SANDIA REPORT
SAND2022-12793
Printed September 2022

Sandia National Laboratories

Multi-fidelity information fusion and resource allocation

J.D. Jakeman, M.S. Eldred, G. Geraci, D.T. Seidl, T.M. Smith, A.A. Gorodetsky, T. Pham, A. Narayan, X. Zheng, R. Ghanem

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

This project created and demonstrated a framework for the efficient and accurate prediction of complex systems with only a limited amount of highly trusted data. These next generation computational multi-fidelity tools fuse multiple information sources of varying cost and accuracy to reduce the computational and experimental resources needed for designing and assessing complex multi-physics/scale/component systems. These tools have already been used to substantially improve the computational efficiency of simulation aided modeling activities from assessing thermal battery performance to predicting material deformation. This report summarizes the work carried out during a two year LDRD project. Specifically we present our technical accomplishments; project outputs such as publications, presentations and professional leadership activities; and the project's legacy.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The ability to make accurate and cost-effective predictions is essential for outer loop decision making processes that require the repeated interrogation of a simulation or experiment, for example uncertainty quantification and design of complex systems. Exclusive use of the highest-fidelity information source to make predictions is intractable when the cost of interrogating that source is high. Approximations of the high-fidelity source, such as reduced physics, reduced order, or response surface models, can be used to reduce the cost of outer-loop analyses [6, 78, 54, 24, 66, 31, 79, 37, 51, 39], however these approaches can introduce significant error when limited high-fidelity data is available. Moreover they are potentially inefficient because they do not leverage information available from other information sources.

To overcome these challenges, we developed novel multi-fidelity (MF) methods to make predictions with certifiable accuracy that integrate high-fidelity information alongside data from multiple cheaper sources.[1] Specifically, we developed two classes of methods. The first class of methods predict statistics of a high-fidelity information source using Monte Carlo (MC) type sampling approaches that leverage correlation between models to reduce the variance of the MC estimator [22, 56, 23]. The second class of methods construct surrogates of high-fidelity information sources using data from multiple models to improve accuracy away from the limited high-fidelity training data [76, 30, 34, 26].

At the start of this project, MF MC sampling methods suffered from the following deficiencies:

- An inability to leverage low-fidelity models that are parameterized differently from the high-fidelity model. This often prevented MF methods being applied to models with different physics when the number or type of parameters changes as more simplifying assumptions are made. For example, RANS simulations that have parameters governing a turbulence submodel could not be combined with potential flow models that do not share these parameters.

- Oracle statistics, e.g. correlation between models, are required to determine the amount of samples to allocate to each model. Approximate correlations were computed using a set of $O(10)$ pilot evaluations, however the cost and inaccuracy of this procedure was not optimally accounted for when allocating resources to each model fidelity.

At the start of this project MF surrogate methods suffered from the following deficiencies

- An inability to use ensembles of models that do not admit a hierarchy of information sources, ordered by their predictive capability. Multi-index methods that can use multi-dimensional hierarchies existed [30], but more general relationships between models

[1] This work builds on the 1 year LDRD [40].

could not be exploited. This limited the number of applications for which MF methods could be successful.

- An inability to exploit the structure of coupled multi-physics systems to allocate resources for generating training data from each system component in a manner that is commensurate with its impact on the accuracy of system predictions. single-fidelity strategies that used training data from a single-fidelity for each component existed [44, 16], but provided no mechanism to utilize multiple models of varying fidelity for each component.
- An inability to generate predictions by enriching a data-driven approximation, based upon experimental data, with data from multiple lower-fidelity simulations models. The method in [43] could combine experiments with one simulation model, however the use of multiple simulation models was not supported.

During the last two years our project was able to successfully address all these challenges. Our resulting methods have already been deployed to various applications across Sandia National Laboratories mission space and led to substantial follow-on funding to support continued development and deployment of MF methods. Our MF MC achievements include:

- A MF sampling method that identifies and exploits a lower-dimensional manifold shared by multiple models, with dissimilar parameterizations, to increase the correlation between the models and thus increase the accuracy of the MF statistics for a fixed budget [83, 85].
- Theoretical results quantifying the impact of inaccurate oracle statistics, determined from a finite pilot sample, on the accuracy of MF statistical estimators [61].
- A model tuning algorithm capable of identifying effective configurations (from a possible continuum of discretization parameters such as mesh resolution and solver tolerances) for low-fidelity approximations that can maximize the performance within a MF analysis [9, 11, 3, 4].
- A multi-arm bandit procedure for MF estimation of statistics without pilot samples that also unifies most existing MF MC methods [80].

Our MF surrogate achievements include:

- A framework that encodes prior knowledge about the non-hierarchical relationships between information sources to efficiently construct surrogates using data from all models [25, 60].
- An adaptive algorithm that can leverage ensembles of models of varying cost and accuracy, for one or more components of a multi-physics model, to reduce the computational cost of constructing the integrated-surrogate by up to two orders of magnitude [38].
- MF Gaussian processes that can be used to efficiently combine non-hierarchical models ensembles and fuse experimental data with multiple lower-fidelity simulation models [71].

2. TECHNICAL HIGHLIGHTS

This chapter summarizes the technical highlights of this project that have appeared or will appear in peer reviewed publications.

2.1. Multi-fidelity networks

MF methods reduce the amount of high-fidelity training data needed to build surrogates by combining data from an ensemble of models with varying cost and accuracy. Most existing MF methods can only use two models [46, 59, 58, 49, 64] or a hierarchy of models (Figure 2-1 left, red-outline) [42, 29, 76], which substantially limits efficiency gains. These methods approximate the discrepancy between the outputs of the consecutive pairs of models (e.g. Low B, Medium B, and High in Figure 2-1 left), which requires less computational resources than single-fidelity methods that use data solely from the high-fidelity model.

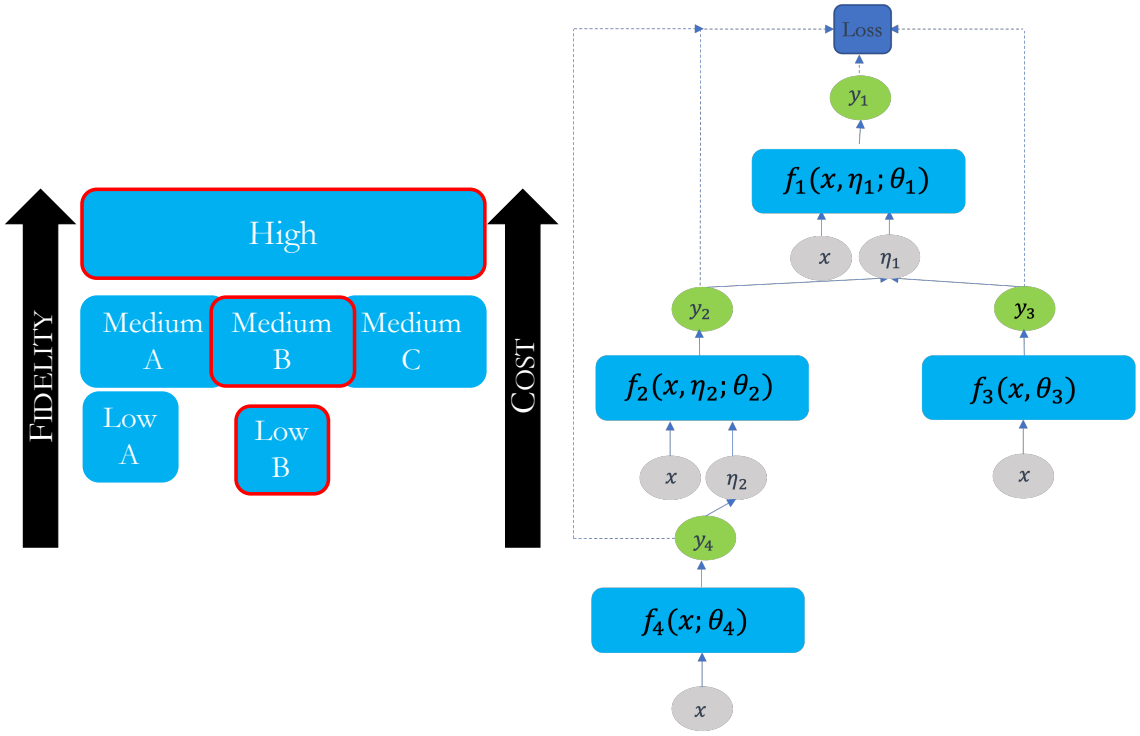


Figure 2-1. (Left) Conceptual representation of a MF model ensemble. (Right) Example of a graphical MFNets surrogate model.

This project developed a generalization of traditional bi-fidelity [59, 49] and hierarchical methods [28, 34, 45] we call MFNets. We used directed acyclic graphs (DAGs) to encode the

relationship between models (Figure 2-1 right). Each node represents a surrogate f_k expressed as a (possibly non-linear) function of a set of exogenous variables x and the combined outputs η_k of lower-fidelity models, e.g. $\eta_0 = [y_1^\top, y_2^\top]^\top$, and parameterized by a set of unknowns θ_k. Directed edges encode the dependencies between models and indicate the transfer of data between each node. For example, edges connect the two medium-fidelity models (f_2, f_3) in the right panel of Figure 2-1 to the high-fidelity model f_1, which can simulate all the physical regimes targeted by the lower-fidelity models but is too computationally expensive to evaluate extensively.

We used maximum likelihood estimation (MLE) to learn all the unknowns $\theta_G = [\theta_1^\top, \ldots, \theta_K^\top]^\top$ of the MF graph simultaneously, using data in the form of input-output pairs (x_k, y_k) from all models thereby allowing the low-fidelity model data to influence the high-fidelity model unknowns and vice versa. We also assumed that the observational noise is independent with variance σ_k^2 so that the negative log likelihood of the graph $L_G(\theta_G)$ can be efficiently computed by traversing the DAG and summing the negative log likelihood of each node.

Our results, presented in [25], demonstrate that, for the problems tested, all-at-once non-hierarchical information fusion based upon networks of polynomial surrogate nodes, significantly outperforms hierarchical MF strategies when the models available do not admit a hierarchy. In recent work, being added to a paper in preparation (Section 3.4), we move beyond using MLE and use Bayesian inference to learn the unknown parameters of the graph. Instead of learning a point estimate of the parameters using MLE, Bayesian inference provides a set of likely parameter values which we use to estimate error in our surrogates.

2.2. Multi-fidelity surrogate modeling of multi-disciplinary systems

Modeling complex systems often involves integrating numerous components from multiple disciplines which are computationally expensive to run. Consequently, uncertainty quantification (UQ) and design can be intractable when one or more component-models are computationally expensive to simulate. Once built, surrogate methods, e.g. [36, 72, 35, 53, 27, 75, 63, 87, 62], can be used to reduce the computational cost of UQ or design, but generating sufficient training data from coupled models can be difficult.

The efficacy of surrogate models of coupled systems can be improved by exploiting the structure of coupled systems [68, 44, 8, 5, 48, 12]. These methods express each of the P component-models of a system as a function of both exogenous inputs γ controlled by the user/modeler, e.g. random or design variables, and inputs ξ that we call coupling variables, whose values are determined by the outputs y of the other components (Figure 2-2). Surrogates of each component are then built and combined using relaxation methods to predict system-level quantities of interest (QoI). This approach can substantially reduce the computational cost of constructing surrogates of coupled system, but can still be intractable when using very expensive high-fidelity components.

This project developed a novel MF version of the above algorithms that exploits the coupling of multi-disciplinary/multi-physics models to reduce the cost of building machine learning surrogates by up to two orders of magnitude [38]. The error in our predictions of system-level QoI is greedily minimized using an active learning procedure that allocates the amount of training

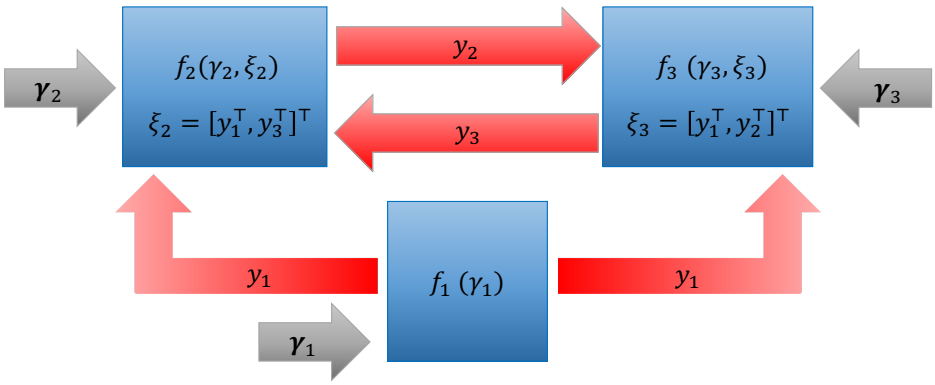


Figure 2-2. Conceptual depiction of a system comprised of coupled components.

data used to construct each component-surrogate, based on the contribution of those surrogates to the error of the integrated-surrogate. When a selection of simulators of varying fidelity and computational cost are available for one or more of the system components, our experimental design algorithm enriches a small number of high-fidelity simulations with larger numbers of simulations from models of lower accuracy and cost, to enable greater exploration and resolution of uncertainty while maintaining deterministic prediction accuracy.

2.3. Model tuning for multi-fidelity Monte Carlo

MC based MF methods combine predictions from multiple low-fidelity models with a high-fidelity model of interest to produce unbiased statistics of a desired accuracy at a fraction of the computational cost. The accuracy and computational cost of these MF MC-based statistics depend on the correlations between all available models and their relative computational cost. Consequently, when creating a model ensemble by using models with different numerical discretizations, such as mesh and time-step size and solver tolerances, it is unclear what discretizations should be used *a priori*. This project has developed *model tuning* algorithms that can be used to select the model discretizations that increase the accuracy of MF statistics for a fixed computational budget. Our results documented in [9] indicate that low-fidelity model tuning can significantly improve efficiency and precision of trajectory simulations and provide an increased edge to MF MC-based methods when compared to standard MC. Further studies of model tuning are also documented in [11].

The model tuning capabilities developed in this project were successfully used to support an ASC L2 milestone [3, 4]. Relative to a hand-tuning approach that selected temporal discretizations based on ad hoc metrics and subject-matter expertise, our optimization-based model tuning approach reduced the cost of quantifying uncertainty in predictions of thermal battery performance by up to an additional factor of six for the top-performing MF estimators, resulting in a total factor of up to 143 relative to the single-fidelity MC reference. As noted in [3], under the assumption of consistent scaling, a notional five month computational campaign could be compressed into a single day. Moreover, optimization-based tuning can save considerable human

time by automating what can be a laborious process for configuring approximations to achieve effective correlation at reduced cost.

2.4. Ensemble approximate control variates

Existing MF sampling methods require a set of evaluations of each model in an ensemble (called a pilot sample) to compute correlations between models. However, in [61] we demonstrated that using correlations based upon pilot samples can lead to inaccurate MF estimates of statistics because these methods assume the estimated correlation is highly accurate. Consequently, we advocated for using an ensemble of MF approximate control variate (ACV) estimators computed from a subset of all available model evaluations. In many situations, codified by our theory, ensemble based estimators produce a MF estimate of expectation that is more accurate than an equivalent cost estimator that uses pilot samples.

In [61], we also demonstrated the utility of using ACV to improve the efficiency of importance-sampling based estimates of rare events. While existing works have used low-fidelity models to choose prudent biasing distributions that drastically reduce the computation cost of importance sampling, e.g. [55], these methods did not leverage the low-fidelity models to reduce the variance (error) of the failure probability estimate. By using the low fidelity model to provide a biasing density and reduce variance using ACV estimation, we were able to further increase the efficiency of existing methods, resulting in orders of magnitude reduction in computational cost relative to a single-fidelity MC estimate.

2.5. PyApprox software

PyApprox [33] is a Python software package for probabilistic analysis of scientific numerical models. Easy to use and extendable tools are provided for constructing surrogates, sensitivity analysis, Bayesian inference, experimental design, and forward uncertainty quantification. This project implemented our novel advances and numerous existing MF surrogate and MC methods in PyApprox which use multiple model discretizations and/or simplified physics to significantly reduce the computational cost of various types of analyses. Existing sampling methods implemented include: multi-level MC [22], MF MC [56], and ACV [23, 10]. Existing surrogate methods implemented include: multi-level Gaussian processes [42], multi-level [76] and multi-index [30] sparse grids, and MF reduced order models [50]. This comprehensive suite of capabilities can be used to compare newly developed algorithms and the use of the best method for a given application.

3. PAPERS IN PREPARATION

This chapter presents the technical advances made by this project that are yet to be published.

3.1. Dissimilar parameterizations

MF sampling methods exploit correlation between an ensemble of models to reduce the error in estimates of the high-fidelity model statistics. Unfortunately correlation significantly decreases when models have differing number of parameters and/or parameters that influence system behavior differently. We refer to such models as having dissimilar parameterizations. As an example, consider a high-fidelity Large Eddy Simulation (LES) model and a lower-fidelity Reynolds-averaged Navier Stokes (RANS) model. Some of the uncertain parameters of the LES model do not have a counterpart in the RANS model and vice versa. A practitioner could apply MF sampling strategies to the subset of the uncertain parameters shared by all models, however in realistic cases, non-shared parameters can have a non-negligible effect on a model's response, contribute signiticantly to its variance, and decrease the correlation among models.

In preliminary work proceeding this LDRD we identified application specific means to address the challenges posed by dissimilar parameterizations [17, 18, 20]. Specifically we identified a low-dimensional manifold that explained the majority of the models' variance. By constructing this manifold we minimized the effect of parameters off the manifold that are not shared among models. We demonstrated the promise of using active subspaces to identify this shared manifold in [13, 14]. In collaboration with Xiaoshu Zeng and Roger Ghanem from University of Southern California, this project generalized and extended our previous work by using the so-called Adaptive Basis (AB) method [19, 82, 84] to identify low-dimensional manifolds. The novel contributions of this work are:

- Use of the adaptive basis (AB) method [77, 86] to improve the efficiency of MF sampling for UQ;

- The formulation of statistical metrics to practically quantify the impact of the manifold dimension on the performance of the MF estimator using a small number of pilot samples.

In the following sections we briefly introduce the AB method and discuss its use for construcing a MF sampling estimator. While we focus on the use of AB for MF sampling, these approaches can also be used to construct surrogates, however this task is left for future research.

3.1.0.1. The Adaptive Basis Method

The Adaptive Basis (AB) method was originally developed for dimension reduction to improve the efficiency of UQ [77, 86]. Here, we use it to identify a low-dimensional manifold that can be used MF sampling of dissimilar models. The AB method uses Polynomial Chaos Expansions (PCEs) [21], which given a set ofvariables $\boldsymbol{\xi}$ and the Hilbert space $\mathcal{H} \subset L^2(\Omega, \mathcal{F}(\mathcal{H}), \mathcal{P})$, approximates a model Q as

$$Q(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \mathcal{J}} Q_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \approx \sum_{\boldsymbol{\alpha} \in \mathcal{J}_p} Q_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}),$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d) \in \mathcal{J} := (\mathbb{N}_0)^d$ is a d-dimensional multi-index, $\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) : \boldsymbol{\alpha} \in \mathcal{J}\}$ are orthonormal polynomials, which are a complete basis for $\mathcal{H}$, and $\mathcal{J}_p = \{\boldsymbol{\alpha} \in \mathcal{J} : |\boldsymbol{\alpha}| \leq p\}$ and $|\boldsymbol{\alpha}| = \sum_{i=1}^{d} \alpha_i$. The PCE coefficients satisfy

$$Q_{\boldsymbol{\alpha}} = \frac{\langle Q, \psi_{\boldsymbol{\alpha}} \rangle}{\langle \psi_{\boldsymbol{\alpha}}^2 \rangle} = \langle Q, \psi_{\boldsymbol{\alpha}} \rangle, \quad \boldsymbol{\alpha} \in \mathcal{J}_p,$$

where $\langle \cdot \rangle$ is the L^2 inner product. For simplicity we will present the case of Gaussian variables, but both the PCE and the AB method (with its integration in the MF estimator) can be extended to the case of generic distributions.

