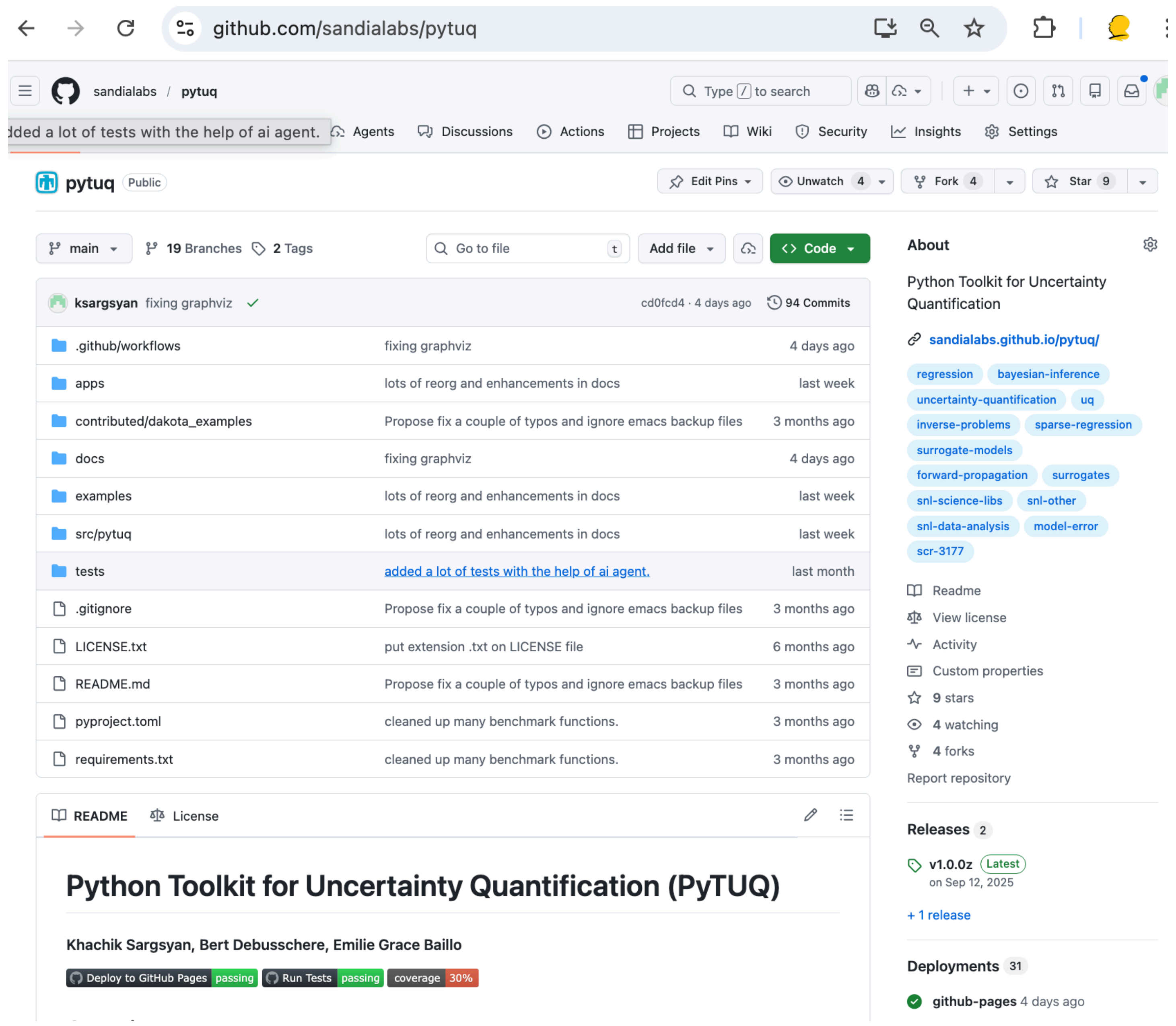
PyTUQ: Python Toolkit for Uncertainty Quantification

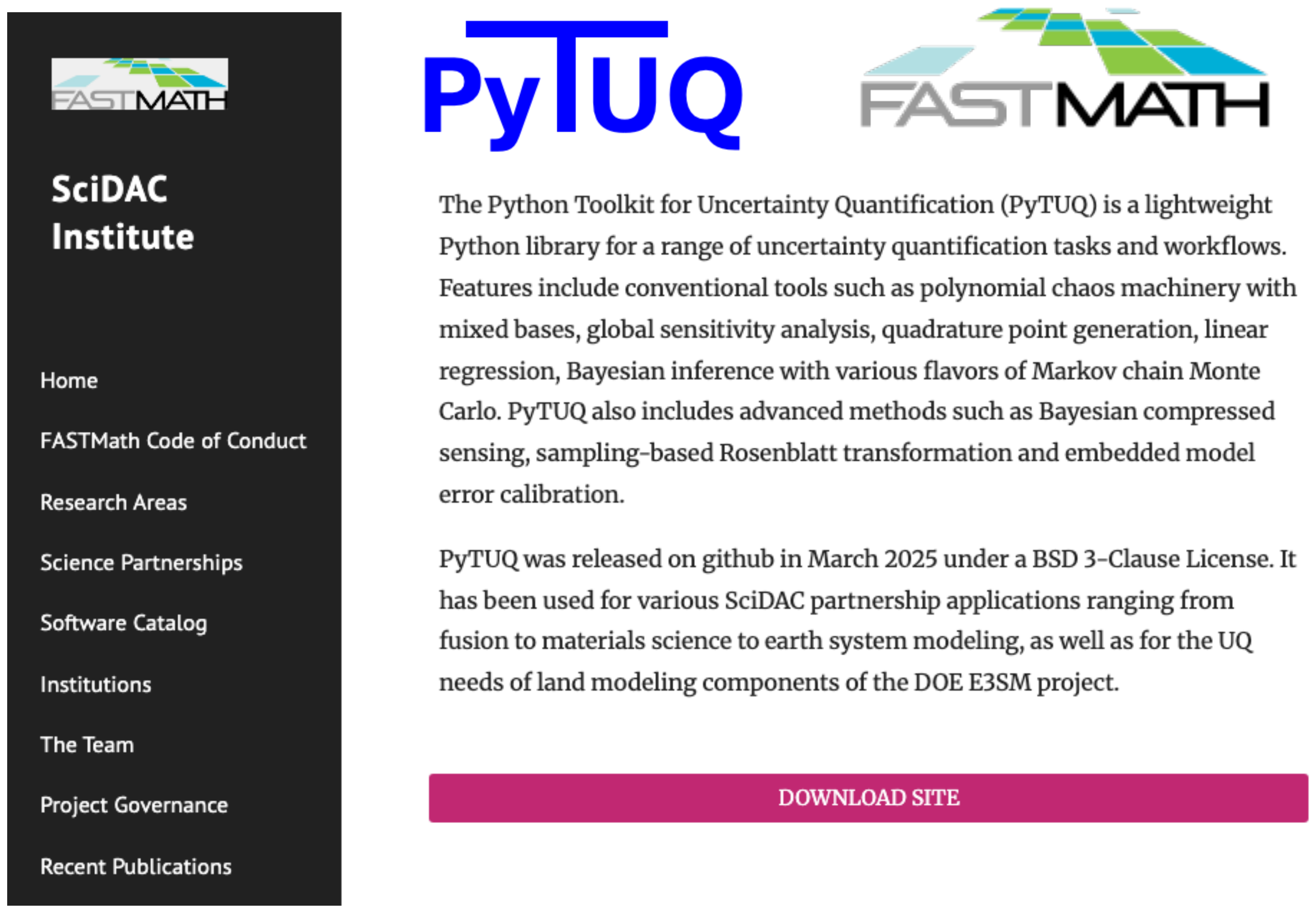
Khachik Sargsyan, Emilie Grace Baillo, Bert Debusschere

Sandia National Laboratories

https://github.com/sandialabs/pytuq

https://sandialabs.github.io/pytuq/





Overview

- Released March 2025.
- Python-only offspring of UQTk.
- Part of SciDAC-6 FASTMath Software catalogue.
- Dependencies: numpy, scipy, matplotlib, (pytorch, quinn, pso, dill).
- License: BSD 3-Clause License.
- Pip installable https://pypi.org/project/pytuq/
- Part of Spack https://packages.spack.io/package.html?name=py-pytuq

K. Sargsyan (ksargsy@sandia.gov)

PyTUQ for computational science

Toolbox for method development and uncertainty quantification workflows for black-box computational models

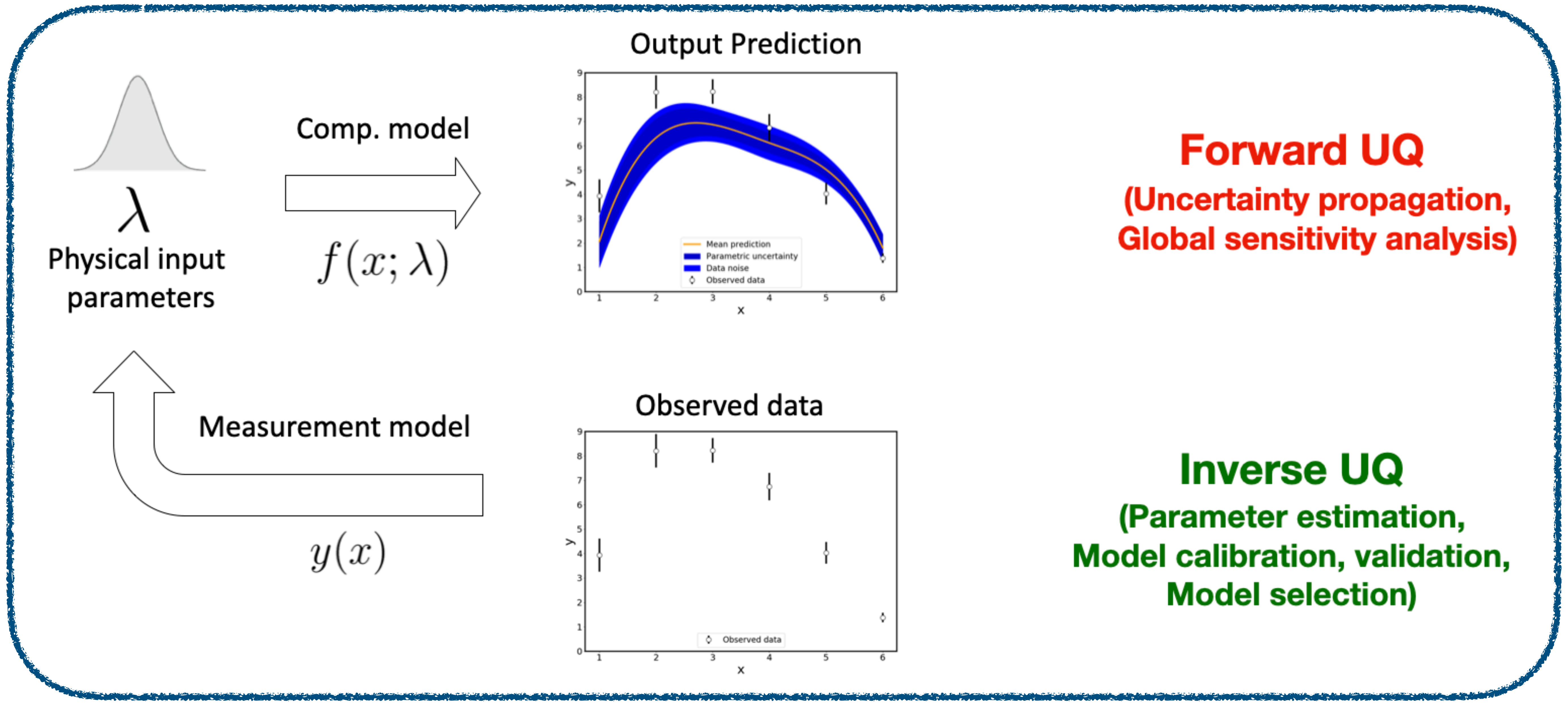
Param 1 | Param 2 | P 3 | P 4 | Param 5 | Data Noise | Surrogate Error | Model Structural Error | Intrinsic Noise

<u>Forward UQ tools:</u>

Surrogates; Sparse learning; Karhunen-Loève; Rosenblatt; PC; GP; GSA

<u>Inverse UQ tools:</u>

Bayesian inference; MCMC; Variational; Model error

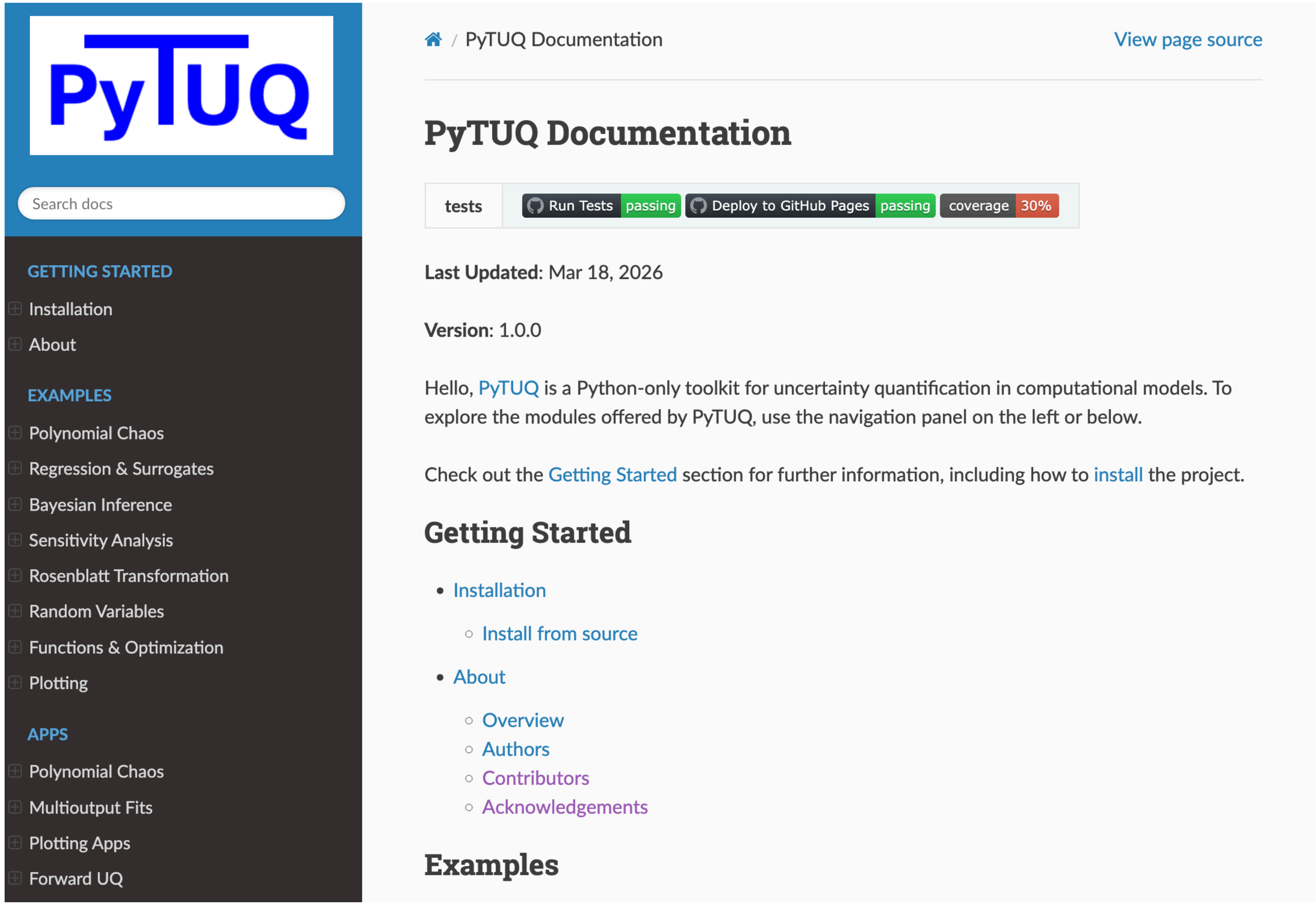


K. Sargsyan (ksargsy@sandia.gov)

PyTUQ Capabilities

- Uncertainty propagation and global sensitivity analysis via polynomial chaos (PC)
- Regression (PC, custom bases, GP)
- Global Sensitivity Analysis (MOAT, Sobol, PC-based)
- Functional tools (Integration, Orthogonalization)
- Linear dimensionality reduction (**Karhunen-Loève**, SVD)
- Reduced-dimensional **KLPC** surrogates for multioutput models
- Bayesian linear regression (VI, Sparse/**Bayesian compressive sensing**, Embedded)
- Optimization (scikit wrappers, particle-swarm, gradient descent)
- Model calibration via Markov chain Monte Carlo (AMCMC, HMC, MALA), MFVI.
- **Embedded model error** estimation
- Random variables (**PC**, MCMC-RV, **Rosenblatt,** GMMs, …**)**
- **Large library** of analytical benchmark functions
- Lots of command-line examples, apps and workflows for common UQ tasks

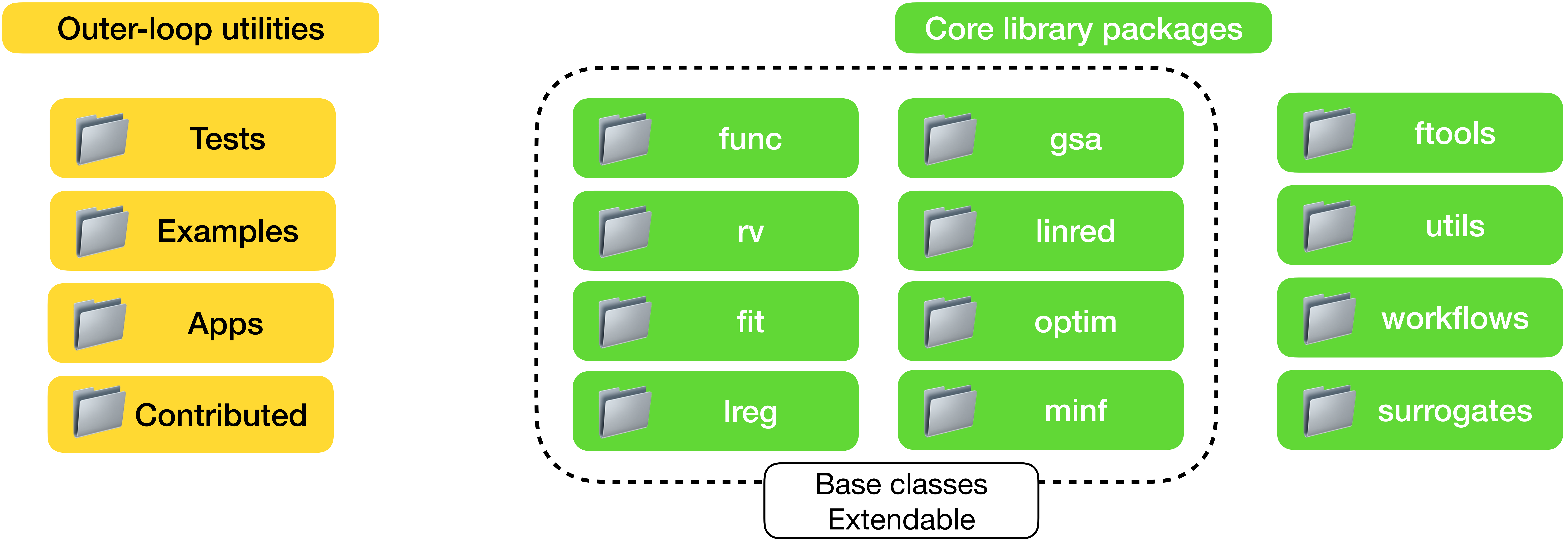
Documentation



https://sandialabs.github.io/pytuq/

K. Sargsyan (ksargsy@sandia.gov)

Overall Structure



- Not intended to comprehensively cover all methods:
  add functionality as needed.
- Key UQ functionalities written with OOP principles, with base classes:
  straightforward extensions via class inheritance.

K. Sargsyan (ksargsy@sandia.gov)

Core library packages

- src
  - func — Multivariate functions $f : \mathbb{R}^d \to \mathbb{R}^o$
  - rv — Multivariate random variables (PCRV, GMM, …)
  - lreg — Linear regression (ANL, LSQ, BCS, …)
  - fit — Nonlinear fits (GP)
  - gsa — Global sensitivity analysis (Sobol, PC, MOAT, …)
  - linred — Linear dimensionality reduction (PCA, KLE, …)
  - optim — Optimization (GD)
  - minf — Model inference (Bayes/MCMC/Embedded Model error)
  - ftools — Function-based methods
  - utils — Plotting, data manipulation …
  - workflows — Utilities chaining a few tasks
  - surrogates — Surrogate wrappers

K. Sargsyan (ksargsy@sandia.gov)

rv

Random variable package

mrv.py

pcrv.py

rosen.py

- Parent random variable class
- Multivariate random variables
- **PCRV**, GMM, MVN, MCMCRV
- Sample, evaluate pdf, cdf
- Operations on random variables: mixture, inverse
-
- Rosenblatt transformation

func

Function package

- func.py
  - bench*.py
  - genz.py
  - toy.py
  - poly.py
- chem.py
- oper.py

- Parent function class
- Multivariate, multioutput functions $f : \mathbb{R}^d \to \mathbb{R}^o$
- Lots of benchmark functions
- Gradients/Hessians
- **Operations on functions: compose, slice, partial, …**
- Plotting routines of various slices
- See *ex_func.py, ex_funcall.py* and *ex_funcgrad.py*

```
Fcn1 = bench.Ishigami()
Fcn2 = benchNd.Ackley(d=3)
Fcn = Fcn1 * Fcn2
Fcn.grad(x)
Fcn.plot_2d(d1, d2)
```

```
pcrv = PCRV(nout, ndim, pctype)
pcrv.setFunction()
newfcn = pcrv.function + bench.Ishigami()
```

For most benchmark functions, credit to https://github.com/Vahanosi4ek

Examples

Examples of varying complexity

- Meant to be run as-is
- The right place to start
- Outputs *.png* and *.txt*

ex_bcs.py*	Bayesian Compressive Sensing
ex_gp.py	Gaussian Process regression
ex_gsa.py*	Global Sensitivity Analysis
ex_*ros.py*	(Inverse) Rosenblatt transform
ex_kl.py*	Karhunen-Loève decomposition
⋮	⋮
ex_lreg.py*	Linear regression
ex_mcmc.py*	Markov chain Monte Carlo
ex_minf.py*	Embedded model error inference
ex_pcrv.py	PC multivariate random variable
ex_uprop.py	Uncertainty propagation
⋮	

Apps

Apps and workflows

- Typically command-line arguments
- Similar to command line ‘apps’ in UQTk

kl_fit.py — KL fit given data
klsurr_fit.py — KL+PC/NN surrogate

wf_uqpc.x
pc_prep.py
pc_sam.py
pc_fit.py — Workflow for PC-based propagation

nn_fit.py — NN surrogate. Quick and dirty.
Relies on QUiNN library *https://github.com/sandialabs/quinn*

mrun.py — Multithreading app:
Creates folders and runs a list of command lines

plot_ens.py — Creates regression (supervised) data from benchmark functions

plot_*.py — Plotting apps

K. Sargsyan (ksargsy@sandia.gov)

Apps

Apps and workflows

- *uqpc/* — workflow for PC-based uncertainty propagation of mulitoutput models
- *iuq/* — workflow for Bayesian calibration enabled by PC surrogate
- *awkies/* — useful awk-based command-line utilities for tabular data

