Global divergences between measures: from Hausdorff distance to Optimal Transport

Jean Feydy   Alain Trouvé

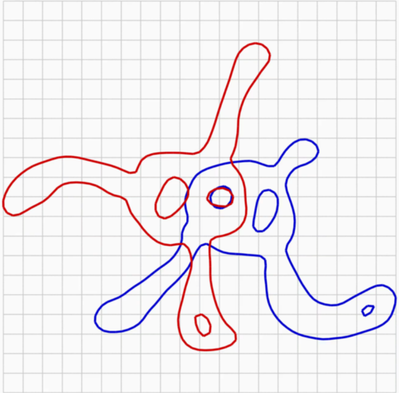
ShapeMI workshop, Miccai, Granada – 20th September, 2018

Écoles Normales Supérieures de Paris et Paris-Saclay

Collaboration with B. Charlier, J. Glaunès (KeOps library);
S.-i. Amari, G. Peyré, T. Séjourné, F.-X. Vialard (OT theory)

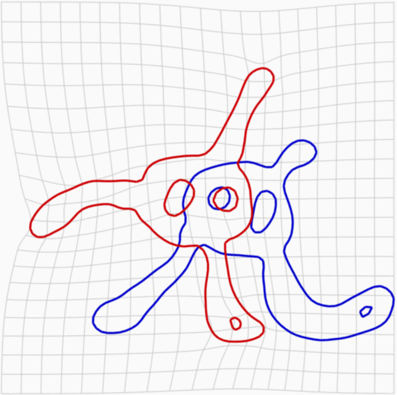
Today: focus on shape registration

Source A, target B,



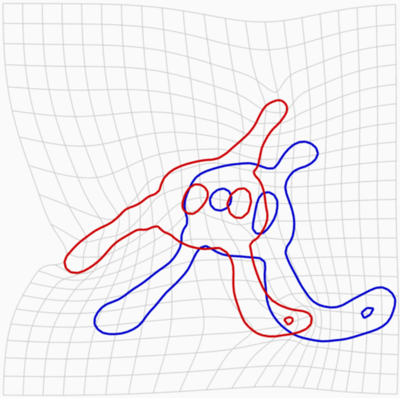
Today: focus on shape registration

Source A, target B, mapping φ



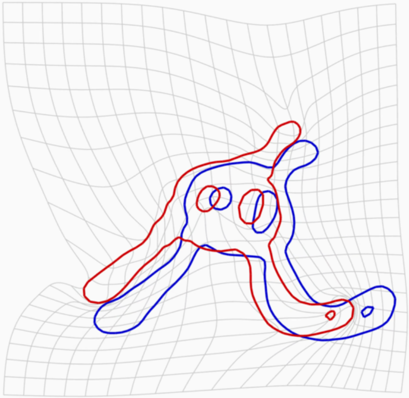
Today: focus on shape registration

Source A, target B, mapping φ



Today: focus on shape registration

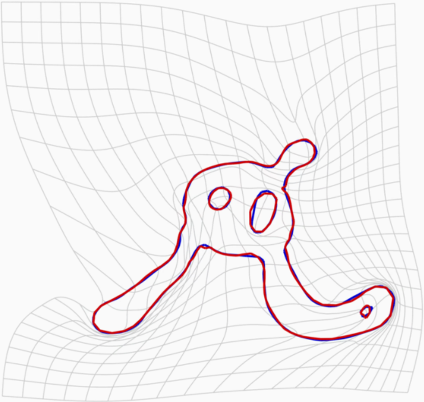
Source A, target B, mapping φ



Today: focus on shape registration

Source A, target B, mapping φ

$$A \xrightarrow[\text{Model}]{\varphi} \varphi(A) = A' \underset{\textbf{Loss}}{\rightleftarrows} B$$



A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

$A' \leftarrow A$
repeat
$\quad L,\ v \leftarrow \text{Loss}(A', B),\ -\partial_{A'}\text{Loss}(A', B)$
$\quad A' \leftarrow A' + \text{Model}(v)$
until $L < \text{tol}$
Output : deformed shape $A' = \varphi(A)$.

A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

1: $A' \leftarrow A$
2: **repeat**
3: $\quad L\,,\, v \leftarrow \text{Loss}(A', B)\,,\ -\partial_{A'}\text{Loss}(A', B)$
4: $\quad A' \leftarrow A' + \text{Model}(v)$
5: **until** $L < \text{tol}$
Output : deformed shape $A' = \varphi(A)$.

“Model” encodes the **prior knowledge** on admissible deformations:

- *smoothing* convolution
- LDDMM/SVF *backprop* + regularization + *shooting*
- *trained* neural network

A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

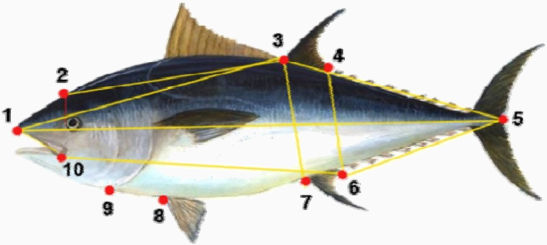
1: $A' \leftarrow A$
2: **repeat**
3: $\quad L\,,\, v \;\leftarrow\; \text{Loss}(A', B)\,,\;\; -\partial_{A'}\text{Loss}(A', B)$
4: $\quad A' \;\leftarrow\; A' \;+\; \text{Model}(\,v\,)$
5: **until** $\;L \;<\; \text{tol}$
Output : deformed shape $\;A' \;=\; \varphi(\,A\,)$.

"Model" encodes the **prior knowledge** on admissible deformations:

- *smoothing* convolution
- LDDMM/SVF *backprop* + regularization + *shooting*
- *trained* neural network

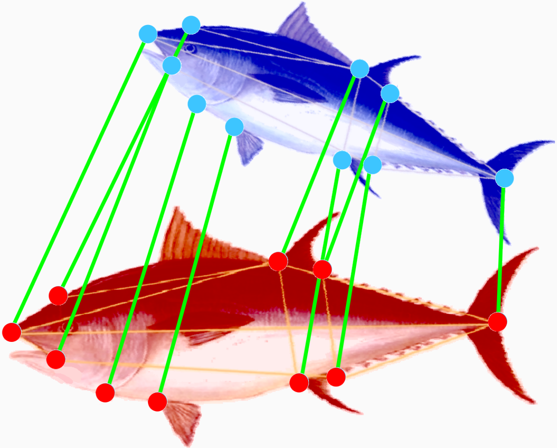
$\Rightarrow$ The *raw* Loss gradient v is what **drives** the registration

On labeled shapes, use a spring energy



Anatomical landmarks from *A morphometric approach for the analysis of body shape in bluefin tuna*, Addis et al., 2009.

On labeled shapes, use a spring energy



Anatomical landmarks from *A morphometric approach for the analysis of body shape in bluefin tuna*, Addis et al., 2009.

Encoding unlabeled shapes as measures

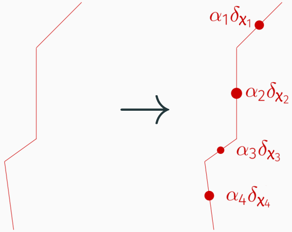
Let's enforce sampling invariance:

$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}.$$

Encoding unlabeled shapes as measures

Let's enforce sampling invariance:

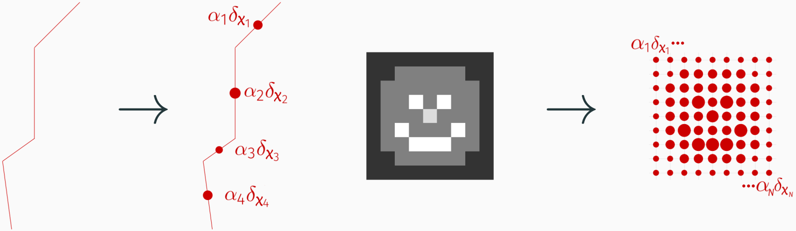
$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$



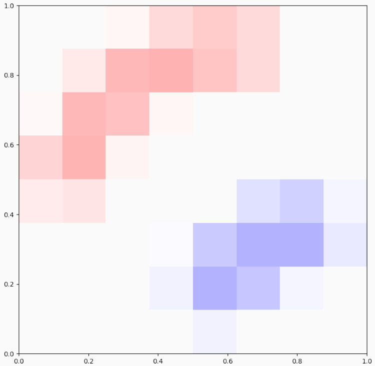
Encoding unlabeled shapes as measures

Let's enforce sampling invariance:

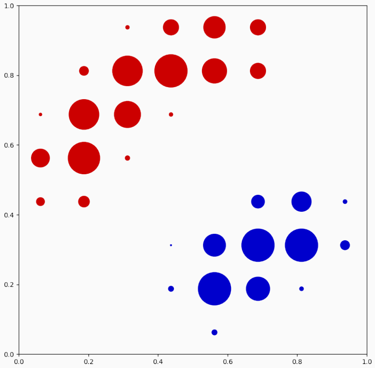
$$\color{red}{A} \longrightarrow \color{red}{\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}}\,, \qquad \color{blue}{B} \longrightarrow \color{blue}{\beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}}\,.$$



A baseline setting: density registration

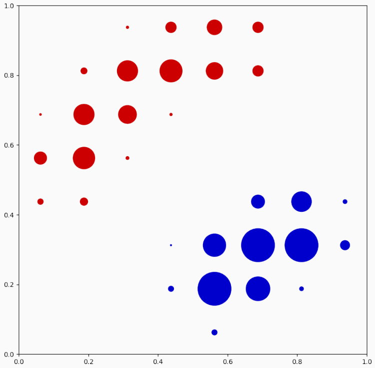


A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

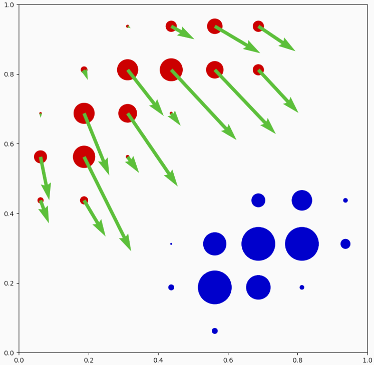
A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i} , \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j} .$$

$$\sum_{i=1}^{N} \alpha_i = 1 = \sum_{j=1}^{M} \beta_j$$

A baseline setting: density registration

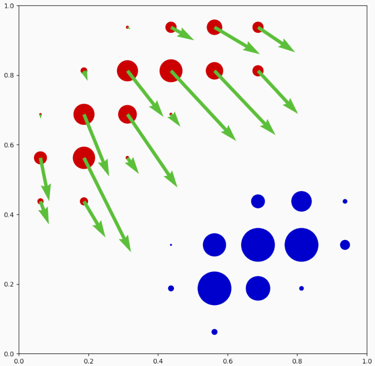


$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

$$\sum_{i=1}^{N} \alpha_i = 1 = \sum_{j=1}^{M} \beta_j$$

Display $v = -\nabla_{x_i} \mathrm{d}(\alpha, \beta)$.

A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

$$\sum_{i=1}^{N} \alpha_i \;=\; 1 \;=\; \sum_{j=1}^{M} \beta_j$$

Display $\nu \;=\; -\nabla_{x_i} \mathrm{d}(\alpha, \beta)$.

Seamless extensions to:

- $\sum_i \alpha_i \neq \sum_j \beta_j$ [Chizat et al., 2018],
- curves and surfaces [Kaltenmark et al., 2017],
- variable weights α_i.

