SAND2022-11146C

Uncertainty Quantification with Multi-Fidelity Networks

USACM Thematic Conference on Uncertainty Quantification for Machine Learning Integrated Physics Modeling (UQ-MLIP)
Crystal City, Arlington, Virginia
August 18-19, 2022

Alex Gorodetsky

Aerospace Engineering, University of Michigan and
Sandia National Laboratories

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

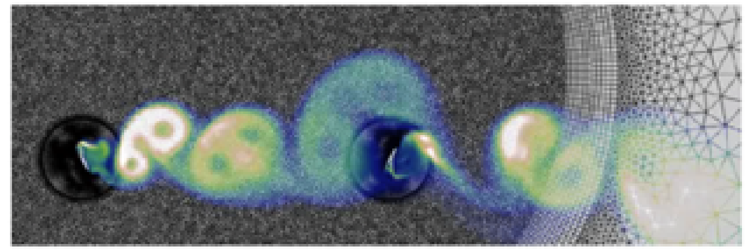
Collaborators

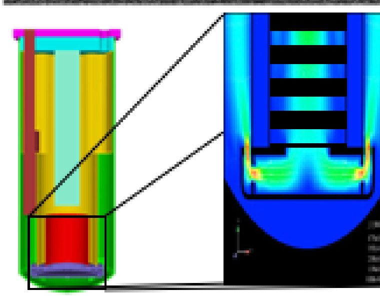
This work is a product of a joint effort

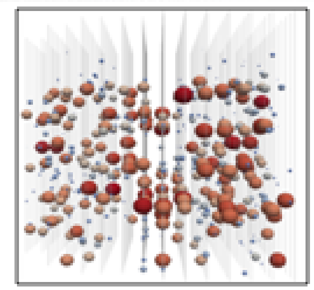
1. University of Michigan: Trung Pham
2. Sandia National Laboratories: Michael Eldred, Gianluca Geraci, John Jakeman

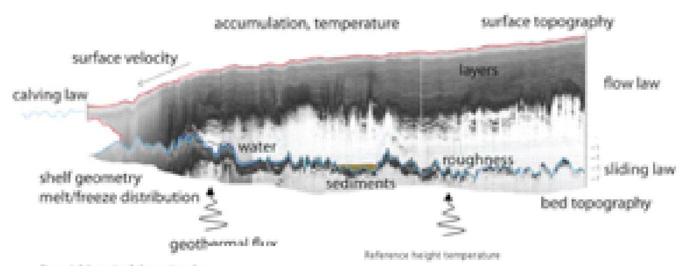
Uncertainty quantification for high-fidelity models

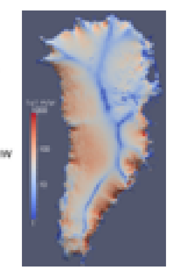
- Determine uncertainty on HF models
  - Expectation, variance
  - Probability of failures
- Severe simulation budget constraints
  - High dimensional PDEs
  - Large-scale computing resources
- Model complexity increases dimensionality
- HF models $\rightarrow$ UQ more important
  - Less validation, study, and analysis
  - Greater exploitation of nonlinearities
  - Uncertainty due to model form, initial conditions, and operating conditions

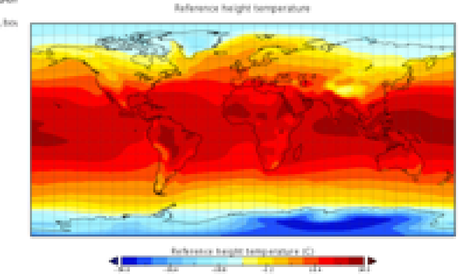










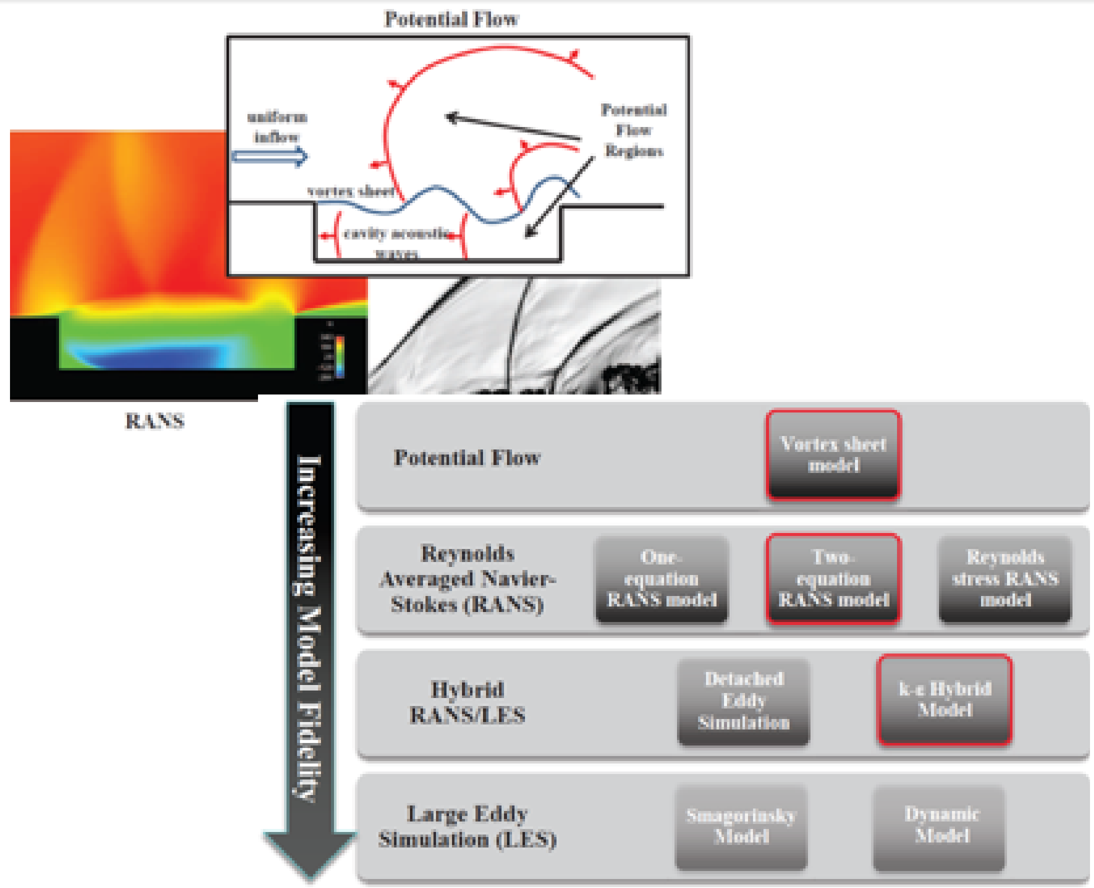


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data

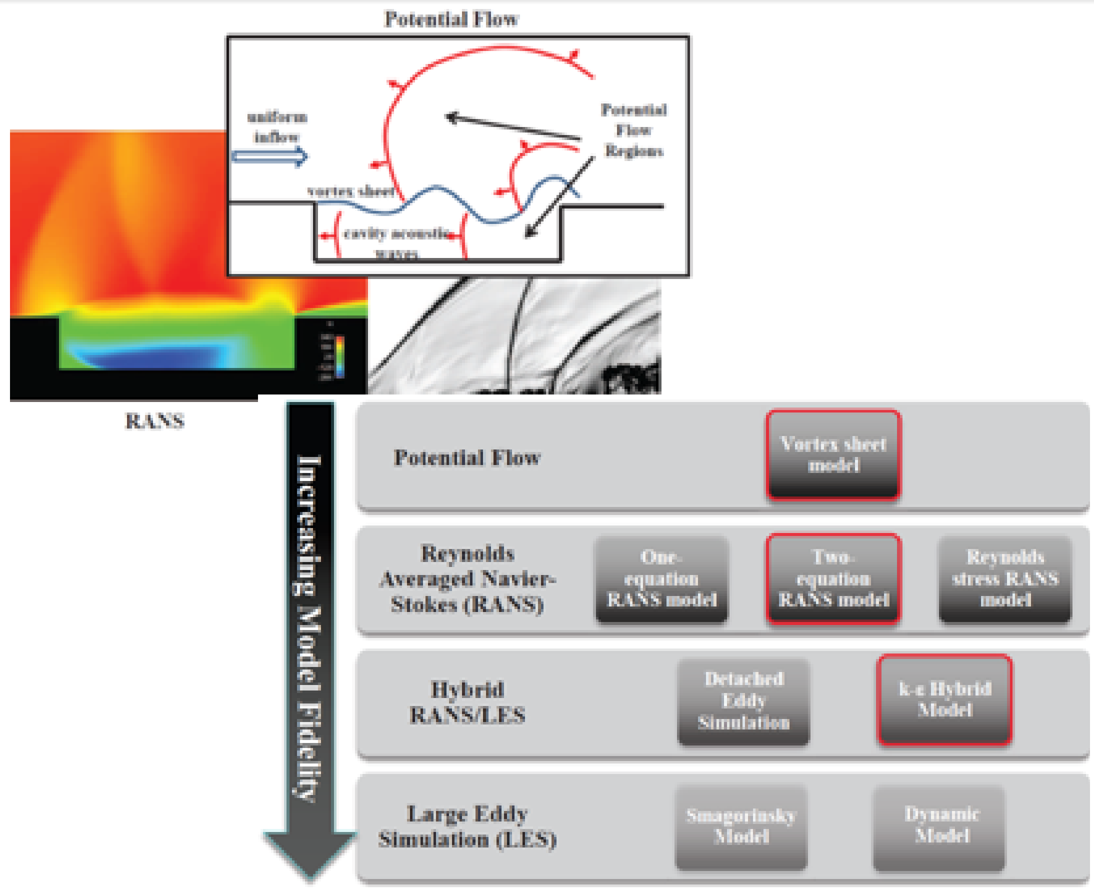


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data
- Challenges

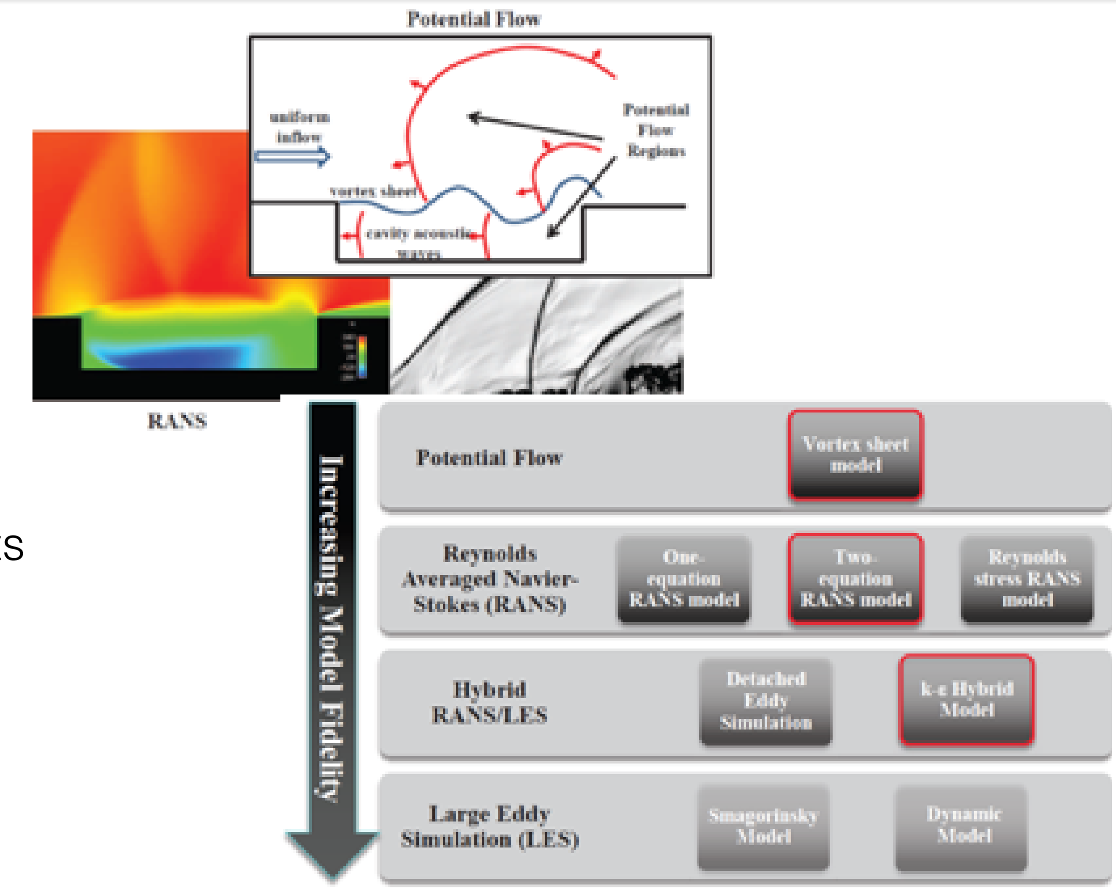


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data
- Challenges
  - Models with varying inputs/outputs

