Gaussian Process Regression in High-Dimensional and Structured Domains

Variational Inference and Active Learning on Manifolds

Yuanxing Cheng, Advisor: Lulu Kang, Chun Liu

Illinois Institute of Technology

Ph.D. Oral Exam, July 27, 2025

Outline

Introduction: Computer Experiments

What is a computer experiment?

Introduction: Computer Experiments

What is a computer experiment? It is a simulation using computer models.

- physical,
- biological,
- engineering, etc.

Challenges

- dimension
- cost of data acquisition

Introduction: Gaussian Process

Why Gaussian Process (GP)?

- simple assumption with an explicit prediction formula
- can perform Uncertainty Quantification (UQ) easily

When using Bayesian Approach to do Gaussian Process regression,

1. assume prior distributions with hyperparameters
2. obtain the posterior distribution
3. maximize the posterior to find the best value for the hyperparameters
4. find posterior predictive distribution at untried x

Energetic Variational Inference Gaussian Process

The first method uses Energetic Variational Inference to find the maximum point of the posterior distribution function.

- Energetic Variational Approach (*Wang Y., Liu C., 2022*)
- Variational Inference (*Wang, Y., Chen, J., Liu, C. et al., 2021*)

We have

- stable posterior approximation
- variable selection

Introduction: Active Learning Manifold Gaussian Process

The second method uses a quick active learning method on a manifold Gaussian process. We aim to achieve the following:

- Dimension reduction
- Active learning

We use:

- Manifold learning: Neural network
- Active learning criteria: Active Learning Cohn

Outline

Gaussian Process Regression

The Gaussian Process model.

$$y_i = y(\boldsymbol{x}_i) = \boldsymbol{g}(\boldsymbol{x}_i)^\top \boldsymbol{\beta} + Z(\boldsymbol{x}_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

- $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ are the data, $\boldsymbol{x}_i \in \Omega \subseteq \mathbb{R}^d$, $y_i \in \mathbb{R}$
- $\boldsymbol{g}(\boldsymbol{x})$ is a p-dimensional vector of basis function, and $\boldsymbol{\beta}$ are their linear coefficients
  - zero order effect (intercept), $\boldsymbol{g}(\boldsymbol{x}) = 1$, $\boldsymbol{\beta} = \beta_0$, $p = 1$
  - first order effect, $\boldsymbol{g}(\boldsymbol{x}) = [1, x_1, x_2, \ldots, x_d]$, $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots, \beta_d]$
  - second order effect, etc.
- The noise $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_\omega^2)$ and is also independent of the other stochastic ingredients of (1).

About Gaussian Process Z(·)

$Z(\cdot) \sim GP(0, \tau^2 \boldsymbol{K})$, which indicates $\mathbb{E}[Z(\boldsymbol{x})] = 0$ and $\mathrm{Cov}\left[Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)\right] = \tau^2 k(\boldsymbol{x}_1, \boldsymbol{x}_2)$.

- τ^2 is a constant variance and $k(\cdot, \cdot; \boldsymbol{\omega}) : \Omega \times \Omega \mapsto \mathbb{R}_+$ is the correlation function of the stochastic process with hyperparameters $\boldsymbol{\omega}$.
- Here we use *Gaussian kernel* function, symmetric, and positive definite. $k(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\omega}) = \exp\left\{-\sum_{j=1}^{d} \omega_j (x_{1j} - x_{2j})^2\right\}$, with $\boldsymbol{\omega} \in \mathbb{R}^d$ and $\boldsymbol{\omega} \geq 0$.
- Here we also define the noise-to-signal ratio $\eta = \sigma_\omega^2 / \tau^2$. It's expected to be small

About response y

With above assumption, $\boldsymbol{y}_n$ is also a gaussian process where $\mathbb{E}[y(\boldsymbol{x})] = \boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{\beta}, \forall \boldsymbol{x} \in \Omega$ and

$$\begin{aligned} \operatorname{Cov}\left[y(\boldsymbol{x}_1), y(\boldsymbol{x}_2)\right] &= \tau^2 k(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\omega}) + \sigma^2 \delta(\boldsymbol{x}_1, \boldsymbol{x}_2), \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \Omega, \\ &= \tau^2 \left[k(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\omega}) + \eta \delta(\boldsymbol{x}_1, \boldsymbol{x}_2)\right], \end{aligned}$$

where $\delta(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1$ if $\boldsymbol{x}_1 = \boldsymbol{x}_2$ and 0 otherwise.

In summary, all the unknown parameters are $\boldsymbol{\beta}, \boldsymbol{\omega}, \tau^2$, and η, and we define $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\omega}, \tau^2, \eta)$.

Hyperparameters learning objective

log-marginal likelihood:

$$\ell(\tau^2, \eta) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\tau^2 - \frac{1}{2}\log\det(\boldsymbol{K}_n + \eta\boldsymbol{I}_n) - \frac{1}{2\tau^2}\boldsymbol{Y}_n^\top(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{Y}_n. \tag{2}$$

differentiating ℓ with respect to τ^2, solve

$$\hat{\tau}^2 = \frac{1}{n}\boldsymbol{Y}_n^\top(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{Y}_n. \tag{3}$$

Final form

$$\ell(\eta) = -\frac{1}{2}\boldsymbol{Y}_n^\top(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{Y}_n - \frac{1}{2}\log\det(\boldsymbol{K}_n + \eta\boldsymbol{I}_n) + C, \tag{4}$$

Outline

Priors

We assume the following prior distributions for the parameters

- $\boldsymbol{\beta} \sim \mathcal{MVN}_p(\mathbf{0}, \nu^2 \boldsymbol{R})$ or $\rho(\boldsymbol{\beta}) \propto 1$ for a non-informative prior.
- $\omega_i \overset{\text{i.i.d.}}{\sim} \text{Gamma}(a_\omega, b_\omega)$, for $i = 1, \ldots, d$
- $\tau^2 \sim \text{Inverse-}\chi^2(df_{\tau^2})$
- $\eta \sim \text{Gamma}(a_\eta, b_\eta)$.

Then we have the sampling distribution

$$\boldsymbol{y}_n | \boldsymbol{\theta} \sim \mathcal{MVN}_n \left(\boldsymbol{G}\boldsymbol{\beta}, \tau^2 (\boldsymbol{K}_n + \eta \boldsymbol{I}_n) \right), \tag{5}$$

where $\boldsymbol{y}_n$ is the vector of y_i's, $\boldsymbol{G}$ is a matrix with each row being $\boldsymbol{g}(\boldsymbol{x}_i)^\top$. The matrix $\boldsymbol{K}_n$ is the $n \times n$ kernel matrix and $\boldsymbol{K}_n[i, j] = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and is symmetric and positive definite. $\boldsymbol{I}_n$ is an $n \times n$ identity matrix.

Conditional posterior distributions of β

We start with the informative prior case. Using the Bayes formula we obtain

$$\begin{aligned} p(\boldsymbol{\theta} \,|\, \boldsymbol{y}_n) &\propto p(\boldsymbol{y}_n \,|\, \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}) \\ p(\boldsymbol{\beta} \,|\, \boldsymbol{y}_n, \boldsymbol{\omega}, \tau^2, \eta) &\propto p(\boldsymbol{y}_n \,|\, \boldsymbol{\beta}, \boldsymbol{\omega}, \tau^2, \eta) \cdot p(\boldsymbol{\beta} \,|\, \boldsymbol{\omega}, \tau^2, \eta) \\ &\propto p(\boldsymbol{y}_n \,|\, \boldsymbol{\theta}) \cdot p(\boldsymbol{\beta}) \end{aligned}$$

Then, $\boldsymbol{\beta}|\boldsymbol{y}_n, \boldsymbol{\omega}, \tau^2, \eta \sim \mathcal{MVN}_d\left(\hat{\boldsymbol{\beta}}_n, \boldsymbol{\Sigma}_{\boldsymbol{\beta}|n}\right)$, where

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}|n} = \left[\frac{1}{\tau^2}\boldsymbol{G}^\top(\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}\boldsymbol{G} + \frac{1}{\nu^2}\boldsymbol{R}^{-1}\right]^{-1}$$

$$\hat{\boldsymbol{\beta}}_n = \boldsymbol{\Sigma}_{\boldsymbol{\beta}|n}\frac{(\boldsymbol{G}^\top(\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}\boldsymbol{y}_n)}{\tau^2}$$

Shrinkage Effect and variable selection

$\boldsymbol{\Sigma}_{\boldsymbol{\beta}|n}$ and $\hat{\boldsymbol{\beta}}_n$ are used to form an interval estimate. When the confidence interval contains 0, the corresponding effect is considered nonsignificant and is therefore discarded.

