A. DKPs are kernel processes

We define a kernel process to be a distribution over positive (semi) definite matrices, $\mathcal{K}(\mathbf{K})$, parameterised by a postive (semi) definite matrix, $\mathbf{K} \in \mathbb{R}^{P\times P}$. For instance, we could take,

$$\mathcal{K}(\mathbf{K}) = \mathcal{W}(\mathbf{K}, N) \qquad \text{or} \qquad \mathcal{K}(\mathbf{K}) = \mathcal{W}^{-1}(\mathbf{K}, \delta + (P+1)), \tag{35}$$

where N is a positive integer and δ is a positive real number. A kernel process is defined by consistency under marginalisation and row/column exchangeability. Consistency under marginalisation implies that if we define $\mathbf{K}^*$ and $\mathbf{G}^*$ as principle submatrices of $\mathbf{K}$ and $\mathbf{G}$, dropping the same rows and columns, then $\mathbf{G}$ being distributed according to a kernel process implies that $\mathbf{G}^*$ is distributed according to that same kernel process,

$$\mathbf{G} \sim \mathcal{K}(\mathbf{K}) \qquad \text{implies} \qquad \mathbf{G}^* \sim \mathcal{K}(\mathbf{K}^*). \tag{36}$$

row/column exchangeability means,

$$\mathbf{G} \sim \mathcal{K}(\mathbf{K}) \qquad \text{implies} \qquad \mathbf{G}_\sigma \sim \mathcal{K}(\mathbf{K}_\sigma) \tag{37}$$

where σ is a permutation of the rows/columns. Note that both the Wishart and inverse Wishart as defined in Eq. (35) are consistent under marginalisation and are row/column exchangeable.

A deep kernel process, $\mathcal{D}$, is the composition of two (or more) underlying kernel processes, $\mathcal{K}_1$ and $\mathcal{K}_2$,

$$\mathbf{G} \sim \mathcal{K}_1(\mathbf{K}), \qquad \mathbf{H} \sim \mathcal{K}_2(\mathbf{G}), \tag{38a}$$

$$\mathbf{H} \sim \mathcal{D}(\mathbf{K}). \tag{38b}$$

We define $\mathbf{K}^*$, $\mathbf{G}^*$ and $\mathbf{H}^*$ as principle submatrices of $\mathbf{K}$, $\mathbf{G}$ and $\mathbf{H}$ respectively, dropping the same rows and columns, and again, $\mathbf{K}_\sigma$, $\mathbf{G}_\sigma$ and $\mathbf{H}_\sigma$ are those matrices with the rows and columns permuted. To establish that $\mathcal{D}$ is consistent under marginalisation, we use the consistency under marginalisation of $\mathcal{K}_1$ and $\mathcal{K}_2$

$$\mathbf{G}^* \sim \mathcal{K}_1(\mathbf{K}^*), \qquad \mathbf{H}^* \sim \mathcal{K}_2(\mathbf{G}^*), \tag{39a}$$

and the definition of the $\mathcal{D}$ as the composition of $\mathcal{K}_1$ and $\mathcal{K}_2$ (Eq. 38)

$$\mathbf{H}^* \sim \mathcal{D}(\mathbf{K}^*). \tag{39b}$$

Likewise, to establish row/column exchangeability, we use row/column exchangeability of $\mathcal{K}_1$ and $\mathcal{K}_2$,

$$\mathbf{G}_\sigma \sim \mathcal{K}_1(\mathbf{K}_\sigma), \qquad \mathbf{H}_\sigma \sim \mathcal{K}_2(\mathbf{G}_\sigma), \tag{40a}$$

and the definition of the $\mathcal{D}$ as the composition of $\mathcal{K}_1$ and $\mathcal{K}_2$ (Eq. 38)

$$\mathbf{H}_\sigma \sim \mathcal{D}(\mathbf{K}_\sigma). \tag{40b}$$

The deep kernel process $\mathcal{D}$ is thus consistent under marginalisation and has exchangeable rows/columns, and hence a deep kernel process is indeed itself a kernel process.

Further, note that we can consider $\mathcal{K}$ to be a deterministic distribution that gives mass to only a single $\mathbf{G}$. In that case, $\mathcal{K}$ can be thought of as a deterministic function which must satisfy a corresponding consistency property,

$$\mathbf{G} = \mathcal{K}(\mathbf{K}), \qquad \mathbf{G}^* = \mathcal{K}(\mathbf{K}^*), \qquad \mathbf{G}_\sigma = \mathcal{K}(\mathbf{K}_\sigma), \tag{41}$$

and this is indeed satisfied by all deterministic transformations of kernels considered here. In practical terms, as long as $\mathbf{G}$ is always a valid kernel, it is sufficient for the elements of $G_{i\neq j}$ to depend only on K_{ij}, K_{ii} and K_{jj} and for G_{ii} to depend only on K_{jj}, which is satisfied by e.g. the squared exponential kernel (Eq. 17) and by the ReLU kernel (Cho & Saul, 2009).

B. The first layer of our deep GP as Bayesian inference over a generalised lengthscale

In our deep GP architecture, we first sample $\mathbf{F}_1 \in \mathbb{R}^{P\times N_1}$ from a Gaussian with covariance $\mathbf{K}_0 = \frac{1}{N_0}\mathbf{X}\mathbf{X}^T$ (Eq. 5a). This might seem odd, as the usual deep GP involves passing the input, $\mathbf{X} \in \mathbb{R}^{P\times N_0}$, directly to the kernel function. However, in the standard deep GP framework, the kernel (e.g. a squared exponential kernel) has lengthscale hyperparameters which can be inferred using Bayesian inference. In particular,

$$k_{\text{param}}(\tfrac{1}{\sqrt{N_0}}\mathbf{x}_i, \tfrac{1}{\sqrt{N_0}}\mathbf{x}_j) = \exp\left(-\tfrac{1}{2N_0}\left(\mathbf{x}_i - \mathbf{x}_j\right)\boldsymbol{\Omega}\left(\mathbf{x}_i - \mathbf{x}_j\right)^T\right). \tag{42}$$

where k_{param} is a new squared exponential kernel that explicitly includes hyperparmeters $\boldsymbol{\Omega} \in \mathbb{R}^{N_0\times N_0}$, and where $\mathbf{x}_i$ is the ith row of $\mathbf{X}$. Typically, in deep GPs, the parameter, $\boldsymbol{\Omega}$, is diagonal, and the diagonal elements correspond to the inverse square of the lengthscale, l_i, (i.e. $\Omega_{ii} = 1/l_i^2$). However, in many cases it may be useful to have a non-diagonal scaling. For instance, we could use,

$$\boldsymbol{\Omega} \sim \mathcal{W}\left(\tfrac{1}{N_1}\mathbf{I}, N_1\right), \tag{43}$$

which corresponds to,

$$\boldsymbol{\Omega} = \mathbf{W}\mathbf{W}^T, \qquad \text{where} \qquad W_{i\lambda} \sim \mathcal{N}\left(0, \tfrac{1}{N_1}\right),\ \mathbf{W} \in \mathbb{R}^{N_0\times N_1}. \tag{44}$$

Under our approach, we sample $\mathbf{F} = \mathbf{F}_1$ from Eq. (5b), so $\mathbf{F}$ can be written as,

$$\mathbf{F} = \mathbf{X}\mathbf{W}, \qquad \mathbf{f}_i = \mathbf{x}_i\mathbf{W}, \tag{45}$$

where $\mathbf{f}_i$ is the ith row of $\mathbf{F}$. Putting this into a squared exponential kernel without a lengthscale parameter,

$$\begin{aligned}
k(\tfrac{1}{\sqrt{N_0}}\mathbf{f}_i, \tfrac{1}{\sqrt{N_0}}\mathbf{f}_j) &= \exp\left(-\tfrac{1}{2N_0}\left(\mathbf{f}_i - \mathbf{f}_j\right)\left(\mathbf{f}_i - \mathbf{f}_j\right)^T\right), \\
&= \exp\left(-\tfrac{1}{2N_0}\left(\mathbf{x}_i\mathbf{W} - \mathbf{x}_j\mathbf{W}\right)\left(\mathbf{x}_i\mathbf{W} - \mathbf{x}_j\mathbf{W}\right)^T\right), \\
&= \exp\left(-\tfrac{1}{2N_0}\left(\mathbf{x}_i - \mathbf{x}_j\right)\mathbf{W}\mathbf{W}^T\left(\mathbf{x}_i - \mathbf{x}_j\right)^T\right), \\
&= \exp\left(-\tfrac{1}{2N_0}\left(\mathbf{x}_i - \mathbf{x}_j\right)\boldsymbol{\Omega}\left(\mathbf{x}_i - \mathbf{x}_j\right)^T\right), \\
&= k_{\text{param}}(\tfrac{1}{\sqrt{N_0}}\mathbf{x}_i, \tfrac{1}{\sqrt{N_0}}\mathbf{x}_j).
\end{aligned} \tag{46}$$

We find that a parameter-free squared exponential kernel applied to $\mathbf{F}$ is equivalent to a squared-exponential kernel with generalised lengthscale hyperparameters applied to the input.