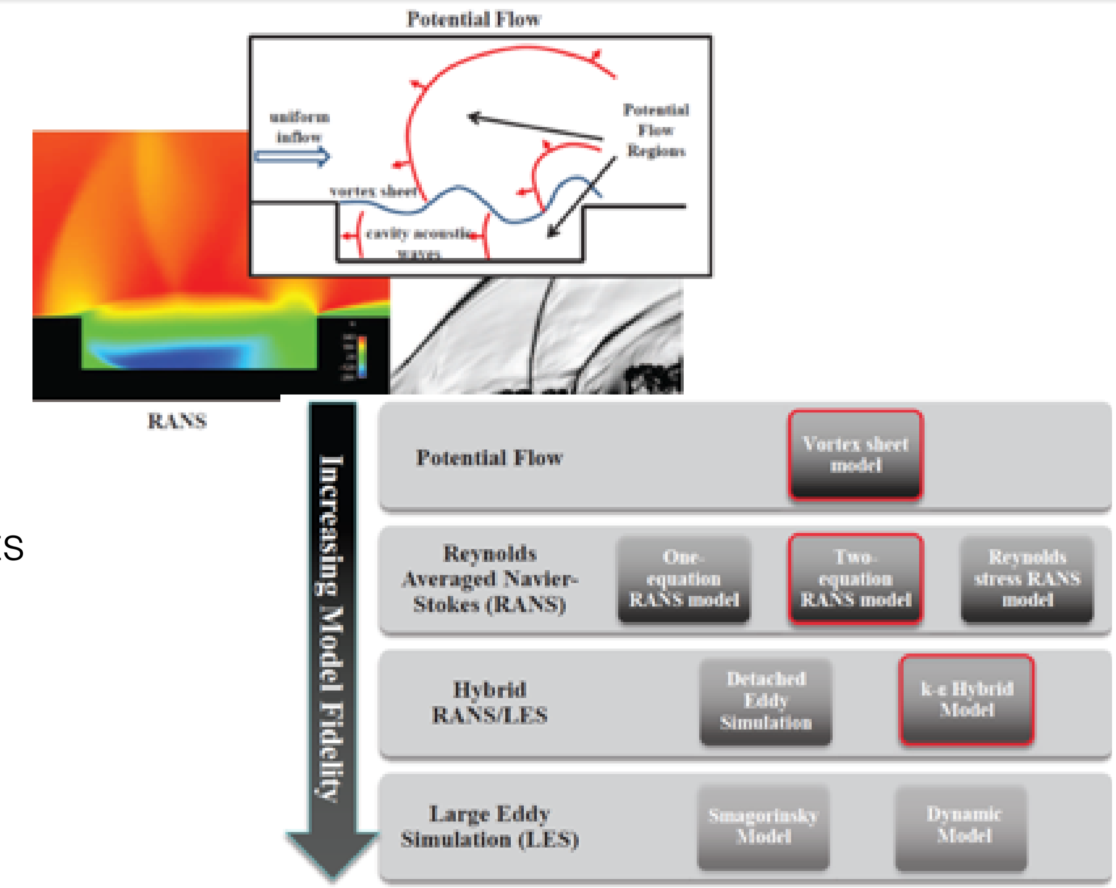


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data
- Challenges
  - Models with varying inputs/outputs
  - Legacy data collected separately

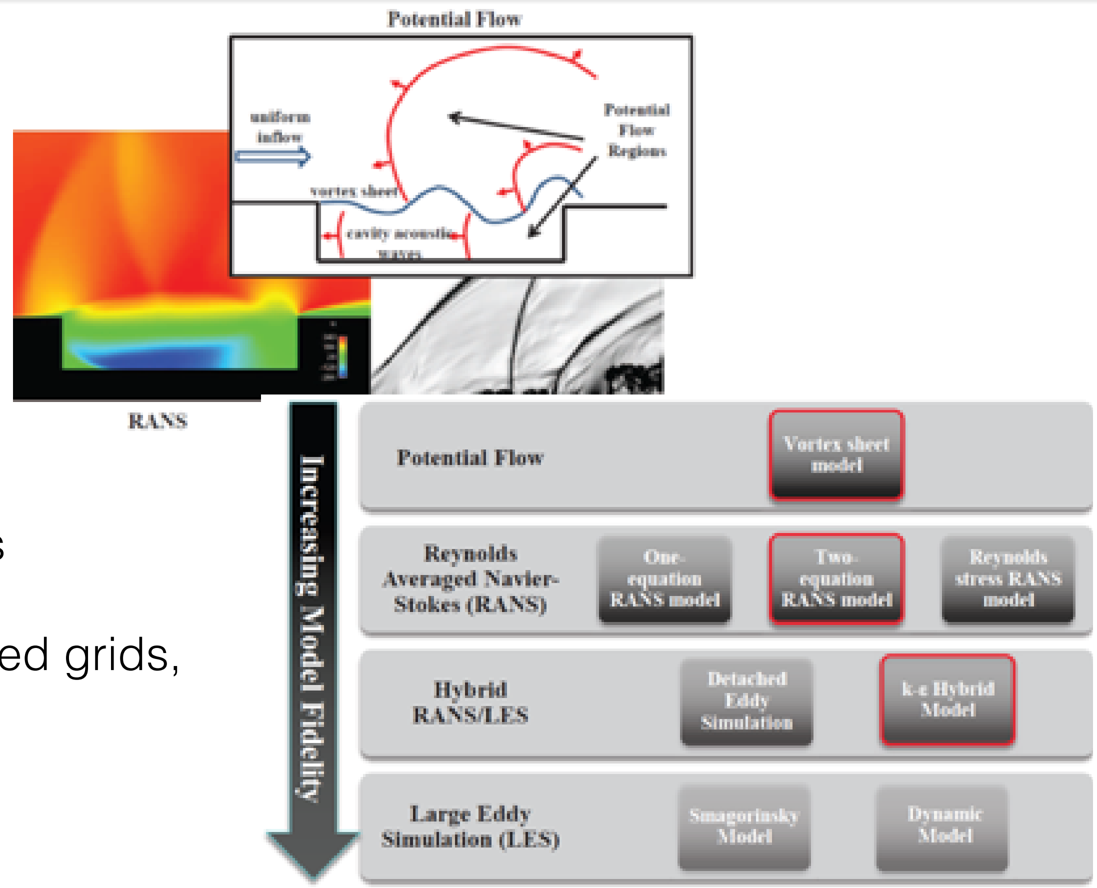


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data
- Challenges
  - Models with varying inputs/outputs
  - Legacy data collected separately
  - Corrupted evaluations: unconverged grids, unexplored parameters

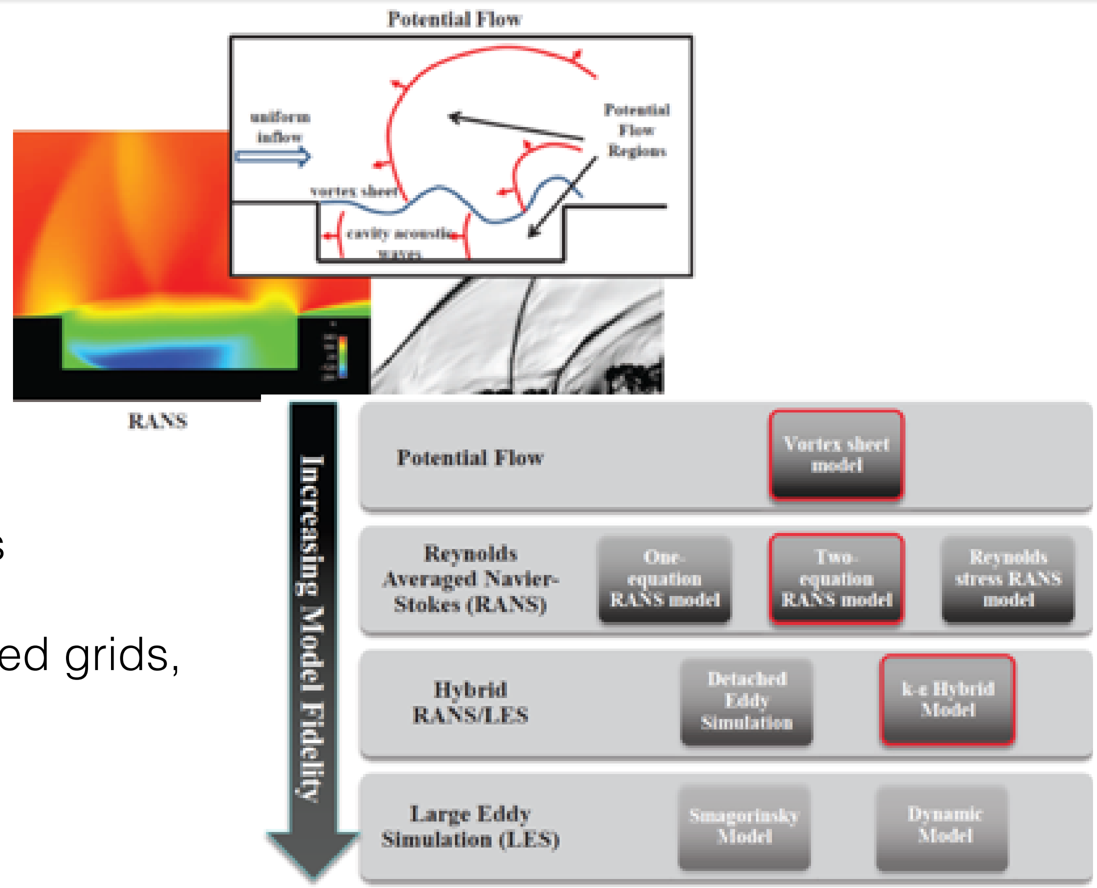


Leveraging multiple models/simulation sources

Key idea

We almost never have just *one* model, instead we develop many over the course of a study/analysis. Can we leverage these?

