Domain Invariant Learning for Gaussian Processes and Bayesian Exploration

Xilong Zhao[1*], Siyuan Bian[1], Yaoyun Zhang[1], Yuliang Zhang[1], Qinying Gu[2], Xinbing Wang[1], Chenghu Zhou[1], Nanyang Ye[1†]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]Shanghai Aritificial Intelligence Laboratory, Shanghai, China
zhaoxilong@sjtu.edu.cn, biansiyuan@sjtu.edu.cn, yaoyunzhang897@gmail.com, blu_26@sjtu.edu.cn, guqinying@pjlab.org.cn, xwang8@sjtu.edu.cn, zhouchsjtu@gmail.com, ynylincoln@sjtu.edu.cn

Abstract

Out-of-distribution (OOD) generalization has long been a challenging problem that remains largely unsolved. Gaussian processes (GP), as popular probabilistic model classes, especially in the small data regime, presume strong OOD generalization abilities. Surprisingly, their OOD generalization abilities have been under-explored before compared with other lines of GP research. In this paper, we identify that GP is not free from the problem and propose a domain invariant learning algorithm for Gaussian processes (DIL-GP) with a min-max optimization on the likelihood. DIL-GP discovers the heterogeneity in the data and forces invariance across partitioned subsets of data. We further extend the DIL-GP to improve Bayesian optimization's adaptability on changing environments. Numerical experiments demonstrate the superiority of DIL-GP for predictions on several synthetic and real-world datasets. We further demonstrate the effectiveness of the DIL-GP Bayesian optimization method on a PID parameters tuning experiment for a quadrotor. The full version and source code are available at: https://github.com/Billzxl/DIL-GP.

1 Introduction

Gaussian processes (GP) have been widely used as models for Bayesian nonparametrics methods in machine learning (Gahungu et al. 2022; Moreno-Muñoz, Feldager, and Hauberg 2022). However, the extrapolation ability of GP especially in the long tail regime suffers from generalization problems, i.e. failing on the non-i.i.d test data (Pilario et al. 2022). How to guarantee the generalization ability of GP on data sampled from out-of-distribution sets is of great importance, also known as the *out-of-distribution (OOD) generalization.* Previous GP methods mitigate the OOD generalization problems often by hand-crafting problem-specific kernel functions to capture the inherent long-tail characteristics of the data. While these methods are effective, it is hard to transfer the success to other types of problems (Pilario et al. 2022; Schmidt, Morales-Álvarez, and Molina 2023; Chen, Yin, and Cui 2023; Jørgensen and Osborne 2022). This can also lead to degenerated sample efficiency of Bayesian optimization when GP is used as the surrogate function (Lei et al. 2021).

To improve OOD generalization, many domain generalization methods have been proposed to learn invariant representations across different domains. This can be achieved by invariant risk minimization (IRM) (Arjovsky et al. 2019), which attempts to tackle this issue by penalizing predictions based on the unstable spurious features in the data collected from different domains. Although methods like IRM seems promising for OOD generalization, data need to be manually split into different domains to enable them to minimize the performance discrepancies across domains. This fundamental restrictions prevent this algorithmic schemes to be practical. First, how data are split can largely influence the final prediction performances. Second, sometimes it is impossible to split the data in continuous environments, especially for Bayesian optimization scenario, where the data is sequentially sampled from trials, such as sensor data collected on a drone flight under turbelent stormy weather.

In this paper, we propose a novel approach, domain invariant learning for Gaussian processes (DIL-GP), to iteratively construct worst-case domains/environments splittings for IRM for learning GP parameters. The optimized partition for data samples enforce GP to generalize on the worst possible data distributions thus benefit the OOD generalization. Then, DIL-GP is further extended as the surrogate function for Bayesian optimization. We demonstrate that the combination of this adversarial partitioning and IRM is key for learning generalizable GP kernels. Numerical experiments on synthetic and real-world datasets shows the superiority of the proposed algorithmic scheme.

2 Related Work

2.1 Gaussian Processes

Gaussian Process(GP), as a non-parametric Bayesian model that models the relationship between data by defining a stochastic process has the characteristics of non-parametricity, flexibility, interpretability, and uncertainty quantification. GP has demonstrated utility on diverse domains, including but not limited to regression analysis (Luo,

*Xilong Zhao is currently a student at MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China.
†Corresponding author.

Nattino, and Pratola 2022; Lalchand et al. 2022) , classification tasks (Zhao et al. 2021; Shi, Yu, and Yu 2021) and optimization (Wuethrich, Schölkopf, and Krause 2021; Cai and Scarlett 2021). To improve GP's generalization ability, previous works focus on improving GP's kernel functions, especially for computation efficiencies. For example, in (Cohen, Daulton, and Osborne 2022), a binary tree kernel is proposed to improve its computation efficiency (Lu, Boukouvalas, and Hensman 2022). Another line of works focus on improving GP's generalization abilities for specific applications. Shared GP kernels are proposed for multiple users for federated learning (Achituve et al. 2021). In (Chen, Tripp, and Hernández-Lobato 2023), meta learning is used to learn adaptable GP kernel parameters for molecular property prediction. Despite their domain-specific successes, general method to improve GP's generalization abilities, especially on out-of-distribution data, is largely underexplored.

