Machine Learning and the 'Mogi' model

Improving the efficiency of ensemble-based methods for volcano deformation analyses

Matthew Head (mshead@illinois.edu)

SZ4D ML/AI Virtual Workshop
2nd August 2023

MODEL-DATA FRAMEWORK

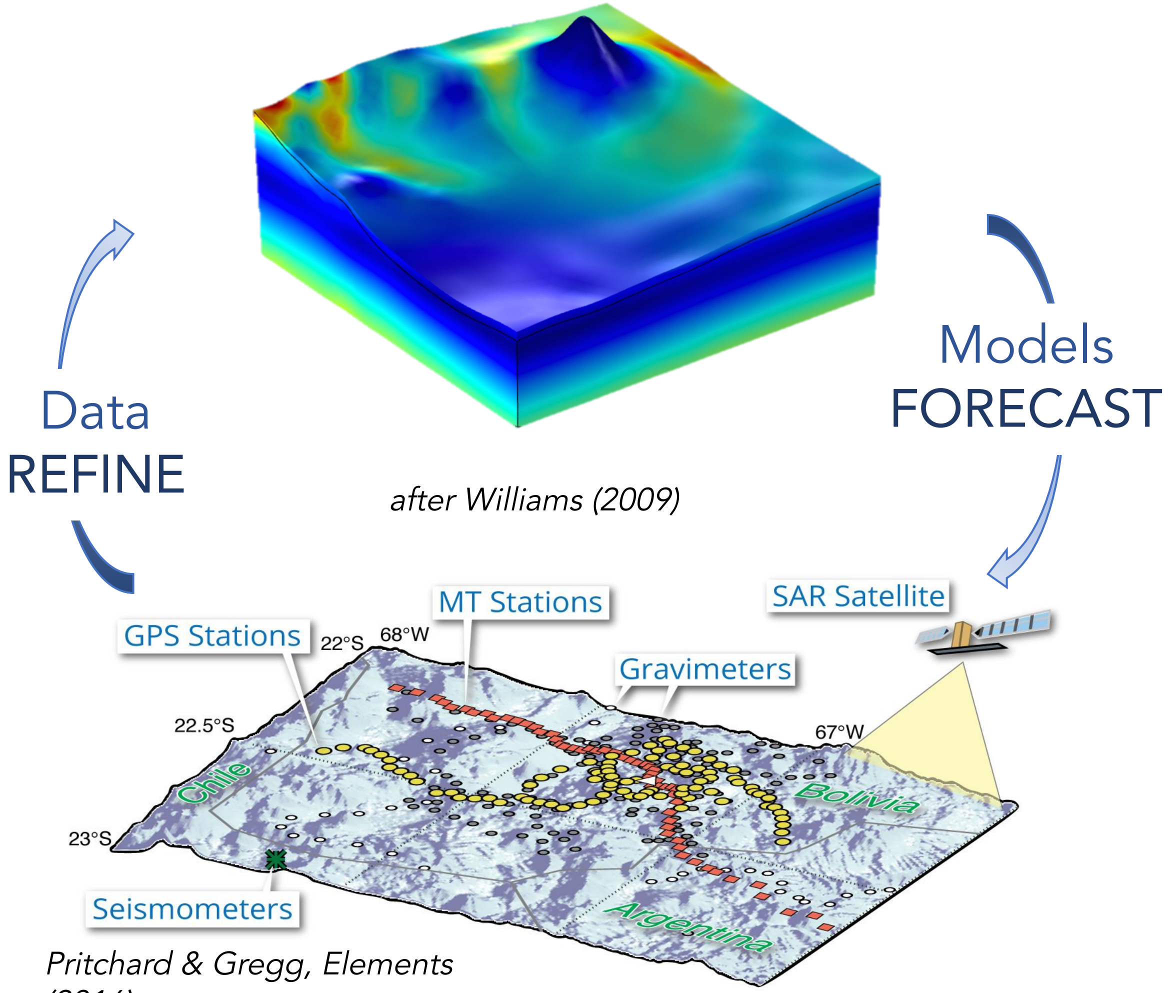
1) Ensemble Kalman Filter (EnKF)

Adapted for analyses of volcanic ground deformation

2) Finite Element Method (FEM)

Physics-based modelling

- Sequentially assimilate and invert geodetic observations
- Constrain the nature of deforming magmatic systems

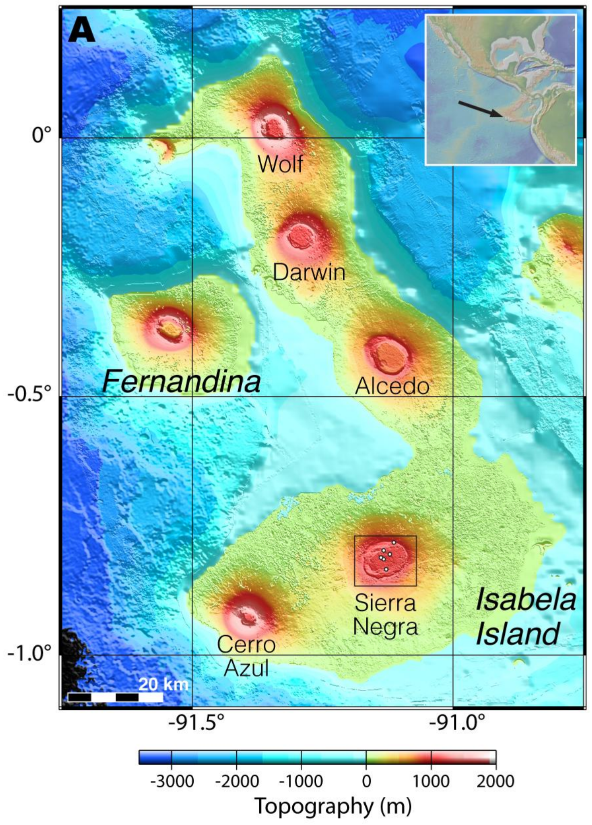


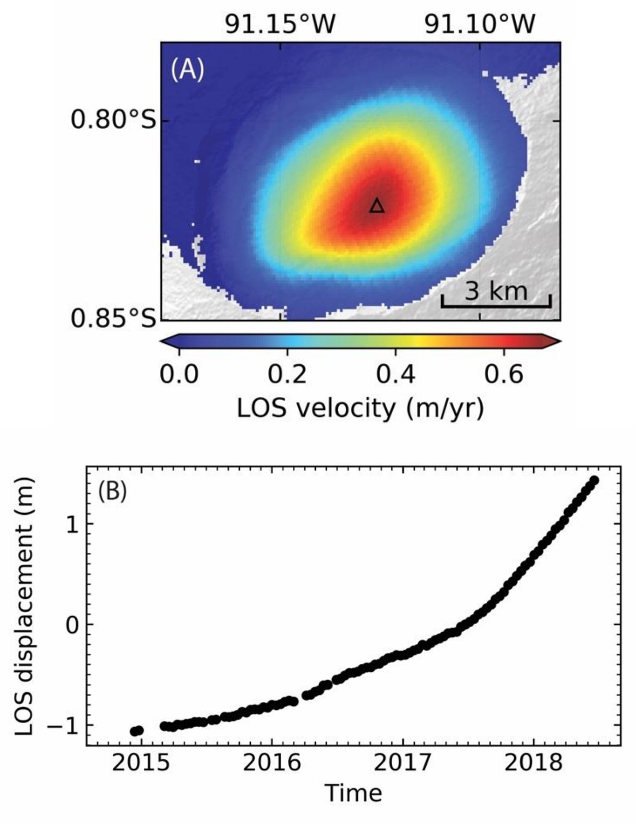
after Williams (2009)

