Gaussian Process Surrogate Models for Neural Networks (Supplementary Material)

Michael Y. Li[1] **Erin Grant**[2] **Thomas L. Griffiths**[3]

[1]Department of Computer Science, Stanford University, Stanford, California, USA
[2]Gatsby Computational Neuroscience Unit, University College London, London, UK
[3]Departments of Psychology and Computer Science, Princeton, NJ, USA

A ADDITIONAL EXPERIMENTAL RESULTS

A.1 RANKING NN GENERALIZATION WITH THE GP MARGINAL LIKELIHOOD

In previous sections, we demonstrated that Gaussian process (GP) surrogate models could yield insight into neural network (NN) behavior. The benefits of GPs extend beyond this. Since the GP marginal likelihood has a closed form expression, many have advocated for using the marginal likelihood in model selection and as an indicator of expected generalization performance [Mackay, 1992]. In this section, we leverage the learned GP surrogate to *rank NNs by their generalization error with the GP marginal likelihood*. In particular, we learn GP surrogates from different NNs at random initialization, and we then study if the marginal likelihood of the surrogates can rank the NNs by test error after training. In the following experiments with varying classes of NN families, we find that we can indeed predict test error using the marginal likelihood of the training set under the learned surrogate GP.

A.1.1 The idealized case: Large-width NNs

Before we consider arbitrary NN families, we check that the marginal likelihood is predictive in an idealized setting. In particular, we consider large-width NNs whose infinite-width analogs are equivalent to GPs [Lee et al., 2017]. If the marginal likelihood is not predictive in this case in which the kernel function can be analytically determined, it is unlikely to be useful in a general setting where the kernel is learned and GPs approximate NNs priors but are not equivalent.

NN hyperparameters. We consider NNs with sine ($\sin$) or Gauss error function (erf)[1] activations and 2 hidden layers of 1024 units each. We randomly initialize the weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train an ensemble of 50 randomly initialized NNs from each family using full-batch (vanilla) gradient descent with learning rates of $\eta \in \{0.01, 0.1\}$.

Target function. The target function is $\sin(0.5x)$.

GP surrogate. We do not learn a kernel from NN predictions as in previous sections. Instead, we use the kernels corresponding to the infinite width analogs of the NNs using the neural-tangents package [Novak et al., 2020].

Results. Fig. (1) compares the performance of these NN families along with the marginal likelihood of the target function under the surrogate model. The performance (mean-squared error (MSE) on the test set) is averaged across each ensemble of NNs. The marginal log-likelihood (MLL) of the target function is higher for the better-performing NN family.

[1] Here, erf is defined as $a\ \mathrm{erf}(bx) + c$, where $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, \mathrm{d}t$.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

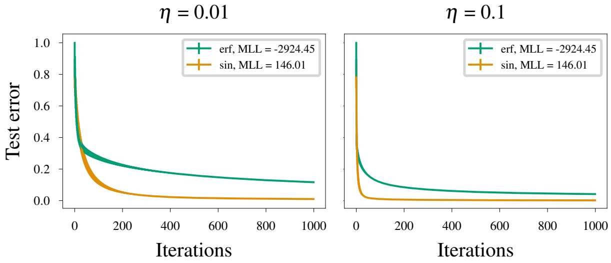

Figure 1: Ranking generalization from MLL in large-width NNs. Mean and standard error of the test MSE of large-width sinusoidal and erf NNs trained with learning rates $\eta = 0.01$ **(left)** and $\eta = 0.1$ **(right)** on the target function of Appendix (A.1.1). The MLL of the target function under the surrogate model corresponding to the limiting kernel for each model family is shown in the legend. Consistent with expectations, the model family whose surrogate assigns higher MLL to the target function achieves lower test error for both values of η.

A.1.2 Small width neural networks and learning the kernel

In the previous experiment, we showed that the marginal likelihood could be predictive when we consider large-width NNs and when we use a corresponding, analytically derived kernel. Is the marginal likelihood predictive when we consider smaller-width NNs and when we learn the kernel empirically?

NN hyperparameters. We consider ensembles of width 16, depth 4 NNs from two families: NNs with $\sin$ activations and NNs with rectified linear unit (ReLU) activations. We randomly initialize weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train an ensemble of 50 randomly initialized NNs from each family on the target functions using full-batch gradient descent with a learning rate of $\eta = 0.1$.

Target function. The target function families mirror the NN model families: We collect predictions from randomly initialized, width 16, depth 4 NNs with $\sin$ or ReLU activations. These target functions are a useful sanity check, as the inductive biases of the model families are perfectly suited for a target function family.

