Model Error Representation via Density Estimation

Khachik Sargsyan, Xun Huan, Habib Najm

Sandia National Laboratories, Livermore, CA

ScramjetUQ Kickoff Meeting
Livermore, CA
Feb 17-18, 2016

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

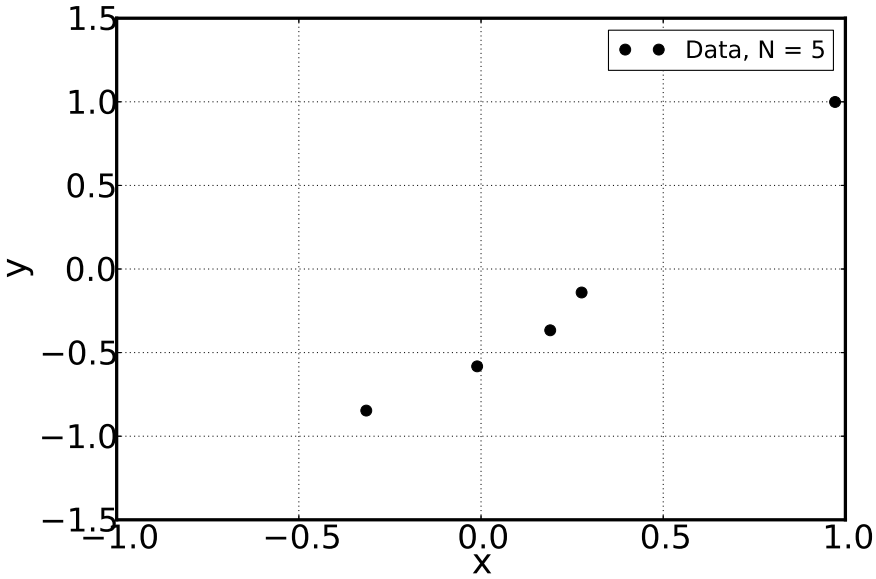
Further acknowledgements:
R. Ghanem(USC), J. Bender(LLNL), Y. Marzouk(MIT), C. Feng(MIT), O. Knio(Duke),
M. Eldred(SNL-NM), C. Safta(SNL-CA), B. Debusschere(SNL-CA),
G. Lacaze(SNL-CA), Z. Vane(SNL-CA), J. Oefelein(SNL-CA).

Main target

Model error = deviation from 'truth', or from a higher-fidelity model

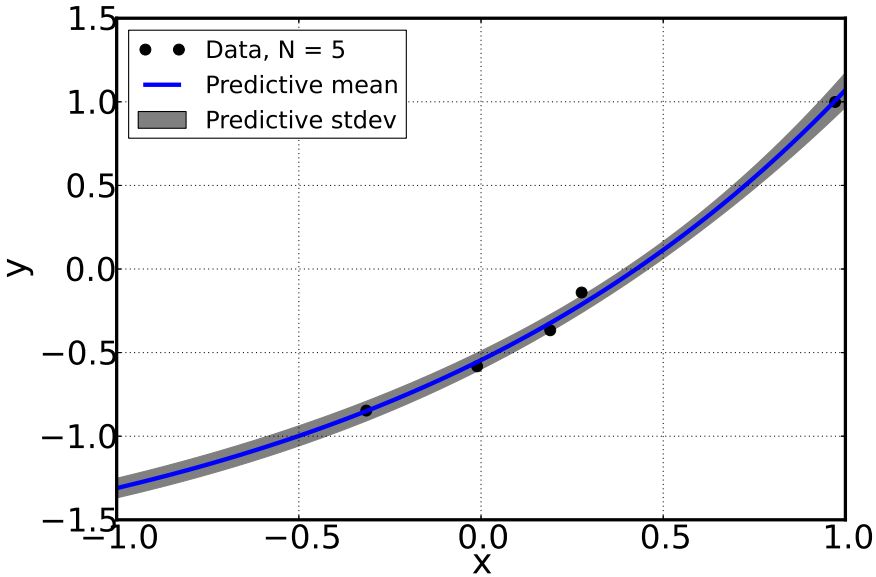
- Represent and estimate the error associated with
  - Simplifying assumptions, parameterizations
  - Mathematical formulation, theoretical framework
  - Numerical discretization
    (connection to Mesh Discretization Error Task)
- ...will be useful for
  - Model validation
  - Model comparison
  - Scientific discovery and model improvement
  - Reliable computational predictions
- Inverse modeling context
  - Given experimental or higher-fidelity model data, estimate the model error

Motivation

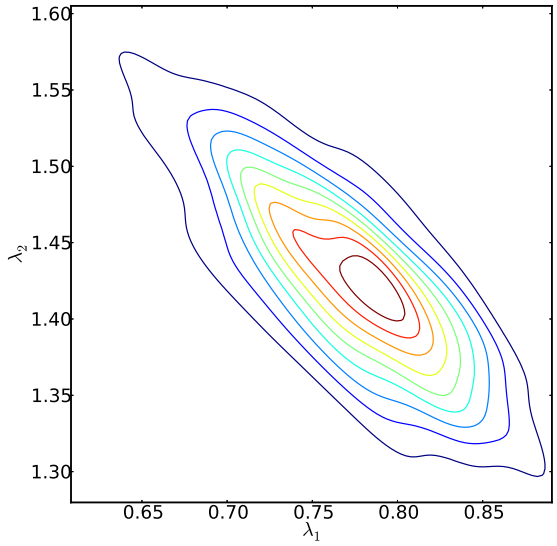


Model-data fit

Motivation

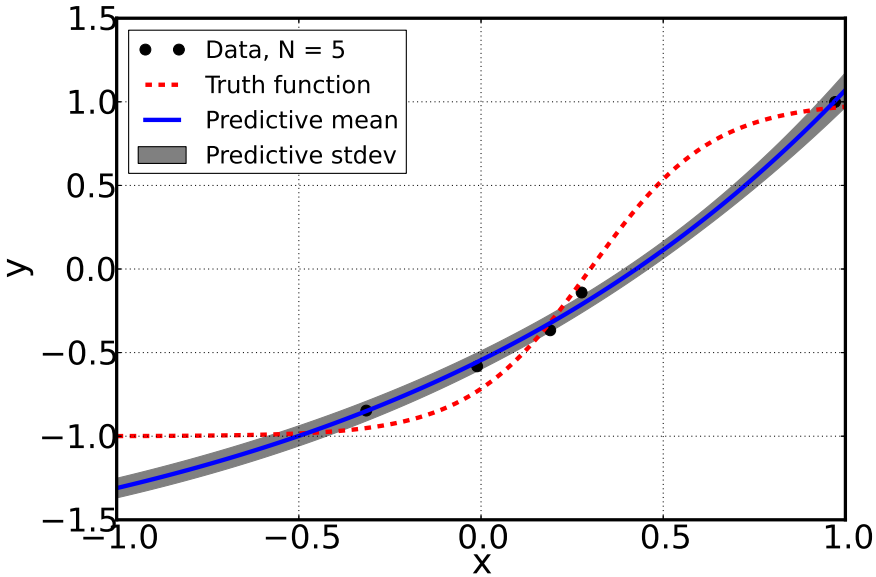


Model-data fit

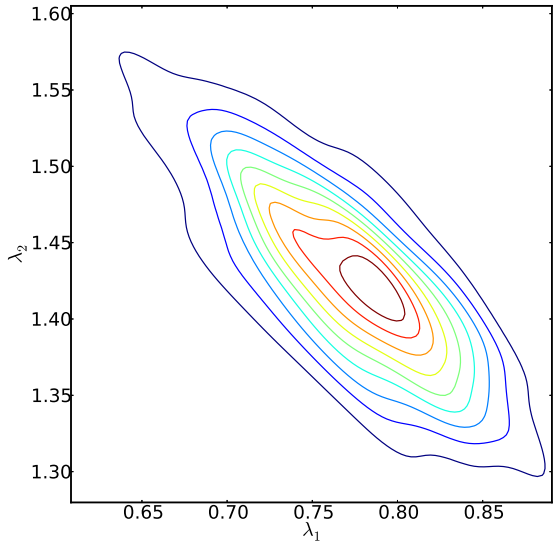


Calibrated parameters

Motivation

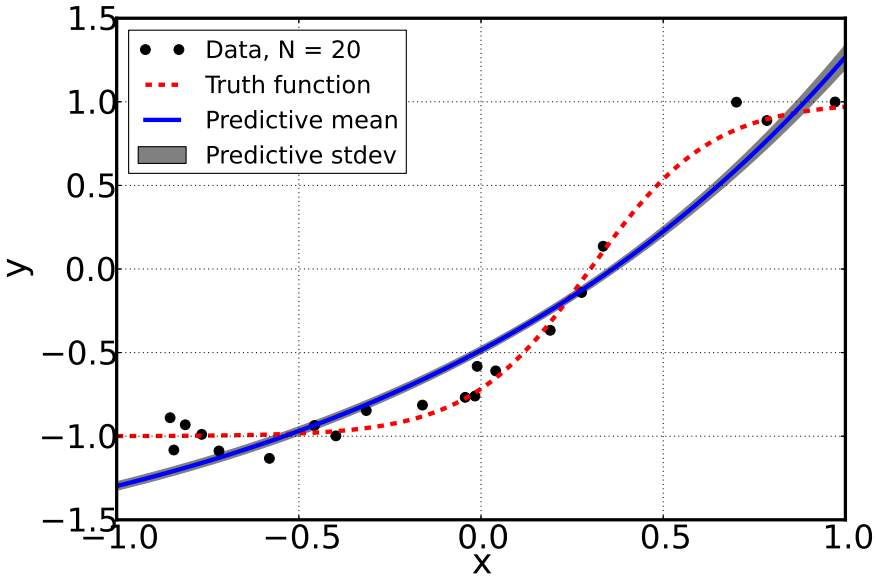


Model-data fit

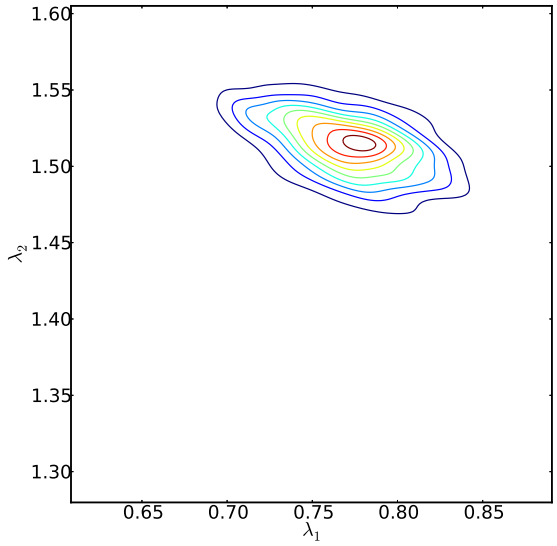


Calibrated parameters

Motivation

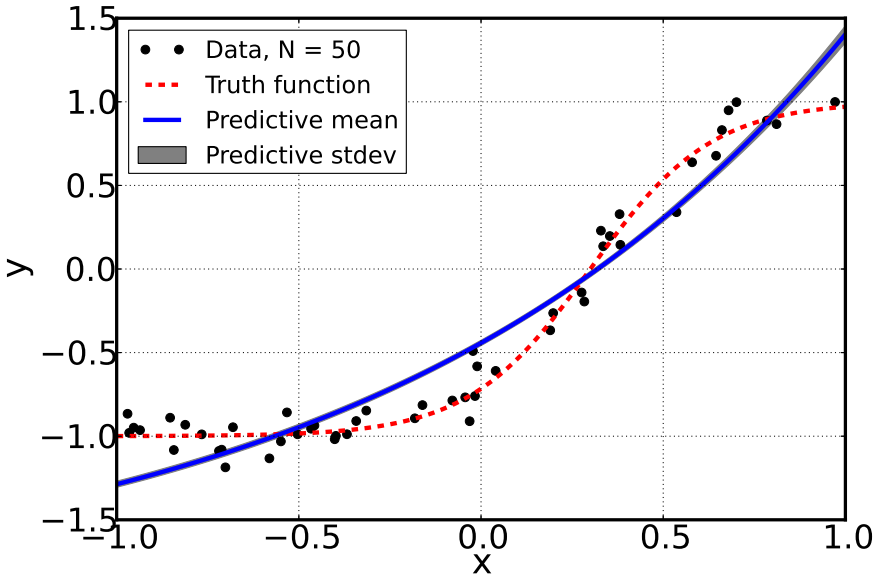


Model-data fit

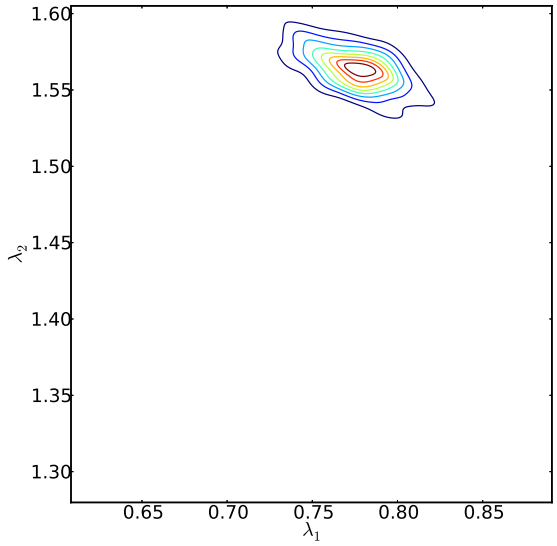


Calibrated parameters

Motivation

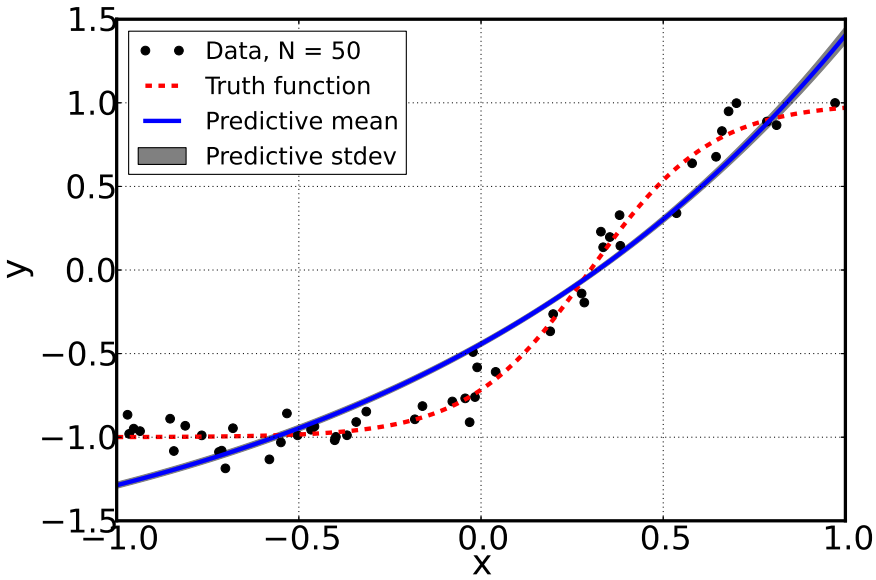


Model-data fit

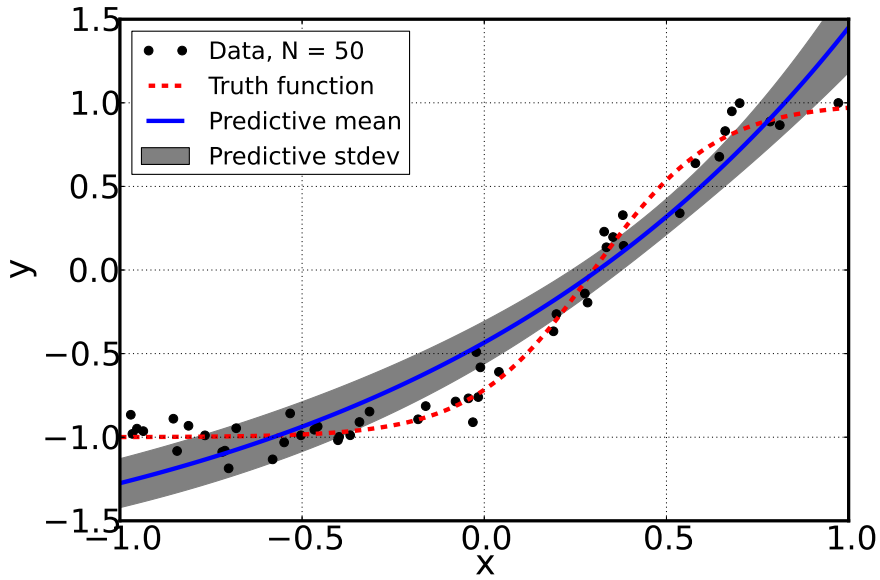


Calibrated parameters

Motivation



Model-data fit



What we want

- If the model is structurally deficient, more data does not help!
- We target model-vs-truth discrepancy

