FitSNAP uncertainty estimation and propagation

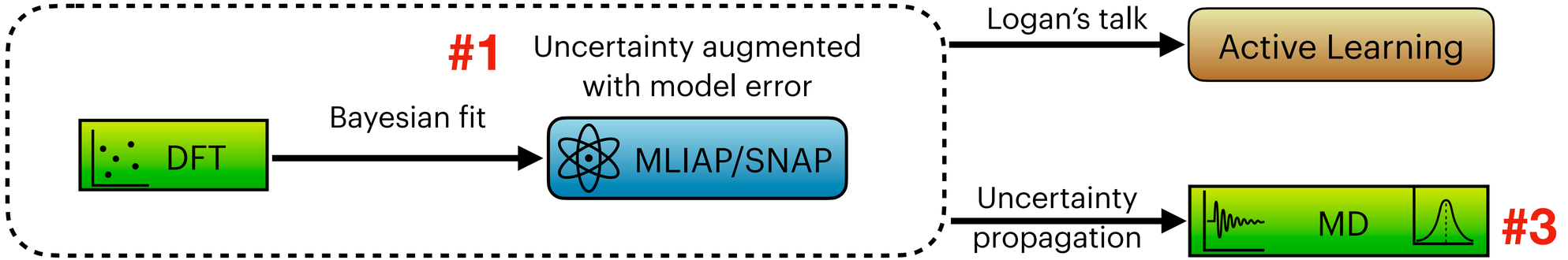
FusMatML All-Hands, Livermore, CA

Sep 8, 2022

Khachik Sargsyan, Logan Williams, Habib N. Najm (SNL-CA)

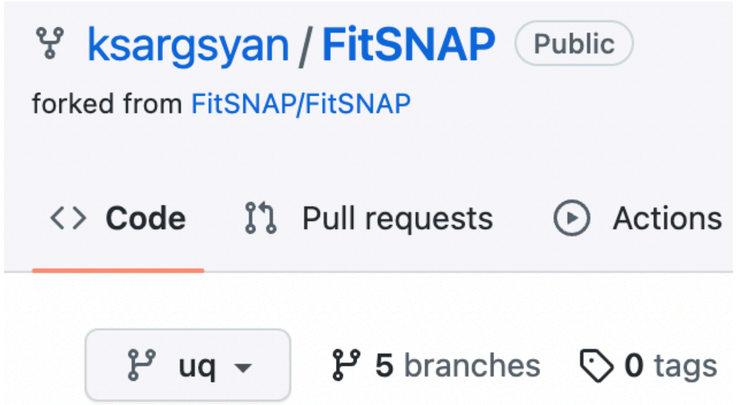
Overview

UQ use case

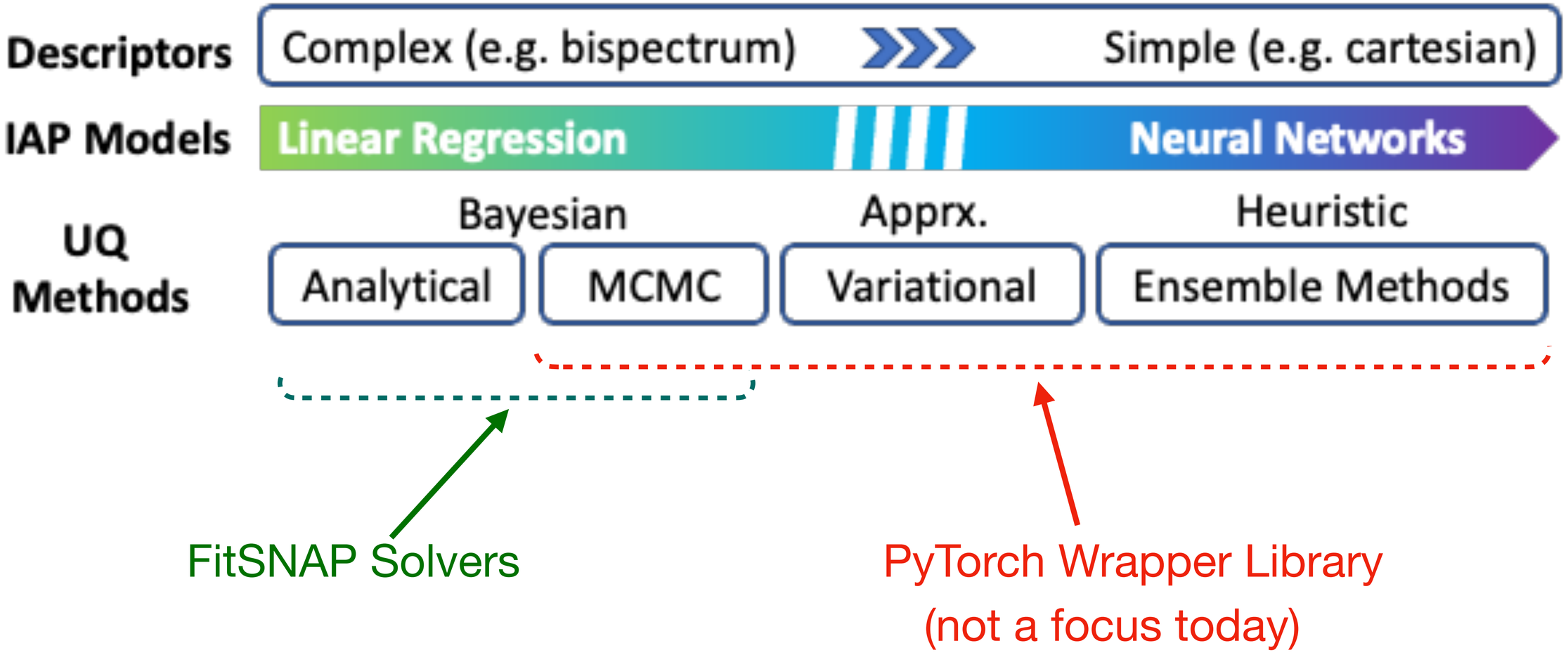


#2

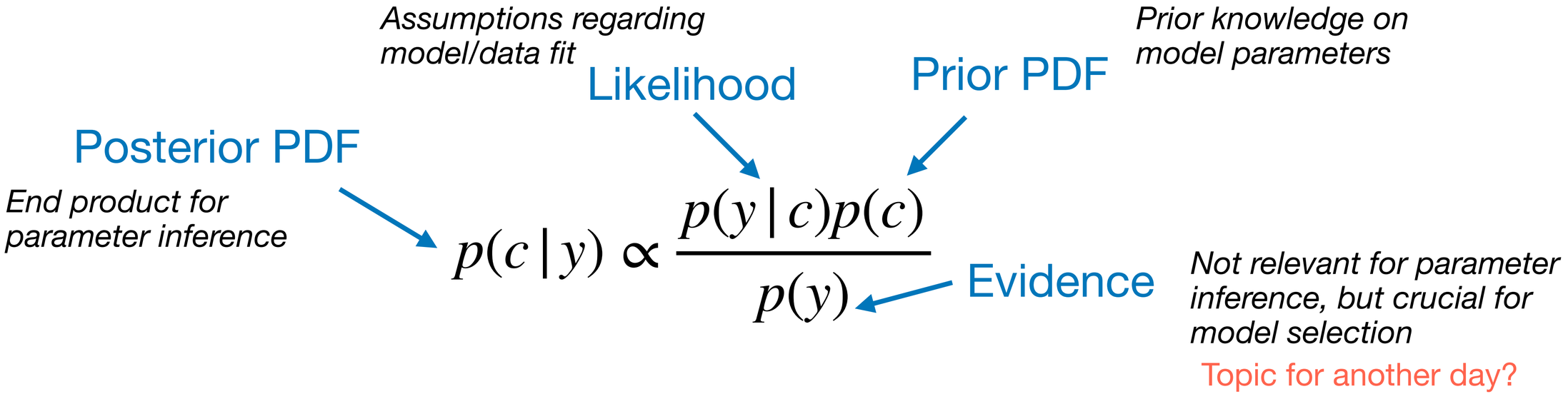
FitSNAP-UQ fork implementation details



#1 Equipping IAPs with uncertainties



Bayesian Parameter Inference



- No closed form for posterior PDF unless very specialized likelihoods are used
- Need to resort to sampling the posterior, rather than evaluating directly
- Markov chain Monte Carlo (MCMC) is the main vehicle for posterior sampling

Bayesian parameter inference is a generalization of deterministic least-squares

- Given a model $f(x, c)$ and data $y_i = y(x_i)$, calibrate parameters c.
  - Linear model $y \approx Ac$ with coefficients c
  - NN model $y \approx NN_c(x)$ with weights/biases c
- Weighted least-squares fit: $$c^* = \text{argmin}_c \sum_{i=1}^{N} w_i^2 \left(f(x_i, c) - y_i\right)^2$$
- Bayesian equivalent: $$p(c\,|\,y) \propto p(y\,|\,c)\; p(c) \propto \prod_{i=1}^{N} \exp\left(-\frac{(f(x_i, c) - y_i)^2}{2\sigma_i^2}\right)$$

$p(y\,|\,c)$: Gaussian i.i.d. Likelihood

$p(c)$: Uniform Prior

Crucial piece: assumptions for likelihood (or data model, or noise model)

Model Data

Likelihood $$p(y \mid c) \propto \prod_{i=1}^{N} \exp\left(-\frac{(f(x_i, c) - y_i)^2}{2\sigma_i^2} \right)$$

