Received: 25 February 2023 | Revised: 30 May 2023 | Accepted: 17 June 2023

DOI: 10.1002/sta4.604

ORIGINAL ARTICLE

WILEY

Density regression and uncertainty quantification with Bayesian deep noise neural networks

Daiwei Zhang[1] | Tianci Liu[2] | Jian Kang[3]

[1]Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, Pennsylvania, 19104, USA

[2]School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, 47907, USA

[3]Department of Biostatistics, University of Michigan, Ann Arbor, Michigan, 48109, USA

Correspondence
Jian Kang, 1415 Washington Heights, Ann Arbor, MI 48109, USA.
Email: jiankang@umich.edu

Funding information
National Science Foundation; NIH, Grant/Award Numbers: NIH R01MH105561, NIH R01DA048993, NIH R01GM124061

Deep neural network (DNN) models have achieved state-of-the-art predictive accuracy in a wide range of applications. However, it remains a challenging task to accurately quantify the uncertainty in DNN predictions, especially those of continuous outcomes. To this end, we propose the Bayesian deep noise neural network (B-DeepNoise), which generalizes standard Bayesian DNNs by extending the random noise variable from the output layer to all hidden layers. Our model is capable of approximating highly complex predictive density functions and fully learn the possible random variation in the outcome variables. For posterior computation, we provide a closed-form Gibbs sampling algorithm that circumvents tuning-intensive Metropolis–Hastings methods. We establish a recursive representation of the predictive density and perform theoretical analysis on the predictive variance. Through extensive experiments, we demonstrate the superiority of B-DeepNoise over existing methods in terms of density estimation and uncertainty quantification accuracy. A neuroimaging application is included to show our model's usefulness in scientific studies.

KEYWORDS
Bayesian methods, machine learning, neural networks

1 | INTRODUCTION

Deep neural networks (DNNs) have achieved outstanding prediction performance in a wide range of artificial intelligence (AI) applications (Berner et al., 2021; Pouyanfar et al., 2018). Despite overwhelming cases of success, a major drawback of standard DNNs is the lack of reliable uncertainty quantification (UQ) (Begoli et al., 2019). UQ is an essential task in safety-critical AI applications (Amodei et al., 2016).

For example, in medical diagnosis, an individualized risk assessment AI model should be able to report its confidence in its predictions. When the AI model is not sufficiently certain in its assessment of a patient, the patient should be referred to human physicians for further evaluation (Jiang et al., 2012; Leibig et al., 2017).

In this work, we seek to solve the problem of UQ in DNN regression tasks, where a representation of the outcome's total random variation cannot be achieved by a finite-length probability vector (unlike in classification tasks) but rather requires an infinite-dimensional predictive probability density function. In a standard DNN regression model, the outcome $y_i \in \mathbb{R}$ and the predictors $\mathbf{x}_i \in \mathbb{R}^P$ are assumed to follow the relation $y_i = f(\mathbf{x}_i) + \epsilon_i$, where the mean function $f(\mathbf{x}_i) = \mathbb{E}(y_i|\mathbf{x}_i)$ is constructed by a DNN, and the random noise ϵ_i follows a zero-mean homoscedastic Gaussian distribution $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ for some unknown $\sigma^2 > 0$. This formulation implies the conditional variance of the outcome variable $\text{Var}(y_i|\mathbf{x}_i)$, given the predictor, to be constant. However, in real applications, the true conditional distribution could be heteroscedastic Gaussian (i.e., $\text{Var}(y_i|\mathbf{x}_i) = \sigma^2(\mathbf{x}_i)$) or not Gaussian at all (e.g., the distribution of ϵ_i is asymmetric or multimodal).

In these cases, simple UQ statistics (e.g., prediction variance or prediction interval width) may fail to capture important patterns in the outcome variable and result in underconfident or overconfident UQ. In order to achieve accurate UQ in DNN regression, it is critical to learn the predictive density of the outcome given the predictors. We refer to the problem of estimating the predictive density function as the *density regression* (DR) problem (Dunson et al., 2007). This works focuses on DR tasks with DNNs. As uncertainty in the outcome variable is quantified by summary statistics of the predictive density, a solution to DR trivially leads to a solution to UQ, regardless of the chosen UQ metric.

1.1 | Related work

To estimate predictive density function, several DNN-based frequentist DR methods have been proposed (Abdar et al., 2021; Caldeira & Nord, 2020; Ståhl et al., 2020; Zhu et al., 2019), such as mixture density networks (Bishop, 1994; Bishop & Nasrabadi, 2006), deep ensembles (Lakshminarayanan et al., 2016), distribution-free methods (Lei & Wasserman, 2014; Pearce et al., 2018), quantile-based models (Romano et al., 2019; Tagasovska & Lopez-Paz, 2019), categorization (Li et al., 2021), and estimation of the predictive cumulative distribution function (CDF) (Huberman et al., 2021).

Compared to ad hoc frequentist DR methods, Bayesian frameworks for DNNs, also known as Bayesian neural networks (BNNs) (MacKay, 1995; Neal, 2012; Xue et al., 2019), provide a more natural and systematic solution to the task of DR. In addition to the asymptotic well-calibratedness of the posterior prediction intervals (Hwang & Ding, 1997; Sun et al., 2021; Wang & Rocková, 2020), the Bayesian framework also improves the prediction accuracy of deterministic DNNs (Izmailov et al., 2018; Kendall & Gal, 2017). Due to the intractability of BNNs' posterior distributions, variational inference (VI) or Markov chain Monte Carlo (MCMC) simulation are required for posterior computation. VI methods (Blei et al., 2017; Kingma & Welling, 2013; Mandt et al., 2017) approximate the posterior distribution with simpler distributions (Graves, 2011; Louizos & Welling, 2016; Lee et al., 2020; Louizos & Welling, 2017; Rezende & Mohamed, 2015). Common randomness-based regularization techniques for DNNs, such as dropout (Gal & Ghahramani, 2016; Molchanov et al., 2017; Srivastava et al., 2014), batch normalization (Ioffe & Szegedy, 2015; Teye et al., 2018), and random weights (Blundell et al., 2015; Hernández-Lobato & Adams, 2015), can be interpreted as special cases of VI. However, although computationally efficient, VI methods induce extra approximation errors in posterior computation, which may lead to underestimated variance or oversimplified covariance structures (Blei et al., 2017).

In contrast to VI, MCMC methods simulate the exact posterior distribution. The most popular MCMC algorithm for modern Bayesian methods is arguably the Metropolis–Hastings (MH) algorithm (Andrieu & Thoms, 2008; Chib & Greenberg, 1995; Hitchcock, 2003). However, even with efficient techniques such as Hamiltonian dynamics (Wenzel et al., 2020; Wilson & Izmailov, 2020), Langevin dynamics (Welling & Teh, 2011), stochastic gradients (Chen et al., 2014; Chen et al., 2016), and mini-batches (Wu et al., 2020) that mitigate the high computation burden (Jospin et al., 2022; Liang et al., 2016), MH-based MCMC methods require intensive hyperparameter tuning in order to compute the posterior distribution efficiently. As an alternative MCMC simulation method, Gibbs sampling algorithms (Geman & Geman, 1984; Gelfand & Smith, 1990; Gelfand, 2000; Roberts & Smith, 1994) draw posterior samples for each model parameter (or a block of model parameters) conditioned on all the other parameters. Although Gibbs sampling methods have been developed for deep generative models such as sigmoid belief networks (Gan et al., 2015), Gibbs sampling cannot be applied to standard predictive BNNs, due to their lack of closed-form posterior full conditional distributions. Finally, most existing MCMC- and VI-based BNN methods focus on UQ exclusively, whereas the problem of DR in Bayesian frameworks has only been studied for linear models (Dunson et al., 2007).

In addition, for DR tasks with DNN, no existing method incorporates latent noise in hidden layers, to our best knowledge. Previous works on latent noise in DNNs primarily use it for regularization (Gulcehre et al., 2016; You et al., 2019). The potential of stochastic activation layers for UQ was briefly discussed in Lee et al. (2019), but the context of this work was classification tasks, where the predictive uncertainty could already be fully characterized by well-calibrated categorical distributions without using any latent noise. More recently, Sun and Liang (2022) formulated DNNs as latent variable models and included kernel maps in the input layer to avoid feature collinearity. Although the proposed model is capable of UQ, the more challenging problem of DR has not been studied.

