THE UNIVERSITY *of* EDINBURGH

Probabilistic Inference in Bayesian Neural Networks

Alisa Sheinkman

Doctor of Philosophy
The University of Edinburgh
July 21, 2025

Abstract

Despite widespread applicability and the dominant role in machine learning, neural networks remain highly non-transparent and are often regarded as black boxes due to the lack of human-understandable interpretations. Conventional deep models tend to be overconfident in predictions, provide poor uncertainty estimates and are sensitive to adversarial attacks. The Bayesian paradigm takes a step further and provides a natural framework to address these challenges by considering infinite ensembles of differently weighted neural networks. The Bayesian neural networks are capable of capturing the uncertainty, improving the accuracy and controlling the model complexity. Unfortunately, for most real-world problems, the exact probabilistic inference is unavailable, and the asymptotically faultless Markov chain Monte Carlo becomes daunting when dealing with large high-dimensional datasets and multimodal posteriors of neural networks. At the same time, faster and computationally appealing optimization-centric variational inference lacks the theoretical justification of the sampling-based methods and is known to underestimate the uncertainty of the true posterior distribution. This thesis systematically studies different aspects of variational inference, namely, theoretical foundations, challenges and means of dealing with those. Further, the practical questions arising when implementing and comparing Bayesian neural networks are addressed, and the dependency of the predictive performance on the architectural choices and the alignment between the model and the inference algorithm are analysed. Finally, this thesis contributes to the development of variational inference techniques and presents a novel kind of Bayesian neural network called a variational bow tie neural network in which we employ sparsity-promoting priors and consider the improved version of the classical coordinate ascent variational inference algorithm.

Lay Summary

Artificial intelligence and machine learning models have been proven to be highly effective in a wide range of tasks, such as image and text recognition and generation, autonomous driving and pharmacological clinical trials. However, despite their popularity, these models often lack explainability, meaning that humans still regard artificial intelligence as a "black box" and find it difficult to understand the way machine learning models behave. The majority of classical deep models tend to be overconfident in their predictions, can be misled by small changes in the input data and do not provide reliable measures of uncertainty.

The Bayesian approach treats models as probabilistic and accounts for the uncertainty in predictions in the principal way. Furthermore, the formulation of Bayesian machine learning models naturally makes them much more transparent and interpretable. A Bayesian model can be seen not as a single model but as an infinite collection of classical models; this enables them to model uncertainty, improve accuracy and be less overconfident. Thus, the approach became a gold standard in safety-critical and decision-making applications.

Unfortunately, for real-world large-scale problems, calculating exact probabilities is a significant challenge, that is because traditional methods of Bayesian statistics are too computationally expensive to be able to achieve good performance. This thesis addresses theoretical foundations and practical concerns related to the implementation and use of Bayesian machine learning models. It systematically studies existing solutions, provides novel tools, and contributes to the development of efficient and reliable ways of understanding and improving AI models.

To summarise, this thesis adopts a probabilistic Bayesian view of modern machine learning, studies arising challenges and proposes possible solutions with interpretability, uncertainty, reliability and practical concerns in mind.

To my parents

Acknowledgements

First and foremost, I am extremely grateful to my supervisor, Sara Wade, who accepted me as a student, when all of a sudden, I approached her in February 2022, already being a year and a half into my PhD course and willing to switch to Bayesian statistics from a completely different field of algebraic geometry. I feel extremely fortunate to be able to learn from Sara and cannot imagine a better supervisor. I am also thankful to my second supervisor, Miguel Anjos, for his support and helpful non-research discussions.

The first year and a half of my PhD were slightly chaotic and not very productive, I would like to thank David Jordan for spending time on me and after all, with the School of Math's help, letting me change the area of research. Even though that first period was not very bright, being in the PhD cohort with Hannah and Patrick made the days more cheerful.

The last paragraph of acknowledgements is often the most sentimental one, and this thesis is no exception. I am not a joyful or an easy-going person, and I am fortunate to have an incredibly understanding family. My parents' immense support helps me to overcome the most uncertain and gloomiest of days, perhaps, this is a rare case when the cliche "without them it would not be possible" can be said without exaggeration. I am thankful to my friends, Dima and Kashin, for being around and being the way they are.

Contents

Glossary

a scalar

a_i element i of a vector

$\boldsymbol{a}$ vector

$\mathbb{R}^D$ D-dimensional real space

$\boldsymbol{a}_{-j}$ vector without the j-th element, $\boldsymbol{a} \setminus a_j$

$\mathcal{D}$ data

$\boldsymbol{x}$ inputs of the data

$\boldsymbol{y}$ outputs of the data

$\tilde{\mathcal{D}}$ unseen (training) data

$\tilde{\boldsymbol{x}}$ input of the previously unseen data

$\tilde{\boldsymbol{y}}$ output of the previously unseen data

L number of hidden layers in the neural network

D_l number of hidden units in the layer l of the neural network

$\mathrm{KL}(q||p)$ reverse Kullback-Leibler divergence

$\mathrm{KL}(p||q)$ forward Kullback-Leibler divergence

$\mathcal{N}(\mu, \sigma^2)$ Gaussian distribution with mean μ and standard deviation σ

Acronyms

ADVI automatic differentiation variational inference

AI artificial intelligence

BBVI black box variational inference

BMA Bayesian model averaging

BNN Bayesian neural network

CAVI coordinate ascent variational inference

DAG directed acyclic graph

DL deep learning

EC empirical coverage

ELBO evidence lower bound

elpd expected log pointwise predictive density

EM expectation-maximization

EP expectation-propagation

GIG generalized inverse Gaussian

GP Gaussian process

HMC Hamiltonian (hybrid) Monte Carlo

LOO-CV leave-one-out cross-validation

lpd log pointwise predictive density

MAP maximum-a-posteriori estimate

MCMC Markov chain Monte Carlo

ML machine learning

MLE maximum likelihood estimate

NN neural network

OOD out-of-distribution

PPC posterior predictive checks

PSIS Pareto-smoothed importance sampling

RMSE root mean squared error

SGD stochastic gradient descent

SVI stochastic variational inference

VAE variational auto-encoders

VBNN variational bow tie neural network

VI variational inference

WAIC Watanabe-Akaike information criterion

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction: Probabilistic Inference Meets Neural Networks

The term artificial intelligence (AI) was coined in 1955 [McCarthy et al., 2006], and has since been used to encompass a class of "intelligent" machines and computational systems capable of solving a broad range of problems[1]. Closely related to AI are the fields of machine learning (ML) and, given the increased popularity of larger models, deep learning (DL), both of which study methods for information processing, learning and reasoning from data [Barber, 2012]. Since the beginning of the century, deep learning has achieved exceptional performance and has become a significant part of not only modern scientific discovery but also everyday life. A few examples where machine learning models are applied include tracking and understanding the spread of SARS-CoV-2 [Brito et al., 2022, Flaxman et al., 2022], brain Magnetic Resonance Imaging analysis [Flandin and Penny, 2007], large language models underlying the success of generative AI chatbots [Touvron et al., 2023], and forecasting Presidential elections [Gelman et al., 2024]. With the increase in available computing power and the tremendous popularity of AI, a lot of questions are being raised about the safety, reliability and black-box behaviour of deep learning; classical deep learning models are not robust to adversarial attacks, can exhibit mysterious behaviour and do not offer human-understandable explanations [Lipton, 2018, Szegedy et al., 2014].

Given the complex and uncertain nature of real-world tasks, modern AI systems can be fundamentally improved by adopting and exploiting the power of the Bayesian framework. The key distinguishing ability of Bayesian modelling is that, unlike the classical deep models, it incorporates domain expertise and provides uncertainty quantification; this makes Bayesian inference particularly useful in the era of big data, elaborate tasks and the need for reliable and robust models we can trust [Broderick et al., 2023, Papamarkou et al., 2024].

[1]The origin story of the name is rather sad and very human. The motivation of McCarthy when creating it was to avoid the term cybernetics and not to invite one of the founding fathers of computer science and artificial intelligence, Norbert Wiener, to a conference [McCarthy, 1996].

1.1 Preliminaries on classical models

Neural networks (NNs) are hierarchical (layered) models mapping the input to the output via a sequence of hidden layers, consisting of hidden units, also known as neurons. This section introduces classical feed-forward neural networks. Such networks come with no backwards pointing connections and no cycles between the units, and are also historically known as multilayer perceptrons and backpropagation networks [Neal, 1995, Rumelhart et al., 1986]. We make a convention that, except briefly mentioning convolutional neural networks (CNNs), recurrent neural networks (RNNs) and Transformers in Section 1.1.2, all the networks considered in this thesis are examples of feed-forward neural networks.

1.1.1 Structure of neural networks

The number of hidden layers L in a neural network is known as the depth of the network, the number of hidden units per layer D_l is the width of the layer l and the total number of hidden units in a network (depth multiplied by width if widths of hidden layers are the same) is known as the network's capacity. Suppose we observe data consisting of inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$, namely $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$, where $\boldsymbol{x}_n$ is a D_0-dimensional real-valued vector and $\boldsymbol{y}_n$ is a D_{L+1}-dimensional vector. For each data entry $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ and for $l = 1, \ldots, L$ the hidden units $\boldsymbol{z}_{n,l} \in \mathbb{R}^{D_l}$ are obtained from the previous layer's units $\boldsymbol{z}_{n,l-1} \in \mathbb{R}^{D_{l-1}}$ by applying an affine transformation followed by a non-linear function g called activation. Specifically,

$$\boldsymbol{z}_l = g(\boldsymbol{W}_l \boldsymbol{z}_{l-1} + \boldsymbol{b}_l),$$

where $\boldsymbol{W}_l \in \mathbb{R}^{D_l \times D_{l-1}}$ and $\boldsymbol{b}_l \in \mathbb{R}^{D_l}$ are the weights and biases of the layer l and we assume $\boldsymbol{z}_0 = \boldsymbol{x}$. Then the units of the output layer of the neural network are given by $\boldsymbol{W}_{L+1}\boldsymbol{z}_L + \boldsymbol{b}_L$. For $d = 1, \ldots, D_l$, the d-th row of the matrix $\boldsymbol{W}_l$ is denoted as $\boldsymbol{W}_{l,d}$ (note, $b_{l,d}$ is a scalar); further, we denote the collections of weights and biases across all layers as $\boldsymbol{W} = \{\boldsymbol{W}_l\}_{l=1}^{L+1}$ and $\boldsymbol{b} = \{\boldsymbol{b}_l\}_{l=1}^{L+1}$. In regression problems, the continuous output $\boldsymbol{y} \in \mathbb{R}^{D_{L+1}}$ is given by

$$\boldsymbol{y} = \boldsymbol{W}_{L+1}\boldsymbol{z}_L + \boldsymbol{b}_L + \boldsymbol{\Sigma}, \tag{1.1}$$

where $\boldsymbol{\Sigma}$ is some noise, typically modelled as $\boldsymbol{\Sigma} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2)$ for some standard deviation $\boldsymbol{\sigma}$. Whereas in classification tasks with a K-valued target, the probability that the output $\boldsymbol{y}$ belongs to the k-th category is given by

$$\mathbb{P}(\boldsymbol{y} = k \mid \boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1}, \boldsymbol{z}_L) = \frac{\exp\left(\boldsymbol{W}_{L+1,k}\boldsymbol{z}_L + b_{L,k}\right)}{\sum_{i=1}^K \exp\left(\boldsymbol{W}_{L+1,i}\boldsymbol{z}_L + b_{L,i}\right)}.$$

