Scalable Variational Gaussian Processes via Harmonic Kernel Decomposition

Shengyang Sun [1 2] **Jiaxin Shi** [3] **Andrew Gordon Wilson** [4] **Roger Grosse** [1 2]

Abstract

We introduce a new scalable variational Gaussian process approximation which provides a high fidelity approximation while retaining general applicability. We propose the harmonic kernel decomposition (HKD), which uses Fourier series to decompose a kernel as a sum of orthogonal kernels. Our variational approximation exploits this orthogonality to enable a large number of inducing points at a low computational cost. We demonstrate that, on a range of regression and classification problems, our approach can exploit input space symmetries such as translations and reflections, and it significantly outperforms standard variational methods in scalability and accuracy. Notably, our approach achieves state-of-the-art results on CIFAR-10 among pure GP models.

1. Introduction

Gaussian Processes (GPs) (Rasmussen & Williams, 2006) are flexible Bayesian nonparametric models which enable principled reasoning about distributions of functions and provide rigorous uncertainty estimates (Srinivas et al., 2010; Deisenroth & Rasmussen, 2011). Unfortunately, exact inference in GPs is impractical for large datasets because of the $\mathcal{O}(N^3)$ computational cost (for a dataset of size N). To overcome the computational roadblocks, sparse Gaussian processes (Snelson & Ghahramani, 2006; Quinonero-Candela & Rasmussen, 2005) use M inducing points to approximate the kernel function, reducing the computational cost to $\mathcal{O}(NM^2 + M^3)$. However, these approaches are prone to overfitting since all inducing points are hyperparameters. Sparse variational Gaussian Processes (SVGPs) (Titsias, 2009; Hensman et al., 2015) offer an effective protection against overfitting by framing a posterior approximation using the inducing points and optimizing them with variational inference. Still, the $\mathcal{O}(M^3)$ complexity prevents SVGPs from scaling beyond a few thousand inducing points, creating difficulties in improving the quality of approximation.

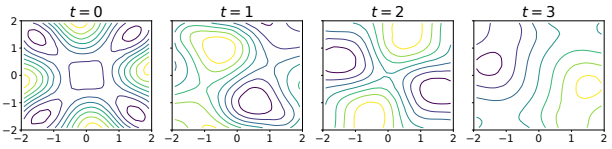

Figure 1. Visualizing the harmonic kernel decomposition. We decompose a 2-dimensional RBF kernel as $k = \sum_{t=0}^{3} k_t$ using the symmetry group of 90^o rotations. We plot the real part of random functions sampled from each $\mathcal{GP}(0, k_t)$. Notice that $\mathcal{GP}(0, k_0)$ is invariant to 90^o rotations; $\mathcal{GP}(0, k_1)$ takes opposite values under 180^o rotations; $\mathcal{GP}(0, k_2)$ is invariant to 180^o rotations but has opposite values under 90^o rotations.

Several approaches impose structure on the inducing points to increase the approximation capacity. Structured kernel interpolation (SKI) (Wilson & Nickisch, 2015; Wilson et al., 2015) approximates the kernel by placing inducing points over a Euclidean grid and exploiting fast structured matrix operations. SKI can use millions of inducing points, but is limited to low-dimensional problems because the grid size grows exponentially with the input dimension. Other approaches define approximate posteriors using multiple sets of inducing points. Cheng & Boots (2017); Salimbeni et al. (2018) propose to decouple the inducing points for modelling means and covariances, leading to a linear complexity with respect to the number of mean inducing points. SOLVE-GP (Shi et al., 2020) reformulates a GP as the sum of two orthogonal processes and uses distinct groups of inducing points for each; this has the benefit of improving the approximation at a lower cost than standard SVGPs.

In this paper, we introduce a more scalable variational approximation for GPs via the proposed *harmonic kernel decomposition (HKD)*, which decomposes the kernel as a sum of orthogonal kernels, $k(\mathbf{x}, \mathbf{x}') = \sum_{t=0}^{T-1} k_t(\mathbf{x}, \mathbf{x}')$, using Fourier series. The HKD reformulates the Gaussian process into an additive GP, where each subprocess models a Fourier component of the function (see Figure 1 for a visualization). We then propose the Harmonic Variational Gaussian Process (HVGP), which uses separate sets of inducing points for the subprocesses. Compared to a standard variational approximation, HVGPs allow us to use a large number of

[1]University of Toronto [2]Vector Institute [3]Microsoft Research New England [4]New York University. Correspondence to: Shengyang Sun <ssy@cs.toronto.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

inducing points at a much lower computational cost. Moreover, HVGPs have an advantage over SKI in that they allow trainable inducing points. Finally, unlike the SOLVE-GP whose decomposition involves only two subprocesses, our HVGP is easily applicable to multiple subprocesses and can be computed efficiently with the discrete Fourier transform.

Empirically, we demonstrate the scalability and general applicability of HVGPs through a range of problems and models including using RBF kernels for modelling earth elevations and using convolutional kernels for image classification. In these experiments, HVGPs significantly outperform standard variational methods by exploiting the input-space symmetries such as translations and reflections. Our model can further exploit parallelism to achieve minimal wall-clock overhead when using 8 groups of inducing points. In CIFAR-10 classification, we show that our method can be integrated with deep convolutional structures to achieve state-of-the-art results for GPs.

2. Background

2.1. Discrete Fourier Transform

Fourier analysis (Baron Fourier, 1878) studies the representation of functions as sums or integrals of sinusoids. For an integrable function f on $\mathbb{R}^d$, its Fourier transform is defined as

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} e^{-2\pi i \boldsymbol{\omega}^\top \mathbf{x}} f(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{1}$$

where $\hat{f}(\boldsymbol{\omega}) \in \mathbb{C}$. In kernel theory, Bochner's Theorem (Bochner, 1959) is a seminal result that uses the Fourier transform to establish a bijection between stationary kernels and positive measures in the spectral domain.

Fourier analysis can be performed over finite sequences as well, via the discrete Fourier transform (DFT) (Cooley et al., 1969). Specifically, given a sequence $\mathbf{x} = [\mathbf{x}_0, ..., \mathbf{x}_{T-1}]^\top$, DFT computes the sequence $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_0, ..., \hat{\mathbf{x}}_{T-1}]^\top$, with

$$\hat{\mathbf{x}}_t = \frac{1}{T} \sum_{s=0}^{T-1} \mathbf{x}_s e^{-i\frac{2\pi t s}{T}},\ t = 0, ..., T-1. \tag{2}$$

Let $\mathbf{F} := [\frac{1}{T} e^{-i\frac{2\pi t s}{T}}]_{t,s=0}^{T-1} \in \mathbb{C}^{T\times T}$ denote the DFT matrix. The DFT can be represented in vector form as $\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$, which naturally leads to the inverse DFT: $\mathbf{x} = \mathbf{F}^{-1}\hat{\mathbf{x}}$.

More generally, if $\mathbf{X} \in \mathbb{C}^{T_1\times\cdots\times T_k}$ is a tensor, the multidimensional DFT computes the tensor $\hat{\mathbf{X}} \in \mathbb{C}^{T_1\times\cdots\times T_k}$,

$$\hat{\mathbf{X}}[t_1, ..., t_k] = \frac{1}{T} \sum_{s_1,...,s_k} X[s_1, ..., s_k] \prod_{j=1}^{k} e^{-i\frac{2\pi t_j s_j}{T_j}},$$

where $T = \prod_j T_j$. The DFT matrix is then a tensor product of one-dimensional DFT matrices.