1.2 | Our contributions

To address the challenges for DR with DNNs, we propose the Bayesian deep noise neural network (B-DeepNoise). B-DeepNoise generalizes standard BNNs by adding latent random noise both before and after every activation layer. Although the latent random noise variables independently follow Gaussian distributions, their composition across multiple layers with nonlinear activations generates highly complex predictive density functions. Moreover, the unique structure of B-DeepNoise induces closed-form posterior full conditional distributions for the model parameters, which eliminates the primary barrier for Gibbs sampling in DNN-based models and therefore makes it possible to simulate the exact posterior distribution without using tuning-intensive MH algorithms.

To our best knowledge, this is the first work on estimating complex predictive density functions by utilizing DNNs with latent random noise. Furthermore, no previous work has developed Gibbs sampling algorithms for DNN-based Bayesian predictive models. In short, our work contributes to the existing literature on DR and UQ with DNNs in the following ways: (1) We propose a Bayesian DNN model for learning complex, non-Gaussian predictive density functions. (2) We develop a virtually tuning-free Gibbs sampling algorithm for posterior computation that uses common samplers only, without the need of MH steps. (3) We perform theoretical analysis for analytic expressions of the predictive densities and variance propagation. (4) We evaluate our model on multiple benchmark datasets and demonstrate its usefulness in a neuroimaging study. The codes for our method is available at github.com/daviddaiweizhang/B-DeepNoise.

2 | MODEL DESCRIPTION

2.1 | DNNs with latent noise variables

Suppose the data consist of features $\boldsymbol{x}^{(n)} \in \mathbb{R}^P$ and outcomes $\boldsymbol{y}^{(n)} \in \mathbb{R}^Q$, with $n \in 1,\ldots,N$.

Let $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ be a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. To specify the nonlinear association between $\boldsymbol{y}^{(n)}$ and $\boldsymbol{x}^{(n)}$, a standard L-layer feed-forward DNN model with Gaussian noise can be represented as

$$\boldsymbol{y}^{(n)} \Big| \boldsymbol{u}_L^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}\left(\boldsymbol{\beta}_L \boldsymbol{u}_L^{(n)} + \boldsymbol{\gamma}_L,\ \boldsymbol{T}_L\right) \tag{1}$$

$$\boldsymbol{u}_{l+1}^{(n)} = h\left(\boldsymbol{\beta}_l \boldsymbol{u}_l^{(n)} + \boldsymbol{\gamma}_l\right),\ l \in 0,\ldots,L-1 \tag{2}$$

$$\boldsymbol{u}_0^{(n)} = \boldsymbol{x}^{(n)} \tag{3}$$

where $\boldsymbol{\beta}_l \in \mathbb{R}^{K_l \times K_{l-1}}$ and $\boldsymbol{\gamma}_l \in \mathbb{R}^{K_l}$ are unknown parameters, K_l is the number of units in the l^{th} layer, and $h(\cdot)$ is an element-wise nonlinear activation function.

In this formulation, $\boldsymbol{u}_L^{(n)}$ is a deterministic function of $\boldsymbol{x}^{(n)}$, which implies that $\boldsymbol{y}^{(n)}|\boldsymbol{x}^{(n)}$ and $\boldsymbol{y}^{(n)}|\boldsymbol{u}_L^{(n)}$ follow the same homoscedastic Gaussian distribution with constant covariance $\boldsymbol{T}_L$.

To model more complex conditional distributions, we propose the deep noise neural network (DeepNoise), which generalizes Equation (2) of the standard DNN model into Equations (4) and (5) by including noise variables before and after every activation layer:

$$\boldsymbol{u}_{l+1}^{(n)} = h\left(\boldsymbol{\beta}_l \boldsymbol{u}_l^{(n)} + \boldsymbol{\gamma}_l + \boldsymbol{\epsilon}_l^{(n)}\right) + \boldsymbol{\delta}_l^{(n)} \tag{4}$$

$$\boldsymbol{\epsilon}_l^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0},\boldsymbol{T}_l),\ \boldsymbol{\delta}_l^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_l),\ l \in 0,\ldots,L-1 \tag{5}$$

where $\boldsymbol{T}_l = \text{diag}\{\boldsymbol{\tau}_l^2\}$ and $\boldsymbol{\Sigma}_l = \text{diag}\{\boldsymbol{\sigma}_l^2\}$ with $\boldsymbol{\tau}^2 = \left(\tau_{l,1}^2,\ldots,\tau_{l,K_l}^2\right)^\top$ and $\boldsymbol{\sigma}_l^2 = \left(\sigma_{l,1}^2,\ldots,\sigma_{l,K_l}^2\right)^\top$. By composing the latent Gaussian noise variables with linear maps and nonlinear activations, DeepNoise is capable of representing a wide range of heteroscedastic Gaussian and non-Gaussian conditional density functions (e.g., asymmetric and multimodal), as illustrated in Figure 1. Intuitively, as Gaussian mixtures are universal approximators of densities (Calcaterra & Boldt, 2008; Goodfellow et al., 2016, [Sec. 3.9.6]; Plataniotis & Hatzinakos, 2017) and DNNs are universal approximators of functions (Lu & Lu, 2020; Scarselli & Tsoi, 1998; Yarotsky, 2017), DeepNoise is designed to be a universal approximator of conditional densities. The nonparametric nature of DeepNoise enables it to approximate increasingly complex conditional density functions by increasing the number of hidden layers and the number of nodes per layer. In the special case where the variance of the noise variables is set to zero in all but the output layer, DeepNoise is reduced to a standard DNN.

2.2 | Model representation and prior specifications

DeepNoise transforms the predictor vector into the outcome vector by iteratively and stochastically applying the linear-noise-nonlinear-noise maps. Thus, the DeepNoise model defined by combining Equations (1) and (3) to (5) is equivalent to

$$\boldsymbol{v}_l^{(n)} \Big| \boldsymbol{u}_l^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}\left(\boldsymbol{\beta}_l \boldsymbol{u}_l^{(n)} + \boldsymbol{\gamma}_l,\ \boldsymbol{T}_l\right),\ l \in 0,\ldots,L \tag{6}$$

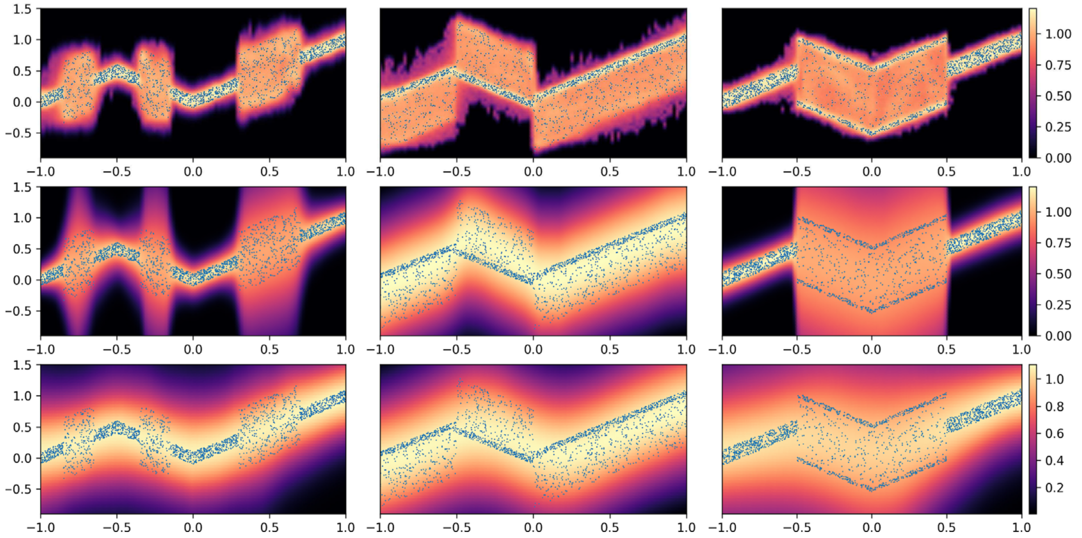


FIGURE 1 Observed data (blue dots) and estimated predictive density (heatmap) by B-DeepNoise (top row), DE (middle row), and BNN (bottom row) for heteroscedastic (left column), asymmetric (middle column), and multimodal (right column) noise.

$$\mathbf{u}_{l+1}^{(n)} \Big| \mathbf{v}_l^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}\left(h\left(\mathbf{v}_l^{(n)}\right),\ \boldsymbol{\Sigma}_l\right), l \in 0,\ldots,L-1 \tag{7}$$

$$\mathbf{u}_0^{(n)} = \mathbf{x}^{(n)},\ \mathbf{v}_L^{(n)} = \mathbf{y}^{(n)} \tag{8}$$

