Efficient Approximate Inference for Stationary Kernel on Frequency Domain

Yohan Jung[1] **Kyungwoo Song**[2] **Jinkyoo Park**[1]

Abstract

Based on the Fourier duality between a stationary kernel and its spectral density, modeling the spectral density using a Gaussian mixture density enables one to construct a flexible kernel, known as a Spectral Mixture kernel, that can model any stationary kernel. However, despite its expressive power, training this kernel is typically difficult because scalability and overfitting issues often arise due to a large number of training parameters. To resolve these issues, we propose an approximate inference method for estimating the Spectral mixture kernel hyperparameters. Specifically, we approximate this kernel by using the finite random spectral points based on Random Fourier Feature and optimize the parameters for the distribution of spectral points by sampling-based variational inference. To improve this inference procedure, we analyze the training loss and propose two special methods: a sampling method of spectral points to reduce the error of the approximate kernel in training, and an approximate natural gradient to accelerate the convergence of parameter inference.

1. Introduction

In constructing a Gaussian process ($\mathcal{GP}$) model, selecting a proper kernel function is vital because the selected kernel determines the overall structure of the target function by specifying the covariance of $\mathcal{GP}$, a prior for the target function. In general, under the assumption that the covariance of two target function values is related to the distance between two inputs, a stationary kernel whose output is a function of the difference between two inputs, is commonly used. One of the main characteristics of the stationary kernel is that it has Fourier duality with its spectral density. That is, stationary kernel $k(x_1 - x_2)$ for inputs x_1 and x_2 can be expressed by an inverse Fourier transform of its spectral density $p(s)$ as

$$k(x_1 - x_2) = \int e^{i2\pi s^{\mathrm{T}}(x_1 - x_2)} p(s) ds. \tag{1}$$

For example, the RBF kernel, which is defined as $k(x_1 - x_2) = \exp(-\frac{1}{2}\|\frac{2\pi(x_1-x_2)}{l}\|^2)$, is obtained when specifying $p(s) = N(s; 0, l^{-2}I)$ with RBF hyperparameter l in Eq. (1). Based on the Fourier duality, selecting a specific kernel is equivalent to modeling the form of $p(s)$. If the model assumption for the spectral density is not flexible enough to express the spectral density of the true kernel, the $\mathcal{GP}$ model with the induced kernel may not explain the dataset well. Thus, a flexible model for $p(s)$ is necessary in order to find a stationary kernel that can describe the dataset well.

As an attempt to construct a flexible kernel, Wilson & Adams (2013) represent $p(s)$ as a Gaussian mixture density $\sum_{q=1}^{Q} w_q N(s; \mu_q, \mathrm{diag}(\sigma_q^2))$, where $\mu_q \in R^D$ and $\mathrm{diag}(\sigma_q^2) \in R^{D \times D}$ are the mean and the diagonal covariance matrix, respectively. The Gaussian mixture density can represent any $p(s)$ flexibly based on the universal approximation theorem (Kostantinos, 2000) and thus, it can induce the flexible kernel (SM kernel) by Eq. (1). In fact, Wilson & Adams (2013); Wilson et al. (2014) demonstrate that the SM kernel can find the suitable stationary kernel that captures the covariance structure of the given datasets. SM kernel has been employed in many applications: for example, prediction on the time-series datasets (Ploysuwan, 2014; Kupilik & Witmer, 2018; Chen et al., 2021), analysis the brain states (Ulrich et al., 2014; 2015), Bayesian optimization (Wu et al., 2017; Raj et al., 2020).

However, the improved expressive power also incurs two challenges at the cost of kernel flexibility: high computational cost and overfitting issue. Specifically, given N data points, training the SM kernel requires $\mathcal{O}((Q + 2QD)N^2)$ computations per iteration in addition to $\mathcal{O}(N^3)$ computations for the inversion of $N \times N$ kernel matrix (Rasmussen, 2004). When N is large, training the SM kernel is generally difficult due to high computational cost. In addition, training the SM kernel by conventional approach, i.e. the maximization of the marginal likelihood, may induce overfitting because of its many parameters $(Q + 2QD)$ (Warnes & Ripley, 1987; Simpson et al., 2021b).

[1] Department of Industrial & Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon , South Korea [2] Department of Artificial Intelligence, University of Seoul, Seoul, South Korea. Correspondence to: Yohan Jung <becre1776@kaist.ac.kr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

To tackle these issues, we propose an approximate inference method for estimating the SM kernel parameters $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ efficiently. To be specific, we approximate the kernel by employing the Random Fourier feature (RFF) with finite spectral points because the approximate kernel by finite spectral points can be trained in a scalable manner (Lazaro-Gredilla et al., 2010). Instead of optimizing the spectral points directly, which could induce the overfitting (Lazaro-Gredilla et al., 2010), we introduce the distribution on the spectral points with the prior distribution, and then optimize the parameters of the distribution through sampling-based variational inference (Kingma & Welling, 2013). Based on the proposed inference scheme, we propose two special strategies to infer the parameters for the distribution of the spectral points efficiently:

- We propose a sampling strategy of spectral points to minimize the error of approximate kernel occurring during training. Specifically, we optimally control the number of spectral points to be sampled from the mixture components $N(\mu_q, \text{diag}(\sigma_q^2))$ for $q = 1, .., Q$ when the total number of spectral points is given.
- We employ a natural gradient to optimize the parameters for the distribution of the spectral points because the natural gradient reflects the curvature of probability density and can thus be used to update probability density parameters efficiently (Amari, 1998).

We validate that the proposed sampling stabilizes the approximate inference, and the approximate natural gradient leads the fast convergence of parameter inference empirically. Our contributions can be summarized as follows:

- **Scalability**: we approximate the SM kernel by using the distribution of the finite spectral points, and use a sampling-based variational inference for training.
- **Stability**: we analyze the evidence lower bound (ELBO) of sampling-based variational inference, and propose a sampling strategy for spectral points to reduce the error of the approximate kernel during the training.
- **Fast convergence**: we propose an approximate natural gradient to update the parameters for the distribution of spectral points effectively.

2. Preliminaries

Spectral Mixture (SM) Kernel : Wilson & Adams (2013) represent the spectral density $p(s)$ as a weighted mixture of Q symmetric Gaussian densities with the hyperparameters $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ as

$$p(s) = \sum_{q=1}^{Q} w_q p_q(s) \tag{2}$$

$$p_q(s) = \frac{N(s|\mu_q, \text{diag}(\sigma_q^2)) + N(-s|\mu_q, \text{diag}(\sigma_q^2))}{2} \tag{3}$$

where $w_q \in R^+$ is the weight parameter, $\mu_q = [\mu_{(q,1)}, .., \mu_{(q,D)}] \in R^D$ the mean parameter, and $\sigma_q^2 = [\sigma_{(q,1)}^2, .., \sigma_{(q,D)}^2] \in R^D$ the covariance parameter, i.e., the diagonal part of $\text{diag}(\sigma_q^2) \in R^{D \times D}$. The symmetry of the spectral density is used to define the kernel as a real-valued function. They then apply the inverse Fourier transform to $p(s)$ using Eq. (1) to derive the SM kernel $k_{SM}(x_1 - x_2)$

$$\sum_{q=1}^{Q} w_q \exp\left(-2\pi^2 (\sigma_q^T (x_1 - x_2))^2\right) \cos\left(2\pi {\mu_q}^T (x_1 - x_2)\right).$$

Random Fourier Feature (RFF): Rahimi & Recht (2008) approximate the stationary kernel $k(x_1 - x_2)$, by applying a Monte Carlo integration to Eq. (1) with M spectral points $\{s_i\}_{i=1}^{M}$ sampled from $p(s)$,

$$\begin{aligned} k(x_1 - x_2) &\approx \frac{1}{M} \sum_{i=1}^{M} \cos(2\pi {s_i}^T (x_1 - x_2)) \qquad (4) \\ &= \phi(x_1, \{s_i\}_{i=1}^{M}) \phi(x_2, \{s_i\}_{i=1}^{M})^T, \end{aligned}$$

$\phi(x, \{s_i\}_{i=1}^{M}) = \frac{1}{\sqrt{M}} \left[\cos\left(2\pi s_{1:M}^T x\right), \sin\left(2\pi s_{1:M}^T x\right)\right] \in R^{2M}$ is a random feature. Then, the random feature matrix $\Phi(X) = \left[\phi(x_1, \{s_i\}_{i=1}^{M}); ...; \phi(x_N, \{s_i\}_{i=1}^{M})\right] \in R^{N \times 2M}$, for inputs $X = \{x_n\}_{n=1}^{N}$ can approximate the kernel Gram matrix $K(X, X) \in R^{N \times N}$ by $\hat{K}(X, X; \{s_i\}_{i=1}^{M}) := \Phi(X)\Phi^T(X)$.

Sparse Spectrum $\mathcal{GP}$: Approximate kernel obtained by the RFF enables the scalable training for $\mathcal{GP}$ model (Lazaro-Gredilla et al., 2010). Let f be a function, with $\mathcal{GP}$ prior, to model the relationship between $X = \{x_n\}_{n=1}^{N}$ and $Y = \{y_n\}_{n=1}^{N}$. Then, the prior distribution of $f(X) = [f(x_1), .., f(x_N)]$ using the approximate kernel of Eq. (4) can be defined as

$$p\left(f(X) | \{s_i\}_{i=1}^{M}\right) = N(f(X); 0, \hat{K}(X, X; \{s_i\}_{i=1}^{M})). \tag{5}$$

If likelihood $p(Y|f(X))$ is defined as $N\left(Y|f(X), \sigma_\epsilon^2 I\right)$, then the marginal likelihood $p(Y|X, \{s_i\}_{i=1}^{M})$ is exactly computed as $N(Y; 0, \hat{K}(X, X; \{s_i\}_{i=1}^{M}) + \sigma_\epsilon^2 I)$.

To find the optimal spectral points $\{s_i\}_{i=1}^{M}$ (regarded as the hyperparameters) and noise hyperparameter σ_ϵ^2, $\log p(Y|X, \{s_i\}_{i=1}^{M})$ is maximized based on gradient-based optimization. Evaluating $\log p(Y|X, \{s_i\}_{i=1}^{M})$ uses $O(NM)$ memory and takes $O(NM^2)$ computational time for computing the inversion and determinant by inversion lemma. When $M \ll N$, employing $\log p(Y|X, \{s_i\}_{i=1}^{M})$ for training takes less training time because the original $\mathcal{GP}$ model training using the exact kernel uses $O(N^2)$ memory and takes $O(N^3)$ computation time (Rasmussen, 2004).

Natural Gradient Optimization (NGO): NGO is known to be efficient in updating the probability density parameters (Amari, 1998). Given the loss $\mathcal{L}(\theta)$ parameterized by the

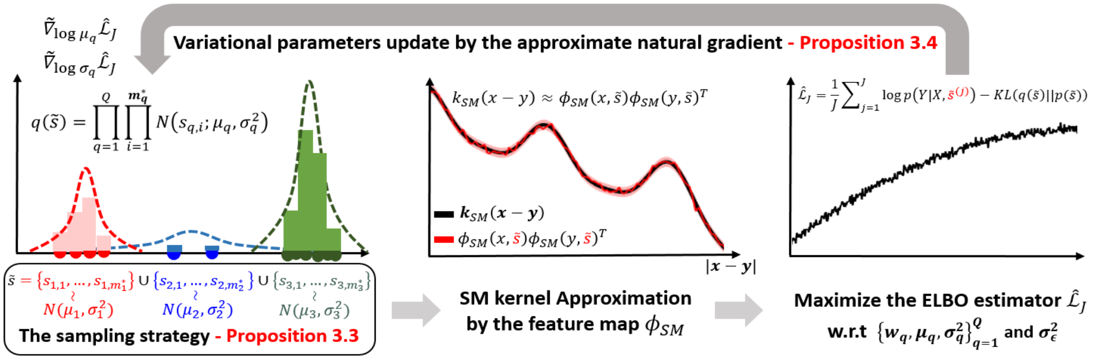

$\tilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J$
$\tilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J$
Variational parameters update by the approximate natural gradient - Proposition 3.4
$q(\tilde{s}) = \prod_{q=1}^{Q} \prod_{i=1}^{m_q^*} N(s_{q,i}; \mu_q, \sigma_q^2)$
$\tilde{s} = \{s_{1,1}, ..., s_{1,m_1^*}\} \cup \{s_{2,1}, ..., s_{2,m_2^*}\} \cup \{s_{3,1}, ..., s_{3,m_3^*}\}$
$N(\mu_1, \sigma_1^2)$
$N(\mu_2, \sigma_2^2)$
$N(\mu_3, \sigma_3^2)$
The sampling strategy - Proposition 3.3
$k_{SM}(x-y) \approx \phi_{SM}(x,\tilde{s})\phi_{SM}(y,\tilde{s})^T$
$k_{SM}(x-y)$
$\phi_{SM}(x,\tilde{s})\phi_{SM}(y,\tilde{s})^T$
$|x-y|$
SM kernel Approximation by the feature map ϕ_{SM}
$\hat{\mathcal{L}}_J = \frac{1}{J}\sum_{j=1}^{J} \log p(Y|X, \tilde{s}^{(j)}) - KL(q(\tilde{s})||p(\tilde{s}))$
Maximize the ELBO estimator $\hat{\mathcal{L}}_J$ w.r.t $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ and σ_ϵ^2

Figure 1. Outline of the proposed approximate inference procedure.

parameter θ of the probability density $p_\theta(z)$, and a small $\epsilon > 0$, the natural gradient $\tilde{\nabla}_\theta \mathcal{L}(\theta)$ can be defined as

$$\tilde{\nabla}_\theta \mathcal{L}(\theta) = \underset{\{\Delta\theta; KL(p_\theta \| p_{\theta+\Delta\theta}) = \epsilon\}}{\arg\min} \mathcal{L}(\theta + \Delta\theta). \tag{6}$$

By assuming that $\Delta\theta$ is small, solving the optimization problem of Eq. (6) using the Lagrangian method with the first-order Taylor approximation of $\mathcal{L}(\theta + \Delta\theta) \approx \mathcal{L}(\theta) + \nabla_\theta \mathcal{L}(\theta)^T \Delta\theta$ and $KL(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^T F_\theta \Delta\theta$ (Pascanu & Bengio, 2013) with Fisher Information matrix $F_\theta = \mathrm{E}_{p_\theta(z)}[\nabla \log p_\theta(z) \nabla \log p_\theta(z)^T]$, induces the natural gradient to be expressed as

$$\tilde{\nabla}_\theta \mathcal{L}(\theta) = \arg\min_{\Delta\theta} \mathcal{L}(\theta) + \nabla_\theta \mathcal{L}(\theta)^T \Delta\theta + \frac{\lambda}{2} \Delta\theta^T F_\theta \Delta\theta,$$

where λ is the Lagrangian multiplier. This explains that natural gradient is a second-order gradient using the local geometric information around θ with Fisher Information matrix F_θ, and why natural gradient could lead the fast convergence of probability density parameter inference.

3. Methodology

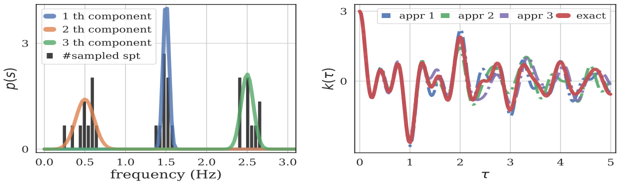
In this section, we first describe the approximate inference method to estimate the hyperparameters of the SM kernel $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$. Then, we analyze our approximate inference and discuss the two special schemes to improve the proposed inference. We describe the outline of the approximate inference scheme in Fig. 1.

3.1. Approximate Inference for $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$

Kernel Approximation: Since the SM kernel can also be expressed as the weighted sum of inverse Fourier transform of each Gaussian component $p_q(s)$ of Eq. (3)

$$k_{SM}(x_1 - x_2) = \sum_{q=1}^{Q} w_q \underbrace{\left(\int e^{i 2\pi s^T (x_1 - x_2)} p_q(s) ds \right)}_{q-th\ integral},$$

the SM kernel can be approximated in an unbiased manner by approximating each q-th integral term for $q = 1, .., Q$ based on RFF of Eq. (4), and then summing these approximated terms after multiplying the weight w_q to q-th approximation.



(a) sampling from Eq. (7) (b) kernel sampling

Figure 2. Generative Process; Fig. 2a shows the sampled spectral points from Eq. (7). (Q=3, m_q=10). Fig. 2b shows 3 samples of the approximate kernel function and its exact kernel function (red).

For approximating each q-th integral term, we consider m_q spectral points, i.e., $\{s_{q,1}, .., s_{q,m_q}\}$, each of which follows the Gaussian distribution with the shared parameters $\{\mu_q, \sigma_q^2\}$ i.e., $\prod_{i=1}^{m_q} N(s_{q,i}; \mu_q, \sigma_q^2)$. We introduce the distribution of the spectral points for all Q mixtures as

$$q(\cup_{q=1}^{Q} \{s_{q,1}, .., s_{q,m_q}\}) = \prod_{q=1}^{Q} \prod_{i=1}^{m_q} N(s_{q,i}; \mu_q, \sigma_q^2), \tag{7}$$

Given the total number of spectral points $M = \sum_{q=1}^{Q} m_q$, the distribution of spectral points in Eq. (7) enables us to define the random feature map with weights $\{w_q\}_{q=1}^{Q}$ as

$$\begin{aligned} &\phi_{\mathrm{SM}}(x; \cup_{q=1}^{Q} \{s_{q,1}, .., s_{q,m_q}\}) \qquad (8) \\ &= \left[\sqrt{w_1} \phi\left(x, \{s_{1,i}\}_{i=1}^{m_1}\right), ..., \sqrt{w_Q} \phi\left(x, \{s_{Q,i}\}_{i=1}^{m_Q}\right) \right], \end{aligned}$$

where ϕ is the random feature map defined in Eq. (4). Fig. 2 shows the sampled spectral points from Eq. (7), and three samples of the approximate kernel constructed by Eq. (8) with its exact kernel. We later elaborate on how to allocate total M spectral points into each mixture component, i.e. $\{m_q\}_{q=1}^{Q}$, that reflects the parameters $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$.

Using the feature map of Eq. (8), we can define the random feature matrix $\Phi^{\text{SM}}(X)$ for $X = \{x_n\}_{n=1}^{N}$, as explained in Eq. (4), and build the unbiased estimator of $K_{\text{SM}}(X,X)$ such that $\text{E}[\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T] = K_{\text{SM}}(X,X)$. Proposition 3.1 states the error bound of this estimator.

Proposition 3.1. *Let $W_0 = \sqrt{\sum_{q=1}^{Q} w_q^2}$ and $M = Qm_1$ under the assumption $m_1 = .. = m_Q$. Then, for a small $\epsilon > 0$, the error bound for the estimator $\hat{K}_{\text{SM}}(X,X) := \Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T$ is obtained as*

$$\Pr\Big(\big\|\hat{K}_{\text{SM}}(X,X) - K_{\text{SM}}(X,X)\big\|_2 \geq \epsilon\Big) \tag{9}$$
$$\leq N\exp\left(\frac{-3\epsilon^2 M}{NW_0Q\big(6\|K_{\text{SM}}(X,X)\|_2 + 3NW_0\sqrt{Q} + 8\epsilon\big)}\right).$$

where $\|\cdot\|_2$ denotes the matrix spectral norm. See Appendix E.1 for a proof.

Intuitively, this means that the error of $\hat{K}_{\text{SM}}(X,X)$ could be reduced in a probabilistic sense as the number of the sampled spectral points M increases.

ELBO estimator $\hat{\mathcal{L}}_J$: To train the parameters $\{\mu_q, \sigma_q^2\}_{q=1}^{Q}$ of the spectral point distributions, weight parameters $\{w_q\}_{q=1}^{Q}$ and noise parameter σ_ϵ^2, we employ sampling-based VI inference (Kingma & Welling, 2013). For brevity, we denote the spectral points $\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\}$ as $\tilde{s}$, i.e. $\tilde{s} = \cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\}$. We derive ELBO estimator $\hat{\mathcal{L}}_J$ of $\log p(Y|X)$ by using the distribution of spectral points in Eq. (7) as variational distribution $q(\tilde{s})$ as follows:

$$\log p(Y|X) \tag{10}$$
$$\geq \int \log p(Y|X,\tilde{s})q(\tilde{s})d\tilde{s} - \text{KL}(q(\tilde{s})||p(\tilde{s}))$$
$$\approx \frac{1}{J}\sum_{j=1}^{J}\log p(Y|X,\tilde{s}^{(j)}) - \text{KL}(q(\tilde{s})||p(\tilde{s})) := \hat{\mathcal{L}}_J$$

where $\tilde{s}^{(j)}$ denotes the spectral points sampled from $q(\tilde{s})$ at j-th times. Sampling $\tilde{s}^{(j)}$ during optimization is conducted by the reparameterization (RP) trick as $s_{q,i}^{(j)} = \mu_q + \sigma_q \circ \epsilon_i$ with $\epsilon_i \sim N(\epsilon; 0, I)$. The marginal likelihood $p(Y|X,\tilde{s}^{(j)})$ can be computed as Eq. (5) by using the random covariance $\hat{K}_{\text{SM}}^{(j)}(X,X) + \sigma_\epsilon^2 I$ constructed by $\tilde{s}^{(j)}$.

For the tractability of KL term, we set the prior distribution $p(\tilde{s})$ to have same form with $q(\tilde{s})$ of Eq. (7) with the parameters $\cup_{q=1}^{Q}\{\tilde{\mu}_{q,i}, \tilde{\sigma}_{q,i}^2\}_{i=1}^{m_q}$. We compute KL term as $\text{KL}(q(\tilde{s})||p(\tilde{s}))=\sum_{q=1}^{Q}\text{KL}(N(\mu_q,\sigma_q^2)||N(\tilde{\mu}_{q,1},\tilde{\sigma}_{q,1}^2))$ by letting $\tilde{\mu}_{q,i}=\mu_q$, $\tilde{\sigma}_{q,i}^2=\sigma_q^2$ for $i \geq 2$. The prior parameters $\{\tilde{\mu}_{q,1},\tilde{\sigma}_{q,1}^2\}_{q=1}^{Q}$ are initialized by using the prior knowledge such as the empirical spectral density of dataset (Tobar, 2018; de Wolff et al., 2020) or the RBF prior. Further details are explained in Appendix C.3.

We denote this scheme as a sampling-based variational sparse spectrum (SVSS), which is the basic approximate inference method in this work.