Data-Model-Truth

- Measurements

$$\overset{\text{data}}{y_i} = \overset{\text{truth}}{g(x_i)} + \overset{\text{data error}}{\epsilon_i^{\mathrm{d}}}$$

- Model

$$\overset{\text{truth}}{g(x_i)} = \overset{\text{model}}{f(x_i;\lambda)} + \overset{\text{model error}}{\delta(x_i)}$$

- Total error budget

$$y_i = \underbrace{f(x_i;\lambda) + \delta(x_i)}_{\text{truth } g(x_i)} + \epsilon_i^{\mathrm{d}}$$

Statistical modeling of errors in calibrating $f(x;\lambda)$

Data Error: $\epsilon_i^{\mathrm{d}} \sim \mathrm{N}(0,\sigma^2)$

Model Error: $\delta(x) \sim \mathrm{GP}(\mu(x), C(x,x'))$

Estimate model parameters λ along with those of $\delta(x)$, ϵ_i^{d}

State-of-the-art: Issues for physical models

$$y_i = \underbrace{f(x_i;\lambda) + \delta(x_i)}_{\text{truth}} + \epsilon_i^{\mathrm{d}}$$

- Explicit additive statistical model for model error $\delta(x)$ Kennedy-O'Hagan (2001).
- Calibrated predictive model $\quad y_{mod}(x) = f(x;\lambda) + \delta(x)$
- Potential violation of physical constraints
  - *e.g.* incompressible flow: $\nabla \bullet v = 0$
- Disambiguation of model error $\delta(x_i)$ and data error ϵ_i^{d}
- Calibration of model error on measured observable does not impact the quality of model predictions on other QoIs
- Physical scientists are unlikely to augment their model with a statistical model error term on select outputs

Model error embedding: key idea

- Ideally, modelers want predictive *errorbars*:
  inserting randomness on the outputs has issues, so...
- Cast input parameters λ as a random variable Λ

$$y_i = f(x_i; \Lambda) + \epsilon_i^{\mathrm{d}}$$

- Embed model error in specific submodel phenomenology
  - a modified transport or constitutive law
  - a modified formulation for a material property
- Allows placement of model error term in locations where key modeling assumptions and approximations are made
  - as a correction or high-order term
  - as a possible alternate phenomenology
- Naturally preserves model structure and physical constraints

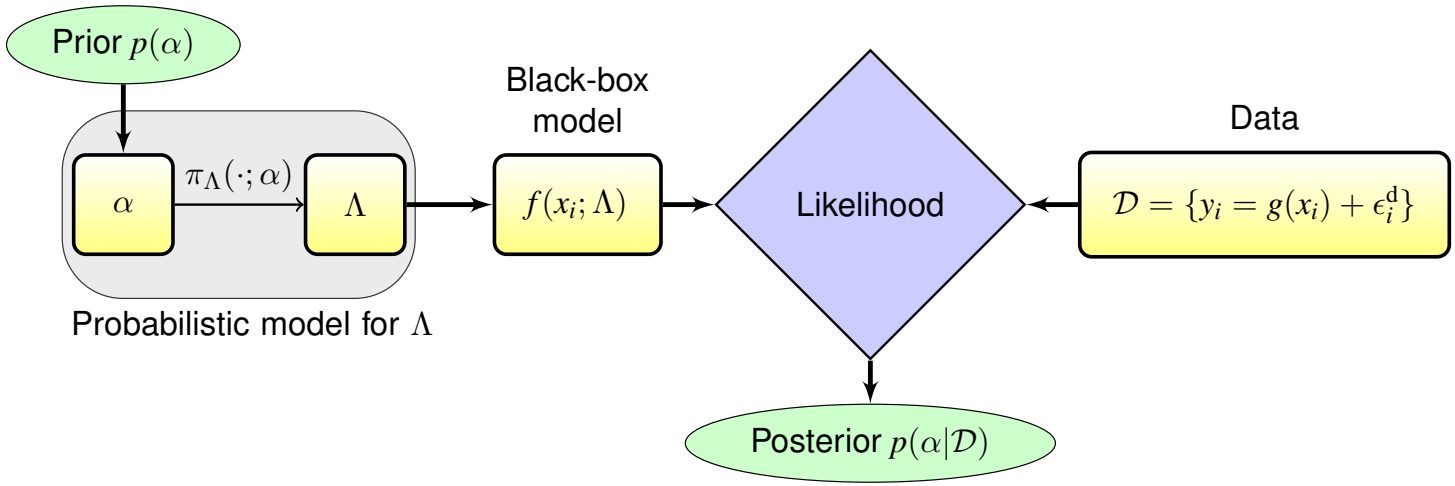
Model error embedding: Bayesian formulation

- In the simplest setting, cast λ as a random variable Λ

$$y_i = f(x_i; \Lambda) + \epsilon_i^{\mathrm{d}}$$

- Calibration turns into density estimation for the PDF of Λ
- Polynomial Chaos parameterization for $\Lambda = \sum_{k=0}^{K} \alpha_k \Psi_k(\xi)$
- Back to parameter estimation, now for $\alpha = (\alpha_0, \ldots, \alpha_K)$

$$\underbrace{p(\alpha|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{L_{\mathcal{D}}(\alpha)}_{\text{Likelihood}} \underbrace{p(\alpha)}_{\text{Prior}}$$



K. Sargsyan, H. Najm, and R. Ghanem, "On the Statistical Calibration of Physical Models". *International Journal for Chemical Kinetics*, 47(4): pp 246–276, 2015.

Likelihood construction: data model

- Data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
- Model $f(x;\lambda)$

- Model parameters $\Lambda = \lambda(\alpha;\xi) = \sum_k \alpha_k \Psi_k(\xi_1, \ldots, \xi_d)$
- Data noise $\epsilon_i^{\mathrm{d}} = \sigma_{\mathcal{D}} \xi_{d+i}$; Full PC germ $\hat{\xi} = (\underbrace{\xi_1, \ldots, \xi_d}_{\text{Model error}}, \underbrace{\xi_{d+1}, \ldots, \xi_{d+N}}_{\text{Data noise}})$
- Infer $\hat{\alpha} = (\alpha, \sigma_{\mathcal{D}})$
- Data generation model; to aid likelihood $p(\mathcal{D}|\hat{\alpha})$ construction

$$
\begin{aligned}
y_i &= f(x_i, \Lambda) + \epsilon_i^{\mathrm{d}} = \\
&= f\left(x_i, \sum_k \alpha_k \Psi_k(\xi_1, \ldots, \xi_d)\right) + \sigma_{\mathcal{D}} \xi_{d+i} = \\
&\overset{NISP}{\approx} \sum_k f_{ik}(\alpha) \Psi_k(\xi_1, \ldots, \xi_d) + \sigma_{\mathcal{D}} \xi_{d+i} = \\
&= h_i(\hat{\xi}; \hat{\alpha})
\end{aligned}
$$

Likelihood options

$$y_i = \underbrace{\sum_k f_{ik}(\alpha)\Psi_k(\xi_1, \ldots, \xi_d) + \sigma_{\mathcal{D}}\xi_{d+i}}_{h_i(\hat{\xi};\hat{\alpha})}$$

Note: for each $\hat{\alpha}$,

the data model $\boldsymbol{h}(\hat{\xi};\hat{\alpha})$ is a multivariate random variable with cheap sampling, and easily accessible mean $\boldsymbol{\mu}(\hat{\alpha})$ and covariance $\Sigma(\hat{\alpha})$

Likelihood options

$$y_i = \underbrace{\sum_k f_{ik}(\alpha)\Psi_k(\xi_1,\ldots,\xi_d) + \sigma_{\mathcal{D}}\xi_{d+i}}_{h_i(\hat{\xi};\hat{\alpha})}$$

Note: for each $\hat{\alpha}$,

the data model $\boldsymbol{h}(\hat{\xi};\hat{\alpha})$ is a multivariate random variable with cheap sampling, and easily accessible mean $\boldsymbol{\mu}(\hat{\alpha})$ and covariance $\Sigma(\hat{\alpha})$

- Full Likelihood: $L(\hat{\alpha}) = p(\mathcal{D}|\hat{\alpha}) = p(y_1,\ldots,y_N|\hat{\alpha}) = \pi_{\boldsymbol{h}}(\boldsymbol{y})$
  - Degenerate if no data noise
  - Requires multivariate KDE
  - Gaussian approximation:
    $L(\hat{\alpha}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}(\hat{\alpha}))^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\hat{\alpha}))\right)$

Likelihood options

$$y_i = \underbrace{\sum_k f_{ik}(\alpha)\Psi_k(\xi_1, \ldots, \xi_d) + \sigma_{\mathcal{D}}\xi_{d+i}}_{h_i(\hat{\xi};\hat{\alpha})}$$

Note: for each $\hat{\alpha}$,

the data model $\boldsymbol{h}(\hat{\xi};\hat{\alpha})$ is a multivariate random variable with cheap sampling, and easily accessible mean $\boldsymbol{\mu}(\hat{\alpha})$ and covariance $\Sigma(\hat{\alpha})$

- Marginalized Likelihood:
  $L(\hat{\alpha}) = p(D|\hat{\alpha}) = \prod_{i=1}^{N} p(y_i|\hat{\alpha}) = \prod_{i=1}^{N} \pi_{h_i}(y_i)$
  - Requires univariate KDE
  - Neglects built-in correlations
  - Gaussian approximation: $L(\hat{\alpha}) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{N} \Sigma_{ii}^{-2}(y_i - \mu_i(\hat{\alpha}))^2\right)$

Likelihood options

$$y_i = \underbrace{\sum_k f_{ik}(\alpha)\Psi_k(\xi_1,\ldots,\xi_d) + \sigma_{\mathcal{D}}\xi_{d+i}}_{h_i(\hat{\xi};\hat{\alpha})}$$

Note: for each $\hat{\alpha}$,

the data model $\boldsymbol{h}(\hat{\xi};\hat{\alpha})$ is a multivariate random variable with cheap sampling, and easily accessible mean $\boldsymbol{\mu}(\hat{\alpha})$ and covariance $\Sigma(\hat{\alpha})$

- Approximate Bayesian Computation (ABC): $L(\hat{\alpha}) = \frac{1}{\epsilon}K\left(\frac{\rho(\mathcal{S}_{\mathcal{M}},\mathcal{S}_{\mathcal{D}})}{\epsilon}\right)$
  - $p(y|\mathcal{D})$ is "centered" on the data
  - The width of the distribution $p(y|\mathcal{D})$ is consistent with the spread of the data around the nominal model prediction

$$L(\hat{\alpha}) \propto \exp\left(-\frac{1}{2\epsilon^2}\sum_{i=1}^{N}\left[(\mu_i(\hat{\alpha})-y_i)^2 + (\sqrt{\Sigma_{ii}(\hat{\alpha})}-\gamma|\mu_i(\hat{\alpha})-y_i|)^2\right]\right)$$

Calibrated predictions – general x

For fixed α, e.g. Maximum a Posteriori (MAP) value,

Uncertain prediction:

$$y = f(x, \lambda(\alpha; \xi))$$

Mean:

$$\mu(x, \alpha) = \mathbb{E}_\xi[f(x, \lambda(\alpha; \xi))]$$

Variance:

$$\sigma^2(x, \alpha) = \mathbb{V}_\xi[f(x, \lambda(\alpha; \xi))]$$

Average over posterior of α

Posterior predictive (PP):

$$p(f|\mathcal{D}) = \int p(f|\alpha)p(\alpha|\mathcal{D})d\alpha = \mathbb{E}_\alpha[p(f|\alpha)]$$

PP mean:

$$\mu_{\text{PP}}(x) = \mathbb{E}_\xi\mathbb{E}_\alpha[f(x, \lambda(\alpha; \xi))] = \mathbb{E}_\alpha\mu(x, \alpha)$$

PP variance:

$$\sigma^2_{\text{PP}}(x) = \underbrace{\mathbb{E}_\alpha[\sigma^2(x, \alpha)]}_{\text{model error}} + \underbrace{\mathbb{V}_\alpha[\mu(x, \alpha)]}_{\text{data noise}}$$

Calibrated predictions – compare to data at x_i

For fixed α, e.g. Maximum a Posteriori (MAP) value,
Uncertain prediction:

$$y_i = f(x_i, \lambda(\alpha; \xi)) + \epsilon_i^{\mathrm{d}} \equiv h(x_i, \lambda(\alpha; \xi))$$

Mean:

$$\mu(x_i, \alpha) = \mathbb{E}_\xi[f(x_i, \lambda(\alpha; \xi))]$$

Variance:

$$\sigma^2(x_i, \alpha) = \mathbb{V}_\xi[f(x_i, \lambda(\alpha; \xi))] + \sigma_{\mathcal{D}}^2$$

Average over posterior of α

Posterior predictive (PP):

$$p(h|\mathcal{D}) = \int p(h|\alpha)p(\alpha|\mathcal{D})d\alpha = \mathbb{E}_\alpha[p(h|\alpha)]$$

PP mean:

$$\mu_{\mathrm{PP}}(x_i) = \mathbb{E}_\xi \mathbb{E}_\alpha[f(x_i, \lambda(\alpha; \xi))] = \mathbb{E}_\alpha \mu(x_i, \alpha)$$

PP variance:

$$\sigma_{\mathrm{PP}}^2(x_i) = \underbrace{\mathbb{E}_\alpha[\sigma^2(x_i, \alpha)]}_{\text{model error}} + \underbrace{\mathbb{V}_\alpha[\mu(x_i, \alpha)] + \mathbb{E}_{\sigma_{\mathcal{D}}}[\sigma_{\mathcal{D}}^2]}_{\text{data noise}}$$

Attribution of error components

$$y_i = \underbrace{\sum_k f_{ik}(\alpha)\Psi_k(\xi_1,\ldots,\xi_d) + \sigma_{\mathcal{D}}\xi_{d+i}}_{h_i(\hat{\xi};\hat{\alpha})}$$

Stochastic dimensions:

- Model error $\xi_1,\ldots,\xi_d$
- Measurement error $\xi_{d+1},\ldots,\xi_{d+N}$
- Posterior uncertainty (α): can be represented via its own PC expansion (using MCMC samples and Rosenblatt transformation)

Full PC expansion: $y_i = \sum f_j \Psi_j(\hat{\hat{\xi}})$

Full stochastic *germ*:

$$\hat{\hat{\xi}} = (\underbrace{\xi_1,\ldots,\xi_d}_{\text{Model error}}, \underbrace{\xi_{d+1},\ldots,\xi_{d+N}}_{\text{Measurment error}}, \underbrace{\xi_{d+N+1},\ldots,\xi_{d+N+N_\alpha}}_{\text{Posterior uncertainty}})$$