2.2 Out-of-Distribution Generalization

Out-of-distribution (OOD) generalization, the task of generalizing under distribution shifts, has been researched in many areas, such as computer vision (Zhang et al. 2022; Huang et al. 2022; Wang et al. 2021; Niu et al. 2022) and natural language processing (Yang et al. 2022; Hendrycks et al. 2020; Sun et al. 2022) Previous works mainly focus on improving the out-of-distribution generalization ability for deep neural networks. For example, (Huang et al. 2022) propose a deep neural network training strategy that randomly drop the most prominent feature for image classification in each iteration. (Dong et al. 2022) propose to ensemble multiple deep neural networks each trained on a subset of the data to improve their out-of-distribution generalization abilities on image classification tasks. While these methods provide insights into how to improve the OOD generalization abilities of deep neural networks, they are reliant on neural architectures not suitable for GP. Besides, typical OOD generalization methods relies on the domain labels to learn invariant features across domains to achieve OOD generalization while the domain labels are hard or impossible to get in some scenarios where GP are widely used (Arjovsky et al. 2019). For example, in the small data regime when data collection is expensive, obtaining domain labels will be even harder. Besides, many existing datasets containing domain shifts may come without providing domain labels. In applications such as Bayesian optimization, where the data are actively sampled , the domain labels are also hard to obtain.

3 Methodology

We aim to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{X} \subseteq \mathcal{R}^d$ to $\mathcal{Y} \subseteq \mathcal{R}$ given the training set with N sample pairs $X = (x_1, \cdots, x_n) \in \mathcal{R}^{n\times d}$ of n inputs with $x_j \in \mathcal{R}^d$ and corresponding outputs $y = (y_1, \cdots, y_n) \in \mathcal{R}^n$.

3.1 Gaussian Process

Gaussian process is a stochastic process $f \sim \mathcal{GP}(\mu, k_\theta)$ with mean function μ and kernel $k_\theta : \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$ (Rasmussen and Williams 2006). The kernel is parameterized with $\theta \in \mathcal{R}^\Theta$. The set of function values $\mathbf{f} = (f(x_1), \cdots, f(x_n))^T \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}_\theta)$ is a joint Gaussian with $\boldsymbol{\mu}_j = \mu(x_j)$ and $\mathbf{K}_{ij} = k_\theta(x_i, x_j)$. In order to predict the corresponding output y^* given the input datum x^*, the output $p(y^*|X, y, x^*) = \mathcal{N}(\mu^*, K^*)$ is a Gaussian with mean and variance given by:

$$\mu^* = \mu(x^*) + \mathbf{K}_\theta^{*x}(\mathbf{K}_\theta^{xx} + \sigma^2 I)^{-1}(y - \mu(x^*)) \quad (1)$$

$$K^* = \mathbf{K}_\theta^{**} - \mathbf{K}_\theta^{*x}(\mathbf{K}_\theta^{xx} + \sigma^2 I)^{-1}\mathbf{K}_\theta^{x*} \quad (2)$$

where $\mathbf{K}_\theta^{**} = [k(x^*, x^*)]$ and $\mathbf{K}_\theta^{*x} = [k(x^*, x_1), \cdots, k(x^*, x_n)]^T$ denotes the kernel matrix and similarly for $\mathbf{K}_\theta^{xx}$ and $\mathbf{K}_\theta^{x*}$. I is the identity matrix. The kernel parameters θ can largely determine the performance of the GP, and the optimal kernel parameters can be obtained by the maximum posterior likelihood method:

$$\begin{aligned}\log p(y|x, \theta) = &-\frac{1}{2}(y - \mu(x))^T(\mathbf{K}_\theta^{xx} + \sigma^2 I)^{-1}(y - \mu(x)) \\ &- \frac{1}{2}\log|\mathbf{K}_\theta^{xx} + \sigma^2 I| - \frac{n}{2}\log(2\pi)\end{aligned} \quad (3)$$

where the first term measures the goodness of fit, the second term is the regularization for the kernel complexity, and the last term is the normalizing constant.

3.2 Data Distribution Shifts Challenges

Data distribution shifts widely exist in real world scenarios as it is practically impossible to collect data from all domains. For example, a computer disease diagnosis system trained in one hospital's data has to generalize to other hospitals. This also pose challenges for GPs, suppose data is sampled from different environments/domains $e \in \mathcal{E}$, $x \sim p(x|e)$. The average negative likelihood (loss) on training and test domains are given below:

$$\text{NLL}_{\text{train}}(\theta) = -\int p_{\text{train}}(e)p(x|e)\log p(y|x, \theta, e) \quad (4)$$

$$\text{NLL}_{\text{test}}(\theta) = -\int p_{\text{test}}(e)p(x|e)\log p(y|x, \theta, e) \quad (5)$$

It is obvious to see the minimizer of $\text{NLL}_{\text{train}}$ is not necessarily suitable for NLL_{test}. This poses challenges for GP's generalization abilities. To solve this problem, an intuitive way is to minimize the negative likelihood on different domains at the same time. For example, the distributional robustness optimization method seeks to optimize on the worst environment's data to achieve OOD generalization (Sagawa* et al. 2020). Invariant risk minimization enforce models to reach local minima for each domain to learn domain invariant representations (Arjovsky et al. 2019). However, these approaches all require prior knowledge about which domain the data belongs to. Next we will present algorithm for inferring domains directly from data without human assigned environment prior.

