COLLEGE OF ENGINEERING
AEROSPACE ENGINEERING
UNIVERSITY OF MICHIGAN

SAND2018-0659C

Multifidelity Model Management using Latent Variable Bayesian Networks

Alex Gorodetsky, University of Michigan

Joint work with: Gianluca Geraci, Mike Eldred, and John Jakeman, Sandia National Labs

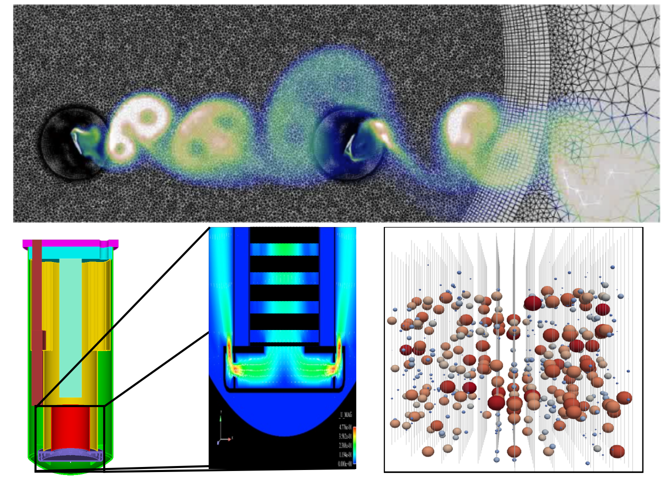
Funded by the DARPA EQUiPS SEQUOIA project

Uncertainty quantification for HF models

- Characterize effect of uncertainty on HF models
- Severe simulation budge
  - High dimensional PDEs
  - Large-scale computing resources
- Significant dimensionality, driven by model complexity
- Higher-fidelity models → UQ more important
  - Less available runs, less study and analysis
  - Nonlinearities become more important
  - Increased effect of model errors, wrong initial conditions, wrong environmental conditions

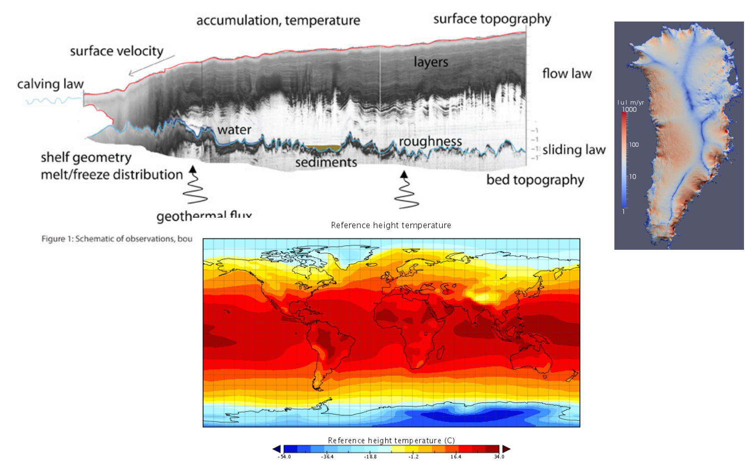
Energy (ASCR, EERE, NE)

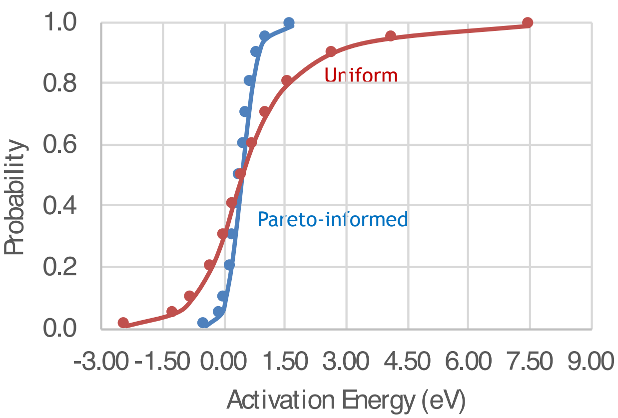
Wind turbines, nuclear reactors

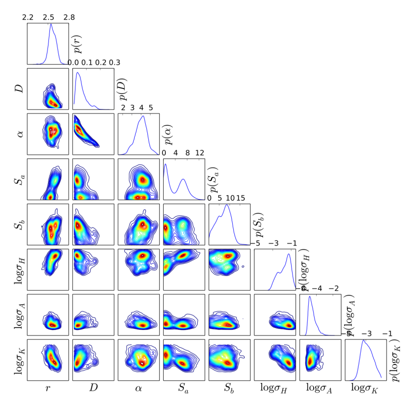


Climate (SciDAC, CSSEF, ACME)