2.2. Gaussian Processes

Given an input domain $\mathcal{X}$, a mean function m, and a kernel function $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, the Gaussian process (Rasmussen & Williams, 2006) $\mathcal{GP}(m, k)$ is a distribution over functions $\mathcal{X} \to \mathbb{R}$. For any finite set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathcal{X}$, the function values $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N)]^\top$ have a multivariate Gaussian distribution:

$$\mathbf{f} \sim \mathcal{N}(m(\mathbf{X}), \mathbf{K}_{\mathbf{ff}}),$$

where $m(\mathbf{X}) = [m(\mathbf{x}_1), ..., m(\mathbf{x}_N)]^\top$, and $\mathbf{K}_{\mathbf{ff}} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^N$. For simplicity we assume $m(\cdot) = 0$ throughout the paper. The observations y are modeled with a density $p(y|f(\mathbf{x}))$, often taken to be Gaussian in the regression setting: $y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$. Let $(\mathbf{X}, \mathbf{y})$ be a training set of size N. The posterior distribution $p(\mathbf{f}^\star|\mathbf{y})$ under a Gaussian observation model is

$$\mathcal{N}(\mathbf{K}_{\star\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{\star\star} - \mathbf{K}_{\star\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}\star}),$$

where $\mathbf{f}^\star = f(\mathbf{X}^\star)$ are function values at test locations. Unfortunately, computing the posterior mean and covariance requires inverting the kernel matrix, an $\mathcal{O}(N^3)$ computation.

Sparse variational Gaussian processes (SVGPs) (Titsias, 2009; Hensman et al., 2013) use inducing points for scalable GP inference. Let $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_M]^\top \in \mathbb{R}^{M\times d}$ be M inducing locations, and $\mathbf{u} = f(\mathbf{Z})$. SVGPs consider an augmented joint likelihood, $p(f(\cdot), \mathbf{u}) = p(f(\cdot)|\mathbf{u})p(\mathbf{u})$, and a variational approximation $q(f(\cdot), \mathbf{u}) = p(f(\cdot)|\mathbf{u})q(\mathbf{u})$, where $q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$ is a parameterized multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\mathbf{S}$. The variational approximation is optimized by maximizing the variational lower bound:

$$\mathcal{L} := \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f}, \mathbf{X})] - \mathrm{KL}\left(q(\mathbf{u})||p(\mathbf{u})\right). \tag{3}$$

Since $\log p(\mathbf{y}|\mathbf{f}, \mathbf{X}) = \sum_{i=1}^N \log p(\mathbf{y}_i|f(\mathbf{x}_i))$ admits stochastic optimization, SVGPs reduce the computational cost to $\mathcal{O}(M^3 + M^2B)$, where B is the minibatch size.

3. Harmonic Kernel Decomposition

In this section, we introduce kernel Fourier series and use them to form the harmonic kernel decomposition. All proofs can be found in Appendix D.2.

3.1. Kernel Fourier Series

We first propose a general method for representing a kernel as a sum of functions. The idea is based on the DFT (see Sec. 2.1). Let $k : \mathcal{X}\times\mathcal{X} \to \mathbb{C}$ be a positive definite kernel. To apply the DFT, we fix the first input $\mathbf{x}$, and construct a finite sequence of kernel values using a transformation $G : \mathcal{X} \to \mathcal{X}$ that applies to the second input:

$$[k(\mathbf{x}, G^0(\mathbf{x}')), k(\mathbf{x}, G^1(\mathbf{x}')), ..., k(\mathbf{x}, G^{T-1}(\mathbf{x}'))], \tag{4}$$

where $G^0(\mathbf{x}) := \mathbf{x}$ and $G^t := G \circ G^{t-1}$. Note that in signal processing, the sequence under DFT usually contains equally-spaced samples along the time domain. Here, we adopt a more general form by using G to exploit symmetries in the input domain.

Definition 3.1 (Kernel Fourier Series)**.** We define T complex-valued functions $k_t : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$, $t = 0, \ldots, T-1$ using the DFT of the sequence (4):

$$k_t(\mathbf{x}, \mathbf{x}') = \sum_{s=0}^{T-1} \mathbf{F}_{t,s} k(\mathbf{x}, G^s(\mathbf{x}')), \tag{5}$$

where $\mathbf{F}$ is the DFT matrix (see Sec. 2.1). The *kernel Fourier series* of $k(\mathbf{x}, G^s(\mathbf{x}'))$ is given by the inverse DFT:

$$k(\mathbf{x}, G^s(\mathbf{x}')) = \sum_{t=0}^{T-1} \mathbf{F}^{-1}_{s,t} k_t(\mathbf{x}, \mathbf{x}'). \tag{6}$$

The inverse DFT matrix is $\mathbf{F}^{-1} = T\mathbf{F}^H$, where $\cdot^H$ is the conjugate transpose.

Given the positive definiteness of k, it is tempting to ask whether k_t is also a (complex-valued) kernel. In the next section, we will study the conditions when this holds and use it to form an orthogonal kernel decomposition.

3.2. Harmonic Kernel Decomposition

We first introduce the following definitions.

Definition 3.2 (T-Cyclic Transformation)**.** A function $G : \mathcal{X} \to \mathcal{X}$ is T-cyclic if T is the smallest integer such that,

$$\forall \mathbf{x} \in \mathcal{X},\ G^T(\mathbf{x}) := \overbrace{G \circ \cdots \circ G}^{T}(\mathbf{x}) = \mathbf{x}. \tag{7}$$

In group theory, $\{G^0, G^1, \ldots, G^{T-1}\}$ forms a cyclic group of order T, and G is the generator of this group. Interestingly, given a T-cyclic G, multiplying k_t with $e^{i\frac{2\pi t}{T}}$ corresponds to a shift by G in the second input.

Proposition 3.3 (Shift)**.** *For any* $0 \le t \le T-1$*,*

$$k_t(\mathbf{x}, G(\mathbf{x}')) = e^{i\frac{2\pi t}{T}} k_t(\mathbf{x}, \mathbf{x}'). \tag{8}$$

From Proposition 3.3 we have $k_0(\mathbf{x}, G(\mathbf{x}')) = k_0(\mathbf{x}, \mathbf{x}')$, and when T is even, $k_{T/2}(\mathbf{x}, G^2(\mathbf{x}')) = k_{T/2}(\mathbf{x}, \mathbf{x}')$. This property is illustrated in Figure 1.

Definition 3.4 (G-Invariant kernels)**.** A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ is G-invariant if,

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X},\ k(G(\mathbf{x}), G(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}'). \tag{9}$$

For example, a polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^t$ is invariant to the rotation transformations.

Theorem 3.5 (Harmonic Kernel Decomposition)**.** *Let G be a T-cyclic transformation, and k be a G-invariant kernel. Then, the following decomposition holds:*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{t=0}^{T-1} k_t(\mathbf{x}, \mathbf{x}'), \tag{10}$$

where $k_t,\ t = 0, \ldots, T-1$ are defined as in Eq. (5). Moreover, for any $0 \le t \le T-1$, k_t is a Hermitian kernel.

The equation follows from the kernel Fourier series of $k(\mathbf{x}, G^0(\mathbf{x}'))$ by noticing that $\mathbf{F}^{-1}_{0,:} = \mathbf{1}$. To prove that k_t is a kernel, we show that $k_t(\mathbf{x}, \mathbf{x}') = \mathbf{F}^H_{:,t} \mathbf{K}(\mathbf{x}, \mathbf{x}') \mathbf{F}_{:,t}$, where $\mathbf{K}(\mathbf{x}, \mathbf{x}') = [k(G^{s_1}(\mathbf{x}), G^{s_2}(\mathbf{x}'))]^{T-1}_{s_1, s_2 = 0}$,

Besides the kernel sum decomposition, we further show that the kernels $k_t,\ t = 0, ..., T-1$ are orthogonal to each other, as identified by the following lemma.

