Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

ACTIVE LEARNING FOR MANIFOLD GAUSSIAN PROCESS REGRESSION

Yuanxing Cheng[1], Lulu Kang[2], Yiwei Wang[3], and Chun Liu[1]

[1]Dept. of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA
[1]Dept. of Mathematics and Statistics, University of Massachusetts, Amherst, MA, USA
[3]Dept. of Mathematics, University of California, Riverside, CA, USA

ABSTRACT

This paper introduces an active learning framework for manifold Gaussian Process (GP) regression, combining manifold learning with strategic data selection to improve accuracy in high-dimensional spaces. Our method jointly optimizes a neural network for dimensionality reduction and a Gaussian process regressor in the latent space, supervised by an active learning criterion that minimizes global prediction error. Experiments on synthetic data demonstrate superior performance over randomly sequential learning. The framework efficiently handles complex, discontinuous functions while preserving computational tractability, offering practical value for scientific and engineering applications. Future work will focus on scalability and uncertainty-aware manifold learning.

1 INTRODUCTION

Gaussian Process Regression (GPR) (Rasmussen and Williams 2006) is a powerful non-parametric framework widely used in machine learning, statistics, and scientific computing for regression and uncertainty quantification tasks. However, standard GPR methods face significant challenges as the input dimension grows, suffering from the *curse of dimensionality*: computational scalability deteriorates, and the ability to capture the underlying data structure diminishes as data becomes increasingly sparse in high-dimensional spaces (Beyer et al. 1999).

To address these limitations, *manifold Gaussian Processes* (mGPs) have emerged as a promising alternative. By leveraging manifold learning techniques, mGPs project high-dimensional data onto a lower-dimensional latent space while preserving its intrinsic geometry (Meilă and Zhang 2024). This approach not only mitigates the challenges of high-dimensional inference but also enhances predictive performance by exploiting the data's underlying structure. Recent advances include graph-based distance metrics (e.g., Isomap (Tenenbaum et al. 2000)) combined with Matérn Gaussian processes (Borovitskiy et al. 2020; Fichera et al. 2023), as well as wrapped Gaussian processes for manifold-valued data (Mallasto and Feragen 2018).

Despite these innovations, the performance of GPR—and by extension, mGPs—heavily depends on the quality and representativeness of the training data. In many real-world applications, acquiring labeled data is expensive or time-consuming, necessitating efficient data acquisition strategies. Active learning for GPR has been extensively studied due to the model's inherent ability to quantify predictive uncertainty, making it a natural fit for sequential data selection. Early work by MacKay (1992) laid the theoretical foundation for Bayesian active learning, later adapted to GPR by Seo et al. (2000), demonstrating superior convergence compared to passive sampling. Subsequent developments include uncertainty sampling (prioritizing high-variance points) and information-theoretic approaches, such as mutual information maximization for sensor placement (Krause et al. 2008) and Bayesian Active Learning by Disagreement (BALD) for parameter-aware selection (Houlsby et al. 2011). These methods have been extended to deep GPR models (Ma et al. 2019) and related tasks like uncertainty quantification and inverse problems (Chen et al. 2021; Heo and Sung 2025).

In practice, active learning strategies must balance *exploration* (reducing global uncertainty) and *exploitation* (optimizing a downstream objective). This trade-off is well-studied in Bayesian optimization (BO) (Frazier 2018), where acquisition functions like Expected Improvement (EI) and Upper Confidence Bound (GP-UCB) (Srinivas et al. 2010) align naturally with active learning objectives. For instance, GP-UCB provides theoretical regret bounds, making it suitable for bandit-style optimization tasks. Cost-sensitive active learning methods further refine this balance by incorporating labeling costs into the acquisition function (Kapoor et al. 2007), which is critical in domains like medical diagnosis. Recent work has also focused on robustness: Martinez-Cantin et al. (2018) combined robust regression (GPs with Student-t likelihood) with outlier diagnostics to improve reliability.

In the context of mGPs, active learning can efficiently refine the model by targeting regions of high uncertainty or high potential improvement. For example, Sauer et al. (2023) and Binois et al. (2019) adopted the *integrated mean-squared error* (IMSE) criterion—originally proposed by Cohn et al. (1996), Cohn (1996)—to minimize prediction error. Kim et al. (2024) further integrated manifold learning with GP-UCB for optimization on manifolds. However, *active learning tailored to mGPs remains underexplored*, particularly for regression tasks where latent space geometry and high-dimensional inputs interact dynamically.

This paper proposes a novel active learning framework for manifold Gaussian Process regression, bridging the gap between efficient data acquisition and dimensionality reduction. Our key contributions include:

- a unified framework integrating manifold learning with IMSE-based active learning for mGPs,
- enhanced efficiency through strategic point selection in the latent space, and
- scalability for high-dimensional data by operating on intrinsic low-dimensional manifolds.

We validate our approach through synthetic experiments, demonstrating improvements over passive sampling and standard active learning methods.

The paper is organized as follows: Section 2 reviews GPR fundamentals. Section 3 introduces our active learning method with active learning Cohn (ALC) acquisition function for mGP regression. Section 4 illustrates the framework with three case studies, and Section 5 discusses future directions. The codes are available through the GitHub link https://github.com/XavierOwen/ALC-AEGP.

