Approximation Inference Techniques for Bayesian Neural Networks
COM3023, Continuous Assessment

Student Number: 720011935

March 21, 2025

Abstract

This report investigates and compares two prominent posterior approximation methods, Variational Inference (VI) and Markov Chain Monte Carlo (MCMC), within the context of Bayesian Neural Networks (BNNs). After establishing the theoretical foundations of BNNs, VI, and MCMC, a BNN is implemented to perform binary classification on flood risk data. The performance and efficiency of both approximation methods are evaluated across various network configurations, including deeper architectures and models with poor prior specifications. Results show that while MCMC, particularly the No-U-Turn Sampler (NUTS), yields more accurate and uncertainty-aware posteriors, it does so at a significantly higher computational cost. Conversely, VI, especially when paired with minibatching, demonstrates far superior efficiency, albeit at the expense of reduced posterior expressiveness and uncertainty quantification. The findings highlight the trade-offs between computational cost and inference quality, emphasizing that the optimal method depends on the application's priorities.

I certify that all material in this document which is not my own work has been identified.

1 Introduction

This report aims to explore and understand the relative strengths and weaknesses of Variational Inference (VI) and Markov Chain Monte Carlo (MCMC) posterior approximation methods for Bayesian Neural Networks (BNNs), particularly the results of the networks they produce and relative efficiencies. This is crucial if the advantages of BNNs are to be realised. First, a theoretical explanation of Artificial Nerual Networks (ANNs), BNNs, VI, and MCMC is provided. A simple BNN is then constructed to perform binary classification on flood risk data. VI and MCMC are used on different permutations of the BNN to compare and evaluate both methods. Finally, the findings are summarised and future research areas discussed.

The popularity of Machine Learning (ML) has increased rapidly due to the success of ANNs [18]. While their inception can be traced back to the introduction of the perceptron in 1958 [25], it has required more recent advances, including the backpropagation algorithm [26] and convolutional neural networks [16], for ANNs to achieve their dominance. ANNs possess several advantages over more traditional ML methods [30]. Firstly, they are incredibly flexible with regards to structure and problem type. Furthermore, they can implicitly find complex non-linear relationships between variables, removing the need for human feature crafting and allowing exploitation of deep connections in data. This second point means that, given sufficient data, neural networks usually outperform other ML methods [6].

Traditional machine learning methods typically begin with an initial set of model parameters—often randomly assigned—and refine them through iterative updates using optimization algorithms like gradient descent. The objective is to identify a parameter set that minimizes a loss function, which maximizes the likelihood that model correctly predicts the labels of the observed data. In contrast, Bayesian learning introduces prior beliefs about the model parameters and data, treating the parameters not as fixed values but as probability distributions. Rather than searching for a single optimal parameter set, Bayesian approaches compute a posterior distribution over all plausible parameters conditioned on the observed data. Predictions are then generated by integrating over this posterior, capturing uncertainty and variability in a way that deterministic methods cannot.

Approaching ANNs from a Bayesian perspective results in a BNN, which builds on the strengths of standard ANNs with further advantages. While typical ANNs struggle from a "black-box" nature with little context provided with an output, BNNs are able to capture and quantify uncertainty in their outputs as they incorporate multiple parameter sets [21]. BNNs are also able to separate aleatoric and epistemic uncertainties [15], provide a way to easily incorporate human knowledge [8], and, particularly when given an informative prior, perform well on small datasets which typical ANNs would struggle with [20].

Given these comparative advantages, it may appear surprising that BNNs are relatively obscure compared to standard frequentist ANNs. Their unpopularity can largely be attributed to the challenge of calculating the posterior distribution, which is usually an intractable problem [24]. Instead, the posterior distribution must be approximated, typically with either VI or MCMC techniques. Both methods have comparative advantages and disadvantages which are crucial to explore and understand if the strengths of BNNs are to be realised.

