Cui, Tianyu; Havulinna, Aki S.; Marttinen, Pekka; Kaski, Samuel

Informative Bayesian Neural Network Priors for Weak Signals

Published in:
Bayesian Analysis

DOI:
10.1214/21-BA1291

Published: 01/01/2021

Document Version
Publisher's PDF, also known as Version of record

Bayesian Analysis (2021) **TBA**, Number TBA, pp. 1–31

Informative Bayesian Neural Network Priors for Weak Signals*

Tianyu Cui[†], Aki Havulinna[‡,§], Pekka Marttinen[†,‡,‖], and Samuel Kaski[†,¶,‖]

Abstract. Encoding domain knowledge into the prior over the high-dimensional weight space of a neural network is challenging but essential in applications with limited data and weak signals. Two types of domain knowledge are commonly available in scientific applications: 1. feature sparsity (fraction of features deemed relevant); 2. signal-to-noise ratio, quantified, for instance, as the proportion of variance explained. We show how to encode both types of domain knowledge into the widely used Gaussian scale mixture priors with Automatic Relevance Determination. Specifically, we propose a new joint prior over the local (i.e., feature-specific) scale parameters that encodes knowledge about feature sparsity, and a Stein gradient optimization to tune the hyperparameters in such a way that the distribution induced on the model's proportion of variance explained matches the prior distribution. We show empirically that the new prior improves prediction accuracy compared to existing neural network priors on publicly available datasets and in a genetics application where signals are weak and sparse, often outperforming even computationally intensive cross-validation for hyperparameter tuning.

Keywords: informative prior, neural network, proportion of variance explained, sparsity.

1 Introduction

Neural networks (NNs) have achieved state-of-the-art performance on a wide range of supervised learning tasks with high a signal-to-noise ratio (S/N), such as computer vision (Krizhevsky et al., 2012) and natural language processing (Devlin et al., 2018). However, NNs often fail in scientific applications where domain knowledge is essential, e.g., when data are limited or the signal is extremely weak and sparse. Applications in genetics often fall into the latter category and are used as the motivating example for our derivations. Bayesian approach (Gelman et al., 2013) has been of interest in the NN community because of its ability to incorporate domain knowledge into reasoning and to provide principled handling of uncertainty. Nevertheless, it is still largely an open

*This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI, grants 319264, 292334, 286607, 294015, 336033, 315896, 341763), and EU Horizon 2020 (INTERVENE, grant no. 101016775). We also acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT.

[†]Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Finland, tianyu.cui@aalto.fi

[‡]Finnish Institute for Health and Welfare (THL), Finland

[§]Institute for Molecular Medicine Finland, FIMM-HiLIFE, Helsinki, Finland

[¶]Department of Computer Science, University of Manchester, UK

[‖]Equal contribution.

 https://doi.org/10.1214/21-BA1291

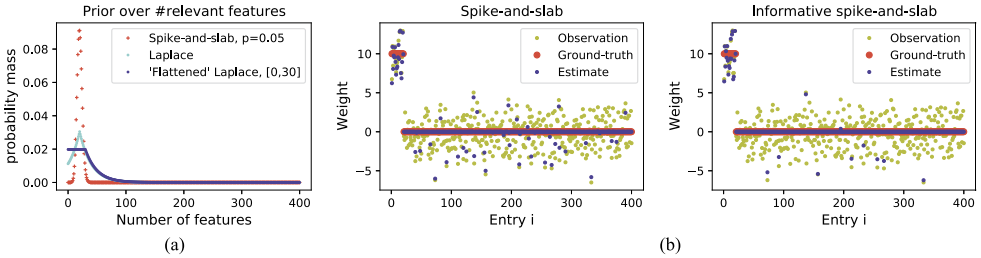

Figure 1: **a)** A spike-and-slab prior with slab probability $p = 0.05$ induces a binomial distribution on the number of relevant features. The proposed informative spike-and-slab can encode a spectrum of alternative beliefs, such as a discretized or 'flattened' Laplace (for details, see Section 3). **b)** The informative spike-and-slab prior can remove false features more effectively than the vanilla spike-and-slab prior with correct slab probability, where features are assumed independent (see Section 6.1).

question how to encode domain knowledge into the prior over Bayesian neural network (BNN) weights, which are often high-dimensional and uninterpretable.

We study the family of Gaussian scale mixture (GSM) (Andrews and Mallows, 1974) distributions, which are widely used as priors for BNN weights. A particular example of interest is the spike-and-slab prior (Mitchell and Beauchamp, 1988)

$$w_{ij}^{(l)} | \sigma, \lambda_i^{(l)}, \tau_i^{(l)} \sim \mathcal{N}(0, \sigma^{(l)2} \lambda_i^{(l)2} \tau_i^{(l)2}); \quad \tau_i^{(l)} \sim \text{Bernoulli}(p), \tag{1}$$

where $w_{ij}^{(l)}$ represents the NN weight from node i in layer l to node j in layer $l+1$. The hyper-parameters $\{\sigma^{(l)}, \lambda_i^{(l)}, p\}$ are often given non-informative hyper-priors (Neal, 2012), such as the inverse Gamma on $\sigma^{(l)}$ and $\lambda_i^{(l)}$, or optimized using cross-validation (Blundell et al., 2015). In contrast, we propose determining the hyper-priors according to two types of domain knowledge often available in scientific applications: ballpark figures on feature sparsity and the signal-to-noise ratio. Feature sparsity refers to the expected fraction of features used by the model. For example, it is known that less than 2% of the genome encodes for genes, which may inform the expectation on the fraction of relevant features in a genetics application. A prior on the signal-to-noise ratio specifies the amount of target variance expected to be explained by the chosen features, and it can be quantified as the proportion of variance explained (PVE) (Glantz et al., 1990). For instance, one gene may explain a tiny fraction of the variance of a given phenotype (prediction target in genetics, e.g. the height of an individual), i.e., the PVE of a gene may be as little as 1%.

Existing scalable sparsity-inducing BNN priors, such as the spike-and-slab prior, are restricted in the forms of prior beliefs about sparsity they can express: conditionally on the slab probability p the number of relevant features follows a Binomial distribution. Specifying a Beta hyper-prior on p could increase flexibility, but this still is more restricted and less intuitive than specifying any distribution directly on the number of

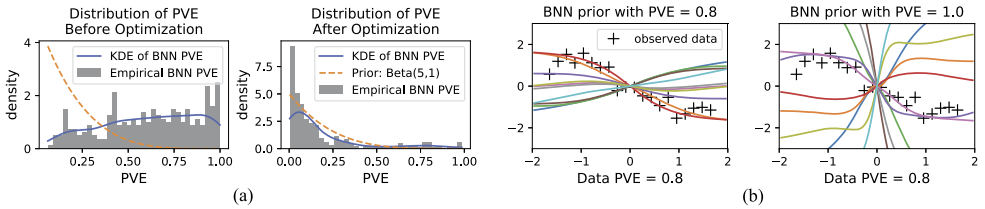

Figure 2: **a)** The empirical distribution and corresponding kernel density estimation (KDE) of the proportion of variance explained (PVE) for a BNN, obtained by simulating from the model, before and after optimizing the hyperparameters according to the prior belief on the PVE. **b)** The data with PVE=0.8 in its generating process are more likely to be generated by a BNN when the mode of the PVE is approximately correctly (left) than incorrectly (right). Colored lines are functions sampled from the BNN (for details, see Section 4).

relevant features, and in practice in the BNN literature a point estimate for p is used. The value of p is either set manually, cross-validated, or optimized as part of MAP estimation (Deng et al., 2019). Moreover, the weights for different features or nodes are (conditionally) independent in (1); thus, incorporating correct features will not help remove false ones. In this paper, we propose a novel informative hyper-prior over the feature inclusion indicators $\tau_i^{(l)}$, called informative spike-and-slab, which can directly model any distribution on the number of relevant features (Figure 1a). In addition, unlike the vanilla spike-and-slab, the $\tau_i^{(l)}$ for different features i are dependent in the new informative spike-and-slab, and consequently false features are more likely to be removed when correct features are included, which can be extremely beneficial when the noise level is high, as demonstrated with a toy example in Figure 1b.

The PVE assumed by a BNN affects the variability of functions drawn from the prior (Figure 2b). Intuitively, when the PVE of a BNN is close to the correct PVE, the model is more likely to recover the underlying data generating function. The distribution of PVE assumed by a BNN is induced by the prior on the model's weights, which in turn is affected by all the hyper-parameters. Thus, hyper-parameters that do not affect feature sparsity, e.g. $\lambda_i^{(l)}$, can be used to encode domain knowledge about the PVE. We propose a scalable gradient-based optimization approach to match the model's PVE with the prior belief on the PVE, e.g., a Beta distribution, by minimizing the Kullback–Leibler divergence between the two distributions w.r.t. chosen hyper-parameters using the Stein gradient estimator (Li and Turner, 2018) (Figure 2a). Although it has been demonstrated that using cross-validation to specify hyper-parameters, e.g. the global scale in the mean-field prior, is sufficient for tasks with a high S/N and a large dataset (Wilson and Izmailov, 2020), we empirically show that being informative about the PVE can improve performance in low S/N and small data regimes, even without computationally intensive cross-validation.

The structure of this paper is the following. Section 2 reviews required background on Bayesian neural networks and Stein gradients. In Section 3, we describe our novel joint

hyper-prior over the local scales which explicitly encodes feature sparsity. In Section 4, we present the novel optimization algorithm to tune the distribution of a model's PVE according to prior knowledge. Section 5 provides the variational inference algorithm for BNNs. Section 6 reviews in detail a large body of related literature on BNNs. Thorough experiments with synthetic and real-world data sets are presented in Section 7, demonstrating the benefits of the method. Finally, Section 8 concludes, including discussion on limitations of our method as well as suggested future directions.

2 Background

2.1 Proportion of Variance Explained

In regression tasks, we assume that the data generating process takes the form

$$y = f(\mathbf{x}; \mathbf{w}) + \epsilon, \tag{2}$$

where $f(\mathbf{x}; \mathbf{w})$ is the unknown target function, and ϵ is the unexplainable noise. The Proportion of Variance Explained (PVE) (Glantz et al., 1990) of $f(\mathbf{x}; \mathbf{w})$ on dataset $\{\mathbf{X}, \mathbf{y}\}$ with input $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and outputs $\mathbf{y} = \{y^{(1)}, \ldots, y^{(N)}\}$, also called the coefficient of determination (R^2) in linear regression, is

$$\mathrm{PVE}(\mathbf{w}) = 1 - \frac{\sum_{i=1}^{N}(y^{(i)} - f(\mathbf{x}^{(i)}; \mathbf{w}))^2}{\sum_{i=1}^{N}(y^{(i)} - \bar{y})^2}. \tag{3}$$

The PVE is commonly used to measure the impact of features $\mathbf{x}$ on the prediction target y, for example in genomics (Marttinen et al., 2014). In general, PVE should be in $[0, 1]$ because the predictions' variance should not exceed that of the data. However, this may not hold at test time for non-linear models such as neural networks if the models have overfitted to the training data, in which case the variance of the residual can exceed the variance of target in the test set. By placing a prior over $\mathbf{w}$ whose $\mathrm{PVE}(\mathbf{w})$ concentrates around the PVE of the data generating process, the hypothesis space of the prior can be made more concentrated around the true model, which eventually yields a more accurate posterior.

2.2 Bayesian neural networks

Variational posterior approximation

Bayesian neural networks (BNNs) (MacKay, 1992; Neal, 2012) are defined by placing a prior distribution on the weights $p(\mathbf{w})$ of a NN. Then, instead of finding point estimators of weights by minimizing a cost function, which is the normal practice in NNs, a posterior distribution of the weights is calculated conditionally on the data. Let $f(\mathbf{x}; \mathbf{w})$ denote the output of a BNN and $p(y|\mathbf{x}, \mathbf{w}) = p(y|f(\mathbf{x}; \mathbf{w}))$ the likelihood. Then, given a dataset of inputs $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and outputs $\mathbf{y} = \{y^{(1)}, \ldots, y^{(N)}\}$, training a BNN means computing the posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$. Variational inference can be used to approximate the intractable $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ with a simpler distribution, $q_\phi(\mathbf{w})$, by minimizing

$\mathrm{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}|\mathbf{X},\mathbf{y}))$. This is equivalent to maximizing the Evidence Lower BOund (ELBO) (Bishop, 2006)

$$\mathcal{L}(\phi) = \mathcal{H}(q_\phi(\mathbf{w})) + \mathrm{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{y},\mathbf{w}|\mathbf{X})]. \tag{4}$$

The first term in (4) is the entropy of the approximated posterior, which can be calculated analytically for many choices of $q_\phi(\mathbf{w})$. The second term is often estimated by the reparametrization trick (Kingma and Welling, 2013), which reparametrizes the approximated posterior $q_\phi(\mathbf{w})$ using a deterministic and differentiable transformation $\mathbf{w} = g(\xi;\phi)$ with $\xi \sim p(\xi)$, such that $\mathrm{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{y},\mathbf{w}|\mathbf{X})] = \mathrm{E}_{p(\xi)}[\log p(\mathbf{y}, g(\xi;\phi)|\mathbf{X})]$, which can be estimated by Monte Carlo integration.