GP surrogate. For each ensemble, we learn the hyperparameters of an spectral mixture kernel (SMK) with $Q = 5$ mixture components by optimizing the marginal likelihood across the ensemble. To optimize, we randomly initialize the kernel hyperparameters and run Adam for 250 iterations with a learning rate of $\eta = 0.1$. We initialize the frequency parameters by sampling from a uniform distribution whose upper limit is the Nyquist frequency. We choose the kernel hyperparameters with the highest objective value across three random initializations.

Results. In Fig. (2), we compare the performances of the two NN families on the two target function families. We also display the kernels learned from NN behavior (*sin surrogate kernel* or *ReLU surrogate kernel*) and learned from the target function family (*data kernel*) directly. Across both experiments, the MLL averaged across the target function family of the better-performing NN family is higher. In general, the structure of a learned kernel reflects the properties of the learned GP prior, and so we can compare kernels to assess similarity between target function and NN families. We see that the data kernel provides a better qualitative match to the kernel of the better-performing model family.

A.1.3 Systematic study of various learning rates and architectures

In this last experiment on ranking generalization performance, we establish that Gaussian process surrogates reliably rank performance across a range of learning rates and gradient descent algorithms.

NN hyperparameters. We consider ensembles of randomly initialized NNs with $\sin$ or ReLU activations and 1 or 3 hidden layers with 256 hidden units in each layer. We randomly initialize the weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train 50 randomly initialized NNs from each family using either vanilla full-batch gradient descent with a constant learning rate of $\eta = 0.01$, or Adam [Kingma and Ba, 2015] using learning rates of $\eta \in \{0.0003, 0.003\}$.

Target function. We consider a target function of $\sin(0.5x)$.

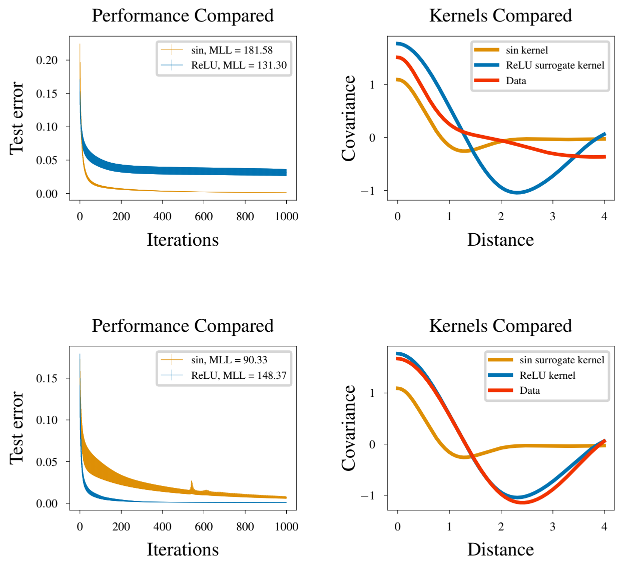

Figure 2: Ranking generalization from MLL in small-width NNs. Mean and standard error of test MSE **(left)** of small-width sinusoidal and rectifier NN ensembles on sin **(top)** and ReLU **(bottom)** target function families, with the target function MLL under the surrogate learned from each model family in the legend. Covariance **(right)** of surrogate kernels alongside data kernels learned from the sin **(top)** and ReLU **(bottom)** target function families. Even in the small-width regime and when the kernel is learned, the model family whose surrogate assigns a higher MLL to the target function attains lower error **(left)**; the surrogate kernel learned from the better-performing model family better matches the data kernel **(right)**.

GP surrogate. For each ensemble, we learn the hyperparameters of an SMK with $Q = 5$ mixture components by optimizing the marginal likelihood across the ensemble. To optimize, we randomly initialize the kernel hyperparameters and run adaptive momentum (Adam) for 250 iterations with a learning rate of $\eta = 0.1$. We choose the kernel hyperparameters with the highest objective value across three random initializations. To randomly initialize the frequency parameters, we uniformly sample from the real-valued interval $(0, 25]$.

Results. In Fig. (3), we find that the marginal likelihood of the better-performing NN family is higher. The marginal likelihood depends on the diagonal noise σ_n^2 added to the Gram matrix. We find that our result are robust across three levels of this diagonal noise $(10^{-3}, 10^{-4}, 10^{-5})$. These results suggest we can rank these NN families when they are not in the asymptotic regime and when we learn the kernel, in contrast to Appendix (A.1.1), as well as when *a priori* no model family should perform better, unlike Appendix (A.1.2).

