Implications of Gaussian process kernel mismatch for out-of-distribution data

Beau Coker [1] **Finale Doshi-Velez** [1]

Abstract

Gaussian processes provide reliable uncertainty estimates in nonlinear modeling, but a poor choice of the kernel can lead to poor generalization. Although learning the hyperparameters of the kernel typically leads to optimal generalization on in-distribution test data, we demonstrate issues with out-of-distribution test data. We then investigate three potential solutions– (1) learning the smoothness using a discrete cosine transform, (2) assuming fatter tails in function-space using a Student-t process, and (3) learning a more flexible kernel using deep kernel learning–and find some evidence in favor of the first two.

1. Introduction

Gaussian processes (GPs) are flexible distributions over functions that are widely used in applications (Williams & Rasmussen, 2006; Deringer et al., 2021; Liu et al., 2020a;b). The covariance of a GP leads to different function-space properties. For example, *amplitude variance* describes the magnitude of the functions, *lengthscale* describes the "flatness" of the functions, and *smoothness* corresponds to the number of times the functions are differentiable.

The covariance of a GP is given by the functional form of a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and its associated hyperparameters. For example, the popular *radial basis function* (RBF) kernel $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(\|\mathbf{x} - \mathbf{x}'\|^2/(2\ell^2))$ has two hyperparameters, σ^2 and ℓ, that control the amplitude variance and lengthscale properties, respectively. In fact, these hyperparameters can be added to any kernel by scaling the input to and output of the kernel (i.e., $\tilde{k}(\mathbf{x}, \mathbf{x}') := \sigma^2 k(\mathbf{x}/\ell, \mathbf{x}'/\ell)$) and, furthermore, can be learned from data, for example by full posterior inference or by optimizing the log marginal likelihood. Like many kernels, though, the smoothness property implied by an RBF kernel is fixed, in this case to functions so smooth they can be differentiated an infinite number of times. Although the Matérn kernel generalizes the RBF kernel by adding a smoothness hyperparameter ν, it is typically fixed to one of $\nu \in \{1/2, 3/2, 5/2\}$ because it is computationally difficult to infer from data (Marin et al., 2021). Unfortunately, if the functions implied by the kernel are too smooth relative to the ground-truth function that generated the data, the generalization error on test data can decay slowly in the number of training observations, even logarithmically slowly in the case of an RBF kernel (Sollich, 2001; van der Vaart & van Zanten, 2011; Jin et al., 2022).

Fortunately, a small lengthscale can effectively compensate for such a smoothness "mismatch" (Sollich & Ashton, 2012; van der Vaart & van Zanten, 2011) by allowing the function to "wiggle" as a substitute for being more rough. However, in this work we demonstrate it can come at the cost of poor performance in an out-of-distribution (OOD) "gap" in the data, because the function misses out on long-range trends captured by the lengthscale. We say it *can* come at cost because the performance depends on the way the model is evaluated and whether all of the hyperparameters are optimized. Even in the best case, though, the OOD performance seems unsatisfactory because the posterior quickly reverts to the prior. We provide a careful examination of this phenomena using a simple 1D dataset.

Figure 1 shows an example of the problem we study. In the left panel, the model is smoother than the process that generated the data. To compensate, the model learns a small lengthscale, resulting in a decent fit of the training data (see its posterior in purple). However, in the OOD gap between the two clusters of training data, the posterior quickly reverts to the prior because of the small lengthscale. This is especially evident in the posterior mean (thick line) reverting downwards to the prior mean of zero. In contrast, the correctly specified model (i.e., that generated the data, shown in blue) identifies the upward trend in the data. While also inferring the amplitude variance and, importantly, a constant mean function (right panel, shown in red) significantly improves the uncertainty quantification in the gap, the posterior mean still poorly models the upward trend because of the small lengthscale.

To address these issues, we investigate three possible solutions: learning the smoothness by manipulating the function in frequency space, using a heavier-tailed function-space

[*]Equal contribution [1]Harvard University. Correspondence to: Beau Coker <beaucoker@g.harvard.edu>.

Accepted to ICML workshop on Structured Probabilistic Inference Generative Modeling, Honolulu, Hawaii, USA. PMLR 202, 2023.
Copyright 2023 by the author(s).

distribution with the Student-t process (Shah et al., 2014), and allowing for more flexible kernel learning using deep kernel learning (Wilson et al., 2016). The first method scales poorly but shows some promise.

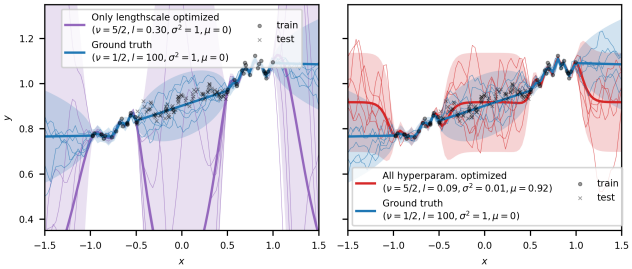


Figure 1: **Optimizing kernel hyperparameters of a mismatched kernel can lead to poor OOD generalization**. *Left:* A small lengthscale allows an overly smooth model (purple) to fit rough training data, but results in a quick reversion to the prior in the out-of-distribution "gap". Notice how the posterior mean (thick line) dips drastically below the test data, towards the prior mean of zero, compared to the correctly specified model (blue). *Right:* Also optimizing the variance hyperparameter and a constant mean function results in much better performance, but the posterior mean still misses the upward trend in the data (red). Shaded regions are ± 2 standard deviations of the posterior.

2. Background

GPs The mean function $m : \mathcal{X} \to \mathbb{R}$ and the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ define a Gaussian process. Together with a likelihood, which in this paper we take to be a Gaussian distribution of constant variance, they constitute a statistical model of noisy function observations. The mean function is often taken to be zero, so the kernel and its hyperparameters encodes most of the assumptions made by the user.

GP learning rates There is a large literature on the asymptotic, ID generalization properties of kernel ridge or GP regression, but a common theme is that when the functions implied by the kernel are smoother than the ground-truth function, then the relative smoothness of the two plays a key role (Sollich, 2001; van der Vaart & van Zanten, 2011; Jin et al., 2022). Thus, it is this setting where learning a lengthscale is especially important to ID generalization. We use it to study the implications for OOD generalization.

