PoRB-Nets: Poisson Process Radial Basis Function Networks

Beau Coker
Department of Biostatistics
Harvard University

Melanie F. Pradier
SEAS
Harvard University

Finale Doshi-Velez
SEAS
Harvard University

Abstract

Bayesian neural networks (BNNs) are flexible function priors well-suited to situations in which data are scarce and uncertainty must be quantified. Yet, common weight priors are able to encode little functional knowledge and can behave in undesirable ways. We present a novel prior over radial basis function networks (RBFNs) that allows for independent specification of functional amplitude variance and lengthscale (i.e., smoothness), where the inverse lengthscale corresponds to the concentration of radial basis functions. When the lengthscale is uniform over the input space, we prove consistency and approximate variance stationarity. This is in contrast to common BNN priors, which are highly nonstationary. When the input dependence of the lengthscale is unknown, we show how it can be inferred. We compare this model's behavior to standard BNNs and Gaussian processes using synthetic and real examples.

1 INTRODUCTION

Neural networks (NNs) are flexible universal function approximators that have been applied with success in many domains. Bayesian neural networks (BNNs) capture function space uncertainty in a principled manner by placing priors over network parameters (Hinton and Neal, 1995). Unfortunately, priors in parameter space often lead to unexpected behavior in function space, making it difficult to incorporate meaningful information about function space properties (Lee, 2004). Two such properties of importance are amplitude variance and lengthscale, including how they might vary over the input space.

While Gaussian processes (GPs) are function priors that can easily encode these properties via the covariance function, there are many situations in which we would prefer BNNs to GPs: BNNs may be computationally more scalable, especially at test time, and they have an explicit parametric expression for posterior samples, which is convenient when additional computation is needed on the function (e.g., finding a minima) (Hernández-Lobato et al., 2014).

Therefore, a natural question arises: can we design BNN priors that encode function space properties as in GPs while retaining the benefits of BNNs? Some approaches use sample-based methods to evaluate the discrepancy between the function space distribution and a reference distribution with desired properties (Flam-Shepherd et al., 2017; Sun et al., 2019). Pearce et al. (2019) explores different BNN architectures to recover equivalent GP kernel combinations in the infinite width limit. While promising, these approaches require challenging optimizations or rely on infinite width assumptions.

As a first step towards more expressivity for BNNs, this work focuses on a particular type of NN called a radial basis function network (RBFN). RBFNs are widely used across scientific disciplines (Dash et al., 2016) and have received renewed interest recently, both from a theoretical (Que and Belkin, 2016) and inferential perspective (Zadeh et al., 2018; Asadi et al., 2020). Importantly, each hidden unit has a center parameter corresponding to a localized activation function, which enables controlling where (over the input space) the hidden units contribute to the complexity of the function.

In this work, we introduce Poisson Process Radial Basis Function Networks (PoRB-Nets), an interpretable family of RBFNs that employ a Poisson process (PP) prior over the center parameters in an RBFN. The proposed formulation enables direct specification of functional amplitude variance and lengthscale, the latter of which can vary over the input space. We show that these properties are *decoupled*; that is, each can be specified independently

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

of the other. Intuitively, PoRB-Nets work by trading off between the concentration and scale of the radial basis functions. Consider that a higher concentration of basis functions allows for a smaller lengthscale but also a larger variance, since the basis functions add up. By making the scale of the basis functions depend inversely on their concentration, PoRB-Nets undo the impact on the variance.

PoRB-Nets have the additional benefit that the choice of the lengthscale determines the network architecture (width of the layer), since the expected number of hidden units is equal to the integral of the PP intensity over the input space. Hidden units are added or deleted from the network during inference to adjust the overall lengthscale to the data, and when the input dependence of the lengthscale is unknown, we show how it can be inferred using a sigmoidal Gaussian Cox process as a prior (Adams et al., 2009). As with GPs, and unlike networks that force a specific property (Anil et al., 2018), these properties can adjust given data. We focus on single-layer RBFNs since our interest is in theoretical properties and examining the true posterior.

Specifically, we make the following contributions: (i) we introduce a novel, intuitive prior formulation for RBFNs that encodes distributional knowledge in function space, decoupling notions of lengthscale and amplitude variance in the same way as a GP with a radial basis function (RBF) kernel; (ii) we prove important theoretical properties of consistency and amplitude stationarity; (iii) we provide an inference algorithm to learn an input dependent lengthscale and (iv) we empirically demonstrate the potential of PoRB-Nets on synthetic and real examples. The code is available at https://github.com/dtak/porbnet.

2 RELATED WORK

Early weight space priors for BNNs. Most classical NN priors aim for regularization and model selection while minimizing the amount of undesired inductive biases (Lee, 2004). MacKay (1992) proposes a hierarchical prior[1] combined with empirical Bayes. Lee (2003) proposes an improper prior for NNs, which avoids the injection of prior biases at the cost of higher sensitivity to overfitting. Robinson (2001) proposes priors to alleviate overparametrization of NN models. We build on classical weight space priors but with the goal of obtaining specific properties in function space.

Function space priors for BNNs. Some works (Flam-Shepherd et al., 2017; Sun et al., 2019) match BNN priors to specific function space priors (e.g., GPs) but rely on sampling function values at a collection of input points. These approaches do not provide guarantees outside of the sampled region, and even in that region, their enforcement of properties is approximate. Neural processes (Garnelo et al., 2018) use meta-learning to identify functional properties that may be present in new functions, but they rely on having many prior examples and do not allow the user to specify basic properties directly. In contrast, we encode functional properties via prior design, without relying on function samples.

Bayesian formulations of RBFN models. Closest to our work are Bayesian formulations of RBFNs. Barber and Schottky (1998) consider a fixed number of hidden units, fixed scale, and use a Gaussian approximation to the posterior distribution, which is available in closed form in this case. Holmes and Mallick (1998) and Andrieu et al. (2001) propose fully Bayesian formulations that employ homogeneous Poisson process priors on the center parameters, but their focus is on inferring the number of hidden units and their formulation does not decouple amplitude variance and lengthscale.