The AB method identifies the model's *important directions* by rotating the Gaussian inputs of a model to form a new basis such that the model can be efficiently approximated by a low-dimensional subspace spanned by the first rotated dimensions. The rotation matrix defines the so-called adapted variables $\boldsymbol{\eta}$ as

$$\boldsymbol{\eta} = \boldsymbol{\eta}(\boldsymbol{\xi}) = \boldsymbol{A}\boldsymbol{\xi}$$

Because $\boldsymbol{\xi}$ are independent Gaussian variables and the rotation is a linear transformation, the PCE over the adapted variables can be written as

$$Q^{\boldsymbol{A}}(\boldsymbol{\eta}) = \sum_{\boldsymbol{\beta} \in \mathcal{J}_p} Q^{\boldsymbol{A}}_{\boldsymbol{\beta}} \psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}),$$

where the superscript $\boldsymbol{A}$ on Q and $Q_{\boldsymbol{\beta}}$ denotes that the expansion is in terms of new variables generated by the rotation matrix $\boldsymbol{A}$.

Since $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are both sets of independent Gaussian variables, $\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) : \boldsymbol{\alpha} \in \mathcal{J}_p\}$ and $\{\psi_{\boldsymbol{\beta}}(\boldsymbol{\eta}) : \boldsymbol{\beta} \in \mathcal{J}_p\}$ span the same space, the expansions $Q^{\boldsymbol{A}}(\boldsymbol{\eta}(\boldsymbol{\xi}))$ and $Q(\boldsymbol{\xi})$ are equivalent. Moreover, the new PCE coefficients can be obtained directly by projection as

$$Q^{\boldsymbol{A}}_{\boldsymbol{\beta}} = \sum_{\boldsymbol{\alpha} \in \mathcal{J}_p} Q_{\boldsymbol{\alpha}} \langle \psi_{\boldsymbol{\alpha}}, \psi^{\boldsymbol{A}}_{\boldsymbol{\beta}} \rangle, \quad \boldsymbol{\beta} \in \mathcal{J}_p,$$

which provides a simple way to transform the PCE from one space to another. Algorithm 1 summarizes the method used to compute the rotation matrix A. In this algorithm $\boldsymbol{e}_i$ is a unit vector of all zeros except the ith entry that has a value of 1.

Algorithm 1: Construction of rotation matrix by Gaussian adaptation

1 Use least squares regression to estimate the coefficients Q_0 and $\{Q_{\boldsymbol{e}_i}\}_{i=1}^d$ of the first-order PCE

$$Q(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\mathcal{J}_1} Q_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = Q_0 + \sum_{i=1}^{d} Q_{\boldsymbol{e}_i}\xi_i\,,$$

2 Construct the first row of the rotation matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ such that

$$\eta_1 = \sum_{i=1}^{d} A_{1i}\xi_i = \sum_{i=1}^{d} Q_{\boldsymbol{e}_i}\xi_i \to A_{1i} = Q_{\boldsymbol{e}_i} \quad \text{for} \quad i = 1,\ldots,d.$$

3 Rank the first-order coefficients $\{Q_{\boldsymbol{e}_i}\}_{i=1}^d$ by absolute value in descending order and record their indices in the original coordinates as $\{\kappa_j\}_{j=1}^d$;

4 For $j = 2,\ldots,d$ construct the j^{th} row of $\boldsymbol{A}$ such that

$$\eta_j = \xi_{\kappa_{j-1}}.$$

5 Perform Gram-Schmidt procedure on $\boldsymbol{A}$ to make it an isometry (rotation matrix).

Once the rotation matrix $\boldsymbol{A}$ has been obtained, dimension reduction can be performed by partitioning the rotated variables $\boldsymbol{\eta}$ as

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{A}_r\boldsymbol{\xi} \\ \boldsymbol{A}_{\neg r}\boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_{\neg r} \end{bmatrix},$$

where $\boldsymbol{\eta}_r$ are defined as the first *r important directions* and the corresponding PCE $Q^{\boldsymbol{A}_r}$ expansion on these r important directions can be written as

$$Q^{\boldsymbol{A}_r}(\boldsymbol{\eta}_r) = \sum_{\boldsymbol{\gamma}\in\mathcal{J}_p^r} Q_{\boldsymbol{\gamma}}^{\boldsymbol{A}_r}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\eta}_r) \approx Q(\boldsymbol{\xi}).$$

Here, $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_r)$ is the r-dimensional multi-index and $\{\psi_{\boldsymbol{\gamma}}(\boldsymbol{\eta}_r) : \boldsymbol{\gamma}\in\mathcal{J}_p^r\}$ is the basis of order up to p defined on $\boldsymbol{\eta}_r$. It is important to note that, the estimation of the PCE coefficients requires evaluating the model in the original coordinates $\boldsymbol{\xi}$, which need to be rotated from $\boldsymbol{\eta}_r$. Here, we use the approximate inverse transformation

$$\boldsymbol{\xi}^r = \boldsymbol{\xi}^r(\boldsymbol{\eta}_r) = \boldsymbol{A}^T \begin{Bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{0} \end{Bmatrix}$$

to map the r-dimensional rotated coordinates $\boldsymbol{\eta}_r \in \mathbb{R}^r$ back to the original coordiates.

3.1.0.2. Embedding AB within MF UQ

In this section we formulate a new MF estimator that exploits the AB method. We adopt the notation consistent with [23] and so that an underscore, e.g. $\underline{\boldsymbol{\xi}}$, denotes a set of random

realizations of the corresponding variable, e.g. $\boldsymbol{\xi}$. Using this notation, our MFAB estimator is a control variate estimator [52, 56, 57] given by

$$\hat{Q}^{MF}(\alpha, \underline{\boldsymbol{\xi}}_H, \underline{\boldsymbol{\xi}}_L^1, \underline{\boldsymbol{\xi}}_L^2) = \hat{Q}_H(\underline{\boldsymbol{\xi}}_H) + \alpha\left(\hat{Q}_L(\underline{\boldsymbol{\xi}}_L^1) - \hat{\mu}_L(\underline{\boldsymbol{\xi}}_L^2)\right),$$

where

$$\begin{aligned}\boldsymbol{\xi}_H &= \boldsymbol{A}_H^{\mathrm{T}}\boldsymbol{\eta}_H\\ \boldsymbol{\xi}_L^1 &= \boldsymbol{A}_L^{\mathrm{T}}\boldsymbol{\eta}_L^1\\ \boldsymbol{\xi}_L^2 &= \boldsymbol{A}_L^{\mathrm{T}}\boldsymbol{\eta}_L^2,\end{aligned}$$

or equivalently

$$\hat{Q}^{MFAB}(\alpha, \underline{\boldsymbol{\eta}}_H, \underline{\boldsymbol{\eta}}_L^1, \underline{\boldsymbol{\eta}}_L^2) = \hat{Q}_H(\boldsymbol{A}_H^{\mathrm{T}}\underline{\boldsymbol{\eta}}_H) + \alpha\left(\hat{Q}_L(\boldsymbol{A}_L^{\mathrm{T}}\underline{\boldsymbol{\eta}}_L^1) - \hat{\mu}_L(\boldsymbol{A}_L^{\mathrm{T}}\underline{\boldsymbol{\eta}}_L^2)\right). \tag{3.1}$$

Assuming that the high- and low-fidelity rotated coordinates $\boldsymbol{\eta}_H$ and $\boldsymbol{\eta}_L$ are arranged in decreasing order of importance (see Algorithm 1), we can define a vector of shared *important directions*

$$\boldsymbol{\eta}_s = [\eta_{s,1}, \ldots, \eta_{s,r_s}]^{\mathrm{T}} \in \mathbb{R}^{r_s},$$

where $r_s = \max(r_H, r_L)$ and

$$\eta_{s,i} = \begin{cases} \eta_{H,i} = \eta_{L,i} & \text{if} \quad i \leq \min(r_H, r_L) \\ \eta_{H,i} & \text{if} \quad r_L < i \leq r_H \\ \eta_{L,i} & \text{if} \quad r_H < i \leq r_L. \end{cases}$$

Consequently, our inverse transformation which approximately recovers the original coordinates is given by

$$\begin{aligned}\boldsymbol{\xi}_H^{r_H} &= \boldsymbol{A}_{H,r_H}^{\mathrm{T}}\boldsymbol{\eta}_{H,r_H} = \begin{bmatrix}\boldsymbol{A}_{H,r_H}^{\mathrm{T}} & \boldsymbol{0}_{d_H\times(r_s-r_H)}\end{bmatrix}\boldsymbol{\eta}_s = \mathcal{A}_{H,r_H}^{\mathrm{T}}\boldsymbol{\eta}_s\\ \boldsymbol{\xi}_L^{r_L} &= \boldsymbol{A}_{L,r_L}^{\mathrm{T}}\boldsymbol{\eta}_{L,r_L} = \begin{bmatrix}\boldsymbol{A}_{L,r_L}^{\mathrm{T}} & \boldsymbol{0}_{d_L\times(r_s-r_L)}\end{bmatrix}\boldsymbol{\eta}_s = \mathcal{A}_{L,r_L}^{\mathrm{T}}\boldsymbol{\eta}_s,\end{aligned}$$

Using this transformation in (3.1) yields

$$\hat{Q}^{MFAB}\left(\underline{\boldsymbol{\eta}}_s, \underline{\boldsymbol{\eta}}_s^2; \alpha, r_H, r_L\right) = \hat{Q}_H^{r_H}(r_H, \underline{\boldsymbol{\eta}}_s) + \alpha\left(\hat{Q}_L^{r_L}(r_L, \underline{\boldsymbol{\eta}}_s) - \hat{\mu}_L^{r_L}(r_L, \underline{\boldsymbol{\eta}}_s^2)\right).$$

As written above, the MFAB estimator allows for sampling the *shared space* only (spanned by $\boldsymbol{\eta}_s$), but it introduces a different rotation on each model, which leads to linking the original coordinates as

$$\boldsymbol{\xi}_H^{r_H} = \boldsymbol{A}_{H,r_H}^{\mathrm{T}}\boldsymbol{A}_{L,r_L}\boldsymbol{\xi}_L^{r_L}.$$

The MSE of the MFAB estimator can be written as

$$\mathrm{MSE}\left[\hat{Q}^{MFAB}\right] = \mathbb{V}ar\left[\hat{Q}^{MFAB}\right] + \left(\mathbb{E}\left[Q_H\left(\mathcal{A}_{H,r_H}^{\mathrm{T}}\boldsymbol{\eta}_s^{(i)}\right)\right] - \mathbb{E}\left[Q_H(\boldsymbol{\xi}_H)\right]\right)^2 = \sigma^2(r_H, r_L) + \delta^2(r_H),$$

where

$$\sigma^2(r_H, r_L) = \mathbb{V}ar\left[\hat{Q}^{MFAB}\right] = \mathbb{V}ar\left[\hat{Q}_H^{r_H}(r_H, \underline{\boldsymbol{\eta}}_s)\right]\left(1 - \frac{\upsilon - 1}{\upsilon}\rho^2_{r_L, r_H}\right),$$

$\rho^2_{r_L,r_H}$ is the correlation between the models sampled at the rotated coordinates $Q_L(\mathcal{A}^{\mathrm{T}}_{L,r_L}\underline{\boldsymbol{\eta}}_s)$ and $Q_H(\mathcal{A}^{\mathrm{T}}_{H,r_H}\underline{\boldsymbol{\eta}}_s)$, *i.e.*

$$\rho^2_{r_L,r_H} = \frac{\mathrm{Cov}^2\left[Q_H\left(\mathcal{A}^{\mathrm{T}}_{H,r_H}\underline{\boldsymbol{\eta}}_s\right), Q_L\left(\mathcal{A}^{\mathrm{T}}_{L,r_L}\underline{\boldsymbol{\eta}}_s\right)\right]}{\mathbb{V}ar\left[Q_H\left(\mathcal{A}^{\mathrm{T}}_{H,r_H}\underline{\boldsymbol{\eta}}_s\right)\right]\mathbb{V}ar\left[\mathcal{A}^{\mathrm{T}}_{L,r_L}\underline{\boldsymbol{\eta}}_s\right]},$$

and the estimator squared bias is

$$\delta^2(r_H) = \left(\mathbb{E}\left[Q_H\left(\mathcal{A}^{\mathrm{T}}_{H,r_H}\boldsymbol{\eta}_s^{(i)}\right)\right] - \mathbb{E}\left[Q_H(\boldsymbol{\xi}_H)\right]\right)^2. \tag{3.2}$$

The efficiency of the MFAB estimator is dependent on the number of coordinates of the models, r_H and r_L. Specifically, the bias of the estimator decreases as $r_H \to d_H$ (where d_H indicates the number of original parameters), while the correlation between the high- and low-fidelity representations (on the shared space) decreases when the number of important directions r_H of the the high-fidelity model is made too large. In the following we present a strategy for balancing these effects.

Estimation of the HF MSE from pilot samples. Let's assume that N_p pilot samples of the high-fidelity model are available, *i.e.* $\{Q_i\}_{i=1}^{N_p}$. We first use Algorithm 1 to construct the rotation matrix $\boldsymbol{A}$ and then, without additional evaluations, we rotate the original variables $\boldsymbol{\xi}_H$ onto the reduced space spanned by $\boldsymbol{\eta}_{H,r_H} = \boldsymbol{A}_{H,r_H}\boldsymbol{\xi}_H$. Next we use ordinary least squares to compute the coefficients $\tilde{Q}$ of the PCE

$$\tilde{Q}(\boldsymbol{\eta}_{H,r_H}) = \sum_{\boldsymbol{\gamma}\in\mathcal{J}_p^r} \tilde{Q}_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\eta}_{H,r_H}).$$

and compute the MSE for the PCE surfaces via

$$\mathrm{MSE}_{PCE} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(Q_i - \tilde{Q}_i\right)^2. \tag{3.3}$$

While this measure does not provide an absolute value for the squared bias of the high-fidelity model, it can provide an indication of the convergence of the AB representation with the number of important directions.

Estimation of the correlation. We also use the PCE constructed from pilot samples, for both the high- and low-fidelity models, to compute the correlation between models for all possible combinations of r_H and r_L. Denoting the PCE representations of the high- and low-fidelity model as

$$\tilde{Q}_H(\boldsymbol{\eta}_{H,r_H}) = \sum_{\gamma_H\in\mathcal{J}_{p_H}^{r_H}} \tilde{Q}_{\gamma_H}\Psi_{\gamma_H}(\boldsymbol{\eta}_{H,r_H})$$

$$\tilde{Q}_L(\boldsymbol{\eta}_{L,r_L}) = \sum_{\gamma_L\in\mathcal{J}_{p_L}^{r_L}} \tilde{Q}_{\gamma_L}\Psi_{\gamma_L}(\boldsymbol{\eta}_{L,r_L}),$$

the correlation in the rotated coordiantes can be estimated as

$$\rho\left(\tilde{Q}_H(\boldsymbol{\eta}_{H,r_H}), \tilde{Q}_L(\boldsymbol{\eta}_{L,r_L})\right) = \frac{\sum_{\gamma_H,\gamma_L \in \mathcal{J}_{p_H}^{r_H}, \mathcal{J}_{p_L}^{r_L}} \tilde{Q}_{\gamma_H} \tilde{Q}_{\gamma_L}}{\sqrt{\left(\sum_{\gamma_H \in \mathcal{J}_{p_H}^{r_H}} \tilde{Q}_{\gamma_H}^2\right)\left(\sum_{\gamma_L \in \mathcal{J}_{p_L}^{r_L}} \tilde{Q}_{\gamma_L}^2\right)}}. \tag{3.4}$$

This expression simply shows that the numerator of the correlation is only impacted by the shared basis terms in the rotated space, while the denominator is impacted by the entire expansion for both the high- and low-fidelity model.

Practical implementation. As illustrated above, there is a trade-off between the correlation among models, which can be maximized by reducing the number of important directions, and the MFAB estimator bias that, in constrast, requires a higher number of important directions to represent the high-fidelity model. We propose the following steps to balance these tradeoffs:

1. Use N_p pilot samples with Algorithm 1 to compute the rotation matrices for both the high- and low-fidelity model;
2. Determine the truncation dimension for the high-fidelity model, r_H, by comparing the errors in (3.3) for $1 \leq r_H \leq d_H$ (this step does not require additional model evaluations);
3. Sample on $\boldsymbol{\eta}_{H,r_H}$ to estimate the effective value for the MSE. This step does require new high-fidelity evaluations, but since r_H is usually much smaller than d_h, the number of samples is usually limited;
4. Set the target variance for the estimator to be commensurate with the squared bias obtained at the previous step using (3.2), *i.e.* $\delta^2 = \sigma^2$;
5. Use a grid search to evaluate Eq. 3.4 to determine the truncation of the low-fidelity model that maximizes its correlation with the high-fidelity model;
6. Optimize the sample allocation (for both models) using the algorithm in [23].

3.1.0.3. Numerical results

Consider the following analytical problem which we will use to illustrate the efficacy of our MFAB estimator

$$\begin{aligned} f(\boldsymbol{x}) =& \exp(x_1 + 0.05x_2) + \exp(0.8x_3) + \exp(0.8x_4 + 0.05x_5 + 0.05x_6) \\ &+ \exp(0.8x_7 + 0.05x_8) + \exp(0.08x_9 + 0.05x_{10}), \\ g(\boldsymbol{y}) =& \exp(0.1y_1 + y_2) + \exp(0.1y_3 + 0.01y_4 + 0.8y_5) + \exp(0.8y_6 + 0.1y_7) + \exp(0.8y_8). \end{aligned} \tag{3.5}$$

Here $f(\boldsymbol{x})$ denotes the high-fidelity model and $g(\boldsymbol{y})$ the low-fidelity model, with computational costs respectively given by $w_H = 1$ and $w_L = 0.01w_H$. These models are parameterized by 10 and 8 independent and identically distributed Gaussian variables $\mathcal{N}(0, 1)$, respectively. The scatter plot in the left panel of Figure 3-1 compares 1000 randomly sampled evaluations of f and g and shows that the correlation between the models in the original coordinates is extremely low, the

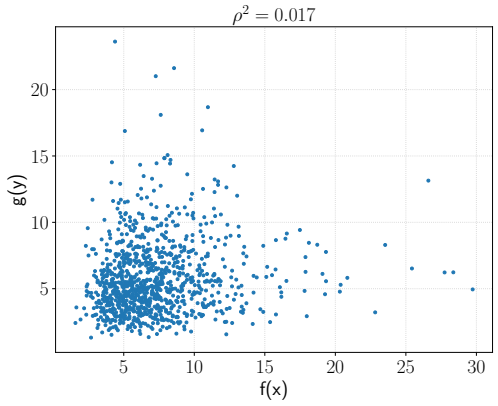


Figure 3-1. Scatter plot for f and g in the original coordinates in Eq. (3.5). The samples corresponding to the same input variables are shared, while the others are left free to vary according to their distribution.

correlation square is approximately 0.017. Such low correlation renders existing MF sampling methods ineffective.