There is a natural correspondence between neural networks and directed acyclic graphs (DAGs), for the feed-forward network, the flow of information goes from the input to the output in such a way that the DAG resembles a chain (see

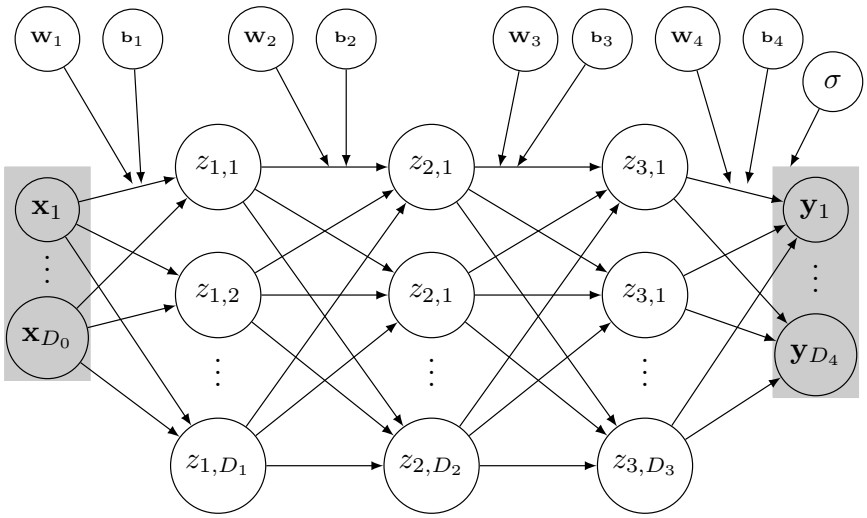


Figure 1.1: DAG of a feed-forward neural network with $L = 3$ hidden layers, input $\boldsymbol{x} \in \mathbb{R}^{D_0}$ and output $\boldsymbol{y} \in \mathbb{R}^{D_4}$ given by Equation (1.1).

Figure 1.1).

The goal of training a neural network is to find a set of weights that, given the input, produces output that is "the closest" to the truth in terms of some loss or error function. Classical neural networks with continuous outputs are trained by minimizing the mean squared error (possibly with some added penalty term) in gradient-based optimization, and in categorical tasks, the role of the error term is played by cross-entropy. The first historical example of training neural networks with such a procedure is known as backpropagation [Rumelhart et al., 1986]. Gradient-based optimization methods used in classical deep models include gradient descent, stochastic gradient descent (SGD), AdaGrad and Adam [Sun et al., 2019b]. And so the performance and convergence of the training procedure of the neural networks rely on a careful choice of the learning rate and the initialization of the weights. Finding a suitable initialization scheme is both vital and challenging [He et al., 2015], for the sake of brevity, here we omit the details and refer to [Arbel et al., 2023, pages 34-36] for a review. Regularization techniques used in neural networks are discussed in Section 1.2.5.

1.1.2 Architectural nuances and challenges

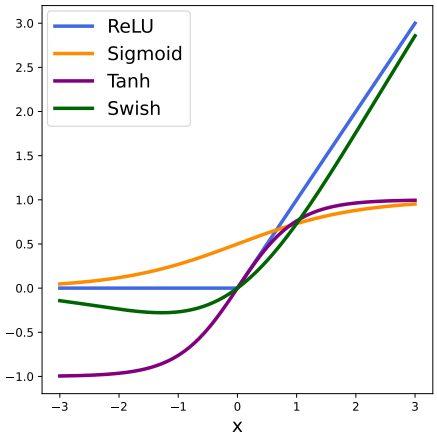
Note that the activation function allows modelling the non-linear relation between the input and the output, and taking the activation function to be linear would reduce neural network to a single linear or logistic regression model. We list and illustrate some of the listed activation functions by Table 1.1 and Figure 1.2a. Another fundamental requirement for the activation function (apart from being non-linear) is differentiability. The sigmoid and the hyperbolic tangent are some of the earliest commonly used activation functions; still, both are often considered to be non-optimal in the sense that these functions saturate and lead to vanishing gradients [Murphy, 2022]. For example, for relatively large positive and negative numbers, the sigmoid function saturates around 1 and 0,

and the tanh function saturates around -1 and 1, which results in zero values of the gradient [Dosovitskiy et al., 2020, Mikolov et al., 2013] (see Figure 1.2a). Nowadays, a popular choice of the activation function is the rectified linear unit function (ReLU), which switches the negative inputs off and leaves the positive ones unchanged. ReLU gained popularity due to its superior accuracy and improved convergence over sigmoid and tanh activations [He et al., 2015, Srivastava et al., 2014]. Unfortunately, with the increase in depth and a wrong choice of initialization, there is a high risk of dying ReLU, that is, the phenomenon occurring in deeper neural networks when ReLU neurons become inactive and only output zeros [Lu, 2020]. In a search for optimal activation function, various both piece-wise linear and smooth generalisations of ReLU were proposed, including examples such as leaky or parametric ReLU (PReLU), softplus (also known as a smooth rectifier), exponential linear unit (ELU) and Swish [Ramachandran et al., 2017].

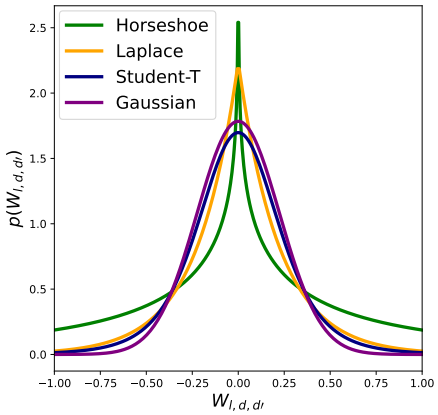
Table 1.1: Some of the common activation functions. Note that in parametric ReLU, $\alpha = 0$ corresponds to ordinary ReLU and $\alpha = 0.01$ to what is known as leaky ReLU. Further, when in the swish function $\alpha = 1$ it is known as a sigmoid linear unit (SiLU).

Name	Notation	Formula
Rectified linear unit	ReLU	$\max(0, x)$
Sigmoid	σ	$\frac{\exp(x)}{\exp(x)+1}$
Hyperbolic tangent	tanh	$\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$
Parametric ReLU	PReLU	$\begin{cases} x, & x > 0, \\ \alpha x, & x \leq 0 \end{cases}$
Exponential ReLU	ELU	$\begin{cases} x, & x > 0, \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases}$
Softplus	SoftPlus	$\log(\exp(x) + 1)$
Swish	swish	$x\sigma(\alpha x)$

Whilst the dimensions of the input and the output are determined by the dimensionality of the data set, the dimension of the weight space, can be tuned to improve prediction performance. In the case of feed-forward neural networks, this amounts to finding optimal depth and width. The universal approximation theorem guarantees that a wide enough feed-forward neural network with a single hidden layer can express any smooth function [Hornik et al., 1989]. At the same time, there are variants of this theorem for deeper architectures [Hanin, 2019, Lu et al., 2017], and both theoretical arguments [Chatziafratis et al., 2020, Eldan and Shamir, 2016, Håstad, 1986, Telgarsky, 2016] and remarkable performance in certain domains [Krizhevsky et al., 2017, Silver et al., 2016] support the ad-



(a) Four of the common activation functions: ReLU, sigmoid, tanh and swish with $\alpha = 1$ (i.e. SiLU).



(b) Four of the possible choices for the density of the weights: Gaussian, Student-t, Laplace and Horseshoe.

Figure 1.2: Several examples of activation functions and priors on the weights.

vantages of increased depth. In practice, constructing a model which is not only expressive but generalizes well remains a major challenge, and should be done in a thoughtful manner. Despite the tremendous success in areas such as natural language processing and computer vision [Dosovitskiy et al., 2020, Krizhevsky et al., 2017, Touvron et al., 2023], often there is no clear understanding of why a particular model generalizes well [Zhang et al., 2021a].

When the model fails to capture the relation between the input and the output of the training set, it is said to underfit and have a high bias. Underfitting occurs when the model is too simple for the task or the training was not performed for long enough. Instead, many modern machine learning models are over-parametrized, and prone to overfitting, especially given the limited size of the dataset. Complex problems demand exploring bigger model spaces, and there is a danger of choosing an excessively over-parametrized model which is going to overfit and has a high variance. Such a model fits 'too well' to the training data and captures the dependencies of the training set which are not present in the test set, this results in a low train error but high test error [Gelman et al., 2020, MacKay, 2003]. To improve generalization abilities, one needs to reduce overfitting without enforcing underfitting of the model.

Given a finite number of observations, the model's generalization abilities are closely tied to the presence of inductive bias. When choosing a model and a training procedure, we make some assumptions about the structure of the dataset and the associated predictive task. In this way, we embed some inductive bias. The so-called no free lunch theorems [Shalev-Shwartz and Ben-David, 2014, Wolpert, 1996] are historically used to dictate that there is no panacea to solve every problem, and no single model can be appropriate in a range of tasks. This argument, however, should not be taken naively; empirical findings [Fernández-Delgado et al., 2014] and connections between generalization abilities and Kolmogorov complexity [Goldblum et al., 2024] show that a model, which combines flexibility and a soft simplicity bias, can perform well across diverse datasets.

With the development of deep learning, different model architectures have been shown to be particularly effective with different types of data. For example, the architecture of convolutional neural networks [Krizhevsky et al., 2017], exploits the assumption of compactness and translation invariance to promote encoding the information from the image data; recurrent neural networks [Hochreiter and Schmidhuber, 1997] are naturally well-suited for the sequential observations; and attention-based Transformers have achieved remarkable performance in language processing tasks [Vaswani et al., 2017]. At the same time, we note that Transformers and CNNs were empirically shown to generalize well across different domains [Dosovitskiy et al., 2020, Goldblum et al., 2024, Gruver et al., 2024].

