d = 2, 3$
- feature distribution: $wp \in \mathcal{M}(F)$, where $w \geq 0$ is a weight and p is a probability distribution over feature space F. Typically $F = \mathcal{F}, \mathcal{L}$.

- **CCFv3 atlas**: $\mathcal{L}$ ontology labels
  - $w = 1$ for location x in foreground tissue
  - $p = \delta_{\ell_x} \in \mathcal{M}(\mathcal{L})$ is the dirac probability distribution on ontology label at the label ℓ_x of x (with $|\mathcal{L}| \sim 500$).

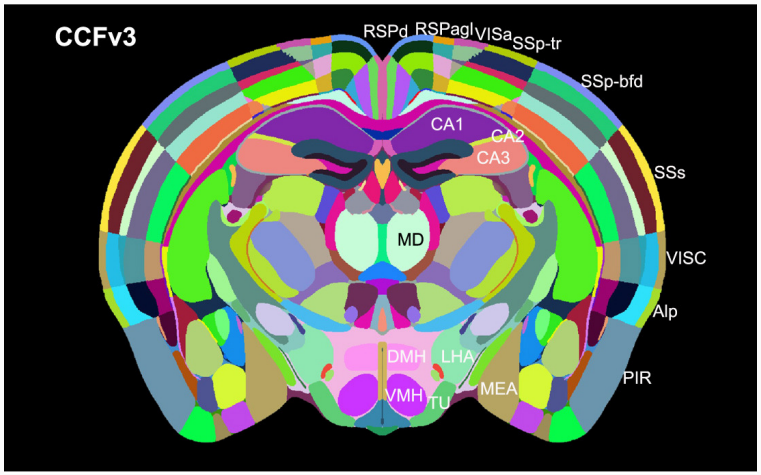


Image varifold framework

- **Image Varifold:** Full acquisition is a linear combination of Dirac indexed by $i \in I$:

$$\mu = \sum_{i \in I} \delta_{x_i} \otimes w_i p_i.$$

  with varifold norm

$$\langle \mu, \mu \rangle_M = \sum_{i,j \in I} w_i w_j K_\sigma(x_i, x_j) \sum_{f,g \in \mathcal{F}} K_F(f,g) p_i(f) p_j(g).$$

  where K_σ is spatial kernel (Gaussian), K_F is a def pos matrix (identity)

- **Computational intensity:** Depending on application we have:
  - $|I| \sim 10^4, 10^5, 10^6, 10^7$ (resolution)
  - $|\mathcal{F}|, |\mathcal{L}| \sim 10, 100, 1000$ (feature size)