Computing fidelities between **measures**:

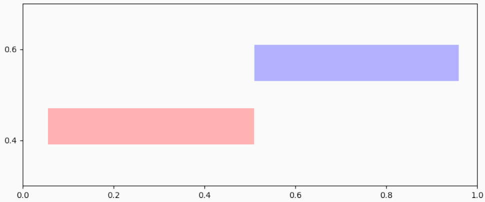
1. **Computer graphics**: Hausdorff distance
2. **Statistics**: kernel distances
3. **Optimal Transport**: Wasserstein distance

Overview

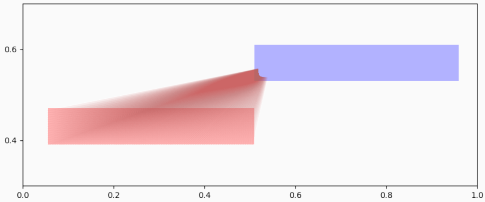
Computing fidelities between **measures**:

1. **Computer graphics:** Hausdorff distance
2. **Statistics:** kernel distances
3. **Optimal Transport:** Wasserstein distance
4. Efficient GPU routines: **KeOps**

The weighted Hausdorff distance



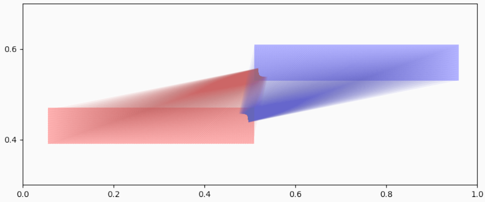
The weighted Hausdorff distance



p-Hausdorff distance:

$$\text{Loss}(\alpha, \beta) \quad = \quad \tfrac{1}{2} \sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p$$

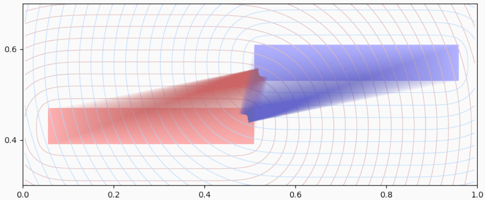
The weighted Hausdorff distance



p-Hausdorff distance:

$$\text{Loss}(\alpha, \beta) \;=\; \tfrac{1}{2}\sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p \;+\; \tfrac{1}{2}\sum_j \beta_j \cdot \min_i \|x_i - y_j\|^p$$

The weighted Hausdorff distance

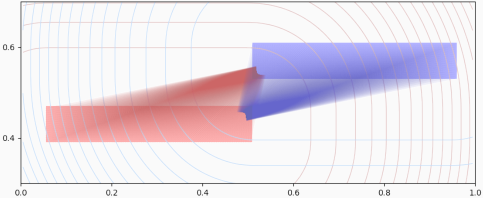


p-Hausdorff distance:

$$\begin{aligned}\text{Loss}(\alpha,\beta) &= \tfrac{1}{2}\textstyle\sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p &&+ \tfrac{1}{2}\textstyle\sum_j \beta_j \cdot \min_i \|x_i - y_j\|^p \\ &= \tfrac{1}{2}\langle\, \alpha \,,\, b \,\rangle &&+ \tfrac{1}{2}\langle\, \beta \,,\, a \,\rangle\end{aligned}$$

$$\text{with} \qquad \begin{aligned} a(x) &= \mathrm{d}(\,x\,,\, \mathrm{supp}(\alpha))^p \\ b(x) &= \mathrm{d}(\,x\,,\, \mathrm{supp}(\beta))^p \end{aligned}$$

The weighted Hausdorff distance

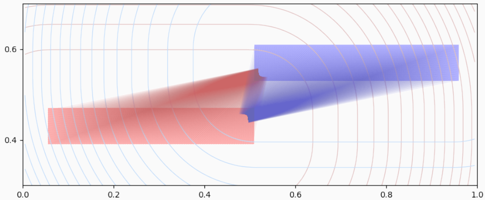


p-Hausdorff distance:

$$
\begin{aligned}
\text{Loss}(\alpha, \beta) &= \tfrac{1}{2} \textstyle\sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p \quad + \quad \tfrac{1}{2} \textstyle\sum_j \beta_j \cdot \min_i \|x_i - y_j\|^p \\
&= \tfrac{1}{2} \langle\, \alpha \,,\, b \,\rangle \quad + \quad \tfrac{1}{2} \langle\, \beta \,,\, a \,\rangle
\end{aligned}
$$

$$
\text{with} \qquad
\begin{aligned}
a(x) &= \mathrm{d}(\, x \,,\, \mathrm{supp}(\alpha))^p \\
b(x) &= \mathrm{d}(\, x \,,\, \mathrm{supp}(\beta))^p
\end{aligned}
$$

The weighted Hausdorff distance

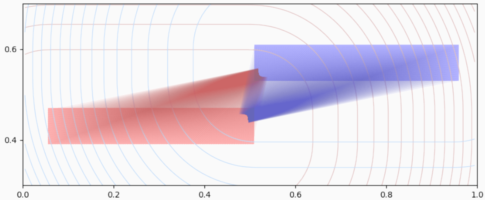


p-Hausdorff distance:

$$
\begin{aligned}
\text{Loss}(\alpha,\beta) &= \tfrac{1}{2}\textstyle\sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p &+& \quad \tfrac{1}{2}\textstyle\sum_j \beta_j \cdot \min_i \|x_i - y_j\|^p \\
&= \tfrac{1}{2}\langle\, \alpha\,,\, b\,\rangle &+& \quad \tfrac{1}{2}\langle\, \beta\,,\, a\,\rangle \\
&= \tfrac{1}{2}\langle\, \alpha\,,\, b - a\,\rangle &+& \quad \tfrac{1}{2}\langle\, \beta\,,\, a - b\,\rangle
\end{aligned}
$$

$$
\text{with} \qquad
\begin{aligned}
a(x) &= \mathrm{d}(\,x\,,\, \text{supp}(\alpha))^p \\
b(x) &= \mathrm{d}(\,x\,,\, \text{supp}(\beta))^p
\end{aligned}
$$

The weighted Hausdorff distance



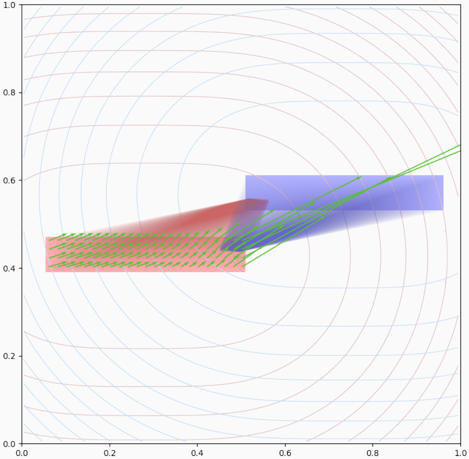
p-Hausdorff distance:

$$\begin{aligned}
\text{Loss}(\alpha,\beta) &= \tfrac{1}{2}\textstyle\sum_i \alpha_i \cdot \min_j \|x_i - y_j\|^p &&+ \tfrac{1}{2}\textstyle\sum_j \beta_j \cdot \min_i \|x_i - y_j\|^p \\
&= \tfrac{1}{2}\langle\, \alpha\,,\, b\,\rangle &&+ \tfrac{1}{2}\langle\, \beta\,,\, a\,\rangle \\
&= \tfrac{1}{2}\langle\, \alpha\,,\, b-a\,\rangle &&+ \tfrac{1}{2}\langle\, \beta\,,\, a-b\,\rangle \\
&= \tfrac{1}{2}\langle\, \alpha-\beta\,,\, b-a\,\rangle
\end{aligned}$$

$$\text{with} \qquad \begin{aligned} a(x) &= \mathrm{d}(\,x\,,\,\mathrm{supp}(\alpha))^p \\ b(x) &= \mathrm{d}(\,x\,,\,\mathrm{supp}(\beta))^p \end{aligned}$$

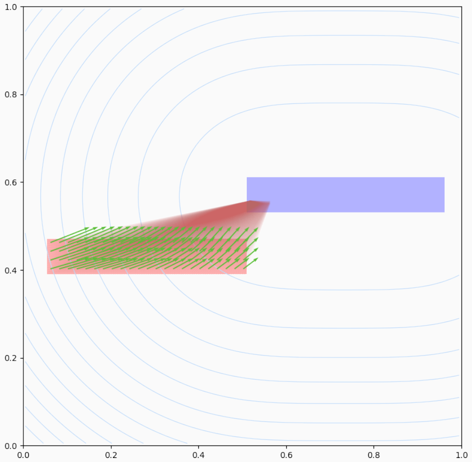
Naive projections in Hausdorff cause imbalance

$$\mathrm{Loss}(\alpha,\beta) \quad = \quad \tfrac{1}{2}\langle\, \alpha,\ b - a \,\rangle \ + \ \tfrac{1}{2}\langle\, \beta,\ a - b \,\rangle$$



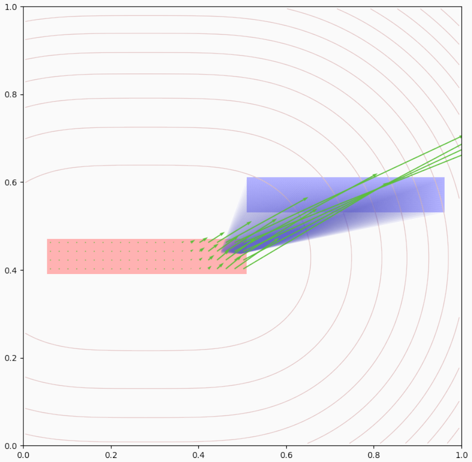
Naive projections in Hausdorff cause imbalance

$$\text{Loss}(\alpha, \beta) \quad = \quad \tfrac{1}{2}\langle\, \alpha,\ b - a \,\rangle \ + \ \tfrac{1}{2}\langle\, \beta,\ a - b \,\rangle$$



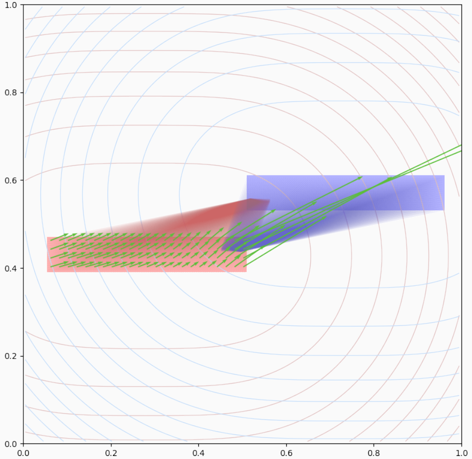
Naive projections in Hausdorff cause imbalance

$$\text{Loss}(\alpha, \beta) \quad = \quad \tfrac{1}{2}\langle\, \alpha,\ b - a \,\rangle \ + \ \tfrac{1}{2}\langle\, \beta,\ a - b \,\rangle$$



Naive projections in Hausdorff cause imbalance

$$\text{Loss}(\alpha, \beta) \quad = \quad \tfrac{1}{2}\langle\, \alpha,\ b - a \,\rangle \ + \ \tfrac{1}{2}\langle\, \beta,\ a - b \,\rangle$$



Kernel distances: distance fields computed through convolutions

Kernel distances, aka. **blurred** SSDs:

$$\text{choose} \qquad a(x) \;=\; -(k\star\alpha)(x) \;=\; -\sum_i \alpha_i\, k(x, x_i)$$

$$\text{and use} \qquad \tfrac{1}{2}\langle\, \alpha-\beta\,,\; b-a\,\rangle \;=\; \tfrac{1}{2}\langle\, \alpha-\beta\,,\; k\star(\alpha-\beta)\,\rangle.$$

Kernel distances: distance fields computed through convolutions

Kernel distances, aka. **blurred** SSDs:

$$\text{choose} \qquad a(x) \;=\; -(k \star \alpha)(x) \;=\; -\sum_i \alpha_i \, k(x, x_i)$$

$$\text{and use} \qquad \tfrac{1}{2}\langle\, \alpha - \beta \,,\, b - a \,\rangle \;=\; \tfrac{1}{2}\langle\, \alpha - \beta \,,\, k \star (\alpha - \beta) \,\rangle.$$

The **Energy Distance:** an underrated kernel, $k(x, y) \;=\; -\|x - y\|$.

$$a(x) \;=\; \sum_i \alpha_i \, \|x - x_i\| \qquad \text{instead of} \quad a(x) \;=\; \min_i \|x - x_i\|$$

$$b(x) \;=\; \sum_j \beta_j \, \|x - y_j\| \qquad \text{instead of} \quad b(x) \;=\; \min_j \|x - y_j\|.$$

Kernel distances: distance fields computed through convolutions

Kernel distances, aka. **blurred** SSDs:

$$\text{choose} \qquad a(x) \;=\; -(k\star\alpha)(x) \;=\; -\sum_i \alpha_i\, k(x,x_i)$$

$$\text{and use} \qquad \tfrac{1}{2}\langle\, \alpha-\beta\,,\; b-a\,\rangle \;=\; \tfrac{1}{2}\langle\, \alpha-\beta\,,\; k\star(\alpha-\beta)\,\rangle.$$

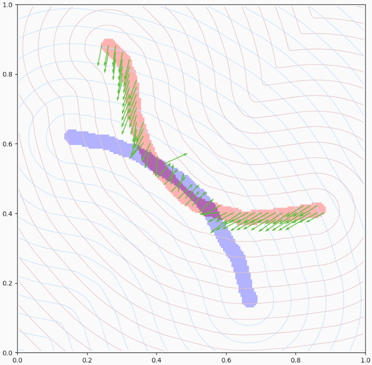
The **Energy Distance:** an underrated kernel, $k(x,y) \;=\; -\|x-y\|$.

$$a(x) \;=\; \sum_i \alpha_i\,\|x-x_i\| \qquad \text{instead of} \quad a(x) \;=\; \min_i \|x-x_i\|$$

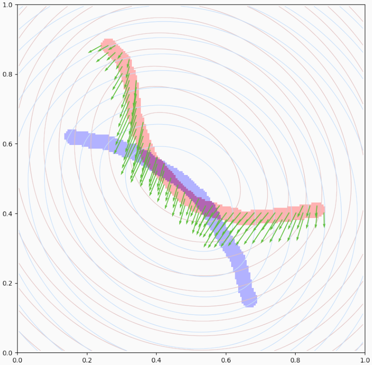
$$b(x) \;=\; \sum_j \beta_j\,\|x-y_j\| \qquad \text{instead of} \quad b(x) \;=\; \min_j \|x-y_j\|.$$

$$\begin{aligned}\text{Loss}(\alpha,\beta) \;=\;& \sum_i\sum_j \alpha_i\beta_j\,\|x_i-y_j\| \\ &-\;\tfrac{1}{2}\sum_i\sum_j \alpha_i\alpha_j\,\|x_i-x_j\| \;-\; \tfrac{1}{2}\sum_i\sum_j \beta_i\beta_j\,\|y_i-y_j\|\end{aligned}$$

The Hausdorff distance is local, the Energy Distance is global



Hausdorff, min



Kernel, $\sum$

An idea from Optimal Transport theory: Sinkhorn divergences

Optimal Transport = Hausdorff + mass spreading constraint

Computational OT [Cuturi, 2013, Peyré and Cuturi, 2018]:

Start from an ε-smoothed **Hausdorff** distance, but let the influence fields a and b **interact** with each other.