3 BACKGROUND

Bayesian neural networks (BNNs). Let $y = f(x \mid \boldsymbol{w}, \boldsymbol{b}) + \epsilon$, where ϵ is a noise variable and $\boldsymbol{w}$ and $\boldsymbol{b}$ refer to the weights and biases of a neural network f respectively. In the Bayesian setting, we assume a prior $\boldsymbol{w}, \boldsymbol{b} \sim p(\boldsymbol{w}, \boldsymbol{b})$. One common choice is i.i.d. normal distributions over each parameter. For better comparison to PoRB-Nets we focus on BNNs with Gaussian $\phi(z) = \exp(-z^2)$ activations. We will refer to such a model as a standard BNN (Neal, 1996).

Radial basis function networks (RBFNs). RBFNs are classical shallow neural networks that approximate arbitrary nonlinear functions through a linear combination of radial kernels (Powell, 1987). They are universal function approximators (Park and Sandberg, 1991) and are widely used across disciplines such as numerical analysis, biology, finance, and classification in spatio-temporal models (Dash et al., 2016). For an input $x \in \mathbb{R}^D$, the output of a single-hidden-layer RBFN of width K is given by:

$$f(x \mid \boldsymbol{\theta}) = b + \sum_{k=1}^{K} w_k \exp\left(-\frac{1}{2} s_k^2 \|x - c_k\|^2\right), \quad (1)$$

where $s_k^2 \in \mathbb{R}$ and $c_k \in \mathbb{R}^D$ are the scale and center parameters, respectively, $w_k \in \mathbb{R}$ are the hidden-to-output

[1] Hierarchical priors are convenient when there is limited parameter interpretability. The addition of upper levels to the prior reduces the influence of the choice made at the top level, making the prior at the bottom level (the original parameters) more diffuse (Lee, 2004).

weights, and $b \in \mathbb{R}$ is the bias parameter. Each k-th hidden unit can be interpreted as a local receptor centered at c_k, with radius of influence s_k and relative importance w_k (Powell, 1987).

Poisson process. A Poisson process (PP) on $\mathbb{R}^D$ is a stochastic process characterized by a positive real-valued intensity function $\lambda(c)$. For any set $\mathcal{C} \subset \mathbb{R}^D$, the number of points in $\mathcal{C}$ follows a Poisson distribution with parameter $\int_{\mathcal{C}} \lambda(c)dc$. The process is *inhomogeneous* if $\lambda(c)$ is non-constant. We use a PP as a prior on the center parameters of an RBFN.

Gaussian Cox process. A Bayesian model consisting of a Poisson process likelihood and a log Gaussian process prior $g(c)$ on the intensity function $\lambda(c)$ is called a log Gaussian Cox Process (Møller et al., 1998). Adams et al. (2009) present an extension, called the *sigmoidal Gaussian Cox process*, which passes the Gaussian process through a scaled sigmoid function. To infer an input dependent lengthscale of an RBFN, we use this process as a model for the intensity function of the PP prior on the center parameters of the RBFN.

4 MODEL

In this section we introduce Poisson Process Radial Basis Function Networks (PoRB-Nets), which achieve two essential desiderata for a functional prior. First, they enable the user to encode the fundamental basic properties of lengthscale (i.e., smoothness), amplitude variance (i.e., signal variance), and (non)stationarity. Second, PoRB-Nets adapt the complexity of the network based on the inputs. For example, if the data suggests that the function needs to be less smooth in a certain input region, then that data can override the prior. Importantly, PoRB-Nets fulfill these desiderata while retaining appealing properties of NN-based models, as discussed in Section 1.

Generative model. As in a standard BNN, we assume a Gaussian likelihood centered on the network output, and independent Gaussian priors on the weight and bias parameters. Unique to the novel PoRB-Net formulation is a Poisson process prior over the set of center parameters and a deterministic dependence of the scale parameters on the Poisson process intensity. The generative model is given by:

$$\begin{aligned}
\{c_k\}_{k=1}^K \,|\, \lambda &\sim \exp\left(-\int_{\mathcal{C}} \lambda(c)dc\right) \prod_{k=1}^K \lambda(c_k) && (2)\\
s_k^2 \,|\, \lambda, c_k &= s_0^2 \lambda^2(c_k) && (3)\\
w_k &\sim \mathcal{N}\left(0, \sigma_w^2\right) && (4)\\
b &\sim \mathcal{N}\left(0, \sigma_b^2\right) && (5)\\
y_n \,|\, x_n, \boldsymbol{\theta} &\sim \mathcal{N}\left(f(x_n; \boldsymbol{\theta}), \sigma^2\right), && (6)
\end{aligned}$$

where $f(x_n; \boldsymbol{\theta})$ is given by Eq. (1); $\lambda : \mathcal{C} \to \mathbb{R}^+$ is the (possibly non-constant) Poisson process intensity; $\boldsymbol{\theta}$ is the set of RBFN parameters, including the centers, weights, bias, and intensity; and s_0^2 is a hyperparameter that defines the scale of the radial basis function when the intensity is one. In practice, s_0^2 allows the user to control the baseline number of hidden units. For example, if computational constraints limit the number of hidden units that can be used, decreasing s_0^2 allows the user to model a smaller lengthscale without adding more units.

Different priors could be considered for the intensity function λ. One simple case is to assume a uniform intensity $\lambda(c) = \lambda$ with $\lambda^2 \sim \text{Gamma}(\alpha_\lambda, \beta_\lambda)$. Under this specific formulation, Section 5 proves that the amplitude variance is stationary as the size of the region $\mathcal{C}$ tends to infinity, and Section 6 proves that the posterior regression function is consistent as the number of observations tends to infinity; such amplitude variance only depends on the variance of the hidden-to-output weights and output bias $\mathbb{V}[f(x)] \approx \sigma_b^2 + \tilde{\sigma}_w^2$, where $\tilde{\sigma}_w^2$ is just σ_w^2 scaled by s_0. We further show that the intensity λ controls the lengthscale.