  Resampling adjust data resolution to kernel bandwidth

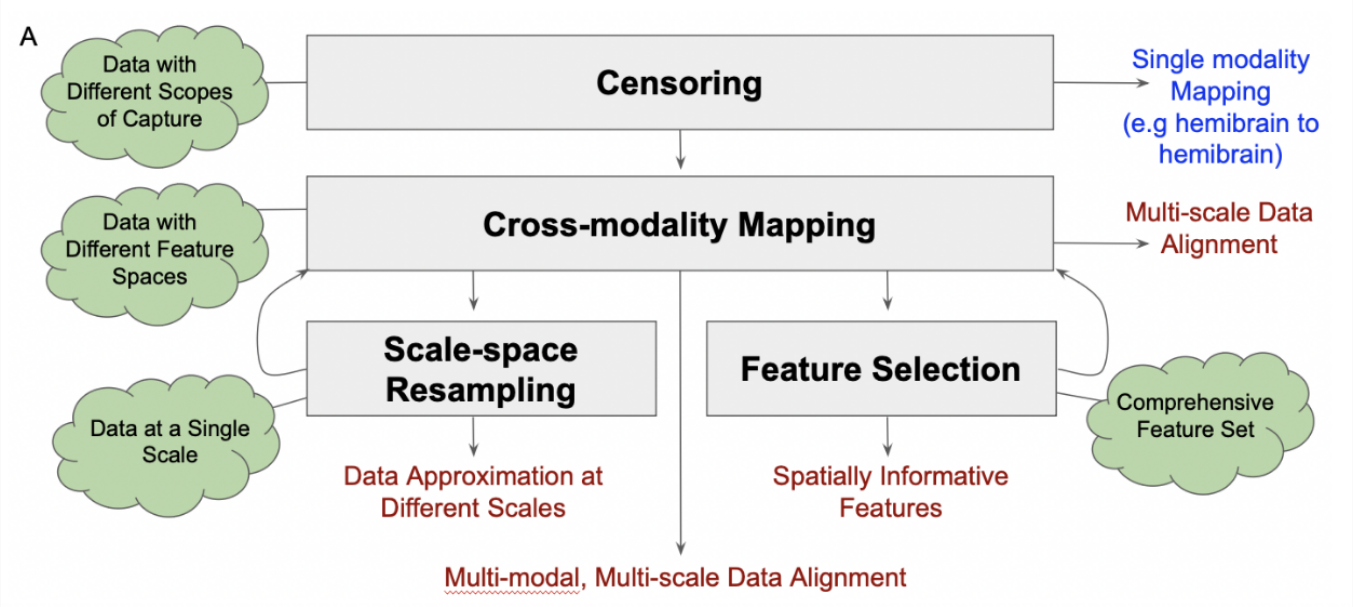
- **Cross modality:** allow us to define distance between objects in the same varifold space. But what's happened when the feature spaces are different ?

Non-Rigid Deformation with LDDMM

Varifold norms

Cross modalities Mapping Using Varifolds

Overview of the xIV-LDDMM toolkit



- Green: input. Red: output. Gray: technologies

Single modality Registration

Source and target: MERFISH with location and gene type $\mathcal{F}$ at molecular scale (feature)

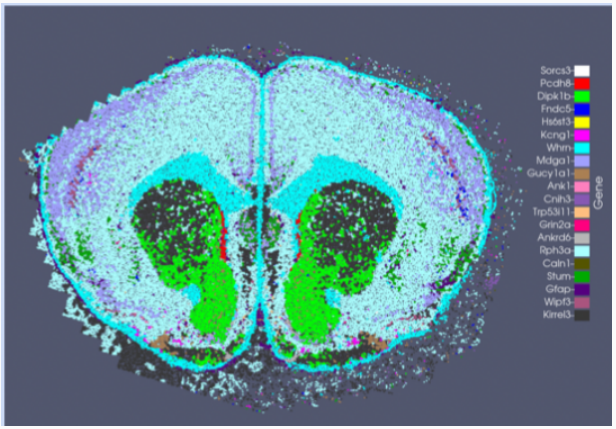
Single modality Registration

Source and target: MERFISH with location and gene type $\mathcal{F}$ at molecular scale (feature)

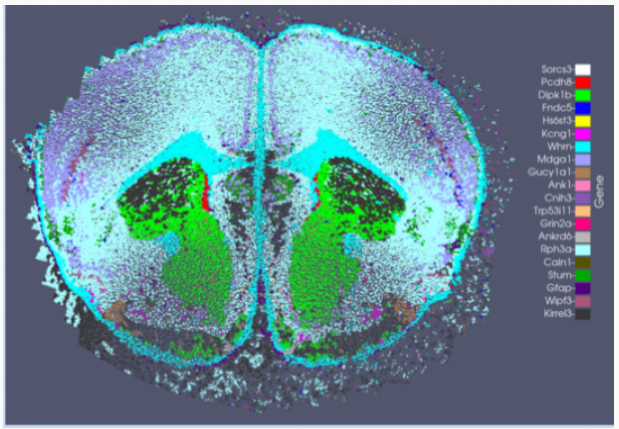
The single modality spatial deformation $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ acts as

$$\varphi \cdot \mu = \sum_{i \in I} \delta_{\varphi(x_i)} \otimes \left(|D\varphi|_{x_i} w_i \right) p_i,$$

where determinant of the Jacobian capture expansion/contraction.