✦ Likelihood contains data noise modeling assumptions,

e.g. $$y_i = f(x_i, c) + \sigma_i \epsilon_i \text{ , where } \epsilon_i \sim \mathcal{N}(0,1)$$

Data Model

Linear Models (e.g. SNAP): analytical posterior PDF

$$f(x;c) = \sum_{k=1}^{K} c_k B_k(x) = Ac$$

A is the design matrix, $A_{ik} = B_k(x_i)$

Likelihood

$$p(y\,|\,c,\sigma) \propto \prod_{i=1}^{N} \exp\left(-\frac{(f(x_i,c)-y_i)^2}{2\sigma^2}\right)$$

Data Model

$$y_i = f(x_i,c) + \sigma\epsilon_i \text{ , where } \epsilon_i \sim \mathcal{N}(0,1)$$

DFT SNAP

Posterior PDF

$$p(c\,|\,y) \sim \mathcal{N}((\underbrace{(A^TA)^{-1}A^Ty}_{c^*}, \underbrace{\hat{\sigma}^2(A^TA)^{-1}}_{\Sigma})$$

The deterministic FitSNAP answer

The results of ‘anl’ FitSNAP-UQ solver

Pushed forward (PF) prediction

$$p(f\,|\,y) \sim \mathcal{N}\left(Ac^*, A\Sigma A^T\right)$$

Posterior PF uncertainty does not capture true discrepancy

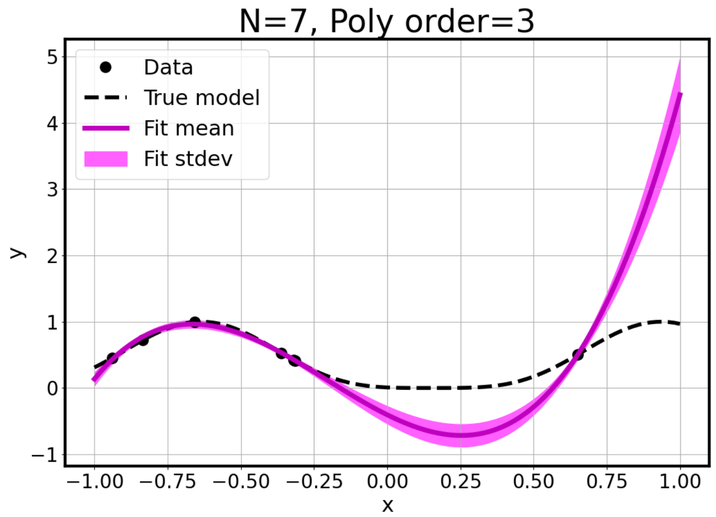
Synthetic data

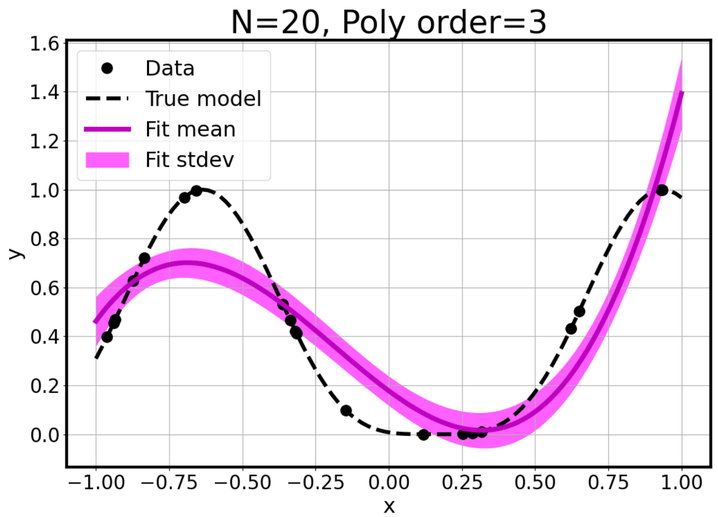
$$y(x) = \sin^4(2x - 0.3)$$

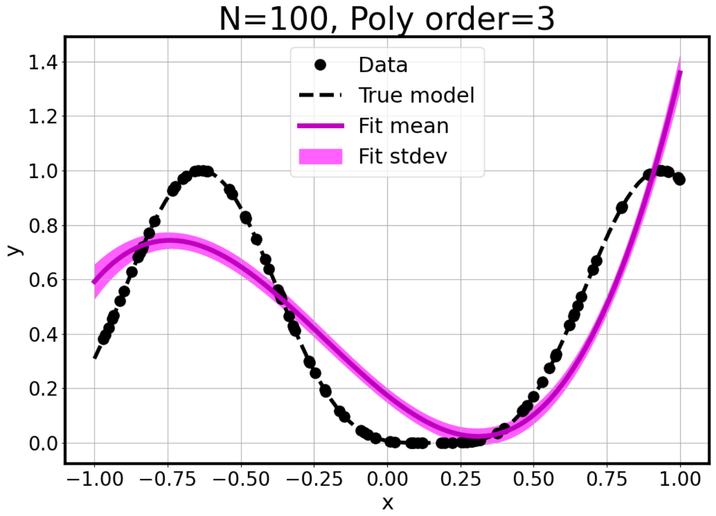
Cubic fit

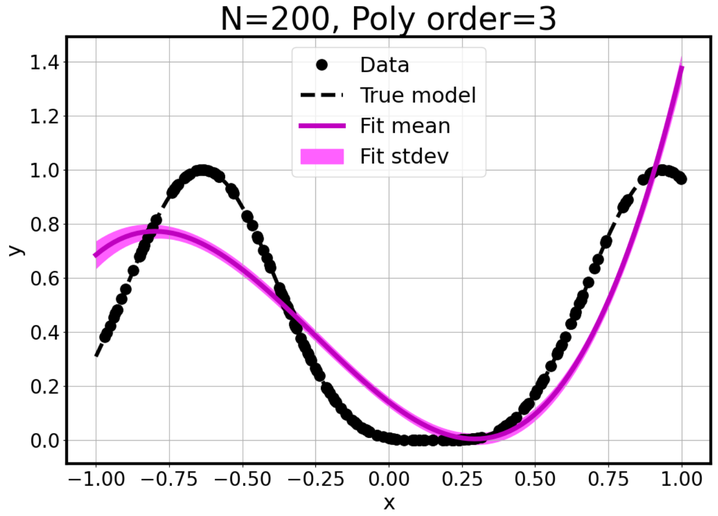
$$y_i \approx \sum_{k=0}^{3} c_k B_k(x)$$

More data leads to overconfident prediction









Better data model that captures model error

Conventional data model

$$y_i = \sum_{k=0}^{K} c_k B_k(x_i) + \sigma\epsilon_i \text{ , where } \epsilon_i \sim \mathcal{N}(0,1)$$

DFT: y_i — SNAP: $\sum_{k=0}^{K} c_k B_k(x_i)$ — Error: $\sigma\epsilon_i$

Maximize, or sample from $(c_0, \ldots, c_K, \sigma)$

Embedded model error

$$y_i \approx \sum_{k=0}^{K} (c_k + d_k\epsilon_k)\, B_k(x) = \sum_{k=0}^{K} c_k B_k(x) + \sum_{k=0}^{K} d_k B_k(x)\epsilon_k$$

SNAP: $\sum_{k=0}^{K} c_k B_k(x)$ — Error: $\sum_{k=0}^{K} d_k B_k(x)\epsilon_k$

Embed uncertainty in coefficients

(still Gaussian, but correlated, and model-informed)

Maximize, or sample from $(c_0, \ldots, c_K,\ d_0, \ldots, d_K)$

Embedded Model Error comes with a price:

Likelihood approximations are needed

$$y_i \approx \sum_{k=0}^{K} (c_k + d_k \xi_k)\, B_k(x) \quad = \quad \sum_{k=0}^{K} c_k B_k(x) + \sum_{k=0}^{K} d_k B_k(x) \xi_k$$

Option 1 (ABC)

$$p(c, d \,|\, y) \propto \prod_{i=1}^{N} \exp\left(- \frac{(\sum_{k=0}^{K} c_k B_k(x_i) - y_i)^2 + (\sqrt{\sum_{k=0}^{K} d_k^2 B_k^2(x_i)} - \alpha\, |\sum_{k=0}^{K} c_k B_k(x_i) - y_i|)^2}{2\epsilon^2} \right)$$

Option 2 (IID)

$$p(c, d \,|\, y) \propto \prod_{i=1}^{N} \exp\left(- \frac{(\sum_{k=0}^{K} c_k B_k(x_i) - y_i)^2}{2 \sum_{k=0}^{K} d_k^2 B_k(x_i)^2} \right)$$

MCMC sampling of c, d

or

simply maximize the posterior for c, d

Pushed forward predictive uncertainty captures the true discrepancy from the data

