# Uncertainty Quantification in Computational Science:

## from Physical Models to Neural Networks

*Khachik Sargsyan*

Feb 17, 2025

Banff, Canada

Sandia National Laboratories

BIRS

BIRS Workshop

“Uncertainty Quantification in Neural Network Models”

# Acknowledgements


**Sandia**

Habib Najm
Bert Debusschere
Cosmin Safta
Tiernan Casey
Pieterjan Robbe
Oscar Diaz-Ibarra
Michael Eldred
John Jakeman
Cristian Lacey
Joy Mueller
Luis Damiano

**fmr. Sandia**

Marta D'Elia
Joshua Hudson
James Oreluk
Prashant Rai
Jason Bender

Youssef Marzouk (MIT)
Roger Ghanem (USC)
Xun Huan (UMich)
Daniel Ricciuto (ORNL)

DOE BES, BER, FES
DOE ASCR, SciDAC

U.S. Department of ENERGY

DOD DARPA

SNL LDRD

Sandia National Laboratories

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Uncertainty Quantification in Computational Models

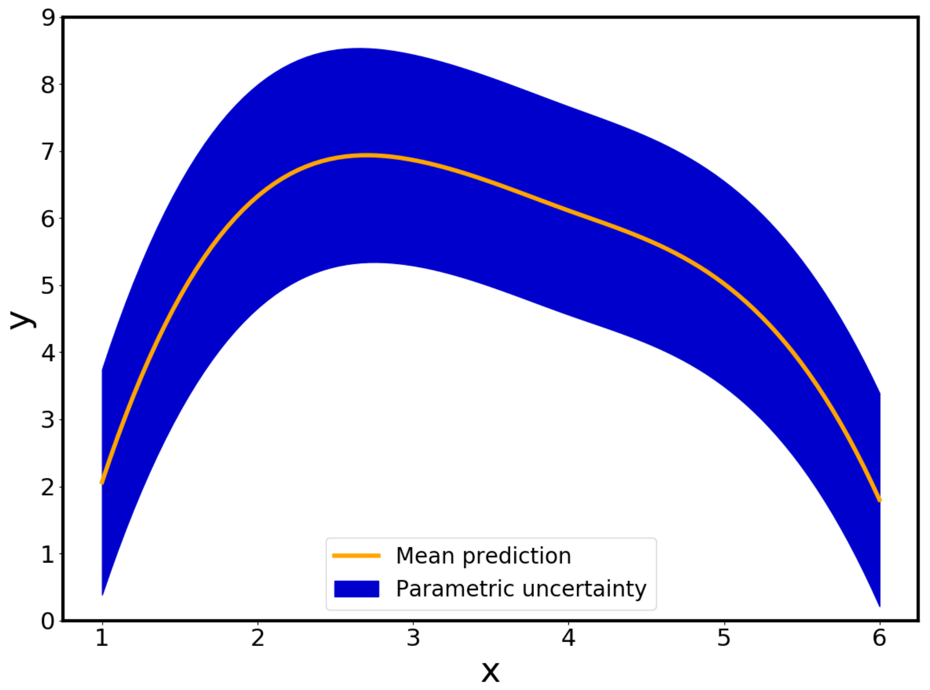
$\lambda$

Physical input parameters

Comp. model

$f(x;\lambda)$

Output prediction




**Forward UQ**

**(Uncertainty propagation, Global sensitivity analysis)**

**Prediction Variance** = **Parametric Uncertainty**



K. Sargsyan (ksargsy@sandia.gov)

# Uncertainty Quantification in Computational Models

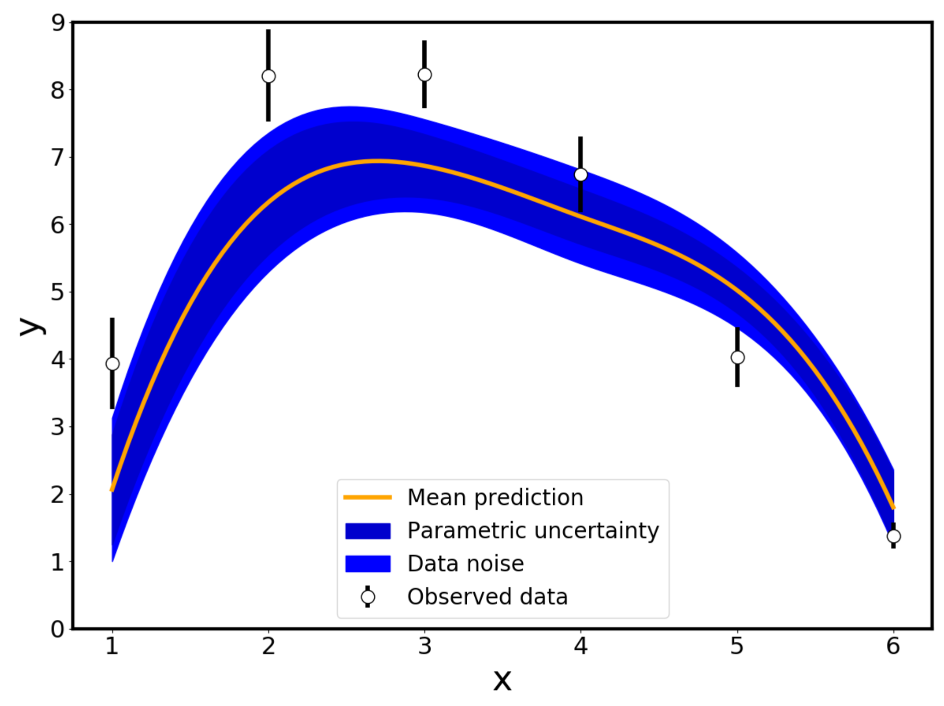
$\lambda$

Physical input parameters

Comp. model

$f(x;\lambda)$

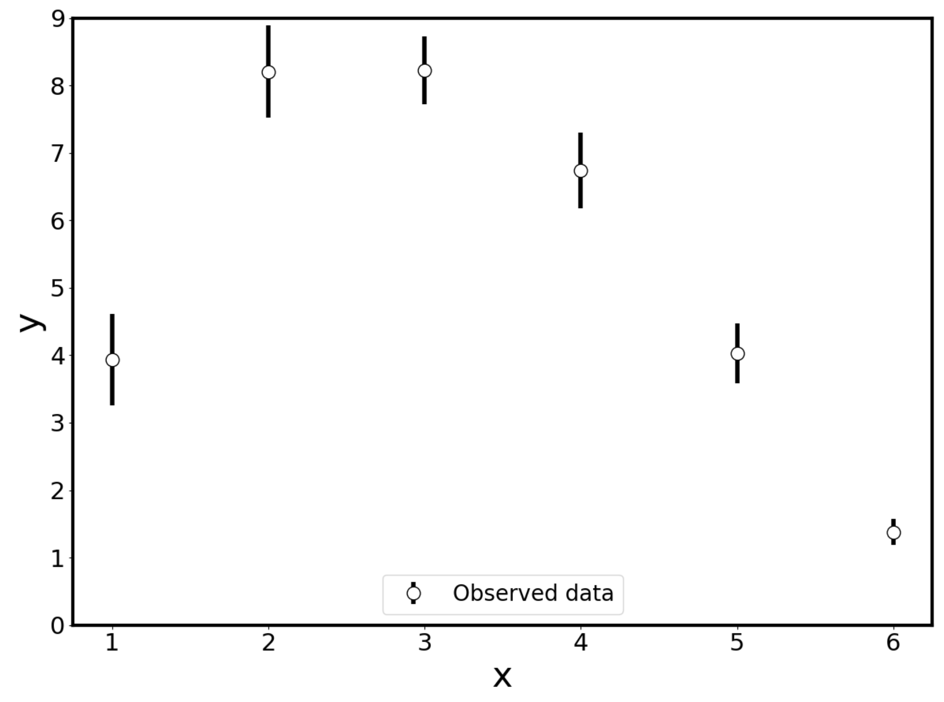
Output prediction




Measurement model

$y(x)$

Observed data




## Forward UQ

**(Uncertainty propagation, Global sensitivity analysis)**

## Inverse UQ

**(Parameter estimation, Model calibration, validation, Model selection)**

**Prediction Variance** = **Parametric Uncertainty** + **Data Noise**



K. Sargsyan (ksargsy@sandia.gov)

# Uncertainty Sources

- Model parameters
- Initial/boundary conditions
- Model geometry
- Lack of knowledge
- Unresolved physics
- Data noise
- Intrinsic stochasticity
- Numerical errors

# UQ Use Cases

- Model validation/prediction
- Model comparison/selection
- Confidence assessment
- Reliability analysis
- Dimensionality reduction
- Optimal design
- Decision support
- Data assimilation


K. Sargsyan (ksargsy@sandia.gov)

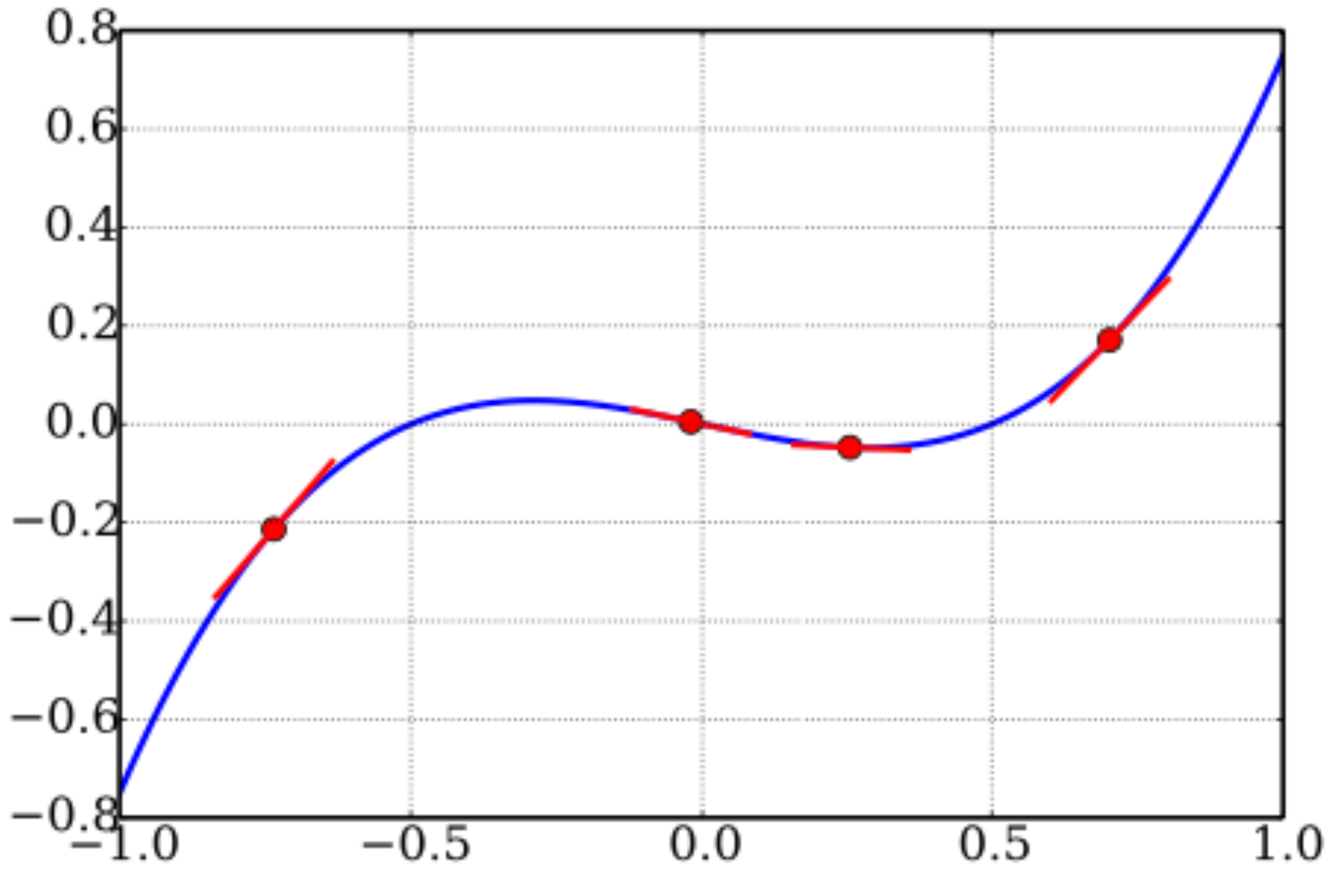
# Forward UQ


K. Sargsyan (ksargsy@sandia.gov)

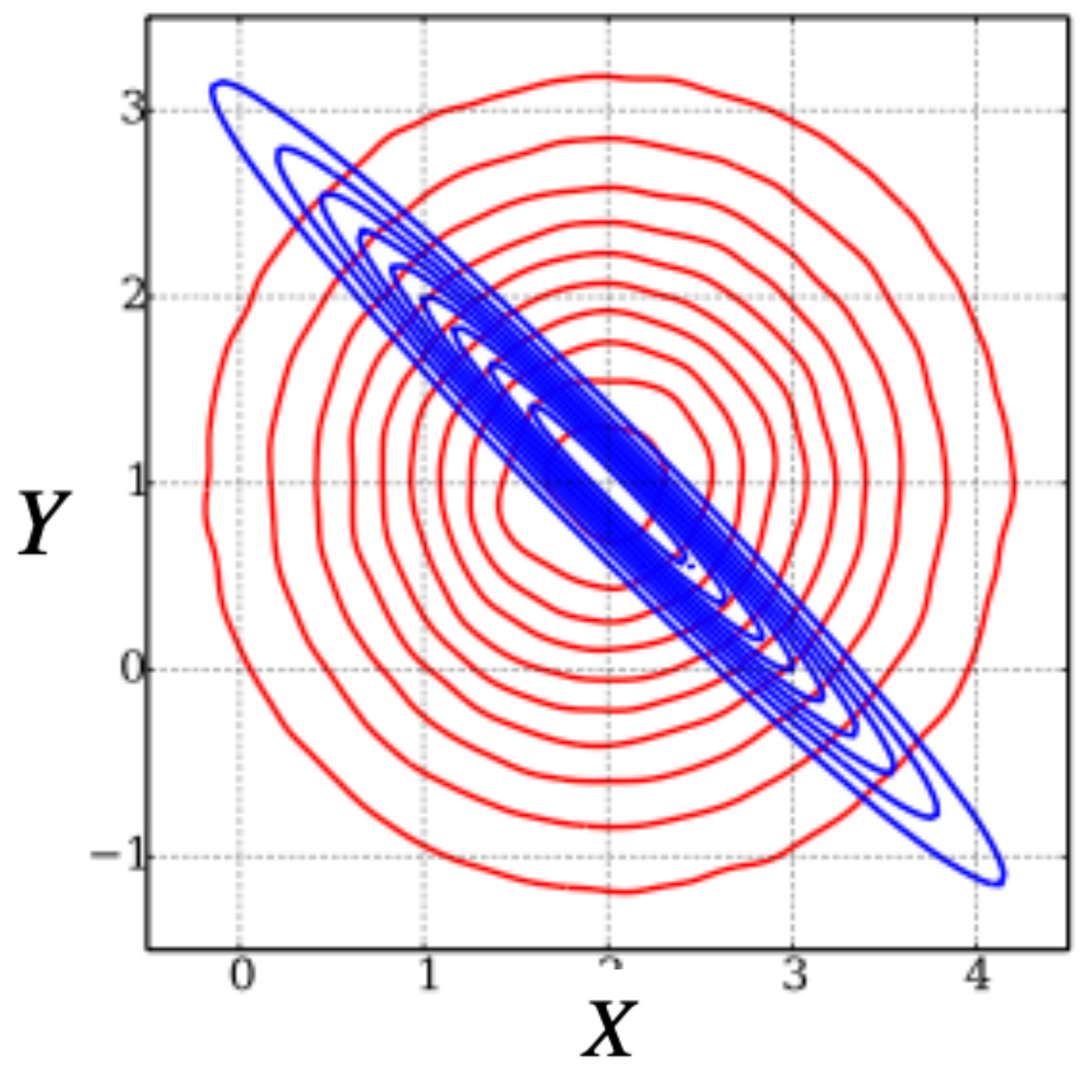
# Forward UQ

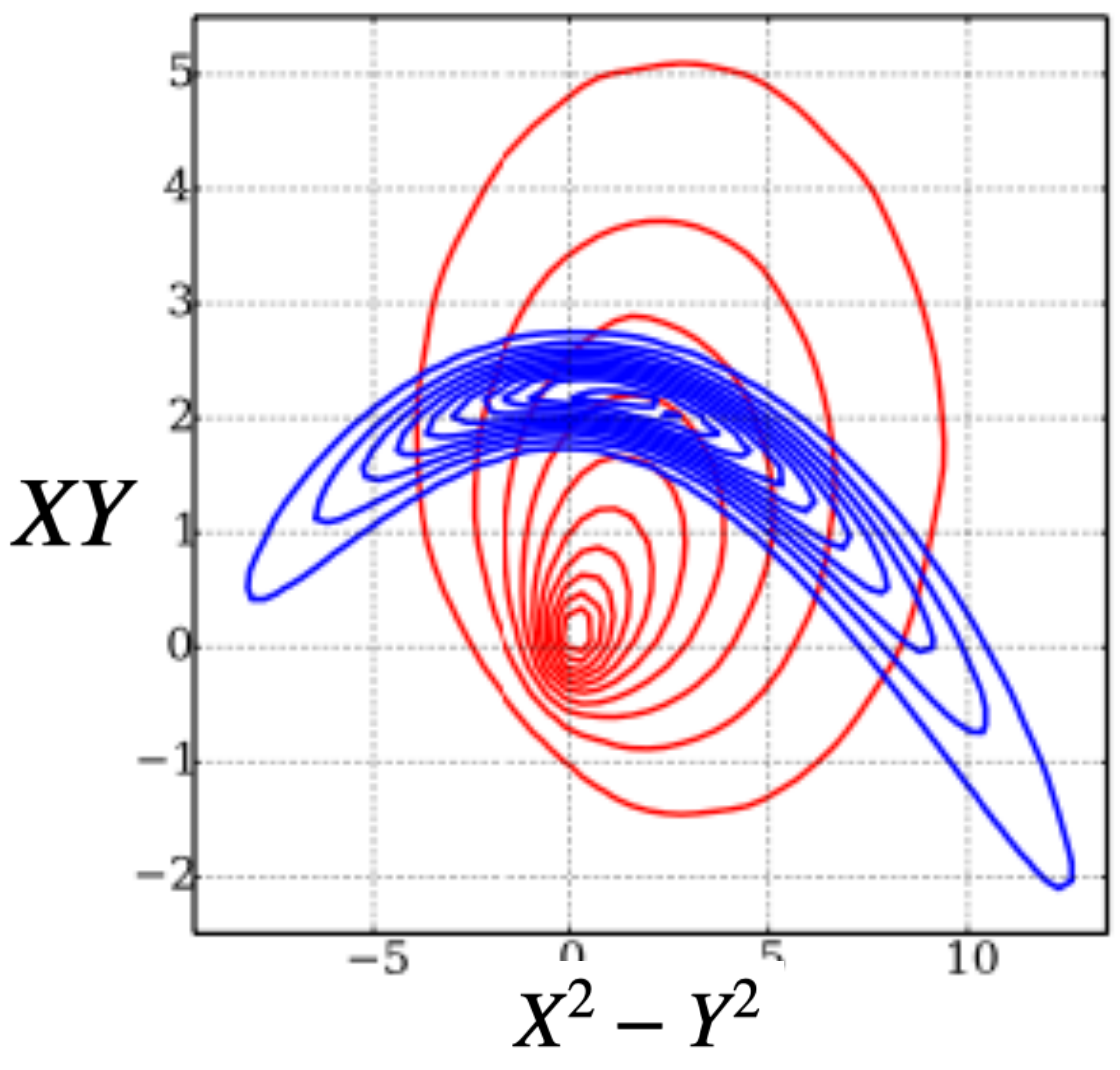
- Local methods:
  - Derivative-based sensitivity
  - Error propagation
- ... miss global nonlinear behavior

$$\Delta y = \frac{df}{dx}\Delta x$$




- Non-probabilistic methods:
  - Evidence theory
  - Fuzzy logic
  - Interval math
- ... miss correlations, tails, ....



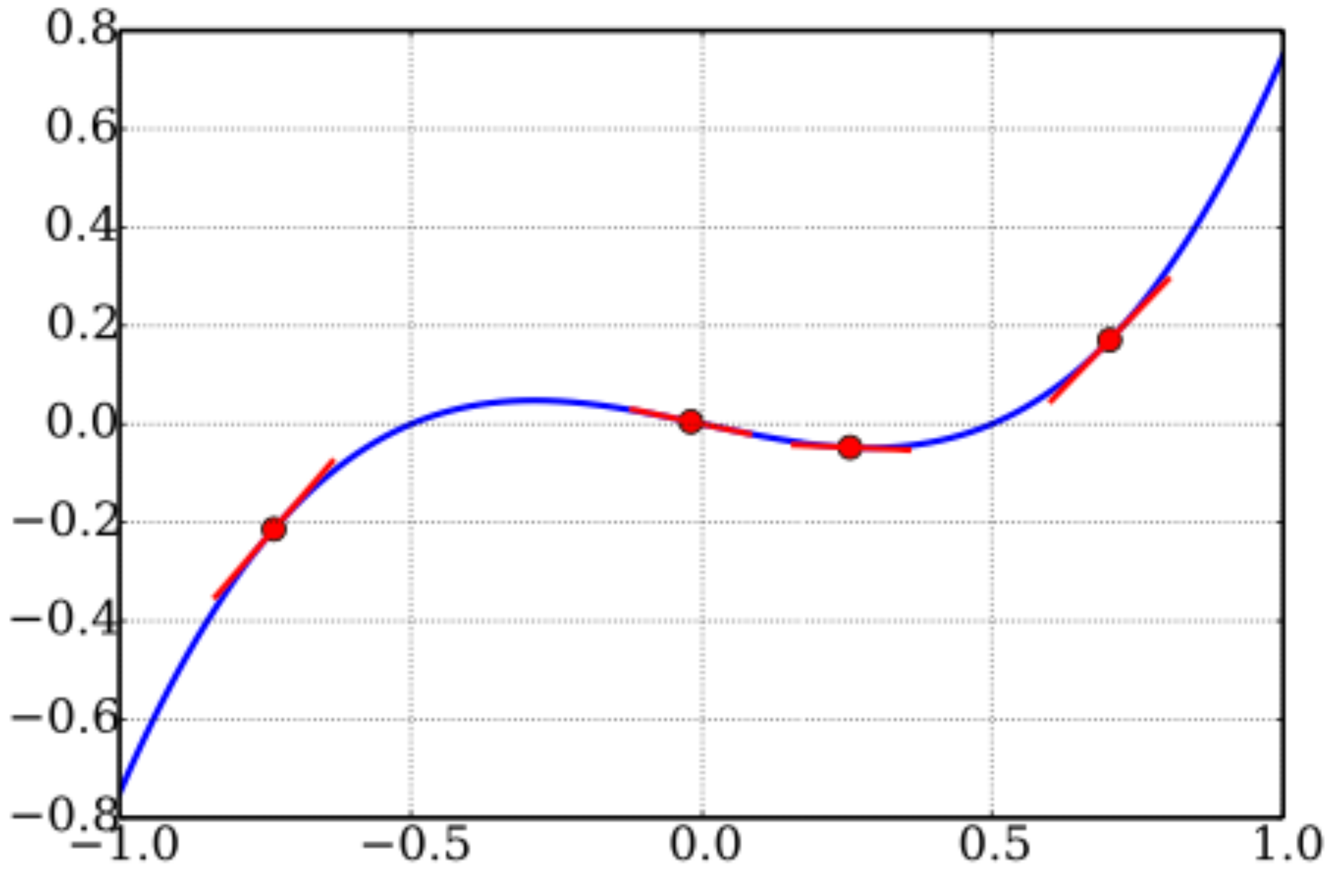



K. Sargsyan (ksargsy@sandia.gov)

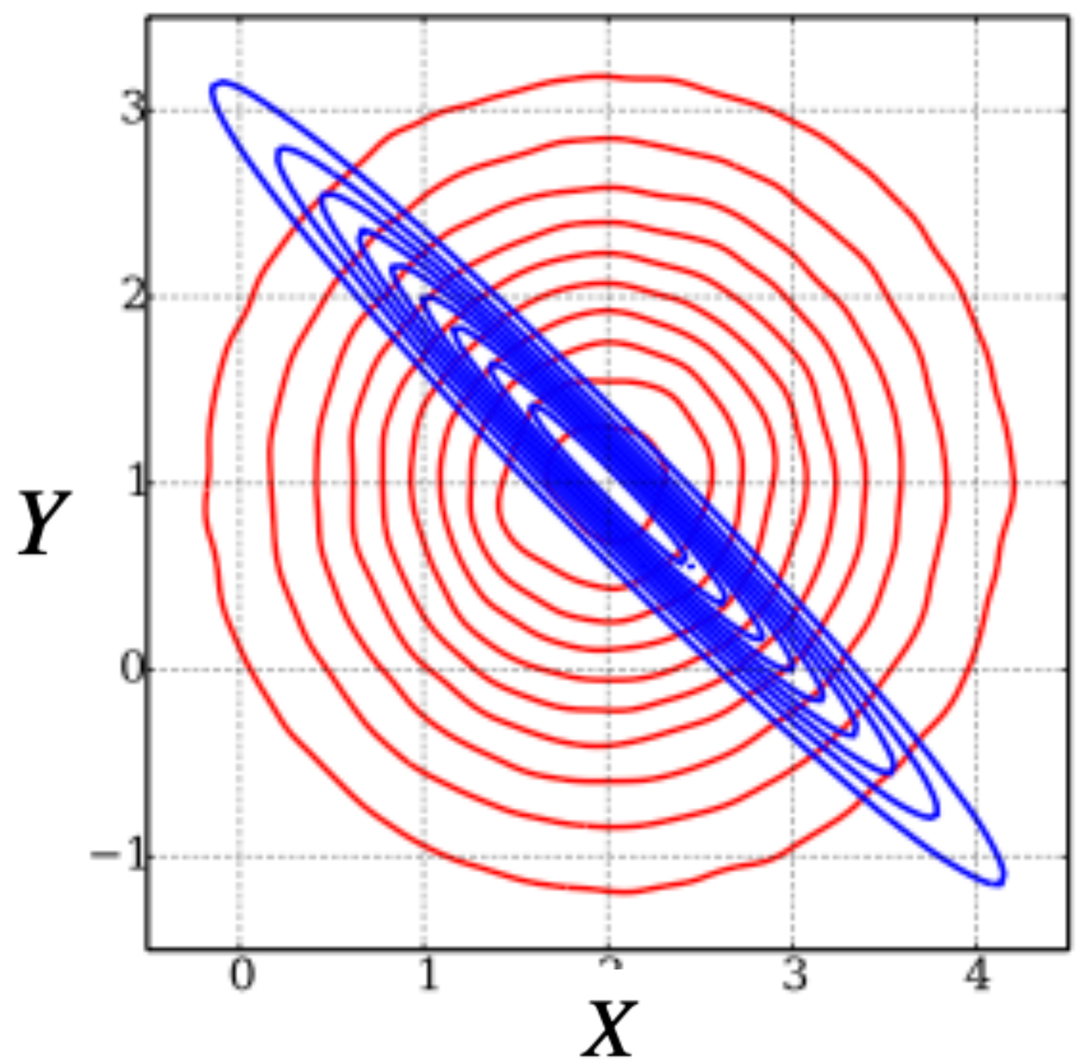
# Forward UQ

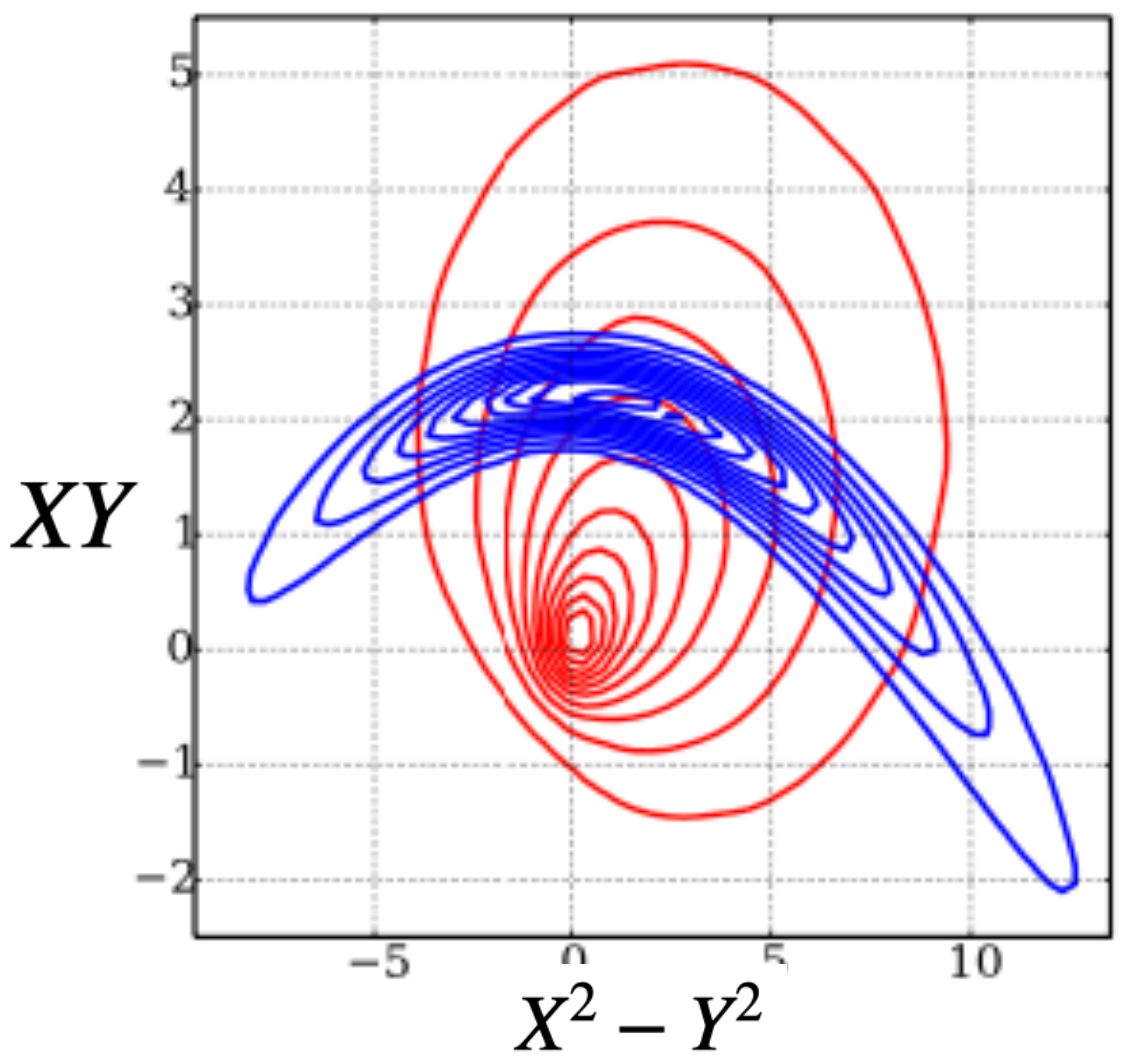
- Local methods:
  - Derivative-based sensitivity
  - Error propagation
- … miss global nonlinear behavior

$$\Delta y = \frac{df}{dx}\Delta x$$



- Non-probabilistic methods:
  - Evidence theory
  - Fuzzy logic
  - Interval math
- … miss correlations, tails, ….




- **Probabilistic methods**: cast all inputs and outputs as random variables



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Main tool — Polynomial Chaos

- Represent a random variable $X$ as a polynomial expansion with respect to standard random variables $\xi$:

Convergent for finite-variance r.v. $X$.

$$\boxed{X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)}$$

- Describes the *random variable* $X$ with a vector of *deterministic* coefficients, *PC modes* $(x_0, x_1, \ldots, x_p)$.
- Theory is solid: in the limit of infinite order and dimensions; but in practice these are modeling choices.
- Enables functional analysis methods for forward UQ.
- PC first introduced by [Wiener, 1938]; revitalized in [Ghanem&Spanos, 1991].

# Fwd UQ: Main tool — Polynomial Chaos

$$X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

Polynomials $\psi_k(\,\cdot\,)$ are orthogonal with respect to the probability measure of $\xi$

$$\int \psi_i(\xi)\psi_j(\xi)\pi_\xi(\xi)d\xi = ||\psi_i||^2 \delta_{ij}$$

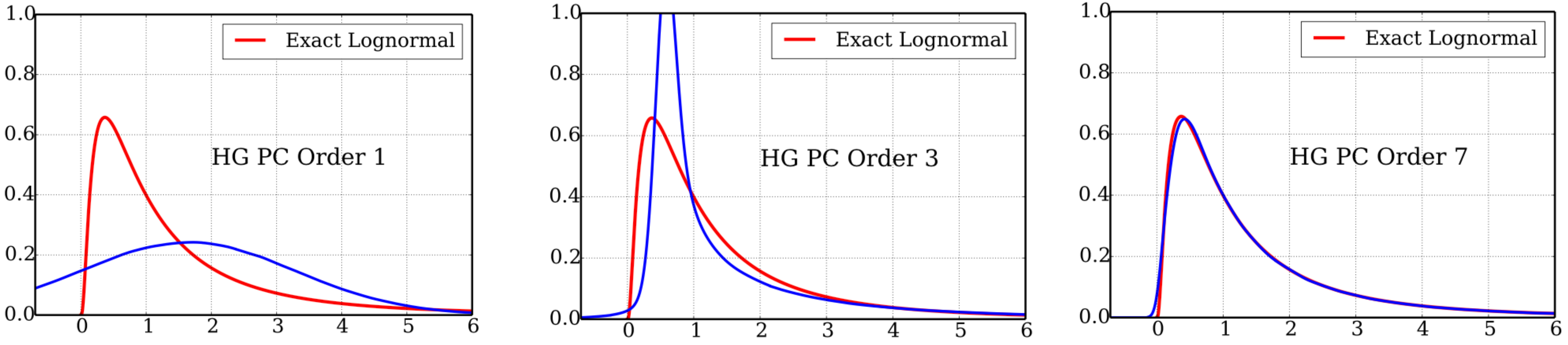
| PC Type | Domain | Density $\pi_\xi(\xi)$ | Polynomial | Free parameters |
|---|---|---|---|---|
| Gauss-Hermite | $(-\infty, +\infty)$ | $\frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}}$ | Hermite | none |
| Legendre-Uniform | $[-1, 1]$ | $\frac{1}{2}$ | Legendre | none |
| Gamma-Laguerre | $[0, +\infty)$ | $\frac{\xi^\alpha e^{-\xi}}{\Gamma(\alpha+1)}$ | Laguerre | $\alpha > -1$ |
| Beta-Jacobi | $[-1, 1]$ | $\frac{(1+\xi)^\alpha (1-\xi)^\beta}{2^{\alpha+\beta+1} B(\alpha+1,\beta+1)}$ | Jacobi | $\alpha > -1, \beta > -1$ |

Most common random-variable/polynomial pairs:
Askey scheme [Xiu&Karniadakis, 2002]; [Knio&LeMaître, 2010].

# Fwd UQ: Toy example — lognormal r.v.

$$X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

$$X = e^{\xi} = 1 + \xi + \frac{1}{2}\xi^2 + \frac{1}{3!}\xi^3 + \ldots$$




… but we typically do not have the explicit mapping available.



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Multi-dimensional PC

$$\boxed{X \simeq \sum_{k=0}^{p} x_k \, \Psi_k(\xi)}$$

$$\xi = (\xi_1, \ldots, \xi_d)$$

$$\Psi_k(\xi_1, \ldots, \xi_d) = \psi_{k_1}(\xi_1) \times \cdots \times \psi_{k_d}(\xi_d)$$

- For example, $X = x_0 + x_1\xi_1 + x_2\xi_2 + \ldots x_d\xi_d + x_{d+1}\xi_1\xi_2 + \ldots + x_*\xi_{d-1}\xi_d + x_{**}\xi_1^2 + \ldots.$
- Usually, the problem dictates how to choose the underlying <u>stochastic dimensionality</u> $d$.


K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Multi-dimensional PC

$$\boxed{X \simeq \sum_{k=0}^{p} x_k \, \Psi_k(\xi)}$$

$$\xi = (\xi_1, \ldots, \xi_d)$$

$$\Psi_k(\xi_1, \ldots, \xi_d) = \psi_{k_1}(\xi_1) \times \cdots \times \psi_{k_d}(\xi_d)$$

- For example, $X = x_0 + x_1\xi_1 + x_2\xi_2 + \ldots x_d\xi_d + x_{d+1}\xi_1\xi_2 + \ldots + x_*\xi_{d-1}\xi_d + x_{**}\xi_1^2 + \ldots.$

- Usually, the problem dictates how to choose the underlying <u>stochastic dimensionality</u> $d$.

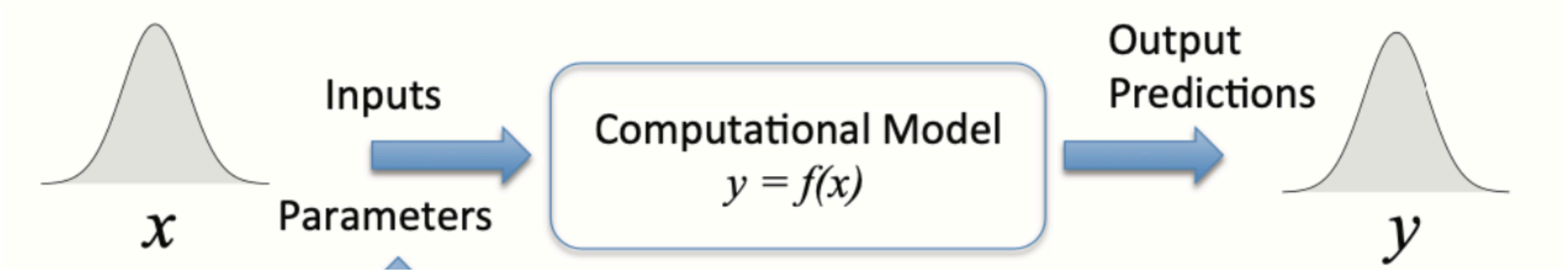
- Fun example: $X \sim Exp(1/2)$ exponential random variable
  - no 1-dimensional finite order expansion
  - But there is exact 2-dimensional PC $X = \xi_1^2 + \xi_2^2$



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: PC Usage in CompSci

$$x = \sum_k x_k \Psi_k(\xi)$$




$$y = \sum_k y_k \Psi_k(\xi)$$

- Strategy:
  - Represent model inputs as PC
  - Sample input PC
  - Evaluate forward model
  - Build PC for model outputs

**This is really regression, or supervised Machine Learning**

- Utility/advantages:
  - Much more efficient than Monte-Carlo propagation
  - Serves as a surrogate model
  - Free extraction of moments $\mathbb{E}[Y] = y_0$ $\mathbb{V}[Y] = \sum_{k\neq 0} y_k^2 ||\Psi_k||^2$
  - Free extraction of sensitivities

# Fwd UQ: Input PC construction is non-trivial

- Given Probability Density Function (PDF)
  - challenging PDF-to-PC map in high-d
- Given samples (see next slides)
- Take from literature
  - potentially lose context
- Elicit from experts,
  - "approx. 2.5"
  - "in a range [4.5-8.8]"
  - "I think 5 ± 0.4"
- Obtain from inverse problem solution
  - Bayesian posterior PDF (see later in this talk)

$$X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

$$\pi_X(x) = ?$$



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: PC Construction

$$X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

- Orthogonal projection (like Fourier):

$$x_k = \frac{1}{||\psi_k||^2} \mathbb{E}[X(\cdot)\psi_k(\xi)]$$

- Need a map $X \leftrightarrow \xi$ to compute the integral $\mathbb{E}[X(\cdot)\psi_k(\xi)] = \int X(\xi)\psi_k(\xi)\pi_\xi(\xi)d\xi$

# Fwd UQ: PC Construction

$$X \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

- Orthogonal projection (like Fourier):

$$x_k = \frac{1}{||\psi_k||^2} \mathbb{E}[X(\,\cdot\,)\psi_k(\xi)]$$

- Need a map $X \leftrightarrow \xi$ to compute the integral $\mathbb{E}[X(\,\cdot\,)\psi_k(\xi)] = \int X(\xi)\psi_k(\xi)\pi_\xi(\xi)d\xi$

- Cumulative distribution function (CDF) transform helps:
  - Legendre-Uniform PC, $\xi$ is uniform: $X = F_X^{-1}\left(\frac{\xi+1}{2}\right)$
  - Gauss-Hermite PC, $\xi$ is normal: $X = F_X^{-1}\left(\Phi(\xi)\right)$

[$F_X(\,\cdot\,)$ is CDF of $X$, and $\Phi(\,\cdot\,)$ is CDF of standard normal]



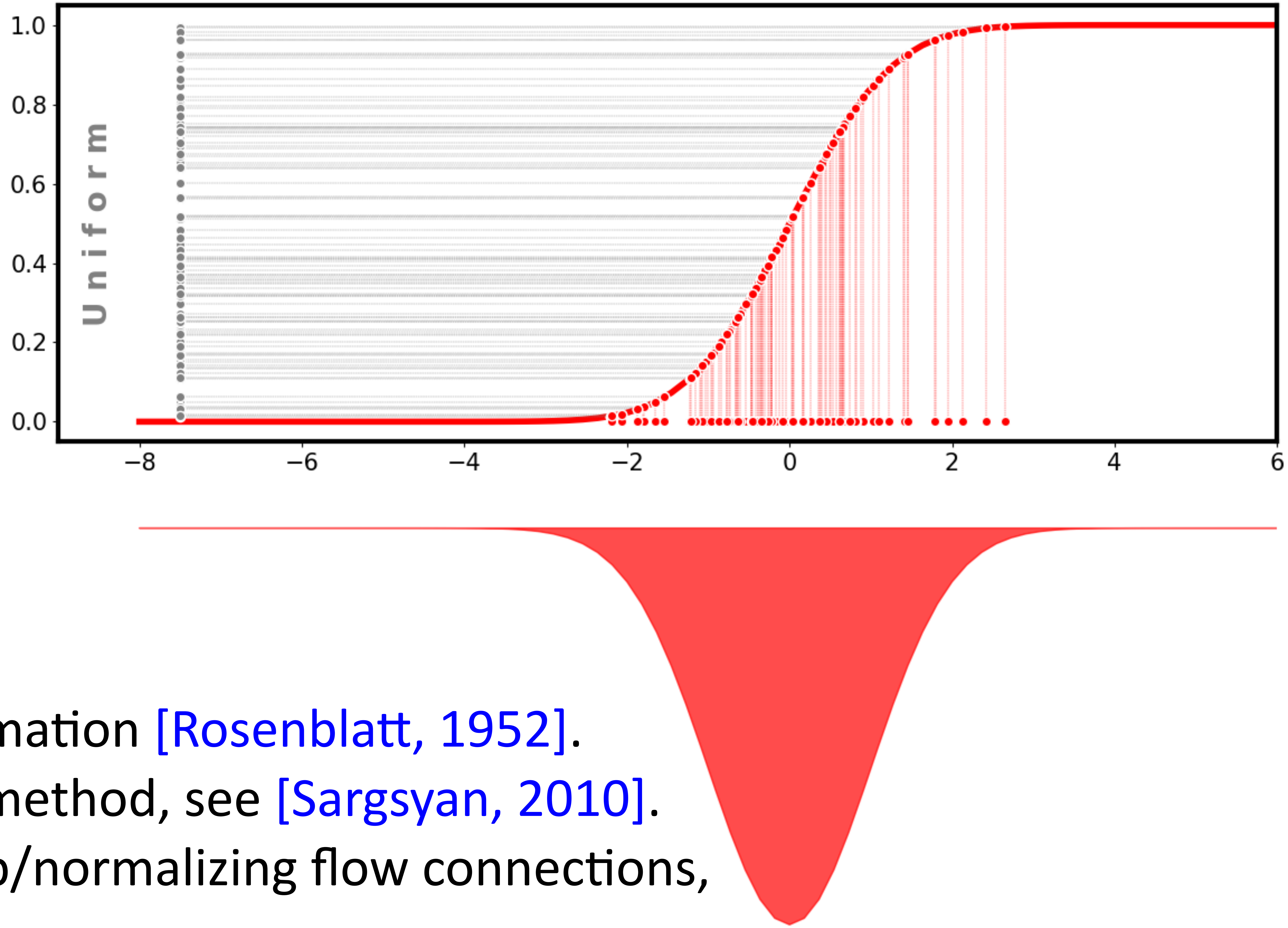
K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: PC Construction aided by CDF

$$X(\xi) \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

General R.V.

Normal




- With the $X \leftrightarrow \xi$ map in place, the PC construction becomes a regression/projection (supervised learning) problem

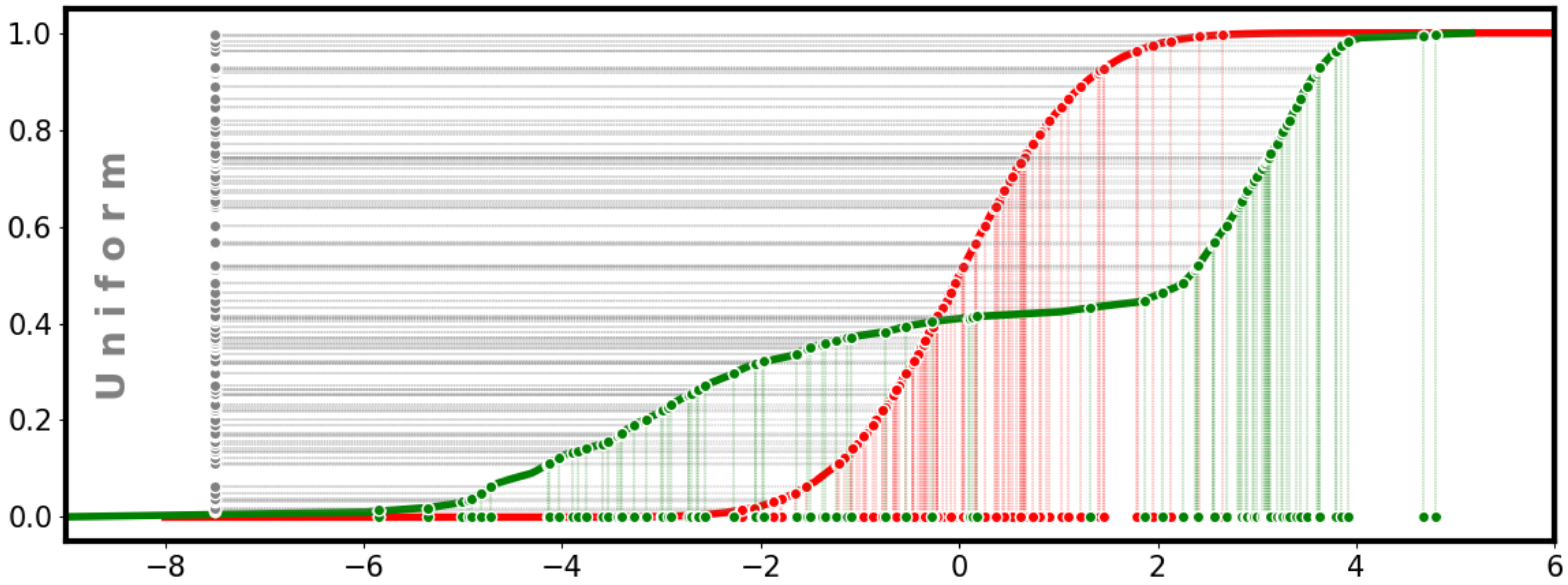
- In >1 dim: Rosenblatt transformation [Rosenblatt, 1952].
- For kernel density estimation method, see [Sargsyan, 2010].
- For more recent transport map/normalizing flow connections, see [Baptista, 2024].


K. Sargsyan (ksargsy@sandia.gov)

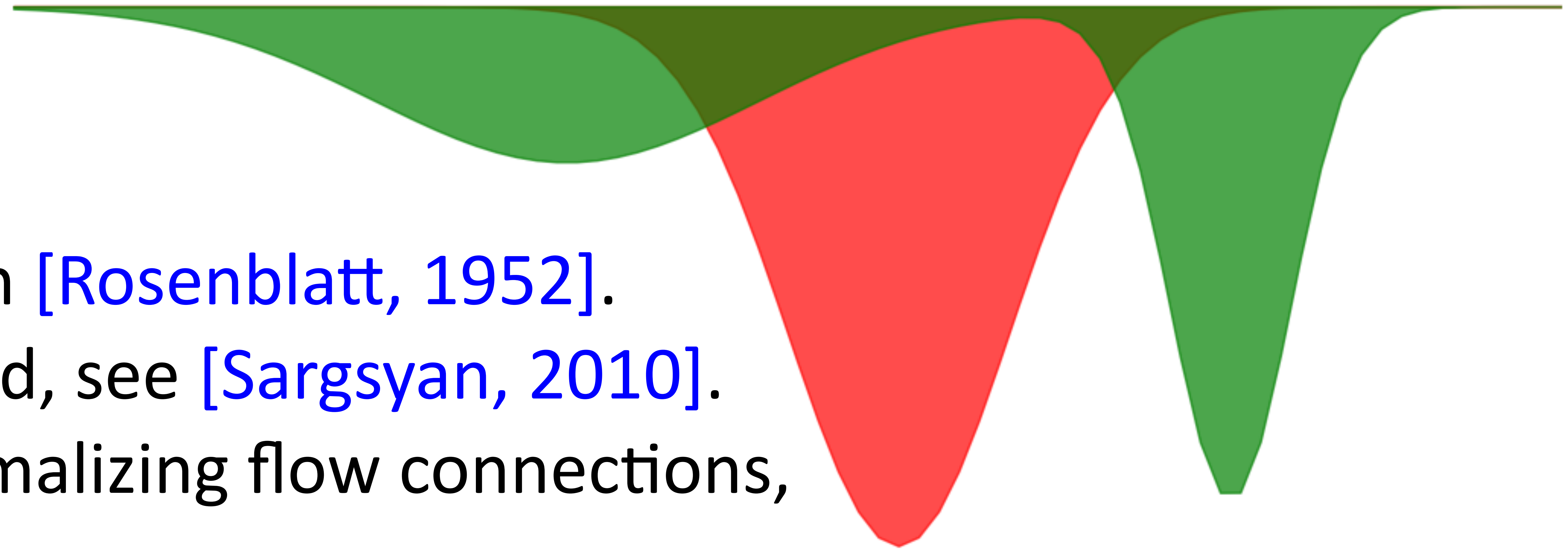
# Fwd UQ: PC Construction aided by CDF

$$X(\xi) \simeq \sum_{k=0}^{p} x_k \, \psi_k(\xi)$$

General R.V. Normal




- With the $X \leftrightarrow \xi$ map in place, the PC construction becomes a regression/projection (supervised learning) problem



- In >1 dim: Rosenblatt transformation [Rosenblatt, 1952].
- For kernel density estimation method, see [Sargsyan, 2010].
- For more recent transport map/normalizing flow connections, see [Baptista, 2024].

# Fwd UQ: Uncertainty Propagation

Input $X = \sum_k x_k \Psi_k(\xi)$

Output $Y = f(X) \approx \sum_k c_k \Psi_k(\xi)$

- Basic task: given PC for the inputs $X$, construct PC for the outputs $Y$.
- Input-output map defined explicitly $Y = f(X)$, or implicitly, e.g. via governing eqn $g(Y, X) = 0$.