μ



$\varphi \cdot \mu$

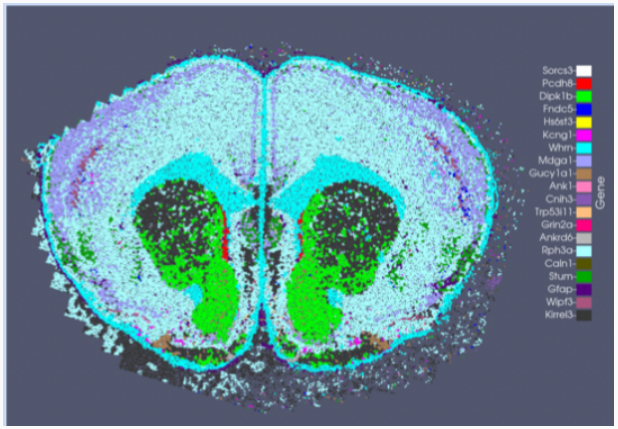
Single modality Registration

Source and target: MERFISH with location and gene type $\mathcal{F}$ at molecular scale (feature)

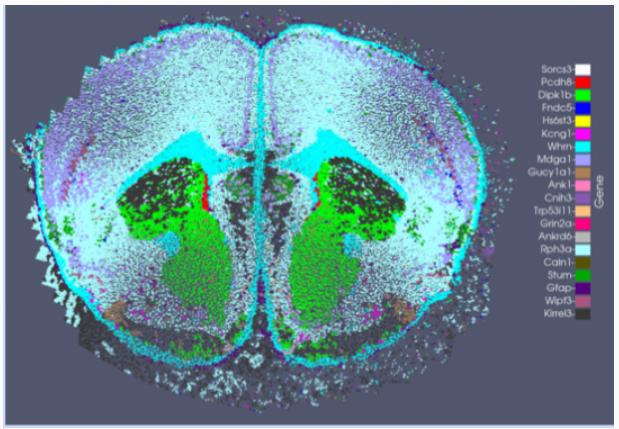
The single modality spatial deformation $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ acts as

$$\varphi \cdot \mu = \sum_{i \in I} \delta_{\varphi(x_i)} \otimes \left(|D\varphi|_{x_i} w_i \right) p_i,$$

where determinant of the Jacobian capture expansion/contraction.



μ



$\varphi \cdot \mu$

Minimize $pen(\varphi) + \|\varphi \cdot \mu_{Source} - \mu_{Target}\|_M^2$ with respect to

- Spatial correspondence: $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, an affine motion and diffeomorphism of $\mathbb{R}^d$
- Hamiltonian formulation (Geodesic shooting) is adapted to update the weight

Cross modality Registration

Source: Atlas with location and atlas ontology $\mathcal{L}$ at tissue scale (feature)

Target: MERFISH with location and gene type $\mathcal{F}$ at molecular scale (feature)

Spatial homogeneity assumption: there exists a (latent) dictionary $(\pi_\ell)_{\ell \in \mathcal{L}}$ where each $\pi_\ell \in \mathcal{M}(\mathcal{F})$.

Cross modality Registration

Source: Atlas with location and atlas ontology $\mathcal{L}$ at tissue scale (feature)

Target: MERFISH with location and gene type $\mathcal{F}$ at molecular scale (feature)

Spatial homogeneity assumption: there exists a (latent) dictionary $(\pi_\ell)_{\ell \in \mathcal{L}}$ where each $\pi_\ell \in \mathcal{M}(\mathcal{F})$.

