Gaussian Processes for Shuffled Regression

Masahiro Kohjima
NTT, Inc.
masahiro.kohjima@ntt.com

Abstract

Shuffled regression is the problem of learning regression functions from shuffled data where the correspondence between the input features and target response is unknown. This paper proposes a probabilistic model for shuffled regression called Gaussian Process Shuffled Regression (GPSR). By introducing Gaussian processes as a prior of regression functions in function space via the kernel function, GPSR can express a wide variety of functions in a nonparametric manner while quantifying the uncertainty of the prediction. By adopting the Bayesian evidence maximization framework and a theoretical analysis of the connection between the marginal likelihood/predictive distribution of GPSR and that of standard Gaussian process regression (GPR), we derive an easy-to-implement inference algorithm for GPSR that iteratively applies GPR and updates the input-output correspondence. To reduce computation costs and obtain closed-form solutions for correspondence updates, we also develop a sparse approximate variant of GPSR using its weight space formulation, which can be seen as Bayesian shuffled linear regression with random Fourier features. Experiments on benchmark datasets confirm the effectiveness of our GPSR proposal.

1 Introduction

The purpose of shuffled regression (SR) is to learn regression functions from shuffled data where the correspondence between the input features and the target response is unknown [1, 2, 3, 4]. This situation often arises with data collected independently from multiple devices/viewpoints [4, 5, 6] or when personal information must be anonymized for privacy reasons, such as in cases involving clinical history [7]. Since traditional supervised regression methods, collectively referred to as coupled regression (CR), rely on data with input-output correspondences (the pairs of features and responses) and so cannot deal with shuffled data, SR has become the focus of recent studies [8, 9, 10, 11, 12].

In SR literature, main focus is the use of linear models [1, 2, 4, 9] (an exception is [12] which uses neural networks as explained in §2). Accordingly, the use of Gaussian processes (GPs) [13] has not been well investigated. Considering the known benefits of GPs such as nonparametric flexibility and uncertainty quantification, efforts are needed to realize their benefits in SR.

This study proposes a probabilistic model based on GPs for SR called Gaussian Process Shuffled Regression (GPSR). By introducing GPs as a prior of regression functions in function space via the kernel function, GPSR can express wide variety of functions in a nonparametric manner while quantifying prediction uncertainty as shown in Table 1.

By use of the Bayesian evidence maximization framework and a theoretical analysis of the connection between the marginal likelihood/predictive distribution of GPSR and that of standard Gaussian process regression (GPR), we derive an easy-to-implement inference algorithm for GPSR using GPR as a subroutine; it iteratively applies GPR for the optimization of the kernel's hyperparameters and solves the quadratic assignment problem (QAP) for the updates of the input-output correspondence. Although this algorithm is valid for small-scale data, it seems prohibitive for handling medium/large-

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Table 1: Comparison of the proposed method (GPSR) with Gaussian process regression (GPR) and the existing methods for shuffled regression (SR). Item (a) indicates which methods can handle shuffled data, while item (b) indicates which methods can express non-linear regression functions. Finally, item (c) shows the methods that can quantify the uncertainty of their prediction.

	GPR [13]	SLR [4]	SDR [12]	GPSR (ours)
(a) shuffled reg.		✓	✓	✓
(b) nonlinear	✓		✓	✓
(c) uncertainty	✓			✓

Table 2: Classification of GPSR, GPR, Bayesian Linear Regression (BLR) and Bayesian Shuffled Linear Regression (BSLR). We establish the connection between these four methods.

		Problem Type CR	Problem Type SR
Model	function space	GPR	GPSR
	weight space	BLR	BSLR

scale data due to the difficulty of solving QAP (QAP is NP-hard in general) and the computation cost of GPR, which scales cubically with the number of samples. Therefore, we also develop a sparse approximate variant of GPSR called Sparse Spectrum GPSR (SS-GPSR) using the weight space formulation of GPSR. This can be seen as Bayesian Shuffled Linear Regression (BSLR) with the random Fourier features, and allows us to reduce both memory and computation cost and to use sorting operations to obtain closed-form solutions of correspondence updating. Note that both developing an algorithm for BSLR and clarifying the relations among four methods in Table 2 are also our contributions. Experiments on benchmark datasets confirm the effectiveness of our proposals.

2 Related Works

This work is positioned within the literature of shuffled regression and that of Gaussian processes. We describe below prior works in these research lines.

Background of shuffled regression (SR). SR [3, 4] which is also called "regression without correspondence" [2, 14], "uncoupled regression" [15, 16], "permuted and unlinked regression" [9, 11], arises in various fields, including flow cytometry for measuring chemical characteristics of cells [4], image/point cloud registration [5, 6], and linkage of health records [7]. Thus various theoretical aspects and problem settings of SR have been studied. For example, Pananjady et al. show that recovering the correspondence between input and output is NP-hard in general [3]; the conditions under which unique solutions can be achieved have been explored [17]. Some existing works consider the setting with additional data such as pairwise comparison data [10, 15, 16, 18]. We focus on the setting where no additional data is, only the shuffled data.

Models and algorithms for SR. Earlier works on SR mainly use linear models [3, 4, 8, 9] and are categorized as shuffled linear regression (SLR). Only in recent years have approaches that utilize more flexible models, such as the neural-network-based method called shuffled deep regression (SDR) [12], been considered, and our study using GPs follows this research line. For comparison, see Table 1. For parameter estimation, variants of the expectation-maximization (EM) [19] are frequently used, e.g., [4, 9, 12, 20]. In contrast, our inference algorithm is derived from the Bayesian evidence maximization framework [21, 22] which has, up to now, not been used in the context of SR. Since the evidence penalizes overly complex models by integrating over parameters, this algorithm offers the way of hyperparameter tuning without validation data while still avoiding overfitting.

Gaussian processes (GPs). For supervised regression, Gaussian process regression (GPR) provides nonparametric flexibility and the quantification of uncertainty. However, naive GPR has limited practicality because its computation cost scales to the cube of the number of data points. For handling datasets with more than a few thousand points, various scalable approaches such as using inducing points [23, 24, 25, 26], grid points [27, 28], and spectral points or called random Fourier features (RFF) [29, 30] have been proposed. These so-called sparse GPs are still an active part of GP research e.g., [31, 32, 33]. Our study adopts the approach using spectral points/RFF [29, 30] to develop SS-GPSR, not only to reduce computation cost, but also because equations for updating the input-output correspondence can be obtained as closed-form solutions via a sorting operation.

3 Preliminaries

In this section, we provide definitions of shuffled data and its connection to coupled data. We also review Gaussian process regression (GPR) for "coupled" data used in supervised regression. All symbols are listed in the Appendix A.

Definition of Shuffled Data. Let $\mathcal{X} \subseteq \mathbb{R}^D$ and $\mathcal{Y} \subseteq \mathbb{R}$ be the input space and response space, respectively. Shuffled data $\mathcal{D}_{\mathrm{SD}}$ are defined by pairs of input-set $\mathbf{X}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{iL}\}$ and response-set $\boldsymbol{y}_i = \{y_{i1}, y_{i2}, \cdots, y_{iL}\}$, i.e., $\mathcal{D}_{\mathrm{SD}} = \{(\mathbf{X}_i, \boldsymbol{y}_i)\}_{i=1}^M$, where $\mathbf{x}_{i\ell} \in \mathcal{X}$, $y_{i\ell} \in \mathcal{Y}$, L is set size [1], and M is the number of pairs. Unlike "coupled" data, the correspondence between input and response is unknown in shuffled data; the response corresponding to input $\mathbf{x}_{i\ell}$ is one of the elements in response-set $\boldsymbol{y}_i = \{y_{i1}, ..., y_{iL}\}$, but it is not known which one. For simplicity, we define the total number of inputs/responses in $\mathcal{D}_{\mathrm{SD}}$ as $N \triangleq ML$ and denote all inputs and all responses as $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^M$, and $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i=1}^M$, respectively. Also we equate symbol $\boldsymbol{y}_i$ with L-dimensional column vector $\mathbf{y}_i = (y_{i1}, \cdots, y_{iL})^\top$, and symbol $\boldsymbol{y}$ with N-dimensional column vector $\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_M^\top)^\top$ whose $\{(i-1)L + \ell\}$-th element is $y_{i\ell}$.

Connection between Shuffled Data and Coupled Data. Shuffled data $\mathcal{D}_{\mathrm{SD}}$ are regarded as a general representation of coupled data $\mathcal{D}_{\mathrm{CD}}$ defined as pairs of input $\mathbf{x}_i \in \mathcal{X}$ and response $y_i \in \mathcal{Y}$, $\mathcal{D}_{\mathrm{CD}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. This is because shuffled data with set size $L = 1$ are equivalent to coupled data. To distinguish it from shuffled data, bold symbols/vectors, etc. related to couple data will henceforth be underlined; we denote all inputs and all responses in $\mathcal{D}_{\mathrm{CD}}$ as $\underline{\mathbf{X}} = \{\mathbf{x}_i\}_{i=1}^N$ and $\underline{\mathbf{y}} = (y_1, \cdots, y_n)^\top$, respectively. Note that early works on shuffled regression such as [2, 3, 4] focus on shuffled data where the number of pairs is $M = 1$. In practical scenarios of data collection, data are often collected multiple times by repeated measurements or by changing the target group of users; even if the input-response correspondence is obscured we can know from which measurement/user-groups the data was collected, so we consider shuffled data where the number of pairs is $M > 1$.

Gaussian Process Regression (GPR) for Coupled Data [13]. GPs are collection of random variables $\{f(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$ for which any finite subset follows a multivariate Gaussian specified by mean function $m(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ and kernel/covariance function $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. GPR for coupled data uses (zero-mean) GPs with kernel k_θ depending on hyperparameter θ as *priors over function* f, which is denoted as $f \sim \mathcal{GP}(0, k_\theta(\mathbf{x}, \mathbf{x}'))$. Each response y_i is assumed to be obtained by adding Gaussian noise to $\mathrm{f}_i = f(\mathbf{x}_i)$, i.e., $y_i = \mathrm{f}_i + \epsilon$, where $\epsilon \sim \mathcal{N}(\epsilon; 0, \beta^{-1})$ and β is the precision. Introducing function values vector $\underline{\mathbf{f}} = (\mathrm{f}_1, \cdots, \mathrm{f}_N)^\top$, this data generative process is summarized as $p(\underline{\mathbf{f}} | \underline{\mathbf{X}}) = \mathcal{N}(\underline{\mathbf{f}}; \mathbf{0}, \underline{\mathbf{K}})$ and $p(\underline{\mathbf{y}} | \underline{\mathbf{f}}) = \mathcal{N}(\underline{\mathbf{y}}; \underline{\mathbf{f}}, \beta^{-1}\mathbf{I}_N)$, where $\underline{\mathbf{K}}$ is the gram matrix whose (i, j)-th element is $k_\theta(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{I}_N$ is the $N \times N$ identity matrix. Figure 1a shows the graphical model representation of GPR. We get the (analytically-tractable) marginal likelihood by integrating over latent function values $\underline{\mathbf{f}}$, $p(\underline{\mathbf{y}} | \underline{\mathbf{X}}) = \int p(\underline{\mathbf{y}} | \underline{\mathbf{f}}) p(\underline{\mathbf{f}} | \underline{\mathbf{X}}) d\underline{\mathbf{f}} = \mathcal{N}(\underline{\mathbf{f}}; \mathbf{0}, \underline{\mathbf{C}}_\theta)$ where $\underline{\mathbf{C}}_\theta = \underline{\mathbf{K}} + \beta^{-1}\mathbf{I}_N$. Thus hyperparameter θ is estimated by the evidence framework [21, 22][2] that maximizes the logarithm of marginal likelihood (evidence) defined as

$$(\text{GPR Marginal L.})\ \mathcal{L}_{\mathrm{GPR}}(\theta; \mathcal{D}_{\mathrm{CD}}) \triangleq \log p(\underline{\mathbf{y}} | \underline{\mathbf{X}}) = -\frac{1}{2} \log |\underline{\mathbf{C}}_\theta| - \frac{1}{2} \underline{\mathbf{y}}^\top \underline{\mathbf{C}}_\theta^{-1} \underline{\mathbf{y}} - \frac{N}{2} \log(2\pi). \quad (1)$$

Similarly, we get the predictive distribution of response y_* for test input $\mathbf{x}_*$ as follows.

$$\begin{aligned} (\text{GPR Prediction}) \quad & p_{\mathrm{GPR}}(y_* | \mathbf{x}_*, \mathcal{D}_{\mathrm{CD}}) = \mathcal{N}(y_*; m_{\mathrm{GPR}}(\mathbf{x}_*), \sigma^2_{\mathrm{GPR}}(\mathbf{x}_*)), \\ & m_{\mathrm{GPR}}(\mathbf{x}_*) = \underline{\mathbf{k}}^\top \underline{\mathbf{C}}_\theta^{-1} \underline{\mathbf{y}}, \quad \sigma^2_{\mathrm{GPR}}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) + \beta^{-1} - \underline{\mathbf{k}}^\top \underline{\mathbf{C}}_\theta^{-1} \underline{\mathbf{k}}, \end{aligned} \quad (2)$$

where $\underline{\mathbf{k}}$ is an N-dimensional vector defined by $\underline{\mathbf{k}} = (k_\theta(\mathbf{x}_*, \mathbf{x}_1), \cdots, k_\theta(\mathbf{x}_*, \mathbf{x}_N))^\top$. Thus GPR can make predictions with information on confidence levels.