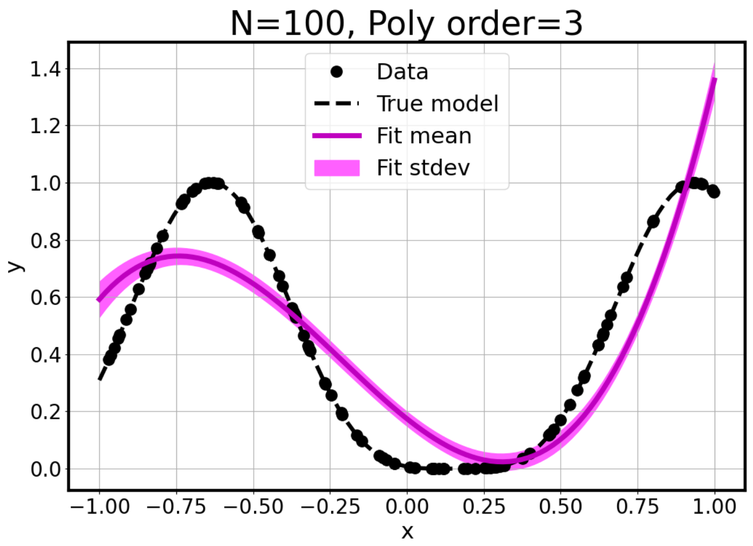
Synthetic data

$$y(x) = \sin^4(2x - 0.3)$$

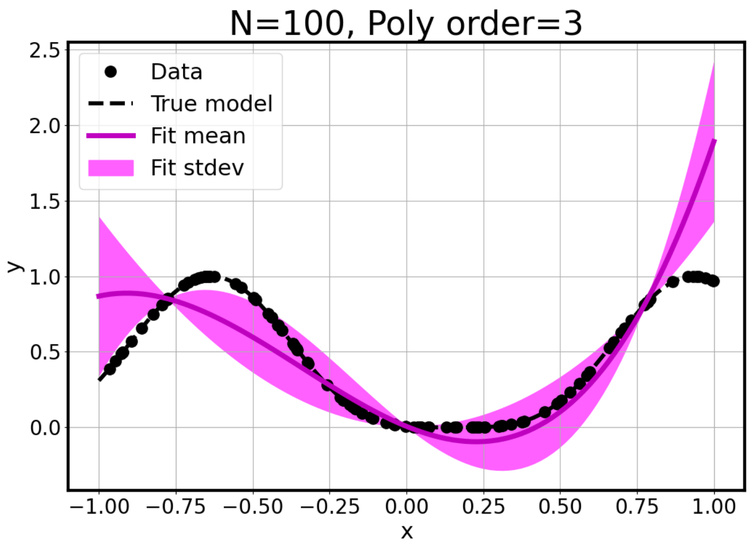
Cubic fit

$$y_i \approx \sum_{k=0}^{3} c_k B_k(x)$$

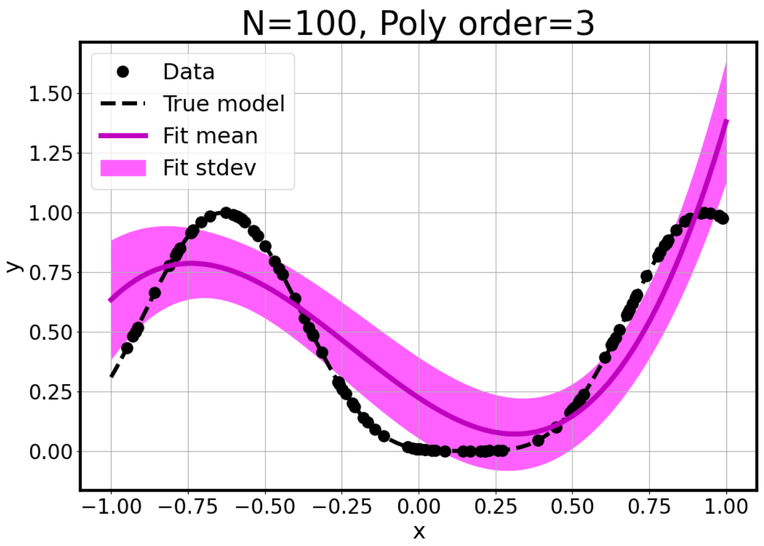
Classical case



Model error, IID likelihood

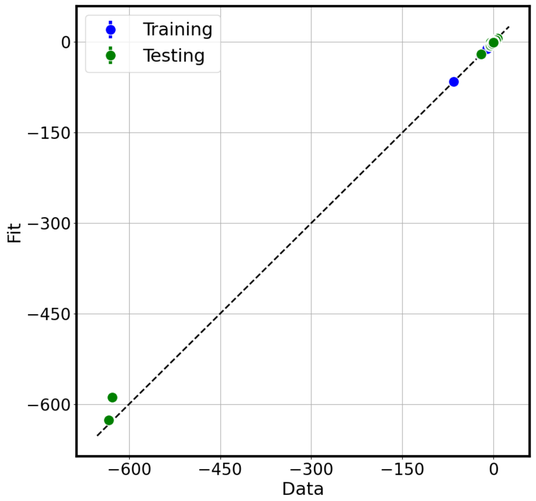


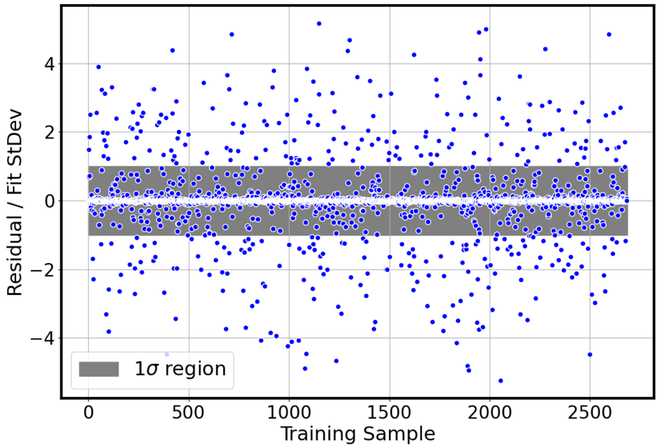
Model error, ABC likelihood

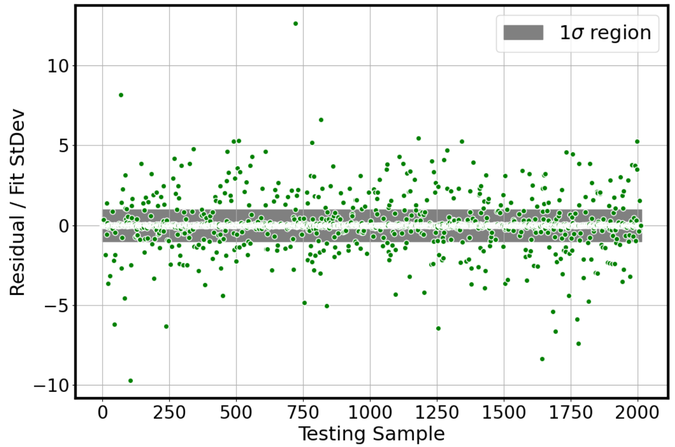


W-ZrC Dataset

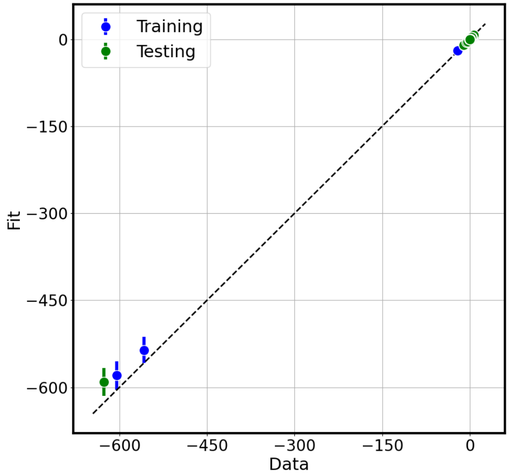
Uncertainty without model error

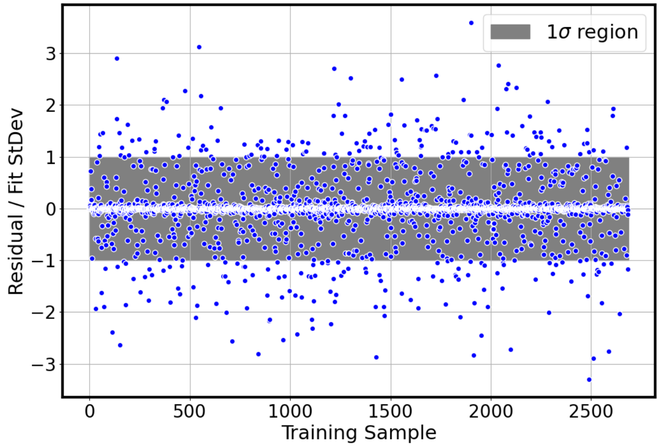


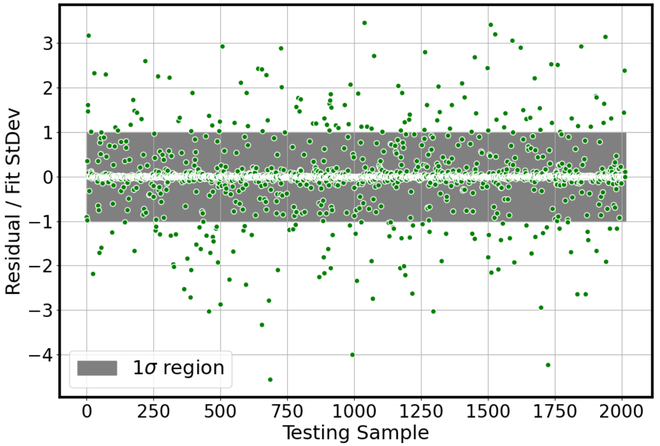




Uncertainty with model error







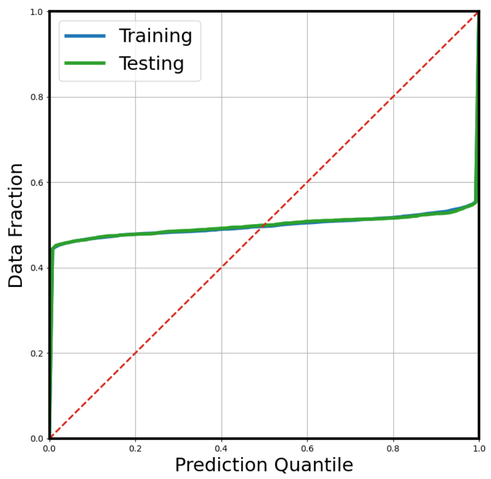
Uncertainty validation: two examples

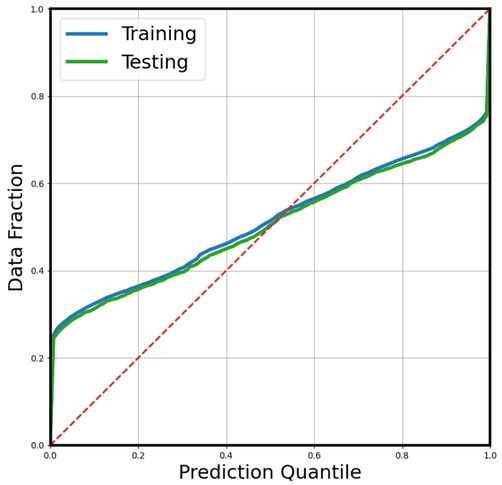
Classical case

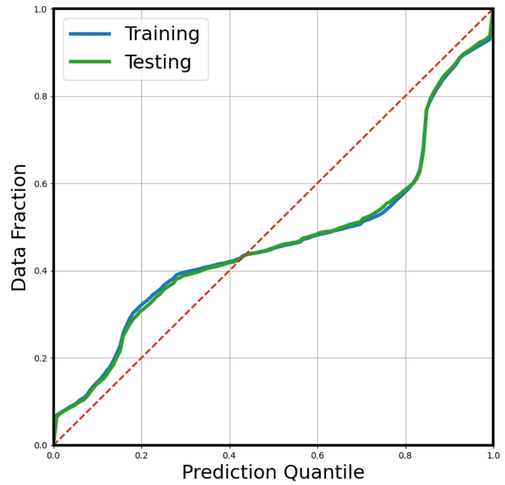
Model error, IID likelihood

Model error, ABC likelihood

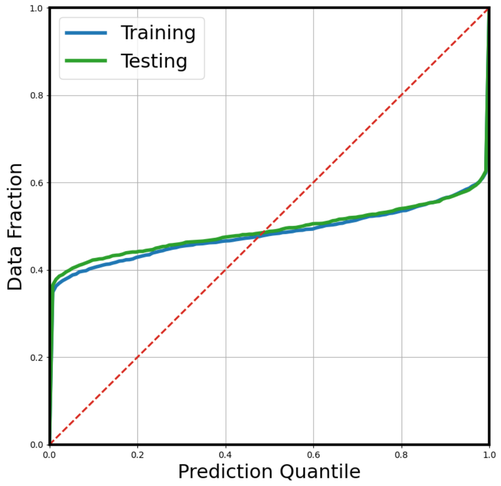
Ta

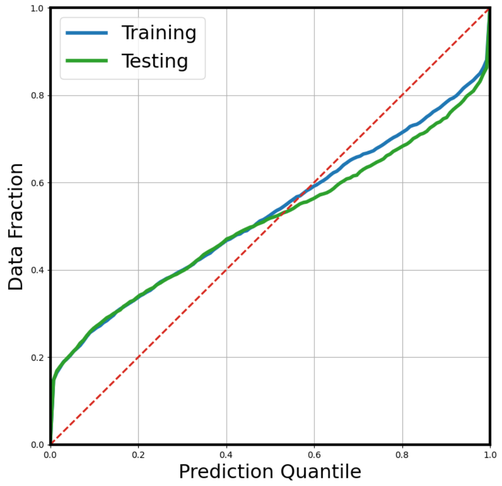


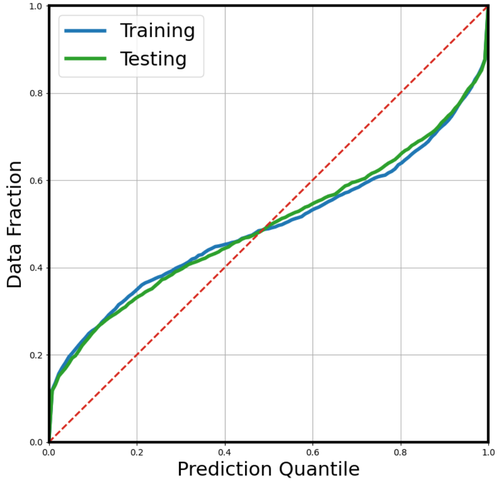




W-ZrC







Model error: choices and challenges

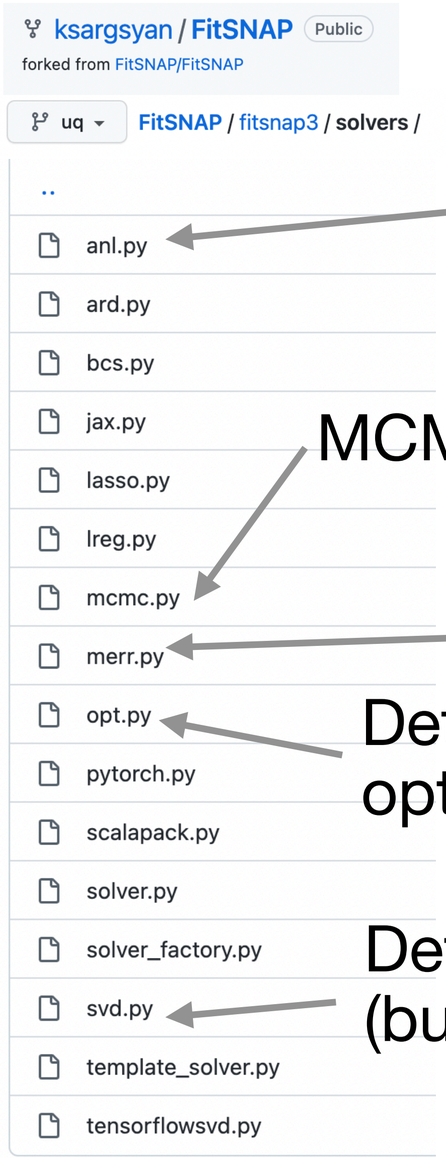
- Embedding type: *e.g.*

additive $y_i \approx \sum_{k=0}^{K} (c_k + d_k \xi_k)\, B_k(x)$ or multiplicative $y_i \approx \sum_{k=0}^{K} (c_k + c_k d_k \xi_k)\, B_k(x)$