Ice sheets, CISM, CESM, ISSM, CSDMS

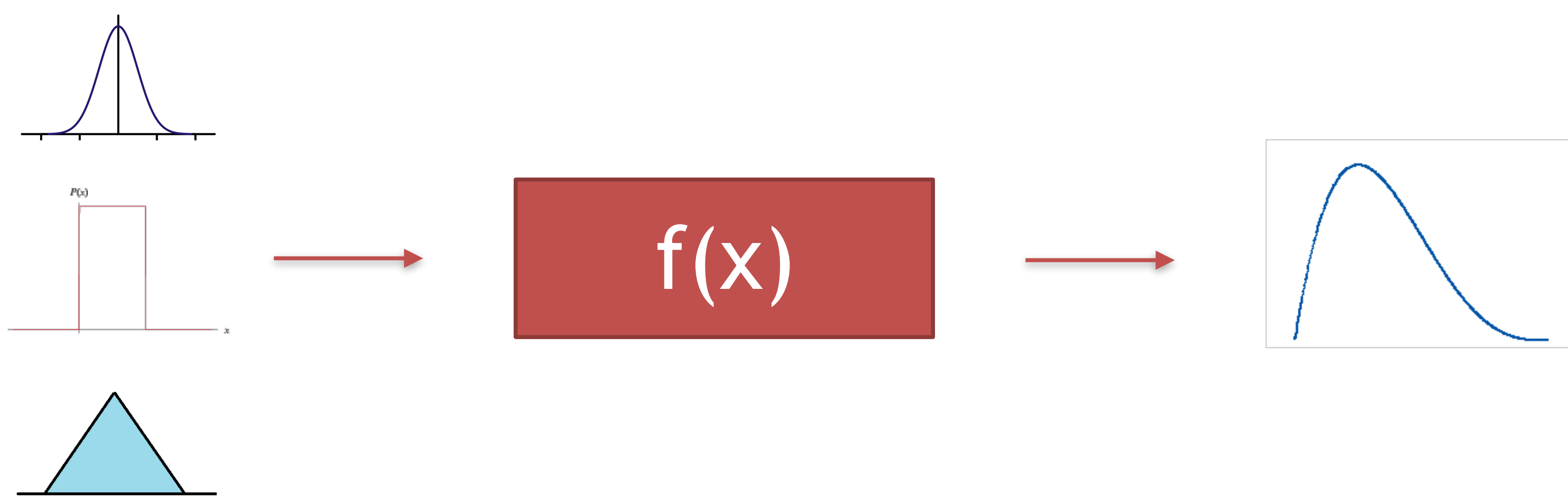






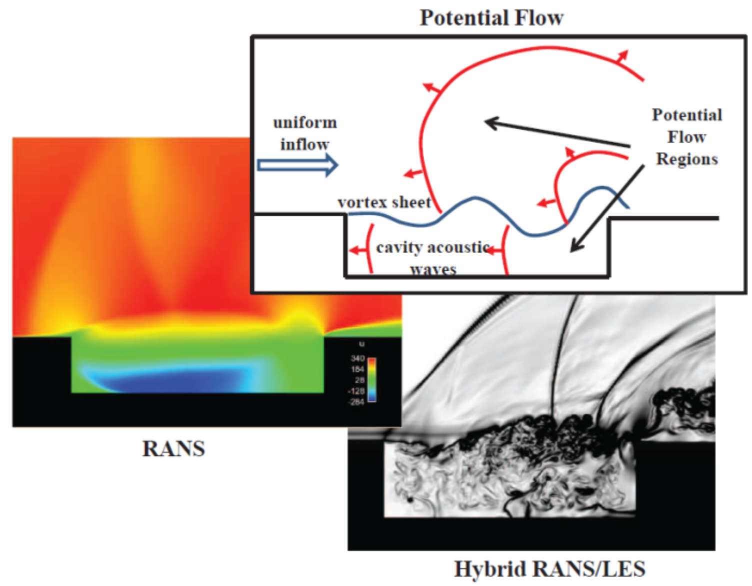
Uncertainty quantification is expensive

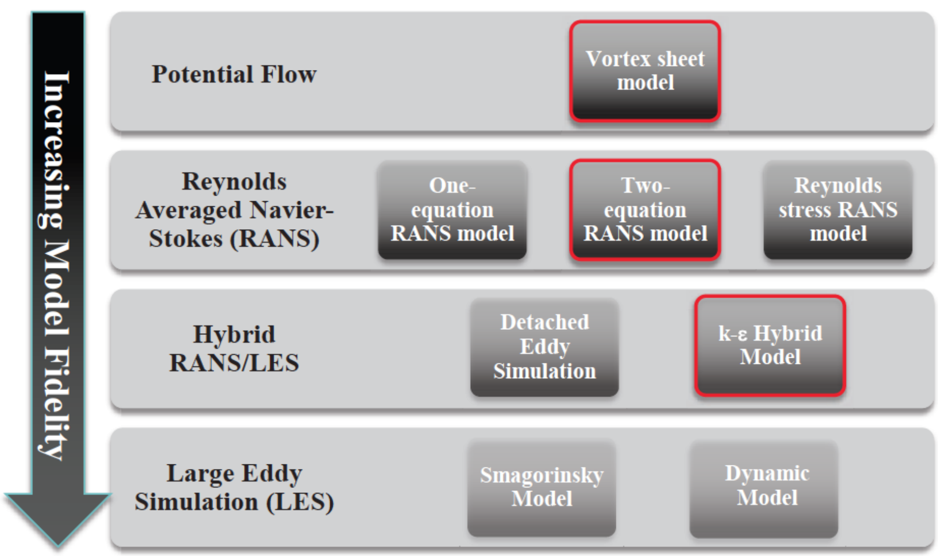
- Sampling methods
  - Monte Carlo methods need order of magnitude increase in samples for every digit of accuracy
  - Generally don’t account for additional structure, e.g., smoothness, sparsity, compressibility, decomposability
- Surrogate based methods like PCE or Gaussian process regression
  - Take advantage of smoothness or compressibility
  - Suffer from curse of dimensionality



Leveraging multiple models for better accuracy

- Hierarchy of fidelities
- Ensemble of peer models
- Discretization levels / resolution controls
- Multi-physics and multi-scale





