Variational Nearest Neighbor Gaussian Process

Luhuan Wu [1] **Geoff Pleiss** [2] **John Cunningham** [1 2]

Abstract

Variational approximations to Gaussian processes (GPs) typically use a small set of inducing points to form a low-rank approximation to the covariance matrix. In this work, we instead exploit a sparse approximation of the precision matrix. We propose variational nearest neighbor Gaussian process (VNNGP), which introduces a prior that only retains correlations within K nearest-neighboring observations, thereby inducing sparse precision structure. Using the variational framework, VNNGP's objective can be factorized over both observations and inducing points, enabling stochastic optimization with a time complexity of $O(K^3)$. Hence, we can arbitrarily scale the inducing point size, even to the point of putting inducing points at every observed location. We compare VNNGP to other scalable GPs through various experiments, and demonstrate that VNNGP (1) can dramatically outperform low-rank methods, and (2) is less prone to overfitting than other nearest neighbor methods.

1. Introduction

Gaussian processes (GPs) provide rich priors over functions (Rasmussen & Williams, 2006). While the GP posterior can be computed in closed form for regression models, variational posterior approximations (Titsias, 2009; Hensman et al., 2013; van der Wilk et al., 2020) have become increasingly popular as they offer numerous advantages. First, variational methods can utilize stochastic minibatching (Hensman et al., 2013), enabling inference for large-scale datasets. Second, variational methods are compatible with non-conjugate likelihoods (Hensman et al., 2015) and simplify inference for interdomain observation models (Lázaro-Gredilla & Figueiras-Vidal, 2009; Álvarez et al., 2010; van der Wilk et al., 2020; Wu et al., 2021). Finally, black box variational methods (Ranganath et al., 2014) make it possible to use GP as components of larger models, such as Deep Gaussian Processes (Damianou & Lawrence, 2013).

To apply variational methods to large-scale GP, it is often necessary to make additional approximations (Titsias, 2009; Hensman et al., 2017). This is because making GP inference with N observations incurs a $\mathcal{O}(N^3)$ computation. A popular variational approach that aids this issue is the Stochastic Variational Gaussian Process method (SVGP) (Hensman et al., 2013), which essentially forms a low-rank approximation to the prior covariance. These approximations offer well understood predictive properties (Bauer et al., 2016), probable performance guarantees (Burt et al., 2019), and competitive performance on numerous datasets. However, there are also many instances when low-rank approximations are ill suited. For example, spatiotemporal datasets often have *intrinsically low lengthscale*, meaning that the data can vary rapidly with small changes in space and time. In these settings, low-rank approximations cannot capture fast variations, often resulting in low fidelity or "blurry" predictions (e.g. Datta et al., 2016a; Pleiss et al., 2020; Wu et al., 2021).

In this paper, we are interested in making variational GP methods more amenable to spatiotemporal data and other domains with intrinsically low lengthscale, while retaining (and even improving) scalability. To do so, we replace SVGP's low-rank approximation of the prior covariance matrix with a *sparse approximation of the prior precision matrix*. Whereas a low-rank approximation assumes that observations are explained through a small number of global latent variables, a sparse precision approximation instead assumes that observations are conditionally independent given their nearest neighboring observations. Mathematically, the joint GP prior is approximated by the product of one-dimensional conditionals, each of which only depends on a small subset of preceding observations based on some predetermined ordering. This assumption is quite reasonable in the context of spatiotemporal modeling. For example, it is reasonable to assume that the weather in Los Angeles and Maine are conditionally independent, given neighboring cities in California.

Sparse precision approximations are common in spatiotemporal modeling throughout the geostatisics literature (e.g.

[1]Department of Statistics, Columbia University [2]Zuckerman Institute, Columbia University. Correspondence to: Luhuan Wu <lw2827@columbia.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Vecchia, 1988; Stein et al., 2004; Datta et al., 2016a; Guinness, 2018; Finley et al., 2019; Katzfuss et al., 2021), where they often achieve higher predictive fidelity than low-rank approximations. However, most existing approaches target exact or MCMC inference and are not compatible with variational methods. As a result, it is not straightforward to use sparse precision methods with non-conjugate or interdomain likelihoods, or as part of larger probabilistic models without resorting to sampling. Recently, Tran et al. (2021) made a first key step in this direction. They introduce Sparse Within Sparse Gaussian Processes (SWSGP), which combines a similar nearest neighbor approximation with SVGP. Instead of exploiting the sparse precision structure, SWSGP introduces sparsity by constructing a hierarchical prior. While successful, we will show an important technical opportunity remains, which we offer here.

In this work, we propose a novel sparse precision approximation that is compatible with stochastic variational inference. Similar to SVGP, our proposed method—the *Variational Nearest Neighbor Gaussian Process* (**VNNGP**) [1]—formulates the variational objective and predictive distribution through a set of M *inducing points*. Unlike SVGP, our method applies 1) a sparse precision approximation to the inducing point distribution and 2) a sparse dependency approximation between observations and inducing points. These approximations enable the VNNGP objective to factorize over data *and* inducing points. Consequently, we can minibatch over data and inducing points when evaluating the variational objective, resulting in *constant time computational complexity with respect to both N and M*. This significant reduction makes it possible to scale M well beyond the limits of SVGP, even to the point of placing inducing points at every observed location. Moreover, unlike SWSGP and other sparse approximations (Vecchia, 1988; Katzfuss et al., 2021), the VNNGP prior and variational distribution both constitute valid GPs, enabling it to more faithfully retain the features of the true GP prior. We compare VNNGP to SVGP, SWSGP, and other scalable methods on numerous toy experiments and benchmark datasets. Our results demonstrate that VNNGP can offer higher fidelity predictions than SVGP and is less prone to overfitting than SWSGP. These advantages are most pronounced on spatiotemporal datasets (including those with non-conjugate observational models), but hold for high dimensional datasets as well.

[1] VNNGP is implemented in the GPyTorch library. See the example in https://docs.gpytorch.ai/en/stable/examples/04_Variational_and_Approximate_GPs/VNNGP.html

2. Background

2.1. Gaussian Processes

Consider any finite dimensional realization $\mathbf{f} = \{f_i\}_{i=1}^N$ of a Gaussian process at locations $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N \in \mathbb{R}^D$:

$$\begin{aligned} f(\cdot) &\sim \mathcal{GP}(0, k(\cdot,\cdot)) \\ f_i &\equiv f(\mathbf{x}_i) \qquad \text{for } i = 1:N \end{aligned} \tag{1}$$

where $k(\cdot,\cdot)$ is a kernel function that encodes the properties of the prior. We can always factorize the joint probability of a GP realization into a chain of conditional probabilities subject to some ordering, for example

$$p(\mathbf{f}) = p(f_1) \prod_{i=2}^{N} p(f_i|\mathbf{f}_{1:i-1}), \tag{2}$$

where $p(f_1) = \mathcal{N}(f_1|0, k_{1,1})$,

$$\begin{aligned} p(f_i|\mathbf{f}_{1:i-1}) = \mathcal{N}(f_i|&\mathbf{k}_{1:i-1,i}^\top \mathbf{K}_{1:i-1,1:i-1}^{-1} \mathbf{f}_{1:i-1}, \\ &k_{i,i} - \mathbf{k}_{1:i-1,i}^\top \mathbf{K}_{1:i-1,1:i-1}^{-1} \mathbf{k}_{1:i-1,i}) \end{aligned} \tag{3}$$

and we define $k_{i,i} \equiv k(\mathbf{x}_i, \mathbf{x}_i)$, $\mathbf{k}_{1:i-1,i} \equiv k(\mathbf{x}_{1:i-1}, \mathbf{x}_i)$, and $\mathbf{K}_{1:i-1,1:i-1} \equiv k(\mathbf{x}_{1:i-1,i}, \mathbf{x}_{1:i-1})$. The following notations will apply this convention accordingly.

2.2. Sparse Precision Approximations

GP approximations based on sparse precision matrices remain popular for years, largely in the geostatistics community (Vecchia, 1988; Datta et al., 2016a; Finley et al., 2019; Katzfuss et al., 2021). They consider the following approximation to Eq 2

$$p(\mathbf{f}) \approx p(f_1) \prod_{i=2}^{N} p(f_i|\mathbf{f}_{n(i)}), \tag{4}$$

where $p(f_1) = \mathcal{N}(f_1|0, k_{1,1})$,

$$\begin{aligned} p(f_i|\mathbf{f}_{n(i)}) = \mathcal{N}(f_i|&\mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1} \mathbf{f}_{n(i)}, \\ &k_{i,i} - \mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1} \mathbf{k}_{n(i),i}) \end{aligned} \tag{5}$$

and $n(i)$ denotes the indices of K nearest neighbors of $\mathbf{x}_i$ in $\{\mathbf{x}_k\}_{k=1}^{i-1}$. Typically K is set to be much smaller than N, and when $K = N$ Eq 4 recovers the original GP. This approximation results in a sparse Cholesky factor of the precision matrix (Datta, 2021b). As a consequence, the model complexity scales $O(NK^3)$. It has seen many successful applications in spatiotemporal problems, e.g. predicting PM polutant levels arocss Europe (Datta et al., 2016b), predicting forest canopy height (Finley et al., 2017), estimating forest biomass (Taylor-Rodriguez et al., 2019), etc.