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}|n} = \left[\frac{1}{\tau^2}\boldsymbol{G}^\top(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{G} + \frac{1}{\nu^2}\boldsymbol{R}^{-1}\right]^{-1}$$

Thus, we use $\boldsymbol{R}$ for regularization and ν as the shrinking coefficient. We define the matrix $\boldsymbol{R}$ to be a diagonal matrix and follow the Hierarchy principle,

- *lower order effects are more likely to be significant than higher order effects*
- *effects of the same order are equally likely to be significant*

Structure of Basis Function and Regularization Matrix

$$\boldsymbol{g}(\boldsymbol{x}_i)^\top = [\ 1 \quad x_1 \quad x_2 \quad \cdots \quad x_d \quad x_1^2 \quad x_1x_2 \quad \cdots \quad x_d^2]$$

Order : 0th 1st 2nd

$$\mathbf{R} = \begin{bmatrix} 1 & & & & & & & & \\ & r & & & & & & & \\ & & r & & & & & & \\ & & & \ddots & & & & & \\ & & & & r & & & & \\ & & & & & r^2 & & & \\ & & & & & & r^2 & & \\ & & & & & & & \ddots & \\ & & & & & & & & r^2 \end{bmatrix}, \quad r > 1$$

Variable Selection step

1. use cross validation to find the best ν
2. with fixed ν, do regression and calculate the interval estimate of $\hat{\boldsymbol{\beta}}_n$
3. if a confidence interval contains 0, it's nonsignificant and will be discarded
4. according to the weak Heredity principle, *higher-order effects are only considered if one or more of its parent effects are significant*; otherwise, they are removed.
   - e.g. x_1x_2 will be discarded if neither x_1 nor x_2 is significant
5. The new model has a new $\boldsymbol{g}$ and $\boldsymbol{\beta}$, repeat the above steps.

Posterior, informative prior case

With the informative prior, the posterior distribution is

$$\begin{aligned} p(\boldsymbol{\omega}, \tau^2, \eta | \boldsymbol{y}_n) &\propto \int p(\boldsymbol{\theta} \,|\, \boldsymbol{y}_n)\, \mathrm{d}\boldsymbol{\beta} \\ &\propto \det(\boldsymbol{\Sigma}_{\boldsymbol{\beta}|n})^{1/2} \\ &\times \exp\left[-\frac{1}{2}\hat{\boldsymbol{\beta}}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{\beta}|n}^{-1}\hat{\boldsymbol{\beta}}_n + \frac{1}{2\tau^2}\boldsymbol{y}_n^\top(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{y}_n\right] \\ &\times \det(\boldsymbol{K}_n + \eta\boldsymbol{I}_n)^{1-/2}(\tau^2)^{-n/2} p(\tau^2) p(\boldsymbol{\omega}) p(\eta) \qquad (6) \end{aligned}$$

And if using the noninformative prior, we can further integrate (6) with respect to τ^2 and obtain $p(\boldsymbol{\omega}, \eta \,|\, \boldsymbol{y}_n)$.

Noninformative prior case

In the noninformative prior case, we have the estimate of $\hat{\beta}_n$ being normal with its variance and mean being

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}|n} = \tau^2 \left[\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{G}\right]^{-1}$$
$$\hat{\boldsymbol{\beta}}_n = \left[\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{G}\right]^{-1} \left[\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}\right] \boldsymbol{y}_n$$

In this case, we can define

$$\begin{aligned} s_n^2 &= \tau^{-2} \hat{\boldsymbol{\beta}}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{\beta}|n}^{-1} \hat{\boldsymbol{\beta}}_n + \boldsymbol{y}_n^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{y}_n \\ &= \boldsymbol{y}_n^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \left[\boldsymbol{G} \left(\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{G}\right)^{-1} \boldsymbol{G}^\top + (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)\right] \\ &\quad \times (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{y}_n. \end{aligned}$$

so that above posterior (6) can be simplified.

The simplified posterior distribution:

$$p(\boldsymbol{\omega}, \tau^2, \eta|\boldsymbol{y}_n) \propto (\tau^2)^{-[\frac{1}{2}(df_{\tau^2}+n-p)+1]} \exp\left(-\frac{1+s_n^2}{2\tau^2}\right) \times \det(\boldsymbol{G}^\top(\boldsymbol{K}_n+\eta\boldsymbol{I}_n)^{-1}\boldsymbol{G})^{-1/2} \times \det(\boldsymbol{K}_n+\eta\boldsymbol{I}_n)^{1-/2} p(\boldsymbol{\omega})p(\eta) \quad (7)$$

Notice, $p(\boldsymbol{\omega}, \tau^2, \eta|\boldsymbol{y}_n) \propto p(\tau^2|\boldsymbol{\omega}, \eta, \boldsymbol{y}_n)p(\boldsymbol{\omega}, \eta \,|\, \boldsymbol{y}_n)$. Define $\hat{\tau}^2 = \left(1+s_n^2\right) / \left(df_{\tau^2}+n-p\right)$, we actually have $\tau^2|\boldsymbol{\omega}, \eta, \boldsymbol{y}_n \sim$ Scaled Inverse-$\chi^2(df_{\tau^2}+n-p, \hat{\tau}^2)$.

We then integrate (7) with τ^2 and obtain

$$p(\boldsymbol{\omega}, \eta|\boldsymbol{y}_n) \propto (\hat{\tau}^2)^{-\frac{1}{2}(df_{\tau^2}+n-p)} \times \det(\boldsymbol{G}^\top(\boldsymbol{K}_n+\eta\boldsymbol{I}_n)^{-1}\boldsymbol{G})^{1-/2} \det(\boldsymbol{K}_n+\eta\boldsymbol{I}_n)^{1-/2} p(\boldsymbol{\omega})p(\eta). \quad (8)$$

Posterior predictive distribution

Given the parameters $(\boldsymbol{\omega}, \tau^2, \eta)$, the posterior predictive distribution of $y(\boldsymbol{x})$ at query point $\boldsymbol{x} \in \Omega$ is the following

$$y(\boldsymbol{x})|\boldsymbol{y}_n, \boldsymbol{\omega}, \tau^2, \eta \sim \mathcal{N}(\hat{\mu}(\boldsymbol{x}), \sigma_n^2(\boldsymbol{x})), \tag{9}$$

where

$$\hat{\mu}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x})^\top \hat{\boldsymbol{\beta}}_n + K(\boldsymbol{x}, \mathcal{X}_n)(\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}(\boldsymbol{y}_n - \boldsymbol{G}\hat{\boldsymbol{\beta}}_n), \tag{10}$$

$$\sigma_n^2(\boldsymbol{x}) = \tau^2 \Big\{ 1 - K(\boldsymbol{x}, \mathcal{X}_n)(\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} K(\mathcal{X}_n, \boldsymbol{x}) + \boldsymbol{c}(\boldsymbol{x})^\top \left[\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{G}\right]^{-1} \boldsymbol{c}(\boldsymbol{x}) \Big\} \tag{11}$$

$$\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} K(\mathcal{X}_n, \boldsymbol{x}),$$

$$K(\boldsymbol{x}, \mathcal{X}_n) = K(\mathcal{X}_n, \boldsymbol{x})^\top = [K(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_n)]$$

Computational Issue

To avoid computational issues caused by ill-conditioned matrices, here we plug in $\hat{\beta}_n$ into $\hat{\mu}(\boldsymbol{x})$ and obtain the alternative formula.

$$\hat{\mu}(\boldsymbol{x}) = \Bigg(\left(\boldsymbol{g}(\boldsymbol{x})^\top - K(\boldsymbol{x}, \mathcal{X}_n)(\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}\boldsymbol{G}\right) \times \left(\boldsymbol{G}^\top (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1}\boldsymbol{G}\right)^{-1} \boldsymbol{G}^\top + K(\boldsymbol{x}, \mathcal{X}_n) \Bigg) (\boldsymbol{K}_n + \eta \boldsymbol{I}_n)^{-1} \boldsymbol{y}_n$$

Sampling Procedure

The sampling procedure is

1. Generate posterior samples from $p(\boldsymbol{\omega}, \tau^2, \eta|\boldsymbol{y}_n)$ or $p(\boldsymbol{\omega}, \eta|\boldsymbol{y}_n)$, depending on which prior is used for $\boldsymbol{\beta}$, and then select the mode.
2. Based on the posterior samples of $(\boldsymbol{\omega}, \tau^2, \eta)$, generate the conditional posterior distribution for $\boldsymbol{\beta}$. If non-informative prior is used, generate the conditional distribution of $\tau^2|\boldsymbol{\omega}, \eta$.
3. Based on the posterior samples of $(\boldsymbol{\omega}, \tau^2, \eta)$ (or $(\boldsymbol{\omega}, \eta)$ for non-informative prior), generate the posterior predictive distribution of any query point $y(\boldsymbol{x})|\boldsymbol{y}_n, \boldsymbol{\omega}, \tau^2, \eta$. For non-informative prior, the posterior predictive distribution $y(\boldsymbol{x})|\boldsymbol{y}_n, \boldsymbol{\omega}, \eta$ is the same except τ^2 is replaced by $\hat{\tau}^2$.

Outline

Preliminary

We start with the continuous formulation.