Write $\boldsymbol{\beta}_l = (\beta_{l,k,k'})$ and $\boldsymbol{\gamma}_l = (\gamma_{l,k})$. To make fully Bayesian inference, we impose normal-inverse-gamma prior distributions on the weight-bias parameters:

$$\begin{array}{ll} \beta_{l,k,k'} \overset{\text{iid}}{\sim} \mathcal{N}(0, \rho_{l,k,k'}^2), & \gamma_{l,k} \overset{\text{iid}}{\sim} \mathcal{N}(0, \xi_{l,k}^2) \\ \rho_{l,k,k'}^2 \overset{\text{iid}}{\sim} \mathcal{IG}(a,b) & \xi_{l,k}^2 \overset{\text{iid}}{\sim} \mathcal{IG}(a,b) \end{array}$$

and assign inverse-gamma prior distributions to the pre-activation and post-activation noise variances:

$$\tau_{l,k}^2 \overset{\text{iid}}{\sim} \mathcal{IG}(a,b), \quad \sigma_{l,k}^2 \overset{\text{iid}}{\sim} \mathcal{IG}(a,b)$$

To use weakly informative priors, we set $a = b = 0.001$. We refer to Equations (6) to (8) along with the prior specifications above as the B-DeepNoise model. Although DeepNoise (i.e., Equations (6) to (8) without the prior distributions) is an effective frequentist model for DR, for the rest of the paper, we focus on B-DeepNoise and conduct theoretical, computational, and empirical analyses in the Bayesian framework.

2.3 | Posterior computation

Compared to standard DNNs and BNNs, the addition of latent random noise not only makes B-DeepNoise more flexible in approximating complex predictive density functions, but it also provides closed-form expressions of the posterior full conditional distributions of the model parameters. The latter advantage makes it possible to derive efficient Gibbs sampling algorithms for B-DeepNoise. In order for a Gibbs sampler to be computationally efficient, a key requirement is that all the posterior full conditional distributions are easy to simulate. To this end, we require the activation function to be piecewise linear, as described in Assumption 1.

Assumption 1. The element-wise activation function h can be expressed as $h(t) = \sum_{j=1}^{J} (b_j t + b_j') \cdot \mathbb{I}\{t \in [c_{j-1}, c_j)\}$ for some $J \in \mathbb{N}_+$, $b_1, \ldots, b_J, b_1', \ldots, b_J' \in \mathbb{R}$, and $-\infty = c_0 < c_1 < \ldots < c_{J-1} < c_J = \infty$.

Remark 1. The family of functions defined in Assumption 1 includes many common activation functions, such as ReLU and leaky ReLU (Maas et al., 2013). Moreover, smooth activation functions can be approximated by piecewise linear functions. For example, the logistic, tanh, and softplus functions can be approximated by the hard sigmoid, hard tanh, and ReLU functions, respectively.

We now derive the posterior full conditional distributions of all the model parameters in B-DeepNoise. Let $\mathcal{TN}_{a,b}(\lambda,\omega^2)$ be a truncated normal distribution on interval $[a,b]$ with location λ and scale ω.

Theorem 1. Suppose the activation function h satisfies Assumption 1. The model parameters in B-DeepNoise have the following posterior full conditional distributions.

$$v_{l,k}^{(n)}\Big|\text{rest} \sim \sum_{j=1}^{J} \pi_{l,k,j}^{(n)} \cdot \mathcal{TN}_{c_{j-1},c_j}\left(\lambda_{l,k,j}^{(n)},\omega_{l,k,j}^2\right) \tag{9}$$

$$\boldsymbol{u}_l^{(n)}\Big|\text{rest} \sim \mathcal{N}\left(\boldsymbol{\mu}_l^{(n)},\boldsymbol{U}_l^{(n)}\right) \tag{10}$$

$$(\boldsymbol{\beta}_{l,k},\gamma_{l,k})\big|\text{rest} \sim \mathcal{N}(\boldsymbol{\eta}_{l,k},\boldsymbol{B}_{l,k}) \tag{11}$$

$$\tau_{l,k}^2\Big|\text{rest} \sim \mathcal{IG}\left(a+0.5N,\ b+0.5\|\hat{\epsilon}_{l,k}\|_2^2\right) \tag{12}$$

$$\sigma_{l,k}^2\Big|\text{rest} \sim \mathcal{IG}\left(a+0.5N,\ b+0.5\|\hat{\boldsymbol{\delta}}_{l,k}\|_2^2\right) \tag{13}$$

$$\rho_{l,k,k'}^2\Big|\text{rest} \sim \mathcal{IG}\left(a+0.5,\ b+0.5\beta_{l,k,k'}^2\right) \tag{14}$$

$$\xi_{l,k}^2\big|\text{rest} \sim \mathcal{IG}(a+0.5,\ b+0.5\gamma_{l,k}^2) \tag{15}$$

where $\pi_{l,k,j}^{(n)}$, $\lambda_{l,k,j}^{(n)}$, $\boldsymbol{\mu}_l^{(n)}$, $\boldsymbol{U}_l^{(n)}$, $\boldsymbol{\eta}_{l,k}$, $\boldsymbol{B}_{l,k}$, $\hat{\epsilon}_{l,k}$, $\hat{\boldsymbol{\delta}}_{l,k}$ are defined in Section S1.1.1.

Remark 2. Theorem 1 shows that the posterior full conditional distribution of every model parameter in B-DeepNoise is either inverse gamma, normal, or mixture of truncated normal. Equations (10) to (15) follow properties of Bayesian linear regression with conjugate priors (Bishop & Nasrabadi, 2006, Sec. 2.3.3). Equation (9) describes the most complicated block of parameters, $v_{l,k}^{(n)}$, because the nonlinear activation function is involved. Intuitively, the piecewise linear property of the activation function causes the posterior full conditional distribution of $\boldsymbol{v}_l^{(n)}$ to be "piecewise normal", that is, a mixture of truncated normal distributions with adjacent truncation endpoints. Thus, sampling $\boldsymbol{v}_l^{(n)}$ only requires samplers for categorical distributions and truncated normal distributions, which are widely available in scientific computation libraries. In addition, the number of the mixing components is very small for common activation functions (e.g., 2 for ReLU and 3 for hard tanh).

The Gibbs sampler for computing the posterior distributions of B-DeepNoise parameters consist of applying Equations (9) to (15) iteratively, as described in Algorithm A1, Appendix A. Because the predictive distribution only depends on the weight-bias parameters and the latent noise variance parameters, the posterior samples of the latent variables do not need to be stored, which saves memory in practice. In addition, to reduce burn-in time, model parameters can be initialized using gradient-based optimizers or pre-trained weights.

B-DeepNoise and BNN have the same posterior computation time complexity: $\mathcal{O}(LK^2N)$, where L is the network depth, K is the maximal network width, and N is the training sample size. Although the introduction of latent random variables in B-DeepNoise increases the time complexity by a constant term, the extra computation price is paid for more information on the outcomes. Unlike BNNs, which only estimate the predictive mean and variance of every input, B-DeepNoise estimates the predictive density itself.

For the choice of posterior sampling algorithms, B-DeepNoise has the options of using Gibbs samplers or MH samplers, while BNN can use MH samplers only, due to the lack of closed-form posterior full conditional distributions. For hyperparameter tuning, our Gibbs sampler for

B-DeepNoise is tuning-free, while gradient-based MH algorithms for BNN, such as Hamiltonian Monte Carlo (HMC), are sensitive to the choice of the integration step size and the number of steps (Hoffman et al., 2014). For scalability, the Gibbs sampling algorithm for B-DeepNoise naturally allows the usage of mini-batches of the whole training dataset, because Gibbs samplers allow arbitrary partitions of the model parameters. That is, at every sampling step, when sampling the latent noise variables (Equations 9 and 10), instead of updating $v_{l,k}^{(n)}$ and $\boldsymbol{u}_l^{(n)}$ for the entire training set, we only update them for a mini-batch of the training samples. (See Algorithm A1 for details.) By contrast, although efforts have been made in developing mini-batch MH methods (Wu et al., 2020), computing the log-likelihood function using subsets of the training samples inevitably introduce extra noise and causes the algorithm's stationary distribution to deviate from the target distribution. In addition, whereas the forward and backward propagation steps in MH algorithms for standard BNNs must be computed sequentially by network layers, our Gibbs sampler for B-DeepNoise allows for sampling model parameters parallelly across the layers, since given the parameters in the odd layers, the parameters in the even layers are conditionally independent (and vice versa), thus allowing for simultaneous, parallel updates. (See Algorithm A1 for details.)