Error Analysis of ELBO Estimator $\hat{\mathcal{L}}_J$: We analyze the gap between the log marginal likelihood $\log p(Y|X)$ and its estimator $\frac{1}{J}\sum_{j=1}^{J}\log p(Y|X,\tilde{s}^{(j)})$ in $\hat{\mathcal{L}}_J$ constructed using the sampled spectral points during training. This error is represented as

$$\log p(Y|X) - \Big(\hat{\mathcal{L}}_J + \text{KL}(q(\tilde{s})||p(\tilde{s}))\Big) \tag{11}$$
$$= \frac{1}{2J}\sum_{j=1}^{J} Y^T \hat{K}_{\text{SM}}^{(j)\,-1}\Big(K_{\text{SM}} - \hat{K}_{\text{SM}}^{(j)}\Big)K_{\text{SM}}^{-1}Y + \log\frac{|\hat{K}_{\text{SM}}^{(j)}|}{|K_{\text{SM}}|},$$

where $K_{\text{SM}}{=}K_{\text{SM}}(X,X){+}\sigma_\epsilon^2 I$ with identity matrix I. $\hat{K}_{\text{SM}}^{(j)}$ denotes the corresponding matrix of $\hat{K}_{\text{SM}}^{(j)}(X,X)$. We bound the error in Eq. (11) as follows:

Proposition 3.2. *The error of ELBO estimator $\hat{\mathcal{L}}_J$ in Eq. (11) is bounded as*

$$\log p(Y|X) - \Big(\hat{\mathcal{L}}_J + \text{KL}(q(\tilde{s})||p(\tilde{s}))\Big) \tag{12}$$
$$\leq \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{2\sigma_\epsilon^4}\right)\text{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$$

where $\text{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$ is integrated over the spectral point $\tilde{s}$ with the density $q(\tilde{s})$. See Appendix E.2 for a proof.

Proposition 3.2 implies that when $\text{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$ becomes smaller for given Y and σ_ϵ^2, $\log p(Y|X) - \hat{\mathcal{L}}_J$ is more tightly bounded. According to Proposition 3.1, $\text{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$ could be smaller when the number of total sampled spectral points M becomes larger. However, a large M can in turn increase the computational cost in proportion to M^2. Thus, given fixed M spectral points, we devise a way to reduce $\text{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$ for the stable training with the tight ELBO estimator $\hat{\mathcal{L}}_J$.

3.2. Strategies for Improving Approximate Inference

Weighted Sampling for Spectral Points: K_{SM} and $\hat{K}_{\text{SM}}$ are made by the Q mixture, $K_{\text{SM}} - \hat{K}_{\text{SM}}$ is decomposed as

$$K_{\text{SM}} - \hat{K}_{\text{SM}} = \sum_{q=1}^{Q} w_q\left(K_q - \hat{K}_q\right) \tag{13}$$

where $K_q \in R^{N\times N}$ is the kernel matrix constructed by the q-th spectral density $p_q(s)$ of Eq. (3), and $\hat{K}_q \in R^{N\times N}$ denotes its estimator matrix constructed by the spectral points $\{s_{q,i}\}_{i=1}^{m_q}$. This decomposition implies that given $M{=}\sum_{q=1}^{Q}m_q$ spectral points, allocating each spectral points $\{m_q\}_{q=1}^{Q}$ depending on the weights $\{w_q\}_{q=1}^{Q}$

and errors $\{\hat{K}_q - K_q\}_{q=1}^{Q}$ can reduce the error $\hat{K}_{\text{SM}} - K_{\text{SM}}$. Hence, we find the optimal number of spectral points $\{m_q^*\}_{q=1}^{Q}$ by minimizing $\mathrm{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_F]$, with the Frobenius norm $\|\cdot\|_F$, because minimizing $\mathrm{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_F]$ induces $\mathrm{E}[\|K_{\text{SM}} - \hat{K}_{\text{SM}}\|_2]$ in Proposition 3.2 to be smaller due to $\|A\|_2 \leq \|A\|_F$ for any matrix $A \in R^{m \times n}$. Also, this objective is analytically computable. Proposition 3.3 states (1) optimization problem of finding the optimal number of the spectral points and (2) its solution.

Proposition 3.3. *Given inputs* $X{=}\{x_n\}_{n=1}^{N}$*, let* m_q *be the number of spectral points sampled from* $N(\mu_q, \sigma_q^2)$*, and* $M{=}\sum_{q=1}^{Q} m_q$ *be the total number of spectral points. Let* $a_q{=}\frac{m_q}{M}$ *be the ratio of spectral points. Then, the optimal* $a_1^*, .., a_Q^*{=}\arg\min_{a_1,..,a_Q} \mathrm{E}[\|\hat{K}_{\text{SM}} - K_{\text{SM}}\|_F]$ *is given*

$$a_q^* = \frac{w_q \left[\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}}{\sum_{q=1}^{Q} w_q \left[\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}} \tag{14}$$

where $g_q(\tau){=}1{+}k_q(2\tau){-}2k_q^2(\tau)$ *and* $k_q(\tau)$ *denotes the* q*-th component term in SM kernel related to* $\{\mu_q, \sigma_q^2\}$*. The optimal spectral point* m_q^* *is obtained as the integer closest to* $\max\{1, Ma_q^*\}$*. See Appendix E.3 for a detailed proof.*

Proposition 3.3 implies that each m_q^* spectral point is proportional to the product of weight w_q and sum of the variances for the q-th random features $\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)$.

Approximate Natural Gradient: Employing a natural gradient for updating the parameters of Gaussian distribution can lead the fast convergence of the parameter inference (Khan & Nielsen, 2018; Khan et al., 2018).

We propose to use the approximate natural gradient that updates the parameters of Eq. (7) in the logarithm domain so that updating these parameters should be numerically stable for Cholesky decomposition error in training. The proposed approximate natural gradient can be easily computed from the original gradient via pre-conditioning and normalization. Proposition 3.4 presents the way to compute approximate natural gradient under valid condition.

Proposition 3.4. *Let* $\mu_q^{(t)}$ *and* $\sigma_q^{(t)}$ *be the* t*-th iterated parameters of* $N(\mu_q, \sigma_q^2)$ *which is* q*-th component distribution for* $q(\tilde{s})$*. The natural gradient of* $\hat{\mathcal{L}}_J$ *w.r.t* μ_q *and* σ_q *in log domain, i.e.* $\widetilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J$ *and* $\widetilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J$*, can be approximated as*

$$\begin{aligned} \widetilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J &\approx \left(\frac{\sigma_q^{(t+1)}}{\mu_q^{(t)}} \right)^2 \circ \nabla_{\log \mu_q} \hat{\mathcal{L}}_J \\ \widetilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J &\approx \frac{1}{2} \nabla_{\log \sigma_q} \hat{\mathcal{L}}_J, \end{aligned} \tag{15}$$

for $\left| \left(\frac{\sigma_q^{(t+1)}}{\mu_q^{(t)}} \right)^2 \circ \nabla_{\log \mu_q} \hat{\mathcal{L}}_N \right| < 1$ *and* $\left| \nabla_{\log \sigma_q} \hat{\mathcal{L}}_N \right| < 1$ *in element-wise sense. See Appendix E.4 for a proof.*

The constraints are satisfied by normalizing the revised gradient by its two norm $\|\cdot\|_2$. The derived gradients are used with the adaptive optimizer like ADAM.

4. Related Work

Natural gradient. Natural gradient has been employed to update the variational parameters in variational inference framework (Honkela et al., 2007; Hoffman et al., 2013; Khan & Lin, 2017). For the training of $\mathcal{GP}$ model, natural gradient has been applied to update the variational parameters of the inducing variables (Hensman et al., 2013; Salimbeni et al., 2018). However, to the best of our knowledge, the proposed method is the first method that employs the natural gradient to update the SM kernel hyperparameters instead of the variational parameters of inducing variables.

Spectral Mixture (SM) Kernel. Fourier duality between the stationary kernel and its spectral density has been employed to construct a flexible kernel. Wilson & Adams (2013) employed a mixture of Gaussian density function to model a spectral density , and then constructed the SM kernel. Later, this approach was extended to construct multi-output spectral kernels (Ulrich et al., 2015; Parra & Tobar, 2017; Simpson et al., 2021a) and non-stationary spectral kernels (Remes et al., 2017; Shen et al., 2019). Also, the hybrid spectral kernels combined with neural network (Remes et al., 2018; Xue et al., 2019) have been proposed.

Stationary Kernel Approximation and Its Inference. Approximate stationary kernel by RFF enables scalable training of the stationary kernel such as RBF, Periodic, and SM kernel by optimizing the spectral points of the approximate kernel (Rahimi & Recht, 2008; 2009; Lazaro-Gredilla et al., 2010; Yang et al., 2015; Tompkins & Ramos, 2018; 2020). However, the direct optimization of the spectral points has known to induce the overfitting.

To relax overfitting issue, variational inference scheme have been employed to train the spectral points of approximate kernel. Gal & Turner (2015) introduced the variational distribution of spectral points along with inducing variables, and used them for training the stationary kernel including SM kernel. Since this work has some common points with the proposed inference, we compare the proposed inference and Gal & Turner (2015), referred as VSS, to clarify the difference of each methods and reveal our contribution:

Kernel Approximation. Let us first remind that how the q-th spectral density $p_q(s)$ in Eq. (3) can construct the q-th component kernel $k_q(\tau)$ as

$$k_q(\tau) = \int e^{i2\pi s^T \tau} p_q(s) ds \approx \frac{1}{m_q} \sum_{m=1}^{m_q} \cos\left(2\pi s_{q,m}^T \tau\right),$$

where $s_{q,m}$ denotes m-th spectral points sampled from

$N(s;\mu_q,\mathrm{diag}(\sigma_q^2))$, and m_q denotes the number of the sampled spectral points. As considering $\tau = x - y$, we can represent the sum of cosine as

$$\frac{1}{m_q}\sum_{m=1}^{m_q}\cos\left(2\pi s_{q,m}^T\tau\right) := \phi_q(x)\phi_q(y)^T$$

where the random feature $\phi_q(x)$ is defined differently depending on the inference method. VSS uses the random feature $\phi_q(x) \in R^{m_q}$ as

$$\phi_q(x) = \sqrt{\frac{2}{m_q}}\left[..., \cos\left(2\pi s_{q,m}^T x + b_{q,m}\right), ...\right],$$

by using random phase parameters $\{b_{q,m}\}_{m=1}^{m_q} \sim U([0,2\pi])$. This leads $O(NM^2)$ computational complexity for training for N dataset, and the training of the additional phase parameters $\{b_{q,m}\}_{m=1}^{m_q}$.

On the other hand, SVSS uses the random feature $\phi_q(x) \in R^{2m_q}$ as

$$\phi_q(x) = \sqrt{\frac{1}{m_q}}\left[..., \cos\left(2\pi s_{q,m}^T x\right), \sin\left(2\pi s_{q,m}^T x\right), ...\right].$$

This results in $O(N(2M)^2)$ computational complexity without training additional phase parameters. We use this feature because this has smaller variance in approximating RBF kernel as analyzed in (Sutherland & Schneider, 2015).

Spectral Points Modeling. As considering the spectral points $\{s_{q,i}\}_{i=1}^{m_q} \sim N(s;\mu_q,\mathrm{diag}(\sigma_q^2))$ are sampled as

$$s_{q,i} = \mu_q + \sigma_q \circ z_i, \qquad z_i \sim N(0, I),$$

VSS introduces the variational distribution on z_i, i.e $\{q(z_i)\}_{i=1}^{m_q}$ with $q(z_i) = N(z_i;\mathbf{m}_i,\mathbf{s}_i)$, to relax the overfitting. In training, VSS optimizes both the parameters of variational distribution of the spectral points $\{\mathbf{m}_i,\mathbf{s}_i\}_{i=1}^{m_q}$ for $q = 1, .., Q$, and kernel hyperparameter $\{w_q,\mu_q,\sigma_q^2\}$ together. However, we believe that using each parameters $\{\mathbf{m}_i,\mathbf{s}_i\}_{i=1}^{m_q}$ for the variational distributions of $\{z_i\}_{i=1}^{m_q}$ seems to be redundant in sense of modeling the weighted Gaussian spectral density $p(s)$ in Eq. (2) because the kernel hyperpameters $\{\mu_q,\sigma_q\}$ and vaiational parameters $\{\mathbf{m}_i,\mathbf{s}_i\}_{i=1}^{m_q}$ are all used for modeling the q-th Gaussian spectral density $p_q(s)$ in Eq. (3).

On the other hand, SVSS introduces the variational distribution on $s_{q,i}$ directly, i.e., $\{q(s_{q,i})\}_{i=1}^{m_q}$ that have the shared variational parameters $\{\mu_q,\sigma_q^2\}$, but could have different prior $p(s_{q,i})$ for each $s_{q,i}$. For training, we optimize variational parameters $\{\mu_q,\sigma_q^2\}_{q=1}^{Q}$, which can also be regraded as the kernel hyperparameters, and weight hyperparameters $\{w_q\}_{q=1}^{Q}$ together. Thus, SVSS uses the smaller number of trainable parameters compared to VSS, and can alleviate overfitting.

Evidence Lower bound of $\log p(Y|X)$. VSS takes the lower bound of $p(Y|X)$ as follows:

$$\log p(Y|X) = \iint p(Y|X,A,s)p(A)p(\tilde{s})d(A)d\tilde{s} \;\geq$$
$$\iint \log p(Y|X,A,\tilde{s})q(A,\tilde{s})dAd\tilde{s} - \mathrm{KL}(q(\tilde{s},A)||p(\tilde{s},A))$$

where $\tilde{s} = \cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\}$ denotes M spectral points, and $A \in R^M$ denotes the random weight. $p(Y|X,A,\tilde{s})$ is computed as $N(Y;\Phi(X)A,\sigma_\epsilon^2 I)$ with random feature matrix $\Phi(X) \in R^{N\times M}$ obtained by the random feature of VSS. This can be understood as Bayesian Linear regression that uses the random feature matrix $\Phi(X)$, random weight parameter A, and variance parameters σ_ϵ^2.

On the other hand, SVSS takes the lower bound of $p(Y|X)$ as shown in Eq. (10). This difference comes from that VSS employs the classical variational inference (VI) that requires the exact expectation terms (analytical form of integral) necessary to update the variational parameters, whereas SVSS uses the stochastic gradient variational bayes (SGVB) (Kingma & Welling, 2013) that uses the estimate of expectations instead of the exact expectations due to the reparametrization (RP) trick.

Due to the use of different lower bound of log marginal likelihood $\log p(Y|X)$, VSS is required to take more variational approximation for random weight A compared to SVSS. This leads that the ELBO of SVSS is closer to $\log p(Y|X)$ than that of VSS; note that $\mathrm{KL}(q(\tilde{s},A)||p(\tilde{s},A)) \geq \mathrm{KL}(q(\tilde{s})||p(\tilde{s}))$. Also, VSS use much more number of the trainable parameters including the variational parameters for $q(A)$ and parameters of the inducing variable, phase, and slope; Further detail are explained in Table 2.

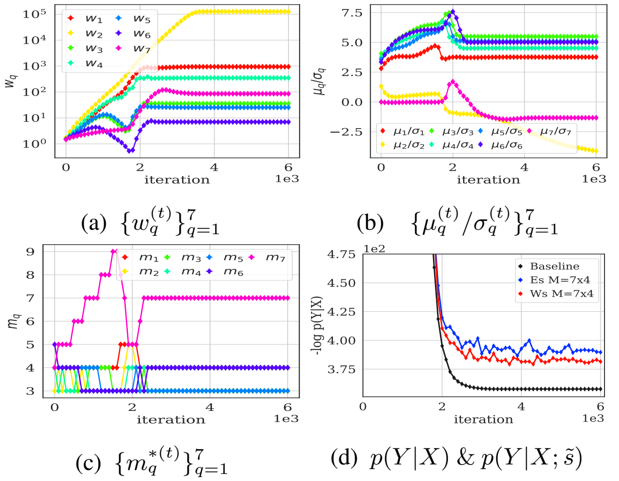
Additional Consideration. Based on the analysis of the ELBO estimator error of SVSS, SVSS can employ the weighted sampling strategy to stabilize the stochastic approximate kernel learning. Also, SVSS can employ the approximate natural gradient to expedite the convergence of parameter inference for the spectral points distribution.

5. Experiments

In this section, we provide the experiments results validating the performances of the proposed model using the various data sets. We provide our implementation at `https://github.com/becre2021/ABInferGSM` and additional results in Appendix D.

5.1. SM Kernel Approximation in Training

Effect of weighted sampling on $\log p(Y|X;\tilde{s})$. We first investigate how the weighted sampling of Proposition 3.3 can reduce the error of the ELBO estimator as analyzed in Proposition 3.2. To this end, we train the SM kernel hyperparameters, $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^7$, with Passenger data set used in (Wilson & Adams, 2013). The conventional log marginal likelihood estimation (MLE-Type 2) is used. We refer to this as Baseline learning method.



(a) $\{w_q^{(t)}\}_{q=1}^7$ (b) $\{\mu_q^{(t)}/\sigma_q^{(t)}\}_{q=1}^7$

(c) $\{m_q^{*(t)}\}_{q=1}^7$ (d) $p(Y|X)$ & $p(Y|X;\tilde{s})$

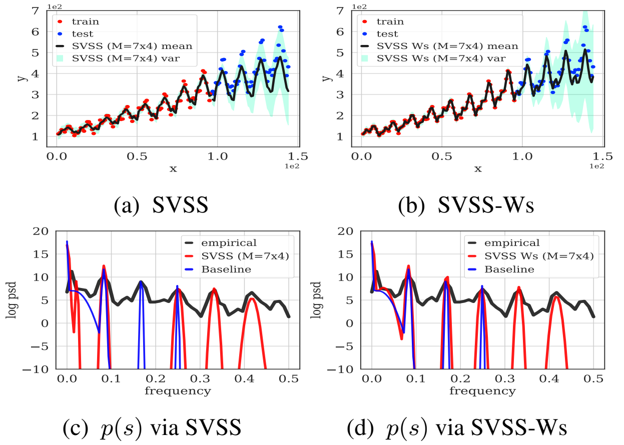
Figure 3. Weighted sampling Effect on $\log p(Y|X;\tilde{s})$. In Fig. 3d, we depict 2 standard deviations error bounds for $\log p(Y|X;\tilde{s})$ (Ws and Es) by filling its color between bounds, respectively.

Figs. 3a and 3b show how the parameters are being optimized by Baseline method over iterations. At each iteration during the training, we allocate the number of spectral points to be sampled from each mixture component using the proposed weighted sampling (Ws) of Proposition 3.3 given a fixed total 28 spectral points, i.e, $M = 28$. As comparing model, we also consider equal sampling (Es) that samples 4 spectral points for each mixture component. Fig. 3c shows the varying number of optimal spectral points sampled from each mixture component by Ws. At each iteration, we also compute $\log p(Y|X;\tilde{s})$ using the approximate kernel matrix $\hat{K}_{\mathrm{SM}}^{(t)}$ constructed by the Ws and Es. Fig. 3d compares the mean and two-standard error of $\log p(Y|X;\tilde{s})$ computed by the two sampling methods (Ws and Es) for 50 samplings at each iterations and its true $\log p(Y|X)$ computed from the exact kernel $K_{\mathrm{SM}}^{(t)}$.

The difference $\log p(Y|X) - \log p(Y|X;\tilde{s})$ in Fig. 3d is the same as the approximation error of Eq. (11). This shows that Ws induces a more stable estimation for $\log p(Y|X)$ than Es during training, and reduces the approximation error as desired from Propositions 3.2 and 3.3.

Effect of weighted sampling on training by SVSS.

We further investigate how the training by SVSS using weighted sampling (SVSS-Ws) helps obtain better SM kernel parameters compared to SVSS (without using weighted sampling) in the same experiment setting. We run a total 10 experiments with the different initialization and compute the test RMSE and MNLL for each prediction experiment on test data set. For prediction, we use the exact kernel K_{SM} with the estimated parameters $\{\hat{w}_q, \hat{\mu}_q, \hat{\sigma}_q^2\}_{q=1}^7$ by SVSS and SVSS-Ws.



(a) SVSS (b) SVSS-Ws

(c) $p(s)$ via SVSS (d) $p(s)$ via SVSS-Ws

Figure 4. Predictive performance for SVSS and SVSS-Ws. Figs. 4a and 4b compare the extrapolation prediction, respectively. Figs. 4c and 4d compare the estimated spectral density of SVSS and SVSS-Ws, respectively.

Among 10 experiments of estimating the hyperparameters, we arbitrarily chose one experiment result (the 6-th experiment result), and depict its prediction results in Fig. 4. For the additional results, we report it in Appendix D.2.

Figs. 4a and 4b show how SVSS (without using weighted sampling) and SVSS-Ws conduct the prediction task, respectively. These figures show that SVSS-Ws predicts more accurately than SVSS. Figs. 4c and 4d show the spectral density $p(s)$ constructed by using the parameters estimated by SVSS and SVSS-Ws, respectively.

In each figure, the estimated spectral density of SVSS and SVSS-Ws is compared to the empirical power spectral density (black) computed directly from the data and the spectral density of the Baseline method (MLE-Type 2) (blue). We observe that the parameters having good prediction results tend to match the spectral modes having the high-energies (it can be checked in Appendix D.2). In Fig. 4d, SVSS-Ws captures the spectral modes more accurately in that SVSS misses the third spectral mode shown in Fig. 4c. We believe that this explains why the Ws could improve SVSS.

Table 1. Comparison of Prediction performance.

	RMSE	MNLL	RMSE (Top 1-5)	MNLL (Top 1-5)
$M = 28\ (Q = 7, m = 4)$				
Baseline (MLE-2)	61.85 ± 22.54	5.66 ± 0.49	26.88 ± 1.29	4.63 ± 0.02
SVSS	94.38 ± 22.30	5.53 ± 0.20	34.11 ± 1.68	4.92 ± 0.04
SVSS-Ws	54.00 ± 11.55	5.62 ± 0.34	28.62 ± 1.12	4.92 ± 0.03
$M = 28\ (Q = 28, m = 1)$				
SVSS	101.04 ± 13.33	5.85 ± 0.15	78.72 ± 6.92	5.72 ± 0.21

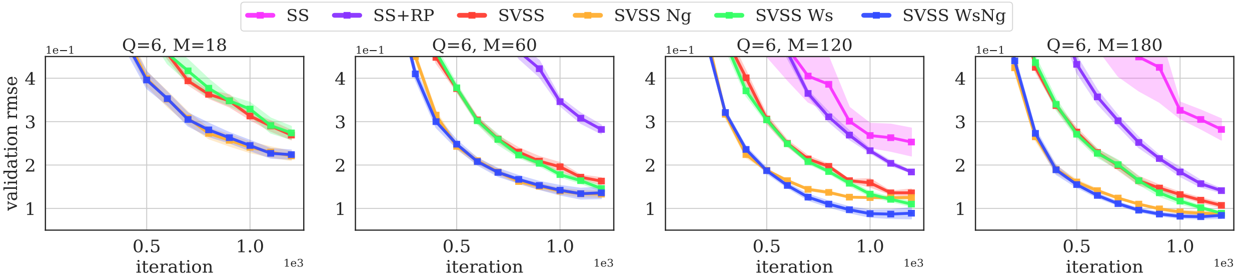
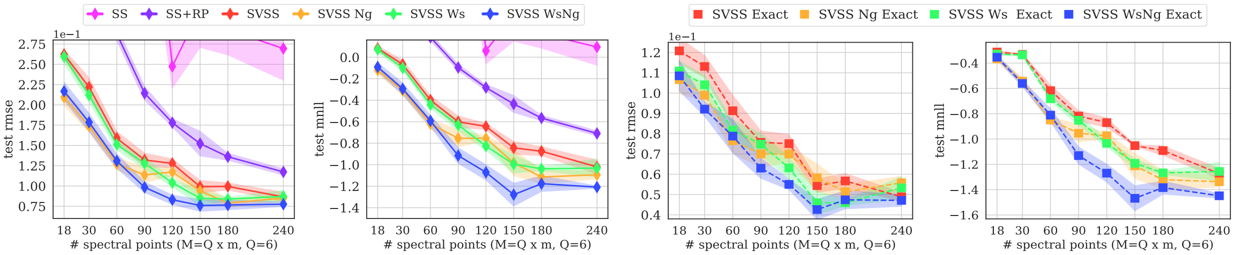


Figure 5. Comparison of training progress over varying M spectral points; the mean and one-standard error of the validation RMSE for 5 experiments are depicted. The predictive distribution is computed as $\frac{1}{J}\sum_{j=1}^{J} p(f^*|x^*, X, Y, \hat{K}_{SM}^{(j)})$ where the predictive mean and variance is averaged over $J = 3$.