Hierarchical prior for unknown input dependence of the lengthscale. In the case when the input-dependence of the lengthscale is unknown, we further model the intensity function $\lambda(c)$ of the Poisson process by a sigmoidal Gaussian Cox process (Adams et al., 2009):

$$\begin{aligned}
g &\sim \mathcal{GP}(0, \Sigma(\cdot,\cdot)) && (7)\\
\lambda(c) &= \lambda^* \sigma(g(c)), && (8)
\end{aligned}$$

where λ^* is an upper bound parameter on the intensity function and $\sigma(z) = (1 + e^{-z})^{-1}$ is the sigmoid function. In the forward pass of the network, we use the posterior mean of g to evaluate $\lambda(c)$.

Contrast to BNNs with Gaussian priors. In Sections 5 and 6, we prove that the proposed formulation has the desired properties described above. However, before doing so, we briefly emphasize that the i.i.d. Gaussian weight space prior commonly used with BNNs does not enjoy these properties. To see why, let us consider a standard feed-forward NN layer with 1-dimensional input and a Gaussian $\phi(z) = \exp(-z^2)$ activation function. We can rewrite the hidden units as $\phi(w_k x + b_k) = \phi(w_k(x - (-b_k/w_k)))$. This means that the corresponding center of the k-th hidden unit is $c_k = -b_k/w_k$ and the scale is $s_k = w_k$. If b_k and w_k have i.i.d. Gaussian priors with zero mean, as in standard BNNs, then the center parameter has a Cauchy distribution centered around zero. This is an important observation that motivates our work: A standard BNN concentrates the center of hidden units near the origin, resulting in nonstationary priors in function space.

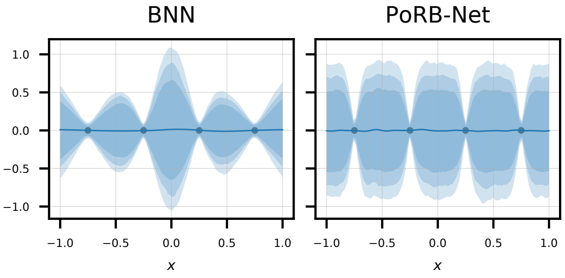

Figure 1: **PoRB-Net captures amplitude stationarity while a standard BNN does not.** Posterior predictive distributions given 4 observations.

5 VARIANCE AND LENGTHSCALE

We now return to the core desiderata: to specify a prior that separately controls a function's lengthscale and amplitude variance, as one could do using a GP with an RBF kernel. To do so, we first derive the covariance of the proposed PoRB-Net model. The full derivations supporting this section are available in Appendix A.

Neal (1996) showed that the covariance function for a single-layer BNN with a *fixed* number of hidden units $\rho(x;\theta_1),\ldots,\rho(x;\theta_K)$ and independent $\mathcal{N}(0,\sigma_w^2)$ and $\mathcal{N}(0,\sigma_b^2)$ priors on the hidden-to-output weights and output bias takes the following general form:

$$\operatorname{Cov}(f(x_1),f(x_2)) = \sigma_b^2 + \sigma_w^2 K \mathbb{E}_\theta\left[\rho(x_1;\theta)\rho(x_2;\theta)\right].$$

We show that the covariance function for a BNN with a *distribution* over the number of hidden units takes an analogous form, replacing the fixed number of hidden units K with its expectation:

$$\operatorname{Cov}(f(x_1),f(x_2)) = \sigma_b^2 + \sigma_w^2 \mathbb{E}\left[K\right] \underbrace{\mathbb{E}_\theta\left[\rho(x_1;\theta)\rho(x_2;\theta)\mid K\right]}_{:=U(x_1,x_2)}.$$

In the PoRB-Net model, $\theta = \{\lambda(\cdot), c_k\}$, $\rho(x;\theta_k) = \phi(\lambda(c_k)s_0\|x-c_k\|)$ where $\phi(z) = \exp(-\frac{1}{2}z^2)$, and $\mathbb{E}\left[K\right] = \int_{\mathcal{C}}\lambda(c)\,dc$. By deriving the form of $U(x_1,x_2)$ for the case of a homogeneous Poisson process, we next show that the covariance becomes increasingly stationary as the region $\mathcal{C}$ increases in size. We then illustrate how the covariance is decoupled from the lengthscale.

A homogeneous PP yields stationarity. In the case of constant intensity $\lambda(c)=\lambda$ defined over $\mathcal{C}=[C_0,C_1]$, the expression of $U(x_1,x_2)$ can be derived in closed form:

$$U(x_1,x_2) = \frac{1}{\mu(\mathcal{C})}\sqrt{\frac{\pi}{s^2}}\exp\left\{-s^2\left(\frac{x_1-x_2}{2}\right)^2\right\} \left[\Phi((C_1-x_m)\sqrt{2s^2}) - \Phi((C_0-x_m)\sqrt{2s^2}\lambda)\right], \quad (9)$$

where $s^2=s_0^2\lambda^2$, Φ is the cumulative distribution function of a standard Gaussian, and $x_m=(x_1+x_2)/2$ is the midpoint of the inputs. As the bounded region $\mathcal{C}$ increases, the second term approaches one, and so the covariance of a PoRB-Net approaches a squared exponential kernel with inverse lengthscale $s_0^2\lambda^2$ and amplitude variance $\tilde{\sigma}_w^2 := \sqrt{\pi/s_0^2}$ (defined for convenience):

$$\operatorname{Cov}\left(f(x_1),f(x_2)\right) \approx \sigma_b^2 + \tilde{\sigma}_w^2\exp\left\{-s_0^2\lambda^2\left(\frac{x_1-x_2}{2}\right)^2\right\}, \quad (10)$$

which is stationary since it only depends on the squared difference between x_1 and x_2. Notice that this result does not rely on an infinite width limit of the network, but only on the Poisson process region $[C_0,C_1]$ being relatively large compared to the midpoint x_m. In practice, $[C_0,C_1]$ can be set larger than the range of observed x values to achieve covariance stationarity over the input domain. Figure 2 shows that over the region $[-5,5]$ the analytical covariance from Equation (9) is fairly constant with only slight drops near the boundaries. In Appendix A we also derive the covariance when $\lambda^2\sim\text{Gamma}(\alpha_\lambda,\beta_\lambda)$, which results in a qualitatively similar shape. In contrast, the covariance function of an RBFN with a Gaussian prior on the center parameters is not approximately stationary. Specifically, for $c_k\sim\mathcal{N}(0,\sigma_c^2)$ and a fixed scale $s^2 = 1/(2\sigma_s^2)$, Williams (1997) shows that $U(x_1,x_2)$ takes the following form, which Figure 2 shows is highly nonstationary:

$$U(x_1,x_2) \propto \underbrace{\exp\left(-\frac{(x_1-x_2)^2}{2(2\sigma_s^2+\sigma_s^4/\sigma_c^2)}\right)}_{\text{Stationary}} \underbrace{\exp\left(-\frac{x_1^2+x_2^2}{2(2\sigma_c^2+\sigma_s^2)}\right)}_{\text{Nonstationary}}.$$