Challenges to managing multiple models

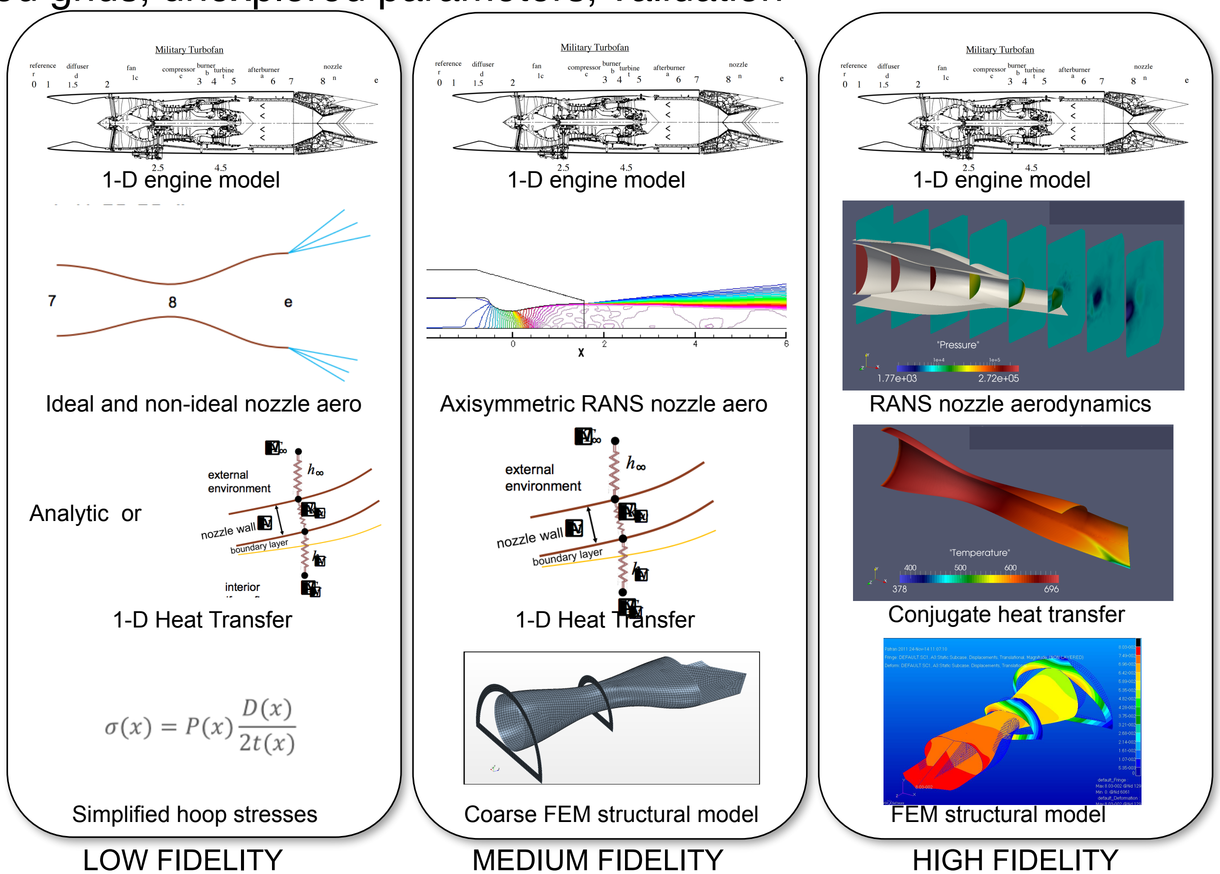
Model fusion

- Models with different inputs/outputs
- Legacy data collected separately for each model
- Simulation codes may not have assumed relationship
  - Corrupted evaluations, unconverged grids, unexplored parameters, validation only for some parameter settings

(a) X47B UCAS (b) Nozzle close-up

Figure 1: Northrop Grumman X-47B UCAS and close up of its nozzle.[24]

http://www.northropgrumman.com/MediaResources/Pages/MediaGallery.aspx?ProductId=UC-10028



Multilevel Monte Carlo and Control Variates

Variance reduction techniques assume model relationships

Monte Carlo	$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$	$\mathbb{V}ar[\hat{Y}] = \frac{\sigma_Y^2}{N}$
Multilevel Monte Carlo Giles 2008 Relies on decay of variance of discrepancies	$\mathbb{E}\left[Y_L\right] = \mathbb{E}\left[Y_0\right] + \sum_{\ell=1}^{L} \mathbb{E}\left[Y_\ell - Y_{\ell-1}\right]$ $\hat{Y}_L^{ML} = \hat{Y}_0 + \sum_{\ell=1}^{L} \hat{\Delta}_l$	$\mathbb{V}ar\left[\hat{Y}_L^{ML}\right] = \frac{\sigma_0^2}{N_0} + \sum_{\ell=1}^{L} \frac{\sigma_{\Delta_l}^2}{N_\ell}$
Control Variates Most general, considers correlations between all models	$\hat{Y}_L^{CV} = \hat{Y}_L - \Sigma_{L,:L}\Sigma_{:L,:L}^{-1}\left[\hat{Y}_{:L} - \mathbb{E}\left[Y_{:L}\right]\right]$	
Multifidelity Monte Carlo Peherstorfer 2016 Approximates CV with sparse precision	$\hat{Y}_L^{MFMC} \approx \hat{Y}_L - \sum_{i=1}^{L} \alpha_i \left(\hat{Y}_\ell - \mathbb{E}\left[Y_\ell\right]\right)$	

Exploiting structure within simulations

Co-kriging or Gaussian Process Regression

- Co-kriging fuses information from multiple sources
  - Bayesian regression, builds a distribution over functions
  - Exploits smoothness properties of information source using kernels (nonparametric) or basis functions (parametric)
- Recursive Co-Kriging for Multifidelity Models*
  - Usage for hierarchies of models
  - High-fidelity GP is written as a sum of the lower fidelity GP and a new variation term

$$Y_L(x) = \rho_{L-1}(x)Y_{L-1}(x) + \delta_L(x)$$

$$\rho_{L-1}(x) = g_{L-1}(x)^T\beta_{L-1}$$

$$Y_1 \sim \mathcal{GP}(m(x), \sigma(x))$$

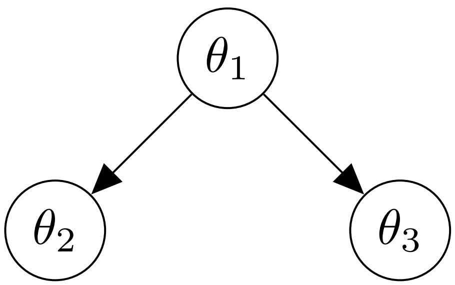
$$\delta_L \sim \mathcal{GP}(m_{\delta_L(x)}\sigma_{\delta_L}(x))$$

Qian and Wu 2004, Gratiet and Garnier 2014, Perdikaris et. al. 2015

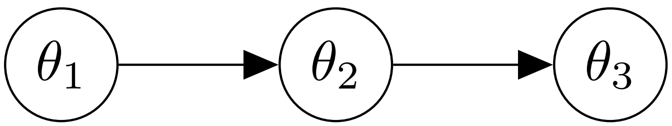
Main Idea

Fuse multiple models by learning their statistical relationships using networks of latent variables

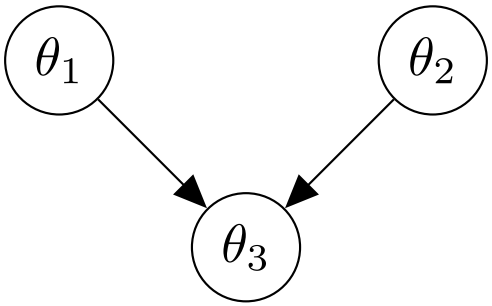
Low-fidelity peers



Hierarchy



High-fidelity peers



Latent variable models

Hidden relationships between complex models

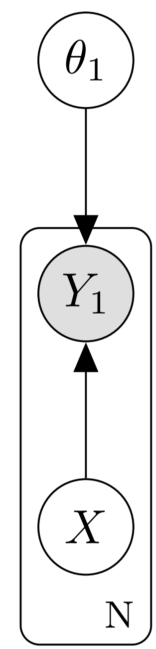
- Introduce hidden variables to explain observed data (surrogate, emulator, etc)
  - Parametric: polynomial chaos expansions, deep networks
  - Nonparametric: Gaussian process regression
- Single fidelity UQ: learn latent variable model → propagate uncertainty through LVM

