Multi-fidelity stochastic modeling with Gaussian processes:

Learning and optimization under uncertainty

Paris Perdikaris

Massachusetts Institute of Technology, Department of Mechanical Engineering

Web: `http://web.mit.edu/parisp/www/`

Email: `parisp@mit.edu`

NIH IMAG/MSM webinar
July 21, 2016

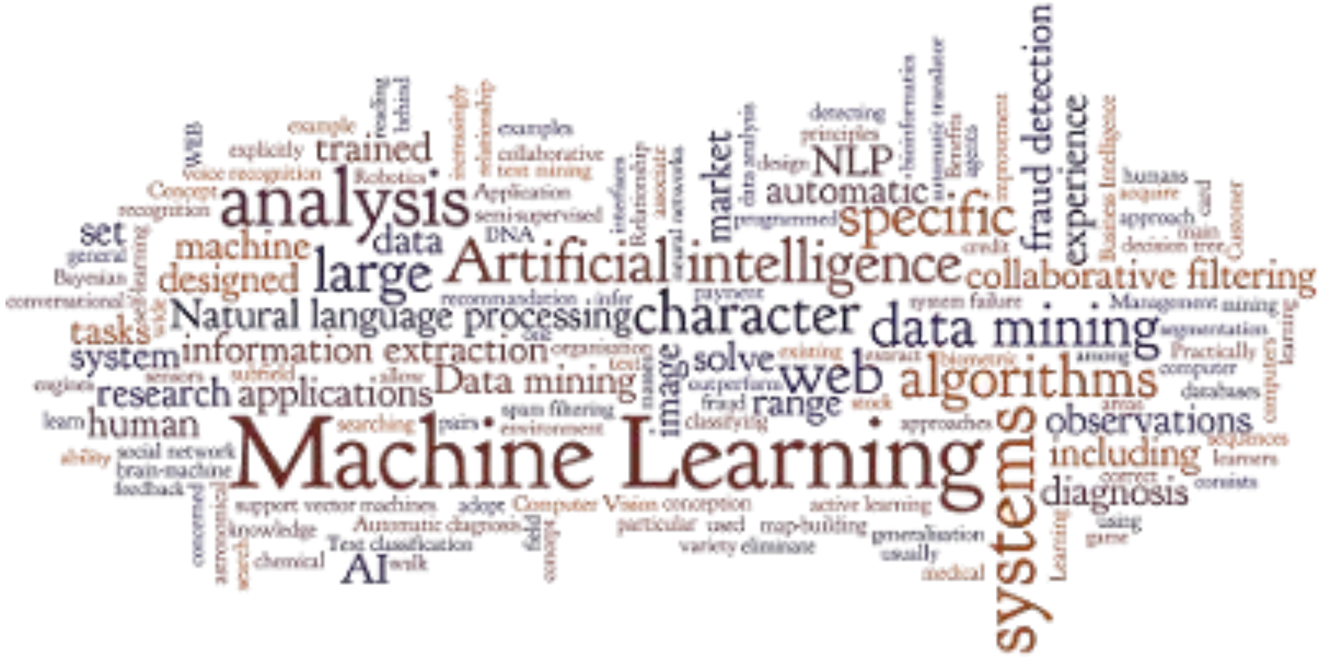
Overview

Goal: Synergistically combine all available information sources to construct accurate response surfaces *(regression, optimization, inverse problems, uncertainty quantification, and beyond).*

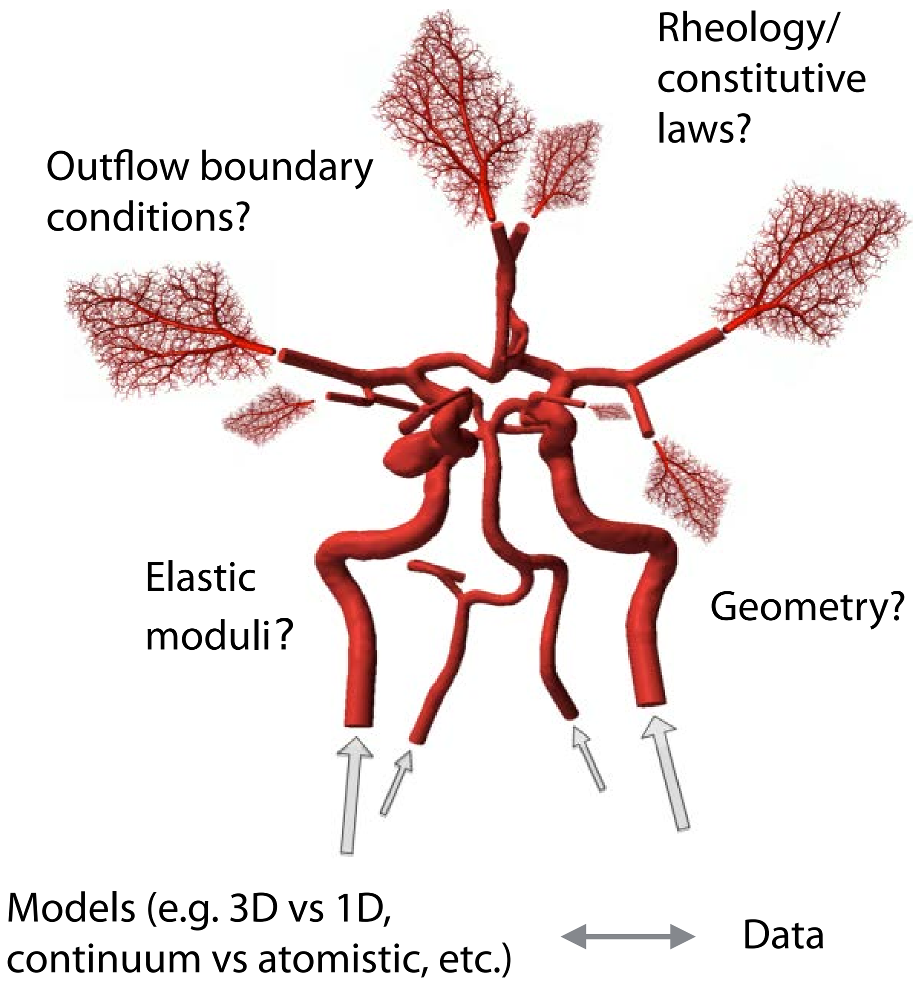
Probabilistic Machine Learning enables:

- Combining seemingly different information sources (e.g. measurements & simulations)
- Exploring cross-correlations between variables and identifying interactions
- Constructing predictive algorithms and perform inference with quantified uncertainty
- Supervised (regression, classification), unsupervised (clustering, dimensionality reduction), reinforcement learning

Multi-fidelity modeling: Utilize cheap low-fidelity models supplemented with a few realizations of high-fidelity models. Exploring cross-correlations can lead to orders of magnitude of speed up in computation.



A motivating example: *Calibration of blood flow simulations*



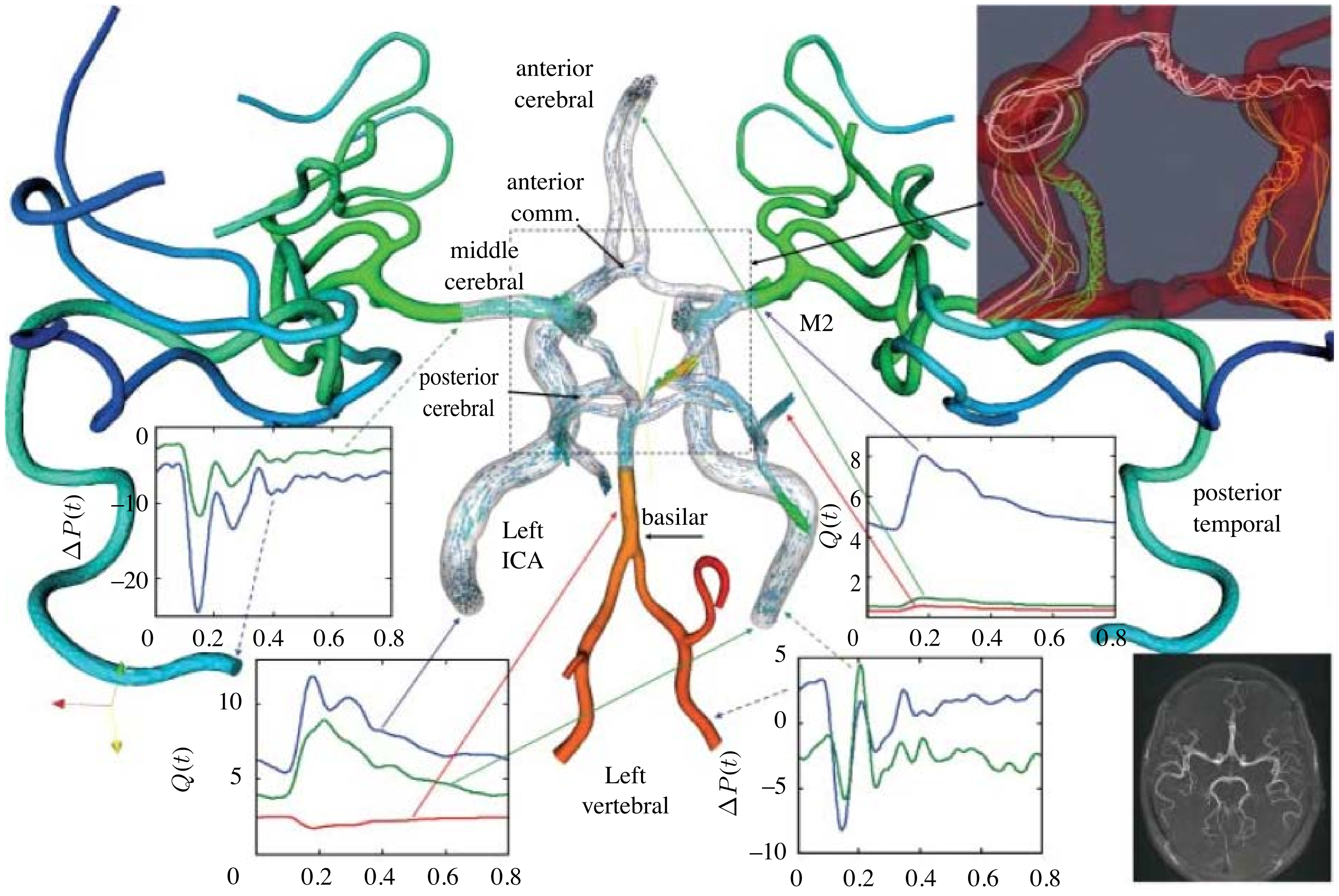
Outflow boundary conditions?

Rheology/ constitutive laws?

Elastic moduli?

Geometry?

Models (e.g. 3D vs 1D, continuum vs atomistic, etc.) ⟷ Data



Questions:

1. How can we construct predictive surrogate models that can seamlessly learn from heterogeneous information sources?
2. How can we quantify the uncertainty/error associated with the surrogate model predictions?
3. How can we optimally acquire new data under a limited budget?
4. How can we scale the workflow to problems of industrial complexity?

Gaussian processes

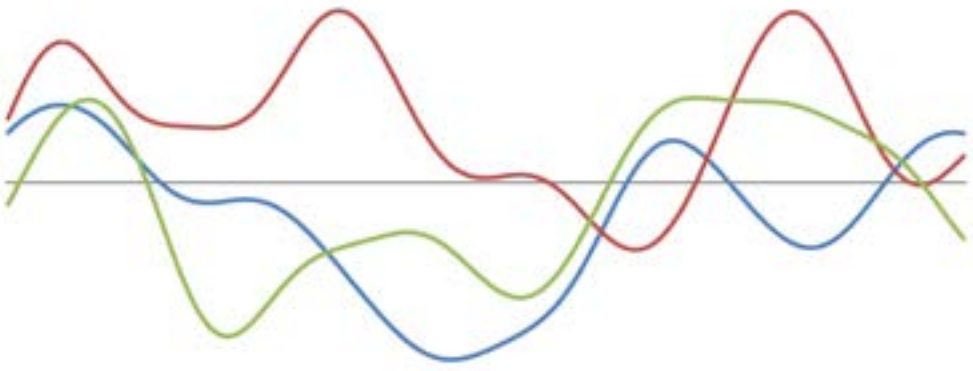
Starting point: The multivariate Gaussian distribution

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Generalization: The Gaussian process

$$\boldsymbol{\mu}_\infty = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{f}} \\ \cdots \\ \cdots \end{bmatrix} \text{ and } \mathbf{K}_\infty = \begin{bmatrix} \mathbf{K}_{\mathbf{ff}} & \cdots \\ \cdots & \cdots \end{bmatrix} \qquad \mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$$

mean function *covariance function*



Samples from a GP prior

Priors over functions: $f \sim \mathcal{GP}(\mu(x), K(\mathbf{x}, \mathbf{x}'; \theta))$

Infinite model, but finite observations: The marginalization property

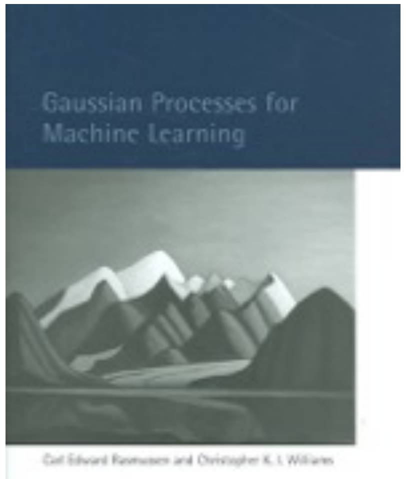
$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Posterior is also Gaussian:

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}(\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})$$



Gaussian Processes for Machine Learning

Rasmussen, C. E. Gaussian processes for machine learning 2006.

Gaussian process regression

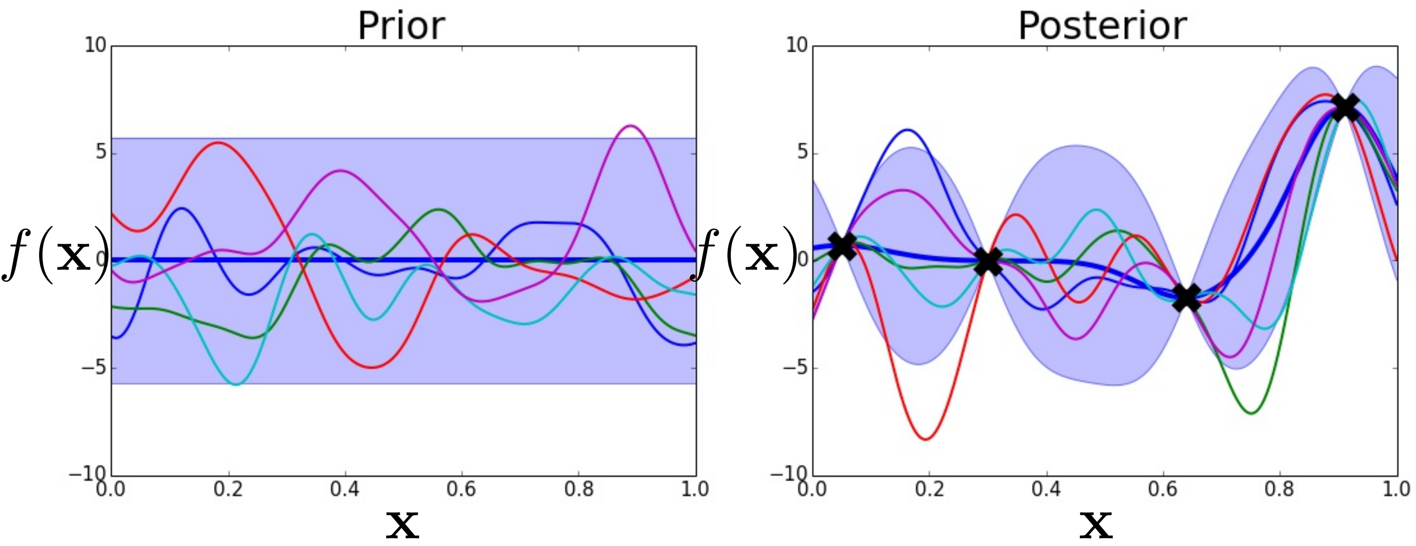
$$y = f(\mathbf{x}) + \epsilon, \qquad f \sim \mathcal{GP}(\mu(x), K(\mathbf{x}, \mathbf{x}'; \theta))$$

History:

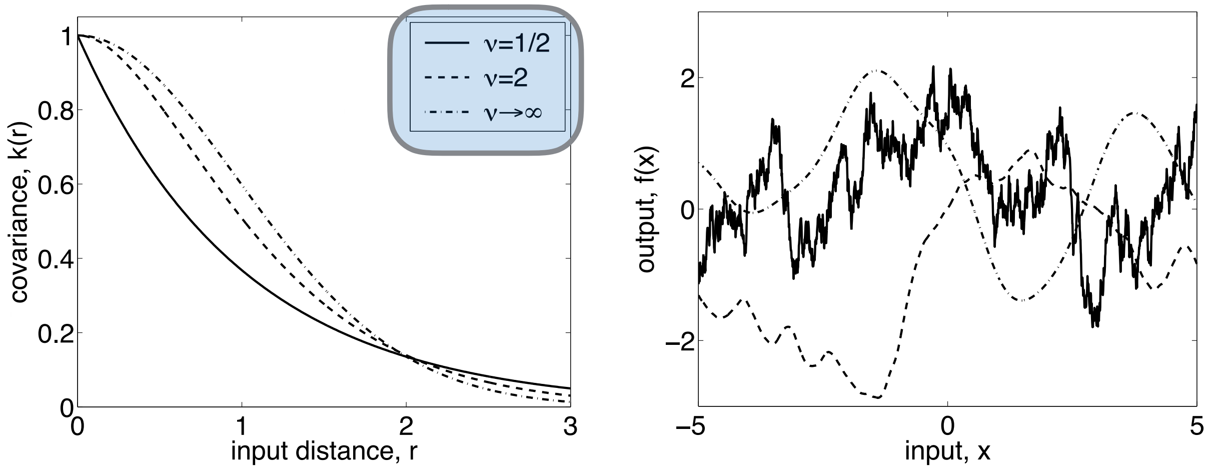
- Wiener–Kolmogorov filtering (1940)
- Kriging (spatial statistics, 1970)
- GP regression (machine learning, 1996)

Workflow:

- Assign a Gaussian process (GP) prior over functions
- Given a training set of observations (x,y) calibrate the GP hyper-parameters
- Use the conditional posterior [f|y] to infer predictions for unobserved x's with quantified uncertainty



covariance function	expression
constant	σ_0^2
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{\ell}r\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{\ell}r\right)$
exponential	$\exp(-\frac{r}{\ell})$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top\Sigma\tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top\Sigma\tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top\Sigma\tilde{\mathbf{x}}')}}\right)$

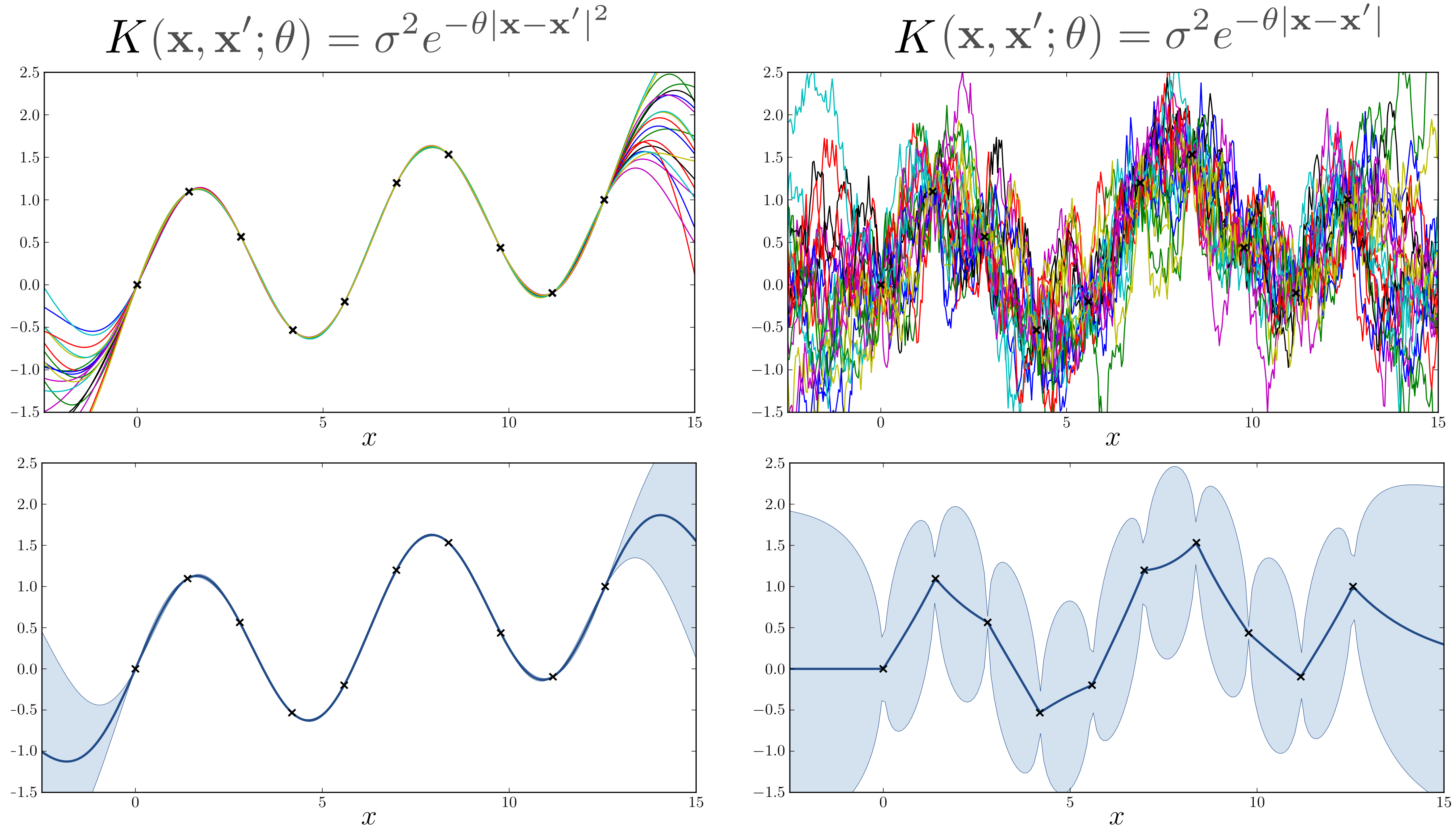


Rasmussen, C. E. Gaussian processes for machine learning 2006.