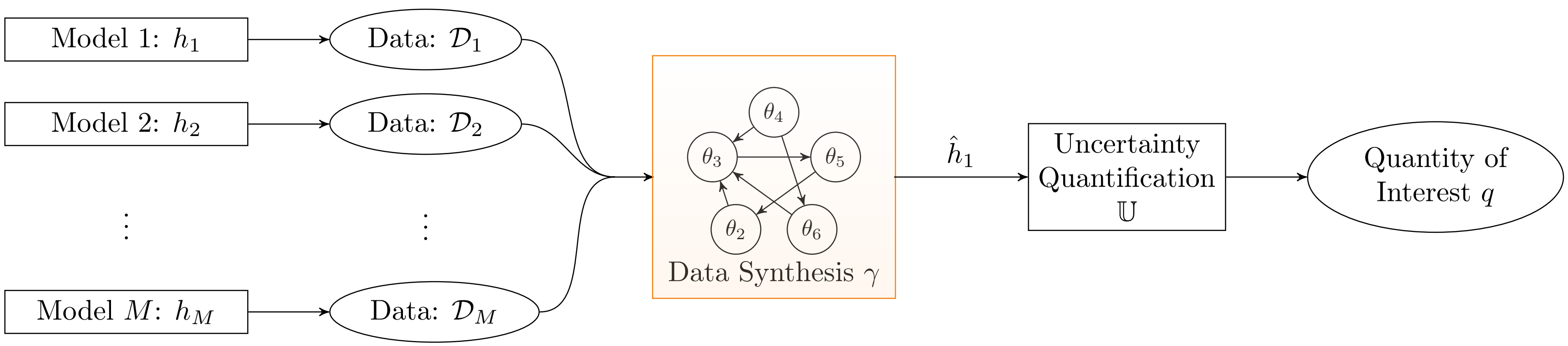
- Sources
  - Hierarchy of fidelities
  - Ensemble of peer models
  - Discretization levels
  - Experimental data
- Challenges
  - Models with varying inputs/outputs
  - Legacy data collected separately
  - Corrupted evaluations: unconverged grids, unexplored parameters
  - Lack of assumed relationships



Contribution in this talk

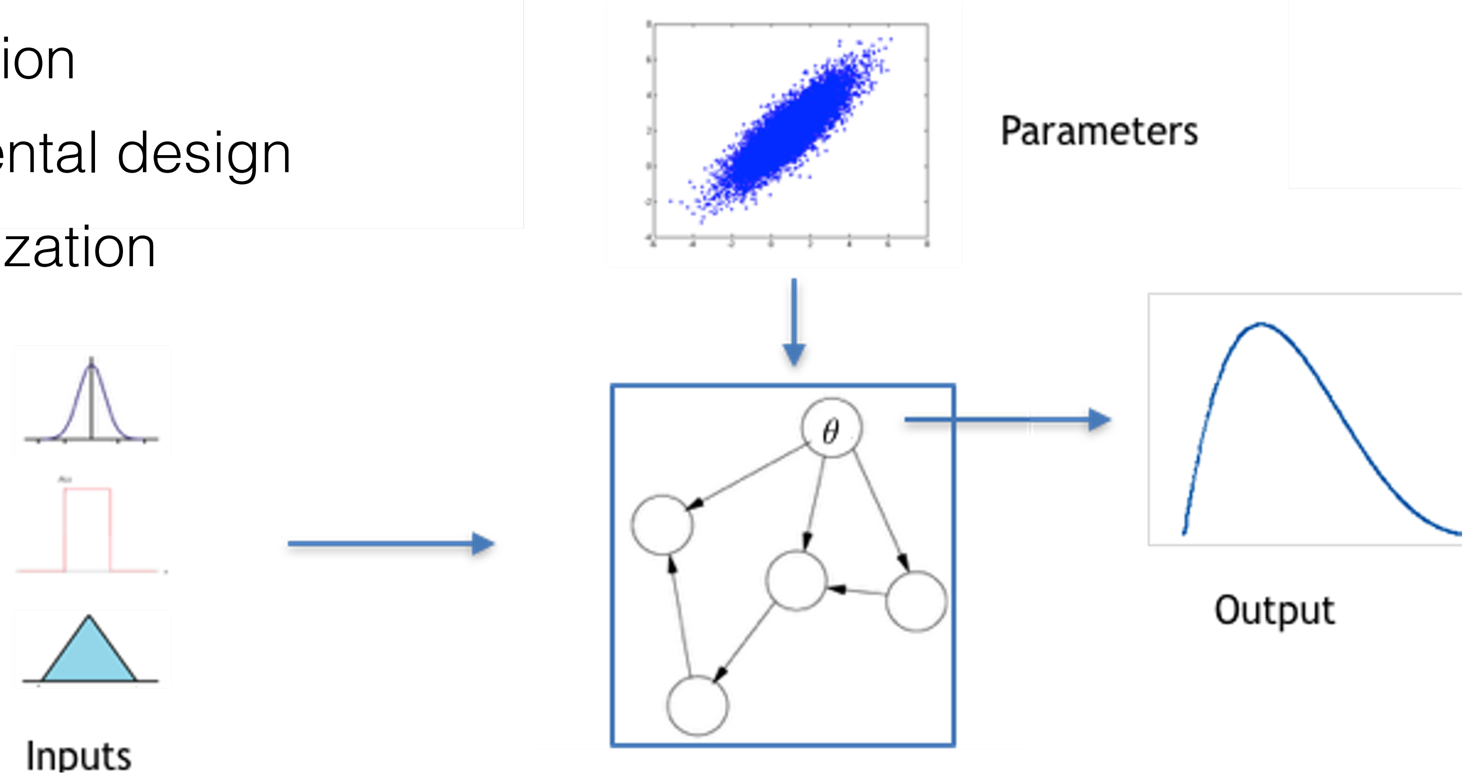
Problem: *A-priori* structural assumptions (hierarchical or otherwise) between models can lack robustness and limit efficiency

Contribution: A new modeling framework for multi-fidelity surrogate models that can flexibly adapt to variety of situations



Can be embedded within many aspects of UQ
Example: Bayesian Monte Carlo

1. Estimation of expectations
2. Within optimization
3. Within experimental design
4. Bayesian optimization



Ghahramani and Rasmussen *Bayesian Monte Carlo*, 2003

Outline

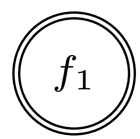
- Overview of existing multi-fidelity modeling paradigms
- MFNets: A graph-based generalization targeting heterogeneous ensemble
- Optimization formulation
- Examples

Existing approaches: discrepancies, recursivity, hierarchies, autoregressive

The standard approach: autoregressive (recursive) modeling

Two-model case

1. A single node represents a surrogate for some information source



The standard approach: autoregressive (recursive) modeling

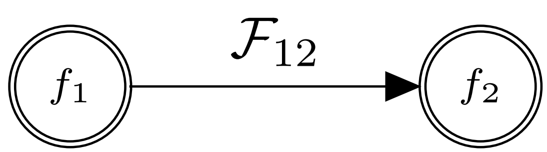
Two-model case

1. A single node represents a surrogate for some information source
2. A new (higher fidelity) information source is introduced

The standard approach: autoregressive (recursive) modeling

Two-model case

1. A single node represents a surrogate for some information source
2. A new (higher fidelity) information source is introduced
3. A recursive relationship is established, by passing f_1 through some functional $\mathcal{F}_{12}$

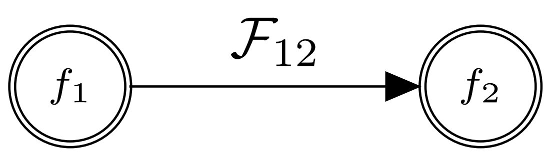


$$f_2 = \mathcal{F}_{12}[f_1] + \delta_2$$

The standard approach: autoregressive (recursive) modeling

Two-model case

1. A single node represents a surrogate for some information source
2. A new (higher fidelity) information source is introduced
3. A recursive relationship is established, by passing f_1 through some functional $\mathcal{F}_{12}$
4. The discrepancy is denoted by δ_2, hope is that it is "simple"
   - Recursive co-kriging: Kennedy and O'Hagan, Gratiet and Garnier

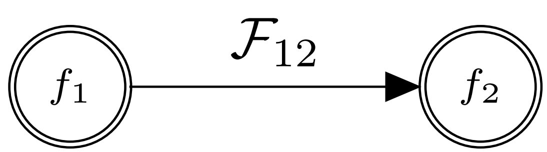


$$\delta_2 = f_2 - \mathcal{F}_{12}[f_1]$$

The standard approach: autoregressive (recursive) modeling

Two-model case

1. A single node represents a surrogate for some information source
2. A new (higher fidelity) information source is introduced
3. A recursive relationship is established, by passing f_1 through some functional $\mathcal{F}_{12}$
4. The discrepancy is denoted by δ_2, hope is that it is "simple"
    - Recursive co-kriging: Kennedy and O'Hagan, Gratiet and Garnier



$$\delta_2 = f_2 - \mathcal{F}_{12}[f_1]$$

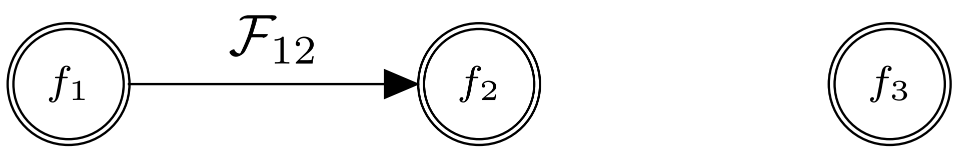
Learning Requirements

- Things to learn: f_1, $\mathcal{F}_{12}$, δ_2
- If any are nonlinearly parameterized, then not a Gaussian process!

The standard approach: autoregressive (recursive) modeling

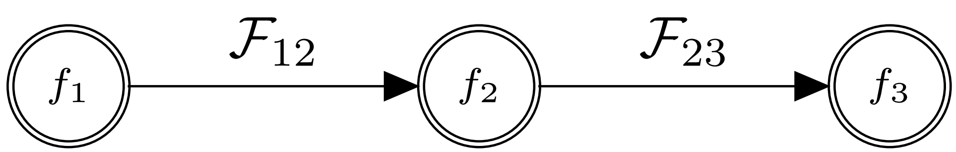
General case

1. Introduce a third (higher fidelity) information source



The standard approach: autoregressive (recursive) modeling
General case

1. Introduce a third (higher fidelity) information source
2. Again establish a recursive relationship

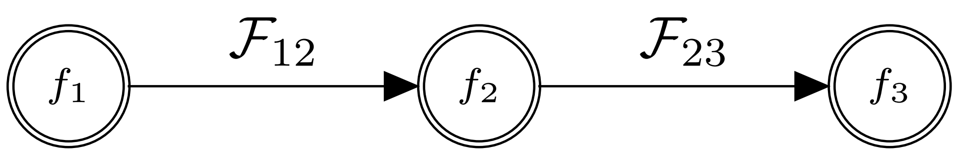


$$f_3 = \mathcal{F}_{23}[f_2] + \delta_3 = \mathcal{F}_{23}\left[\mathcal{F}_{12} f_1 + \delta_2\right] + \delta_3$$

The standard approach: autoregressive (recursive) modeling

General case

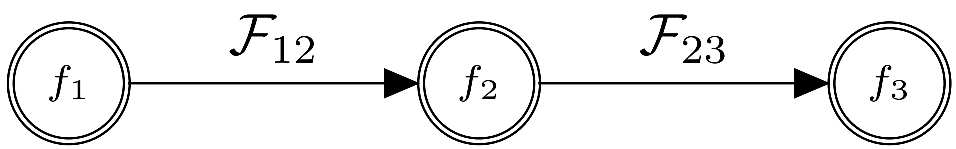
1. Introduce a third (higher fidelity) information source
2. Again establish a recursive relationship
3. Again learn the discrepancy



$$\delta_3 = f_3 - \mathcal{F}_{23}[f_{23}]$$

The standard approach: autoregressive (recursive) modeling
General case

1. Introduce a third (higher fidelity) information source
2. Again establish a recursive relationship
3. Again learn the discrepancy