Lemma 3.6 (Orthogonality)**.** *For any $0 \le t_1 \ne t_2 \le T-1$, let $\mathcal{H}_k, \mathcal{H}_{k_{t_1}}, \mathcal{H}_{k_{t_2}}$ be the RKHSs corresponding to the kernel k, k_{t_1}, k_{t_2}, respectively. Then for any $f \in \mathcal{H}_{k_{t_1}}$ and $g \in \mathcal{H}_{k_{t_2}}$, $\langle f, g \rangle_{\mathcal{H}_k} = 0$.*

Because the $\mathcal{H}_k$ inner product of $f \in \mathcal{H}_1, g \in \mathcal{H}_2$ is always zero, we immediately obtain that the RKHSs for k_t are disjoint except for the function $f \equiv 0$.

Proposition 3.7 (Disjoint)**.** *For any $0 \le t_1 \ne t_2 \le T-1$,*

$$\mathcal{H}_{k_{t_1}} \cap \mathcal{H}_{k_{t_2}} = \{0\}, \tag{11}$$

The kernel decomposition and orthogonality translate to the RKHS orthogonal sum decomposition as follows:

Theorem 3.8 (Orthogonal Sum Decomposition of RKHS)**.** *The RKHS $\mathcal{H}_k$ admits an orthogonal sum decomposition,*

$$\mathcal{H}_k = \bigoplus_{t=0}^{T-1} \mathcal{H}_{k_t}. \tag{12}$$

Specifically, for any function $f \in \mathcal{H}_k$, f has the unique decomposition $f = \sum_{t=0}^{T-1} f_t,\ f_t \in \mathcal{H}_{k_t}$, and $f_t(\mathbf{x}) = \langle f, k_t(\mathbf{x}, \cdot) \rangle_{\mathcal{H}_k}$. The RKHS norm of f is equal to

$$\|f\|^2_{\mathcal{H}_k} = \sum_{t=0}^{T-1} \|f_t\|^2_{\mathcal{H}_{k_t}}. \tag{13}$$

3.3. Examples of Harmonic Kernel Decomposition

The HKD relies on the (G, k) pair where G is a T-cyclic transformation and k is a kernel invariant to G. In this section we provide examples of such kernels and transformations. Notably, all inner-product kernels and stationary kernels[1] can be decomposed with the HKD when paired with an appropriately chosen G.

[1]This includes, e.g., polynomial, Gaussian, Matérn, periodic, arccosine, and rational quadratic kernels.

An opening example. We start with a toy example to illustrate the kernel decomposition. Let $k(\theta, \theta') = e^{-i(\theta-\theta')} + e^{-2i(\theta-\theta')}$ for $\theta \in [0, 2\pi)$. The transformation $G(\theta) = (\theta + \frac{2\pi}{T}) \mod 2\pi$ is T-cyclic. Based on the kernel Fourier series, we obtain $k_1(\theta, \theta') = e^{-i(\theta-\theta')}$, $k_2(\theta, \theta') = e^{-2i(\theta-\theta')}$, and $k_t = 0$ otherwise. We observe that the RKHS of k_2 contains periodic functions with basic period π, while the RKHS of k_1 contains functions with basic period 2π. In this way, our method decomposes $\mathcal{H}_k$ into orthogonal RKHSs.

Inner-product kernels are kernels of the form,

$$k(\mathbf{x}, \mathbf{x}') = h(\mathbf{x}^H\mathbf{x}, \mathbf{x}^H\mathbf{x}', \mathbf{x}'^H\mathbf{x}'), \tag{14}$$

where the function h ensures that k is positive semi-definite (Hofmann et al., 2008). For a matrix $\mathbf{R} \in \mathbb{C}^{d\times d}$, which is unitary (i.e. $\mathbf{R}\mathbf{R}^H = \mathbf{I}$), the kernel k is G-invariant:

$$\begin{aligned} k(G(\mathbf{x}), G(\mathbf{x}')) &= h(\mathbf{x}^H\mathbf{R}^H\mathbf{R}\mathbf{x}, \mathbf{x}^H\mathbf{R}^H\mathbf{R}\mathbf{x}', \mathbf{x}'^H\mathbf{R}^H\mathbf{R}\mathbf{x}') \\ &= h(\mathbf{x}^H\mathbf{x}, \mathbf{x}^H\mathbf{x}', \mathbf{x}'^H\mathbf{x}') = k(\mathbf{x}, \mathbf{x}'), \end{aligned}$$

Examples include reflections, rotations, and permutations. Moreover, if $\underbrace{\mathbf{R}\cdots\mathbf{R}}_{T} = \mathbf{I}$, the mapping $G(\mathbf{x}) = \mathbf{R}\mathbf{x}$ is T-cyclic.

Stationary kernels are kernels of the form,

$$k(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x} - \mathbf{x}'), \tag{15}$$

where κ is a positive-type function (Berlinet & Thomas-Agnan, 2011). Let $T = 2$ and $G(\mathbf{x}) = -\mathbf{x}$; then G is T-cyclic. For real kernels whose $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$, the kernel is symmetric (i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$). Then k is G-invariant:

$$k(G(\mathbf{x}), G(\mathbf{x}')) = \kappa(\mathbf{x}' - \mathbf{x}) = k(\mathbf{x}', \mathbf{x}) = k(\mathbf{x}, \mathbf{x}').$$

Similarly, we can prove that inner-product kernels are negation-invariant. Stationary kernels are also invariant to the T-cyclic transformation $G_i(\mathbf{x}) = \mathbf{x} + \frac{2\pi}{T}\mathbf{e}_i$ on a multidimensional torus $\mathbb{T}^d = \underbrace{\mathbb{S}^1 \times \cdots \times \mathbb{S}^1}_{d}$, where $\mathbb{S}^1$ represents a one-dimensional circle.

3.4. Resolving Complex-Valued Kernels

From $\mathbf{F} = [\frac{1}{T}e^{-i\frac{2\pi ts}{T}}]_{t,s=0}^{T-1}$, we know that the DFT introduces complex values whenever $T > 2$. Therefore, k_t is Hermitian but not necessarily real-valued. For example, the decomposition in Fig. 1 introduces imaginary values when $t = 1, 3$. Since k is real-valued, we can obtain a real-valued kernel decomposition by pairing up k_ts. Specifically, for a T-cyclic transformation G, we have

$$(k_t + k_{T-t})(\mathbf{x}, \mathbf{x}') = \frac{2}{T}\sum_{s=0}^{T-1} \cos(\frac{2\pi ts}{T}) k(\mathbf{x}, G^s(\mathbf{x}')),$$

In this way we obtain a real-valued decomposition with $\lfloor T/2 \rfloor + 1$ kernels.

3.5. Multi-Way Transformations

Previously we considered the Fourier series along one transformation orbit: $k(\mathbf{x}, G^0(\mathbf{x}')), ..., k(\mathbf{x}, G^{T-1}(\mathbf{x}'))$. We can extend it to multi-way transformations, akin to a multidimensional DFT. Let $T_1, ..., T_J \in \mathbb{N}$ and G_j be a T_j-cyclic transformation for $j = 1, ..., J$, respectively. We further assume that all transformations commute, i.e., $\forall 1 \le j_1, j_2 \le J$,

$$\forall \mathbf{x} \in \mathcal{X}, G_{j_1}(G_{j_2}(\mathbf{x})) = G_{j_2}(G_{j_1}(\mathbf{x})). \tag{16}$$

Due to commutativity, we can use the indices $(t_1, ..., t_J)$ to represent applying each G_j for t_j times,

$$G^{(t_1,...,t_J)}(\mathbf{x}) := G_1^{t_1} \cdots G_J^{t_J}(\mathbf{x}), \tag{17}$$

where $G := G_1 \otimes \cdots \otimes G_J$. Moreover, if a kernel k is G_j-invariant for all $j = 1, ..., J$, then k is G-invariant.