- denote the posterior, either (6) or (8) to be ρ^*
- define dynamic flow map $\boldsymbol{x} = \phi(\boldsymbol{z}, t) : \mathbb{R}^d \to \mathbb{R}^d$, smooth and one-to-one.
  - It is designed to transform an initial pdf $\rho^0(\boldsymbol{z})$ to $\rho^*(\boldsymbol{x})$ as $t \to \infty$
  - velocity: $\dot{\phi}(z, t) := \boldsymbol{u}(\phi(z, t), t)$, $z \in \mathbb{R}^d$, $t > 0$
  - transportation equation: $\dot{\rho} + \nabla \cdot (\rho \boldsymbol{u}) = 0$, $\rho(\boldsymbol{x}, 0) = \rho^0(\boldsymbol{x})$ and $\rho(\boldsymbol{x}, \infty) = \rho^*(\boldsymbol{x})$
- the energy functional $\mathcal{F}[\boldsymbol{\phi}_t]$ is the KL divergence of ρ and ρ^*: $\mathcal{F}[\boldsymbol{\phi}_t] = \int \rho(\boldsymbol{x}) \ln\left(\frac{\rho}{\rho^*}\right) \mathrm{d}\boldsymbol{x}$. And we expect $\mathcal{F}[\boldsymbol{\phi}_t] \to 0$ as $t \to \infty$.

Energy Dissipation Law

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\boldsymbol{\phi}_t] = -\triangle(\boldsymbol{\phi}_t, \dot{\boldsymbol{\phi}}_t), \tag{12}$$

The $\triangle(\boldsymbol{\phi}_t, \dot{\boldsymbol{\phi}}_t)$ represents the rate of energy dissipation,

- $\triangle(\boldsymbol{\phi}_t, \dot{\boldsymbol{\phi}}_t) \geq 0$ so that $\mathcal{F}(\boldsymbol{\phi}_t)$ decreases with time
- a simple quadratic functional in terms of $\dot{\boldsymbol{\phi}}_t$

$$\triangle(\boldsymbol{\phi}_t, \dot{\boldsymbol{\phi}}_t) = \int_{\Omega_t} \rho_{[\boldsymbol{\phi}_t]} \|\dot{\boldsymbol{\phi}}_t\|_2^2 \,\mathrm{d}\boldsymbol{x},$$

where

- $\rho_{[\boldsymbol{\phi}_t]}$ denotes the pdf of the distribution at t
- Ω_t is the current support
- $\|\boldsymbol{a}\|_2 = \boldsymbol{a}^\top \boldsymbol{a}$, $\forall \boldsymbol{a} \in \mathbb{R}^d$.

Solving, step 1

As stated in *Wang, Y., Chen, J., Liu, C. et al., 2021*

1. perform discretization on space parameter, which is called the "approximation-then-variation" approach and obtain the particle version of the PDE.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}_h(\{\boldsymbol{x}_i(t)\}_{i=1}^N) = -\triangle_h(\{\boldsymbol{x}_i(t)\}_{i=1}^N, \{\dot{\boldsymbol{x}}_i(t)\}_{i=1}^N). \quad (13)$$

Here $\boldsymbol{x}_i(t) = \phi(\boldsymbol{x}_i(0), t)$ where $\boldsymbol{x}_i(0)$ is sampled from the initial distribution ρ^0, and

$$\mathcal{F}_h(\boldsymbol{x}_i) = \frac{1}{N}\sum_{i=1}^N \left(\ln\left(\frac{1}{N}\sum_{j=1}^N K_h(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) + V(\boldsymbol{x}_i)\right)$$

where $K_h(\boldsymbol{x}_i, \boldsymbol{x}_j) \propto \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/h^2)$, h is kernel bandwidth and $V = -\log\rho^*$.

Solving, step 2

2. Using the energetic variational approach, by taking variation of $\mathcal{F}_h$ with respect to $\boldsymbol{x}_t$, and $\frac{1}{2}\Delta_h$ with respect to $\dot{\boldsymbol{x}}_t$,

$$\frac{\delta \frac{1}{2}\triangle_h}{\delta \dot{\boldsymbol{x}}_i(t)} = -\frac{\delta \mathcal{F}_h}{\delta \boldsymbol{x}_i}, \quad \text{for } i = 1, \ldots, N. \tag{14}$$

Using above quadratic form dissipation, we have

$$\triangle_h = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{u}\|^2$$

Then assuming $\boldsymbol{u}(\boldsymbol{x}_i(t), t) \approx \dot{x}_i(t)$, we have

$$\frac{\delta \frac{1}{2}\triangle_h}{\delta \dot{\boldsymbol{x}}_i(t)} = \frac{\dot{\boldsymbol{x}}_i(t)}{N}$$

Solving, step 3

3 using implicit Euler for time discretization, the numerical scheme is

$$\frac{1}{N}\frac{\boldsymbol{x}_i^{n+1}-\boldsymbol{x}_i^n}{\tau} = -\frac{\delta \mathcal{F}_h}{\delta \boldsymbol{x}_i}\left(\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N\right)$$

Using Proximal Point Algorithm, we need to solve

$$\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N = \arg\min_{\{\boldsymbol{x}_i\}_{i=1}^N} J(\{\boldsymbol{x}_i\}_{i=1}^N)$$

- $J(\{\boldsymbol{x}_i\}_{i=1}^N) := \frac{1}{2\tau}\sum_{i=1}^N \|\boldsymbol{x}_i - \boldsymbol{x}_i^n\|^2/N + \mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$
- $\{\boldsymbol{x}(t)\}_{i=1}^N$ is the locations of particles at time t
- $\dot{\boldsymbol{x}}_i(t)$ is the derivative of $\boldsymbol{x}_i$ with t
- subscript h is the bandwidth parameter of the kernel

EVI as sampling method road map

$\left\{\left(\boldsymbol{x}_0^i\right)\right\}_{i=1}^N$
initial particles from initial distribution (uniform, normal, etc.)

↓

implicit Euler updates

↓

particles approximate target densities that minimize $\mathcal{F}_h$
the mode maximize posterior (6) or (8)

About the diffusion term

With KL divergence as the energy functional, EnVarA gives

$$\mathcal{F}[\phi_t] = \int \rho \ln \rho + \rho V \, \mathrm{d}\boldsymbol{x} \implies \rho \boldsymbol{u} = -(\nabla \rho + \rho \nabla V)$$

Combine with the transportation equation we have

$$\dot{\rho} = \nabla \cdot (\nabla \rho + \rho \nabla V)$$

We can set the diffusion term to 0. And this correspond to use $\mathcal{F}[\phi_t] = \int \rho V \, \mathrm{d}\boldsymbol{x}$ and thus

$$\mathcal{F}_h(\boldsymbol{x}_i) = V(\boldsymbol{x}_i) = -\log \rho^*(\boldsymbol{x}_i)$$

In this case, the scheme ends up a simple gradient flow that should converges to the local maximum points of $\rho^*(\boldsymbol{x}_i)$.

Two variants of EVI-GP

- EVI-Post: Generate N particles and they will approximate the posterior samples through implicit Euler updates
- EVI-MAP: This variant corresponds to Maximum A Posteriori (MAP), and is thus named EVI-MAP. Simply set $\mathcal{F}(\boldsymbol{x}) = V(\boldsymbol{x})$ and $N = 1$. This is equivalent to proximal point algorithm.

Outline

Performance Metric

- Performance is measured by the **Standardized Root Mean Squared Predictive Error (RMSPE)**.

$$\text{standardized RMSPE} = \left(\sqrt{\frac{1}{N}\sum_i (\hat{y}_i - y_i)^2}\right) \Big/ \text{std. dev.}(y_i)$$

- $\hat{y}_i$ is the predicted value at x_i
- y_i is the true value at x_i
- $\sigma(y_i)$ is the standard deviation of all y_i
- The closer the RMSPE is to **0**, the better the model performance.
- Our method (EVIGP) is compared against the R packages: **gpfit**, **mlegp**, and **laGP**.

Example 1: $x\sin(x)$ (Setup)

- **Data**: training data size $n_{train} = 11$, testing data size $n_{test} = 100$, from latin hypercube sampling on $[0, 1]$ and then scaled to $[0, 10]$
- **Noise**: $\epsilon \sim \mathcal{N}(0, 0.5^2)$
- **Priors**: $a_\omega = a_\eta = 1$, $b_\omega = b_\eta = 5$.
- **EVI Settings**: Time step $h = 0.02$, scale $\tau = 1$. Initially, 100 EVI particle uniformly distributed at $[0, 0.1] \times [0.1, 0.4]$
- **Solver**: L-BFGS (PyTorch) with history size 50, max inner iterations 100, and max epoch 500.

