The Limitations of Large Width in Neural Networks: A Deep Gaussian Process Perspective

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large width limits have been a recent focus of deep learning research: modulo computational practicalities, do wider networks outperform narrower ones? Answering this question has been challenging, as conventional networks gain representational power with width, potentially masking any negative effects. Our analysis in this paper decouples capacity and width via the generalization of neural networks to Deep Gaussian Processes (Deep GP), a class of hierarchical models that subsume neural nets. In doing so, we aim to understand how width affects standard neural networks once they have sufficient capacity for a given modeling task. Our theoretical and empirical results on Deep GP suggest that *large width is generally detrimental to hierarchical models*. Surprisingly, we prove that even nonparametric Deep GP converge to Gaussian processes, effectively becoming shallower without any increase in representational power. The posterior, which corresponds to a mixture of data-adaptable basis functions, becomes less data-dependent with width. Our tail analysis demonstrates that width and depth have opposite effects: depth accentuates a model's non-Gaussianity, while width makes models increasingly Gaussian. We find there is a "sweet spot" that maximizes test set performance before the limiting GP behavior prevents adaptability, occurring at width $= 1$ or width $= 2$ for nonparametric Deep GP. These results make strong predictions about the same phenomenon in conventional neural networks: we show empirically that many neural network architectures need $10 - 500$ hidden units for sufficient capacity—depending on the dataset—but further width degrades test performance.

1 Introduction

Research has shown that deeper neural networks tend to be more expressive and efficient than wider networks under a variety of metrics [e.g. 19, 56, 60, 63, 67, 68, 70, 75]. Nevertheless, there is resurgent interest in wide models due in part to empirical successes [e.g. 83] and theoretical analyses of limiting behavior. When randomly initialized to create a distribution over functions, neural networks converge to Gaussian processes (**GP**) as width increases. This result, first proved for 2-layer networks [62], has been extended to deeper networks [49, 57], convolutional networks [35, 64], and other architectures [44, 79]. A similar limit exists for gradient-trained networks, which behave increasingly like kernel machines under the neural tangent kernel [e.g. 5, 7, 26, 45, 50, 80].

While these limits simplify analyses, there is something unsettling about reducing neural networks to kernel methods. Neal [62, p. 161] describes the GP limit as "disappointing," noting that "infinite networks do not have hidden units that represent 'hidden features'... often seen [as the] interesting aspect of neural network learning." Recent work indeed shows that learned hierarchical features can be exponentially more efficient than the fixed shallow representations of kernels [e.g. 3, 4, 7, 10, 12, 19, 37, 38, 53, 82]. At the same time, wider networks can more accurately model complex functions

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

[40]. Thus, wide limits appear to confound opposing phenomenon: increased capacity makes them more expressive, yet the loss of hierarchical features seems to make them less expressive. This may explain the mixed empirical performance of limiting models: outperforming finite width models in some scenarios [e.g. 8, 35, 51], yet falling short on more complex tasks [e.g. 7, 10, 32, 50, 73].

This paper aims to decouple these effects of large width. Our goal is to understand the inductive biases of wide networks, after a network has "sufficient" capacity for a given modeling task. We ask: *If we control for the effects of increased capacity, what—if any—value remains in wide networks?*

To achieve this control, we note that a typical neural network layer corresponds to a finite basis, where elementwise nonlinearities transform each hidden feature into a *single basis function*. In order to decouple width from capacity, one could generalize these layers so that each nonlinearity produces any number of basis functions; if each hidden feature gives rise to an infinite and universal basis, then hidden layers would have infinite representational capacity *regardless of width*. This generalization is in fact a well-studied class of hierarchical models—Deep Gaussian Processes (**Deep GP**) [17, 22, 24, 25, 28, 30, 41, 71]—where standard neural net layers are replaced with vector-valued Gaussian processes. Indeed, typical neural networks are a degenerate Deep GP subclass [1, 2, 31, 65].

We therefore have a generalization of neural networks where capacity is controlled, from which we can glean insights about conventional networks that have sufficient representational power for a given modeling task. Surprisingly, despite using Gaussian processes as the primary hierarchical component, we prove that *Deep GP converge to (single-layer) GP in their infinite width limit* (Thm. 1). Troubling implications immediately ensue: large width is strictly detrimental to Deep GP, as the limiting model collapses to a shallower version of itself. We support this theorem with an analysis of neural network and Deep GP posteriors, which *become less adaptable as width increases*. Specifically, we show that the posterior mean corresponds to a mixture of functions drawn from data-dependent (and thus adaptive) reproducing kernel Hilbert spaces, formalizing the above claim from Neal [62]. As width increases, this mixture collapses to the data-independent kernel of the limiting GP, implying that wider models have less feature learning. Finally, we present a novel tail analysis which indicates that *width and depth have opposite effects*: depth accentuates non-Gaussianity, sharpening peaks and fattening tails, whereas width increases Gaussianity (Thms. 2 and 3).

Our theoretical results hold for Deep GP and conventional (parametric) neural networks alike. Experiments confirm that—after a model achieves sufficient capacity—*width becomes harmful to model fit and performance*. For nonparametric Deep GP, a width of 1 or 2 often achieves the best performance and fit. Neural networks naturally require more hidden units; nevertheless, performance degrades after a certain width. On small datasets ($N \leq 1000$) with low dimensionality, we find that models with ≤ 16 hidden units achieve best test set performance. On larger datasets like CIFAR10, this "sweet spot" occurs later (at ≈ 500 hidden units for sufficiently deep models), yet performance degrades beyond this width. Our findings strongly suggest that narrower models have better inductive biases, and wide models perform well *in spite of*—not because of—large width.

2 Setup

2.1 Related Work

Effects of width. Works have shown that, given finite parameters, deeper models are more expressive than wider models [56, 60, 67, 68, 75]. Similarly to our work, Aitchison [2] shows that width potentially harms prior and posterior expressivity of deep linear models. Nevertheless, wide models have been shown to have favorable optimization landscapes [6, 26, 52, 63, 74] and are resistant to overfitting via double descent [11, 18, 61]. Our work controls for these factors by examining nonparametric hierarchical models with exact Bayesian inference, and thus does not disagree with these other works. Infinite width limits have received renewed interest in Bayesian [35, 44, 50, 57, 62, 64, 79] and non-Bayesian [5, 7, 20, 26, 39, 45, 50, 58, 80, 81] settings. Most of these works show that neural networks converge to kernel methods, though recent work suggests that this limiting behavior can be avoided with different parameterizations [e.g. 20, 39, 58, 81]. Similarly to Lee et al. [50], our Deep GP limit analysis sequentially increases the width of each layer, though we hypothesize a similar proof exists where the width of all layers increases simultaneously (akin to [57]).

Deep GP are introduced in Damianou and Lawrence [25]. A large portion of Deep GP research has thus far focused on scalable approximate inference methods [17, 22, 23, 30, 41, 65, 71, 77]. Though

prior work has studied tail properties of neural networks [76, 84] and Deep GP with RBF kernels [55], our work is—to the best of our knowlege—the first general result for Deep GP tails. Duvenaud et al. [31] and Dunlop et al. [27] investigate pathological behaviors that arise with depth, while Agrawal et al. [1] note that "bottlenecked" Deep GP improve performance and correlations among predictive tasks. Our work complements these analysis by thoroughly characterizing the effects of width.

Connections between Deep GP and neural networks. Many researchers have noted connections between neural networks and Deep GP [e.g. 22, 29, 33, 54]. Duvenaud et al. [31] suggest that infinitely-wide neural networks with intermediate bottleneck layers are nonparametric Deep GP. Agrawal et al. [1] formalize this connection, but note that not all Deep GP can be constructed from bottlenecked neural networks (see Appx. E). In contrast to these prior works, we avoid reducing Deep GP to neural networks, and instead reduce neural networks to degenerate Deep GP.