(a) prediction by the approximate kernel $\hat{K}_{SM}$

(b) prediction by the exact kernel K_{SM}

Figure 6. Comparison of prediction performance over varying M spectral points; the mean and one-standard error of validation RMSE for 5 experiments are depicted. Predictive distribution using the approximate kernel $\hat{K}_{SM}$ is computed as described in Fig. 5. Predictive distribution of exact K_{SM} is computed as $p(f^*|X^*, X, Y, K_{SM})$ for the predictive inputs X^*, i.e., the conventional prediction method.

Table 1 shows the mean and one-standard error of test RMSE and MNLL computed using 10 experiments and its top-5 experiments because the result of 10 experiments is somewhat volatile due to the model complexity $(Q = 7)$ and its parameter initialization affected by random seeds. Additionally, we consider the SVSS allocating $M = 28$ spectral points to each of $Q = 28$ mixture components (the most flexible model under the constrained computational cost $M = 28$, without the weighted sampling) to validate the necessity of the weighted sampling; note that we fix $M = 28$ to ensure that two models have the computational cost. This table indicates that SVSS-Ws results in a superior prediction than SVSS employing computational cost $M = 28$.

5.2. Ablation Study

In this section, we investigate how the number of the sampled spectral points affects the inference procedure of SVSS using the weighted sampling of Proposition 3.3, and the approximate natural gradient of Proposition 3.4 for learning the SM kernel hyperparameters. We set $Q = 6$ for the mixture components and consider $M \in \{18, 30, 60, 90, 120, 150, 180, 240\}$ spectral points while using the bike dataset $(N{=}17379, D{=}17)$ in UCI benchmark set (Dua & Graff, 2017). We equally divide five partitions of the dataset and randomly select the training, validation, and test set with a ratio of 8:1:1 for each partition. We pick the best kernel parameters with the lowest RMSE on validation set and used them for conducting predictions on test set.

We conduct an ablation study comparing the following inference methods:

- SS: optimizes $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ using the approximate kernel without a reparameterization (RP) trick, which is similar inference with GM in A la Carte (Yang et al., 2015) without fastfood trick (Le et al., 2013) designed for a high-dimension data.
- SS+RP: SS with reparameterization trick (RP).
- SVSS: optimizes $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ via $\hat{\mathcal{L}}_J$ in Eq. (10) with $J = 1$ (i.e. SS+RP+KL using RBF prior).
- SVSS-Ws: SVSS combined with Proposition 3.3.
- SVSS-Ng: SVSS combined with Proposition 3.4.
- SVSS-WsNg: SVSS combined with Propositions 3.3 and 3.4.

For the SVSS-Ws and SVSS-WsNg using the weighted sampling, we randomly select r percentage subset of $X{=}\{x_n\}_{n=1}^{N}$ every iteration to scalably compute a_q^* of Eq. (14) for large N; $r = 1$ means that the entire input X is used. Here, we use $r = .05$.

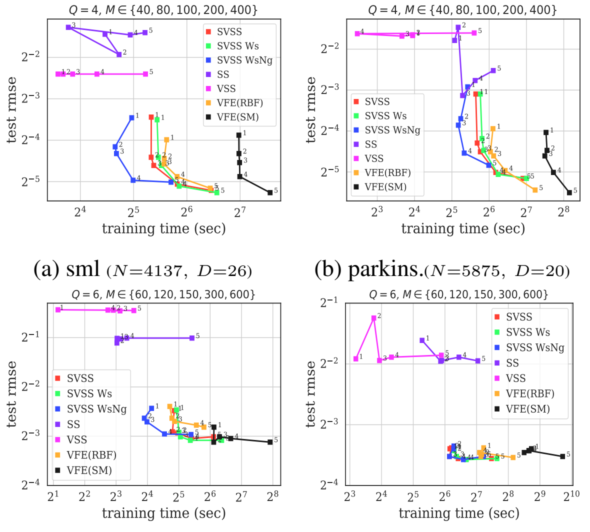
Fig. 5 shows how the validation RMSE decreases as the training proceeds (up to 1200 iterations) for $M \in \{18, 60, 120, 180\}$, thus showing the effect of the number of M samples on the learning trends. The observations found from theses results are summarized as:

- Comparison between SS and SS+RP shows that the effect of RP trick. RP trick reduces the number of the iterations for the parameters to reach the local optimum by adding a random perturbed noise to the sampled spectral points every iterations.
- Comparison between SS+RP and SVSS shows the effect of KL terms using RBF prior. KL term reduces the number of the iterations by introducing the RBF spectral density prior on $q(\tilde{s})$ in Eq. (7).
- Comparison between SVSS and SVSS-Ws shows the effect of weighted sampling. Weighted sampling seems to help find the parameters having the lower validation RMSE (Q=6, M=120, 180).
- Comparison between SVSS and SVSS-Ng or between SVSS-Ws and SVSS-WsNg shows the effect of the natural gradient. It seems that natural gradient leads the fast convergence of parameter inferences consistently, and thus helps to obtain the good parameters earlier.

Fig. 6 compares the prediction performances of all the models under the varying M spectral points. We compare the two types of prediction; the prediction using approximate kernel $\hat{K}_{SM}$ and the the exact kernel K_{SM} with the hyperparameters estimated by the proposed approximate inferences. When a small M is used, SVSS-Ng and SVSS-WsNg obtains the lowest RMSE and MNLL for both the approximate kernel and exact kernel, as shown in Figs. 6a and 6b. As M increases, SVSS-Ws and SVSS-WsNg lead the trained model to have the lower test RMSE and MNLL compared to the SVSS and SVSS-Ng, respectively. This trend confirms that the weighted sampling improves the prediction performance due to the optimal allocation of the spectral points. We check the consistent results for different UCI datasets, and those results are reported in Appendix D.3.

5.3. UCI Dataset Regression Task

We conduct a regression task on the large-scale and high-dimensional datasets that are difficult to be trained using a conventional inference. We add the following approximate inference methods for comparison : (1) VFE: variational inducing variable inference (Titsias, 2009). We use the SM kernel and the RBF (ARD) kernel that is widely used for $\mathcal{GP}$ regression, and (2) VSS: variational sparse spectrum approximation (Gal & Turner, 2015) that uses the classical variational approximation of the spectral points and inducing variables. SM kernel is used. The detailed comparison with baseline methods is described in Appendix B.



(a) sml (N=4137, D=26)

(b) parkins.(N=5875, D=20)

(c) pol*(N=15000, D=26)

(d) keggd.* (N=48827, D=22)

Figure 7. Trade-off between the prediction performance and the training time for different M sampled spectral points. For the datasets denoted by asterisk *, we equally divide 5 partitions of the dataset and obtain the averaged result as the regression task done in (Wilson et al., 2016) due to the memory issue of VFE.

Fig. 7 shows how the number of the sampled spectral points affects the trade-off relationships between the prediction performance and the training time. The y-axis represents the averaged mean of test RMSE, and the x-axis represents the averaged training time taken for obtaining the best parameters on validation set within 1500 iterations. These statistics, marked with a point in each figure, are computed using 5 repeated experiments for all the models. In addition, each point is also marked with the number of spectral points (the smallest M: 1, the largest M: 5). This result shows that SVSS and its variants require an order of magnitude smaller training time to achieve a similar prediction performance to VFE. Additional results are reported in the Appendix D.4.

6. Conclusion

We propose an approximate inference method (SVSS) to train the SM kernel efficiently, and extra strategies to improve SVSS; weighted sampling method for improving stability and the approximate natural gradient for expediting the convergence of the parameter inference.

Acknowledgement

This research was supported by the Korea Agency for Infrastructure Technology Advancement and funded by the Ministry of Land, Infrastructure, and Transport (No. 22PIYR-B153277-04). Specially, we would like to thank Prof. Chulkwang Kwak for the Proofs of Propositions' discussion, Fangying Chen for proofreading, and Prof. Seyoung Yun for the introduction advice and ML algorithm analysis courses.

References

Amari, S.-I. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Chen, K., van Laarhoven, T., and Marchiori, E. Gaussian processes with skewed laplace spectral mixture kernels for long-term forecasting. *Machine Learning*, 110(8): 2213–2238, 2021.

de Wolff, T., Cuevas, A., and Tobar, F. MOGPTK: The Multi-Output Gaussian Process Toolkit. *Neurocomputing*, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.09.085. URL https://github.com/GAMES-UChile/mogptk.

Delbridge, I., Bindel, D., and Wilson, A. G. Randomly projected additive gaussian processes for regression. In *International Conference on Machine Learning*, pp. 2453–2463. PMLR, 2020.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Fang, Y., Loparo, K. A., and Feng, X. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490, 1994.

Gal, Y. and Turner, R. Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *International Conference on Machine Learning*, pp. 655–664, 2015.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 282–290, 2013.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Honkela, A., Tornio, M., Raiko, T., and Karhunen, J. Natural conjugate gradient in variational inference. In *International Conference on Neural Information Processing*, pp. 305–314. Springer, 2007.

Khan, M. and Lin, W. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Artificial Intelligence and Statistics*, pp. 878–887, 2017.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International Conference on Machine Learning*, pp. 2611–2620, 2018.

Khan, M. E. and Nielsen, D. Fast yet simple natural-gradient descent for variational inference in complex models. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pp. 31–35. IEEE, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kostantinos, N. Gaussian mixtures and their applications to signal processing. *Advanced signal processing handbook: theory and implementation for radar, sonar, and medical imaging real time systems*, pp. 3–1, 2000.

Kupilik, M. and Witmer, F. Spatio-temporal violent event prediction using gaussian process regression. *Journal of Computational Social Science*, 1(2):437–451, 2018.

Lazaro-Gredilla, M., Quinonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

Le, Q., Sarlós, T., and Smola, A. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

Lopez-Paz, D., Sra, S., Smola, A., Ghahramani, Z., and Schölkopf, B. Randomized nonlinear component analysis. In *International conference on machine learning*, pp. 1359–1367, 2014.

Parra, G. and Tobar, F. Spectral mixture kernels for multi-output gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 6681–6690, 2017.

Pascanu, R. and Bengio, Y. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Ploysuwan, T. Spectral mixture kernel for pattern discovery and time series forecasting of electricity peak load. In *TENCON 2014-2014 IEEE Region 10 Conference*, pp. 1–5. IEEE, 2014.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rahimi, A. and Recht, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pp. 1313–1320, 2009.

Raj, H., Dey, S., Gupta, H., and Srijith, P. Improving adaptive bayesian optimization with spectral mixture kernel. In *International Conference on Neural Information Processing*, pp. 370–377. Springer, 2020.

Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pp. 63–71. Springer, 2004.

Remes, S., Heinonen, M., and Kaski, S. Non-stationary spectral kernels. In *Advances in neural information processing systems*, pp. 4642–4651, 2017.

Remes, S., Heinonen, M., and Kaski, S. Neural non-stationary spectral kernel. *arXiv preprint arXiv:1811.10978*, 2018.

Salimbeni, H., Eleftheriadis, S., and Hensman, J. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *International Conference on Artificial Intelligence and Statistics*, pp. 689–697, 2018.

Shen, Z., Heinonen, M., and Kaski, S. Harmonizable mixture kernels with variational fourier features. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3273–3282. PMLR, 2019.

Simpson, F., Boukouvalas, A., Cadek, V., Sarkans, E., and Durrande, N. The minecraft kernel: Modelling correlated gaussian processes in the fourier domain. In *International Conference on Artificial Intelligence and Statistics*, pp. 1945–1953. PMLR, 2021a.

Simpson, F., Lalchand, V., and Rasmussen, C. E. Marginalised gaussian processes with nested sampling. *Advances in Neural Information Processing Systems*, 34: 13613–13625, 2021b.

Sutherland, D. J. and Schneider, J. On the error of random fourier features. *arXiv preprint arXiv:1506.02785*, 2015.

Titsias, M. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Tobar, F. Bayesian nonparametric spectral estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 10148–10158, 2018.

Tompkins, A. and Ramos, F. Fourier feature approximations for periodic kernels in time-series modelling. In *AAAI*, pp. 4155–4162, 2018.

Tompkins, A. and Ramos, F. Periodic kernel approximation by index set fourier series features. In *Uncertainty in Artificial Intelligence*, pp. 486–496. PMLR, 2020.

Tropp, J. A. et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Ulrich, K. R., Carlson, D. E., Lian, W., Borg, J. S., Dzirasa, K., and Carin, L. Analysis of brain states from multiregion lfp time-series. In *Advances in Neural Information Processing Systems*, pp. 2483–2491, 2014.

Ulrich, K. R., Carlson, D. E., Dzirasa, K., and Carin, L. Gp kernels for cross-spectrum analysis. In *Advances in neural information processing systems*, pp. 1999–2007, 2015.

Warnes, J. and Ripley, B. Problems with likelihood estimation of covariance functions of spatial gaussian processes. *Biometrika*, 74(3):640–642, 1987.

Wilson, A. and Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pp. 1067–1075, 2013.

Wilson, A. G., Gilboa, E., Cunningham, J. P., and Nehorai, A. Fast kernel learning for multidimensional pattern extrapolation. In *NIPS*, pp. 3626–3634, 2014.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378, 2016.

Wu, J., Poloczek, M., Wilson, A. G., and Frazier, P. Bayesian optimization with gradients. *Advances in neural information processing systems*, 30, 2017.

Xue, H., Wu, Z.-F., and Sun, W.-X. Deep spectral kernel learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4019–4025. AAAI Press, 2019.

Yang, Z., Wilson, A., Smola, A., and Song, L. A la carte–learning fast kernels. In *Artificial Intelligence and Statistics*, pp. 1098–1106, 2015.

Supplementary Material :
Efficient Approximate Inference for Stationary Kernel on Frequency domain

A. Algorithm

A.1. Implementation

We use PyTorch (1.7.0) (Paszke et al., 2019) and employ RTX2080TI-11GB and V100-16GB for GPU. We provide our implementation at `https://github.com/becre2021/ABInferGSM`. We describe the SVSS algorithm combined with Propositions 3.3 and 3.4 as follow:

Algorithm 1 Approximate Inference for the spectral density parameters $\{w_q, \mu_q, \sigma_q\}_{q=1}^{Q}$ and noise parameter σ_ϵ.

```
INPUT: Dataset $D = \{x_n, y_n\}_{n=1}^{N}$, Parameters $\{w_q, \mu_q, \sigma_q\}_{q=1}^{Q}$ and $\sigma_\epsilon$,
    #Parameters update iterations $T$, #Sampled spectral points $M = \sum_{q=1}^{Q} m_q$, #Sampling for ELBO $J$
OUTPUT: Optimal parameters $\{w_q^*, \mu_q^*, \sigma_q^*\}_{q=1}^{Q}$ and $\sigma_\epsilon^*$
for $t = 1 \dots T$ do
    Step 1. Compute the optimal number of spectral points $\{m_q^*\}_{q=1}^{Q}$ satisfying $\sum_{q=1}^{Q} m_q^* = M$ by Proposition 3.3.
    Step 2. Sample spectral points $\widetilde{s}_{(j)} = \bigcup_{q=1}^{Q} \{s_{q,1}, .., s_{q,m_q^*}\}$ for $j = 1, .., J$ such that
        $s_{q,i} = \mu_q + \sigma_q \circ \epsilon_i \quad \epsilon_i \sim N(0, I)$
    Step 3. Compute the approximate natural gradients for updating $\{\log \mu_q, \log \sigma_q\}_{q=1}^{Q}$ by Proposition 3.4
        $\left\{\widetilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J, \widetilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J\right\}_{q=1}^{Q}, \quad \left\{\nabla_{\log w_q} \hat{\mathcal{L}}_J\right\}_{q=1}^{Q}, \text{ and } \quad \nabla_{\log \sigma_\epsilon} \hat{\mathcal{L}}_J$
    Step 4. Update $\{w_q, \mu_q, \sigma_q\}_{q=1}^{Q}$ and $\sigma_\epsilon$ in log domain by ADAM with the estimated gradients
end for
```

A.2. Practical Implementation of Proposition 3.3.

Given the inputs $X = \{x_n\}_{n=1}^{N}$, and kernel hyperparameters $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$, the optimal ratio of spectral points a_q^* in Proposition 3.3 is computed as

$$a_q^* = \frac{w_q \left[\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}}{\sum_{q=1}^{Q} w_q \left[\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}}.$$

However, when applying a_q^* directly during training via SVSS, there are some issue; (1) some dominant weight parameters in training, i.e. $w_q \gg w_1, .., w_{q-1}, w_{q+1}, .., w_Q$, or (2) the sum of variance terms $\sum_{i=1}^{N} \sum_{i<j} g_q(x_i - x_j)$ make $a_q^* \approx 1$ and $a_i^* = 0$ for $i = 1, .., q-1, q+1, .., Q$, which samples all spectral points from single component $N(\mu_q, \sigma_q^2)$, i.e. $m_q = M$.

To resolve these issues, in the implementation, (1) we replace w_q by $\frac{1}{1+\exp{(-w_q/T)}}$ with $T = \text{median}\{w_1, .., w_Q\}$, i.e bounded increasing function w.r.t w_q, and (2) we guarantee the minimal number of each sampled spectral points, i.e. $m_q \geq m_0$ for $q = 1, .., Q$ with the minimal number of spectral points m_0. As a result, these tricks stabilize the ELBO estimator $\hat{\mathcal{L}}_J$ and improve the training by SVSS.

B. Comparison with Baseline Inference methods

We employ the following baseline inference methods to validate the proposed inference SVSS.

- VFE : denotes the variational inducing variables inference method (Titsias, 2009). We refer to the implementation [1]. We not only use the SM kernel, but also the RBF (ARD) kernel because the RBF is one of the most widely used kernels for $\mathcal{GP}$ regression.
- SS: denotes the GM in A la Carte (Yang et al., 2015) without the fastfood trick (Le et al., 2013) designed for a high-dimensional dataset. This inference method optimizes the parameters of SM kernel $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ using the approximate kernel $\hat{K}_{\mathrm{SM}}(X, X)$ without a reparameterization (RP) trick. That is, unlike SVSS, SS samples the Gaussian random noise $\epsilon \sim N(0, I)$ at first iteration (like initialization), and optimizes the parameters of $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ without updating the random noise ϵ during training. Since we could not find the published code, we implement the code by referring to the implementation of Sparse Spectrum $\mathcal{GP}$ (Lazaro-Gredilla et al., 2010), which is designed for the RBF-ARD kernel [2].
- VSS: denotes the variational sparse spectrum approximation of $\mathcal{GP}$ (Gal & Turner, 2015), which is compared in detail Section 4. We refer to the implementation [3]. Notably, VSS introduces the variational distribution of spectral points to relax the overfitting problem described in (Lazaro-Gredilla et al., 2010), and use SM kernel.

Table 2. Time complexity and trainable parameters for the inference methods

Inference	Time Complexity	Trainable Parameters
VFE (RBF)	$\mathcal{O}(NM^2 + M^3)$	kernel $\{\sigma_f^2, l\}$, inducing inputs $\{u_m\}_{m=1}^{M}$, noise σ_ϵ^2
VFE (SM)	$\mathcal{O}(NM^2 + M^3)$	kernel $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$, inducing inputs $\{u_m\}_{m=1}^{M}$, noise σ_ϵ^2
SS	$\mathcal{O}(4NM^2 + 2M^3)$	kernel $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$, noise σ_ϵ^2
VSS	$\mathcal{O}(NM^2 + M^3)$	kernel $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$, inducing inputs $\{u_m\}_{m=1}^{M}$, noise σ_ϵ^2, slope $\{\alpha_0, \beta_0\}$ spectral points $\{\mathbf{m}_m, \mathbf{s}_m\}_{m=1}^{M}$, phases $\{\alpha_m, \beta_m\}_{m=1}^{M}$, random weight $\{m, S\}$
SVSS	$\mathcal{O}(4NM^2 + 2M^3)$	spectral points $\{\mu_q, \sigma_q^2\}_{q=1}^{Q}$ and weights $\{w_q\}_{q=1}^{Q}$, noise σ_ϵ^2

Table 2 describes the time complexity and trainable parameters of the baseline inference methods where N, M, and Q denote the number of data, the number of total spectral points (inducing inputs), and the number of the mixture components for the SM kernel respectively. Although all inference methods have the same time complexity, the single iteration time for each inference method could be different because the number of operations required to update the parameters based on the gradient is proportional to the number of parameters (Rasmussen, 2004).

[1] https://github.com/GPflow/GPflow/blob/develop/gpflow/models/svgp.py

[2] http://www.tsc.uc3m.es/~miguel/downloads.php

[3] https://github.com/yaringal/VSSGP

C. Experimental Details

C.1. Preprocessing of Datasets

Airline Passenger: we employ the raw dataset X and Y as implemented in (Wilson & Adams, 2013) [4] (matlab version).

UCI benchmark dataset: We normalize X by $\frac{X-\mu(X_{\text{train}})}{\sigma(X_{\text{train}})}$ and Y by $\frac{Y-\mu(Y_{\text{train}})}{\sigma(X_{\text{train}})}$ with the mean $\mu(X_{\text{train}})$ and the standard deviation $\sigma(X_{\text{train}})$ for the training inputs X_{train} and the mean $\mu(Y_{\text{train}})$ and the standard deviation $\sigma(Y_{\text{train}})$ for the training outputs Y_{train}. We employ the processing code implemented in (Delbridge et al., 2020) [5].

C.2. Initialization of the variational distribution of spectral points $q(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\})$ of Eq. (7)

Initialization for a low-dimensional dataset ($D = 1, 2$): we refer to the initialization method implemented for (Wilson et al., 2014) [6] and (de Wolff et al., 2020) [7]. Specifically, we first compute the empirical power spectral density (PSD) by applying the conventional FFT tool (scipy.signal.weltch). We then sample spectral points from the obtained empirical normalized PSD, and fit the Gaussian mixture model to the sampled spectral points, which results in the parameters $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$. These parameters are set as the initial parameters for the weights $\{w_q\}_{q=1}^{Q}$ and $\{\mu_q, \sigma_q^2\}_{q=1}^{Q}$ for $q(\tilde{s}) = \prod_{q=1}^{Q}\prod_{i=1}^{m_q} N(s_{q,i};\mu_q,\sigma_q^2)$.