C. BNNs as deep kernel processes

Here we show that standard, finite BNNs, infinite BNNs and infinite BNNs with bottlenecks can be understood as deep kernel processes.

C.1. Stanard finite BNNs (and general DGPs)

Standard, finite BNNs are deep kernel processes, albeit ones which do not admit an analytic expression for the probability density. In particular, the prior for a standard Bayesian neural network (Fig. 3 top) is,

$$\mathrm{P}\left(\mathbf{W}_\ell\right) = \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{w}_\lambda^\ell; \mathbf{0}, \mathbf{I}/N_{\ell-1}\right), \qquad \mathbf{W}_\ell \in \mathbb{R}^{N_{\ell-1}\times N_\ell}, \tag{47a}$$

$$\mathbf{F}_\ell = \begin{cases} \mathbf{X}\mathbf{W}_1 & \text{for } \ell = 1, \\ \phi\left(\mathbf{F}_{\ell-1}\right)\mathbf{W}_\ell & \text{otherwise,} \end{cases} \qquad \mathbf{F}_\ell \in \mathbb{R}^{P\times N_\ell}, \tag{47b}$$

where $\mathbf{w}_\lambda^\ell$ is the λth column of $\mathbf{W}_\ell$. In the neural-network case, ϕ is a pointwise nonlinearity such as a ReLU. Integrating out the weights, the features, $\mathbf{F}_\ell$, become Gaussian distributed, as they depend linearly on the Gaussian distributed weights, $\mathbf{W}_\ell$,

$$\mathrm{P}\left(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}\right) = \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}_\lambda^\ell; \mathbf{0}, \mathbf{K}_\ell\right) = \mathrm{P}\left(\mathbf{F}_\ell|\mathbf{K}_\ell\right), \tag{48}$$

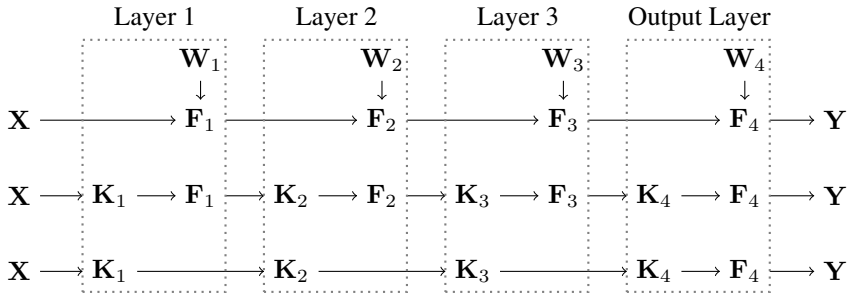

Figure 3. A series of generative models for a standard, finite BNN. **Top**. The standard model, with features, $\mathbf{F}_\ell$, and weights $\mathbf{W}_\ell$ (Eq. 47). **Middle**. Integrating out the weights, the distribution over features becomes Gaussian (Eq. 50), and we explicitly introduce the kernel, $\mathbf{K}_\ell$, as a latent variable. **Bottom**. Integrating out the activations, $\mathbf{F}_\ell$, gives a deep kernel process, albeit one where the distributions $\mathrm{P}\left(\mathbf{K}_\ell|\mathbf{K}_{\ell-1}\right)$ cannot be written down analytically, but where the expectation, $\mathbb{E}\left[\mathbf{K}_\ell|\mathbf{K}_{\ell-1}\right]$ is known (Eq. 51).

where

$$\mathbf{K}_\ell = \tfrac{1}{N_{\ell-1}}\phi(\mathbf{F}_{\ell-1})\phi^T(\mathbf{F}_{\ell-1}). \tag{49}$$

Crucially, $\mathbf{F}_\ell$ depends on the previous layer activities, $\mathbf{F}_{\ell-1}$ only through the kernel, $\mathbf{K}_\ell$. As such, we could write a generative model as (Fig. 3 middle),

$$\mathbf{K}_\ell = \begin{cases} \frac{1}{N_0}\mathbf{X}\mathbf{X}^T & \text{for } \ell = 1, \\ \frac{1}{N_{\ell-1}}\phi(\mathbf{F}_{\ell-1})\phi^T(\mathbf{F}_{\ell-1}) & \text{otherwise,} \end{cases} \tag{50a}$$

$$\mathrm{P}\left(\mathbf{F}_\ell|\mathbf{K}_\ell\right) = \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}_\lambda^\ell; \mathbf{0}, \mathbf{K}_\ell\right), \tag{50b}$$

where we have explicitly included the kernel, $\mathbf{K}_\ell$, as a latent variable. This form highlights that BNNs *are* deep GPs, in the sense that $\mathbf{F}_\lambda^\ell$ are Gaussian, with a kernel that depends on the activations from the previous layer. Indeed note that *any* deep GP (i.e. including those with kernels that cannot be written as a function of the Gram matrix) as a kernel, $\mathbf{K}_\ell$, is by definition a matrix that can be written as the outer product of a potentially infinite number of features, $\phi(\mathbf{F}_\ell)$ where we allow ϕ to be a much richer class of functions than the usual pointwise nonlinearities (Hofmann et al., 2008). We might now try to follow the approach we took above for deep GPs, and consider a Wishart-distributed Gram matrix, $\mathbf{G}_\ell = \frac{1}{N_\ell}\mathbf{F}_\ell\mathbf{F}_\ell^T$. However, for BNNs we encounter an issue: we are not able to compute the kernel, $\mathbf{K}_\ell$ just using the Gram matrix, $\mathbf{G}_\ell$: we need the full set of features, $\mathbf{F}_\ell$.

Instead, we need an alternative approach to show that a neural network is a deep kernel process. In particular, after integrating out the weights, the resulting distribution is chain-structured (Fig. 3 middle), so in principle we can integrate out $\mathbf{F}_\ell$ to obtain a distribution over $\mathbf{K}_\ell$ conditioned on $\mathbf{K}_{\ell-1}$, giving the DKP model in Fig. 3 (bottom),

$$\mathrm{P}\left(\mathbf{K}_\ell|\mathbf{K}_{\ell-1}\right) = \int \mathbf{d}\mathbf{F}_{\ell-1}\ \delta_{\mathrm{D}}\left(\mathbf{K}_\ell - \tfrac{1}{N_\ell}\phi(\mathbf{F}_{\ell-1})\phi^T(\mathbf{F}_{\ell-1})\right)\mathrm{P}\left(\mathbf{F}_{\ell-1}|\mathbf{K}_{\ell-1}\right), \tag{51}$$

