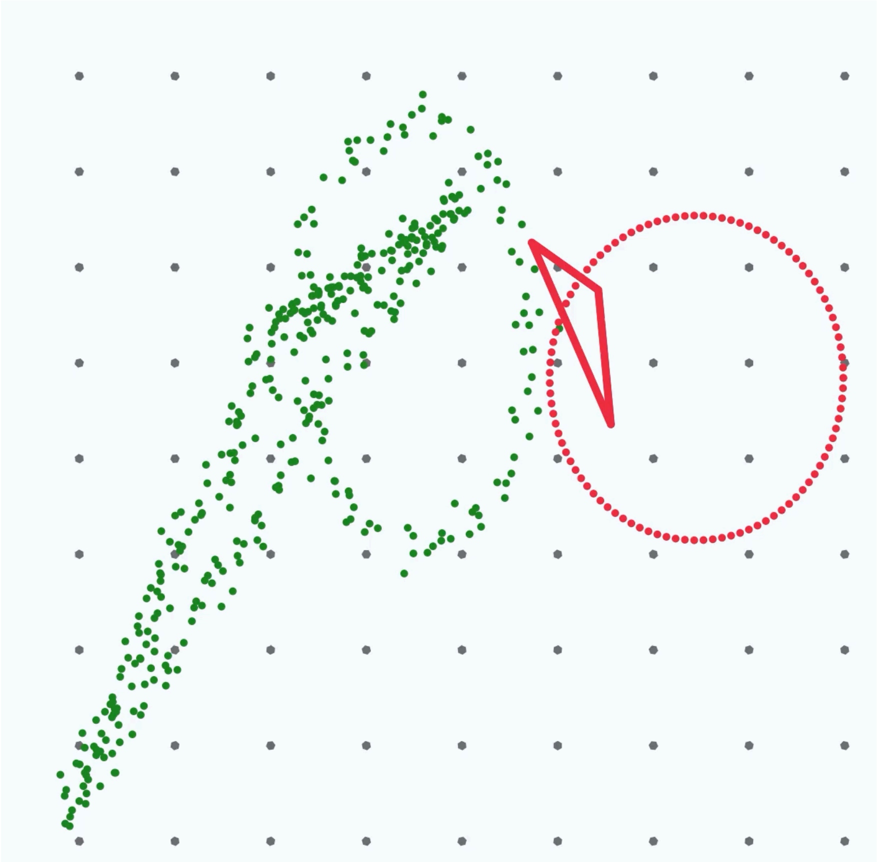
Neural Estimation of Energy Mover's Distance

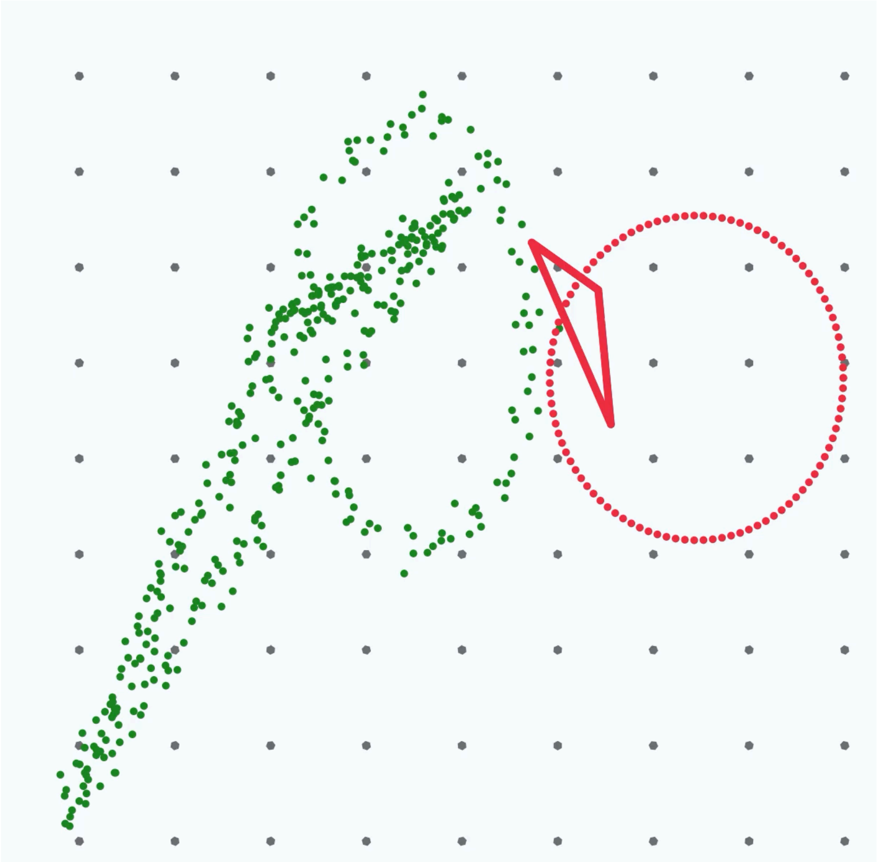
NEEMo

Ouail Kitouni

Fitting Arbitrary Geometries



Fitting Arbitrary Geometries



Robust and Provably Monotonic Networks

Ouail Kitouni[*], Niklas Nolte[*], Mike Williams
NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Laboratory for Nuclear Science, MIT

Finding NEEMo: Geometric Fitting using Neural Estimation of the Energy Mover's Distance

Ouail Kitouni, Niklas Nolte, Mike Williams
The NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{kitouni,nnolte,mwill}@mit.edu

Niklas Nolte

Ouail Kitouni

Mike Williams

Robust and Provably Monotonic Networks

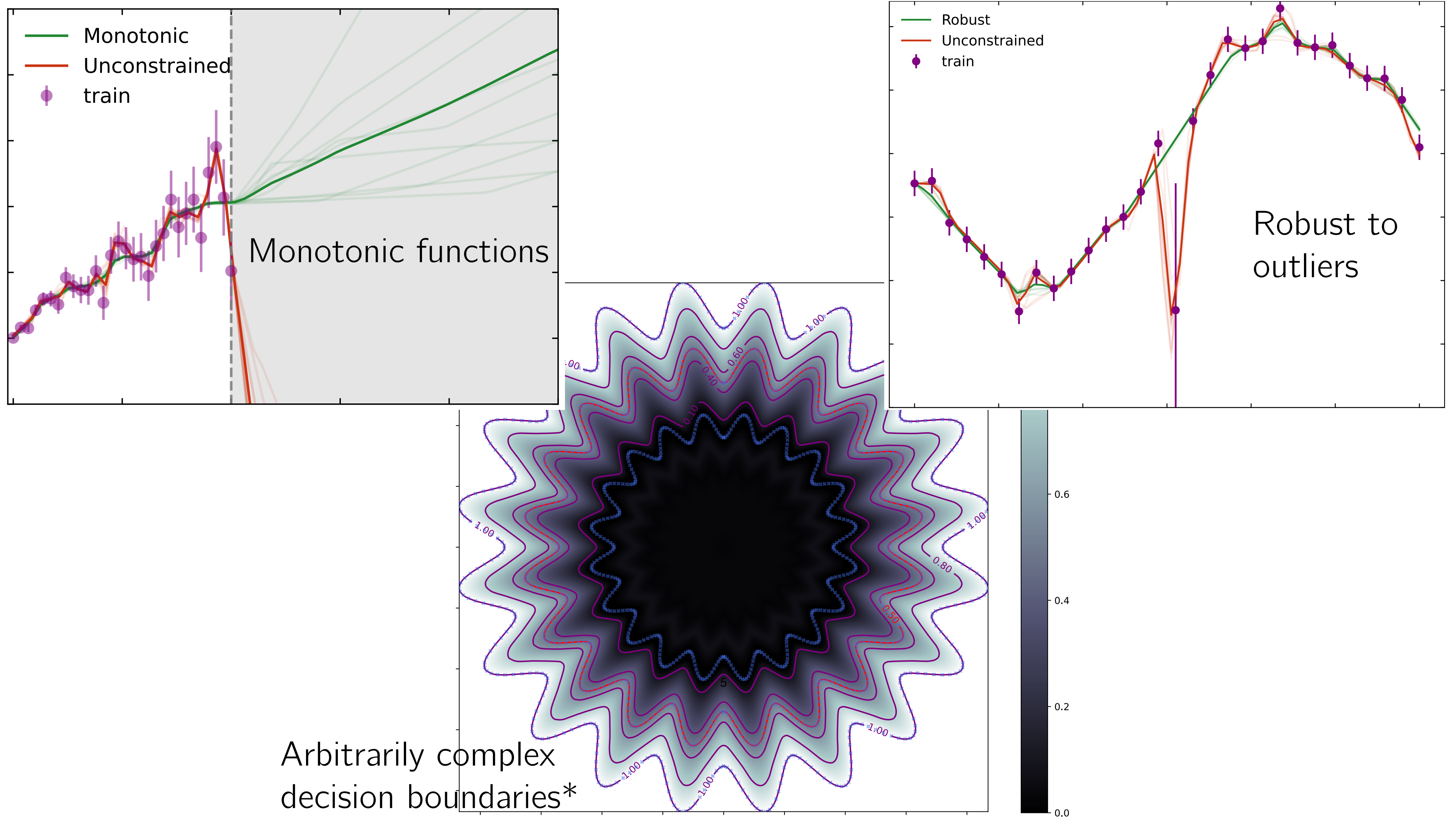
Ouail Kitouni,* Niklas Nolte,* Mike Williams
NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Laboratory for Nuclear Science, MIT