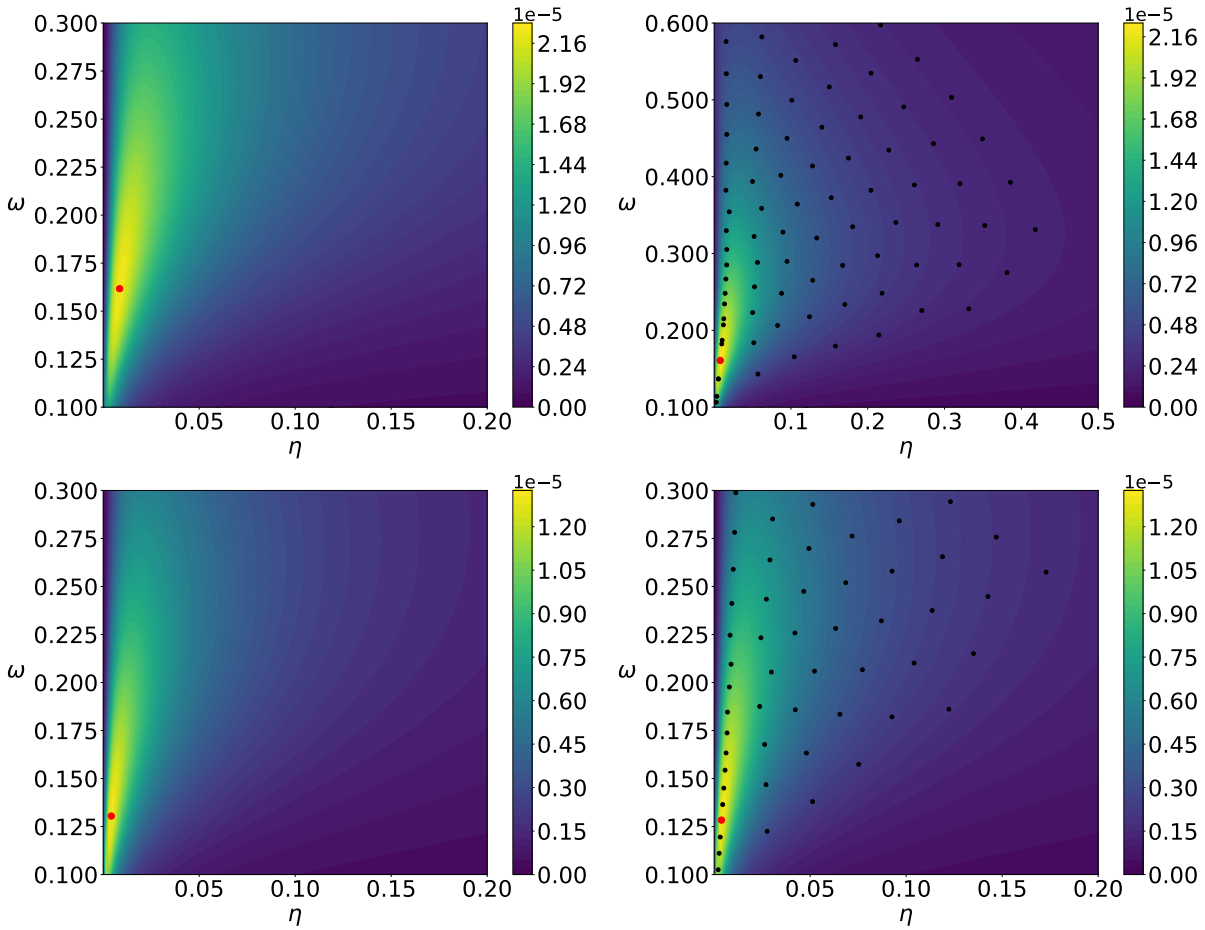


Figure: Contour plot of $x\sin(x)$ case, where left two using EVI-MAP and right two using EVI-post, top two are from constant mean model and bottom two are from linear mean model. Red points are those with the highest posterior

Example 1: $x\sin(x)$ (Result)

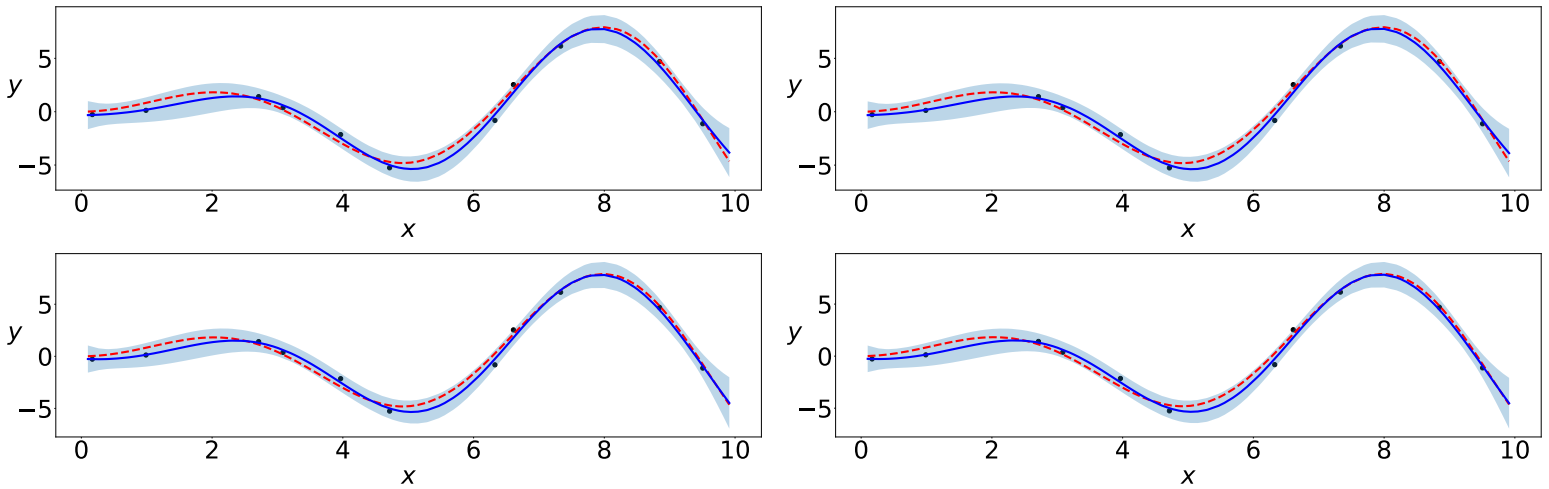


Figure: The gaussian process prediction. Black dots are training data, red curve is the true $x\sin(x)$ and blue curve is the predicted, light blue area shows the predictive condifence interval

Example 1: $x\sin(x)$ (Comparison)

The test was conducted for 1000 times, and the mean of RMSPE of all methods tested are listed below, where the column for R packages only shows the best among them.

Type of mean	EVIGP	EVIGP with 0 diffusion	gpfit/mlegp/lagp
Constant	0.1318	0.1310	0.1275
Linear	0.1211	0.1195	0.1583

Table: standardized RMSPE, $x\sin(x)$ case

Example 1: $x\sin(x)$ (Robustness)

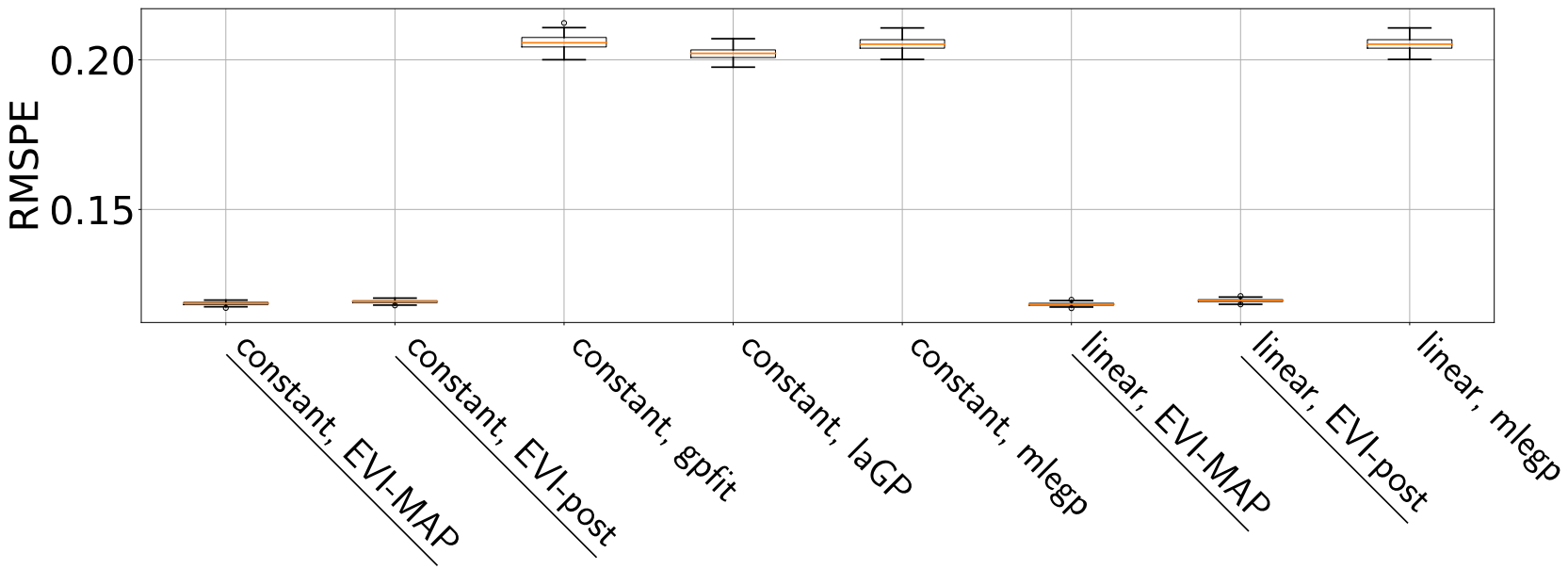


Figure: The mean standardized RMSPE from all different cases and methods as indicated in the label.