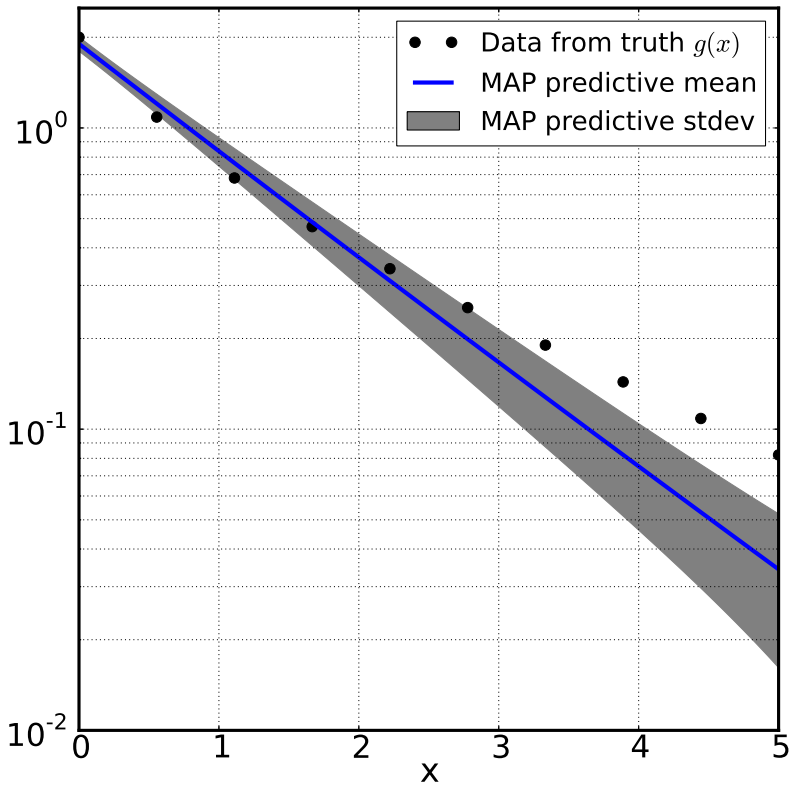
Posterior predictive variance:

$$\sigma^2_{\text{PP}}(x_i) = \mathbb{E}_\alpha[\sigma^2(x_i,\alpha)] + \mathbb{E}_{\sigma_{\mathcal{D}}}[\sigma^2_{\mathcal{D}}] + \mathbb{V}_\alpha[\mu(x_i,\alpha)]$$

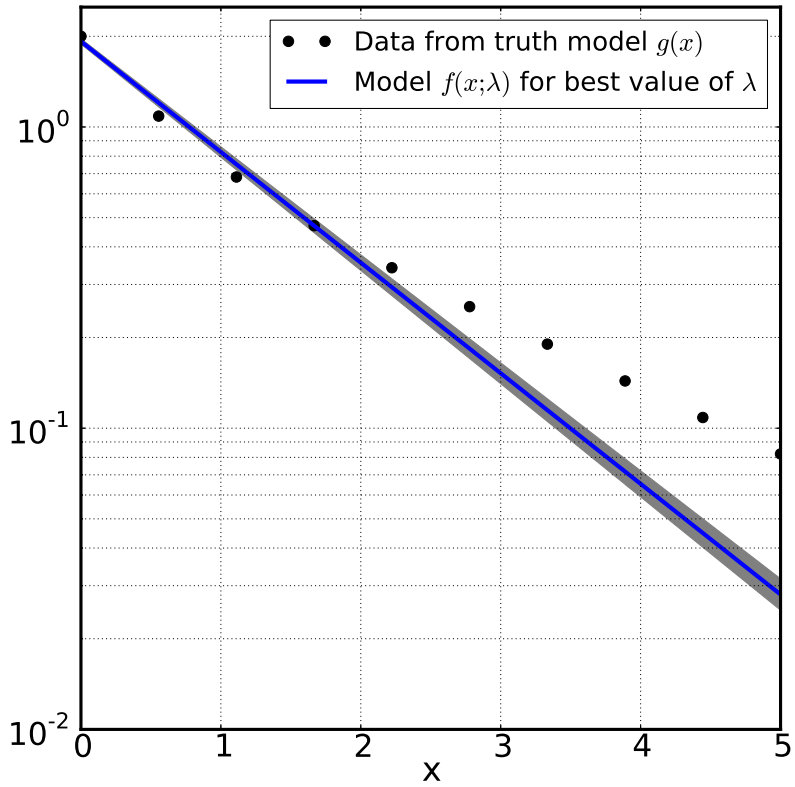
Predictions account for model error

Calibrating single-exponential models
with data from a double exponential model $g(x) = e^{-0.5x} + e^{-2x}$

Linear-exponential $f(x, \lambda) = e^{\lambda_1 + \lambda_2 x}$



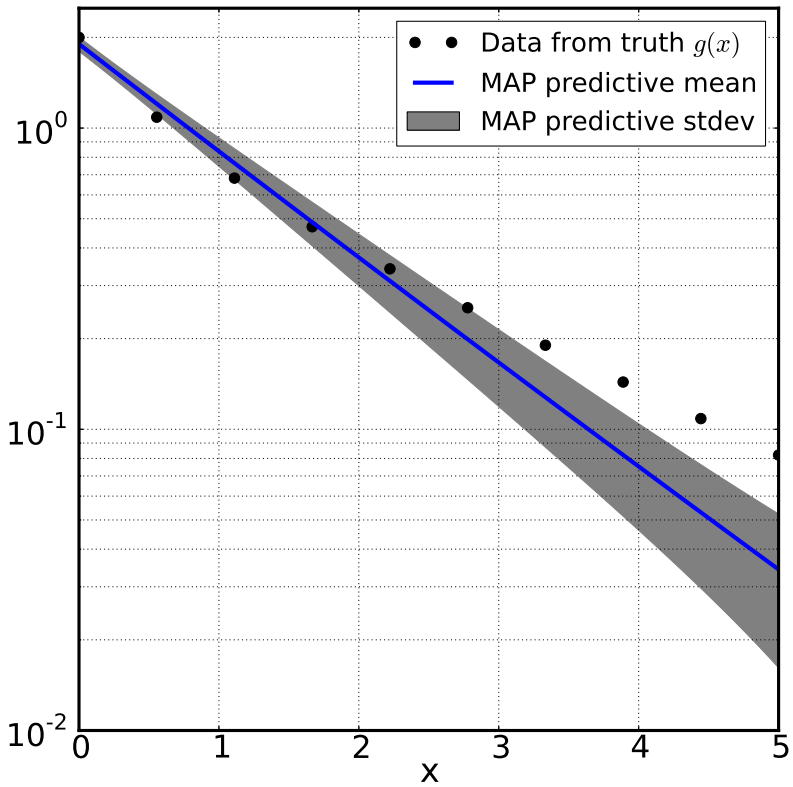
Additive Gaussian error



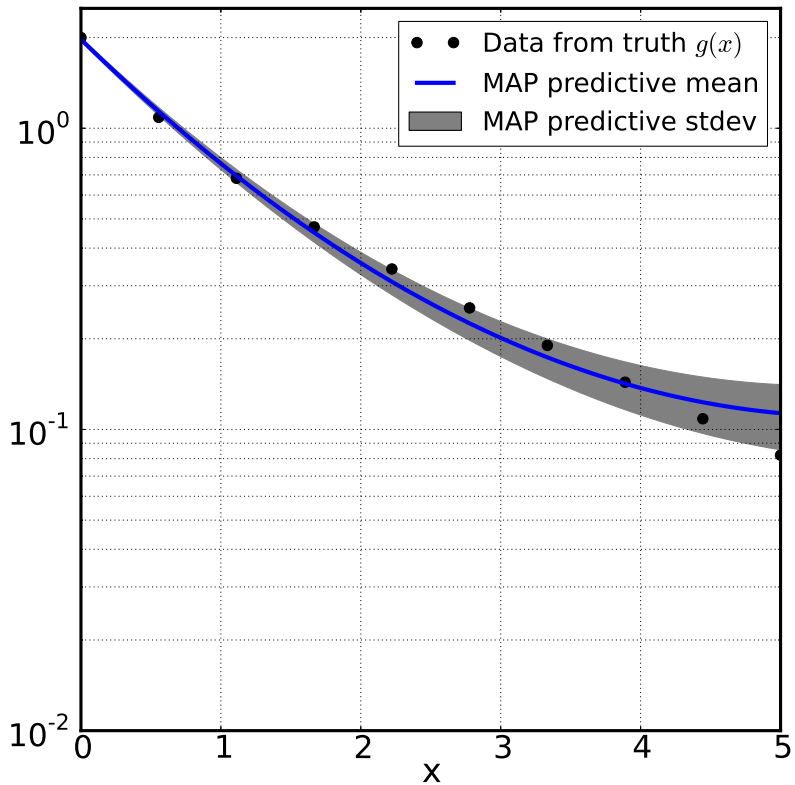
Predictions account for model error

Calibrating single-exponential models
with data from a double exponential model $g(x) = e^{-0.5x} + e^{-2x}$

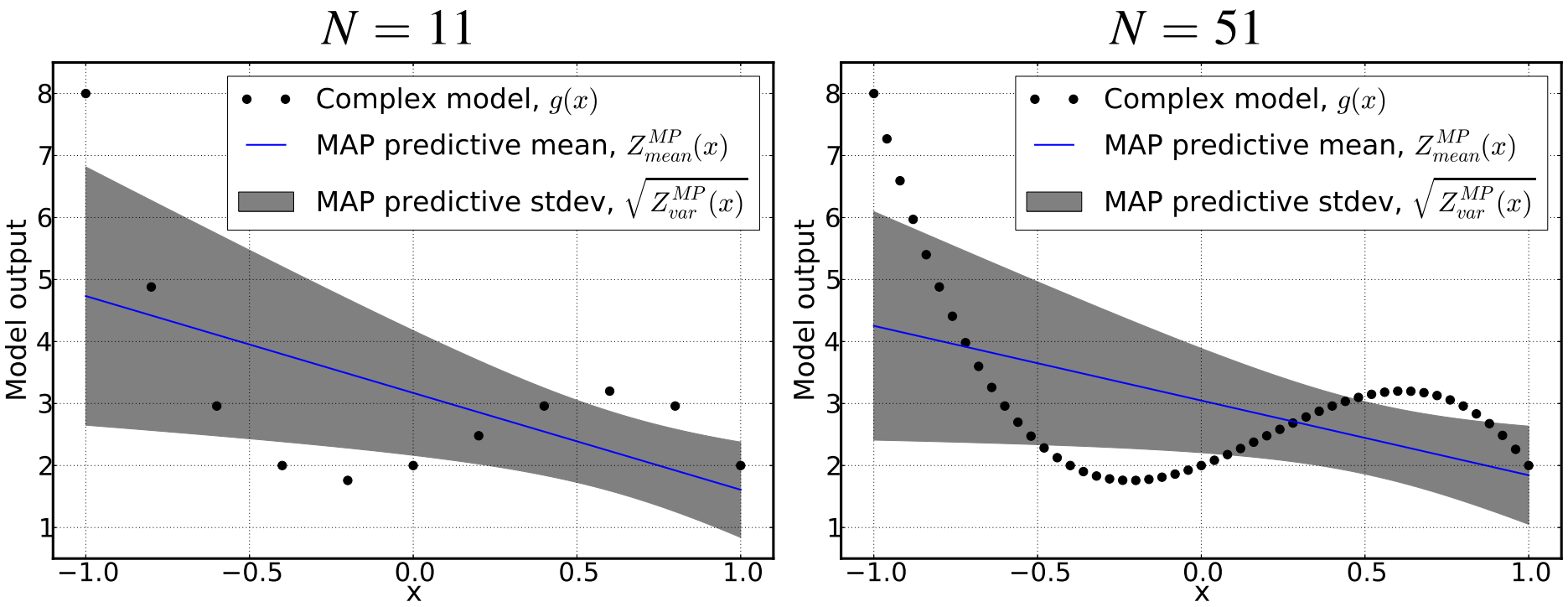
Linear-exponential $f(x, \lambda) = e^{\lambda_1 + \lambda_2 x}$



Quadratic-exponential $f_2(x, \lambda) = e^{\lambda_1 + \lambda_2 x + \lambda_3 x^2}$

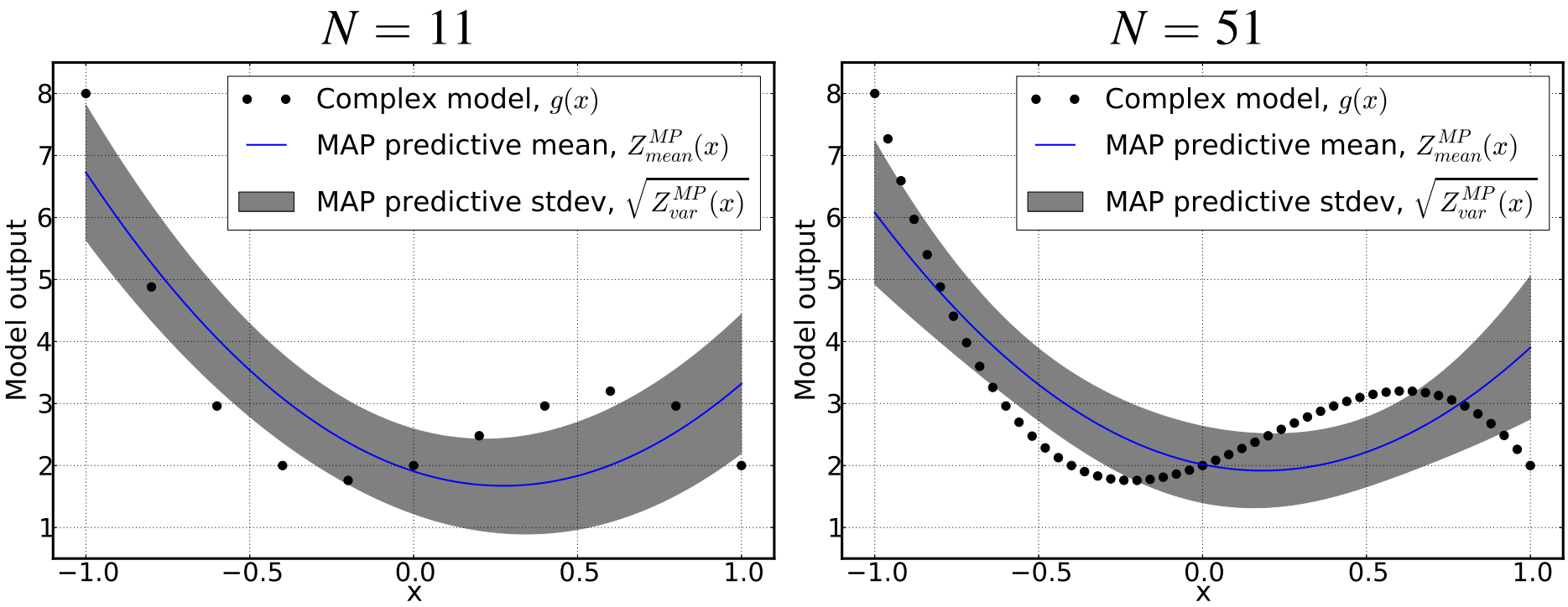


Test problem – Cubic data fit by a line – ABC



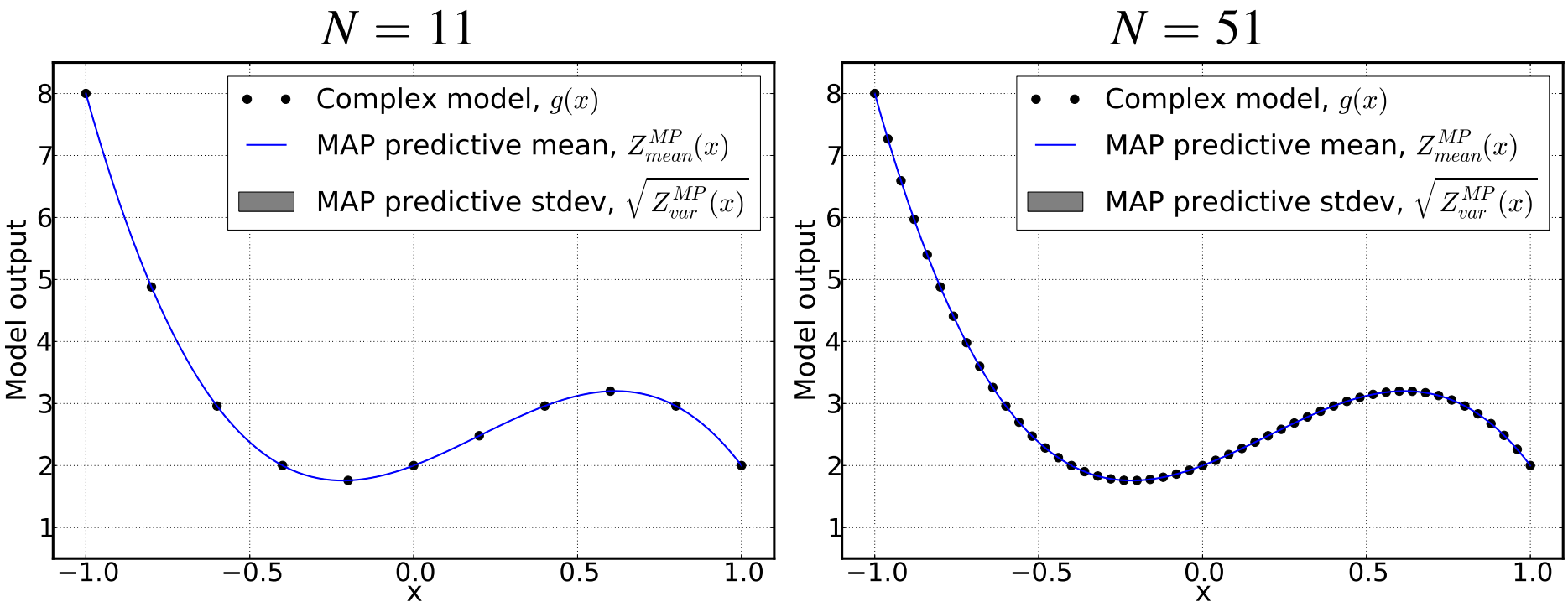
- MAP predictive mean centered on data
- MAP predictive standard deviation captures range of discrepancy
- Increasing number of data points has a small effect on both predictive mean and stdev.