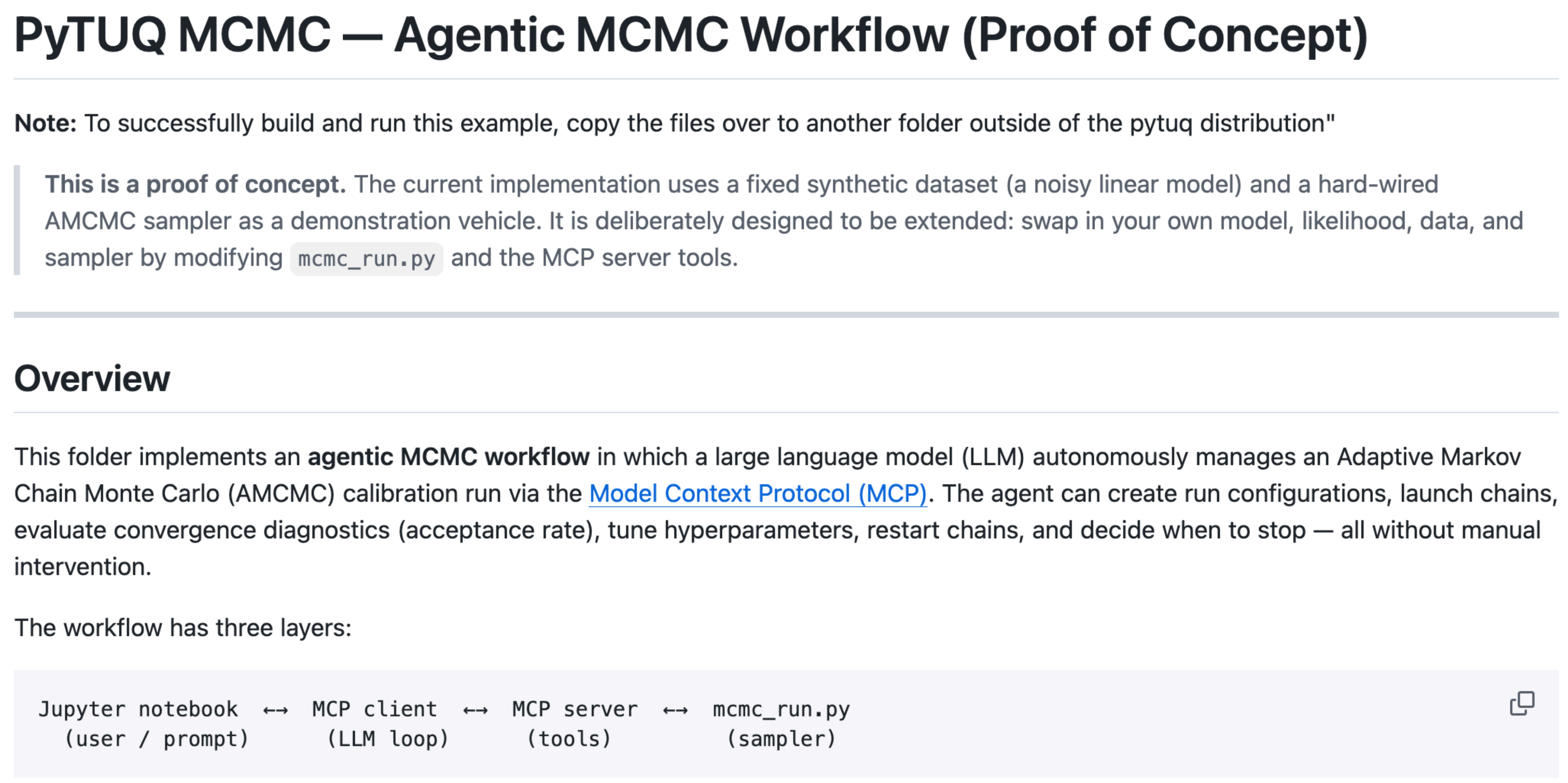
K. Sargsyan (ksargsy@sandia.gov)

Contributed

Contributed packages/modules

Dakota wrapper examples for surrogate construction

Agentic MCMC proof-of-concept



PyTUQ MCMC — Agentic MCMC Workflow (Proof of Concept)

Note: To successfully build and run this example, copy the files over to another folder outside of the pytuq distribution"

> **This is a proof of concept.** The current implementation uses a fixed synthetic dataset (a noisy linear model) and a hard-wired AMCMC sampler as a demonstration vehicle. It is deliberately designed to be extended: swap in your own model, likelihood, data, and sampler by modifying `mcmc_run.py` and the MCP server tools.

Overview

This folder implements an **agentic MCMC workflow** in which a large language model (LLM) autonomously manages an Adaptive Markov Chain Monte Carlo (AMCMC) calibration run via the Model Context Protocol (MCP). The agent can create run configurations, launch chains, evaluate convergence diagnostics (acceptance rate), tune hyperparameters, restart chains, and decide when to stop — all without manual intervention.

The workflow has three layers:

```
Jupyter notebook  ←→  MCP client  ←→  MCP server  ←→  mcmc_run.py
  (user / prompt)      (LLM loop)      (tools)          (sampler)
```

Room for external contributions

Methodological Highlights

Inv UQ: Case for Embedded Model Error

- Allows meaningful extrapolation
- Respects physics
- Disambiguates model and data errors
- Predictive uncertainty attribution

$$\boxed{g(x_i) = f(\lambda + \delta_\alpha(x_i), x_i) + \epsilon_i}$$

- Infer physical parameters λ and model-error parameters α together

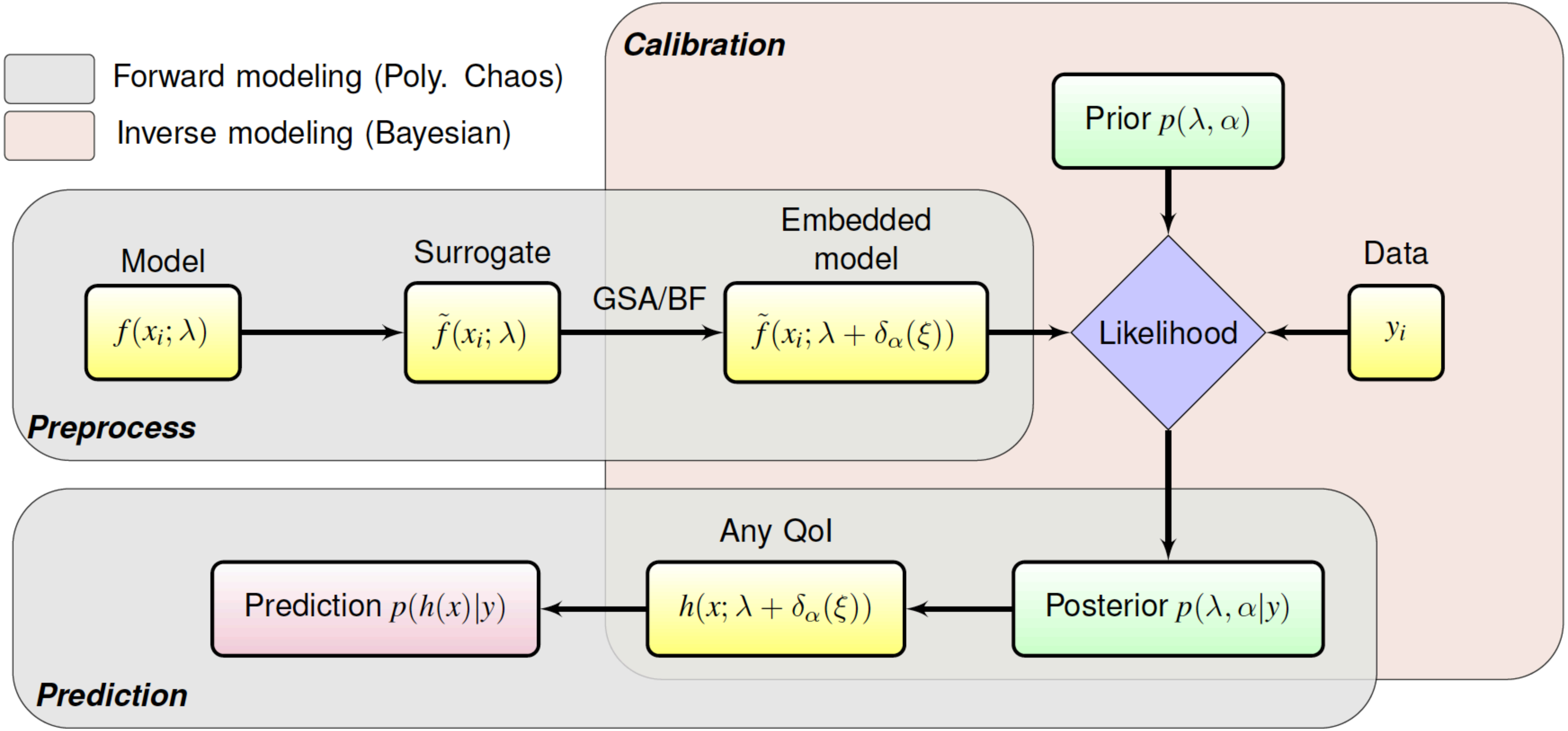
- In practice, cast the parameters as a PC expansion $\lambda = \sum_k \alpha_k \Psi_k(\xi)$

- Propagate (forward UQ) PC $f\left(\sum_k \alpha_k \Psi_k(\xi), x_i\right) \approx \sum_k f_k(\alpha)\Psi_k(x_i)$

- Build approximate likelihood functions based on data model $g(x_i) = \sum_k f_k(\alpha)\Psi_k(x_i) + \epsilon_i$
  (e.g. match moments, or Gaussian iid approximation)

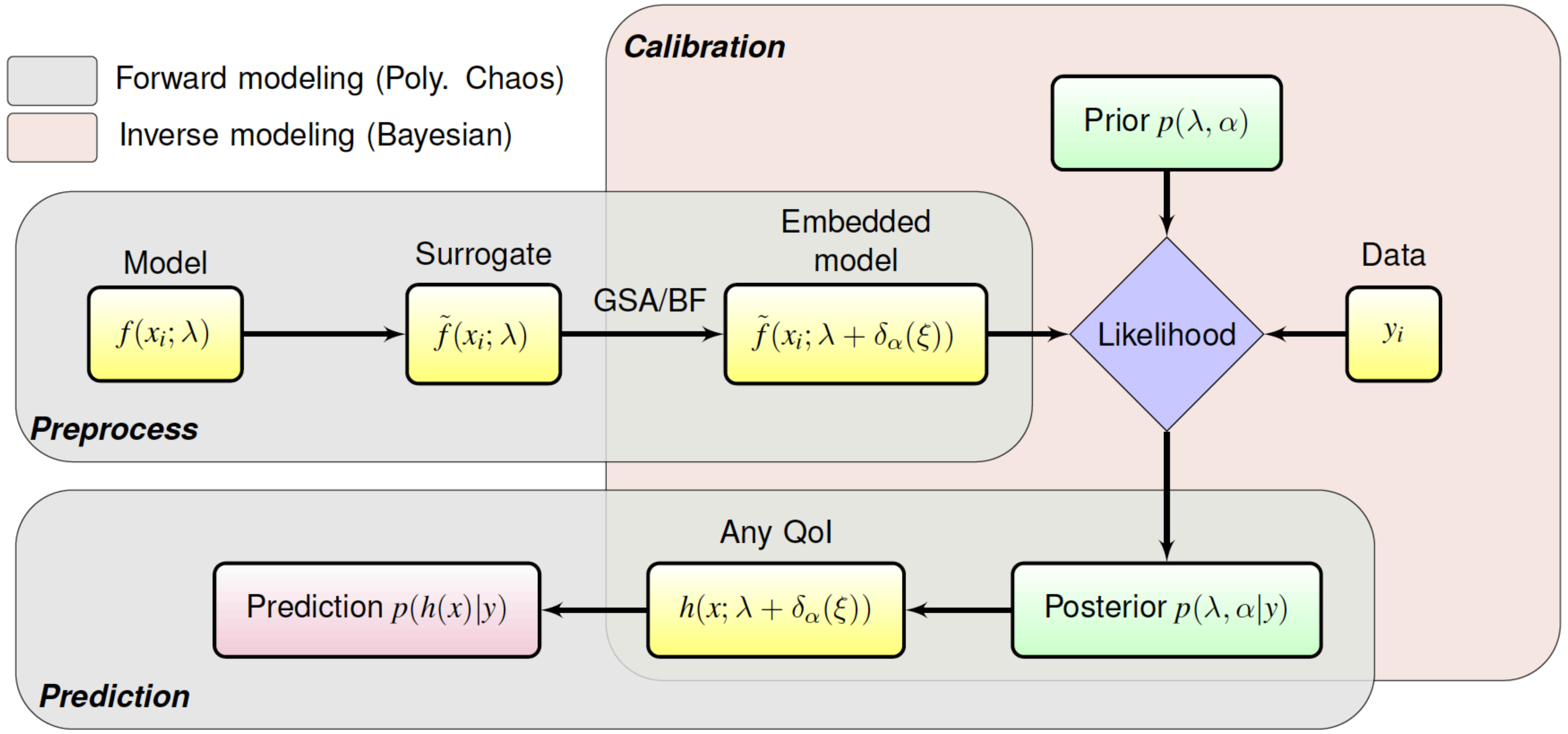
K. Sargsyan (ksargsy@sandia.gov)

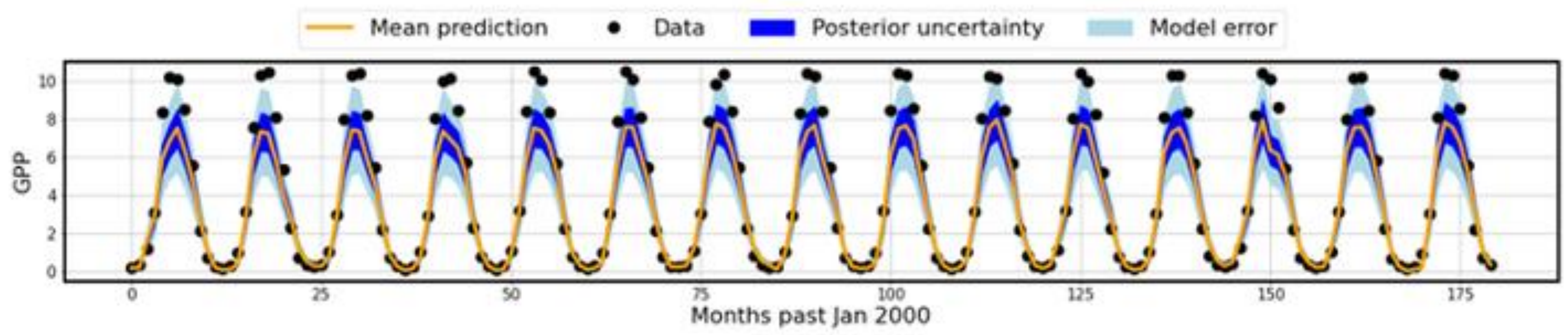
Inv UQ: Surrogate-based Bayesian Inference with Embedded Model Error

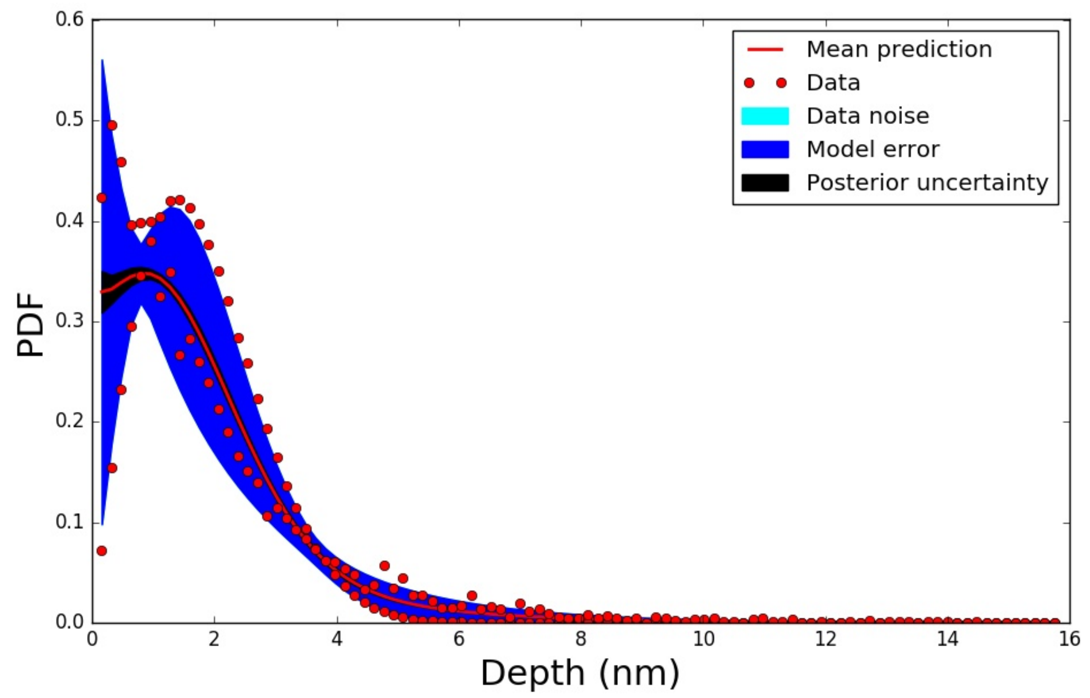


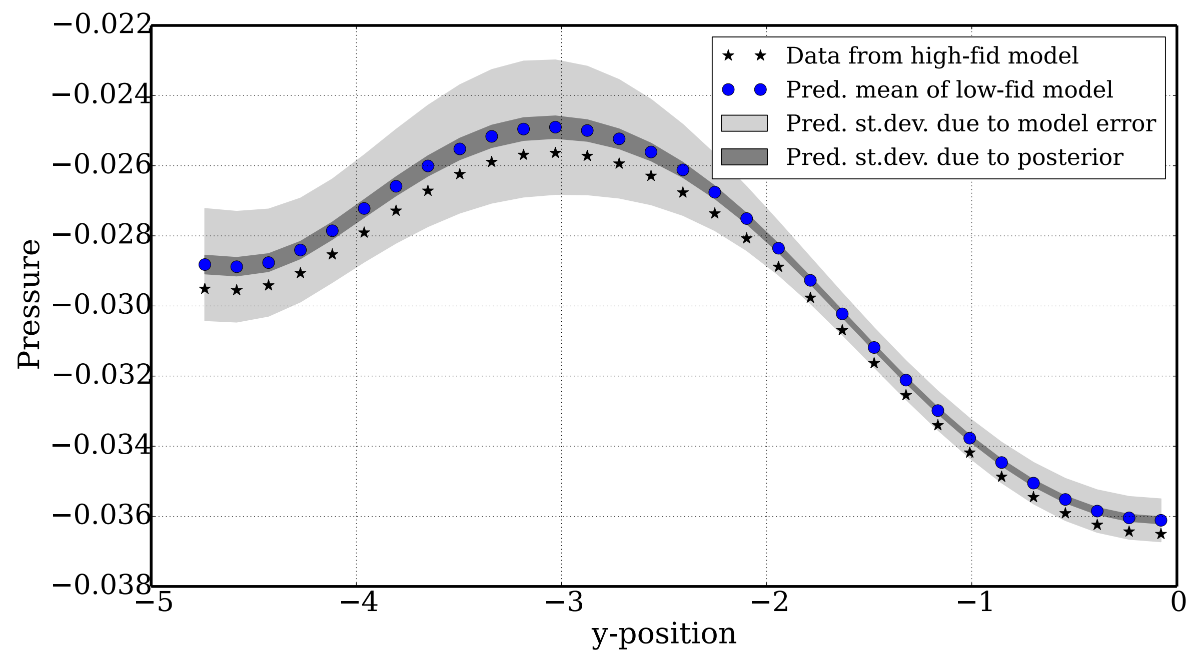
 K. Sargsyan (ksargsy@sandia.gov)

EME workflow and applications









[1] K. Sargsyan, H. N. Najm, R. Ghanem, “On the Statistical Calibration of Physical Models”, International Journal for Chemical Kinetics, 47:4, pp. 246–276, 2015.

[2] K. Sargsyan, X. Huan, H. N. Najm. “Embedded Model Error Representation for Bayesian Model Calibration”, International Journal of Uncertainty Quantification, 9:4, pp. 365–394, 2019.

[3] T.R. Younkin, K. Sargsyan, T. Casey, H.N. Najm, J.M. Canik, D.L. Green, R.P. Doerner, D. Nishijima, M. Baldwin, J. Drobny, D. Curreli, B.D. Wirth, “Quantification of the effect of uncertainty on impurity migration in PISCES-A simulated with GITR”, Nuclear Fusion, 62:5, p.056007, 2022.

[4] O. Cekmer, K. Sargsyan, S. Blondel, H. Najm, D. Bernholdt,, B.D. Wirth, “Uncertainty quantification for incident helium flux in plasma-exposed tungsten”, International Journal for Uncertainty Quantification, 8:5, pp.429–446, 2018.

[5] X. Huan, C. Safta, K. Sargsyan, G. Geraci, M. S. Eldred, Z. P. Vane, G. Lacaze, J. C. Oefelein, Habib N. Najm, “Global Sensitivity Analysis and Estimation of Model Error, toward Uncertainty Quantification in Scramjet Computations” AIAA Journal, 56:3, pp.1170–1184, 2018.

[6] C. Safta, M. Blaylock, J. Templeton, S. Domino, K. Sargsyan, H. Najm, “Uncertainty Quantification in LES of Channel Flow”, International Journal for Numerical Methods in Fluids, 83, pp.376–401, 2016.

[7] F. Ghahari, K. Sargsyan, G. Parker, D. Swensen, M. Celebi, H. Haddadi, E. Taciroglu, “Performance Based Earthquake Early Warning for Tall Buildings”, Earthquake Spectra, 40:2, 2024.

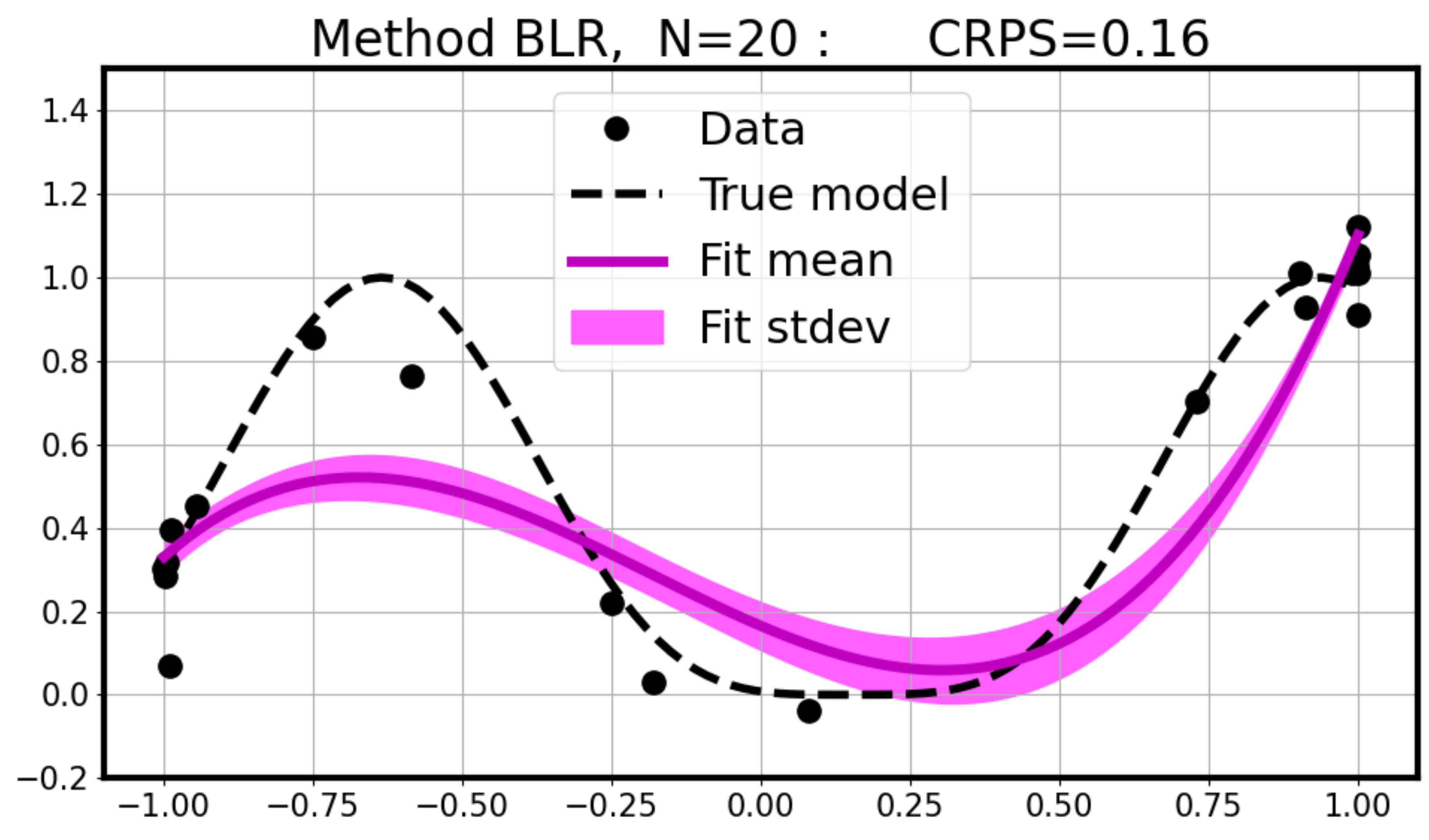
[8] F. Ghahari, K. Sargsyan, M. Celebi, E. Taciroglu, “Quantifying modeling uncertainty in simplified beam models for building response prediction”, Structural Control and Health Monitoring, 29:11, p.e3078, 2022.

