SAND2019-1879C

Machine-Learning Error Models for Approximate Solutions to Parameterized Systems of Nonlinear Equations

Brian A. Freno
Kevin T. Carlberg
Sandia National Laboratories

SIAM Conference on Computational Science and Engineering
February 25, 2019

Outline

- Introduction
- Parameterized Systems of Nonlinear Equations
- Proposed Approach
- Numerical Experiments
- Summary

Outline

- Introduction
  - Motivation
  - Solution Approximations
  - Uncertainty Quantification
- Parameterized Systems of Nonlinear Equations
- Proposed Approach
- Numerical Experiments
- Summary

Motivation

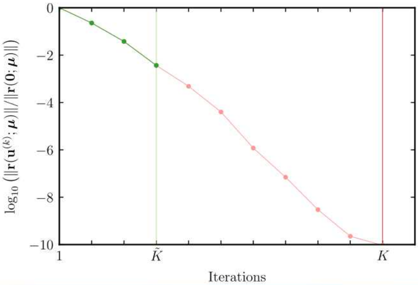
- Many-query problems can impose a formidable computational burden
- **Solution approximations** can exchange fidelity for speed

Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely end the iterative process
- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements
- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely end the iterative process
- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements
- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

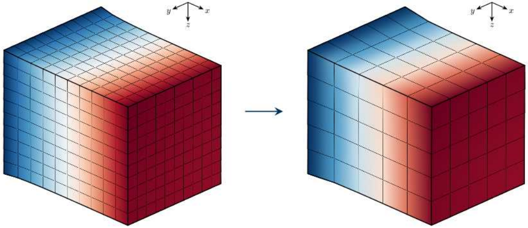


Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely end the iterative process

- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements

- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions



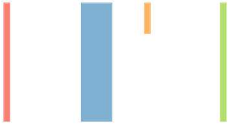
Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely end the iterative process

- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements

- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

$$\tilde{\mathbf{u}}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}$$



Uncertainty Quantification

- Solution approximations require less time than high-fidelity models but introduce an error (i.e., epistemic uncertainty)
- Ultimate task should account for **all sources of uncertainty**
- We quantify the uncertainty by
  1) **engineering features** informative of the error
     - cheaply computable
     - generated by approximate model
  2) applying **machine learning regression** techniques to construct a mapping from these features to a distribution of the error
- This work matures our previously developed capabilities:
  - Hand-selecting one feature and applying Gaussian process regression
    M. Drohmann and K. Carlberg (2015)
  - Modeling dynamical systems error using machine learning methods
    S. Trehan et al. (2017)

Outline

- Introduction
- Parameterized Systems of Nonlinear Equations
  - Overview
  - Approximate Solutions
  - Approaches for Error Quantification
- Proposed Approach
- Numerical Experiments
- Summary

Parameterized Systems of Nonlinear Equations

Parameterized systems of nonlinear equations

$$\mathbf{r}(\mathbf{u}(\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbf{0}$$

- $\mathbf{r} : \mathbb{R}^{N_{\mathbf{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{N_{\mathbf{u}}}$ residual, nonlinear in at least $\mathbf{u}(\boldsymbol{\mu})$
- $\mathbf{u} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{N_{\mathbf{u}}}$ state (solution vector)
- $\boldsymbol{\mu} \in \mathcal{D}$ parameters in parameter domain $\mathcal{D} \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$

Quantity of Interest

Scalar-valued quantity of interest

$$s(\boldsymbol{\mu}) := g(\mathbf{u}(\boldsymbol{\mu}))$$

- $s : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ quantity of interest
- $g : \mathbb{R}^{N_{\mathbf{u}}} \to \mathbb{R}$ quantity of interest functional

Approximate Solutions

- Computing the exact solution $\mathbf{u}(\boldsymbol{\mu})$ can be
  - prohibitively expensive (large $N_{\mathbf{u}}$)
  - unnecessary (inexact solutions suffice for optimization convergence)
- Such cases require an approximate solution $\tilde{\mathbf{u}} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{N_{\mathbf{u}}}$
- Approximate solution leads to approximated quantity of interest

$$\tilde{s}(\boldsymbol{\mu}) := g(\tilde{\mathbf{u}}(\boldsymbol{\mu})),$$

where $\tilde{s} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$

Approximate Solutions (continued)

We consider 3 approaches for computing approximate solutions:

1) Inexact solutions

2) Lower-fidelity models

3) Model reduction

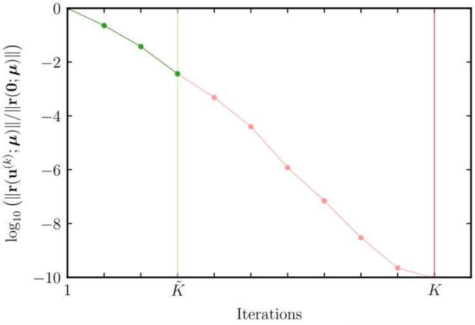
Inexact Solutions

- Iterative solution to nonlinear equations: sequence of approximations

$$\mathbf{u}^{(k)}, \quad k = 0, \ldots, K$$

- Approximate solution $\mathbf{u}^{(\tilde{K})}$ can be obtained after iteration $\tilde{K}$

$$\tilde{\mathbf{u}}(\boldsymbol{\mu}) = \mathbf{u}^{(\tilde{K})}$$

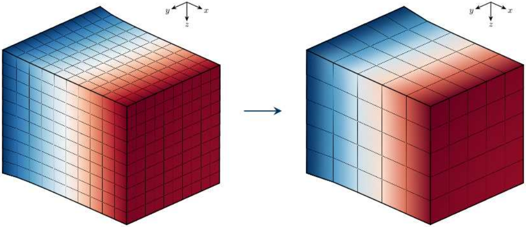


Lower-Fidelity Models

Fidelity reduction approaches

- Neglect physical phenomena
- Reduce spatial accuracy
  - Use lower-order finite differences or elements
  - Coarsen the mesh and prolongate (interpolate) the solution:

$$\tilde{\mathbf{u}} = \mathbf{p}(\mathbf{u}_{\mathrm{LF}}), \quad \mathbf{p} : \mathbb{R}^{N_{\mathbf{u}_{\mathrm{LF}}}} \to \mathbb{R}^{N_{\mathbf{u}}}$$



Model Reduction

Model reduction restricts approximate solution $\tilde{\mathbf{u}}$ to $m_{\mathbf{u}}$-dimensional affine trial subspace $\mathrm{Ran}(\boldsymbol{\Phi}_{\mathbf{u}}) + \bar{\mathbf{u}} \subseteq \mathbb{R}^{N_{\mathbf{u}}}$ with $m_{\mathbf{u}} \ll N_{\mathbf{u}}$:

$$\tilde{\mathbf{u}}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}$$

- $\boldsymbol{\Phi}_{\mathbf{u}} \in \mathbb{R}_{\star}^{N_{\mathbf{u}} \times m_{\mathbf{u}}}$ trial basis, computed using
  - proper orthogonal decomposition (POD)
  - the reduced-basis method
  - variants that employ gradient information
- $\hat{\mathbf{u}} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{m_{\mathbf{u}}}$ generalized coordinates of the approx. solution
- $\bar{\mathbf{u}} \in \mathbb{R}^{N_{\mathbf{u}}}$ prescribed reference state

Model Reduction (continued)

- $\mathbf{r}(\boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}; \boldsymbol{\mu}) = \mathbf{0}$ is overdetermined: $N_{\mathbf{u}}$ equations, $m_{\mathbf{u}}$ unknowns
- Second step projects residual onto an $m_{\mathbf{u}}$-dimensional test subspace $\mathrm{Ran}(\boldsymbol{\Psi}_{\mathbf{u}}) \subseteq \mathbb{R}^{N_{\mathbf{u}}}$:

$$\boldsymbol{\Psi}_{\mathbf{u}}^T\mathbf{r}(\boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}; \boldsymbol{\mu}) = \mathbf{0}$$

- $\boldsymbol{\Psi}_{\mathbf{u}} \in \mathbb{R}_\star^{N_{\mathbf{u}} \times m_{\mathbf{u}}}$ test basis, common choices include
  - Galerkin projection: $\boldsymbol{\Psi}_{\mathbf{u}} = \boldsymbol{\Phi}_{\mathbf{u}}$
  - Least-squares Petrov–Galerkin projection: $\boldsymbol{\Psi}_{\mathbf{u}} = \dfrac{\partial \mathbf{r}}{\partial \mathbf{v}}(\boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}; \boldsymbol{\mu})\boldsymbol{\Phi}_{\mathbf{u}}$

Approaches for Error Quantification

- Regardless of approach, it is essential to quantify error incurred by employing approximate solution $\tilde{\mathbf{u}}$ in lieu of exact solution $\mathbf{u}$

- Existing approaches include
  - Data-fit mapping between parameters and the error
    - Inspired by multifidelity design optimization
  - Reduced-Order Model Error Surrogates (ROMES) method

    M. Drohmann and K. Carlberg, 2015
    - Quantity-of-interest error approximated using dual-weighted residuals
    - Normed state-space error approx. using residual norm and error bounds

- This work focuses on quantifying two errors:

  1) Error in quantity of interest: $\delta_s(\boldsymbol{\mu}) := s(\boldsymbol{\mu}) - \tilde{s}(\boldsymbol{\mu})$
  2) Normed state-space error: $\delta_{\mathbf{u}}(\boldsymbol{\mu}) := \|\mathbf{e}(\boldsymbol{\mu})\|_2$, where $\mathbf{e}(\boldsymbol{\mu}) := \mathbf{u}(\boldsymbol{\mu}) - \tilde{\mathbf{u}}(\boldsymbol{\mu})$