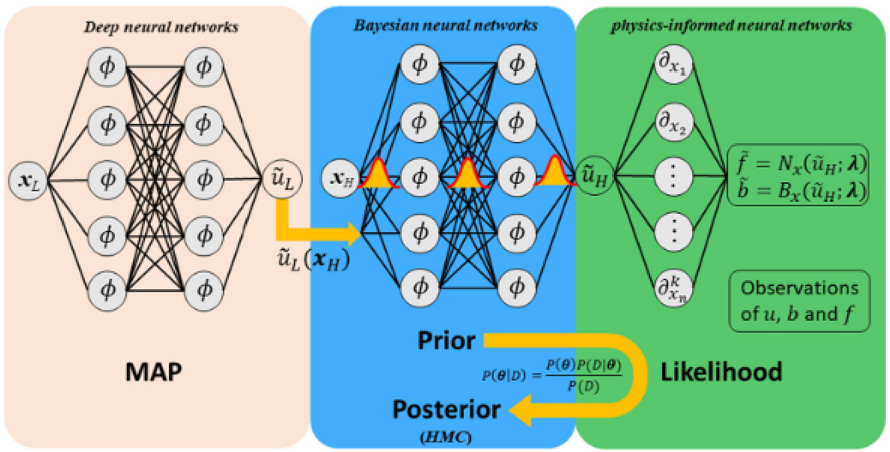
$$\delta_3 = f_3 - \mathcal{F}_{23}[f_{23}]$$

Challenges

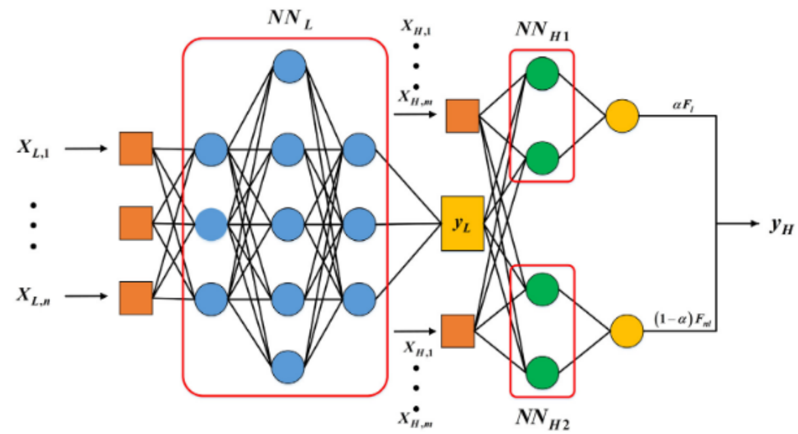
- Techniques often require nested sample points and step-wise optimization [Gratiet 2014, Perdikaris 2017]
- Many approx. formats: GP, PCE, Radial Basis, Neural Networks, etc.
- In GP: techniques that assume separable $\mathcal{F}_{ij}[f_i] = \rho_{ij}(x) f_i(x)$ treat parameters of ρ_{ij} as hyperparameters that are optimized separately to retain Gaussianity.

Examples in cases with two models

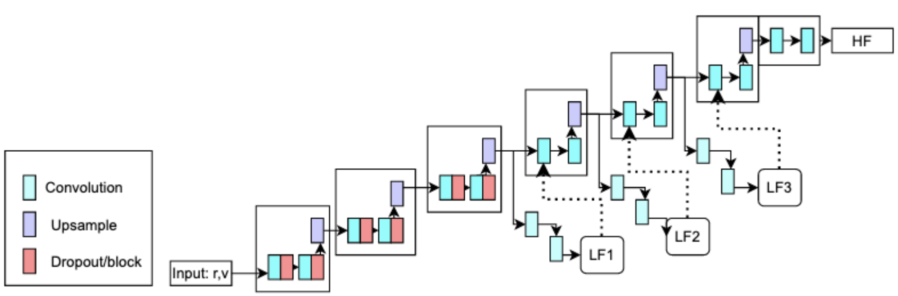
While extremely classical, still many new papers with new $\mathcal{F}$



Meng, Babaee, Karniadakis, 2021



Zhang, Xie, Ti, Zhu, Zheng, 2021



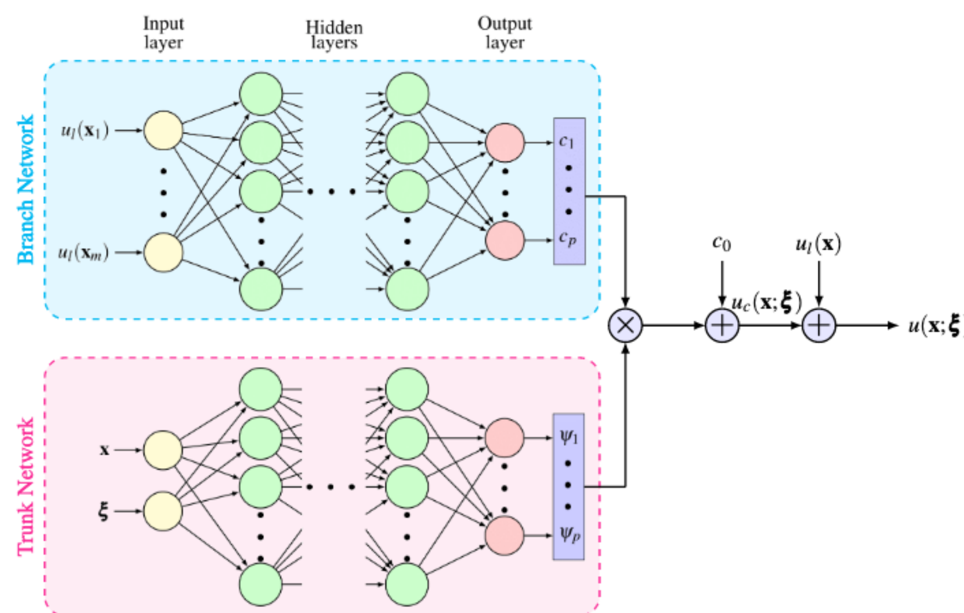
Partin, Geraci, Rushdi, Eldred, Schiavazzi, 2022

$\mathcal{F}$ has a NN structure

Examples in cases with two models

While extremely classical, still many new papers with new $\mathcal{F}$

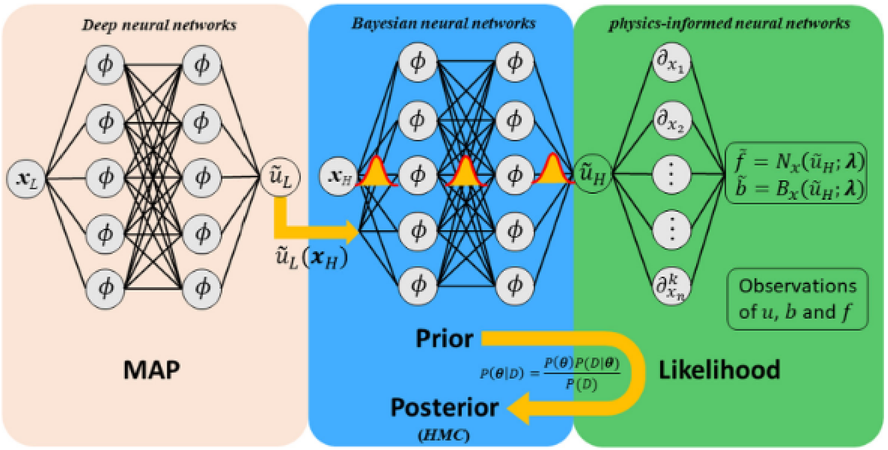
Discrepancy has a NN structure



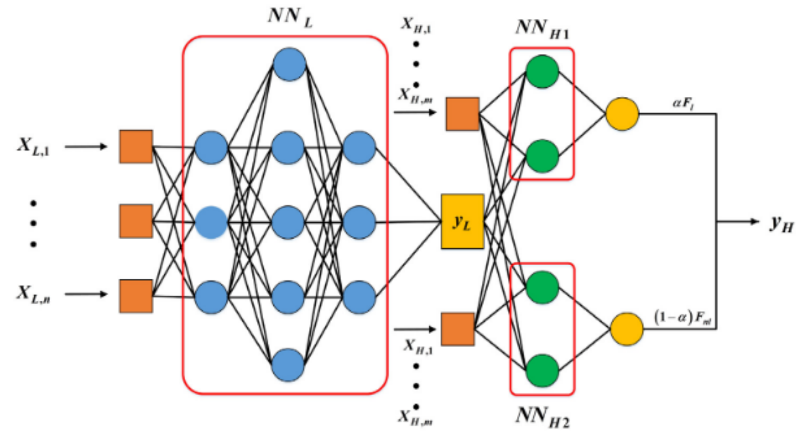
De, Hassanaly, Reynolds, King, Doostan, 2022

Examples in cases with two models

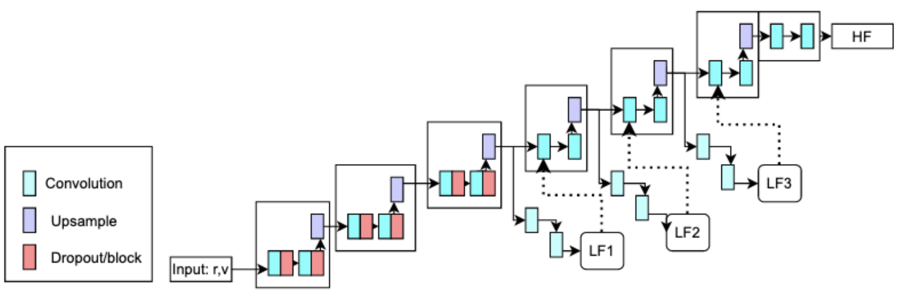
While extremely classical, still many new papers with new $\mathcal{F}$



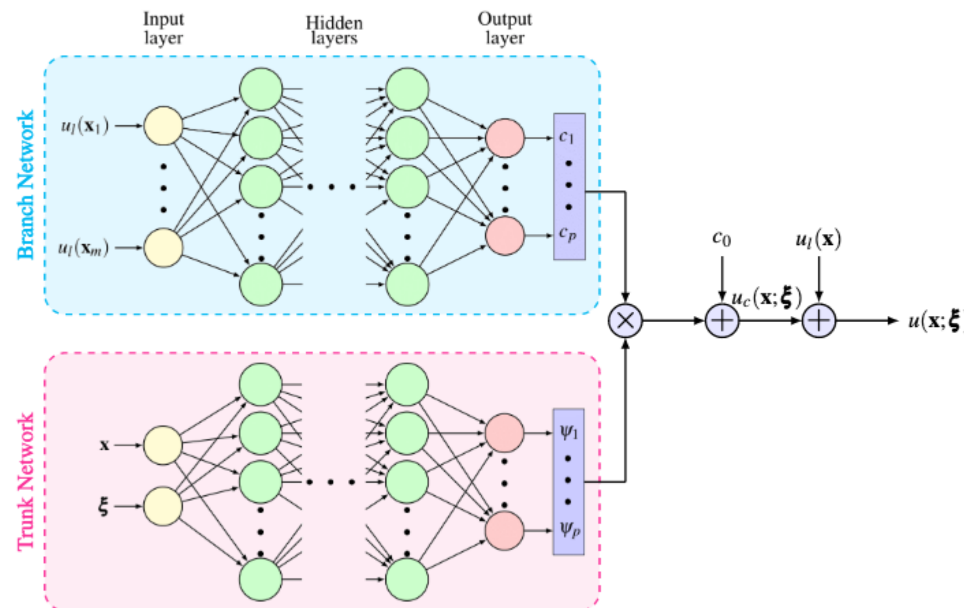
Meng, Babaee, Karniadakis, 2021



Zhang, Xie, Ti, Zhu, Zheng, 2021



Partin, Geraci, Rushdi, Eldred, Schiavazzi, 2022



De, Hassanaly, Reynolds, King, Doostan, 2022

A new approach: MFNETS

Why do we need another paradigm?