- Degenerate (Gaussian) likelihoods: resort to approximate Bayesian computation (ABC) or independent (IID) assumptions

- Difficult posterior PDFs for MCMC, choice of priors for embedding parameters
- Which coefficients to embed the model error in? Later we inform this by MD GSA
- Connect predictive uncertainty and the residual error with an extrapolation metric
- Major challenge: data sizes are large, linear algebra chokes

#2 FitSNAP-UQ options so far



Analytical Bayesian linear regression

```
[SOLVER]
solver = ANL
nsam = 133
cov_nugget = 1.e-10
```

MCMC sampling

```
[SOLVER]
solver = MCMC
nsam = 133
mcmc_num = 1000
mcmc_gamma = 0.01
```

Embedded model error

```
[SOLVER]
solver = MERR
nsam = 133
cov_nugget = 1.e-10
merr_method = abc
merr_mult = 1
merr_cfs = 0 2 5
```

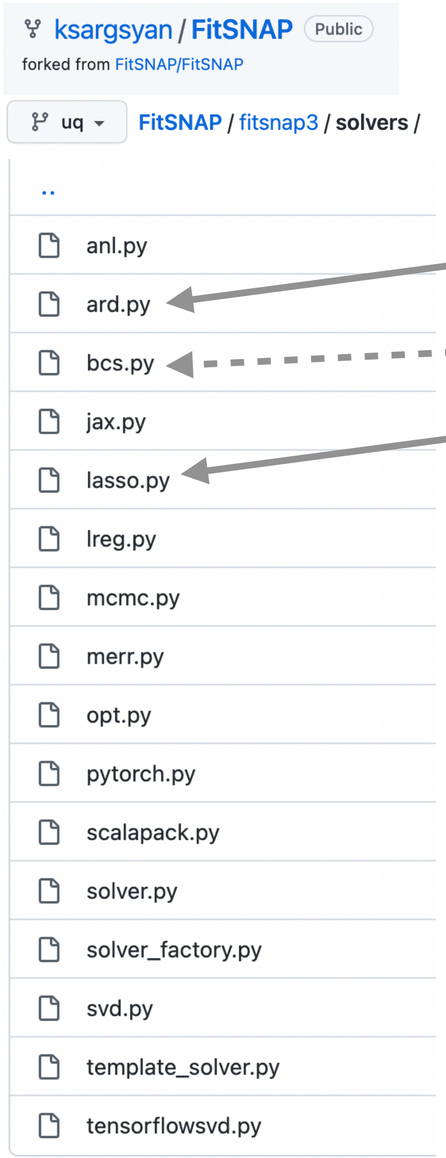
Deterministic optimization

```
[SOLVER]
solver = OPT
```

Deterministic least-squares (business-as-usual)

```
[SOLVER]
solver = SVD
```

FitSNAP sparse learning



Currently in FitSNAP, deterministic

UQ analogue

Automatic relevance determination

Bayesian Compressive Sensing

Lasso regression

```
[SOLVER]
solver = BCS
nsam = 133
```

Currently experimental, without bells and whistles

Topic for another day?