GP smoothness The smoothness of a GP can be measured by the eigenspectrum of the kernel integral operator, with fatter tails implying rougher processes. The eigenspectrum is known only in some cases (Zhu et al., 1998; Hawkins, 1989; Bach & Jordan, 2002; Velikanov & Yarotsky, 2021), but it can be estimated empirically (see Appendix B). Unfortunately, the smoothness is often fixed by the kernel.

3. In and Out of Distribution Behavior of Mismatched GPs

We demonstrate empirically how a small lengthscale allows an overly smooth model to fit training data and nearby, "in-distribution" (ID) test data, but has unintended consequences for further away, OOD test data. Intuitively, this happens because a small lengthscale results in a quick reversion to the prior, thus ignoring long-range trends in the data.

We generate data by sampling functions from a rough, non-differentiable process — a Matérn GP with $\nu = 1/2$. We allow for a long-range trend by setting the lengthscale to $\ell = 100$. For the model, we always use a kernel that implies a smoother, "mismatched" GP: Matérn ($\nu \in \{3/2, 5/2, \infty\}$, arccos (order $\in \{1, 2\}$), and piecewise polynomial (degree $\in 1, 2$)). We start by assuming the train and test sets come from the same distribution (i.e., the test distribution is ID), writing $\mathcal{D}^{\text{train}} = \{\mathbf{x}_n^{\text{train}}, \mathbf{y}_n^{\text{train}}\}_{n=1}^{N^{\text{train}}}$ and $\mathcal{D}^{\text{ID}} = \{\mathbf{x}_n^{\text{ID}}, \mathbf{y}_n^{\text{ID}}\}_{n=1}^{N^{\text{ID}}}$, where $\mathbf{x}_n^{\text{train}}, \mathbf{x}_n^{\text{ID}} \sim p_{\text{ID}}(\mathbf{x})$. Later, we use an OOD test set $\mathcal{D}^{\text{OOD}} = \{\mathbf{x}_n^{\text{OOD}}, \mathbf{y}_n^{\text{OOD}}\}_{n=1}^{N^{\text{OOD}}}$, where $\mathbf{x}_n^{\text{OOD}} \sim p_{\text{OOD}}(\mathbf{x})$. We use a noise variance of 0.1.

GPs are typically evaluated — both for training hyperparameters and measuring posterior performance — by the *negative log marginal likelihood* (NLML), which measures the (negative log of the) average likelihood that functions drawn from the GP place on the entire dataset. For some models (e.g., Bayesian neural networks), the NLML is intractable, so it is common to instead report the (negative log of the) average likelihood that functions drawn from the model place *on each point individually*. We call this the DiagNLML because it ignores the covariance between the function evaluated at different inputs. Although the NLML can be computed exactly for GPs, we will demonstrate that it prefers drastically different kernel hyperparameters than the DiagNLML when evaluated on OOD data and, arguably, the hyperparameters preferred by the DiagNLML are more desirable based on a visual examination of the posterior.

In our notation, since we distinguish between the dataset used for conditioning, $\mathcal{D}^c \in \{\emptyset, \mathcal{D}^{\text{train}}\}$ (yielding the prior or posterior, respectively), and the dataset used for prediction, $\mathcal{D}^p \in \{\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{ID}}, \mathcal{D}^{\text{OOD}}\}$, we write the two metrics as

$$\text{NLML}_{\mathcal{D}^c}^{\mathcal{D}^p} = \frac{1}{N^p} \log \int p(\mathbf{y}^p \mid \mathbf{f}^p) p(\mathbf{f}^p \mid \mathcal{D}^c)\, d\mathbf{f}^p$$

$$\text{DiagNLML}_{\mathcal{D}^c}^{\mathcal{D}^p} = \frac{1}{N^p} \sum_{n=1}^{N} \log \int p(y_n^p \mid f_n^p) p(f_n^p \mid \mathcal{D}^c)\, df_n^p$$

where $\mathbf{f}^p = \{f_n^p\}_{n=1}^{N^p} = \{f(\mathbf{x}_n^p)\}_{n=1}^{N^p}$ is the function evaluated on the inputs. For simplicity, we abuse notation and write, e.g, $\text{NLML}_{\text{prior}}^{\text{train}}$ instead of $\text{NLML}_{\emptyset}^{\mathcal{D}^{\text{train}}}$. We also use the same superscript and subscript notation on the root-mean-squared-error (RMSE) of the posterior mean.

To choose the hyperparameters, we follow the standard GP training procedure of minimizing the NLML of the prior on the training data, i.e., $\text{NLML}^{\text{train}}_{\text{prior}}$. We always optimize the lengthscale with a gridsearch. Using the learned hyperparameters, we compute the posterior in closed-form.

A small lengthscale minimizes the training objective ($\text{NLML}^{\text{train}}_{\text{prior}}$) and effectively compensates for the smoothness mismatch on ID test data. Figure 2 illustrates how the standard training procedure learns a relatively small lengthscale, $\ell = 0.29$, which permits the overly smooth model to fit the rough training data much better than the model with the larger lengthscale of $\ell = 1.0$. This is consistent with previous work (Sollich & Ashton, 2012). Section A.1 shows this conclusion is robust across other random datasets, different model kernels, and whether the other hyperparameters are optimized.

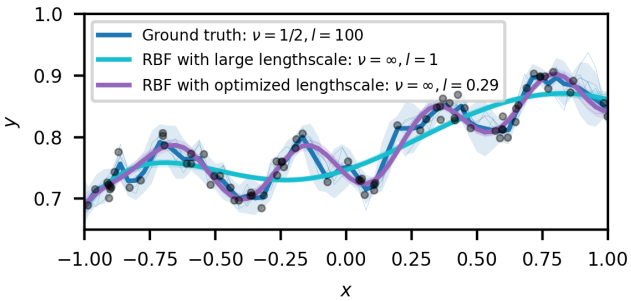


Figure 2: **A small lengthscale allows a smoother model to fit the training data and any nearby test data**. Posterior of RBF GPs compared to the rougher, ground-truth Matérn GP with $\nu = 1/2$.

