LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-827489

Correspondence of NNGP Kernel and the Matérn Kernel

A. L. Muyskens, B. W. Priest, I. R. Goumiri, M. D. Schneider

October 4, 2021

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Correspondence of NNGP Kernel and the Matérn Kernel *

Amanda Muyskens†, Benjamin W. Priest , Imène R. Goumiri , and Michael D. Schneider

Abstract. Kernels representing limiting cases of neural network architectures have recently gained popularity. However, the application and performance of these new kernels compared to existing options, such as the Matérn kernel, is not well studied. We take a practical approach to explore the neural network Gaussian process (NNGP) kernel and its application to data in Gaussian process regression. We first demonstrate the necessity of normalization to produce valid NNGP kernels and explore related numerical challenges. We further demonstrate that the predictions from this model are quite inflexible, and therefore do not vary much over the valid hyperparameter sets. We then demonstrate a surprising result that the predictions given from the NNGP kernel correspond closely to those given by the Matérn kernel under specific circumstances, which suggests a deep similarity between overparameterized deep neural networks and the Matérn kernel. Finally, we demonstrate the performance of the NNGP kernel as compared to the Matérn kernel on three benchmark data cases, and we conclude that for its flexibility and practical performance, the Matérn kernel is preferred to the novel NNGP in practical applications.

Key words. kernel, prediction, interpolation, Matérn covariance, neural network, Gaussian process regression

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction.

Gaussian process regression is a convenient machine learning model that can flexibly approximate non-linear functions and provides principled uncertainty quantification of those non-linear predictions. A Gaussian process model is a continuous generalization of the normal distribution under the assumption that any finite set of data follows a joint multivariate normal distribution. Accordingly, the data model is fully specified by a mean and a covariance, which are often assumed to be parametric functions. The covariance function is commonly called a kernel, and the mean is usually assumed to be zero without a loss of generality. Defining new kernel functions is challenging in general since the forms must only produce positive definite covariance matrices to be a valid model. Functions like radial basis functions (RBF) and the larger class of Matérn kernels are popular choices that are avantageous based on their universality and flexibility to interpolate general data. In 1996, Neal showed that properly initialized single-layer feedforward neural networks converge to particular Gaussian process kernels in the infinite width limit [17]. More recent work has generalized this result to deep neural networks (DNNs) [11, 13]. Investigators have explored these dual representations in an attempt to learn about the behavior of different DNNs architectures [8, 5, 12, 1, 20]. We focus here on the NNGP kernel (also called the conjugate kernel), which represents DNNs of a specified depth with infinitely wide, fully connected layers, with independent identically distributed normal weights where only the last layer is trained.

In this paper, we take a practical approach to the NNGP kernel and explore the im-

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 with IM release number LLNL-TR-827489. Funding for this work was provided by LLNL Laboratory Directed Research and Development grant 19-SI-004.

†LLNL (muyskens1@llnl.gov).

plications of employing this kernel in model prediction. Prediction from Gaussian process models (kriging) is derived from the conditional multivariate normal distribution and can be most simply described as a weighted average of the training observations. These weights are determined as a function of the kernel matrix, whose values are determined by the set of hyperparameters. Therefore, we explore the effect of the hyperparameters of these kernels by computing and comparing their kriging weights directly. We compare kriging weights from the NNGP to those derived from the classical Matérn covariance function. We demonstrate a surprising practical equivalence in prediction between these NNGP kernels and the Matérn covariance kernel, and demonstrate the accuracy of these kernels on three example benchmark datasets.

2. Background: Gaussian process Regression.

We assume a linear model for a univariate response vector y observed at training locations $X_{\text{train}} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T]^T$ to be defined as $y = (y(x_1), y(x_2), ..., y(x_n))^T$ for n training observations.

We will assume that $\mathbf{f} \in \mathbb{R}^n$ constitute evaluations of a continuous, surrogate discrimination function $f_\theta : \mathbb{R}^L \to \mathbb{R}$ on X_{train}. We further assume that $\mathbf{y} \in \mathbb{R}^n$ are the "observed" realizations of f_θ on X_{train} perturbed by homoscedastic Gaussian noise $\boldsymbol{\epsilon}$. We seek to interpolate f_θ's response $\mathbf{f}_* \in \mathbb{R}^m$ to the unknown testing data $X^*_{\text{test}} = [\mathbf{x}_1^{*T}, \ldots, \mathbf{x}_m^{*T}]^T$.

The GP assumption on f_θ amounts to the assertion that $f_\theta \sim \mathcal{GP}(\mathbf{0}, k_\theta(\cdot,\cdot))$, where k_θ is a positive semidefinite kernel function with parameters θ. $f_\theta \sim \mathcal{GP}(\mathbf{0}, k_\theta(\cdot,\cdot))$ imposes the following Bayesian prior model on $\mathbf{f}$, the true evaluations of f on X_{train}:

$$\begin{aligned} \frac{\mathbf{y}}{\sigma} &= \mathbf{f} + \boldsymbol{\epsilon}, \\ \mathbf{f} &= [f_\theta(\mathbf{x}_1), \ldots, f_\theta(\mathbf{x}_n)]^\top \sim \mathcal{N}(\mathbf{0}, K_{\mathbf{ff}}), \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \tau^2 I_n). \end{aligned} \tag{2.1}$$

Here $K_{\mathbf{ff}}$ is an $n \times n$ positive definite covariance matrix on the training data whose (i,j)th element is $k_\theta(\mathbf{x}_i, \mathbf{x}_j)$, and τ^2 is the variance of the unbiased homoscedastic noise. The definition of GP regression then specifies that the joint distribution of all training and testing responses $\mathbf{y}$ and $\mathbf{f}^*$ is given by

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(0, \sigma^2 \begin{bmatrix} K_{\mathbf{ff}} + \tau^2 I_n & K_{\mathbf{f}*} \\ K_{*\mathbf{f}} & K_{**} \end{bmatrix}\right). \tag{2.2}$$

Here $K_{\mathbf{f}*} = K_{*\mathbf{f}}^\top$ is the cross-covariance matrix between the training and testing data; that is, the (i,j)th element of K_{f*} is $k(\mathbf{x}_i, \mathbf{x}_j^*)$. Similarly, K_{**} is the covariance matrix of the testing data, and has (i,j)th element $k_\theta(\mathbf{x}_i^*, \mathbf{x}_j^*)$. Finally, we are able to compute the posterior distribution of the testing response $\mathbf{f}^*$ on X^*_{test} as

$$\begin{aligned} \mathbf{f}^* \mid X_{\text{train}}, X^*_{\text{test}}, \mathbf{y} &\sim \mathcal{N}(\bar{\mathbf{f}}^*, \sigma^2 C), \\ \bar{\mathbf{f}}^* &\equiv K_{*\mathbf{f}}(K_{\mathbf{ff}} + \tau^2 I_n)^{-1}\mathbf{y}, \\ C &\equiv K_{**} - K_{*\mathbf{f}}(K_{\mathbf{ff}} + \tau^2 I_n)^{-1} K_{\mathbf{f}*}. \end{aligned} \tag{2.3}$$