State-Space Error

The residual can be approximated about the approximate solution $\tilde{\mathbf{u}}$:

$$\mathbf{r}(\mathbf{u}(\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbf{0} = \mathbf{r}(\boldsymbol{\mu}) + \mathbf{J}(\boldsymbol{\mu})\mathbf{e}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

and rearranged to approximate the state-space error:

$$\mathbf{e}(\boldsymbol{\mu}) = -\mathbf{J}(\boldsymbol{\mu})^{-1}\mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

- $\mathbf{r}(\boldsymbol{\mu}) := \mathbf{r}(\tilde{\mathbf{u}}(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{N_\mathbf{u}}$ residual from approximate solution
- $\mathbf{J}(\boldsymbol{\mu}) := \dfrac{\partial \mathbf{r}}{\partial \mathbf{v}}(\tilde{\mathbf{u}}(\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{N_\mathbf{u}\times N_\mathbf{u}}$ Jacobian of residual at $\tilde{\mathbf{u}}(\boldsymbol{\mu})$

Error in the Quantity of Interest

The quantity of interest also can be approximated:

$$s(\boldsymbol{\mu}) = \tilde{s}(\boldsymbol{\mu}) + \frac{\partial g}{\partial \mathbf{u}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{e}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

and combined with the state-space error approximation to yield

$$\delta_s(\boldsymbol{\mu}) = \underbrace{-\frac{\partial g}{\partial \mathbf{v}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{J}(\boldsymbol{\mu})^{-1}}_{\mathbf{y}(\boldsymbol{\mu})^T}\mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

- $\mathbf{y}(\boldsymbol{\mu})$ is the dual or adjoint
- dual-weighted residual d is weighted sum of residual elements:

$$d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T\mathbf{r}(\boldsymbol{\mu}) = \sum_{i=1}^{N_{\mathbf{u}}} y_i(\boldsymbol{\mu})r_i(\boldsymbol{\mu})$$

Drawbacks to using the Dual-Weighted Residual

- **Computational Cost:** requires solving $N_{\mathbf{u}}$ linear equations
- **Implementation:** requires Jacobian – not always available
- **Uncertainty Quantification:** low-bias error estimate not assured

Nonetheless, structure provides insight into quantity-of-interest error

Normed State-Space Error

- Residual-based bounds commonly used *a posteriori* to quantify $\delta_{\mathbf{u}}(\boldsymbol{\mu})$

  A. Buffa et al., 2012; M. A. Grepl and A. T. Patera, 2005; G. Rozza et al., 2008

- Assuming Lipschitz continuity for the residual $\mathbf{r}(\cdot;\boldsymbol{\mu})$, then

$$\frac{\|\mathbf{r}(\boldsymbol{\mu})\|}{\beta(\boldsymbol{\mu})} \leq \delta_{\mathbf{u}}(\boldsymbol{\mu}) \leq \frac{\|\mathbf{r}(\boldsymbol{\mu})\|}{\alpha(\boldsymbol{\mu})},$$

  where α and β are Lipschitz constants

- Drawbacks to using error bounds
  - **Sharpness:** upper/lower bounds can overpredict/underpredict actual error by several orders of magnitude
  - **Implementation:** difficult to compute true Lipschitz constants
  - **Uncertainty Quantification:** do not produce statistical distribution over $\delta_{\mathbf{u}}(\boldsymbol{\mu})$ – cannot quantify epistemic uncertainty

Outline

- Introduction
- Parameterized Systems of Nonlinear Equations
- Proposed Approach
  - Overview
  - Feature Engineering
  - Regression-Function Approximation
  - Training and Test Data
- Numerical Experiments
- Summary

Overview

- We aim to construct statistical models of
  - quantity-of-interest error δ_s
  - normed state-space error $\delta_{\mathbf{u}}$
- We apply high-dimensional regression methods from machine learning
- We use a larger number of inexpensive error indicators, resulting in less costly, more accurate error models

Error Model

- Assume there exist $N_{\mathbf{x}}$ *error indicators* or *features* $\mathbf{x}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\mathbf{x}}}$
  - available from solution approximation
  - cheaply computable
  - informative of the error $\delta(\boldsymbol{\mu}) \in \mathbb{R}$
- We model the nondeterministic mapping $\mathbf{x}(\boldsymbol{\mu}) \mapsto \delta(\boldsymbol{\mu})$

$$\delta(\boldsymbol{\mu}) = f(\mathbf{x}(\boldsymbol{\mu})) + \epsilon(\mathbf{x}(\boldsymbol{\mu}))$$

- f: deterministic regression function
- ϵ: stochastic noise
  - Mean-zero random variable
  - Accounts for irreducible error due to omitted explanatory variables
  - Epistemic – additional features can enable zero noise

Regression Model

- Regression function defines conditional expectation of error given the features:

$$\mathrm{E}[\delta(\boldsymbol{\mu}) \,|\, \mathbf{x}(\boldsymbol{\mu})] = f(\mathbf{x}(\boldsymbol{\mu}))$$

- We construct models of
  - deterministic regression function $\hat{f}(\approx f)$
  - stochastic noise $\hat{\epsilon}(\approx \epsilon)$,

  which yield a statistical model for the approximate-solution error

$$\hat{\delta}(\boldsymbol{\mu}) = \hat{f}(\mathbf{x}(\boldsymbol{\mu})) + \hat{\epsilon}(\mathbf{x}(\boldsymbol{\mu}))$$

Regression Model Objectives

- **Low Cost:** Should employ cheaply computable features $\mathbf{x}$
- **Low Noise Variance:** Should exhibit low noise variance, reduce epistemic uncertainty introduced by approximate solution
- **Generalize:** Empirical distributions of $\hat{\delta}$ and δ should be close on test set **not** used to train model – should not overfit on training data

Regression Model Construction Steps

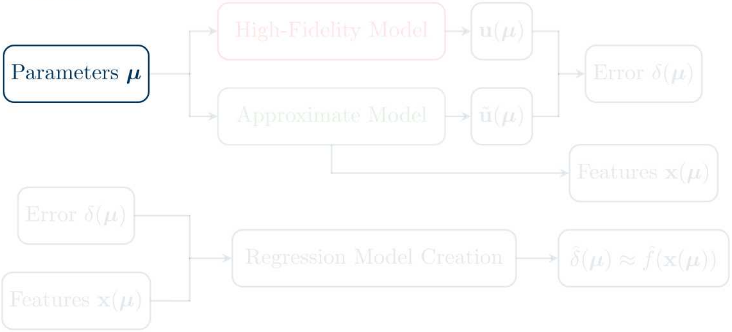
1) **Feature engineering**
   - Cheaply computable features $\mathbf{x}$ from approximate model
   - Informative of the error – construct low-noise-variance model
   - Low dimensional (small $N_{\mathbf{x}}$) such that less training data is needed

2) **Regression-function approximation**
   - Construct $\hat{f}$ using regression methods from machine learning
   - Approximate mapping from features $\mathbf{x}$ to error δ using a training set

3) **Noise approximation**
   - Mean-zero, constant-variance Gaussian random variable: $\hat{\epsilon} \sim \mathcal{N}(0, \hat{\sigma}^2)$
   - $\hat{\sigma}^2$ is sample variance of regression-model noise on a test set (mean squared error on test set)

Summary

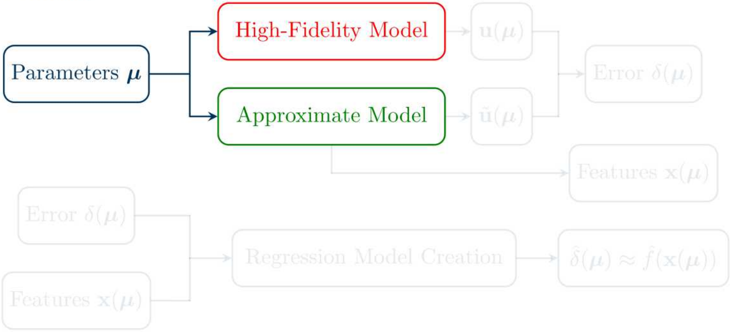
Training



Application

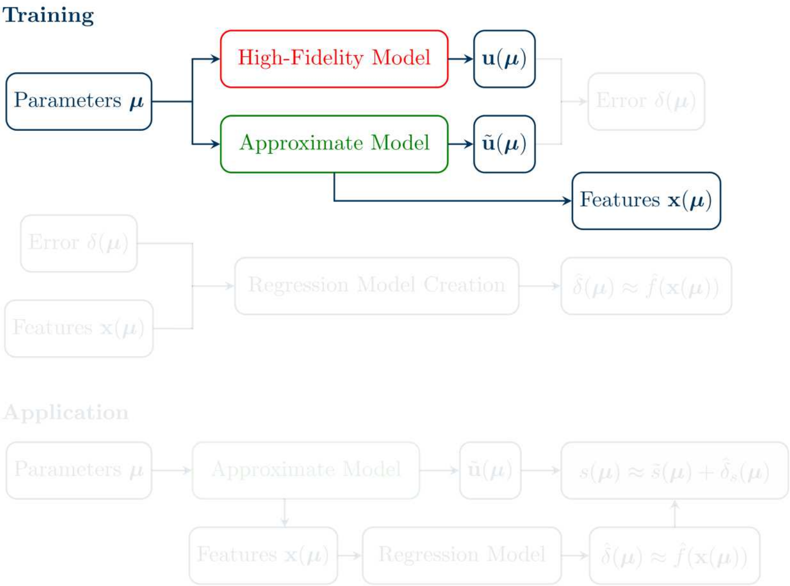
Summary

Training



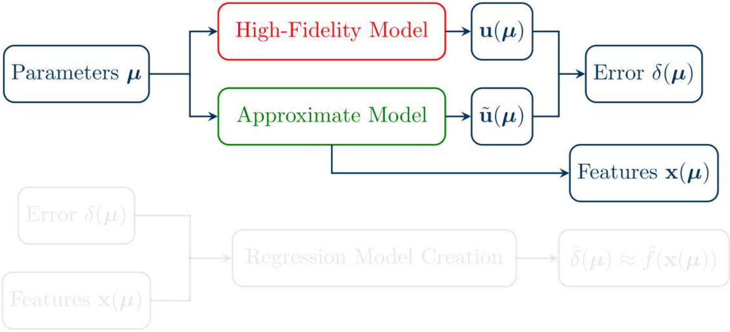
Application

Summary


Training
High-Fidelity Model
$\mathbf{u}(\boldsymbol{\mu})$
Parameters $\boldsymbol{\mu}$
Error $\delta(\boldsymbol{\mu})$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Error $\delta(\boldsymbol{\mu})$
Regression Model Creation
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$
Features $\mathbf{x}(\boldsymbol{\mu})$
Application
Parameters $\boldsymbol{\mu}$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
$s(\boldsymbol{\mu}) \approx \tilde{s}(\boldsymbol{\mu}) + \hat{\delta}_s(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Regression Model
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$