where $\mathrm{P}\left(\mathbf{F}_{\ell-1}|\mathbf{K}_{\ell-1}\right)$ is given by Eq. (50b) and δ_{D} is the Dirac-delta function consisting of a point-mass at zero. Using this integral to write out the generative process only in terms of $\mathbf{K}_\ell$ gives the deep kernel process in Fig. 3 (bottom). While this distribution exists in principle, it cannot be evaluated analytically. But we can explicitly evaluate the expected value of $\mathbf{K}_\ell$ given $\mathbf{K}_{\ell-1}$ using results from Cho & Saul (2009). In particular, we take Eq. 50a, write out the matrix-multiplication explicitly as a series of vector outer products, and note that as $\mathbf{f}_\lambda^\ell$ is IID across ℓ, the empirical average is equal to the expectation of a single term, which is computed by Cho & Saul (2009),

$$\begin{aligned}\mathbb{E}\left[\mathbf{K}_{\ell+1}|\mathbf{K}_\ell\right] &= \tfrac{1}{N_\ell}\textstyle\sum_{\lambda=1}^{N_\ell}\mathbb{E}\left[\phi(\mathbf{f}_\lambda^\ell)\phi^T(\mathbf{f}_\lambda^\ell)|\mathbf{K}_\ell\right] = \mathbb{E}\left[\phi(\mathbf{f}_\lambda^\ell)\phi^T(\mathbf{f}_\lambda^\ell)|\mathbf{K}_\ell\right], \\ &= \int \mathbf{d}\mathbf{f}_\lambda^\ell\ \mathcal{N}\left(\mathbf{f}_\lambda^\ell; \mathbf{0}, \mathbf{K}_\ell\right)\phi(\mathbf{f}_\lambda^\ell)\phi^T(\mathbf{f}_\lambda^\ell) \equiv \mathbf{K}(\mathbf{K}_\ell).\end{aligned} \tag{52}$$

Finally, we define this expectation to be $\mathbf{K}(\mathbf{K}_\ell)$ in the case of NNs.

C.2. Infinite NNs

We have found that for standard finite neural networks, we were not able to compute the distribution over $\mathbf{K}_\ell$ conditioned on $\mathbf{K}_{\ell-1}$ (Eq. (51)). To resolve this issue, one approach is to consider the limit of an infinitely wide neural network. In this limit, the $\mathbf{K}_\ell$ becomes a deterministic function of $\mathbf{K}_{\ell-1}$, as $\mathbf{K}_\ell$ can be written as the average of N_ℓ IID outer products, and as N_ℓ grows to infinity, the law of large numbers tells us that the average becomes equal to its expectation,

$$\lim_{N_\ell\to\infty}\mathbf{K}_{\ell+1}=\lim_{N_\ell\to\infty}\tfrac{1}{N_\ell}\textstyle\sum_{\lambda=1}^{N_\ell}\phi(\mathbf{f}_\lambda^\ell)\phi^T(\mathbf{f}_\lambda^\ell)=\mathbb{E}\left[\phi(\mathbf{f}_\lambda^\ell)\phi^T(\mathbf{f}_\lambda^\ell)|\mathbf{K}_\ell\right]=\mathbf{K}(\mathbf{K}_\ell). \tag{53}$$

C.3. Infinite NNs with bottlenecks

In infinite NNs, the kernel is deterministic, meaning that there is no flexibility/variability, and hence no capability for representation learning (Aitchison, 2019). Here, we consider infinite networks with bottlenecks that combine the tractability of infinite networks with the flexibility of finite networks (Aitchison, 2019). The trick is to separate flexible, finite linear "bottlenecks" from infinite-width nonlinearities. We keep the nonlinearity infinite in order to ensure that the output kernel is deterministic and can be computed using results from Cho & Saul (2009). In particular, we use finite-width $\mathbf{F}_\ell\in\mathbb{R}^{P\times N_\ell}$ and infinite width $\mathbf{F}'_\ell\in\mathbb{R}^{P\times M_\ell}$, (we send M_ℓ to infinity while leaving N_ℓ finite),

$$\mathrm{P}\left(\mathbf{W}_\ell\right)=\textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{w}_\lambda^\ell;\mathbf{0},\mathbf{I}/M_{\ell-1}\right)\quad M_0=N_0, \tag{54a}$$

$$\mathbf{F}_\ell=\begin{cases}\mathbf{X}\mathbf{W}_\ell & \text{if } \ell=1,\\ \phi(\mathbf{F}'_{\ell-1})\mathbf{W}_\ell & \text{otherwise,}\end{cases} \tag{54b}$$

$$\mathrm{P}\left(\mathbf{M}_\ell\right)=\textstyle\prod_{\lambda=1}^{M_\ell}\mathcal{N}\left(\mathbf{m}_\lambda^\ell;\mathbf{0},\mathbf{I}/N_\ell\right), \tag{54c}$$

$$\mathbf{F}'_\ell=\mathbf{F}_\ell\mathbf{M}_\ell. \tag{54d}$$

This generative process is given graphically in Fig. 4 (top).

Integrating over the expansion weights, $\mathbf{M}_\ell\in\mathbb{R}^{N_\ell\times M_\ell}$, and the bottleneck weights, $\mathbf{W}_\ell\in\mathbb{R}^{M_{\ell-1}\times N_\ell}$, the generative model (Fig. 4 second row) can be rewritten,

$$\mathbf{K}_\ell=\begin{cases}\frac{1}{N_0}\mathbf{X}\mathbf{X}^T & \text{for } \ell=1,\\ \frac{1}{M_{\ell-1}}\phi\left(\mathbf{F}'_{\ell-1}\right)\phi^T\left(\mathbf{F}'_{\ell-1}\right) & \text{otherwise,}\end{cases} \tag{55a}$$

$$\mathrm{P}\left(\mathbf{F}_\ell|\mathbf{K}_\ell\right)=\textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}_\lambda^\ell;\mathbf{0},\mathbf{K}_\ell\right), \tag{55b}$$

$$\mathbf{G}_\ell=\tfrac{1}{N_\ell}\mathbf{F}_\ell\mathbf{F}_\ell^T, \tag{55c}$$

$$\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{G}_\ell\right)=\textstyle\prod_{\lambda=1}^{M_\ell}\mathcal{N}\left(\mathbf{f}'^\ell_\lambda;\mathbf{0},\mathbf{G}_\ell\right). \tag{55d}$$

Remembering that $\mathbf{K}_{\ell+1}$ is the empirical mean of M_ℓ IID terms, as $M_\ell\to\infty$ it converges on its expectation

$$\lim_{M_\ell\to\infty}\mathbf{K}_{\ell+1}=\lim_{M_\ell\to\infty}\tfrac{1}{M_\ell}\textstyle\sum_{\lambda=1}^{N_\ell}\phi\left(\mathbf{f}'^\ell_\lambda\right)\phi^T\left(\mathbf{f}'^\ell_\lambda\right)=\mathbb{E}\left[\phi(\mathbf{f}'^\ell_\lambda)\phi^T(\mathbf{f}'^\ell_\lambda)|\mathbf{G}_\ell\right]=\mathbf{K}(\mathbf{G}_\ell). \tag{56}$$

and we define the limit to be $\mathbf{K}(\mathbf{G}_\ell)$. Note if we use standard (e.g. ReLU) nonlinearities, we can use results from Cho & Saul (2009) to compute $\mathbf{K}(\mathbf{G}_\ell)$. Thus, we get the following generative process,

$$\mathbf{K}_\ell=\begin{cases}\frac{1}{N_0}\mathbf{X}\mathbf{X}^T & \text{for } \ell=1,\\ \mathbf{K}(\mathbf{G}_{\ell-1}) & \text{otherwise,}\end{cases} \tag{57a}$$

$$\mathrm{P}\left(\mathbf{G}_\ell\right)=\mathcal{W}\left(\mathbf{G}_\ell;\tfrac{1}{N_\ell}\mathbf{K}_\ell,N_\ell\right). \tag{57b}$$

Finally, eliminating the deterministic kernels, $\mathbf{K}_\ell$, from the model, we obtain exactly the deep GP generative model in Eq. 8 (Fig. C.3 fourth row).