Importance of the prior

The choice of the covariance kernel has a big impact on the model as it is tightly related to:

- ○ The smoothness of the sample paths, hence the regularity of the predictor.
- ○ The accuracy and uncertainty of the predictor.
- ○ The conditioning of the correlation matrix, hence the efficiency of the learning algorithms.



$$K(\mathbf{x}, \mathbf{x}'; \theta) = \sigma^2 e^{-\theta|\mathbf{x}-\mathbf{x}'|^2}$$

$$K(\mathbf{x}, \mathbf{x}'; \theta) = \sigma^2 e^{-\theta|\mathbf{x}-\mathbf{x}'|}$$

Samples from a GP posterior

Posterior mean and variance

Training & prediction

Hyper-parameter estimation:

fequentist approach

The vector of hyper-parameters $\boldsymbol{\theta}$ is determined by maximizing the marginal log-likelihood of the observed data (the so called model evidence), i.e.,

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I}| - \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi \tag{8}$$

Bayesian approach

Assign priors over the hyper parameters and marginalize them out using MCMC.

Prediction:

If we consider a Gaussian likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_\epsilon^2 \boldsymbol{I})$ then the posterior distribution $p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X})$ is tractable and can be used to perform predictive inference for a new output f_*, given a new input $\boldsymbol{x}_*$ as

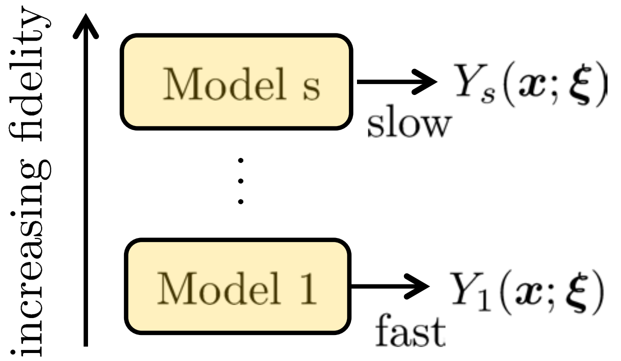
$$p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2), \tag{5}$$

$$\mu_*(\boldsymbol{x}_*) = \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \tag{6}$$

$$\sigma_*^2(\boldsymbol{x}_*) = \boldsymbol{k}_{**} - \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{N*}, \tag{7}$$

where $\boldsymbol{k}_{*N} = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_*, \boldsymbol{x}_N)]$, $\boldsymbol{k}_{N*} = \boldsymbol{k}_{*N}^T$, and $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$. Predictions are computed using the posterior mean μ_*, while prediction uncertainty is quantified through the posterior variance σ_*^2.

Multi-fidelity modeling

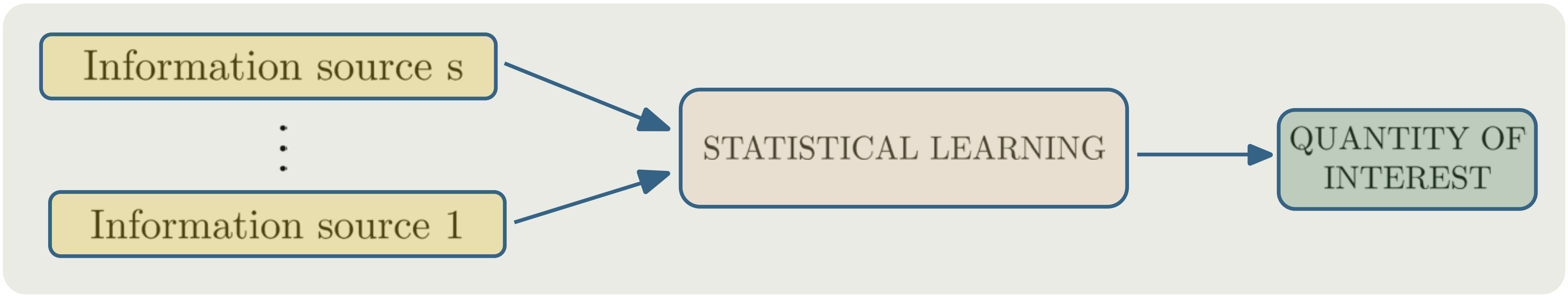


increasing fidelity

Model s → $Y_s(\boldsymbol{x};\boldsymbol{\xi})$ slow

⋮

Model 1 → $Y_1(\boldsymbol{x};\boldsymbol{\xi})$ fast



Information source s

⋮

Information source 1

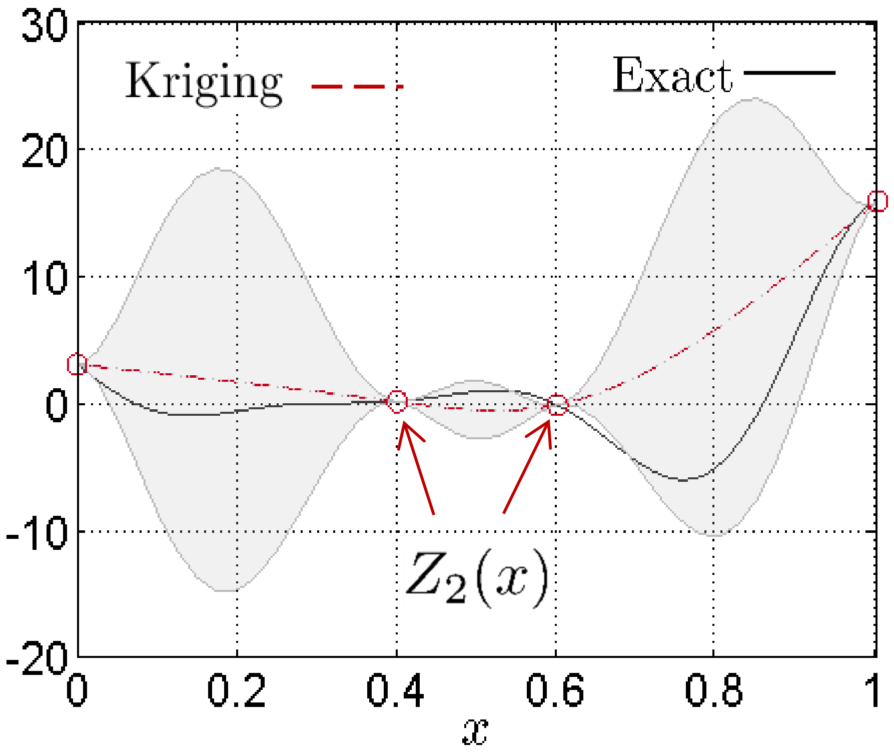
STATISTICAL LEARNING

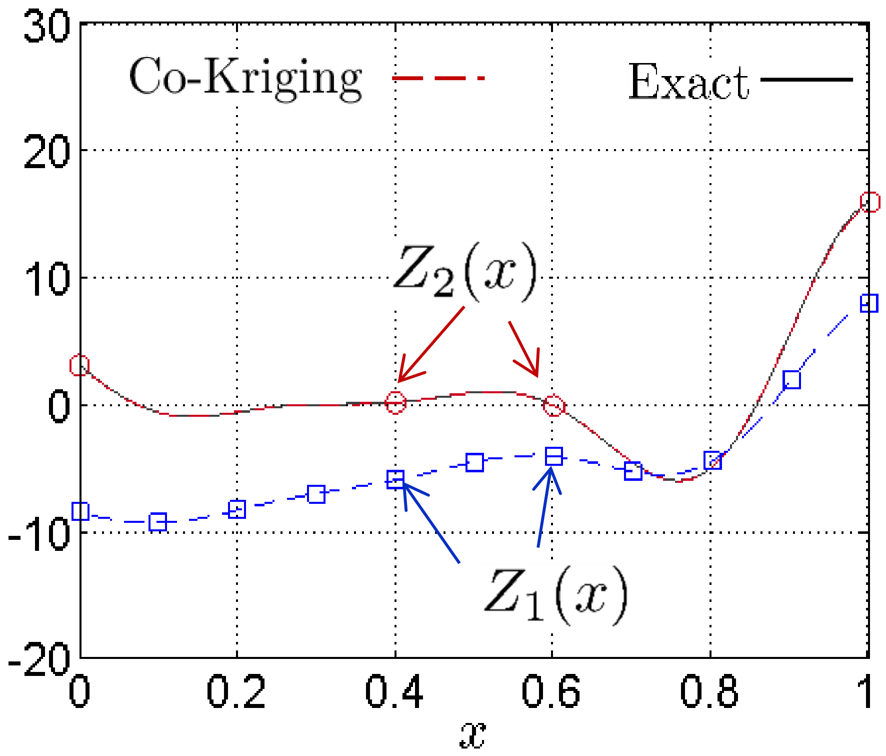
QUANTITY OF INTEREST

Number of runs is limited by time
and computational resources

We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling

Predicting the Output from a Complex Computer Code When Fast Approximations Are Available

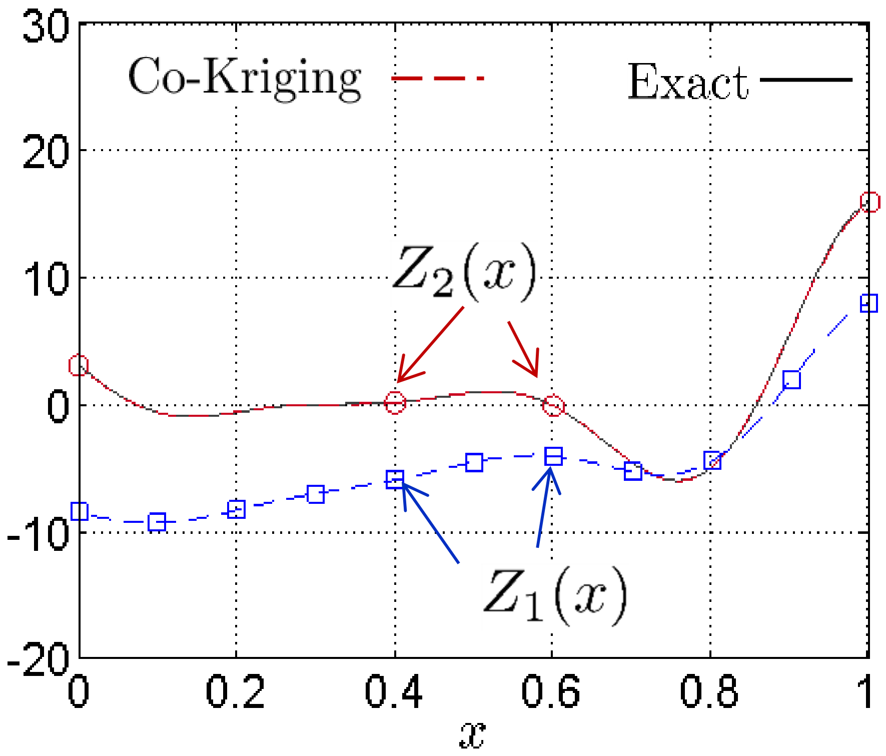
M. C. Kennedy; A. O'Hagan

Biometrika, Vol. 87, No. 1. (Mar., 2000), pp. 1-13.

Auto-regressive model:

$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x}) f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x})$$

$$t = 1, \dots, s$$



Predictive posterior

$$p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2),$$

$$\mu_*(\boldsymbol{x}_*) = \boldsymbol{k}_{*N}(\boxed{\boldsymbol{K}} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y},$$

$$\sigma_*^2(\boldsymbol{x}_*) = \boldsymbol{k}_{**} - \boldsymbol{k}_{*N}(\boxed{\boldsymbol{K}} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{N*},$$

$$\begin{array}{c} N_1 \\ N_2 \end{array} \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

Block covariance matrix

Multi-fidelity modeling via recursive GPs

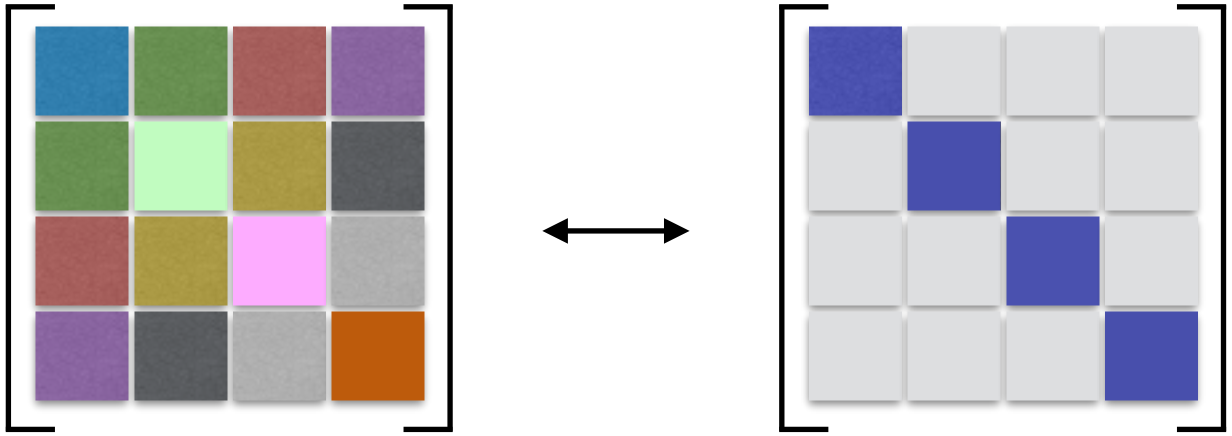
Key idea: Replace f_{t-1} with the GP posterior of the previous level $\tilde{f}_{t-1}$

$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x})\boxed{f_{t-1}(\mathbf{x})} + \delta_t(\mathbf{x}) \longrightarrow \tilde{f}_{t-1} \sim f_{t-1}|\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{t-1}$$

This allows for a static condensation procedure on the fully coupled covariance matrix yielding a decoupled problem, i.e. s independent GP regression problems.

Fully coupled*

Recursive**



Theorem (LeGratiet, 2014):

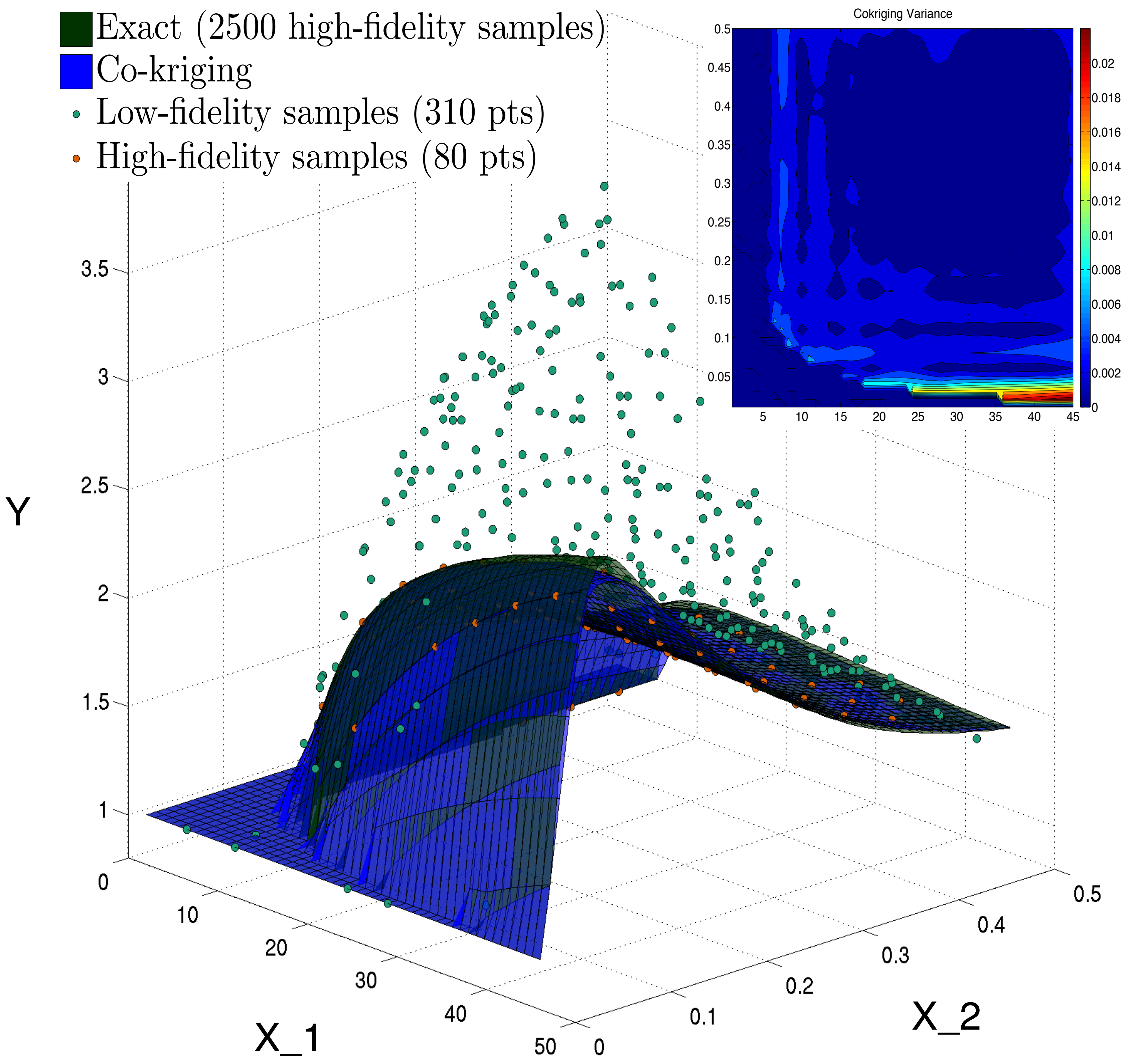
The predictive posterior of the recursive scheme has exactly the same distribution with the the fully coupled model given a nested experimental design.

Cost: $1 \times \left(\sum_{i=1}^{s} N_i \times \sum_{i=1}^{s} N_i\right)$ $s \times (N_t \times N_t)$

* M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

** L. Le Gratiet and J. Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5), 2014.

Example application: *Regression*



Using deep hierarchies

Goal: Develop multi-fidelity algorithms for learning general nonlinear correlations, hence extending the linear AR(1) scheme of Kennedy & O'Hagan.