The quantity we refer to as the kringing weights (H) is the matrix (or vector if X^*_{test} is one location) applied to the data vector $\mathbf{y}$ in order to obtain the predictions as

$$H = K_{*\mathbf{f}}(K_{\mathbf{ff}} + \tau^2 I_n)^{-1}. \tag{2.4}$$

2.1. NNGP Kernel. A fully-connected DNN with M hidden layers and widths $\{n^\ell\}_{\ell=0}^M$ has parameters consisting of weight matrices $\left\{W^\ell \in \mathbb{R}^{n^\ell \times n^{\ell-1}}\right\}_{\ell=1}^M$ and biases $\left\{\mathbf{b}^\ell \in \mathbb{R}^\ell\right\}_{\ell=1}^M$. We initialize the weights and biases of our hypothetical DNN with i.i.d. $\mathcal{N}(0,1)$ variables. We use hyperparameters σ_a and σ_b (effectively, variance priors for the weight and bias variables) to modify the variance of these parameter initializations. The output of such a DNN on input $\mathbf{x}$ is $\mathbf{h}^M(\mathbf{x})$, computed recursively as

$$\begin{aligned} \mathbf{h}^1(\mathbf{x}) &= \frac{\sigma_a}{\sqrt{n^0}} W^1 \mathbf{x} + \sigma_b b^1, \\ \mathbf{h}^\ell(\mathbf{x}) &= \frac{\sigma_a}{\sqrt{n^{\ell-1}}} W^\ell \phi\left(\mathbf{h}^{\ell-1}(\mathbf{x})\right) + \sigma_b b^\ell, \end{aligned} \tag{2.5}$$

where here ϕ is an element-wise *activation function*. Translating the action of a DNN layer in the infinite width limit to kernel form requires obtaining a dual form of the nonlinearity ϕ given positive definite kernel matrix K. Several popular activation functions have known dual forms (see [3] for some examples). We will follow other research on this topic by focusing on the popular rectified linear unit (ReLU) activation, which fortunately has a known analytic dual form given by

$$\begin{aligned} V_{\phi_{\mathrm{ReLU}}}(K)(\mathbf{x},\mathbf{x}') &= \frac{\sqrt{K(\mathbf{x},\mathbf{x})K(\mathbf{x}',\mathbf{x}')}}{2\pi}\left(\sin c + (\pi - c)\cos c\right) \\ V_{\phi'_{\mathrm{ReLU}}}(K)(\mathbf{x},\mathbf{x}') &= \frac{1}{2\pi}(\pi - c) \\ c &= \arccos\left(\frac{K(\mathbf{x},\mathbf{x}')}{\sqrt{K(\mathbf{x},\mathbf{x})K(\mathbf{x}',\mathbf{x}')}}\right). \end{aligned} \tag{2.6}$$

Using this dual activation and the notation of Equation (2.5) and following the formulation of [21], we can express the NNGP recursively as

$$\begin{aligned} \Sigma^1(\mathbf{x},\mathbf{x}') &= \frac{\sigma_a^2}{n^0}\langle \mathbf{x},\mathbf{x}'\rangle + \sigma_b^2, \\ \Sigma^\ell(\mathbf{x},\mathbf{x}') &= \sigma_a^2 V_{\phi_{\mathrm{ReLU}}}(\Sigma^{\ell-1})(\mathbf{x},\mathbf{x}') + \sigma_b^2, \\ k_{(M,\sigma_a,\sigma_b)}(\mathbf{x},\mathbf{x}') &= \Sigma^M(\mathbf{x},\mathbf{x}'). \end{aligned} \tag{2.7}$$

2.2. Matérn Kernel. The posterior distribution given in Equation (2.3) depends on the choice of kernel function. We will analyze the Matérn kernel, which is a stationary and isotropic kernel that is commonly used in the spatial statistics GP literature due to its flexibility and favorable properties [18]. A general expression for the kernel is

$$k_{(\sigma^2,\nu,\ell)}(\mathbf{x},\mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{\ell}\right)^\nu B_\nu\left(\sqrt{2\nu}\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{\ell}\right), \tag{2.8}$$

where $\nu > 0$ is a smoothness parameter, $\ell > 0$ is a correlation-length scale hyperparameter, $\sigma^2 > 0$ is a scale parameter, Γ is the Gamma function, and $B_\nu(\cdot)$ is a modified Bessel function of the second kind. Note that as $\nu \to \infty$, the Matérn kernel converges pointwise to the popular radial basis function (RBF) kernel. We will compare the predictions of the Matérn and NNGP kernels because the former is considered to be state-of-the art in terms of flexibility and general applicability to realistic data.

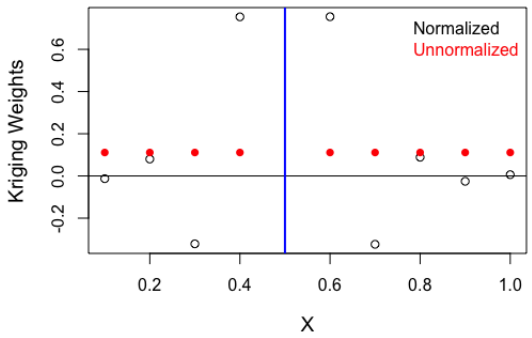

Figure 1. *Comparison of kriging weights from a 1-dimensional Gaussian process regression compared on data normalized to the unit hypershpere and unnormalized.*

3. Application of the NNGP Kernel.

In this section, we discuss the practical application of the NNGP kernel for Gaussian prcoess regression. We examine the common normalization scheme conventional in applying the NNGP as well as limit parameters σ_a and σ_b to sets that give valid covariance matrices. Next, we explore kriging weights given by the NNGP kernel under those valid parameter sets and compare the results to the kriging weights of the common Matérn kernel.

3.1. Normalization.

It is conventional (see [11, 13]) to embed the data on the unit hypershere prior to applying the NNGP kernel. Formally, the unit hypersphere embedding we employ is as follows. Much of the published research analyzing the NNGP involves image data, where such normalization amounts to normalizing the image intensity across all images. We will employ a slightly different normalization method to accomodate lower dimensional data. Let X_i be the i^{th} column of the raw training input data. Then create training data matrix $X^\star$ to have twice as many dimensions so that each column in X is two in $X^\star$ as

$$X^\star_{2i+1,2i+2} = [cos(X_i\pi), sin(X_i\pi)]\,. \tag{3.1}$$

Finally, these columns are additionally normalized by their L2-norms so

$$X_i = \frac{X_i^\star}{||X_i^\star||_2}. \tag{3.2}$$

Finally, define X_{train} and $X^\star_{test}$ to be the matrices containing these columns normalized by these two procedures.