### Intrusive methods

- Project governing equations
- Intrusive arithmetics
- Results in set of equations for the PC modes
- Elegant, one solution captures all dynamics
- Requires redesign of computer code PCEs
- Aliasing, long-time horizon

[Debusschere, 2004]

### Non-intrusive methods

- Project outputs of interest
- Sampling to evaluate projection operator
- Can use existing code as black box
- Embarrassingly parallel
- Only computes PCEs for quantities of interest
- Suffers from curse of dimensionality


K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Non-intrusive: Projection vs Regression

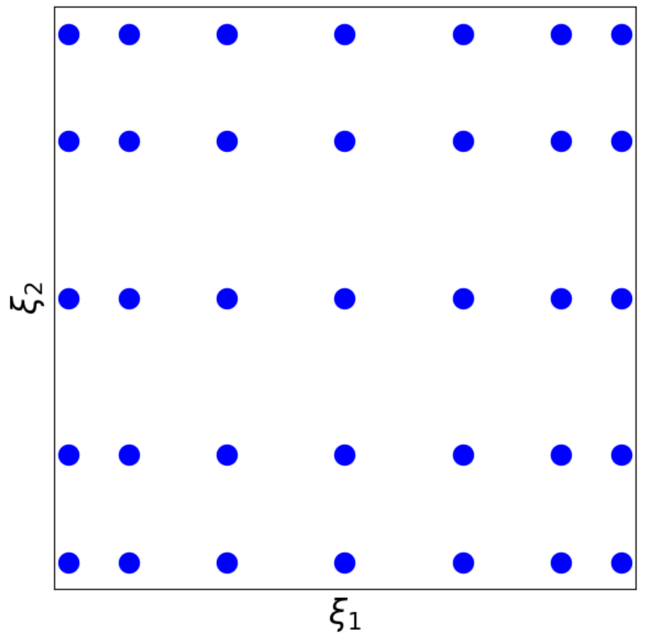
$$f(X(\xi)) \approx \sum_k c_k \Psi_k(\xi)$$

## Projection

$$argmin_c || f(\xi) - \sum_k c_k \Psi_k(\xi) ||_{L_2}$$

$$\boxed{c_k = \frac{\langle f(\xi)\Psi_k(\xi)\rangle}{||\Psi_k||^2}}$$

$$\langle f(\xi)\Psi_k(\xi)\rangle = \int_\xi f(\xi)\Psi_k(\xi)\pi(\xi)d\xi$$



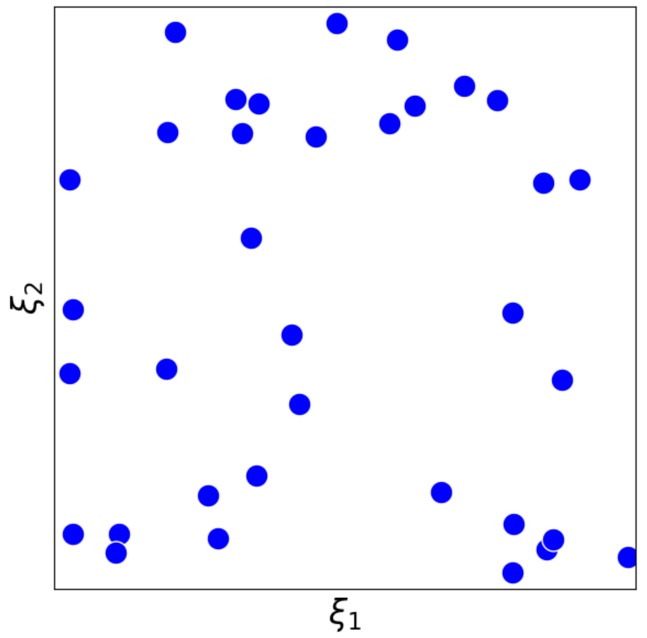

$$\approx \sum_{q=1}^{Q} w_q f(\xi^{(q)})\Psi_k(\xi^{(q)})$$

## Regression

$$argmin_c || f(\xi) - \sum_k c_k \Psi_k(\xi) ||_{\ell_2}$$

$$\boxed{c = (\Psi^T\Psi)^{-1}\Psi^T f}$$

$$f = (f(\xi^{(1)}, \ldots, \xi^{(N)})$$




$$\Psi_{nk} = \Psi_k(\xi^{(n)})$$



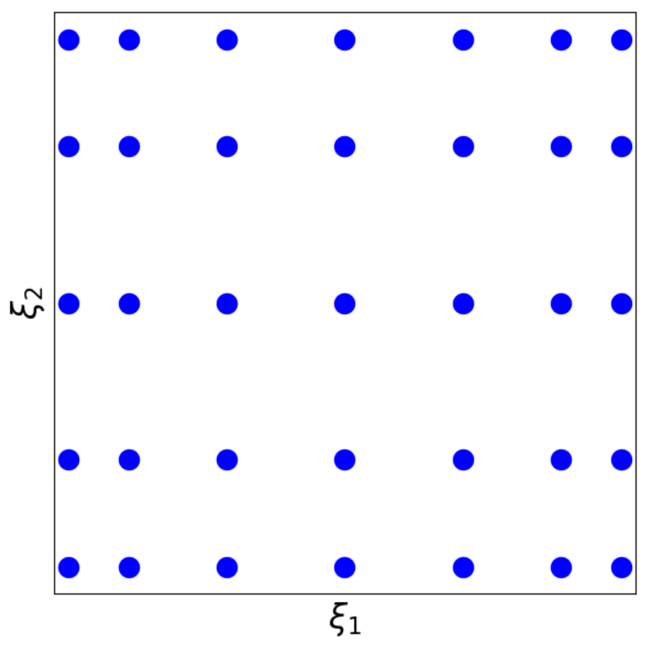
K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Non-intrusive: Projection vs Regression

$$f(X(\xi)) \approx \sum_k c_k \Psi_k(\xi)$$

## Projection

- Full tensor-product quadrature won't scale with dimensionality
- Can integrate with Monte-Carlo: but inherits slow converges of MC methods
- Sparse quadratures: requires smoothness, negative weights non-stable
- Not robust wrt code failures (missing samples)

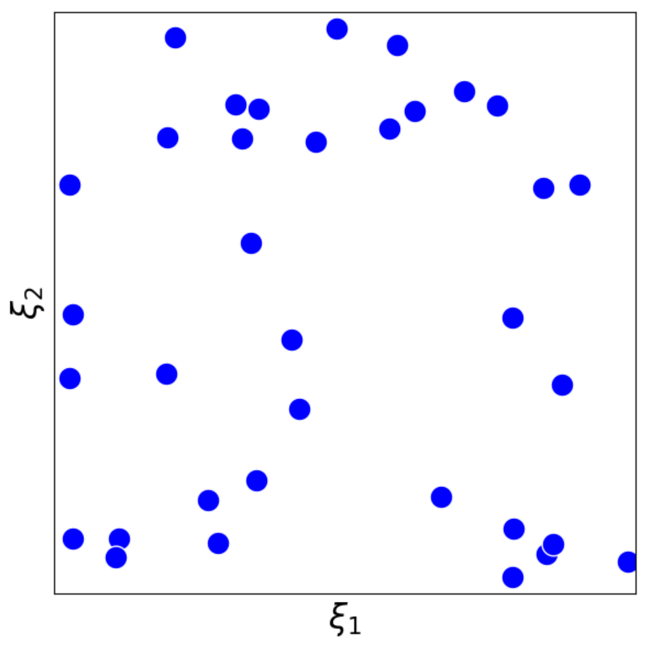



$$argmin_c || f(\xi) - \sum_k c_k \Psi_k(\xi) ||_{L_2}$$

## Regression

- Flexible sample selection
- Allows for sparse basis selection
- Allows for regularization, Bayesian extension
- Robust wrt noise and code failures

- May be prone to overfitting




$$argmin_c || f(\xi) - \sum_k c_k \Psi_k(\xi) ||_{\ell_2}$$



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Surrogate construction is the cornerstone

Input $X = \sum_k x_k \Psi_k(\xi)$

More often than not, this is a linear expansion driven by physics experts

$X = \mu + \sigma\xi$ so, $X \leftrightarrow \xi$

Output $Y = f(X) \approx \sum_k c_k \Psi_k(\xi)$ becomes a polynomial fit $Y = P_c(x)$

Complex physical model (PDE, climate, chemistry, …) $f(X) \approx P_c(X)$ Surrogate, proxy, metamodel, …

- Surrogate is constructed by sampling the full model at *training* samples $f(x^{(1)}), \ldots, f(x^{(N)})$ and performing regression/fit
- In any sample intensive task (such as uncertainty propagation, sensitivity, model calibration, etc…) the preconstructed surrogate replaces the full model $f(x)$
- Polynomial form has some advantages, but one can use higher capacity surrogate forms such as, e.g., Neural Networks $NN_w(x)$

# Fwd UQ: Global Sensitivity Analysis

- Forward model: $Y = f(X_1, X_2, \ldots, X_d)$

- Variance-based decomposition, also called Sobol sensitivity index [Sobol, 2001; Saltelli, 2010]

$$S_i = \frac{\mathbb{V}_{X_i}[\mathbb{E}_{X_{\sim i}}[Y|X_i]]}{\mathbb{V}[Y]} \qquad\qquad T_i = \frac{\mathbb{E}_{X_{\sim i}}[\mathbb{V}_{X_i}[Y|X_{\sim i}]]}{\mathbb{V}[Y]} = 1 - \frac{\mathbb{V}_{X_{\sim i}}[\mathbb{E}_{X_i}[Y|X_{\sim i}]]}{\mathbb{V}[Y]}$$

captures the fraction of variance explained by the $i$-th parameter.

- Typically the integrals $\mathbb{E}$ and $\mathbb{V}$ are computed via Monte-Carlo sampling, but …
- … polynomial chaos allows exact extraction of these indices without sampling [Crestaux, 2009].

| Param 1 | Param 2 | P 3 | P 4 | Param 5 |
|---|---|---|---|---|

Prediction Variance = Parametric Uncertainty + Data Noise

# Fwd UQ: Challenges

- Model $f(X)$ is **expensive**
  - e.g. climate model that runs days on a supercomputer
  - not enough training samples for an accurate surrogate
  - solution: Bayesian regression, surrogate itself comes with uncertainty [Sargsyan, 2017]

\+ Surrogate Error

- The forward model $f(X, \omega)$ itself is **stochastic**
  - e.g. in chemical reaction networks, molecular dynamics, etc..
  - pick smooth summary quantity $g(X) = \mathbb{E}_\omega[f(X, \omega)]$
  - solution: polynomial-chaos approach to capture intrinsic noise [Mueller, 2023]

\+ Intrinsic Noise

| Param 1 | Param 2 | P 3 | P 4 | P 5 |
|---|---|---|---|---|

Prediction Variance = Parametric Uncertainty + Data Noise



K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Challenges

- **Input** is **high-dimensional**: $f(x) = f(x_1, \ldots, x_d)$ for $d = O(100) - O(1000)$
  - large number of uncertain inputs
  - in PC case: too many polynomial bases: $\Psi_k(x_1, \ldots, x_d) = \psi_{k_1}(x_1) \times \cdots \times \psi_{k_d}(x_d)$
  - Truncation order $p$ leads to $K = (p+d)!/p!d!$ bases — grows too fast!
  - solution: sparse learning; find active low-dim subspaces [Constantine, 2015];
    $l_1$-regularization; compressed sensing [Sargsyan, 2014];
    See overview [Kontolati, 2022].

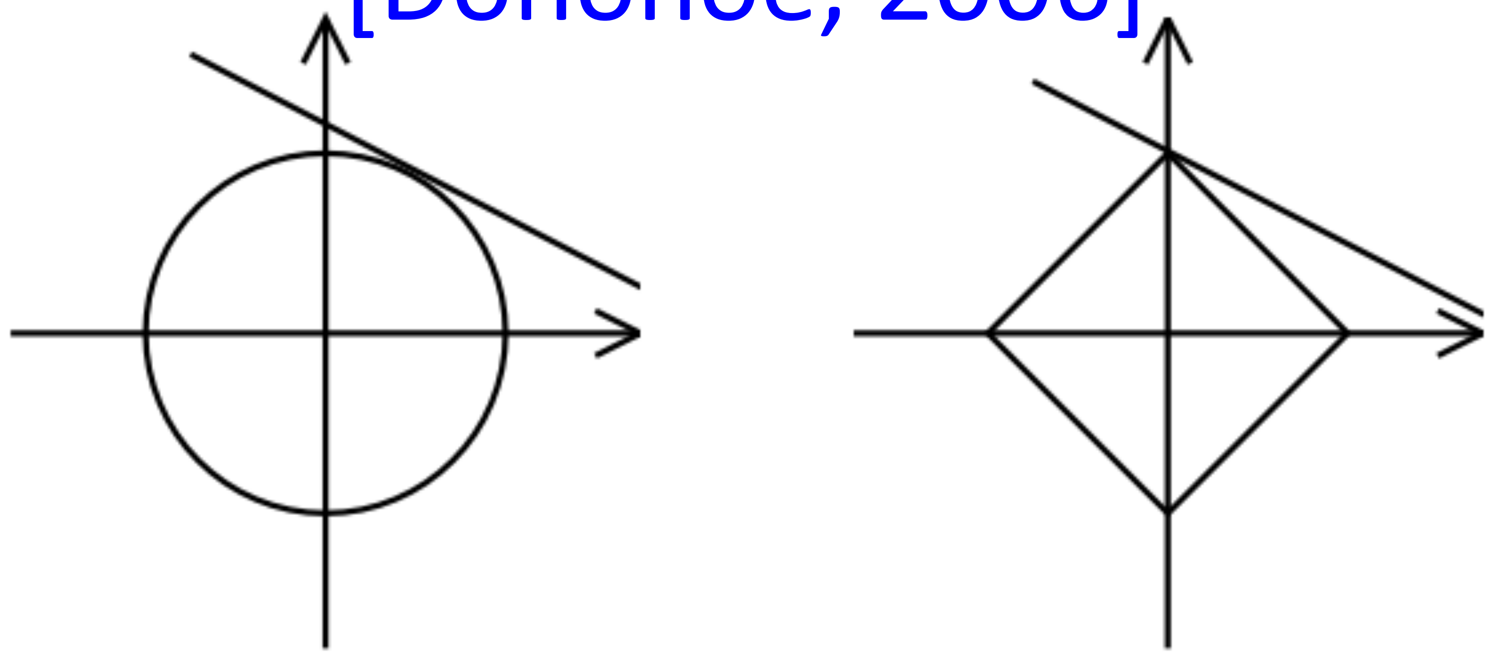
# Fwd UQ: High-D Input: Compressive sensing

Compressive sensing, LASSO, Basis Pursuit:

$$argmin_c ||y - \Psi c||_2^2 + ||c||_1$$

**Closest Convex apprx.**

Roots in sparse signal discovery literature: [Donohoe, 2006]



$$argmin_c ||y - \Psi c||_2 \text{ s.t. } ||c||_1 < \epsilon$$

$$argmin_c ||c||_1 \text{ s.t. } ||y - \Psi c||_2 < \epsilon$$

**Equivalent formulations**

**Extensions:**

- Bayesian Compressive Sensing (BCS): coeffs. with uncertainties, related to relevance vector machine (RVM) [Babacan, 2010], [Sargsyan, 2014],
- Weighted regularization: always better, with judicious choice of weights [Peng, 2013].
- Iterative growth of basis: exploits polynomial structure; increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Jakeman, 2015].


K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: Challenges

- The **output** of model $f(x)$ is **high-dimensional**
  - e.g. spatio-temporal fields in climate models (sea surface temperature, …)
  - .. or multiple observables of the model.

  - solution: reduce the dimensionality — manifold learning, diffusion maps, autoencoders, or linear, by principal components, or Karhunen-Loève expansions [Pringle, 2023; Mueller, 2025].

# Fwd UQ: 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) $\eta_m$

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

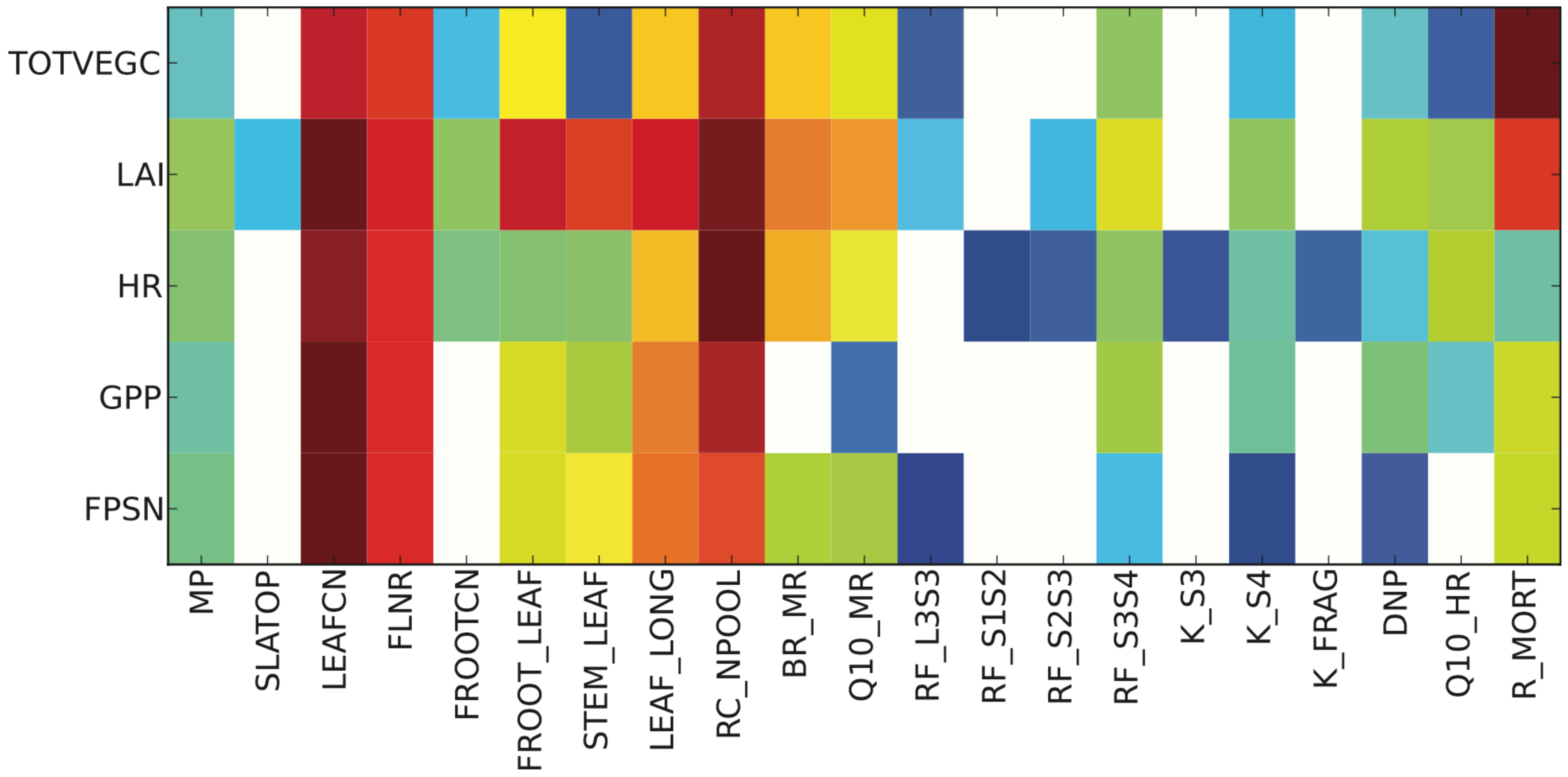
# E3SM Land Model (ELM)

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

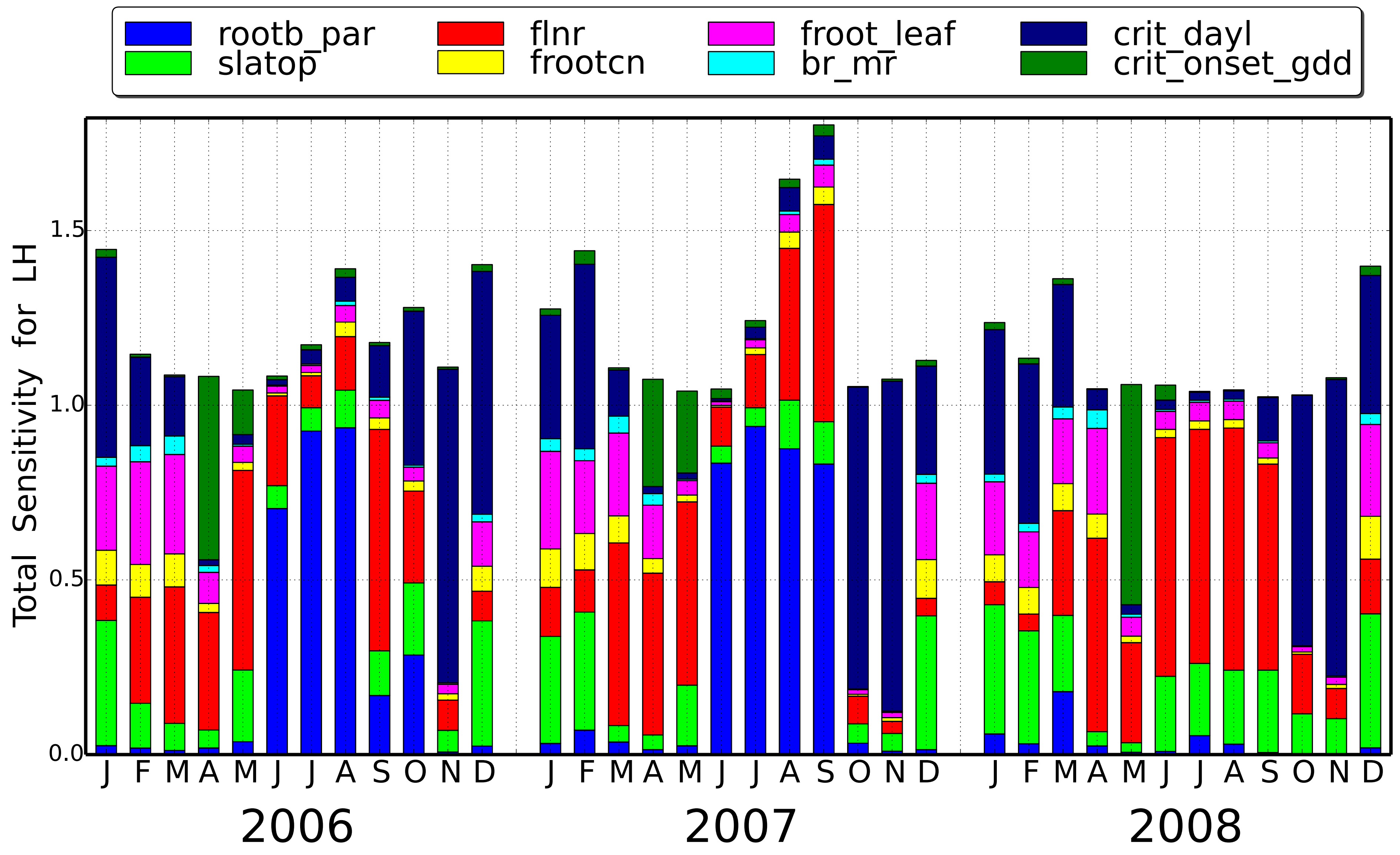
E3SM
Energy Exascale
Earth System Model

*e3sm.org*

**~50 inputs; 5 averaged outputs**



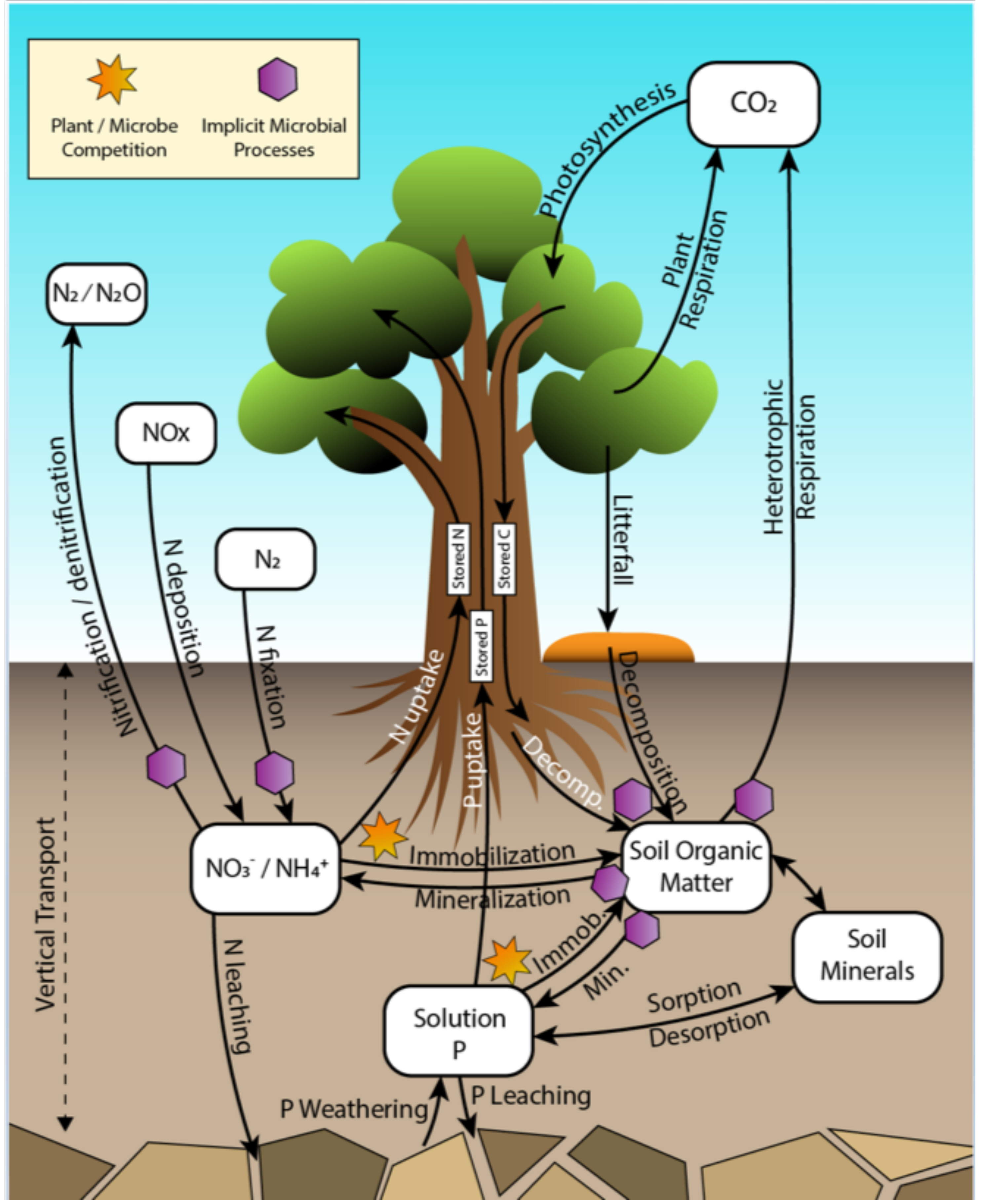

**~15 inputs; 60 temporal outputs**



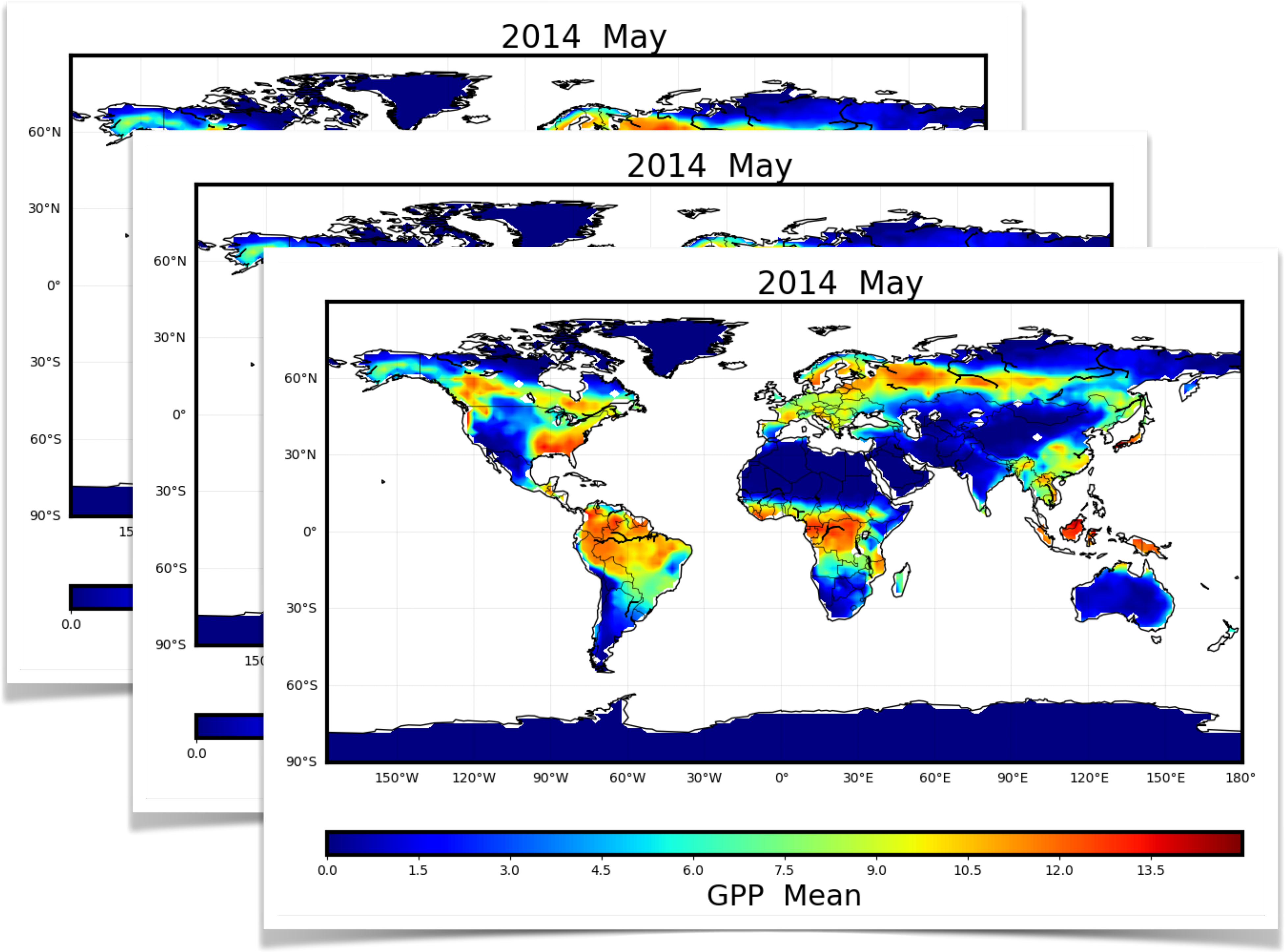



K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model: Gross Primary Productivity




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

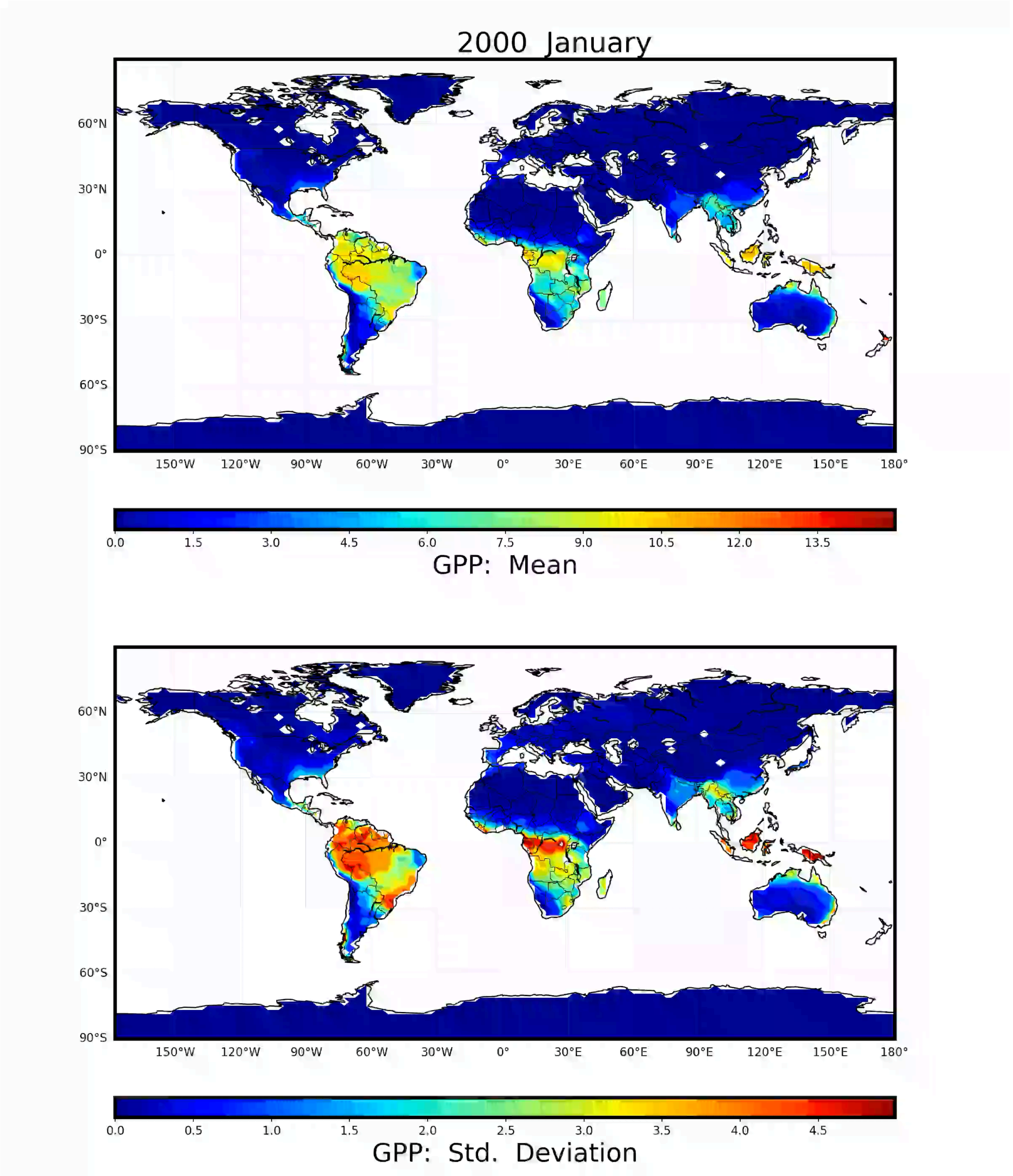



**Model Ensemble (275 samples)**

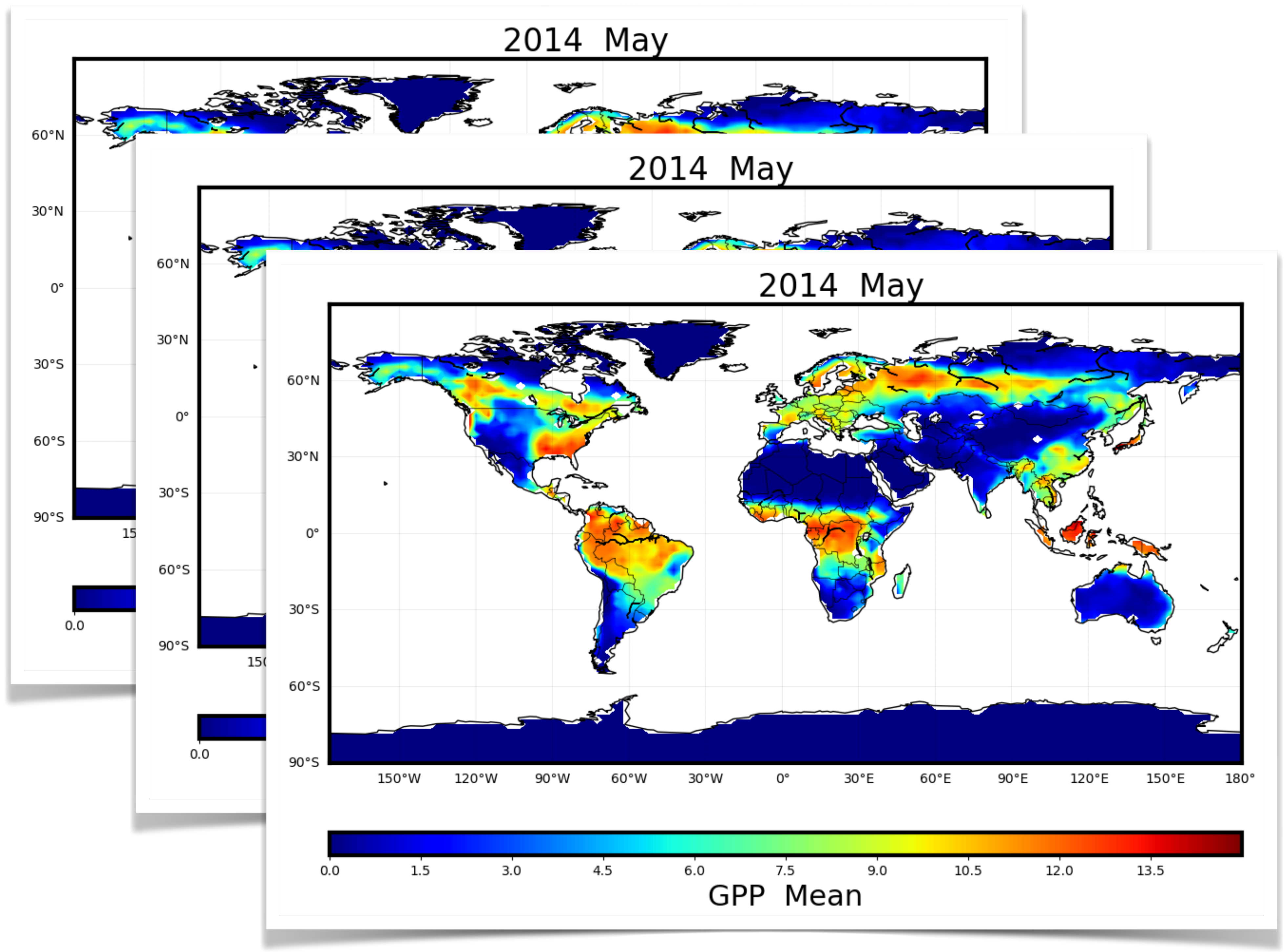


K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model: Gross Primary Productivity



