

Feature Preserving Shrinkage on Bayesian Neural Networks Via the R2D2 Prior

Tsai Hor Chan , Dora Yan Zhang, Guosheng Yin , *Fellow, IEEE*, and Lequan Yu , *Member, IEEE*

Abstract—Bayesian neural networks (BNNs) treat neural network weights as random variables, which aim to provide posterior uncertainty estimates and avoid overfitting by performing inference on the posterior weights. However, selection of appropriate prior distributions remains a challenging task, and BNNs may suffer from catastrophic inflated variance or poor predictive performance when poor choices are made for the priors. Existing BNN designs apply different priors to weights, while the behaviours of these priors make it difficult to sufficiently shrink noisy signals or they are prone to overshrinking important signals in the weights. To alleviate this problem, we propose a novel R2D2-Net, which imposes the R^2 -induced Dirichlet Decomposition (R2D2) prior to the BNN weights. The R2D2-Net can effectively shrink irrelevant coefficients towards zero, while preventing key features from over-shrinkage. To approximate the posterior distribution of weights more accurately, we further propose a variational Gibbs inference algorithm that combines the Gibbs updating procedure and gradient-based optimization. This strategy enhances stability and consistency in estimation when the variational objective involving the shrinkage parameters is non-convex. We also analyze the evidence lower bound (ELBO) and the posterior concentration rates from a theoretical perspective. Experiments on both natural and medical image classification and uncertainty estimation tasks demonstrate satisfactory performances of our method.

Index Terms—Bayesian neural network, medical image analysis, shrinkage priors, uncertainty estimation, variational Inference.

I. INTRODUCTION

IN THE past decades, deep neural networks (DNNs) have shown great success in solving various tasks with high-dimensional features. Most of the state-of-the-art (SOTA) DNN architectures adopt frequentist approaches to train a single set of weights. These models cannot address the epistemic (i.e., model-wise) uncertainties, which may cause overfitting when the number of observations is limited [22]. Failure to address the epistemic uncertainties would lead to poor generalization performance for out-of-distribution data, as the model cannot

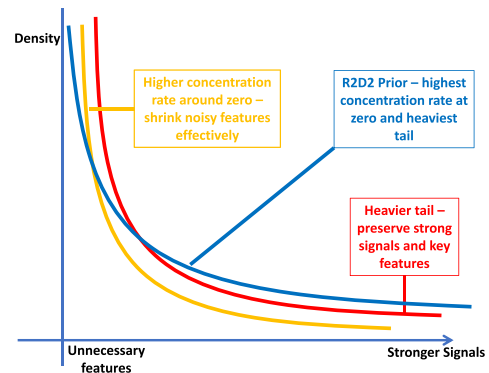


Fig. 1. An illustrative comparison of priors with different tail behaviors and concentration rates at zero. A prior with a heavier tail preserves stronger signals by putting more weights on them. A prior with a larger concentration rate around zero can shrink unnecessary or trivial features more effectively.

learn robust representations from limited training observations. Such frequentist methods also lack uncertainty estimates as they typically only provide point estimates [22]. The recent emergence of Bayesian deep learning frameworks provides a practical solution to quantifying uncertainties in deep learning models.

Bayesian neural networks (BNNs) refine SOTA deep learning architectures with Bayesian approaches, which enable neural networks to quantify uncertainties arising from models [21], [37]. BNNs also act as a natural regularization technique that mitigates the bias of the model by performing inference based on posterior distributions of model weights. Most of the existing BNN architectures adopt zero-mean multivariate Gaussian distributions as the prior distributions for the weights [37]. However, such multivariate Gaussian prior distributions often lead to many unnecessary nodes with large variances, which further results in large variances in posterior predictions. The consequence can be catastrophic because most of the deep BNNs without appropriate priors may underfit the data and thus lead to poor predictions [15], [42]. Therefore, variable shrinkage priors (which can shrink coefficients unrelated to tasks to zero) are needed to reduce the noise in coefficients and alleviate the variance inflation issue.

Recently, several works [15], [29], [34], [43] attempt to adopt global-local shrinkage priors to mitigate the problem of large variances. These priors are able to shrink the coefficients and alleviate the under-fitting problem in BNNs. Although existing shrinkage priors demonstrate superior performance in variable

Received 1 August 2024; revised 1 May 2025; accepted 19 May 2025. Date of publication 23 May 2025; date of current version 6 August 2025. This work was supported in part by the Research Grants Council of Hong Kong under Grant 17308321, Grant 27206123, Grant C5055-24G, and Grant T45-401/22-N, in part by Patrick SC Poon endowment fund, in part by Hong Kong Innovation and Technology Fund under Grant ITS/273/22 and Grant ITS/274/22, in part by National Natural Science Foundation of China under Grant 62201483, and in part by Guangdong Natural Science Fund under Grant 2024A1515011875. Recommended for acceptance by J. Yuan. (Corresponding author: Lequan Yu.)

The authors are with the Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong (e-mail: lqyu@hku.hk).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TPAMI.2025.3572766>, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2025.3572766

TABLE I
TAIL DECAY AND CONCENTRATION AT ZERO OF COMMONLY USED
GLOBAL–LOCAL SHRINKAGE PRIORS [47]

Priors	Tail Decay	Concentration at 0
Horseshoe	$\mathcal{O}\left(\frac{1}{\beta^2}\right)$	$\mathcal{O}\left(\log\left(\frac{1}{ \beta }\right)\right)$
Horseshoe+	$\mathcal{O}\left(\frac{\log \beta }{\beta^2}\right)$	$\mathcal{O}\left(\log^2\left(\frac{1}{ \beta }\right)\right)$
Dirichlet–Laplace	$\mathcal{O}\left(\frac{1}{\exp\{\sqrt{2} \beta \}}\right)$	$\mathcal{O}\left(\frac{1}{ \beta ^{1-a^*}}\right)$
Generalized Double Pareto	$\mathcal{O}\left(\frac{1}{ \beta ^{1+\alpha}}\right)$	$\mathcal{O}(1)$
R2D2	$\mathcal{O}\left(\frac{1}{ \beta ^{1+2b}}\right)$	$\mathcal{O}\left(\frac{1}{ \beta ^{1-2a_\pi}}\right)$

Here, β represents the model parameters, and a^* , α , b , α_π are the respective hyperparameters controlling the shrinkage behaviours, whose precise definitions can be found in the original papers.

selection, their properties are subject to several limitations. For instance, these priors have either a low concentration rate around zero or light tails. A low concentration rate around zero leads to weak shrinkage effects, while the variance of prediction remains large. Light (or thin) tails under-weigh the effects of large coefficients, which would over-shrink important signals [47]. Fig. 1 presents an illustrative comparison of the the tail and spiking behaviours of different priors. In particular, Gaussian distribution has the lightest tail and tends to assign almost zero weight to large signals. This leads to over-regularization as well as poor feature representation learning, especially when the architecture is deep. Furthermore, the optimization of shrinkage parameters would be difficult with stochastic variational inference, when the joint distribution of the shrinkage and weight parameters is not convex [17], [20].

The aforementioned limitations motivate us to select a prior that has the highest concentration rate at zero and heaviest tails, which are crucial to predictive models with a large number of parameters — especially for DNNs. Table I presents the concentration rates around zero and the tail thickness of commonly used shrinkage priors. By comparing the rates of various priors, we are prompted to select the R2D2 prior [47] for the neural network weights and propose a novel BNN design — the R2D2-Net. The R2D2-Net is more effective in model selection than designs based on other existing shrinkage priors, because it can choose more important weights in predictive tasks.

Contribution summary: 1) We propose a novel BNN design — the R2D2-Net by specifying an R2D2 prior on the model weights, which improves the shrinkage effect and the predictive performance over existing priors. 2) We develop the stochastic variational Gibbs inference algorithm that integrates the Gibbs sampling procedure and gradient-based optimization. It estimates the shrinkage parameters more effectively, particularly when the joint distribution of the shrinkage and weight parameters is not convex. 3) We theoretically analyze the evidence lower bound (ELBO) in the variational inference of BNN and develop analytical forms of the Kullback–Leibler (KL) divergences of the shrinkage parameters. Theoretical analysis validates that the R2D2-Net possesses the minimax posterior concentration rate under the polynomial boundedness assumption. 4) Extensive synthetic and real data experiments demonstrate the outstanding performance of R2D2-Net on inference.

predictive and uncertainty estimation tasks compared with various combinations of existing priors and inference algorithms. Codes are available at <https://github.com/HKU-MedAI/r2d2bnn>.

II. RELATED WORKS

Global–Local Shrinkage Priors: High-dimensional regression often suffers from the curse of dimensionality, which motivates novel approaches to shrinkage of coefficients and variable selection. Global–local shrinkage priors are a class of shrinkage priors that can be essentially expressed as a global–local scale Gaussian family. Existing shrinkage priors possess desirable theoretical and empirical properties that can effectively perform coefficient shrinkage. Carvalho et al. proposed the Horseshoe prior, which exhibits Cauchy-like flat and heavy tails and maintains a high concentration rate at zero. Although the Horseshoe prior and its variants [3], [32] are shown to be able to shrink the coefficients, their tail thickness and concentration rates at zero are less desirable compared with some recently proposed global–local shrinkage priors. A higher concentration rate at zero allows the model to shrink unnecessary coefficients toward zero more aggressively, and a heavier tail can avoid shrinking key coefficients that have large values with strong signals. Zhang et al. proposed the R^2 -induced Dirichlet Decomposition (R2D2) prior, which specifies a prior based on the R^2 value of a model fit. The R2D2 prior demonstrates optimal behaviors both in the tails and concentration at zero, which potentially provides the best shrinkage performance while preserving the important signals in the weights.

Bayesian Neural Networks: BNNs specify prior distributions on the weights and bias parameters of the neural network. A vanilla BNN assumes a zero-mean multivariate Gaussian distribution on the parameters. The MC Dropout approach [12] randomly drops out weights to produce posterior samples from a trained frequentist neural network. Moreover, the variance inflation exacerbates as the number of layers increases, making it extremely difficult to build and optimize deep BNNs [9]. Most of the existing works focus on small architectures (e.g., LeNet) and small datasets (e.g., CIFAR 100), while several works [9], [25] attempt to scale up to more modern and deeper architectures (e.g., ResNet101).