Pritchard & Gregg, Elements (2016)

SIERRA NEGRA, GALAPAGOS

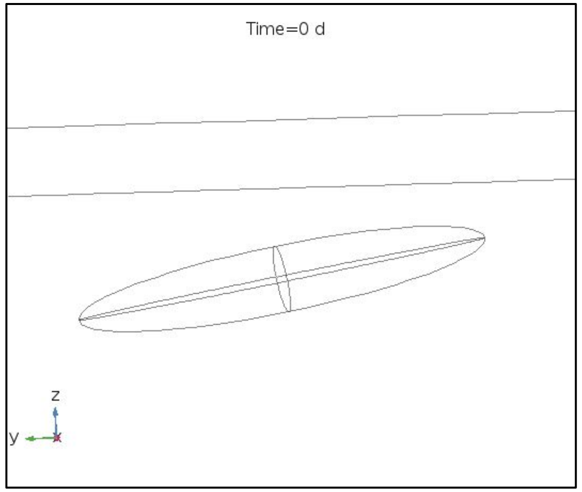
- Inversion of InSAR observations (Gregg et al., 2022)
  - March 2015 – January 2018
  - 3D model geometry
  - Refine 8 parameters
- Assess mechanical stability
  - Assumed pressure evolution
  - 10-day rolling failure forecast
  - 25th June – 5th July 2018
- Eruption began 26th June 2018

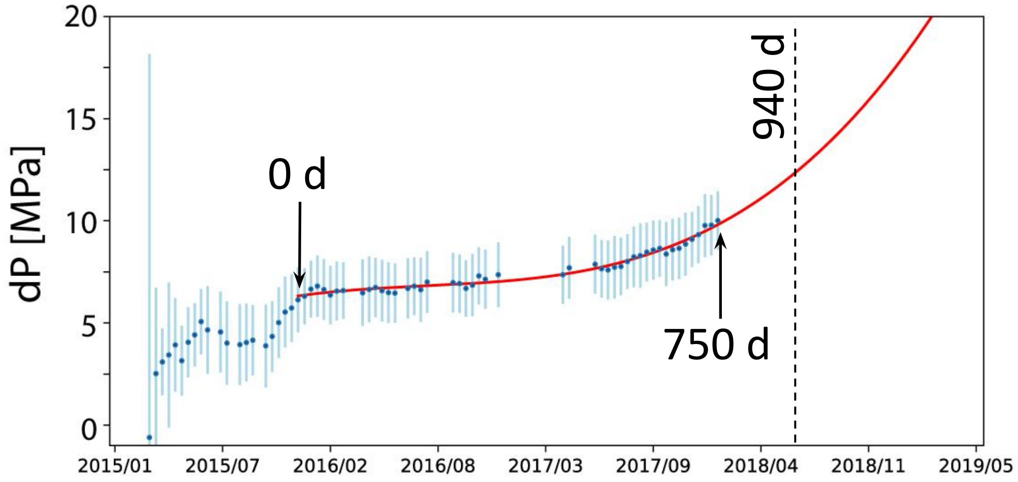




SIERRA NEGRA, GALAPAGOS

- Inversion of InSAR observations (Gregg et al., 2022)
  - March 2015 – January 2018
  - 3D model geometry
  - Refine 8 parameters
- Assess mechanical stability
  - Assumed pressure evolution
  - 10-day rolling failure forecast
  - 25th June – 5th July 2018
- Eruption began 26th June 2018



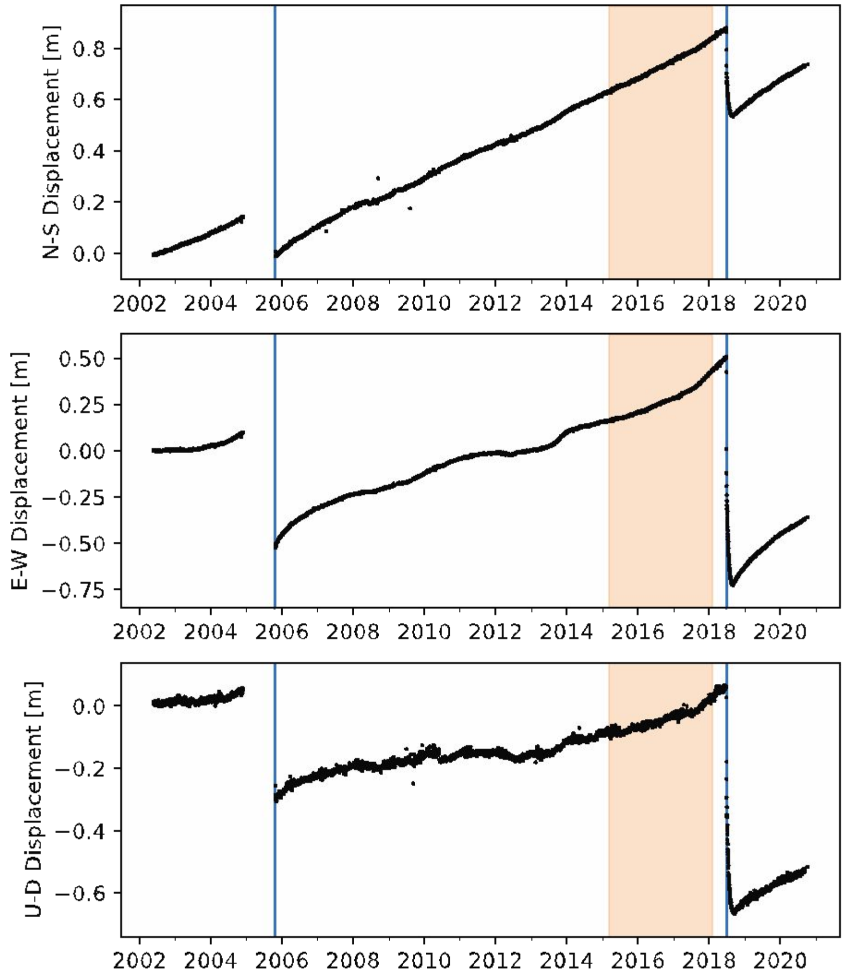


LIMITATIONS

- Computational load
  - 240 ensemble members
  - 67 observations, 6 iterations
  - 1680 core hours (~1min per FEM)
- GPS observations
  - Temporally dense
  - Non-feasible with current approach
- Determine ‘true’ stress state after the 2005 eruption
  - Need entire timeseries
  - Temporal downsampling, or… ?

