Bayesian Inference of Interatomic Potentials:

Model errors and active learning

MIT CESMIX-UQ Seminar

Mar 24, 2022

Khachik Sargsyan, Logan Williams, Katherine Johnston, Habib Najm (SNL-CA)

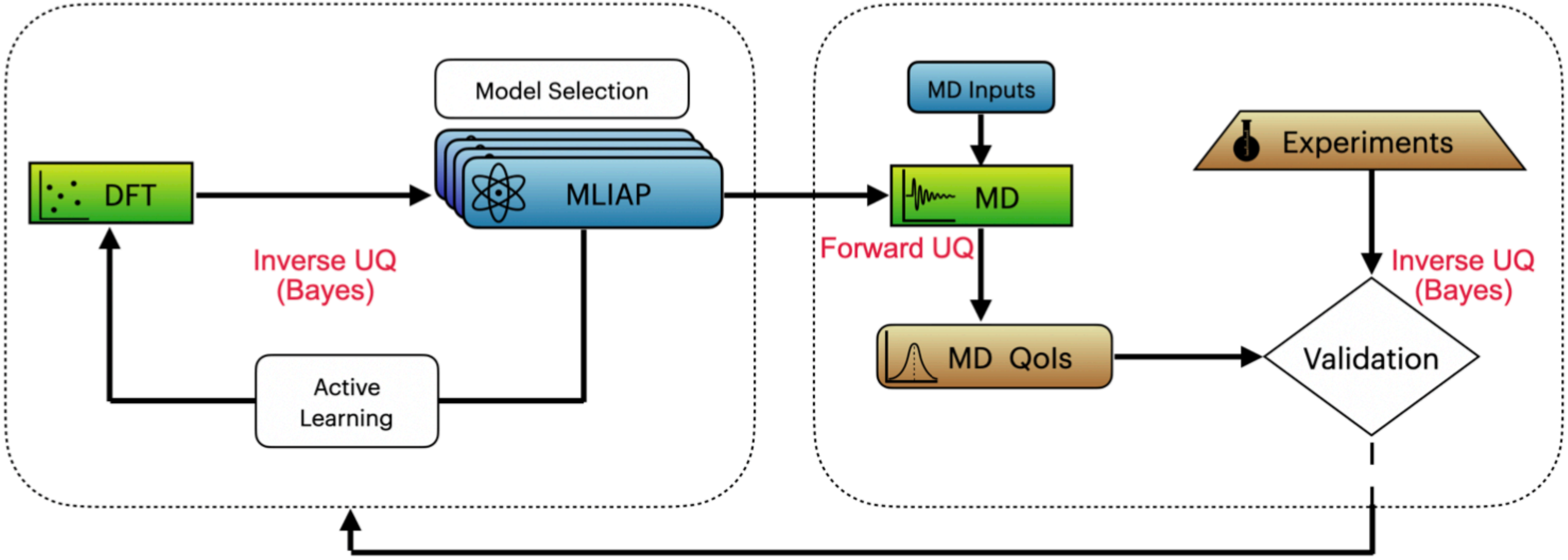
Sandia National Laboratories

FusMatML Project Overview

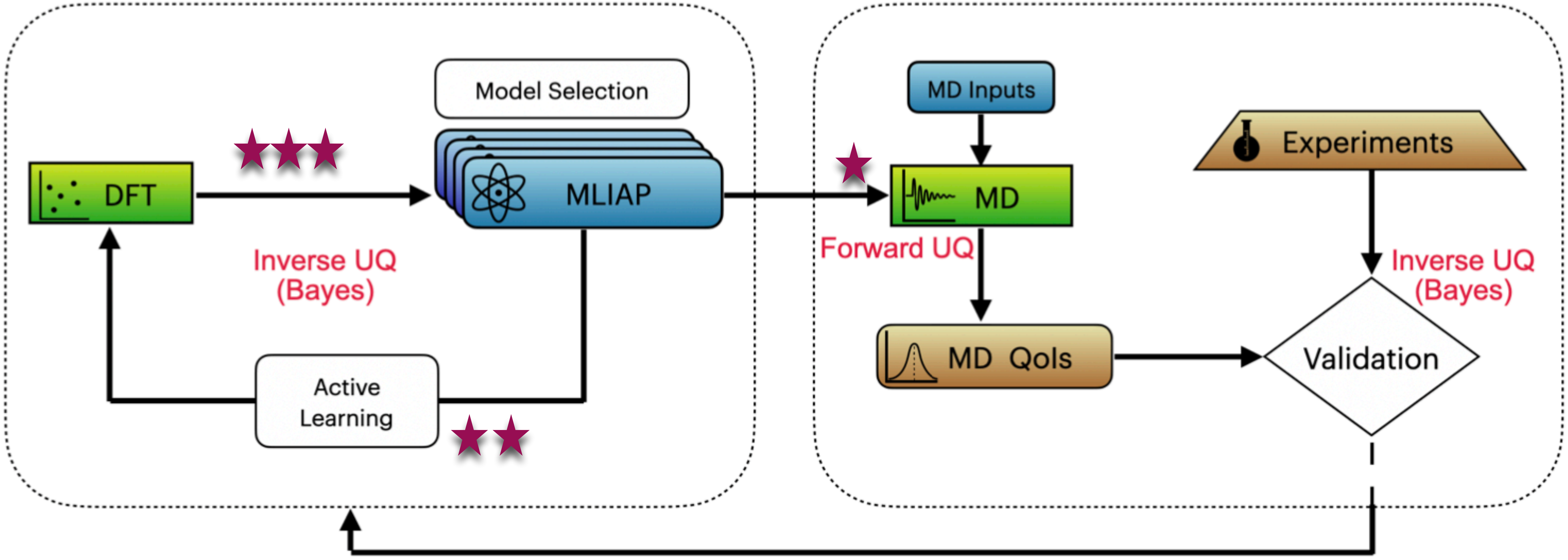
Funded by DOE ASCR / FES

- PI: Aidan Thompson (SNL-NM)
  - Advance the state-of-the-art in atomistic modeling of plasma-materials interactions connecting ab initio calculations, large-scale molecular dynamics simulations, and experimental characterization to impact fusion energy research.
  - Generate a new class of machine learning interatomic potentials (MLIAP) systematically optimized to robustly measure and control QoI uncertainty for the complex structural and chemical environments required for plasma-materials interactions
- UQ Thrust: Habib Najm, Khachik Sargsyan, Logan Williams (SNL-CA)
  - Deploy novel ML/UQ to advance the development of robust MLIAPs

Big Picture


Model Selection
DFT
MLIAP
Inverse UQ
(Bayes)
Active
Learning
MD Inputs
MD
Forward UQ
Experiments
Inverse UQ
(Bayes)
MD QoIs
Validation

Big Picture



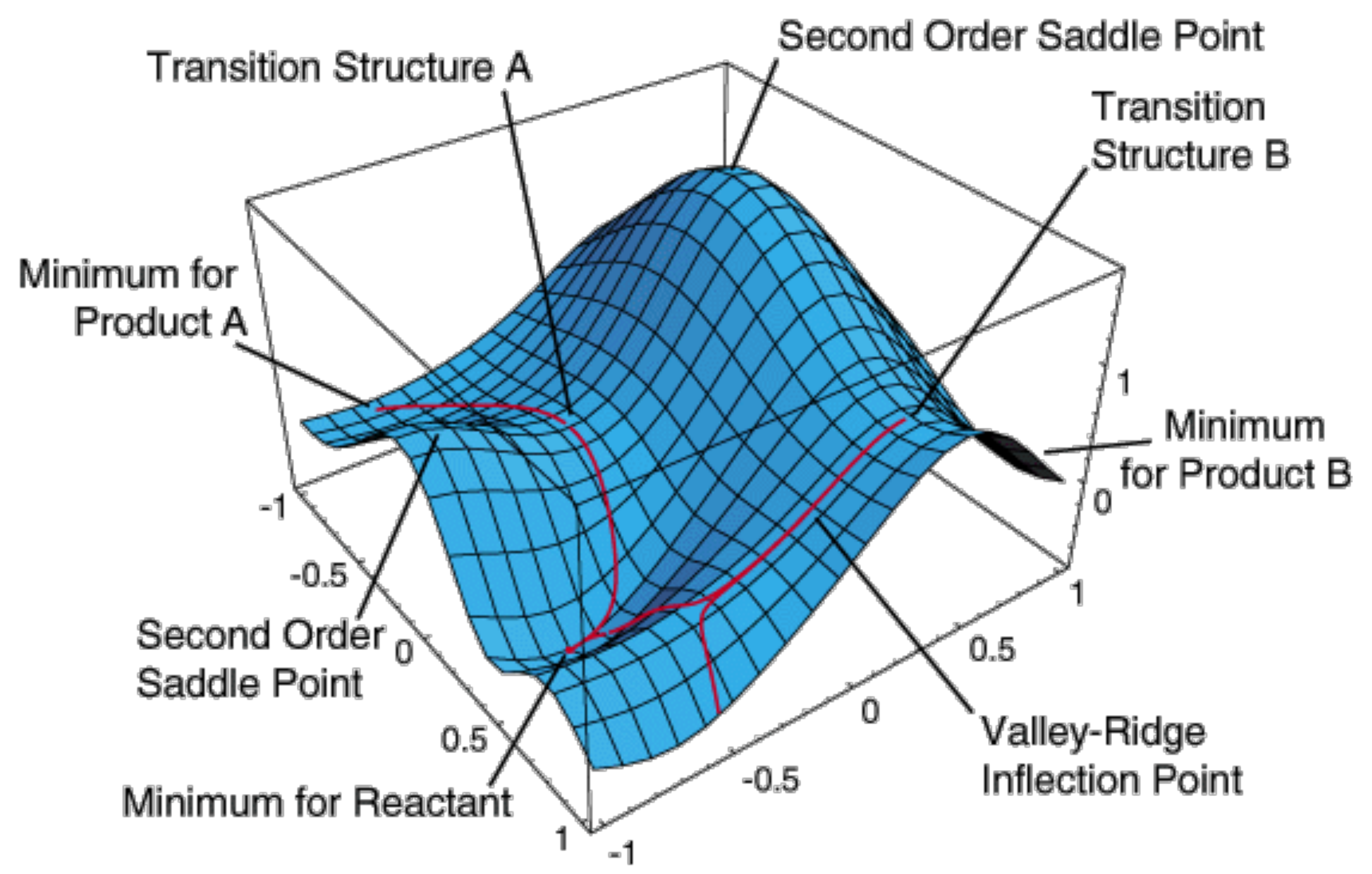
Most of this talk ★★★ Bayesian inference of IAPs, model errors