Figure 1 demonstrates the difference in kriging weight for prediction at the vertical line produced from a 1-dimensional example for normalized and unormalized implementation of the NNGP kernel. Note that if the data is not normalized, the prediction in this case is approximately the mean of the training data.

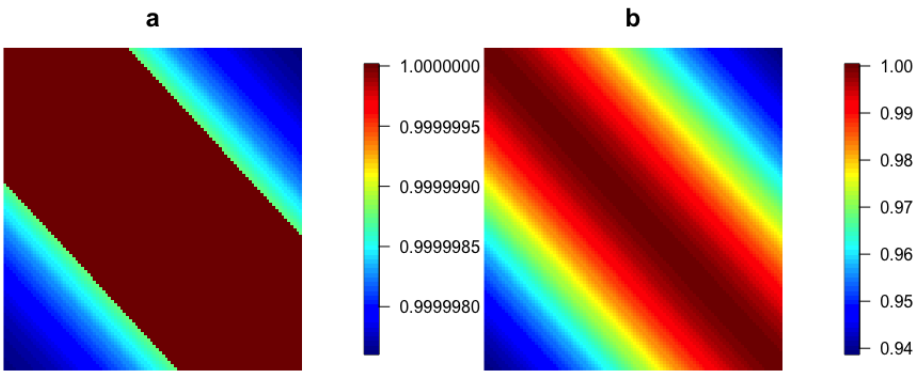

Figure 2. *a. Invalid NNGP kernel b. Valid NNGP Kernel*

3.2. Valid Hyperparameter Sets. Once the data is normalized as described, we examine ranges of the NNGP kernel hyperparameters σ_a and σ_b. Assuming that X_{train} is a 1-dimensional grid on the interval $(0, 1]$, we formed the NNGP kernel $k_{(M,\sigma_a,\sigma_b)}$ for hyperparameters on a 20×20 grid for $\sigma_a, \sigma_b \in [0.1, 2.0]$ for various depths. Some kernels generated were numerically invalid covaraince matrices because they were not positive definite and had off-diagonal covariance entries equal to the diagonal entry. This produces "flat" regions in the matrix where the correlation is numerically equal to 1 for non-zero distances. Examples of an invalid kernel matrix and a valid kernel matrix are in Figure 2. In the invalid kernel matrix, the correlation over the entire domain is extremely high, and although there is a wider range of correlation in the valid kernel example, data across the domain being correlated as a minimum of 0.94 is still unreasonably high for most realistic data.

Figure 3 evaluates which hyperparameter sets produce valid kernel matrices. As the depth of the corresponding NN increases, the region of hyperparameter space that produces degenerate matrices increases. Finally, Figure 4 demonstrates the variation in the kriging weights for all considered depths and parameters in Figure 3. Even in these diverse parameter settings, the kriging weights have very little variation and therefore will produce extremely similar predictions.

3.3. Kriging Weights Correspondence to Matérn Kernel. In this section we demonstrate the surprising correspondence between kriging weights produced from the NNGP kernel and the Matérn kernel with smoothness parameter $\nu = \frac{3}{2}$. Although the kriging weights (H) from the two kernels are extremely similar under certain conditions, this correspondence is not numerically exact, and actually the correlation function (k_θ) that produces the kernel matrices do not themselves correspond over the entire data range. Figure 5 provides an example of this phenomenon. Figure 5a shows that the NNGP and Matérn and NNGP correlation functions diverge as distance increases, and yet Figure 5b shows that the resulting kriging weights agree. There is significant agreement in the correlation function for the normalized distances less than 0.5.

Computationally efficient Gaussian process methods including [6] and [4] have exploited locality-induced sparsity in order to gain their computational efficiency. Further, high-frequency trends (local correlations) are more important to accurate prediction than low-frequency

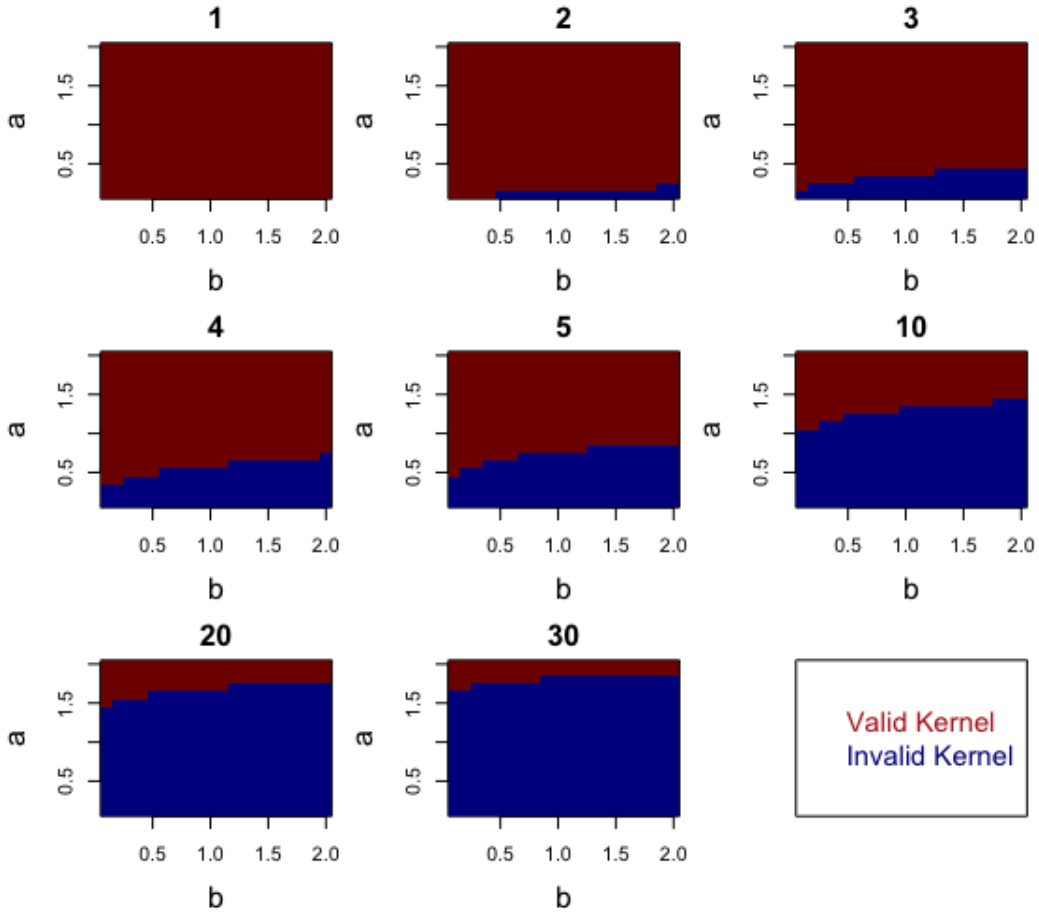

Figure 3. *Valid NNGP hyperparameter sets that produce positive definite kernel matrices for various depths for NNGP hyperparameters σ_a and σ_b.*