Gaussian scale mixture priors over weights

The *Gaussian scale mixture* (GSM) (Andrews and Mallows, 1974) is defined to be a zero mean Gaussian conditional on its scales. In BNNs, it has been combined with *Automatic Relevance Determination* (ARD) (MacKay, 1994), a widely used approach for feature selection in non-linear models. An ARD prior in BNNs means that all of the outgoing weights $w_{ij}^{(l)}$ from node i in layer l share a same scale $\lambda_i^{(l)}$ (Neal, 2012). We define the input layer as layer 0 for simplicity. A GSM ARD prior on each weight $w_{ij}^{(l)}$ can be written in a *hierarchically parametrized* form as follows:

$$w_{ij}^{(l)}|\lambda_i^{(l)},\sigma^{(l)} \sim \mathcal{N}(0,\sigma^{(l)2}\lambda_i^{(l)2}); \quad \lambda_i^{(l)} \sim p(\lambda_i^{(l)};\theta_\lambda), \tag{5}$$

where $\sigma^{(l)}$ is the layer-wise global scale shared by all weights in layer l, which can either be set to a constant value or estimated using non-informative priors, and $p(\lambda_i^{(l)};\theta_\lambda)$ defines a hyper-prior on the local scales. The marginal distribution of $w_{ij}^{(l)}$ can be obtained by integrating out the local scales given $\sigma^{(l)}$:

$$p(w_{ij}^{(l)}|\sigma^{(l)}) = \int \mathcal{N}(0,\sigma^{(l)2}\lambda_i^{(l)2})p(\lambda_i^{(l)};\theta_\lambda)\mathrm{d}\lambda_i^{(l)}. \tag{6}$$

The hyper-prior of local scales $p(\lambda_i^{(l)};\theta_\lambda)$ determines the distribution of $p(w_{ij}^{(l)}|\sigma^{(l)})$. For example, a Dirac delta distribution $\delta(\lambda_i^{(l)}-1)$ reduces $p(w_{ij}^{(l)}|\sigma^{(l)})$ to a Gaussian with mean zero and variance $\sigma^{(l)2}$, whereas an inverse Gamma distribution on $\lambda_i^{(l)}$ makes $p(w_{ij}^{(l)}|\sigma^{(l)})$ equal to a student-t distribution (Gelman et al., 2013; Fortuin et al., 2021).

Many sparsity inducing priors in the Bayesian paradigm can be interpreted as Gaussian scale mixture priors with additional local scale variables $\tau_i^{(l)}$:

$$w_{ij}^{(l)}|\lambda_i^{(l)},\tau_i^{(l)},\sigma^{(l)} \sim \mathcal{N}(0,\sigma^{(l)2}\lambda_i^{(l)2}\tau_i^{(l)2}); \quad \lambda_i^{(l)} \sim p(\lambda_i^{(l)};\theta_\lambda); \quad \tau_i^{(l)} \sim p(\tau_i^{(l)};\theta_\tau). \tag{7}$$

For example, the spike-and-slab prior (Mitchell and Beauchamp, 1988) is the 'gold standard' for sparse models and it introduces binary local scales $\tau_i^{(l)}$, interpreted as feature inclusion indicators, such that

$$w_{ij}^{(l)}|\lambda_i^{(l)},\tau_i^{(l)},\sigma^{(l)} \sim (1-\tau_i^{(l)})\delta(w_{ij}^{(l)}) + \tau_i^{(l)}\mathcal{N}(0,\sigma^{(l)2}\lambda_i^{(l)2}), \tag{8}$$

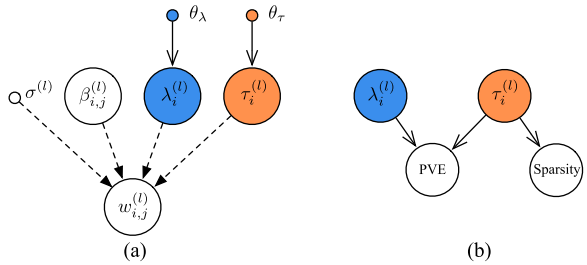

Figure 3: **a)** Non-centered parametrization of the GSM prior. **b)** The model's PVE is determined by $p(\lambda_i^{(l)}; \theta_\lambda)$ and $p(\tau_i^{(l)}; \theta_\tau)$ jointly, but sparsity is determined by $p(\tau_i^{(l)}; \theta_\tau)$ alone. Therefore, we determine the distribution $p(\tau_i^{(l)}; \theta_\tau)$ according to the prior knowledge about sparsity, and then tune $p(\lambda_i^{(l)}; \theta_\lambda)$ conditionally on the previously selected $p(\tau_i^{(l)}; \theta_\tau)$ to accommodate the prior knowledge about the PVE.

where $\tau_i^{(l)} \sim \text{Bernoulli}(p)$. In (7), the weight $w_{ij}^{(l)}$ equals 0 with probability $1 - p$ (the spike) and with probability p it is drawn from another Gaussian (the slab). Continuous local scales $\tau_i^{(l)}$ lead to other shrinkage priors, such as the horseshoe (Piironen and Vehtari, 2017a) and the Dirichlet-Laplace (Bhattacharya et al., 2015), which are represented as global-local (GL) mixtures of Gaussians.

Gaussian scale mixtures, i.e., (7), are often written with an equivalent *non-centered parametrization* (Papaspiliopoulos et al., 2007) (Figure 3a),

$$w_{ij}^{(l)} = \sigma^{(l)} \beta_{ij}^{(l)} \lambda_i^{(l)} \tau_i^{(l)}; \quad \beta_{ij}^{(l)} \sim \mathcal{N}(0, 1); \quad \lambda_i^{(l)} \sim p(\lambda_i^{(l)}; \theta_\lambda); \quad \tau_i^{(l)} \sim p(\tau_i^{(l)}; \theta_\tau), \tag{9}$$

which has a better posterior geometry for inference (Betancourt and Girolami, 2015) than the *hierarchical parametrization*. Therefore, non-centered parametrization has been widely used in the BNN literature (Louizos et al., 2017; Ghosh et al., 2018), and we follow this common practice as well.

The hyper-parameter θ_τ in $p(\tau_i^{(l)}; \theta_\tau)$ controls the prior sparsity level, often quantified by the number of relevant features. However, for continuous hyper-priors, e.g. the half-Cauchy prior in the horseshoe, which do not force weights exactly to zero, it is not straightforward to select the hyper-parameter θ_τ according to prior knowledge. Piironen and Vehtari (2017b) propose to choose θ_τ based on the *effective* number of features defined as the total shrinkage in linear regression. However, this definition relies heavily on the linearity assumption. Thus it is non-trivial to apply on nonlinear models, such as neural networks. On the other hand, the existing discrete hyper-priors on $\tau_i^{(l)}$ model only restricted forms of sparsity, such as the Binomial distribution in the spike-and-slab prior in (8). In Section 3, we propose an informative spike-and-slab prior consisting of a new class of discrete hyper-priors over the local scales $\tau_i^{(l)}$, capable of representing any type of sparsity. Moreover, the informative spike-and-slab makes $\tau_i^{(l)}$ dependent, which leads to a heavier penalization on false features than in the independent priors, such as the vanilla spike-and-slab, after correct features have been included.

It is well known that the scale parameter of the fully factorized Gaussian prior on BNNs weights affects the variability of the functions drawn from the prior (Neal, 2012), and thus the PVE. When the PVE of the BNN has much probability around the correct PVE, the model is more likely to recover the true data generating mechanism (demonstration in Figure 2). As we will show in Section 4, for a BNN with the GSM prior defined in (9), the hyper-priors on the local scales, $p(\lambda_i^{(l)};\theta_\lambda)$ and $p(\tau_i^{(l)};\theta_\tau)$, control the PVE jointly[1] (Figure 3b). However, Figure 3b also shows how sparsity is determined by $p(\tau_i^{(l)};\theta_\tau)$ alone. Consequently, we propose choosing $p(\tau_i^{(l)};\theta_\tau)$ based on the prior knowledge on sparsity, and after that tuning the $p(\lambda_i^{(l)};\theta_\lambda)$ to achieve the desired level of the PVE, such that in the end our joint prior incorporates both types of prior knowledge.

2.3 Stein Gradient Estimator

Ultimately we want to match the distribution of the PVE for a BNN prior with our prior belief by minimizing the Kullback-Leibler divergence between these two distributions. However, the distribution of a BNN's PVE is analytically intractable, similarly to most functional BNN priors. Thus the gradient of the KL-divergence is also intractable, which makes common gradient based optimization inapplicable. Fortunately, Stein Gradient Estimator (SGE) (Li and Turner, 2018) provides an approximation of the gradient of the log density (i.e., $\nabla_{\mathbf{z}} \log q(\mathbf{z})$), which only requires samples from $q(\mathbf{z})$ instead of its analytical form. Central for the derivation of the SGE is the Stein's identity (Liu et al., 2016):

Theorem 1 (Stein's identity)**.** *Assume that $q(\mathbf{z})$ is a continuous differentiable probability density supported on $\mathcal{Z} \subset \mathbb{R}^d$, $\mathbf{h} : \mathcal{Z} \to \mathbb{R}^{d'}$ is a smooth vector-valued function $\mathbf{h}(\mathbf{z}) = [h_1(\mathbf{z}), \ldots, h_{d'}(\mathbf{z})]^T$, and $\mathbf{h}$ is in the Stein class of q such that*

$$\lim_{\mathbf{z}\to\infty} q(\mathbf{z})\mathbf{h}(\mathbf{z}) = 0 \quad \textit{if } \mathcal{Z} = \mathbb{R}^d. \tag{10}$$

Then the following identity holds:

$$\mathbb{E}_q[\mathbf{h}(\mathbf{z})\nabla_{\mathbf{z}} \log q(\mathbf{z})^T + \nabla_{\mathbf{z}}\mathbf{h}(\mathbf{z})] = 0. \tag{11}$$

SGE estimates $\nabla_{\mathbf{z}} \log q(\mathbf{z})$ by inverting (11) and approximating the expectation with K Monte Carlo samples $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)}\}$ from $q(\mathbf{z})$, such that $-\mathbf{HG} \approx K\overline{\nabla_{\mathbf{z}}\mathbf{h}}$, where $\mathbf{H} = (\mathbf{h}(\mathbf{z}^{(1)}), \ldots, \mathbf{h}(\mathbf{z}^{(K)})) \in \mathbb{R}^{d'\times K}$, $\overline{\nabla_{\mathbf{z}}\mathbf{h}} = \frac{1}{K}\sum_{k=1}^{K} \nabla_{\mathbf{z}^{(k)}}\mathbf{h}(\mathbf{z}^{(k)}) \in \mathbb{R}^{d'\times d}$, and the matrix $\mathbf{G} = (\nabla_{\mathbf{z}^{(1)}} \log q(\mathbf{z}^{(1)}), \ldots, \nabla_{\mathbf{z}^{(K)}} \log q(\mathbf{z}^{(K)}))^T \in \mathbb{R}^{K\times d}$ contains the gradients of $\nabla_{\mathbf{z}} \log q(\mathbf{z})$ for the K samples. Thus a ridge regression estimator is designed to estimate G by adding an l_2 regularizer:

$$\hat{\mathbf{G}}^{\text{Stein}} = \underset{\mathbf{G}\in\mathbb{R}^{K\times d}}{\arg\min} \|\overline{\nabla_{\mathbf{z}}\mathbf{h}} + \frac{1}{K}\mathbf{HG}\|_F^2 + \frac{\eta}{K^2}\|\mathbf{G}\|_F^2, \tag{12}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix and the penalty $\eta \geq 0$. By solving (12), the SGE is obtained:

$$\hat{\mathbf{G}}^{\text{Stein}} = -K(\mathbf{K} + \eta\mathbf{I})^{-1}\mathbf{H}^T\overline{\nabla_{\mathbf{z}}\mathbf{h}}, \tag{13}$$

[1] The scale $\sigma^{(l)}$ is often estimated using a non-informative prior or cross-validated.

where $\mathbf{K} = \mathbf{H}^T\mathbf{H}$ is the kernel matrix, such that $\mathbf{K}_{ij} = \mathcal{K}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \mathbf{h}(\mathbf{z}^{(i)})^T\mathbf{h}(\mathbf{z}^{(j)})$, and $(\mathbf{H}^T\overline{\nabla_{\mathbf{z}}\mathbf{h}})_{ij} = \sum_{k=1}^{K} \nabla_{z_j^{(k)}}\mathcal{K}(\mathbf{z}^{(i)}, \mathbf{z}^{(k)})$, where $\mathcal{K}(\cdot,\cdot)$ is the kernel function. It has been shown that the default RBF kernel satisfies Stein's identity, and we adopt it in this work. In Section 4, we will use SGE to learn the hyper-parameters of the GSM prior for BNN weights such that the resulting distribution of the PVE matches our prior knowledge about the strength of the signal.