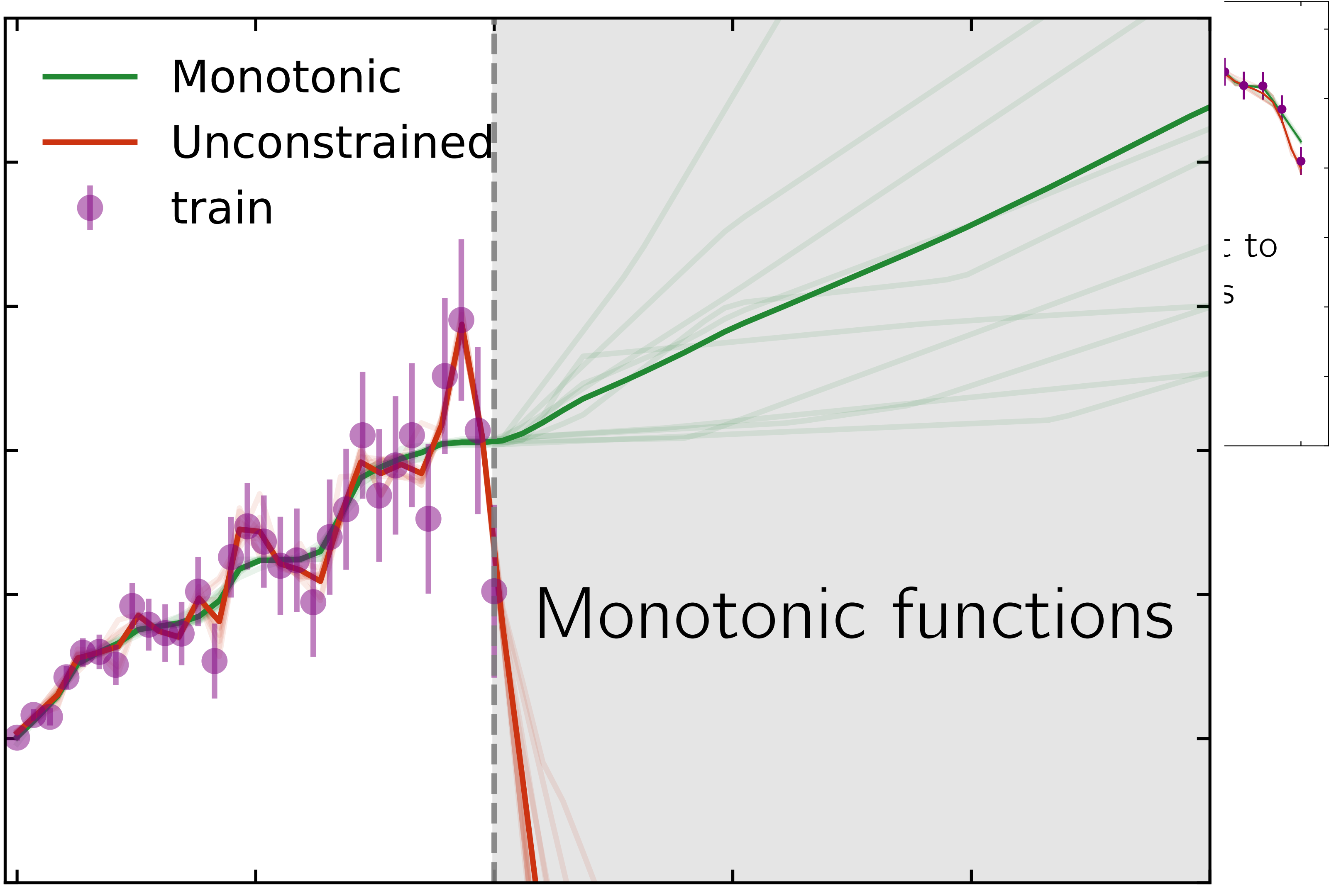
Niklas Nolte

Ouail Kitouni

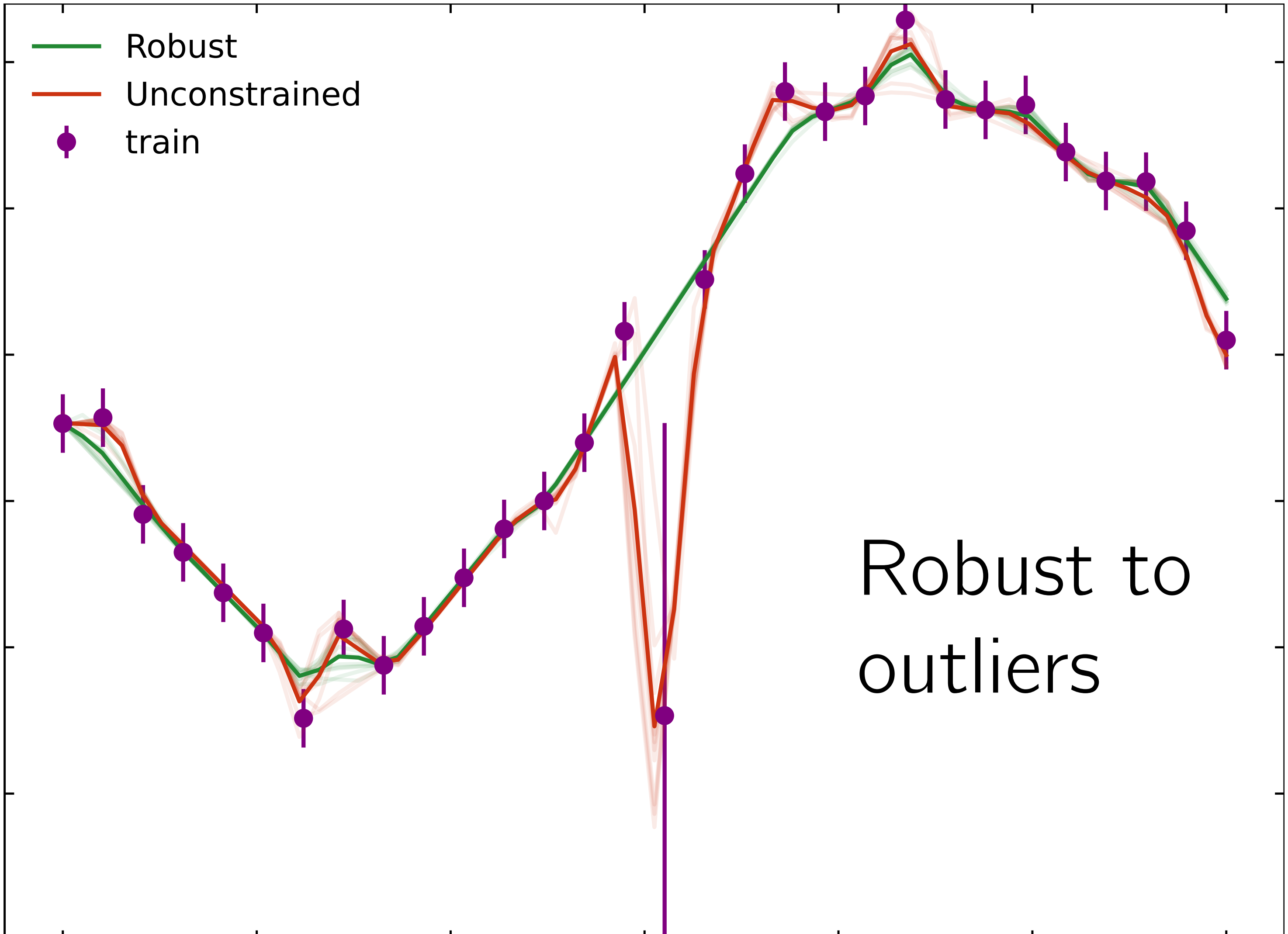
Mike Williams

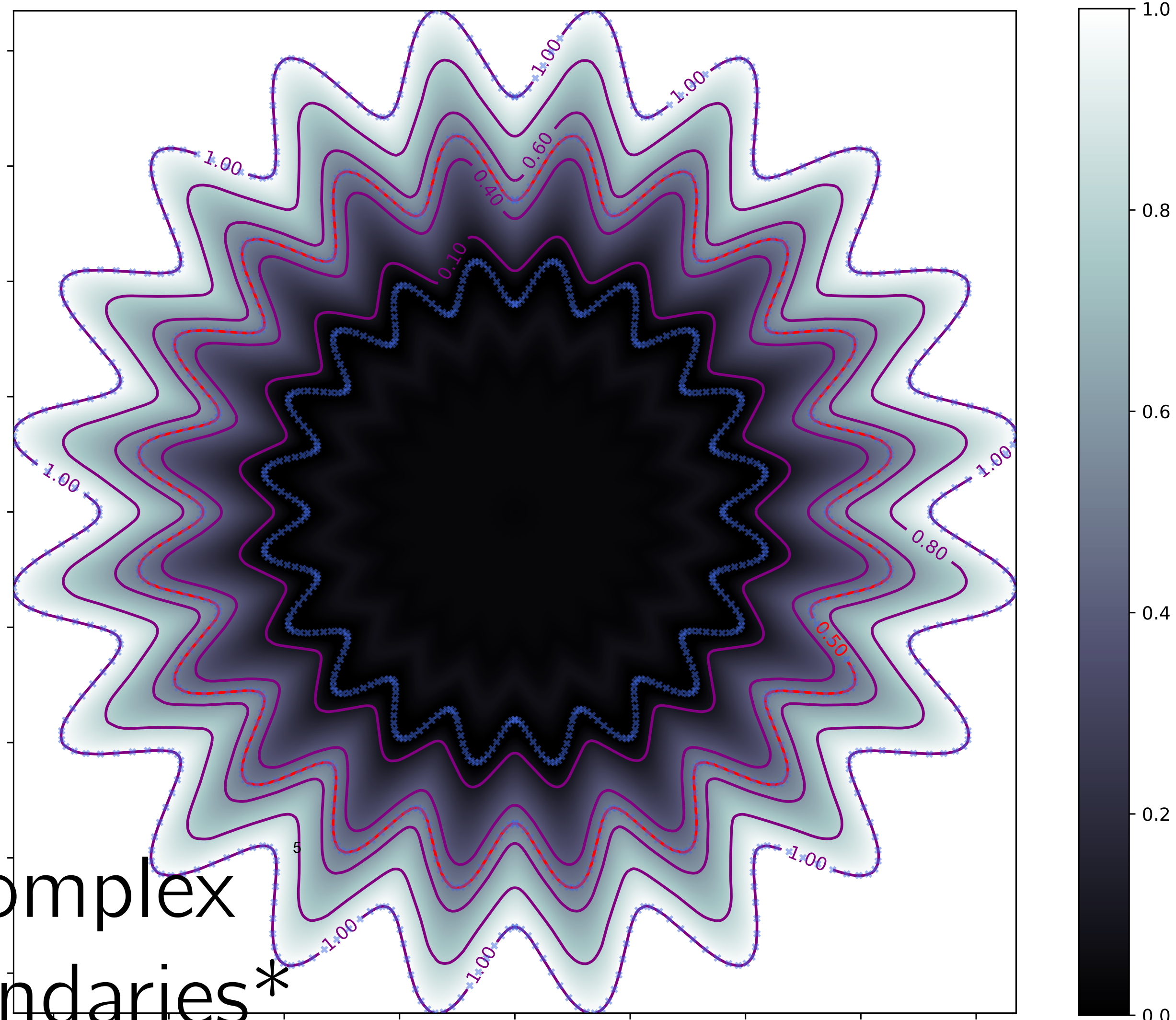
Part 1
Lipschitz
Networks


Monotonic
Unconstrained
train
Monotonic functions
Robust
Unconstrained
train
Robust to outliers
Arbitrarily complex decision boundaries*
1.00
0.60
0.40
0.10
0.80
0.50
0.6
0.4
0.2
0.0