Single fidelity training

Inputs	$x \in \mathcal{X}$
Outputs	$y \in \mathbb{R}^n$
Latent Vars	$\theta \in \mathbb{R}^p$
Mapping	$\phi : \mathcal{X} \to \mathbb{R}^{n \times p}$
LVM	$y = \phi(x)\theta$

$$\mathbb{E}\left[Y\right] \approx \mathbb{E}\left[\phi(x)\theta\right] = \mathbb{E}\left[\phi(x)\right]\theta$$

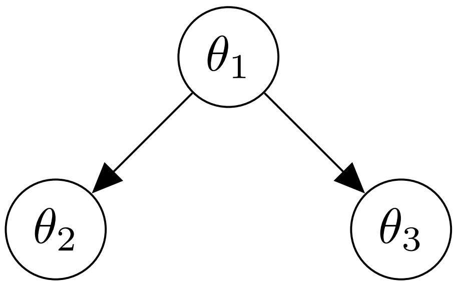
- Maximum Likelihood
  - Least squares
- Regularization MAP estimate
  - Ridge Regression (L_2)
  - Sparse Regression (L_1 or L_0)
- Fully Bayesian



Latent variable networks

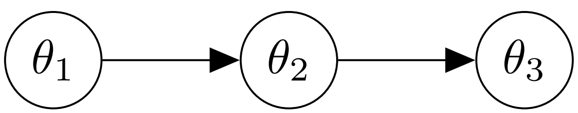
Graphs encode interpretable structure

- Peer low fidelity models
  - Example: Model 1 has a composite turbulence models, second two models have components



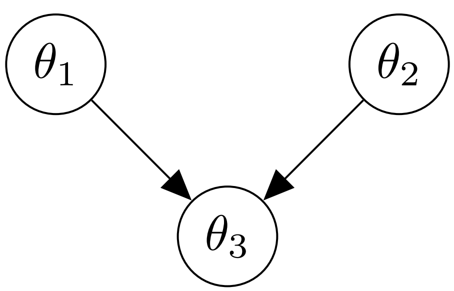
$$p(\theta_1, \theta_2, \theta_3) = p(\theta_1)p(\theta_2|\theta_1)p(\theta_3|\theta_1)$$

- Distinct model hierarchies
  - Example: refined discretization as in a multilevel scheme



$$p(\theta_1, \theta_2, \theta_3) = p(\theta_1)p(\theta_2|\theta_1)p(\theta_3|\theta_2)$$

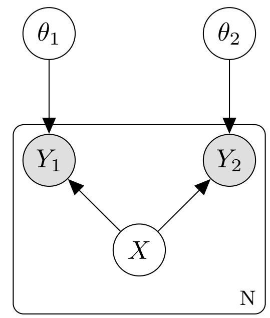
- Peer high fidelity models:
  - Example: independent high-fidelity models with an overlapping prediction



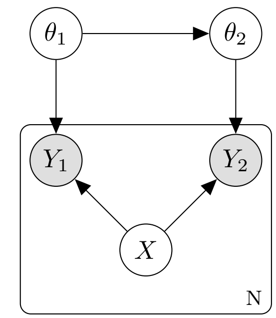
$$p(\theta_1, \theta_2, \theta_3) = p(\theta_1)p(\theta_2)p(\theta_3|\theta_1, \theta_2)$$

Learning network structure from data

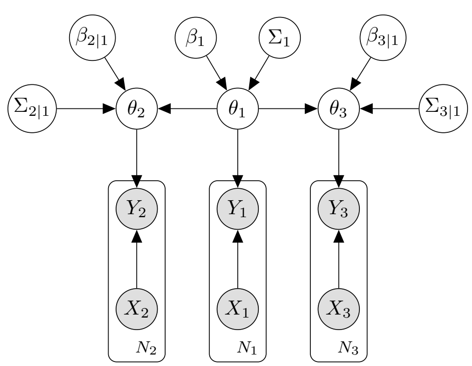
Can we distinguish between



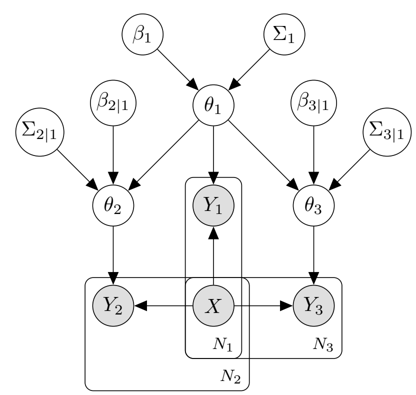
and



1. Learn model structure and transition probabilities by maximizing some score
   - AIC, BIC, etc.
2. Learn only parameters of a given structure
   - Fix a parametric family for conditional probabilities
   - Include uncertainty over CPD parameters → hierarchical Bayes



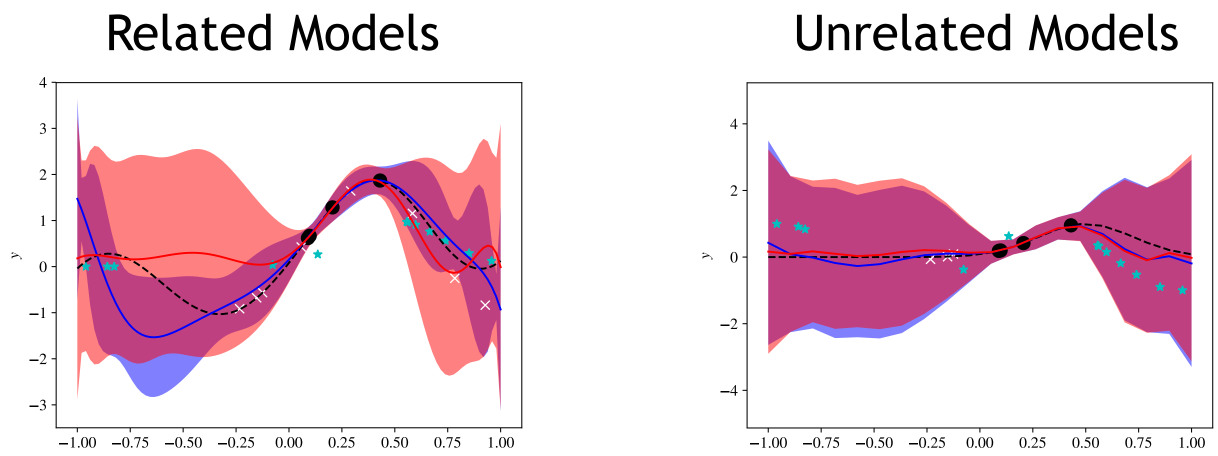
Independent model, different data sets



Independent model, shared data

Why learn network structure?