A.2 CORRELATION SENSITIVITY

We present some additional results to supplement our analysis from Section 4.3.1 where we demonstrated that discrepancy in lengthscale profiles between data and neural network predicts the generalization gap. Correlation can be sensitive to outliers. Does any single dataset account for the negative correlations? To answer this, we characterize how the correlation changes as a result of dropping each dataset. Specifically, for each UCI dataset, we remove that dataset and then compute the correlation between lengthscale profile correlation and generalization gap for the remaining datasets. We plot the resulting distribution of correlations in Fig. (4). We find there is a tight spread around the correlation computed from all the UCI datasets. Importantly, when we remove any UCI dataset, we still see moderate to high negative correlations between lengthscale profile correlation and generalization gap.

A.3 PROPERTIES OF THE SPECTRAL MIXTURE KERNEL AND THE MATERN KERNEL

We describe how the various hyperparameters of the SMK and MK kernel affect the GP prior. We begin with the spectral mixture kernel. The mixture weights w are signal variances and control the scale of the function values. The mixture means (μ) encode periodic behavior. The variances (τ) are (inverse) lengthscales, which control the smoothness. The (ARD) Matérn kernel (MK) kernel has lengthscales θ, which controls the smoothness of the function with respect to each dimension. ν is

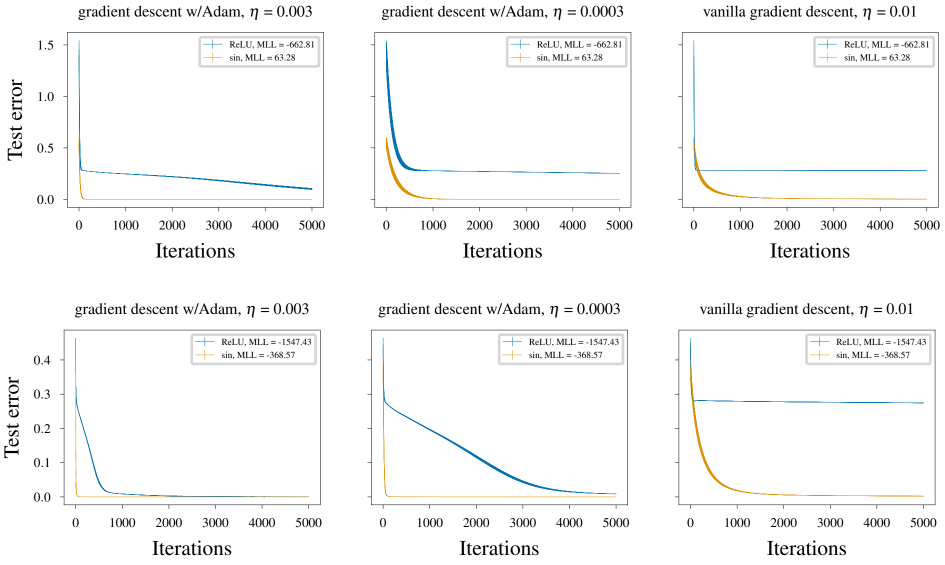

Figure 3: Ranking generalization performance from MLL across different learning algorithms and architectures. Each panel displays mean and standard error of test MSE of an NN family trained on the target function $\sin(0.5x)$ with noise; legend displays MLL of the training data under the surrogate for one of two NN families: 1-layer (256 hidden units) sinusoidal or rectifier NNs **(top)**); 3-layer (256 hidden units) sinusoidal or rectifier NNs **(bottom)**. NNs are trained with batch gradient descent with Adam (learning rates $\eta = 0.003$, $\eta = 0.0003$) or vanilla batch gradient descent ($\eta = 0.01$). Across architectures and learning algorithms, the NN family whose surrogate assigns higher MLL to the target function achieves lower test error.