Initialization for a high-dimensional UCI dataset ($D \geq 3$): we employ the random initialization using uniform distribution $U(a, b)$. We refer to the initialization method for ARD and GM that are used in the supplementary material (Yang et al., 2015). Since the initial values for the parameters a, b of Uniform distribution U are not proper to the baseline inference methods (possibly due to the preprocessing of the dataset), we take the following parameterization for U and follow the step described in the supplementary material (Yang et al., 2015).

Thus, we initialize the weight parameters as $w_q = 1$ for all $q = 1,..,Q$, the mean parameters $\mu_q \in R^D \sim U(0, 0.25)$ and the standard deviation parameters $\sigma_q \in R^D \sim U(0.05, 0.5)$ in element-wise sense. We initialize the noise parameters $\sigma_\epsilon = \text{std}(Y_{\text{train}})/20$. Given $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$ and σ_ϵ^2 , we run 100 optimization iterations with these initial candidate parameters to minimize $-\log p(Y_{\text{train}}|X_{\text{train}})$. We repeat this procedure five times, and select the best parameters that have the minimum of $-\log p(Y_{\text{train}}|X_{\text{train}})$ out of the five candidate parameters.

For the RBF (ARD) kernel, we initialize the length scale parameters $l \in R^D \sim U(.5, 1.0)$ in element-wise, and conduct similar procedure by minimizing $-\log p(Y_{\text{train}}|X_{\text{train}})$.

C.3. Initialization of the prior distribution of spectral points $p(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\})$

Before explaining the initialization of prior distribution of spectral points, we first want to explain the KL terms in $\hat{\mathcal{L}}_J$ in terms of the prior distribution of spectral points. For tractability of KL terms, we set the prior distribution $p(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\}) = \prod_{q=1}^{Q}\prod_{i=1}^{m_q} N(s_{q,i};\tilde{\mu}_{q,i},\tilde{\sigma}_{q,i}^2)$ to have same form with $q(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\})$ of Eq. (7) using the parameters $\cup_{q=1}^{Q}\{\tilde{\mu}_{q,i},\tilde{\sigma}_{q,i}^2\}_{i=1}^{m_q}$. We compute the KL term as

$$\text{KL}(q(\tilde{s})||p(\tilde{s}))=\sum_{q=1}^{Q}\text{KL}(N(s_{q,1};\mu_q,\sigma_q^2))||N(s_{q,1};\tilde{\mu}_{q,1},\tilde{\sigma}_{q,1}^2)),$$

by letting $\tilde{\mu}_{q,i} = \mu_q$ and $\tilde{\sigma}_{q,i}^2 = \sigma_q^2$ for $i \geq 2$, which implies $\text{KL}(N(s_{q,i};\mu_q,\sigma_q^2))||N(\tilde{\mu}_{q,i},\tilde{\sigma}_{q,i}^2)) = 0$.

When the maximizing the ELBO estimator $\hat{\mathcal{L}}_J$ of Eq. (10), the KL term in $\hat{\mathcal{L}}_J$ is minimized; each variational distribution of the spectral points $q(s_{q,i}) = N(s_{q,i};\mu_q,\sigma_q^2)$ is pushed to become closer to its prior distribution $p(s_{q,i}) = N(\tilde{\mu}_{q,1},\tilde{\sigma}_{q,1}^2)$ in KL distance sense while it is also pushed to maximize the likelihood term $\frac{1}{J}\sum_{j=1}^{J}\log p(Y|X,\tilde{s}^{(j)})$.

- This acts as the regularizer that prevents the collapsing phenomenon $q(s_{1,i}) = .. = q(s_{Q,i})$ for all i, i.e. single mode for Q mixture components, that might be happen when only maximizing the likelihood term $\frac{1}{J}\sum_{j=1}^{J}\log p(Y|X,\tilde{s}^{(j)})$ in $\hat{\mathcal{L}}_J$.

[4] https://people.orie.cornell.edu/andrew/code/

[5] https://github.com/idelbrid/Randomly-Projected-Additive-GPs/

[6] https://people.orie.cornell.edu/andrew/pattern/#Kronecker

[7] https://github.com/GAMES-UChile/mogptk/

- This could help the parameter inference of $q(s_{q,i}) = N(s_{q,i}; \mu_q, \sigma_q^2)$ by letting learnable parameters $\{\mu_q, \sigma_q^2\}_{q=1}^{Q}$ exploring the optimal parameter around the prior parameters $\{\tilde{\mu}_{q,1}, \tilde{\sigma}_{q,1}^2\}_{q=1}^{Q}$ during training.

Using the described meaning of the prior distribution $p(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\})$ in optimizing $\hat{\mathcal{L}}_J$, we initialize the parameters of the prior distribution as follows:

Initialization for a low-dimensional dataset ($D = 1, 2$): In this case, since the initial parameters of $q(\cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\})$ can be obtained by using the empirical spectral density, which can be used as good prior, we set the parameters of prior distribution $p(s_{q,i}) = N(\tilde{\mu}_{q,1}, \tilde{\sigma}_{q,1}^2)$ by adding the small random noise to the initial parameters of μ and scaling down σ such as 0.1σ obtained from the initialization in Appendix C.2; scaling down is conducted because finding the frequency region that having the condensed high spectral energy is important for training the SM kernel hyperparameters as shown in Appendix D.2.

Initialization for a high-dimensional UCI dataset ($D \geq 3$): In this case, we assign the small values for $\tilde{\mu}_{q,1}$ by $\tilde{\mu}_{q,1} \sim U(0, 0.05)$, $\tilde{\sigma}_{q,1} \sim U(0, 0.05)$. Assigning the small values for $\tilde{\mu}_{q,1}$ reflects our prior knowledge that the RBF kernel (the case of $\mu_q = 0$ for all q) could be suitable for describing the UCI dataset, and the optimized parameters of the SM kernel are expected to be, at least, better model parameters than the parameters of the RBF kernel.

C.4. Training Details

SM Kernel Approximation in Training in Section 5.1: For the baseline learning method (maximization of log marginal likelihood known as MLE-Type 2), SVSS, and SVSS-Ws, we use the Adam optimizer (Kingma & Ba, 2014) with the learning rate $lr = .005$. For the sampling rate r to compute $a^* = [a_1^*, .., a_Q^*]$ in Proposition 3.3 efficiently, we use $r = .5$ for SVSS-Ws. For the minimal spectral points, we set $m_q \geq \max\{\frac{M}{10Q}, 1\}$.

Ablation Study in Section 5.2: For all inference methods, we use the Adam optimizer with $lr = .005$. We run 1000 and 1200 iterations for training the Parkinsons and the Bike dataset, respectively. For the sampling rate, we use the $r = .05$. For the minimal spectral points, we set $m_q \geq \max\{\frac{M}{10Q}, 1\}$.

UCI Regression Experiment in Section 5.3: For all inference methods, we use the Adam optimizer with $lr \in \{.01, .005\}$ and report the better result in sense of RMSE. We run 1500 iterations for training. For the sampling rate, we use the $r = .05$. For the minimal spectral points, we set $m_q \geq \max\{\frac{M}{10Q}, 1\}$.

D. Additional Experiment Results

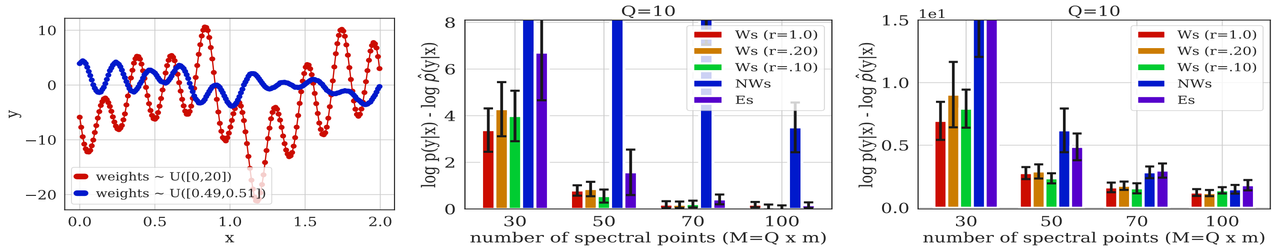
D.1. Validation of the Claims for Propositions 3.1 to 3.3.

Research questions. In this experiment, we verify the following claims:

- The error of the ELBO estimator $\log p(Y|X) - \frac{1}{J}\sum_{j=1}^{J}\log \hat{p}(Y|X, \tilde{s}^j)$ in Eq. (11) is reduced as the number of the sampled spectral points M increases as claimed in Propositions 3.1 and 3.2.
- The weighted sampling of Proposition 3.3 reduces the error of ELBO estimator as claimed in Propositions 3.2 and 3.3.
- Using the subset of dataset X for computing a_q^* in Proposition 3.3 can reduce the error of ELBO estimator stably. This enables us to compute a_q^* for the large N dataset because evaluating a_q^* takes $O(N^2)$.

Experimental settings. We consider $X = \{0, .01, .., 1.99\}$ as inputs ($N = 200$), and generate $Y = f(X) + \epsilon$ with $\epsilon \sim N(0, .05^2)$ as the sample of random function f following $\mathcal{GP}$ prior using SM kernel ($Q = 10$). The weights of SM kernel hyperparameter are initialized in two ways: case 1) $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$ and case 2) $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$. Other parameters are initialized as $\{\mu_q\}_{q=1}^{Q} \sim U(0, 5)$ and $\{\sigma_q\}_{q=1}^{Q} \sim U(0, .05)$. These initializations are considered to demonstrate that the proposed sampling can reduce the error of ELBO estimator well in both cases where the target signal Y is modeled by the f using SM kernel having the extremely different values of the weights $\{w_q\}_{q=1}^{Q}$ and similar values of the weights $\{w_q\}_{q=1}^{Q}$.

To verify the effectiveness of the weighted sampling (Ws), we consider the following naive approaches: (1) equal sampling (Es) by $a_q = 1/Q$ and (2) naive weighted sampling (NWs) by $a_q = w_q / \sum_{q=1}^{Q} w_q$. For the Ws, we consider the sampling rate $r \in \{0.1, 0.2, 1.0\}$ to sample the fractions of X. For example, $r = 1.0$ denotes full data X is used for computing a_q^*.



(a) two signals (b) $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$ (c) $\{w_q\}_{q=1}^{Q} \sim U(.49, .51)$

Figure 8. Comparison of the error of ELBO estimator $\log p(Y|X) - \log \hat{p}(Y|X, \tilde{s}^{(1)})$ over varying sampling rate $r \in \{0.1, 0.2, 1.0\}$; figure (a) denotes the X and Y sampled by the SM kernel using (1) $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$ (red) and (2) $\{w_q\}_{q=1}^{Q} \sim U(.49, .51)$ (blue). Figure (b) and (c) denote the error of corresponding ELBO estimator for $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$ and $\{w_q\}_{q=1}^{Q} \sim U(.49, .51)$ respectively (Ws: weighted sampling of Proposition 3.3, Es: equal sampling, NWs: naive weighted sampling).

Experimental results. Fig. 8 shows the sampled function from GP prior function f that uses the described two SM kernel hyperparameter setting and its error of ELBO estimator $\log p(Y|X) - \frac{1}{J}\sum_{j=1}^{J}\log p(Y|X, \tilde{s}^{(j)})$ with $J = 1$ in Eq. (11). We consider the number of total spectral points $M \in \{30, 70, 50, 100\}$ with $Q = 10$, and depict the mean and two-standard error of the ELBO estimator error for 500 experiments. Figs. 8b and 8c show the results of case 1) $\{w_q\}_{q=1}^{Q} \sim U(0, 20)$ and case 2) $\{w_q\}_{q=1}^{Q} \sim U(0.49, 0.51)$, respectively.

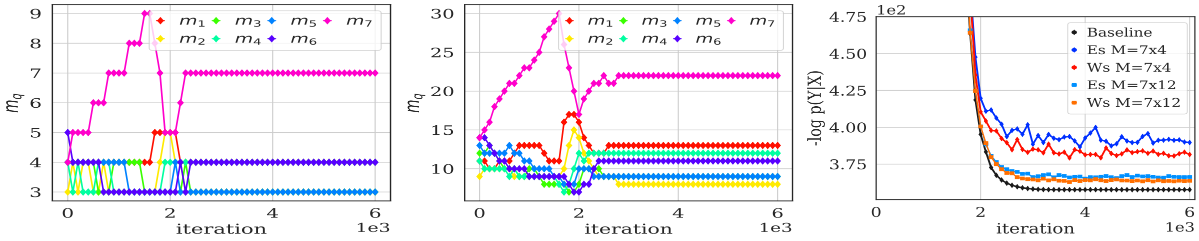
In Figs. 8b and 8c, the weighted sampling (Ws) obtains the smaller error of ELBO estimator than Es and NWs over $M \in \{30, 70, 50, 100\}$. We obtain the following conclusion from this experiment:

- In both Figs. 8b and 8c, as the number of spectral points M increases, the ELBO estimator error for Ws and Es is reduced, which is consistent with what Propositions 3.1 and 3.2 imply.
- In Figs. 8b and 8c, obtaining the smaller error by Ws, compared to Es, implies that the weighted sampling of Proposition 3.3, that is designed to reduce the approximation gap of the ELBO estimator $\hat{\mathcal{L}}_J$ based on Proposition 3.2, enables the ELBO estimator $\hat{\mathcal{L}}_J$ to be tight as desired.
- In Figs. 8b and 8c, Ws obtains the similar level of the ELBO estimator error in regardless of the sampling rate $r \in \{0.1, 0.2, 1.0\}$, which implies the weighted sampling is quite stable.

- In Fig. 8c, the effectiveness of employing the other parameters $\{\mu_q^2, \sigma_q^2\}_{q=1}^{Q}$ and dataset X for the weighed sampling of Proposition 3.3 is revealed because the dataset in Fig. 8c is generated by the weight $\{w_q\}_{q=1}^{Q} \sim U(.49, .51)$.

D.2. Additional Results of Section 5.1

Additional results: Effect of weighted sampling. We report the additional result for the experiment as shown in Section 5.1. Figs. 9a and 9b denote the result of the weighted sampling with $M = 7 \times 4$ and $M = 7 \times 12$, respectively. Fig. 9c denotes the true log marginal likelihood $\log p(Y|X)$ and its estimator $\log p(Y|X, \tilde{s})$ for $M = 7 \times 4$ and $M = 7 \times 12$. We can confirm additionally that as the the sampled spectral points M increases, the estimator $\log p(Y|X, \tilde{s})$ becomes closer to the $\log p(Y|X)$. $\log p(Y|X, \tilde{s})$ of the weighted sampling (Ws) is still closer to true $\log p(Y|X)$ compared to thaf of the equal sampling (Es).



(a) $\{m_q^{*(t)}\}_{q=1}^{7}$ ($M = 7 \times 4$)

(b) $\{m_q^{*(t)}\}_{q=1}^{7}$ ($M = 7 \times 12$)

(c) $p(Y|X)$ & $p(Y|X; \tilde{s})$

Figure 9. Extension results of Fig. 3; Figs. 9a and 9b denote the result of the weighted sampling with $M = 7 \times 4$ and $M = 7 \times 12$, respectively. Fig. 9c denotes the true log marginal likelihood $\log p(Y|X)$ and its estimator $\log p(Y|X, \tilde{s})$.

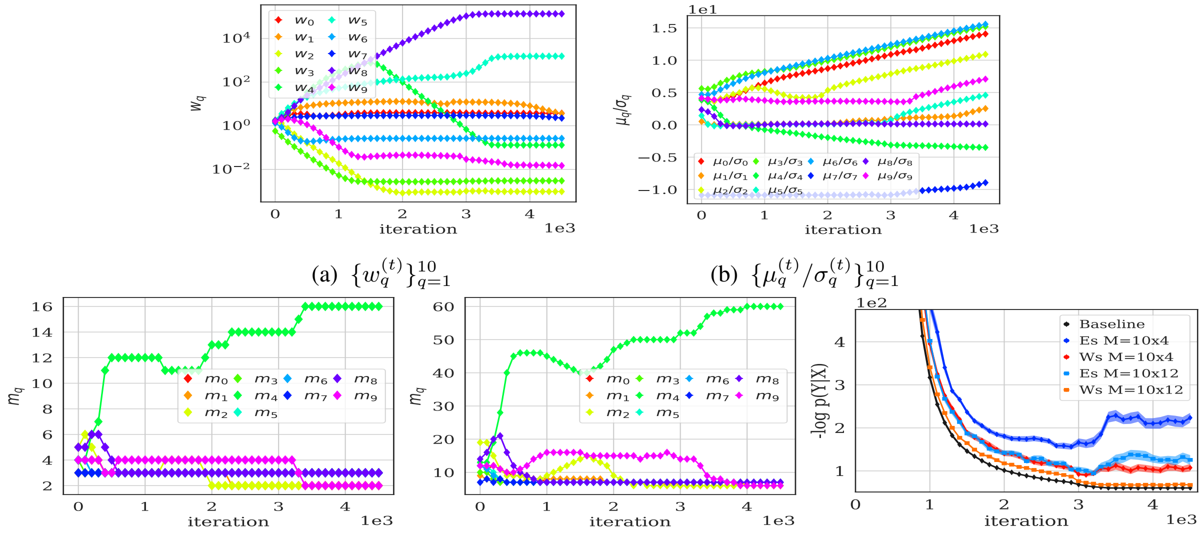
Additional results: Effect of weighted sampling on training by SVSS. Since 10 experiment result of SVSS-Ws and SVSS reported in Table 1, seems somewhat volatile because of the model complexity ($Q = 7$) and its parameter initialization affected by random seeds, we raise the model complexity of SM kernel $Q = 10$ from $Q = 7$, and conduct the experiment under the same setting except model complexity $Q = 10$. Table 3 shows that the result of SVSS-Ws and SVSS becomes stable for 10 experiment result. The training by SVSS-Ws results in more accurate prediction than that of SVSS.

Table 3. Comparison of Prediction performance.

	RMSE	MNLL	RMSE (Top 1-5)	MNLL (Top 1-5)
$Q = 10$				
Baseline (MLE-2)	29.61 ± 0.60	4.79 ± 0.03	28.44 ± 1.78	4.78 ± 0.12
$M = 40$ ($Q = 10, m = 4$)				
SVSS	50.56 ± 9.75	5.08 ± 0.10	35.25 ± 2.06	4.92 ± 0.02
SVSS-Ws	44.58 ± 9.86	5.04 ± 0.09	29.64 ± 2.17	4.90 ± 0.02
$M = 120$ ($Q = 10, m = 12$)				
SVSS	32.62 ± 1.20	4.77 ± 0.03	30.63 ± 1.41	4.72 ± 0.02
SVSS-Ws	33.56 ± 1.34	4.76 ± 0.03	30.76 ± 0.50	4.69 ± 0.01

Additional results: Other Dataset for Section 5.1. We train the SM kernel hyperparameters ($Q = 10$) with CO2 dataset used for SM kernel work (Wilson & Adams, 2013) under the same experiment setting in Section 5.1 except model complexity $Q = 10$.

Figs. 10a and 10b denote the varying parameters of SM kernel ($Q = 10$) ordained by baseline training method (MLE-Type 2). Figs. 10c and 10d denote the optimal allocation by the weighted sampling for $M = 10 \times 4$ and $M = 10 \times 12$, respectively. Fig. 10e denotes the true log marginal likelihood $\log p(Y|X)$ and its estimator $\log p(Y|X, \tilde{s})$ for $M = 10 \times 4$ and $M = 10 \times 12$. We can see that weighted sampling of Proposition 3.3 reduces the error of $\log p(Y|X, \tilde{s})$ as claimed through Propositions 3.2 and 3.3.



(a) $\{w_q^{(t)}\}_{q=1}^{10}$

(b) $\{\mu_q^{(t)}/\sigma_q^{(t)}\}_{q=1}^{10}$

(c) $\{m_q^{*(t)}\}_{q=1}^{10}$ $(M = 10 \times 4)$

(d) $\{m_q^{*(t)}\}_{q=1}^{10}$ $(M = 10 \times 12)$

(e) $p(Y|X)$ & $p(Y|X; \tilde{s})$

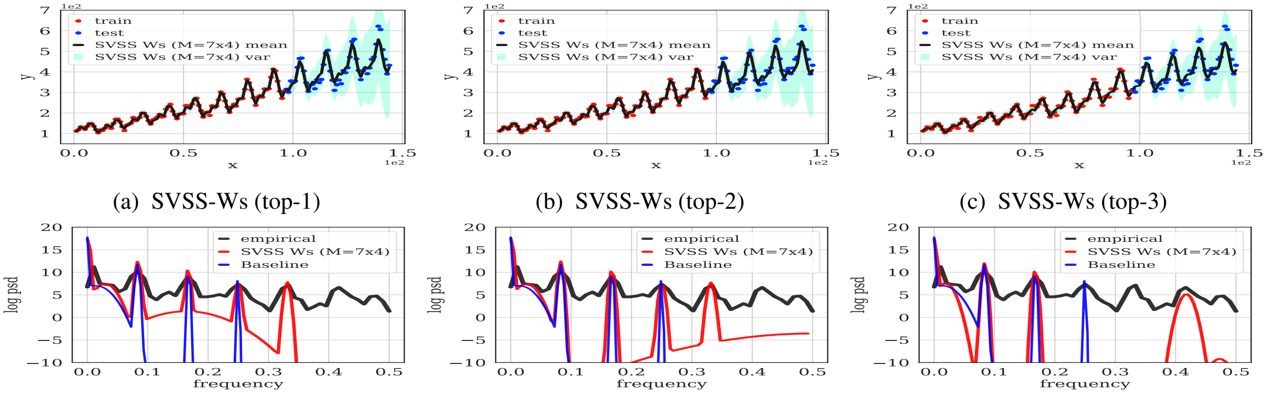
Figure 10. Effect of the weighted sampling on $\log p(Y|X; \tilde{s})$ on CO2 dataset; Figs. 10a and 10b denote the result of the weighted sampling with $M \in \{10 \times 4, 10 \times 12\}$, respectively. Figs. 10c and 10d show the optimal allocation of spectral points with $M \in \{10 \times 4, 10 \times 12\}$ during training, respectively. Fig. 10e denotes the true log marginal likelihood $\log p(Y|X)$ and its estimator $\log p(Y|X, \tilde{s})$.

Appendix D.2 reports the corresponding prediction results on CO2 dataset, obtained by SVSS and SVSS-Ws.

Table 4. Comparison of Prediction performance.