2.2 A Covariance Perspective on Gaussian Process Limiting Behavior

To decouple the effects of increasing width and capacity, we first prove a new result about GP limits for a more general class of models, including Deep GP as well as typical neural networks. This result forms a necessary foundation for the subsequent theorems that are one main contribution of this work. To begin, note that the proof technique introduced by Neal [62] and extended by others [35, 44, 49, 57, 64, 79] relies on the multivariate central limit theorem, which requires a model with additive structure. Deep GP do not generally decompose in an additive manner, so we establish a more general proof technique. For simplicity, we first present it in the context of neural networks, and then extend it to a more general class of models.

Consider the 2-layer neural network $f_2(\mathbf{f}_1(\mathbf{x}))$, with $\mathbf{f}_1 : \mathbb{R}^D \to \mathbb{R}^{H_1}$ and $f_2 : \mathbb{R}^{H_1} \to \mathbb{R}$:

$$\mathbf{f}_1(\cdot) = \mathbf{W}_1^\top(\cdot) + \beta \mathbf{b}_1, \qquad f_2(\cdot) = \tfrac{1}{\sqrt{H_1}} \mathbf{w}_2^\top \boldsymbol{\sigma}(\cdot) + \beta b_2. \tag{1}$$

$\boldsymbol{\sigma}(\cdot)$ is an elementwise nonlinearity, β is a positive constant, and $\mathbf{W}_1$, $\mathbf{b}_1$, $\mathbf{w}_2$, and b_2 are i.i.d. Normal. With randomly initialized parameters, $f_2(\mathbf{f}_1(\cdot)) : \mathbb{R}^D \to \mathbb{R}$ is a prior distribution over functions, and this distribution converges to a GP in the infinite width limit [62].

Lemma 1. *The neural network defined in Eq. (1) is a Gaussian process if and only if—for any finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$—the conditional prior covariance $\mathbb{E}_{\mathbf{f}_2|\mathbf{X},\mathbf{W}_1,\mathbf{b}_1}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ is almost surely equal to the marginal prior covariance $\mathbb{E}_{\mathbf{f}_2|\mathbf{X}}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$, where $\mathbf{f}_2 \mid \mathbf{X} \triangleq [f_2(\mathbf{f}_1(\mathbf{x}_1)), \ldots, f_2(\mathbf{f}_1(\mathbf{x}_N))]$.*

Proof. By definition, $f_2(\mathbf{f}_1(\cdot))$ is a GP if and only if $\mathbf{f}_2 \mid \mathbf{X}$ is multivariate Gaussian for any $\mathbf{X}$. From Eq. (1), we have $p(\mathbf{f}_2 \mid \mathbf{X}, \mathbf{W}_1, \mathbf{b}_1) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X}))$, where $[\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X})]_{ij} = \beta^2 + \frac{1}{H_1}\boldsymbol{\sigma}(\mathbf{W}_1^\top\mathbf{x}_i + \beta\mathbf{b}_1)^\top\boldsymbol{\sigma}(\mathbf{W}_1^\top\mathbf{x}_j + \beta\mathbf{b}_1)^\top$ ($\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X})$ is the appropriate kernel Gram matrix). Using Jensen's inequality, we have a lower bound on the characteristic function of $\mathbf{f}_2 \mid \mathbf{X}$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{f}_2|\mathbf{X}}\left[\exp\left(i\mathbf{t}^\top\mathbf{f}_2\right)\right] &= \mathop{\mathbb{E}}_{\mathbf{W}_1,\mathbf{b}_1}\left[\mathop{\mathbb{E}}_{\mathbf{f}_2|\mathbf{X},\mathbf{W}_1,\mathbf{b}_1}\left[\exp\left(i\mathbf{t}^\top\mathbf{f}_2\right)\right]\right] && \text{(law of total expectation)}\\
&= \mathbb{E}_{\mathbf{W}_1,\mathbf{b}_1}\left[\exp\left(-\tfrac{1}{2}\mathbf{t}^\top\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X})\mathbf{t}\right)\right] && \text{(char. func. of a Gaussian)}\\
&\geq \exp\left(-\tfrac{1}{2}\mathbf{t}^\top \mathop{\mathbb{E}}_{\mathbf{W}_1,\mathbf{b}_1}\left[\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X})\right]\mathbf{t}\right). && \text{(convexity of exp)}
\end{aligned}$$

This lower bound happens to be the characteristic function of $\mathcal{N}(\mathbf{0}, \mathbb{E}_{\mathbf{W}_1,\mathbf{b}_1}\left[\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X})\right])$. Since exp is strictly convex, the characteristic function of $\mathbf{f}_2 \mid \mathbf{X}$ equals the Gaussian lower bound if and only if $p(\mathbf{K}_{\mathbf{W}_1,\mathbf{b}_1}(\mathbf{X},\mathbf{X}) \mid \mathbf{W}_1, \mathbf{b}_1) = \mathbb{E}_{\mathbf{f}|\mathbf{X},\mathbf{W}_1,\mathbf{b}_1}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ is a constant with probability 1. □

Seeing that $\frac{1}{H_1}\boldsymbol{\sigma}(\mathbf{W}_1^\top\mathbf{x}_i + \beta\mathbf{b}_1)^\top\boldsymbol{\sigma}(\mathbf{W}_1^\top\mathbf{x}_j + \beta\mathbf{b}_1)^\top$ becomes a.s. constant as $H_1 \to \infty$, Lemma 1 re-establishes the result of Neal [62] (see Appx. E.1). Critically, unlike Neal's proof, Lemma 1 neither relies on the central limit theorem nor requires $f_2(\mathbf{f}_1(\cdot))$ to be a neural network; it holds if $p(\mathbf{f}_2 \mid \mathbf{f}_1(\mathbf{x}_1), \ldots, \mathbf{f}_1(\mathbf{x}_N))$ is Gaussian. Therefore, we can generalize it to a larger class of models:

Lemma 2. *Let $f_2(\mathbf{f}_1(\cdot)) : \mathbb{R}^D \to \mathbb{R}$ be a hierarchical model where $f_2(\cdot) : \mathbb{R}^{H_1} \to \mathbb{R}$ is a GP and $\mathbf{f}_1(\cdot) : \mathbb{R}^D \to \mathbb{R}^{H_1}$ is a random vector-valued function (including a multilayer hierarchical model). Then $f_2(\mathbf{f}_1(\cdot))$ is a GP if and only if $\mathbb{E}_{\mathbf{f}_2|\mathbf{X},\mathbf{f}_1(\cdot)}\left[\mathbf{f}_2\mathbf{f}_2^\top\right] = \mathbb{E}_{\mathbf{f}_2|\mathbf{X}}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ a.s. for all $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$.*

The covariance perspective from Lemmas 1 and 2 is revealing about GP limits. As $\mathbb{E}_{\mathbf{f}_2|\mathbf{X},\mathbf{f}_1(\cdot)}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ converges to $\mathbb{E}_{\mathbf{f}_2|\mathbf{X}}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ the model output becomes less and less dependent on $\mathbf{f}_1(\cdot)$. In other words, $f_2(\mathbf{f}_1(\cdot))$ *loses its hierarchical nature*. Crucially, Lemma 2 has no requirements about $f_2(\mathbf{f}_1(\cdot))$ transitioning from a finite to infinite basis, and it does not require $f_2(\mathbf{f}_1(\cdot))$ to have additive structure. We demonstrate its generality in the next section with surprising—and troubling—implications.