Enforce a **mass spreading** constraint on the spring system:

all of α should be linked to all of β.

Optimal Transport = Hausdorff + mass spreading constraint

Computational OT [Cuturi, 2013, Peyré and Cuturi, 2018]:

Start from an ε-smoothed **Hausdorff** distance, but let the influence fields a and b **interact** with each other.

Enforce a **mass spreading** constraint on the spring system:

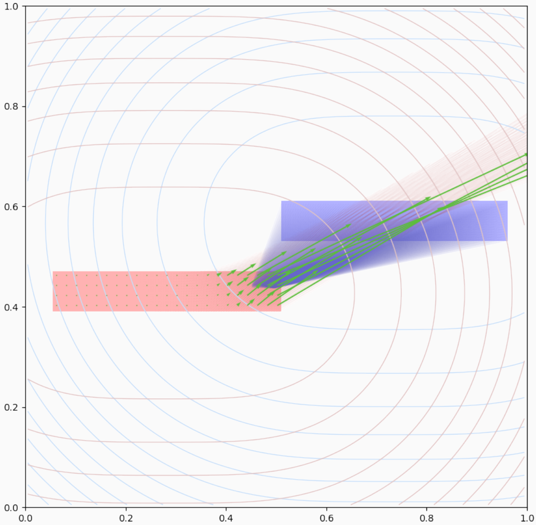
all of α should be linked to all of β.

In practice: use the 5-line **Sinkhorn** algorithm.

Updates a and b alternatively.

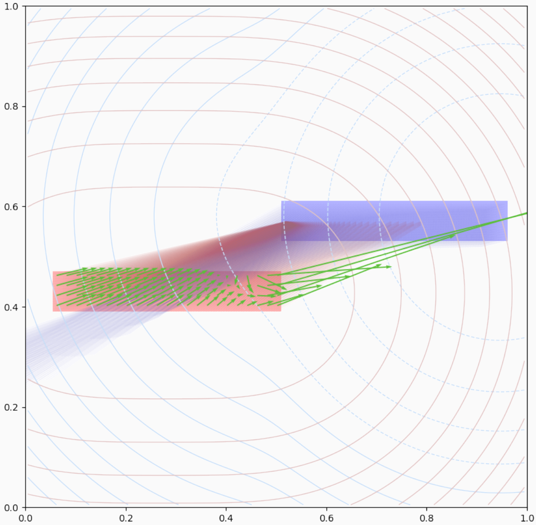
Converges in about 10-20 steps – x2 convolutions.

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



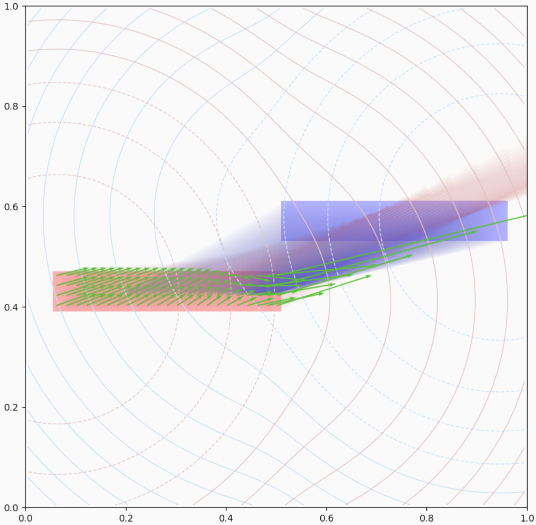
Iteration 1/2

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



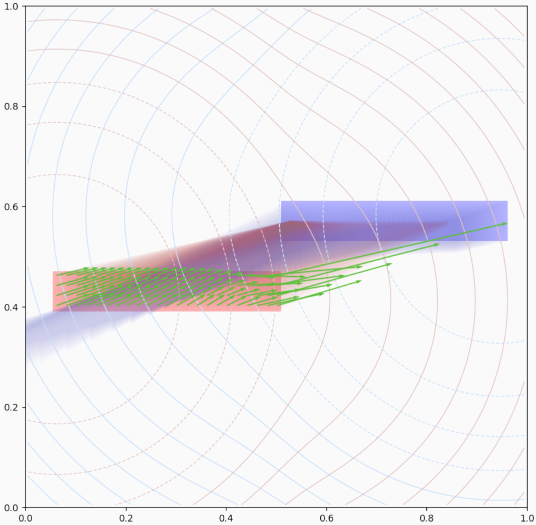
Iteration 1

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



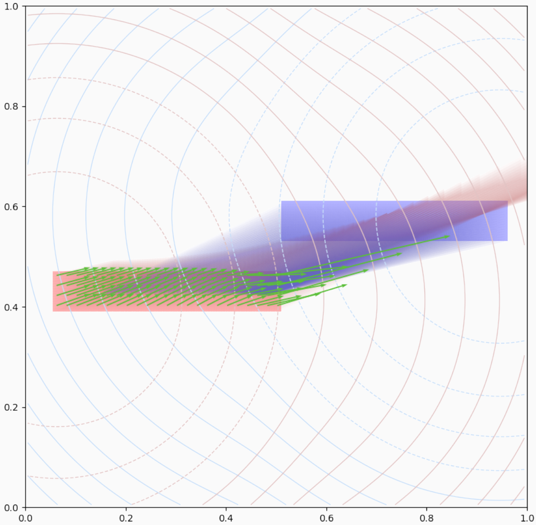
Iteration 1 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



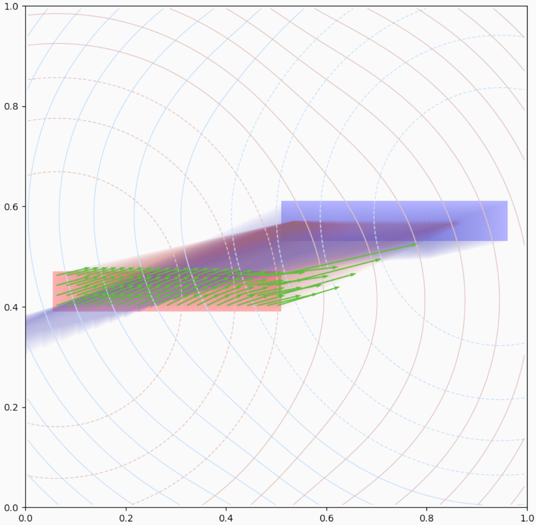
Iteration 2

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



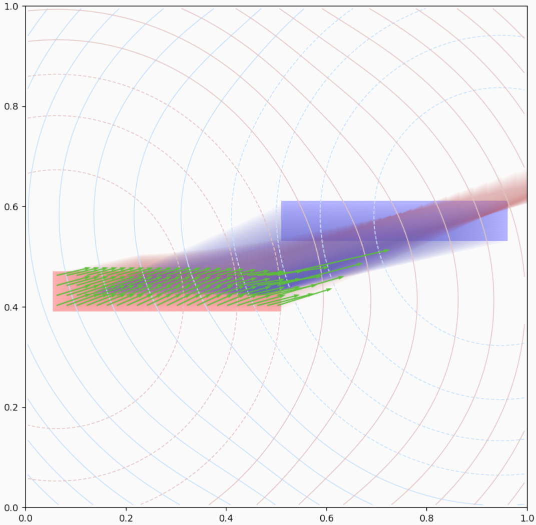
Iteration 2 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



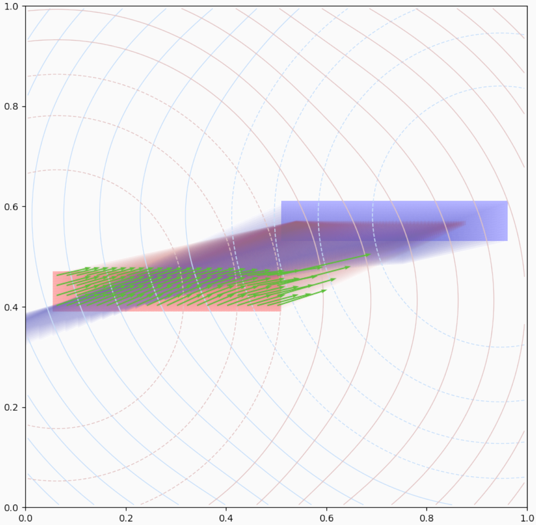
Iteration 3

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



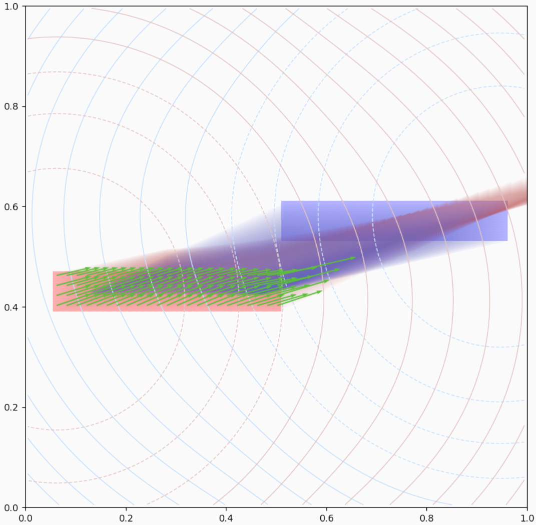
Iteration 3 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



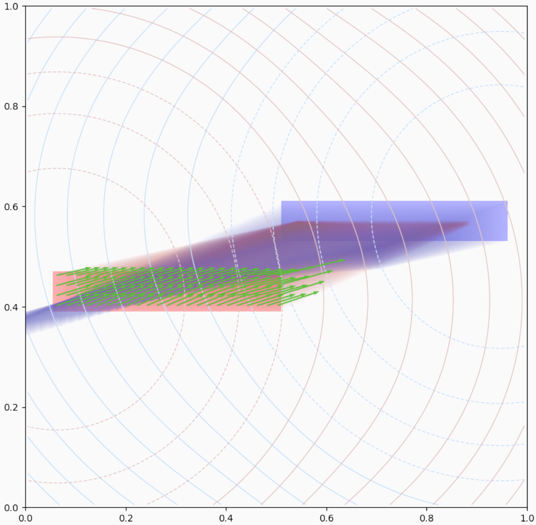
Iteration 4

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



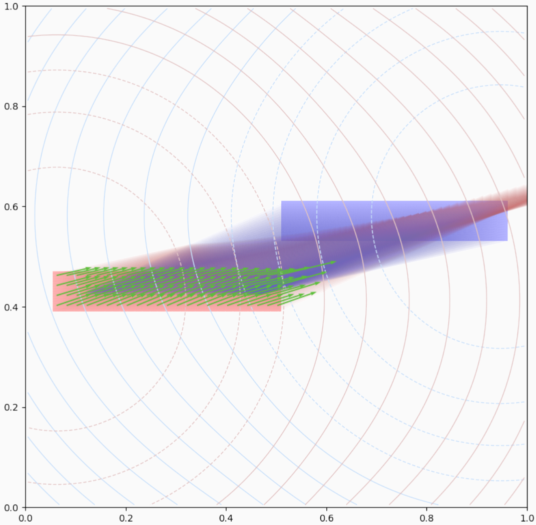
Iteration 4 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



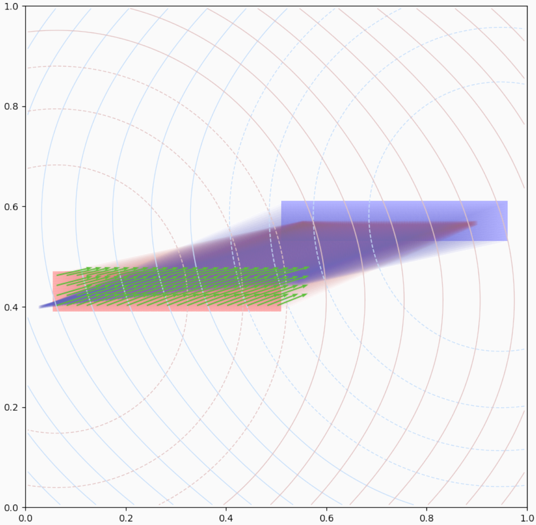
Iteration 5

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



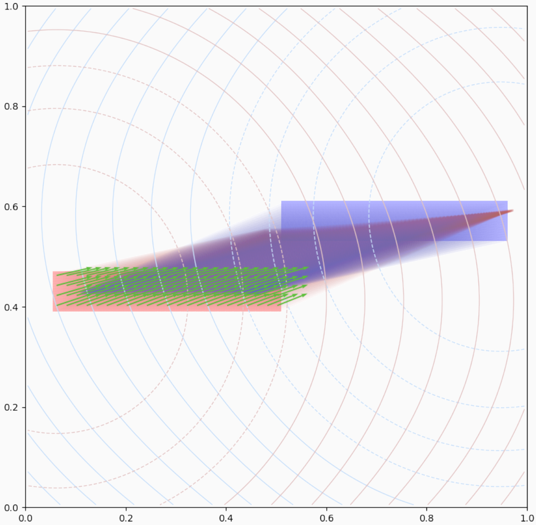
Iteration 5 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