	RMSE	MNLL	RMSE (Top 1-5)	MNLL (Top 1-5)
$Q = 10$				
Baseline (MLE-2)	1.24 ± 0.07	1.64 ± 0.03	1.11 ± 0.01	1.58 ± 0.01
$M = 40$ $(Q = 10, m = 4)$				
SVSS	5.65 ± 1.21	2.95 ± 0.10	3.22 ± 0.42	2.78 ± 0.10
SVSS-Ws	4.34 ± 0.61	2.86 ± 0.07	2.62 ± 0.28	2.81 ± 0.13

Additional prediction results: Effect of weighted sampling on training by SVSS. We report the additional results described in Section 5.1. Figs. 11 and 12 show the top 1-3 prediction results and its spectral densities of the SVSS-Ws and SVSS (without weighted sampling), respectively; we pick the top 3 estimated parameters that have led the most accurate prediction (extrapolation) out of 10 experiments. Upon the closer analysis of experimental results with the accurate predictions (top 3 estimated parameters), we found that the spectral densities composed of the estimated parameters were in good agreement with the empirical power spectral density constructed directly from the data. In particular, the constructed spectral densities tend to precisely match the spectral modes having the high-energies (low frequency area) as shown in Figs. 11d to 11f and Figure Figs. 12d to 12f. Thus, the results of sixth-parameters described in Figs. 4c and 4d, where SVSS misses the third spectral mode, explains why the weighted sampling could improve the SVSS.



(a) SVSS-Ws (top-1)

(b) SVSS-Ws (top-2)

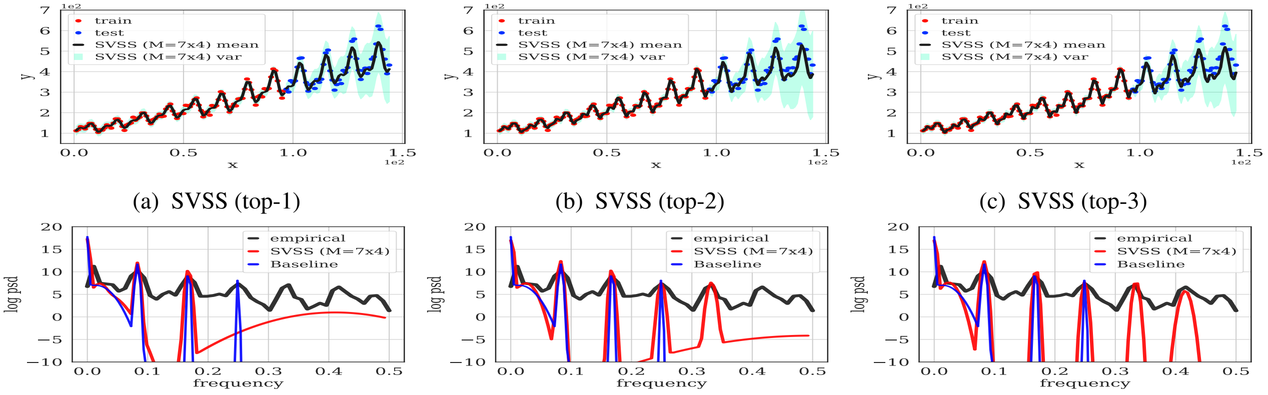
(c) SVSS-Ws (top-3)

(d) $p(s)$ via SVSS-Ws (top-1)

(e) $p(s)$ via SVSS-Ws (top-2)

(f) $p(s)$ via SVSS-Ws (top-3)

Figure 11. Predictive performance of SVSS-Ws with the chosen parameters. Figs. 11a to 11c describe the extrapolation results with the top 3 estimated parameters out of 10 experiments, top-1: $(\mathrm{rmse}: 25.74, \mathrm{mnll}: 4.92)$, top-2: $(\mathrm{rmse}: 27.30, \mathrm{mnll}: 4.89)$, top-3: $(\mathrm{rmse}: 28.58, \mathrm{mnll}: 4.85)$. Figs. 11d to 11f denote the corresponding spectral densities of SVSS-Ws. Additionally, we depict the empirical power spectral density and the spectral density using the best parameter obtained by Baseline learning methods (MLE-Type2).



(a) SVSS (top-1)

(b) SVSS (top-2)

(c) SVSS (top-3)

(d) $p(s)$ via SVSS (top-1)

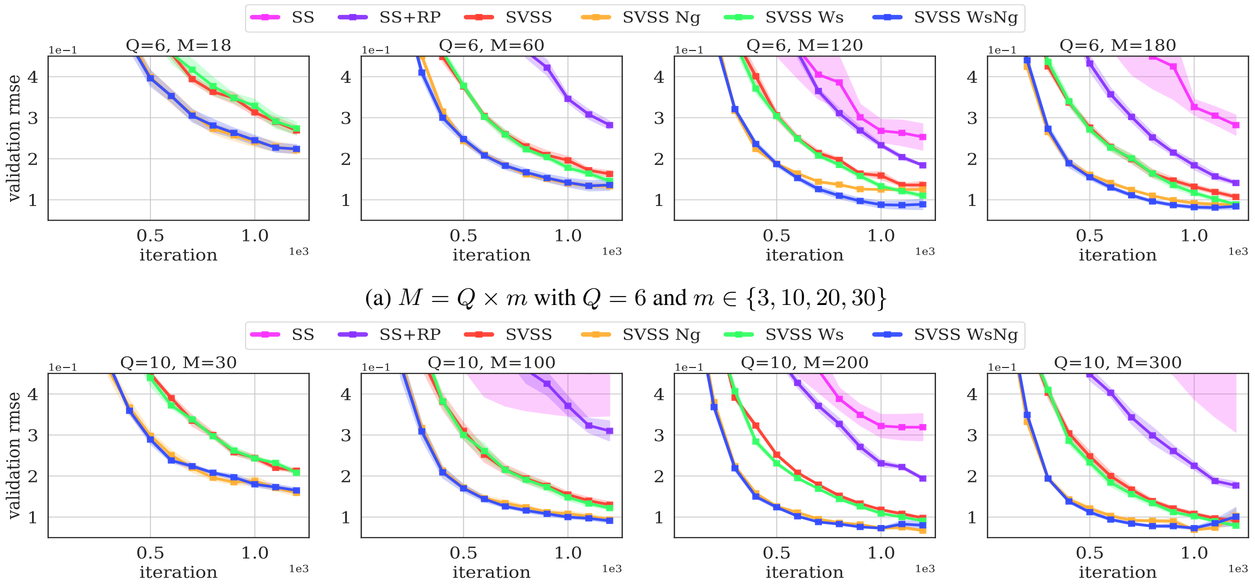
(e) $p(s)$ via SVSS (top-2)

(f) $p(s)$ via SVSS (top-3)

Figure 12. Predictive performance of SVSS with the chosen parameters. Figs. 12a to 12c describe the extrapolation results with the top 3 estimated parameters out of 10 experiments, top-1: $(\mathrm{rmse}: 28.06, \mathrm{mnll}: 4.82)$, top-2: $(\mathrm{rmse}: 33.90, \mathrm{mnll}: 4.90)$, top-3: $(\mathrm{rmse}: 38.87, \mathrm{mnll}: 4.91)$. Figs. 12d to 12f denote the corresponding spectral densities of SVSS. We depict the same empirical power spectral density and the spectral density as shown in Fig. 11 for comparison.

D.3. Additional Results of Section 5.2

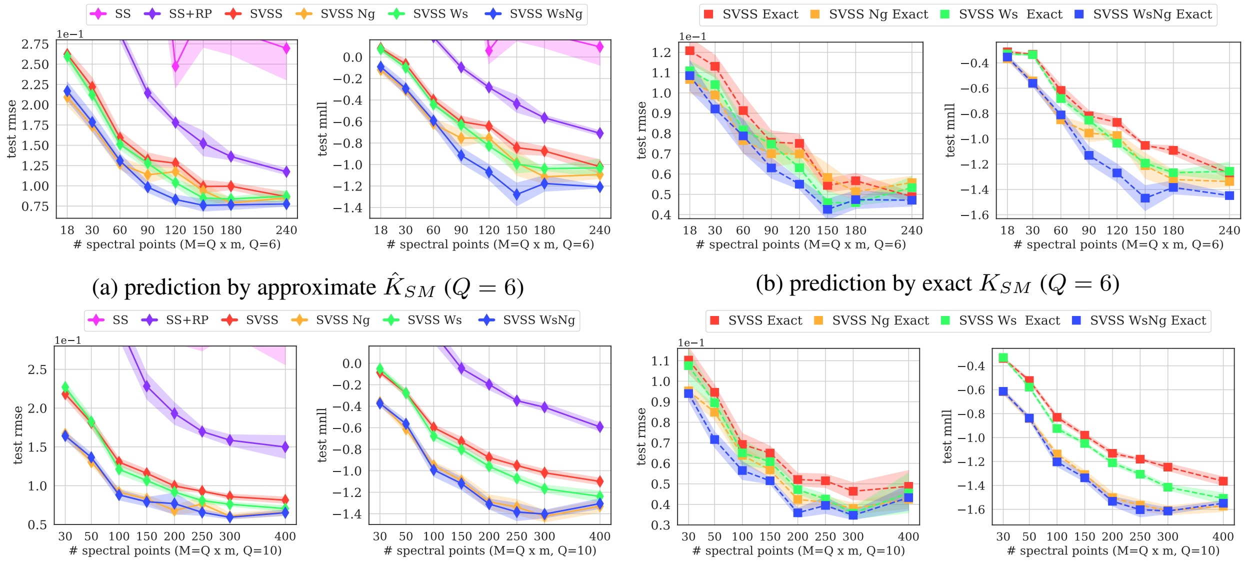
Bike $(N{=}17379, D{=}17)$ dataset. We conduct the ablation study to investigate how the inference procedure of SVSS can be improved. We change the number of M spectral points to investigate how M affects the inference procedure. We use the Bike $(N{=}17379, D{=}17)$ dataset. For the weighted sampling, we randomly select the 5 percent of $X = \{x_n\}_{n=1}^{N}$ every iteration, i.e. $r = .05$, to compute a_q^* of Proposition 3.3 scalably. For the experiment procedure to report the statics results, we follow the same procedure as described in Section 5.2.



(a) $M = Q \times m$ with $Q = 6$ and $m \in \{3, 10, 20, 30\}$

(b) $M = Q \times m$ with $Q = 10$ and $m \in \{3, 10, 20, 30\}$

Figure 13. Comparison of training progress over varying M spectral points and $Q \in \{6, 10\}$.



(a) prediction by approximate $\hat{K}_{SM}$ $(Q = 6)$

(b) prediction by exact K_{SM} $(Q = 6)$

(c) prediction by approximate $\hat{K}_{SM}$ $(Q = 10)$

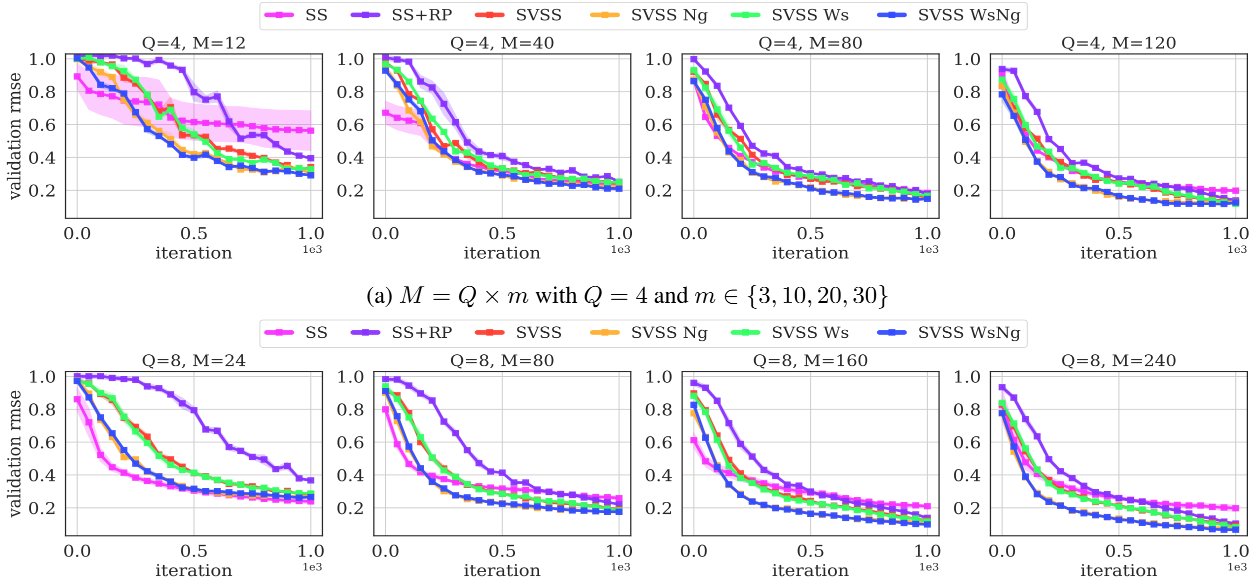
(d) prediction by exact K_{SM} $(Q = 10)$

Figure 14. Comparison of prediction over varying M spectral points and fixed $Q \in \{6, 10\}$.

Experimental results. Fig. 13 shows the incremental improvements by RP tricks, RBF spectral density prior, weighted sampling of Proposition 3.3, and the natural gradient of Proposition 3.4 as described in Section 5.2.

Fig. 14 shows that as M increases, SVSS-Ws and SVSS-WsNg lead the trained model to have the lower test RMSE and MNLL compared to the SVSS and SVSS-Ng, respectively.

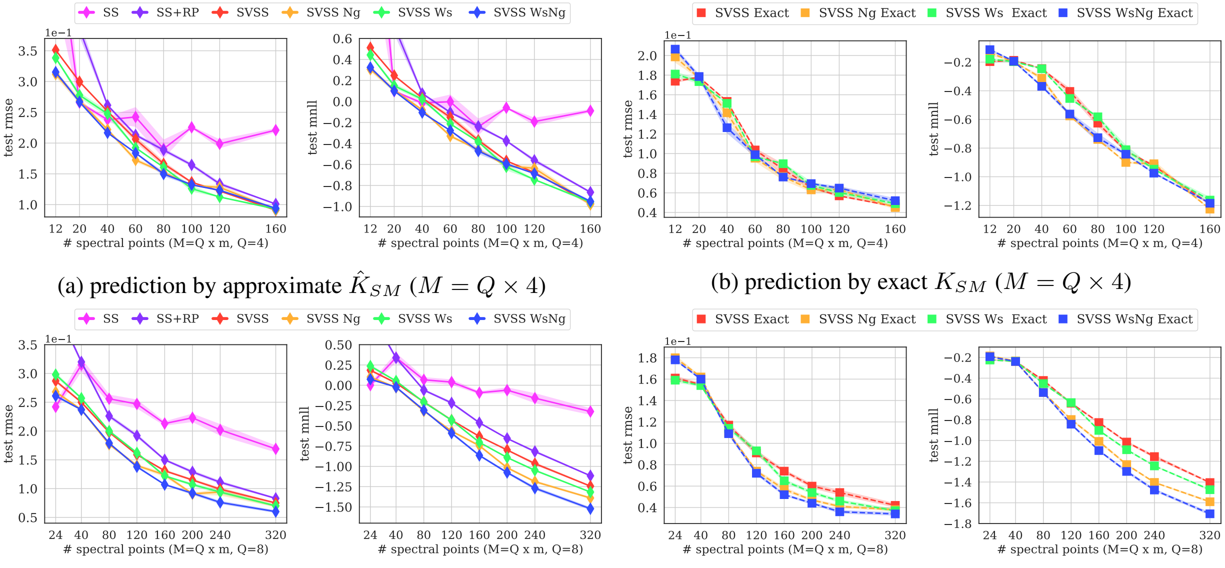
Parkinsons (N=5875, D=20) dataset. We conduct the similar experiment with Parkinsons (N=5875, D=20) dataset. For the efficient computation of the weighted sampling on large-scale dataset, we use $r = .05$.



(a) $M = Q \times m$ with $Q = 4$ and $m \in \{3, 10, 20, 30\}$

(b) $M = Q \times m$ with $Q = 8$ and $m \in \{3, 10, 20, 30\}$

Figure 15. Comparison of training progress over varying M spectral points and $Q \in \{4, 8\}$.



(a) prediction by approximate $\hat{K}_{SM}$ ($M = Q \times 4$)

(b) prediction by exact K_{SM} ($M = Q \times 4$)

(c) prediction by approximate $\hat{K}_{SM}$ ($M = Q \times 8$)

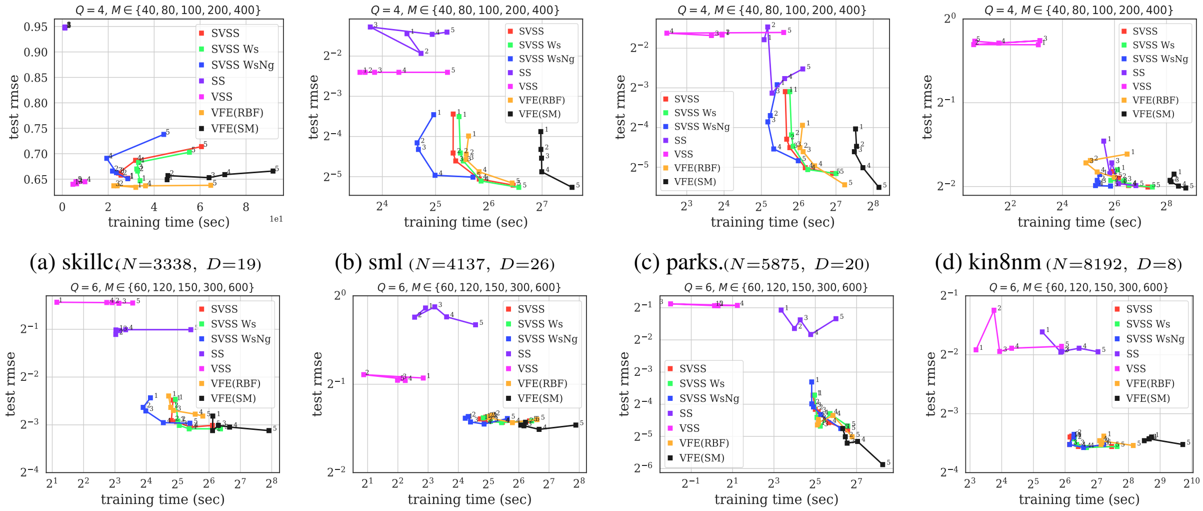
(d) prediction by exact K_{SM} ($M = Q \times 8$)

Figure 16. Comparison of prediction over varying M spectral points and fixed $Q \in \{4, 8\}$.

Experimental results. Figs. 15 and 16 show the similar tendency described as shown in Figs. 13 and 14, respectively.

D.4. Additional Results of Section 5.3

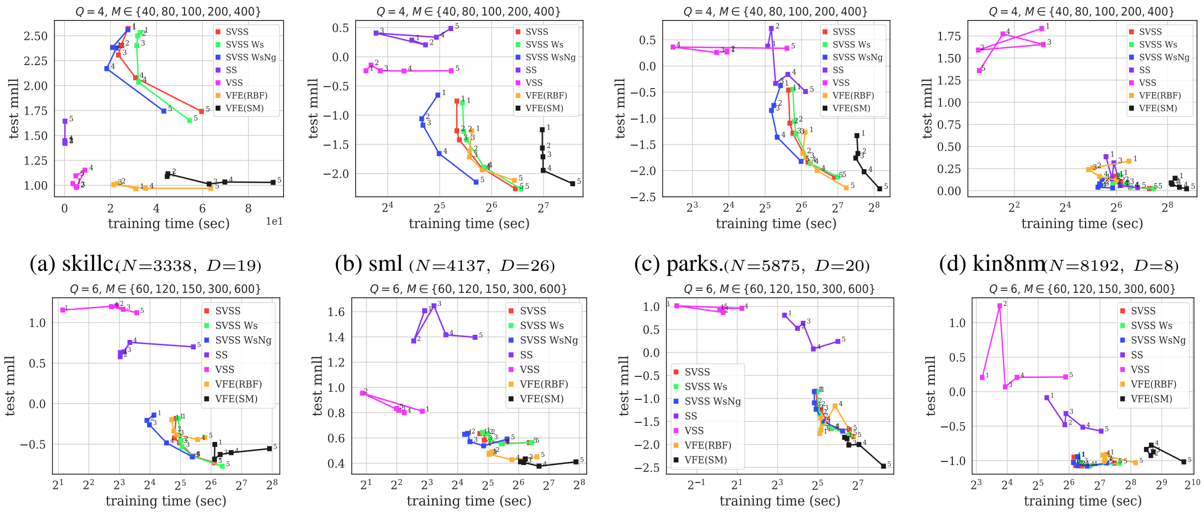
Experimental settings. For the large-scale UCI Dataset Regression Task, we conduct 5 experiments, and for each experiment, randomly select the training, validation, and test sets with a ratio of 8:1:1 using a different random seed. We pick the best kernel parameters that have the lowest validation RMSE obtained by the prediction using the approximate kernel $\hat{K}_{SM}$, and use the chosen parameters to predict the test set with the exact kernel K_{SM}. We set $Q = 4$ and $Q = 6$ (data sets denoted by asterisk *) for SM kernel. For the SVSS-Ws and SVSS-WsNg, we use the sampling rate $r = .05$ for the weighted sampling. For the VFE, we use $2M$ inducing variables to use the same size of Gram matrix with SVSS.



(a) skillc(N=3338, D=19) (b) sml (N=4137, D=26) (c) parks.(N=5875, D=20) (d) kin8nm (N=8192, D=8)

(e) pol* (N=15000, D=26) (f) elev.*(N=16599, D=18) (g) bike* (N=17379, D=17) (h) keggd.*(N=48827, D=22)

Figure 17. Test RMSE vs Training time taken for obtaining the estimated parameters on validation set (within 1500 iterations) ; for the datasets denoted by asterisk *, we equally divide 5 partitions of the dataset and obtain the averaged result as the regression task done in (Wilson et al., 2016) due to the memory issue of VFE.

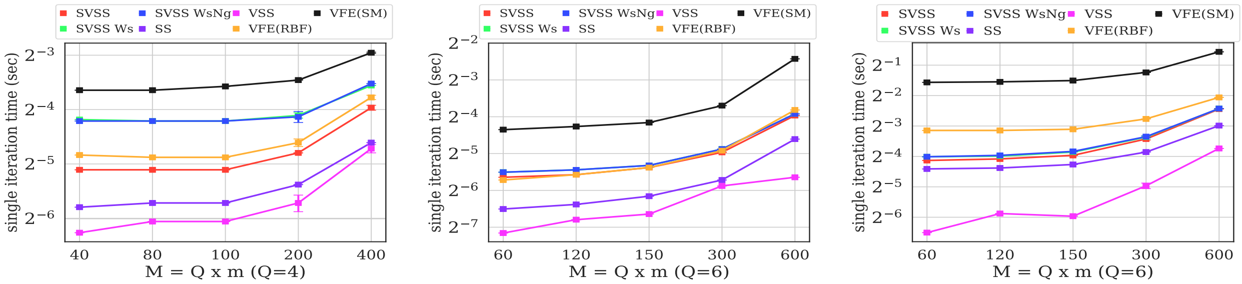


(a) skillc(N=3338, D=19) (b) sml (N=4137, D=26) (c) parks.(N=5875, D=20) (d) kin8nm(N=8192, D=8)

(e) pol* (N=15000, D=26) (f) elev.*(N=16599, D=18) (g) bike* (N=17379, D=17) (h) keggd.*(N=48827, D=22)

Figure 18. Test MNLL vs Training time taken for obtaining the estimated parameters on validation set (within 1500 iterations) ; these results are obtained from the same experiment as shown in Fig. 17.