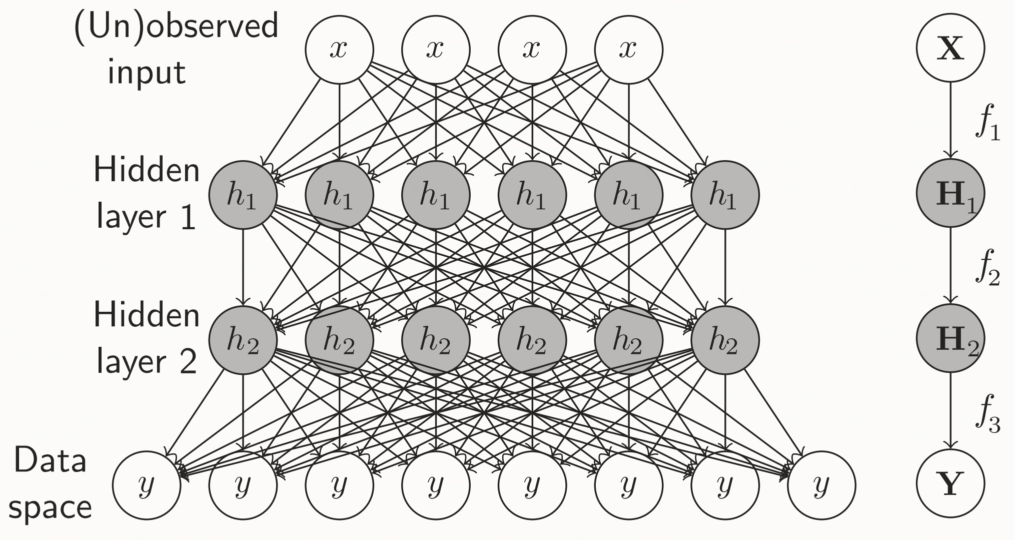
$$f_t(\boldsymbol{x}) = \rho f_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x})$$

classical AR(1)

···▶

$$f_t(\boldsymbol{x}) = g_{t-1}(f_{t-1}(\boldsymbol{x})) + \delta_t(\boldsymbol{x})$$

non-linear autoregression via deep GP priors



$\mathbf{Y} = f_3(f_2(\cdots f_1(\mathbf{X}))), \qquad \mathbf{H}_i = f_i(\mathbf{H}_{i-1}), \qquad f_i \sim \mathcal{GP}$

Technical approach:

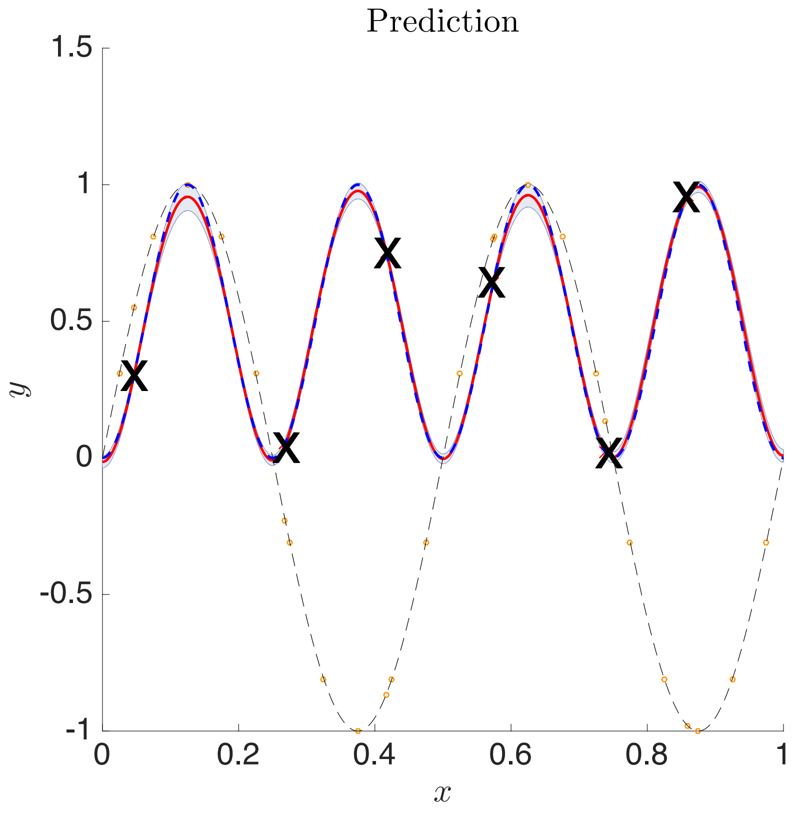
- Obtain a recursive inference scheme by conditioning on the GP posterior of the previous fidelity level
- Propagate uncertainty across fidelity levels using a variational re-formulation of Gaussian process regression that enables supervised learning with uncertain inputs
- The resulting predictive scheme generalizes the linear AR(1) model without increasing the computational complexity

Kennedy, M.C., and A. O'Hagan. "Predicting the output from a complex computer code when fast approximations are available." Biometrika 87.1 (2000): 1-13.
Damianou, A.C., and N.D. Lawrence. "Deep gaussian processes." arXiv preprint arXiv:1211.0358 (2012).
Girard, A., et al. "Gaussian process priors with uncertain inputs? application to multiple-step ahead time series forecasting." (2003).

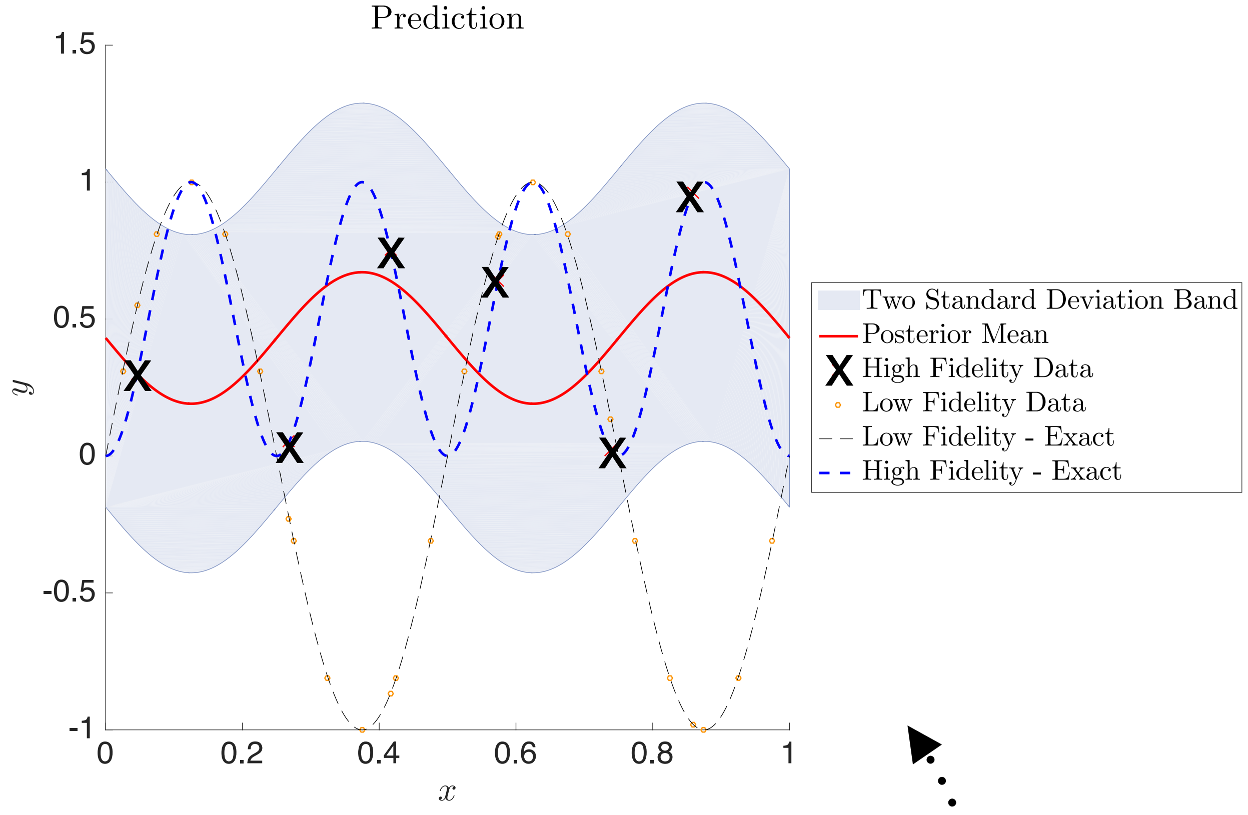
Multi-fidelity modeling using deep networks

1D Example:

$$f_{LF} = \sin(4\pi x)$$
$$f_{HF} = f_{LF}^2,$$

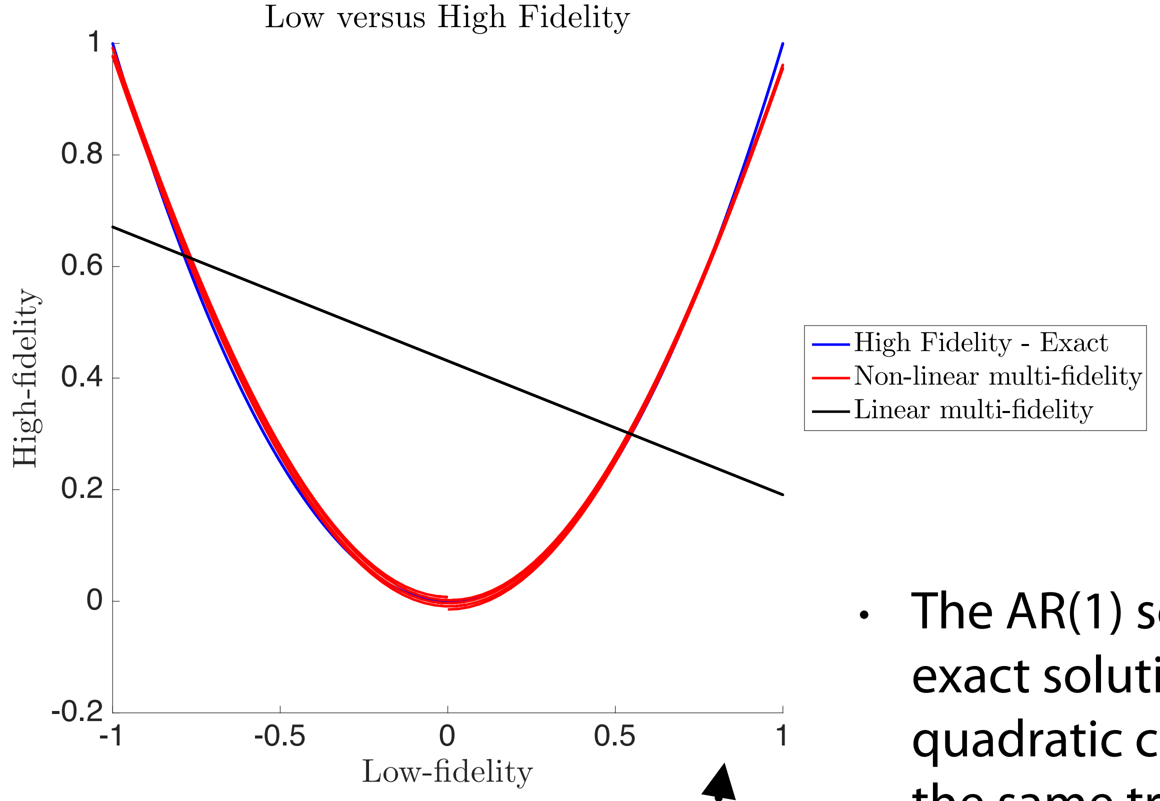


(a)



(b)

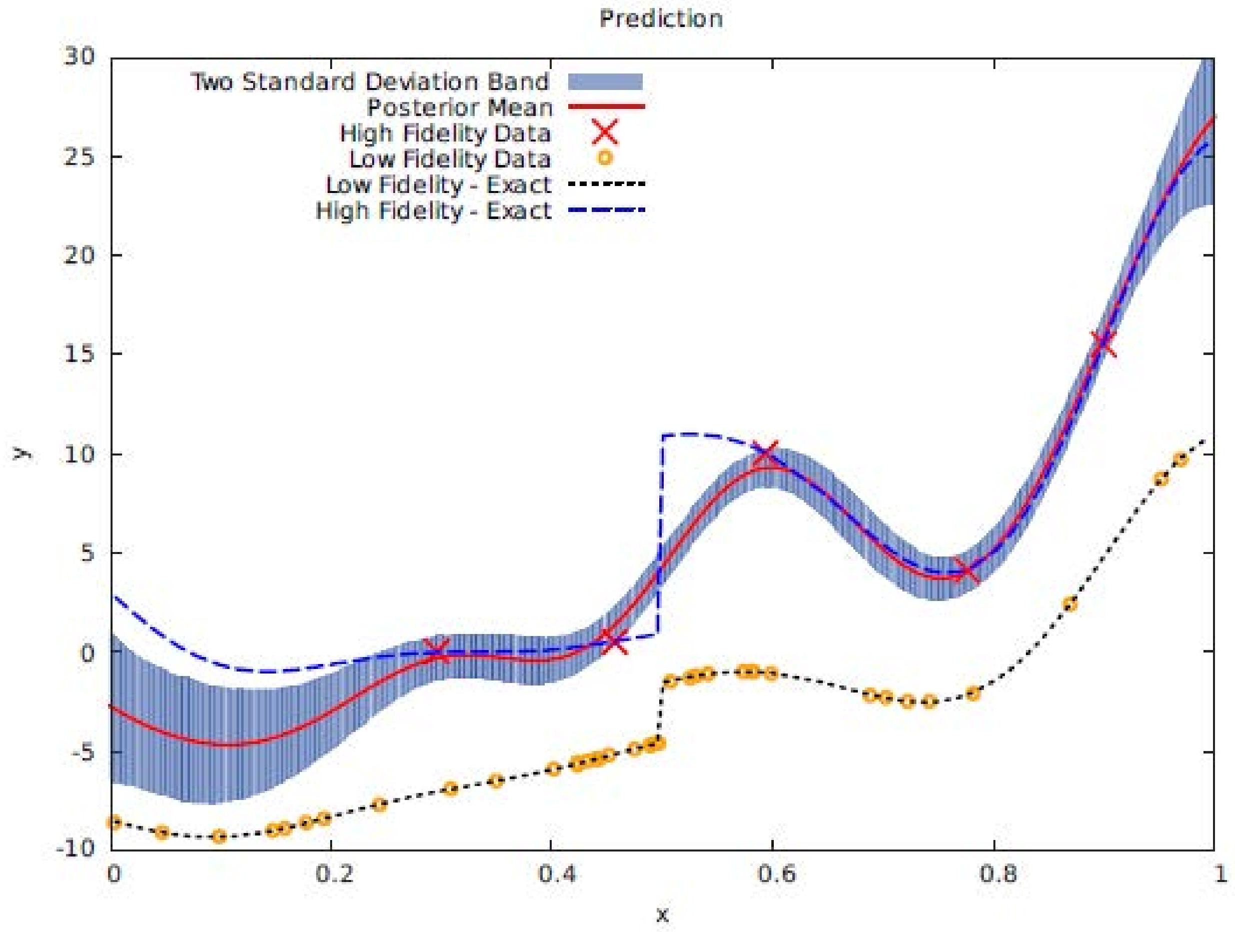
- The deep multi-fidelity predictor is able to capture the exact solution and recover the quadratic correlation structure using only 5 high-fidelity observations
- Notice how around x=0.4, 0.85 it captures the right trend in the exact solution, despite the fact that the low-fidelity data is suggesting the opposite.

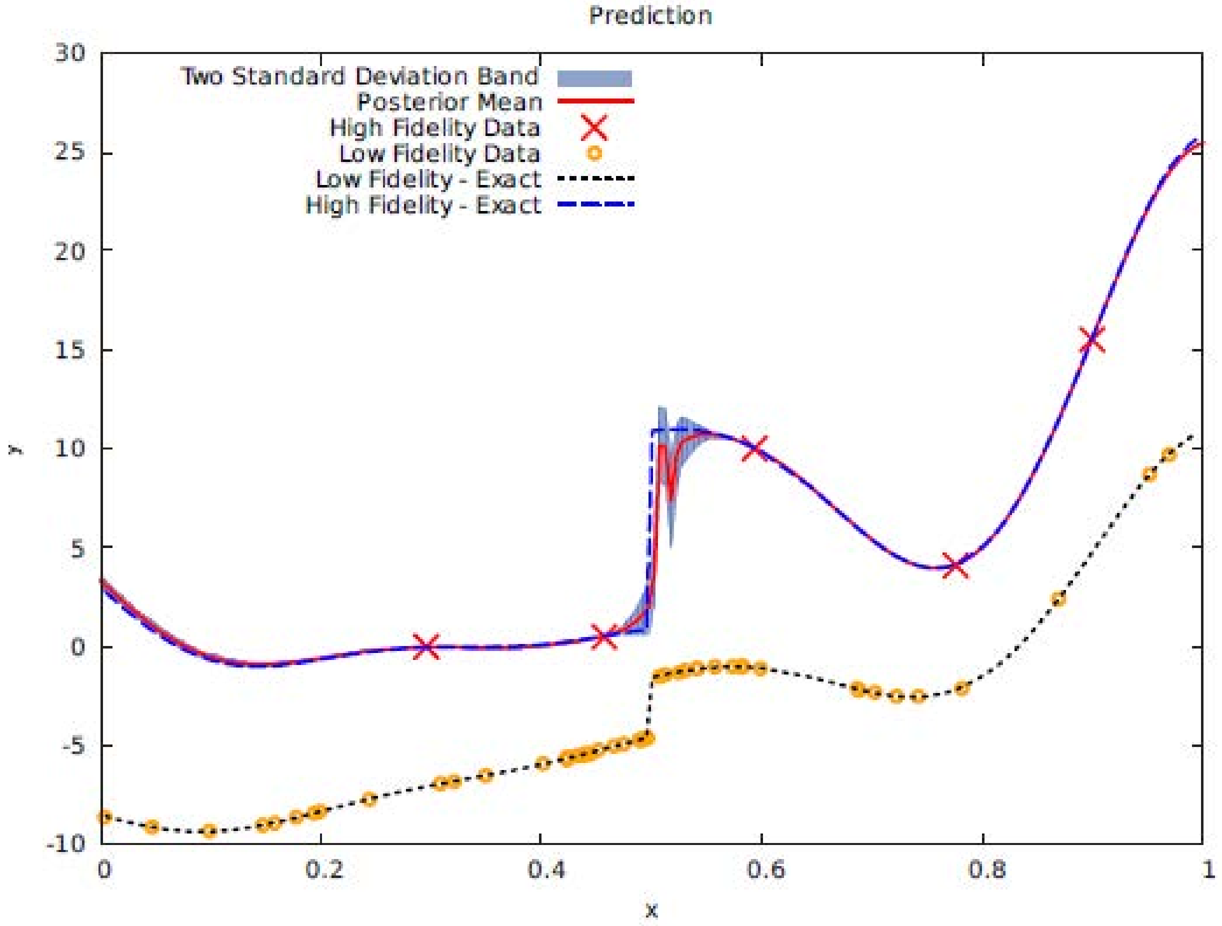


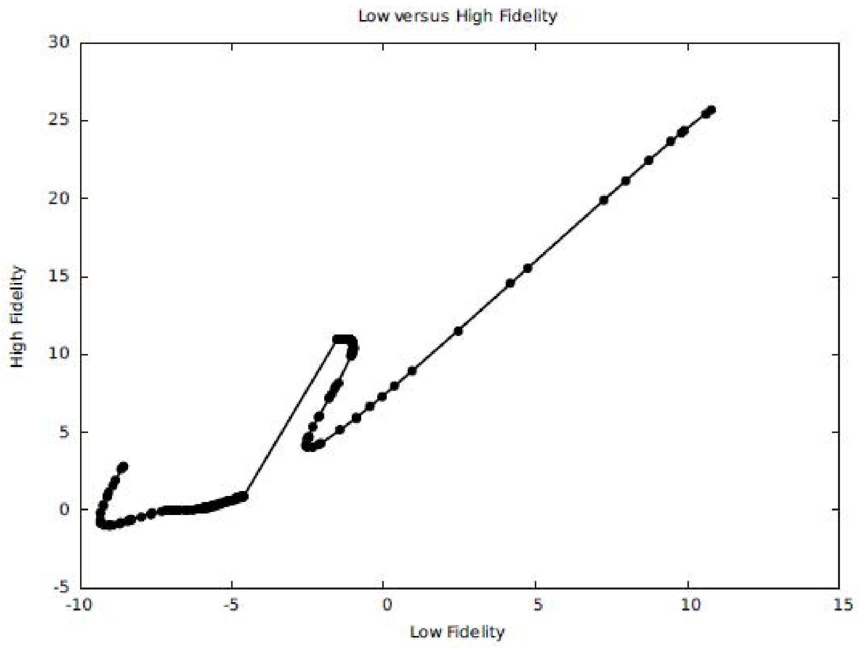
(c)

- The AR(1) scheme ***fails*** to capture the exact solution and recover the quadratic correlation structure using the same training set

Multi-fidelity modeling using deep networks







$$\begin{bmatrix} f_1(h) \\ f_2(h) \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_1(h,h') & \rho k_1(h,h') \\ \rho k_1(h,h') & \rho^2 k_1(h,h') + k_2(h,h') \end{bmatrix} \right)$$

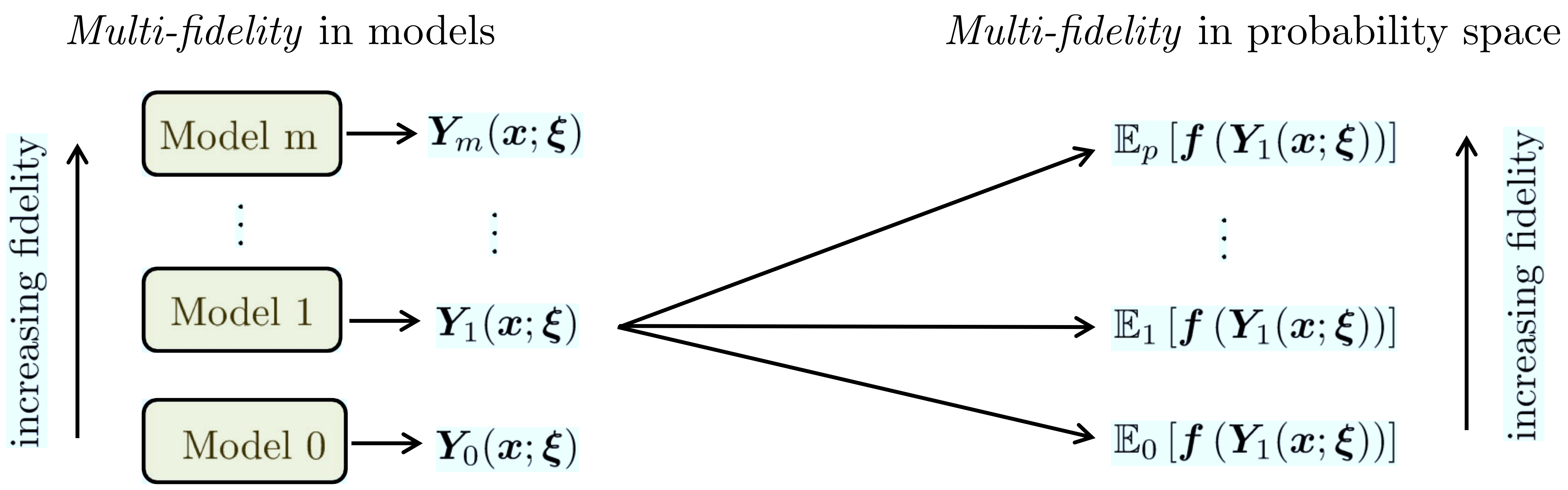
where

$$x \longmapsto h := h(x) \longmapsto \begin{bmatrix} f_1(h(x)) \\ f_2(h(x)) \end{bmatrix}.$$

The high fidelity code is modeled by $f_2(h(x))$ and the low fidelity one by $f_1(h(x))$.

