HATANN: High-dimensional Approximation using Taylor expansions and Adaptive Nearest Neighbors

Ben Longaker, Prajwal Prathiksh, Skandan Subramanian, George Biros

Cosmic Horizons Conference, July 16, 2026

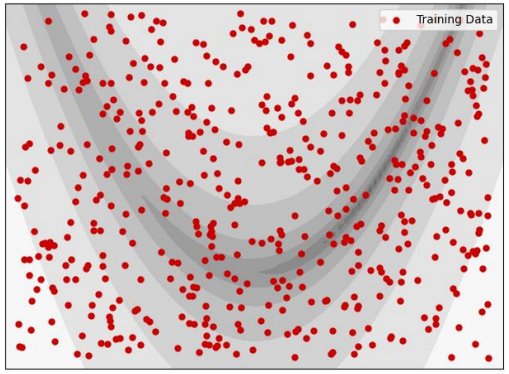
Problem Statement

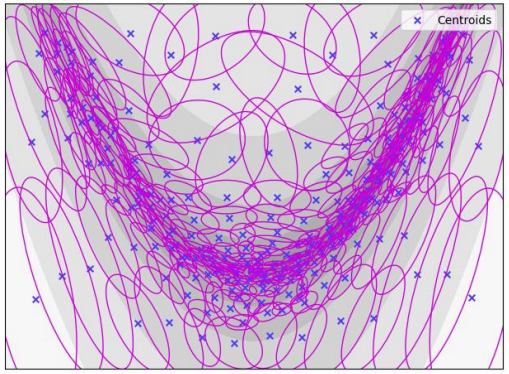
Given an **oracle** $f(x)$, some initial data, and desired tolerance ϵ, build a **surrogate** $\hat{f}$ that can **rapidly** approximate f for unknown points with pointwise error bounded by ϵ.

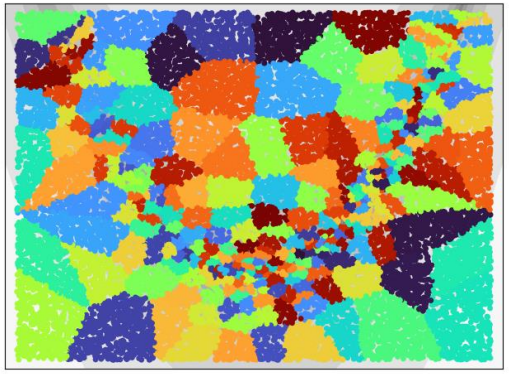
HATANN: a new surrogate technique that controls error with Taylor expansions and approximates unseen points using nearest neighbors.

Different from traditional supervised learning:

- **Access to f:** we can generate new data.
- We desire **mathematical error guarantees.**
- **Goal:** apply the algorithm to a wide range of problems.
  - CosmicAI: rapidly solve chemical networks present in MHD codes.







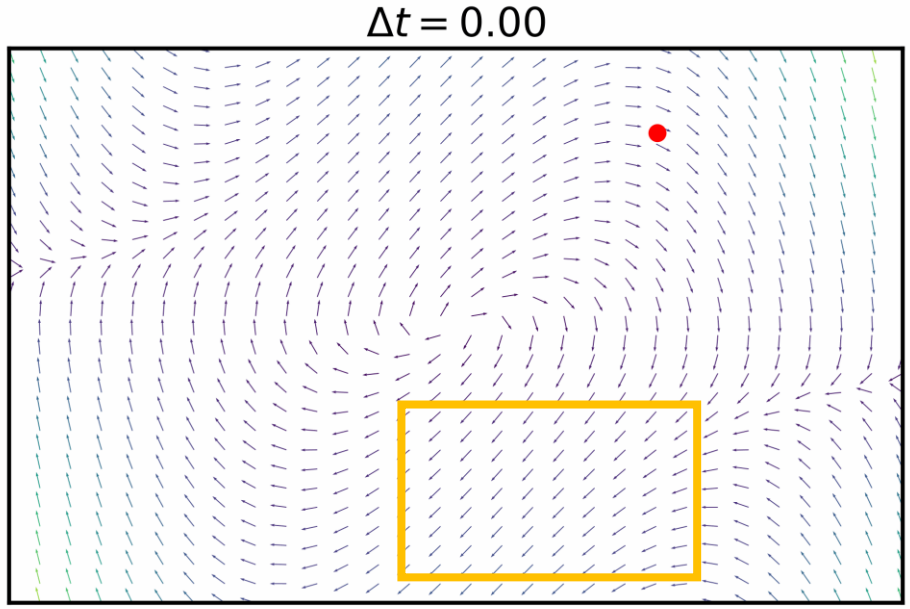
Solving ODEs: Predicting the Flow Map

- **Flow map**: Maps a state and parameters to the state at later time:

$$\Phi_{\Delta t}(y(t)) = y(t + \Delta t)$$

$$\begin{cases} \frac{dy}{dt} = g(y; p) \\ y(0) = y_0 \end{cases}$$

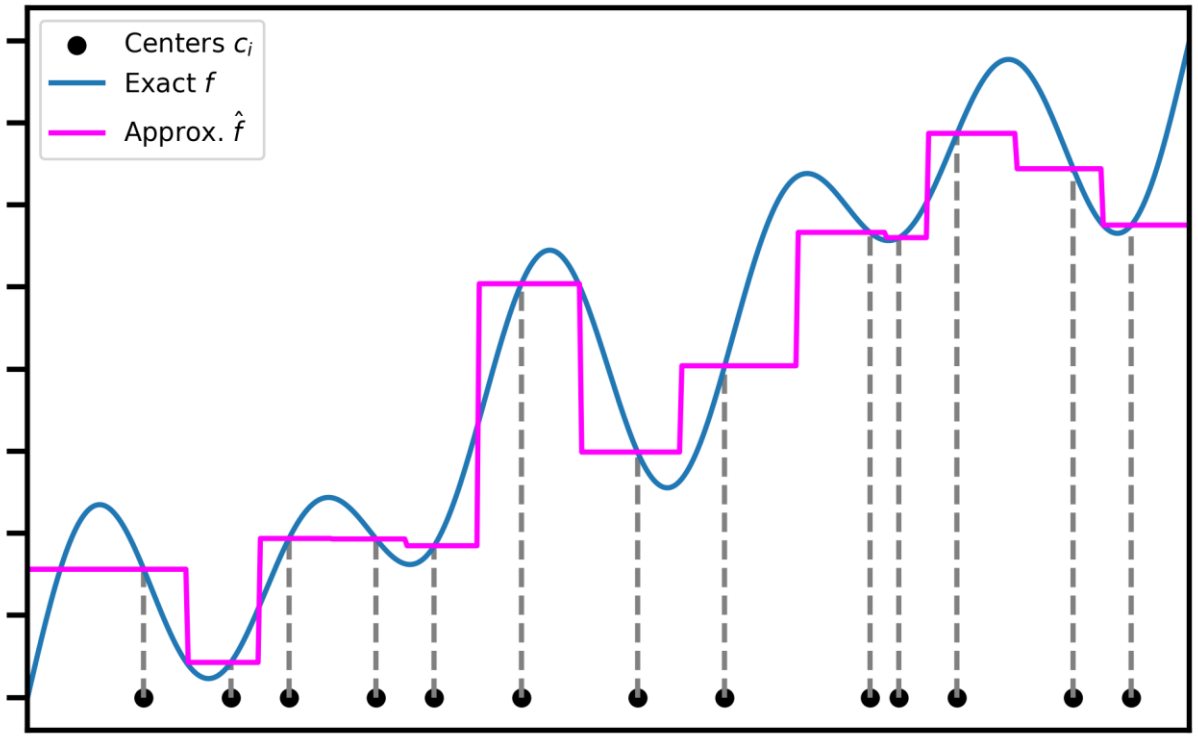
- **Idea**: Many flow maps are approximately linear in small regions of the domain.
- The shape of these regions is determined by the curvature of the flow map.



Nearest Neighbors Interpolation and Derivatives

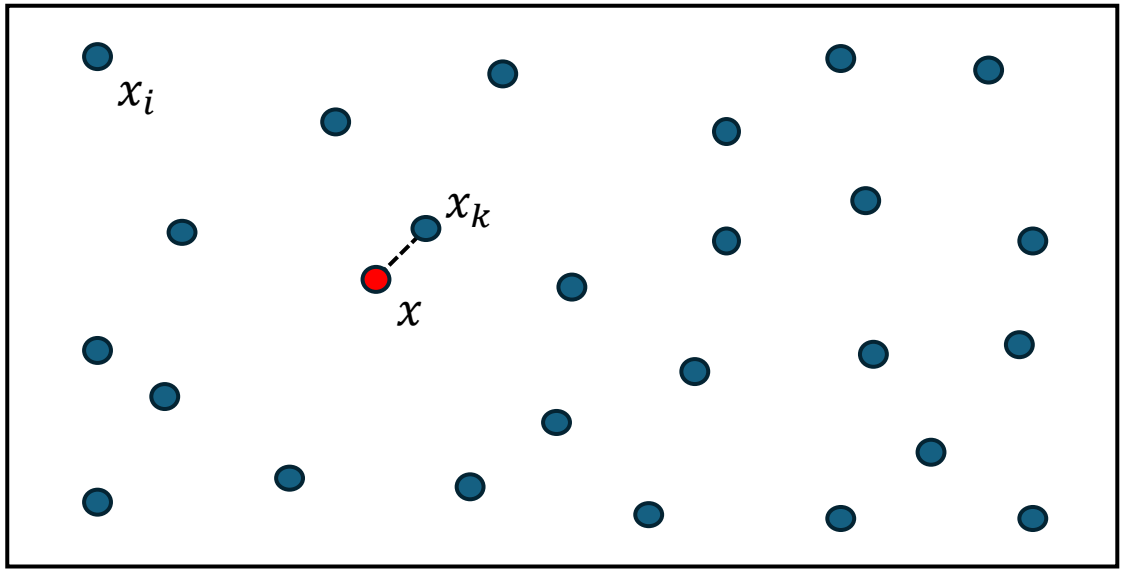
Assume we are given **centroids** and their function values $\big(x_i, f(x_i)\big)$ for $i = 1,2, \dots, n$.
Given a new x, estimate $f(x)$.

1. Find the nearest neighbor of x, indexed by k.
2. Set $\hat{f}(x) = f(x_k)$.



Taylor's Remainder Theorem: The error is proportional to the distance:

$$|f(x) - \hat{f}(x)| = O(\|x - x_k\|)$$

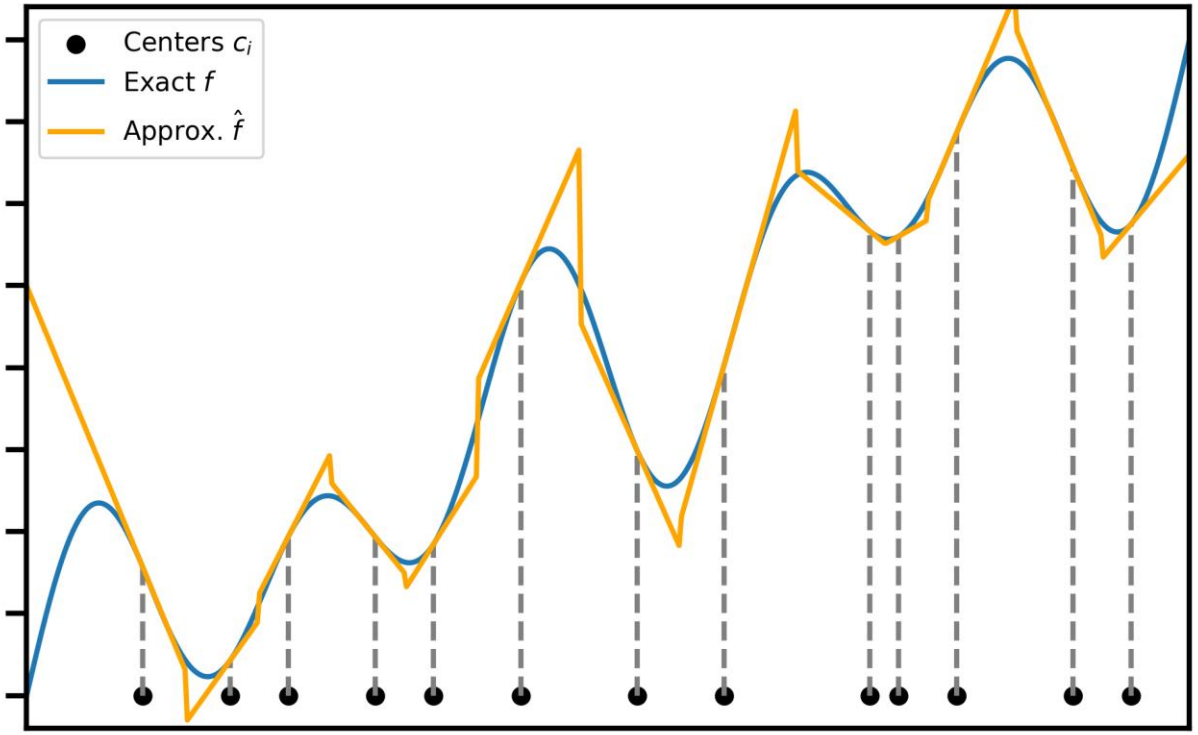


Nearest Neighbors Interpolation and Derivatives

Now assume we are given gradients: $\left(x_i, f(x_i), \nabla f(x_i)\right)$ for $i = 1,2, \dots, n$.

How can we increase the accuracy of the approximation $\hat{f}$?

1. Find the nearest neighbor of $x \rightarrow$ obtain x_k.
2. Compute first-order Taylor expansion: $\hat{f}(x) = f(x_k) + \nabla f(x_k) \cdot (x - x_k)$.



Taylor's Remainder Theorem: The error is proportional to the distance **squared**:

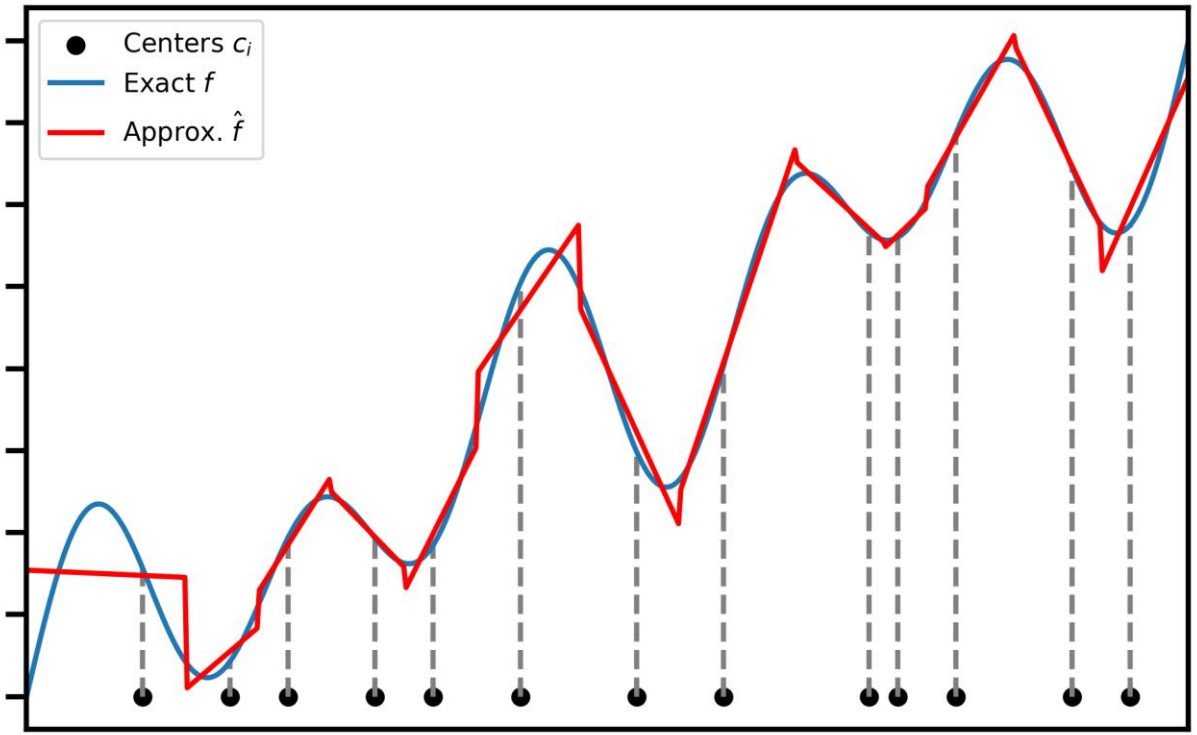
$$|f(x) - \hat{f}(x)| = O(\|x - x_k\|^2)$$

Nearest Neighbors Interpolation and Derivatives

Best of both worlds: **linear regression.**

- Does not require derivatives to compute!
- Find local linear regression model using all points inside of a cluster:

$$w^*, b^* = \underset{w \in \mathbb{R}^d, b \in \mathbb{R}}{\arg\min} \sum_{x_j \in \mathcal{C}} (f(x_j) - w^\top x_j - b)^2$$



Nearest Neighbors Interpolation and Derivatives

This implies that, inside a given cluster:

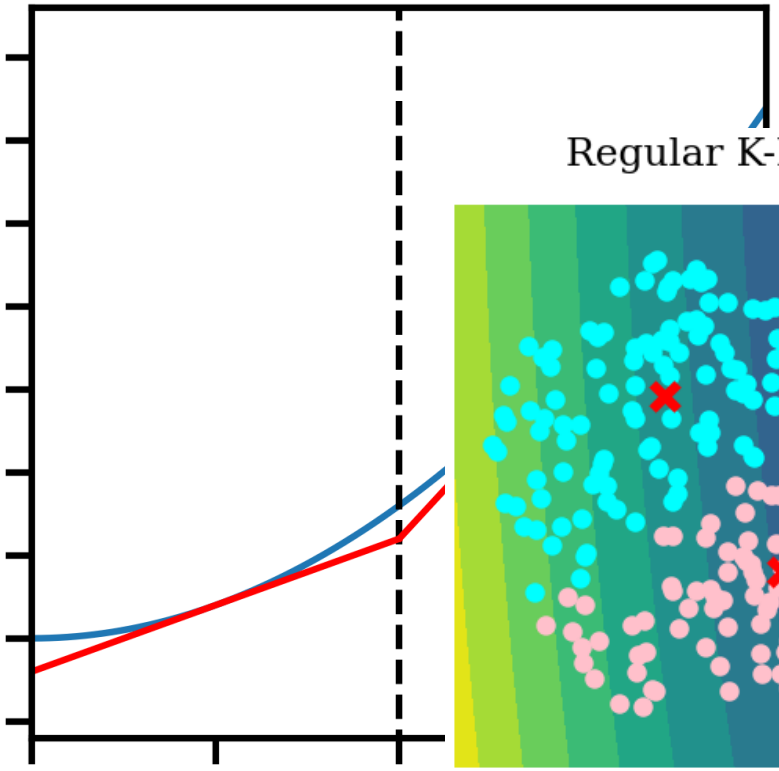
$$\begin{aligned}\int_{\mathcal{C}} |f(x) - \hat{f}_{\mathrm{LR}}(x)|^2 \, dx &\approx \sum_{x_j \in \mathcal{C}} (f(x_j) - w^\top (x_j - c) - b)^2 \\ &\leq \sum_{x_j \in \mathcal{C}} (f(x_j) - \nabla f(c)^\top (x_j - c) - f(c))^2 \\ &\approx \int_{\mathcal{C}} |f(x) - \hat{f}_{\mathrm{Taylor}}(x)|^2 \, dx.\end{aligned}$$

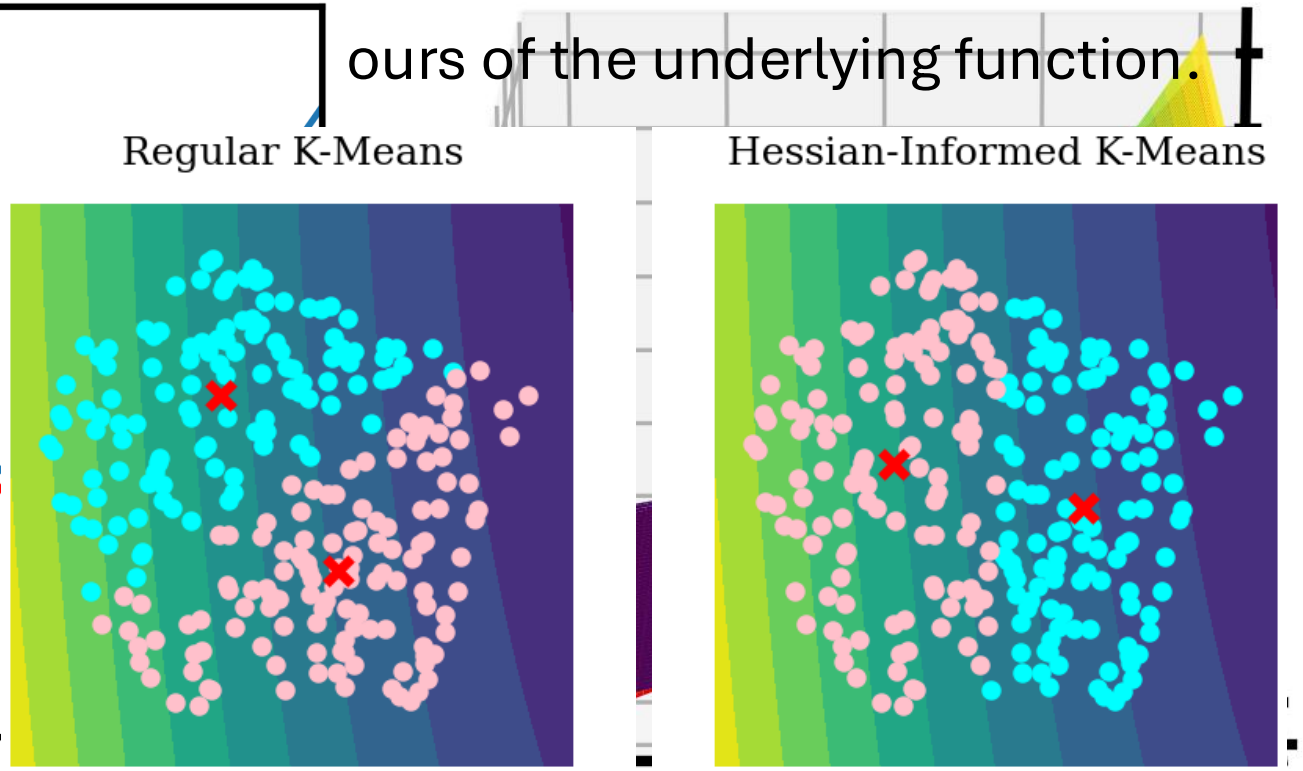
Summing over all clusters, the L^2-error is smaller for linear regression.

Strategy: use clustering and local linear regression models.

Subdividing a Cluster: Hessian-informed Clustering

Theorem (informal): If $f: \mathbb{R}^d \to \mathbb{R}$ is approximately quadratic, and $\hat{f}(x; c_k)$ is a local linear regression model using the data in $\mathcal{C}_k$, then the reclustering that minimizes the sum of linear surrogate errors over the data is **2-means** applied to the data $|H|^{1/2}\hat{X}$, where $|H|$ is the **absolute Hessian** of f at the centroid c_k.