Letting $t = (t_1, ..., t_J)$ be a multi-index, we compute the J-way kernel Fourier series from a multidimensional DFT:

$$k_t(\mathbf{x}, \mathbf{x}') = \sum_{s=(0,...,0)}^{(T_1-1,...,T_J-1)} \prod_{j=1}^{J} \mathbf{F}^{(j)}_{t_j,s_j} k(\mathbf{x}, G^s(\mathbf{x}')),$$

where $\mathbf{F}^{(j)} \in \mathbb{C}^{T_j\times T_j}$ is the DFT matrix of the j-th transformation. Similar to Theorem 3.5, these k_ts also form an HKD of k.

Taking a two-dimensional RBF kernel as an example, we can check that it is invariant to negation along either dimension: $G_1([x_1, x_2]^\top) = [-x_1, x_2]^\top, G_2([x_1, x_2]^\top) = [x_1, -x_2]^\top$. Because G_1 and G_2 commute, this forms a 2-way transformation $G = G_1 \otimes G_2$, which corresponds to an HKD with $2 \times 2 = 4$ sub-kernels.

4. Harmonic Variational Gaussian Processes

In this section, we explore the implications of the HKD, and propose a scalable inference strategy for variational Gaussian processes. All proofs can be found in Sec D.3 in the appendix.

4.1. Variational Inference for Decomposed GPs

Given the kernel decomposition[2] $k = \sum_{t=0}^{T-1} k_t$, the Gaussian process can be represented in an additive formulation,

$$f = \sum_{t=0}^{T-1} f_t, \quad f_t \sim \mathcal{GP}(0, k_t). \tag{18}$$

For $t = 0, ..., T-1$, we introduce inducing points $\mathbf{Z}_t$ and denote by $\mathbf{u}_t := f_t(\mathbf{Z}_t)$ the inducing variables. Let p_t represent $\mathcal{GP}(0, k_t)$. We consider an augmented model,

$$f = \sum_{t=0}^{T-1} f_t, \quad p_t(f_t(\cdot), \mathbf{u}_t) = p_t(f_t(\cdot)|\mathbf{u}_t)p_t(\mathbf{u}_t), \tag{19}$$

[2] t can be a multi-index for multi-way transformations.

and define the variational posterior approximation as

$$f = \sum_{t=0}^{T-1} f_t, \qquad f_t(\cdot) \sim p_t(f_t(\cdot)|\mathbf{u}_t),\ \mathbf{u}_{0:T-1} \sim q(\mathbf{u}_{0:T-1}). \tag{20}$$

To understand how well this variational distribution approximates the true GP posterior, we compare it with a standard SVGP, for which the quality of approximation has been studied extensively by Burt et al. (2019). For simplicity, we focus our analysis on the case where inducing points are shared across all subprocesses: $\mathbf{Z}_0 = ... = \mathbf{Z}_{T-1} := \mathbf{Z}$, and we assume a complex-valued kernel decomposition without using the techniques in Section 3.4. Then, we demonstrate that Eq. (20) is equivalent to an SVGP with inducing points $\{G^t(\mathbf{Z})\}_{t=0}^{T-1}$.

Theorem 4.1. *Consider an SVGP with inducing points $\{G^t(\mathbf{Z})\}_{t=0}^{T-1}$. Let $\mathbf{v}_t := f(G^t(\mathbf{Z}))$ be the inducing variables and $\mathbf{V} := [\mathbf{v}_0, ..., \mathbf{v}_{T-1}]^\top \in \mathbb{C}^{T\times m}$. Suppose its variational distribution is*

$$q_{\text{svgp}}(f(\cdot), \mathbf{V}) = p(f(\cdot)|\mathbf{V})\mathcal{N}(\text{vec}(\mathbf{V})|\text{vec}(\mathbf{M}_v), \mathbf{S}_v),$$

where $\mathbf{M}_v \in \mathbb{C}^{T\times m}$, $\mathbf{S}_v \in \mathbb{C}^{Tm\times Tm}$ are the mean and covariance, respectively. Let $\mathbf{U} := [\mathbf{u}_0, ..., \mathbf{u}_{T-1}]^\top \in \mathbb{C}^{T\times m}$. Then, Eq. (20) and q_{svgp} have the same marginal distribution of $f(\cdot)$ if $q(\mathbf{u}_{0:T-1})$ is defined as

$$q(\text{vec}(\mathbf{U})) = \mathcal{N}(\text{vec}(\mathbf{F}^H\mathbf{M}_v), (\mathbf{I}\otimes\mathbf{F}^H)\mathbf{S}_v(\mathbf{I}\otimes\mathbf{F})).$$

The proof is based on showing the bijective linearity $\mathbf{U} = \mathbf{F}^H\mathbf{V}$. Since the theorem assumes shared inducing points, our variational approximation in Eq. (20) has a larger capacity than SVGPs with inducing points $\{G^t(\mathbf{Z})\}_{t=0}^{T-1}$. Therefore, if the inducing points $\{G^t(\mathbf{Z})\}_{t=0}^{T-1}$ match the input distribution well, our variational posterior can approximate the true posterior accurately.

4.2. Harmonic Variational Gaussian Processes

The additive GP reformulation in Eq. (18) ensures the independence between $\mathbf{u}_{0:T-1}$ in the prior, and we demonstrated the orthogonality of the decomposed RKHSs in Theorem 3.8. Thus it is tempting to modelling the variational posterior separately within each RKHS. Now we introduce the Harmonic Variational Gaussian Process (HVGPs), which enforces independence between $\mathbf{u}_t$ by letting $q(\mathbf{u}_{0:T-1}) = \prod_{t=0}^{T-1} q_t(\mathbf{u}_t)$. Then the variational posterior becomes

$$f = \sum_{t=0}^{T-1} f_t,\ q_t(f_t(\cdot), \mathbf{u}_t) = p_t(f_t(\cdot)|\mathbf{u}_t)q_t(\mathbf{u}_t). \tag{21}$$

In other words, HVGPs use a variational posterior independently for each GP. We set $q_t(\mathbf{u}_t) = \mathcal{N}(\boldsymbol{\mu}_t, \mathbf{S}_t)$ as Gaussians. In particular, if each q_t uses m inducing points, we term the model a $T \times m$ model. The variational posterior can be optimized by maximizing the ELBO,

$$\mathbb{E}_{q(\{f_t\}_{t=0}^{T-1})}[\log p(\mathbf{y}|\sum_{t=0}^{T-1} f_t, \mathbf{X})] - \sum_{t=0}^{T-1} \text{KL}\left(q_t(\mathbf{u}_t)||p_t(\mathbf{u}_t)\right).$$

where $q(\{f_t(\cdot)\}_{t=0}^{T-1}) := \prod_{t=0}^{T-1}\int p_t(f_t(\cdot)|\mathbf{u}_t)q_t(\mathbf{u}_t)d\mathbf{u}_t$.