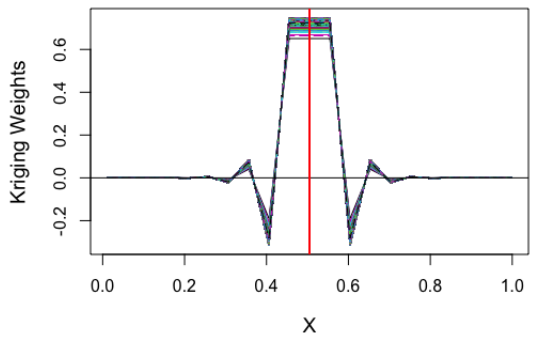

Figure 4. *Variation of kriging weights over valid NNGP hyperparameter sets and depths.*

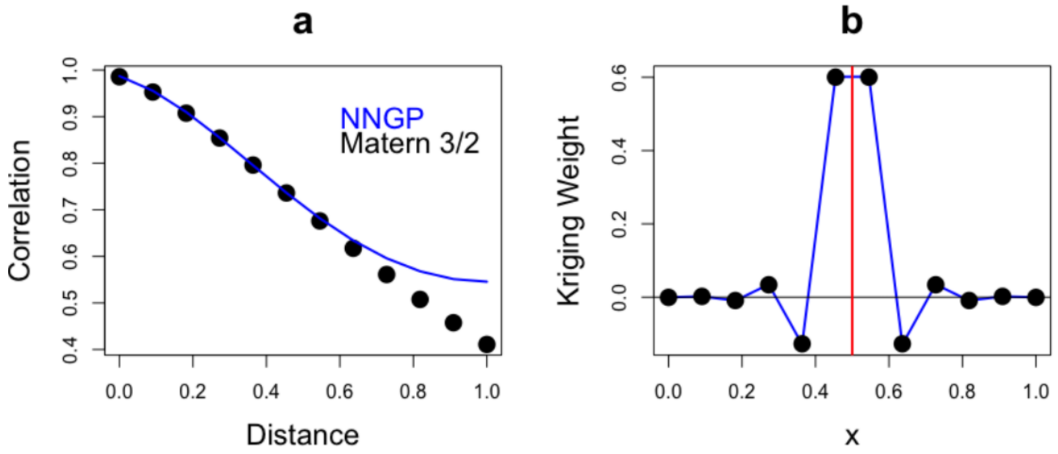

Figure 5. *Comparison of correlation functions (a) and kriging weights (b) of NNGP (blue) and Matérn* $\frac{3}{2}$ *(black) kernels in 1-D for prediction at* $x = 0.5$ *(red).*

trends (long-scale correlations) [18]. Our findings extend this knowledge, by demonstrating similar predictions are produced with only local correlation similarity.

As the number of observations increases, the similarity between the kringing weights also increases. As there are more observations normalized to $[0, 1]$, this implies that there are more observations with corresponding correlations. Using data generated both on grid (Figure 6), and generated using a quasi-random sobol sequence (Figure 7), we demonstrate the maximum absolute difference between the Matérn kriging weights ($\nu = \frac{3}{2}$) and those from the NNGP. In both cases, with data sizes at about 150, the largest difference in kriging weights is less than 0.00005.

Although we demonstrate the similarity between the kriging weights only from data in 1-dimension, a similar correspondence can be seen in higher dimensional data. The next section shows the practicality of this correspondence in higher dimensional examples. Also, [14] demonstrates that the Matérn kriging weights are only dependent on the ν parameter when the nugget parameter τ^2 is small (very close to 0). However, when τ^2 is large, the kriging weights also depend on the range parameter ρ. We omit the results here for similcity, but if τ^2 is large, there is still a correspondence in predictions, but ρ must also be appropriately selected. In summary, under large and densely-sampled data, predictions from the NNGP kernel are practically similar to those from the Matérn kernel with $\nu = \frac{3}{2}$. The next section will demonstrate similar these predictions are in several benchmark data cases.

4. Numerical Demonstrations.

In this section, we numerically demonstrate the importance of the conclusions of the previous section by applying Gaussian process regression with the NNGP kernel on several benchmark datasets.

Throughout this section, we will compare Gaussian processes with three kernel (k_θ) assumptions.

1. NNGP kernel with σ_a, σ_b varied, depth= 2
2. Matérn kernel with $\nu = \frac{3}{2}$ and $\rho = 1$

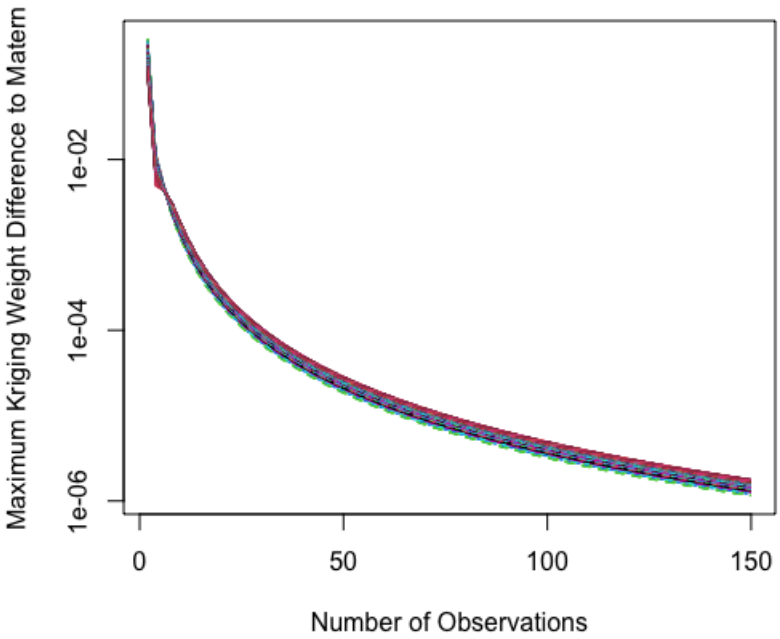

Figure 6. *Maximum absolute difference between kriging weights produced from the Matérn $\nu = \frac{3}{2}$ and those from the NNGP kernel in a grid in 1-D.*

3. Matérn kernel with ν, ρ varied

Note that we compare to the NNGP kernel with low depth so that the entire range of σ_a and σ_b yields a valid covariance model, but the results should not be very different if another depth was selected since the kriging weights, and therefore predictions, from various depths are similar (Figure 4). Also, we do not perform a grid search to optimize parameters on a training subset of the data, but instead report just the best performance of the testing data to demonstrate the best performance from each kernel. To compare the results of these three models, we consider several summary statistics. First, best accuracy of the models is compared via root mean squared error (minRMSE) of the model predictions.