In the following we use $N_p = 40$ pilot samples of both the high- and low-fidelity (which have been obtained in the original coordinates) to build the rotation matrix $\boldsymbol{A}$; Figure 3-2 plots the mean MSE measure introduced in (3.3) using the rotation matrix computed. Next we estimate the

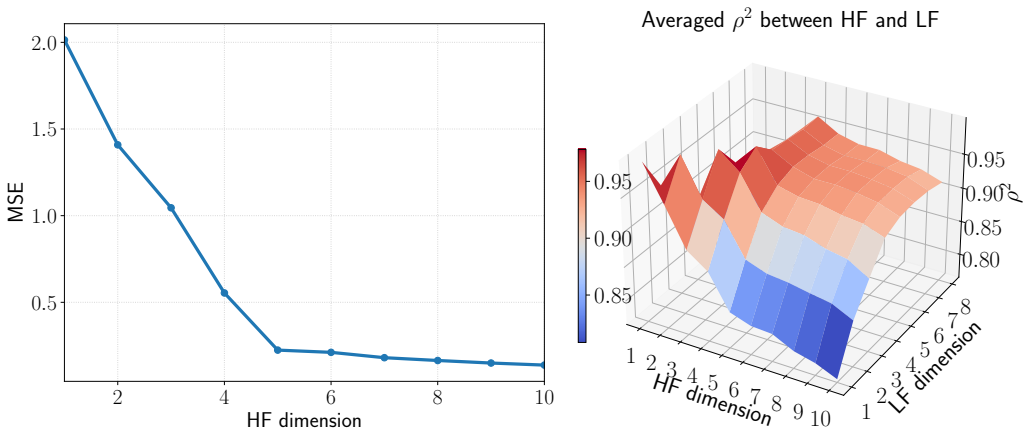


Figure 3-2. MSE measure (3.3) as function of the high-fidelity truncation dimension r_H (left) and correlation estimation via (3.4) as function of both the high- and low-fidelity truncation r_H and r_L (right) for the exponential test problem Eq. (3.5).

actual value for the MSE, by evaluating a number N_{r_H} of additional high-fidelity samples, incurring an additional computational cost, which must be included in the cost of constructing the MFAB estimator. We increase the of additional samples used proportionally to r_H. Specifically, we use: $N_{r_H} = 10$ for $r_H = 1$, $N_{r_H} = 20$ for $r_H = 3$, and $N_{r_H} = 30$ for $r_H = 5$, which respectively corresponds to $\delta^2 = 3.1, 0.5$, and 0.04. For each assigned r_H, we select the low-fidelity truncation r_L that corresponds to the maximum correlation (Figure 3-2 right) and solve the MF sample allocation problem for $\sigma^2 = \delta^2$.

In Figure 3-3, we show the probability density functions of the values of three estimators obtained

using 500 trials using different realizations of the pilot samples. Specifically, we compare the single-fidelity MC estimator that uses only the high-fidelity model, a traditional MF estimator that samples the models in the original coordinates and our MFAB estimator. The correlation of the MFAB estimator decreases when we increase the truncation dimension and the estimator bias decreases. Consistent with the results reported in Figure 3-2, setting $r_H = 5$ produces a very small bias and setting $r_H = 6$ practically eliminates the bias. For either of these high-fidelity truncation dimensions, the estimator variance can be made equal to the estimator bias by setting the low-fidelity truncation to $r_L = 5$.

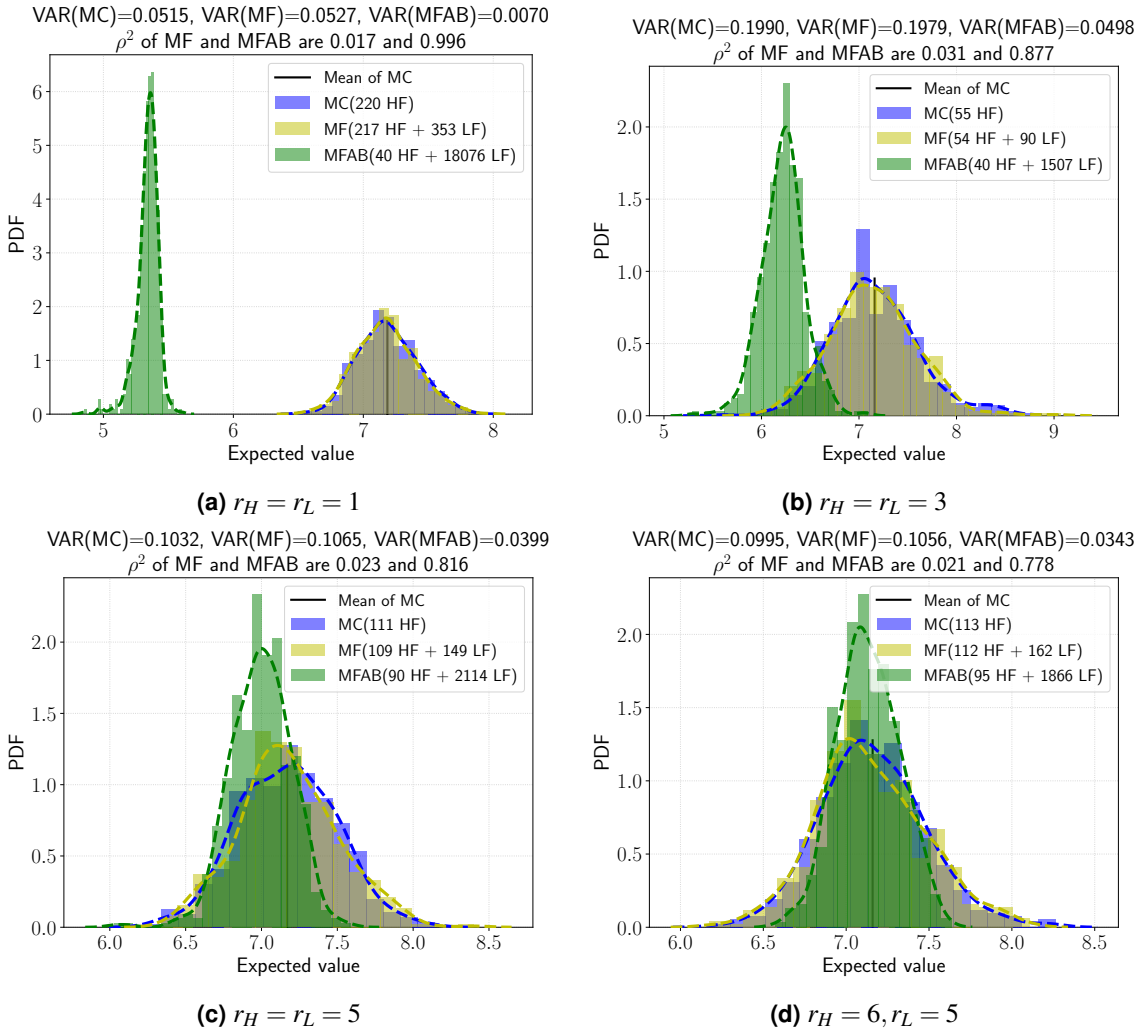
For this problem, our MFAB estimator reduces the variance of a single-fidelity MC estimator by approximately a 70%. Such a reduction cannot be obtained using the traditional MF MC method due to the lack of correlation between the models evaluated in their original coordinates. The MF estimator obtains a similar accuracy to the single-fidelity estimator because the optimization procedure used to allocate samples to each model determines that the low-fidelity model is not useful and assigns the vast majority of evaluations to the high-fidelity model. This result demonstrates the utility of using AB within MF estimation to enable considerable computational savings that cannot be obtained with existing methods.

3.2. Multi-arm bandits

ACV MC methods require an estimate of the correlation between models, to determine the amount of samples to allocate to each model to efficiently estimate statistics. The required correlation matrix is typically computed using a set of $O(10)$ pilot evaluations. However, the error in the approximation correlation matrix is typically not accounted for and the computational cost of computing the pilot sample is not optimized relative to the cost of the subsequent MF analysis. In collaboration with Akil Narayan, Yiming Xu, and Alex Gorodetsky, we developed an multi-arm bandit (MAB)-based approximate control variate method that does not require *a priori* estimates of correlation.

MAB learning has become an attractive framework for studying the exploration-exploitation trade-offs of making decisions in uncertain environments, e.g. determining how much budget they should allocate to running each version of an advertisement when the revenue each version will attract is initially unknown [70]. Given a slot machine with A arms, each with an unknown reward distribution, classical MAB algorithms attempt to decide which arm to pull next given the outcomes of the previous rounds to maximize the total expected reward. Effective algorithms balance the trade-off between exploration and exploitation, e.g. whether to pull more arms that have not been played much to improve estimates of rewards (exploration) or to pull the arms that have returned the highest rewards so far (exploitation).

In [81], our collaborators demonstrated the promise of using multi-arm bandit optimization to balance the cost of estimating the correlation between models with the cost of minimizing the error in MF estimates of statistics. Their work was enabled by realizing that the arms of the MAB problem can be used to represent subsets of models, pulling an arm represents evaluating all models in a subset, and reward represents improvement in the MSE of the MF estimator. This insight was then used to formulate an adaptive explore-then-commit algorithm (AETC) that



(a) $r_H = r_L = 1$ **(b)** $r_H = r_L = 3$

(c) $r_H = r_L = 5$ **(d)** $r_H = 6, r_L = 5$

Figure 3-3. Probability density functions for 500 realizations of the MC estimator and the two MF estimator based on the original coordinates (MF) or on the AB embedded method (MFAB) for the expected value estimation. The vertical solid line reports the mean value of the MC estimator.

automatically explores (computes correlation and selects the best subsets of models) and then commits (estimates the MF statistics using the remaining budget).

The preliminary work in [81] used a linear regression MC estimator (LRMC) that assumes a linear relationship between the high- and low-fidelity models. In this project we developed theory that shows that LRMCs are equivalent to a class of ACV estimators that do not require linearity and used this equivalence to improve the efficiency of the MAB-based LRMC MF estimators developed in [81]. Moreover we also show this algorithm works when the computational costs of running each model is not known a priori and/or is stochastic, e.g. the run time depends on the random parameters of varies due to load balancing on high performance computing machines. In the following we present a summary of our algorithm and some initial numerical results. A full description of our algorithm and theoretical advances will be made available in [80].

3.2.1. Multi-fidelity Best Linear Unbiased Estimator (BLUE)

Given a scalar valued model $X_0 = f_0(Z)$ parameterized by a set of random variables $Z = (Z_1, \ldots, Z_P)$ with probability measures ρ_Z our goal is to predict the expected value of the model output

$$\mu_0 = \mathbb{E}[X_0]$$

using MC quadrature. Unfortunately, when evaluating the model $X_0^{(i)} = f_0(Z^{(i)})$, with $Z^{(i)} \sim \rho_Z$, is computationally expensive, constructing the MC estimator

$$\hat{\mu}_0^{\mathrm{MC}} = \frac{1}{N}\sum_{n=1}^{N} X_0^{(i)} \approx \mu_0$$

accurately using large number of samples N is intractable. In this work we reduce the computational cost of constructing an unbiased estimate of μ_0 using n lower fidelity models $f_n(Z)$, $n \in [N] = \{1, \ldots, N\}$, of varying reduced cost and accuracy. Specifically, we estimate the expected value using the so called MF Best Linear Unbiased Estimator (BLUE).

Given a specified collection of subsets $S \in \{[N] \cup \{0\}\}$, e.g. $S = \{\{0,1\},\{1,2\},\{2,3\}\}$ that results in MLMC like partitioning of samples among 3 models, for each subset $T \in S$ BLUE generates M_T evaluations of each model $t \in T$ at the same set of input samples $Z_T = \{Z^{(m)}\}_{m=1}^{M_T}$ and collects them in a matrix

$$W_T = [X_T^{(1)}, \ldots, X_T^{(M_T)}] \quad \text{with} \quad X_T^{(m)} = [f_{t_1}^{(m)}, \ldots, f_{t_{|T|}}^{(m)}]^\top,$$

where $t_i, i = 1, \ldots |T|$ are the elements of the set T, the samples Z^K and Z^L are statistically independent if the sets $K \neq L$, and $\mathsf{C}_n^{(m)}$ denotes the computational cost of generating the model evaluation $X_n^{(m)}$. Then, given a restriction matrix $R_T \in \mathbb{R}^{|T| \times N+1}$ that satisfies

$$R_T v = [v_t]_{t \in T} \quad \forall v \in \mathbb{R}^{N+1}, \qquad\qquad P_T = R_T^\top,$$

the model data is assembled into a linear system of the form

$$W_S = H_S \hat{\mu}_S + \varepsilon_S$$

where

$$\begin{aligned} W_S &= [W_T^\top]_{T \in S} & H_S &= [H_T^\top]_{T \in S} & \varepsilon_S &= [\varepsilon_T^\top]_{T \in S} \\ W_T &= [W_T^{(m)}]_{m=1}^{M_T} & H_T &= [R_T]_{m=1}^{M_T} & \varepsilon_T &= [W_T^{(m)} - R_T \hat{\mu}]_{m=1}^{M_T} \end{aligned}$$

and $[v_i]_{i=1}^{M}$ represents stacking matrices side by side.

This system of equations can then be solved to estimate the means $\hat{\mu}_S^{\mathrm{BLUE}}$ of each model by solving the generalized least squares problem

$$\min_{\hat{\mu} \in \mathbb{R}^{N+1}} \|W_S - H_S \hat{\mu}\|^2_{\mathrm{Cov}[\varepsilon,\varepsilon]^{-1}}$$

which has the analytical solution

$$\hat{\mu}_S^{\mathrm{BLUE}} = \Phi^{-1} w, \qquad \Phi = \sum_{T \in S} M_T P^\top \Sigma_T^{-1} R_T, \qquad w = \sum_{T \in S} P_T \Sigma_T^{-1} \sum_{m=1}^{M_T} W_T^{(m)}. \tag{3.6}$$

Here $\Sigma_T = \mathrm{Cov}[X_T - \mu_T, X_T - \mu_T]$ is computed using a small pilot sample that is (often incorrectly) assumed to accurately estimate Σ_T.

The vector $\hat{\mu}_S^{\mathrm{BLUE}} \in \mathbb{R}^{N+1}$ is an estimate of the mean of each model. However,for a general vector α one can also compute $\hat{\mu}_{S,\alpha}^{\mathrm{BLUE}} = \alpha \hat{\mu}_S^{\mathrm{BLUE}}$. The variance of this quantity, which we will use shortly, is given by

$$\mathbb{V}[\hat{\mu}_{S,\alpha}^{\mathrm{BLUE}}] = \alpha^\top \Phi^{-1} \alpha. \tag{3.7}$$

3.2.2. Multi-arm bandits for BLUE

In this work we are concerned with minimizing the variance of the BLUE estimator of the highest fidelity model mean μ_0. This requires an estimate of the covariance matrix Σ_S, which can be used to compute $\Sigma_T, T \subseteq S$, and optimizing the number of evaluations M_T of each model set. In past work a pilot sample, i.e. a small number of evaluations of each models, was used to compute Σ_S and this estimate was used to optimize M_T. However, the inaccuracy of the correlations is ignored and the cost of computing the pilot is not optimized relative to the cost of the subsequent MF analysis. In this work we will use multi-arm bandit optimization to remove the reliance on inaccurate pre-computed covariances, which substantially reduces the computational cost of computing BLUEs.

First assume that the following linear relationship between the high-fidelity model and an ensemble $S_\dagger$ of low-fidelity models

$$X_0 = X_{S_\dagger}^\top \beta_{S_\dagger} + \beta_0 + \eta_{S_\dagger}.$$

Under this assumption, the variance of the BLUE estimator $\hat{\mu}_{S,\beta_S}^{\mathrm{BLUE}}$ of μ_0 is given by (3.7) with $\beta_S = [\beta_0, \beta_{S_\dagger}^\top]^\top$. To avoid the need for an a priori estimate of model correlations, we use multi-arm bandit resource allocation with an adaptive explore then commit (to exploitation) policy.

The exploration round is used to compute estimates of β_S using the same number of samples of each model, i.e.

$$\widehat{\beta}_S = (W_{S_\dagger^+}^\top W_{S_\dagger^+})^{-1} W_{S_\dagger^+}^\top W_{\{0\}} \qquad\qquad W_{S_\dagger^+} = [1, W_{S_\dagger}]$$

In the exploitation phase we use the remaining budget to generate additional evaluations of the low-fidelity models (but not the high) and compute a BLUE estimator $\hat{\mu}_{0,S}^{\mathrm{AETC}}$ conditioned on these fixed value of $\hat{\beta}_S$ using (3.6). Fixing the coefficients introduces a bias into the BLUE estimator that is caused by the use of empirical estimation of the covariance between models; this bias is ignored by existing literature. Consequently, we must only commit to the exploitation phase when this bias is predicted to be commensurate with the variance that can be obtained by committing the remaining budget to reducing variance using a BLUE.

Let t denote the number of evaluations of each model in the exploration round and M the number of exploration rounds, Then the asymptotic mean squared error (MSE) which is the sum of the squared-bias and variance of our estimator is given by

$$\widehat{\mathcal{L}}_S(z;M) = \frac{k_M(S)}{z} + \frac{\gamma_M(S)}{B - \widehat{c}_{\text{epr}} z}$$

Here

$$k_m(S) = \text{tr}(\bar{x}_{S_+}\bar{x}_{S_+}^\top \widehat{\Lambda}_S^{-1})\widehat{\sigma}_S^2 \qquad \gamma_m(S) = \min_{\substack{\mathcal{M} \\ \sum_{T\subseteq S}\widehat{c}_T m_T \leq 1}} \widehat{\beta}_S^\top \left(\sum_{T\subseteq S} m_T R_T^\top \widehat{\Sigma}_T^{-1} R_T\right)^{-1} \widehat{\beta}_S,$$

and

$$\widehat{\Lambda}_T = \frac{1}{M}\sum_{m\in[M]} W_T^{(m)}(W_T^{(m)})^\top \qquad \bar{x}_T = \frac{1}{M}\sum_{m\in[M]} W_T^{(m)}$$

$$\widehat{\sigma}_T^2 = \frac{1}{M-|T|-1}\sum_{m\in[M]}\left(X_0^{(m)} - (W_T^{(m)})^\top \widehat{\beta}_T\right)^2 \qquad \widehat{\Sigma}_T = \frac{1}{M}\sum_{m\in[M]} W_T^{(m)}(W_T^{(m)})^\top - \bar{x}_T\bar{x}_T^\top$$

$$\widehat{c}_T = \frac{1}{M}\sum_{m\in[M]} \mathsf{C}_T^{(m)}.$$

The optimization of γ is used to optimize the number of samples allocated to each low-fidelity model during the exploitation phase. Here $\mathcal{M} = \{m_T\}_{T\subseteq S}$ is the allocation of samples to all subsets $T \subseteq S$ that do not involve the high-fidelity model. For sets T involving the high-fidelity model M_T must be fixed to the number of samples M taken during the exploration round.

We use the asymptotic MSE as a loss function for multi-arm bandit optimization. The minimizer and optimum of this loss can be estimated as

$$M_S^*(M) = \underset{0<z<B/\widehat{c}_{\text{epr}}}{\arg\min}\ \widehat{\mathcal{L}}_S(z;M) = \frac{B}{\widehat{c}_{\text{epr}} + \sqrt{\frac{\widehat{c}_{\text{epr}}\gamma_M(S)}{k_M(S)}}} \qquad \mathcal{L}_S^*(m) = \frac{(\sqrt{\widehat{c}_{\text{epr}}k_M(S)} + \sqrt{\gamma_M(S)})^2}{B}.$$

The estimated optimal loss of exploiting model S after M rounds of exploration is given by

$$\mathcal{R}_S = \widehat{\mathcal{L}}_S(\min(\widehat{M}_S^*, M); M). \tag{3.8}$$

When $\min(\widehat{M}_S^*, M) <= M$ the bias is predicted to be smaller than the variance that can be obtained during exploitation at which time we must switch from exploration to exploitation.