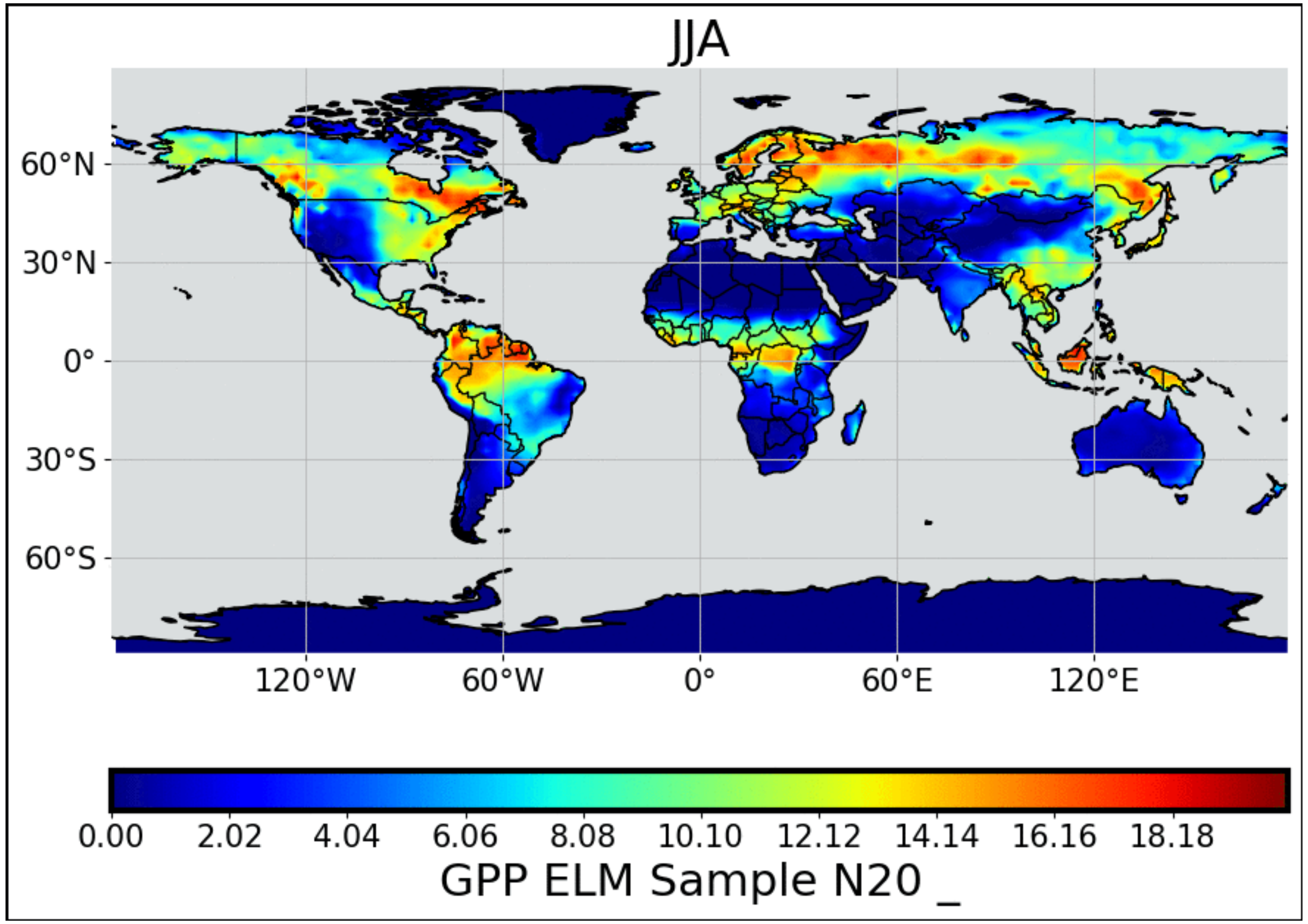

**10 inputs; spatio-temporal output 4000 cells x 180 months**



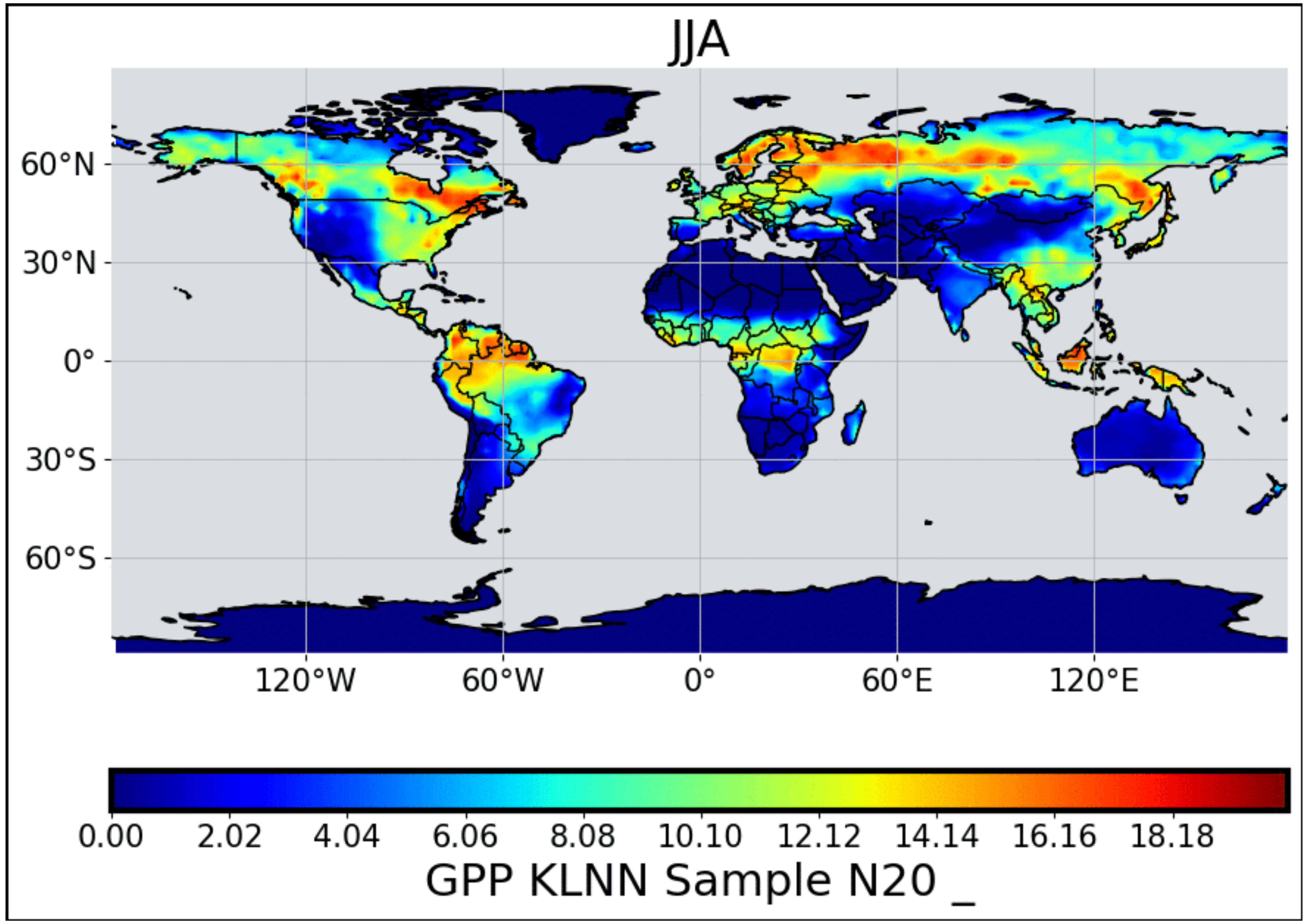

**Model Ensemble (275 samples)**

# E3SM Land Model: Gross Primary Productivity

**ELM: Single simulation several hours**

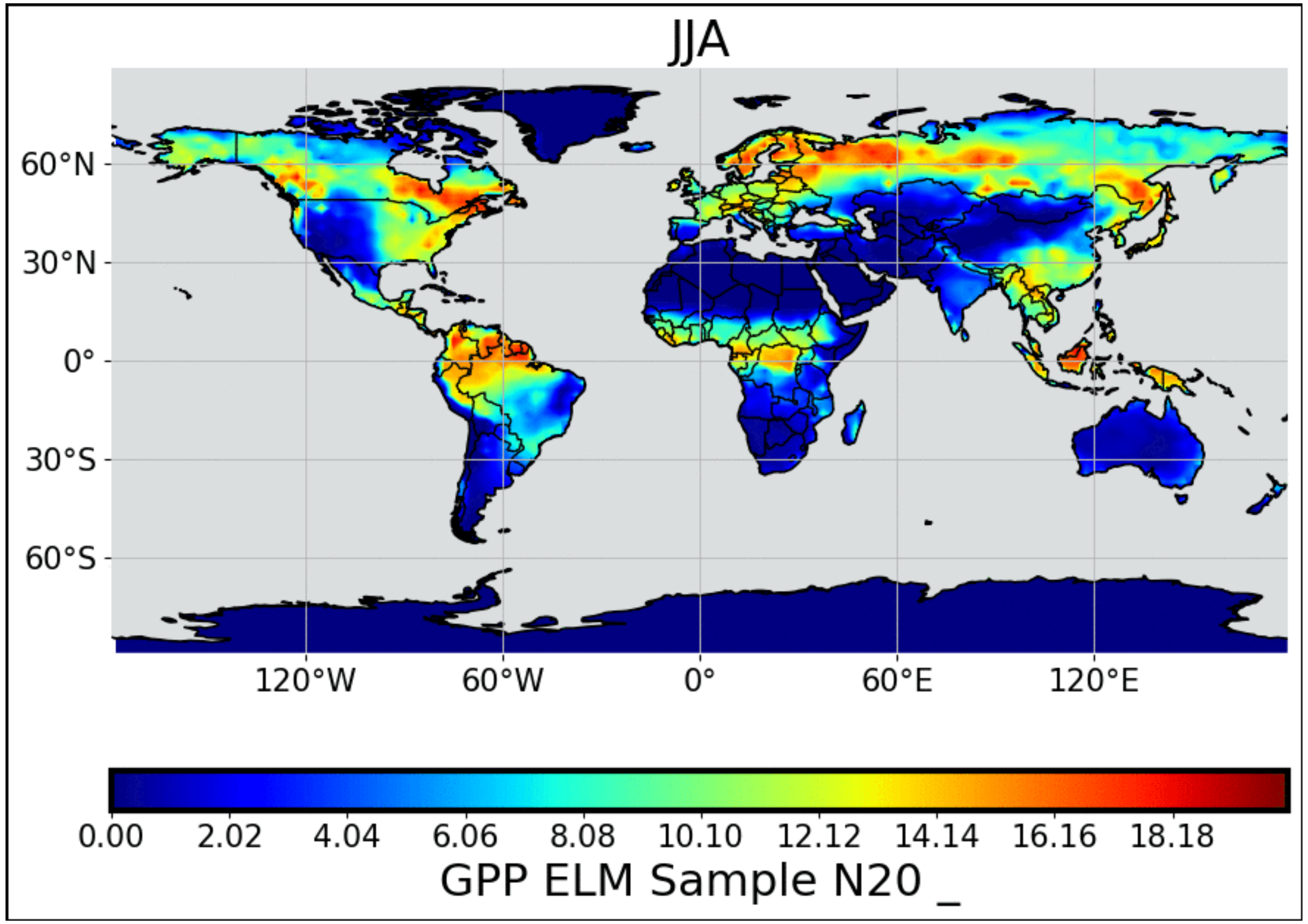



**KL+NN Surrogate: Single simulation ~1 sec**

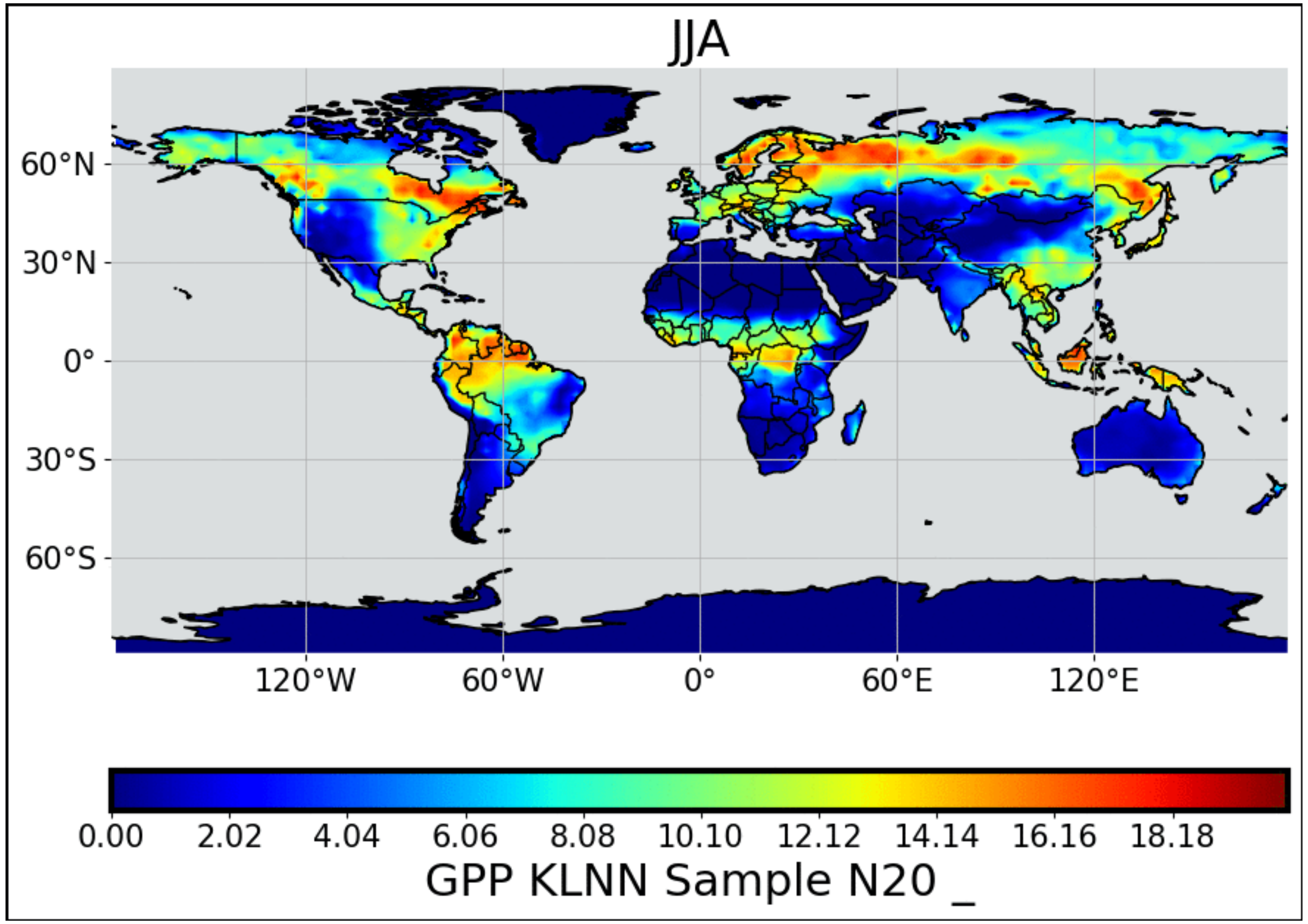


# E3SM Land Model: Gross Primary Productivity

**ELM: Single simulation several hours**




**KL+NN Surrogate: Single simulation ~1 sec**

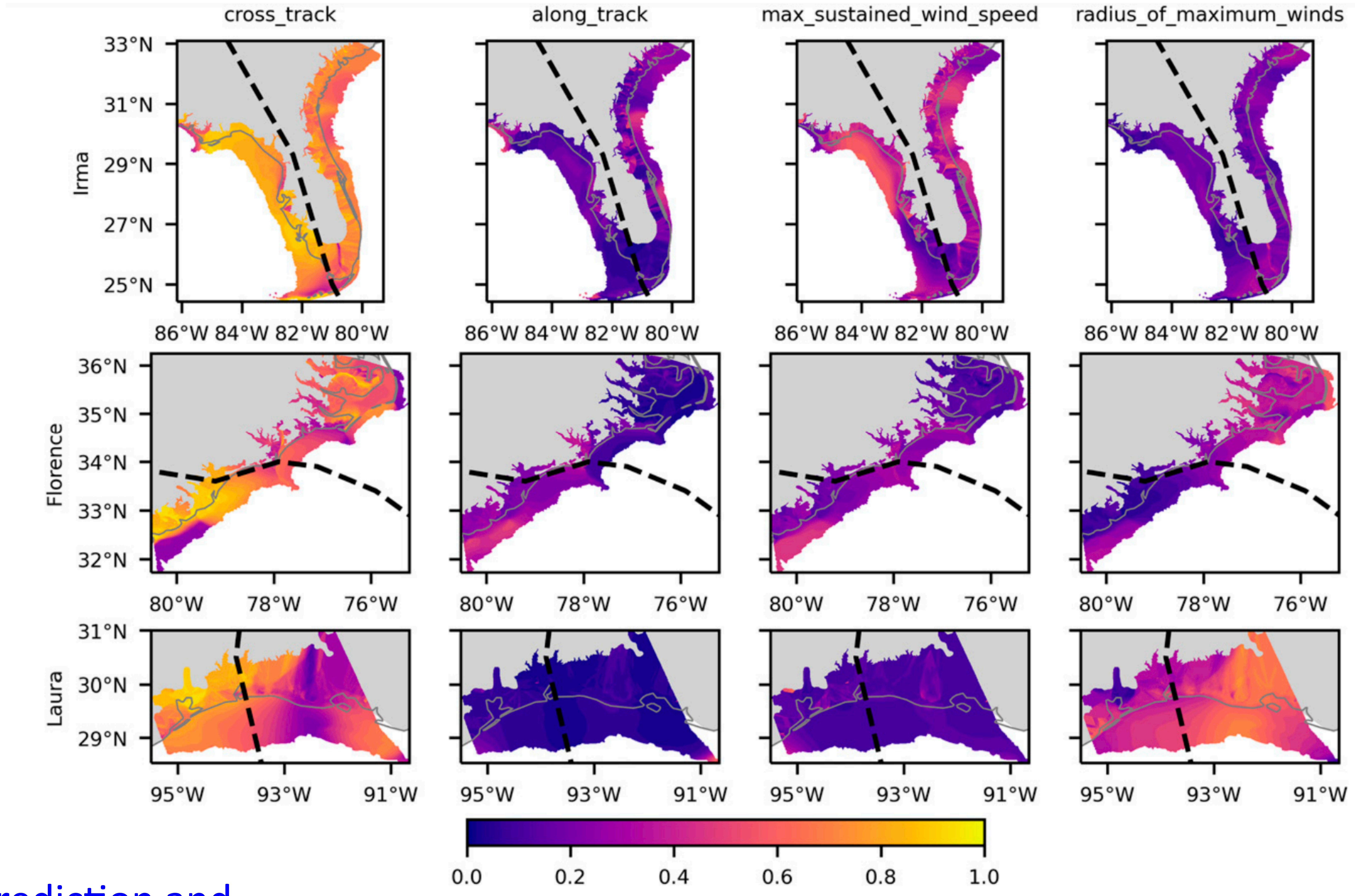





K. Sargsyan (ksargsy@sandia.gov)

# 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]

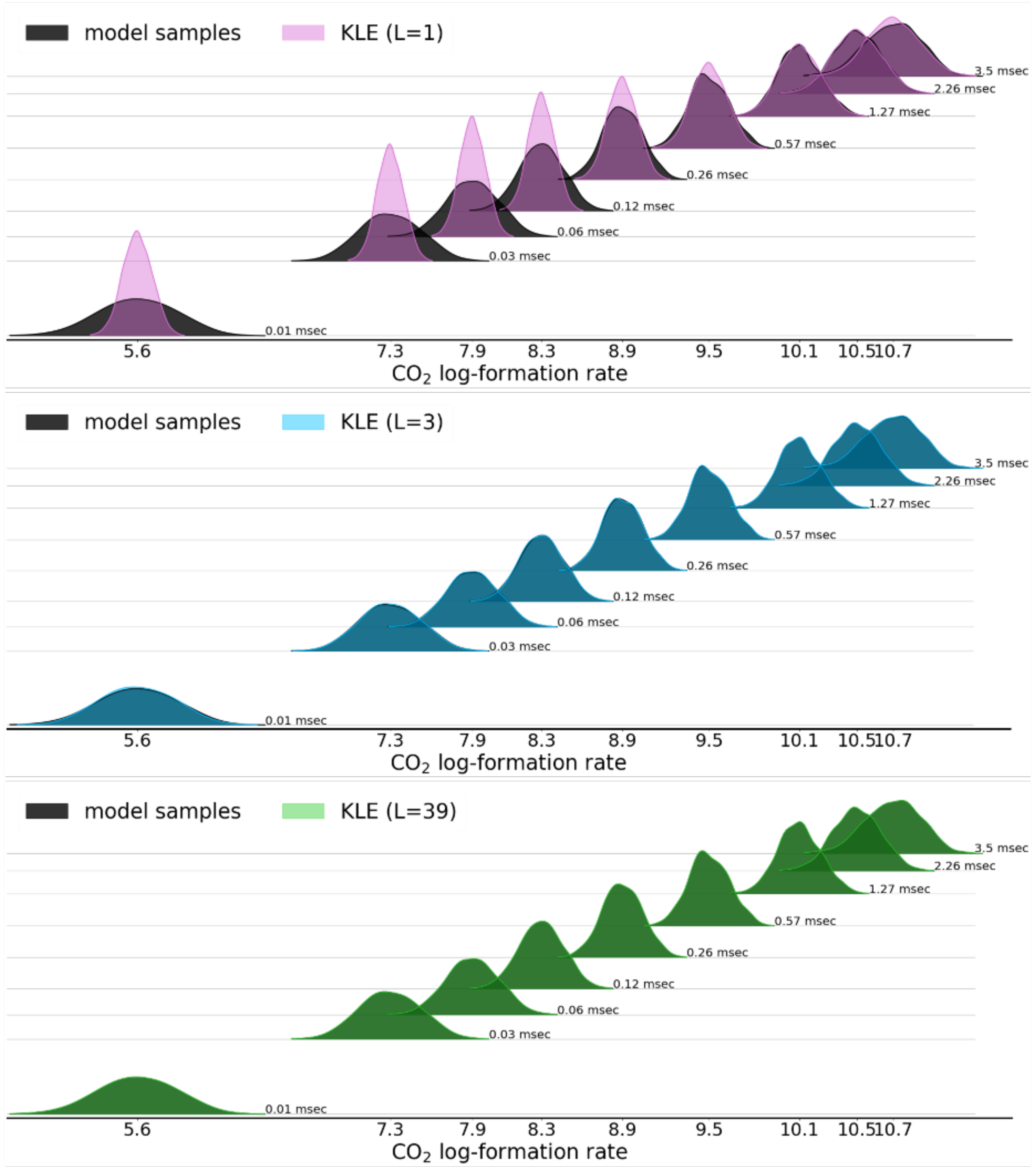


K. Sargsyan (ksargsy@sandia.gov)

# Chemical Catalysis

## $CO$ oxidation on a $RuO_2(110)$ surface

| Forward Processes | | | Reverse Processe | |
|---|---|---|---|---|
| [1] | Adsorption: | $CO \xrightarrow{k_1} CO(\mathtt{cus})$ | Desorption: | $CO(\mathtt{cus}) \xrightarrow{k_{-1}} CO$ |
| [2] | Adsorption: | $CO \xrightarrow{k_2} CO(\mathtt{br})$ | Desorption: | $CO(\mathtt{br}) \xrightarrow{k_{-2}} CO$ |
| [3] | Adsorption: | $O_2 \xrightarrow{k_3} O(\mathtt{cus}) + O(\mathtt{cus})$ | Desorption: | $O(\mathtt{cus}) + O(\mathtt{cus}) \xrightarrow{k_{-3}} O_2$ |
| [4] | Adsorption: | $O_2 \xrightarrow{k_4} O(\mathtt{br}) + O(\mathtt{br})$ | Desorption: | $O(\mathtt{br}) + O(\mathtt{br}) \xrightarrow{k_{-4}} O_2$ |
| [5] | Adsorption: | $O_2 \xrightarrow{k_5} O(\mathtt{br}) + O(\mathtt{cus})$ | Desorption: | $O(\mathtt{br}) + O(\mathtt{cus}) \xrightarrow{k_{-5}} O_2$ |
| [6] | Diffusion: | $CO(\mathtt{cus}) \xrightarrow{k_6} CO(\mathtt{cus})$ | | |
| [7] | Diffusion: | $CO(\mathtt{br}) \xrightarrow{k_7} CO(\mathtt{br})$ | | |
| [8] | Diffusion: | $CO(\mathtt{cus}) \xrightarrow{k_8} CO(\mathtt{br})$ | Diffusion: | $CO(\mathtt{br}) \xrightarrow{k_{-8}} CO(\mathtt{cus})$ |
| [9] | Diffusion: | $O(\mathtt{cus}) \xrightarrow{k_9} O(\mathtt{cus})$ | | |
| [10] | Diffusion: | $O(\mathtt{br}) \xrightarrow{k_{10}} O(\mathtt{br})$ | | |
| [11] | Diffusion: | $O(\mathtt{cus}) \xrightarrow{k_{11}} O(\mathtt{br})$ | Diffusion: | $O(\mathtt{br}) \xrightarrow{k_{-11}} O(\mathtt{cus})$ |
| [12] | Formation: | $CO(\mathtt{cus}) + O(\mathtt{cus}) \xrightarrow{k_{12}} CO_2$ | | |
| [13] | Formation: | $CO(\mathtt{br}) + O(\mathtt{br}) \xrightarrow{k_{13}} CO_2$ | | |
| [14] | Formation: | $CO(\mathtt{br}) + O(\mathtt{cus}) \xrightarrow{k_{14}} CO_2$ | | |
| [15] | Formation: | $CO(\mathtt{cus}) + O(\mathtt{br}) \xrightarrow{k_{15}} CO_2$ | | |

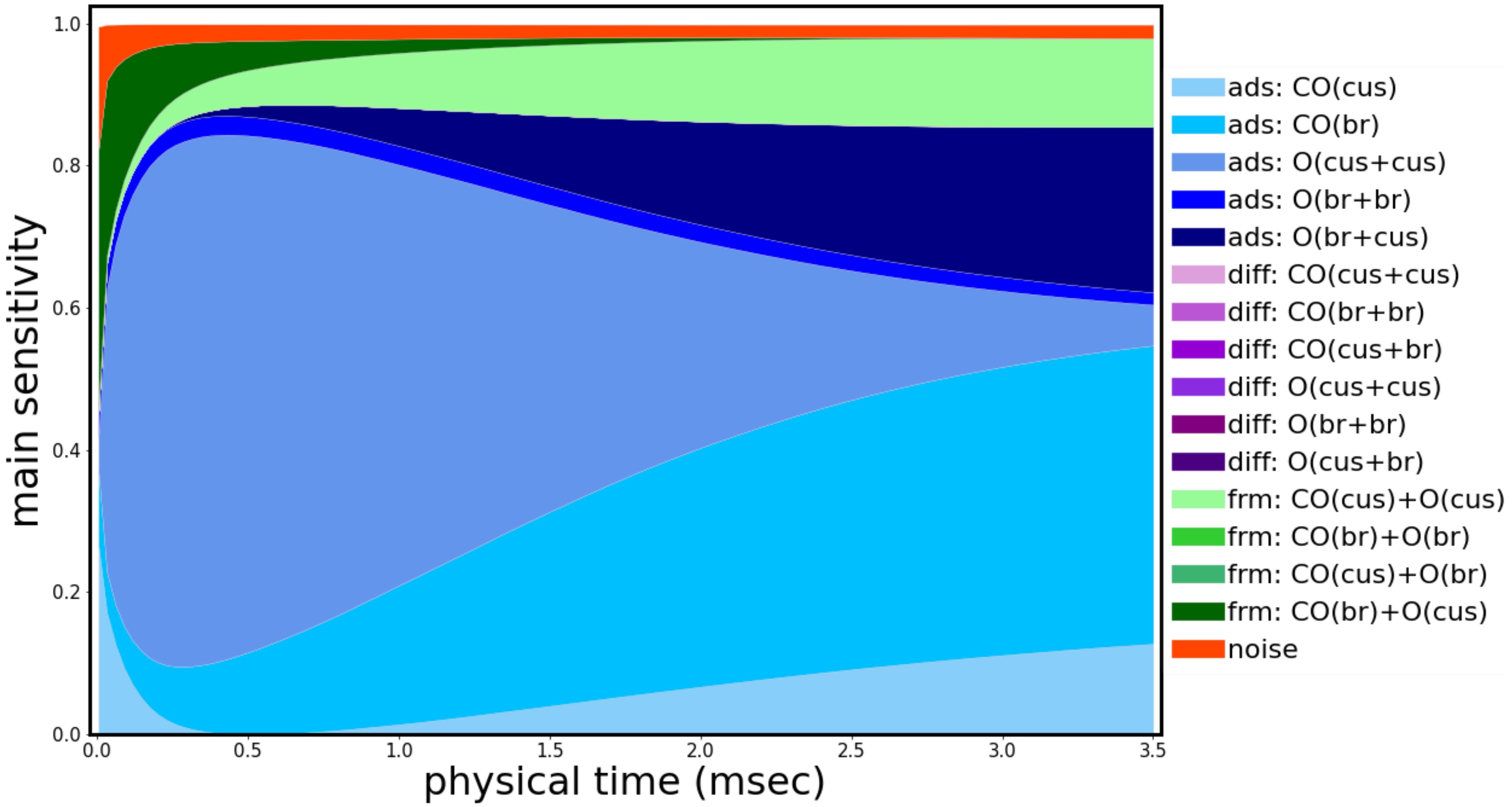



[J. Mueller, K. Sargsyan, C. Daniels, H. Najm, “Polynomial Chaos Surrogate Construction for Random Fields with Parametric Uncertainty”, *SIAM/ASA JUQ*, 2025]

# Chemical Catalysis

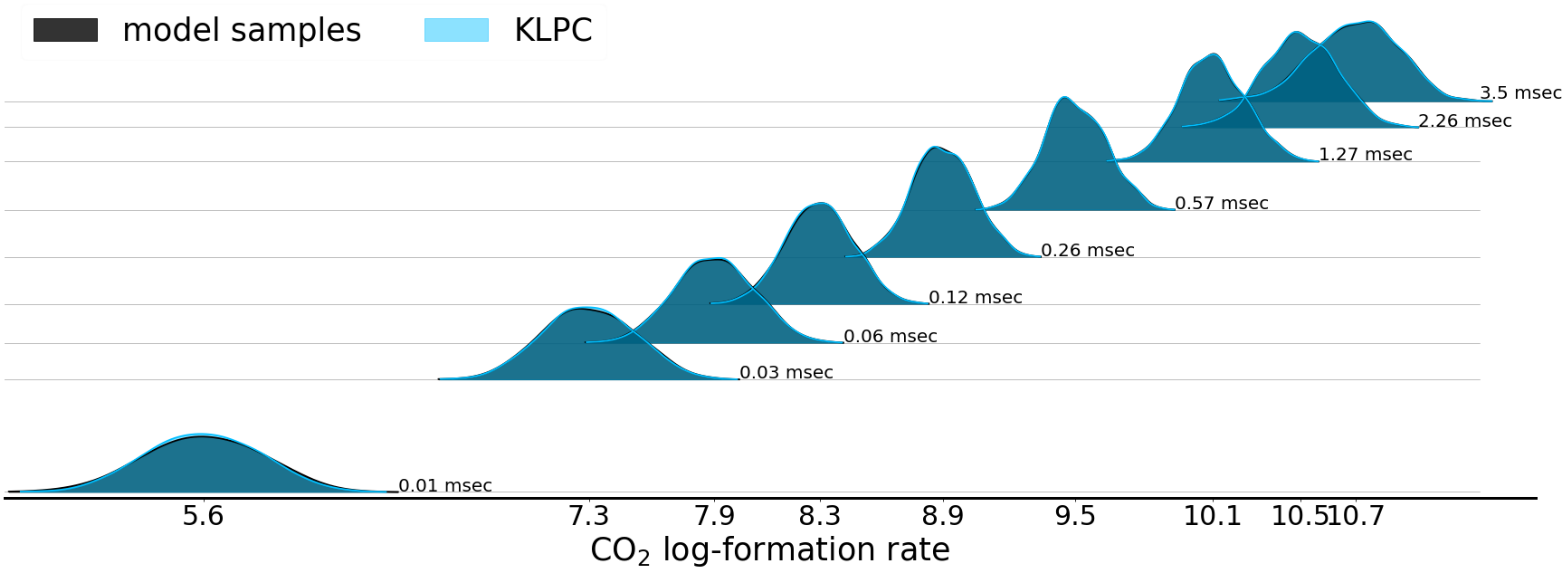
## $CO$ oxidation on a $RuO_2(110)$ surface

| | Forward Processes | | Reverse Processe | |
|---|---|---|---|---|
| [1] | Adsorption: | $\mathrm{CO} \xrightarrow{k_1} \mathrm{CO}(\mathtt{cus})$ | Desorption: | $\mathrm{CO}(\mathtt{cus}) \xrightarrow{k_{-1}} \mathrm{CO}$ |
| [2] | Adsorption: | $\mathrm{CO} \xrightarrow{k_2} \mathrm{CO}(\mathtt{br})$ | Desorption: | $\mathrm{CO}(\mathtt{br}) \xrightarrow{k_{-2}} \mathrm{CO}$ |
| [3] | Adsorption: | $\mathrm{O}_2 \xrightarrow{k_3} \mathrm{O}(\mathtt{cus}) + \mathrm{O}(\mathtt{cus})$ | Desorption: | $\mathrm{O}(\mathtt{cus}) + \mathrm{O}(\mathtt{cus}) \xrightarrow{k_{-3}} \mathrm{O}_2$ |
| [4] | Adsorption: | $\mathrm{O}_2 \xrightarrow{k_4} \mathrm{O}(\mathtt{br}) + \mathrm{O}(\mathtt{br})$ | Desorption: | $\mathrm{O}(\mathtt{br}) + \mathrm{O}(\mathtt{br}) \xrightarrow{k_{-4}} \mathrm{O}_2$ |
| [5] | Adsorption: | $\mathrm{O}_2 \xrightarrow{k_5} \mathrm{O}(\mathtt{br}) + \mathrm{O}(\mathtt{cus})$ | Desorption: | $\mathrm{O}(\mathtt{br}) + \mathrm{O}(\mathtt{cus}) \xrightarrow{k_{-5}} \mathrm{O}_2$ |
| [6] | Diffusion: | $\mathrm{CO}(\mathtt{cus}) \xrightarrow{k_6} \mathrm{CO}(\mathtt{cus})$ | | |
| [7] | Diffusion: | $\mathrm{CO}(\mathtt{br}) \xrightarrow{k_7} \mathrm{CO}(\mathtt{br})$ | | |
| [8] | Diffusion: | $\mathrm{CO}(\mathtt{cus}) \xrightarrow{k_8} \mathrm{CO}(\mathtt{br})$ | Diffusion: | $\mathrm{CO}(\mathtt{br}) \xrightarrow{k_{-8}} \mathrm{CO}(\mathtt{cus})$ |
| [9] | Diffusion: | $\mathrm{O}(\mathtt{cus}) \xrightarrow{k_9} \mathrm{O}(\mathtt{cus})$ | | |
| [10] | Diffusion: | $\mathrm{O}(\mathtt{br}) \xrightarrow{k_{10}} \mathrm{O}(\mathtt{br})$ | | |
| [11] | Diffusion: | $\mathrm{O}(\mathtt{cus}) \xrightarrow{k_{11}} \mathrm{O}(\mathtt{br})$ | Diffusion: | $\mathrm{O}(\mathtt{br}) \xrightarrow{k_{-11}} \mathrm{O}(\mathtt{cus})$ |
| [12] | Formation: | $\mathrm{CO}(\mathtt{cus}) + \mathrm{O}(\mathtt{cus}) \xrightarrow{k_{12}} \mathrm{CO}_2$ | | |
| [13] | Formation: | $\mathrm{CO}(\mathtt{br}) + \mathrm{O}(\mathtt{br}) \xrightarrow{k_{13}} \mathrm{CO}_2$ | | |
| [14] | Formation: | $\mathrm{CO}(\mathtt{br}) + \mathrm{O}(\mathtt{cus}) \xrightarrow{k_{14}} \mathrm{CO}_2$ | | |
| [15] | Formation: | $\mathrm{CO}(\mathtt{cus}) + \mathrm{O}(\mathtt{br}) \xrightarrow{k_{15}} \mathrm{CO}_2$ | | |




[J. Mueller, K. Sargsyan, C. Daniels, H. Najm, "Polynomial Chaos Surrogate Construction for Random Fields with Parametric Uncertainty", *SIAM/ASA JUQ*, 2025]

# Chemical Catalysis




[J. Mueller, K. Sargsyan, C. Daniels, H. Najm, "Polynomial Chaos Surrogate Construction for Random Fields with Parametric Uncertainty", *SIAM/ASA JUQ*, 2025]

# Fwd UQ: General Recap

- Essentially a parametric study over uncertain model inputs (supervised ML)
- Global sensitivity / variance decomposition is a bi-product
- Polynomial Chaos is a major tool


K. Sargsyan (ksargsy@sandia.gov)

# Fwd UQ: General Recap

- Essentially a parametric study over uncertain model inputs (supervised ML)
- Global sensitivity / variance decomposition is a bi-product
- Polynomial Chaos is a major tool

Questions to consider:

- How many input parameters?
- How many output QoIs?
- How expensive is the model? How many training simulations?
- How noisy is the model? Intrinsic noise? Code failures / fault-tolerance?

Other enhancements (not covered today):

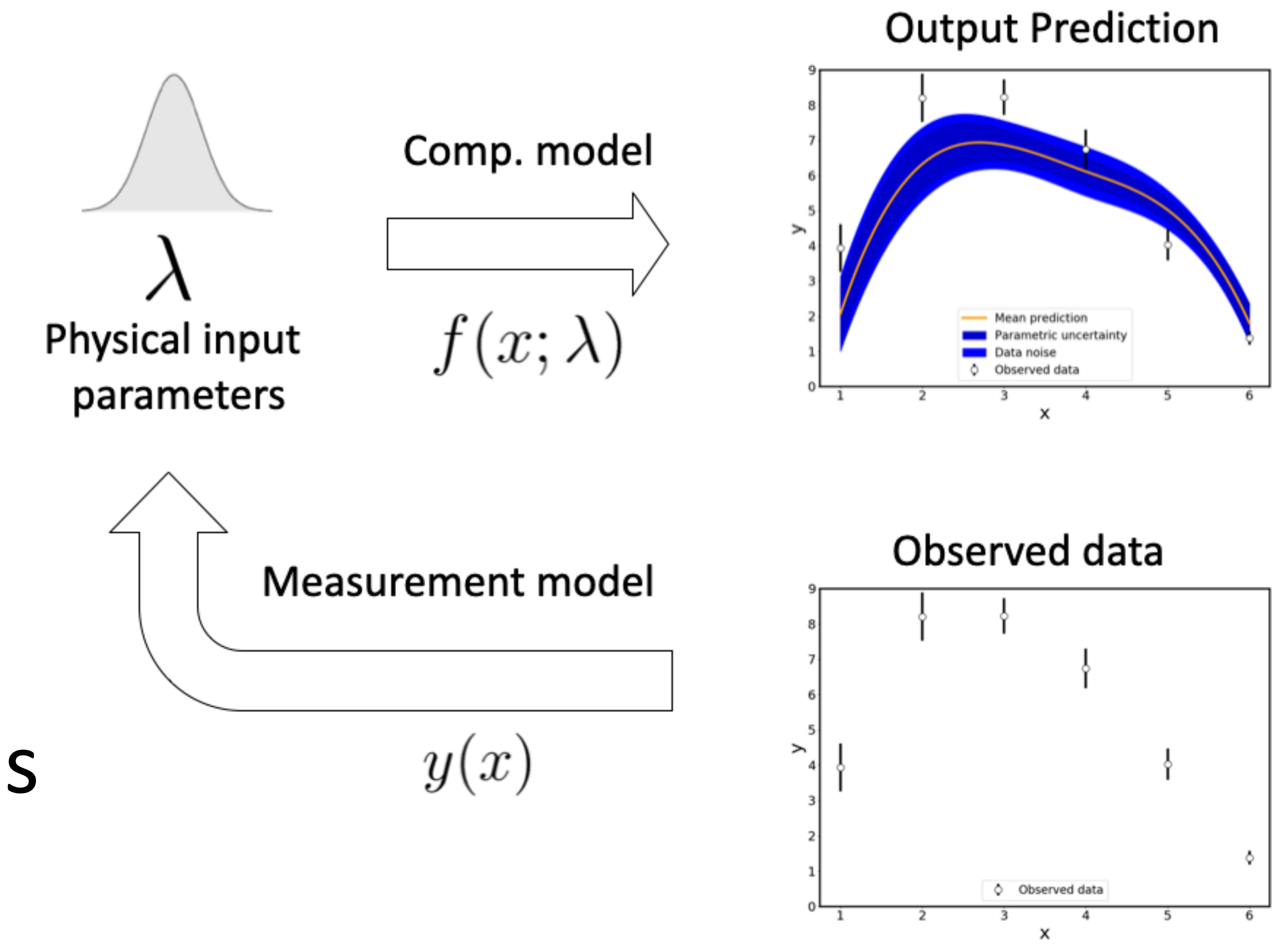
- Multilevel/multifidelity: optimal combinations of coarse/fine mesh and low/high fidelity
- Low-rank expansions: CP tensor decomposition, tensor trains (TT) …
- Nested sampling schemes; Nonisotropic sparse quadrature
- Adaptive sampling, active learning


K. Sargsyan (ksargsy@sandia.gov)

# Inverse UQ

# Inverse UQ

- Compare observational/measurement data with the model
- Tune model parameters (and sometimes model form) to fit the data




- Bayesian methods are best suited for probabilistic inverse problems
  - Meaningful aggregation of various sources of uncertainty
  - Rigorous mathematical footing

## 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 $y_i = f(x_i; \lambda) + \epsilon_i$

[Tarantola, 2005]

# Inverse UQ

- Prior : knowledge of $\lambda$ before seeing data (expert opinion, previous analysis, etc…)
- Likelihood : forward model and measurement noise
- Posterior : updated knowledge of $\lambda$, combining the prior and the likelihood
- Evidence : normalizing constant, useful for model selection, not for parameter estimation

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 $y_i = f(x_i; \lambda) + \epsilon_i$


K. Sargsyan (ksargsy@sandia.gov)

# 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 $\epsilon_i$ 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 $\lambda$-space.
- **Likelihood** is key:
  - It incorporates statistical assumptions about the discrepancy between model and data.
  - It requires model evaluation at a proposed parameter value $\lambda$.

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

# Inv UQ: Model Comparison and Selection

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

- Evidence is not relevant for parameter estimation, but…
- It becomes crucial for model selection
- Consider a set of models $\{M_1, M_2, \dots\}$
- Evidence (aka marginal likelihood) is defined as

$$p(y \,|\, M_k) = \int p(y \,|\, \lambda, M_k) p(\lambda \,|\, M_k) d\lambda$$

- Compromise between fitting data and model complexity:

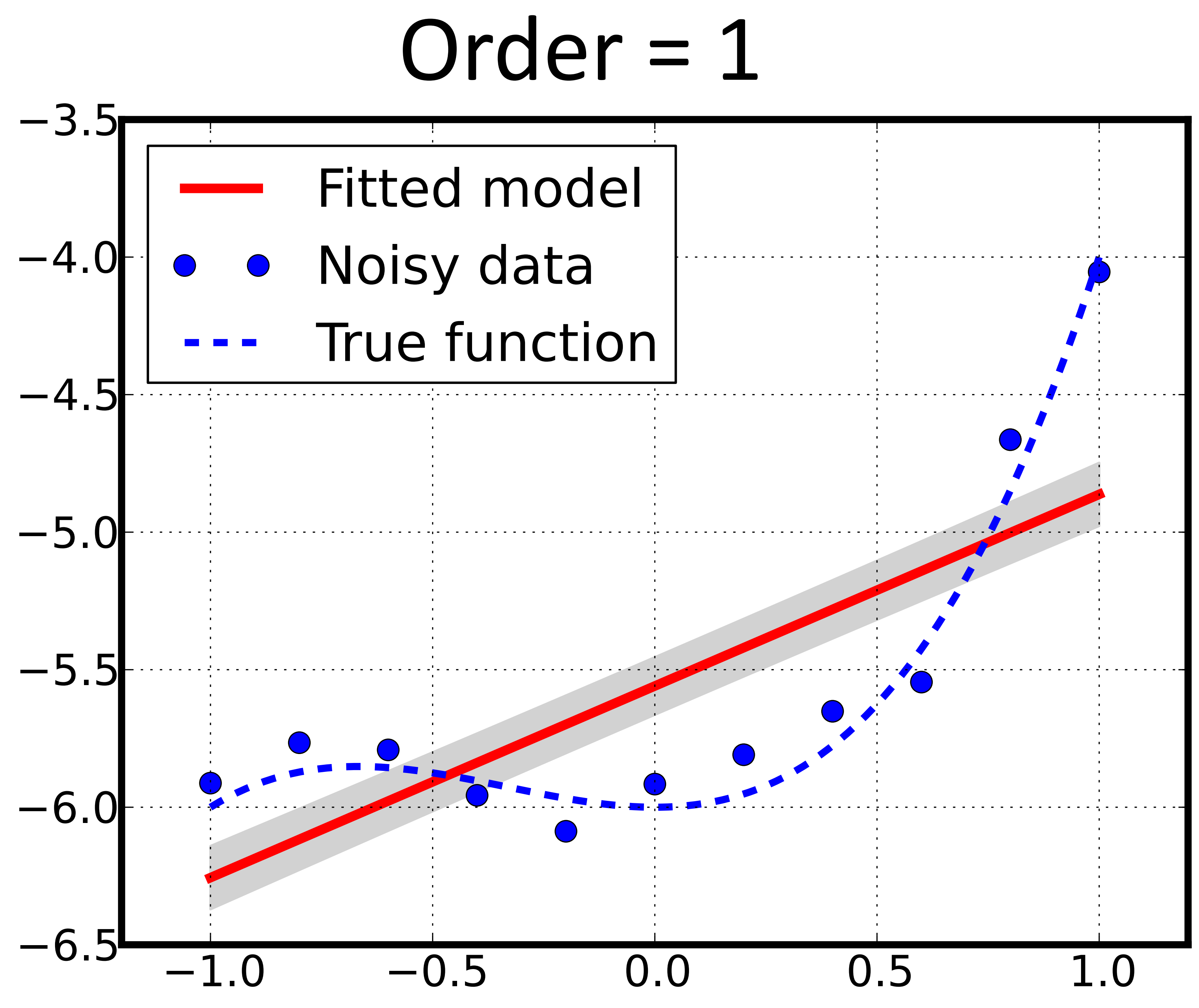
  Occam's razor principle… helps avoid overfitting

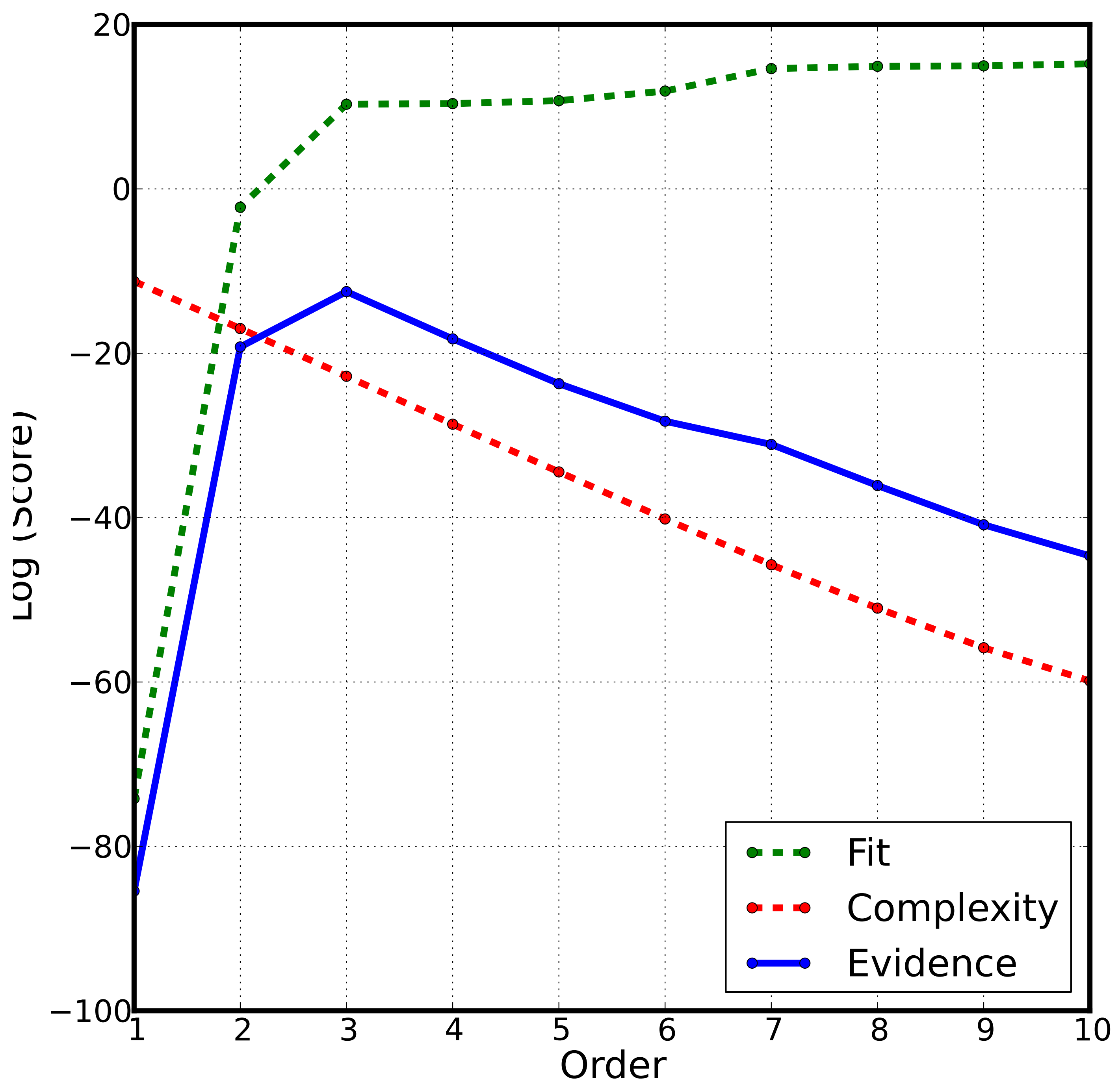
- Model selection: Choose model with maximal evidence
- Model comparison: compute Bayes Factor $BF_{21} = \dfrac{p(y \,|\, M_2)}{p(y \,|\, M_1)}$

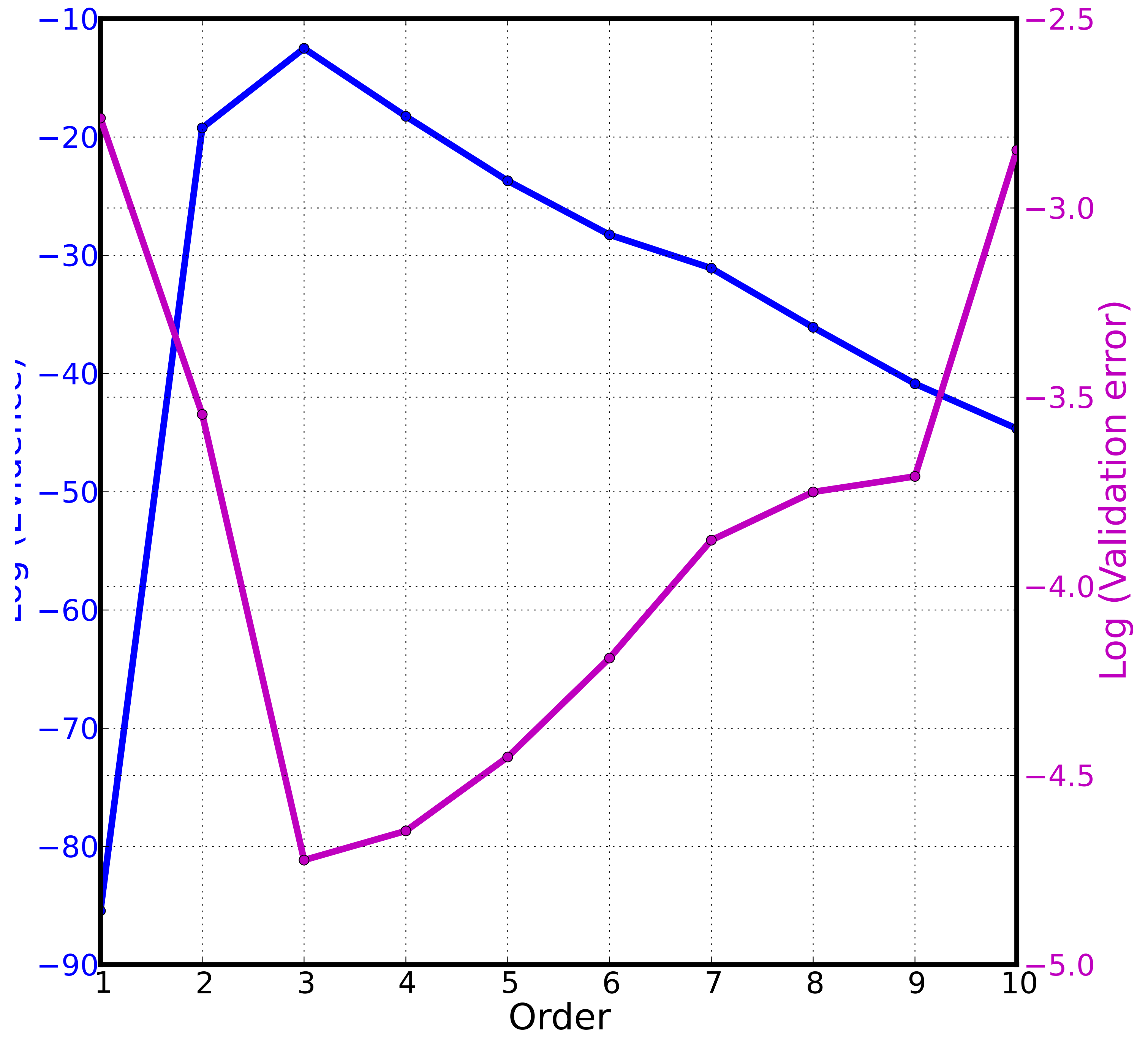

K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: Model Evidence = Fit + Complexity

$$\log p(y) = \int \log p(y) p(\lambda|y) d\lambda = \int \log \left[\frac{p(y|\lambda)p(\lambda)}{p(\lambda|y)}\right] p(\lambda|y) d\lambda = \underbrace{\int \log p(y|\lambda) p(\lambda|y) d\lambda}_{Fit} \quad \underbrace{- \int \log \left[\frac{p(\lambda|y)}{p(\lambda)}\right] p(\lambda|y) d\lambda}_{Complexity}$$

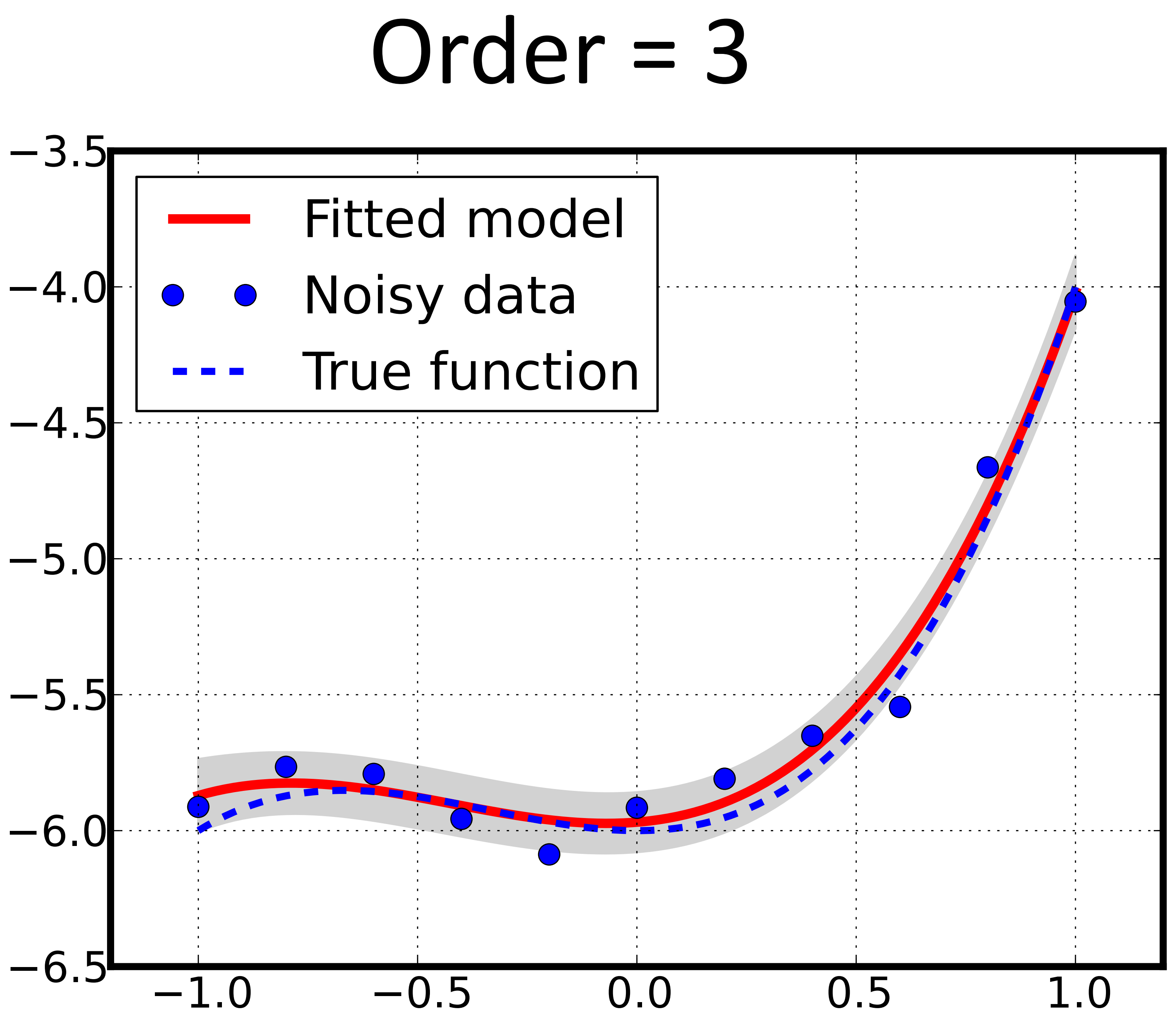


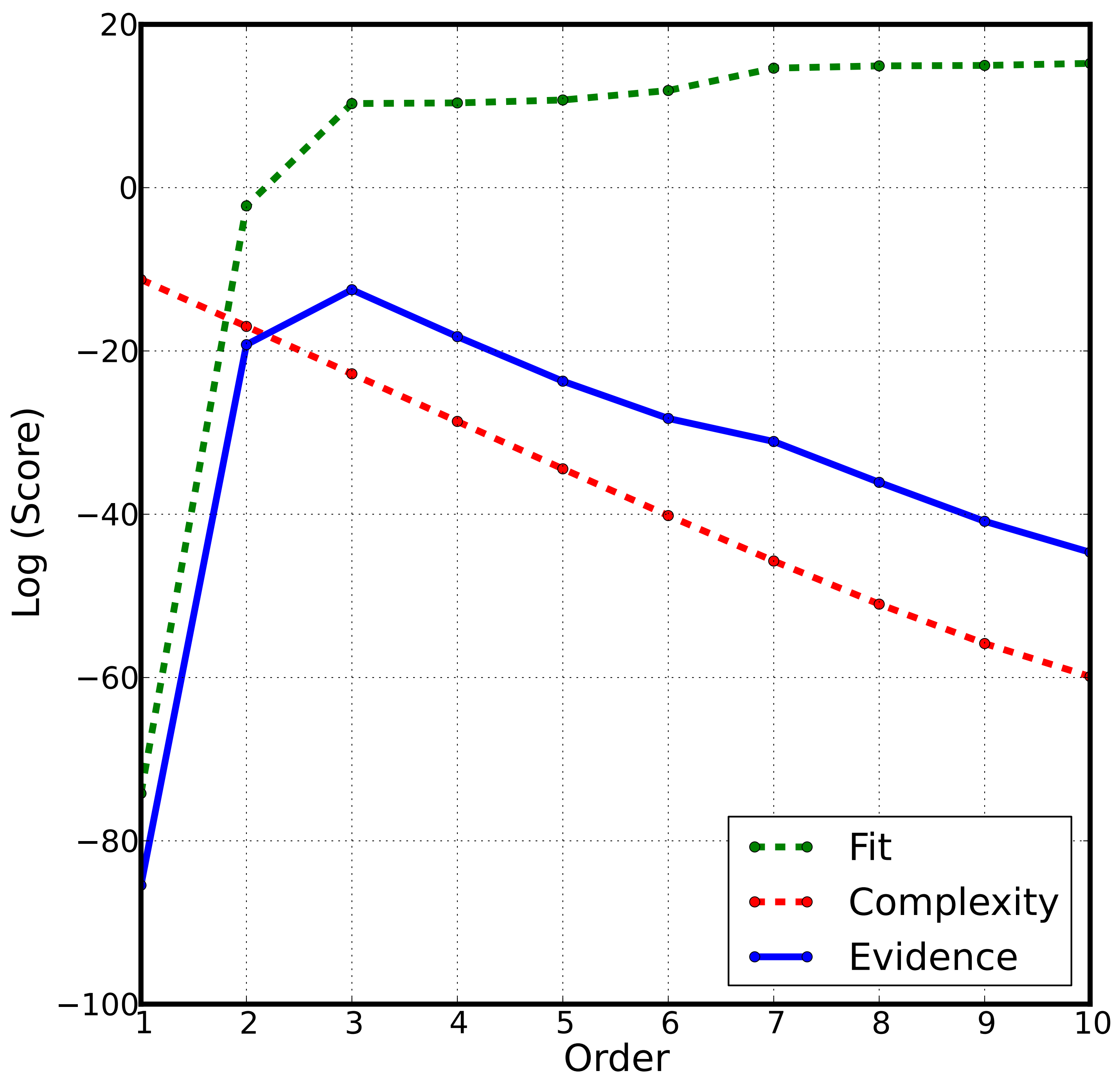



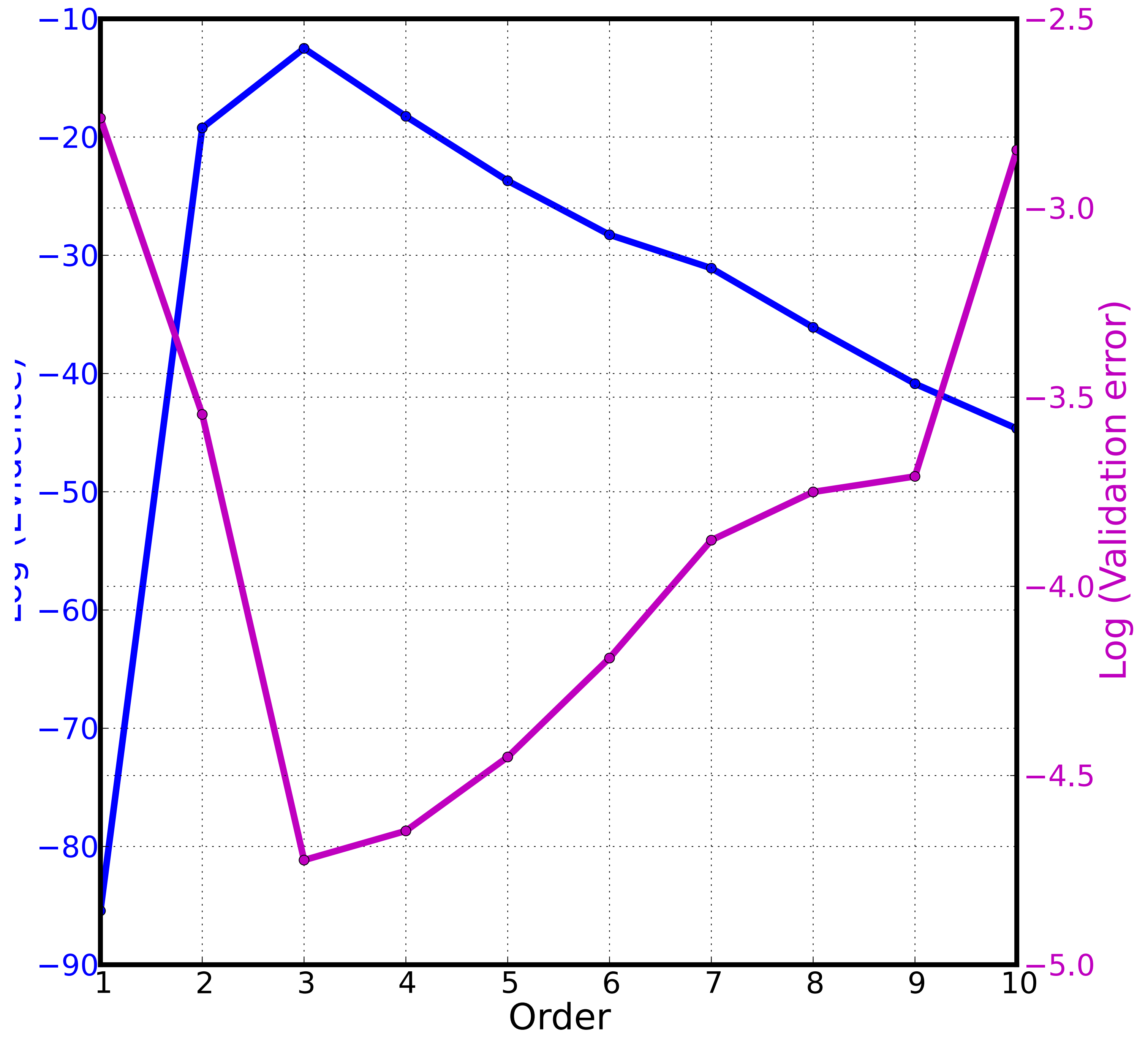



# Inv UQ: Model Evidence = Fit + Complexity

$$\log p(y) = \int \log p(y) p(\lambda|y) d\lambda = \int \log \left[\frac{p(y|\lambda)p(\lambda)}{p(\lambda|y)}\right] p(\lambda|y) d\lambda = \underbrace{\int \log p(y|\lambda) p(\lambda|y) d\lambda}_{Fit} \underbrace{- \int \log \left[\frac{p(\lambda|y)}{p(\lambda)}\right] p(\lambda|y) d\lambda}_{Complexity}$$