- Sparse learning prunes some of the coefficients as part of the fit
- Achieves dimensionality reduction (important, e.g. for uncertainty propagation later)

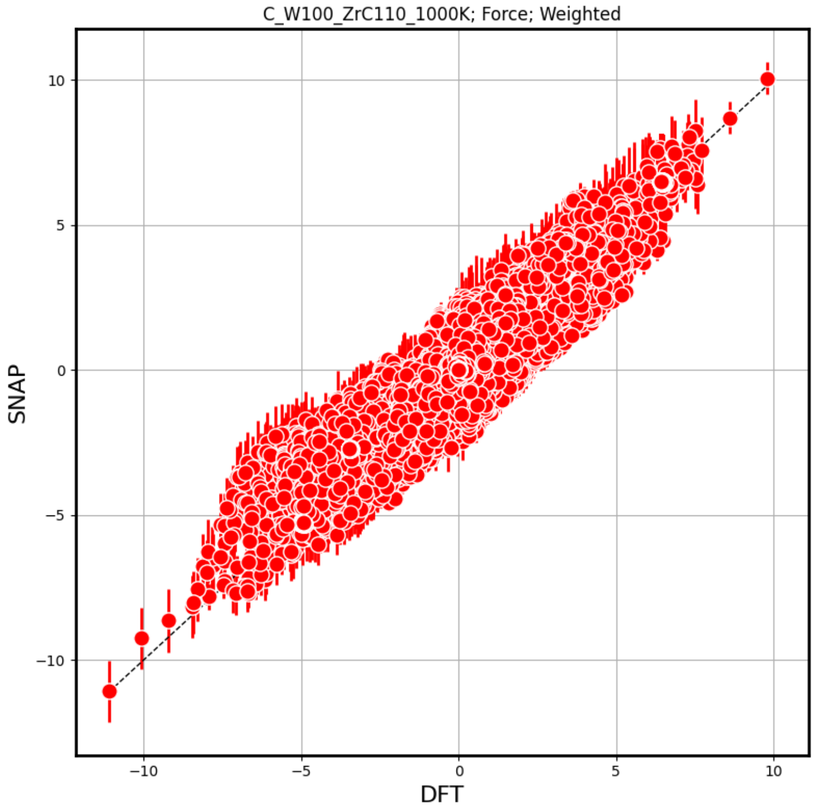
$$\sum_{k=0}^{K} c_k B_k(x)$$

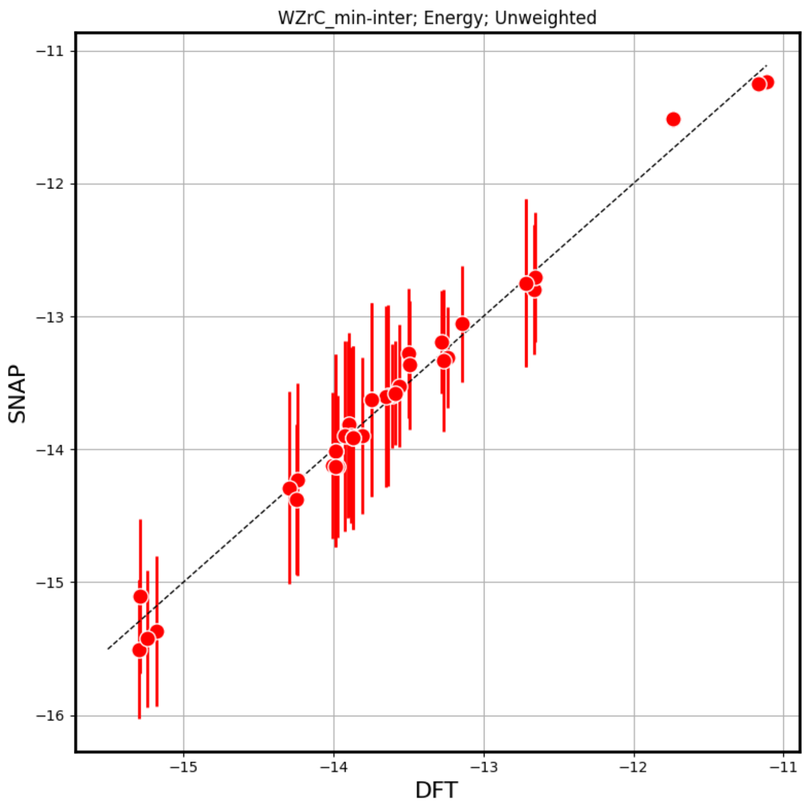
- Currently, FitSNAP fills the pruned coefficients with 0

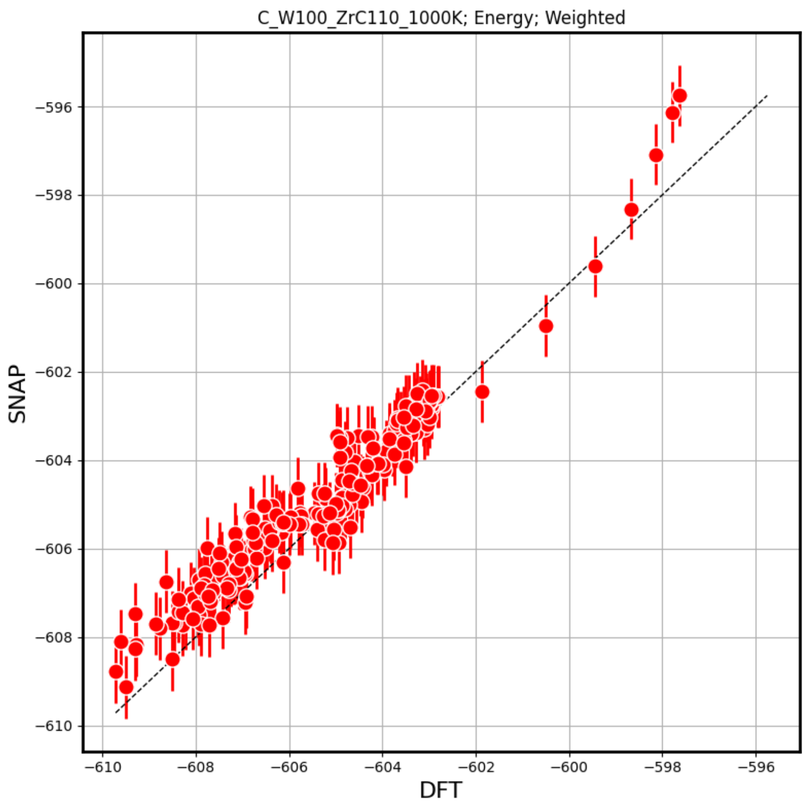
Plotting option added for quick visuals

```
[EXTRAS]
plot = 2
```

- Options are 0 (no plot), 1 (mean SNAP), 2 (mean + errorbars),
- Plots 'diagonal' plots (DFT-vs-SNAP) for all groups, weighted and unweighted,
- Requires matplotlib, and may take time to generate all png files.







FitSNAP-UQ: end product

Coefficient Samples

UQ solvers create the requested number (nsam) of snapcoeff files, e.g.

Sample # 25

WZrC_pot_025.snapcoeff

```
# fitsnap fit generated on 2021-10-30 00:46:00.217256

3 31
W 0.4033335777 1.0
 -15.3500742786061313          #  B[0]
 0.0325450949477414861         #  B[1, 0, 0, 0]
 0.0786028816107458284         #  B[2, 1, 0, 1]
 -0.144382530698556222         #  B[3, 1, 1, 2]
 -0.738222137312165794         #  B[4, 2, 0, 2]
 -0.140285804611108733         #  B[5, 2, 1, 3]
 0.288874324985492814          #  B[6, 2, 2, 2]
 -0.0217902885756043087        #  B[7, 2, 2, 4]
 -0.377006851861265813         #  B[8, 3, 0, 3]
 -0.845242239468904089         #  B[9, 3, 1, 4]
 1.11733162664206276           #  B[10, 3, 2, 3]
```

Samples can be created as a postprocessing step, too, given Coefficient's mean and covariance.