3 Prior knowledge about sparsity

In this section, we propose a new hyper-prior for the local scales $p(\tau_i^{(l)}; \theta_\tau)$ to model prior beliefs about sparsity. The new prior generates the local scales conditionally on the number of relevant features, which allows us to explicitly express prior knowledge about the number of relevant features. We focus on the case where each local scale $\tau_i^{(l)}$ is assumed to be binary with domain $\{0, 1\}$, analogously to the feature inclusion indicators in the spike-and-slab prior.

3.1 Prior on the number of relevant features

We control sparsity by placing a prior on the number of relevant features m using a probability mass function $p_m(m; \theta_m)$, where $0 \leq m \leq D$ (dimension of the dataset). Intuitively, if p_m concentrates close to 0, a sparse model with few features is preferred; if p_m places much probability mass close to D, then all of the features are likely to be used instead. Hence, unlike other priors encouraging shrinkage, such as the horseshoe, our new prior easily incorporates experts' knowledge about the number of relevant features. In practice, $p_m(m; \theta_m)$ is chosen based on the available prior knowledge. When there is a good idea about the number of relevant features, a unimodal distribution, such as a discretized Laplace, can be used:

$$p_m(m; \mu, s_m) = c_n \exp\left\{-\frac{s_m|m - \mu_m|}{2}\right\}, \tag{14}$$

where μ_m is the mode, s_m is the precision, and c_n is the normalization constant. Often only an interval for the number of relevant features is available. Then it is possible to use, for example, a 'flattened' Laplace (Figure 1):

$$\begin{aligned} p_m(m; \mu_-, \mu_+, s_m) &= c_n \exp\left\{-\frac{s_m\mathcal{R}(m; \mu_+, \mu_-)}{2}\right\}, \\ \mathcal{R}(m; \mu_-, \mu_+) &= \max\left\{(m - \mu_+), (\mu_- - m), 0\right\}, \end{aligned} \tag{15}$$

where $[\mu_-, \mu_+]$ defines the interval where the probability is uniform and reaches its maximum value, and c_n is the corresponding normalization constant. Equation 14 and 15 include the (discretized) exponential distribution as a special case with $\mu_m = 0$ and $\mu_- = \mu_+ = 0$ respectively; it has been widely studied in sparse deep learning literature (Polson and Ročková, 2018; Wang and Ročková, 2020). The 'flattened' Laplace, with a high precision s_m, is a continuous approximation of the distribution with a uniform

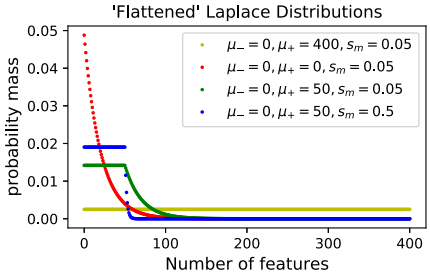

Figure 4: A visualization of 'flattened' Laplace prior with different hyper-parameters. 'Discretized' exponential (red) and uniform (yellow) distributions are special cases of the 'flattened' Laplace.

probability mass within $[\mu_-, \mu_+]$ and 0 outside of $[\mu_-, \mu_+]$ (blue in Figure 4). If there is no prior information about sparsity, a discrete uniform prior over $[0, D]$ is a plausible alternative. See Figure 4 for a visualization.

3.2 Feature allocation

Conditionally on the number of features m, we introduce indicators $I_i \in \{0, 1\}$ to denote if a feature i is used by the model ($I_i = 1$) or not ($I_i = 0$), such that $m = \sum_{i=1}^{D} I_i$. We then marginalize over m using $p_m(m; \theta_m)$. We assume there is no prior knowledge about relative importance of features (this assumption can be easily relaxed if needed), i.e., $\{I_i\}_{i=1}^{D}$ has a jointly uniform distribution given m:

$$p(\{I_i\}_{i=1}^{D} | m) = c_d \delta[m - \sum_{i=1}^{D} I_i], \tag{16}$$

where the normalization constant is $c_d = \binom{D}{m}^{-1}$. Now we can calculate the joint distribution of $\{I_i\}_{i=1}^{D}$ by marginalizing out m:

$$p(\{I_i\}_{i=1}^{D}; \theta_m) = \sum_{m=0}^{D} p_m(m; \theta_m) p(\{I_i\}_{i=1}^{D} | m) = p_m(\sum_{i=1}^{D} I_i; \theta_m) \binom{D}{\sum_{i=1}^{D} I_i}^{-1}. \tag{17}$$

When the local scale variables $\tau_i^{(l)}$ are binary, the $\tau_i^{(l)}$ take the role of the identity variables I_i. Thus we obtain a joint distribution over discrete scale parameters τ_i as

$$p(\tau_1^{(l)}, \ldots, \tau_D^{(l)}; \theta_\tau) = p_m(\sum_{i=1}^{D} \tau_i^{(l)}; \theta_m) \binom{D}{\sum_{i=1}^{D} \tau_i^{(l)}}^{-1}, \tag{18}$$

where θ_τ represents the same set of hyper-parameters as θ_m, and the distribution $p_m(\sum_{i=1}^{D} \tau_i^{(l)}; \theta_m)$ models the beliefs of the number of relevant features. In general,

the local scales $\{\tau_i^{(l)}\}_{i=1}^D$ in (18) are dependent. However, when $p_m(\cdot)$ is set to a Binomial (or its Gaussian approximation), the joint distribution of $\{\tau_i^{(l)}\}_{i=1}^D$ factorizes into a product of independent Bernoullis corresponding to the vanilla spike-and-slab with a fixed slab probability (8). We refer to (18) as the *informative spike-and-slab* prior.

In BNNs, we suggest to use the informative spike-and-slab prior on the first layer to inform the model about sparsity on the feature level. For hidden layers, we do not assume prior domain knowledge, as they encode latent representations where such knowledge is rare. However, an exponential prior on the number of hidden nodes could be applied on each hidden layers to infer optimal layer sizes and perform Bayesian compression (Kingma et al., 2015; Molchanov et al., 2017; Louizos et al., 2017), and it can achieve near minimax rate of posterior concentration in sparse deep learning (Polson and Ročková, 2018). It also may improve the current subnetwork inference (Daxberger et al., 2020), that uses the simplest Gaussian priors without any sparsity, by improving subnetwork selection with explicit sparsity inducing priors. We leave these ideas as promising topics for the future. Instead, in this work, we use the standard Gaussian scale mixture priors in (5) for the hidden layers.

4 Prior knowledge on the PVE

After incorporating prior knowledge about sparsity in the new informative hyper-prior $p(\tau_i^{(l)}; \theta_\tau)$, in this section we introduce an optimization approach to determine the hyper-parameters (i.e., θ_λ) of the hyper-prior for the other local scale parameters $p(\lambda_i^{(l)}; \theta_\lambda)$ in the GSM prior (9), based on domain knowledge about the PVE.

4.1 PVE for Bayesian neural networks

According to the definition of the PVE in (3), and assuming the noise ϵ has a zero mean, the PVE for a regression model in (2) can be written as

$$\text{PVE}(\mathbf{w}) = 1 - \frac{\text{Var}(\epsilon)}{\text{Var}(f(\mathbf{X}; \mathbf{w})) + \text{Var}(\epsilon)} = \frac{\text{Var}(f(\mathbf{X}; \mathbf{w}))}{\text{Var}(f(\mathbf{X}; \mathbf{w})) + \text{Var}(\epsilon)}. \tag{19}$$

When $f(\mathbf{x}; \mathbf{w})$ is a BNN, $\text{PVE}(\mathbf{w})$ has a distribution induced by the distribution on $\mathbf{w}$. We denote the variance of the unexplainable noise ϵ by σ_ϵ^2. We use $\mathbf{w}_{(\boldsymbol{\sigma}, \theta_\lambda, \theta_\tau)}$ to denote the BNN weights with a GSM prior (i.e., (9)) parametrized by hyper-parameters $\{\boldsymbol{\sigma}, \theta_\lambda, \theta_\tau\}$, where $\boldsymbol{\sigma}$ is $\{\sigma^{(0)}, \ldots, \sigma^{(L)}\}$. The PVE of a BNN with a GSM prior can be written as

$$\text{PVE}(\mathbf{w}_{(\boldsymbol{\sigma}, \theta_\lambda, \theta_\tau)}, \sigma_\epsilon) = \frac{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(\boldsymbol{\sigma}, \theta_\lambda, \theta_\tau)}))}{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(\boldsymbol{\sigma}, \theta_\lambda, \theta_\tau)})) + \sigma_\epsilon^2}. \tag{20}$$

The noise σ_ϵ and layer-wise global scales $\boldsymbol{\sigma}$ are usually given the same non-informative priors (Zhang and Bondell, 2018) or set to a default value (Blundell et al., 2015). The hyper-parameter θ_τ is specified as described in Section 3. The remaining hyper-parameter θ_λ we optimize to make the distribution of the PVE match our prior knowledge about the PVE.

4.2 Optimizing hyper-parameters according to prior PVE

Denote the available prior knowledge about the PVE by $p(\rho)$. In practice such a prior may be available from previous studies, and here we assume it can be encoded into the reasonably flexible Beta distribution. If no prior information is available, a uniform prior, i.e., Beta$(1,1)$, can be used. We incorporate such knowledge into the prior by optimizing the hyper-parameter θ_λ such that the distribution induced by the BNN weight prior, $p(\mathbf{w};\boldsymbol{\sigma},\theta_\lambda,\theta_\tau)$, on the BNN model's PVE denoted by $q_{\theta_\lambda}(\rho(\mathbf{w}))$,[2] is close to $p(\rho)$. We achieve this by minimizing the Kullback–Leibler divergence from $q_{\theta_\lambda}(\rho(\mathbf{w}))$ to $p(\rho)$ w.r.t. the hyper-parameter θ_λ, i.e., $\theta_\lambda^* = \arg\min_{\theta_\lambda} \mathrm{KL}[q_{\theta_\lambda}(\rho(\mathbf{w}))|p(\rho)]$.

However, the KL divergence is not analytically tractable because the $q_{\theta_\lambda}(\rho(\mathbf{w}))$ is defined implicitly, such that we can only sample from $q_{\theta_\lambda}(\rho(\mathbf{w}))$ but can not evaluate its density. We first observe that the KL divergence can be approximated by:

$$\begin{aligned}\mathrm{KL}[q_{\theta_\lambda}(\rho(\mathbf{w}))|p(\rho)] &= \mathrm{E}_{p(\mathbf{w};\theta_\lambda)}\left[\log\frac{q_{\theta_\lambda}(\rho(\mathbf{w}))}{p(\rho(\mathbf{w}))}\right] = \mathrm{E}_{p(\xi)}\left[\log\frac{q(\rho(g(\xi;\theta_\lambda)))}{p(\rho(g(\xi;\theta_\lambda)))}\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M}\log q_{\theta_\lambda}(\rho(g(\xi^{(m)};\theta_\lambda))) - \log p(\rho(g(\xi^{(m)};\theta_\lambda))),\end{aligned} \tag{21}$$

by reparametrization and Monte Carlo integration. Here we assume that parameters, which are optimized to match the $p(\rho)$, of the GSM distribution $p(\mathbf{w};\theta_\lambda)$ can be reparametrized by a deterministic function $\mathbf{w} = g(\xi;\theta_\lambda)$ with $\xi \sim p(\xi)$. This includes common distributions over scales, such as the half-Cauchy or the inverse gamma. For non-reparametrizable distributions, such as the logistic distribution (Stefanski, 1991; Izmailov et al., 2021), score function estimators can be used instead, but we leave this for future work. Moreover, since $\mathrm{PVE}(\mathbf{w})$ is another deterministic function of $\mathbf{w}$ given data $\mathbf{X}$, we have $\mathrm{PVE}(\mathbf{w}) = \rho(\mathbf{w}) = \rho(g(\xi;\theta_\lambda))$. The expectation is approximated by M samples from $p(\xi)$. Then the gradient of the KL w.r.t. θ_λ can be calculated by:

$$\begin{aligned}\nabla_{\theta_\lambda}\mathrm{KL}\left[q_{\theta_\lambda}(\rho(\mathbf{w}))|p(\rho)\right] &\approx \frac{1}{M}\sum_{m=1}^{M}\nabla_{\theta_\lambda}\left[\log q_{\theta_\lambda}(\rho(g(\xi^{(m)};\theta_\lambda))) - \log p(\rho(g(\xi^{(m)};\theta_\lambda)))\right] \\ &= \frac{1}{M}\sum_{m=1}^{M}\nabla_{\theta_\lambda}\rho(g(\xi^{(m)};\theta_\lambda))\left[\nabla_\rho \log q_{\theta_\lambda}(\rho) - \nabla_\rho \log p(\rho)\right].\end{aligned} \tag{22}$$

The first term $\nabla_{\theta_\lambda}\rho(g(\xi^{(m)};\theta_\lambda))$ can be calculated exactly with back-propagation packages, such as PyTorch. The last term, the gradient of the log density $\nabla_\rho \log p(\rho)$, is tractable for a prior with a tractable density, such as the Beta distribution. However, the derivative $\nabla_\rho \log q_{\theta_\lambda}(\rho)$ is generally intractable, as the distribution of the PVE of a BNN $q_{\theta_\lambda}(\rho)$ is implicitly defined by (20). We propose to apply SGE (Section 2.3) to estimate $\nabla_\rho \log q_{\theta_\lambda}(\rho)$, which only requires samples from $q_{\theta_\lambda}(\rho)$.