How well does $\prod_{t=0}^{T-1} q_t(\mathbf{u}_t)$ approximate the optimal Gaussian variational posterior $q^\star(\mathbf{u}_{0:T-1})$? $q^\star$ has a covariance $\mathbf{S}^\star \in \mathbb{R}^{Tm\times Tm}$, while HVGPs induce block diagonal structures. Fortunately, we can show that $\mathbf{S}^\star$ is approximately block diagonal if the input distribution is invariant to G.

Theorem 4.2. *If the input distribution p is invariant to G, i.e., the random variable $\mathbf{x} \sim p$ and the random variable $G(\mathbf{x}), \mathbf{x} \sim p$ are identically distributed, then $\mathbf{S}^\star$ becomes block diagonal when the training size $N \to \infty$.*

Theorem 4.2 indicates that the independent variational distributions $\prod_{t=0}^{T-1} q_t(\mathbf{u}_t)$ in HVGPs accurately approximate $q^\star(\mathbf{u}_{0:T-1})$ when the input distribution is symmetric under the transformation G. The symmetry further makes it easy for $\{G^t(\mathbf{Z})\}_{t=0}^{T-1}$ to match the input distribution, which by Theorem 4.1 renders that the variational approximation in Eq. (20) with the optimal $q^\star(\mathbf{u}_{0:T-1})$ would be close to the true posterior.

We illustrate the gist with a *flip-mnist* problem, where each digit in the MNIST dataset is randomly flipped up-and-down or left-and-right. For a RBF kernel, we consider two variations of HVGPs in terms of the transformation: 1) Negation. We split input dimensions into two groups and negate them separately, resulting in a 4×100 model. 2) Flipping the image up-and-down or left-and-right, resulting in a 4×100 model. We compare them with SVGPs using $100, 400$ inducing points, shown in Figure 2. This experiment highlights the importance of matching the transformation G with the data distribution.

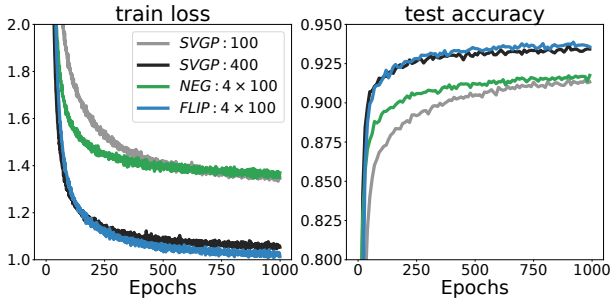

Figure 2. Flip-MNIST. We plot how *left*: train loss and *right*: test accuracy evolve with training. We observe the HVGP using *FLIP:* 4×100 perform similarly with the SVGP using 400 inducing points while the HVGP using *NEG:* 4×100 perform substantially worse.

4.3. Computational Cost

Assume that we have a J-way transformation, and each way is $\tilde{T}_j$-cyclic. After decomposition this results in $\tilde{T} = \prod_{j=1}^{J} \tilde{T}_j$ complex-valued kernels and subsequently, $T = \prod_{j=1}^{J}(\lfloor \tilde{T}_j/2 \rfloor + 1) \geq \tilde{T}/2^J$ real-valued kernels. Let $T \times m$ represent using m inducing points for each $t \in \{0, \ldots, T-1\}$, and assume the mini-batch size $B = \mathcal{O}(m)$.

Time Complexity. The computational cost boils down to the cost of computing $k_t(\mathbf{Z}_t, \mathbf{Z}_t)$ and the cost of variational inference. To compute $k_t(\mathbf{Z}_t, \mathbf{Z}_t)$, we need the kernel values $k(\mathbf{Z}_t, G^s(\mathbf{Z}_t))$ for $s = 0, ..., \tilde{T} - 1$. If we assume the cost of applying G is c_G, then computing $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ requires $\mathcal{O}(Tm \times \tilde{T}c_G + Tm^2 \times \tilde{T})$ operations. Variational inference costs $\mathcal{O}(Tm^3)$ time. Therefore, the overall complexity is $\mathcal{O}(Tm^3 + 2^J T^2 m^2 + 2^J T^2 m c_G)$. We note $2^J \leq T$, and for a single-way transformation, the cost simplifies to $\mathcal{O}(Tm^3 + T^2m^2 + T^2 m c_G)$. In contrast, a SVGP with Tm inducing points has the time complexity $\mathcal{O}(T^3m^3)$. Furthermore, HVGPs support straightforward parallelisms by locating computations of k_t on separate devices.

Space Complexity. For computing $k_t(\mathbf{Z}_t, \mathbf{Z}_t)$, we need the kernel values $k(\mathbf{Z}_t, G^s(\mathbf{Z}_t))$ for $s = 0, \ldots, \tilde{T}-1$, which implies the memory cost $\mathcal{O}(Tm^2 \times \tilde{T})$. Adding the $\mathcal{O}(Tm^2)$ memory for keeping variational approximations, the overall space complexity is, $\mathcal{O}(2^J T^2 m^2)$.

5. Related Works

The idea of applying Fourier analysis to kernels goes back at least to Bochner (1959). In machine learning, this led to a flowering of large-scale kernel methods based on random features (Rahimi & Recht, 2008; Yu et al., 2016; Dao et al., 2017). Bochner's theorem also allows designing stationary kernels by modeling a spectral density (Wilson & Adams, 2013; Samo & Roberts, 2015; Parra & Tobar, 2017; Benton et al., 2019). On hyperspheres, zonal kernels are the counterpart of stationary kernels. Their spectral decomposition is given by spherical harmonics (Thomson & Tait, 1888; Morimoto, 1998). Although closely related, none of these works have considered the discrete Fourier transform adopted in our method.

HVGPs share many similarities with the works that propose decoupled (Cheng & Boots, 2017; Salimbeni et al., 2018) and orthogonal (Shi et al., 2020) inducing points. In particular, Shi et al. (2020) is also based on an orthogonal decomposition of the kernel and uses distinct groups of inducing points for them. However, their decomposition involves matrix inversion while ours can be computed using fast Fourier transforms.

Structured Kernel Interpolation (SKI) (Wilson & Nickisch, 2015; Wilson et al., 2015; Evans & Nair, 2018; Izmailov et al., 2018) places inducing points on a grid, leading to a structured $\mathbf{K}_{\mathbf{uu}}$ that allows fast matrix-vector multiplications. For one-dimensional data, SKI exploits the Toeplitz structure of $\mathbf{K}_{\mathbf{uu}}$ generated by stationary kernels. They first embed the Toeplitz matrix into a circulant matrix $\mathbf{C}$, and use the fact that circulant matrices can be diagonalized by the DFT (Tee, 2007) to enable fast computations:

$$\mathbf{C} = \mathbf{F}^{-1}\mathrm{diag}(\mathbf{F}\mathbf{c})\mathbf{F}. \tag{22}$$

Here $\mathbf{F}$ is the DFT matrix, and $\mathbf{c}$ is the first column of $\mathbf{C}$. This equation highlights a connection with our HKD: If we let $\mathbf{C} = [k(G^{t_1}(\mathbf{x}'), G^{t_2}(\mathbf{x}))]_{t_1,t_2=0}^{T-1}$, then $\mathbf{c} = [k(\mathbf{x}, G^0(\mathbf{x}')), ..., k(\mathbf{x}, G^{T-1}(\mathbf{x}'))]$ is the sequence we constructed in Eq. (4), and the eigenvalues $\mathbf{F}\mathbf{c}$ recover our decomposition $[k_t(\mathbf{x}, \mathbf{x}')]_{t=0}^{T-1}$ by the discrete Fourier transform. In other words, our approach generalizes the structure of one-dimensional equally-spaced grids in SKI into arbitrary cyclic groups. Moreover, our method allows trainable inducing locations, which plays an important role in combating the curse of dimensionality.