2.4 | Predictive density

We further evaluate the properties of the predictive density function. Theorem 2 expresses the predictive density in a recursive formulation. $\phi_{\mathcal{MVN}}(\cdot|\boldsymbol{\mu},\boldsymbol{\Sigma})$ be a multivariate normal density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Theorem 2. For $l \in \{0,...,L\}$, let $\boldsymbol{\theta}_l = \{\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l,\boldsymbol{T}_l,\boldsymbol{\Sigma}_l\}$ and $\boldsymbol{\Theta}_l = \{\boldsymbol{\theta}_{l'}\}_{l'=0}^{l}$. In a B-DeepNoise model, the conditional density of the output $\boldsymbol{y} = \boldsymbol{v}_L$ given input value $\boldsymbol{x}$ and model parameters $\boldsymbol{\Theta}_l$ can be iteratively constructed by

$$f(\mathbf{v}_l|\mathbf{x},\boldsymbol{\Theta}_l) = \int_{\mathbb{R}^{K_{l-1}}} \phi_{\mathcal{MVN}}(\mathbf{v}_l|\boldsymbol{\mu}_l,\boldsymbol{S}_l) f(\mathbf{v}_{l-1}|\mathbf{x},\boldsymbol{\Theta}_{l-1}) d\mathbf{v}_{l-1}$$

where $\boldsymbol{\mu}_l = \boldsymbol{\gamma}_l + \boldsymbol{\beta}_l h(\mathbf{v}_{l-1})$ and $\boldsymbol{S}_l = \boldsymbol{T}_l + \boldsymbol{\beta}_l \boldsymbol{\Sigma}_l \boldsymbol{\beta}_l^\top$ for $l \in \{1,...,L\}$, and

$$f(\mathbf{v}_0|\mathbf{x},\boldsymbol{\theta}_0) = \phi_{\mathcal{MVN}}(\mathbf{v}_0|\boldsymbol{\gamma}_0 + \boldsymbol{\beta}_0 \mathbf{x},\ \boldsymbol{T}_0)$$

Remark 3. Theorem 2 shows that the predictive density given an input value can be expressed as a continuous mixture of multivariate normal density (CMMVN), where the mixing density is a CMMVN over the output values of the previous layer. Although this highly flexible predictive density does not have a closed-form expression, it can be simulated easily by adding normal noise to the intermediate values of the hidden layers, as stated in Equations (6) and (7).

In standard DNNs and BNNs, the outcome variable follow a normal distribution with the variance equal to by the variance of the outcome noise variable. As a more general model, B-DeepNoise propagates variations in the latent noise variables to produce a complex distribution in the outcome. When the variances of the latent random noise variables are all zero, B-DeepNoise is reduced to a standard BNN. A natural question is how the variance in the output variable can be decomposed by variances of the latent noises variables. To this end, Theorem 3 bounds the outcome variance by the other model parameters.

Theorem 3. Let $\boldsymbol{y}|\mathbf{x},\boldsymbol{\Theta}$ be the output value of the B-DeepNoise model given input value $\boldsymbol{x}$ and model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l,\boldsymbol{T}_l,\boldsymbol{\Sigma}_l\}_{l=0}^{L}$. Let $g(\mathbf{x},\boldsymbol{\Gamma})$ be the output value of the standard DNN model with the same activation function and weight-bias parameters $\boldsymbol{\Gamma} = \{\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l\}_{l=0}^{L}$. (Note that $\boldsymbol{y}$ equals to $g(\mathbf{x},\boldsymbol{\Gamma})$ with probability one when all the latent noise variances are zero.) Assume the activation function h is Lipschitz continuous with Lipschitz constant C_h, and define $d_l^2 = \|\boldsymbol{\beta}_l\|_2^2$. Then

$$\mathrm{Var}(\boldsymbol{y}|\mathbf{x},\boldsymbol{\Theta}) + [\mathrm{E}(\boldsymbol{y}|\mathbf{x},\boldsymbol{\Theta}) - g(\mathbf{x},\boldsymbol{\Gamma})]^2 \leq \sum_{l=0}^{L} \left[d_l^2 \sum_{k=1}^{K_{l-1}} \sigma_{l-1,k}^2 + \sum_{k=1}^{K_l} \tau_{l,k}^2 \right] \left[\prod_{l'=l+1}^{L} d_{l'}^2 \right] C_h^{2(L-l)}$$

Remark 4. According to Theorem 3, given the model parameters, the predictive variance of B-DeepNoise is bounded by the latent noise variances, the spectrum norm of the weight matrices, and the Lipschitz constant of the activation function. In addition, the same upper bound holds for the squared distance of B-DeepNoise's predictive mean from the corresponding deterministic DNN's output value. The expression of this bound can be simplified for common activation functions using global bounds of model parameters, as shown in Theorem 1.

Corollary 1. Let $K = \max_{-1 \le l \le L} K_l$, $d^2 = \max_{0 \le l \le L} \|\boldsymbol{\beta}_l\|_2^2$, $\sigma^2 = \max_{0 \le l \le L-1} \|\boldsymbol{\sigma}_l^2\|_\infty$, $\tau^2 = \max_{0 \le l \le L} \|\boldsymbol{\tau}_l^2\|_\infty$.
Suppose activation function h is ReLU, leaky ReLU, hard sigmoid, or hard tanh. Then

$$\mathrm{Var}(\boldsymbol{y}|\mathbf{x},\boldsymbol{\Theta}) + [\mathrm{E}(\boldsymbol{y}|\mathbf{x},\boldsymbol{\Theta}) - g(\mathbf{x},\boldsymbol{\Gamma})]^2 \quad \le 3KL\left(d^{2L+2}+1\right)(\sigma^2+\tau^2)$$

Proofs of all theoretical results are provided in Section S1.2.

3 | METHOD COMPARISON

We applied B-DeepNoise and existing methods to synthetic and real data to evaluate their accuracy for predictive density estimation and uncertainty quantification.

3.1 | Experiments on synthetic continuous data

We used synthetic datasets to compare the predictive density estimated by each method with the ground truth. The input variable x was one-dimensional and uniformly distributed on $[-1,1]$. The output variable y was also one-dimensional, and its conditional median was a linear spline with respect to x. We designed the noise distribution to be heteroscedastic, asymmetric, or multimodal, as shown by the training data points (blue dots) in Figure 1. The training sample size varied among 1000, 2000, and 4000. Every experimental setting was repeated for 20 times.

We used a B-DeepNoise model with four hidden layers and 50 nodes in each layer, with the hard tanh function as the activation function. The prior distributions of the variance parameters were set to $\mathcal{IG}(0.001,0.001)$. We used gradient descent to initialize the model parameters and drew 500 posterior samples. B-DeepNoise was compared against backpropagation (BP), variational inference (VI) (Ritter & Karaletsos, 2022), Bayesian neural networks (BNN) with HMC (Neal, 2011), and deep ensemble (DE) (Lakshminarayanan et al., 2016). The baseline methods used identical architecture as B-DeepNoise, and the hyperparameters were selected according to the original authors' recommendations. Details of the experiments are described in Section S2.

As visualized in Figure 1, B-DeepNoise successfully captured key characteristics of the noise densities. The estimated predictive distributions identified variation in the output variance, opposite directions of skewedness, and abrupt changes between unimodal to bimodal distributions. In contrast, predictive densities estimated by the baseline methods were all unimodal and symmetric.

To quantitatively evaluate the accuracy of the estimated predictive density functions, we numerically computed the L_1 distance between the inverse CDFs of the true and estimated predictive distributions. This metric is equivalent to the simultaneous quantile loss (Tagasovska & Lopez-Paz, 2019), which measures the distance between two distributions by considering all quantiles jointly and is applicable to arbitrary distributions, including heteroscedastic, skewed, and multimodal ones. It is also generalizable to multivariate density functions by using multivariate CDFs. The estimation errors are reported in Table 1. Among all the methods, B-DeepNoise had the smallest error in all but two settings. Moreover, for all the

TABLE 1 Predictive density estimation error (in unit of 0.001) on synthetic datasets for various noise densities and training sizes.

	heteroscedastic noise			skewed noise			multimodal noise		
Method	$N=1000$	$N=2000$	$N=4000$	$N=1000$	$N=2000$	$N=4000$	$N=1000$	$N=2000$	$N=4000$
BP	52 ± 6	40 ± 6	33 ± 2	100 ± 1	99 ± 2	98 ± 2	82 ± 8	78 ± 2	78 ± 2
VI	103 ± 1	103 ± 1	101 ± 1	103 ± 2	102 ± 1	98 ± 1	173 ± 2	175 ± 1	175 ± 0
BNN	103 ± 2	102 ± 2	95 ± 2	105 ± 2	104 ± 2	101 ± 1	128 ± 8	128 ± 6	122 ± 6
DE	**49 ± 4**	39 ± 4	32 ± 2	101 ± 1	100 ± 3	97 ± 1	**82 ± 1**	78 ± 1	78 ± 1
B-DeepNoise	72 ± 4	**38 ± 3**	**26 ± 2**	**84 ± 6**	**53 ± 7**	**39 ± 3**	98 ± 6	**60 ± 4**	**47 ± 5**

three types of noise, B-DeepNoise's accuracy improved much faster than the baseline methods as the training sample size increased, especially on the skewed and multimodal data. These simulations illustrate the accuracy of B-DeepNoise in learning complex predictive density functions. For uncertainty quantification, since uncertainty quantification is measured by summary statistics of the predictive density (e.g., variance, 95% prediction intervals), the superior performance of B-DeepNoise in predictive density estimation implies its superior performance in uncertainty quantification accuracy.