The *cross modality spatial deformation* (φ, π) acts as

$$(\varphi, \pi) \cdot \mu^A = (\varphi, \pi) \cdot \sum_{i \in I^A} \delta_{x_i} \otimes \underbrace{w_i^A p_i^A}_{\in \mathcal{M}(\mathcal{L})}$$

Remember that since μ^A is an atlas, $w_i^A = 1$ and p_i is a Dirac at ℓ_{x_i} ("one hot").

$$\begin{aligned}(\varphi, \pi) \cdot \mu^A &\doteq \varphi \cdot \sum_{i \in I^A} \delta_{x_i} \otimes \underbrace{\pi_{\ell_{x_i}}}_{\in \mathcal{M}(\mathcal{F})} \\ &= \sum_{i \in I} \delta_{\varphi(x_i)} \otimes \underbrace{|D\varphi|_{x_i} \pi_{\ell_{x_i}}}_{\in \mathcal{M}(\mathcal{F})}.\end{aligned}$$

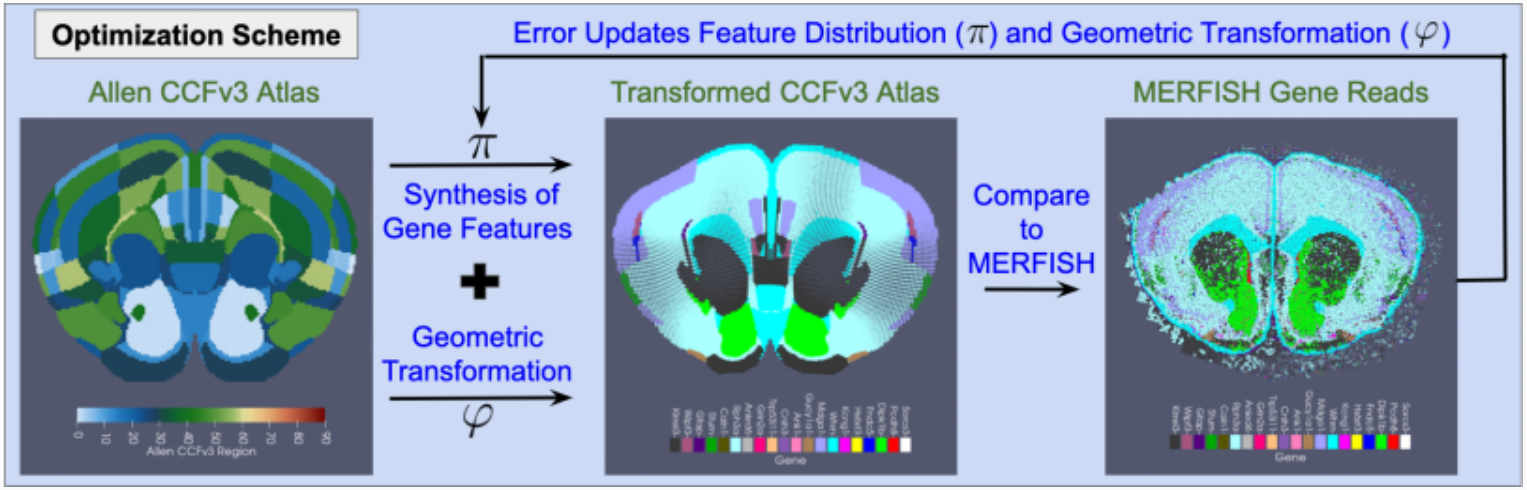
where determinant of the Jacobian capture expansion/contraction.

Warning: notation switch between the 2 papers...