D. Standard approximate posteriors over features and weights fail to capture symmetries

We have shown that it is possible to represent DGPs and a variety of NNs as deep kernel processes. Here, we argue that standard deep GP approximate posteriors are seriously flawed, and that working with deep kernel processes may alleviate these flaws.

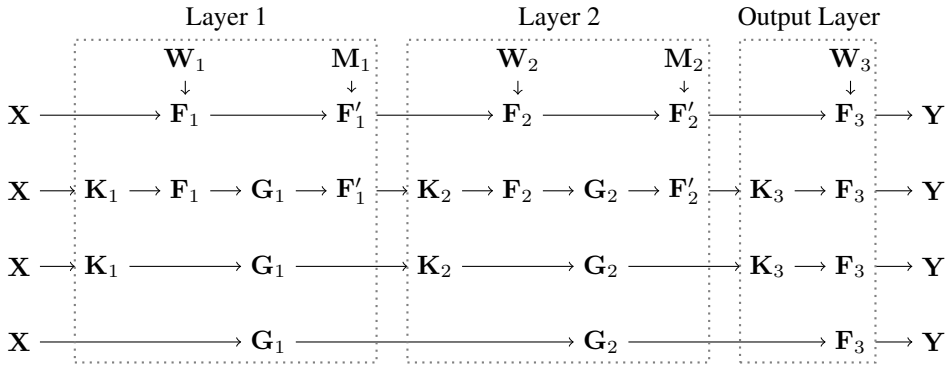

Figure 4. A series of generative models for an infinite network with bottlenecks. **First row**. The standard model. **Second row**. Integrating out the weights. **Third row**. Integrating out the features, the Gram matrices are Wishart-distributed, and the kernels are deterministic. **Last row**. Eliminating all deterministic random variables, we get a model equivalent to that for DGPs (Fig. 1 bottom).

In particular, we show that the true DGP posterior has rotational symmetries and that the true BNN posterior has permutation symmetries that are not captured by standard variational posteriors.

D.1. Permutation symmetries in DNNs posteriors over weights

Permutation symmetries in neural network posteriors were known in classical work on Bayesian neural networks (e.g. MacKay, 1992). Here, we spell out the argument in full. Taking $\mathbf{P}$ to be a permutation matrix (i.e. a unitary matrix with $\mathbf{P}\mathbf{P}^T = \mathbf{I}$ with one 1 in every row and column), we have,

$$\phi(\mathbf{F})\mathbf{P} = \phi(\mathbf{F}\mathbf{P}). \tag{58}$$

i.e. permuting the input to a nonlinearity is equivalent to permuting its output. Expanding two steps of the recursion defined by Eq. (47b),

$$\mathbf{F}_\ell = \phi(\phi(\mathbf{F}_{\ell-2})\mathbf{W}_{\ell-1})\mathbf{W}_\ell, \tag{59}$$

multiplying by the identity,

$$\mathbf{F}_\ell = \phi(\phi(\mathbf{F}_{\ell-2})\mathbf{W}_{\ell-1})\mathbf{P}\mathbf{P}^T\mathbf{W}_\ell, \tag{60}$$

where $\mathbf{P} \in \mathbb{R}^{N_{\ell-1} \times N_{\ell-1}}$, applying Eq. (58)

$$\mathbf{F}_\ell = \phi(\phi(\mathbf{F}_{\ell-2})\mathbf{W}_{\ell-1}\mathbf{P})\mathbf{P}^T\mathbf{W}_\ell, \tag{61}$$

defining permuted weights,

$$\mathbf{W}'_{\ell-1} = \mathbf{W}_{\ell-1}\mathbf{P}, \qquad \mathbf{W}'_\ell = \mathbf{P}^T\mathbf{W}_\ell, \tag{62}$$

the output is the same under the original or permuted weights,

$$\mathbf{F}_\ell = \phi(\phi(\mathbf{F}_{\ell-2})\mathbf{W}'_{\ell-1})\mathbf{W}'_\ell = \phi(\phi(\mathbf{F}_{\ell-2})\mathbf{W}_{\ell-1})\mathbf{W}_\ell. \tag{63}$$

Introducing a different perturbation between every pair of layers we get a more general symmetry,

$$\mathbf{W}'_1 = \mathbf{W}_1\mathbf{P}_1, \tag{64a}$$
$$\mathbf{W}_\ell = \mathbf{P}_{\ell-1}^T\mathbf{W}_\ell\mathbf{P}_\ell \quad \text{for } \ell \in \{2, \dots, L\}, \tag{64b}$$
$$\mathbf{W}'_{L+1} = \mathbf{P}_L\mathbf{W}_{L+1}, \tag{64c}$$

where $\mathbf{P}_\ell \in \mathbb{R}^{N_{\ell-1} \times N_{\ell-1}}$. As the output of the neural network is the same under any of these permutations the likelihoods for original and permuted weights are equal,

$$\mathrm{P}\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}_1, \dots, \mathbf{W}_{L+1}\right) = \mathrm{P}\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}'_1, \dots, \mathbf{W}'_{L+1}\right), \tag{65}$$

and as the prior over elements within a weight matrix is IID Gaussian (Eq. 47a), the prior probability density is equal under original and permuted weights,

$$\mathrm{P}\left(\mathbf{W}_1, \ldots, \mathbf{W}_{L+1}\right) = \mathrm{P}\left(\mathbf{W}'_1, \ldots, \mathbf{W}'_{L+1}\right). \tag{66}$$

Thus, the joint probability is invariant to permutations,

$$\mathrm{P}\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}_1, \ldots, \mathbf{W}_{L+1}\right)\mathrm{P}\left(\mathbf{W}_1, \ldots, \mathbf{W}_{L+1}\right) = \mathrm{P}\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}'_1, \ldots, \mathbf{W}'_{L+1}\right)\mathrm{P}\left(\mathbf{W}'_1, \ldots, \mathbf{W}'_{L+1}\right), \tag{67}$$

and applying Bayes theorem, the posterior is invariant to permutations,

$$\mathrm{P}\left(\mathbf{W}_1, \ldots, \mathbf{W}_{L+1}|\mathbf{Y}, \mathbf{X}\right) = \mathrm{P}\left(\mathbf{W}'_1, \ldots, \mathbf{W}'_{L+1}|\mathbf{Y}, \mathbf{X}\right). \tag{68}$$

Due in part to these permutation symmetries, the posterior distribution over weights is extremely complex and multimodal. Importantly, it is not possible to capture these symmetries using standard variational posteriors over weights, such as factorised posteriors, but it is not necessary to capture these symmetries if we work with Gram matrices and kernels, which are invariant to permutations (and other unitary transformations; Eq. 14).

D.2. Rotational symmetries in deep GP posteriors

To show that deep GP posteriors are invariant to unitary transformations, $\mathbf{U}_\ell \in \mathbb{R}^{N_\ell \times N_\ell}$, where $\mathbf{U}_\ell\mathbf{U}_\ell^T = \mathbf{I}$, we define transformed features, $\mathbf{F}'_\ell$,

$$\mathbf{F}'_\ell = \mathbf{F}_\ell\mathbf{U}_\ell. \tag{69}$$

To evaluate $\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}'_{\ell-1}\right)$, we begin by substituting for $\mathbf{F}'_{\ell-1}$,

$$\begin{aligned}
\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}'_{\ell-1}\right) &= \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}'^{\ell}_\lambda; \mathbf{0}, \mathbf{K}\left(\frac{1}{N_{\ell-1}}\mathbf{F}'_{\ell-1}\mathbf{F}'^{T}_{\ell-1}\right)\right), && (70)\\
&= \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}'^{\ell}_\lambda; \mathbf{0}, \mathbf{K}\left(\frac{1}{N_{\ell-1}}\mathbf{F}_{\ell-1}\mathbf{U}_{\ell-1}\mathbf{U}^T_{\ell-1}\mathbf{F}^T_{\ell-1}\right)\right), && (71)\\
&= \textstyle\prod_{\lambda=1}^{N_\ell}\mathcal{N}\left(\mathbf{f}'^{\ell}_\lambda; \mathbf{0}, \mathbf{K}\left(\frac{1}{N_{\ell-1}}\mathbf{F}_{\ell-1}\mathbf{F}^T_{\ell-1}\right)\right), && (72)\\
&= \mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}_{\ell-1}\right). && (73)
\end{aligned}$$