Test problem – Cubic data fit by a quadratic – ABC



- MAP predictive mean centered on data
- MAP predictive standard deviation captures range of discrepancy
- Increasing number of data points has a small effect on both predictive mean and stdev.

Test problem – Cubic data fit by a cubic – ABC

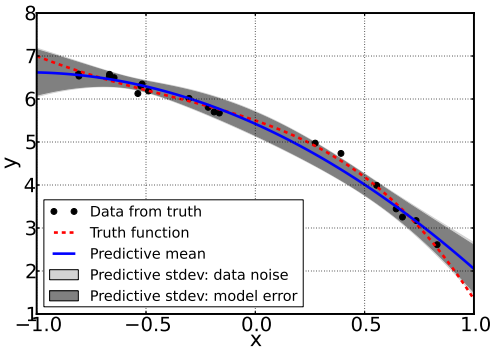


- MAP predictive mean centered on data
- MAP predictive standard deviation captures range of discrepancy
- Increasing number of data points has a small effect on both predictive mean and stdev.

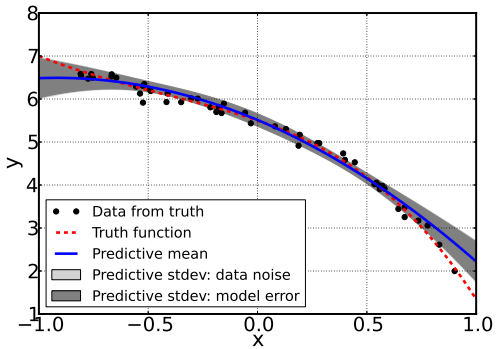
More data leads to 'leftover' model error

Calibrating a quadratic $f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2$
w.r.t. 'truth' $g(x) = 6 + x^2 - 0.5(x+1)^{3.5}$ measured with noise $\sigma = 0.1$.

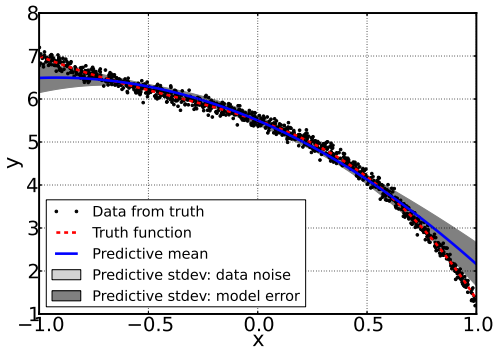
$N = 20$



$N = 50$

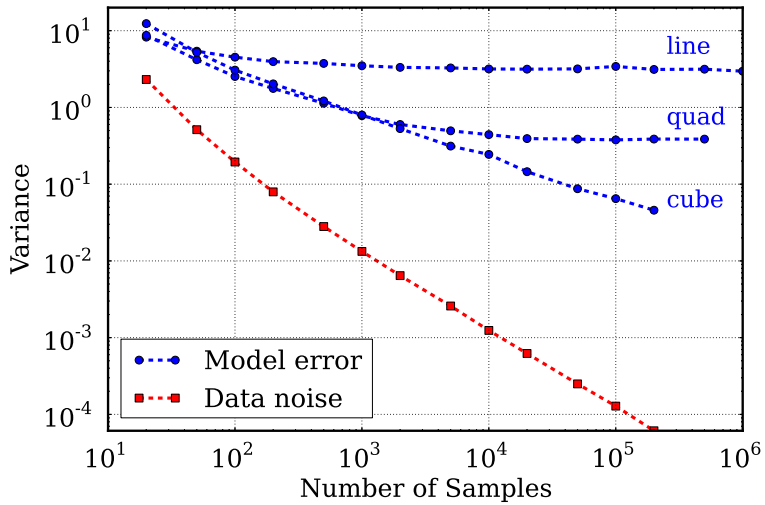


$N = 1000$



Summary of features:

- Well-defined model-to-model calibration
- Model-driven discrepancy correlations
- Respects physical constraints
- Disambiguates model and data errors
- Calibrated predictions of multiple QoIs



ScramjetUQ plan

- FY16
  [M5.1: Demo ME estimation and propagation for P1]
  - Identify a set of low fidelity models [Oct-Jan]
  - Embed ME in a few parameters at a time [Jan-Mar]
  - Calibrate with estimation of model error terms [Apr-Jun]
  - Propagate ME for observable and other QoIs [Jul-Sep]

- FY17-18
  - Automate and explore calibration over range of conditions of high-fidelity model
  - Attribute predictive model error to different parameters
  - Scale up to P2
  - Include experimental data

Challenges and Mitigation

- Density estimation is more challenging than parameter estimation
  - Inverse problem is ill-posed or intractable

  $\Rightarrow$ Employ approximate or empirical likelihoods

- Potentially a high-dimensional Bayesian problem
  - Full posterior may be inaccessible...

  $\Rightarrow$ Resort to optimization algorithms in no-noise case
  - ... or hard to sample from

  $\Rightarrow$ Adaptive MCMC algorithms, including LIS

- Sparse data (expensive high-fidelity simulations)
  - With low information content, calibration may struggle

  $\Rightarrow$ More informative priors/regularization

Current status: low-fi versus high-fi calibration

- Task: Low fidelity model calibration with respect to data from high-fidelity model
- Data: the nominal run from forward UQ study (quadrature point at the origin)
- QoI: 1D profiles/slices of temperature and/or pressure; potentially parameterized if common shapes emerge
- Simplified (low-fi) model to be calibrated: constant (in time and space) assumptions for Smagorinsky constant C_R and turbulent Prandtl number Pr_t.

$$f(x;\lambda) \approx g(x), \text{ where } \lambda = (C_R, Pr_t)$$

As a first step, pre-build a surrogate for $f(x;\lambda)$. That means a set of (~ 20) runs at predefined values of $\lambda = (C_R, Pr_t)$.

Current status: low-fi versus high-fi calibration

- Task: Low fidelity model calibration with respect to data from high-fidelity model
- Data: the nominal run from forward UQ study (quadrature point at the origin)
- QoI: 1D profiles/slices of temperature and/or pressure; potentially parameterized if common shapes emerge
- Simplified (low-fi) model to be calibrated: constant (in time and space) assumptions for Smagorinsky constant C_R and turbulent Prandtl number Pr_t.

$$f(x;\lambda) \approx g(x), \text{ where } \lambda = (C_R, Pr_t)$$

As a first step, pre-build a surrogate for $f(x;\lambda)$. That means a set of (~ 20) runs at predefined values of $\lambda = (C_R, Pr_t)$.

Current status: low-fi versus high-fi calibration

- Task: Low fidelity model calibration with respect to data from high-fidelity model
- Data: the nominal run from forward UQ study (quadrature point at the origin)
- QoI: 1D profiles/slices of temperature and/or pressure; potentially parameterized if common shapes emerge
- Simplified (low-fi) model to be calibrated: constant (in time and space) assumptions for Smagorinsky constant C_R and turbulent Prandtl number Pr_t.

$$f(x;\lambda) \approx g(x), \text{ where } \lambda = (C_R, Pr_t)$$

As a first step, pre-build a surrogate for $f(x;\lambda)$. That means a set of (~ 20) runs at predefined values of $\lambda = (C_R, Pr_t)$.

Main demo: Run $f(x;\lambda)$ on-the-fly, during calibration. Hands-on P1.

Code status

`UQTk`: lightweight UQ library, mostly C++ with Python interface (www.sandia.gov/uqtoolkit)

- `pce` : PC utilities, both intrusive and non-intrusive
- `quad` : Quadrature generation, including sparse
- `mcmc` : MCMC sampling, single-site and adaptive

- `infer` : Inference with density estimation; multivariate P.C. R.V., includes

```
infer_model(*Func, likelihoodType, ...)
```

- `model_inf` : Command line MCMC sampler, employs a few standard forward functions, e.g. a polynomial surrogate.

Ideally, a black-box C/C++ function `Func(...)` needs to be implemented using LES code-base to replicate $f(x;\lambda)$.

Connections to other tasks

- Major dependence on LES code/availability

  Conceptual connections: support and take advantage of

- GSA/Forward UQ:
  - pick the nominal run as a high-fidelity data
  - model error attribution similar in flavor to Sobol/GSA

- Multifidelity:
  - stochastic representation of model discrepancy from surrogate
  - compare with embedded model error representation

- Mesh discretization error:
  - low-fi (coarse mesh) versus high-fi (dense mesh)
  - represent error induced by resolution coarsening

Summary

What we have done

- The methodology is tested and fine tuned
- Identified ‘model simplification’ axis for the first set of demos
- Code base is ready to be unleashed
- Xun up to speed (and more!) with the method

What will we do next

- Carefully select conditions and output quantities of interest for calibration
- Build surrogates for static constant treatment cases for demo
- Get hands on a working case and implement model error calibration directly
- Keep talking to MDE, Multifi and Forward UQ.

Summary

What we have done

- The methodology is tested and fine tuned
- Identified 'model simplification' axis for the first set of demos
- Code base is ready to be unleashed
- Xun up to speed (and more!) with the method

What will we do next

- Carefully select conditions and output quantities of interest for calibration
- Build surrogates for static constant treatment cases for demo
- Get hands on a working case and implement model error calibration directly
- Keep talking to MDE, Multifi and Forward UQ.

Full Likelihood

$$L(\alpha) = p(D|\alpha) = \pi_f(y_{\text{data},1}, \ldots, y_{\text{data},N}|\alpha)$$

where:
$\pi_f(\cdot, \alpha)$: N-variate density of the random variable $(f_1, \ldots, f_N)$
with $f_i = f(x_i, \lambda(\alpha))$

Problem: $\pi_f(\cdot)$ is degenerate in general when $N > M$

Consider a case with $M = 1$, $\lambda \sim N(\mu, \sigma^2)$, and $f = \lambda x$
Let $N = 2$, hence $(f_1, f_2) = (\lambda x_1, \lambda x_2)$ for any λ sample
With $f_1/x_1 = f_2/x_2 = \lambda$, (f_1, f_2) are dependent and
$\pi_f(\cdot|\mu, \sigma)$ is non-zero only along the line $f_2 = (x_2/x_1)f_1$

hence $\pi_f(y_{\text{data},1}, y_{\text{data},2}|\mu, \sigma)$ is non-zero only along the line
$y_{\text{data},2}/x_2 = y_{\text{data},1}/x_1$

Marginalized Likelihood

$$L(\alpha) = p(D|\alpha) = \prod_{i=1}^{N} \pi_{f_i}(y_{\text{data},i}|\alpha)$$

where $\pi_{f_i}(\cdot, \alpha)$ is the univariate density of the RV $f_i = f(x_i, \lambda(\alpha))$

Problem: the likelihood has multiple singularities corresponding to α values leading to vanishing marginal variances at each x_i

Gaussian example: Let $f_i \sim \mathrm{N}(\mu_i(\alpha), \sigma_i(\alpha)^2)$, then

$$L(\alpha) = \frac{1}{(2\pi)^{N/2}} \prod_{i=1}^{N} \frac{1}{\sigma_i(\alpha)} \exp\left(\frac{(\mu_i(\alpha) - y_{\text{data},i})^2}{2\sigma_i(\alpha)^2} \right)$$

Multiple singularities, $\sigma_i(\alpha) = 0$, $i = 1, \ldots, N$

Posterior maximization always finds one of these singularities, fitting one point perfectly, while misfitting the rest ($\Rightarrow$ priors)

Approximate Bayesian Computation (ABC)

Employ a kernel density as a pseudo-likelihood to enforce select constraints:

- Uncertain prediction $p(y|D)$ is centered on the data
  - With $\mu_i(\alpha) = \mathrm{E}_\xi[f(x_i, \lambda(\xi;\alpha))]$:

    minimize $\| \mu_i(\alpha) - y_{\mathrm{data},i} \|_2^2$
- The width of the distribution $p(y|D)$ is consistent with the spread of the data around the nominal model prediction
  - With $\sigma_i^2(\alpha) = \mathrm{V}_\xi[f(x_i, \lambda(\xi,\alpha))]$:

    minimize $\| \sigma_i(\alpha) - \gamma|\mu_i(\alpha) - y_{\mathrm{data},i}| \|_2^2$
  - γ is a factor that specifies the desired match between σ_i and the discrepancy $|\mu_i(\alpha) - y_{\mathrm{data},i}|$, on average

ABC Likelihood

With $\rho(\mathcal{S})$ being a metric of the statistic $\mathcal{S}$, use the kernel function as an ABC likelihood:

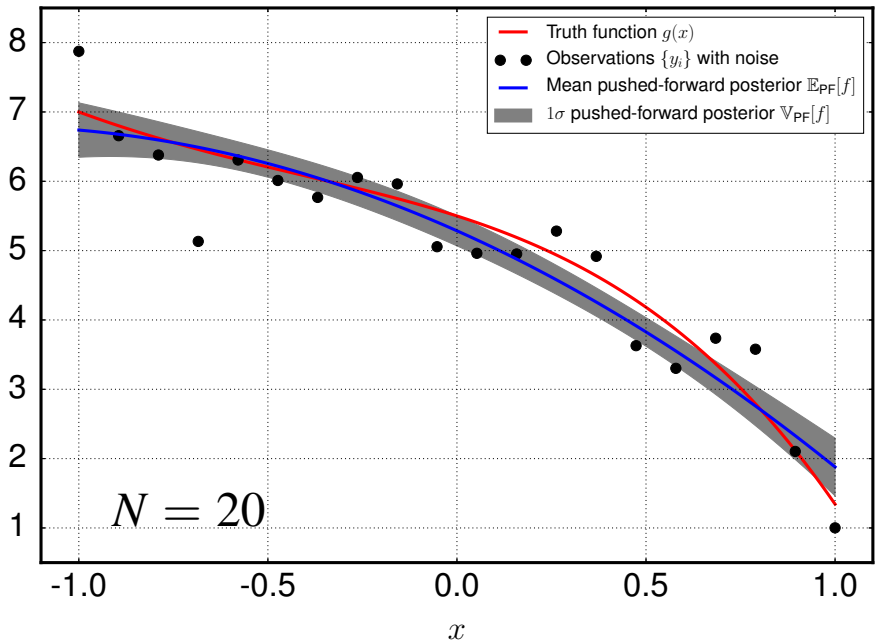
$$L_{\mathrm{ABC}}(\alpha) = \frac{1}{\epsilon} K\left(\frac{\rho(\mathcal{S})}{\epsilon}\right)$$

