Statistical Emulation with Input Dimension Reduction for Complex Forward Models in Remote Sensing

Bledar Alex Konomi

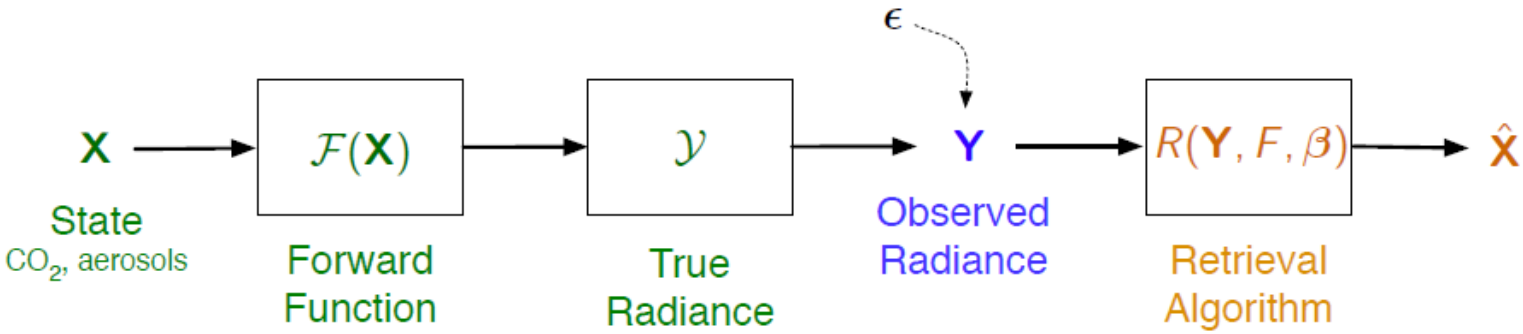
University of Cincinnati

August 14, 2025

This research is joint with: **Eric Herrison Gyamfi** (PhD student Univ. of Cincinnati), **Emily Kang** (Univ. of Cincinnati), **Gang Yang** (former PhD Student Univ. of Cincinnati), **Jon Hobbs** (Jet Propulsion Laboratory).

UNIVERSITY OF Cincinnati

Motivation: Orbiting Carbon Observatory-2 (OCO-2)

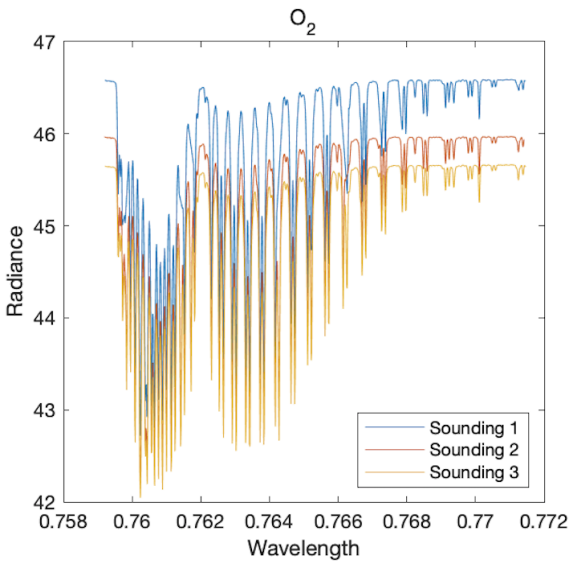


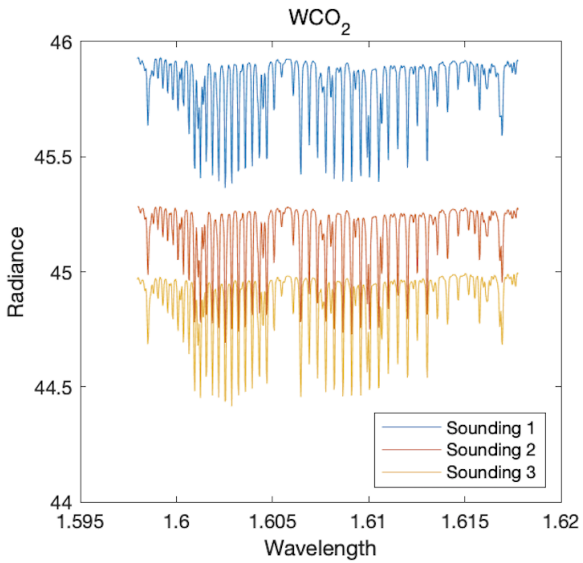
- The remote sensing observing system is a complex data-generating process with several key components.
  - The observed radiance $\boldsymbol{Y}$ is a function of geophysical variables, called the state vector $\boldsymbol{X}$ (including CO_2).
  - The observed radiance $\boldsymbol{Y}$ and $\boldsymbol{X}$ are linked through a forward model $F(\boldsymbol{X})$ and an error (model error+instrumental error).
  - The observed radiance is used to produce estimate of atmospheric variables, such as CO_2, based on the retrieval algorithm (using the forward model).

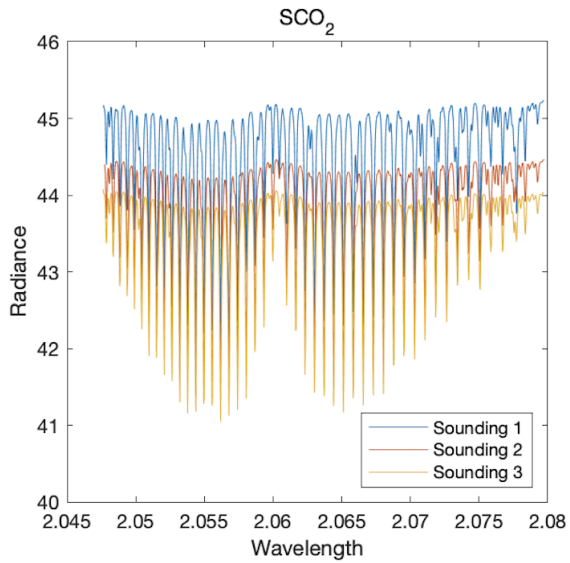
UNIVERSITY OF Cincinnati

Example of OCO2 observed radiances

The collection of observed radiances from the three spectral bands, O_2, weak CO_2 (WCO_2) and strong CO_2 (SCO_2) at a particular time makes up a sounding. A spectral band consist of **thousands of radiances** at **irregular** wavelengths.