Summary

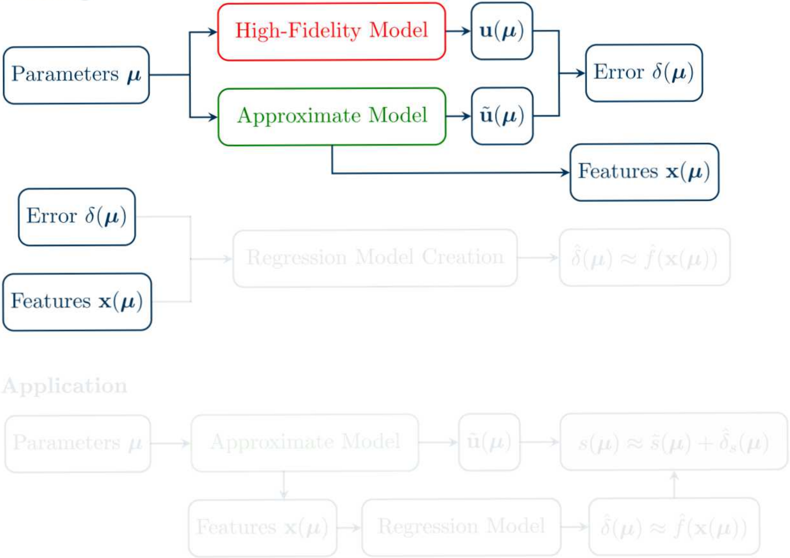
Training



Application

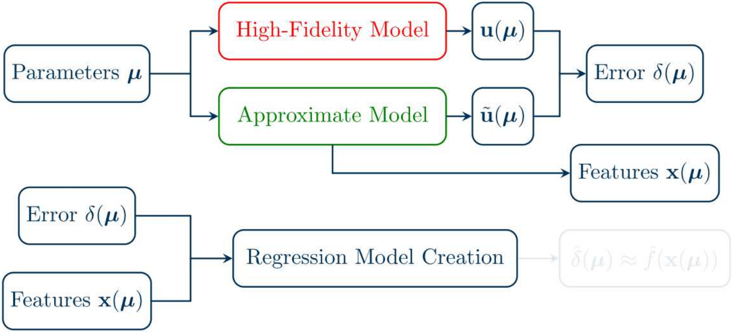
Summary

Training



Summary

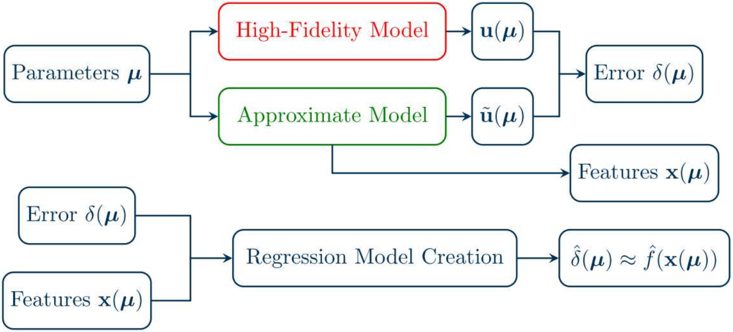
Training



Application

Summary

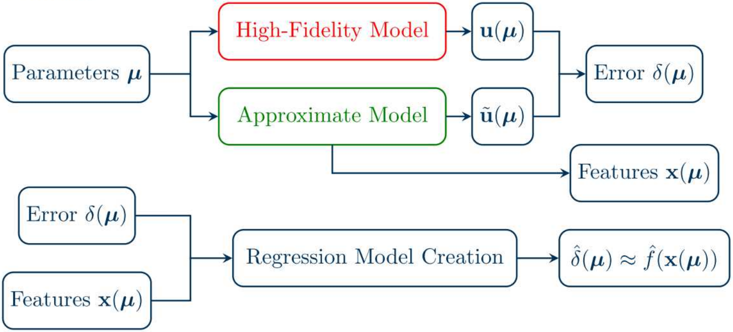
Training



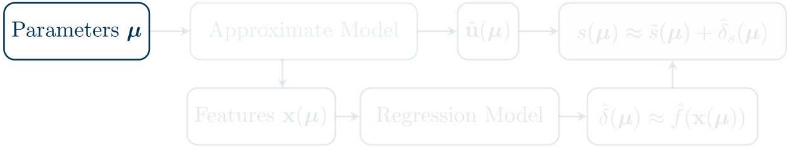
Application

Summary

Training

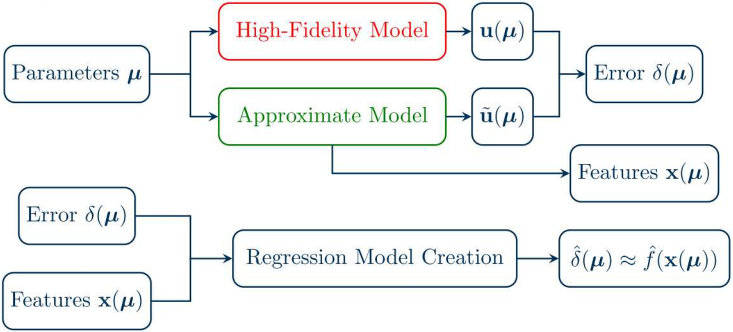


Application

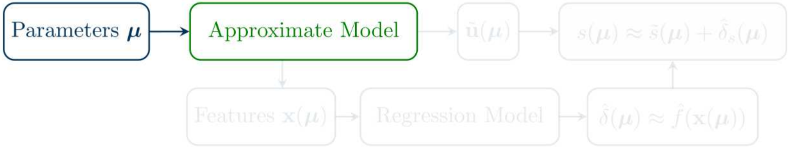


Summary

Training

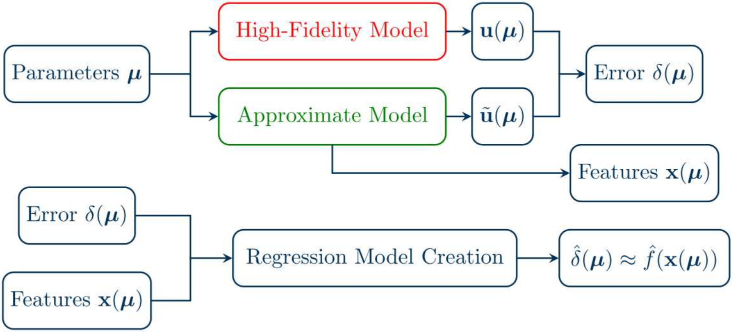


Application

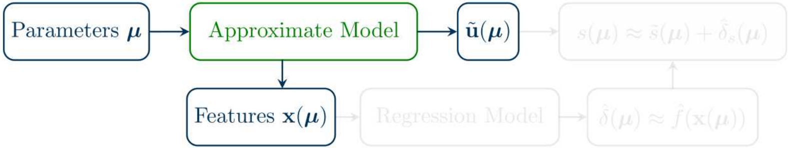


Summary

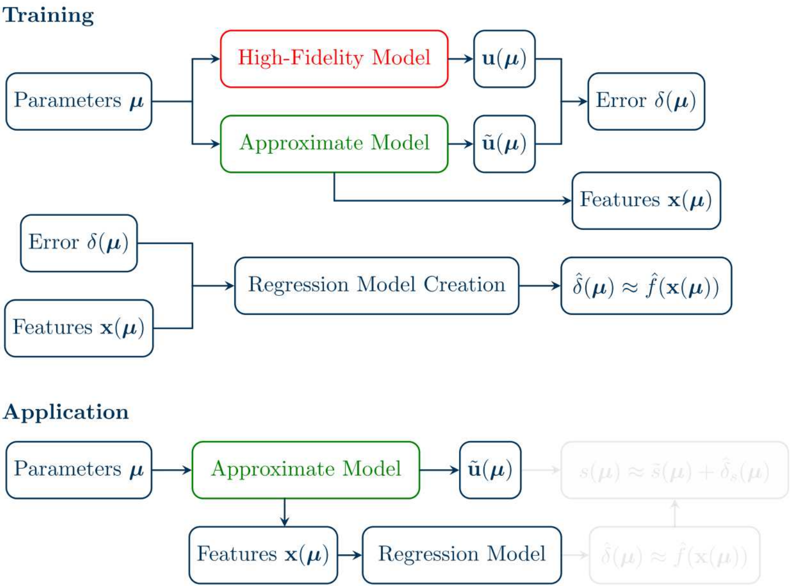
Training



Application

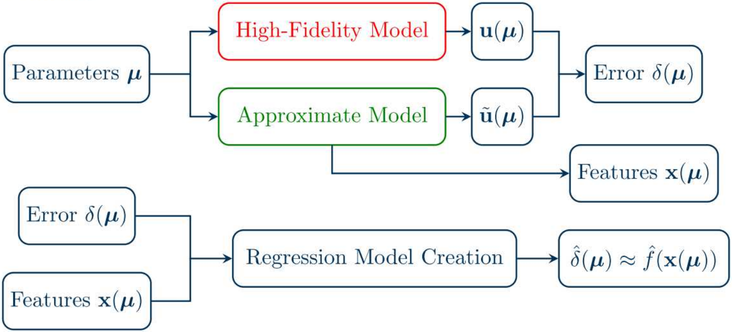


Summary

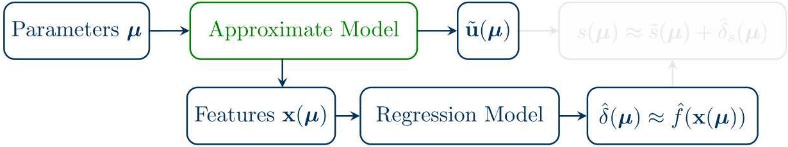

Training
High-Fidelity Model
$\mathbf{u}(\boldsymbol{\mu})$
Parameters $\boldsymbol{\mu}$
Error $\delta(\boldsymbol{\mu})$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Error $\delta(\boldsymbol{\mu})$
Regression Model Creation
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$
Features $\mathbf{x}(\boldsymbol{\mu})$
Application
Parameters $\boldsymbol{\mu}$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
$s(\boldsymbol{\mu}) \approx \tilde{s}(\boldsymbol{\mu}) + \hat{\delta}_s(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Regression Model
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$

Summary

Training

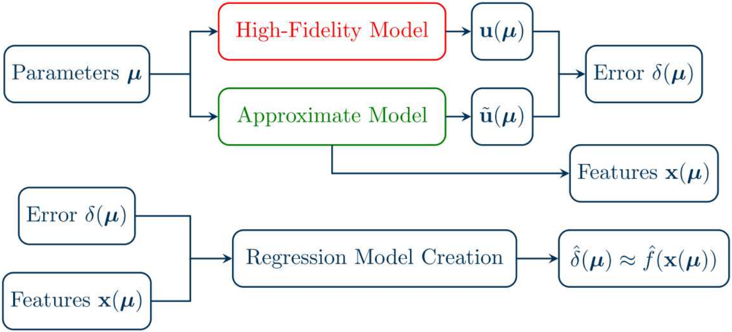


Application

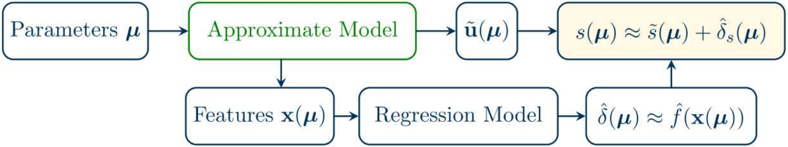


Summary

Training



Application



Feature Engineering: Parameters

$$\mathbf{x}(\boldsymbol{\mu}) = \boldsymbol{\mu}$$

- The mapping $\boldsymbol{\mu} \mapsto \delta(\boldsymbol{\mu})$ is deterministic, but often complex
  - Can be oscillatory for ROMs since $\delta(\boldsymbol{\mu}) \approx 0$ when $\boldsymbol{\mu} \in \mathcal{D}_{\text{Train}}^{\text{ROM}}$
- Could yield zero noise variance if
  - Large amount of training data
  - High-capacity regression model
- Typically low-quality features
- Inspired by 'multifidelity correction' methods for optimization

  Alexandrov et al., 2001; Gano et al., 2005; Eldred et al., 2004

Feature Engineering: Dual-Weighted Residual

$$\mathbf{x}(\boldsymbol{\mu}) = d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$$

- First-order approximation of QoI error $\delta_s(\boldsymbol{\mu})$
- Small number ($N_{\mathbf{x}} = 1$) of high-quality features
- High computational cost and significant implementation effort
- ROMES method uses approximation for dual-weighted residual

  M. Drohmann and K. Carlberg, 2015

Feature Engineering: Parameters and Residual (Approximations)

$$\mathbf{x}(\boldsymbol{\mu}) = [\boldsymbol{\mu}; \mathbf{r}(\boldsymbol{\mu})]$$

- DWR is weighted sum of residual vector elements $d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$
- Avoids implementation and costs associated with dual vector $\mathbf{y}(\boldsymbol{\mu})$
- Large number ($N_{\mathbf{x}} = N_{\boldsymbol{\mu}} + N_{\mathbf{u}}$) of low-quality features
- Approaches to reduce number of features and improve quality