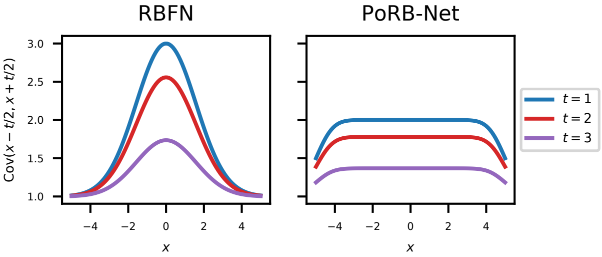

Figure 2: **PoRB-Net captures amplitude stationarity while an RBFN with a Gaussian prior on the centers does not.** The lines are $\operatorname{Cov}(x-t/2, x+t/2)$ for different t. We set all of $\sigma_w^2 = s_0^2 = s^2 = \lambda = 1$ and $\mathcal{C}=[-5,5]$.

Decoupling of variance and lengthscale. From Equation 9, notice the variance is $\mathbb{V}[f(x)]\approx\sigma_b^2+\tilde{\sigma}_w^2$, which has no dependence on the intensity λ, freeing it to act as an inverse lengthscale. This is a point of differentiation of PoRB-Nets. If the scale were fixed or independent of the intensity, as is the case in previous priors over RBFNs (e.g., Holmes and Mallick (1998)), the variance would be $\mathbb{V}[f(x)] \approx \sigma_b^2 + \lambda\tilde{\sigma}_w^2$. Intuitively this happens because a higher intensity implies a higher number of basis functions, which implies a higher amplitude variance as the basis functions add up. If we instead allow the scale parameters s^2 to increase as a function of the intensity, thus making the radial basis functions more narrow, we can counteract the impact of their concentration on the amplitude.

To support the hypothesis that the intensity λ controls the lengthscale, we examine the average number of upcrossings of $y = 0$ of sample functions. For a GP with an RBF kernel, the expected number of upcrossings u over the unit interval is inversely related to the lengthscale l via $u = (2\pi l)^{-1}$. Figure 3 shows a histogram of the upcrossings from functions drawn from a PoRB-Net with a stepwise intensity $\lambda(c)$ (greater above $x = 0$). Notice the lengthscale is clearly smaller above $x = 0$ but the amplitude variance $\mathbb{V}[f(x)]$ is approximately constant for all x.

An inhomogeneous PP yields non-stationarity. When the intensity is a non-constant function $\lambda(c)$, then Equation (9) does not hold. However, we find that setting the scale parameter of each hidden unit to $s_k^2 = s_0^2\lambda(c_k)^2$, where $\lambda(c_k)$ is the intensity evaluated at the center parameter c_k, allows for an input dependent lengthscale that is approximately decoupled from the variance.

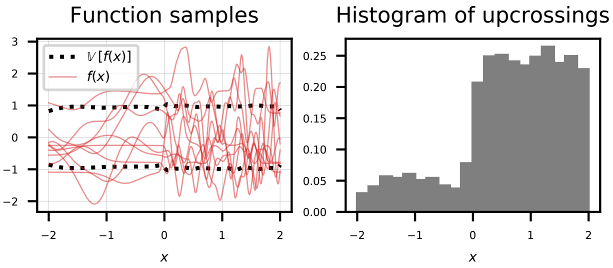

Figure 3: **PoRB-Nets decouple lengthscale (as measured by the upcrossings) and variance**.

6 CONSISTENCY

In this section, we study *consistency of predictions*. That is, as the number of observations goes to infinity, whether the posterior predictive concentrates around the true function. When dealing with priors that can produce an unbounded number of parameters, consistency is a basic but important property. To our knowledge, we are the first to provide consistency for RBFNs with a Poisson distributed number of hidden units (no consistency guarantees were derived by Andrieu et al. (2001)).

Define $r_0(x)$ to be the true regression function and $\hat{r}_n(x) = \mathbb{E}_{\hat{f}_n}[Y \mid X]$ to be the estimated regression function, where $\hat{p}_n$ is the estimated density in parameter space based on n observations. The estimator $\hat{r}_n(x)$ is said to be consistent with respect to the true regression function $r_0(x)$ if, as n tends to infinity:

$$\int (\hat{r}_n(x) - r_0(x))^2 \, dx \xrightarrow{p} 0. \tag{11}$$

Doob's theorem shows that Bayesian models are consistent as long as the prior places positive mass on the true parameter (Miller, 2018). For finite dimensional parameter spaces, one can ensure consistency by simply restricting the set of zero prior probability to have arbitrarily small or zero measure. Unfortunately, in infinite dimensional parameter spaces, this set might be very large (Freedman, 1963). In our case where functions correspond to uncountably infinite sets of parameters, we cannot restrict this set of inconsistency to have measure zero.

Instead, we aim to show a strong form of consistency called Hellinger consistency. We closely follow the approach of Lee (2000), who shows consistency for standard BNNs with normal priors on the parameters. Formally, let $(x_1, y_1), \ldots, (x_n, y_n) \sim p_0$ be the observed data drawn from the ground truth density p_0 and define the Hellinger distance between joint densities p and p_0 over (X, Y) as:

$$D_H(p, p_0) = \sqrt{\iint \left(\sqrt{p(x,y)} - \sqrt{p_0(x,y)}\right)^2 \, dx\, dy}.$$

The posterior is said to be consistent over Hellinger neighborhoods if for all $\epsilon > 0$,

$$p(\{f : D_H(p, p_0) \leq \epsilon\}) \xrightarrow{p} 1.$$

Lee (2000) shows that Hellinger consistency of joint density functions implies frequentist consistency as described in Equation (11). The following theorem describes an analogous result for PoRB-Nets with homogeneous intensities.

