Understanding the Trade-offs in Accuracy and Uncertainty Quantification: Architecture and Inference Choices in Bayesian Neural Networks

Alisa Sheinkman[1] (✉) and Sara Wade[1]

School of Mathematics and Maxwell Institute for Mathematical Sciences,
University of Edinburgh, Edinburgh, UK
a.sheinkman@sms.ed.ac.uk, sara.wade@ed.ac.uk

Abstract. As modern neural networks get more complex, specifying a model with high predictive performance and sound uncertainty quantification becomes a more challenging task. Despite some promising theoretical results on the true posterior predictive distribution of Bayesian neural networks, the properties of even the most commonly used posterior approximations are often questioned. Computational burdens and intractable posteriors expose miscalibrated Bayesian neural networks to poor accuracy and unreliable uncertainty estimates. Approximate Bayesian inference aims to replace unknown and intractable posterior distributions with some simpler but feasible distributions. The dimensions of modern deep models, coupled with the lack of identifiability, make Markov chain Monte Carlo (MCMC) tremendously expensive and unable to fully explore the multimodal posterior. On the other hand, variational inference benefits from improved computational complexity but lacks the asymptotical guarantees of sampling-based inference and tends to concentrate around a single mode. The performance of both approaches heavily depends on architectural choices; this paper aims to shed some light on this, by considering the computational costs, accuracy and uncertainty quantification in different scenarios including large width and out-of-sample data. To improve posterior exploration, different model averaging and ensembling techniques are studied, along with their benefits on predictive performance. In our experiments, variational inference overall provided better uncertainty quantification than MCMC; further, stacking and ensembles of variational approximations provided comparable accuracy to MCMC at a much-reduced cost.

Keywords: Approximate Bayesian Inference · Bayesian Deep Learning · Ensembles · Out-of-Distribution · Uncertainty Quantification.

1 Introduction

Despite the tremendous success of deep learning in areas such as natural language processing [45] and computer vision [27,8], often there is no clear understanding of why a particular model performs well [55,44]. Even though the universal

approximation theorem guarantees that a wide enough feed-forward neural network with a single hidden layer can express any smooth function [24], in practice, constructing a model which is not only expressive but generalizes well is challenging. In contrast, the so-called no free lunch theorem [51] dictates that there is no panacea to solve every problem, and one should be careful when designing a model appropriate to the task. Many modern machine learning models are over-parametrized and prone to overfitting, especially given the limited size of the dataset. Complex problems demand exploring bigger model spaces, and there is a danger of choosing an excessively over-parametrized model, which is going to overfit and have a high variance. Additionally, conventional deep models do not offer human-understandable explanations and lack interpretability [31]. By default, classical neural networks do not address the uncertainty associated with their parameters and whilst there exist proposals enabling neural networks (NNs) to provide some uncertainty estimates, they are often miscalibrated [17]. As a result, these models are typically overconfident, provide a low level of uncertainty even when data variations occur [38], and are easily fooled and are susceptible to adversarial attacks [44,35]. At the same time, reliable uncertainty quantification (UQ) is crucial for any decision-making process, and it is not enough to obtain a point estimate of the prediction.

The key distinguishing property of the Bayesian framework is that it incorporates domain expertise and deals with uncertainty quantification in a principled way: by marginalizing with respect to the posterior distribution of parameters. As a result, Bayesian models are more resistant to distribution shifts and can improve the accuracy and calibration of classical deep models [50]. Nevertheless, the reliability of uncertainty estimates and the gap between within-the-sample and out-of-sample performance still require improvement [11]. The posterior distributions arising in Bayesian neural networks (BNNs) are analytically unavailable and highly multimodal, and the core challenge lies in estimating the posterior [39]. One should not only find a model that matches the task but, as importantly, achieve the alignment between the model and the applied inference algorithm [15]; and the most theoretically grounded sampling methods and approximation techniques are limited by the computing budget, size of the dataset, and sheer number of parameters. We list several characteristics of classical and Bayesian neural networks in the Table 1.

Outline. In this work, we consider some of the challenges and nuances of Bayesian neural networks and evaluate the performance with different architectures and for different posterior inference algorithm choices. Specifically, we study the sensitivity of BNNs to the choice of width in Section 2.3, depth in Section 2.4, and investigate the performance of BNNs under the distribution shift in Section 2.5. Across all the experiments in Section 2, we observe that for different inference algorithms, one model can provide strikingly diverse performances. The challenge of comparative model assessment is addressed in Section 3.1, where we introduce the estimated pointwise loglikelihood as a measure of model utility. While given some set of models, the Bayesian approach has the potential to deal with the model choice by comparing posterior model probabilities, such com-

Table 1: Some of the challenges and properties of classical and Bayesian neural networks.

Property	Classical NN	Bayesian NN
Interpretability	poor	improved ✓
Robustness to OOD	poor	improved ✓
Adversarial attacks	sensitive	less sensitive ✓
Overconfidence	typical	less typical ✓
Training outcome	point estimate $\cdot$	posterior distribution $\mathbb{P}$
Incorporate prior	no	yes
Require initialization	yes	yes

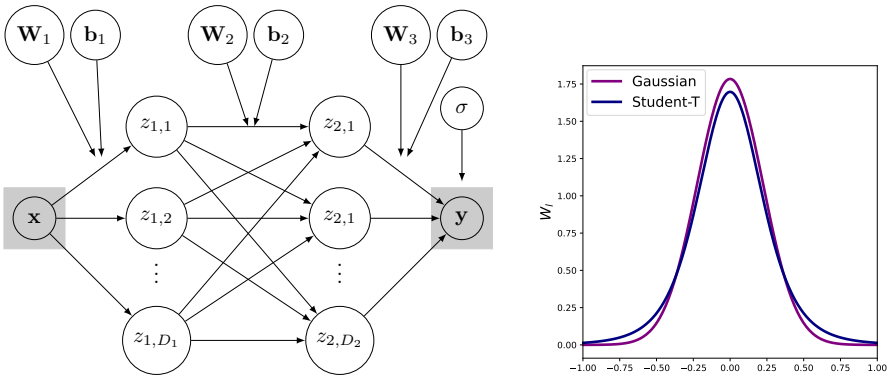
parison tends to favour one candidate disproportionally strongly [36]. Thus, the classical Bayesian model averaging (BMA) based on model probabilities [22] is only optimal if the true model is among the comparison set. In response to the limitations of BMA, in Sections 3.2 and 3.3 we consider ensembling, stacking and pseudo-BMA [54].