Besides inducing points in the data space, a number of works have investigated inducing features in the frequency domain. However, these inducing features are either limited to specific kernels (Lázaro-Gredilla & Figueiras-Vidal, 2009; Hensman et al., 2017) or involve numerical approximations (Dutordoir et al., 2020; Burt et al., 2020). The implementation of Dutordoir et al. (2020) only supports data up to 8 dimensions.

Incorporating invariances with respect to input-space transformations into Gaussian processes is also investigated in a stream of works (Ginsbourger et al., 2016; Van der Wilk et al., 2019). Our work is orthogonal to them since we are not designing invariant models. Instead, we proposed a general inference method for GPs that can benefit from invariances in the data distribution. Relatedly, Solin & Särkkä (2020); Borovitskiy et al. (2020) studied Gaussian processes on Riemannian manifolds.

6. Experiments

We present empirical evaluations in this section. All results were obtained using NVIDIA Tesla P100 GPUs, except in Sec 6.3 we used NVIDIA Tesla T4. Code is available at `https://github.com/ssydasheng/Harmonic-Kernel-Decomposition`.

6.1. Earth Elevation

We adopt GPs to fit the ETOPO1 elevation data of the earth (Amante & Eakins, 2009). ETOPO1 bedrock models the Earth's elevations from the bedrock surface underneath the ice sheets. A location is represented by the (longitude,

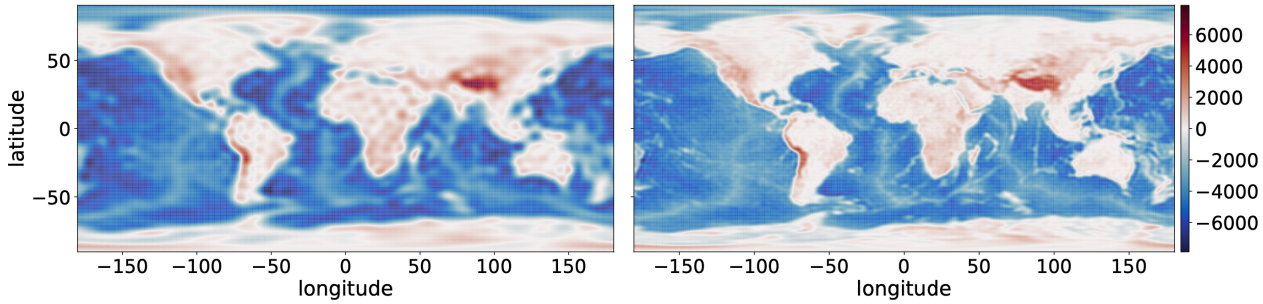

Figure 3. Predictive means of Earth elevations. We compare *left*: SVGP (M=1000) and *right*: HVGP (13×1000). The transformation in the HVGP moves each point eastwards by 15° longitude. We observe that the HVGP (13×1000) fits the data more finely.

Model	Test RMSE	Test NLL	Time
SKI	**0.145**	1.313	1.18h
1k	0.252	0.040	0.38h
3k	0.208	-0.146	2.47h
5k	0.196	-0.203	8.70h
7x1k	0.189	-0.246	1.55h
13x1k	**0.177**	**-0.314**	4.25h

Table 1. Test performances on Earth elevations.

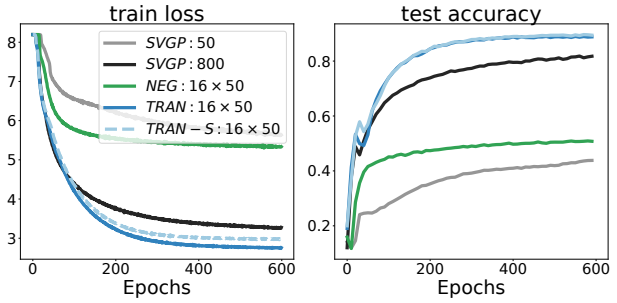

Figure 4. Translate-MNIST. We plot how *left: train loss* and *right: test acc* evolve with training. We compare $16 \times M$ HVGPs with 1) *NEG*: negations; 2) *TRAN*: translations; 3) *TRAN-S*: translations with shared inducing points. We observe the HVGPs using translations even outperform the SVGP with 800 inducing points while the HVGP using negations performs substantially worse. Moreover, though the HVGP with *TRAN-S* has only 50 trainable inducing points, it performs similarly with the *TRAN* model.

latitude) pair, where longitude $\in [-180, 180]$ and latitude $\in [-90, 90]$. We build the dataset by choosing a location every 0.1 degrees of longitude and latitude, resulting in $6,480,000$ data points. The dataset is randomly split for 72% training, 8% validating, and 20% testing. We use a three dimensional RBF kernel between the Euclidean coordinates of any two (longitude, latitude) locations. Because moving two locations eastwards by the same amount of longitudes preserves their Euclidean distance, the kernel is invariant to the T-cyclic longitude translation:

$$G([\text{lon}, \text{lat}]^\top) = [\text{lon} + \frac{360}{T}, \text{lat}]^\top, \tag{23}$$

We set the period $T = 12$ and $T = 24$, so that G moves a point eastwards by 30 and 15 degrees, respectively. Then we resolve Hermitian kernels to obtain $\lfloor 12/2 \rfloor + 1 = 7, \lfloor 24/2 \rfloor + 1 = 13$ real-valued kernels following Sec 3.4.

We compare SVGPs with $1k, 3k, 5k$ inducing points and the HVGPs with $7 \times 1k, 13 \times 1k$ inducing points. We parallelize HVGPs using 4 GPUs, while SVGPs use only 1 GPU since it cannot be easily parallelized. All models are optimized using the Adam optimizer with learning rate 0.01 for 100K iterations. We also compare with SKI (Wilson & Nickisch, 2015). SKI runs into an out-of-memory error because of the large dataset, so we train it using a random $600,000$ subset of the training data. The performances are shown in Table 3 and the predictive means are visualized in Figure 3. From both the table and the figure, we observe using more inducing points in variational GP models fits the dataset substantially better. Moreover, because of the decomposed structures and the parallelisms, HVGPs use more inducing points but run faster. In comparison, SKI uses $1M$ inducing points and achieves the best RMSE, but its NLL is much worse compared to variational GPs.

6.2. Translate-MNIST

The Elevation experiment uses one-way translations in HVGPs. In this section, we consider two-way translations for the *Translate-MNIST* dataset. The dataset is obtained by translating every MNIST image leftwards and downwards by random numbers of pixels. We use an RBF kernel with shared lengthscales. The HVGP uses a 2-way transformation G by translating the image leftwards or downwards by 4 pixels. Since the MNIST images are of size 28×28, G is $(7, 7)$ cyclic. After resolving Hermitian kernels, we arrive at $(1 + \lfloor 7/2 \rfloor) \times (1 + \lfloor 7/2 \rfloor) = 16$ groups.

We compare the 16×50 translation HVGP with SVGPs using 50 and 800 inducing points. We further consider a

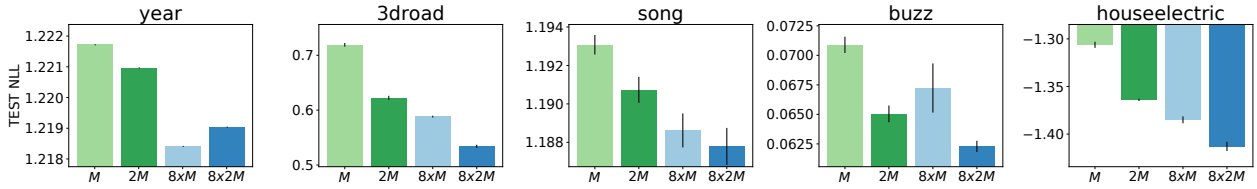

Figure 5. Test negative log-likelihoods on regression benchmarks. We compare using $M, 2M, 8 \times M, 8 \times 2M$ for $M = 1000$. We observe that the $8 \times M, 8 \times 2M$ outperform the standard M and $2M$ inducing points for the most datasets.

