# Boundary estimation and Hamilton-Jacobi equations on point clouds

Jeff Calder

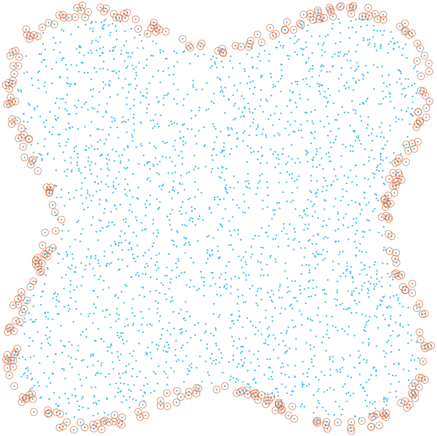
School of Mathematics
University of Minnesota

Mathematics Colloquium, University of Utah
April 14, 2022

Joint work with:
Mahmood Ettehad (IMA), Sangmin Park (CMU), Dejan Slepčev (CMU)

Research supported by NSF-DMS 1713691, 1944925, the Alfred P. Sloan Foundation, and the McKnight Foundation

# Boundary of a point cloud



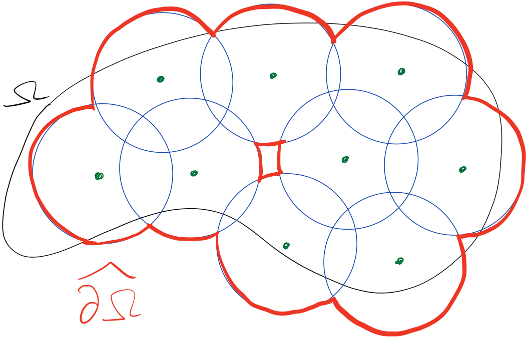
**Goal:** Identify "boundary points" of a point cloud, in a way that allows setting boundary conditions for solving PDEs.

# Previous work

To fix some notation, $\mathcal{X} = \{x^1, \ldots, x^n\}$ is an i.i.d. sample from $\Omega \subset \mathbb{R}^d$ with density $\rho$.

1 [Devroye & Wise, 1980] set

$$\Omega_n = \bigcup_{i=1}^{n} B(x^i, r) \text{ and } \widehat{\partial\Omega}_n = \partial\Omega_n.$$



# Previous work

To fix some notation, $\mathcal{X} = \{x^1, \ldots, x^n\}$ is an i.i.d. sample from $\Omega \subset \mathbb{R}^d$ with density $\rho$.

1 [Devroye & Wise, 1980] set

$$\Omega_n = \bigcup_{i=1}^{n} B(x^i, r_n) \quad \text{and} \quad \widehat{\partial\Omega}_n = \partial\Omega_n.$$

* [Cuevas and Rodriguez-Casal, 2004] showed that

$$d_H(\partial\Omega_n, \partial\Omega) \sim (n^{-1}\log(n))^{\frac{1}{d}}$$

provided $r_n \sim (n^{-1}\log(n))^{\frac{1}{d}}$.

* Computation of $\Omega_n$ is via alpha-shapes, which are only computationally feasible in $d = 2, 3$ dimensions.
* [Casal 2007] and [Aamari, Aaron, & Levrard, 2021] improve the rate by interpolating the boundary points better.

2 [Cuevas and Fraiman, 1997] use kernel density estimators to detect the boundary as a level set of $\hat{\rho}$.

3 [Lachiéze-Rey & Vega, 2017] Voronoi-cell based boundary estimator (similar to alpha-shapes for complexity).

# Previous work

4 [Wu & Wu, 2019] and [Aaron & Cholaquidis, 2020] use the size of the vector

$$\sum_{j\,:\,|x^i - x^j| \le r} (x^i - x^j).$$

- There are many other works that use similar ideas, but without theoretical guarantees
  - BORDER [Xia et al., 2006] and BRIM [Qiu et al., 2007].

# Posing the problem

There are 2 different ways to pose the problem:

1. Estimate $\partial\Omega$ from the i.i.d. sample $\mathcal{X}$.
   - Computationally very hard in high dimensions.
2. Estimate the points in the sample $\mathcal{X}$ that are close (within $\varepsilon$) of the boundary.
   - As we will show, this is tractable in high dimensions.
   - This is all we need to set boundary conditions for solving PDEs on $\mathcal{X}$.

# Distance to the boundary

We first change gears and look at estimating the distance to the boundary

$$d_\Omega(x) = \mathrm{dist}(x, \partial\Omega).$$

Provided $B(x,r) \cap \partial\Omega$ is not empty

$$\begin{aligned} d_\Omega(x) &= \max_{y\in B(x,r)\cap\Omega} \{d_\Omega(x) - d_\Omega(y)\} \\ &= \max_{y\in B(x,r)\cap\Omega} \{\nabla d_\Omega(x)\cdot(x-y)\} + O(r^2) \\ &= \max_{y\in B(x,r)\cap\Omega} \{\nu(x)\cdot(x-y)\} + O(r^2), \end{aligned}$$

since $\nabla d_\Omega(x) = \nu(x)$ is the inward normal vector.

**Note:** Estimating $d_\Omega$ boils down to estimating the inward normal vector $\nu(x)$.

# First order estimator of $d_\Omega$

For each $x^0 \in \mathcal{X}$ we define the normal vector estimator

$$\hat{v}_r(x^0) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{1}_{B(x^0,r)}(x^i)}{\hat{\theta}(x^i)}(x^i - x^0), \qquad \hat{\nu}_r(x^0) = \frac{\hat{v}_r(x^0)}{|\hat{v}_r(x^0)|},$$

$$\hat{\theta}(x) = \frac{1}{\omega_d n}\left(\frac{2}{r}\right)^d \sum_{j=1}^{n} \mathbb{1}_{B(x,r/2)}(x^j).$$

We define the first order distance function estimator $\hat{d}_r^1 : \mathcal{X} \to \mathbb{R}$ by

$$\hat{d}_r^1(x^0) = \max_{x^i \in B(x^0,r)\cap\mathcal{X}} (x^0 - x^i)\cdot\hat{\nu}_r(x^0).$$

# First order error estimates

**Theorem (Calder, Park, Slepcev, 2021)**

*Let* $x^0 \in \mathcal{X}$ *with* $d_\Omega(x^0) \le cr$ *and* $\gamma > 2$. *Then for* $r \ge C_\gamma \left(\frac{\log n}{n}\right)^{\frac{1}{d+2}}$, *both of*

$$|\hat{\nu}_r(x^0) - \nu(x^0)| \le Cr,$$

*and*

$$|d_\Omega(x^0) - \hat{d}_r^1(x^0)| \le Cr^2$$

*hold with probability at least* $1 - 5dn^{-\gamma}$.

The result is first order since $d_\Omega = O(r)$ near the boundary. Taking the smallest $r$ allowed yields errors

$$|d_\Omega(x^0) - \hat{d}_r^1(x^0)| \le C\left(\frac{\log n}{n}\right)^{\frac{2}{d+1}}.$$

# First order error estimates

## Theorem (Calder, Park, Slepcev, 2021)

*Let* $x^0 \in \mathcal{X}$ *with* $d_\Omega(x^0) \leq cr$ *and* $\gamma > 2$. *Then for* $r \geq C_\gamma \left(\frac{\log n}{n}\right)^{\frac{1}{d+2}}$, *both of*

$$|\hat{\nu}_r(x^0) - \nu(x^0)| \leq C r^2,$$

*and*

$$|d_\Omega(x^0) - \hat{d}_r^1(x^0)| \leq C r^2$$

*hold with probability at least* $1 - 5dn^{-\gamma}$.