2 Empirical Study of Limiting Scenarios

2.1 Architecture Components

Whilst the dimensions of the input and the output are determined by the dimensionality of the data set, the dimension of the weight space plays an essential part in specifying neural networks and can be tuned to improve prediction performance. In the case of feed-forward neural networks, this amounts to finding optimal depth and width. While the universal approximation theorem advocates for single-layer neural networks [24], variants of the universal approximation theorem exist for deeper networks [32,19]. Further, deep neural networks gained popularity due to their expressiveness and tremendous success in real-world applications, allowed by the increase in available computing power [5]. At the same time, the more parameters one has, the more nuanced the choice of the model becomes. No matter what the prediction task is, overly complex models suffer from the curse of dimensionality which causes not only poor performance but also computational problems.

On a slightly different line, we recall the seminal result first obtained for neural networks with one hidden layer [34] and then extended to arbitrary depth [33] which states that under general conditions, as the width of a BNN tends to infinity, the distribution of the network's output induced by the prior converges to a Gaussian process (GP) with a neural network kernel, also known as the neural network GP (NNGP); there is a similar correspondence relating GPs and distributions induced by the posterior [25]. When defining BNNs, choosing a prior and understanding how properties and prior beliefs on the weight space translate to the functions is a major challenge. Generally, we require priors which



(a) Example of the DAG of the neural network when $L = 2$.

(b) Examples of priors of $\boldsymbol{W}_l$ for $l = 2, \ldots, L+1$ and $D_l = 20$.

Fig. 1: Examples of the directed acyclic graph (DAG) of the neural network and of the priors used in the experiments.

are: (1) interpretable, e.g. we want to be able to specify the hyperparameters of the prior based on the task at hand; (2) have large support, i.e. prior should not concentrate around a small subset of the parameter space; (3) lead to feasible inference and favour reasonable approximations of the posterior and predictive distributions.

Finally, to specify any neural network, one needs to choose the activation function, which (apart from being nonlinear) is required to be differentiable. In our experiments, we consider the widely-used rectified linear unit function (ReLU) defined as $\max(0, x)$, which switches the negative inputs off and leaves the positive ones unchanged, as well as the sigmoid activation function defined as $\sigma(x) = \exp(x)/(\exp(x) + 1)$.

2.2 Setup of the Experiments

In the experiments, we consider the following BNN, illustrated by the Figure 1a

$$\begin{aligned} &\boldsymbol{y} \sim \mathrm{N}\left(\boldsymbol{b}_{L+1} + \boldsymbol{W}_{L+1}\boldsymbol{z}_L, \boldsymbol{\sigma}\right), \quad \boldsymbol{\sigma} \sim |\mathrm{N}(0, 0.001)| \\ &\boldsymbol{z}_l = g\left(\boldsymbol{b}_l + \boldsymbol{W}_l \boldsymbol{z}_{l-1}\right) \text{ for } l = 1, \ldots, L, \end{aligned} \tag{1}$$

where we consider two different choices of activations g, namely, the ReLU and the sigmoid; $|\mathrm{N}(,)|$ denotes a half-normal distribution; and $\boldsymbol{z}_0 = \boldsymbol{x}$. We consider two possible choices of priors on the weights (illustrated by Figure 1b): (i) Gaussian priors as the most conventional choice [1,12]; (ii) Student-t priors, motivated by the observation that empirical weight distributions of SGD-trained networks are heavy-tailed [13,18]. We finish specifying the model by placing Gaussian

priors on the biases, that is:

$$\boldsymbol{W}_1 \sim F\left(0, \frac{1}{4L}\right), \quad \boldsymbol{W}_l \sim F\left(0, \frac{4}{D_{l-1}}\right) \quad \text{for} \quad l = 2, \ldots, L+1,$$
$$\boldsymbol{b}_l \sim \mathrm{N}\left(0, \frac{1}{4L}\right) \quad \text{for} \quad l = 1, \ldots, L+1,$$

where the notation $F(\mu, \sigma^2)$ represents a distribution with mean μ and scale σ, and specifically, here is chosen as either Gaussian or Student-t with 5 degrees of freedom. To avoid divergence in wider networks and mitigate the damage caused by the nonlinear deformation [20], the weights' variance is scaled by the inverse of the preceding layer's width.

The BNN defined by Equation (1) and trained with automatic differentiation variational inference (ADVI) [28], which assumes a mean-field (diagonal) Gaussian variational family, is referred to as mfVIR or mfVIS, depending on the choice of the activation: ReLU or sigmoid, respectively. The model trained with the Hamiltonian Monte Carlo (HMC), using the No U-Turn Sampler (NUTS) [23] is denoted as HMCR or HMCS. For simplicity, we often refer to one-layer neural networks of particular width D as to mfVIRD, mfVISD, HMCRD or HMCSD (e.g. one-layer BNN with 20 hidden units and ReLU activation trained with mean-field VI is called mfVIR20). All experiments are implemented with Numpyro [40], ArviZ [29], JAX [4] and Flax [21]. We record the run time of the approximate inference (TT), the root mean squared error RMSE and empirical coverage for the function and observations (EC). Note that we compute empirical coverage as a fraction of observations contained within the 95% confidence interval (CI), this means that in the ideal settings the computed EC should equal to 0.95. If EC > 0.95 then the confidence intervals are too wide; a worse scenario occurs when EC < 0.95 as it means that the CIs are too narrow and the model is overconfident in predictions. Details on the computed metrics and the corresponding formulas are discussed in the supplementary, where we provide further information on the initialization and parameters for the inference algorithms[1].

The absence of the test log-likelihood among the recorded metrics is motivated by the observation that the higher test log-likelihood does not necessarily correspond to a more accurate posterior approximation nor to lower predictive error (such as RMSE) [7].