To address the above issues, sparsification methods have been adopted to shrink unnecessary neurons to prevent variance inflation. Utilization of sparsity–induced priors [27] has become a more popular approach than variational dropout methods [30], [38]. Ghosh et al. proposed to place the Horseshoe prior on the variances of weights to resolve the large prediction variance problem. However, due to the relatively low concentration rate around zero, the shrinkage effect is compromised. Moreover, the relative lighter tails of the Horseshoe prior than R2D2 limits its capability to preserve important signals, which likely leads to over-shrinkage.

Posterior Computation of BNNs: Conventional methods for computing BNN posteriors are mostly sampling-based [7], [31], [44], [46], which are practical for neural networks since they follow closely the stochastic optimization schema [46]. Existing

works on sampling-based approaches can be categorized into two prominent families: Langevin dynamics [44] and Hamiltonian dynamics [7], [16]. Despite their success, these sampling-based methods are often considered inefficient in terms of computational complexity. When tackling large-scale problems, simplified computational metrics need to be introduced for better computational efficiency [46]. Recently, there have been approaches utilizing variational inference (VI) to learn the probabilistic assumptions in BNNs.

Classical BNN training paradigms widely adopt a mean field VI approach to approximating the posteriors, which assume independent marginal distributions [10], [14], [15], [30], [35], [37]. The VI typically approximates a posterior distribution $p(\theta|\mathbf{y})$ by a variational posterior distribution q obtained from a candidate set $\mathcal{Q}$ by maximizing an evidence lower bound (ELBO):

$$\max_{q \in \mathcal{Q}} \mathbb{E}_{\theta \sim q} [\log p(\mathbf{y}|\theta)] - \text{KL}(q||\pi), \quad (1)$$

where $\text{KL}(q||\pi) = \mathbb{E}_{q \in \mathcal{Q}} [\log p(\theta|\cdot)] + \mathbb{H}[\pi(\theta)]$, is the KL divergence between q and the prior distribution π , $\mathbb{H}(\pi(\theta))$ is the entropy of the distribution, and $p(\mathbf{y}|\theta)$ is the likelihood. Most of the works [10], [12], [15], [37] assume Gaussian prior distributions on weights and hence the KL divergence can be approximated by $\text{KL}(q||\pi) = \sum_{j,l} \text{KL}(q(w_{jl}|\cdot)||\pi(w_{jl}|\cdot))$, where $q(w_{jl}|\cdot)$ is the variational posterior and $\pi(w_{jl}|\cdot) = \mathcal{N}(w_{jl}|0, 1)$ is the standard normal prior distribution on weights. However, Gaussian assumption is made for convenience and the estimation of KL suffers from large approximation error. By comparing the KL divergence of the analytical distributions of the hierarchical prior (e.g., Horseshoe, R2D2), a more accurate approximation of the ELBO can be obtained.

Sparsifying Neural Networks: Another related field of our work is neural sparsification, which focuses on compressing neural networks to prune unnecessary neurons and improve the space efficiency [19], [27], [30], [39]. Sparsity-induced prior is also a popular choice in this field [15], [27]. Despite similarities in these approaches, our work focuses on a BNN design with shrinkage priors which can improve its capability of predictive and uncertainty estimation instead of compressing the existing architectures.

III. PRELIMINARIES

Deep Neural Network (DNN): A DNN with L layers can be written as

$$f_l(\mathbf{x}) = \frac{1}{\sqrt{d_{l-1}}} W_l \phi(f_{l-1}(\mathbf{x})) + b_l, \quad l \in \{1, \dots, L\},$$

where $\mathbf{x}$ is the input feature vector, ϕ is a nonlinearity activation function (e.g., the rectified linear unit (ReLU), $\phi(a) = \max(0, a)$), d_{l-1} is the dimension of the input of layer $l-1$, $b_l \in \mathbb{R}^{d_l}$ is a vector containing the bias parameters for layer l , and W_l is the weight tensor. For linear layers, we have $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$, and for convolutional layers, $W_l \in \mathbb{R}^{d_l \times d_{l-1} \times k_0 \times k_0}$, where k_0 is the kernel size. Let $\mathbf{w}_l = \{W_l, b_l\}$ denote the set of weight and bias parameters of layer l , and let w_{jl} denote the j th element of the parameter vector at layer l , and let $p_l = |\mathbf{w}_l|$, the

size of set $\mathbf{w}_l$. The trainable network parameters are summarized as $\theta = \{\mathbf{w}_l\}_{l=1}^L$.

Notation for Sparsity-induced BNN: Let $K_n = |\theta|$ denote the total number of parameters, and let D_n denote the dimension of the input and they are assumed to grow with the training size n . We further allow L_n , the number of layers, increasing with n . Let H_l be the number of hidden units at layer l with $\bar{H} = \max_{1 \leq l \leq L_n-1} H_l$. Let $\gamma = \{1_{w_{jl} > 0} : j \in \{1, \dots, H_l\}, l \in \{1, \dots, L_n\}\}$ denote the connectivity of each parameter w_{jl} .

Bayesian Neural Network (BNN): A BNN specifies a prior $\pi(\theta)$ on the trainable weights θ . Given the dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ of n pairs of observations, we aim to estimate the posterior distribution of the weights,

$$p(\theta|\mathcal{D}) = \frac{\pi(\theta) \prod_{i=1}^n p(y_i|f(\theta, \mathbf{x}_i))}{p(\mathcal{D})},$$

where $p(y_i|f(\theta, \mathbf{x}_i))$ is the likelihood function and $p(\mathcal{D})$ is the normalization term.

Multivariate Gaussian: The density of a p -dimensional multivariate Gaussian distribution is defined as

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\},$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$ is the mean vector and $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ is the covariance matrix.

Generalized Inverse Gaussian: The generalized inverse Gaussian distribution is denoted as $Z \sim \text{giG}(\chi, \rho, \lambda_0)$, which has the density function,

$$f(z; \chi, \rho, \lambda_0) = \frac{(\rho/\chi)^{\lambda_0/2}}{2K_{\lambda_0}(\sqrt{\rho\chi})} z^{\lambda_0-1} \exp\{-(\rho z + \chi/z)/2\},$$

where $K_{\lambda_0}(\cdot)$ is a modified Bessel function of the second kind. Specifically, an inverse Gaussian distribution of the form $f(x; \mu, \lambda) = (\frac{\lambda}{2\pi x^3})^{1/2} \exp(-\frac{\lambda(x-\mu)^2}{2\mu^2 x})$ is a special case of giG with $\rho = \lambda/\mu^2$, $\chi = \lambda$, and $\lambda_0 = -1/2$.

IV. METHODOLOGY

To achieve the best variable shrinkage performance, we impose the R2D2 prior on the neural network weights, leading to the R2D2-Net.

By placing the R2D2 prior on the weights, irrelevant weights can be shrunk effectively towards zero and important weights can be well preserved. We also propose a variational Gibbs inference procedure and develop analytical forms of KL divergences of the shrinkage parameters to obtain better estimates of posterior distributions of the weights. Fig. 2 presents the overall workflow of the R2D2-Net.

A. The R2D2-Net

Consider a linear model,

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, n, \quad (2)$$

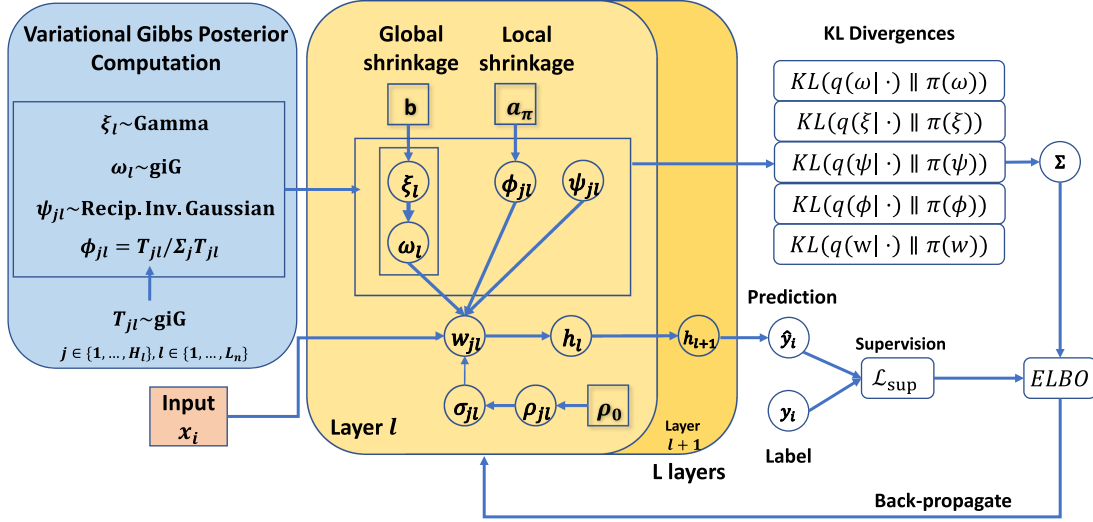


Fig. 2. Overview of the proposed R2D2-Net with the yellow part representing the graphical model of each neuron and the blue part summarizing the variational Gibbs inference for computing the posterior distribution of weights.

where y_i is the response, x_i is the p -dimensional vector of covariates for the i th observation, $\beta = (\beta_1, \dots, \beta_p)^\top$ is a vector of coefficients, and ϵ_i is the error term. The R2D2-Net specifies a prior on the R^2 from the model fit of (2). The R^2 of linear prediction is given by

$$R^2(\beta) = \frac{\text{var}(\mathbf{X}^\top \beta)}{\text{var}(\mathbf{X}^\top \beta) + \sigma^2},$$

where β can be viewed as the weight tensor of the convolutional or the linear layer and $\mathbf{X} \in \mathbb{R}^{p \times n}$ is the data matrix. By specifying a beta prior on $R^2(\beta)$, the marginal R2D2 prior has the form,

$$\begin{aligned} \beta_j | \psi_j, \omega, \phi_j &\sim \mathcal{N}(0, \psi_j \phi_j \omega \sigma^2 / 2), \quad \psi_j \sim \text{Exp}(1/2), \\ \phi &\sim \text{Dir}(a_\pi, \dots, a_\pi), \quad \omega | \xi \sim \text{Ga}(a, \xi), \quad \xi \sim \text{Ga}(b, 1), \end{aligned} \quad (3)$$

where Exp denotes an exponential distribution, Ga a Gamma distribution, and Dir a Dirichlet distribution, and ϕ_j is the j th element of ϕ . The R2D2 prior has the highest concentration rate at zero and heavier tails than other global-local priors [47]. Therefore, it can more effectively shrink the coefficients of unimportant covariates to zero. For coefficients that have large signals (i.e., large norms), the heavy-tail nature of the R2D2 prior is able to avoid over-shrinking these coefficients, thus preserving the ability to extract key features from the input data.

