What Are Bayesian Neural Network Posteriors Really Like?

Pavel Izmailov [1] **Sharad Vikram** [2] **Matthew D. Hoffman** [2] **Andrew Gordon Wilson** [1]

Abstract

The posterior over Bayesian neural network (BNN) parameters is extremely high-dimensional and non-convex. For computational reasons, researchers approximate this posterior using inexpensive mini-batch methods such as mean-field variational inference or stochastic-gradient Markov chain Monte Carlo (SGMCMC). To investigate foundational questions in Bayesian deep learning, we instead use full-batch Hamiltonian Monte Carlo (HMC) on modern architectures. We show that (1) BNNs can achieve significant performance gains over standard training and deep ensembles; (2) a single long HMC chain can provide a comparable representation of the posterior to multiple shorter chains; (3) in contrast to recent studies, we find posterior tempering is not needed for near-optimal performance, with little evidence for a "cold posterior" effect, which we show is largely an artifact of data augmentation; (4) BMA performance is robust to the choice of prior scale, and relatively similar for diagonal Gaussian, mixture of Gaussian, and logistic priors; (5) Bayesian neural networks show surprisingly poor generalization under domain shift; (6) while cheaper alternatives such as deep ensembles and SGMCMC can provide good generalization, their predictive distributions are distinct from HMC. Notably, deep ensemble predictive distributions are similarly close to HMC as standard SGLD, and closer than standard variational inference.

1. Introduction

Over the last 25 years, there have been several strong arguments favouring a Bayesian approach to deep learning (e.g., MacKay, 1995; Neal, 1996; Blundell et al., 2015; Gal, 2016; Wilson & Izmailov, 2020). Bayesian inference for neural networks promises improved predictions, reliable uncertainty estimates, and principled model comparison, naturally supporting active learning, continual learning, and decision-making under uncertainty. The Bayesian deep learning community has designed multiple successful practical methods inspired by the Bayesian approach (Blundell et al., 2015; Gal & Ghahramani, 2016; Welling & Teh, 2011; Kirkpatrick et al., 2017; Maddox et al., 2019; Izmailov et al., 2019; Daxberger et al., 2020), with applications ranging from astrophysics (Cranmer et al., 2021) to automatic diagnosis of Diabetic Retinopathy (Filos et al., 2019), click-through rate prediction in advertising (Liu et al., 2017) and fluid dynamics (Geneva & Zabaras, 2020).

However, inference with modern BNNs is distinctly challenging. We wish to compute a Bayesian model average corresponding to an integral over a multi-million dimensional multi-modal posterior, with unusual topological properties like mode-connectivity (Garipov et al., 2018; Draxler et al., 2018), under severe computational constraints.

There are therefore many unresolved questions about Bayesian deep learning practice. Variational procedures typically provide unimodal Gaussian approximations to the multimodal posterior. Practically successful methods such as deep ensembles (Lakshminarayanan et al., 2017; Fort et al., 2019) have a natural Bayesian interpretation (Wilson & Izmailov, 2020), but only represent modes of the posterior. While Stochastic MCMC (Welling & Teh, 2011; Chen et al., 2014; Zhang et al., 2020b) is computationally convenient, it could be providing heavily biased estimates of posterior expectations. Moreover, Wenzel et al. (2020) question the quality of standard Bayes posteriors, citing results where "cold posteriors", raised to a power $1/T$ with $T < 1$, improve performance.

Additionally, Bayesian deep learning methods are typically evaluated on their ability to generate useful, well-calibrated predictions on held-out or out-of-distribution data. However, strong performance on benchmark problems does not imply that the algorithm accurately approximates the true Bayesian model average (BMA).

In this paper, we investigate fundamental open questions in Bayesian deep learning, using multi-chain full-batch Hamiltonian Monte Carlo (HMC, Neal et al., 2011). HMC is a highly-efficient and well-studied Markov Chain Monte

[1]New York University [2]Google Research. Correspondence to: Pavel Izmailov <pi390@nyu.edu>, Andrew Gordon Wilson <andrewgw@cims.nyu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Carlo (MCMC) method that is guaranteed to asymptotically produce samples from the true posterior. However it is enormously challenging to apply HMC to modern neural networks due to its extreme computational requirements: HMC can take *tens of thousands of training epochs* to produce a single sample from the posterior. To address this computational challenge, we parallelize the computation over hundreds of Tensor processing unit (TPU) devices.

We argue that full-batch HMC provides the most precise tool for studying the BNN posterior to date. Indeed, we are not proposing HMC as a computationally efficient method for practical applications. Rather, using our implementation of HMC we are able to explore fundamental questions about posterior geometry, the performance of BNNs, approximate inference, effect of priors and posterior temperature.

In particular, we show: (1) BNNs can achieve significant performance gains over standard training and deep ensembles; (2) a single long HMC chain can provide a comparable performance to multiple shorter chains; (3) in contrast to recent studies, we find posterior tempering is not needed for near-optimal performance, with little evidence for a "cold posterior" effect, which we show is largely an artifact of data augmentation; (4) BMA performance is robust to the choice of prior scale, and relatively similar for diagonal Gaussian, mixture of Gaussian, and logistic priors over weights. This result highlights the importance of architecture relative to parameter priors in specifying the prior over functions. (5) While Bayesian neural networks have good performance for out-of-distribution (OOD) detection, they show surprisingly poor generalization under domain shift; (6) while cheaper alternatives such as deep ensembles and SGMCMC can provide good generalization, their predictive distributions are distinct from HMC. Notably, deep ensemble predictive distributions are similarly close to HMC as standard SGLD, and closer than standard variational inference.

We additionally show how to effectively deploy full batch HMC on modern neural networks, including insights about how to tune crucial hyperparameters for good performance, and parallelize sampling over hundreds of TPUs. Our HMC samples and implementation is a public resource. We hope this resource will serve as a reference in evaluating and calibrating more practical alternatives to HMC, and aid researchers in pursuing a better understanding of approximate inference in Bayesian deep learning.