2.3 Increasing the Width of the Network

We consider a simple synthetic dataset with one-dimensional input and output:

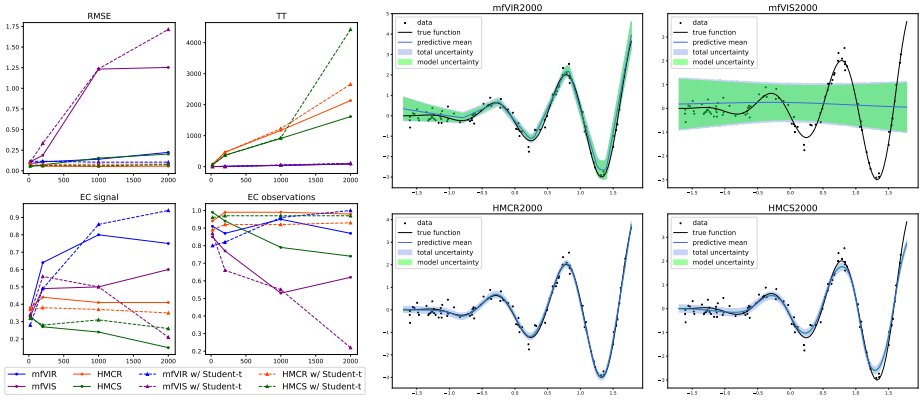
$$\boldsymbol{x} \sim \mathrm{Unif}([0, 2]), \ \boldsymbol{y} = \sin(10\boldsymbol{x})\boldsymbol{x}^2 + \boldsymbol{\epsilon}, \ \boldsymbol{\epsilon} \sim \mathrm{N}(0, 0.25).$$

The training data $\boldsymbol{\mathcal{D}}$ consists of $N = 500$ observations and the new data for testing $\tilde{\boldsymbol{\mathcal{D}}}$ consists of $\tilde{N} = 100$ observations. We first study the performances

[1] The code is available on GitHub.

of mfVIR, mfVIS, HMCR and HMCS with 1 hidden layer and either Gaussian or Student-t priors as the width increases, and illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units by the Figure 2a. The predictions of the four combinations of activation and inference algorithm with Gaussian priors when $D_1 = 2000$ are provided on the Figure 2b; similar results were obtained when weights have Student-t distribution, the figures are presented in the supplementary. For either choice of priors, performance of the mfVIS dips with the increase in the dimension of the hidden layer; moreover, for $D_1 = 1000$ and $D_1 = 2000$ its posterior predictive distribution fails to capture the data, and, in fact, degenerates to the prior (Figure 2b). An explanation of why such behaviour occurs was obtained via the correspondence of Gaussian processes and BNNs. While as the width increases the true posterior of a BNN converges to a NNGP posterior [25], any optimal mean-field Gaussian variational posterior of a BNN with odd (up to a constant offset) Lipschitz activation function converges to the prior predictive distribution of the NNGP [6]. In other words, the mean-field variational approximations of wide BNNs with sigmoid activations ignore the data. If one abandons the mean-field assumption and proposes a full-rank variational family, then using variational inference (VI) for wider networks would take at least a hundred times more time than using HMC, which undermines the benefits of using VI. Such degenerate behaviour is not observed with HMC(Figure 2b, but this comes at a significant increase in training time. For wider networks, the HMCR model exhibits a better performance than the HMCS both in terms of accuracy and uncertainty quantification. In terms of predictive accuracy, HMC is preferred over mfVI in all of the combinations of the activation function and width. However, in terms of uncertainty quantification, the HMC is inferior to mfVI (with one exception of a BNN with Student-t priors, sigmoid activation and 2000 hidden units). In our experiment, HMC underestimates the uncertainty of the signal much more than VI (Figures 2a and 2b). Note that whilst variational inference is often cursed to underestimate the uncertainty[46], that is not always the case [3,14]. Markov chain Monte Carlo (MCMC) methods are known to struggle to effectively explore multimodal posteriors [39,26], and a lack of uncertainty could be a result of poor mixing of the chain.

General summary. In wider networks, the ReLU is preferred over the sigmoid activation for both HMC and mfVI. Crucially, **when it comes to the mean-field VI the sigmoid activation should only be used when the limited width is suitable for the task at hand**. It is reasonable to suppose that the same could be said about any odd (up to adding a constant) activation function. Further, while the HMC was preferred over the mfVI when looking at accuracy alone, the required computational resources could be an obstacle. Moreover, uncertainty quantification is far from ideal for HMC (CIs are too narrow for the signal); instead, mfVI with the ReLU achieves a good balance between accuracy, UQ, and time, particularly for wider networks.



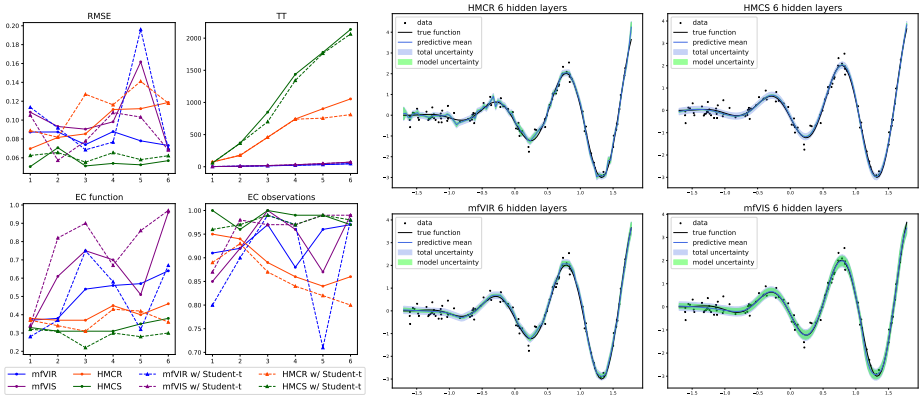
(a) Metrics of methods as the number of hidden units increases.

(b) Predictions and uncertainty estimates for each method with $D_1 = 2000$ and Gaussian priors.

Fig. 2: Predictive performance of BNNs as the width increases.