- Can learn when models are related and when they are not



- Reduction in uncertainty for related models, no degradation for unrelated models
- Efficient learning of correlations between models
  - Don’t consider all possible correlations
  - Don’t assume a specific set of correlations to consider

Computational complexity

- Learning the general structure of M models must consider $\mathcal{O}(2^{M^2})$ graphs
- We can make two common assumptions

1. Known variable ordering: certain models cannot be lower-fidelity than others

$$j > i \implies \theta_j \notin \text{parents}(\theta_i)$$

   - Now "only" $\mathcal{O}(2^M)$ candidates

2. Limit number of parents of each node

$$\mathcal{O}\left(k\binom{M}{k}\right) = \mathcal{O}\left(kM^{k+1}\right)$$

Maximum likelihood UQ for joint normality

Relationship to control variates

- Recall 1-1 mapping of MC estimate of mean and solution of a least squares system

$$\arg\min\|\mathbf{y} - \phi(\mathbf{x})\theta\| = \left(\phi(\mathbf{x})^T\phi(\mathbf{x})\right)^{-1}\phi(\mathbf{x})^T\mathbf{y} \qquad \phi(x) = 1 \implies \theta = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$$

- We have derived a similar result for joint Gaussian LVN

$$\theta_1, \theta_c \sim \mathcal{N}(\mu, \Sigma) \quad \mu = \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix}\Sigma_{11} & \Sigma_{1c} \\ \Sigma_{c1} & \Sigma_{cc}\end{bmatrix}$$

  - Likelihood model

$$(y_1, y_c)|x, \mu_1 \sim \mathcal{N}\left(\begin{bmatrix}\phi_1(x)^T\mu_1 \\ \phi_2(x)^T\mu_c\end{bmatrix}, \mathbf{C}(x)\right) \quad \text{with} \quad \mathbf{C}(x) = \begin{bmatrix}c_{11}(x) & c_{12}(x) \\ c_{21}(x) & c_{22}(x)\end{bmatrix}$$

$$\hat{\theta}^{MLE} = \hat{\mu}_1^{MLE} = \left[\phi_1(\mathbf{x})^T\mathbf{A}\phi_1(\mathbf{x})\right]^{-1}\phi_1(\mathbf{x})^T\mathbf{A}\left[y_1 - \mathbf{c}_{12}(\mathbf{x})c_{22}(\mathbf{x})^{-1}\left[y_c - \phi_2(\mathbf{x})\mu_c\right]\right],$$

Under a constant basis a MLE with LVN is multivariate control variates

Computationally challenging, need to estimate entire covariance matrix and invert large scale matrix

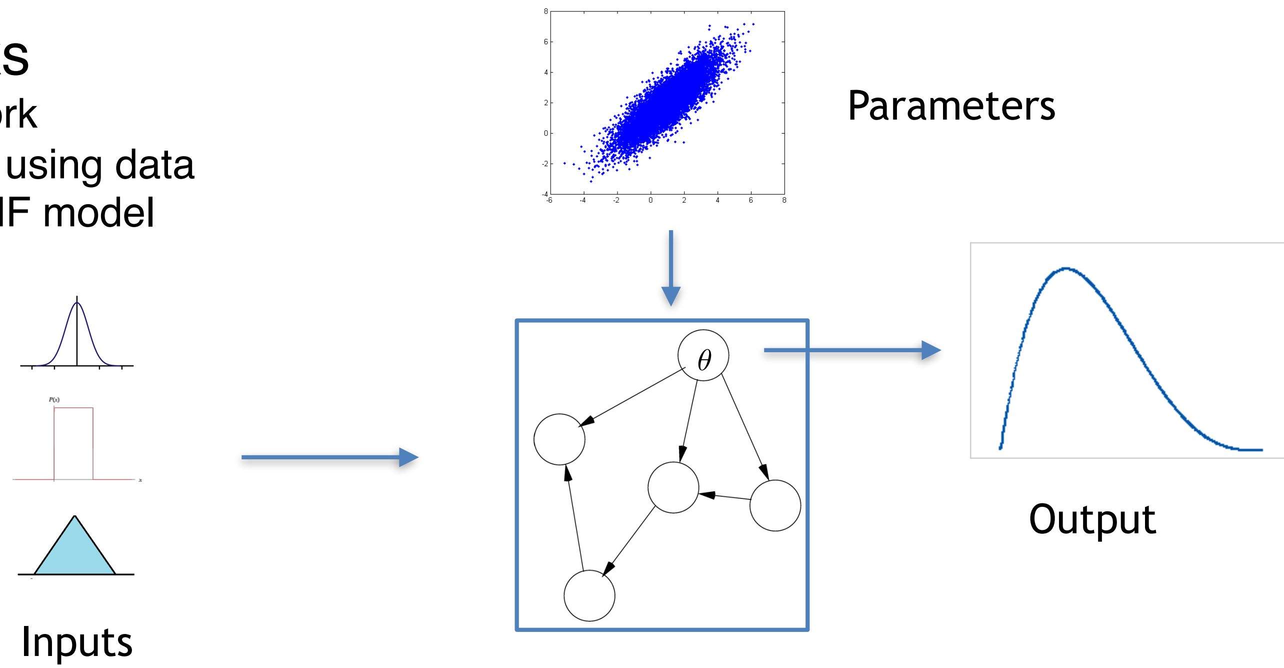
Bayesian UQ for networked models

From Bayesian Monte Carlo* to UQ through Bayesian LVNs

- Bayesian Monte Carlo
  - Set prior for output of function (i.e., for its parameters)
  - Update prior from data
  - Propagate posterior through integrals

$$\theta \sim \mathcal{N}(m, C) \implies \hat{Y} \sim \mathcal{N}\left(\mathbb{E}\left[\phi(x)\right] m, \mathbb{E}\left[\phi(x)\right] C \mathbb{E}\left[\phi(x)^T\right]\right)$$