3.3 Inferring Domains for Domain Invariant Learning

Intuitively, as data are sampled from the mixture of different domains, if the model can perform well on the worst possible domain, the model will be suitable for other domains. As

the domain information is unknown in our setting, we partition the data into different subsets to construct domains for domain invariant learning. The data are partitioned to create challenges for invariant learning methods. Here we use the invariant risk minimization penalty as in (Arjovsky et al. 2019) but other variants can also be used. We denote the domain index for each data as $q_i \in [0, \cdots, \mathcal{E}]$ to represent which domain the data pair (x_i, y_i) belongs to. We first give the likelihood on environment e:

$$\log p(y|x, \theta, e) = -\frac{1}{2}((y - \mu(x)) * \mathbb{1}(q = e))^T(\mathbf{K}_\theta^{xx} + \sigma^2 I)^{-1}((y - \mu(x)) * \mathbb{1}(q = e)) - \frac{1}{2}\log|\mathbf{K}_\theta^{xx} + \sigma^2 I| - \frac{n}{2}\log(2\pi) \tag{6}$$

where $*$ is the element-wise multiplication operation, $q = [q_1, q_2, \cdots, q_n]^T$, $\mathbb{1}$ is the identity function. Then our objective is to maximize the invariant risk minimization penalty with regard to the domain index q:

$$\max_q \sum_{e \in \text{supp}(e)} \|\nabla_w|_{w=1} \log p(y|x, w * \theta, e)\|^2 \tag{7}$$

Where $\text{supp}(e)$ represents the set of possible values for the environment variable e. As q_i is a positive discrete variable, the above optimization problem is NP-hard problem. We fix $\mathcal{E}$ to be two and use a soft subsitute for $\mathbb{1}(q = e)$: $\mathbb{1}(q = e) \approx \text{sigmoid}(\tilde{q}), \tilde{q} \in \mathcal{R}^n$. This facilities the gradient computation and leads to stable empirical performances.

3.4 Domain Invariant Learning for Gaussian Processes

After the domain labels are inferred, we finally derive the domain invariant learning for Gaussian processes (DIL-GP). The algorithm is shown in Algorithm 1. Compared with the vanilla GP, our proposed algorithmic scheme is less prone to over-fitting to the majority group of the data and can provide fairer and better performances especially in novel domains. As one of the most important applications of GP, Bayesian optimization is widely used for black-box optimization. The generalization ability of surrogate model on unseen data is pivotal for the success of Bayesian optimization (Hvarfner, Hutter, and Nardi 2022; Daulton, Balandat, and Bakshy 2021; Astudillo et al. 2021; Zhang, Zhang, and Frazier 2021). We extend DIL-GP for Bayesian optimization to further demonstrate its OOD generalization ability.

Theorem 1. Under mild assumptions, given $\delta \in (0, 1)$, DIL-GP's OoD risk is strictly no larger than vanilla GP's OoD risk $R_{\text{DIL-GP}} = \mathbb{E}_{x^*}(\mu_{\text{DIL-GP}}(x^*) - f(x^*))^2 \leq R_{\text{GP}} = \mathbb{E}_{x^*}(\mu_{\text{GP}}(x^*) - f(x^*))^2$, with probability $\geq 1 - \delta$.

The proof of this theorem can be found in the Appendix.

3.5 Bayesian Optimization with DIL-GP

To adapt surrogate model in Bayesian optimization in fast-changing environments, we propose to incorporate DIL-GP as the surrogate model for Bayesian optimizations. Previous Bayesian optimization methods tend to ignore the domain shifts existed in the sampled data thus the optimized black-box model may perform poorly in novel domains. The DIL Bayesian optimization algorithm is shown in Algorithm 2. Here, D_t is a data set that includes all queried data points $(x_i, f(x_i))$ so far, $\alpha(x \mid D_t)$ is an acquisition function used to determine the next point to query given the current Gaussian process model. Commonly used acquisition functions include expected improvement (EI) and upper confidence bound (UCB). At each iteration, the algorithm fits a Gaussian process model to the current data set, selects the next point to query based on the acquisition function, and adds the new data point to the data set. The algorithm continues until a stopping criterion is met, and the optimal solution $\hat{x}$ is the point with the highest function value among all known data points.

We prove an upper bound on the cumulative regret of DIL-Bayesian optimization (DIL-BO), where the procedure for proving is in the Appendix:

Theorem 2. (Convergence of DIL-BO) Given $\delta \in (0, 1)$, denote γ_t the maximum information gain after observing t observations. If run the BO process with $\beta_t = B +$

Algorithm 1: Domain Invariant Learning for Gaussian Processes

Require: Input data pairs $\{x_i, y_i\}, i = 1, \cdots, n$, number of inner maximization steps T_1, number of outer minimization steps T_2, initial kernel functions $K_\theta(\cdot, \cdot)$ parameterized by θ, inner maximization learning rate η_1, outer minimization learning rate η_2, IRM penalty coefficient λ.

Ensure: Optimized kernel parameters θ

1: Initialize kernel matrix K_θ^{xx} with input data $\{x_i, y_i\}, i = 1, \cdots, n$
2: Initialize the domain index vector $\tilde{q}$
3: **for** $t_{\text{inner}} = 1$ **to** T_1 **do**
4: **for** $t_{\text{outter}} = 1$ **to** T_2 **do**
5: Update $\tilde{q}$ with gradient ascent according to Eq 7:

$$\tilde{q}_{t+1} \leftarrow \tilde{q}_t + \eta_1 \nabla_{\tilde{q}} \sum_{e \in \text{supp}(e)} \|\nabla_w|_{w=1} \log p(y|x, w * \theta, e)\|^2$$

6: **end for**
7: Update θ with gradient descent on the likelihood with the IRM penalty term:

$$\theta_{t+1} \leftarrow \theta_t - \eta_2 \nabla_\theta \Bigg[\log p(y|x, \theta) + \lambda \sum_{e \in \text{supp}(e)} \|\nabla_w|_{w=1} \log p(y|x, w * \theta, e)\|^2 \Bigg]$$

8: **end for**
9: Output optimized kernel parameters θ.