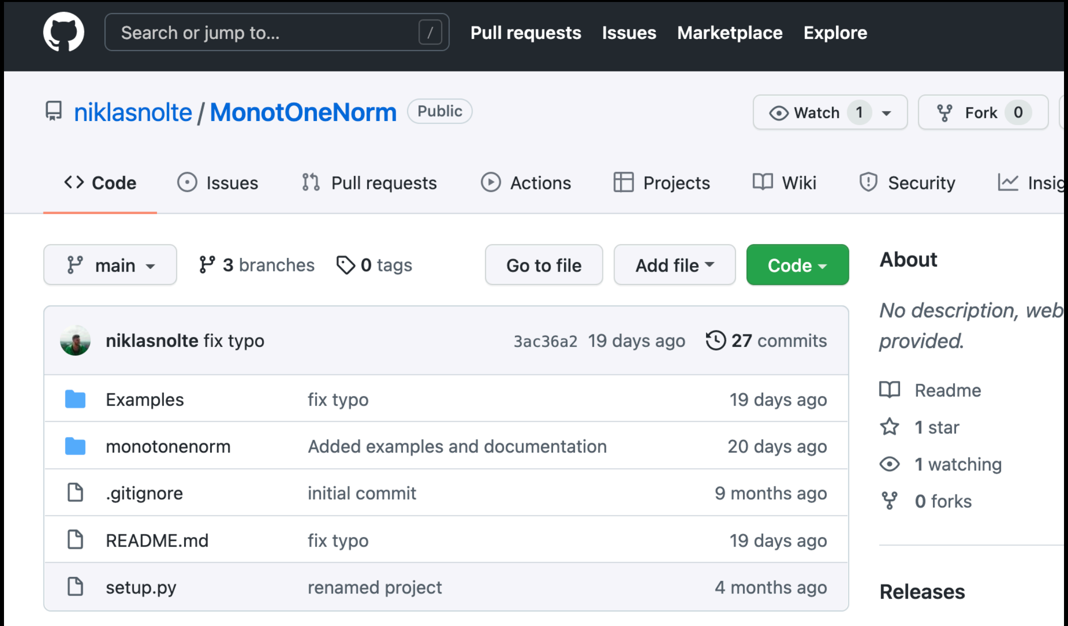
Monotonic
Unconstrained
train
Monotonic functions


Robust
Unconstrained
train
Robust to
outliers



Arbitrarily complex decision boundaries*

MontoneNorm



https://github.com/niklasnolte/MonotOneNorm

pip install monotonenorm
conda install monotonenorm -c okitouni

How does it work?

Robustness: Definition (more formally)

Small changes in the input should not lead to large changes in the output:

$$|f(x+\epsilon) - f(x)| \leq \lambda\epsilon \qquad \forall \epsilon > 0$$

Thus we would like our Neural Network to represent a Lipschitz continuous function.

Robustness: Definition (more formally)

Robustness is achieved by constraining the operator 1-norm of the weight matrices of each layer such that

$$\prod_{l=0}^{L} \|W^l\|_1 \leq \lambda$$

where λ is Lipschitz constant of the resulting network with respect to the ∞-norm.
Universal Lipschitz-λ function approximation requires activations with gradient 1 almost everywhere.
→ **GroupSort***: reorders inputs

*Sorting out Lipschitz function approximation [https://arxiv.org/abs/1811.05381]

Monotonicity using weight norm

g(x) is a λ-Lipschitz neural network. Adding the following residual connection

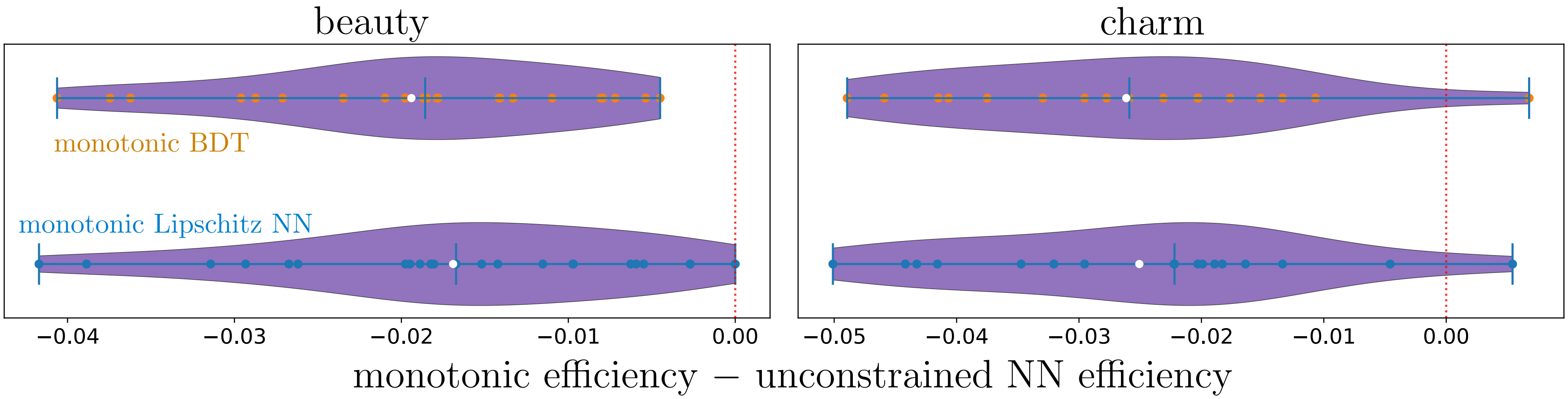
$$f(\mathbf{x}) = g(\mathbf{x}) + \lambda \sum_{i \in I} x_i$$

makes output monotonic since

$$\frac{\partial f}{\partial x_i} = \frac{\partial g}{\partial x_i} g(\mathbf{x}) + \lambda \geq 0 \quad \forall i \in I$$

Monotonic Lipschitz Networks LHCb RUN 3 trigger

This architecture is being used in the LHCb heavy-flavor RUN 3trigger.



Part 2
Neural **E**stimation of **E**nergy **Mo**ver's distance

Robust and Provably Monotonic Networks

Ouail Kitouni,* Niklas Nolte,* Mike Williams
NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Laboratory for Nuclear Science, MIT

Finding NEEMo: Geometric Fitting using Neural Estimation of the Energy Mover's Distance

Ouail Kitouni, Niklas Nolte, Mike Williams
The NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{kitouni,nnolte,mwill}@mit.edu

Niklas Nolte

Ouail Kitouni

Mike Williams

Finding NEEMo: Geometric Fitting using Neural Estimation of the Energy Mover's Distance

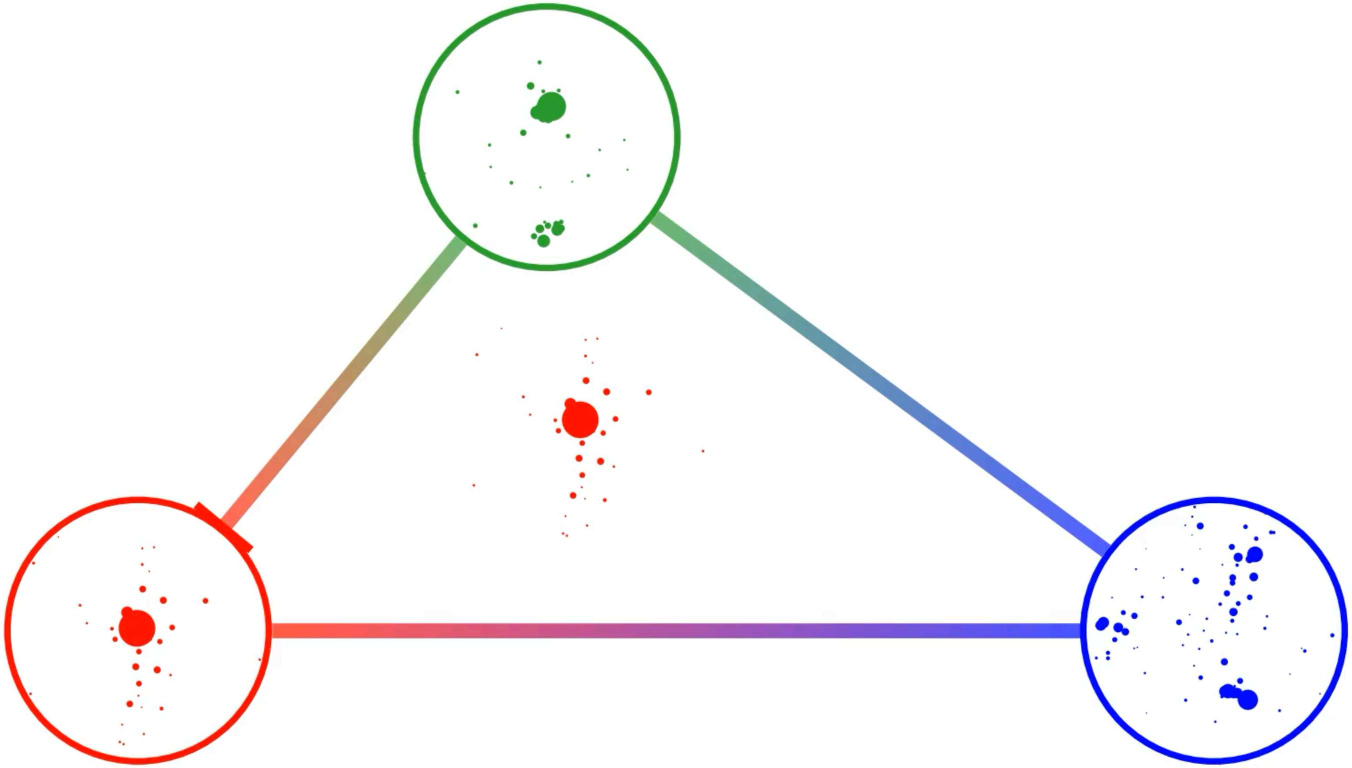
Ouail Kitouni, Niklas Nolte, Mike Williams
The NSF AI Institute for Artificial Intelligence and Fundamental Interactions
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{kitouni,nnolte,mwill}@mit.edu