LIMITATIONS

- Computational load
  - 240 ensemble members
  - 67 observations, 6 iterations
  - 1680 core hours (~1 min per FEM)
- GPS observations
  - Temporally dense
  - Non-feasible with current approach
- Determine ‘true’ stress state after the 2005 eruption
  - Need entire timeseries
  - Temporal downsampling, or… ?



OBJECTIVES

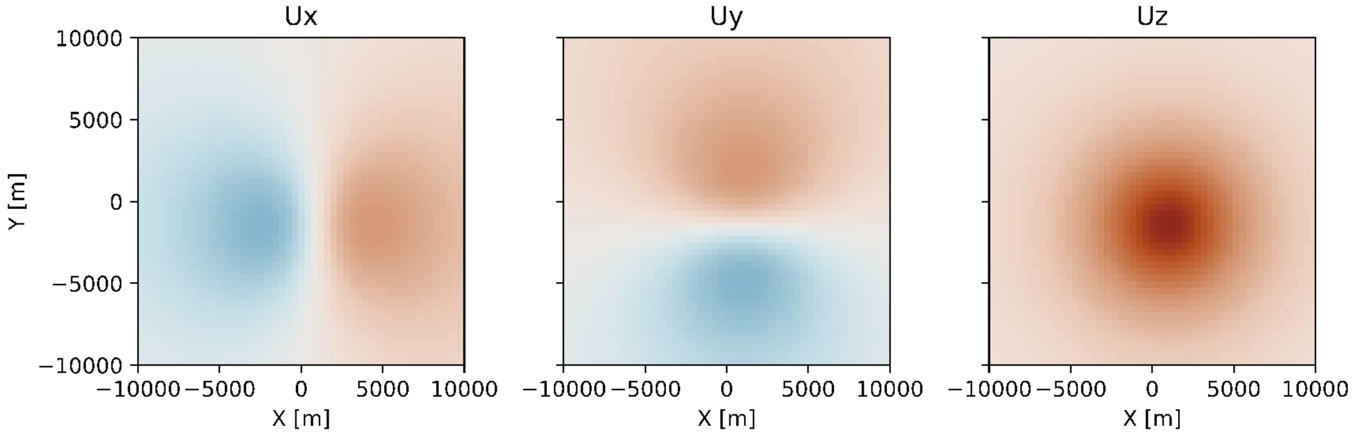
- Reduce the number of FEMs evaluated for the EnKF analysis
- Predict deformation field for a given model state
  - Regression algorithm
  - Must be accurate and precise
- Implemented within python
- Start with a simple approach
  - *k*-nearest neighbors

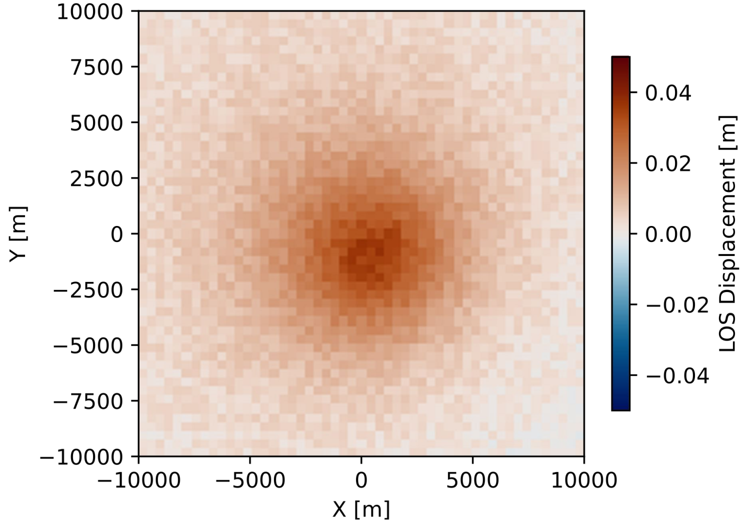
SYNTHETIC TEST

- 'Mogi' point source (Mogi, 1957)
  - Analytical model
  - 5 free parameters

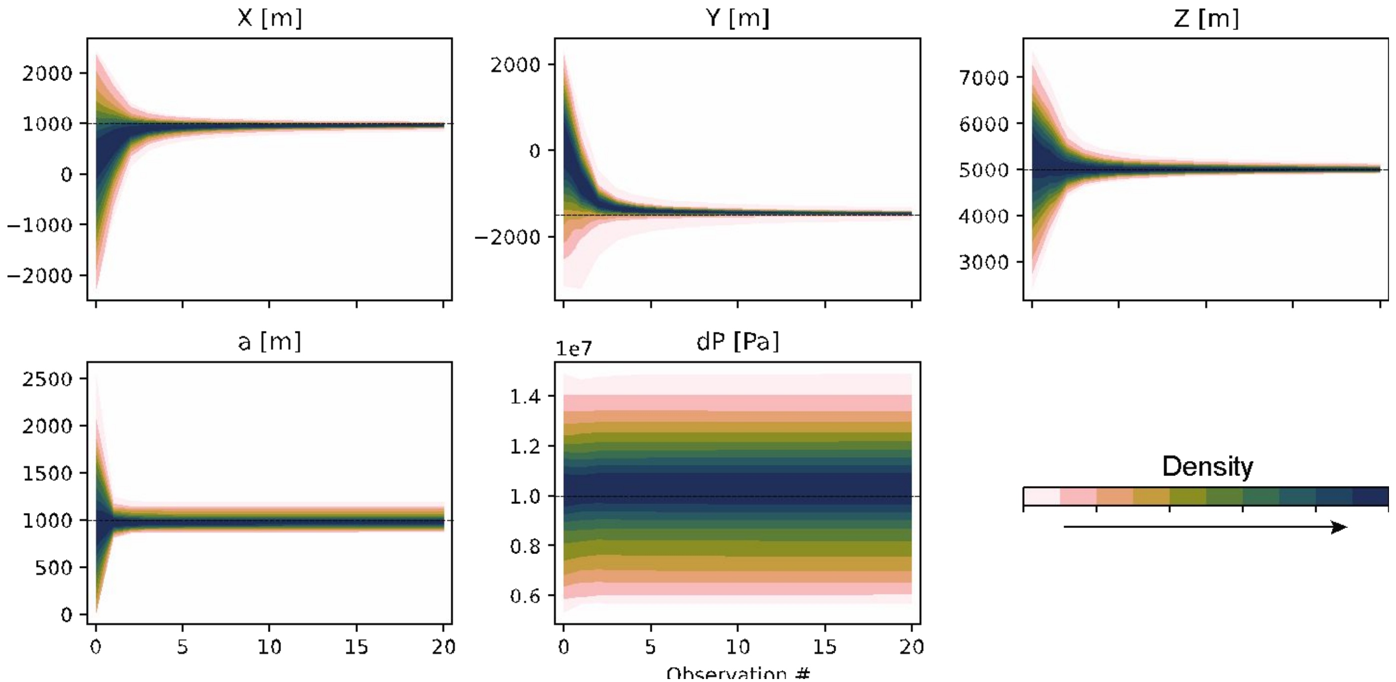
$$\begin{pmatrix} Ux \\ Uy \\ Uz \end{pmatrix} = \frac{(1-\upsilon)}{\mu} a^3 \Delta P \begin{pmatrix} x/R^3 \\ y/R^3 \\ z/R^3 \end{pmatrix}$$