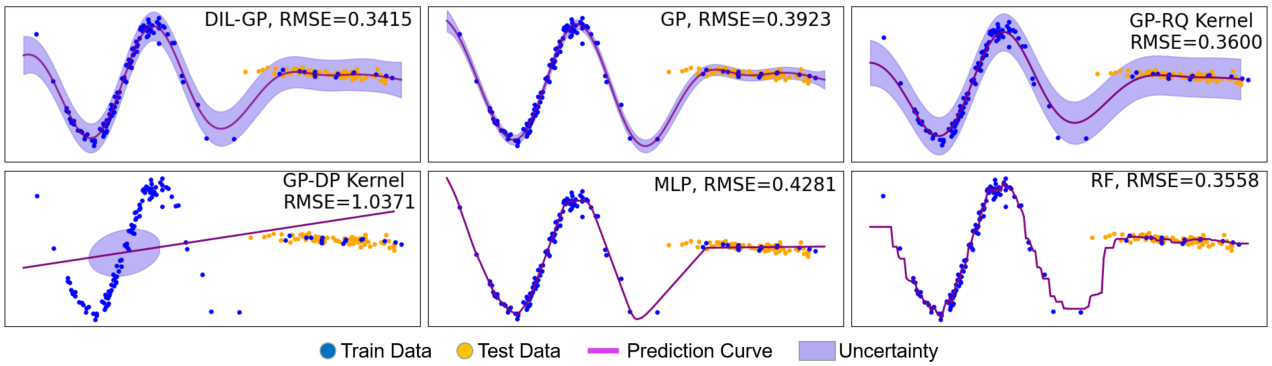

Figure 1: Results on the one-dim synthetic dataset.

Algorithm 2: DIL-GP Bayesian Optimization

Require: Black-box function to be optimized $f(\beta)$, Search space β, Acquisition function α,number of Bayesian optimization steps T_{BO}.

Ensure: Optimized solution β

1: Randomly sample initial trial points $\boldsymbol{\beta} \sim \beta$.
2: Compute the initial queried data point set $D_0 = (\boldsymbol{\beta}, f(\boldsymbol{\beta}))$
3: **for** $t = 0$ **to** $T_{\text{BO}} - 1$ **do**
4: Use Algorithm 1 to fit a GP model: DIL-GP(D_t);
5: Find the next trial point with DIL-GP(D_t) using the acquisition function:

$$\hat{\beta} = \arg\min_{\beta \in \beta} \alpha(\beta \mid \text{DIL-GP}(D_t))$$

6: Evaluate $f(\hat{\beta})$;
7: $D_{t+1} \leftarrow D_t \cup (\hat{\beta}, f(\hat{\beta}))$;
8: **end for**
9: Output the optimized solution $\hat{\beta} = \arg\min_{\beta \in D_t} f(\beta)$;

$\sigma\sqrt{2(\gamma_{t-1} + 1 + \log(4/\delta)}$, with probability$\geq 1 - \delta/4$, the upperbound of the cumulative regret R_T satisfies:

$$R_T = \sum_{t=1}^{T} r_t \leq \beta_T \sqrt{C_1 T \gamma_T} \tag{8}$$

4 Experiments

In the experiments, we compare the proposed DIL-GP with GP with Gaussian kernel (GP), GP with the rational quadratic kernel (GP-RQ Kernel)(MacKay 2003), GP with the dot product kernel (GP-DP Kernel)(Rasmussen and Williams 2006), random forest (RF)(Breiman 2001), and multi-layer perceptron (MLP)(Rumelhart et al. 1986). For DIL-GP, we use the Gaussian kernel for all experiments to demonstrate the algorithms' generalization ability without the need for special kernel treatments. Detailed settings can be found in the Appendix.

4.1 Synthetic Dataset

We first generate a one-dim synthetic dataset to demonstrate the effectiveness of our model. This dataset comprises two clusters of data with Gaussian distributions, which simulate two distinct domains. Specifically, the first cluster $X_1 \sim \mathcal{N}(0, 1)$, and the second cluster $X_2 \sim \mathcal{N}(6.5, 1)$. Our training set consists of one hundred samples from cluster X_1 and fifteen samples from cluster X_2 while the test set consists of eighty samples from cluster X_2. This setting generates a typical non-i.i.d distribution shifts scenario. The rooted mean square error (RMSE) is used as the comparison metric. For GP-related methods, we define another uncertainty metric—coverage rate, which represents the proportion of test set data that falls within the standard deviation region for GP models. When RMSE values are similar, higher coverage rates on the test set show better uncertainty discovery. Our results, presented in Figure 1, reveal that GP baselines, MLP, and RF all overfit on the first cluster. Moreover, the GP baselines exhibit overconfidence on the data, without considering the heterogeneity, leading to overfitting and worse test set performances, as demonstrated in Table 1. The GP-DP kernel fails to fit the data, indicating that changing kernels may improve GP's generalization performances on special applications but may not be generalizable to other domains.

We further demonstrate the effectiveness of our model on a two-dimensional dataset comprising more complex functions. Specifically, we sample two distinct clusters from Gaussian distributions with varying means and variances, and generate labels using a trigonometric function (Details in Appendix). The first cluster $X_1 \sim \mathcal{N}(\begin{bmatrix}0.3\\0.3\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 0.01\end{bmatrix})$, and the second cluster $X_2 \sim \mathcal{N}(\begin{bmatrix}0.7\\0.7\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 0.01\end{bmatrix})$.The training set consists of one hundred samples from cluster one and fifteen samples from cluster two, while the test set solely comprises eighty samples from cluster two. We benchmark all methods to predict the two-dimensional data and present the experimental results in Table 1. The experimental results on the 2D dataset show that MLP suffers great performance degradation when facing heterogeneous data. RF, GP baseline, GP-RQ kernel