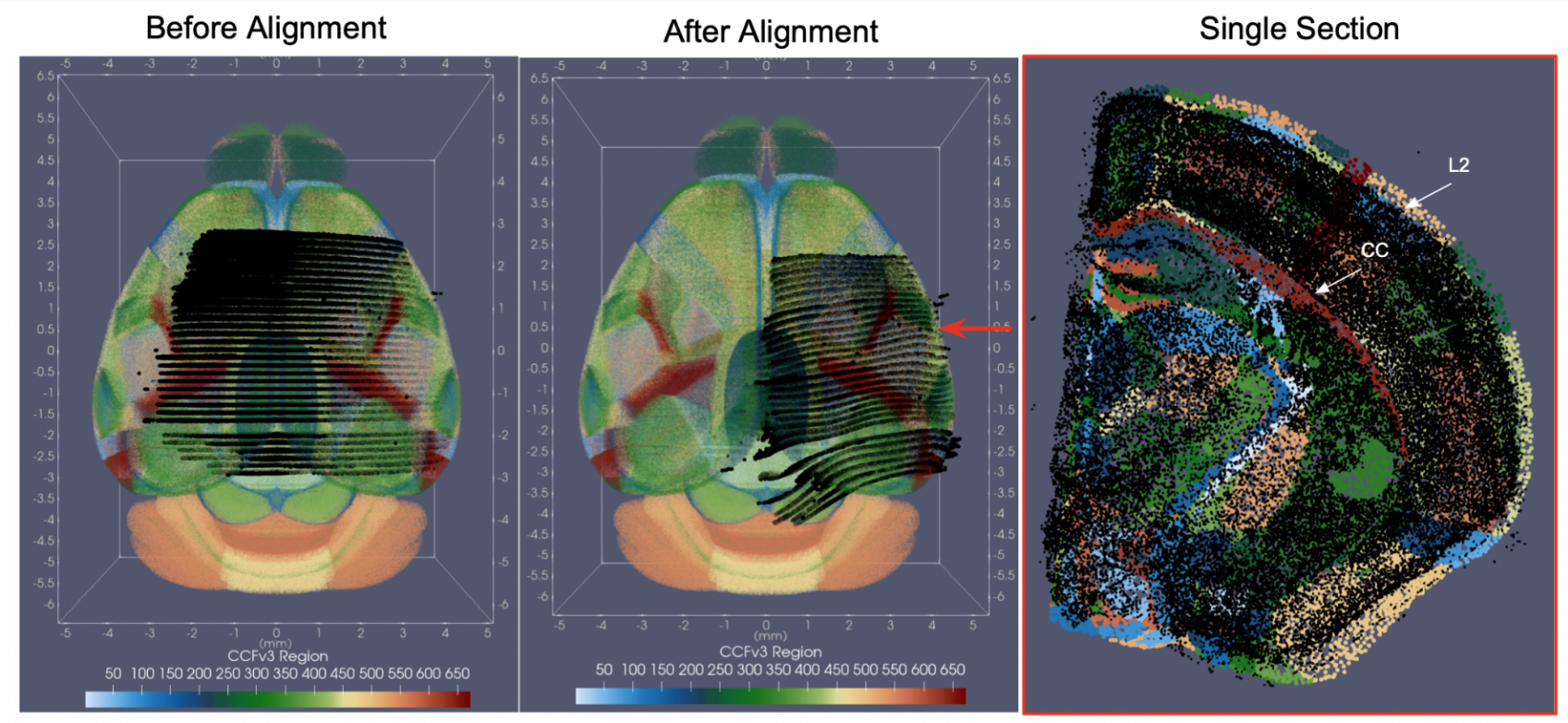
Deformations of varifolds

Minimize $pen(\varphi) + pen(\pi) + \|(\varphi, \pi) \cdot \mu^A - \mu^{Target}\|_M^2$ with respect to

- Spatial correspondence: $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, an affine motion and diffeomorphism of $\mathbb{R}^d$
- Feature correspondence: $(\pi_\ell)_{\ell \in \mathcal{L}}$ where each $\pi_\ell \in \mathcal{M}(\mathcal{F})$ is a (latent) distribution over $\mathcal{F}$ which should be similar to the $w_i p_i$'s (of the target) in region ℓ.
- KL penalty: $pen(\pi_\ell) = \frac{M_\ell^A}{\sum_{f \in \mathcal{F}} M_f^{Target}} \sum_{f \in \mathcal{F}} \pi_\ell(f) \log\left(\frac{\bar{\pi}_\ell(f)}{1/|\mathcal{F}|}\right)$ where M_ℓ^A (resp. M_f^{Target}) is total mass in region ℓ (resp. feature f), $\bar{\pi}_\ell(f) \doteq \frac{\pi_\ell(f)}{\sum_{f \in \mathcal{F}} \pi_\ell(f)}$.