$$minRMSE = \min_{\theta} \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{Y}_i - Y_i)^2}, \tag{4.1}$$

where $\hat{Y}_i = H_{i\theta}Y$. Similarly, we report the worst accuracy of each kernel (maxRMSE) as

$$maxRMSE = \max_{\theta} \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{Y}_i - Y_i)^2}. \tag{4.2}$$

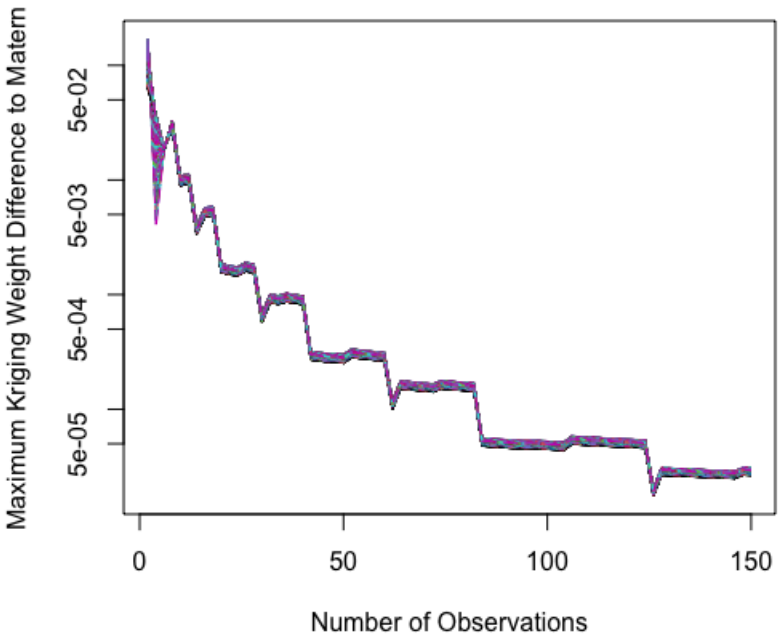

Figure 7. *Maximum absolute difference between kriging weights produced from the Matérn $\nu = \frac{3}{2}$ and those from the NNGP kernel sampled according to the quasi-random sobol sequence in 1-D.*

Next, we consider several statistics describing the differences in kriging weights between the best NNGP kernel and each kernel. Formally, define the best performing NNGP kernel parameter set to have corresponding kriging weights matrix H_{NNGP}, which is dimension $m \times n$ when we utilize n training data observations in order to predict at m testing locations. Define $\tilde{\theta}$ to be the parameter set for each kernel that minimizes the maximum absolute difference in the kriging weights as compared to H_{NNGP}. Then for each kernel type we report the maximum (maxdiff), minimum (mindiff), mean (meandiff), and standard deviation (sddiff) of the difference in the kriging weights as compared to those from the best-performing NNGP kernel.

$$\text{maxdiff} = \max_{n,m} |H_{\tilde{\theta}} - H_{NNGP}| \tag{4.3}$$

$$\text{mindiff} = min_{n,m} |H_{\tilde{\theta}} - H_{NNGP}| \tag{4.4}$$

$$\text{meandiff} = \frac{1}{nm} \sum_{i=1}^{nm} |H_{\tilde{\theta}} - H_{NNGP}| \tag{4.5}$$

$$\text{sddiff} = \frac{1}{nm-1}\sqrt{\sum_{i=1}^{nm} |H_{\tilde{\theta}} - H_{NNGP}|^2} \tag{4.6}$$

Finally, we look at the distribution of the differences across θ values for each kernel form. For each parameter set θ, we compute the maximum difference in the kriging weights produced as related to those from the mean kriging weight $\bar{H}$. Then we report summary statistics of that maximum difference over the grid search of values tested. These summaries demonstrate the variance in the kriging weights over various sets tests, ie a more flexible kernel form. Define n_θ to be the number of θ parameter sets for each kernel type. Then we report similar distributional statistics of the kriging weights defined as

$$\text{maxkw} = \max_{\theta} max|H_\theta - \bar{H}| \tag{4.7}$$

$$\text{minkw} = \min_{\theta} max|H_\theta - \bar{H}| \tag{4.8}$$

$$\text{meankw} = \frac{1}{nm}\sum_{i=1}^{nm} max|H_\theta - \bar{H}| \tag{4.9}$$

$$\text{sdkw} = \frac{1}{n_\theta - 1}\sqrt{\sum_{i=1}^{n_\theta} max(H_\theta - H_{NNGP})^2} \tag{4.10}$$

The standard deviation over 100 simulation iterations of each statistic is also reported.

4.1. Friedman Function Data. Data in this comparsion is simulated from the common surrogate model Friedman function defined as

$$Y_i = 10sin(\pi x_{1i} * x_{2i}) + 20(x_{3i} - 0.5)^2 + 10x_{4i} + 5x_{5i} + \epsilon_i, \tag{4.11}$$

where $\epsilon_i \sim N(0,1)$ for $x_i \in [0,1]$ for $i = 1,2,3,...5$. We sample 500 training locations according to a random Latin Hypercube design, and model the responses as follows:

$$Y_i \sim GP(X_i\beta, k_\theta(x_i, x_i)), \tag{4.12}$$

where k_θ is one of the three outline kernel models in this section. Example realizations from this function plotted by the input variables can be seen in Figure 8

4.2. MODIS Satellite Dataset. Our data are sourced from the numerical comparisons in [7] and can be downloaded at https://github.com/finnlindgren/heatoncomparison. It is land surface temperature data from latitude ranging from 34.29519 to 37.06811 and longitudes from -95.91153 to -91.28381 collected on August 4, 2016. The 148,309 observations were collected. on a paritially missing 500×300 grid. The cloud cover from August 6, 2016 was used in order to develop a realistic testing/training data split. There are 42,740 testing obbservations