Robustness to out-of-distribution (OOD) data remains a significant challenge in deep learning [Carlini and Wagner, 2017, Hendrycks and Dietterich, 2018, Hendrycks et al., 2021], many of the classical NNs can be easily misled and are susceptible to adversarial attacks [Nguyen et al., 2015, Szegedy et al., 2014, Uesato et al., 2018, Zhang et al., 2020b, Zong et al., 2024]. Additionally, conventional deep models do not offer human-understandable explanations and lack interpretability [Lipton, 2018]. While explainable AI (XAI) and methods for interpreting the reasoning behind black-box model decisions are an active line of contemporary research [Alvarez-Melis and Jaakkola, 2018, Guidotti et al., 2018], there is no consensus on what can serve as a satisfactory explanation and what cannot [Confalonieri et al., 2021].

Finally, in any decision-making process, reliable uncertainty quantification is crucial, and it is not enough to obtain a point estimate of the prediction. By default, classical neural networks do not address the uncertainty associated with their parameters, and whilst there exist proposals enabling NNs to provide some uncertainty estimates, they are often miscalibrated [Gal, 2016, Guo et al., 2017]. As a result, these models are typically overconfident and provide a low level of uncertainty even when data variations occur [Ashukha et al., 2020, Hein et al., 2019, Zhang et al., 2024].

1.2 Bayesian perspective

This section introduces Bayesian inference and focuses on various aspects of Bayesian neural networks (BNNs). The idea of looking at neural networks through the lenses of statistics and probability originated more than forty years ago. The Boltzmann machine can be seen as perhaps the first example of an undirected neural network, endowed with probability structures, and learned via Gibbs sampling [Ackley et al., 1985] or later using mean-field approximations [Peterson and Anderson, 1987]. Denker et al. [1987] proposed assigning and learning the probability over the weight space of a feed-forward neural network, interestingly, the way of introducing a prior on the weights was called "throwing darts at weight space". Tishby et al. [1989] extended this idea and considered posterior predictive distributions of neural networks in the context of the choice of architecture. The Laplace approximation with either diagonal [Denker and LeCun,

1990] or full [Buntine and Weigend, 1991] covariance matrix was then proposed to obtain the output feed-forward neural network. The Laplace approximations with full covariances were further advocated by MacKay [1992], which studied model choice, regularization and uncertainty quantification in Bayesian neural networks. Around the same time, Neal [1992a] trained BNNs using the Hybrid or as widely known nowadays Hamiltonian Monte Carlo (HMC), while Hinton and van Camp [1993] proposed a method, which was formulated in the information theory language, and can be seen as a variational way of obtaining a fully-factorized Gaussian approximation of a BNN. An important connection between belief networks and feed-forward neural networks [Neal, 1992b] allowed for probabilistic interpretation within the context of graphical models, and variational inference (VI) algorithms for sigmoid belief networks were developed [Saul et al., 1996]. Another big step of Neal [1995] connected infinitely wide Bayesian neural networks with Gaussian processes (GPs). Finally, building on the mixtures of factorized distributions [Bishop et al., 1998, Jaakkola and Jordan, 1998], a variational inference algorithm[2] with a full-covariance Gaussian family was designed for BNNs with Gaussian priors [Barber and Bishop, 1998].

1.2.1 Bayesian inference

In the language of probabilistic modelling, unknown quantities are treated as random variables equipped with probability distributions, and probabilistic inference aims to compute these distributions.

Consider the dataset $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$. Given new inputs $\tilde{\boldsymbol{x}}$, we wish to be able to draw conclusions about the likely values of the unseen $\tilde{\boldsymbol{y}}$, and to achieve that, we introduce a set of parameters $\boldsymbol{\theta}$, which cannot be directly observed and called latent variables. In Bayesian modelling, one places some prior distribution over the latent variables $p(\boldsymbol{\theta})$ such that it encodes the prior beliefs on which values it can take. In this way, one introduces a statistical model $p(\mathcal{D}, \boldsymbol{\theta})$ and probabilistic inference aims to obtain the predictive distribution $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \mathcal{D})$. The product rule of probabilities leads the Bayes' theorem [Bayes, 1763, Laplace, 1891]

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}. \tag{1.2}$$

In other words, the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is proportional to the product of the prior $p(\boldsymbol{\theta})$ and a function of $\boldsymbol{\theta}$ known as the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$. If one is able to obtain the posterior $p(\boldsymbol{\theta}|\mathcal{D})$, then the posterior predictive distribution is given by averaging the $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{\theta})$ over the posterior $p(\boldsymbol{\theta}|\mathcal{D})$:

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \mathcal{D}) = \int p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}, \tag{1.3}$$

where we assumed that $\tilde{\boldsymbol{y}}$ and $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{\theta}$. While frequentists statistics obtains point estimates, the cornerstone of Bayesian infer-

[2]Barber and Bishop [1998] called variational inference "ensemble learning", but we do not use this terminology to avoid confusion with contemporary ensembles of neural networks (for a discussion of modern deep ensembles see Section 1.2.5).

ence lies in computing the posterior $p(\boldsymbol{\theta}|\mathcal{D})$, marginalizing over which deals with the uncertainty in a principal and fundamental way. In this way, Bayesian models not only account for the uncertainty but also distinguish the two types of uncertainty; $p(\boldsymbol{\theta}|\mathcal{D})$ encompasses the epistemic uncertainty coming from the model, while the aleatoric data uncertainty is encoded in $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}}, \boldsymbol{\theta})$.

The normalizing constant $p(\mathcal{D})$ in Equation (1.2) is called the evidence, the marginal likelihood or the prior predictive distribution, and is calculated by taking the integral over all possible values of $\boldsymbol{\theta}$ (sum if $\boldsymbol{\theta}$ is discrete)

$$p(\mathcal{D}) = \int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{1.4}$$

The model's marginal probability $p(\mathcal{D})$ is a key component in model comparison based on Bayes factors, with higher values providing stronger support for the model [Jeffreys, 1939, Kass and Raftery, 1995]; it also arises in Bayesian model averaging (BMA) [Hoeting et al., 1999], which combines predictive distributions obtained by several models based on models' marginal probabilities. Suppose $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are parameters of the models M_1 and M_2, about which we have some prior beliefs given by the corresponding priors $p(M_1)$ and $p(M_2)$. Using Bayes theorem for $p(M_1|\mathcal{D})$ and $p(M_2|\mathcal{D})$, the ratio of posterior probabilities of models, the posterior odds, can be written as a product of two ratios:

$$\frac{p(M_1|\mathcal{D})}{p(M_2|\mathcal{D})} = \frac{p(\mathcal{D}|M_1)}{p(\mathcal{D}|M_2)} \times \frac{p(M_1)}{p(M_2)},$$

where the second ratio is known as prior odds and the first ratio, the ratio of evidences of two models, is the Bayes factor for a model M_1 over model M_2, that is

$$\text{Bayes factor} = \frac{p(\mathcal{D}|M_1)}{p(\mathcal{D}|M_2)} = \frac{\int p(\boldsymbol{\theta}_1|M_1)p(\mathcal{D}|\boldsymbol{\theta}_1, M_1)d\boldsymbol{\theta}_1}{\int p(\boldsymbol{\theta}_2|M_2)p(\mathcal{D}|\boldsymbol{\theta}_2, M_2)d\boldsymbol{\theta}_2}.$$

The Bayes factor reflects the evidence given by $\mathcal{D}$ against model M_2 in favor of model M_1, Kass and Raftery [1995] provided a heuristic for model choice based on the posterior odds with values exceeding 3 favouring model M_1.

However, we note that the Bayes factors are not tractable for the majority of Bayesian deep learning models; and even when computed or approximated, are not always reliable [Moran et al., 2023, Wilson and Izmailov, 2020, X. Xu and Xu, 2019] (for an example of failure of Bayes factors in classical Bayesian models we refer to [Gelman et al., 2013, pages 183-184]). We further discuss the suboptimality of Bayes factors and Bayesian model averaging in Chapter 3.

1.2.2 Bayesian neural networks

Classical neural networks have several limitations, they tend to be overconfident in predictions, do not generalize well and do not come with uncertainty estimates (see Section 1.1.2). Bayesian neural networks emerge as a compelling extension of conventional deep models and are naturally capable of quantifying

uncertainty. Consider the feed-forward neural network we introduced above with weights treated as random variables which we now wish to infer. To design a BNN we need to endow weights with some prior distribution $p(\boldsymbol{W})$, and we can proceed with the biases $\boldsymbol{b}$ (and other parameters of the network) in a similar way. Denote the set of parameters of the network as $\boldsymbol{\theta}$, e.g. the model described by Equation (1.1) and illustrated by Figure 1.1 has $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\sigma}\}$. Then by integrating over the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ we can obtain the posterior predictive distribution of the output $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$. While the idea of the prior is simple: it should encode our beliefs about the structure of the task, the weight space of neural networks is usually high-dimensional and identifying the connection between the imposed prior and the resulting prediction is challenging. The choice of prior is not a straightforward task and is considered in Section 1.2.4.

Marginalizing over the posterior provides uncertainty quantification and improves standard neural networks in a principled way [Papamarkou et al., 2024, Wilson, 2020]. Bayesian neural networks have been found successful in decision-making and safety-critical applications, including the detection of adversarial examples and hallucinations [Farquhar et al., 2024, Smith and Gal, 2018], healthcare and drug discovery [Gruver et al., 2023, Klarner et al., 2023], autonomous driving [McAllister et al., 2017], computer vision [Kendall and Gal, 2017] and large language models [Melo et al., 2024]. However, the high-dimensional and complex nature of Bayesian neural networks makes computations challenging, and the reliability and efficiency of approximation techniques are an active line of research [Coker et al., 2022, Franssen and Szabó, 2022, Ghosh et al., 2019, Izmailov et al., 2021, Papamarkou et al., 2022, Trippe and Turner, 2018]. Recall, that total uncertainty can be defined as the sum of epistemic and aleatoric uncertainties. Many of the real-world scenarios provide data with high aleatoric uncertainty, i.e. the data comes with a high level of noise which does not depend on the number of data points we take. The epistemic or the model's uncertainty comes from the parameters of the model and is, in contrast, reduced with the increase in observations. BNNs account for and distinguish these two types of uncertainty: the uncertainty of the weights is encoded by the posterior distribution and the aleatoric uncertainty can be modelled through the likelihood [Gal, 2016, Gal and Ghahramani, 2016]. As a result, Bayesian models are more resistant to distribution shifts and are able to improve the accuracy and calibration of classical deep models [Ovadia et al., 2019]. We also note that there have been several not always agreeing perspectives on the interpretation of the aleatoric and epistemic uncertainties, and there is a concern that in the context of modern machine learning, this decomposition may be overly simple [Smith et al., 2024].