2 BACKGROUND ON GPR

The Gaussian Process Regression (GPR) model assumes the following probabilistic representation for the response $y(\boldsymbol{x})$:

$$y(\boldsymbol{x}) = \mu(\boldsymbol{x}) + Z(\boldsymbol{x}) + \varepsilon,$$

where

$$Z(\boldsymbol{x}) \sim GP(0, \tau^2 k(\cdot,\cdot)), \quad \text{and } \varepsilon \overset{\text{iid}}{\sim} N(0, \sigma^2).$$

Denote $\Omega \subset \mathbb{R}^p$ as the input domain and output $y(\boldsymbol{x}) \in \mathbb{R}^1$. Here, $k(\cdot,\cdot) : \Omega \times \Omega \to \mathbb{R}_+$ is the correlation function of the process $Z(\boldsymbol{x})$. Common choices include the Matérn and Gaussian correlation functions (or SE-ARD kernels):

$$\text{Matérn: } \prod_{l=1}^{p} \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu}|x_{il} - x_{jl}|}{\theta_l} \right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu}|x_{il} - x_{jl}|}{\theta_l} \right), \quad \text{Gaussian: } \exp\left\{ -\sum_{l=1}^{p} \frac{(x_{il} - x_{jl})^2}{\theta_l} \right\},$$

where K_ν is the modified Bessel function of the second kind and $\boldsymbol{\theta} \in \mathbb{R}_+^p$ are length-scale parameters. The mean function $\mu(\boldsymbol{x})$ determines the model class:

- *Ordinary kriging*: $\mu(\boldsymbol{x})$ is an unknown constant;
- *Universal kriging*: $\mu(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}$ (linear in inputs);

- *General case*: $\mu(\boldsymbol{x})$ is an unknown combination of basis functions, selectable via stepwise forward regression (Joseph et al. 2008; Joseph and Kang 2011) or shrinkage methods (Hung 2011; Kang et al. 2024).

For simplicity, we assume $\mu(\boldsymbol{x}) = 0$ throughout this paper. The proposed methods can be extended to non-zero mean cases through response centering or trend removal in preprocessing.

Given observed data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, we need to estimate the unknown parameters that the length-scale parameters $\boldsymbol{\theta}$, the variance of $Z(\cdot)$ τ^2, and the variance of the noise σ^2. To estimate σ^2, if the training data contain replications at some if not all design points, then σ^2 can be estimated in advance using pooled sample variance. The response vector $\boldsymbol{y}$ follows the multivariate normal distribution $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \tau^2\boldsymbol{K}_n + \sigma^2\boldsymbol{I}_n)$, where $\boldsymbol{K}_n$ is the $n \times n$ correlation matrix with $\boldsymbol{K}_{n,ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{I}_n$ is the n-dim diagonal matrix.

The parameters $(\boldsymbol{\theta}, \tau^2, \sigma^2)$ can be estimated via maximum likelihood method, and the $-2\log$-likelihood of the data is

$$Q \triangleq -2\log L = n\log\tau^2 + \log\det(\boldsymbol{K}_n + \rho\boldsymbol{I}_n) + \frac{1}{\tau^2}\left(\boldsymbol{y}^\top(\boldsymbol{K}_n + \rho\boldsymbol{I}_n)^{-1}\boldsymbol{y}\right). \tag{1}$$

Here $\rho = \sigma^2/\tau^2$ is the noise-to-signal ratio. If $\rho = 0$, the GPR becomes an interpolation model which is usually used to model deterministic computer simulation outputs. If the data does not contain replications at any design point and the noise variance σ^2 is not zero, we can consider ρ as an unknown parameter and estimate it by maximizing the log-likelihood. Given the other parameter values, the MLE of τ^2 is

$$\hat{\tau}^2 = \frac{1}{n}\boldsymbol{y}^\top(\boldsymbol{K}_n + \rho\boldsymbol{I}_n)^{-1}\boldsymbol{y}$$

obtained by solving $\partial Q/\partial\tau^2 = 0$. Replace τ^2 by $\hat{\tau}^2$ in Q and minimize Q with respect to $(\boldsymbol{\theta}, \rho)$ if σ^2 cannot be estimated from the pooled variance from replicated observations.

The predictive distribution of y at any new query point $\boldsymbol{x}$ is also normally distributed with

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

$$s_n^2(\boldsymbol{x}) \triangleq \mathrm{var}(y(\boldsymbol{x})|\boldsymbol{y}, \tau^2, \rho, \boldsymbol{\theta}) = \hat{\tau}^2\left[1 - \boldsymbol{k}(\boldsymbol{x})'(\boldsymbol{K}_n + \hat{\rho}\boldsymbol{I}_n)^{-1}\boldsymbol{k}(\boldsymbol{x})\right] + \hat{\sigma}^2, \tag{3}$$

where $\boldsymbol{k}_n(\boldsymbol{x})$ is the correlation between $\boldsymbol{x}$ and $\boldsymbol{x}_i$, i.e., $\boldsymbol{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$. The GPR predictor is the conditional mean $\hat{y}(\boldsymbol{x}) = \boldsymbol{k}_n(\boldsymbol{x})'(\boldsymbol{K}_n + \hat{\rho}\boldsymbol{I}_n)^{-1}\boldsymbol{y}$. It is also the Best Linear Unbiased Predictor (BLUP) (Santner et al. 2003) for $y(\boldsymbol{x})$ under the GP assumption. We can construct the predictive confidence interval for $y(\boldsymbol{x})$ from (3).

3 ACTIVE LEARNING FOR MANIFOLD GAUSSIAN PROCESS REGRESSION

3.1 Manifold Gaussian Process Regression

The *Manifold Gaussian Process* (mGP) (Calandra, Peters, Rasmussen, and Deisenroth 2016) addresses key limitations of standard Gaussian Process Regression (GPR) by jointly learning a data transformation and a GPR model. Standard GPs rely on covariance functions (e.g., squared exponential) that encode smoothness assumptions, which may be inadequate for modeling discontinuous or complex functions (e.g., robotics tasks with contact dynamics). The mGP framework decomposes the regression task into:

$$F = G \circ M,$$

where $M : \mathscr{X} \to \mathscr{H}$ is a deterministic mapping from the input space $\mathscr{X}$ to a latent feature space $\mathscr{H} \subset \mathbb{R}^Q$, and $G : \mathscr{H} \to \mathscr{Y}$ is a GPR model on $\mathscr{H}$. Here $\mathscr{Y} \subset \mathbb{R}$ is the space of the response. For the map M, we employ multilayer neural networks. In each layer, the transformation is defined as $\boldsymbol{Z}_i = T_i(\boldsymbol{X}) = \sigma_M(\boldsymbol{W}_i\boldsymbol{Z}_{i-1} + \boldsymbol{B}_i)$,

where σ_M is the activation function and $\boldsymbol{W}_i$ and $\boldsymbol{B}_i$ are the weight matrix and bias vector for the i-th layer. Consequently, $M(\boldsymbol{X}) = T_l \circ T_{l-1} \circ \cdots \circ T_1(\boldsymbol{Z}_0)$ with $\boldsymbol{Z}_0 = \boldsymbol{X}$, and the collection of these hyperparameters is denoted by $\boldsymbol{\theta}_M$.