Example 2: OTL Circuit (Setup)

- **Function**: A 6-dimensional electronic circuit function.

$$V_m(\boldsymbol{x}) = \frac{(V_{b1}+0.74)\beta(R_{c2}+9)}{\beta(R_{c2}+9)+R_f} + \frac{11.35R_f}{\beta(R_{c2}+9)+R_f} + \frac{0.74R_f\beta(R_{c2}+9)}{(\beta(R_{c2}+9)+R_f)R_{c1}}, V_{b1} = \frac{12R_{b2}}{R_{b1}+R_{b2}}$$

- **Data**: $n_{\text{train}} = 200$, $n_{\text{test}} = 1000$.
- **Noise**: $\epsilon \sim \mathcal{N}(0, 0.02^2)$.
- **Priors**: $a_\omega = a_\eta = 1.1$, $b_\omega = b_\eta = 2.5$.
- **EVI Settings**: Time step $h = 0.001$, scale $\tau = 1$
- **Informative Prior**: A 5-fold cross-validation was used to select a shrinkage parameter $\nu = 2.65$ for variable selection.
- **Solver**: L-BFGS (PyTorch) with history size 50, max inner iterations 100, and max epoch 500

informative prior and shirinkage effect

With an informative prior, we first use cross-validation to choose the parameter ν and then select parameters based on their significance.

- $\boldsymbol{R}$ is a *diagonal matrix*. And on the diagonal, the values are
  - 1, corresponds to the intercept term in $\boldsymbol{g}$
  - $r = 4/3$, linear effect
  - r^2, quadratic effect,
- ν is searched in a evenly spaced grid from 0 to 5 with 0.05 grid size. A 5-fold cross validation is used to find ν with the smallest standardized RMSPE. The final result is $\nu = 2.65$.

To better illustrate, we rename $R_{b1}, R_{b2}, R_f, R_{c1}, R_{c2}, \beta$ to be $x_1, x_2, \ldots, x_6$, respectively.

Example 2: OTL circuit (Comparison)

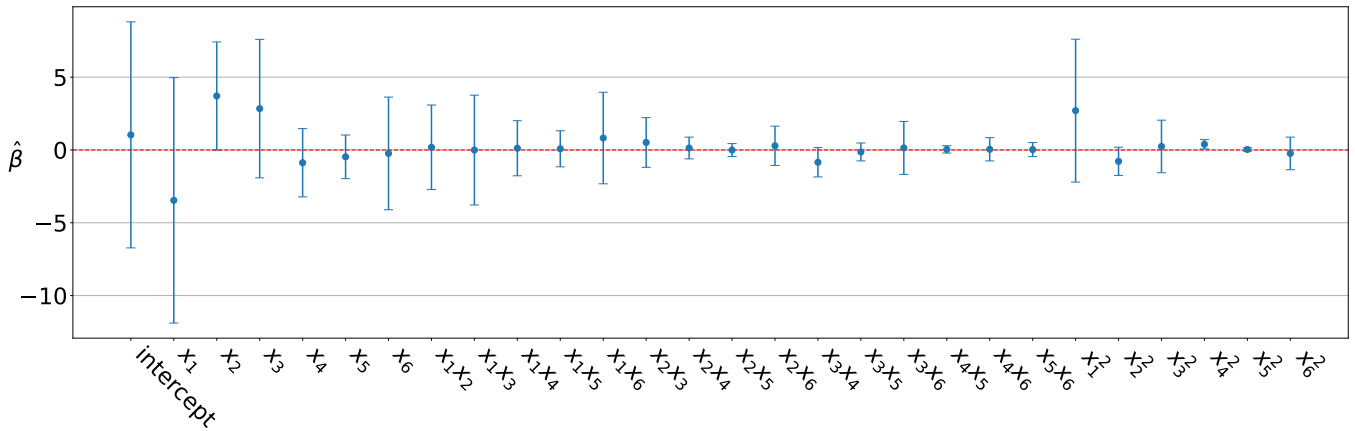
The mean of RMSPE of all methods tested are listed below.

Type of mean	EVIGP	mlegp/lagp
Constant	0.02207	0.04892
Linear	0.01401	0.03593
Quadratic	0.02181	0.03012
Quadratic, after selection	0.01927	N/A

Table: standardized RMSPE

Example 2: OTL circuit (Significance of effects)

We kept only $x_2, x_3, x_4, x_3x_4, x_2^2$ and regress the data again.



Example 2: OTL circuit (Robustness)

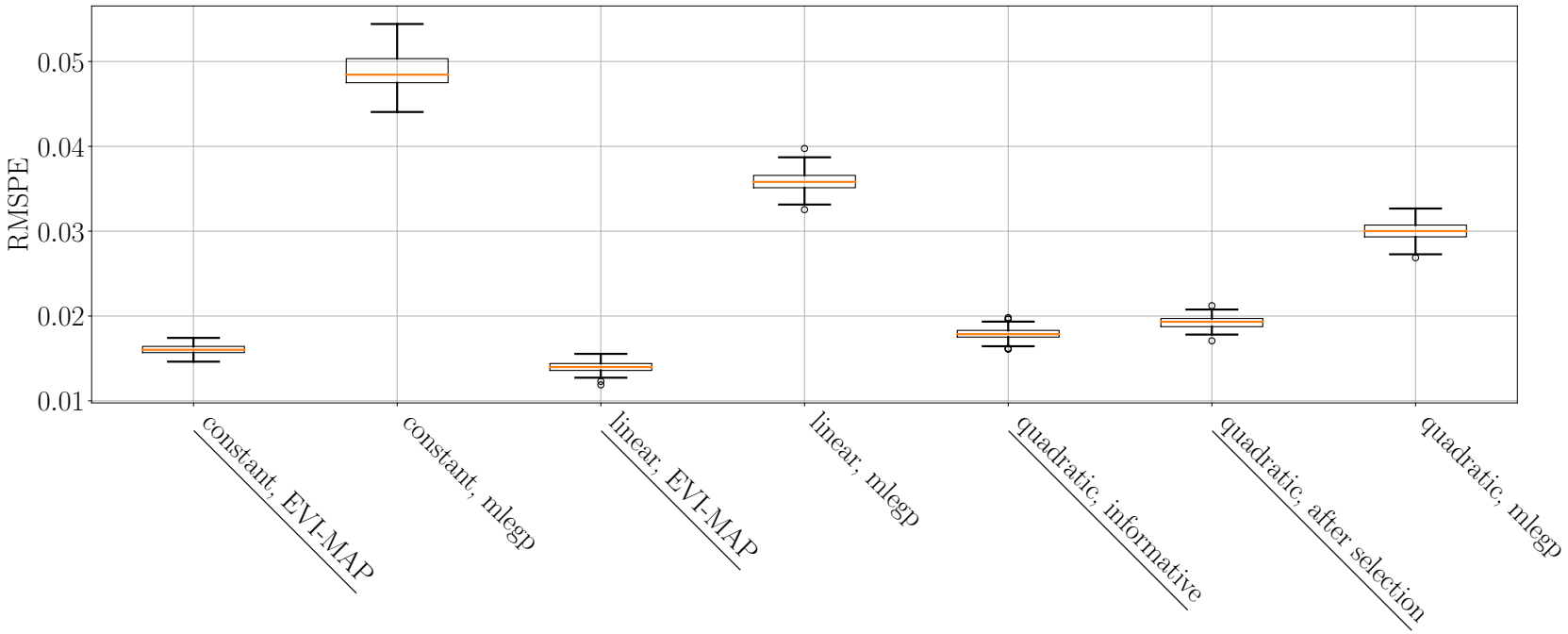


Figure: The mean standardized RMSPE from all different cases and methods as indicated in the label.

Example 3: Borehole Function (Setup)

- **Problem**: An 8-dimensional function modeling water flow through a borehole.

$$f(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_\omega)\left(1 + \frac{2LT_u}{\ln(r/r_\omega) r_\omega^2 K_\omega} + \frac{T_u}{T_l}\right)}$$

- **Priors**: $a_\omega = a_\eta = 1$, $b_\omega = b_\eta = 4$.
- **Informative Prior**: Cross-validation resulted in shrinkage parameter $\nu = 0.95$.
- variables are also renamed as $x_1, x_2, \ldots, x_8$.
- The rest settings are inherited from the OTL example.

Example 3: Borehole Function (Comparison)

The mean of RMSPE of all methods tested are listed below.