Niklas Nolte

Ouail Kitouni

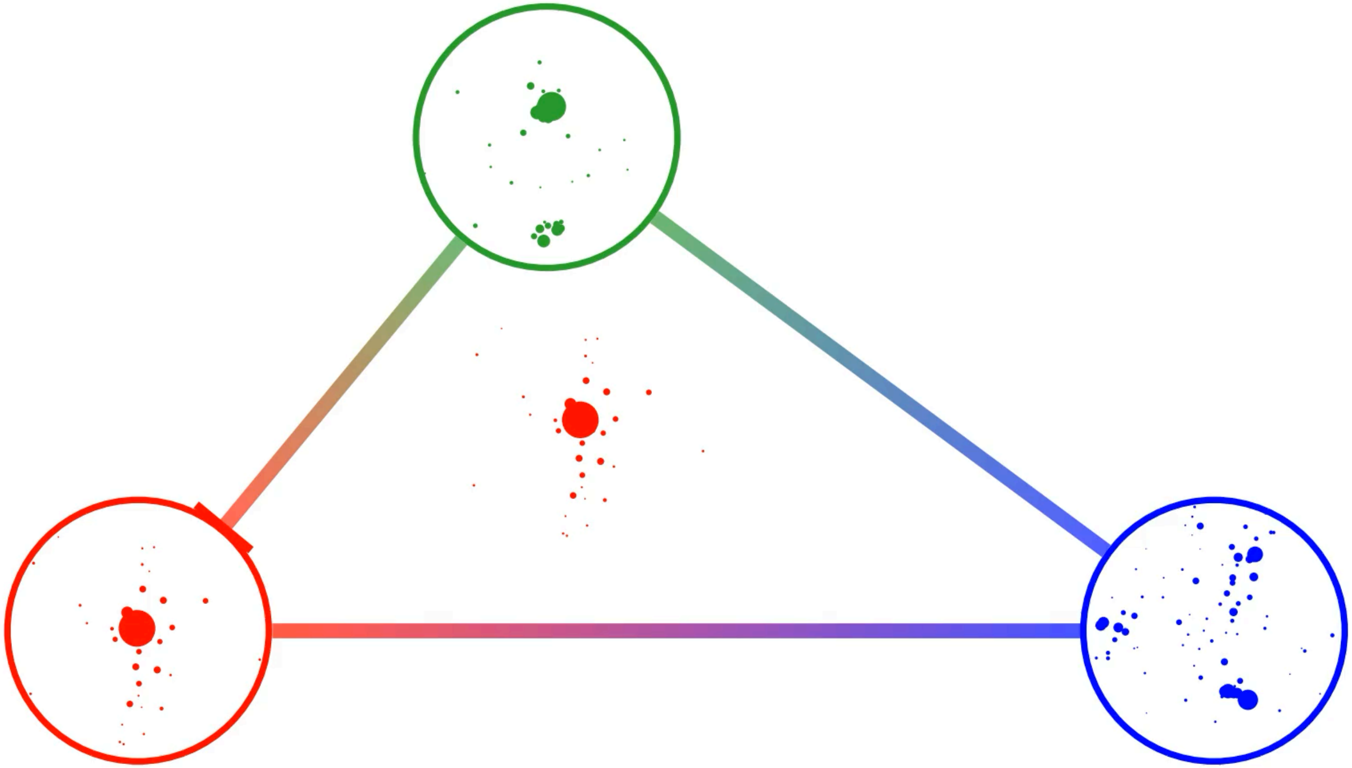
Mike Williams

Optimal Transport - Energy Mover's Distance



The Metric Space of Collider Events [arxiv.org/abs/1902.02346]

Optimal Transport - Energy Mover's Distance



The Metric Space of Collider Events [arxiv.org/abs/1902.02346]

Earth Mover's Distance

The primal formulation of the EMD is an optimization over joint probability distributions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \Pi(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y) \sim \gamma} \left[||x - y||_2 \right],$$

Earth Mover's Distance

The primal formulation of the EMD is an optimization over joint probability distributions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \Pi(\mathbb{P},\mathbb{Q})} \mathbb{E}_{(x,y)\sim\gamma}\left[||x - y||_2 \right],$$

Event 1

Earth Mover's Distance

The primal formulation of the EMD is an optimization over joint probability distributions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \Pi(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y)\sim\gamma}\left[\,||x - y||_2\right],$$

Event 1

Event 2

Earth Mover's Distance

The primal formulation of the EMD is an optimization over joint probability distributions

$$\mathrm{EMD}(\underbrace{\mathbb{P}}_{\text{Event 1}}, \underbrace{\mathbb{Q}}_{\text{Event 2}}) = \inf_{\gamma \in \Pi(\mathbb{P},\mathbb{Q})} \mathbb{E}_{(x,y)\sim \underbrace{\gamma}_{\text{Energies}}} \left[||x - y||_2 \right],$$

Earth Mover's Distance

The primal formulation of the EMD is an optimization over joint probability distributions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \Pi(\mathbb{P},\mathbb{Q})} \mathbb{E}_{(x,y)\sim\gamma}\left[\,||x - y||_2\right],$$

Event 1

Event 2

Energies

Spatial Coordinates

Kantorovich-Rubinstein - Dual Formulation

The dual formulation is an optimization over 1-Lipschitz continuous functions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}}\left[f(x)\right] - \mathbb{E}_{x \sim \mathbb{Q}}\left[f(x)\right],$$

Kantorovich-Rubinstein - Dual Formulation

The dual formulation is an optimization over 1-Lipschitz continuous functions

$$\mathrm{EMD}(\mathbb{P}, \mathbb{Q}) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}}\big[f(x)\big] - \mathbb{E}_{x \sim \mathbb{Q}}\big[f(x)\big],$$

Kantorovich potential

NEEMo Algorithm

Parametrized shape:

θ

Target Distribution

$\mathbb{Q} = \{e^i, \mathbf{x}^i\}_{i=1}^n$

Forward pass $\longrightarrow$

Backward pass $\dashrightarrow$

NEEMo Algorithm

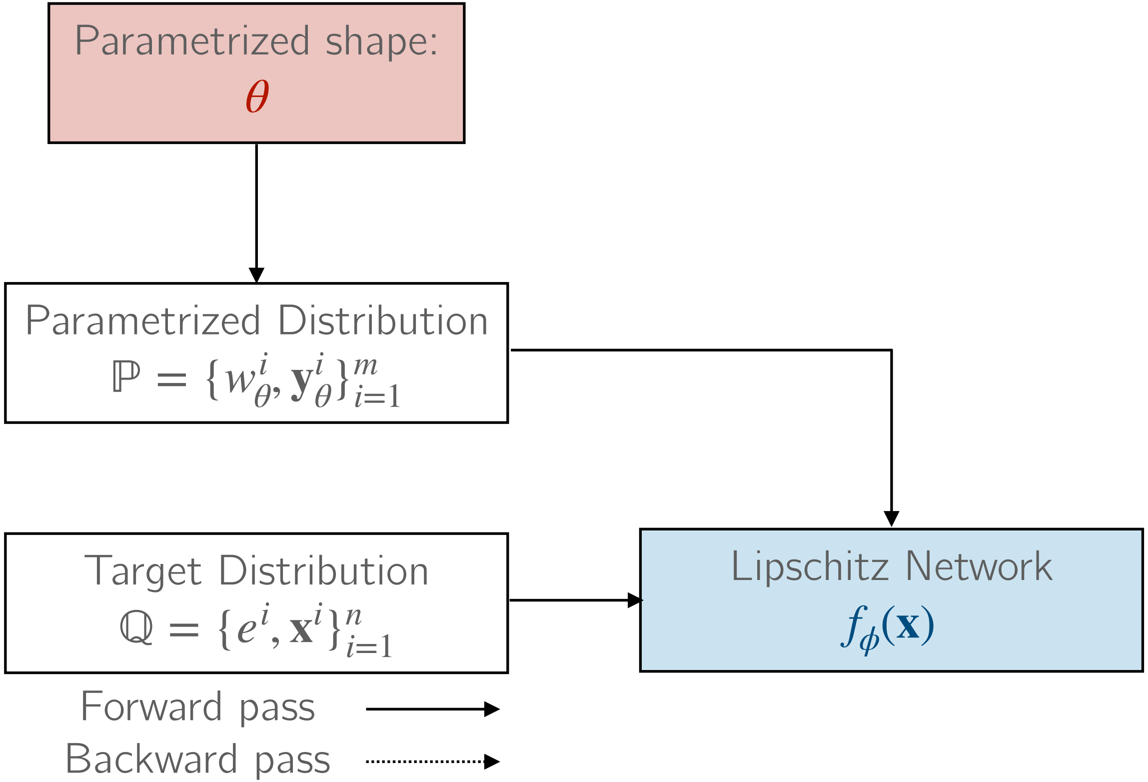
Parametrized shape:
θ

Parametrized Distribution
$\mathbb{P} = \{w_\theta^i, \mathbf{y}_\theta^i\}_{i=1}^m$