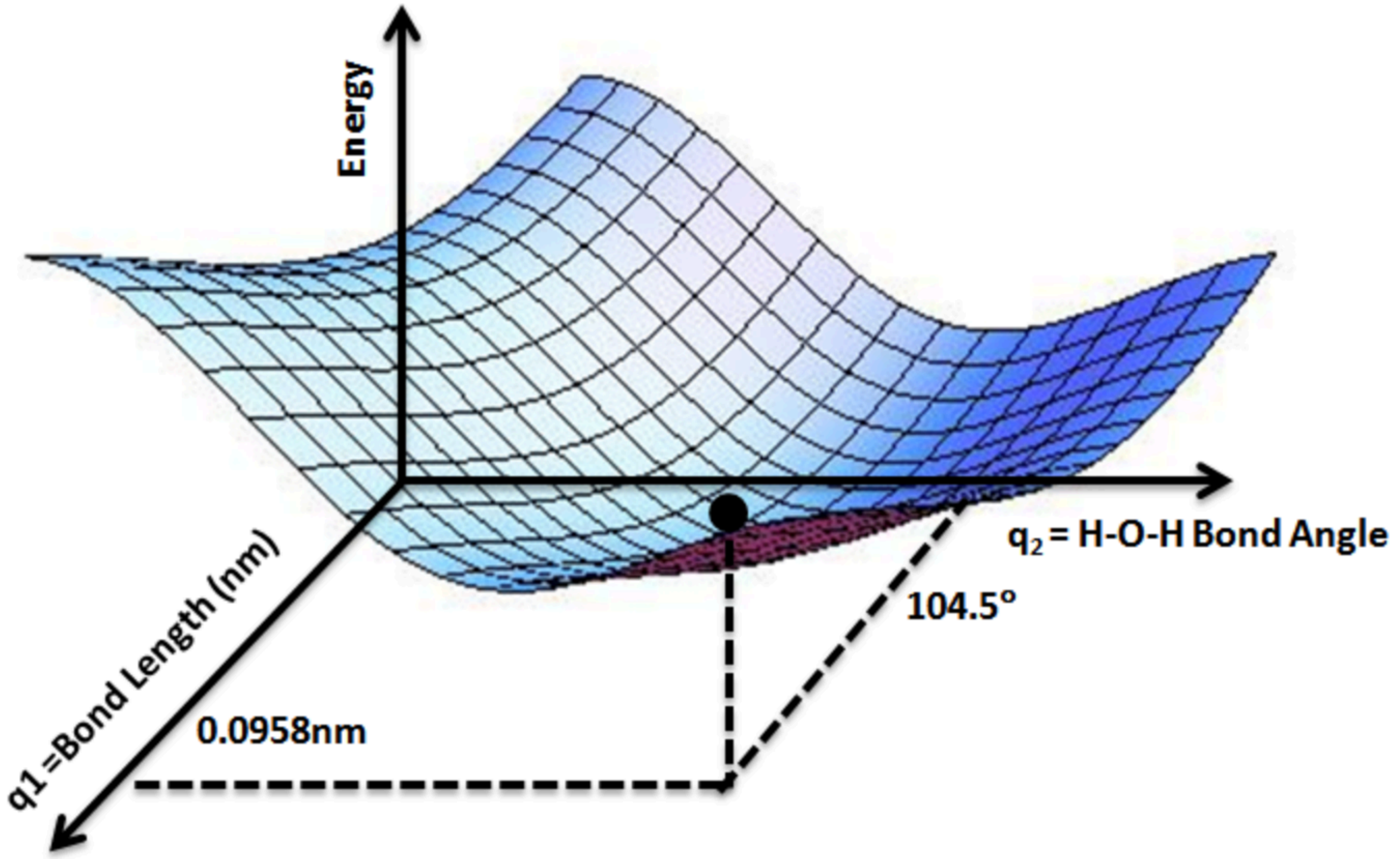
Some preliminaries ★★ Active learning

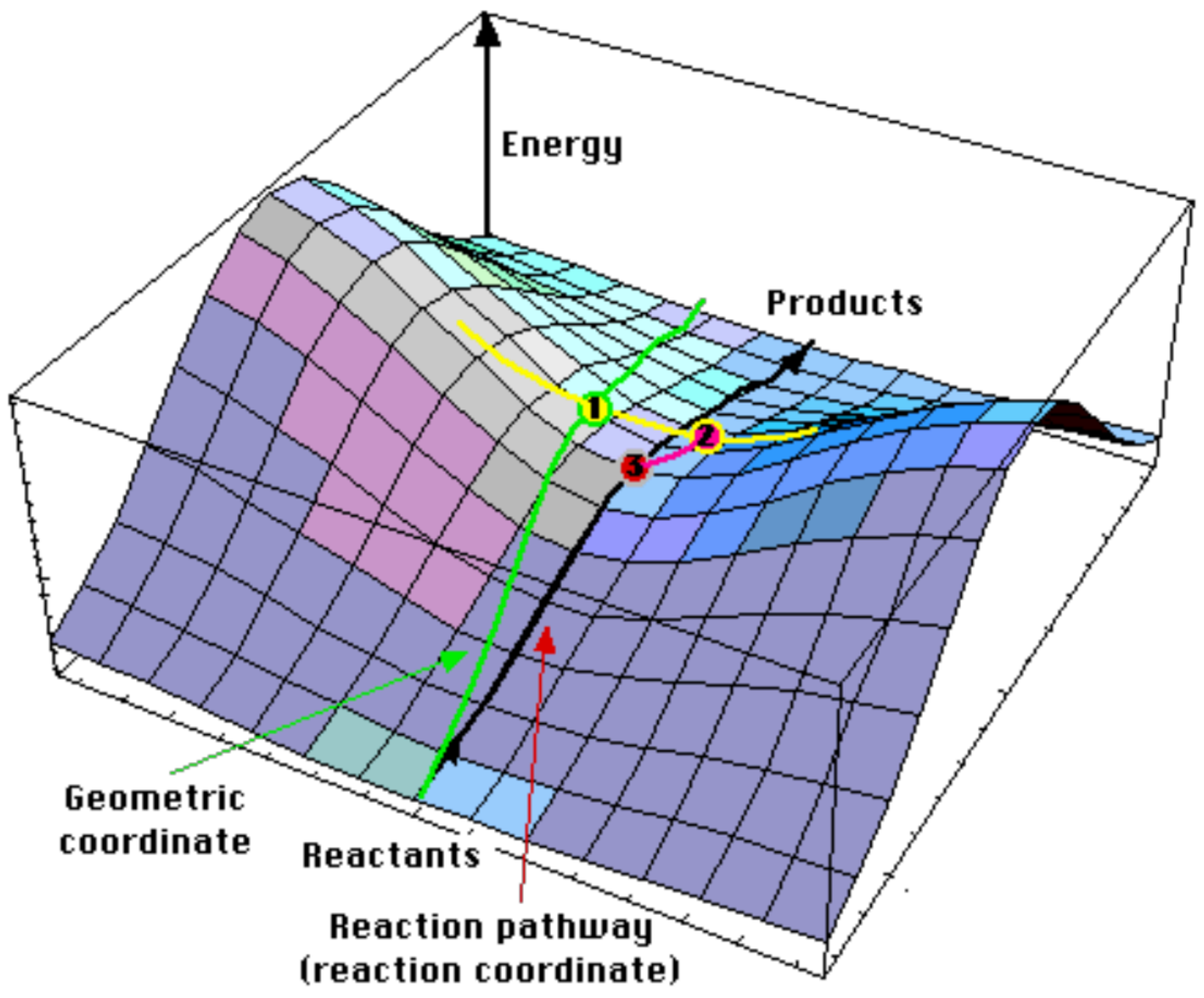
One slide ★ Propagation of IAP uncertainties through MD

Interatomic Potentials

- Object of interest: potential energy E of a system defined by a configuration x , where x encapsulates coordinates of all atoms in the system
- Typically additive form. $E(x) = E_{ref} + \sum_i E(x_i) + \ldots$ using local environments

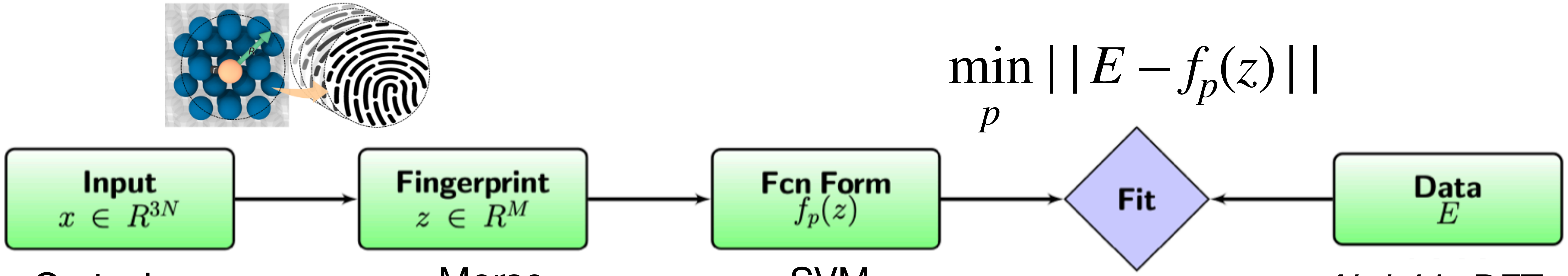






Ingredients of MLIAPs

$$\min_p ||E - f_p(z)||$$



Cartesian

Morse
Symm. Fcns
PIP
Bispectrum
ACE
…

SVM
NN
GP
LR Tensor
Lin. Regression
…

Ab initio DFT
include Force, Stress

- Training data (x_i, E_i) for $i = 1,\ldots,S$ and $x_i \in R^{3N}$
- Input representation, aka fingerprint, aka descriptor $x \rightarrow z(x)$
- Parametrized functional form of the approximation class $f_p(z)$
- Loss function: $\min_p \sum_{i=1}^{S} [E_i - f_p(z_i)]^2$ + regularization

State-of-the-art: largely manual and lacking systematic UQ

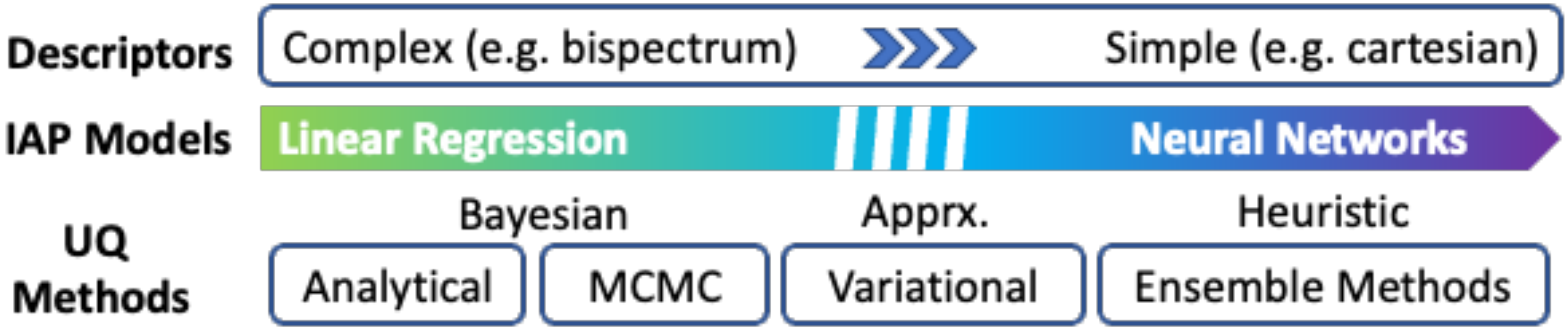
MLIAP Construction

- Good training set selection: active learning
- Fingerprint choice: invariances, symmetries
- Functional form choice: model selection
- Loss function: regularization, weighting energies and forces

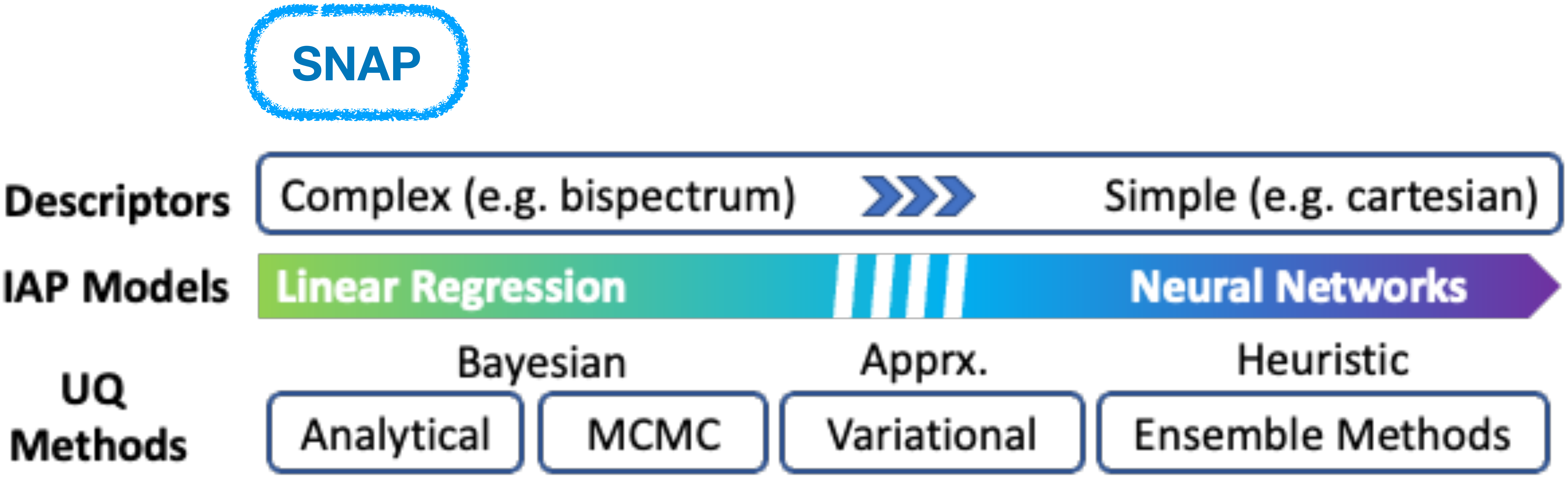
MLIAP Usage

- Find reaction pathways, saddle points
- Pipe the IAPs to MD simulations

Equipping parametric fits with uncertainties



Equipping parametric fits with uncertainties



Spectral neighbor analysis potential (SNAP) details

A.P. Thompson et al. “Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials”, *Journal of Computational Physics*, 285(15), pp. 316-330, 2015.

- Uses **bispectrum** as fingerprints:

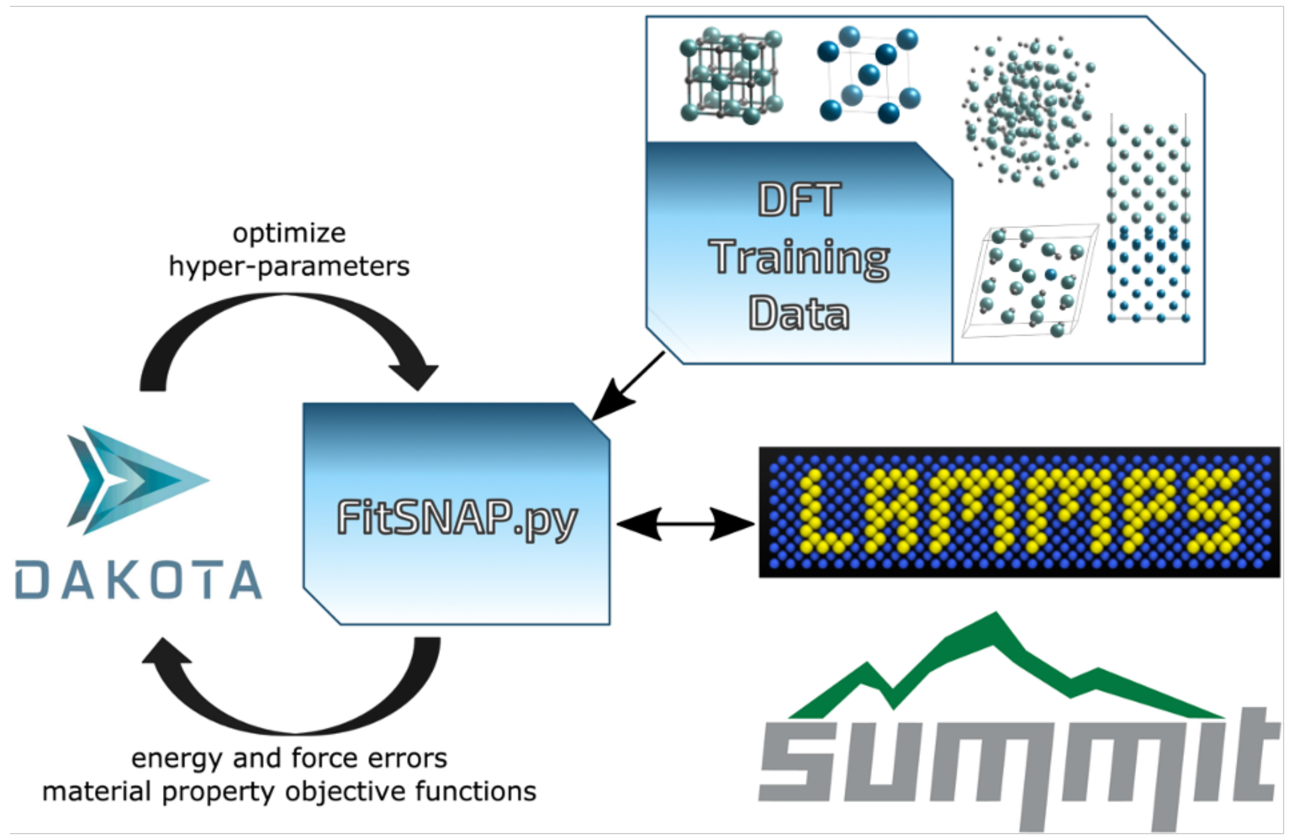
- built on hyper spherical harmonics basis functions

- respects rotational, permutational, translational invariances

- incorporates forces and stresses as well

- tunable complexity/order

$$\boxed{E(x) \approx \sum_k c_k B_k(x)}$$



- Uses **linear regression** as model form:

- built on hyper spherical harmonics basis functions

- generalized to quadratic form as well

M. Wood and A. Thompson , “Extending the accuracy of the SNAP interatomic potential form”, *Journal of Chemical Physics*, 148, 2018.