The behaviour of BNNs as the width tends to infinity is often studied through the lens of Gaussian processes. The seminal result first obtained for neural networks with one hidden layer [Neal, 1995] and then extended to arbitrary depth [Matthews et al., 2018] states that the distribution of the BNN's output induced by the prior converges the neural network Gaussian Process (NNGP), that is GP with a neural network kernel. If the width of some layers is held at finite width, then the limit of the BNN corresponds to a bottleneck NNGP, that is a Deep Gaussian Process resulting from the composition of GPs with neural network kernels [Agrawal et al., 2020]. Later, [Hron et al., 2022] proved that

under certain assumptions, a similar result holds for the distribution of the BNN induced by posterior. The correspondence with GPs became a useful tool when evaluating the quality of BNN approximations and there is ongoing research for which GPs serve as a guide for a true posterior [Rasmussen and Williams, 2005]. We will encounter this correspondence in Section 3.2.2, when reasoning about the example of the mean-field approximation of the BNN which completely ignores the data as the width goes to infinity.

1.2.3 Bayesian inference in neural networks

In Bayesian neural networks, as in the vast majority of modern statistical models (excluding rare rather simple choices of the prior and the likelihood), neither of the integrals in Equations (1.3) and (1.4) is available in the closed form, and computing the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ by simply using Equation (1.2) is not feasible. This leads to the approximate Bayesian inference methods which broadly could be divided into two paradigms: the first reaches the posterior by using sampling, and the second follows an optimization objective and approximates unavailable distributions with some tractable distribution.

Markov chain Monte Carlo (MCMC) methods are widely used and considered to be the gold standard for obtaining unavailable posterior distributions. The idea of MCMC is to construct a Markov chain of random variables by sequentially sampling draws and then utilising Monte Carlo integration. Asymptotically the chain is guaranteed to reach the true posterior, this makes MCMC an appealing method, as long as one is able to sample for a long enough time. Some of the commonly known simulation methods are Metropolis-Hastings [Hastings, 1970], Gibbs sampling [Geman and Geman, 1984] and a workhorse of modern Bayesian modelling Hamiltonian Monte Carlo [Neal, 1995]. Since high-dimensional posteriors arising in BNNs often make classical MCMC computationally intractable, several improvements have been developed including variations of the MCMC with stochastic gradient [Welling and Teh, 2011, Zhang et al., 2020a], adaptive step sizes [Hoffman and Gelman, 2014], normalizing flow proposals [Brofos et al., 2022], importance sampling [Martino et al., 2018] and particle filtering [Chopin et al., 2020].

The second class of methods posits an optimization objective over some family of tractable distributions and obtains the approximation of the true posterior by following this objective. Note that the optimization-based methods are sometimes referred to as deterministic in the sense that such methods assume a certain predetermined form of approximation distribution [Barber, 2012]. Optimization-based methods include Laplace approximation [Tierney and Kadane, 1986], expectation-maximization (EM) [Dempster et al., 1977], loopy belief propagation [Murphy et al., 1999] later extended to expectation-propagation (EP) [Minka, 2001] and, finally, what can be seen as the generalization of the above methods, variational inference [Jordan et al., 1999]. Even though not considered in this thesis, a broader framework known as generalized variational inference is worth mentioning and is an active line of research [Bissiri et al., 2016, Knoblauch et al., 2022, Matsubara et al., 2022, Walker, 2006].

Despite the promise of asymptotic guarantees and numerous improvements

in MCMC, the sampling-based framework may still be computationally challenging. Computational efficiency gained by replacing sampling with optimization allows approximate Bayesian inference to handle high-dimensional models and large data. However, compared to the Markov chain Monte Carlo, variational inference does not offer asymptotic guarantees, and the quality of the resulting approximation often requires close attention. This motivates the focus of this thesis on understanding and improving variational inference methods; namely, in Chapter 2 the state-of-the-art VI methods are explored, in Chapter 3 we evaluate the empirical performance of VI compared to HMC in various BNNs, and in Chapter 4 a novel type of variational BNNs is designed.

While this thesis heavily focuses on the optimization-based paradigm and the perspective of neural networks, the reader interested in the classical results on sampling-based methods is referred to [Robert et al., 1999], on the overview of modern developments and challenges of MCMC to [Angelino et al., 2016, Papamarkou et al., 2022], and in the general historical overview of Bayesian computation to [Martin et al., 2020].

1.2.4 Priors

A grand challenge of Bayesian neural networks, besides computing the posterior, lies in specifying sensible priors. In the Bayesian framework, one encodes the assumption on the problem structure by placing some prior distribution on the parameters of the network, and then the careful choice of prior allows for the reasonable inductive bias [Wilson and Izmailov, 2020].

The choice of prior is a central part of Bayesian modelling and understanding how properties and prior beliefs on the weight space translate to the functions is a major challenge. In the classical Bayesian framework, the prior expresses the beliefs we have about the parameters of the model before seeing any data (the prior beliefs) [Gelman et al., 2013]. The prior should not be simply seen as a probability distribution; in fact, by definition, it only makes sense in the context of the resulting likelihood [Gelman et al., 2017]. Generally, the prior choice depends on the structure of the model, data and training algorithm, and we require priors which are both:

1. Interpretable. For example, we want to be able to specify the hyperparameters of the prior subjectively based on the task at hand.

2. Priors with large support. We do not want a prior that concentrates around a small subset of the parameter space.

3. Lead to feasible inference and favour reasonable approximations of the posterior and predictive distributions.

Neural networks come with high-dimensional weight spaces and interpreting the flow of parameters from the input to the output is usually impossible. The general sensible requirement for the prior on the weights of the neural network is to induce large prior support on a wide class of functions. Research on objective priors in BNNs is very limited [Papamarkou et al., 2022] and since even defining

the concept of the prior beliefs about the weights of neural networks seems problematic, the priors of BNNs are typically non-informative or weakly informative [Arbel et al., 2023, Graves, 2011, Lampinen and Vehtari, 2001]. The rationale of weakly informative priors is to impose some constraints in order to regularize the model and ease computing the posterior [Gelman et al., 2013]. As we will see in Section 1.2.5, estimating a BNN with certain priors is equivalent to optimizing a classical NN under some suitable regularization technique.

In practice, the most common choice of prior on the weights of a Bayesian neural network is the Gaussian distribution. However, there is no clear evidence that Gaussian priors are preferable over other possible choices. In fact, increasing the depth of BNNs with zero mean fixed variance independent Gaussian weights leads to overestimated uncertainty quantification [Ghosh et al., 2019]. This naturally leads to the question of the optimality of such a prior choice. Further, in certain tasks correlated Gaussian priors are known to achieve better performance. The independent Gaussian weights of deep BNNs may cause the cold posteriors effect, that is, the effect when the tempered posteriors with $T < 1$ obtain better posterior predictive performance than the original posterior ($T = 1$) [Wenzel et al., 2020a]. Along the same line, Fortuin [2022] observe that weights of a classical neural network trained with stochastic gradient descent have heavy tails and consider tasks where choosing heavy-tailed priors reduces the cold posterior effect. Further, Peluchetti et al. [2020] show that infinitely wide Bayesian neural networks with heavy-tailed priors do not converge to GPs but to stable processes, which are stochastic processes whose finite-dimensional distributions are multivariate stable distributions. Note that Gaussian distributions are a special case of stable distributions; in this sense, the obtained limit extends the correspondence between GPs and BNNs of Matthews et al. [2018].

A popular alternative to simple Gaussian priors are hierarchical priors; in particular, Normal scale mixtures are known to provide improvement in prediction performance and uncertainty quantification. For certain choices of scale, Gaussian scale mixtures fall into the class of heavy-tailed distributions, classical examples are the Student-t (ST) distribution, which can be seen as a mixture with Inverse-Gamma (IG) scales, and the Laplace distribution, which results from exponential mixing [Polson and Scott, 2010], for illustration see Figure 1.2b. Gaussian scale mixtures are known to be an effective regularization and sparsity-inducing tool in Bayesian modelling [Polson and Sokolov, 2019] and BNNs are no exception, examples include horseshoe distribution [Ghosh et al., 2019], a scale mixture of two Gaussian densities with small and large variances to mimic the classical spike-and-slab prior [Blundell et al., 2015] and mixture prior with Automatic Relevance Determination [Nalisnick et al., 2019]. For further details on sparsity-inducing priors, we refer to Sections 2.4.5 and 4.2.2.

Given the lack of interpretability associated with the weight spaces of neural networks, another line of research focuses on the characteristics of the function space induced by the prior. The equivalence of the BNNs in the infinite-width limit to the GPs provides a good illustration of how such an approach works [Matthews et al., 2018, Rasmussen and Williams, 2005]. Theoretical properties of finite-width BNNs with respect to GPs are less clear. Empirical and theoretical evidence suggests that the hidden units of finite-width BNN with independent

Table 1.2: Some of the challenges of classical and Bayesian neural networks.

Challenge	Classical NN	Bayesian NN
Interpretability	poor	improved
Non-robustness to OOD	yes	improved
Adversarial attacks	sensitive	less sensitive
Overconfidence	typical	less typical
Training outcomes	point estimate	posterior distribution
Choice of prior	no	yes
Initialization	yes	yes

Gaussian weights are dependent [Vladimirova et al., 2021] and get heavier-tailed (i.e. become more non-Gaussian) with the increase of depth [Vladimirova et al., 2019]. We complement the discussion of this and the preceding Sections 1.1.1, 1.1.2 and 1.2.2 to 1.2.4 by the Table 1.2, where we list several key challenges arising in classical and Bayesian neural networks.

1.2.5 Connecting classical and Bayesian perspectives

Many have tackled problems of neural networks in a non-Bayesian way. In this section, we discuss popular regularization techniques used in classical deep models and draw a correspondence between regularization in NNs and enforcing suitable priors in their Bayesian analogues.

Recall that frequentists' maximum likelihood estimates (MLE) of parameters are obtained as $\arg\max p(\mathcal{D} \mid \boldsymbol{\theta})$ and maximum-a-posteriori (MAP) estimates are given by $\arg\max p(\boldsymbol{\theta} \mid \mathcal{D})$. Solutions arising in classical machine learning often have certain probabilistic interpretations and can be derived from Bayesian models [Khan and Rue, 2023]. Obtaining weights in classical NNs boils down to gradient-based minimization of some data error; for example, finding weights that minimize the mean squared error is equivalent to the MLE solution, which can be obtained as the MAP estimate of a BNN with flat priors on the weights (e.g. uniform on the real line).

The simplest techniques for improving generalization and reducing overfitting add some penalty term $\lambda r(\boldsymbol{W})$ to the data error, where $r(\boldsymbol{W})$ is regularization penalty, λ is the weight on the penalty. The ℓ_1- [Tibshirani, 1996] and ℓ_2- (also known as weight decay in the machine learning literature) regularizations add, respectively, ℓ_1- and ℓ_2-norm of the weights [Hinton and van Camp, 1993, MacKay, 1992]. In the context of linear models, this corresponds to Lasso and ridge regressions, for a comprehensive review of Bayesian regularization we refer to [Polson and Sokolov, 2019]. From a probabilistic perspective on neural networks, the data error term corresponds to the negative log-likelihood (NLL) of the training dataset and adding the penalty term is equivalent to placing some prior distribution of the weights [MacKay, 1992]. Indeed, if one places a flat, improper prior on the weights of the neural network (e.g. uniform priors), then the maximum-a-posteriori estimate equals to the maximum likelihood estimate.