Type of mean	EVIGP	mlegp/lagp
Constant	0.0108	0.3349
Linear	0.04190	0.1808
Quadratic	0.03182	0.2022
Quadratic, after selection	0.02310	N/A

Table: standardized RMSPE

Example 3: Borehole Function (Robustness)

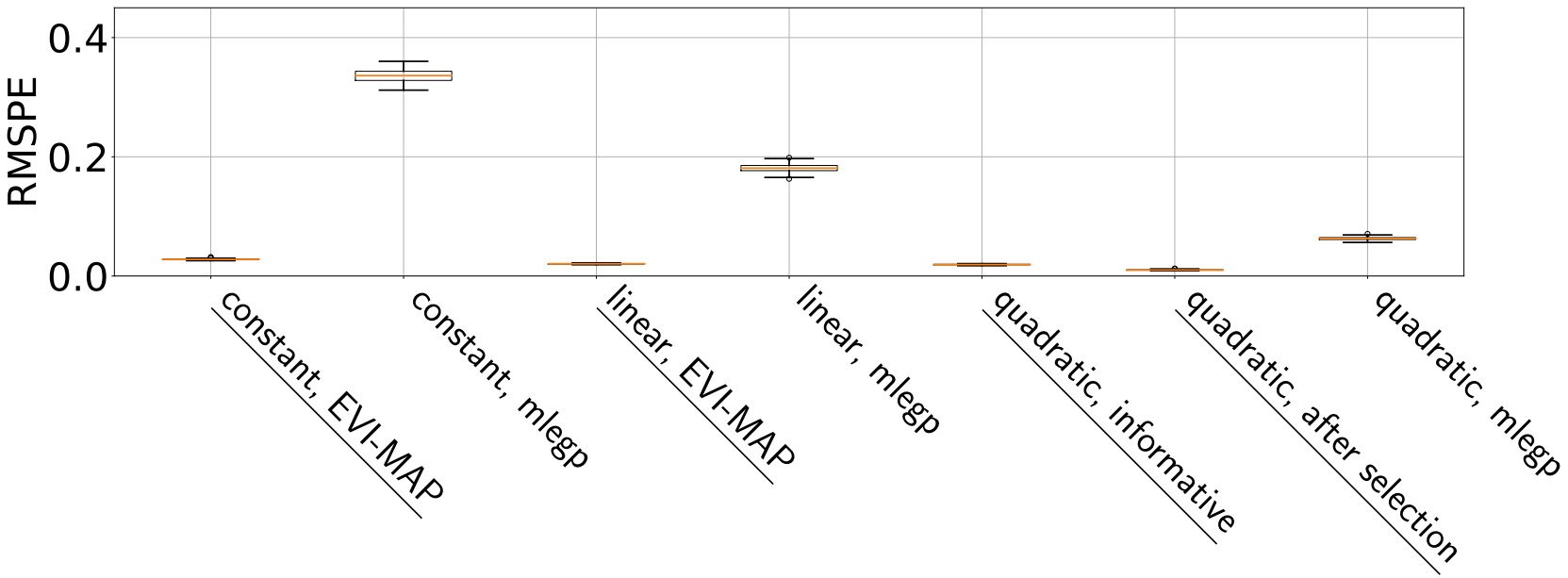


Figure: The mean standardized RMSPE from all different cases and methods as indicated in the label.

Outline

Manifold Gaussian Process

The manifold Gaussian Process (mGP) framework decomposes the regression task into: $F = G \circ M$

- $M : \mathcal{X} \to \mathcal{H}$ is a deterministic mapping, $\mathcal{H} \subset \mathbb{R}^Q$ latent feature space
- $G : \mathcal{H} \to \mathcal{Y}$ is a GPR model on $\mathcal{H}$

Specifically,

$$\boldsymbol{Y}_n \sim \mathcal{N}\big(0,\, \tau^2(\tilde{\boldsymbol{K}}_n + \eta \boldsymbol{I}_n)\big),$$

where $\tilde{K}_{n,ij} = k(M(\boldsymbol{x}_i), M(\boldsymbol{x}_j))$. Mapping M can be parameterized via a neural net with parameters θ_M, optimized jointly with GP hyperparameters.

Posterior distribution

Correspondingly,

$$(y(\boldsymbol{x})|\boldsymbol{y}, \tau^2, \eta, \boldsymbol{\theta}) = \tilde{\boldsymbol{k}}_n(\boldsymbol{x})'(\tilde{\boldsymbol{K}}_n + \hat{\eta}\boldsymbol{I}_n)^{-1}\boldsymbol{y}, \tag{15}$$

$$s_n^2(\boldsymbol{x}) \triangleq \mathrm{Var}\left[y(\boldsymbol{x})|\boldsymbol{y}, \tau^2, \eta, \boldsymbol{\theta}\right] \tag{16}$$

$$= \hat{\tau}^2\left[1 - \tilde{\boldsymbol{k}}(\boldsymbol{x})'(\tilde{\boldsymbol{K}}_n + \hat{\eta}\boldsymbol{I}_n)^{-1}\tilde{\boldsymbol{k}}(\boldsymbol{x})\right] + \hat{\sigma}^2, \tag{17}$$

And the negative log likelihood is:

$$\mathsf{NLML}(\boldsymbol{\theta}_{\mathsf{mGP}}) = n\log\tau^2 + \log\det(\tilde{\boldsymbol{K}}_n + \eta\boldsymbol{I}_n) + \frac{1}{\tau^2}\boldsymbol{y}^\top(\tilde{\boldsymbol{K}}_n + \eta\boldsymbol{I}_n)^{-1}\boldsymbol{y},$$

Neural Network Architecture

In each layer, the transformation is defined as

$$\boldsymbol{Z}_i = T_i(\boldsymbol{X}) = \sigma_M(\boldsymbol{W}_i \boldsymbol{Z}_{i-1} + \boldsymbol{B}_i)$$

- σ_M is the activation function
- $\boldsymbol{W}_i$ and $\boldsymbol{B}_i$ are the weight matrix and bias vector for the i-th layer

$$M(\boldsymbol{X}) = T_l \circ T_{l-1} \circ \cdots \circ T_1(\boldsymbol{Z}_0)$$

with $\boldsymbol{Z}_0 = \boldsymbol{X}$.

A batch normalizing layer is also applied so that kernel values have

- kernel values
- well-behaved gradients

Activation function

Log-sigmoid:

$$\text{log-}\sigma(x) := \log\left(\frac{1}{1+e^{-x}}\right) = -\log(1+e^{-x}).$$

Its derivative:

$$\frac{d}{dx}\,\text{log-}\sigma(x) = \frac{1}{1+e^{x}} = \sigma(-x),$$

- smoother transition
- not magnitude explosion

A combined loss

- reconstruction loss from the autoencoder architecture
- L_2 regularization

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \text{NLML} + \alpha\mathcal{L}_{\text{reconstruction}} + \beta\mathcal{L}_{\text{regularization}} \\ &= \text{NLML} + \frac{\alpha}{N}\sum_{i=1}^{N} \|M_{\text{decoder}}(M_{\text{encoder}}(\boldsymbol{x}_i)) - \boldsymbol{x}_i\|^2 \\ &\quad + \beta\sum_{l} \|\boldsymbol{W}_l\|_F^2.\end{aligned}$$

Outline

Integrated Mean-Squared Error

Active Learning Cohn selects new data points $\boldsymbol{x}_{n+1}$ that minimize the model's integrated mean-squared error (IMSE) over the input space.

$$\breve{s}^2_{n+1}(\boldsymbol{x}|\boldsymbol{x}_{n+1}) \triangleq \hat{\tau}^2 \left[1 - \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})^\top (\tilde{\boldsymbol{K}}_{n+1} + \eta \boldsymbol{I}_{n+1})^{-1} \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})\right],$$

$$\text{IMSE}(\boldsymbol{x}_{n+1}) \triangleq \int_{\mathcal{X}} \breve{s}^2_{n+1}(\boldsymbol{x}|\boldsymbol{x}_{n+1})\, \mathrm{d}\boldsymbol{x},$$

- $\boldsymbol{x}_{n+1}$ is selected from a finite candidate pool $\mathcal{X}_{\text{cand}} \subset \mathcal{X}$.
- $\tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x}) = [\tilde{k}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \tilde{k}(\boldsymbol{x}, \boldsymbol{x}_{n+1})]^\top$ $\tilde{\boldsymbol{K}}_{n+1}$ denote the kernel vector and $(n+1) \times (n+1)$ correlation matrix evaluated in the learned latent space $\mathcal{H} = M(\mathcal{X})$ using the current mGP parameters.

Active Learning Cohn

Minimizing the IMSE is equivalent to maximizing the expected reduction in posterior variance, thus

$$\begin{aligned}\text{ALC}(\boldsymbol{x}_{n+1}) &= \int_{\mathcal{X}} s_n^2(\boldsymbol{x}) - \breve{s}_{n+1}^2(\boldsymbol{x}|\boldsymbol{x}_{n+1})\,\mathrm{d}\boldsymbol{x} \\ &\propto \int_{\mathcal{X}} \hat{\tau}^2 \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})^\top (\tilde{\boldsymbol{K}}_{n+1} + \eta \boldsymbol{I}_{n+1})^{-1} \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},\end{aligned}$$

where $s_n^2(\boldsymbol{x})$ is computed via (16) except that k function is replaced by $\tilde{k}$, and thus $\boldsymbol{k}_n(\boldsymbol{x})$ by $\tilde{\boldsymbol{k}}_n(\boldsymbol{x})$ and $\boldsymbol{K}_n$ by $\tilde{\boldsymbol{K}}_n$.