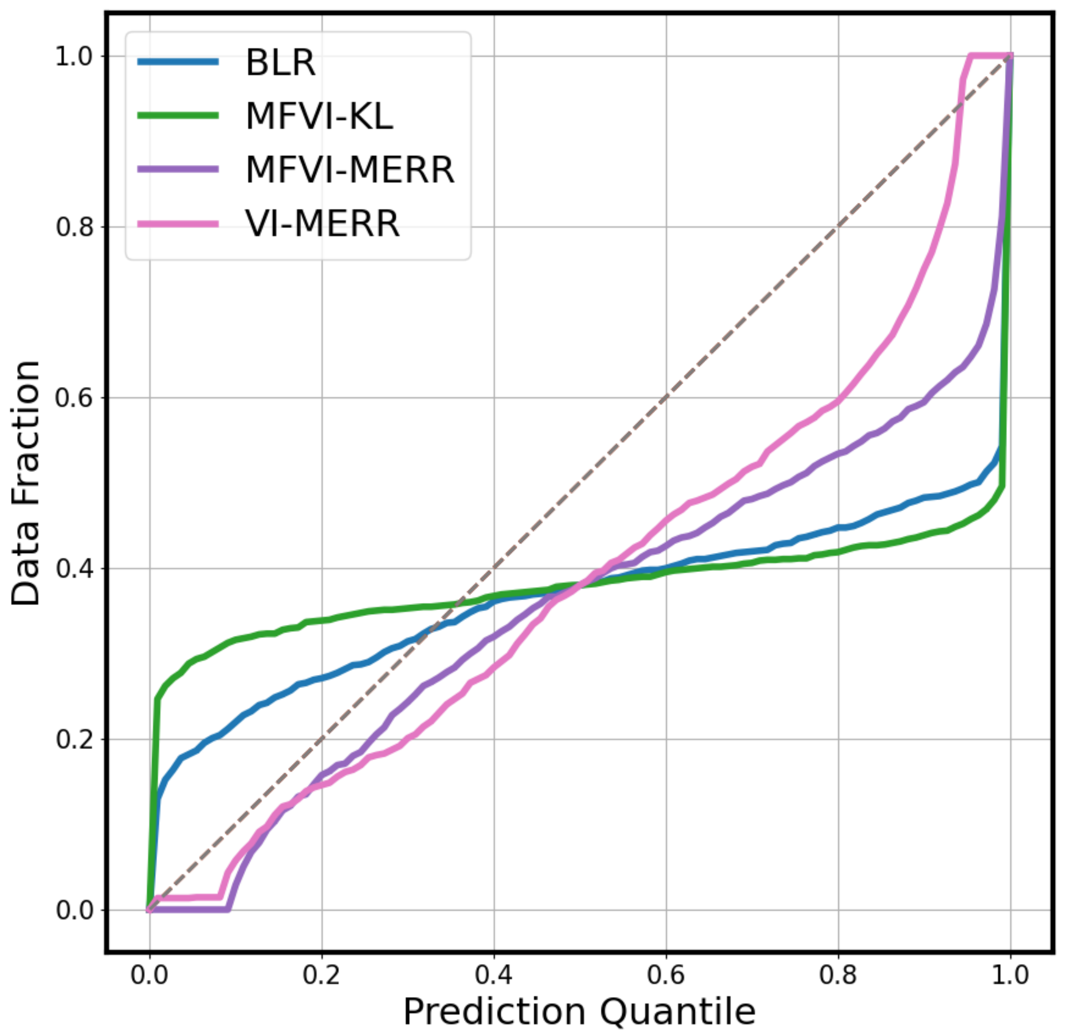
[9] A. Hegde, E. Weiss, W. Windl, H.N. Najm, C. Safta, “A Bayesian Calibration Framework with Embedded Model Error for Model Diagnostics”, International Journal for Uncertainty Quantification, 2024.

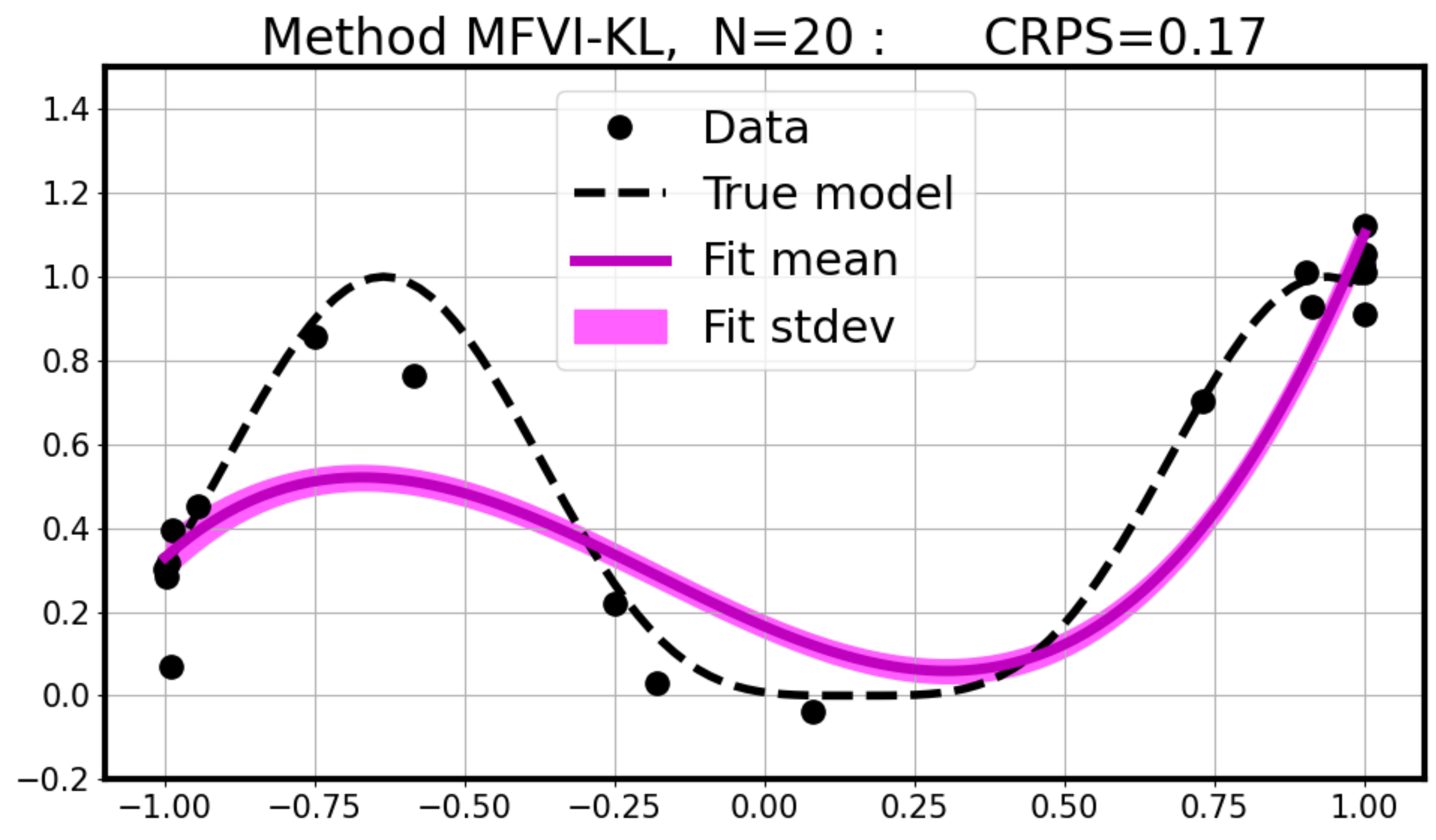
Embedded Model Error	Variational Inference
Casting physical parameters λ as r.v. in PC family $\lambda = \sum_k \alpha_k \psi_k(\xi)$	Posterior $p(\lambda \mid D)$ is apprx. in a variational family $q_\alpha(\lambda)$
Infer α Bayesian/MCMC	Infer α Optimization/SGD
Uses ABC/Moment matching in the outputs	KL distance of inputs
Needs PC-based propagation	No need for uncertainty propagation

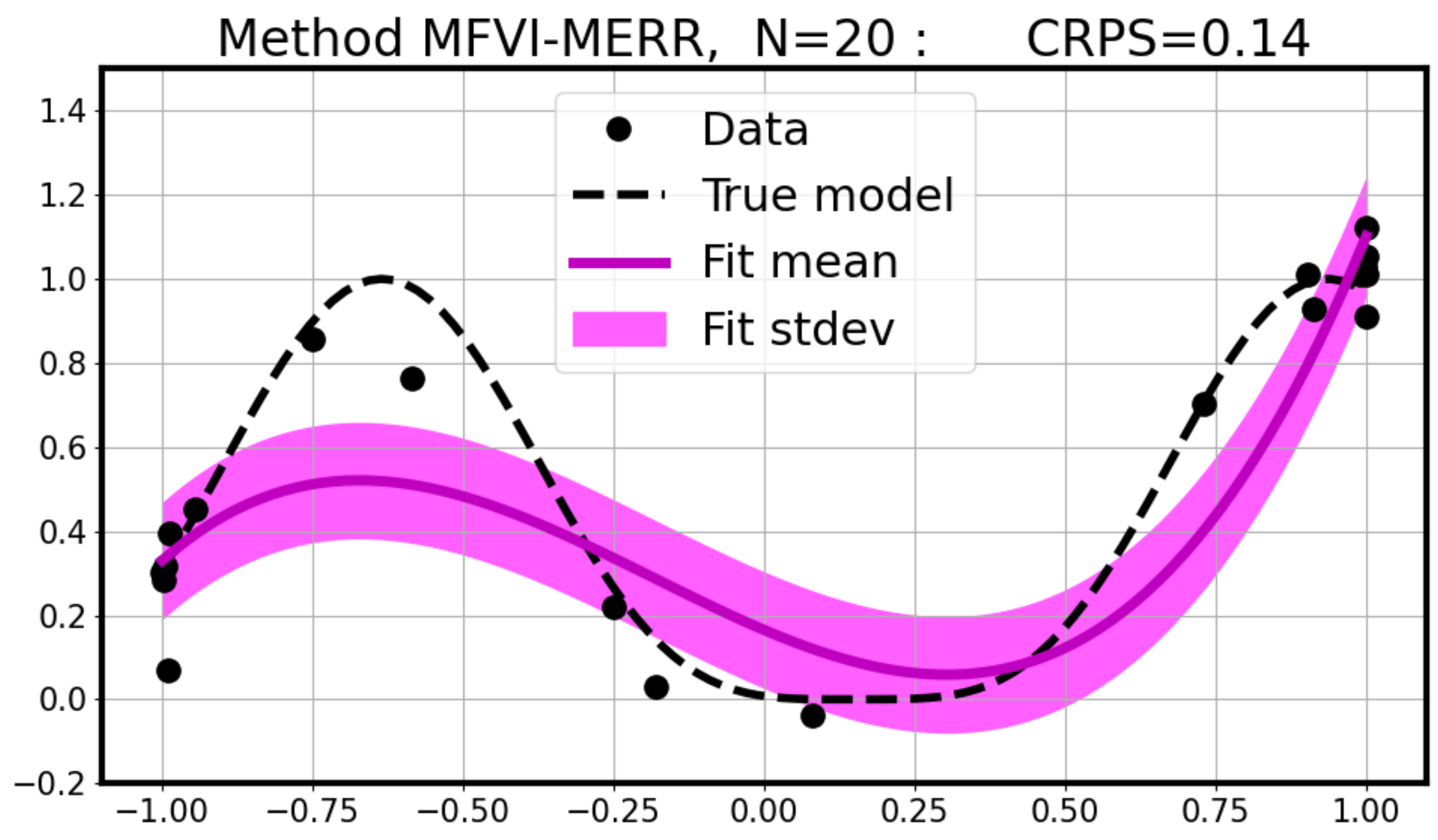
K. Sargsyan (ksargsy@sandia.gov)