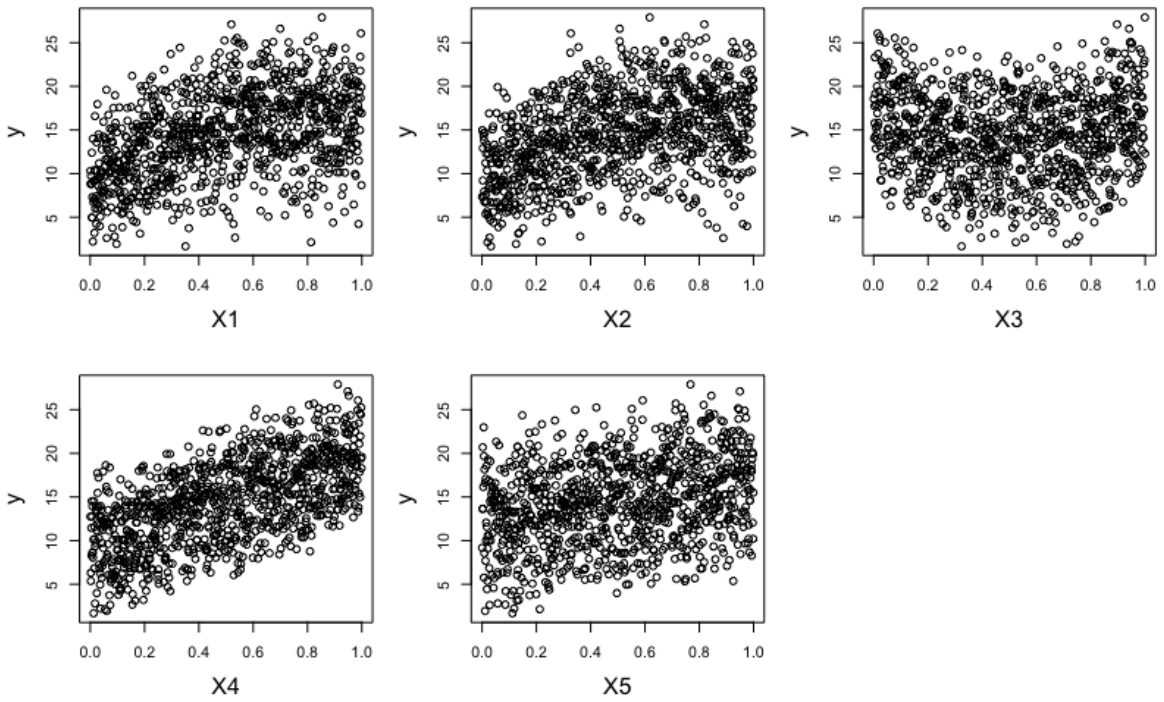

Figure 8. *Marginal plots of the responses from the Friedman function against each input variable.*

and 105,569 training observations with the pattern as seen in Figure 9. Since our comparison is meant to compare kernel functions, we fist subtract the Gaussian filtering mean from the data as in [15]. Then our comparison fits the residuals from this analysis as the ground truth. Both the training data and testing data are randomly downsampled so that kriging estimators can be implemented in typical laptop constraints. We randomly sample 500 training and 500 testing locations. The data is formally modeled

$$Y_i \sim GP(\mu_i, k_\theta(x_i, x_i)), \tag{4.13}$$

where k_θ is one of the three outline kernel models in this section, and μ_i is the moving window mean in [15]. We compare the performance of the aformentioned three models in Table 1.

4.3. Borehole Function. The borehole function ([19]) is a common benchmark function used to demonstrate computer simulation modeling ([2, 10]). It is meant to model the flow of water through a borehole, and includes parameters such as the radius of the borehole in meters (r_w), the hydraulic conductivity of the borehole in m/yr (K_w) and 6 other parmeters that effect the flow. This 8-dimensional funtion is defined as:

$$y = \frac{2\pi T_u[H_u - H_l]}{\log(\frac{r}{r_w})[1 + \frac{2LT_u}{\log(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}]} \tag{4.14}$$

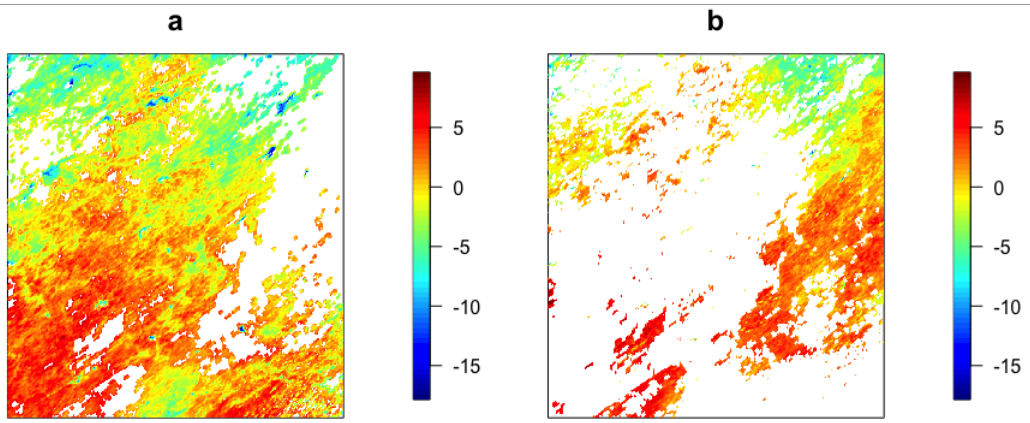

Figure 9. *a. Full training data b. Full testing data . We subset 500 random samples from each of these for data in a single simulation iteration.*

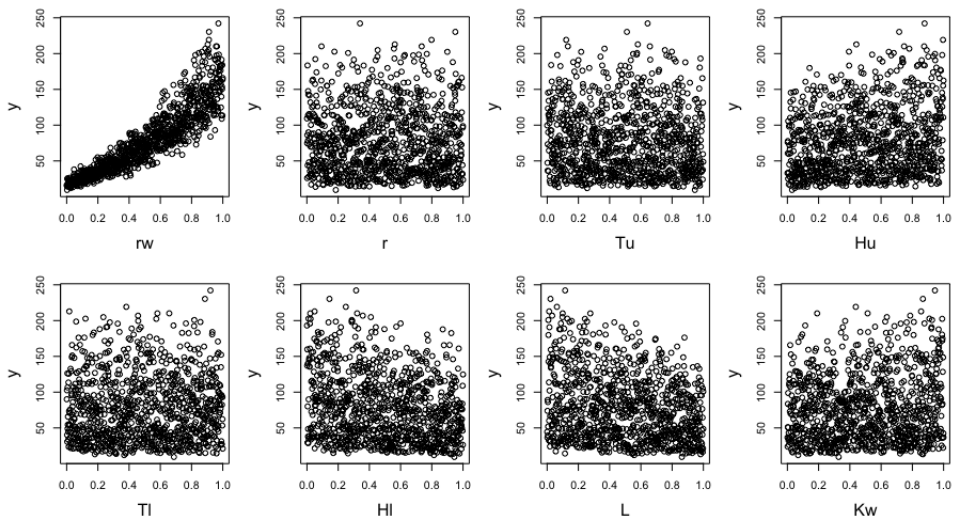

Figure 10. *Marginal plots of the responses from the borehole function against each input variable.*

where $r_w \in [0.05, 0.15]$, $H_u \in [990, 1100]$, $r \in [100, 5000]$, $H_l \in [700, 820]$, $T_u \in [63070.115600]$, $L \in [1120, 1680]$, $T_l \in [63.1, 116]$, $K_w \in [9855, 12045]$.

We formally model this function as we modeled the Friedman function with a linear mean. Formally,

$$Y_i \sim GP(X_i\beta, k_\theta(x_i, x_i)), \tag{4.15}$$

where k_θ is one of the three outline kernel models in this section. Example realizations from this function plotted by the input variables can be seen in Figure 10.