The result is first order since $d_\Omega = O(r)$ near the boundary. Taking the smallest $r$ allowed yields errors

$$|d_\Omega(x^0) - \hat{d}_r^1(x^0)| \leq C \left(\frac{\log n}{n}\right)^{\frac{2}{d+1}}.$$

# Distance to the boundary

To get a second order estimator, we go back to the formula

$$d_\Omega(x) = \max_{y \in B(x,r)\cap\Omega} \left\{ d_\Omega(x) - d_\Omega(y) \right\},$$

and use the second order Taylor expansion

$$d_\Omega(x) - d_\Omega(y) = \frac{1}{2}(\nabla d_\Omega(x) + \nabla d_\Omega(y)) \cdot (x-y) + O(r^3).$$

This yields, provided $B(x,r) \cap \partial\Omega$ is not empty

$$\begin{aligned}
d_\Omega(x) &= \max_{y \in B(x,r)\cap\Omega} \left\{ d_\Omega(x) - d_\Omega(y) \right\} \\
&= \max_{y \in B(x,r)\cap\Omega} \left\{ \frac{1}{2}(\nabla d_\Omega(x) + \nabla d_\Omega(y)) \cdot (x-y) \right\} + O(r^3) \\
&= \max_{y \in B(x,r)\cap\Omega} \left\{ \frac{1}{2}(\nu(x) + \nu(y)) \cdot (x-y) \right\} + O(r^3).
\end{aligned}$$

# Second order estimator of $d_\Omega$

Our Taylor expansion would suggest the second order estimator

$$\max_{x^i \in B(x^0,r)\cap X_n} (x^0 - x^i)\cdot \frac{1}{2}(\hat{\nu}_r(x^0) + \hat{\nu}_r(x^i))$$

This test has difficulties with false positives at interior points, where $\hat{\nu}_r(x^0)$ and $\hat{\nu}_r(x^i)$ are not reliable, and can cancel each other out.

To avoid this problem, we define the second order estimator with cutoff

$$\hat{d}_r^2(x^0) = \max_{x^i \in B(x^0,r)\cap \mathcal{X}} (x^0 - x^i)\cdot \left[\hat{\nu}_r(x^0) + \frac{\hat{\nu}_r(x^i) - \hat{\nu}_r(x^0)}{2} \mathbb{1}_{\mathbb{R}_+}(\hat{\nu}_r(x^i)\cdot \hat{\nu}_r(x^0))\right].$$

# Second order error estimates

## Theorem (Calder, Park, Slepcev, 2021)

*Let* $x^0 \in \mathcal{X}$ *with* $d_\Omega(x^0) \le cr$ *and* $\gamma > 2$. *Then for* $r \ge C_\gamma \left(\frac{\log n}{n}\right)^{\frac{1}{d+4}}$, *both of*

$$|\hat{\nu}_r(x^0) - \nu(x^0)| \le Cr^2,$$

*and*

$$|d_\Omega(x^0) - \hat{d}_r^2(x^0)| \le Cr^3$$

*hold with probability at least* $1 - 5dn^{-\gamma}$.

Taking the smallest $r$ allowed yields errors

$$|d_\Omega(x^0) - \hat{d}_r^2(x^0)| \le C\left(\frac{\log n}{n}\right)^{\frac{3}{d+4}}.$$

# Estimating the boundary for solving PDEs

For solving PDEs with Dirichlet conditions, we want an estimator of the boundary points $\widehat{\partial\Omega} \subset \mathcal{X}$ that...

- Identifies sufficiently many boundary points so that BC are attained as $n \to \infty$.
- Does not identify any interior points as boundary points.

# Estimating the boundary for solving PDEs

Defining

$$\partial_r \Omega = \{x \in \Omega \, : \, d_\Omega(x) \le r\},$$

we ask that our boundary estimator should satisfy

(1) $$\mathcal{X} \cap \partial_\varepsilon \Omega \subset \widehat{\partial\Omega} \subset \partial_{2\varepsilon}\Omega.$$

Given an empirical estimator $\hat{d}_r$ we define the test $\widehat{T}_{\varepsilon,r} : \mathcal{X} \to \{0,1\}$ by

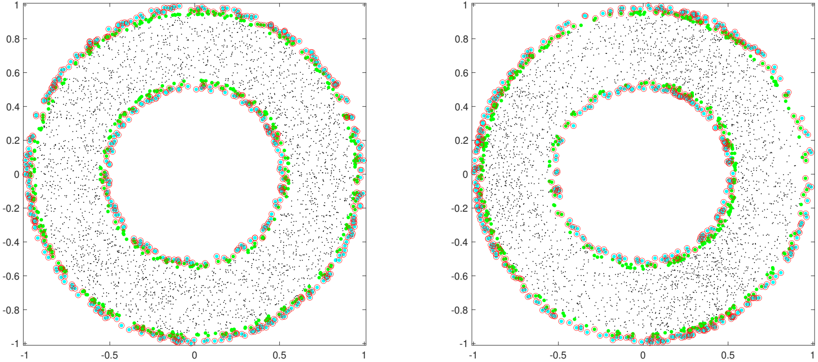
(2) $$\boxed{\widehat{T}_{\varepsilon,r}(x^0) = \begin{cases} 1 & \text{if } \hat{d}_r(x^0) < \frac{3\varepsilon}{2} \\ 0 & \text{otherwise.} \end{cases}}$$

Provided $\varepsilon \gtrsim r^3$, the second order test satisfies (1) with high probability.

Using our lower bound on $r$ from before, we can identify the boundary with resolution

$$\varepsilon \sim \left(\frac{\log n}{n}\right)^{\frac{3}{d+1}}.$$

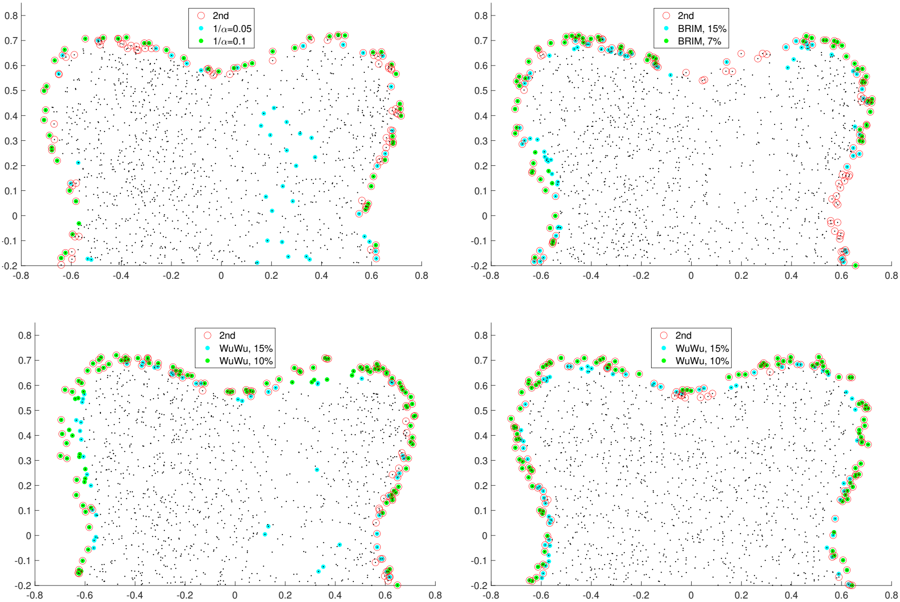
# Experiments




(a) Uniform density (b) Sinusoidal density

- Blue points satisfy $d_\Omega \leq \varepsilon$.
- Green points satisfy $\varepsilon < d_\Omega \leq 2\varepsilon$
- Red points are identified by our second order test.

# Comparison with other methods



# Extension to the manifold setting

We can extend our method to the manifold setting by:

- Estimating the tangent space with PCA.
- Projecting the normal estimation onto the estimated tangent space.