With the marginal distributions of weights in (3), we can formulate the layers of the R2D2-Net. We assign the R2D2 distribution in (3) as the prior for w_{jl} , the j th element of parameters of the l th layer $\mathbf{w}_l$. We can then compose the R2D2-Net by specifying a combination of convolutional layers and linear layers.

B. Variational Gibbs Inference for Optimization

We adopt a mean field approach to computing the ELBO by factorizing $q(\theta)$ into a product of the marginal distributions of all neurons. First, we update $\mathbf{w}$ and ρ by back-propagating the

ELBO in (1). We initialize the weight parameters $\mathbf{w}_l$ with a reparameterized Gaussian distribution, $w_{jl} \sim \mathcal{N}(\mu_{jl}, \sigma_{jl}^2 \psi_{jl} \phi_{jl} \omega_l)$, where each standard deviation σ_{jl} is reparameterized by introducing a parameter ρ_{jl} such that $\sigma_{jl} = \log(1 + e^{\rho_{jl}})$. We assign an individual variance term σ_{jl} to each weight, which is different from Zhang et al. [47] who assume the same $\sigma_l = \sigma_l \mathbf{1}$ for all weight parameters in layer l . The distribution of σ_l in Zhang et al. is updated by the regression MSE, which is analogous to learning the variance of neurons by back-propagation of task-specific loss. Therefore, under the deep learning setting, we distinctively specify a variance parameter σ_l for each neuron and learn them by back-propagating the task-specific loss. We set the prior values of $\psi = \{\psi_{jl}\}_{j=1}^{p_l} \prod_{l=1}^L$, $\phi = \{\phi_{jl}\}_{j=1}^{p_l} \prod_{l=1}^L$, $\omega = \{\omega_l\}_{l=1}^L$, $\xi = \{\xi_l\}_{l=1}^L$ with the prior distribution defined in (3) and $\mu_{jl} = 0$, $\rho_{jl} = \rho_0$ for the first step. With the weight parameter samples, we are able to compute the ELBO using (1). The trainable parameters $\mathbf{w}$ and ρ can be updated by back-propagating the ELBO.

We then update shrinkage parameters using the updated $\mathbf{w}$ and σ . Following the Gibbs sampling procedures proposed by Zhang et al. we develop our variational Gibbs inference algorithm to update the shrinkage parameters alternatively using their individual posterior distributions. We first sample ψ_{jl}, ω_l and ξ_l ,

$$\begin{aligned} \psi_{jl}^{-1} &| w_{jl}, \phi_{jl}, \sigma_{jl}^2 \\ &\sim \text{InvGaussian} \left(\mu = \sqrt{\sigma_{jl}^2 \phi_{jl} \omega_l / 2} / |w_{jl}|, \lambda = 1 \right), \\ \omega_l &| \mathbf{w}_l, \phi_l, \xi_l, \sigma_l^2 \\ &\sim \text{giG} \left(\chi = \sum_{j=1}^{p_l} 2w_{jl}^2 / (\sigma_{jl}^2 \phi_{jl} \psi_{jl}), \rho = 2\xi_l, \lambda_0 = a_l - \frac{p_l}{2} \right), \end{aligned}$$

$$\xi_l | \omega_l \sim \text{Ga}(a_l + b_l, 1 + \omega_l).$$

To sample $\phi_l | \mathbf{w}_l, \psi_l, \xi_l, \sigma_l^2$, we first draw $T_{1l}, \dots, T_{p_l l}$ independently with $T_{jl} \sim \text{giG}(\chi = 2w_{jl}^2 / (\sigma_{jl}^2 \psi_{jl}), \rho = 2\xi_l, \lambda_0 =$

Algorithm 1: The Variational Gibbs Inference Algorithm.**Input:**Number of layers L ; Prior distributions of weight parameters $\pi(\theta)$;Total numbers of parameters of each layer $\{p_l\}_{l=1}^L$;Local shrinkage parameters $\phi = \{\phi_{jl}\}_{j=1}^{p_l} \sim \mathcal{L}, \psi = \{\psi_{jl}\}_{j=1}^{p_l} \sim \mathcal{L}$;Global shrinkage parameters $\omega = \{\omega_l\}_{l=1}^L, \xi = \{\xi_l\}_{l=1}^L$;Input data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$;Hyperparameters a_π, ρ_0, b .**Output:**The posterior distribution of the weights $p(\theta|\mathcal{D})$.

- 1: Initialize $\pi(w_{jl}) \sim \mathcal{N}(0, (\log(1 + e^{\rho_0}))^2)$, $a_l = p_l a_\pi$, $b_l = b$, $\mu_{w_{jl}} = 0$
- 2: $\mathbf{h}_1 = \mathbf{x}_i$
- 3: Sample $\psi_{jl} \sim \text{Exp}(1/2)$, $\phi_{jl} \sim \text{Dir}(a_\pi, \dots, a_\pi)$, $\xi_l \sim \text{Ga}(b_l, 1)$, $\omega_l | \xi_l \sim \text{Ga}(a_l, \xi_l)$
- 4: **for** each step **do**
- 5: **for** l in $1 : L$ **do**
- 6: **for** w_{jl} in $\mathbf{w}_l$ **do**
- 7: Sample $w_{jl} \sim \mathcal{N}(\mu_{w_{jl}}, \psi_{jl} \phi_{jl} \omega_l \sigma_{jl}^2 / 2)$ ▷ Sample weights
- 8: Sample $\rho_{jl} \sim \mathcal{N}(\mu_{\rho_{jl}}, \sigma_{\rho_{jl}}^2)$
- 9: Set $\sigma_{jl} = \log(1 + e^{\rho_{jl}})$ ▷ Reparameterized Gaussian
- 10: Sample $\omega \sim \text{giG}(\chi = \sum_{j=1}^{p_l} 2w_{jl}^2 / (\sigma_{jl}^2 \phi_{jl} \psi_{jl}), \rho = 2\xi_l, \lambda_0 = a_l - \frac{p_l}{2})$
- 11: Sample $\xi_l \sim \text{Ga}(a_l + b_l, 1 + \omega_l)$
- 12: Sample $\psi_{jl}^{-1} \sim \text{InvGaussian}(\mu = \sqrt{\sigma_{jl}^2 \phi_{jl} \omega_l / 2} / |w_{jl}|, \lambda = 1)$
- 13: Sample $T_{jl} \sim \text{giG}(\chi = 2w_{jl}^2 / (\sigma_{jl}^2 \psi_{jl}), \rho = 2\xi_l, \lambda_0 = a_l - \frac{p_l}{2})$
- 14: Set $\phi_{jl} = T_{jl} / \sum_j T_{jl}$
- 15: **end for**
- 16: Compute $\mathbf{h}_{l+1} = \mathbf{w}_l \mathbf{h}_l + \text{bias}_l$
- 17: **end for**
- 18: Obtain prediction $\hat{\mathbf{y}}_n$ from $\mathbf{h}_{L-1}$
- 19: Compute the supervision loss, $\text{KL}(q||\pi)$ (Table II), and the ELBO.
- 20: Back-propagate the ELBO to update the mean and variance of θ .
- 21: **end for**
- 23: **return** Posterior distribution $p(\theta|\mathcal{D})$.

$a_l - \frac{p_l}{2}$), and then set $\phi_{jl} = \frac{T_{jl}}{T_l}$ with $T_l = \sum_j T_{jl}$. We repeat the above steps to train the R2D2-Net iteratively till convergence or early stopping criteria are met (e.g., the loss does not decrease). Algorithm 1 presents the detailed workflow of the variational Gibbs inference, which leverages the advantages of both posterior computation and gradient-based estimation to obtain better approximation of the shrinkage parameters.

C. Estimation of KL Divergences With Variational Posterior Distributions

In light of the importance of obtaining an accurate estimate of the KL loss in variational inference, we utilize the full posterior distribution obtained in variational Gibbs inference and the R2D2 prior to obtain a more accurate estimate of the KL loss. The KL divergence of the variational posterior q and the prior π can be divided into several components as follows,

$$\begin{aligned} \text{KL}(q(\theta|\cdot)||\pi(\theta)) &= \text{KL}(q(\xi|\cdot)||\pi(\xi)) \\ &+ \text{KL}(q(\omega|\cdot)||\pi(\omega)) + \text{KL}(q(\psi|\cdot)||\pi(\psi)) \\ &+ \text{KL}(q(\phi|\cdot)||\pi(\phi)) + \text{KL}(q(\mathbf{w}|\cdot)||\pi(\mathbf{w})). \end{aligned}$$

We can obtain the closed-form solutions of the KL divergences for ξ, ω , and ψ . We approximate the KL divergence of ϕ using samples from the variational posterior distribution $q(\phi|\cdot)$. Table II presents the closed forms of the KL divergences on the shrinkage parameters ξ_l, ω_l , and ψ_{jl} . The closed forms in Table II can be obtained by using $\mathbb{E}(X), \mathbb{E}(X^{-1})$ and $\mathbb{E}(\log X)$ for $X \sim \text{giG}$, which are given in the supplementary materials together with the detailed derivations of the KL losses.

V. POSTERIOR CONSISTENCY OF R2D2-NET

Posterior consistency measures the speed of posterior concentration rates around its true density function, which is a major metric for validating a prior. We extend the theoretical results in Zhang et al. [47], which establish theoretical properties of the R2D2 prior, and the results from Sun et al. [41], which formulate the convergence rates of sparse DNNs under general priors.