4.4. Results. Table 1 summarizes 100 iterations of these benchmark examples as outlined. The NNGP and Matérn $\nu = \frac{3}{2}$ kernels perform relatively similarly, but the Matérn model with

varied parameters has the smallest RMSE in all cases by a wide margin. Note that although we have demonstrated the convergence of the kriging weights under ideal conditions, this "correspondence" is not absolute in the sense the compared kernels yield similar but not identical predictions. In terms of maxRMSE, the Matérn $\nu = \frac{3}{2}$ produces the best worst-case prediction error so if one were to implement a kernel without optimization for testing purposes, this would be the best kernel selection in general. In the Friedman function and the borehole function data examples, our results are as expected where the kriging weights from the Matérn kernel are generally close to those of the NNGP kernel. However, we see a large deviance in the NNGP kriging weights in some iterations. When a matrix is close to singular (as some valid NNGP kernels are), when data is observed very closely, numerical difficulties can cause the kriging weights to become artificially large. This is clearly the case in the MODIS data example, and ultimately results in extremely poor performance for a few of the iterations. This numerical challenge is seen here in the NNGP kernel, but not the Matérn kernel results. An example iteration of the kriging weights from the various kernels are compared in Figure 11.

5. Discussion. In this manuscript, we have demonstrated the practical considerations of utilization of the NNGP kernel for Gaussian process regression. We have shown normalization to the unit hypersphere is necessary in order to obtain meaningful predictions. We have also shown that there are NNGP parameter sets that return degenerate covariance matrices, and that the regions of parameter space that have this numerical issue tend to increase in size as depth increases. We have demonstrated that for useable parameter combinations, when the data is sufficiently large, predictions from the NNGP kernel are approximately equivalent to those from the Matérn $\nu = \frac{3}{2}$ model.

We interpret this result to mean that theoretical well-converged neural networks in the infinite width limit are essentially Gaussian processes with the well-known Matérn kernel function. Determining uncertainty quantification (UQ) is currently a major research focus in the study of DNNs and other machine learning methods [9]. As predictions from GP models and idealized DNNs under certain architectural assumptions are the same, further research could demonstrate whether the UQ from the equivalent GP model could be realistically be applied to trained DNNs.

Further, it is generally accepted that DNNs demonstrate complex, non-local dependence in the data. However, the practical correspondence to the Matérn kernel challenges this idea, at least for the specific architecture the NNGP represents. In [15], it is shown that the kriging weights from Gaussian process regression with a Matérn are sparse in distance to the prediction location, meaning they depend most largely on nearest neighbors for similar predictions. Therefore, since the predictions from the NNGP are similar to those from the Matérn, in this case, they are also locally-dominated predictions.

Our work could be continued through the comparision of the performance of these kernels in higher dimensional cases. For example, [16] utilize a PCA reduction and GP regression in order to perform image classification. Even with this data reduction, the dimension of data fit is 50, which is much higher than the examples we have considered in this manuscript. This higher dimensional data is a more typical dataset for neural networks, and GP models are known to struggle in prediction in these cases.

Statistic	Data	NNGP	Matérn $\nu = \frac{3}{2}$	Matérn
minRMSE	Friedman	0.054 (0.005)	0.026 (0.007)	**0.005**(0.002)
	MODIS	4.294 (1.534)	4.892 (2.288)	**2.216** (0.126)
	Borehole	1.520 (0.210)	1.288 (0.394)	**0.406** (0.144)
maxRMSE	Friedman	0.063 (0.007)	0.026 (0.007)	4.198 (0.231)
	MODIS	4425.090(13059.737)	4.892 (2.288)	16.054 (11.756)
	Borehole	1.960 (0.354)	1.288 (0.394)	88.990 (5.556)
maxdiff	Friedman	0.000 (0.000)	0.299 (0.450)	0.225 (0.355)
	MODIS	0.000 (0.000)	8.719 (10.611)	7.659 (10.944)
	Borehole	0.000 (0.000)	0.209 (0.033)	0.201 (0.030)
mindiff	Friedman	0.000 (0.000)	1.2e-08 (1.3e-08)	1.4e-08 (1.6e-08)
	MODIS	0.000 (0.000)	1.8e-12 (1.8e-12)	1.9e-12 (2.6e-12)
	Borehole	0.000 (0.000)	5.7e-07 (5.3e-07)	8.0e-07 (6.8e-07)
meandiff	Friedman	0.000 (0.000)	0.003 (0.000)	0.003 (0.001)
	MODIS	0.000 (0.000)	0.004 (0.005)	0.011 (0.012)
	Borehole	0.000 (0.000)	0.011 (0.001)	0.012 (0.001)
sddiff	Friedman	0.000 (0.000)	0.005 (0.001)	0.006 (0.003)
	MODIS	0.000 (0.000)	0.072 (0.067)	0.102 (0.117)
	Borehole	0.000 (0.000)	0.012 (0.001)	0.013 (0.001)
maxkw	Friedman	0.526 (1.202)	0.000 (0.000)	1.289 (0.365)
	MODIS	116177.8(532722.5)	0.000 (0.000)	30.469 (17.109)
	Borehole	0.015 (0.004)	0.000 (0.000)	0.649 (0.139)
minkw	Friedman	1.3e-06 (4.0e-07)	0.000 (0.000)	2.6e-05 (5.6e-06)
	MODIS	1.3e-06 (1.9e-06)	0.000 (0.000)	7.4e-08 (1.4e-08)
	Borehole	1.5e-05 (4.2e-06)	0.000 (0.000)	4.7e-04 (8.0e-05)
meankw	Friedman	0.000 (0.000)	0.000 (0.000)	0.011 (0.000)
	MODIS	3.079 (10.018)	0.000 (0.000)	0.056 (0.017)
	Borehole	0.001 (0.000)	0.000 (0.000)	0.020 (0.001)
sdkw	Friedman	0.002 (0.004)	0.000 (0.000)	0.030 (0.001)
	MODIS	379.826(1637.020)	0.000 (0.000)	0.480 (0.205)
	Borehole	0.001 (0.000)	0.000 (0.000)	0.030 (0.002)

Table 1

Performance of various kernel models for the data examples described in Section 4. Parthenses contain the standard deviation of the estimate over 100 random iterative draws.

Additionally, NNGP is not the only GP kernel that has been identified corresponding to other neural network architectures. One example is the neural tangent kernel (NTK), which is similar in form to the NNGP kernel and is used in the study of training dynamics [8]. Furthermore, kernel correspondences to infinite width limits of other architecture types such as convolutional, recurrent, and graph neural networks have emerged in the literature [5, 20]. Further studies could explore these other kernels, and evaluate if there exists prediction equivalence to the predictions from these models.