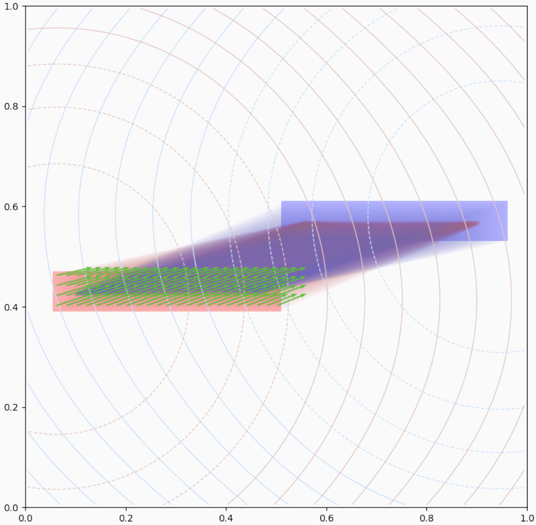
Iteration 10

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



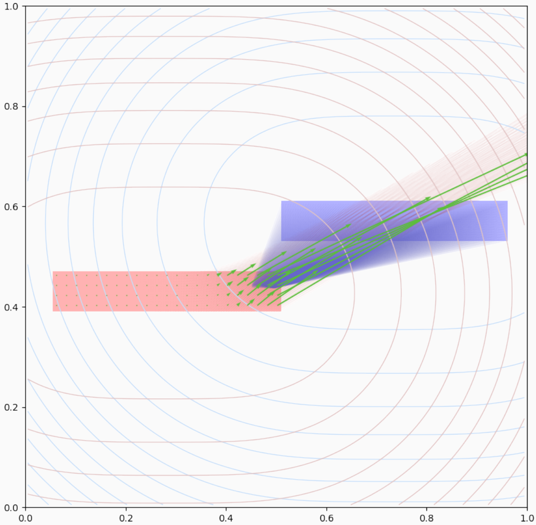
Iteration 10 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



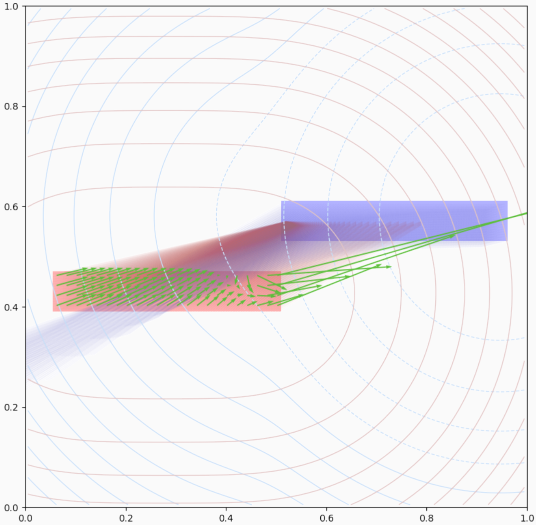
Iteration 20

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



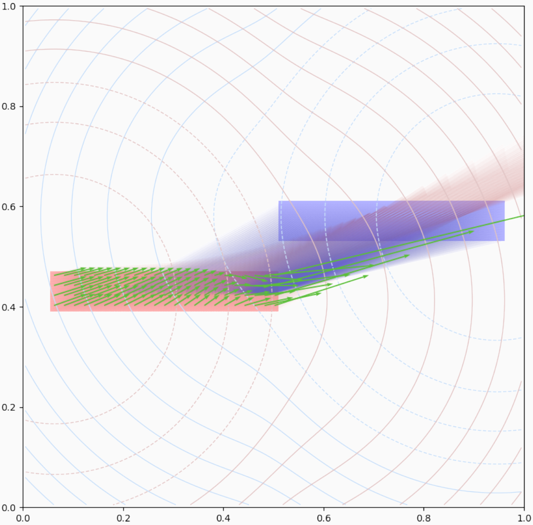
Iteration 1/2

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



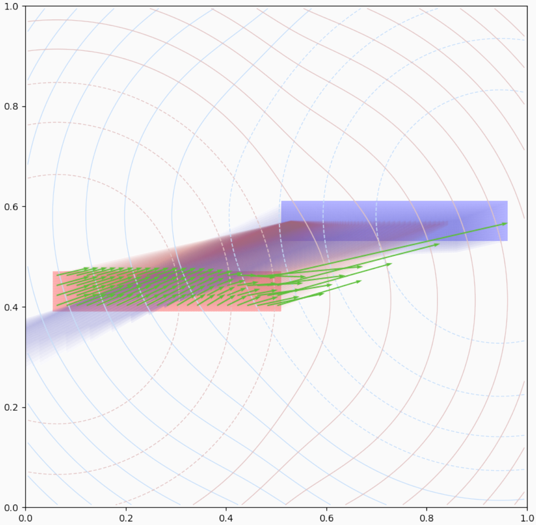
Iteration 1

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



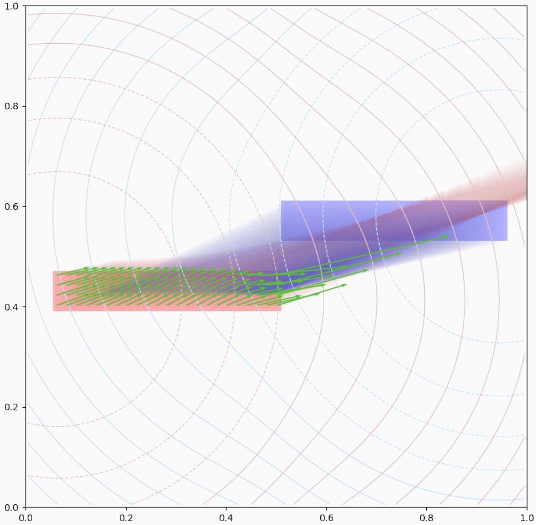
Iteration 1 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



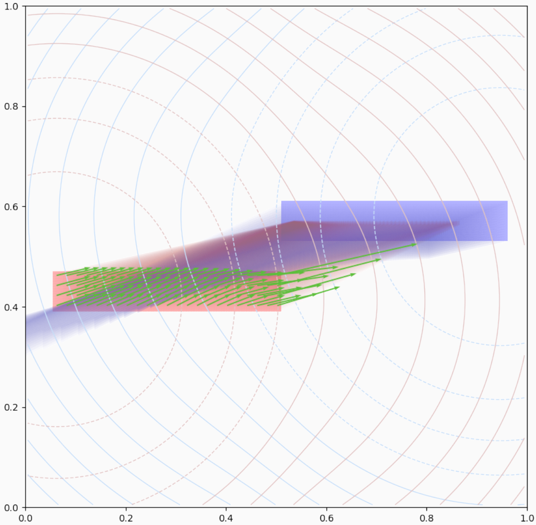
Iteration 2

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



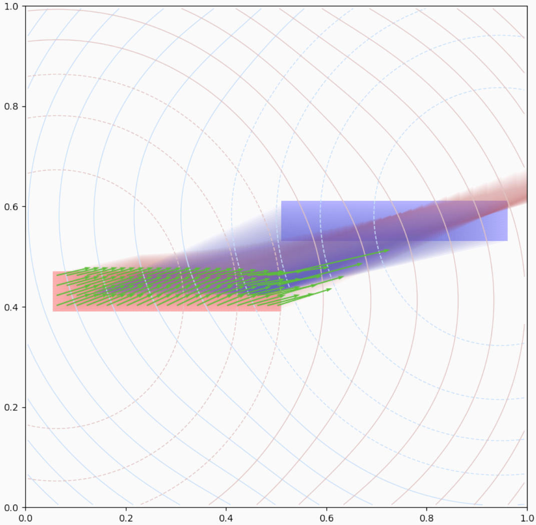
Iteration 2 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