Regularity Conditions: Let $\mu(\theta, \gamma, \mathbf{x})$ be the prediction of a BNN given an input $\mathbf{x}$, where θ is the weights and biases of the BNN and γ is the set of all shrinkage parameters of the BNN. The regularity conditions of sparse DNNs are specified as follows.

TABLE II
ANALYTICAL FORMS OF KL-DIVERGENCES OF THE SHRINKAGE PARAMETERS $(\xi_l, \omega_l, \psi_{jl})$

	Prior π	Variational Posterior q	Closed Form of KL-divergence
ξ_l	Gamma	Gamma	$\mathbb{E}_q \left[\log \left(\frac{(1 + \omega_l)^{a_l + b_l}}{\Gamma(a_l + b_l)} \xi_l^{a_l + b_l - 1} e^{-(1 + \omega_l)\xi_l} \right) \right] - \mathbb{E}_q \left[\log \left(\frac{1}{\Gamma(b_l)} \xi_l^{b_l - 1} e^{-\xi_l} \right) \right]$
ω_l	Gamma	Generalized InvGaussian	$\mathbb{E}_q \left[\log \left(\frac{(\rho/\chi)^{\lambda_0/2}}{2K_{\lambda_0}(\sqrt{\rho\chi})} \omega_l^{\lambda_0 - 1} e^{(-\rho\omega_l + \chi/\omega_l)/2} \right) \right] - \mathbb{E}_q \left[\log \left(\frac{\xi_l^{a_l}}{\Gamma(a_l)} \omega_l^{a_l - 1} e^{-\omega_l \xi_l} \right) \right]$
ψ_{jl}	Exp	Reciprocal InvGaussian	$\mathbb{E}_q \left[\log \left(\frac{1}{\psi_{jl}\sqrt{2\pi}} \exp \left(-\frac{(1 - \psi_{jl}\mu)^2}{2\psi_{jl}\mu} \right) \right) \right] - \mathbb{E}_q \left[\log \left(\frac{1}{2} e^{-\frac{1}{2}\psi_{jl}} \right) \right]$

A.1 The input $\mathbf{x}$ is bounded by 1 entry-wisely, i.e., $\mathbf{x} \in \Omega = [-1, 1]^{D_n}$, and the density of $\mathbf{x}$ is bounded in its support Ω uniformly with respect to n .

A.2 The unknown regression mean function $\mu^*(\mathbf{x})$ can be well approximated by a sparse DNN model such that $\mu(\theta^*, \gamma^*, \mathbf{x})$ satisfies the following conditions:

A.2.1 $\|\mu^*(\mathbf{x}) - \mu(\theta^*, \mathbf{x})\|_{L^2(\Omega)} \leq \varpi_n$ where the approximation error $\varpi_n \rightarrow 0$ as sample size $n \rightarrow \infty$.

A.2.2 The network structure satisfies: $r_n L_n \log n + r_n \log \bar{H} + s_n \log D_n \leq C_0 n^{1-\epsilon}$, where $0 < \epsilon < 1$ is a small constant, $r_n = |\gamma^*|$ denotes the connectivity of γ^* , and s_n denotes the input dimension of γ^* .

A.2.3 The network weights are polynomially bounded: $\|\theta^*\|_\infty < E_n$, where $E_n = n^{C_1}$ for some constant $C_1 > 0$.

A.3 The activation function ϕ is Lipschitz continuous with a Lipschitz constant of 1.

To ensure the minimax rate, we further assume the polynomial bounding condition from [4], [40], which is slightly stronger than similar conditions in the existing literature on sparsity-induced regression problems. Bolcskei et al. [4] proved that if the network parameters are bounded in absolute value by some polynomial $g(r_n)$, i.e., $\|\theta^*\|_\infty \leq g(r_n)$, then the approximation error $\varpi = \mathcal{O}(r_n^{-\alpha^*})$ for some constant α^* . Condition A.1 is a typical assumption for posterior consistency (e.g., see [33], [41], [47]) where all bounded data can be normalized to satisfy this assumption. Condition A.3 is satisfied by many conventional activation functions such as sigmoid, tanh, and ReLU.

Theorem 1: Consider a DNN with L_n layers and at most K_n connections, where both L_n and K_n are increasing with n . Let $k_n \asymp \sqrt{r_n(\log K_n)/n}$ and denote $\mathbb{P}^*$ and $\mathbb{E}^*$ the respective probability measure and expectation with respect to the data $\mathcal{D}$. Assume that conditions A.1–A.3 hold, if the hyperparameter $a_\pi \leq \log(1 - D_n^{-1+u})/(2 \log k_n)$, and $E_n/(L_n \log n + \log \bar{H})^{1/2} \lesssim b \lesssim n^\alpha$ for some constant $\alpha > 0$, then there exists an error sequence $\epsilon_n^2 = \mathcal{O}(\varpi_n^2) + \mathcal{O}(\zeta_n^2)$, with $\zeta_n^2 = [r_n L_n \log n + r_n \log \bar{H} + s_n \log D_n]/n$, such that $\lim_{n \rightarrow \infty} \epsilon_n = 0$ and $\lim_{n \rightarrow \infty} n \epsilon_n^2 = \infty$, and the posterior distribution satisfies

$$1) \mathbb{P}^*\{p(d(p_\theta, p_{\mu^*}) > 4\epsilon_n | \mathcal{D}) \geq 2e^{-cn\epsilon_n^2}\} \leq 2e^{-cn\epsilon_n^2},$$

$$2) \mathbb{E}_\mathcal{D}^*\{p(d(p_\theta, p_{\mu^*}) > 4\epsilon_n | \mathcal{D})\} \leq 4e^{-2cn\epsilon_n^2},$$

for sufficiently large n , where c is a constant, d is the Hellinger distance between two density distributions, $p[\cdot]$ represents the

posterior distribution, p_{μ^*} denotes the underlying true data distribution, and p_θ denotes the data distribution reconstructed by the BNN based on its posterior samples.

Theorem 1 establishes the Bayesian contraction rate for the R2D2-Net under the Hellinger metric. The detailed proof of this theorem follows [40], [41] and is provided in the supplementary materials.

Remark 1: The minimax ϵ -convergence rate may not hold if we do not assume E_n to be bounded by a polynomial. When the assumption is loosen to $\log(E_n) = \mathcal{O}(\log D_n)$, we have the ϵ -neighbourhood contracting rate $\epsilon_n = (a_n \log D_n)^{1/2}/n$ for the Horseshoe prior [2] and $\epsilon_n = (a_n \log D_n)^{1/2}/\sqrt{n}$ for the spike-and-slab prior [41] under the linear regression settings. If the polynomial assumption is violated, the R2D2 prior also provides a near-minimax ϵ -convergence rate under a relaxed assumption given by Zhang et al. [47] with tighter assumptions on α_π and b , where the spike-and-slab prior shares the same rate under such assumptions [13], [41].

VI. SIMULATION STUDY

We first apply our method to simulated scenarios to validate the predictive and shrinkage performance of the R2D2-Net. We control the depth to observe how the performance varies as the depth of the network increases. We also test the inference performance using Hamiltonian Monte Carlo (HMC) fitted weights on the respective prior, which is treated as the oracle truth.

A. Experimental Setup

Scenarios: We generate the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ with $n = 10000$ and each data point x_{ij} , the j th component of $\mathbf{x}_i$, is sampled from a uniform distribution $\mathcal{U}(-5, 5)$, and the noise $\epsilon_i \sim \mathcal{N}(0, 3^2)$. We design three simulation scenarios: (1) **Polynomial case:** $y_i = x_i^3 + \epsilon_i$; (2) **Low-dimensional non-linear regression:** $y_i = x_{i1}x_{i2} + x_{i3}x_{i4} + \epsilon_i$; (3) **High-dimensional non-linear regression:** $y_i = f(\mathbf{x}_i) + \epsilon_i$, where f is a two-layer perceptron with randomly initialized weights and *Relu* non-linearity. Additional scenarios and results are presented in the supplementary materials. In contrast to other scenarios, the data in Scenario 3 are generated from a randomly initialized neural network. The features are hence mostly noise (or trivial) features and shrinkage methods are expected to underperform as they shrink noise features to zeros.

For each scenario, we randomly generate five sets of data.

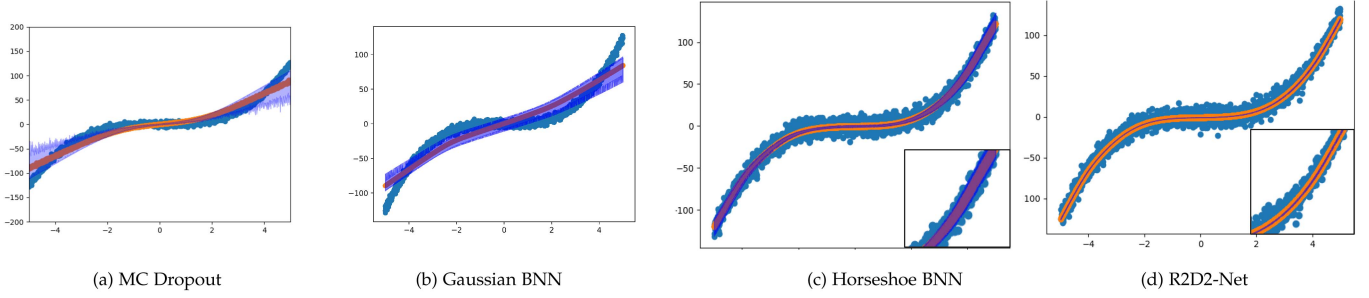


Fig. 3. Prediction mean and confidence intervals of R2D2-Net at test time on $y_i = x_i^3 + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, 9)$. The number of layers is 3 and the number of samples is 100 during the validation phase. The blue dots are the ground truth data points, the yellow line is the mean of prediction and the blue shadow is the prediction interval. We observe that the R2D2-Net yields a smaller prediction variance than MC Dropout, Gaussian BNN, and Horseshoe BNN.