[2] Hyper-parameters σ and θ_τ are omitted for simplicity as they are not optimized.

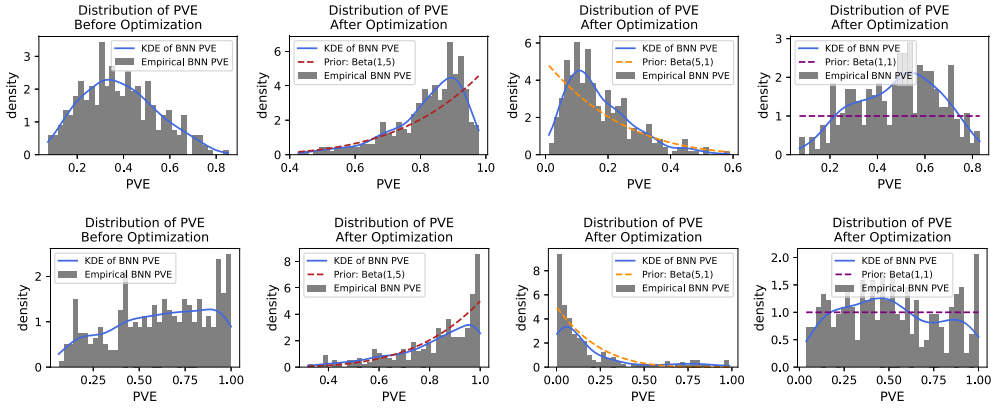

Figure 5: Empirical distributions (50 samples) of the BNN's PVE before (first column) and after optimizing hyper-parameters according to three different prior PVEs: Beta(1,5), Beta(5,1) and Beta(1,1) (last three columns). Top: Results for mean-field Gaussian prior, where the local scale is optimized. Bottom: Results for hierarchical Gaussian prior, where the hyper-parameter of the inverse gamma prior is optimized.

When noise σ_ϵ and layer-wise global scales $\boldsymbol{\sigma}$ are given non-informative priors, e.g., Inv-Gamma$(0.001, 0.001)$, drawing samples directly according to (20) is unstable, because the variance of the non-informative prior does not exist. Fortunately, if all activation functions of the BNN are positively homogeneous (e.g., ReLU), we have the following theorem (proof is given in Supplementary (Cui et al., 2021)):

Theorem 2. *Assume that a BNN has L layers with the Gaussian scale mixture prior in the form of (9) and all activation functions positively homogeneous, e.g., ReLU. Then:*

$$Var(f(\mathbf{X}; \mathbf{w}_{(\boldsymbol{\sigma},\theta_\lambda,\theta_\tau)})) = Var(f(\mathbf{X}; \mathbf{w}_{(1,\theta_\lambda,\theta_\tau)})) \prod_{l=0}^{L} \sigma^{(l)2}. \tag{23}$$

Now instead of giving non-informative priors to all layer-wise global scales, i.e., $\sigma_\epsilon = \sigma^{(0)} = \ldots = \sigma^{(L)} \sim$ Inv-Gamma$(0.001, 0.001)$, we propose to use $\sigma^{(0)} = \ldots = \sigma^{(L-1)} = 1$, and $\sigma_\epsilon = \sigma^{(L)} \sim$ Inv-Gamma$(0.001, 0.001)$. By substituting (23) with these specifications into (20), we can write the PVE as

$$\text{PVE}(\mathbf{w}_{(\sigma,\theta_\lambda,\theta_\tau)}, \sigma_\epsilon) = \frac{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(1,\theta_\lambda,\theta_\tau)}))}{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(1,\theta_\lambda,\theta_\tau)})) + \frac{\sigma_\epsilon^2}{\sigma^{(L)2}}} = \frac{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(1,\theta_\lambda,\theta_\tau)}))}{\text{Var}(f(\mathbf{X}; \mathbf{w}_{(1,\theta_\lambda,\theta_\tau)})) + 1}, \tag{24}$$

where the non-informative inverse Gamma distribution has canceled out, avoiding sampling from it.

In practice, we find that using the whole training set, its subset, or even a simulated dataset with a distribution similar to the test set to compute the PVE yields similar results. Moreover, we observe that the learned hyper-parameters are not sensitive to

the number of samples of SGE. This is because the distribution of the BNN's PVE is relatively simple and usually unimodal (e.g., Figure 5), and it depends only on a small number of trainable hyper-parameters (see Supplementary for more details). Figure 5 illustrates the proposed approach, where we applied the method on two 3-layer BNNs containing 100-50-30-1 nodes and the ReLU activation. We considered two GSM ARD priors for the BNN weights: a mean-field Gaussian prior

$$w_{ij}^{(l)}|\lambda_i^{(l)} \sim \mathcal{N}(0, \sigma^{(l)2}\lambda_i^{(l)2}); \quad \lambda_i^{(l)} = \sigma_\lambda, \tag{25}$$

and a hierarchical Gaussian prior

$$w_{ij}^{(l)}|\lambda_i^{(l)} \sim \mathcal{N}(0, \sigma^{(l)2}\lambda_i^{(l)2}); \quad \lambda_i^{(l)} \sim \text{Inv-Gamma}(\alpha, \beta), \tag{26}$$

with the non-informative prior on the noise and the last layer-wise global scale. For the Gaussian prior, it can be reparametrized by

$$w_{ij}^{(l)} = \sigma_\lambda \sigma^{(l)} \xi_{ij}^{(l)}; \quad \xi_{ij}^{(l)} \sim \mathcal{N}(0, 1), \tag{27}$$

and we optimized σ_λ according to the prior PVE. For the hierarchical Gaussian prior, we optimized β while fixing $\alpha = 2$ because the shape parameter of the Gamma distribution is non-reparametrizable, and we use the following reparametrization

$$w_{ij}^{(l)} = \beta \sigma^{(l)} \xi_{ij}^{(l)} \eta_i^{(l)}; \quad \xi_{ij}^{(l)} \sim \mathcal{N}(0, 1); \quad \eta_i^{(l)} \sim \text{Inv-Gamma}(2, 1). \tag{28}$$

From Figure 5, we see that after optimizing the hyperpriors, the simulated empirical distributions of the PVE are close to the corresponding prior knowledge in both cases, especially when the prior knowledge is informative, such as the Beta(1, 5) or Beta(5, 1). Moreover, we observe that even when we have fixed the shape parameter α of the inverse Gamma, the prior is still flexible enough to approximate the prior PVE well. We provide the whole algorithm of learning hyper-parameters in the Supplementary.

5 Learning BNNs with variational inference

We use variational inference to approximate the posterior distribution with the new informative priors. Sampling algorithms, such as MCMC, are known to be computationally expensive for spike-and-slab priors especially with high-dimensional datasets. According to (4), the ELBO of VI can be rewritten as,

$$\mathcal{L}(\phi) = \mathrm{E}_{q_\phi(\mathbf{w}, \sigma_\epsilon)}[\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma_\epsilon)] - \mathrm{KL}[q_\phi(\mathbf{w}, \sigma_\epsilon)|p(\mathbf{w}, \sigma_\epsilon)], \tag{29}$$

where the first term is the expected log-likelihood and the second term is the Kullback–Leibler divergence from the approximate posterior to the prior. For regression problems studied in this work, we use the Gaussian likelihood,

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma_\epsilon) = \mathcal{N}(\mathbf{y}; f(\mathbf{X}; \mathbf{w}), \sigma_\epsilon^2), \tag{30}$$

Random variables in BNNs	Prior	Variational posterior
$\beta_{ij}^{(l)}$	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\sigma)$
σ_ϵ, $\sigma^{(l)}$, $\lambda_i^{(l)}$	Inv-Gamma(α,β)	Log-Normal(μ,σ)
$\tau_i^{(l)}$	Bernoulli(p), `Info`(θ_τ)	`Con-Bern`(p)

Table 1: Variational posteriors for GSM prior. `Info` represents the informative spike-and-slab prior defined in Equation 18, and `Con-Bern` is the concrete Bernoulli distribution originally proposed by Maddison et al. (2016).

by assuming the unexplainable noise ϵ belongs to $\mathcal{N}(0,\sigma_\epsilon^2)$. For priors in (7), we use a fully factorized variational distribution family to approximate posteriors such that

$$\begin{aligned} q(\mathbf{w},\sigma_\epsilon;\phi) &= q(\sigma_\epsilon;\phi_{\sigma_\epsilon})\prod_{i,j,l} q(w_{ij}^{(l)};\phi_{w_{ij}^{(l)}}) \\ =&q(\sigma_\epsilon;\phi_{\sigma_\epsilon})\prod_{i,j,l} q(\beta_{ij}^{(l)}|\phi_{\beta_{ij}^{(l)}})\prod_{i,l} q(\lambda_i^{(l)}|\phi_{\lambda_i^{(l)}})q(\tau_i^{(l)}|\phi_{\tau_i^{(l)}})\prod_{l} q(\sigma^{(l)}|\phi_{\sigma^{(l)}}), \end{aligned} \tag{31}$$

according to the non-centered parametrization in (9). The explicit form of each variational posterior depends on the corresponding prior, as shown in Table 1.

Now we have defined everything required for calculating the ELBO in (29). To optimize the ELBO, we apply the reparametrization trick (Kingma and Welling, 2013) to make the sampling process of the expected log-likelihood differentiable, and we compute the KL divergence analytically (except for the informative prior, `Info`, where reparametrization trick is used). The variational parameters are updated with Adam (Kingma and Ba, 2014). We give the full algorithm in the Supplementary.

6 Related literature

Priors on the number of relevant features have been applied on small datasets to induce sparsity in shallow models, e.g. NNs with one hidden layer (Vehtari, 2001), including the geometric (Insua and Müller, 1998) and truncated Poisson (Denison et al., 1998; Insua and Müller, 1998; Andrieu et al., 2000; Kohn et al., 2001) distributions. However, those approaches rely on the reversible jump Markov chain Monte Carlo (RJMCMC) to approximate the posterior (Phillips and Smith, 1996; Sykacek, 2000; Andrieu et al., 2013), which does not scale up to deep architectures and large datasets. In this work, we incorporate such prior beliefs into the hyper-prior on the local scales of the Gaussian scale mixture prior; thus, the posterior can be approximated effectively by modern stochastic variational inference (Hoffman et al., 2013), even for deep NN architectures and large datasets.

Priors on PVE have been proposed to fine-tune hyper-parameters (Zhang and Bondell, 2018) and to construct shrinkage priors (Zhang et al., 2020) for Bayesian linear regression, where the distribution on the model PVE is analytically tractable. Simulation and grid search has been used to incorporate prior knowledge about a point

estimate for the PVE in Bayesian reduced rank regression (Marttinen et al., 2014), and the prediction variance in BNN with a Gaussian prior (Wilson and Izmailov, 2020). In the Supplementary, we develop a novel Monte Carlo approach to model the log-linear relationship between the global scale of the Mean-Field Gaussian prior and the prediction variance of the BNN to avoid computationally expensive grid search, and we use this to set the variance according to a point estimate of the PVE, but we find that the resulting nonhierarchical Gaussian prior is not flexible enough.