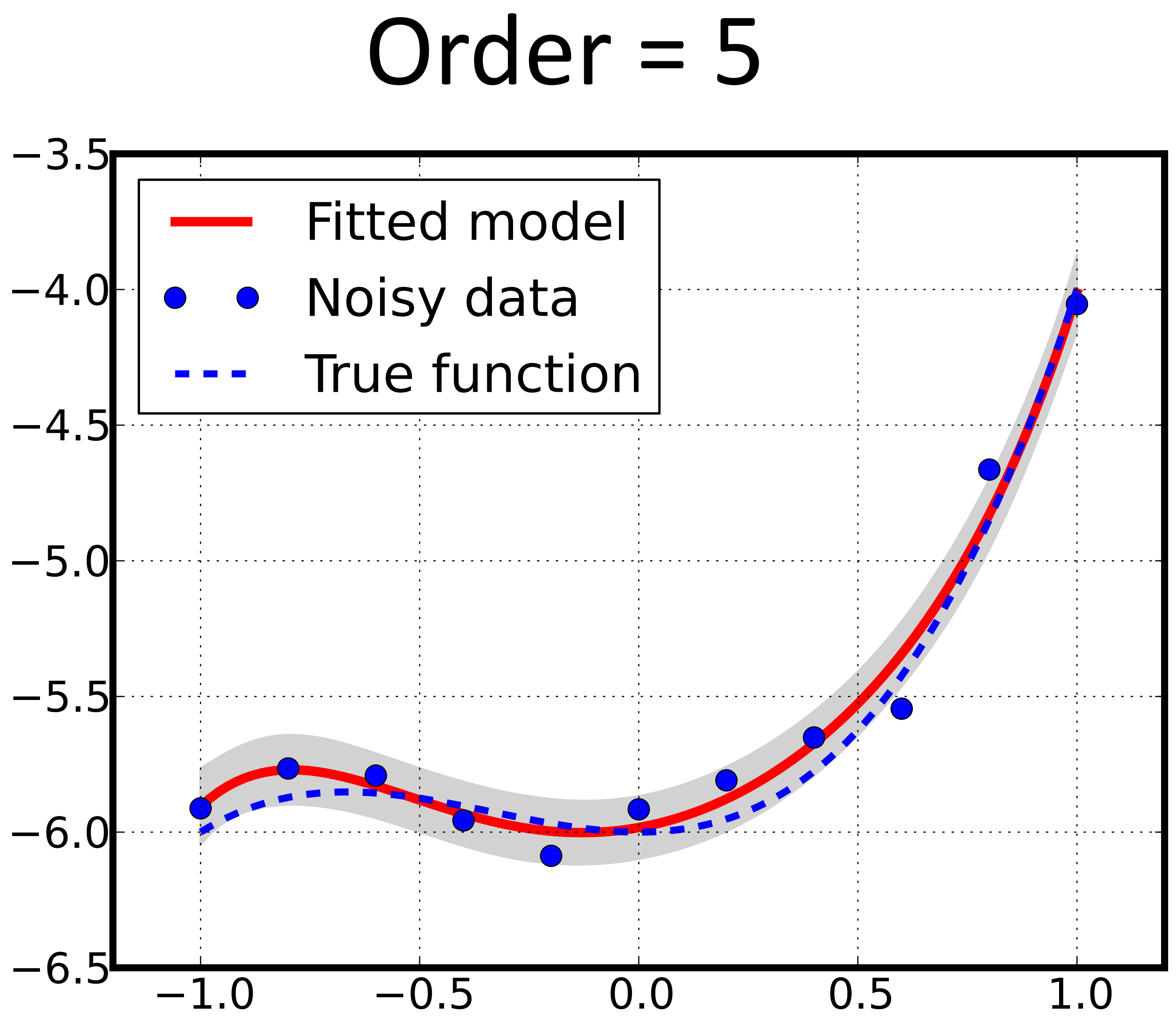



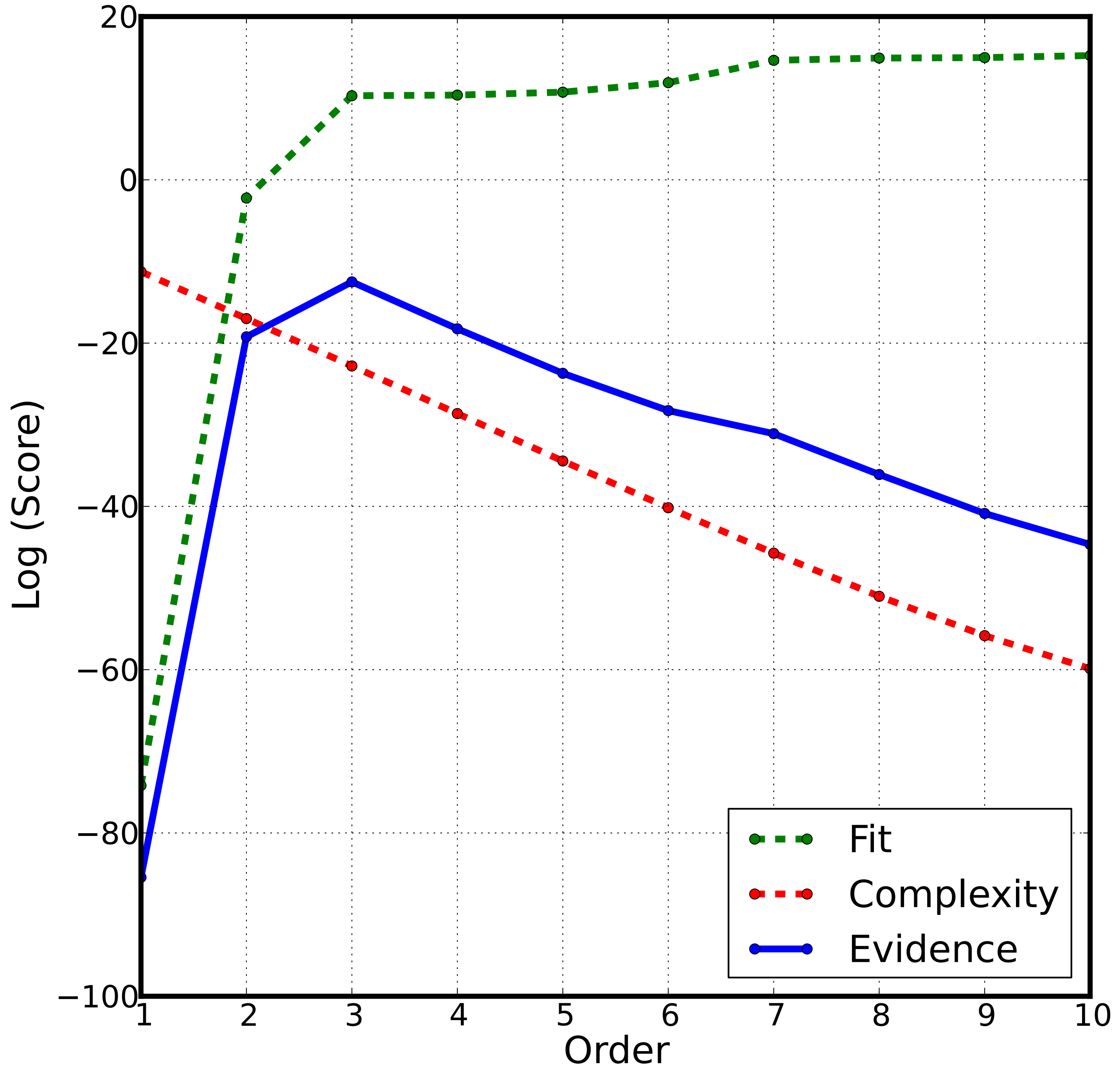


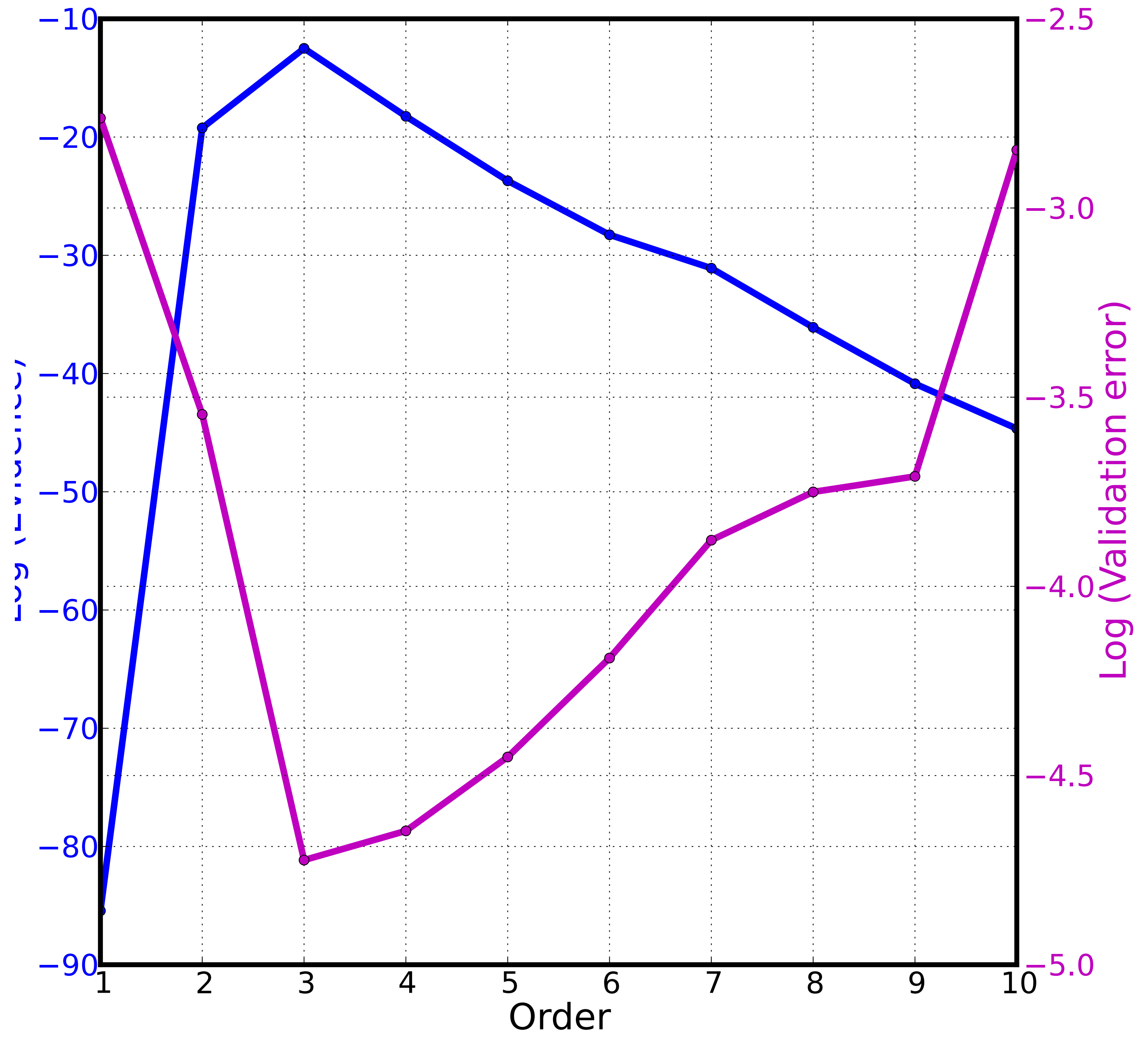
K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: Model Evidence = Fit + Complexity

$$\log p(y) = \int \log p(y) p(\lambda|y) d\lambda = \int \log \left[\frac{p(y|\lambda)p(\lambda)}{p(\lambda|y)}\right] p(\lambda|y) d\lambda = \underbrace{\int \log p(y|\lambda) p(\lambda|y) d\lambda}_{Fit} \underbrace{- \int \log \left[\frac{p(\lambda|y)}{p(\lambda)}\right] p(\lambda|y) d\lambda}_{Complexity}$$

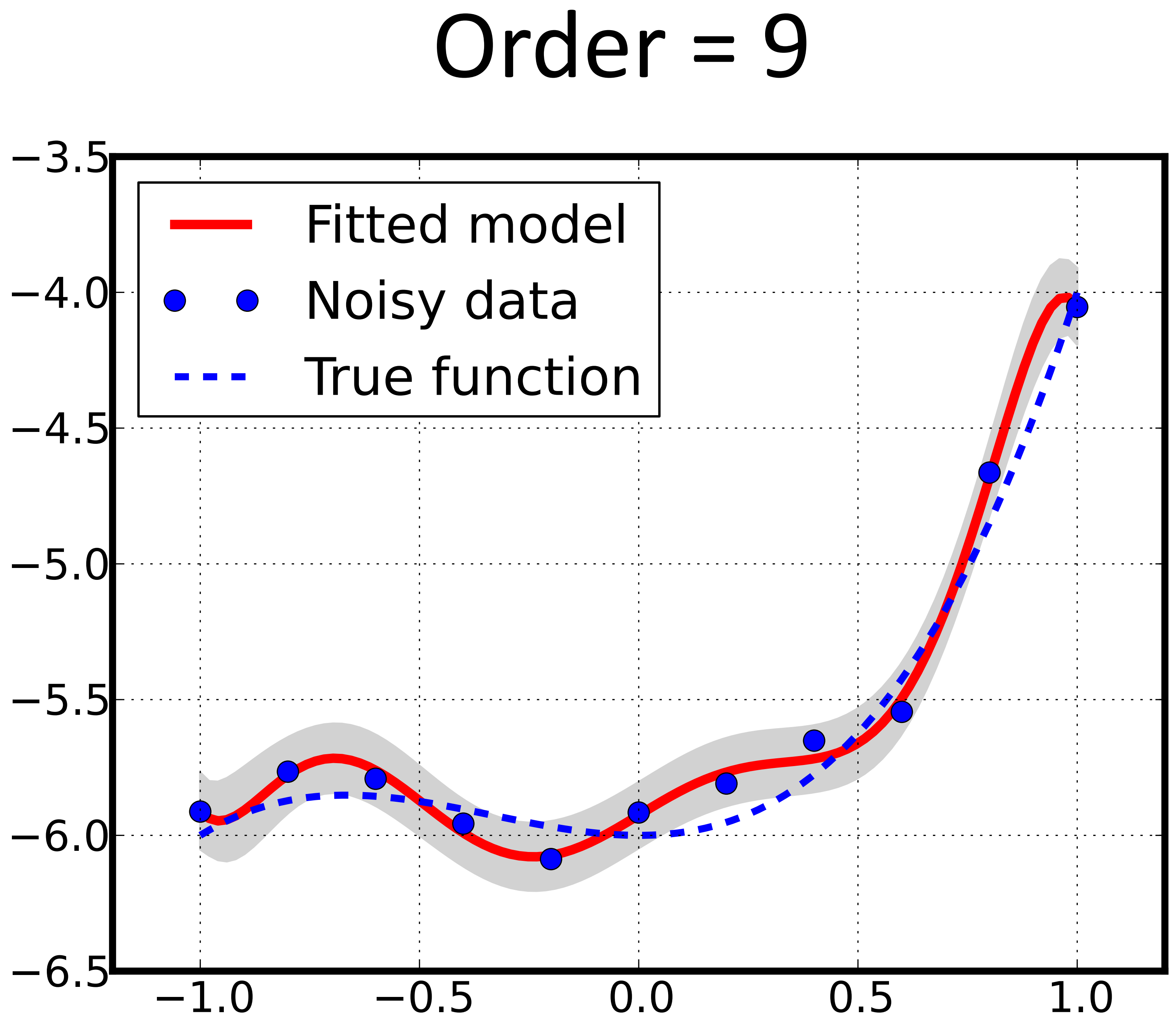


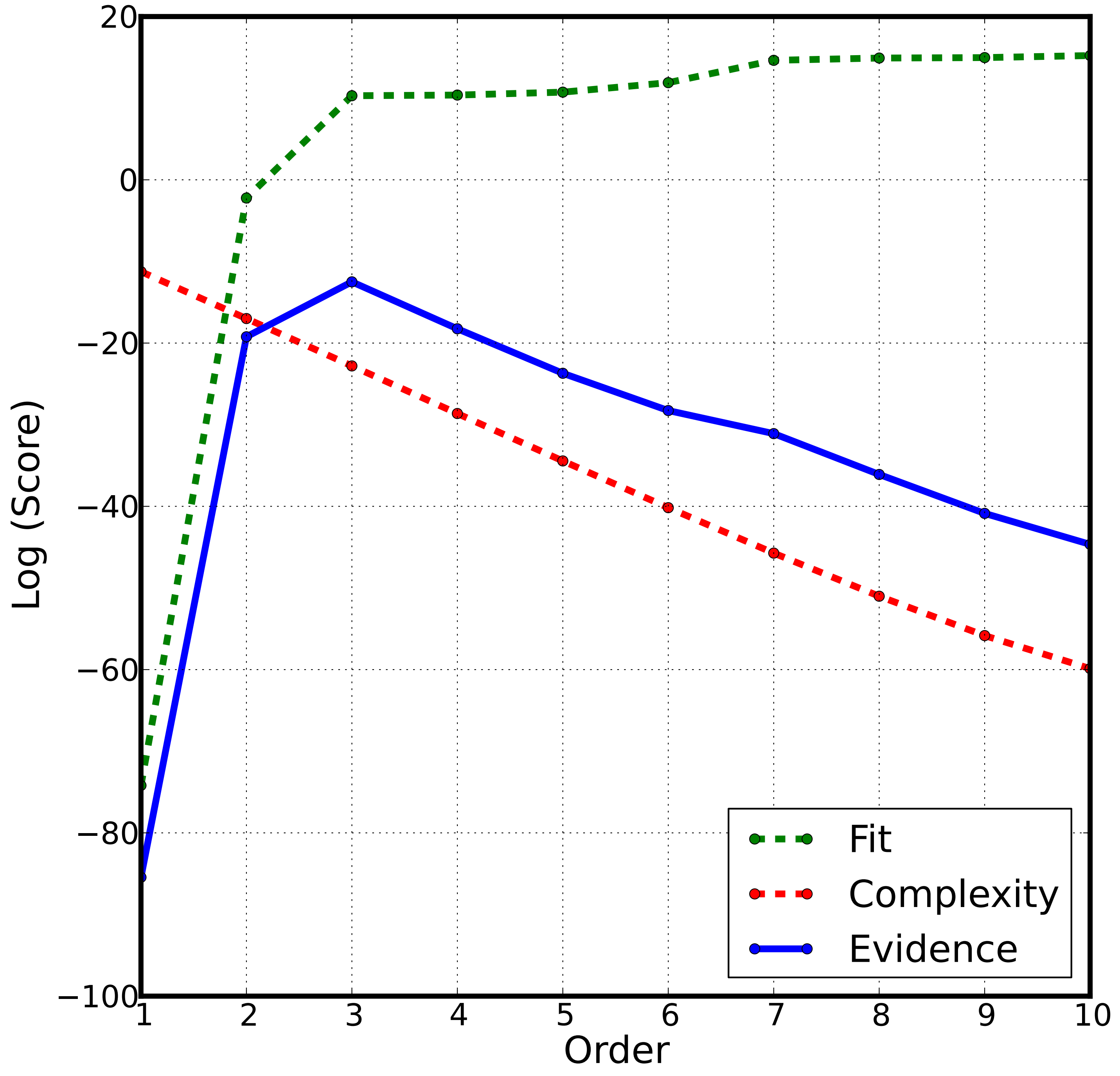



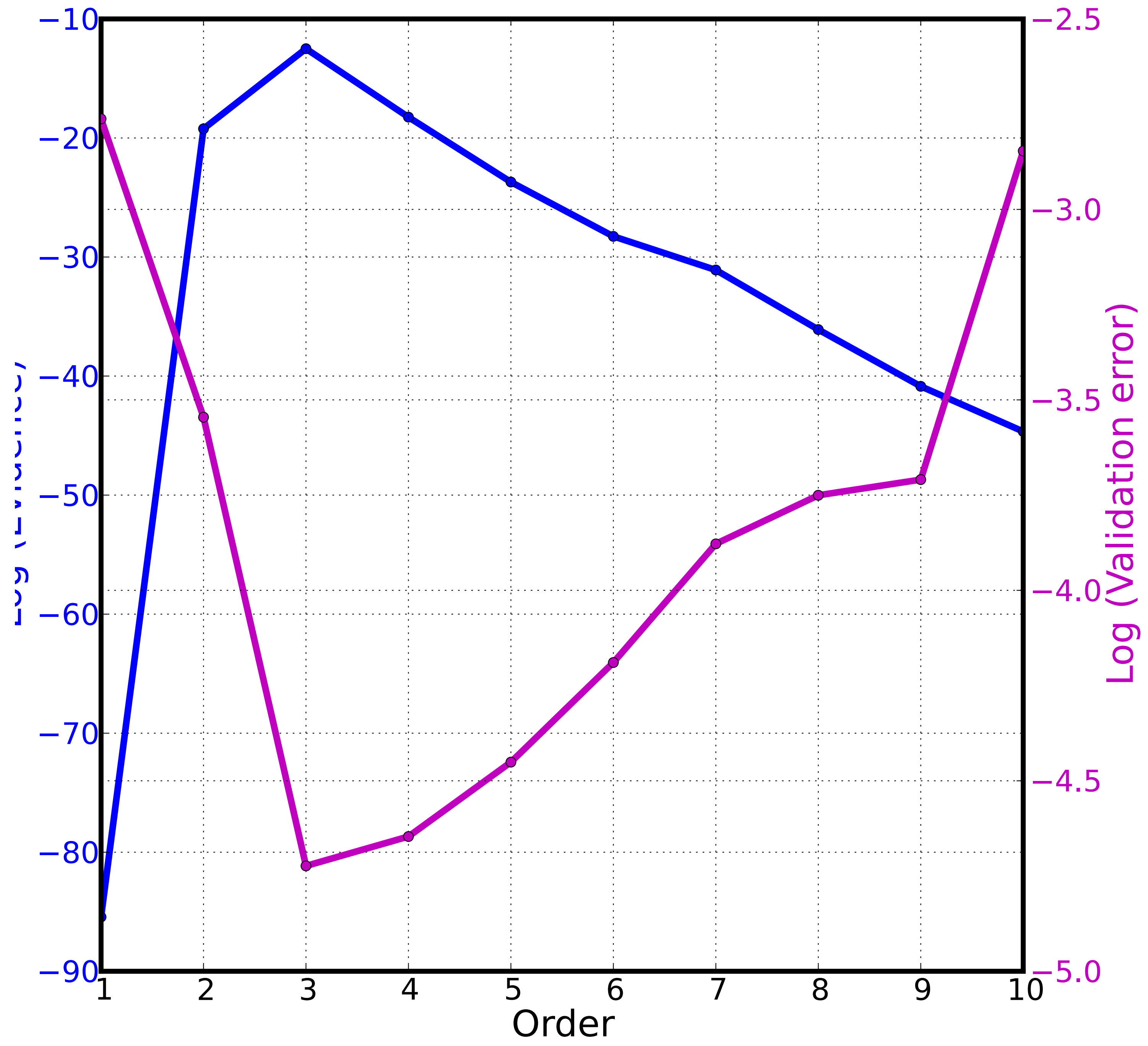



# Inv UQ: Model Evidence = Fit + Complexity

$$\log p(y) = \int \log p(y) p(\lambda|y) d\lambda = \int \log \left[ \frac{p(y|\lambda)p(\lambda)}{p(\lambda|y)} \right] p(\lambda|y) d\lambda = \underbrace{\int \log p(y|\lambda) p(\lambda|y) d\lambda}_{Fit} \underbrace{- \int \log \left[ \frac{p(\lambda|y)}{p(\lambda)} \right] p(\lambda|y) d\lambda}_{Complexity}$$





K. Sargsyan (ksargsy@sandia.gov)

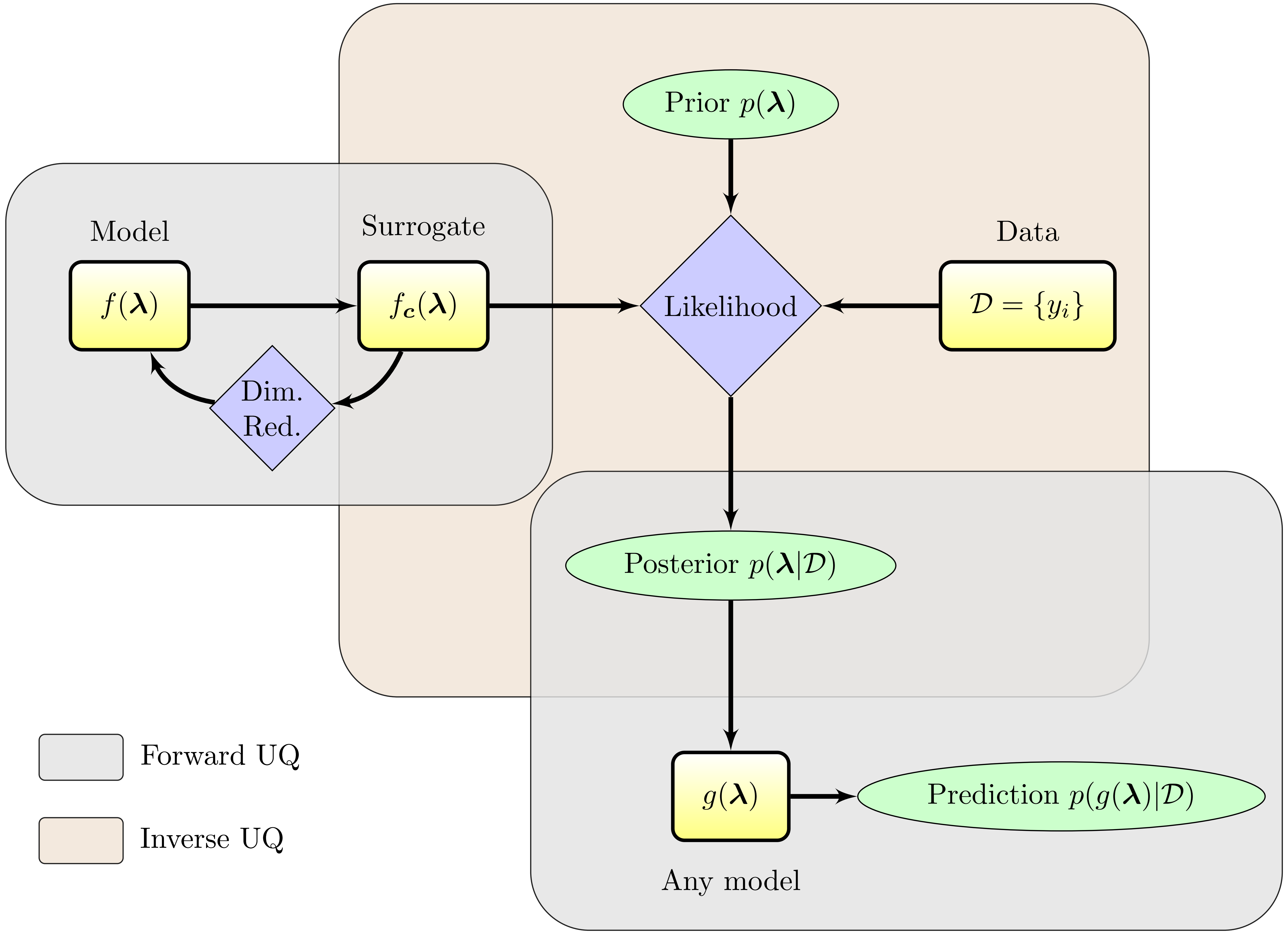
# Inv UQ: Challenges

- Model $f(\lambda)$ is **expensive**
  - Prebuilt and use a surrogate (forward UQ exercise)
  - Not too relevant for NNs

- Model $f(\lambda)$ is assumed perfect
  - Can not ignore **model structural error**
  - Incorporate and correct for model deficiencies

- Model input is **high-dimensional**: $f(\lambda) = f(\lambda_1, \ldots, \lambda_d)$ for $d \gg 1$
  - MCMC has hard time traveling the high-d posterior surface
  - Extremely relevant for NNs

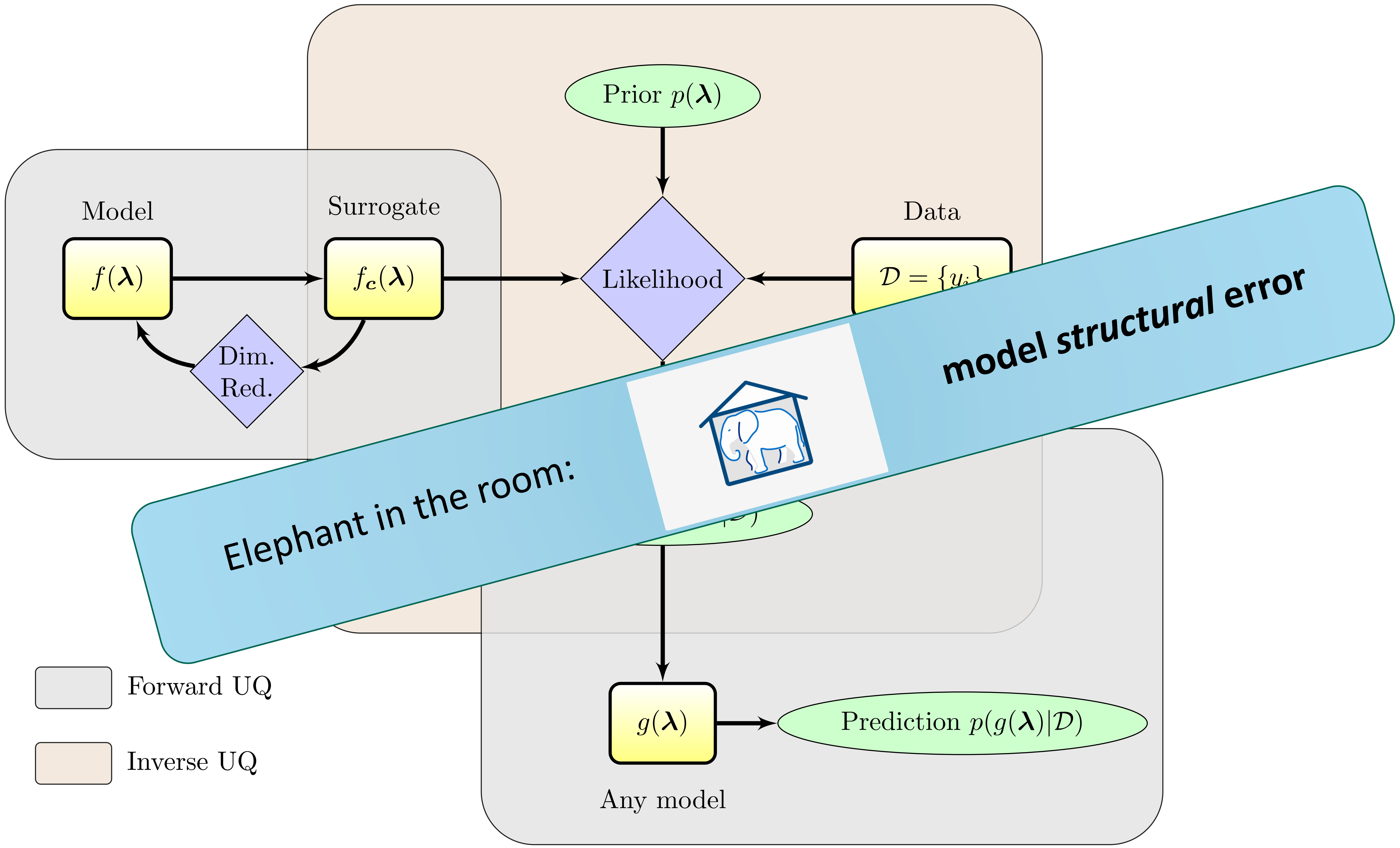
# Inv UQ: Surrogate-enabled Bayesian inference




\+ Surrogate Error

Prediction Variance = Parametric Uncertainty + Data Noise

# Inv UQ: Surrogate-enabled Bayesian inference




\+ **Model Structural Error**

\+ **Surrogate Error**

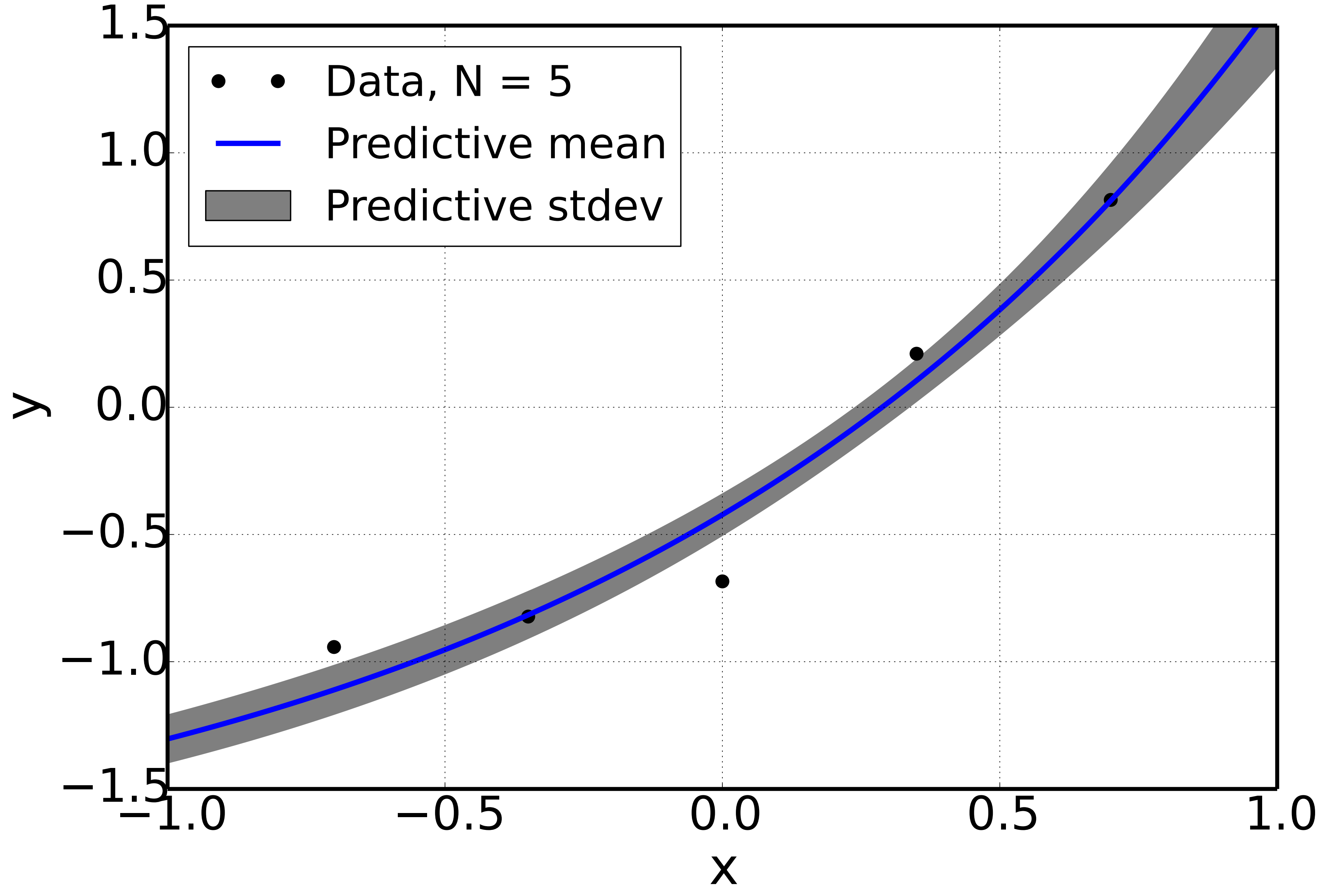
**Prediction Variance** = **Parametric Uncertainty** + **Data Noise**

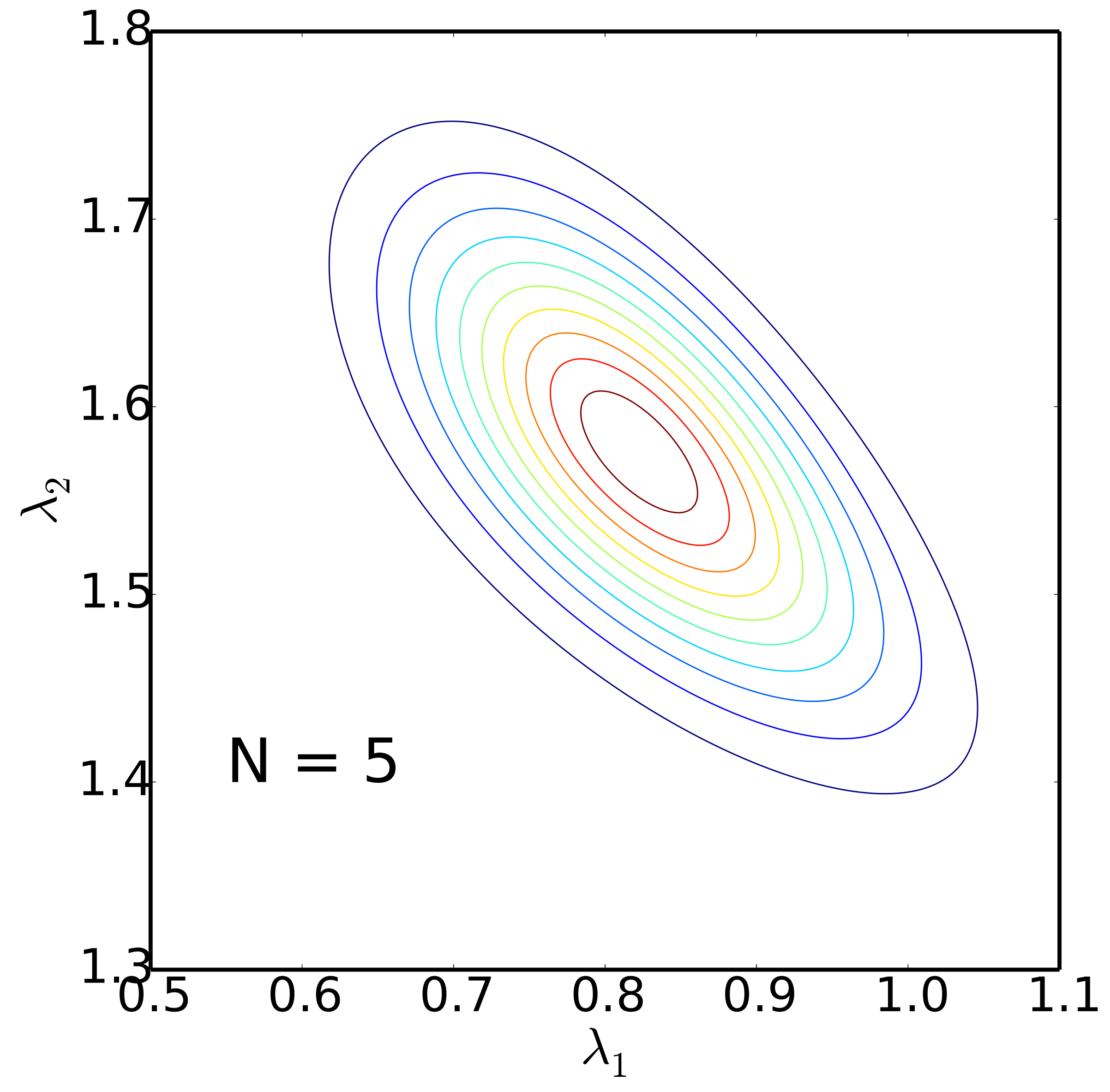
# Inv UQ: Model Error

Model error: otherwise called (with slightly altered meanings):

model discrepancy, model structural error, model inadequacy, model misspecification, model form error, model uncertainty.

**Ignoring model error leads to overconfident and biased predictions**





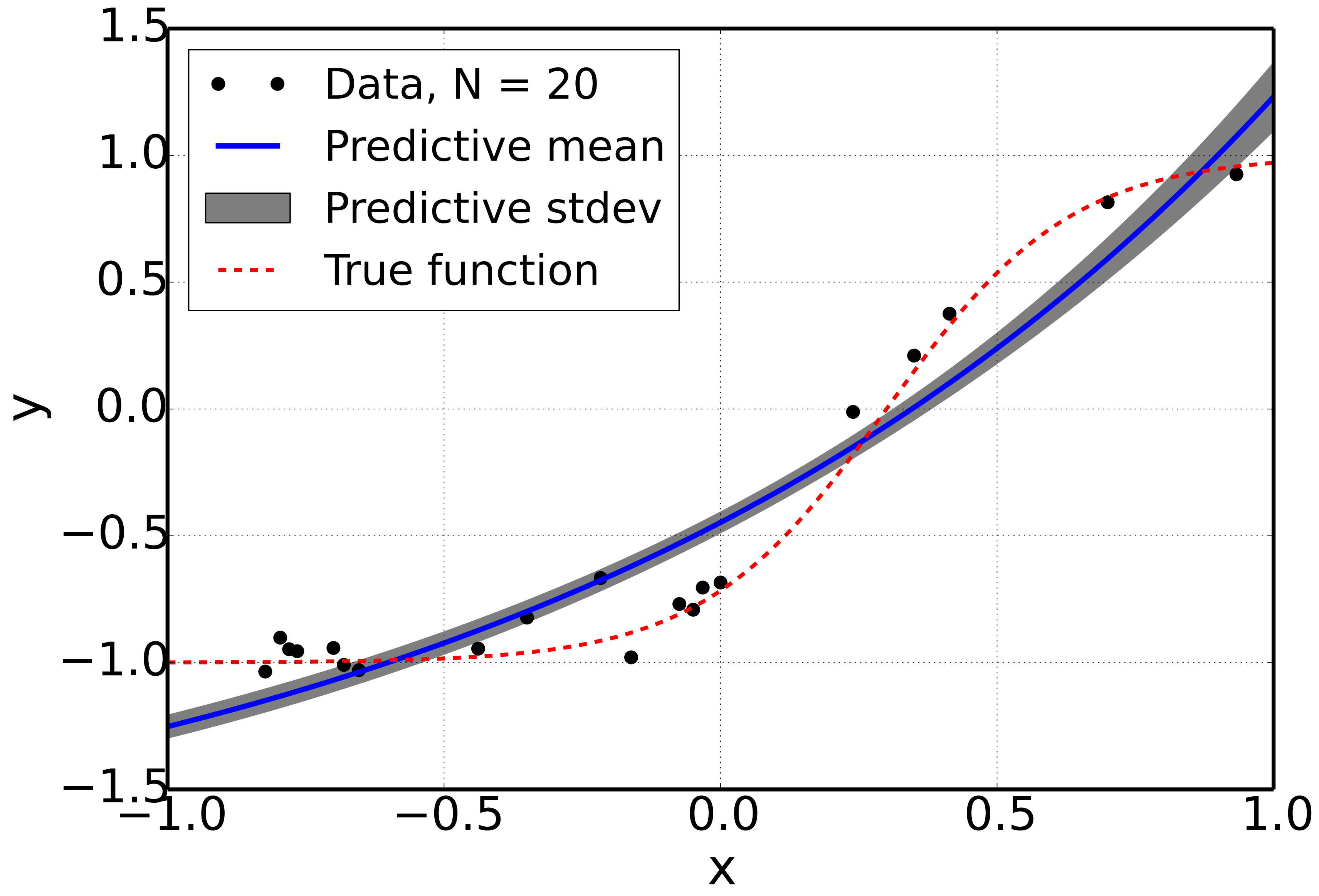

K. Sargsyan (ksargsy@sandia.gov)

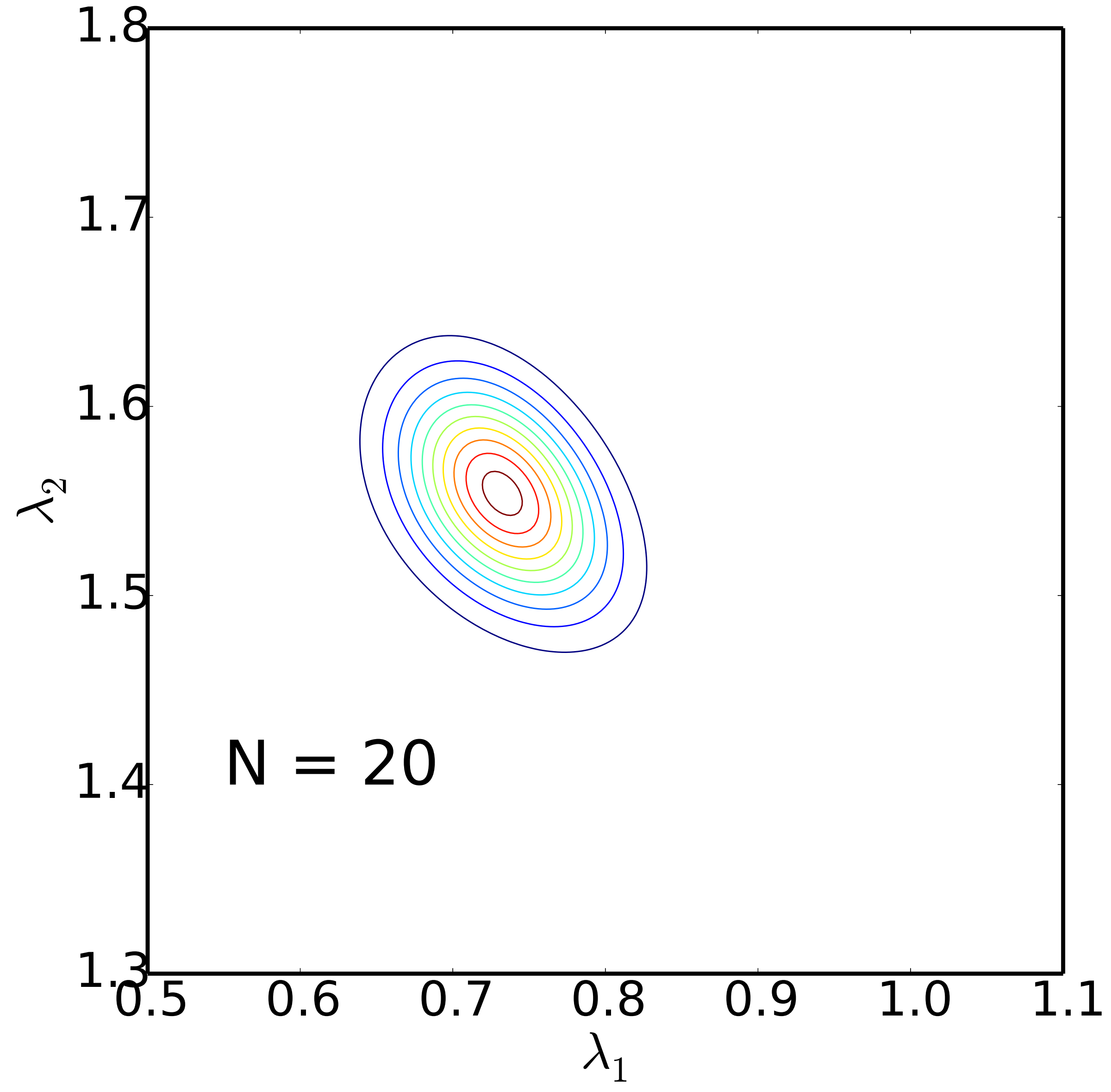
# Inv UQ: Model Error

Model error: otherwise called (with slightly altered meanings):

model discrepancy, model structural error, model inadequacy, model misspecification, model form error, model uncertainty.

**Ignoring model error leads to overconfident and biased predictions**





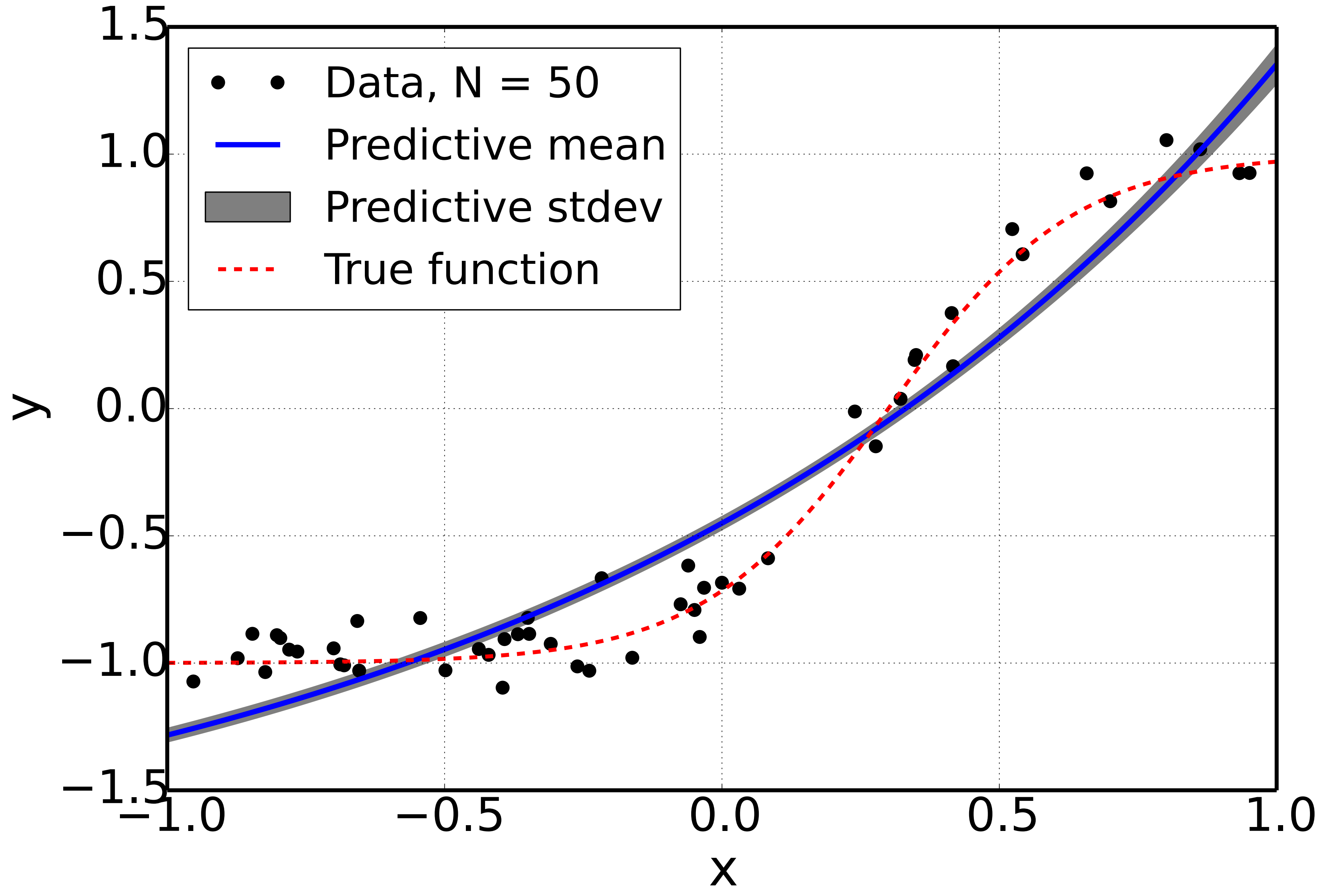
K. Sargsyan (ksargsy@sandia.gov)

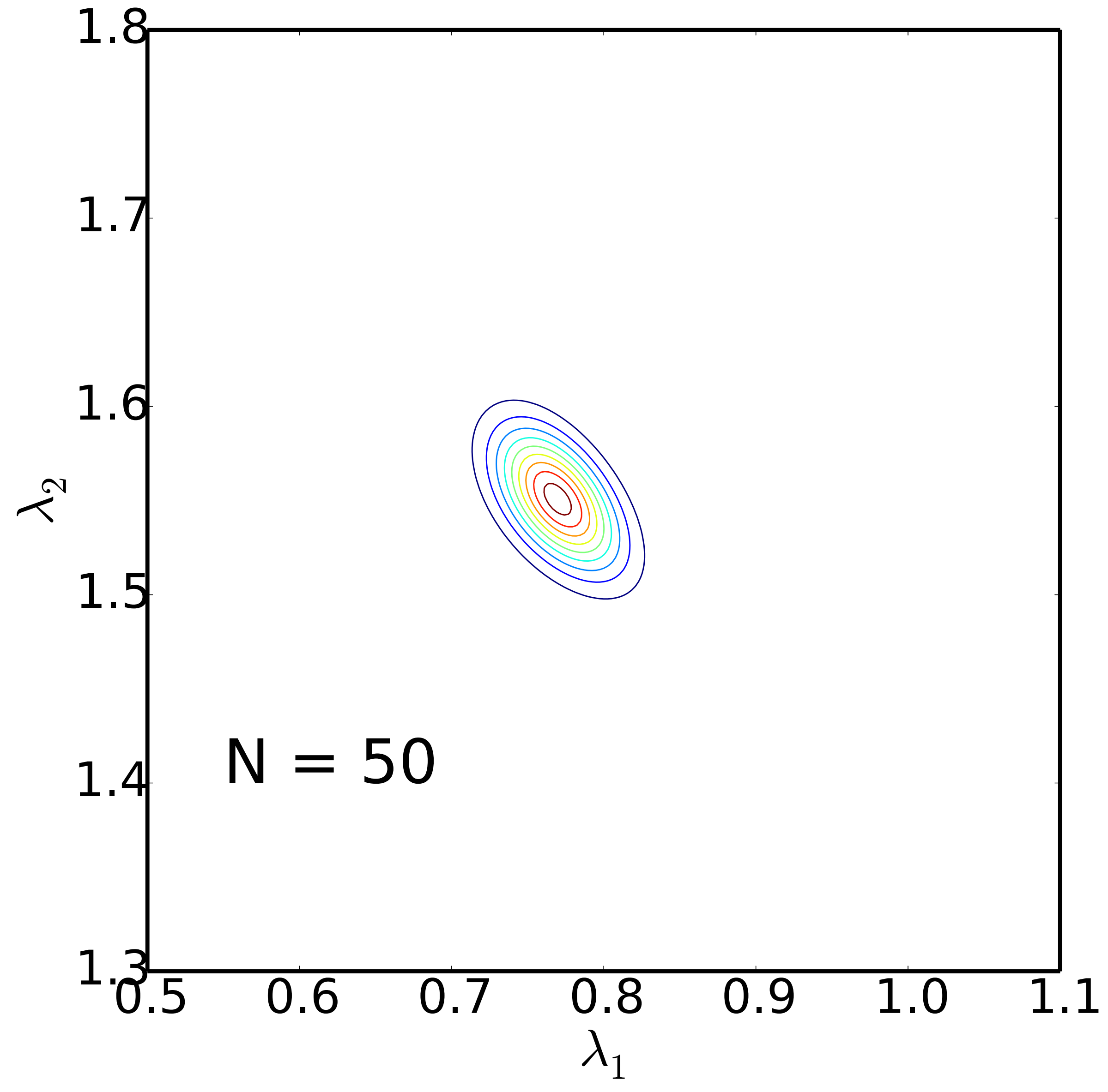
# Inv UQ: Model Error

Model error: otherwise called (with slightly altered meanings):

model discrepancy, model structural error, model inadequacy, model misspecification, model form error, model uncertainty.