In the exploitation phase we compute the MF estimate of the high-fidelity model mean. Specifically, we allocate the remaining budget to evaluating the low-fidelity models.

3.2.3. Algorithm

Our AETC strategy is summarized in Algorithm 2. In the exploration phase we first draw $N+2$ random samples of all model outputs. This number is the minimum number required to compute the quantities needed throughout the algorithm. In step 2, we compute the expected computational costs $\hat{c}_n$ (since we want to avoid computational effort to compute these costs *a priori* or the cost is stochastic, e.g. due to load balancing on HPC machines) of each model and the maximum number of exploration rounds assuming that one evaluation of each model is taken in each round. The algorithm then proceeds through a series of exploration rounds until it determines that it needs to commit to evaluating only a subset of models.

Each exploration round involves computing the mean squared error of the MF estimate $\hat{\mu}_{0,S}^{\mathrm{AETC}}$ for all subsets in $S \subseteq [N]$ (step 5). The number of samples is then incremented to improve our estimates of β in the exploration phase. Steps 8-11 implement a bisection strategy intended to increment the number of samples efficiently, without increasing them to much, so that the bias of the exploration phase is dominated by the variance that can be achieved in the exploitation phase. Step 13 returns $\hat{\mu}_{0,S}^{\mathrm{AETC}}$ when the bias of exploration and the variance of exploitation are balanced. At the end of each exploration phase we must update the maximum number of exploration rounds to account for changes to the expected computational cost of each round.

The exploration phase is used to compute an estimate $\hat{\beta}_{S^+}$ of β_{S^+}. In the exploitation phase we use this estimate in combination with additional evaluations of the low fidelity models, with the remaining computational budget, to compute an improved estimate of $\hat{\mu}_0$. Specifically we use the remaining budget to generate additional evaluations $V_T^{(m)}$ of the low-fidelity model using the optimal values of $M_T, T \subseteq S$.

$$\hat{\mu}_{0,S}^{\mathrm{AETC}} = \hat{\beta}_{S^+}^{\top} \Phi^{-1} w, \qquad \Phi = \sum_{T \in S} M_T P^{\top} \hat{\Sigma}_T^{-1} R_T, \qquad w = \sum_{T \in S} P_T \hat{\Sigma}_T^{-1} \sum_{m=1}^{M_T} V_T^{(m)}. \tag{3.9}$$

3.2.4. Numerical results

Consider a parameterized equation that governs displacement in linear elasticity defined over a square spatial domain $D = [0,1]^2$:

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{z}) = -\boldsymbol{F}, \tag{3.10}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^{2\times 2}$ is the two-dimensional plane stress tensor, related to the displacement $\boldsymbol{u} = (u,v)^T$ through,

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_x & \sigma_{xy} \\ \sigma_{xy} & \sigma_y \end{pmatrix}, \qquad \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix} = \frac{\kappa(\boldsymbol{z},\boldsymbol{x})}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{pmatrix} u_x \\ v_y \\ \frac{1}{2}(u_x + v_y) \end{pmatrix},$$

Here, the loading force $\boldsymbol{F}$ is non-zero only at the lower right of the structure, having value $\boldsymbol{F} = (0,-1)^T$, $\nu = 0.3$ is the Poisson's ratio, and the Young's modulus κ is uncertain. Specifically,

Algorithm 2: AETC-OPT algorithm for MF estimation of expectations

Input: B: total budget, models $\{X_n\}_{n=0}^{N+1}$

Output: an estimator for y

collect $N+2$ independent samples of $(X_0,\cdots,X_N)$ for exploration
compute $\widehat{c}_{\text{epr}} = \sum_{n=0}^{N} \hat{c}_n$ and the maximum exploration rounds $M = \lfloor \frac{B}{\widehat{c}_{\text{epr}}} \rfloor$
while $n+2 \le t \le M$ **do**
  for $S \subseteq [n]$ **do**
    compute the optimal loss $\mathcal{R}_S$ using (3.8)
  end for
  find the optimal model $\widehat{S}^* = \arg\min_{S\subseteq[n]} \mathcal{R}_S$
  if $m^*_{\widehat{S}^*}(t) > 2t$ **then**
    collect t additional exploration samples and let $t \leftarrow 2t$
  else if $t < m^*_{\widehat{S}^*}(t) \le 2t$ **then**
    collect $\left(\left\lceil \frac{t+m^*_{\widehat{S}^*}(t)}{2} \right\rceil - t \right)$ additional exploration samples and let $t \leftarrow \left\lceil \frac{t+m^*_{\widehat{S}^*}(t)}{2} \right\rceil$
  else
    return $\hat{\mu}^{\text{AETC}}_{0,S}$ defined in (3.9)
  end if
  update $\widehat{c}_{\text{epr}}$ and the maximum exploration rounds $M = \lfloor \frac{B}{\widehat{c}_{\text{epr}}} \rfloor$
end while

we let $\boldsymbol{z} \in \mathbb{R}^4$ be a random vector with independent components uniformly distributed on $[-1,1]$ and model κ as a truncated Karhunen-Loève expansion, given by

$$\kappa(\boldsymbol{z},\boldsymbol{x}) = 1 + 0.5 \sum_{i=1}^{4} \sqrt{\lambda_i}\phi_i(\boldsymbol{x}) z_i,$$

where (λ_i, ϕ_i) are ordered eigenpairs of an exponential covariance kernel on D. The geometry, boundary conditions, and loading in the structure is shown in Figure 3-4.

In the following we will use MF sampling methods to compute the expected value of a scalar QoI E which is the structural *compliance* or energy norm of the solution, measuring the elastic energy absorbed in the structure as a result of loading, that is

$$E(\boldsymbol{z}) = \int_D (\boldsymbol{u}(\boldsymbol{x},\boldsymbol{z}) \cdot \boldsymbol{F}) \mathrm{d}\boldsymbol{x}$$

We solve (3.10) for each fixed $\boldsymbol{z}$ via the finite element method with standard bilinear square isotropic finite elements on a rectangular mesh [7]. We form a ensemble of models by solving the governing equations with different mesh discretizations. To complement the highest-fidelity model, which uses a mesh size $h = 2^{-7}$, we create four low-fidelity models based on more economical discretizations: $h = 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}$. The computational time required to run each model is considered deterministic and inversely proportional to the mesh size squared, i.e., h^2; this corresponds to using a linear solver of optimal linear complexity. Moreover, we normalize cost so that the model with the lowest fidelity has unit cost, i.e.,

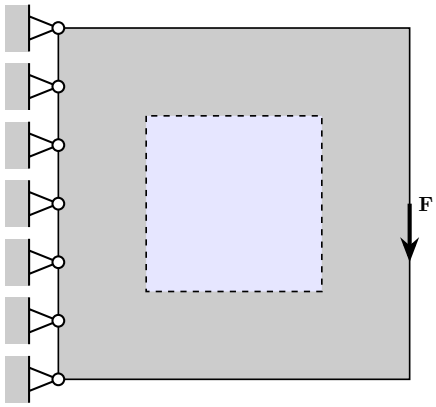


Figure 3-4. Geometry, boundary conditions, and loading for the linear elastic structure with the square domain.

$c_0 = 4096, c_1 = 64, c_2 = 16, c_3 = 4, c_4 = 1$. The correlations between the outputs of X_0 and X_1, X_2, X_3, X_4, computed using a large number of samples, are 0.976, 0.940, 0.841, -0.146, respectively.

In the left of Figure 3-5 we plot the accuracy of various MF procedures for increasing total budgets B, ranging from 4×10^5 to 2×10^6, incremented by 4×10^5 at a time. All MF estimators are more efficient than single-fidelity MC estimation, that is for a fixed total budget the MF estimators are always more accurate than the single-fidelity MC estimator. The original adaptive explore-then-commit (AETC) algorithm from [80] is more effective than MF MC (MFMC) [56] even though the latter requires estimates of correlation which we assume (impractically) does not impact the total budget. The arm (model subset) selected by AETC is $S = \{2,3,4\}$, whereas MFMC uses all four low-fidelity models.

The original AETC algorithm assumes that once the optimal arm has been identified all models in the chosen subset are evaluated the same number of times during the exploitation phase used to compute the expected compliance. This is sub-optimal and thus the AETC is not as efficient as the best unbiased linear estimator with an optimal sample allocation [69] (SAOB). However, this is not a fair comparison because we have again assumed that the correlation matrix that SAOB requires is computed for free. Thus SAOB can be considered the best performance that AETC like algorithms can achieve. When we apply our new algorithm (AETC-OPT-E) we obtain the best possible error subset of models $S = \{1,2,3,4\}$ (i.e. all low-fidelity models are used) and without the need for oracle correlation information.

We now repeat the above experiment with X_2 removed from consideration (middle of Figure 3-5). In this case, none of the subsets of the low-fidelity models satisfies the assumptions in [56, condition (20)], making MFMC non-applicable, however we use the previous MFMC result as a baseline for comparison for convenience. The results are similar and reported in Figure 3-5. The optimal model subset selected by AETC and AETC-opt is $S = \{1,3,4\}$, however AETC-OPT-E is

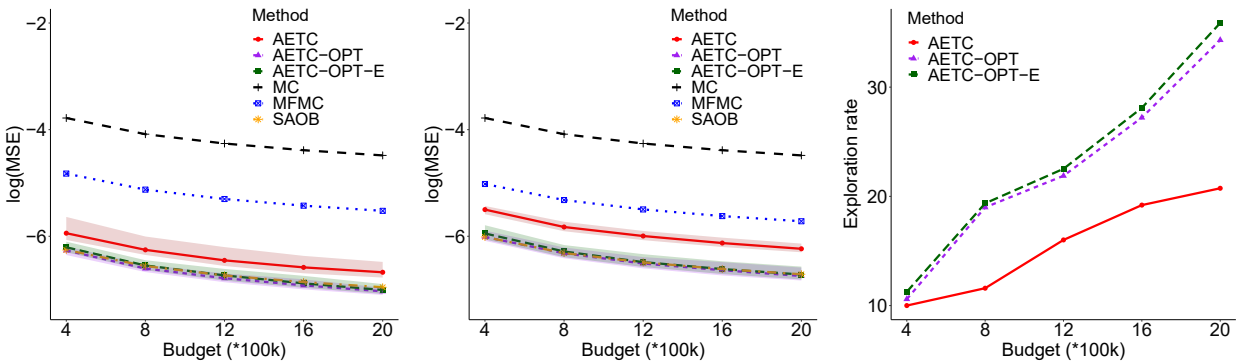


Figure 3-5. Comparison of the LRMC estimator given by the AETC algorithm, the optimal BLUE estimators given by the AETC-OPT and AETC-OPT-E algorithms, the MC estimator, the MFMC estimator, and the SAOB estimator as the total budget increases from 4×10^5 to 2×10^6 when the full low-fidelity models are available (left) and when the second low-fidelity model is removed (middle). For the latter we also compare the average exploration rates of AETC, AETC-OPT, and AETC-OPT-E (right).

significantly more accurate. Without X_2, AETC cannot select the original optimal model and thus has to spend more budget on exploitation to use highly correlated surrogates (i.e. X_1). This observation is verified in the right plot of Figure 3-5.

3.3. Non-hierarchical Gaussian processes

As previously mentioned, much of the MF modeling literature is focused on hierarchical relationships between information sources, and MF Gaussian process (GP) surrogates [42] are no exception. In this LDRD we developed non-hierarchical GP surrogates for unstructured (i.e. non-nested) datasets. Specifically, we consider a specific relationship where we assume that the low-fidelity models are marginally independent on the high-fidelity model. That is if the high-fidelity model is given, then the low-fidelity models are no longer independent of one another. In other words, information about the parameters used in one set of physics will inform the other set of physics because they are coupled together in a known way through the high-fidelity model.

Given multiple low-fidelity models, marginal independence leads to a special instance of MFNets we call a peer network, where all low-fidelity peer models have edges to the high-fidelity model but no direct edges between peers exist. Given a collection of $N_{\text{peer}} + 1$ models $f_k(x), k = 0, \ldots, N_{\text{peer}}$, where the $k = 0$ indexes the high-fidelity (HF) and the indices $k = 1, \ldots, N_{\text{peer}}$ its peer models, assuming marginal independence means that the HF model satisfies

$$f_0(x) = \sum_{k=1}^{N_{\text{peer}}} \rho_k(x) f_k(x) + \delta(x). \tag{3.11}$$

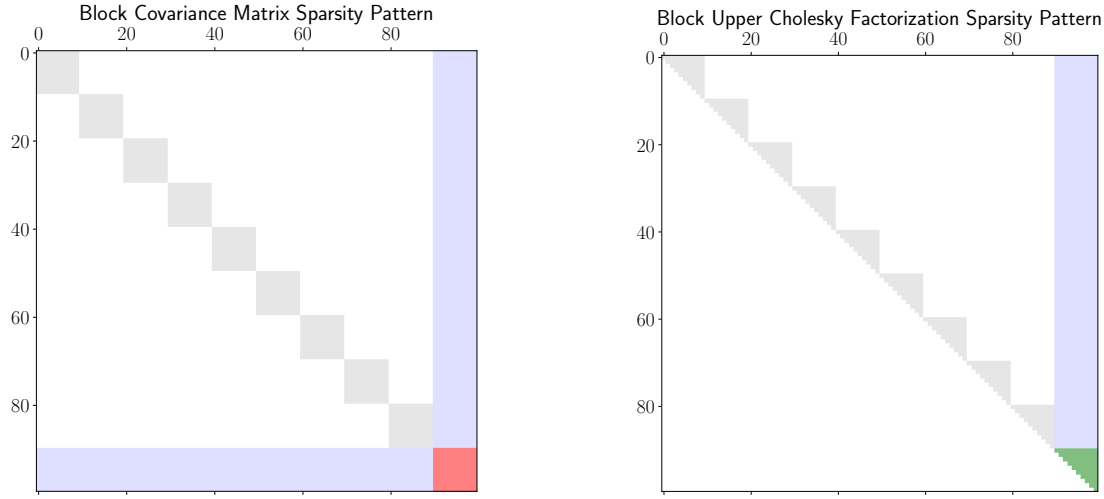
In the following we assume f_k and δ are Gaussian processes, the peer models are connected to the HF model through correlation polynomials $\rho_k(x)$, and the discrepancy $\delta(x)$ is used to account for systematic biases between the peers and HF model.

3.3.1. Algorithm

Given training data consisting of a collection of N_k feature-target pairs (X_k, y_k) for each model, we construct each GP simultaneously using all available data by constructing the block covariance matrix for the GP of f_0 and estimate its hyper-parameters via gradient-based minimization of the negative marginal log-likelihood (NMLL). Specifically, let $C_k(X_i, X_j)$ be the covariance matrix obtained by evaluating the GP kernel for $f_k, k = 1, \ldots, N_{\text{peer}}$ or the discrepancy δ when $k = 0$ at input point set pairs X_i and X_j. The entries of the block covariance matrix C for the peer network GP are obtained by computing the pairwise expectations between the different models in the network:

$$\begin{aligned}
\mathbb{E}[f_k f_k] &= C_k(X_k, X_k), \quad k = 1, \ldots N_{\text{peer}}, \\
\mathbb{E}[f_k f_0] &= [\mathbb{1}_{N_k} \otimes \rho_k(X_0)] \odot C_k(X_k, X_0), \quad k = 1, \ldots N_{\text{peer}}, \\
\mathbb{E}[f_0 f_0] &= C_0(X_0, X_0) + \sum_{k=1}^{N_{\text{peer}}} [\rho_k(X_0) \otimes \rho_k(X_0)] \odot C_k(X_0, X_0), \\
\mathbb{E}[f_i f_j] &= 0, \quad i, j \geq 1, i \neq j,
\end{aligned} \tag{3.12}$$

where $\mathbb{1}_{N_k}$ denotes a vector of ones of length N_k, $\otimes$ is the tensor-product, and $\odot$ denotes the Hadamard (entrywise) product.



(a) C **(b)** U_C

Figure 3-6. Sparsity patterns for the block covariance matrix C and its upper Cholesky factorization U_C for a collection of 10 models with 10 data points in each model. There are matrices associated with the peers (gray), off-diagonal, full matrices that couple peer and HF data (blue), the HF covariance matrix (red), and the upper Cholesky factorization U_S for the Schur complement matrix S (green). We note that U_C^{-1} has the same sparsity pattern as U_C.

The block covariance matrix C is sparse, and its Cholesky factorization may be computed using an algorithm with $O(\hat{N}^3)$ complexity [71], where $\hat{N} := \max_i (N_i)$ and N_i is the number of data points in the training set for f_i. Similarly, the evaluation of the NMLL and its gradient with

respect to hyper-parameters θ may be computed with an approach that has $O(\hat{N}^3)$ complexity. This result is significant because naive implementations have complexity $O(N^3)$, where $N := \sum_{i=0}^{N_{\text{peer}}} N_i$ is the total number of build points.

When conducting linear algebra operations, we reorder the models so that the matrices for the peer models are considered first and matrices associated with the HF model are considered last. As an example, consider a collection of 10 models where $N_k = 10$ for each model. The sparsity patterns for the block covariance matrix and its upper Cholesky factorization are shown in Figure 3-6. This "arrow" structure is the critical factor that allows us to minimize storage requirements (i.e. store only the non-zero blocks) and develop efficient algorithms. In this instance we may reduce the cost of training the peer GP network from $O(100^3)$ to $O(10^3)$, a savings on the order of 1000X.

To achieve these savings, consider the NMLL, which is a function of the block covariance matrix C and concatenation of target vectors $y := \left[y_1, \ldots, y_{N_{\text{peer}}}, y_0\right]^\top$. The standard approach for evaluating the NMLL requires the Cholesky factorization of the covariance matrix and a Cholesky-based linear solve for the vector α [65]:

$$\text{NMLL} = \frac{1}{2}\log\left(\det C\right) + \frac{1}{2} y^\top \underbrace{C^{-1} y}_{\alpha} + \frac{N}{2}\log\left(2\pi\right). \tag{3.13}$$

Computing the gradient of the NMLL requires these components and the inverse of the block covariance matrix:

$$\frac{\partial\left(\text{NMLL}\right)}{\partial\theta} = -\frac{1}{2}\text{tr}\left(\underbrace{\left(\alpha\alpha^\top - C^{-1}\right)}_{Q}\frac{\partial C}{\partial\theta}\right) = -\frac{1}{2}\left\langle Q, \frac{\partial C}{\partial\theta}\right\rangle_F = -\frac{1}{2}\left\langle Q_C, \frac{\partial C}{\partial\theta}\right\rangle_F, \tag{3.14}$$

where $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner product. While the Q matrix in the expression for the gradient is dense, we only need to compute its values where C is non-zero (Q_C).

We now describe how to efficiently compute these quantities. To do so we express the block covariance matrix as

$$C = \begin{bmatrix} A & B \\ B^\top & D \end{bmatrix},$$

where the A submatrix is the block diagonal matrix of peer covariances $A_k = C_k(X_k, X_k),\ k = 1, \ldots, N_{\text{peer}}$, B and $B^\top$ contain the off-diagonal terms, and D is the covariance matrix for f_0. In Figure 3-6a these are the gray, blue, and red matrices, respectively.