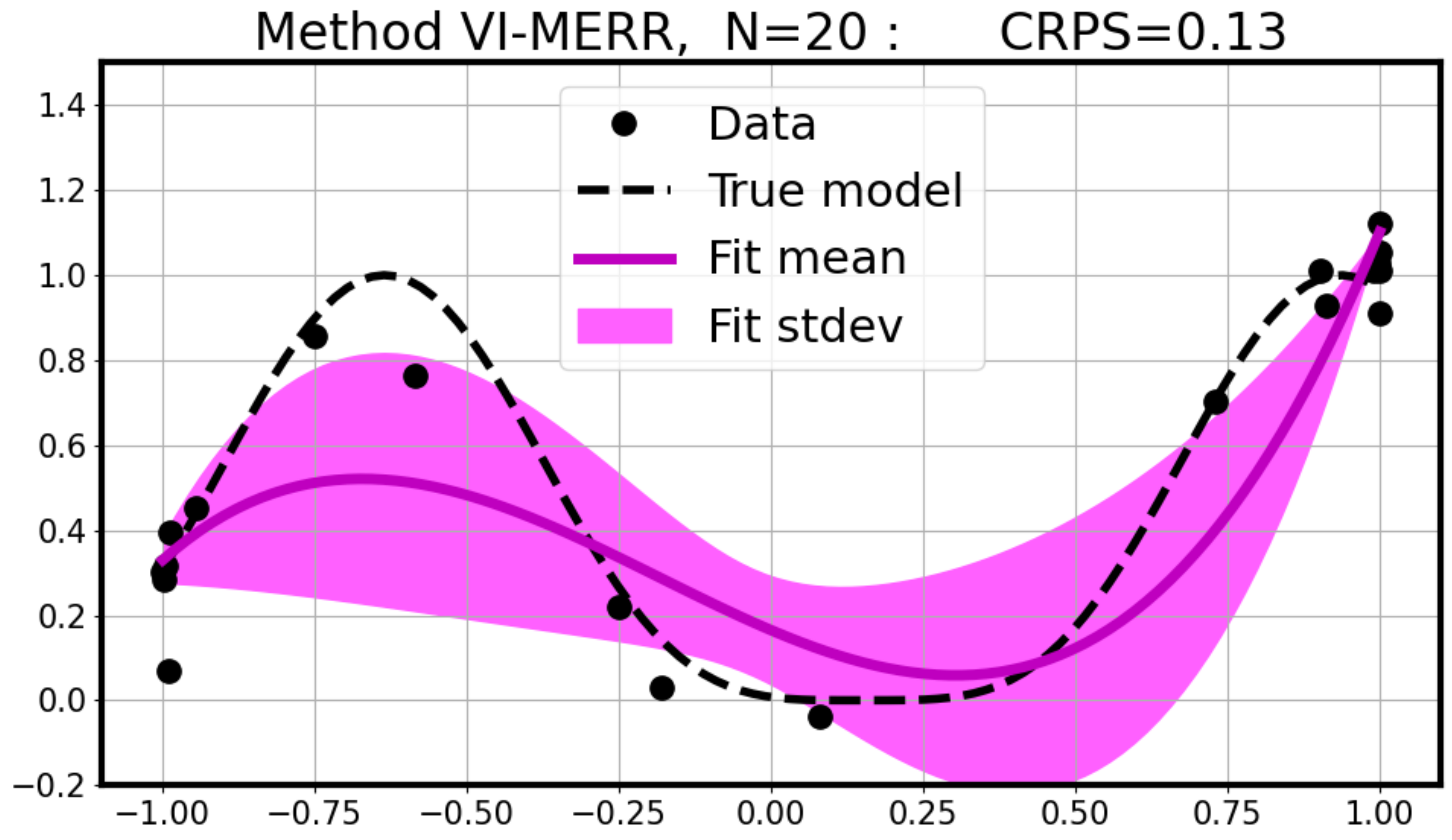
N=20, o=3





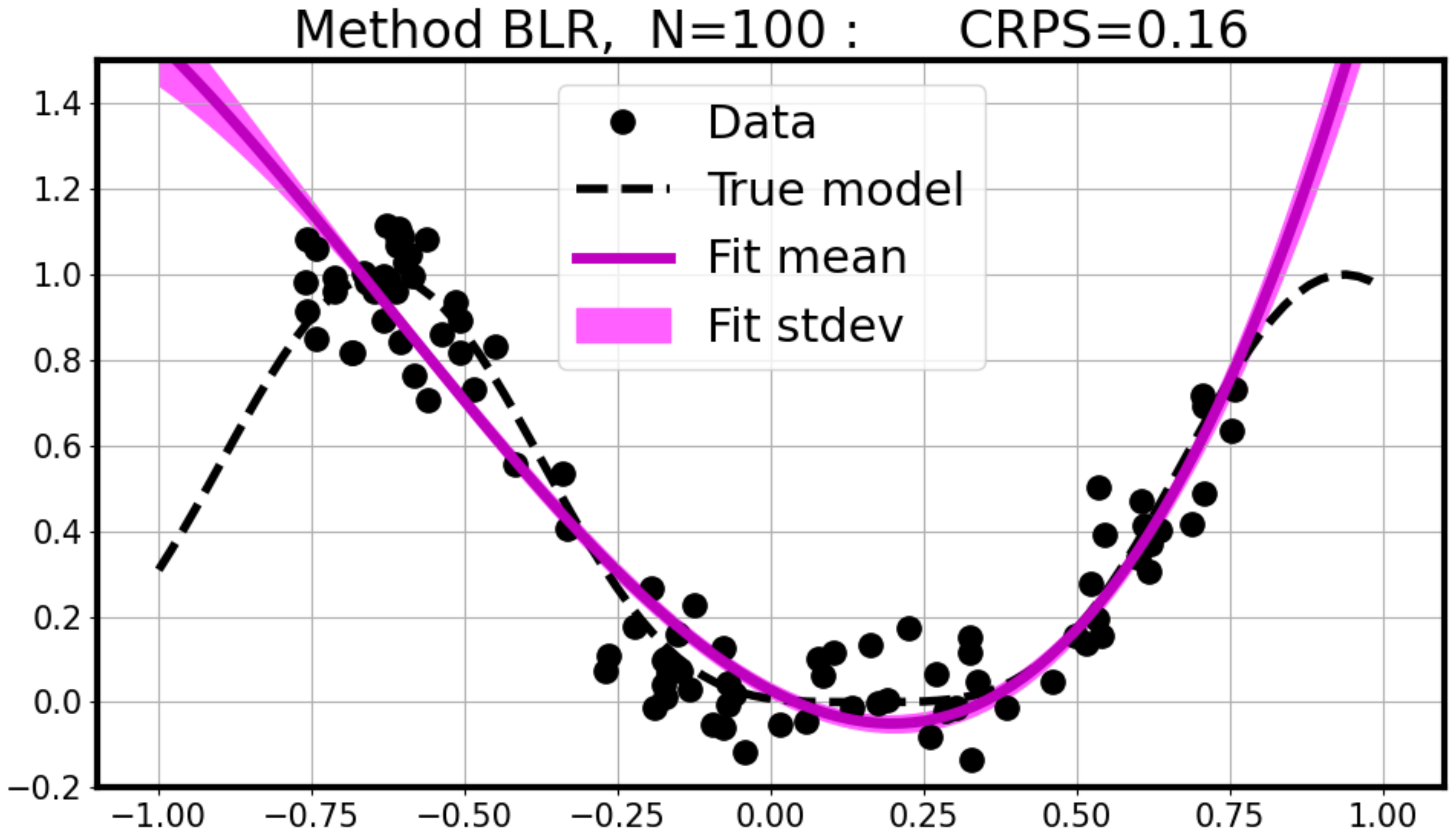


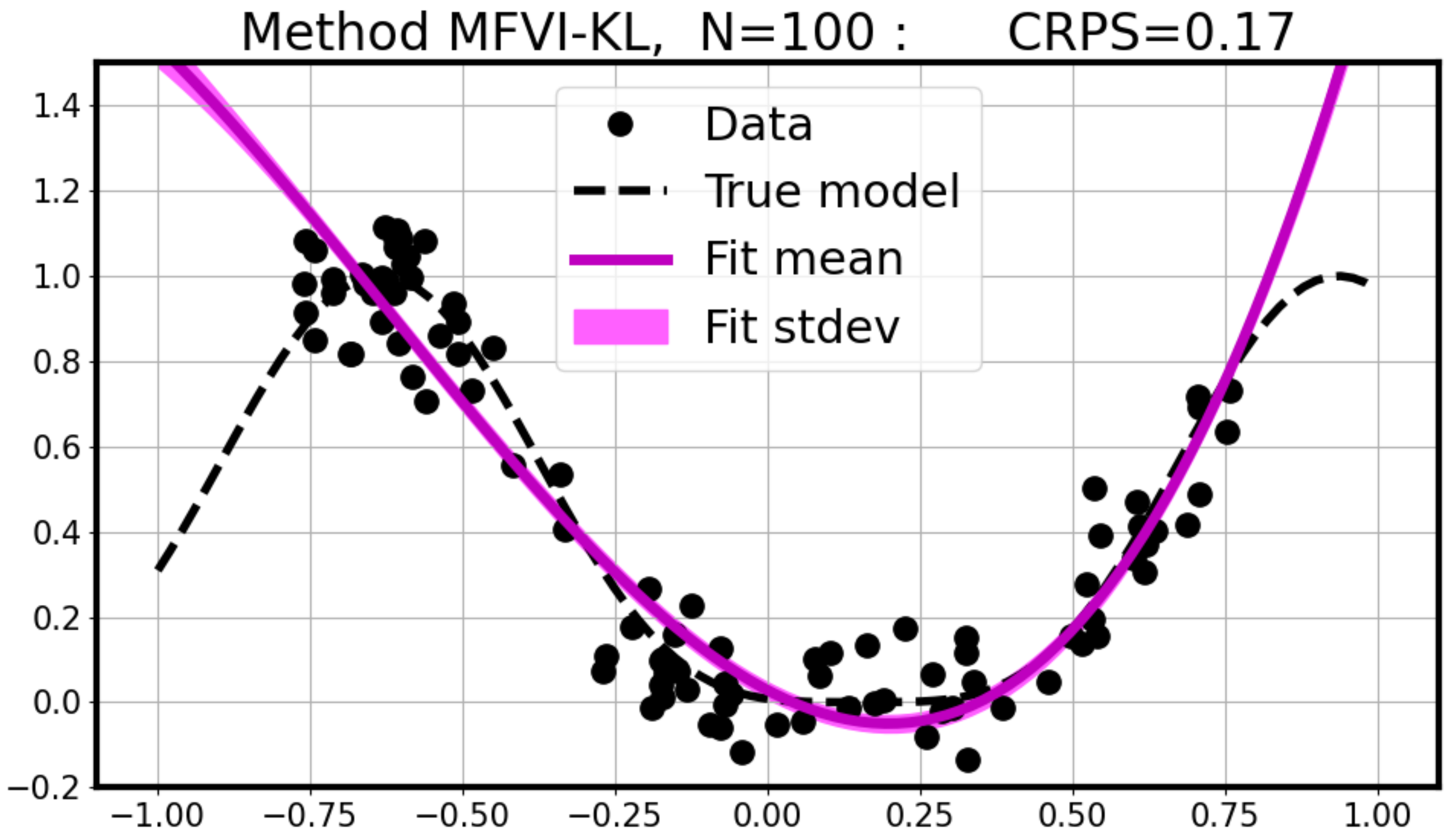


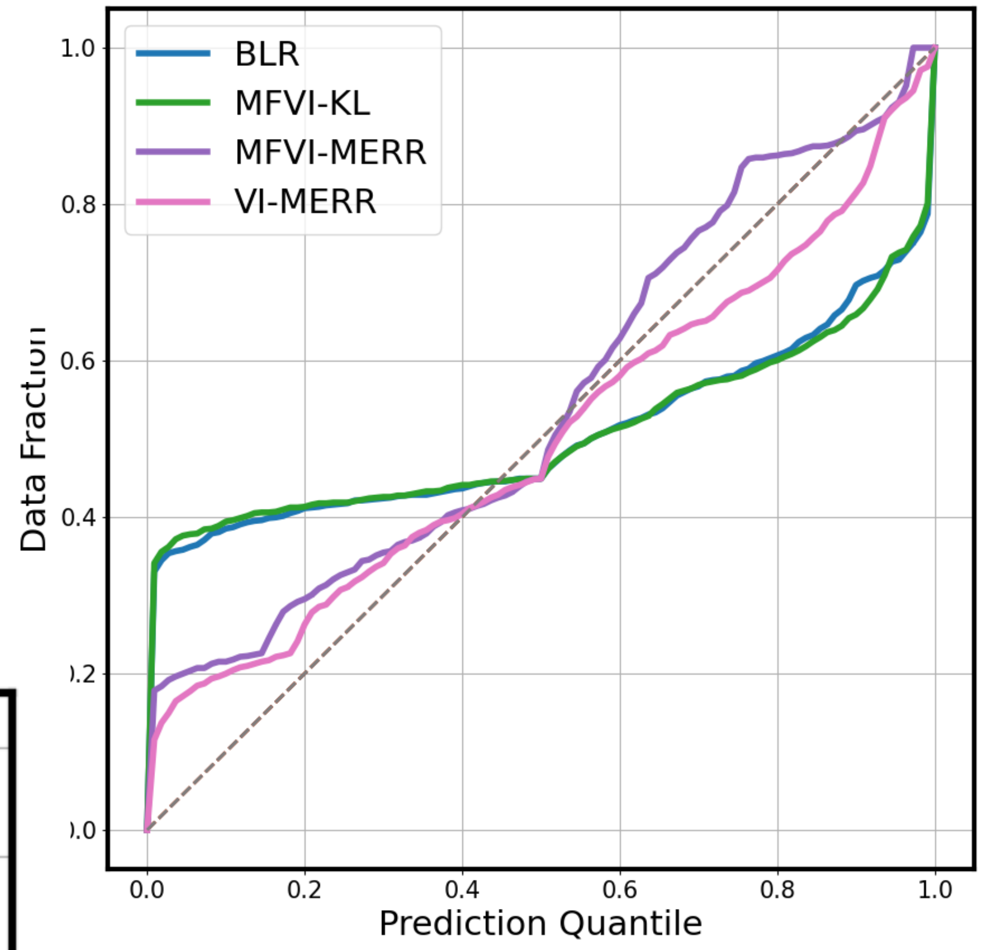


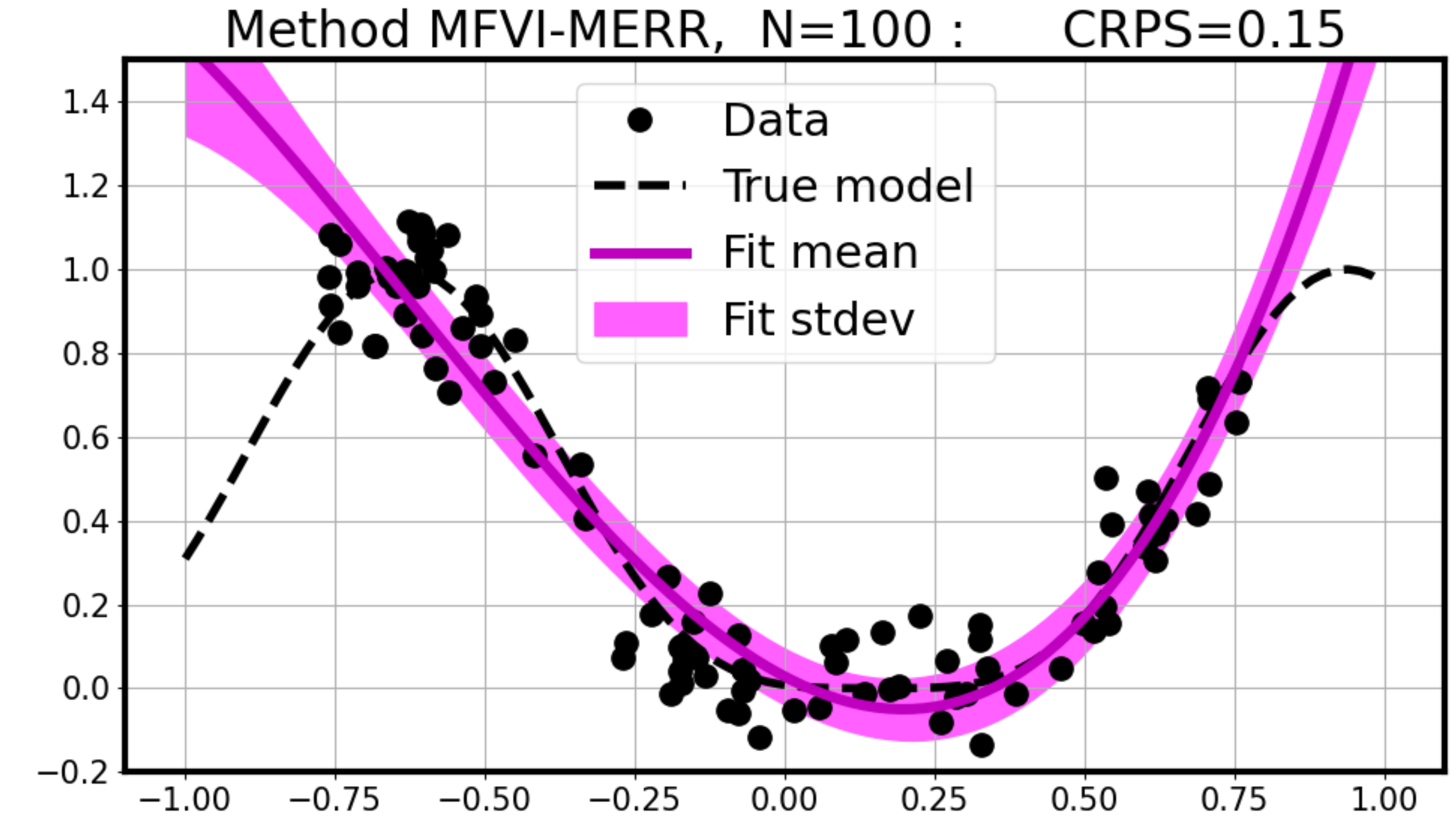
K. Sargsyan (ksargsy@sandia.gov)