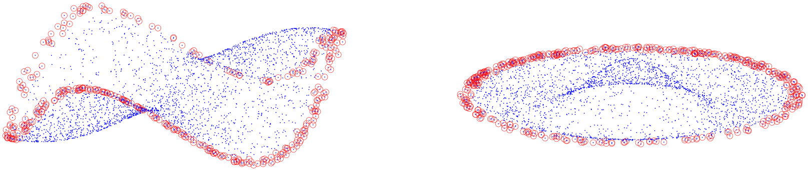


Figure: Boundary points of a point cloud on a manifold identified the second order test, $n = 3000$, $r = 0.21$, $\varepsilon = 0.05$. The point cloud is represented by blue dots, and the boundary points identified are circled in red.

# Solving PDEs on point clouds

We now turn to solving PDEs on point clouds. We assume that we have computed a boundary set $\partial_\varepsilon \mathcal{X} \subset \mathcal{X}$ that satisfies

$$\mathcal{X}_\varepsilon \subset \Omega_\varepsilon \quad \text{and} \quad \partial_\varepsilon \mathcal{X} \subset \partial_{2\varepsilon}\Omega, \tag{3}$$

where $\mathcal{X}_\varepsilon = \mathcal{X} \setminus \partial_\varepsilon \mathcal{X}$ and $\Omega_\varepsilon = \Omega \setminus \partial_\varepsilon \Omega$.

**Main Point:** We will show with a series of examples that (3) is sufficient to ensure that boundary conditions (Dirichlet/Neumann/Robin) are preserved in the limit as $n \to \infty$ and $\varepsilon \to 0$.

# Graph eikonal equation

We first consider the graph eikonal equation

$$(4) \qquad \begin{cases} \displaystyle\min_{y \in B_0(x^i,\varepsilon)\cap\mathcal{X}} \left\{u_\varepsilon(y) - u_\varepsilon(x^i) + |y - x^i|\right\} = 0, & \text{if } x^i \in \mathcal{X}_\varepsilon \\ u_\varepsilon(x^i) = 0, & \text{if } x^i \in \partial_\varepsilon \mathcal{X}, \end{cases}$$

where $u_\varepsilon : \mathcal{X} \to \mathbb{R}$ and $B_0(x,\varepsilon) := B(x,\varepsilon) \setminus \{x\}$.

## Theorem (Calder, Park, Slepcev, 2021)

*Assume* $\varepsilon \leq \frac{R}{8}$. *Let* $u_\varepsilon$ *solve* (4) *and let* $0 < t \leq \min\{\frac{1}{d}, \frac{1}{2} - \frac{4\varepsilon}{R}\}$. *Assume that*

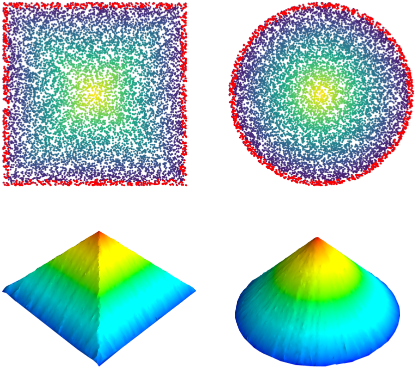
$$\mathcal{X}_\varepsilon \subset \Omega_\varepsilon \quad \textit{and} \quad \partial_\varepsilon \mathcal{X} \subset \partial_{2\varepsilon}\Omega,$$

*Then*

$$-2\varepsilon \leq u_\varepsilon - d_\Omega \leq 2d_\Omega \left(t + \frac{4\varepsilon}{R}\right) \quad \textit{on } \mathcal{X}$$

*holds with probability at least* $1 - 2n \exp\left(-\frac{\omega_{d-1}}{4(d+1)} \rho_{min} n \varepsilon^d (2t)^{\frac{d+1}{2}}\right)$.

# Numerical results



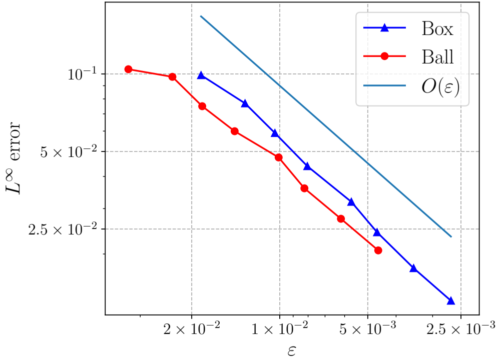



Figure: Plots of the solution to the graph eikonal equation (4) for $n = 10^4$ for both the box and ball domains, and error plots for varying $\varepsilon$ averaged over $100$ trials. The red points indicate the detected boundary points used in solving (4).

# Second order equations with Robin condition

To proceed in generality, we assume there exists $C_\nu$ such that

$$|\hat{\nu}_\varepsilon(x^i) - \nu(x^i)| \le C_\nu \varepsilon \tag{5}$$

for all $x^i \in \mathcal{X} \cap \partial_{2\varepsilon}\Omega$. The graph PDEs we solve will involve the graph Laplacian

$$\mathcal{L}_\varepsilon u(x^i) = \frac{2}{\sigma_\eta n \varepsilon^{d+2}} \sum_{j=1}^{n} \eta\left(\frac{|x^i - x^j|}{\varepsilon}\right)(u(x^j) - u(x^i)), \tag{6}$$

where $\sigma_\eta = \int_{\mathbb{R}^d} \eta(|z|) z_1^2 \, dz$, $\eta$ is compactly supported on $[0,1]$, and $\int_{\mathbb{R}^d} \eta(|z|)\, dz = 1$.

We define the normal derivative $\nabla_\nu u(x) = \nabla u(x) \cdot \nu$ and the approximate normal derivative $\widehat{\nabla}_\nu$ by

$$\widehat{\nabla}_\nu u(x^i) = \frac{u(p_n(x^i + \varepsilon \hat{\nu}_\varepsilon(x^i))) - u(x^i)}{\varepsilon}, \tag{7}$$

where $p_n : \Omega \to \mathcal{X}$ is the closest point map.

# Robin-type boundary conditions

We consider the following graph Poisson equation with Robin-type boundary conditions

(8)
$$\begin{cases} \mathcal{L}_\varepsilon u(x^i) = f(x^i), & \text{if } x^i \in \mathcal{X}_\varepsilon \\ \gamma u(x^i) - (1-\gamma)\widehat{\nabla}_\nu u(x^i) = g(x^i), & \text{if } x^i \in \partial_\varepsilon \mathcal{X}. \end{cases}$$

Here, $\gamma \in (0,1]$ and $f$ and $g$ are given smooth functions.