2.4 Increasing the Depth of the Networks

Consider the data of Section 2.3 and neural networks defined by Equation (1) with the number of layers L varying from 1 to 6 and a fixed number of hidden units in each layer $D_h = 20$. Figure 3a provides the recorded metrics, and Figure 3b illustrates the predictions of the four combinations of activation and inference algorithm with $L = 6$ and Gaussian priors (analogous figures for Student-t priors are provided in the supplementary). First, observe that overall both RMSE and empirical coverage of mfVIR approximations improve with the increase of depth, with one exception of $L = 5$ and Student-t priors, when the prediction quality of the network drops drastically. The mfVIS follows a similar pattern, except for the case of $L = 5$ and Gaussian priors. Indeed, the approximate posteriors of deep neural networks obtained with the mean-field variational inference were shown to be as flexible as the much richer approximate posteriors of shallower BNNs [10]. We do not obtain the same improvement in the prediction quality of models trained with HMC: for either choice of priors, the performance of HMCR falls, whilst the HMCS does not improve as the depth increases. This undesirable behaviour could be a result of the multimodality of distributions in overparametrized models combined with the challenges of MCMC in exploring the high-dimensional space [26,39]. Compared to the findings of Section 2.3, we note that the deeper NNs are less sensitive to the choice of the activation function. It is needless to say that the HMC algorithm scales rather poorly, and as the number of layers changes from $L = 1$ to $L = 6$, the time needed to train HMCR and HMCS gets more than 15 and 30 times greater, respectively. We note that for models with more than one hidden layer, training of the network with sigmoid activations takes roughly twice as much time as the network with ReLU. The striking discrepancy in training times could arise due to the difference in the leapfrog integrator step sizes [2].



(a) Metrics of methods as the number of hidden layers increases.

(b) Predictions and uncertainty estimates for each method when $L = 6$ and Gaussian priors.

Fig. 3: Prediction performance of BNNs as the depth increases.

General summary. In terms of the training time, HMC becomes less and less feasible with the increase in depth. With the need to explore high-dimensional parameter spaces, multimodality of the posteriors should be kept in mind as an arising challenge for both mfVI and HMC. **In terms of the balance between accuracy and UQ, the mean-field variational inference with ReLU activation function is able to outperform MCMC with the increase in depth.**

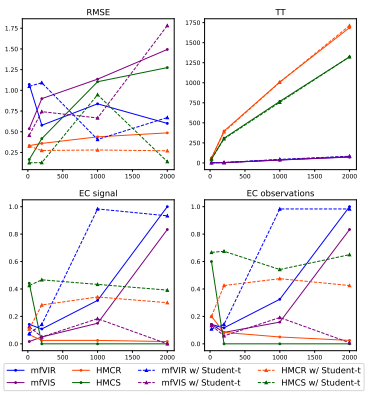
2.5 Out-of-Distribution Prediction

While it is not surprising that the accuracy and the quality of uncertainty quantification of any model decreases under a distribution shift, reliable uncertainty estimates that are robust to the out-of-distribution (OOD) data become exceptionally important in safety-critical applications. The challenge is especially intricate since better accuracy and lower calibration error of a certain model on the in-domain data do not imply better accuracy and lower calibration error in the OOD settings [38]. Here, we wish to validate the models' predictive abilities when the test data points come from previously unseen regions of data space. The kind of out-of-distribution data we consider could be described as 'complement-distributions', such data arises in open-set recognition or could be the result of an adversary [9]. Note that in Section 3.3 as well as in the supplementary, we consider a much milder example with 'related-distributions' data. We split the training data used in Sections 2.3 and 2.4 into the train and test data covering complement regions of the function. Specifically, $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}_c \sqcup \tilde{\boldsymbol{\mathcal{D}}}_c$, the observed data $\boldsymbol{\mathcal{D}}_c$ consists of $N = 370$, the new data $\tilde{\boldsymbol{\mathcal{D}}}_c$ consists of $\tilde{N} = 130$

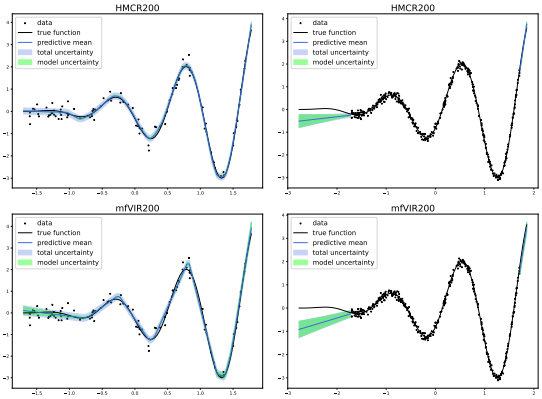
and the observed and the new data are disjoint (see Figure 4):

$$\boldsymbol{\mathcal{D}}_c = \{(x_n, y_n) \mid x_n \in [-1.7, 1.7]\},$$
$$\tilde{\boldsymbol{\mathcal{D}}}_c = \{(x_n, y_n) \mid x_n \in [-2.8, -1.7) \cup (1.7, 1.9)]\}.$$

Strictly speaking, we do not expect any model to be robust to such an extreme case and, mainly, want to asses and better understand the quality of the uncertainty estimates. In this experiment, we are hoping that the relationship between the distributions of the observed and the new data makes this challenge somewhat tractable. On Figure 4a we illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units; Figure 4b compares non-OOD and OOD predictions obtained by the BNNs with ReLU activation, Gaussian priors and $D_1 = 200$. The poor performance of the mfVIS, especially for wider networks, is not surprising, however, we notice that for wide networks HMCS with Gaussian priors suffers from much higher RMSE than HMCS with Student-t priors and mfVIR and HMCR with either choice of priors. And while HMCR has a lower RMSE than any model trained with mean-field VI, the ability of HMC to capture the uncertainty deteriorates, and it becomes overconfident. Whilst HMCR200 and mfVIR200 do not show any of the expected increase in the uncertainty, on certain regions both methods are able to provide accurate predictive mean (see Figure 4b for examples with Gaussian priors, the right-hand side region of the function, where $x > 1.5$). Finally, as the width of the network increases, mfVIR outperforms all of the methods.



(a) Out-of-sample metrics of all the methods compared.



(b) Within-the sample (left) and out-of-sample (right) predictions and uncertainty estimates for $D_1 = 200$, ReLU activation and Gaussian priors.

Fig. 4: Out-of-distribution prediction for the 'complement-distributions' data.