**Ignoring model error leads to overconfident and biased predictions**






K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: How to Correct for Model Error

*External:* Conventional *statistical* Gaussian Process correction *[Kennedy, O'Hagan, 2001]*

$$g(x_i) = f(\lambda; x_i) + \delta_\alpha(x_i) + \epsilon_i$$

Challenges when it comes to *physical* models.

# Inv UQ: How to Correct for Model Error

*External:* Conventional *statistical* Gaussian Process correction *[Kennedy, O'Hagan, 2001]*

$$g(x_i) = f(\lambda; x_i) + \delta_\alpha(x_i) + \epsilon_i$$

Challenges when it comes to *physical* models.

---

*Embedded Intrusive:* Model-specific corrections: changing the model/code

$$g(x_i) = \tilde{f}(\lambda; x_i; \delta_\alpha(x_i)) + \epsilon_i$$

# Inv UQ: How to Correct for Model Error

*External:* Conventional *statistical* Gaussian Process correction *[Kennedy, O'Hagan, 2001]*

$$g(x_i) = f(\lambda; x_i) + \delta_\alpha(x_i) + \epsilon_i$$

Challenges when it comes to *physical* models.

---

*Embedded Intrusive:* Model-specific corrections: changing the model/code

$$g(x_i) = \tilde{f}(\lambda; x_i; \delta_\alpha(x_i)) + \epsilon_i$$

---

*Embedded Non-Intrusive:* Model-agnostic: cast deterministic model parameter as a random variable *[Sargsyan, 2019]*

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

# 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 $\lambda$ and model-error parameters $\alpha$ 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)

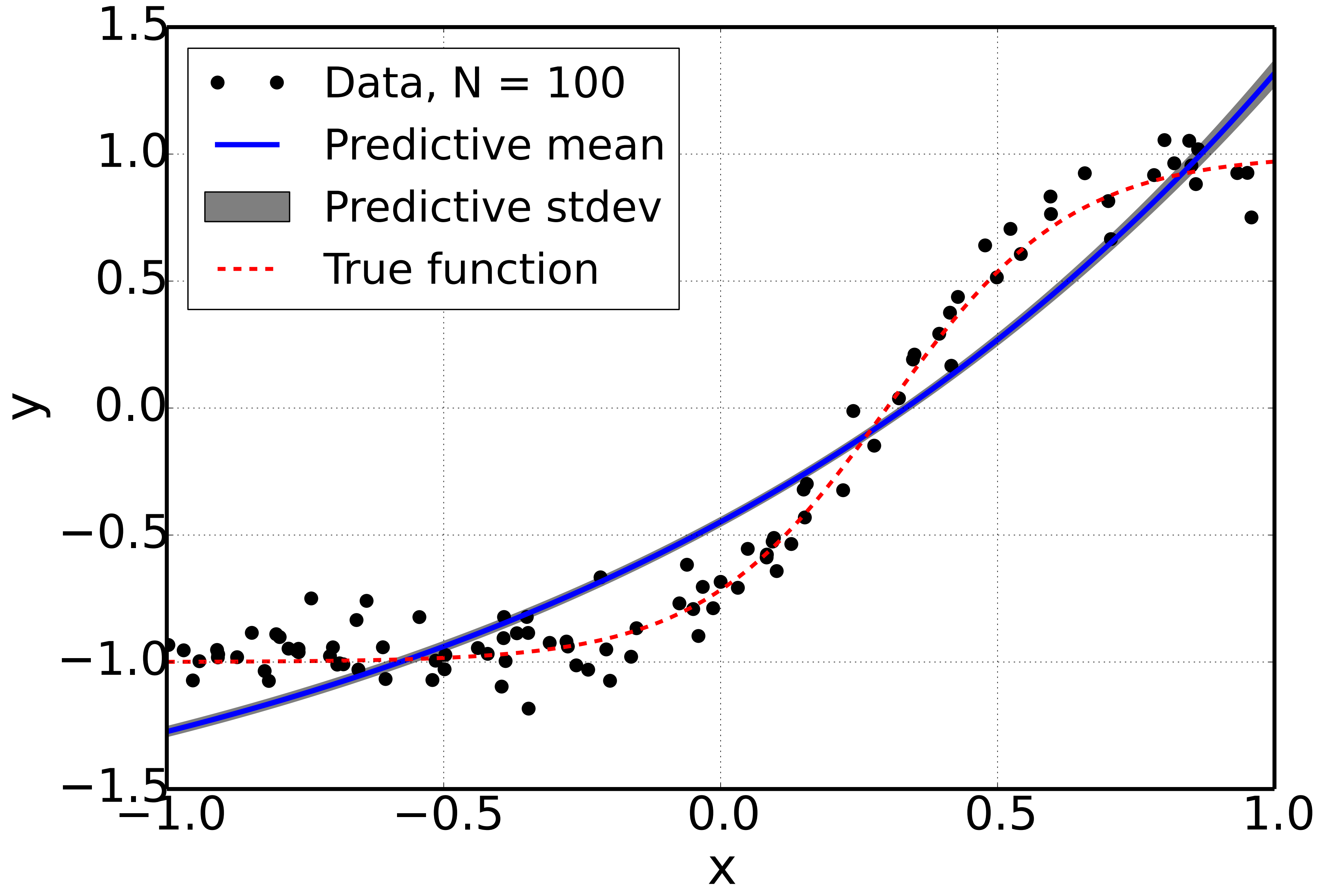

K. Sargsyan (ksargsy@sandia.gov)

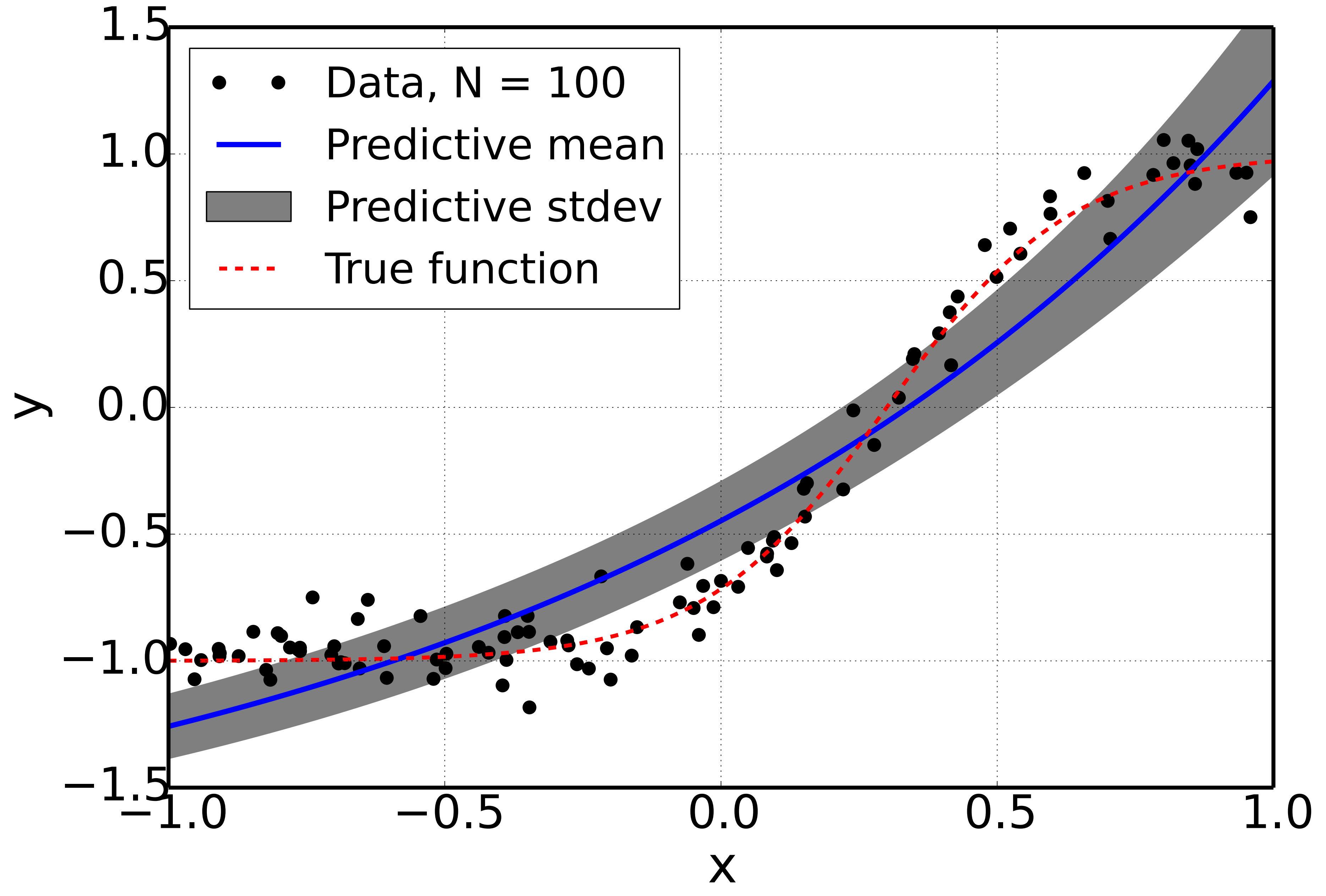
# Inv UQ: Embedded Model Error

**Without model error**

Predictive uncertainty captures model error

**With model error**

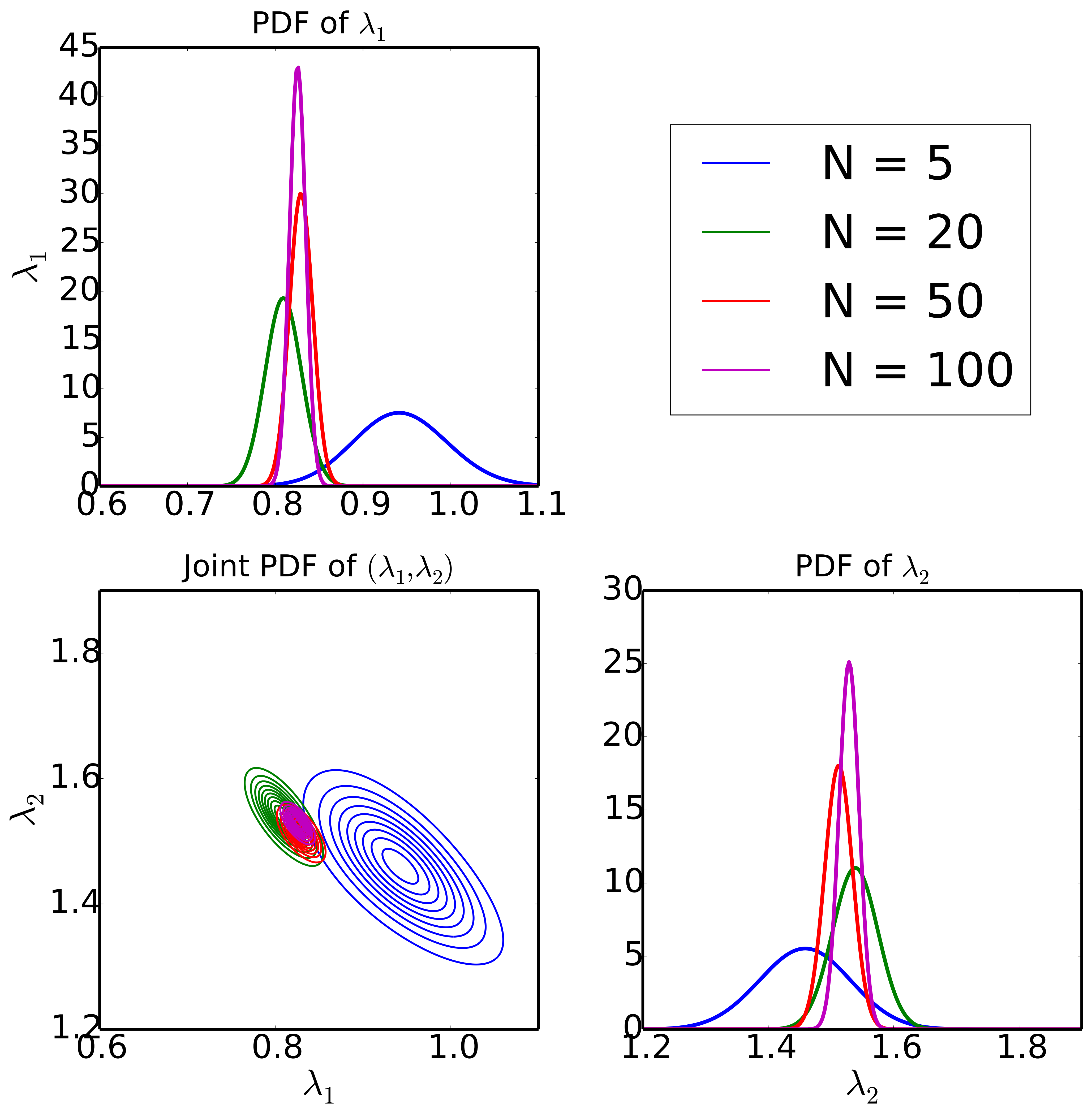


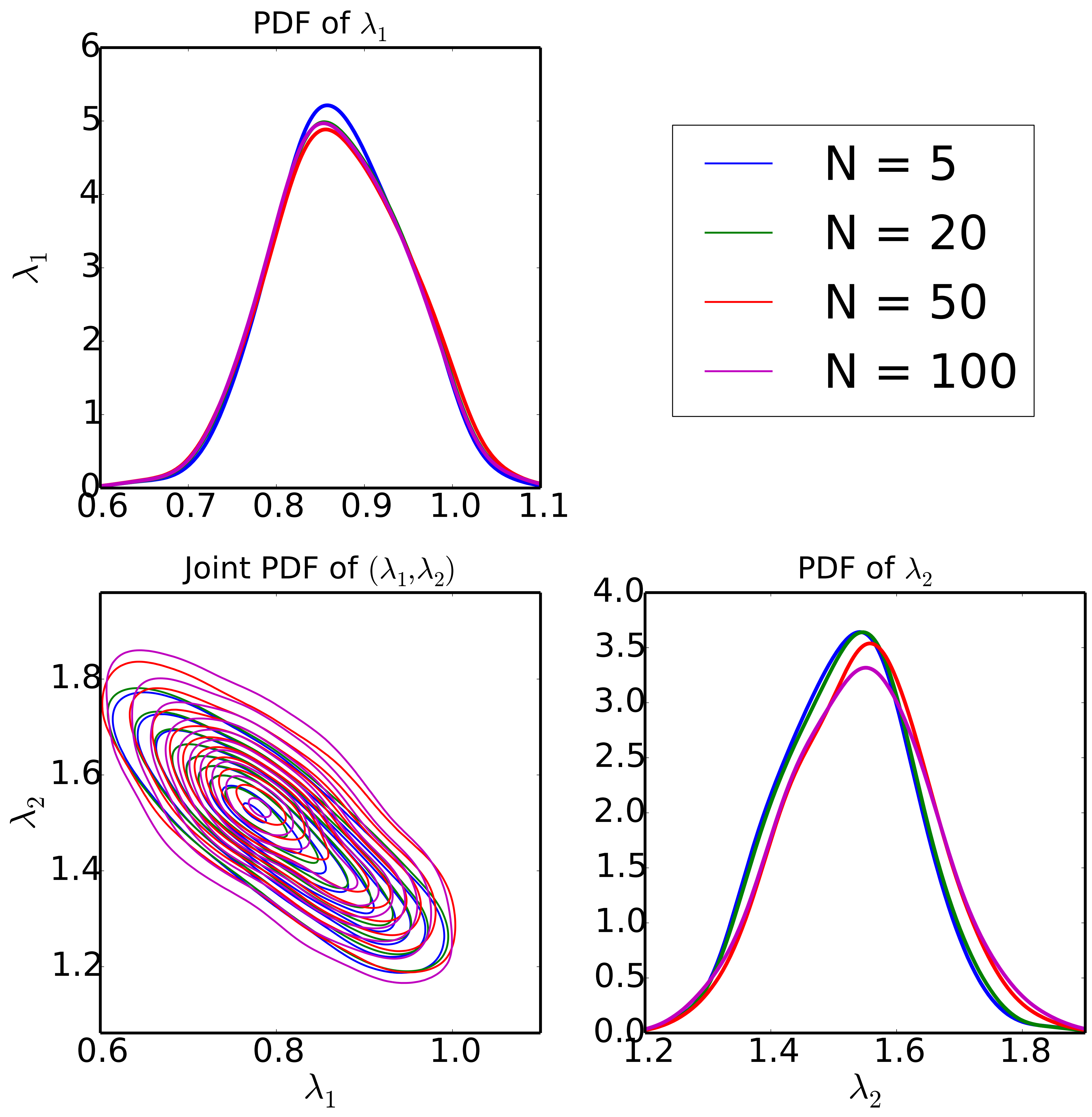
# Inv UQ: Embedded Model Error

Stable prediction of “physical” parameters

**Without model error**



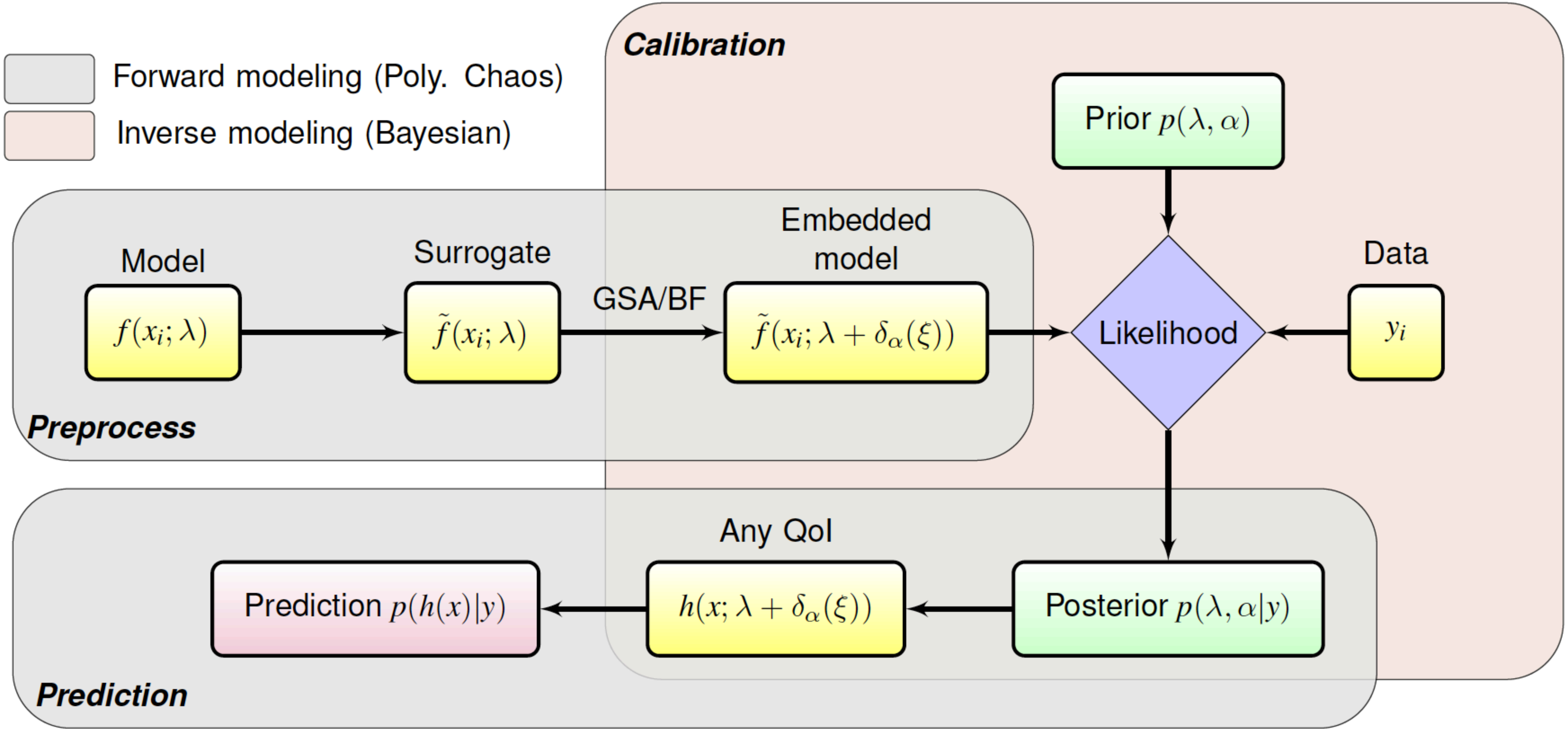

**With model error**





K. Sargsyan (ksargsy@sandia.gov)

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

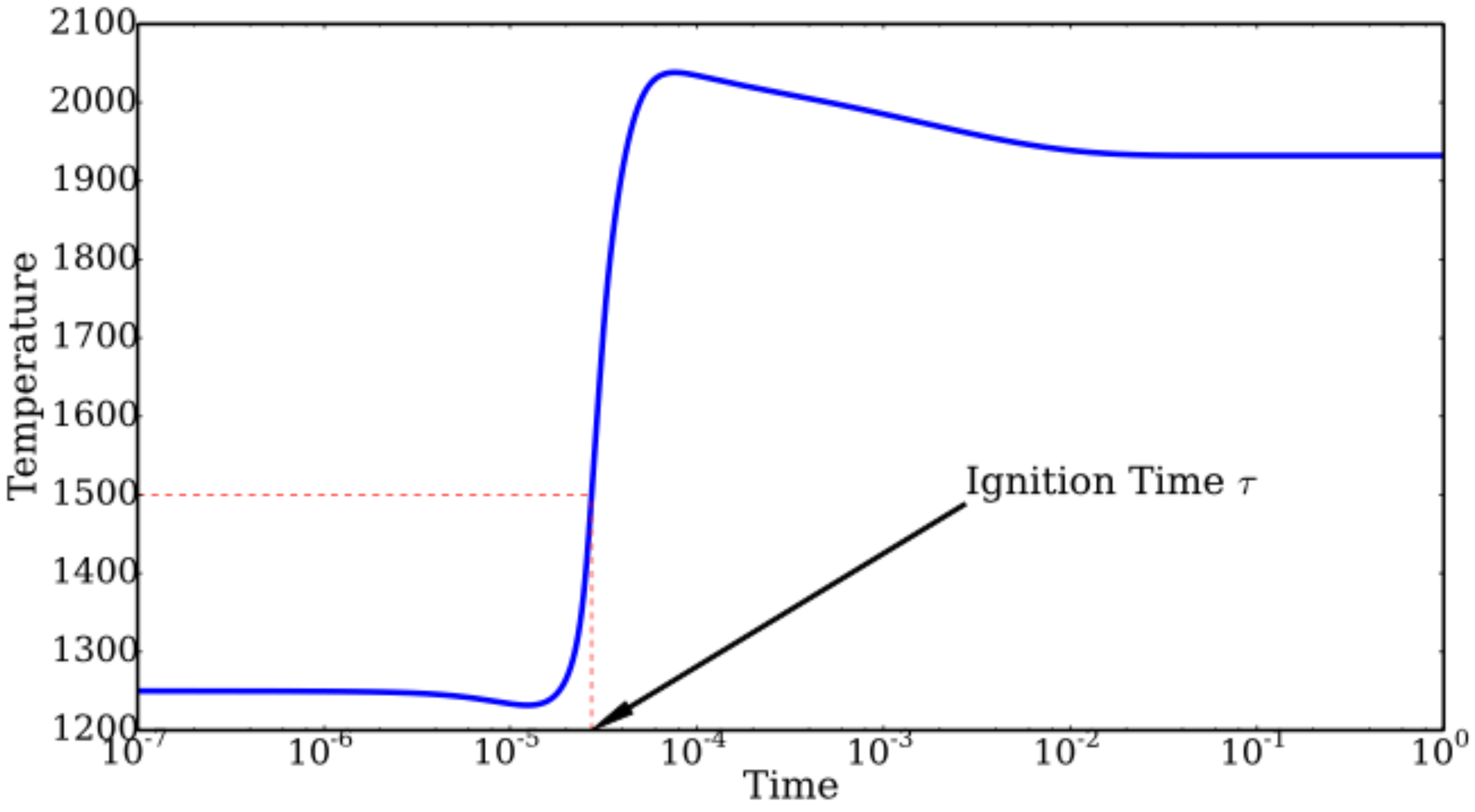


# Application: Ignition time in chemical kinetics

- Two-step global reaction model calibrated against shock tube experimental data
- Operating conditions: pressure $P$, initial temperature $T_0$ and equiv. ratio $\phi$.

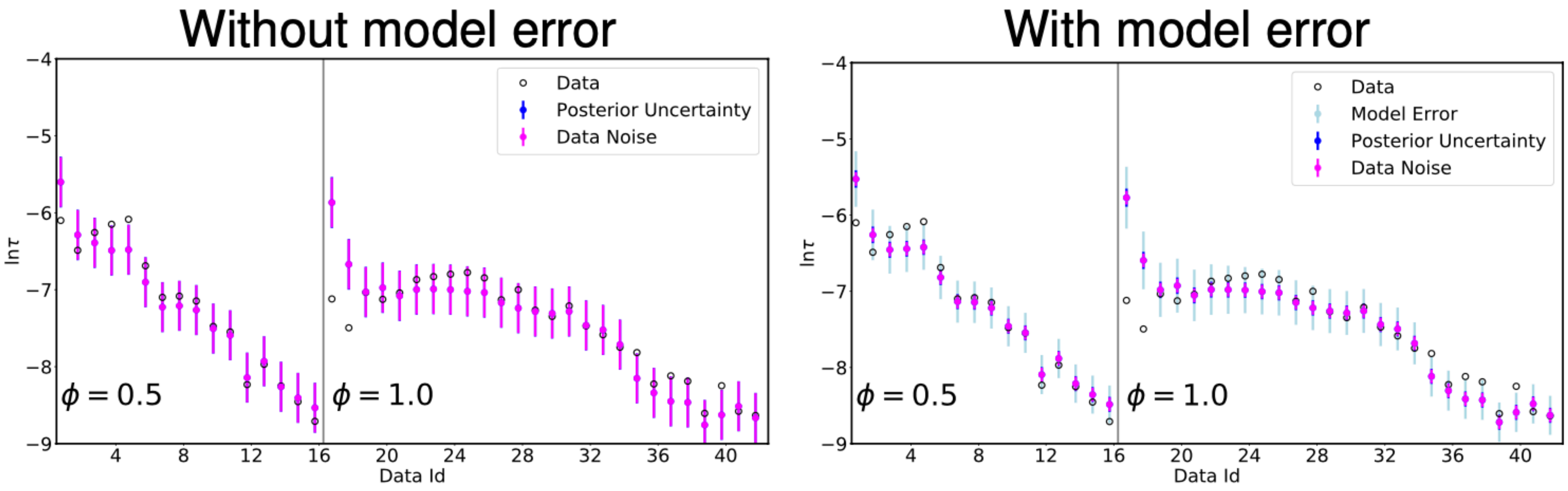
$$
\begin{aligned}
C_{12}H_{26} + \tfrac{25}{2}O_2 &\xrightarrow{k_1} 12CO + 13H_2O \\
CO + \tfrac{1}{2}O_2 &\underset{k_{2b}}{\overset{k_{2f}}{\rightleftharpoons}} CO_2.
\end{aligned}
$$

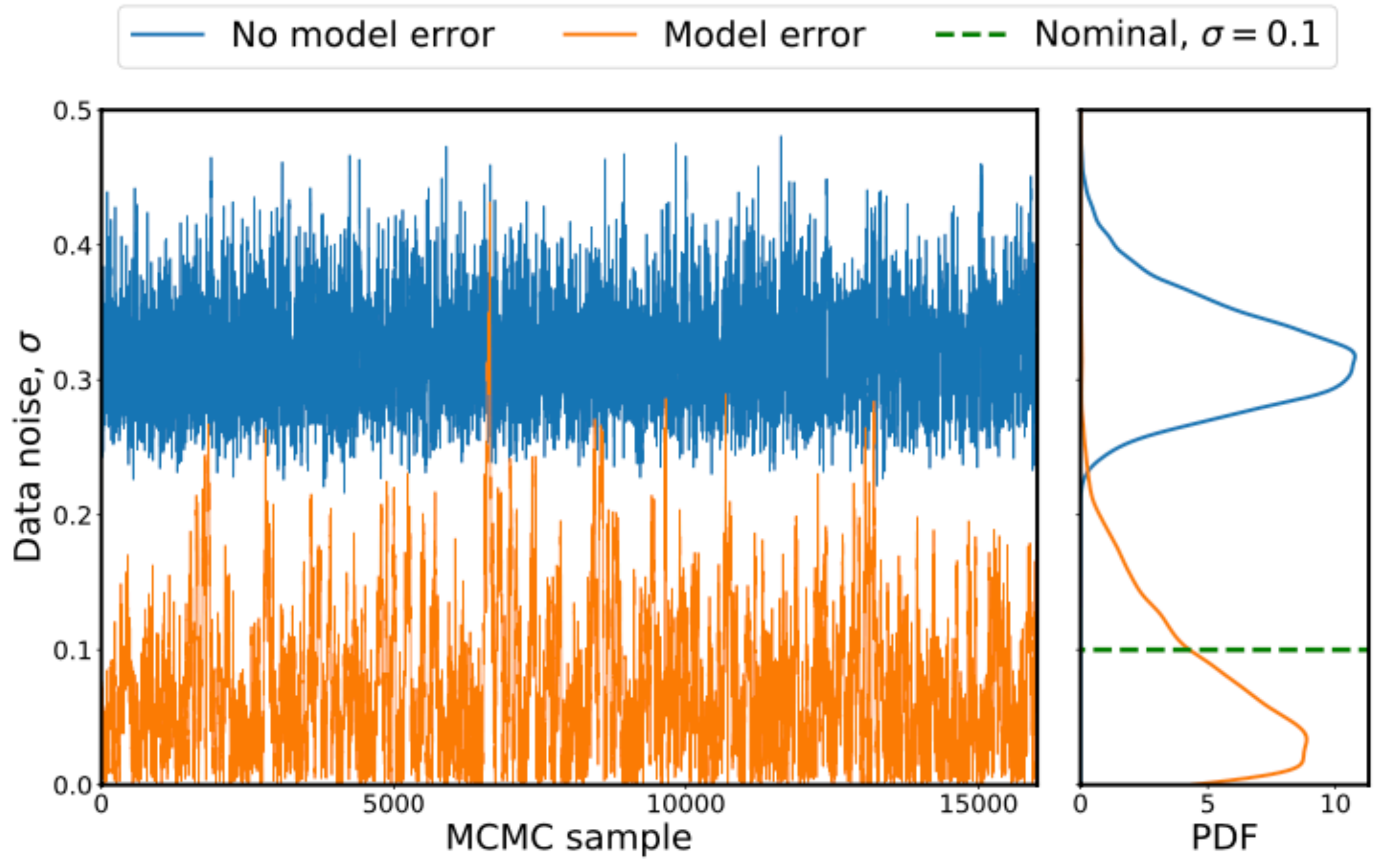
$$k_1 = Ae^{\left(-\frac{E}{RT}\right)}[C_{12}H_{26}]^{0.25}[O_2]^{1.25}$$




- QoI: log(ignition time)
- Embedding in $(\ln A, E) = \sum_k \alpha_k \Psi_k(\xi)$

# Application: Ignition time in chemical kinetics



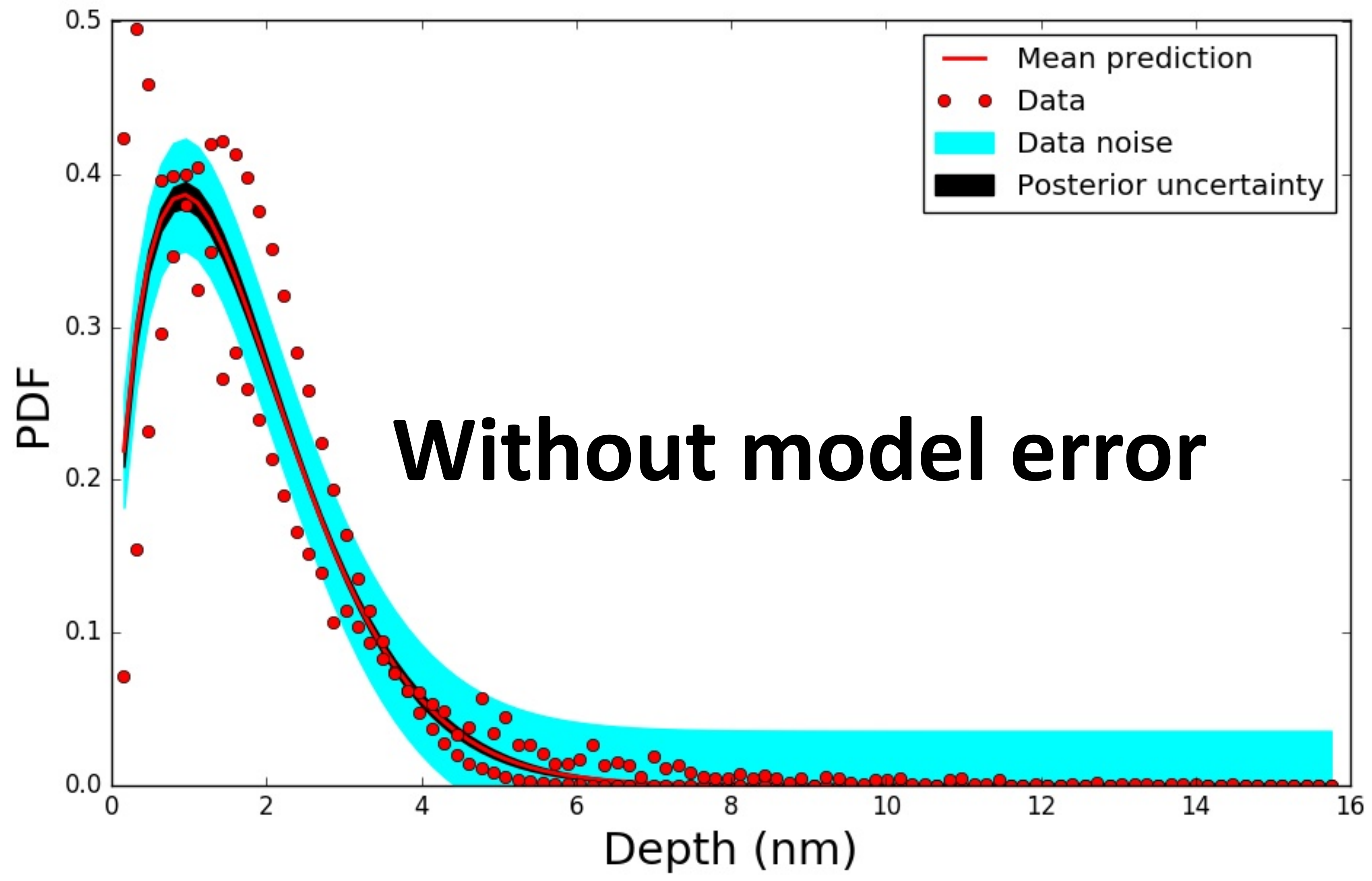

- Model error disambiguated from data error
- Data error correctly captured
- Meaningful extrapolative predictions

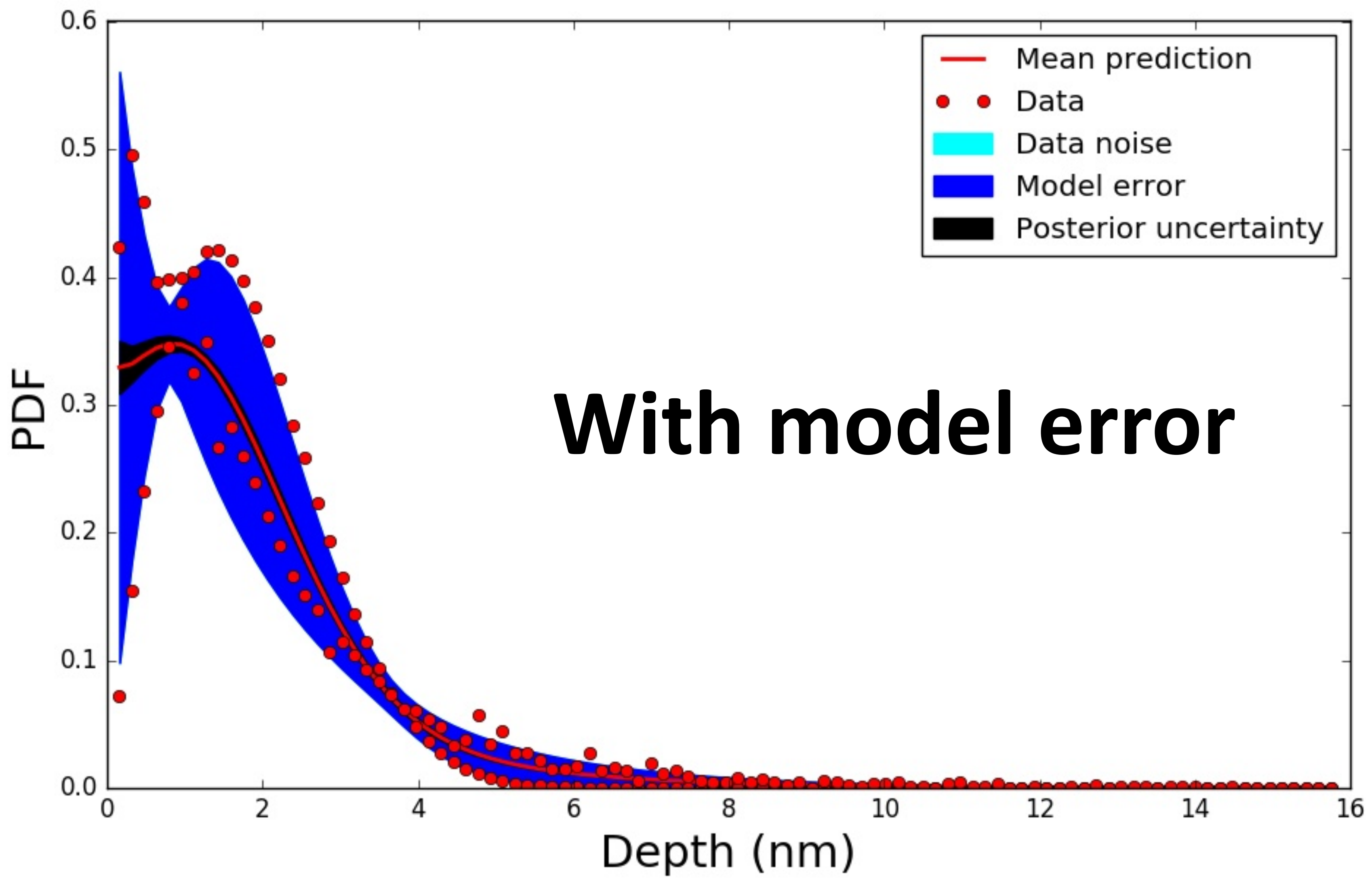
[K. Sargsyan, X. Huan, H. Najm, "Embedded Model Error Representation for Bayesian Model Calibration", IJUQ, 2019]


K. Sargsyan (ksargsy@sandia.gov)

# Fusion modeling

- Cluster-dynamics code, Xolotl
- Simulate surface response of a tungsten plasma-facing component as a function of incident Helium flux
- Constructing uncertain input profiles for tungsten depth to propagate through Xolotl (PSI code)
- Data from two sources - ‘model error’ captures uncertainty due to data heterogeneity



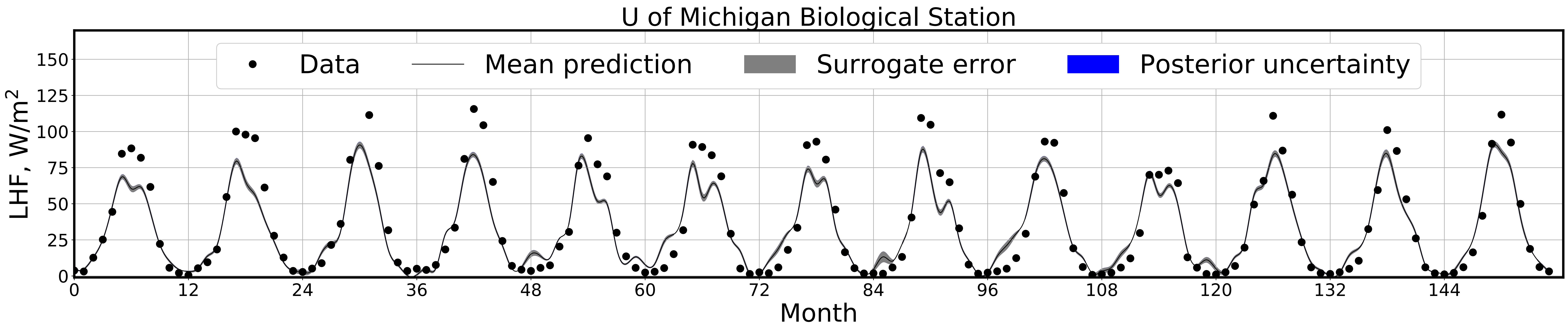

Ignoring model error wrongly attributes uncertainty to data

[O. Cekmer, K. Sargsyan, S. Blondel, H. Najm, D. Bernholdt,, B.D. Wirth, “Uncertainty quantification for incident helium flux in plasma-exposed tungsten”, IJUQ, 2018]


K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model





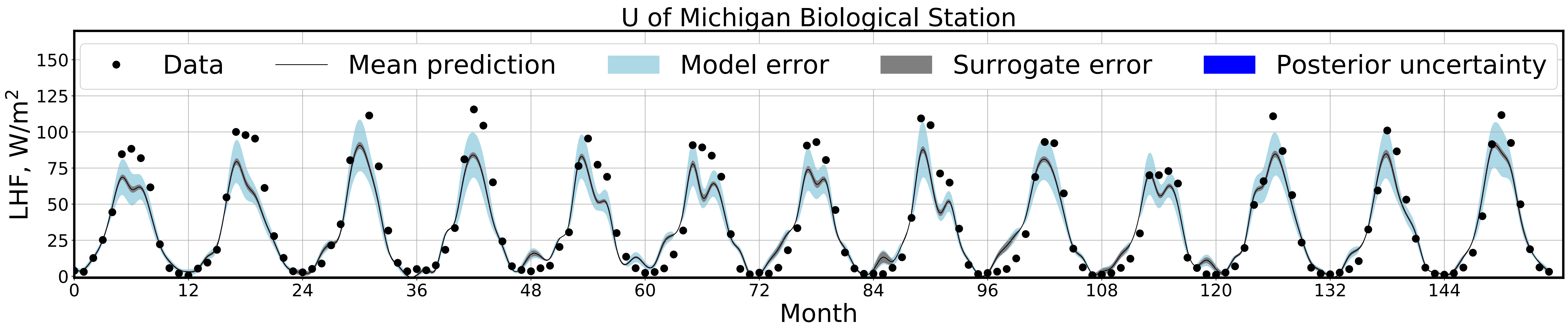

Conventional calibration without model error

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



K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model



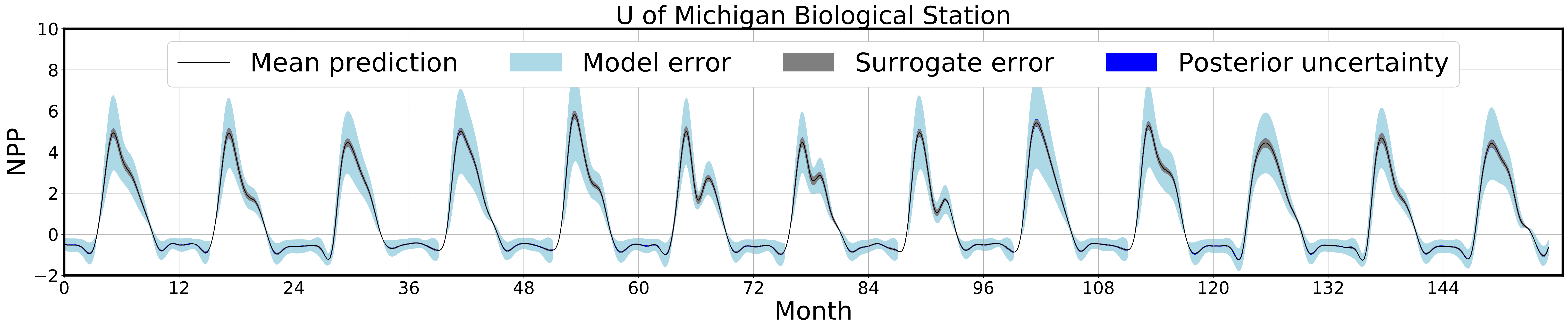

Predictive uncertainty attributed to model error

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



K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model




Allows meaningful prediction of other QoIs (e.g. NPP with no data)

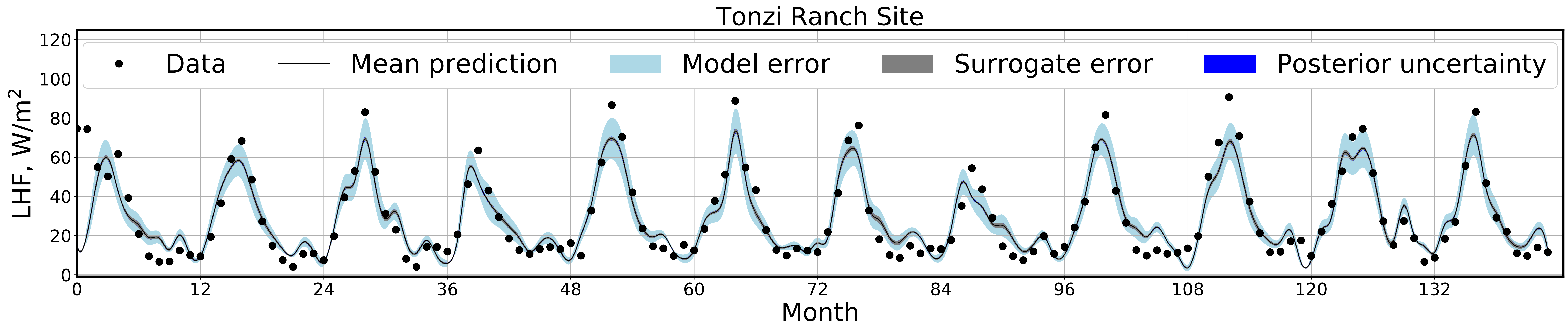
- LHF = Latent Heat Flux
- NPP = Net Primary Productivity



K. Sargsyan (ksargsy@sandia.gov)

# E3SM Land Model






Allows (dangerous!) extrapolation to other sites

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



K. Sargsyan (ksargsy@sandia.gov)

| Embedded Model Error | Variational Inference |
|---|---|
| Casting physical parameters $\lambda$ 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 $\alpha$ Bayesian/MCMC | Infer $\alpha$ 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)

# Inv UQ: High-dimensionality (scalability) challenge

✓ Hamiltonian MCMC
✓ Transitional MCMC


K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: High-dimensionality (scalability) challenge

- Starting point of MCMC becomes important. Multiple chains. Tempering to explore/exploit.
- Multimodality, ridges (low-d manifolds with similar posterior values) in posterior shape.
- Important to use good proposal distributions.
- Infer only the most sensitive inputs. Forward UQ / GSA as a preprocessing step.

- MCMC Flavors
  - ✓ Langevin MCMC
  - ✓ Hamiltonian MCMC
  - ✓ Adaptive MCMC
  - ✓ Transitional MCMC
  - ✓ Likelihood-informed subspace / Dimension-Independent MCMC, see [Cui, 2016].

- Variational approximations, see [Blei, 2017].
- Transport maps to directly map prior to posterior, see [Parnot, 2018].
- Amortized inference (pre-build a map from data to posterior), see [Ganguly, 2023].
- Approximate Bayesian computation, see [Sunnåker, 2013].

# Inv UQ: General Recap

- Parameter estimation / model calibration / inverse modeling
- Bayesian inference is a major tool
- Do not confuse with term ‘inference’ in ML!


K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: General Recap

- Parameter estimation / model calibration / inverse modeling
- Bayesian inference is a major tool
- Do not confuse with term ‘inference’ in ML!

Questions to consider:

- How expensive is the model (can we afford MCMC or pre-build surrogate)?
- How trustworthy is the model (model structural error)?
- What is the data measurement noise model (build the likelihood)?
- How many parameters to tune (advanced MCMC methods to operate in low-d)?