We will show that the solution of (8) converges as $n \to \infty$ and $\varepsilon \to 0$ to the solution of the Robin problem

(9)
$$\begin{cases} -\rho^{-1}\text{div}(\rho^2 \nabla u) = f, & \text{in } \Omega \\ \gamma u - (1-\gamma)\nabla_\nu u = g, & \text{on } \partial\Omega. \end{cases}$$

# Error estimate

## Theorem (Calder, Park, Slepcev, 2021)

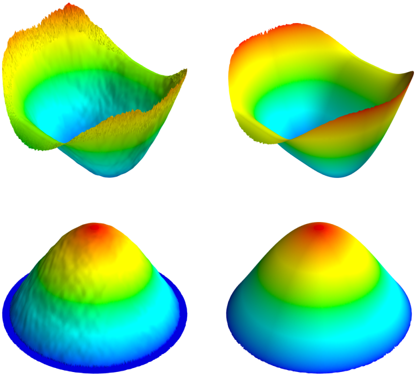
*Let* $\varepsilon > 0$ *and assume* $C_\nu \varepsilon \leq 1$. *Let* $u$ *be the solution of* (9) *with* $\gamma > 0$, *and let* $u_\varepsilon$ *satisfy* (8). *Then for any* $0 < \lambda \leq \varepsilon^{-1}$ *and* $t > 0$, *the event that*

$$|u(x^i) - u_\varepsilon(x^i)| \leq C\varepsilon$$

*holds for all* $x^i \in \mathcal{X}$ *has probability at least*

$$1 - n \exp\left(-\frac{1}{6}\omega_d \rho_{min} n \varepsilon^{2d}\right) - 2n \exp\left(-Cn\varepsilon^{d+4}\right).$$

# Numerical results



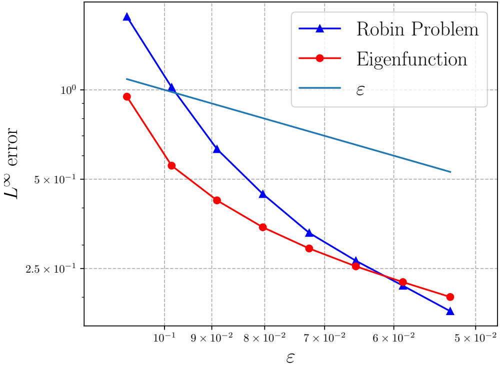



Figure: On the left, plots of the solution to the Robin problem and principal Dirichlet eigenvector for $n = 10^5$ points on the disk, compared to the exact solutions of each problem. On the right we show an error plot for varying $\varepsilon$ averaged over 100 trials.

# Dirichlet eigenfunction

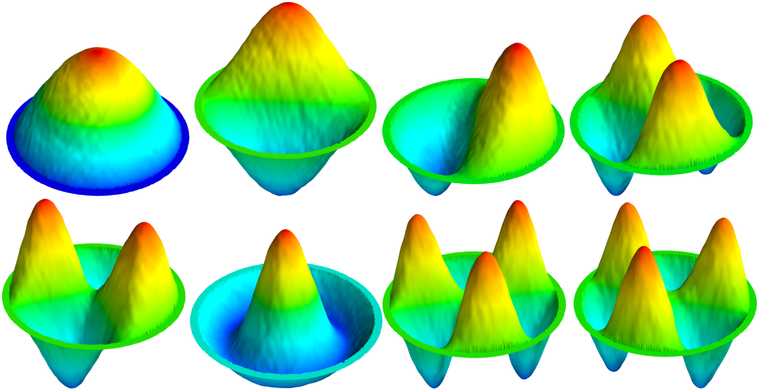
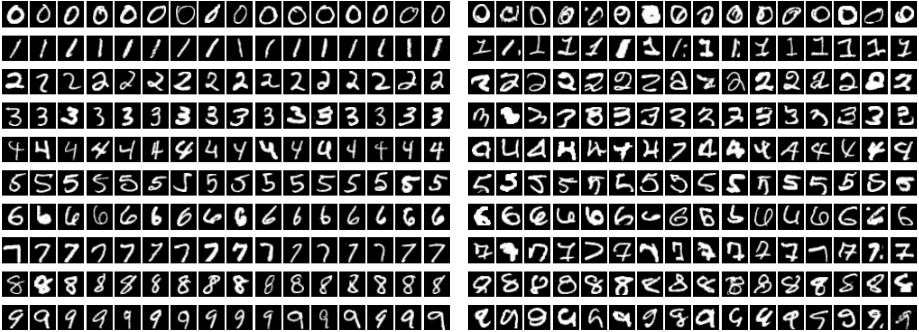


Figure: First 7 Laplacian Dirichlet eigenfunctions on the disk computed via approximation with graph Laplacian eigenvectors with $n = 10^5$ points.

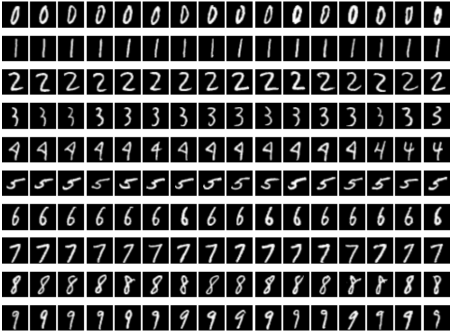
# MNIST



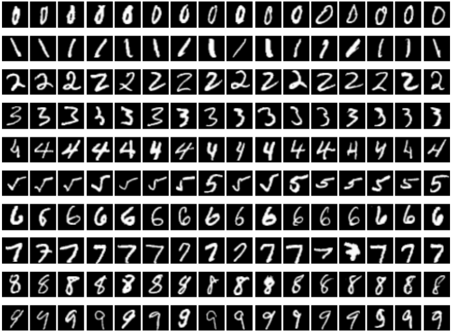
(a) Random digits (b) Boundary digits

Figure: MNIST experiments.

# MNIST



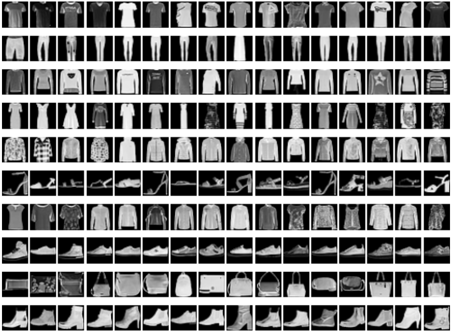
(a) Eigen Median digits



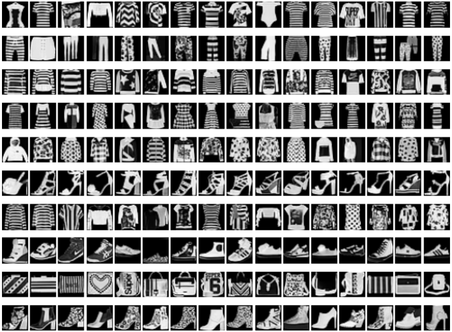
(b) Eikonal Median digits

Figure: MNIST experiments.

# FashionMNIST



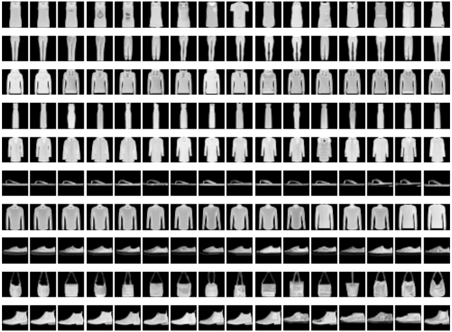
(a) Random images



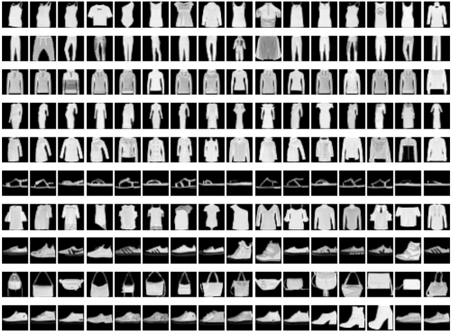
(b) Boundary images

Figure: FashionMNIST experiments.

# FashionMNIST



(a) Eigen Median images



(b) Eikonal Median images

Figure: FashionMNIST experiments.

# Paper and Code

**Paper:**

J. Calder, S. Park, and D. Slepčev (2021). **Boundary Estimation from Point Clouds: Algorithms, Guarantees and Applications**. arXiv:2111.03217.