To evaluate $\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}_{\ell-1}\right)$, we substitute for $\mathbf{F}'_\ell$ in the explicit form for the multivariate Gaussian probability density,

$$\begin{aligned}
\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}_{\ell-1}\right) &= -\tfrac{1}{2}\,\mathrm{Tr}\left(\mathbf{F}'^{T}_\ell\mathbf{K}^{-1}_{\ell-1}\mathbf{F}'_\ell\right) + \mathrm{const}, && (74)\\
&= -\tfrac{1}{2}\,\mathrm{Tr}\left(\mathbf{K}^{-1}_{\ell-1}\mathbf{F}'_\ell\mathbf{F}'^{T}_\ell\right) + \mathrm{const}, && (75)\\
&= -\tfrac{1}{2}\,\mathrm{Tr}\left(\mathbf{K}^{-1}_{\ell-1}\mathbf{F}_\ell\mathbf{U}_\ell\mathbf{U}^T_\ell\mathbf{F}^T_\ell\right) + \mathrm{const}, && (76)\\
&= -\tfrac{1}{2}\,\mathrm{Tr}\left(\mathbf{K}^{-1}_{\ell-1}\mathbf{F}_\ell\mathbf{F}^T_\ell\right) + \mathrm{const}, && (77)\\
&= \mathrm{P}\left(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}\right). && (78)
\end{aligned}$$

where $\mathbf{K}_{\ell-1} = \mathbf{K}\left(\frac{1}{N_{\ell-1}}\mathbf{F}_{\ell-1}\mathbf{F}^T_{\ell-1}\right)$, and the constant depends only on $\mathbf{F}_{\ell-1}$. Combining these derivations, each of these conditionals is invariant to rotations of $\mathbf{F}_\ell$ and $\mathbf{F}_{\ell-1}$,

$$\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}'_{\ell-1}\right) = \mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}_{\ell-1}\right) = \mathrm{P}\left(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}\right). \tag{79}$$

The same argument can straightforwardly be extended to the inputs, $\mathrm{P}\left(\mathbf{F}_1|\mathbf{X}\right)$,

$$\mathrm{P}\left(\mathbf{F}'_1|\mathbf{X}\right) = \mathrm{P}\left(\mathbf{F}_1|\mathbf{X}\right), \tag{80}$$

and to the final probability density, for output activations, $\mathbf{F}_{L+1}$ which is not invariant to permutations,

$$\mathrm{P}\left(\mathbf{F}_{L+1}|\mathbf{F}'_L\right) = \mathrm{P}\left(\mathbf{F}_{L+1}|\mathbf{F}'_L\right), \tag{81}$$

Therefore, we have,

$$\mathrm{P}\left(\mathbf{F}'_1,\ldots,\mathbf{F}'_L,\mathbf{F}_{L+1},\mathbf{Y}|\mathbf{X}\right)=\mathrm{P}\left(\mathbf{Y}|\mathbf{F}_{L+1}\right)\mathrm{P}\left(\mathbf{F}_{L+1}|\mathbf{F}'_L\right)\left(\prod_{\ell=2}^{L}\mathrm{P}\left(\mathbf{F}'_\ell|\mathbf{F}'_{\ell-1}\right)\right)\mathrm{P}\left(\mathbf{F}'_1|\mathbf{X}\right), \tag{82}$$

$$=\mathrm{P}\left(\mathbf{Y}|\mathbf{F}_{L+1}\right)\mathrm{P}\left(\mathbf{F}_{L+1}|\mathbf{F}_L\right)\left(\prod_{\ell=2}^{L}\mathrm{P}\left(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}\right)\right)\mathrm{P}\left(\mathbf{F}_1|\mathbf{X}\right), \tag{83}$$

$$=\mathrm{P}\left(\mathbf{F}_1,\ldots,\mathbf{F}_L,\mathbf{F}_{L+1},\mathbf{Y}|\mathbf{X}\right). \tag{84}$$

Therefore, applying Bayes theorem the posterior is invariant to rotations,

$$\mathrm{P}\left(\mathbf{F}'_1,\ldots,\mathbf{F}'_L,\mathbf{F}_{L+1}|\mathbf{X},\mathbf{Y}\right)=\mathrm{P}\left(\mathbf{F}_1,\ldots,\mathbf{F}_L,\mathbf{F}_{L+1}|\mathbf{X},\mathbf{Y}\right). \tag{85}$$

Importantly, these posterior symmetries are not captured by standard variational posteriors with non-zero means (e.g. Salimbeni & Deisenroth, 2017).

D.3. The true posterior over features in a DGP has zero mean

We can use symmetry to show that the posterior of $\mathbf{F}_\ell$ has zero mean. We begin by writing the expectation as an integral,

$$\mathbb{E}\left[\mathbf{F}_\ell|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right]=\int\mathbf{dF}\ \mathbf{F}\,\mathrm{P}\left(\mathbf{F}_\ell{=}\mathbf{F}|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right). \tag{86}$$

Changing variables in the integral to $\mathbf{F}'=-\mathbf{F}$, and noting that the absolute value of the Jacobian is 1, we have

$$=\int\mathbf{dF}'\ \left(-\mathbf{F}'\right)\mathrm{P}\left(\mathbf{F}_\ell{=}\left(-\mathbf{F}'\right)|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right), \tag{87}$$

using the symmetry of the posterior,

$$=\int\mathbf{dF}'\ \left(-\mathbf{F}'\right)\mathrm{P}\left(\mathbf{F}_\ell{=}\mathbf{F}'|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right), \tag{88}$$

$$=-\,\mathbb{E}\left[\mathbf{F}_\ell|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right], \tag{89}$$

the expectation is equal to minus itself, so it must be zero

$$\mathbb{E}\left[\mathbf{F}_\ell|\mathbf{F}_{\ell-1},\mathbf{F}_{\ell+1}\right]=\mathbf{0}. \tag{90}$$

E. Difficulties with VI in deep Wishart processes

The deep Wishart generative process is well-defined as long as we admit nonsingular Wishart distributions (Uhlig, 1994; Srivastava et al., 2003). The issue comes when we try to form a variational approximate posterior over low-rank positive definite matrices. This is typically the case because the number of datapoints, P is usually far larger than the number of features. In particular, the only convenient distribution over low-rank positive semidefinite matrices is the Wishart itself,

$$\mathrm{Q}\left(\mathbf{G}_\ell\right)=\mathcal{W}\left(\mathbf{G}_\ell;\tfrac{1}{N_\ell}\mathbf{\Psi},N_\ell\right). \tag{91}$$

However, a key feature of most variational approximate posteriors is the ability to increase and decrease the variance, independent of other properties such as the mean, and in our case the rank of the matrix. For a Wishart, the mean and variance are given by,

$$\mathop{\mathbb{E}}_{\mathrm{Q}(\mathbf{G}_\ell)}\left[\mathbf{G}_\ell\right]=\mathbf{\Psi}, \tag{92}$$

$$\mathop{\mathbb{V}}_{\mathrm{Q}(\mathbf{G}_\ell)}\left[G^\ell_{ij}\right]=\tfrac{1}{N_\ell}\left(\Psi^2_{ij}+\Psi_{ii}\Psi_{jj}\right). \tag{93}$$

Initially, this may look fine: we can increase or decrease the variance by changing N_ℓ. However, remember that N_ℓ is the degrees of freedom, which controls the rank of the matrix, $\mathbf{G}_\ell$. As such, N_ℓ is fixed by the prior: the prior and approximate

posterior must define distributions over matrices of the same rank. And once N_ℓ is fixed, we no longer have independent control over the variance.