Theorem 1. *(Consistency of PoRB-Nets) A radial basis function network with a homogeneous Poisson process prior on the location of hidden units is Hellinger consistent as the number of observations goes to infinity.*

Proof. Leveraging the results and proof techniques from Lee (2000), we use bracketing entropy from empirical process theory to bound the posterior probability outside Hellinger neighborhoods. We need to check that this model satisfies two key conditions. Informally, the first condition is that the prior probability placed on parameters larger in absolute value than a bound B_n, where B_n is allowed to grow with the data, is asymptotically bounded *above* by an exponential term $\exp(-nt)$, for some $t > 0$. The second condition is that the prior probability placed on KL neighborhoods of the ground truth density function p_0 is asymptotically bounded *below* by an exponential term $\exp(-n\nu)$, for some $\nu > 0$. The proof is in the Appendix B. □

Note that consistency of predictions does not imply concentration of the posterior in weight space, since radial basis function networks, like other deep neural models, are not identifiable.

7 INFERENCE

We infer the posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$ over the network parameters $\boldsymbol{\theta}$ with Markov-Chain Monte Carlo (MCMC) and model predictions for new observations and their associated uncertainties with the posterior predictive distribution:

$$p(y^\star | x^\star, \mathcal{D}) = \int p(y^\star | x^\star, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathcal{D}) d\boldsymbol{\theta}.$$

The inference algorithm can be broken down into three steps. Step 1 updates the network weight, center, and bias parameters $\left(\{w_k, c_k\}_{k=1}^K, b\right)$ conditional on the network width K and intensity function with Hamiltonian Monte-Carlo (HMC) (Neal, 1996). Step 2 updates the network width K conditional on the network parameters and intensity function with birth and death Metropolis-Hastings (MH) steps. Finally, Step 3 updates the Poisson process intensity conditional on the other network parameters and network width. In the case of a homogeneous intensity with a Gamma prior, we use an MH step. In the case of a inhomogeneous intensity defined by Equations 7 and 8 we follow the inference procedure of Adams et al. (2009) for a sigmoidal Gaussian Cox process, treating the current center parameters $\{c_k\}$ as the observed events. This involves introducing three auxiliary variables: a collection of "thinned" center parameters $\{\tilde{c}_m\}$, the number of thinned center parameters M, and the latent GP evaluated at the thinned center parameters $\{\tilde{g}_m\}$. Step 3 requires updating each of these auxiliary variables, along with the latent GP values $\{g_k\}$ evaluated at the current center parameters $\{c_k\}$. For convenience we define $\mathbf{g}_{M+K}$ as vector concatenating $\{\tilde{g}_m\}_{m=1}^M$ and $\{g_k\}_{k=1}^K$ and $\mathbf{c}_{M+K}$ as the vector concatenating $\{\tilde{c}_m\}_{m=1}^M$ and $\{c_k\}_{k=1}^K$. We also define $L(\boldsymbol{\theta})$ as the likelihood of the data given all network parameters. We next describe these steps in more detail assuming a sigmoidal Gaussian Cox process prior on an inhomogeneous intensity $\lambda(c)$, but the full details of the inference procedure are available in the Appendix C.

Step 1: Update network weights, bias, and centers. The full conditional distribution of the weights, bias, and centers can be written as:

$$\begin{aligned}
&p(\{w_k\}, b, \{c_k\} \mid K, \{c_m\}, \{\tilde{g}_m\}, \{\tilde{g}_k\}) \\
&\propto L(\boldsymbol{\theta}) \exp\left\{-\frac{1}{2\sigma_b^2} b^2\right\} \exp\left\{-\frac{1}{2\sigma_w^2} \sum_{k=1}^K w_k^2\right\} \\
&\quad |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2} \mathbf{g}_{M+K}^T \Sigma^{-1} \mathbf{g}_{M+K}\right\},
\end{aligned}$$

where Σ is the kernel matrix of the GP underlying the intensity evaluated at all of the center parameters. We use HMC, which requires tuning L leap-frog steps of size ϵ, to propose updates from this distribution.

Step 2: Update network width K. We adapt the network width with birth or death Metropolis-Hastings (MH) steps chosen with equal probability. For a birth step, we propose a weight w' and a center c' from their prior distributions, and we propose a GP function value g' (representing $g(c')$) from the GP conditioned on the current function values $\mathbf{g}_{M+K}$ observed at $\mathbf{c}_{M+K}$. For the death step, we propose to delete the k'th hidden unit by uniformly selecting among the existing hidden units. Therefore, we can write the hidden unit birth and death proposal densities as follows:

$$\begin{aligned}
q(K \to K+1) &\propto \mathcal{N}(w'; 0, \sigma_w^2) \\
&\qquad p(g' \mid c', \mathbf{c}_{M+K}, \mathbf{g}_{M+K}) / \mu(\mathcal{C}) \\
q(K \to K-1) &= 1/K
\end{aligned}$$

Note that since the GP has a zero mean function, we propose c' uniformly over $\mu(\mathcal{C})$, but for any fixed intensity we propose from the density $\lambda(c)/\Lambda$. The acceptance rates work out to:

$$\begin{aligned}
a_{\text{birth}} &= \frac{L(\boldsymbol{\theta}')}{L(\boldsymbol{\theta})} \frac{\lambda^* \sigma(g') \mu(\mathcal{C})}{K+1} \\
a_{\text{death}} &= \frac{L(\boldsymbol{\theta}')}{L(\boldsymbol{\theta})} \frac{K}{\lambda^* \sigma(g_{k'}) \mu(\mathcal{C})}.
\end{aligned}$$