Spectral Representation of Shift Invariant Kernel. One bottleneck of GPs is that, in a direct implementation, the memory and computation requirements for inverting covariance $\underline{\mathbf{C}}_\theta$ scale as $O(N^2)$ and $O(N^3)$, respectively. Accordingly, Lazaro et al. [29] use the spectral representation or the

[1] This could be extended to a setup where L is different for each sample i but that is omitted for simplicity.

[2] Also called empirical Bayes, Type-II maximum likelihood or generalized maximum likelihood [34, 35].

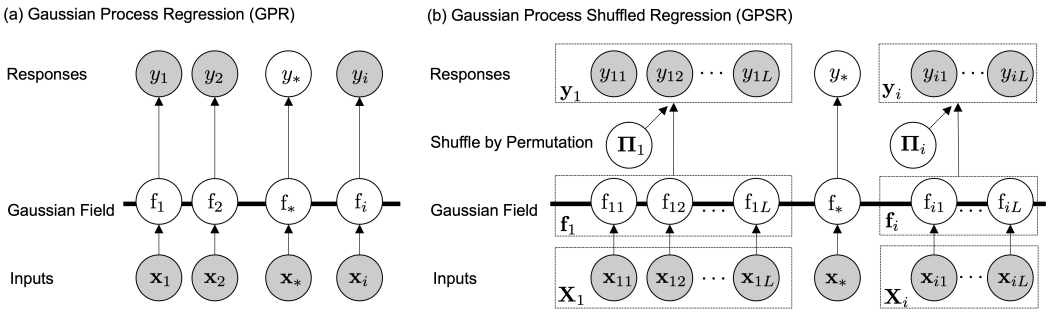


Figure 1: Graphical models of (a) Gaussian process regression (GPR) and (b) proposed Gaussian process shuffled regression (GPSR). Shaded nodes represent observed variables. The thick horizontal bar represents a set of fully connected nodes. Dependency on hyperparameters is omitted. GPSR includes GPR as a special case since these two are identical when set size $L = 1$.

so called random Fourier features (RFF) $\psi : \mathcal{X} \to \mathcal{F}_\psi \subseteq \mathbb{R}^H$ that approximate the (shift-invariant) kernel function,

$$k(\mathbf{x}, \mathbf{x}') \approx \boldsymbol{\psi}(\mathbf{x})^\top \psi(\mathbf{x}'),\ \boldsymbol{\psi}(\mathbf{x}) = (\cos(2\pi \mathbf{s}_1^\top \mathbf{x}), \sin(2\pi \mathbf{s}_1^\top \mathbf{x}), \cdots, \cos(2\pi \mathbf{s}_{H'}^\top \mathbf{x}), \sin(2\pi \mathbf{s}_{H'}^\top \mathbf{x}))^\top, \quad (3)$$

where $H' = H/2$ and $\mathbf{s}_h$ is a D-dimensional vector sampled from the kernel's spectral density, e.g., $\mathcal{N}(\mathbf{s}; \mathbf{0}_D, (4\pi^2 \boldsymbol{\Lambda})^{-1})$ for the (ARD) Gaussian kernel defined by $k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 \exp\big(\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \boldsymbol{\Lambda}(\mathbf{x} - \mathbf{x}')\big)$. This, together with the weight-space-view formulation of GPR, enables us to create a sparse-spectrum approximation of GPR through Bayesian linear regression (BLR) on feature space $\mathcal{F}_\psi \subseteq \mathbb{R}^H$. This reduces the memory requirement to $O(NH)$ since we no longer store the $N \times N$ covariance matrix $\underline{\mathbf{C}}_\theta$. As a result, the predictive distribution and the logarithm of marginal distribution for BLR, denoted by $p_{\mathrm{BLR}}(y_* | \mathbf{x}_*, \mathcal{D}_{\mathrm{CD}})$ and $\mathcal{L}_{\mathrm{BLR}}$, can be computed with lower memory usage and computation cost. See Appendix B for thorough definitions and detailed explanations. Although any positive definite kernel can be adopted in our method, we assume that kernel k is shift-invariant when deriving the approximate variant of GPSR using RFF.

4 Gaussian Process Shuffled Regression (GPSR)

This section presents our probabilistic model based on GPs for shuffled regression (GPSR).

4.1 Generative Process and Marginal Likelihood of GPSR

GPSR is constructed by the following generative process of shuffled data $\mathcal{D}_{\mathrm{SD}} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^M$. Similar to GPR [13] in the previous section, we put GPs prior on f, $f \sim \mathcal{GP}(0, k_\theta(\mathbf{x}, \mathbf{x}'))$, and denote function values at i-th input-set $\mathbf{X}_i = \{\mathbf{x}_{i1}, \cdots, \mathbf{x}_{iL}\}$ as $\mathbf{f}_i = (f(\mathbf{x}_{i1}), \cdots, f(\mathbf{x}_{iL}))^\top$. So $N\ (=ML)$-dimensional vector $\mathbf{f} = (\mathbf{f}_1^\top, \cdots, \mathbf{f}_M^\top)^\top$ is subject to the multivariate Gaussian given by

$$p(\mathbf{f} | \mathbf{X}) = \mathcal{N}(\mathbf{f}; \mathbf{0}_N, \mathbf{K}), \quad (4)$$

where $\mathbf{K}$ is the $N \times N$ gram matrix whose $((i-1)L+\ell, (i'-1)L+\ell')$-th-th element is $k_\theta(\mathbf{x}_{i\ell}, \mathbf{x}_{i'\ell'})$. For each sample $i = 1 \cdots M$, using $L \times L$ latent permutation matrix[3] $\boldsymbol{\Pi}_i$, response-set $\mathbf{y}_i = (y_{i1}, \cdots, y_{iL})^\top$ is determined by shuffling latent function values $\mathbf{f}_i$ and adding Gaussian noise, i.e., $\mathbf{y}_i$ follows $p(\mathbf{y}_i | \mathbf{f}_i) = \mathcal{N}(\mathbf{y}_i; \boldsymbol{\Pi}_i \mathbf{f}_i, \beta^{-1} \mathbf{I}_L)$. This yields the joint probability distribution of all responses $\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_M^\top)^\top$ given $\mathbf{f}$ as

$$p(\mathbf{y} | \mathbf{f}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\Pi} \mathbf{f}, \beta^{-1} \mathbf{I}_N), \quad \boldsymbol{\Pi} = \mathrm{diag}(\boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2, \cdots, \boldsymbol{\Pi}_M). \quad (5)$$

From Eq. (4) and (5), we get the marginal likelihood for GPSR by integrating over function value $\mathbf{f}$,

$$p(\mathbf{y} | \mathbf{X}) = \int p(\mathbf{y} | \mathbf{f}) p(\mathbf{f} | \mathbf{X}) d\mathbf{f} = \mathcal{N}(\mathbf{y}; \mathbf{0}_N, \mathbf{S}),\ \mathbf{S} = \boldsymbol{\Pi} \mathbf{K} \boldsymbol{\Pi}^\top + \beta^{-1} \mathbf{I}_N = \boldsymbol{\Pi} \underbrace{(\mathbf{K} + \beta^{-1} \mathbf{I}_N)}_{\mathbf{C}_\theta} \boldsymbol{\Pi}^\top. \quad (6)$$

[3] A permutation matrix is a square binary matrix that contains exactly one 1 in each row and column, with all other elements being 0.

Figure 1b shows the graphical model representation of GPSR. Note that the order of generating function values $\mathbf{f}$ and shuffling by $\mathbf{\Pi}$ can be exchanged; the identical marginal likelihood is derived even if we first shuffle inputs $\mathbf{X}$ and then generate the function values of shuffled inputs.

For further analysis of the marginal likelihood, we denote the logarithm of (6) as symbol $\mathcal{L}_{\mathrm{GPSR}}$,

$$(\text{GPSR Marginal L.})\ \mathcal{L}_{\mathrm{GPSR}}(\theta;\mathcal{D}_{\mathrm{SD}}) \triangleq \log p(\mathbf{y}|\mathbf{X}) = \underbrace{-\frac{1}{2}\log|\mathbf{S}|}_{\text{complexity}} \underbrace{-\frac{1}{2}\mathbf{y}^\top\mathbf{S}^{-1}\mathbf{y}}_{\text{data fit}} - \frac{N}{2}\log(2\pi), \quad (7)$$

and define *pseudo coupled data* (PCD) as follows:

Definition 4.1. (Pseudo Coupled Data) *Given shuffled data $\mathcal{D}_{\mathrm{SD}} = \{(\{\mathbf{x}_{i\ell}\}_{\ell=1}^L, \{y_{i\ell}\}_{\ell=1}^L)\}_{i=1}^M$ and permutation matrices $\{\mathbf{\Pi}_i\}_{i=1}^M$, we define the inversely permuted response $\tilde{\mathbf{y}}^\top = (\tilde{\mathbf{y}}_1^\top, \cdots, \tilde{\mathbf{y}}_M^\top)$ as*

$$(\text{Inversely Permuted Response}) \quad \tilde{\mathbf{y}}_i \triangleq \mathbf{\Pi}_i^\top \mathbf{y}_i,\ (\text{i.e., } \tilde{\mathbf{y}} = \mathbf{\Pi}^\top\mathbf{y}). \quad (8)$$

We also denote the ℓ-th element of $\tilde{\mathbf{y}}_i$ as $\tilde{y}_{i\ell}$. Pseudo coupled data $\mathcal{D}_{\mathrm{PCD}}$ is defined as a set of $N{=}ML$ pairs of input $\mathbf{x}_{i\ell}$ with inversely permuted response $\tilde{y}_{i\ell}$, $\mathcal{D}_{\mathrm{PCD}} = \{(\mathbf{x}_{i\ell}, \tilde{y}_{i\ell})\}_{i,\ell=1}^{M,L}$.

Marginal likelihood of GPSR is equivalent to that of GPR (1) with pseudo coupled data.