M. Raissi, and G.E. Karniadakis. "Deep Multi-fidelity Gaussian Processes." arXiv preprint arXiv:1604.07484 (2016).

Multi-fidelity in physical models and in probability space



Multi-fidelity in models

Multi-fidelity in probability space

increasing fidelity

Model m $\longrightarrow \boldsymbol{Y}_m(\boldsymbol{x};\boldsymbol{\xi})$

$\vdots \qquad \vdots$

Model 1 $\longrightarrow \boldsymbol{Y}_1(\boldsymbol{x};\boldsymbol{\xi})$

Model 0 $\longrightarrow \boldsymbol{Y}_0(\boldsymbol{x};\boldsymbol{\xi})$

$\mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_1(\boldsymbol{x};\boldsymbol{\xi})\right)\right]$

$\vdots$

$\mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_1(\boldsymbol{x};\boldsymbol{\xi})\right)\right]$

$\mathbb{E}_0\left[\boldsymbol{f}\left(\boldsymbol{Y}_1(\boldsymbol{x};\boldsymbol{\xi})\right)\right]$

increasing fidelity

PROCEEDINGS OF THE ROYAL SOCIETY A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

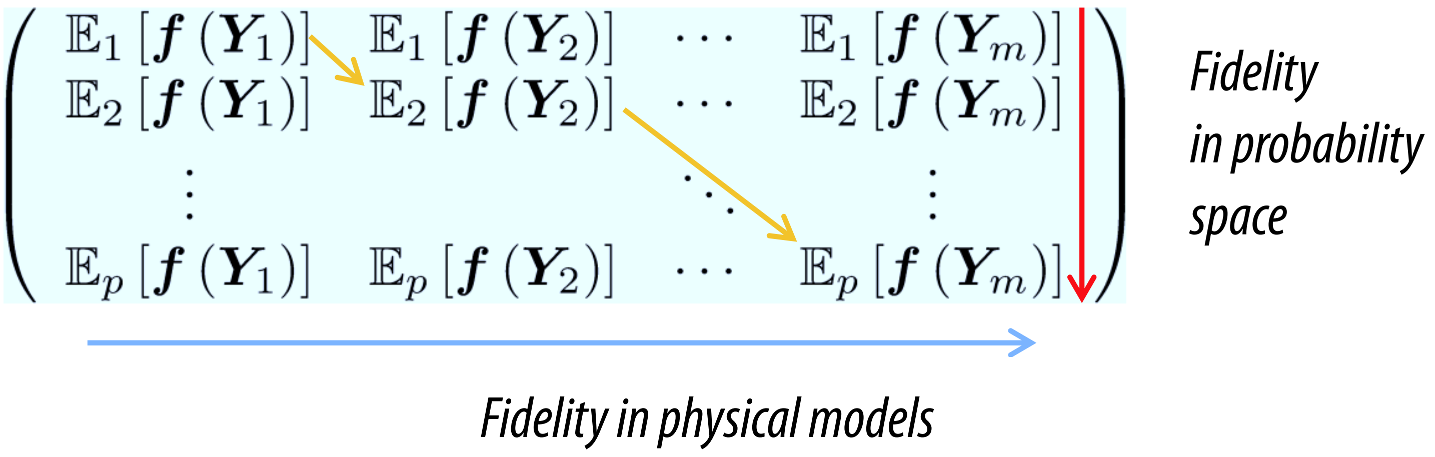
Multi-fidelity modeling via recursive co-kriging and Gaussian Markov random fields

P. Perdikaris[1], D. Venturi[1], J.O. Royset[2], and G.E. Karniadakis[1]

[1]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

[2]Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, USA

$$\mathbb{E}_{k+1}[f(\mathbf{Y}_l(\mathbf{x};\xi))] = \rho_{k+1}\mathbb{E}_k[f(\mathbf{Y}_l(\mathbf{x};\xi))] + \delta_{k+1}(\mathbf{x}), \quad k \leq p, \quad l \leq m$$



$$\begin{pmatrix} \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \\ \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \\ \vdots & & \ddots & \vdots \\ \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \end{pmatrix}$$

Fidelity in probability space

Fidelity in physical models

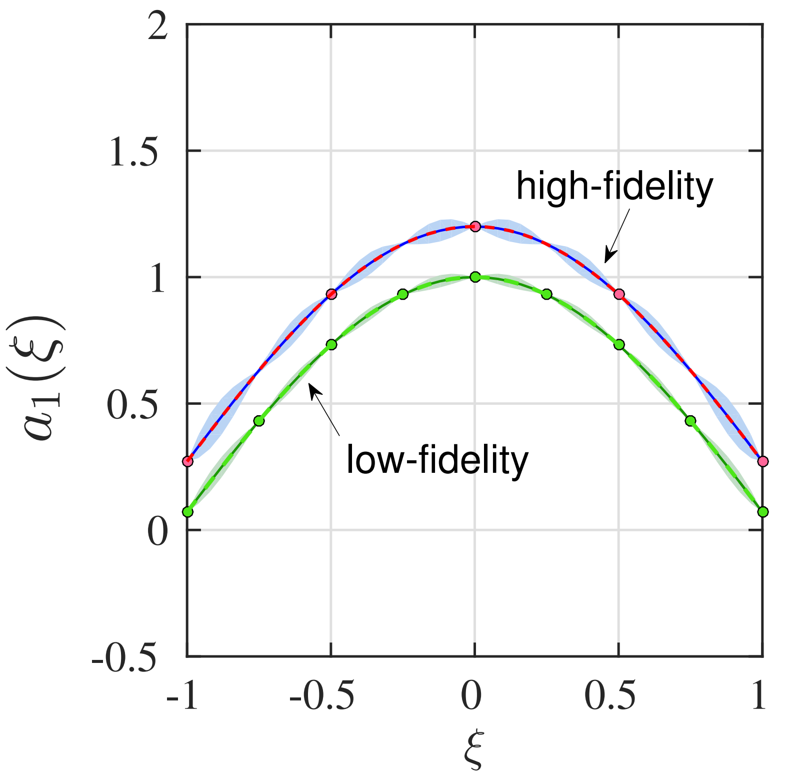
Extensions to vector-valued outputs

The Bayesian framework can be extended to predict

random fields of the form $u(\boldsymbol{x},\boldsymbol{\xi}) \cong \sum_{i=1}^{k} a_i(\boldsymbol{\xi}) L_i(\boldsymbol{x}).$

by applying the ***multi-fidelity*** inference algorithms

to the vector of Galerkin coefficients $\boldsymbol{a}(\boldsymbol{\xi}) = [a_1(\boldsymbol{\xi})\cdots a_k(\boldsymbol{\xi})]$



Capturing cross-correlations between multiple outputs

Separable covariance structure: $\boldsymbol{C}_j(\boldsymbol{\xi},\boldsymbol{\xi}';\boldsymbol{\theta}_j) = r_j(\boldsymbol{\xi},\boldsymbol{\xi}'\boxed{;\boldsymbol{\theta}_j})\boxed{\boldsymbol{\Sigma}_j}$

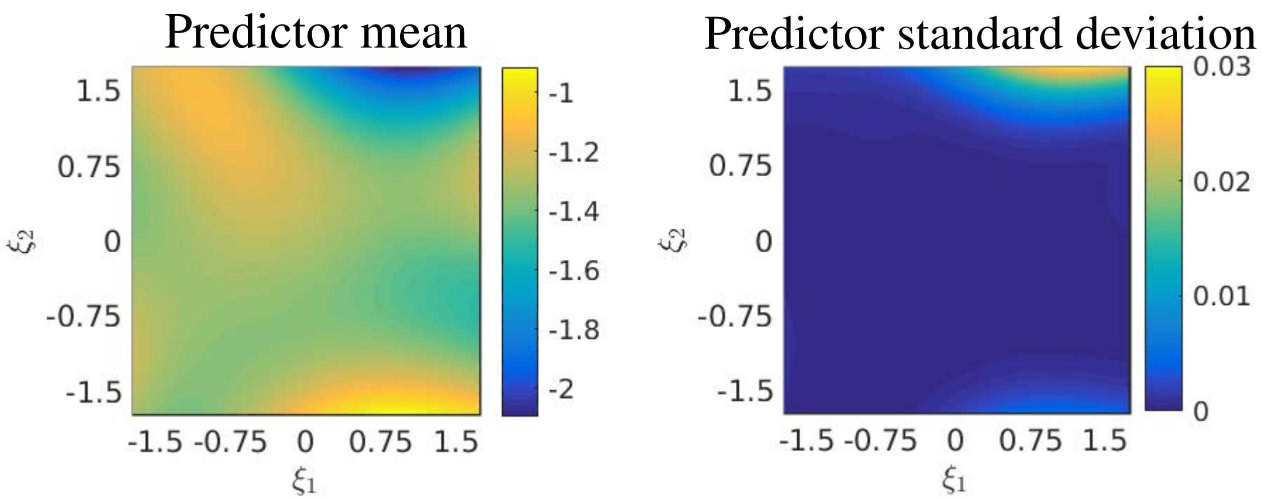
Linear model of coregionalization: $\boldsymbol{C}_j(\boldsymbol{\xi},\boldsymbol{\xi}';\boldsymbol{\theta}_j) = \boxed{\boldsymbol{B}}\,[\mathbf{diag}\,(r_1(\boldsymbol{\xi},\boldsymbol{\xi}'\boxed{;\boldsymbol{\theta}_1}),\ldots,r_k(\boldsymbol{\xi},\boldsymbol{\xi}'\boxed{;\boldsymbol{\theta}_k}))]\,\boxed{\boldsymbol{B}^T}$

learned from the data

Example: Stochastic Burgers equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \dfrac{1}{2}\dfrac{\partial^2 u}{\partial x^2} + f(x,t) \\ \text{Periodic B.C.} \\ u(x,0,\boldsymbol{\xi}) = u_0(x;\boldsymbol{\xi}) \end{cases}$$

$$u(x,t,\xi_1,\xi_2) = \sum_{q=-N/2}^{N/2} \underline{\underline{a_q(t,\xi_1,\xi_2)}}e^{iqx}$$



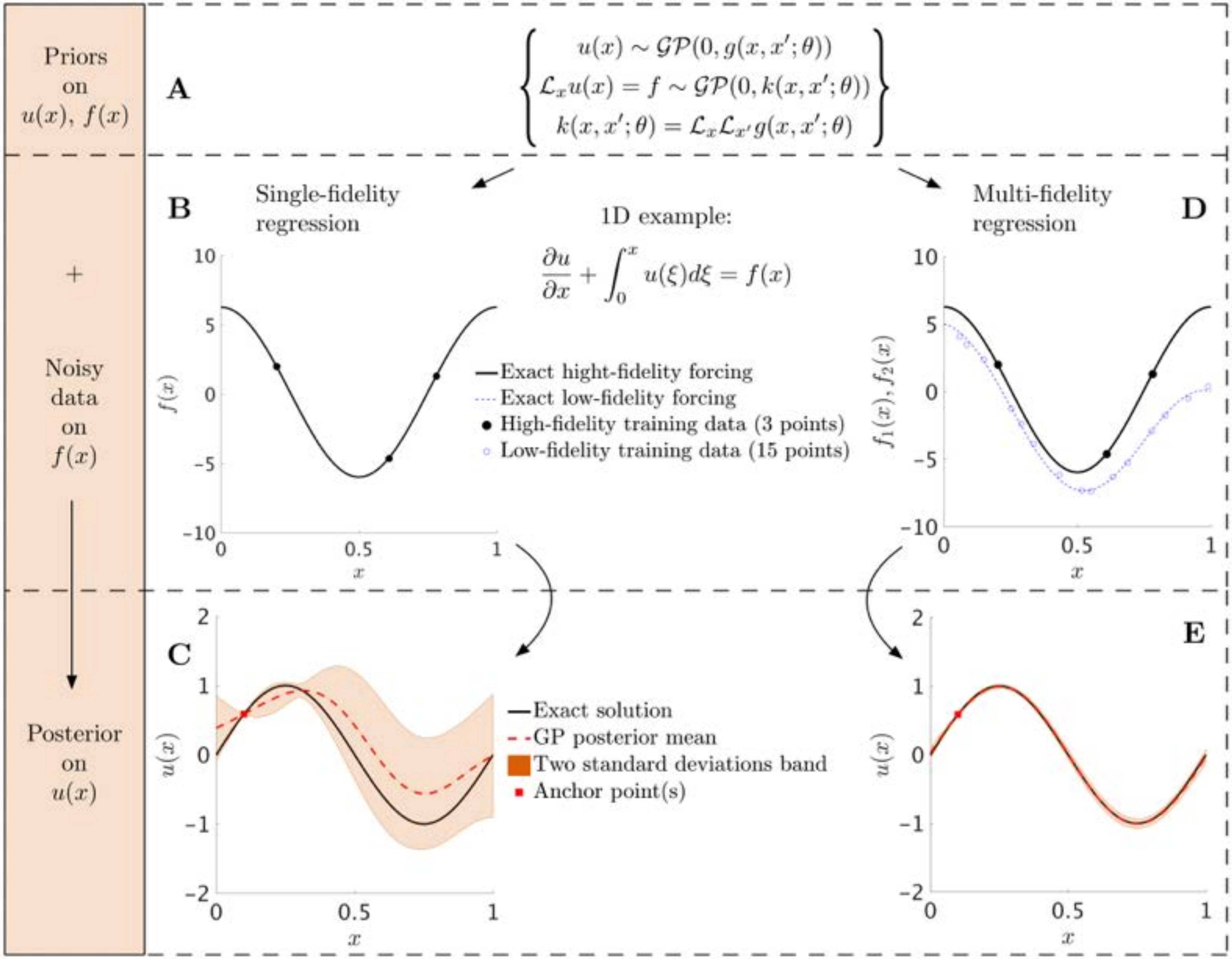
Inferred solution field at t=1

Multi-fidelity:

N=15, low-fidelity, 64 train. points
N=20, medium-fidelity, 29 train.points
N=60, high-fidelity, 11 train.points

Using the predicted coefficients & posterior variance we can reconstruct random fields with quantified uncertainty

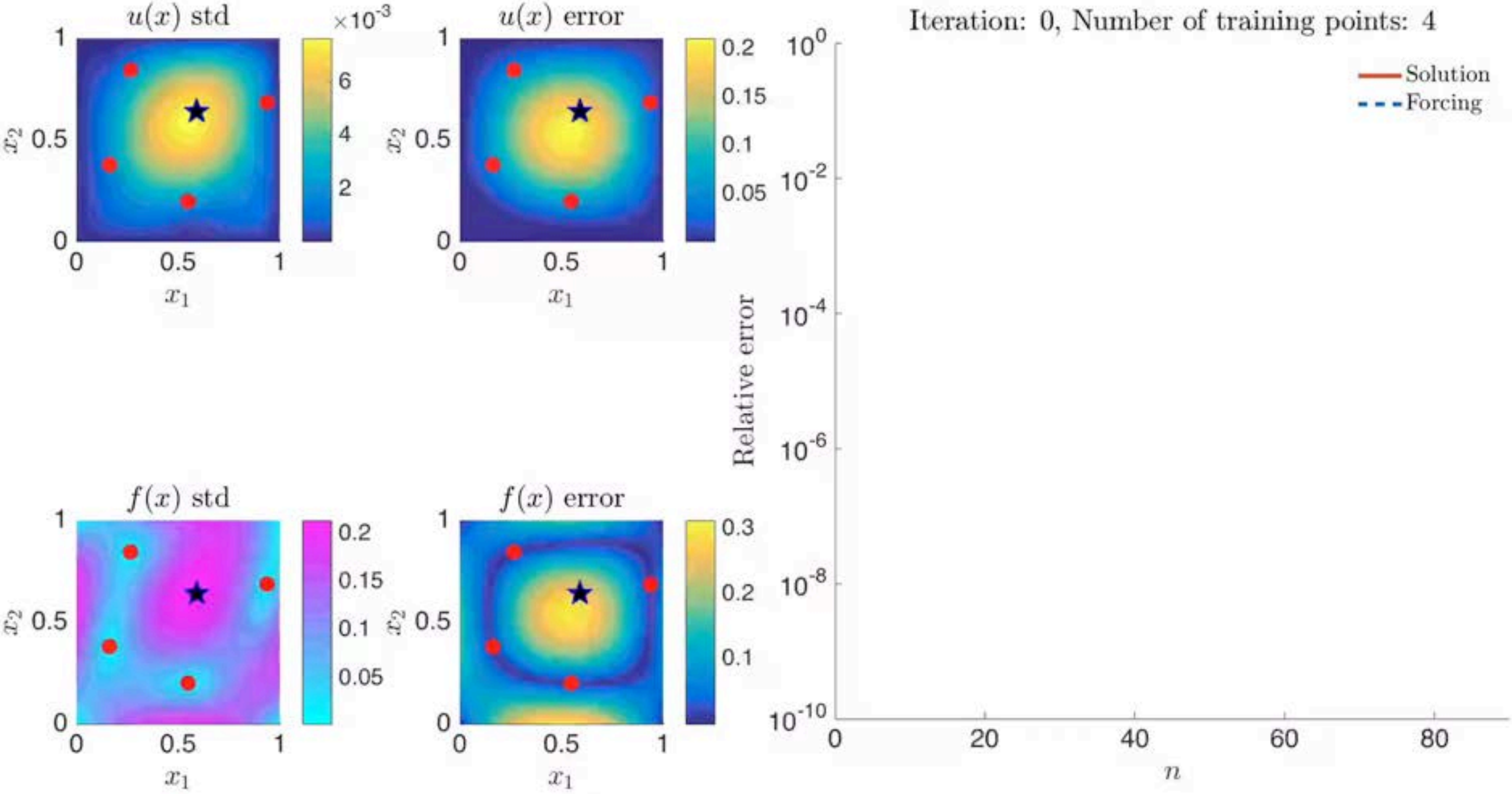
Example application: *Solution of linear differential equations*



Raissi, M., P. Perdikaris, and G.E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, http://128.84.21.199/abs/1607.04805, 2016

Example application: *Adaptive refinement via active learning*

$$\frac{\partial^2}{\partial x_1^2}u(x) + \frac{\partial^2}{\partial x_2^2}u(x) = f(x)$$



● denotes the training data actively collected by the scheme.

★ denotes the next sampling point suggested by the active learning scheme.

Raissi, M., P. Perdikaris, and G.E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, http://128.84.21.199/abs/1607.04805, 2016

Bayesian Optimization

BO provides a strategy to transform:

$$\mathbf{x}^\star = \min_{\mathbf{x}\in\mathbb{R}^d} ||f(\mathbf{x}) - y^\star|| \quad \text{(potentially intractable)}$$

into a series of problems:

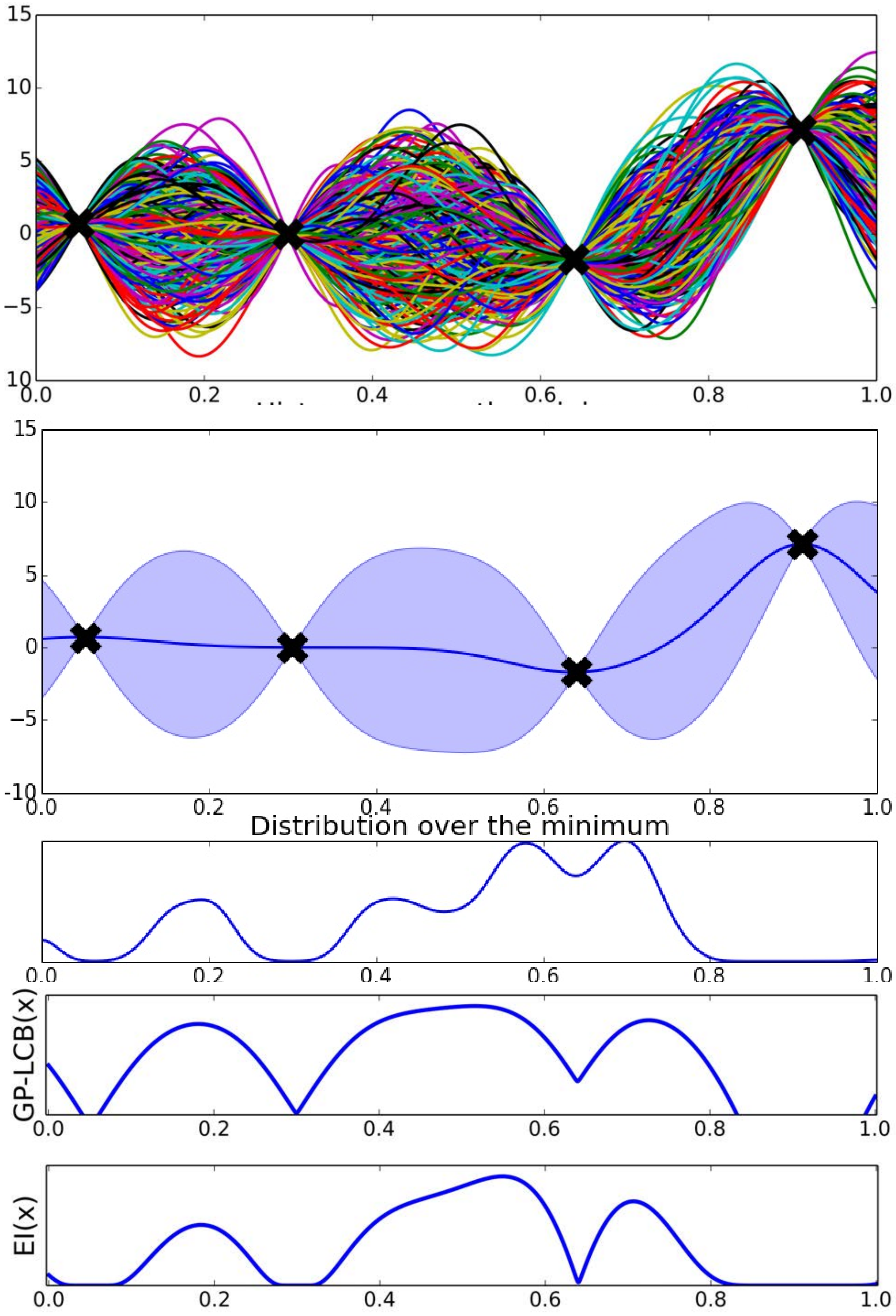
$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbb{R}^d} \alpha(\mathbf{x}; \mathcal{D}_n, \mathcal{M}_n)$$

where:

- The so called acquisition function is inexpensive to evaluate
- Acquisition function gradients are typically available
- Still a non-convex optimization but efficient solvers are available (DIRECT, CMA, gradient descent)

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.