2.3. Stochastic Variational Gaussian Processes

Stochastic variational Gaussian process (SVGP) (Hensman et al., 2013) defines a small set of M inducing points $\mathbf{u} = \{u_j\}_{j=1}^M$ which are GP latent variables at locations $\{\mathbf{z}_j\}_{j=1}^M$. It considers the joint latent generative process:

$$p(\mathbf{u}, \mathbf{f}) = \mathcal{N}(\begin{pmatrix}\mathbf{u}\\ \mathbf{f}\end{pmatrix} | 0, \begin{pmatrix}\mathbf{K}_{\mathbf{z},\mathbf{z}} & \mathbf{K}_{\mathbf{z},\mathbf{x}}\\ \mathbf{K}_{\mathbf{x},\mathbf{z}} & \mathbf{K}_{\mathbf{x},\mathbf{x}}\end{pmatrix}) \tag{6}$$

and an independent observation model

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^N p(y_i|f_i) \tag{7}$$

where $p(y_i|f_i)$ is an arbitrary likelihood function. SVGP is optimized by maximizing the evidence lower bound (ELBO) given below:

$$\mathcal{L}_{\text{SVGP}} = \sum_{i=1}^N \mathop{\mathbb{E}}_{q(f_i)} \left[\log p(y_i|f_i)\right] - \text{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right], \tag{8}$$

where $q(\mathbf{u})$ is the variational posterior for $\mathbf{u}$ and $q(f_i) = \int p(f_i|\mathbf{u})q(\mathbf{u})\mathrm{d}\mathbf{u}$ is the variational posterior for f_i. The SVGP approximation relies on the low-rank matrix $\mathbf{K}_{\mathbf{x},\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{z},\mathbf{x}}$ instead of the full-rank matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ to reduce the size of any matrix inversion to M. Therefore, we refer to it as a low-rank approximation.

SVGP has several advantages compared to exact GP and other scalable methods. First, as is shown in Eq 8, the ELBO is factorzied over data points. Therefore, it is amenable to stochastic optimization and the computational complexity reduces to $O(M^3)$. Moreover, due to its variational inference nature, SVGP can be applied to problems with non-conjugate likelihoods, inter-domain observations or latent variable models based on GPs. However, since SVGP assumes $M \ll N$, it may struggle to obtain accurate predictions for large-scale data that are not inherently low-rank structured (Wu et al., 2021; Tran et al., 2021).

2.4. Other Related Works

Variational GP inference was first pursued by Csató et al. (2000) and Gibbs & MacKay (2000). More recent works connect variational inference with scalable methods. Titsias (2009) ties inducing point approximations and variational inference, and Hensman et al. (2013) extends this approach to be compatible with minibatched optimization. Recent extensions focus on inducing points in linearly transformed domains (Lázaro-Gredilla & Figueiras-Vidal, 2009; van der Wilk et al., 2020), inducing points with exploitable algebraic kernel structures (Wilson & Nickisch, 2015; Wu et al., 2021), and inducing points with separable structure (Cheng & Boots, 2017; Salimbeni et al., 2018; Shi et al., 2020). Beyond inducing point approximations, Hensman et al. (2017) and John & Hensman (2018) combine variational inference with finite basis approximations.

Nearest neighbor approximations. In addition to the sparse precision approximations pursued by the geostatistics community (see introduction), the machine learning community has proposed nearest neighbor GP approximations (Kim et al., 2005; Nguyen-Tuong et al., 2008; Gramacy & Apley, 2015; Park & Apley, 2018; Jankowiak & Pleiss, 2021). Besides SWSGP which is introduced before, we review three methods that are most related to our method. Bui & Turner (2014) proposes a sparse prior by imposing a tree or chain structure over inducing points and calibrating the posterior approximation with a KL divergence minimization. This approach uses a message-passing inference algorithm and learns the hyperparameters by maximizing the marginal likelihood over inducing points. Stochastic gradient descent GP (sgGP) forms a (biased) stochastic GP objective that optimizes K-nearest-neighboring data points per iteration (Chen et al., 2020). This method is an approximation to exact GP inference, and therefore is limited to Gaussian likelihood. Liu & Liu (2019) use amortized variational inference for data points within a small neighborhood, where the variational covariance is constructed to have a sparse Cholesky factor. The latter two methods have computational complexity cubic in the number of nearest neighbors.

3. Variational Nearest Neighbor GP

In this section, we develop VNNGP, a highly scalable variational GP method. VNNGP considers the same observation model in Eq 7 and makes the following nearest neighbor approximations to the latent generative process

$$p(\mathbf{u}) \approx \prod_{j=1}^M p(u_i|\mathbf{u}_{n(j)}), \quad p(\mathbf{f}|\mathbf{u}) \approx \prod_{i=1}^N p(f_i|\mathbf{u}_{n(i)}) \tag{9}$$

where each conditional denotes a standard GP predictive distribution, $n(j)$ denotes the inducing point indices corresponding to $\mathbf{z}_j$'s (at most) K nearest neighbors selected from $\{\mathbf{z}_1, \cdots, \mathbf{z}_{j-1}\}$ (with a special case of $n(1) = \emptyset$), and we overload the notation $n(i)$ to denote the inducing point indices of $\mathbf{x}_i$'s K nearest neighbors in all $\{\mathbf{z}_j\}_{j=1}^M$.

Because of this construction, VNNGP forms a sparse approximation to the prior precision matrix $\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}$ (Datta, 2021b). Its Cholesky factor has at most $K+1$ non-zero elements per row, corresponding to the K-nearest-neighbor dependency structure. Figure 1 plots the Cholesky factor of $\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}$ for $M = 20$ inducing points with varying number of nearest neighbors K. We further see that, as K increases, the Cholesky factor becomes denser and approaches the exact one. In this case, $K = 10$ yields a quality approximation. Note that the sparsity pattern of the Cholesky factor depends on the ordering of inducing points and the nearest neighbor sets. In Figure 1 we use the coordinate ordering of 1-dimensional inducing points, thereby rendering a banded sparsity pattern.

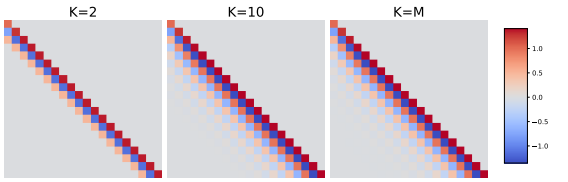

Figure 1. The Cholesky factor of prior precision matrix for $M = 20$ inducing points (with even spacing of 1) by VNNGP with different number of nearest neighbors K. We use a squared exponential kernel with outputscale 1 and lengthscale 1. For each row of the Cholesky factor, at most $K + 1$ elements are non-zero, and when $K = 10$ the approximation is qualitatively close to the exact one.

Ordering of inducing points. The nearest neighbor approximation in Eq 9 is subject to a particular ordering of inducing points. In all following experiments, we apply random ordering and we find that different orderings do not lead substantial differences on VNNGP's performance.

Variational family. We consider a special choice of the variational distribution $q(\mathbf{f}, \mathbf{u})$ that will greatly simplify the computation without sacrificing much prediction accuracy. Specifically, we use a mean-field variational approximation for $\mathbf{u}$ and approximate the posterior for $\mathbf{f}$ with the nearest neighbor prior:

$$q(\mathbf{u}) = \prod_{j=1}^{M} q(u_j) = \prod_{j=1}^{M} \mathcal{N}(u_j | m_j, s_j), \tag{10}$$

$$q(\mathbf{f}) = \prod_{i=1}^{N} q(f_i) = \prod_{i=1}^{N} \int p(f_i | \mathbf{u}_{n(i)}) q(\mathbf{u}_{n(i)}) \mathrm{d}\mathbf{u}_{n(i)}. \tag{11}$$

The choice of mean-field or sparse variational approximations are common for large-scale GP when full-rank approximations are impractical (Liu & Liu, 2019; Wu et al., 2021; Tran et al., 2021). Empirically, we found that VNNGP with a mean-field variational distribution outperforms SVGP with a full-rank one for most cases. However, we also discuss the extension to a more expressive variational family in appendix B.

Optimization objective is to maximize the VNNGP's ELBO as follows

$$\mathcal{L}_{\text{VNNGP}} = \underbrace{\sum_{i=1}^{N} \mathop{\mathbb{E}}_{q(f_i)} \left[\log p(y_i | f_i)\right]}_{\text{data likehood}} - \underbrace{\mathrm{KL}\left[q(\mathbf{u}) \,\|\, p(\mathbf{u})\right]}_{\text{KL divergence}}, \tag{12}$$

Note that the ELBO breaks into a data likelihood term and a KL divergence term. In the following , we first discuss the computation of these two terms. After that, we present the advantages of VNNGP. Finally, we compare VNNGP to other related works based on nearest neighbor approximations.

3.1. Computing the Data Likelihood Term

Similar to SVGP, the data likelihood term for VNNGP factorizes over data points. Unlike SVGP that considers all M inducing points in making the prediction for one data point, VNNGP makes use of the sparse prior in Eq 9 such that the prediction is based on only K nearest inducing points. As a result, VNNGP's data likelihood term is reduced to

$$\sum_{i=1}^{N} \int p(f_i | \mathbf{u}_{n(i)}) q(\mathbf{u}_{n(i)}) \log p(y_i | f_i) df_i d\mathbf{u}_{n(i)} \tag{13}$$

where only K nearest inducing points are involved in computing the expected log likelihood for each data point.

3.2. Computing the KL Divergence

We now show how we can write KL divergence as a summation over inducing points. Note that it is not possible for the standard SVGP: SVGP's KL requires accessing parameters for all inducing points at the same time.