Step 3: Update Poisson process intensity λ. We adopt an inference procedure similar to (Adams et al., 2009) with two crucial differences: the "events" $\{c_k\}$ (center parameters in our case) are unobserved and the full conditional of the function values $\mathbf{g}_{M+K}$ includes the likelihood $L(\boldsymbol{\theta})$ of the data $\mathcal{D}$, since the forward pass of the network uses the posterior mean of g to evaluate the intensity $\lambda(c) = \lambda^* \sigma(g(c))$. We proceed as follows: i) update the number M of thinned centers using birth and death

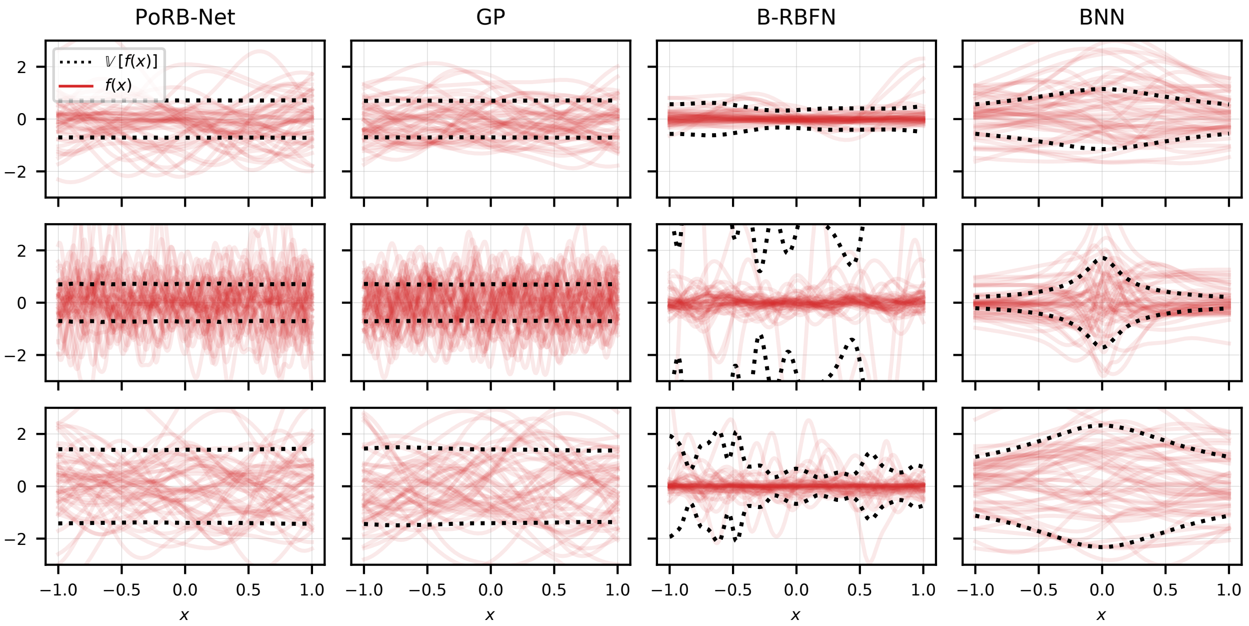

Figure 4: **PoRB-Net allows for easy specification of lengthscale and amplitude like a GP.** We show prior samples from PoRB-Net with a homogeneous intensity, a GP with RBF kernel, B-RBFN (Andrieu et al., 2001), and a BNN (Neal, 1996) with a Gaussian activation. Compared to the first row, the second row has lower lengthscale and similar amplitude, while the third row has higher amplitude and similar lengthscale.

steps, analogous to updating the number of actual centers K; ii) update the thinned center parameters $\{c_m\}_{m=1}^M$ using MH steps with perturbative proposals; iii) update the GP function values $\mathbf{g}_{M+K}$ using HMC.

8 RESULTS

Next we empirically demonstrate desirable properties of PoRB-Net. In particular, PoRB-Net allows for (a) easy specification of lengthscale and amplitude variance information (analogous to a GP), and (b) learning of an input-dependent lengthscale. We report additional empirical results on synthetic and real datasets in Appendix D.

PoRB-Net allows for easy specification of stationary lengthscale and signal variance. Figure 4 shows prior function samples from different models (columns) with different prior settings (rows). Compared to the top row, the second row has a smaller overall lengthscale and the bottom row has a higher overall variance. We plot 50 function samples (red lines) and the estimated variance based on 10,000 function samples (black, dotted line). Like a GP, the amplitude variance of PoRB-Net is constant over the input space and does not depend on the lengthscale. On the other hand, the model of Andrieu et al. (2001) (B-RBFN), which effectively assumes a homogeneous Poisson process prior on the center parameters but does not rescale the basis functions based on the intensity, has a variance that changes over the input space and *does* depend on the lengthscale. For a standard BNN (last column), the amplitude variance and lengthscale are concentrated near the origin and the variance increases as we decrease the lengthscale (from 1st to 2nd row).

PoRB-Net can recover a known, input dependent lengthscale. Figure 5 illustrates the capacity of PoRB-Net to infer an input-dependent lengthscale. Here the true function is a GP with a sinusoidal lengthscale (see kernel in the Appendix D). The right panel shows the center parameter intensity, inferred from noisy (x, y) observations, corresponds to the inverse of the true lengthscale.

PoRB-Nets exhibit competitive performance on synthetic and real datasets. We compare the performance of PoRB-Nets, GPs, and single-layer BNNs with Gaussian activations, with the first two sets of models trained with and without inferring the input dependence of the lengthscale. For the GP models, to use a constant lengthscale we use a regular GP with an RBF kernel; to infer an input dependent lengthscale we use the nonstationary GP model of Heinonen et al. (2016), which we denote by LGP.

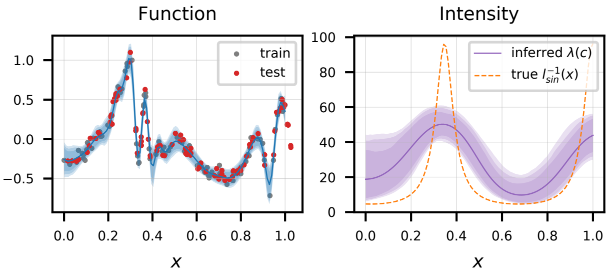

Figure 5: **PoRB-Net is able to learn input-dependent lengthscale information.** The ground truth synthetic example has been generated from a nonstationary GP with a sinusoidal lengthscale function $l_{\text{sin}}(x)$.