2. Background

Bayesian neural networks. The goal of classical learning is to find a single best setting of the parameters for the model, typically through maximum-likelihood optimization. In the Bayesian framework, the learner instead infers a *posterior* distribution $p(w|\mathcal{D})$ over the parameters w of the model after observing the data $\mathcal{D}$. The posterior distribution is given by Bayes' rule: $p(w|\mathcal{D}) \propto p(\mathcal{D}|w)p(w)$, where $p(\mathcal{D}|w)$ is the likelihood of $\mathcal{D}$ given by the model with parameters w, and $p(w)$ is the prior distribution over the parameters. The predictions of the model on a new test example x are then given by the *Bayesian model average* (BMA)

$$p(y|x,\mathcal{D}) = \int_w p(y|x,w)p(w|\mathcal{D})dw, \tag{1}$$

where $p(y|x,w)$ is the predictive distribution for a given value of the parameters w. This BMA is particularly compelling in Bayesian deep learning, because the posterior over parameters for a modern neural network can represent many complementary solutions to a given problem, corresponding to different settings of parameters (Wilson & Izmailov, 2020). Unfortunately, the BMA integral in Eq. (1) cannot be evaluated in closed form for neural networks, so one must resort to approximate inference. Moreover, approximating Eq. (1) is challenging due to a high dimensional and sophisticated posterior $p(w|\mathcal{D})$. For a detailed discussion of Bayesian deep learning, see e.g. Wilson & Izmailov (2020).

Markov Chain Monte Carlo. The integral in Eq. (1) can be approximated by sampling: $p(y|x,\mathcal{D}) \approx \frac{1}{M}\sum_{i=1}^{M} p(y|x,w_i)$, where $w_i \sim p(w|\mathcal{D})$ are samples drawn from the posterior. MCMC methods construct a Markov chain that, if simulated for long enough, generates approximate samples from the posterior. In this work, we focus on Hamiltonian Monte Carlo (Neal et al., 2011), a method that produces asymptotically exact samples assuming access to the unnormalized posterior density $p(\mathcal{D}|w)p(w)$ and its gradient.

3. Related Work

The bulk of work on Bayesian deep learning has focused on scalable approximate inference methods. These methods include stochastic variational inference (Hoffman et al., 2013; Graves, 2011; Blundell et al., 2015; Kingma et al., 2015; Molchanov et al., 2017; Louizos & Welling, 2017; Khan et al., 2018; Zhang et al., 2018; Wu et al., 2018; Osawa et al., 2019; Dusenberry et al., 2020), dropout (Srivastava et al., 2014; Gal & Ghahramani, 2016; Kendall & Gal, 2017; Gal et al., 2017), the Laplace approximation (MacKay, 1992; Kirkpatrick et al., 2017; Ritter et al., 2018; Li, 2000; Daxberger et al., 2020), expectation propagation (Hernández-Lobato & Adams, 2015), and leveraging the stochastic gradient descent (SGD) trajectory, either for a deterministic approximation, or sampling as in SGLD (Mandt et al., 2017; Maddox et al., 2019; Izmailov et al., 2018; Wilson & Izmailov, 2020). Foong et al. (2019) and Farquhar et al. (2020) additionally consider the role of expressive posterior approximations in variational inference.

While these (and many other) methods often provide improved predictions or uncertainty estimates, to the best of our knowledge *none* of these methods have been directly evaluated on their ability to match the true posterior distribution using practical architectures and datasets. Moreover, many of these methods are often designed with train-time constraints in mind, to take roughly the same amount of compute as regular SGD training. To evaluate approximate inference procedures, and explore fundamental questions in Bayesian deep learning, we attempt to construct a posterior approximation of the highest possible quality, ignoring the practicality of the method.

The Monte Carlo literature for Bayesian neural networks has mainly focused on stochastic gradient-based methods (Welling & Teh, 2011; Ahn et al., 2014; Chen et al., 2014; Ma et al., 2015; Ahn et al., 2012; Ding et al., 2014; Zhang et al., 2020b; Garriga-Alonso & Fortuin, 2021) for computational efficiency reasons. These methods are fundamentally biased: (1) they omit the Metropolis-Hastings (MH) correction, and (2) the noise from subsampling the data perturbs their stationary distribution. In particular, Betancourt (2015) argues that HMC is incompatible with data subsampling. Notably, Zhang et al. (2020a) recently proposed a stochastic gradient MCMC method that is asymptotically exact.

Since the classic work of Neal (1996), there have been a few recent attempts at using full-batch HMC in BNNs (e.g.; Cobb & Jalaian, 2020; Wenzel et al., 2020). These studies tend to use relatively short trajectory lengths (generally not considering a number of leapfrog steps greater than 100), and tend to focus on relatively small datasets and networks. We on the other hand experiment with practical architectures and datasets and use up to 10^5 leapfrog steps per iteration to ensure good mixing.

Our work is aimed at *understanding* the properties of true Bayesian neural networks. In a similar direction, Hron et al. (2020); Novak et al. (2018) explore the infinite-width Gaussian process (GP) (Williams & Rasmussen, 2006) limits of BNNs. In particular, these works propose GP limits that can be used as an approximation of the true BNN posterior.

In another recent work, Wenzel et al. (2020) have explored the effect of the posterior temperature in Bayesian neural networks. We discuss their results in detail in Section 7, and provide our own exploration of the posterior temperature with a different result: we find that BNNs achieve strong performance at temperature 1 and do not require posterior tempering. Moreover, the scope of our paper extends well beyond temperature scaling, revealing for instance that while BNNs can provide strong in-domain generalization, they surprisingly suffer on the covariate shift problems that approximate inference methods are often applied to. We also show that while deep ensembles are often treated as a non-Bayesian alternative, they in fact provide higher fidelity approximations of the Bayesian model average than standard approximate inference procedures, as argued in Wilson & Izmailov (2020). We also explore several other key questions, including prior selection and posterior geometry.