We first break the prior into a chain of conditionals using the nearest neighbor assumption in Eq 9 and utilize the mean-field assumption in Eq 10:

$$\begin{aligned} \mathrm{KL}\left[q(\mathbf{u}) \,\|\, p(\mathbf{u})\right] &= \mathrm{KL}\left[\prod_{j=1}^{M} q(u_j) \,\|\, \prod_{j=1}^{M} p(u_j | \mathbf{u}_{n(j)})\right] \\ &= \mathop{\mathbb{E}}_{q(\mathbf{u})}\left[\sum_{j=1}^{M} \log \frac{q(u_j)}{p(u_j | \mathbf{u}_{n(j)})}\right] \end{aligned} \tag{14}$$

exchanging the order of summation and expectation and integrating out irrelevant variables

$$= \sum_{j=1}^{M} \mathop{\mathbb{E}}_{q(\mathbf{u}_{n(j)})}\left[\mathbb{E}_{q(u_j)}\left[\log \frac{q(u_j)}{p(u_j | \mathbf{u}_{n(j)})}\right]\right] \tag{15}$$

$$= \sum_{j=1}^{M} \mathop{\mathbb{E}}_{q(\mathbf{u}_{n(j)})}\left[\mathrm{KL}\left[q(u_j) \,\|\, p(u_j | \mathbf{u}_{n(j)})\right]\right] \tag{16}$$

where

$$p(u_j | \mathbf{u}_{n(j)}) = \mathcal{N}(u_j | \mathbf{k}_{n(j),j}^{\top} \mathbf{K}_{n(j),n(j)}^{-1} \mathbf{u}_{n(j)}, \tag{17}$$

$$k_{j,j} - \mathbf{k}_{n(j),j}^{\top} \mathbf{K}_{n(j),n(j)}^{-1} \mathbf{k}_{n(j),j}). \tag{18}$$

3.3. Advantages

Having factorized the data likelihood term and KL divergence over data points and inducing points, we are now ready to present VNNGP's key advantages as follows.

Stochastic optimization. The summation structure of VNNGP's ELBO, as detailed in Eq 13 and 16, makes it immediately available for stochastic optimization: in every training

iteration, we randomly sample a mini-batch of training data indices $\mathcal{I} = \{i_k\}_{k=1}^{N_b}$ and a mini-batch of inducing point indices $\mathcal{J} = \{j_l\}_{l=1}^{M_b}$; we then optimize the unbiased estimate of the ELBO as follows

$$\mathcal{L}_{\text{VNNGP}} \approx \frac{N}{N_b} \sum_{i \in \mathcal{I}} \mathop{\mathbb{E}}_{q(f_i)} \left[\log p(y_i|f_i)\right] - \frac{M}{M_b} \sum_{j \in \mathcal{J}} \mathop{\mathbb{E}}_{q(\mathbf{u}_{n(j)})} \left[\text{KL}\left[q(u_j) \parallel p(u_j|\mathbf{u}_{n(j)})\right]\right]. \tag{19}$$

We include the exact computation of ELBO in Appendix A. Note that the sampling distributions for data points and inducing points can be arbitrary as long as both remain marginally a uniform distribution. We emphasize that this is a key advantage over SVGP: we are not able to perform stochastic optimization for inducing points with SVGP.

Computational complexity. Once the nearest neighbor structure is computed, optimizing the objective in Eq 19 requires $O((N_b + M_b)K^3)$ computations for inverting $K \times K$ kernel matrices induced by nearest neighbor mini-batches. See Appendix A for mathematical details.

The computational overhead comes from determining nearest neighbor structures for observations and inducing points. The brute-force complexity of finding any point's K nearest neighbors within M points is $O(KM)$. With the help of modern similarity search packages such as Johnson et al. (2017), we are able to dramatically speed up billion-scale similarity search with gpus. For example, for medium-sized datasets, e.g. Protein ($N = 25.6$K, $D = 9$), it takes no more than 30 seconds to build up nearest neighbor structures with $K = 256$ and $M = N$ on an NVIDIA RTX2080 gpu; for the largest dataset we experiment with (Covtype: $N = 372$K and $D = 54$), it takes approximately 12 minutes with $K = 256$ and $M = N$.

More inducing points. Since the training complexity is free of N and M, we are able to place inducing points at every observed location and scale M to N. We enjoy several benefits by doing so. First, we greatly boost the model capacity, as opposed to a low-rank approximation where several observations "share" one inducing point. Moreover, we avoid optimizing inducing point locations as the standard SVGP training procedure requires. This is a huge advantage since it saves training cost and the optimization of inducing locations is in general non-convex and more difficult. Finally, we only need to compute the nearest neighbor structure once before training, or otherwise we need to recompute it frequently since inducing locations are being updated.

3.4. Comparison to Related Methods

The method most related to VNNGP is the Sparse Within Sparse Gaussian Process (SWSGP) method of Tran et al. (2021). SWSGP also builds upon SVGP and further imposes sparsity over M inducing points $\mathbf{u}$ by defining a hierarchical prior. Essentially, in each training iteration, SWSGP randomly samples a minibatch of data point indices $\mathcal{I} = \{i_k\}_{k=1}^{N_b}$ and optimizes the following ELBO

$$\mathcal{L}_{\text{SWSGP}} = \frac{N}{N_b} \sum_{i \in \mathcal{I}} \mathop{\mathbb{E}}_{q(f_i)} \left[\log p(y_i|f_i)\right] - \frac{1}{N_b} \sum_{i \in \mathcal{I}} \text{KL}\left[q(\mathbf{u}_{n(i)}) \parallel p(\mathbf{u}_{n(i)})\right]), \tag{20}$$

where $n(i)$ are indices of inducing point locations from $\{\mathbf{z}_j\}_{j=1}^{M}$ that are top-K nearest to $\mathbf{x}_i$. Similar to VNNGP, evaluating Eq. 20 requires $O(N_b K^3)$ computation.

There are two key differences between VNNGP and SWSGP. First, the two methods introduce sparsity through a different generative process. SWSGP applies a hiearchical prior; Consequently, the marginal prior for either $p(\mathbf{f})$ or $p(\mathbf{u})$ is no longer Gaussian. VNNGP, on the other hand, considers a sparse approximation to the prior precision matrix. The resulting process is still a GP but under an approximated kernel. While it is not immediately clear how these differing marginal distribution impact downstream performance, in this sense the VNNGP generative process more faithfully replicates the exact GP prior.

The second difference is the training objective. As can be seen, the first term of Eq. 20 (data likelihood) is identical to that of VNNGP (Eq. 19). However, the KL divergence term is where two objectives differ. SWSGP optimizes a "local" KL divergence that only involves inducing points within a local neighborhood around current batch of training data. In fact, we prove (see appendix C) that SWSGP's KL divergence always underestimates the SVGP KL divergence:

$$\text{KL}\left[q(\mathbf{u}_{n(i)}) \parallel p(\mathbf{u}_{n(i)})\right]) \leq \text{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right], \tag{21}$$

for any subset $n(i)$. Additionally, the amount of underestimation depends on size of $n(i)$.

We hypothesize that SWSGP's KL, due to its "local" characteristic, may not sufficiently regularize the variational distribution—especially when $K \ll M$—which may make the model more prone to overfitting. Conversely, though VNNGP's KL does not equal the exact KL, it does (in expectation) consider the joint distribution over all inducing points. Optimizing this term will regularize the variational distribution towards the GP prior at all input locations. We investigate our hypotheses in the experiments section.

Another method that also combines nearest neighbor approximations with variational inference is Amortized Inference GP (Liu & Liu, 2019). However, their method avoids computing certain terms of the KL divergence, and in doing so makes it impossible to simulatenously optimize kernel

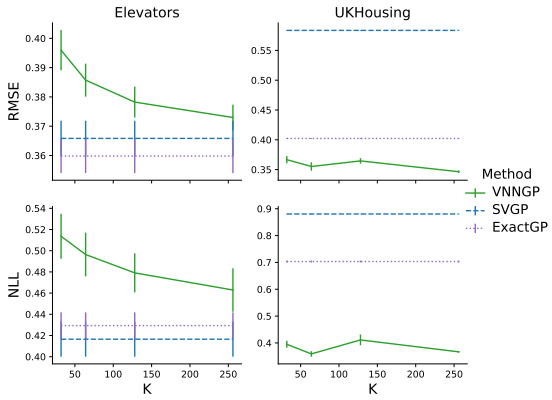

Figure 2. Test set RMSE and NLL as a function of number of nearest neighbors K for VNNGP (green) on Elevators (left column) and UKHousing datasets (right column). Results for SVGP (blue dashed) and exact GP (purple dotted) are included as baselines.

hyperparameters using the ELBO. By contrast, the VNNGP objective enables gradient-based kernel learning.

4. Experiments

We first demonstrate how sensitive our model is to the number of nearest neighbors K. Recognizing that our method is similar to SWSGP, our next task is to characterize the differences between two methods and investigate our hypotheses in Sec 3.4. Finally, we evaluate VNNGP and other methods on a wide range of real regression and classification datasets. Unless stated otherwise, for all experiments, VNNGP applies a random ordering of inducing points, VNNGP and SWSGP set inducing points at observed locations and use a mean-field variational approximation, and SVGP uses 1024 inducing points and a full-rank variational approximation.

4.1. Sensitivity to number of nearest neighbors K

Our first goal is to see how the number of nearest neighbors K impact VNNGP's performance. We choose two datasets, Elevators with dimension $D = 16$ (Asuncion & Newman, 2007), and UKHousing (https://landregistry.data.gov.uk/), a spatial dataset with $D = 2$, as we expect low-rank approximations to work better in high dimensional settings while nearest neighbor methods are more suitable for spatial problems that are sparse in nature. We fit VNNGP with $K \in [32, 64, 128, 256]$. Since these two datasets are modeled with Gaussian likelihood, we are able to fit exact GP (Wang et al., 2019) as a baseline. We also include SVGP as a low-rank variational alternative. We present their test set predictive performance, the root mean squared error (RMSE) and negative log likelihood (NLL), in Figure 2. From the figure, we see that as K increases, both test RMSE and NLL for VNNGP decrease, suggesting a better performance. By comparing the three methods, we note that exact GP does not always yield the best RMSE or NLL. This result could be due to model misspecification, as observed in previous literature (Wang et al., 2019; Potapczynski et al., 2021). Moreover, SVGP performs better on the high-dimensional Elevators dataset, whereas VNNGP with $K = 32$ already outperforms SVGP and is less sensitive to K on the spatial UKHousing dataset.