  - Use $m_{\mathbf{r}} \ll N_{\mathbf{u}}$ principal component coefficients: $\hat{\mathbf{r}}(\boldsymbol{\mu})$
  - Sample $n_{\mathbf{r}} \ll N_{\mathbf{u}}$ elements of residual: $\mathbf{P}\mathbf{r}(\boldsymbol{\mu})$, where $\mathbf{P} \in \{0, 1\}^{n_{\mathbf{r}} \times N_{\mathbf{u}}}$
  - Use $m_{\mathbf{r}} \ll N_{\mathbf{u}}$ gappy principal component coefficients: $\hat{\mathbf{r}}_{\mathrm{g}}(\boldsymbol{\mu})$

Feature Engineering: Residual Norm with/without Parameters

$$\mathbf{x}(\boldsymbol{\mu}) = \|\mathbf{r}(\boldsymbol{\mu})\| \qquad \text{or} \qquad \mathbf{x}(\boldsymbol{\mu}) = [\boldsymbol{\mu}; \|\mathbf{r}(\boldsymbol{\mu})\|]$$

- DWR can be bounded using the Cauchy–Schwarz inequality:

$$|d(\boldsymbol{\mu})| \leq \|\mathbf{y}(\boldsymbol{\mu})\|_2 \|\mathbf{r}(\boldsymbol{\mu})\|$$

- Normed state-space error $\delta_{\mathbf{u}}(\boldsymbol{\mu})$ can be bounded:

  M. Drohmann and K. Carlberg, 2015

$$\frac{\|\mathbf{r}(\boldsymbol{\mu})\|}{\beta(\boldsymbol{\mu})} \leq \delta_{\mathbf{u}}(\boldsymbol{\mu}) \leq \frac{\|\mathbf{r}(\boldsymbol{\mu})\|}{\alpha(\boldsymbol{\mu})}$$

- $\boldsymbol{\mu}$ can be omitted ($\mathbf{x}(\boldsymbol{\mu}) = \|\mathbf{r}(\boldsymbol{\mu})\|$) if
  - $\boldsymbol{\mu}$ is not indicative of error
  - $N_{\boldsymbol{\mu}}$ is too large relative to training data
- Requires computing entire residual vector $\mathbf{r}(\boldsymbol{\mu})$
- Small number of potentially low-quality features

Regression-Function Approximation

We consider several different regression models

- Ordinary least squares (OLS)
  - Linear (OLS: Linear)
  - Quadratic expansion of features (OLS: Quadratic)
- Support vector regression (SVR)
  - Linear kernel (SVR: Linear)
  - Gaussian (radial basis function) kernel (SVR: RBF)
- Random forest (RF)
- k-nearest neighbors (k-NN)
- Artificial neural network (ANN)

Training and Test Data

Training Data

- Set of parameter training instances $\mathcal{D}_{\text{train}} \subset \mathcal{D}$
- Train regression models from high-fidelity and approx. solutions
  - Cross-validated to tune regression-model hyper-parameters
- Used to compute principal components of residuals

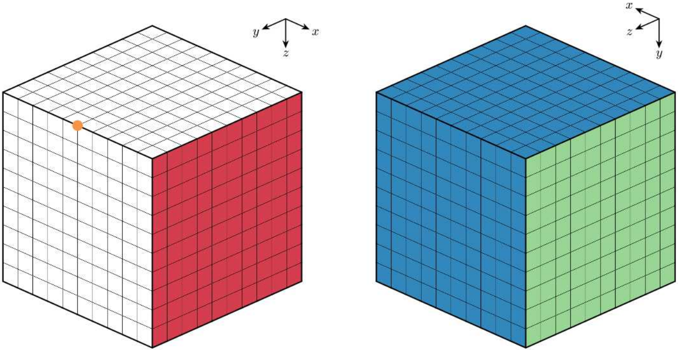
Test Data

- Set of parameter test instances $\mathcal{D}_{\text{test}} \subset \mathcal{D}$ **not** used for training ($\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset$)
- Used to assess regression models and quantify stochastic noise

Outline

- Introduction
- Parameterized Systems of Nonlinear Equations
- Proposed Approach
- Numerical Experiments
  - Cube: Reduced-Order Modeling
  - PCAP: Reduced-Order Modeling
  - Burgers' Equation: Inexact Solutions and Coarse Solution Prolongation
- Summary

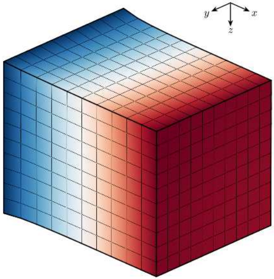
Cube: Reduced-Order Modeling

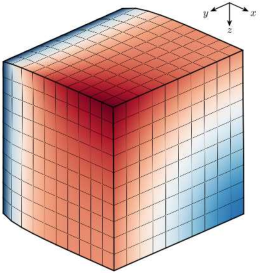


- Applied traction (Neumann boundary condition)
- Planar constraint (Dirichlet boundary condition)
- Complete constraint (Dirichlet boundary condition)
- Node of interest