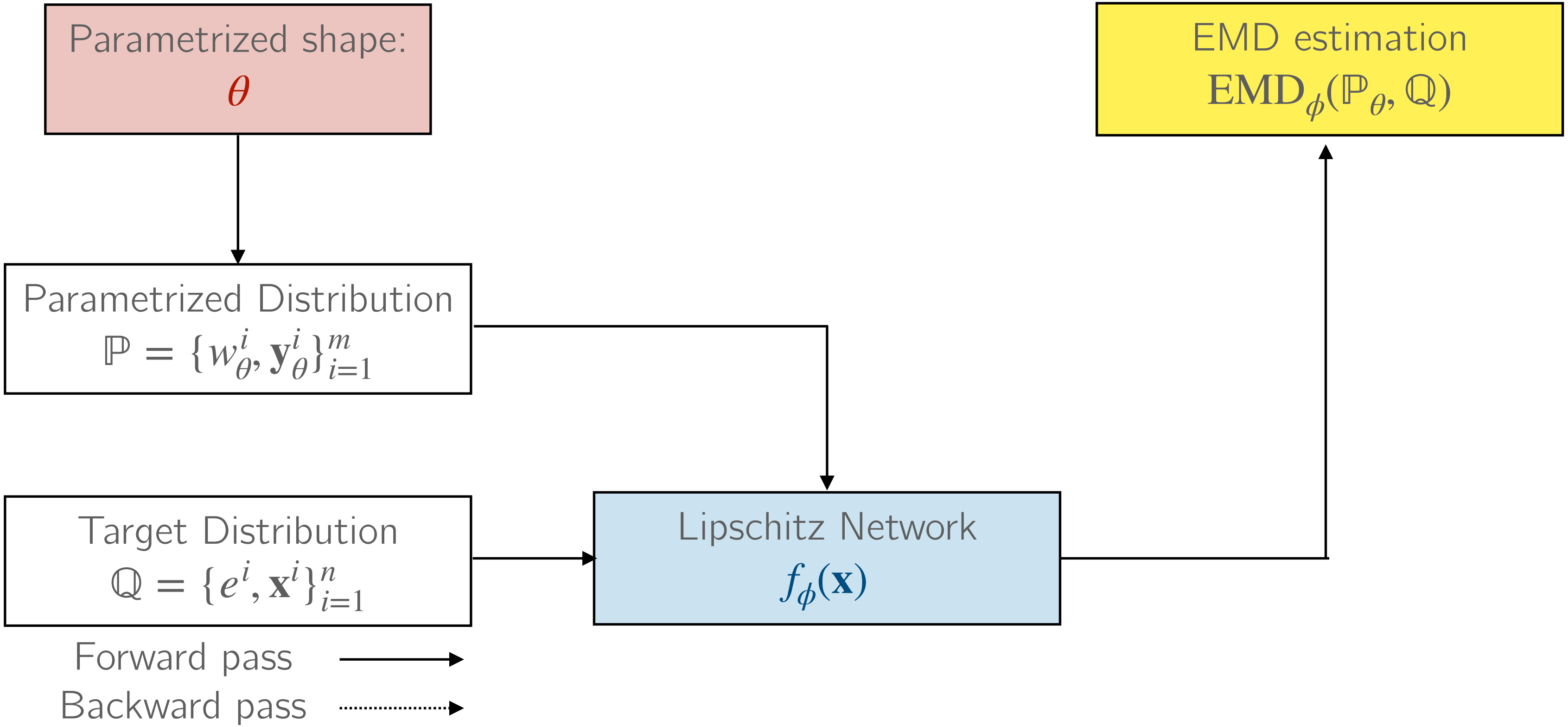
Target Distribution
$\mathbb{Q} = \{e^i, \mathbf{x}^i\}_{i=1}^n$