The mGP defines a valid GP over $\mathscr{X}$ with the covariance function:

$$\tau^2 \tilde{k}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tau^2 k(M(\boldsymbol{x}_1), M(\boldsymbol{x}_2)),$$

where k is the standard kernel (e.g., SE-ARD or neural network kernel) with hyperparameters $\boldsymbol{\theta}_G$. This formulation allows the model to adaptively learn feature representations that align with the regression objective, overcoming the limitations of unsupervised transformations (e.g., PCA or random embeddings) that may not preserve the task-relevant structure.

The mGP optimizes the parameters $\boldsymbol{\theta}_{\text{mGP}} = [\boldsymbol{\theta}_M, \boldsymbol{\theta}_G]$ jointly by minimizing the Negative Log Marginal Likelihood (NLML):

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

where $\tilde{\boldsymbol{K}}_n$ is the kernel matrix evaluated on the transformed inputs $M(\boldsymbol{X})$. Gradients are computed via backpropagation through M (if M is a neural network) and the kernel hyperparameters $\boldsymbol{\theta}_G$. The predictive distribution at a test point $\boldsymbol{x}_{n+1}$ is a normal distribution with mean and variance specified by (2) and (3) except $\boldsymbol{k}_n$ is replaced by $\tilde{\boldsymbol{k}}_n$ and $\boldsymbol{K}_n$ by $\tilde{\boldsymbol{K}}_n$.

One can see that the Deep Gaussian Process (or DGP) (Damianou and Lawrence 2013) is closely connected with mGP since both involve latent space transformations before applying GPR. However, they are distinct extensions of standard GPR because DGPs learn this transformation hierarchically through layers of GPs, while mGPs often assume a geometric prior and embed the data accordingly. The mGP framework offers several key advantages over traditional GPs and unsupervised feature learning methods. By jointly optimizing the feature mapping M and GPR G under a supervised objective, it learns task-aware representations that can handle discontinuities (e.g., step functions) and multi-scale patterns more effectively than standard kernels like SE-ARD or neural network covariances. The flexibility of parameterizing M as a neural network (e.g., $[1-6-2]$ layers) allows it to unwrap complex input geometries, while retaining the GP's probabilistic uncertainty quantification—critical for robotics and control applications. However, this approach also introduces challenges: the joint optimization $\boldsymbol{\theta}_{\text{mGP}} = [\boldsymbol{\theta}_M, \boldsymbol{\theta}_G]$ is non-convex and may converge to suboptimal local minima, particularly with high-dimensional $\boldsymbol{\theta}_M$. Additionally, sparse data around discontinuities can lead to overfitting in the deterministic mapping M, causing misaligned feature representations. While probabilistic extensions (e.g., Bayesian neural networks for M) could mitigate this, they often sacrifice tractability. Despite these limitations, the mGP's empirical performance on benchmarks in Calandra et al. (2016) demonstrates its superiority in scenarios where input-space geometry violates standard GP assumptions.

3.2 Active Learning for mGP

The Active Learning Cohn (ALC) criterion (Cohn et al. 1996; Seo et al.) is a foundational information-theoretic strategy for sequential experimental design in Gaussian Process Regression (GPR). It selects new data points $\boldsymbol{x}_{n+1}$ that maximize the model's average predictive uncertainty over the input space. Formally, this is achieved by minimizing the *integrated mean squared error (IMSE)* across the input space (Cohn 1996), which corresponds to reducing the average posterior variance. Neglecting the constant noise variance, the IMSE is defined as:

$$\begin{aligned}
\breve{s}^2_{n+1}(\boldsymbol{x}|\boldsymbol{x}_{n+1}) &\triangleq \hat{\tau}^2 \left[1 - \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})^\top (\tilde{\boldsymbol{K}}_{n+1} + \rho \boldsymbol{I}_{n+1})^{-1} \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})\right], \\
\text{IMSE}(\boldsymbol{x}_{n+1}) &\triangleq \int_{\mathscr{X}} \breve{s}^2_{n+1}(\boldsymbol{x}|\boldsymbol{x}_{n+1})\, \mathrm{d}\boldsymbol{x},
\end{aligned}$$

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

Minimizing the IMSE is equivalent to maximizing the expected reduction in posterior variance, leading to the ALC acquisition function:

$$\text{ALC}(\boldsymbol{x}_{n+1}) = \int_{\mathscr{X}} s_n^2(\boldsymbol{x}) - s_{n+1}^2(\boldsymbol{x}|\boldsymbol{x}_{n+1})\,\mathrm{d}\boldsymbol{x} \propto \int_{\mathscr{X}} \hat{\tau}^2 \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})^\top (\tilde{\boldsymbol{K}}_{n+1} + \rho \boldsymbol{I}_{n+1})^{-1} \tilde{\boldsymbol{k}}_{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

where $s_n^2(\boldsymbol{x})$ is computed via (3) 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 $\tilde{\boldsymbol{K}}_n$. To approximate the integration numerically, we introduce a fixed reference set $\mathscr{X}_{\text{ref}}$ of size m sampled from $\mathscr{X}$ using Latin Hypercube Design. The ALC objective becomes:

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

In contrast to Sauer et al. (2023), which employs DGPs, our method integrates ALC with mGPs. At each active learning iteration, the most informative training point is selected via:

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

and subsequently removed from the candidate pool $\mathscr{X}_{\text{cand}}$. This selection procedure iteratively reduces overall model uncertainty, leveraging the mGP's ability to capture complex geometric structures in latent space. To enhance computational efficiency, we adopt two strategies. First, we support batch selection by choosing B design points per iteration, indexed by round r. Second, instead of evaluating ALC on the full candidate set, we incorporate a pre-screening step: among the candidate set $\mathscr{X}_{\text{cand}}$, we identify the top $K > B$ points with the highest predictive variance, denoted $\mathscr{X}_r^*$:

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

and compute ALC only on $\mathscr{X}_{\text{cand}}^{(r)}$. We then select the B points with the highest ALC scores. The complete procedure is outlined in Algorithm 1.