At a high level, we see qualitative similarity between PoRB-Nets and GPs that infer the lengthscale, and PoRB-Nets and GPs that do not infer the lengthscale, but the BNNs look different from the rest. This is due to the nonstationarity of the prior, which has higher variability near the origin. All models except the GP are inferred using HMC (including the LGP).

We use four synthetic datasets — all drawn from GPs with known lengthscale functions $l(x)$ — and six real, nonstationary time series datasets – four from mimic (Johnson et al., 2016), the CBOE volatility index over one year starting in October 2018 ("finance"), and the motorcycle dataset (Silverman, 1985). The datasets drawn using a sinusoidal lengthscale $l_{\text{sin}}(x)$ and an increasing lengthscale (from left to right) $l_{\text{inc}}(x)$ can be seen in Figures 5 and 6, respectively. $l_{\text{const}}(x)$ is a constant lengthscale, on which the GP with a stationary, RBF kernel not surprisingly performs best (with PoRB-Net coming in second).

To highlight differences in model behavior rather than prior specification, we first identify the variance and lengthscale parameters that optimize the log marginal likelihood of the GP. We then match the overall variance and lengthscale (as measured by the number of upcrossings mentioned in Section 5) of the BNN and PoRB-Net to the GP by a grid search over the model parameters. Note that the BNN will still have a different input dependence of variance and upcrossings over the input space (both concentrated near the origin). Since adjusting the lengthscale of PoRB-Net adjusts the prior expected number of hidden units, and during inference they can further adapt to the data, we train BNNs with 25, 50, and 100 units, roughly corresponding to the range of units used by PoRB-Net.

There are two main takeaways from these results:

- Examining the posterior predictives in Figure 6 qualitatively, both PoRB-Net and the LGP adapt the local lengthscale to the smoothness of the data, though the effect is more pronounced in the LGP. In contrast, the BNN underestimates uncertainty near $x \approx .2$ in the synthetic dataset (top row) and overestimates uncertainty near $x \approx .8$ in the real dataset (bottom row).

Table 1: **Test Log Likelihoods.** For the BNN, we show the best(worst) performance among models of size 25, 50, and 100 units.

	PoRB-Net†	PoRB-Net	GP	LGP	BNN
sin*	0.77	**0.82**	0.73	0.81	0.79 (0.74)
inc*	-0.40	0.00	-0.23	**0.18**	-0.15 (-0.28)
inc2*	0.66	**0.75**	0.54	0.18	0.68 (0.63)
const*	0.28	0.33	**0.41**	0.24	0.01 (-0.30)
mimic1	0.89	0.95	0.83	0.90	**1.05** (0.91)
mimic2	0.53	**0.60**	0.56	0.54	0.47 (0.39)
mimic3	-0.63	**-0.57**	-0.67	-0.58	-0.59 (-0.65)
mimic4	-1.72	-1.53	-1.85	-1.44	**-0.59** (-1.38)
finance	-1.41	-0.52	-1.97	**0.03**	-0.73 (-2.63)
motor.	**0.18**	0.16	0.17	0.14	0.16 (0.12)

**synthetic dataset* †*infers homogeneous intensity*

- The test log likelihoods in Table 1 show PoRB-Net exhibits strong performance across the datasets. In contrast, the performance of the BNN varies greatly by the number of hidden units. PoRB-Nets remove this choice by averaging over different numbers of units, fully taking advantage of the Bayesian paradigm.

Test RMSEs, posterior predictives, and inferred intensities for all datasets are available in the Appendix D. Note that HMC is a gold standard for posterior inference; the fact that the standard BNN lacks desirable properties under HMC demonstrates that its failings come from the model and not the inference.

9 CONCLUSION

This work presents a novel Bayesian prior for neural networks called PoRB-Net that allows for easy encoding and inference of two basic functional properties: amplitude variance and lengthscale. We provide a principled inference scheme and future work can address how it can be scaled.

Under standard BNN formulations, we show that it is impossible to get such properties. The essential pieces to achieve these properties were: i) a center-scale parametrization (instead of classical weight-bias), ii) an automatic adaptation of the number of hidden units, and iii) a rescaling of the radial basis functions based on their concentration.