**Code** for all experiments is on GitHub

`https://github.com/sangmin-park0/BoundaryTest`

The boundary estimation method is implemented in the **GraphLearning** python package

`https://github.com/jwcalder/GraphLearning` (pip install graphlearning)

**Python** Notebook Example:

`https://colab.research.google.com/drive/1tWOSZ9vZEAZO8T248EAiOCNtpzmpDFDT?usp=sharing`

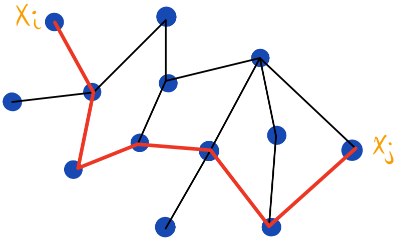
# Graph distance functions

Suppose we have a graph $G$ on $n$ vertices $\mathcal{X}$ with edge weights $w_{ij}$.

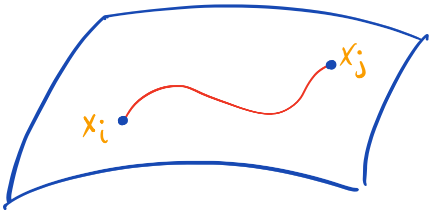
Set $I_n = \{1, \ldots, n\}$. The graph distance $d_G : \mathcal{X} \times \mathcal{X} \to R$ is defined by

$$d_G(x_i, x_j) = \min_{m \geq 1} \min_{\tau \in I_n^m} \left\{ w_{i,\tau_1}^{-1} + \sum_{i=1}^{m-1} w_{\tau_i,\tau_{i+1}}^{-1} + w_{\tau_m,j}^{-1} \right\}, \tag{10}$$

Graph Distance




Geodesic Distance

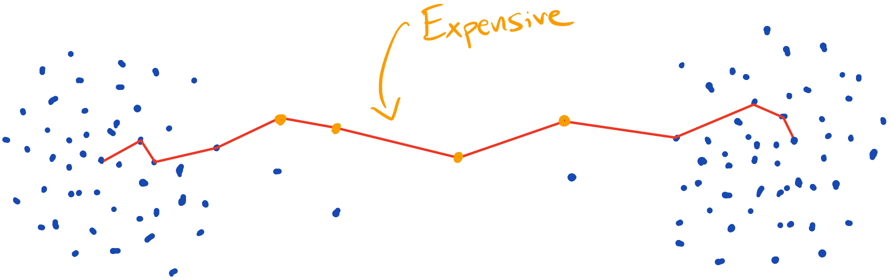


# Graph distance functions: density weighting

The weighted graph distance $d_{G,f} : \mathcal{X} \times \mathcal{X} \to R$ is defined by
(11)

$$d_{G,f}(x_i, x_j) := \min_{m \geq 1} \min_{\tau \in I_n^m} \left\{ w^{-1}_{i,\tau_1} f(x_{\tau_1}) + \sum_{i=1}^{m-1} w^{-1}_{\tau_i,\tau_{i+1}} f(x_{\tau_{i+1}}) + w^{-1}_{\tau_m,j} f(x_{\tau_j}) \right\}.$$

It is common to choose $f = \hat{\rho}^{-\alpha}$, for some density estimation $\hat{\rho}$.



# Prior work/References

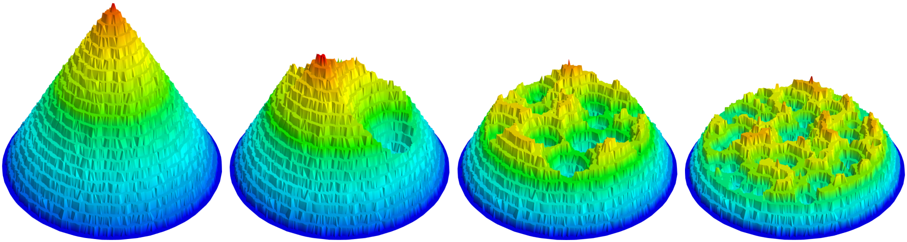
**Applications of graph distances:**

- Dimensionality reduction (e.g., ISOMAP) [Tenenbaum et al., 2000]
- Semi-supervised learning on graphs, e.g., [Bijral, et al, 2003] [Chapelle and Zien, 2005]
- Graph classification [Borgwardt and Kriegel, 2005]
- Data depth [Calder, Park and Slepcev, 2021] [Molina-Fructuoso and Murray, 2022]

**Discrete to continuum:**

- $k$-nn graphs [Alamgir and Von Luxburg, 2012]
- Geodesic manifold disatnce [Hwang, Damelin, and Hero, 2016]
- Geodesic distance on Euclidean domains [Bungert, Calder, and Roith, 2021]

# Lack of robustness to corrupted edges



(a) Graph distance function with corrupted edges

Figure: From left to right we added an increasing number of corrupted edges (0, 10, 50, and 200) with edge weight $w_{ij} = 1$.

# Graph distance functions: The eikonal equation

Let us define the graph distance to a set $\Gamma$ by

$$d_{G,f}(x,\Gamma) := \min_{x_j \in \Gamma} d_{G,f}(x_i, x_j).$$

If $G$ is connected then $u(x) = d_{G,f}(x,\Gamma)$ is the unique solution of the graph eikonal equation

$$\begin{cases} \max\limits_{x_j \in \mathcal{X}} w_{ji}(u(x_i) - u(x_j)) = f(x_i), & \text{if } x_i \in \mathcal{X} \setminus \Gamma \\ u(x_i) = 0, & \text{if } x_i \in \Gamma. \end{cases} \tag{12}$$

# The p-eikonal equation

For $p > 0$, we define the $p$-eikonal operator $\mathcal{A}_{G,p} : F(\mathcal{X}) \to F(\mathcal{X})$ by

(13) $$\mathcal{A}_{G,p}u(x_i) = \sum_{j=1}^{n} w_{ji}(u(x_i) - u(x_j))_+^p,$$

where $a_+ := \max\{a, 0\}$ is the positive part.

For $\Gamma \subset \mathcal{X}$ and $f \in F(\mathcal{X})$, we consider the $p$-eikonal equation

(14) $$\begin{cases} \mathcal{A}_{G,p}u = f, & \text{in } \mathcal{X} \setminus \Gamma \\ u = 0, & \text{on } \Gamma. \end{cases}$$

**References:**

- The $p$-eikonal equation originally appeared in [Desquenes, Elmoataz and Lezoray, 2013] with applications to image processing ($p = 1, 2, \infty$).

# Well-posedness

Let $K$ be the maximum unweighted degree of the graph, and $G^\alpha$ be the graph with weights $w_{ij}^\alpha$.

## Theorem (Calder, Ettehad, 2022)

*Let $p > 0$ and $f > 0$. If $G$ is connected, then the $p$-eikonal equation*

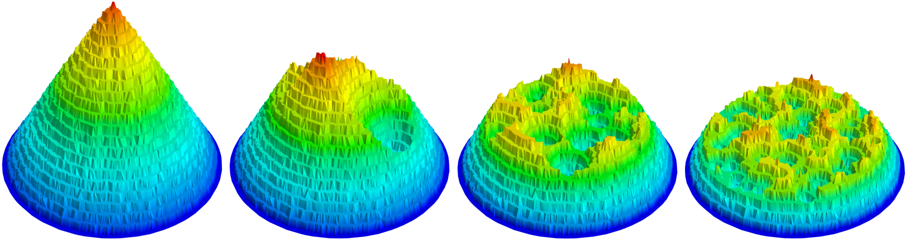
$$\begin{cases} \mathcal{A}_{G,p} u = f, & \text{in } \mathcal{X} \setminus \Gamma \\ u = 0, & \text{on } \Gamma. \end{cases} \tag{15}$$

*has a unique solution $u \in F(\mathcal{X})$, and*

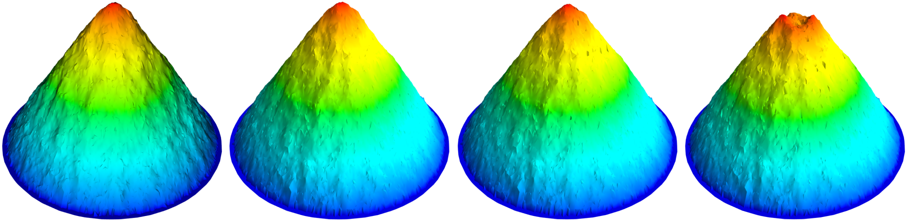
$$K^{-\frac{1}{p}} \left( \min_{\mathcal{X}} f^{\frac{1}{p}} \right) d_{G^{\frac{1}{p}}}(x_i, \Gamma) \le u(x_i) \le \left( \max_{\mathcal{X}} f^{\frac{1}{p}} \right) d_{G^{\frac{1}{p}}}(x_i, \Gamma). \tag{16}$$

- Note that the estimates above imply that $u$ recovers the graph distance as $p \to \infty$.