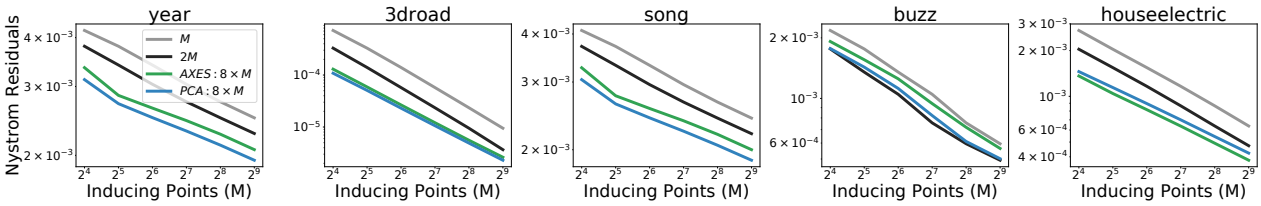

Figure 6. Nyström approximation errors measured by $\mathrm{trace}(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{K_{uf}})$. We compare SVGPs using $M, 2M$ inducing points, and HVGPs with $8 \times M$ inducing points. For the transformation in HVGPs, we include both the negation along the standard axes and along the principal components. We observe that negations along PCA directions usually outperform negations along standard axes, by demonstrating smaller approximation errors. And both variations of HVGPs perform better than the SVGPs. We also observe a consistency between the Nyström approximation errors and the regression performances. For example, the SVGP $(2M)$ performs better than the HVGP $(8 \times M)$ for the *buzz* dataset in Figure 5, and this is similarly reflected by the Nyström approximation errors.

variation of the HVGP by sharing the inducing points $\mathbf{Z}$ as in Theorem 4.1. We also include a 16×50 HVGP with 4-way negations whose transformation does not match the input-space distribution. We optimize all models using the Adam optimizer with learning rate 0.001 for 100K iterations. The results are shown in Figure 4.

6.3. Regression Benchmarks

We also evaluate our method on standard regression benchmarks, whose training data sizes range from 200 thousand to 1 million. Following Wang et al. (2019), we use the Matérn 3/2 kernel with shared lengthscales. We consider the J-way composition of negations. Specifically, we conduct negations over PCA directions. We split the PCA directions into J subsets, and applying negations over these subsets results in a $2^J \times m$ model. A visual comparison between the negation along axes and the negation along principal directions is shown in Figure 7.

We compare SVGPs using M and $2M$ inducing points with HVGPs using $8 \times M$ and $8 \times 2M$ inducing points for $M = 1000$. For HVGPs, we use 3-way negations over PCA directions, and we use 8 GPUs to place the computations of each GP in parallel. The results for negative log likelihoods (NLLs) are reported in Figure 5. We also report the root mean squared error (RMSE) performances in Figure 10 in the Appendix.

In Figure 8 we plot the evolution of the test negative log-

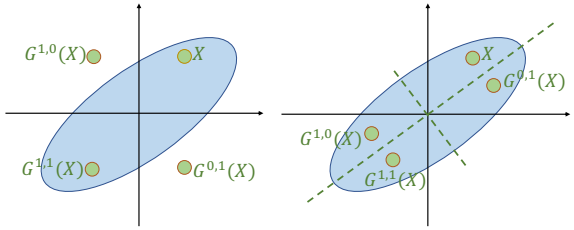

Figure 7. Negation along axes (*left*) and along principled directions (*right*). The shaded area represents the input distribution. We observe that $G^{1,0}(\mathbf{x}), G^{0,1}(\mathbf{x})$ are out of the data distribution when transforming along axes. In comparison, when transforming along PCA directions, the whole orbit is in-distribution.

likelihoods during training, and the training time per iteration, for the *3droad* dataset. In Figure 8, we observe that using more inducing points enables learning the dataset more quickly and converging to a better minima. Furthermore, due to the benefit of parallelism, the $8 \times M$ HVGP has comparable running time compared to the standard SVGP with M inducing points. And it is much faster compared to the SVGP with $2M$ inducing points in spite of the improved performance. Moreover, the computational bottleneck of SVGPs lies in the Cholesky decomposition, which does not support easy parallelism.

The performance of variational GPs relies largely on how well can the inducing points summarize the dataset, which

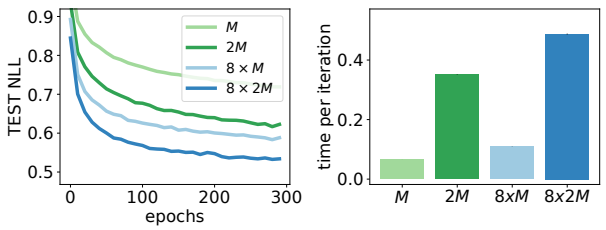

Figure 8. Test negative log-likelihoods during training and the training time per iteration for the *3droad* dataset.

can be measured by the accuracy of Nyström approximation $\mathbf{K_{ff}} \approx \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}$ (Drineas et al., 2005; Titsias, 2009; Burt et al., 2019). We compare the Nyström approximation errors with all methods, using the trace norm defined as $\mathrm{trace}(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}})$. For HVGPs, the trace norm is computed as

$$\sum_{t=0}^{T-1} \mathrm{trace}(\mathbf{K}_{t,\mathbf{ff}} - \mathbf{K}_{t,\mathbf{fu}}\mathbf{K}_{t,\mathbf{uu}}^{-1}\mathbf{K}_{t,\mathbf{uf}}). \tag{24}$$

where we use $\mathbf{K}_{t,\cdot}$ to represent the kernel k_t. For each dataset, we randomly sample 3000 points as $\mathbf{X}$ and use a Matérn 3/2 kernel whose lengthscales are set based on the median heuristic. We initialize the inducing points $\mathbf{Z}$ using K-means and optimize them to minimizing the trace error. We compare SVGPs, HVGPs with negations along axes, and HVGPs with negations along principled directions. The results are shown in Figure 6.

6.4. CIFAR-10 Classification

In this subsection we conduct experiments on the CIFAR10 classification problem using deep convolutional Gaussian processes (Blomqvist et al., 2019; Dutordoir et al., 2019), which combine the deep GP with the convolutional inducing features (Van der Wilk et al., 2017). Following the settings in Shi et al. (2020), we compare HVGP with SVGP on both one-layer and multi-layer convolutional GPs.

For HVGPs we use the negation transformations on the inducing filters $G(\mathbf{z}) = -\mathbf{z}$. We compare HVGPs using 2xM, 4xM inducing points with SVGPs using M, 2M inducing points. For the HVGP (4xM), we use 4 GPUs to achieve parallelism. We also compare with the 2-way decomposed model in Shi et al. (2020) termed as M+M. The results are summarized in Table 2. We observe that using more inducing filters results in better performances. In particular, the HVGP (4xM) achieves the best NLLs. Because of the parallelism, the HVGP (4xM) also has comparable running time with the HVGP (2xM), and both are faster than 2M and M+M for deep models.