Experimental results. Fig. 17 compares the trade-off relationships between the test RMSE and the training time for different M spectral points. For the datasets on the first rows (skillcraft, sml, parkinsons, and kin8nm), $Q = 4$ and $M \in \{40, 80, 100, 200, 400\}$ are used. For the second rows (pol, elevators, bike, and keggdirected), $Q = 6$ and $M \in$



(a) sml (N=4137, D=26) (b) pol* (N=15000, D=26) (c) keggd.*(N=48827, D=22)

Figure 19. Single iteration time (second) for training over M spectral points; we report the single iteration time measured for the experiments as shown in Figs. 17 and 18.

$\{60, 120, 150, 300, 600\}$ are used. In each figure, the y-axis represents the averaged mean of test RMSE, and the x-axis represents the averaged training time taken for obtaining the best parameters on validation set within 1500 iterations. These statistics, marked with a point in each figure, are computed using 5 repeated experiments for all the models. In addition, each point is also marked with the magnitude of spectral points (the smallest M: 1, the largest M: 5). Fig. 18 compares the trade-off between the test MNLL and the training time obtained from the same experiments as shown in Fig. 17, where y-axis represents the averaged mean of test MNLL.

Figs. 17 and 18 show that SVSS and its variants require an order of magnitude smaller training time to achieve a similar level of the prediction performance to that of the VFE (SM) for the most of the datasets. Note that training time of SVSS-WsNg is generally smaller than that of SVSS and SVSS-Ws due to the fast convergence of parameter inference by the natural gradient.

Fig. 19 compares the single iteration time (seconds) for the experiment as shown in Fig. 18. Generally, SVSS and its variants take less single iteration time compared to that of VFE (SM).

Tables 5 and 6 report the mean and one standard of the RMSE and MNLL for the prediction on the test set, respectively; We report the fractions of the results for Q=4, $M \in \{100, 400\}$ (wine, skillcraft, sml, parkinsons, and kin8nm) and Q=6, $M \in \{150, 600\}$ (pol, elevators, bike, and keggdirected), to present the specific statistics. In these tables, SVSS denotes the prediction using $\hat{K}_{SM}$ with the parameters estimated by SVSS. The Exact, Exact-W, Exact-WN denote the prediction using the exact K_{SM} with the parameters estimated by SVSS, SVSS-Ws, and SVSS-WsNg, respectively.

We confirm the following results:

- Comparison between SVSS and Exact, Exact-W, and Exact-WN shows that the prediction using K_{SM} with the parameters $\{\hat{w}_q, \hat{\mu}_q, \hat{\sigma}_q^2\}_{q=1}^{Q}$ and $\hat{\sigma}_\epsilon^2$ estimated by SVSS and its variants have led to more accurate prediction compared to the prediction using the approximate kernel $\hat{K}_{SM}$.
- Comparison between VFE (SM) and Exact, Exact-W, and Exact-WN shows that the parameters estimated by SVSS and its variants have led to the similar level of test RMSE and the slightly degraded test MNLL compared to VFE (SM) for $Q = 4,\ M \in \{100, 400\}$ and $Q = 6,\ M \in \{150, 600\}$, while they take much less training time as shown in Figs. 17 and 18.
- Comparison between Exact-W and Exact-WN shows that the parameters estimated by SVSS-WsNg have led the slightly degraded RMSE and MNLL compared to those of SVSS-Ws although SVSS-WsNg takes less training time due to the fast convergence by natural gradient as shown in Figs. 17 and 18. This should be more concerned so that the SVSS replaces the VFE by employing the advantage of the SVSS that it takes less training time compared to VFE generally.

Table 5. Test RMSE of UCI Datasets regression task

			Test RMSE							
Dataset	N	D	VFE (RBF)	VFE (SM)	SS	VSS	**SVSS**	**Exact**	**Exact-W**	**Exact-WN**
M=100 (Q=4, m=25)										
wine	1599	11	0.460 ± .038	0.435 ± .032	0.856 ± .095	0.644 ± .024	0.459 ± .023	0.431 ± .017	**0.423** ± .018	**0.423** ± .017
skillcraft	3338	19	**0.637** ± .014	0.653 ± .016	1.034 ± .025	0.641 ± .013	0.641 ± .014	0.662 ± .013	0.670 ± .015	0.663 ± .012
sml	4137	26	0.042 ± .002	0.043 ± .002	0.741 ± .014	0.189 ± .002	0.076 ± .003	**0.041** ± .002	**0.041** ± .003	0.050 ± .002
parkinsons	5875	20	0.041 ± .001	**0.041** ± .001	0.172 ± .013	0.312 ± .004	0.085 ± .001	0.044 ± .003	0.045 ± .003	0.069 ± .002
kin8nm	8192	8	0.304 ± .004	0.260 ± .006	0.320 ± .022	0.836 ± .001	0.294 ± .006	0.259 ± .004	0.262 ± .005	**0.258** ± .004
M=400 (Q=4, m=100)										
wine	1599	11	0.460 ± .038	**0.447** ± .032	1.280 ± .095	0.663 ± .024	0.465 ± .038	0.464 ± .042	**0.447** ± .029	0.464 ± .018
skillcraft	3338	19	**0.638** ± .014	0.666 ± .015	1.202 ± .016	0.645 ± .016	0.689 ± .015	0.714 ± .013	0.703 ± .014	0.738 ± .029
sml	4137	26	0.028 ± .001	**0.026** ± .001	0.682 ± .027	0.189 ± .002	0.033 ± .002	0.027 ± .003	**0.026** ± .002	0.031 ± .002
parkinsons	5875	20	0.023 ± .002	**0.022** ± .001	0.157 ± .049	0.329 ± .010	0.037 ± .003	0.028 ± .004	0.028 ± .004	0.035 ± .003
kin8nm	8192	8	0.270 ± .002	**0.247** ± .003	0.266 ± .006	0.832 ± .012	0.258 ± .004	0.249 ± .004	0.249 ± .004	0.251 ± .004
M=150 (Q=6, m=25)										
pol*	15000	26	0.154 ± .003	**0.124** ± .003	0.764 ± .013	0.729 ± .003	0.175 ± .001	0.131 ± .002	**0.124** ± .002	0.153 ± .002
elevator*	16599	18	0.384 ± .002	**0.361** ± .001	0.677 ± .042	0.523 ± .006	0.383 ± .003	0.373 ± .003	0.374 ± .002	0.372 ± .001
bike*	17379	17	0.042 ± .021	**0.027** ± .037	0.270 ± .064	0.544 ± .060	0.087 ± .101	0.053 ± .002	0.039 ± .002	0.059 ± .002
keggd*	48827	22	**0.087** ± .002	0.091 ± .002	0.287 ± .024	0.260 ± .002	0.090 ± .003	**0.087** ± .002	**0.087** ± .002	**0.087** ± .002
M=600 (Q=6, m=100)										
pol*	15000	26	0.142 ± .002	**0.115** ± .003	0.809 ± .004	0.732 ± .003	0.136 ± .002	0.124 ± .002	0.118 ± .002	0.128 ± .003
elevator*	16599	18	0.379 ± .003	**0.363** ± .004	1.076 ± .065	0.516 ± .000	0.382 ± .003	0.375 ± .004	0.375 ± .004	0.382 ± .004
bike*	17379	17	0.031 ± .002	**0.017** ± .002	0.339 ± .041	0.525 ± .008	0.066 ± .006	0.036 ± .003	0.039 ± .003	0.037 ± .002
keggd*	48827	22	0.086 ± .002	0.087 ± .002	0.169 ± .008	0.276 ± .002	0.087 ± .003	**0.085** ± .002	**0.085** ± .002	0.087 ± .002

Table 6. Test MNLL of UCI Datasets regression task

			Test MNLL							
Dataset	N	D	VFE (RBF)	VFE (SM)	SS	VSS	**SVSS**	**Exact**	**Exact-W**	**Exact-WN**
M=100 (Q=4, m=25)										
wine	1599	11	0.720 ± .089	**0.637** ± .069	4.572 ± .833	0.990 ± .044	0.951 ± .079	0.759 ± .052	0.733 ± .056	0.787 ± .069
skillcraft	3338	19	**1.006** ± .019	1.013 ± .019	9.992 ± .785	0.986 ± .026	2.589 ± .129	2.307 ± .097	2.402 ± .115	2.377 ± .080
sml	4137	26	**-1.716** ± .029	-1.713 ± .040	3.615 ± .149	-0.240 ± .009	-1.158 ± .035	-1.422 ± .011	-1.423 ± .025	-1.172 ± .027
parkinsons	5875	20	-1.755 ± .025	**-1.760** ± .021	-0.347 ± .076	0.258 ± .012	-1.031 ± .012	-1.285 ± .013	-1.297 ± .014	-0.850 ± .009
kin8nm	8192	8	0.239 ± .012	0.073 ± .020	0.347 ± .112	1.653 ± .231	0.214 ± .025	0.064 ± .015	0.078 ± .017	**0.060** ± .013
M=400 (Q=4, m=100)										
wine	1599	11	0.724 ± .057	0.734 ± .049	79.80 ± 61.93	1.064 ± .025	0.729 ± .045	**0.609** ± .037	0.699 ± .071	0.724 ± .065
skillcraft	3338	19	**0.970** ± .027	1.029 ± .026	14.41 ± .549	1.095 ± .087	2.444 ± .101	1.741 ± .018	1.651 ± .053	1.745 ± .022
sml	4137	26	-2.117 ± .128	-2.174 ± .069	2.320 ± .219	-0.24 ± .010	-2.103 ± .041	-2.257 ± .019	**-2.261** ± .021	-2.146 ± .019
parkinsons	5875	20	-2.330 ± .054	**-2.348** ± .043	-0.606 ± .330	0.338 ± .045	-1.889 ± .107	-2.128 ± .033	-2.120 ± .054	-1.824 ± .007
kin8nm	8192	8	0.120 ± .012	**0.022** ± .021	0.092 ± .014	1.358 ± .136	0.059 ± .014	0.025 ± .010	0.027 ± .009	0.032 ± .011
M=150 (Q=6, m=25)										
pol*	15000	26	-0.383 ± .013	**-0.627** ± .024	3.622 ± .142	1.164 ± .021	-0.344 ± .007	-0.481 ± .005	-0.532 ± .004	-0.261 ± .008
elevator*	16599	18	0.471 ± .007	**0.406** ± .002	3.109 ± .043	0.820 ± .020	0.648 ± .018	0.584 ± .018	0.590 ± .018	0.571 ± .010
bike*	17379	17	-1.685 ± .070	**-2.012** ± .179	0.633 ± .040	1.014 ± .021	-1.009 ± .023	-1.249 ± .013	-1.547 ± .023	-1.235 ± .021
keggd*	48827	22	-1.010 ± .022	-0.838 ± .031	0.239 ± .122	0.070 ± .005	-1.006 ± .023	-1.023 ± .022	-1.027 ± .023	**-1.029** ± .023
M=600 (Q=6, m=100)										
pol*	15000	26	-0.418 ± .021	-0.555 ± .033	2.486 ± .010	1.122 ± .007	-0.662 ± .017	-0.729 ± .015	**-0.771** ± .014	-0.656 ± .014
elevator*	16599	18	0.451 ± .021	**0.411** ± .055	1.397 ± .151	0.833 ± .026	0.612 ± .034	0.564 ± .015	0.564 ± .016	0.591 ± .017
bike*	17379	17	−1.836 ± .085	**−2.472** ± .123	0.414 ± .203	0.957 ± .015	-1.505 ± .074	-1.679 ± .058	-1.792 ± .053	-1.710 ± .024
keggd*	48827	22	-1.030 ± .023	-1.018 ± .024	-0.572 ± .057	0.213 ± .035	-1.033 ± .029	-1.037 ± .029	**-1.038** ± .029	-1.035 ± .029

Performance difference between VSS and SVSS. In this experiment, we observe that VSS often yields significantly inferior prediction results compared to SVSS as shown in Figs. 17 and 18 and Tables 5 and 6. For these results, we guess the reason that since VSS employs the much large number of trainable parameters as described in Table 2, the training performance of VSS could be more sensitive to those initial parameters, which could result in inferior prediction results. Therefore, we check this claim by conducting the regression task on the Gas dataset in UCI benchmark dataset. We compare the VSS, SVSS, SVSS-Ws (using the weighted sampling), and SVSS-WsNg (using both the weighted sampling and the approximate natural gradient).

Fig. 20 shows the training ELBO (left) and its RMSE on validation set (right) for 5 different random seeds. In left figure, we see that all inference methods shows consistent training ELBO progress for 5 experiments; SVSS, SVSS-Ws, and SVSS-WsNg obtain higher ELBO than VSS as described in the lower bound part of Section 4. In the right figure, VSS shows the volatile prediction progress. Especially, out of the 5 experiments, VSS shows the similar validation RMSE progress for 2 experiments, and totally different progress for the left 3 experiments; note that the results of the RMSE at 0-th iteration are inconsistent for 5 experiments. On the other hand, SVSS, SVSS-Ws, and SVSS-WsNg show the consistent initial RMSE at 0-th iteration for 5 experiments, and leads the stable progress of RMSE on validation set. We believe that this explains why SVSS and its variants results in superior prediction results than VSS.

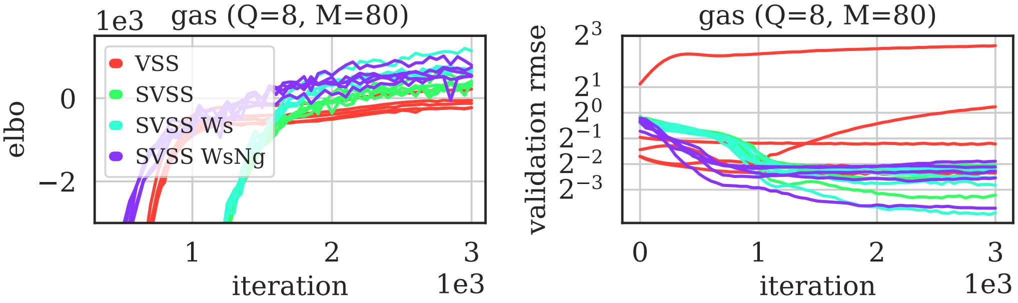


Figure 20. Comparison between VSS and SVSS on Gas $(N = 2568,\ D = 128)$

E. Proofs

E.1. Proof for Proposition 3.1

To prove Proposition 3.1, we introduce Lemma E.1 for the Matrix Bernstein inequality and Lemma E.2 for the unbiasedness of the random feature map. Using Lemmas E.1 and E.2, we prove Proposition 3.1.

Lemma E.1. *Let $\{X_t\}_{t=1}^{T} \in R^{N \times N}$ be a finite sequence of independent, random, Hermitian matrix. For all t, we assume that* $\mathrm{E}[X_t] = 0$ *and* $\|X_t\|_2 \leq L$ *where* $\|\cdot\|_2$ *is spectral norm. We define* $Y = \sum_{t=1}^{T} X_t$ *and* $v(Y) = \left\|\sum_{t=1}^{T} \mathrm{E}\left[X_t^2\right]\right\|_2$. *Then, we obtain the Matrix Bernstein bound of* $\mathrm{E}[\|Y\|_2]$ *as*

$$\mathrm{E}[\|Y\|_2] \leq \sqrt{2v(Y)\log(N)} + \frac{1}{3} L \log N$$

Similarly, for $\epsilon \geq 0$, *we obtain the Matrix Bernstein bound of* $\Pr\big(\|Y\|_2 \geq \epsilon\big)$ *as*

$$\Pr\big(\|Y\|_2 \geq \epsilon\big) \leq N \exp\left(\frac{-\epsilon^2/2}{v(Y) + L\epsilon/3}\right)$$

Proof. See the proof of Theorem 6.6.1 in (Tropp et al., 2015). □

Lemma E.2. *Let* $\tilde{s} = \cup_{q=1}^{Q}\{s_{q,1}, .., s_{q,m_q}\}$ *be the spectral points and us assume the variational distribution* $q(\tilde{s}) = \prod_{q=1}^{Q}\prod_{i=1}^{m_q} N(s_{q,i}; \mu_q, \sigma_q^2)$ *where* m_q *is the number of spectral points drawn from the q-th spectral density component. If we independently sample the spectral points* $s_{q,i}$ *from* $N(s_{q,i}; \mu_q, \sigma_q^2)$ *by the reparameterization trick as* $s_{q,i} = \mu_q + \sigma_q \circ \epsilon_i$ *and* $\epsilon_i \sim N(\epsilon; 0, I)$, *we can define the random feature map* $\phi_{\mathrm{SM}}(x; \tilde{s})$ *as*

$$\phi_{\mathrm{SM}}(x; \tilde{s}) = \left[\sqrt{w_1}\phi\left(x, \{s_{1,i}\}_{i=1}^{m_1}\right), ..., \sqrt{w_Q}\phi\left(x, \{s_{Q,i}\}_{i=1}^{m_Q}\right)\right] \in R^{1 \times 2M},$$

where $M = \sum_{q=1}^{Q} m_q$ *is the total number of spectral points and* $\phi\big(x, \{s_{q,i}\}_{i=1}^{m_q}\big)$ *is the feature map of q-component of mixture distribution defined as*

$$\phi\big(x, \{s_{q,i}\}_{i=1}^{m_q}\big) = \frac{1}{\sqrt{m_q}}[\cos\left(2\pi s_{q,1}^T x\right), \sin\left(2\pi s_{q,1}^T x\right), .., \cos\left(2\pi s_{q,m_q}^T x\right), \sin\left(2\pi s_{q,m_q}^T x\right)] \in R^{1 \times 2m_q}$$

Then, for the inputs $x_1, x_2 \in R^D$, *the* $\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T$ *is the unbiased estimator* $k_{SM}(x_1 - x_2)$ *with the parameters* $\{w_q, \mu_q, \sigma_q^2\}_{q=1}^{Q}$, *i.e.,* $\mathrm{E}_{\tilde{s} \sim q(\tilde{s})}[\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T] = k_{SM}(x_1 - x_2)$

Proof. To show that $\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T$ is the unbiased estimator of $k_{SM}(x_1 - x_2)$, we first compute the $\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T$ as

$$\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T = \sum_{q=1}^{Q} \frac{w_q}{m_q} \sum_{i=1}^{m_q} \cos\left(2\pi s_{q,i}^T(x_1 - x_2)\right)$$

Then, we show that $\mathrm{E}_{\tilde{s} \sim q(\tilde{s})}[\phi_{\mathrm{SM}}(x_1; \tilde{s})\phi_{\mathrm{SM}}(x_2; \tilde{s})^T]$ turns out to be $k_{SM}(x_1 - x_2)$ as

$$\begin{aligned}
\mathrm{E}_{\tilde{s} \sim q(\tilde{s})}\left[\sum_{q=1}^{Q} \frac{w_q}{m_q} \sum_{i=1}^{m_q} \cos\left(2\pi s_{q,i}^T(x_1 - x_2)\right)\right] &= \sum_{q=1}^{Q} w_q \mathrm{E}_{s_{q,1} \sim N(s_{q,1}; \mu_q, \sigma_q^2)}\left[\cos\left(2\pi s_{q,1}^T(x_1 - x_2)\right)\right] \\
&= \sum_{q=1}^{Q} w_q \mathrm{E}_{s_{q,1}}\left[\frac{e^{i2\pi s_{q,1}^T(x_1 - x_2)} + e^{-i2\pi s_{q,1}^T(x_1 - x_2)}}{2}\right] \\
&= \sum_{q=1}^{Q} w_q k_q(x_1 - x_2) = k_{SM}(x_1 - x_2)
\end{aligned}$$

where $k_q(x_1 - x_2) = \mathrm{E}_{s_{q,1} \sim N(s_{q,1}; \mu_q, \sigma_q^2)}[e^{i2\pi s_{q,1}^T(x_1 - x_2)}]$ is the q-th component of SM kernel. In the first equality, we use linearity of expectation and $\{s_{q,i}\}_{i=1}^{m_q}$, that is independently sampled from $N(s_{q,i}; \mu_q, \sigma_q^2)$. In the second equality, we use Euler's identity. In the second equality, we use the symmetric property of stationary kernel. □

Proposition E.3. *Given $X = \{x_n\}_{n=1}^{N}$, let us define the random feature matrix $\Phi^{\text{SM}}(X) = [\phi_{\text{SM}}(x_1;\tilde{s}); ...; \phi_{\text{SM}}(x_N;\tilde{s})] \in R^{N \times 2M}$. Additionally, let us denote $W_0 = \big(\sum_{q=1}^{Q} w_q^2\big)^{1/2}$ and $M = Qm_1$ under the assumption of $m_1 = .. = m_Q$. Then, for a small $\epsilon > 0$, the error bound of $\hat{K}_{\text{SM}}(X,X) := \Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T$ using the matrix spectral norm $\|\cdot\|_2$ is obtained as*

$$\Pr\Big(\big\|\hat{K}_{\text{SM}}(X,X) - K_{\text{SM}}(X,X)\big\|_2 \geq \epsilon\Big) \leq N \exp\Big(\frac{-3\epsilon^2 M}{NW_0Q\big(6\|K_{\text{SM}}(X,X)\|_2 + 3NW_0\sqrt{Q} + 8\epsilon\big)}\Big)$$

Proof. We employ Lemma E.1 of the matrix Bernstein bound inequality to prove our statement. We basically follow the proof structure of theorem 3 in (Lopez-Paz et al., 2014).