4.2. Comparison to SWSGP

Building on the analysis in Section 3.4, we now aim to compare VNNGP to the related SWSGP method. Recall that the primary difference between these methods is the training objective (the KL divergence term in particular). We first empirically demonstrate how these two objectives differ, and then study its surprisingly significant impact on model fitting and selection.

KL divergence comparison. We first investigate the difference between the VNNGP and SWSGP KL divergence terms. Specifically, we are interested in how their KL terms change as we vary K. We initialize VNNGP and SWSGP with the same configuration: the inducing points are placed on a 1-dimensional grid with even spacing of 1, the variational means $\{m_i\}$ are obtained by sampling a GP under a squared exponential kernel with lenghthscale 5 and outputscale 1, and the variational variances $\{s_i\}$ are set to 1. For both methods, we compute their KL terms by varying the kernel lengthscale parameter and # of nearest neighbors K. A special note is that when $K = M$, KL by either method should be equal to the KL by SVGP in Eq. 8.

We plot the SWSGP and VNNGP KL terms as a function of K under different kernel lengthscales in Figure 3. From this figure, we first confirm that VNNGP and SWSGP recover the SVGP KL divergence (dotted blue line) at $K = M$. Morever, both methods tend to underestimate KL (recall from Section 3.4 that SWSGP provably underestimates this term). Notably, VNNGP converges rapidly to the true KL divergence, especially for small lengthscale settings. For example, when the lengthscale is 5, the VNNGP KL divergence is indistinguishable from the true KL divergence for all $K \geq 5$. We include results under different settings in appendix D.1 and observe similar patterns.

Fitting models in high observational noise setting. We now study the impact of KL divergence on the inferred posterior distribution. Recall that the KL term of the ELBO regularizes the variational distribution towards the GP prior. If the regularization is not sufficient, the ELBO will be dominated by the data likelihood term, which could overfit the predictive distribution to the observed data.

Hence, we conduct a simulation experiment in the presence of high observational noise: We draw $N = 50$ observations from a GP under a squared exponential kernel with length-

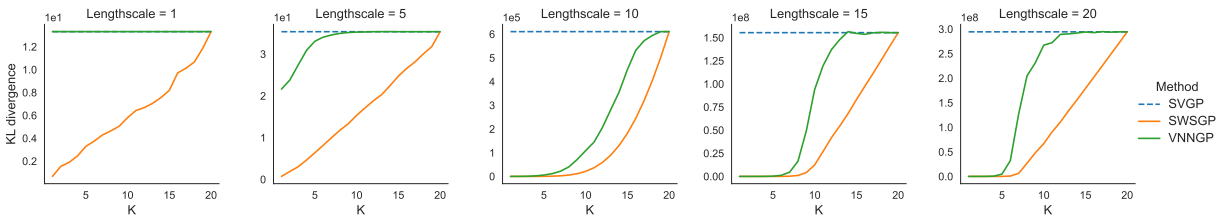

Figure 3. KL divergence computed by VNNGP (green) and SWSGP (orange) as a function of K for different prior kernel lengthscales. Both methods tend to underestimate the KL (SWSGP provably), and converges to the exact value computed by SVGP (blue dotted) as $K \to M$. Morever, VNNGP's KL is closer to the exact one compared to SWSGP for the same K across all cases.

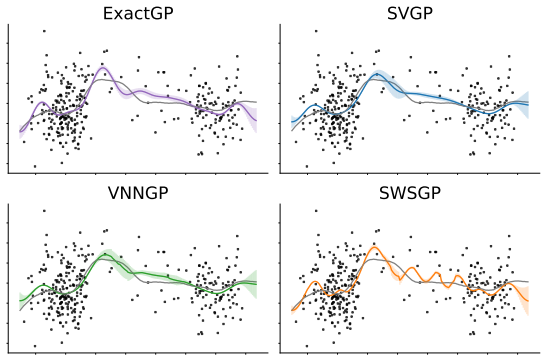

Figure 4. Posterior by Exact GP (purple), SVGP (blue), VNNGP (green) and SWSGP (orange), with 95% uncertainty interval. The grey line is to the true data generating process, and scattering black dots are highly noisy observations.

scale 5 and outputscale 5, and add iid noise from $\mathcal{N}(0, 5)$. There are two dense clusters of observations around $x = 0$ and $x = 50$, and scarcely scattered observations in between. We fit four models—exact GP, SVGP, VNNGP and SWSGP—to the observed data. We fix all hyperparameters corresponding to the true data generating process, and only optimize variational parameters. For the three variational methods, inducing points are set to the locations of observed data ($M = N$). All variational models use mean-field variational approximations. Both VNNGP and SWSGP use $K = 5$ nearest neighbors.

In Fig 4, we display the true data generating process, noisy observations and posteriors of all four models. The SVGP and VNNGP posteriors are qualitatively close to that of exact GP, whereas SWSGP's posterior deviates from the underlying GP function in the data-sparse region. We suspect SWSGP's behavior to be due to its underestimating local KL divergence. As we discuss in Section 3.4, the "local" KL divergence of SWSGP will not regularize the variational distribution towards a globally smooth distribution. In particular, the SWSGP KL divergence concentrates on the data-dense regions, leading to a lack of regularization in the data-sparse region. By contrast, the VNNGP model is effectively regularized towards the GP prior throughout the input space.

Model selection. Finally, we turn to model selection by maximizing the ELBO. Our goal is to see whether VNNGP or SWSGP is sensitive to different settings of hyperparameters, in particular, the likelihood noise.

We first fit exact GP to the UKHousing dataset under a Matern 5/2 kernel and a Gaussian likelihood. We copy and fix the exact GP kernel hyperparameters to SVGP, VNNGP, and SWSGP models. However, we vary the value of likelihood noise from 0.001 to 1.0. We then optimize only the variational parameters of these three methods for 300 epochs. In Figure 5, we plot the negative training ELBO and test NLL as a function of likelihood noise for each method. We denote the noise parameters that maximize each method's training objective, as well as the noise parameter selected by the exact GP. In agreement with prior work (Bauer et al., 2016), we find that SVGP tends to overestimate the noise parameter. Both nearest neighbor methods tend to underestimate the likelihood noise. However, the ELBO of SWSGP monotonically decreases as likelihood noise decreases, whereas the VNNGP ELBO achieves an optimum as a noise of 0.1. If we use the ELBO as a model selection criterion (as is often the case for variational methods), the resulting SWSGP model will have very low likelihood noise which results in poor test NLL. Conversely, the VNNGP training ELBO is highly predictive of the test NLL, as both are roughly optimized as the same likelihood noise value. We note that SWSGP's behavior in this setting is likely due to having $M = N$. We repeat this experiment for SWSGP models that use $M \ll N$ inducing points in Appendix D.2, and find that these models favor more reasonable likelihood noise values. These observations suggest that the ELBO is a good model selection criteria for VNNGP; however, ELBO optimization may result in extreme hyperparameters for SWSGP with large M.

4.3. Predictive performance on real datasets

We conduct an extensive evaluation of our method on real datasets. We consider a wide range of high dimensional and spatiotemporal datasets from the UCI repository (Asuncion & Newman, 2007). In addition we include three spatial datasdets, UKHousing as mentioned in Section 4.1, Precipi-

Dataset	N	D	ExactGP	sgGP	SVGP	SWSGP-M	SWSGP-N	VNNGP
			NLL					
PoleTele	9.6K	26	$-.509 \pm .005$	$-.558 \pm .016$	$-.708 \pm .003$	$-.898 \pm .029$	$-.962 \pm .023$	$\mathbf{-1.160 \pm .014}$
Elevators	10.6K	16	$.464 \pm .011$	$\mathbf{.432 \pm .013}$	$\mathbf{.417 \pm .017}$	$.691 \pm .049$	$.767 \pm .026$	$.463 \pm .020$
Bike	11.1K	17	$-.744 \pm .007$	53.364 ± 27.690	$\mathbf{-1.849 \pm .016}$	$-1.684 \pm .024$	$-1.405 \pm .096$	$-1.690 \pm .057$
Kin40K	25.6K	8	$-.149 \pm .001$	$.701 \pm .139$	$-.399 \pm .003$	$-.747 \pm .016$	$\mathbf{-1.010 \pm .005}$	$\mathbf{-1.016 \pm .004}$
Protein	25.6K	9	$1.044 \pm .003$	$.914 \pm .004$	$.967 \pm .005$	$.909 \pm .006$	$1.505 \pm .080$	$\mathbf{.671 \pm .009}$
KEGG	31.2K	20	$-.713 \pm .004$	$\mathbf{-1.013 \pm .035}$	$-1.000 \pm .011$	$\mathbf{-1.039 \pm .010}$	$14.925 \pm .838$	$\mathbf{-1.039 \pm .018}$
KEGGU	40.7K	26	$-.471 \pm .003$	$-.673 \pm .005$	$-.680 \pm .002$	$\mathbf{-.719 \pm .004}$	$41.675 \pm .427$	$\mathbf{-.715 \pm .004}$
Precipitation	64.8K	3	—	—	$.818 \pm .004$	$.396 \pm .001$	$1.809 \pm .224$	$\mathbf{.145 \pm .002}$
UKHousing	116K	2	$.703 \pm .003$	$.598 \pm .001$	$.879 \pm .004$	$.455 \pm .003$	$1.050 \pm .013$	$\mathbf{.367 \pm .003}$
3DRoad	278K	3	$.993 \pm .000$	$.712 \pm .001$	$.325 \pm .005$	$.584 \pm .003$	$\mathbf{-1.476 \pm .079}$	$-.048 \pm .406$
Covtype	372K	54	—	—	$.234 \pm .001$	$.172 \pm .001$	$\mathbf{.069 \pm .000}$	$.132 \pm .000$

Table 1. Test set NLL (mean $\pm$ 1 standard error over 3 random seeds) on high-dimensional datasets (top 7 ones) and spatial datasets (bottom 4 ones). VNNGP achieves lowest NLL for most datasets. The result for RMSE is included in Table 2 in appendix.