Proposition 4.2. (Equivalence of Marginal Likelihood) *$\mathcal{L}_{\mathrm{GPSR}}(\theta;\mathcal{D}_{\mathrm{SD}}) = \mathcal{L}_{\mathrm{GPR}}(\theta;\mathcal{D}_{\mathrm{PCD}})$ holds.*

Proof. By the multiplicativity of determinant and the properties of permutation matrix ($\mathbf{\Pi}^\top = \mathbf{\Pi}^{-1}$ and $|\mathbf{\Pi}||\mathbf{\Pi}^\top| = 1$), we get

$$\begin{aligned}\mathcal{L}_{\mathrm{GPSR}}(\theta;\mathcal{D}_{\mathrm{SD}}) &= -\frac{1}{2}\log(|\mathbf{\Pi}||\mathbf{C}_\theta||\mathbf{\Pi}^\top|) - \frac{1}{2}\mathbf{y}^\top(\mathbf{\Pi}\mathbf{C}_\theta^{-1}\mathbf{\Pi}^\top)\mathbf{y} - \frac{N}{2}\log(2\pi) \quad (9)\\ &= -\frac{1}{2}\log|\mathbf{C}_\theta| - \frac{1}{2}\underbrace{(\mathbf{\Pi}^\top\mathbf{y})^\top\mathbf{C}_\theta^{-1}(\mathbf{\Pi}^\top\mathbf{y})}_{\tilde{\mathbf{y}}^\top\mathbf{C}_\theta^{-1}\tilde{\mathbf{y}}} - \frac{N}{2}\log(2\pi) = \mathcal{L}_{\mathrm{GPR}}(\theta;\mathcal{D}_{\mathrm{PCD}}).\ \square\end{aligned}$$

This result allows us to use GPR as a subroutine of the inference detailed in the next subsection.

4.2 Inference

Similar to GPR [13], we adopt the Bayesian evidence maximization framework [21] for estimating kernel hyperparameter θ and permutation matrix $\mathbf{\Pi}$. To effectively utilize the Proposition 4.2, we adopt the alternating optimization scheme that iteratively updates θ and $\mathbf{\Pi}$.

Optimization of θ. Proposition 4.2 allows us to employ exactly same approach as GPR when estimating hyperparameter θ. We update θ by applying an optimization routine such as scaled conjugate gradient (SCG) and L-BFGS using the following partial derivatives of $\mathcal{L}_{\mathrm{GPSR}}$ w.r.t. the hyperparameters,

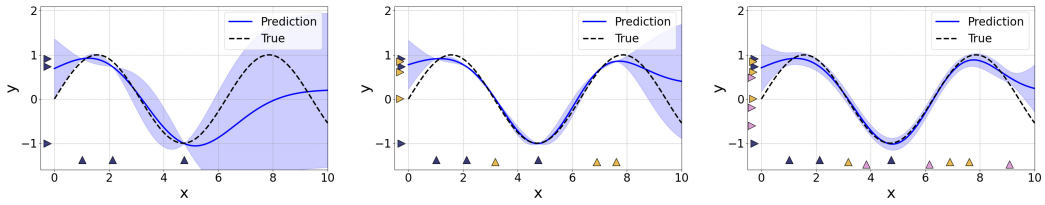
$$\frac{\partial\mathcal{L}_{\mathrm{GPSR}}(\theta;\mathcal{D}_{\mathrm{SD}})}{\partial\theta} = \frac{\partial\mathcal{L}_{\mathrm{GPR}}(\theta;\mathcal{D}_{\mathrm{PCD}})}{\partial\theta} = -\frac{1}{2}\mathrm{tr}\Big(\mathbf{C}_\theta^{-1}\frac{\partial\mathbf{C}_\theta}{\partial\theta}\Big) - \frac{1}{2}(\mathbf{C}_\theta^{-1}\tilde{\mathbf{y}})^\top\frac{\partial\mathbf{C}_\theta}{\partial\theta}(\mathbf{C}_\theta^{-1}\tilde{\mathbf{y}}). \quad (10)$$

Optimization of $\mathbf{\Pi}$. Considering that permutation matrix $\mathbf{\Pi}$ is a discrete variable, we cast this as a known combinatorial optimization problem. By extracting the terms related to $\mathbf{\Pi}$ from $\mathcal{L}_{\mathrm{GPSR}}$ (Eq. (9)) with multiplication by -2 for simplicity, we get

$$(\mathbf{\Pi}^\top\mathbf{y})^\top\mathbf{C}_\theta^{-1}\mathbf{\Pi}^\top\mathbf{y} = \mathrm{tr}\{\mathbf{C}_\theta^{-1}\mathbf{\Pi}^\top\mathbf{y}(\mathbf{\Pi}^\top\mathbf{y})^\top\} = \mathrm{tr}(\mathbf{C}_\theta^{-1}\mathbf{\Pi}^\top\mathbf{y}\mathbf{y}^\top\mathbf{\Pi}) \triangleq \mathcal{U}_{\mathrm{QAP}}(\mathbf{\Pi}), \quad (11)$$

where we used the trace trick ($\mathbf{z}^\top\mathbf{A}\mathbf{z} = \mathrm{tr}(\mathbf{A}\mathbf{z}\mathbf{z}^\top)$). We can regard $\mathcal{U}_{\mathrm{QAP}}$ as the objective function of the quadratic assignment problem (QAP), and so we can employ one of the existing QAP solvers [36] [4]. In the later experiments, we use *simulated annealing*, which is easy to implement and closely related to Markov Chain Monte Carlo (MCMC) used in existing shuffled regression methods [4, 20]. For more details on the inference procedure, see Appendix C. Note that QAP is a NP-hard problem and this procedure seems prohibitive when handling medium/large scale data; we derive an approximation algorithm for GPSR that can avoid solving QAP in §5.

[4] Here $\mathbf{\Pi}$ has block diagonal structure and minor modification of constraint conditions may be needed.



(a) $M = 1$ (b) $M = 2$ (c) $M = 3$

Figure 2: Predictive mean and variance of GPSR for various sample sizes, M=1 to 3, with set size L=3. Marker $\triangle$ indicates the elements in the input-set and response-set in shuffled data. Same color means same index i. GPSR captures the true function ($y = \sin x$) as the sample size is increased.

4.3 Making Predictions

We derive the predictive distribution of y_* for input $\mathbf{x}_*$ in GPSR by considering the joint distribution of the $(N+1)$-dimensional vector $\mathbf{y}' = (\mathbf{y}^\top, y_*)^\top$. Note that $(\mathbf{x}_*, y_*)$ is a "coupled" pair not included in $\mathcal{D}_{\text{SD}}$ as shown in Fig. 1b. We define the latent function value at $\mathbf{x}_*$ as $\mathrm{f}_* = f(\mathbf{x}_*)$ and $\mathbf{f}' = (\mathbf{f}^\top, \mathrm{f}_*)^\top$. From GPs prior on f, $p(\mathbf{f}'|\mathbf{X}') = \mathcal{N}(\mathbf{f}'; \mathbf{0}_{(N+1)}, \mathbf{K}')$ where $\mathbf{K}'$ is $(N+1) \times (N+1)$ gram matrix computed using $\mathbf{X}' = \mathbf{X} \cup \mathbf{x}_*$. From Eq. (5), and assuming $p(\mathrm{y}_*|\mathrm{f}_*) = \mathcal{N}(\mathrm{y}_*|\mathrm{f}_*, \beta^{-1})$ similar to GPR, $\mathbf{y}'$ is determined by following $p(\mathbf{y}'|\mathbf{f}') = \mathcal{N}(\mathbf{y}'; \mathbf{\Pi}'\mathbf{f}', \beta^{-1}\mathbf{I}_{(N+1)})$ where $\mathbf{\Pi}' = \mathrm{diag}(\mathbf{\Pi}, 1)$. Then, the marginal distribution of $\mathbf{y}'$ is given by

$$p(\mathbf{y}'|\mathbf{X}') = \int p(\mathbf{y}'|\mathbf{f}')p(\mathbf{f}'|\mathbf{X}')d\mathbf{f}' = \mathcal{N}(\mathbf{y}'; \mathbf{0}_{(N+1)}, \mathbf{S}'), \text{ where } \mathbf{S}' = \mathbf{\Pi}'\mathbf{K}'\mathbf{\Pi}'^\top + \beta^{-1}\mathbf{I}_{(N+1)}. \quad (12)$$

Note that $\mathbf{S}'$ can be cast in the following block form:

$$\mathbf{S}' = \begin{pmatrix} \mathbf{\Pi} & \mathbf{0}_N \\ \mathbf{0}_N^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{C}_\theta & \mathbf{k} \\ \mathbf{k}^\top & k_\theta(\mathbf{x}_*, \mathbf{x}_*) + \beta^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{\Pi}^\top & \mathbf{0}_N \\ \mathbf{0}_N^\top & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{S} & \mathbf{\Pi}\mathbf{k} \\ (\mathbf{\Pi}\mathbf{k})^\top & k_\theta(\mathbf{x}_*, \mathbf{x}_*) + \beta^{-1} \end{pmatrix}, \quad (13)$$

where $\mathbf{k}$ is the N-dimensional vector whose $\{(i-1)L + \ell\}$-th element is $k_\theta(\mathbf{x}_{i\ell}, \mathbf{x}_*)$. The predictive distribution of y_* given shuffled data is the conditional Gaussian of (12); it is given by

$$\begin{aligned} &\text{(GPSR Prediction)} \quad p_{\text{GPSR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\text{SD}}) = \mathcal{N}(y_*; m_{\text{GPSR}}(\mathbf{x}_*), \sigma^2_{\text{GPSR}}(\mathbf{x}_*)), \quad (14) \\ &m_{\text{GPSR}}(\mathbf{x}_*) = (\mathbf{\Pi}\mathbf{k})^\top \mathbf{S}^{-1}\mathbf{y} = \mathbf{k}^\top \mathbf{\Pi}^\top (\mathbf{\Pi}\mathbf{C}_\theta^{-1}\mathbf{\Pi}^\top)\mathbf{y} = \mathbf{k}^\top \mathbf{C}_\theta^{-1}(\mathbf{\Pi}^\top \mathbf{y}) = \mathbf{k}^\top \mathbf{C}_\theta^{-1}\tilde{\mathbf{y}}, \\ &\sigma^2_{\text{GPSR}}(\mathbf{x}_*) = k_\theta(\boldsymbol{x}_*, \boldsymbol{x}_*) + \beta^{-1} - (\mathbf{\Pi}\mathbf{k})^\top \mathbf{S}^{-1}\mathbf{\Pi}\mathbf{k} = k_\theta(\mathbf{x}_*, \mathbf{x}_*) + \beta^{-1} - \mathbf{k}^\top \mathbf{C}_\theta^{-1}\mathbf{k}. \end{aligned}$$

Note that $\mathbf{\Pi}^\top = \mathbf{\Pi}^{-1}$ is used in the above derivation. It is obvious, similar to Proposition 4.2, that the predictive distribution of GPSR is equivalent to that of GPR (2) given pseudo-coupled data.

Proposition 4.3. (Equivalence of Prediction) $p_{\text{GPSR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\text{SD}}) = p_{\text{GPR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\text{PCD}})$ *holds.*

Proof. The proof is completed by replacing $\underline{\mathbf{y}}$ in Eq. (2) by response vector of PCD $\tilde{\mathbf{y}}$. □

Figure 2 demonstrates the effectiveness of GPSR with the derived inference algorithm and predictive distribution for the simple toy problem. GPSR can output a prediction with its uncertainty and capture the true function as the sample size M is increased. However, the prediction and inference procedure presented in this section is prohibitive for handling medium/large scale data since it has two difficulties: (i) need to compute the inversion of the covariance matrix $\mathbf{C}_\theta$ at the cost of $O(N^3)$ operations (same as GPR), and (ii) need to solve QAP for estimating permutation matrix $\mathbf{\Pi}$. The next section provides a sparse variant of GPSR that can resolve these two difficulties at once.

5 Sparse Spectrum GPSR (SS-GPSR): Approximation of GPSR using RFF.

The key idea behind our sparse variant of GPSR, sparse spectrum GPSR (SS-GPSR), is to use a weight-space-view of GPSR and random Fourier features (RFF) [29, 30]. Here we start with a model with a finite-dimensional feature map $\phi : \mathcal{X} \to \mathcal{F}_\phi$ whose inner product corresponds to kernel function k. We then consider using RFF defined in (3) to approximate GPSR with a shift-invariant kernel that may not have a corresponding finite-dimensional feature map.