We perform a block Cholesky factorization $C = U_C^\top U_C$ using the Schur complement $S = D - B^\top A^{-1} B$:

$$C = \begin{bmatrix} I & 0 \\ B^\top A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} U_A^\top & 0 \\ B^\top U_A^{-1} & U_S^\top \end{bmatrix} \begin{bmatrix} U_A & U_A^{-\top}B \\ 0 & U_S \end{bmatrix}.$$

The Cholesky factors U_A are obtained by computing the Cholesky decomposition of the peer covariance matrices in A (i.e. $U_{A_k} = \text{Chol}(A_k)$). We then calculate $X := U_A^{-\top}B$ to determine the off-diagonal term in U_C and use it to obtain the intermediate quantity $Y := U_A^{-1}X$. Finally, we compute the Schur complement $S = D - B^\top Y$ and its Cholesky decomposition U_S. All of these steps may be performed in a block by block fashion such that the overall complexity is $O(\hat{N}^3)$. Furthermore, as shown in Figure 3-6b, the sparsity of U_C allows us to solve the linear system $CF = G$ where $F, G \in \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{L}}$ using block-wise forward and backward substitution with complexity $O(\hat{N}^2 L)$.

We note that $C^{-1} = U_C^{-1}U_C^{-\top}$ and that the inverse of a block triangular matrix is also block triangular. Consequently the expression for the inverse upper block Cholesky factorization is

$$U_C^{-1} = \begin{bmatrix} U_A^{-1} & Z \\ 0 & U_S^{-1} \end{bmatrix},$$

which requires the calculation of two inverse block triangular matrices and the unknown matrix Z. The matrices U_A^{-1} and U_S^{-1} are obtained through block triangular solves at $O(\hat{N}^3)$ cost. To determine Z we first examine

$$U_C^{-1}U_C = \begin{bmatrix} U_A^{-1} & Z \\ 0 & U_S^{-1} \end{bmatrix} \begin{bmatrix} U_A & X \\ 0 & U_S \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix},$$

which leads us to conclude that $Z = -U_A^{-1}XU_S^{-1} = -YU_S^{-1}$. Therefore the complexity of computing Z block by block is also $O(\hat{N}^3)$.

Our final tasks are to solve the linear system $C\alpha = y$ for $\alpha := \left[\alpha_1, \ldots, \alpha_{N_{\text{peer}}}, \alpha_0\right]^\top$ and construct C^{-1} to form Q_C. As mentioned previously, the sparsity of C allows us to only compute select portions of the dense matrix Q. We only need to consider outer products $\alpha_k\alpha_k^\top$ that contribute to the gradient (i.e. skip outer products between peer α_k). The general form of the inverse block covariance matrix is

$$C^{-1} = U_C^{-1}U_C^{-\top} = \begin{bmatrix} U_A^{-1} & Z \\ 0 & U_S^{-1} \end{bmatrix} \begin{bmatrix} U_A^{-\top} & 0 \\ Z^\top & U_S^{-\top} \end{bmatrix} = \begin{bmatrix} (U_A^{-1}U_A^{-\top} + ZZ^\top) & ZU_S^{-\top} \\ U_S^{-1}Z^\top & U_S^{-1}U_S^{-\top} \end{bmatrix}.$$

The outer product $ZZ^\top$ may be computed at a reduced cost by only calculating the terms that arise from the interactions between peer blocks. The rest of the entries of C^{-1} are already represented in a block-wise manner and are obtained through block triangular solves at a complexity of $O(\hat{N}^3)$.

Hyper-parameter estimation may be performed in an all-at-once or sequential manner. In the sequential approach the hyper-parameters for each peer GP are determined independently and are

are fixed when optimizing the remaining hyper-parameters (correlation polynomial coefficients and discrepancy kernel hyper-parameters). Both approaches yield the same results when each pair of input training points sets (X_k, X_0) $k = 1, \ldots, N_{\text{peer}}$ are nested (i.e. the points in X_0 are a subset of the points in X_k). A proof of this equivalence for hierarchical GP networks is present in [29].

After performing hyper-parameter estimation, it is natural to ask how to use the MF GP surrogate to predict values for the HF model f_0 that are not present in the training set. For GPs we may invoke the properties of the multivariate normal distribution to derive expressions for the posterior mean and covariance of the HF model at prediction points X_* by conditioning on the training data. For convenience we introduce the prediction matrices $P_k := \mathbb{E}[f_0(X_*) f_k(X_k)]$ and β_k:

$$
\begin{aligned}
P_k &= [\rho_k(X_*) \otimes \mathbb{1}_{N_k}] \odot C_k(X_*, X_k),\ k = 1 \ldots N_{\text{peer}}, \\
P_0 &= C_0(X_*, X_k) + \sum_{k=1}^{N_{\text{peer}}} [\rho_k(X_*) \otimes \rho_k(X_k)] \odot C_k(X_*, X_0), \\
P &:= \left[P_1, \ldots, P_{N_{\text{peer}}}, P_0\right]^\top, \\
\beta &:= \left[\beta_1, \ldots, \beta_{N_{\text{peer}}}, \beta_0\right]^\top = C^{-1} P^\top.
\end{aligned}
\tag{3.15}
$$

The mean μ_0 and covariance Σ_0 for the HF model are

$$
\begin{aligned}
\mu_0 &= P\alpha = \sum_{k=0}^{N_{\text{peer}}} P_k \alpha_k, \\
\Sigma_0 &= \mathbb{E}[f_0(X_*) f_0(X_*)] - P\beta, \\
&= C_0(X_*, X_*) + \sum_{k=1}^{N_{\text{peer}}} [\rho_k(X_*) \otimes \rho_k(X_*)] \odot C_k(X_*, X_*) - \sum_{k=0}^{N_{\text{peer}}} P_k \beta_k.
\end{aligned}
\tag{3.16}
$$

3.3.2. Numerical results

Consider the following ensemble of models that conforms to the structure assumed in(3.11)

$$
\begin{aligned}
f_2(x) &= 0.4x - x \sin\left(\frac{5\pi x}{2}\right), \\
f_1(x) &= 2x^2 - \frac{1}{2}\sin(2\pi x + 0.3), \\
f_0(x) &= -\frac{1}{2}x^3 \sin\left(\frac{\pi x}{2}\right) + f_1(x) + f_2(x).
\end{aligned}
\tag{3.17}
$$

Now given training data obtained from these 1D models, we generate approximations $f_0(x), f_1(x)$, and $f_2(x)$ using the peer network approach described in this section and compare them to standard single-fidelity (SF) GPs that are constructed in isolation. Each GP in the network has a squared exponential kernel and the correlation polynomials are constants.

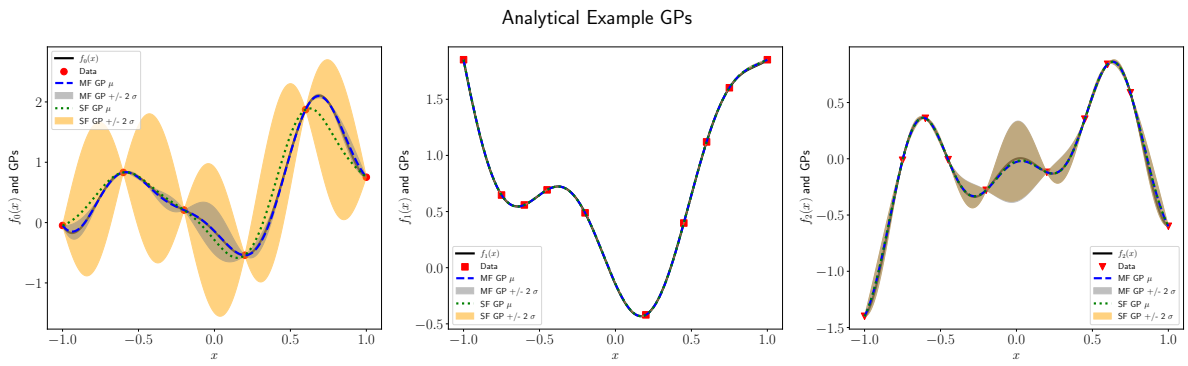


Figure 3-7. Peer network multi-fidelity (MF) and single-fidelity (SF) GPs for the model problem with nested training data from analytic functions f_0, f_1, and f_2 (left-to-right). Here the input point sets X_1 and X_2 contain the same 10 points, and X_0 consists of a subset of 5 of those points.

Figure 3-7 depicts the the functions (3.17) and their approximations. We note that the SF and MF approaches produce identical approximations for $f_1(x)$ and $f_2(x)$, but their representations of $f_0(x)$ are dramatically different. The MF peer network strategy produces a GP with a mean that is close to $f_0(x)$ and has considerably lower uncertainty than that of the SF GP, and the estimated values of the (constant) correlation polynomials are $\rho_1 = 1$ and $\rho_2 = 1.08$. These results clearly show the benefit of our MF approach.

Another application that can benefit from our methods is digital twin modeling for a fleet of assets. We may represent the relationship between digital twins using a peer network and develop models for each asset that fuse information from the fleet. A manuscript on this topic is in preparation [71].

3.4. Bayesian MFNets

The non-hierarchical Gaussian processes presented in Section 3.3 are a specific case of our MFNets framework [25] that targets ensembles of low-fidelity models which cannot be a prior ordered in terms of accuracy cost. Unlike the original algorithm, the work in Section 3.3 uses the probabilistic nature of Gaussian processes to efficiently estimate error in the MF surrogate. This section presents a generalization of the algorithm in Section 3.3 that can be used to construct MF surrogates of any un-ordered, partially ordered or fully ordered ensemble of models. This work was carried out in collaboration with Alex Gorodetsky and Trung Pham at the University of Michigan.

3.4.1. MFNets

Our goal is to learn the relationship between surrogates $f_k : \mathcal{X}_k \to \mathbb{R}$ of each information source in an ensemble of M models. We use a directed acyclic graph to encode the relationship between the individual surrogates f_k, and this graph represents the MF model. A DAG $\mathcal{G}$ is a tuple $(V, \mathcal{E})$ of nodes and edges, respectively, where the nodes are isomorphic to the positive integers, and thus

can be indexed by $k = 1, 2, \ldots, M$. The graph consists of $M = |V|$ nodes representing M information sources. The directed edges $(j \to i)$ or e_{ji} encode explicit dependencies between node (source) j to node i. We refer to the parents of a node $k \in V$ as those nodes that have an edge exiting them and entering k, i.e., $pa(k) = \{\ell \in V : (\ell \to k) \in \mathcal{E}\}$. The direct children of a node k are denoted by $children(k) = \{\ell \in V : (k \to \ell) \in \mathcal{E}\}$ while the set of all its successors is $successors(k)$. A path along the graph is a sequence of nodes along a set of directed edges. A path, denoted $path(i_1, i_2, \ldots, i_m)$, exists on a DAG if $(i_j \to i_{j+1}) \in \mathcal{E}$ for $j = 1, \ldots, m-1$. Finally, we will denote the ancestors of a node k by $anc(k)$. The ancestors are all those nodes ℓ from which there exists a path in $\mathcal{G}$ to node k. The *roots* of the graph are those nodes with no parents.

The use of DAGs provides a general framework for constructing existing MF surrogates while facilitating the constructions of new surrogates that exploit different relationships between models. For example, the non-hierarchical Gaussian processes of the previous section can be represented by a graph with edges that connect all low-fidelity models to the high-fidelity model but has no edges between the low-fidelity models. Hierarchical methods, such as hierarchical Gaussian processes correspond to DAGs that connect a lower-fidelity model to the next highest-fidelity model in the hierarchy.

In [25] we constructed MFNets surrogates for model relationships encoded with arbitrary DAGs using maximum likelihood estimation that provided a point estimate of the unknown surrogates parameters. In this work we develop an efficient Markov Chain MC (MCMC) procedure for sampling the posterior distribution of the unknown parameters.

3.4.2. Algorithm

Assume that each surrogate f_k is a linear subspace model parameterized as a linear combination of functions $f_k(x;\theta) = V_k^{\mathrm{T}}(x)\theta$, where $V : \mathcal{X}_k \to \mathbb{R}^p$, $\theta \in \mathbb{R}^p$ and $p \in \mathbb{Z}_+$.[1] We sometimes parameterize the basis functions explicitly so that $V_k(x) = [v_{k1}(x), \ldots, v_{kp}(x)]$ for $v_{ki} : \mathcal{X}_k \to \mathbb{R}$. If the bases $\{v_{ki}\}$ are complete in L_2 as $p \to \infty$, then this surrogate *converges for all functions in* L_2. When the bases are evaluated at n inputs $\boldsymbol{x}$, then $V_k^\top(\boldsymbol{x}) \in \mathbb{R}^{n \times p}$ represents a Vandermonde-like matrix whose rows correspond to the basis functions evaluated at each input and whose columns correspond to the evaluation of a single basis function at all inputs. Using these definitions and assuming that the child of each node is a multiplicative and additive correction of its parents, the surrogate of a node k satisfies

$$f_k(x) = \sum_{j \in pa(k)} \rho_{jk}(x) f_j(x) + V_k^{\mathrm{T}}(x)\beta_k,$$

where we assume that $\rho_{jk} = W_{jk}(x)^\top \alpha_{jk}$.

Our work in [25] developed a backwards propagation scheme for computing the gradients of the likelihood function which was used to efficient estimate unknown parameters using MLE. These gradients can also be used to sample the posterior distribution of the unknown parameters using gradient-based MCMC methods such as Hamiltonian MC. However, we found that the effective

[1] $\mathbb{Z}_+$ denotes the set of positive integers.

sample size of the MCMC sampler can be significantly improved by leveraging the analytical conditionals of MFNets surrogates for Gibbs sampling. In this section, we derive an efficient computation of the analytical conditionals and use them to construct an efficient Gibbs sampler.

In the following we will assume we have already collected data from each model in the form of input-output pairs $\left(x_k^{(j)}, y_k^{(j)}\right)$ for $k = 1, \ldots, M$, and $j = 1, \ldots, n_k$, where $n_k \in \mathbb{Z}_+$, $x_k^{(j)} \in \mathcal{X}_k \subseteq \mathbb{R}^d$, $d \in \mathbb{Z}_+$, and $y_k^{(j)} \in \mathbb{R}$. We will use bold letters to indicate ordered collections of items. For example, the sets of training samples and associated values, respectively given by $\boldsymbol{x}_k = \left[x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(n_k)}\right]$ and $\boldsymbol{y}_k = \left[y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(n_k)}\right]$. Lastly, let $\mathcal{Y} = \{\boldsymbol{y}_k\}_{k=1}^M$ denote the collection of all data of all models.

The graphical structure of MFNets implies that outputs of a node k are linear with respect to all parameters β_i for $i \in anc(k) \cup \{k\}$. In this case, when all the edge parameters are fixed, specifying a Gaussian prior will yield a classical linear-Gaussian inverse problem. Similarly, when all parameters except the edge parameters α_{ij} are fixed, the outputs of node k become linear with respect to α_{ij}. The implementation of our Gibbs sampler is shown in Algorithm 3.

Let $\alpha_e^{(i)}$ for $e \in \mathcal{E}$ and $\beta^{(i)}$ denote a sample of the edge and node parameters in the i^{th} loop, respectively. After taking the inputs and initializing the parameters, in each loop the algorithm runs through every edge of the graph to draw a sample of the edge parameters, and then fixes all edge parameters to draw a sample of the node parameters.

By fixing either the edge or the node parameters, the conditional posterior becomes linear. Specifically, given a vector of inputs $\boldsymbol{x}$, the noisy observations $\boldsymbol{y} = A(\boldsymbol{x})\theta + \varepsilon, \varepsilon \sim \mathcal{N}(0, \Gamma)$, and a Gaussian prior for the parameter $p(\theta) = \mathcal{N}(m, \Sigma)$, the linear-Gaussian likelihood is $p(\boldsymbol{y} \mid \boldsymbol{x}, \theta) = \mathcal{N}(A(\boldsymbol{x})\theta, \Gamma)$ and the mean and variance of the Gaussian posterior $\mathcal{N}_p\left(m_p^{(\theta)}, \Sigma_p^{(\theta)}\right)$ are given by

$$m_p^{(\theta)} = m + \Sigma A(\boldsymbol{x})^{\mathrm{T}} \left(A(\boldsymbol{x})\Sigma A^{\mathrm{T}}(\boldsymbol{x}) + \Gamma\right)^{-1} (\boldsymbol{y} - A(\boldsymbol{x})m) \tag{3.18}$$

$$\Sigma_p^{(\theta)} = \Sigma - \Sigma A(\boldsymbol{x})^{\mathrm{T}} \left(A(\boldsymbol{x})\Sigma A^{\mathrm{T}}(\boldsymbol{x}) + \Gamma\right)^{-1} A(\boldsymbol{x})\Sigma \tag{3.19}$$

In the following sections, we derive the computation of the required vectors and matrices, i.e., $m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}, m_p^{(\beta)}$ and $\Sigma_p^{(\beta)}$.

Node models First, consider the posterior on β_k when the edge-functions are fixed in the case that data is obtained only for node k. If node k is a root node then we simply have

Algorithm 3: Gibbs sampler for MFNets

Require: MFNets $\mathcal{G}$; locations and the corresponding observed data; number of samples n;
Ensure: n samples of the parameters

1: Set initial parameters $\alpha_e^{(0)}$ for $e \in \mathcal{E}$ and $\beta^{(0)}$
2: **for** $i = 1, 2, \ldots, n$ **do**
3: $\quad \mathcal{E}_1 = \emptyset, \mathcal{E}_2 = \mathcal{E}$
4: $\quad$ **for all** edge e of $\mathcal{G}$ **do**
5: $\quad\quad \mathcal{E}_2 \leftarrow \mathcal{E}_2 \setminus \{e\}$
6: $\quad\quad$ Get the matrices from Algorithm 6
7: $\quad\quad$ Draw a sample $\alpha_e^{(i)}$ from $p\left(\mathcal{Y}, \alpha_e \mid \alpha_{e_1}^{(i)}, \alpha_{e_2}^{(i-1)}, \beta^{(i-1)} \text{ for } e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2\right) = \mathcal{N}_p\left(m_p^{(\alpha_e)}, \Sigma_p^{(\alpha_e)}\right)$
8: $\quad\quad \mathcal{E}_1 \leftarrow \mathcal{E}_1 \cup \{e\}$
9: $\quad\quad$ Update $\mathcal{G}$ using $\alpha_e^{(i)}$
10: $\quad$ **end for**
11: $\quad$ Get the matrices from Algorithm 5
12: $\quad$ Draw a sample $\beta^{(i)}$ from $p\left(\beta \mid \mathcal{Y}, \alpha_e^{(i)} \text{ for } e \in \mathcal{E}\right) = \mathcal{N}_p\left(m_p^{(\beta)}, \Sigma_p^{(\beta)}\right)$
13: $\quad$ Update $\mathcal{G}$ using $\beta^{(i)}$
14: $\quad$ Collect i^{th} sample $\left\{\alpha_e^{(i)} \text{ for } e \in \mathcal{E}, \beta^{(i)}\right\}$ of the parameters
15: **end for**

$A_k(\boldsymbol{x}) = V_k(\boldsymbol{x})\beta_k$. Otherwise, we perform a recursive calculation along the graph

$$\begin{aligned} f_k(x) &= \sum_{j \in pa(k)} \rho_{jk}(x) f_j(x) + V_k^{\mathrm{T}}(x)\beta_k \\ &= \sum_{j \in pa(k)} \rho_{jk}(x) \left(\sum_{\ell \in pa(j)} \rho_{\ell j}(x) f_\ell(x) + V_j^{\mathrm{T}}(x)\beta_j \right) + V_k^{\mathrm{T}}(x)\beta_k \\ &= \sum_{j \in pa(k)} \rho_{jk}(x) \sum_{\ell \in pa(j)} \rho_{\ell j}(x) f_\ell(x) + \sum_{j \in pa(k)} \rho_{jk}(x) V_j^{\mathrm{T}}(x)\beta_j + V_k^{\mathrm{T}}(x)\beta_k \\ &= \sum_{j \in pa(k)} \sum_{\ell \in pa(j)} \rho_{jk}(x)\rho_{\ell j}(x) f_\ell(x) + \sum_{j \in pa(k)} \rho_{jk}(x) V_j^{\mathrm{T}}(x)\beta_j + V_k^{\mathrm{T}}(x)\beta_k \end{aligned} \tag{3.20}$$

Note the linearity of f_k with respect to its immediate parent nodes $j \in pa(k)$ and the nonlinearity of the interactions between the parameters of the edges, due to the products of ρ_{jk} and $\rho_{\ell j}$.