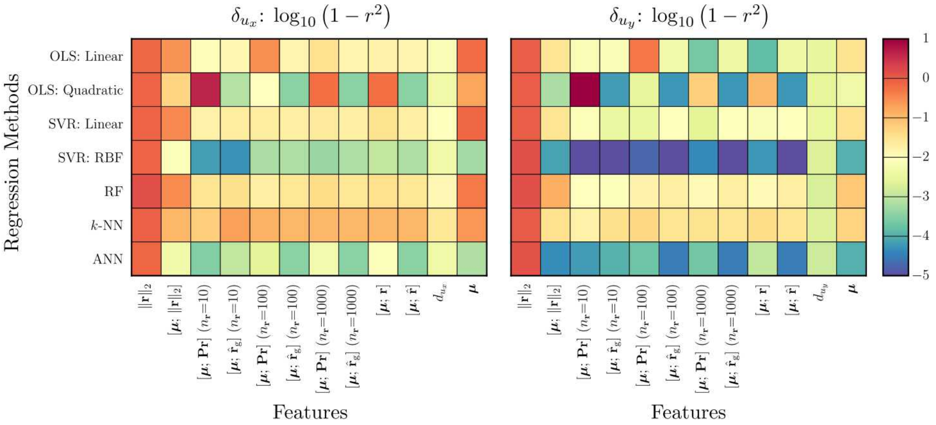
Cube: Overview

- $N_{\mathbf{u}} = 3410$ – deliberately small to compute $d(\boldsymbol{\mu})$ and use $\mathbf{r}(\boldsymbol{\mu})$
- $N_{\boldsymbol{\mu}} = 3$ parameters: $\boldsymbol{\mu} = [E;\, \nu;\, t]$
  - $E \in [75, 125]$ GPa, $\nu \in [0.20, 0.35]$, $t \in [40, 60]$ GPa
- 8 HF runs → up to $m_{\mathbf{u}} = 8$ ROM basis functions (2 used – 99.49%)



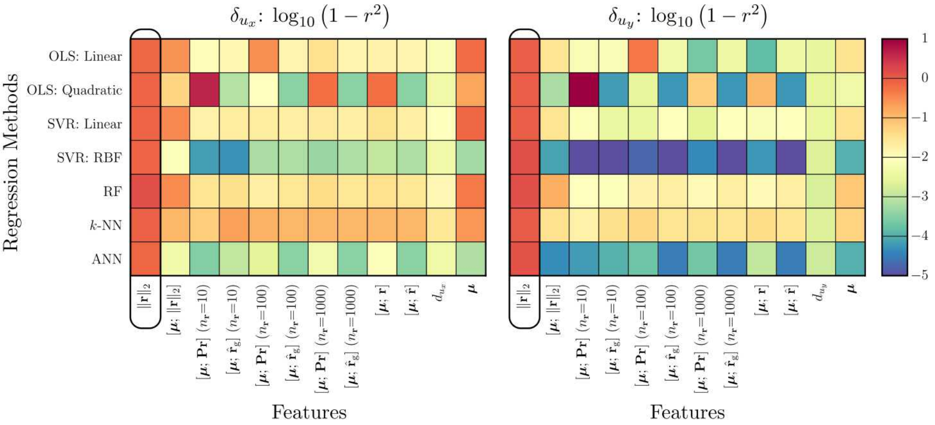


Cube: Variance Unexplained for QoI Error Prediction



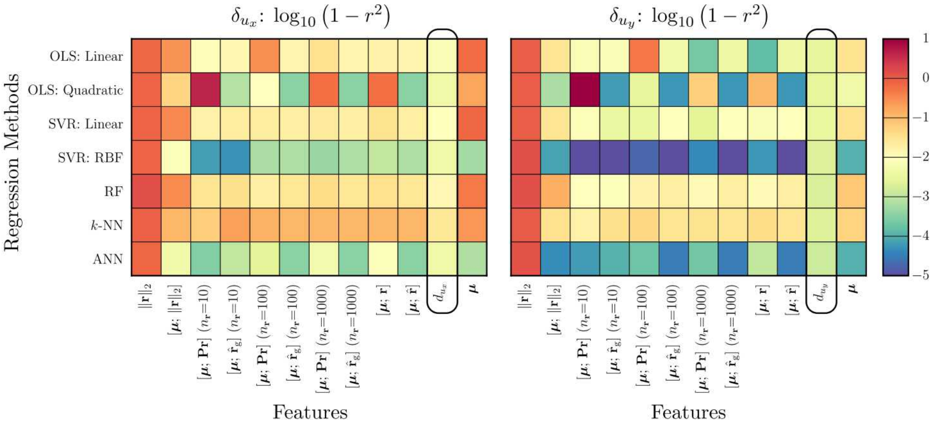
Sandia National Laboratories

Cube: Variance Unexplained for QoI Error Prediction



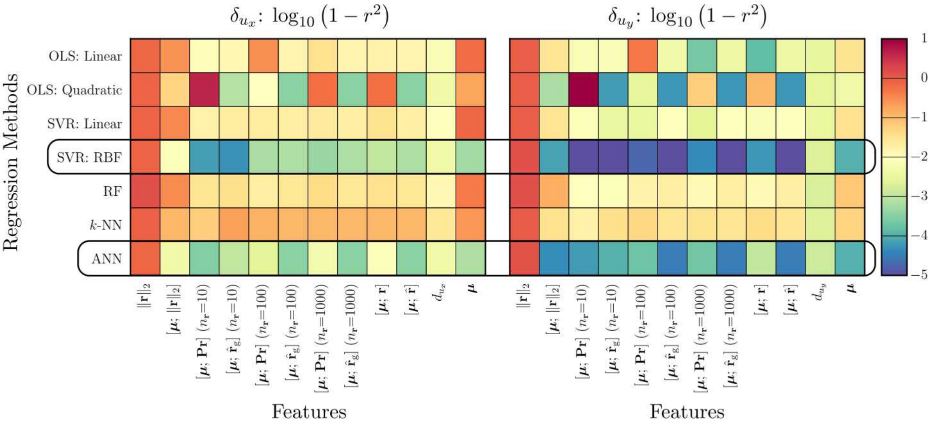
- $\|\mathbf{r}\|$ yields highest variance unexplained

Cube: Variance Unexplained for QoI Error Prediction



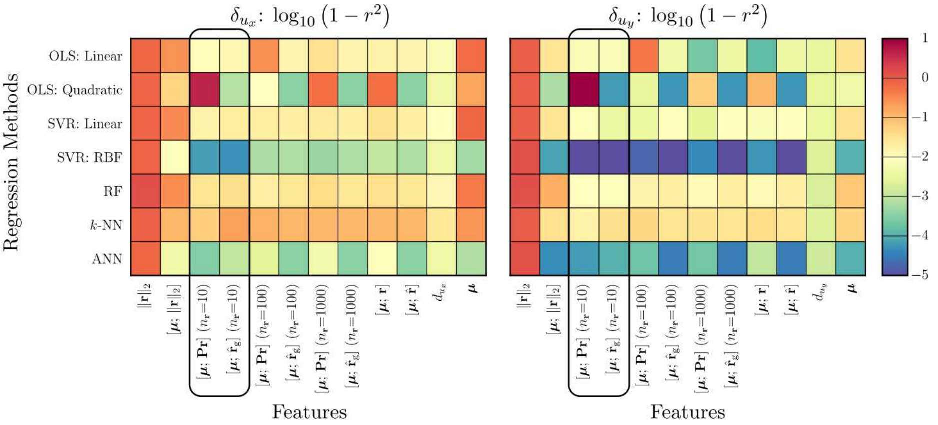
- $\|\mathbf{r}\|$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly

Cube: Variance Unexplained for QoI Error Prediction



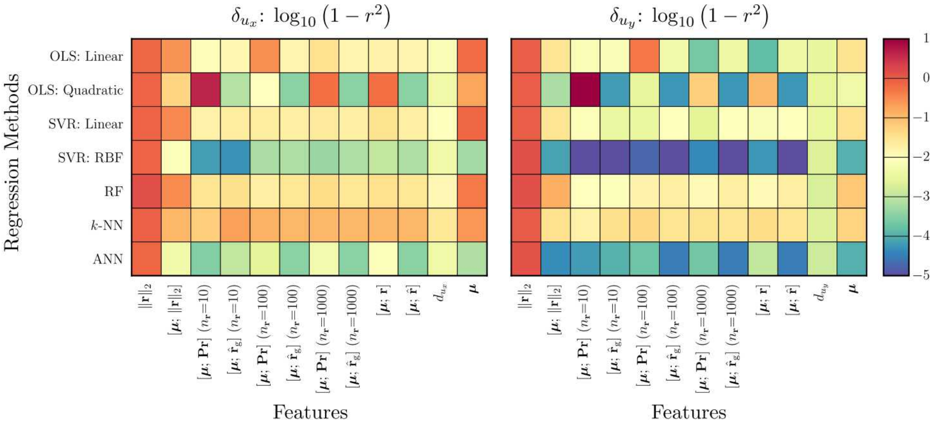
- $\|\mathbf{r}\|$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly
- SVR: RBF and ANN yield lowest variance unexplained

Cube: Variance Unexplained for QoI Error Prediction



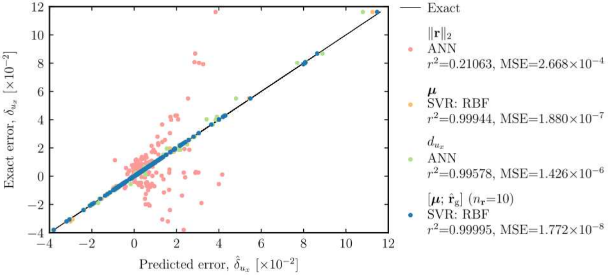
- $\|\mathbf{r}\|$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 3410$)

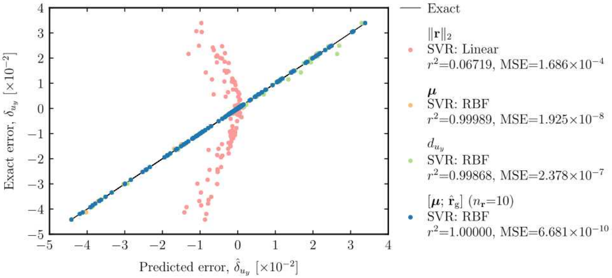
Cube: Variance Unexplained for QoI Error Prediction