Introducing ℓ_1 penalty term to the loss function of the classical neural network is equivalent to placing a Laplacian prior on the weights and obtaining a MAP estimate. In a similar way, ℓ_2-regularization can be seen as a MAP estimate of a neural network with Gaussian prior. Even though introducing weight priors brings some probabilistic flavour to neural networks, point estimates, such as MAP and MLE, do not benefit from marginalization over the posterior and often suffer from overfitting.

Dropout [Srivastava et al., 2014] is another explicit regularization technique, which randomly switches off the nodes of the neural network by adding multiplicative noise (MN) to the input of each layer during the training. From a Bayesian perspective, Gal and Ghahramani [2016] have shown that dropout with Bernoulli noise can be seen as a variational approximation of a deep Gaussian process; Kingma et al. [2015] provided similar analogy between the dropout and BNNs with log-uniform priors. Additionally, Nalisnick et al. [2019] have uncoupled dropout with a specific choice of variational inference algorithm and shown that dropout with MN is equivalent to certain Gaussian scale mixture priors on the weights.

Finally, while the idea of combining the outputs of several neural networks is not novel [Hansen and Salamon, 1990, Levin et al., 1990], deep non-Bayesian ensembles [Huang et al., 2022, Lakshminarayanan et al., 2017] and their relation to the Bayesian framework have received a lot of attention [D' Angelo and Fortuin, 2021, Flöge et al., 2024, Wenzel et al., 2020b, Wild et al., 2023, Wilson and Izmailov, 2020, Wu and A Williamson, 2024]. Wilson and Izmailov [2020] argued that these can be seen as Bayesian model averaging and proposed an improvement of the stochastic gradient descent weight averaging (SWA) by defining Gaussian posterior approximations over neural network weights. Wild et al. [2023] lifted the task of minimizing loss functions onto a space of probability measures and applied Wasserstein gradient flows (WSG) to establish a connection between deep ensembles of non-Bayesian neural networks and variational approximations of BNNs. Pearce et al. [2020] obtained multiple MAP parameter estimates of neural networks and justified that these approximate the true posterior in a Bayesian manner. While the heuristics connecting ensembles of conventional and Bayesian neural networks provide theoretical insights about existing algorithms, it does not yet provide a straightforward recipe for optimal model and inference choice in practice; in fact, naive interpretation of the theory may lead to controversial experimental results [Wild et al., 2023].

1.3 Contributions of the thesis

The thesis is structured as follows:

- Chapter 2 adopts an optimization view on approximate Bayesian inference and focuses on the major variational inference algorithms. We begin with variational algorithms tailored to conditionally conjugate models and gradually build up to reach the so-called black box[3] algorithms that

[3]Throughout the thesis, "black box", that refers to some variational inference algorithm, is

avoid manual, model-specific derivations and well-suited for probabilistic programming frameworks. The chapter then discusses the key challenges associated with variational inference and surveys a range of solutions proposed to deal with those, this includes an extended exploration of strategies to overcome the limitations of the mean-field factorized variational family. Further, the frequentists' theoretical guarantees and the convergence rates of stochastic optimization algorithms to a local optimal are surveyed. Finally, we provide a comprehensive review of variational inference techniques that have been developed and studied in the specific context of Bayesian neural networks, with particular attention being paid to methods designed to enforce sparsity in the weights of the network. The chapter culminates with a proposed taxonomy of variational inference methods, identifying the room for improvement along the branches of this taxonomy.

- Some of the work of Chapter 3 was presented in [Sheinkman and Wade, 2025]. There, we contribute to the empirical investigation of the challenges of classical and Bayesian deep learning and the principal role of the architecture specification in neural networks. In practice, the choice of model and suitable learning procedure is not a straightforward task. We demonstrate that even theoretically profound mathematical algorithm does not automatically render perfect computer implementation. The chapter sheds some light on the behaviour of Bayesian neural networks trained by both sampling and optimization-based inference. Chapter provides a systematic study of accuracy, uncertainty quantification and computational costs in different scenarios including large width and out-of-distribution data. In addition, we investigate the benefits of the predictive model assessment and different model averaging strategies.

- Chapter 4 contains the work which appears in [Sheinkman and Wade, 2024], and advances variational inference in Bayesian neural networks. Specifically, we develop a novel kind of BNNs, termed a variational bow tie neural network with shrinkage (VBNN). These models are designed to address the challenges observed in Chapter 3 and to improve robustness as the network's width and depth increase. We consider a relaxed version of the standard feed-forward rectified neural network with sparsity-promoting priors on the weights and employ Polya-Gamma (PG) data augmentation trick to yield a conditionally linear and Gaussian activations. After constructing the variational bow tie neural network, we derive a variational inference algorithm that avoids assumptions on the distributional form and independence across layers. To improve the scalability of the algorithm, we propose two strategies: a stochastic variant with subsampling and a post-hoc node selection procedure to obtain a sparse posterior, thereby the storage and computational costs of predictions are reduced. Furthermore, to improve accuracy and uncertainty quantification, we consider ensembles of variational approximations obtained by running several parallel variational algorithms with

used without the hyphen to be in agreement with the terminology introduced in [Ranganath et al., 2014].

different initializations. In this way, the approach is capable of accounting for the multimodality of posterior distributions arising in Bayesian neural networks.

- Finally, Chapter 5 presents the concluding discussion, which reflects on the contributions of this thesis and outlines future research directions within the broader context of Bayesian modelling in the era of big data and ever expanding models.

Chapter 2

Variational Inference

Approximate Bayesian inference aims to replace unknown and intractable distributions with some simpler but feasible distributions. Often this is achieved by minimizing certain measures of dissimilarity between the true posteriors and their approximations, examples of such methods are Laplace approximation [MacKay, 1992], loopy belief propagation [Murphy et al., 1999], expectation propagation (EP) [Minka, 2001] and, the central object of this chapter, variational inference (VI) [Jordan et al., 1999].

Tracing back when the term "variational inference" was brought into practice is obscure. The method itself originates in statistical physics [Parisi, 1988] and was used to infer the weights since the early days of Bayesian neural networks (BNNs) [Hinton and van Camp, 1993, Peterson and Anderson, 1987]. We note that in the context of logistic regression, [Jaakkola and Jordan, 1997] used the notions of variational distributions, approximations, parameters and methods, and, finally, the foundational work of [Jordan et al., 1999] has the tools and the terminology we use nowadays including the phrase "variational inference".

This chapter aims to study key variational inference methods from both algorithmic and theoretical perspectives as well as in the context of Bayesian neural networks.

2.1 Inference as optimization

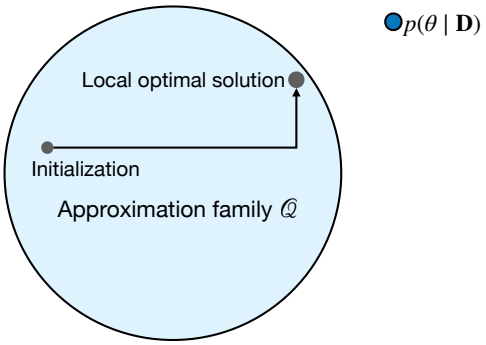
Given a model with joint density $p(\mathcal{D}, \boldsymbol{\theta})$, where observations are denoted by $\mathcal{D}$ and the latent variables by $\boldsymbol{\theta}$, variational inference approximates the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ by

$$q^*(\boldsymbol{\theta}) = \arg\min_{q(\boldsymbol{\theta})\in\mathcal{Q}} \mathrm{D}\left(p(\boldsymbol{\theta}|\mathcal{D}), q(\boldsymbol{\theta})\right), \tag{2.1}$$

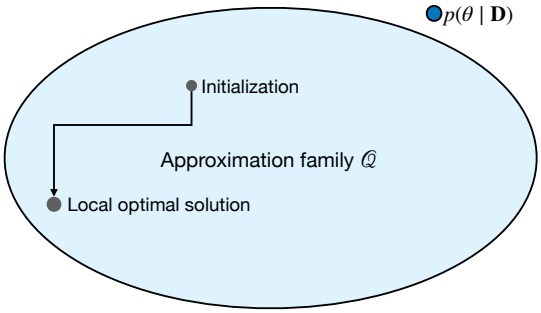
where the approximation of the posterior $q(\boldsymbol{\theta})$ is taken from some family of distributions $\mathcal{Q}$, which we call the variational family, and $\mathrm{D}(\ ,\ )$ measures the discrepancy between the true posteriors and its variational approximation. The function D is called a divergence; it is required to be non-negative and satisfy $D(p, q) = 0$ if and only if $p = q$, but does not need to be symmetric.

The complexity of the optimization task Equation (2.1) is largely determined

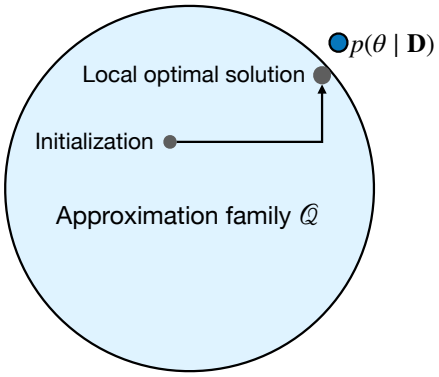
by the complexity of the variational family $\mathcal{Q}$. Choosing a flexible family allows one to get as close as possible to the true posterior density, whilst an overly complex family may not lead to tractable computations. When approximations of probabilistic models are taken from the specified families of distribution, two questions can be raised: (1) Does the chosen family contain an approximation which retains the favourable properties of the true posterior? (2) If yes, can the optimization objective and algorithm select that good candidate? Figure 2.1 illustrates different combinations of "Yes" and "No" answers to the questions above; note, option (b) is far from being optimal but would require more computational resources than option (a), while both (c) and (d) provide plausible approximations of the posterior but (d) requires fewer resources.



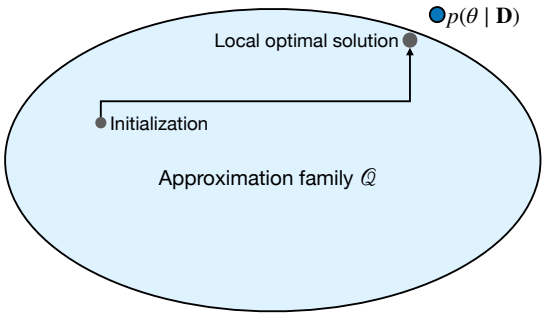
(a) The chosen family does not contain members retaining the properties of the true posterior.



(b) The chosen family contains distributions retaining the properties of the true posterior, but the algorithm picks an unsuitable local optimum.