To enable a simplified exposition, we simplify the expression for f_k so that

$$f_k(x) = \sum_{j \in anc(k)} PathProduct\left[\hat{P}(j,k)\right] V_j^{\mathrm{T}}(x)\beta_j + V_k^{\mathrm{T}}(x)\beta_k, \tag{3.21}$$

where the $PathProduct\left[\hat{P}(j,k)\right]$ function computes the sum of the products of the edge functions along all the paths from node j to k, and $\hat{P}(j,k)$ is the set of all the paths from node j to k. For example, considering the graph in Figure 3-8, the expression at node 3 becomes

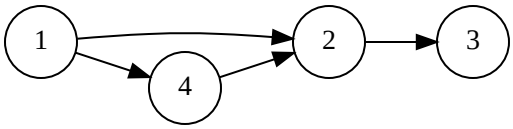


Figure 3-8. An example of a 4 node MFNets graph.

$$f_3(x) = V_1^{\mathrm{T}}(x)\,\beta_1\,(\rho_{12}\rho_{23} + \rho_{14}\rho_{42}\rho_{23}) + V_4^{\mathrm{T}}(x)\,\beta_4\rho_{42}\rho_{23} + V_2^{\mathrm{T}}(x)\,\beta_2\rho_{23} + V_3^{\mathrm{T}}(x)\,\beta_3. \qquad (3.22)$$

In Algorithm 4 we show how the *PathProduct*$[\cdot]$ function is computed recursively. The algorithm utilizes a depth-first search strategy to enumerate all paths from source node i to target node j and performs the necessary computations along each path with the location $\boldsymbol{x}_k$ of node k. We note that the filter set $\mathcal{F} = \{r,t\}$ is to remove the paths from node i to j passing through the edge $(r \to t)$; otherwise, $\mathcal{F} = \emptyset$. Such removal is needed in Algorithms 7 and 8. Moreover, Algorithm 4 can avoid recomputing products if each node ℓ remembers its visited status and the quantity $\hat{\rho}_{\ell j}$ which is the path product from node ℓ to j, i.e., $\hat{\rho}_{\ell j} = \mathit{PathProduct}[\ell, j]$. In other words, if a node ℓ is visited, then we can reuse $\hat{\rho}_{\ell j}$ to calculate other path products.

Algorithm 4: Compute the sum of the products of the edge functions along all the paths from node i to j

Require:*PathProduct*(MFNets $\mathcal{G}$, source node i, target node j, location $\boldsymbol{x}_k$, $\mathcal{A} = anc(j) \cup \{j\}$, filter set $\mathcal{F}$)
if node i is visited **then**
  return $\hat{\rho}_{ij}$
end if
Set node i to be visited
$\mathcal{S} = children(i) \cap \mathcal{A}$
if $\mathcal{F}$ is not empty and $i = r$ **then**
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{t\}$
end if
for $s \in \mathcal{S}$ **do**
  $\rho_{is} = \alpha_{is} \cdot W_{is}^{\mathrm{T}}(\boldsymbol{x}_k)$ ▶ dot product between vector and matrix = vector
  $\hat{\rho}_{ij} \leftarrow \hat{\rho}_{ij} + \rho_{is} * \mathit{PathProduct}(\mathcal{G}, s, j, \boldsymbol{x}_k, \mathcal{A}, \mathcal{F})$ ▶ element-wise product between vectors = vector
end for
return $\hat{\rho}_{ij}$

The *PathProduct*$[\cdot]$ function is used in Algorithm 5 to compute the matrix A of the linear model. Supposing observed data is available at node k, for each node $i \in \mathcal{A}$ we compute $\hat{\rho}_{ik}$ and the matrix a_i. Then, A can be assembled using a_i. However, care must be taken to address two practical issues that arise when constructing A. First, if data is available at multiple nodes, care must be taken to guarantee the correct correspondence between $A(\boldsymbol{x})\beta$ and $\boldsymbol{y}$. Second, if a node does not depend on some other nodes, A must be filled with zeros at appropriate entries.

Algorithm 5: Sample node parameters when fixing edge parameters

Require: MFNets $\mathcal{G}$: basis, graph structure; node k; locations $x_k = \{x_k^{(1)}, \ldots, x_k^{(m_k)}\}$;
Ensure: Compute A, linear-model computation for β

```
A = anc(k) ∪ {k}, F = ∅
ρ̂_kk = 1, ρ̂_ik = 0 for i ∈ anc(k)
for i ∈ A do
    ρ̂_ik = PathProduct(G, i, k, x_k, A, F)
    a_i = ρ̂_ik * V_i^T(x_k)
end for
Assemble A from a_i for i ∈ A
```

After assembling the matrix A, an analytical Gaussian posterior can be readily obtained and sampled according to (3.18) and (3.19). Next we consider sampling the edge parameters α_{ij}.

Edge models Equation (3.20) contains products of edge functions. Consequently, even when fixing edge parameters β the resulting surrogate is not linear in all the parameters of the edge functions and thus we cannot construct an analytic expression for the joint posterior of all edge parameters $\{\alpha_{ij} : (i \to j) \in \mathcal{E}\}$. However, fixing the parameters of all but one edge functions does result in a linear-Gaussian inverse problem[2]. When fixing all node parameters and all edge parameters except α_{ij}, we observe that only node $k \in \mathit{successors}(j) \cup \{j\}$ depends on α_{ij}, and based on (3.21) the expression at node k can be written as

$$\begin{aligned} f_k(x) &= f_i(x)\rho_{ij}\mathit{PathProduct}\left[\hat{P}(j,k)\right] + \left[\sum_{\ell \in anc(k)} \mathit{PathProduct}\left[\tilde{P}(\ell,k,e_{ij})\right] V_\ell^{\mathrm{T}}(x)\beta_\ell + V_k^{\mathrm{T}}(x)\beta_k\right] \\ &= f_i(x)\mathit{PathProduct}\left[\hat{P}(j,k)\right] W_{ij}^{\mathrm{T}}(x)\alpha_{ij} + D_k = C_k\alpha_{ij} + D_k, \end{aligned} \tag{3.23}$$

where $\tilde{P}(\ell,k,e_{ij})$ is the sets of all the paths from node ℓ to k excluding the paths which pass through the edge e_{ij}. Therefore, to sample α_{ij}, we need to compute C_k and D_k for $k \in \mathit{successors}(j) \cup \{j\}$ and $\boldsymbol{y}_k \neq \emptyset$.

Algorithm 6 shows how to compute the required vectors and matrices to draw a sample of the edge parameters α_{ij} when fixing all other parameters. The depth-first search strategy starts with putting node j into a first-in-first-out (FIFO) queue and then visits each successors s of j, which must have observed data, to compute C_s and D_s. To construct the linear model for α_{ij}, we can stack all C_s matrices to make A and subtract D_s from the data $\boldsymbol{y}_s$. Equations (3.18) and (3.19), can then be used to obtain the the posterior mean and covariance of α_{ij}, and draw a sample from the posterior.

Algorithms 7 and 8 used to compute D and C are straightforward. We calculate the path products first, and then D_k and C_k according to (3.23). We note that the computation of D_k needs to remove all paths passing through the edge $(i \to j)$, i.e., $\mathcal{F} = \{i, j\}$.

[2]Technically, one can group edges that do not lie along the same path from some ancestral node to the target node k. However, for simplicity of presentation, we fix all but one edge at a time.

Algorithm 6: Compute the required vectors and matrices to sample α_{ij}

Require: MFNets $\mathcal{G}$; edge e_{ij}; locations x_i and x_j;
Ensure: Compute A, linear-model computation for α_{ij}

queue = FIFOQueue()
queue.put(j)
$D_j = ComputeD(\mathcal{G}, i, j, j)$ ▶ Algorithm 7
$C_j = ComputeC(\mathcal{G}, i, j, j)$ ▶ Algorithm 8
while *queue* is not empty **do**
  $l = queue.get()$
  for $s \in children(l)$ **do**
    if $\boldsymbol{y}_s \neq \emptyset$ **then**
      $D_s = ComputeD(\mathcal{G}, i, j, s)$
      $C_s = ComputeC(\mathcal{G}, i, j, s)$
      queue.put(s)
    end if
  end for
end while
Assemble A from C_s, subtract D_s from data $\boldsymbol{y}_s$

Algorithm 7: $ComputeD(\mathcal{G}, i, j, k)$

$\mathcal{A} = anc(k) \cup \{k\}$, $\mathcal{F} = \{i, j\}$
$\hat{\rho}_{kk} = \mathbf{1}, D_k = \mathbf{0}, \hat{\rho}_{sk} = \mathbf{0}$ for $s \in anc(k)$
for $s \in \mathcal{A}$ **do**
  $\hat{\rho}_{sk} = PathProduct(\mathcal{G}, s, k, \boldsymbol{x}_k, \mathcal{A}, \mathcal{F})$
  $D_k \leftarrow D_k + \hat{\rho}_{sk} * \left(\beta_s \cdot V_s^{\mathrm{T}}(\boldsymbol{x}_k)\right)$
end for
return D_k

Algorithm 8: $ComputeC(\mathcal{G}, i, j, k)$

$\mathcal{A}_k = anc(k) \cup \{k\}$, $\mathcal{A}_i = anc(i) \cup \{i\}$, $\mathcal{F} = \emptyset$
$\hat{\rho}_{ii} = \mathbf{1}, f_i = \mathbf{0}, \hat{\rho}_{si} = \mathbf{0}$ for $s \in anc(i)$
for $s \in \mathcal{A}_i$ **do**
  $\hat{\rho}_{si} = PathProduct(\mathcal{G}, s, i, \boldsymbol{x}_k, \mathcal{A}_i, \mathcal{F})$
  $f_i \leftarrow f_i + \hat{\rho}_{si} * \left(\beta_s \cdot V_s^{\mathrm{T}}(\boldsymbol{x}_k)\right)$
end for
$\hat{\rho}_{kk} = \mathbf{1}, \hat{\rho}_{jk} = PathProduct(\mathcal{G}, j, k, \boldsymbol{x}_k, \mathcal{A}_k, \mathcal{F})$
$C_k = (f_i * \hat{\rho}_{jk}) * W_{ij}^{\mathrm{T}}(\boldsymbol{x}_k)$ ▶ row-wise product product between vector and matrix = matrix
return C_k

3.4.3. Numerical results

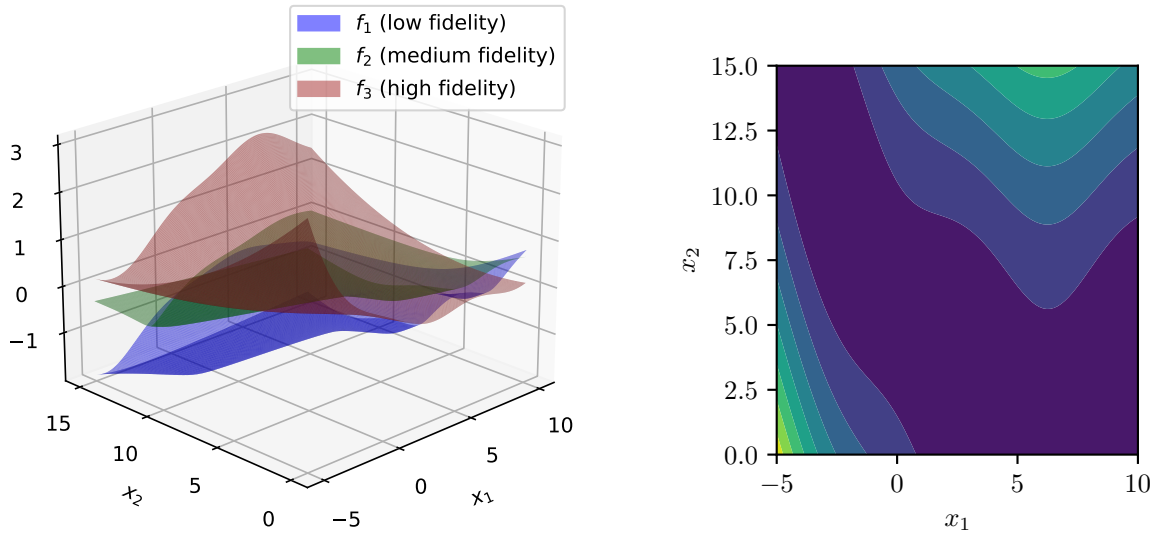
Consider a high-fidelity model given by the two-dimensional Branin function of $x = (x_1, x_2)$ [73]

$$f_3(x) = \left(\frac{-1.275x_1^2}{\pi^2} + \frac{5x_1}{\pi} + x_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos(x_1) + 10, \tag{3.24}$$

In the following we will use our Bayesian MFNets algorithm to build a surrogate of this model using medium- and low-fidelity functions given by

$$f_2(x) = 10\sqrt{f_3(x-2)} + 2(x_1 - 0{,}5) - 3(x_2 - 1) - 1, \tag{3.25}$$
$$f_1(x) = f_2(1.2(x+2)) - 3x_2 + 1. \tag{3.26}$$

The functions f_1 and f_2 follow the setup in [59]. The surfaces of these functions are plotted in Figure 3-9.



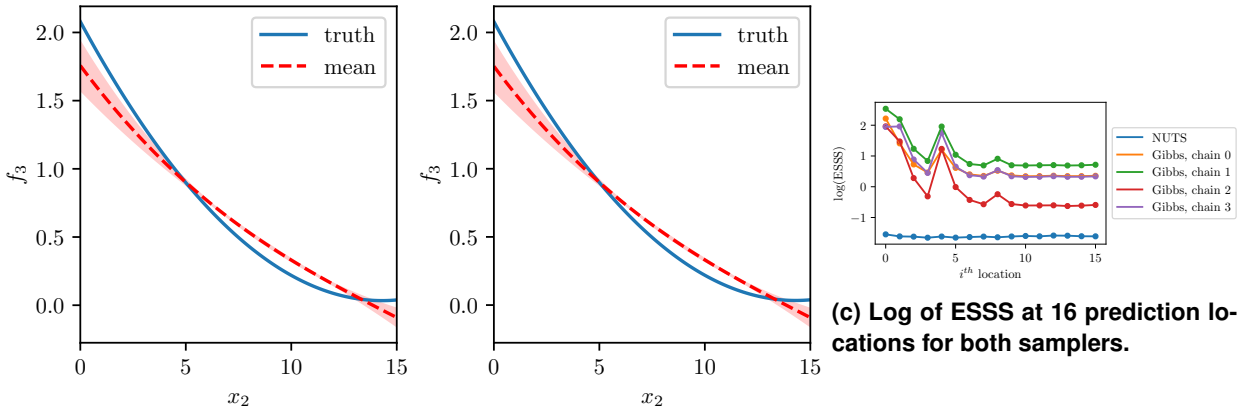
(a) 3D plot of functions. **(b) Contour plot of f_3.**

Figure 3-9. Plots of the MF Branin functions.

To learn the high-fidelity function f_1, we choose a 3-node graph with increasing fidelity, i.e., 1 (low) $\to$ 2 (mid) $\to$ 3 (high). We also use a third-degree polynomial as the node basis $V(x)$ and a first-degree polynomial as the edge basis $W(x)$. We then use both our Gibbs sampler and the NUTS sampler [32], implemented in PyMC3 [67], to train the surrogate using 80, 40, and 20 data points generated from f_1, f_2, and f_3, respectively. Both samplers start from the MAP point calculated using PyMC3.

In Figures 3-10a and Figures 3-10b, we show a cross section of the high-fidelity model at $x_1 = -3.95$, the mean of the outputs of node 3, and the uncertainty bound of the mean when using Gibbs and NUTS sampling, respectively. The pointwise variance obtained from both samplers are very close. The uncertainty bound depends depends on the noise standard deviation σ_ε and the

variance of the prior, which we have assumed to be a Gaussian with mean zero and unit variance. Finally, we calculate the effective sampler size per second (ESSS) of both samplers which depends on the locations at which predictions are desired. The first chain of the Gibbs sampler starts at the MAP point while the other three chains use perturbations of the MAP point. Regardless of the initial point used by Gibbs, Figure 3-10c, show that by leveraging the structure of our MFNets surrogates, the ESSS of the Gibbs sampler is orders of magnitude larger than that of NUTS.



(a) Gibbs sampler, 10^4 samples **(b) NUTS, 10^4 samples** **(c) Log of ESSS at 16 prediction locations for both samplers.**

Figure 3-10. Figures a and b depict a cross section of the high-fidelity model (node 3) at $x_1 = -3.95$ for $\sigma_\varepsilon = 0.01$. Figure c depicts the effective sample size of each sampler.

3.5. Plasma physics mission exemplar

The Z machine is one of Sandia's flagship experimental facilities for studying high energy density physics such as material response and magnetized liner inertial fusion (MagLIF). Z delivers mega-amps of current from energy storage units arranged in an outer circle, tens of meters in diameter, to a target in the center on the scale of a centimeter. One of the key energy delivery mechanisms are the magnetically insulated transmission lines (MITL). During the course of an experiment lasting tens of nanoseconds, species are desorbed from the MITL surface into the near vacuum space between cathode and anode, creating a plasma.

During this project we deployed our MF sampling algorithms to estimate the uncertainty in predictions from a model of expanding plasma. This example, first proposed in [40], was designed to investigate the effectiveness of MF methods at reducing the cost of quantifying uncertainty in high-energy density physics models relevant to predicting the power flow in the Sandia Z machine. Specifically, the plasma expansion problem is a simple prototype designed to evaluate how well the plasma codes can simulate low density plasmas in a very low pressure environment, similar to that of the MITL during a Z shot.

In the following we modify the problem setup used in [40]. Specifically, we use the same initial condition, QoI and uncertain parameters, but repeat the analysis in [40] using a different

numerical model[3] and then show the benefits of the model tuning algorithms developed in this project.

3.5.1. Plasma Model

Drekar is a five-moment multi-fluid electromagnetic/electrostatic fluid solver [47, 15], developed at Sandia. Multi-fluid continuum "fluid" plasma equations are derived from Boltzmann's equation by taking moments of conserved quantities - mass, momentum and energy - with respect to the velocity distribution function f_0. This results in a set of transport equations for mass, momentum and energy for each species (resembling Euler equations of compressible fluid dynamics with additional body force terms). Collision source terms are derived by taking moments of the Boltzmann collision integral.