Uncertainty Quantification in Remote Sensing

- Probabilistic assessment and uncertainty quantification of remote sensing retrievals are crucial to answer scientific hypotheses appropriately.
- Hobbs et al. (2017) and Turmon and Braverman (2019) suggest using simulation-based experiments, known as **observing system uncertainty experiments (OSUEs)**, to quantify various sources of uncertainty in probabilistic assessment of remote sensing retrievals.
  - Such experiments are supposed to be cost-effective as they are based on simulations.
  - The computational cost of the forward model $\boldsymbol{F}(\cdot)$ makes such an experiment computationally infeasible
- We would like to build a statistical emulator $\hat{\boldsymbol{F}}(\cdot)$, which does not comprise much on accuracy while enhancing computational efficiency.

UNIVERSITY OF Cincinnati

Challenges

- Developing a statistical emulator for the OCO-2 forward model $F(\cdot)$ presents some challenges:
  - The outputs of $F(\cdot)$ are radiances at hundreds of wavelengths from three bands, the O_2 band, the weak CO_2 band, and the strong CO_2 band.
  - $F(\cdot)$ is a nonlinear complex function with high-dimensional inputs state vector; $\boldsymbol{X}$ is m-dimensional with $m = 52$.
  - Directly emulating the relationship between high-dimensional outputs and inputs can be complicated, in terms of both computation and modeling.
  - We have $n = 10,849$ observations of radiances and state vectors based on which we will build a statistical emulator. This size of data can cause computational issues for a Gaussian process (GP) emulator.

UNIVERSITY OF Cincinnati

Proposed Framework

- The proposed framework consists of three steps:
  - Step 1: : Functional principal component analysis through the Principal Analysis by Conditional Estimation (PACE; Yao et al., 2005) is used to model irregularly spaced radiances and represent them via estimated low-dimensional functional principal component (FPC) scores. Model the FPCA error with a computational efficient multivariate GP.
  - Step 2: : Use Gradient-based Kernel Dimension Reduction (gKDR; Fukumizu and Leng, 2014) to avoid gradient function calculation and storage used in Active Subspace (Ma et al. 2021).
  - Step 3: : Construct a Gaussian process emulator using the low-dimensional pairs for the FPCA coefficients and the FPCA error.

UNIVERSITY OF Cincinnati

Model Set-up

- Let $Y_{ij}(w_{ij}, \boldsymbol{X}_i)$ be the spectrum radiance at wavelength w_{ij} from the i^{th} sounding, $i = 1, \ldots, n$; $j = 1, \ldots, n_i$. We define:

$$\boldsymbol{Y}_i = (Y_{i1}, Y_{i2}, \ldots, Y_{in_i})'; \boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{im})'.$$

- We would like to build a statistical emulator of $F(\cdot)$ to reproduce radiance spectra given an input $\boldsymbol{X}$.
- To achieve this, we perform dimension reduction for both input space of $\boldsymbol{X}$ and output space of $\boldsymbol{Y}$ and then build a GP emulator on the low-dimensional spaces.

UNIVERSITY OF Cincinnati

FPCA via PACE + Error

- The functional output $F(\omega, \boldsymbol{X})$ can be approximated by a truncated representation with leading eigenfunctions that explain the majority of variability

$$F(\omega, \boldsymbol{X}) = \mu(\omega) + \sum_{k=1}^{K} \xi_k(\boldsymbol{X})\phi_k(\omega) + \boldsymbol{e}(\omega, \boldsymbol{X})$$

- We perform FPCA via the Principal Analysis by Conditional Estimation (PACE; Yao et al., 2005):
  - The FPC scores $\xi_k(\boldsymbol{X})$ are estimated through conditional expectation.
- We approximate $e(w, \boldsymbol{X}) \approx \tilde{e}(\cdot, \boldsymbol{U})$ where $\tilde{e}(\cdot, \boldsymbol{U})$ is modeled as a multivariate Gaussian process.

Inputs:
$f : R^d \to R,\ \mathbf{x} \in R^d,\ Y$ (output)

Active Subspace (AS)

Main Idea:

$$\mathbf{H} = E\left[\nabla_{\mathbf{x}} f(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x})^T\right]$$
$$\approx \frac{1}{n}\sum_{i=1}^{n} (\nabla_x f_i)(\nabla_x f_i)^T = \hat{H}$$

Eigendecompose $\hat{\mathbf{H}}$

$$\mathbf{H} = [W \dot{:} V] diag(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2)[W \dot{:} V]^T$$

$\mathbf{\Phi}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$

$W \in \mathcal{R}^{d \times D} \to$ leading D eigenvectors of $\hat{\mathbf{H}}$

(Requires: $\nabla_{\mathbf{x}} f(\mathbf{x})$)

Project: $\phi = \mathbf{W}^\top \mathbf{x}$ (*active variables, D-dim*)

Model gradients often unavailable

Both seek $f(\mathbf{x}) \mapsto f(\mathbf{W}^\top \mathbf{x})$
AS: model gradients;
gKDR: kernel gradients

gKDR

Math:

SDR: $Y \perp \mathbf{x} \mid \mathbf{W}^\top \mathbf{x}$

Estimate $M(\mathbf{x})$ using kernel gradients:

$$\hat{M}(\mathbf{x}) = \nabla_{\mathbf{x}} \mathbf{k}_{\mathcal{X}}(\mathbf{x})^\top (\mathbf{G}_{\mathcal{X}} + n\epsilon_n \mathbf{I})^{-1} \mathbf{G}_{\mathcal{Y}} (\mathbf{G}_{\mathcal{X}} + n\epsilon_n \mathbf{I})^{-1} \nabla_{\mathbf{x}} \mathbf{k}_{\mathcal{X}}(\mathbf{x})$$

Eigendecompose $\hat{\mathbf{M}}$

$$\mathbf{M} = [W \dot{:} V] diag(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2)[W \dot{:} V]^T$$

$\mathbf{\Phi}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$

$W \in \mathcal{R}^{d \times D} \to$ leading D eigenvectors of $\hat{\mathbf{M}}$

(Requires: $\nabla_{\mathbf{x}} k_{\mathcal{X}}$, *not* $\nabla_{\mathbf{x}} f$)