We split 80% of the data as the training set and 20% as the testing set. All methods are trained on 100 epochs and a batch size of 1024, with possible early stopping when the loss does not decrease for 5 epochs. For other priors included in the experiments, we report the performance with the best-performing optimization algorithms.

Competitive Methods: We compare our method with a variety of existing BNN designs. The hyperparameter settings of each benchmark method and the summary of uncertainty measures used are presented in the supplementary materials.

We divide the baseline methods into combinations of priors and inference algorithms. We consider the following common priors for BNNs:

- 1) **Gaussian Prior (Gauss)**
 - 2) **Horseshoe Prior (HS)**
 - 3) **Spike-and-slab Prior (SaS)**,
- and the following inference algorithms:

- 1) **Stochastic Variational Inference (SVI) [35]:** classical mean field variational inference which back-propagates the stochastic gradient on the BNN parameters;
- 2) **Stochastic Gradient MCMC (SGMCMC) [31];**
- 3) **Stochastic Gradient Langevin Dynamics (SGLD) [44].**

B. Experimental Results

Predictive Performance: R2D2-Net Achieves Competitive Performance with Deeper Layers: We compare the prediction MSE and variance of each BNN design. When $L = 0$, the model is equivalent to a linear regression. Table III presents the simulation results, and Fig. 3 shows the prediction means and variances of R2D2-Net and the baseline BNN designs.

We observe that the R2D2-Net yields the smallest prediction error among all competitive designs, and overall a low inference error compared to the oracle truth.

The R2D2-Net also shows greater improvement in prediction performance (i.e., smaller prediction MSE) as the number of layers increases. This demonstrates that the R2D2-Net is more capable of supporting deeper BNN architectures than other BNN designs. On the other hand, when compared to traditional sampling-based optimization algorithms, we observe that using the Gibbs posterior as the variational posterior would lead

to more stable convergence performance and better predictive results. This empirically demonstrates the effectiveness of using the Gibbs posterior in variational inference. This highlights the variance inflation issues of vanilla BNNs.

Shrinkage Performance. R2D2 Prior Best Shrinks Unnecessary Neurons to Zero: We study the shrinkage performance of R2D2-Net in comparison with the Horseshoe BNN [15]. We investigate the distribution of the smallest node weight vectors to compare the shrinkage performance among different priors. We plot the distributions of coefficients w_{jl} with the smallest magnitude $\|\mathbb{E}[w_{jl}]\|$ (Fig. 5). We observe that the weight samples of the R2D2-Net have the highest concentration rate at zero compared with Horseshoe BNN and Gaussian BNN. This validates that the highest concentration rate property of the R2D2 prior also holds when generalized to neural networks. We also validate that the R2D2-Net has the best shrinkage performance than existing BNNs with other priors.

Inference Accuracy: We also evaluate the inference accuracy of R2D2-Net in addition to predictive accuracy. We consider a two-layer MLP (multi-layer perceptron) and use the HMC algorithm to generate the oracle truth posteriors of the BNN parameters. We use the L2-normalized difference between the learned parameters and the oracle truth as the inference error.

Impact of Hyperparameters: We investigate how sensitive the R2D2-Net is to the changes in the hyperparameters, such as a_π, b and ρ_0 . We perform the evaluation using the simulation scenarios in Section VI. Fig. 4 presents the results using an R2D2 MLP with $L = 3$. We observe that our method is robust to changes in these hyperparameters. The performance of R2D2-Net is more sensitive to the prior variance parameter ρ_0 than the other hyperparameters a_π and b of the R2D2 prior in (3).

VII. EXPERIMENTS ON REAL DATA

We further validate the capability of R2D2-Net with real datasets (i.e., TinyImageNet) and larger architectures (i.e., residual nets).

A. Experimental Setup

Datasets: We evaluate the R2D2-Net on standard computer vision datasets in comparison with existing methods. Table IV

TABLE III
SIMULATION RESULTS ON MSE AND INFERENCE ERROR UNDER THE R2D2-NET COMPARED WITH DIFFERENT BNN DESIGNS AND OPTIMIZATION ALGORITHMS ON MLP WITH DIFFERENT NUMBERS OF LAYERS $L = 0, 1, 2, 3$

Non-Trivial Features									
Scenario 1: Polynomial Case									
Prior	Inf. Alg.	$L = 0$		$L = 1$		$L = 2$		$L = 3$	
		MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.
Gauss	SVI	363.22	1.19	9.4	369.86	9.46	1056.8	15.4	2720.93
	SGMCMC	363.27	1.17	9.61	452.32	18.64	1034.01	89.98	2555.65
	SGLD	363.22	203.97	22.16	252.52	163.75	149.21	116.06	1021.6
SaS	SVI	1432.12	0.43	213.79	122.4	36.1	535.9	49.89	905.75
	SGMCMC	1364.4	0.13	131.28	153.58	94.84	434.9	145.85	718.72
	SGLD	370.54	174.1	320.98	65.28	168.11	110.75	44.65	336.49
HS	SVI	862.26	2.47	313.72	19.44	113.35	445.99	22.98	552.67
	SGMCMC	361.7	2.68	50.48	32.73	119.57	217.48	142.48	574.41
	SGMCMC	378.91	142.8	18.18	101.83	37.05	164.00	46.62	361.11
R2D2	SGLD	358.41	2.79	14.66	77.72	29.3	170.89	20.61	342.5
	SVGI	414.36	0.10	10.23	50.28	9.18	103.48	8.81	332.67
Scenario 2: Low-dimensional Non-linear Regression									
Prior	Inf. Alg.	$L = 0$		$L = 1$		$L = 2$		$L = 3$	
		MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.
Gauss	SVI	540.2	0.42	24.03	417.64	10.74	1279.85	12.57	2726.73
	SGMCMC	540.17	0.42	274.17	242.59	19.6	1109.39	181.44	2765.65
	SGLD	552.66	75.25	400.31	92.34	55.81	235.75	64.71	494.19
SaS	SVI	814.26	0.9	459.42	73.19	28.77	438.81	30.12	751.85
	SGMCMC	754.62	1.86	468.78	66.36	44.7	392.22	51.98	428.58
	SGLD	547.05	62.87	499.41	44.92	420.5	101.58	90.91	350.62
HS	SVI	560.73	4.85	435.69	34.06	136.51	265.6	31.3	479.22
	SGMCMC	546.9	3.13	322.29	32.46	102.57	172.64	118.96	363.33
	SGMCMC	509.29	4.48	35.09	34.66	93.42	249.92	130.47	513.81
R2D2	SGLD	509.38	0.31	31.50	63.14	55.44	183.46	52.56	352.72
	SVGI	509.23	0.26	22.38	46.48	10.26	84.52	12.24	348.51
Trivial Features									
Scenario 3: High-dimensional Non-linear Regression									
Prior	Inf. Alg.	$L = 0$		$L = 1$		$L = 2$		$L = 3$	
		MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.	MSE	Inf. Err.
Gauss	SVI	5.52	32.5	5.82	6948.52	5.57	8648.4	5.58	10519.14
	SGMCMC	6.04	33.32	7.21	9715.69	5.84	9605.35	7.12	10368.32
	SGLD	5.39	12.32	6.13	628.15	4.16	789.62	4.07	979.67
SaS	SVI	5.23	19.58	4.81	1100.96	4.62	1543.72	4.41	1803.17
	SGMCMC	5.24	20.24	5.3	1328.69	5.44	1138.89	5.15	1320.50
	SGLD	7.33	12.45	4.65	841.21	4.17	905.34	4.14	1120.53
HS	SVI	4.68	14.32	8.79	649.24	7.94	814.56	5.99	1023.07
	SGMCMC	4.49	36.05	6.04	1447.7	6.43	623.02	6.25	2009.38
	SGMCMC	7.1	11.71	8.51	711.03	4.61	602.83	4.89	942.12
R2D2	SGLD	4.68	11.63	4.83	532.27	6.00	547.56	4.16	787.50
	SVGI	4.35	11.47	4.63	480.26	4.15	510.87	4.06	719.15

SaS: spike-and-slab prior; SVI: stochastic variational inference; SGMCMC: stochastic gradient MCMC; SGLD: stochastic gradient langevin dynamics; SVGI: stochastic variational Gibbs inference.

We report the mean of the ten replicates.

provides a summary of the datasets. For **image classification**, we use CIFAR 10, CIFAR 100, and TinyImageNet as the benchmark datasets. We perform 5-fold cross-validation to evaluate each method. We use accuracy, macro F1 score, and area under the receiver operating curve (AUROC) as the evaluation metric, and report the mean and standard deviation of each metric.

For **uncertainty estimation**, we assess the performance of the neural networks using the out-of-distribution (OOD) detection task, with AUROC and the area under the precision-recall curve (AUPR) as the evaluation metrics. We treat the images in the CIFAR 10 [23] dataset as the in-distribution data and the images from the fashion MNIST, OMNIGLOT, and

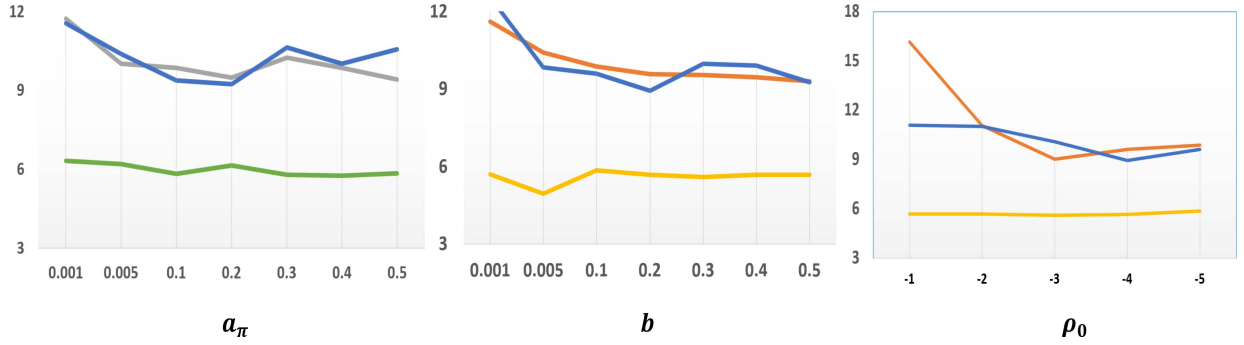


Fig. 4. Ablation studies of our method to different hyperparameters. We run the three simulation scenarios (S1–S3) with an R2D2 MLP with $L = 3$, and report the testing MSEs with respect to different values of hyperparameters a_π (left), b (middle), ρ_0 (right).