Forward pass $\longrightarrow$
Backward pass $\dashrightarrow$

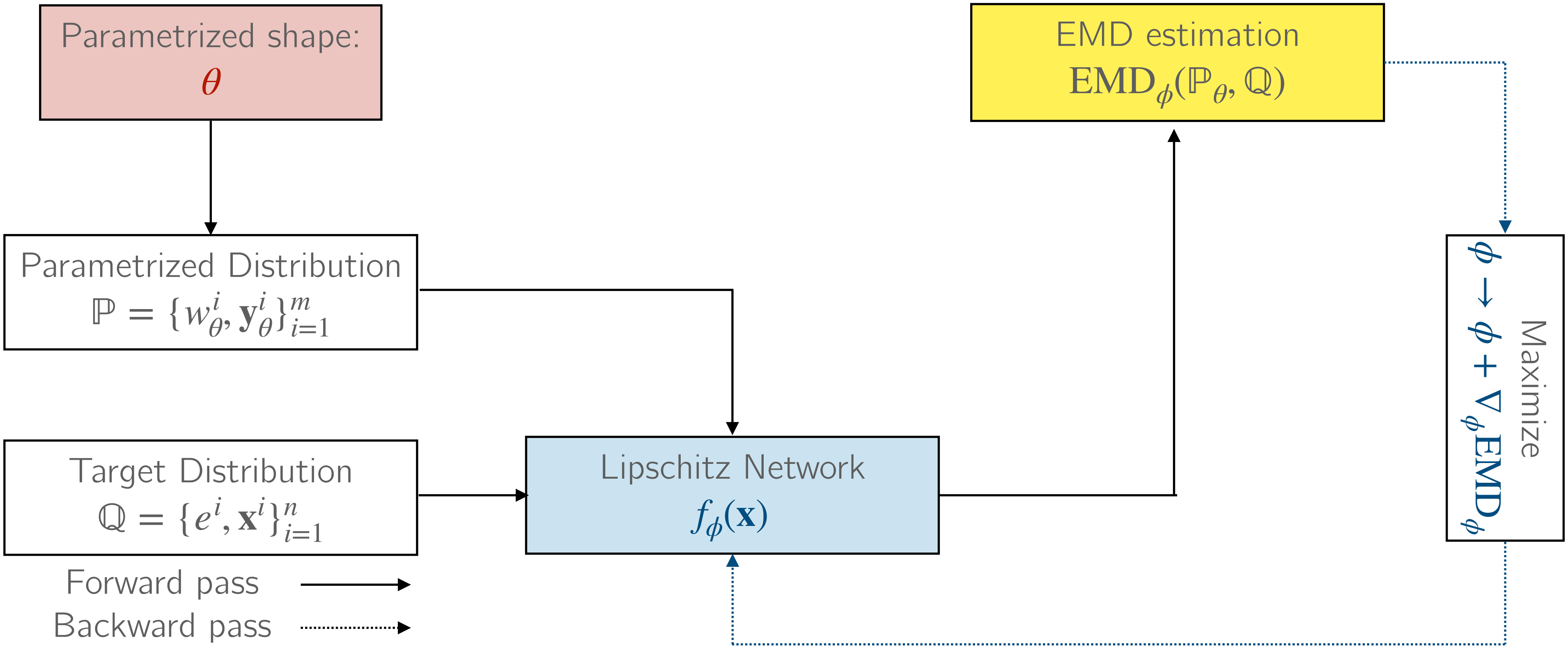
NEEMo Algorithm



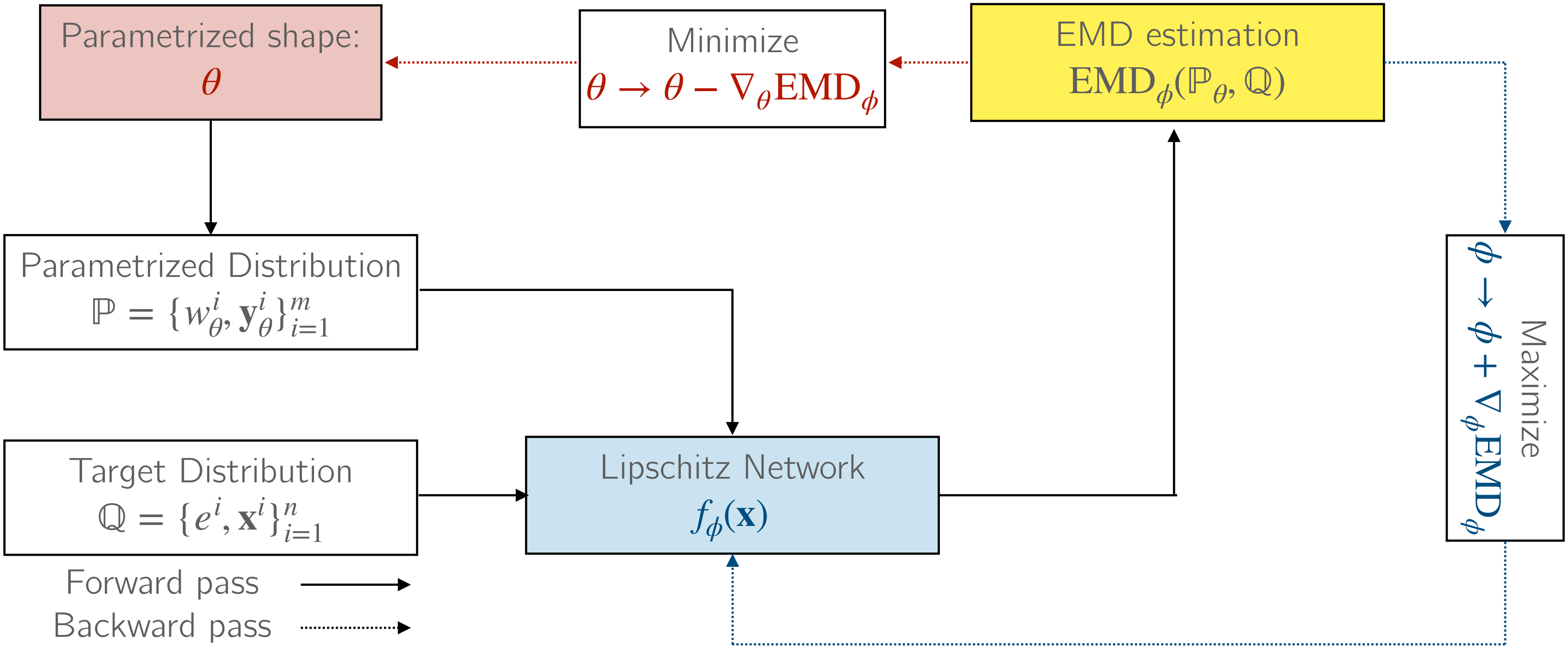
NEEMo Algorithm



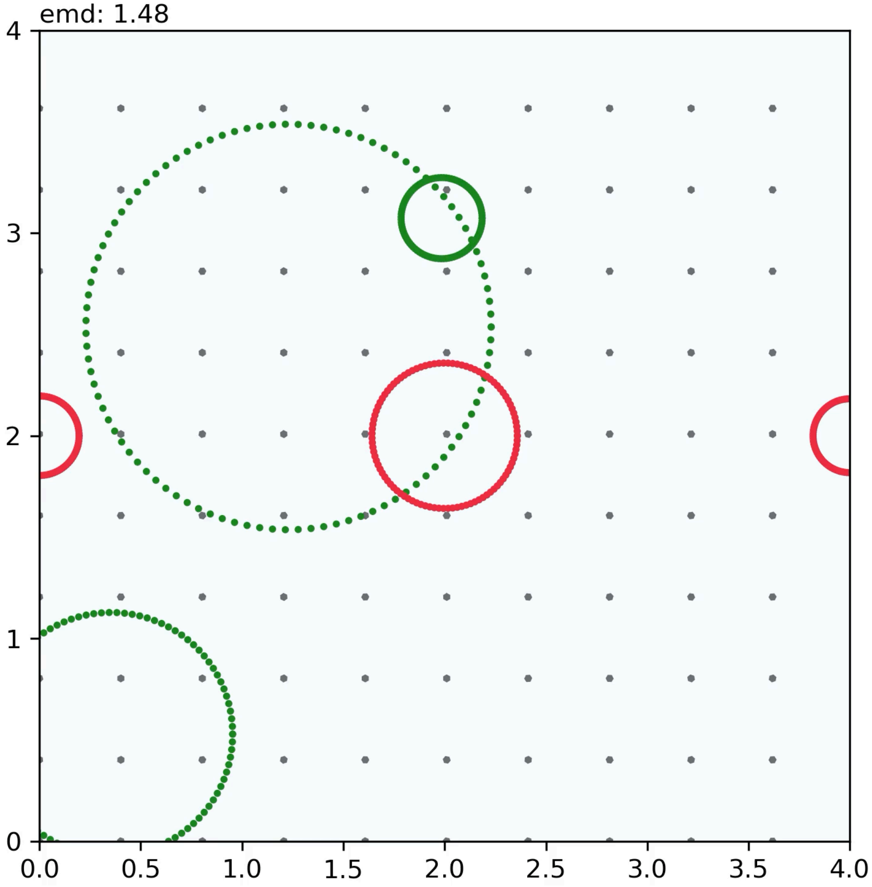
NEEMo Algorithm



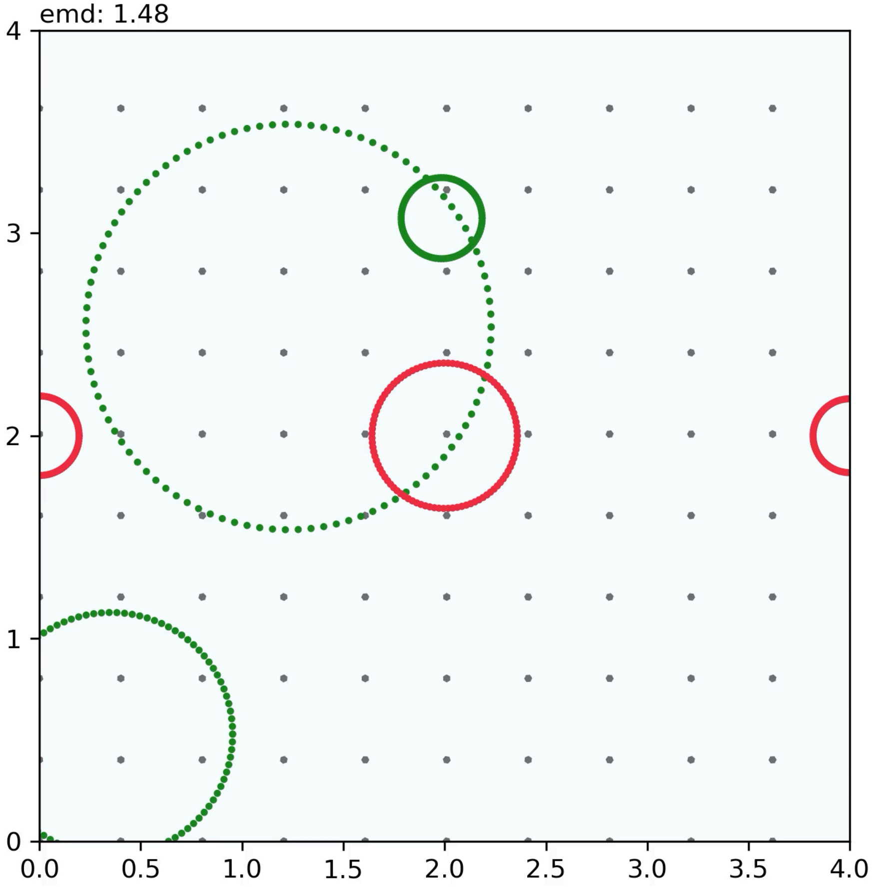
NEEMo Algorithm



Fitting Arbitrary Geometries

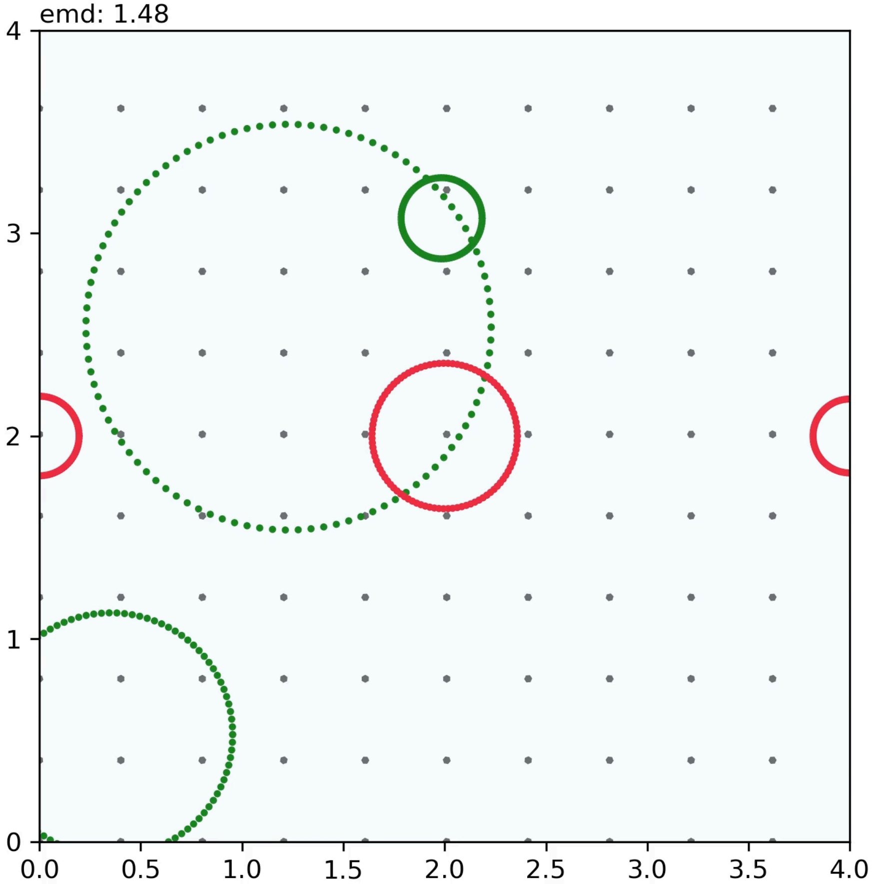


Fitting Arbitrary Geometries



Fitting Arbitrary Geometries

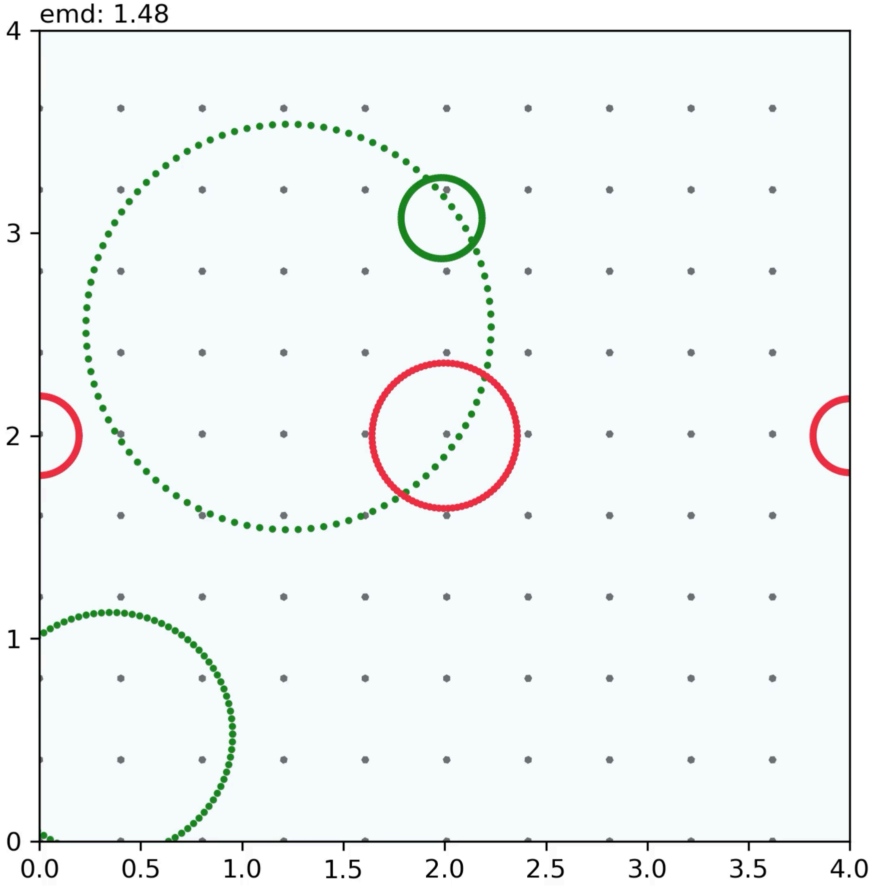
Possible use cases:



Fitting Arbitrary Geometries

Possible use cases:

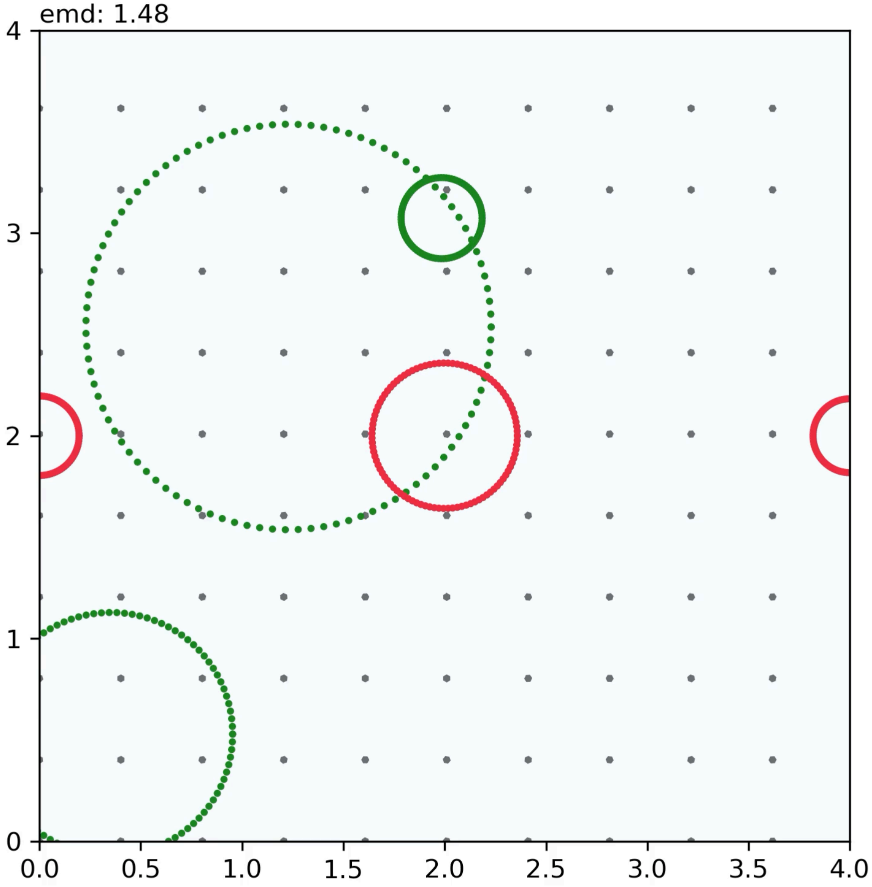
- Floating term for a uniform background to mitigate pileup