4. HMC for Deep Neural Networks

We use full-batch Hamiltonian Monte Carlo (HMC) to sample from the posterior over the parameters for Bayesian neural networks. In this section, we show how to make HMC effective for modern Bayesian neural networks, discussing important details such as hyper-parameter specification. In the next sections, we use the HMC samples to explore fundamental questions about approximate inference in modern deep learning. We summarize HMC in Appendix Algorithm 1 and Algorithm 2. Intuitively, HMC is simulating the dynamics of a particle sliding on the plot of the density function that we are trying to sample from.[1]

Implementation. To scale HMC to modern neural network architectures and for datasets like CIFAR-10 and IMDB, we parallelize the computation over 512 TPUv3 devices[2] (Jouppi et al., 2020). We execute HMC in a single-program multiple-data (SPMD) configuration, wherein a dataset is sharded evenly over each of the devices and an identical HMC implementation is run on each device. Each device maintains a synchronized copy of the Markov chain state, where the full-batch gradients needed for leapfrog integration are computed using cross-device collectives. We release our JAX (Bradbury et al., 2018) implementation.

HMC hyper-parameters. We set the hyper-parameters of HMC to ensure that the Metropolis-Hastings accept rates are high and the correlation of samples is low. Specifically, we set the *trajectory length* in each HMC iteration to be $\hat{\tau} = \frac{\pi \alpha_{\text{prior}}}{2}$, where α_{prior} is the standard deviation of the prior; when applied to spherical Gaussian distributions, HMC with trajectory length $\hat{\tau}$ will generate exact samples. We set the *step size* to the highest value that still provides high MH accept rates: in general higher step sizes lead to lower accept probabilities. In our main experiments, we run 3 independent HMC chains and combine the samples from all chains. In Appendix B we provide a detailed discussion and extensive ablations of the HMC hyper-parameters, verifying that our choices lead to optimal results in practice.

Neural network architectures. In our evaluation, following Wenzel et al. (2020), we primarily focus on two architectures: ResNet-20-FRN and CNN-LSTM. ResNet-20-FRN is a residual architecture (He et al., 2016) of depth 20 with batch normalization layers (Ioffe & Szegedy, 2015) replaced

[1]For a detailed introduction to HMC please see Neal et al. (2011). See also interactive visualization here: `http://chi-feng.github.io/mcmc-demo/`.

[2]We use other hardware configurations in several experiments. We state the hardware that we used in the corresponding sections.

with filter response normalization (FRN; Singh & Krishnan, 2020). Batch normalization makes the likelihood harder to interpret by creating dependencies between training examples, whereas the outputs of FRN layers are independent across inputs. We use Swish (SiLU) activations (Hendrycks & Gimpel, 2016; Elfwing et al., 2018; Ramachandran et al., 2017) instead of ReLUs to ensure smoothness of the posterior density surface, which we found improves acceptance rates of HMC proposals without hurting the overall performance. The CNN-LSTM is a long short-term memory network (Hochreiter & Schmidhuber, 1997) adapted from Wenzel et al. (2020) without modifications.

Datasets and Data Augmentation. In our main evaluations we use the CIFAR image classification datasets (Krizhevsky et al., 2014) and the IMDB dataset (Maas et al., 2011) for sentiment analysis. We do not use any data augmentation, both because the random augmentations introduce stochasticity into the evaluation of the posterior log-density and its gradient, and because the expected randomly perturbed log-likelihood does not have a clean interpretation as a valid likelihood function (Wenzel et al., 2020).

5. How Well does HMC Mix?

The primary goal of our paper is to construct accurate samples from the posterior, and use them to understand the properties of Bayesian neural networks better. In this section we consider several diagnostics to evaluate whether our HMC sampler has converged, and discuss their implications to the posterior geometry. We consider mixing in both *weight space* and *function space*. A distribution over weights w combined with a neural network architecture $f(x, w)$ induces a distribution over functions $f(x)$. Ultimately, since we are using functions to make predictions, we care mostly about mixing in function space.

Summary: HMC is able to mix surprisingly well in function space, and better than in parameter space. Geometrically, HMC is able to explore connected basins of the posterior with high functional diversity.

5.1. $\hat{R}$ Diagnostics

We apply the Gelman et al. (1992) "$\hat{R}$" potential-scale-reduction diagnostic to our HMC runs. Given two or more chains, $\hat{R}$ estimates the ratio between the between-chain variance (i.e., the variance estimated by pooling samples from all chains) and the average within-chain variance (i.e., the variances estimated from each chain independently). Intuitively, if the chains are stuck in isolated modes, then combining samples from multiple chains will yield greater diversity than taking samples from one chain. For the precise mathematical definition of $\hat{R}$, please see the Appendix D.

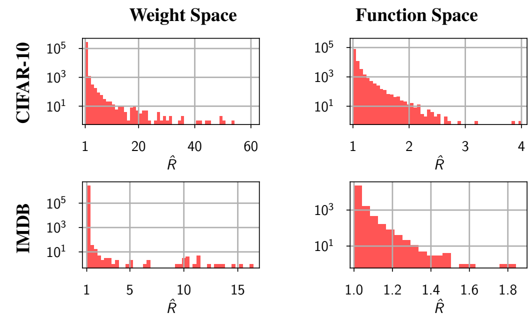

Figure 1. **Log-scale histograms of $\hat{R}$ convergence diagnostics.** Function-space $\hat{R}$s are computed on the test-set softmax predictions of the classifiers and weight-space $\hat{R}$s are computed on the raw weights. About 91% of CIFAR-10 and 98% of IMDB posterior-predictive probabilities get an $\hat{R}$ less than 1.1. Most weight-space $\hat{R}$ values are quite small, but enough parameters have very large $\hat{R}$s to make it clear that the chains are sampling from different distributions in weight space.

We compute $\hat{R}$ using TensorFlow Probability's implementation[3] (Lao et al., 2020) for both the weights and the test-set softmax predictions on CIFAR-10 with ResNet-20-FRN and on IMDB with CNN-LSTM. We report the results in Figure 1. On both IMDB and CIFAR, the bulk of the function-space $\hat{R}$ values is concentrated near 1, meaning intuitively that a single chain can capture the diversity of predictions on most of the test data points nearly as well as multiple chains. The mixing is especially good on the IMDB dataset, where only 2% of inputs correspond to $\hat{R}$ larger than 1.1. In Appendix C we apply HMC to a synthetic regression problem and show that HMC can indeed mix in the prediction space: different HMC chains provide very similar predictions.

In weight space, although most parameters show no evidence of poor mixing, some have very large $\hat{R}$s, indicating that there are directions in which the chains fail to mix.