- $\|\mathbf{r}\|$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 3410$)

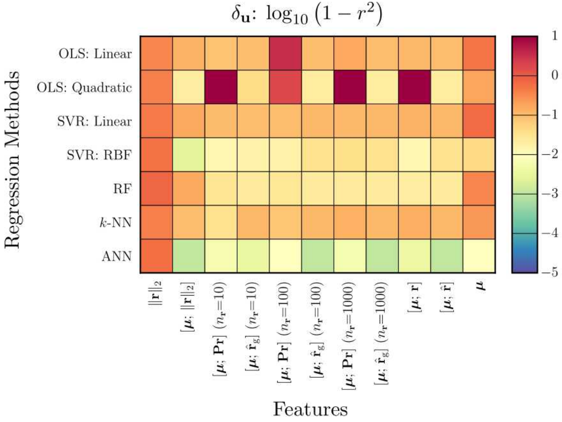
Cube: QoI Error Predictions



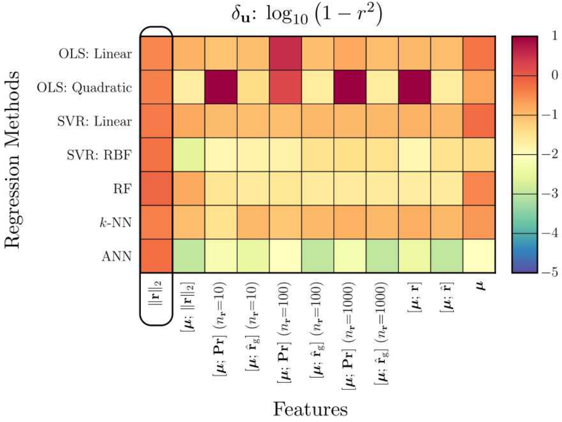


- Our method beats previous state-of-the-art methods with $r^2 > 0.9999$ in both cases

Cube: Variance Unexplained for Normed State-Space Error Prediction

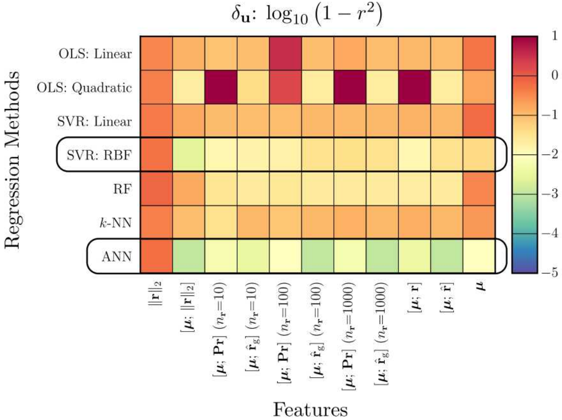


Cube: Variance Unexplained for Normed State-Space Error Prediction



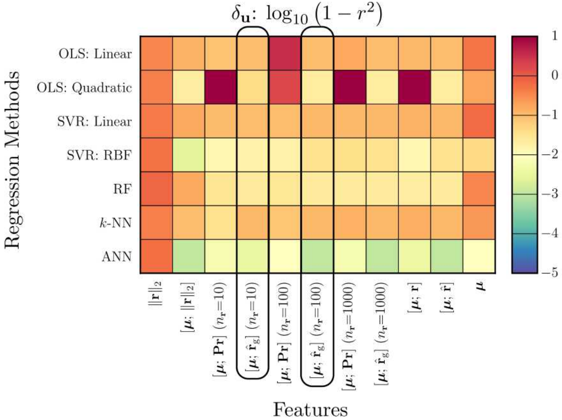
- $\|\mathbf{r}\|$ yields highest variance unexplained

Cube: Variance Unexplained for Normed State-Space Error Prediction



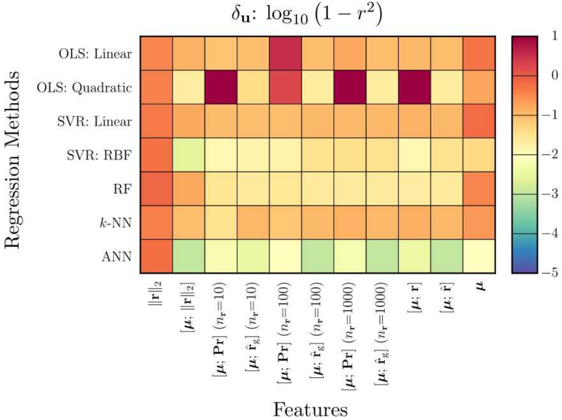
- $\|\mathbf{r}\|$ yields highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained

Cube: Variance Unexplained for Normed State-Space Error Prediction



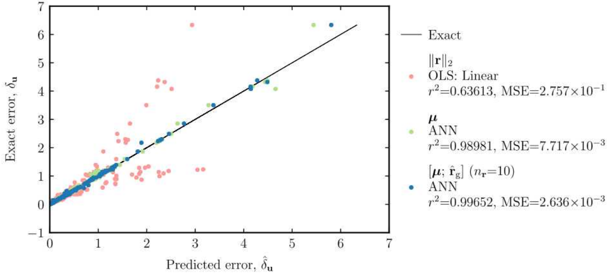
- $\|\mathbf{r}\|$ yields highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_{\mathrm{g}}]$ yields low variance unexplained with few samples (compared to $N_{\mathbf{u}} = 3410$)

Cube: Variance Unexplained for Normed State-Space Error Prediction



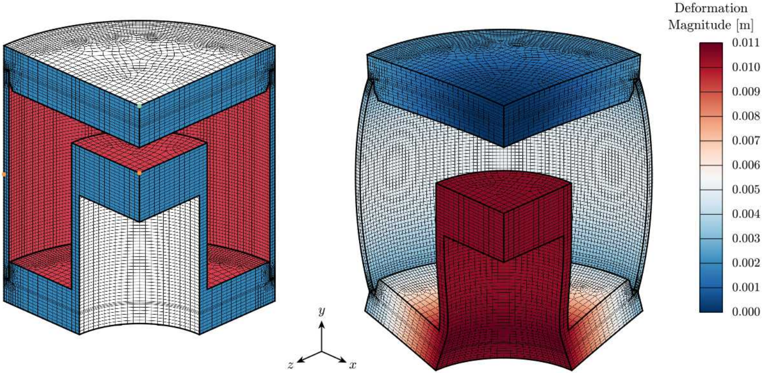
- $\|\mathbf{r}\|$ yields highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_{\mathrm{g}}]$ yields low variance unexplained with few samples (compared to $N_{\mathbf{u}} = 3410$)

Cube: Normed State-Space Error Predictions



- Our method beats previous state-of-the-art methods with $r^2 > 0.996$

Predictive Capability Assessment Project: Reduced-Order Modeling

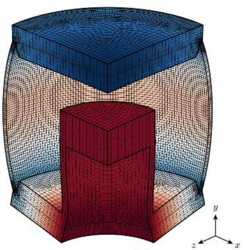


- Applied pressure (Neumann boundary condition)
- Planar constraint (Dirichlet boundary condition)
- Complete constraint (Dirichlet boundary condition)
- Nodes of interest

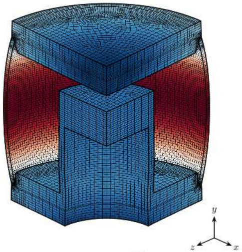
PCAP: Overview

- $N_{\mathbf{u}} = 274,954$ for quarter of domain
- $N_{\boldsymbol{\mu}} = 3$ parameters: $\boldsymbol{\mu} = [E; \nu; p]$
  - $E \in [50, 100]$ GPa, $\nu \in [0.20, 0.35]$, $p \in [6, 10]$ GPa
- 8 HF runs $\rightarrow$ up to $m_{\mathbf{u}} = 8$ ROM basis functions (5 used – 99.90%)
- 30 parameter training instances for regression model