We first define the error matrix E as

$$E = \Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T - K_{\text{SM}}(X,X)$$

and then show that E can be represented as $\sum_{q=1}^{Q}\sum_{i=1}^{m_q} E_i^q$ such that E_i^q satisfies the condition of X_t in Lemma 1. For the spectral point $s_{q,i} \sim N(s_{q,i};\mu_q,\sigma_q^2)$, we define the Z_q^i as

$$Z_q^i = \big[\exp\big(i2\pi s_{q,i}^T x_1\big); ...; \exp\big(i2\pi s_{q,i}^T x_N\big)\big] \in R^{N \times 1}$$

and we show $\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T = \sum_{q=1}^{Q}\sum_{i=1}^{m_q}\frac{w_q}{m_q}\text{Re}\left(Z_q^i\overline{Z_q^i}^T\right)$ as

$$\begin{aligned}
[\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T]_{n,m} &= \sum_{q=1}^{Q}\frac{w_q}{m_q}\sum_{i=1}^{m_q}\cos\big(2\pi s_{q,i}^T(x_n - x_m)\big)\\
&= \sum_{q=1}^{Q}\sum_{i=1}^{m_q}\frac{w_q}{m_q}\text{Re}\big(\exp\big(i2\pi s_{q,i}^T(x_n - x_m)\big)\big)\\
&= \sum_{q=1}^{Q}\sum_{i=1}^{m_q}\frac{w_q}{m_q}\text{Re}\left(\left[Z_q^i\overline{Z_q^i}^T\right]_{n,m}\right)
\end{aligned}$$

Then, using $K_{SM}(X,X) = \sum_{q=1}^{Q}\sum_{i=1}^{m_q}\frac{w_q}{m_q}\text{E}_{s_{q,i}}\left[\text{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right]$ by the previous factorization and Lemma 2, we can factorize the error matrix $E = \sum_{q=1}^{Q}\sum_{i=1}^{m_q} E_i^q$ where

$$E_i^q = \frac{w_q}{m_q}\left(\text{Re}\left(Z_q^i\overline{Z_q^i}^T\right) - \text{E}_{s_{q,i}}\left[\text{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right]\right)$$

We can check that $E_i^q \in R^{N \times N}$ satisfy the condition of Lemma 1 because each E_i^q is independent random and symmetric matrix satisfying $\text{E}_{s_{q,i}}[E_i^q] = 0$.

Next, for the application of Lemma 1, we will find upper bound of $\|E_i^q\|_2$ and $\left\|\sum_{q=1}^{Q}\sum_{i=1}^{m_q}\text{E}_{s_{q,i}}[(E_i^q)^2]\right\|_2$. To this end, we define the following notation and compute the necessary terms as

$$\begin{aligned}
&c_q^i = [\cos\big(2\pi s_{q,i}^T x_1\big); ..; \cos\big(2\pi s_{q,i}^T x_N\big)] \in R^{N \times 1},\\
&s_q^i = [\sin\big(2\pi s_{q,i}^T x_1\big); ..; \sin\big(2\pi s_{q,i}^T x_N\big)] \in R^{N \times 1},\\
&\text{Re}\left(Z_q^i\overline{Z_q^i}^T\right) = c_q^i c_q^{i\,T} + s_q^i s_q^{i\,T}.
\end{aligned}$$

Then, we first bound $\|E_i^q\|_2$ as

$$\begin{aligned}
\|E_i^q\|_2 &= \frac{w_q}{m_q}\left\|\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right) - \mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right]\right\|_2 \\
&\leq \frac{w_q}{m_q}\left(\left\|\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right\|_2 + \left\|\mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right]\right\|_2\right) \\
&\leq \frac{w_q}{m_q}\left(\left\|\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right\|_2 + \|c_q^i\|^2 + \|s_q^i\|^2\right) \\
&\leq \frac{W_0}{m_0}\left(2\|c_q^i\|_2^2 + 2\|s_q^i\|_2^2\right) \leq \frac{W_0}{m_0}4N \qquad \cdots \qquad (*)
\end{aligned}$$

In the second and third inequality, we use the triangle inequality and Jensen's inequality. In addition, we use $\left\|\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\right\|_2 = \sup_{|v|_2^2=1} v^T\left(c_q^i {c_q^i}^T + s_q^i {s_q^i}^T\right)v = \|c_q^i\|_2^2 + \|s_q^i\|_2^2 \leq 2N$ for the fourth inequality.

For another bound of $\left\|\sum_{q=1}^{Q}\sum_{i=1}^{m_q}\mathrm{E}_{s_{q,i}}[(E_i^q)^2]\right\|_2$, we bound $\mathrm{E}_{s_{q,i}}[(E_i^q)^2]$ in the sense of the matrix inequality $\preccurlyeq$ where $A \preccurlyeq B$ implies $B - A$ is a Positive Semi definite (PSD) matrix.

$$\begin{aligned}
\frac{m_q^2}{w_q^2}\mathrm{E}_{s_{q,i}}[(E_i^q)^2] &= \mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^{iT}}\right)^2\right] - \left(\mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^{iT}}\right)\right]\right)^2 \\
&\preccurlyeq \mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^{iT}}\right)^2\right] \\
&= \mathrm{E}_{s_{q,i}}\left[({c_q^i}^T c_q^i)c_q^i {c_q^i}^T + ({s_q^i}^T s_q^i)s_q^i {s_q^i}^T + ({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\right] \\
&\preccurlyeq N\mathrm{E}_{s_{q,i}}\left[c_q^i {c_q^i}^T + s_q^i {s_q^i}^T\right] + \mathrm{E}_{s_{q,i}}\left[({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\right] \\
&= N\mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^{iT}}\right)\right] + \mathrm{E}_{s_{q,i}}\left[({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\right]
\end{aligned}$$

The first inequality holds because $\left(\mathrm{E}_{s_{q,i}}\left[\mathrm{Re}\left(Z_q^i\overline{Z_q^{iT}}\right)\right]\right)^2$ is PSD matrix. Second inequality is explained using ${c_q^i}^T c_q^i + {s_q^i}^T s_q^i = N$,

$$N\mathrm{E}_{s_{q,i}}\left[c_q^i {c_q^i}^T + s_q^i {s_q^i}^T\right] - \mathrm{E}_{s_{q,i}}\left[({c_q^i}^T c_q^i)c_q^i {c_q^i}^T + ({s_q^i}^T s_q^i)s_q^i {s_q^i}^T\right] = \mathrm{E}_{s_{q,i}}\left[({s_q^i}^T s_q^i)c_q^i {c_q^i}^T + ({c_q^i}^T c_q^i)s_q^i {s_q^i}^T\right],$$

which is PSD matrix because, for any $a \in R^N$,

$$\begin{aligned}
& a^T\mathrm{E}_{s_{q,i}}\left[({s_q^i}^T s_q^i)c_q^i {c_q^i}^T + ({c_q^i}^T c_q^i)s_q^i {s_q^i}^T\right]a \\
&= \sum_{m=1}^{N}\sum_{n=1}^{N} a_m a_n \mathrm{E}_{s_{q,i}}\left[({s_q^i}^T s_q^i)\mathrm{cos}(2\pi s_{q,i}^T x_m)\mathrm{cos}(2\pi s_{q,i}^T x_n) + ({c_q^i}^T c_q^i)\mathrm{sin}(2\pi s_{q,i}^T x_m)\mathrm{sin}(2\pi s_{q,i}^T x_n)\right] \\
&= \mathrm{E}_{s_{q,i}}\left[({s_q^i}^T s_q^i)a^T(c_q^i {c_q^i}^T)a + ({c_q^i}^T c_q^i)a^T(s_q^i {s_q^i}^T)a\right] \geq 0
\end{aligned}$$

where $({s_q^i}^T s_q^i) \geq 0$, $a^T(c_q^i {c_q^i}^T)a \geq 0$, $({c_q^i}^T c_q^i) \geq 0$, and $a^T(s_q^i {s_q^i}^T)a \geq 0$.

Then, we can bound $\left\|\sum_{q=1}^{Q}\sum_{i=1}^{m_q}\mathrm{E}[(E_i^q)^2]\right\|_2$ with the defined $W_0 = \left(\sum_{q=1}^{Q} w_q^2\right)^{1/2}$ and $m_1 = .. = m_Q$ as

$$\begin{aligned}
\left\|\sum_{q=1}^{Q}\sum_{i=1}^{m_q}\mathrm{E}[(E_i^q)^2]\right\|_2 &\le \left\|\sum_{q=1}^{Q}\sum_{i=1}^{m_q}\frac{w_q^2}{m_q^2}\Big(N\mathrm{E}_{s_{q,i}}\Big[\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\Big] + \mathrm{E}_{s_{q,i}}\Big[({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\Big]\Big)\right\|_2 \\
&= \frac{W_0}{m_1}\left\|\sum_{q=1}^{Q} w_q\Big(N\mathrm{E}_{s_{q,i}}\Big[\mathrm{Re}\left(Z_q^i\overline{Z_q^i}^T\right)\Big] + \mathrm{E}_{s_{q,i}}\Big[({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\Big]\Big)\right\|_2 \\
&\le \frac{W_0}{m_1}\Big(N\|K_{SM}(X,X)\|_2 + \sum_{q=1}^{Q} w_q\Big\|\mathrm{E}_{\mathbf{s}_{q,i}}\Big[({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\Big]\Big\|_2\Big) \\
&\le \frac{W_0}{m_1}\Big(N\|K_{SM}(X,X)\|_2 + \sum_{q=1}^{Q} w_q\mathrm{E}_{s_{q,i}}\Big[\|({s_q^i}^T c_q^i)\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\|_2\Big]\Big) \\
&\le \frac{W_0}{m_1}\Big(N\|K_{SM}(X,X)\|_2 + \frac{N}{2}\sum_{q=1}^{Q} w_q\mathrm{E}_{s_{q,i}}\Big[\|\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\|_2\Big]\Big) \\
&\le \frac{W_0 N}{m_1}\Big(\|K_{SM}(X,X)\|_2 + \frac{N}{2}W_0\sqrt{Q}\Big) \qquad \cdots \qquad (**)
\end{aligned}$$

In the first inequality, we use that $\|A\|_2 \le \|B\|_2$ if $B - A$ is PSD matrix and the summation of PSD matrices is PSD matrix. In the second and third inequality, we use triangle inequality and Jensen's inequality. In the fourth inequality, we use $|{s_q^i}^T c_q^i| \le \frac{N}{2}$. In the last inequality, we use

$$\begin{aligned}
\mathrm{E}_{s_{q,i}}\Big[\|\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)\|_2\Big] &= \sup_{|v|_2^2=1}\mathrm{E}_{s_{q,i}}\Big[v^T\left(s_q^i {c_q^i}^T + c_q^i {s_q^i}^T\right)v\Big] \\
&= \sup_{|v|_2^2=1}\mathrm{E}_{s_{q,i}}\Big[\sum_{n=1}^{N}\sum_{m=1}^{N} v_n v_m \sin 2\pi s_{q,i}^T\left(x_n + x_m\right)\Big] \le \sup_{|v|_2^2=1}\left(\sum_{n=1}^{N} v_n\right)^2 = N,
\end{aligned}$$

and $\sum_{q=1}^{Q} w_q \le W_0\sqrt{Q}$ by the Cauchy–Schwarz inequality.

After applying the derived upper bounds of $(*)$ and $(**)$ to the L and $v(Y)$ in Lemma 1, we bound the probability of the event $\{\tilde{s};\left\|\Phi^{\mathrm{SM}}(X)\Phi^{\mathrm{SM}}(X)^T - K_{\mathrm{SM}}(X,X)\right\|_2 \ge \epsilon\}$ as

$$\Pr\Big(\left\|\Phi^{\mathrm{SM}}(X)\Phi^{\mathrm{SM}}(X)^T - K_{\mathrm{SM}}(X,X)\right\|_2 \ge \epsilon\Big) \le N\exp\Big(\frac{-3\epsilon^2 M}{NW_0Q\big(6\|K_{\mathrm{SM}}(X,X)\|_2 + 3NW_0\sqrt{Q} + 8\epsilon\big)}\Big)$$

□

E.2. Proof for Proposition 3.2

Proof for the ELBO Estimator $\hat{\mathcal{L}}_J$ in Eq. (10).

Let $q(\tilde{s})$ be the variational distribution of the spectral points defined in Eq. (7). We derive the ELBO estimator $\hat{\mathcal{L}}_j$ as follows:

$$\begin{aligned}
\log p(Y|X) &= \log \int p(Y|X,\tilde{s})\frac{p(\tilde{s})}{q(\tilde{s})}q(\tilde{s})d\tilde{s} \\
&\geq \int \log\left(p(Y|X,\tilde{s})\frac{p(\tilde{s})}{q(\tilde{s})}\right)q(\tilde{s})d\tilde{s} \\
&= \int \log p(Y|X,\tilde{s})q(\tilde{s})dS - (q(\tilde{s})||p(\tilde{s})) \\
&\approx \frac{1}{J}\sum_{j=1}^{J}\log p(Y|X,\tilde{s}^{(j)}) - \mathrm{KL}(q(\tilde{s})||p(\tilde{s})) := \hat{\mathcal{L}}_j
\end{aligned}$$

where $p(Y|X,\tilde{s}^{(j)}) = N(Y;0,\hat{K}^{(j)}_{\mathrm{SM}}(X,X)+\sigma_\epsilon^2 I)$ is the log marginal likelihood using the approximate kernel $\hat{K}^{(j)}_{\mathrm{SM}}(X,X)$ with the spectral points $\tilde{s}^{(j)} = \cup_{q=1}^{Q}\{s^{(j)}_{q,1},..,s^{(j)}_{q,m_q}\}$ that are sampled from $q(\tilde{s})$ at j-th times. In the first inequality, we use the Jensen inequality.

Proposition E.4. *The error of ELBO estimator $\hat{\mathcal{L}}_J$ in Eq. (11) is bounded as*

$$0 \leq \log p(Y|X) - \hat{\mathcal{L}}_J \leq \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{2\sigma_\epsilon^4}\right)\mathrm{E}_{q(\tilde{s})}\left[\|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}\|_2\right] + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})).$$

Proof. Let us denote $\hat{K}^{(j)}_{\mathrm{SM}} = \hat{K}^{(j)}_{\mathrm{SM}}(X,X) + \sigma_\epsilon^2 I \in R^{N\times N}$ and $K_{\mathrm{SM}} = K_{\mathrm{SM}}(X,X) + \sigma_\epsilon^2 I \in R^{N\times N}$. Then, the approximation gap, $\log p(Y|X) - \hat{\mathcal{L}}_J$, between the log marginal likelihood $\log p(Y|X)$ and the ELBO estimator $\hat{\mathcal{L}}_J$, is derived as

$$\begin{aligned}
\log p(Y|X) - \hat{\mathcal{L}}_J &= \frac{1}{2J}\sum_{j=1}^{J} Y^T\left(\hat{K}^{(j)\,-1}_{\mathrm{SM}} - K_{\mathrm{SM}}{}^{-1}\right)Y + \frac{1}{2J}\sum_{j=1}^{J}\log\frac{|\hat{K}^{(j)}_{\mathrm{SM}}|}{|K_{\mathrm{SM}}|} + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})) \\
&= \frac{1}{2J}\sum_{j=1}^{J}\underbrace{Y^T\hat{K}^{(j)\,-1}_{\mathrm{SM}}\left(K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}}\right)K^{-1}_{\mathrm{SM}}Y}_{\text{(1) term}} + \frac{1}{2J}\sum_{j=1}^{J}\underbrace{\log\frac{|\hat{K}^{(j)}_{\mathrm{SM}}|}{|K_{\mathrm{SM}}|}}_{\text{(2) term}} + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})).
\end{aligned}$$

In the second equality, we use $A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}$ where $A \in R^{N\times N}$ and $B \in R^{N\times N}$ are the invertible matrix. We bound the (1) term and (2) term in the above equality.

For the (1) term, we bound it using matrix 2-norm as

$$\begin{aligned}
Y^T\hat{K}^{(j)\,-1}_{\mathrm{SM}}(K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}})K^{-1}_{\mathrm{SM}}Y &\leq \|Y^T\hat{K}^{(j)\,-1}_{\mathrm{SM}}(K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}})K^{-1}_{\mathrm{SM}}Y\|_2 \\
&\leq \|Y\|_2\left\|\hat{K}^{(j)\,-1}_{\mathrm{SM}}\left(K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}}\right)K^{-1}_{\mathrm{SM}}Y\right\|_2 \\
&\leq \|Y\|_2^2\left\|\hat{K}^{(j)\,-1}_{\mathrm{SM}}\left(K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}}\right)K^{-1}_{\mathrm{SM}}\right\|_2 \\
&\leq \|Y\|_2^2\|\hat{K}^{(j)\,-1}_{\mathrm{SM}}\|_2\|K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}}\|_2\|K^{-1}_{\mathrm{SM}}\|_2 \\
&\leq \frac{\|Y\|_2^2}{\sigma_\epsilon^4}\|K_{\mathrm{SM}} - \hat{K}^{(j)}_{\mathrm{SM}}\|_2
\end{aligned}$$

For the second - fourth inequalities, we employ that matrix inequality $\|AB\|_2 \leq \|A\|_2\|B\|_2$ for $A \in R^{m\times n}, B \in R^{n\times p}$. For the last inequality, we employ that the last eigenvalue $\sigma_N(K_{\mathrm{SM}}) = \sigma_N(K_{\mathrm{SM}}(X,X) + \sigma_\epsilon^2 I) \geq \sigma_\epsilon^2$. Thus, the first eigenvalue $\|K^{-1}_{\mathrm{SM}}\|_2 = \sigma_1(K^{-1}_{\mathrm{SM}}) = \frac{1}{\sigma_N(K_{\mathrm{SM}})} \leq \frac{1}{\sigma_\epsilon^2}$. Similarly, $\|\hat{K}^{(j)\,-1}_{\mathrm{SM}}\|_2 \leq \frac{1}{\sigma_\epsilon^2}$.

For the (2) term, we bound it by using that the log determinant function $f(A) = \log|A|$ is a concave function for the symmetric matrix $A \in R^{N\times N}$. Using the first-order inequality of concave function $f(A)$, that is $f(A) \leq f(B) +$

$\langle \nabla f(B), A - B \rangle$, $\nabla f(A) = A^{-1}$, and the matrix inner product $\langle A, B \rangle = \mathrm{Tr}\,(A^T B)$ with $A, B \in R^{N \times N}$, we obtain the following upper bound of $\log |\hat{K}_{\mathrm{SM}}^{(j)}|$ as

$$\log |\hat{K}_{\mathrm{SM}}^{(j)}| \leq \log |K_{\mathrm{SM}}| + \mathrm{Tr}\left(K_{\mathrm{SM}}^{-1} (\hat{K}_{\mathrm{SM}}^{(j)} - K_{\mathrm{SM}}) \right),$$

which implies the upper bound of $\log |\hat{K}_{\mathrm{SM}}^{(j)}| - \log |K_{\mathrm{SM}}|$ as

$$\log |\hat{K}_{\mathrm{SM}}^{(j)}| - \log |K_{\mathrm{SM}}| \leq \mathrm{Tr}\left(K_{\mathrm{SM}}^{-1} (\hat{K}_{\mathrm{SM}}^{(j)} - K_{\mathrm{SM}}) \right) \leq \mathrm{Tr}\left(K_{\mathrm{SM}}^{-1} \right) \| (\hat{K}_{\mathrm{SM}}^{(j)} - K_{\mathrm{SM}}) \|_2 \leq \frac{N}{\sigma_\epsilon^2} \| (\hat{K}_{\mathrm{SM}}^{(j)} - K_{\mathrm{SM}}) \|_2.$$

For the second inequality, we use $\mathrm{Tr}\,(AB) \leq \mathrm{Tr}\,(A) \|B\|_2$ for the P.S.D matrix A and symmetric real number matrix B that is introduced in (Fang et al., 1994) theorem 1.

For the last inequality, we use $\mathrm{Tr}\left(K_{\mathrm{SM}}^{-1} \right) = \sum_{n=1}^{N} \frac{1}{\sigma_i (K_{\mathrm{SM}}(X,X)) + \sigma_\epsilon^2} \leq \frac{N}{\sigma_\epsilon^2}$ where $\sigma_i(K_{\mathrm{SM}}(X,X))$ denotes i-th eigenvalue for P.S.D matrix $K_{\mathrm{SM}}(X,X) \in R^{N \times N}$.

By employing the above two inequalities for (1) term and (2) term, we can obtain the upper bound of $\log p(Y|X) - \hat{\mathcal{L}}_J$ as

$$\begin{aligned}
\log p(Y|X) - \hat{\mathcal{L}}_J &\leq \frac{1}{2J} \sum_{j=1}^{J} \underbrace{\frac{\|Y\|_2^2}{\sigma_\epsilon^4} \|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}^{(j)}\|_2}_{\text{upper bound of (1) term}} + \underbrace{\frac{N}{\sigma_\epsilon^2} \|\hat{K}_{\mathrm{SM}}^{(j)} - K_{\mathrm{SM}}\|_2}_{\text{upper bound of (2) term}} + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})) \\
&\leq \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{\sigma_\epsilon^4} \right) \frac{1}{2J} \sum_{j=1}^{J} \|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}^{(j)}\|_2 + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})) \\
&\leq \limsup_{J \longrightarrow \infty} \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{\sigma_\epsilon^4} \right) \frac{1}{2J} \sum_{j=1}^{J} \|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}^{(j)}\|_2 + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})) \\
&= \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{2\sigma_\epsilon^4} \right) \mathrm{E}_{q(\tilde{s})} \left[\|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}\|_2 \right] + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})),
\end{aligned}$$

where $\hat{K}_{\mathrm{SM}} = \hat{K}_{\mathrm{SM}}(X,X) + \sigma_\epsilon^2 I \in R^{N \times N}$ is the random kernel Gram matrix constructed by the set of spectral point $\tilde{s}$ sampled from $q(\tilde{s})$. In the first inequality, we use the derived upper bound for (1) term and (2) term. Thus, we can obtain the result of Proposition 3.2 as

$$0 \leq \log p(Y|X) - \hat{\mathcal{L}}_J \leq \left(\frac{\|Y\|_2^2 + N\sigma_\epsilon^2}{2\sigma_\epsilon^4} \right) \mathrm{E}_{q(\tilde{s})} \left[\|K_{\mathrm{SM}} - \hat{K}_{\mathrm{SM}}\|_2 \right] + \mathrm{KL}(q(\tilde{s})||p(\tilde{s})).$$

where $0 \leq \log p(Y|X) - \hat{\mathcal{L}}_J$ is obtained from that $\hat{\mathcal{L}}_J$ is the ELBO estimator of $\log p(Y|X)$.

□

E.3. Proof for Proposition 3.3

We first explain the decomposition introduced in Eq. (13). Then, we introduce Lemma E.5 for the variance of the random feature map described in (Sutherland & Schneider, 2015). We employ Lemma E.5 to prove Proposition 3.3. Additionally, we prove Lemma E.7 that constraining the minimum number of the spectral points in the SVSS inference, to relax that all M spectral points are sampled from a single component of the spectral density, reduces kernel approximation error.

Proof for the decomposition of Eq. (13).