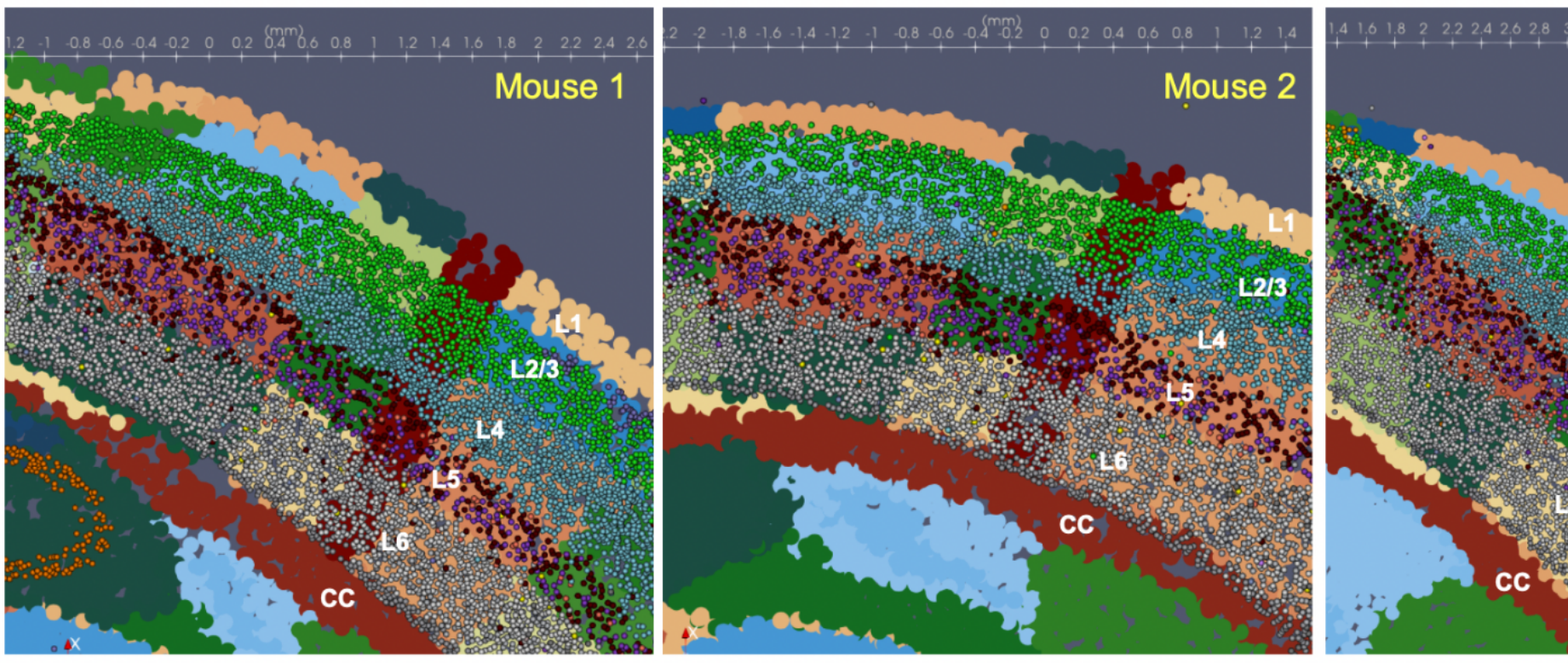


CCFv3 and BARseq: Global Geometric Alignment



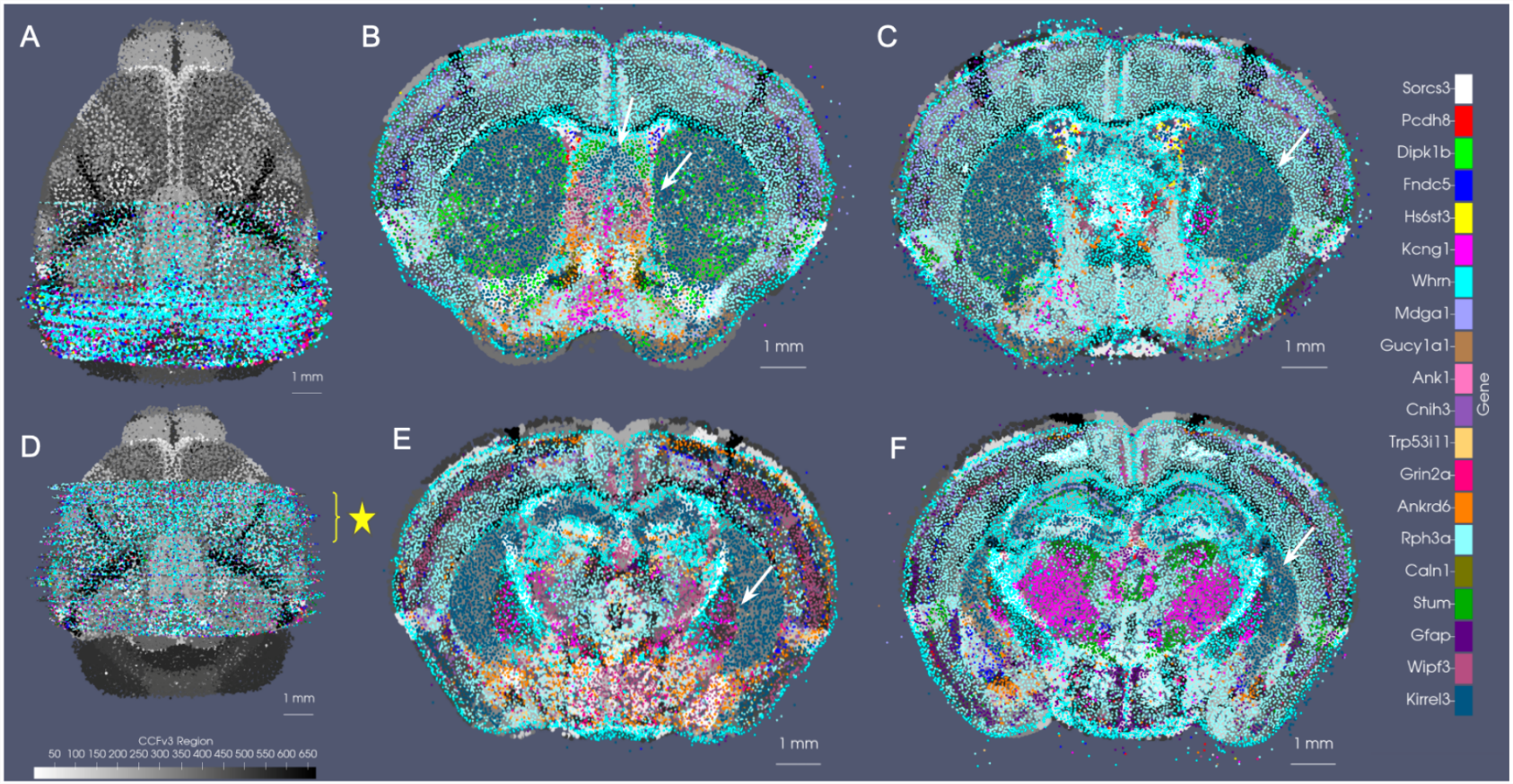
- Black dots: ∼ 30 coronal hemi-sections of BARseq spatial transcriptomics data
- Regions denoted by color CCFv3
- Good overlap of low cell density area (BARseq) with CCFv3 corpus callosum (CC), and layer 2/3 cells (BARseq) with CCFv3 layer 2/3.