The computation efficiency of posterior sampling methods hinges crucially on fast convergence rates and well-mixed behaviors. It is worth noting that the optimal choice of such a method can be both model-specific and data-dependent (Papaspiliopoulos & Roberts, 2008). In this regard, B-DeepNoise has an edge over conventional BNNs thanks to its compatibility with both Metropolis Hastings (MH) algorithms and Gibbs samplers, thereby offering a broader range of options for choosing the most suitable sampling method. In our simulations, we compared the convergence of the proposed Gibbs sampler on the B-DeepNoise model with HMC on the B-DeepNoise model. The trace plots of the negative log-likelihood (Figure S2) highlight that, compared to B-DeepNoise + HMC, B-DeepNoise + Gibbs converged within fewer steps and reached lower negative log-likelihood values, which indicates a higher per-step efficiency and more accurate posterior predictive densities. Despite the superior empirical convergence performance of the Gibbs sampler, it remains essential to be mindful of the potential theoretical challenges with Gibbs samplers that can lead to inadequate mixing behaviors (Johnson & Jones, 2015; Papaspiliopoulos & Roberts, 2008; Román & Hobert, 2012). For example, when the model parameters are highly correlated or when the noise variance is small, Gibbs samplers may be less efficient than MH algorithms. While a comprehensive discussion on the theoretical issues surrounding Gibbs samplers are beyond the scope of this paper, we recommend in practice the identification of the most effective sampling methods by examining the effective sample sizes and inspecting the trace plots of the negative log-likelihood.

3.2 | Experiments on real continuous data

We applied B-DeepNoise and baselines methods to nine regression datasets on the UCI Machine Learning Repository (Dua & Graff, 2017). Experiment setup was similar to (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2016), where each dataset was randomly split into training and testing sets for five times on Protein Structure and 20 times in the other datasets. Prediction accuracy was measured by the testing root mean squared error (RMSE) of the predictive mean, and uncertainty quantification accuracy was measured by the average negative log likelihood (NLL) of the predictive density on the testing data, as in Hernández-Lobato and Adams (2015), Gal and Ghahramani (2016), and Lakshminarayanan et al. (2016). The NLL is equivalent to the empirical cross-entropy between the estimated and the true predictive density functions, which penalizes both over- and under-confident predictions. See Quinonero-Candela et al. (2005) for NLL as a valid metric and a proper scoring rule for evaluating predictive uncertainty. Another common UQ accuracy metric is the width of the 95% uncalibrated prediction interval (UPI) (Pearce et al., 2018). However, due to model miscalibration, the 95% UPI might overcover or undercover 95% of the observed outcomes. To better measure UQ accuracy, we computed the 95% calibrated prediction intervals (CPI), defined as the minimal x% UPI that covers at least 95% of the observed outcomes. In other words, the 95% CPI is the miscalibration-adjusted version of the 95% UPI. (See Section S3.1.3 for details.) Using the three aforementioned metrics, we compared B-DeepNoise against BP, VI, BNN, DE, and dropout Monte Carlo (DMC) (Gal & Ghahramani, 2016). Network architectures and experiment setups are similar to those in Section 3.1. (See Section S3 for details.)

Table 2 shows the experiment results. Compared with the baseline methods, B-DeepNoise had the least RMSE on all except two datasets, which indicates a superior prediction accuracy. Moreover, B-DeepNoise's predictive distribution was also overall more accurate than the other methods, since the NLL of B-DeepNoise was the smallest on all except two datasets. The high UQ accuracy of B-DeepNoise was further demonstrated by its uniformly narrowest WCPI-95. In comparison, the baseline methods not only had significantly larger WCPI-95s, but some methods could not produce valid 95% CPIs at all on some of the datasets (as indicated by ∞ in Table 2), because even their 100% prediction intervals could not cover at least 95% of the observed testing outcomes.

In addition, B-DeepNoise's performance was overall more stable than the baseline methods, as reflected by its generally smaller standard errors with respect to all the metrics. In short, the experiments on the UCI datasets demonstrated the superior DR and UQ accuracy of B-DeepNoise compared to existing methods.

For computation efficiency, the total runtimes of all methods are reported in Table S2. Overall, the computation cost of B-DeepNoise is comparable with BNN.

4 | NEUROIMAGING-BASED PREDICTION OF GENERAL INTELLIGENCE FOR ADOLESCENTS

We demonstrate the usefulness of B-DeepNoise in scientific studies by applying it to the neuroimaging data in the Adolescent Brain Cognitive Development (ABCD) Study (Casey et al., 2018). The dataset contains 1191 subjects recruited from multiple study sites in the United States. For every subject, we use B-DeepNoise to predict the density of the general intelligence score (g-score) (Dubois et al., 2018; O'Shea et al., 2016) using

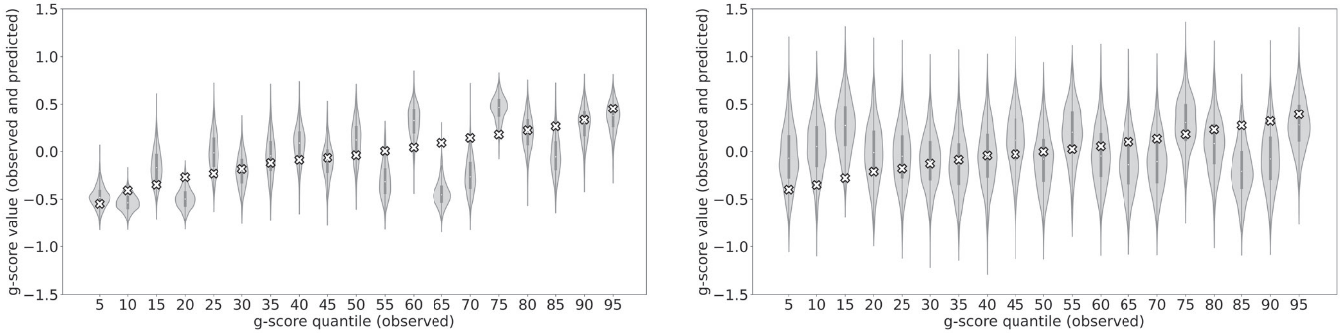
TABLE 2 Experiment results for nine real datasets from the UCI Machine Learning Repository.