4 EXAMPLES

We evaluate the proposed *Active Learning Manifold Gaussian Process (ALmGP)* framework on four benchmark experiments. These experiments are designed to assess the effectiveness of ALmGP under varying geometric structures and data complexities. The acquisition strategy is based on the ALC criterion and model optimization is carried out using the L-BFGS algorithm implemented in the PyTorch library (Paszke et al. 2019). The optimizer employs strong Wolfe line search conditions (`line_search_fn = strong_wolfe`), with early stopping triggered when the relative RMSE change

$$\text{Relative RMSE Change} = \left|\frac{\mathscr{L}_{\text{current}} - \mathscr{L}_{\text{previous}}}{\mathscr{L}_{\text{current}} - \mathscr{L}_{\text{initial}}}\right|$$

falls below a predefined tolerance threshold. To ensure positivity and improve stability during optimization, all hyperparameters of the mGP model are squared before being passed to the optimizer. All neural network weights are initialized using PyTorch's default scheme, while GP hyperparameters—output variance and length scale—are initialized to 1.

Algorithm 1 Active Learning for Manifold Gaussian Process Regression (ALmGP).

Input: Initial training dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n_0}$ with small sample size n_0; reference set $\mathscr{X}_{\text{ref}} \subset \mathscr{X}$ and candidate set $\mathscr{X}_{\text{cand}} \subset \mathscr{X}$; tuning parameters including screening size K, batch size B, convergence threshold `Tol`, maximum total sample size $N_{\max}$, and initial values and optimization settings for fitting the mGP.

Output: Trained mGP model and estimated parameters after active data acquisition.

Step 0: Fit the mGP model using the training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n_0}$ by minimizing the negative log marginal likelihood (NLML) with respect to $\boldsymbol{\theta}_{\text{mGP}}$ using the L-BFGS algorithm (Nocedal and Wright 2006).

for $r = 1, 2, \ldots, \lfloor N_{\max}/B \rfloor$ **do**

Step 1: Compute posterior predictive variance for all points in $\mathscr{X}_{\text{cand}}$, and select the top K points with highest variance. Denote this screened set as $\mathscr{X}_{\text{cand}}^{(r)}$.

Step 2: For each $\boldsymbol{x} \in \mathscr{X}_{\text{cand}}^{(r)}$, compute the ALC acquisition score $\text{ALC}(\boldsymbol{x})$ using the current mGP model.

Step 3: Select the top B points in $\mathscr{X}_{\text{cand}}^{(r)}$ with the highest ALC values. Add them to the training set and remove them from $\mathscr{X}_{\text{cand}}$.

Step 4: Re-train or update the mGP model using the expanded training dataset.

Step 5: Evaluate the cross-validation error or training mean squared error (MSE). If the error falls below the threshold `Tol`, terminate the loop and return the final fitted mGP model.

end for

Model performance is evaluated using the Root Mean Square Error (RMSE) computed on an independently generated test set, which has n_{test} number of points sampled from $\mathscr{X}$ using Latin Hypercube Design:

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

where $\hat{y}_i$ and y_i denote the predicted and ground truth values at test location $\boldsymbol{x}_i$, respectively. Each experiment is repeated 10 times to ensure statistical robustness. The mean and range of RMSE values are reported and compared against a baseline strategy using completely random sampling B design points from $\mathscr{X}_{\text{cand}}$.

4.1 Piecewise Trigonometric Function

This example features a piecewise trigonometric function adapted from Sauer et al. (2023), with noise $\varepsilon \sim \mathscr{N}(0, 0.1^2)$:

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

The initial dataset consists of $n_0 = 10$ points generated using Latin Hypercube Design over $[0,1]$. The test set contains 500 uniformly spaced points, and both $\mathscr{X}_{\text{cand}}$ and $\mathscr{X}_{\text{ref}}$ consist of 100 evenly spaced points in the same interval. The neural network uses [1-6-2] architecture and LogSigmoid activation. The batch size is $B = 1$, and the total budget is $N_{\max} = 50$. L-BFGS uses `history_size` = 20, a learning rate of 0.001, a maximal number of iterations per optimization step 20 and a maximum of 5,000 training iterations with early stopping tolerance `Tol` = 10^{-5}. The active learning process ends with 15 new samples and achieves an average completion RMSE of 0.156. Figure 1 shows the details of the training data, the fitted function, the latent space, and the comparison of RMSE of active learning for mGP using the proposed ALC strategy and random sampling.