# Robustness



(a) Graph distance function with corrupted edges



(b) $p$-eikonal equation with $p = 1$ with corrupted edges

# Robustness

## Theorem (Calder, Ettehad, 2022)

*Let $\delta W$ have nonnegative entries, and set $\tilde{G} = (\mathcal{X}, W + \delta W)$ and $\delta G = (\mathcal{X}, \delta W)$. Let $u, \tilde{u} \in F(\mathcal{X})$ satisfy*

$$\begin{cases} \mathcal{A}_{\tilde{G},p}\tilde{u}(x_i) = \mathcal{A}_{G,p}u(x_i) = f(x_i), & \textit{if } x_i \in \mathcal{X} \setminus \Gamma \\ \tilde{u}(x_i) = u(x_i) = 0, & \textit{if } x_i \in \Gamma. \end{cases} \tag{17}$$

*Then for all $x_i \in \mathcal{X}$ we have*

$$0 \le \frac{u(x_i) - \tilde{u}(x_i)}{u(x_i)} \le \left( \max_{\mathcal{X}\setminus\Gamma} \frac{\mathcal{A}_{\delta G,p}u}{f} \right)^{\frac{1}{p}}. \tag{18}$$

- The theorem can be simplified to give the weaker bound

$$0 \le \frac{u(x_i) - \tilde{u}(x_i)}{u(x_i)} \le C \left( \frac{f_{max}}{f_{min}} \right)^{\frac{1}{p}} \|\delta W\|_{u,1}^{\frac{1}{p}},$$

$$\|A\|_{u,1} = \max_{1\le j\le n} \sum_{i=1}^{n} |A_{ij}| 1_{u(x_j)>u(x_i)}.$$

# Discrete to continuum

- Let $x_1, x_2, \ldots, x_n$ be a sequence of $i.i.d$ random variables on $\Omega \subset \mathbb{R}^d$ with Lipschitz and positive density $\rho$ and set

(19) $$\mathcal{X} := \{x_1, x_2, \ldots, x_n\}.$$

- Assume that $\Omega \subset \mathbb{R}^d$ is open, bounded and connected with a $C^{1,1}$ boundary $\partial\Omega$.
- We define the $p$-eikonal operator on a random geometric graph as

$$\mathcal{A}_{n,\varepsilon} u(x) := \frac{1}{n\sigma_p \varepsilon^p} \sum_{y \in \mathcal{X}} \eta_\varepsilon\big(|x-y|\big)\big(u(x)-u(y)\big)_+^p,$$

where $\eta_\varepsilon(t) := \frac{1}{\varepsilon^d}\eta(\frac{t}{\varepsilon})$ and set $\sigma_p := \int_{\mathbb{R}^d} \eta_\varepsilon(|z|)|z_1|^p dz$

- Let $\Gamma \subset \mathcal{X}$ such that

$$\operatorname{dist}(\Gamma, \partial\Omega) \geq R,$$

where $R$ is the reach of $\partial\Omega$.

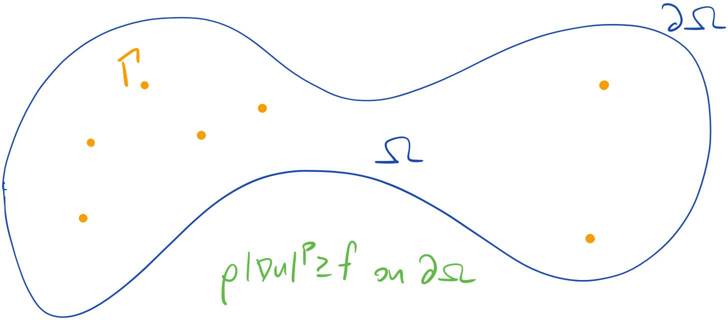
# Discrete to continuum

For $p \geq 1$ we consider the $p$-eikonal equation with arbitrary right hand side $f$:

$$\begin{cases} \mathcal{A}_{n,\varepsilon} u(x) = f(x) & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = 0 & \text{if } x \in \Gamma. \end{cases}$$

**Continuum limit:** State-constrained eikonal equation [Capuzzo-Dolcetta & Lions, 1990]

$$\begin{cases} \rho |\nabla u|^p = f & \text{in } \Omega \setminus \Gamma \\ u = 0 & \text{on } \Gamma. \end{cases}$$



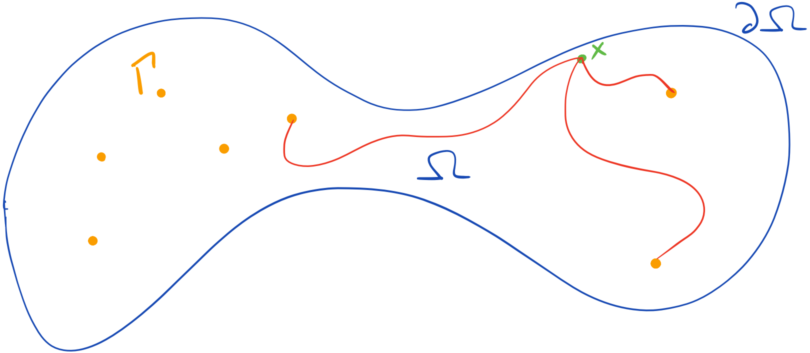
# Discrete to continuum

Define the geodesic weighted distance

$$d_f(x,y) := \inf\left\{\int_0^1 f(\gamma(t))|\gamma'(t)|\,dt \;:\; \gamma \in C^1([0,1];\overline{\Omega}), \gamma(0)=x, \text{ and } \gamma(1)=y\right\}.$$

and set

$$u(x) = \min_{y\in\Gamma} d_f(x,y).$$



# Discrete to continuum

Define the geodesic weighted distance

$$d_f(x,y) := \inf\left\{ \int_0^1 f(\gamma(t))|\gamma'(t)|\,dt \,:\, \gamma \in C^1([0,1];\overline{\Omega}), \gamma(0) = x, \text{ and } \gamma(1) = y \right\}.$$

and set

$$u(x) = \min_{y\in\Gamma} d_f(x,y).$$

Then $u$ is the unique viscosity solution of the state constrained eikonal equation

$$\begin{cases} |\nabla u| = f & \text{in } \Omega \setminus \Gamma \\ u = 0 & \text{on } \Gamma. \end{cases}$$

In particular, the solution of the continuum problem

$$\begin{cases} \rho|\nabla u|^p = f & \text{in } \Omega \setminus \Gamma \\ u = 0 & \text{on } \Gamma. \end{cases}$$

is given by $u(x) = d_g(x,\Gamma)$, where $g = \rho^{-\frac{1}{p}} f^{\frac{1}{p}}$.