Next we turn to the main question of this paper: *does a lengthscale smaller than the ground-truth negatively impact OOD behavior?* We find that the answer depends on how the model is evaluated (NLML vs. DiagNLML) and whether other hyperparameters (i.e., beyond lengthscale) are also optimized (including a constant mean function, which is otherwise set to zero). We again draw noisy data from a GP with a Matérn kernel and a large lengthscale, $\ell = 100$, but we instead assume there is a gap in the training data, with the OOD data in the gap. An example of this dataset, and the basic intuition of our results, is shown in Figure 1.

When only the lengthscale is inferred, an overly smooth model provides poor OOD predictions and marginal uncertainty (i.e., RMSE and DiagLML), though surprisingly good full covariance uncertainty (i.e., NLML) Figure 3 shows the intuition for a single dataset. The center panel shows various metrics as a function of the lengthscale, which is the only hyperparameter we optimize in this experiment. Because of the smoothness mismatch, the standard training objective, $\text{NLML}^{\text{train}}_{\text{prior}}$ shown in purple, is minimized by a fairly small lengthscale. Unfortunately, a small lengthscale leads to poor OOD predictions (i.e., $\text{RMSE}^{\text{OOD}}_{\text{post}}$, shown in brown) and poor OOD marginal uncertainty (i.e., $\text{DiagNLML}^{\text{OOD}}_{\text{post}}$, shown in orange). This is the poor OOD behavior we highlight in this work. Interestingly, though, a small lengthcale does not result in a poor OOD *full covariance* uncertainty (i.e., $\text{NLML}^{\text{OOD}}_{\text{post}}$, shown in green). This happens because the NLML rewards the model for compensating for the smoothness mismatch with a small lengthscale, enough so to outway the poor predictions.

The posteriors in Figure 3 illustrate the dramatic difference in the hyperparameters preferred by the posterior OOD NLML and DiagNLML. On the left is the model that minimizes the posterior OOD DiagNLML. Notice the posterior mean strongly reverts to the prior mean of 0 in the gap and sharply differs from the posterior of the model that generated the data (shown in blue). This is in contrast to the model that minimizes the posterior OOD NLML, shown on the right. It is difficult to imagine an application where the green model on the left would be preferred to the orange model on the right, but nonetheless this is the preference of the standard evaluation metric, the NLML.

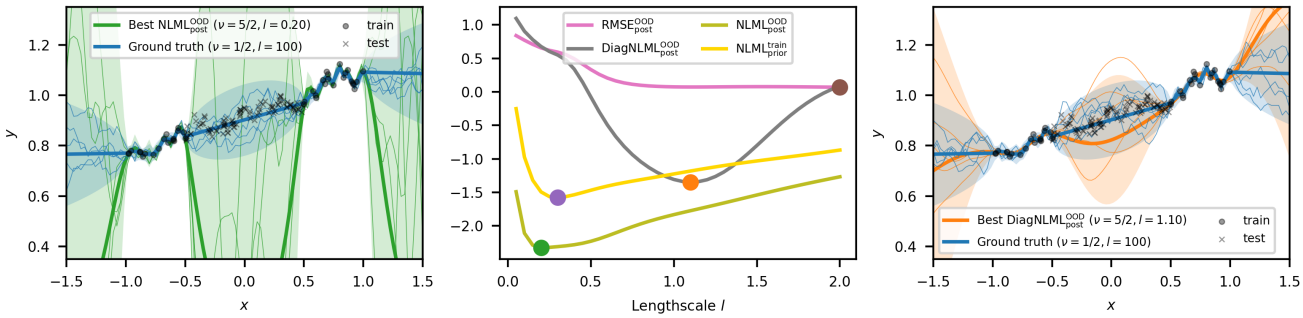


Figure 3: **A small lengthscale nearly minimizes the full covariance NLML on OOD data (green), but has poor predictions and marginal variance (brown and orange)**. *Left:* Posterior of the model with the best $\text{NLML}^{\text{OOD}}_{\text{post}}$. *Middle:* Various prior and posterior metrics as a function of lengthscale. The $\text{NLML}^{\text{train}}_{\text{prior}}$ is minimized during inference and the rest are used for evaluation on OOD data. *Right:* Posterior of the model with the best $\text{DiagNLML}^{\text{OOD}}_{\text{post}}$.

This behavior is illustrated across datasets and kernels by Figure 4, which compares the average performance across several metrics based on two criteria for selecting the lengthscale: minimizing the NLML of the prior on the training data (i.e., the standard training objective) and the minimizing the DiagNLML of the posterior on the OOD test data. Of course, the latter criteria cannot be minimized in practice because it requires access to the test data, but we provide it to show the best possible choice of the lengthscale for this metric. In other words, we are comparing what happens in standard practice based on the training data to what we would ideally like to optimize if we somehow had access to the OOD test data. Any new method designed to alter the lengthscale chosen by the standard objective could not result in performance better than what we show. We use the DiagNLML instead of the NLML because we have already seen how the NLML prefers models with very poor predictions in the situation we analyze.

Immediately we see the expected tradeoff in the lengthscale. As in the single dataset in Figure 3, models that minimize the training objective tend to have a smaller lengthscale and do poorly in OOD prediction ($\text{RMSE}_{\text{post}}^{\text{OOD}}$) and OOD marginal uncertainty calibration ($\text{DiagNLML}_{\text{post}}^{\text{OOD}}$). But models that minimize the OOD marginal uncertainty do relatively worse on the other metrics, i.e., ID metrics and OOD NLML. Therefore, if one were to force the lengthscale to be larger to do better on OOD data (in terms of RMSE or DiagNLML), the model would do worse on ID data or OOD NLML. Note for comparision, for every metric we also show the best possible choice of the lengthscale for that metric as a black bar (i.e., "best in category"), which by construction agrees with the purple bar in the case of $\text{DiagNLML}_{\text{post}}^{\text{OOD}}$.

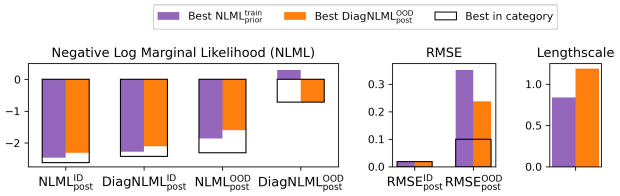


Figure 4: **Across datasets and kernels, the best lengthscale for posterior DiagNLML on OOD data ($\text{DiagNLML}_{\text{post}}^{\text{OOD}}$) is relatively large while the best lengthscale for other metrics is relatively small.** For each kernel and dataset, we select the lengthscale that minimizes the corresponding metric.