(Bayesian) Parameter Inference

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

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

Posterior PDF   Prior PDF   Model   Data

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

Likelihood

- ✦ Prior contains previous knowledge or regularization
- ✦ Likelihood contains data noise modeling assumptions,

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

Data Model

Linear Models: luxury of closed-form posteriors

$$f(x, c) = Ac$$

- Gaussian likelihood with fixed σ

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

- Leads to Gaussian posterior PDF

$$p(c \,|\, y) \propto p(y \,|\, c) p(c) \sim \mathcal{N}\left((A^T A)^{-1} A^T y, \sigma^2 (A^T A)^{-1} \right)$$

- As well as Gaussian push-forward and posterior predictive

Linear Models: unknown σ

- ✦ Gaussian likelihood with inferred σ

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

- ✦ Leads to normal-inverse-gamma posterior PDF

$$p(c,\sigma^2\,|\,y) \propto p(y\,|\,c,\sigma^2)p(c,\sigma^2) \sim NIG$$

- ✦ … but only its marginals are interesting/useful

$$p(c\,|\,y,\sigma^2) \sim \mathcal{N}\left((A^TA)^{-1}A^Ty, \sigma^2(A^TA)^{-1}\right)$$

$$p(\sigma^2\,|\,y) \sim IG\left(\frac{N-K}{2}, \frac{N-K}{2}\hat{\sigma}^2\right)$$

$$p(c\,|\,y) \sim St\left((A^TA)^{-1}A^Ty, \sigma^2(A^TA)^{-1}, N-K\right)$$

Effective stdv. = residual RMSE

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 $\neq$ 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

Posterior uncertainty does not capture true discrepancy

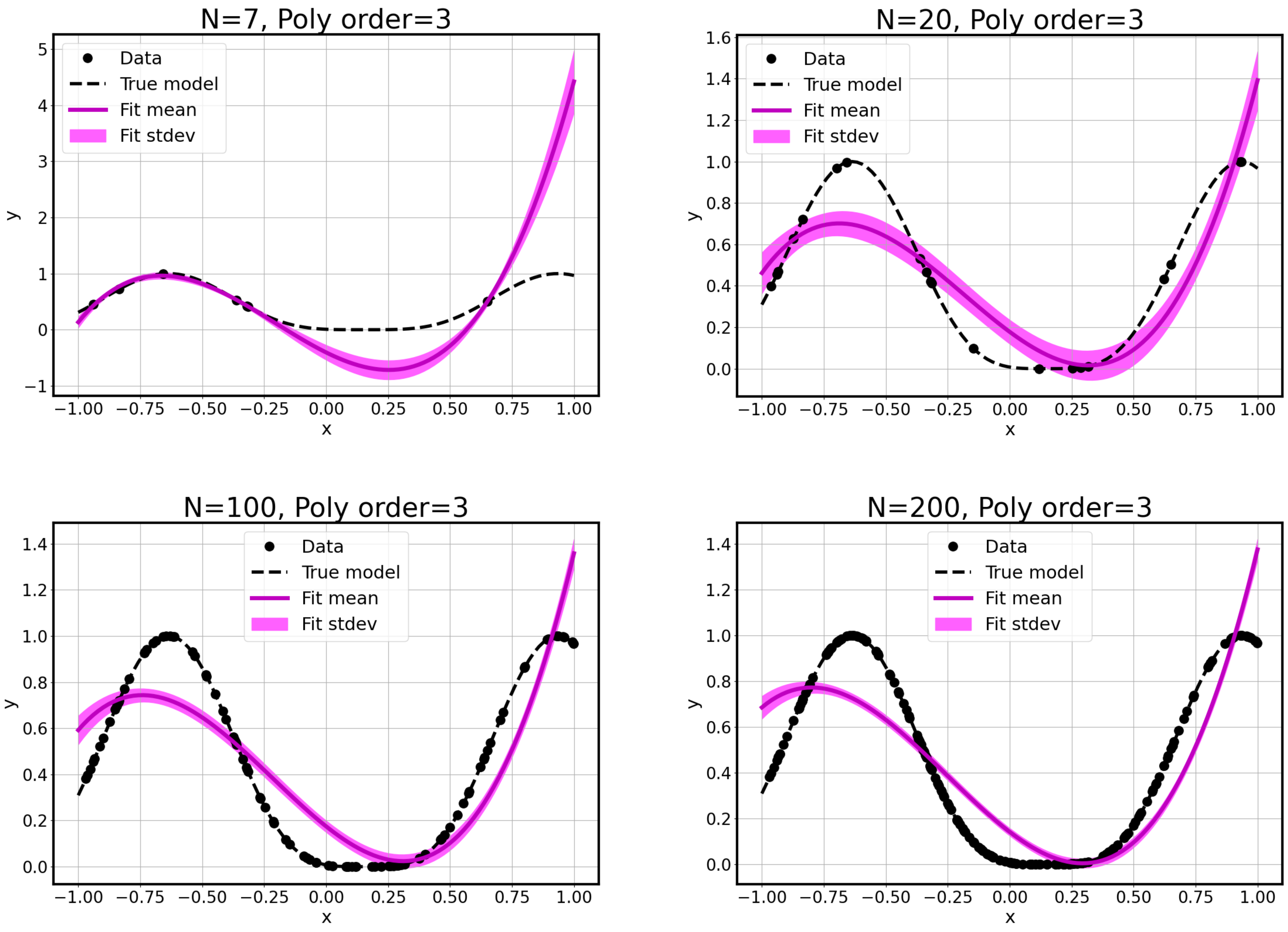
Synthetic data

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

Cubic fit

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

More data leads to overconfident prediction



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

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

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

$$y_i \approx \sum_{k=0}^{P} \left(c_k + d_k \xi_k\right) B_k(x) \quad = \quad \underbrace{\sum_{k=0}^{P} c_k B_k(x)}_{\text{Model}} + \underbrace{\sum_{k=0}^{P} d_k B_k(x) \xi_k}_{\text{Model error}}$$

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

Embedded Model Error: likelihood options

Classical data model

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

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

MCMC sampling of c

Embedded model error

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

Option 1 (IID)

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

MCMC sampling of c, d

or

simply optimize the posterior for c, d

Embedded Model Error: likelihood options

Classical data model

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

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

MCMC sampling of c

Embedded model error

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

Option 2 (ABC)

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

Pushed forward predictive uncertainty captures the true discrepancy from the data