Fitting Arbitrary Geometries

Possible use cases:

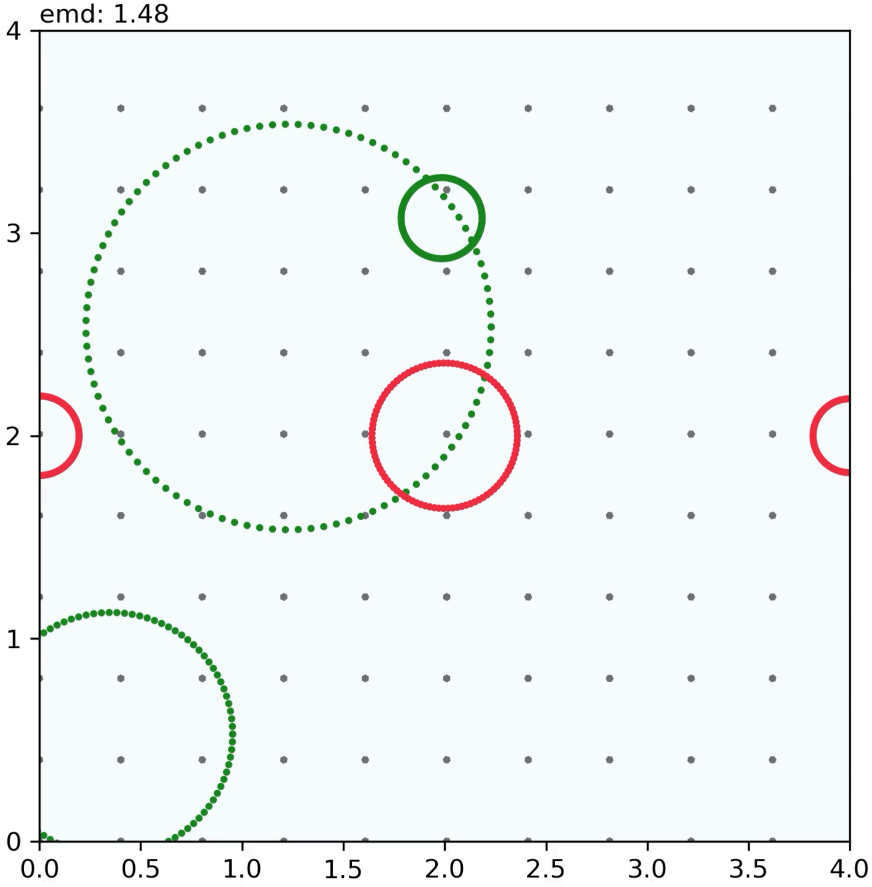
- Floating term for a uniform background to mitigate pileup
- Clustering with jet energy estimation



Fitting Arbitrary Geometries

Possible use cases:

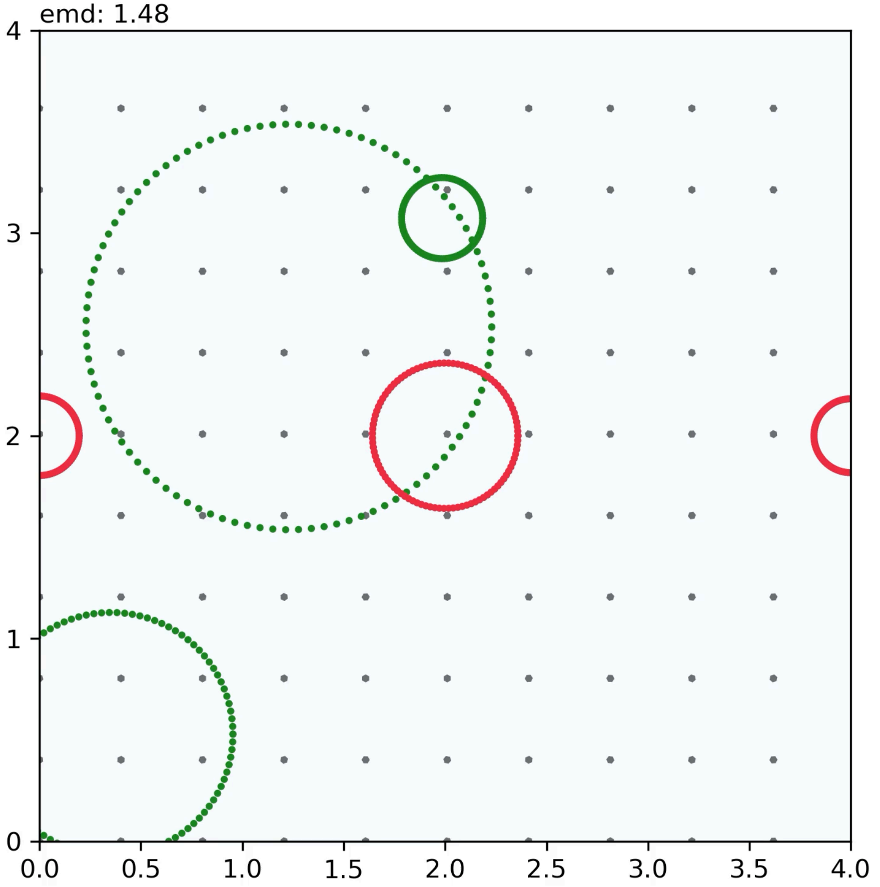
- Floating term for a uniform background to mitigate pileup
- Clustering with jet energy estimation
- Variable shape clusters (learned radii, ellipses, arbitrary shapes, etc.)



Fitting Arbitrary Geometries

Possible use cases:

- Floating term for a uniform background to mitigate pileup
- Clustering with jet energy estimation
- Variable shape clusters (learned radii, ellipses, arbitrary shapes, etc.)
- New (and old) observables: N-subjet, N-circle, triangularness, etc.

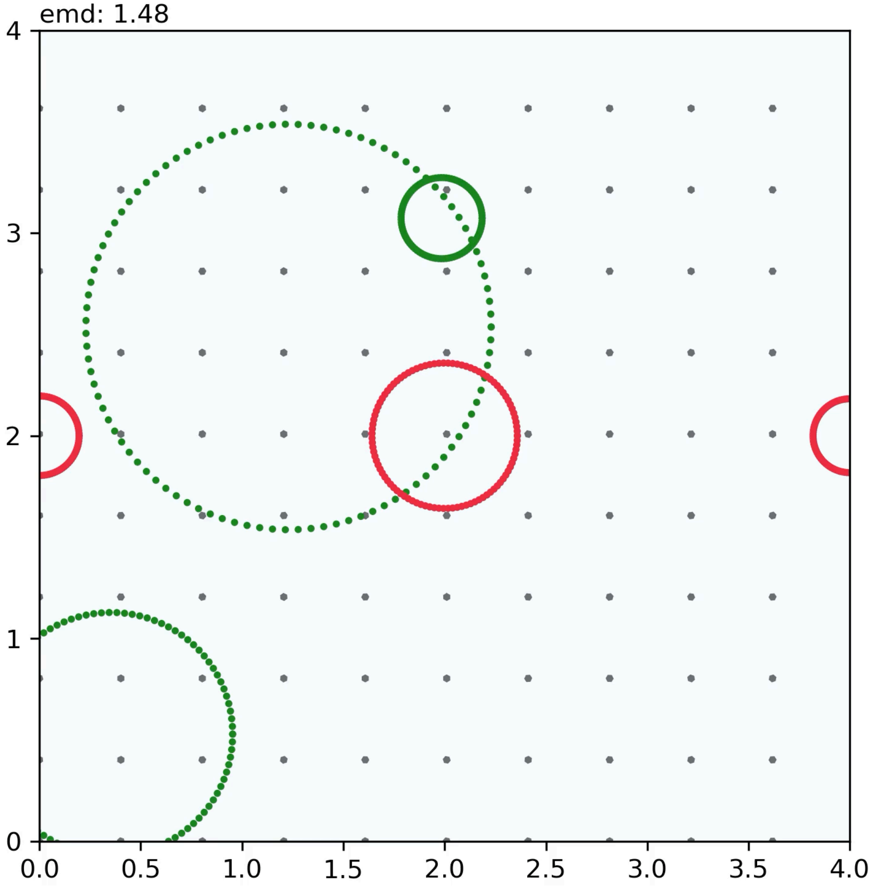


Fitting Arbitrary Geometries

Possible use cases:

- Floating term for a uniform background to mitigate pileup
- Clustering with jet energy estimation
- Variable shape clusters (learned radii, ellipses, arbitrary shapes, etc.)
- New (and old) observables: N-subjet, N-circle, triangularness, etc.