3 Deep Gaussian Processes Collapse to Shallow Gaussian Processes

Deep GP [17, 24, 25, 71] are hierarchical models where layers $\mathbf{f}_1(\cdot)\ldots f_L(\cdot)$ are (vector-valued) GP:

$$\text{DGP}(\mathbf{x}) = f_L \circ \ldots \circ \mathbf{f}_1(\mathbf{x}), \quad \mathbf{f}_i(\cdot) = [f_i^{(1)}(\cdot), \ldots, f_i^{(H_i)}(\cdot)], \quad f_i^{(j)}(\cdot) \overset{\text{i.i.d}}{\sim} \mathcal{GP}\left[0, k_i(\cdot,\cdot)\right]. \quad (2)$$

H_i is the width of the ith GP, and the output dimensions of each $\mathbf{f}_i(\cdot)$ are independent. By using GP as the primary hierarchical building blocks, Deep GP are generally nonparametric and, assuming the GP layers use universal kernels [59], have infinite representational capacity (see Appx. B.1).

Deep GP versus GP. Deep GP seek to offer more expressivity: conventional single-layer GP—though also nonparametric—are inherently limited by the choice of the prior covariance function [17, 71]. For example, a GP with a RBF covariance is not suitable for data with discontinuities or sharp changes. However, stacking two RBF GP together—$f_2(\mathbf{f}_1(\cdot))$—can overcome this limitation, since $\mathbf{f}_1(\mathbf{x})$ can encode a warping of $\mathbf{x}$ that "smoothes" the input data for $f_2(\cdot)$ (as we will show in Fig. 1). Empirically, Deep GP have been shown to offer much more accurate predictive posteriors than standard GP [e.g. 16, 22, 24, 25, 28, 41, 71].

Deep GP versus neural networks. (Bayesian) feed-forward neural networks are a strict subclass of Deep GP, albeit a degenerate one [2, 54, 65]. The first neural network layer is a GP with a linear kernel, while subsequent layers are GP with the kernel $k(\mathbf{z},\mathbf{z}') = \beta^2 + \frac{1}{H_{i-1}}\sum_{i=1}^{H_{i-1}}\sigma(z_i)\,\sigma(z_i)$. A neural network, unlike other Deep GP, does not have infinite capacity. Put loosely, a single neural network hidden unit corresponds to a single basis, while a single Deep GP unit can correspond to an infinite basis. We refer the reader to [1, 2, 22, 29, 31, 65] and Appx. B.2 for more discussion on this connection. The critical takeaway is that all of our Deep GP results apply to neural networks as well.

3.1 Wide Deep GP are Gaussian Processes

Having established a model where width does not effect capacity, we now establish what remaining effects width has. Empirical evidence suggests that the choice of width impacts Deep GP predictions [17, 41]. In practice it is common to make Deep GP as wide as comparably-sized neural networks; Salimbeni and Deisenroth [71] for example train Deep GP with ≥ 30 units per layer.

Surprisingly, here we prove that—in the limit of infinite width—Deep GP collapse to single-layer Gaussian processes. Our proof relies on the conditional covariance analysis of the previous section. If the GP layers have non-pathological covariance functions[1]—the Deep GP conditional covariance becomes almost surely constant with width (see Lemma 3, Appx. E). Combining this with Lemma 2:

Theorem 1. *Let $f_L \circ \ldots \circ \mathbf{f}_1(\mathbf{x})$ be a zero-mean Deep GP (Eq. 2), where each layer satisfies Assumptions 1 and 2 (non-pathological prior covariances that scale with dimensionality—see Appx. E.2). Then $\lim_{H_L\to\infty}\cdots\lim_{H_1\to\infty} f_L \circ \ldots \circ \mathbf{f}_1(\mathbf{x})$ converges in distribution to a (single-layer) GP.*

(See Appx. E for proof.) The implications of Thm. 1 are paradoxical and unsettling. Deep GP are motivated as a more powerful model than standard GP. However, as we make the model wider, we arrive back where we started—a Gaussian process (although one with a different prior covariance).

A neural network gains representational power in its GP limit, transitioning from a finite-basis model to a nonparametric model. The Deep GP limit on the other hand has no additional representational power, since Deep GP are already universal approximators at any width. (Indeed this fact motivates their use as a control.) The only difference between finite and infinite width Deep GP is the prior distribution itself: transitioning from non-Gaussian to Gaussian with increasing width. In the next section, we investigate how this transition affects model performance.

[1] Any textbook kernel (isotropic kernels, dot product kernels, etc.) or any covariance function with a Fourier-Steiljes representation is "non-pathological;" see Appx. E.2 for formal assumptions.

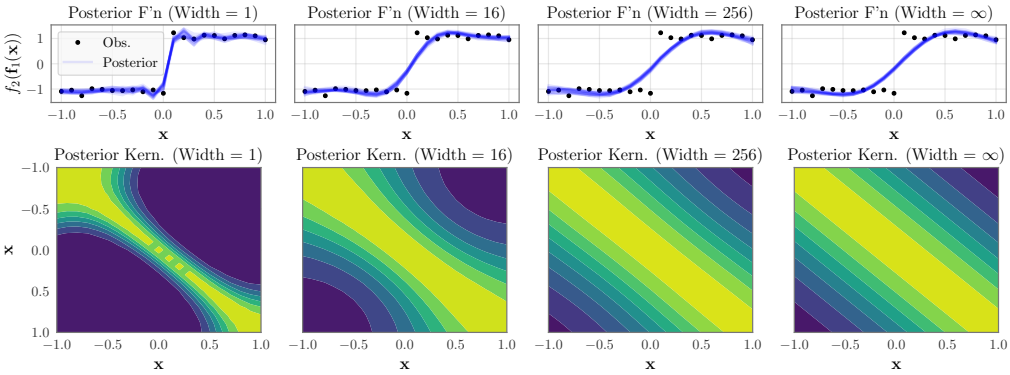

Figure 1: **Top:** Posterior of 2-layer RBF Deep GP fit to a noisy step function. A width-1 Deep GP fits the discontinuity at $\mathbf{x} = 0$. As with increases, the Deep GP converges to a GP with a stationary covariance unable to fit the step. **Bottom:** Average posterior covariance $\mathbb{E}_{\mathbf{f}_1(\mathbf{x}),\mathbf{f}_1(\mathbf{x}')|\mathbf{y}}\left[k_2(\mathbf{f}_1(\mathbf{x}), \mathbf{f}_1(\mathbf{x}'))\right]$. The width-1 posterior covariance is non-stationary, with little covariance around $\mathbf{x} = 0$. As width increases, the posterior covariance becomes stationary (as seen by the kernel's constant diagonals).

4 Large Width Limits the Adaptability of Hierarchical Posteriors

Even with Thm. 1 and its troubling suggestions, it is not immediately clear exactly what is lost in the infinite-width limit. Here we quantify specific differences in the predictive capabilities of narrow versus wide models. In particular, we analyze Deep GP/neural network posterior distributions, rather than focusing on a single model trained through optimization. We show that these posteriors correspond to a mixture of *data-dependent adaptable bases*; however, as width increases this mixture collapses to the (data-independent) basis of the limiting GP. This result formalizes the often vague notion of *feature learning*, and demonstrates that it is indeed lost in kernel limits.