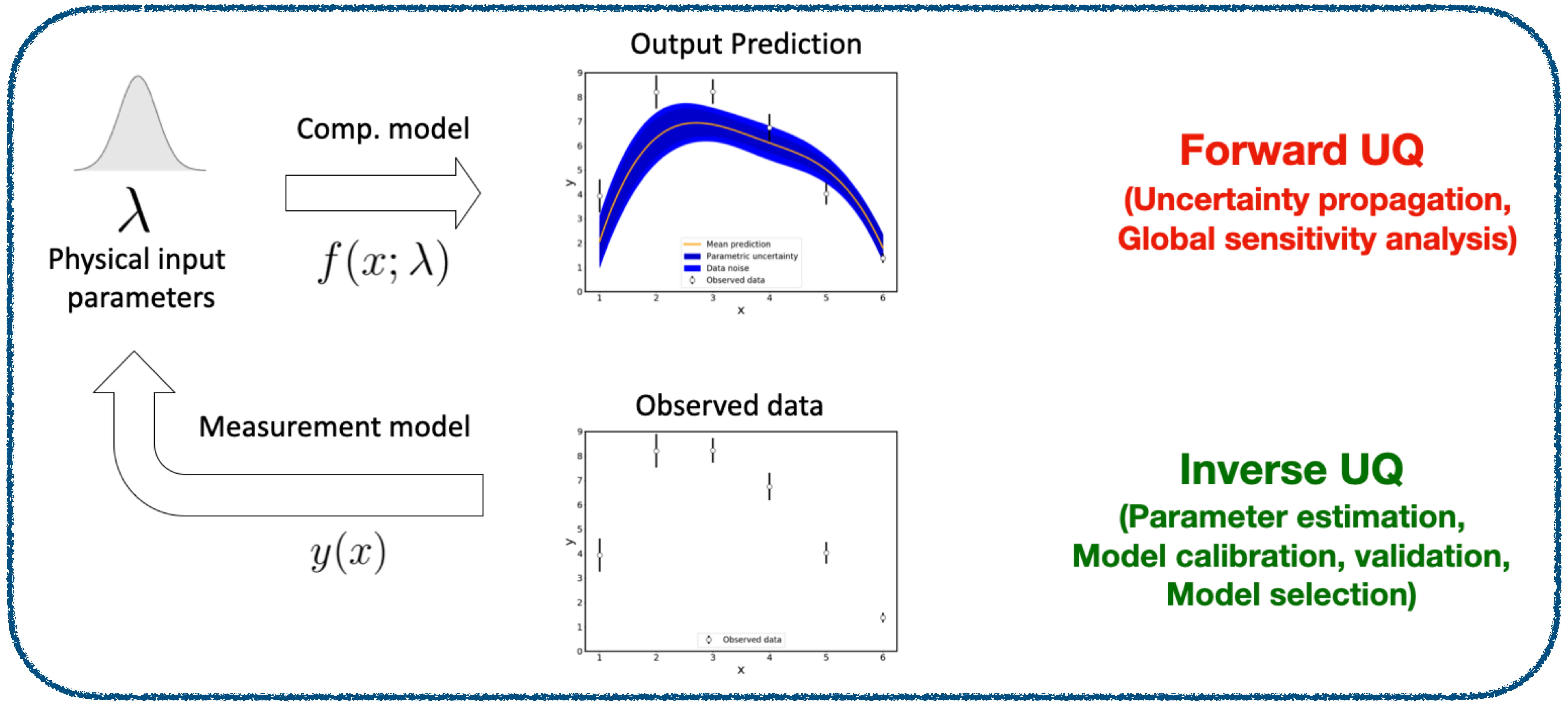
# UQ in Comp. Science: Summary

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

Probabilistic framework for
uncertainty quantification and attribution for
black-box computational models of physical phenomena

*Main tool for forward UQ:*
Surrogates; PC; GSA

*Main tool for inverse UQ:*
Bayesian inference; MCMC



# UQ Software: incomplete list

- DAKOTA: UQ, Optimization and more. Targeted for High Performance Systems
  https://dakota.sandia.gov

- UQTk: Relatively small C/C++ library for a range of UQ tasks
  https://www.sandia.gov/uqtoolkit

- PyApprox https://github.com/sandialabs/pyapprox
- MUQ https://mituq.bitbucket.io/
- OpenTURNS https://openturns.github.io/www/
- UQPy https://github.com/SURGroup/UQpy
- UQLab https://www.uqlab.com/
- ChaosPy https://github.com/jonathf/chaospy
- PSUADE https://github.com/LLNL/psuade
- ....


K. Sargsyan (ksargsy@sandia.gov)

# Literature: General

*Books:*

- R. Ghanem, P. Spanos, “Stochastic Finite Elements: A Spectral Approach”, Springer Verlag, (1991).
- A. Tarantola, "Inverse Problem Theory and Methods for Model Parameter Estimation", SIAM, (2005).
- O. Le Maître, O. Knio, “Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics”, Springer-Verlag, (2010).
- D. Xiu, “Numerical Methods for Stochastic Computations: A Spectral Method Approach”, Princeton U. Press (2010).
- A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, "Global Sensitivity Analysis. The Primer", Wiley (2005).
- P. Constantine, "Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies", SIAM (2015).

*Classic:*

- N. Wiener, “The Homogeneous Chaos”, *American Journal of Mathematics*, 60:4, 897-936 (1938).
- M. Rosenblatt, “Remarks on a Multivariate Transformation”, *Ann. Math. Statist.,* 23:3, 470-472 (1952).

# Literature: Forward UQ


- H. Najm, “Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics”, *Ann. Rev. Fluid Mech.*, 41:1, 35-52, (2009).
- B. Debusschere, H. Najm, P. Pébay, O. Knio, R. Ghanem, and O. Le Maître, “Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Processes”, *SIAM Journal on Scientific Computing,* 26:2, 698-719 (2004).
- D. Xiu, D., G. Karniadakis, “The Wiener-Askey Polynomial Chaos for Stocahstic Differential Equations”, *SIAM J. Sci. Comp.*, 24:2, 619-644, (2002).
- K. Kontolati , D. Loukrezis, D. Giovanis, L. Vandanapu, M. Shields, "A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems", 464, 111313, (2022).
- D. Donoho, "Compressed Sensing", IEEE Transactions on Information Theory, 52:4, (2006).
- S. Babacan, R. Molina, A. Katsaggelos, "Bayesian Compressive Sensing Using Laplace Priors", IEEE Transactions on Image Processing, 19:1, (2010)
- J. Peng, J. Hampton, A. Doostan, "A weighted l1-minimization approach for sparse polynomial chaos expansions", Journal of Comp. Physics, 267, 92-111, (2014).
- J. Jakeman, M. Eldred, K. Sargsyan, "Enhancing l1-minimization estimates of polynomial chaos expansions using basis selection", Journal of Comp. Physics, 289, 18-34, (2015).
- I.M. Sobol, “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”, *Mathematics and Computers in Simulation*, 55:1-3, 271-280 (2001).
- A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, “Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index”, *Computer Physics Communications* 181:2, 259-270 (2010).
- T. Crestaux, O. Le Maître, J.-M. Martinez, “Polynomial chaos expansion for sensitivity analysis”, *Reliability Engineering & System Safety*, 94:7, 1161-1172 (2009).
- K. Sargsyan, B. Debusschere, H. Najm, and O. Le Maître, “Spectral Representation and Reduced Order Modeling of the Dynamics of Stochastic Reaction Networks via Adaptive Data Partitioning”, *SIAM Journal on Scientific Computing*, 31:6 (2010).
- K. Sargsyan, C. Safta, H. Najm, B. Debusschere, D. Ricciuto, P. Thornton, “Dimensionality reduction for complex models via Bayesian compressive sensing”, *Int. J. Uncertainty Quantification*, 4:1, 63-93, (2014).
- K. Sargsyan, “Surrogate models for uncertainty propagation and sensitivity analysis”, in *Handbook of Uncertainty Quantification*, ed.: H. Owhadi, R. Ghanem, D. Higdon, Springer, pp. 673–698 (2017).
- D. Ricciuto, K. Sargsyan, P. Thornton, “The Impact of Parametric Uncertainties on Biogeochemistry in the E3SM Land Model”, *J of Advances in Modeling Earth Systems*, 10:2, 297-319, (2018).
- J. Mueller, K. Sargsyan, H. Najm, “Polynomial Chaos Surrogate Construction for Stochastic Models with Parametric Uncertainty”, *SIAM/ASA Journal on Uncertainty Quantification*, 13:1, (2025).

# Literature: Inverse UQ


- Y. Marzouk, H. Najm, “Dimensionality Reduction and Polynomial Chaos Acceleration of Bayesian Inference in Inverse Problems”, *J. Comp. Phys.*, 228:6, 1862-1902, (2009).
- R. Baptista, Y. Marzouk, O. Zahm, "On the Representation and Learning of Monotone Triangular Transport Maps", Foundations of Computational Mathematics, 24, 2063-2108, (2024)
- M. Parno, Y. Marzouk, “Transport Map Accelerated Markov Chain Monte Carlo”, SIAM/ASA Journal on Uncertainty Quantification, 6:2, (2018).
- M. Sunnåker, A. Busetto, E. Numminen, J. Corander, M. Foll, C. Dessimoz, “Approximate Bayesian Computation”. PLoS Comput Biol 9(1): e1002803 (2013).
- A. Ganguly, S. Jain, U. Watchareeruetai, "Amortized Variational Inference: A Systematic Review", Journal of Artificial Intelligence Research 78, 167-215 (2023).
- D. Blei, A. Kucukelbir, J. McAuliffe, "Variational Inference: A Review for Statisticians", Journal of the American Statistical Association, 112:518, 859-877 (2017).
- T. Cui, K. Law, Y. Marzouk, "Dimension-independent likelihood-informed MCMC", Journal of Computational Physics, 304, 109-137 (2016).

- M. Kennedy and A. O’Hagan, “Bayesian calibration of computer models”, *Journal of the Royal Statistical Society, Series B*. 63, 425-464, (2001).
- K. Sargsyan, H. Najm, R. Ghanem, “On the Statistical Calibration of Physical Models”, *Int. J. Chem. Kinetics*, 47:4, 246-276, (2015).
- K. Sargsyan, X. Huan, H. Najm. “Embedded Model Error Representation for Bayesian Model Calibration”, *Int. J. Uncertainty Quantification*, 9:4, (2019).

# ML4UQ: Traditional UQ in a language of (scientific) ML

| | | |
|---|---|---|
| Surrogate construction | → | Supervised ML |
| Dimensionality reduction | → | Unsupervised ML |
| Sensitivity analysis | → | Interpretability/Explainability |
| Optimal design | → | Active learning |
| Multifidelity analysis | → | Transfer learning |
| Extrapolation | → | OOD (out-of-distribution) |
| Rosenblatt map | → | Generative ML |


K. Sargsyan (ksargsy@sandia.gov)

## ML4UQ: Traditional UQ in a language of (scientific) ML

| | | |
|---|---|---|
| • Surrogate construction | ⟶ | • Supervised ML |
| • Dimensionality reduction | ⟶ | • Unsupervised ML |
| • Sensitivity analysis | ⟶ | • Interpretability/Explainability |
| • Optimal design | ⟶ | • Active learning |
| • Multifidelity analysis | ⟶ | • Transfer learning |
| • Extrapolation | ⟶ | • OOD (out-of-distribution) |
| • Rosenblatt map | ⟶ | • Generative ML |

## UQ4ML: Traditional (and not so much) UQ as a tool for ML

**Rest of the talk: overview of UQ for NN methods**

# Probabilistic NN == Bayesian NN

Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence". Nature, 2015

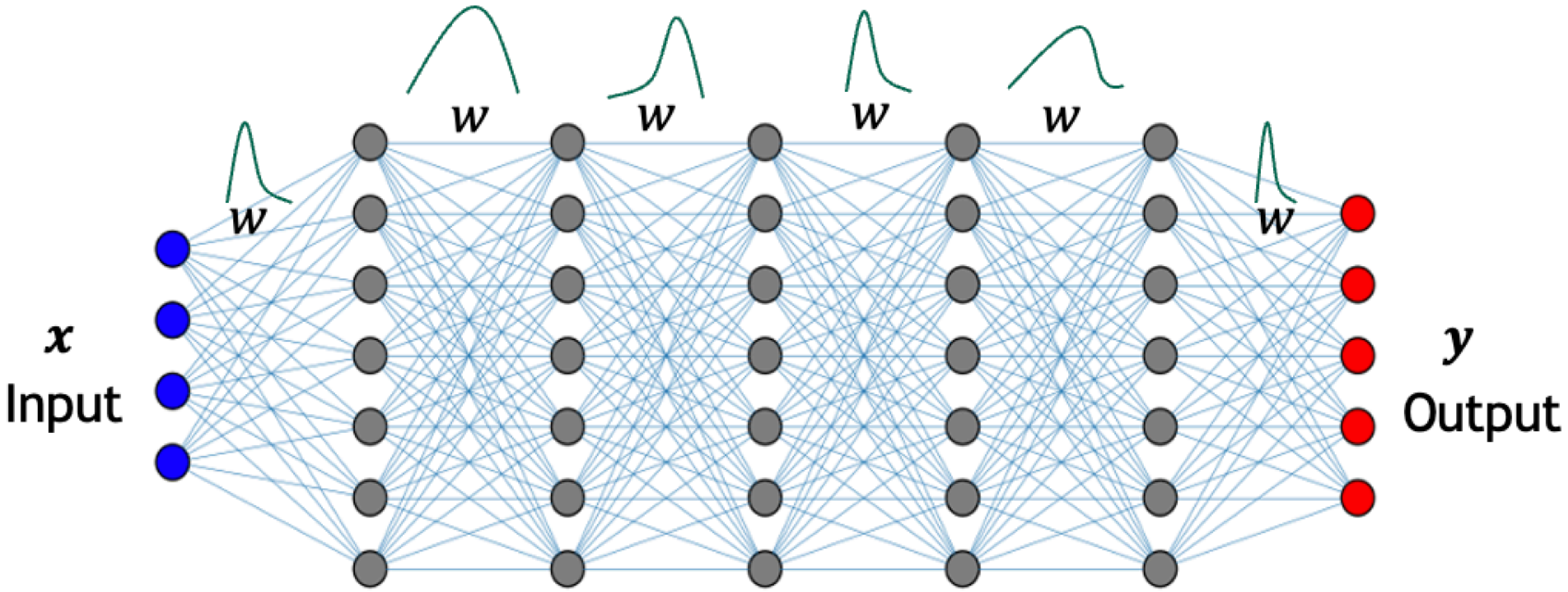
*"Nearly all approaches to probabilistic programming are **Bayesian** since it is hard to create other coherent frameworks for automated reasoning about uncertainty"*

- Bayesian NN methods have been around since 90s *[MacKay, 1992; Neal, 1996]*

- Full Bayesian treatment was infeasible back then….
  - … and still is, generally, not industry-standard by any means.

# UQ-for-NN: Bayesian perspective

Training for NN weights reformulated as a Bayesian inference problem




Posterior, Likelihood, Prior:

$$p(w|y) \propto p(y|w)\, p(w)$$

$$\propto \exp\left(-\frac{||y - f_w(x)||^2}{2\sigma^2}\right) \exp\left(-\frac{||w||^2}{2\lambda^2}\right)$$

Negative Log-Posterior $\simeq a||y - f_w(x)||^2 + b||w||^2 \simeq$ Training Loss Function

- ✓ Markov chain Monte Carlo (MCMC) sampling; Hamiltonian MC *[Levy, 2018]*
- ◉ Tuning is an art: essentially infeasible outside academic examples

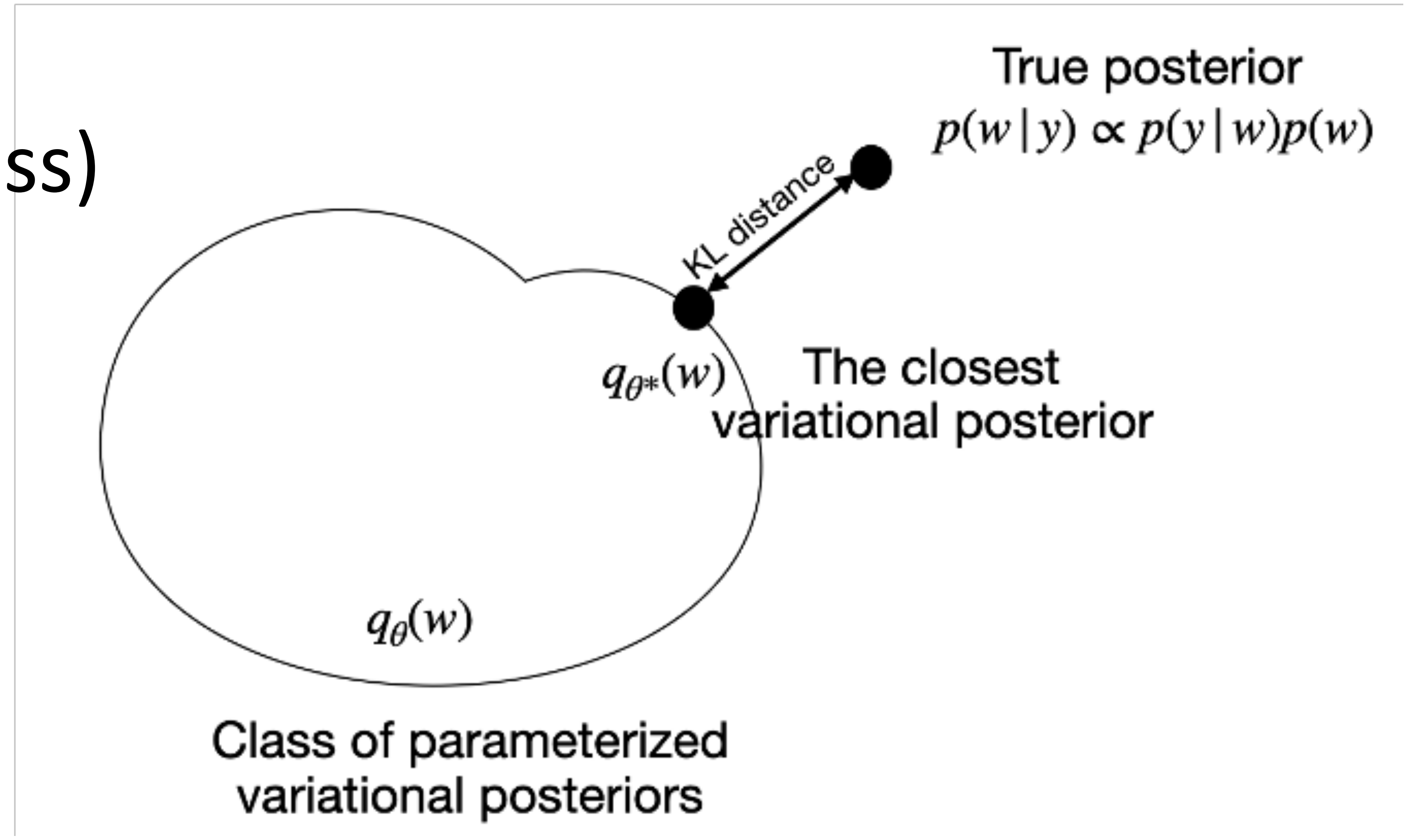

K. Sargsyan (ksargsy@sandia.gov)

# UQ-for-NN: Variational Approximation

- Bayes by Backprop *[Blundell, 2015]*
  - has become mainstream in ML literature
  - also called BNN

  - Mean-field VI (i.e. i.i.d. normal variational class)
  - Reparameterization trick
  - Gaussian mixture prior: wide and narrow
  - Variational st.dev. $\sigma = ln(1 + e^{\rho})$

- SVI, ADVI, BBVI, BBBVI, CCVI, CATVI, ….

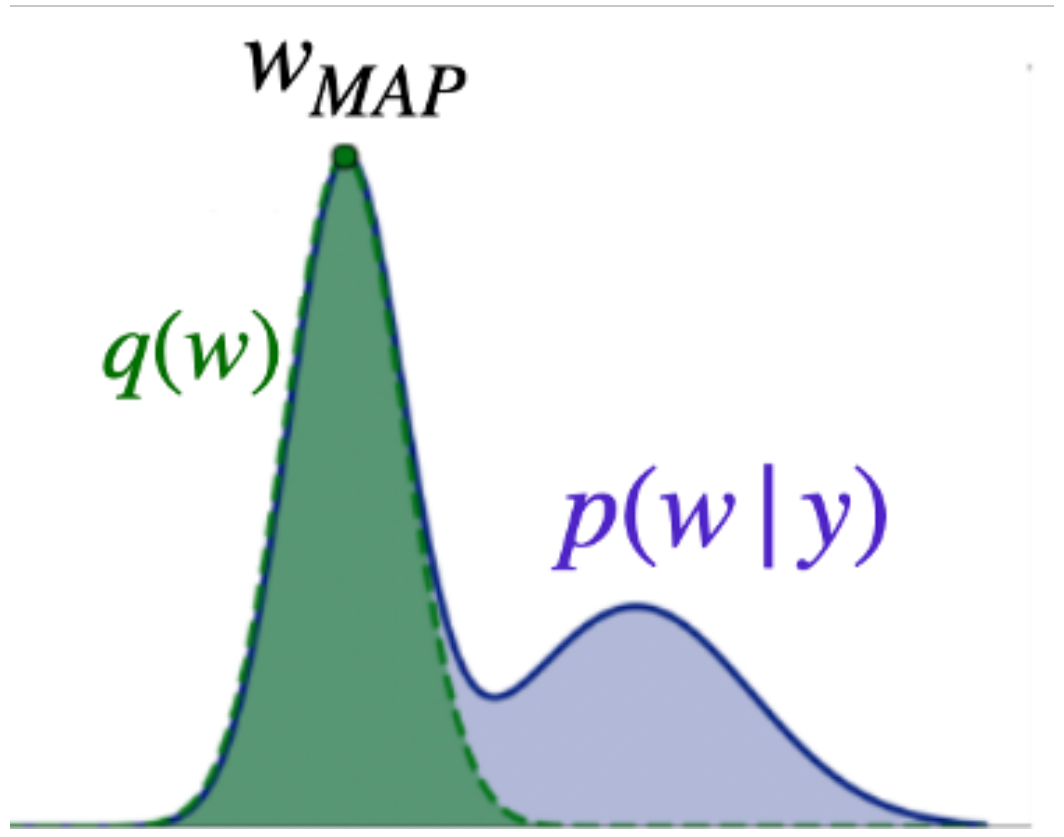



◉ Typically underestimates predictive uncertainty
◉ Restricted to variational class
◉ Hard to train


K. Sargsyan (ksargsy@sandia.gov)

# UQ-for-NN: Approximate Methods

- **Probabilistic backprop, or PBP** *[Hernandez-Lobato, 2015]*
  - Layer-to-layer updates from $\mathcal{N}(\mu, \sigma^2)$ to $\mathcal{N}(\mu_{new}, \sigma^2_{new})$
  - Deriving back propagation formulas for this update
  - $\mu, \sigma^2 \rightarrow \mu_{new}, \sigma^2_{new}$ updates similar to PC propagation (first order HG-PC)
- **Tractable Approximate Gaussian Inference (TAGI)** *[Goulet, 2021]*
  - Gaussian analytical propagation of uncertainties
  - See TAGI talk tomorrow morning
- **Laplace methods**: *[Ritter, 2018]*
  - ✓ Relies on Gaussian apprx near maximum;
  - ✓ Can be generalized to GMM
  - ◉ Hessian computation challenging
  - ◉ Fails to explore the full posterior



# UQ-for-NN: other methods

- **Ensembling methods:** work surprisingly well!
  - ✓ Deep Ensembles *[Lakshminarayanan, 2017];*
  - ✓ Interpreting ensembles from Bayesian perspective *[Garipov, 2018; Fort, 2019]*
  - ✓ Randomized MAP Sampling *[Pearce, 2020]*
  - ✓ MC-Dropout *[Gal, 2015]*
  - ✓ Stochastic Weight Averaging – Gaussian (SWAG) *[Maddox, 2019]*:shipped w PyTorch1.6
  - ✓ Delta-UQ *[Anirudh, 2021]*
  - ✓ Ensemble-VI *[Olivier, 2021]*
  - ◉ Lacks theoretical backing; expense $\times N$ (albeit parallelizable)

- **Direct learning of predictive RV**
  - ✓ Distance-based methods *[Postels, 2022]*,
  - ✓ DEUP *[Lahlou, 2023]*,
  - ✓ AVUC *[Krishnan, 2020]*.

- **Other**
  - ✓ Information-bottleneck UQ *[Guo, 2023]*,
  - ✓ Conformal UQ *[Hu, 2022]*,
  - ✓ Bayesian Last Layer *[Watson, 2021]*.



K. Sargsyan (ksargsy@sandia.gov)

# Major challenges in UQ-for-NN

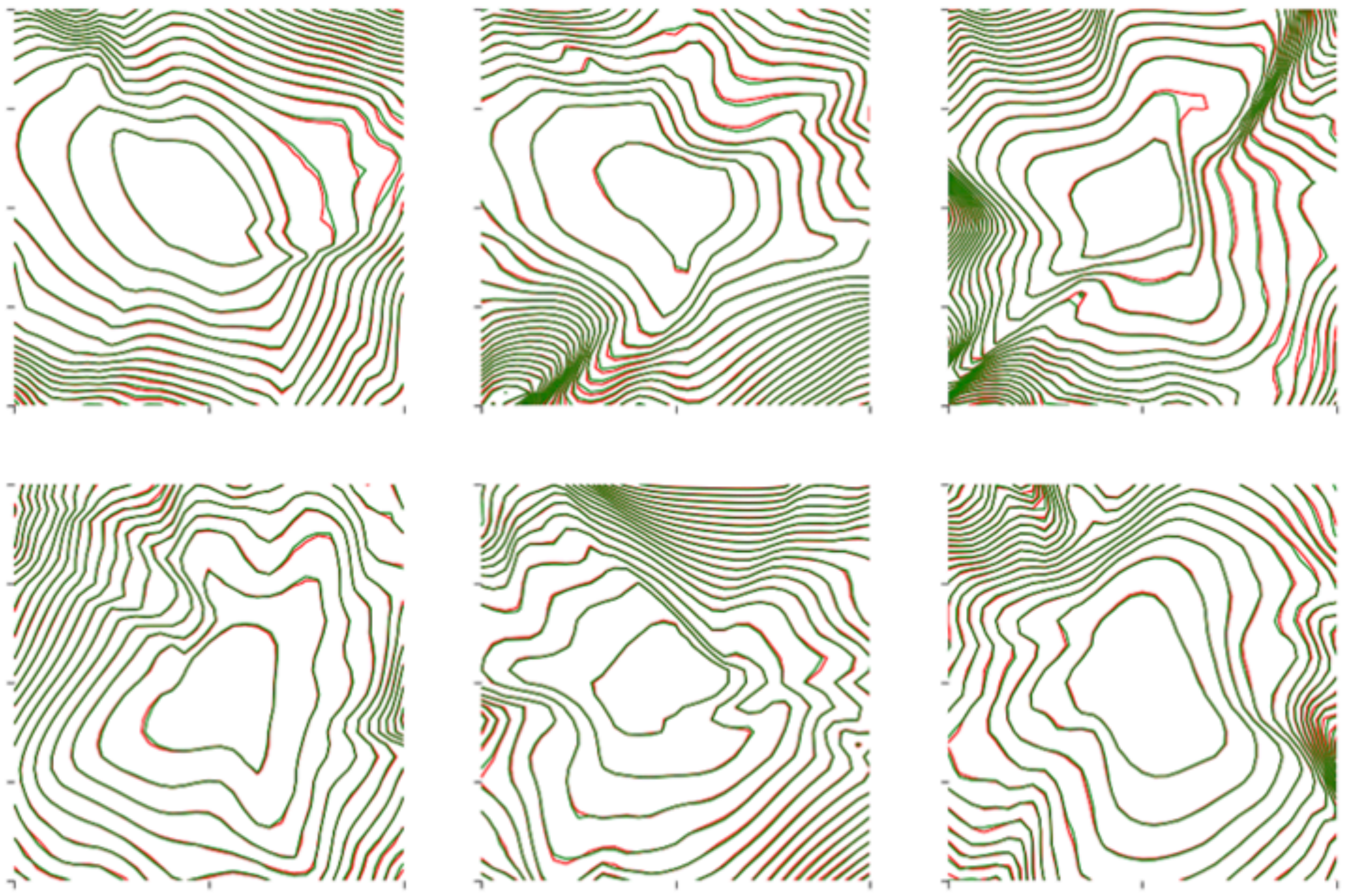
✓ Complicated posterior distribution (**loss surface**):

- visualization, categorization and analysis of loss surface is key to help understand and characterize NN performance [Wu, 2017; Li, 2018; Garipov, 2018; Fort, 2019; Yang, 2021, Liu, 2021; Geniesse, 2024; Xie, 2024].

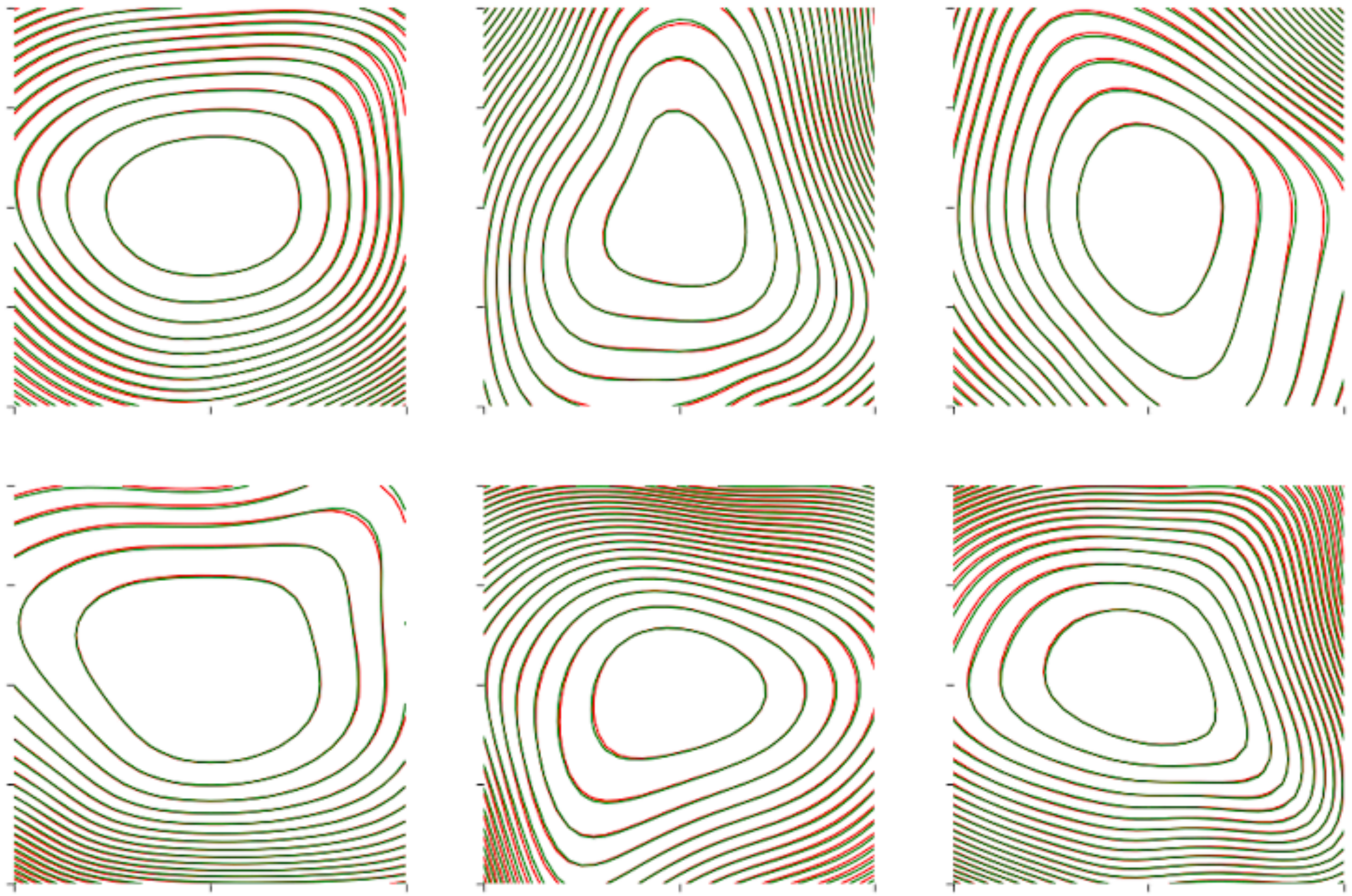
- invariances and symmetries: permuting some weights leads to the same loss,
- multimodality: multiple local minima in the weight space,
- “ridges”: low-d manifolds with same or similar loss
- incorporating prior knowledge should regularize the loss/log-posterior landscapes, making them more amenable to sampling and analysis.
- impact of architectural regularization:
  - physics-driven rewiring (invariance, symmetries, positivity),
  - numerical convenience (ResNet/NODE, weight reparam., layer/batch norm.)

# ResNet shortcuts regularize loss landscape

Conventional MLP: $x_{n+1} = \sigma(W_n x_n + b_n)$



ResNet: $x_{n+1} = x_n + \sigma(W_n x_n + b_n)$



See also [LI, 2018].


K. Sargsyan (ksargsy@sandia.gov)

# Major challenges in UQ-for-NN

✓ **Prior on weights** hard to elicit/interpret/defend:

- what does a uniform/gaussian prior on weight matrix elements mean?
- perhaps a prior is needed in the 'matrix'-space, or…
- how the prior should be related to initialization?
- driven by outputs or physics-constraints: function-space regularization, penalize for being away from prior *[Olivier, 2023]*.
- regularize with non-trivial centers, anchored ensembles, randomized MAP Sampling *[Pearce, 2020; Ghorbanian, 2024]*.

# Major challenges UQ-for-NN

✓ Large number of weights:

- scales linearly with depth and quadratically with width,
- hard to visualize the high-d surface,
- super high-d challenge of conventional inference,

- selective weight uncertainties (e.g. BLL)
- architectural regularization, e.g. weight parameterization.

# Weight-parameterization as an architectural regularization

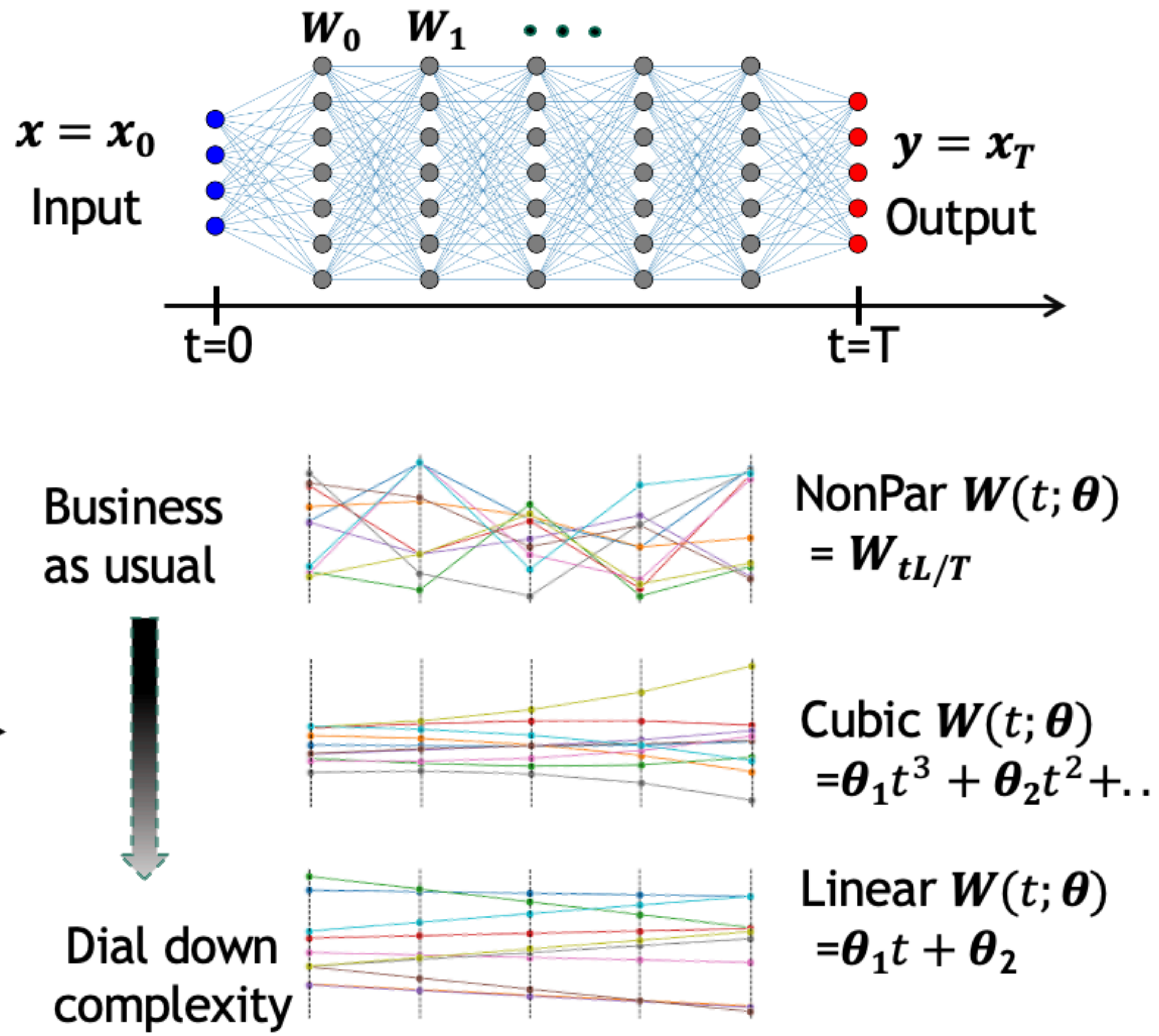
ResNet: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\alpha}_n \boldsymbol{\sigma}(\boldsymbol{W}_n \boldsymbol{x}_n + \boldsymbol{b}_n)$

Training for weight matrices $\boldsymbol{W}_0, \boldsymbol{W}_1, \ldots$

Heavily overparameterized, does not generalize well

Parameterize $\boldsymbol{W}(t;\boldsymbol{\theta})$ and train for $\boldsymbol{\theta}$'s.

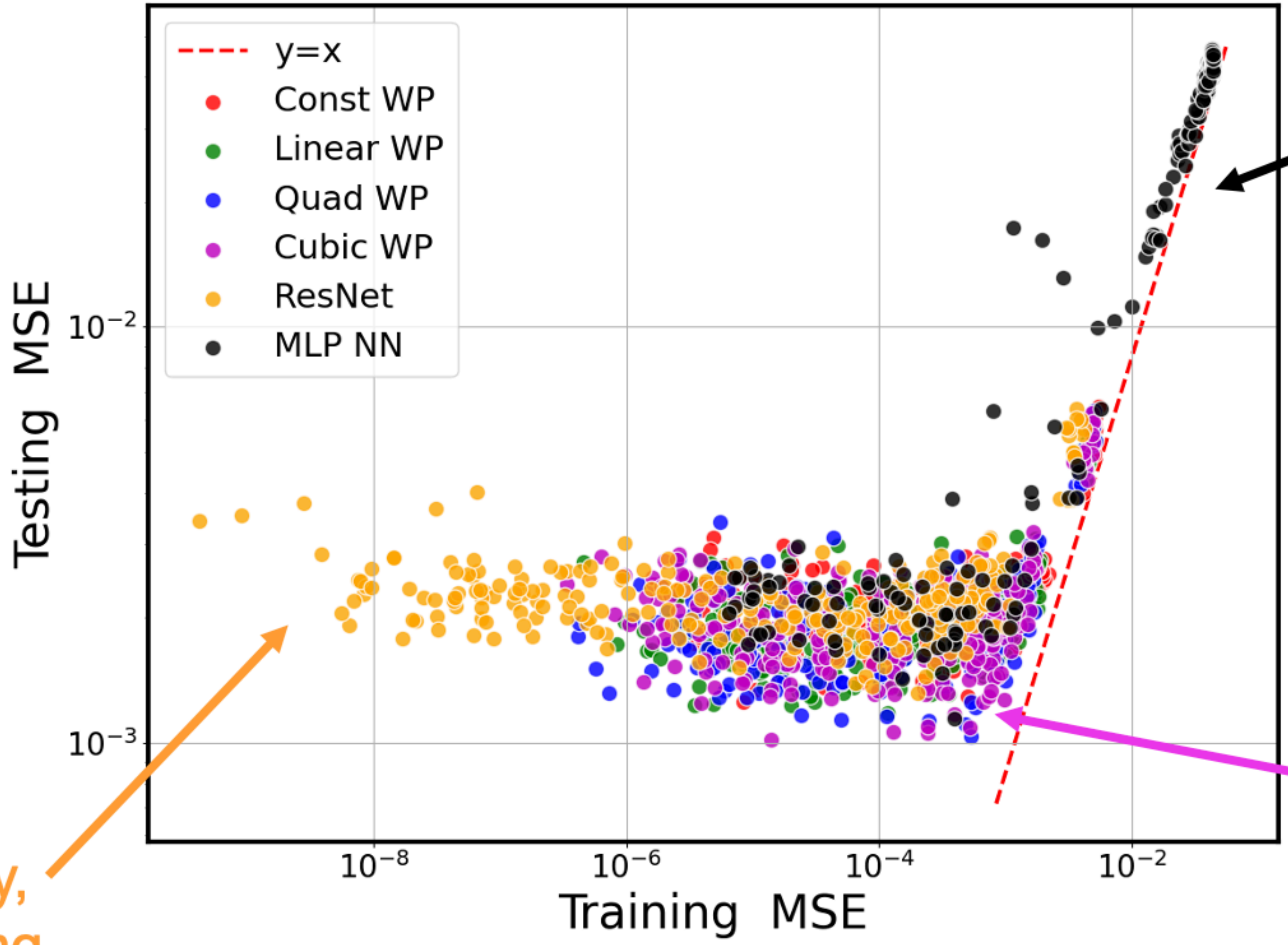
Parameterization of weight functions reduces capacity and improves generalization





K. Sargsyan (ksargsy@sandia.gov)

# Weight-parameterization as an architectural regularization




MLP:
bad accuracy

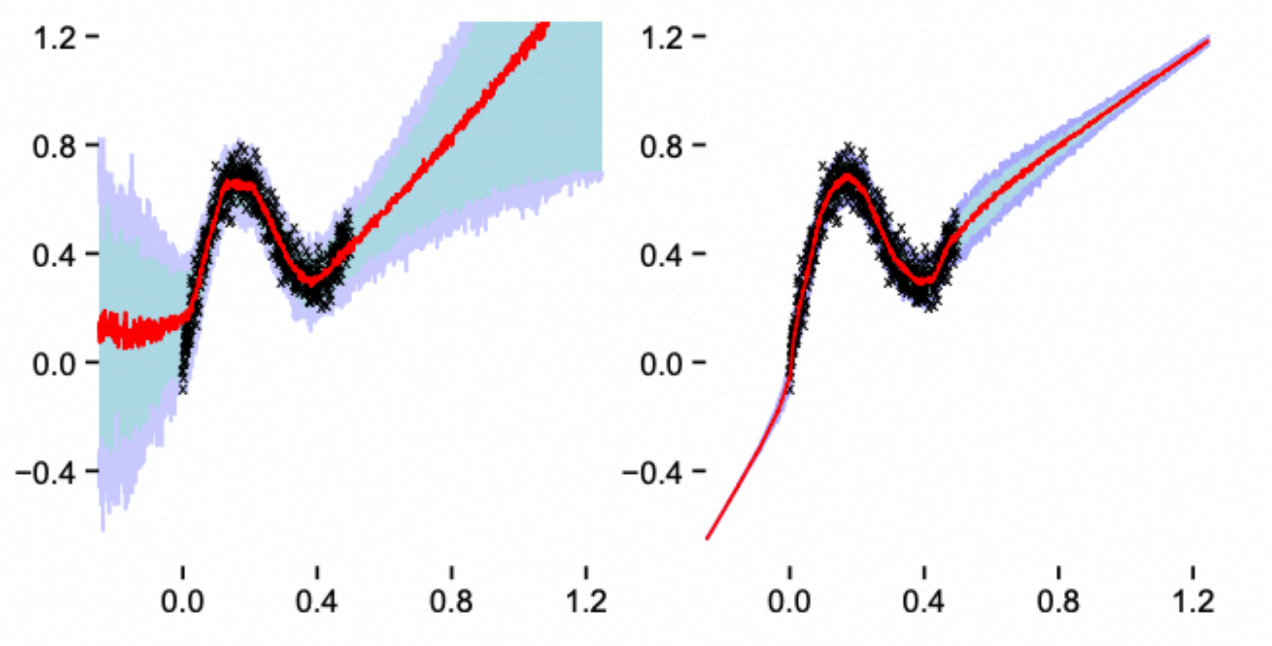
ResNet:
ok accuracy,
but overfitting

WP:
perfect tradeoff



K. Sargsyan (ksargsy@sandia.gov)

# Major challenges UQ-for-NN




✓ Established benchmarks:

- still a lot of eyeballing and 1d fit examples,
- striving to match a GP as a reference
- recent work specific to Bayesian NN *[Yao, 2019; Navratil, 2021; Nado, 2021; Staber, 2022; Basora, 2023]*.
- UCI Dataset, both regression and classification
  - https://github.com/treforevans/uci_datasets

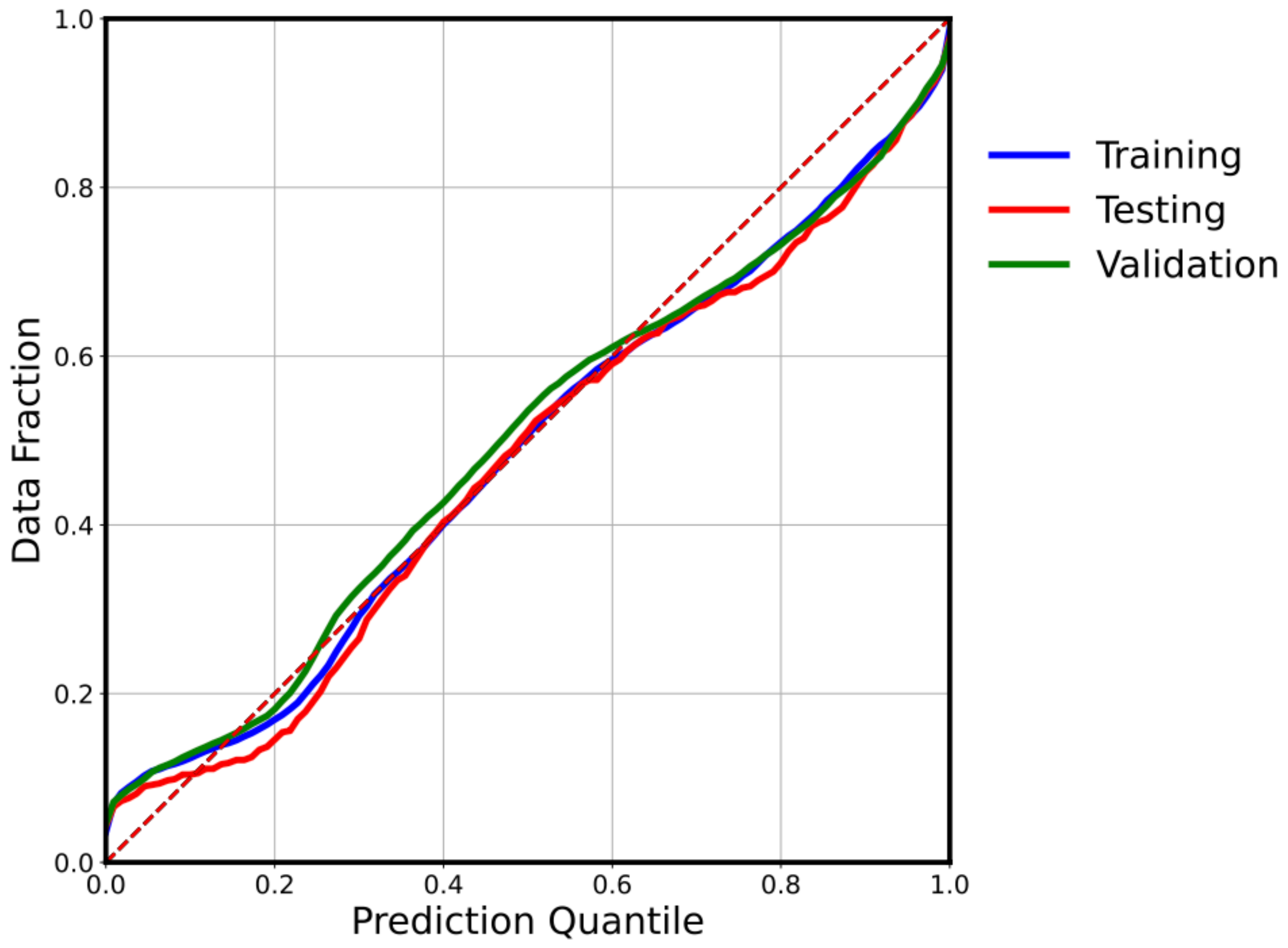

K. Sargsyan (ksargsy@sandia.gov)

# Major challenges in UQ-for-NN

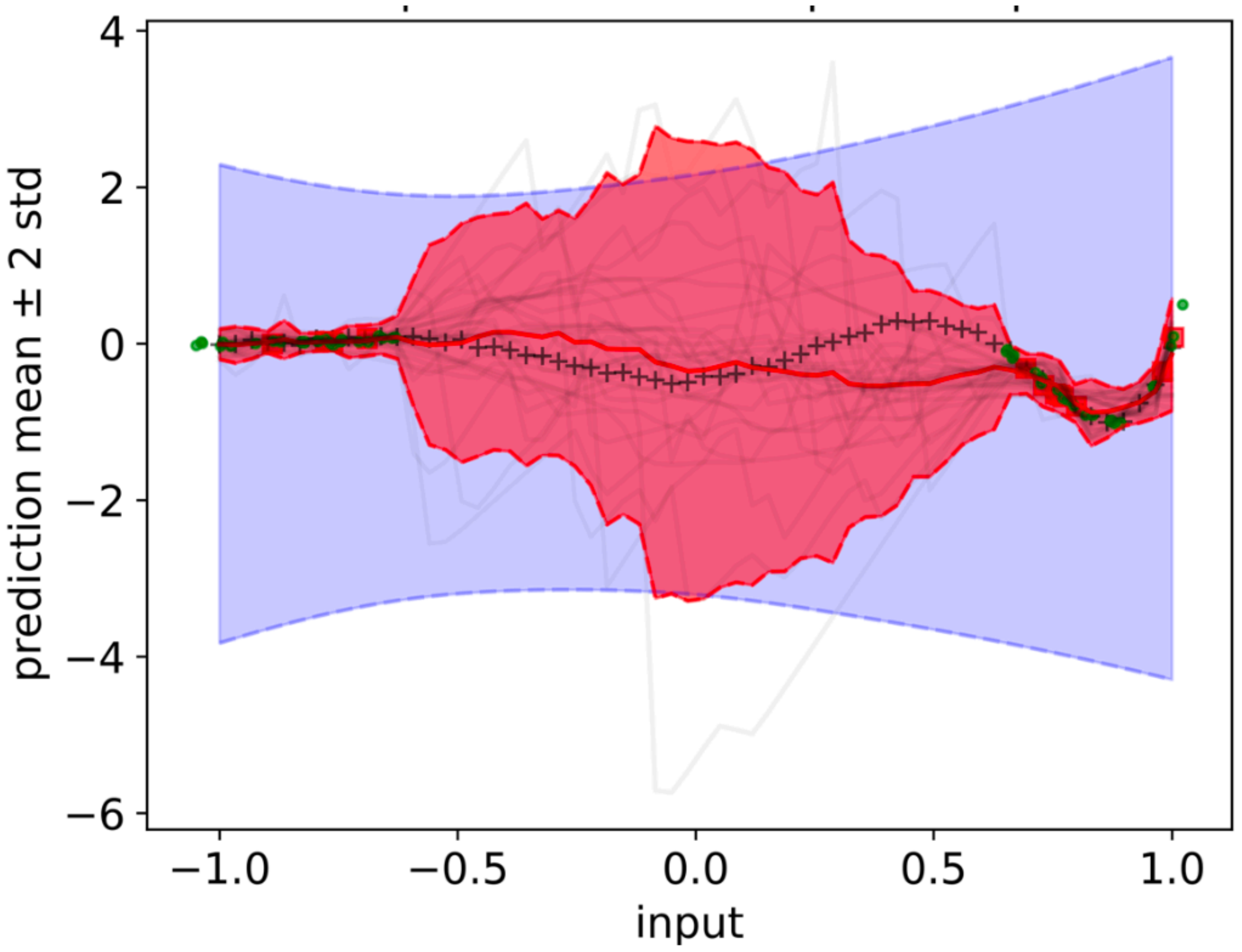
How to measure the quality of uncertainty estimation in NNs?

What part of a success is due to simple initialization?