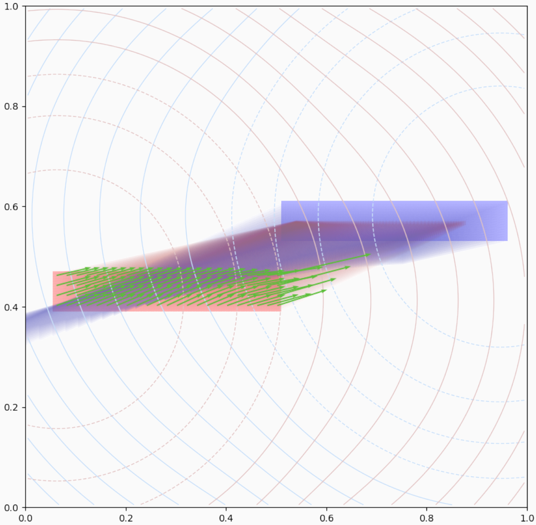
Iteration 3

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



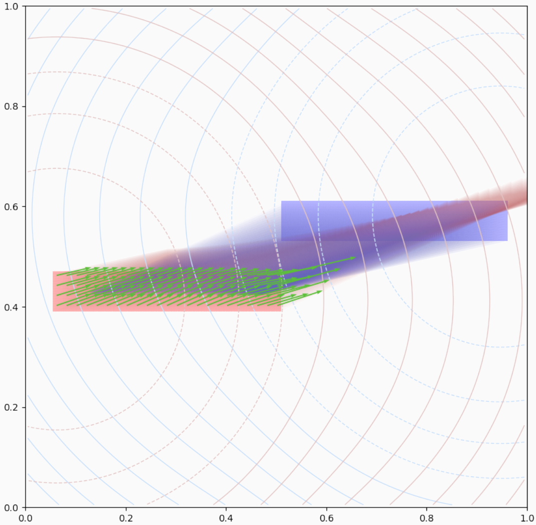
Iteration 3 ½

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



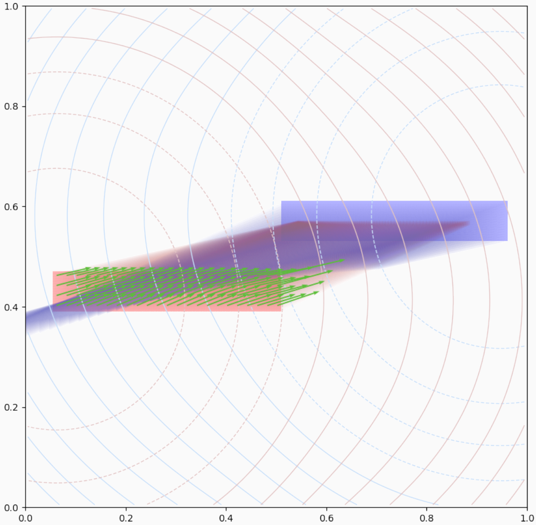
Iteration 4

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



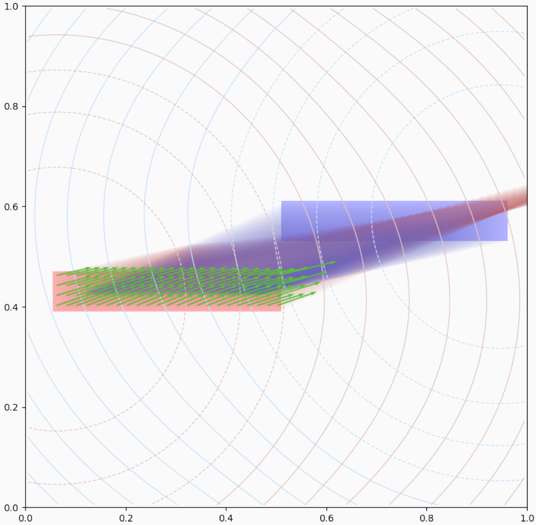
Iteration 4 ½

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



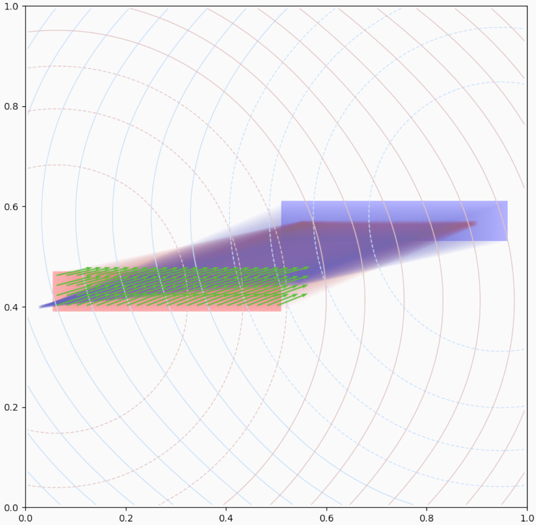
Iteration 5

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



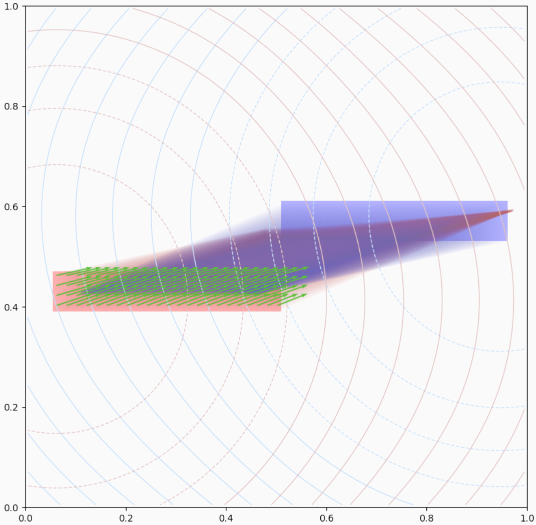
Iteration 5 ½

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



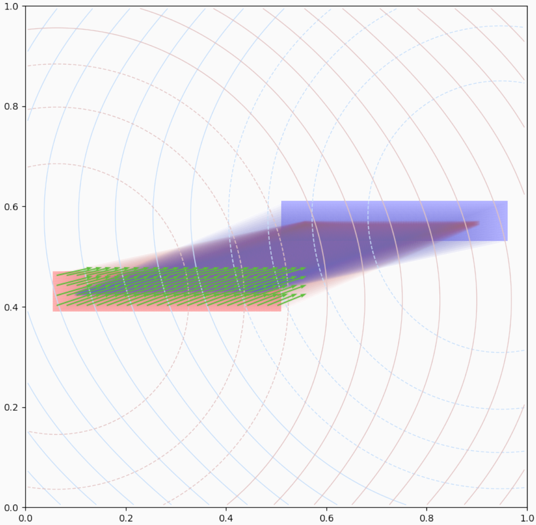
Iteration 10

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



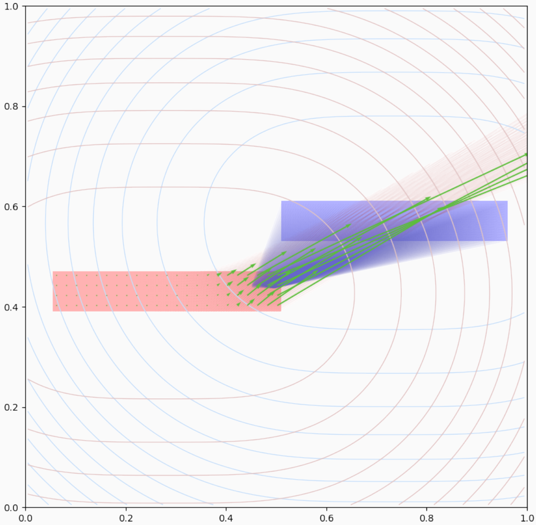
Iteration 10 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



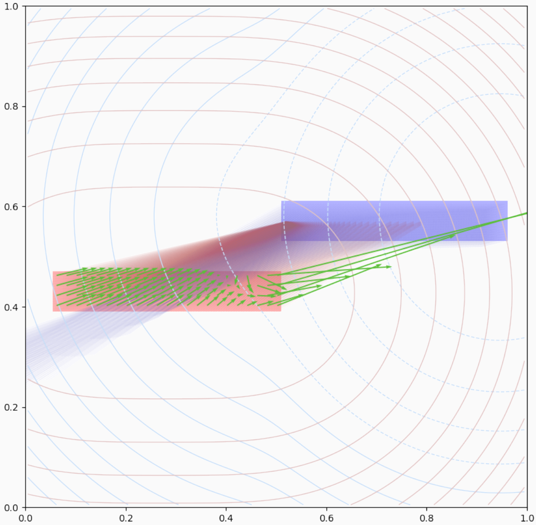
Iteration 20

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



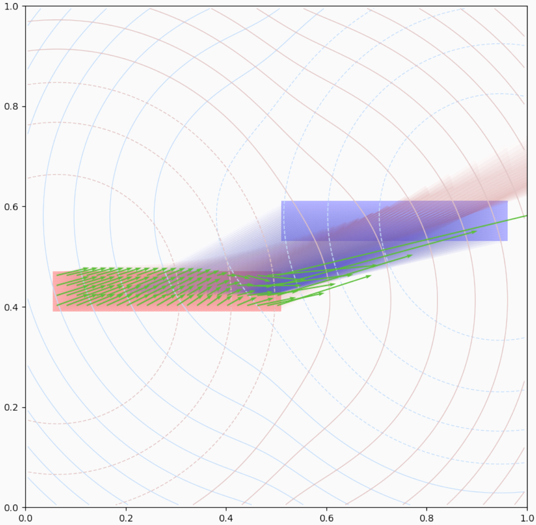
Iteration 1/2

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



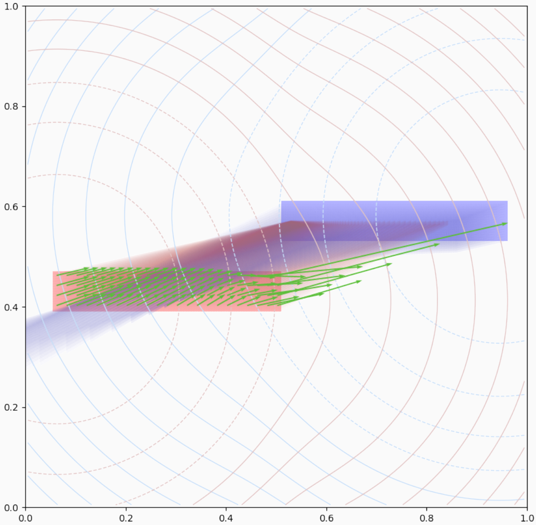
Iteration 1

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



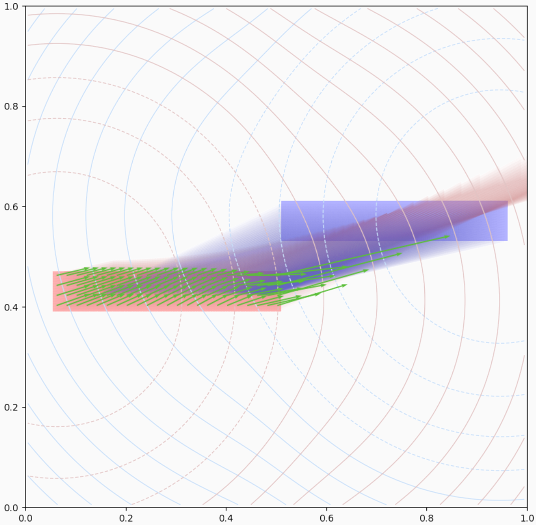
Iteration 1 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



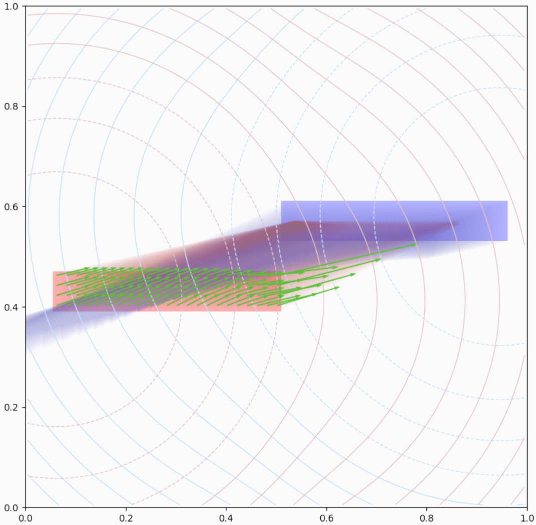
Iteration 2

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



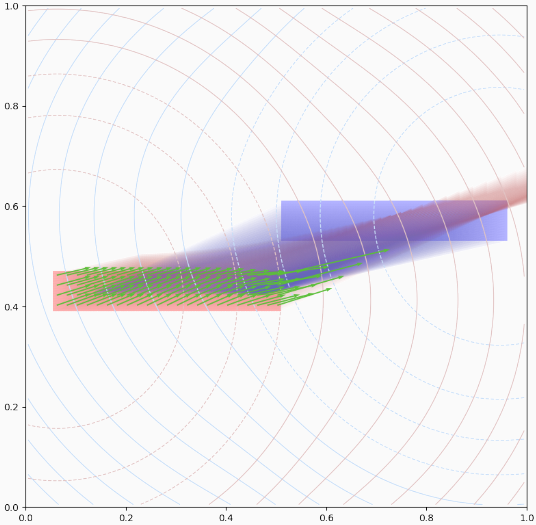
Iteration 2 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



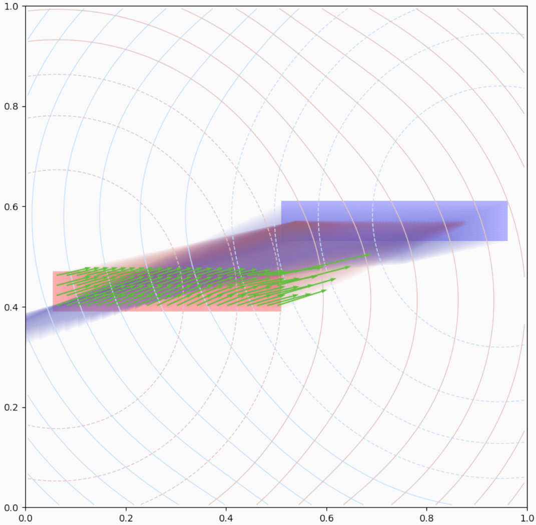
Iteration 3

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



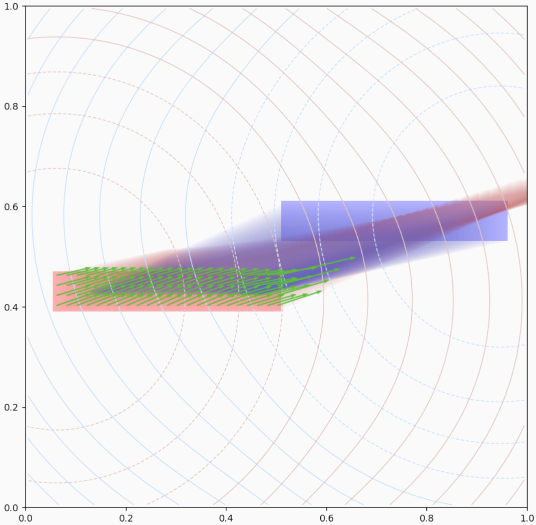
Iteration 3 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



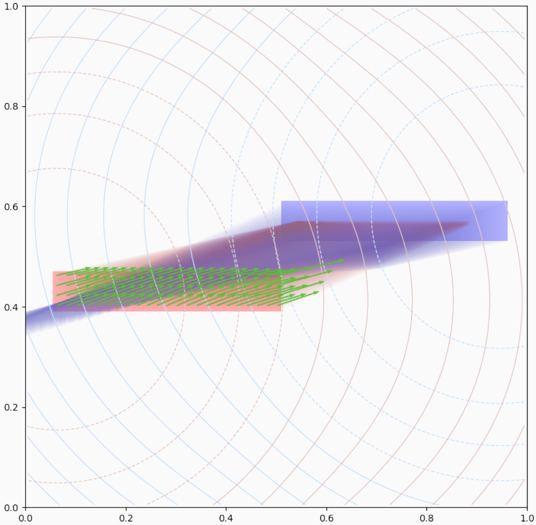
Iteration 4

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



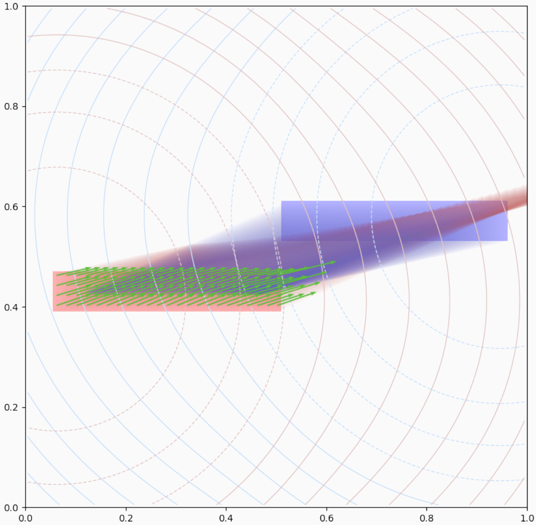
Iteration 4 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



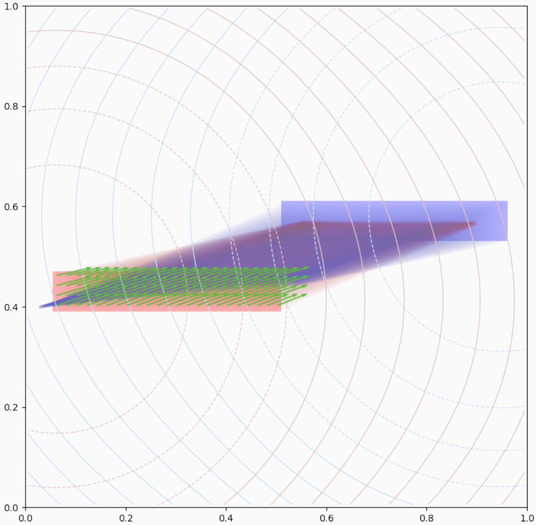
Iteration 5

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



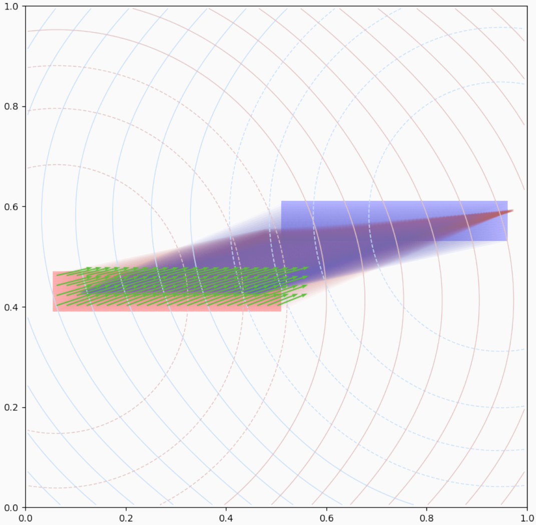
Iteration 5 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