- 
ours of the underlying function.



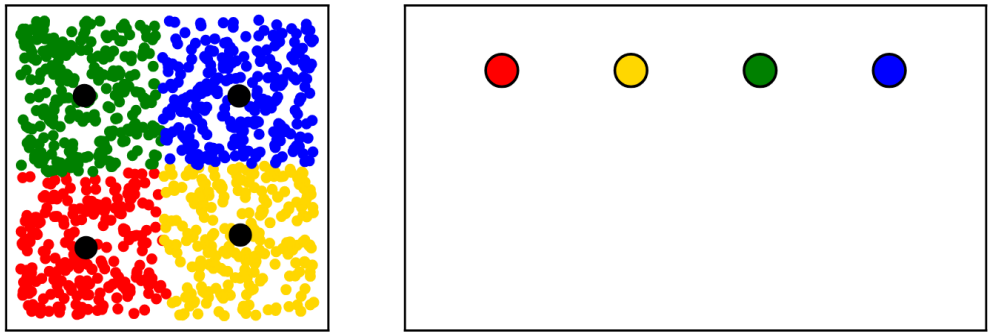
This allows us to scale into high dimensions!

Putting it together: Hierarchical Clustering

Initialization: Initial data $\left(x_i, f(x_i)\right)$.

Step 1: Cluster initial points via k-means.

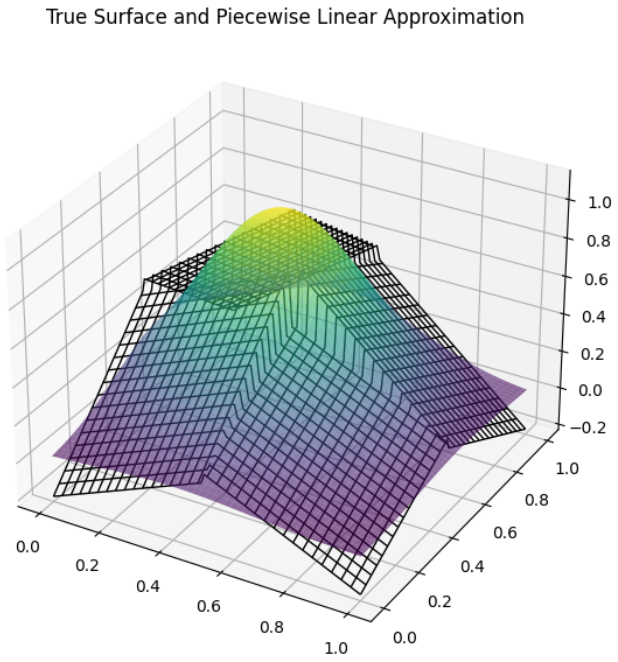
Step 2: Fit local models (linear regression). Measure the error inside each cluster.