Uncertainty-Accuracy Plot




Posterior predictive with no data —> Prior predictive





K. Sargsyan (ksargsy@sandia.gov)

# QUiNN (Quantification of Uncertainties in Neural Networks)

**Deterministic** — **torch.nn.module**

**Probabilistic** — **wrapper(torch.nn.module)**

**Usage:**

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

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

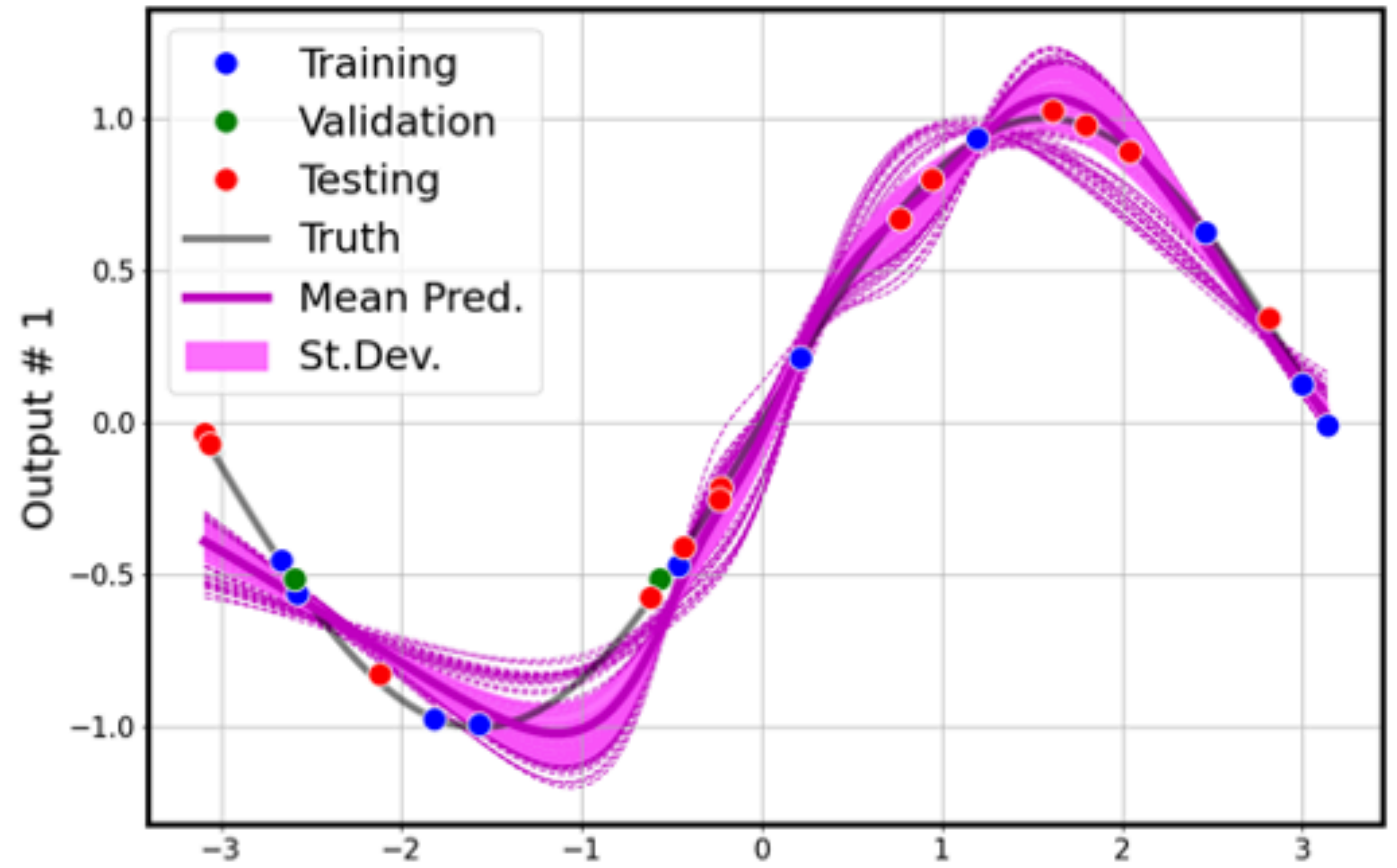
```
uqnet = VI_NN(nnet)
```

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

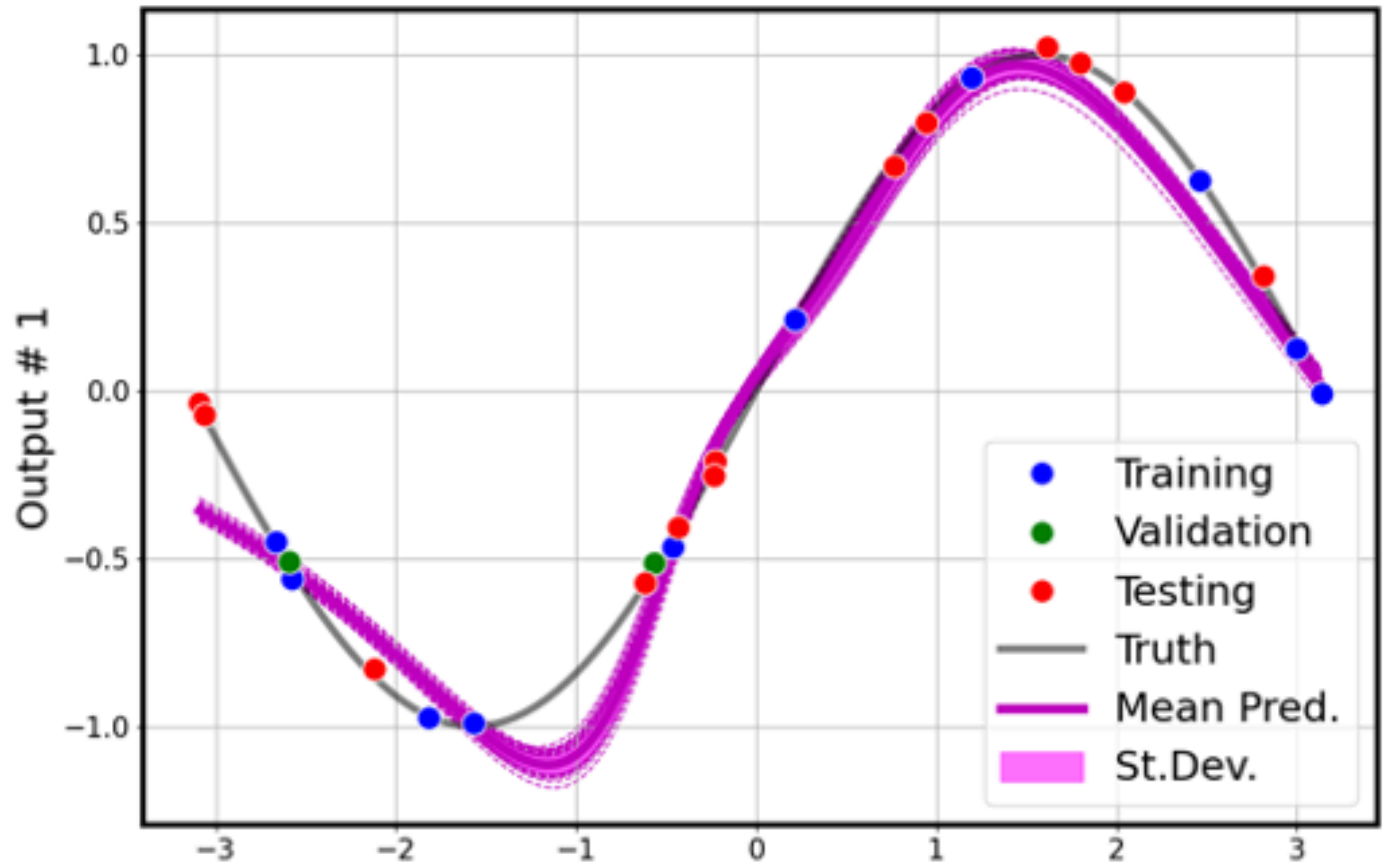
```
uqnet = Ens_NN(nnet, nens=nmc)
```

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

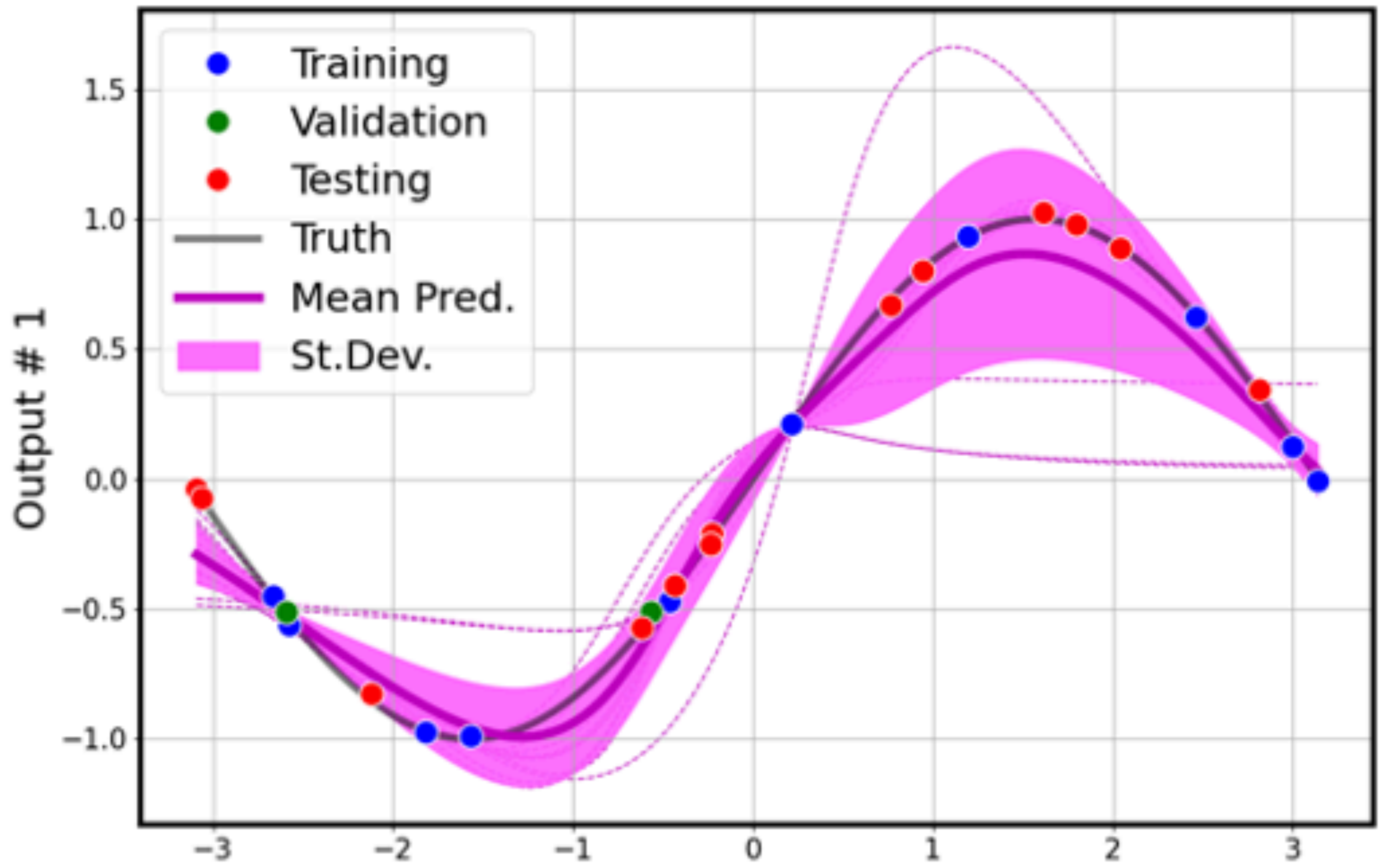
**Option 1: MCMC**




**Option 2: Variational Inference**




**Option 3: Ensembling**



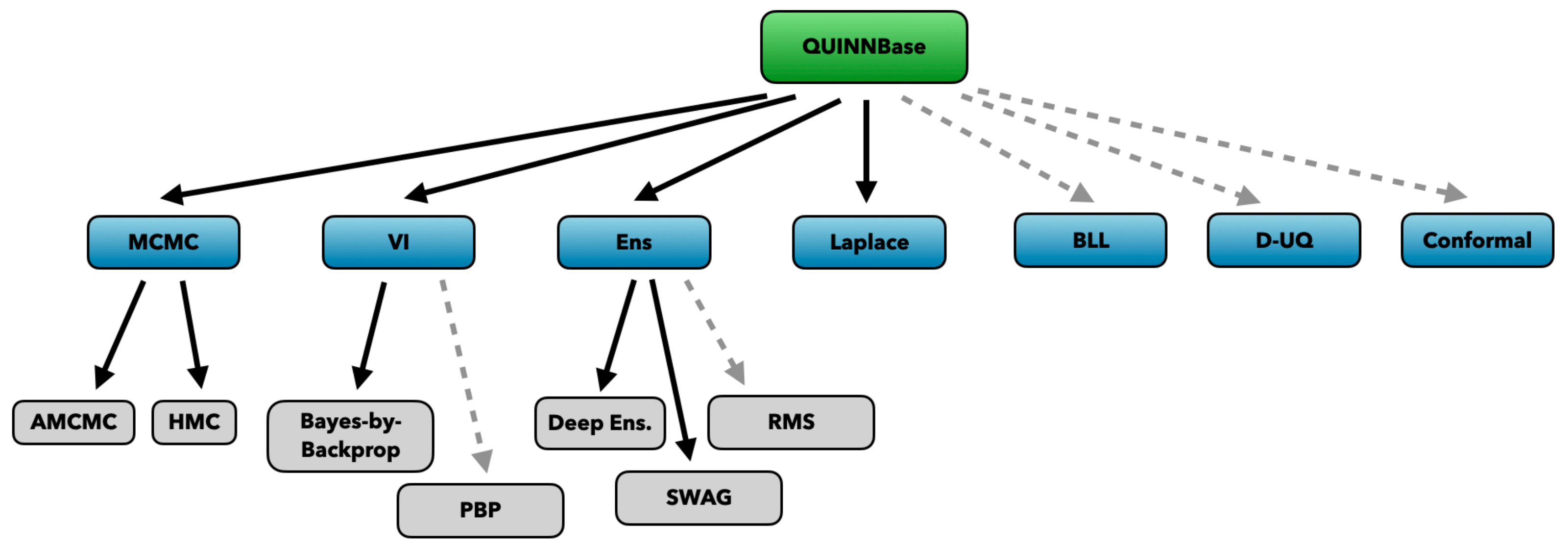

***github.com/sandialabs/quinn***

# UQ4NN Summary

- An attempt to overview the methods

- Major challenges (and ingredients to success)
  - Most methods rely on loss landscape
  - Meaning of priors/regularization and its interplay with initialization
  - Benchmarks and metrics/diagnostics of accuracy
  - High-dimensionality: selective augmentation of uncertainties, architectural regul.

- Implemented in QUiNN: [github.com/sandialabs/quinn](https://github.com/sandialabs/quinn) modular code as a wrapper to categories of methods (MCMC/HMC, VI, RMS, Ensembling, Laplace, SWAG)





K. Sargsyan (ksargsy@sandia.gov)

# UQ4NN Literature - I

**General probabilistic NN:**

- Z. Ghahramani, "Probabilistic machine learning and artificial intelligence". Nature 521, 452–459 (2015)
- D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks". Neural Computation 4 448–472 (1992)
- R. M. Neal, "Bayesian Learning for Neural Networks". Springer, New York (1996)

UQ for NN methods:

- D. Lévy, M. D. Hoffman, and J. Sohl-Dickstein, "Generalizing Hamiltonian Monte Carlo with Neural Networks". ICLR (2018)
- C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, "Weight uncertainty in neural networks". arXiv:1505.05424 (2015)
- J.M. Hernández-Lobato, R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks". ICML (2015)
- H. Ritter, A. Botev, D. Barber, “A Scalable Laplace Approximation for Neural Networks”, ICLR (2018)
- E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, P. Hennig, "Laplace Redux-Effortless Bayesian Deep Learning” Advances in neural inf. proc. systems 34 (2021)
- Y. Gal, Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning". ICML (2016)
- B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles". NIPS'17. 6405–6416 (2017)
- T. Pearce, F. Leibfried, A. Brintrup, "Uncertainty in Neural Networks: Approximately Bayesian Ensembling". Artificial Intelligence and Statistics, 108:234-244 (2020)s" Machine Learning: Science and Technology, 3-4 (2022)
- Y. Yang , L. Hodgkinson, R. Theisen, J. Zou, J. E. Gonzalez , K. Ramchandran, M. Mahoney. “Taxonomizing local versus global structure in neural network loss landscapes”, NIPS (2021)
- R. Anirudh, J. J. Thiagarajan. "Delta-UQ: Accurate Uncertainty Quantification via Anchor Marginalization", arxiv.org/abs/2110.02197 (2021)
- R. Krishnan, O. Tickoo, “Improving model calibration with accuracy versus uncertainty optimization”. arXiv:2012.07923 (2020)

# UQ4NN Literature - II


- W.J. Maddox, P Izmailov, T. Garipov, D.P. Vetrov, A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning", NIPS (2019)
- T Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, A. G-Wilson, "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs", NIPS (2018)
- S. Fort, H. Hu, B. Lakshminarayanan , Deep Ensembles: A Loss Landscape Perspective", arxiv.org/abs/1912.02757, (2019)
- H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, "Visualizing the Loss Landscape of Neural Nets, NIPS (2018)
- Y. Hu, J. Musielewicz, Z. W. Ulissi and A. J. Medford, "Robust and scalable uncertainty estimation with conformal prediction for machine-learned interatomic potentials" Machine Learning: Science and Technology, 3-4 (2022)
- L. Guo, H. Wu, W. Zhou, Y. Wang, T. Zhou, "IB-UQ: Information bottleneck based uncertainty quantification for neural function regression and neural operator learning", https://arxiv.org/abs/2302.03271 (2023)
- J. Postels, M. Segu, T. Sun, L. Sieber, L. Van Gool, F. Yu , F. Tombari, "On the Practicality of Deterministic Epistemic Uncertainty", ICLR (2022)
- J. Watson, J. A Lin, P. Klink, J. Pajarinen, J. Peters, "Latent Derivative Bayesian Last Layer Networks", AISTATS (2021)
- S. Lahlou, M. Jain, H. Nekoei, V. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, Y. Bengio "DEUP: Direct Epistemic Uncertainty Prediction", TMLR (2023)
- J.-A. Goulet, L. Nguyen, S. Amiri, Tractable Approximate Gaussian Inference for Bayesian Neural Networks, Journal of Machine Learning Research 22 (2021)
- T. Xie, et. al., "Evaluating Loss Landscapes from a Topology Perspective", arxiv.org/abs/2411.09807, (2024)
- C. Geniesse et. al, "Visualizing Loss Functions as Topological Landscape Profiles", Symmetry and Geometry in Neural Representations, (2024)
- A. Olivier, S. Mohammadi, A. Smyth, M. Adams, "Bayesian neural networks with physics-aware regularization for probabilistic travel time modeling", Comp.-aided Civil and Infrastructure Engineering, 38, 2614–2631, (2023)
- J. Ghorbanian, N. Casaprima, A. Olivier, "Empowering approximate Bayesian neural networks with functional priors through anchored ensembling for mechanics surrogate modeling applications", Computer Methods in Applied Mechanics and Engineering, 117645, (2024)

# UQ4NN Literature: Benchmarks

- UCI Dataset, https://archive.ics.uci.edu/datasets; Interface: https://github.com/treforevans/uci_datasets
- J. Yao, W. Pan, S. Ghosh, F. Doshi-Velez, "Quality of Uncertainty Quantification for Bayesian Neural Network Inference", https://arxiv.org/abs/1906.09686 (2019)
- J. Navratil, B. Elder, M. Arnold, S. Ghosh, P. Sattigeri, "Uncertainty Characteristics Curves: A Systematic Assessment of Prediction Intervals", https://arxiv.org/abs/2106.00858 (2021)
- Z. Nado et al. "Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning", https://arxiv.org/abs/2106.04015 (2021), https://github.com/google/uncertainty-baselines
- B. Staber, S. Da Veiga, "Benchmarking Bayesian neural networks and evaluation metrics for regression tasks", https://arxiv.org/abs/2206.06779 (2022)
- L. Basora, A. Viens, M. Arias Chao, X. Olive, "A Benchmark on Uncertainty Quantification for Deep Learning Prognostics", https://arxiv.org/abs/2302.04730 (2023)

# Trunk

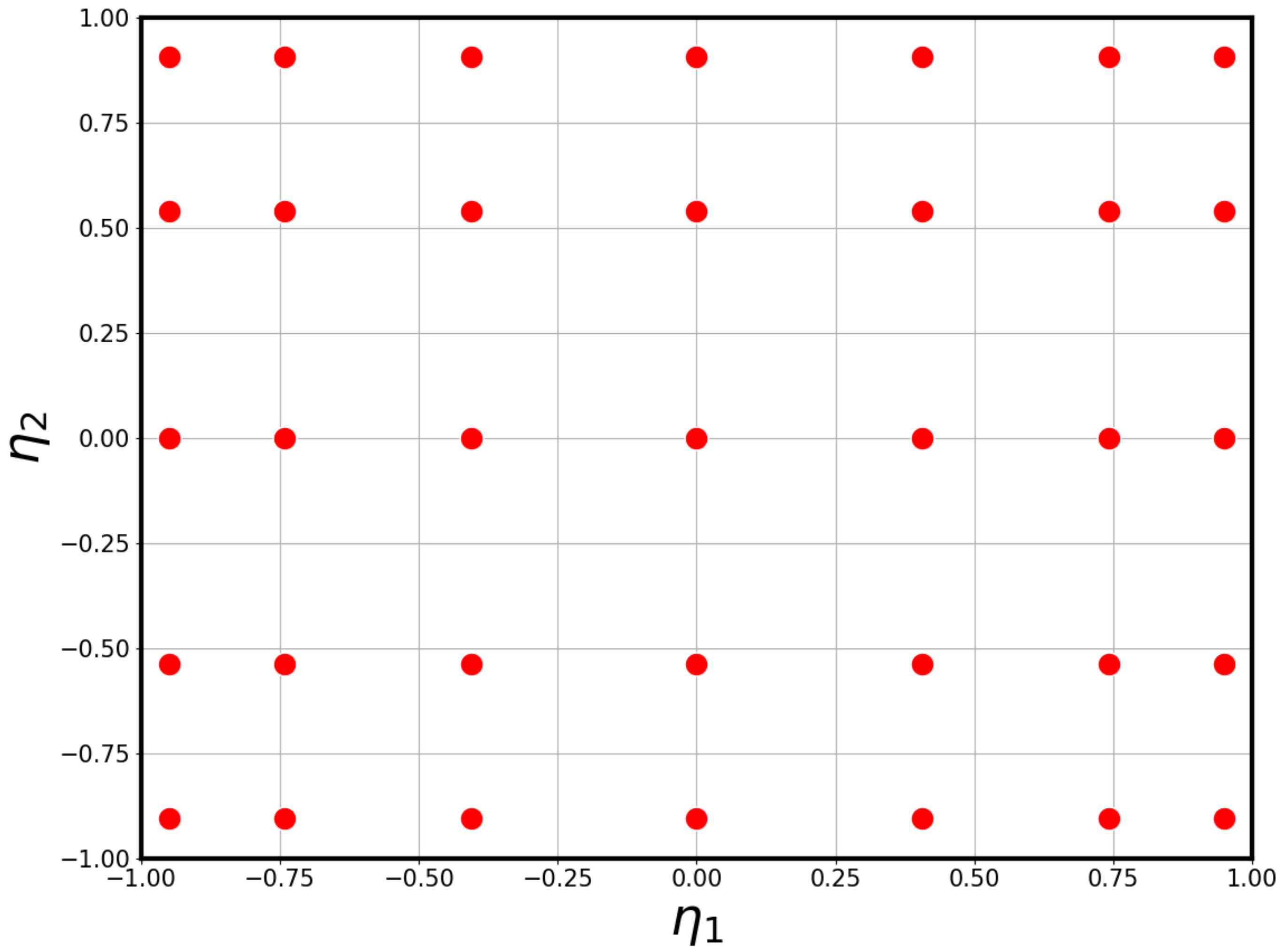

K. Sargsyan (ksargsy@sandia.gov)

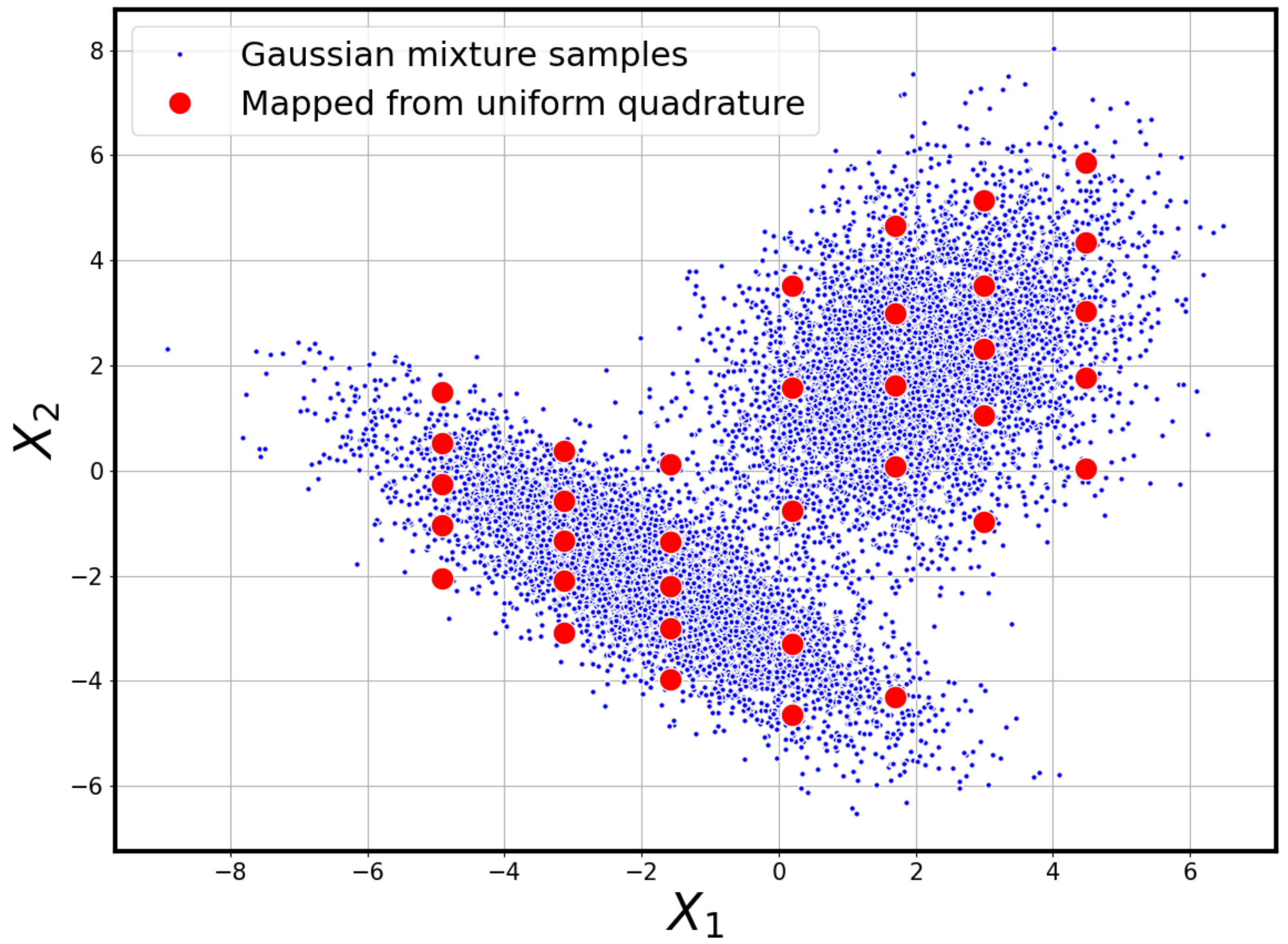
# Fwd UQ: Rosenblatt Transformation

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

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

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




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

# Fwd UQ: PC Moment Extraction

$$\boxed{X \simeq \sum_{k=0}^{p} x_k\, \psi_k(\xi)}$$

$$\int \psi_i(\xi)\psi_j(\xi)\pi_\xi(\xi)d\xi = ||\psi_i||^2 \delta_{ij}$$

- Orthogonality helps extract moments analytically

$$\mathbb{E}[X] = \int_\xi \sum_{k=0}^{p} x_k\, \psi_k(\xi)\ \pi(\xi)\ d\xi = x_0$$

$$\mathbb{V}[X] = \mathbb{E}[(X - x_0)^2] = \int_\xi \left(\sum_{k=1}^{p} x_k\, \psi_k(\xi)\right)\left(\sum_{m=1}^{p} x_m\, \psi_m(\xi)\right)\ \pi(\xi)\ d\xi = \sum_{k=1}^{p} x_k^2 ||\psi_k||^2$$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \; c_1\psi_1(\xi_1) \; + \; c_2\psi_1(\xi_2) \; + \; c_3\psi_1(\xi_3) \; +$$

$$+ \; c_4\psi_2(\xi_1) \; + \; c_5\psi_1(\xi_1)\psi_1(\xi_2) \; + \; c_6\psi_1(\xi_1)\psi_1(\xi_3) \; + \; c_7\psi_2(\xi_2) \; + \; c_8\psi_1(\xi_2)\psi_1(\xi_3) \; + \; c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \; c_1^2\langle\psi_1^2\rangle \; + \; c_2^2\langle\psi_1^2\rangle \; + \; c_3^2\langle\psi_1^2\rangle \; +$$

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_7^2\langle\psi_2^2\rangle \; + \; c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_9^2\langle\psi_2^2\rangle$$

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d+p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \boxed{c_1\psi_1(\xi_1)} + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ \boxed{c_4\psi_2(\xi_1)} + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \boxed{c_1^2\langle\psi_1^2\rangle} + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ \boxed{c_4^2\langle\psi_2^2\rangle} + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities $\boxed{\xi_1}$ $\xi_2$ $\xi_3$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10.$

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \ c_1\psi_1(\xi_1) \ + \boxed{c_2\psi_1(\xi_2)} + \ c_3\psi_1(\xi_3) \ +$$

$$+ \ c_4\psi_2(\xi_1) \ + \ c_5\psi_1(\xi_1)\psi_1(\xi_2) \ + \ c_6\psi_1(\xi_1)\psi_1(\xi_3) \ + \boxed{c_7\psi_2(\xi_2)} + \ c_8\psi_1(\xi_2)\psi_1(\xi_3) \ + \ c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \ c_1^2\langle\psi_1^2\rangle \ + \boxed{c_2^2\langle\psi_1^2\rangle} + \ c_3^2\langle\psi_1^2\rangle \ +$$

$$+ \ c_4^2\langle\psi_2^2\rangle \ + \ c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \boxed{c_7^2\langle\psi_2^2\rangle} + \ c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities $\xi_1$ $\boxed{\xi_2}$ $\xi_3$

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10.$

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \; c_1\psi_1(\xi_1) \; + \; c_2\psi_1(\xi_2) \; + \; \boxed{c_3\psi_1(\xi_3)} \; +$$

$$+ \; c_4\psi_2(\xi_1) \; + \; c_5\psi_1(\xi_1)\psi_1(\xi_2) \; + \; c_6\psi_1(\xi_1)\psi_1(\xi_3) \; + \; c_7\psi_2(\xi_2) \; + \; c_8\psi_1(\xi_2)\psi_1(\xi_3) \; + \; \boxed{c_9\psi_2(\xi_3)}$$

Variance contributions

$$Var(g) = 0 + \; c_1^2\langle\psi_1^2\rangle \; + \; c_2^2\langle\psi_1^2\rangle \; + \; \boxed{c_3^2\langle\psi_1^2\rangle} \; +$$

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_7^2\langle\psi_2^2\rangle \; + \; c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; \boxed{c_9^2\langle\psi_2^2\rangle}$$

Main effect sensitivities $\xi_1 \quad \xi_2 \quad \boxed{\xi_3}$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \boxed{c_1\psi_1(\xi_1)} + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ \boxed{c_4\psi_2(\xi_1)} + \boxed{c_5\psi_1(\xi_1)\psi_1(\xi_2)} + \boxed{c_6\psi_1(\xi_1)\psi_1(\xi_3)} + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \boxed{c_1^2\langle\psi_1^2\rangle} + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ \boxed{c_4^2\langle\psi_2^2\rangle} + \boxed{c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \boxed{c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Total sensitivities $\boxed{\xi_1}$ $\xi_2$ $\xi_3$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \ c_1\psi_1(\xi_1) \ + \boxed{c_2\psi_1(\xi_2)} + \ c_3\psi_1(\xi_3) \ +$$
$$+ \ c_4\psi_2(\xi_1) \ + \boxed{c_5\psi_1(\xi_1)\psi_1(\xi_2)} + \ c_6\psi_1(\xi_1)\psi_1(\xi_3) \ + \boxed{c_7\psi_2(\xi_2)} + \boxed{c_8\psi_1(\xi_2)\psi_1(\xi_3)} + \ c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \ c_1^2\langle\psi_1^2\rangle \ + \boxed{c_2^2\langle\psi_1^2\rangle} + \ c_3^2\langle\psi_1^2\rangle \ +$$
$$+ \ c_4^2\langle\psi_2^2\rangle \ + \boxed{c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \ c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \boxed{c_7^2\langle\psi_2^2\rangle} + \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \ c_9^2\langle\psi_2^2\rangle$$

Total sensitivities $\xi_1$ $\boxed{\xi_2}$ $\xi_3$

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \ c_1\psi_1(\xi_1) \ + \ c_2\psi_1(\xi_2) \ + \ \boxed{c_3\psi_1(\xi_3)} \ +$$

$$+ \ c_4\psi_2(\xi_1) \ + \ c_5\psi_1(\xi_1)\psi_1(\xi_2) \ + \ \boxed{c_6\psi_1(\xi_1)\psi_1(\xi_3)} + \ c_7\psi_2(\xi_2) \ + \ \boxed{c_8\psi_1(\xi_2)\psi_1(\xi_3)} + \boxed{c_9\psi_2(\xi_3)}$$

## Variance contributions

$$Var(g) = 0 + \ c_1^2\langle\psi_1^2\rangle \ + \ c_2^2\langle\psi_1^2\rangle \ + \ \boxed{c_3^2\langle\psi_1^2\rangle} \ +$$

$$+ \ c_4^2\langle\psi_2^2\rangle \ + \ c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ \boxed{c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \ c_7^2\langle\psi_2^2\rangle \ + \ \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \boxed{c_9^2\langle\psi_2^2\rangle}$$

Total sensitivities $\xi_1$ $\xi_2$ $\boxed{\xi_3}$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$, number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \ c_1\psi_1(\xi_1) \ + \ c_2\psi_1(\xi_2) \ + \ \boxed{c_3\psi_1(\xi_3)} \ +$$

$$+ \ c_4\psi_2(\xi_1) \ + \ c_5\psi_1(\xi_1)\psi_1(\xi_2) \ + \ \boxed{c_6\psi_1(\xi_1)\psi_1(\xi_3)} + \ c_7\psi_2(\xi_2) \ + \ \boxed{c_8\psi_1(\xi_2)\psi_1(\xi_3)} + \boxed{c_9\psi_2(\xi_3)}$$

## Variance contributions

$$Var(g) = 0 + \ c_1^2\langle\psi_1^2\rangle \ + \ c_2^2\langle\psi_1^2\rangle \ + \ \boxed{c_3^2\langle\psi_1^2\rangle} \ +$$

$$+ \ c_4^2\langle\psi_2^2\rangle \ + \ c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ \boxed{c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \ c_7^2\langle\psi_2^2\rangle \ + \ \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \boxed{c_9^2\langle\psi_2^2\rangle}$$

Total sensitivities $\xi_1 \quad \xi_2 \quad \boxed{\xi_3}$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \ c_1\psi_1(\xi_1) \ + \ c_2\psi_1(\xi_2) \ + \ c_3\psi_1(\xi_3) \ +$$

$$+ \ c_4\psi_2(\xi_1) \ + \ \boxed{c_5\psi_1(\xi_1)\psi_1(\xi_2)} + \ c_6\psi_1(\xi_1)\psi_1(\xi_3) \ + \ c_7\psi_2(\xi_2) \ + \ c_8\psi_1(\xi_2)\psi_1(\xi_3) \ + \ c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \ c_1^2\langle\psi_1^2\rangle \ + \ c_2^2\langle\psi_1^2\rangle \ + \ c_3^2\langle\psi_1^2\rangle \ +$$

$$+ \ c_4^2\langle\psi_2^2\rangle \ + \ \boxed{c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \ c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ c_7^2\langle\psi_2^2\rangle \ + \ c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \ + \ c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities $\boxed{(\xi_1, \xi_2)}$ $(\xi_1, \xi_3)$ $(\xi_2, \xi_3)$



K. Sargsyan (ksargsy@sandia.gov)

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \; c_1\psi_1(\xi_1) \; + \; c_2\psi_1(\xi_2) \; + \; c_3\psi_1(\xi_3) \; +$$

$$+ \; c_4\psi_2(\xi_1) \; + \; c_5\psi_1(\xi_1)\psi_1(\xi_2) \; + \; \boxed{c_6\psi_1(\xi_1)\psi_1(\xi_3)} + \; c_7\psi_2(\xi_2) \; + \; c_8\psi_1(\xi_2)\psi_1(\xi_3) \; + \; c_9\psi_2(\xi_3)$$

## Variance contributions

$$Var(g) = 0 + \; c_1^2\langle\psi_1^2\rangle \; + \; c_2^2\langle\psi_1^2\rangle \; + \; c_3^2\langle\psi_1^2\rangle \; +$$

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; \boxed{c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \; c_7^2\langle\psi_2^2\rangle \; + \; c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_9^2\langle\psi_2^2\rangle$$

## Joint sensitivities $(\xi_1, \xi_2)$ $\boxed{(\xi_1, \xi_3)}$ $(\xi_2, \xi_3)$

# Sensitivity indices are directly computable from PC

$$g(\boldsymbol{\xi}) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + \; c_1\psi_1(\xi_1) \; + \; c_2\psi_1(\xi_2) \; + \; c_3\psi_1(\xi_3) \; +$$

$$+ \; c_4\psi_2(\xi_1) \; + \; c_5\psi_1(\xi_1)\psi_1(\xi_2) \; + \; c_6\psi_1(\xi_1)\psi_1(\xi_3) \; + \; c_7\psi_2(\xi_2) \; + \; \boxed{c_8\psi_1(\xi_2)\psi_1(\xi_3)} + \; c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + \; c_1^2\langle\psi_1^2\rangle \; + \; c_2^2\langle\psi_1^2\rangle \; + \; c_3^2\langle\psi_1^2\rangle \; +$$

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_7^2\langle\psi_2^2\rangle \; + \; \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \; c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities $(\xi_1, \xi_2)$ $(\xi_1, \xi_3)$ $\boxed{(\xi_2, \xi_3)}$

# Fwd UQ: High-D Input: Compressive sensing

Least-squares regression: $argmin_c ||f(\xi) - \sum_k c_k \Psi_k(\xi)||_{\ell_2}$

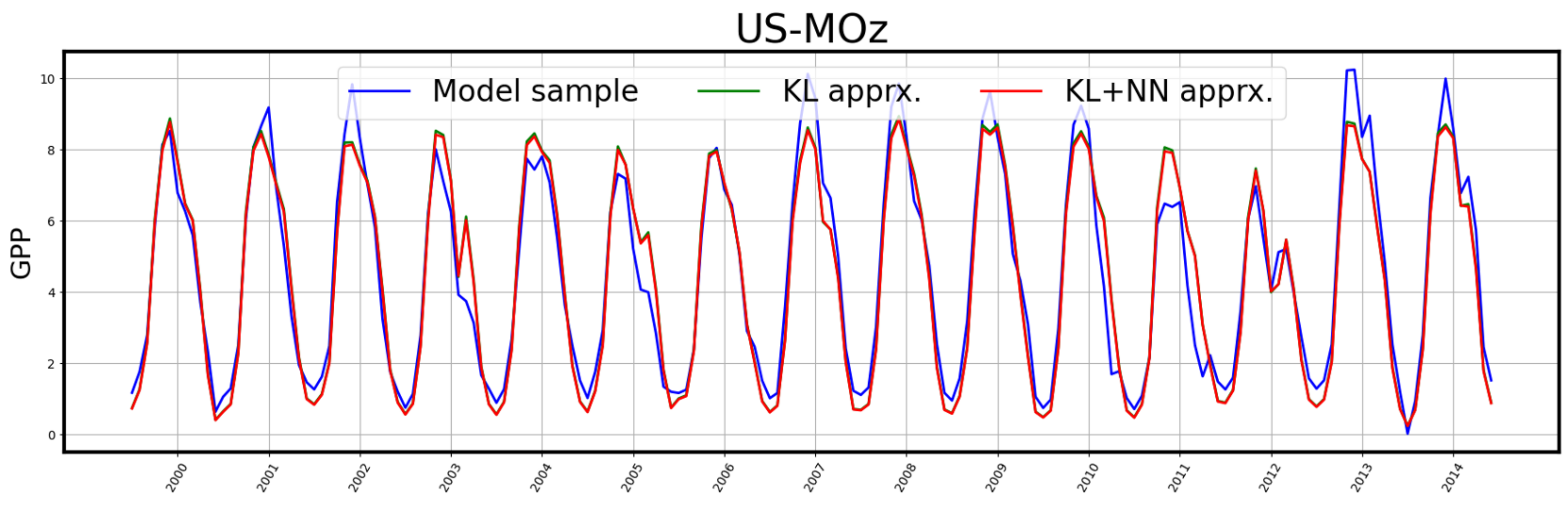
… in matrix notation: $argmin_c ||y - \Psi c||_2$

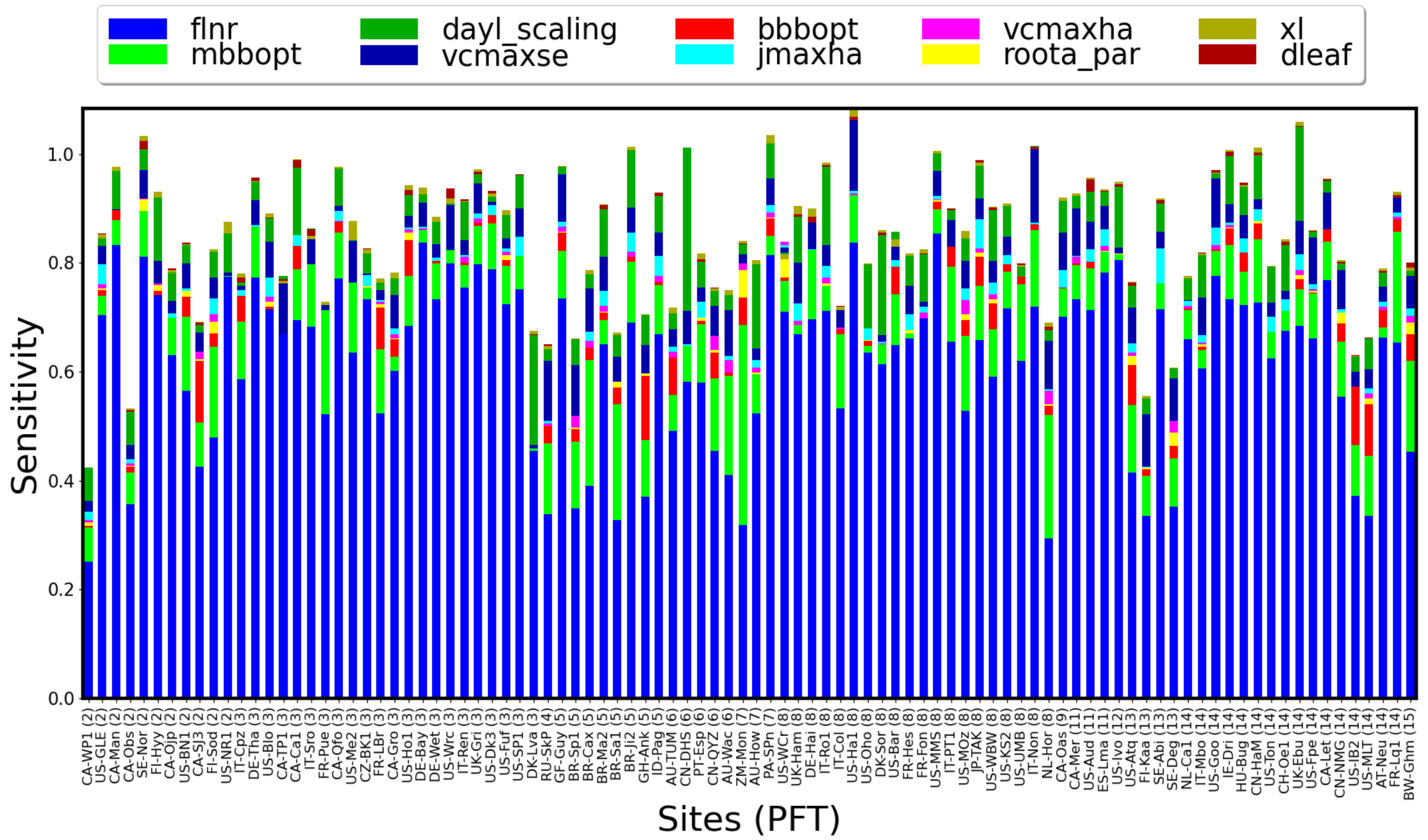
Tikhonov regularization, or Ridge regression (weight decay in ML language) $argmin_c ||y - \Psi c||_2^2 + ||c||_2^2$

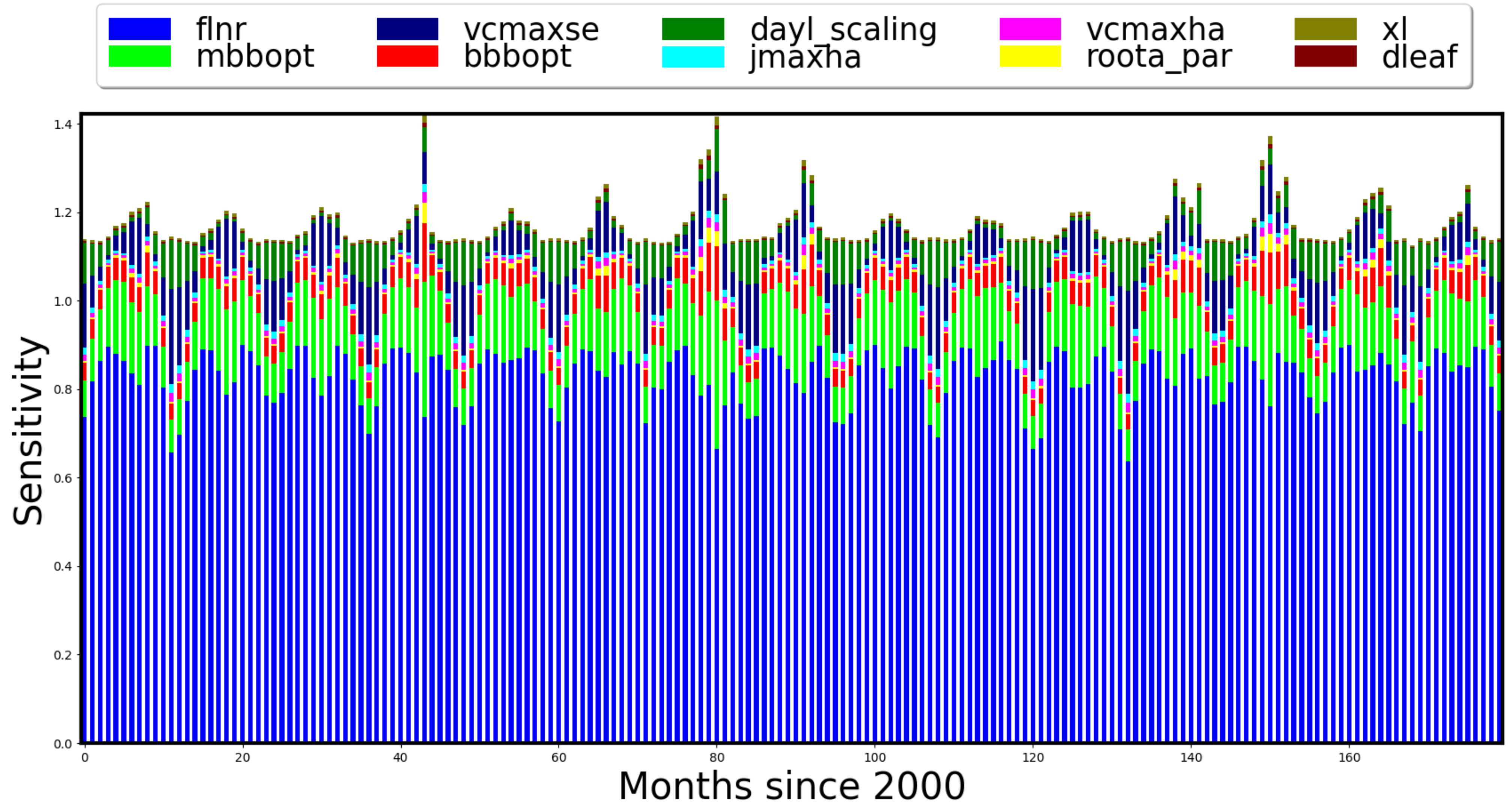
Sparsest solution: $argmin_c ||y - \Psi c||_2^2 + ||c||_0$ **Difficult Problem**

Compressive sensing, LASSO, Basis Pursuit: $argmin_c ||y - \Psi c||_2^2 + ||c||_1$ **Closest Convex apprx.**