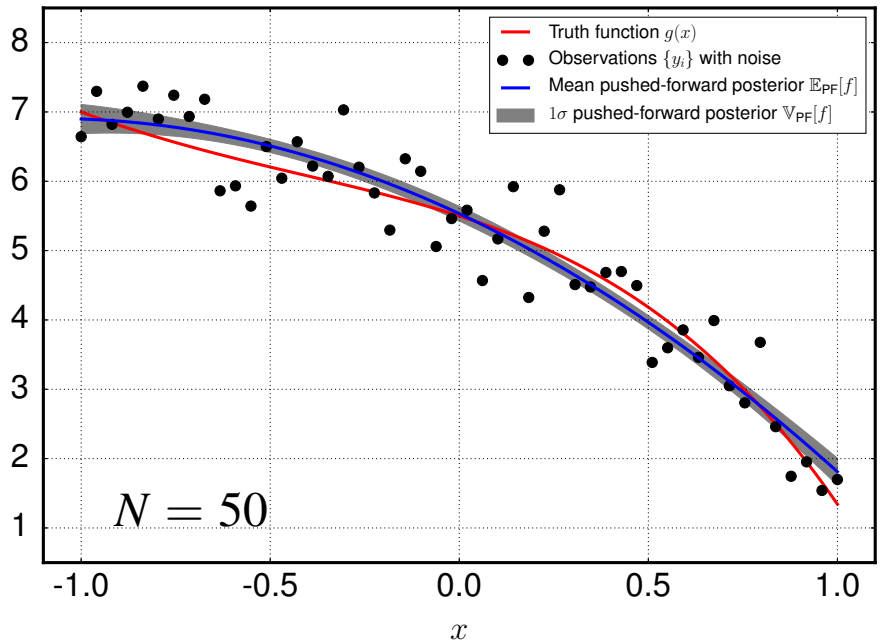
where ϵ controls the severity of the consistency control

Propose the Gaussian kernel density:

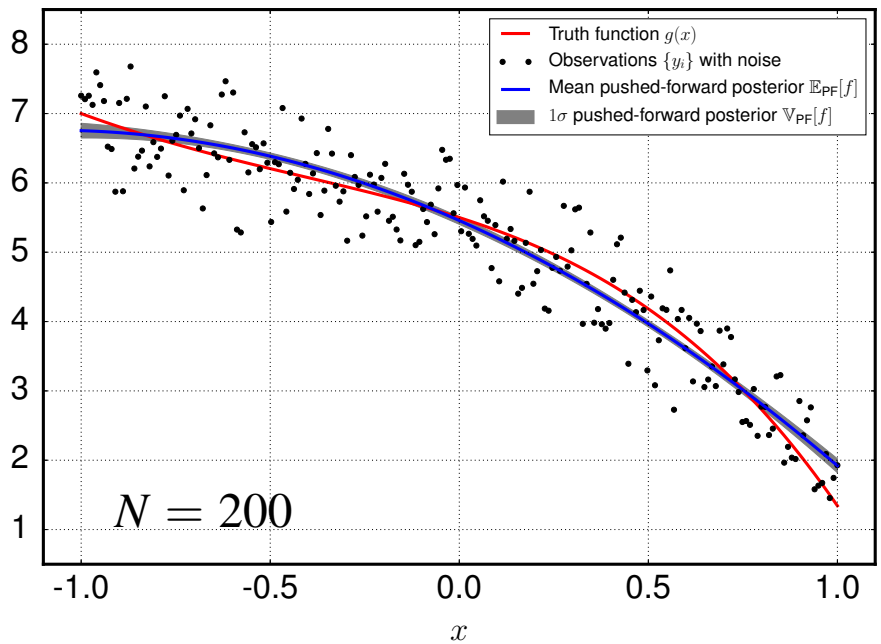
$$L_{\epsilon}(\alpha) = \frac{1}{\epsilon\sqrt{2\pi}} \prod_{i=1}^{N} \exp\left(-\frac{(\mu_i(\alpha) - y_{\mathrm{d},i})^2 + (\sigma_i(\alpha) - \gamma|\mu_i(\alpha) - y_{\mathrm{d},i}|)^2}{2\epsilon^2}\right)$$

Quadratic-fit – Classical Bayesian likelihood

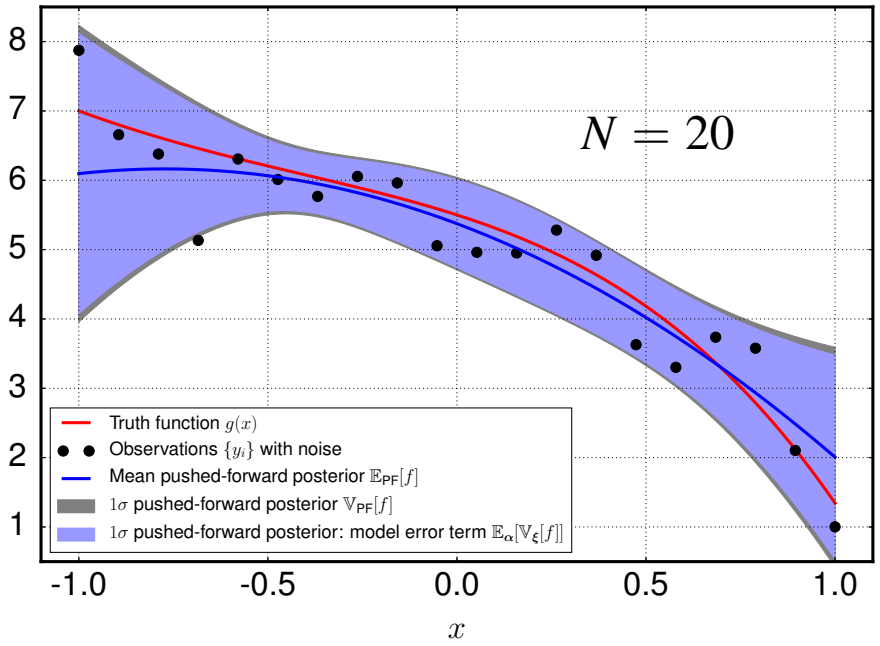


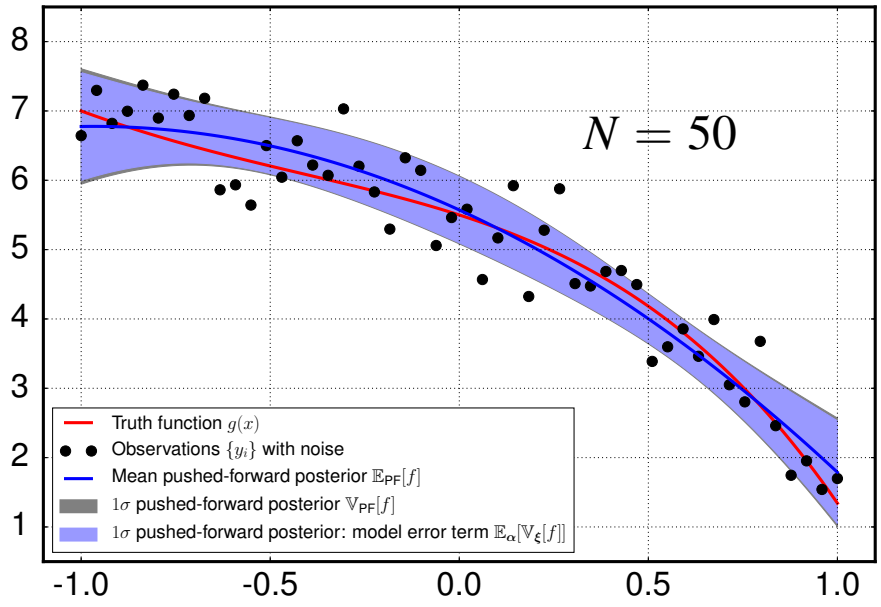


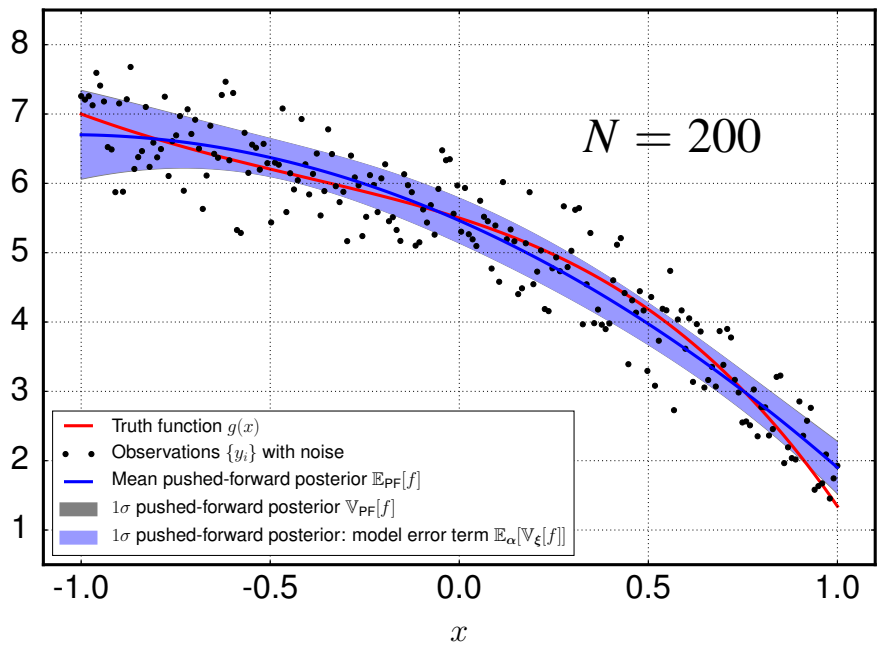
- With additional data, predictive uncertainty around the wrong model is indefinitely reducible
- Predictive uncertainty not indicative of discrepancy from truth



Quadratic-fit – ModErr – MargGauss

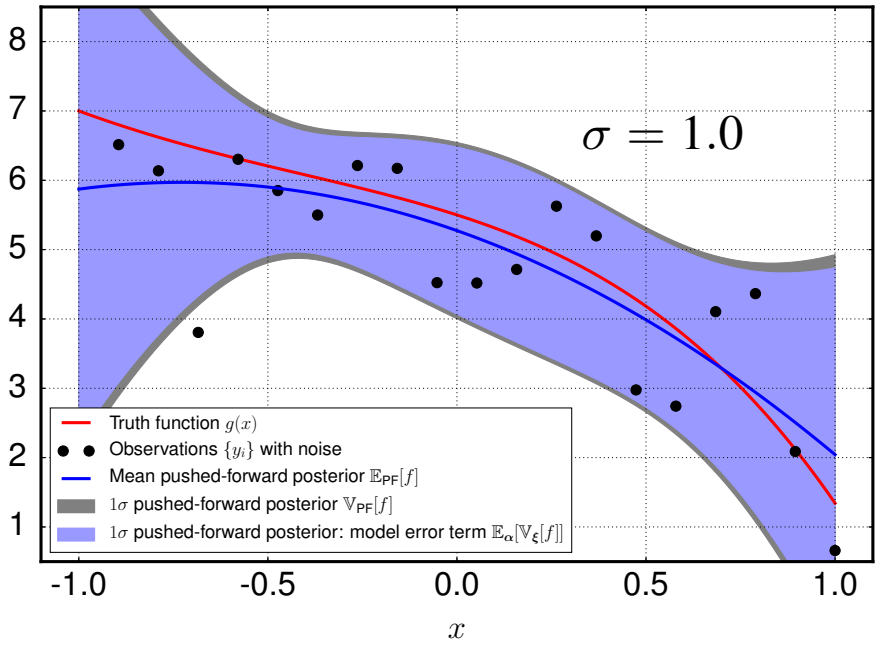


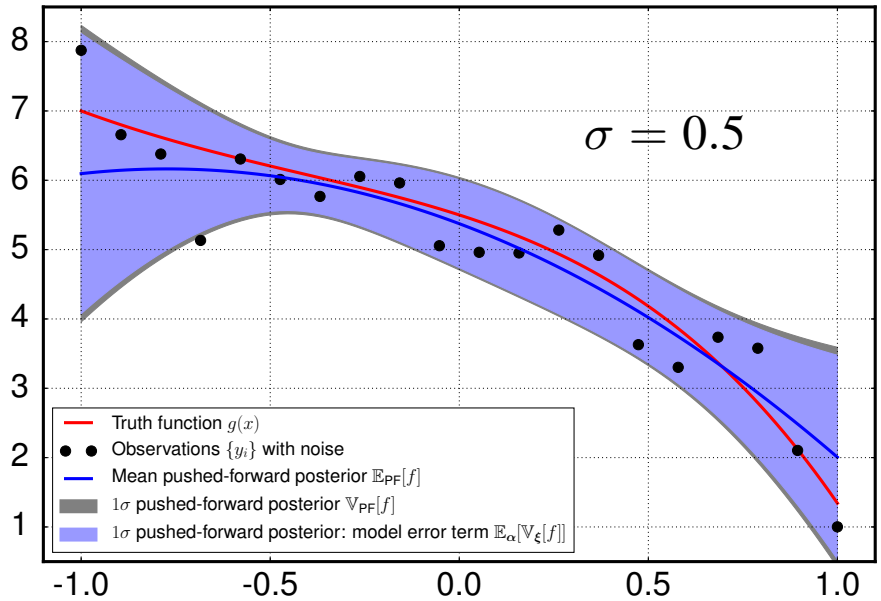


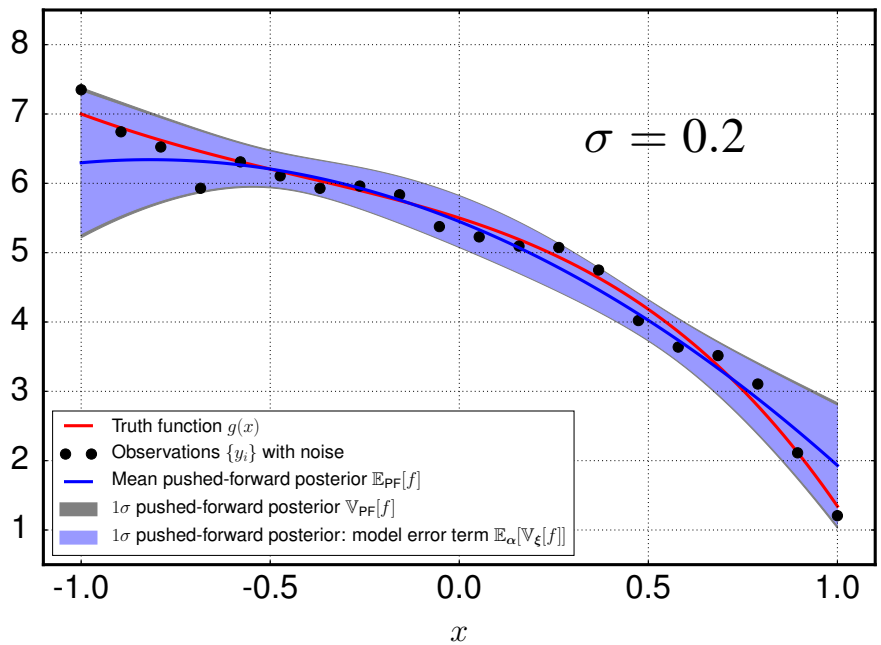


- With additional data, predictive uncertainty due to data noise is reducible
- Predictive uncertainty due to model error is not reducible