General summary. In terms of the accuracy alone, the HMC with ReLU is more robust to the out-of-distribution data, however, that comes with the largest

computational costs among all the models. We already saw in Section 2.3 that uncertainty quantification with HMC degrades with increasing width. In OOD settings, this becomes even more extreme, with very overconfident predictions that do not cover the truth (an empirical coverage of almost zero). **Finally, with the increase in depth, in the extreme OOD settings, the mfVI with ReLU becomes almost as accurate as HMC with ReLU and provides better UQ at a much lower cost.**

3 Bayesian Model Averaging and Stacking

3.1 Predictive Methods for Model Assessment

When considering synthetic datasets, we can choose a desired metric and sample any number of data points, so that evaluation of the model's performance becomes trivial. For example, in Section 2.5 we have specifically created an extreme case when the training data $\boldsymbol{\mathcal{D}}_c$ and the new data $\tilde{\boldsymbol{\mathcal{D}}}_c$ were covering disjoint regions of the true function. In reality, the new previously unseen data is not available, and one can only estimate the expected out-of-sample predictive performance. Suppose that we only observe $\boldsymbol{\mathcal{D}}$, the unseen observations $\tilde{\boldsymbol{\mathcal{D}}}$ are generated by $p_t(\tilde{\boldsymbol{\mathcal{D}}})$, and we wish to be able to assess the generalization ability of the model without having access to the test data. To keep the notation simple, we omit the dependency on $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ when writing down the posteriors in this section. Given a new data point $\tilde{y}_n$, the log score $\log p(\tilde{y}_n|\boldsymbol{\mathcal{D}})$ is one of the most common utility functions used in measuring the quality of the predictive distribution. The log score benefits from being a local and proper scoring rule [48]. Then, the expected log pointwise predictive density for a new dataset serves as a measure of the predictive accuracy of a given model:

$$\text{elpd} = \sum_{n=1}^{\tilde{N}} \int p_t(\tilde{\mathrm{D}}_n) \log p(\tilde{y}_n|\boldsymbol{\mathcal{D}}) \mathrm{d}\tilde{\mathrm{D}}_n,$$

where $p(\tilde{y}_n|\boldsymbol{\mathcal{D}})$ is model's posterior predictive distribution. In the absence of $\tilde{\boldsymbol{\mathcal{D}}}$, one might obtain an estimate of the expected log pointwise predictive density by re-using the observed $\boldsymbol{\mathcal{D}}$. Here, we review the approach that employs leave-one-out cross-validation (LOO-CV), which can be seen as a natural framework for assessing the model's predictive performance [47].

To obtain the Bayesian leave-one-out cross-validation (LOO-CV) estimate of the expected utility $\widehat{\text{elpd}}_{\text{loo}}$ and avoid re-fitting the model N times, one could use importance sampling. However, the classical importance weights would have a large variance, and the obtained estimates would be noisy. Recently, the problem was solved with Pareto smoothed importance sampling (PSIS), which allows evaluating the LOO-CV expected utility in a reliable yet efficient way [47]:

$$\widehat{\text{elpd}}_{\text{loo}} = \sum_{n=1}^{N} p(y_n|x_n, \boldsymbol{\mathcal{D}}_{-n}) = \sum_{n}^{N} \log \left(\frac{\sum_{s=1}^{S} r_i^s p(y_n|\boldsymbol{\theta}^s)}{\sum_{s=1}^{S} r_i^s} \right), \tag{2}$$

where r_i^s are the smoothed importance weights, which benefit from smaller variance than the classical weights. We refer to the individual logarithms in the sum as $\widehat{\text{elpd}}_{\text{loo},n}$. The advantage of PSIS is that the estimated shape parameter of the Pareto distribution provides a diagnostic of the reliability of the resulting expected utility. Although methods of model selection which reuse the data can be vulnerable to overfitting when the size of the dataset is too small and/or the data is sparse, it is (relatively) safe to use cross-validation to compare a small number of models and given a large enough dataset [49]. In the supplementary material, we implement $\widehat{\text{elpd}}_{\text{loo}}$ in the empirical experiment, where we additionally consider posterior predictive checks (PPC) and an alternative to the LOO-CV approach of estimating the expected log pointwise utility.

3.2 Alternatives to Classical Bayesian Model Averaging

Let $\mathcal{M} = \{M_1, \ldots, M_K\}$ be a collection of models and denote the parameters of each of the M_k as $\boldsymbol{\theta}_k$. The assumptions one has on the prediction task and on $\mathcal{M}$ with respect to the true data-generating process can be categorized into three scenarios: $\mathcal{M}$-closed, $\mathcal{M}$-open and $\mathcal{M}$-complete. If $M_k \in \mathcal{M}$ for some k recovers the true data generating process, then we are in the $\mathcal{M}$-closed case. The task is $\mathcal{M}$-complete if there exists a true model but it is not included in $\mathcal{M}$ (e.g. for computational reasons). Finally, we are in the $\mathcal{M}$-open scenario when the true model is not in $\mathcal{M}$ and the data-generating mechanism cannot be conceptually formalized to provide an explicit model [48]. The Bayesian framework allows to define the probabilities over the model space, and for the $\mathcal{M}$-closed case, classical Bayesian Model Averaging (BMA) would give optimal performance. The BMA solution provides an averaged predictive posterior as [22]

$$p(\tilde{\boldsymbol{y}} \mid \boldsymbol{\mathcal{D}}) = \sum_{k=1}^{K} p(\tilde{\boldsymbol{y}} \mid \boldsymbol{\mathcal{D}}, M_k) p(M_k \mid \boldsymbol{\mathcal{D}}), \tag{3}$$

$$\text{where } p(M_k \mid \boldsymbol{\mathcal{D}}) \propto p(\boldsymbol{\mathcal{D}} \mid M_k) p(M_k). \tag{4}$$

However, in the $\mathcal{M}$-open and $\mathcal{M}$-complete prediction tasks, BMA is not appropriate as it gives a strong preference to a single model and so assumes that this particular model is the true one. Now, if we replace the weights $p(M_k|\boldsymbol{\mathcal{D}})$ with the products of Bayesian LOO-CV densities $\prod_{n=1}^{N} p(y_n \mid x_n, \boldsymbol{\mathcal{D}}_{-n}, M_k)$, we arrive at pseudo-Bayesian model averaging (pseudo-BMA). In other words, the weights w_k of pseudo-BMA are proportional to the estimated log pointwise predictive density $\exp(\widehat{\text{elpd}}_{\text{loo}}^{k})$ introduced in Section 3.1. One could further correct

each $\widehat{\text{elpd}}_{\text{loo}}$ estimate of Equation (2) by the standard errors and obtain

$$w_k = \frac{\exp(\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}})}{\sum_{k=1}^{K} \exp(\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}})},$$
$$\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}} = \widehat{\text{elpd}}_{\text{loo}}^{k} - \frac{1}{2}\sqrt{\sum_{n=1}^{N}\left(\widehat{\text{elpd}}_{\text{loo},n}^{k} - \frac{\widehat{\text{elpd}}_{\text{loo}}^{k}}{N}\right)^2},$$

where for each model M_k we find $\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}}$ by utilizing a log-normal approximation. Fortunately, we have already seen that these densities can be efficiently estimated with PSIS.