Project: $\phi = \mathbf{W}^\top \mathbf{x}$ (*EDR space, D-dim*)

Requires kernel choice; intensive matrix algebra

UNIVERSITY OF Cincinnati

Gradient-based Kernel Dimension Reduction (gKDR)

- The Gradient-based Kernel Dimension Reduction (gKDR; Fukumizu and Leng, 2014) reduces the input dimension by finding a projection matrix $\boldsymbol{W}$ onto a p-dimensional subspace ($p < m$) , which also accounts for the role of outputs $\boldsymbol{Y}$, such that

$$p(\xi|\boldsymbol{X}) = p(\xi|\boldsymbol{W}'\boldsymbol{X})$$

  It has some advantages over the active subspace method:
  - gKDR uses positive definite kernels (reproducing kernel Hilbert spaces; RKHS) for nonparametric estimation of the projection matrix.
  - It does not require the gradient function of $F(\cdot)$ as in the active subspace method (AS; Constantine et al., 2014; Ma et al., 2021).
  - It doesn't assume any specific parametric models for $p(\xi|\boldsymbol{X})$ and can be used for multivariate outputs.
  - Compared to some other dimension reduction methods, it is computationally inexpensive.

Gaussian Process (GP) Emulation

- After dimension reduction:

$$Y(w) \Rightarrow \boldsymbol{\xi} = (\xi_1, \ldots, \xi_K)' \quad \textit{and} \quad \boldsymbol{X} \in \mathcal{R}^m \Rightarrow \boldsymbol{U} = \boldsymbol{W}'\boldsymbol{X} \in \mathcal{R}^p$$

- For each ξ_k, we consider a Gaussian process in the reduced input space $\boldsymbol{U}$:

$$\tilde{\xi}_k \sim \mathcal{NNGP}\big(\mu(\cdot), \sigma^2 R(\cdot,\cdot)\big)$$

where $\mu(\cdot)$ mean function, σ^2 the variance, and $R(\cdot,\cdot)$ GP correlation.

- For the multivariate error $\tilde{\mathbf{e}}$ we consider a Partial Parallel Gaussian process (Gu and Berger 2016) in the reduced input space $\boldsymbol{U}$:

$$\tilde{\mathbf{e}} \sim \mathcal{GP}\big(\boldsymbol{\mu}(\cdot), \Sigma \otimes R(\cdot,\cdot)\big),$$

where $\boldsymbol{\mu}$ is a vector of mean functions, Σ a diagonal matrix, $R(\cdot,\cdot)$ GP correlation.

Summary of Emulator-Construction Steps

- FPCA via PACE: For $i = 1, \ldots, n$,

$$Y_i(\omega, \boldsymbol{X}_i) \Rightarrow \mu(\omega); \{\phi_1(w), \ldots, \phi_K(\omega)\}; \boldsymbol{\xi}_i = (\xi_{i1}, \ldots, \xi_{iK})$$

- gKDR:

$$\boldsymbol{X}_i \Rightarrow \boldsymbol{U}_i \text{ where } \boldsymbol{U}_i = \boldsymbol{W}'\boldsymbol{X}$$

- NNGP emulation: For $k = 1, \ldots, K$,

$$\tilde{\xi}_k|\boldsymbol{U} \sim \mathcal{GP}(\cdot, \cdot) \Rightarrow \hat{\xi}_k(\boldsymbol{U}^*)$$

- PPNNGP emulation for the error:

$$\tilde{\mathbf{e}}|\boldsymbol{U} \sim \mathcal{MGP}(\cdot, \cdot) \Rightarrow \hat{\mathbf{e}}(\boldsymbol{U}^*)$$

- Reconstruction:

$$\hat{F}(\omega_j, \boldsymbol{X}^*) = \mu(\omega_j) + \sum_{k=1}^{K} \hat{\xi}_k(\boldsymbol{U}^*)\phi_k(\omega_j) + \hat{e}(\omega_j, \boldsymbol{U}^*)$$

UNIVERSITY OF Cincinnati

PACE, cont'd

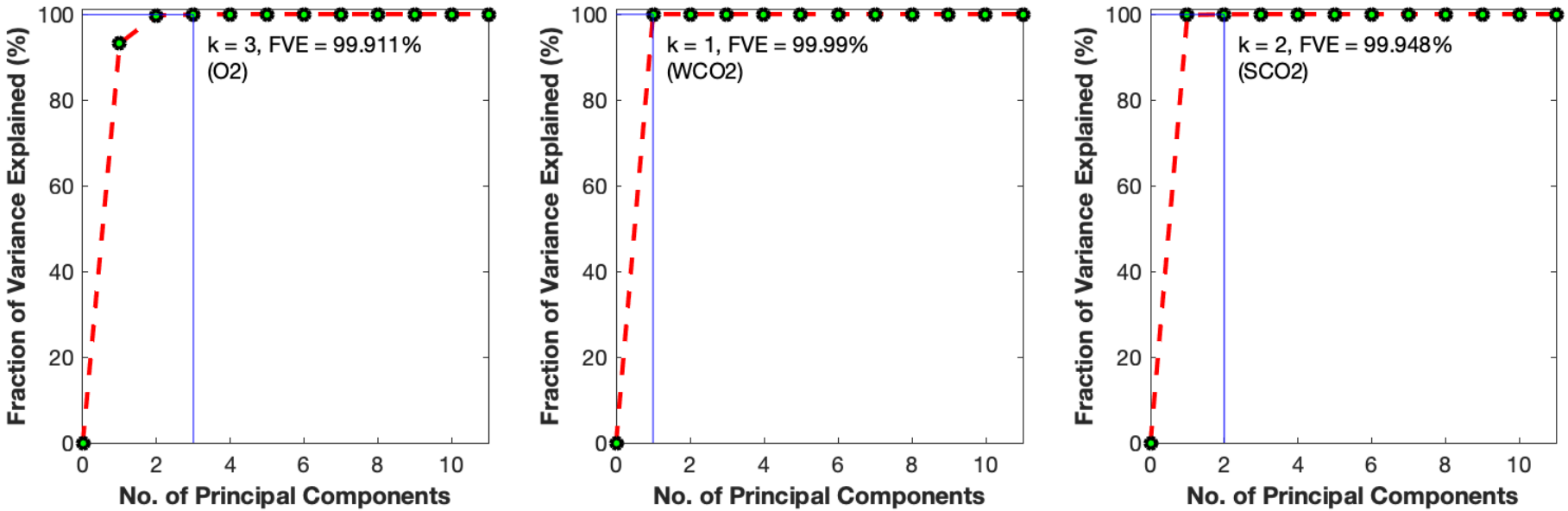


Figure: Cumulative fraction of variance explained with varying K in FPCA for the three bands.