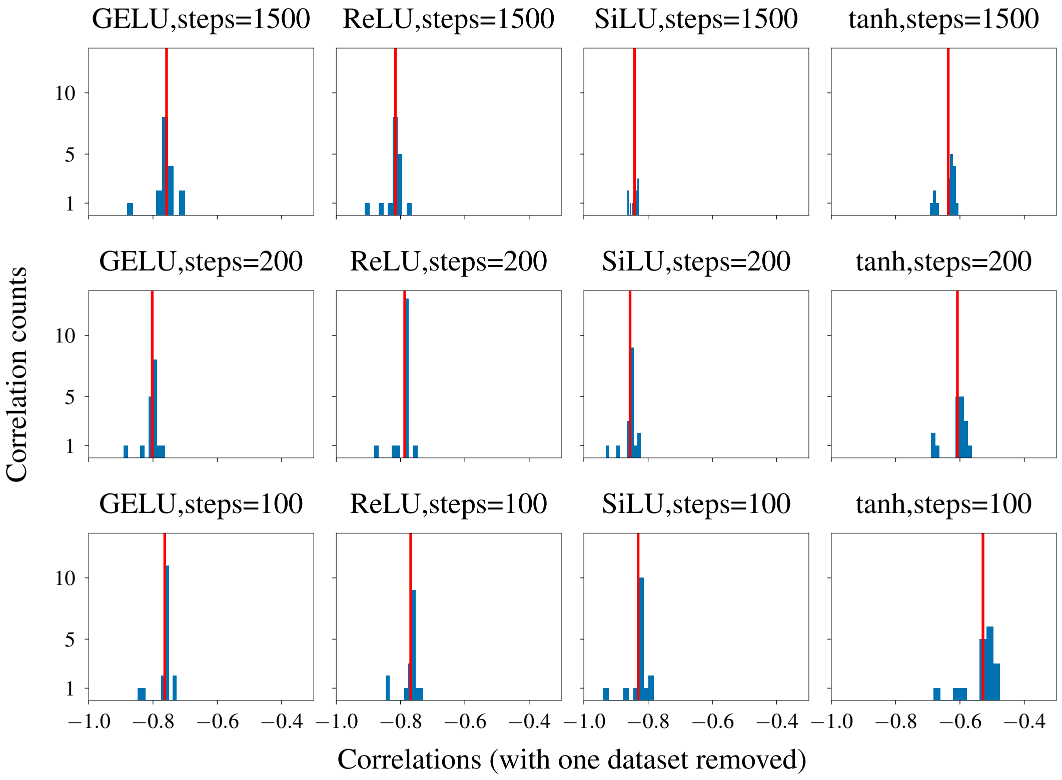

Figure 4: Sensitivity analysis of generalization gap and lengthscale profile relationship. Each panel a histogram and mean (red line) of correlations obtained by recomputing the correlation between lengthscale profile correlation and generalization gap after removing each UCI dataset. Across datasets and architectures, even when a single dataset is removed, there remains an negative correlation between generalization gap and lenthscale profile correlation. Therefore, the inverse relationship between generalization gap and lengthscale profile correlation demonstrated in Section 4.3.1 is robust to outlier datasets.

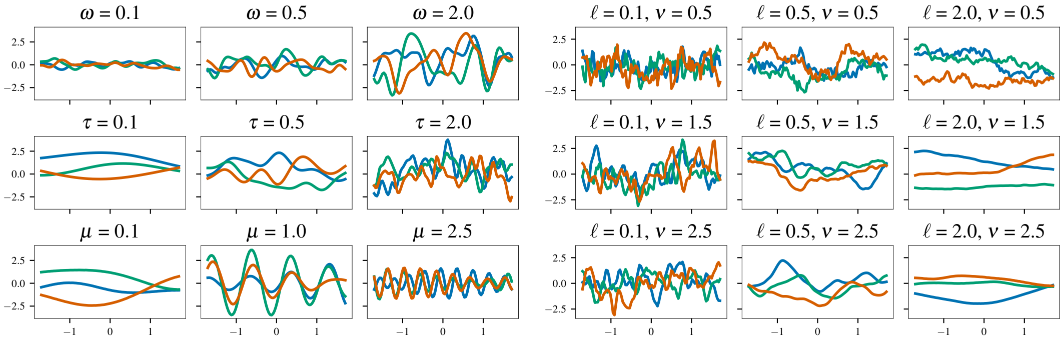

Figure 5: Illustrating the effect of GP kernel hyperparameters on the GP prior. (**Left**) Samples from a GP prior with SMK with varying mixture weights ω, mixture scale τ, and mixture means μ. (**Right**) Samples from a GP prior with Matern kernel with varying ν and ℓ (lengthscale). GPs are flexible models whose properties can be controlled through hyperparameters.

another hyperparameter that also modulates smoothness, and the Matern covariance function admits a simple expression when ν is a half-integer. $\nu = 2.5$ corresponds to twice differentiable functions and $\nu = 1.5$ corresponds to once differentiable functions.

In Fig. (5), we vary the hyperparameters of the SMK $(w, \mu, \tau)$and Matern kernels (ν, θ) and illustrate how they impact the prior over functions.

References

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *Proc. ICLR*, 2017.

David J. C. Mackay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in Python. In *Proc. ICLR*, 2020.