1. Not all information sources are obviously hierarchical
   - Depends on QoI
   - Peer models are common
   - Discretizations and numerical issues affect hierarchies

Why do we need another paradigm?

1. Not all information sources are obviously hierarchical
   - Depends on QoI
   - Peer models are common
   - Discretizations and numerical issues affect hierarchies
2. In the sampling context, moving beyond hierarchical can often extract more correlations
   - MIMC [Haji-Ali 2016] vs. MLMC [Giles 2008]
   - Generalized ACV [Gorodetsky 2020] vs. recursive control variate approaches (MFMC) [Peherstorfer 2016]
   - Multilevel Blue [Schaden and Ullman 2020]

Why do we need another paradigm?

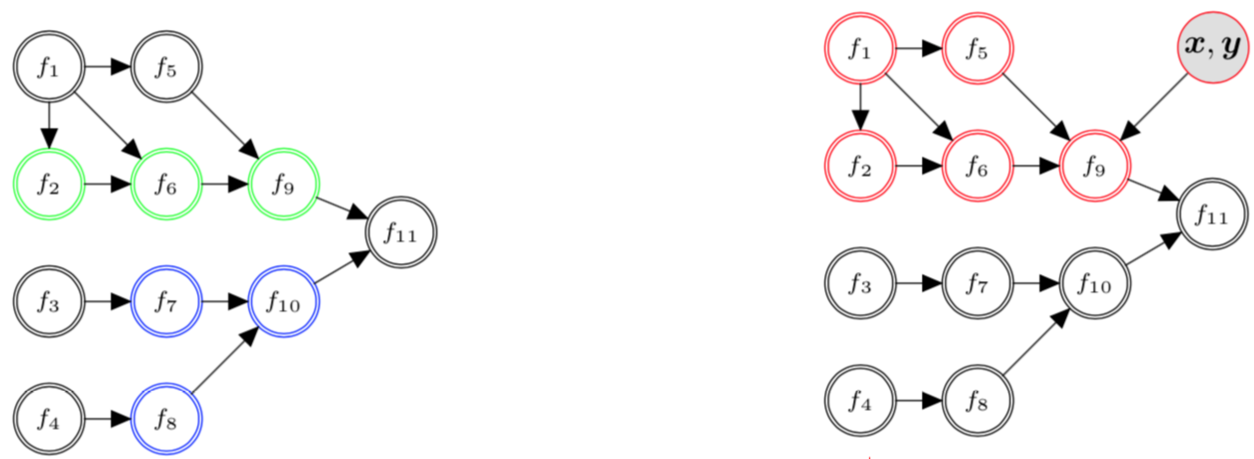
1. Not all information sources are obviously hierarchical
   - Depends on QoI
   - Peer models are common
   - Discretizations and numerical issues affect hierarchies
2. In the sampling context, moving beyond hierarchical can often extract more correlations
   - MIMC [Haji-Ali 2016] vs. MLMC [Giles 2008]
   - Generalized ACV [Gorodetsky 2020] vs. recursive control variate approaches (MFMC) [Peherstorfer 2016]
   - Multilevel Blue [Schaden and Ullman 2020]
3. Need more training algorithms to consider non-GP cases and non-nested sampling

Multifidelity Networked Surrogate Models (MFNets)

Main idea: model relationships between surrogate outputs

An MFNet is a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes representing information sources $\mathcal{V} = \{f_1, \ldots, f_M\}$ and directed edges $\mathcal{E} = \{(j \to i)\}$ representing their relationships

$$f_i(x) = \mathcal{F}_i[\{f_j(x); j \in \text{pa}(i)\}] + \delta_i(x).$$



(a) Sample structure of a multifidelity surrogate.

(b) Evaluating (10) for $k = 9$ requires data from f_9 and traversing the ancestors of f_9 (depicted in red).

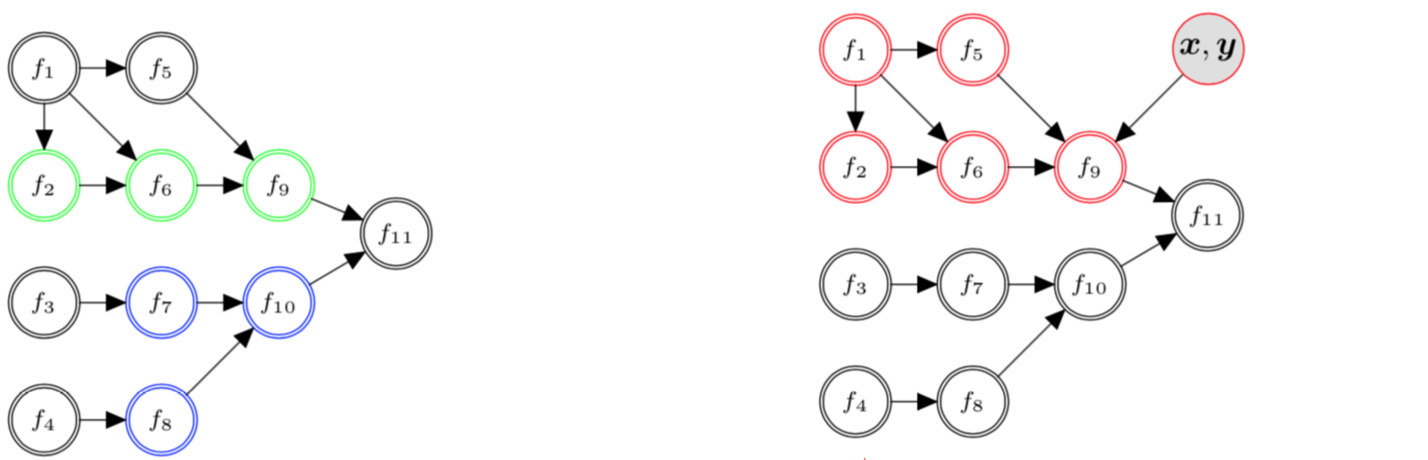
Figure 1: An example DAG used to define a multifidelity surrogate. This structure exhibits a complicated relationship between each function and the high-fidelity f_{11}. Both hierarchical and peer relationships are exhibited within these networks. For instance the left panel shows an example of hierarchical structure ($f_2 \to f_6 \to f_9$) in green and example of peer structure ($f_7 \to f_{10}, f_8 \to f_{10}$) in blue.

Multifidelity Networked Surrogate Models (MFNets)

Main idea: model relationships between surrogate outputs

An MFNet is a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes representing information sources $\mathcal{V} = \{f_1, \ldots, f_M\}$ and directed edges $\mathcal{E} = \{(j \to i)\}$ representing their relationships

$$f_i(x) = \sum_{j \in \text{pa}(i)} \rho_{ji}(x) f_j(x) + \delta_i(x).$$

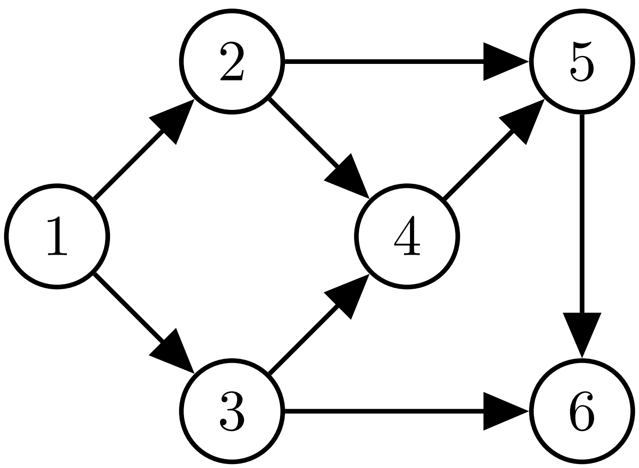


(a) Sample structure of a multifidelity surrogate.

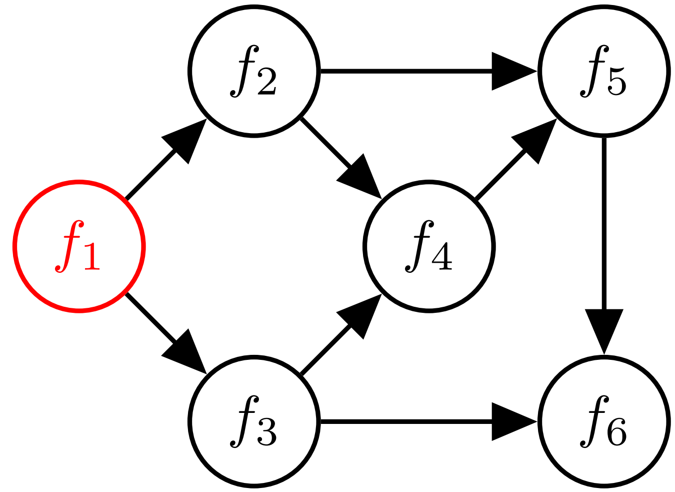
(b) Evaluating (10) for $k = 9$ requires data from f_9 and traversing the ancestors of f_9 (depicted in red).

Figure 1: An example DAG used to define a multifidelity surrogate. This structure exhibits a complicated relationship between each function and the high-fidelity f_{11}. Both hierarchical and peer relationships are exhibited within these networks. For instance the left panel shows an example of hierarchical structure ($f_2 \to f_6 \to f_9$) in green and example of peer structure ($f_7 \to f_{10}, f_8 \to f_{10}$) in blue.

Linear parametric approximations



(a) DAG.



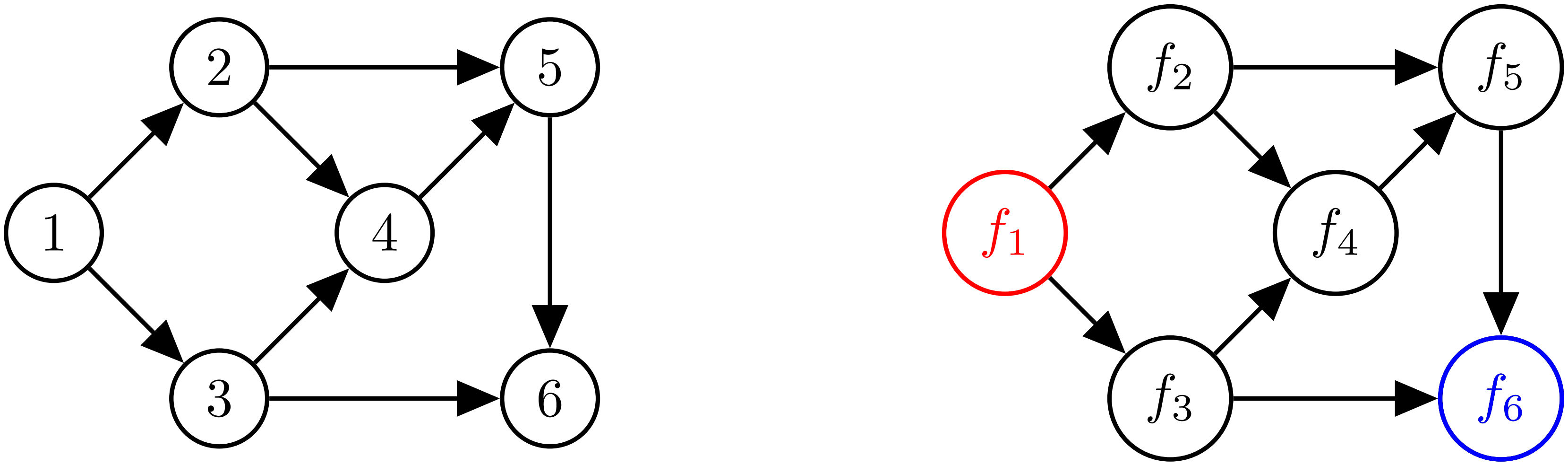
(b) Multi-fidelity surrogate.