2 Background

2.1 Bayesian Neural Networks

An Artificial Neural Network (ANN) is a function $f : \mathbb{R}^n \to \mathbb{R}^m$ that maps an input vector $x \in \mathbb{R}^n$ to an output vector $y \in \mathbb{R}^m$ through a series of layers composed of affine transformations followed by non-linear activation functions [7]. A feed-forward network with L layers can be expressed as:

$$f(x;\theta) = f_L(f_{L-1}(...f_2(f_1(x))...)) \tag{1}$$

where θ denotes the trainable parameters—weights and biases—of the network. Each layer l computes:

$$f_l(h_{l-1}) = \sigma(W_l h_{l-1} + b_l) \tag{2}$$

with W_l and b_l as the layer's weight matrix and bias vector, and σ as a non-linear activation function. The network is trained by minimizing a loss function:

$$\mathcal{L}(\theta) = \sum_{i=0}^{n} \ell(f(x_i;\theta), y_i) \tag{3}$$

Bayesian Neural Networks (BNNs) extend this framework by treating the parameters θ as random variables with prior distributions. Instead of optimizing for a single best parameter set, BNNs use Bayes' theorem to compute a posterior distribution over parameters given observed data $\mathcal{D}$ [22]:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{4}$$

where $p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(y_i|x_i,\theta)$ is the likelihood of the data, $p(\theta)$ is the prior over weights and biases, and $p(\mathcal{D})$ is the model evidence. Predictions are then made by integrating over all possible parameter configurations:

$$p(y|x,\mathcal{D}) = \int p(y|x,\theta)p(\theta|\mathcal{D})d\theta \tag{5}$$

This approach enables BNNs to quantify uncertainty, incorporate prior knowledge, and perform more robustly on small datasets [6]. However, computing the posterior and marginal likelihood involves intractable integrals, particularly in high-dimensional parameter spaces:

$$p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta \tag{6}$$

Thus, we rely on approximation inference to estimate the posterior to implement BNNs.

2.2 Variational Inference

One common method of approximation inference is variational inference (VI) [11]. This involves approximating $p(\theta|\mathcal{D})$ with a simpler variational distribution, $q(\theta|\lambda)$, parametrised by λ, by minimising the Kullback-Leibler (KL):

$$D_{KL}(q(\theta|\lambda)||p(\theta|\mathcal{D})) \tag{7}$$

Since, again, $p(\theta|\mathcal{D})$ is unknown, KL divergence is minimised indirectly by minimising the Evidence Lower Bound (ELBO) which is independent of $p(\theta|\mathcal{D})$:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\theta|\lambda)}[\log p(\mathcal{D},\theta) - \log q(\theta|\lambda)] \tag{8}$$

where $p(\mathcal{D},\theta) = p(\mathcal{D}|\theta)p(\theta)$. Gradient descent is then performed using ELBO to optimise λ.

Specifically, this report uses a VI variant called Automatic Differentiation Variational Inference (ADVI), chosen for its strong performance with high-dimensional parameters, like those in BNNs [17]. ADVI transforms constrained parameters θ into unconstrained parameters ζ such that $\sigma = T(\zeta)$, where T is a transformation. ADVI models ζ with a multivariate Gaussian

approximation, $q(\zeta) = \mathcal{N}(\zeta|\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ is the mean vector, $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix, and d is the number of parameters in the BNN. ζ is then optimised.

VI is computationally efficient and scalable due to the use of gradient descent as an optimiser, which can be computed quickly and is highly parallelisable [3][23]. However, the necessary restriction and simplification of the posterior means that, particularly when the true posterior is complex, high dimensional, and multi-model as in BNNs, VI can struggle to find an accurate approximation [4]. Furthermore, KL divergence, which VI aims to minimise, is an asymmetric measurement meaning that it penalises underestimation of high-probability regions of $p(\theta|\mathcal{D}$ but not low-probability ones. In practice, this leads to an underestimation of uncertainty and the spread of the posterior, which limits one of the key advantage of BNNs in providing uncertainty with every prediction [19].

2.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods generate samples from the posterior distribution by constructing a Markov chain whose stationary distribution is the target posterior, $p(\theta|\mathcal{D})$ [29]. Among the various MCMC algorithms, this report employs the No-U-Turn Sampler (NUTS), selected for its strong performance in high-dimensional parameter spaces. NUTS builds on Hamiltonian Monte Carlo (HMC), which uses Hamiltonian dynamics to interpret a parameter set θ as a position in a search space, navigating this space using an evolving momentum variable r [13]. In theory, MCMC methods will, in the limit, generate samples that exactly represent the true posterior. In practice, a burn-in period is used to allow the chain to converge to the approximate posterior. Only the samples drawn after this period are retained.