- Bayesian Latent Networks
  - Set prior distribution over network
  - Update distribution or find MAP using data
  - Propagate uncertainty though HF model



*See Bayesian Monte Carlo by Ghahramani and Rasmussen 2003

Examples

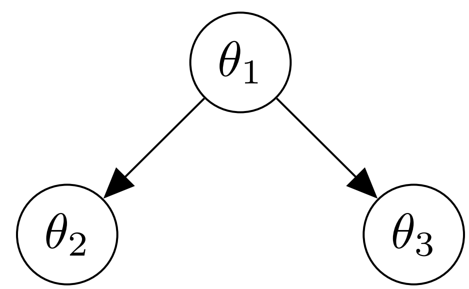
- Distinguish between three graphs: peers, recursive, and independent
- Use Legendre polynomial basis, total order in multivariate
- Prior on coefficients an multivariate normal with a diagonal covariance and decaying entries

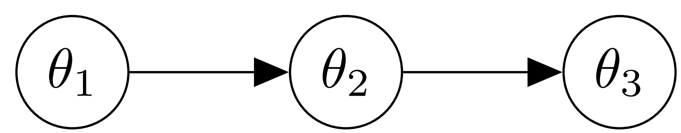
$$f(x) = \sum_{\alpha} \theta_{1,\alpha} \phi_{\alpha_1}(x_1) \dots \phi_{\alpha_d}(x_d)$$

$$\theta_1 \sim \mathcal{N}\left(0, \mathrm{diag}\left[w_1, \dots, w_P\right]\right)$$

$$\theta_j \sim \mathcal{N}\left(m_j \theta_i, c_j \mathbf{I}\right)$$

$$y_j \sim \mathcal{N}\left(\phi(x)\theta_j, \eta_j\right)$$

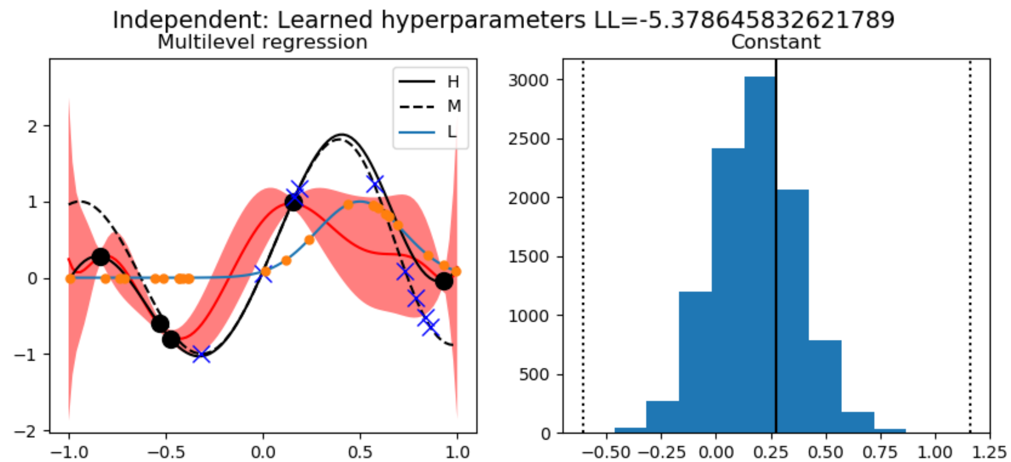


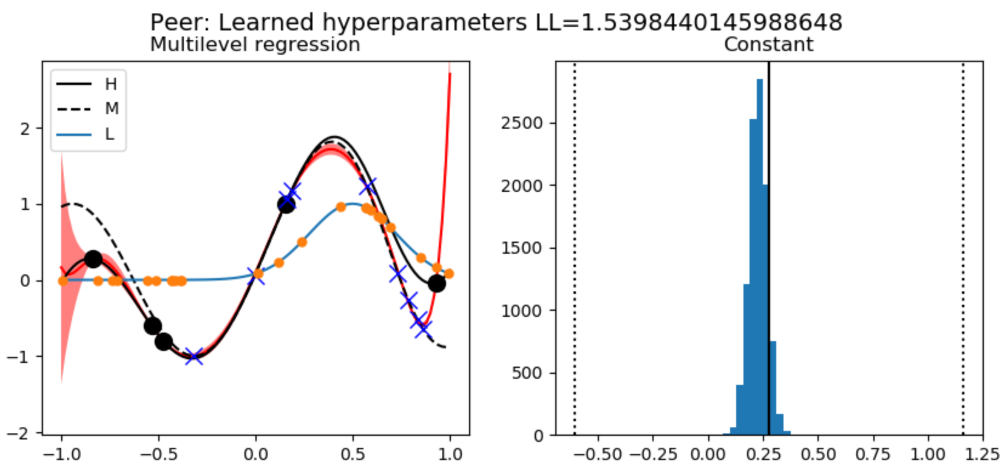


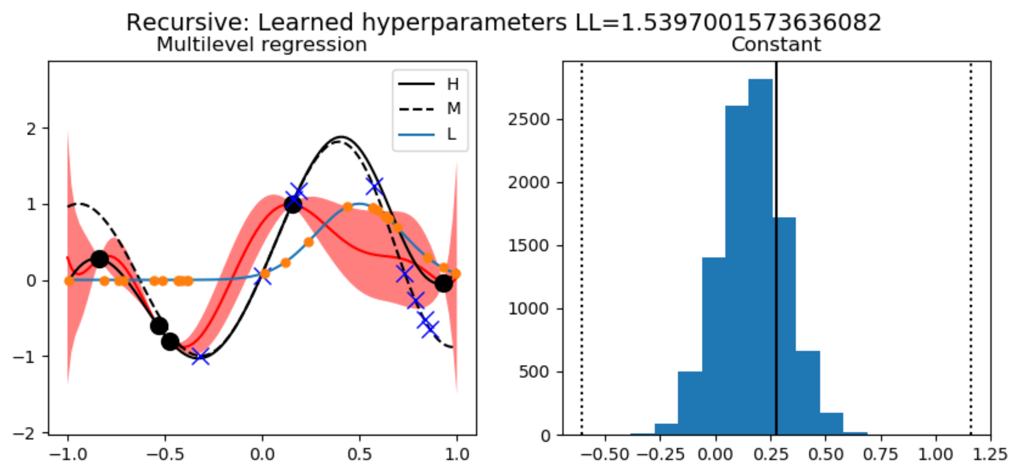
Related models

- 10th order polynomials
- [5, 9, 16] evaluations per model
- Taking advantage of smoothness yields better accuracy than MC

$$f_L(x) = \exp\left[-10\,(x-0.5)^2\right]$$
$$f_M(x) = f_L(x) + \sin(5x)$$
$$f_H(x) = f_M(x) + x^3$$