Priors on the BNNs weights BNNs with a fully factorized Gaussian prior were proposed by Graves (2011) and Blundell et al. (2015). They can be interpreted as NNs with dropout by using a mixture of Dirac-delta distributions to approximate the posterior (Gal and Ghahramani, 2016). Nalisnick et al. (2019) extended these works and showed that NNs with any multiplicative noise could be interpreted as BNNs with GSM ARD priors. Priors over weights with low-rank assumptions, such as the k-tied normal (Swiatkowski et al., 2020) and rank-1 perturbation (Dusenberry et al., 2020) were found to have better convergence rates and ability to capture multiple modes when combined with ensemble methods. Matrix-variate Gaussian priors were proposed by Neklyudov et al. (2017) and Sun et al. (2017) to improve the expressiveness of the prior by accounting for the correlations between the weights. Some priors have been proposed to induce sparsity, such as the log-uniform (Molchanov et al., 2017; Louizos et al., 2017), log-normal (Neklyudov et al., 2017), horseshoe (Louizos et al., 2017; Ghosh et al., 2018), and spike-and-slab priors (Deng et al., 2019). Fortuin (2021) provided a comprehensive review about priors in Bayesian deep learning. However, none of the works can explicitly encode domain knowledge into the prior on the NN weights.

Informative priors of BNNs Building of informative priors for NNs has been studied in the function space. One common type of prior information concerns the behavior of the output with certain inputs. Noise contrastive priors (NCPs) (Hafner et al., 2018) were designed to encourage reliable high uncertainty for OOD (out-of distribution) data points. Gaussian processes were proposed as a way of defining functional priors because of their ability to encode rich functional structures. Flam-Shepherd et al. (2017) transformed a functional GP prior into a weight-space BNN prior, with which variational inference was performed. Functional BNNs (Sun et al., 2019) perform variational inference directly in the functional space, where meaningful functional GP priors can be specified. Pearce et al. (2019) used a combination of different BNN architectures to encode prior knowledge about the function. Although functional priors avoid working with uninterpretable high-dimensional weights, encoding sparsity of features into the functional space is non-trivial.

7 Experiments

In this section, we first compare the proposed informative sparse prior with alternatives in a feature selection task on synthetic toy data sets. We then apply it on seven public

Name	$p(\lambda_i)$	$p(\tau_i)$[4]	Hyper-prior
`BetaSS`	vague Inv-Gamma	Bernoulli(p)	$p \sim \text{Beta}(2, 38)$
`DeltaSS`	vague Inv-Gamma	Bernoulli(p)	$p = 0.05$
`InfoSS`	vague Inv-Gamma	FL$(0, 30, 5)$	NA
`HS`	$C^+(0, 1)$	$\tau \sim C^+(0, (\frac{p_0}{n-p_0})^2)$	NA

Table 2: Details of the four Gaussian scale mixture priors included in the comparison on the synthetic datasets. The vague Inv-Gamma represents a diffuse inverse gamma prior Inv-Gamma(0.001,0.001). FL denotes the 'flattened' Laplace prior defined in the main text. NA means that the model is defined without the corresponding hyperprior.

UCI real-world datasets,[3] where we tune the level of noise and the fraction of informative features. Next, we incorporate prior knowledge about PVE into a Bayesian Wavenet prior to evaluate its effectiveness in large-scale time series prediction tasks. Finally, in a genetics application we show that incorporating domain knowledge on both sparsity and the PVE can improve results in a realistic scenario.

7.1 Synthetic data

Setup We first validate the performance of the informative spike-and-slab prior proposed in Section 3 on a feature selection task, using synthetic datasets similar to those discussed by Van Der Pas et al. (2014) and Piironen and Vehtari (2017b). Instead of a BNN, we here use linear regression, i.e., a NN without hidden layers, which enables comparing the proposed strategy of encouraging sparsity with existing alternatives.

Consider n datapoints generated by:

$$y_i = w_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2), \quad i = 1, \ldots, n, \tag{32}$$

where each observation y_i is obtained by adding Gaussian noise with variance σ_ϵ^2 to the signal w_i. We set the number of datapoints n to 400, the first $p_0 = 20$ signals $\{w_i | i = 1, \ldots, 20\}$ equal to A (signal levels), and the rest of the signals to 0. We consider 3 different noise levels $\sigma_\epsilon \in \{1, 1.5, 2\}$, and 10 different signal levels $A \in \{1, 2, \ldots, 10\}$. For each parameter setting (30 in all), we generate 100 data realizations. The model in (32) can be considered a linear regression: $\mathbf{y} = X\hat{\mathbf{w}}^T + \boldsymbol{\epsilon}$, where $X = I$ and $\hat{\mathbf{w}} = (A, \ldots, A, 0, 0, \ldots, 0)$ with the first p_0 elements being A, so this is a feature selection task where the number of features and datapoints are equal. We use the mean squared error (MSE) between the posterior mean signal $\bar{\mathbf{w}}$ and the true signal $\hat{\mathbf{w}}$ to measure the performance.

Parameter settings For estimation we use linear regression with the correct structure. We consider 4 different Gaussian scale mixture priors on $\mathbf{w}$ that all follow the general form

$$w_i \sim \mathcal{N}(0, \sigma^2 \lambda_i^2 \tau_i^2); \quad \sigma = 1; \quad \lambda_i \sim p(\lambda_i); \quad \tau_i \sim p(\tau_i). \tag{33}$$

[3] https://archive.ics.uci.edu/ml/index.php.

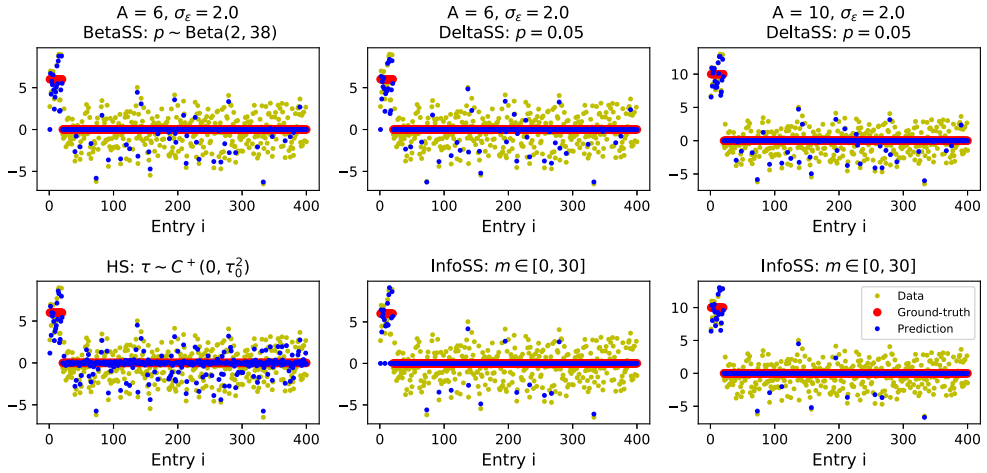

Figure 6: *Synthetic datasets*: A detailed visualization of the features selected by the four priors with noise level $\sigma = 2$ and signal level $A = 6$ (first two columns). Results for the two best models (`DeltaSS` and `InfoSS`) are additionally shown with a larger signal $A = 10$ (the last column). Beta spike-and-slab (`BetaSS`) is slightly worse than Delta spike-and-slab (`DeltaSS`), because the latter uses the correct slab probability. Informative spike-and-slab (`InfoSS`) outperforms alternatives by making the signals dependent. Horseshoe (`HS`) with the correct sparsity level overfits to the noise.

For all the spike-and-slab (`SS`) priors, we place a diffuse inverse Gamma prior on $p(\lambda_i)$. For the `BetaSS` and `DeltaSS` priors, we assume that $p(\tau_i) = \text{Bernoulli}(p)$, and define $p \sim \text{Beta}(2, 38)$ for `BetaSS` and $p = 0.05$ for `DeltaSS`, which both reflect the correct level of sparsity. For the informative spike-and-slab, `InfoSS`, we use the 'flattened' Laplace (FL) prior defined in (15) with $\mu_- = 0, \mu_+ = 30$, and $s_m = 5$, to encode prior knowledge that the number of non-zero signals is (approximately) uniform on $[0, 30]$. We place an informative half-Cauchy prior $C^+(0, \tau_0^2)$ on the global shrinkage scale τ with $\tau_0 = \frac{p_0}{n - p_0}$ in the horseshoe (`HS`) (Piironen and Vehtari, 2017b), to assume the same sparsity level as the other priors.[5] The details of the different priors are shown in Table 2.

Results Figures 6 and 7 show the results on synthetic datasets. Figure 6 provides detailed visualizations of results for the four priors with two representative signal levels $A = 6$ and $A = 10$ and with noise level $\sigma_\epsilon = 2$. Figure 7 summarizes the results for all the signal and noise levels. We first observe that the relationship between MSE and signal level is not monotonic, which is consistent with previous studies (Piironen and Vehtari, 2017b). Intuitively, when true signals are extremely weak, models can shrink all signals to 0 to make the error between estimated and the true signals small, and when the signals are strong, models can identify the true signals easily. We see that `BetaSS` with $p \sim \text{Beta}(2, 38)$ and `DeltaSS` with $p = 0.05$ perform similarly when the

[5] The effective number of features (Piironen and Vehtari, 2017b) is set to 20.

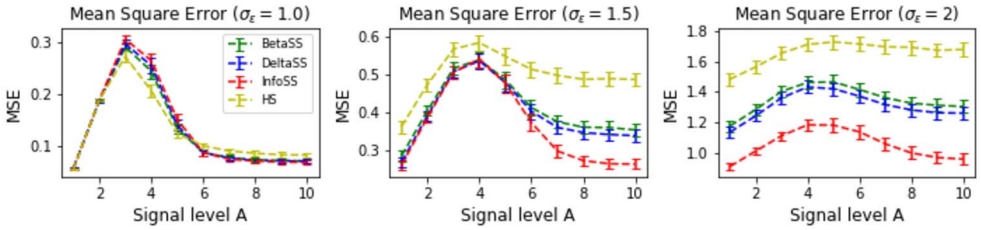

Figure 7: *Synthetic datasets*: Mean squared error (MSE) between the estimated and true signals. The bars represent the 95% confidence intervals over 100 datasets. The novel `InfoSS` prior is indistinguishable from the other SS priors when for the noise level is low. However, `InfoSS` is significantly more accurate than the other priors when there is more noise.

noise level is low, but `DeltaSS` is better than `BetaSS` for higher noise levels. This is because the `DeltaSS` prior is more concentrated close to the true sparsity level; thus, it penalizes false signals more strongly (Top left and bottom panels in Figure 6). The `InfoSS` prior has indistinguishable performance from the other `SS` priors when the noise level is low, but with high noise, e.g., $\sigma_\epsilon = 2.0$, `InfoSS` is significantly better, especially when the signal is large ($A > 6$). This is because `InfoSS` places a prior on the number of features directly, which makes the signals w_i dependent, and consequently including correct signals can help remove incorrect signals. In contrast, the signals are independent of each other in the other priors considered; thus, selecting true signals will not help remove false findings. Another observation is that the Horseshoe prior (`HS`) works well when there is little noise (e.g. $\sigma_\epsilon = 1$), but for larger value of σ_ϵ `HS` is much worse than all the spike-and-slab alternatives because it can easily overfit the noise (bottom-left panel in Figure 6).

7.2 Public real-world UCI datasets

Setup We analyze 7 publicly available datasets:[6] *Bike sharing*, *California housing prices*, *Energy efficiency*, *Concrete compressive strength*, *Yacht hydrodynamics*, *Boston housing*, and *kin8nm dataset*, whose characteristics, the number of individuals N and the number of features D, are given in Table 4. We carry out two types of experiments: Original datasets: we analyze the datasets as such, in which case there is no domain knowledge about sparsity; Extended datasets: we concatenate 100 irrelevant features with the original features and add Gaussian noise to the dependent variable such that the PVE in the data is at most 0.2, which allows us to specify informative priors about sparsity (the number of relevant features is at most the number of original features) and the PVE (less than 0.2). We examine whether the performance can be improved by encoding this extra knowledge into the prior. We use 80% of data for training and 20% for testing. We use the MSE and PVE[7] (i.e., R^2) on a test set to evaluate the

[6] https://archive.ics.uci.edu/ml/index.php.
[7] This is also consistent with the Mean Squared Error (MSE) when the residuals have zero mean.