Unlike Variational Inference (VI), MCMC does not impose restrictions on the shape of the posterior. This flexibility allows NUTS, given enough iterations, to closely approximate the true posterior, enabling a Bayesian Neural Network (BNN) to accurately model predictive uncertainty. However, this comes at a significant computational cost. MCMC methods are considerably more expensive than VI, both in total runtime and per iteration. They require a lengthy burn-in period, during which no usable samples are collected [31]. Moreover each NUTS iteration involves repeated differentiation of a complex function more computationally demanding than evaluating the ELBO in VI [27]. As a result, while MCMC methods can deliver more accurate posterior estimates than VI, they do so at a higher computational expense.

3 Proposed Method

3.1 Dataset Bias

The objective of this Bayesian neural network is to predict whether an area is at risk of flooding. The dataset used to train the network consists of 67330 samples with binary labels (at risk of flood or not at risk) and 10 dimensions:

- **Categorical:** Lithology, land use, building type, substrate
- **Continuous:** Elevation, slope, imperviousness, Normalized Difference Vegetation Index (NDVI), distance to nearest river, distance to nearest road

1802 samples, 2.68%, contained missing values. Furthermore, the dataset was heavily biased towards non-flood risk samples (93.35% of samples were negative) which risked producing biased model parameters. A standard approach is to oversample or create synthetic data for the minority positive class. However, both methods are unable to maintain the true feature distributions within classes, which negatively affects Bayesian models which rely on true data distributions to capture uncertainty [1]. Therefore, undersampling was used. Any sample containing missing values was removed. The data was then split using a standard 70/30 ratio into train and test

sets. Finally, the training data was randomly undersampled until the number of positive and negative samples were equal. This results in a relatively small training set of 6142 samples. However, a key advantage of Bayesian methods is their strong performance on smaller datasets relative to frequentist methods [20].

3.2 Dataset Pre-Processing

Continuous features which did not already follow Gaussian distributions were subjected to z-score standardisation, which gave these features a mean of 0 and standard deviation 1. This strategy was picked to increase model stability by reducing the difference in scale of the continuous variables [14]. Elevation, slope, and distance to road all followed non-Gaussian distributions with large ranges, so log was first applied before z-score normalisation.

Figure 1: Raw Distance to Road Distribution

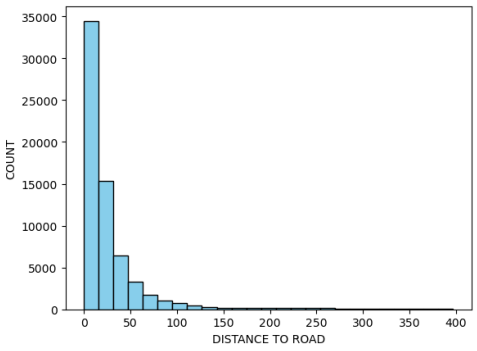
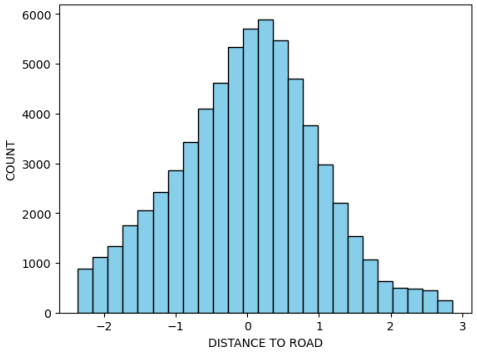


Figure 2: Pre-Processed Distance to Road Distribution



NDVI already exhibited a normal distribution and was therefore min-max normalised to scale values between 0 and 1. Imperviousness, due to a large number of zero values, was also min-max normalised rather than standardised. Ideally, this feature would be decomposed into two: a binary indicator capturing whether the value is zero, and a continuous variable representing the magnitude of non-zero values. However, implementing this transformation would require additional context about the feature and dataset to ensure its validity.