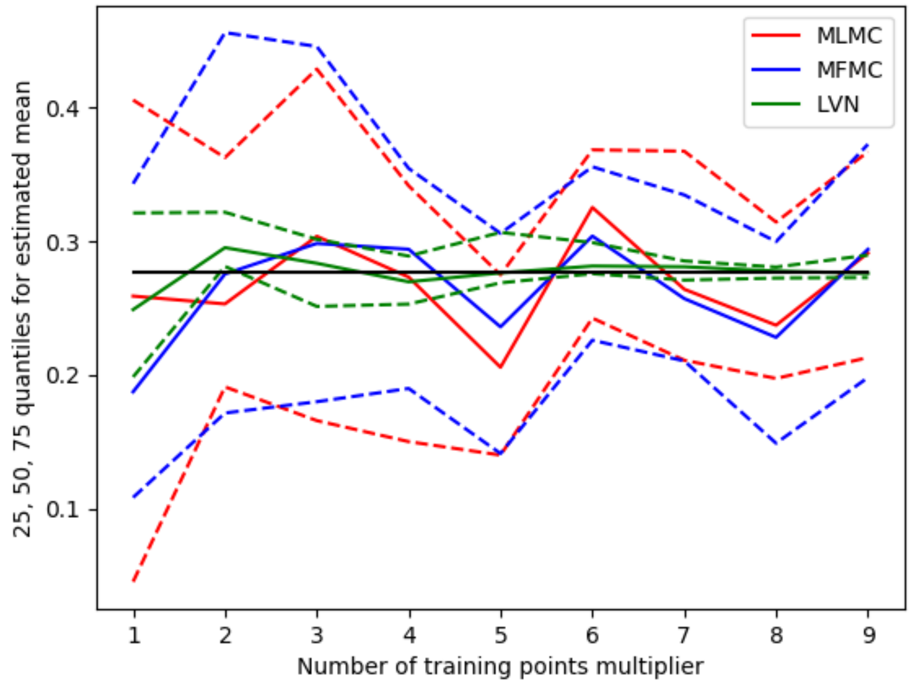


Related models

UQ comparison: Recursive Network vs MLMC vs MFMC

- Compare convergence MLMC, MFMC, and LVN estimators
- Multiplier of [5, 10, 15] samples per fidelity
- Repeat each experiment 20 times
- MLMC and MFMC have very similar performance
- LVN has much tighter bounds and faster convergence

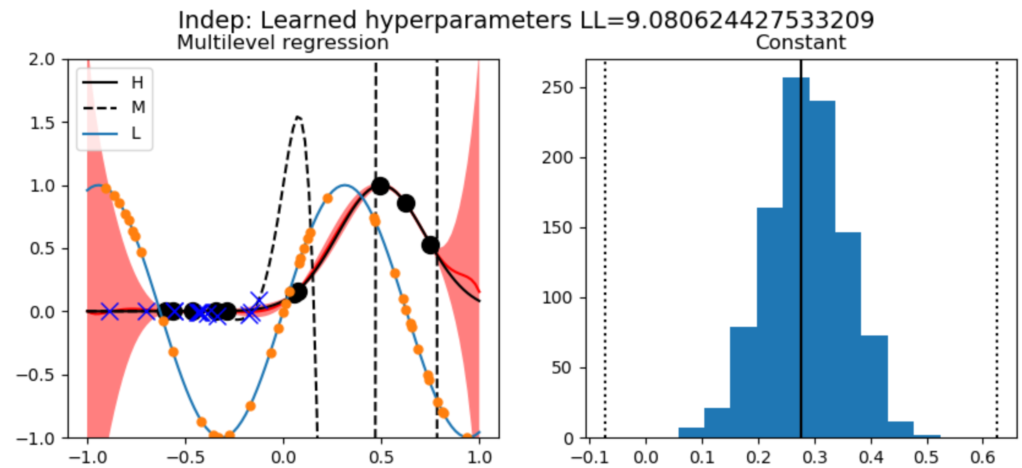
Multiplier	Recursive	Peer	Independent
1	7	13	0
2	7	13	0
3	11	9	0
4	5	15	0
5	9	11	0
6	10	10	0
7	6	14	0
8	3	17	0
9	6	14	0