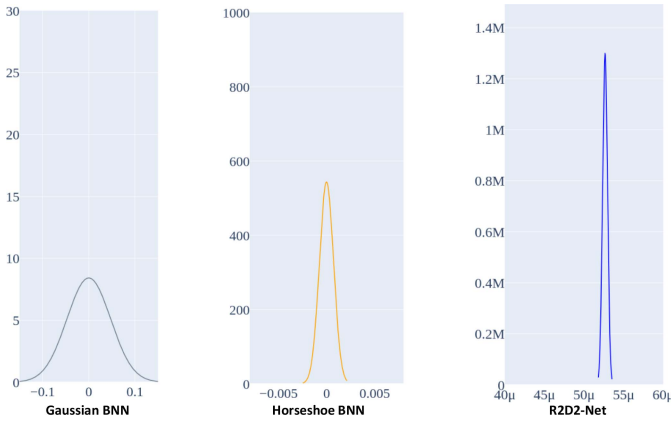


Fig. 5. Density plots of the weight samples of Gaussian BNN, Horseshoe BNN, and R2D2-Net. We choose the weights that have the least magnitude from the first layer of a three-layer MLP. We observe that R2D2-Net has the highest concentration rate at zero.

TABLE IV
SUMMARY OF DATASETS

Datasets	No. Classes	No. Training	No. Testing
MNIST	10	60,000	10,000
Fashion-MNIST	10	60,000	10,000
OMNIGLOT	50	13,180	19,280
SVHN	10	73,257	26,032
CIFAR-10	10	60,000	10,000
CIFAR-100	100	60,000	10,000
TinyImageNet	200	80,000	20,000
DRD	2	50	100

SVHN [45] as the OOD samples. In contrast to some existing approaches [28], [36], we train the classifier with in-distribution data only (i.e., the classifier does not see the OOD data during training).

Competitive Methods: Additionally, we consider a series of classical neural network designs for comparison:

- 1) *Frequentist CNN (Freq):* the original frequentist neural network architecture;
- 2) *MC Dropout [12] (MCD):* using repeated dropouts on trained weights to draw Monte Carlo samples of the weights of the BNNs (reproduced from [12]);

- 3) *RadialBNN [10] (Radial):* sampling from the hyperspherical coordinate system to resolve the problem in the original MFVI where the probability mass is far from the true mean; and it is implemented as a comparable optimization algorithm with the Gaussian prior;
- 4) *Deep Ensembles [25] (DE):* it uses a finite ensemble of deep neural networks to approximate the posterior weight distribution.

In addition to existing BNN designs, we add two entropy-based uncertainty estimation methods for comparison. Because we adopt entropy as the uncertainty metric (as this is a classic metric for classification uncertainty), the OOD performance may be slightly worse than their respective state-of-the-art (SOTA) performances: (1) **DPN** [28]: it assumes a Dirichlet distribution on the classification output and trains an OOD classifier by minimizing the KL divergence between the prior and posterior distributions; (2) **EDL** [36]: in addition to DPN [28], EDL trains the classifier with the cross-entropy loss and the KL divergence between the prior and posterior distributions.

B. Image Classification: R2D2 Shrinkage Improves Predictive Performance

Table V presents the image classification results of our R2D2-Net in comparison with existing methods.

We assess our method on standard image classification benchmarks — CIFAR 10, CIFAR 100, and TinyImageNet. We fix the model architecture as AlexNet [24] for fair comparison. Not only does our proposed method outperform the existing BNN designs, but it also occasionally outperforms the frequentist design. It is noteworthy that since BNNs impose a natural regularization on the weights, it is difficult for BNN designs to outperform their frequentist counterpart. This demonstrates that choosing the R2D2 prior can potentially lead to the best variable selection outcome. The R2D2 prior can select a suitable subset of weights with its shrinkage properties, while the frequentist design cannot. Hence, its prediction performance can be more satisfactory than the original frequentist design. On the other hand, from the standard deviations, we observe that adopting the Gibbs posterior of the R2D2 prior in variational inference would lead to more consistent and stable convergence results than other optimization algorithms.

TABLE V
IMAGE CLASSIFICATION RESULTS [%] OF OUR PROPOSED METHOD ON CIFAR 10 AND CIFAR 100 WITH THE ALEXNET [24]

Prior	Inference Alg.	CIFAR 10		CIFAR 100		TinyImageNet	
		AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy
Freq		92.70 (1.5)*	65.03 (1.4)	90.95 (0.2)	31.05 (0.4)	88.37 (1.3)	18.30 (0.4)
MCD	SVI	90.09 (0.2)	55.08 (0.6)	87.67 (1.3)	21.92 (1.1)	86.23 (1.7)	17.28 (1.5)
DE		90.67 (0.6)	62.41 (0.5)	87.97 (0.9)	24.63 (0.4)	86.25 (0.4)	13.73 (0.5)
Gauss	SVI	91.37 (1.2)	60.28 (1.5)	87.24 (1.2)	23.6 (0.6)	87.64 (0.2)	16.82 (0.9)
	MFVI	91.11 (0.9)	59.27 (1.1)	87.69 (0.9)	23.06 (0.2)	86.01 (0.3)	12.78 (0.6)
	Radial	91.22 (0.8)	63.24 (0.8)	89.20 (1.0)	25.70 (0.5)	84.35 (0.6)	12.12 (0.5)
	SGMCMC	83.79 (0.9)	41.91 (0.8)	78.71 (1.0)	10.32 (1.2)	84.37 (0.3)	10.19 (0.5)
	SGLD	89.34 (1.3)	52.22 (1.0)	79.21 (1.2)	12.31 (1.3)	84.42 (0.3)	10.15 (0.6)
SaS	SVI	91.69 (0.4)	60.84 (0.7)	89.68 (0.1)	29.01 (0.2)	85.37 (0.2)	14.49 (0.3)
	SGMCMC	84.10 (0.3)	43.03 (0.4)	76.44 (0.8)	8.35 (1.2)	76.74 (1.1)	4.54 (1.5)
	SGLD	86.14 (0.2)	45.13 (0.3)	80.74 (0.1)	9.89 (0.2)	71.27 (2.3)	3.87 (1.4)
HS	SVI	91.99 (0.8)	65.01 (0.3)	91.37 (0.2)	33.27 (0.3)	88.71 (2.0)	20.33 (1.2)
	SGMCMC	89.71 (0.5)	54.96 (0.8)	80.12 (0.1)	14.92 (0.3)	85.71 (1.3)	14.08 (1.1)
R2D2	SGMCMC	88.36 (0.3)	55.37 (0.4)	85.48 (0.1)	20.18 (0.3)	77.83 (0.9)	6.17 (1.2)
	SVGI	92.49 (0.2)	65.10 (0.02)*	92.48 (0.03)*	36.12 (0.5)*	88.76 (0.5)*	20.55 (0.4)*

Standard deviations are shown in brackets. **Boldface** represents the best performance among BNN designs, while * represents the best performance among all models.

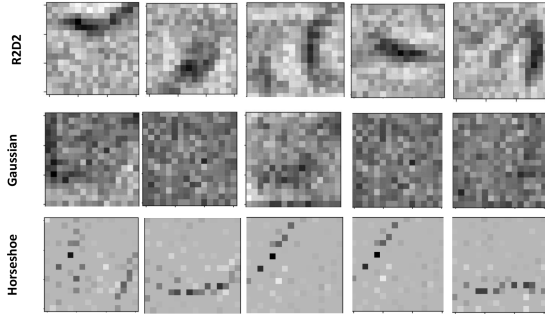


Fig. 6. Five largest-norm convolutional filters of the R2D2-Net, Gaussian BNN, and Horseshoe BNN. We use a simple CNN with one convolutional layer and one linear layer for illustrative purposes.

We visualize the five largest-norm filters of the Gaussian BNN, Horseshoe BNN, and the R2D2-Net to compare their capabilities to select features (Fig. 6). We observe that the largest filters of Gaussian BNN can capture the pattern of the image, while the noise is heavy. On the contrary, the Horseshoe BNN shrinks noise more effectively, while it suffers from feature losses (i.e., over-shrinkage). Compared to both above, the filters of R2D2-Net have less noise, while preserving most of the features. Since the R2D2 prior has heavier tails than the Horseshoe prior, it can preserve large signals in the filter weights and avoid over-shrinkage, as demonstrated by the difference in filter patterns in Fig. 6.

C. Uncertainty Estimation: R2D2 Shrinkage Captures Important Variance.

We further compare the performance of uncertainty estimation with the existing BNN designs. We additionally include two entropy-based uncertainty estimation methods: DPN [28] and EDL [36], which estimate uncertainties based on the assumption of Dirichlet distribution on latent probabilities. We use the classification entropy as the uncertainty measure. The entropy of classification is defined as

$$\mathbb{H}[p(\mu|\mathcal{D})] = - \sum_{c=1}^K p(\mu_c|\mathcal{D}) \log p(\mu_c|\mathcal{D}),$$

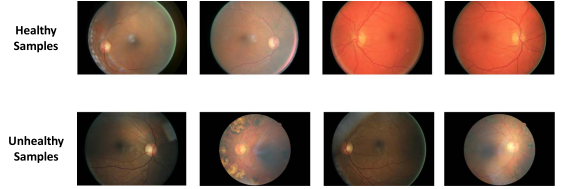


Fig. 7. In-distribution (healthy) and OOD (unhealthy) samples detected by our method.

where $P(\mu_c|\mathcal{D})$ is the predictive probability of class c , and K is the number of classes for classification. More information on baseline methods and uncertainty measures is given in the supplementary materials. We adopt the OOD detection task to evaluate the performance of the R2D2-Net for estimating the uncertainty in the data. The OOD detection aims to identify whether the input data are in-distribution or from a different dataset. Tables VI and VII present the AUROC and the AUPR of the R2D2-Net using the classification entropy as the uncertainty measure. We treat MNIST and CIFAR 10 as the in-distribution datasets and FashionMNIST, OMNIGLOT, and SVHN as the OOD datasets. We also include a medical image benchmark, Diabetic Retinopathy Detection (DRD) dataset, where we treat healthy samples as in-distribution data and unhealthy samples as the OOD dataset. Results are presented in Table VIII. Examples of the healthy and unhealthy samples detected can be found in Fig. 7. It is clear that our R2D2-Net demonstrates a satisfactory performance over the baseline methods. This shows that using an R2D2 prior on the weights can effectively shrink the noises in parameters while maintaining a non-trivial variance structure. These preserved variances can be used to represent the model-wise uncertainties. Hence, the R2D2-Net can produce more accurate uncertainty estimates than existing Bayesian and non-Bayesian approaches.