Name of the prior	$p(\lambda_i^{(l)}),\ \forall l \geq 0$	$p(\tau_i^{(0)})$	$p(\tau_i^{(l)}),\ \forall l \geq 1$
`MF+CV`	$\lambda_i^{(l)} = \sigma_\lambda$	NA	NA
`SS+CV`	$\lambda_i^{(l)} = \sigma_\lambda$	Bernoulli(p)	Bernoulli(p)
`HS`	$C^+(0,1)$	$p(\tau^{(0)}) = C^+(0, 10^{-5})$	$p(\tau^{(l)}) = C^+(0, 10^{-5})$
`HMF`	vague Inv-Gamma	NA	NA
`InfoHMF`	vague Inv-Gamma	FL$(0, D, 1)$	NA
`HMF+PVE`	Inv-Gamma$(2, \beta)$	NA	NA
`InfoHMF+PVE`	Inv-Gamma$(2, \beta)$	FL$(0, D, 1)$	NA

Table 3: Seven Gaussian scale mixture priors included in the comparison on the UCI datasets. Hyper-parameters in `MF+CV` and `SS+CV` (local scale σ_λ and spike probability p) are chosen via 5-fold cross-validation. The hyper-parameter β is optimized to match the prior PVE, and μ_+ is equal to the number of features in the corresponding original dataset without the artificially added irrelevant features.

performance. We repeat each experiment 30 times to obtain confidence intervals, and we give implementation details in the Supplementary. In the Supplementary, we also provide further ablation analyses to separate the effects of two methodological novelties by creating extended datasets with irrelevant features only and with noisy dependent variable only. We also provide sensitivity analyses of each prior to its hyper-parameters on ablation datasets.

Parameter settings We considered 7 different priors: 1. mean-field (independent) Gaussian prior with cross-validation (`MF+CV`) (Blundell et al., 2015); 2. delta spike-and-slab prior with cross-validation (`SS+CV`) (Blundell et al., 2015); 3. horseshoe prior (`HS`) (Ghosh and Doshi-Velez, 2017); 4. Hierarchical Gaussian prior (`HMF`), which is the same as `MF+CV` except that the hyperparameters receive a fully Bayesian treatment instead of cross-validation. 5. The `InfoHMF`, which incorporates domain knowledge on feature sparsity in the `HMF` by applying the proposed informative prior in the input layer; 6. `HMF+PVE` instead includes the informative prior on the PVE, and finally, 7. `InfoHMF+PVE` includes both types of domain knowledge. Lasso regression (Tibshirani, 2011) with cross-validated regularization (`Lasso+CV`) and functional Gaussian processes prior (`GP`) (Sun et al., 2019) with the RBF kernel are included as other standard baselines.

The hyper-parameters for `MF+CV` priors and `SS+CV` prior are chosen by 5-fold cross-validation on the training set from grids $\sigma_\lambda \in \{e^{-2}, e^{-1}, 1, e^1, e^2\}$ and $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, which are wider than used in the original work by Blundell et al. (2015). We define `HS` as suggested by Ghosh et al. (2018), such that the scale $\tau_i^{(l)} = \tau^{(l)}$ is shared by all weights in each layer l. In the `HMF`, we use a non-informative prior on the local scales $\lambda_i^{(l)}$. For `GP`, we use the hyper-parameters in the original implementation. We regard `GP`, `MF+CV`, `SS+CV`, `HS`, and `HMF` as strong benchmarks to compare our novel informative priors against. For `InfoHMF`, we use the 'flattened' Laplace (FL) prior with $\mu_- = 0, \mu_+ = D$ (D is the number of features in the original dataset) on the input layer to encode the prior knowledge about feature sparsity. For `HMF+PVE` and `InfoHMF+PVE`, we optimize the

hyper-parameter β to match the PVE of the BNN with a Beta$(5.0, 1.2)$ for the original datasets (the mode equals 0.95), and with Beta$(1.5, 3.0)$ for the extended datasets (the mode equals 0.20). For all priors that are not informative about the PVE (except the HS), we use an Inv-Gamma(0.001,0.001) for the all layer-wise global scales $\boldsymbol{\sigma}$ and the noise σ_ϵ. For priors informative on the PVE, the non-informative prior is used only for the last layer-wise global scale $\sigma^{(L)}$ and noise σ_ϵ (see Section 4). The details about each prior are summarized in Table 3.

Results The results, in terms of MSE (see test PVE in Supplementary), are reported in Table 4. For the original datasets, we see that incorporating the prior knowledge on the PVE (`HMF+PVE` and `InfoHMF+PVE`) always yields at least as good performance as the corresponding prior without this knowledge (`HMF` and `InfoHMF`). Indeed, `HMF+PVE` has the (shared) highest accuracy in all datasets except Boston. The new proposed informative sparsity inducing prior (`InfoHMF`) does not here improve the performance, as we do not have prior knowledge on sparsity in the original datasets. Among the non-informative priors, `HMF` is slightly better than the rest, except for the Boston housing dataset, where the horseshoe prior (`HS`) achieves the highest test PVE, which demonstrates the benefit of the fully Bayesian treatment vs. cross-validation of the hyperparameters. Inference with the functional `GP` prior is computationally expensive because the complexity of the spectral Stein gradient estimator is cubic to the number of functions. Thus only a small number of functions can be used for large regression tasks such as Bike, which harms the performance. On small datasets, e.g., Concrete, the `GP` prior has competitive performance. The linear method, `Lasso+CV`, is worse than all BNNs in most datasets.

In the extended datasets with the 100 extra irrelevant features **and** noise added to the target, knowledge on both the PVE and sparsity improves performance significantly. For most of the datasets both types of prior knowledge are useful, and consequently `InfoHMF+PVE` is the most accurate on 4 out of 7 datasets. Furthermore, its PVE is also close to 20% of the maximum test PVE in the corresponding original dataset, reflecting the fact that noise was injected to keep only 20% of the signal (see Supplementary). We find that the horseshoe (`HS`) works better than the `HMF` on small datasets, especially Boston, where the `HS` outperform others. The priors `MF+CV` and `SS+CV` do not work well for the extended datasets, and they are even worse than `Lasso+CV`, because cross-validation has a large variance on the noisy datasets especially for flexible models such as BNNs. The more computationally intensive repeated cross-validation (Kuhn and Johnson, 2013) might alleviate the problem, but its further exploration is left for future work. The `GP` priors fail to capture any signal in extended datasets, because they do not induce any sparsity in the feature space, which might be possible to improve with an ARD prior on the kernel. Overall, we conclude that by incorporating knowledge on the PVE and sparsity into the prior the performance can be improved; however, the amount of improvement can be small if the dataset is large (California and Bike) or when the prior knowledge is weak (the original datasets).

[9] The dimension $P = D$ in the original datasets, while $P = 100 + D$ in the extended datasets.

Original (P, N)	California (**9**, 20k)	Bike (**13**, 17k)	Concrete (**8**, 1k)	Energy (**8**, 0.7k)	Kin8nm (**8**, 8.1k)	Yacht (**6**, 0.3k)	Boston (**3**, 0.5k)
`Lasso+CV`	0.378	0.589	0.446	0.112	0.594	0.338	0.449
`GP`	0.314 (0.001)	0.169 (0.000)	**0.124 (0.001)**	0.035 (0.001)	0.167 (0.001)	0.176 (0.003)	0.261 (0.000)
`MF+CV`	0.220 (0.001)	0.067 (0.000)	0.193 (0.003)	0.185 (0.001)	0.087 (0.001)	0.049 (0.001)	0.216 (0.001)
`SS+CV`	0.221 (0.001)	0.074 (0.001)	0.154 (0.002)	0.110 (0.000)	0.095 (0.001)	0.111 (0.005)	0.198 (0.000)
`HS`	0.215 (0.001)	0.073 (0.000)	0.172 (0.004)	0.106 (0.002)	0.097 (0.001)	0.078 (0.004)	**0.190 (0.002)**
`HMF`	0.211 (0.002)	0.067 (0.001)	0.128 (0.003)	0.042 (0.005)	0.072 (0.001)	**0.014 (0.002)**	0.204 (0.002)
`HMF +PVE`	0.208 (0.002)	**0.065 (0.001)**	**0.124 (0.003)**	**0.034 (0.001)**	**0.071 (0.001)**	**0.014 (0.001)**	0.202 (0.003)
`InfoHMF`	0.211 (0.002)	0.066 (0.001)	0.130 (0.003)	0.045 (0.001)	0.072 (0.001)	0.023 (0.001)	0.201 (0.002)
`InfoHMF +PVE`	**0.207 (0.002)**	0.066 (0.001)	0.125 (0.002)	0.041 (0.002)	0.072 (0.001)	0.017 (0.002)	0.198 (0.002)
Extended (P, N)	California (**109**, 20k)	Bike (**113**, 17k)	Concrete (**108**, 1k)	Energy (**108**, 0.7k)	Kin8nm (**108**, 8.1k)	Yacht (**106**, 0.3k)	Boston (**103**, 0.5k)
`Lasso+CV`	0.867	0.913	0.956	0.854	0.899	0.893	0.985
`GP`	1.002 (0.002)	0.998 (0.002)	1.006 (0.008)	1.004 0.009	1.001 (0.003)	1.040 (0.015)	0.983 (0.000)
`MF+CV`	0.947 (0.006)	1.048 (0.011)	1.0652 (0.028)	0.976 (0.039)	1.014 (0.009)	1.049 (0.063)	1.016 (0.049)
`SS+CV`	0.878 (0.006)	0.911 (0.010)	1.0652 (0.029)	0.969 (0.038)	0.926 (0.008)	1.048 (0.062)	1.014 (0.048)
`HS`	0.972 (0.008)	1.017 (0.010)	**0.914 (0.032)**	0.850 (0.035)	0.883 (0.012)	0.888 (0.054)	**0.910 (0.048)**
`HMF`	0.866 (0.006)	0.850 (0.008)	0.940 (0.039)	0.849 (0.018)	0.872 (0.008)	0.989 (0.059)	0.966 (0.047)
`HMF +PVE`	0.864 (0.006)	0.850 (0.008)	0.937 (0.030)	**0.838 (0.018)**	0.865 (0.010)	0.914 (0.072)	0.957 (0.046)
`InfoHMF`	0.864 (0.007)	0.836 (0.006)	0.939 (0.021)	0.862 (0.030)	0.856 (0.010)	0.903 (0.076)	0.961 (0.039)
`InfoHMF +PVE`	**0.861 (0.006)**	**0.827 (0.005)**	0.927 (0.036)	0.841 (0.018)	**0.846 (0.010)**	**0.886 (0.061)**	0.914 (0.035)

Table 4: MSE with 1.96 standard error of the mean (in parentheses) for each prior on UCI datasets. The first seven rows show the experimental results on the original datasets where we have no prior information, and the last seven rows on extended datasets with 100 irrelevant features and injected noise added. The best result in each column has been boldfaced. The dimension (P)[9]and size (N) are shown for each dataset. We see that both information about sparsity and PVE improve the performance, especially when prior information is available (on extended datasets).

7.3 Web traffic time series prediction

Setup In this experiment, we use a web traffic time series dataset[10] to predict future web traffic of Wikipedia articles with historical data. The dataset contains around 145,000 websites with corresponding daily web traffic in 550 days. We randomly selected 50,000 websites in experiments and we use 80% of the selected data for training, 10% for validation, and 10% for testing. We consider four different prediction periods t: 7, 14, 21, and 28 days, i.e., we predict the daily web traffic of each Wikipedia article for the next t days. We evaluate the performance of each model by the MSE and PVE on the prediction periods in the test set. We repeat each experiment 30 times to obtain confidence intervals.

Model settings We use a deep autoregressive neural network architecture which contains eight causal dilation convolutional layers with 32 1D filters of width two and an exponentially increasing dilation rate (similar to the Wavenet architecture (Oord et al., 2016)), followed by two fully connected layers with 128 and 64 hidden nodes. We consider 4 types of priors used in the previous section: 1. mean-field Gaussian prior with cross-validated local scales (`MF+CV`); 2. the horseshoe (`HS`); 3. Hierarchical Gaussian prior with the noninformative Inv-Gamma prior on the local scales (`HMF`); 4. Hierarchical Gaussian prior with trained informative Inv-Gamma prior (`HMF+PVE`) such that the prior PVE is adjusted to Beta(1, 1), i.e., uniform on $[0, 1]$. Each local scale is shared by all parameters of the same filter in convolutional layers. We do not use any informative spike-and-slab prior here because we have no prior knowledge about the number of relevant features in this task. We use teacher forcing during training, and apply walk-forward validation for hyper-parameter fine-tuning.

Results The experimental results, in terms of MSE and PVE on the test set, are shown in Table 5. We observe that when the prediction period is short (e.g., 7 days), `HMF` achieves similar performance as the informative version `HMF+PVE`. However, when the prediction period is long (e.g., 14-28 days), `HMF+PVE` is significantly better than `HMF` and other alternatives. Moreover, `HMF+PVE` has a much lower standard error compared with `HMF`. One explanation is that long-term prediction tasks have a lower signal-to-noise ratio compared with short-term prediction tasks, because prediction errors will accumulate over time. Although the true PVE is unavailable as a prior knowledge, a less informative prior over PVE (e.g., $U[0, 1]$ in `HMF+PVE`) provides sufficient probability density on the true PVE compared with `HMF`, whose induced prior PVE is highly concentrated on 1 and gives almost 0 probability density on the true PVE. A detailed comparison of summary statistics of induced model PVE distribution of `HMF` and `HMF+PVE` prior is provided in Section 6.1 of the Supplementary.