Coefficient Mean/Covariance pair

For all but MCMC option, the coefficients are multivariate normal: mean.npy and covariance.npy are produced

anl: $\mathcal{N}(\underbrace{(A^TA)^{-1}A^Ty}_{c*}, \underbrace{\hat{\sigma}^2(A^TA)^{-1}}_{\Sigma})$

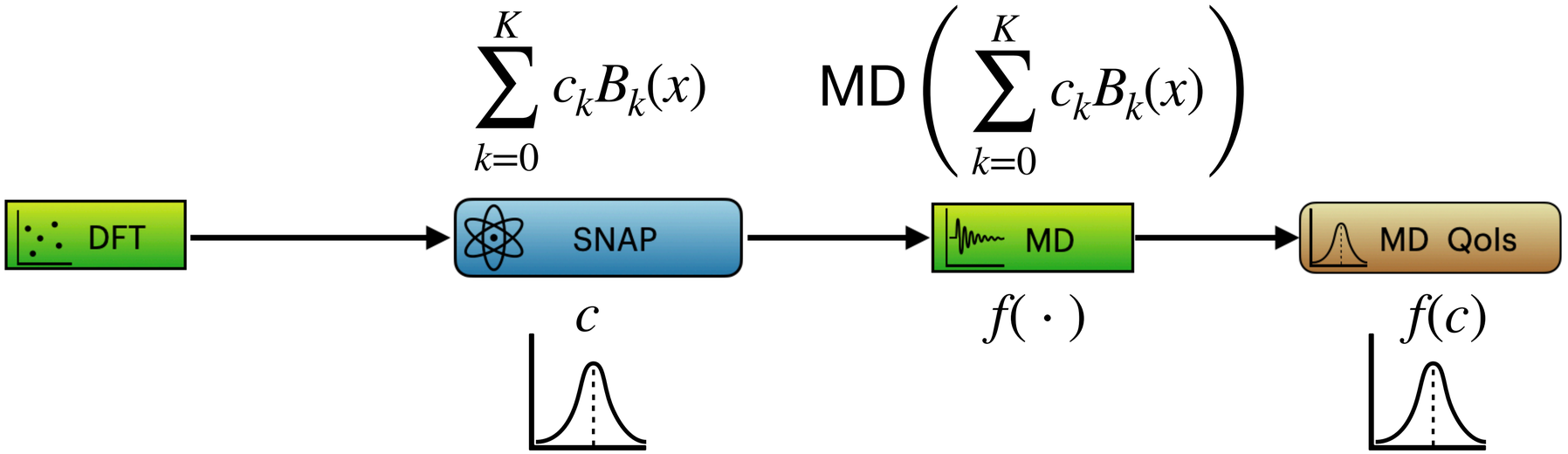
merr: $\mathcal{N}(\hat{c}, \quad \text{diag}(\hat{d}^2))$ additive

$\mathcal{N}(\hat{c}, \quad \hat{c}^2 \cdot \text{diag}(\hat{d}^2))$ multiplicative

Note 1: some elements of $\hat{d}$ can be zero
(embed model error in select coefficients only)

Note 2: instead of best $(\hat{c}, \hat{d})$ one can sample (c, d) with MCMC

#3 Uncertainty Propagation through MD



Monte-Carlo Propagation: Sample c (with FitSNAP-UQ), evaluate MD for each sample, gather statistics.

Polynomial Chaos Propagation: See next.

Polynomial Chaos (PC) preliminaries

- Functional representation of a random variable

$$X = \sum_{j=0}^{J} x_j \Psi_j(\xi)$$

$\Psi_j(\cdot)$'s are orthogonal polynomials $\int_{\xi} \Psi_j(\xi)\Psi_i(\xi)\pi(\xi)d\xi = 0$, if $j \neq i$

ξ is a standard random variable

e.g., Hermite-Gauss (HG), $\xi \sim \mathcal{N}(0,1)$

or Legendre-Uniform (LU), $\xi \sim \mathcal{U}[-1,1]$

Convenient for :

- Moment estimation $\mathbb{E}[X] = x_0, \quad \mathbb{V}[X] = \sum_{j=1}^{J} x_j^2 ||\Psi_j||^2$
- Global sensitivity analysis (GSA) / variance decomposition
- Uncertainty propagation through a forward model $f(X)$

Uncertainty Propagation with PC

Input PC

$$X = \sum_{j=0}^{J} x_j \Psi_j(\xi)$$

Forward model

$$f(X)$$

Output PC

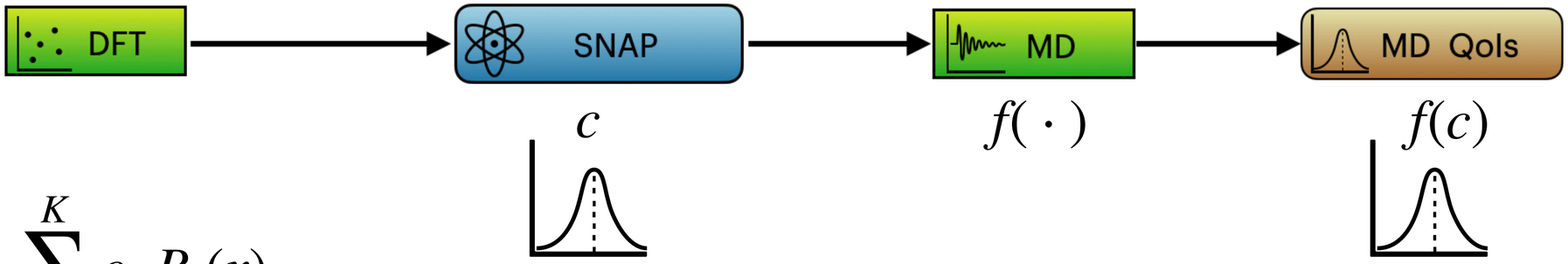
$$f(X) = \sum_{j=0}^{J} f_j \Psi_j(\xi)$$

Uncertainty propagation problem: given x_j 's, find f_j 's

Steps:

1. Sample stochastic 'germ' $\xi^{(n)}$
2. Compute corresponding model input samples $X^{(n)}$
3. Evaluate the model $f(X^{(n)})$
4. Use regression or projection to evaluate coefficients f_j 's
5. Postprocess: evaluate PDFs, moments, global sensitivity analysis

SNAP as a first order PC serves as input



$$\sum_{k=0}^{K} c_k B_k(x)$$

SNAP coefficients form a multivariate normal r.v.

(a.k.a. first order Gauss-Hermite PC)

$$c \sim \mathcal{N}(\mu, \Sigma)$$

Cholesky factorization

to build 1st order input PC

$$\Sigma = LL^T$$

$$\begin{aligned} c_0 &= \mu_1 + l_1\xi_1 \\ c_1 &= \mu_2 + l_1\xi_1 + l_2\xi_2 \\ &\vdots \qquad\qquad \cdots \\ c_K &= \mu_{K+1} + l_1\xi_1 + l_2\xi_2 + \cdots + l_{K+1}\xi_{K+1} \end{aligned}$$

Input PC

MD

$$f(c) = \sum_{j=0}^{J} f_j \Psi_j(\xi)$$

Output PC

W-ZrC SNAP → LAMMPS

MD (LAMMPS) Output QoIs:

- W bulk modulus
- ZrC bulk modulus
- W 100 surface energy
- W 110 surface energy
- W 111 surface energy
- ZrC 200 surface energy
- ZrC 110 surface energy

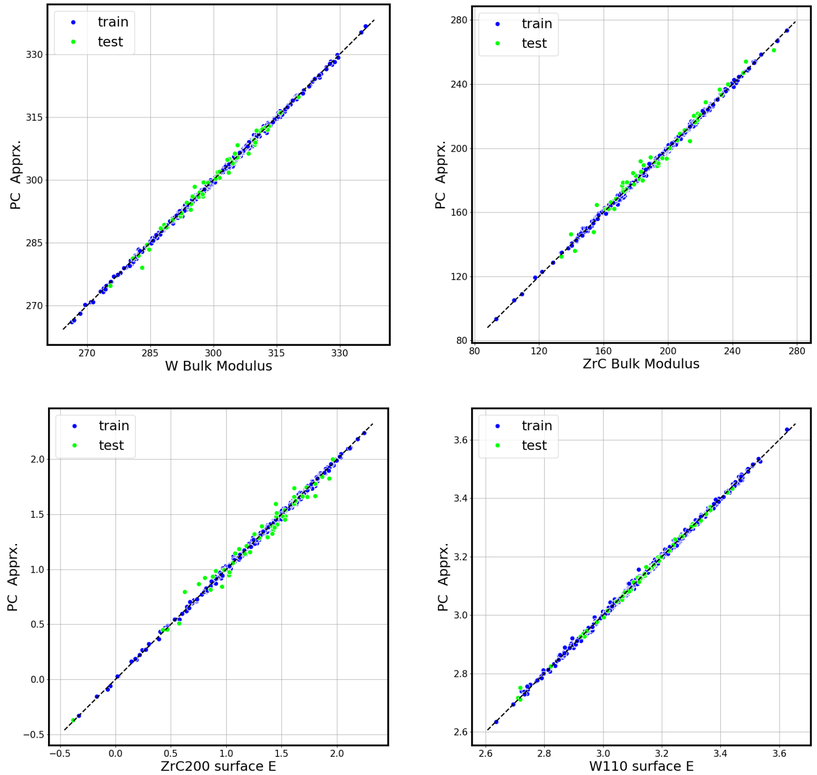
- The bulk moduli are the ones calculated by getting the system energy in LAMMPS of a crystal under different amounts of compression (different volumes) and fitting a curve to the E vs. V plot.