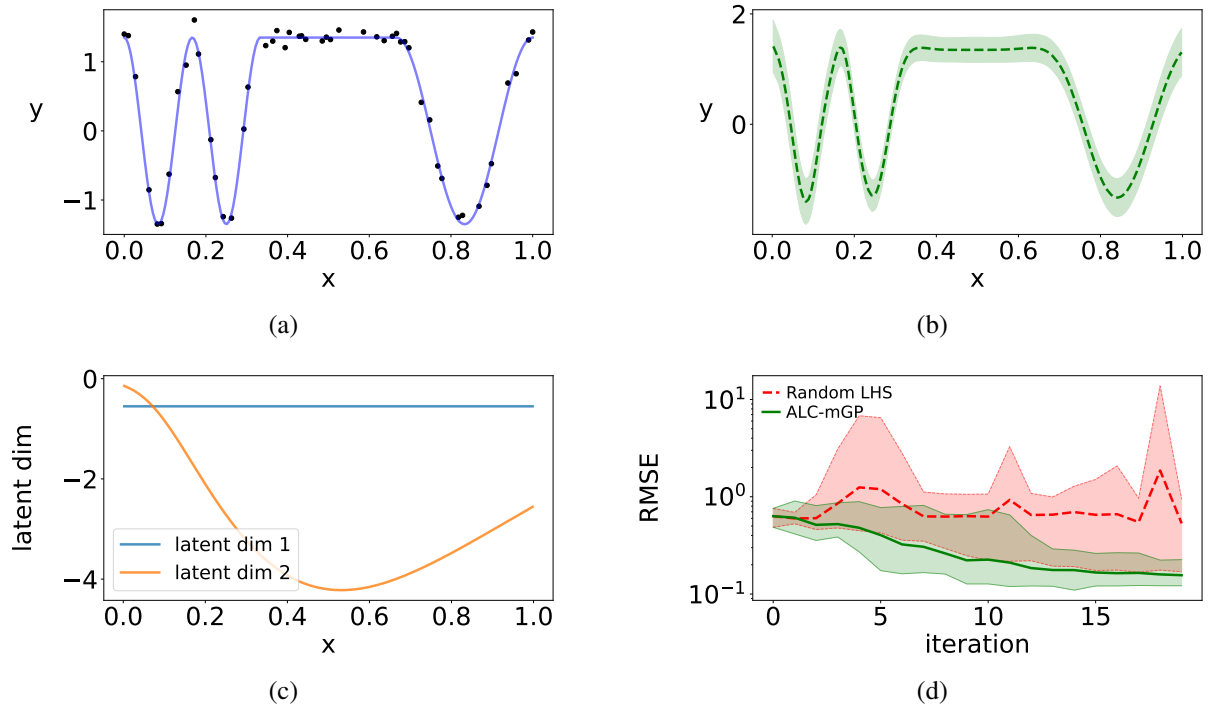


Figure 1: (a) The black dots indicate the final selected training data and the blue solid line is the true piecewise trigonometric function. (b) The green dashed line represents the final predicted value the green shaded area denotes the prediction confidence interval. (c) Learned a two-dimensional latent space. (d) Comparison of test RMSE over iterations: our method (green, solid) vs. random acquisition (red, dashed). Lines show the mean of 10 runs; shaded areas indicate the minimum to maximum range.

4.2 Two-Dimensional Deterministic Function

We consider a two-dimensional function also from (Calandra et al. 2016), defined as:

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

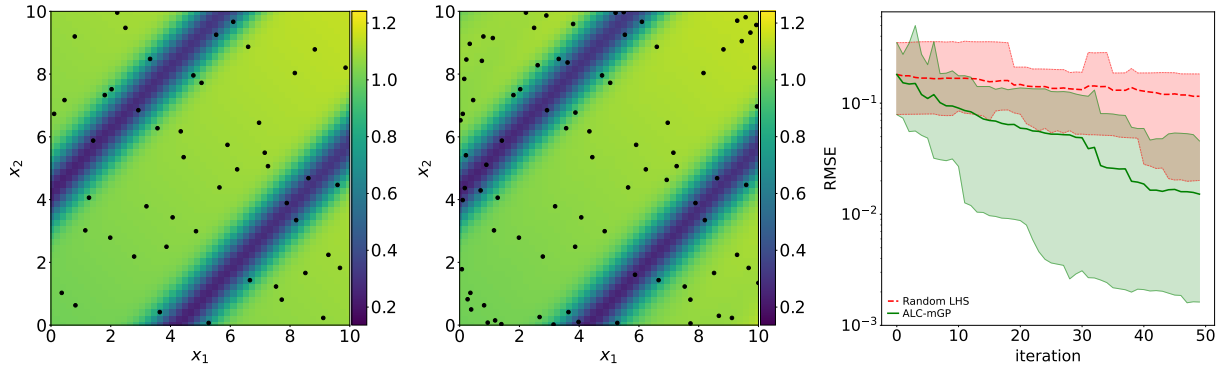
which is then rotated by 45°. The function $\phi(\cdot;\mu,\sigma^2)$ is the PDF of $\mathcal{N}(\mu,\sigma^2)$. Then initial training data ($n_0 = 50$), test data, and the reference set $\mathcal{X}_{\text{ref}}$ are all of size 500 and sampled from $[0,10]^2$ using LHD. Predictions are also evaluated on a grid over $\mathcal{X}$ with mesh size 0.2. This neural network maps the 2D input to a latent space, and the architecture is $[2-10-3]$, with the same LogSigmoid activation. The active learning proceeds with batch size $B = 1$ and budget $N_{\max} = 50$. The L-BFGS is run with `history_size` = 50, learning rate 0.01, a maximal number of iterations per optimization step 50, and a max of 5000 iterations. The active learning procedure concludes with 100 samples after 50 iterations, reaching an average RMSE of 0.0152 for the proposed method. Figure 2 and 3 show the detailed results.

4.3 Three-Dimensional Function on the Unit Sphere

This example uses a test function defined on the 3D unit sphere:

$$f(x,y,z) = \cos(x) + y^2 + e^z, \quad \text{where } x^2+y^2+z^2 = 1.$$

It can be reformulated on the disk as $g(x,y) = \cos(x) + y^2 + e^{1-x^2-y^2}$ for $x^2+y^2 \le 1$. We construct training, test, and reference sets (each of size 500) by sampling $(v,\alpha) \in [-1,1] \times [0,2\pi]$ using Latin



(a) (b) (c)

Figure 2: (a) The heat map is the true function value with the black dots representing the initial training data. (b) The heat map is the final predicted value, and the black dots are the overall training data. (c) Comparison of test RMSE over iterations: our method (green, solid) vs. random acquisition (red, dashed). Lines show the mean of 10 runs; shaded areas indicate the minimum to maximum range.