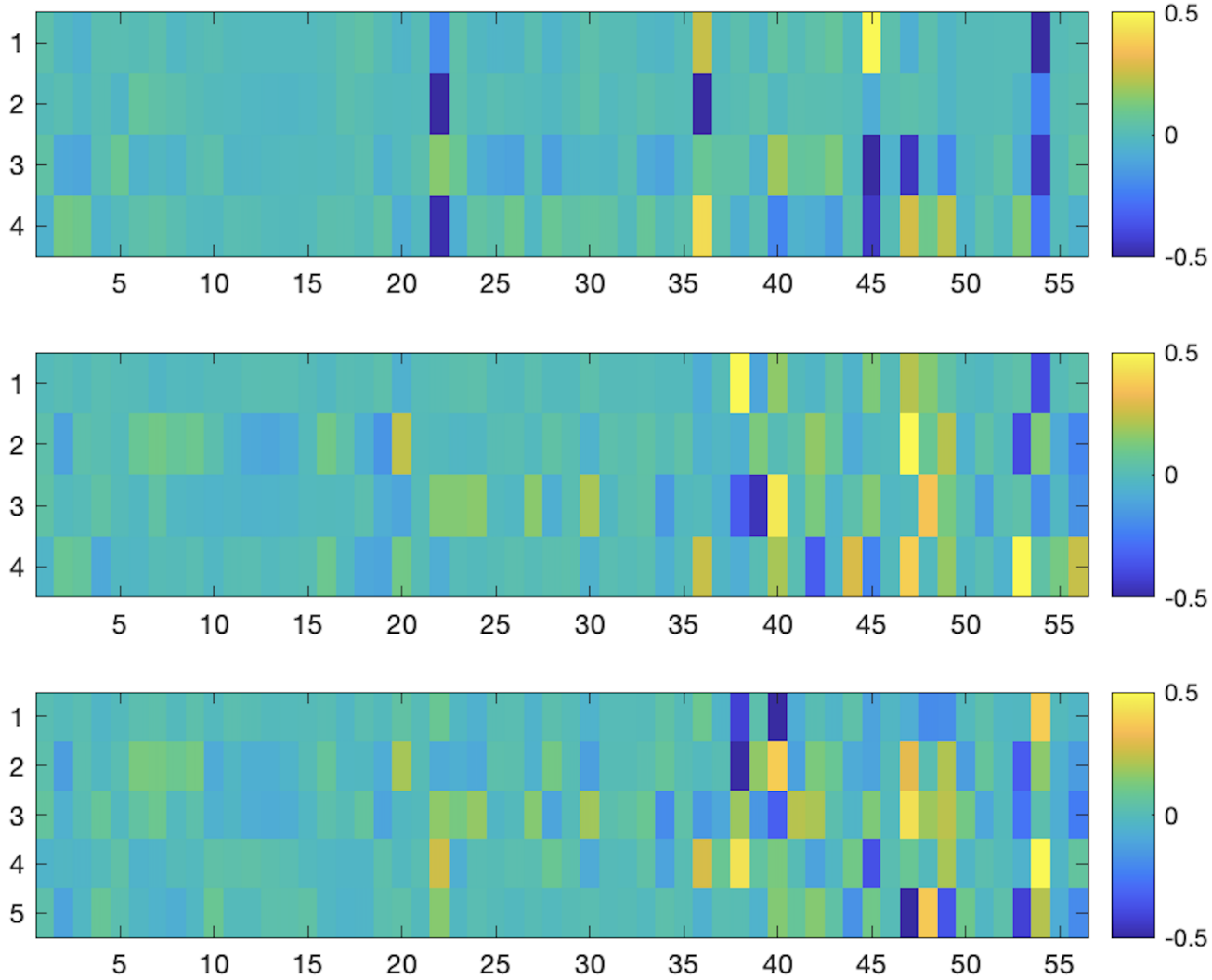
We choose $K = 3$, 1, and 2, for the O_2, WCO_2, and SCO_2 bands, respectively. Despite the large variation explained the small scale variation can be crucial for the retrieval inference (that is why it is important to model $e(w, \boldsymbol{X})$).

Choose p in gKDR

- For $p = 2, 3, \ldots, 6$, we perform gKDR and obtain the corresponding projection matrices.
- For computational efficiency the estimation process is spit in two parts:
  - First we use simple k-nearest neighbor (kNN) method to select best combination of $\{c_1, c_2\}$ (variance parameters) that gives the least error for each d
  - Then we select p such that the resulting GP emulator gives the smallest RMSPE for FPC scores in a 5-fold cross validation.
- We choose $p = 5$, 5, and 6 for O_2, WCO_2, and SCO_2 bands respectively.