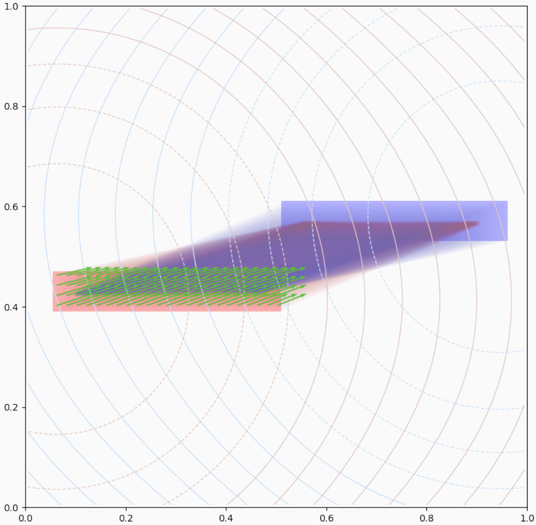
Iteration 10

The Sinkhorn algorithm, in practice; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



Iteration 10 ½

The Sinkhorn algorithm, in practice; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



Iteration 20

In our paper: theoretical guarantees

Theorem (F., Séjourné, Vialard, Trouvé, Amari, Peyré; 2018)

We define a symmetric Sinkhorn divergence:

$$d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta).$$

Then, for all probability measures α, β and regularization $\varepsilon > 0$:

In our paper: theoretical guarantees

Theorem (F., Séjourné, Vialard, Trouvé, Amari, Peyré; 2018)

We define a symmetric Sinkhorn divergence:

$$d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta).$$

Then, for all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

$$0 \leqslant d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \textit{with equality iff.} \quad \alpha = \beta$$

In our paper: theoretical guarantees

Theorem (F., Séjourné, Vialard, Trouvé, Amari, Peyré; 2018)

We define a symmetric Sinkhorn divergence:

$$d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta).$$

Then, for all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

$$0 \leqslant d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \text{is convex and differentiable}$$

In our paper: theoretical guarantees

Theorem (F., Séjourné, Vialard, Trouvé, Amari, Peyré; 2018)

We define a symmetric Sinkhorn divergence:

$$\mathrm{d}_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta).$$

Then, for all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

$$0 \leqslant \mathrm{d}_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \textit{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto \mathrm{d}_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \textit{is convex and differentiable}$$

$$\textit{Wasserstein}(\alpha, \beta) \xleftarrow{\varepsilon \to 0} \mathrm{d}_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \xrightarrow{\varepsilon \to +\infty} \textit{Energy}(\alpha, \beta)$$

In our paper: theoretical guarantees

Theorem (F., Séjourné, Vialard, Trouvé, Amari, Peyré; 2018)

We define a symmetric Sinkhorn divergence:

$$d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta).$$

Then, for all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

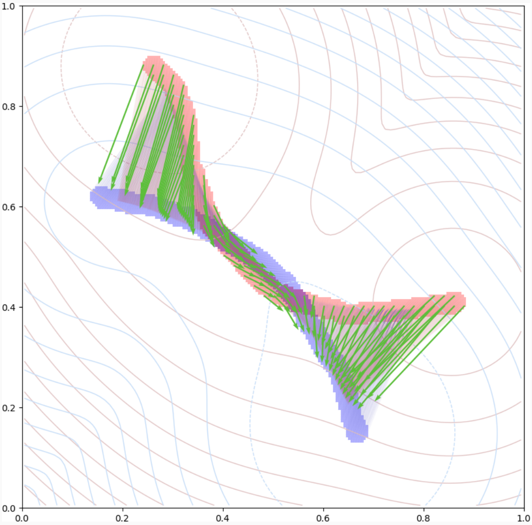
$$0 \leqslant d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \quad \text{is convex and differentiable}$$

$$\text{Wasserstein}(\alpha, \beta) \xleftarrow{\varepsilon \to 0} d_{\varepsilon\text{-Sinkhorn}}(\alpha, \beta) \xrightarrow{\varepsilon \to +\infty} \text{Energy}(\alpha, \beta)$$

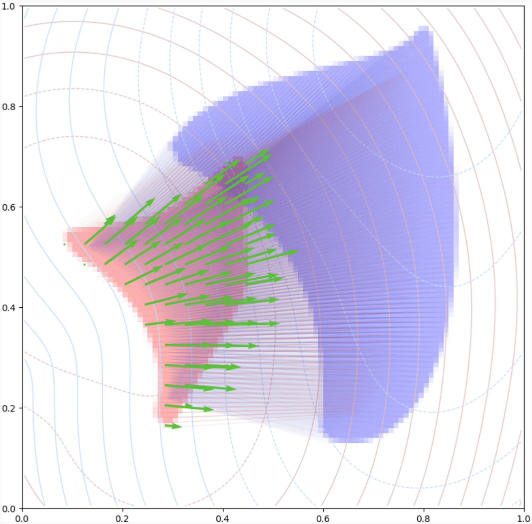
These results can be generalized to arbitrary ***feature*** *spaces – e.g. (position,orientation,curvature).*

The ε-Sinkhorn divergence; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



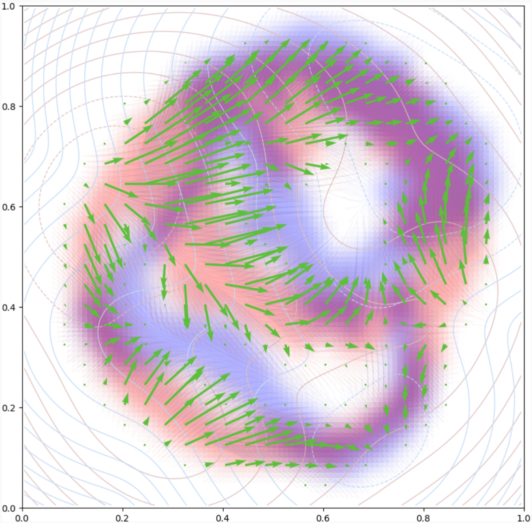
A high-quality gradient.

The ε-Sinkhorn divergence; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



A high-quality gradient.

The ε-Sinkhorn divergence; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



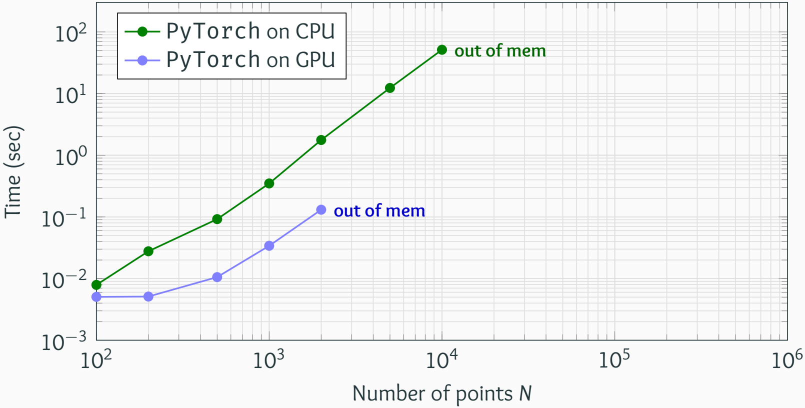
A high-quality gradient.

(Data from the Spectral Log-Demons paper.)

In practice

KErnel OPerationS, with autodiff, without memory overflows

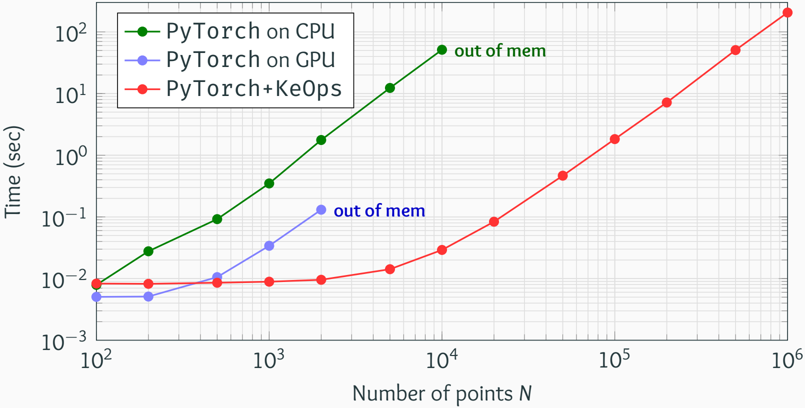
Kernel norm + gradient with *N* vertices on a cheap laptop's GPU (GTX960M)



$\Longrightarrow$ `pip install pykeops` $\Longleftarrow$

(Thanks Benjamin and Joan!)

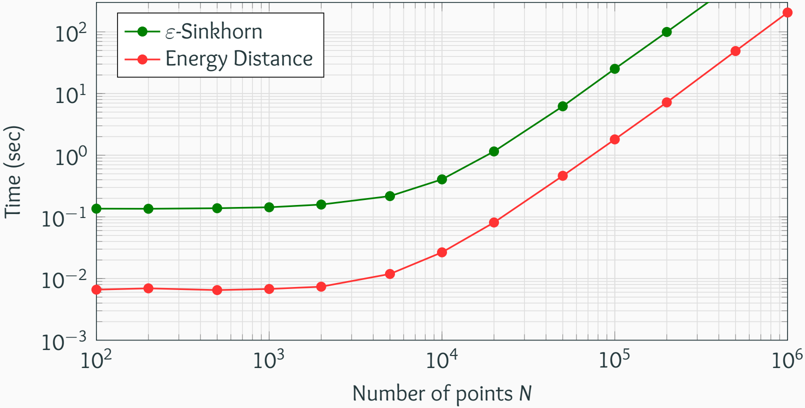
Kernel norm + gradient with N vertices on a cheap laptop's GPU (GTX960M)



$\Longrightarrow$ `pip install pykeops` $\Longleftarrow$

(Thanks Benjamin and Joan!)

Fidelity + gradient with N vertices on a **cheap laptop**'s GPU (GTX960M)

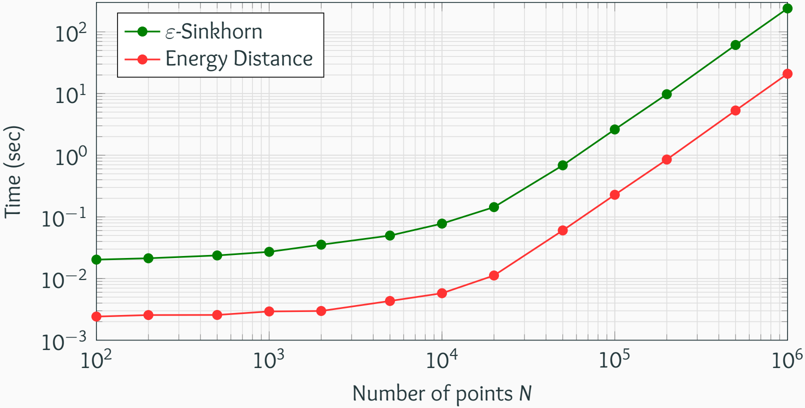


KErnel OPerationS, with autodiff, without memory overflows

$\Longrightarrow$ `pip install pykeops` $\Longleftarrow$

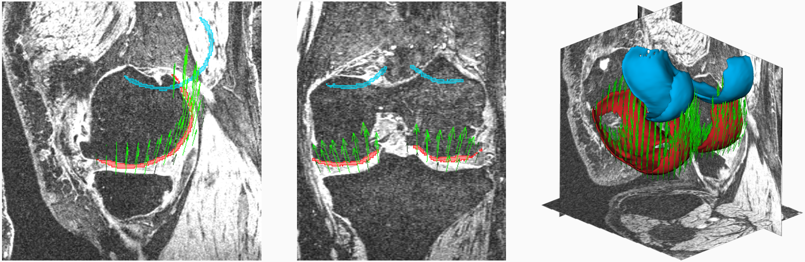
(Thanks Benjamin and Joan!)

Fidelity + gradient with N vertices on a **high-end** GPU (Tesla P100)



We provide a reference PyTorch implementation

`github.com/jeanfeydy/global-divergences`.



Gradient of the Energy Distance, computed in 0.5s on my laptop.
Data from the OsteoArthritris Initiative:
52,319 and 34,966 voxels out of a 192-192-160 volume.

Conclusion

Robust, **geometry-aware** loss functions are easy to compute.

Conclusion

Robust, **geometry-aware** loss functions are easy to compute.

- Try using $k(x, y) \; = \; -\|x - y\|$!

Conclusion

Robust, **geometry-aware** loss functions are easy to compute.

- Try using $k(x,y) \;=\; -\|x-y\|$!
- Sinkhorn $\;=\;$ Hausdorff + mass **spreading** constraint
  $\simeq$ best you can do without topology or landmarks
  $\simeq$ 20-50 convolutions through the data
  $\rightarrow$ Is it worth it?