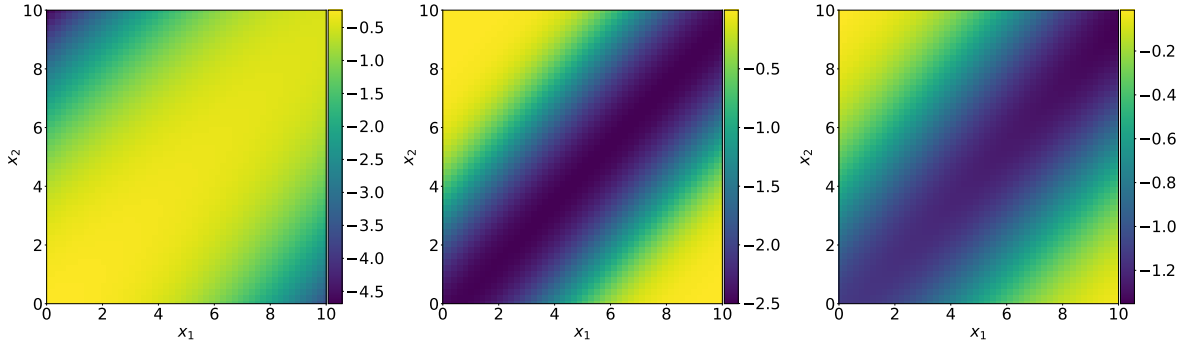


Figure 3: The three heat maps represent three latent dimensions with respect to the original input (x_1, x_2).

Hypercube Design and mapping them to (x, y, z) via:

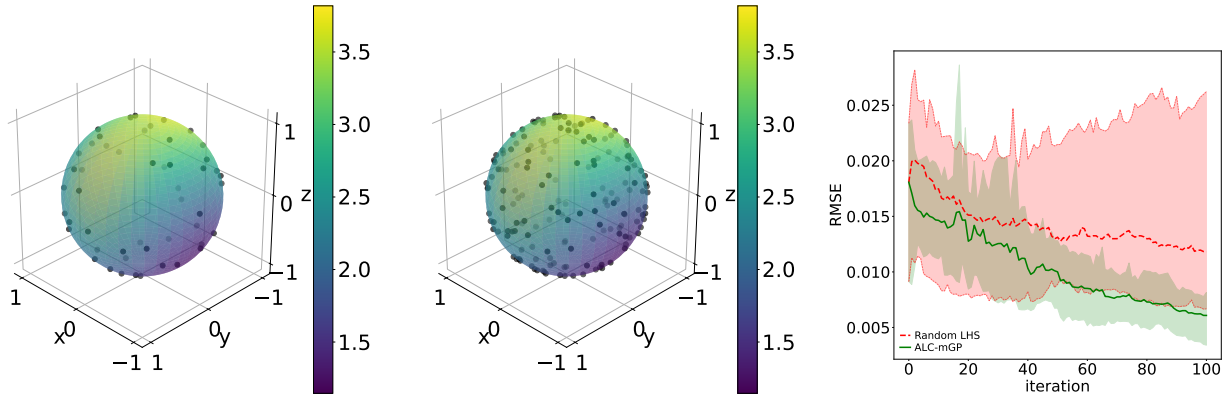
$$z = v, \quad x = \sqrt{1-v^2}\cos(\alpha), \quad y = \sqrt{1-v^2}\sin(\alpha).$$

The neural network has architecture [3-10-2], and training starts from $n_0 = 50$ samples. Active learning is run with $N_{\max} = 100$ and $B = 1$. The L-BFGS is configured with `history_size` = 50, a learning rate of 0.01, a maximal number of iterations per optimization step 20 and 5,000 training iterations. The active learning stops when it reaches the full budget of 100 samples, yielding the averaged RMSE as low as 6×10^{-3} for the proposed method. See the results in Figure 4 and 5.

4.4 Borehole

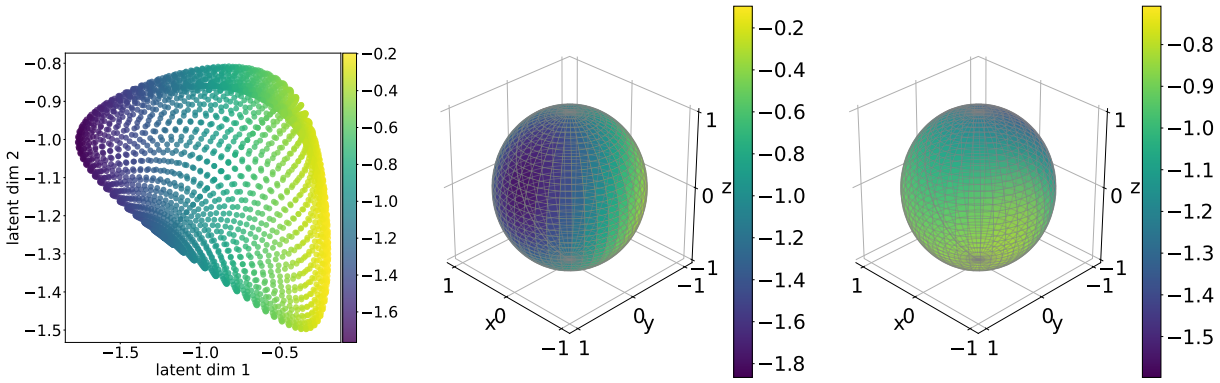
This example models the groundwater flow function of the borehole, a well-known benchmark for nonlinear regression. The function depends on eight physical parameters, such as radius and aquifer transitivity. The detailed definition of the variable and the math expression of the function are included in Kang et al. (2023). Inputs are generated via LHS and mapped to physical ranges.