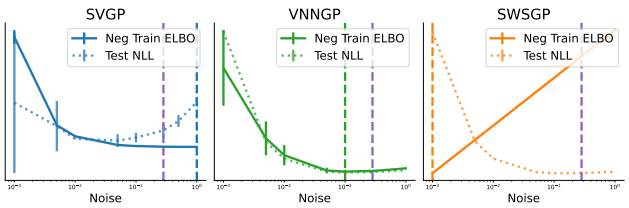

Figure 5. Negative training ELBO (solid line) and test NLL (dotted line) as a function of the likelihood noise by SVGP (blue), VNNGP (green) and SWSGP (orange) on UKHousing dataset. All ELBO and NLL values are scaled to $[0, 1]$. The x-axis is in log-scale. The noise value picked by maximizing ELBO for each method is indicated by the vertical dashed line with corresponding color. The vertical purple line indicates the value (0.28) learned by exact GP.

tation (a monthly precipitation dataset with $D = 3$) (Lyon, 2004; Lyon & Barnston, 2005) and Covtype (a tree cover dataset with $D = 54$) [2] (Blackard & Dean, 1999). Our goal is primarily to compare VNNGP to other variational methods; however, we include the additional baselines of **Exact GP**—utilizing the methodology of Wang et al. (2019))—and **sgGP** (Chen et al., 2020)—which is a non-variational nearest neighbor benchmark. We do not compare to the Amortized Inference GP since it is not amenable to kernel learning. For VNNGP and SWSGP, we tune the number of nearest neighbors K in $\{32, 256\}$. Additionally, we include two variants of the SWSGP method: with SWSGP-N we set all inducing points at observed locations, and with SWSGP-M we allow using $M < N$ inducing points – the latter is what Tran et al. (2021) originally experimented with. In particular for SWSGP-M, we recompute nearest neighbors every epoch and we tune M in $\{1024, 2048, 4096, 8192\}$. For sgGP, the subsampled datasets are constructed by selecting a random point $\mathbf{x}, y$ and its 15 nearest neighbors as in Chen et al. (2020). Each dataset is randomly split to 64% training, 16% validation and 20% testing sets. Precipitation dataset uses a student-t likelihood and Covtype dataset uses a bernoulli likelihood, otherwise a Gaussian likelihood is used. All kernels are Matern 5/2 with a separate lengthscale per dimension. We run each model with three random seeds. For each random seed, VNNGP applies different random ordering of inducing points. See appendix D.3.1 for more details.

We report test NLL in Table 1, and the result for test RMSE is included in Table 2 of appendix. From the tables, we make three key comparisons. (i) Exact GP and sgGP cannot run on problems with non-Gaussian likelihoods (Precipitation and Covtype) while variational methods are still viable. (ii) VNNGP substaintially outperforms other nearest neighbor methods. Specifically, we note that sgGP and SWSGP-N somtimes obtain a very high NLL, while SWSGP-M does not suffer from this issue. We suspect this result is attributed to that sgGP and SWSGP-N tend to learn a very small likelihood noise on certain datasets (we report the learned noise in Table 3 of appendix); Also recall from Section 4.2 that maximizing the ELBO of SWSGP-N leads to extremely small noise values. (iii) While there are a few datasets (Elevators and Bike) where SVGP excels, VNNGP outperforms SVGP especially on large-scale, spatial datasets. SVGP's performance is limited by using $M = 1024 \ll N$ inducing points, which is detrimental on these problems, while VNNGP has the ability to scale M to N with a sparse approximation.

Finally, we note that both VNNGP and SWSGP-N use much more parameters than SVGP. It is natural to ask whether this fact will impede their convergence speeds. In Figure 6, we plot the training loss versus number of training iterations for SVGP, VNNGP and SWSGP-N on Bike and UKHousing. We observe that all three methods reach convergence by the end of training. Moreover, VNNGP's convergence speed is comparable to SVGP. This result demonstrates that the larger number of model parameters does not hurt VNNGP's optimization process, making

[2] The task for Covtype is predicting whether the primary tree cover at a given location is pine trees or other types of trees. Despite this dataset is not 2 or 3-dimensional, its features essentially codify location information. Therefore we categorize Covtype as a spatial dataset.

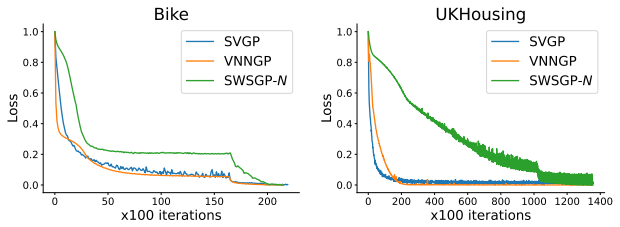

Figure 6. Training loss versus iterations on Bike and UKhousing. The training loss is averaged over every 100 iterations and all loss values are standardized. Despite VNNGP has more parameters, its convergence speed is comparable to SVGP.

it a practical model to use.

5. Discussion

In this work, we propose VNNGP, a scalable GP method that combines a sparse precision approximation with variational inference. This sparse approximation allows VNNGP to use orders of magnitude more inducing points than other variational methods. We perform an extensive empirical evaluation and show that VNNGP obtains strong performance compared to other baselines.

We consider two extensions for future works. Similar to Liu & Liu (2019) and Jafrasteh et al. (2021), we can apply amortized learning to variational parameters. This could greatly reduce the number of parameters and potentially accelerate optimization. Another extension is to select nearest neighbors using metrics other than Euclidean distance. For example, Kang & Katzfuss (2021) suggest that the prior covariance function can be a metric to select the nearest neighbor set. This could enhance VNNGP's accuracy on problems with anisotropic or periodic covariance structure.

References

Álvarez, M., Luengo, D., Titsias, M., and Lawrence, N. D. Efficient multioutput Gaussian processes through variational inducing kernels. In *Artificial Intelligence and Statistics*, pp. 25–32, 2010.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. Understanding probabilistic sparse gaussian process approximations. In *Advances in Neural Information Processing Systems*, 2016.

Blackard, J. A. and Dean, D. J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

Bui, T. D. and Turner, R. E. Tree-structured gaussian process approximations. *Advances in Neural Information Processing Systems*, 27, 2014.

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.

Chen, H., Zheng, L., Al Kontar, R., and Raskutti, G. Stochastic gradient descent in correlated settings: A study on gaussian processes. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Csató, L., Fokoué, E., Opper, M., Schottky, B., and Winther, O. Efficient approaches to gaussian process classification. In *Advances in neural information processing systems*, pp. 251–257. Citeseer, 2000.

Damianou, A. and Lawrence, N. D. Deep gaussian processes. In *Artificial intelligence and statistics*, pp. 207–215. PMLR, 2013.

Datta, A. Nearest-neighbor sparse cholesky matrices in spatial statistics. *Wiley Interdisciplinary Reviews: Computational Statistics*, pp. e1574, 2021a.

Datta, A. Sparse cholesky matrices in spatial statistics. *arXiv preprint arXiv:2102.13299*, 2021b.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016a.

Datta, A., Banerjee, S., Finley, A. O., Hamm, N. A., and Schaap, M. Nonseparable dynamic nearest neighbor gaussian process models for large spatio-temporal data with an application to particulate matter analysis. *The annals of applied statistics*, 10(3):1286, 2016b.

Finley, A. O., Datta, A., Cook, B. C., Morton, D. C., Andersen, H. E., and Banerjee, S. Applying nearest neighbor gaussian processes to massive spatial data sets: Forest canopy height prediction across tanana valley alaska. *arXiv preprint arXiv:1702.00434*, 535, 2017.

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. Efficient algorithms for bayesian nearest neighbor gaussian processes. *Journal of Computational and Graphical Statistics*, 28(2):401–414, 2019.

Gibbs, M. N. and MacKay, D. J. Variational gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, 2000.

Gramacy, R. B. and Apley, D. W. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

Guinness, J. Permutation and grouping methods for sharpening gaussian process approximations. *Technometrics*, 60(4):415–429, 2018.

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

Hensman, J., Matthews, A. G. d. G., Filippone, M., and Ghahramani, Z. Mcmc for variationally sparse gaussian processes. *arXiv preprint arXiv:1506.04000*, 2015.

Hensman, J., Durrande, N., Solin, A., et al. Variational fourier features for gaussian processes. *J. Mach. Learn. Res.*, 18(1):5537–5588, 2017.

Jafrasteh, B., Villacampa-Calvo, C., and Hernández-Lobato, D. Input dependent sparse gaussian processes. *arXiv preprint arXiv:2107.07281*, 2021.

Jankowiak, M. and Pleiss, G. Scalable cross validation losses for gaussian process models. *arXiv preprint arXiv:2105.11535*, 2021.

John, S. and Hensman, J. Large-scale cox process inference using variational fourier features. In *International Conference on Machine Learning*, pp. 2362–2370, 2018.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Kang, M. and Katzfuss, M. Correlation-based sparse inverse cholesky factorization for fast gaussian-process inference. *arXiv preprint arXiv:2112.14591*, 2021.

Katzfuss, M., Guinness, J., et al. A general framework for vecchia approximations of gaussian processes. *Statistical Science*, 36(1):124–141, 2021.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. Analyzing nonstationary spatial data using piecewise gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668, 2005.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. R. Inter-domain gaussian processes for sparse inference using inducing features. In *NIPS*, volume 22, pp. 1087–1095. Citeseer, 2009.

Liu, L. and Liu, L. Amortized variational inference with graph convolutional networks for gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2291–2300. PMLR, 2019.

Lyon, B. The strength of el niño and the spatial extent of tropical drought. *Geophysical Research Letters*, 31(21), 2004.