Quadratic-fit – ModErr – MargGauss

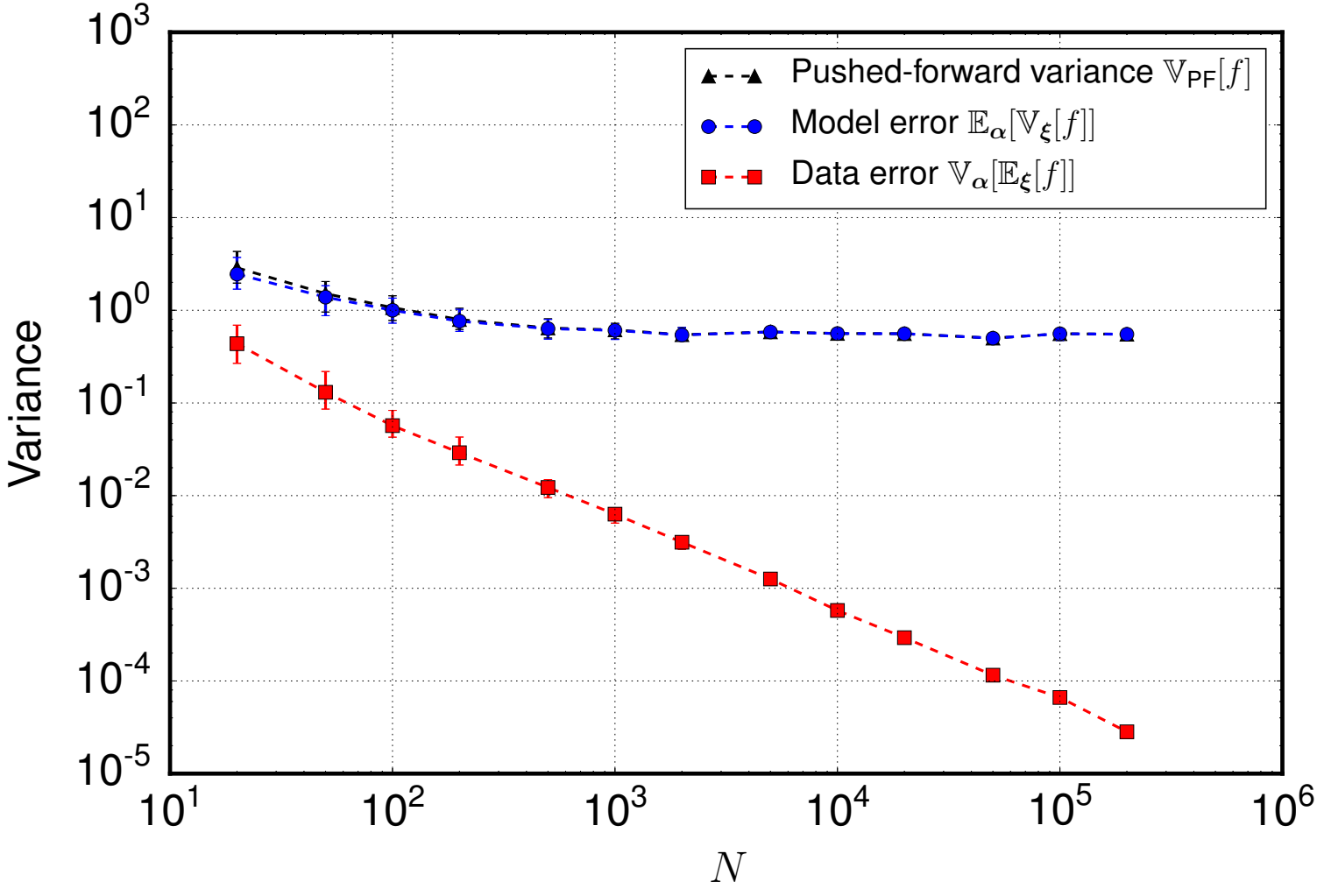






- Predictive uncertainty composed of both model-error and data-noise components
- The data-noise component is reducible with lower-noise in the data

Quadratic-fit – ModErr – MargGauss

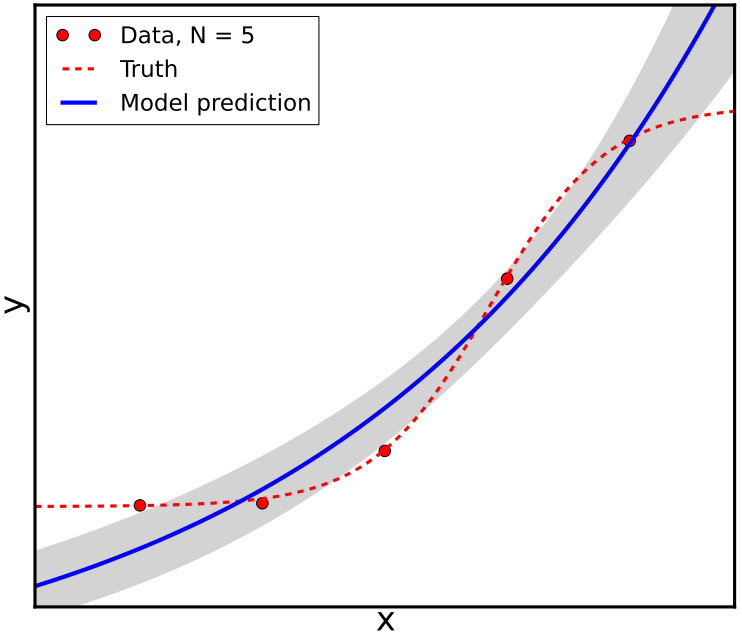


Calibrating a quadratic $f(x)$ w.r.t. $g(x) = 6 + x^2 + 0.5(x+1)^{3.5}$

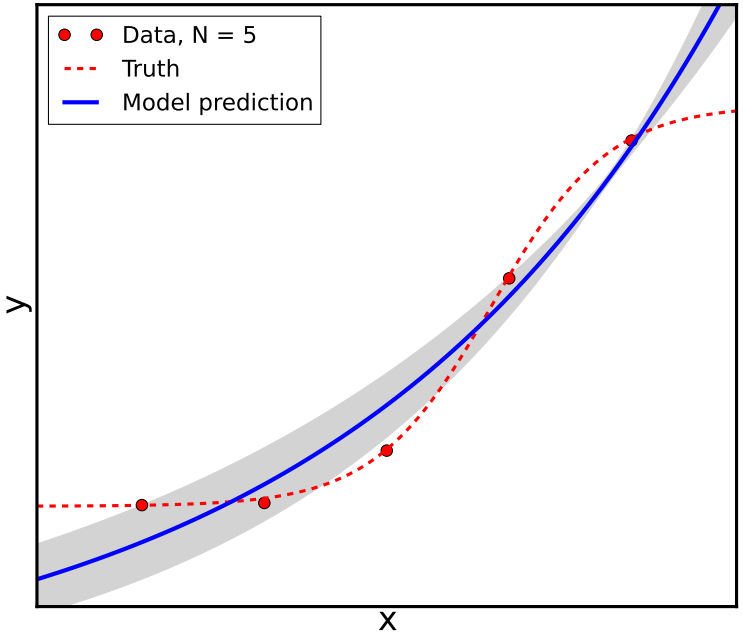
Predictions account for model error: example 1

Calibrating an exponential model $f(x; \lambda_1, \lambda_2) = \lambda_2 e^{\lambda_1 x} - 2$
with data from a hyperbolic tangent model $g(x) = \tanh(3(x - 0.3))$

Additive Gaussian error



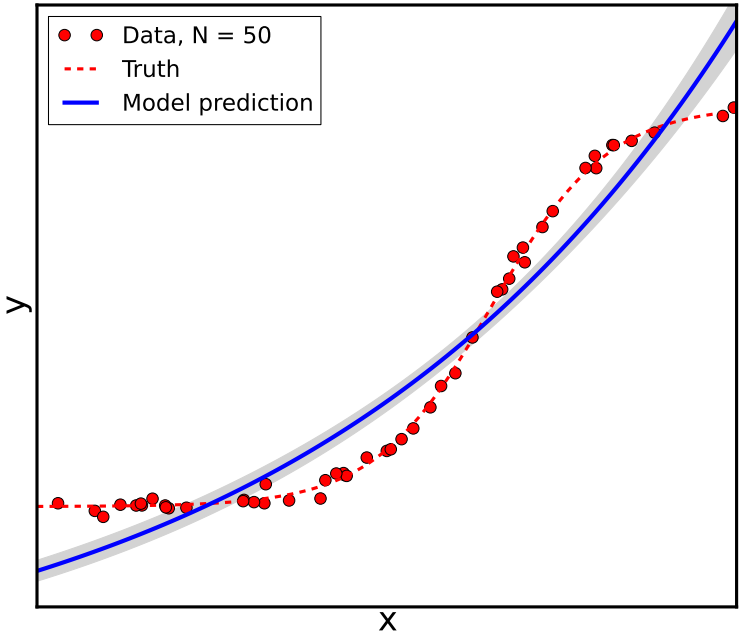
Embedded model error



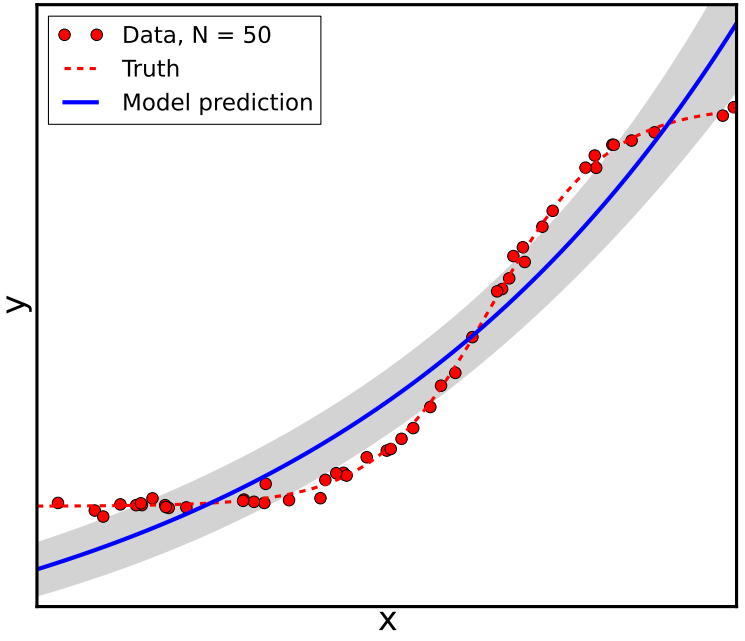
Predictions account for model error: example 1

Calibrating an exponential model $f(x; \lambda_1, \lambda_2) = \lambda_2 e^{\lambda_1 x} - 2$
with data from a hyperbolic tangent model $g(x) = \tanh(3(x - 0.3))$

Additive Gaussian error



Embedded model error



Model Error formulation – noisy data

$$y = f(x, \lambda) + \epsilon$$

Let $\epsilon \sim N(0, \sigma^2)$. With N *i.i.d.* data points we have

$$y_i = f(x_i, \lambda) + \epsilon_i, \quad i = 1, \ldots, N$$

For Hermite-Gaussian PC:

$$\begin{aligned}
\lambda &= \sum_{k=0}^{P} \alpha_k \Psi_k(\xi_1, \cdots, \xi_d), \qquad \alpha \equiv (\alpha_0, \cdots, \alpha_P) \\
f(x, \lambda) &= \sum_{k=0}^{P} f_k(x, \alpha) \Psi_k(\xi_1, \cdots, \xi_d) \\
y_i &= h(x_i, \alpha, \sigma, \xi) = \sum_{k=0}^{P} f_k(x_i, \alpha) \Psi_k(\xi_1, \cdots, \xi_d) + \sigma \xi_{d+i}
\end{aligned}$$

Augmented PC germ $\xi = (\xi_1, \cdots, \xi_d, \xi_{d+1}, \cdots, \xi_{d+N})$

Chemistry problem – ABC

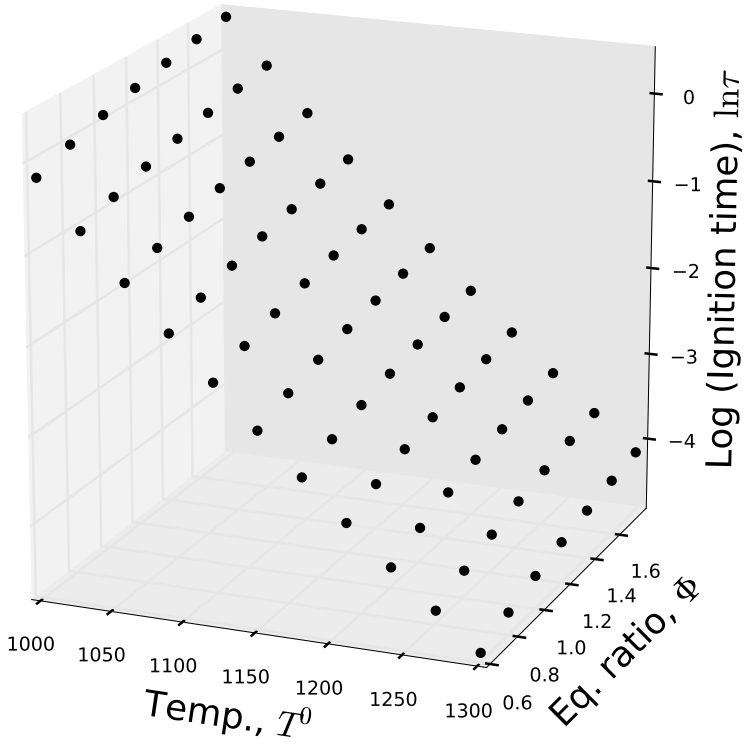
- Homogeneous ignition, methane-air mixture
- Single-step global reaction model calibrated against a detailed chemical kinetic model

- Data: ignition time; range of initial T & equivalence ratio
- Single-step model:

$$\mathrm{CH_4} + 2\mathrm{O_2} \rightarrow \mathrm{CO_2} + 2\mathrm{H_2O}$$

$$\begin{aligned} \mathfrak{R} &= [\mathrm{CH_4}][\mathrm{O_2}]k_f \\ k_f &= A\exp(-E/R^oT) \end{aligned}$$

- $(\ln A, E) = \sum_k \boldsymbol{\alpha}_k \Psi_k(\xi)$

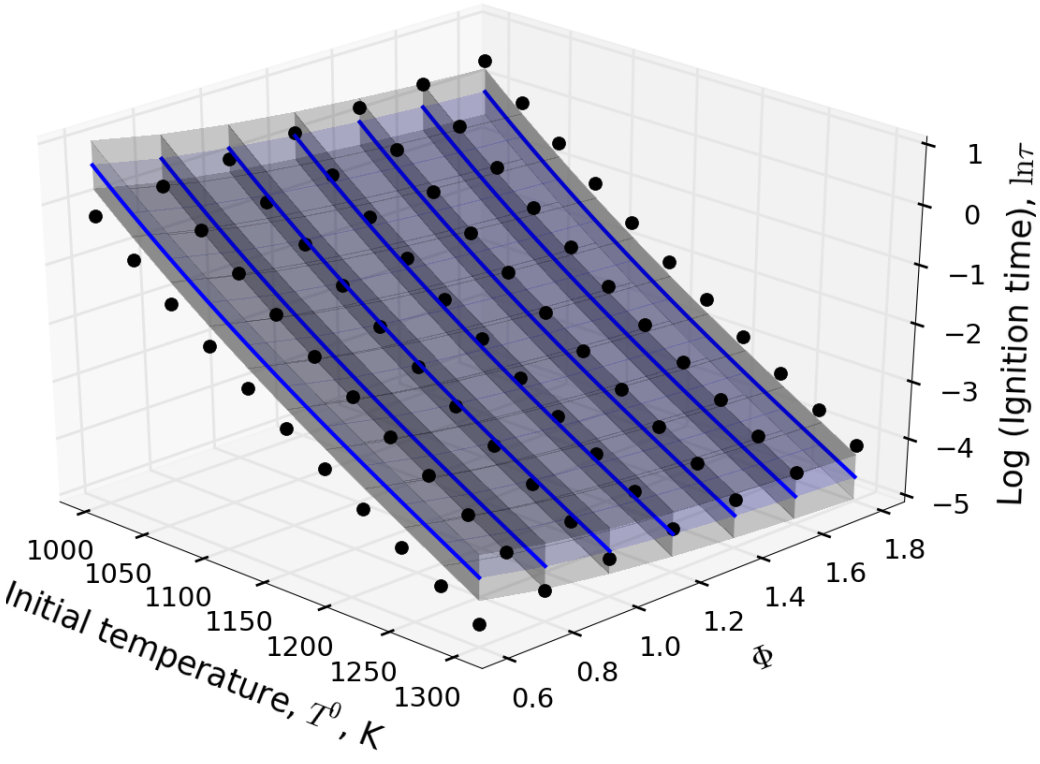


Quality of Uncertain Calibrated Model Predictions

Calibrated uncertain fit model is consistent with the detailed-model data.