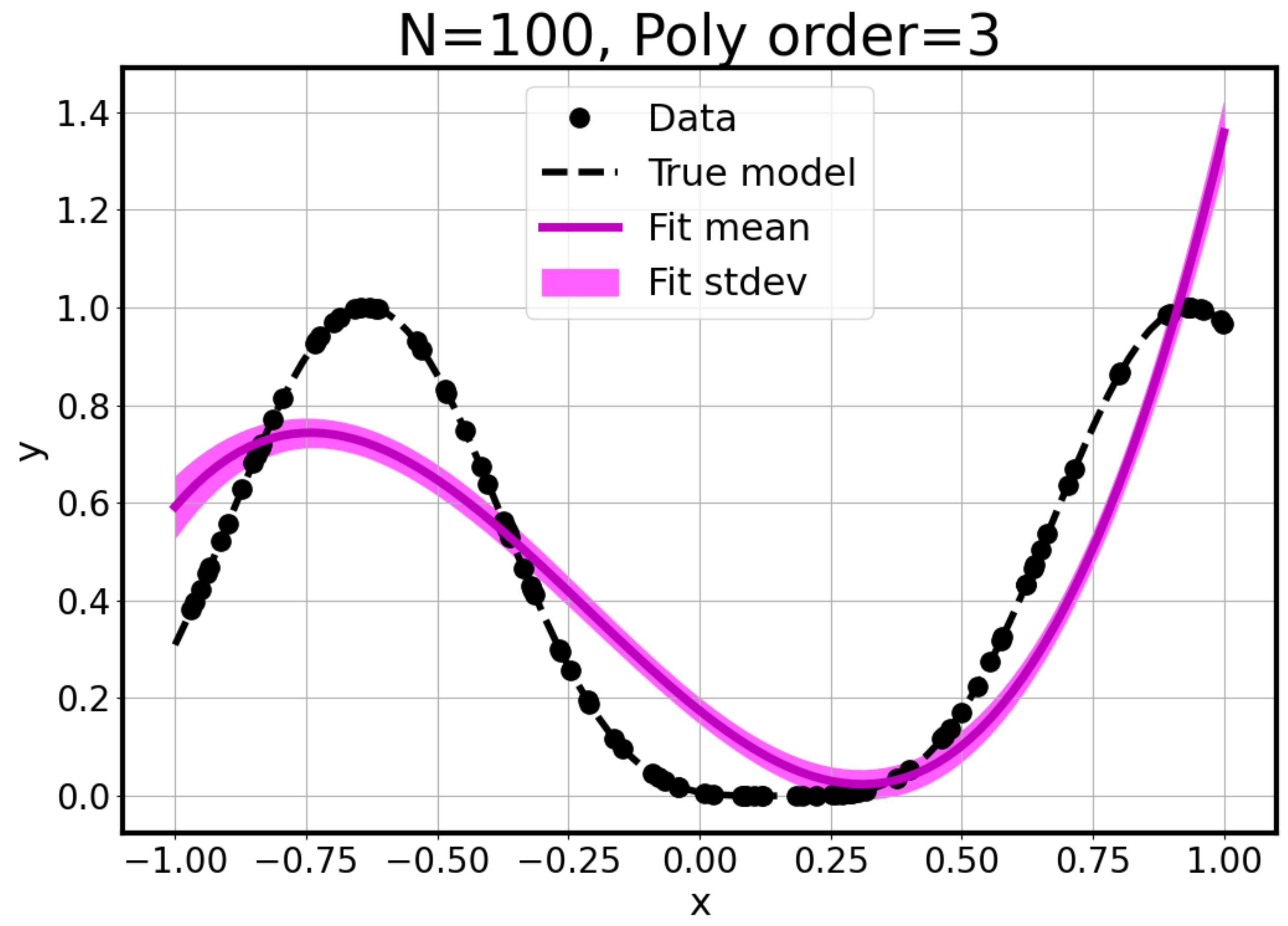
Synthetic data

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

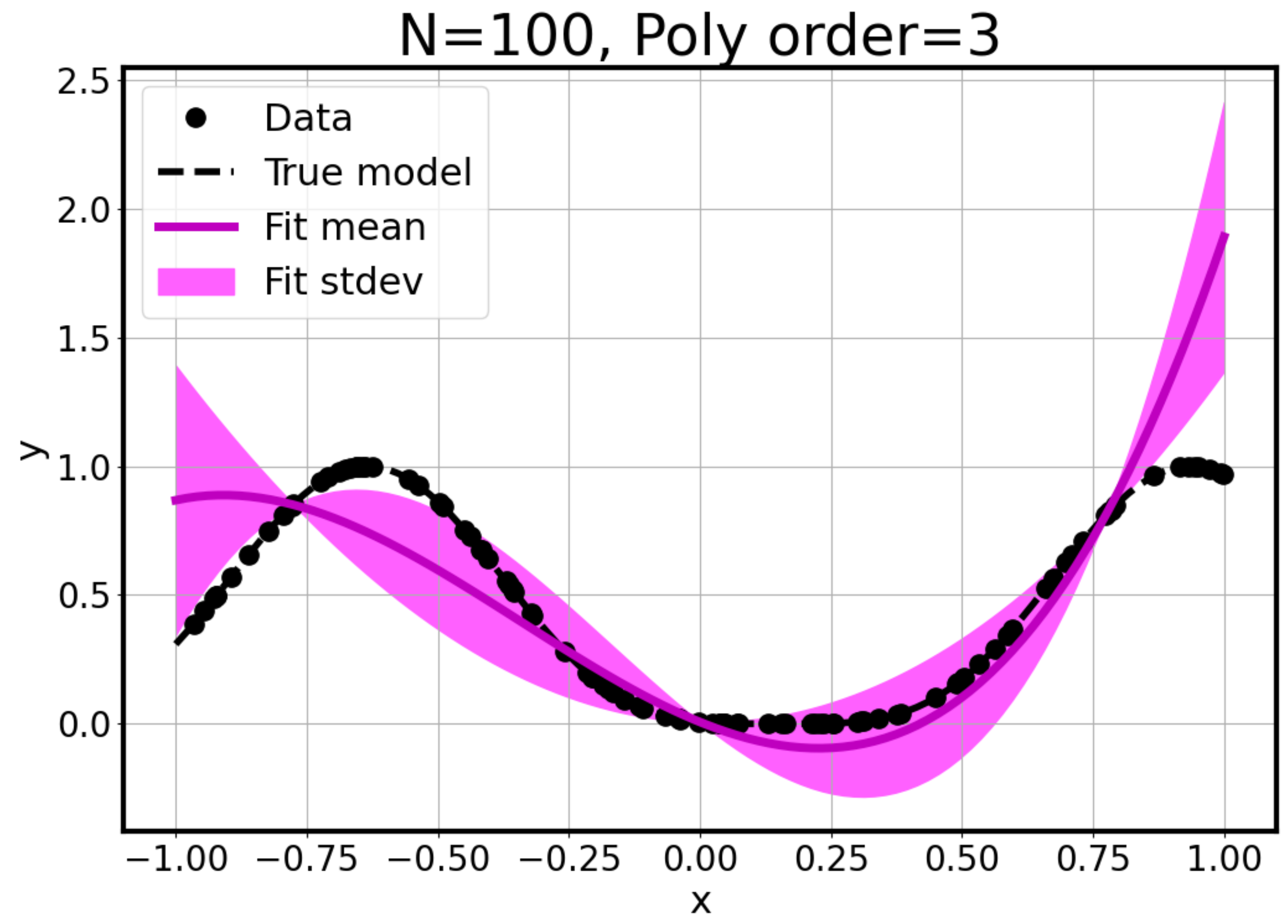
Cubic fit

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

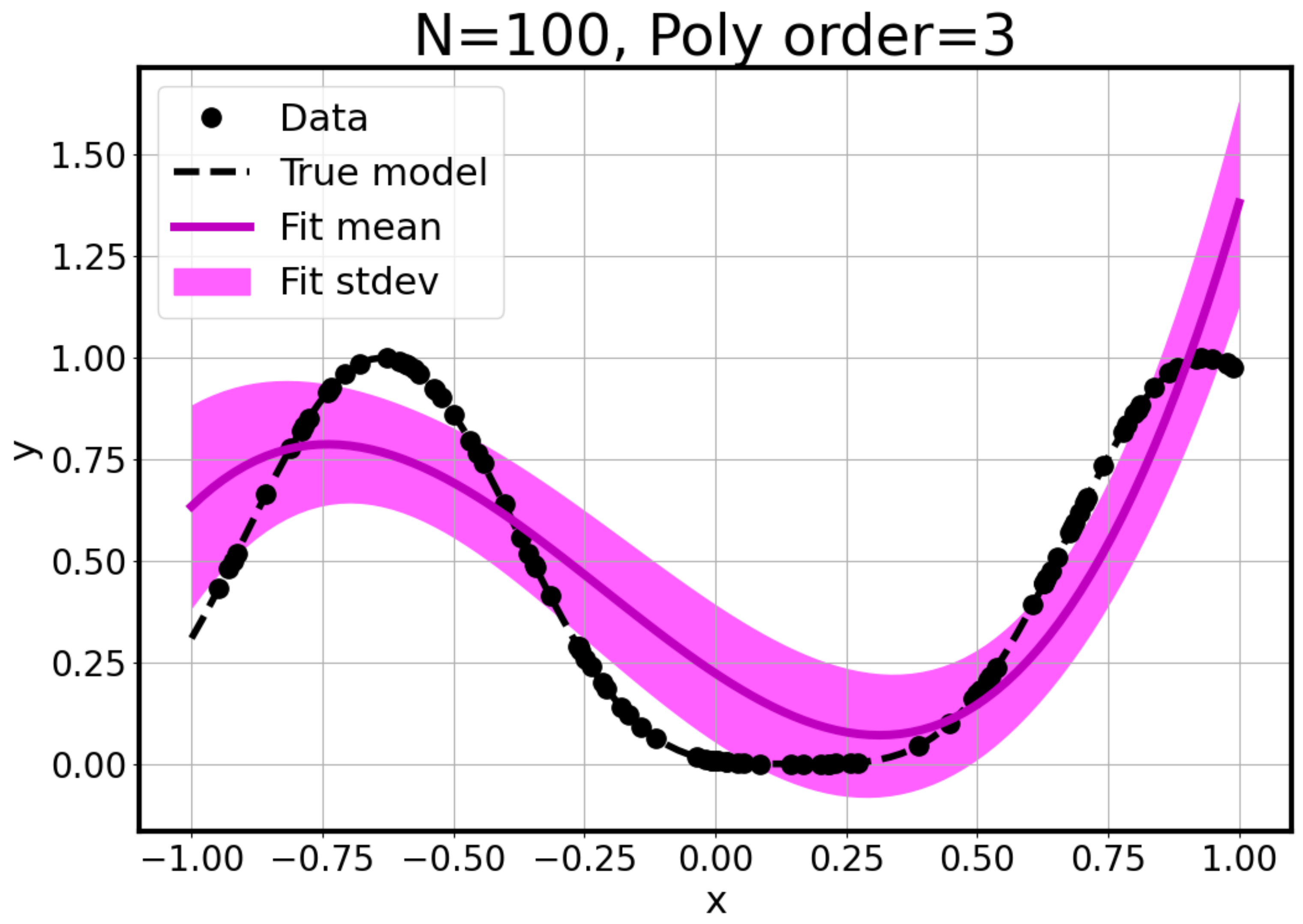
Classical case



Model error, IID likelihood

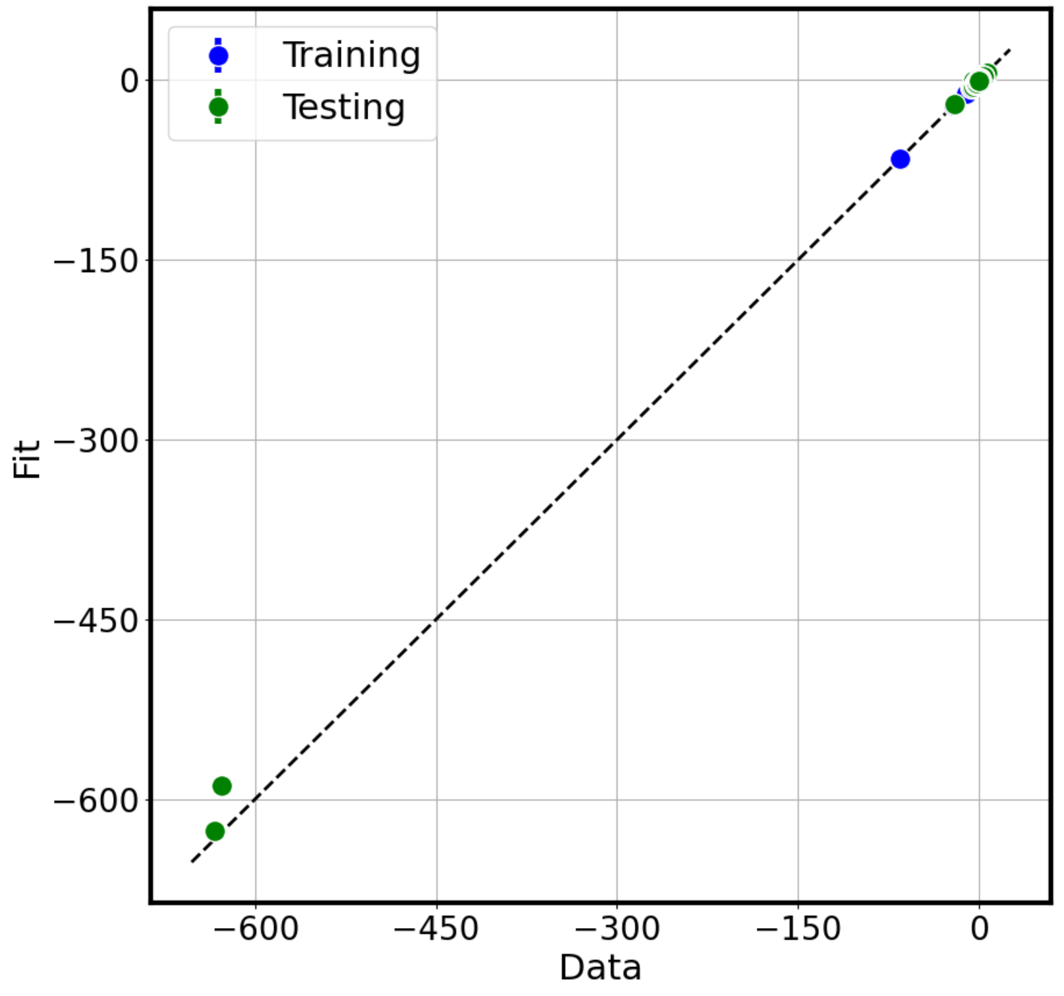


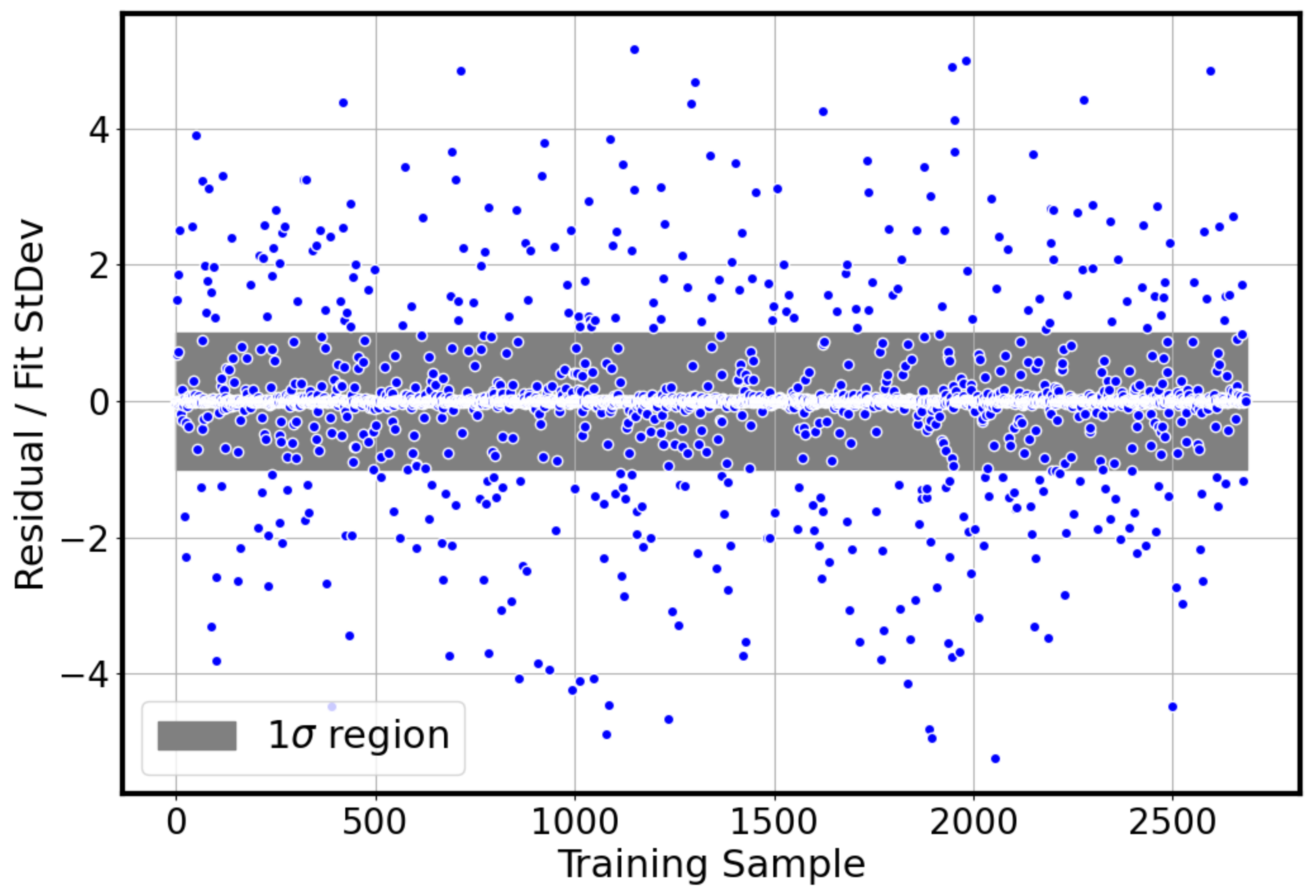
Model error, ABC likelihood

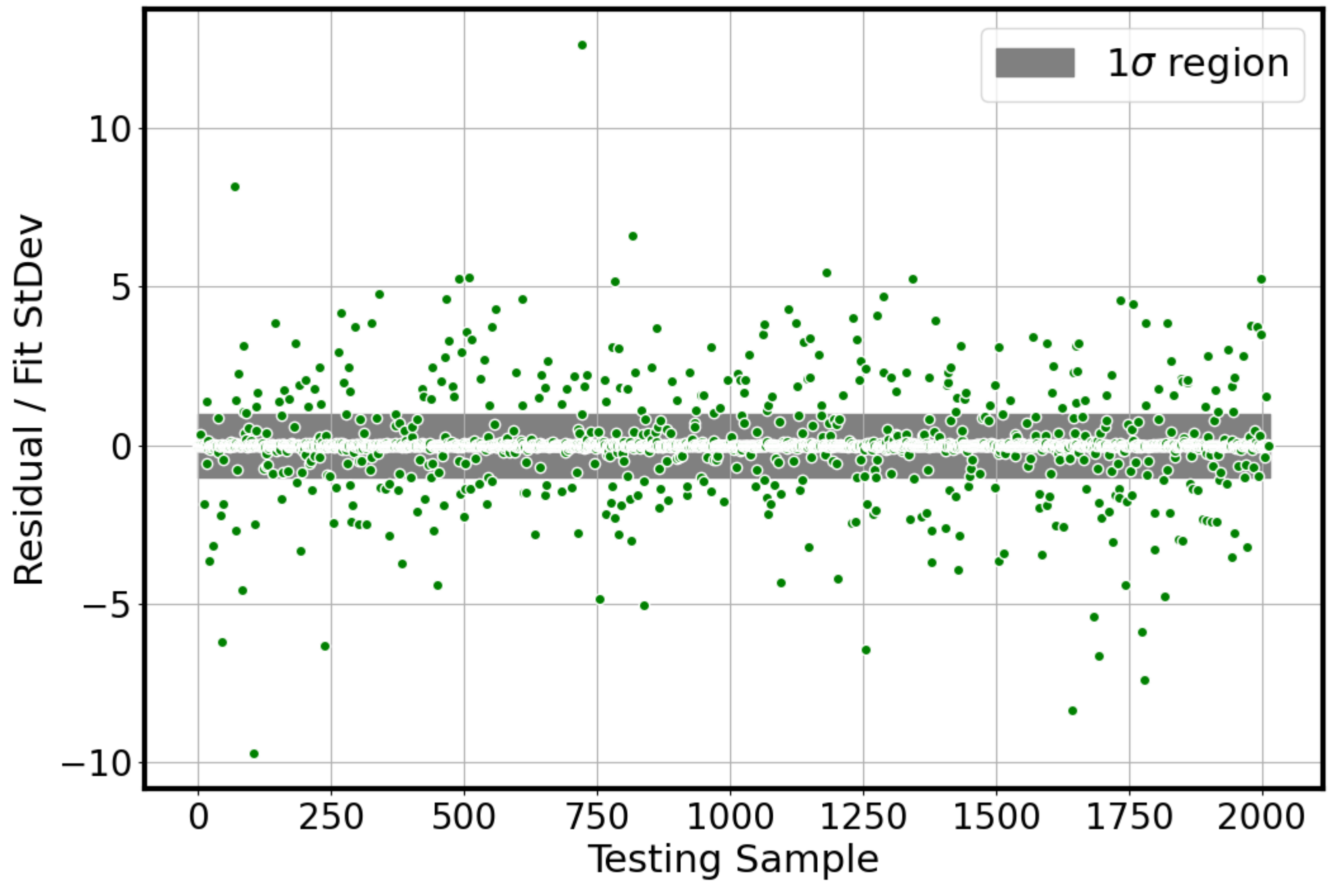


W-ZrC Dataset

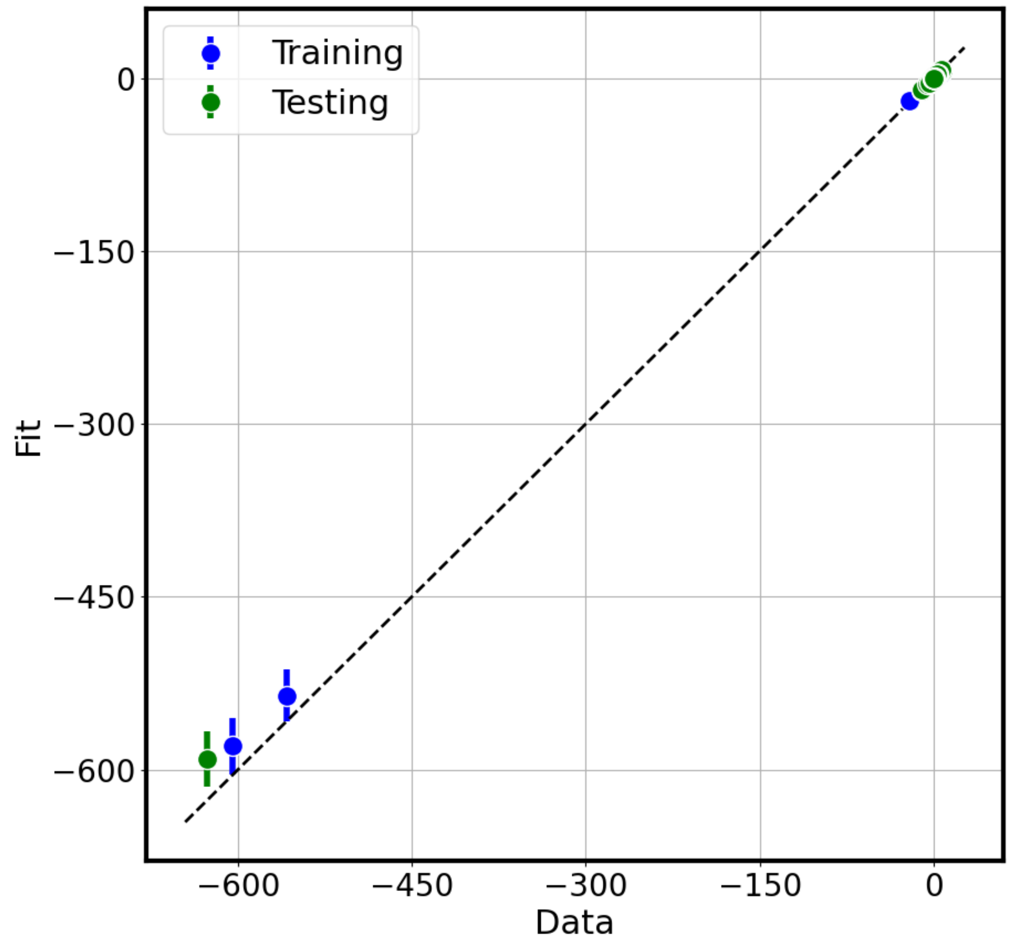
Uncertainty without model error

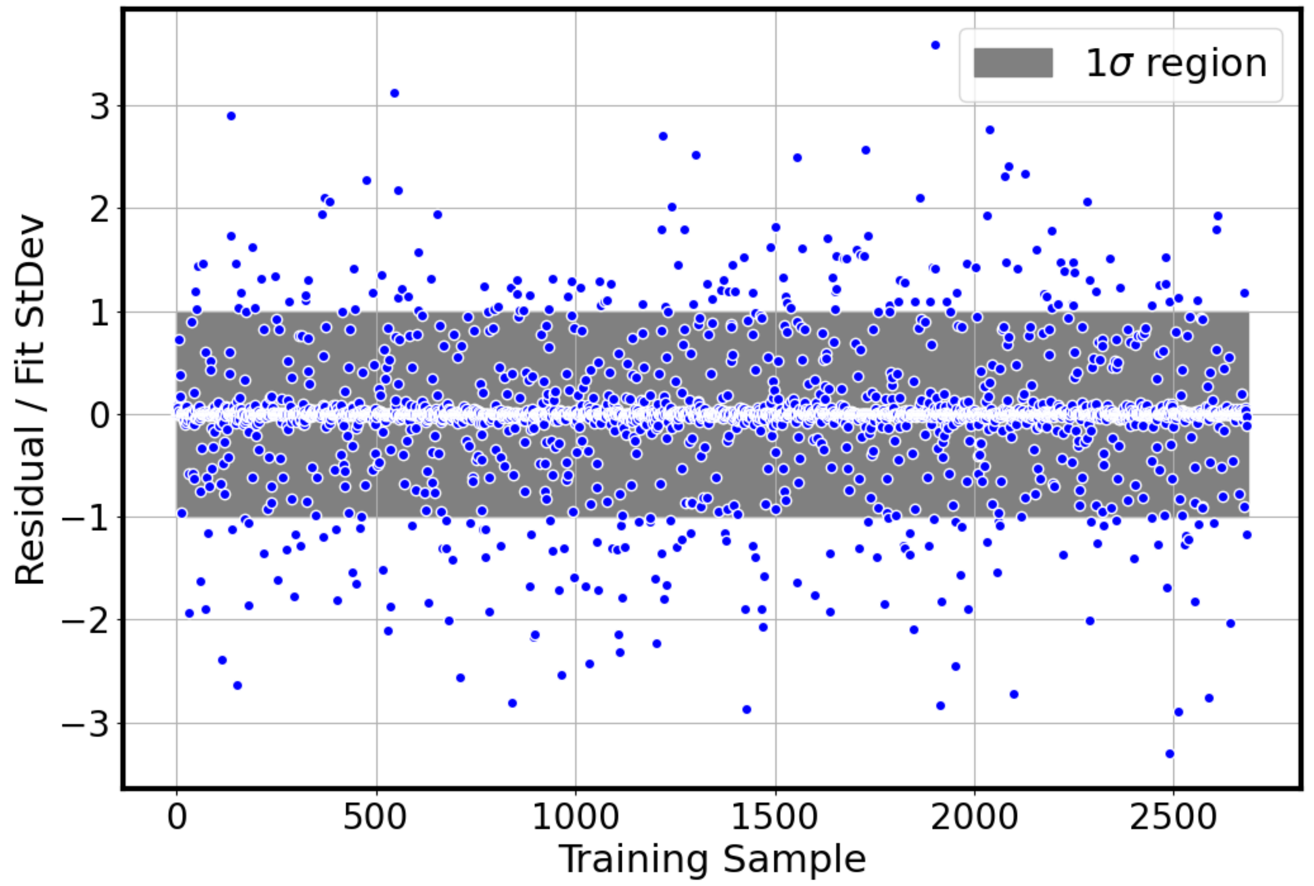


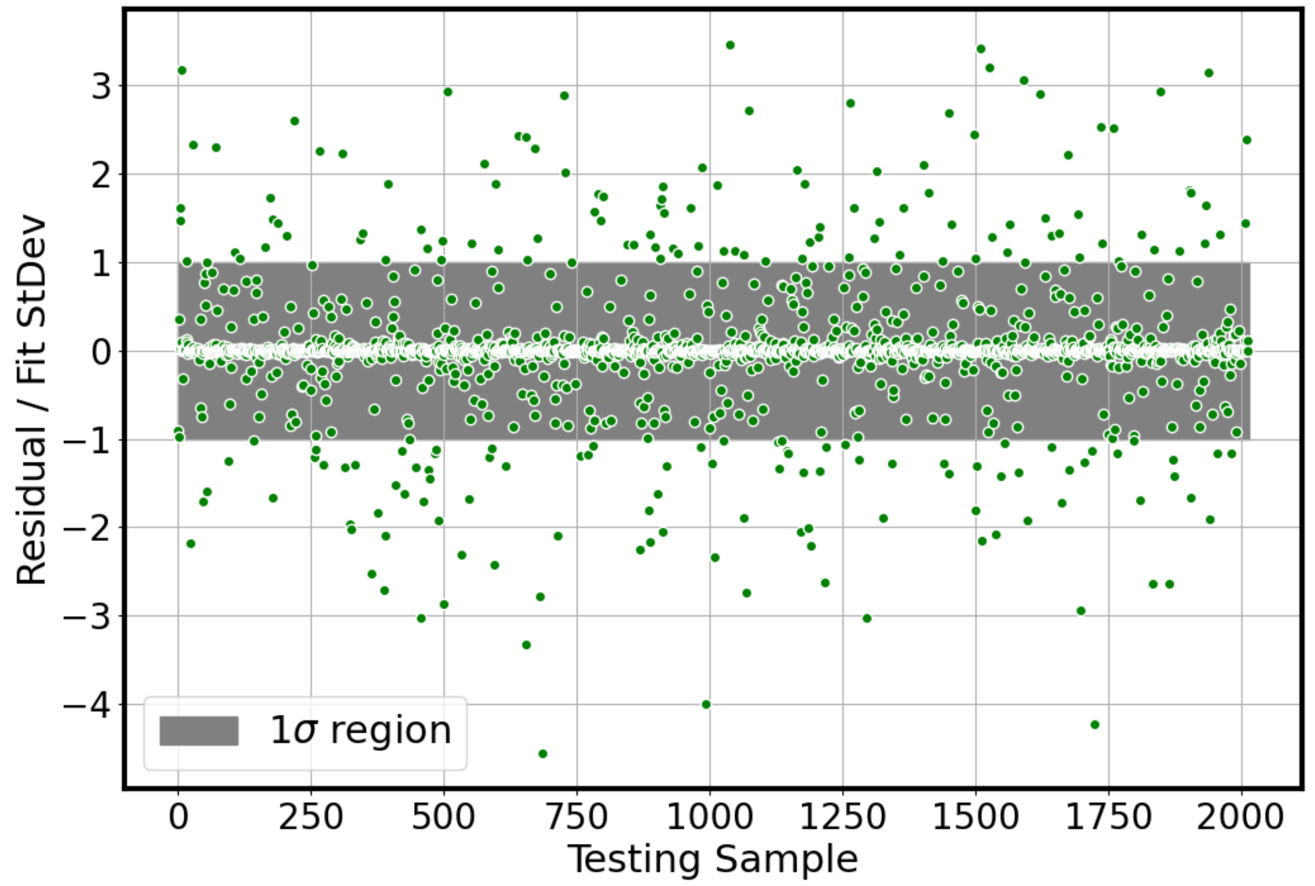




Uncertainty with model error







Several challenges/choices

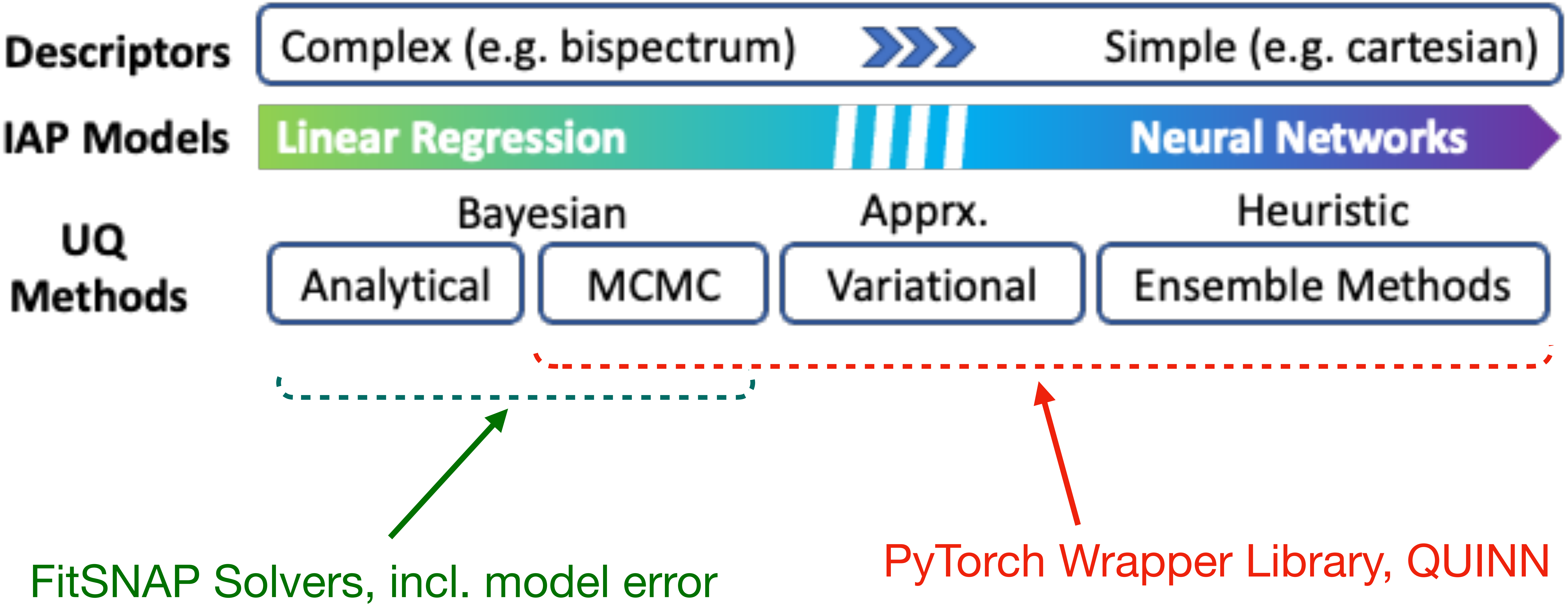
- Embedding type: *e.g.*