Lyon, B. and Barnston, A. G. Enso and the spatial extent of interannual precipitation extremes in tropical land areas. *Journal of Climate*, 18(23):5095–5109, 2005.

Nguyen-Tuong, D., Peters, J., and Seeger, M. Local gaussian process regression for real time online model learning and control. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pp. 1193–1200, 2008.

Park, C. and Apley, D. Patchwork kriging for large-scale gaussian process regression. *The Journal of Machine Learning Research*, 19(1):269–311, 2018.

Pleiss, G., Jankowiak, M., Eriksson, D., Damle, A., and Gardner, J. Fast matrix square roots with applications to gaussian processes and bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

Potapczynski, A., Wu, L., Biderman, D., Pleiss, G., and Cunningham, J. P. Bias-free scalable gaussian processes via randomized truncations. *arXiv preprint arXiv:2102.06695*, 2021.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

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. PMLR, 2018.

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.

Stein, M. L., Chi, Z., and Welty, L. J. Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66 (2):275–296, 2004.

Taylor-Rodriguez, D., Finley, A. O., Datta, A., Babcock, C., Andersen, H.-E., Cook, B. D., Morton, D. C., and Banerjee, S. Spatial factor models for high-dimensional and large spatial data: An application in forest variable mapping. *Statistica Sinica*, 29:1155, 2019.

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

Tran, G.-L., Milios, D., Michiardi, P., and Filippone, M. Sparse within sparse gaussian processes using neighbor information. In *International Conference on Machine Learning*, pp. 10369–10378. PMLR, 2021.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. A framework for interdomain and multioutput gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

Vecchia, A. V. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312, 1988.

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 Nickisch, H. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pp. 1775–1784, 2015.

Wu, L., Miller, A., Anderson, L., Pleiss, G., Blei, D., and Cunningham, J. Hierarchical inducing point gaussian process for inter-domian observations. In *International Conference on Artificial Intelligence and Statistics*, pp. 2926–2934. PMLR, 2021.

A. Computing the ELBO for VNNGP

In this section, we derive the exact computation of the ELBO for VNNGP and analyze the computational complexity.

Following Eq 12, and using the derivations from Eq 13 and Eq 16, we have

$$\mathcal{L}_{\text{VNNGP}} = \underbrace{\sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(y_i|f_i)]}_{\text{data likehood}} - \underbrace{\text{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right]}_{\text{KL divergence}}, \tag{22}$$

$$= \underbrace{\sum_{i=1}^{N} \int q(f_i) \log p(y_i|f_i) df_i}_{\text{data likelihood}} - \underbrace{\sum_{j=1}^{M} \mathbb{E}_{q(\mathbf{u}_{n(j)})} \left[\text{KL}\left[q(u_j) \parallel p(u_j|\mathbf{u}_{n(j)})\right]\right]}_{\text{kl divergence}}. \tag{23}$$

where

$$q(f_i) = \mathbb{E}_{q(\mathbf{u})}\left[p(f_i|\mathbf{u}_{n(i)})\right] \tag{24}$$

$$= \mathcal{N}(f_i|\mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1}\mathbf{m}_{n(i)}, k_{i,i} - \mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1}\mathbf{k}_{n(i),i} + \mathbf{k}_{n(i),i}^\top \mathbf{S}_{n(i)}\mathbf{k}_{n(i),i}), \tag{25}$$

and $\mathbf{S}_{n(i)}$ is a diagonal matrix with diagonal elements being $\mathbf{s}_{n(i)}$.

- The data likelihood term is tractable for Gaussian likelihood; otherwise, techniques such as MCMC sampling are required. The major complexity in evaluating each expected likelihood term comes from computing $q(f_i)$, which essentially resides at inverting the $K \times K$ matrix $\mathbf{K}_{n(i),n(i)}$ which takes $O(K^3)$ time.
- To compute the KL divergence term, we first compute each conditional KL term. Given fixed $\mathbf{u}_{n(j)}$,

$$\text{KL}\left[q(u_j) \parallel p(u_j|\mathbf{u}_j)\right] = \text{KL}\left[\mathcal{N}(u_j|m_j, s_j) \parallel \mathcal{N}(u_j|\mathbf{b}_j^\top \mathbf{u}_{n(j)}, f_j)\right] \tag{26}$$

$$= \frac{1}{2}\left[\log f_j - \log s_j - 1 + f_j^{-1}s_j + f_j^{-1}(m_j - \mathbf{b}_j^\top \mathbf{u}_{n(j)})^2\right], \tag{27}$$

  where $f_j = k_{j,j} - \mathbf{k}_{n(j),j}^\top \mathbf{K}_{n(j),n(j)}^{-1}\mathbf{k}_{n(j),j}$ and $\mathbf{b}_j = \mathbf{K}_{n(j),n(j)}^{-1}\mathbf{k}_{n(j),j}$. Integrating over $q(\mathbf{u}_{n(j)})$ and summing over inducing points we obtain,

$$\text{kl divergence} = \frac{1}{2}\sum_{j=1}^{M} \log f_j - \log s_j - 1 + f_j^{-1}\left[s_j + (\mathbf{b}_j^2)^\top \mathbf{s}_{n(j)} + (m_j - \mathbf{b}_j^\top \mathbf{m}_{n(j)})^2\right], \tag{28}$$

  where $\mathbf{b}_j^2$ denotes the element-wise squared of vector $\mathbf{b}_j$. Computing each summation term in the KL divergence requires inverting an $K \times K$ matrix, i.e. $\mathbf{K}_{n(j),n(j)}$, which is $O(K^3)$ complexity.

Therefore, the overall complexity of estimating the ELBO using a mini-batch of N_b data points and a mini-batch of M_b inducing points is $O(N_b K^3 + M_b K^3)$.

B. Extension to more expressive variational family

We now consider more expressive variational approximations and discuss how to mini-batch the VNNGP's ELBO computation. We assume the following variational family:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top) \tag{29}$$

$$q(\mathbf{f}|\mathbf{u}) = \prod_{i=1}^{N} p(f_i|\mathbf{u}_{n(i)}) \tag{30}$$

where $\tilde{\mathbf{L}}$ is the lower-triangular Cholesky factor for variational posterior covariance (potentially full-rank). The mean-field case discussed in appendix A is a special case where we make $\tilde{L}$ diagonal.

Similarly, the ELBO under this variational family decomposes into the data likelihood term and the KL divergence term as in Eq 23 with the modification that

$$q(f_i) = \mathbb{E}_{q(\mathbf{u})}\left[p(f_i|\mathbf{u}_{n(i)})\right] \tag{31}$$

$$= \mathcal{N}(f_i|\mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1}\mathbf{m}_{n(i)}, k_{i,i} - \mathbf{k}_{n(i),i}^\top \mathbf{K}_{n(i),n(i)}^{-1}\mathbf{k}_{n(i),i} + \mathbf{k}_{n(i),i}^\top \tilde{\mathbf{L}}_{n(i)}\tilde{\mathbf{L}}_{n(i)}^\top \mathbf{k}_{n(i),i}). \tag{32}$$

- The likelihood computation is similar to the mean-field case in appendix A, except that we are now integrating over $q(f_i)$ in Eq 32 instead of $q(f_i)$ in Eq 25. The former involves a potentially full-rank $K \times K$ variational covariance $\tilde{\mathbf{L}}_{n(i)}\tilde{\mathbf{L}}_{n(i)}^\top$ while the latter involves a diagonal one.
- KL divergence.

  For mathematical convenicence, we consider the Cholesky composition of prior precision for inducing points $\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1} = \mathbf{L}^\top\mathbf{L}$ where $\mathbf{L}$ is a lower triangular matrix. Each row of $\mathbf{L}$ could be separately computed at $O(K^3)$ cost (see Datta (2021a) for details). The KL divergence can be computed as follows:

$$\mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right] = \mathrm{KL}\left[\mathcal{N}(\mathbf{m}, \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top) \parallel \mathcal{N}(0, (\mathbf{L}^\top\mathbf{L})^{-1})\right] \tag{33}$$

$$= \frac{1}{2}[-2\log|\mathbf{L}| - 2\log|\tilde{\mathbf{L}}| - M + \mathrm{tr}(\mathbf{L}^\top\mathbf{L}\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top) + \mathbf{m}^\top\mathbf{L}^\top\mathbf{L}\mathbf{m}] \tag{34}$$

  Here we take a special treatment of the trace term:

$$\mathrm{tr}(\mathbf{L}^\top\mathbf{L}\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top) = \mathrm{tr}(\mathbf{L}\tilde{\mathbf{L}}(\mathbf{L}\tilde{\mathbf{L}})^\top) \text{ (by cyclic property of trace)} \tag{35}$$

$$= \sum_{i=1}^M\sum_{j=1}^M (\mathbf{L}\tilde{\mathbf{L}})_{ij}^2 \text{ (by definition of trace)} \tag{36}$$

$$= \sum_{i=1}^M\sum_{j=1}^M\sum_{k=1}^M L_{ij}^2\tilde{L}_{jk}^2 \text{ (expanding the } (\mathbf{L}\tilde{\mathbf{L}}_{ij}) \text{ terms)} \tag{37}$$

$$= \sum_{i=1}^M\sum_{j=1}^M \sum_{k\in n(i)\cup\{i\}} \mathbf{L}_{ik}^2\tilde{\mathbf{L}}_{kj}^2 \text{ (using row-sparsity of } \mathbf{L}) \tag{38}$$

  Now we re-write the KL by decomposing each term:

$$\mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right] = \frac{1}{2}[-2\sum_{i=1}^M \log L_{ii} - 2\sum_{i=1}^M \tilde{L}_{ii} - M + \sum_{i=1}^M\sum_{j=1}^M \sum_{k\in n(i)\cup\{i\}} \mathbf{L}_{ik}^2\tilde{\mathbf{L}}_{kj}^2 + \sum_{i=1}^M(\sum_{k\in n(i)\cup\{i\}} L_{ik}m_k)^2] \tag{39}$$

$$= \frac{1}{2}\sum_{i=1}^M[-2\log L_{ii} - 2\tilde{L}_{ii} - 1 + \sum_{k\in n(i)\cup\{i\}} \mathbf{L}_{ik}^2\sum_{j=1}^M\tilde{\mathbf{L}}_{kj}^2 + (\sum_{k\in n(i)\cup\{i\}} L_{ik}m_k)^2], \tag{40}$$

  which again factorizes over inducing points $i = 1:M$. However, one should note that computing each KL term now scales $O(MK^3)$ due to the complexity in trace term $\sum_{k\in n(i)\cup\{i\}}\sum_{j=1}^M \mathbf{L}_{ik}^2\tilde{\mathbf{L}}_{kj}^2$. One can alleviate this issue by (i) obtaining a stochastic estimate for the trace term (i.e. sub-sample j) (ii) introduce sparsity structure in $\tilde{\mathbf{L}}$ / variational posterior.