# E3SM Land Model: Gross Primary Productivity



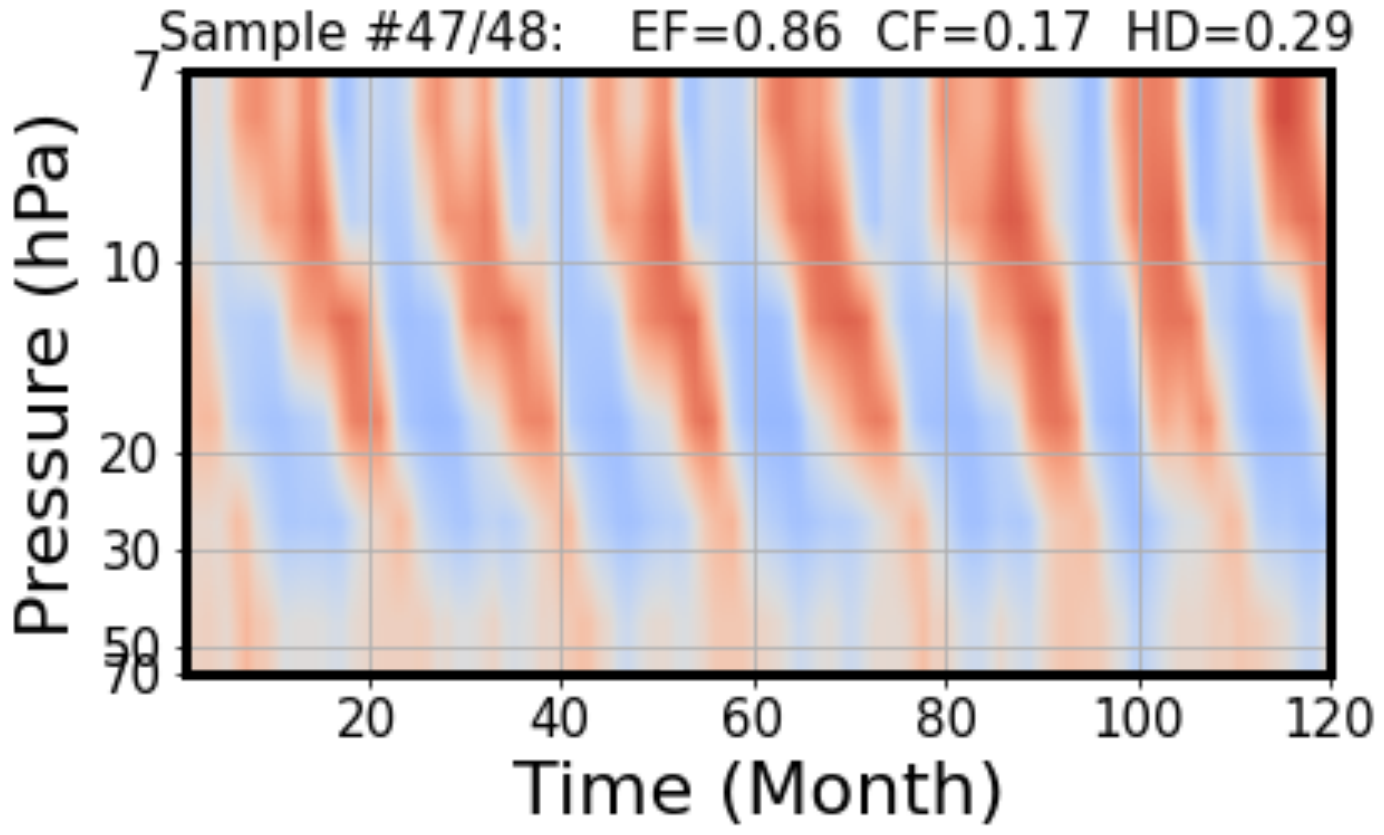



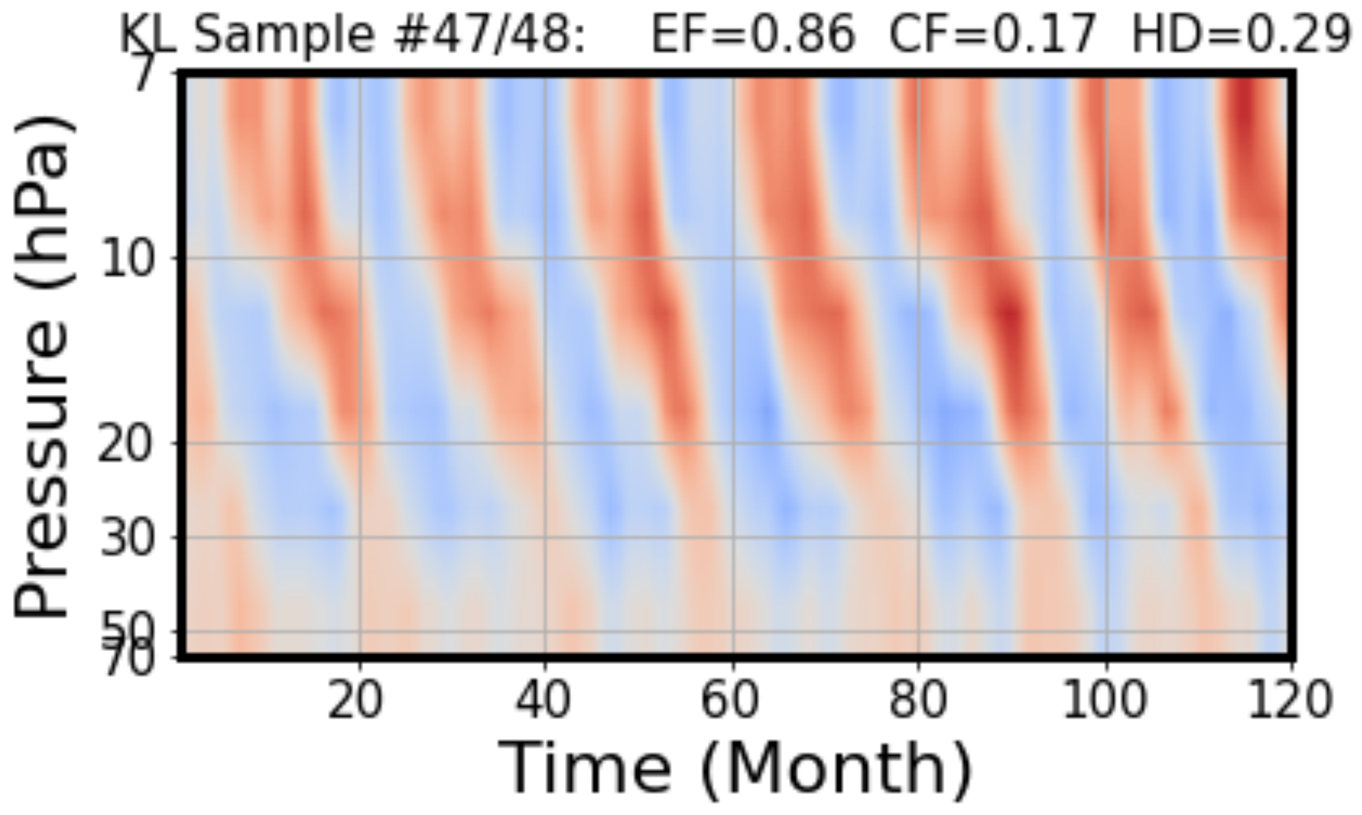



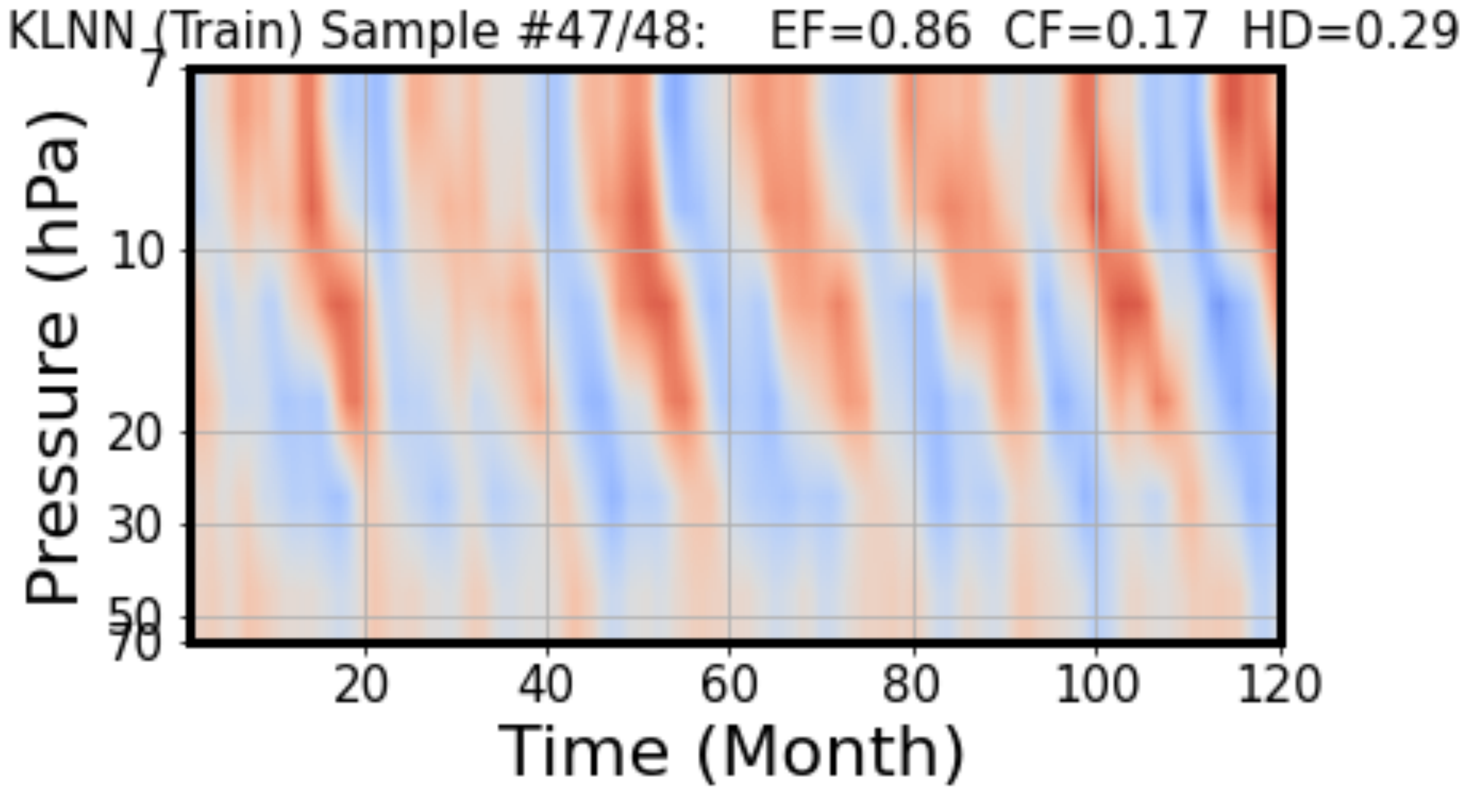



K. Sargsyan (ksargsy@sandia.gov)

# E3SM Quasi-Biennial Oscillation





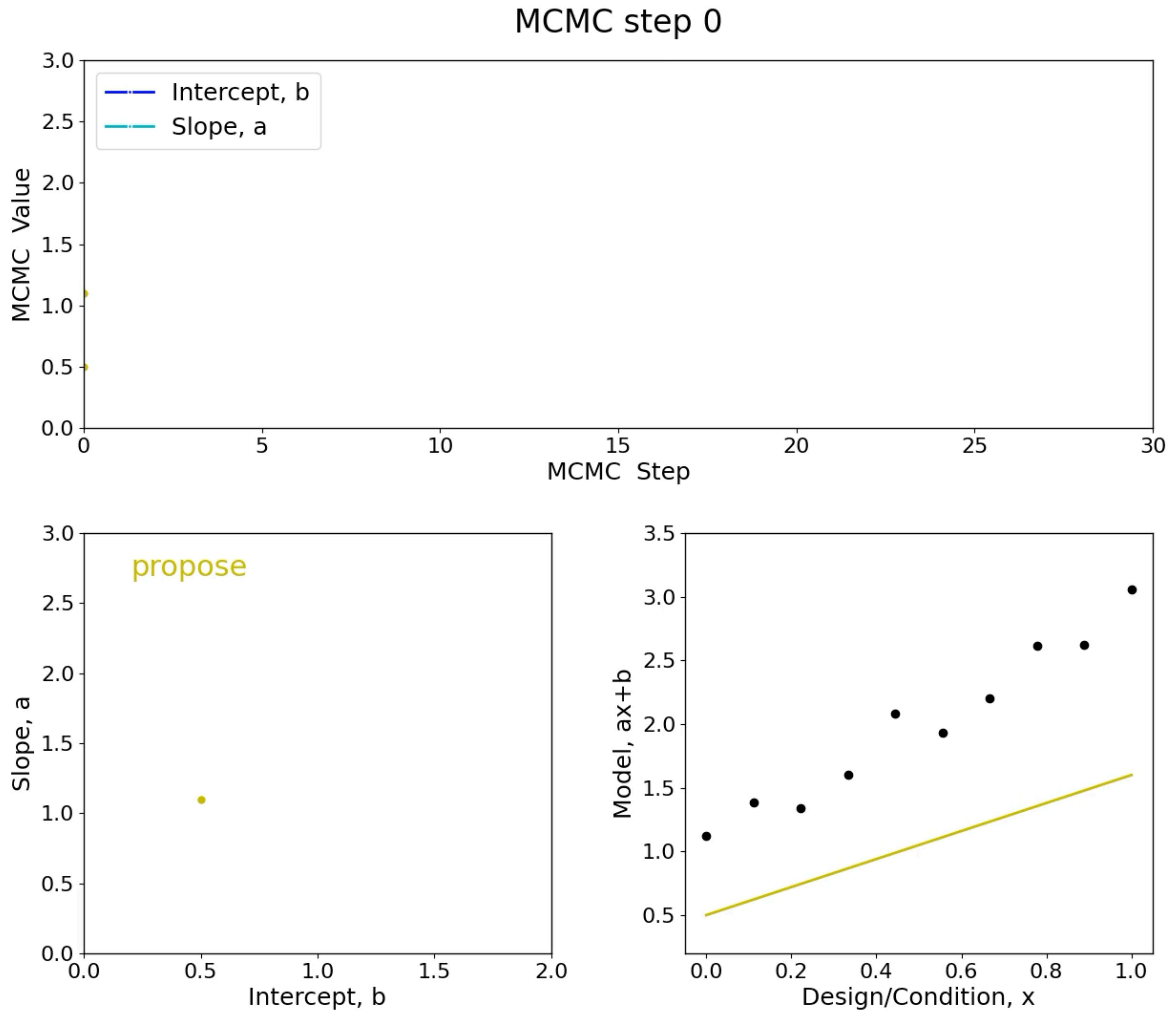

K. Sargsyan (ksargsy@sandia.gov)

# 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 $\alpha$ and move to step $t+1$.




*Note: only posterior ratio matters

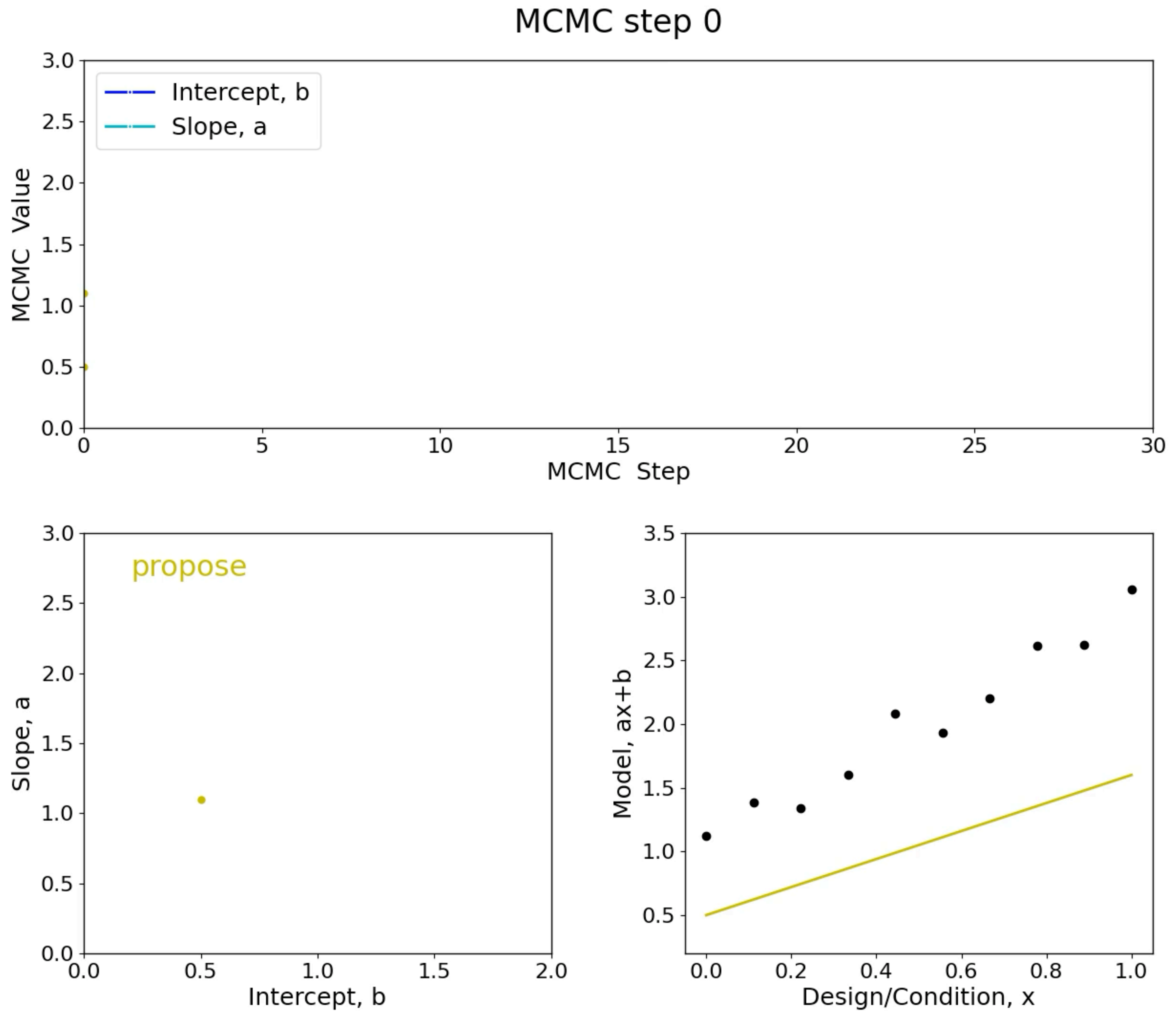

K. Sargsyan (ksargsy@sandia.gov)

# 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 $\alpha$ and move to step $t + 1$.




*Note: only posterior ratio matters


K. Sargsyan (ksargsy@sandia.gov)

# Inv UQ: EME Likelihood Construction

Non-intrusive setting : $$g(x_i) = f(\lambda + \delta_\alpha; x_i) + \epsilon_i$$

Price to pay: true likelihood is (near) degenerate:

$$p(g \,|\, \lambda, \alpha) = \pi_f(g)$$

need approximation:

gauss. marginal product $$p(g \,|\, \lambda, \alpha) \approx \prod_i p(g_i \,|\, \lambda, \alpha) \propto \prod_i \exp\left(-\frac{(g_i - \mu_i(\lambda, \alpha))^2}{2\sigma_i^2(\lambda, \alpha)}\right)$$

Approximate Bayesian Computation $$p(g \,|\, \lambda, \alpha) \approx \prod_i \exp\left(-\frac{(g_i - \mu_i(\lambda, \alpha))^2 + \gamma_1(\sigma_i(\lambda, \alpha) - \gamma_2 \,|\, g_i - \mu_i(\lambda, \alpha) \,|\,)^2}{2\epsilon^2}\right)$$

# Inv UQ: EME Predictive Variance

model error

$$\sum_k f_{ki}(\lambda, \alpha)\Psi_k(\xi) + \epsilon_i$$

MCMC (posterior uncertainty)

data noise

Optionally, and in practice, also surrogate error

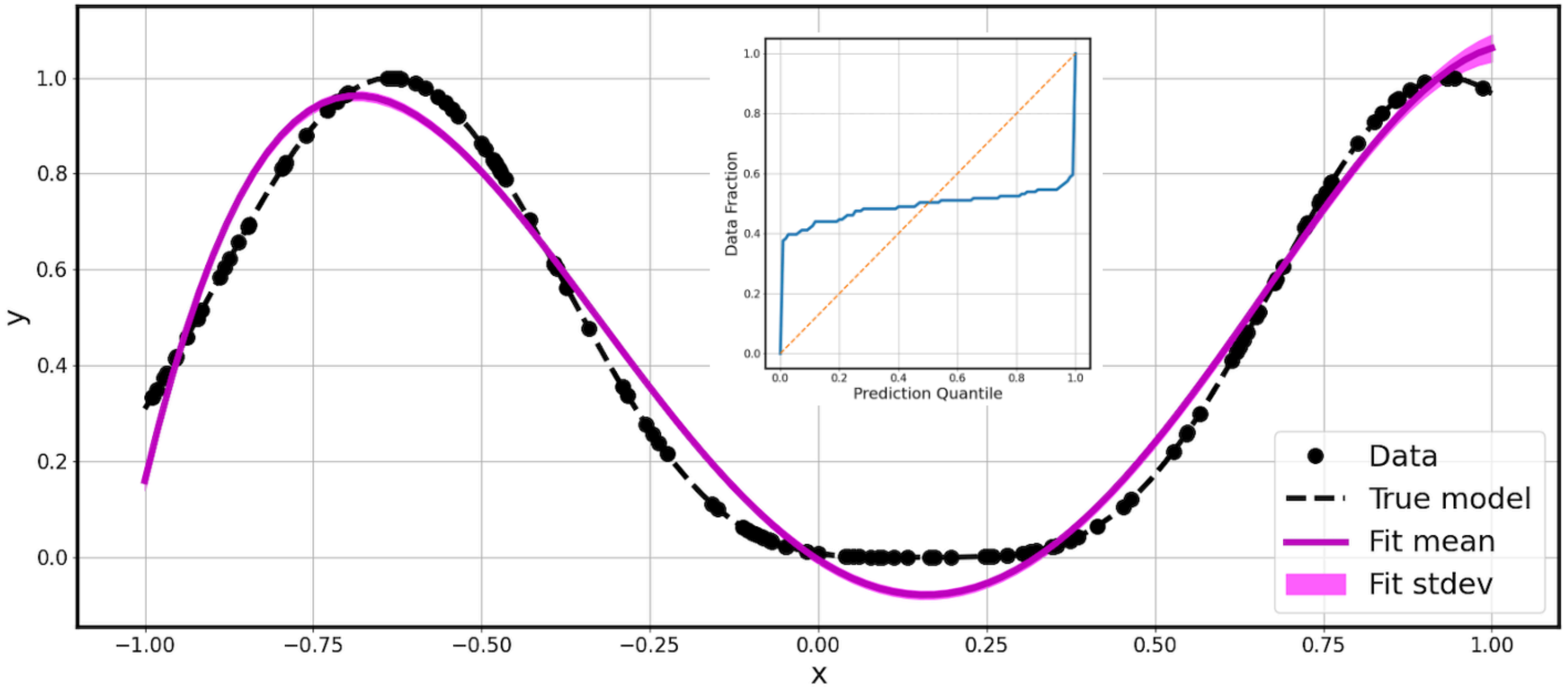
In principle, can construct one big PC : $\sum_k f_{ki}\Psi_k(\xi_1, \xi_2, \ldots, \xi_m)$

with germs $\xi_i$'s corresponding to different uncertainty sources.

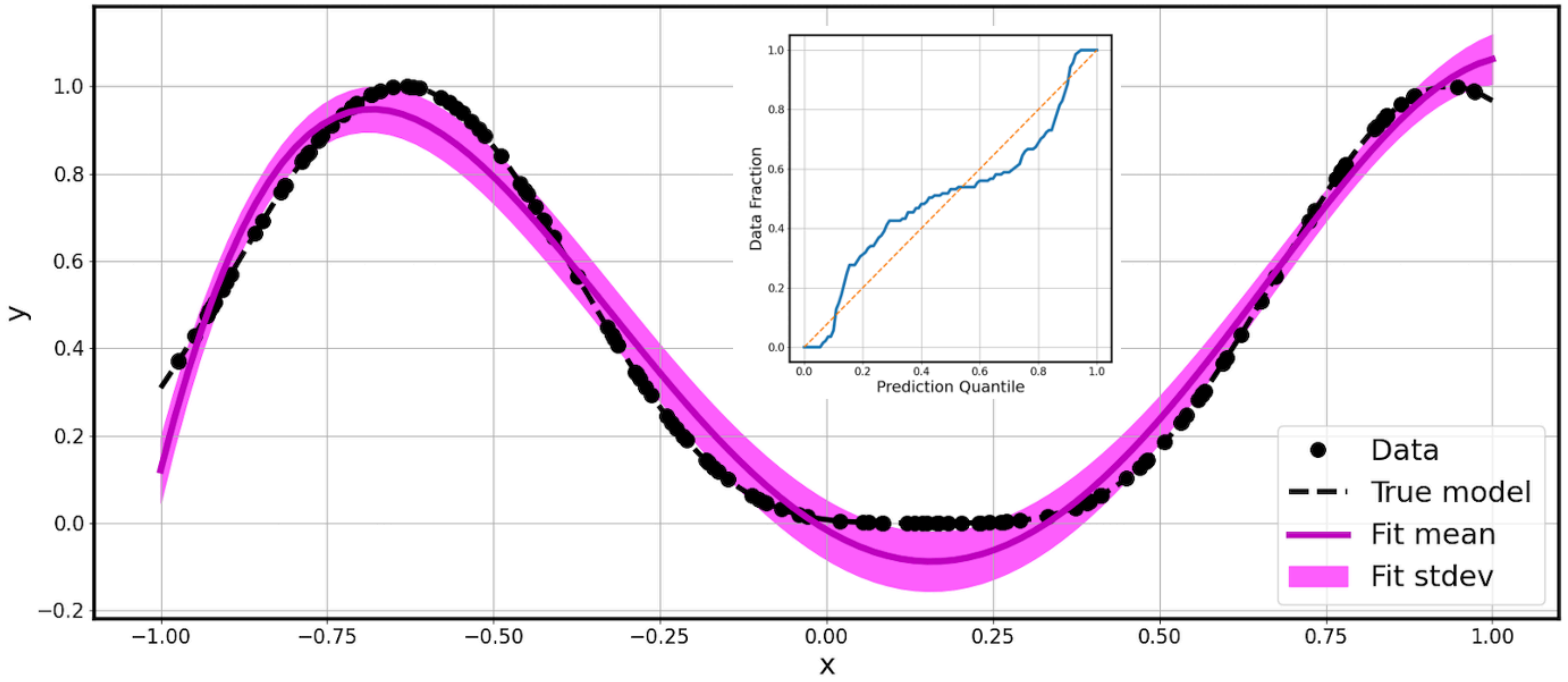


K. Sargsyan (ksargsy@sandia.gov)

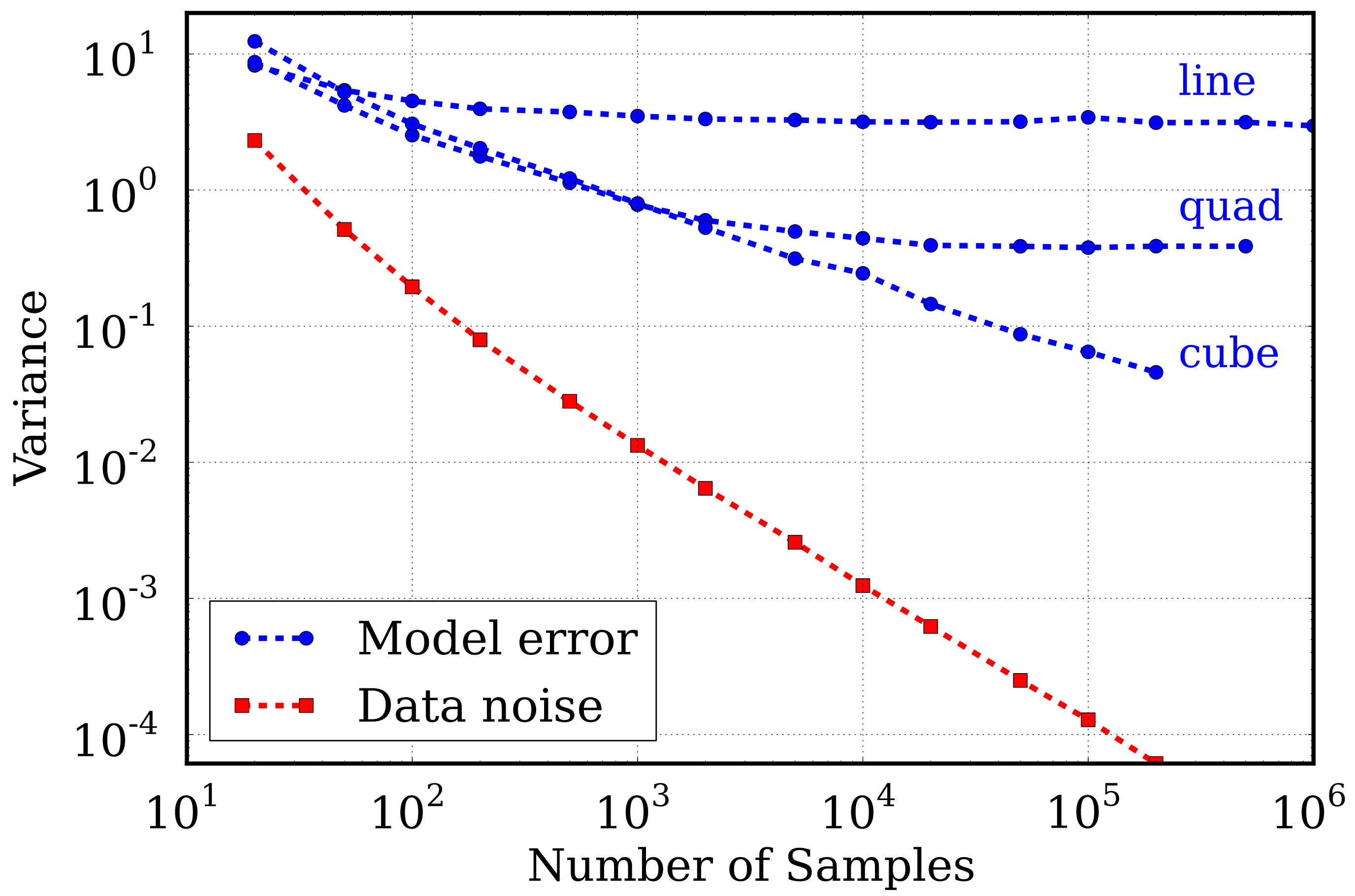
# Inv UQ: Leftover uncertainty due to model error



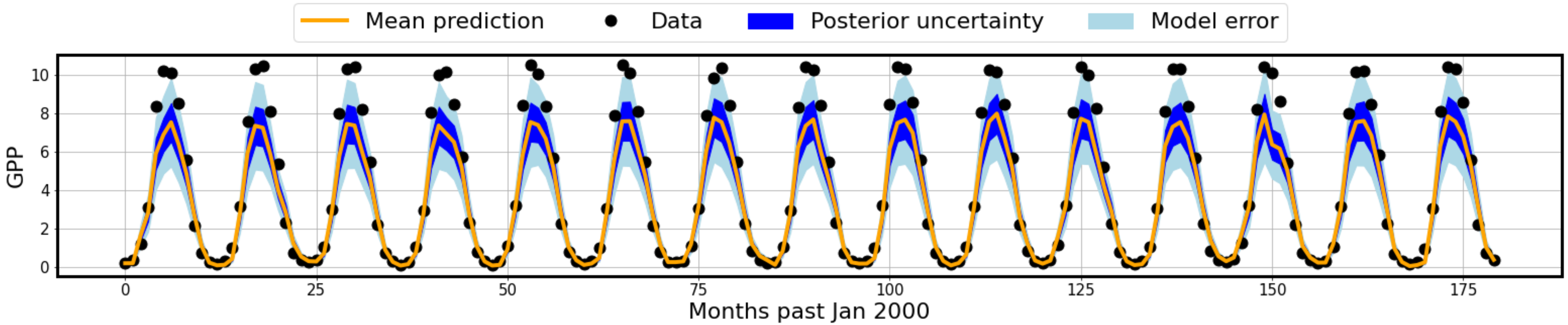

without model error




With model error



# ELM Model Error



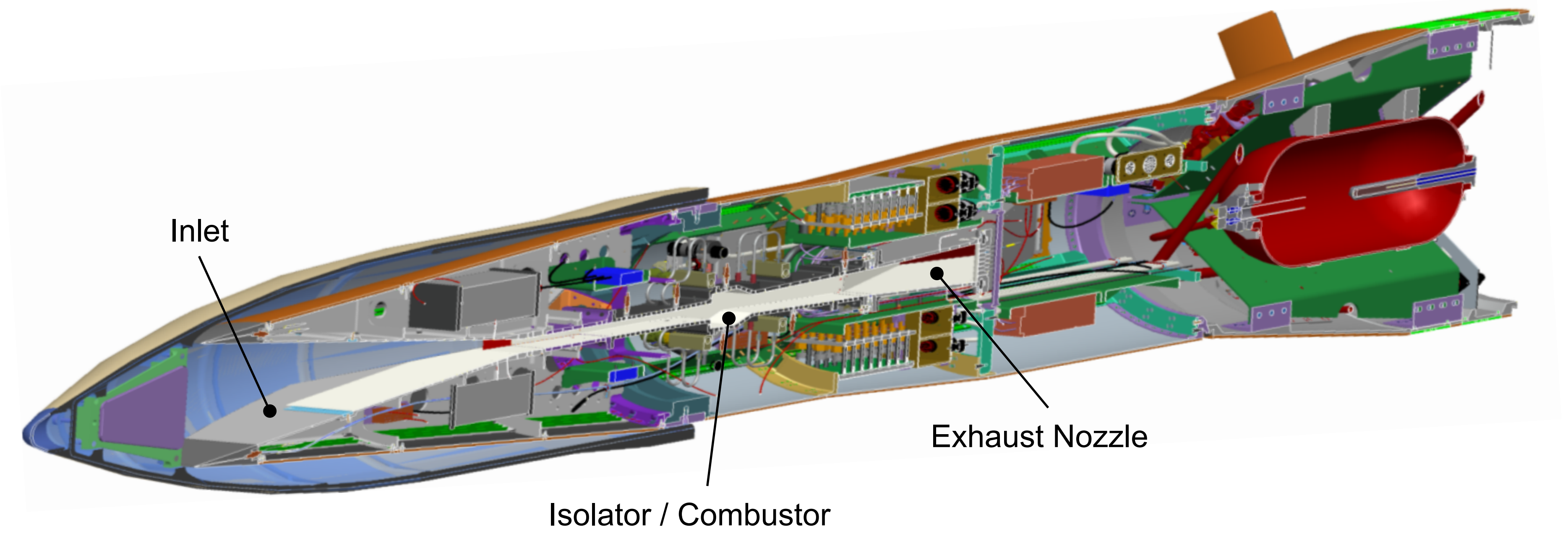
# UQ in a Scramjetapplication

NASALangleyHypersonicInternationalFlightResearchand Experimentation(HIFiRE)directconnectrig(HDCR)

LEScomputationofsupersonicturbulentmultiphasecombustion RAPTOR code by Joe Oefelein (Sandia,Georgia Tech)

GSAandforwardUQ

Resolution: $d_{inj}/\Delta_x = 16$
Number of cells: 66 M
Run time: 31 days on 2432 processors
CPU time for convergence: 1.8 M hours





K. Sargsyan (ksargsy@sandia.gov)

# Randomized MAP Sampling (RMS)

[Pearce, 2020]

- Consider log-posterior: $-\log P(w\,|\,y) = ||y - NN_w(x)||^2 + R(w)$

- Consider regularized training problem $\min\left(\alpha\,||y - NN_w(x)||^2 + \beta\,||w - w*||^2\right)$

- If one samples $w*$ from prior $\sim e^{-R(w)}$, the set of deterministic solutions
  approximately forms the posterior $P(w\,|\,y)$

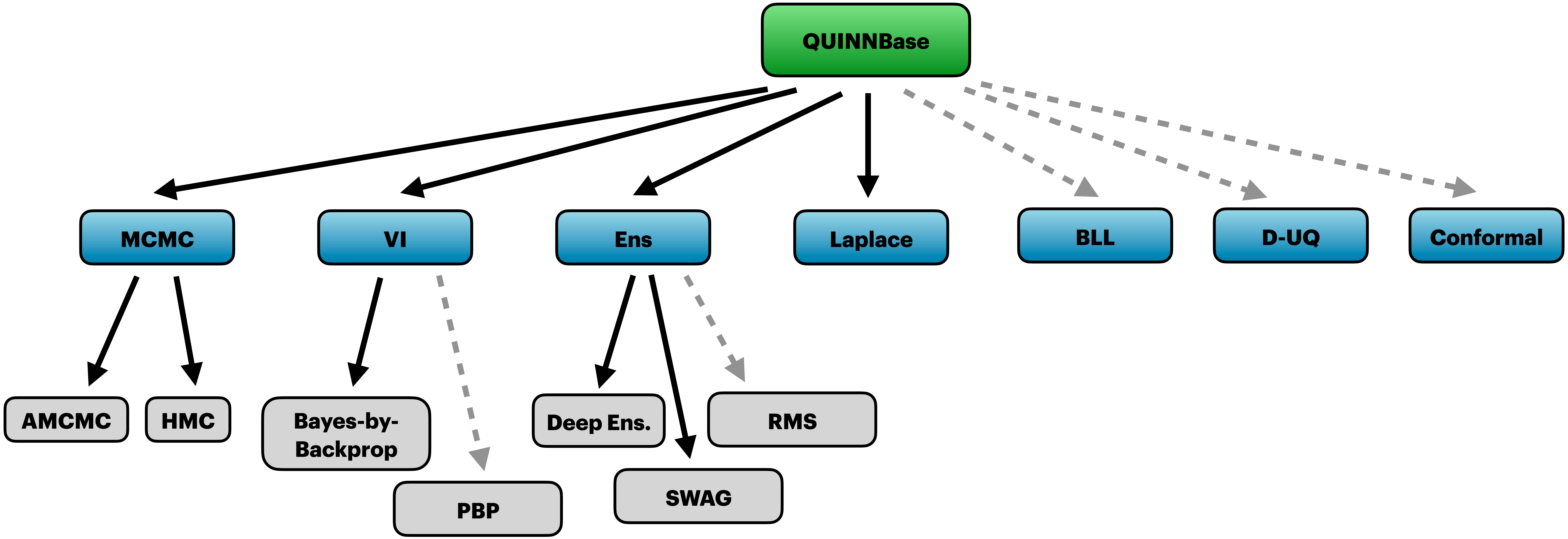
- It is exact for gaussian priors, linear models:

but the authors show that it extends well to larger class, including NNs

- What is missing: proper attribution of uncertainty: is it really
  RMS or the initialization that drives the good results?

# QUiNN (Quantification of Uncertainties in Neural Networks)

```
uqnet = QUINNBase(torch.nn.module)
```




***github.com/sandialabs/quinn***

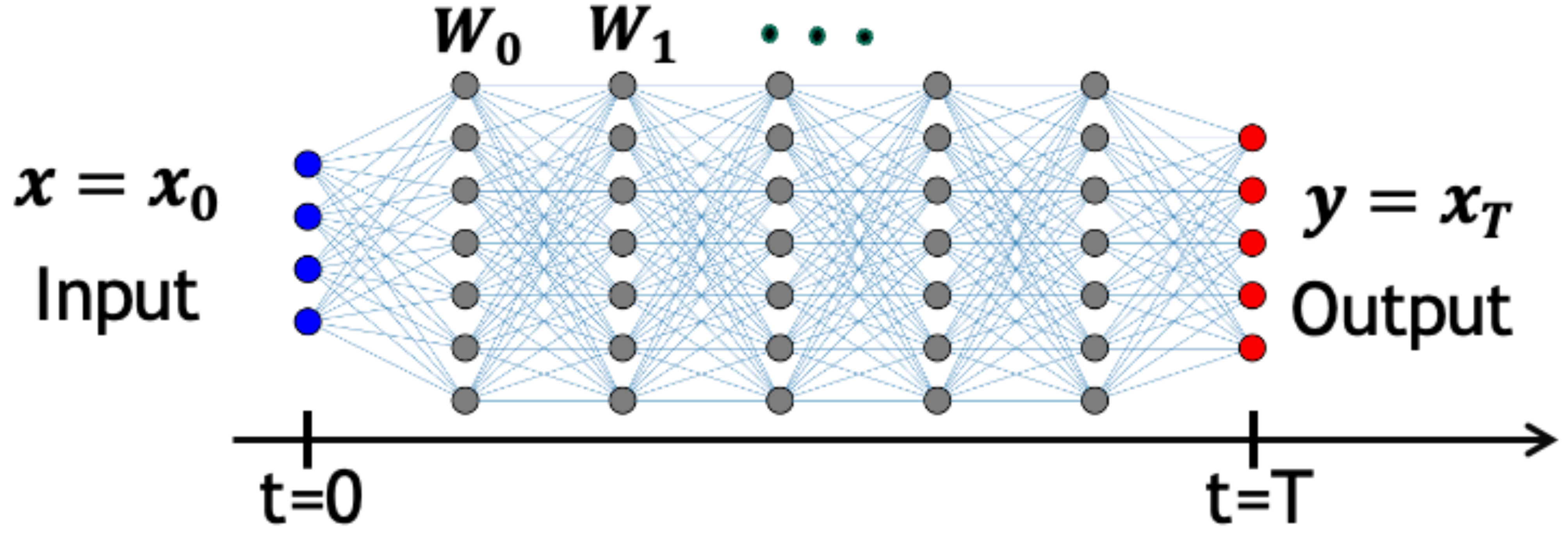


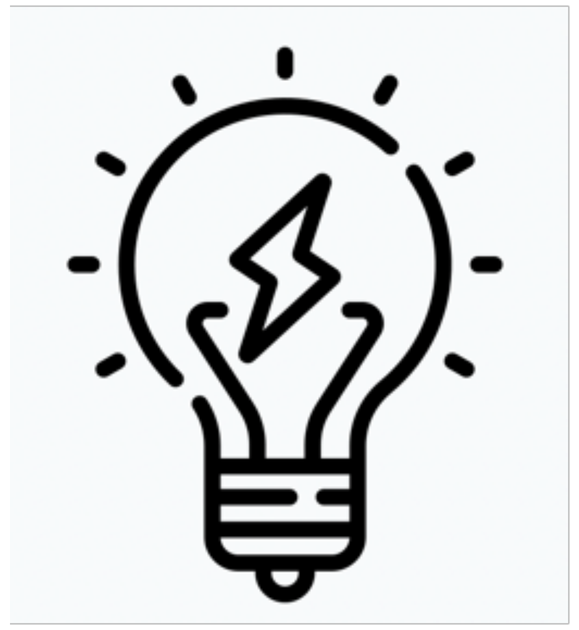
K. Sargsyan (ksargsy@sandia.gov)

# Weight Parameterization inspired by NODE analogy

Neural ODE: $$\frac{dx}{dt} = \sigma(W(t)x + b(t))$$

ResNet: $$x_{n+1} = x_n + \sigma(W_n x_n + b_n)$$

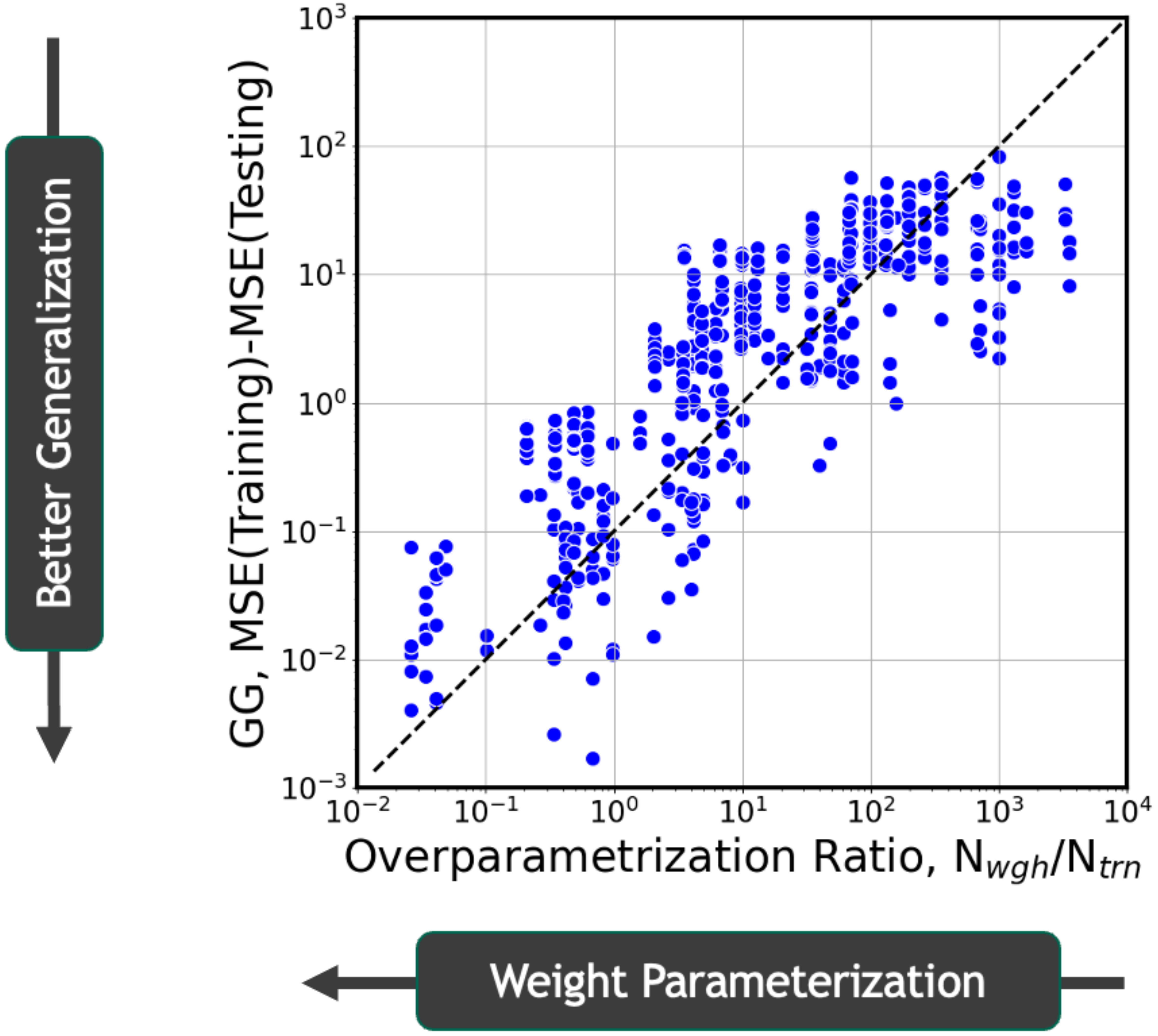


Parameterize weight matrices with respect to time (aka depth)

$W(t;\theta)$ and train for $\theta$'s.

# Weight Parameterization improves generalization



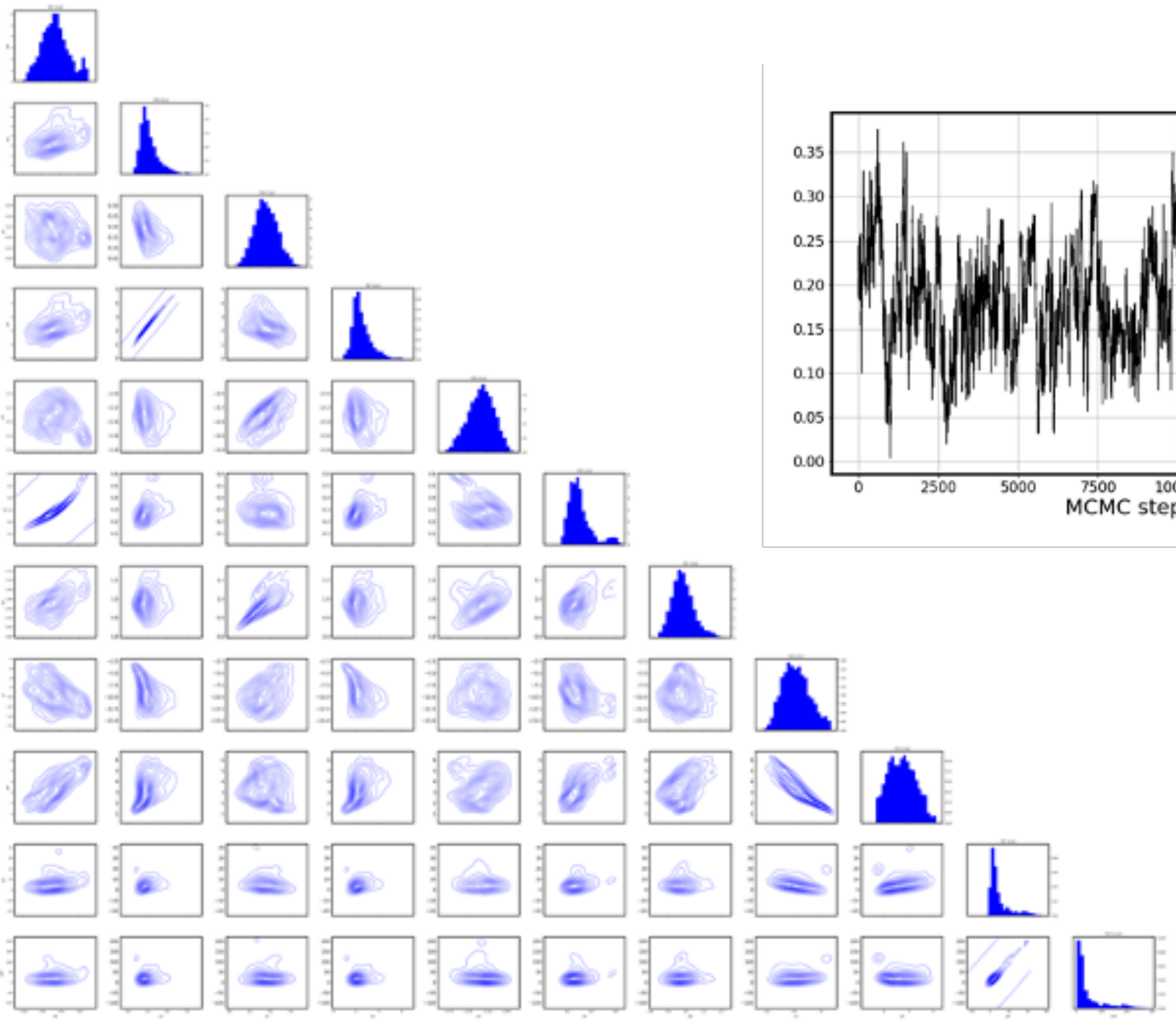

- Generalization Gap correlates with overparameterization

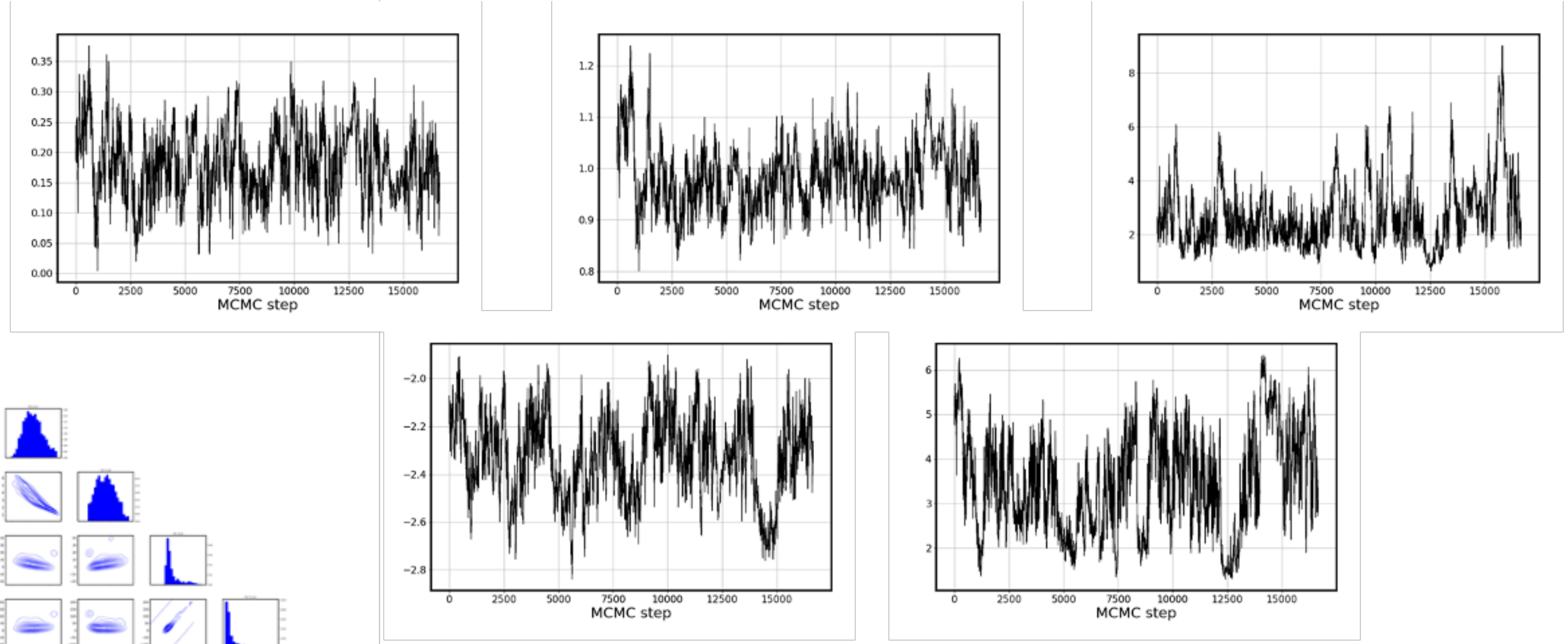
- Weight-parameterized ResNets reduce Generalization Gap

Each dot is a training run with varying weight parameterization functions

# WP ResNet enables UQ

- Number of parameters in ResNets, as well as MLPs, **grows with linearly depth**.
- Number of parameters  in weight-parameterized ResNets is **independent of depth**.
- We can easily achieve regimes with manageable MCMC dimensionality and posterior PDFs that out-of-box MCMC methods can easily sample.





# WP ResNet enables UQ

- Number of parameters in ResNets, as well as MLPs, **grows with linearly depth**.
- Number of parameters in weight-parameterized ResNets is **independent of depth**.
- We can easily achieve regimes with manageable MCMC dimensionality and posterior PDFs that out-of-box MCMC methods can easily sample.