Dataset	BP	VI	BNN	DMC	DE	B-DeepNoise
Root mean squared error (RMSE)						
Yacht Hydrodynamics	3.09 ± 1.28	2.10 ± 0.43	2.95 ± 0.01	3.30 ± 1.14	2.02 ± 0.76	**0.64** ± **0.32**
Boston Housing	3.36 ± 1.05	2.75 ± 0.67	**2.32** ± **0.00**	3.10 ± 0.88	3.16 ± 1.11	2.84 ± 0.69
Energy Efficiency	2.45 ± 0.32	0.68 ± 0.09	1.00 ± 0.00	1.46 ± 0.18	2.56 ± 0.32	**0.45** ± **0.07**
Concrete Strength	5.98 ± 0.62	4.68 ± 0.52	7.09 ± 1.95	6.11 ± 0.47	5.45 ± 0.55	**4.54** ± **0.46**
Wine Quality	0.64 ± 0.05	0.66 ± 0.07	**0.62** ± **0.03**	**0.62** ± **0.04**	**0.62** ± **0.04**	0.63 ± 0.04
Kin8nm	0.08 ± 0.00	0.09 ± 0.00	**0.07** ± **0.00**	2.27 ± 0.23	**0.07** ± **0.00**	**0.07** ± **0.00**
Power Plant	4.02 ± 0.15	3.89 ± 0.20	4.27 ± 0.12	4.12 ± 0.15	3.98 ± 0.15	**3.62** ± **0.18**
Naval Propulsion	**0.00** ± **0.00**	**0.00** ± **0.00**	0.02 ± 0.00	509.93 ± 0.00	**0.00** ± **0.00**	**0.00** ± **0.00**
Protein Structure	4.08 ± 0.06	4.35 ± 0.09	4.93 ± 0.27	4.02 ± 0.04	3.91 ± 0.03	**3.64** ± **0.03**
Negative log likelihood (NLL)						
Yacht Hydrodynamics	1.39 ± 0.33	2.64 ± 0.04	1.85 ± 0.00	2.29 ± 1.14	1.09 ± 0.19	**0.45** ± **0.22**
Boston Housing	3.01 ± 0.90	2.40 ± 0.12	**2.28** ± **0.00**	2.42 ± 0.88	2.37 ± 0.27	**2.28** ± **0.17**
Energy Efficiency	1.77 ± 0.59	1.36 ± 0.06	1.46 ± 0.00	1.79 ± 0.18	1.54 ± 0.25	**0.55** ± **0.19**
Concrete Strength	3.41 ± 0.47	3.03 ± 0.21	3.18 ± 0.04	3.19 ± 0.47	3.04 ± 0.22	**2.84** ± **0.14**
Wine Quality	2.07 ± 0.86	10.05 ± 4.03	1.03 ± 0.43	**0.92** ± **0.04**	1.03 ± 0.25	0.95 ± 0.09
Kin8nm	−1.01 ± 0.21	1.67 ± 0.44	−1.17 ± 0.06	−0.92 ± 0.02	−1.30 ± 0.04	−**1.31** ± **0.04**
Power Plant	2.81 ± 0.06	2.92 ± 0.10	2.86 ± 0.03	2.80 ± 0.03	2.77 ± 0.05	**2.70** ± **0.06**
Naval Propulsion	−4.67 ± 1.29	−7.19 ± 0.48	−5.71 ± 0.37	−4.10 ± 0.03	−5.15 ± 0.21	−**7.25** ± **0.07**
Protein Structure	2.80 ± 0.19	3.25 ± 0.03	2.95 ± 0.09	2.77 ± 0.01	**2.54** ± **0.05**	2.65 ± 0.09
Width of 95% empirical prediction intervals (WCPI-95)						
Yacht Hydrodynamics	2.63 ± 1.17	6.01 ± 1.72	6.41 ± 0.08	∞	2.34 ± 0.91	**1.26** ± **0.90**
Boston Housing	∞	9.81 ± 1.60	11.20 ± 0.11	10.51 ± 1.90	∞	**9.06** ± **1.91**
Energy Efficiency	3.70 ± 1.05	2.60 ± 0.46	3.55 ± 0.03	6.28 ± 1.14	3.55 ± 0.79	**1.65** ± **0.32**
Concrete Strength	∞	∞	27.96 ± 8.00	25.45 ± 2.68	17.40 ± 3.02	**16.00** ± **2.29**
Wine Quality	∞	∞	∞	∞	2.88 ± 0.34	**2.48** ± **0.25**
Kin8nm	0.30 ± 0.02	∞	0.31 ± 0.02	11.24 ± 1.11	**0.26** ± **0.01**	**0.26** ± **0.01**
Power Plant	14.89 ± 0.52	15.28 ± 0.76	16.15 ± 0.91	∞	14.76 ± 0.56	**13.90** ± **0.62**
Naval Propulsion	∞	**0.00** ± **0.00**	**0.00** ± **0.00**	2548.88 ± 0.00	**0.00** ± **0.00**	**0.00** ± **0.00**
Protein Structure	16.32 ± 0.63	∞	17.65 ± 0.81	∞	14.06 ± 0.37	**11.90** ± **0.19**

the 2-back task score (Cohen et al., 2016), general psychopathology factor (Caspi et al., 2014; Carver et al., 2017; Murray et al., 2016), demographic information (age, sex, parental education level, household marital status, household income, and ethnic backgrounds), and brain functional magnetic resonance imagings (fMRIs). (See Casey et al., 2018; Zhang et al., 2020, for details of the ABCD data.) We randomly selected 90% of the samples for training and the rest for testing. The model hyperparameters are similar to those in Sections 3.1 and 3.2. See Section S4 for details of the experiment setup.

To illustrate the predictive density functions learned by B-DeepNoise, we selected 19 testing subjects that correspond to the 5%, 10%, …, 90%, 95% quantiles of the observed g-score. The results are shown in Figure 1a. The prediction distributions estimated by B-DeepNoise have successfully covered the observed outcomes, with only a couple of samples located near the tails of the predictive distributions. To further assess B-DeepNoise's UQ accuracy, we removed the imaging predictors and refit the model with the non-imaging predictors only. As shown in Figure 2b, when the imaging information was not available, B-DeepNoise widened the prediction intervals to account for the higher degree of uncertainty. In contrast, the predictive densities in the imaging-included model are not only more concentrated but also exhibited greater magnitude of heteroscedasticity and skewedness. These results indicate that B-DeepNoise is able to appropriately adjust the predictive density to reflect its subject-level prediction confidence.

Furthermore, we investigated the most influential neuroimaging features on the predictive mean of the g-score, where influence is measured by the average absolute value of the gradient of the predictive mean with respect to the feature in the B-DeepNoise model.



(a) Imaging features included as predictors

(b) Imaging features excluded as predictors

FIGURE 2 Observed g-scores (white crosses) and predictive densities (violin plots) estimated by B-DeepNoise for 19 testing subjects.

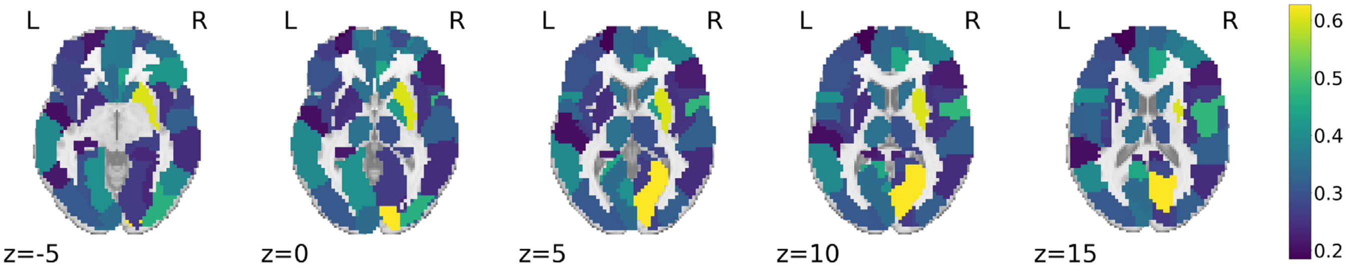


FIGURE 3 Influence of brain regions on the predictive mean of the g-score.

As shown in Table S3, the 2-back task score had the highest influence (1.32) on the predicted mean g-score, and the magnitude of influence was much higher than the other features (0.63 or less). This result is consistent with the current understanding that memory is a one of the major components that encompass cognitive abilities (Thompson et al., 2019). The rest of the most influential features were primarily imaging features, and Figure 3 visualizes the influence of brain regions (Tzourio-Mazoyer et al., 2002) on predicted g-scores. Many of the corresponding brain regions have been well studied for their associations with general intelligence in existing studies. To name a few, bilateral calcarine is associated with the intelligence quotient (IQ) of children and adolescents (Kilroy et al., 2011); putamen has been identified with verbal IQ in healthy adults (Grazioplene et al., 2015); the right paracentral lobule has been found to be associated with functioning decline in at-risk mental state patients (Sasabayashi et al., 2021). Overall, neuroimaging regions that are the most influential for B-DeepNoise's predicted mean g-score are supported by existing findings in the literature. Although causes for individual differences in cognitive abilities are multifaceted (Dubois et al., 2018), and our analysis does not make causal claims on the relationships between the features and the outcomes, B-DeepNoise's ability of providing predictive densities of the g-score makes the model useful for studies that involve quantification of uncertainty in general intellectual capacity.

5 | CONCLUSION

In this work, we have presented B-DeepNoise, a novel Bayesian model based on deep neural networks for density regression, which generalizes the task of uncertainty quantification. We have demonstrated our model's theoretical and computational properties and evaluated its performance on synthetic and real regression data.

A limitation of B-DeepNoise is the lack of advantage on classification tasks. Although B-DeepNoise combined with a softmax activation in the output layer is capable of handling categorical outcomes (Section S5.2), it (or any other stochastic models) is not expected theoretically to have superior uncertainty quantification accuracy than a standard DNN classifier. See Section S5.1 for a demonstration.

For future works, we are interested in developing frequentist versions of our method and exploring the possibility of using them for outcome selection.

ACKNOWLEDGEMENTS

This research was partially supported by NIH grants NIH R01MH105561 (D.Z and J.K.), NIH R01DA048993 (J.K.), NIH R01GM124061 (J.K.), and NSF grant IIS2123777 (J.K.).