# Discrete to continuum

Let $u_{n,\varepsilon}$ be the solution of

$$\begin{cases} \mathcal{A}_{n,\varepsilon} u_{n,\varepsilon}(x) = f(x) & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u_{n,\varepsilon}(x) = 0 & \text{if } x \in \Gamma. \end{cases}$$

## Theorem (Calder, Ettehad, 2022)

*There exists* $C, c > 0$ *such that for* $\varepsilon$ *sufficiently small and any* $0 < \lambda \leq 1$ *we have*

$$\mathbb{P}\left[\max_{x \in \mathcal{X}} (d_g(x, \Gamma) - u_{n,\varepsilon}(x)) \leq C(\sqrt{\varepsilon} + \lambda)\right] \geq 1 - 2n\exp(-cn\varepsilon^d\lambda^2).$$

*and*

$$\mathbb{P}\left[\max_{x \in \mathcal{X}} (u_{n,\varepsilon}(x) - d_g(x, \Gamma)) \leq C\left(\sqrt{\varepsilon} + \left(n\varepsilon^{p+d}\right)^{\frac{1}{p}} + \lambda\right)\right] \geq 1 - 3n^2\exp(-cn\varepsilon^d\lambda^2).$$

# Discrete to continuum

## Theorem (Calder, Ettehad, 2022)

*There exists $C, c > 0$ such that for $\varepsilon$ sufficiently small and any $0 < \lambda \leq 1$ we have*

$$\mathbb{P}\left[\max_{x\in\mathcal{X}}(d_g(x,\Gamma) - u_{n,\varepsilon}(x)) \leq C(\sqrt{\varepsilon}+\lambda)\right] \geq 1 - 2n\exp(-cn\varepsilon^d\lambda^2).$$

$$\mathbb{P}\left[\max_{x\in\mathcal{X}}(u_{n,\varepsilon}(x) - d_g(x,\Gamma)) \leq C\left(\sqrt{\varepsilon}+\left(n\varepsilon^{p+d}\right)^{\frac{1}{p}}+\lambda\right)\right] \geq 1 - 3n^2\exp(-cn\varepsilon^d\lambda^2).$$

In order for the results to be non-vacuous, we require that

(20) $$n\varepsilon^d \gg \log(n) \ \text{ and } \ n\varepsilon^{d+p} \ll 1$$

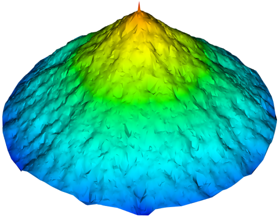
which can be reformulated as

(21) $$\left(\frac{\log(n)}{n}\right)^{\frac{1}{d}} \ll \varepsilon \ll \left(\frac{1}{n}\right)^{\frac{1}{p+d}}.$$

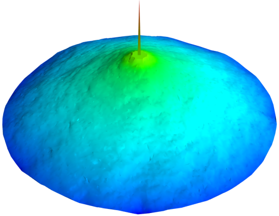
For any $p > 0$ we can find feasible $\varepsilon$ (we use $p \geq 1$).

A similar lower bound appears in $p$-Laplacian learning [Slepcev & Thorpe, 2019].

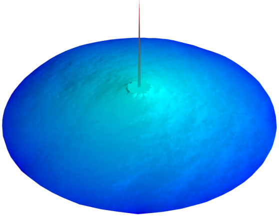
# Discrete to continuum



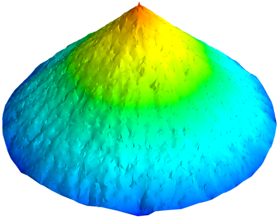
(c) $\varepsilon = 0.03, p = 1$



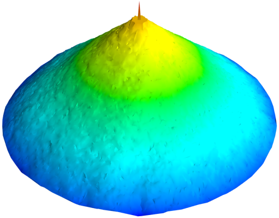
(d) $\varepsilon = 0.06, p = 1$



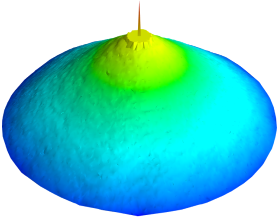
(e) $\varepsilon = 0.09, p = 1$



(f) $\varepsilon = 0.03, p = 2$



(g) $\varepsilon = 0.06, p = 2$



(h) $\varepsilon = 0.09, p = 2$

# Discrete to continuum

**Main ideas in proof:**

- Pointwise consistency $\mathcal{A}_{n,\varepsilon}\varphi(x) \approx \rho|\nabla\varphi|^p$ for smooth $\varphi$, with high probability.
- The $O(\sqrt{\varepsilon})$ rate comes from a doubling variables argument in the viscosity solutions framework.
- Rate requires Lipschitzness of $u_{n,\varepsilon}$, we show that

$$|u_{n,\varepsilon}(x) - u_{n,\varepsilon}(y)| \leq c_p\gamma_p^{-1}\max_{\mathcal{X}} f^{\frac{1}{p}}\, d_\Omega(x,y) + \gamma_p\left(n\varepsilon^{p+d}\right)^{\frac{1}{p}}, \quad \text{for all } x,y\in\mathcal{X}$$

  with probability at least $1 - n^2\exp\left(-\frac{c_d r^d}{2^{2d+3}}\rho_{min} n\varepsilon^d\right)$. The proof uses a geodesic cone barrier function with an additional spike:

$$v_{\beta,y}(x) := \beta(1-\delta_y(x)) + d_\Omega(x,y)$$

- State constrained boundary condition handled with domain perturbation results.

# Applications

Given a set $\Gamma \subset \mathcal{X}$ and a density estimation $\hat{\rho} : \mathcal{X} \to \mathbb{R}$, we consider solving the density reweighted $p$-eikonal equation

$$\begin{cases} \mathcal{A}_{G,p} u = \hat{\rho}^{-\alpha}, & \text{in } \mathcal{X} \setminus \Gamma \\ u = 0, & \text{on } \Gamma, \end{cases} \tag{22}$$

where the exponent $\alpha$ is a tunable parameter. We denote the solution of (22) by

$$D_{\Gamma}^{p,\alpha}(x) = u(x).$$

When $\Gamma = \{x\}$ is a single point we write $D_x^{p,\alpha}$.

# Data depth

Recall the **geometric median**:

$$x_* \in \arg\min_{x \in \mathbb{R}^d} \sum_{i=1}^{n} |x_i - x|.$$

We can generalize this to the $p$-eikonal graph setting as follows:

$$x_{p,\alpha} \in \arg\min_{x \in \mathcal{X}} \sum_{x_i \in \mathcal{X}} D_x^{p,\alpha}(x_i).$$

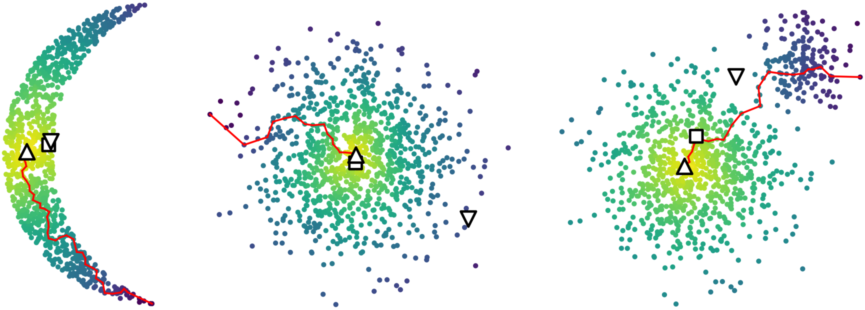
Then we can define data depth as the distance to the median

$$\mathsf{depth}_{p,\alpha}(x) = \max_{\mathcal{X}} D_{x_{p,\alpha}}^{p,\alpha} - D_{x_{p,\alpha}}^{p,\alpha}(x).$$