CCFv3 and BARseq: : Local Geometric Alignment



- Small spheres: BARseq cell center colored according to layer-specific cell type (L2/3 (green), L4/5 (blue), L5 (purple), L6 (grey))
- Plain circles color: CCFv3 Region
- Boundaries between cell types align to cortical layer delineations in the CCFv3, and both corpus callosum (CC) and layer 1 (L1) accurately align with low cell density areas.

CCFv3 and MERFISH



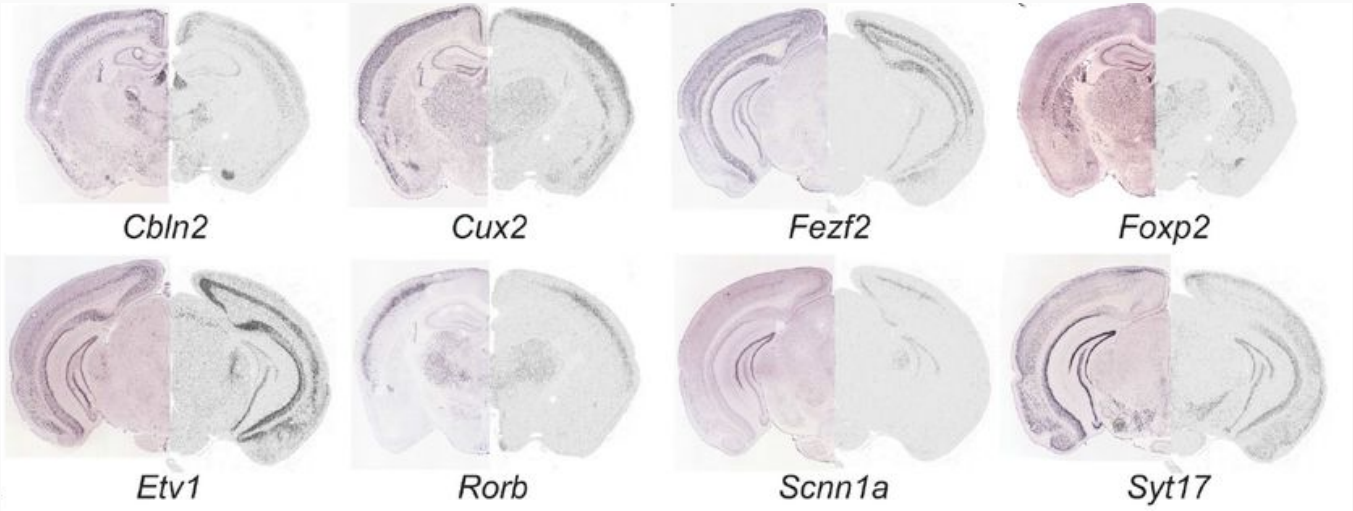
20 selected variable genes. Resolution is 200μm

User Difficulties

- Understanding the basic principles of kernel-based LDDMM methods
- Calibrating the set of parameters
- Assessing the quality of a deformation

Cross-Modality Data Comparison

- **Task:** Compare different spatial scales (tissue-level vs. molecular-level) and feature types (anatomical ontology vs. gene expression) ...



"The expression patterns of representative genes in Allen Brain Atlas (left half) compared to the current dataset (right half)."

X. Chen et al. *Whole-cortex in situ sequencing reveals input-dependent area identity.* Nature, 2014.

- **Goal:** Automate the comparison and quantify similarity across data modalities.
- **Idea:** Embed all data types (transcriptomics and atlases) into a shared (kernel) varifold space for unified analysis.