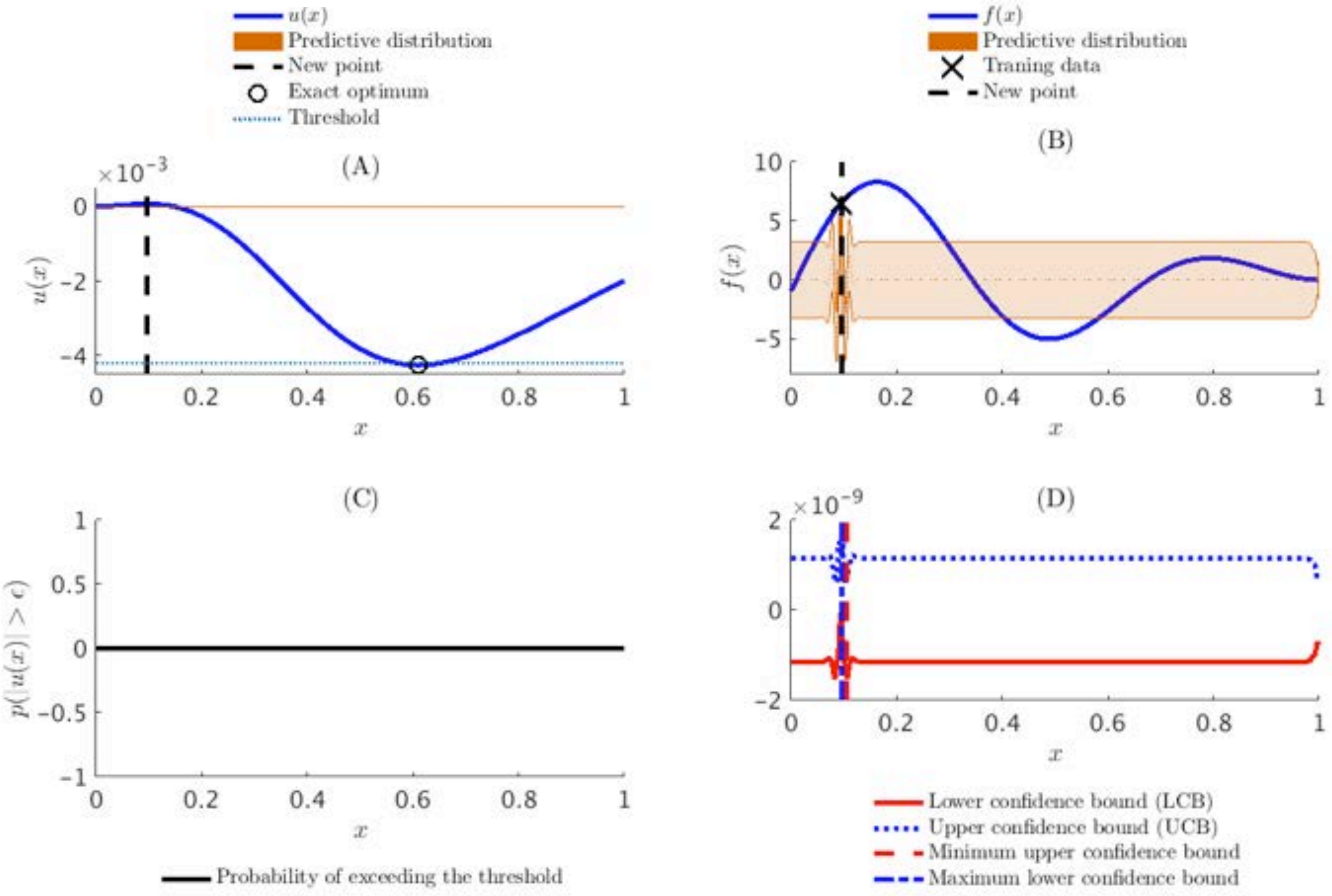


Jones, D. R. A taxonomy of global optimization methods based on response surfaces. Journal of global optimization 21:345–383, 2001.

Example application: *Probability of failure in linear elasticity*

$$\mathcal{L}_x u(x) := \frac{d^4}{dx^4} u(x) = f(x)$$
$$u(0) = u'(0) = 0$$
$$u''(1) = 0$$
$$u'''(1) = f(1)$$

1. Given noisy observations of the loading $f(x)$, solve for the displacement $u(x)$.
2. Find the maximum displacement $|u(x)|$.
3. Given the threshold ϵ, find the probability of failure.



Multi-fidelity Bayesian optimization

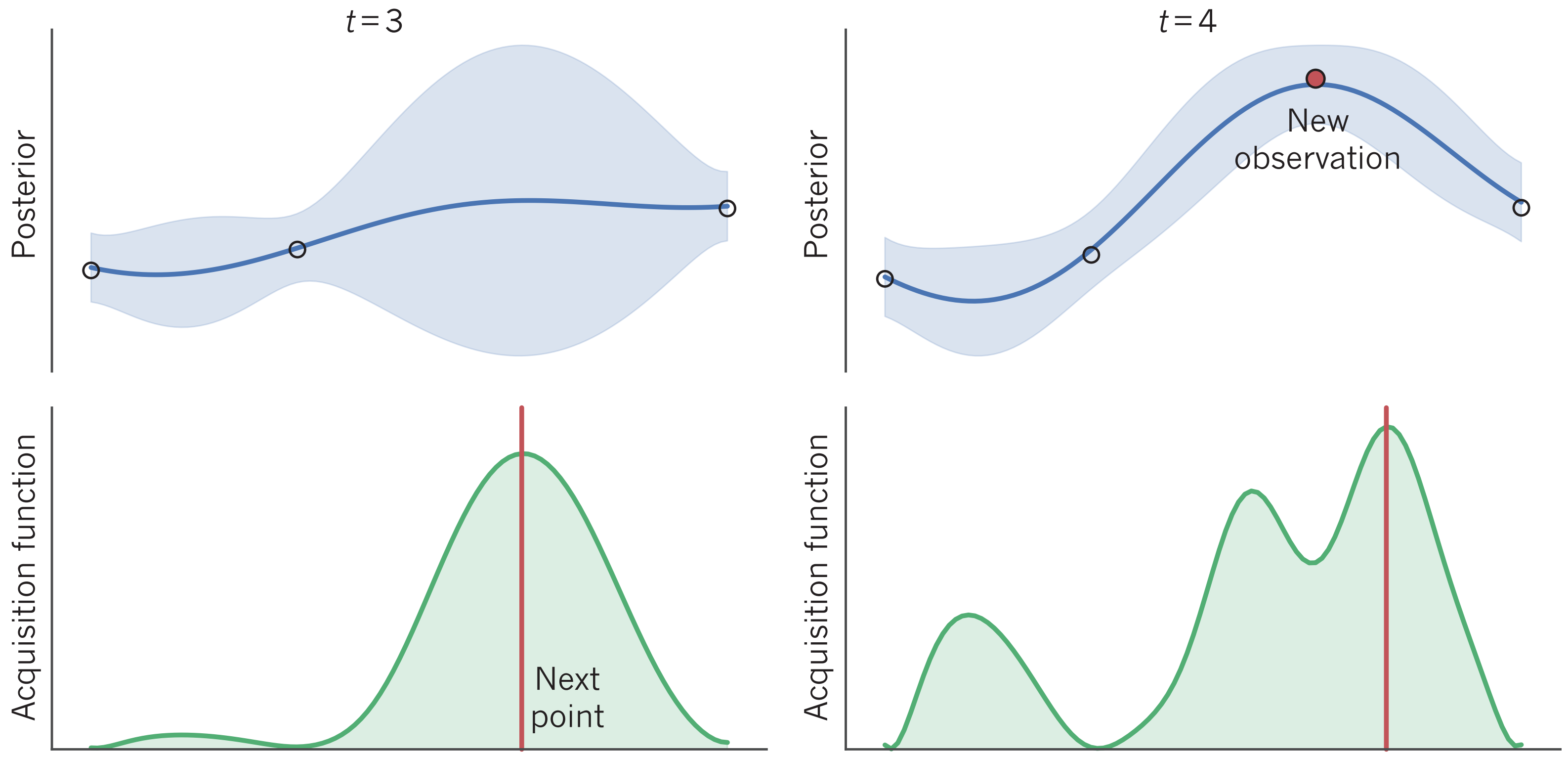
Goal: Identify a set of parameters that generates a response matching a target performance $y^\star$

$$\min_{\mathbf{x}\in\mathbb{R}} ||f(\mathbf{x}) - y^\star||$$

Idea: We model the response of a system using deep multi-fidelity surrogates

$$y = f_t(f_{t-1}(...(f_1(\mathbf{x})))), \quad f_i \sim \mathcal{GP}(\mu_i(\mathbf{x}), \Sigma_t)$$

Then the surrogate posterior distribution along with an acquisition function suggest a sampling plan than balances exploration vs exploitation towards identifying a global optimum



Example: 1D function maximization

P. Perdikaris, and G.E Karniadakis. "Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond." J. R. Soc. Interface (2016)

Calibration of blood flow simulations

Goal:

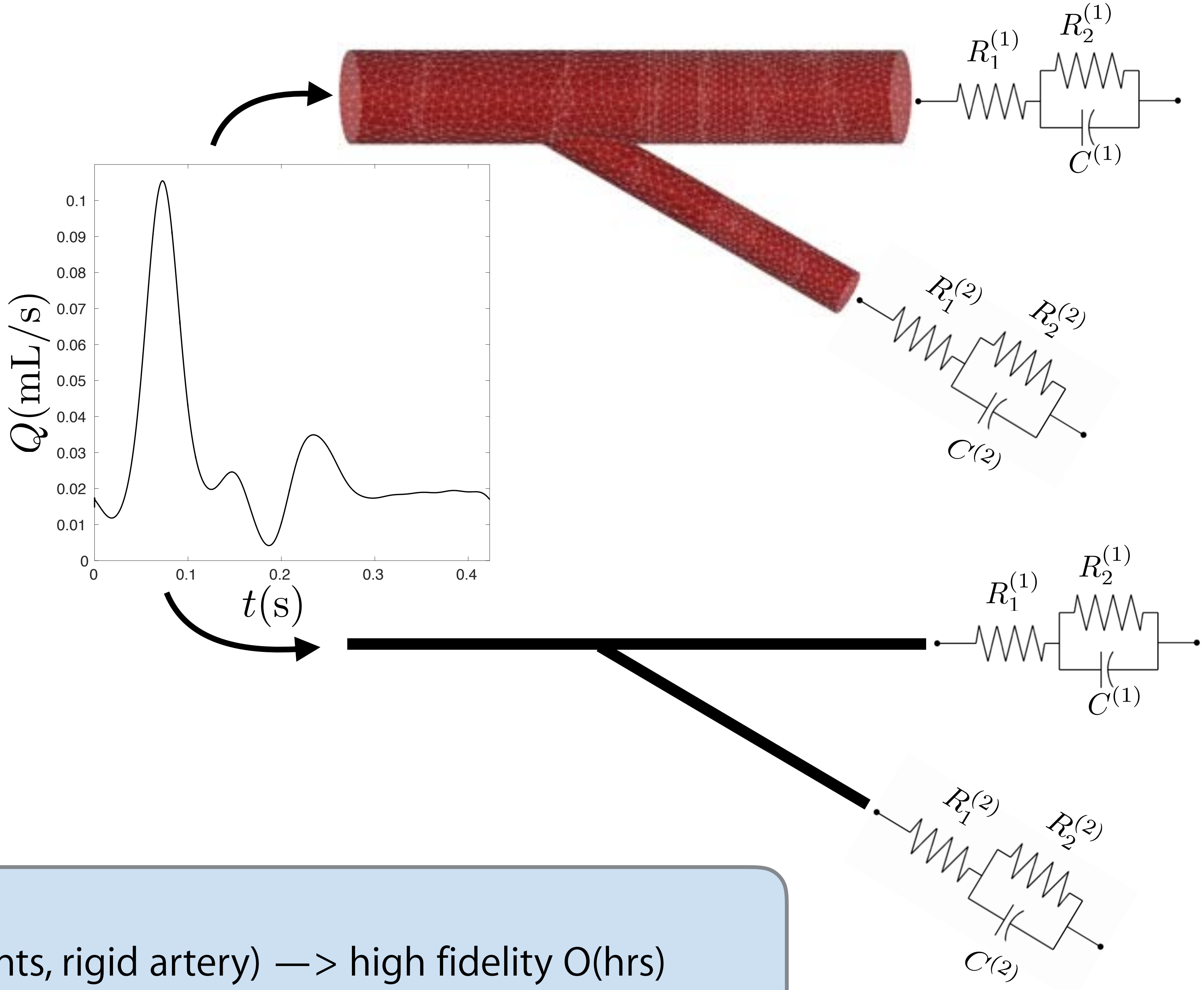
Calibrate the outflow boundary condition parameters to match a target inlet systolic pressure, i.e.,

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} |p_s^\star - p_s(\mathbf{x})|^2,$$

$$\mathbf{x} = [R_T^{(1)}, R_T^{(2)}]$$

$$\mathcal{X} = [10^{10}, 10^{11}] \times [10^{11}, 10^{12}]$$

$$p_s^\star = 47\text{mmHg}$$

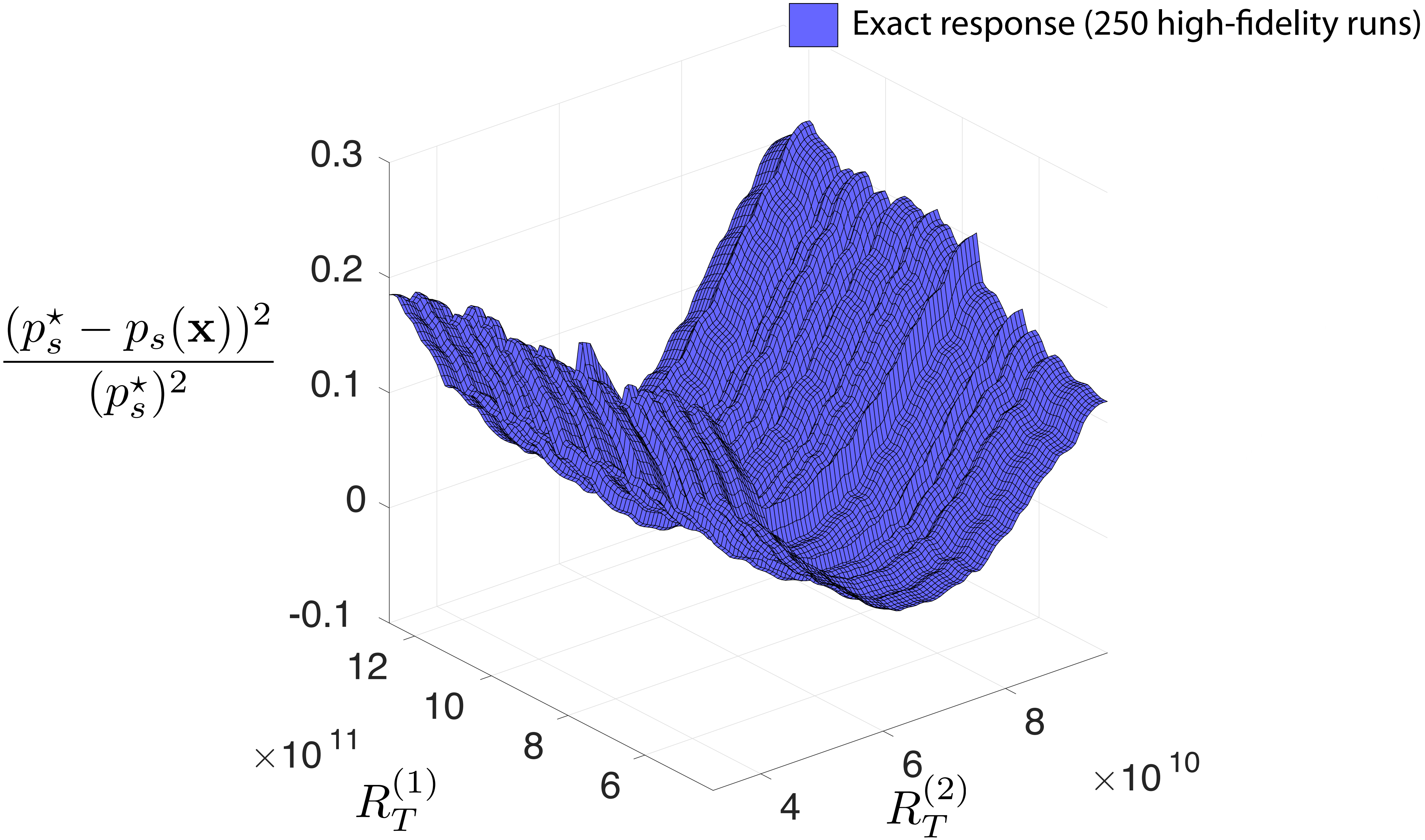


Multi-fidelity approach:

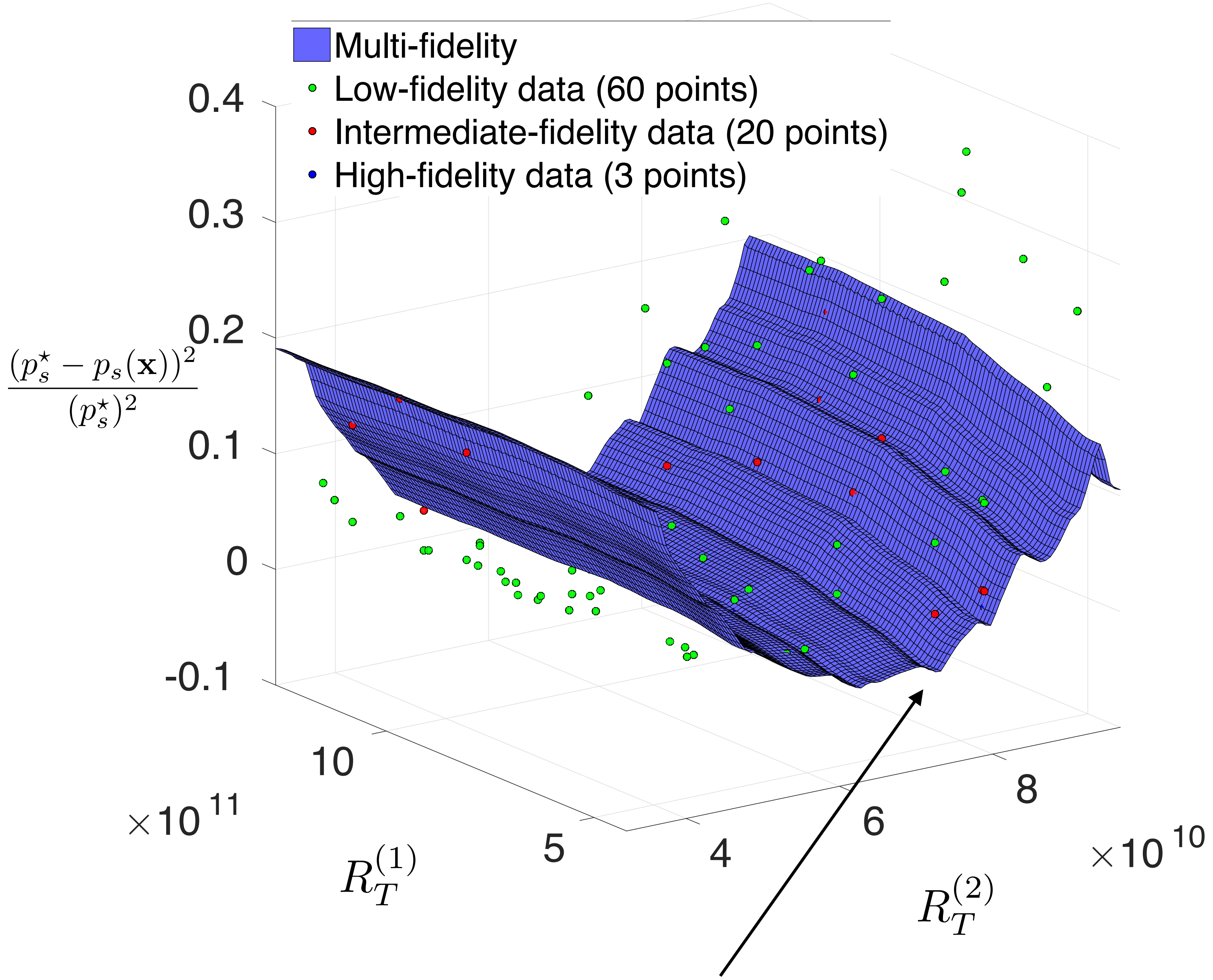
1.) 3D Navier-Stokes (spectral/hp elements, rigid artery) —> high fidelity O(hrs)
2.) Non-linear 1D-FSI (DG, compliant artery) —> intermediate fidelity O(mins)
3.) Linearized 1D-FSI solver around an inaccurate reference state —> low fidelity O(s)