Hierarchical posteriors correspond to a data-adaptable bases. Consider the (finite-width) 2-layer Deep GP $f_2(\mathbf{f}_1(\cdot))$, where $k_1(\cdot,\cdot)$ and $k_2(\cdot,\cdot)$ are the covariance functions of $\mathbf{f}_1(\cdot)$ and $f_2(\cdot)$. Given training data $\mathbf{X}, \mathbf{y}$, define $\mathbf{F}_1 \triangleq [\mathbf{f}_1(\mathbf{x}_1), \ldots, \mathbf{f}_1(\mathbf{x}_N)]$ and $\mathbf{f}_2 \triangleq [f_2(\mathbf{f}_1(\mathbf{x}_1)), \ldots, f_2(\mathbf{f}_1(\mathbf{x}_N))]$. Let $\mathbf{x}^*$ be a test input, and let $\mathbf{f}_1^*$ and f_2^* equal $\mathbf{f}_1(\mathbf{x}^*)$ and $f_2(\mathbf{f}_1(\mathbf{x}^*))$ (see Fig. 5 in Appx. B.3 for a graphical model). Crucially, $\mathbf{f}_2$ and f_2^* only depend on $\mathbf{F}_1$ and $\mathbf{f}_1^*$ through the covariances $\mathbf{K}_2(\mathbf{F}_1, \mathbf{F}_1)$, $\mathbf{k}_2(\mathbf{F}_1, \mathbf{f}_1^*)$, and $k_2(\mathbf{f}_1^*, \mathbf{f}_1^*)$ (which we abbreviate as $\mathbf{K}_2$, $\mathbf{k}_2^*$, and k_2^{**}):

$$p(\mathbf{f}_2 \mid \mathbf{K}_2) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_2), \quad p(f_2^* \mid k_2^{**}, \mathbf{k}_2^*, \mathbf{K}_2, \mathbf{f}_2) \sim \mathcal{N}\left(\mathbf{k}_2^{*\top}\mathbf{K}_2^{-1}\mathbf{f}_2,\ k_2^{**} - \mathbf{k}_2^{*\top}\mathbf{K}_2^{-1}\mathbf{k}_2^*\right),$$

By D-separation [e.g. 15, Ch. 8], we can factorize the posterior distribution as:

$$p(f_2^*, \mathbf{f}_2, \mathbf{K}_2, \mathbf{k}_2^*, k_2^{**} \mid \mathbf{y}) \;=\; p(f_2^* \mid \mathbf{f}_2, \mathbf{K}_2, \mathbf{k}_2^*, k_2^{**})\, p(\mathbf{f}_2 \mid \mathbf{K}_2, \mathbf{y})\, p(\mathbf{K}_2, \mathbf{k}_2^*, k_2^{**} \mid \mathbf{y}). \tag{3}$$

See derivation in Appx. B.3. Applying the factorization in Eq. (3), the posterior mean is:

$$\mathop{\mathbb{E}}_{f_2^*|\mathbf{y}}\left[f_2^*\right] = \mathop{\mathbb{E}}_{\mathbf{K}_2,\mathbf{k}_2^*|\mathbf{y}}\left[\mathop{\mathbb{E}}_{\mathbf{f}_2|\mathbf{K}_2,\mathbf{y}}\left[\mathbf{k}_2^{*\top}\mathbf{K}_2^{-1}\mathbf{f}_2\right]\right] = \mathop{\mathbb{E}}_{\mathbf{K}_2,\mathbf{k}_2^*|\mathbf{y}}\left[\mathbf{k}_2^{*\top}\overbrace{\mathbf{K}_2^{-1}\mathop{\mathbb{E}}_{\mathbf{f}_2|\mathbf{K}_2,\mathbf{y}}\left[\mathbf{f}_2\right]}^{\boldsymbol{\alpha}}\right] \tag{4}$$

$$= \mathop{\mathbb{E}}_{\mathbf{f}_1(\mathbf{x}^*),\mathbf{f}_1(\mathbf{x}_1),\ldots,\mathbf{f}_1(\mathbf{x}_N)|\mathbf{y}}\left[\textstyle\sum_{i=1}^N \alpha_i\, k_2(\mathbf{f}_1(\mathbf{x}_i), \mathbf{f}_1(\mathbf{x}^*))\right], \tag{5}$$

where the second line follows from $\mathbf{K}_2$ and $\mathbf{k}_2^*$ being deterministic given $\mathbf{f}_1(\mathbf{x}^*)$, $\mathbf{f}_1(\mathbf{x}_1)$, …, $\mathbf{f}_1(\mathbf{x}_N)$. The term inside the Eq. (5) expectation is a function from the reproducing kernel Hilbert space (RKHS) defined by $k_2(\mathbf{f}_1(\cdot), \mathbf{f}_1(\cdot))$. We can thus interpret this expectation as an infinite mixture of functions from different Hilbert spaces. Because the mixture distribution $p(\mathbf{f}_1(\mathbf{x}^*), \mathbf{f}_1(\mathbf{x}_1), \ldots, \mathbf{f}_1(\mathbf{x}_N) \mid \mathbf{y})$ depends on $\mathbf{y}$, Eq. (5) is an *adaptive data-dependent mixture of RKHS*.

Adaptability is lost in the Gaussian process limit. Recall from Lemma 2 that the conditional prior covariance becomes deterministic as $f_2(\mathbf{f}_1(\cdot))$ converges to a GP. In other words, the prior and posterior distributions over $\mathbf{K}_2$ and $\mathbf{k}_2^*$ become atomic: $p(\mathbf{K}_2, \mathbf{k}_2^*) = p(\mathbf{K}_2, \mathbf{k}_2^* \mid \mathbf{y}) = \delta\left[\mathbf{K}_{\text{lim}}, \mathbf{k}_{\text{lim}}^*\right]$, where $\mathbf{K}_{\text{lim}}$ and $\mathbf{k}_{\text{lim}}^*$ are shorthand for $\mathbb{E}\left[\mathbf{f}_2\mathbf{f}_2^\top\right]$ and $\mathbb{E}\left[\mathbf{f}_2 f_2^*\right]$ respectively. Eq. (4) thus collapses to:

$$\lim_{H_1\to\infty} \mathop{\mathbb{E}}_{f_2^*|\mathbf{y}}\left[f_2^*\right] = \mathop{\mathbb{E}}_{\delta[\mathbf{K}_{\text{lim}},\mathbf{k}_{\text{lim}}^*]}\left[\mathbf{k}_2^{*\top}\boldsymbol{\alpha}\right] = \textstyle\sum_{i=1}^N \alpha_i\, k_{\text{lim}}(\mathbf{x}_i, \mathbf{x}^*), \tag{6}$$

which is no longer a mixture of functions from different RKHS. It is instead a function from a single RKHS (that of the limiting GP prior). In other words, while Deep GP (and neural networks) perform *kernel learning (or feature learning)* to adapt to training data, this ability is lost with large width.

Example. Consider a Deep GP with RBF covariances $k_1(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2/(2D)\right)$ and $k_2(\mathbf{f}_1(\mathbf{x}), \mathbf{f}_1(\mathbf{x}')) = \exp\left(-\|\mathbf{f}_1(\mathbf{x}) - \mathbf{f}_1(\mathbf{x}')\|^2/(2H_1)\right)$. As we show in Appx. G, this Deep GP converges to a GP with $k_{\text{lim}}(\mathbf{x}, \mathbf{x}') = \exp(\exp(-\|\mathbf{x} - \mathbf{x}'\|^2/(2D)) - 1)$. Note that this limiting covariance is *stationary* and is ill-equipped to model the data step in Fig. 1. However, because $\mathbf{f}_1(\cdot)$ is nonlinear, $k_2(\mathbf{f}_1(\mathbf{x}), \mathbf{f}_1(\mathbf{x}'))$ is thus *nonstationary*. Fig. 1 (top left) shows that the width-1 Deep GP posterior accurately models both this data. The posterior covariance $\mathbb{E}_{\mathbf{f}_1(\mathbf{x}), \mathbf{f}_1(\mathbf{x}')|\mathbf{y}}\left[k_2(\mathbf{f}_1(\mathbf{x}), \mathbf{f}_1(\mathbf{x}'))\right]$ (bottom left) features long-range correlations near $\mathbf{x} = \pm 1$ and short-range correlations near $\mathbf{x} = 0$. As width increases, we lose this nonstationarity and the posterior becomes a worse fit.