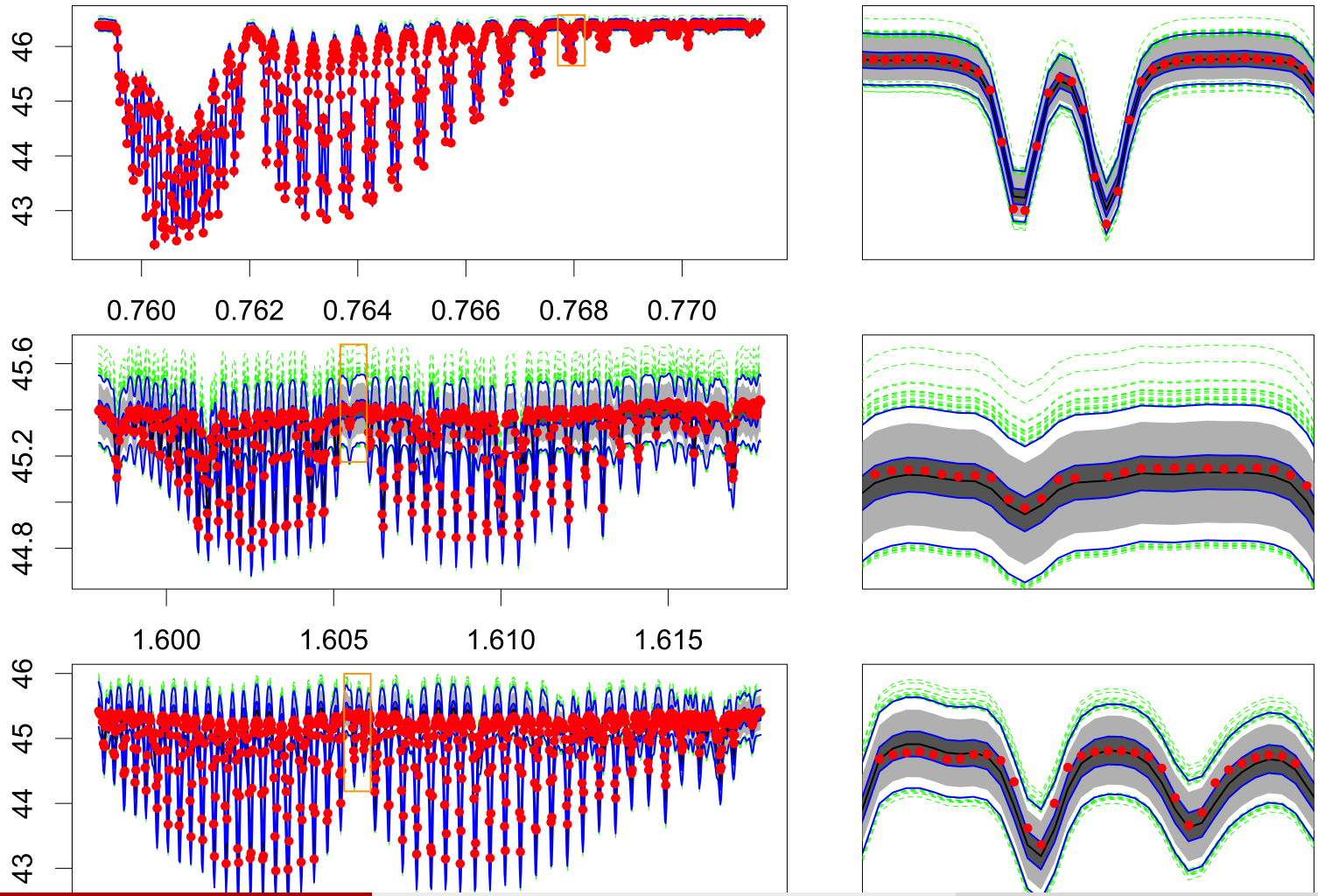
Estimated projection matrix $\hat{\mathbf{B}}_{\hat{d}}$ in gKDR

Figure 2 displays the estimated projection matrix $\hat{\mathbf{B}}_{\hat{d}}$ for 3 bands.



Reconstructing Radiances

- We treat $F^{(i)}(\cdot)$ as curves. Functional boxplots (Sun and Genton, 2011) can be used to visualize results.



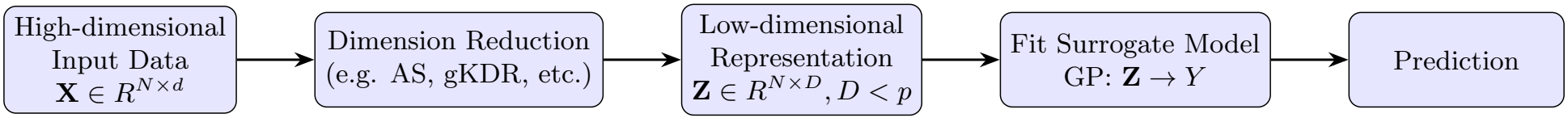
Comparing gKDR-E with AS-E

- We compare our emulator (gKDR-E) with the Active-Subspace-based emulator (AS-E) Ma et al. (2021).
- RMSPEs are calculated for each spectrum band. Our emulator outperforms AS-E. Furthermore, our work explores the approximation properties of the resulting emulator.

	AS-E			gKDR-E		
Bands	RMSPE	$P_{CI}(95\%)$	CRPS	RMSPE	$P_{CI}(95\%)$	CRPS
O_2	0.2307	0.9186	0.1255	0.0753	0.9153	0.0305
WCO_2	0.2507	0.9500	0.1332	0.0633	0.9450	0.0307
SCO_2	0.4184	0.9346	0.2254	0.1503	0.8344	0.0765

Extending the Input Dimension Reduction

- **Gaussian Processes (GPs)** model complex systems due to **predictions** with **uncertainty quantification** (UQ).
- **Limitation:** GPs scale poorly in high dimensions – accuracy drops and computation becomes costly (**curse of dimensionality**).
- **Conventional approach:** Dimension reduction $\rightarrow$ GP modeling (two-stage pipeline).



- **Recent works:** Emerging gradient-free methods exist but often lack a *fully Bayesian* framework (Tripathy et al. 2016 and Gautier et al. 2021).
- **Our contribution:** A unified, fully Bayesian GP with integrated dimension reduction.

Methodology: GP with Built-in Dimension Reduction

- Let $\mathbf{x} \in \mathbb{R}^p$ be $p-$ high-dimensional inputs and $y = f(\mathbf{x}) : \mathcal{R}^p \to \mathcal{R}$ the response.
- Let $\mathbf{z} = W^T\mathbf{x} \in \mathbb{R}^D$ be $D-$ low-dimensional inputs via projection matrix, $W \in \mathbb{R}^{p\times D}$ and $g(\mathbf{z}) : \mathcal{R}^D \to \mathcal{R}$ link function.
- Assume $\mathbf{W}$ defined on Stiefel manifold, $\mathcal{V}_{p,D}$

$$\mathcal{V}_{p,D} = \{W \in \mathcal{R}^{p\times D} : W^T W = \mathcal{I}_D\}, \quad \mathcal{I}_D = \text{identity matrix}$$

- Goal: replace costly $f(\mathbf{x})$ with $g(\mathbf{z})$, assuming $f(\mathbf{x}) \approx g(\mathbf{z})$.
- W maps $\mathbb{R}^p \to \mathbb{R}^D$ where $D < p$, enabling dimension reduction (DR) in GP modeling.
- For n input points and $Y = (y_1, \cdots, y_n)^T \in \mathbb{R}^n$, GP model is defined on $Z = (\mathbf{z_1}, \cdots, \mathbf{z_n})^T \in \mathbb{R}^{n\times D}$:

$$Y \sim \text{GP}(\mu_Y, \Sigma(Z)), \quad \Sigma(Z) = \tau^2[C(Z; \theta_D, W) + g]$$

- τ^2 is process variance, g is nugget, $C(Z; \theta_D, W)$ is an isotropic kernel with lengthscale θ_D and $\mu_Y = 0$.