Implications for the Posterior Geometry. The fact that a single HMC chain is able to mix well in function (prediction) space suggests that the posterior contains connected regions which correspond to high functional diversity. Indeed, a single HMC chain is extremely unlikely to jump between isolated modes, but appears able to produce samples with diverse predictions. Prior work on *mode connectivity* (Garipov et al., 2018; Draxler et al., 2018) has shown that there exist paths of high density connecting different modes of the posterior. Our observations suggest a stronger version of mode connectivity: not only do mode-connecting paths exist between functionally diverse modes, but also at least some of these paths can be leveraged by Monte Carlo methods to efficiently explore the posterior. In Appendix E we provide visualizations of the posterior density surface to provide further intuition.

[3] tfp.mcmc.potential_scale_reduction

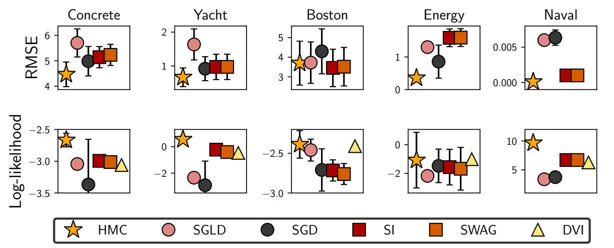

Figure 2. **UCI regression datasets.** Performance of Hamiltonian Monte Carlo (HMC), stochastic gradient Langevin dynamics (SGLD), stochastic gradient descent (SGD), subspace inference (SI) (Izmailov et al., 2019), SWAG (Maddox et al., 2019) and deterministic variational inference (DVI; Wu et al., 2018). We use a fully-connected architecture with a single hidden layer of 50 neurons. The results reported for each method are mean and standard deviation computed over 20 random train-test splits. For SI, SWAG and DVI we report the results presented in Izmailov et al. (2019). **Top:** test root-mean-squared error. **Bottom:** test log-likelihood. HMC performs on par with or better than all other baselines in each experiment, often providing a significant improvement.

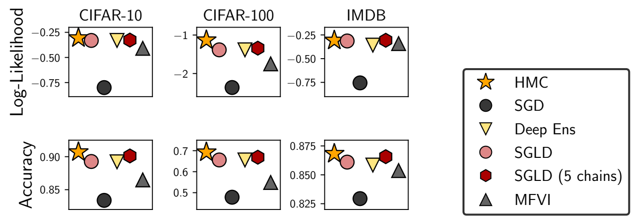

Figure 3. **Image and text classification.** Performance of Hamiltonian Monte Carlo (HMC), stochastic gradient Langevin dynamics (SGLD) with 1 and 5 chains, mean field variational inference (MFVI), stochastic gradient descent (SGD), and deep ensembles. We use ResNet-20-FRN on CIFAR datasets, and CNN-LSTM on IMDB. Bayesian neural networks via HMC outperform all baselines on all datasets. For full results, see Appendix G.

Does HMC converge? In Appendix F we study the convergence of the accuracy and log-likelihoods for individual HMC samples and the BMA ensembles. Based on the results of this ablation, we set the number of burn-in iterations to 50 to ensure that the HMC chains converge before we begin collecting the samples.

6. Evaluating Bayesian Neural Networks

Now that we have a procedure for effective HMC sampling, we are primed to explore exciting questions about the fundamental behaviour of Bayesian neural networks, such as the role of tempering, the prior over parameters, generalization performance, and robustness to covariate shift. In this section we evaluate Bayesian neural networks in various problems using our implementation of HMC. Throughout the experiments, we use posterior temperature $T = 1$.

We emphasize that the main goal of our paper and this section in particular is to *understand* the behaviour of true BNNs using HMC as a precise tool, and *not* to argue for HMC as a practical method for Bayesian deep learning.

Summary: Bayesian neural networks achieve strong results, outperforming even large deep ensembles in a range of evaluations. Surprisingly, however, BNNs are *less* robust to distribution shift than conventionally-trained models.

6.1. Regression on UCI Datasets

Bayesian deep learning methods are often evaluated on small-scale regression problems using fully connected networks (e.g., Wu et al., 2018; Izmailov et al., 2019; Maddox et al., 2019). Following these works, we evaluate BNNs using HMC on five UCI regression datasets: *Concrete*, *Yacht*, *Boston*, *Energy* and *Naval*. For each of these datasets, we construct 20 random 90-to-10 train-test splits and report the mean and standard deviation of performance over the splits. We use a fully connected neural network with a single hidden layer of size 50 and 2 outputs representing the predictive mean and standard deviation. For HMC we used a single chain with 10 burn-in iterations and 90 iterations of sampling. For more details, please see Appendix A.

We report the results in Figure 2. HMC typically outperforms all the baselines, often by a significant margin, both in test RMSE and log-likelihood. On the *Boston* dataset, HMC achieves a slightly higher average RMSE compared to subspace inference and SWAG (Izmailov et al., 2019; Maddox et al., 2019) but outperforms both these methods significantly in terms of log-likelihood.

6.2. Image Classification on CIFAR

Next, we evaluate Bayesian neural networks using HMC on image classification problems. We use the ResNet-20-FRN architecture on CIFAR-10 and CIFAR-100. We picked a random subset of 40960 of the 50000 images for each of the datasets to be able to evenly shard the data across the TPU devices; we use the same subset for both HMC and the baselines. We run 3 HMC chains using step size 10^{-5} and a prior variance of $1/5$, resulting in 70,248 leapfrog steps per sample. In each chain we discard the first 50 samples as burn-in, and then draw 240 samples (720 in total for 3 chains)[4]. For SGLD, we use a single chain with 1000 burn-in epochs and 9000 epochs of sampling producing 900 samples; we also report the performance of an ensemble of 5 independent SGLD chains. Next, we report the performance

[4]In total, on CIFAR-10 our HMC run requires as many computations as *over 60 million epochs* of standard SGD training.