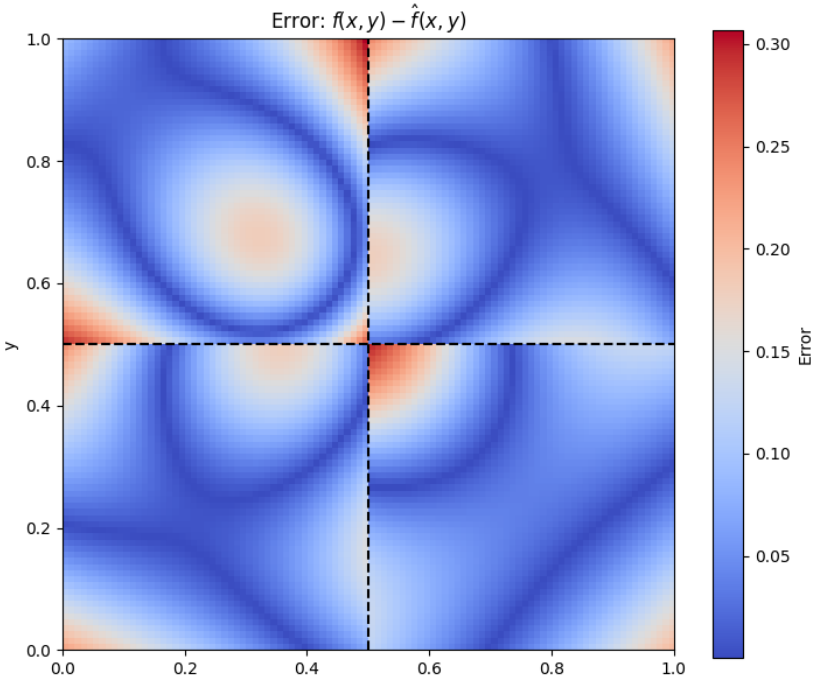


Step 3: If the err
mark the cluste
cluster, comput

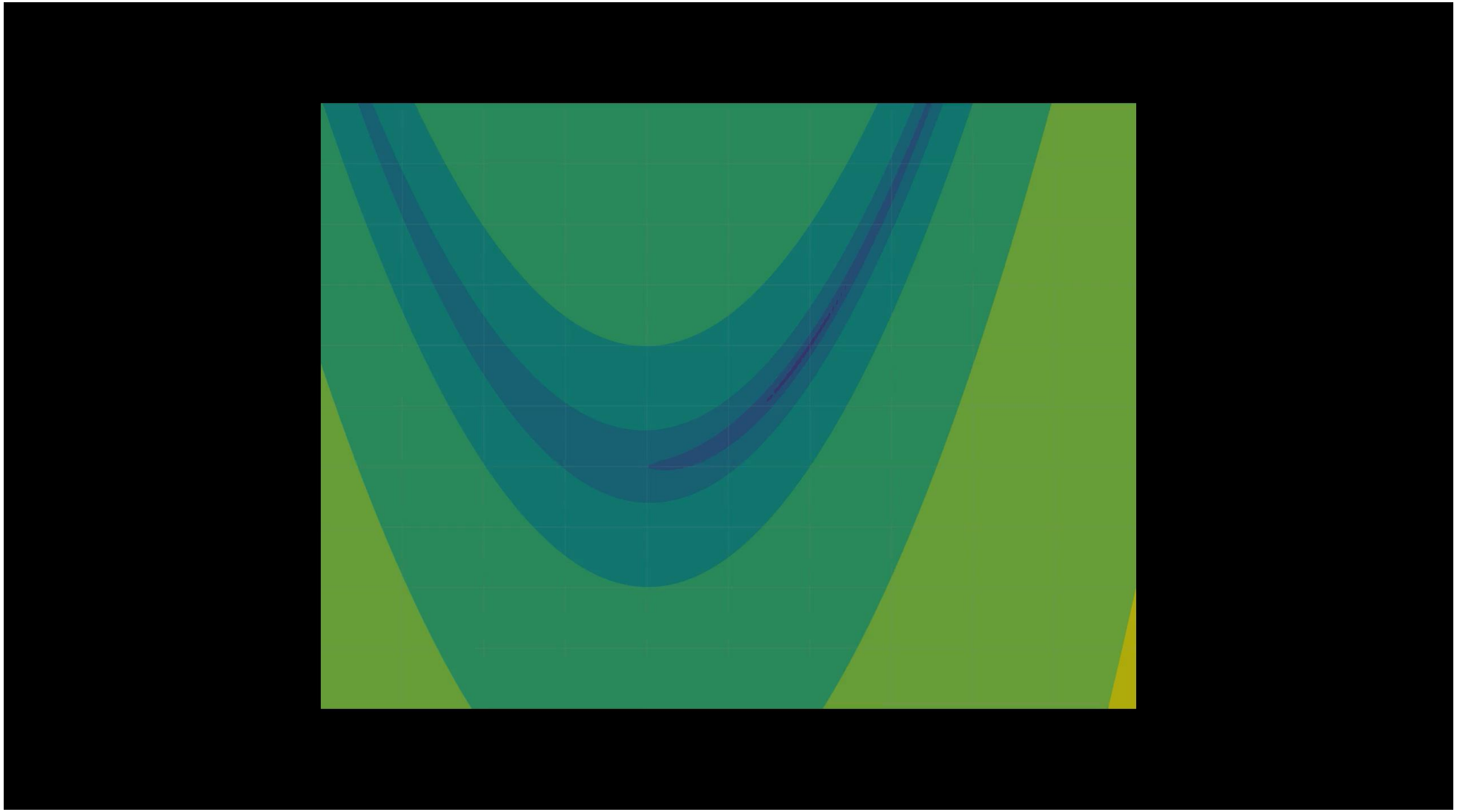
Step 4: If the err
further. Use **Hes**
considers the cu

Step 5: Repeat
cluster is marke
of binary trees.
and trust region





Example: Rosenbrock Function

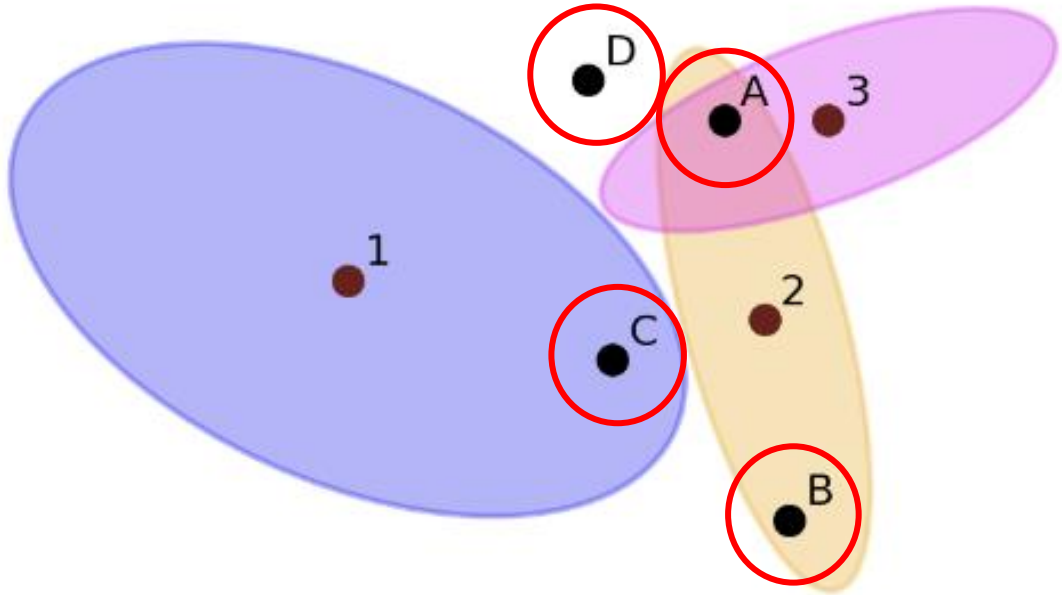


Inference: Adaptive Nearest Neighbors

- After surrogate construction, we have
  - Centroids $\{c_j\}_{j=1}^{N}$
  - Local linear models $\{w_j, b_j\}_{j=1}^{N}$
  - **Trust regions** $\{Q_j\}_{j=1}^{N}$: obtained by fitting ellipses around final clusters.

How to pick the right cluster?