All in a unified framework given by the Energy Mover's Distance*

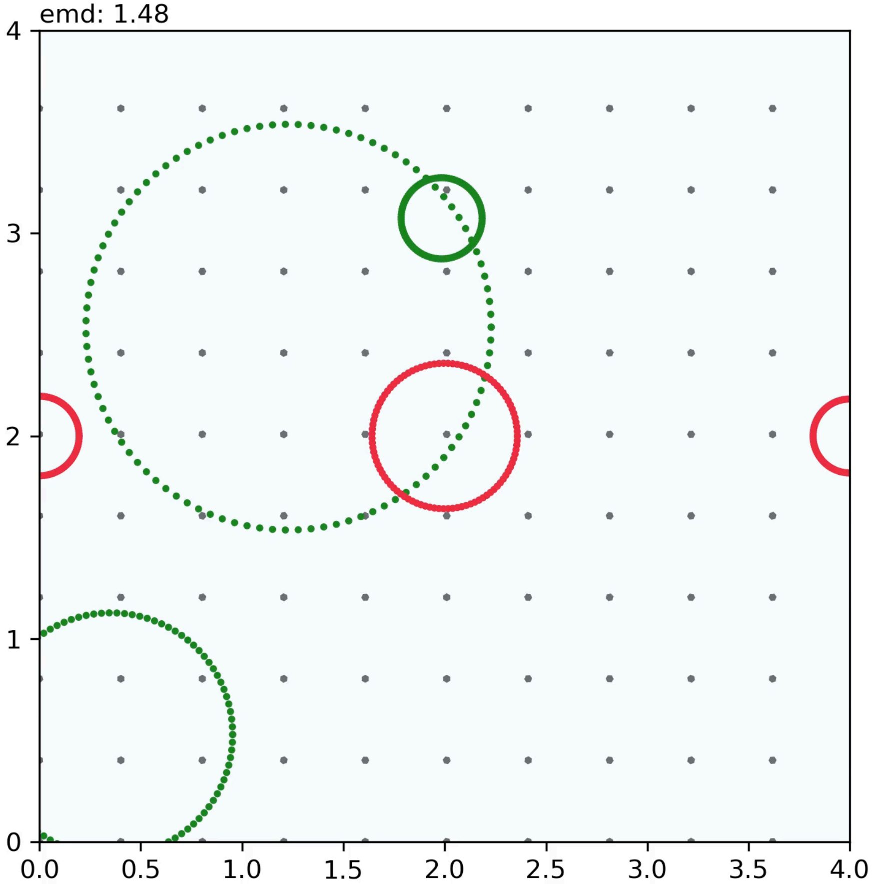


Fitting Arbitrary Geometries

Possible use cases:

- Floating term for a uniform background to mitigate pileup
- Clustering with jet energy estimation
- Variable shape clusters (learned radii, ellipses, arbitrary shapes, etc.)
- New (and old) observables: N-subjet, N-circle, triangularness, etc.

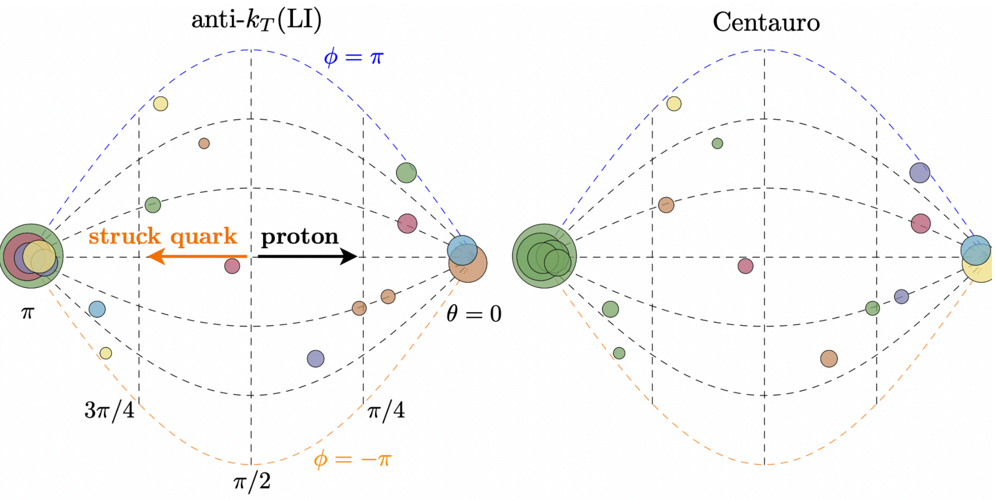
All in a unified framework given by the Energy Mover's Distance*



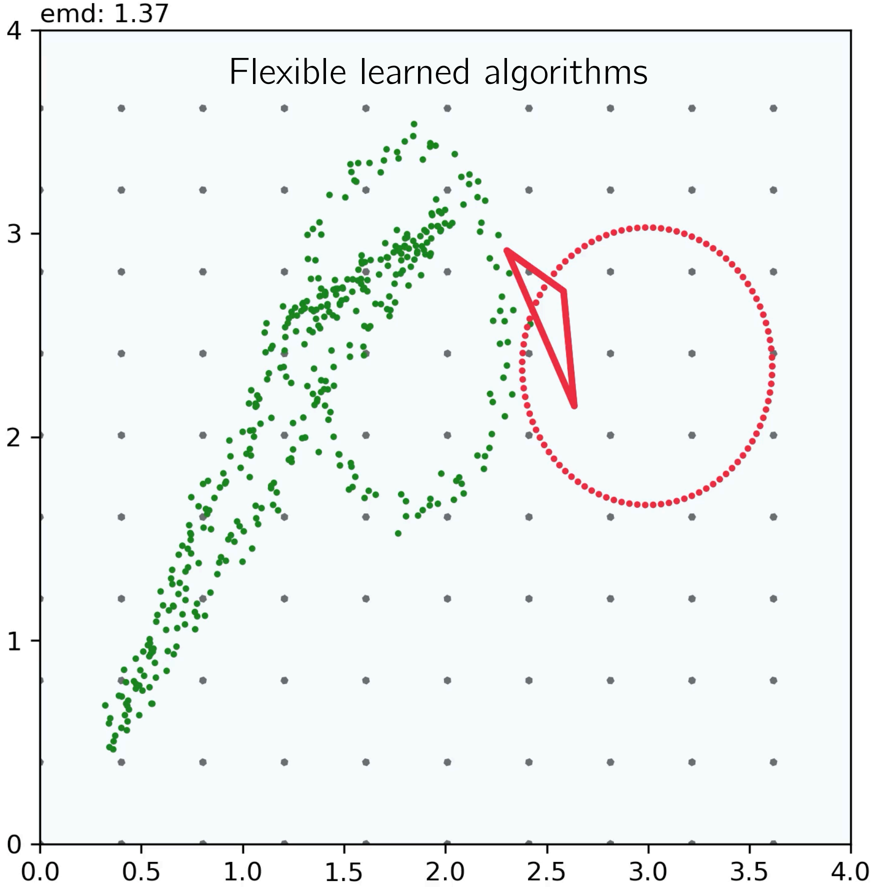
*Can You Hear the Shape of a Jet [https://indi.to/rbQ5j]

EIC - Applications

Hand-designed algorithms

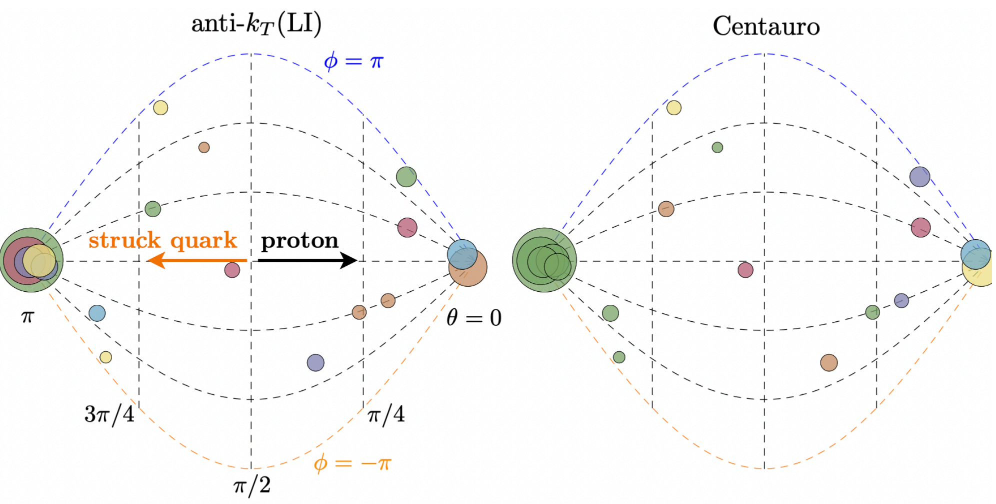


Asymmetric jet clustering in deep-inelastic scattering
[arxiv.org/pdf/2006.10751.pdf]



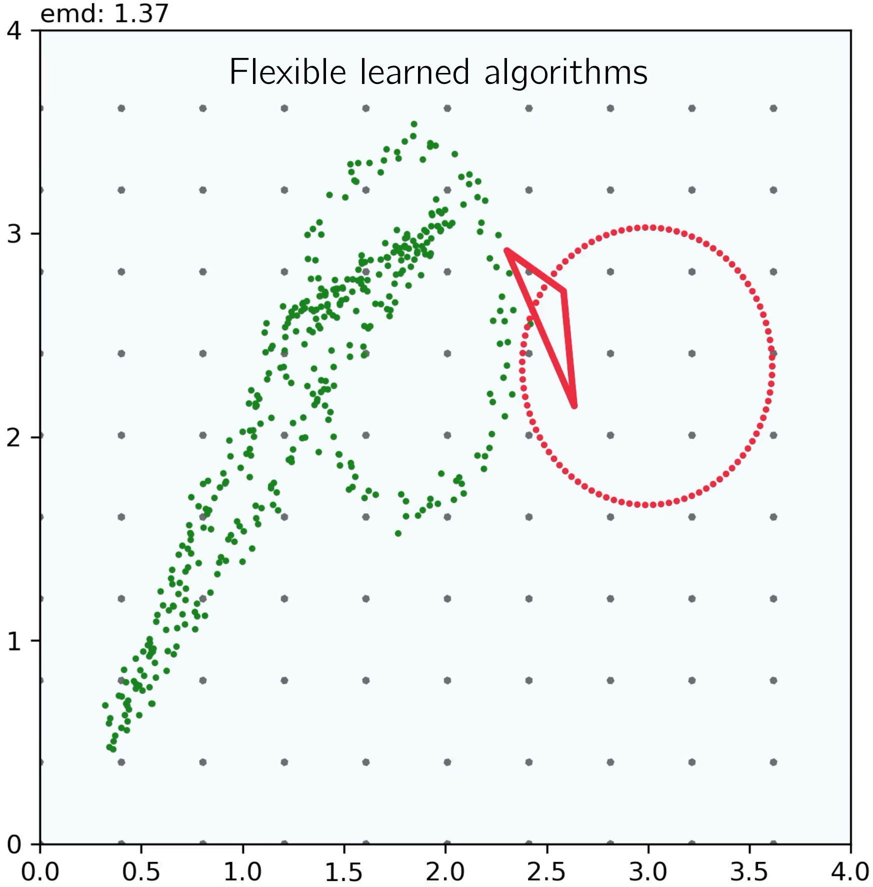
EIC - Applications

Hand-designed algorithms



Asymmetric jet clustering in deep-inelastic scattering
[arxiv.org/pdf/2006.10751.pdf]

Flexible learned algorithms



Questions

https://arxiv.org/abs/2112.00038
https://arxiv.org/abs/2209.15624

NEEMo

https://github.com/okitouni/EnergyMover-Dual/tree/neurips2022

Monotonenorm

https://github.com/niklasnolte/MonotOneNorm

pip install monotonenorm

conda install monotonenorm -c okitouni