When all hyperparameters are inferred, the performance improves but the posterior still quickly reverts to the prior. Figure 5 demonstrates that inferring all of the hyperparameters (including a constant mean function) tends to improve the OOD DiagNLML and RMSE, in particular when using the standard procedure of selecting the hyperparameters. In the case of Matérn kernel with $\nu = 5/2$, the

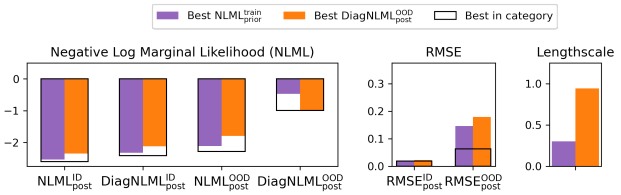


Figure 5: **Optimizing all hyperparameters improves DiagNLML and RMSE of the posterior on OOD data, but the lengthscale is even smaller.**

dramatic difference in the posterior can be seen by comparing the left and right panels of Figure 1. However, we argue that the solution provided by optimizing all of the hyperparameters, while clearly preferable, is unsatisfactory. The posterior still tends to revert to the prior in the gap because of the small lengthscale, it is just that the prior is much better calibrated to the data. Any long-range patterns in the data cannot be identified, though.

There are two modes when minimizing the posterior OOD RMSE. Figure 6 demonstrates how models with small and large lengthscales can perform similarly but for different reasons. If the lengthscale is small, the posterior does not make an attempt to capture any trends in the data and just reverts to the prior. This solution will never perform very well, but it will also never perform very badly. If the lengthscale is large, the posterior does make an attempt but it will typically be incorrect to some degree. Generally we argue the large lengthscale solution is preferable since at least it *could* identify the trend, but in this case reverting to the prior mean of zero indeed gave better predictions.

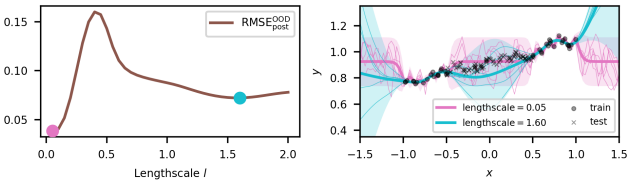


Figure 6: **A small or large lengthscale can minimize the RMSE**. The larger lengthscale better identifies the upward trend, but has poor uncertainty quantification. The smaller lengthscale quickly reverts to the prior.

4. Beyond Standard GPs

So far we have seen a trade-off in choosing the lengthscale when the model is overly smooth. If the lengthscale is too large, the model cannot fit ID. If the lengthscale is too small, the model quickly reverts to the prior outside of the data. Next we investigate three extensions of standard GPs, hoping to perform well on ID and OOD data simultaneously.

Manipulate the smoothness in frequency space. Earlier we stated that the smoothness of a GP cannot be optimized with

standard methods, so the lengthscale attempts to compensate for any smoothness mismatch. But what if the smoothness could be optimized? We propose a method (DCTGP), inspired by the low-pass filtered BNN (Yao et al., 2022) and spectral mixture kernel (Wilson & Prescott, 2013), that manipulates the smoothness of a GP using a *discrete cosine transform* (DCT). Specifically we manipulate function samples evaluated on a grid of evenly-spaced inputs. We then (1) transform to frequency space using a DCT, (2) manipulate the tail of the frequency spectra, for which there is a trainable hyperparameter, and finally (3) transform back to function space using an inverse DCT. These three steps correspond to a single linear transformation $\mathbf{A}$, so this method is still a GP but with Gram matrix $\mathbf{AKA}^\top$ instead of $\mathbf{K}$. See Appendix C for details. Unfortunately, evaluating functions on a grid of points scales poorly, so we deem this a toy method that could perhaps inspire scalable alternatives.

Use heavy tails in function space. A heavy-tailed distribution over functions could allow the posterior to adapt to a smoothness mismatch without adjusting the lengthscale as significantly. We use a Student-t process (STP), which replaces the multivariate Gaussian distribution in a GP with a multivariate Student's t-distribution (Shah et al., 2014). Note that STPs have the same posterior mean as GPs (given the same hyperparameters), but differ from GPs in two important ways. First, the posterior variance of the STP is outcome dependent (i.e., it depends on $\{y_n^{\text{train}}\}_{n=1}^{N_{\text{train}}}$) and, second, the marginal likelihood is different, so the optimized hyperparameters could be different. Bayesian neural networks and deep GPs also imply a heavy-tailed function-space distribution, but we leave them to future work to avoid introducing approximate inference as a confounding factor.

Use deep kernels. By passing the inputs through a neural network before applying a standard kernel, like RBF, *deep kernel learning* (DKL) aims to create more expressive kernels (Wilson et al., 2016). Perhaps it will identify a kernel that models the rough training data without losing the long-range trends captured by the lenghthscale.

DCTGP provides substantial improvements, STP provides some improvements, and DKL performs worse. Figure 7 shows the ID and OOD performance, analogously to Figures 4 and 5 (the purple bar is the same). The DCTGP performs the best, significantly improving the OOD DiagNLML and RMSE. Notice the lengthscale is typically much larger, which makes sense given that the smoothness can adapt. Although STP provides some improvement, we find that similar improvements could be made simply by plugging the posterior kernel of the standard GP into a t-distribution. In other words, for the same kernel, a t-distribution places more mass on the data than a Gaussian distribution, likely due to the kernel mismatch, but there was little to no value in training an STP instead of a GP. DKL performs very poorly on OOD uncertainty quantification (the values are outside of the plot), possibly due to overfitting of the training data, which is a known problem with DKL (Ober et al., 2021). We also show in Appendix B that if the weights of the neural network are drawn randomly, there is little impact on the eigenspectrum of the kernel. This suggests the neural network may not be able to significantly impact the smoothness implied by the

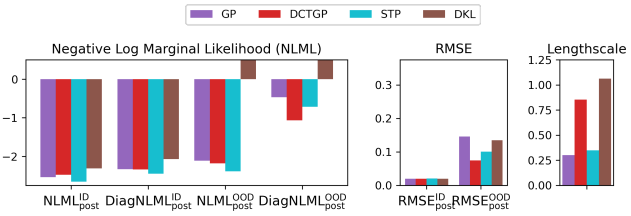


Figure 7: **DCTGP and STP improve the DiagNLML$^{\text{OOD}}_{\text{post}}$ over the standard GP**.