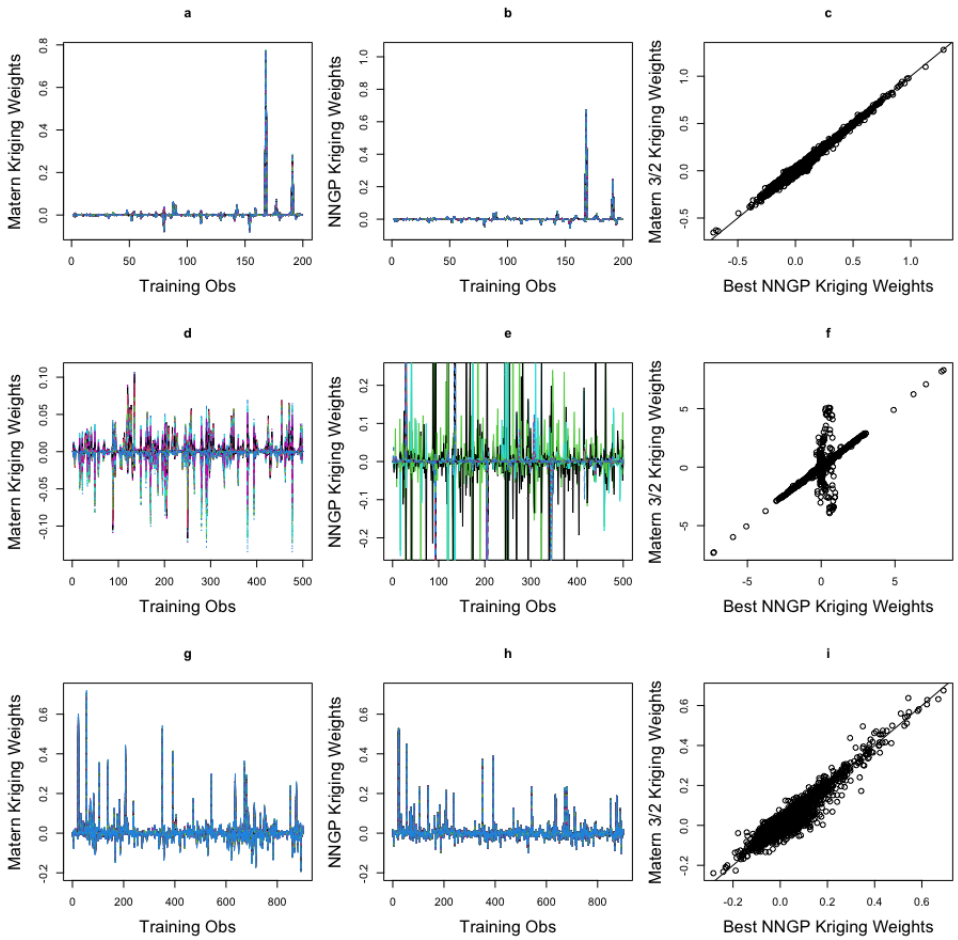

Figure 11. *Plots of kriging weights of the Matérn (a,d,g), NNGP (b, e, h), and a scatterplot comparison of the most performant NNGP kriging weights vs. the kriging weights of the Matérn* $\nu = \frac{3}{2}$ *(c, f, i). These results are for one iteration simulation for the Friedman function (a,b,c), MODIS satellite data (d,e,f), and borehole function (g,h,i).*

In conclusion, the NNGP kernel allows us to study predictions from an infinitely wide neural network with the weights and biases as i.i.d. $N(0, 1)$ variables. This has allowed us to understand that predictions from this theoretical neural network . Future research will need to extend our results to understand whether these conclusions generalize to more complex NN architectures. However, this kernel is more interesting in theory than in application. It has numerical challenges where it produces non-valid kernel matrices, particularly in deeper architectures. It under-performed in several benchmark datasets as compared to the more common Matérn kernel. Therefore, in application for its flexibility and performance, the classic Matérn kernel appears to remain the most practical kernel model for application of Gaussian process models to data of the types explored in this study.

Acknowledgments. This work was performed under the auspices of the U.S. Department

of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 with IM release number LLNL-TR-827489. Funding for this work was provided by LLNL Laboratory Directed Research and Development grant 19-SI-004. This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

REFERENCES

[1] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang, *On exact computation with an infinitely wide neural net*, arXiv preprint arXiv:1904.11955, (2019).

[2] D. A. Cole, R. B. Christianson, and R. B. Gramacy, *Locally induced gaussian processes for large-scale simulation experiments*, Statistics and Computing, 31 (2021), pp. 1–21.

[3] A. Daniely, R. Frostig, and Y. Singer, *Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity*, in Advances In Neural Information Processing Systems, 2016, pp. 2253–2261.

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

[5] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, *Deep convolutional networks as shallow gaussian processes*, arXiv preprint arXiv:1808.05587, (2018).

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

[7] M. J. Heaton, A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, et al., *A case study competition among methods for analyzing large spatial data*, Journal of Agricultural, Biological and Environmental Statistics, 24 (2019), pp. 398–425.

[8] A. Jacot, F. Gabriel, and C. Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, in Advances in neural information processing systems, 2018, pp. 8571–8580.

[9] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, *Neural network-based uncertainty quantification: A survey of methodologies and applications*, IEEE access, 6 (2018), pp. 36218–36234.

[10] M. Katzfuss, J. Guinness, and E. Lawrence, *Scaled vecchia approximation for fast computer-model emulation*, arXiv preprint arXiv:2005.00386, (2020).

[11] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, *Deep neural networks as gaussian processes*, in International Conference on Learning Representations, 2018.

[12] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington, *Wide neural networks of any depth evolve as linear models under gradient descent*, arXiv preprint arXiv:1902.06720, (2019).

[13] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, *Gaussian process behaviour in wide deep neural networks*, in Internation Conference on Learning Representation, 2018.

[14] A. Muyskens, B. Priest, I. Goumiri, and M. Schneider, *An identifiability and sensitivity analysis*

of matérn gaussian process hyperparameters on prediction, arXiv preprint arXiv:TBA, (2021).

[15] A. Muyskens, B. Priest, I. Goumiri, and M. Schneider, *MuyGPs: Scalable Gaussian Process Hyperparameter Estimation Using Local Cross-Validation*, arXiv preprint arXiv:2104.14581, (2021).

[16] A. L. Muyskens, I. R. Goumiri, B. W. Priest, M. D. Schneider, R. E. Armstrong, J. M. Bernstein, and R. Dana, *Star-galaxy image separation with computationally efficient gaussian process classification*, arXiv preprint arXiv:2105.01106, (2021).

[17] R. M. Neal, *Priors for infinite networks*, in Bayesian Learning for Neural Networks, Springer, 1996, pp. 29–53.

[18] M. L. Stein, *Interpolation of spatial data: some theory for kriging*, Springer Science & Business Media, 2012.

[19] B. A. Worley, *Deterministic uncertainty analysis*, tech. report, Oak Ridge National Lab., TN (USA), 1987.

[20] G. Yang, *Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are gaussian processes*, arXiv preprint arXiv:1910.12478, (2019).

[21] G. Yang and H. Salman, *A fine-grained spectral perspective on neural networks*, arXiv preprint arXiv:1907.10599, (2019).