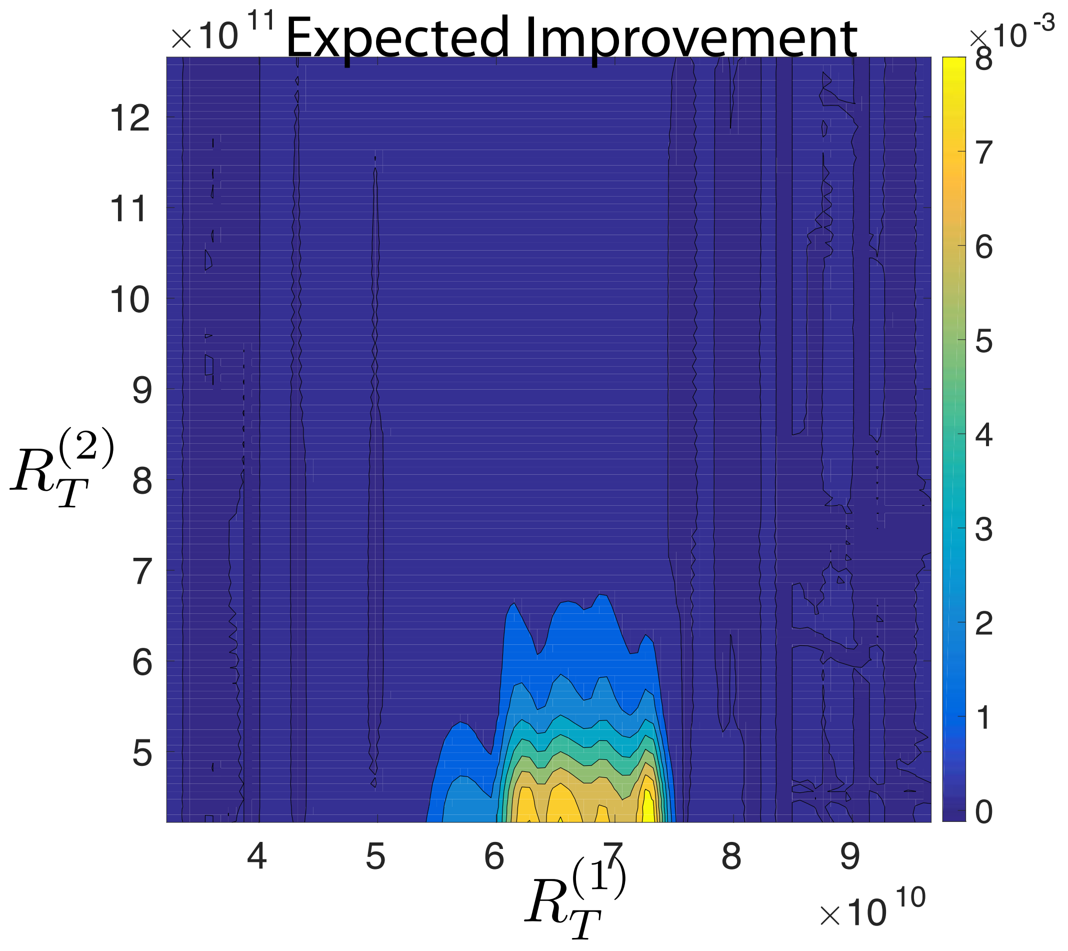
P. Perdikaris, G.E. Karniadakis Model inversion via multi-fidelity Bayesian Optimization: A new paradigm for parameter estimation in hemodynamics, and beyond (under review), 2016

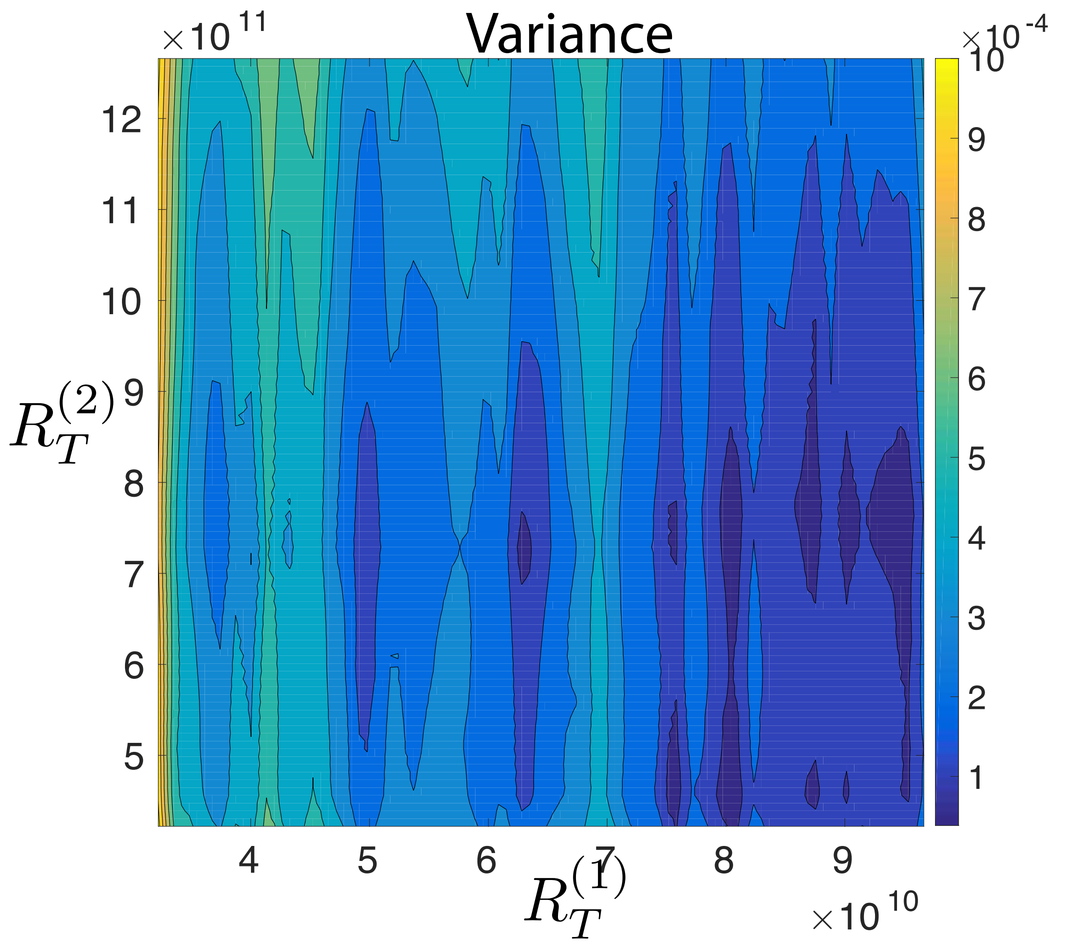
Calibration of blood flow simulations



Calibration of blood flow simulations







Decreased the relative error to $\mathcal{O}(10^{-3})$ after 3 iterations of BO, mainly sampling the lowest fidelity (cheapest) solver.

Limitations, challenges & future directions

<u>Scalability:</u> GPs suffer from a cubic scaling with the data

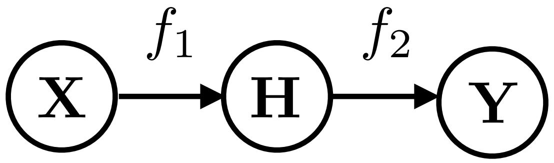
✓ Low-rank approximations to the covariance
Snelson, E., and Z. Ghahramani. "Sparse Gaussian processes using pseudo-inputs."

✓ Frequency-domain learning algorithms
De Baar, J. H. S., R.P. Dwight, and H. Bijl. "Speeding up kriging through fast estimation of the hyperparameters in the frequency-domain."

✓ Stochastic variational inference
Hensman, J., N. Fusi, and N.D. Lawrence. "Gaussian processes for big data."

<u>Discontinuities and non-stationarity:</u> GPs struggle to model discontinuous data

✓ Use warping functions to transform into a jointly stationary input space



- Log, sigmoid, betaCDF —> "*Warped GPs*" *Snelson, E., C.E. Rasmussen, and Z.Ghahramani. "Warped gaussian processes."*
- Neural networks —> "*Manifold GPs*" *Calandra, R., et al. "Manifold Gaussian processes for regression."*
- Gaussian processes —> "*Deep GPs*" *Damianou, A. C., and N.D. Lawrence. "Deep gaussian processes."*

<u>High-dimensions:</u> Tensor product kernels suffer from the curse of dimensionality, i.e. the require an exponentially increasing amount of training data

✓ Data-driven additive kernels
P. Perdikaris, D. Venturi, G.E. Karniadakis "Multi-fidelity information fusion algorithms for high dimensional systems and massive data-sets"

✓ Unsupervised dimensionality-reduction (*GPLVM*, *deep auto-encoders*)
Lawrence, N.D. "Gaussian process latent variable models for visualisation of high dimensional data."

Learning from big data

Bottlenecks:

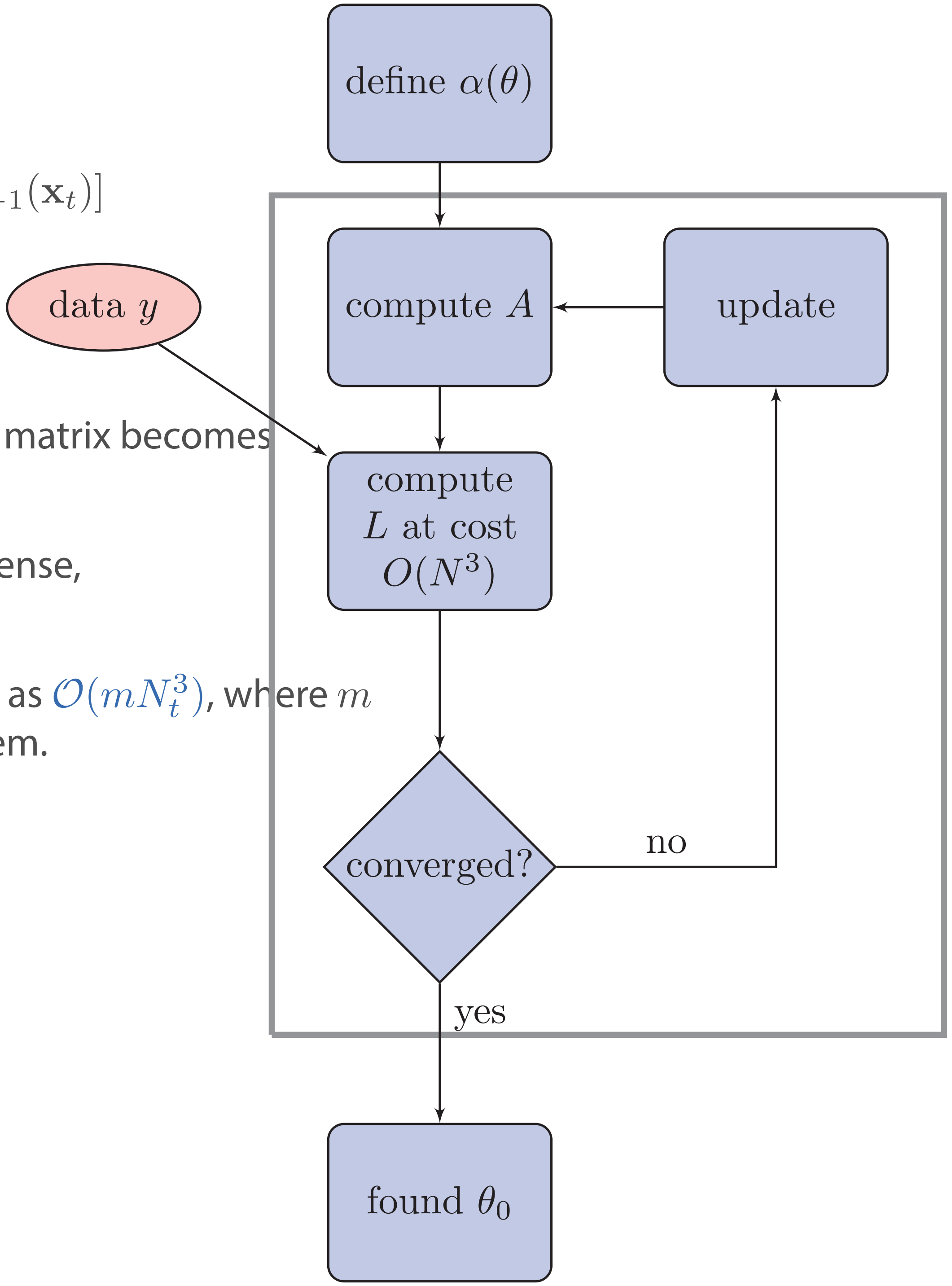
At each co-kriging level t maximize the likelihood of the observations y_t

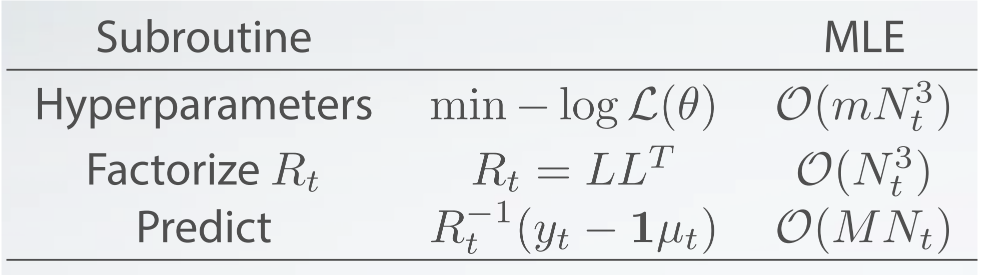
$$\min_{\{\mu_t,\sigma_t^2,\sigma_{\epsilon_t}^2,\rho_{t-1},\theta_t\}} \quad \frac{n}{2}\log(\sigma_t^2) + \frac{1}{2}\log|R_t(\theta_t) + \sigma_{\epsilon_t}^2 I| +$$

$$+\frac{1}{2\sigma_t^2}[y_t(\mathbf{x}_t) - \mathbf{1}_t\mu_t - \rho_{t-1}\hat{y}_{t-1}(\mathbf{x}_t)]^T[R_t(\theta_t) + \sigma_{\epsilon_t}^2 I]^{-1}[y_t(\mathbf{x}_t) - \mathbf{1}_t\mu_t - \rho_{t-1}\hat{y}_{t-1}(\mathbf{x}_t)]$$

We face the following challenges:

1. For small noise variance $\sigma_{\epsilon_t}^2$ and/or tightly clustered observations the correlation matrix becomes increasingly ill-conditioned.
2. Each iteration step for minimizing the log-likelihood requires the inversion of a dense, ill-conditioned $N_t \times N_t$ covariance matrix.
3. The total cost for estimating the hyper-parameters at each co-kriging level scales as $\mathcal{O}(mN_t^3)$, where m is the number of iterations required to solve the non-convex minimization problem.





Subroutine		MLE
Hyperparameters	$\min -\log \mathcal{L}(\theta)$	$\mathcal{O}(mN_t^3)$
Factorize R_t	$R_t = LL^T$	$\mathcal{O}(N_t^3)$
Predict	$R_t^{-1}(y_t - \mathbf{1}\mu_t)$	$\mathcal{O}(MN_t)$

Rasmussen, C. E. Gaussian processes for machine learning 2006.

O(N) learning algorithms

Wiener–Khinchin theorem:

$$S(\omega) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-2\pi\omega\tau} d\tau$$

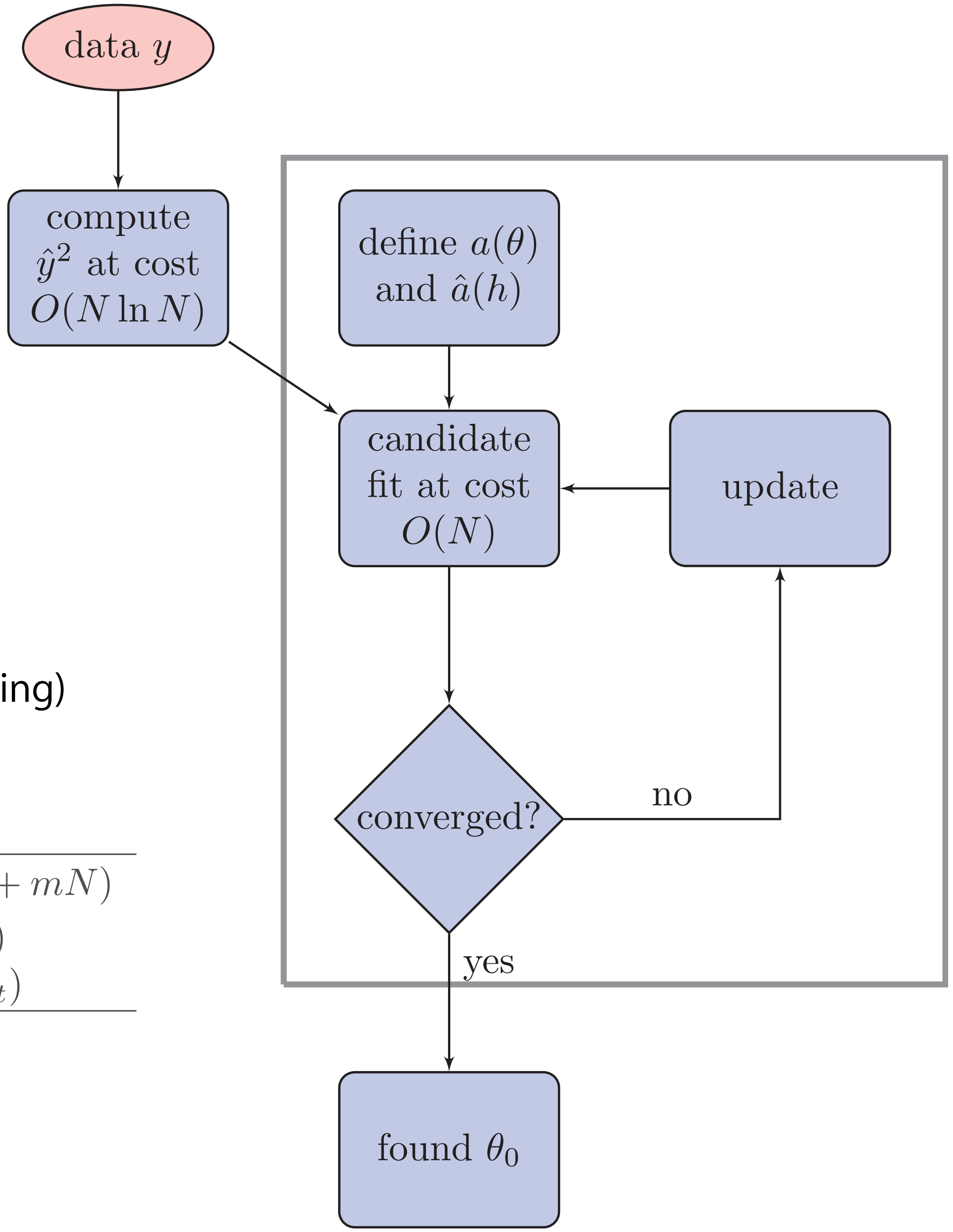
...i.e. the power spectral density of a wide-sense stationary process is the Fourier transform of its autocorrelation function.

We can speed up the hyperparamter estimation
by learning the sample variogram in the frequency domain:

$$\text{FSV} : \min_{\theta} \sum_{n=1}^{N} |\log \hat{y}_n^2 - \log \hat{r}(\theta)|^2$$

(Frequency Sample Variogram fitting)

Subroutine		MLE	GMRF	FSV
Hyperparameters	$\min -\log \mathcal{L}(\theta)$	$\mathcal{O}(mN_t^3)$	$\mathcal{O}(mN_t^{3/2})$	$\mathcal{O}(N_t \log N_t + mN)$
Factorize R_t	$R_t = LL^T$	$\mathcal{O}(N_t^3)$	$\mathcal{O}(N_t^{3/2})$	$\mathcal{O}(N_t^3)$
Predict	$R_t^{-1}(y_t - \mathbf{1}\mu_t)$	$\mathcal{O}(MN_t)$	$\mathcal{O}(MN_t)$	$\mathcal{O}(MN_t)$



data y

compute $\hat{y}^2$ at cost $O(N \ln N)$

define $a(\theta)$ and $\hat{a}(h)$

candidate fit at cost $O(N)$

update

converged?

no

yes

found θ_0

De Baar, J. H. S., R. P. Dwight, and H. Bijl. "Speeding up kriging through fast estimation of the hyperparameters in the frequency-domain." Computers & Geosciences 54 (2013): 99-106.