(c) The chosen family contains distributions retaining the properties of the true posterior and the algorithm picks an suitable local optimum.



(d) The chosen family contains distributions retaining the properties of the true posterior, and the algorithm picks a suitable local optimum at a higher cost than (c).

Figure 2.1: Some of the possible scenarios when approximations of probabilistic models come from the specified families of distribution.

The most common choice is the mean-field family, which assumes that the variational posterior factorizes across mutually independent latent variables (or blocks of latent variables). Suppose $q(\boldsymbol{\theta})$ with $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_J\}$ is characterized by variational parameters $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_J\}$, then a generic element of the mean-field

family $\mathcal{Q}$ has a factorized form

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) = \prod_{j=1}^{J} q_{\lambda_j}(\theta_j). \tag{2.2}$$

Since each variational factor $q_{\lambda_j}(\theta_j)$ of Equation (2.2) is governed by a distinct variational parameter λ_j. The limitations of the mean-field variational family are discussed in Section 2.4.2; we overview the techniques which go beyond the factorized approach in Sections 2.2.3 and 2.4.4.

Further, let the discrepancy between the variational family and true posterior be measured by the reverse Kullback-Leibler (KL) divergence function [Blei et al., 2017] defined as

$$\mathrm{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})) = \mathbb{E}_{q(\boldsymbol{\theta})}[\log q(\boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{\theta}|\mathcal{D})]. \tag{2.3}$$

Note that KL divergence is not symmetric, and the reverse and forward divergences are not equal, that is $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$. To avoid computing unavailable posterior, we express Equation (2.3) as

$$\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})\right) = \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}, \boldsymbol{\theta})\right] + \log p(\mathcal{D}), \tag{2.4}$$

where $p(\mathcal{D})$ is the marginal likelihood after integration of the model parameters, and as such $\log p(\mathcal{D})$ is referred to as the (log) model evidence. We note that in the information-theoretic formulation, the negative of the $\mathbb{E}_{q(\boldsymbol{\theta})}\left[\log q(\boldsymbol{\theta})\right]$ is called the Shannon entropy of $q(\boldsymbol{\theta})$ [MacKay, 2003]. Jensen's inequality provides the lower bound for the evidence known as the evidence lower bound (ELBO):

$$\begin{aligned} \log p(\mathcal{D}) &\geq \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}, \boldsymbol{\theta})\right] - \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log q(\boldsymbol{\theta})\right] \\ &= \mathrm{ELBO}(q). \end{aligned} \tag{2.5}$$

Variational inference has deep roots in statistical physics where the negative of the ELBO is known as the variational free energy [Parisi, 1988]. Comparing Equation (2.4) and Equation (2.5), we find that minimizing the KL divergence is equivalent to maximizing the evidence lower bound, and the variational inference approximates the true posterior by

$$q^*(\boldsymbol{\theta}) = \arg\max_{q \in \mathcal{Q}} \mathrm{ELBO}(q). \tag{2.6}$$

Outline of the chapter. We explore several scenarios for the classes of models and variational families and outline variational inference methods that can be employed in these scenarios. We begin with a variational algorithm suitable for conditionally conjugate models in Section 2.2, we proceed with model-agnostic black box methods in Section 2.3. Further, in Section 2.4 the properties and nuances of the existing algorithms are discussed, where in Sections 2.4.4 to 2.4.6 the focus is made on recent advances, including variational algorithms arising in Bayesian neural networks.

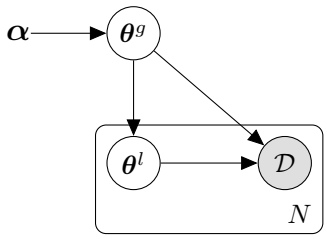


Figure 2.2: Example of a directed acyclic graph (DAG) for a model with global and local variables.

2.2 Conditionally conjugate models

This section focuses on conjugate exponential family models [Brown, 1986, Diaconis and Ylvisaker, 1979, Efron, 2022] and gives an overview of model-specific variational inference methods. In Section 2.2.1, we begin by overviewing exponential family models with local and global variables, Section 2.2.2 proceeds with an algorithm suitable for conditionally conjugate models and mean-field variational families, further, we discuss improvements to scale with large data sets and introduce structural families in Section 2.2.3.

2.2.1 Models with local and global variables

Often statistical models have latent variables specific to every data point, examples include mixture models [Lindsay, 1995], latent Dirichlet allocation (LDA) topic models [Blei, 2012], hierarchical Dirichlet process (HDP) [Liang et al., 2013], hidden Markov models (HMM) [Ghahramani and Jordan, 1995] and models with global and global shrinkage priors [Polson and Scott, 2010]. In such models, latent variables $\boldsymbol{\theta}$ are split into two parts: a vector of global variables $\boldsymbol{\theta}^g$ and a collection of vectors of local variables specific to data points, $\boldsymbol{\theta}^l = \{\theta_1^l, \ldots, \theta_N^l\}$. We illustrate an example of such a model by Figure 2.2. Suppose that $\boldsymbol{\alpha}$ governs $\boldsymbol{\theta}^g$, then the joint distribution of the model is

$$p(\mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}^g|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(\mathcal{D}_n|\theta_n^l, \boldsymbol{\theta}^g) p(\theta_n^l|\boldsymbol{\theta}^g). \tag{2.7}$$

The conditional density of $\boldsymbol{\theta}^g$ given observations and all the other latent variables, namely $p(\boldsymbol{\theta}^g|\mathcal{D}, \theta^l, \boldsymbol{\alpha})$, is known as its complete or full conditional. Note that θ_n^l given $\boldsymbol{\theta}^g, \mathcal{D}_n$ is conditionally independent of $\boldsymbol{\theta}_{-n}^l$ and $\mathcal{D}_{-n}$, so that the complete conditional of local variable θ_n^l equivalent to $p(\theta_n^l|\mathcal{D}_n, \boldsymbol{\theta}^g)$. If the complete conditional is in the same family as the prior then we call the model conditionally conjugate and its likelihood and prior are called conjugate pair. Suppose that the conditional density of $(\mathcal{D}_n, \theta_n^l)$ given $\boldsymbol{\theta}^g$ is in the exponential family, and that the prior on global parameters $\boldsymbol{\theta}^g$ is the corresponding conjugate prior:

$$p(\mathcal{D}_n, \theta_n^l, |\boldsymbol{\theta}^g) = h^l(\mathcal{D}_n, \theta_n^l) \exp\left((\boldsymbol{\theta}^g)^T t^l(\mathcal{D}_n, \theta_n^l) - a^l(\boldsymbol{\theta}^g) \right),$$
$$p_\alpha(\boldsymbol{\theta}^g) = h^g(\boldsymbol{\theta}^g) \exp\left(\boldsymbol{\alpha}^T t^g(\boldsymbol{\theta}^g) - a^g(\boldsymbol{\alpha}) \right),$$

where t^l and t^g are vectors of sufficient statistics, a^l and a^g are log-normalizers, and h^l and h^g are base measures for local and global terms respectively, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \alpha_2)^T$ is the natural parameter of the prior and $t^g(\boldsymbol{\theta}^g) = (\boldsymbol{\theta}^g, -a^l(\boldsymbol{\theta}^g)^T)^T$. The complete conditionals of latent variables are

$$p(\boldsymbol{\theta}^g|\mathcal{D}, \boldsymbol{\theta}^l, \boldsymbol{\alpha}) = h^g(\boldsymbol{\theta}^g) \exp\left(\eta^g\left(\mathcal{D}, \boldsymbol{\theta}^l, \boldsymbol{\alpha}\right) t^g(\boldsymbol{\theta}^g) - a^g(\eta^g(\mathcal{D}, \boldsymbol{\theta}^g, \boldsymbol{\alpha}))\right),$$
$$p(\theta_n^l|\mathcal{D}_n, \boldsymbol{\theta}^g) = h^l(\theta_n^l) \exp\left(\eta^l\left(\mathcal{D}_n, \boldsymbol{\theta}^g\right) t^l(\theta_n^l) - a^l\left(\eta^l\left(\mathcal{D}_n, \boldsymbol{\theta}^g\right)\right)\right).$$

Thus, we have constructed a conditionally conjugate model, which will be the focus of this section.

2.2.2 Coordinate ascent variational inference

Given the mean-field assumption and the optimization objective of Equation (2.6), a direct way of finding $q^*(\boldsymbol{\theta})$ is by differentiating the objective function with respect to each variational factor q_j [Bishop, 2006][4]. This yields the variational update:

$$q^*_{\lambda_j}(\theta_j) \propto \exp\left(\mathbb{E}_{\boldsymbol{\lambda}_{-j}}\left[\log\left(p(\theta_j|\boldsymbol{\theta}_{-j}, \mathcal{D})\right)\right]\right), \tag{2.8}$$

where $\mathbb{E}_{\boldsymbol{\lambda}_{-j}}$ denotes the expectation with respect to $\prod_{i\neq j} q_{\lambda_i}(\theta_i)$. More generally, we simplify the notation, and when considering the expectation with respect to the (component of the) variational density with parameter λ, we often write $\mathbb{E}_\lambda$ instead of the $\mathbb{E}_{q_\lambda}$. Iteratively updating each variational factor by Equation (2.8) whilst keeping the others fixed is guaranteed to climb up the ELBO's local optimum. This procedure is known as coordinate ascent variational inference (CAVI) algorithm. To be able to easily apply CAVI we require models for which the complete conditionals $p(\theta_j|\boldsymbol{\theta}_{-j}, \mathcal{D})$ are available in the closed form and the expectations are tractable. This motivates the choice of conditionally conjugate models in the context of variational inference. Recall the conditionally conjugate model given by Equation (2.7), with such a model, we can consider the following mean-field variational family

$$q(\boldsymbol{\theta}) = q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g) \prod_{n=1}^{N} q_{\lambda_n}(\theta_n^l), \tag{2.9}$$
$$q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g) = h^g(\boldsymbol{\theta}^g) \exp\left(\boldsymbol{\gamma}^T t^g(\boldsymbol{\theta}^g) - a(\boldsymbol{\gamma})\right),$$
$$q_{\lambda_n}(\theta_n^l) = h^l(\theta_n^l) \exp\left(\lambda_n^T t^l(\theta_n^l) - a(\lambda_n)\right),$$

where $q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g)$ and $q_{\lambda_n}(\theta_n^l)$ are in the same exponential families as the model's complete conditionals and come with variational parameters $\boldsymbol{\gamma}$ and λ_n. With the model and variational family at hand, the CAVI algorithm can be employed and given Algorithm 1, where the updates for optimal variational parameters are

[4] Alternatively, we can use the chain rule, consider ELBO as a function of q_{λ_j}, and note that it (up to a constant) is the negative KL divergence between q_{λ_j} and $\log p(\theta_j, \boldsymbol{\theta}_{-j}, \mathcal{D})$ [Blei et al., 2017].