5. Conclusion

GPs are often used as the benchmark for neural-network-based uncertainty quantification methods, like Bayesian neural networks or deep ensembles (Lakshminarayanan et al., 2017), but it is important to remember that GPs are less flexible. Although they are often consistent in function space, they are not necessarily consistent in kernel space. They have a fixed kernel with typically only a few hyperparameters and, even if this kernel is correctly specified, they may not be able to recover the ground-truth hyperparameters (Zhang, 2004). We have shown that the hyperparameters can effectively compensate for kernel mismatch near the training data, but away from the data — where uncertainty quantification is most valuable — GPs can behave poorly due to one hyperparameter compensating for another. Although fatter tails in function-space provided some improvement, only learning the mismatched smoothness with the DCTGP significantly boosted performance. Since the DCTGP does not scale well as proposed, in the future we will explore neural-network-based models and deep GPs as we hypothesize depth may aid in adapting the smoothness separately from the lengthscale. This will enable the ultimate goal of generalizing well to ID and OOD test data when the kernel is unknown and thus likely misspecified.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1750358. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Bach, F. R. and Jordan, M. I. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., and Csányi, G. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16): 10073–10141, 2021.

Hawkins, D. Some practical problems in implementing a certain sieve estimator of the gaussian mean function. *Communications in Statistics—Simulation and Computation*, 18:481–500, 1989.

Jin, H., Banerjee, P. K., and Montúfar, G. Learning curves for gaussian process regression with power-law priors and targets. In *International Conference on Learning Representations (ICLR)*, 2022.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Liu, B., Kiskin, I., and Roberts, S. An overview of gaussian process regression for volatility forecasting. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 681–686. IEEE, 2020a.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31 (11):4405–4423, 2020b.

Marin, O., Geoga, C., and Schanen, M. On automatic differentiation for the matérn covariance. In *35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. The promises and pitfalls of deep kernel learning. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1206–1216. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/ober21a.html.

Shah, A., Wilson, A. G., and Ghahramani, Z. Student-t processes as alternatives to gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

Sollich, P. Gaussian process regression with mismatched models. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

Sollich, P. and Ashton, S. Learning curves for multi-task gaussian process regression. *Advances in Neural Information Processing Systems*, 25, 2012.

van der Vaart, A. and van Zanten, H. Information rates of nonparametric gaussian process methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

Velikanov, M. and Yarotsky, D. Universal scaling laws in the gradient descent training of neural networks. *arXiv:2105.00507 [cs.LG]*, 2021.

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

Wilson, A. G. and Prescott, R. A. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning (ICML)*, 2013.

Wilson, A. G., Zhiting, H., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Yao, J., Yacoby, Y., Coker, B., Pan, W., and Doshi-Velez, F. An empirical analysis of the advantages of finite- v.s. infinite-width bayesian neural networks. In *Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems (ICML)*, 2022.

Zhang, H. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99, 2004.

Zhu, H., Williams, C. K. I., Rohwer, R., Morciniec, M., and Hammel, M. *Gaussian Regression and Optimal Finite Dimensional Linear Models.* Springer-Verlag, Berlin, 1998.

A. Additional experiments

A.1. In-Distribution (ID) Behavior

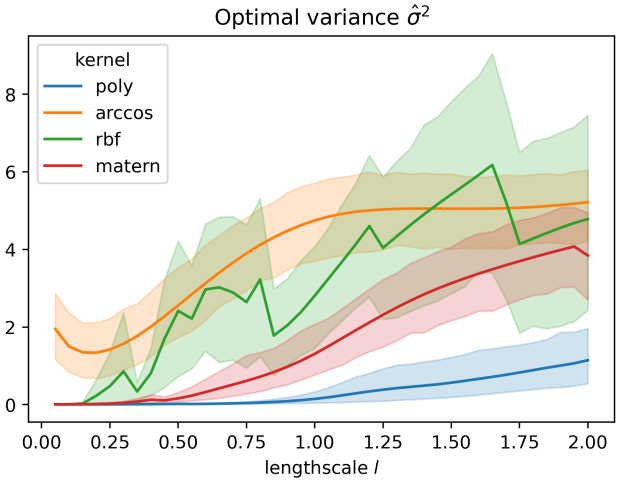
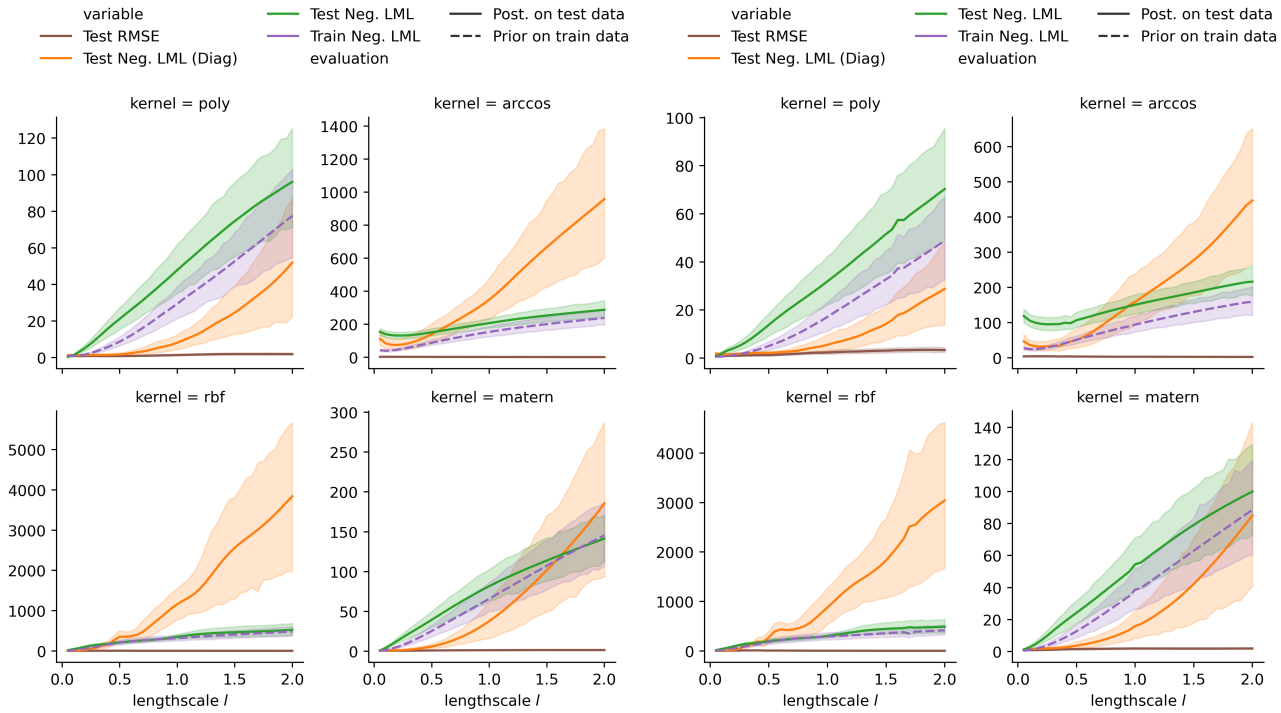