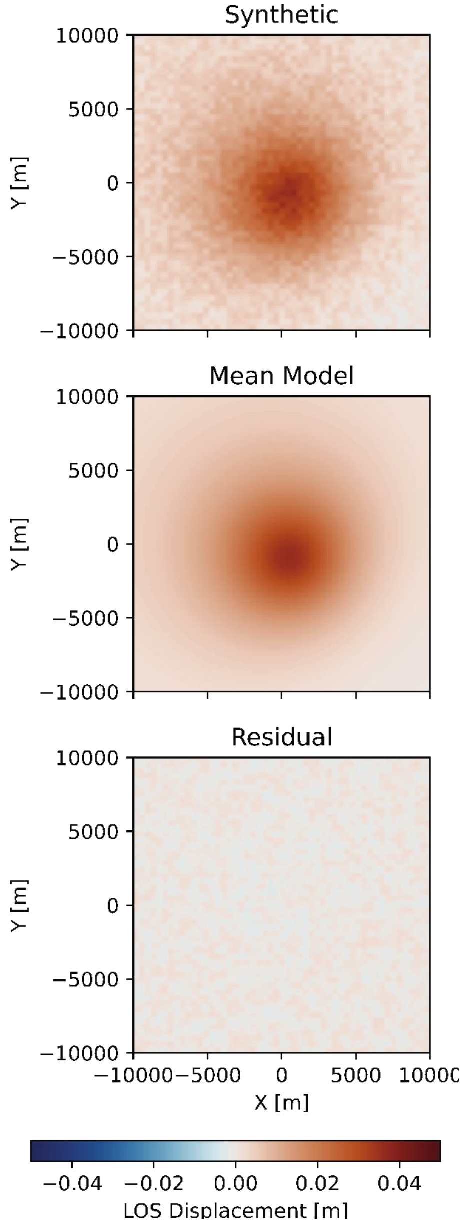
- Output 51 x 51 grid (2601 pixels)
- Line-of-sight displacement (InSAR)
  - Add noise





ENKF ANALYSIS





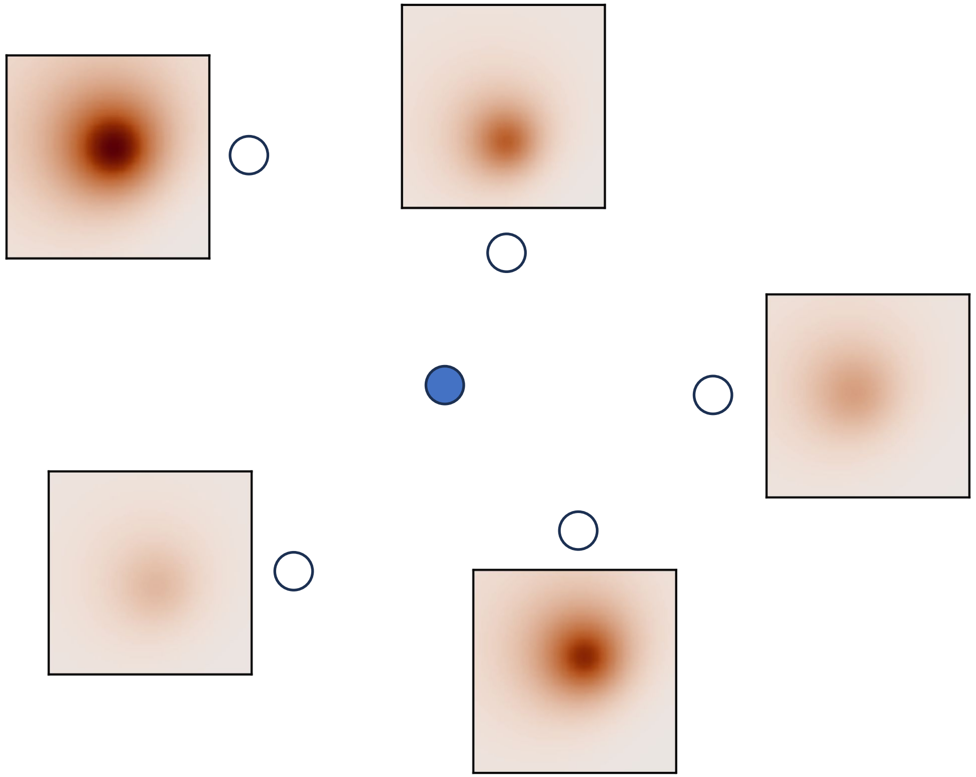
>>> from sklearn.neighbors

import KNeighborsRegressor

- Prediction by local interpolation of *k*-nearest neighbors
- Only one hyperparameter, *k*
- 'uniform' or 'distance' weighting
- Different distance metrics

- Doesn't perform well for datasets with large number of features
  - Here we only have 5

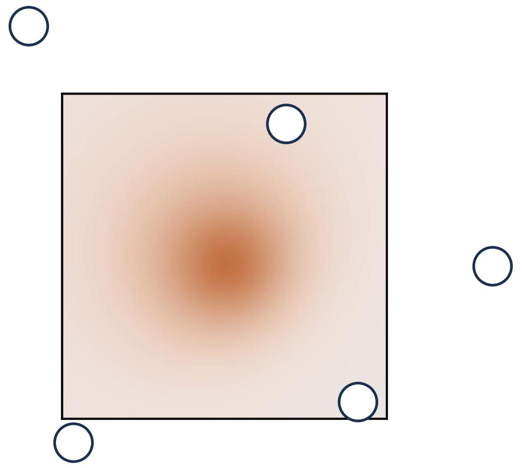
- Performs particularly badly with sparse datasets



>>> from sklearn.neighbors

import KNeighborsRegressor

- Prediction by local interpolation of *k*-nearest neighbors
- Only one hyperparameter, *k*
- ‘uniform’ or ‘distance’ weighting
- Different distance metrics



- Doesn’t perform well for datasets with large number of features
  - Here we only have 5