High-dimensional kernel design

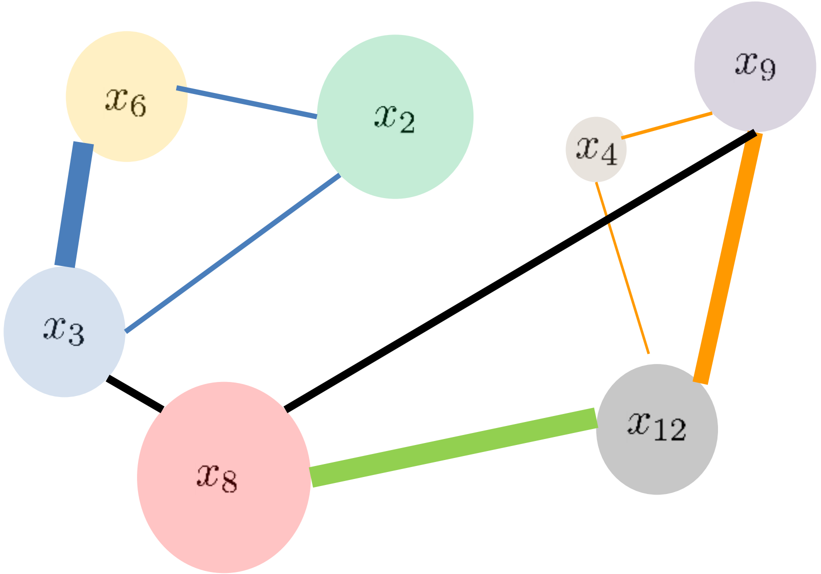
Given a set of scattered observations $y(\mathbf{x})$ we can construct a hierarchical functional representation of the form

$$y(\mathbf{x}) = y_0 + \sum_{1\leq i\leq d} y_i(x_i) + \sum_{1\leq i<j\leq d} y_{ij}(x_i,x_j) + \sum_{1\leq i<j<k\leq d} y_{ijk}(x_i,x_j,x_k) + \cdots$$

This facilitates the computation of sensitivity indices that characterize the active interactions in the data:

$$D_i = \int_0^1 y_i^2(x_i)dx_i \approx \int_0^1 \left[\sum_{r=1}^{k_i} \alpha_r^i \phi_r(x_i)\right]^2 dx_i = \sum_{r=1}^{k_i} (\alpha_r^i)^2$$

$$D_{ij} = \int_0^1\int_0^1 y_{ij}^2(x_i,x_j)dx_idx_j \approx \int_0^1\int_0^1 \left[\sum_{p=1}^{l_i}\sum_{q=1}^{l'_j} \beta_{pq}^{ij}\phi_p(x_i)\phi_q(x_j)\right]^2 dx_idx_j = \sum_{p=1}^{l_i}\sum_{q=1}^{l'_j}(\beta_{pq}^{ij})^2$$



Maximal cliques ($N_{\mathcal{C}} = 5$):

$\mathcal{C}_1 = \{x_2, x_3, x_6\}$

$\mathcal{C}_2 = \{x_3, x_8\}$

$\mathcal{C}_3 = \{x_8, x_9, x_{12}\}$

$\mathcal{C}_4 = \{x_4, x_9, x_{12}\}$

$\mathcal{C}_5 = \{x_1, x_5, x_7, x_{10}, x_{11}\}$ (inactive)

⇨

$$\kappa(\mathbf{x},\mathbf{x}';\theta) = \sum_{q=1}^{N_{\mathcal{C}}} \kappa_q(\mathbf{x}_q, \mathbf{x}'_q;\theta_q),$$

Goal: Solve local low-dimensional FSV fitting problems to train the clique-wise kernels.

High dimensions and large data-sets

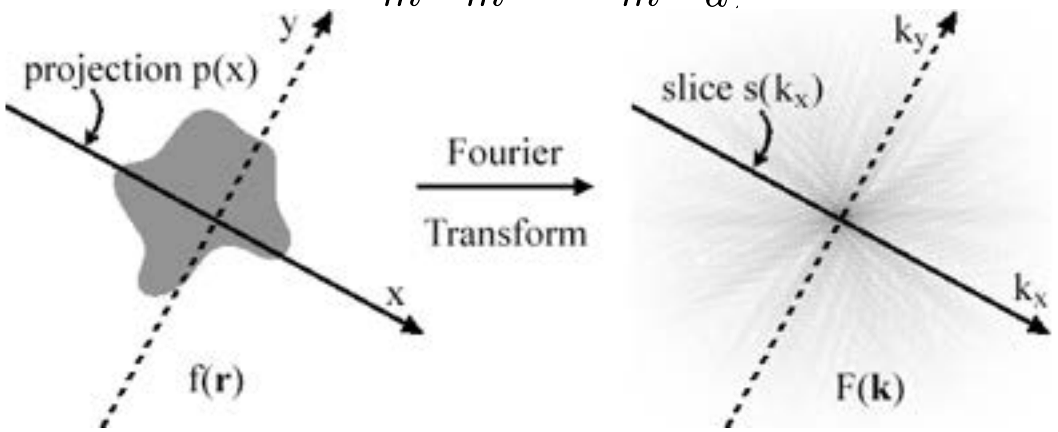
Problem: Inference with FSV fitting becomes intractable as it requires storage and operation on N^d frequencies

Step 1: Utilize the ANOVA expansion to project the data onto the sub-space defined by each maximal-clique, and identify the contribution of each maximal clique in the ***d***-dimensional power spectrum

$$\mathcal{P}_q y(x) = f_0 + \sum_{i \in \mathcal{C}_q} y_i(x_i) + \sum_{i,j \in \mathcal{C}_q} y_{ij}(x_i, x_j) + \sum_{i,j,k \in \mathcal{C}_q} y_{ijk}(x_i, x_j, x_k) + \cdots$$

Step 2: Use the Fourier projection-slice theorem to decompose the global high-dimensional optimization problem into local low-dimensional tasks.

$$\mathcal{F}_m \mathcal{P}_m = \mathcal{S}_m \mathcal{F}_d;$$



Now we can solve $N_{\mathcal{C}}$ FSV problems that involve N^m points, where $m = \mathbf{card}\{\mathcal{C}_q\} \ll d$, $1 \leq q \leq N_{\mathcal{C}}$.

$\Rightarrow$ Learning can be performed by training low-dimensional clique-wise kernels with $\mathcal{O}(N)$ cost!!

Perdikaris P., D. Venturi, G.E. Karniadakis Multi-fidelity information fusion algorithms for high dimensional systems and massive data-sets, SIAM J. Sci. Comput. (2016)

Forward UQ in a 100-dimensional PDE

Helmholtz equation in 2 input dimensions and 100 random variables:

$$\begin{cases} (\lambda^2 - \nabla^2)u(\mathbf{x};\omega) = f(\mathbf{x};\omega), \quad \mathbf{x} = (x,y), \quad \mathbf{x} \in \mathcal{D} = [0, 2\pi]^2, \\ u(\mathbf{x};\omega)|_{\partial\mathcal{D}} = 0, \\ f(\mathbf{x};\omega) = \dfrac{2}{d}\left\{ \displaystyle\sum_{i=1}^{d/4}[\omega_i \sin(ix) + \omega_{i+d/4}\cos(ix)] + \sum_{i=1}^{d/4}[\omega_{i+d/2}\sin(iy) + \omega_{i+3d/4}\cos(iy)] \right\} \end{cases}$$

Rough forcing term with 100 random variables

Numerical approximation: $u(\mathbf{x}) = \sum_{i=1}^{N_{dof}} w_i \Phi_i(\mathbf{x}) = \sum_{e=1}^{N_{el}} \sum_{p=0}^{P} w_p^e \phi_p^e(\mathbf{x}_e(\xi))$ ⇨

Multi-fidelity

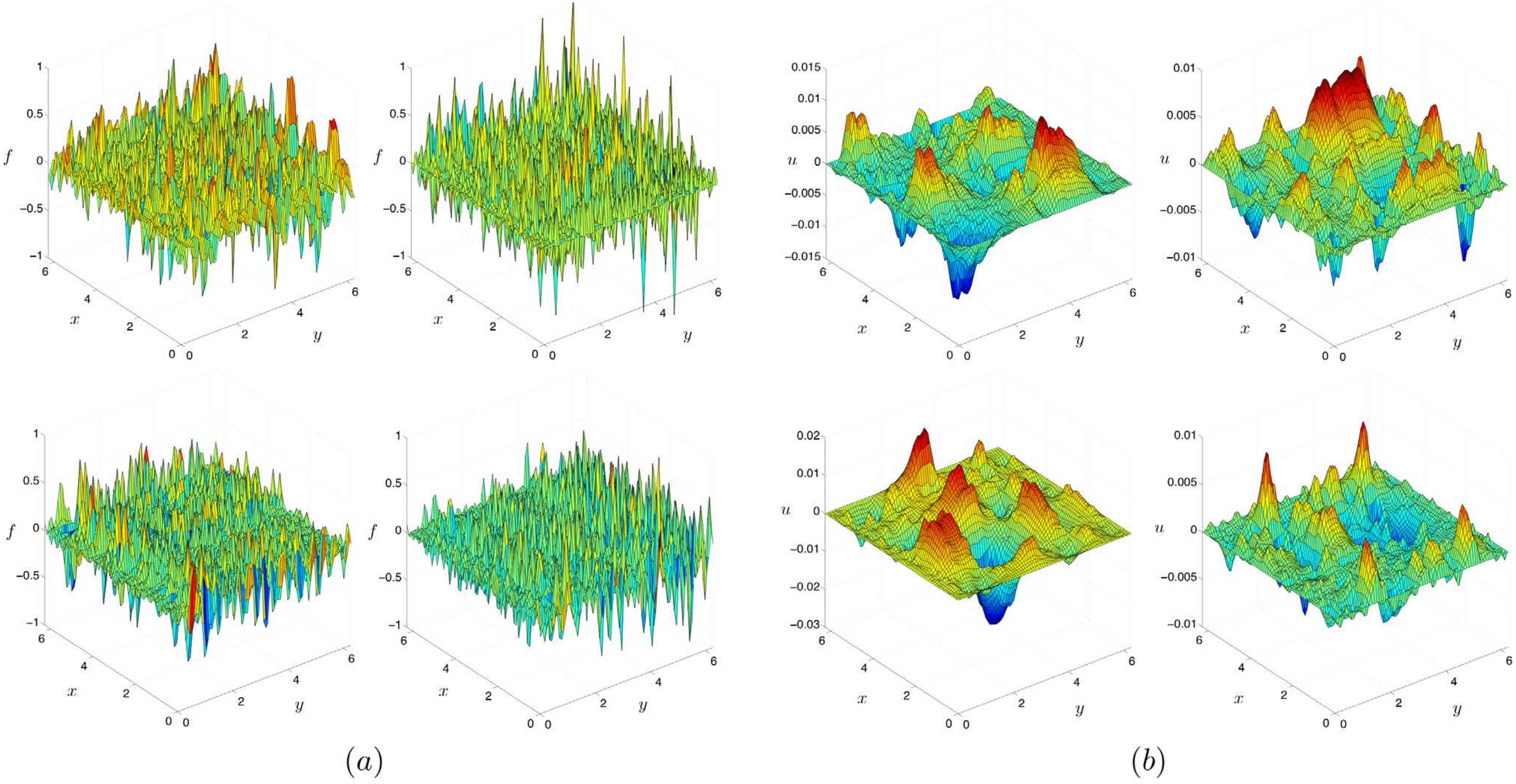
$N_{el} = 16, P = 4$	(10,000 samples)
$N_{el} = 64, P = 8$	(1,000 samples)
$N_{el} = 144, P = 10$	(100 samples)

Quantity of interest: $E_k(\omega) = \dfrac{1}{2}\displaystyle\int_0^{2\pi} u^2(x,t;\omega)dx$

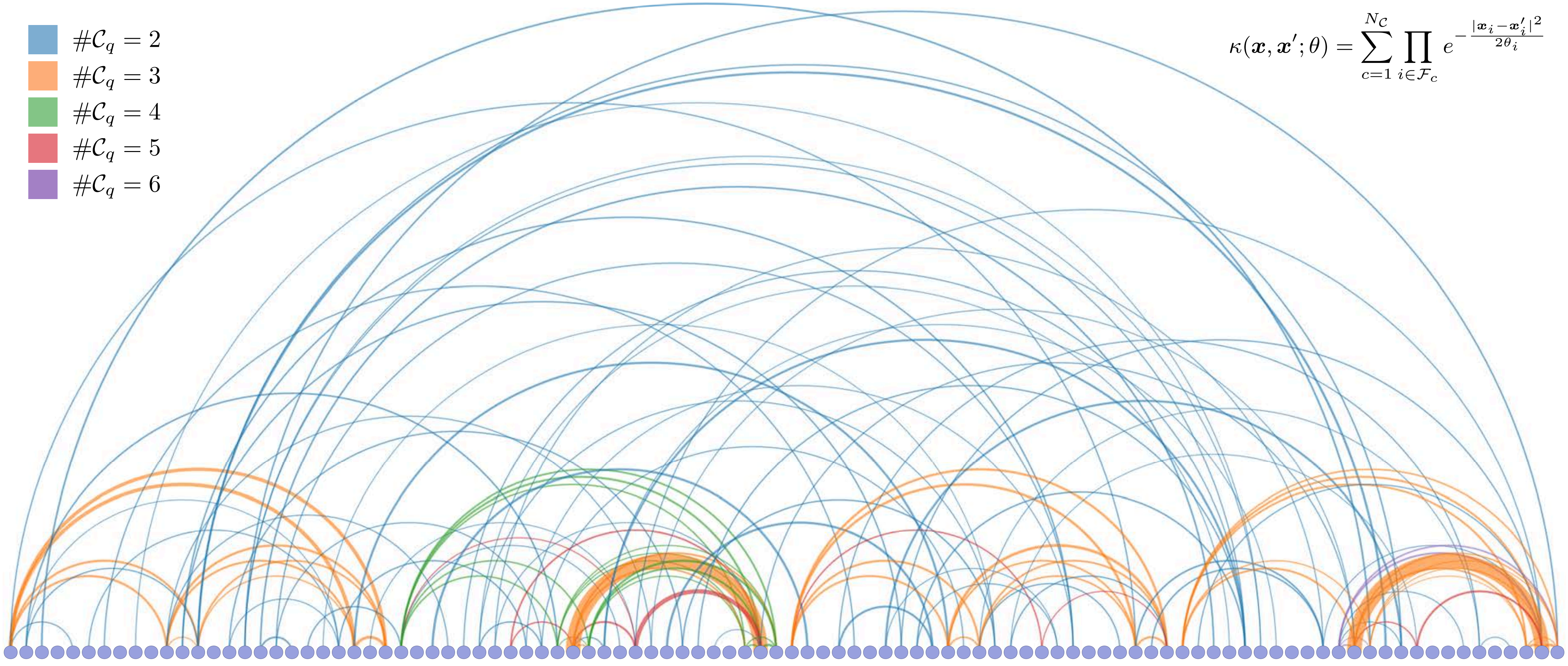
UQ in a 100-dimensional stochastic PDE

Samples of the random forcing term

Samples of the high-fidelity solution

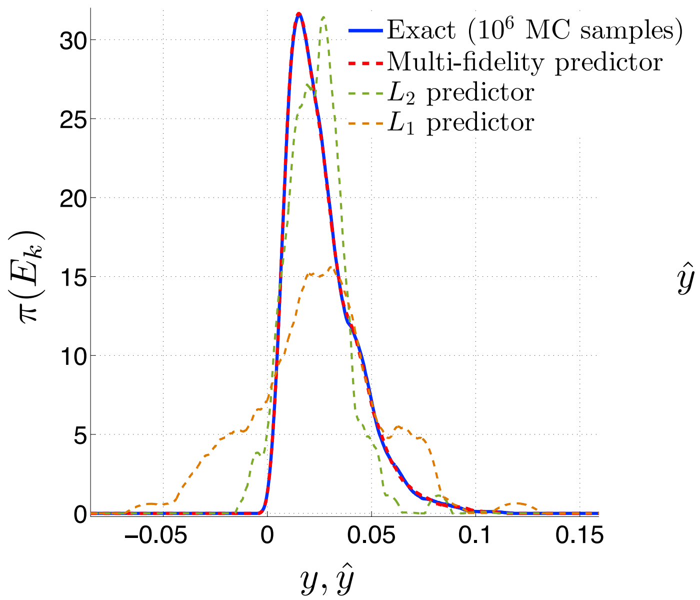


UQ in a 100-dimensional stochastic PDE

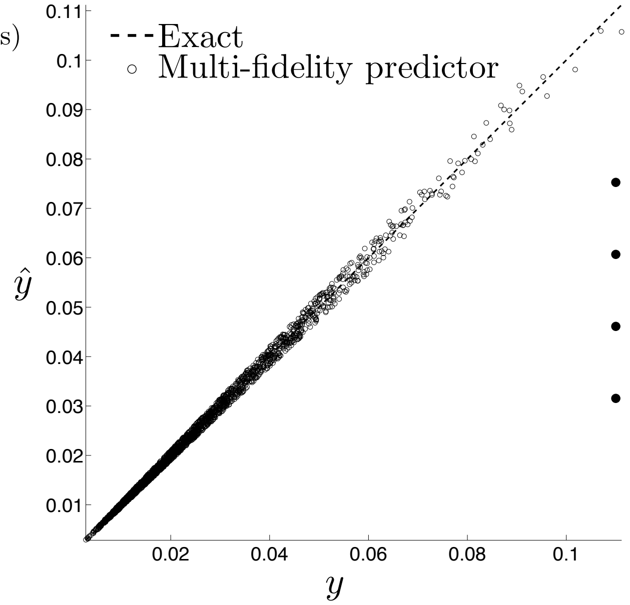


$$\kappa(\boldsymbol{x},\boldsymbol{x}';\theta)=\sum_{c=1}^{N_{\mathcal{C}}}\prod_{i\in\mathcal{F}_c} e^{-\frac{|\boldsymbol{x}_i-\boldsymbol{x}_i'|^2}{2\theta_i}}$$

(a)



(b)



(c)

- Non-trivial dimensionality
- Complex clique structure/interactions
- Accurate estimation of the solution PDF
- Orders of magnitude speed-up vs brute force MC

Perdikaris P., D. Venturi, G.E. Karniadakis Multi-fidelity information fusion algorithms for high dimensional systems and massive data-sets, SIAM J. Sci. Comput. (2016)

Summary

- General data-driven framework for supervised learning from variable-fidelity information sources

- Systematically combine seemingly different physical models (simulations, empirical correlations, noisy measurements, etc.), and different approximation methods in probability space (collocation, sparse grids, MC, etc.)

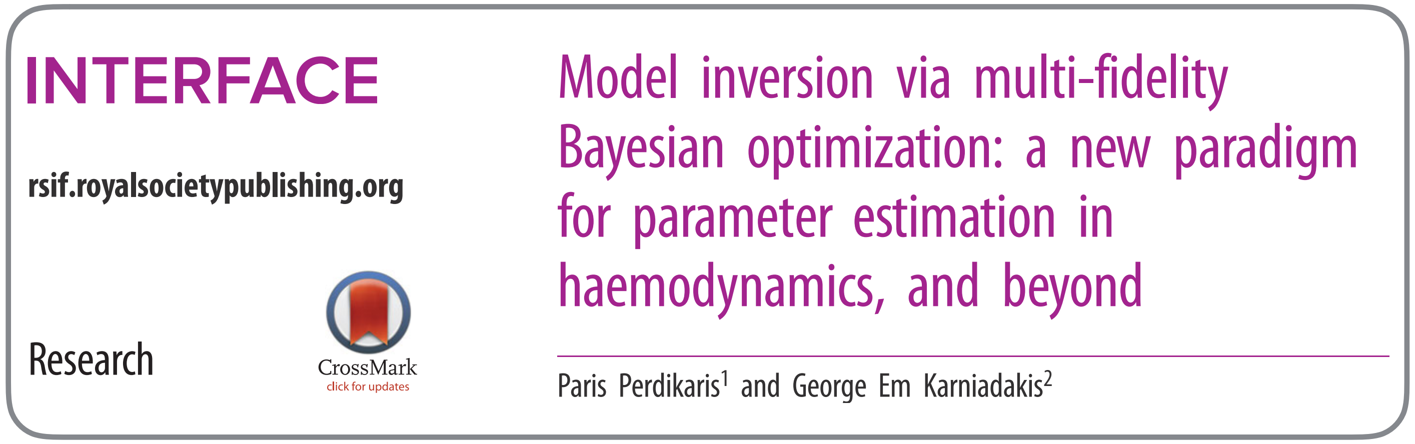
- Exploiting cross correlation between models can lead to orders of magnitude of speed up

- Applications in uncertainty quantification, optimization, inverse problems, data assimilation, and beyond

Taking the Human Out of the Loop: A Review of Bayesian Optimization

The paper introduces the reader to Bayesian optimization, highlighting its methodical aspects and showcasing its applications.

By Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas



INTERFACE

rsif.royalsocietypublishing.org

Research

CrossMark
click for updates

Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond

Paris Perdikaris[1] and George Em Karniadakis[2]

Questions?

Many thanks to Linda Petzold and George Karniadakis for the invite & Šeila Selimović and Grace Peng for arranging the broadcasting!

This work received support from DARPA grant N66001-15-2- 4055

Web: `http://web.mit.edu/parisp/www/`
Email: `parisp@mit.edu`

Model inversion in high-dimensions

Goal: Developed scalable algorithms for solving high- dimensional inverse problems

Optimization in physical space $\min_{\mathbf{x}\in\mathbb{R}^d} \|g(\mathbf{x}) - y^\star\|$

non-linear dim reduction

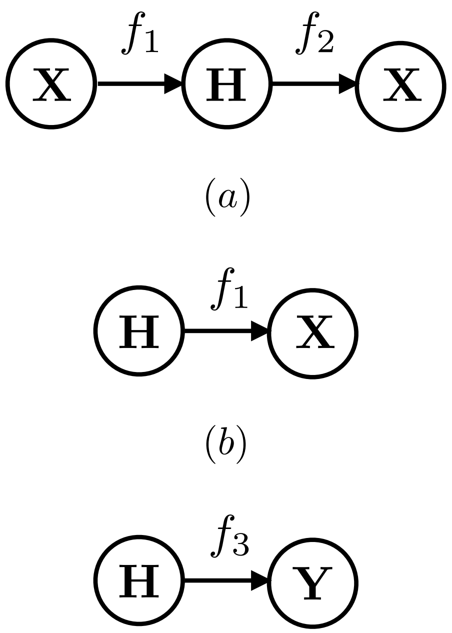
$q << d$

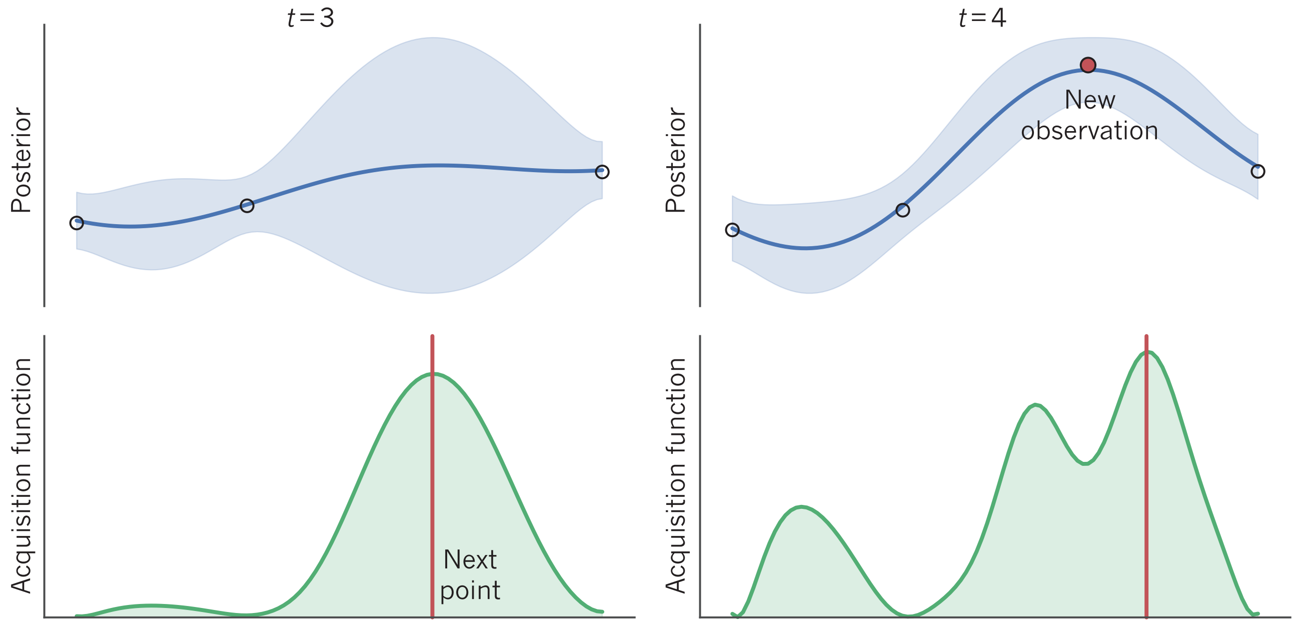
$\min_{\mathbf{h}\in\mathbf{H}^q} \|g(\mathbf{h}) - y^\star\|$ *Optimization in latent space*

Deep auto-encoder (supervised)

GPLVM (unsupervised)

$f_i \sim \mathcal{GP}$





Bayesian optimization in latent space

Technical approach:

- Non-linear dimensionality reduction using supervised deep auto-encoders and/or unsupervised GPLVMs
- Bayesian optimization in the low-dimensional latent space

Lawrence, N. D. "Gaussian process latent variable models for visualisation of high dimensional data." Advances in neural information processing systems 16.3 (2004): 329-336.
Shahriari, Bobak, et al. "Taking the human out of the loop: A review of bayesian optimization." Proceedings of the IEEE 104.1 (2016): 148-175.

Model inversion in high-dimensions

Example in 200 dimensions:

$$u_{xx} - \lambda^2 u = \sum_{k=1}^{K} w_k \sin(k\pi x)$$

$$u(0) = 0, \qquad u(1) = 0$$

Goal: Given a reference solution u*(x), recover the K = 200 weights, i.e.:

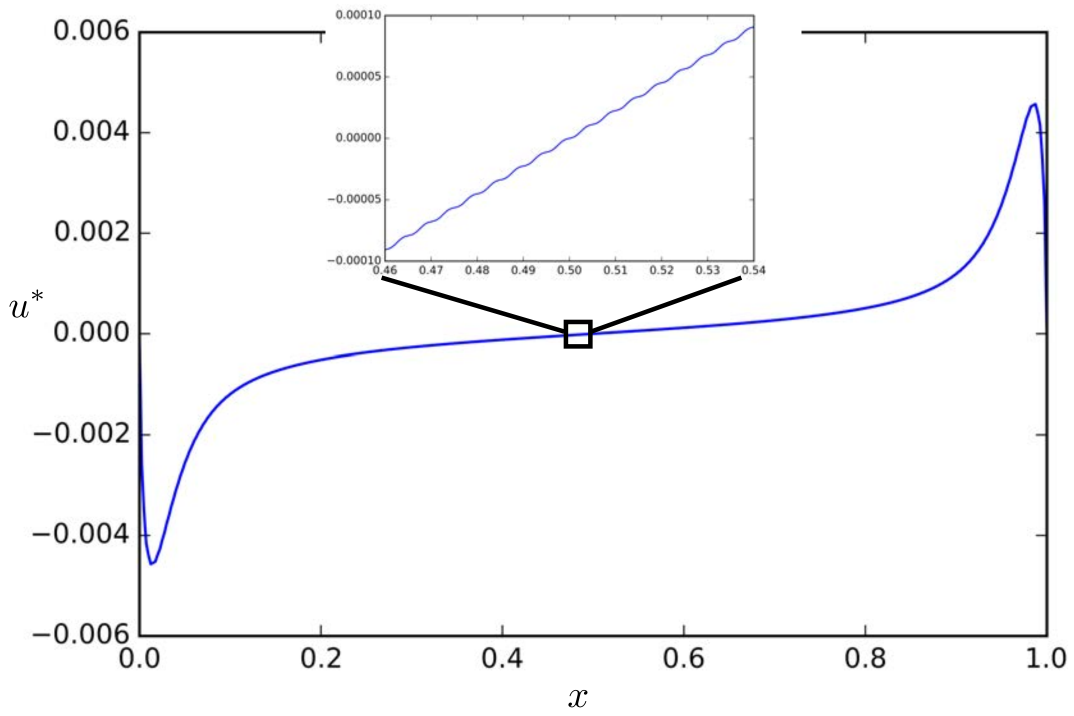
$$\min_{w_k \in \mathbb{R}^{200}} \|u(w_k) - u^*\|_2$$

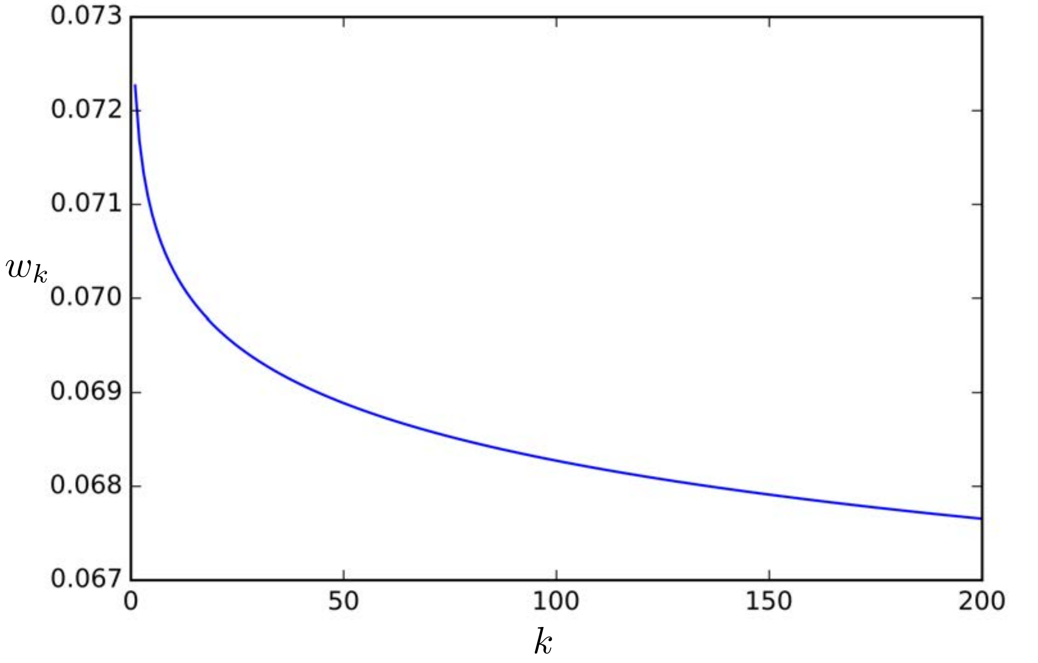
$$w_k = s(k) = \alpha \exp(\beta k^{\gamma}), \qquad \alpha, \beta, \gamma \in [0, 0, 5]$$

Workflow:

1. Create a training set containing a collection of randomly sampled spectra X
2. Identify a low-dimensional latent representation by training a one-layer deep auto-encoder on the random spectra $X \xrightarrow{f_1} H \xrightarrow{f_2} X$
3. Perform Bayesian optimization in latent space $H \xrightarrow{f_3} Y$
4. Use the deep auto-encoder posterior to map the optimal h* back to the physical domain x* $H \xrightarrow{f_2} X$

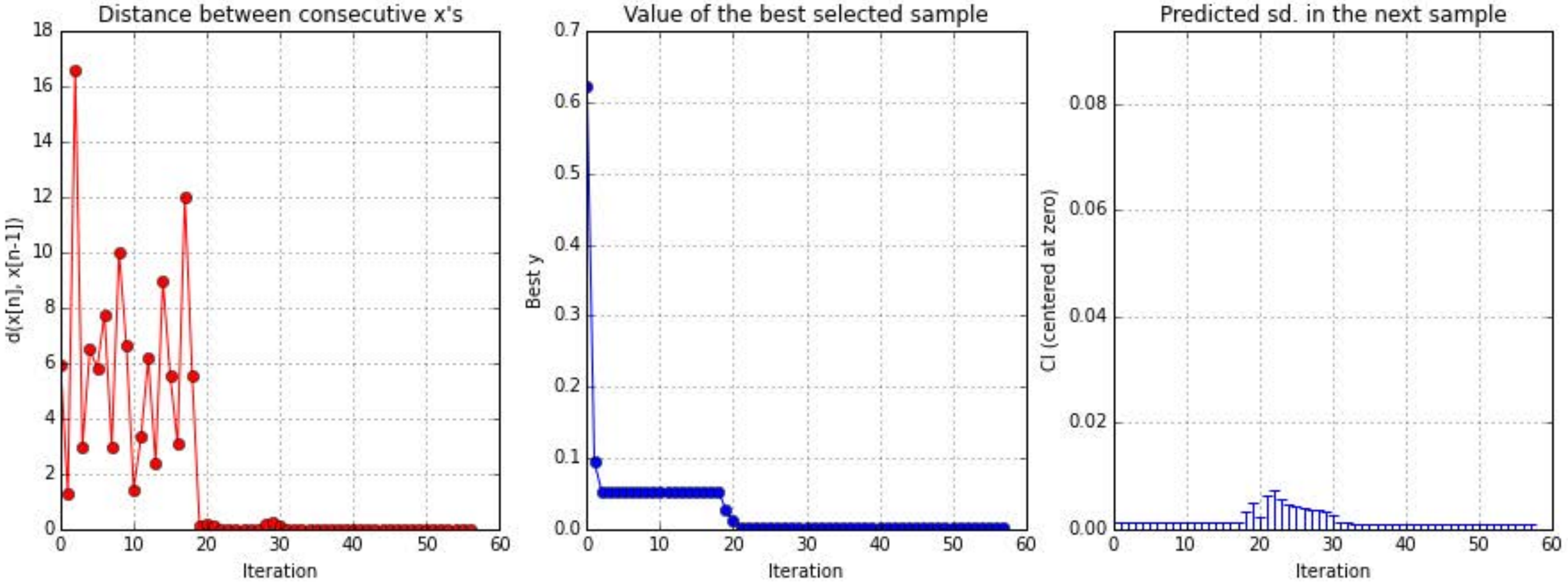
Reference solution u(x)*





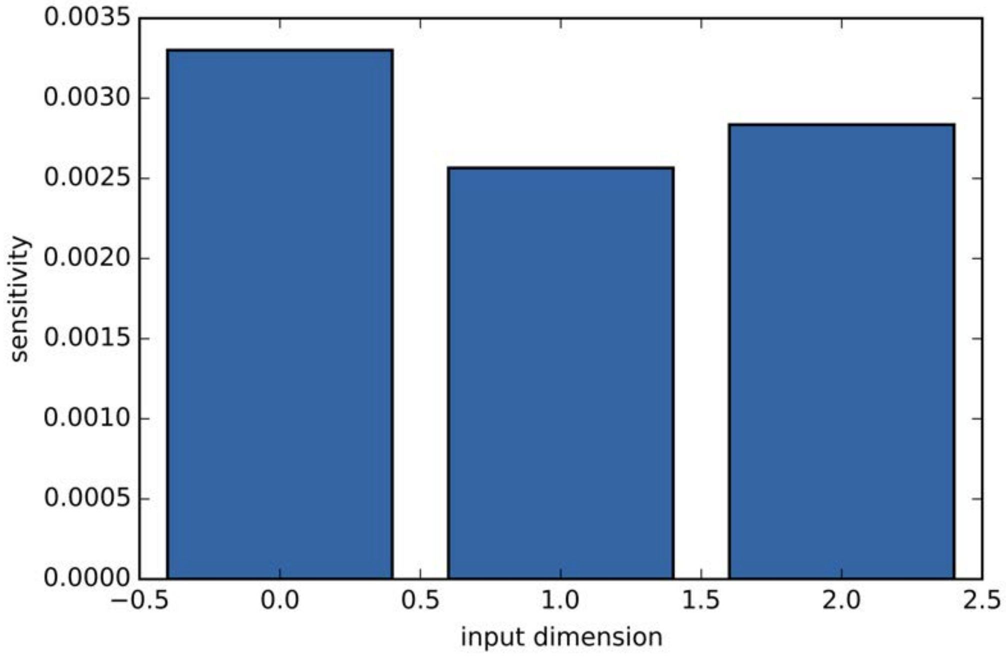
Decay of reference weights

Model inversion in high-dimensions



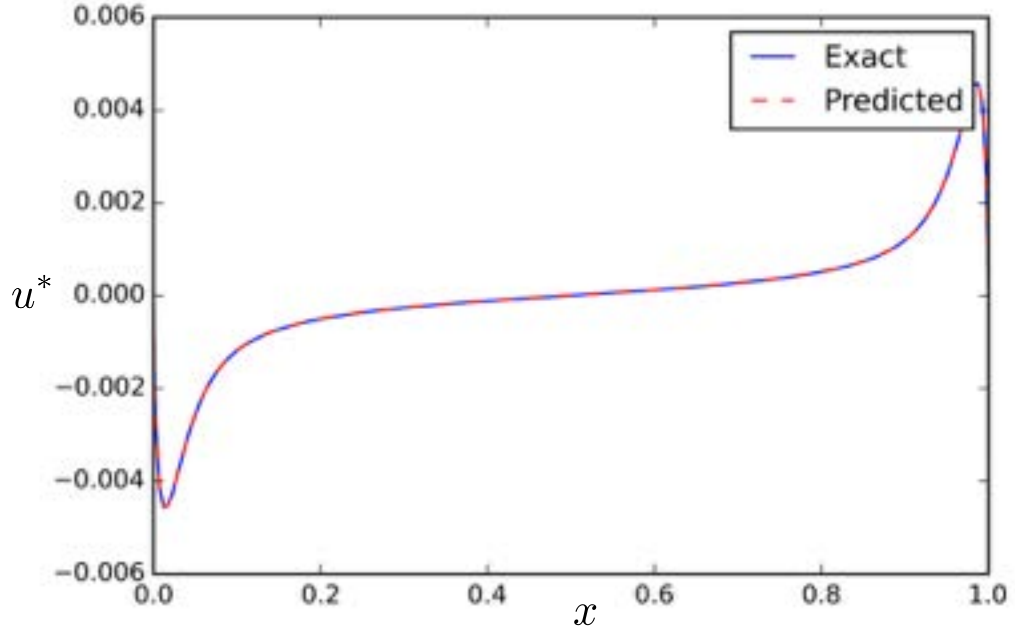
Convergence of Bayesian optimization in latent space

(notice that only 21 evaluations of the PDE are required to get an accuracy of O(1e-3))



Dimensionality of the discovered latent space

(notice how the deep auto-encoder recovered the 3-parameter dependence of the random spectra)

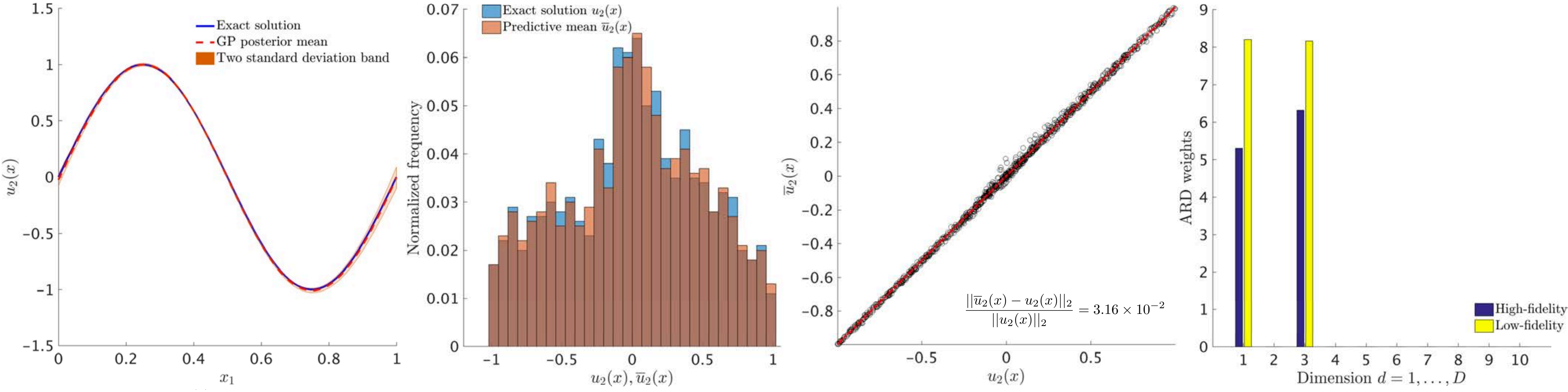


Accuracy of learned weights in reconstructing the reference solution

(Relative L2 error: 3.750090e-03)

Example: Poisson equation (10D)

$$\begin{aligned}
&\boldsymbol{y}_i = f_i(\boldsymbol{x}_i) + \boldsymbol{\epsilon_i},\ i = 1, 2,\\
&\boldsymbol{\epsilon_1} \sim \mathcal{N}(0, 0.3I),\ \boldsymbol{\epsilon_2} \sim \mathcal{N}(0, 0.05I),\\
&f_2(x) = -8\pi^2 \sin(2\pi x_1) \sin(2\pi x_3),\ \text{and}\ f_1(x) = 0.8 f_2(x) - 40 \prod_{d=1}^{10} x_d + 30,\\
&\boldsymbol{u}_0 = u_2(\boldsymbol{x}_0) + \boldsymbol{\epsilon}_0\ \text{with}\ \boldsymbol{\epsilon}_0 \sim \mathcal{N}(0, 0.01I),\\
&u_2(x) = \prod_{d=1}^{D} \sin(2\pi x_d)
\end{aligned}$$



Remarks:

- Noisy, multi-fidelity data on f, noisy boundary conditions on u.
- Good accuracy and automatic detection of effective dimensionality
- A-posteriori error estimates through the GP posterior variance