Figure 3: Raw NDVI Distribution

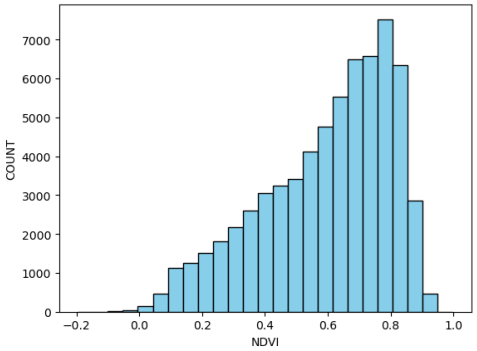
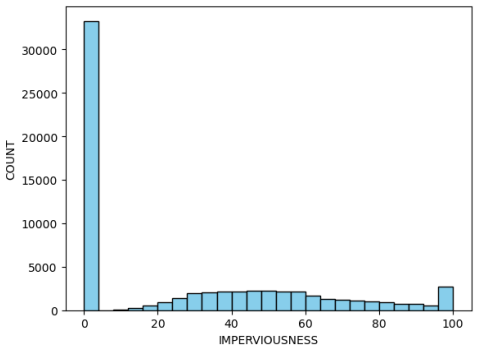


Figure 4: Raw Imperviousness Distribution



All the non-continuous classification features were One-Hot-Encoded. This was because One-Hot-Encoding would scale the classification inputs in line with the standardised continuous inputs, increasing model stability. Furthermore, one superfluous dimension was removed from each feature's One-Hot-Encoded vector to reduce unnecessary complexity and avoid the "curse of dimensionality".

3.3 Bayesian Neural Network Design

The BNN studied was a fully connected network with 32 input nodes and a single output node. The number of hidden layers and hidden nodes varies across experiments. Each layer uses the tanh activation function, except for the final layer which uses sigmoid to output a binary classification probability. The prior distribution for each layer's weights and biases was a multivariate Gaussian distribution with a mean of 0 and standard deviations of 1, $\mathcal{N}(0, I)$. This is the standard prior choice for BNNs [10]. While it has no inherent advantage over other informative priors, unless more information is known about the system and data this Gaussian generally performs as well as any other choice of prior [28]. The BNN used also facilitates mini-batch learning.

3.4 Experiments

To compare the efficiency of Variational Inference (VI) and Markov Chain Monte Carlo (MCMC), the number of iterations required for each method to converge was first determined. For VI, convergence was assessed by plotting the Evidence Lower Bound (ELBO) against the number of iterations and identifying the point at which the curve stabilised. For MCMC, convergence was evaluated by comparing the marginal energy distribution to the energy transition distribution across varying numbers of burn-in samples. When these two distributions align, the MCMC approximation is considered to have converged [2].

With convergence points established, each method's performance was assessed using accuracy, precision, and recall metrics. Computational cost—measured in runtime—was also recorded to compare efficiency. To evaluate how well each method captured predictive uncertainty, the Kullback-Leibler (KL) divergence between posterior approximations was computed, and confidence intervals were visualised. These experiments were repeated on a deeper BNN and a BNN with a poor prior to examine the impact of posterior dimensionality and priors.

All experiments used the No-U-Turn Sampler (NUTS) with four independent chains. Minibatch size was fixed at 50. Posterior evaluations were conducted via Monte Carlo estimation: 2000 weight samples were drawn from each approximated posterior, with each sample used to construct an ANN. Final results were computed as the average output of the sampled models.

4 Results

To find the necessary number of iteration or burn-in samples for VI and MCMC to reach convergence, the number of iterations was varied when training a BNN with 2 hidden layers of 5 nodes using a Gaussian prior with $\mu = 0$ and $\Sigma = I$.