At low densities collisions are infrequent and non-equilibrium velocity distributions occur. As densities increase so do collision rates which act to maintain equilibrium. The ramification is that for low densities, kinetic (PIC) codes are accurate because the velocity distribution is computed instead of assumed and efficient because lower densities require fewer particles in the simulation and low collision rates (an expensive component of PIC codes). However, for higher density, fluid codes become more efficient and approach the accuracy of kinetic codes because higher densities result in higher collision rates which is a mechanism by which equilibrium (Maxwellian) is maintained. Additionally, computational cost for PIC solvers scales with both mesh size and particle count, while continuum solvers mainly scale with mesh size.

Multi-fluid equations are derived based on an assumption that the underlying velocity distribution function for each species is in equilibrium, the so called Maxwell-Boltzmann distribution which is the most probable distribution state. Each species can have a different distribution. Implicit in the derivation is an assumption that $f_{0,s}$ where s designates species, is not far from an equilibrium state due to high collisionality and therefore, intra-species $s-s$, $t-t$, collisional interactions have no net effect and only inter-species $(s-t)$ collisions modify other species. Equations for charged species are coupled together by Lorentz forces and collision source terms. At low densities, inter-species collisions are infrequent and can be neglected. In this case assuming no additional body forces, Lorentz forces alone account for species coupling. The system of conservation equations for each species s are;

$$\begin{aligned}
\frac{\partial \rho_s}{\partial t} + \nabla \cdot [\rho_s \mathbf{u}_s] &= \mathcal{C}_s^i + \mathcal{S}_s^i \\
\frac{\partial \rho_s \mathbf{u}_s}{\partial t} + \nabla \cdot [\rho_s \mathbf{u}_s \otimes \mathbf{u}_s + P_s \mathbf{I}] &= \frac{q_s \rho_s}{m_s} (\mathbf{E} + \mathbf{u}_s \times \mathbf{B}) + \mathcal{C}_s^i + \mathcal{S}_s^i \\
\frac{\partial \mathcal{E}_s}{\partial t} + \nabla \cdot [(\mathcal{E}_s + P_s) \mathbf{u}_s] &= \frac{q_s \rho_s}{m_s} \mathbf{u}_s \cdot \mathbf{E} + \mathcal{C}_s^i + \mathcal{S}_s^i
\end{aligned}$$

[3] On subsequent investigation, conducted during this project, we determined that the result in Section 3.5 of [40] was incorrect. A software bug led caused the correlations between the different particle in cell models to be estimated incorrectly and the improvement of the MF method used being over-stated. In the following we repeat that analysis using a different numerical model.

where $\mathbf{u}_s$ is the Cartesian velocity vector, ρ_s is the species mass density, m_s is the species particle mass, $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic induction vector, $\mathcal{E}_s$ is the total energy density and P_s is the isotropic scalar pressure. Collisional interactions, $\mathcal{C}_s^i$ are modeled as source terms and general body forces are represented by $\mathcal{S}_s^i$ where i is a vector of different types of collisions (e.g., scattering, ionization, recombination, charge exchange) and likewise for body forces. The total energy density is the sum of internal (e_s) and kinetic,

$$\rho_s \mathcal{E}_s = \rho_s e_s + \frac{1}{2}\rho_s ||\mathbf{u}_s||^2.$$

In addition, an equation of state relating isotropic pressure to internal energy and density is required to complete the system of equations.

In the Drekar multi-fluid plasma model, the Lorentz forces are computed by solving Maxwell's equations for the electric and magnetic fields, which introduce a time-step size restriction based on the speed of light. To circumvent this restriction, a fully implicit solver strategy is used. This allows time-step size to be determined by the hydrodynamic CFL which is much larger than the speed of light CFL which improves code efficiency. Increased fidelity both spatial and temporal, is achieved by a reduction in mesh spacing and time-step size. This defines a cost/accuracy trade off that must be balanced when running plasma simulations.

3.5.2. Fixed model ensemble

In this section we apply a non-hierarchical MF sampling method to estimate the expected QoI using a fixed ensemble of 5 models. Each of these models is obtained using the different mesh resolutions shown in Table 3-1; the models are ordered from highest to lowest fidelity. Specifically we use a pilot consisting of 100 samples of each model to estimate the correlation between models and predict the variance reduction that would be obtained using different estimators for varying computational budgets (total cost). The left panel of Figure 3-11 tabulates the correlation the five models.

Table 3-1. The numerical discretization and computational costs of each model within the ensemble. Cost is the median (over 100 runs) CPU time required to run one simulation.

Model	N_x	Cost (seconds)
f_0	1024	291
f_1	512	99
f_2	256	40
f_3	128	21
f_4	32	13

The middle panel of Figure 3-11, plots the variance of the MC estimator which uses only high-fidelity information against a set of increasing cost targets (in CPU seconds). We compare this variance with the variance of the ACV estimator (ACVMF) [23], with optimized sample allocation, obtained using an equivalent cost. We also plot the variance reduction obtained using a popular alternative multi-level MC (MLMC) method [22]. The ACV strategy is almost an order

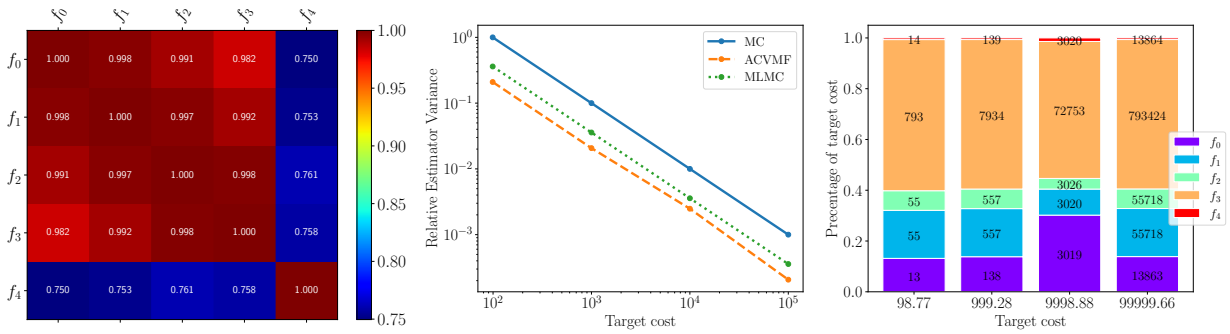


Figure 3-11. (Right) Correlations between the five plasma models. (Middle) Variance of different estimators as a function of total cost. (Right) Samples allocated by ACVMF to each model for increasing total cost.

of magnitude more accurate than the single-fidelity MC estimator. The non hierarchical ACV estimator is also significantly (2 times) more accurate than the hierarchical MLMC estimator even though the models a prior appears to admit a hierarchy.

The right of Figure 3-11 plots the percentage of the total cost due to evaluating each model within the ensemble against an increasing set of total costs. The numbers within the bars represent the number of samples allocated to each model. This plot shows that all models are being used to estimate the mean of the high-fidelity model. The exact allocations are optimized to maximize accuracy (minimize estimator variance) for the specified total cost. The large cost of the high-fidelity model f_0 is clear. Despite only 13 samples being used, when the total cost is 98.77, running this one simulation accounts for approximately 10% of the total cost. In contrast almost 800 evaluations of f_3 are used but these evaluations only allocate approximately 60% of the total cost. The number of high-fidelity evaluations increases as the allowable budget is increased, however the number of samples allocated to this model is reduced by using the other lower-fidelity models.

3.5.3. Model tuning

A model tuning study was defined using Dakota [1, 2] in order to identify the most effective values of t_{refine} (controlling the number of time-steps which increases with t_{refine}) and N_x (the number of mesh cells). Figure 3-12 shows the results of a parametric study using a discrete grid of .1:.1:1. for t_{refine} and 128:20:328 for N_x, where a clear minimum is evident for large t_{refine} and small N_x. This surface was then optimized using efficient global optimization (EGO) [41] in Dakota, treating t_{refine} and N_x as continuous within the previous bounds, to confirm convergence to the optimal hyper-parameters. While the underlying Gaussian process modeling was likely hindered by discrete truncation in N_x, EGO was still successful in locating the vicinity of the minimum at $(t_{refine}, N_x) = (0.9981, 128.4)$ using an initial GP build followed by 42 refinement iterations.

Within the specified hyper-parameter ranges, ACV performance varied by a significant factor of 25.6, indicating the strong benefits of having an automated capability to identify the most

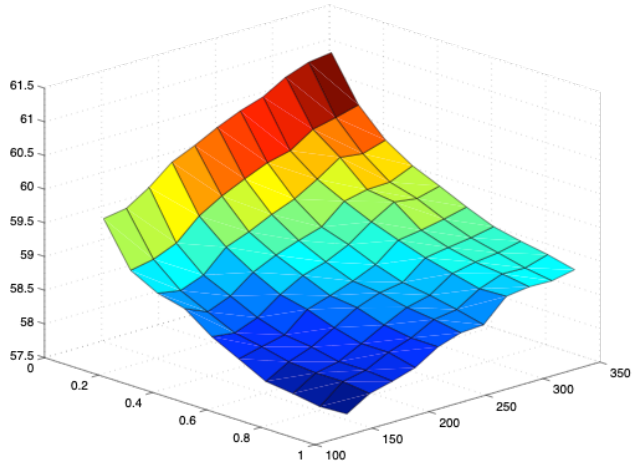


Figure 3-12. Surface plot of ACV performance over hyper-parameters t_{refine} and N_x. Each ordinate reflects the log of the optimized estimator variance (low values are better).

effective hyper-parameter settings. With the optimized hyper-parameter values, tuned ACV outperformed the corresponding single-fidelity analysis[4] by one order of magnitude (estimator variance reduced by a factor of 10.5). For the worst case hyper-parameters, ACV performs worse than single-fidelity MC by a factor of 2.45, largely due to the fact that the LF model becomes 2.33 times more expensive than the HF model, providing little value in the MF context.

[4]corresponding in the sense of having the same total computational budget (1000 equivalent HF simulations)

4. PROJECT OUTPUTS

This chapter summarizes this projects collaboration and outreach activities and its dissemination of research results. Specifically we detail all publications, presentations, organization of conference minisymposia, career development, team building and partnerships.

4.1. Publications

During this project we produced 6 accepted papers [25, 38, 9, 61, 11, 85], a preprint that will be submitted shortly [33], and four additional papers are in preparation and will be submitted early in FY23 [80, 83, 60, 71].

4.2. Presentations

During this project we presented a total of 22 conference and workshop presentations. The details of these presentations are as follows.

1. Multi-fidelity information fusion via network models for uncertainty quantification in aerospace dynamical systems. AIAA SciTech Forum. January 11-15 2021.
2. Uncertainty Quantification with Multifidelity Strategies Based on Models with Dissimilar Parameterizations. WCCM & ECCOMAS Congress. January 11-15 2021.
3. Multi-fidelity Data-Driven Networks for Bayesian Learning and Prediction. WCCM & ECCOMAS Congress. January 11-15 2021.
4. MFNets: Multi-Fidelity Data-Driven Networks for Data Analysis. SIAM Conference on Computational Science and Engineering. March 1-5 2021.
5. Recent Advances in Adaptive Refinement of Multifidelity Surrogates for UQ. SIAM Conference on Computational Science and Engineering. March 1-5 2021.
6. Estimating Approximate Control Variate Weights: with Applications in Importance Sampling and Rare Event Estimation. SIAM Conference on Computational Science and Engineering. March 1-5 2021.
7. Exploring Important Directions for Multifidelity Uncertainty Quantification by Basis Adaptation Method. SIAM Conference on Computational Science and Engineering. March 1-5 2021.

8. Multi-Fidelity Information Fusion. Computing and Information Sciences External Review Board Meeting. March 31 2021.

9. Adaptive resource allocation for surrogate modeling of systems comprised of multiple disciplines with varying fidelity. International Conference on Computational Methods for Coupled Problems in Science and Engineering. June 14-16 2021.

10. Multi-fidelity Machine Learning. Keynote seminar at the SNL Machine Learning and Deep Learning Workshop. July 19-22 2021.

11. Enhancing Multifidelity UQ with model tuning. 16th U.S. National Congress on Computational Mechanics (USNCCM16). 25-29 July, 2021.

12. Adaptive Basis for Multifidelity Uncertainty Quantification. 16th U.S. National Congress on Computational Mechanics (USNCCM16). 25-29 July, 2021.

13. Improving Digital Twins by Learning from a Fleet of Assets. Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering, and Technology. September 26-29 2021.

14. PyApprox: Approximation and Probabilistic Analysis of Data. SIAM Conference on Uncertainty Quantification. April 12-15 2022.

15. Bayesian Learning of Multifidelity Surrogate Networks: MCMC and Variational Inference Approaches. SIAM Conference on Uncertainty Quantification. April 12-15 2022.

16. Optimal Experimental Design for MFNets: Applications to Bayesian Learning of Multi-Fidelity Data-Driven Surrogates. SIAM Conference on Uncertainty Quantification. April 12-15 2022.

17. All-at-Once (and Bi-Level) Model Tuning for Multifidelity Sampling . SIAM Conference on Uncertainty Quantification. April 12-15 2022.

18. Improving Digital Twins by Learning from a Fleet of Assets. SIAM Conference on Uncertainty Quantification. April 12-15 2022.

19. Model Tuning for Multifidelity Sampling in Dakota. 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS). 5-9 June 2022.

20. Estimating Approximate Control Variate Weights: with Applications in Importance Sampling and Rare Event Simulation. 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS). 5-9 June 2022.

21. Embedded uncertainty estimation for data-driven surrogates to enable trustworthy ML for UQ. Machine Learning and Deep Learning Conference (MLDL2022). 25-28 July, 2022.

22. Uncertainty Quantification with Multi-Fidelity Networks. USACM Thematic Conference on Uncertainty Quantification for Machine Learning Integrated Physics Modeling (MLIP). 18-19 Aug, 2022.

4.3. Conference organization

The members of this project organized multi-part minisymposia on multi-fidelity methods at seven major international conferences. These sessions were organized to establish a community of researches focusing on multi-fidelity methods and increase the visibility and gain feedback on our teams' research. The minisymposia organized are listed below.

1. Multifidelity sampling approaches for Forward/Inverse UQ and Optimization Under Uncertainty. SIAM CSE. March 1-5 2021.
   https://www.siam.org/conferences/cm/conference/cse21
2. Uncertainty quantification for coupled multi-physics, multi-scale and multi-fidelity modeling. IX International Conference of Computational Methods for Coupled Problems In Science and Engineering. June 5-7. 2021.
   https://congress.cimne.com/coupled2021/frontal/default.asp
3. Data-enhanced modeling and uncertainty quantification of systems with multiple fidelities. US Congress on Computational Mechanics, to be held July 25-29th, 2021.
   http://16.usnccm.org/
4. Advanced multilevel and multifidelity UQ strategies: applications, generalized model hierarchies, and data-driven approaches. SIAM UQ. April 12-15 2022.
   https://www.siam.org/conferences/cm/conference/uq22
5. Multi-fidelity methods for uncertainty quantification and optimization. ECCOMAS. 5-9 June 2022. https://www.eccomas.org/2022/03/02/eccomas-2022-a-fully-in-person-congress/
6. Multi-fidelity methods for uncertainty quantification and optimization. SIAM CSE. 26 February - 3 March 2023
   https://www.siam.org/conferences/cm/conference/cse23
7. Machine learning and uncertainty quantification for coupled multi-physics, multi-scale and multi-fidelity modeling. X International Conference of Computational Methods for Coupled Problems In Science and Engineering. June 5-7. 2023.
   https://coupled2023.cimne.com/

4.4. Career development

When this project began in FY 21 our team comprised of two senior members of technical staff (G. Geraci and T. Smith), one principal member of technical staff (J. Jakeman) and one distinguished member of technical staff (M. Eldred). During the project G. Geraci was promoted to become a principal member of technical staff. We also engaged the services of an early career researcher (T. Seidl) to help with multi-fidelity Gaussian process surrogate modeling. We hosted bi-weekly research meetings to coordinate our LDRD research and our related multi-fidelity deployment activities. During these meetings we mentored two other early career staff (T. Portone and B. Reuter) involved in synergistic activities. Despite not having any experience with

multi-fidelity methods at the beginning of this project, all three early career staff members are now experts in multi-fidelity methods and will be able to support proliferation of these methods through out SNL.

4.5. Team building and partnerships

The success of this project was enhanced by several collaborations with new and existing partners. These collaborations are summarized here:

- Alex Gorodetsky (University of Michigan) - multi-fidelity MC and surrogate methods
- Geoffery Bomarito and Jim Warner (National Aeronautics and Space Administration) - model tuning for optimizing low-fidelity numerical discretization parameters for effective multi-fidelity MC estimation
- Lorenzo Tamellini (National Research Council Italy) Doug Allaire (Texas A&M)- multi-fidelity multi-physics surrogate modeling
- Akil Narayan (University of Utah) and Elisabeth Ullmann (Technical University of Munich) - Multi-arm bandit resource allocation
- Roger Ghanem (University of Southern California) - multi-fidelity methods for dissimilar parameterizations
- Sharlotte Kramer (SNL 1528) and Dan Bolintineanu (1516) - solid mechanics materials modeling
- Scott Roberts (SNL 1513) - thermal battery modeling
- Bert Debusschere (SNL 8351) - cyber systems modeling
- Matthew Hoffman (Los Alamos National Laboratories) - land-ice modeling

Collaborations formed or strengthened during this project led to two funded joint proposals with external institutions. Specifically, J.D Jakeman collaborated with Matthew Hoffman (Los Alamos National Laboratories) on a SCIDAC (BER/ASCR) proposal "*Framework for Antarctic System Science in E3SM*." J.D. Jakeman also collaborated with Andrew Christlieb (Michigan State University) on a Mathematical Multifaceted Integrated Capabilities Center proposal "*Center for Hierarchical and Robust Modeling of Non-Equilibrium Transport*."

5. PROJECT LEGACY

This project delivered next generation computational multi-fidelity tools that substantially reduce the computational and experimental resources needed for designing and assessing complex multi-physics/scale/component systems. The chapter discusses the ways this project contributed to the CIS LDRD investment area and SNL goals.

This project developed novel theoretically justified algorithms and associated software tools that advanced state of the art multi-fidelity modeling and uncertainty quantification. Our agile, mathematically-rigorous approach for multi-fidelity, multiscale, and multi-physics simulation for making accurate and cost-effective predictions using multiple information sources provides a fundamental advancement for tractably and accurately solving forward, inverse, calibration, and design optimization problems. Our advances have aided the development of trusted artificial intelligence by providing surrogate emulators that enable learning with limited high-fidelity data by fusing that data with lower-quality information that is more readily accessible. Moreover, our multi-fidelity sampling methods provide a means to utilize recently developed data-driven machine learning models of dynamical systems alongside of high-fidelity first principles baesd models to reduce cost of outerloop analyses.

The novel capabilities developed are broadly applicable to applications across numerous SNL mission spaces, from moving beyond nuclear weapons with Z-Next to developing inherently safe, resilient, modular nuclear weapon research themes. Each of these themes requires the certified prediction of complex systems using sparse high-fidelity data, but possess an ensemble of lower-fidelity information alternatives that can be leveraged to decrease the cost of forward, inverse, and design analyses.

The aforementioned advances and capabilities were disseminated via numerous internal and external outreach activities. The project team produced 11 peer-reviewed papers, gave 22 conference and workshop presentations at major international conferences and established SNL as a leader in multi-fidelity modeling. We also engaged in numerous cross center interactions between 1400, 1500, and 1600 and established or strengthened collaborations with numerous external institutions.

The collaborations we established has enabled rapid technology insertion in a number of mission areas. We summarize these activities below.