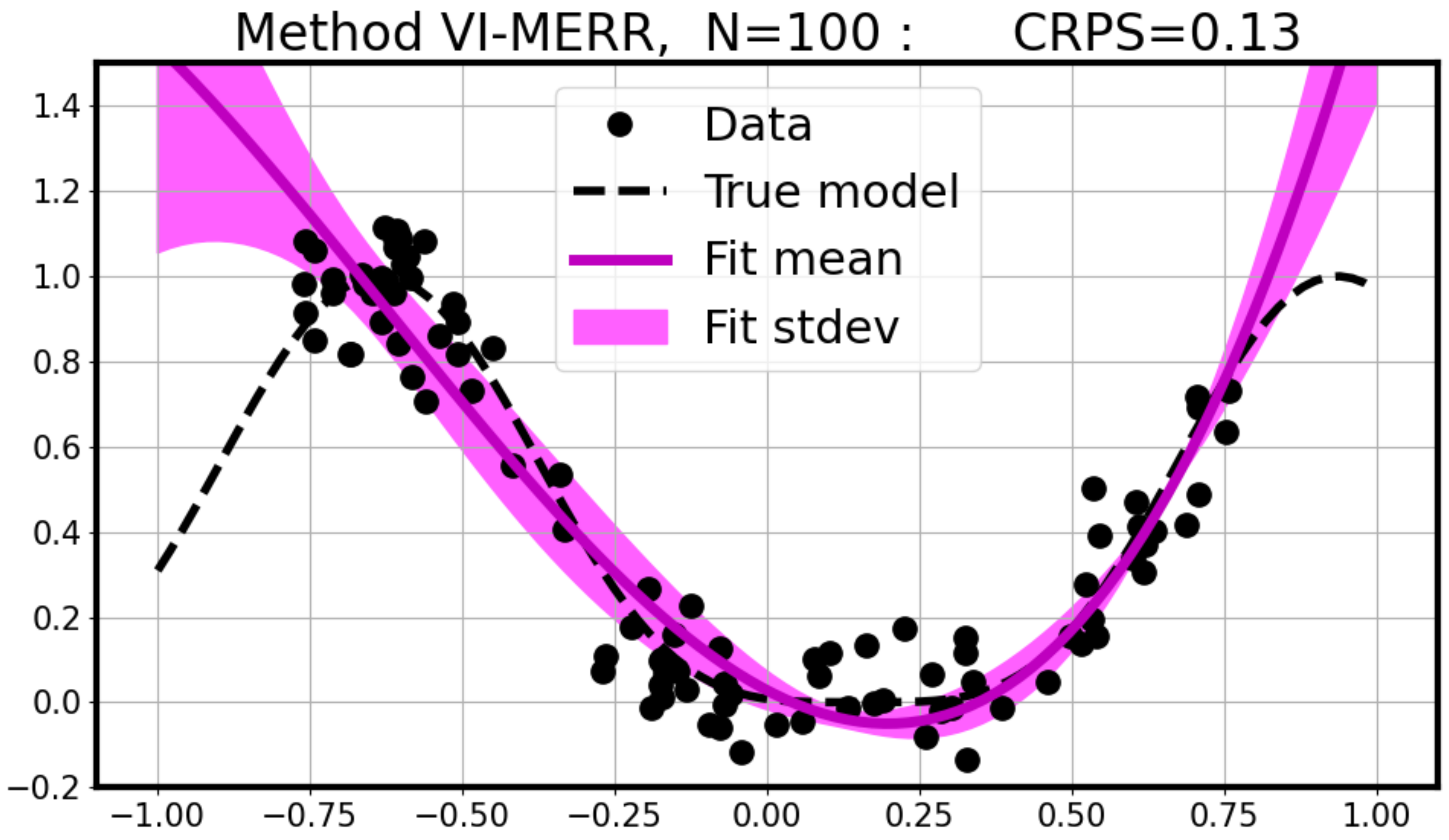
N=100, o=3


Method BLR, N=100 : CRPS=0.16
Data
True model
Fit mean
Fit stdev


Method MFVI-KL, N=100 : CRPS=0.17
Data
True model
Fit mean
Fit stdev


BLR
MFVI-KL
MFVI-MERR
VI-MERR
Prediction Quantile

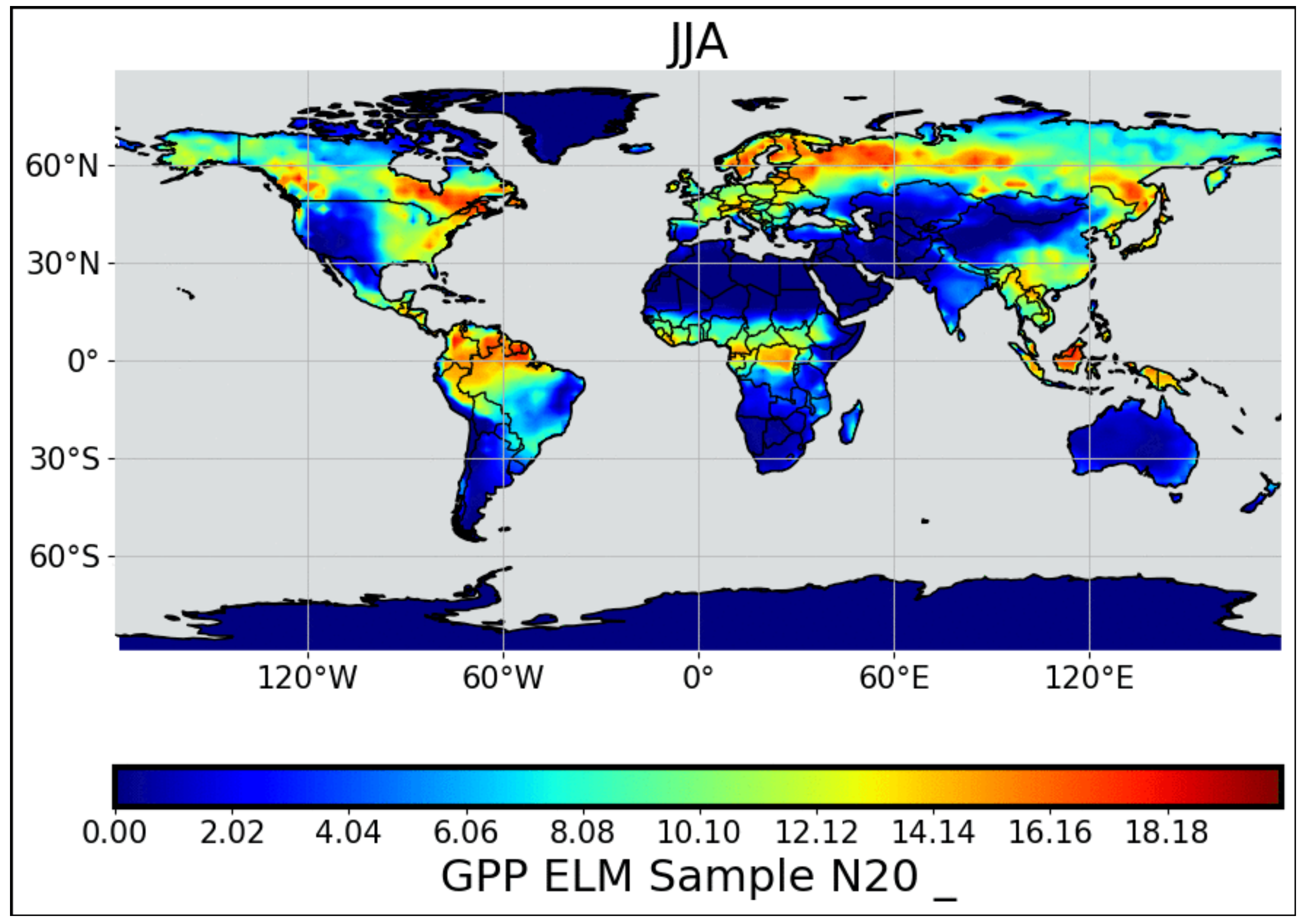

Method MFVI-MERR, N=100 : CRPS=0.15
Data
True model
Fit mean
Fit stdev

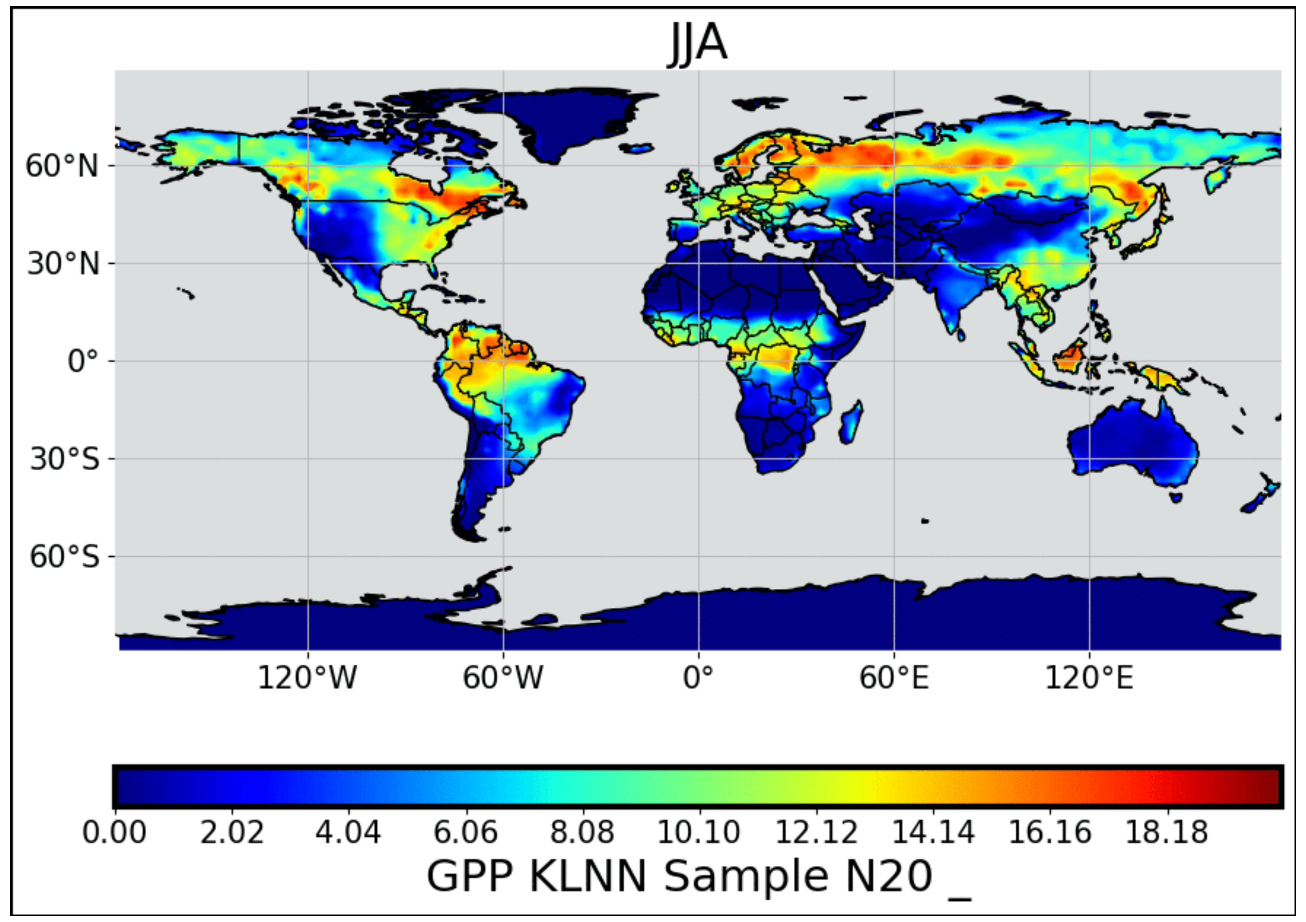

Method VI-MERR, N=100 : CRPS=0.13
Data
True model
Fit mean
Fit stdev

Application Examples

Reduced dim. surrogate-enabled calibration

E3SM Land Model UQ



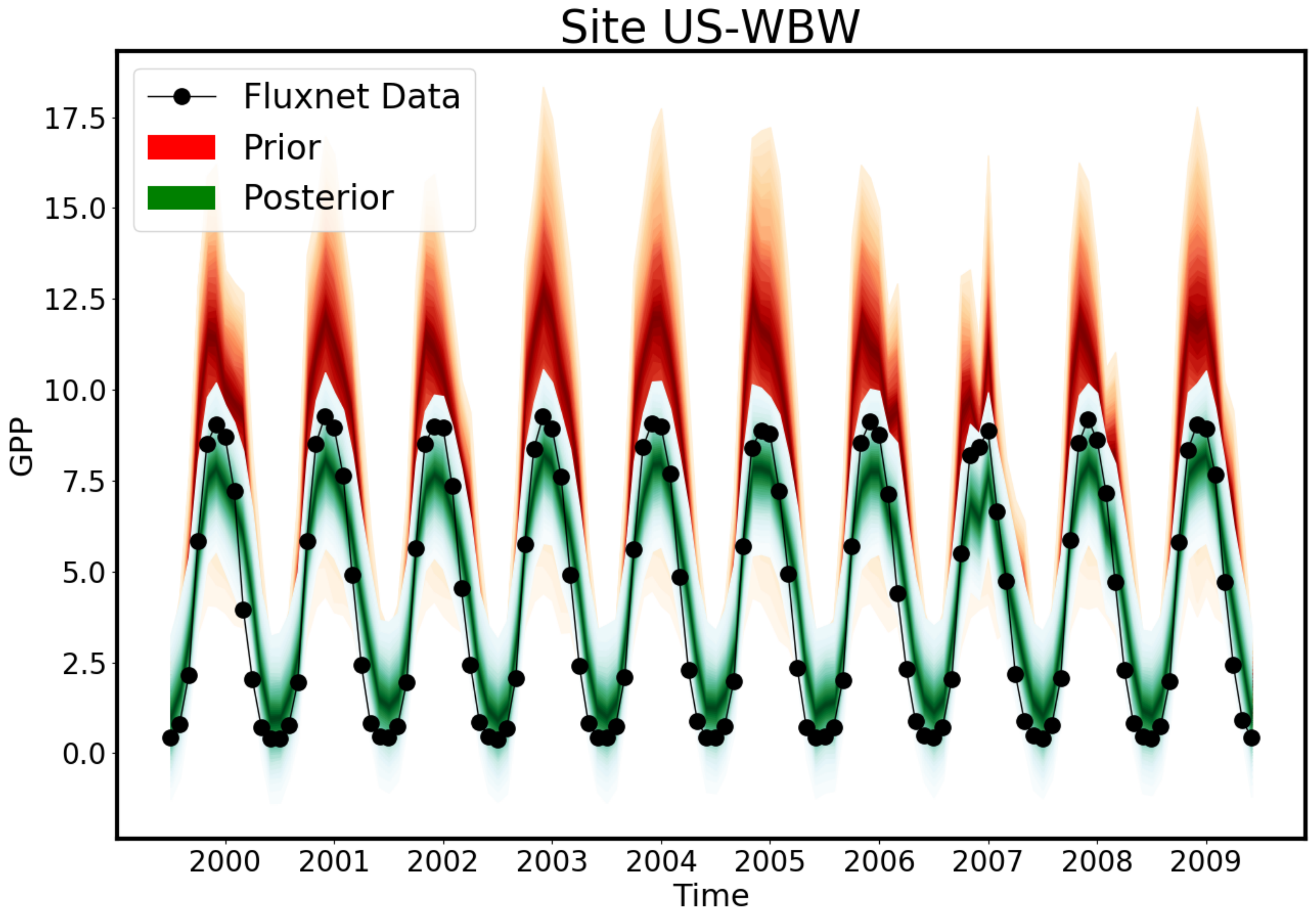


KL+NN spatiotemporal surrogate, 10 params

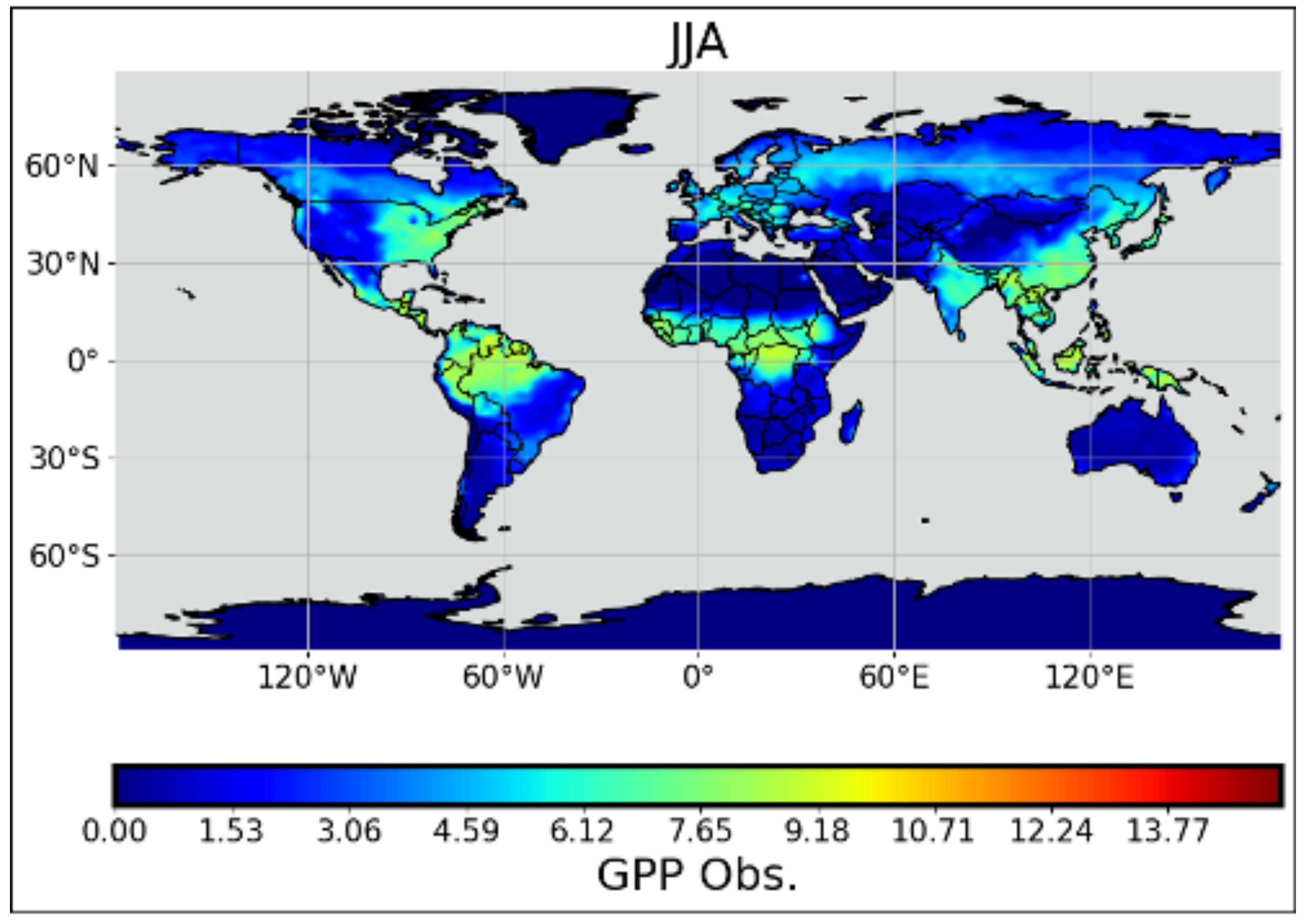
E3SM
Energy Exascale
Earth System Model

Spatial slice of calibrated model

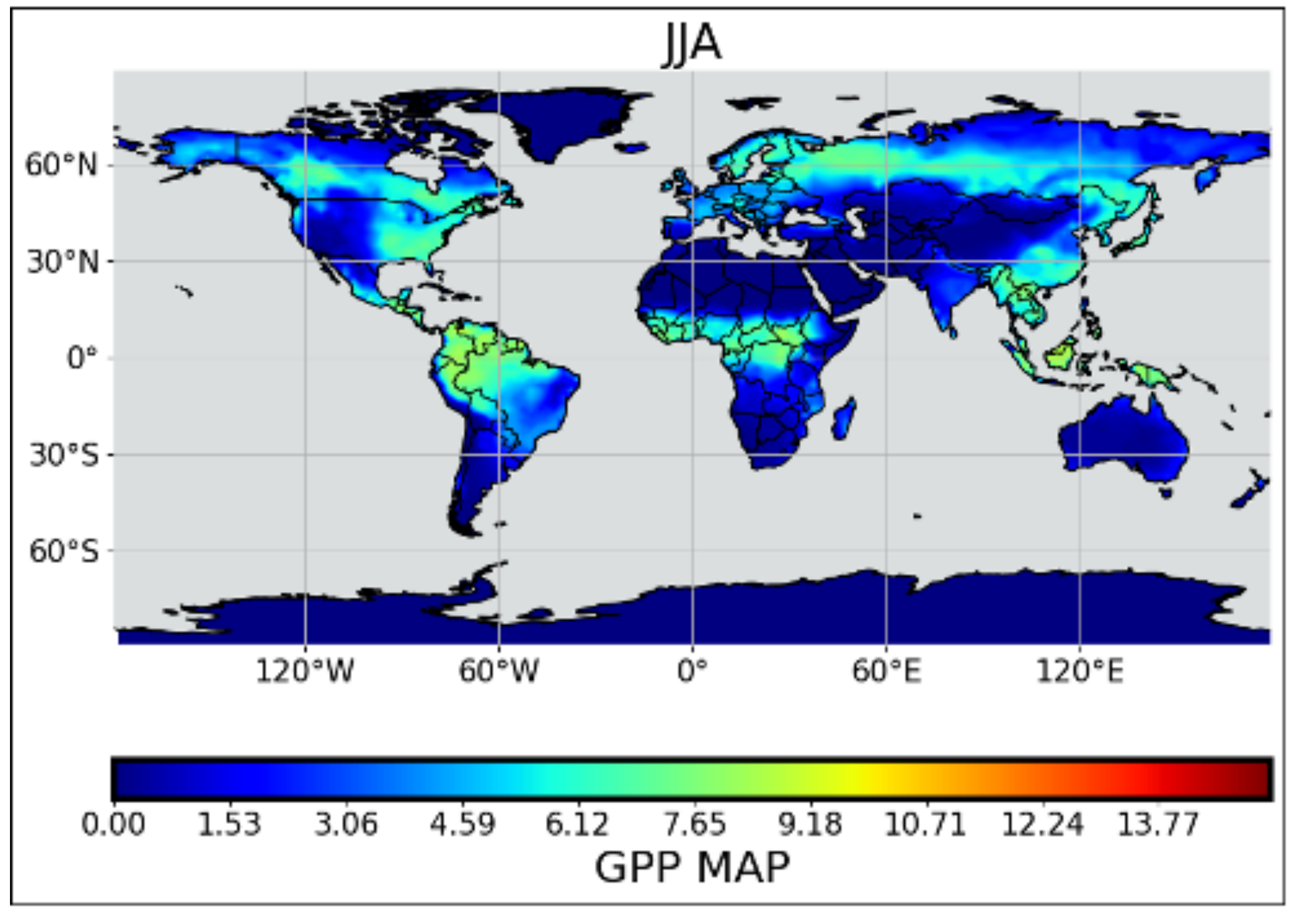
Temporal slice of calibrated model



FLUXCOM Data



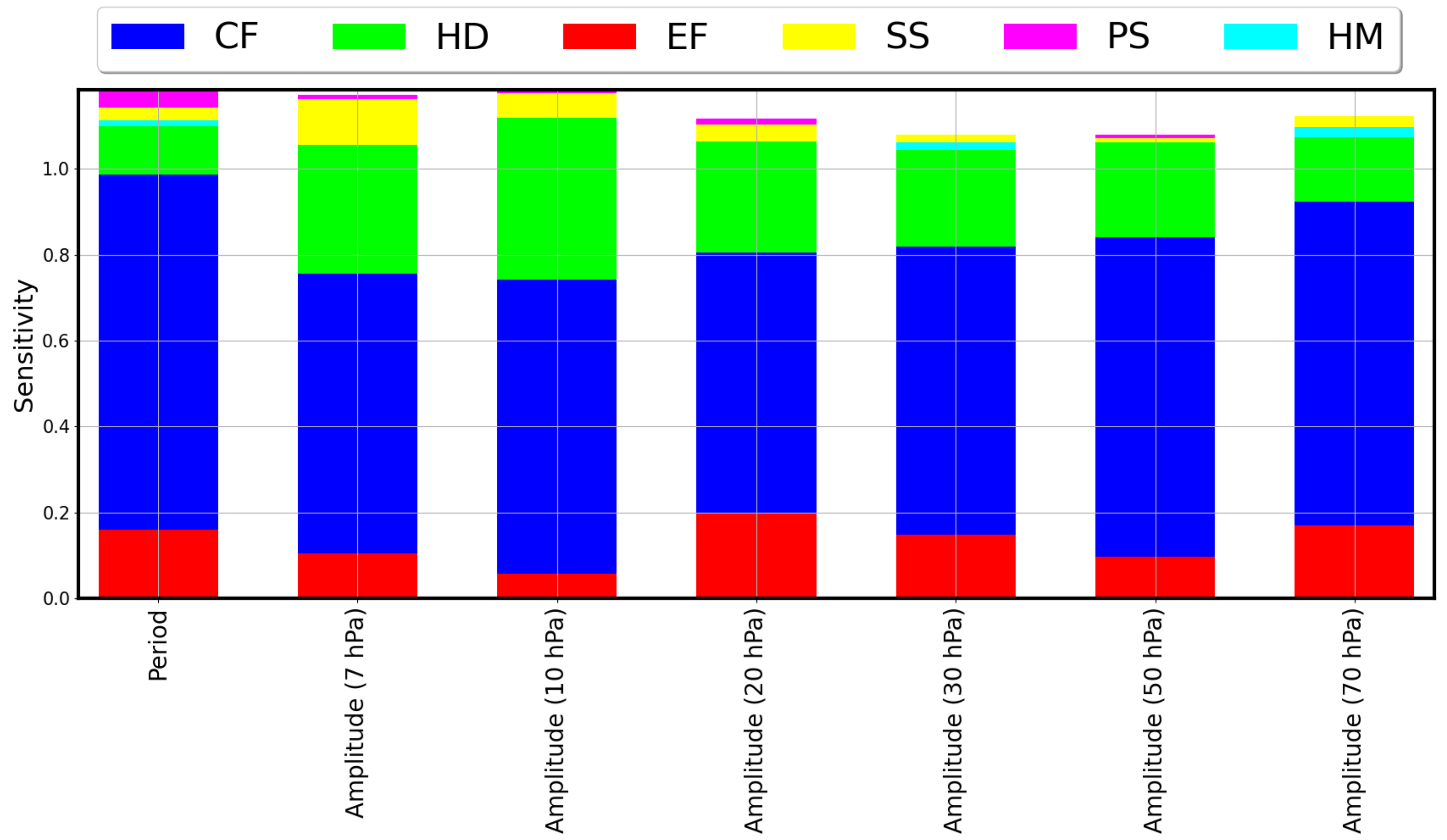
Calibrated Model



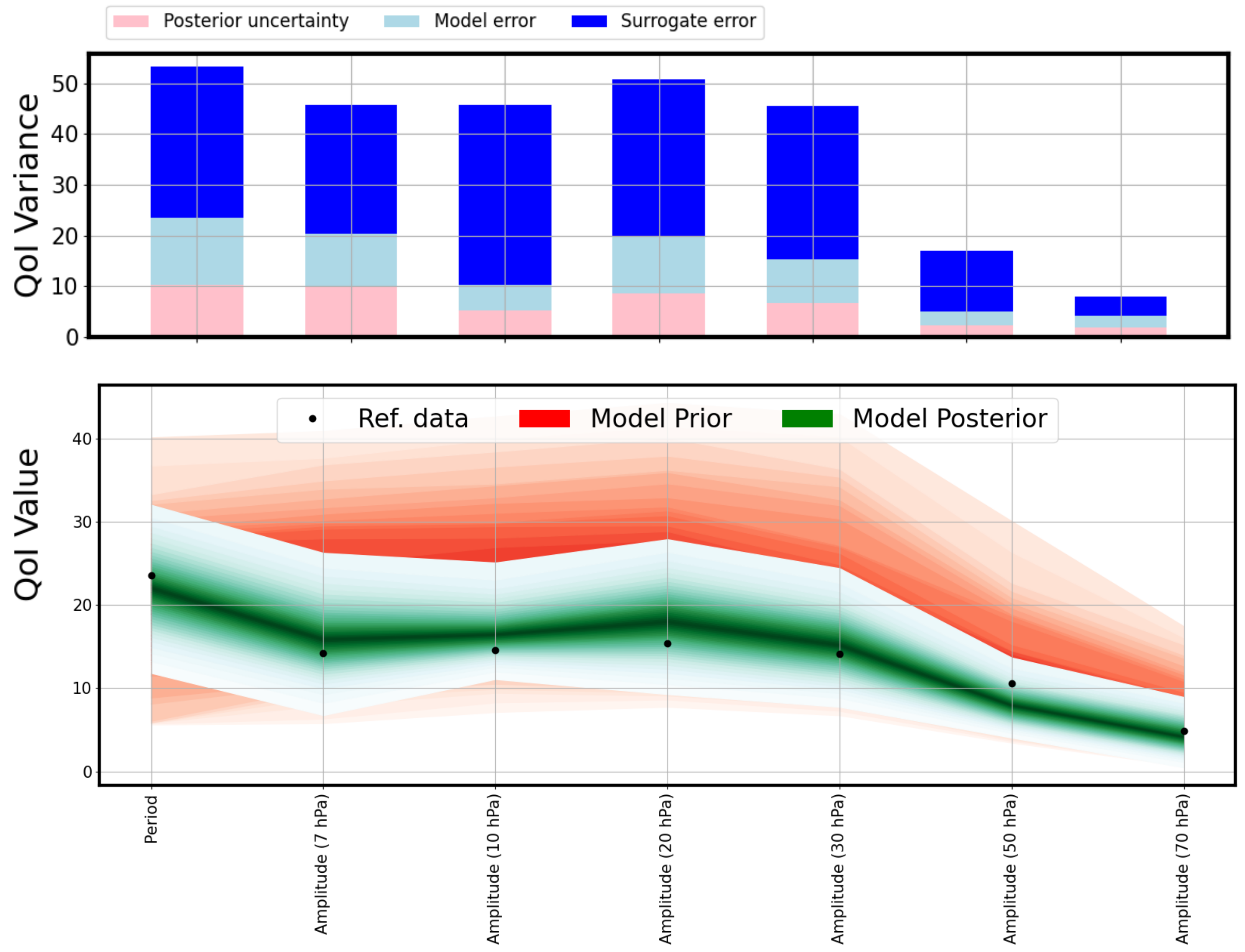
Model error estimation

E3SM: Quasi-biennial Oscillation

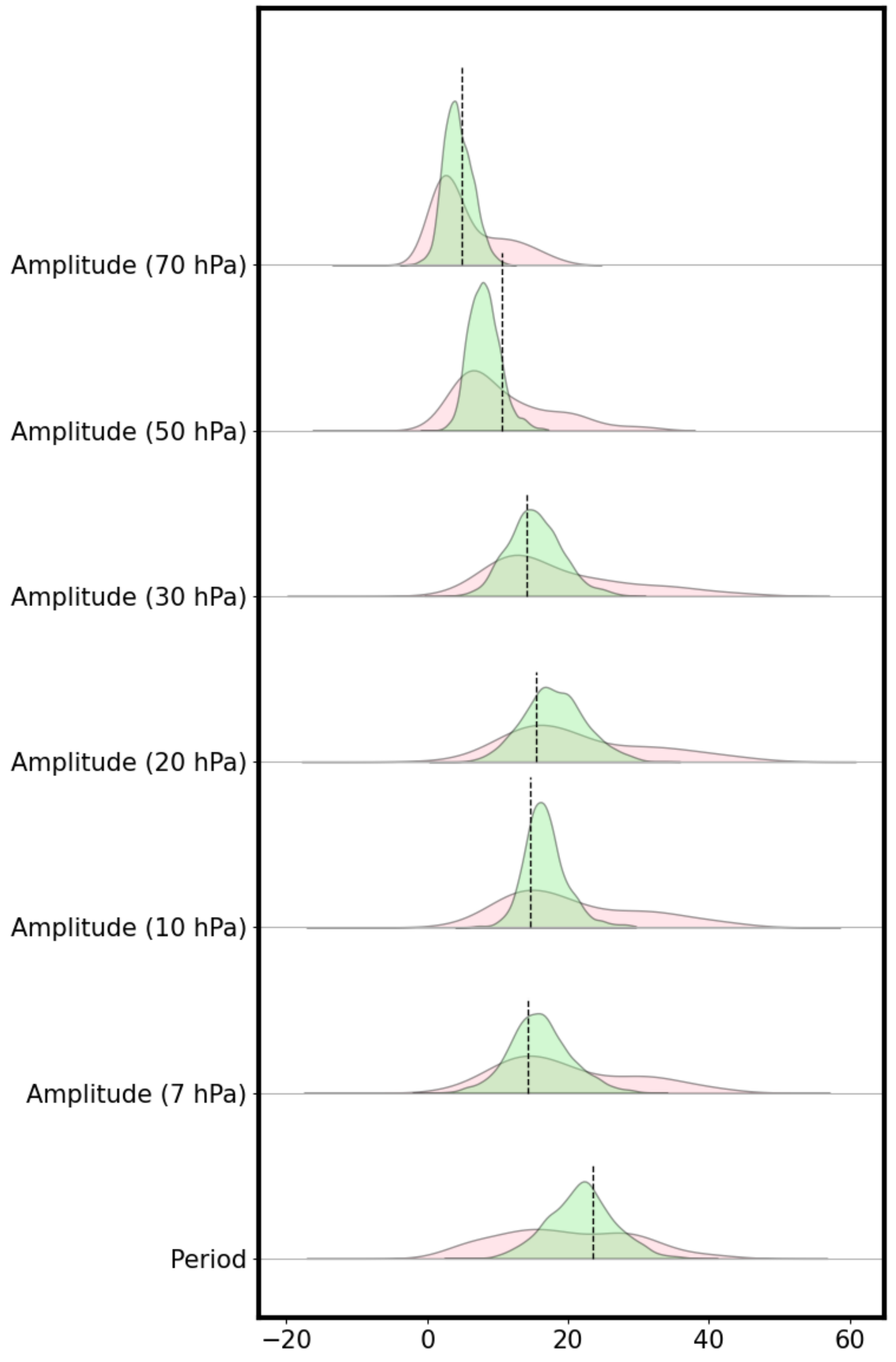
6-param perturbed ensemble, PC surrogate + GSA

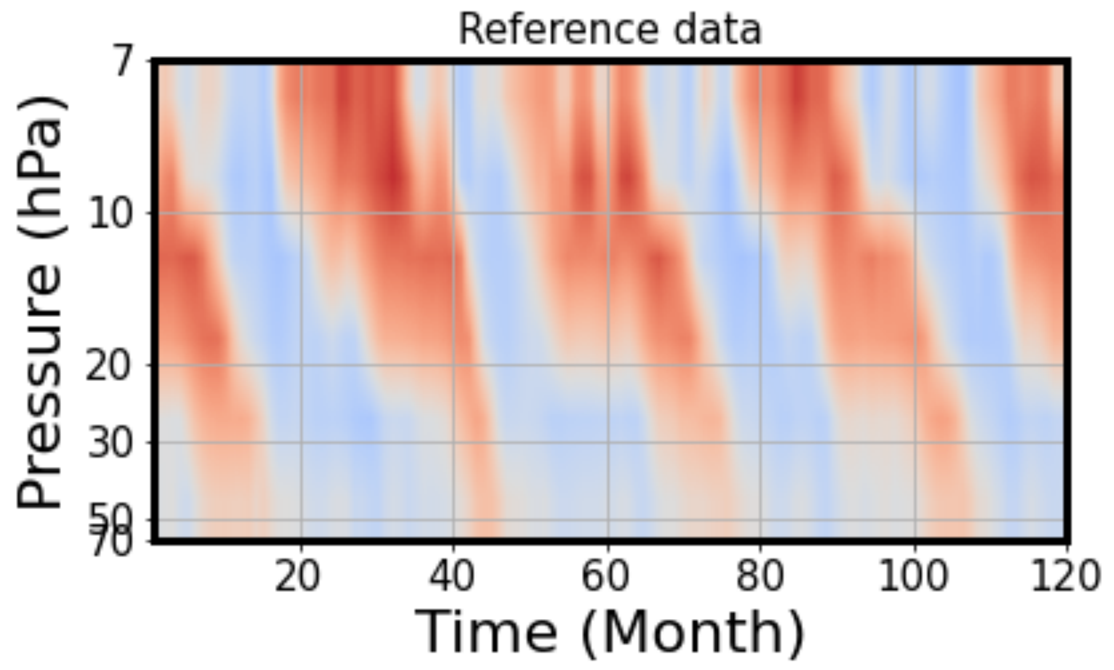


Calibration with model error and uncertainty attribution



Marginal posterior predictives



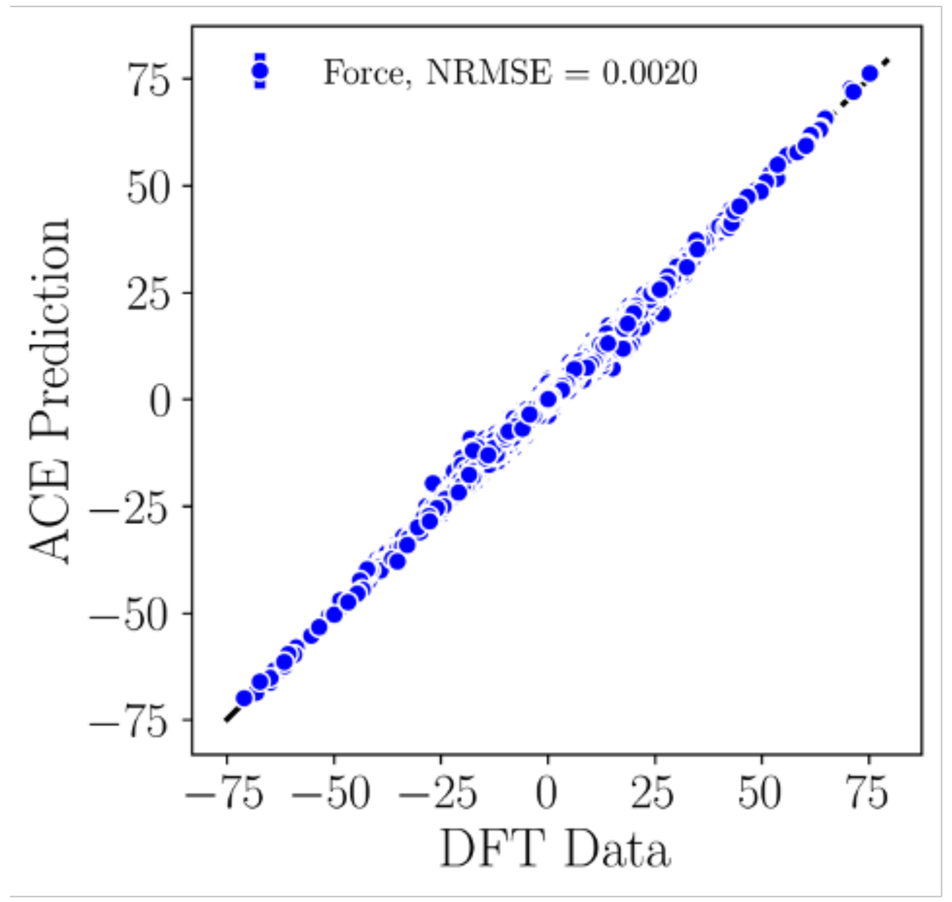


KL dimensionality reduction of wind fields, KL+PC surrogate to enable calibration