M	Model	ACC	NLL	sec/iter
384x0, 1K	M	65.70±0.06	1.65±0.00	0.21
	2M	**67.84±0.07**	1.52±0.00	0.39
	M+M	67.67± 0.07	**1.50 ±0.01**	0.39
	2xM	66.26±1.11	1.76±0.17	0.45
	4xM	67.76±0.05	**1.51±0.01**	0.52
384x1, 1K	M	76.40±0.02	1.03±0.00	0.16
	2M	77.11±0.10	1.00±0.00	0.47
	M+M	**77.48±0.10**	0.98±0.01	0.41
	2xM	77.09±0.18	1.00±0.00	0.37
	4xM	77.30±0.17	**0.95±0.00**	0.36
384x2, 1K	M	79.01±0.11	0.86±0.00	0.17
	2M	80.27±0.04	0.81±0.00	0.52
	M+M	79.98 ±0.21	0.80±0.01	0.46
	2xM	80.04±0.04	0.80±0.00	0.37
	4xM	**80.52±0.20**	**0.75±0.01**	0.37
384x3, 1K	M	82.41±0.08	0.73±0.01	0.40
	2M	-	-	-
	M+M	83.26±0.19	0.69±0.01	1.24
	2xM	**84.97±0.08**	0.60±0.00	0.90
	4xM	**84.85±0.11**	**0.58±0.00**	0.90

Table 2. Deep Convolutional GPs for CIFAR-10 classification. Previous SOTA (Shi et al., 2020) achieves ACC=80.33, NLL=0.82, and 1.25 sec/iter. We use $384\text{x}\ell, 1K$ to represent a $(\ell+1)$-layer model with a respective number of inducing points in each layer. We compare M, 2M, M+M, 2xM, 4xM. We used 4 GPUs for the 4xM model to achieve parallelism. For the four-layer model, using 2M inducing points did not fit in memory. Instead we used a model with $(700\text{x}3, 1600)$ inducing points and achieved ACC=82.89 ± 0.05, NLL=0.73 ± 0.00, and 1.10 sec/iter.

7. Conclusion

We presented the harmonic kernel decomposition which exploited input-space symmetries to obtain an orthogonal kernel sum decomposition, based on which we introduced a scalable variational GP model and analyzed how well the model approximates the true posterior of the GP. We validated its superior performances in terms of scalability and accuracy through a range of empirical evaluations.

Acknowledgements

We thank Wesley Maddox, Greg Benton, Sanyam Kapoor, Michalis Titsias, Radford Neal, and anonymous reviewers for their insightful comments and discussions on this project. We also thank the Vector Institute for providing the scientific computing resources. SS was supported by the Connaught Fellowship. RG acknowledges support from the CIFAR Canadian AI Chairs program.

References

Amante, C. and Eakins, B. Etopo1 1 arc-minute global relief model: procedures, data sources and analysis. noaa technical memorandum nesdis ngdc-24. *National Geophysical Data Center, NOAA*, 10:V5C8276M, 2009.

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Baron Fourier, J. B. J. *The analytical theory of heat*. The University Press, 1878.

Benton, G., Maddox, W. J., Salkey, J., Albinati, J., and Wilson, A. G. Function-space distributions over kernels. In *Advances in Neural Information Processing Systems*, pp. 14965–14976, 2019.

Berlinet, A. and Thomas-Agnan, C. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Blomqvist, K., Kaski, S., and Heinonen, M. Deep convolutional Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 582–597. Springer, 2019.

Bochner, S. *Lectures on Fourier Integrals: With an Author's Supplement on Monotonic Functions, Stieltjes Integrals and Harmonic Analysis; Translated from the Original German by Morris Tenenbaum and Harry Pollard*. Princeton University Press, 1959.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth (he/him), M. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, 2020.

Burt, D., Rasmussen, C. E., and Van Der Wilk, M. Rates of convergence for sparse variational Gaussian process regression. In *International Conference on Machine Learning*, pp. 862–871, 2019.

Burt, D. R., Rasmussen, C. E., and van der Wilk, M. Variational orthogonal features. *arXiv preprint arXiv:2006.13170*, 2020.

Cheng, C.-A. and Boots, B. Variational inference for Gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pp. 5184–5194, 2017.

Cooley, J., Lewis, P., and Welch, P. The finite Fourier transform. *IEEE Transactions on audio and electroacoustics*, 17(2):77–85, 1969.

Dao, T., De Sa, C. M., and Ré, C. Gaussian quadrature for kernel features. In *Advances in neural information processing systems*, pp. 6107–6117, 2017.

Deisenroth, M. and Rasmussen, C. Pilco: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, pp. 465–473, 2011.

Drineas, P., Mahoney, M. W., and Cristianini, N. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(12), 2005.

Dutordoir, V., van der Wilk, M., Artemev, A., Tomczak, M., and Hensman, J. Translation insensitivity for deep convolutional Gaussian processes. *arXiv preprint arXiv:1902.05888*, 2019.

Dutordoir, V., Durrande, N., and Hensman, J. Sparse Gaussian processes with spherical harmonic features. In *International Conference on Machine Learning*, pp. 2793–2802. PMLR, 2020.

Evans, T. and Nair, P. Scalable Gaussian processes with grid-structured eigenfunctions (gp-grief). In *International Conference on Machine Learning*, pp. 1417–1426, 2018.

Ginsbourger, D., Roustant, O., and Durrande, N. On degeneracy and invariances of random fields paths with applications in Gaussian process modelling. *Journal of statistical planning and inference*, 170:117–128, 2016.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pp. 282, 2013.

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pp. 351–360, 2015.

Hensman, J., Durrande, N., and Solin, A. Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research*, 18(1):5537–5588, 2017.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *The annals of statistics*, pp. 1171–1220, 2008.

Izmailov, P., Novikov, A., and Kropotov, D. Scalable Gaussian processes with billions of inducing inputs via tensor train decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 726–735. PMLR, 2018.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pp. 1087–1095, 2009.

Morimoto, M. *Analytic functionals on the sphere*. American Mathematical Society, 1998.

Park, Y.-J., Tagade, P. M., Samsung, R., and Choi, H.-L. Deep matrix-variate Gaussian processes. *RN*, 50:0, 2018.

Parra, G. and Tobar, F. Spectral mixture kernels for multi-output gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 6684–6693, 2017.

Quinonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6: 1939–1959, 2005.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

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

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 4588–4599, 2017.

Salimbeni, H., Cheng, C.-A., Boots, B., and Deisenroth, M. Orthogonally decoupled variational Gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 8711–8720, 2018.

Samo, Y.-L. K. and Roberts, S. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236*, 2015.

Shi, J., Titsias, M., and Mnih, A. Sparse orthogonal variational inference for Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1932–1942, 2020.

Snelson, E. and Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2006.

Solin, A. and Särkkä, S. Hilbert space methods for reduced-rank gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, pp. 1015–1022, 2010.

Tee, G. J. Eigenvectors of block circulant and alternating circulant matrices. *New Zealand Journal of Mathematics*, 36(8):195–211, 2007.

Thomson, S. W. and Tait, P. G. *Treatise on natural philosophy*. 1888.

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

Van der Wilk, M., Rasmussen, C. E., and Hensman, J. Convolutional Gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 2849–2858, 2017.

Van der Wilk, M., Bauer, M., John, S., and Hensman, J. Learning invariances using the marginal likelihood. In *Advances in Neural Information Processing Systems*, pp. 9938–9948, 2019.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, pp. 14648–14659, 2019.

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. and Nickisch, H. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pp. 1775–1784, 2015.

Wilson, A. G., Dann, C., and Nickisch, H. Thoughts on massively scalable Gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pp. 1975–1983, 2016.