- Performs particularly badly with sparse datasets

>>> from sklearn.preprocessing

ML algorithms typically require features to be of similar magnitude

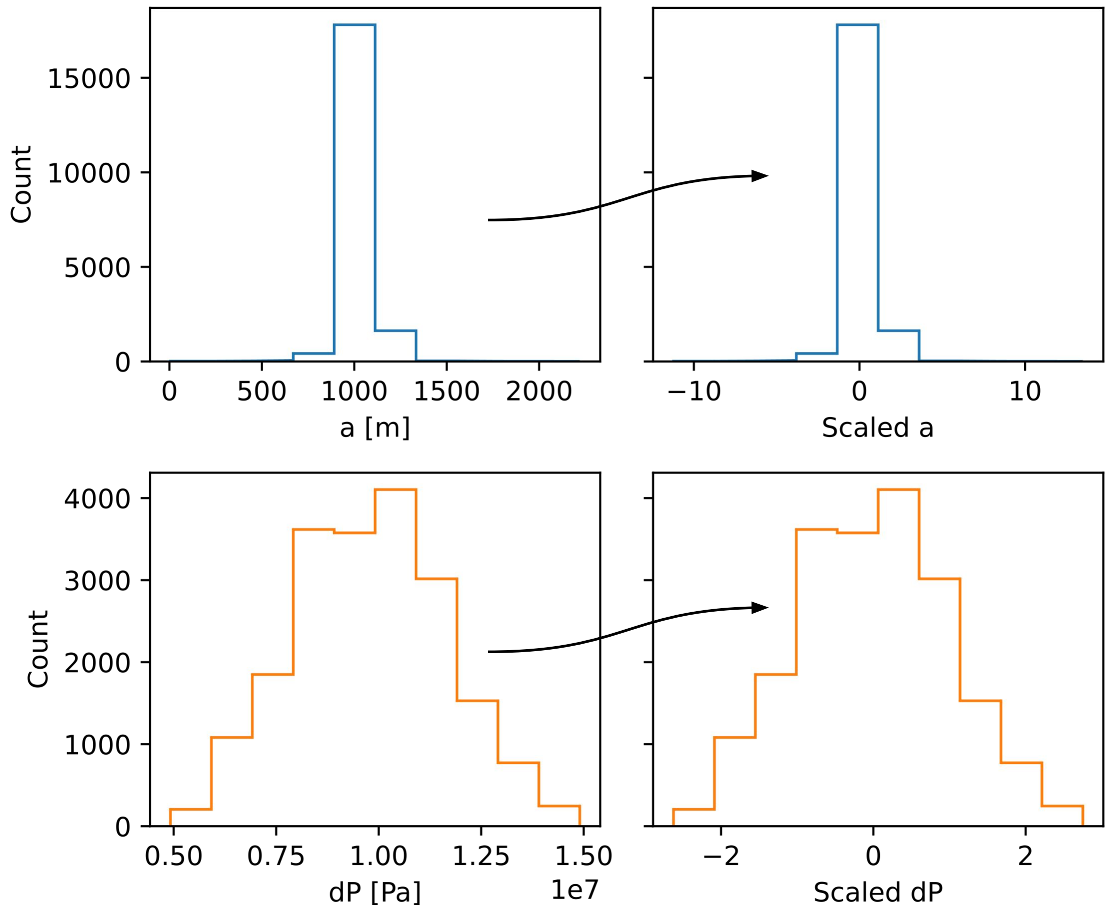
import StandardScaler

- Remove mean
- Scale to unit variance

Outliers can negatively influence the sample mean and variance

import RobustScaler

- Remove median
- Scale based on interquartile range



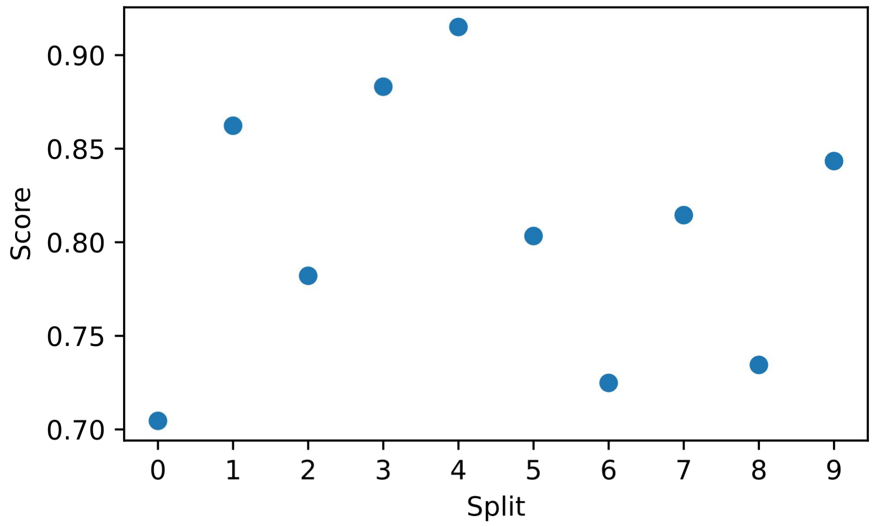
>>> from sklearn.model_selection

import train_test_split

- Randomly splits input data into training and testing subsets
- Calculate score for 10 splits
- Not suitable evaluation for use case

import cross_val_score

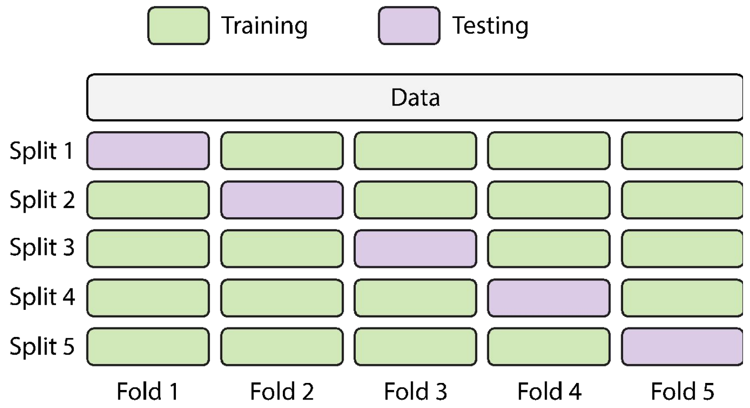
- Later folds achieve score >0.95
- Capable of predicting output as EnKF refines parameters
- Later model states are a poor predictor of early model states



>>> from sklearn.model_selection

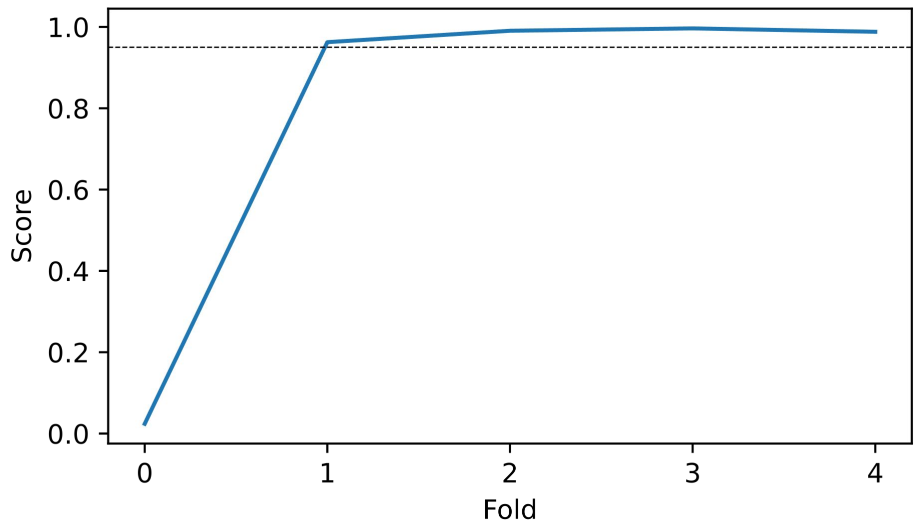
import train_test_split

- Randomly splits input data into training and testing subsets
- Calculate score for 10 splits
- Not suitable evaluation for use case



import cross_val_score

- Later folds achieve score >0.95
- Capable of predicting output as EnKF refines parameters
- Later model states are a poor predictor of early model states