Figure: From graph to surrogate.

For this graph, we have

$$f_1(x) = \delta_1(x) = V_1^\top(x)\beta_1$$

Linear parametric approximations



(a) DAG. (b) Multi-fidelity surrogate.

Figure: From graph to surrogate.

For this graph, we have

$$f_1(x) = \delta_1(x) = V_1^\top(x)\beta_1$$

$$f_6(x) = \sum_{j \in \text{pa}(6)} \rho_{j6}(x) f_j(x) + \delta_6(x) = \rho_{36}(x) f_3(x) + \rho_{56}(x) f_5(x) + \delta_6(x),$$

where $\rho_{36}(x) = W_{36}^\top(x)\alpha_{36}$, $\rho_{56}(x) = W_{56}^\top(x)\alpha_{56}$, and $\delta_6(x) = V_6^\top(x)\beta_6$.

Motivating Gibbs sampling: separability in expansion

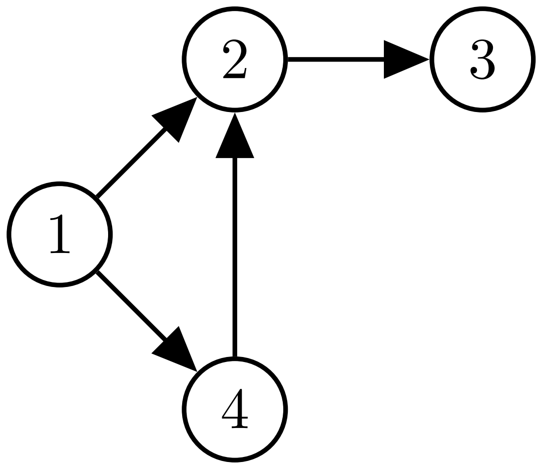


Figure: Directed acyclic graph (DAG).

$$f_3(x) = V_1^\top(x)\,\beta_1\,(\rho_{12}\rho_{23} + \rho_{14}\rho_{42}\rho_{23}) + V_4^\top(x)\,\beta_4\rho_{42}\rho_{23} + V_2^\top(x)\,\beta_2\rho_{23} + V_3^\top(x)\,\beta_3, \text{ where } \rho_{ji}(x) = W_{ji}^\top(x)\alpha_{ji}$$

Motivating Gibbs sampling: separability in expansion

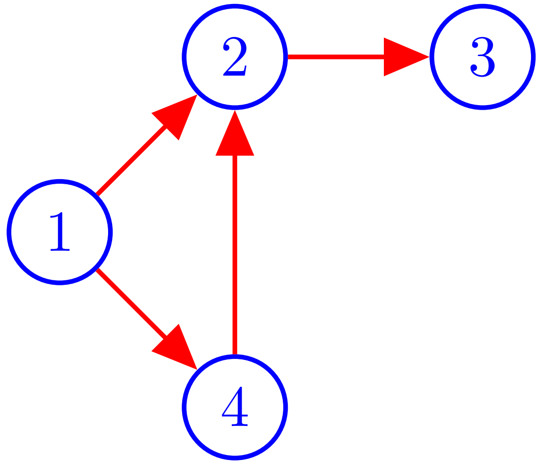


Figure: Directed acyclic graph (DAG).

$$f_3(x) = V_1^\top(x)\,\beta_1\left(\rho_{12}\rho_{23} + \rho_{14}\rho_{42}\rho_{23}\right) + V_4^\top(x)\,\beta_4\rho_{42}\rho_{23} + V_2^\top(x)\,\beta_2\rho_{23} + V_3^\top(x)\,\beta_3, \text{ where } \rho_{ji}(x) = W_{ji}^\top(x)\alpha_{ji}$$

If all edge paramters $\{\alpha_{ji}\}_{j,i=1}^{M}$ are fixed, the functions $\{f_i\}_{i=1}^{M}$ are linear w.r.t the node parameters $\{\beta_i\}_{i=1}^{m_\beta}$.

Motivating Gibbs sampling: separability in expansion

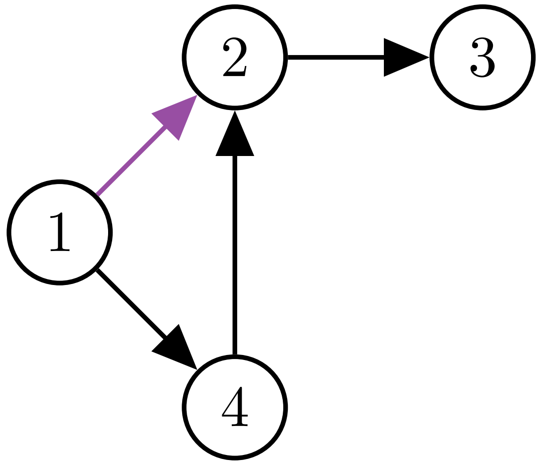


Figure: Directed acyclic graph (DAG).

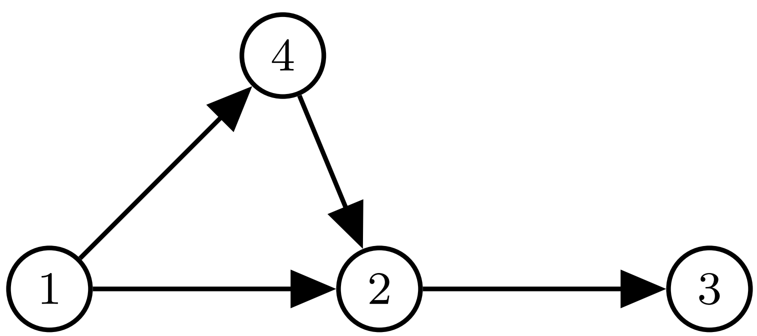
$$f_3(x) = V_1^\top(x)\,\beta_1\left(\rho_{12}\rho_{23} + \rho_{14}\rho_{42}\rho_{23}\right) + V_4^\top(x)\,\beta_4\rho_{42}\rho_{23} + V_2^\top(x)\,\beta_2\rho_{23} + V_3^\top(x)\,\beta_3, \text{ where } \rho_{ji}(x) = W_{ji}^\top(x)\alpha_{ji}$$

If all paramters except α_{ji} are fixed, the functions $\{f_i\}_{i=1}^{m_f}$ are linear w.r.t the edge parameter α_{ji}.

Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

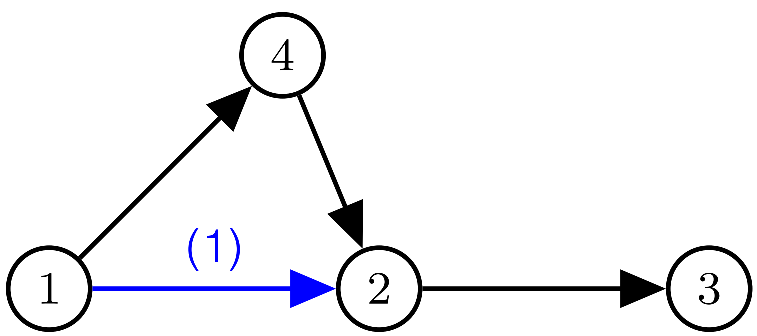
1: Set inital parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **for each** edge e of $\mathcal{G}$ **do**
4: Draw a sample $\alpha_e^{(i)}$ from $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
5: Update $\mathcal{G}$ using $\alpha_e^{(i)}$
6: **end for**
7: Draw a sample $\beta^{(i)}$ from $p\left(\beta \mid \mathcal{D}, \hat{\alpha}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
8: Update $\mathcal{G}$ using $\beta^{(i)}$
9: Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
10: **end for**



Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

1: Set inital parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **for each** edge e of $\mathcal{G}$ **do**
4: Draw a sample $\alpha_e^{(i)}$ from $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
5: Update $\mathcal{G}$ using $\alpha_e^{(i)}$
6: **end for**
7: Draw a sample $\beta^{(i)}$ from $p(\beta \mid \mathcal{D}, \hat{\alpha}) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
8: Update $\mathcal{G}$ using $\beta^{(i)}$
9: Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
10: **end for**

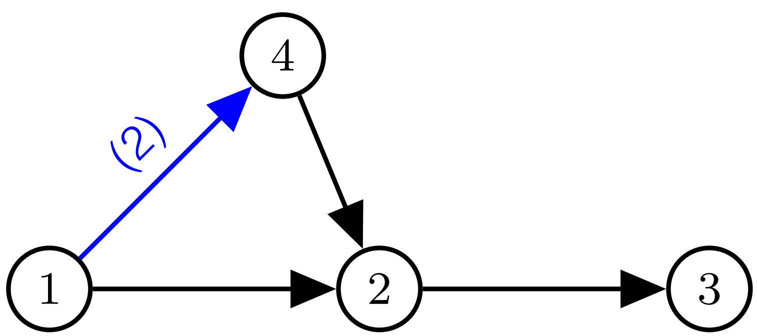


Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

```
Set inital parameters α_e^(0) for e ∈ ℰ and β^(0)
for i = 1, 2, ..., n do
    for each edge e of 𝒢 do
        Draw a sample α_e^(i) from p(α_e | 𝒟, α̃, β̃) = 𝒩_p(m_p^(α_e), Σ_p^(α_e))
        Update 𝒢 using α_e^(i)
    end for
    Draw a sample β^(i) from p(β | 𝒟, α̂) = 𝒩_p(m_p^(β), Σ_p^(β))
    Update 𝒢 using β^(i)
    Collect i^th sample {α_e^(i) for e ∈ ℰ, β^(i)} of the parameters
end for
```

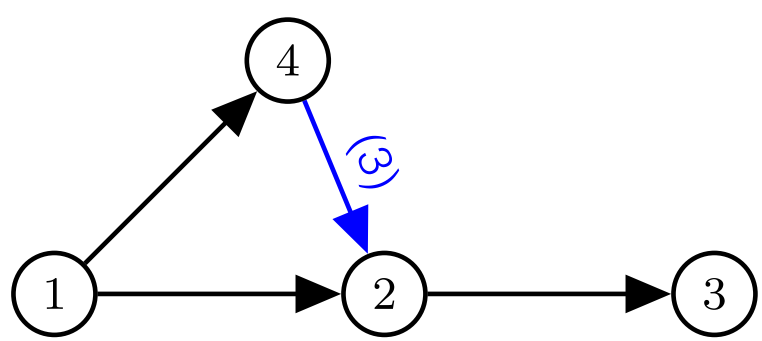
Where: line 1: $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$; line 4: $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$; line 7: $p\left(\beta \mid \mathcal{D}, \hat{\alpha}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$; line 9: $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$.



Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

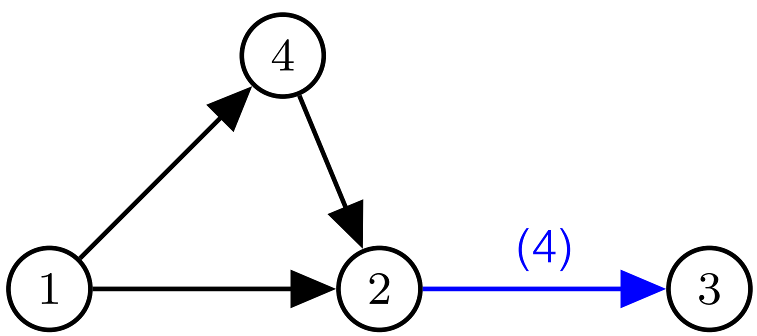
1: Set inital parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **for each** edge e of $\mathcal{G}$ **do**
4: Draw a sample $\alpha_e^{(i)}$ from $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
5: Update $\mathcal{G}$ using $\alpha_e^{(i)}$
6: **end for**
7: Draw a sample $\beta^{(i)}$ from $p\left(\beta \mid \mathcal{D}, \hat{\alpha}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
8: Update $\mathcal{G}$ using $\beta^{(i)}$
9: Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
10: **end for**



Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

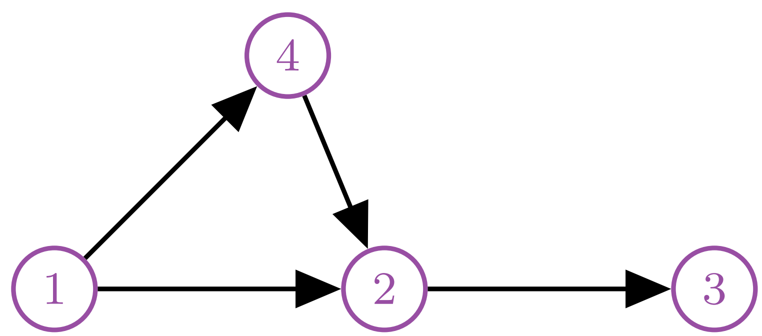
1: Set inital parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **for each** edge e of $\mathcal{G}$ **do**
4: Draw a sample $\alpha_e^{(i)}$ from $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
5: Update $\mathcal{G}$ using $\alpha_e^{(i)}$
6: **end for**
7: Draw a sample $\beta^{(i)}$ from $p\left(\beta \mid \mathcal{D}, \hat{\alpha}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
8: Update $\mathcal{G}$ using $\beta^{(i)}$
9: Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
10: **end for**



Gibbs sampling: Algorithm

Require: MFNets $\mathcal{G}$; input-output pairs $\mathcal{D}$; number of samples n;
Ensure: n samples of the parameters

1: Set inital parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **for each** edge e of $\mathcal{G}$ **do**
4: Draw a sample $\alpha_e^{(i)}$ from $p\left(\alpha_e \mid \mathcal{D}, \tilde{\alpha}, \tilde{\beta}\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
5: Update $\mathcal{G}$ using $\alpha_e^{(i)}$
6: **end for**
7: Draw a sample $\beta^{(i)}$ from $p\left(\beta \mid \mathcal{D}, \hat{\alpha}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
8: Update $\mathcal{G}$ using $\beta^{(i)}$
9: Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
10: **end for**



Examples

Synthetic example with stochastic models

UNIVERSITY OF MICHIGAN

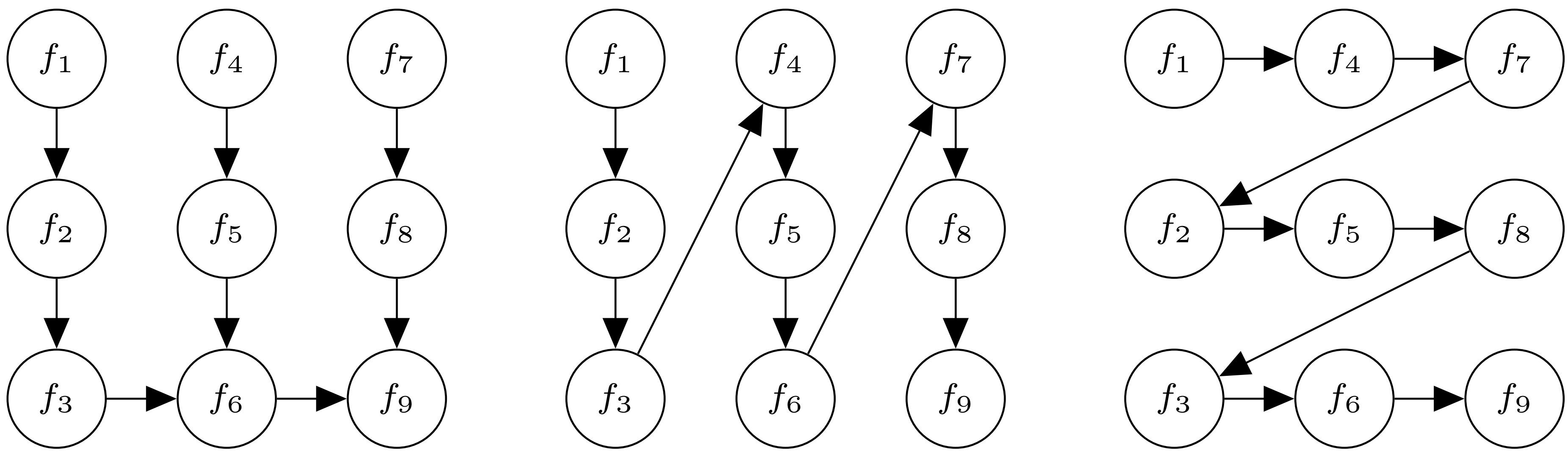
$$f_k(x) = \left(2 + (2x_1^5 + 2x_2^5)\Delta_1 + 3x_1x_2 + (x_1^2 + x_2^2 + 5x_1^2x_2^2)\Delta_2 + 0.5x_1 + 0.5x_2\right)(1 + \mathbb{E}[\mathcal{N}(0,1)])\,,$$

	Δ_1	Δ_2	N (Monte Carlo)
f_1	0	0	5
f_2	0	0	10
f_3	0	0	100
f_4	0	1	5
f_5	0	1	10
f_6	0	1	100
f_7	1	1	5
f_8	1	1	10
f_9	1	1	100

Synthetic example with stochastic models

UNIVERSITY OF MICHIGAN

$$f_k(x) = \left(2 + (2x_1^5 + 2x_2^5)\Delta_1 + 3x_1x_2 + (x_1^2 + x_2^2 + 5x_1^2x_2^2)\Delta_2 + 0.5x_1 + 0.5x_2\right)\left(1 + \mathbb{E}[\mathcal{N}(0,1)]\right),$$



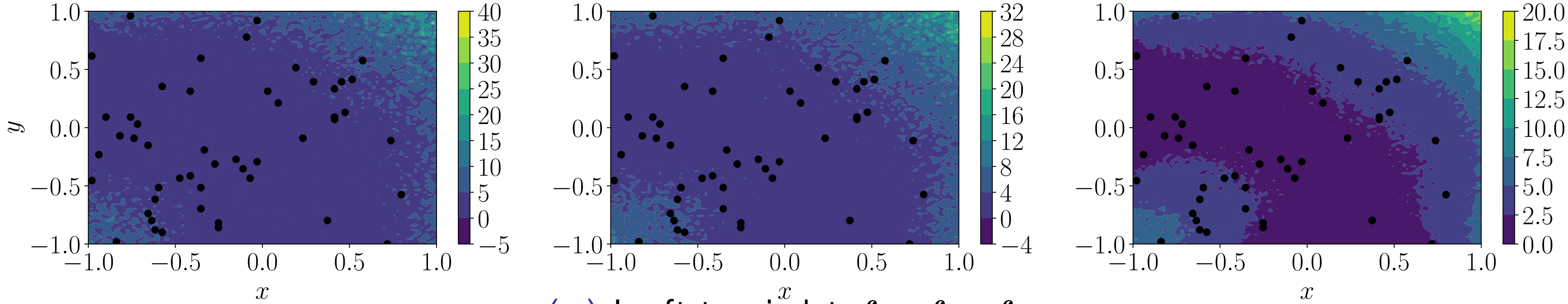
(a) Natural ordering

(b) Hierarchical ordering by model fidelity Δ first

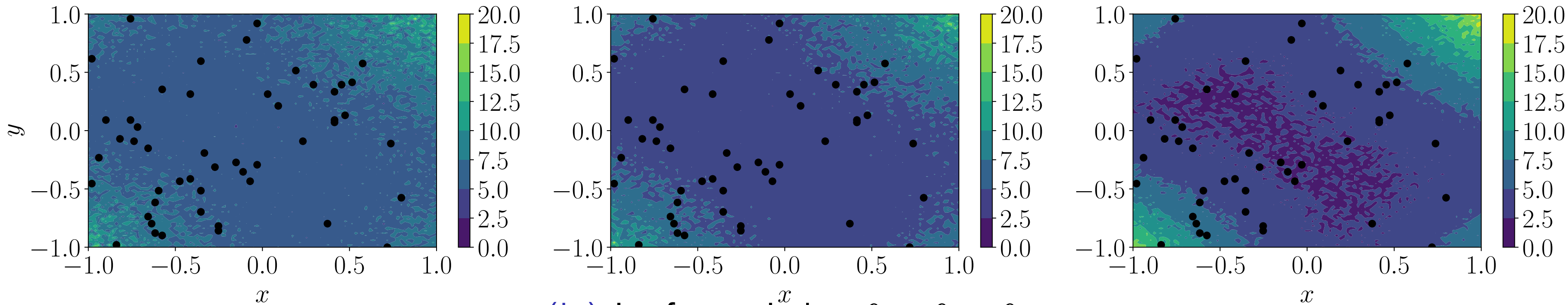
(c) Alternate Hierarchical ordering by noise fidelity N first.

Figure: Analytical noise test case: models' natural structure versus two candidate hierarchical orderings.