To go about resolving this issue, we need to find a distribution over low-rank matrices with independent control of the mean and variance. The natural approach is to use a non-central Wishart, defined as the outer product of Gaussian-distributed vectors with non-zero means. While this distribution is easy to sample from and does give independent control over the rank, mean and variance, its probability density is prohibitively costly and complex to evaluate (Koev & Edelman, 2006).

F. Singular (inverse) Wishart processes at the input layer

In almost all cases of interest, our the kernel functions $\mathbf{K}(\mathbf{G})$ return full-rank matrices, so we can use standard (inverse) Wishart distributions, which assume that the input matrix is full-rank. However, this is not true at the input layer as $\mathbf{K}_0 = \frac{1}{N_0}\mathbf{X}\mathbf{X}^T$ will often be low-rank. This requires us to use singular (inverse) Wishart distributions which in general are difficult to work with (Uhlig, 1994; Srivastava et al., 2003; Bodnar & Okhrin, 2008; Bodnar et al., 2016). As such, instead we exploit knowledge of the input features to work with a smaller, full-rank matrix, $\mathbf{\Omega} \in \mathbb{R}^{N_0 \times N_0}$, where, remember, N_0 is the number of input features in $\mathbf{X}$. For a deep Wishart process,

$$\tfrac{1}{N_0}\mathbf{X}\mathbf{\Omega}\mathbf{X}^T = \mathbf{G}_1 \sim \mathcal{W}\left(\tfrac{1}{N_1}\mathbf{K}_0, N_1\right), \qquad \text{where} \qquad \mathbf{\Omega} \sim \mathcal{W}\left(\tfrac{1}{N_1}\mathbf{I}, N_1\right), \tag{94}$$

and for a deep inverse Wishart process,

$$\tfrac{1}{N_0}\mathbf{X}\mathbf{\Omega}\mathbf{X}^T = \mathbf{G}_1 \sim \mathcal{W}^{-1}(\delta_1\mathbf{K}_0, \delta_1 + P + 1), \qquad \text{where} \qquad \mathbf{\Omega} \sim \mathcal{W}^{-1}(\delta_1\mathbf{I}, \delta_1 + N_0 + 1). \tag{95}$$

Now, we are able to use the full-rank matrix, $\mathbf{\Omega}$ rather than the low-rank matrix, $\mathbf{G}_1$ as the random variable for variational inference. For the approximate posterior over $\mathbf{\Omega}$, in a deep inverse Wishart process, we use

$$\mathrm{Q}\left(\mathbf{\Omega}\right) = \mathcal{W}^{-1}\big(\delta_1\mathbf{I} + \mathbf{V}_1\mathbf{V}_1^T, \delta_1 + \gamma_1 + (N_0 + 1)\big). \tag{96}$$

Note in the usual case where there are fewer inducing points than input features, then the matrix $\mathbf{K}_0$ will be full-rank, and we can work with $\mathbf{G}_1$ as the random variable as usual.

G. Approximate posteriors over output features

To define approximate posteriors over inducing outputs, we are inspired by global inducing point methods (Ober & Aitchison, 2020). In particular, we take the approximate posterior to be the prior, multiplied by a "pseudo-likelihood",

$$\mathrm{Q}\left(\mathbf{F}_{L+1}|\mathbf{G}_L\right) \propto \mathrm{P}\left(\mathbf{F}_{L+1}|\mathbf{G}_L\right) \prod_{\lambda=1}^{N_{L+1}} \mathcal{N}\left(\mathbf{v}_\lambda; \mathbf{f}_\lambda^{L+1}, \mathbf{\Lambda}_\lambda^{-1}\right). \tag{97}$$

This is valid both for global inducing inputs and (for small datasets) training inputs, and the key thing to remember is that in either case, for any given input (e.g. an MNIST handwritten 2), there is a desired output (e.g. the class-label "2"), and the top-layer global inducing outputs, $\mathbf{v}_\lambda$, express these desired outcomes. Substituting for the prior,

$$\mathrm{Q}\left(\mathbf{F}_{L+1}|\mathbf{G}_L\right) \propto \prod_{\lambda=1}^{N_{L+1}} \mathcal{N}\left(\mathbf{f}_\lambda^{L+1}; \mathbf{0}, \mathbf{K}(\mathbf{G}_L)\right) \mathcal{N}\left(\mathbf{v}_\lambda; \mathbf{f}_\lambda^{L+1}, \mathbf{\Lambda}_\lambda^{-1}\right), \tag{98}$$

and computing this value gives the approximate posterior in the main text (Eq. 21).

H. Using eigenvalues to compare deep Wishart, deep residual Wishart and inverse Wishart priors

One might be concerned that the deep inverse Wishart processes in which we can easily perform inference are different to the deep Wishart processes corresponding to BNNs (Sec. C.1) and infinite NNs with bottlenecks (App. C.3). To address these concerns, we begin by noting that the (inverse) Wishart priors can be written in terms of samples from the standard (inverse) Wishart

$$\mathbf{G} = \mathbf{L}\mathbf{\Omega}\mathbf{L}^T, \qquad \mathbf{G}' = \mathbf{L}\mathbf{\Omega}'\mathbf{L}^T, \tag{99}$$

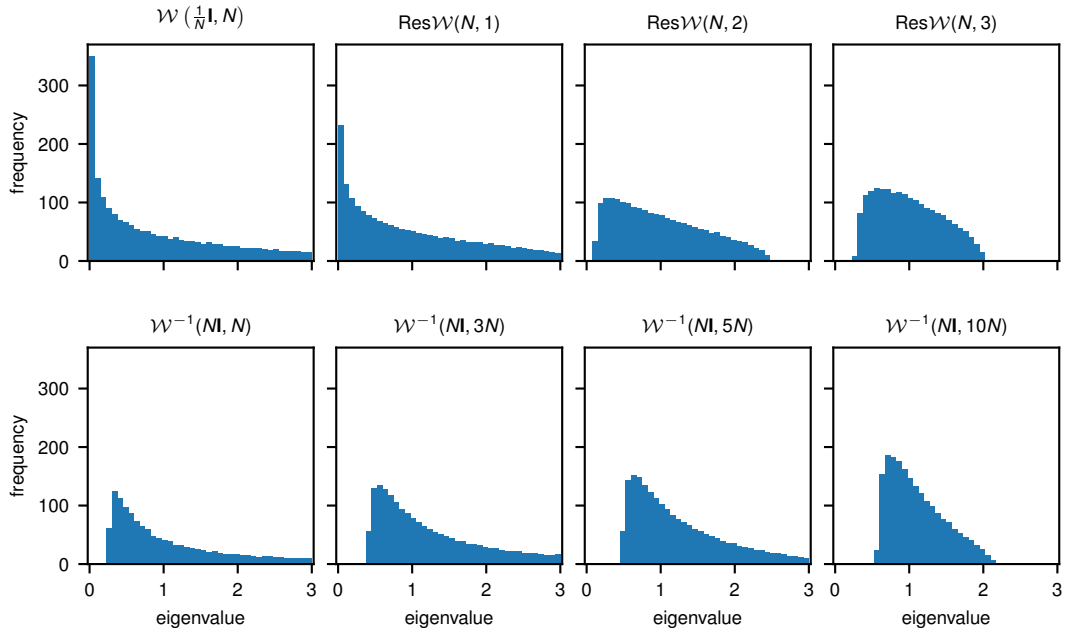

Figure 5. Eigenvalue histograms for a single sample from the labelled distribution, with $N = 2000$.

where $\mathbf{K} = \mathbf{L}\mathbf{L}^T$ such that,

$$\mathbf{\Omega} \sim \mathcal{W}\left(\tfrac{1}{N}\mathbf{I}, N\right), \qquad \mathbf{\Omega}' \sim \mathcal{W}^{-1}(N\mathbf{I}, \lambda N), \tag{100}$$

$$\mathbf{G} \sim \mathcal{W}\left(\tfrac{1}{N}\mathbf{K}, N\right), \qquad \mathbf{G}' \sim \mathcal{W}^{-1}(N\mathbf{K}, \lambda N). \tag{101}$$