Since $\hat{K}_{\text{SM}}(X,X)$ is defined as $\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T$ where $\tilde{s} = \cup_{q=1}^{Q}\{s_{q,1},..,s_{q,m_q}\}$ is the set of the spectral points sampled from $q(S)$ in Eq. (7) and $\Phi^{\text{SM}}(X)$ is the random feature matrix defined as

$$\begin{aligned}
&\Phi^{\text{SM}}(X) = [\phi_{\text{SM}}(x_1;\tilde{s});..;\phi_{\text{SM}}(x_N;\tilde{s})] \in R^{N\times 2M}\\
&\phi_{\text{SM}}(x;\tilde{s}) = \left[\sqrt{w_1}\phi\left(x,\{s_{1,i}\}_{i=1}^{m_1}\right),...,\sqrt{w_Q}\phi\left(x,\{s_{Q,i}\}_{i=1}^{m_Q}\right)\right] \in R^{1\times 2M},
\end{aligned}$$

$\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T$ can also be decomposed as

$$\Phi^{\text{SM}}(X)\Phi^{\text{SM}}(X)^T = \sum_{q=1}^{Q} w_q\left(\Phi_{\{s_{q,i}\}_{i=1}^{m_q}}(X)\Phi_{\{s_{q,i}\}_{i=1}^{m_q}}(X)^T\right).$$

Thus, $\hat{K}_{\text{SM}} - K_{\text{SM}} = \hat{K}_{\text{SM}}(X,X) - K_{\text{SM}}(X,X)$ can be decomposed as

$$\hat{K}_{\text{SM}} - K_{\text{SM}} = \sum_{q=1}^{Q} w_q\left(\hat{K}_q - K_q\right),$$

where the $\hat{K}_q \in R^{N\times N}$ is $\Phi_{\{s_{q,i}\}_{i=1}^{m_q}}(X)\Phi_{\{s_{q,i}\}_{i=1}^{m_q}}(X)^T$ and the $K_q \in R^{N\times N}$ is the Gram matrix corresponding to the q-th component spectral density of SM kernel.

Lemma E.5. *Let $\phi_m(\tau)$ be the unbiased estimator of the stationary kernel $k(\tau)$ obtained by random feature using m sampled spectral points. Then,* $\text{Var}\left(\phi_m(\tau)\right)$ *and* $\text{Cov}\left(\phi_m(\tau_1),\phi_m(\tau_2)\right)$ *can be obtained as*

$$\begin{aligned}
\text{Var}\left(\phi_m(\tau)\right) &= \frac{1}{2m}\left(1 + k(2\tau) - 2k^2(\tau)\right)\\
\text{Cov}\left(\phi_m(\tau_1),\phi_m(\tau_2)\right) &= \frac{1}{m}\left(\frac{k(\tau_1-\tau_2)+k(\tau_1+\tau_2)}{2} - k(\tau_1)k(\tau_2)\right)
\end{aligned}$$

Proof. Let $p(s)$ be the spectral density of the kernel $k(\tau)$ to satisfy $k(\tau) = \int_s e^{i2\pi s^T\tau}p(s)ds$ by Bochner's theorem. Using the RFF (Rahimi & Recht, 2008), we define the unbiased estimator $\phi_m(\tau) = \frac{1}{m}\sum_{i=1}^{m}\cos\left(2\pi s_i^T\tau\right)$ with $s_i \sim p(S)$. Then, $\text{Cov}\Big(\phi_m(\tau_1),\phi_m(\tau_2)\Big)$ is computed as

$$\begin{aligned}
&\text{Cov}\Big(\phi_m(\tau_1),\phi_m(\tau_2)\Big)\\
&= \sum_{i,j}\frac{1}{m^2}\text{Cov}\Big(\cos\left(2\pi s_i^T\tau_1\right),\cos\left(2\pi s_j^T\tau_2\right)\Big)\\
&= \sum_{i}\frac{1}{m^2}\text{Cov}\Big(\cos\left(2\pi s_i^T\tau_1\right),\cos\left(2\pi s_i^T\tau_2\right)\Big)\\
&= \frac{1}{m}\text{Cov}\Big(\cos\left(2\pi s_1^T\tau_1\right),\cos\left(2\pi s_1^T\tau_2\right)\Big)\\
&= \frac{1}{m}\left(\frac{\text{E}_{s_1}\Big[\cos\left(2\pi s_1^T(\tau_1+\tau_2)\right)+\cos\left(2\pi s_1^T(\tau_1-\tau_2)\right)\Big]}{2} - \text{E}_{s_1}\Big[\cos\left(2\pi s_1^T\tau_1\right)\Big]\text{E}_{s_1}\Big[\cos\left(2\pi s_1^T\tau_2\right)\Big]\right)\\
&= \frac{1}{m}\left(\frac{k(\tau_1+\tau_2)+k(\tau_1-\tau_2)}{2} - k(\tau_1)k(\tau_2)\right)
\end{aligned}$$

In the second equality, we use $\mathrm{Cov}\Big(\cos\left(2\pi s_i^T \tau_1\right), \cos\left(2\pi s_j^T \tau_2\right) \Big) = 0$ for $i \neq j$, because the spectral points are independently sampled. In the third equality, we use that each s_i for $i = 1, .., m$ is identically sampled from $p(s)$. In the last equality, we employ the Bochner's theorem. For the variance of $\mathrm{Var}\left(\phi_m(\tau)\right)$ is obtained when $\tau_1 = \tau_2$.

□

Proposition E.6. *Given inputs $X = \{x_n\}_{n=1}^N$, let m_q be the number of spectral points sampled from $N(\mu_q, \sigma_q^2)$, and $M = \sum_{q=1}^Q m_q$ be the total number of spectral points. Let $a_q = \frac{m_q}{M}$ be the ratio of spectral points. Then, the optimal $a_1^*, .., a_Q^* = \arg\min_{a_1,..,a_Q} \mathrm{E}[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F]$ is given*

$$a_q^* = \frac{w_q \left[\sum_{i=1}^N \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}}{\sum_{q=1}^Q w_q \left[\sum_{i=1}^N \sum_{i<j} g_q(x_i - x_j)\right]^{1/2}}$$

where $g_q(\tau) = 1 + k_q(2\tau) - 2k_q^2(\tau)$ and $k_q(\tau) = \exp\left(-2\pi^2(\tau^T \sigma_q)^2\right) \cos\left(2\pi \mu_q^T \tau\right)$. The integer m_q^* denotes the integer closest to $\max\{1, Mp_q^*\}$.

Proof. Let us denote random spectral points $\tilde{s} = \cup_{q=1}^Q \{s_{q,1}, .., s_{q,m_q}\}$ and consider the variational distribution of spectral points $q(\tilde{s}) = \prod_{q=1}^Q \prod_{i=1}^{m_q} N(s_{q,i}; \mu_q, \sigma_q^2)$. Our objective is to find the optimal number of spectral points $\{m_1^*, .., m_Q^*\}$ which minimizes the error $E[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F^2]$ by solving the following optimization problem:

$$\min_{m_1,..,m_Q} \sum_{i,j=1}^N \mathrm{Var}\left(\phi_{SM}(x_i - x_j)\right) \quad \text{s.t} \quad \sum_{q=1}^Q m_q = M \quad \forall m_q \in \mathbb{Z}^+$$

where $\phi_{SM}(\tau_p) = \sum_{q=1}^Q \frac{w_q}{m_q} \sum_{i=1}^{m_q} \cos\left(2\pi s_{q,i}^T \tau_p\right)$. Since the Jensen inequality implies that

$$0 \leq \mathrm{E}[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F]^2 \leq \mathrm{E}\left[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F^2\right],$$

minimizing $\mathrm{E}[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F^2]$ leads to minimizing $\mathrm{E}[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F]$. The $E[\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F^2]$ is analytically computed as $\sum_{n,m=1}^N \mathrm{Var}\left(\phi_{SM}(x_i - x_j)\right)$ because of $\|\hat{K}_{\mathrm{SM}} - K_{\mathrm{SM}}\|_F^2 = \sum_{i,j=1}^N \left(\phi_{SM}(x_i - x_j) - k_{\mathrm{SM}}(x_i - x_j)\right)^2$ and $E\left[\phi_{SM}(x_i - x_j)\right] = k_{\mathrm{SM}}(x_i - x_j)$.

Since this optimization is somewhat tricky integer programming problem, we take relaxation by transforming variable as $a_q = \frac{m_q}{M}$. Then, the above optimization problem is modified as

$$\min_{a_1,..,a_Q} \sum_{i,j=1}^N \mathrm{Var}\left(\phi_{SM}(x_i - x_j)\right) \quad \text{s.t} \quad \sum_{q=1}^Q a_q = 1 \quad \forall a_q \in [0, 1]$$

The transformed optimization is a convex optimization problem, and the optimal solution can be obtained by applying the Lagrangian method with the KKT condition. Let $\mathcal{L}(a_1, .., a_Q, \lambda)$ be the Lagrangian operator with the multiplier λ.

$$\mathcal{L}(a_1, .., a_Q, \lambda) = \sum_{i,j=1}^N \mathrm{Var}\left(\phi_{SM}(x_i - x_j)\right) + \lambda \left(\sum_{q=1}^Q a_q - 1\right)$$

If we represent $\phi_{SM}(\tau_p) = \sum_{q=1}^Q w_q \phi_{m_q}^q(\tau_p)$ where $\phi_{m_q}^q(\tau_p) = \frac{1}{m_q} \sum_{i=1}^{m_q} \cos\left(2\pi s_{q,i}^T \tau_p\right)$, then $\sum_{i,j=1}^N \mathrm{Var}\left(\phi_{SM}(x_i - x_j)\right)$

can be computed as

$$\begin{aligned}\sum_{i,j=1}^{N}\mathrm{Var}\left(\phi_{SM}(x_i-x_j)\right) &= \sum_{i,j=1}^{N}\mathrm{Var}\left(\sum_{q=1}^{Q} w_q\phi^{q}_{m_q}(x_i-x_j)\right)\\ &= \sum_{i,j=1}^{N}\sum_{q=1}^{Q} w_q^2\mathrm{Var}\left(\phi^{q}_{m_q}(x_i-x_j)\right)\\ &= \sum_{q=1}^{Q}\frac{w_q^2}{2Mp_q}\left[\sum_{i,j=1}^{N} g_q(x_i-x_j)\right]\end{aligned}$$

where $\mathrm{Var}(\phi^q_{m_q}(\tau)) = \frac{1}{2m_q}g_q(\tau)$, $g_q(\tau) = 1 + k_q(2\tau) - 2k_q^2(\tau)$, and $k_q(\tau)$ is the q-th component of SM kernel, i.e., $k_q(\tau) = \exp\left(-2\pi^2(\tau^T\sigma_q)^2\right)\cos\left(2\pi\mu_q^T\tau\right)$ by Lemma E.7 and Lemma E.7. Note that $\sum_{i,j=1}^{N} g_q(x_i-x_j)$ is computed as $2\sum_{i=1}^{N}\sum_{i<j} g_q(x_i-x_j)$ because of $g_q(x_i-x_i)=0$ for $i=1,..,N$ and $g_q(x_i-x_j)=g_q(x_j-x_i)$ for $i\neq j$.

Based on KKT condition, solving the following conditions $\frac{\partial\mathcal{L}(a_1,..,a_Q,\lambda)}{\partial\lambda}=0$ and $\frac{\partial\mathcal{L}(a_1,..,a_Q,\lambda)}{\partial a_q}=0$ for $q=1,..,Q$ leads to the optimal solution $\{a_1^*,..,a_Q^*\}$.

$$a_q^* = \frac{w_q\left[\sum_{i=1}^{N}\sum_{i<j} g_q(x_i-x_j)\right]^{1/2}}{\sum_{q=1}^{Q} w_q\left[\sum_{i=1}^{N}\sum_{i<j} g_q(x_i-x_j)\right]^{1/2}}$$

where $\frac{\partial\mathcal{L}(a_1,..,a_Q,\lambda)}{\partial a_q}$ is computed as $\frac{-w_q^2}{Ma_q^2}\sum_{i=1}^{N}\sum_{i<j} g_q(x_i-x_j)+\lambda$. □

Lemma E.7. *Let $a^* = [a_1^*,..,a_Q^*]\in R^Q$ be the optimal ratio described in Proposition 3.3 and $a_0 = [1/Q,..,1/Q]\in R^Q$ be the base ratio for equal sampling. Then, the mixed ratio $a_\alpha = \alpha a^* + (1-\alpha)a_0$ with $\alpha\in[1/2,1]$ allocates the spectral points $\{m_q^\alpha\}_{q=1}^{Q}$ with $m_q^\alpha = Ma_\alpha$ that has the smaller kernel approximation error than $m_q^0 = M/Q$ for the equal sampling.*

Proof. Let us denote the approximation error $\sum_{i,j=1}^{N}\mathrm{Var}\left(\phi_{SM}(x_i-x_j)\right) := V(p)$ with $a=[a_1,..,a_Q]$ and $\sum_{q=1}^{Q} a_q = 1$. Then, $V(p)$ is a convex function in a because the $\frac{\partial^2 V}{\partial a_i\partial a_j}$ is a P.S.D matrix because of $g_q(x_i-x_j)\geq 0$ for all q, x_i, x_j. Thus, we directly induce that $V(a_a)\leq \alpha V(a^*) + (1-\alpha)V(a_0)\leq V(a_0)$ by $V(a^*)\leq V(a_0)$. □

E.4. Proof for Proposition 3.4

We describe how the natural gradient update for the Gaussian density parameters is conducted in Lemmas E.8 and E.9. Finally, we prove Proposition 3.4 by employing Lemmas E.8 and E.9.

Lemma E.8. *If the probability density $p_\theta(z)$ belongs the class of the exponential family distribution, $p_\theta(z)$ has other parameterization of natural parameter η and mean parameter $\boldsymbol{m}$, i.e., $p_\theta(z) = p_\eta(z) = p_{\boldsymbol{m}}(z)$. Moreover, if $p_\theta(z)$ satisfy the minimal representation condition that η has one-to-one correspondence with $\boldsymbol{m}$, then natural gradient $\tilde{\nabla}_\eta \mathcal{L}(\eta)$ with to respect to the natural parameters η, can be represented as*

$$\tilde{\nabla}_\eta \mathcal{L}(\eta) = \nabla_{\boldsymbol{m}} \mathcal{L}(\boldsymbol{m}).$$

In the case that $p_\theta(Z)$ is the Gaussian distribution $N(Z;\theta)$ with $\theta = \{\mu, \Sigma\}$ that satisfies the condition of the previous statement, $p_\theta(Z)$ has the other parameterization of $\eta = [\eta_1, \eta_2]$ and $\boldsymbol{m} = [m_1, m_2]$ as

$$\begin{aligned} \eta_1 = \Sigma^{-1}\mu \,, \quad \eta_2 = -\frac{1}{2}\Sigma^{-1} \\ m_1 = \mu \,, \quad m_2 = \mu\mu^T + \Sigma \end{aligned}$$

Then, $\nabla_{\boldsymbol{m}} \mathcal{L}(\boldsymbol{m})$ with respect to $\boldsymbol{m}$ is calculated as

$$\begin{aligned} \nabla_{m_1} \mathcal{L}(\boldsymbol{m}) &= \nabla_\mu \mathcal{L}(\theta) - 2\left[\nabla_\Sigma \mathcal{L}(\theta)\right]\mu \\ \nabla_{m_2} \mathcal{L}(\boldsymbol{m}) &= \nabla_\Sigma \mathcal{L}(\theta). \end{aligned}$$

Proof. See the proof of Theorem 1 in (Khan & Nielsen, 2018) and the proof of Appendix B.1 in (Khan & Lin, 2017). □

Lemma E.9. *Given the loss $\mathcal{L}(\theta)$ parameterized by Gaussian density parameter $\theta = \{\mu, \Sigma\}$ with $p_\theta(Z) = N(Z;\theta)$, the natural gradient update for μ and Σ can be obtained as*

$$\begin{aligned} \mu^{(t+1)} &= \mu^{(t)} + \beta \Sigma^{(t+1)} \nabla_\mu \mathcal{L}(\theta) \\ \Sigma^{-1(t+1)} &= \Sigma^{-1(t)} + \beta\big(-2\nabla_{\Sigma^{(t)}} \mathcal{L}(\theta)\big) \end{aligned}$$

where (t) denotes the t-th iteration and β is a learning rate.

Proof. See the proof of Theorem 1 in (Khan et al., 2018). □

Proposition E.10. *Let $\mu_q^{(t)}$ and $\sigma_q^{(t)}$ be the t-th iterated parameters of $N(\mu_q, \sigma_q^2)$ which is q-th component distribution for $q(S)$. The natural gradient of $\hat{\mathcal{L}}_J$ w.r.t μ_q and σ_q in log domain, i.e. $\tilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J$ and $\tilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J$, can be approximated as*

$$\tilde{\nabla}_{\log \mu_q} \hat{\mathcal{L}}_J \approx \left(\frac{\sigma_q^{(t+1)}}{\mu_q^{(t)}}\right)^2 \circ \nabla_{\log \mu_q} \hat{\mathcal{L}}_J, \qquad \tilde{\nabla}_{\log \sigma_q} \hat{\mathcal{L}}_J \approx \frac{1}{2}\nabla_{\log \sigma_q} \hat{\mathcal{L}}_J$$

under the condition $\left|\left(\frac{\sigma_q^{(t+1)}}{\mu_q^{(t)}}\right)^2 \circ \nabla_{\log \mu_q} \hat{\mathcal{L}}_J\right| < 1$ and $\left|\nabla_{\log \sigma_q} \hat{\mathcal{L}}_J\right| < 1$ in element-wise sense

Proof. When applying the result of Lemma E.9 with the diagonal covariance matrix $\Sigma = \mathrm{Diag}(\sigma_q^2)$, we obtain the following update rule as

$$\mu_{t+1} = \mu_t + \beta\sigma_{t+1}^2 \circ \nabla_\mu \mathcal{L} \,, \qquad \sigma_{t+1}^{-2} = \sigma_t^{-2} + \beta\big(-2\nabla_{\sigma^2}\mathcal{L}\big),$$

where $\circ$ denotes the element-wise product. After we transform the results of Lemma 6 in logarithm domain, we linearly approximate $\log \mu_{t+1}$ as

$$\begin{aligned}
\log \mu_{t+1} &= \log \Big(\mu_t + \beta \sigma_{t+1}^2 \circ \nabla_{\mu} \mathcal{L} \Big) \\
&= \log \mu_t + \log \Big(1 + \beta \mu_t^{-1} \sigma_{t+1}^2 \circ \nabla_{\mu} \mathcal{L} \Big) \\
&= \log \mu_t + \log \Big(1 + \beta \frac{\sigma_{t+1}^2}{\mu_t^2} \circ \nabla_{\log \mu} \mathcal{L} \Big) \\
&= \log \mu_t + \sum_{t=1}^{\infty} \frac{(-1)^{t+1}}{t} \Big(\beta \frac{\sigma_{t+1}^2}{\mu_t^2} \circ \nabla_{\log \mu} \mathcal{L} \Big)^t \\
&= \log \mu_t + \beta \frac{\sigma_{t+1}^2}{\mu_t^2} \circ \nabla_{\log \mu} \mathcal{L} + \mathcal{O}\Big(\big(\beta \frac{\sigma_{t+1}^2}{\mu_t^2} \circ \nabla_{\log \mu} \mathcal{L} \big)^2 \Big) \\
&\approx \log \mu_t + \beta \Big(\frac{\sigma_{t+1}^2}{\mu_t^2} \circ \nabla_{\log \mu} \mathcal{L} \Big).
\end{aligned}$$

In the third equality, we employ $\nabla_{\mu} \mathcal{L} = \frac{1}{\mu_t} \nabla_{\log \mu} \mathcal{L}$ obtained by chain rule. In the fourth equality, assuming the learning rate satisfies $|\beta| < 1$, we assume $\left| \frac{\sigma_{t+1}^2}{\mu_t^2} \nabla_{\log \mu} \mathcal{L} \right| < 1$ and employ the maclaurin series of $\log{(1+x)} = \sum_{t=1}^{\infty} \frac{(-1)^{t+1}}{t} x^t$ for $x \in (-1, 1]$. We approximate $\log \sigma_{t+1}^{-2}$ as

$$\begin{aligned}
\log \sigma_{t+1}^{-2} &= \log \Big(\sigma_t^{-2} - 2\beta \nabla_{\sigma^2} \mathcal{L} \Big) \\
&= \log \sigma_t^{-2} + \log \Big(1 - 2\beta \sigma_t^2 \nabla_{\sigma^2} \mathcal{L} \Big) \\
&= \log \sigma_t^{-2} + \log \Big(1 + \beta (-2\sigma_t^2) \frac{0.5}{\sigma_t^2} \nabla_{\log \sigma} \mathcal{L} \Big) \\
&= \log \sigma_t^{-2} + \sum_{t=1}^{\infty} \frac{(-1)^{t+1}}{t} \Big(- \beta \nabla_{\log \sigma} \mathcal{L} \Big)^t \\
&= \log \sigma_t^{-2} + -\beta \nabla_{\log \sigma} \mathcal{L} + \mathcal{O}\Big(\big(- \beta \nabla_{\log \sigma} \mathcal{L} \big)^2 \Big) \\
&\approx \log \sigma_t^{-2} + -\beta \nabla_{\log \sigma} \mathcal{L}.
\end{aligned}$$

In the third equality, we use $\nabla_{\sigma^2} \mathcal{L} = \frac{0.5}{\sigma_t^2} \nabla_{\log \sigma} \mathcal{L}$. In the fourth equality, we assume $\left| \nabla_{\log \sigma} \mathcal{L} \right| < 1$ and apply the maclaurin series of $\log{(1+x)}$ as well. Considering the iterative update rule of $\log \mu_t$ and $\log \sigma_t$ with the learning rate β, we can think of the approximate natural gradient for the $\log \mu$ and $\log \sigma$ as

$$\widetilde{\nabla}_{\log \mu} \mathcal{L} \approx \frac{\sigma_{t+1}^2}{\mu_t^2} \nabla_{\log \mu} \mathcal{L} \,, \qquad \widetilde{\nabla}_{\log \sigma} \mathcal{L} \approx \frac{1}{2} \nabla_{\log \sigma} \mathcal{L}.$$

□

F. Limitations

- We observe that employing the weighted sampling of Proposition 3.3 for the SVSS directly might degrade the parameter inference by SVSS; when one of weight parameters is extremely larger than other weight parameters, i.e. $w_q \gg w_1, .., w_{q-1}, w_{q+1}, .., w_Q$, the majority of spectral points are sampled from only a single mixture component, i.e., m_q=M, and m_1=..=m_{q-1}=m_{q+1}=..=m_Q=0. This hinders the parameters of other mixture components from being trained. Thus, we use the practical implementation for the weighted sampling approach as described in Appendix A.2. Although the proposed strategy is proven to be effective empirically, we have not provided the theoretical foundation to justify the modification.

- We observe that employing the natural gradient of Proposition 3.4 accelerates the convergence of the parameter estimation due to the effective use of the local geometric information using the Fisher Information matrix. However, the optimized parameters do not occasionally produce the robust prediction performance on the test set; as more iterations are allowed for training the parameters by SVSS-Ng or SVSS-WsNg, the RMSE and MNLL for the prediction on test set tend to be degraded compared to SVSS or SVSS-Ws, respectively. In the future study, we plan to develop a more robust natural gradient-based optimization scheme so that the trained parameters enable the model to predict the test set accurately.

- In the large-scale UCI benchmark regression task, we observe that the exact prediction using $K_{SM}(X, X)$ with the parameters estimated by SVSS and its variants shows the comparable test RMSE compared to the VFE using SM kernel, the benchmark approximate inference for GP. However, its MNLL on test set, that represents the quality of predictive uncertainty estimation, tends to be slightly degraded to that of VFE. This problem should be tackled so that the SVSS replaces the VFE by employing the advantage of the SVSS that using SVSS takes less training time compared to VFE generally.