In summary, we explore the possibility of extending VNNGP to more expressive variational family. We show that under the full-rank variational family, the VNNGP's ELBO still factorizes over data points and inducing points. Computing each term now takes more computations and but techniques can be applied to improve the scalability.

C. KL divergence for SWSGP

Here we re-state and prove the two claims about KL divergence for SWSGP in Section 3.4.

Consider a fixed prior $p(\mathbf{u})$ and a fixed variational posterior $q(\mathbf{u})$ over M inducing points $\mathbf{u}$. Then we claim that

1. SWSGP's KL divergence is always underestimating, if not equal to, SVGP's KL divergence. That is, for any subset $\mathcal{S} \in \{1, 2, \cdots, M\}$.

$$\mathrm{KL}\left[q(\mathbf{u}_{\mathcal{S}}) \parallel p(\mathbf{u}_{\mathcal{S}})\right] \leq \mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right], \tag{41}$$

2. The amount of underestimation depends on the size of the nearest neighbor set. More precisely, for any two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ such that $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \{1, \cdots, M\}$,

$$\mathrm{KL}\left[q(\mathbf{u}_{\mathcal{S}_1}) \parallel p(\mathbf{u}_{\mathcal{S}_1})\right] \leq \mathrm{KL}\left[q(\mathbf{u}_{\mathcal{S}_1}) \parallel p(\mathbf{u}_{\mathcal{S}_2})\right]. \tag{42}$$

The proof for the above claims will immediately follow once we prove the following proposition.

Proposition C.1. *Let $p(\mathbf{u})$ and $q(\mathbf{u})$ be two distributions for an M-dimensional random variable $\mathbf{u}$, and let $\mathbf{u}_1$ be any sub-vector of $\mathbf{u}$, then*

$$\mathrm{KL}\left[q(\mathbf{u}_1) \parallel p(\mathbf{u}_1)\right] \leq \mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right]. \tag{43}$$

Proof. The proof relies on the chain rule of probability distribution, and the fact that the KL divergence is non-negative.

Denote $\mathbf{u}_2$ as the remaining sub-vector by taking $\mathbf{u}_1$ out of $\mathbf{u}$. It follows that

$$\begin{aligned}
&\mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right] && (44)\\
&= \int q(\mathbf{u}) \log \frac{q(\mathbf{u})}{p(\mathbf{u})} d\mathbf{u} && (45)\\
&= \int \int q(\mathbf{u}_1) q(\mathbf{u}_2|\mathbf{u}_1) \log \frac{q(\mathbf{u}_1) q(\mathbf{u}_2|\mathbf{u}_1)}{p(\mathbf{u}_1) p(\mathbf{u}_2|\mathbf{u}_1)} d\mathbf{u}_1 d\mathbf{u}_2 && (46)\\
&= I + II, && (47)
\end{aligned}$$

where

$$I \equiv \int q(\mathbf{u}_1) \log \frac{q(\mathbf{u}_1)}{p(\mathbf{u}_1)} \left(\int q(\mathbf{u}_2|\mathbf{u}_1) d\mathbf{u}_2 \right) d\mathbf{u}_1 = \mathrm{KL}\left[q(\mathbf{u}_1) \parallel p(\mathbf{u}_1)\right], \tag{48}$$

$$II \equiv \int q(\mathbf{u}_1) \left(\int q(\mathbf{u}_2|\mathbf{u}_1) \log \frac{q(\mathbf{u}_2|\mathbf{u}_1)}{p(\mathbf{u}_2|\mathbf{u}_1)} \right) d\mathbf{u}_1 = \int q(\mathbf{u}_1) \mathrm{KL}\left[q(\mathbf{u}_2|\mathbf{u}_1) \parallel p(\mathbf{u}_2|\mathbf{u}_1)\right] d\mathbf{u}_1. \tag{49}$$

Note that for each $\mathbf{u}_1$ fixed, $\mathrm{KL}\left[q(\mathbf{u}_2|\mathbf{u}_1) \parallel p(\mathbf{u}_2|\mathbf{u}_1)\right]$ defines a (conditional) KL divergence and is non-negative. Therefore, $\mathrm{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right] = I + II \geq I$, i.e. Eq 43 holds. □

D. Experiment details and additional results

D.1. KL divergence comparison

In Figure 7, we vary the setting in Section 4.2: different orderings of inducing points, different kernel functions, and real data v.s. synthetic data. From this figure, we observe similar patterns as from Figure 3.

D.2. Model selection

We include additional results for the model selection experiment in Section 4.2. We refer the models in Section 4.2 by VNNGP-N and SWSGP-N that set inducing points on observed locations. Here, we consider the alternative versions, VNNGP-M and SWSGP-M, which use $M \ll N$ inducing points for $M \in \{1024, 8192\}$, and the inducing point locations are learned during optimization. Correspondingly, the nearest neighbors are updated every training epoch for VNNGP and SWSGP. The other configurations are the same as Section 4.2 (especially, SVGP uses 1024 inducing points for all times). All inducing point locations are initialized by k-means clustering of observed locations.

Comparing Figure 8 to Figure 5, we can see that (1) VNNGP-M and VNNGP-N have consistent behaviors in model selection. (2) While maximizing the ELBO of SWSGP-N tends to favor a model with small likelihood noise, this is not the case for SWSGP-M. As the noise parameter becomes extremely small, the training ELBO of SWSGP-M decreases. As a result, optimizing SWSGP-M leads to a model that has a "medium" likelihood noise and a low test NLL. These observations suggest that maximizing the ELBO for SWSGP-N (and potentially for SWSGP-M with M close to N) can lead to undesired model selection outcome, while this does not hold for SWSGP-M with a medium size of inducing points.

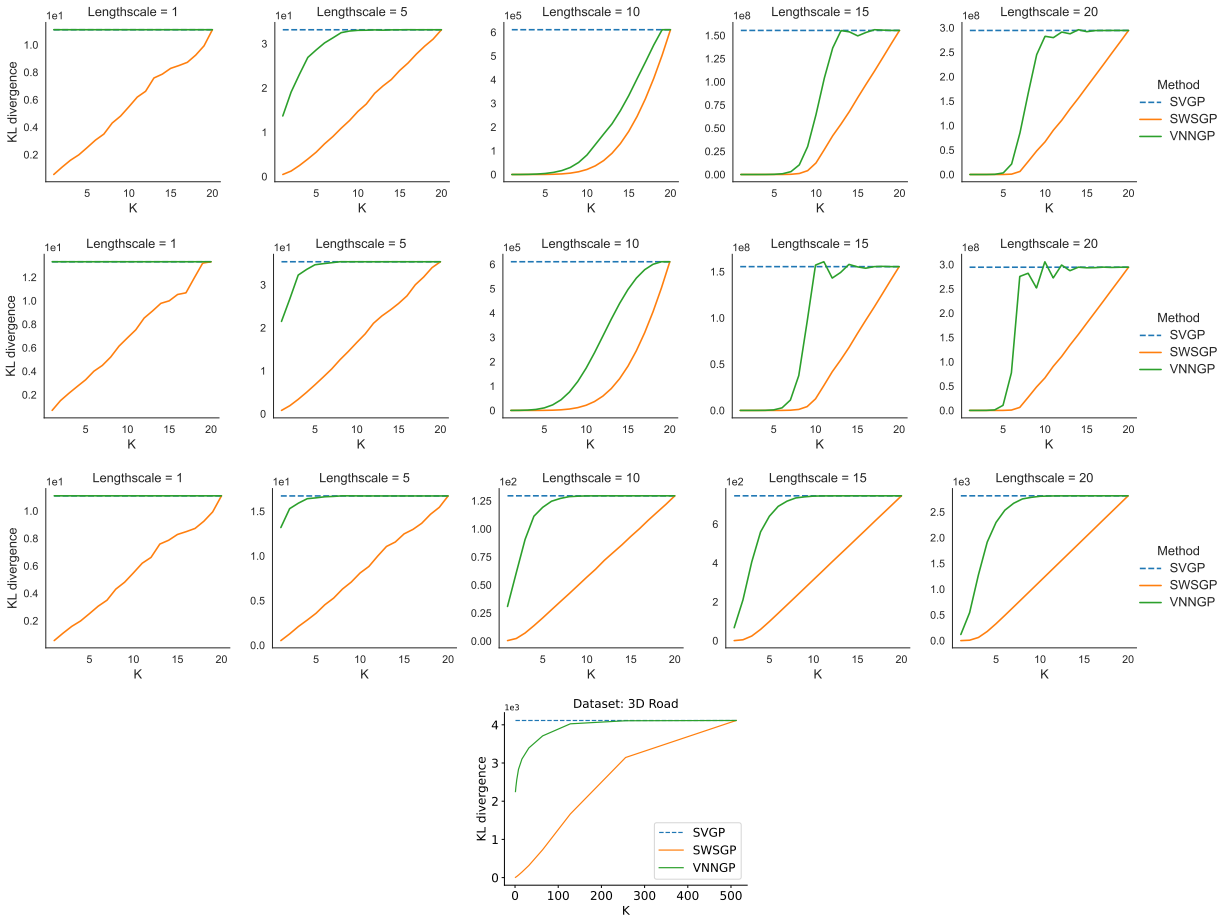

Figure 7. KL comparison under different settings than the setting of Figure 3. Row 1: A different random ordering of induing points. Row 2: Coordinate ordering of inducing points. Row 3: Matern (2.5) kernel. Row 4: Real dataset (3DRoad).