ORCID

Daiwei Zhang https://orcid.org/0000-0002-5019-622X

REFERENCES

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., & Makarenkov, V. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in ai safety. arXiv preprint arXiv:1606.06565.

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, *18*(4), 343–373.

Begoli, E., Bhattacharya, T., & Kusnezov, D. (2019). The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, *1*(1), 20–23.

Berner, J., Grohs, P., Kutyniok, G., & Petersen, P. (2021). The modern mathematics of deep learning. arXiv preprint arXiv:2105.04026.

Bishop, C. M. (1994). Mixture density networks.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, Vol. 4: Springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*(518), 859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural network. In *International Conference on Machine Learning*, PMLR, pp. 1613–1622.

Calcaterra, C., & Boldt, A. (2008). Approximating with Gaussians. arXiv preprint arXiv:0805.3795.

Caldeira, J., & Nord, B. (2020). Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms. *Machine Learning: Science and Technology*, *2*(1), 15002.

Carver, C. S., Johnson, S. L., & Timpano, K. R. (2017). Toward a functional view of the p factor in psychopathology. *Clinical Psychological Science*, *5*(5), 880–889.

Casey, B. J., Cannonier, T., Conley, M. I., Cohen, A. O., Barch, D. M., Heitzeg, M. M., Soules, M. E., Teslovich, T., Dellarco, D. V., Garavan, H., & Orr, C. A. (2018). The adolescent brain cognitive development (ABCD) study: Imaging acquisition across 21 sites. *Developmental Cognitive Neuroscience*, *32*, 43–54.

Caspi, A., Houts, R. M., Belsky, D. W., Goldman-Mellor, S. J., Harrington, H., Israel, S., Meier, M. H., Ramrakha, S., Shalev, I., Poulton, R., & Moffitt, T. E. (2014). The p factor: One general psychopathology factor in the structure of psychiatric disorders? *Clinical Psychological Science*, *2*(2), 119–137.

Chen, C., Carlson, D., Gan, Z., Li, C., & Carin, L. (2016). Bridging the gap between stochastic gradient MCMC and stochastic optimization. In *Artificial Intelligence and Statistics*, PMLR, pp. 1051–1060.

Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, PMLR, pp. 1683–1691.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, *49*(4), 327–335.

Cohen, A. O., Breiner, K., Steinberg, L., Bonnie, R. J., Scott, E. S., Taylor-Thompson, K., Rudolph, M. D., Chein, J., Richeson, J. A., Heller, A. S., & Silverman, M. R. (2016). When is an adolescent an adult? Assessing cognitive control in emotional and nonemotional contexts. *Psychological Science*, *27*(4), 549–562.

Dua, D., & Graff, C. (2017). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml

Dubois, J., Galdi, P., Paul, L. K., & Adolphs, R. (2018). A distributed brain network predicts general intelligence from resting-state human neuroimaging data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*(1756), 20170284.

Dunson, D. B., Pillai, N., & Park, J.-H. (2007). Bayesian density regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *69*(2), 163–183.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, PMLR, pp. 1050–1059.

Gan, Z., Henao, R., Carlson, D., & Carin, L. (2015). Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, PMLR, pp. 268–276.

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American Statistical Association*, *95*(452), 1300–1304.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*(410), 398–409.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*, Vol. 1: MIT press Cambridge.

Graves, A. (2011). Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, *24*.

Grazioplene, R. G., G. Ryman, S., Gray, J. R., Rustichini, A., Jung, R. E., & DeYoung, C. G. (2015). Subcortical intelligence: Caudate volume predicts IQ in healthy adults. *Human Brain Mapping*, *36*(4), 1407–1416.

Gulcehre, C., Moczulski, M., Denil, M., & Bengio, Y. (2016). Noisy activation functions. In *International Conference on Machine Learning*, PMLR, pp. 3059–3068.

Hernández-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, PMLR, pp. 1861–1869.

Hitchcock, D. B. (2003). A history of the Metropolis–Hastings algorithm. *The American Statistician*, *57*(4), 254–257.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(1), 1593–1623.

Huberman, D. B., Reich, B. J., & Bondell, H. D. (2021). Nonparametric conditional density estimation in a deep learning framework for short-term forecasting. *Environmental and Ecological Statistics*, *29*, 1–15.

Hwang, J. G., & Ding, A. A. (1997). Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, *92*(438), 748–757.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, PMLR, pp. 448–456.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., & Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407.

Jiang, X., Osl, M., Kim, J., & Ohno-Machado, L. (2012). Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, *19*(2), 263–274.

Johnson, A. A., & Jones, G. L. (2015). Geometric ergodicity of random scan Gibbs samplers for hierarchical one-way random effects models. *Journal of Multivariate Analysis*, *140*, 325–342.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? arXiv preprint arXiv:1703.04977.

Kilroy, E., Liu, C. Y., Yan, L., Kim, Y. C., Dapretto, M., Mendez, M. F., & Wang, D. J. J. (2011). Relationships between cerebral blood flow and iq in typically developing children and adolescents. *Journal of Cognitive Science*, *12*(2), 151.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2016). Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv preprint arXiv:1612.01474.

Lee, J., Humt, M., Feng, J., & Triebel, R. (2020). Estimating model uncertainty of neural networks in sparse information form. In *International Conference on Machine Learning*, PMLR, pp. 5702–5713.

Lee, J., Shridhar, K., Hayashi, H., Iwana, B. K., Kang, S., & Uchida, S. (2019). Probact: A probabilistic activation function for deep neural networks. arXiv preprint arXiv:1905.10761, 5, 13.

Lei, J., & Wasserman, L. (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(1), 71–96.

Leibig, C., Allken, V., Ayhan, M. S., Berens, P., & Wahl, S. (2017). Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, *7*(1), 1–14.

Li, R., Reich, B. J., & Bondell, H. D. (2021). Deep distribution regression. *Computational Statistics & Data Analysis*, *159*, 107203.

Liang, F., Kim, J., & Song, Q. (2016). A bootstrap Metropolis–Hastings algorithm for Bayesian analysis of big data. *Technometrics*, *58*(3), 304–318.

Louizos, C., & Welling, M. (2016). Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, PMLR, pp. 1708–1716.

Louizos, C., & Welling, M. (2017). Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, PMLR, pp. 2218–2227.

Lu, Y., & Lu, J. (2020). A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in Neural Information Processing Systems*, *33*.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, *30*, Citeseer, pp. 3.

MacKay, D. J. C. (1995). Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, *6*(3), 469.

Mandt, S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, *18*, 1–35.

Molchanov, D., Ashukha, A., & Vetrov, D. (2017). Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, PMLR, pp. 2498–2507.

Murray, A. L., Eisner, M., & Ribeaud, D. (2016). The development of the general factor of psychopathology factor through childhood and adolescence. *Journal of Abnormal Child Psychology*, *44*(8), 1573–1586.

Neal, R. M. (2011). MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, *2*(11), 2.

Neal, R. M. (2012). *Bayesian learning for neural networks*, Vol. 118: Springer Science & Business Media.

O'Shea, A., Cohen, R., Porges, E. C., Nissim, N. R., & Woods, A. J. (2016). Cognitive aging and the hippocampus in older adults. *Frontiers in Aging Neuroscience*, *8*, 298.

Papaspiliopoulos, O., & Roberts, G. (2008). Stability of the Gibbs sampler for Bayesian hierarchical models.

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International Conference on Machine Learning*, PMLR, pp. 4075–4084.

Plataniotis, K. N., & Hatzinakos, D. (2017). Gaussian mixtures and their applications to signal processing, *Advanced Signal Processing Handbook*: CRC Press, pp. 89–124.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, *51*(5), 1–36.

Quinonero-Candela, J., Rasmussen, C. E., Sinz, F., Bousquet, O., & Schölkopf, B. (2005). Evaluating predictive uncertainty challenge. In *Machine Learning Challenges Workshop*, Springer, pp. 1–27.

Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning*, PMLR, pp. 1530–1538.

Ritter, H., & Karaletsos, T. (2022). TyXe: Pyro-based Bayesian neural nets for Pytorch. *Proceedings of Machine Learning and Systems*, *4*, 398–413.

Roberts, G. O., & Smith, A. F. M. (1994). Simple conditions for the convergence of the Gibbs sampler and metropolis-hastings algorithms. *Stochastic Processes and Their Applications*, *49*(2), 207–216.

Román, J. C., & Hobert, J. P. (2012). Convergence analysis of the gibbs sampler for Bayesian general linear mixed models with improper priors.