We analyze the performance against the computational cost for each prior in Figure 8. Obviously, `HMF+PVE` takes more training time because of the additional optimization step that learns the prior parameters according to prior PVE. However, the additional computational cost is small compared with the whole training time, and it improves the performance significantly, especially in the noisy long-term prediction.

[10] https://www.kaggle.com/c/web-traffic-time-series-forecasting.

Periods	7 days		14 days		21 days		28 days	
Metrics	MSE	PVE	MSE	PVE	MSE	PVE	MSE	PVE
MF+CV	0.582	0.278	0.615	0.164	0.686	0.118	0.701	0.095
	(0.016)	(0.013)	(0.017)	(0.031)	(0.031)	(0.015)	(0.043)	(0.011)
HS	0.500	0.301	0.556	0.189	0.652	0.120	0.629	0.101
	(0.008)	(0.006)	(0.011)	(0.010)	(0.054)	(0.041)	(0.019)	(0.013)
HMF	**0.481**	**0.322**	0.589	0.179	0.660	0.085	0.664	0.066
	(0.012)	**(0.009)**	(0.022)	(0.023)	(0.047)	(0.072)	(0.043)	(0.051)
HMF+PVE	0.482	0.320	**0.546**	**0.227**	**0.613**	**0.138**	**0.622**	**0.109**
	(0.013)	(0.010)	**(0.014)**	**(0.011)**	**(0.014)**	**(0.013)**	**(0.017)**	**(0.011)**

Table 5: MSE and PVE with 1.96 standard error of the mean (in parentheses) for each prior on time series prediction tasks. The best result in each column has been boldfaced. We see that for short term predictions, HMF is competitive with the informative prior (HMF+PVE). However, for long term prediction tasks which are more noisy, HMF+PVE is significantly better than the alternatives.

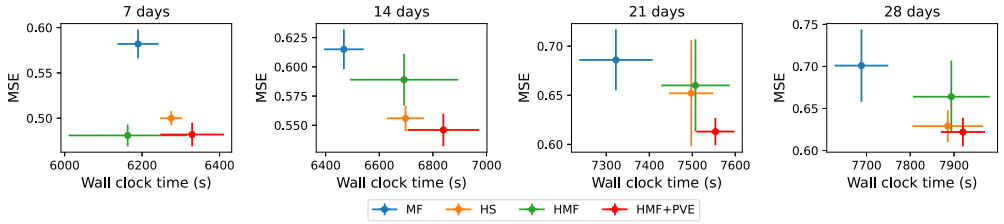

Figure 8: MSE vs. wall clock training time(s) plot with 1.96 standard error. MF has the smallest training time (without cross-validation), but has the highest MSE. HS and HMF are close in training time, because the number of parameters that needed to learn are almost the same. Although HMF+PVE requires an additional optimization step to learn hyper-parameters, it costs little compared with the whole training time, which improves the performance consistently.

7.4 Metabolite prediction using genetic data

Setup Genome-wide association studies (GWASs) aim to learn associations between genetic variants called SNPs (input features) and phenotypes (targets). Ultimately, the goal is to predict a given phenotype given the SNPs of an individual. This task is extremely challenging because 1. the input features are very high-dimensional and strongly correlated and 2. the features may explain only a tiny fraction of the variance of the phenotype, e.g. less 1%. In such a case, neural networks may overfit severely and have worse accuracy than the simple prediction by mean. Typical approaches employ several heuristic but crucial preprocessing steps to reduce the input dimension and correlation. However, strong domain knowledge about sparsity and the amount of variance explained by the SNPs is available, and we show that by incorporating this knowledge in the informative prior we can accurately predict where alternatives fail.

Name of the prior	$p(\lambda_i^{(l)}),\ \forall l \geq 0$	$p(\tau_i^{(0)})$	$p(\tau_i^{(l)}),\ \forall l \geq 1$
`HS`	$C^+(0,1)$	$p(\tau^{(0)}) = C^+(0, 10^{-5})$	$p(\tau^{(l)}) = C^+(0, 10^{-5})$
`MF+CV`	$\lambda_i^{(l)} = \sigma_\lambda$	NA	NA
`MF+PVE`	$\lambda_i^{(l)} = \sigma_\lambda^{\text{prior}}$	NA	NA
`SS+CV`	$\lambda_i^{(l)} = \sigma_\lambda$	Bernoulli(p)	Bernoulli(p)
`SS+PVE`	$\lambda_i^{(l)} = \sigma_\lambda^{\text{prior}}$	Bernoulli(p)	Bernoulli(p)
`InfoMF+CV`	$\lambda_i^{(l)} = \sigma_\lambda$	FL$(0, \mu_+, 1)$	NA
`InfoMF+PVE`	$\lambda_i^{(l)} = \sigma_\lambda^{\text{prior}}$	FL$(0, \mu_+, 1)$	NA
`HMF`	vague Inv-Gamma	NA	NA
`HMF+PVE`	Inv-Gamma$(2, \beta)$	NA	NA
`InfoHMF`	vague Inv-Gamma	FL$(0, \mu_+, 1)$	NA
`InfoHMF+PVE`	Inv-Gamma$(2, \beta)$	FL$(0, \mu_+, 1)$	NA

Table 6: Nine Gaussian scale mixture priors included in the genetics experiment. Hyper-parameters: the local scale σ_λ and the spike probability p are chosen via 5-fold cross-validation on the training set; the informative hyper-parameters $\sigma_\lambda^{\text{prior}}$, β are optimized to match the prior PVE; μ_+ equals 20% number of SNPs in corresponding gene.

We apply the proposed approach on the FINRISK dataset (Borodulin et al., 2018), which contains genetic data and 228 different metabolites as phenotypes for 4,620 individuals. We select six genes that have previously been found to be associated with the metabolites (Kettunen et al., 2016). We use the SNPs in each gene as features to predict the metabolite most strongly associated with the gene, resulting in 6 different experiments. We make predictions using the posterior mean and evaluate the performance by MSE (smaller is better) and PVE (larger is better) on test data. We use 50% of the data for training and 50% for testing, and we repeat this 50 times for each of the six experiments (i.e., for each gene), to account for the variability due to BNN training.

Parameter settings We train BNNs with 1 hidden layer having 100 hidden nodes; the complexity of the data prevents the use of more complex models, but even the single hidden layer increases flexibility and improves accuracy (see results). We consider nine priors: the horseshoe (`HS`); the mean-field Gaussian with local scales $\lambda_i^{(l)}$ set by cross-validation (`MF+CV`) or optimized using the PVE (`MF+PVE`); the delta spike-and-slab, where the slab probability p is cross-validated and the local scales are set either by cross-validation or the PVE (`SS+CV` and `SS+PVE`); the mean-field prior including the informative prior about feature sparsity, and the local scales either cross-validated (`InfoMF+CV`) or set using the PVE (`InfoMF+PVE`); the hierarchical Gaussian prior with noninformative prior over local scales (`HMF`) or with informative prior over local scales using PVE (`HMF+PVE`); and finally, hierarchical Gaussian prior with informative prior about feature sparsity, and noninformative local scales prior (`InfoHMF`) or informative local scales prior with PVE (`InfoHMF+PVE`). We use the 'flattened' Laplace (FL) prior with $\mu_- = 0, \mu_+ = 0.2D$, where D is the number of SNPs in a given gene, to reflect the prior belief that less than 20% of the SNPs in the gene affect the phenotype. To encode

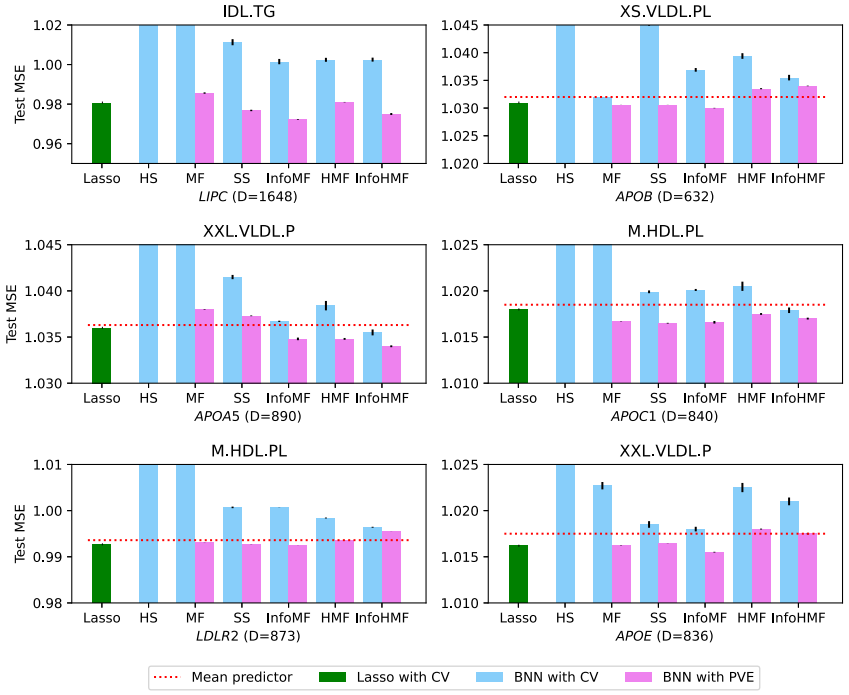

Figure 9: Each panel shows the results for one experiment of predicting a given metabolite (specified in the title, e.g., IDL.TG) using the SNPs in one gene (specified below the panel, e.g., *LIPC*). Each bar shows the average MSE over 50 repeated experiments, and the error bar is the corresponding 95% CI. Blue bars indicate priors not including knowledge on the PVE (with cross-validated hyper-parameters), while purple bars show priors that incorporate the knowledge on the PVE. Green bars show Lasso linear regression with cross-validated regularization. Red dashed lines show the mean predictor. Some results are out of scale for illustration purposes (e.g., the mean predictor of the first panel). In summary, prior knowledge on both the PVE and sparsity improves the performance in most experiments.

the knowledge of the PVE, we optimize the hyper-parameter σ_λ to match the mode of the PVE with previous findings (Kettunen et al., 2016). The priors are summarized in Table 6. We include Lasso regression with cross-validated regularization as a strong linear baseline (Lello et al., 2018).

Results Figure 9 shows results for the 6 experiments (genes). We see that using the prior knowledge on the PVE always improves accuracy (purple bars). Without the prior on the PVE the mean-field Gaussian prior can overfit severely (blue bars), and cannot even outperform the mean predictors (red dashed lines). See comparisons of induced model PVE of each prior in the Supplementary. Furthermore, the novel informative sparse prior performs better than or similarly to the spike-and-slab prior with

the cross-validated slab probability (compare `InfoMF` vs. `SS`). However, it is notable that applying the `SS` prior requires computationally intensive cross-validation to set the slab probability p, which is avoided by the `InfoMF`. The hierarchical priors, i.e., `HS`, `HMF`, and `InfoHMF`, are in general less accurate than the non-hierarchical priors with PVE, although prior knowledge on sparsity and PVE consistently improve their performance. We hypothesize that this genetics dataset is rather simple, and therefore the non-hierarchical priors are flexible enough to capture the signal. We also notice that although BNNs with informative priors are better than the Lasso in most cases, the performance is similar for gene *LDLR2* and gene *APOB*, which indicates that the true effect can be captured by a linear model. Overall, the highest accuracy is achieved by `SS+PVE` or `InfoMF+PVE` priors in most experiments, and `InfoHMF+PVE` in *APOA5*.

8 Conclusion

In this paper, we provided an approach to incorporate two types of domain knowledge, on feature sparsity and the proportion of variance explained (PVE), into the widely used Gaussian scale mixture priors for BNNs. Specifically, we proposed to use a new informative spike-and-slab prior on the input layer to reflect the belief about feature sparsity, and to tune the model's PVE with prior knowledge on the PVE, by optimizing the hyper-parameters of the local scales for all neural network weights. We demonstrated the utility of the approach on simulated data, publicly available datasets, time series prediction data, and in a genetics application, where they outperformed strong commonly used baselines without computationally expensive cross-validation.

The informative spike-and-slab is not limited to the Gaussian scale mixtures, but can be generalized to all scale mixture distributions. One limitation of using the PVE to reflect the signal-to-noise ratio is that it is only defined for priors with finite second moments and regression tasks. Therefore, for some heavy-tailed distributions, such as the horseshoe, and for classification tasks, other measures of signal-to-noise ratio should be developed as part of future work. Moreover, we use variational inference in this work due to its computational feasibility and for a fair comparison with baselines implemented with VI. However, VI is known to underestimate the posterior uncertainty and only approximate one mode of the true BNN posterior (Wilson and Izmailov, 2020). Combining VI with a deep ensemble (Wilson and Izmailov, 2020) or using a stochastic gradient MCMC (Wenzel et al., 2020; Izmailov et al., 2021) may yield better posterior approximations, and exploring this further is another possible future direction.