	1-Dim Dataset		2-Dim Dataset	
Method	RMSE	Coverage Rate	RMSE	Coverage Rate
Random Forest	0.3558 ± 0.0840	-	0.7938 ± 0.1841	-
MLP	0.4321 ± 0.0427	-	0.9133 ± 0.4151	-
GP	0.3923 ± 0.0153	0.9125 ± 0.0374	0.8022 ± 0.1373	0.9225 ± 0.0537
GP-RQ Kernel	0.3600 ± 0.0064	0.9625 ± 0.1256	0.7921 ± 0.2107	0.8732 ± 0.1436
GP-DP Kernel	1.0371 ± 0.0021	0 ± 0	0.7133 ± 0.0096	0.6339 ± 0.2311
DIL-GP	$\mathbf{0.3415 \pm 0.0109}$	0.9625 ± 0.1000	$\mathbf{0.6583 \pm 0.0230}$	0.8822 ± 0.0775

Table 1: Quantitative comparison between different methods on synthetic datasets (Mean $\pm$ Max deviation over 5 runs)

	King Housing Dataset				Automobile Dataset
Method	domain1	domain2	domain3	domain4	test dataset
Random Forest	0.450 ± 0.023	0.560 ± 0.031	0.623 ± 0.106	0.890 ± 0.037	0.992 ± 0.206
MLP	0.536 ± 0.137	0.650 ± 0.347	0.787 ± 0.762	0.913 ± 0.989	0.983 ± 0.072
GP	0.371 ± 0.009	0.594 ± 0.056	0.800 ± 0.084	0.989 ± 0.100	0.889 ± 0.008
GP - RQ Kernel	0.365 ± 0.016	0.442 ± 0.093	0.604 ± 0.173	0.901 ± 0.075	0.847 ± 0.351
GP - DP Kernel	0.556 ± 0.007	0.663 ± 0.006	0.812 ± 0.010	1.056 ± 0.012	0.938 ± 0.013
DIL-GP	$\mathbf{0.291 \pm 0.016}$	$\mathbf{0.371 \pm 0.011}$	$\mathbf{0.489 \pm 0.042}$	$\mathbf{0.543 \pm 0.058}$	$\mathbf{0.837 \pm 0.069}$

Table 2: Quantitative comparison of RMSE between different methods on King Housing Dataset and Automobile Dataset (Mean $\pm$ Max deviation over 5 runs)

Method	Hover	Fig-8	Sin-forward	Spiral-up
Random Forest	0.3740 ± 0.0273	0.4444 ± 0.0252	0.3962 ± 0.0036	0.3871 ± 0.0310
MLP	0.3735 ± 0.0391	0.4433 ± 0.0517	0.3582 ± 0.0058	0.3733 ± 0.0399
GP	0.3739 ± 0.0187	0.4433 ± 0.0241	0.3579 ± 0.0032	0.3727 ± 0.0333
GP-RQ Kernel	0.3733 ± 0.0213	0.4492 ± 0.0293	0.3572 ± 0.0040	0.3724 ± 0.0283
GP-DP Kernel	0.3731 ± 0.0238	0.4665 ± 0.0371	0.3469 ± 0.0052	0.3474 ± 0.0222
DIL-GP	$\mathbf{0.3610 \pm 0.0200}$	$\mathbf{0.4081 \pm 0.0262}$	$\mathbf{0.3367 \pm 0.0043}$	$\mathbf{0.3349 \pm 0.0275}$

Table 3: Errors between the actual trajectory and the desired trajectory (Mean $\pm$ Max deviation over 5 runs)

and GP-DP kernel also pose significant gaps with DIL-GP due to the lack of domain generalization capability.

In the two-dimensional generative dataset, DIL-GP shows more obvious advantages compared to the one-dimensional data. This may be due to the fact that higher dimensional data domains contain synergistic relationships with each other, while the domain invariant learning in DIL-GP learns more robust representations for OoD generalization. Subsequently, we conduct experiments on more complex real-world dataset to check its performance.

4.2 Real-world Datasets

We test the practicability of the algorithm on real-world datasets. The first dataset we consider is a regression dataset (Kaggle) of house sales prices from King County, USA. In this house prices predicting dataset, houses built in different periods are considered as different domains. Houses built in different periods possess distinct construction materials and are affected by the cutural heritage values. This complicates the relationship between domain and selling price. Therefore, solving the out-of-distribution problem is a challenging task.

We sample a dataset of size 1304 from it, using the 17 variables including the house size, number of floors and the latitude and longitude locations to predict the transaction prices of the house. To simulate non-i.i.d. distribution shifts, we split the dataset according to the built year of the houses. We divide the dataset into five domains, using data with build year from 1980 to 2015 as the training dataset (D0) and build year from 1900 to 1979 as test dataset. The test dataset is further divided into four domains with a span of 20 years (D1-D4). The motivation for selecting earlier-built houses as the test set is that the value of houses built in earlier periods is harder to predict, making the problem more challenging (Klein 1975). Our results, presented in Figure 2, demonstrate that DIL-GP achieves good results especially in domains with a long time interval from the training set. DIL-GP is also more stable across multiple domains, demonstrat-

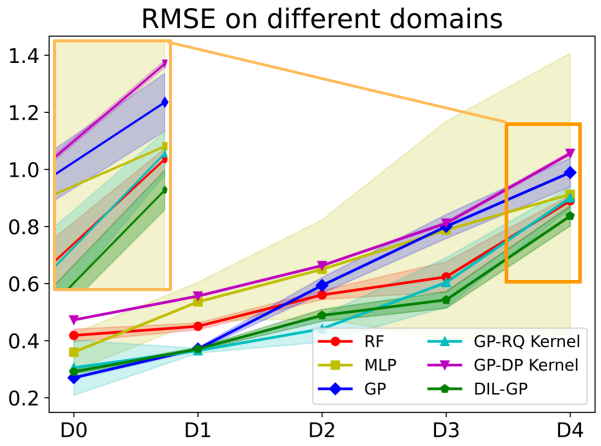

Figure 2: RMSE of methods on different domains (D0-D4) (Mean $\pm \frac{1}{2}$ Max deviation over 5 runs).