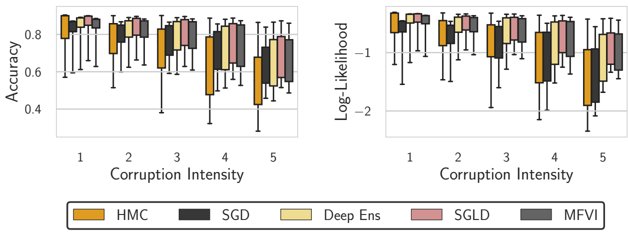

Figure 4. **OOD Robustness.** Accuracy and log-likelihood of HMC, SGD, deep ensembles, SGLD and MFVI under covariate shift, where the CIFAR-10 test set is corrupted in 16 different ways at intensities on the scale of 1 to 5. We use the ResNet-20-FRN architecture. Boxes capture the quartiles of performance over corruption types, with the whiskers indicating the minimum and maximum. HMC is surprisingly the worst of the considered methods: even a single SGD solution provides better OOD robustness.

of a mean field variational inference (MFVI) solution; we initialize the mean of MFVI with a solution pre-trained with SGD and use an ensemble of 50 samples from the VI posterior at evaluation. Finally, we report the performance of a single SGD solution and a deep ensemble of 50 models. For more details, see Appendix A.

We report the results in Figure 3 and Appendix G.. Bayesian neural networks outperform all baselines in terms of accuracy and log-likelihood on both datasets. In terms of ECE, SGD provides the worst results across the board, and the rest of the methods are competitive.

BNNs under distribution shift. Bayesian methods are often specifically applied to covariate shift problems (e.g., Ovadia et al., 2019; Wilson & Izmailov, 2020; Dusenberry et al., 2020). We evaluate the the performance of HMC and baselines on the CIFAR-10-C dataset (Hendrycks & Dietterich, 2019), which applies a set of corruptions to CIFAR-10 with varying intensities. Following the setup in Ovadia et al. (2019), we use the same 16 corruptions, evaluating the performance at all intensities. We report the results in Figure 4. Surprisingly, we find that Deep Ensembles and SGLD are consistently more robust to distribution shift than HMC-based BNNs. For high corruption intensities, even a single SGD model outperforms the HMC ensemble. We note that while BNNs are not robust to covariate shift, they can *detect* it (see Appendix I).

In Appendix H we provide further exploration of this effect, where we see HMC samples are significantly less robust to many types of noise compared to conventionally-trained SGD models. We see in the Appendix that the performance of HMC-based BNNs under data corruption can be significantly improved by using posterior tempering.

Inspired by our findings, Izmailov et al. (2021) provide a detailed explanation for why high-fidelity Bayesian model averaging can fail under covariate shift.

6.3. Language Classification on IMDB

We use a CNN-LSTM architecture on the IMDB binary text classification dataset. In Figure 3 we report the results for HMC and the same baselines as in Section 6.2. We use HMC with a step size of 10^{-5} and a prior variance of $1/40$, resulting in 24,836 leapfrog steps per sample. We run 3 chains, burning-in for 50 samples, and drawing 400 samples per chain (1,200 total). For more details, please see Appendix A and Appendix G.. Analogously to the image classification experiments, HMC outperforms the baselines on accuracy and log-likelihood.

7. Do We Need Cold Posteriors?

Multiple works have considered tempering the posterior in Bayesian neural networks (e.g. Wenzel et al., 2020; Wilson & Izmailov, 2020; Zhang et al., 2020b; Aitchison, 2020). Specifically, we can consider a distribution $p_T(w|\mathcal{D}) \propto \big(p(\mathcal{D}|w) \cdot p(w)\big)^{1/T}$, where w are the parameters of the network, $\mathcal{D}$ is the training dataset, $p(\mathcal{D}|w)$ is the likelihood of $\mathcal{D}$ for the network with parameters w and T is the *temperature*. Note that at temperature $T = 1$, p_T corresponds to the standard Bayesian posterior over the parameters of the network. Temperatures $T < 1$ correspond to *cold posteriors*, distributions that are sharper than the Bayesian posterior. Similarly, temperatures $T > 1$ correspond to *warm posteriors* which are softer than the Bayesian posterior. See Appendix Figure 9(e) for a visualization.

Wenzel et al. (2020) argue that Bayesian neural networks require a cold posterior, and the performance at temperature $T = 1$ is inferior to even a single model trained with SGD. The authors refer to this phenomenon as *the cold posteriors effect*. However, our results are different:

Summary: We show that cold posteriors are not needed to obtain near-optimal performance with Bayesian neural networks and may even hurt performance. We show that the cold posterior effect is largely an artifact of data augmentation.

7.1. Testing the Cold Posteriors Effect

Wenzel et al. (2020) demonstrate the cold posteriors with two main experiments: ResNet-20 on CIFAR-10 and CNN-LSTM on IMDB. In these experiments the authors show poor performance at temperature $T = 1$, with strong benefits from decreasing the temperature. However, for the CIFAR-10 experiment, it is apparent (Wenzel et al., 2020, Appendix K, Figure 28) that the results at $T = 1$ are near-optimal for the ResNet on CIFAR-10 if data augmentation is turned off and batch normalization is replaced with filter response normalization, which is in fact necessary for a

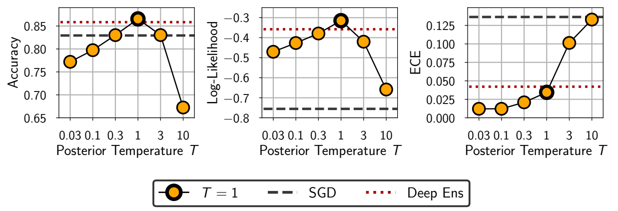

Figure 5. **Effect of posterior temperature.** Log-likelihood, accuracy and expected calibration error using the CNN-LSTM on the IMDB dataset as a function of posterior temperature T. For both the likelihood and accuracy $T = 1$ provides optimal performance, while for the ECE the colder posteriors provide a slight improvement. For all three metrics, the posterior at $T = 1$ outperforms the SGD baseline as well as a deep ensemble of 10 models.

clear Bayesian interpretation of the inference procedure.

Furthermore, in Section 6, we show that Bayesian neural networks can achieve performance superior to SGD and even deep ensembles at temperature $T = 1$, in particular using the same ResNet-20-FRN model on CIFAR-10 and CNN-LSTM model on IMDB used by Wenzel et al. (2020).