Note that as the standard Wishart and inverse Wishart have uniform distributions over the eigenvectors (Shah et al., 2014), they differ only in the distribution over eigenvalues of $\mathbf{\Omega}$ and $\mathbf{\Omega}'$. We plotted the eigenvalue histogram for samples from a Wishart distribution with $N = P = 2000$ (Fig. 5 top left). This corresponds to an IID Gaussian prior over weights, with 2000 features in the input and output layers. Notably, there are many very small eigenvalues, which are undesirable as they eliminate information present in the input. To eliminate these very small eigenvalues, a common approach is to use a ResNet-inspired architecture (which is done even in the deep GP literature, e.g. Salimbeni & Deisenroth, 2017). To understand the eigenvalues in a residual layer, we define a Res$\mathcal{W}$ distribution by taking the outer product of a weight matrix with itself,

$$\mathbf{W}\mathbf{W}^T = \mathbf{\Omega}'' \sim \text{Res}\mathcal{W}(N, \alpha), \tag{102}$$

where the weight matrix is IID Gaussian, plus the identity matrix, with the identity matrix weighted as α,

$$\mathbf{W} = \tfrac{1}{\sqrt{1+\alpha^2}}\left(\sqrt{\tfrac{1}{N}}\boldsymbol{\xi} + \alpha\mathbf{I}\right), \qquad \xi_{i,\lambda} \sim \mathcal{N}(0, 1). \tag{103}$$

With $\alpha = 1$, there are still many very small eigenvalues, but these disappear as α increases. We compared these distributions to inverse Wishart distributions (Fig. 5 bottom) with varying degrees of freedom. For all degrees of freedom, we found that inverse Wishart distributions do not produce very small eigenvalues, which would eliminate information. As such, these eigenvalue distributions resemble those for Res$\mathcal{W}$ with α larger than 1.

I. Doubly stochastic variational inference in deep inverse Wishart processes

Due to the doubly stochastic results in Sec. 5.3, we only need to compute the conditional distribution over a single test/train point (we do not need the joint distribution over a number of test points). As such, we can decompose $\mathbf{G}$ and $\mathbf{\Psi}$ as,

$$\mathbf{G}_\ell = \begin{pmatrix} \mathbf{G}^\ell_{\text{ii}} & \mathbf{g}^{\ell T}_{\text{it}} \\ \mathbf{g}^\ell_{\text{it}} & g^\ell_{\text{tt}} \end{pmatrix}, \qquad \mathbf{\Psi} = \begin{pmatrix} \mathbf{\Psi}_{\text{ii}} & \boldsymbol{\psi}^T_{\text{it}} \\ \boldsymbol{\psi}_{\text{it}} & \psi_{\text{tt}} \end{pmatrix}, \tag{104}$$

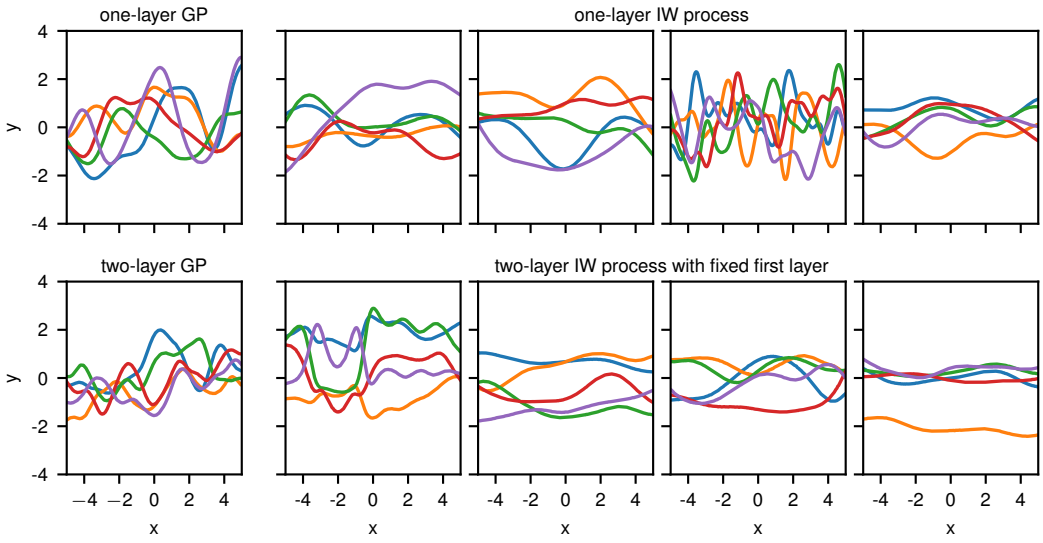

Figure 6. Samples from a one-layer (top) and a two-layer (bottom) deep IW process prior (Eq. 18). On the far left, we have included a set of samples from a GP with the same kernel, for comparison. This GP is equivalent to sending $\delta_0 \to \infty$ in the one-layer deep IW process and additionally sending $\delta_1 \to \infty$ in the two-layer deep IW process. All of the deep IW process panels use the same squared-exponential kernel with bandwidth 1. and $\delta_0 = \delta_1 = 0$. For each panel, we draw a single sample of the top-layer Gram matrix, $\mathbf{G}_L$, then draw multiple GP-distributed functions, conditioned on that Gram matrix.

where $\mathbf{G}^\ell_{\text{ii}}, \mathbf{\Psi}_{\text{ii}} \in \mathbb{R}^{P_\text{i} \times P_\text{i}}$, $\mathbf{g}^\ell_{\text{it}} \in \mathbb{R}^{P_\text{i} \times 1}$ and $\boldsymbol{\psi}_{\text{it}} \in \mathbb{R}^{P_\text{i} \times 1}$ are column-vectors, and g^ℓ_{tt} and ψ_{tt} are scalars. Taking the results in Eq. (34) to the univariate case,

$$g^\ell_{\text{tt}\cdot\text{i}} = g^\ell_{\text{tt}} - \mathbf{g}^{T\ell}_{\text{it}} \left(\mathbf{G}^\ell_{\text{ii}}\right)^{-1} \mathbf{g}^\ell_{\text{it}}, \qquad \psi_{\text{tt}\cdot\text{i}} = \psi_{\text{tt}} - \boldsymbol{\psi}^T_{\text{it}} \mathbf{\Psi}^{-1}_{\text{ii}} \boldsymbol{\psi}_{\text{it}}. \tag{105}$$

As $g^\ell_{\text{tt}\cdot\text{i}}$ is univariate, its distribution becomes Inverse Gamma,

$$g^\ell_{\text{tt}\cdot\text{i}} | \mathbf{G}^\ell_{\text{ii}}, \mathbf{G}_{\ell-1} \sim \text{InverseGamma}\left(\alpha = \tfrac{1}{2}\left(\delta_\ell + P_\text{t} + P_\text{i} + 1\right), \beta = \tfrac{1}{2}\psi_{\text{tt}\cdot\text{i}}\right). \tag{106}$$

As $\mathbf{g}^\ell_{\text{it}}$ is a vector rather than a matrix, its distribution becomes Gaussian,

$$\left(\mathbf{G}^\ell_{\text{ii}}\right)^{-1} \mathbf{g}^\ell_{\text{it}} | g^\ell_{\text{tt}\cdot\text{i}}, \mathbf{G}^\ell_{\text{ii}}, \mathbf{G}_{\ell-1} \sim \mathcal{N}\left(\mathbf{\Psi}^{-1}_{\text{ii}} \boldsymbol{\psi}_{\text{it}}, g^\ell_{\text{tt}\cdot\text{i}} \mathbf{\Psi}^{-1}_{\text{ii}}\right). \tag{107}$$

J. Samples from the 1D prior and approximate posterior

First, we drew samples from a one-layer (top) and two-layer (bottom) deep inverse Wishart process, with a squared-exponential kernel (Fig. 6). We found considerable differences in the function family corresponding to different prior samples of the top-layer Gram matrix, $\mathbf{G}_L$ (panels). While differences across function classes in a one-layer IW process can be understood as equivalent to doing inference over a prior on the lengthscale, this is not true of the two-layer process, and to emphasise this, the panels for two-layer samples all have the same first layer sample (equivalent to choosing a lengthscale), but different samples from the Gram matrix at the second layer. The two-layer deep IW process panels use the same, fixed input layer, so variability in the function class arises only from sampling $\mathbf{G}_2$.