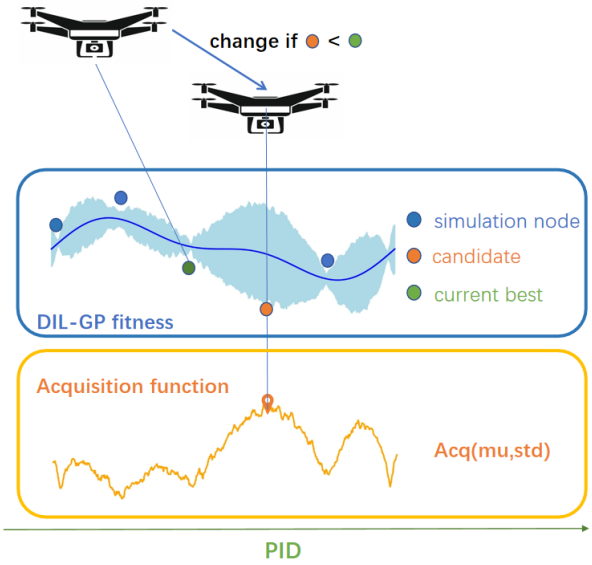

Figure 3: Bayesian optimization for PID tuning. A GP function is fitted to the existing data points, and a scoring function is used to compute scores for various parameter settings. The highest-scoring parameter setting is selected as the candidate PID and added to the data points. If this candidate outperforms the current best, it becomes the new best. This process is iteratively repeated until the termination condition is met.

ing its effectiveness in OOD generalization. MLP outperforms DIL-GP in some experiments but suffers significant instability. We provide quantitative results in Table 2.

The second real-world dataset under consideration is the Automobile dataset from UCI datasets with more than one hundred samples with fourteen dimensions' features (including fuel-type, aspiration, body-style, compression-ratio, etc) for insurance risk rating of automotive types. The term "insurance rating" refers to the extent to which an automotive type is more risky than its market price indicates in actuarial science. Different types of automobiles have implications for various usage scenarios and demand-supply relation ships and so on. It is reasonable to consider the types of automobiles as different domains. However, it is hard to make a clear statement which specific types of cars should be grouped in the same domain to train a good predictor, making it a challenging problem for previous methods reliant on domain labels. For evaluation, we use the "sedan" and "hardtop" types as the training domains, while "wagon", "hatchback", and "convertible" types are for test domains, which we do not provide for training algorithms. As shown in Table 1, although the domain labels are not provided in the training as previous OOD generalization methods, DIL-GP attains the lowest RMSE on the novel domains. This further verifies DIL-GP's min-max formulations' adaptability for practical problems and its OOD generalization abilities under diverse kinds of distribution shifts.

4.3 Bayesian Optimization for Quadrotors PID Tuning

The Proportional-Integral-Derivative (PID) control is a widely used feedback control algorithm that has been applied in various applications, such as quadrotor control (Johnson and Moradi 2005). This control algorithm employs the weighted feedback of the errors, consisting of three components: proportional, integral, and derivative of errors. PID control takes in sensor data and determines the motor speeds of rotors for quadrotor flight control. Tuning the PID parameters is tedious but challenging due to the complex dynamics, sensitivity to changes and external influences. It usually requires human experts to achieve optimum performance: the weights for the three components need to be carefully tuned to achieve stable control during quadrotor flight under turbulent environments (Shi et al. 2021). This makes Bayesian optimization suitable for this problem, as it tunes the parameters in a black-box manner. At the same time, the fact that the sensor data is quickly changing during flight, making it hard to distinguish any environment information. A well-designed approach needs to enable the optimization to obtain a PID that still achieves good performance on environments that are less seen in the training environment. In our experiment, we simulate the flight process of a quadrotor and use the PID control to fly the quadrotor along the set waypoints. We use Bayesian optimization to optimize the weights for PID control to minimize the averaged control errors (ACEs) defined as the mean squared error between the true coordinates and the desired waypoints. During the quadrotor simulation, the wind force is simulated with the widely-used Dryden model (Specification 1980) to determine the wind speed in three directions at each time slot. We simulate two different wind domains. The first wind domain is for winds that have a mean value of zero and change more rapidly, and the second wind domain is for winds that have a non-zero mean value and are more stable. This is to test whether Bayesian optimization can find suitable PID weights to enable quadrotor to remain stable even under unseen environment. Representing the wind force as a binary