<u>additive</u> $y_i \approx \sum_{k=0}^{P} (c_k + d_k\xi_k)\, B_k(x)$ or <u>multiplicative</u> $y_i \approx \sum_{k=0}^{P} (c_k + c_k d_k \xi_k)\, B_k(x)$

- 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?
- Connect predictive uncertainty and the residual error with an extrapolation metric
- Weighting between energies, forces and stresses
- Major challenge: data sizes are large, linear algebra chokes

Equipping parametric fits with uncertainties



QUINN: PyTorch Wrappers for UQ

Deterministic → **Probabilistic**

torch.nn.module → **wrapper(torch.nn.module)**

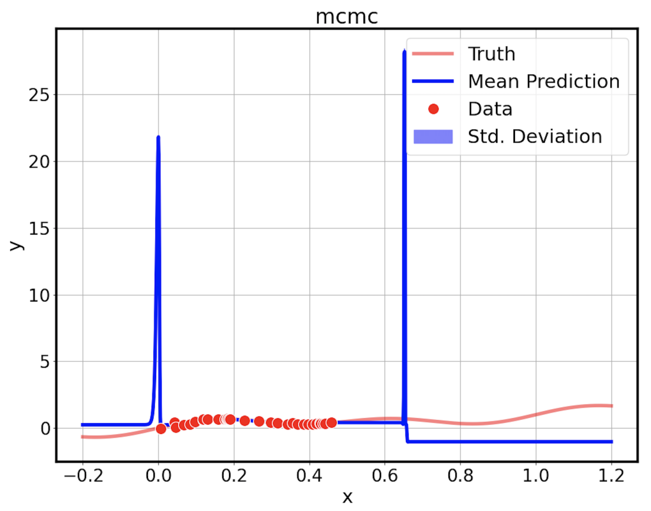
Usage:

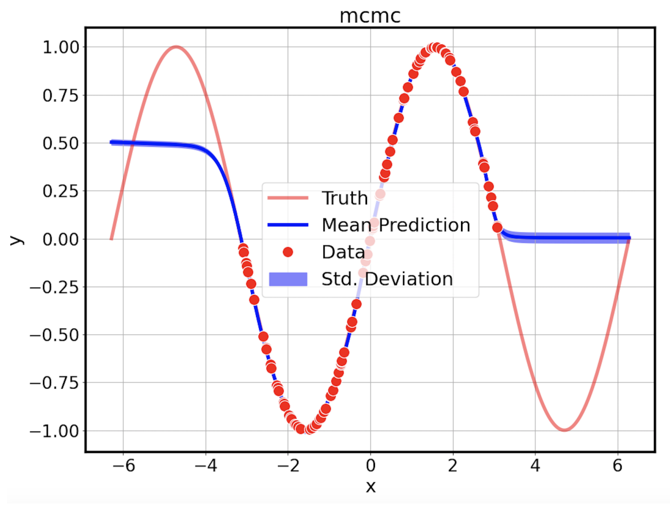
```python
uqnet = MCMC_NN(nnet)
```

```python
class MCMC_NN(QUiNNBase):
    def __init__(self, nnmodule, verbose=True):
        super(MCMC_NN, self).__init__(nnmodule)
        self.verbose = verbose
```