TABLE VI
THE OOD DETECTION PERFORMANCE OF THE R2D2-NET COMPARED WITH VARIOUS BNN DESIGNS UNDER THE LeNet [26], USING CIFAR 10 AS THE IN-DISTRIBUTION DATASET

Models/Priors	Fashion MNIST		OMNIGLOT		SVHN	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
DE	71.02	76.81	86.77	90.35	61.01	62.59
MCD	81.80	74.22	80.03	82.06	68.58	78.53
Gauss-SVI	74.49	86.78	78.58	81.82	70.57	79.30
Gauss-MFVI	85.45	79.55	89.17	91.64	76.02	85.08
Gauss-Radial	83.86	81.26	75.39	74.91	67.74	81.91
HS-SVI	80.76	75.99	86.66	90.18	70.11	78.39
SaS-SVI	91.55	88.89	89.03	88.64	78.25	85.11
DPN	87.07	83.75	87.07	83.75	57.48	77.76
EDL	89.26	86.16	66.53	67.12	69.57	83.74
R2D2-Net	92.85	94.09	91.95	92.25	79.84	89.24

The best performance among all methods is highlighted in boldface.

TABLE VII
THE OOD DETECTION PERFORMANCE OF THE R2D2-NET COMPARED WITH VARIOUS BNN DESIGNS UNDER THE LeNet [26], USING MNIST AS THE IN-DISTRIBUTION DATASET

Models	Fashion MNIST		OMNIGLOT		SVHN	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
DE	90.70	91.08	99.70	91.08	99.21	99.68
MCD	81.80	74.22	80.03	82.06	99.96	99.96
Gauss-SVI	98.36	98.36	99.17	99.38	98.95	99.10
Gauss-MFVI	98.52	98.48	98.94	99.11	99.91	99.96
Gauss-Radial	98.2	97.94	98.52	98.73	99.64	99.85
HS-SVI	80.76	75.99	99.06	99.65	99.35	98.74
SaS-SVI	98.28	98.16	99.57	99.57	99.90	99.96
DPN	98.70	98.80	99.96	99.96	99.96	99.96
EDL	73.43	80.22	72.61	81.42	63.43	85.09
R2D2-Net	98.75	98.84	99.64	99.65	99.31	99.69

The best performance among all methods is highlighted in boldface.

TABLE VIII
THE OOD DETECTION PERFORMANCE (%) ON DRD [1], WITH THE LeNet [26] ARCHITECTURE

Models	AUROC	AUPR
DE	59.67	56.58
MCD [12]	59.52	60.95
Gauss-SVI	69.12	74.80
Gauss-MFVI	63.56	66.79
Gauss-RAD	66.75	77.63
HS-SVI	69.80	77.88
SaS-SVI	70.64	75.19
DPN [28]	60.57	65.32
EDL [36]	53.01	58.22
R2D2-Net	71.04	78.11

TABLE IX
SUMMARY OF MODELS USED IN THE EXPERIMENTS AND THEIR DEPTH

Models	# Params (Freq)	# Params (Bayes)
LeNet	62K	124K
AlexNet	2.8M	5.6M
ResNet50	25.6M	51.2M
ResNet101	44.5M	89M

Freq stands for frequentist networks, and Bayes stands for the Bayesian counterparts.

satisfactorily on different architectures including modern architectures at the large scale (e.g., ResNet101).

D. Ablation Analysis

Performances with Various Architectures: We further apply R2D2 layers to different neural network architectures to evaluate the performance. We summarize the model architectures used in this experiment and their complexity in Table IX. We choose LeNet [26] and AlexNet [24] to benchmark the performance of different BNN designs. Table X presents the results on CIFAR 10. We observe that for most architectures our proposed BNN design achieves SOTA performance compared with existing BNN methods. This demonstrates that the R2D2-Net performs

VIII. DISCUSSION

We highlight the differences between our work and that of Zhang et al. [47]. Essentially, we adopt the prior in Zhang et al. and adapt the Gibbs inference algorithm to the deep learning context. The R2D2 prior has several desirable properties: the highest concentration rate at zero and the heaviest tails, which are crucial to the development of BNN models. A well-known work is the Horseshoe BNN [15] using the Horseshoe prior [6], which can effectively sparsify neural networks. We highlight the disadvantages of the Horseshoe prior (i.e., the lighter tail

TABLE X
IMAGE CLASSIFICATION RESULTS OF R2D2-NET WITH DIFFERENT MODEL ARCHITECTURES COMPARED TO DIFFERENT COMBINATIONS OF PRIORS AND INFERENCE ALGORITHMS ON THE CIFAR 10 DATASET

Model/Prior	Inf. Alg.	LeNet		AlexNet		ResNet50		ResNet101	
		AUROC	ACC	AUROC	ACC	AUROC	ACC	AUROC	ACC
Freq		91.24	61.21	92.70*	65.03	96.23	79.25*	96.75	79.20
DE		93.75	63.11	90.06	62.94	96.60	77.62	96.82	79.01
MCD		91.50	58.76	91.21	62.76	96.44	77.24	96.83	79.54*
Gauss	SVI	91.31	60.03	91.21	62.64	95.59	73.62	95.53	73.34
	MFVI	92.41	63.39	91.11	59.27	96.48	78.19	95.65	73.37
	Radial	91.74	61.29	91.22	63.24	95.39	74.03	96.34	72.99
	SGMCMC	80.34	33.86	83.39	42.00	96.80	75.05	96.19	73.07
	SGLD	81.78	36.47	89.34	52.22	88.08	48.00	88.09	49.34
SaS	SVI	91.87	60.50	91.69	60.84	94.18	74.15	94.17	70.26
	SGMCMC	81.63	35.06	83.10	40.82	96.71	78.91	96.76	79.36
	SGLD	88.76	34.00	86.23	44.29	93.03	58.26	90.18	53.49
HS	SVI	92.42*	60.13	91.99	65.01	96.96	78.90	97.08	79.14
	SGMCMC	86.98	49.35	86.32	49.63	95.90	78.17	95.74	75.73
R2D2-Net	SVGI	90.43	61.53*	92.49	65.10*	96.97*	79.10	97.12*	79.20

and lower concentration rate at zero) and show that the R2D2 prior is a better choice for variable shrinkage in neural networks.

R2D2-Net for Sparsity-Induced Deep Learning: Sparsity-induced deep learning aims to resolve the excessive over-parameterization of modern deep neural networks [33], [40]. Bayesian methods impose sparsity on neural network weights via spike-and-slab priors, whereas the posterior contraction rate around the optimal predictor and the posterior consistency are the key factors to a good choice of prior. The R2D2 prior has the near-minimax posterior contraction rate as the spike-and-slab priors. It also yields a strongly consistent posterior [47], which makes it a competitive candidate over the existing spike-and-slab priors for sparsity-induced deep learning.

The R2D2 prior is also able to facilitate neural network sparsification, which aims to prune the neural network for smaller time complexity. Existing works widely adopt the variational dropout or Horseshoe prior to shrink unnecessary neurons to achieve neural network sparsification. However, the relatively low concentration rates around zero of these priors make the pruning process ineffective, while the light tails of these priors cause some important features to be overshrunk. Since the R2D2 prior has a relatively higher concentration rate around zero and a heavier tail compared to existing sparsity-induced priors, it also performs superiorly in neural network sparsification.

The Analogy of R^2 : The R2D2 prior is loosely based on a prior on the goodness-of-fit in a regression model. Under the deep learning settings, the analogy of R^2 may not be applicable in the settings of DNN/BNN, as it is difficult to interpret the “goodness-of-fit”. However, the properties of the R2D2 prior (i.e., the high spiking rate and heavy tail) still hold as we place such a prior on each individual neuron.

Different Shrinkage Profiles: Recently, many new shrinkage priors have been proposed by introducing more parameters in the prior to more flexibly determine the shrinkage behaviours. For instance, the triple gamma prior [5] added an extra layer to the double gamma prior to more specifically capture the shrinkage patterns. It possesses the concentration rate of $\mathcal{O}((1/\sqrt{\beta})^{1-2a^\zeta})$ at zero, which is comparatively lower than that of the R2D2 prior. And it possesses a tail thickness of $\mathcal{O}((1/\sqrt{\beta})^{2c^\zeta+1})$ which is comparatively thinner than that of the R2D2 prior. In modern deep learning settings, effective shrinkage and feature

preservation are important for optimal DNN and BNN performance, hence the R2D2 prior remains the competitive candidate for priors on neural network weights.

Hyperparameters: We adopt grid search to perform hyperparameter tuning. However, Gruber and Kastner [18] proposed a data-driven approach such that the key hyperparameters (e.g., shrinkage) can be estimated from data. Hence, including the parameters into the stochastic back-propagation may be feasible in future extensions of the R2D2-Net.

Limitations: Our proposed method tackles the scalability constraints of Bayesian neural networks and is validated on some modern architectures (e.g., ResNet-based). Due to a lack of mature research in Bayesian designs of more modern architectures (such as Bayesian attention mechanisms and transformers), the extension of R2D2-Net to these architectures would be non-trivial, although it opens the possibilities of Bayesian foundation models. On the other hand, we take Monte Carlo samples of weights from the posterior distribution, which could potentially be a computational burden. Integration of recent efficient sampling techniques of BNNs [9], [11] would decrease the posterior inference complexity.