An alternative way to obtain the averaged predictive posterior given the set of $p(\tilde{\boldsymbol{y}} \mid \boldsymbol{\mathcal{D}}, M_k)$ is to employ the stacking approach [54]. Define the set $S^K = \{\boldsymbol{w} \in [0,1]^K | \sum_k^K w_k = 1\}$, then the stacking weights are found as the optimal (according to the logarithmic score) solution of the following problem

$$\begin{aligned}\boldsymbol{w} &= \max_{\boldsymbol{w}\in S^K} \frac{1}{N}\sum_{n=1}^{N}\log\sum_{k=1}^{K} w_k p(y_n \mid \boldsymbol{\mathcal{D}}_{-n}, M_k),\\ &= \max_{\boldsymbol{w}\in S^K} \frac{1}{N}\sum_{n=1}^{N}\log\sum_{k=1}^{K} w_k \left(\frac{\sum_{s=1}^{S} r_i^s p(y_n \mid \boldsymbol{\theta}_k^s, M_k)}{\sum_{s=1}^{S} r_i^s}\right),\end{aligned}$$

where a PSIS estimate of the predictive LOO-CV density is used, and r_i^s are the smoothed (truncated) importance weights.

Finally, we recall that deep ensembles of classical non-Bayesian NNs [30] behave similarly to Bayesian model averages, and both lead to solutions strongly favouring one single model [50]. In contrast, the ensembles of BNN posteriors in Equation (3) with $p(M_k \mid \boldsymbol{\mathcal{D}}) = K^{-1}$ can be seen as a trivial case of BMA, which combines models and does not give preference to a single solution. Alternatively, when implementing variational inference and combining BNNs, the analogy can be drawn with the simplified version of adaptive variational Bayes, which combines variational posteriors with certain weights and, under certain conditions, attains optimal contraction rates [37].

3.3 Ensembles and Averages

We compare three model averaging methodologies: deep ensembles of Bayesian neural networks, stacking and pseudo-BMA based on PSIS-LOO [54]. We do not consider the Bayesian Bootstrap (BB) [43] motivated by the recent observation that in the settings of modern neural networks deep ensembles of non-Bayesian NNs and BB are equivalent, and both are often misspecified [52]. Combining several estimates of BNNs can be effective not only when predictions are coming from different models, but also when dealing with several predictions obtained by the same model [37]. This is of particular use for multimodal posteriors arising

in BNNs, where different modes could be explored by random initializations [54]. Additionally, recall that the ELBO, the objective of variational inference, is a non-convex function, so that the optimum is only local and depends on the starting point. We note that combining models trained with HMC and VI would be meaningless for several reasons. First of all, training a set of HMC models becomes rather expensive: for instance, training the HMCR20 once takes the same amount of time as 35 trainings of mfVIR20. Second, the estimates of the log pointwise predictive densities (provided in the supplementary) for HMC and VI have different scales and are not easily compared; in this case, the result of averaging HMC and VI would be equivalent to classical BMA.

Now consider the mfVIR20 model with Gaussian priors and the 'complement-distributions' data of Section 2.5. We choose 10 random initialization points, obtain 10 posterior predictive distributions and compute estimated expected log pointwise predictive densities. We then construct ensemble, pseudo-BMA and stacking approximations; the results are illustrated Figure 5. Ensembling and stacking are superior to pseudo-BMA, which has worse accuracy and fails to capture any uncertainty. Similar results for the mfVIR20 model with Student-t priors are provided in the supplementary material. While here we focus only on models with 20 hidden units, it would be reasonable to assume that not only do the performance of individual models depend on architectural choices, but the model averaging techniques are themselves influenced by these modelling choices (for empirical justification of this claim, the reader is referred to the supplementary material, where we consider the data simulated when designing a novel rocket booster [16,42] and provide the results of ensembling and averaging for various architectures).

Given the nature of the test data we use, the predictions as well as the $\widehat{\text{elpd}}_{\text{loo}}$ estimates may be unreliable. Thus, we consider a simpler data-generating mechanism in which test data comes from a slightly broader region; such a scenario could be called an OOD task with 'related-distributions' [9]. Specifically, the data are generated as follows:

$$\boldsymbol{x} \sim \text{Unif}([0,1]), \quad \boldsymbol{y} = \sin(10\boldsymbol{x})\boldsymbol{x}^2 + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim 0.05\text{N}(0,1).$$

The data for training $\boldsymbol{\mathcal{D}}_r$ and the testing $\tilde{\boldsymbol{\mathcal{D}}}_r$ consist of $N = 450$ and $\tilde{N} = 50$ observations, respectively, where $\tilde{\boldsymbol{\mathcal{D}}}_r$ comes from the broader region than $\boldsymbol{\mathcal{D}}_r$, i.e. $(\min_{n=1\ldots N}(x_n), \max_{n=1\ldots N}(x_n)) \subsetneq (\min_{n=1\ldots\tilde{N}}(\tilde{x}_n), \max_{n=1\ldots\tilde{N}}(\tilde{x}_n))$. For 10 posterior predictive distributions of mfVIR20 with Gaussian priors (results for Student-t priors are provided in the supplementary), we compare ensembling, pseudo-BMA and stacking in Figure 5 (similar results with having Student-t priors are presented in the supplementary, where we additionally provide the results of ensembling and averaging in the deeper networks.). Whilst the total uncertainty estimates of pseudo-BMA are, somewhat, adequate, the model uncertainty is underestimated. Both stacking and deep ensembles lead to improved predictive performance and uncertainty quantification, with stacking showing some better gains compared to deep ensembles (see e.g. improved coverage of stacking on the right-hand side of Figure 5).