Figure 5: ELBO Against Iterations of ADVI

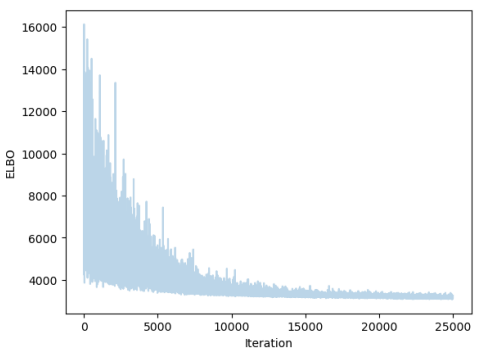


Figure 6: ELBO Against Iterations of Minibatched ADVI

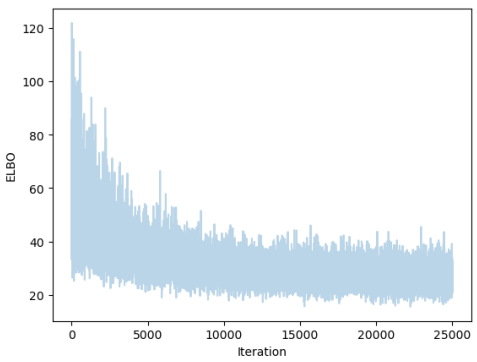
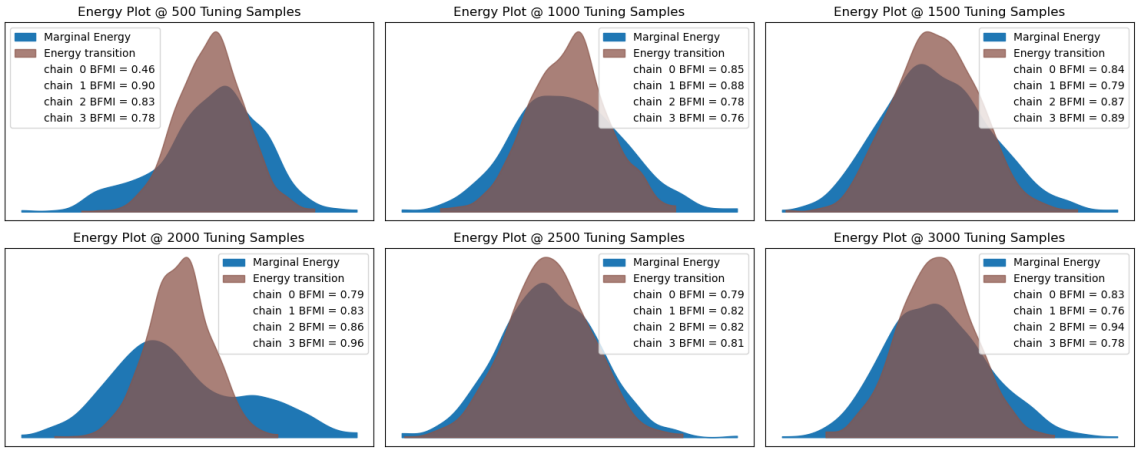


Figure 7: Marginal Energy Against Energy Transition for NUTS with Different Numbers of Burn-In Samples



NUTS, ADVI, and minibatch ADVI were then used to train three different BNNs:

- A standard BNN: 2 hidden layers of 5 nodes using a Gaussian prior with $\mu = 0$ and $\Sigma = I$
- A deep BNN: 4 hidden layers of 10 nodes using a Gaussian prior with $\mu = 0$ and $\Sigma = I$
- A poor prior BNN: 2 hidden layers of 5 nodes using a Gaussian prior with $\mu = 10$ and $\Sigma = 10I$

The metrics of the resulting BNNs (Table 1), comparison between posteriors (Table 2), and confidence intervals (Figures 8 & 9) are given.

BNN	**Approximation**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Runtime (s)**
Standard	NUTS	80.5	24.2	86.3	769
	Minibatch NUTS	78.3	8.7	22.8	56
	ADVI	77.9	20.9	79.4	22
	Minibatch ADVI	75.3	19.1	79.5	6
Deep	NUTS	83.0	27.4	89.2	7805
	ADVI	77.1	20.3	79.3	66
	Minibatch ADVI	76.3	19.3	76.8	8
Poor Prior	NUTS	82.3	25.8	83.6	2618
	ADVI	49.5	5.3	37.2	14
	Minibatch ADVI	86.4	25.0	48.7	11

Table 1: Metrics for BNNs trained using different posterior approximations

BNN	Approximation	KL Divergence with NUTS				
		W_0	W_1	W_2	W_3	W_{out}
Standard	Minibatch NUTS	0.333	0.186	-	-	∞
	ADVI	0.044	0.081	-	-	∞
	Minibatch ADVI	0.022	0.166	-	-	∞
Deep	ADVI	0.031	0.018	0.129	0.029	0.636
	Minibatch ADVI	0.032	0.035	0.024	∞	0.364
Poor Prior	ADVI	0.563	0.777	-	-	3.683
	Minibatch ADVI	0.323	0.419	-	-	2.562

Table 2: KL divergence between the NUTS approximation and other posterior approximations for each layer's weights in different BNNs