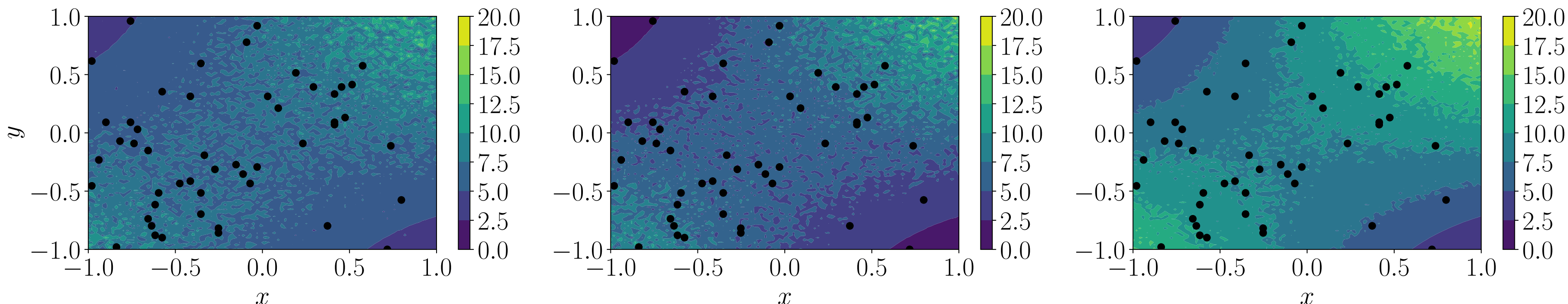
Synthetic response surfaces



(a) Left to right f_7, f_8, f_9



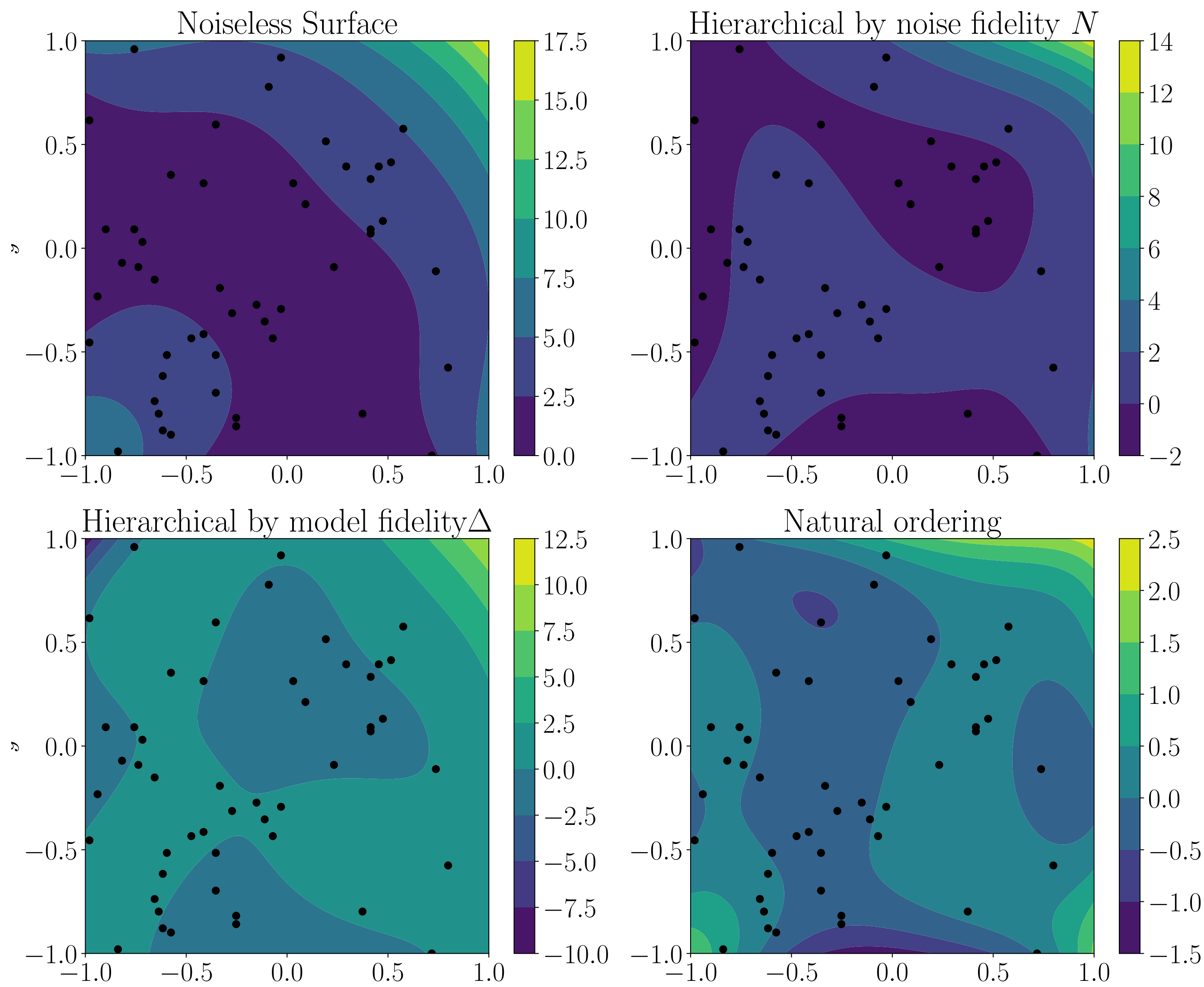
(b) Left to right f_4, f_5, f_6



(c) Left to right f_1, f_2, f_3

Highest fidelity model and pointwise errors

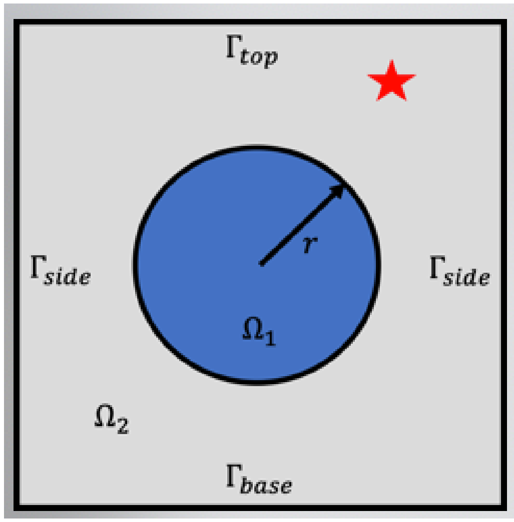
Natural ordering obtains order of magnitude lower errors



Thermal block

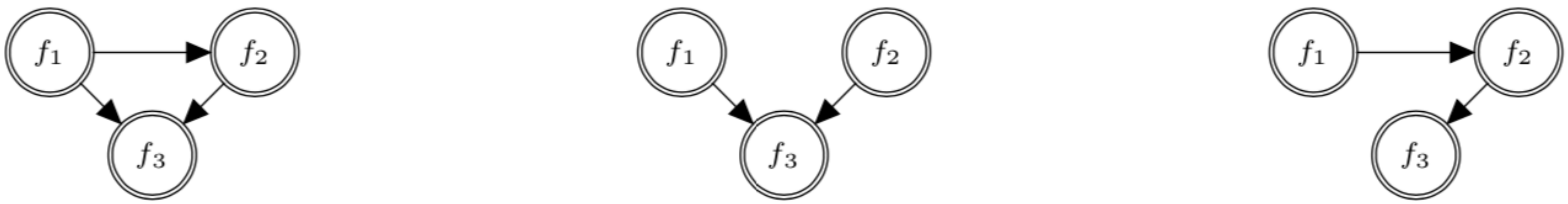
$$\begin{cases} -\text{div}(\kappa(\mu_0)\nabla u(\boldsymbol{\mu})) = 0 & \text{in } \Omega, \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_{top}, \\ \kappa(\mu_0)\nabla u(\boldsymbol{\mu}) \cdot \mathsf{n} = 0 & \text{on } \Gamma_{side}, \\ \kappa(\mu_0)\nabla u(\boldsymbol{\mu}) \cdot \mathsf{n} = \mu_1 & \text{on } \Gamma_{base}. \end{cases}$$

$$\kappa(\mu_0) = \begin{cases} \mu_0 & \text{in } \Omega_1, \\ 1 & \text{in } \Omega_2, \end{cases}$$



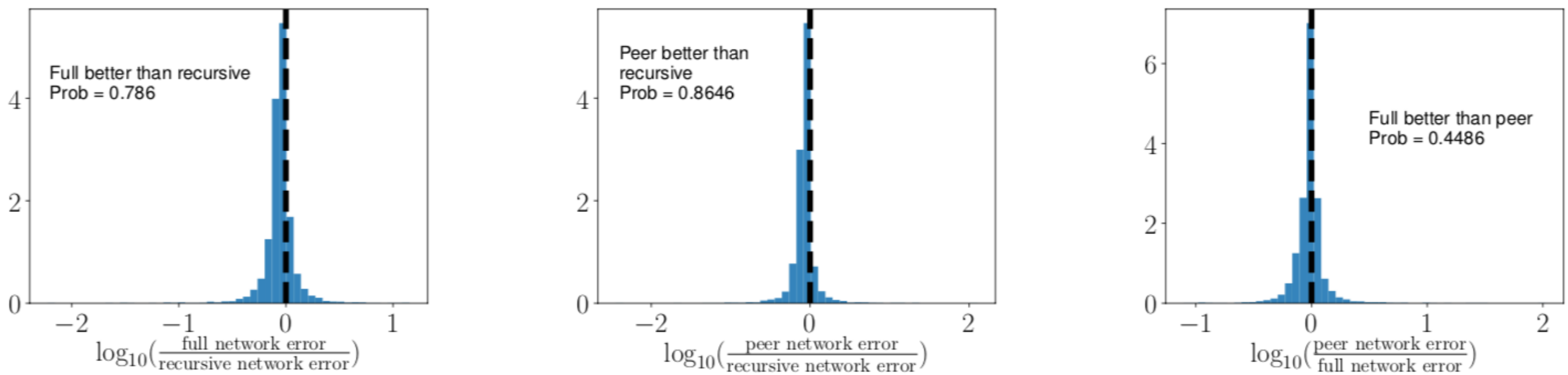
1. Three models: two finite element models with different mesh resolution, and reduced order model.
2. Constructed using RBniCS [Hesthaven, Rozza, Samm, 2015]
3. Predict temperature at (0.5, 0.8)

Thermal block results
Non-recursive solution is best over 85% realizations of training data



(a) Full (b) Peer (c) Hierarchical

Figure 9: Thermal block model network structures.



(a) Full vs. hierarchical/recursive. (b) Peer vs. hierarchical/recursive. (c) Peer vs. full.

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem
- Data can be used to update and inform the modeling structure only if this possibility is encoded.

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem
- Data can be used to update and inform the modeling structure only if this possibility is encoded.
- MFNets can increase accuracy compared to standard recursive approaches

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem
- Data can be used to update and inform the modeling structure only if this possibility is encoded.
- MFNets can increase accuracy compared to standard recursive approaches
- Challenges for Bayesian inference due to manifolds of equal probability
  - Investigating better priors to collapse the manifold
  - Investigating approximation quality of variational inference

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem
- Data can be used to update and inform the modeling structure only if this possibility is encoded.
- MFNets can increase accuracy compared to standard recursive approaches
- Challenges for Bayesian inference due to manifolds of equal probability
  - Investigating better priors to collapse the manifold
  - Investigating approximation quality of variational inference
- References

  - Gorodetsky, A. A., Jakeman, J. D., and Geraci, G. "MFNets: data efficient all-at-once learning of multifidelity surrogates as directed networks of information sources." *Computational Mechanics* 68:4 (2021): 741-758.
  - Gorodetsky, A. A., Jakeman, J.D., Geraci, G., Eldred, M.S., "MFNETS: multifidelity data-driven networks for Bayesian learning and prediction." *International Journal of Uncertainty Quantification* 10:6 (2020) : 595-622

Summary

- Determining how to efficiently fuse data from different simulations remains a *modeling* problem
- Data can be used to update and inform the modeling structure only if this possibility is encoded.
- MFNets can increase accuracy compared to standard recursive approaches
- Challenges for Bayesian inference due to manifolds of equal probability
  - Investigating better priors to collapse the manifold
  - Investigating approximation quality of variational inference
- References

  - Gorodetsky, A. A., Jakeman, J. D., and Geraci, G. "MFNets: data efficient all-at-once learning of multifidelity surrogates as directed networks of information sources." *Computational Mechanics* 68:4 (2021): 741-758.
  - Gorodetsky, A. A., Jakeman, J.D., Geraci, G., Eldred, M.S., "MFNETS: multifidelity data-driven networks for Bayesian learning and prediction." *International Journal of Uncertainty Quantification* 10:6 (2020) : 595-622

- Code: `http://github.com/goroda/MFNetsSurrogates`

Future directions

1. Develop automated ways to learn / average over graph structures themselves
2. Increasing alignment of input space to increase correlations via active basis methods
3. Variational version of the approach can be embedded in outer-loop applications
4. MCMC approach is both necessary and yet still quite costly, can this be done better?
5. Explore all the different ideas for $\mathcal{F}$ and discrepancy functions in the context of MFNets
6. Expand applications

Thanks!

Acknowledgements

This project was funded by the Laboratory Directed Research Development (LDRD) program at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.