Supplementary Material

Supplementary material of "Informative Bayesian Neural Network Priors for Weak Signals" contains proofs and implementation details (DOI: 10.1214/21-BA1291SUPP; .pdf).

References

Andrews, D. F. and Mallows, C. L. (1974). "Scale mixtures of normal distributions." *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1): 99–102. MR0359122. 2, 5

Andrieu, C., De Freitas, J. F., and Doucet, A. (2000). "Robust full Bayesian methods for neural networks." In *Advances in Neural Information Processing Systems*, 379–385. 14

Andrieu, C., De Freitas, N., and Doucet, A. (2013). "Reversible jump MCMC simulated annealing for neural networks." *arXiv preprint arXiv:1301.3833*. 14

Betancourt, M. and Girolami, M. (2015). "Hamiltonian Monte Carlo for hierarchical models." *Current Trends in Bayesian Methodology with Applications*, 79: 30. MR3644666. 6

Bhattacharya, A., Pati, D., Pillai, N. S., and Dunson, D. B. (2015). "Dirichlet–Laplace priors for optimal shrinkage." *Journal of the American Statistical Association*, 110(512): 1479–1490. MR3449048. doi: https://doi.org/10.1080/01621459.2014.960967. 6

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. MR2247587. doi: https://doi.org/10.1007/978-0-387-45528-0. 5

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). "Weight uncertainty in neural networks." *arXiv preprint arXiv:1505.05424*. 2, 10, 15, 19

Borodulin, K., Tolonen, H., Jousilahti, P., Jula, A., Juolevi, A., Koskinen, S., Kuulasmaa, K., Laatikainen, T., Männistö, S., Peltonen, M., et al. (2018). "Cohort profile: the National FINRISK study." *International Journal of Epidemiology*, 47(3): 696–696i. 24

Cui, T., Havulinna, A., Marttinen, P., and Kaski, S. (2021). "Supplementary material for: Informative Bayesian Neural Network Priors for Weak Signals." *Bayesian Analysis*. doi: https://doi.org/10.1214/21-BA1291SUPP. 12

Daxberger, E., Nalisnick, E., Allingham, J. U., Antorán, J., and Hernández-Lobato, J. M. (2020). "Expressive yet tractable Bayesian deep learning via subnetwork inference." *arXiv preprint arXiv:2010.14689*. 10

Deng, W., Zhang, X., Liang, F., and Lin, G. (2019). "An adaptive empirical Bayesian method for sparse deep learning." In *Advances in Neural Information Processing Systems*, 5564–5574. 3, 15

Denison, D., Mallick, B., and Smith, A. (1998). "Automatic Bayesian curve fitting." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2): 333–350. MR1616029. doi: https://doi.org/10.1111/1467-9868.00128. 14

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*. 1

Dusenberry, M. W., Jerfel, G., Wen, Y., Ma, Y.-a., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. (2020). "Efficient and scalable Bayesian neural nets with rank-1 factors." *arXiv preprint arXiv:2005.07186*. 15

Flam-Shepherd, D., Requeima, J., and Duvenaud, D. (2017). "Mapping Gaussian process priors to Bayesian neural networks." In *NIPS Bayesian Deep Learning Workshop*. 15

Fortuin, V. (2021). "Priors in Bayesian deep learning: A review." *arXiv preprint arXiv:2105.06868*. 15

Fortuin, V., Garriga-Alonso, A., Wenzel, F., Rätsch, G., Turner, R., van der Wilk, M., and Aitchison, L. (2021). "Bayesian neural network priors revisited." *arXiv preprint arXiv:2102.06571*. 5

Gal, Y. and Ghahramani, Z. (2016). "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." In *International Conference on Machine Learning*, 1050–1059. 15

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC. MR3235677. 1, 5

Ghosh, S. and Doshi-Velez, F. (2017). "Model selection in Bayesian neural networks via horseshoe priors." *arXiv preprint arXiv:1705.10388*. MR4048993. 19

Ghosh, S., Yao, J., and Doshi-Velez, F. (2018). "Structured variational learning of Bayesian neural networks with horseshoe priors." In *International Conference on Machine Learning*, 1739–1748. MR4048993. 6, 15, 19

Glantz, S. A., Slinker, B. K., and Neilands, T. B. (1990). *Primer of Applied Regression and Analysis of Variance*, volume 309. McGraw-Hill New York. 2, 4

Graves, A. (2011). "Practical variational inference for neural networks." In *Advances in Neural Information Processing Systems*, 2348–2356. 15

Hafner, D., Tran, D., Lillicrap, T., Irpan, A., and Davidson, J. (2018). "Noise contrastive priors for functional uncertainty." *arXiv preprint arXiv:1807.09289*. 15

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). "Stochastic variational inference." *The Journal of Machine Learning Research*, 14(1): 1303–1347. MR3081926. 14

Insua, D. R. and Müller, P. (1998). "Feedforward neural networks for nonparametric regression." In *Practical Nonparametric and Semiparametric Bayesian Statistics*, 181–193. Springer. MR1630081. doi: https://doi.org/10.1007/978-1-4612-1732-9_9. 14

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. (2021). "What are Bayesian neural network posteriors really like?" *arXiv preprint arXiv:2104.14421*. 11, 26

Kettunen, J., Demirkan, A., Würtz, P., Draisma, H. H., Haller, T., Rawal, R., Vaarhorst, A., Kangas, A. J., Lyytikäinen, L.-P., Pirinen, M., et al. (2016). "Genome-wide study

for circulating metabolites identifies 62 loci and reveals novel systemic effects of LPA." *Nature Communications*, 7(1): 1–9. 24, 25

Kingma, D. P. and Ba, J. (2014). "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*. 14

Kingma, D. P., Salimans, T., and Welling, M. (2015). "Variational dropout and the local reparameterization trick." *arXiv preprint arXiv:1506.02557*. 10

Kingma, D. P. and Welling, M. (2013). "Auto-encoding variational Bayes." *arXiv preprint arXiv:1312.6114*. 5, 14

Kohn, R., Smith, M., and Chan, D. (2001). "Nonparametric regression using linear combinations of basis functions." *Statistics and Computing*, 11(4): 313–322. MR1863502. doi: https://doi.org/10.1023/A:1011916902934. 14

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks." In *Advances in Neural Information Processing Systems*, 1097–1105. 1

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*, volume 26. Springer. MR3099297. doi: https://doi.org/10.1007/978-1-4614-6849-3. 20

Lello, L., Avery, S. G., Tellier, L., Vazquez, A. I., de Los Campos, G., and Hsu, S. D. (2018). "Accurate genomic prediction of human height." *Genetics*, 210(2): 477–497. 25

Li, Y. and Turner, R. E. (2018). "Gradient estimators for implicit models." In *International Conference on Learning Representations*. 3, 7

Liu, Q., Lee, J., and Jordan, M. (2016). "A kernelized Stein discrepancy for goodness-of-fit tests." In *International Conference on Machine Learning*, 276–284. 7

Louizos, C., Ullrich, K., and Welling, M. (2017). "Bayesian compression for deep learning." In *Advances in Neural Information Processing Systems*, 3288–3298. 6, 10, 15

MacKay, D. J. (1992). "A practical Bayesian framework for backpropagation networks." *Neural Computation*, 4(3): 448–472. MR1278216. doi: https://doi.org/10.1007/BF02430635. 4

MacKay, D. J. (1994). "Bayesian nonlinear modeling for the prediction competition." *ASHRAE Transactions*, 100(2): 1053–1062. 5

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). "The concrete distribution: A continuous relaxation of discrete random variables." *arXiv preprint arXiv:1611.00712*. 14

Marttinen, P., Pirinen, M., Sarin, A.-P., Gillberg, J., Kettunen, J., Surakka, I., Kangas, A. J., Soininen, P., O'Reilly, P., Kaakinen, M., et al. (2014). "Assessing multivariate gene-metabolome associations with rare variants using Bayesian reduced rank regression." *Bioinformatics*, 30(14): 2026–2034. 4, 15

Mitchell, T. J. and Beauchamp, J. J. (1988). "Bayesian variable selection in linear

regression." *Journal of the American Statistical Association*, 83(404): 1023–1032. MR0997578. 2, 5

Molchanov, D., Ashukha, A., and Vetrov, D. (2017). "Variational dropout sparsifies deep neural networks." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2498–2507. JMLR.org. 10, 15

Nalisnick, E., Hernandez-Lobato, J. M., and Smyth, P. (2019). "Dropout as a structured shrinkage prior." In *International Conference on Machine Learning*, 4712–4722. 15

Neal, R. M. (2012). *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media. 2, 4, 5, 7

Neklyudov, K., Molchanov, D., Ashukha, A., and Vetrov, D. P. (2017). "Structured Bayesian pruning via log-normal multiplicative noise." In *Advances in Neural Information Processing Systems*, 6775–6784. 15

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499*. 22

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). "A general framework for the parametrization of hierarchical models." *Statistical Science*, 59–73. MR2408661. doi: https://doi.org/10.1214/088342307000000014. 6

Pearce, T., Zaki, M., Brintrup, A., and Neely, A. (2019). "Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions." *arXiv preprint arXiv:1905.06076*. 15

Phillips, D. B. and Smith, A. F. (1996). "Bayesian model comparison via jump diffusions." *Markov Chain Monte Carlo in practice*, 215: 239. MR1397970. 14

Piironen, J. and Vehtari, A. (2017a). "On the hyperprior choice for the global shrinkage parameter in the horseshoe prior." In *Artificial Intelligence and Statistics*, 905–913. 6

Piironen, J. and Vehtari, A. (2017b). "Sparsity information and regularization in the horseshoe and other shrinkage priors." *Electronic Journal of Statistics*, 11(2): 5018–5051. 6, 16, 17

Polson, N. G. and Ročková, V. (2018). "Posterior concentration for sparse deep learning." In *Advances in Neural Information Processing Systems*, 930–941. MR3796894. doi: https://doi.org/10.1109/tnnls.2017.2665555. 8, 10

Stefanski, L. A. (1991). "A normal scale mixture representation of the logistic distribution." *Statistics & Probability Letters*, 11(1): 69–70. MR1093420. doi: https://doi.org/10.1016/0167-7152(91)90181-P. 11

Sun, S., Chen, C., and Carin, L. (2017). "Learning structured weight uncertainty in Bayesian neural networks." In *Artificial Intelligence and Statistics*, 1283–1292. 15

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). "Functional variational Bayesian neural networks." *arXiv preprint arXiv:1903.05779*. 15, 19

Swiatkowski, J., Roth, K., Veeling, B. S., Tran, L., Dillon, J. V., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). "The k-tied normal distribution: A compact parameterization of Gaussian mean field posteriors in Bayesian neural networks." *arXiv preprint arXiv:2002.02655*. 15

Sykacek, P. (2000). "On input selection with reversible jump Markov chain Monte Carlo sampling." In *Advances in Neural Information Processing Systems*, 638–644. 14

Tibshirani, R. (2011). "Regression shrinkage and selection via the lasso: a retrospective." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3): 273–282. MR2815776. doi: https://doi.org/10.1111/j.1467-9868.2011.00771.x. 19

Van Der Pas, S. L., Kleijn, B. J., and Van Der Vaart, A. W. (2014). "The horseshoe estimator: Posterior concentration around nearly black vectors." *Electronic Journal of Statistics*, 8(2): 2585–2618. MR3285877. doi: https://doi.org/10.1214/14-EJS962. 16

Vehtari, A. (2001). *Bayesian model assessment and selection using expected utilities*. Helsinki University of Technology. MR2715465. 14

Wang, Y. and Ročková, V. (2020). "Uncertainty quantification for sparse deep learning." *arXiv preprint arXiv:2002.11815*. 8

Wenzel, F., Roth, K., Veeling, B. S., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). "How good is the Bayes posterior in deep neural networks really?" *arXiv preprint arXiv:2002.02405*. 26

Wilson, A. G. and Izmailov, P. (2020). "Bayesian deep learning and a probabilistic perspective of generalization." *arXiv preprint arXiv:2002.08791*. MR3724986. doi: https://doi.org/10.1214/17-BA1082. 3, 15, 26

Zhang, Y. and Bondell, H. D. (2018). "Variable selection via penalized credible regions with Dirichlet–Laplace global-local shrinkage priors." *Bayesian Analysis*, 13(3): 823–844. MR3807868. doi: https://doi.org/10.1214/17-BA1076. 10, 14

Zhang, Y. D., Naughton, B. P., Bondell, H. D., and Reich, B. J. (2020). "Bayesian regression using a prior on the model fit: The R2-D2 shrinkage prior." *Journal of the American Statistical Association*, 1–13. 14