Use Deep Gaussian Process

Limitations of Standard GPs:

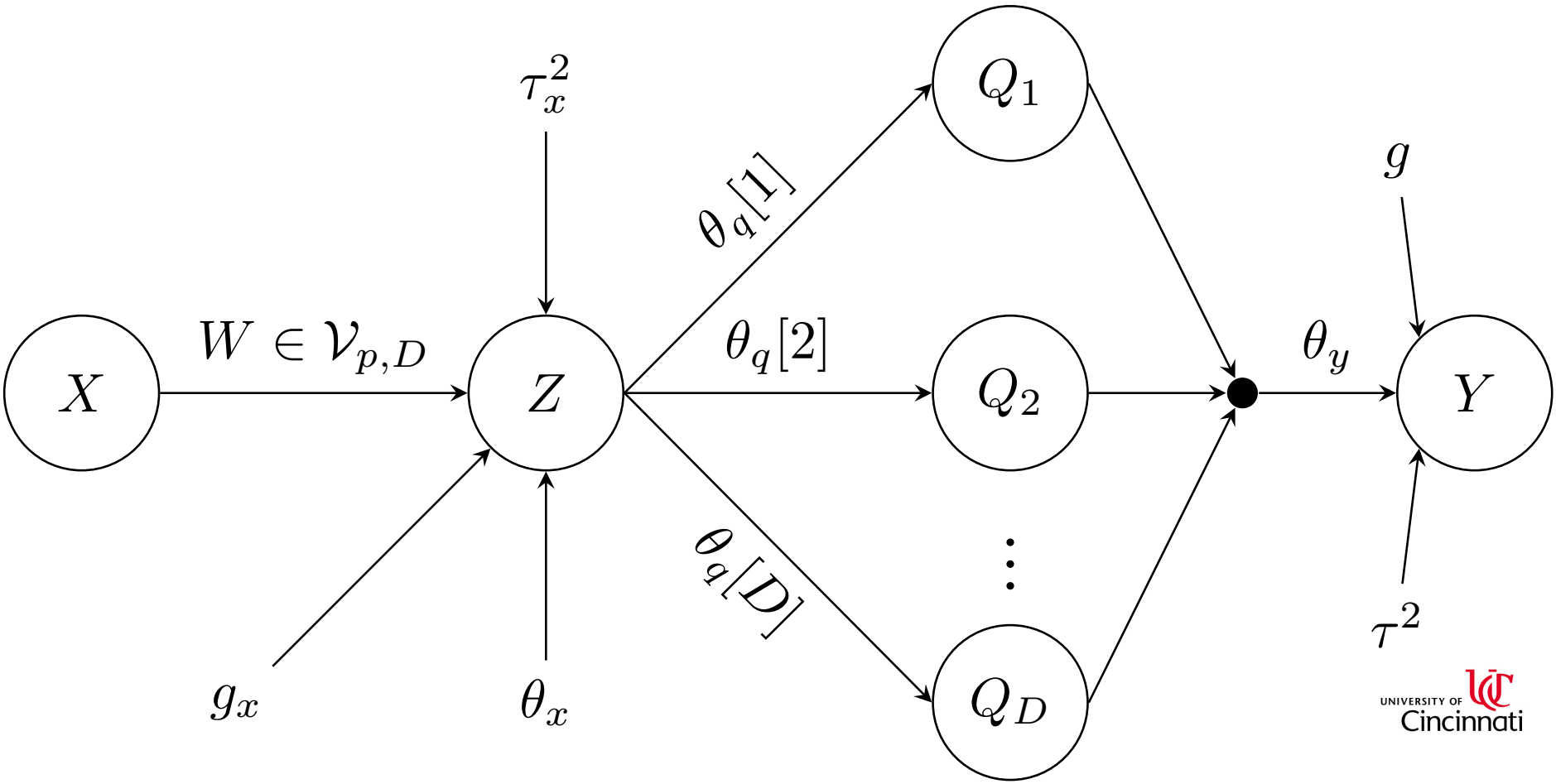
- Assume uniform statistical behavior across input space(stationarity).
- Struggle with input-output relationship changing patterns .

Benefits of DGPs:

- Adapt to changing relationships across input regions(non-stationary).
- Stack multiple GPs layers to:
  - Model varying smoothness.
- Maintain GP strengths:
  - Accurate predictions.
  - Reliable uncertainty estimates.

UNIVERSITY OF Cincinnati

Two-layer DGP Model with built-in Dimension Reduction



Two-layer DGP Model with built-in Dimension Reduction

$$Q_j \sim^{\text{ind}} \mathcal{N}(0, C_{\theta_Q[j],W}(Z)), \quad Q_j \in \mathbb{R}^n, \quad W \in \mathcal{V}_{p,D}, \quad j = 1, \ldots, D.$$

$$\theta_Q = (\theta_Q[1], \cdots, \theta_D[D]) \in \mathbb{R}^D, \quad Q = [Q_1, \cdots, Q_D] \in \mathbb{R}^{n \times D}$$

$$Y \mid Q \sim \mathcal{N}(0, \tau^2[C_{\theta_y}(Q) + g\mathcal{I}_n])$$

$$\mathcal{L}(Y \mid X) \propto \int \mathcal{L}(Y \mid Q)\mathcal{L}(Q \mid Z)\mathcal{L}(Z \mid X)\, dZdQ$$

- Q: Latent variable and $Q_1, \cdots, Q_D$ be the latent nodes
- $C_\theta(\cdot)$: Covariance function with parameters θ.
- θ_Q, θ_y: Covariance hyperparameters for latent/output layers.
- g, g_x: Nugget terms (noise parameters).
- τ^2, τ_x^2: Variance scale parameters.