5 The Difference Between Width and Depth: A Tail Analysis

Our work so far has troubling implications for large width. On the other hand, empirical evidence has shown that depth improves Deep GP performance—as it does for neural nets [e.g. 41, 65, 71] (though pathologies can emerge [27, 31]). Through a novel tail analysis, we show that width makes Deep GP priors more Gaussian, while depth makes them less Gaussian. In other words, *width and depth have opposite effects on Deep GP tails*, results that again also apply to typical neural networks.

Deep GP/neural networks are sharply peaked and heavy tailed. The proof technique used in Lemma 1 can be used to similarly bound the moment generating function of Deep GP marginals:

$$\mathop{\mathbb{E}}_{\mathbf{f}_2}\left[e^{\mathbf{t}^\top \mathbf{f}_2}\right] = \mathop{\mathbb{E}}_{\mathbf{F}_1}\left[\mathop{\mathbb{E}}_{\mathbf{f}_2|\mathbf{F}_1}\left[e^{\mathbf{t}^\top \mathbf{f}_2}\right]\right] \geq \exp\left(\frac{1}{2}\mathbf{t}^\top \mathop{\mathbb{E}}_{\mathbf{F}_1}\left[\mathbf{K}_2(\mathbf{F}_1, \mathbf{F}_1)\right]\mathbf{t}\right) = \mathop{\mathbb{E}}_{\mathbf{g}\sim\mathcal{N}(\mathbf{0}, \mathbf{K}_{\text{lim}})}\left[e^{\mathbf{t}^\top \mathbf{g}}\right], \quad (7)$$

where $\mathbf{K}_{\text{lim}} = \mathbb{E}_{\mathbf{f}_2}\left[\mathbf{f}_2\mathbf{f}_2^\top\right] = \mathbb{E}_{\mathbf{F}_1}\left[\mathbf{K}_2(\mathbf{F}_1, \mathbf{F}_1)\right]$. Generalizing these bounds to deeper models, we have:

Theorem 2. *Let $f_L \circ \ldots \circ \mathbf{f}_1(\cdot)$ be a zero-mean Deep GP. Given a finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, define $\mathbf{f}_\ell = [(f_\ell \circ \ldots \circ \mathbf{f}_1(\mathbf{x}_1)), \ldots, (f_\ell \circ \ldots \circ \mathbf{f}_1(\mathbf{x}_N))]$ for $\ell \in [1, L]$, and define $\mathbf{K}_{lim} = \mathbb{E}_{\mathbf{f}_L}\left[\mathbf{f}_L\mathbf{f}_L^\top\right]$. If $\mathbb{E}_{\mathbf{f}_\ell|\mathbf{f}_{\ell-1}}\left[\mathbf{f}_\ell\mathbf{f}_\ell^\top\right]$ is not constant a.s. for any $\ell \in [2, L]$, then, $p(\mathbf{f}_L = \mathbf{0}) > \mathcal{N}(\mathbf{g} = \mathbf{0}; \mathbf{0}, \mathbf{K}_{lim})$.*

Theorem 3. *Let $\mathbf{t} \in \mathbb{R}^N$. Using the same setup, notation, and assumptions as Thm. 2, the odd moments of $\mathbf{t}^\top\mathbf{f}_L$ are zero and the even moments larger than 2 are strictly super-Gaussian, i.e. $\mathbb{E}_{\mathbf{f}_L}\left[(\mathbf{t}^\top\mathbf{f}_L)^r\right] > \mathbb{E}_{\mathbf{g}\sim\mathcal{N}(\mathbf{0}, \mathbf{K}_{lim})}\left[(\mathbf{t}^\top\mathbf{g})^r\right]$ for all even $r \geq 4$. Moreover, if $k_L(\cdot, \cdot)$ is finite almost everywhere, the moment generating function $\mathbb{E}_{\mathbf{f}_L}\left[\exp(\mathbf{t}^\top\mathbf{f}_L)\right]$ exists and is strictly super-Gaussian.*

(See Appx. F for proofs.) Thm. 2 states that Deep GP marginals are more sharply peaked than a moment-matched Gaussian, while Thm. 3 states that they are also more heavy tailed.

Increasing depth leads to sharper peaks and heavier tails. To understand how depth affects this tail behavior, we examine the Jensen gap in Eq. (7). Consider a 3-layer Deep GP $f_3(\mathbf{f}_2(\mathbf{f}_1(\cdot)))$. If we extend Eq. (7) to 3-layer models, we see that the Jensen gap cascades:

$$\underbrace{\mathop{\mathbb{E}}_{\mathbf{F}_1}\left[\mathop{\mathbb{E}}_{\mathbf{F}_2|\mathbf{F}_1}\left[\exp\left(\frac{1}{2}\mathbf{t}^\top\mathbf{K}_3\mathbf{t}\right)\right]\right]}_{\text{MGF of 3-layer Deep GP marginal}} \geq \underbrace{\mathop{\mathbb{E}}_{\mathbf{F}_1}\left[\exp\left(\frac{1}{2}\mathbf{t}^\top \mathop{\mathbb{E}}_{\mathbf{F}_2|\mathbf{F}_1}\left[\mathbf{K}_3\right]\mathbf{t}\right)\right]}_{\text{MGF of 2-layer Deep GP marginal}} \geq \underbrace{\exp\left(\frac{1}{2}\mathbf{t}^\top \mathop{\mathbb{E}}_{\mathbf{F}_1}\left[\mathop{\mathbb{E}}_{\mathbf{F}_2|\mathbf{F}_1}\left[\mathbf{K}_3\right]\right]\mathbf{t}\right)}_{\text{MGF of } \mathcal{N}(\mathbf{0}, \mathbb{E}_{\mathbf{F}_1}\left[\mathbb{E}_{\mathbf{F}_2|\mathbf{F}_1}\left[\mathbf{K}_3\right]\right])},$$

where $\mathbf{K}_3$ is short for $\mathbf{K}_3(\mathbf{F}_2(\mathbf{F}_1(\mathbf{X})), \mathbf{F}_2(\mathbf{F}_1(\mathbf{X})))$. The middle term is the moment generating function of a 2-layer Deep GP marginal (where the second layer has covariance $\mathbb{E}_{\mathbf{f}_2(\cdot)}\left[k_3(\mathbf{f}_2(\cdot), \mathbf{f}_2(\cdot))\right]$). The right-most term is the moment generating function of a (single-layer) Gaussian. Generalizing this cascade, we see that deeper models are more heavy-tailed. A similar analysis on the characteristic function shows that the peak at the prior mean also becomes sharper with depth (see Appx. F).

Adding additional layers to a Deep GP will change the model's prior covariance, and thus the effects of depth cannot solely be explained by a tail analysis [27, 31]. Nevertheless, if we control for this change in covariance, we indeed see that depth leads to heavier tails. In Fig. 2 we compare 2-layer and 3-layer Deep GP. The 3-layer models use GP layers with additively-decomposing RBF covariances, while the 2-layer models use layers constructed to match the 3-layer models' prior covariance (see Appx. H for construction details). The $N = 2$ marginal densities for the 3-layer models (bottom row) are more stretched than the 2-layer densities (top row). We further confirm these effects in Appx. D.