5.1 Weight Space View of GPSR: Bayesian Shuffled Linear Regression (BSLR)

We derive the predictive distribution and inference algorithm for SS-GPSR from the following Bayesian shuffled linear regression (BSLR) models defined as

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \alpha^{-1}\mathbf{I}_H), \quad p(\mathbf{y}_i|\mathbf{X}_i, \mathbf{w}) = \mathcal{N}(\mathbf{y}_i|\mathbf{\Pi}_i\mathbf{\Phi}_i\mathbf{w}, \beta^{-1}\mathbf{I}_L), \tag{15}$$

where $\mathbf{\Phi}_i$ is an $L \times H$ design matrix whose ℓ-th row is $\boldsymbol{\phi}(\mathbf{x}_{i\ell})^\top$. We can see that BSLR is a weight-space-view of GPSR as follows. Using the equivalent expression of (15) $\mathbf{y}_i = \mathbf{\Pi}_i\mathbf{\Phi}_i\mathbf{w} + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{\epsilon}_i|\mathbf{0}_L, \beta^{-1}\mathbf{I}_L)$, we have the following mean and covariance of $\mathbf{y}_i$ and $\mathbf{y}_j$:

$$\begin{aligned} \mathbb{E}_{\mathbf{w},\boldsymbol{\epsilon}}[\mathbf{y}_i] &= \mathbf{\Pi}_i\mathbf{\Phi}_i\mathbb{E}_{\mathbf{w}}[\mathbf{w}] + \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{\epsilon}_i] = \mathbf{0}_L, \\ \mathbb{E}_{\mathbf{w},\boldsymbol{\epsilon}}[\mathbf{y}_i\mathbf{y}_j^\top] &= \mathbf{\Pi}_i\mathbf{\Phi}_i\mathbb{E}_{\mathbf{w}}[\mathbf{w}\mathbf{w}^\top]\mathbf{\Phi}_j^\top\mathbf{\Pi}_j^\top + \mathbb{E}_{\epsilon}[\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_j^\top] = \mathbf{\Pi}_i(\alpha^{-1}\mathbf{\Phi}_i\mathbf{\Phi}_j^\top + \delta_{ij}\beta^{-1}\mathbf{I}_L)\mathbf{\Pi}_j^\top. \end{aligned} \tag{16}$$

So the joint distribution of response-set $\mathbf{y}_1, \cdots, \mathbf{y}_M$ is given by zero-mean Gaussian with $N \times N$ covariance $\mathbf{R} = \mathbf{\Pi}(\alpha^{-1}\mathbf{\Phi}\mathbf{\Phi}^\top + \beta^{-1}\mathbf{I}_N)\mathbf{\Pi}^\top$; $\mathbf{\Phi}$ is $N \times H$ design matrix $\mathbf{\Phi} = (\mathbf{\Phi}_1^\top, \cdots, \mathbf{\Phi}_M^\top)^\top$. Thus we can confirm that BSLR is a weight-space-view of GPSR since $\mathbf{R}$ is equivalent to the covariance $\mathbf{S}$ in GPSR (6) if the kernel is defined by inner product of the feature map, $k(\mathbf{x}, \mathbf{x}') = \alpha^{-1}\phi(\mathbf{x})^\top\phi(\mathbf{x})$.

5.2 Prediction and Bayesian Evidence Maximization Framework for BSLR

We derive the predictive distribution and inference algorithm in BSLR. From Bayes rule, $p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}})$ $\propto p(\mathbf{w}, \mathbf{y}|\mathbf{X}) = p(\mathbf{w})\prod_{i=1}^M p(\mathbf{y}_i|\mathbf{X}_i, \mathbf{w})$ holds and thus a posterior distribution of $\mathbf{w}$ is given by

$$p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}}) = \mathcal{N}(\mathbf{w}; \bar{\mathbf{w}}, \mathbf{A}^{-1}),\ \bar{\mathbf{w}} = \beta\mathbf{A}^{-1}\mathbf{\Phi}^\top(\mathbf{\Pi}^\top\mathbf{y}) = \beta\mathbf{A}^{-1}\mathbf{\Phi}^\top\tilde{\mathbf{y}},\ \mathbf{A} = \alpha\mathbf{I}_H + \beta\mathbf{\Phi}^\top\mathbf{\Phi}. \tag{17}$$

So we obtain the following predictive distribution $p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) = \int p(y_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}})d\mathbf{w}$ and marginal likelihood $\mathcal{L}_{\mathrm{BSLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{SD}}) = \log\int p(\mathbf{y}, \mathbf{w}|\mathbf{X})d\mathbf{w}$ (see Appendix D for derivation):

$$\begin{aligned} &\text{(BSLR Prediction)} \quad p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) = \mathcal{N}(y_*; m_{\mathrm{BSLR}}(\mathbf{x}_*), \sigma^2_{\mathrm{BSLR}}(\mathbf{x}_*)), \\ &\quad m_{\mathrm{BSLR}}(\mathbf{x}_*) = \bar{\mathbf{w}}^\top\phi(\mathbf{x}_*), \quad \sigma^2_{\mathrm{BSLR}}(\mathbf{x}_*) = \beta^{-1} + \phi(\mathbf{x}_*)^\top\mathbf{A}^{-1}\phi(\mathbf{x}_*). \end{aligned} \tag{18}$$

$$\begin{aligned} &\text{(BSLR Marginal L.)} \ \mathcal{L}_{\mathrm{BSLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{SD}}) = \frac{H}{2}\log\alpha + \frac{N}{2}\log\Big(\frac{\beta}{2\pi}\Big) - \frac{1}{2}\log|\mathbf{A}| - \frac{1}{2}E(\bar{\mathbf{w}}), \\ &\quad E(\bar{\mathbf{w}}) = \alpha\bar{\mathbf{w}}^\top\bar{\mathbf{w}} + \beta||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 = \beta\mathbf{y}^\top\mathbf{y} + \bar{\mathbf{w}}^\top\mathbf{A}\bar{\mathbf{w}} - 2\beta\sum\nolimits_{i=1}^M \underbrace{\mathrm{tr}(\mathbf{\Pi}_i\mathbf{\Phi}_i\bar{\mathbf{w}}\mathbf{y}_i^\top)}_{\mathcal{U}_{\mathrm{LAP}}(\mathbf{\Pi}_i)}. \end{aligned} \tag{19}$$

Similar to GPSR, the predictive distribution and marginal likelihood of BSLR are equivalent to those of BLR given pseudo-coupled data (proofs are given in Appendix E).

Proposition 5.1. (Equivalence of Prediction) $p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) = p_{\mathrm{BLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{PCD}})$ *holds.*

Proposition 5.2. (Equivalence of Marginal Likelihood) $\mathcal{L}_{\mathrm{BSLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{SD}}) = \mathcal{L}_{\mathrm{BLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{PCD}})$ *holds.*

We can also show the predictive distribution of BSLR (18) is equivalent to that of GPSR (14) using the connection between GPSR and GPR, GPR and BLR (Appendix B), BLR and BSLR.

Proposition 5.3. (Equivalence of Prediction) $p_{\mathrm{GPSR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) = p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}})$ *holds if the kernel function k is given by $k(\mathbf{x}, \mathbf{x}') = \alpha^{-1}\phi(\mathbf{x})^\top\phi(\mathbf{x})$.*

Proof. Applying Proposition 4.3, B.1 and 5.1 sequentially, we get $p_{\mathrm{GPSR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) = p_{\mathrm{GPR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{PCD}}) = p_{\mathrm{BLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{PCD}}) = p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}})$. □

Proposition 5.3 states the predictive distribution of GPSR can be obtained via BSLR for kernels that admit finite-dimensional feature maps (regardless of whether the kernel is shift-invariant). For shift-invariant kernels that do not admit finite-dimensional feature maps, an approximate predictive distribution can be constructed using RFF ψ (3) as feature map ϕ. Therefore, we can more efficiently construct the (approximate) predictive distribution using BSLR when the number of samples N exceeds the dimension size H since the expression (18) involves the inverse of $H \times H$ matrix $\mathbf{A}$ instead of $N \times N$ gram matrix $\mathbf{C}_\theta$.

5.3 Inference

By leveraging the result of Proposition 5.2, a simple inference algorithm that iteratively updates the posterior of $\mathbf{w}$ and $\mathbf{\Pi}$ can be derived, similar to GPSR. Thanks to the equivalence of marginal likelihood, we can estimate posterior of $\mathbf{w}$ by exactly same way of BLR (update $\bar{\mathbf{w}}$ and $\mathbf{A}^{-1}$ by (17)). Also we can update $\mathbf{\Pi}$ by minimizing the trace term $\mathcal{U}_{\mathrm{LAP}}$ in (19) that can be seen as the objective of linear assignment problem (LAP). This is a simple instance of LAP, so the *optimal* permutation matrix $\mathbf{\Pi}_i$ can be found by a sort operation in $O(L \log L)$ without using tailored algorithms for LAP [37, 38, 39] [5]. This is because the objective function $\mathcal{U}_{\mathrm{LAP}}(\mathbf{\Pi}) = (\mathbf{\Pi}_i \mathbf{v}_i)^\top \mathbf{y}_i$ is maximized by setting the permutation matrix such that the ℓ-th largest element of response vector $\mathbf{y}_i$ and that of the predictive mean vector $\mathbf{v}_i \triangleq \mathbf{\Phi}_i \bar{\mathbf{w}}$ match each other for all ℓ. For more details on the inference procedure, see Appendix F.

Thus our SS-GPSR using the predictive distribution (18) and the above inference procedure reduces memory and computation costs associated with GPs and avoids solving QAP. Although existing studies also use sort operations [4, 12], to the best of our knowledge, this is the first to show that the permutation matrix can be estimated by sorting, even in the evidence maximization framework.

6 Experiments

6.1 Setting

Data. We evaluated GPSR and SS-GPSR using four publicly available data sets found in the UCI machine learning repository[6]: airfoil data (Airfoil), concrete compressive strength data (Concrete), Boston housing data (Housing), auto-MPG data (MPG). We also made Housing-1D with input feature dimension of $D = 1$ by extracting RM (average number of rooms) features for visualization purposes. We prepared 5 data sets by randomly dividing the data and using 60% for training, 20% for validation, and 20% for testing. We made the shuffled data used for training/validation by randomly dividing the training/validation data into M sets each with L elements (i.e., $M \approx 0.6 \times \mathrm{original\ datasize}/L$) and shuffling the indices in each set. Note that validation data were used only for neural network-based baselines as explained in next paragraph. The test data in the evaluations were coupled data. We used test mean squared error (test MSE) as the performance metric. We ran 5 trials using 5 sets of training, validation and test data.

Baselines. We compared (SS-)GPSR with linear-model-based method (SLR) [4] and the state-of-the-art method that uses neural networks (SDR) [12]. Both methods use a (Gaussian-based) MSE loss and were trained by the (stochastic) sparse EM algorithm. SDR adopted early-stopping with validation data (which were also shuffled data) [42]. As oracle baselines, we also examined linear-regression (LR), GPR, and deep-regression (DR) using coupled data with data size of $N = LM$. Both SDR and DR used a one-hidden-layer feedforward network with ReLU activation function.

Hyperparameters. GPSR (and Oracle GPR) used Gaussian kernels as in §3, and SS-GPSR used its RFF approximation with dimension size $H = 100$ in common for all datasets. The other hyperparameters such as precision α and β in (SS-)GPSR are estimated when applying GPR or BLR in inference procedure (see Alg. 1 and 2 in Appendix). So GPSR and SS-GPSR do not use validation data unlike SDR. Moreover, due to their slow convergence, both GPSR and SDR used warm initialization similar to [12], i.e., the initial value of permutation matrix is set using the estimated result of SLR. For SS-GPSR, ten different runs were performed with various initial permutation matrices, and the solution that maximized the marginal likelihood was selected. For more details of implementations and hyperparameters such as optimization setting for SDR, see Appendix G.

6.2 Results

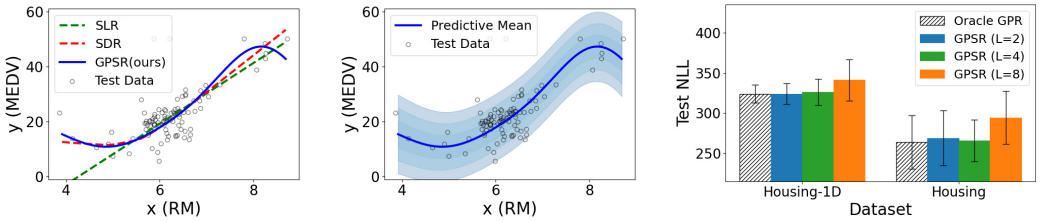
Table 3 shows the results of the experiments. We observe that SDR, GPSR and SS-GPSR which can express nonlinear functions, outperform SLR in almost all settings. When comparing SDR and

[5] This problem also can be seen as an optimal transport problem between two 1-D point clouds consisting of the same number of elements having unit mass [40, 41].