- The ASC V&V program used our algorithms implemented in Dakota to support a level 2 milestone demonstrating the utility of our new multi-fidelity uncertainty quantification of thermal battery models. Our model tuning algorithms were able to further reduce the cost of uncertainty quantification by an additional factor of 6, leading to a total factor of 143 relative to a single-fidelity MC strategy. Our success has led the V&V program to providing substantial support for futher deployment of multi-fidelity methods in FY23.

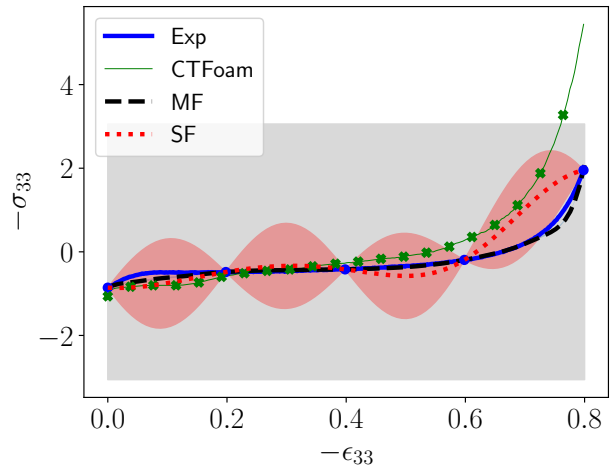


Figure 5-1. Multi-fidelity surrogates of material stress σ strain ε curves built using PyApprox. Exp (blue line represents) densely sampled experimental data for validation. CTFoam (green line) represents a singe fidelity surrogate of the simulation data (green crosses). SF (dashed red line) represents a single-fidelity Gaussian process surrogate that uses on the experimental data (blue dots) with uncertainty bands depicted by the red shaded region. The black dotted dash line represents the much more accurate multi-fidelity Gaussian process that uses both experimental and simulation data.

- Supported by the ASC AML program, the multi-fidelity Gaussian processes in PyApprox were used to fuse data from material deformation experiments with simulation data. We have been asked to support further deployment of this technology in FY23. Figure 5-1 demonstrates the improved accuracy of a multi-fidelity surrogate predicting stress-strain curves from limited experimental data and larger amounts of simulation data relative to surrogates that use only one of these data sources
- The SCIDAC project "*Probabilistic Sea Level Projections from Ice Sheet and Earth System Models*" used PyApprox to produce probabilistic projections of mass-loss from Humboldt Glacier Greenland. We determined that a Multi-fidelity sampling algorithm can reduce the computational cost of computing expected mass loss by a factor of 20, relative to a single-fidelity MC strategy.

To ensure the future adoption of the multi-fidelity novel algorithms developed by this project, we implemented these algorithms in two software packages PyApprox [33] and Dakota [1, 2]. PyApprox is intended to support rapid prototyping of new methods and provide extensive educational materials to increase the multi-fidelity methods user base. Dakota complements PyApprox by providing capabilities that target high performance computing applications.

PyApprox and the multi-fidelity software in Dakota were both submitted for consideration for the Society for Industrial and Applied Mathematics, James H. Wilkinson Prize for Numerical Software Prize in FY22. This award is yet to be announced. The capabilities of these packages are summarized below.

PyApprox PyApprox is a Python software package for machine learning, experimental design and multi-fidelity modeling. It provides implementation of the majority of existing multi-fidelity

surrogate and MC algorithms. Much of its capabilities have been supported by this project and the preceding one year LDRD project entitled "*Learning Hidden Structure in Multi-Fidelity Information Sources for Efficient Uncertainty Quantification.*" PyApprox can be downloaded at `https://sandialabs.github.io/pyapprox/index.html`. PyApprox and its dependencies can be easily installed by following the instructions at `https://sandialabs.github.io/pyapprox/install.html`. PyApprox comes with extensive documentation and is tested on every push of commits to the master branch of the Git repository. The status of installations and tests can be found at `https://github.com/sandialabs/pyapprox/actions`. Numerous tutorials on multi-fidelity modeling can be found at `https://sandialabs.github.io/pyapprox/auto_tutorials/index.html#multi-fidelity-methods`.

Dakota Dakota is a comprehensive C++ software package for iterative systems analysis [74, 1, 2], including uncertainty quantification, design optimization, model calibration, and sensitivity analysis. Dakota is open source and can be downloaded from `https://dakota.sandia.gov/`. It contains a variety of multilevel and MF algorithms for optimization and UQ, including surrogate-based optimization methods (hierarchical trust-region model management), sampling-based UQ methods (multilevel MC, MF MC, multilevel-control variate MC, and approximate control variate) and surrogate-based UQ approaches (polynomial chaos, stochastic collocation, and functional tensor train). Dakota was the implementation target for the model tuning effort funded by this LDRD, including bi-level and all-at-once formulations for solving the hyper-parameter optimization problem that identifies the most effective low-fidelity configurations.

Dakota provides a vehicle for production deployment to DOE mission applications on high-performance computing platforms, and a number of algorithms first prototyped in PyApprox can either be migrated to C++ or, through a new interoperability initiative, interfaced as an external Python library.

REFERENCES

[1] Brian M. Adams, Michael S. Eldred, Gianluca Geraci, Russell W. Hooper, John D. Jakeman, Kathryn A. Maupin, Jason A. Monschke, Ahmad A. Rushdi, Adam J. Stephens, Laura P. Swiler, Timothy M. Wildey, William J. Bohnhoff, Keith r. Dalbey, Mohamed S. Ebeida, John P. Eddy, Patricia D. Hough, Mohammad Khalil, Kenneth T. Hu, Elliott M. Ridgway, Dena M. Vigil, and Justin G. Winokur. Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.10 User's Manual. Technical report, 2014.

[2] Brian M. Adams, Michael S. Eldred, Gianluca Geraci, Russell W. Hooper, John D. Jakeman, Kathryn A. Maupin, Jason A. Monschke, Ahmad A. Rushdi, Adam J. Stephens, Laura P. Swiler, Timothy M. Wildey, William J. Bohnhoff, Keith r. Dalbey, Mohamed S. Ebeida, John P. Eddy, Patricia D. Hough, Mohammad Khalil, Kenneth T. Hu, Elliott M. Ridgway, Dena M. Vigil, and Justin G. Winokur. Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.10 Theory Man. Technical report, 2014.

[3] Brian M. Adams, Michael S. Eldred, Gianluca Geraci, Teresa Portone, Elliott M. Ridgway, J. Adam Stephens, and Timothy M. Wildey. Deployment of multifidelity uncertainty quantification for thermal battery assessment; Part I: Algorithms and single cell results. Technical Report SAND2022-11856, Sandia National Laboratories, Albuquerque, NM, September 2022.

[4] Brian M. Adams, Michael S. Eldred, Gianluca Geraci, Teresa Portone, Elliott M. Ridgway, J. Adam Stephens, and Timothy M. Wildey. Deployment of multifidelity uncertainty quantification for thermal battery assessment; Part II: Full battery results. Technical Report SAND2022-11851, Sandia National Laboratories, Albuquerque, NM, September 2022.

[5] Sergio Amaral, Douglas Allaire, and Karen Willcox. A decomposition-based approach to uncertainty analysis of feed-forward multicomponent systems. *International Journal for Numerical Methods in Engineering*, 100(13):982–1005, 2014.

[6] David Amsallem, Matthew J Zahr, and Charbel Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[7] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S Lazarov, and Ole Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[8] M. Arnst, R. Ghanem, E. Phipps, and J. Red-Horse. Measure transformation and efficient quadrature in reduced-dimensional stochastic modeling of coupled problems. *International Journal for Numerical Methods in Engineering*, 92(12):1044–1080, 2012.

[9] Geoffrey Bomarito, Gianluca Geraci, James Warner, Patrick Leser, William Leser, Michael S. Eldred, John Jakeman, and Alex Gorodetsky. *Improving Multi-Model Trajectory Simulation Estimators using Model Selection and Tuning*.

[10] G.F. Bomarito, P.E. Leser, J.E. Warner, and W.P. Leser. On the optimization of approximate control variates with parametrically defined estimators. *Journal of Computational Physics*, 451:110882, 2022.

[11] G.F Bomarito, J.E Warner, P.E Leser, W.P. Leser, M.S Eldred, J.D Jakeman, and A.A Gorodetsky. *Improving Multi-Model Trajectory Simulation Estimators using Model Selection and Tuning*. AIAA SciTech. American Institute of Aeronautics and Astronautics, 2023. Accepted.

[12] Kevin Carlberg, Sofia Guzzetti, Mohammad Khalil, and Khachik Sargsyan. The network uncertainty quantification method for propagating uncertainties in component-based systems. *arXiv*, 2020.

[13] Paul G. Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

[14] Paul G Constantine, Michael Emory, Johan Larsson, and Gianluca Iaccarino. Exploiting active subspaces to quantify uncertainty in the numerical simulation of the hyshot ii scramjet. *Journal of Computational Physics*, 302:1–20, 2015.

[15] M.M. Crockatt, S. Mabuza, J.N. Shadid, S. Conde, T.M. Smith, and R.P. Pawlowski. An implicit monolithic AFC stabilization method for the CG finite element discretization of the fully-ionized ideal multifluid electromagnetic plasma system. 464:1–28, 2022.

[16] Sam Friedman, Benson Isaac, Seyede Fatemeh Ghoreishi, and Douglas L Allaire. Efficient decoupling of multiphysics systems for uncertainty propagation. *Proceedings of the AIAA SciTech Forum*, 2018.

[17] G. Geraci and M.S. Eldred. Leveraging intrinsic principal directions for multifidelity uncertainty quantification. *SAND2018-10817*, 2018.

[18] G. Geraci, M.S. Eldred, A.A. Gorodetsky, and J.D. Jakeman. Towards leveraging active direction for efficient multifidelity uq strategies. In *6th European Conference on Computational Mechanics (ECCM 6)*, 2018.

[19] G. Geraci, X. Zeng, A.A. Gorodetsky, M.S. Eldred, J.D. Jakeman, and R. Ghanem. Uncertainty quantification with multifidelity strategies based on models with dissimilar parametrizations. In *14th WCCM & ECCOMAS Congress 2020*, 2021.

[20] Gianluca Geraci, Michael S. Eldred, Alex A. Gorodetsky, and John D. Jakeman. Recent advancements in Multilevel-Multifidelity techniques for forward UQ in the DARPA Sequoia project. In *AIAA Scitech 2019 Forum*, number January, 2019.

[21] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[22] Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[23] A.A. Gorodetsky, G. Geraci, M.S. Eldred, and J.D. Jakeman. A generalized approximate control variate framework for multifidelity uncertainty quantification. *Journal of Computational Physics*, 408:109257, 2020.

[24] A.A. Gorodetsky and J.D. Jakeman. Gradient-based optimization for regression in the functional tensor-train format. *Journal of Computational Physics*, 374:1219 – 1238, 2018.

[25] A.A. Gorodetsky, J.D. Jakeman, and G. Geraci. MFNets: data efficient all-at-once learning of multifidelity surrogates as directed networks of information sources. *Computational Mechanics*, 68(4):741–758, 2021.

[26] A.A. Gorodetsky, J.D. Jakeman, G. Geraci, and M. S. Eldred. MFNets: Multi-fidelity data-driven networks for Bayesian learning and prediction. *International Journal for Uncertainty Quantification*, 10(6):595–622, 2020.

[27] Alex A. Gorodetsky, Cosmin Safta, and John D. Jakeman. Reverse-mode differentiation in arbitrary tensor network format: with application to supervised learning. *Journal of Machine Learning Research*, 23(143):1–29, 2022.

[28] Loic Le Gratiet and Claire Cannamela. Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics*, 57(3):418–427, 2015.

[29] Loic Le Gratiet and Josselin Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5):365–386, 2014.

[30] A. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone. Multi-index stochastic collocation for random pdes. *Computer Methods in Applied Mechanics and Engineering*, 306:95 – 122, 2016.

[31] H. Harbrecht, J.D. Jakeman, and P. Zaspel. Cholesky-based experimental design for Gaussian process and kernel-based emulation and calibration. *Communications in Computational Physics*, 29(4):1152–1185, 2021.

[32] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

[33] J.D. Jakeman. Pyapprox: Enabling efficient model analysis., 8 2022.

[34] J.D. Jakeman, M.S. Eldred, G. Geraci, and A. Gorodetsky. Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 2019.

[35] J.D. Jakeman, A. Narayan, and T. Zhou. A generalized sampling and preconditioning scheme for sparse approximation of polynomial chaos expansions. *SIAM Journal on Scientific Computing*, 39(3):A1114–A1144, 2017.

[36] J.D. Jakeman and S.G. Roberts. Local and dimension adaptive stochastic collocation for uncertainty quantification. In J. Garcke and M. Griebel, editors, *Sparse Grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 181–203. Springer Berlin Heidelberg, 2013.

[37] John D. Jakeman, Fabian Franzelin, Akil Narayan, Michael Eldred, and Dirk Plfüger. Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351:643 – 666, 2019.

[38] John D. Jakeman, Sam Friedman, Michael S. Eldred, Lorenzo Tamellini, Alex A. Gorodetsky, and Doug Allaire. Adaptive experimental design for multi-fidelity surrogate modeling of multi-disciplinary systems. *International Journal for Numerical Methods in Engineering*, 123(12):2760–2790, 2022.

[39] John D. Jakeman, Drew P. Kouri, and J. Gabriel Huerta. Surrogate modeling for efficiently, accurately and conservatively estimating measures of risk. *Reliability Engineering & System Safety*, page 108280, 2022.

[40] John Davis Jakeman, Michael S. Eldred, Gianluca Geraci, Thomas M. Smith, and Alex A. Gorodetsky. Learning hidden structure in multi-fidelity information sources for efficient uncertainty quantification (ldrd 218317), 9 2020.

[41] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[42] 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.

[43] Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[44] Ksenia N. Kyzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1151–1171, 2018.

[45] Xuhui Meng, Hessam Babaee, and George Em Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.

[46] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.

[47] S. T. Miller, E. C. Cyr, J. N. Shadid, R. M. J. Kramer, E. G. Phillips, S. Conde, and R. P. Pawlowski. IMEX and exact sequence discretization of the multi-fluid plasma model. 397:108806, 2019.

[48] A. Mittal, X. Chen, C. H. Tong, and G. Iaccarino. A flexible uncertainty propagation framework for general multiphysics systems. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):218–243, 2016.

[49] Mohammad Motamed. A multi-fidelity neural network surrogate sampling method for uncertainty quantification. *International Journal for Uncertainty Quantification*, 10(4):315–332, 2020.

[50] A. Narayan, C. Gittelson, and D. Xiu. A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36(2):A495–A521, 2014.

[51] A. Narayan and J.D. Jakeman. Adaptive leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36(6):A2952–A2983, 2014.

[52] Leo WT Ng and Karen E Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for numerical methods in Engineering*, 100(10):746–772, 2014.

[53] F. Nobile, R. Tempone, and C.G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[54] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[55] Benjamin Peherstorfer, Tiangang Cui, Youssef Marzouk, and Karen Willcox. Multifidelity importance sampling. *Computer Methods in Applied Mechanics and Engineering*, 300:490–509, 2016.

[56] Benjamin. Peherstorfer, Karen. Willcox, and Max. Gunzburger. Optimal model management for multifidelity monte carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, 2016.

[57] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[58] Michael Penwarden, Shandian Zhe, Akil Narayan, and Robert M. Kirby. Multifidelity modeling for physics-informed neural networks (PINNs), 2021.

[59] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 473(2198), 2017.

[60] T. Pham, A. Gorodetsky, and J.D. Jakeman. Mfnets: Bayesian multi-fidelity surrogate modeling. In preparation.

[61] Trung Pham and Alex A. Gorodetsky. Ensemble approximate control variate estimators: Applications to multi-fidelity importance sampling, 2021.

[62] T. Qin, Z. Chen, J.D. Jakeman, and D. Xiu. Data-driven learning of nonautonomous systems. *SIAM Journal on Scientific Computing*, 43(3):A1607–A1624, 2021.

[63] T. Qin, Z. Chen, J.D. Jakeman, and D. Xiu. Deep learning of parameterized equations with applications to uncertainty quantification. *International Journal for Uncertainty Quantification*, 11(2):63–82, 2021.

[64] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335:736–746, 2017.

[65] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[66] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[67] Fonnesbeck C. Salvatier J, Wiecki TV. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[68] Francois Sanson, Olivier Le Maitre, and Pietro Marco Congedo. Systems of gaussian process models for directed chains of solvers. *Computer Methods in Applied Mechanics and Engineering*, 352:32 – 55, 2019.

[69] Daniel Schaden and Elisabeth Ullmann. On multilevel best linear unbiased estimators. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):601–635, 2020.

[70] Eric M. Schwartz, Eric T. Bradlow, and Peter S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

[71] T. Seidl and J.D. Jakeman. Building digital twins using data from a fleet of assets. In preparation.

[72] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, JUL 2008.

[73] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved July 13, 2022, from `http://www.sfu.ca/~ssurjano`.

[74] Laura P. Swiler, Michael S. Eldred, and Brian M. Adams. *Dakota: Bridging Advanced Scalable Uncertainty Quantification Algorithms with Production Deployment*, pages 1–43. Springer International Publishing, Cham, 2016.

[75] L.P. Swiler, M. Gulian, A.L. Frankel, C. Safta, and J.D. Jakeman. A survey of constrained Gaussian process regression: approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2):119–156, 2020.

[76] A. Teckentrup, P. Jantsch, C. Webster, and M. Gunzburger. A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1046–1074, 2015.

[77] Ramakrishna Tipireddy and Roger Ghanem. Basis adaptation in homogeneous chaos spaces. *Journal of Computational Physics*, 259:304–317, 2014.

[78] Kyle Washabaugh, David Amsallem, Matthew Zahr, and Charbel Farhat. *Nonlinear Model Reduction for CFD Problems Using Local Reduced-Order Bases*. 2012.

[79] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[80] Y. Xu, A. Narayan, J.D. Jakeman, A. Gorodetsky, and E. Ullman. Generalized bandit-learning multifidelity approximation. In preparation.

[81] Yiming Xu, Vahid Keshavarzzadeh, Robert M Kirby, and Akil Narayan. A bandit-learning approach to multifidelity approximation. *SIAM Journal on Scientific Computing*, 44(1):A150–A175, 2022.

[82] X. Zeng, G. Geraci, M.S. Eldred, and R. Ghanem. Exploring important directions for multifidelity uncertainty quantification by basis adaptation method. In *SIAM Conference on Computational Science and Engineering (CSE21)*, 2021.

[83] X. Zeng, G. Geraci, M.S. Eldred, J.D. Jakeman, A.A. Gorodetsky, and R. Ghanem. Multi-fidelity uncertainty quantification for models with dissimilar parameterizations. In preparation.

[84] X. Zeng, G. Geraci, M.S. Eldred, J.D. Jakeman, A.A. Gorodetsky, and R. Ghanem. Adaptive basis for multifidelity uncertainty quantification. In *16th U.S. National Congress on Computational Mechanics (USNCCM16)*, 2021.

[85] X. Zeng, G. Geraci, A. Gorodetsky, J. Jakeman, M. Eldred, and R. Ghanem. *Improving Bayesian networks multifidelity surrogate construction with basis adaptation*. AIAA SciTech. American Institute of Aeronautics and Astronautics, 2023. Accepted.

[86] Xiaoshu Zeng, John Red-Horse, and Roger Ghanem. Accelerated basis adaptation in homogeneous chaos spaces. *Computer Methods in Applied Mechanics and Engineering*, 386:114109, 2021.

[87] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	libref@sandia.gov

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.