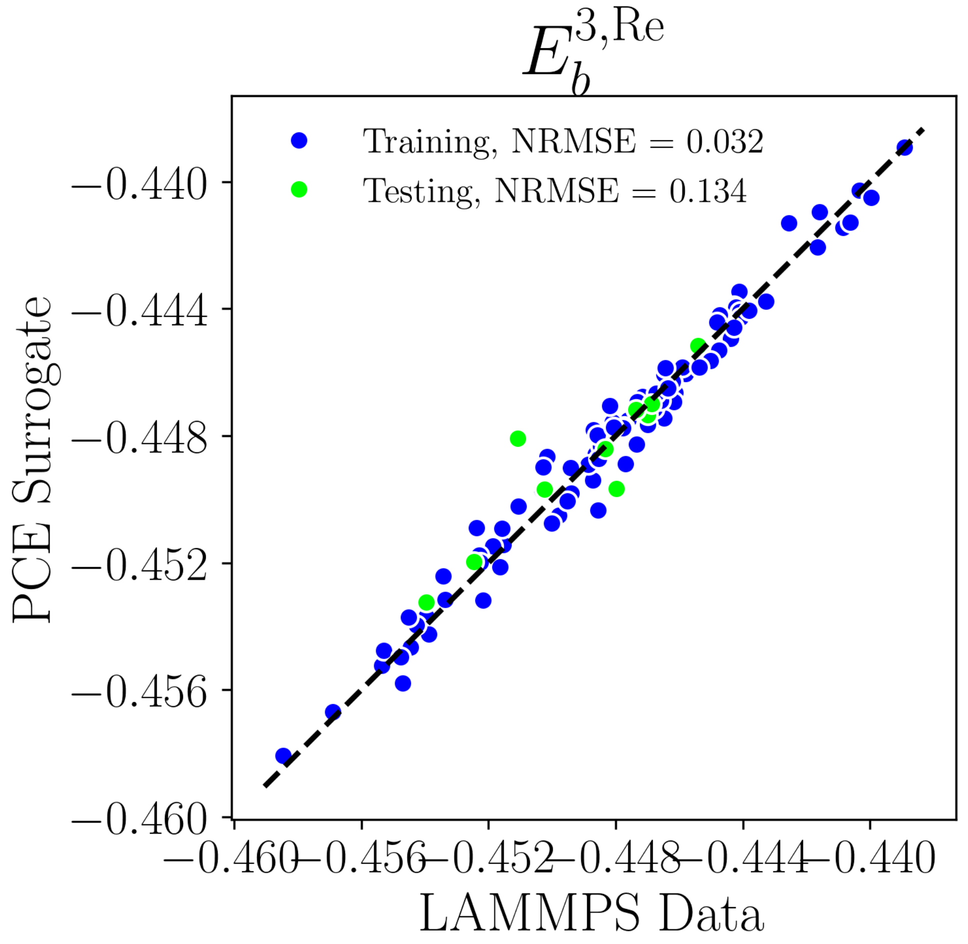
Energy Exascale Earth System Model

MLIAP Uncertainty estimation

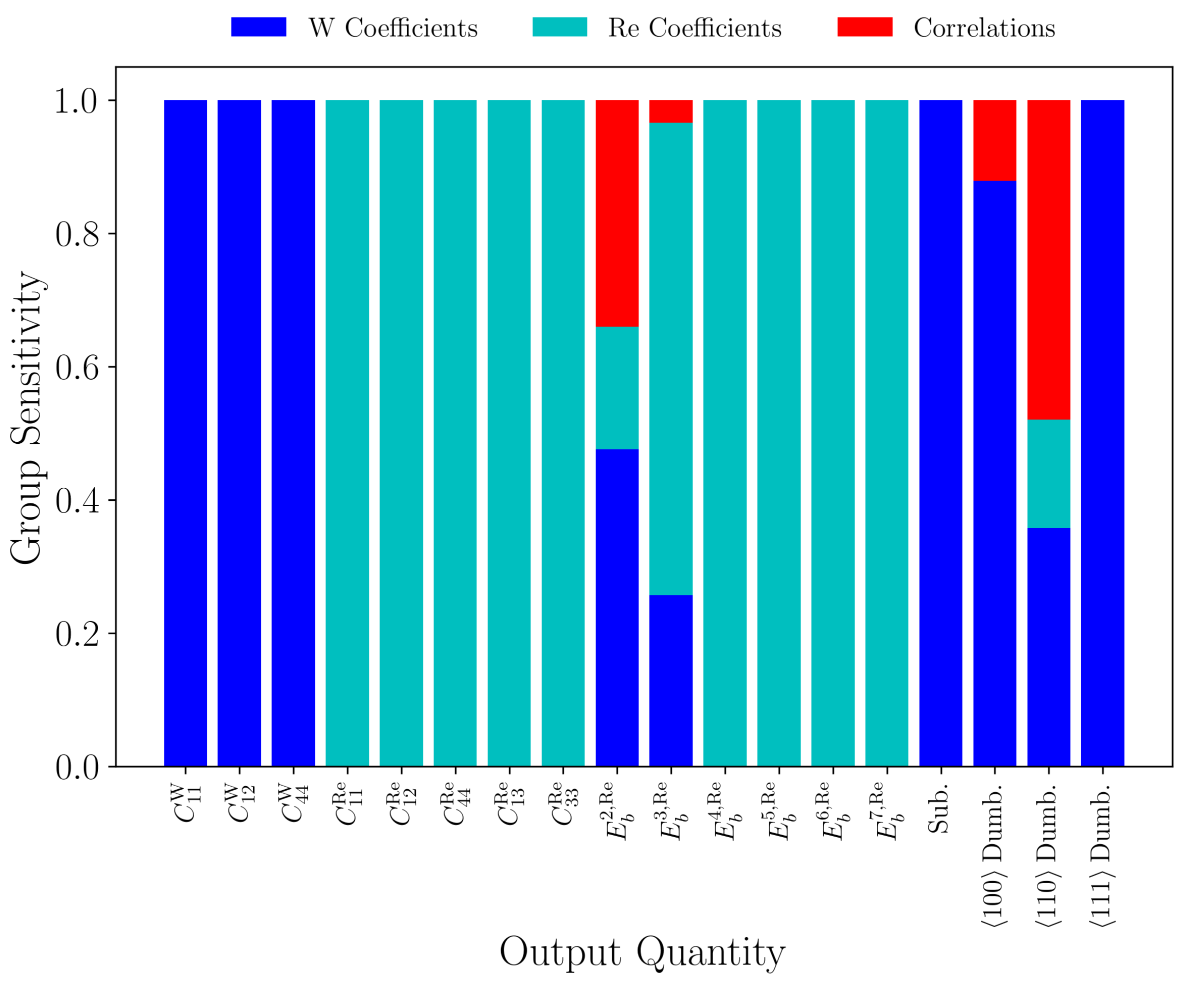
Bayesian linear regression (BLR) to fit, with UQ, interatomic potential to DFT data



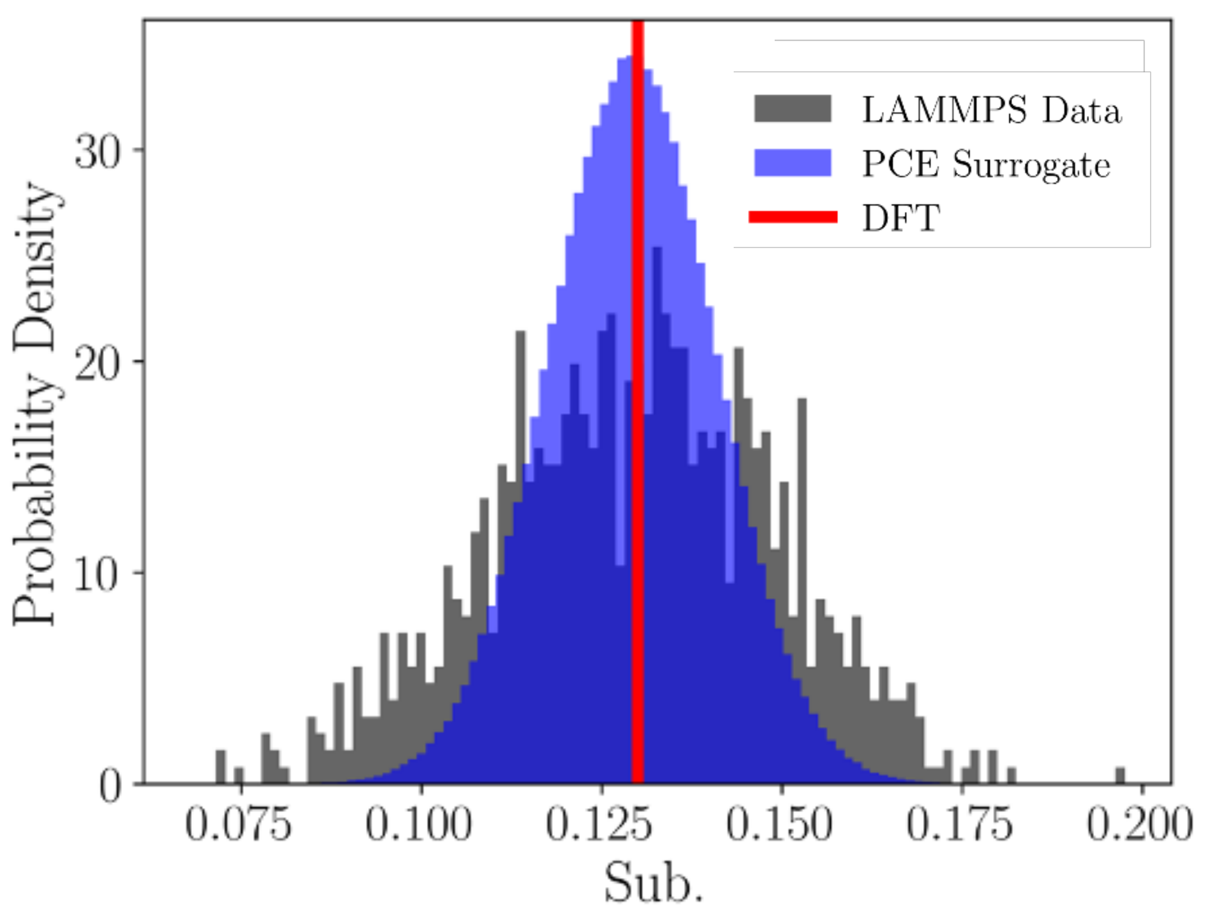
PC-based propagation of uncertainty through MD QoIs



GSA and uncertainty attribution

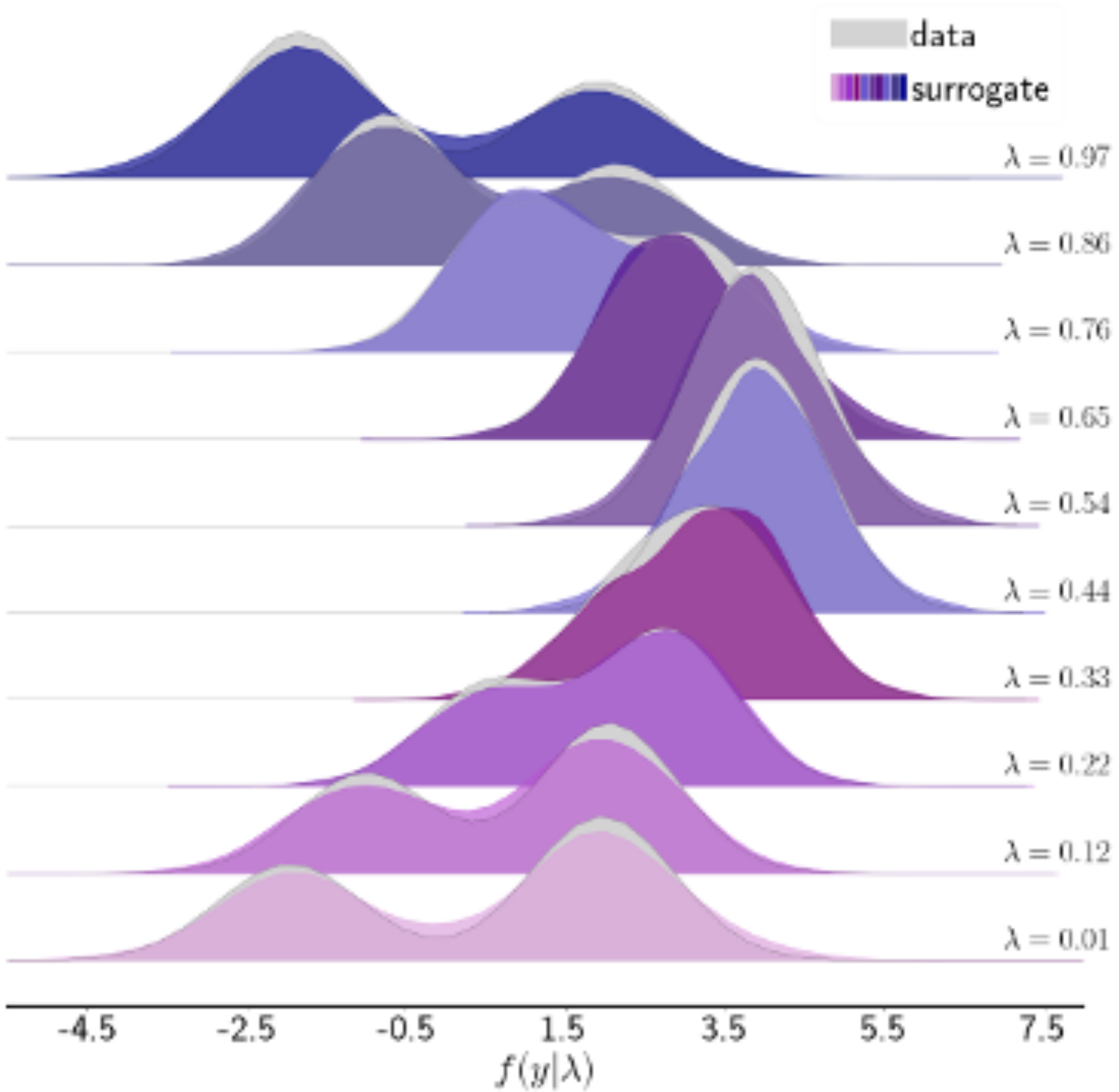


Validation with DFT data



Figs courtesy of Cristian Lacey

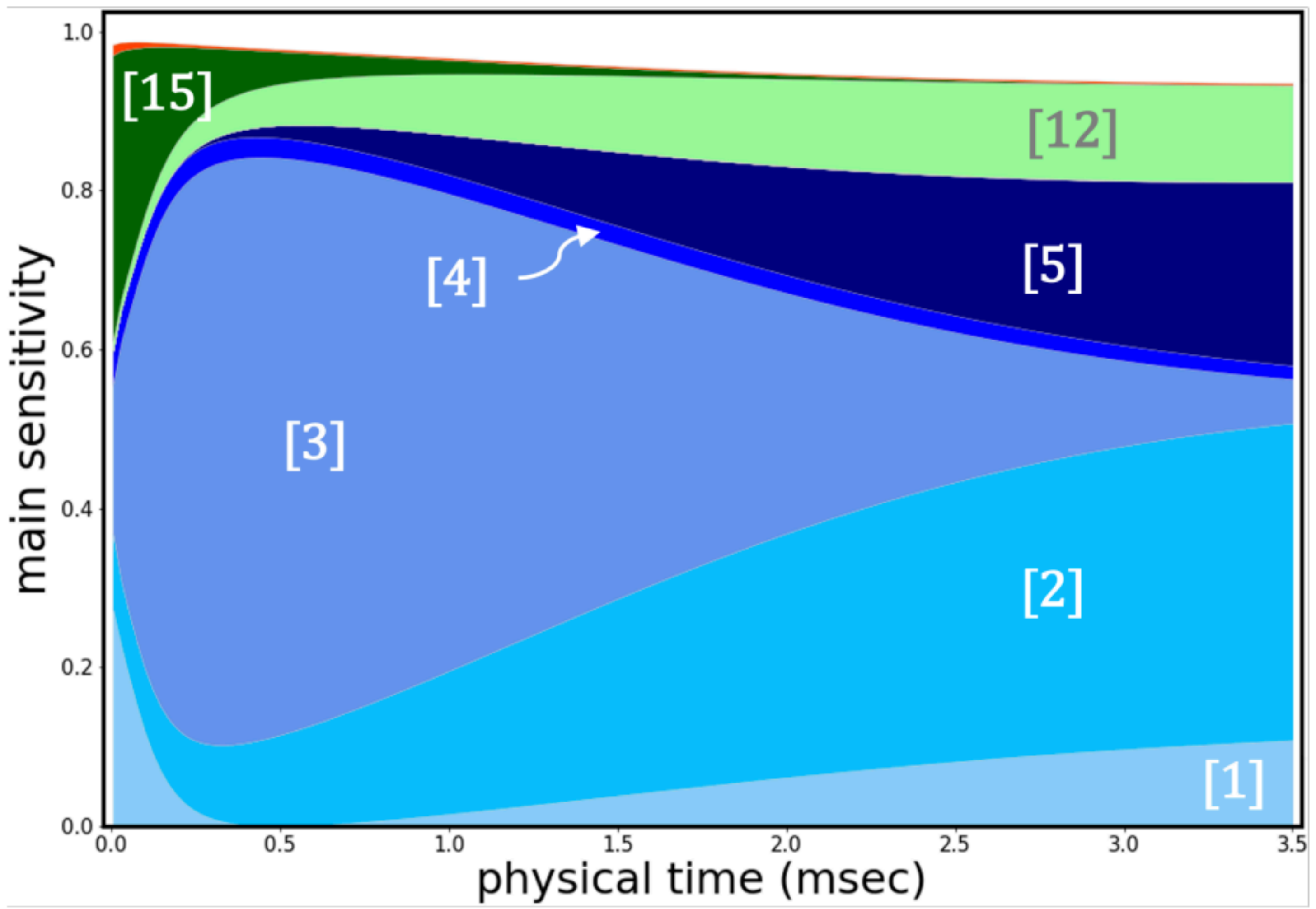
Parametric-Stochastic Surrogate



Rosenblatt transformation merged with parametric fit to build a joint stochastic-parametric PC surrogate

Karhunen-Loève expansion to extend it to random fields

Global sensitivity analysis of parameteric and stochastic uncertainties wrt time-dependent QoIs



Figs courtesy of Joy Bahr-Mueller

Summary and Future

- Lightweight, python-only UQ library
- Standard methods as well as advanced method development
- OOP allows extending with new methods via inherited wrappers
- Part of SciDAC FASTMath software catalogue
- Serves a large set of DOE applications of interest

Coming up:

Math documentation/manual

Usage tutorials

Weighted BCS

Intrusive PC arithmetics

Sparse quadrature

Low-rank tensors

Trunk

Examples

Examples of varying complexity

- *ex_evidence.py*
- *ex_func*.py*
- *ex_integrate.py*
- *ex_genz1d.py*
- *ex_mindex*.py*
- ⋮
- *ex_optim.py*
- *ex_mixture.py*
- *ex_quad.py*
- *ex_sampling.py*
- *ex_slice.py*
- ⋮

- Meant to be run out of the box
- Usually refer to one of these if requests come
- … or write a quick one per request
- Produce screen outputs, and *.png* (and some *.txt*) files
- Script to run all examples together

run_examples.sh

```bash
#!/bin/bash -e

mkdir -p runex
cd runex

SCRDIR=$(dirname "$0")


echo $SCRDIR
for ex_scr in "$SCRDIR"/../examples/*py; do
    echo "========================================"
    start=$SECONDS
    ex_scr_name=$(basename "$ex_scr" .py)
    echo "Running  $ex_scr"
    "$ex_scr" > ${ex_scr_name}.log
    duration=$(( SECONDS - start ))
    echo "It took $duration sec."
    #echo "========================================"
done
```

High-d Output: Karhunen-Loève (KL) expansion

$$f(\lambda; z) \approx \bar{f}(z) + \sum_{m=1}^{M} \eta_m(\lambda)\sqrt{\mu_m}\phi_m(z)$$

Uncertain Parameters

"Certain" Conditions

- High-d output, i.e. $N \gg 1$ different operating conditions, e.g. spatio-temporal output $z = (x, y, t)$
- Eigen-pairs $(\mu_m; \phi_m(z))$ are found via eigensolves
- Reduces the analysis to $M \ll N$ latent-space
- Parallel to SVD, except…
    - is centralized (first subtract the mean)
    - often comes with the continuous form
    - has random variable interpretation for the latent features (aka left singular vectors) η_m

$$F_{ki} = \sum_{m=1}^{M} U_{km}\Sigma_{mm}V_{im}$$

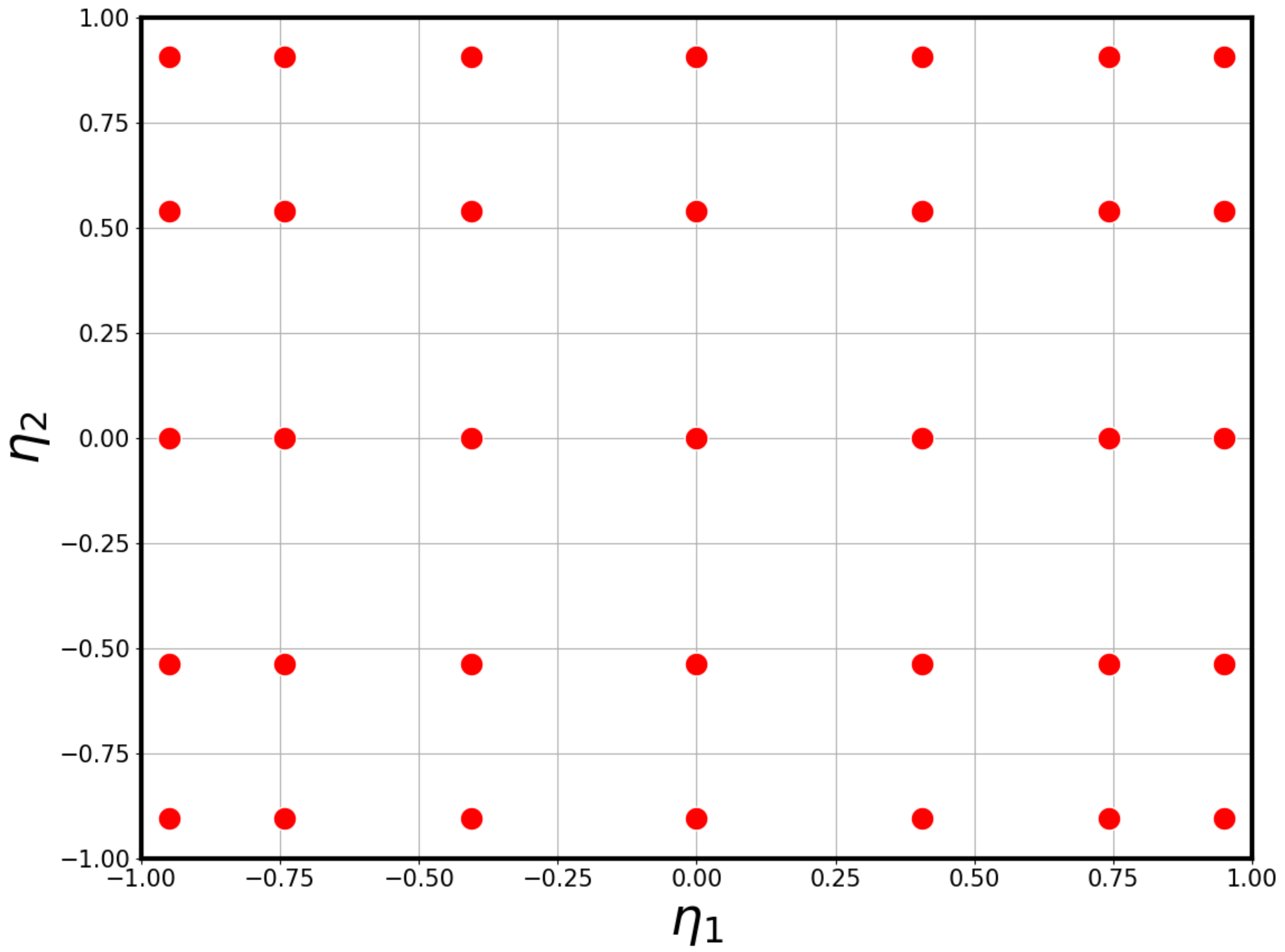
K. Sargsyan (ksargsy@sandia.gov)