ALC Approximation

Numerically, we introduce a fixed reference set $\mathcal{X}_{\text{ref}}$ of size m sampled from $\mathcal{X}$ using Latin Hypercube Design. The ALC objective becomes:

$$\text{ALC}(\boldsymbol{x}_{n+1}) \propto \sum_{\boldsymbol{x} \in \mathcal{X}_{\text{ref}}} \hat{\tau}^2 \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})^\top (\tilde{\boldsymbol{K}}_{n+1} + \eta \boldsymbol{I}_{n+1})^{-1} \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x}).$$

At each active learning iteration, the most informative training point is selected via:

$$\boldsymbol{x}^*_{n+1} = \underset{\boldsymbol{x} \in \mathcal{X}_{\text{cand}}}{\arg\max}\, \text{ALC}(\boldsymbol{x}),$$

and subsequently removed from the candidate pool $\mathcal{X}_{\text{cand}}$.

Improvement

- batch acquisition: acquire B design points per iteration
- pre-screening: shrinking the candidate set, leaving only those with high predictive variance

1. among the candidate set $\mathcal{X}_{\text{cand}}$, we identify the top $K > B$ points with the highest predictive variance, denoted $\mathcal{X}_r^*$:

$$\mathcal{X}_{\text{cand}}^{(r)} = \left\{ \boldsymbol{x}_1^{(r)}, \boldsymbol{x}_2^{(r)}, \ldots, \boldsymbol{x}_K^{(r)} \right\},$$

2. compute ALC on $\mathcal{X}_{\text{cand}}^{(r)}$.
3. select the B points with the highest ALC scores

Algorithm: ALmGP (Part 1: Setup)

Algorithm 1: Active Learning for Manifold Gaussian Process

1: **Input:** Initial training dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n_0}$; reference set $\mathcal{X}_{\text{ref}}$ and candidate set $\mathcal{X}_{\text{cand}}$; tuning parameters $K, B, \texttt{Tol}, N_{\max}$; and initial mGP settings.
2: **Output:** Trained mGP model and estimated parameters.
3: **Step 0:** Fit the mGP model using the initial training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n_0}$.

Algorithm: ALmGP (Part 2: Main Loop)

Algorithm 1: Active Learning for Manifold Gaussian Process (Cont.)

4: **for** $r = 1, 2, \ldots, \lfloor N_{\max}/B \rfloor$ **do**
5: **Step 1:** Screen candidates: Select the top K points with the highest posterior variance from $\mathcal{X}_{\text{cand}}$.
6: **Step 2:** Score candidates: Compute the ALC acquisition score for each screened point.
7: **Step 3:** Acquire new data: Select top B points with the highest ALC scores and add to the training set.
8: **Step 4:** Update model: Re-train or update the mGP model with the expanded training set.
9: **Step 5:** Check convergence: If error is below `Tol`, terminate.
10: **end for**

Outline

Experimental Setup

- **Optimizer**: L-BFGS with strong Wolfe line search from PyTorch.
- **Stability**: Model hyperparameters are squared to ensure positivity.
- **Metric**: Root Mean Square Error (RMSE) on an independent test set.

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{test}}} \sum_i (\hat{y}_i - y_i)^2}$$

- **Robustness**: Each experiment is repeated 10 times.
- **Baseline**: Performance is compared against random point acquisition.
- **Data generation**: Latin Hypercube Design

Example 1: Piecewise Trigonometric Function (Setup)

- **Function**: A 1D function with three distinct regions and Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$.

$$F(x) = \begin{cases} 1.35\cos(12\pi x), & x \in [0, 0.33] \\ 1.35, & x \in [0.33, 0.66] \\ 1.35\cos(6\pi x), & x \in [0.66, 1] \end{cases}$$

- **Parameters**:
  - Initial data: $n_0 = 10$
  - Budget: $N_{\max} = 20$ (with batch size $B = 1$)
  - Neural Network: [1-6-2] architecture

Example 1: Piecewise Trigonometric Function (Results)

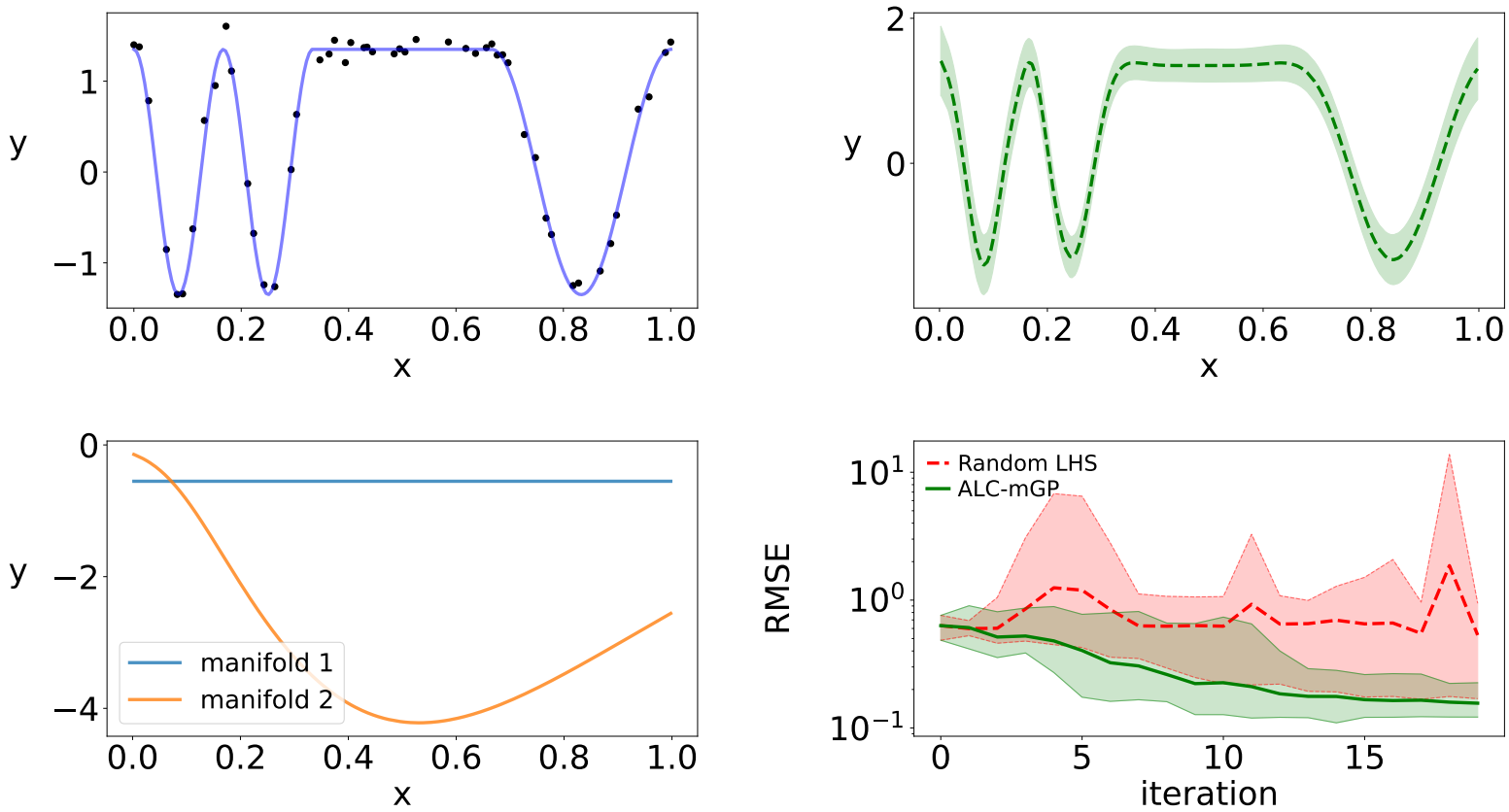


Figure: Final selected data, model prediction, learned latent space, and RMSE comparison against random sampling. The final average RMSE was **0.156**.

Example 2: 2D Deterministic Function (Setup)

- **Function**: A 2D function composed of Gaussian PDFs and a linear term, rotated by $45°$.

$$f(x_1, x_2) = 1 - \phi(x_2; 3, 0.5^2) - \phi(x_2; -3, 0.5^2) + \frac{x_1}{100}$$

- **Parameters**:
  - Initial data: $n_0 = 50$
  - Budget: $N_{\max} = 50$ (with batch size $B = 1$)
  - Neural Network: $[2\text{-}10\text{-}3]$ architecture

Example 2: 2D Deterministic Function (Results)