Romano, Y., Patterson, E., & Candes, E. (2019). Conformalized quantile regression. *Advances in neural information processing systems*, *32*.

Sasabayashi, D., Takayanagi, Y., Takahashi, T., Nishiyama, S., Mizukami, Y., Katagiri, N., Tsujino, N., Nemoto, T., Sakuma, A., Katsura, M., & Ohmuro, N. (2021). Reduced cortical thickness of the paracentral lobule in at-risk mental state individuals with poor 1-year functional outcomes. *Translational Psychiatry*, *11*(1), 1–9.

Scarselli, F., & Tsoi, A. C. (1998). Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, *11*(1), 15–37.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Ståhl, N., Falkman, G., Karlsson, A., & Mathiason, G. (2020). Evaluation of uncertainty quantification in deep learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, pp. 556–568.

Sun, Y., & Liang, F. (2022). A kernel-expanded stochastic neural network. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *84*(547-578).

Sun, Y., Xiong, W., & Liang, F. (2021). Sparse deep learning: A new framework immune to local traps and miscalibration. *Advances in Neural Information Processing Systems*, *34*, 22301–22312.

Tagasovska, N., & Lopez-Paz, D. (2019). Single-model uncertainties for deep learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, *32*, pp. 6417–6428. Advances in Neural Information Processing Systems.

Teye, M., Azizpour, H., & Smith, K. (2018). Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, PMLR, pp. 4907–4916.

Thompson, W. K., Barch, D. M., Bjork, J. M., Gonzalez, R., Nagel, B. J., Nixon, S. J., & Luciana, M. (2019). The structure of cognition in 9 and 10 year-old children and associations with problem behaviors: Findings from the ABCD study's baseline neurocognitive battery. *Developmental Cognitive Neuroscience*, *36*, 100606.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., & Joliot, M. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, *15*(1), 273–289.

Wang, Y., & Rocková, V. (2020). Uncertainty quantification for sparse deep learning. In *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 298–308.

Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, Citeseer, pp. 681–688.

Wenzel, F., Roth, K., Veeling, B. S., Świątkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., & Nowozin, S. (2020). How good is the Bayes posterior in deep neural networks really? arXiv preprint arXiv:2002.02405.

Wilson, A. G., & Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. arXiv preprint arXiv:2002.08791.

Wu, T.-Y., Rachel Wang, Y. X., & Wong, W. H. (2020). Mini-batch Metropolis–Hastings with reversible SGLD proposal. *Journal of the American Statistical Association*, *2020*, 1–9.

Xue, Y., Cheng, S., Li, Y., & Tian, L. (2019). Reliable deep-learning-based phase imaging with uncertainty quantification. *Optica*, *6*(5), 618–629.

Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks*, *94*, 103–114.

You, Z., Ye, J., Li, K., Xu, Z., & Wang, P. (2019). Adversarial noise layer: Regularize neural network by adding noise. In *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 909–913.

Zhang, D., Li, L., Sripada, C., & Kang, J. (2020). Image response regression via deep neural networks. arXiv preprint arXiv:2006.09911.

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, *394*, 56–81.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhang, D., Liu, T., & Kang, J. (2023). Density regression and uncertainty quantification with Bayesian deep noise neural networks. *Stat*, *12*(1), e604. https://doi.org/10.1002/sta4.604

APPENDIX A: GIBBS SAMPLING ALGORITHM FOR B-DEEPNOISE

Based on the posterior full conditional distributions in Theorem 1, we describe the Gibbs sampling algorithm for B-DeepNoise in Algorithm A1. Partition the training sample indices $\{1,\ldots,N_{\text{train}}\}$ into T mini-batches $\{\mathcal{T}_t : t=1,\ldots,T\}$ (i.e., $\bigcup_{t=1}^{T}\mathcal{T}_t=\{1,\ldots,N_{\text{train}}\}$ and $\mathcal{T}_{t_1}\cap\mathcal{T}_{t_2}=\emptyset\ \forall t_1\neq t_2$). Without loss of generality, assume L is an even number.

Algorithm 1: Posterior sampler for Bayesian deep noise neural network (B-DeepNoise).

```
Input: training features and targets $[\boldsymbol{x}^{(n)},\boldsymbol{y}^{(n)}]_{n=1}^{N_{\text{train}}}$, testing features $[\boldsymbol{x}^{(n)}]_{n=1}^{N_{\text{test}}}$, number of posterior samples $M$, number of realizations of the
        predictive distribution $R$, training sample mini-batches $[\mathcal{T}_t]_{t=1}^{T}$.
Output: posterior predictive samples for testing targets $[[[\hat{\boldsymbol{y}}_{m,r}^{(n)}]_{r=1}^{R}]_{m=1}^{M}]_{n=1}^{N_{\text{test}}}$
Randomly initialize $[\boldsymbol{\beta}_l]_{l=0}^{L},\ [\boldsymbol{\gamma}_l]_{l=0}^{L},\ [\boldsymbol{\tau}_l^2]_{l=0}^{L},\ [\boldsymbol{\sigma}_l^2]_{l=0}^{L},\ [[\boldsymbol{u}_l^{(n)}]_{l=1}^{L}]_{n=1}^{N_{\text{train}}},\ [[\boldsymbol{v}_l^{(n)}]_{l=0}^{L-1}]_{n=1}^{N_{\text{train}}}$ ;
$[\boldsymbol{u}_0^{(n)}]_{n=1}^{N_{\text{train}}} \leftarrow [\boldsymbol{x}^{(n)}]_{n=1}^{N_{\text{train}}},\quad [\boldsymbol{v}_L^{(n)}]_{n=1}^{N_{\text{train}}} \leftarrow [\boldsymbol{y}^{(n)}]_{n=1}^{N_{\text{train}}}$ ;
for $m \leftarrow 1,\ldots,M$ do
    for $t \leftarrow 1,\ldots,T$ do
        for $l' \leftarrow \{0,1\}$ do
            for $l'' \leftarrow 0,\ldots,L/2-1$ do
                $l \leftarrow l' + 2l''$ ;
                for $n \in \mathcal{T}_t$ do
                    Sample $\boldsymbol{v}_l^{(n)} \mid \boldsymbol{u}_l^{(n)},\boldsymbol{u}_{l+1}^{(n)},\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l,\boldsymbol{\sigma}_l^2,\boldsymbol{\tau}_l^2$ by Equation (9) ;
                    Sample $\boldsymbol{u}_{l+1}^{(n)} \mid \boldsymbol{v}_l^{(n)},\boldsymbol{v}_{l+1}^{(n)},\boldsymbol{\beta}_{l+1},\boldsymbol{\gamma}_{l+1},\boldsymbol{\sigma}_{l+1}^2,\boldsymbol{\tau}_{l+1}^2$ by Equation (10) ;
                end
                Sample $\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l \mid \boldsymbol{u}_l,\boldsymbol{v}_l,\boldsymbol{\tau}_l^2,\boldsymbol{\rho}_l^2,\boldsymbol{\xi}_l^2$ by Equation (11);
                Sample $\boldsymbol{\tau}_l^2,\boldsymbol{\sigma}_l^2,\boldsymbol{\rho}_l^2,\boldsymbol{\xi}_l^2 \mid \boldsymbol{v}_l,\boldsymbol{u}_l,\boldsymbol{\beta}_l,\boldsymbol{\gamma}_l,\boldsymbol{u}_{l+1}$ by Equations (12) to (15);
            end
        end
        $\hat{\boldsymbol{\beta}}_m \leftarrow [\boldsymbol{\beta}_l]_{l=0}^{L},\ \hat{\boldsymbol{\gamma}}_m \leftarrow [\boldsymbol{\gamma}_l]_{l=0}^{L},\ \hat{\boldsymbol{\tau}}_m^2 \leftarrow [\boldsymbol{\tau}_l^2]_{l=0}^{L},\ \hat{\boldsymbol{\sigma}}_m^2 \leftarrow [\boldsymbol{\sigma}_l^2]_{l=0}^{L}$ ;
    end
end
for $n \leftarrow 1,\ldots,N_{test}$ do
    for $m \leftarrow 1,\ldots,M$ do
        for $r \leftarrow 1,\ldots,R$ do
            Sample $\boldsymbol{y}_m^{(n)} \mid \boldsymbol{x}^{(n)},\hat{\boldsymbol{\beta}}_m,\hat{\boldsymbol{\gamma}}_m,\hat{\boldsymbol{\tau}}_m^2,\hat{\boldsymbol{\sigma}}_m^2$ by Equations (6) and (7) ;
            $\hat{\boldsymbol{y}}_{m,r}^{(n)} \leftarrow \boldsymbol{y}_m^{(n)}$
        end
    end
end
```