To further understand the effect of posterior temperature T, we compare the performance of the CNN-LSTM model at different T using our HMC sampler. In all runs we used a fixed prior variance $\alpha^2 = \frac{1}{40}$. We report the results in Figure 5. The performance of the BNN at $T = 1$ is better than the SGD baseline as well as a deep ensemble of 50 independent models. Moreover, the performance at $T = 1$ is better compared to all other temperatures we tested in terms of both test accuracy and log-likelihood.

We also note that while posterior tempering does not seem necessary for good predictive performance with BNNs, it may be helpful under distribution shift (Appendix H). Wilson & Izmailov (2020) additionally argue that tempering may be a reasonable procedure in general, and is not necessarily at odds with Bayesian principles.

Role of data augmentation. Our results are in contrast with Wenzel et al. (2020), who argue that cold posteriors are needed for good performance with BNNs, and claim that data augmentation “likely does not account for the cold posterior effect”. In Appendix J we provide an additional study of what may have caused the poor performance of BNNs in Wenzel et al. (2020), using the code for inference provided by Wenzel et al. (2020). We identify data augmentation as the key factor responsible for the cold posterior effect, and also show that batch normalization does not significantly influence this effect: when the data augmentation is turned off, we do not observe the cold posteriors effect[5]. Data augmentation cannot be naively incorporated in the Bayesian neural network model (see the discussion in Appendix K of Wenzel et al. (2020)), and arguably it may be reasonable to decrease the temperature when using data augmentation: intuitively, data augmentation increases the amount of data observed by the model, and should lead to higher posterior contraction. We leave incorporating data augmentation in our HMC evaluation framework as future work.

[5]In a concurrent work, Fortuin et al. (2021) also note that data augmentation strengthens the cold posteriors effect.

Prior	Gaussian	MoG	Logistic
Accuracy	0.866	0.863	**0.869**
Log Likelihood	-0.311	-0.317	**-0.304**

Table 1. **Effect of prior.** BMA accuracy and log-likelihood under different prior families using CNN-LSTM on IMDB. We produce 80 samples from a single HMC chain for each of the priors. The heavier-tailed logistic prior provides slightly better performance compared to the Gaussian and mixture of Gaussians (MoG) priors.

8. What is the Effect of Priors in Bayesian Neural Networks?

Bayesian deep learning is often criticized for the lack of intuitive priors over the parameters. For example, Wenzel et al. (2020) hypothesize that the popular Gaussian priors of the form $\mathcal{N}(0, \alpha^2 I)$ are inadequate and lead to poor performance. Tran et al. (2020) propose a new prior for BNNs inspired by GPs (Williams & Rasmussen, 2006) based on this hypothesis. In concurrent work, Fortuin et al. (2021) also explore several alternatives to standard Gaussian priors inspired by the cold posteriors effect. Wilson & Izmailov (2020), on the other hand, argue that vague Gaussian priors in the parameter space induce useful function-space priors.

In Section 6 we have shown that Bayesian neural networks can achieve strong performance with vague Gaussian priors. In this section, we explore the sensitivity of BNNs to the choice of the prior scale as well as several alternative prior families, as a step towards a better understanding of the role of the prior in BNNs.

> **Summary:** High-variance Gaussian priors over parameters of BNNs lead to strong performance. The results are robust with respect to the prior scale. Mixture of Gaussian and logistic priors over parameters are similar in performance to Gaussian priors. These results highlight the relative importance of architecture over parameter priors in specifying a useful prior over functions.

In Table 1, we report BMA accuracy and log-likelihood for two non-Gaussian priors on the IMDB dataset: *logistic* and *mixture of Gaussians* (MoG). For the MoG prior we use a mixture of two Gaussians centered at 0, one with variance

	Metric	HMC (reference)	SGD	Deep Ens	MFVI	SGLD	SGHMC	SGHMC CLR	SGHMC CLR-Prec
						SGMCMC			
CIFAR-10	Accuracy	89.64 ± 0.25	83.44 ± 1.14	88.49 ± 0.10	86.45 ± 0.27	89.32 ± 0.23	89.38 ± 0.32	**89.63** $\mathbf{\pm 0.37}$	87.46 ± 0.21
	Agreement	94.01 ± 0.25	85.48 ± 1.00	91.52 ± 0.06	88.75 ± 0.24	91.54 ± 0.15	91.98 ± 0.35	**92.67** $\mathbf{\pm 0.52}$	90.96 ± 0.24
	Total Var	0.074 ± 0.003	0.190 ± 0.005	0.115 ± 0.000	0.136 ± 0.000	0.110 ± 0.001	0.109 ± 0.001	**0.099** $\mathbf{\pm 0.006}$	0.111 ± 0.002
CIFAR-10-C	Accuracy	70.91 ± 0.93	71.04 ± 1.80	76.99 ± 0.39	75.40 ± 0.34	**78.80** $\mathbf{\pm 0.17}$	78.20 ± 0.25	76.43 ± 0.39	73.42 ± 0.39
	Agreement	86.00 ± 0.44	72.01 ± 0.82	79.29 ± 0.18	75.47 ± 0.27	77.99 ± 0.22	78.98 ± 0.22	**80.93** $\mathbf{\pm 0.73}$	79.65 ± 0.35
	Total Var	0.133 ± 0.004	0.334 ± 0.007	0.220 ± 0.003	0.245 ± 0.002	0.214 ± 0.002	0.203 ± 0.002	**0.194** $\mathbf{\pm 0.010}$	0.205 ± 0.005

Table 2. **Evaluation of cheaper alternatives to HMC.** Agreement and total variation between predictive distributions of HMC and approximate inference methods: deep ensembles, mean field variational inference (MFVI), and stochastic gradient Monte Carlo (SGMCMC) variations. For all methods we use ResNet-20-FRN trained on CIFAR-10 and evaluate predictions on the CIFAR-10 and CIFAR-10-C test sets. For CIFAR-10-C we report the average results across all corruptions and corruption intensities. We additionally report the results for HMC for reference: we compute the agreement and total variation between one of the chains and the ensemble of the other two chains. For each method we report the mean and standard deviation of the results over three independent runs. MFVI provides the worst approximation of the predictive distribution. Deep ensembles despite often being considered non-Bayesian, significantly outperform MFVI. SG-MCMC methods provide the best results with SGHMC-CLR showing the best overall performance.