Figure 8: The optimal amplitude variance decreases with the lengthscale.



(a) Non-lengthscale hyperparameters fixed to ground-true values.

(b) All hyperparameters optimized.

Figure 9: $\text{NLML}_{\text{ID}}^{\text{post}}$ performance as a function of lengthscale, using the experimental setup as in Figure 2.

A.2. Out-of-Distribution (OOD) Behavior

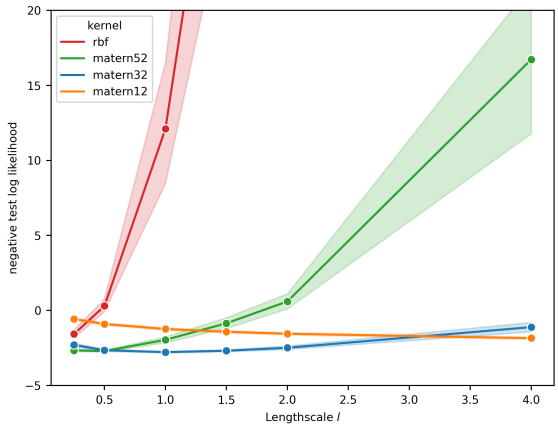
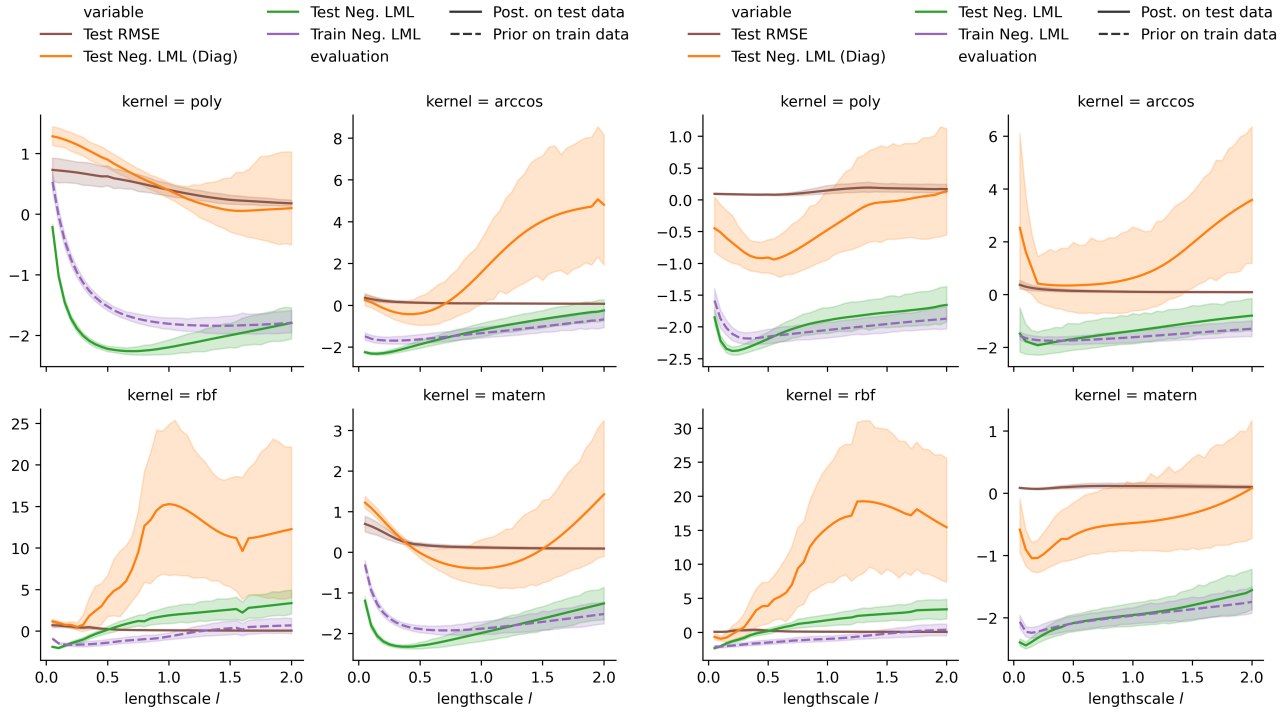