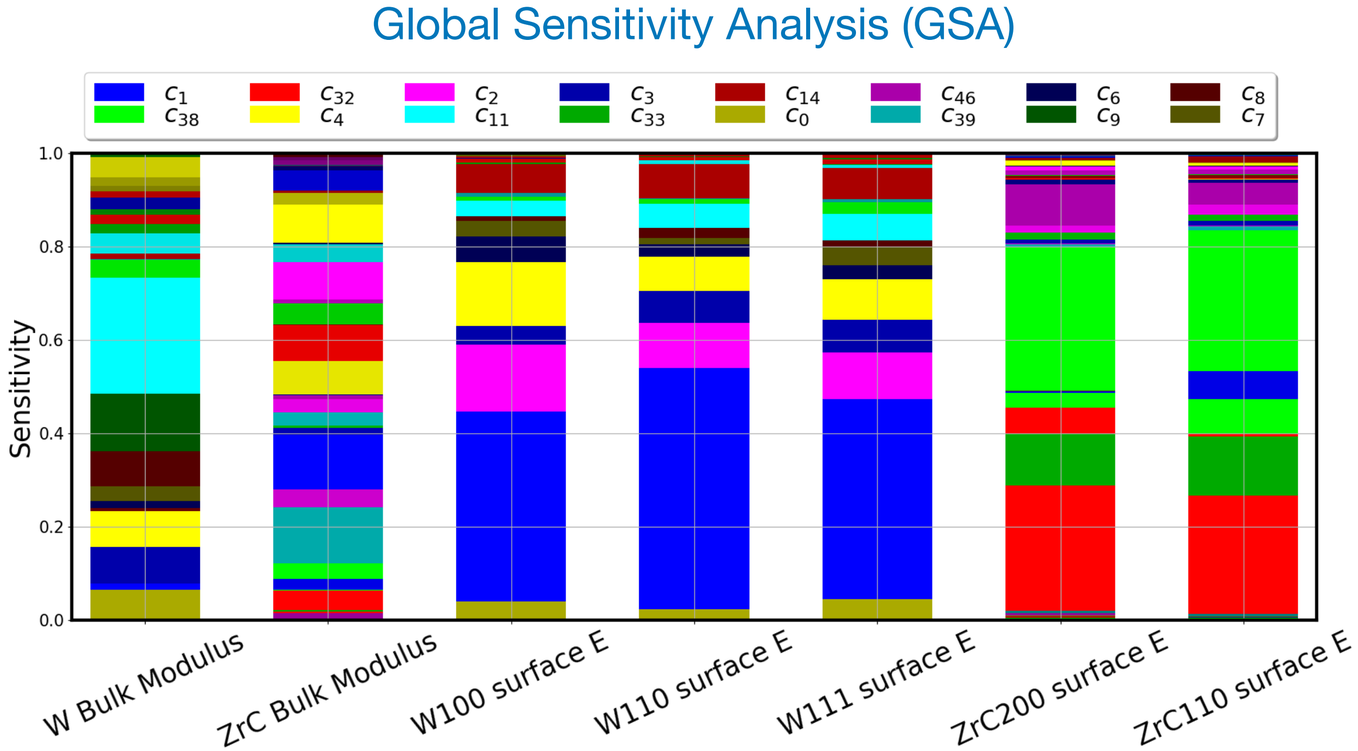
- The surface energies are calculated by taking a crystal and inserting a vacuum space at a given crystallographic surface (100, or 111, etc.) and calculating the system energy in LAMMPS (and subtracting the original crystal's energy). These are being calculated such that the atoms are allowed to relax.

Dimensionality reduction with GSA

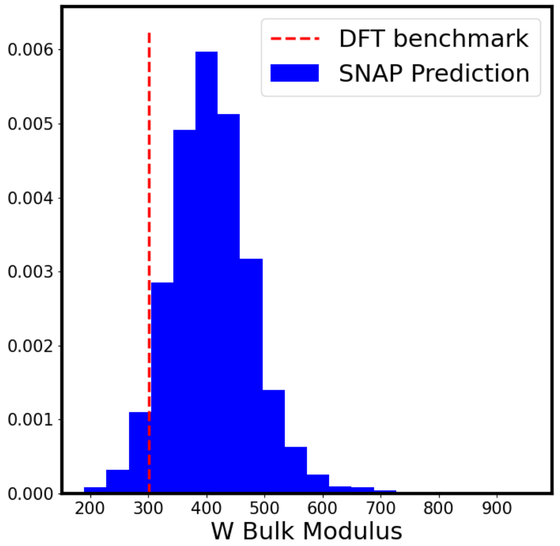
- Major challenge for model error construction and uncertainty propagation: high-dimensional input (too many bispectrum coefficients)

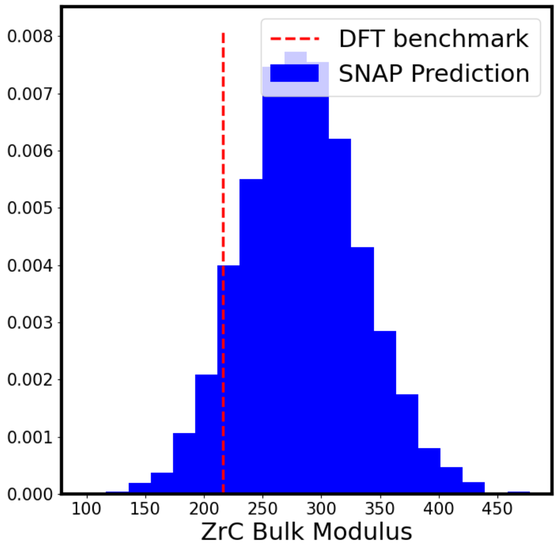
- Dimensionality reduction approach: perturb nominal coefficients, and decompose output variance into coefficient contributions (mechanically, this is Legendre-Uniform PC), and pick e.g. top 10

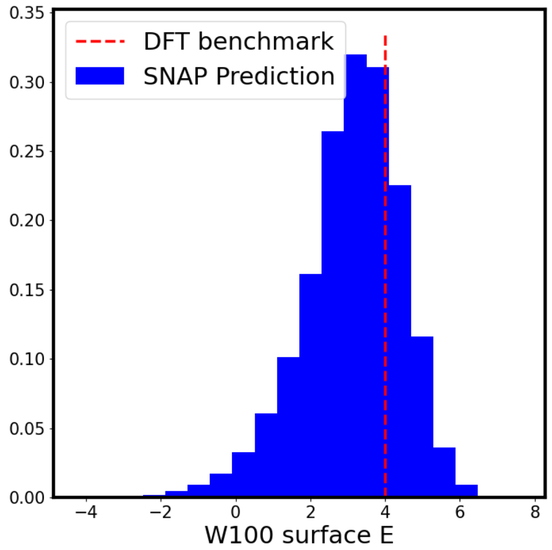


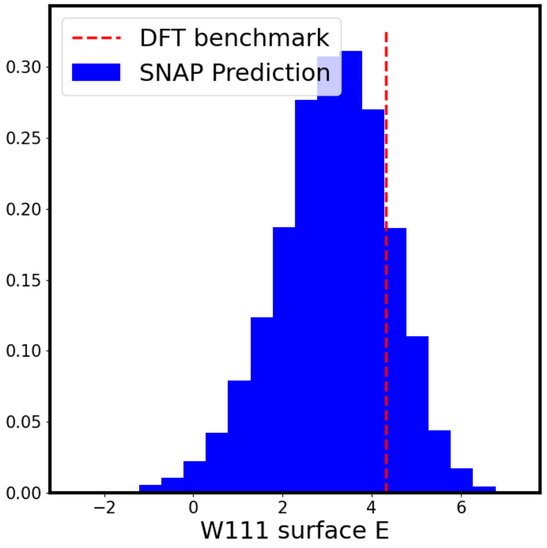


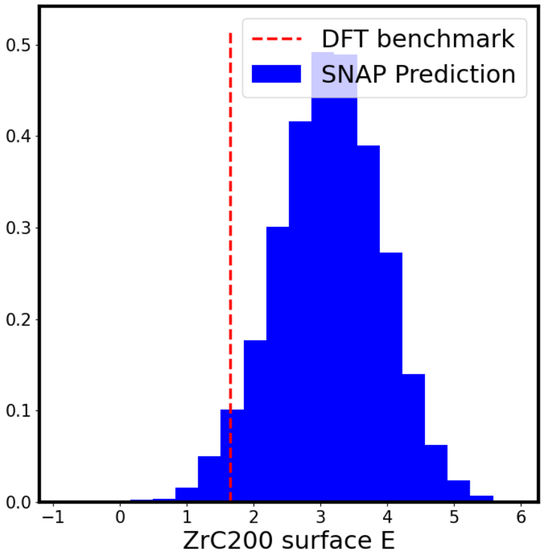
LAMMPS QoIs with uncertainty vs. DFT benchmarks

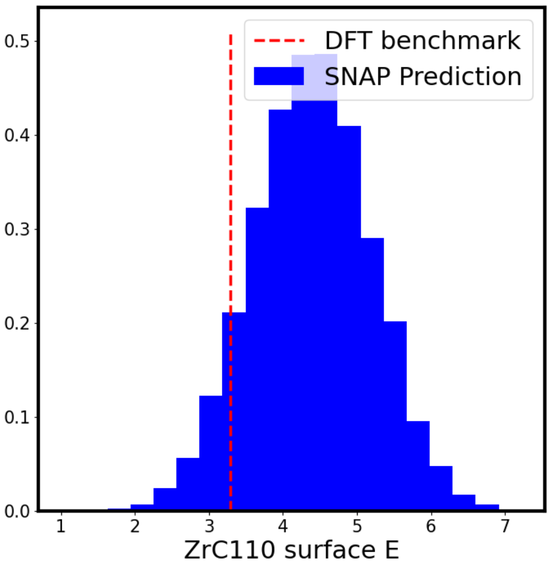












Uncertainty propagation: summary/challenges

- SNAP coefficient uncertainty propagation via Polynomial Chaos (PC) methods
- Demo on W-ZrC problem
- Global sensitivity analysis (GSA) quantifies relative importance of SNAP coefficients
- Helps inform/understand how to better fit SNAP