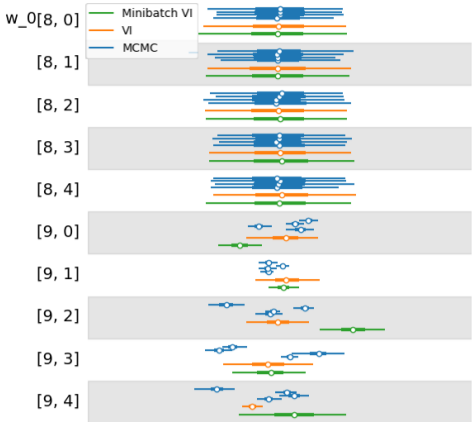


Figure 8: Confidence Intervals of Different Input Weight Posterior Approximations in a Standard BNN

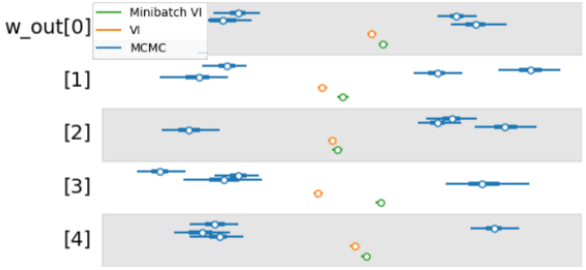


Figure 9: Confidence Intervals of Different Output Weight Posterior Approximations in a Standard BNN

5 Discussion

The necessary number of iterations for both ADVI and minibatch ADVI to reach convergence was found experimentally to be 25000 for the standard BNN setup used (Figures 5 and 6). For NUTS, 2500 burn-in samples were necessary (Figure 7). These settings were used as standard for the different BNN configurations.

NUTS consistently performed slightly better than ADVI and Minibatch ADVI, with both permutations of VI producing either lower accuracy or significantly worse recall (Table 1). Considering the nature of the data, recall is of particular importance: the priority is to identify all the positive at risk of flooding samples. The superiority of NUTS is expected; MCMC approximates the posterior far more accurately than VI [11].

The disparity between accuracy, precision, and recall of NUTS and the VI methods is small, especially for the standard and deep BNNs. This suggests that the maximum a posteriori of the dataset can able to label most samples, and accurately understanding the uncertainty in the posterior is less important. The differences between predictions can be seen far more clearly when considering KL divergence (Table 2). While KL divergence between NUTS and the VI methods is small for early weights, it is very significant at the weights towards the end of the network. This is because the posterior of the weights at the end of the network is naturally more uncertain, and the ADVI methods struggle to capture this uncertainty accurately [5]. The difference can be clearly seen in Figures 8 and 9. In Figure 8, which shows input weights, the VI and MCMC methods demonstrate similar confidence intervals, indicating that they are both

capturing the uncertainty well. However, in Figure 9, the VI methods do not overlap at all with the MCMC posterior, display almost no uncertainty, and fail to capture the multi-modal nature of the posterior evident from the different means of the MCMC chains.

The comparative advantage of MCMC over VI is even more clear in the extreme BNNs. In the Deep BNN, NUTS outperformed both ADVIs by a more significant margin than with the standard BNN, indicating its superiority in approximating high dimensional posteriors [2]. This is likely due to the simplifications assumed by VI being unable to model the more complex distribution. Furthermore, MCMC was unaffected by the poor choice of prior, while crippled VI. The cause of this most probably lies in the fact that MCMC is asymptotically exact, meaning it is guaranteed to perfectly model the posterior given sufficient iterations, despite the choice of prior. In contrast, VI is not asymptotically exact, and the use of KL divergence as an objective actually penalises deviation from the prior [9].

However, VI methods demonstrate a huge lead over MCMC in computational efficiency. ADVI was consistently one or two orders of magnitude faster than NUTS, with this disparity increasing in the extreme BNNs. Furthermore, VI methods are well suited to minibatching [32], which generally increased ADVI's efficiency by another order of magnitude with only a slight drop off in performance. In fact, when provided a poor prior, minibatching appeared to improve ADVI's performance, likely because minibatching facilitates a more data-driven, less prior dependent exploration of the posterior space [12]. MCMC methods like NUTS are work poorly with minibatching because they are (in the limit) exact methods which require the full distribution of data [2]. Minibatching adds noise to the training process which negatively affects NUTS, as evidenced by the results in Table 1, so MCMC cannot benefit from the increase efficiency minibatching provides.