UNIVERSITY OF Cincinnati

Projection Matrix (W) Prior

$$W \sim \mathcal{ML}(W; F), W \in \mathcal{V}_{p,D} \quad \text{- Matrix Langevin } (\mathcal{ML}) \text{ prior}$$

- $\mathcal{ML}-$distribution with respect to Haar measure μ on $\mathcal{V}_{p,D}$:

$$\pi_{\mathcal{ML}}(W; F) = \frac{1}{c(F)} \exp(\text{tr}(F^T W)), \quad c(F) = {}_0\mathcal{F}_1\left(\frac{d}{2}, \frac{F^T F}{4}\right)$$

- ${}_0\mathcal{F}_1\left(\frac{d}{2}, \frac{F^T F}{4}\right)$ is hypergeometric function of order $\frac{d}{2}$ with matrix $F^T F/4$.

Parameterization of F via SVD:

$$F = M\Lambda V^T, \quad \Lambda = \text{diag}(\{\lambda_1, \cdots, \lambda_D\}) \text{ where}$$

Posterior Inference Summary

- Perform fully Bayesian inference for DGP models via MCMC.
- Posterior inference for W, Q and λ is intractable.
- Hybrid MCMC framework prioritizing UQ:
  - Metropolis-Hastings (MH) for $\{g_x, \theta_x, \theta_Q, g, \theta_y\}$.
  - Hamiltonian Monte Carlo (HMC) for W (or $Z = WX$).
  - Elliptical Slice Sampling (ESS) for $Q_1, \cdots, Q_D$ and λ - requires no tuning as recently employed in Sauer et al. 2023.

UNIVERSITY OF Cincinnati

Numerical Experiment 1: Synthetic Data

Data Generation: Employed in Tripathy 2016.

- Inputs $\mathbf{x} \sim \mathcal{N}_d(0, \mathbf{I})$; $d = 10$.
- Output: $Y = \mathbf{a}_0 + \mathbf{a}^T\phi + \phi^T\mathbf{A}\phi + \epsilon$, $\phi = \mathbf{W}^\top\mathbf{x}$, $\epsilon \sim \mathcal{N}(0, 0.01)$.

Scenario 1: 2D Input subspace

$$\mathbf{W} = \begin{pmatrix} 0.008 & -0.184 & 0.343 & -0.053 & 0.081 & 0.066 & -0.412 & 0.654 & 0.485 & 0.040 \\ 0.067 & -0.415 & 0.482 & 0.076 & 0.210 & 0.538 & 0.078 & -0.200 & -0.291 & 0.348 \end{pmatrix}^T$$

$$\mathbf{a}_0 = -0.06976, \quad \mathbf{a} = (0.4376, 0.9870)^T, \quad \mathbf{A} = \begin{pmatrix} -0.9257 & -0.3840 \\ -0.4174 & -0.6766 \end{pmatrix}$$

Experimental Setup:

- $n = 600$ samples; training set is 80%, test set 20%.
- Baseline Methods:Active Subspace (AS), gradient kernel dimension reduction (gKDR), Gaussian process maximum likelihood estimate (GP-MLE)
- Performance metrics: root mean square prediction error (RMSPE), Nash-Sutcliffe model efficiency coefficient (NSME), Continuous Ranked Probability Score (CRPS), Score, Bayesian Information Criterion (BIC) and mean log pointwise predicted density (MLPPD)

- Baseline methods:
  - **Active Subspace (AS):** Identifies dominant input directions via the second-moment matrix of simulator gradients $\nabla_x f(x)$ (**Constantine et al. 2014**). **Requires direct gradient access**, making it impractical for black-box simulators.
  - **gKDR:** Builds on the sufficient DR framework using gradients of the *input kernel*, not the simulator (**Fukumizu and Leng 2014**).
  - **GP-MLE:** Employs MLE to estimate W and the covariance hyperparameters (**Tripathy et al. 2016**).
- DGP **A** - layer (**D**) denotes DGP with **A** layer(s) and input subspace **D** where $\mathbf{A}, \mathbf{D} = 1, 2, 3$.
- DGP **A** - layer (**D**) W/o represents DGP with **A** layer(s) and input subspace **D** without DR.
- DGP **A** - layer (**D**) Truth represents DGP with **A** layer(s) and input subspace **D** with DR but uses the true W.

UNIVERSITY OF Cincinnati

Method (D)	RMSPE	NSME	CRPS	Score	BIC
AS (2)	0.2596	0.9717	0.7833	−232.3104	245.53
gKDR (2)	0.1849	0.9907	0.5637	281.7241	240.30
GP-MLE (2)	0.0946	0.9976	0.5347	449.8206	246.11
DGP 1-layer (1)	0.1055	0.9945	0.1023	260.5579	248.01
DGP 1-layer (2)	**0.0815**	**0.9982**	**0.0162**	***715.9930***	***249.20***
DGP 1-layer (3)	0.1272	0.9940	0.2499	712.2032	247.52
DGP 2-layer (1)	0.1437	0.9929	0.1046	643.7125	247.00
DGP 2-layer (2)	0.1100	0.9972	0.1442	266.6124	248.33
DGP 2-layer (3)	0.1592	0.9949	0.5855	540.6212	246.07
DGP 3-layer (1)	0.1937	0.9873	0.9335	206.9284	244.02
DGP 3-layer (2)	0.1821	0.9902	0.9890	146.8134	245.06
DGP 3-layer (3)	0.1941	0.9850	0.5073	694.1730	243.90
DGP 1-layer (10) W/o	0.2194	0.9821	0.2376	220.6803	240.00
DGP 2-layer (10) W/o	0.2175	0.9856	0.7510	317.6212	243.45
DGP 3-layer (10) W/o	0.2070	0.9702	0.6450	104.6729	243.98
DGP 1-layer (2) Truth	***0.0706***	***0.9993***	***0.0118***	**324.2333**	***250.08***
DGP 2-layer (2) Truth	0.0981	0.9941	0.1496	317.5155	246.59
DGP 3-layer (2) Truth	0.1629	0.9945	0.2646	331.7272	247.00

Numerical Experiment 2

2D Input Subspace: Employed in Sauer et al. 2023

- Inputs $x \in [0,1]^{10}$ from Latin Hyperpercube Sample
- Projected by known 10×2 matrix W.
- $z = (z_1, z_2) = W^\top x$ with response function given as

$$f(z) = 10 z_1 \exp(-z_1^2 - z_2^2); z_j = (z_j - 0.5) \cdot 6 + 1, j = 1, 2.$$

- W is the same as numerical experiment 1
- $n = 300$ samples; training set is 80%, test set 20%.