- High-dimensional input space (large number of SNAP coefficients)
  - Embed model error uncertainty in a few coefficients only, informed by a GSA
  - Use *sparse* PC methods as needed
- Challenging nonlinear/noisy MD QoIs
  - Need smooth/robust behavior with respect to SNAP coefficients
- Large uncertainties often trigger unphysical MD outputs

Summary

- **Embedded model error** leads to data model with baked-in uncertainty
  - Data model assumptions are crucial
  - Meaningful model-error uncertainty capturing the true residual better
  - Non-negligible coefficient uncertainty that can be propagated through MD
  - Many modeling/method decisions to make
- **FitSNAP-UQ fork**
  - ‘anl’ works as-is, but uncertainties are not well-calibrated
    - still useful for oiling the workflows
    - ways to improve while remaining analytically tractable
  - ‘merr’ is better, but many choices and hand-holding needed.
    - some choices are exposed in the user input file
    - need more automation, perhaps include model selection methods
- **Uncertainty propagation** of SNAP through MD/LAMMPS via polynomial chaos
  - Demo on W-ZrC problem
  - Global sensitivity analysis (GSA) quantifies relative importance of SNAP coefficients
  - Helps inform/understand how to better fit SNAP

Extras

Elephant in the room: model is assumed to be *the* correct model behind data

Model Data err.

$$y_i = f(x_i, c) + \sigma_i \epsilon_i$$

Truth

Model ≠ Truth

Ignoring model error hurts in a few ways:

✦ One gets biased estimates of parameters c (crucial if the model is physical, and/or c is propagated through other models)

✦ More data leads to overconfident predictions (we become more and more certain about the wrong values of the data)

✦ More evident when there is no (observational/experimental) data error: *e.g.* DFT is data, and IAP is model

Capturing Model Error in Likelihood (a.k.a. Data Model)

External correction (Kennedy-O'Hagan):

$$y_i = f(x_i, c) + \delta(x_i) + \sigma_i \epsilon_i$$

- Kennedy, O'Hagan, "Bayesian Calibration of Computer Models". *J Royal Stat Soc: Series B (Stat Meth),* 63: 425-464, 2001.

Internal correction (embedded model error):

$$y_i = f(x_i, c + \delta(x_i)) + \sigma_i \epsilon_i$$

- Allows meaningful usage of calibrated model
- 'Leftover' noise term even with no data error
- Respects physics (not too relevant in our context)

- Sargsyan, Najm, Ghanem, "On the Statistical Calibration of Physical Models". *Int. J. Chem. Kinet.*, 47: 246-276, 2015.
- Sargsyan, Huan, Najm, "Embedded Model Error Representation for Bayesian Model Calibration". *Int. J. Uncert. Quantif.*, 9(4): 365-394, 2019.

Embedded Model Error for Linear Regression Models

$$y_i \approx \sum_{k=0}^{P} c_k B_k(x) + \sigma_i \epsilon_i$$

Note:

No formal distinction between internal and external corrections, but internal allows for interpretation and model-informed error

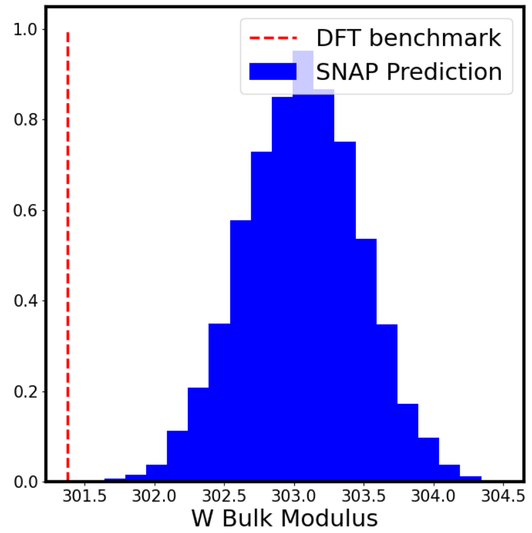
‘Embed’ uncertainty in all (or selected) coefficients

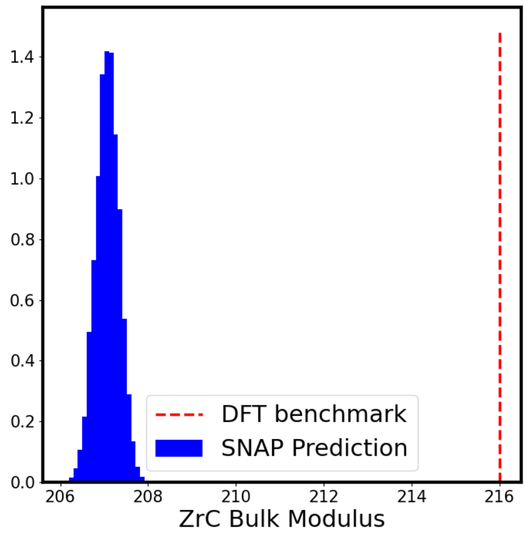
Model Model error

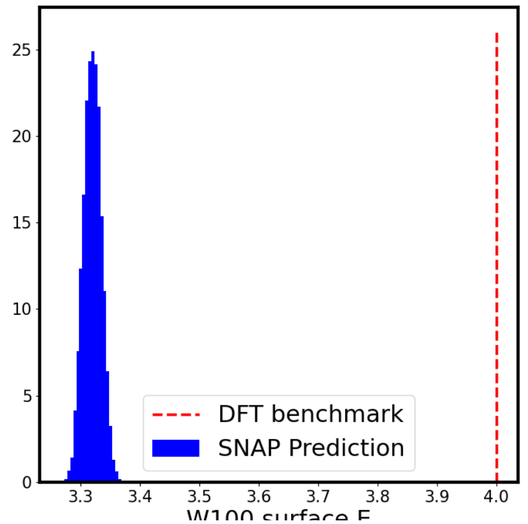
$$y_i \approx \sum_{k=0}^{P} (c_k + d_k \xi_k)\, B_k(x) = \sum_{k=0}^{P} c_k B_k(x) + \sum_{k=0}^{P} d_k B_k(x) \xi_k$$

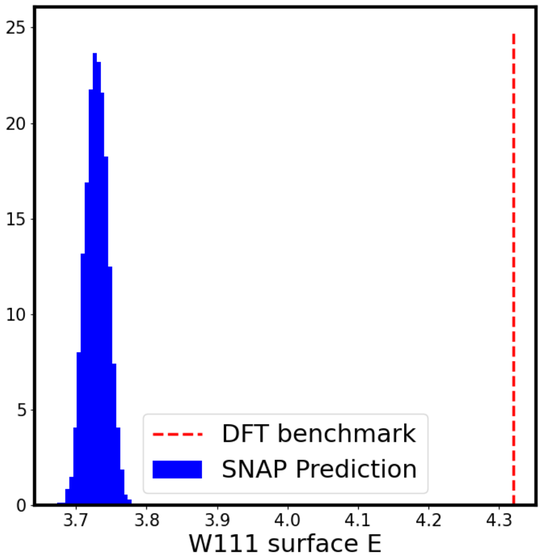
(still Gaussian, but correlated, and model-informed)

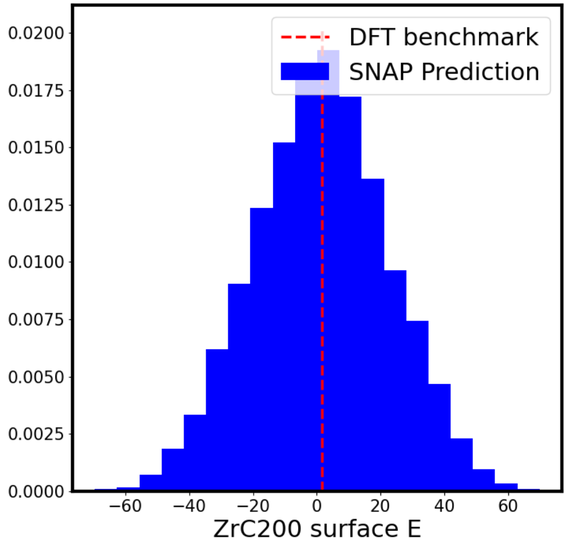
LAMMPS QoIs with uncertainty vs. DFT benchmarks

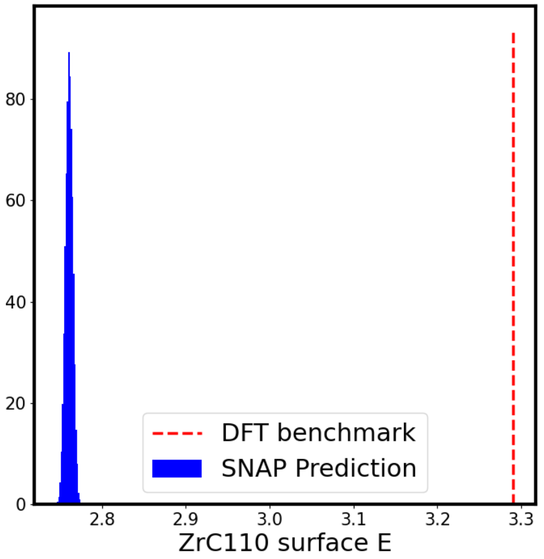












Other/future topics:

Model selection, Bayes Factor

Sparse SNAP, sparse PC

GSA details

Less relevant now: QUiNN (UQ with NN/PyTorch)