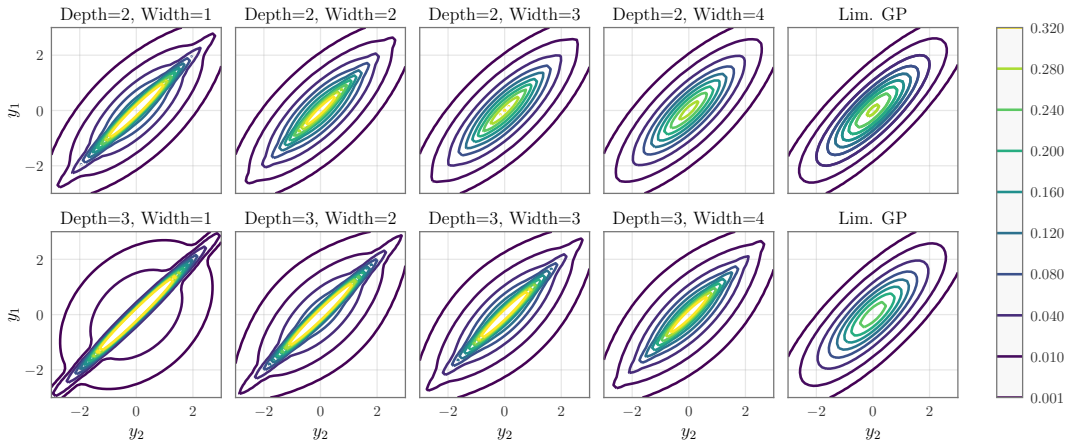

Figure 2: Marginal density $p(y_1, y_2 \mid \mathbf{x}_1, \mathbf{x}_2)$ for zero-mean Deep GP of various depths and widths on the $N = 2$ dataset $\mathbf{x}_1 = -0.5$, $\mathbf{x}_2 = 0.5$. All 2-layer models have the same second moments (covariance is that of the 3-layer width-1 RBF-RBF-RBF Deep GP). **Left to right:** width increases, the marginal becomes increasingly Gaussian, tails become thinner, and the peak at $[y_1, y_2] = \mathbf{0}$ loses density. **Top to bottom:** depth increases, tails become fatter, and the peak becomes sharper.

Increasing width leads to flatter peaks and Gaussian tails. Conversely, consider what happens when we make the model wider. We define the sequence of increasingly wide 2-layer Deep GP:

$$\left\{ \mathrm{DGP}^{(m)}(\cdot) \triangleq \frac{1}{\sqrt{m}} \sum_{i=1}^{m} f_2^{(i)}(f_1^{(i)}(\cdot)) \right\}, \qquad \begin{aligned} f_1^{(i)}(\cdot) &\overset{\text{i.i.d}}{\sim} \mathcal{GP}\left[0, k_1(\cdot,\cdot)\right], \\ f_2^{(i)}(\cdot) &\overset{\text{i.i.d}}{\sim} \mathcal{GP}\left[0, k_2(\cdot,\cdot)\right]. \end{aligned} \tag{8}$$

$\mathrm{DGP}^{(m)}(\cdot)$ is a width-m Deep GP, where the second layer decomposes additively over the m dimensions. By linearity of expectation, each model in the sequence shares the same prior covariance: $\mathbb{E}[\mathrm{DGP}^{(1)}(\mathbf{x})\,\mathrm{DGP}^{(1)}(\mathbf{x}')] = \mathbb{E}[\mathrm{DGP}^{(2)}(\mathbf{x})\,\mathrm{DGP}^{(2)}(\mathbf{x}')] = \ldots \triangleq k_{\text{lim}}(\mathbf{x}, \mathbf{x}')$. Though each model has the same marginal covariance, the *conditional* covariance $\mathbb{E}_{\mathbf{f}_2|\mathbf{F}_1}\left[\mathbf{f}_2\mathbf{f}_2^\top\right] = \frac{1}{m}\sum_{i=1}^{m} \mathbf{K}_2(\mathbf{f}_1^{(i)}, \mathbf{f}_1^{(i)})$ becomes increasingly concentrated around $\mathbf{K}_{\text{lim}}(\mathbf{X}, \mathbf{X})$ as m increases. This consequentially shrinks the Jensen gap in Eq. (7), and so the Deep GP marginals become increasingly Gaussian. We again visualize this effect in Fig. 2, which depicts marginal densities from 2-layer and 3-layer Deep GP of various width (see Appx. H for details). Compared with the limiting GP (right), the width-1 densities (left) appear sharper near $[0, 0]$ and more stretched at the tails. As width increases, the peaks and tails look increasingly Gaussian (see also Fig. 7 in Appx. D). In this sense, width has the opposite effect as depth—deeper marginals are less Gaussian, while wider marginals are more Gaussian.

6 Experiments

6.1 Regression with Deep GP and Bayesian Neural Networks

To isolate the effects of width and depth, each experiment compares Deep GP/Bayesian neural networks that share the same first and second prior moments, and the Deep GP models use GP layers with universal kernels. To remove any potential side effects from approximate inference methods, we sample Deep GP/neural network posteriors using NUTS [43] and do not use any stochastic inducing point [41, 71] or finite basis [22] approximations. This inference is costly and scales cubically with N; therefore, we subsample all training datasets to $N \leq 1000$. See Appx. H for experimental details.

Effect of width. We compare 2-layer Deep GP of various width on 6 regression datasets from the UCI dataset repository [9] (see Appx. D for 3-layer results). The first GP layers use a RBF kernel for the prior covariance, while the second layers use a sum of one-dimensional RBF covariance functions. We additionally compare against the limiting (single-layer) GP with the same prior covariance (**Lim. GP**). For each dataset, we choose hyperparameters that maximize the Lim. GP log marginal likelihood. In Fig. 3 (top row) we see a near-monotonic performance degradation as width increases. The width-2 optimum may represent the "sweet spot" for Deep GP width, but it may instead be a side-effect of

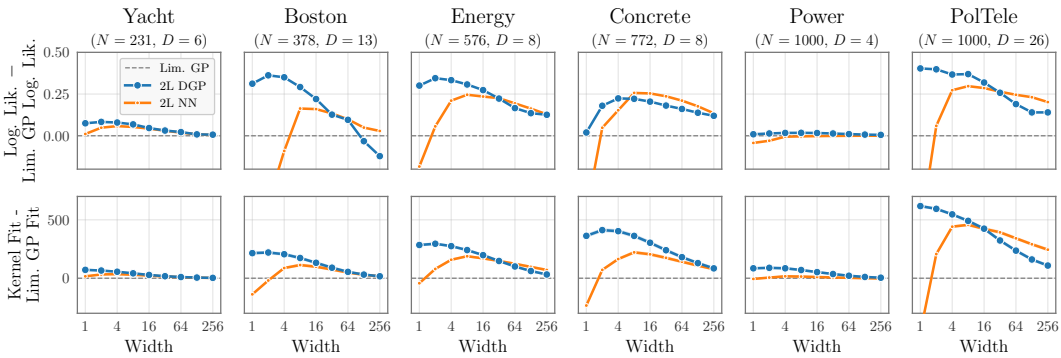

Figure 3: **Top:** Test set log likelihood (LL) of 2-layer Deep GP (and neural network) regression as a function of width (higher is better). Numbers are shifted so that 0 corresponds to the limiting GP log likelihood. Narrow models achieve the best log likelihood, and performance degrades with width. **Bottom:** Fit of the posterior kernel $k(\mathbf{f}_1(\cdot), \mathbf{f}_1(\cdot))$ on the training data, as measured by Gaussian log marginal likelihood (higher is better). 0 corresponds to the limiting GP log marginal likelihood. Fit becomes increasingly worse with width.

inference difficulties for width-1 models (see Appx. D for a control experiment). Regardless, as our theory predicts, *width is detrimental to Deep GP predictive performance*.