Option 1: MCMC

- The right thing to do, but extremely challenging
- Practically unusable for complex models



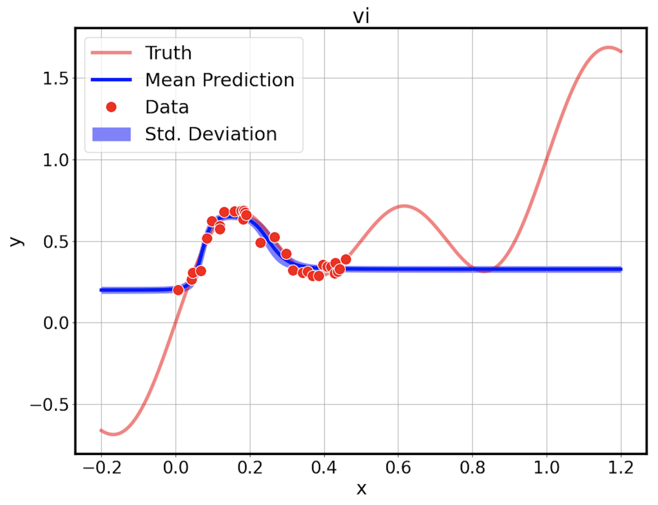


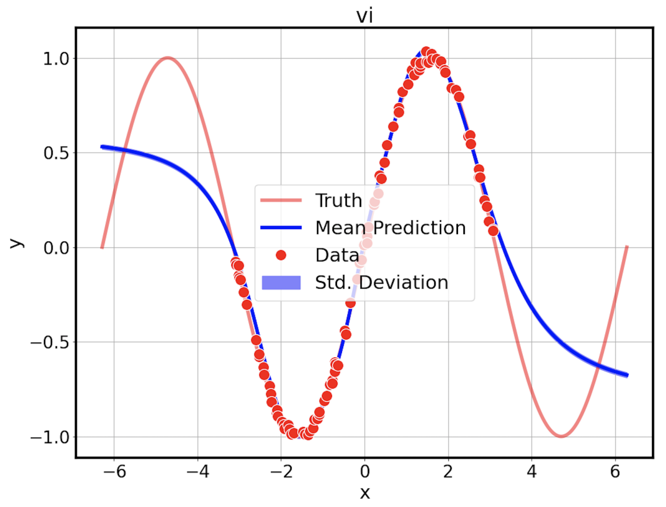

```python
uqnet = VI_NN(nnet)
```

```python
class VI_NN(QUiNNBase):
    def __init__(self, nnmodule, verbose=False):
        super(VI_NN, self).__init__(nnmodule)
        self.bmodel = BNet(nnmodule)
        self.verbose = verbose
```

Option 2: Variational Inference

- Practically feasible
- Many hyperparameters to tune
- Does not represent extrapolative uncertainties well



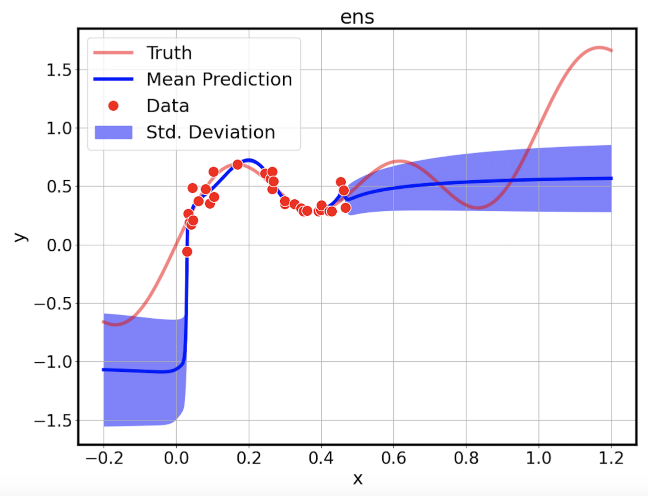


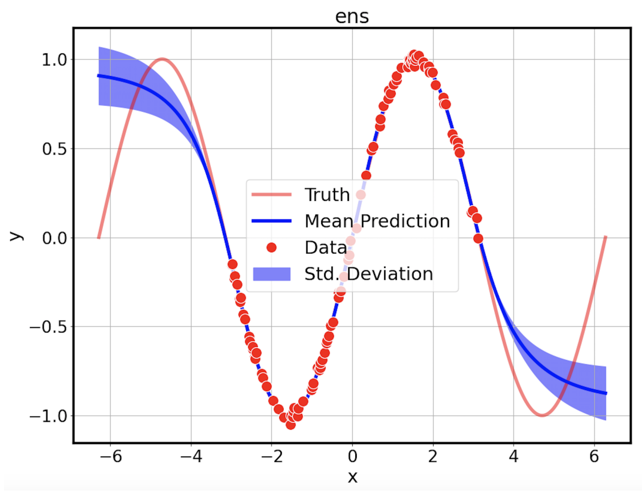

```python
uqnet = Ens_NN(nnet, nens=nmc)
```

```python
class Ens_NN(QUiNNBase):
    def __init__(self, nnmodule, nens=1, verbose=False):
        super(Ens_NN, self).__init__(nnmodule)
        self.verbose = verbose
        self.nens = nens
```

Option 3: Ensembling

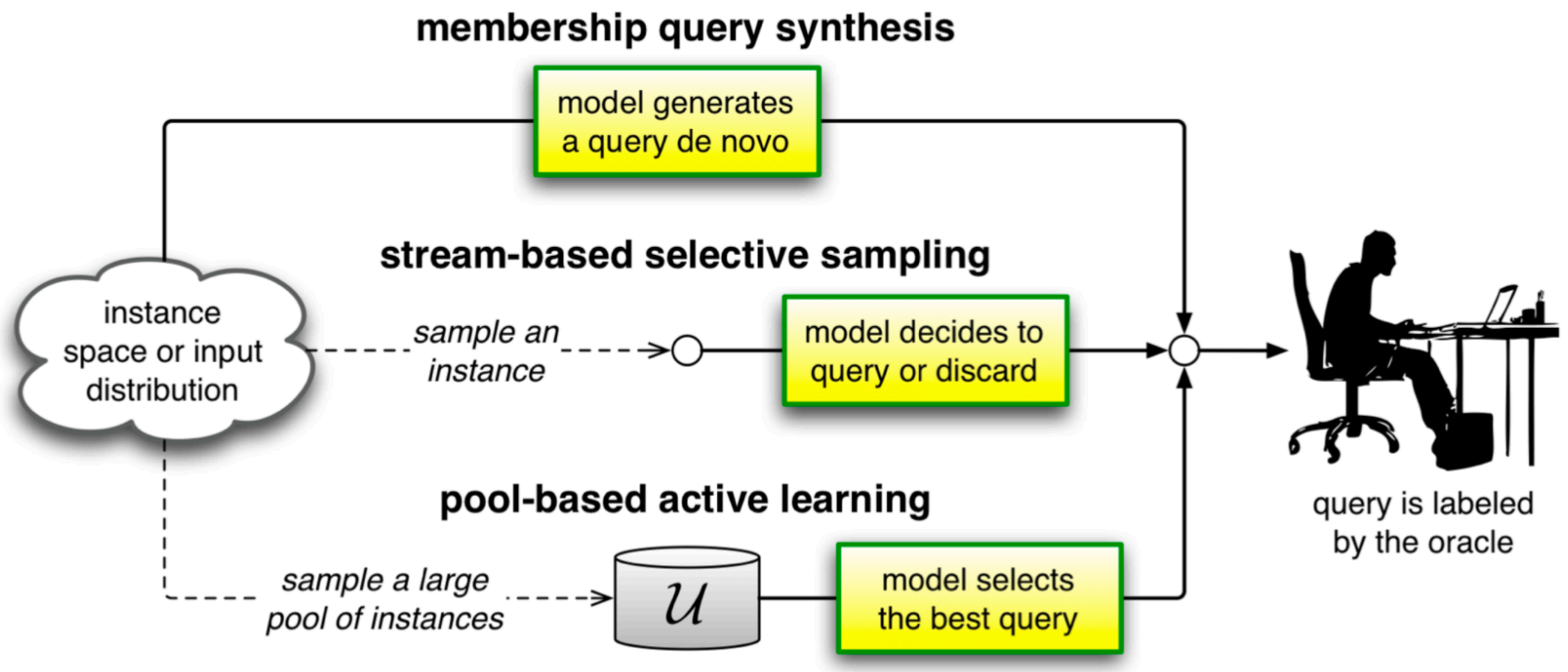
- Heuristic, unfortunately….
- … but works best for complex models
- Query-by-Committee (QBC)





Active Learning: motivation

- Choose the training samples adaptively
- Achieve greater accuracy with fewer training samples
- In conventional ML, minimize human effort of labeling images
- For us, minimize the number of *ab initio* QM calculations
- (aka optimal experimental/computational design)



Detect and query extrapolative (high-uncertainty?) configurations on-the-fly and get QM data for those.

Key: query strategy, whether to query QM or not. If such decision can be made reliably, then one does not need to start with a very good training set.

[B. Settles, "Active learning literature survey", Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009]

Active Learning: query strategies

Uncertainty sampling: an active learner queries the instances about which it is least certain how to label. Straightforward for probabilistic models.

Query-by-committee: committee of competing models, that are consistent with the current training set. The most informative query is considered to be the instance about which they most disagree. Key is to have a meaningful set of models. Need a measure of disagreement. Again, Bayesian/probabilistic is the best bet, but there are also non-probabilistic methods such as query-by-boosting and query-by-bagging.

Expected model change: which query would lead to greatest model change, e.g. largest gradient length.

Variance Reduction and Fisher Information Ratio: in regression setting, minimizing the variance component of generalization error (usually some sort of approximation or via Fisher).

Estimated error reduction: Estimate the expected future error that would result if some new instance x is labeled and added to training set, and then select the instance that minimizes that expectation. Naively retrain with all potential new points. Practical if incremental training is possible, e.g. GP, or linear MLIP such as in this paper.

Active Learning: optimality conditions

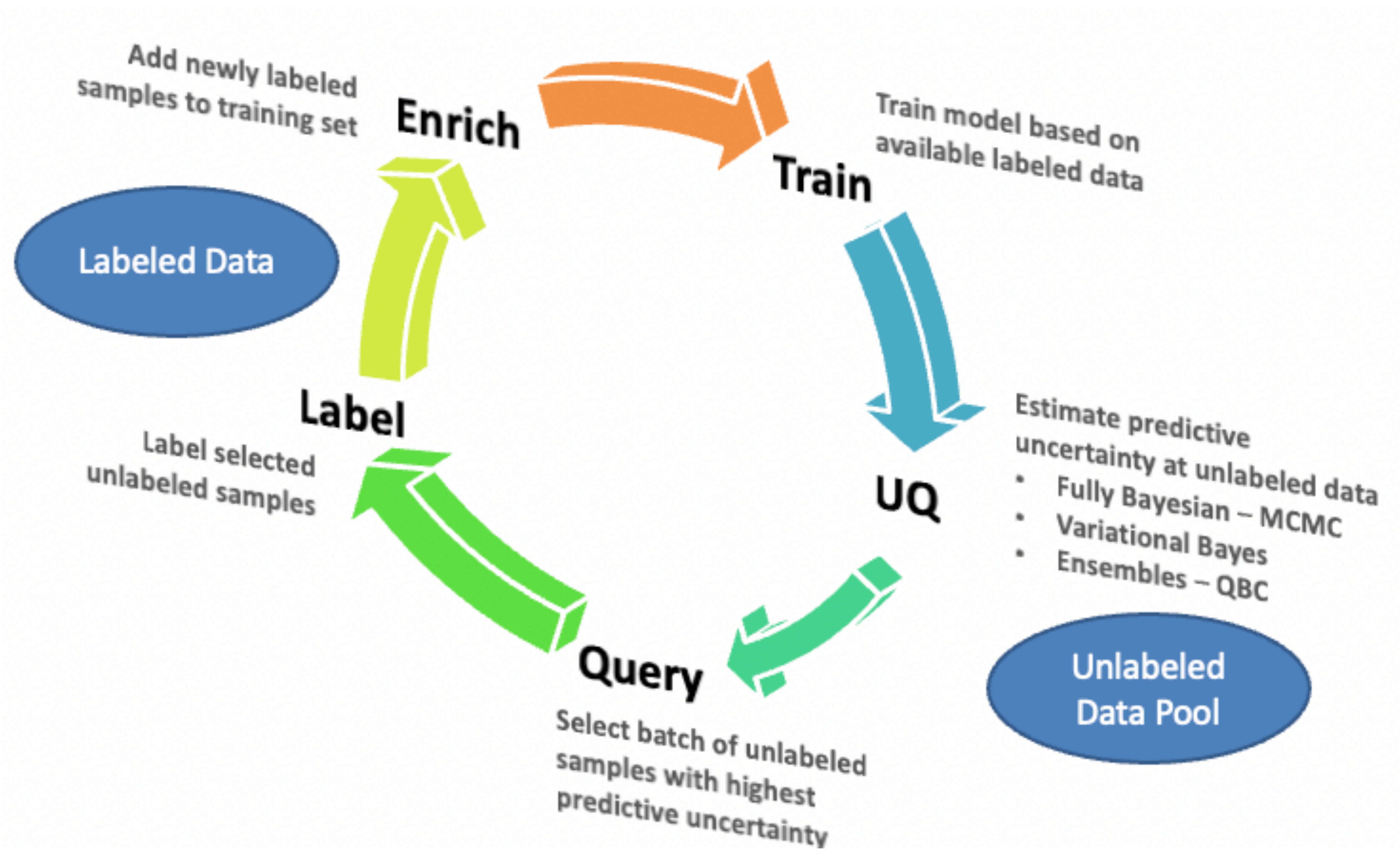
Straight out of wiki….

- **A**-optimality ("**average**" or **trace**)
  - One criterion is **A-optimality**, which seeks to minimize the trace of the inverse of the information matrix. This criterion results in minimizing the average variance of the estimates of the regression coefficients.
- **C**-optimality
  - This criterion minimizes the variance of a best linear unbiased estimator of a predetermined linear combination of model parameters.
- **D**-optimality (**determinant**)
  - A popular criterion is **D-optimality**, which seeks to minimize $|(X'X)^{-1}|$, or equivalently maximize the determinant of the information matrix $X'X$ of the design. This criterion results in maximizing the differential Shannon information content of the parameter estimates.
- **E**-optimality (**eigenvalue**)
  - Another design is **E-optimality**, which maximizes the minimum eigenvalue of the information matrix.
- **T**-optimality
  - This criterion maximizes the trace of the information matrix.