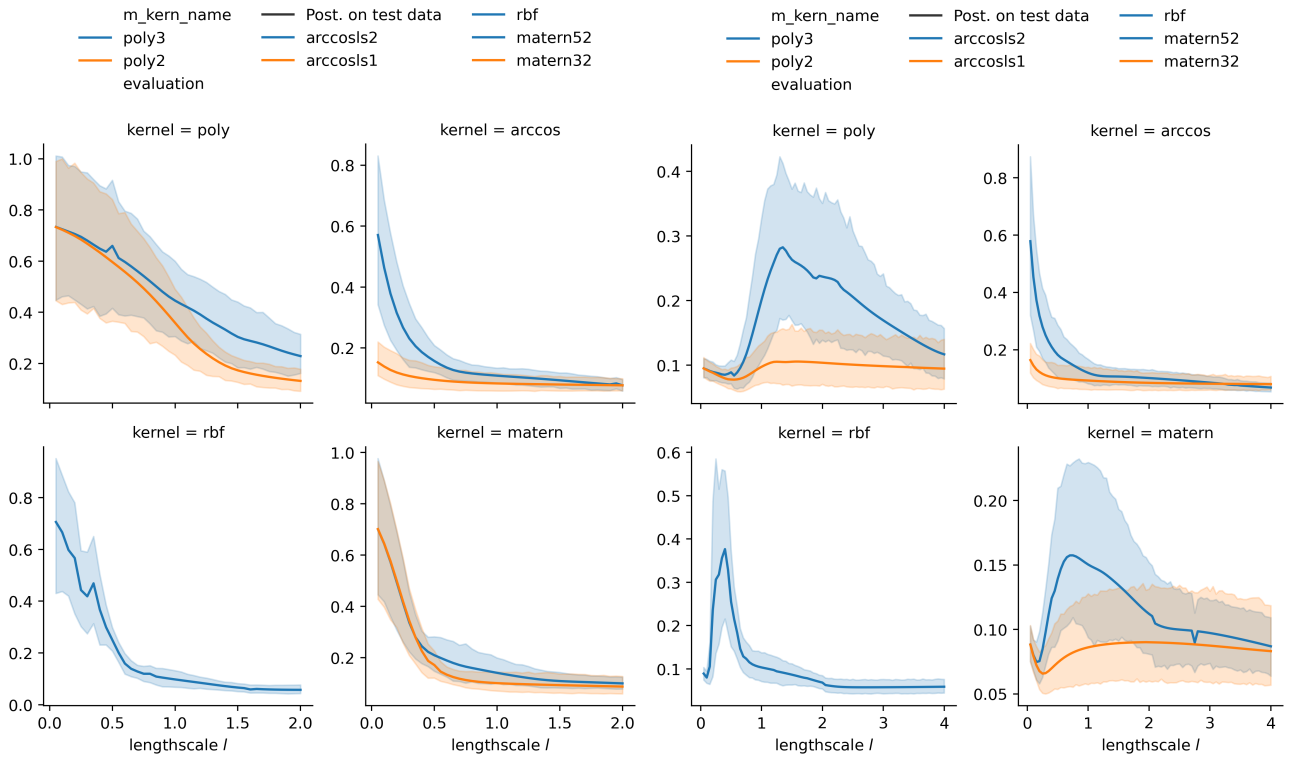
Figure 10: $\mathrm{NLML}_{\mathrm{ID}}^{\mathrm{post}}$ as a function of lengthscale for various Matérn kernels. The ground truth model is a Matérn GP with $\ell = \sigma^2 = 1$ and $\nu = 3/2$. The optimal value of the lengthscale can be different from the ground truth if the smoothness is misspecified.



(a) Non-lengthscale hyperparameters fixed to ground-true values.

(b) All hyperparameters optimized.

Figure 11: OOD NLML performance as a function of lengthscale, using the experimental setup as in Figure 3 but averaged over 10 function draws.



(a) Non-lengthscale hyperparameters fixed to ground-true values.

(b) All hyperparameters optimized.

Figure 12: $\text{RMSE}_{\text{OOD}}^{\text{post}}$ performance as a function of lengthscale, using the experimental setup as in Figure 3 but averaged over 10 function draws. We also break down results by kernel class, with the blue line representing a kernel that yields a more smooth process as compared to the kernel represented by the orange line.

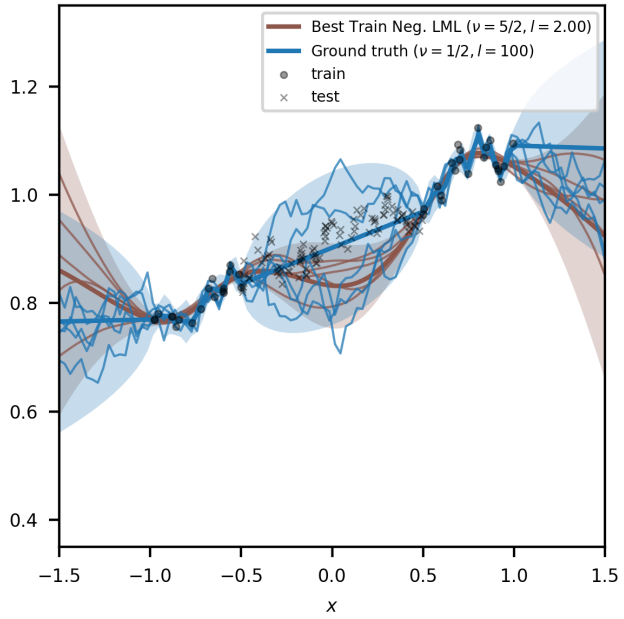


Figure 13: Best test RMSE model (among those tested) in Figure 3.

B. Smoothness of non-traditional kernels

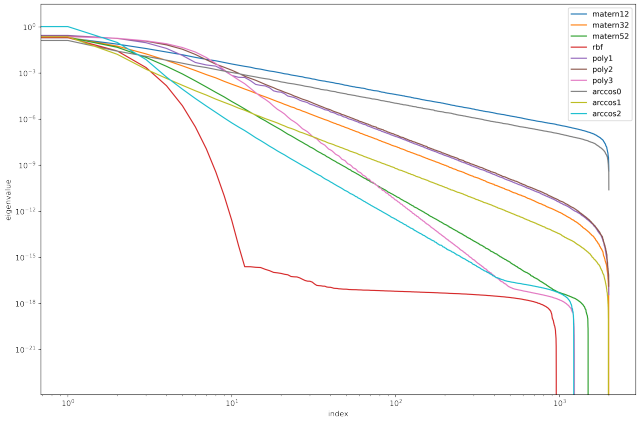


Figure 14: Estimated eigenvalues of kernels used in experiments. Matern12 (i.e., Matérn with $\nu = 1/2$) is used for data generation because it is the smoothest (slowest eigenvalue decay).