We repeat the experiment for 2-layer neural networks (and 3-layer models in Appx. D), where here the Lim. GP corresponds to the arc-cosine kernel [21, 49]. Fig. 3 indicates an optimal width with regards to test set log likelihood, usually between $8 - 16$ hidden units. We expect this optimum exists (and differs from the Deep GP optimum) because narrow models have too few basis functions for these datasets. Thus, after sufficient capacity, *width is harmful to Bayesian neural networks*.

Adaptable versus non-adaptable RKHS. One way to measure the "fit" of a kernel $k(\cdot, \cdot)$ on a regression training dataset $\mathbf{X}, \mathbf{y}$ is the Gaussian log marginal likelihood $\log \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})$, where σ^2 is an observational noise parameter [e.g. 69]. To demonstrate how Deep GP/neural network posteriors correspond to adaptable RKHS mixtures, the bottom row of Fig. 3 plots the "kernel fit" of $k_2(\mathbf{f}_1(\cdot), \mathbf{f}_1(\cdot))$ for posterior samples of $\mathbf{f}_1(\cdot)$ (see Eq. 5). A higher fit corresponds to a model that is better adapted to the dataset $\mathbf{X}, \mathbf{y}$. We see that narrower Deep GP almost universally achieve better kernel fit than wider Deep GP, which converge to the same fit as the limiting GP. (Standard deviations, depicted by shaded regions, are generally imperceptible.) Bayesian neural networks achieve best "kernel fit" at $8 - 16$ hidden units, and then converge to the limiting GP with further width.

Effect of depth, controlling for covariance. Table 1 displays Deep GP test set log likelihood as a function of depth. Again, we isolate the tail effects of depth by ensuring that all models share the same first and second moments. We construct a GP and a 2-layer Deep GP that match the moments of a width-2 3-layer Deep GP with RBF covariances, and we use hyperparameters that maximize the limiting GP marginal likelihood for each dataset. Note that computing the limiting covariance of ≥ 3 layer models involves intractable integrals that we approximate with quadrature (see Appx. G). Our findings confirm that—in this controlled setting—depth unlike width improves test set performance.

6.2 Standard (Optimized, Non-Bayesian) Neural Networks

Here our goal is to represent "real world" models without a Bayesian treatment, and thus we do not control the prior covariance across models. However, we scale the outputs of layer-ℓ by $1/\sqrt{H_\ell}$ to match the GP-limiting neural net construction in Eq. (1). We choose the L_2 regularization parameter so that it corresponds to a parameter prior of $\mathcal{N}(0, 20)$.[2] All models are trained without data augmentation. Fig. 4 depicts test set accuracy for increasingly wide models trained on MNIST [48] (left) and CIFAR-10 [47] (right). On MNIST we examine two sequences of 3-layer (non-convolutional) networks: one where the width of both hidden layers increases simultaneously ($H_1 = H_2$), and one where all but the final layer have fixed width ($H_1 = 256$). We test these

[2]The L_2 parameter equivalent to this prior decreases with N, and—due to the $\sqrt{1/H_\ell}$ scaling—effectively increases with width. The $\mathcal{N}(0, 20)$ prior is equivalent to a weight decay of 10^{-4} for $N = 50000$ and $H_\ell = 100$.

Table 1: Test set log likelihood (LL) of Deep GP regression as a function of depth (higher is better). Depth $= 1$ refers to the limiting GP. For each dataset, the models are constructed to have the same first and second moments. Unlike width, deeper models generally have better performance.

Depth	Yacht ($N = 231$, $D = 6$)	Boston ($N = 378$, $D = 13$)	Energy ($N = 576$, $D = 8$)	Concrete ($N = 772$, $D = 8$)	Power ($N = 1000$, $D = 4$)	PolTele ($N = 1000$, $D = 26$)
1	-0.532	-0.890	-0.477	-0.663	$\mathbf{-0.249}$	-0.476
2	-0.520	-0.684	-0.434	$\mathbf{-0.573}$	-0.260	-0.381
3	$\mathbf{-0.482}$	$\mathbf{-0.609}$	$\mathbf{-0.383}$	-0.620	-0.251	$\mathbf{-0.318}$

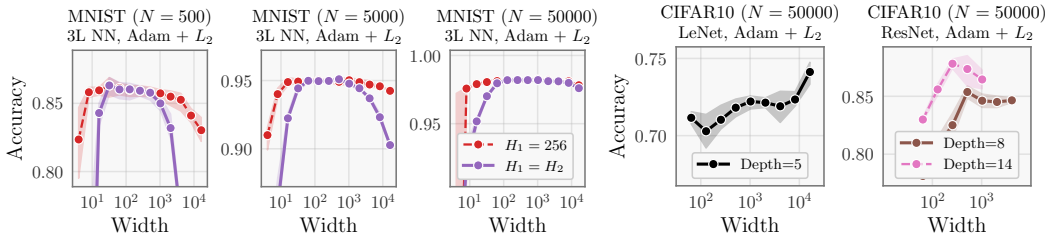

Figure 4: Effect of width on standard (non-Bayesian) neural networks. Shaded regions depict standard error. **Left:** 3-layer feed-forward networks trained on various subsets of MNIST. Test set accuracy is maximized when width ≈ 100, after which width becomes decremental. **Right:** LeNet and wide ResNets trained on CIFAR-10. The best LeNet has ≥ 16000 hidden units, likely because this model is too shallow for sufficient capacity. For deeper ResNet models, accuracy is best when width ≤ 500.

sequences on variously-sized subsets of the training data, and observe a maximum test set accuracy for width ≈ 100 networks. On CIFAR-10 we examine 5-layer LeNet models [48] and 8- and 14-layer wide ResNet models [42, 83]. For LeNet, we only vary the width of the fully connected layers, and we observe varied performance behavior with width, possibly indicating LeNet is too shallow for sufficient capacity on this dataset. Indeed, deeper ResNets—where we increase width of all layers simultaneously—achieve best accuracy when width ≤ 500. These results reiterate that even in typical networks, once sufficient capacity is reached, *large width adversely affects model performance*.

7 Discussion

This paper shows that, across typical neural networks, Deep GP, and Bayesian neural networks, *width is fundamentally detrimental after sufficient capacity is reached.*

Even with that result, we can ask when large width might still be beneficial? Our results assume that layers are conditionally Gaussian, and different priors may have different effects. Researchers have touted benefits of width, both in terms of a more favorable optimization landscape [6, 26, 52, 63, 74] and better generalization through the double descent phenomenon [11, 18, 61]. Moreover, there are (non-Bayesian) infinite-width constructions that avoid the limiting kernel behavior studied in this work [e.g. 20, 39, 58, 81]. We emphasize that our results do not conflict with these prior works, but rather reflect a different perspective. The models we study correspond to a Gaussian prior on parameters. Relaxing this correspondence, for example by changing the weight decay parameter, may lessen the consequences of width that we observe. Nevertheless, our results suggest that the inductive bias of width is harmful, even if these undesirable effects can be avoided via careful construction.

Finally, it is worth considering when one might still choose a conventional shallow GP over a deep model. An often-touted benefit of Gaussian processes is the ability to encode prior domain knowledge via the choice of covariance function. In Appx. C, we prove that certain prior covariances cannot be expressed by adaptable hierarchical models. For example, a Deep GP that is composed of stationary GP layers cannot model anti-correlations a priori (Thm. 4, Appx. C), whereas (single-layer) stationary GP can have positive and negative prior covariances. Nevertheless, Deep GP are capable of modeling many common covariance functions, including the RBF, Matérn, and rational quadratic kernels. In Appx. C we demonstrate a 2-layer Deep GP construction of any width that is capable of producing prior covariances that match most isotropic kernels (Thm. 5, Appx. C). In other words, a Deep GP can match the first and second moments of most GP, while also offering an adaptable posterior.