6 Conclusion

This report has investigated the relative strengths and weaknesses of Variational Inference (VI) and Markov Chain Monte Carlo (MCMC) techniques in approximating posterior distributions for Bayesian Neural Networks (BNNs). Through both theoretical exploration and practical experimentation using flood risk data, it was found that MCMC methods, specifically the No-U-Turn Sampler (NUTS), produce more accurate and uncertainty-aware models than VI methods which allow full exploitation of the uncertainty accompanying every prediction in Bayesian learning. MCMC's asymptotic exactness allowed it to outperform VI particularly in high-dimensional and poorly specified scenarios, where VI's simplifying assumptions and reliance on the prior were detrimental.

However, this superior performance comes at a significant computational cost. VI methods, especially when combined with minibatching, offered orders of magnitude faster runtimes, making them attractive for large-scale or time-constrained applications. While they struggled to capture the full complexity of the posterior, especially in later layers of deeper networks, they still produced reasonably accurate predictions with some trade-offs in uncertainty representation.

Overall, the choice between VI and MCMC depends on the specific application needs: MCMC is ideal when uncertainty quantification is paramount, whereas VI is more practical when computational efficiency is the priority. Future research could explore hybrid methods or further enhancements to VI that improve its expressiveness while maintaining efficiency, particularly in real-world applications with high-dimensional data and limited resources.

References

[1] IM Alkhawaldeh, I Albalkhi, and AJ Naswhan. Challenges and limitations of synthetic minority oversampling techniques in machine learning. *World Journal of Methodology*, 13(5):373–378, December 20 2023.

[2] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018.

[3] Kush Bhatia, Nikki Lijing Kuang, Yi-An Ma, and Yixin Wang. Statistical and computational trade-offs in variational inference: A case study in inferential model selection, 2023.

[4] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015.

[6] Lorenzo Brigato and Luca Iocchi. On the effectiveness of neural ensembles for image classification with small datasets. *CoRR*, abs/2111.14493, 2021.

[7] Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan. *Neural Network Design*. Martin Hagan, Stillwater, OK, USA, 2nd edition, 2014.

[8] Julia M. Flores, Ann E. Nicholson, Andrew Brunskill, Kevin B. Korb, and Stefano Mascaro. Incorporating expert knowledge when learning bayesian network structure: a medical case study. *Artificial Intelligence in Medicine*, 53(3):181–204, 2011.

[9] Andrew YK Foong, David R Burt, Yingzhen Li, and Richard E Turner. On the expressiveness of approximate inference in bayesian neural networks. *arXiv preprint arXiv:1909.00719*, 2019.

[10] Vincent Fortuin. Priors in bayesian deep learning: A review, 2022.

[11] Ankush Ganguly and Samuel W. F. Earp. An introduction to variational inference, 2021.

[12] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pages 2348–2356, 2011.

[13] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 448–456. PMLR, 2015.

[15] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. Risk Acceptance and Risk Communication.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[17] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference, 2016.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Charles C. Margossian and Lawrence K. Saul. The shrinkage-delinkage trade-off: An analysis of factorized gaussian approximations for variational inference, 2023.

[20] Daniel McNeish. On using bayesian methods to address small sample problems. *Structural Equation Modeling A Multidisciplinary Journal*, 23, 05 2016.

[21] John Mitros and Brian Mac Namee. On the validity of bayesian neural networks for uncertainty estimation, 2019.

[22] Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer-Verlag, 1996.

[23] Willie Neiswanger, Chong Wang, and Eric Xing. Embarrassingly parallel variational inference in nonconjugate models, 2015.

[24] Theodore Papamarkou, Jacob Hinkle, M. Todd Young, and David Womble. Challenges in markov chain monte carlo for bayesian neural networks, 2021.

[25] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[26] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[27] Tim Salimans, Diederik P. Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap, 2015.

[28] Daniele Silvestro and Tobias Andermann. Prior choice affects ability of bayesian neural networks to identify unknowns. *CoRR*, abs/2005.04987, 2020.

[29] Joshua S. Speagle. A conceptual introduction to markov chain monte carlo methods, 2020.

[30] Jack V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11):1225–1231, 1996.

[31] Don van Ravenzwaaij, Pete Cassey, and Scott D. Brown. A simple introduction to markov chain monte–carlo sampling. *Psychonomic Bulletin & Review*, 25(1):143–154, 2018.

[32] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference, 2018.