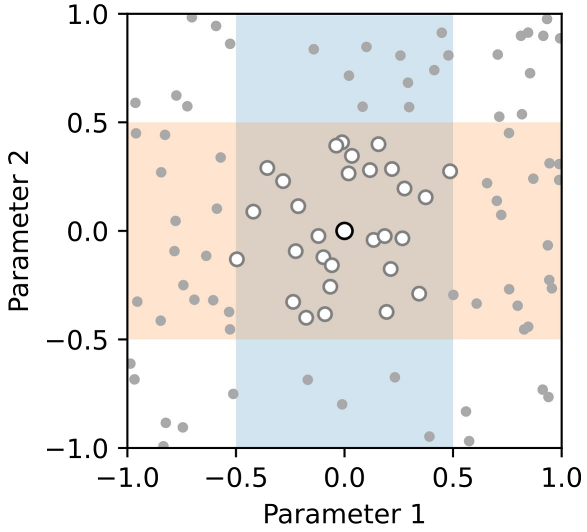


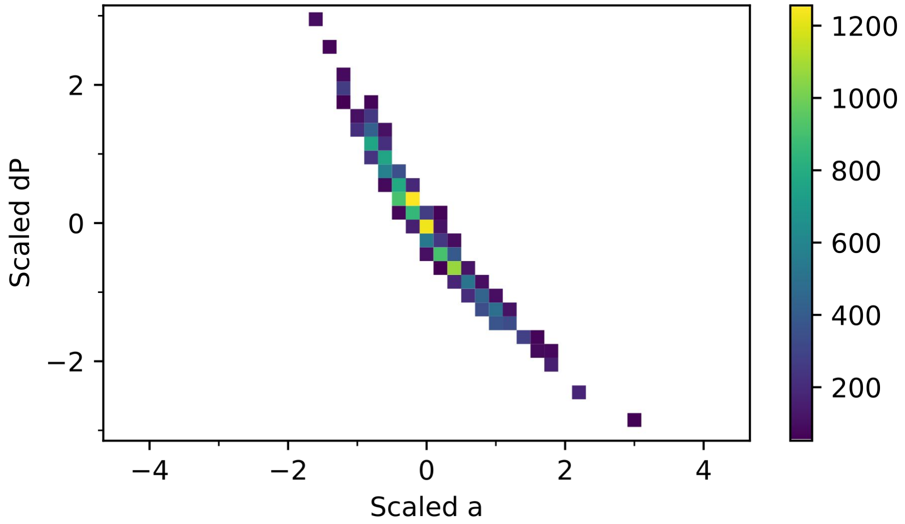
ENSEMBLE DIVISION

import StandardScaler

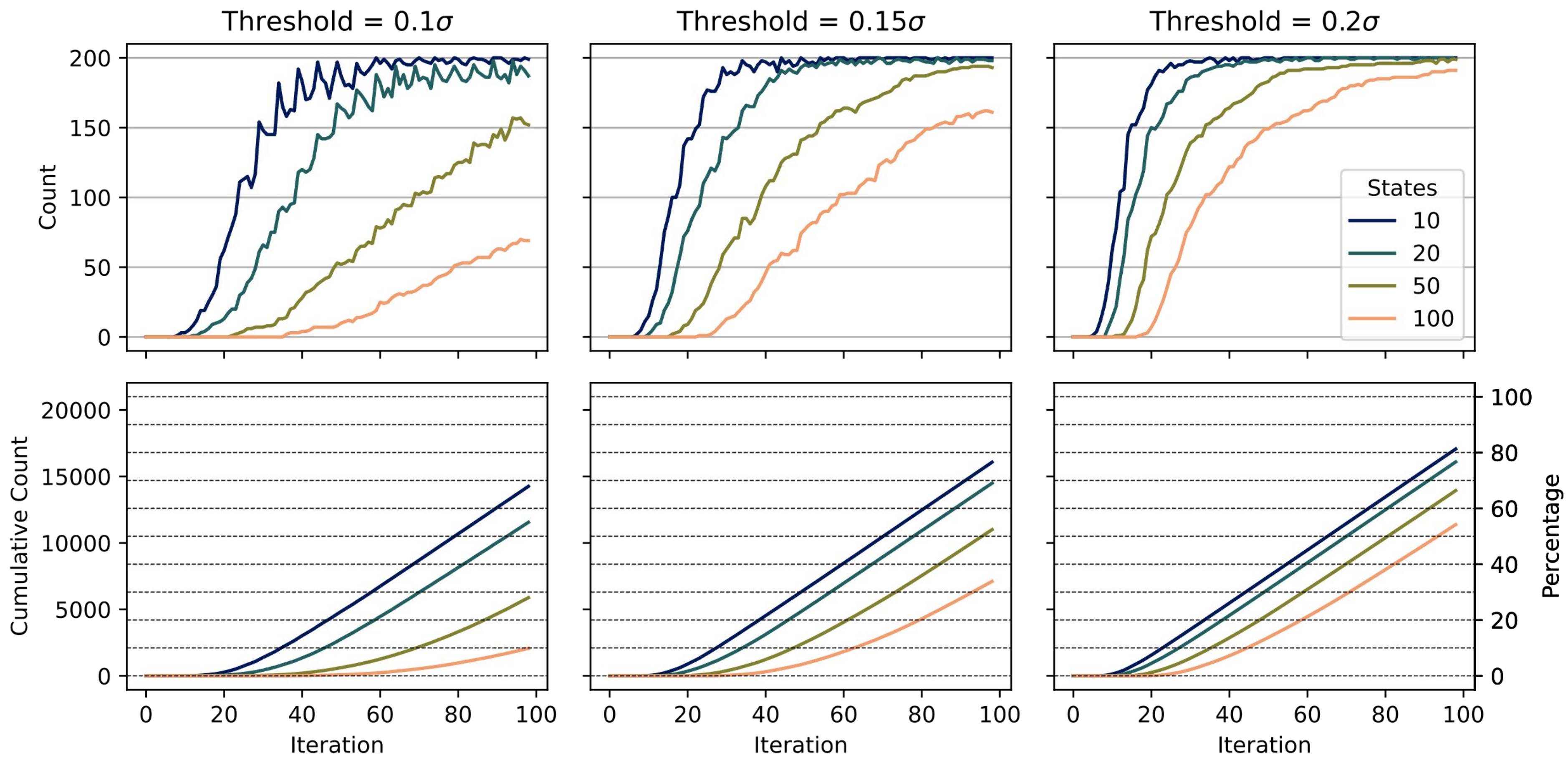
- Standardise model parameters

- Criteria to identify 'similar' model states
  - Threshold, as function of variance
  - Count within threshold