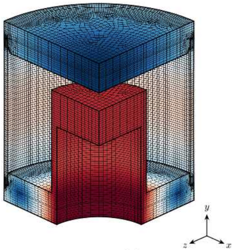
PCAP: Basis Functions



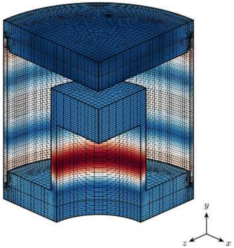
1: 85.03%



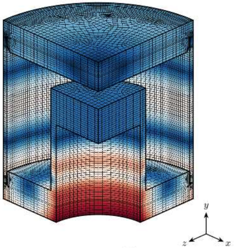
2: 95.69%



3: 99.35%

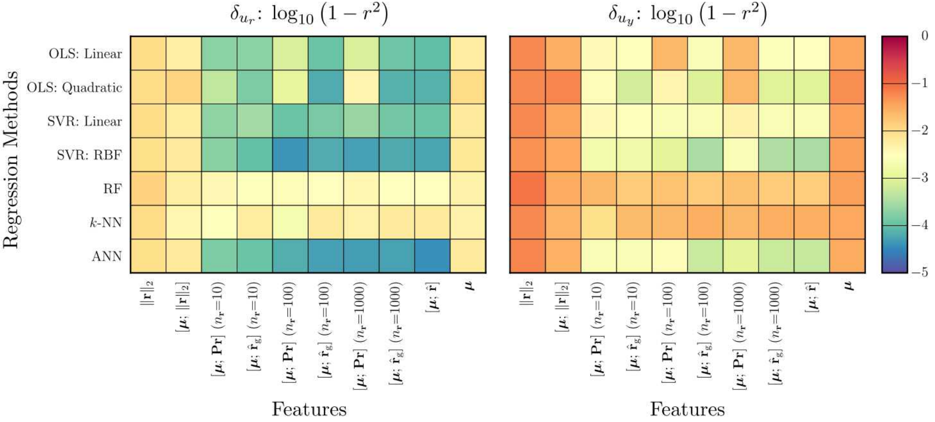


4: 99.77%



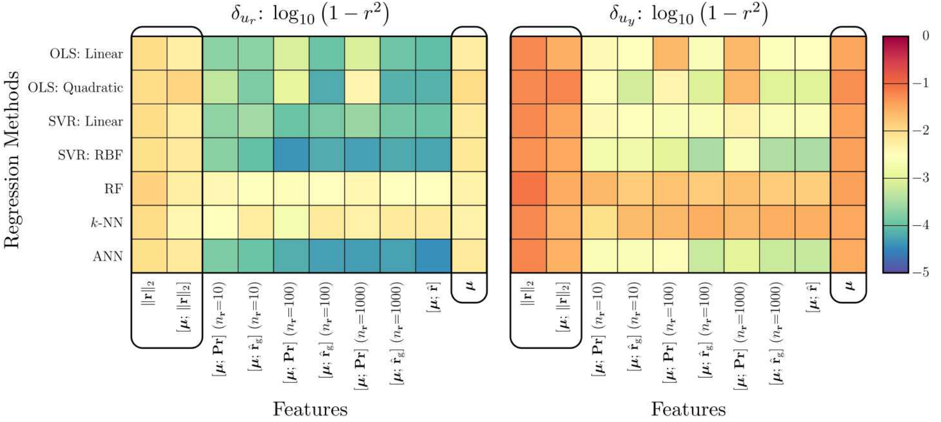
5: 99.90%

PCAP: Variance Unexplained for QoI Error Prediction



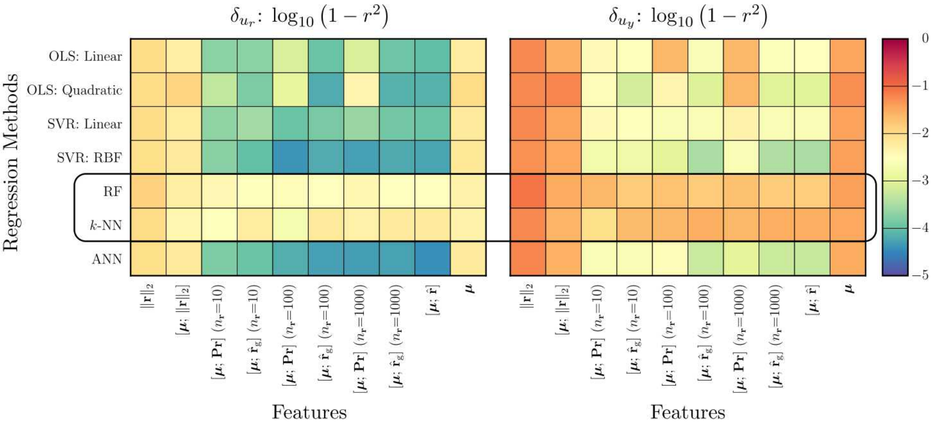
Sandia National Laboratories

PCAP: Variance Unexplained for QoI Error Prediction



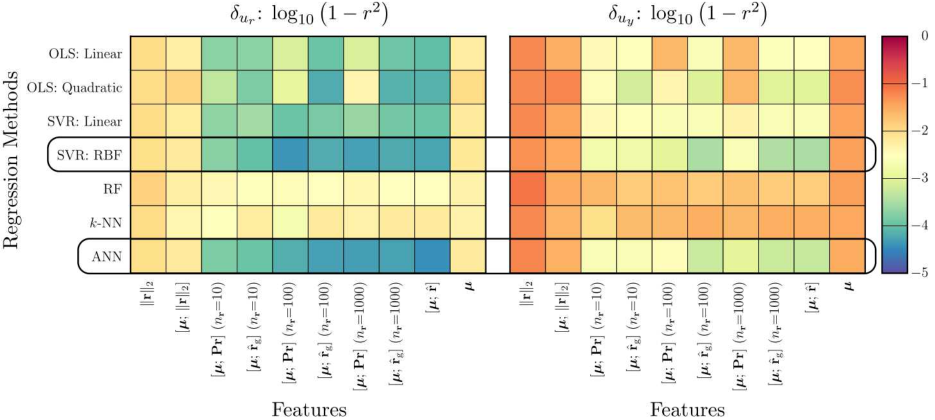
- $\|\mathbf{r}\|$, $[\boldsymbol{\mu}; \|\mathbf{r}\|]$, and $\boldsymbol{\mu}$ yield highest variance unexplained

PCAP: Variance Unexplained for QoI Error Prediction



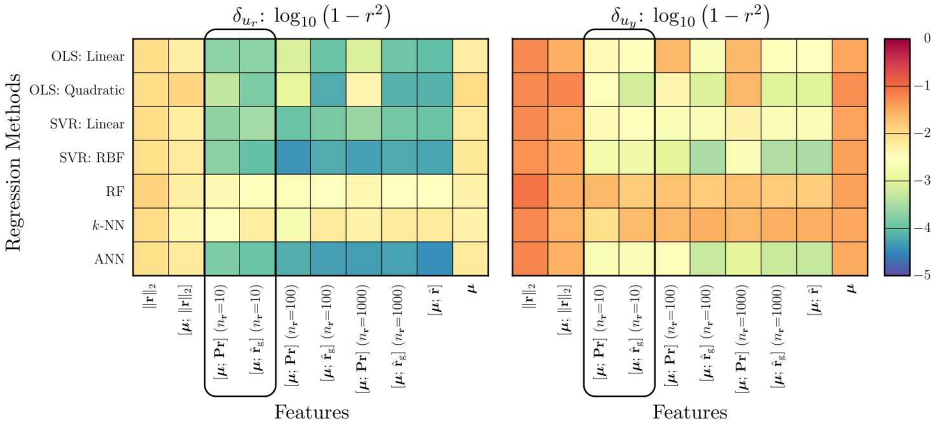
- $\|\mathbf{r}\|$, $[\boldsymbol{\mu}; \|\mathbf{r}\|]$, and $\boldsymbol{\mu}$ yield highest variance unexplained
- RF and k-NN yield highest variance unexplained

PCAP: Variance Unexplained for QoI Error Prediction



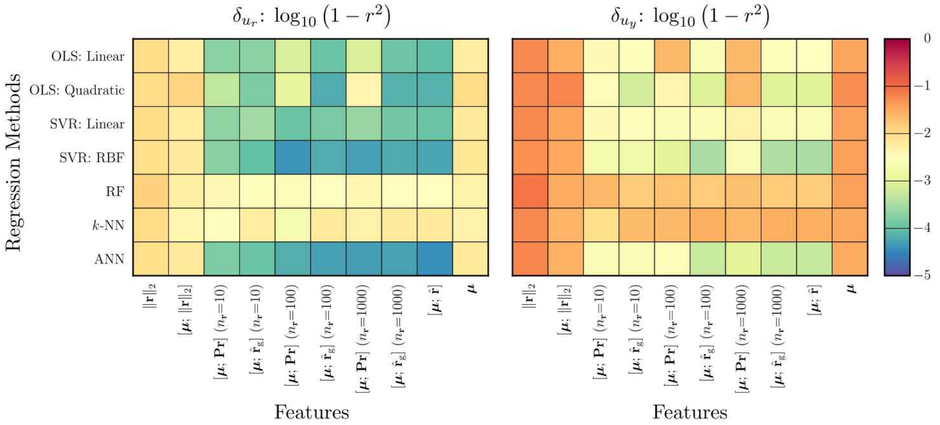
- $\|\mathbf{r}\|$, $[\boldsymbol{\mu}; \|\mathbf{r}\|]$, and $\boldsymbol{\mu}$ yield highest variance unexplained
- RF and k-NN yield highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained

PCAP: Variance Unexplained for QoI Error Prediction



- $\|\mathbf{r}\|$, $[\boldsymbol{\mu}; \|\mathbf{r}\|]$, and $\boldsymbol{\mu}$ yield highest variance unexplained
- RF and k-NN yield highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\text{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 274,954$)

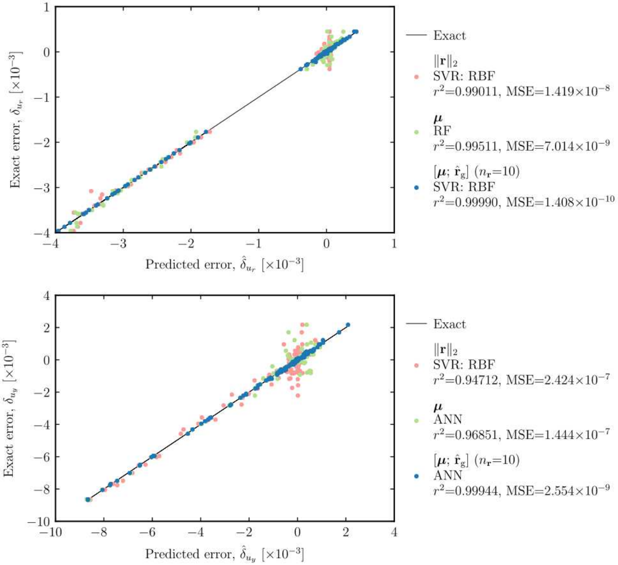
PCAP: Variance Unexplained for QoI Error Prediction