Over the range of (T^0, Φ):

- MAP predictive mean ignition-time is centered on the data
- MAP predictive stdv is consistent with the scatter of the data



K. Sargsyan, HNN, and R. Ghanem
"On the Statistical Calibration of Physical Models"
Int. J. Chem. Kin., 47(4): 246-276, 2015

TransCom3 Experiment of CO_2 Flux Inversion

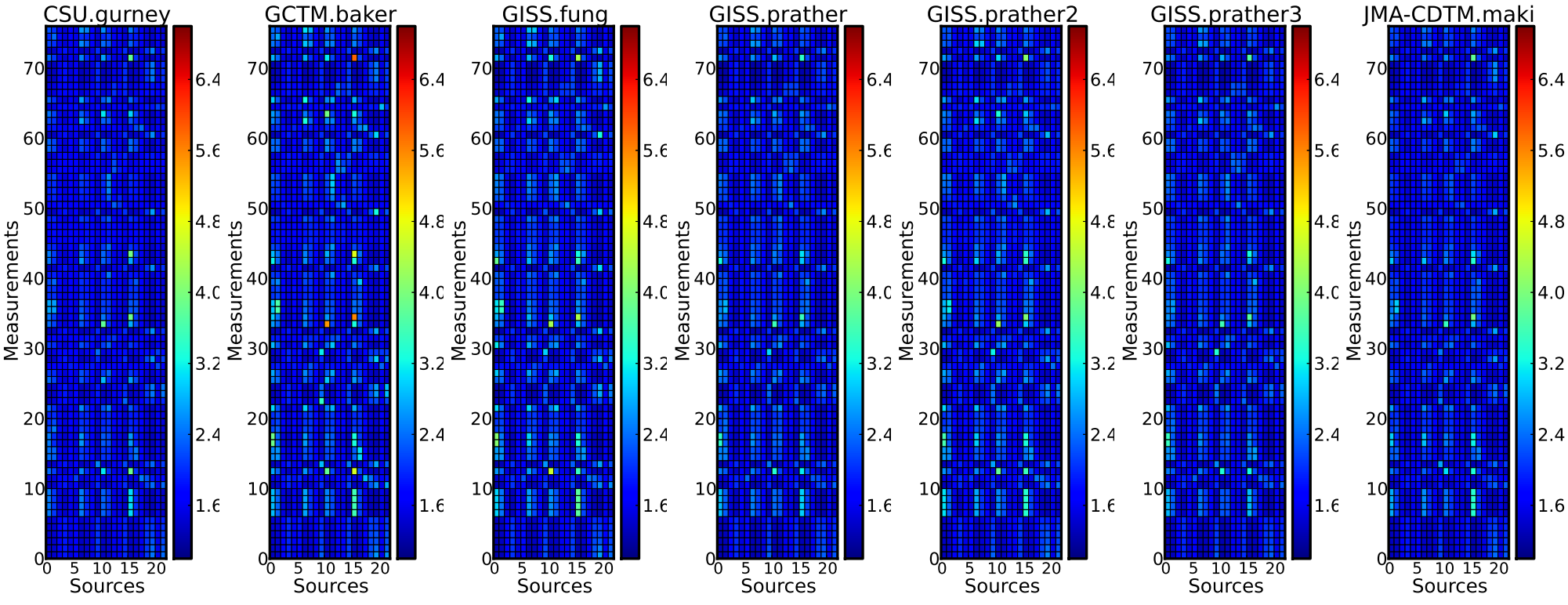
[Gurney *et al.*, Tellus B, 2003]

- Observations $\mathbf{d}$ at $N = 77$ sites around the world
- Inverse problem: find fluxes $\mathbf{s}$ at $M = 22$ locations
- Linearized 'response' model $\mathbf{R}$, such that $\mathbf{d} \approx \mathbf{Rs}$

$$\mathbf{d} = \mathbf{Rs} + \boldsymbol{\epsilon}_\mathbf{d}$$

- Model $\mathbf{R}$ is never perfect thus contaminating the inversion
- The inferred values of $\mathbf{s}$ compensate for model deficiencies
- $\boldsymbol{\epsilon}_\mathbf{d}$ is meant to capture data errors, but is 'entangled' with model errors

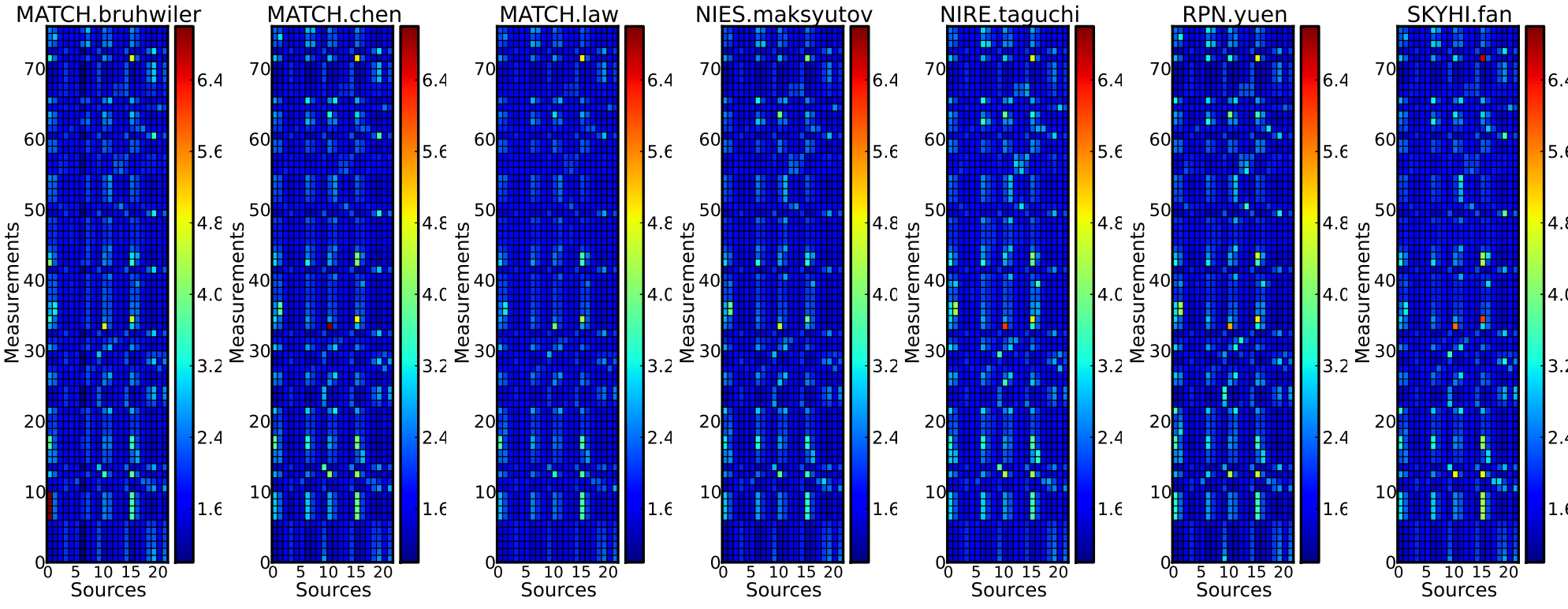
Consider 14 different response models $\mathbf{R}$



Infer fluxes $\mathbf{s}$, given measurements $\mathbf{d}$ to satisfy $\mathbf{d} \approx \mathbf{Rs}$

- Conventional additive Gaussian error (least-squares): $\mathbf{d} = \mathbf{Rs} + \xi$
- Embed probabilistic model for fluxes $\boldsymbol{s}$: $\mathbf{d} = \mathbf{R}(\mu_{\mathbf{s}} + \mathbf{C}_{\mathbf{s}}\xi)$

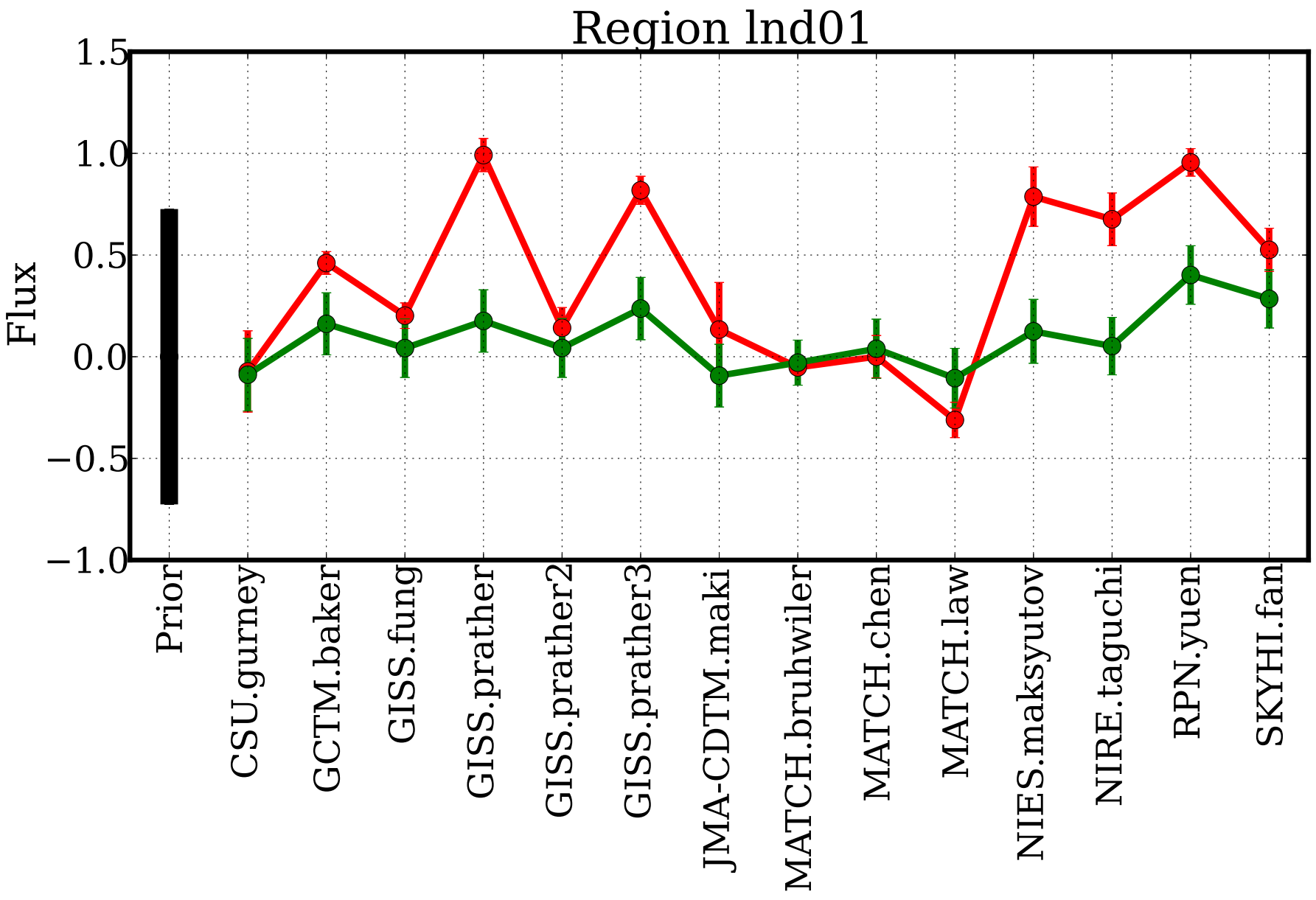
Consider 14 different response models $\mathbf{R}$



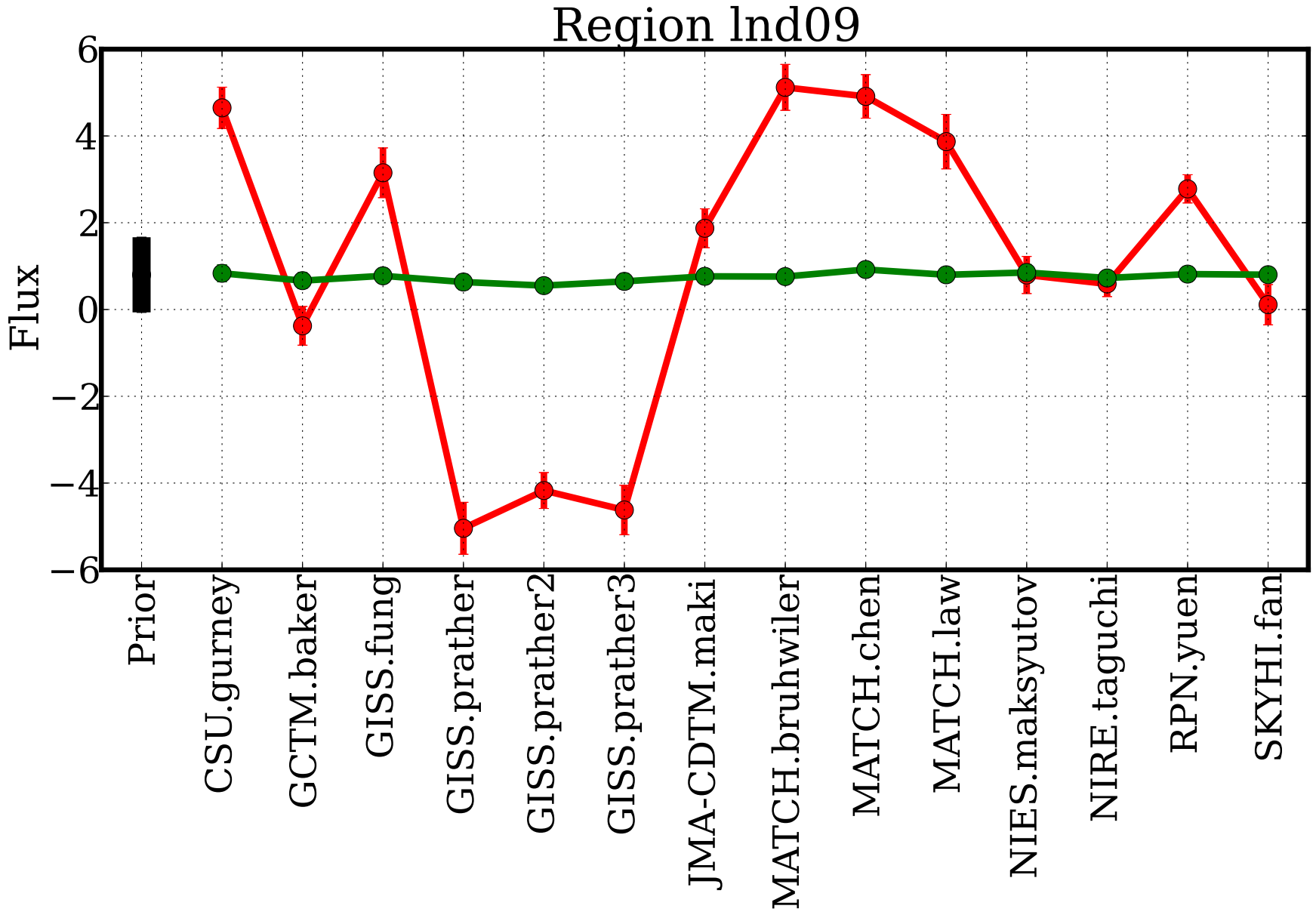
Infer fluxes $\mathbf{s}$, given measurements $\mathbf{d}$ to satisfy $\mathbf{d} \approx \mathbf{Rs}$