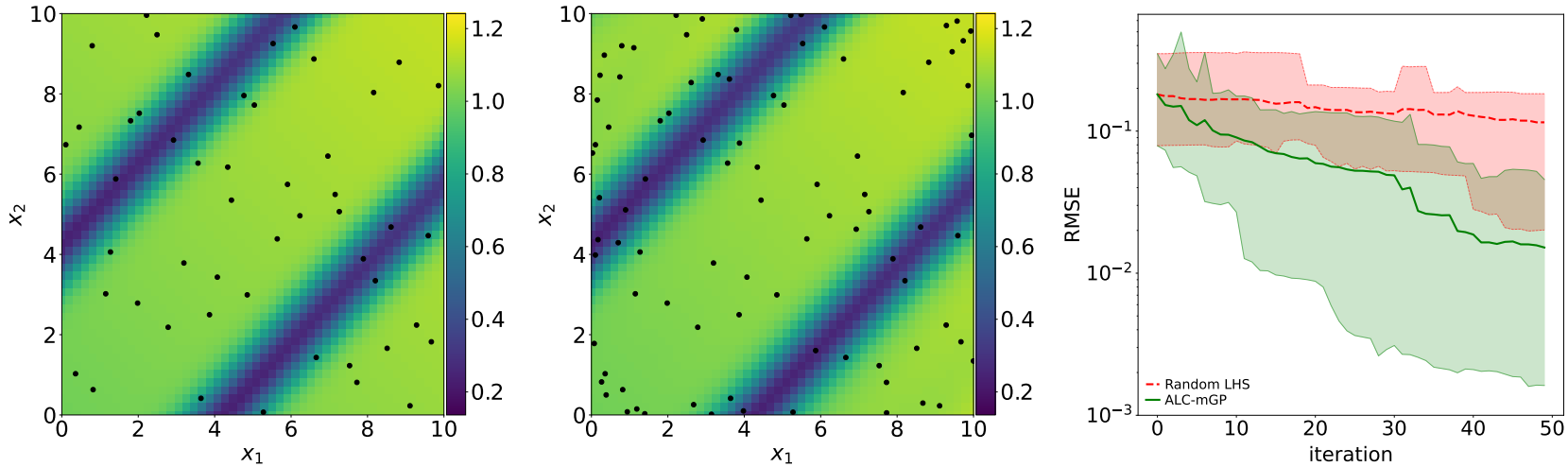


Figure: True function heatmap, final prediction heatmap, and RMSE comparison.

The final average RMSE was **0.0152**.

Example 3: Function on the 3D Sphere (Setup)

- **Function**: A function defined on the surface of a 3D unit sphere ($x^2 + y^2 + z^2 = 1$).

$$f(x, y, z) = \cos(x) + y^2 + e^z$$

- **Parameters**:
  - Initial data: $n_0 = 50$
  - Budget: $N_{\max} = 100$ (with batch size $B = 1$)
  - Neural Network: $[3\text{-}10\text{-}2]$ architecture

Example 3: Function on the 3D Sphere (Results)

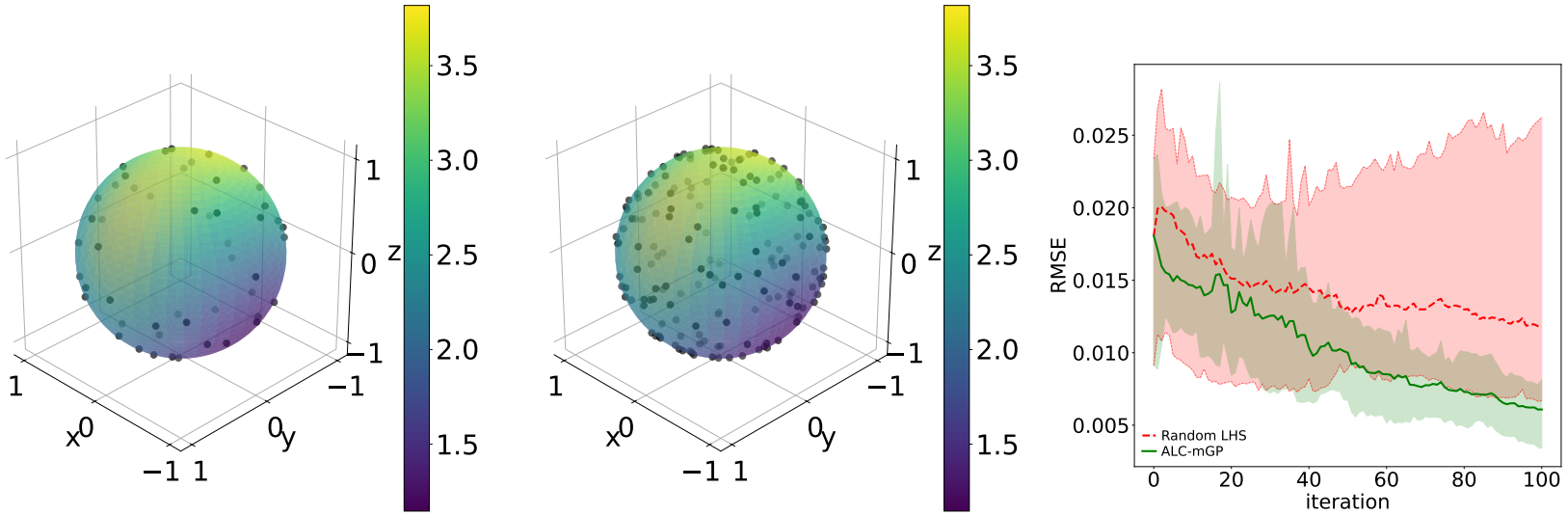


Figure: True function on the sphere, final prediction, and RMSE comparison.

The final average RMSE was $\mathbf{6 \times 10^{-3}}$.

Example 4: 8D Borehole Function (Setup)

- **Function**: A well-known 8-dimensional benchmark function that models groundwater flow through a borehole.
- **Challenge**: A complex, high-dimensional, and nonlinear regression problem.
- **Parameters**:
  - Initial data: $n_0 = 50$
  - Budget: $N_{\max} = 150$ (with batch size $B = 1$)
  - Neural Network: $[8\text{-}30\text{-}4]$ architecture

Example 4: 8D Borehole Function (Results)

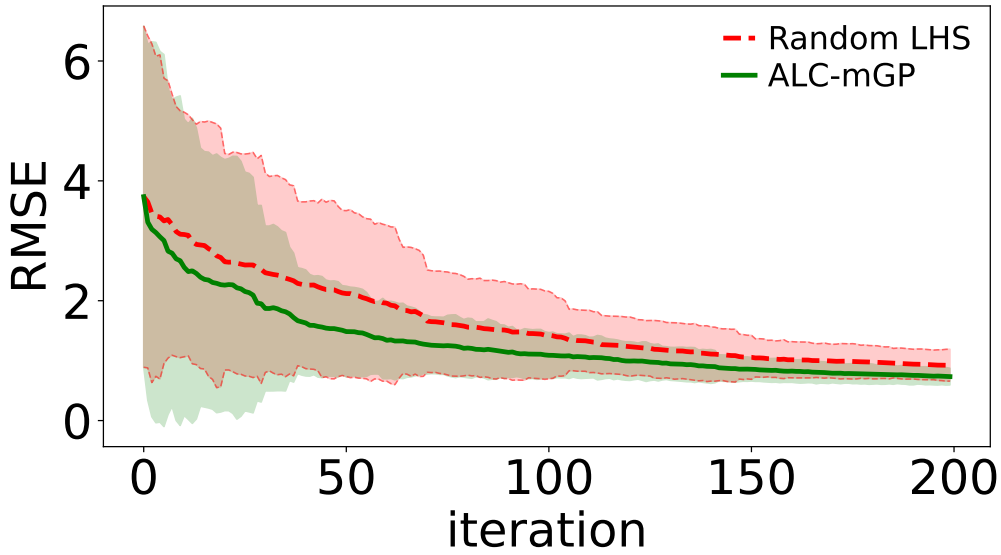


Figure: Test RMSE comparison for the 8D Borehole function.

The final average RMSE was **0.605**.

Discussion & Limitations

This work sought to extend Gaussian Process (GP) regression to complex systems with high dimensionality and scarce data.

- **EVIGP Framework:** We introduced an energetic variational inference (EVI) approach that uses particle-based dynamics for stable posterior approximation and variable selection.
- **ALMGP Framework:** We developed a manifold-based active learning strategy using an autoencoder to perform GP regression in a learned low-dimensional space.

Its limitations

- **EVI Sensitivity:** The EVIGP framework is sensitive to the setting of mean function.
- **Manifold Quality:** The quality of the learned latent space depends heavily on the autoencoder's stability and expressiveness, which can risk distorting important features.

Conclusion & Future Work

The methods developed in this dissertation make GP modeling more adaptive to structural constraints and more scalable for scientific surrogate modeling. For future research directions:

- **For EVI:** Future research can explore adaptive kernels, hierarchical priors, and connections to Wasserstein gradient flows to improve flexibility.
- **For Active Learning:** We can integrate manifold regularization into the autoencoder or develop an end-to-end framework that co-optimizes for uncertainty and geometry.
- **Unified System:** A powerful next step is to combine both contributions into a single system that dynamically evolves its inference and acquisition strategies, unlocking new applications in autonomous experimentation.

Acknowledgements

I would like to express my sincere gratitude to the following individuals and institutions for their invaluable support throughout my Ph.D. journey:

- **My Advisors:** Dr. Lulu Kang and Dr. Chun Liu, for their exceptional guidance, mentorship, and unwavering support.
- **My Committee:** Thank you to the members of my dissertation committee for their insightful feedback and encouragement.
- **IIT, OSG:** For providing the resourcesthat made this research possible
- **Collaborators & Colleagues:** A special thanks to my peers and collaborators for the stimulating discussions and support.
- **Family & Friends:** Finally, I am deeply grateful to my family and friends for their constant love and encouragement.

Thank You

Questions?