The neural network has architecture [8-30-4], and training starts from $n_0 = 50$ samples. Active learning is run with $N_{\max} = 150$ and $B = 1$. The L-BFGS uses `history_size` = 50, learning rate 0.001, a maximal number of iterations per optimization step 100, and 10,000 training iterations. Early stopping is triggered



(a) (b) (c)

Figure 4: (a) The heat map represents the true function value on the 3-dim unit sphere and the dots are the initial training data. (b) The heat map is the final predicted value and the black dots are the overall training data. (c) Comparison of test RMSE over iterations: our method (green, solid) vs. random acquisition (red, dashed). Lines show the mean of 10 runs; shaded areas indicate the minimum to maximum range.



(a) (b) (c)

Figure 5: (a) The heat map represents the true function value f with respect to the two latent dimensions. (b) The heat map represents the 1st latent dimension with respect to (x,y,z). (c) The heat map represents the 2nd latent dimension with respect to (x,y,z).

when the relative test RMSE change is below 10^{-8}. The proposed active learning stops when it reaches the full budget of 150 samples, yielding an average RMSE as low as 0.6. See the results in Figure 6.

5 CONCLUSION

This paper introduced an active learning framework for manifold Gaussian Process (mGP) regression, combining the strengths of manifold learning with sequential experimental design. By integrating the Active Learning Cohn criterion with mGPs, we demonstrated how to strategically select training points that simultaneously improve predictive accuracy and exploit low-dimensional data structure. The proposed method addresses key limitations of standard GPs in high-dimensional spaces, where traditional covariance functions often fail to capture complex or discontinuous patterns. Our experiments validated the framework's

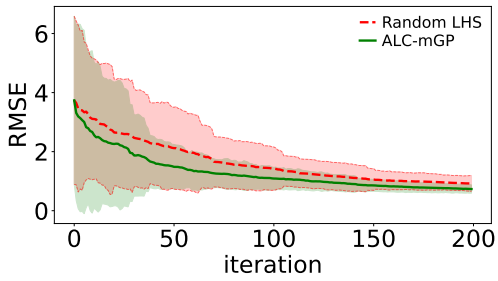


Figure 6: Comparison of test RMSE over iterations: our method (green, solid) vs. random acquisition (red, dashed). Lines show the mean of 10 runs; shaded areas indicate the minimum to maximum range.

efficacy in reducing integrated mean squared prediction error (IMSE) while maintaining computational tractability through efficient hyperparameter optimization and neural network-based manifold learning.

Several promising extensions emerge from this work. First, the current framework assumes a deterministic manifold mapping M. A Bayesian treatment of M (e.g., via Bayesian neural networks or variational inference) could quantify uncertainty in the latent space, improving robustness when training data is sparse or noisy. This would require advances in approximate inference to handle the non-conjugacy between neural networks and GPs. Second, the computational cost of active learning scales with the reference set size m is used for IMSE approximation. Developing sparse approximations or gradient-based sampling for the ALC integral could enhance scalability to very high-dimensional input spaces. Techniques from stochastic optimization, such as mini-batch reference sampling and merit exploration. Lastly, the interpretability of learned manifolds could be improved by incorporating domain-specific constraints (e.g., physics-informed neural networks) or disentangled representations. This would bridge the gap between black-box flexibility and interpretable modeling, particularly in scientific applications where mechanistic understanding is paramount.

ACKNOWLEDGMENTS

The four authors' work was partially supported by NSF Grant DMS-2429324. Part of this research was performed while L. Kang was visiting the Institute for Mathematical and Statistical Innovation (IMSI) at the University of Chicago from March 3 to May 24, 2025, which is supported by the National Science Foundation (Grant DMS-1929348).

REFERENCES

Beyer, K., J. Goldstein, R. Ramakrishnan, and U. Shaft. 1999. "When Is "Nearest Neighbor" Meaningful?". In *Proceedings of the International Conference on Database Theory*. January 10–12, Berlin, Heidelberg, 217–235.

Binois, M., J. Huang, R. B. Gramacy, and M. L. and. 2019. "Replication or Exploration? Sequential Design for Stochastic Simulation Experiments". *Technometrics* 61(1):7–23.

Borovitskiy, V., A. Terenin, P. Mostowsky, and M. Deisenroth (he/him). 2020. "Matérn Gaussian Processes on Riemannian Manifolds". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Volume 33, 12426–12437. IEEE: Curran Associates, Inc. New York, U.S.A.

Calandra, R., J. Peters, C. E. Rasmussen, and M. P. Deisenroth. 2016. "Manifold Gaussian Processes for Regression". In *2016 International Joint Conference on Neural Networks (IJCNN)*. July 24-29, Vancouver, BC, Canada, 3338-3345.

Chen, J., L. Kang, and G. L. and. 2021. "Gaussian Process Assisted Active Learning of Physical Laws". *Technometrics* 63(3):329–342.

Cohn, D. A. 1996. "Neural Network Exploration Using Optimal Experiment Design". *Neural Networks* 9(6):1071–1083.

Cohn, D. A., Z. Ghahramani, and M. I. Jordan. 1996. "Active Learning with Statistical Models". *Journal of Artificial Intelligence Rsearch* 4:129–145.

Damianou, A., and N. D. Lawrence. 2013, 29 Apr–01 May. "Deep Gaussian Processes". In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, edited by C. M. Carvalho and P. Ravikumar, Volume 31 of *Proceedings of Machine Learning Research*, 207–215. Scottsdale, Arizona, USA: PMLR.

Fichera, B., S. Borovitskiy, A. Krause, and A. G. Billard. 2023. "Implicit Manifold Gaussian Process Regression". In *Advances in Neural Information Processing Systems*, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Volume 36, 67701–67720: Curran Associates, Inc.

Frazier, P. I. 2018. "Bayesian Optimization". In *Recent Advances in Optimization and Modeling of Contemporary Problems*. October 19, Phoenix, Arizona, 255-278.