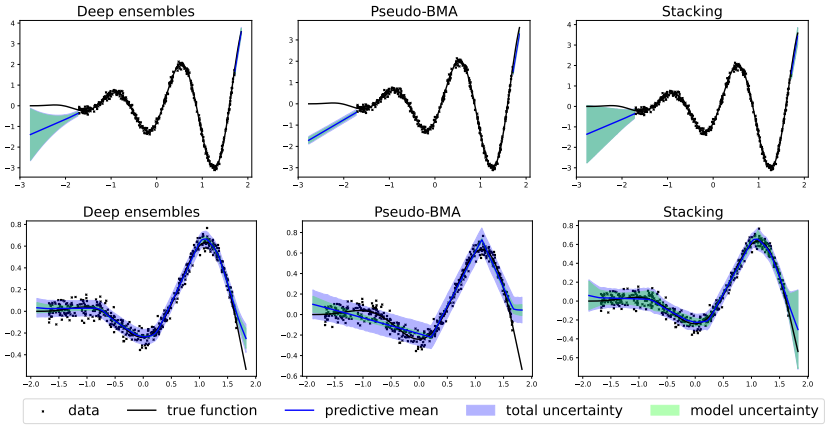


Fig. 5: Predictions obtained by ensembling, stacking and pseudo-BMA when applied to mfVIR20 with Gaussian priors in the 'complement-distributions' (top) and 'related-distributions' (bottom) OOD tasks. Pseudo-BMA is worse than the other methodologies, and stacking provides improvements over DE in uncertainty quantification.

General summary. We observe that, similar to BMA, the pseudo-BMA is not preferable in $\mathcal{M}$-open and $\mathcal{M}$-complete settings. Namely, in 'complement-distributions' and 'related-distributions' experiments, pseudo-BMA was confirmed to be inferior to stacking and ensembles of BNNs both in terms of the predictive accuracy and uncertainty quantification. **Stacking and ensembles of BNNs performed comparable to each other and provided an improvement, with modest gains for stacking, which is especially significant in terms of uncertainty quantification in the OOD setting.**

4 Discussion

The message of an optimist's conclusion could question the common belief that the mean-field variational approximations are generally overly restrictive and do not capture the true posterior and the uncertainty well. Even with increases in computing power, the computational costs of sampling algorithms suggest that it may not be feasible for most modern neural networks and datasets. Moreover, although HMC is often considered as a gold standard, we have seen this may not be the case for BNNs due to complexity and multimodality of the posterior. Indeed, in a variety of experiments considered in Sections 2.3 to 2.5 and 3.3 **mfVI overall provided better uncertainty quantification than HMC,** and in out-of-distribution settings, the empirical coverage of the latter was close to zero. We note that **for single-layer neural networks, HMC outperformed mfVI only in terms of accuracy.** At the same time, for deeper networks and in out-of-distribution scenarios, the accuracy of mfVI was often comparable to HMC.

Further, in Section 2.4 we confirmed that even for slightly deeper networks the time needed for HMC becomes a burden, which makes variational inference a very attractive alternative to sampling. Nevertheless, in Section 2.3 we observed that **the restrictions imposed by the factorized families can obstruct models from effectively learning from the data**. In real-life scenarios where one is required to evaluate the future predictive performance of the model before applying it to the unseen data, the estimate of the expected log pointwise predictive density can serve as a reliable diagnostic and thus, PSIS-LOO estimates can be beneficial for model assessment and combination. In Section 3.3, **stacking and ensembles of BNNs were shown to be a possible solution when dealing with multimodal posteriors, helping to both improve accuracy and uncertainty quantification even in the extreme OOD scenario.** We find that stacked or ensembled variational approximations are competitive to HMC at a much-reduced cost. Finally, we note that overall in our experiments, there was no considerable and systematic difference in the performance between the BNNs with Gaussian and Student-t priors.

This work highlights the model's sensitivity to architectural choices, namely, width, depth and activation function. Future work could study the performance of various more elaborate than Gaussian or Student-t choices of priors placed on the weights, including sparsity-inducing priors which have been shown to improve the accuracy and calibration [3,41]. Further, an important avenue for research is to consider the so-called structured variational inference with less restrictive variational families, and more generally, study the trade-off between the expressiveness of the variational family and scalability. Finally, given the multimodal nature of distributions arising in Bayesian neural networks, a promising avenue for research is to continue improving model combination techniques. This includes developing a better understanding of the number of models required for optimal performance with existing ensembling methods, as well as exploring more advanced approaches such as adaptive variational Bayes frameworks [37] or hierarchical stacking and pointwise model combination [53].

References

1. Arbel, J., Pitas, K., Vladimirova, M., Fortuin, V.: A primer on bayesian neural networks: review and debates. arXiv preprint arXiv:2309.16314 (2023) 4
2. Betancourt, M.J., Byrne, S., Girolami, M.: Optimizing the integrator step size for hamiltonian monte carlo (2015) 7
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: Proceedings of The International conference on machine learning. pp. 1613–1622. PMLR (2015) 6, 15
4. Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: Jax: composable transformations of python+numpy programs (2018), `http://github.com/jax-ml/jax` 5
5. Chatziafratis, V., Nagarajan, S.G., Panageas, I.: Better depth-width trade-offs for neural networks through the lens of dynamical systems. In: Proceedings of The International Conference on Machine Learning. pp. 1469–1478. PMLR (2020) 3