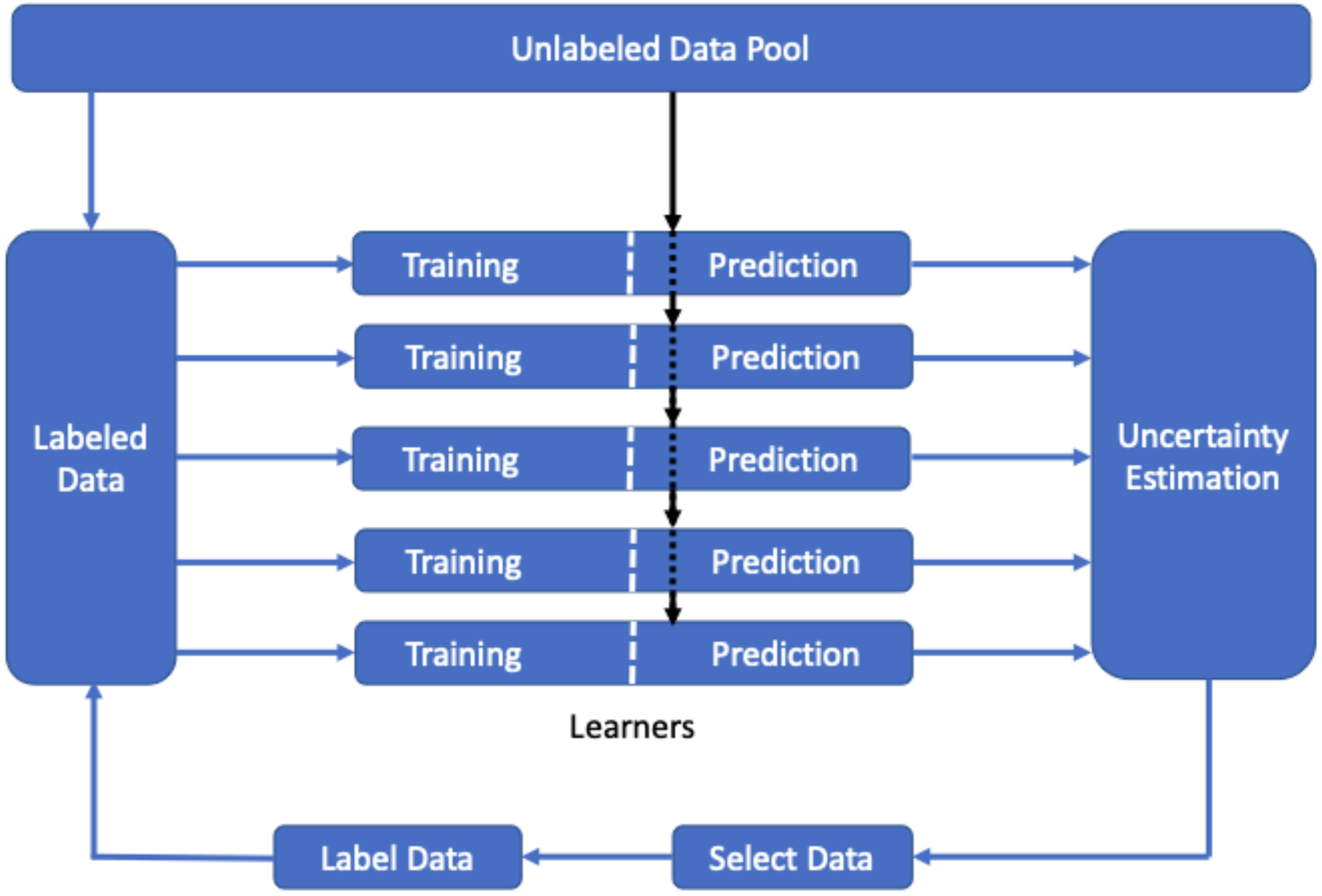
Other optimality-criteria are concerned with the variance of predictions:

- **G**-optimality
  - A popular criterion is **G-optimality**, which seeks to minimize the maximum entry in the diagonal of the hat matrix $X(X'X)^{-1}X'$. This has the effect of minimizing the maximum variance of the predicted values.
- **I**-optimality (**integrated**)
  - A second criterion on prediction variance is **I-optimality**, which seeks to minimize the average prediction variance *over the design space*.
- **V**-optimality (**variance**)
  - A third criterion on prediction variance is **V-optimality**, which seeks to minimize the average prediction variance over a set of m specific points.[9]

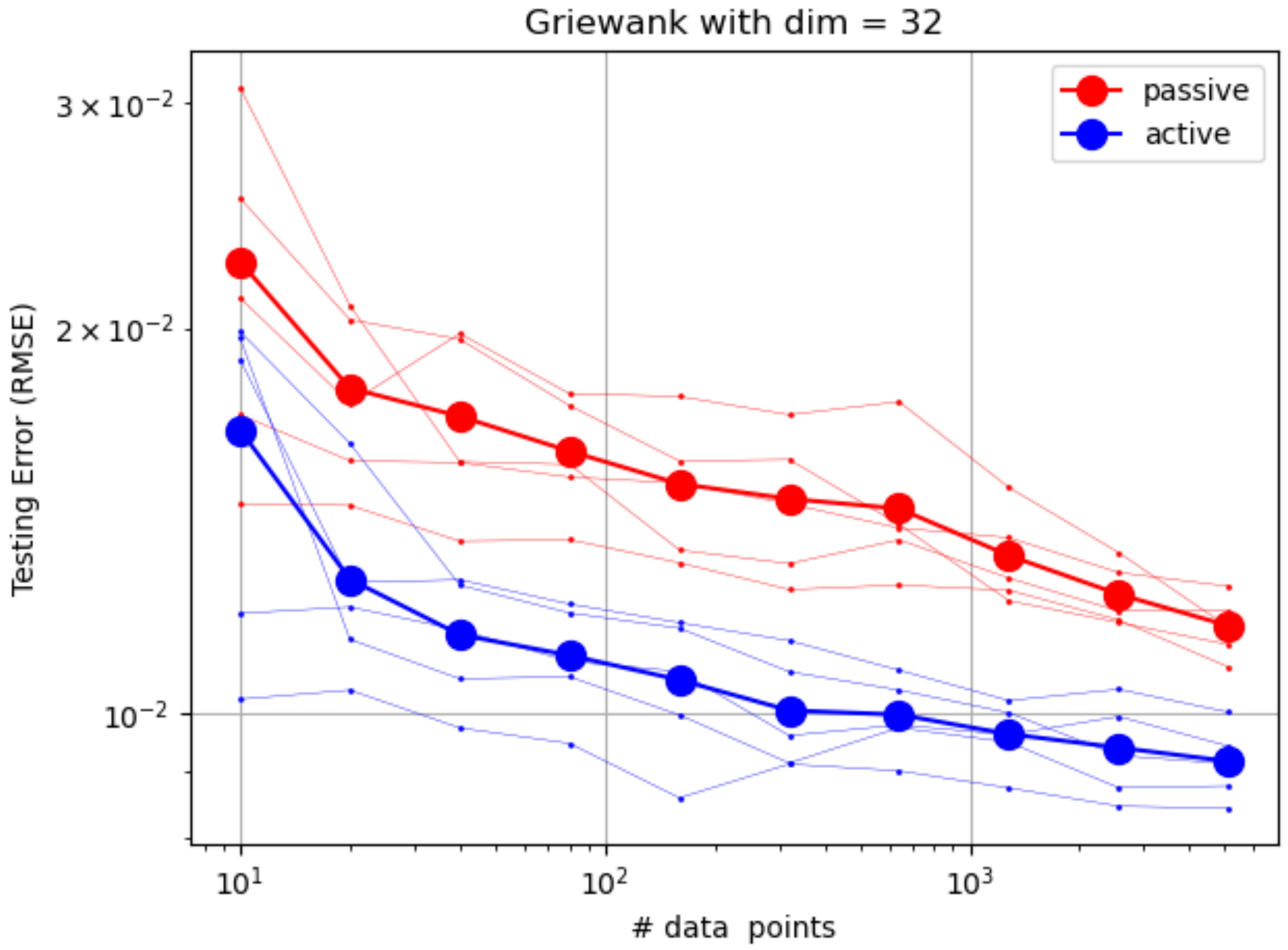
Active Learning Loop



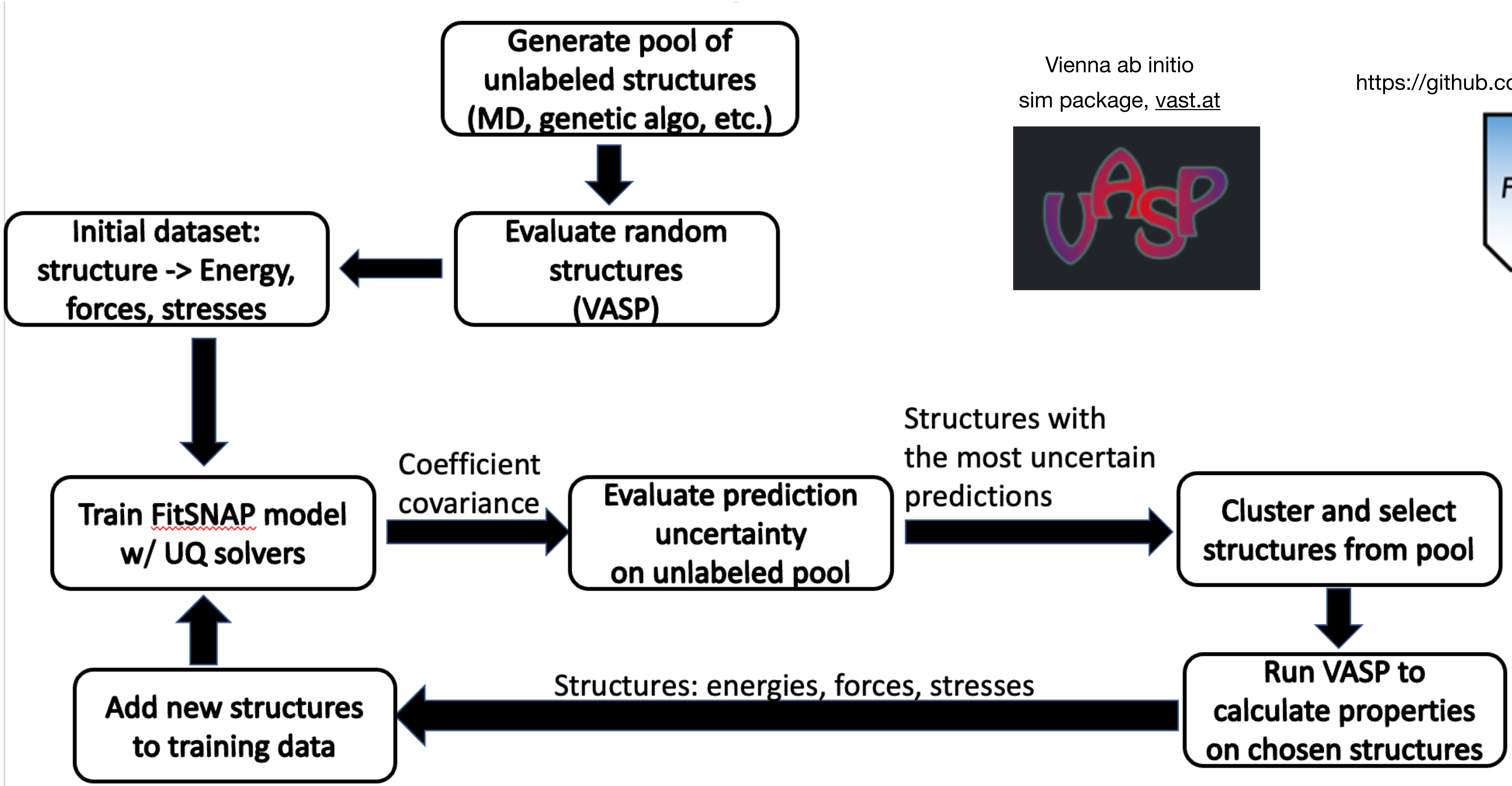
Active Learning: Query-by-Committee



- Start with a training set of N points
- Launch K learners, each with fN training points (f=0.8)
- Evaluate the learners' performance at all points in the pool
- Select training points from the pool that correspond to the highest 'disagreement' and add them to the training set



Active Learning: planned workflow



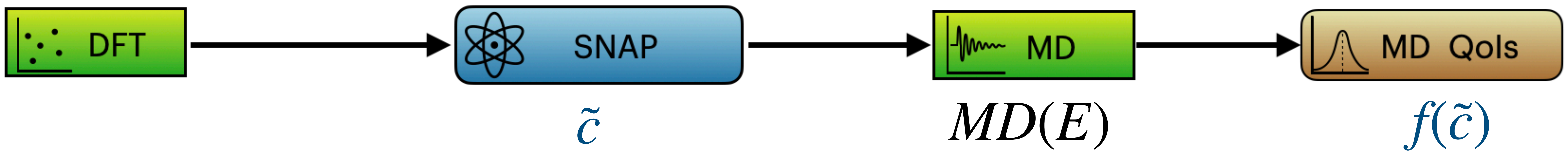
Vienna ab initio sim package, vast.at

https://github.com/FitSNAP/FitSNAP

Active Learning: Questions….

- Good starting point perhaps? Bayesian coresets?
- Search is in configuration space x, but the data is $E(x), F(x), S(x)$
- 'Newness'/extrapolation, measures of well-sampledness
  - convex hull in/out
  - distance from training set
  - disagreement/st.dev. in ensemble of models (QBC)
  - kernel density estimation
- Clustering method to diversify the selection of new batch
- Metric should perhaps be driven by the ‘outer’ task
  - reaction search
  - forward UQ in MD, outlier/anomaly detection

Forward UQ Plan



$$E \approx \sum_{k=0}^{P} \underbrace{(c_k + d_k \xi_k)}_{\tilde{c}} B_k(x)$$

SNAP coefficients form a first order Gauss-Hermite Polynomial Chaos (PC)

- Sample SNAP coefficients
- Evaluate MD QoIs
- Build PC for MD QoIs, possibly multilevel/multifidelity
- Evaluate PDF/statistics of QoIs
- Challenges: high-d input, noisy MD simulations

Summary

- Embedded **model error** for Bayesian inference of MLIAPs
    - Leads to data model with baked-in uncertainty
    - Meaningful model-error uncertainty capturing the true residual
    - Non-negligible coefficient uncertainty that can be propagated through MD
    - Choices to make: priors, likelihoods, MCMC sampler, where to embed…

- Initiating a workflow for **active learning** via QBC
    - Anchored in uncertainty estimation, even if heuristic
    - Promising results on toy models
    - Engaging FitSNAP-VASP feedback
    - Choices to make: query strategy, UQ method, metric of ‘newness’…

Extras

Posterior Predictions

- ✦ Bayesian inference hinges on likelihood function or data noise (DN) model

  e.g. Gaussian i.i.d. $y_i = f(x_i, c) + \sigma\epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0,1)$

- ✦ After we obtain posterior PDF $p(c \mid y)$, there are two useful predictive quantities:

  Push-forward (PF): $p(f \mid y)$

  Posterior predictive (PP): $p(y^* \mid y) = \int p(y^* \mid c) p(c \mid y) dc$

PP = PF + DN

Mean_PP = Mean_PF

Var_PP = Var_PF + σ^2

Posterior PDF sampling via MCMC

Posterior PDF   Prior PDF

$$p(c \mid y) \propto p(y \mid c)p(c)$$

Likelihood

- ✦ The likelihood requires assumptions regarding model/data relationships
- ✦ No closed form expression for posterior PDF unless very specialized likelihoods are used
- ✦ Need to resort to sampling the posterior, rather than evaluating directly
- ✦ Markov chain Monte Carlo is the main vehicle for posterior sampling

Linear Models: luxury of analytical answers

- ✦ Linear least-squares regression (polynomial, bispectrum, …) $y \approx Ac$

$$c^* = \text{argmin}_c \, || \overbrace{WA}^{\tilde{A}} \, c - \overbrace{Wy}^{\tilde{y}} ||^2$$

where W is a diagonal matrix of weights, e.g. w_i driven by Dakota.