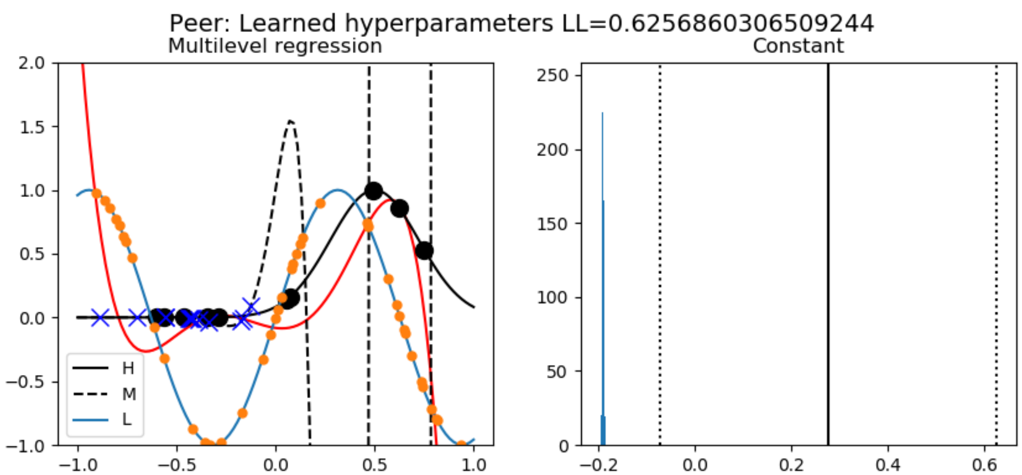


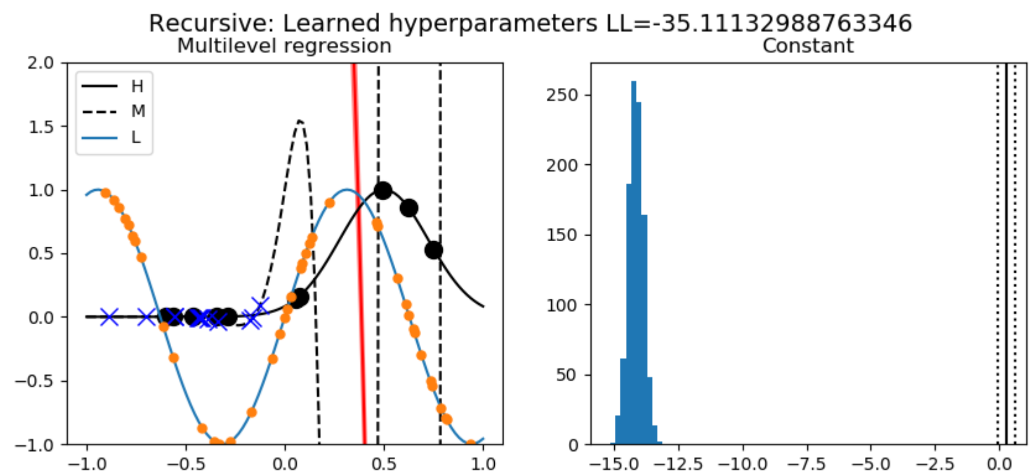
Unrelated models

- 10th order polynomials
- [10, 20, 40] evaluations per model
- Restricted bounds on hyperparameters

$$f_L(x) = \sin(5x)$$
$$f_M(x) = \cos(10x)\exp(10x)$$
$$f_H(x) = \exp\left(-10\left(x-0.5\right)^2\right)$$



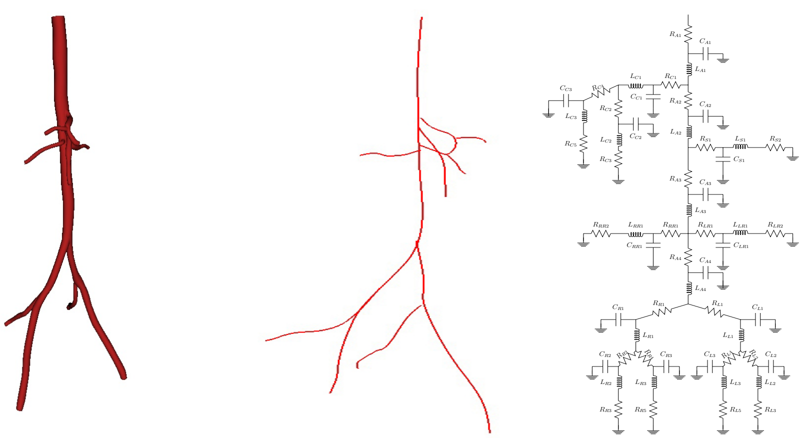




Cardiovascular flow

Based on work by Fleeter, C. Geraci, G. et. al. CCR Proceedings 2017

- Hemodynamic modeling
- Complex workflow
  - Medical image data → vascular anatomy
  - Numerical solutions of equations governing blood flow in elastically deformable vessels
  - A variety of intermediate models
- Uncertainties are prevalent
  - Spatial variability of material properties
  - Uncertain anatomy



(a) 3D model. (b) 1D model. (c) 0D model.

Fig. 6.2: Schematic view of the three model fidelities for the aorto-femoral model.

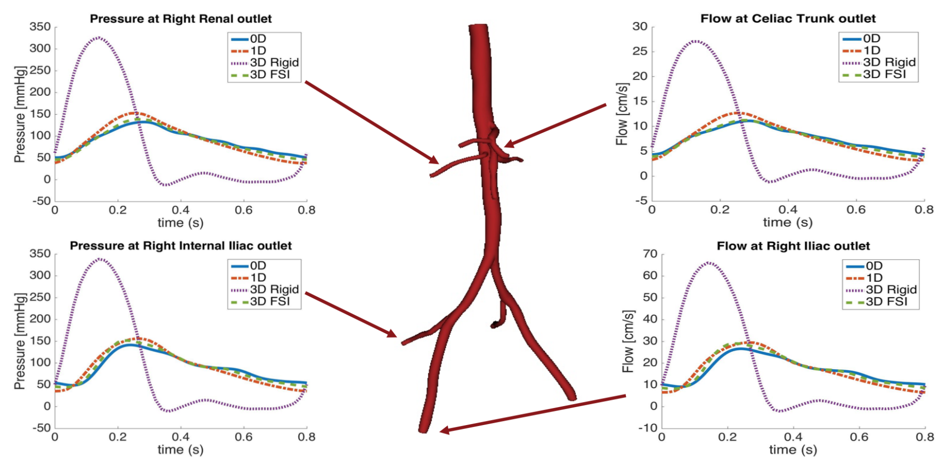


Fig. 5.1: Flows and pressure comparisons between model fidelities at selected outlets of aorto-femoral model with pulastile inlet flow and resistance outlet boundary conditions.

Learning the network

Preliminary results

- Nine uncertain parameters (outlet resistances)
- N = [100, 2000, 10000] available runs
- All models yield approximately same mean
- Does not seem to be enough evidence to use recursive or peer over independent

Solver	Cost (1 simulation)	Effective Cost (No. 3D Simulations)
3D	96 hr	1
1D	11.67 min	2E-3
0D	5 sec	1.45E-5

Log Odds Ratio

N	Rec	Peer	Indep
10/20/40	0.11	0.11	1
20/40/80	0.7	0.7	1
40/80/160	0.5	0.5	1

Summary and Future work

Takeaways

- Described a **modeling framework** for fusing multiple simulation models
- Parametric latent variable network can be learned, even from minimal data.
- **Learning relationships** instead of assuming them yields **robustness**
- Learning is important to take advantage of **smoothness** while avoiding errors made from model relationships

Future work

- Leverage a wide variety of new high dimensional Gaussian latent factor learning algorithms
- Use Bayesian score
- Use this framework within optimization, design, and control