$\frac{1}{40}$ and the other with variance $\frac{1}{160}$. We pick a logistic prior with a variance of $\frac{1}{40}$. We additionally provide the results for a Gaussian prior with variance $\frac{1}{40}$. We approximate the BMA using 80 samples from a single HMC chain for each of the priors. We find that the heavier-tailed logistic prior performs slightly better than the Gaussian and MoG.

In Appendix K, we additionally show that performance of BNNs with Gaussian priors $\mathcal{N}(0, \alpha I)$ is fairly robust to the choice of α with vague priors avhieving the best results.

Importance of Architecture in Prior Specification. We often think of the prior narrowly in terms of a distribution over parameters $p(w)$. But the prior that matters is the prior over functions $p(f(x))$ that is induced when a prior over parameters $p(w)$ is combined with the functional form of a neural network $f(x, w)$. All of the results in this section point to the relative importance of the architecture in defining the prior over functions, compared to the prior over parameters. A vague prior over parameters is not necessarily vague in function-space. Moreover, while the details of the prior distribution over parameters $p(w)$ have only a minor effect on performance, the choice of architecture certainly has a major effect on performance.

9. Do Scalable BDL Methods and HMC Make Similar Predictions?

While HMC shows strong performance in our evaluation in Section 6, in most realistic BNN settings it is an impractical method. However, HMC can be used as a *reference* to evaluate and calibrate more practical alternatives. In this section, we evaluate the *fidelity* of SGMCMC, variational methods, and deep ensembles in representing the predictive distribution (BMA) given by our HMC reference.

Summary: While SGMCMC and Deep Ensembles can provide good generalization accuracy and calibration, their predictive distributions differ from HMC. Deep ensembles are similarly close to the HMC predictive distribution as SGLD, and closer than standard variational inference.

We consider two primary metrics: *agreement* and *total variation*. We define the agreement between the predictive distributions $\hat{p}$ of HMC and p of another method as the fraction of the test data points for which the top-1 predictions of $\hat{p}$ and p are the same: $\frac{1}{n}\sum_{i=1}^{n} I[\arg\max_j \hat{p}(y=j|x_i) = \arg\max_j p(y=j|x_i)]$, where $I[\cdot]$ is the indicator function and n is the number of test data points x_i. We define the total variation metric between $\hat{p}$ and p as the total variation distance between the predictive distributions averaged over the test data points: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\sum_j \left|\hat{p}(y=j|x_i) - p(y=j|x_i)\right|$. The agreement (higher is better) captures how well a method is able to capture the top-1 predictions of HMC, while the total variation (lower is better) compares the predictive probabilities for each of the classes. To provide an additional comparison of the predictive distributions between HMC and other methods, in Appendix L we study the distribution of predictive entropies and the calibration curves.

In Table 2 we report the agreement and total variation metrics as well as the predictive accuracy on the CIFAR-10 and CIFAR-10-C test sets for a deep ensemble of 10 models and several SGLD variations: standard SGLD (Welling & Teh, 2011), SGLD with momentum (SGHMC) (Chen et al., 2014), SGLD with momentum and a cyclical learning rate schedule (SGHMC-CLR) (Zhang et al., 2020b) and SGLD with momentum, cyclical learning rate schedule and a preconditioner (SGHMC-CLR-Prec) (Wenzel et al., 2020). All methods were trained on CIFAR-10. For more details, please see Appendix A.

Overall, the absolute value of agreement achieved by all methods is fairly low on CIFAR-10 and especially on CIFAR-10-C. More advanced SGHMC-CLR and SGHMC-CLR-Prec methods provide a better fit of the HMC predictive distribution while not necessarily improving the accuracy. Notably, these methods are also *less* robust to the data corruptions in CIFAR-10-C, again suggesting that higher fidelity representations of the predictive distribution can lead to decreased robustness, as we found in section 6.2.

Deep ensembles provide a reasonable approximation to the HMC predictive distribution, outperforming both SGLD and SGHMC in terms of total variation on CIFAR-10 and in terms of agreement on CIFAR-10-C. These results support the argument that deep ensembles, while not typically characterized as a Bayesian method, provide a *higher fidelity* approximation to a Bayesian model average than methods that are conventionally accepted as Bayesian inference procedures in modern deep learning (Wilson & Izmailov, 2020).

In Appendix H we explore the performance of HMC, SGD, deep ensembles, and SGMCMC variations under different corruptions individually. Interestingly, the behavior of SGLD and SGHMC-CLR-Prec appears more similar to that of deep ensembles than that of HMC. So, while both SGMCMC and deep ensembles are very compelling practically, they provide relatively distinct predictive distributions from HMC. Mean-field VI methods are particularly far from the HMC predictive distribution. Thus, we should be very careful when making judgements about *true* Bayesian neural networks based on the SGMCMC or MFVI performance.

10. Discussion

Despite the rapidly increasing popularity of approximate Bayesian inference in modern deep learning, little is known about the behaviour of truly Bayesian neural networks. To the best of our knowledge, our work provides the first realistic evaluation of BNNs with precise and exhaustive posterior sampling. We establish several properties of Bayesian neural networks, including good generalization performance, lack of a cold posterior effect, and a lack of robustness to covariate shift. We hope that our observations and the tools that we develop will facilitate fundamental progress in understanding the behaviour of Bayesian neural networks.

Should we use Bayesian neural networks? On most of the problems considered in this work, the best results both in terms of uncertainty calibration and predictive accuracy were achieved by Bayesian neural networks. We believe that our results provide motivation to use BNNs with accurate posterior approximation in practical applications and hope that our work will inspire the community to produce new accurate and scalable approximate inference methods for Bayesian deep learning.