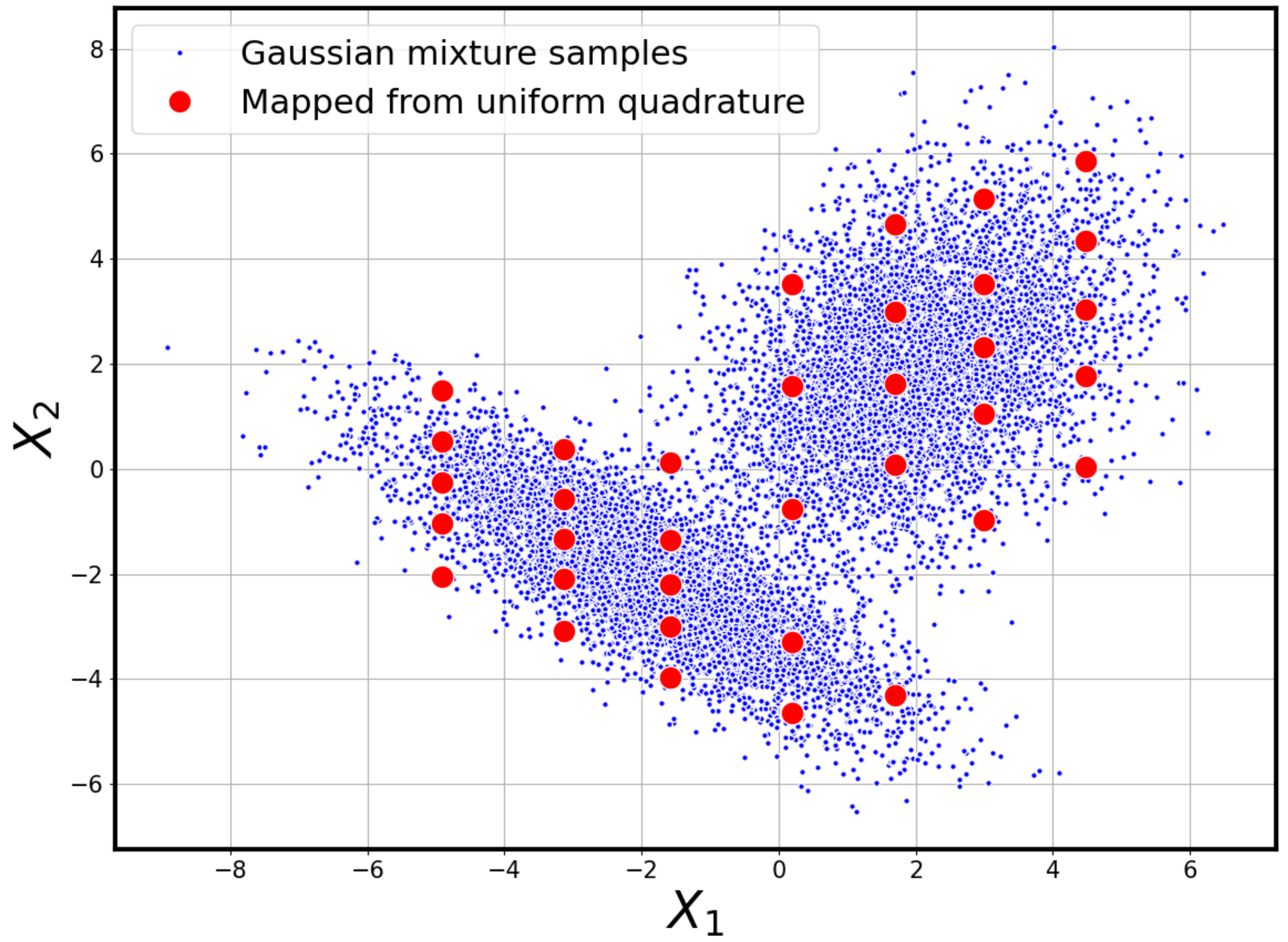
Fwd UQ: Rosenblatt Transformation

- Multivariate generalization of CDF thm [Rosenblatt, 1952].
- KDE-based method, given samples [Sargsyan, 2010].
- Conditional CDFs are hard to evaluate in high-d.

$$\xi_1 = F_1(x_1)$$
$$\xi_2 = F_{2|1}(x_2 \,|\, x_1)$$
$$\vdots$$
$$\xi_n = F_{n|n-1,\dots,1}(x_n \,|\, x_{n-1}, \dots, x_1)$$

$$\frac{\sum_{s=1}^{S} \exp\left(-\frac{(x_1 - x_1^{(s)})^2 + \ldots + (x_{n-1} - x_{n-1}^{(s)})^2}{2h^2}\right) \times \Phi\left(\frac{x_n - x_n^{(s)}}{h}\right)}{\sum_{s=1}^{S} \exp\left(-\frac{(x_1 - x_1^{(s)})^2 + \ldots + (x_{n-1} - x_{n-1}^{(s)})^2}{2h^2}\right)}$$





- As soon as this $X \leftrightarrow \xi$ map is built, PC construction becomes a polynomial fit problem.

Inv UQ: Bayesian likelihood and MCMC

Bayes formula

$$\underbrace{p(\lambda|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|\lambda)}^{\text{Likelihood}} \; \overbrace{p(\lambda)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- Collected data $\{(x_i, y_i)\}_{i=1}^{N}$
- Data model $y_i = f(\lambda; x_i) + \epsilon_i$

- Denominator $p(y)$ is not important
- Likelihood derived from data model assumptions
- For example, gaussian i.i.d. noise ϵ_i leads to

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

- **Markov chain Monte Carlo (MCMC)** samples from posterior by marching in the λ-space.
- **Likelihood** is key:
  - It incorporates statistical assumptions about the discrepancy between model and data.
  - It requires model evaluation at a proposed parameter value λ.

… but it is often infeasible to use model online in an MCMC loop, hence we pre-construct a model surrogate.

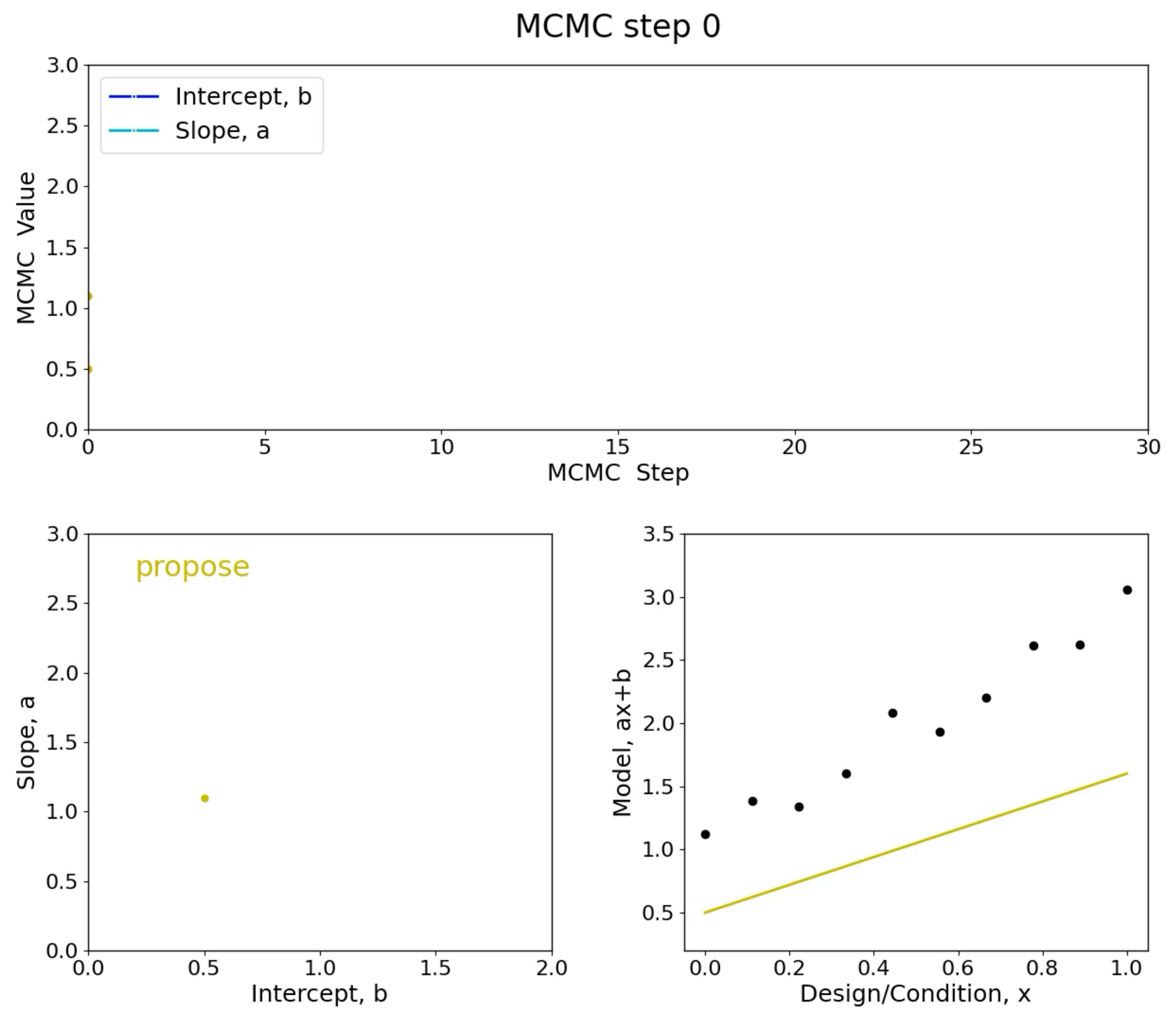
K. Sargsyan (ksargsy@sandia.gov)

Inv UQ: MCMC Example: Line Fit

Linear Model: $f(\overbrace{a,b}^{\lambda};x) = ax + b$

- Metropolis-Hastings algorithm
- Accept/reject mechanism in the parameter space

- Generate a random candidate at step t,
  $\lambda' \sim \pi(\lambda' | \lambda_t)$
- Calculate the acceptance probability
  $$\alpha = \min\left(1, \frac{p(\lambda'|y)}{p(\lambda_t|y)}\frac{\pi(\lambda_t|\lambda')}{\pi(\lambda_t|\lambda')}\right)$$
- Accept with probability α and move to step $t+1$.



*Note: only posterior ratio matters

K. Sargsyan (ksargsy@sandia.gov)

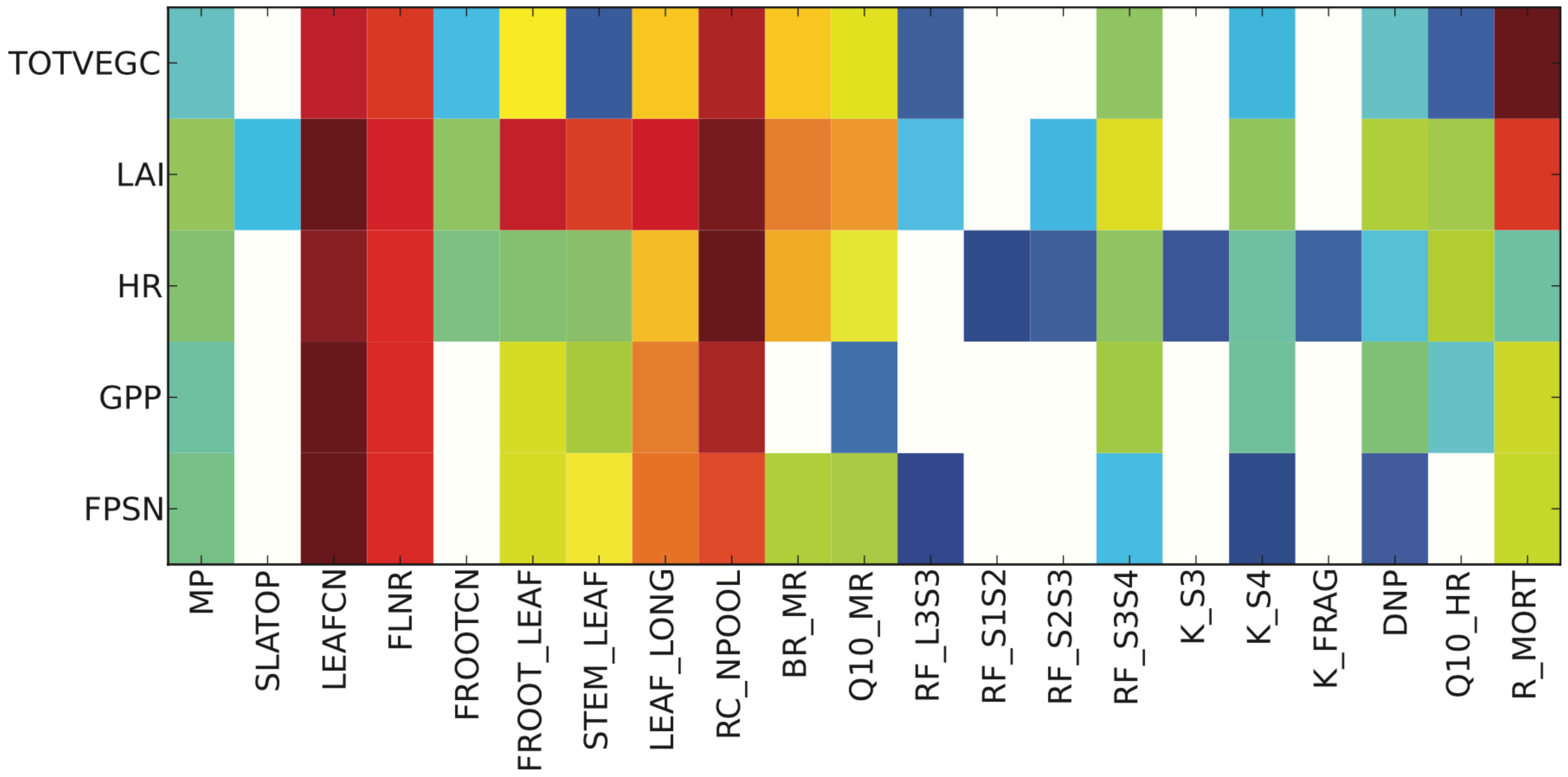
E3SM Land Model (ELM)

- US Dept of Energy (DOE) sponsored Earth system model
- Land, atmosphere, ocean, ice, human system components
- High-resolution, employ DOE leadership-class computing facilities