Next, we exploited kernel flexibilities in IW processes by training a one-layer deep IW model with a fixed kernel bandwidth on data generated from various bandwidths. The first row in Figure 7 shows posterior samples from one-layer deep IW processes trained on different datasets. For each panel, we first sampled five full $\mathbf{G}_1$ matrices using Eq.(34a) and (34b). Then for each $\mathbf{G}_1$, we use Gaussian conditioning to get a posterior distribution on testing locations and drew one sample from the posterior plotted as a single line. Remarkably, these posterior samples exhibited wiggling behaviours that were consistent with training data even outside the training range, which highlighted the additional kernel flexibility in IW processes. On the other hand, when model bandwidth was fixed, samples from vanilla GPs with fixed bandwidth in the second row displayed almost identical shapes outside the training range across different sets of training data.

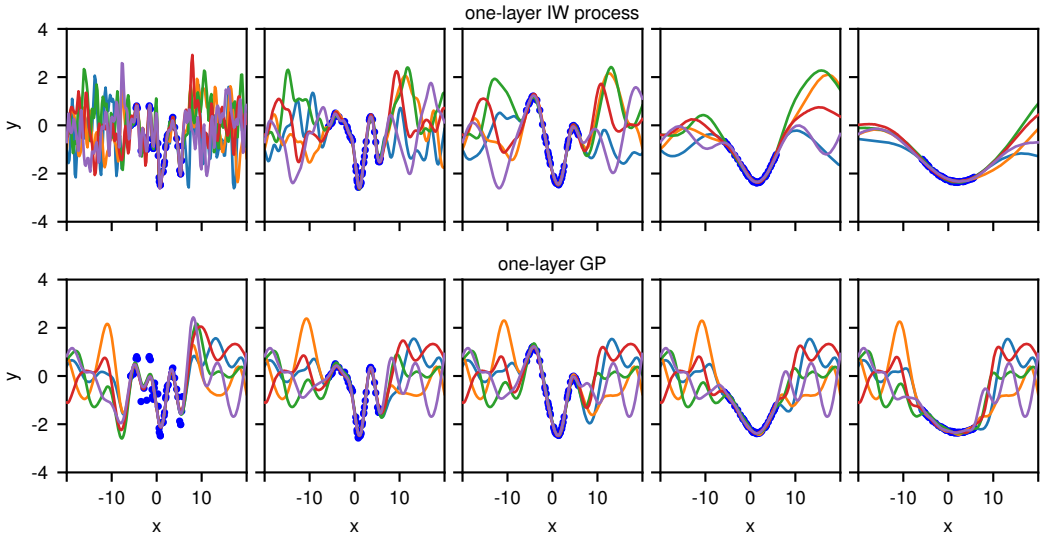

Figure 7. The additional flexibility in a one-layer deep IW process can be used to capture mismatch in the kernel. We plot five posterior function samples from trained IW processes in the first row, and samples from trained GPs below. We generate different sets of training data from a GP with different kernel bandwidths $(0.5, 1, 2, 5, 10)$ across columns, while we keep the kernel bandwidth in all models being 1.

K. Why we care about the ELBO

While we have shown that DIWP offers some benefits in predictive performance, it gives much more dramatic improvements in the ELBO. While we might think that predictive performance is the *only* goal, there are two reasons to believe that the ELBO itself is also an important metric. First, the ELBO is very closely related to PAC-Bayesian generalisation bounds (e.g. Germain et al., 2016). In particular, the bounds are generally written as the average training log-likelihood, plus the KL-divergence between the approximate posterior over parameters and the prior. This mirrors the standard form for the ELBO,

$$\mathcal{L} = \mathop{\mathbb{E}}_{\mathrm{Q}(z)} \left[\log \mathrm{P}\left(x|z\right)\right] - D_{\mathrm{KL}}\left(\mathrm{Q}\left(z\right) || \, \mathrm{P}\left(z\right)\right), \tag{108}$$

where x is all the data (here, the inputs, $\mathbf{X}$ and outputs, $\mathbf{Y}$), and z are all the latent variables. Remarkably, Germain et al. (e.g. 2016) present a bound on the test-log-likelihood that is exactly the ELBO per data point, up to additive constants. As such, in certain circumstances, optimizing the ELBO is equivalent to optimizing a PAC-Bayes bound on the test-log-likelihood. Similar results are available in Rivasplata et al. (2019). Second, we can write down an alternative form for the ELBO as the model evidence, minus the KL-divergence between the approximate and true posterior,

$$\mathcal{L} = \log \mathrm{P}\left(x\right) - D_{\mathrm{KL}}\left(\mathrm{Q}\left(z\right) || \, \mathrm{P}\left(z|x\right)\right) \leq \log \mathrm{P}\left(x\right). \tag{109}$$

As such, for a fixed generative model, and hence a fixed value of the model evidence, $\log \mathrm{P}\left(x\right)$, the ELBO measures the closeness of the variational approximate posterior, $\mathrm{Q}\left(z\right)$ and the true posterior, $\mathrm{P}\left(z|x\right)$. As we are trying to perform Bayesian inference, our goal should be to make the approximate posterior as close as possible to the true posterior. If, for instance, we can set $\mathrm{Q}\left(z\right)$ to give better predictive performance, but be further from the true posterior, then that is fine in certain settings, but not when the goal is inference. Obviously, it is desirable for the true and approximate posterior to be as close as possible, which corresponds to larger values of $\mathcal{L}$ (indeed, when the approximate posterior equals the true posterior, the KL-divergence is zero, and $\mathcal{L} = \log \mathrm{P}\left(x\right)$).

L. Differences with Shah et al. (2014)

For a one-layer deep inverse Wishart process, using our definition in Eq. (18)

$$\mathbf{K}_0 = \tfrac{1}{N_0}\mathbf{X}\mathbf{X}^T, \tag{110a}$$
$$\mathrm{P}\left(\mathbf{G}_1|\mathbf{K}_0\right) = \mathcal{W}^{-1}(\delta_1\mathbf{K}_0, \delta_1 + (P+1))\,, \tag{110b}$$
$$\mathrm{P}\left(\mathbf{y}_\lambda|\mathbf{K}_1\right) = \mathcal{N}\left(y_\lambda; \mathbf{0}, \mathbf{K}\left(\mathbf{G}_1\right)\right). \tag{110c}$$

Importantly, we do the nonlinear kernel transformation *after* sampling the inverse Wishart, so the inverse-Wishart sample acts as a generalised lengthscale hyperparameter (App. B), and hence dramatically changes the function family.

In contrast, for Shah et al. (2014), the nonlinear kernel is computed *before*, the inverse Wishart is sampled, and the inverse Wishart sample is used directly as the covariance for the Gaussian,

$$\mathbf{K}_0 = \mathbf{K}\left(\tfrac{1}{N_0}\mathbf{X}\mathbf{X}^T\right), \tag{111a}$$
$$\mathrm{P}\left(\mathbf{G}_1|\mathbf{K}_0\right) = \mathcal{W}^{-1}(\delta_1\mathbf{K}_0, \delta_1 + (P+1))\,, \tag{111b}$$
$$\mathrm{P}\left(\mathbf{y}_\lambda|\mathbf{K}_1\right) = \mathcal{N}\left(y_\lambda; \mathbf{0}, \mathbf{G}_1\right). \tag{111c}$$

This difference in ordering, and in particular, the lack of a nonlinear kernel transformation between the inverse-Wishart and the output is why Shah et al. (2014) were able to find trivial results in their model (that it is equivalent to multiplying the covariance by a random scale).