[6] `https://archive.ics.uci.edu/ml/index.php`

Table 3: Results on benchmark datasets with various set sizes L. Average and standard deviation of test MSE are shown. Bold/underline means the 1st/2nd best (lowest) MSE among the methods for shuffled regression. Note that the performance of oracle methods including linear regression (LR), deep regression (DR) and GPR trained using coupled data are also displayed in the Dataset column.

Dataset		Existing Methods		Proposed Methods	
Source	L	SLR	SDR	GPSR	SS-GPSR
Airfoil	2	23.66 ± 1.39	19.57 ± 1.73	**9.05 ± 0.99**	12.48 ± 1.36
LR: 22.33 ± 1.72	4	22.82 ± 1.62	20.15 ± 1.91	14.26 ± 1.46	**14.11 ± 2.07**
DR: 17.79 ± 2.49	8	24.05 ± 1.66	21.73 ± 2.02	19.44 ± 2.14	**17.21 ± 2.64**
GPR: 6.42 ± 0.47	16	25.18 ± 2.01	23.83 ± 1.99	**22.44 ± 2.28**	24.22 ± 1.78
Concrete	2	139.79 ± 4.13	101.62 ± 26.15	**45.05 ± 7.64**	52.20 ± 2.64
LR: 119.06 ± 7.00	4	119.66 ± 4.96	101.04 ± 27.47	67.22 ± 8.79	**58.57 ± 4.31**
DR: 82.66 ± 37.06	8	128.62 ± 4.62	121.67 ± 20.89	97.79 ± 5.54	**68.19 ± 3.84**
GPR: 39.21 ± 11.03	16	161.95 ± 30.28	165.32 ± 33.61	146.74 ± 34.55	**137.26 ± 20.75**
Housing-1D	2	51.40 ± 6.78	37.08 ± 14.47	**33.60 ± 8.29**	34.45 ± 8.37
LR: 41.09 ± 7.54	4	40.95 ± 7.69	35.19 ± 9.59	**33.64 ± 8.26**	34.85 ± 8.63
DR: 41.25 ± 23.52	8	41.53 ± 8.42	35.72 ± 8.74	**35.22 ± 8.45**	36.24 ± 8.68
GPR: 33.70 ± 8.01	16	43.07 ± 8.88	37.78 ± 8.61	**37.18 ± 9.19**	37.93 ± 8.51
Housing	2	33.68 ± 6.10	12.31 ± 4.71	**11.19 ± 4.51**	11.70 ± 4.95
LR: 23.86 ± 5.69	4	24.57 ± 5.11	11.73 ± 3.53	**11.44 ± 4.91**	12.70 ± 5.17
DR: 11.03 ± 3.38	8	27.74 ± 8.11	18.25 ± 7.42	18.84 ± 8.14	**17.11 ± 8.04**
GPR: 9.78 ± 4.22	16	45.03 ± 16.43	**39.11 ± 15.88**	41.25 ± 17.86	46.88 ± 23.72
MPG	2	12.31 ± 1.30	8.96 ± 1.55	8.21 ± 1.20	**7.60 ± 1.29**
LR: 11.92 ± 1.10	4	13.24 ± 1.59	10.54 ± 1.75	8.71 ± 1.83	**8.47 ± 1.42**
DR: 7.69 ± 1.00	8	15.15 ± 3.25	11.55 ± 3.75	10.94 ± 3.21	**9.05 ± 1.80**
GPR: 7.75 ± 0.93	16	21.50 ± 6.49	15.38 ± 4.23	18.43 ± 6.72	**10.65 ± 3.03**



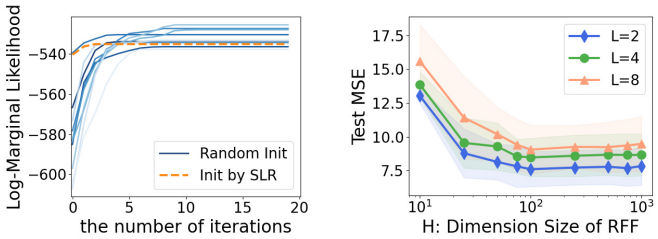
(a) Learned regression functions (b) Predictive uncertainty (c) Negative log-likelihood

Figure 3: (a) Learned regression function (predictive mean) and (b) predictive uncertainty of GSPR for Housing-1D ($L = 8$). The shaded regions represent ± 1, ± 2, and ± 3 standard deviations, respectively. (c) Comparison of negative log-likelihood on test data (Test NLL).

(SS-)GPSR, either of our proposals (GPSR or SS-GPSR) demonstrates the best performance across all datasets, except for Housing ($L = 16$), confirming our methods' effectiveness. Note that GPSR and SS-GPSR perform similarly well across a comparable number of datasets, so we cannot judge which is superior in terms of prediction performance. From Fig. 3a, we can see that GPSR well captures the non-linear structure thanks to its nonparametric flexibility, and this contributes to its superior performance comparable to that of oracle methods for Housing-1D. Figure 3b illustrates that GPSR effectively captures the inherent uncertainty in the data, while Fig. 3c reports the negative log-likelihood on the test data, demonstrating the method's effectiveness in uncertainty quantification.

We also investigated the detailed behavior of SS-GPSR as illustrated in Fig. 4. In Fig. 4a, we observe that SS-GPSR with the random initialization scheme can identify a better solution than warm initialization (in terms of marginal likelihood) and converges within about ten iterations. Given that SDR requires a longer time to converge without warm initialization [12] [7], these findings highlight

[7] We confirmed that GPSR has a similar convergence behavior to SDR.



(a) Convergence behavior (b) Effect of RFF

Figure 4: (a) Convergence behavior of SS-GPSR with different initialization schemes (10 different runs for random init.) for MPG ($L = 8$), and (b) effect of the dimension size of RFF on SS-GPSR for MPG ($L = 2{\sim}8$).

the effectiveness of SS-GPSR. Figure 4b further explores the impact of the dimension size of RFF, H. It is shown that although MSE performance is degraded when H is smaller than 100, it stabilizes when H exceeds 100. Thus, SS-GPSR utilizing RFF works well without requiring sensitive tuning of the dimension size. These results support the usefulness of SS-GPSR.

7 Conclusion

In this study, we proposed GPSR and SS-GPSR for learning regression functions from shuffled data. By theoretically analyzing the connection between GPSR, GPR and their weight-space formulations, we derived the easy-to-implement inference algorithms for (SS-)GPSR, and confirmed their effectiveness by experiments on public benchmark datasets.

Limitation and Future Work. The proposed methods are the first to apply GPs for shuffled regression, and a promising direction for future work is to extend their applicability to higher-dimensional and more complex settings. This may involve incorporating advanced techniques such as deep kernel learning [43], which combines neural networks to learn task-specific feature representations, as well as sparse GPs described in §2, which offer scalable solutions for large-scale problems. Additionally, exploring fully Bayesian approaches that introduce prior distributions over the kernel parameters and permutation matrix offers another promising avenue for future research, as similar techniques have been successfully applied in related domains [20, 33].

Societal Impact. Shuffled regression methods, including ours, are useful for analyzing e.g., independently collected data as mentioned in §1. However, it's important to note that coupled (test) data with input-output correspondence is necessary to evaluate the regression performance (That's why the experiments in this paper are based on benchmark data). When applying the method to real-world applications without coupled (test) data, it's crucial to carefully assess the performance of the estimated model when making predictions or decisions. Furthermore, the interpretation of input-output correspondences via latent variable representations requires careful consideration, as such usage falls outside our primary scope and may pose risks in scenarios involving privacy concerns.

Acknowledgments and Disclosure of Funding

We would like to thank the anonymous reviewers for their insightful comments and constructive feedback, which have helped improve the quality and clarity of this paper. This research was supported by the Human Informatics Laboratories at NTT, Inc. as part of the authors' regular employment responsibilities. The authors declare no competing interests related to this work.

References

[1] Alexandra Carpentier and Teresa Schlüter. Learning relationships between data obtained independently. In *Artificial Intelligence and Statistics*, pages 658–666, 2016.

[2] Daniel J Hsu, Kevin Shi, and Xiaorui Sun. Linear regression without correspondence. *Advances in Neural Information Processing Systems*, 30, 2017.

[3] Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *IEEE Transactions on Information Theory*, 64(5):3286–3300, 2017.

[4] Abubakar Abid and James Zou. A stochastic expectation-maximization approach to shuffled linear regression. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 470–477, 2018.

[5] Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59:259–284, 2004.

[6] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. Generalized shuffled linear regression. In *IEEE/CVF International Conference on Computer Vision*, pages 6474–6483, 2021.

[7] Xu Shi, Xiaoou Li, and Tianxi Cai. Spherical regression under mismatch corruption with application to automated knowledge translation. *Journal of the American Statistical Association*, 116(536):1953–1964, 2021.

[8] Martin Slawski, Mostafa Rahmani, and Ping Li. A sparse representation-based approach to linear regression with partially shuffled labels. In *Uncertainty in Artificial Intelligence*, pages 38–48, 2020.

[9] Guanhua Fang and Ping Li. Regression with label permutation in generalized linear model. In *International Conference on Machine Learning*, pages 9716–9760, 2023.

[10] Ikko Yamane, Yann Chevaleyre, Takashi Ishida, and Florian Yger. Mediated uncoupled learning and validation with bregman divergences: Loss family with maximal generality. In *Artificial Intelligence and Statistics*, pages 4768–4801, 2023.

[11] Martin Slawski and Bodhisattva Sen. Permuted and unlinked monotone regression in r^ d: an approach based on mixture modeling and optimal transport. *Journal of Machine Learning Research*, 25(183):1–57, 2024.

[12] Masahiro Kohjima. Shuffled deep regression. In *AAAI Conference on Artificial Intelligence*, pages 13238–13245, 2024.

[13] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[14] Manolis C Tsakiris, Liangzu Peng, Aldo Conca, Laurent Kneip, Yuanming Shi, and Hayoung Choi. An algebraic-geometric approach for linear regression without correspondences. *IEEE Transactions on Information Theory*, 66(8):5130–5144, 2020.

[15] Liyuan Xu, Junya Honda, Gang Niu, and Masashi Sugiyama. Uncoupled regression from pairwise comparison data. *Advances in Neural Information Processing Systems*, pages 3992–4002, 2019.

[16] Ikko Yamane, Junya Honda, Florian Yger, and Masashi Sugiyama. Mediated uncoupled learning: Learning functions without direct input-output correspondences. In *International Conference on Machine Learning*, pages 11637–11647, 2021.

[17] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing with random linear measurements. *IEEE Transactions on Information Theory*, 64(5):3237–3253, 2018.

[18] Masahiro Kohjima, Yuta Nambu, Yuki Kurauchi, and Ryuji Yamamoto. General algorithm for learning from grouped uncoupled data and pairwise comparison data. In *International Conference on Neural Information Processing*, pages 153–164, 2022.

[19] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[20] Zhenbang Wang, Emanuel Ben-David, and Martin Slawski. Regularization for shuffled data problems via exponential family priors on the permutation group. In *Artificial Intelligence and Statistics*, pages 2939–2959, 2023.

[21] David JC MacKay. Hyperparameters: optimize, or integrate out? In *Maximum Entropy and Bayesian Methods: Santa Barbara, California, USA, 1993*, pages 43–59. Springer, 1996.

[22] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[23] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18, 2005.

[24] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[25] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[26] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

[27] Andrew Gordon Wilson. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge Cambridge, UK, 2014.

[28] Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

[29] Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

[30] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

[31] Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized gp models. In *International Conference on Machine Learning*, pages 1302–1311, 2015.