Heo, J., and C.-L. Sung. 2025. "Active Learning for A Recursive Non-Additive Emulator for Multi-Fidelity Computer Experiments". *Technometrics* 67(1):58–72.

Houlsby, N., F. Huszár, Z. Ghahramani, and M. Lengyel. 2011. "Bayesian Active Learning for Classification and Rreference Learning". *arXiv preprint arXiv:1112.5745v1*.

Hung, Y. 2011. "Penalized Blind Kriging in Computer Experiments". *Statistica Sinica* 21(3):1171.

Joseph, V. R., Y. Hung, and A. Sudjianto. 2008. "Blind Kriging: A New Method for Developing Metamodels". *Journal of mechanical design* 130(3):031102.

Joseph, V. R., and L. Kang. 2011. "Regression-Based Inverse Distance Weighting with Applications to Computer Experiments". *Technometrics* 53(3):254–265.

Kang, L., Y. Cheng, Y. Wang, and C. Liu. 2023. "Energetic Variational Gaussian Process Regression for Computer Experiments". *arXiv preprint arXiv:2401.00395*.

Kang, L., Y. Cheng, Y. Wang, and C. Liu. 2024. "Energetic Variational Gaussian Process Regression". In *2024 Winter Simulation Conference (WSC)*, 3542–3553 https://doi.org/10.1109/WSC63780.2024.10838889.

Kapoor, A., K. Grauman, R. Urtasun, and T. Darrell. 2007. "Active Learning with Gaussian Processes for Object Categorization". In *2007 IEEE 11th International Conference on Computer Vision*. October 14-21, Rio de Janeiro, Brazil, 1-8.

Kim, H., D. Sanz-Alonso, and R. Yang. 2024. "Optimization on Manifolds via Graph Gaussian Processes". *SIAM Journal on Mathematics of Data Science* 6(1):1–25.

Krause, A., A. Singh, and C. Guestrin. 2008. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". *Journal of Machine Learning Research* 9(2):235–284.

Ma, C., Y. Li, and J. M. Hernandez-Lobato. 2019, 09–15 Jun. "Variational Implicit Processes". In *Proceedings of the 36th International Conference on Machine Learning*, edited by K. Chaudhuri and R. Salakhutdinov, Volume 97 of *Proceedings of Machine Learning Research*, 4222–4233: PMLR.

MacKay, D. J. C. 1992, 07. "Information-Based Objective Functions for Active Data Selection". *Neural Computation* 4(4):590–604.

Mallasto, A., and A. Feragen. 2018. "Wrapped Gaussian Process Regression on Riemannian Manifolds". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 18-23, Salt Lake City, UT, USA, 5580-5588.

Martinez-Cantin, R., K. Tee, and M. McCourt. 2018, 09–11 Apr. "Practical Bayesian Optimization in the Presence of Outliers". In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, edited by A. Storkey and F. Perez-Cruz, Volume 84 of *Proceedings of Machine Learning Research*, 1722–1731: PMLR.

Meilă, M., and H. Zhang. 2024. "Manifold Learning: What, How, and Why". *Annual Review of Statistics and Its Application* 11:393–417.

Nocedal, J., and S. J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research. New York: Springer.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, *et al*. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Volume 32: Curran Associates, Inc.

Rasmussen, C. E., and C. K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press.

Santner, T. J., B. J. Williams, W. I. Notz, and B. J. Williams. 2003. *The Design and Analysis of Computer Experiments*, Volume 1. New York: Springer.

Sauer, A., R. B. Gramacy, and D. H. and. 2023. "Active Learning for Deep Gaussian Process Surrogates". *Technometrics* 65(1):4–18.

Seo, S., M. Wallat, T. Graepel, and K. Obermayer. "Gaussian Process Regression: Active Data Selection and Test Point Rejection". In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Volume 3, 241–246 vol.3.

Seo, S., M. Wallat, T. Graepel, and K. Obermayer. 2000. "Gaussian Process Regression: Active Data Selection and Test Point Rejection". In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Volume 3. July 27, Como, Italy, 241-246.

Srinivas, N., A. Krause, S. Kakade, and M. Seeger. 2010. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In *ICML'10: Proceedings of the 27 th International Conference on Machine Learning*. June 21-24, Haifa, Israel, 1015–1022.

Tenenbaum, J. B., V. de Silva, and J. C. Langford. 2000. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". *Science* 290(5500):2319–2323.

AUTHOR BIOGRAPHIES

YUANXING CHENG is a Ph.D student in the Department of Applied Mathematics at the Illinois Institute of Technology in Chicago, IL. Advised by Dr. Lulu Kang and Dr. Chun Liu, he has worked on the topic of uncertainty quantification. His email address is ycheng46@hawk.illinoistech.edu.

LULU KANG is an Associate Professor in the Department of Mathematics and Statistics at the University of Massachusetts Amherst. Her research interests include Statistical Learning/Machine Learning, Statistical Design of Experiments, Uncertainty Quantification, Bayesian Statistical Modeling, Approximate Inference, and Optimization. She serves as an associate editor for *Technometrics* and *SIAM/ASA Journal on Uncertainty Quantification*. Her email address is lulukang@umass.edu and her website is https://sites.google.com/umass.edu/lulukang/home.

YIWEI WANG is an Assistant Professor in the Department of Mathematics at the University of California, Riverside. His research interests include mathematical modeling and scientific computing with applications in physics, material science, biology, and data science. His email address is yiwei.wang@ucr.edu and his website is https://profiles.ucr.edu/app/home/profile/yiweiw.

CHUN LIU is a Professor and Chair of the Department of Applied Mathematics at the Illinois Institute of Technology, Chicago, IL. His research interests center around partial differential equations, calculus of variations, and their applications in complex fluids. His e-mail address is cliu124@iit.edu and his website is https://www.iit.edu/directory/people/chun-liu.