ENSEMBLE DIVISION

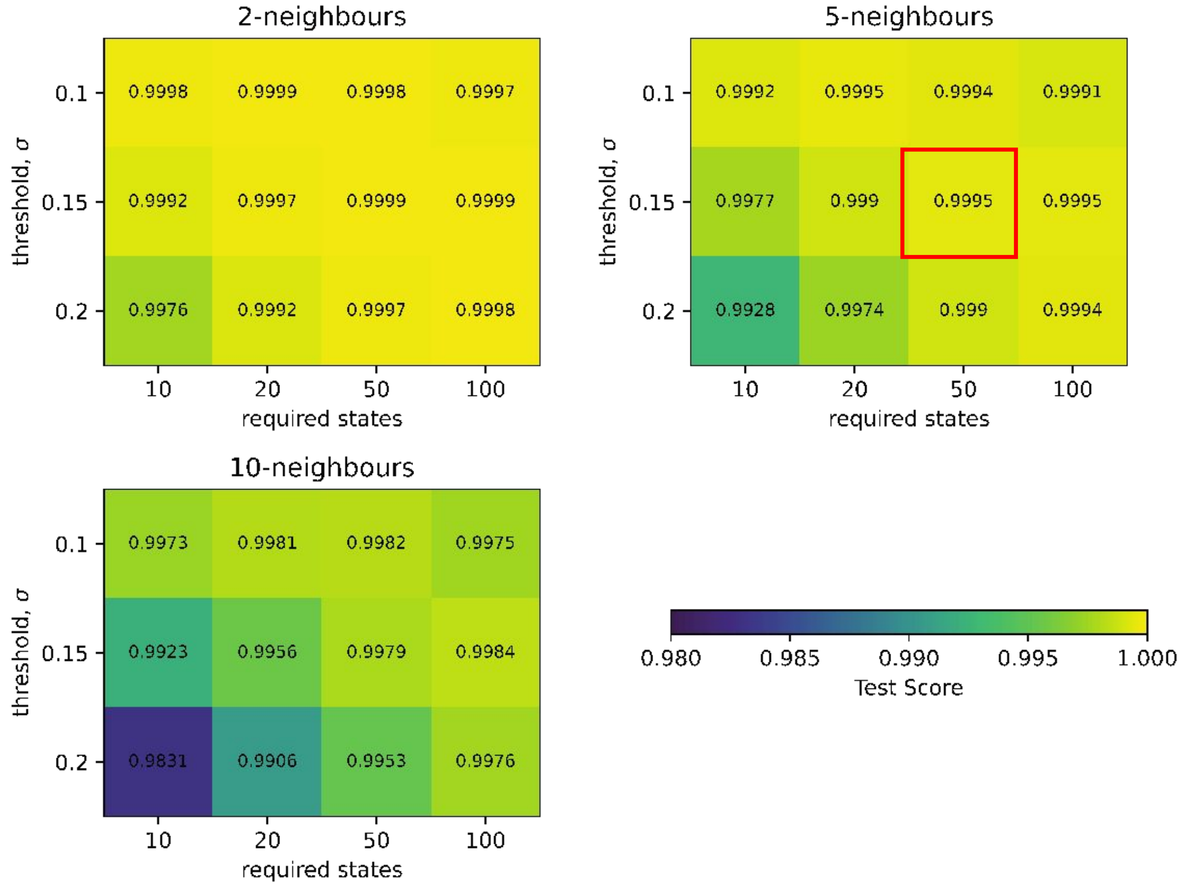


MODEL EVALUATION

KNeighborsRegressor.score()

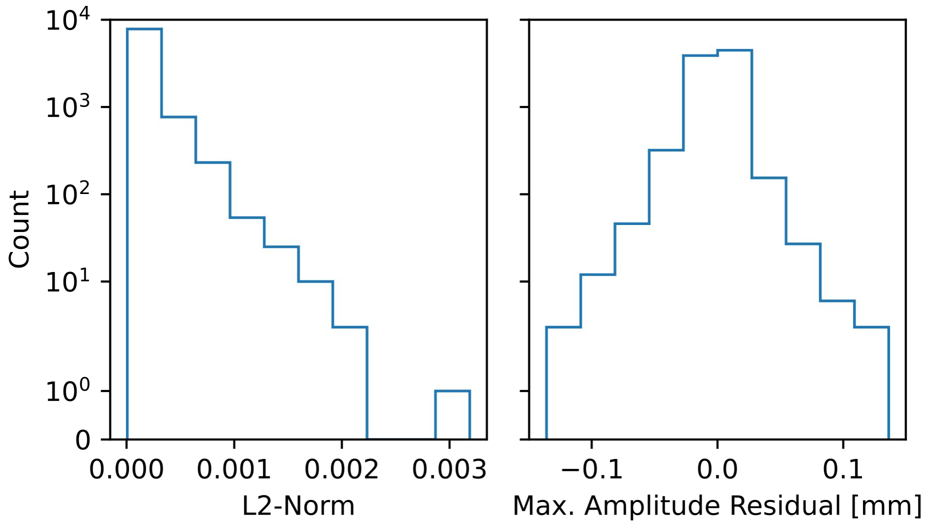
- Consider different numbers of neighbors, *k*

- Proportional to
  - number of required states
- Inversely proportional to
  - number of neighbours
  - threshold width