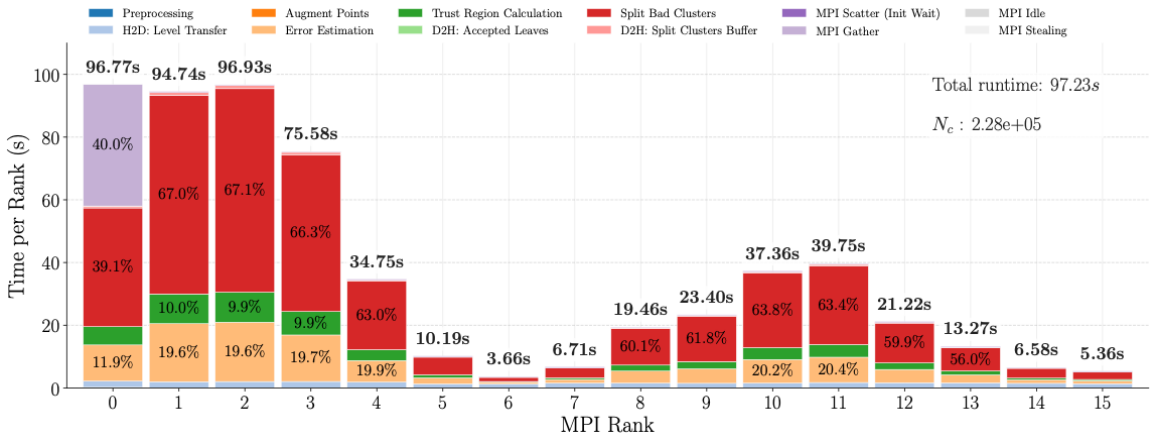
Use a metric-adaptive criterion. First look for a small handful of candidates.

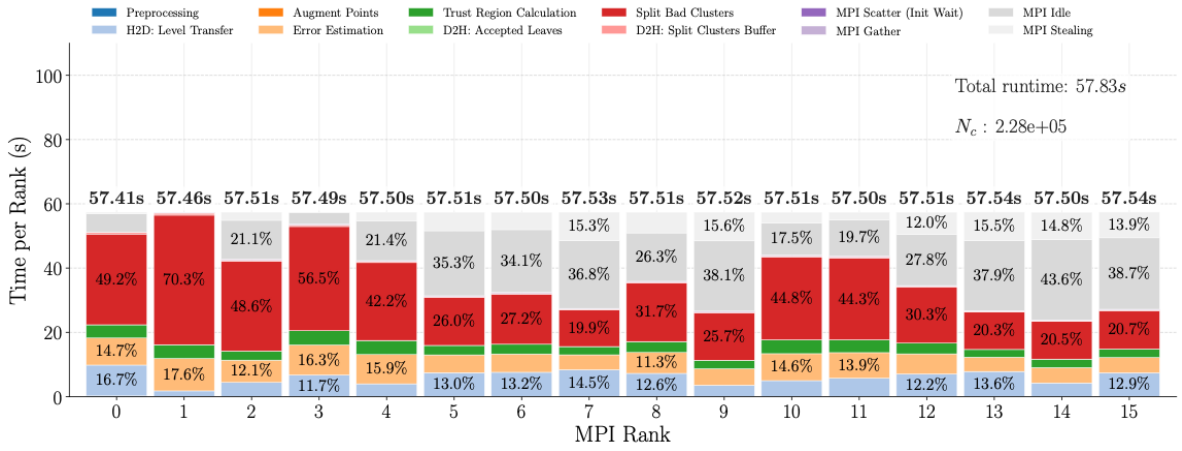


A: Overlapping regions.
B: Trust region anisotropy.
C: Specific distance metric.
D: No trust region.

GPU Algorithm: Work Stealing

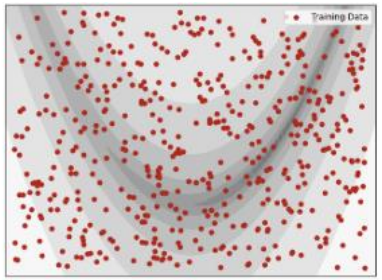
- After initial clustering, each cluster can be treated independently. Need an algorithm to do load balancing.
- Some processes may finish before others if they work on parts of the domain that are less complicated.
- Solution: **work stealing.** If a process finishes, enter **thief** state: make attempts to steal half of work from the busiest processor.



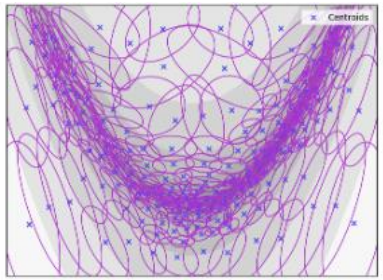


Results: Rosenbrock Function

r_{tol}	N	N_c	p_o	e_{in}	e	e_{nh}	e_{ng}
Rosenbrock ($d = 2, N_0 = 8.45\text{E}4$)							
0.1	1.17E5	361	0.8%	3.38E-2	3.39E-2	6.17E-2	3.20E-1
0.05	1.74E5	776	1.0%	1.25E-2	1.25E-2	5.87E-2	3.08E-1
0.02	3.57E5	1,962	1.5%	3.89E-3	3.92E-3	3.58E-2	2.19E-1
0.01	7.05E5	4,063	1.9%	2.58E-3	2.60E-3	1.46E-2	1.38E-1
0.005	1.38E6	8,051	3.3%	1.47E-3	1.49E-3	7.57E-3	1.02E-1
Rosenbrock ($d = 5, N_0 = 1.86\text{E}5$)							
0.1	1.12E7	30,110	18.1%	4.79E-2	5.72E-2	1.02E-1	2.94E-1
0.05	3.74E7	100,396	37.9%	2.71E-2	4.17E-2	6.95E-2	2.46E-1
0.02	1.50E8	401,212	65.4%	1.37E-2	3.22E-2	4.79E-2	2.10E-1
0.01	3.97E8	1,064,029	80.4%	8.14E-3	2.77E-2	3.89E-2	1.92E-1
0.005	1.03E9	2,749,440	89.8%	4.53E-3	2.45E-2	3.26E-2	1.76E-1