Moreover, the Gibbs sampling based on conditional distributions may not tackle the multimodal posteriors satisfactorily. With the recent development of multimodal Gibbs inference [8], it would be interesting to be integrated into the SVI paradigm in future works.

IX. CONCLUSION

In this work, we propose a novel BNN design — the R2D2-Net. We develop a variational Gibbs inference algorithm to better approximate the posterior distributions of the network weights. Extensive experiments on synthetic and real datasets validate the performance of our proposed BNN design on both image classification and image uncertainty estimation tasks. Our proposed method can be potentially applied to different data domains, such as graphs and Bayesian graph neural networks. The R2D2-Net also has great real application potential in reinforcement learning, recommendation systems, and biomedical imaging for its capability in predictive inference and uncertainty estimation.

ACKNOWLEDGMENT

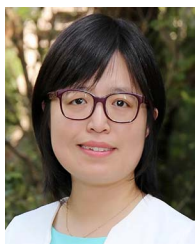
The authors would like to thank the associate editor and referees for many constructive suggestions that have significantly improved the paper.

REFERENCES

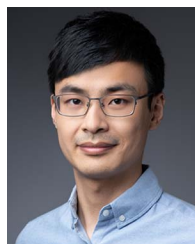
- [1] S. Farquhar, A. Filos, A. N. Gomez, and T. G. J. Rudner, "A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks," *Stat.*, vol. 22, 2019, Art. no. 1050.
- [2] A. Armagan, D. B. Dunson, J. Lee, W. U. Bajwa, and N. Strawn, "Posterior consistency in linear models under shrinkage priors," *Biometrika*, vol. 1000, no. 4, pp. 1011–1018, 2013.
- [3] A. Bhadra, J. Datta, N. G. Polson, and B. Willard, "The horseshoe estimator of ultra-sparse signals," *Bayesian Anal.*, vol. 120, no. 4, pp. 1105–1131, 2017.
- [4] H. Bolcskei, P. Grohs, G. Kutyniok, and P. Petersen, "Optimal approximation with sparsely connected deep neural networks," *SIAM J. Math. Data Sci.*, vol. 10, no. 1, pp. 8–45, 2019.
- [5] A. Cadonna, S. Frühwirth-Schnatter, and P. Knaus, "Triple the gamma—A unifying shrinkage prior for variance and variable selection in sparse state space and TVP models," *Econometrics*, vol. 80, no. 2, 2020, Art. no. 20.
- [6] C. M. Carvalho, N. G. Polson, and J. G. Scott, "Handling sparsity via the horseshoe," in *Proc. Artif. Intell. Statist.*, 2009, pp. 73–80.
- [7] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1683–1691.
- [8] W. Chen, M. Zhang, B. Paige, J. M. Hernández-Lobato, and D. Barber, "Diffusive Gibbs sampling," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 7731–7747, doi: [10.5555/3692070.3692373](https://doi.org/10.5555/3692070.3692373).
- [9] M. Dusenberry et al., "Efficient and scalable Bayesian neural nets with rank-1 factors," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2782–2792.
- [10] S. Farquhar, M. A. Osborne, and Y. Gal, "Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1352–1362.
- [11] G. Franchi, A. Bursuc, E. Aldea, S. Dubuisson, and I. Bloch, "Encoding the latent posterior of Bayesian neural networks for uncertainty quantification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2027–2040, Apr. 2024.
- [12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
- [13] D. Gan, G. Yin, and Y. D. Zhang, "The GR2D2 estimator for the precision matrices," *Brief. Bioinf.*, vol. 230, no. 6, 2022, Art. no. bbac426.
- [14] P. N. Garner and S. Tong, "A Bayesian approach to recurrence in neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 430, no. 8, pp. 2527–2537, Aug. 2021.
- [15] S. Ghosh and F. Doshi-Velez, "Model selection in Bayesian neural networks via horseshoe priors," *J. Mach. Learn. Res.*, vol. 200, no. 182, pp. 1–46, 2019.
- [16] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *J. Roy. Stat. Soc. Ser. B: Stat. Methodol.*, vol. 730, no. 2, pp. 123–214, 2011.
- [17] M. Griffin and P. D. Hoff, "Structured shrinkage priors," *J. Comput. Graphical Statist.*, vol. 330, no. 1, pp. 1–14, 2024.
- [18] L. Gruber and G. Kastner, "Forecasting macroeconomic data with Bayesian VARs: Sparse or dense? it depends!," *Int. J. Forecasting*, 2025, doi: [10.1016/j.ijforecast.2025.02.001](https://doi.org/10.1016/j.ijforecast.2025.02.001).
- [19] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding," in *Proc. 4th Int. Conf. Learn. Representations*, San Juan, Puerto Rico, 2016.
- [20] M. D. Hoffman, D. M. C. B. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, 2013.
- [21] L. V. Jospin, "Hands-on Bayesian neural networks—A tutorial for deep learning users," *IEEE Comput. Intell. Mag.*, vol. 17.2, pp. 29–48, 2022.
- [22] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5580–5590.
- [23] A. Krizhevsky et al., "Learning Multiple Layers of Features from Tiny Images," Master's thesis, University of Tront, 2009.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 84–90.
- [25] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6405–6416.
- [26] Y. LeCun et al., "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1989.
- [27] C. Louizos, K. Ullrich, and M. Welling, "Bayesian compression for deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
- [28] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.
- [29] T. Matsubara, C. J. Oates, and F.-X. Briol, "The ridgelet prior: A covariance function approach to prior specification for Bayesian neural networks," *J. Mach. Learn. Res.*, vol. 220, no. 157, pp. 1–57, 2021.
- [30] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2498–2507.
- [31] C. Nemeth and P. Fearnhead, "Stochastic gradient Markov chain monte carlo," *J. Amer. Stat. Assoc.*, vol. 1160, no. 533, pp. 433–450, 2021.
- [32] J. Piironen and A. Vehtari, "Sparsity information and regularization in the horseshoe and other shrinkage priors," *Electron. J. Statist.*, vol. 110, no. 2, pp. 5018–5051, 2017.
- [33] N. G. Polson and V. Ročková, "Posterior concentration for sparse deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 938–949.
- [34] A.-L. Popkes et al., "Interpretable outcome prediction with sparse Bayesian neural networks in intensive care," 2019, *arXiv:1905.02599*.
- [35] T. G. J. Rudner, Z. Chen, Y. W. Teh, and Y. Gal, "Tractable function-space variational inference in Bayesian neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 22686–22698.
- [36] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3183–3193.
- [37] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to Bayesian convolutional neural network with variational inference," 2019, *arXiv:1901.02731*.
- [38] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," in *Proc. 34th Conf. Uncertainty Artif. Intell.*, Monterey, CA, USA, 2018, pp. 560–569, [Online]. Available: <http://auai.org/uai2018/proceedings/papers/207.pdf>
- [39] S. Srinivas and R. V. Babu, "Generalized dropout," 2016, *arXiv:1611.06791*.
- [40] Y. Sun, Q. Song, and F. Liang, "Consistent sparse deep learning: Theory and computation," *J. Amer. Stat. Assoc.*, vol. 1170, no. 540, pp. 1981–1995, 2022.
- [41] Y. Sun, Q. Song, and F. Liang, "Learning sparse deep neural networks with a spike-and-slab prior," *Statist. Probability Lett.*, vol. 180, 2022, Art. no. 109246.
- [42] M. Tomczak, S. Swaroop, A. Foong, and R. Turner, "Collapsed variational bounds for Bayesian neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 25412–25426.
- [43] B.-H. Tran, S. D. R. Milios, and M. Filippone, "All you need is a good functional prior for Bayesian deep learning," *J. Mach. Learn. Res.*, vol. 230, no. 74, pp. 1–56, 2022.
- [44] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 681–688.
- [45] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
- [46] H. Yu, M. Hartmann, B. Williams, and A. Klami, "Scalable stochastic gradient riemannian Langevin dynamics in non-diagonal metrics," *Trans. Mach. Learn. Res.*, vol. 8, 2023, Art. no. 2023.
- [47] Y. D. Zhang, B. P. Naughton, H. D. Bondell, and B. J. Reich, "Bayesian regression using a prior on the model fit: The R2-D2 shrinkage prior," *J. Amer. Stat. Assoc.*, vol. 117, pp. 862–874, 2020.



Tsai Hor Chan received the BSc degree in actuarial science from the University of Hong Kong, in 2020. He is currently working toward the PhD degree with the Department of Statistics and Actuarial Science, University of Hong Kong. His research interest focuses in Bayesian nonparametrics, statistical deep learning, medical informatics, and computer vision.



Dora Yan Zhang received the PhD degree in statistics from North Carolina State University, in 2016. She is currently an assistant professor in Department of Statistics and Actuarial Science, University of Hong Kong. Her main research areas include statistical genetics, biostatistics, bioinformatics, health informatics, and public health.



Lequan Yu (Member, IEEE) received the BEng degree from the Department of Computer Science and Technology, Zhejiang University, Hangzhou, China, in 2015, and the PhD degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2019. He conducted his postdoctoral training with Stanford University during 2019–2021. He is currently an assistant professor in Department of Statistics and Actuarial Science, University of Hong Kong. His research focuses on medical image analysis, computer vision,

machine learning and AI in healthcare.



Guosheng Yin (Fellow IEEE) received the PhD degree in biostatistics from the University of North Carolina, Chapel Hill, in 2003. In 2003–2009, he worked as assistant professor and also associate professor (tenured) in Department of Biostatistics with University of Texas M.D. Anderson Cancer Center. He worked as Head of Department of Statistics and Actuarial Science at University of Hong Kong in 2017–2023 and then as chair in statistics with Imperial College London, in 2023–2024. He is currently chair professor of Statistics and Patrick SC Poon Endowed

professor in Department of Statistics and Actuarial Science with the University of Hong Kong. His main research areas include AI, machine learning, clinical trial methodology, adaptive design, Bayesian methods, and survival analysis. He has published more than 260 peer-reviewed papers and two books in the areas of clinical trial design and methods.