[32] James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

[33] Vidhi Lalchand, Wessel Bruinsma, David Burt, and Carl Edward Rasmussen. Sparse gaussian process hyperparameters: optimize or integrate? *Advances in Neural Information Processing Systems*, pages 16612–16623, 2022.

[34] Grace Wahba. A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. *The annals of statistics*, pages 1378–1402, 1985.

[35] José M Bernardo and Adrian FM Smith. *Bayesian theory*. John Wiley & Sons, 2009.

[36] Eliane Maria Loiola, Nair Maria Maia De Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[37] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[38] Dimitri P Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

[39] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446, 2011.

[41] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[43] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[44] David JC MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions are clearly explained in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the assumptions and proofs in main text. Some propositions used in the proofs are explained in Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed explanation of the experimental setting are provided in §6 and Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not provide the code but provide implementation details of proposed methods and baselines in § 6 and Appendix G. Also, the experiment is conducted in open-access data (provided in UCI machine learning repository).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed explanation of the data and hyperaparameters are provided in §6 and Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide both mean and standard deviations in Table 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the Conclusion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide URLs of the used code and dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in our research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Symbol Definitions

Table 4 shows the symbols used in our paper. Bold symbols/vectors related to couple data are underlined.

Table 4: Notation summary for the paper

Symbol	Description
$\mathcal{X}$	input space $\mathcal{X} \subseteq \mathbb{R}^D$ where D is the dimension size of input space
$\mathcal{Y}$	response space $\mathcal{Y} \subseteq \mathbb{R}$
$\mathcal{F}_\nu$	feature space constructed by given feature map ν on $\mathcal{X}$, $\mathcal{F}_\nu \subseteq \mathbb{R}^H$
k_θ	kernel function with hyperparameter θ used for Gaussian process, $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
ψ	random Fourier features approximating (shift-invariant) kernel, $\psi : \mathcal{X} \to \mathcal{F}_\psi$
ϕ	feature map used for linear regression model, $\phi : \mathcal{X} \to \mathcal{F}_\phi$
$\mathbf{w}$	H-dimensional vector parameter (regression coefficients) of linear regression model
α	hyperparameter (precision) in a prior distribution on $\mathbf{w}$
β	precision parameter of observation noise (commonly used regardless of the model)
L	the size of input/response-set in shuffled data (which reduces to coupled data when L=1)
M	the number of pairs in shuffled data
N	the total number of inputs/responses $N = ML$
$\mathcal{D}_{\mathrm{SD}}$	shuffled data (pairs of input-set $\mathbf{X}_i$ and response-set $\boldsymbol{y}_i$), $\mathcal{D}_{\mathrm{SD}} = \{(\mathbf{X}_i, \boldsymbol{y}_i)\}_{i=1}^M$
$\mathbf{X}_i$	i-th input-set $\mathbf{X}_i = \{\mathbf{x}_{i1}, \cdots, \mathbf{x}_{iL}\}$ where $\mathbf{x}_{i\ell} \in \mathcal{X}$
$\boldsymbol{y}_i$	i-th response-set $\boldsymbol{y}_i = \{y_{i1}, \cdots, y_{iL}\}$ where $y_{i\ell} \in \mathcal{Y}$
$\mathbf{y}_i$	L-dimensional vector representation of $\boldsymbol{y}_i$, $\mathbf{y}_i = (y_{i1}, \cdots, y_{iL})^\top$
$\mathbf{X}$	all inputs in shuffled data $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^M$
$\mathbf{y}$	N-dimensional vector representation of all responses $\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_M^\top)^\top$
$\mathbf{f}_i$	L-dimensional vector rep. of function values of i-th input set, $\mathbf{f}_i = (\mathrm{f}_{i1}, \cdots, \mathrm{f}_{iL})^\top$
$\mathbf{f}$	N-dimensional vector representation of all function values, $\mathbf{f} = (\mathbf{f}_1^\top, \cdots, \mathbf{f}_M^\top)^\top$
$\boldsymbol{\Pi}_i$	$L \times L$ permutation matrix used for shuffling function values $\mathbf{f}_i$ (or $\tilde{\boldsymbol{\Phi}}_i \mathbf{w}$)
$\boldsymbol{\Pi}$	$N \times N$ permutation matrix defined by $\boldsymbol{\Pi} = \mathrm{diag}(\boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2, \cdots, \boldsymbol{\Pi}_M)$
$\mathbf{K}$	$N \times N$ gram matrix whose $((i-1)L+\ell, (i'-1)L+\ell')$-th element is $k_\theta(\mathbf{x}_{i\ell}, \mathbf{x}_{i'\ell'})$
$\mathbf{C}_\theta$	$N \times N$ covariance matrix $\mathbf{C}_\theta = \mathbf{K} + \beta^{-1}\mathbf{I}_N$ ($\mathbf{I}_N$ is the $N \times N$ identity matrix)
$\mathbf{S}$	$N \times N$ covariance matrix defined by $\mathbf{S} = \boldsymbol{\Pi}\mathbf{K}\boldsymbol{\Pi}^\top + \beta^{-1}\mathbf{I}_N = \boldsymbol{\Pi}\mathbf{C}_\theta\boldsymbol{\Pi}^\top$
$\boldsymbol{\Phi}_i$	$L \times H$ design matrix whose ℓ-th row is $\phi(\mathbf{x}_{i\ell})^\top$
$\boldsymbol{\Phi}$	$N \times H$ design matrix whose $\{(i-1)L+\ell\}$-th row is $\phi(\mathbf{x}_{i\ell})^\top$
$\mathcal{D}_{\mathrm{CD}}$	coupled data (pairs of input $\mathbf{x}_i \in \mathcal{X}$ and response $y_i \in \mathcal{Y}$), $\mathcal{D}_{\mathrm{CD}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$
$\underline{\mathbf{X}}$	all inputs in coupled data $\underline{\mathbf{X}} = \{\mathbf{x}_i\}_{i=1}^N$
$\underline{\mathbf{y}}$	N-dimensional vector representation of all responses, $\underline{\mathbf{y}} = (y_1, \cdots, y_N)^\top$
$\underline{\mathbf{f}}$	N-dimensional vector representation of all latent function values, $\underline{\mathbf{f}} = (\mathrm{f}_1, \cdots, \mathrm{f}_N)^\top$
$\underline{\mathbf{K}}$	$N \times N$ gram matrix whose (i,j)-th element is $k_\theta(\mathbf{x}_i, \mathbf{x}_j)$
$\underline{\mathbf{C}}_\theta$	$N \times N$ covariance matrix $\underline{\mathbf{C}}_\theta = \underline{\mathbf{K}} + \beta^{-1}\mathbf{I}_N$ ($\mathbf{I}_N$ is the $N \times N$ identity matrix)
$\underline{\boldsymbol{\Phi}}$	$N \times H$ design matrix whose i-th row is $\phi(\mathbf{x}_i)^\top$

B Details of Bayesian Linear Regression for Coupled Data

Bayesian linear regression (BLR) with parameter $\mathbf{w} \in \mathbb{R}^H$ and feature map $\phi : \mathcal{X} \to \mathcal{F}_\phi \subseteq \mathbb{R}^H$ is derived by considering the following prior distribution and model,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \alpha^{-1}\mathbf{I}_H), \quad p(y_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(y_i; \mathbf{w}^\top\phi(\mathbf{x}_i), \beta^{-1}), \tag{20}$$

where α and β are hyperparameters. From Bayes theorem, $p(\mathbf{w}|\mathcal{D}_{\mathrm{CD}}) \propto p(\underline{\mathbf{y}}, \mathbf{w}|\underline{\mathbf{X}}) = p(\mathbf{w})\prod_{i=1}^N p(y_i|\mathbf{x}_i, \mathbf{w})$ holds and thus a posterior distribution of $\mathbf{w}$ given coupled data $\mathcal{D}_{\mathrm{CD}}$ is written as

$$\text{(BLR Posterior)} \quad p(\mathbf{w}|\mathcal{D}_{\mathrm{CD}}) = \mathcal{N}(\mathbf{w}; \bar{\underline{\mathbf{w}}}, \underline{\mathbf{A}}^{-1}), \ \bar{\underline{\mathbf{w}}} = \beta\underline{\mathbf{A}}^{-1}\underline{\boldsymbol{\Phi}}^\top\underline{\mathbf{y}}, \ \underline{\mathbf{A}} = \alpha\mathbf{I}_H + \beta\underline{\boldsymbol{\Phi}}^\top\underline{\boldsymbol{\Phi}}, \tag{21}$$

where $\underline{\mathbf{\Phi}}$ is the $N \times H$ design matrix whose i-th row is $\phi(\mathbf{x}_i)^\top$. We get the predictive distribution of y_* at test point $\mathbf{x}_*$ by integrating over parameter $\mathbf{w}$:

(BLR Prediction)

$$p_{\mathrm{BLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{CD}}) = \int p(y_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|\mathcal{D}_{\mathrm{CD}}) d\mathbf{w} = \mathcal{N}(y_*; m_{\mathrm{BLR}}(\mathbf{x}_*), \sigma^2_{\mathrm{BLR}}(\mathbf{x}_*)), \quad (22)$$

$$m_{\mathrm{BLR}}(\mathbf{x}_*) = \underline{\bar{\mathbf{w}}}^\top \phi(\mathbf{x}_*), \quad \sigma^2_{\mathrm{BLR}}(\mathbf{x}_*) = \beta^{-1} + \phi(\mathbf{x}_*)^\top \underline{\mathbf{A}}^{-1} \phi(\mathbf{x}_*).$$

Thus the memory required for Bayesian linear regression is $O(NH)$ to store design matrix $\underline{\mathbf{\Phi}}$. Also, the inverse of $H \times H$ matrix $\underline{\mathbf{A}}$ can be computed in $O(H^3)$ by direct computation.

Remark. Note that $\underline{\bar{\mathbf{w}}}$ in (21) is both the mean and the mode of the posterior distribution $p(\mathbf{w}|\mathcal{D}_{\mathrm{CD}})$, so $\underline{\bar{\mathbf{w}}}$ is equivalent to the penalized maximum likelihood/maximum a posteriori (MAP) estimator. In non-Bayesian/frequentist schemes, however, the predictive distribution is constructed by plugging the estimator into the model (without integration over a posterior distribution), i.e., $p(y_*|\mathbf{x}_*, \underline{\bar{\mathbf{w}}}) = \mathcal{N}(y_*; \underline{\bar{\mathbf{w}}}^\top \phi(\mathbf{x}_*), \beta^{-1})$, and so the predictive variance is different from (22).

It is known that the predictive distribution of BLR (22) and that of GPR (2) are equivalent when the kernel function k is defined by the dot product of feature map ϕ.

Proposition B.1. (Equivalence of Prediction, e.g., [22]) *$p_{\mathrm{GPR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{CD}}) = p_{\mathrm{BLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{CD}})$ holds if the kernel function k is given by $k(\mathbf{x}, \mathbf{x}') = \alpha^{-1}\phi(\mathbf{x})^\top \phi(\mathbf{x})$.*

We define the logarithm of marginal likelihood for BLR by symbol $\mathcal{L}_{\mathrm{BLR}}$ as follows:

$$(\text{BLR Marginal L.})\ \mathcal{L}_{\mathrm{BLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{CD}}) \triangleq \log p(\underline{\mathbf{y}}|\underline{\mathbf{X}}) = \int p(\underline{\mathbf{y}}, \mathbf{w}|\underline{\mathbf{X}}) d\mathbf{w}. \quad (23)$$

Note that $\mathcal{L}_{\mathrm{BLR}}$ is an analytically-tractable quantity computed as

$$\mathcal{L}_{\mathrm{BLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{CD}}) = \frac{H}{2}\log\alpha + \frac{N}{2}\log\frac{\beta}{2\pi} - \frac{1}{2}\log|\underline{\mathbf{A}}| - \frac{\alpha}{2}\underline{\bar{\mathbf{w}}}^\top \underline{\bar{\mathbf{w}}} - \frac{\beta}{2}||\underline{\mathbf{y}} - \underline{\mathbf{\Phi}}\underline{\bar{\mathbf{w}}}||^2. \quad (24)$$

C Inference Procedure of GPSR

We detail the inference procedure of GPSR explained in §4.2. As stated, simulated annealing (SA) is used for solving QAP since it is easy to implement and closely related to Markov Chain Monte Carlo (MCMC) used in existing shuffled regression methods [4, 20]. Although not explicitly mentioned in the main text, we can estimate both the precision parameter β and the kernel parameter θ in the GPR subroutine. The equivalence of marginal likelihoods between GPSR and GPR (shown in Prop. 4.2) enables the use of existing GPR as subroutines, making the algorithm easier to implement.