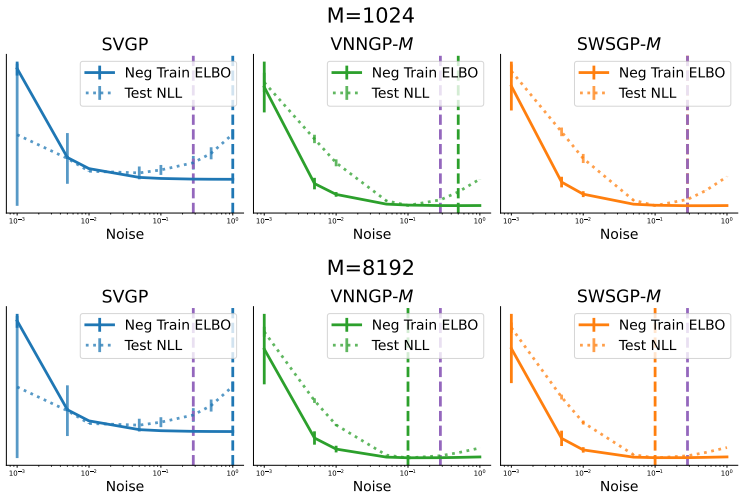

Figure 8. Negative ELBO in the last training iteration (solid line) and test set NLL (dotted line) as a function of the likelihood noise parameter by SVGP (blue), VNNGP (green) and SWSGP (orange) on UKHousing dataset. All ELBO and NLL values are scaled to $[0, 1]$. The x-axis (Noise) is in log-scale. The best noise parameter picked by maximizing ELBO for each method is indicated by the vertical dashed line with corresponding color. The vertical purple line indicates the noise parameter ($= 0.28$) learned by exact GP. **Left**: both VNNGP and SWSGP use $M = 1024$ inducing points. **Right**: both VNNGP and SWSGP use $M = 8192$ inducing point. For all cases, SVGP use 1024 inducing points.

D.3. Predictive performance on real dataset

Here we provide training details and additional results for Section 4.3.

D.3.1. Training details

For all methods, we use an Adam optimizer and a MultiStepLR scheduler dropping the learning rate by a factor of 10 at the 75% and 90% of the optimization iterations; all kernels are Matern 5/2 kernel with separate lengthscale per dimension; the kernel lengthscales, outputscale and likelihood noise parameter (if any) are all initialized as 0.6931, except that exact GP is initiliazed with lengthscale $= 0.01$ on 3DRoad dataset. All hyperparameters are picked by the best validation NLL.

All methods are trained with $\{300, 500\}$ iterations and learning rate of $\{0.005, 0.01\}$ for datasets of size below $50K$, and with $\{100, 300\}$ iterations and learning rate of $\{0.005, 0.001\}$ for above $50K$.

For sgGP, we train using minibatches of 16 data points. As suggested by Chen et al. (2020), the minibatches are constructed by sampling one training data point and selecting its 15 nearest neighbors. To accelerate optimization, we accumulate the gradients of 1,024 minibatches before per- forming an optimization step (these 1,024 minibatch updates can be performed in parallel, enabling GPU acceleration).

For SVGP, we use $M = 1024$ inducing points and a full-rank variational approximation. The inducing point locations are initilaized by k-means clustering of observed locations

VNNGP, SWSGP-M and SWSGP-N use a mean-field variational approximation. SWSGP-M chooses $M \in \{1024, 2048, 4096, 8192\}$, and the inducing point locations are initilaized by k-means clustering of observed locations. All three methods tune K in $\{32, 256\}$.

D.3.2. Predictive RMSE table

The test set RMSE results for reported in Table 2. From the table, we observe that VNNGP obtains comparable RMSE values to other state-of-the-arts and excels at some datasets.

			RMSE					
Dataset	N	D	ExactGP	sgGP	SVGP	SWSGP-M	SWSGP-N	VNNGP
PoleTele	9.6K	26	$.095 \pm .000$	$\mathbf{.092 \pm .002}$	$.113 \pm .000$	$.101 \pm .002$	$.098 \pm .003$	$\mathbf{.091 \pm .002}$
Elevators	10.6K	16	$\mathbf{.360 \pm .006}$	$\mathbf{.358 \pm .006}$	$\mathbf{.366 \pm .006}$	$.399 \pm .008$	$.456 \pm .001$	$.373 \pm .004$
Bike	11.1K	17	$.054 \pm .003$	$.400 \pm .153$	$\mathbf{.028 \pm .001}$	$.037 \pm .002$	$.042 \pm .006$	$.043 \pm .003$
Kin40K	25.6K	8	$.099 \pm .001$	$.458 \pm .076$	$.145 \pm .001$	$.118 \pm .002$	$\mathbf{.097 \pm .001}$	$\mathbf{.096 \pm .001}$
Protein	25.6K	9	$\mathbf{.533 \pm .002}$	$\mathbf{.534 \pm .006}$	$.633 \pm .003$	$.600 \pm .004$	$.592 \pm .005$	$.565 \pm .003$
KEGG	31.2K	20	$\mathbf{.085 \pm .002}$	$\mathbf{.087 \pm .003}$	$.089 \pm .001$	$\mathbf{.086 \pm .001}$	$\mathbf{.085 \pm .002}$	$\mathbf{.085 \pm .001}$
KEGGU	40.7K	26	$\mathbf{.117 \pm .000}$	$.122 \pm .001$	$.122 \pm .000$	$\mathbf{.118 \pm .001}$	$\mathbf{.116 \pm .001}$	$.118 \pm .000$
Precipitation	64.8K	3	—	—	$.635 \pm .004$	$.485 \pm .001$	$.506 \pm .001$	$\mathbf{.432 \pm .002}$
UKHousing	116K	2	$.402 \pm .001$	$.359 \pm .001$	$.582 \pm .003$	$.382 \pm .001$	$.498 \pm .000$	$\mathbf{.346 \pm .002}$
3DRoad	278K	3	$.165 \pm .000$	$.187 \pm .001$	$.329 \pm .001$	$.432 \pm .001$	$\mathbf{.097 \pm .017}$	$.298 \pm .064$
Covtype	372K	54	—	—	$.671 \pm .001$	$\mathbf{.657 \pm .001}$	$.670 \pm .001$	$.671 \pm .001$

Table 2. Test set RMSE (mean $\pm$ 1 standard error over 3 random seeds) on high-dimensional datasets (top 8 ones) and spatial datasets (bottom 4 ones).

D.3.3. Learned likelihood noise table

We report the learned value of likelihood noise for all models in Table 3. Note that Covtype dataset uses Bernoulli likelihood which does not have noise parameter, therefore it is not included.

We observe in Table 1 that sgGP and SWSGP-N can sometimes obtain a very high NLL values. We suspect that is due to they learn a particularly small likelihood noise on corresponding datasets. For example, sgGP's NLL is approximately 53 on Bike, and its learned noise is around 0; SWSGP-N's NLL is approximately 41 on KEGGU, and its learned noise is around 0.

			noise					
Dataset	N	D	ExactGP	sgGP	SVGP	SWSGP-M	SWSGP-N	VNNGP
PoleTele	9.6K	26	$.020 \pm .000$	$.002 \pm .000$	$.015 \pm .000$	$.005 \pm .000$	$.000 \pm .000$	$.002 \pm .000$
Elevators	10.6K	16	$.166 \pm .001$	$.102 \pm .001$	$.132 \pm .001$	$.062 \pm .000$	$.000 \pm .000$	$.067 \pm .000$
Bike	11.1K	17	$.016 \pm .000$	$.000 \pm .000$	$.002 \pm .000$	$.002 \pm .000$	$.000 \pm .000$	$.001 \pm .000$
Kin40K	25.6K	8	$.020 \pm .000$	$.000 \pm .000$	$.031 \pm .000$	$.009 \pm .000$	$.000 \pm .000$	$.001 \pm .000$
Protein	25.6K	9	$.182 \pm .003$	$.022 \pm .002$	$.417 \pm .001$	$.339 \pm .001$	$.000 \pm .000$	$.018 \pm .000$
KEGG	31.2K	20	$.019 \pm .000$	$.006 \pm .000$	$.008 \pm .000$	$.007 \pm .000$	$.000 \pm .000$	$.005 \pm .000$
KEGGU	40.7K	26	$.023 \pm .000$	$.013 \pm .000$	$.015 \pm .000$	$.014 \pm .000$	$.000 \pm .000$	$.013 \pm .000$
Precipitation	64.8K	3	—	—	$.182 \pm .001$	$.030 \pm .000$	$.000 \pm .000$	$.005 \pm .000$
UKHousing	116K	2	$.280 \pm .001$	$.112 \pm .000$	$.345 \pm .002$	$.144 \pm .000$	$.015 \pm .000$	$.070 \pm .003$
3DRoad	278K	3	$.584 \pm .000$	$.001 \pm .000$	$.126 \pm .000$	$.203 \pm .001$	$.000 \pm .000$	$.001 \pm .000$
Covtype	372K	54	—	—	$.000 \pm .000$	$.000 \pm .000$	$.000 \pm .000$	$.000 \pm .000$

Table 3. The learned likelihood noise (mean $\pm$ 1 standard error over 3 random seeds) on high-dimensional datasets (top 8 ones) and spatial datasets (bottom 3 ones).