University of Cincinnati

Method (D)	RMSPE	NSME	CRPS	Score	BIC
AS (2)	0.6190	0.8329	0.5420	90.2640	601.39
gKDR (2)	0.7791	0.7562	0.6429	88.7980	596.24
GP-MLE (2)	0.6052	0.8400	0.5113	91.0290	604.08
DGP 1-layer (1)	0.4795	0.9035	0.5938	104.6915	615.28
DGP 1-layer (2)	0.5302	0.8873	0.4549	63.3460	612.07
DGP 1-layer (3)	0.5948	0.7691	0.5863	72.6261	608.65
DGP 2-layer (1)	0.4778	0.9079	0.4097	126.6638	616.90
DGP 2-layer (2)	0.4217	0.9192	0.4490	100.6576	618.34
DGP 2-layer (3)	0.5616	0.7897	0.5677	73.6139	610.71
DGP 3-layer (1)	0.4823	0.8937	**0.3705**	87.8325	616.10
DGP 3-layer (2)	**0.4045**	**0.9240**	0.4178	***128.5832***	***619.20***
DGP 3-layer (3)	0.5221	0.8895	0.5377	104.7551	615.37
DGP 1-layer (10) W/o	0.6761	0.7635	0.7710	80.3565	600.03
DGP 2-layer (10) W/o	0.6028	0.8459	0.6382	54.3635	603.22
DGP 3-layer (10) W/o	0.6339	0.8096	0.6329	69.3009	601.48
DGP 1-layer (2) Truth	0.4344	0.9150	**0.4994**	53.8692	617.46
DGP 2-layer (2) Truth	0.4917	0.8929	0.5207	90.0178	616.95
DGP 3-layer (2) Truth	***0.3215***	***0.9371***	0.5308	**113.6072**	***620.88***

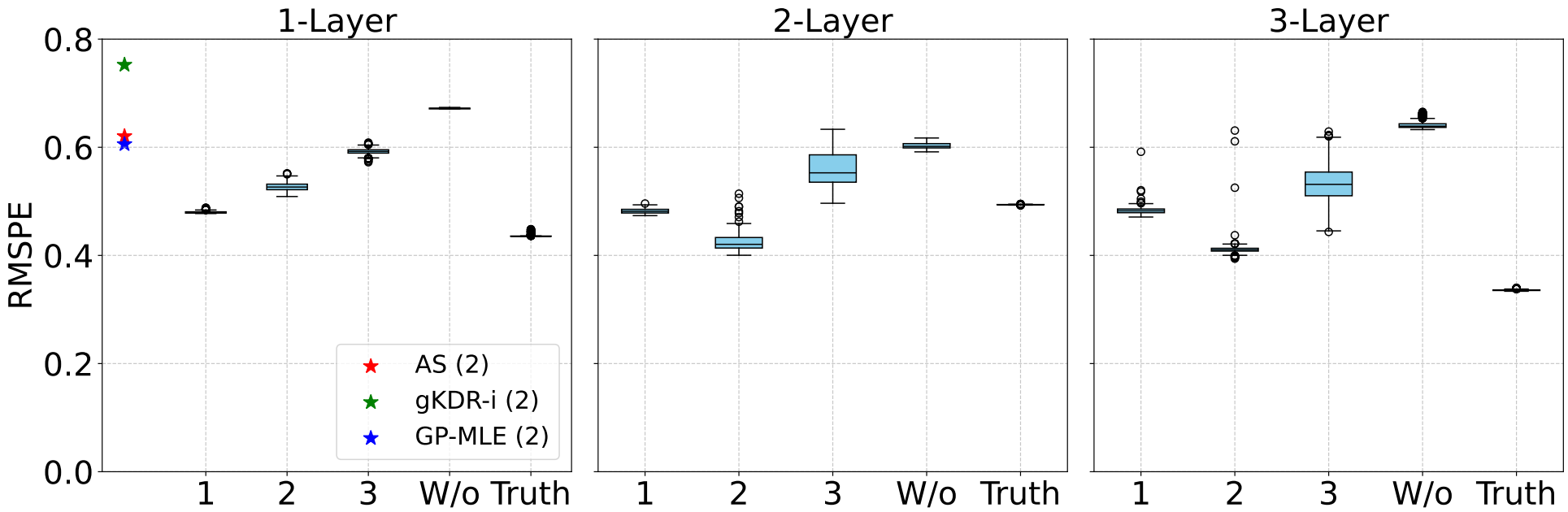


Figure: RMSPE comparisons across different method for train size 240

Discussion and Conclusion

Simulation Results:

- Models with built-in DR consistently outperformed models W/o.
- DGPs with appropriate layer depth adapted well, regardless of the complexity.
- Very deep models (3 layers) sometimes showed diminishing returns or overfitting in low-complexity settings.
- Performance gains from added depth were more apparent when the response surface was complex.

Conclusion:

- Model selection is key since data complexity is not known beforehand.
- Start with a moderate number of DGP layers and D, tune for the data at hand.
- Fully Bayesian DGPs with dimension reduction provide flexible modeling and robust performance across a range of complexities.

Thank You for the Attention!

Questions+Answers?

References

- P. Ma*, A. Mondal, **B. A. Konomi**, J. Hobbs, J. Song, E. L. Kang, “Statistical Emulation for High-Dimensional Functional Outputs in Large-Scale Observing System Uncertainty Experiments", Accepted to ***Technometrics***, **2021**.
- K. Fukumizu C. Leng, “*Gradient-Based Kernel Dimension Reduction for Regression*", Journal of the American Statistical Association, **2014**.
- P. G. Constantine, E. Dow, Q. Wang, “*Active subspace methods in theory and practice: applications to kriging surfaces*", SIAM Journal on Scientific Computing, **2014**.
- Datta, A. and Banerjee, S. and Finley, A. O. and Gelfand, A. E.,“*Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets*", Journal of the American Statistical Association, **2016**.
- Gramacy, R.B. and Apley, D. W.,“*Local Gaussian Process Approximation for Large Computer Experiments*", Journal of Computational and Graphical Statistics, **2015**.