We focused on Gaussian activations because they have

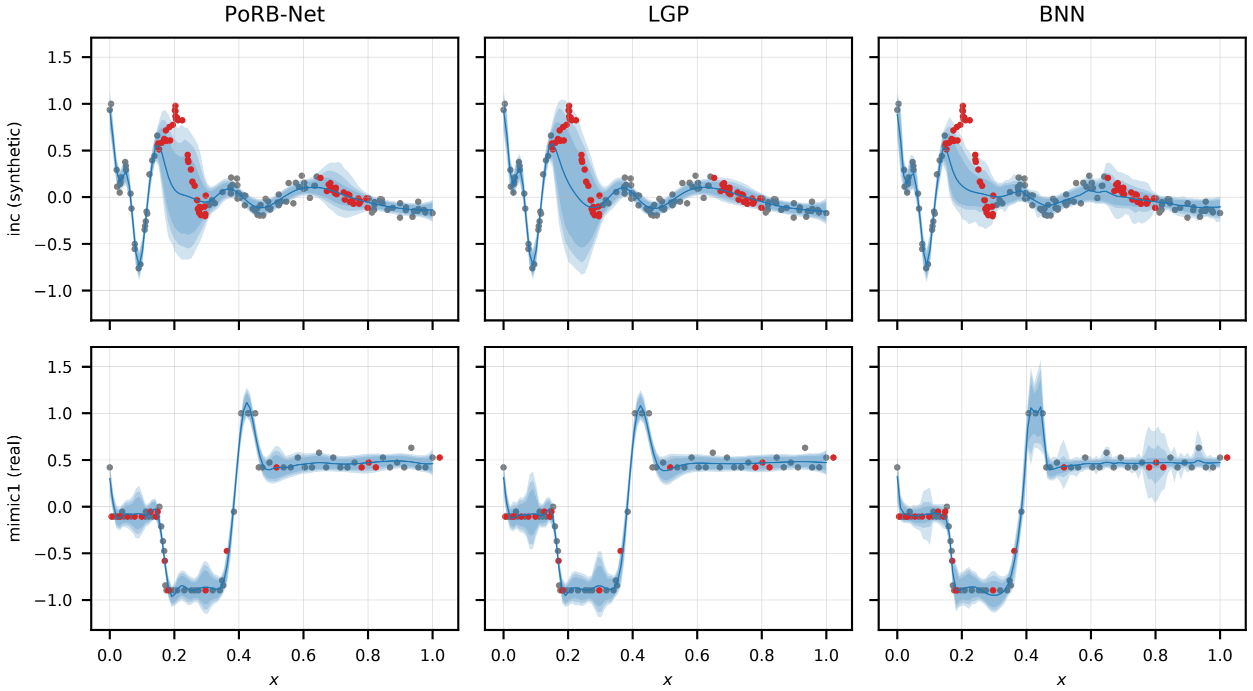

Figure 6: **PoRB-Net posterior predictive captures non-stationary patterns in real scenarios, adapting the lengthscale locally as needed.** Priors for all models have been matched to have about the same amplitude variance and lengthscale. BNNs exhibit undesired uncertainty while PoRB-Nets and LGPs adapt the local uncertainty to the data. Gray points used for training and red points used for testing.

a limited region of effect, unlike other popular activations like tanh or ReLU. Exploring how to get desirable properties for those activations seems challenging, and remains an area for future exploration. That said, we emphasize that RBFNs are commonly used in many practical applications, as surveyed in (Dash et al., 2016).

Finally, all of our work was developed in the context of single-layer networks. From a theoretical perspective this is not an overly restrictive assumption, as single layer networks are still universal function approximators (Park and Sandberg, 1991). However, deep RBFNs, where only the last layer has a radial basis function parameterization, have received renewed interest (Zadeh et al., 2018), so exploring deep PoRB-Nets is an interesting area of future work.

Given the popularity of NNs and the need for uncertainty quantification in them, understanding prior assumptions—which will govern how we will quantify uncertainty—is essential. If prior assumptions are not well understood and not properly specified, the Bayesian framework makes little sense: the posteriors that we find may not be ones that we expect or want. Though we focus on RBFNs, our work provides an important step toward specifying NN priors with desired basic functional properties.

References

Adams, R. P., Murray, I., and MacKay, D. J. C. (2009). Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Andrieu, C., Freitas, N. d., and Doucet, A. (2001). Robust full Bayesian learning for radial basis networks. *Neural Computation*, 13(10):2359–2407.

Anil, C., Lucas, J., and Grosse, R. (2018). Sorting out Lipschitz function approximation. *arXiv:1811.05381 [cs.LG]*.

Asadi, K., Parr, R. E., Konidaris, G. D., and Littman, M. L. (2020). Deep RBF value functions for continuous control. *arXiv:2002.01883 [cs.LG]*.

Barber, D. and Schottky, B. (1998). Radial basis functions: a Bayesian treatment. In *Advances in Neural Information Processing Systems 10*.

Dash, C. S. K., Behera, A. K., Dehuri, S., and Cho, S.-B. (2016). Radial basis function neural networks: a topical state-of-the-art survey. *Open Computer Science*, 6(1).

Flam-Shepherd, D., Requeima, J., and Duvenaud, D. (2017). Mapping Gaussian process priors to Bayesian neural networks. In *NIPS Bayesian Deep Learning Workshop*.

Freedman, D. A. (1963). On the aymptotic behavior of Bayes' estimates in the discrete case. *The Annals of Mathematical Statistics*, 34(4):1386–1403.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018). Neural processes. *arXiv:1807.01622 [cs.LG]*.

Heinonen, M., Mannerström, H., Rousu, J., Kaski, S., and Lähdesmäki, H. (2016). Non-stationary gaussian process regression with hamiltonian monte carlo. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. *arXiv:1406.2541 [stat.ML]*.

Hinton, G. E. and Neal, R. M. (1995). Bayesian learning for neural networks.

Holmes, C. C. and Mallick, B. K. (1998). Bayesian radial basis functions of variable dimension. *Neural computation*, 10(5):1217–1233.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Lee, H. (2004). *Bayesian nonparametrics via neural networks*. SIAM.

Lee, H. K. (2000). Consistency of posterior distributions for neural networks. *Neural Networks: The Official Journal of the International Neural Network Society*, 13(6):629–642.

Lee, H. K. (2003). A noninformative prior for neural networks. *Machine Learning*, 50(1-2):197–212.

MacKay, D. J. (1992). *Bayesian methods for adaptive models*. PhD Thesis, California Institute of Technology.

Miller, J. W. (2018). A detailed treatment of Doob's theorem. *arXiv:1801.03122 [math.ST]*.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482.

Neal, R. M. (1996). Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer.

Park, J. and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257.

Pearce, T., Tsuchida, R., Zaki, M., Brintrup, A., and Neely, A. (2019). Expressive priors in Bayesian neural networks: kernel combinations and periodic functions. *arXiv:1905.06076 [stat.ML]*.

Powell, M. J. D. (1987). Algorithms for Approximation. pages 143–167. Clarendon Press.

Que, Q. and Belkin, M. (2016). Back to the future: radial basis function networks revisited. In *Artificial Intelligence and Statistics*, pages 1375–1383.

Robinson, M. (2001). *Priors for Bayesian Neural Networks*. PhD thesis, University of British Columbia.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society*, 47:1–52.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional variational Bayesian neural networks. In *International Conference on Learning Representations*.

Williams, C. K. (1997). Computing with infinite networks. In *Advances in Neural Information Processing Systems 9*.

Zadeh, P. H., Hosseini, R., and Sra, S. (2018). Deep-RBF Networks Revisited: Robust Classification with Rejection. *arXiv:1812.03190 [cs.LG]*.