Algorithm 1 shows the pseudo code that implements the inference procedure of GPSR. At initialization (line 1-3), permutation matrix $\mathbf{\Pi}$ is initialized. Note that, as reported in [12], convergence can be accelerated by warm-initialization, i.e., the initial setting of $\mathbf{\Pi}$ is set by using the result of e.g., SLR [4]. Then, the response vector of pseudo-coupled data (Definition 4.1) $\mathcal{D}_{\mathrm{PCD}}$, $\tilde{\mathbf{y}} = \mathbf{\Pi}^\top \mathbf{y}$, is set. At each (algorithmic) time step t, GPR-step (line 5-6) and Correspondence Update (CU)-Step (line 7-14) are iteratively conducted. As the GPR-step, the parameters of θ are optimized by applying GPR using pseudo-coupled data $\mathcal{D}_{\mathrm{PCD}}$ represented by $\mathbf{X}$ and $\tilde{\mathbf{y}}$. Next, as the CU-step, permutation matrix $\mathbf{\Pi}$ is updated by randomly swapping its rows following SA. Let us denote the permutation matrices before and after random swapping as $\mathbf{\Pi}^{(\mathrm{old})}$ and $\mathbf{\Pi}^{(\mathrm{new})}$, respectively. SA accepts this swapping following probability q:

$$q = \min(1, \exp(-\Delta/T)), \quad \Delta = \mathcal{L}(\mathbf{\Pi}^{(\mathrm{new})}) - \mathcal{L}(\mathbf{\Pi}^{(\mathrm{old})}). \quad (25)$$

where T is the temperature parameter. From the definition of q, the random swapping is accepted with probability 1 when the solution is improved. Otherwise, SA accepts the swapping with probability $\exp(-\Delta/T)$. i.e., the solution is updated with a certain probability even when the solution becomes worse, which helps to escape from local optima solutions.

Note that the above algorithm can be extended or modified in various ways, such as using the sparse GP methods described in §2 [23, 24, 25, 26, 28, 33] for the GPR-step and using optimization methods other than SA shown in [36] for the CU-Step.

Algorithm 1 Inference Procedure of GPSR

Input: kernel function k (with hyperparameter θ, β), shuffled data $\mathcal{D}_{\mathrm{SD}}$, optimization parameter (e.g., # of iterations/steps T_{max}, S_{max}, initial temperature T_{init}, and cooling rate γ)
Output: hyperparameter θ, β, response vector of pseudo coupled data $\tilde{\mathbf{y}}$, permutation matrix $\mathbf{\Pi}$

/* Initialization */
Randomly initialize $\mathbf{\Pi}$ (Optional: $\mathbf{\Pi}$ could be initialized by solving SLR).
Set the response vector of pseudo coupled data (Definition 4.1) as $\tilde{\mathbf{y}} \leftarrow \mathbf{\Pi}^\top \mathbf{y}$.
for $t = 1$ to T_{max} **do**
  /* GPR-Step (Optimization of hyperparameter using partial derivative Eq. (10)) */
  Update θ and β by SCG/L-BFGS (This could be done by e.g., GPR.fit($\mathbf{X}, \tilde{\mathbf{y}}$) in scikit-learn)
  /* Correspondence Update-Step (Optimization of $\mathbf{\Pi}$) by simulated annealing*/
  Set temperature $T \leftarrow T_{init}$
  for $s = 1$ to S_{max} **do**
    Randomly select $i \in \{1, \cdots, N\}$ and $\ell, \ell' \in \{1, \cdots, L\}$ $(\ell \neq \ell')$.
    Swap $\{(i-1)L + \ell\}$-th row and $\{(i-1)L + \ell'\}$-th row of $\mathbf{\Pi}$ with probability q (Eq. (25))
    Update temperature $T \leftarrow \gamma T$.
  end for
  Update the response vector of pseudo coupled data, $\tilde{\mathbf{y}} \leftarrow \mathbf{\Pi}^\top \mathbf{y}$
end for

D Details of Bayesian Shuffled Linear Regression (BSLR)

Here we derive BSLR's predictive distribution (18) and marginal likelihood (19). We start with the parameter posterior distribution as follows:

Derivation of posterior distribution $p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}})$ (17)**.** From the BSLR model (15), the joint distribution $p(\mathbf{w}, \mathbf{y}|\mathbf{X})$ is written as

$$
\begin{aligned}
&p(\mathbf{w}, \mathbf{y}|\mathbf{X}) = p(\mathbf{w}) \textstyle\prod_{i=1}^{M} p(\mathbf{y}_i|\mathbf{X}_i, \mathbf{w}) \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \exp\Big(-\frac{\alpha}{2}\mathbf{w}^\top\mathbf{w}\Big)\Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big(-\frac{\beta}{2}(\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\mathbf{w})^\top(\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\mathbf{w})\Big) \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big(-\frac{\alpha}{2}\mathbf{w}^\top\mathbf{w} - \frac{\beta}{2}\mathbf{w}^\top\mathbf{\Phi}^\top \underbrace{\mathbf{\Pi}^\top\mathbf{\Pi}}_{\text{cancel out}} \mathbf{\Phi}\mathbf{w} + \beta\mathbf{w}^\top\mathbf{\Phi}^\top\mathbf{\Pi}^\top\mathbf{y} - \frac{\beta}{2}\mathbf{y}^\top\mathbf{y}\Big) \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{\mathbf{w}^\top\Big(\beta\mathbf{\Phi}^\top(\mathbf{\Pi}^\top\mathbf{y})\Big) - \frac{1}{2}\mathbf{w}^\top\Big(\underbrace{\alpha\mathbf{I}_H + \beta\mathbf{\Phi}^\top\mathbf{\Phi}}_{\mathbf{A}}\Big)\mathbf{w} - \frac{\beta}{2}\mathbf{y}^\top\mathbf{y}\Big\} \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}\Big(\mathbf{w}^\top\mathbf{A}\mathbf{w} - 2\mathbf{w}^\top\beta\mathbf{\Phi}^\top(\mathbf{\Pi}^\top\mathbf{y}) + \beta\mathbf{y}^\top\mathbf{y}\Big)\Big\}. \qquad (26)
\end{aligned}
$$

By the trick of adding $0 = \bar{\mathbf{w}}^\top\mathbf{A}^{-1}\bar{\mathbf{w}} - \bar{\mathbf{w}}^\top\mathbf{A}^{-1}\bar{\mathbf{w}}$ and $\mathbf{I}_H = \mathbf{A}^{-1}\mathbf{A}$, Eq. (26) is expanded into

$$
\begin{aligned}
p(\mathbf{w}, \mathbf{y}|\mathbf{X}) &= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}\Big(\mathbf{w}^\top\mathbf{A}\mathbf{w} - 2\mathbf{w}^\top\mathbf{A}\underbrace{\mathbf{A}^{-1}\beta\mathbf{\Phi}^\top(\mathbf{\Pi}^\top\mathbf{y})}_{\bar{\mathbf{w}}} + \beta\mathbf{y}^\top\mathbf{y}\Big)\Big\} \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}\Big(\mathbf{w}^\top\mathbf{A}\mathbf{w} - 2\mathbf{w}^\top\mathbf{A}\bar{\mathbf{w}} + \beta\mathbf{y}^\top\mathbf{y} + \bar{\mathbf{w}}^\top\mathbf{A}\bar{\mathbf{w}} - \bar{\mathbf{w}}^\top\mathbf{A}\bar{\mathbf{w}}\Big)\Big\} \\
&= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top\mathbf{A}(\mathbf{w} - \bar{\mathbf{w}}) - \frac{1}{2}\Big(\beta\mathbf{y}^\top\mathbf{y} - \bar{\mathbf{w}}^\top\mathbf{A}\bar{\mathbf{w}}\Big)\Big\}. \qquad (27)
\end{aligned}
$$

Thus we get the posterior distribution (17) from $p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}}) \propto p(\mathbf{w}, \mathbf{y}|\mathbf{X}) \propto \exp\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top\mathbf{A}(\mathbf{w} - \bar{\mathbf{w}})\}$.

Derivation of predictive distribution $p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}})$ (18). From the property of marginal and conditional Gaussian distributions, we get

$$\begin{aligned} p_{\mathrm{BSLR}}(y_*|\mathbf{x}_*, \mathcal{D}_{\mathrm{SD}}) &= \int p(y_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|\mathcal{D}_{\mathrm{SD}}) d\mathbf{w} \\ &= \int \mathcal{N}(y_*; \phi(\mathbf{x}_*)^\top \mathbf{w}, \beta^{-1}) \mathcal{N}(\mathbf{w}; \bar{\mathbf{w}}, \mathbf{A}^{-1}) d\mathbf{w} \\ &= \mathcal{N}(y_*; \phi(\mathbf{x}_*)^\top \bar{\mathbf{w}}, \beta^{-1} + \phi(\mathbf{x}_*)^\top \mathbf{A}^{-1} \phi(\mathbf{x}_*)). \end{aligned} \tag{28}$$

Derivation of logarithm of marginal distribution $\mathcal{L}_{\mathrm{BSLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{SD}})$ $(= \log \int p(\mathbf{y}, \mathbf{w}|\mathbf{X}) d\mathbf{w})$ (19).

By the trick of adding $0 = \bar{\mathbf{w}}^\top \mathbf{A}^{-1} \bar{\mathbf{w}} - \bar{\mathbf{w}}^\top \mathbf{A}^{-1} \bar{\mathbf{w}}$ and $\mathbf{I}_N = \mathbf{\Pi}^\top \mathbf{\Pi}$ with the definitions $\bar{\mathbf{w}} = \beta \mathbf{A}^{-1} \mathbf{\Phi}^\top \mathbf{\Pi}^\top \mathbf{y}$ and $\mathbf{A} = \alpha \mathbf{I}_H + \beta \mathbf{\Phi}^\top \mathbf{\Phi}$, we can expand the final term of the argument of the exponential function in (27) as follows.

$$\begin{aligned} &-\frac{1}{2}\Big(\beta \mathbf{y}^\top \mathbf{y} - \bar{\mathbf{w}}^\top \mathbf{A} \bar{\mathbf{w}}\Big) = -\frac{1}{2}\Big(\beta \mathbf{y}^\top \mathbf{y} - 2\bar{\mathbf{w}}^\top \mathbf{A} \bar{\mathbf{w}} + \bar{\mathbf{w}}^\top \mathbf{A} \bar{\mathbf{w}}\Big) \\ &= -\frac{1}{2}\Big(\beta \mathbf{y}^\top \mathbf{y} - 2\bar{\mathbf{w}}^\top \underbrace{\mathbf{A}\mathbf{A}^{-1}}_{\text{cancel out}} \beta \mathbf{\Phi}^\top \mathbf{\Pi}^\top \mathbf{y} + \bar{\mathbf{w}}^\top (\alpha \mathbf{I}_H + \beta \mathbf{\Phi}^\top \mathbf{\Phi}) \bar{\mathbf{w}}\Big) \\ &= -\frac{1}{2}\Big(\beta \mathbf{y}^\top \mathbf{y} - 2\bar{\mathbf{w}}^\top \beta \mathbf{\Phi}^\top \mathbf{\Pi}^\top \mathbf{y} + \beta \bar{\mathbf{w}}^\top \mathbf{\Phi}^\top \underbrace{\mathbf{\Pi}^\top \mathbf{\Pi}}_{\mathbf{I}_N} \mathbf{\Phi} \bar{\mathbf{w}} + \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}}\Big) \\ &= -\frac{1}{2}\Big(\beta (\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}})^\top (\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}) + \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}}\Big) = -\frac{\beta}{2}||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 - \frac{\alpha}{2}\bar{\mathbf{w}}^\top \bar{\mathbf{w}}. \end{aligned} \tag{29}$$