**Note:** Other approaches include first finding the "boundary" nodes and defining depth as distance to the boundary.

- [Calder, Park, & Slepcev, 2021]
- [Molina-Fructuoso and Murray, 2022]

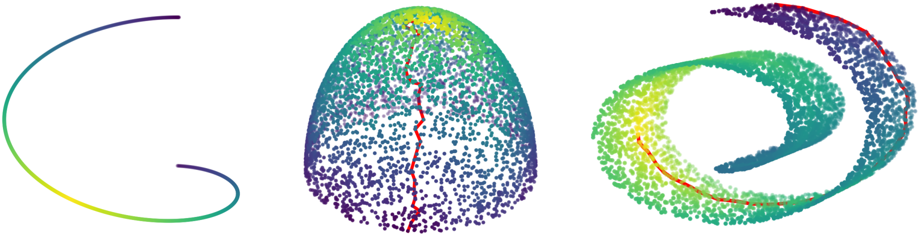
# Data depth



(i) Moon (j) Gaussian (k) Gaussian mixture

Figure: The $p$-eikonal medians and depth on 2D toy datasets with $p = 1$. The medians are shown for $\alpha = -1$ ($\triangledown$), $\alpha = 0$ ($\square$) and the $\alpha = 1$ ($\triangle$), while the points are colored by the $\alpha = 1$ data depth.

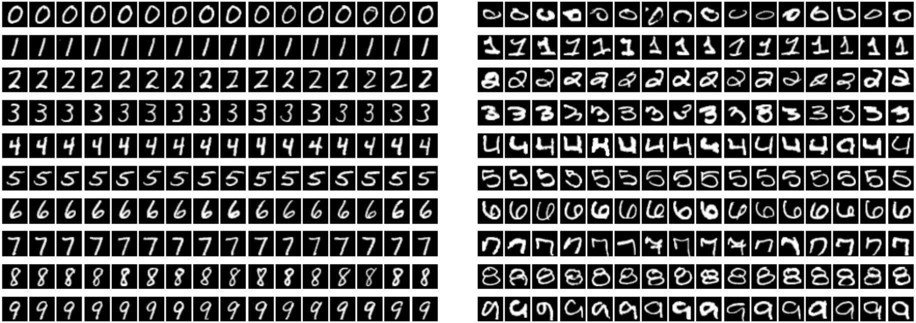
# Data depth



(a) Helix (b) Half Sphere (c) Swiss Roll

Figure: The $p$-eikonal data depth on 3D toy datasets sampled from manifolds embedded in $\mathbb{R}^3$. We use $p = 1$ and $\alpha = 1$.

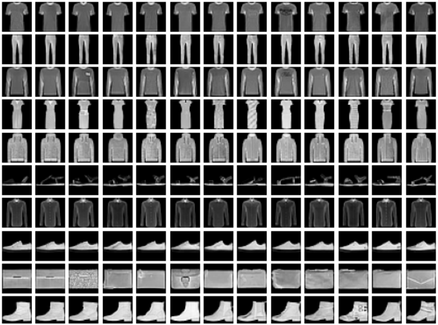
# Data depth



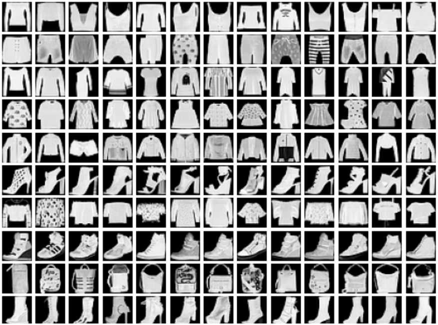
(a) Deepest images (median) (b) Shallowest images (outliers)

Figure: Comparison of deepest (median) images to shallowest (outlier) images from each MNIST digit.

# Data depth



(a) Deepest images (median)



(b) Shallowest images (outliers)

Figure: Comparison of deepest (median) images to shallowest (outlier) images from each FashionMNIST class.

# Data depth

(a) MNIST (b) FashionMNIST

Figure: Paths from shallowest point to median for each class.

# Semi-supervised learning

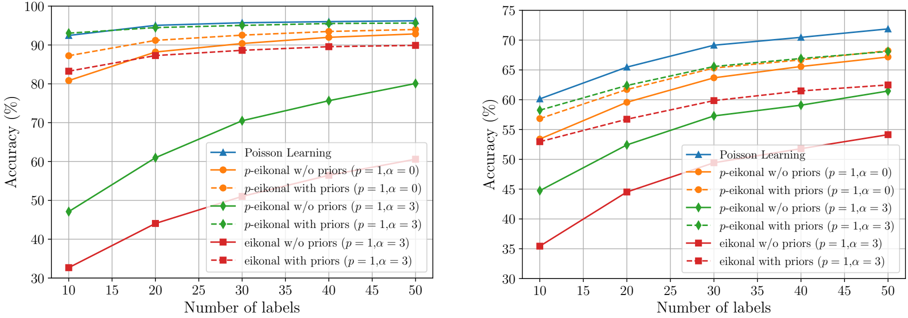
- Suppose we have $k$ classes, and for each class $j = 1, \ldots, k$, we are provided some labeled nodes $\Gamma_j \subset \mathcal{X}$.
- The label prediction $\ell_i$ for an unlabeled node $x_i \notin \Gamma_j$ for any $j$, is the label of the closest labeled node, under the distance $D_\Gamma^{p,\alpha}$, that is

$$\ell_i = \operatorname*{arg\,min}_{1 \le j \le k} D_{\Gamma_j}^{p,\alpha}(x_i).$$

- We can incorporate prior information about class sizes by introducing weights $s_j$ in the label decision [Calder et al, 2020]

$$\ell_i = \operatorname*{arg\,min}_{1 \le j \le k} \left\{ s_j D_{\Gamma_j}^{p,\alpha}(x_i) \right\}.$$

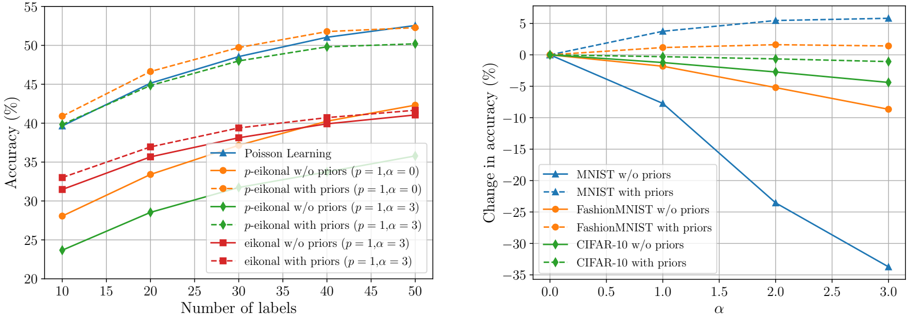
# Semi-supervised learning




(a) MNIST (b) FashionMNIST

Figure: Comparison of the $p$-eikonal equation with $p = 1$ for semi-supervised image classification to Poisson learning [Calder et al., 2020] and the eikonal equation.

# Semi-supervised learning




(a) CIFAR-10 (b) Accuracy vs $\alpha$

Figure: (a) Accuracy results for the $p$-eikonal equation with $p = 1$ for semi-supervised image classification on CIFAR-10, and (b) change in accuracy as the density reweighting exponent $\alpha$ is adjusted.

# Paper and Code

**Paper:**

J. Calder & M. Ettehad (2022). **Hamilton-Jacobi equations on graphs with applications to semi-supervised learning and data depth**. arXiv:2202.08789.

**Code** for all experiments is on GitHub

`https://github.com/jwcalder/peikonal`

The $p$-eikonal equation is implemented in the **GraphLearning** python package

`https://github.com/jwcalder/GraphLearning` (pip install graphlearning)