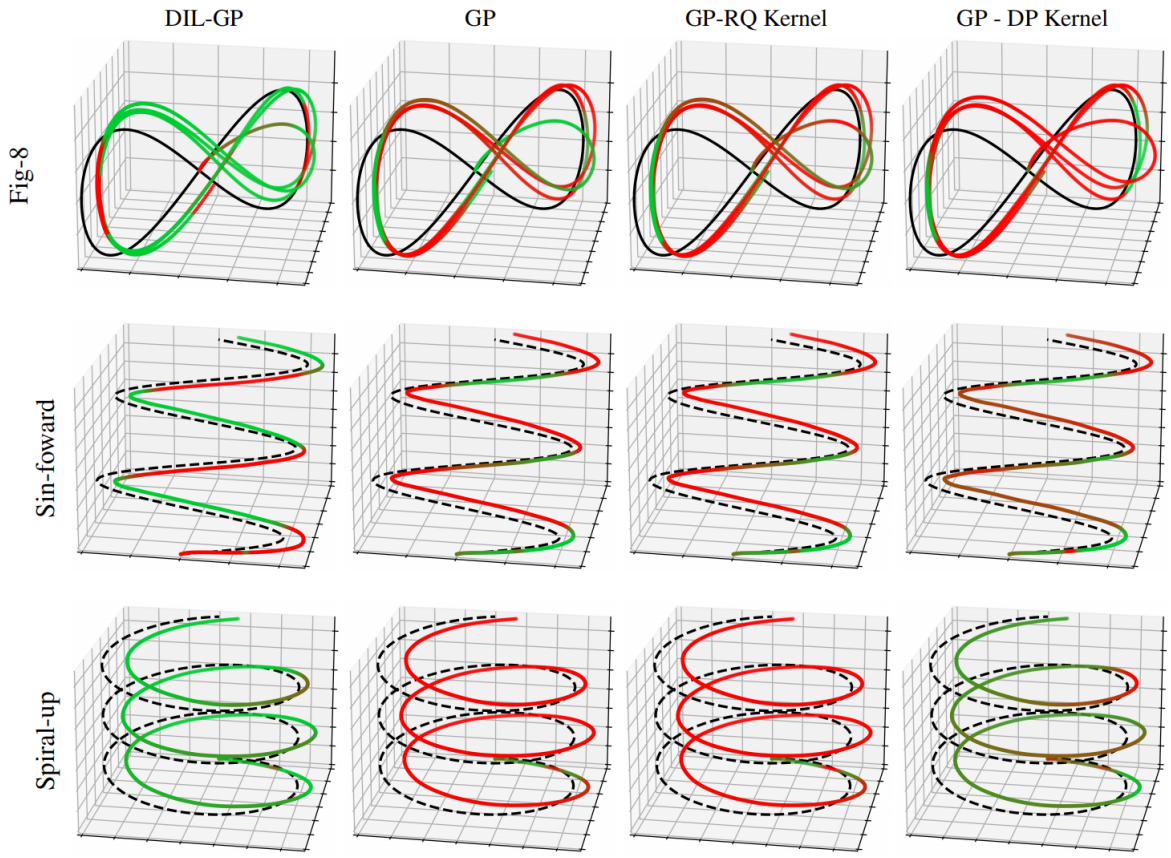

Figure 4: Visualizations of the quadrotor flight simulations on various trajectories. The black line is the desired trajectory. The green line denotes lower deviation from the desired trajectory while the red line denotes farther distances. BO using DIL-GP finds the best PID parameters, which enable accurate controls across all trajectories types under challenging turbulent wind conditions.

in the horizontal and vertical directions, the wind was experimentally distributed uniformly with a mean of (0,0) and a variance of (5,2.5) in the first domain, and with a mean of (3,1) and a variance of (2,1) in the second domain.

For comparison, we implement Bayesian optimization's surrogate model with DIL-GP, other GP variants, RF, and MLP. For GP-related methods,we use the upper confidence bound method as the acquisition function. The overall flow of the experiment is shown in Figure 3. Bayesian optimization alternatively finds the next PID weights for trial and updates the surrogate model to predict the next PID weights for trial. Three flight trajectories obtained by PID optimized with different methods are shown in Figure 4. Other trajectories results can also be found in Appendix. The results shown are the mean results of five experiments, each run with a different random seed. The PID parameters obtained through DIL-GP Bayesian optimization demonstrates superior control performance on the quadrotor in unseen domain, with smaller control errors in the flight trajectory. The numerical comparison of all six methods is presented in Table 3. As shown in Table 3, the variations in wind environments across the different experiments had a significant impact on the error values, resulting in larger deviations. Due to limited OOD generalization abilities, baseline surrogate models struggles in predicting better solutions, while DIL-GP Bayesian optimization finds robust PID weights, with which the quadrotor can follow the desired paths more accurately. The error-iteration curves of ACE during Bayesian optimization are in Appendix for reference.

5 Conclusion

In this paper, we propose a domain invariant learning approach for Gaussian processes (DIL-GP) and its Bayesian optimization extension to improve their generalization abilities. Numerical experiments under challenging synthetic and real-world datasets demonstrate the effectiveness of the proposed min-max formulation of domain invariant learning for Gaussian processes. With the proposed framework, DIL-GP automatically discovers the heterogeneity in the data and achieves OOD generalization on various benchmarks. We further demonstrate that the Bayesian optimization algorithm with DIL-GP's superiority in a PID tuning problem.

Acknowledgments

Nanyang Ye was supported in part by National Natural Science Foundation of China under Grant No.62106139, 61960206002, 62272301, 62020106005, 62061146002, 62032020, in part by Shanghai Artificial Intelligence Laboratory and the National Key R&D Program of China (Grant NO.2022ZD0160100).

References

Achituve, I.; Shamsian, A.; Navon, A.; Chechik, G.; and Fetaya, E. 2021. Personalized Federated Learning With Gaussian Processes. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 8392–8406. Curran Associates, Inc.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Astudillo, R.; Jiang, D.; Balandat, M.; Bakshy, E.; and Frazier, P. 2021. Multi-Step Budgeted Bayesian Optimization with Unknown Evaluation Costs. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 20197–20209. Curran Associates, Inc.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Cai, X.; and Scarlett, J. 2021. On Lower Bounds for Standard and Robust Gaussian Process Bandit Optimization. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 1216–1226. PMLR.

Chen, K.; Yin, F.; and Cui, S. 2023. Compressible spectral mixture kernels with sparse dependency structures for Gaussian processes. *Signal Processing*, 213: 109179.

Chen, W.; Tripp, A.; and Hernández-Lobato, J. M. 2023. Meta-learning Adaptive Deep Kernel Gaussian Processes for Molecular Property Prediction. In *The Eleventh International Conference on Learning Representations*.

Cohen, M. K.; Daulton, S.; and Osborne, M. A. 2022. Log-Linear-Time Gaussian Processes Using Binary Tree Kernels. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Daulton, S.; Balandat, M.; and Bakshy, E. 2021. Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 2187–2200. Curran Associates, Inc.