Challenging conventional wisdom. A conventional wisdom has emerged that deep ensembles are a non-Bayesian alternative to variational methods, that standard priors for neural networks are poor, and that cold posteriors are a problematic result for Bayesian deep learning. Our results highlight that one should take care in uncritically repeating such claims. In fact, deep ensembles appear to provide a higher fidelity representation of the Bayesian predictive distribution than widely accepted approaches to approximate Bayesian inference. If anything, the takeaway from the relatively good performance of deep ensembles is that we would benefit from approximate inference being *closer* to the Bayesian ideal! Moreover, the details over the priors in weight space can have a relatively minor effect on performance, and there is no strong evidence that standard Gaussian priors are particularly bad. In fact, there are many reasons to believe these priors have useful properties (Wilson & Izmailov, 2020). Similarly, on close inspection, we found no evidence for a general cold posterior effect, which we identify as largely an artifact of data augmentation. Although we see here that tempering does not in fact seem to be required, as argued in Wilson & Izmailov (2020) tempering is also not necessarily unreasonable or even divergent from Bayesian principles. Even the results we found that are less favourable to Bayesian deep learning are contrary to the current orthodoxy. Indeed, higher fidelity Bayesian inference surprisingly appears to suffer more greatly from covariate shift, despite the popularity of approximate Bayesian inference procedures in this setting.

Acknowledgements

The authors would like to thank many people at Google and NYU for helpful discussions, especially Rodolphe Jenatton, Rif A. Saurous, Jasper Snoek, Pavel Sountsov, Florian Wenzel, Marc Finzi, Wesley Maddox, Greg Benton, and the entire TensorFlow Probability team. This research is supported with Cloud TPUs from Google's TPU Research Cloud (TRC), and by an Amazon Research Award, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science.

References

Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

Ahn, S., Shahbaba, B., and Welling, M. Distributed stochastic gradient mcmc. In *International conference on machine learning*, pp. 1044–1052. PMLR, 2014.

Aitchison, L. A statistical theory of cold posteriors in deep neural networks. *arXiv preprint arXiv:2008.05912*, 2020.

Betancourt, M. The fundamental incompatibility of hamiltonian monte carlo and data subsampling. *arXiv preprint arXiv:1502.01510*, 2015.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Cobb, A. D. and Jalaian, B. Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting. *arXiv preprint arXiv:2010.06772*, 2020.

Cranmer, M., Tamayo, D., Rein, H., Battaglia, P., Hadden, S., Armitage, P., Ho, S., and Spergel, D. N. A bayesian neural network predicts the dissolution of compact planetary systems. 2021.

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

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R., and Neven, H. Bayesian sampling using stochastic gradient thermostats. 2014.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially no barriers in neural network energy landscape. *arXiv preprint arXiv:1803.00885*, 2018.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Farquhar, S., Smith, L., and Gal, Y. Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. *arXiv preprint arXiv:2002.03704*, 2020.

Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G., Kenton, Z., Smith, L., Alizadeh, M., de Kroon, A., and Gal, Y. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*, 2019.

Foong, A. Y., Burt, D. R., Li, Y., and Turner, R. E. On the expressiveness of approximate inference in bayesian neural networks. *arXiv preprint arXiv:1909.00719*, 2019.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

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

Gal, Y. *Uncertainty in deep learning*. PhD thesis, PhD thesis, University of Cambridge, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gal, Y., Hron, J., and Kendall, A. Concrete dropout. *arXiv preprint arXiv:1705.07832*, 2017.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Neural Information Processing Systems*, 2018.

Garriga-Alonso, A. and Fortuin, V. Exact langevin dynamics with stochastic gradients. *arXiv preprint arXiv:2102.01691*, 2021.

Gelman, A., Rubin, D. B., et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

Geneva, N. and Zabaras, N. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, 2020.

Graves, A. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356. Citeseer, 2011.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

Hron, J., Bahri, Y., Novak, R., Pennington, J., and Sohl-Dickstein, J. Exact posterior distributions of wide bayesian neural networks. *arXiv preprint arXiv:2006.10541*, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence (UAI)*, 2018.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. Subspace inference for Bayesian deep learning. *Uncertainty in Artificial Intelligence*, 2019.

Izmailov, P., Nicholson, P., Lotfi, S., and Wilson, A. Dangers of Bayesian model averaging under covariate shift. *To appear.*, 2021.

Jouppi, N. P., Yoon, D. H., Kurian, G., Li, S., Patil, N., Laudon, J., Young, C., and Patterson, D. A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, 63(7):67–78, 2020.

Kendall, A. and Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable Bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

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

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A., Nair, V., and Hinton, G. The CIFAR-10 dataset. 2014. http://www.cs.toronto.edu/kriz/cifar.html.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D., and Dillon, J. V. tfp. mcmc: Modern markov chain monte carlo tools built for modern hardware. *arXiv preprint arXiv:2002.01184*, 2020.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

Li, D. X. On default correlation: A copula function approach. *Journal of Fixed Income*, 9(4):43–54, 2000.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Liu, X., Xue, W., Xiao, L., and Zhang, B. Pbodl: Parallel bayesian online deep learning for click-through rate prediction in tencent advertising system. *arXiv preprint arXiv:1707.00802*, 2017.

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

Ma, Y.-A., Chen, T., and Fox, E. B. A complete recipe for stochastic gradient mcmc. *arXiv preprint arXiv:1506.04696*, 2015.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Association for Computational Linguistics*, 2011.

MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

MacKay, D. J. Probable networks and plausible predictions?a review of practical Bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505, 1995.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate Bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

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

Neal, R. *Bayesian Learning for Neural Networks*. Springer Verlag, 1996. ISBN 0387947248.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. Practical deep learning with bayesian principles. *arXiv preprint arXiv:1906.02506*, 2019.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Ritter, H., Botev, A., and Barber, D. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Singh, S. and Krishnan, S. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11237–11246, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. All you need is a good functional prior for bayesian deep learning. *arXiv preprint arXiv:2011.12829*, 2020.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

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

Williams, C. K. and Rasmussen, C. E. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

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

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., and Gaunt, A. L. Deterministic variational inference for robust Bayesian neural networks. *arXiv preprint arXiv:1810.03958*, 2018.

Yao, J., Pan, W., Ghosh, S., and Doshi-Velez, F. Quality of uncertainty quantification for bayesian neural network inference. *arXiv preprint arXiv:1906.09686*, 2019.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pp. 5852–5861, 2018.

Zhang, R., Cooper, A. F., and De Sa, C. Amagold: Amortized metropolis adjustment for efficient stochastic gradient mcmc. In *International Conference on Artificial Intelligence and Statistics*, pp. 2142–2152. PMLR, 2020a.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020b.