Algorithm 1 Coordinate ascent variational inference

Require: threshold ζ
Initialize variational hyperparameters $\boldsymbol{\gamma}^{(0)}$
  while $\Delta\text{ELBO} > \zeta$ **do**
    for $n \in \{1, ..., N\}$ **do**
      set $\lambda_n^{(t)} = \mathbb{E}_{\boldsymbol{\gamma}^{(t-1)}} \left[\eta^l \left(\mathcal{D}_n, \boldsymbol{\theta}^g\right)\right]$
    end for
    Update $\boldsymbol{\gamma}^{(t)} = \mathbb{E}_{\boldsymbol{\lambda}^{(t)}} \left[\eta^g \left(\mathcal{D}, \boldsymbol{\theta}^g, \alpha\right)\right]$
  end while
Ensure: variational posterior q

obtained in the closed form using Equation (2.8)

$$\lambda_n^* = \mathbb{E}_{\boldsymbol{\gamma}} \left[\eta^l \left(\mathcal{D}_n, \boldsymbol{\theta}^g\right)\right], \tag{2.10}$$

$$\boldsymbol{\gamma}^* = \mathbb{E}_{\boldsymbol{\lambda}} \left[\eta^g \left(\mathcal{D}, \boldsymbol{\theta}^g, \alpha\right)\right]. \tag{2.11}$$

The sum Equation (2.11) goes through each data point implying that the CAVI algorithm does not scale very well when applied to models with many local variables. This brings us to the next section where we scale variational inference to large data. [Neal and Hinton, 1998] stated the formulation of the expectation-maximization (EM) algorithm [Dempster et al., 1977] in terms of the statistical physics and the variational free energy [Parisi, 1988]; such formulation implies that EM can be seen as a version of variational inference with certain s simplifying assumptions (namely, delta function variational posterior providing a point estimate of $\boldsymbol{\theta}^g$ in the M-step, a flat prior on $\boldsymbol{\theta}^g$ and update $q(\boldsymbol{\theta}^l) \propto p(\boldsymbol{\theta}^l|\mathcal{D}, \boldsymbol{\theta}^g)$ in the E-step).

2.2.3 Stochastic variational inference.

To update the parameters of the variational family in Equation (2.9) using Equations (2.10) and (2.11), the CAVI has to cycle through the entire, potentially, very large dataset and, thus, becomes computationally expensive and inefficient. Instead of the classical coordinate ascent, the maximum of the ELBO could be reached in a gradient-based optimization, this leads to a more efficient and scalable method known as stochastic variational inference (SVI) [Hoffman et al., 2013]. Instead of simply sweeping through the entire dataset, SVI utilizes stochastic optimization with noisy natural gradients of the ELBO. Note that compared to the classical Euclidean gradient, the natural one captures the geometry of the probability parameters and thus, benefits from faster convergence of the optimization algorithm [Amari, 1998]. Moreover, computing natural Riemannian gradients of the ELBO with respect to variational parameters corresponding to posteriors from the exponential family accounts to computing the coordinate updates and subtracting the previous settings of parameters [Hoffman et al., 2013, Sato, 2001]. We begin by uniformly sampling an index $s \sim \text{Uniform}(1, \ldots, N)$, computing the local variation parameter λ_s and setting the intermediate global parameter $\hat{\boldsymbol{\gamma}}$ to be the ordinary coordinate ascent update, but where $\mathcal{D}$ is re-

placed with a dataset $\mathcal{D}_s^{(N)}$ formed of N replicates of the sample $\mathcal{D}_s$. Then, a noisy estimate of the natural gradient of ELBO with respect to global variational parameters $\boldsymbol{\gamma}$ is given by

$$\begin{aligned}\widehat{\nabla}_{\boldsymbol{\gamma}} \operatorname{ELBO}(q) &= \mathbb{E}_q \left[\eta^g \left(\left(\boldsymbol{\theta}_s^l\right)^{(N)}, \mathcal{D}_s^{(N)}, \alpha \right) \right] - \boldsymbol{\gamma} \\ &= \boldsymbol{\alpha} + N \left(\mathbb{E}_{\lambda_s} \left[t(\boldsymbol{\theta}_s^l, \mathcal{D}_s) \right], 1 \right)^T - \boldsymbol{\gamma} \\ &= \hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}.\end{aligned}$$

In other words, above we found a rather simple expression for an unbiased estimate of the natural gradient of the ELBO, i.e. $\mathbb{E}_q \left[\widehat{\nabla}_{\boldsymbol{\gamma}} \operatorname{ELBO}(q) \right] = \nabla \boldsymbol{\gamma} \operatorname{ELBO}(q)$, yet it is much cheaper to compute through subsampling. Given the step size sequence ρ_t satisfying the Robbins and Monro conditions of Equation (2.12), the stochastic gradient descent (SGD) algorithm is capable of bringing ELBO to a local maximum [Robbins and Monro, 1951]. When the classical gradient is replaced with the natural one, the algorithm follows realizations $h_t(\boldsymbol{\gamma})$ of $\widehat{\nabla}_{\boldsymbol{\gamma}} \operatorname{ELBO}(q)$ and sets the update parameters to be

$$\boldsymbol{\gamma}^{(t)} = \boldsymbol{\gamma}^{(t-1)} + \rho_t h_t(\boldsymbol{\gamma}^{(t-1)}),$$

where

$$h_t(\boldsymbol{\gamma}^{(t-1)}) = \hat{\boldsymbol{\gamma}}^t - \boldsymbol{\gamma}^{(t-1)},$$

and

$$\sum \rho_t = \infty, \quad \sum \rho_t^2 < \infty. \tag{2.12}$$

Thus, the variational update of the global parameter is

$$\boldsymbol{\gamma}^{(t)} = (1 - \rho_t)\boldsymbol{\gamma}^{(t-1)} + \rho_t \widehat{\boldsymbol{\gamma}}^t. \tag{2.13}$$

The SVI procedure is outlined in the Algorithm 2, where choosing the sequence ρ_t to be adaptive outperforms a preset learning-rate [Ranganath et al., 2013]. The stochastic optimization algorithm can be improved by using mini-batches, and as long as the size of the mini-batch S satisfies $S \ll N$, the computational savings are obtained. For each point $\mathcal{D}_s$ of the mini-batch, the intermediate global variational parameters $\widehat{\boldsymbol{\gamma}}_s$ are obtained to give the re-scaled update of the global parameter:

$$\boldsymbol{\gamma}^{(t)} = (1 - \rho_t)\boldsymbol{\gamma}^{(t)} + \frac{\rho_t}{S} \sum_{s \in S^t} \widehat{\boldsymbol{\gamma}}_s^t,$$

where S denotes the indices of the data points in the mini-batch. To obtain a more flexible approximation and ease some of the restrictions of the mean-field family, one could allow the dependency between the latent variables [Barber and Wiegerinck, 1998] or add further variables to the family [Bishop et al., 1998, Jaakkola and Jordan, 1998]. Variational inference in this case is often described as

Algorithm 2 Stochastic variational inference

Require: Step size sequence ρ_t, threshold ζ for $\boldsymbol{\lambda}$
Initialize variational hyperparameters $\boldsymbol{\gamma}^{(0)}$
while $\Delta\boldsymbol{\lambda} > \zeta$ **do**
Sample an index $s \sim \text{Uniform}(1, .., N)$ for a data point $\mathcal{D}_s$.

Compute local parameter $\boldsymbol{\lambda} = \mathbb{E}_{\boldsymbol{\gamma}^{(t-1)}}\left[\eta^l\left(\boldsymbol{\theta}^g, \mathcal{D}_s\right)\right]$

Find $\widehat{\boldsymbol{\gamma}}^t = \boldsymbol{\alpha} + N\left(\mathbb{E}_{\lambda_s^{(t)}}\left[t(\theta_s^l, \mathcal{D}_s)\right], 1\right)^T$

Update global variational parameter $\boldsymbol{\gamma}^{(t)} = (1-\rho_t)\boldsymbol{\gamma}^{(t-1)} + \rho_t\widehat{\boldsymbol{\gamma}}^t$
end while
Ensure: variational posterior $q_{\boldsymbol{\gamma}}$

structured (or sometimes structural) [Barber, 2012]. Assuming the conditionally conjugate model of Equation (2.7), one can define the variational family which has the dependencies between the global $\boldsymbol{\theta}^g$ and local $\boldsymbol{\theta}^l$ variables:

$$q(\boldsymbol{\theta}) = q_{\boldsymbol{\gamma}}(\theta_m^g)\prod_{n=1}^{N} q_{\lambda_n}(\theta_n^l|\boldsymbol{\theta}^g).$$

Then the SVI of the Algorithm 2 can be extended to its structured version (SSVI) [Hoffman and Blei, 2015]. In Chapter 4 we will encounter an example of structured variational inference and derive an extended version of the CAVI algorithm.

2.3 Black box variational inference.

While the SVI algorithm scales up the CAVI algorithm to large data, it still requires model-specific derivations of variational updates. In this sense, both algorithms are far from automatic and limit variational inference to the class of conditionally conjugate models, where all complete conditionals are in the exponential family and one can analytically obtain ELBO. Real-world problems, especially those arising in Bayesian deep learning, bring up many different models which are not conditionally conjugate and do not have a tractable ELBO. And even if the models are conditionally conjugate, the requirement for tedious and manual derivations prevented VI from becoming widely applied and popular. This brings one to a black box variational inference (BBVI), a class of methods which do not require the optimization objective given by Equation (2.6) to be analytically tractable, and which promise to avoid limitations of the conditionally conjugate family, model-specific derivations. We begin this section by presenting a VI algorithm suitable for models with evaluable log-likelihoods Section 2.3.1, then in Section 2.3.2 we consider algorithms for models with differentiable log-likelihoods, finally, the most commonly used in probabilistic programming black box variational inference is discussed in Section 2.3.