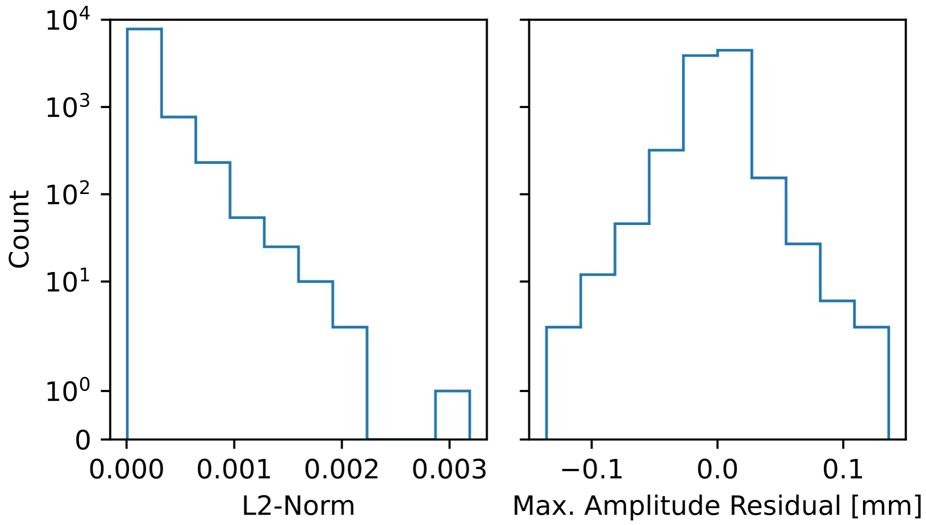
MODEL EVALUATION

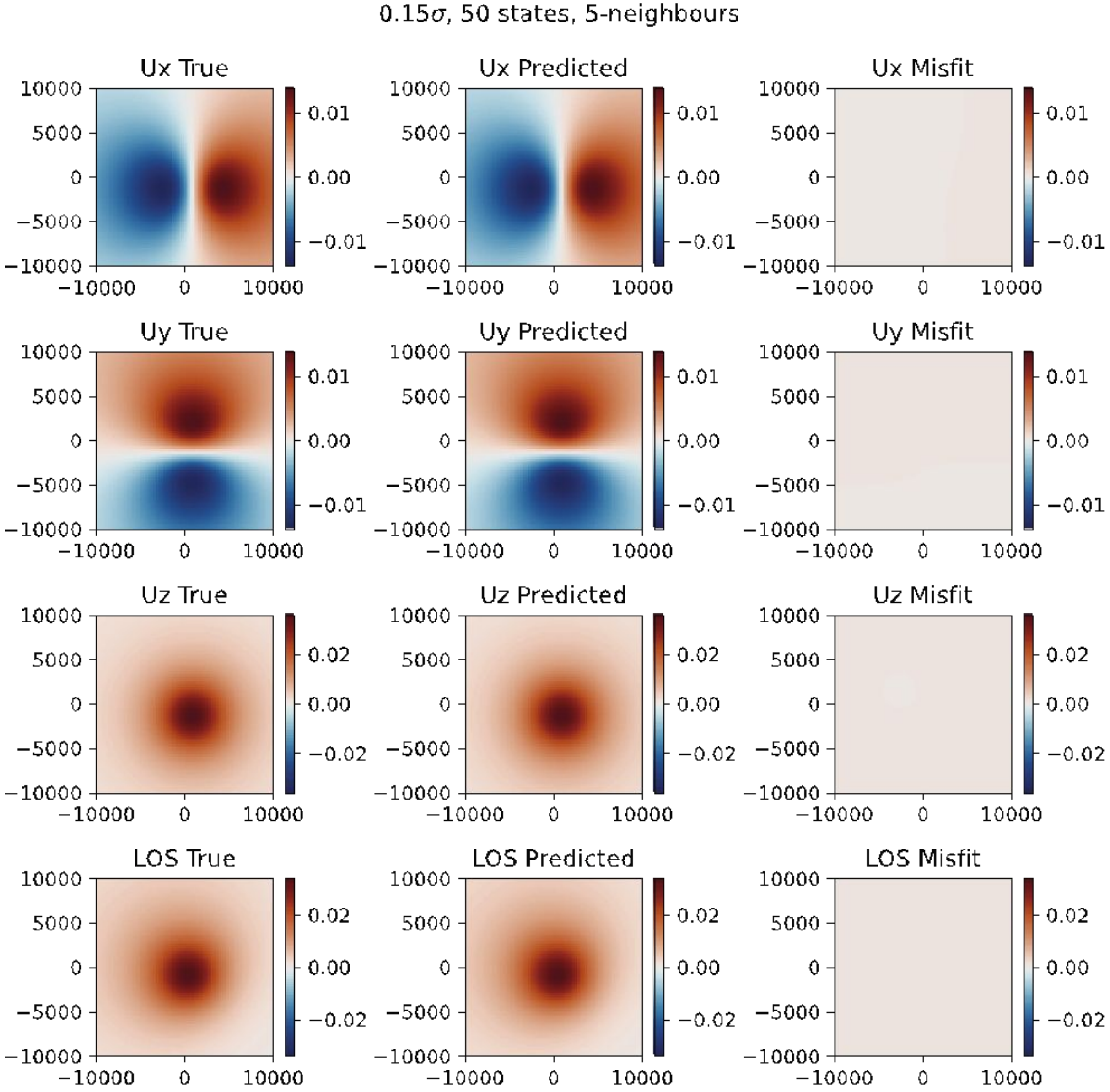
- L2-Norm
- Maximum Amplitude Residual



MODEL EVALUATION

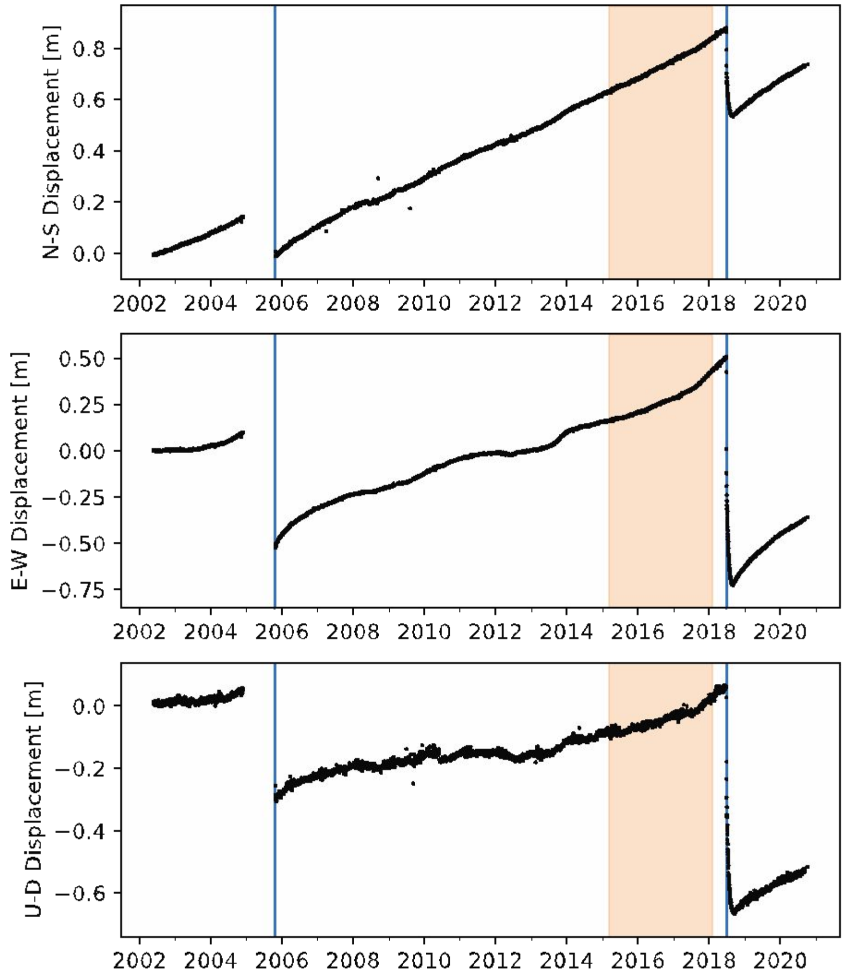
- L2-Norm
- Maximum Amplitude Residual





FUTURE WORK

- Alternative analytical model
  - 'Yang' ellipsoid (Yang et al., 1988)
  - 8 free parameters
- Overpressure trajectories
- Finite Element Model
  - Topography
- Alternative ML algorithms?
  - k-NN performance decreases with added complexity



THANK YOU FOR LISTENING!

ANY QUESTIONS?

Many thanks to our colleagues and collaborators:
Jack Albright, Andy Bell, Trish Gregg, Pete LaFemina, Zhong Lu, Yan Zhan