Conclusion

Our work:

- Miccai2017: proof of concept

Conclusion

Our work:

- Miccai2017: proof of concept
- ShapeMi2018:
  - link with statistics and **computer graphics**
  - reference **implementation** on sparse data
  - theoretical **guarantees**

Conclusion

Our work:

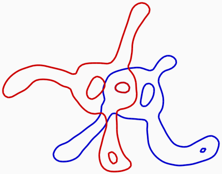
- Miccai2017: proof of concept
- ShapeMi2018:
    - link with statistics and **computer graphics**
    - reference **implementation** on sparse data
    - theoretical **guarantees**
- 2019:
    - **evaluation** in varied settings
    - separable **volumetric** implementation

Thank you for your attention.

Any questions ?

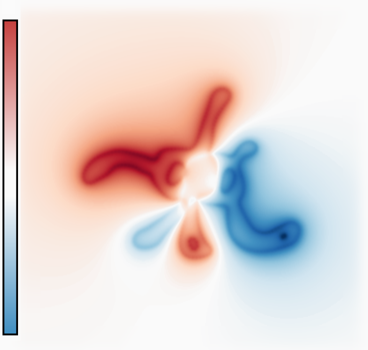
An idea from statistics: Kernel distances

Kernel fidelities: the simplest formula for $d(\alpha, \beta)$



Raw signal $(\alpha - \beta)$.

Kernel fidelities: the simplest formula for $d(\alpha, \beta)$

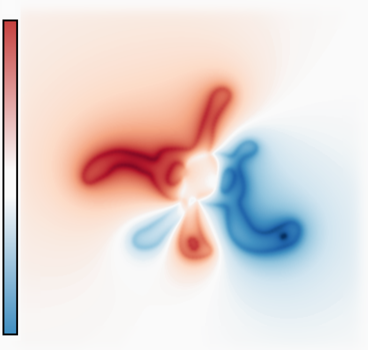


Blurred signal $g \star (\alpha - \beta)$.

Choose a symmetric blurring function g, a **kernel** $k = g \star g$:

$$\mathrm{d}_k(\alpha, \beta) \;=\; \tfrac{1}{2} \| \, g \star \alpha - g \star \beta \, \|_{L^2}^2$$

Kernel fidelities: the simplest formula for $d(\alpha, \beta)$

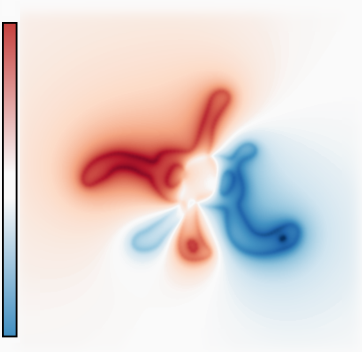


Blurred signal $g \star (\alpha - \beta)$.

Choose a symmetric blurring function g, a **kernel** $k = g \star g$:

$$\begin{aligned} \mathrm{d}_k(\alpha, \beta) &= \tfrac{1}{2} \| \, g \star \alpha - g \star \beta \, \|_{L^2}^2 \\ &= \tfrac{1}{2} \langle \, \alpha - \beta \, | \, k \star (\alpha - \beta) \, \rangle \end{aligned}$$

Kernel fidelities: the simplest formula for $d(\alpha, \beta)$

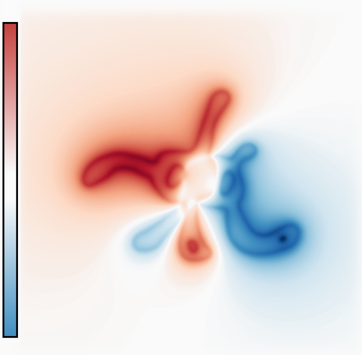


Blurred signal $g \star (\alpha - \beta)$.

Choose a symmetric blurring function g, a **kernel** $k = g \star g$:

$$
\begin{aligned}
\mathrm{d}_k(\alpha, \beta) &= \tfrac{1}{2} \| \, g \star \alpha - g \star \beta \, \|_{L^2}^2 \\
&= \tfrac{1}{2} \langle \, \alpha - \beta \mid k \star (\alpha - \beta) \, \rangle \\
&= - \sum_{i,j} k(x_i, y_j) \, \alpha_i \, \beta_j + \cdots
\end{aligned}
$$

Kernel fidelities: the simplest formula for $d(\alpha, \beta)$



Blurred signal $g \star (\alpha - \beta)$.

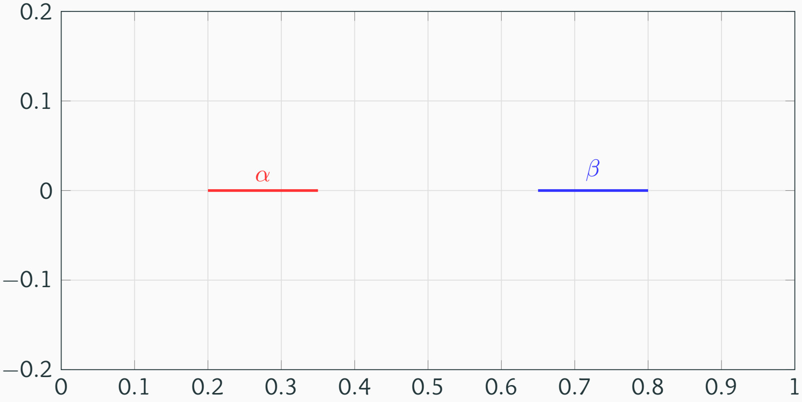
Choose a symmetric blurring function g, a **kernel** $k = g \star g$:

$$\begin{aligned} \mathrm{d}_k(\alpha, \beta) &= \tfrac{1}{2} \| \, g \star \alpha - g \star \beta \, \|_{L^2}^2 \\ &= \tfrac{1}{2} \langle \, \alpha - \beta \mid k \star (\alpha - \beta) \, \rangle \\ &= - \sum_{i,j} k(x_i, y_j) \, \alpha_i \, \beta_j + \cdots \\ &= \tfrac{1}{2} \langle \, \alpha - \beta \mid b^k - a^k \, \rangle \end{aligned}$$

with $a^k = -k \star \alpha$, $b^k = -k \star \beta$.

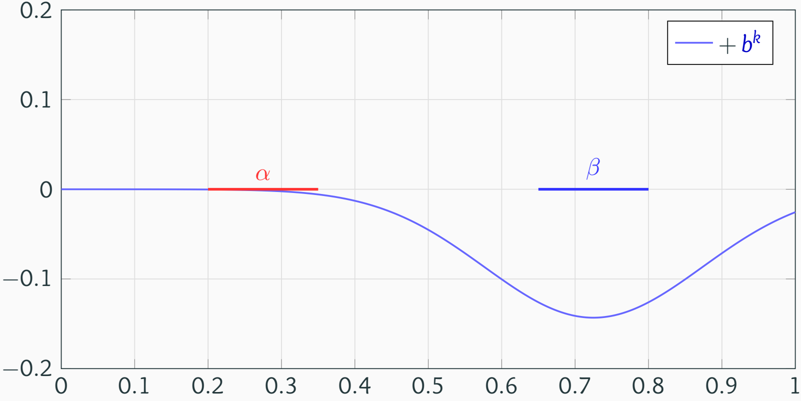
The registration flows along the gradient of $b^k - a^k$

$$k(x-y) \;=\; \exp(-\|x-y\|^2/.2^2)$$



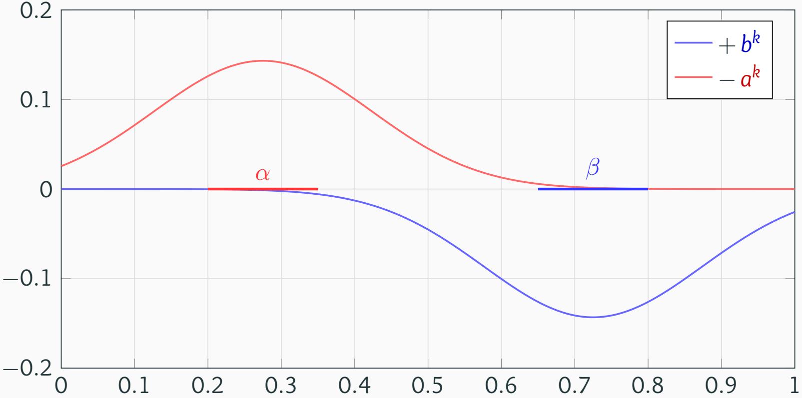
The registration flows along the gradient of $b^k - a^k$

$$k(x-y) \;=\; \exp(-\|x-y\|^2/.2^2)$$



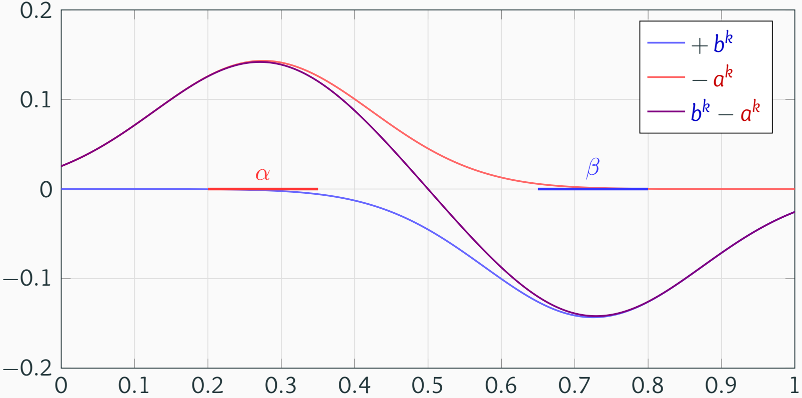
The registration flows along the gradient of $b^k - a^k$

$$k(x-y) \;=\; \exp(-\|x-y\|^2/.2^2)$$



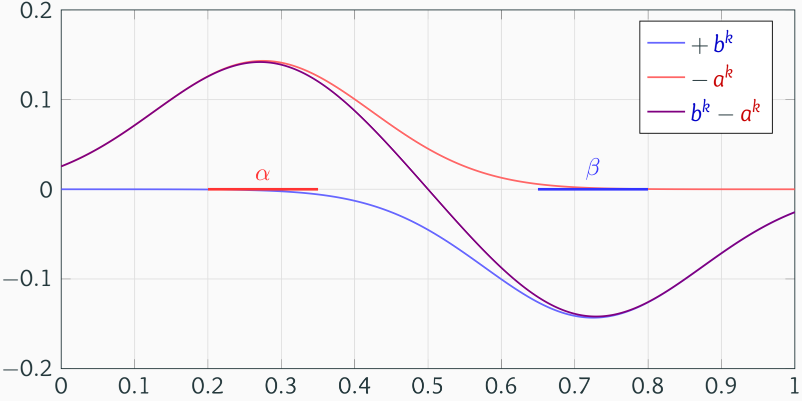
The registration flows along the gradient of $b^k - a^k$

$$k(x-y) \;=\; \exp(-\|x-y\|^2/.2^2)$$



The registration flows along the gradient of $b^k - a^k$

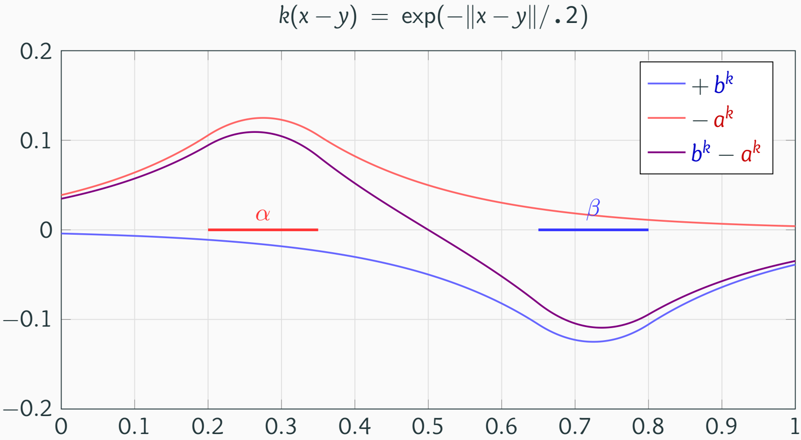
$$k(x-y) \;=\; \exp(-\|x-y\|^2/.2^2)$$



$$\mathrm{d}_k(\alpha,\beta) \;=\; \tfrac{1}{2}\langle\, \alpha-\beta \,|\, k\star(\alpha-\beta)\,\rangle$$

$$\nabla_{x_i}\mathrm{d}_k(\alpha,\beta) \;=\; \nabla\big[k\star(\alpha-\beta)\big](x_i) \;=\; \nabla b^k(x_i) - \nabla a^k(x_i)$$