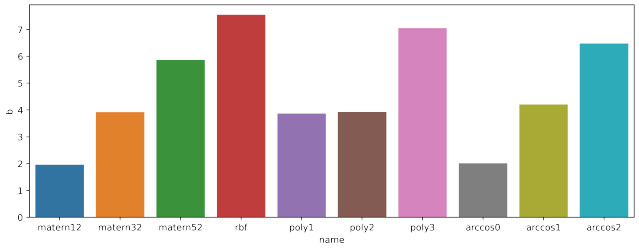


Figure 15: Estimated power-law decay rates of the eigenvalues of kernels used in experiments.

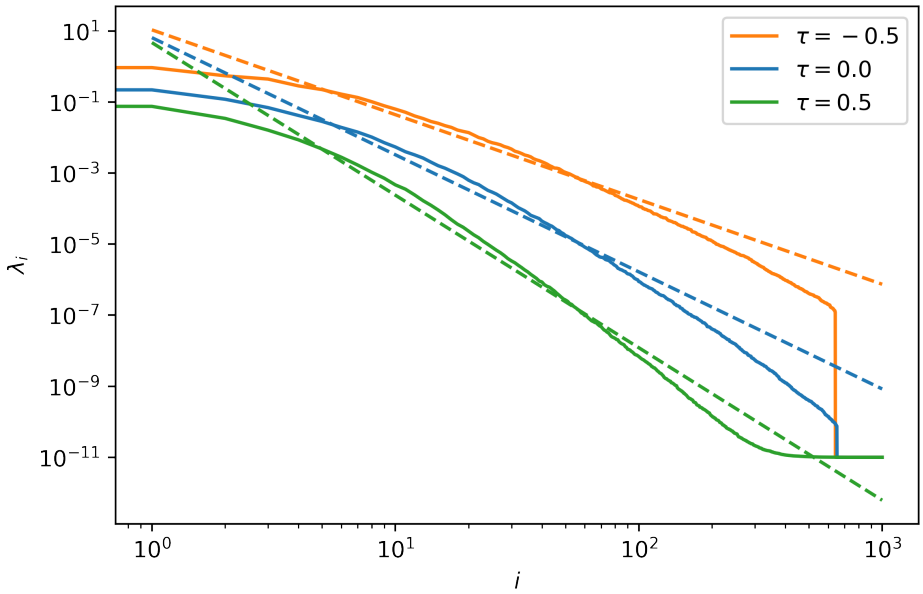


Figure 16: Eigenspectrum for a DCTGP with a Matérn ($\nu = 3/2$) kernel with different τ parameters. Larger values of τ correspond to faster eigenvalue decays and thus smoother functions.

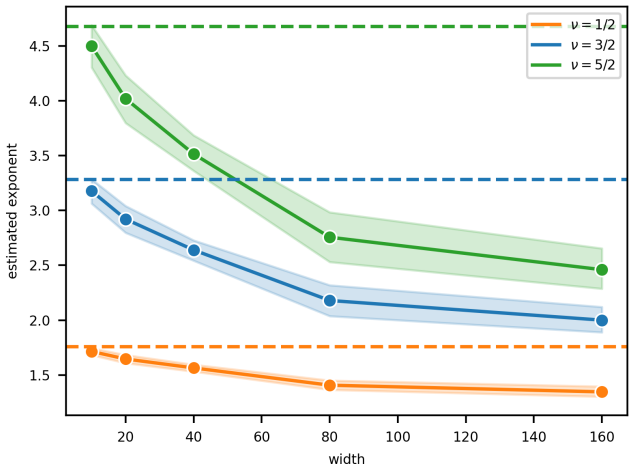


Figure 17: Impact of width on eigenspectrum of deep Matérn kernel

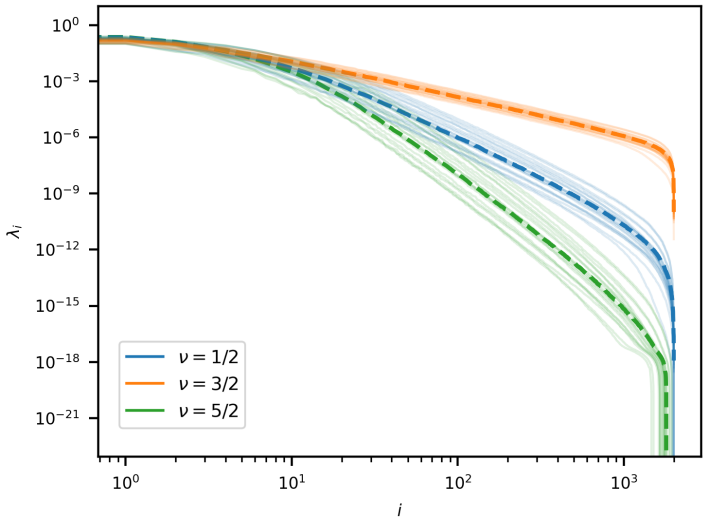


Figure 18: Prior distribution over eigenspectrums from deep Matérn kernels. The dotted line is the eigenspectrum of the regular Matérn kernel

C. DCTGP Details

We propose a toy method (DCTGP) that manipulates the smoothness of the model using a *discrete cosine transform* (DCT), which is a linear transformation $\mathbf{T}$ that decomposes any discrete signal into a weighted sum of cosines of different frequencies. By taking the signal to be the function evaluated on a grid of N^{grid} inputs, the smoothness of the function can be altered by adjusting the weight on the smallest frequency cosines and then applying the inverse transform, $\mathbf{T}^\top$. Specifically we multiply the frequencies by $\mathbf{J} := \text{diag}(\mathbf{j}^{-\tau})$, where $\mathbf{j} := (1, \ldots, N^{\text{grid}})$ indexes the frequences and τ is a trainable parameter that adjusts the smoothness of the function. Since the whole transformation $\mathbf{A} := \mathbf{T}^\top \mathbf{J} \mathbf{T}$ is linear, this still defines a Gaussian process with Gram matrix $\mathbf{K} = \mathbf{A}\mathbf{K}\mathbf{A}^\top$.