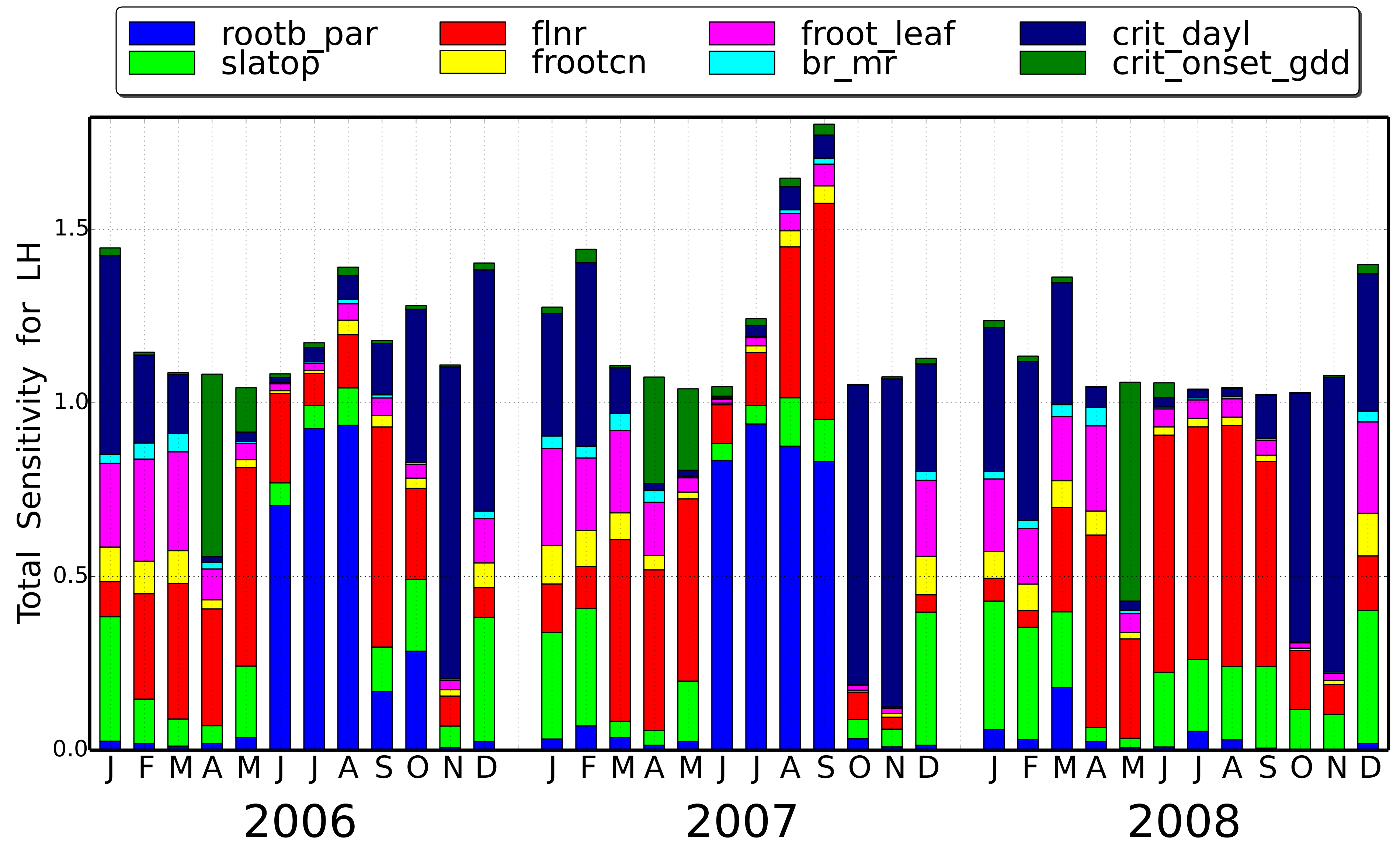
E3SM
Energy Exascale
Earth System Model

e3sm.org

~50 inputs; 5 averaged outputs

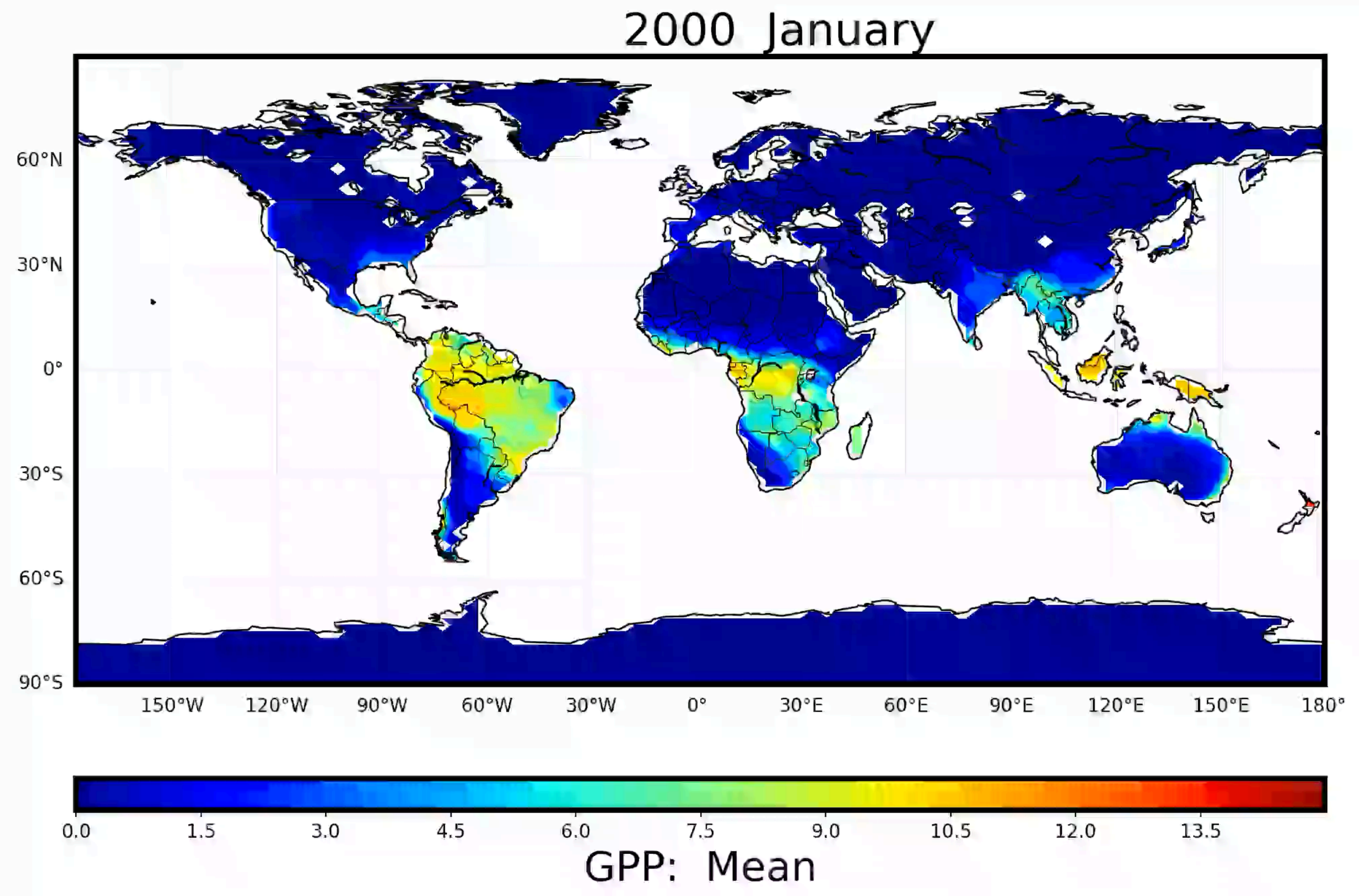


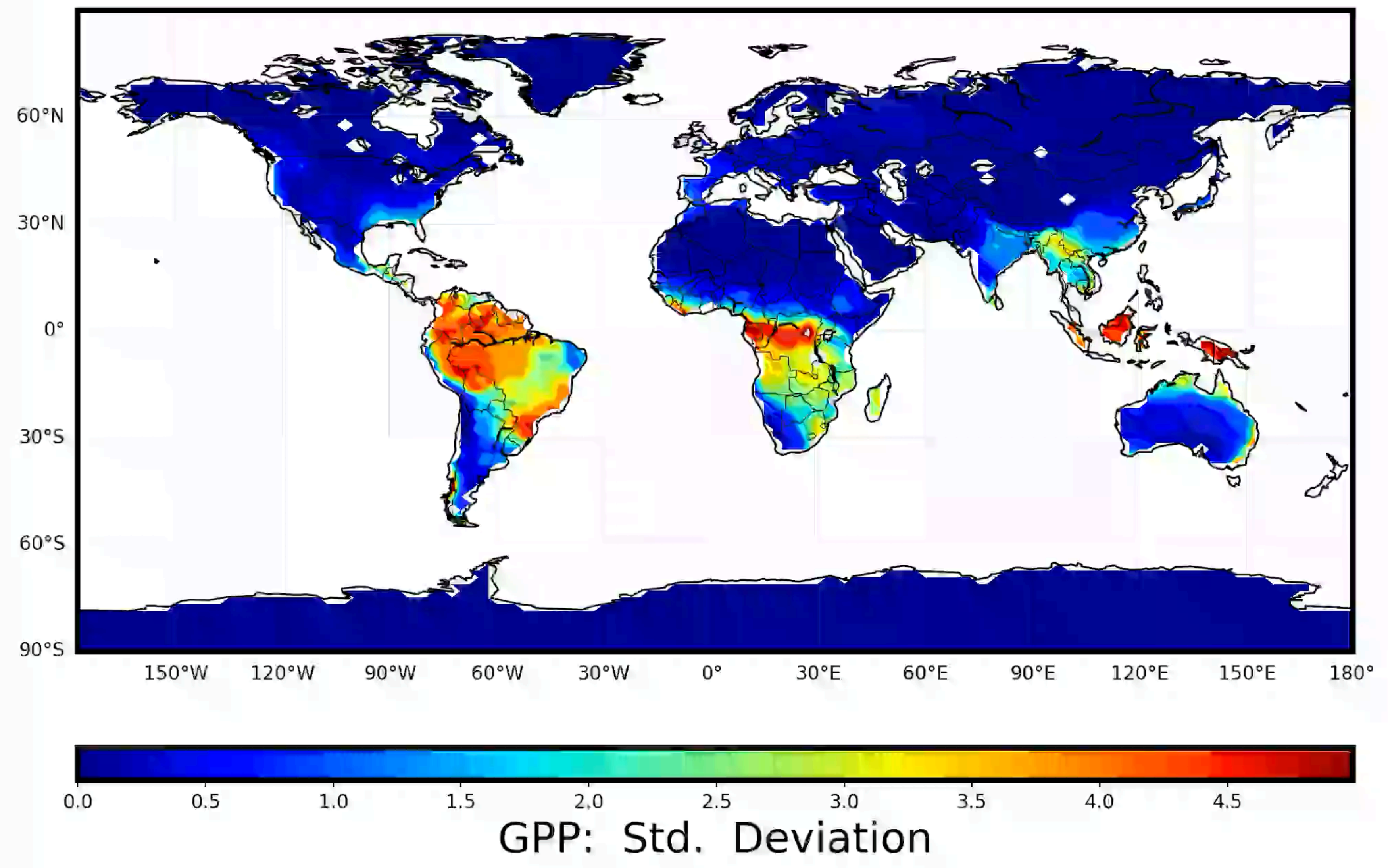
~15 inputs; 60 temporal outputs



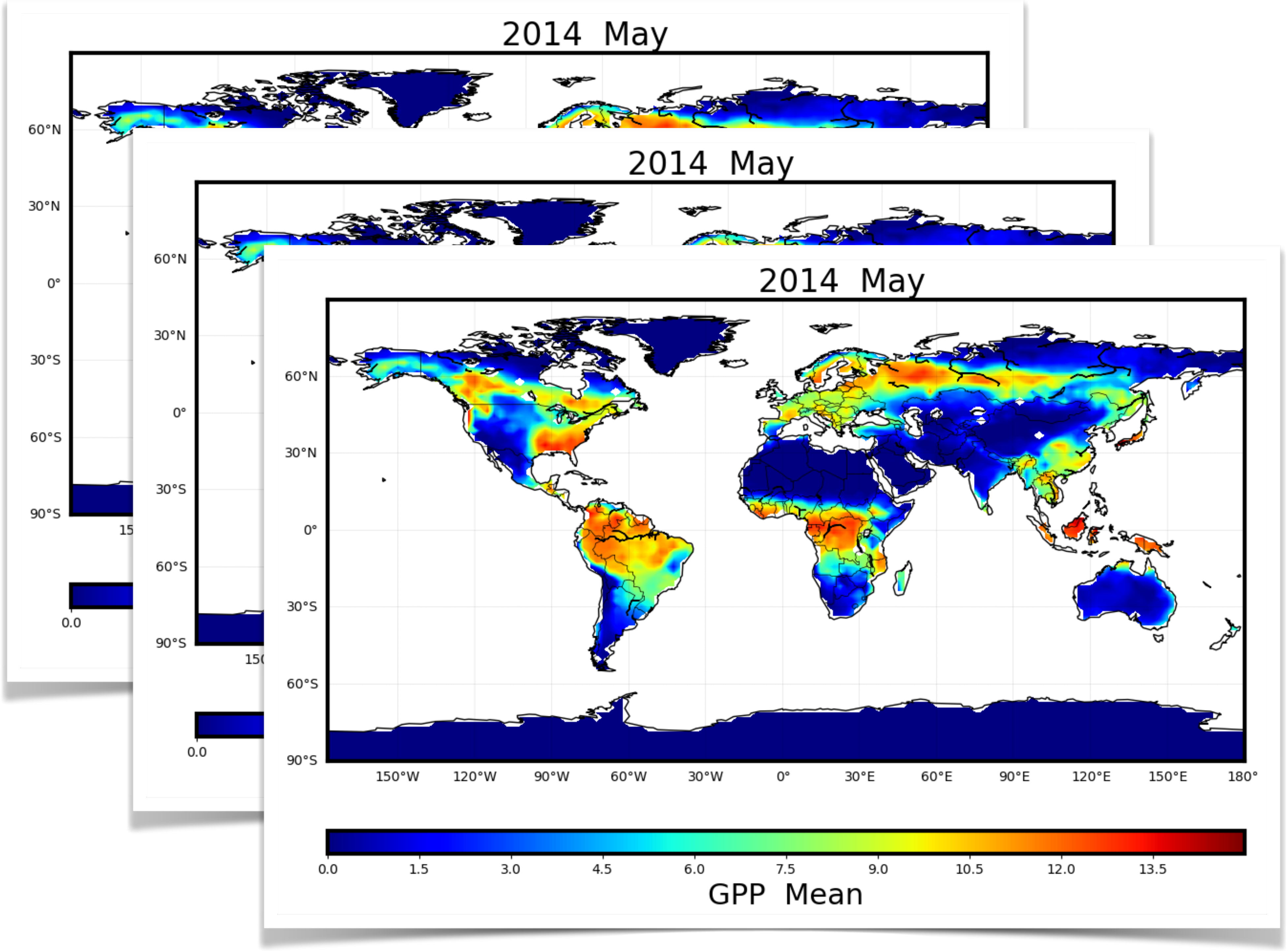
K. Sargsyan (ksargsy@sandia.gov)

E3SM Land Model: Gross Primary Productivity





10 inputs; spatio-temporal output 4000 cells x 180 months

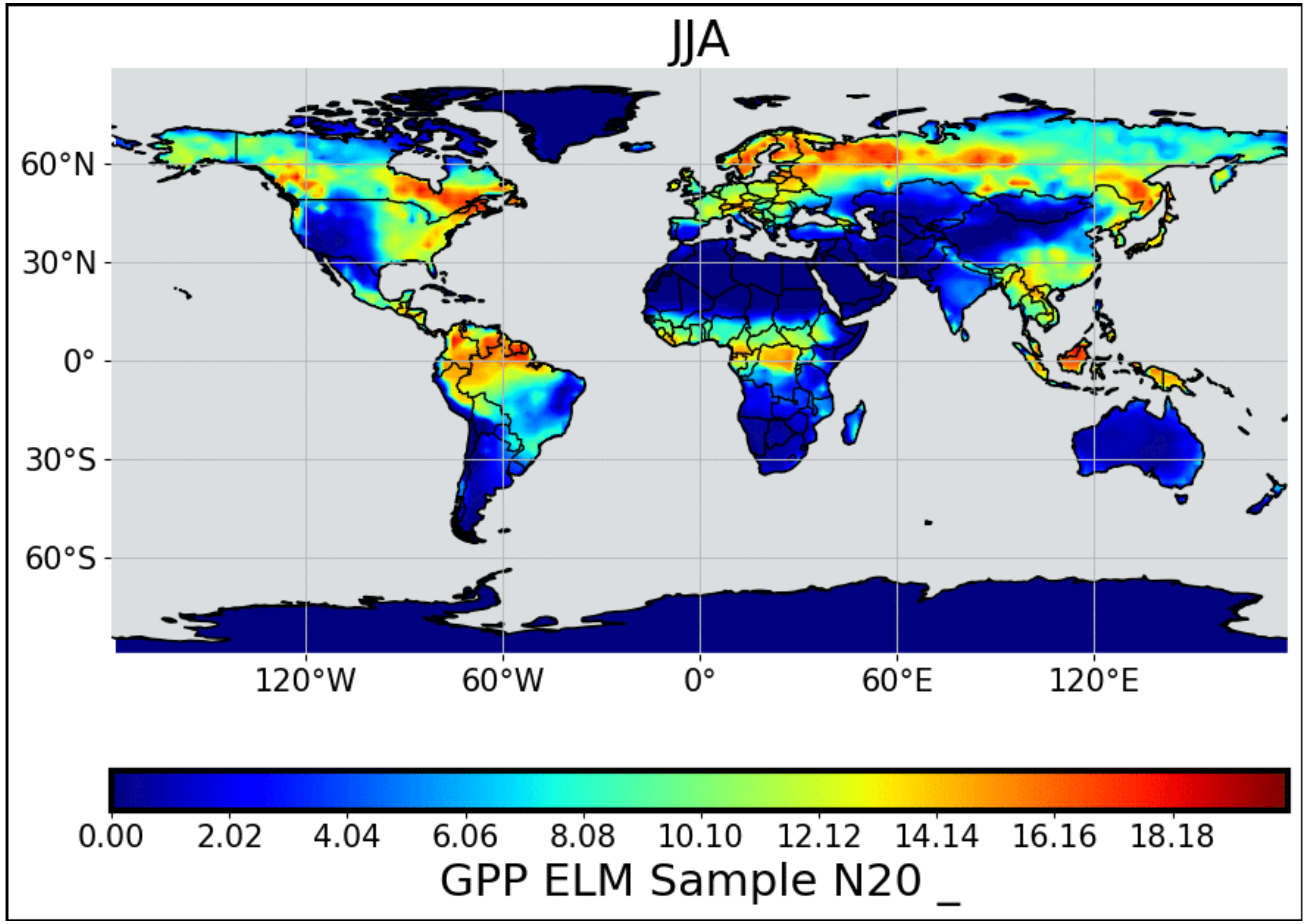


Model Ensemble (275 samples)

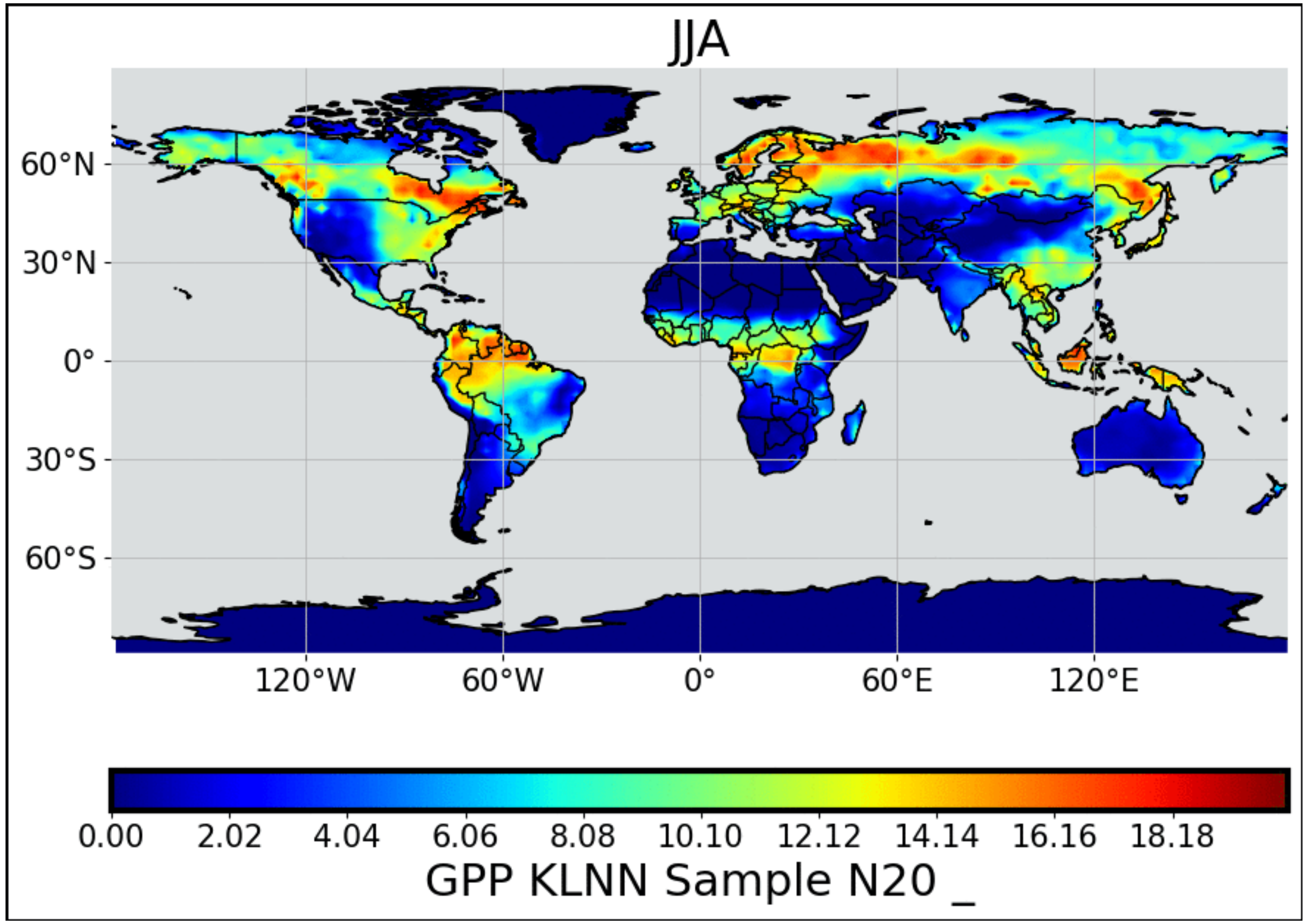
K. Sargsyan (ksargsy@sandia.gov)

E3SM Land Model: Gross Primary Productivity

ELM: Single simulation several hours

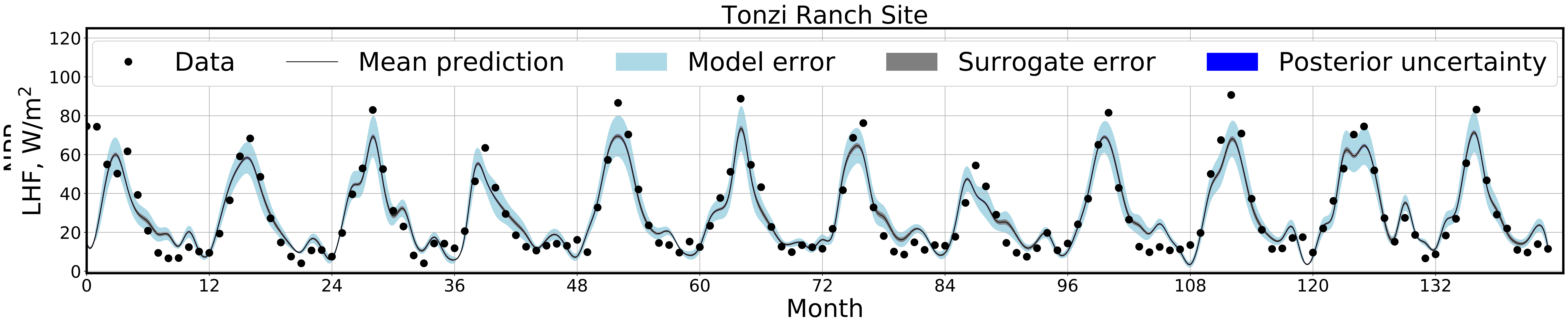


KL+NN Surrogate: Single simulation ~1 sec



K. Sargsyan (ksargsy@sandia.gov)

E3SM Land Model

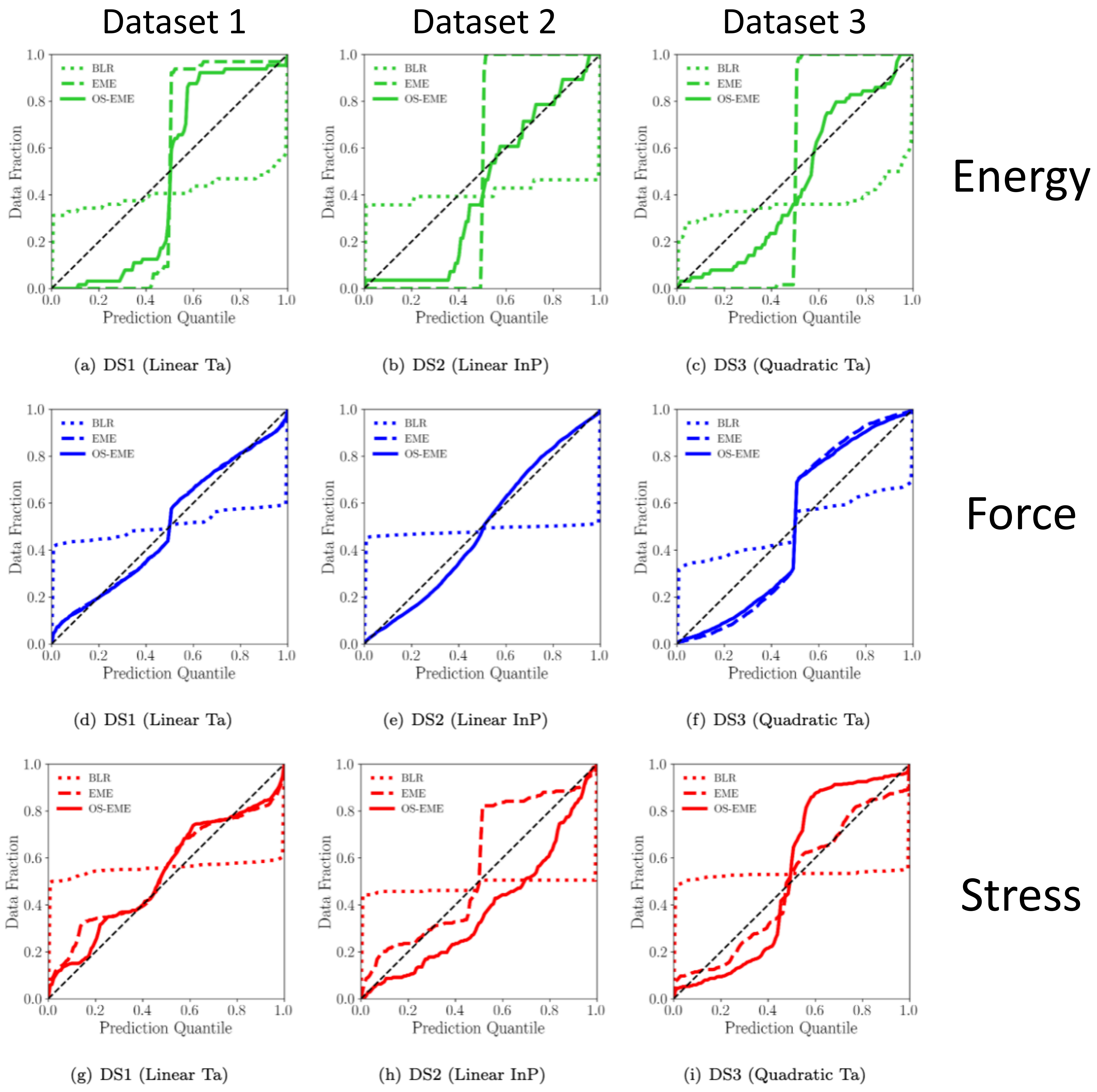


- LHF = Latent Heat Flux
- NPP = Net Primary Productivity

MLIAP Uncertainty estimation with embedded model error

FES Partnership ThermChem:
Integrated Thermomechanical Model of First-wall Components Under Evolving Chemistry and Microstructure During Fusion Reactor Operation

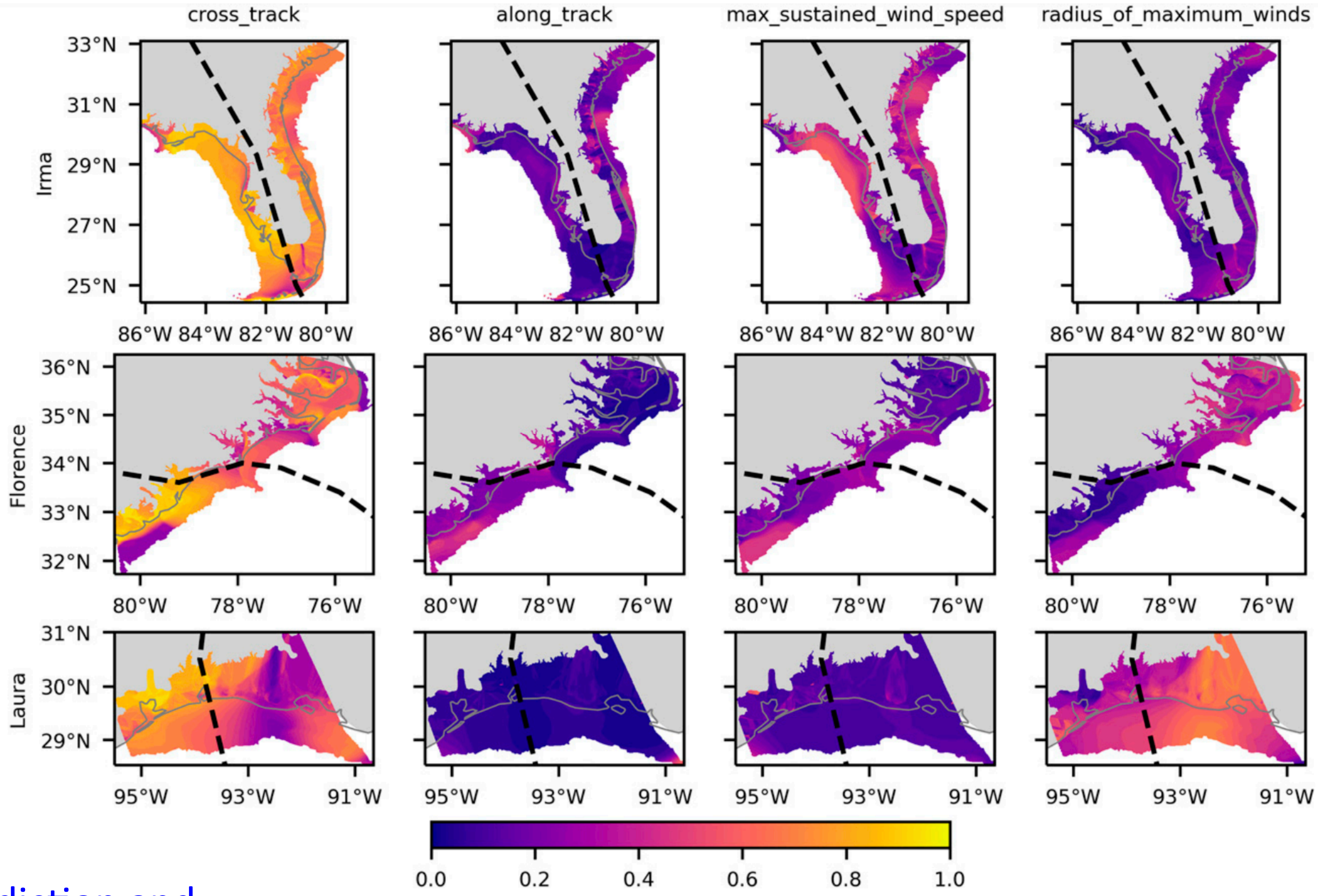
- Develop ML Interatomic Potentials (MLIAPs) with embedded model error uncertainties.
- Extend EME with output-type specific corrections.
- Demo improved (better calibrated) uncertainties, capturing discrepancy between DFT data and MLIAP.



Figs courtesy of Cristian Lacey

Hurricane Modeling

Sensitivity indices of maximum water surface elevation to 4 parameters for 3 hurricane forecasts.



[Pringle et al., "Efficient Probabilistic Prediction and Uncertainty Quantification of Tropical Cyclone–Driven Storm Tides and Inundation", *AI4ES*, 2023]

K. Sargsyan (ksargsy@sandia.gov)

PyTUQ's current usage

- BER: E3SM Land Model UQ

  Reduced-dimensional surrogate, calibration, embedded model error, GSA

- BER Partnership: Quasi-biennal Oscillation

- BES: Exascale Catalytic Chemistry

  Stochastic surrogate, GSA, covariance estimation

- FES Partnership: ThermChem-FW

  Linear regression, Bayesian compressive sensing, Uncertainty propagation, GSA

- Interlab LDRD: Radiography UQ

  Optimization, Inference/MCMC, Variational inference

- ....

K. Sargsyan (ksargsy@sandia.gov)

Summary and Future

What is in UQTk but not in PyTUQ:

Low-rank tensor (canonical) decomposition

Sparse quadrature

Weighted BCS

PC germ maps

Non-standard, custom PC

Transitional MCMC, Single-site MCMC

Some evidence computation scripts

Data-free inference

Intrusive arithmetics

Coming up:

Math documentation/manual

Usage tutorials

K. Sargsyan (ksargsy@sandia.gov)