Dong, Q.; Muhammad, A.; Zhou, F.; Xie, C.; Hu, T.; Yang, Y.; Bae, S.-H.; and Li, Z. 2022. ZooD: Exploiting Model Zoo for Out-of-Distribution Generalization. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 31583–31598. Curran Associates, Inc.

Gahungu, P.; Lanyon, C.; Álvarez, M. A.; Bainomugisha, E.; Smith, M. T.; and Wilkinson, R. 2022. Adjoint-aided inference of Gaussian process driven differential equations. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 17233–17247. Curran Associates, Inc.

Hendrycks, D.; Liu, X.; Wallace, E.; Dziedzic, A.; Krishnan, R.; and Song, D. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Huang, Z.; Wang, H.; Huang, D.; Lee, Y. J.; and Xing, E. P. 2022. The Two Dimensions of Worst-case Training and Their Integrated Effect for Out-of-domain Generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 9621–9631. IEEE.

Hvarfner, C.; Hutter, F.; and Nardi, L. 2022. Joint Entropy Search For Maximally-Informed Bayesian Optimization. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Johnson, M. A.; and Moradi, M. H. 2005. *PID control*. Springer.

Jørgensen, M.; and Osborne, M. A. 2022. Bezier Gaussian Processes for Tall and Wide Data. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 24354–24366. Curran Associates, Inc.

Klein, B. 1975. Our new monetary standard: the measurement and effects of price uncertainty, 1880–1973. *Economic Inquiry*, 13(4): 461–484.

Lalchand, V.; Bruinsma, W.; Burt, D. R.; and Rasmussen, C. E. 2022. Sparse Gaussian Process Hyperparameters: Optimize or Integrate? In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Lei, B.; Kirk, T. Q.; Bhattacharya, A.; Pati, D.; Qian, X.; Arroyave, R.; and Mallick, B. K. 2021. Bayesian optimization with adaptive surrogate models for automated experimental design. *Npj Computational Materials*, 7(1): 194.

Lu, X.; Boukouvalas, A.; and Hensman, J. 2022. Additive Gaussian Processes Revisited. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 14358–14383. PMLR.

Luo, H.; Nattino, G.; and Pratola, M. T. 2022. Sparse Additive Gaussian Process Regression. *Journal of Machine Learning Research*, 23(61): 1–34.

MacKay, D. J. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

Moreno-Muñoz, P.; Feldager, C.; and Hauberg, S. r. 2022. Revisiting Active Sets for Gaussian Process Decoders. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 6603–6614. Curran Associates, Inc.

Niu, Y.; Chen, L.; Zhou, C.; and Zhang, H. 2022. Respecting Transfer Gap in Knowledge Distillation. In *NeurIPS*.

Pilario, K. E. S.; Ching, P. M. L.; Calapatia, A. M. A.; and Culaba, A. B. 2022. Predicting Drying Curves in Algal Biorefineries using Gaussian Process Autoregressive Models. *Digital Chemical Engineering*, 4: 100036.

Rasmussen, C. E.; and Williams, C. K. I. 2006. *Gaussian processes for machine learning.* Adaptive computation and machine learning. MIT Press. ISBN 026218253X.

Rumelhart, D. E.; Hinton, G. E.; McClelland, J. L.; et al. 1986. A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(45-76): 26.

Sagawa*, S.; Koh*, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.

Schmidt, A.; Morales-Álvarez, P.; and Molina, R. 2023. Probabilistic Attention Based on Gaussian Processes for Deep Multiple Instance Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Shi, G.; Azizzadenesheli, K.; O' Connell, M.; Chung, S.-J.; and Yue, Y. 2021. Meta-Adaptive Nonlinear Control: Theory and Algorithms. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 10013–10025. Curran Associates, Inc.

Shi, W.; Yu, D.; and Yu, Q. 2021. A Gaussian Process-Bayesian Bernoulli Mixture Model for Multi-Label Active Learning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 27542–27554. Curran Associates, Inc.

Specification, M. 1980. Flying qualities of piloted airplanes. Technical report, MIL-F-8785C.

Sun, T.; Wang, W.; Jing, L.; Cui, Y.; Song, X.; and Nie, L. 2022. Counterfactual reasoning for out-of-distribution multimodal sentiment analysis. In *Proceedings of the 30th ACM International Conference on Multimedia*, 15–23.

Wang, T.; Yue, Z.; Huang, J.; Sun, Q.; and Zhang, H. 2021. Self-Supervised Learning Disentangled Group Representation as Feature. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 18225–18240.

Wuethrich, M.; Schölkopf, B.; and Krause, A. 2021. Regret Bounds for Gaussian-Process Optimization in Large Domains. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 7385–7396. Curran Associates, Inc.

Yang, L.; Zhang, S.; Qin, L.; Li, Y.; Wang, Y.; Liu, H.; Wang, J.; Xie, X.; and Zhang, Y. 2022. GLUE-X: Evaluating Natural Language Understanding Models from an Out-of-distribution Generalization Perspective. *arXiv preprint arXiv:2211.08073*.

Zhang, H.; Zhang, Y.; Liu, W.; Weller, A.; Schölkopf, B.; and Xing, E. P. 2022. Towards Principled Disentanglement for Domain Generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 8014–8024. IEEE.

Zhang, Y.; Zhang, X.; and Frazier, P. 2021. Constrained Two-step Look-Ahead Bayesian Optimization. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 12563–12575. Curran Associates, Inc.

Zhao, G.; Dougherty, E.; Yoon, B.-J.; Alexander, F.; and Qian, X. 2021. Efficient Active Learning for Gaussian Process Classification by Error Reduction. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 9734–9746. Curran Associates, Inc.