- Conventional additive Gaussian error (least-squares): $\mathbf{d} = \mathbf{Rs} + \xi$
- Embed probabilistic model for fluxes $\boldsymbol{s}$: $\mathbf{d} = \mathbf{R}(\mu_{\mathbf{s}} + \mathbf{C}_{\mathbf{s}}\xi)$

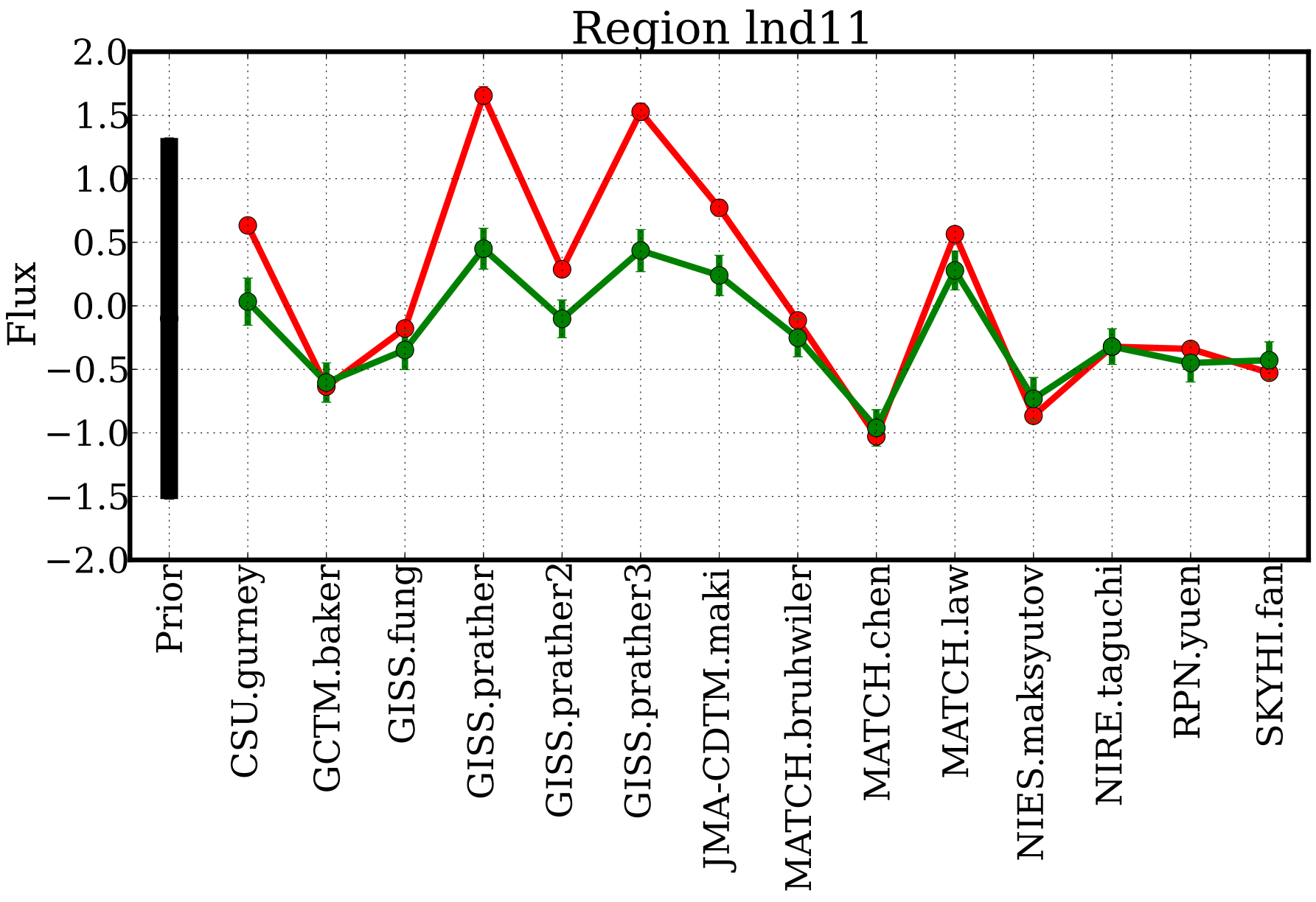
Inferred fluxes show less variability across models



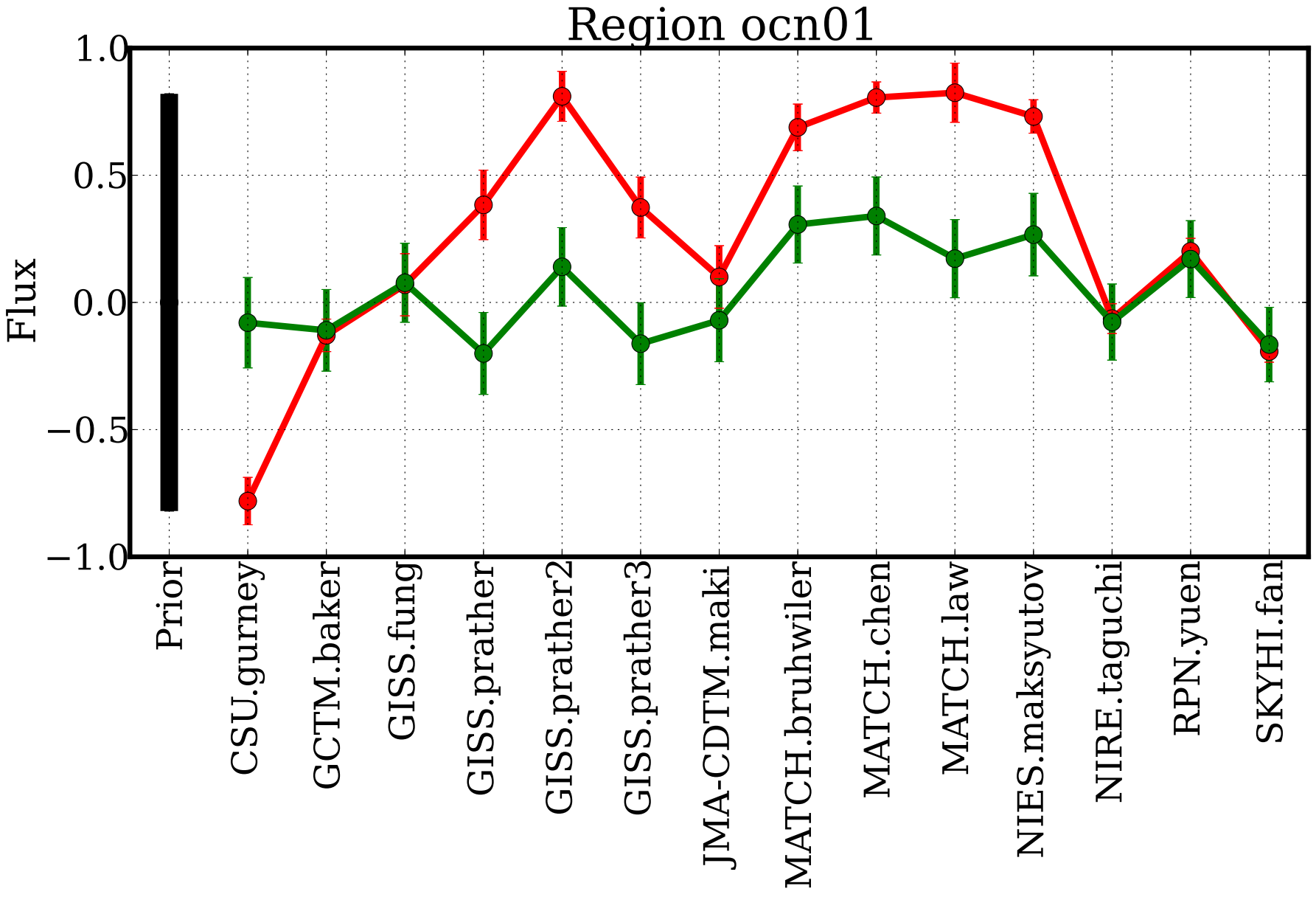
Inferred fluxes show less variability across models



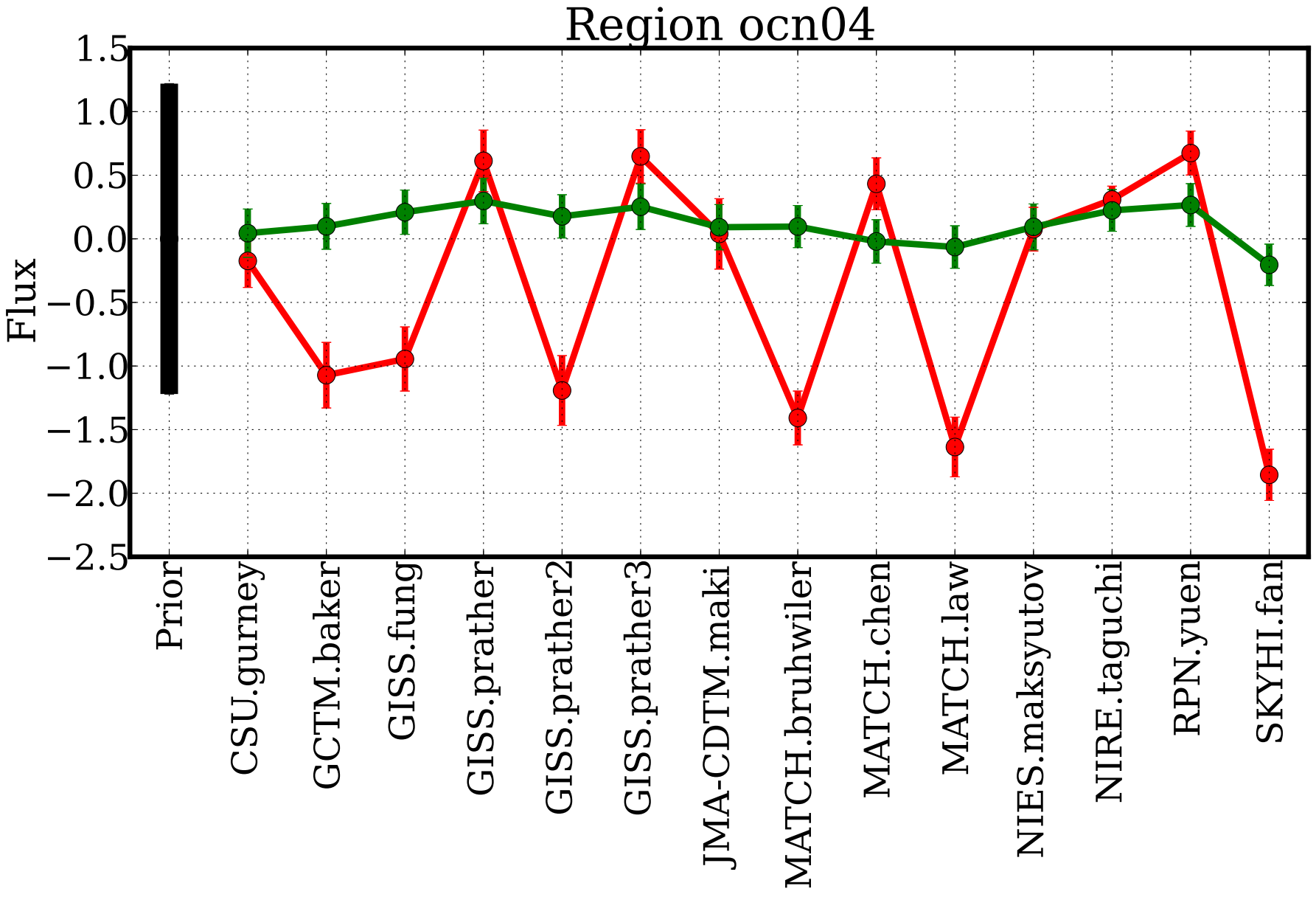
Inferred fluxes show less variability across models



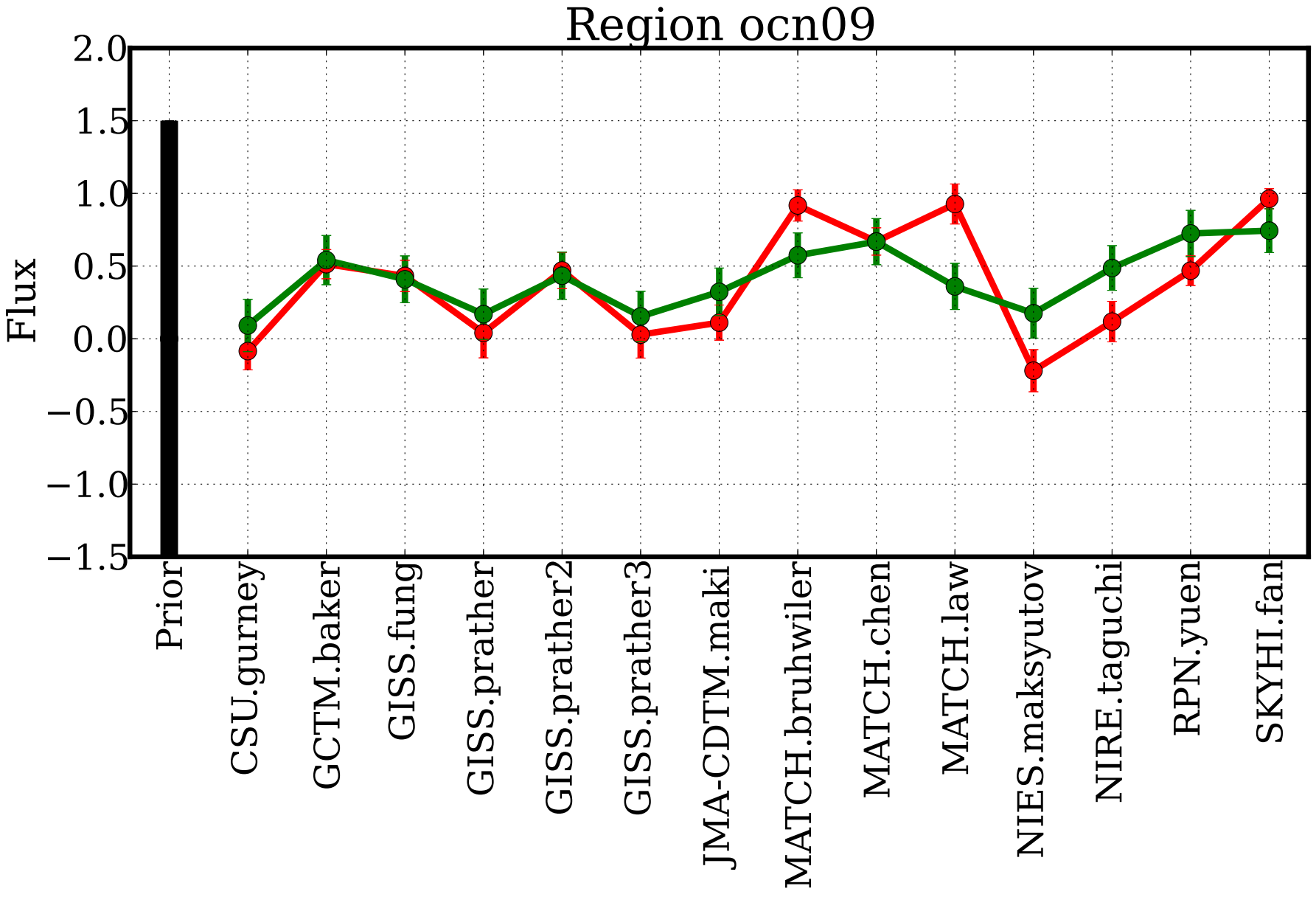
Inferred fluxes show less variability across models



Inferred fluxes show less variability across models



Inferred fluxes show less variability across models



Inferred fluxes show less variability across models