6. Coker, B., Bruinsma, W.P., Burt, D.R., Pan, W., Doshi-Velez, F.: Wide mean-field bayesian neural networks ignore the data. In: International Conference on Artificial Intelligence and Statistics. pp. 5276–5333. PMLR (2022) 6
7. Deshpande, S.K., Ghosh, S., Nguyen, T.D., Broderick, T.: Are you using test log-likelihood correctly? Transactions on machine learning research (2024) 5
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020) 1
9. Farquhar, S., Gal, Y.: What 'out-of-distribution' is and is not. In: NeurIPS ML Safety Workshop (2022) 8, 13
10. Farquhar, S., Smith, L., Gal, Y.: Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. Advances in Neural Information Processing Systems **33**, 4346–4357 (2020) 7
11. Foong, A., Burt, D., Li, Y., Turner, R.: On the expressiveness of approximate inference in bayesian neural networks. Advances in Neural Information Processing Systems **33**, 15897–15908 (2020) 2
12. Fortuin, V.: Priors in bayesian deep learning: A review. International Statistical Review **90**(3), 563–591 (2022) 4
13. Fortuin, V., Garriga-Alonso, A., Ober, S.W., Wenzel, F., Rätsch, G., Turner, R.E., van der Wilk, M., Aitchison, L.: Bayesian neural network priors revisited (2022) 4
14. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016) 6
15. Gelman, A., Vehtari, A., Simpson, D., Margossian, C.C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.C., Modrák, M.: Bayesian workflow (2020) 2
16. Gramacy, R.B., Lee, H.K.H.: Bayesian treed gaussian process models with an application to computer modeling. Journal of the American Statistical Association **103**(483), 1119–1130 (2008) 13
17. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of The International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1321–1330. PMLR (06–11 Aug 2017) 2
18. Gurbuzbalaban, M., Simsekli, U., Zhu, L.: The heavy-tail phenomenon in sgd. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 3964–3975. PMLR (18–24 Jul 2021) 4
19. Hanin, B.: Universal function approximation by deep neural nets with bounded width and relu activations. Mathematics **7**(10), 992 (2019) 3
20. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015) 5
21. Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., van Zee, M.: Flax: A neural network library and ecosystem for JAX (2024), `http://github.com/google/flax` 5
22. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: A tutorial. Statistical Science **14**(4), 382–417 (1999) 3, 11
23. Hoffman, M.D., Gelman, A.: The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research **15**(47), 1593–1623 (2014) 5

24. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359–366 (1989) 2, 3
25. Hron, J., Novak, R., Pennington, J., Sohl-Dickstein, J.: Wide bayesian neural networks have a simple weight posterior: theory and accelerated sampling. In: Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 8926–8945. PMLR (17–23 Jul 2022) 3, 6
26. Izmailov, P., Vikram, S., Hoffman, M.D., Wilson, A.G.G.: What are bayesian neural network posteriors really like? In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 4629–4640. PMLR (18–24 Jul 2021) 6, 7
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (May 2017) 1
28. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. Journal of machine learning research **18**(14), 1–45 (2017) 5
29. Kumar, R., Carroll, C., Hartikainen, A., Martin, O.: Arviz a unified library for exploratory analysis of bayesian models in python. Journal of Open Source Software **4**(33), 1143 (2019) 5
30. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017) 12
31. Lipton, Z.C.: The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (2018) 2
32. Lu, Z., Pu, H., Wang, F., Hu, Z., Wang, L.: The expressive power of neural networks: a view from the width. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6232–6240 (2017) 3
33. Matthews, A.G., Hron, J., Rowland, M., Turner, R., Ghahramani, Z.: Gaussian process behaviour in wide deep neural networks. ICLR (2018) 3
34. Neal, R.: Bayesian learning for neural networks. Springer Science & Business Media **118** (1995) 3
35. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 427–436 (2015) 2
36. Oelrich, O., Ding, S., Magnusson, M., Vehtari, A., Villani, M.: When are bayesian model probabilities overconfident? (2020) 3
37. Ohn, I., Lin, L.: Adaptive variational bayes: Optimality, computation and applications. The Annals of Statistics **52**(1), 335–363 (2024) 12, 15
38. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Advances in Neural Information Processing Systems. vol. 32 (2019) 2, 8
39. Papamarkou, T., Hinkle, J., Young, M.T., Womble, D.: Challenges in Markov chain Monte Carlo for Bayesian neuralnetworks. Statistical Science **37**(3), 425–442 (2022) 2, 6, 7
40. Phan, D., Pradhan, N., Jankowiak, M.: Composable effects for flexible and accelerated probabilistic programming in numpyro. In: Program Transformations for ML Workshop at NeurIPS (2019) 5
41. Polson, N.G., Ročková, V.: Posterior concentration for sparse deep learning. Advances in Neural Information Processing Systems **31** (2018) 15

42. Rogers, S., Aftosmis, M., Pandya, S., Chaderjian, N., Tejnil, E., Ahmad, J.: Automated cfd parameter studies on distributed parallel computers. In: 16th AIAA Computational Fluid Dynamics Conference. p. 4229 (2003) 13
43. Rubin, D.B.: The bayesian bootstrap. The annals of statistics pp. 130–134 (1981) 12
44. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014 (2014) 1, 2
45. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023) 1
46. Trippe, B., Turner, R.: Overpruning in variational bayesian neural networks (2018) 6
47. Vehtari, A., Gelman, A., Gabry, J.: Practical bayesian model evaluation using leave-one-out cross-validation and waic. Statistics and Computing **27**(5), 1413–1432 (Aug 2016) 10
48. Vehtari, A., Ojanen, J.: A survey of bayesian predictive methods for model assessment, selection and comparison. Statistics Surveys **6**, 142 – 228 (2012). `https://doi.org/10.1214/12-SS102` 10, 11
49. Vetari, A., Gabry, J., Magnusson, M., Yao, Y., Gelman, A.: Efficient leave-one-out cross-validation and waic for bayesian models (2019), `https://mc-stan.org/loo` 11
50. Wilson, A.G., Izmailov, P.: Bayesian deep learning and a probabilistic perspective of generalization. Advances in Neural Information Processing Systems **33**, 4697–4708 (2020) 2, 12
51. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Computation **8**(7), 1341–1390 (10 1996) 2
52. Wu, L., A Williamson, S.: Posterior uncertainty quantification in neural networks using data augmentation. In: Proceedings of The 27th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 238, pp. 3376–3384. PMLR (02–04 May 2024) 12
53. Yao, Y., Vehtari, A., Gelman, A.: Stacking for non-mixing bayesian computations: The curse and blessing of multimodal posteriors. The Journal of Machine Learning Research **23**(1), 3426–3471 (2022) 15
54. Yao, Y., Vehtari, A., Simpson, D., Gelman, A.: Using stacking to average bayesian predictive distributions (with discussion). Bayesian Analysis **13**(3) (Sep 2018). `https://doi.org/10.1214/17-ba1091` 3, 12, 13
55. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021) 1