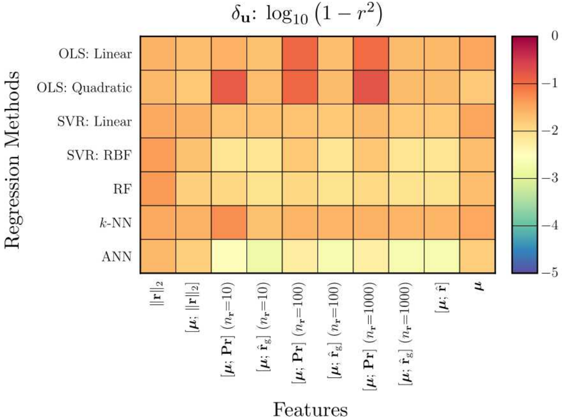
- $\|\mathbf{r}\|$, $[\boldsymbol{\mu}; \|\mathbf{r}\|]$, and $\boldsymbol{\mu}$ yield highest variance unexplained
- RF and k-NN yield highest variance unexplained
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_g]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_{\mathbf{u}} = 274,954$)

PCAP: QoI Error Predictions

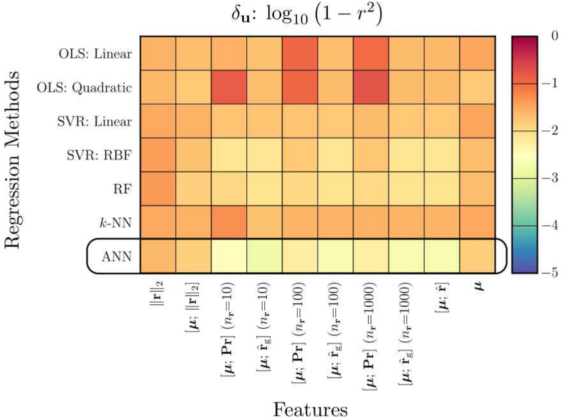
- Our method beats previous state-of-the-art methods with $r^2 > 0.9994$ in both cases



PCAP: Variance Unexplained for Normed State-Space Error Prediction

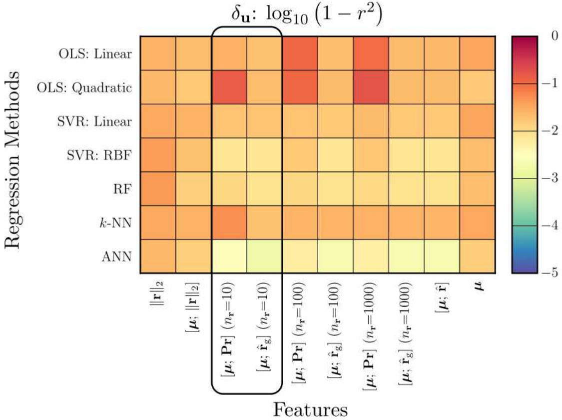


PCAP: Variance Unexplained for Normed State-Space Error Prediction



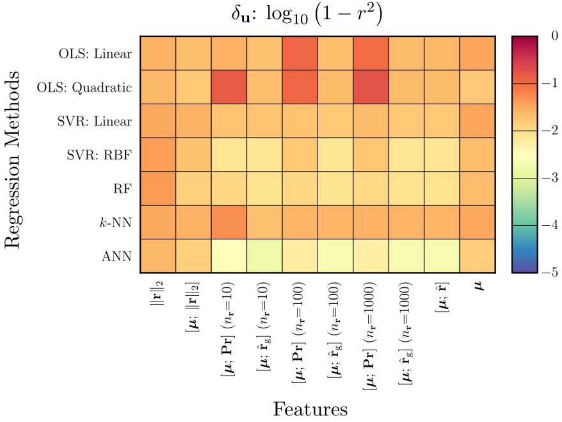
- ANN yields lowest variance unexplained

PCAP: Variance Unexplained for Normed State-Space Error Prediction



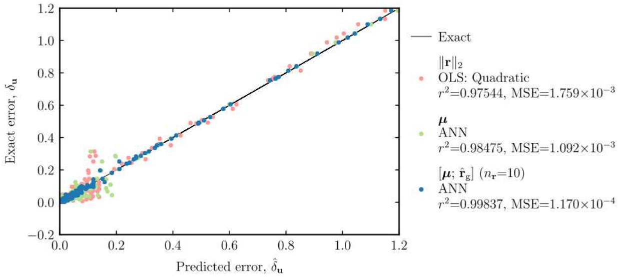
- ANN yields lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 274,954$)

PCAP: Variance Unexplained for Normed State-Space Error Prediction



- ANN yields lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 274,954$)

PCAP: Normed State-Space Error Predictions

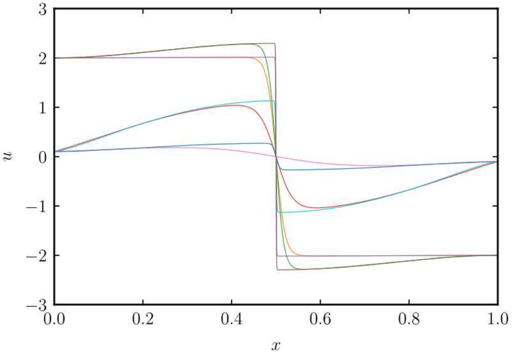


- Our method beats previous state-of-the-art methods with $r^2 > 0.998$

Burgers' Equation: Inexact Solutions and Coarse Solution Prolongation

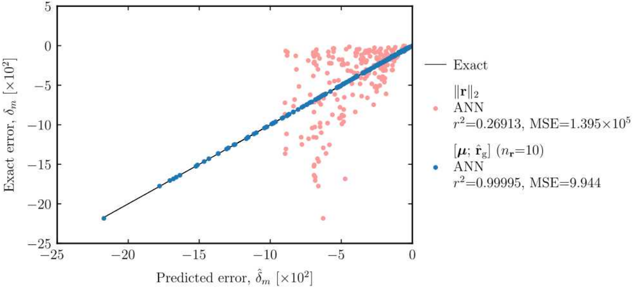
$$uu_x - \frac{1}{R}u_{xx} = \alpha \sin 2\pi x$$

$$u(0) = u_a, \quad u(1) = -u_a$$



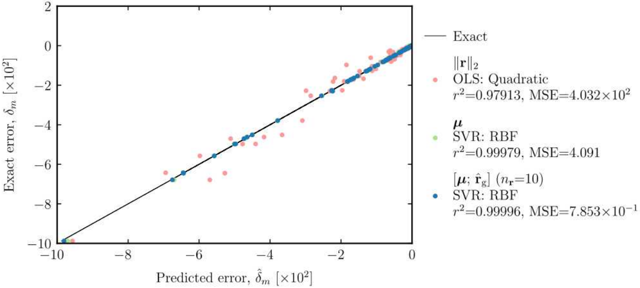
- $N_{\mathbf{u}} = 1999$
- $N_{\boldsymbol{\mu}} = 3$ parameters: $\boldsymbol{\mu} = [\alpha;\, u_a;\, R]$
  - $\alpha \in [0.1,\, 2.0]$, $u_a \in [0.1,\, 2.0]$, $R \in [50,\, 1000]$
- Quantity of interest s is the slope m at $x = \frac{1}{2}$
- $\tilde{K} = 1$ and $\tilde{K} = 2$ or $N_{\mathbf{u}_{\mathrm{LF}}} = 499$ and $N_{\mathbf{u}_{\mathrm{LF}}} = 999$

Burgers' Equation, Inexact Solutions: QoI Error Predictions



- Our method beats previous state-of-the-art method with $r^2 > 0.9999$

Burgers' Equation, Coarse Mesh Prolongation: QoI Error Predictions



- Our method beats previous state-of-the-art methods with $r^2 > 0.9999$

Outline

- Introduction
- Parameterized Systems of Nonlinear Equations
- Proposed Approach
- Numerical Experiments
- Summary
  - Feature Choices
  - Feature Reduction

Feature Choices

- Norm of the residual, $\|\mathbf{r}\|$
  - Low-quality single feature
  - Expensive to compute and performs poorly
- Dual-weighted residual, d
  - High-quality single feature
  - Performs well for small amounts of training data
  - Very expensive to compute
- Parameters $\boldsymbol{\mu}$
  - Only perform well with SVR: RBF or ANN
  - Do not perform well with OLS: Linear
- Parameters and gappy principal components of residual, $[\boldsymbol{\mu}; \hat{\mathbf{r}}_{\mathrm{g}}]$
  - Perform the best with $r^2 > 0.996$ for each experiment
  - Only require about 13 features

Feature Reduction

- Gappy PCA more effective than directly sampling the residual
- Little benefit to using $n_{\mathbf{r}} \geq 100$ samples; more samples are more expensive and do not perform much better
- Often, only $n_{\mathbf{r}} = 10$ samples are necessary to get accurate prediction

References

- B. Freno and K. Carlberg
  Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations
  Computer Methods in Applied Mechanics and Engineering (2019) `arXiv:1808.02097`
- N. Alexandrov et al.
  Approximation and model management in aerodynamic optimization with variable-fidelity models
  AIAA Journal of Aircraft (2001)
- A. Buffa et al.
  A priori convergence of the greedy algorithm for the parametrized reduced basis method
  ESAIM: Mathematical Modelling and Numerical Analysis (2012)
- M. Drohmann and K. Carlberg
  The ROMES method for statistical modeling of reduced-order-model error
  SIAM/ASA Journal on Uncertainty Quantification (2015)
- M. S. Eldred et al.
  Second-order corrections for surrogate-based optimization with model hierarchies
  AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2004)
- S. E. Gano et al.
  Hybrid variable fidelity optimization by using a kriging-based scaling function
  AIAA Journal (2005)
- M. A. Grepl and A. T. Patera
  A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations
  ESAIM: Mathematical Modelling and Numerical Analysis (2005)
- G. Rozza et al.
  Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations
  Archives of Computational Methods in Engineering (2008)
- S. Trehan et al.
  Error modeling for surrogates of dynamical systems using machine learning
  International Journal for Numerical Methods in Engineering (2017)

Questions?

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.