- ✦ Equivalent unweighted least-squares w/ scaled data: $c^* = \text{argmin}_c \, ||\tilde{A}c - \tilde{y}||_2^2$ $\quad c^* = \text{argmin}_c \, \frac{||\tilde{A}c - \tilde{y}||_2^2}{\sigma^2}$

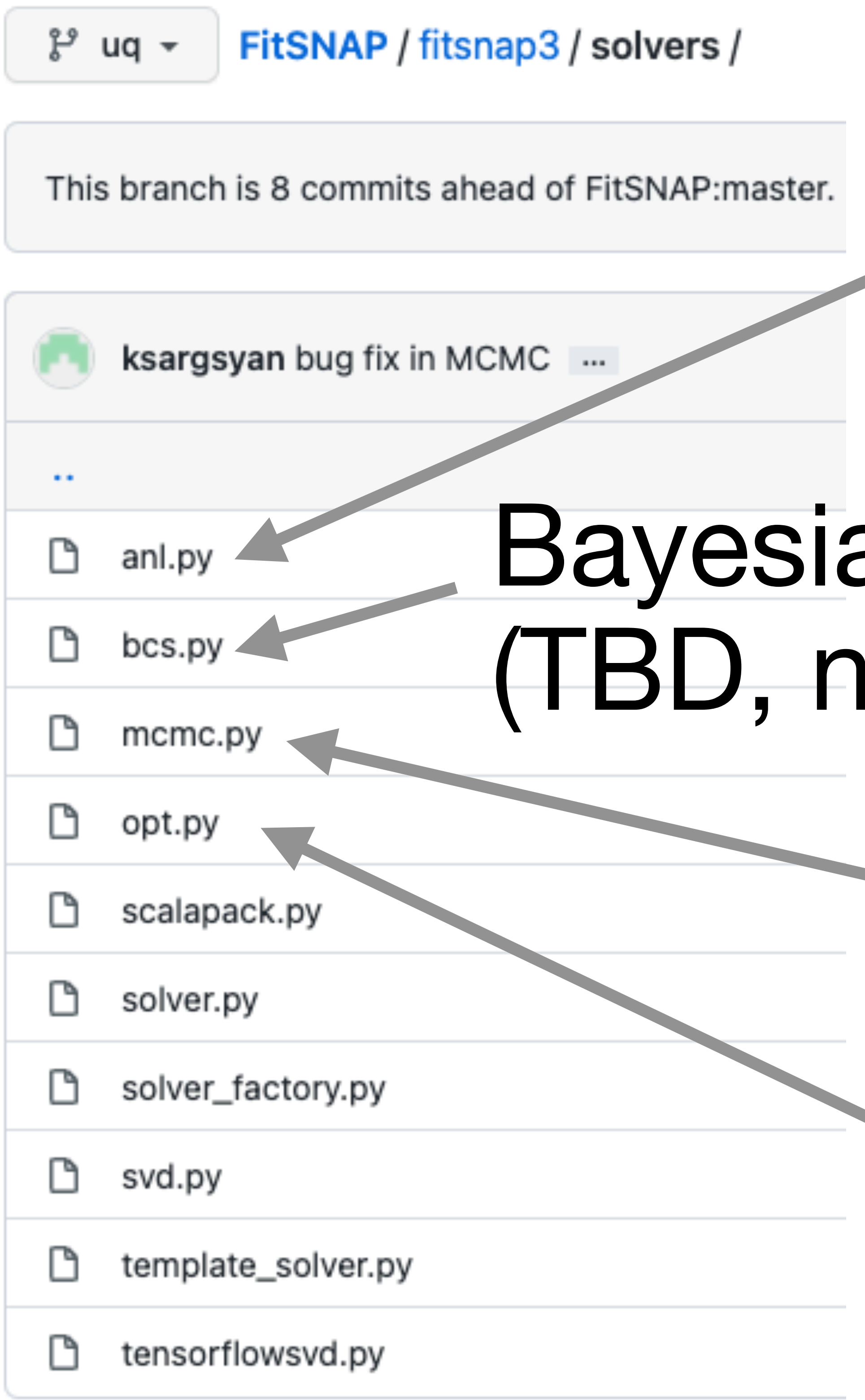
- ✦ Deterministic: $c^* = (\tilde{A}^T\tilde{A})^{-1}\tilde{A}^T y$

- ✦ Bayesian posterior PDF… $c = \mathcal{N}\left(c^*, (\tilde{A}^T\tilde{A})^{-1}\right)$ $\quad c = \mathcal{N}\left(c^*, \sigma^2(\tilde{A}^T\tilde{A})^{-1}\right)$

- ✦ … but it is better to include σ

FitSNAP solvers with Uncertainty

Forked and created a UQ branch



merr.py

Analytical Bayesian linear regression

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

Bayesian compressive sensing (TBD, need bispectrum pruning)

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

MCMC

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

Optimization via scipy.optimize

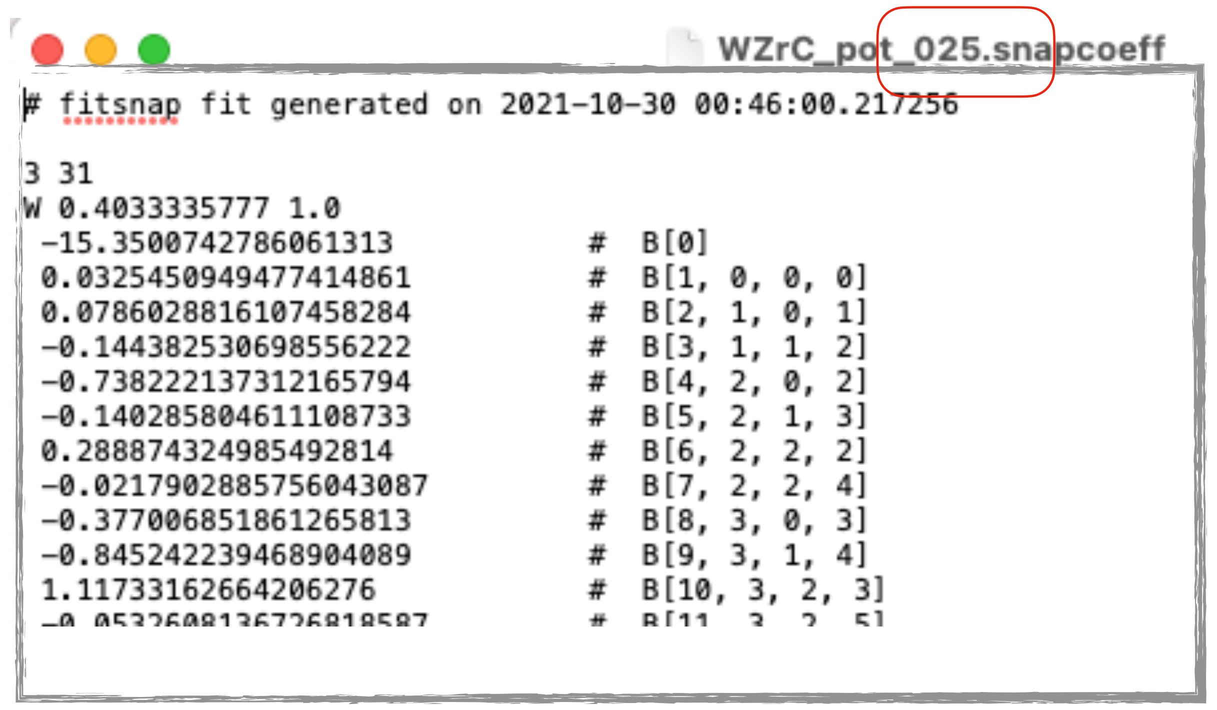
```
[SOLVER]
solver = OPT
```

Model error, TBD

this talk

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

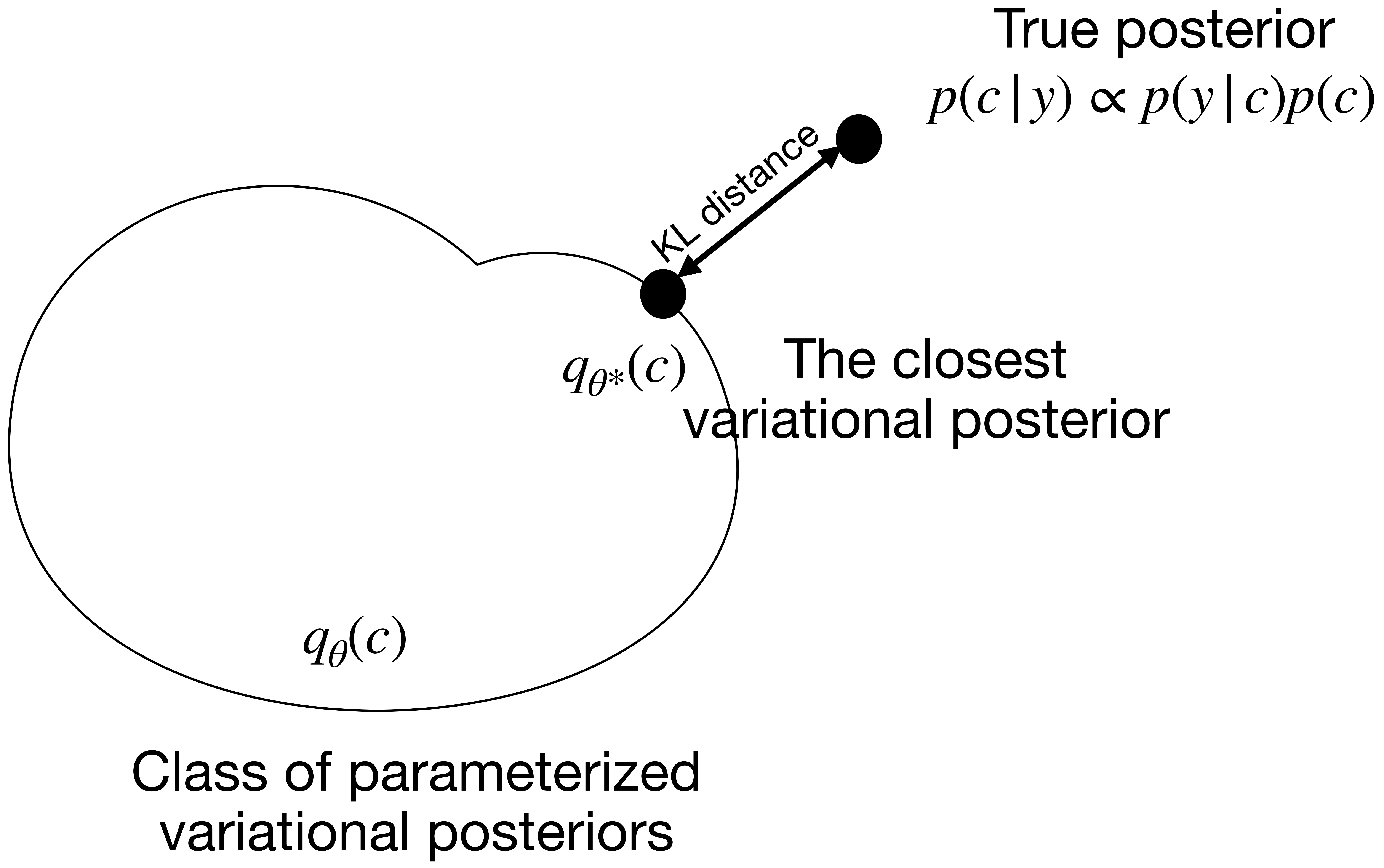
Sample # 25


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]
```

Variational inference finds an approximate posterior PDF



Variational inference

- ✦ Kullback-Leibler Divergence: ‘distance’ metric between PDFs

$$KL(p_1||p_2) = \int \ln\left(\frac{p_1(x)}{p_2(x)}\right)p_1(x)dx$$

- ✦ KL between Variational and True Posteriors:

$$KL\left(q_\theta(c)||p(c|y)\right) = \ldots = \underbrace{KL\left(q_\theta(c)||p(c)\right) - \int q_\theta(c)\ln p(y|c)dc}_{\int q_\theta(c)\left[\ln q_\theta(c) - \ln p(c) - \ln p(y|c)\right] dc} + const$$

Minimize this:

many flavors exist, e.g. with stochastic gradient descent and Monte-Carlo sampling

Weighted interpolation [Ischtwan 1994; Dowes, 2007-09; Maisuradze, 2009]

Permutationally invariant polynomials [Xie, 2010]

Gaussian processes [Bartok, Csanyi 2010-15; Mills, 2012; Rupp, 2013; Cui, 2016; U
Guan, 2018; Schmitz, 2018]

Low-rank tensor expansions [Jackle, 1996; Baranov, 2015; Rai, 2017, 2018] Suppor
machines, kernel regression [Le, 2009; Balabin, 2011;

Dral, 2017]

Neural networks (NN) [Blank, 1995; Tai No, 1997; Prudente, 1998; Lorenz, 2004; Wit
Manzhos, 2006-09; Malshe, 2008; Le, 2009] [Behler, 2010-16; Handley, 2010, 2014;
Li, 2013; Dolgirev, 2016; Khorshidi, 2016; Peterson, 2016; Carr, 2016; Kolb, 2016; S
Chmiela, 2017; Cubuk, 2017; McGibbon, 2017; Smith, 2017; Schutt, 2017; Yao, 201
2017; Bereau, 2018; Lubbers, 2018; Unke, 2018; Wang, 2018; Natarajan, 2018; Zha
Onat, 2018]

Challenges Galore

Bayes MLIAP + Model Error

- Likelihood choice
- Prior selection for model-error embedding parameters
- Incorporation of DFT errors?
- Large set of training data, matrix inversions infeasible
- Weighting between energies, forces and stresses