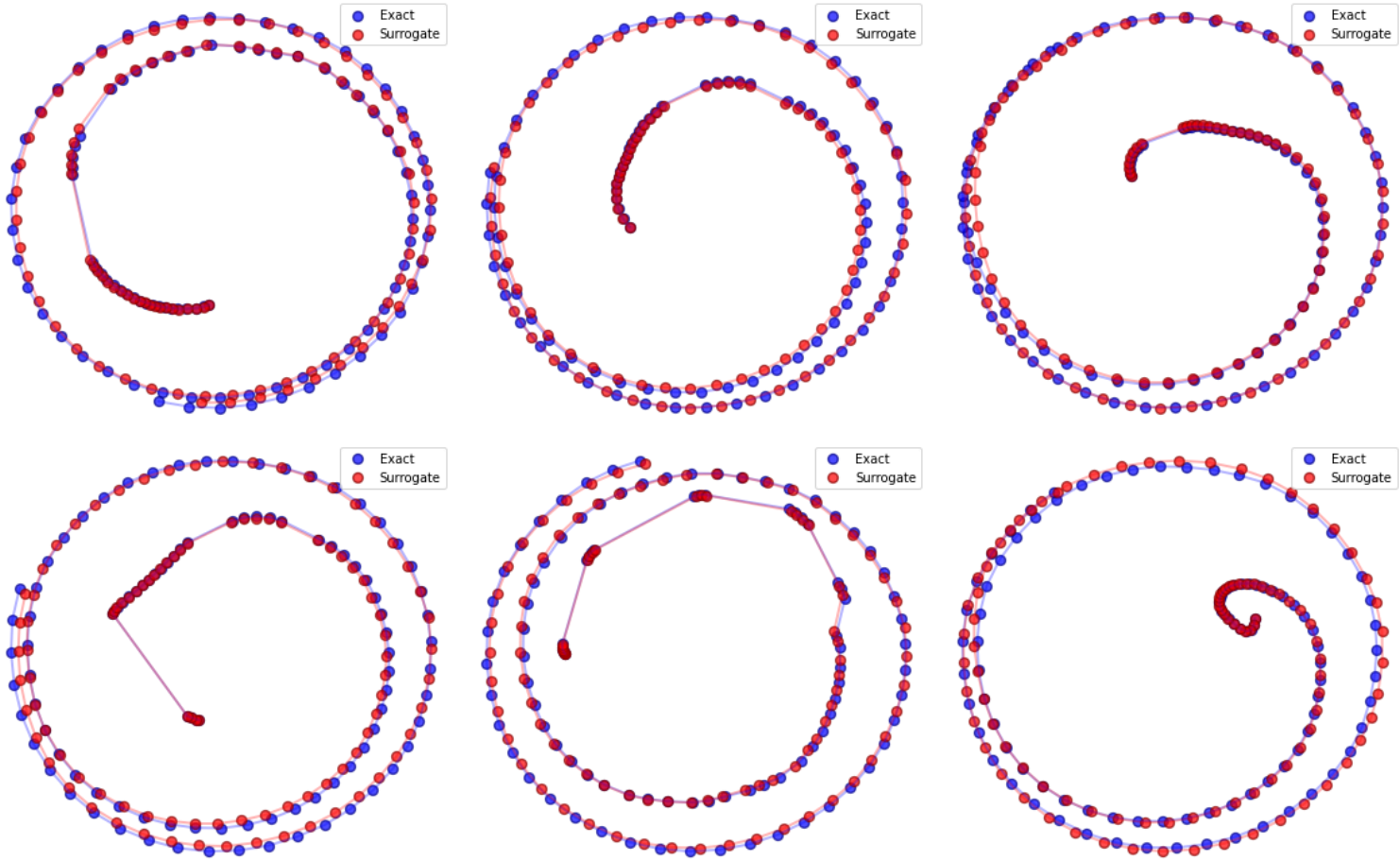


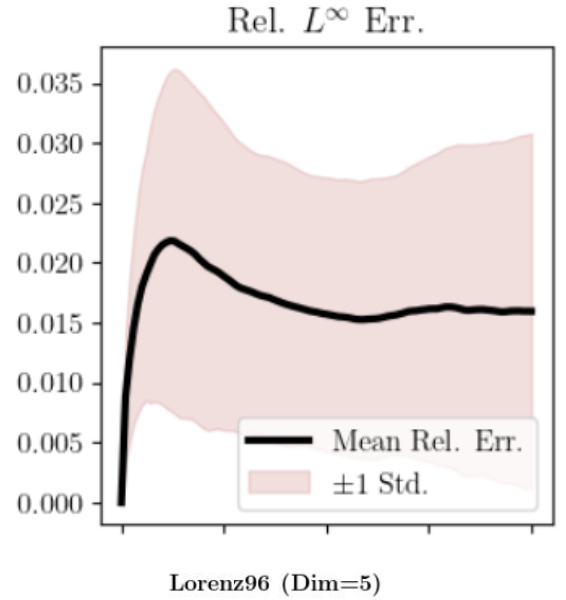
Initial Seed (Training) Data



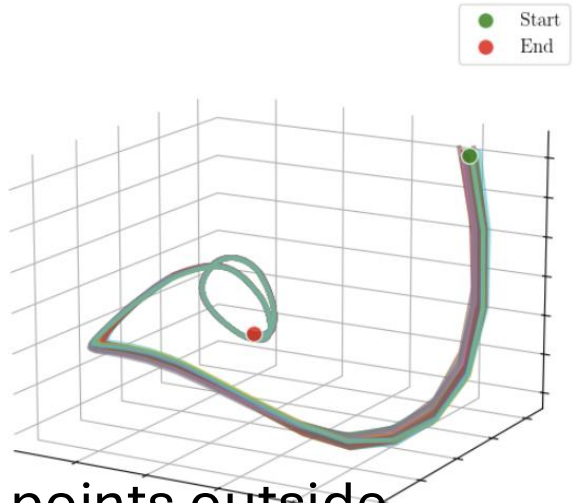
Final Clusters

Results: Lorenz96 Model, uniform sampling





Lorenz96 (Dim=5)



- **80 dimensions:** 7.93% L2 error in trust region, with 34.7% of test points outside.

Thank you!

blongaker@utexas.edu

Prajwal Prathiksh

Skandan Subramanian

George Biros

Appendix