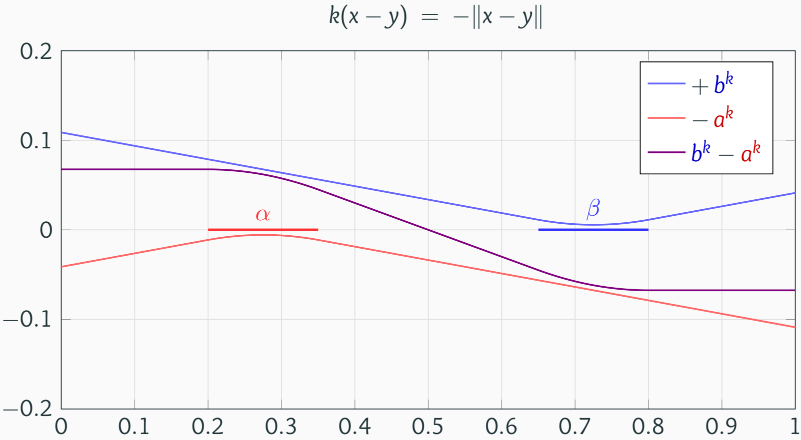
The registration flows along the gradient of $b^k - a^k$



$$\mathrm{d}_k(\alpha,\beta) \;=\; \tfrac{1}{2}\big\langle\, \alpha-\beta \,|\, k\star(\alpha-\beta)\,\big\rangle$$

$$\nabla_{x_i}\mathrm{d}_k(\alpha,\beta) \;=\; \nabla\big[k\star(\alpha-\beta)\big](x_i) \;=\; \nabla b^k(x_i) - \nabla a^k(x_i)$$

The registration flows along the gradient of $b^k - a^k$

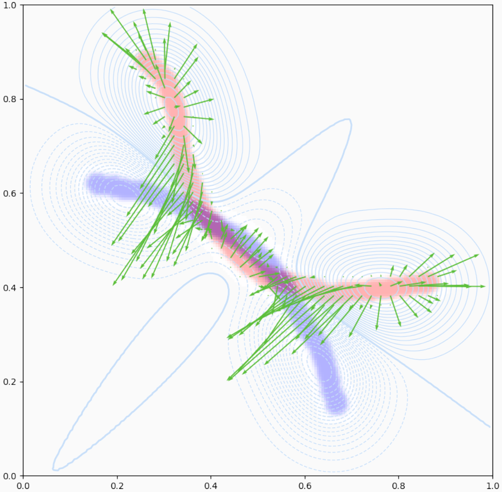


$$\mathrm{d}_k(\alpha,\beta) \;=\; \tfrac{1}{2}\langle\, \alpha-\beta \,|\, k\star(\alpha-\beta)\,\rangle$$

$$\nabla_{x_i}\mathrm{d}_k(\alpha,\beta) \;=\; \nabla\big[k\star(\alpha-\beta)\big](x_i) \;=\; \nabla b^k(x_i) - \nabla a^k(x_i)$$

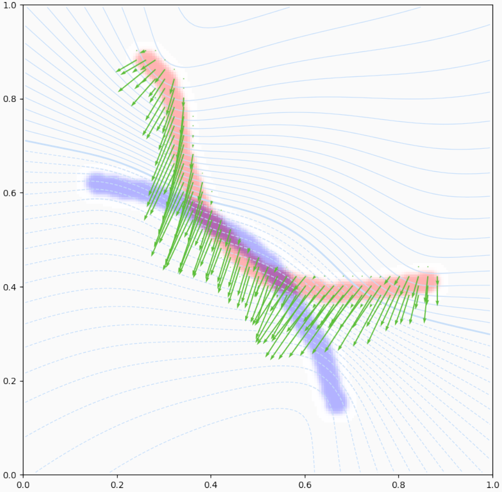
The Energy Distance is scale-invariant, robust

$$k(x-y) \;=\; \exp(-\|x-y\|^2/.1^2)$$



The Energy Distance is scale-invariant, robust

$$k(x-y) \;=\; -\|x-y\|$$



The SoftMin interpolates between a sum and a minimum

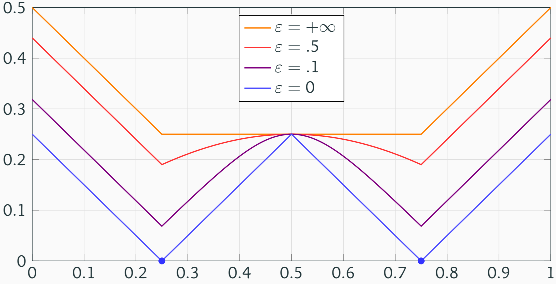
$$\log\Big(e^{c} \,+\, e^{d} \Big) \;=\; \max(c,d) \,+\, \log\big(\underbrace{e^{c-\max(c,d)} \,+\, e^{d-\max(c,d)}}_{\in\,[1,2]} \big)$$

The SoftMin interpolates between a sum and a minimum

$$\log\Big(e^c \,+\, e^d \Big) \;=\; \max(c,d) \,+\, \log\Big(\underbrace{e^{c-\max(c,d)} \,+\, e^{d-\max(c,d)}}_{\in[1,2]} \Big)$$

Building on this, for a regularization parameter $\varepsilon > 0$, we define

$$b^\varepsilon(x) \;=\; \min_{y\sim\beta}{}^{\varepsilon} \;\|x-y\| \;=\; -\varepsilon\,\log \sum_{j=1}^{M} \beta_j \,\exp\Big(-\,\tfrac{1}{\varepsilon}\|x-y_j\|\Big)$$



$b^\varepsilon(x)$, with $\beta = \frac{1}{2}\delta_{.25} + \frac{1}{2}\delta_{.75}$

An idea from computer graphics: Hausdorff distances

Can we go further?

$$
\begin{array}{cc}
 & \begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \\
\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{array} &
\begin{pmatrix}
\|x_1 - y_1\| & \|x_1 - y_2\| & \cdots & \|x_1 - y_M\| \\
\|x_2 - y_1\| & \|x_2 - y_2\| & \cdots & \|x_2 - y_M\| \\
\vdots & \vdots & \ddots & \vdots \\
\|x_N - y_1\| & \|x_N - y_2\| & \cdots & \|x_N - y_M\|
\end{pmatrix}
\end{array}
$$

Can we go further?

$$
\begin{array}{cc}
 & \begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \\
\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{array} &
\begin{pmatrix}
\|x_1 - y_1\| & \|x_1 - y_2\| & \cdots & \|x_1 - y_M\| \\
\|x_2 - y_1\| & \|x_2 - y_2\| & \cdots & \|x_2 - y_M\| \\
\vdots & \vdots & \ddots & \vdots \\
\|x_N - y_1\| & \|x_N - y_2\| & \cdots & \|x_N - y_M\|
\end{pmatrix}
\end{array}
$$

Can we go further?

$$
\begin{array}{cc}
 & \begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \\
\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{array} &
\begin{pmatrix}
\|x_1 - y_1\| & \|x_1 - y_2\| & \cdots & \|x_1 - y_M\| \\
\|x_2 - y_1\| & \|x_2 - y_2\| & \cdots & \|x_2 - y_M\| \\
\vdots & \vdots & \ddots & \vdots \\
\|x_N - y_1\| & \|x_N - y_2\| & \cdots & \|x_N - y_M\|
\end{pmatrix}
\begin{array}{l}
\to \quad \sum_j \beta_j \|x_1 - y_j\| \\
\to \quad \sum_j \beta_j \|x_2 - y_j\| \\
\vdots \\
\to \quad \sum_j \beta_j \|x_N - y_j\|
\end{array}
\end{array}
$$

Energy Distance : $\sum_j \beta_j \, \|x_i - y_j\| \quad = \quad b^k(x_i)$

Can we go further?

$$\begin{array}{cc}
 & \begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \\
\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{array} &
\begin{pmatrix}
\|x_1 - y_1\| & \|x_1 - y_2\| & \cdots & \|x_1 - y_M\| \\
\|x_2 - y_1\| & \|x_2 - y_2\| & \cdots & \|x_2 - y_M\| \\
\vdots & \vdots & \ddots & \vdots \\
\|x_N - y_1\| & \|x_N - y_2\| & \cdots & \|x_N - y_M\|
\end{pmatrix}
\begin{array}{l}
\to \quad \min_j \|x_1 - y_j\| \\
\to \quad \min_j \|x_2 - y_j\| \\
\vdots \\
\to \quad \min_j \|x_N - y_j\|
\end{array}
\end{array}$$

Energy Distance : $\sum_j \beta_j \|x_i - y_j\| \quad = \quad b^k(x_i)$

Hausdorff Distance : $\min_j \|x_i - y_j\| \quad = \quad \mathrm{d}(x_i, \mathrm{supp}(\beta))$

Can we go further?

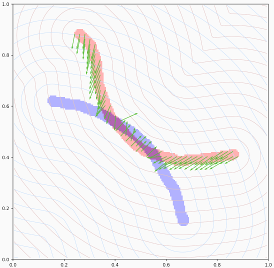
$$
\begin{array}{c|cccc|cc}
 & \beta_1 & \beta_2 & \cdots & \beta_M & & \\
\hline
\alpha_1 & \|x_1 - y_1\| & \|x_1 - y_2\| & \cdots & \|x_1 - y_M\| & \to & \min^{\varepsilon}_{y\sim\beta} \|x_1 - y\| \\
\alpha_2 & \|x_2 - y_1\| & \|x_2 - y_2\| & \cdots & \|x_2 - y_M\| & \to & \min^{\varepsilon}_{y\sim\beta} \|x_2 - y\| \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\
\alpha_N & \|x_N - y_1\| & \|x_N - y_2\| & \cdots & \|x_N - y_M\| & \to & \min^{\varepsilon}_{y\sim\beta} \|x_N - y\|
\end{array}
$$

Energy Distance : $\sum_j \beta_j \|x_i - y_j\| = b^k(x_i)$

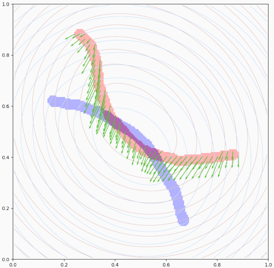
ε-SoftMin : $\min^{\varepsilon}_{y\sim\beta} \|x_i - y\| = b^{\varepsilon}(x_i)$

Hausdorff Distance : $\min_j \|x_i - y_j\| = \mathrm{d}(x_i, \mathrm{supp}(\beta))$

The SoftMin fidelity interpolates between Hausdorff and ED

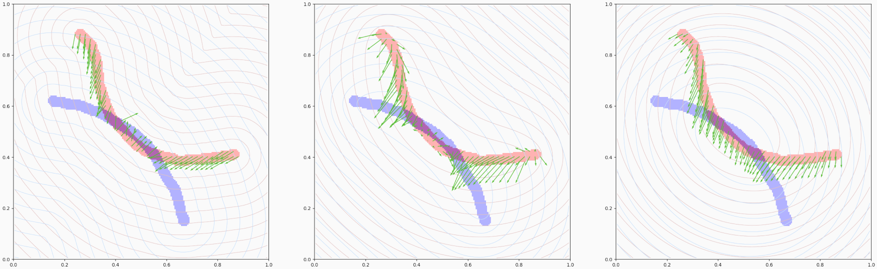


Hausdorff, min



Kernel, $\sum$

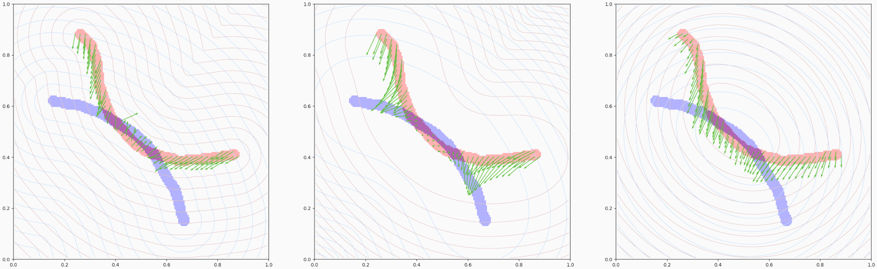
The SoftMin fidelity interpolates between Hausdorff and ED



Hausdorff, min	SoftMin, $\min^{\varepsilon}$	Kernel, $\sum$
$\varepsilon = 0$	$\longleftrightarrow$	$\varepsilon = +\infty$

$$\max^{\varepsilon}(c, d) \;=\; \varepsilon \, \log\big(\, \exp(\tfrac{c}{\varepsilon}) + \exp(\tfrac{d}{\varepsilon}) \,\big)$$

The SoftMin fidelity interpolates between Hausdorff and ED



Hausdorff, min	SoftMin, $\min^{\varepsilon}$	Kernel, $\sum$
$\varepsilon = 0$	$\longleftrightarrow$	$\varepsilon = +\infty$

$$\max^{\varepsilon}(c, d) \;=\; \varepsilon \log\Big(\exp\big(\tfrac{c}{\varepsilon}\big) + \exp\big(\tfrac{d}{\varepsilon}\big)\Big)$$

You can also use it with e.g. $\|x - y\|^2$ instead of $\|x - y\|$.

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018
- *Sinkhorn entropies and divergences*,
  F., Séjourné, Vialard, Amari, Trouvé, Peyré, 2018

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018
- *Sinkhorn entropies and divergences*,
  F., Séjourné, Vialard, Amari, Trouvé, Peyré, 2018
- *Optimal Transport for diffeomorphic registration*,
  F., Charlier, Vialard, Peyré, 2017

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. (2018).
Unbalanced optimal transport: Dynamic and kantorovich formulations.
Journal of Functional Analysis, 274(11):3090–3123.

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transport.
In *Advances in neural information processing systems*, pages 2292–2300.

Kaltenmark, I., Charlier, B., and Charon, N. (2017).
A general framework for curve and surface comparison and registration with oriented varifolds.
In *Computer Vision and Pattern Recognition (CVPR)*.

Peyré, G. and Cuturi, M. (2018).
Computational optimal transport.
arXiv preprint arXiv:1803.00567.