References

[1] D. Agrawal, T. Papamarkou, and J. Hinkle. Wide neural networks with bottlenecks are deep Gaussian processes. *Journal of Machine Learning Research*, 21(175):1–66, 2020.

[2] L. Aitchison. Why bigger is not always better: on finite and infinite neural networks. In *ICML*, 2020.

[3] Z. Allen-Zhu and Y. Li. What can ResNet learn efficiently, going beyond kernels? In *NeurIPS*, 2019.

[4] Z. Allen-Zhu and Y. Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

[5] Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *NeurIPS*, 2019.

[6] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *ICML*, 2019.

[7] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *NeurIPS*, 2019.

[8] S. Arora, S. S. Du, Z. Li, R. Salakhutdinov, R. Wang, and D. Yu. Harnessing the power of infinitely wide deep nets on small-data tasks. In *ICLR*, 2020.

[9] A. Asuncion and D. Newman. Uci machine learning repository, 2007. URL http://archive.ics.uci.edu/ml/index.php.

[10] Y. Bai and J. D. Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. In *ICLR*, 2020.

[11] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[12] Y. Bengio, O. Delalleau, and N. Le Roux. The curse of dimensionality for local kernel machines. 2005.

[13] P. Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

[14] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

[15] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[16] K. Blomqvist, S. Kaski, and M. Heinonen. Deep convolutional Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019.

[17] T. Bui, D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *ICML*, 2016.

[18] Y. Cao and Q. Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *NeurIPS*, 2020.

[19] M. Chen, Y. Bai, J. D. Lee, T. Zhao, H. Wang, C. Xiong, and R. Socher. Towards understanding hierarchical learning: Benefits of neural representations. In *NeurIPS*, 2020.

[20] L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *NeurIPS*, 2018.

[21] Y. Cho and L. Saul. Kernel methods for deep learning. In *NeurIPS*, 2009.

[22] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep Gaussian processes. In *ICML*, 2017.

[23] Z. Dai, A. C. Damianou, J. González, and N. D. Lawrence. Variational auto-encoded deep Gaussian processes. In *ICLR*, 2016.

[24] A. Damianou. *Deep Gaussian processes and variational propagation of uncertainty*. PhD thesis, University of Sheffield, 2015.

[25] A. Damianou and N. Lawrence. Deep Gaussian processes. In *AISTATS*, 2013.

[26] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *ICML*, 2019.

[27] M. M. Dunlop, M. A. Girolami, A. M. Stuart, and A. L. Teckentrup. How deep are deep Gaussian processes? *Journal of Machine Learning Research*, 19(54):1–46, 2018.

[28] V. Dutordoir, M. Wilk, A. Artemev, and J. Hensman. Bayesian image classification with deep convolutional Gaussian processes. In *AISTATS*, 2020.

[29] V. Dutordoir, J. Hensman, M. van der Wilk, C. H. Ek, Z. Ghahramani, and N. Durrande. Deep neural networks as point estimates for deep Gaussian processes. *arXiv preprint arXiv:2105.04504*, 2021.

[30] V. Dutordoir, H. Salimbeni, E. Hambro, J. McLeod, F. Leibfried, A. Artemev, M. van der Wilk, J. Hensman, M. P. Deisenroth, and S. John. Gpflux: A library for deep Gaussian processes. *arXiv preprint arXiv:2104.05674*, 2021.

[31] D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani. Avoiding pathologies in very deep networks. In *AISTATS*, 2014.

[32] S. Fort, G. K. Dziugaite, M. Paul, S. Kharaghani, D. M. Roy, and S. Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *NeurIPS*, 2020.

[33] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[34] J. R. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *NeurIPS*, 2018.

[35] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison. Deep convolutional networks as shallow Gaussian processes. In *ICLR*, 2019.

[36] M. G. Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of Machine Learning Research*, 2(Dec):299–312, 2001.

[37] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. Limitations of lazy training of two-layers neural networks. In *NeurIPS*, 2019.

[38] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. When do neural networks outperform kernel methods? In *NeurIPS*, 2020.

[39] E. Golikov. Towards a general theory of infinite-width limits of neural classifiers. In *ICML*, 2020.

[40] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*. MIT press Cambridge, 2016.

[41] M. Havasi, J. M. Hernández-Lobato, and J. J. Murillo-Fuentes. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In *NeurIPS*, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[43] M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[44] J. Hron, Y. Bahri, J. Sohl-Dickstein, and R. Novak. Infinite attention: NNGP and NTK for deep attention networks. In *ICML*, 2020.

[45] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[47] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[48] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[49] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. In *ICLR*, 2018.

[50] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*, 2019.

[51] J. Lee, S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, and J. Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. In *NeurIPS*, 2020.

[52] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

[53] Y. Li, T. Ma, and H. R. Zhang. Learning over-parametrized two-layer neural networks beyond NTK. In *COLT*, 2020.

[54] C. Louizos and M. Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *ICML*, 2016.

[55] C.-K. Lu, S. C.-H. Yang, X. Hao, and P. Shafto. Interpretable deep Gaussian processes with moments. In *AISTATS*, 2020.

[56] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *NeurIPS*, 2017.

[57] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *ICLR*, 2018.

[58] S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[59] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

[60] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In *NeurIPS*, 2014.

[61] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. In *ICLR*, 2020.

[62] R. M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[63] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In *ICML*, 2017.

[64] R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *ICLR*, 2019.

[65] S. W. Ober and L. Aitchison. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. *arXiv preprint arXiv:2005.08140*, 2020.

[66] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[67] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *NeurIPS*, 2016.

[68] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. In *ICML*, 2017.

[69] C. E. Rasmussen and C. Williams. *Gaussian processes for machine learning*, volume 1. MIT Press, 2006.

[70] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[71] H. Salimbeni and M. Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *NeurIPS*, 2017.

[72] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, pages 811–841, 1938.

[73] V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, J. Ragan-Kelley, L. Schmidt, and B. Recht. Neural kernels without tangents. In *ICML*, 2020.

[74] M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

[75] M. Telgarsky. Benefits of depth in neural networks. In *COLT*, 2016.

[76] M. Vladimirova, J. Verbeek, P. Mesejo, and J. Arbel. Understanding priors in Bayesian neural networks at the unit level. In *ICML*, 2019.

[77] Y. Wang, M. Brubaker, B. Chaib-Draa, and R. Urtasun. Sequential inference for deep Gaussian process. In *AISTATS*, 2016.

[78] A. Yaglom. *Correlation theory of stationary and related random functions*. Springer Series in Statistics, New York, 1987.

[79] G. Yang. Tensor programs I: Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In *NeurIPS*, 2019.

[80] G. Yang. Tensor programs II: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

[81] G. Yang and E. J. Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[82] G. Yehudai and O. Shamir. On the power and limitations of random features for understanding neural networks. In *NeurIPS*, 2019.

[83] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.

[84] J. A. Zavatone-Veth and C. Pehlevan. Exact priors of finite neural networks. *arXiv preprint arXiv:2104.11734*, 2021.

Checklist

1) For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appx. A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2) If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Assumptions are listed in the theorem/lemma statements, as well as in Appx. E.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appx. C, E and F.
3) If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is in supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appx. H.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our results for optimized networks are run 3-5 times, and indicate as such in the results figures. Due to compute resource constraints we only have one HMC run for each experiment in Sec. 6.1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appx. H.
4) If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] In Secs. 6.1 and 6.2 we supply citations for all datasets. In Appx. H we supply citations for all software packages.

(b) Did you mention the license of the assets? [No] To the best of our knowledge, none of the datasets we use have licenses. We mention that all software packages we use are open source.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use commonly-available public datasets that, to the best of our knowledge, do not pose any privacy or content concerns.

5) If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]