Substituting (29) into (27), we get the following expression of joint distribution:

$$p(\mathbf{w}, \mathbf{y}|\mathbf{X}) = \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top \mathbf{A}(\mathbf{w} - \bar{\mathbf{w}}) - \frac{\beta}{2}||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 - \frac{\alpha}{2}\bar{\mathbf{w}}^\top \bar{\mathbf{w}}\Big\}.$$

From the property used for normalizing constant of Gaussian distributions, $\int \exp\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top \mathbf{A}(\mathbf{w} - \bar{\mathbf{w}})\} d\mathbf{w} = (2\pi)^{\frac{H}{2}} |\mathbf{A}|^{-1/2}$ holds and thus we get

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \int p(\mathbf{y}, \mathbf{w}|\mathbf{X}) d\mathbf{w} \\ &= \int \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} \exp\Big\{-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top \mathbf{A}(\mathbf{w} - \bar{\mathbf{w}}) - \frac{\beta}{2}||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 - \frac{\alpha}{2}\bar{\mathbf{w}}^\top \bar{\mathbf{w}}\Big\} d\mathbf{w} \\ &= \Big(\frac{\alpha}{2\pi}\Big)^{\frac{H}{2}} \Big(\frac{\beta}{2\pi}\Big)^{\frac{N}{2}} (2\pi)^{\frac{H}{2}} |\mathbf{A}|^{-1/2} \exp\Big\{-\frac{\beta}{2}||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 - \frac{\alpha}{2}\bar{\mathbf{w}}^\top \bar{\mathbf{w}}\Big\}. \end{aligned} \tag{30}$$

We get $\mathcal{L}_{\mathrm{BSLR}}(\mathbf{w}; \mathcal{D}_{\mathrm{SD}})$ (19) by taking the logarithm of (30). The expression of $E(\bar{\mathbf{w}})$ using trace function is obtained by the trace trick ($\mathbf{z}^\top \mathbf{z} = \mathrm{tr}(\mathbf{z}\mathbf{z}^\top)$).

$$\begin{aligned} E(\bar{\mathbf{w}}) &= \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}} + \beta ||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 \\ &= \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}} + \beta \sum\nolimits_{i=1}^{M} (\mathbf{y}_i - \mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}})^\top (\mathbf{y}_i - \mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}}) \\ &= \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}} + \beta \sum\nolimits_{i=1}^{M} \{\mathbf{y}_i^\top \mathbf{y}_i - 2\mathrm{tr}(\mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}} \mathbf{y}_i^\top) + (\mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}})^\top \mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}}\} \\ &= \alpha \bar{\mathbf{w}}^\top \bar{\mathbf{w}} - 2\beta \sum\nolimits_{i=1}^{M} \mathrm{tr}(\mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}} \mathbf{y}_i^\top) + \beta \sum\nolimits_{i=1}^{M} \{\mathbf{y}_i^\top \mathbf{y}_i + \bar{\mathbf{w}}^\top \mathbf{\Phi}_i^\top \underbrace{\mathbf{\Pi}^\top \mathbf{\Pi}}_{\mathbf{I}_N} \mathbf{\Phi}_i \bar{\mathbf{w}}\} \\ &= -2\beta \sum\nolimits_{i=1}^{M} \mathrm{tr}(\mathbf{\Pi}_i \mathbf{\Phi}_i \bar{\mathbf{w}} \mathbf{y}_i^\top) + \beta \mathbf{y}^\top \mathbf{y} + \bar{\mathbf{w}}^\top \underbrace{(\alpha \mathbf{I}_H + \beta \mathbf{\Phi}^\top \mathbf{\Phi})}_{\mathbf{A}} \bar{\mathbf{w}}. \end{aligned}$$

E Connection Between BLR and BSLR

Here we show the connection between BLR and BSLR. Similar to the relationship between GPSR and GPR discussed in § 4, we can prove the equivalence of their predictive distributions and marginal likelihoods using the definition of PCD. We begin by presenting a proof about predictive distributions.

Proof of Proposition 5.1. We need to show the equivalence of the posterior mean of BSLR $\bar{\mathbf{w}}$ (17) and that of BLR $\underline{\bar{\mathbf{w}}}$ (21) when pseudo coupled data $\mathcal{D}_{\mathrm{PCD}}$ are given. The design matrices ($\mathbf{\Phi}$,$\underline{\mathbf{\Phi}}$) and posterior variances ($\mathbf{A}$,$\underline{\mathbf{A}}$) are equivalent from the definition. So we get $\bar{\mathbf{w}} = \beta \mathbf{A}^{-1} \mathbf{\Phi}^\top \mathbf{\Pi}^\top \mathbf{y} = \beta \underline{\mathbf{A}}^{-1} \underline{\mathbf{\Phi}}^\top \tilde{\mathbf{y}} = \underline{\bar{\mathbf{w}}}$. □

Marginal likelihood of BSLR (19) is also equivalent to that of BLR (23) with pseudo coupled data.

Proof of Proposition 5.2. Similar to the proof of Prop. 5.1, we can show the equivalence of the design matrices, posterior means and variances. Equivalence of the remaining terms can be shown as follows:

$$||\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}||^2 = (\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}})^\top \underbrace{\mathbf{\Pi}\mathbf{\Pi}^\top}_{\mathbf{I}_N} (\mathbf{y} - \mathbf{\Pi}\mathbf{\Phi}\bar{\mathbf{w}}) = ||\mathbf{\Pi}^\top \mathbf{y} - \mathbf{\Phi}\bar{\mathbf{w}}||^2 = ||\tilde{\mathbf{y}} - \underline{\mathbf{\Phi}\bar{\mathbf{w}}}||^2. \quad \square$$

F Inference Procedure for SS-GPSR (BSLR with RFF)

We detail the inference procedure of SS-GPSR explained in §5.2. Although not explicitly mentioned in the main text, we can estimate both the parameter posterior and hyperparameters α, β in the BLR subroutine [22, 44]. The equivalence of marginal likelihoods between BSLR and BLR (shown in Prop. 5.2) enables the use of existing BLR as subroutines, making the algorithm easier to implement.

Algorithm 2 shows the pseudo code for inference procedure. Overall structure is analogous to Alg. 1. At initialization (line 1-5), permutation matrix $\mathbf{\Pi}$ and the response vector of pseudo coupled data (Definition 4.1) $\tilde{\mathbf{y}}$ is set. Also, vector $\mathbf{s}_h$ used in RFF is sampled from the kernel's spectral density. At each (algorithmic) time step t, BLR-step (line 7-8) and Correspondence Update (CU)-Step (line 9-13) are iteratively conducted. As the BLR-step, posterior of the parameters $\mathbf{w}$ and hyperparameter α, β is update by applying BLR using pseudo coupled data represented by $\mathbf{X}$ and $\tilde{\mathbf{y}}$. Next, as CU-step, permutation matrix $\tilde{\mathbf{y}}$ is updated by sort operation.

G Implementation Details of Experiments

This appendix explains the details of our experimental setting and implementation.

Dataset: As stated in the Experiments section, we used four publicly available data sets, Airfoil, Concrete, Housing, and MPG. The description of each dataset and the pre-processing we applied are as follows. Airfoil records 1503 samples obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections with 5 features including angle, velocity and so on. Concrete records 1030 samples of the compressive strength of concrete with 8 features including the amount of cement, water and so on. Housing records 506 samples of home prices (MEDV: Median value of owner-occupied homes) in Boston with 13 socioeconomic and environmental features such as average number of rooms per dwelling (RM). MPG records 398 samples of the city-cycle fuel consumption in miles per gallon of automobiles with 7 features including weight and horsepower. We excluded samples with missing values and converted the categorical feature into a one-hot vector for MPG. Input features were standardized for all data sets.

Oracle method (LR, GPR and DR): For oracle methods other than DR, we used scikit-learn[8]. Specifically, LR uses sklearn.linear_model.LinearRegression and GPR uses sklearn.gaussian_process.GaussianProcessRegressor with default settings (we adopted Gaussian kernel for GPR). Details of DR are presented in the next paragraph.

Neural network-based methods (DR and SDR): As stated in the Experiments section, we used a one-hidden-layer feedforward neural network with the ReLU activation function for neural network-based methods (DR and SDR). Hyperparameters of these methods were set following [12]. The number of units was set to 20 for all problems. The parameters of DR were optimized using Adam [45] with a learning rate of 0.001. The parameters of SDR were optimized by the stochastic sparse EM algorithm using Adam with a learning rate of 0.001. The mini-batch size of DR and that of SDR were 32 and $32/L$, respectively. The maximum number of epochs was 2000 in common. We used

[8] `https://scikit-learn.org`

Algorithm 2 Inference of SS-GPSR (BSLR using Random Fourier Features)

Input: dimension size of RFF H, shuffled data $\mathcal{D}_{\mathrm{SD}}$, maximum number of iterations T_{max}.
Output: parameter posterior $(\bar{\mathbf{w}}, \mathbf{A}^{-1})$, hyperparameter α, β, and response vector of PCD $\tilde{\mathbf{y}}$

/* Initialization */
Randomly initialize $\mathbf{\Pi}$ (Optional: $\mathbf{\Pi}$ could be initialized by solving SLR).
Set the response vector of pseudo coupled data (Definition 4.1) as $\tilde{\mathbf{y}} \leftarrow \mathbf{\Pi}^\top \mathbf{y}$.
Sort the elements of $\mathbf{y}_i = (y_{i1}, \cdots, y_{iL})^\top$. // Ascending sort to simplify CU-Step.
Sample $\mathbf{s}_h$ from the kernel's spectral density.
for $t = 1$ to T_{max} **do**
  /* BLR-Step (Optimization of parameter's posterior distribution and hyperparameters) */
  Update $\bar{\mathbf{w}}, \mathbf{A}^{-1}, \alpha, \beta$ (This could be done by e.g., BaysianRidge.fit($\mathbf{X}, \tilde{\mathbf{y}}$) in scikit-learn).
  /* Correspondence Update-Step (Optimization of $\mathbf{\Pi}$ by sorting operation) */
  for $i = 1$ to M **do**
    $\mathbf{ids} \leftarrow \mathrm{argsort}(\mathbf{v}_i)$ // Get the indices sorting elements of $\mathbf{v}_i (= \mathbf{\Phi}_i \bar{\mathbf{w}})$ in ascending order.
    $\tilde{\mathbf{y}}_i \leftarrow \mathbf{y}_i[\mathrm{argsort}(\mathbf{ids})]$ //Re-orderring of $\mathbf{y}_i$ by (argsort of) $\mathbf{ids}$ provides $\tilde{\mathbf{y}}_i \leftarrow \mathbf{\Pi}^\top \mathbf{y}_i$.
  end for
end for

early-stopping with validation data (coupled data for DR and shuffled data for SDR) [42]. The above was implemented using PyTorch [46]. Experiments were run on a computer with Apple M1.

Proposed methods (GPSR and SS-GPSR): Here, we provide details of the proposed methods (GPSR and SS-GPSR). Their inference algorithms are summarized in Algorithm 1 (Appendix C) and Algorithm 2 (Appendix F), respectively. As mentioned in the Experiments section, GPSR utilized warm initialization, i.e., the initial value of the permutation matrix is set using the estimated result of SLR. For SS-GPSR, ten different runs were performed with various initial permutation matrices, and the solution that maximized the marginal likelihood was selected. For GPSR, the maximum number of iterations T_{max} was set to 100. The parameter for simulated annealing was configured with $S_{max} = 10^{-1} N \log_2(L)$, an initial temperature of $T = 1.0$ and a cooling rate $\gamma = 0.99$. In the case of SS-GPSR, the maximum number of iterations T_{max} was set to 20. The dimension size of RFF H was fixed at 100 for all datasets. Note that for GPR and BLR steps in Algorithm 1 and 2, we used the scikit-learn analogous to oracle methods (BLR uses sklearn.linear_model.BayesianRidge). So hyperparameters such as precision α and β in (SS-)GPSR are estimated when applying GPR or BLR. So GPSR and SS-GPSR do not use validation data unlike SDR.