Algorithm 3 Black box variational inference (score)

Require: Step size sequence ρ_t, threshold ζ for $\boldsymbol{\lambda}$
Initialize variational hyperparameters $\boldsymbol{\lambda}^{(0)}$
  while $\Delta\boldsymbol{\lambda} > \zeta$ **do**
    Draw S samples $\boldsymbol{\theta}_s \sim q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ for $s = 1, \ldots, S$
    $\boldsymbol{\lambda}^{(t)} = \boldsymbol{\lambda}^{(t-1)} + \rho_t \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s)(\log p(\mathcal{D}, \boldsymbol{\theta}_s) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s))$
  end while
Ensure: variational posterior q

2.3.1 Score gradient

We wish to obtain an unbiased gradient estimator of the ELBO having minimum assumptions and in the general modelling settings. Then the unbiased gradient estimates of the exact gradient can be used in the stochastic optimization algorithm to obtain the variational parameters [Kingma and Welling, 2014, Ranganath et al., 2014]. Specifically, let $p(\mathcal{D}, \boldsymbol{\theta})$ be some generic probabilistic model, and $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ be the variational distribution. The log-derivative identity and the fact that the expectation of the score function $\nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ with respect to any q is zero, allow rewriting $\nabla_{\boldsymbol{\lambda}} \mathrm{ELBO}(q)$:

$$\nabla_{\boldsymbol{\lambda}} \mathrm{ELBO}(q) = \mathbb{E}_{q_{\boldsymbol{\lambda}}} \left[\nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) \left(\log p(\mathcal{D}, \boldsymbol{\theta}) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) \right) \right].$$

The Monte Carlo integration gives the score function gradient estimator (also known as REINFORCE gradient estimator) [Ranganath et al., 2014]:

$$\widehat{\nabla}_{\boldsymbol{\gamma}} \mathrm{ELBO}(q) = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s)(\log p(\mathcal{D}, \boldsymbol{\theta}_s) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s)), \tag{2.14}$$

where $\boldsymbol{\theta}_s \sim q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ for $s = 1, \ldots, S$ and we require to be able to

1. Sample from the variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$.
2. Evaluate the score function $\nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$.
3. Evaluate $\log p(\mathcal{D}, \boldsymbol{\theta}) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$.

Note that the requirements for the model and the variational posterior made above allow the BBVI with score gradient to be applied when dealing with both continuous and discrete random variables. The Equation (2.14) together with the stochastic gradient descent lead to Algorithm 3. Unfortunately, the gradient of the score function in the Monte Carlo estimator of Equation (2.14) would most likely have high variance, resulting in poor posterior approximation or an impractically long time needed to converge. To reduce the variance, [Ranganath et al., 2014] proposes Rao-Blackwellization and using the score function control variates. In practice, however, control variates are not always available (e.g. univariate cases [Kucukelbir et al., 2017]) and a good choice is model-dependent. In the next section, we consider the second approach to expressing and assessing the gradient of ELBO which generally exhibits lower variance and under some

assumptions is preferable over the score gradient [Kingma and Welling, 2014, Rezende et al., 2014, Titsias and Lázaro-Gredilla, 2015].

2.3.2 Reparametrization gradient

When applying the score gradient, one needs to be able to sample from the variational distribution and evaluate the model's log-likelihood, the logarithm of the variational approximation and its gradient with respect to the variational parameter. Here we restrict our attention to models and families that are log-differentiable with respect to the latent variables and consider another way of expressing the gradient as an expectation. The approach implements the reparametrization trick [Kingma and Welling, 2014] also known as coordinate [Rezende et al., 2014] and invertable [Titsias and Lázaro-Gredilla, 2015] transformation. The idea of the trick is to standardize the variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ by expressing $\boldsymbol{\theta}$ as a deterministic transformation of a random noise. We introduce an auxiliary random variable $\boldsymbol{\epsilon} \sim r(\boldsymbol{\epsilon})$ and a differentiable transformation $\boldsymbol{\theta} = t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon})$, so that variational parameters are absorbed and become a part of transformation but not the noise ϵ. Suppose for example, that $q_{(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the suitable reparametrization is

$$\begin{aligned} \boldsymbol{\theta} &= t_{(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\boldsymbol{\epsilon}), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), \\ t_{(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\boldsymbol{\epsilon}) &= \boldsymbol{\mu} + |\boldsymbol{\Sigma}|^{\frac{1}{2}}\boldsymbol{\epsilon}. \end{aligned}$$

A natural question is whether the class of latent variables and variational families suited to the reparametrization trick is broad enough. Addressing this, [Kingma and Welling, 2014] provide explicit differentiable transformation and auxiliary variables for variational distributions $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ that have tractable inverse CDFs or come from the location-scale family. The reparametrization, or so-called pathwise gradient, is

$$\nabla_{\boldsymbol{\lambda}} \operatorname{ELBO}(q) = \mathbb{E}_{r(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{\theta}} \left(\log p(\mathcal{D}, \boldsymbol{\theta}) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) \right) \nabla_{\boldsymbol{\lambda}} t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon}) \right].$$

The Monte Carlo approximation gives the reparametrization gradient estimator:

$$\widehat{\nabla}_{\boldsymbol{\gamma}} \operatorname{ELBO}(q) = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\theta}} \left(\log p(\mathcal{D}, \boldsymbol{\theta}_s) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s) \right) \nabla_{\boldsymbol{\lambda}} t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon}_s), \tag{2.15}$$

where $\boldsymbol{\epsilon}_s \sim r(\boldsymbol{\epsilon}), \boldsymbol{\theta}_s = t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon}_s)$ and we require to be able to

1. Transform $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ so that $\boldsymbol{\theta} = t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon})$.
2. Differentiate $\log p(\mathcal{D}, \boldsymbol{\theta}) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

Similar to Section 2.3.2, we utilize stochastic gradient descent and obtain the BBVI with reparametrization gradient in Algorithm 4. The reparametrization gradient (when available) has a better-behaved variance than the score gradient and thus, is preferable. The intuition behind this is based on the observation that the gradient of the model's joint probability gradient is better informed about the

Algorithm 4 Black box variational inference (reparametrization)

Require: Step size sequence ρ_t, threshold ζ for ELBO
Initialize variational hyperparameters $\boldsymbol{\lambda}^{(0)}$
  while $\Delta\text{ELBO} > \zeta$ **do**
    Draw the noise, calculate the latent variables:
    $\boldsymbol{\epsilon}_s \sim r(\boldsymbol{\epsilon}), \boldsymbol{\theta}_s = t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon}_s)$ for $s = 1, \ldots, S$
    $\boldsymbol{\lambda}^{(t)} = \boldsymbol{\lambda}^{(t-1)} + \rho_t \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\theta}} \left(\log p(\mathcal{D}, \boldsymbol{\theta}_s) - \log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_s)\right) \nabla_{\boldsymbol{\lambda}} t_{\boldsymbol{\lambda}}(\boldsymbol{\epsilon}_s)$
  end while
Ensure: variational posterior q

direction of the maximum posterior mode than the gradient of the score function. Further, note that the gradients with respect to $\boldsymbol{\theta}$ allow backpropagation and can be computed more efficiently than gradients with respect to $\boldsymbol{\lambda}$. The drawback of the reparametrization gradient is that it limits the choice of model and family to $\log p(\mathcal{D}, \boldsymbol{\theta})$ and $\log q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ which are differentiable with respect to $\boldsymbol{\theta}$ and cannot be applied to models with discrete random variables. Nevertheless, neither approach requires the optimization objective given by Equation (2.6) to be analytically tractable and, thus, can be described as a black box method.

2.3.3 Automatic and black box

Suppose that the model $p(\mathcal{D}, \boldsymbol{\theta}^o)$ is differentiable and has continuous parameters $\boldsymbol{\theta}^o$ ($\mathcal{D}$ might be discrete), and we wish to find a suitable variational posterior $q_{\boldsymbol{\lambda}^o}(\boldsymbol{\theta}^o)$. To be able to choose a variational family independent of the model, we transform the support of the prior so that the latent variables are now members of the real space. Let T be a differentiable isomorphic transformation, then

$$
\begin{aligned}
&T : \operatorname{supp}(p(\boldsymbol{\theta}^o)) \to \mathbb{R}^k, \\
&T : \boldsymbol{\theta}^o \mapsto \boldsymbol{\theta}, \quad \boldsymbol{\lambda}^o \mapsto \boldsymbol{\lambda}, \\
&p(\mathcal{D}, \boldsymbol{\theta}) = p(D, T^{-1}(\boldsymbol{\theta}))|\det J_{T^{-1}}(\boldsymbol{\theta})|.
\end{aligned}
$$

Since the latent variables now live in $\mathbb{R}^k$, the variational family can be chosen to be Gaussian, either a mean-field with a diagonal covariance matrix or a full-rank with a positive semi-definite matrix containing off-diagonal elements. In this case, we are able to apply the BBVI with the reparametrization gradient Algorithm 4, find the variational posterior in the real space $q_{\boldsymbol{\lambda}}(T(\boldsymbol{\theta}^o))$, and the inverse of the transformation T returns the posterior back to the original latent variables:

$$
q_{\boldsymbol{\lambda}^o}(\boldsymbol{\theta}^o) = q_{\boldsymbol{\lambda}}(T(\boldsymbol{\theta}^o))|\det J_T(\boldsymbol{\theta}^o)|.
$$

The method described above leads to a practical realization of variational inference called automatic differentiation variational inference (ADVI) [Kucukelbir et al., 2017], where the manual computation of the reparametrization gradient is avoided by using automatic differentiation. In black box VI, stochastic gradient optimization of the intractable objective can be replaced with deterministic optimization of the sample average approximation of that objective [Burroni et al.,

2024]; further, combined with the transformation of the model's parameters, this results in a faster method called deterministic ADVI (DADVI) [Giordano et al., 2024]. In practice, ADVI is, perhaps, one of the most widely used variational inference methods, and has been implemented in several probabilistic programming (PPL) frameworks such as Stan [Carpenter et al., 2017], Pyro (NumPyro) [Phan et al., 2019], Turing [Ge et al., 2018], Tensorflow Probability [Dillon et al., 2017] and PyMC3 [Salvatier et al., 2016]. Further, in Chapter 3, where we study the empirical performance of Bayesian neural networks, we implement the ADVI with the mean-field Gaussian family.

2.4 When, why and how of variational inference

Section 2.4.1 provides a structured summary of the variational inference algorithms that have been introduced in previous sections. We then consider limitations of VI and means of surpassing those in Section 2.4.2. Further, theoretical aspects and frequentist properties are discussed in Section 2.4.3, and an important line of research on extending classes of variational families is considered in Section 2.4.4. Finally, we discuss variational inference in the context of Bayesian neural networks in Sections 2.4.5 and 2.4.6.

2.4.1 Overview

We began this chapter by formulating the task of approximate Bayesian inference through the lenses of an optimization objective (as Equation (2.1)). Then we decided to measure the dissimilarity between the approximation and the true posterior with the Kullback-Leibler divergence and focused on the mean-field variational family with an exception in Section 2.2.3, where we very briefly mentioned the structural variational inference, and in Section 2.3.3, where full covariance matrix of the multivariate Normal was allowed (note, this can be seen as a particular case of structural VI). In Section 2.2.2, we restricted our attention to the conditionally conjugate exponential models for which the closed-form updates of the coordinate ascent variational inference are available. We then observed that each iteration of the CAVI algorithm cycles through every point of the dataset, which leads to poor scalability and computational burdens. In contrast to simple coordinate descent, stochastic variational inference of Section 2.2.3 utilizes stochastic gradients and mini-batches, and was introduced as a scalable extension of simple coordinate descent. Both CAVI and SVI are limited to conditionally conjugate models and require tedious model-specific derivations, easing these requirements brings us up to the next level of variational inference known as black box VI. Section 2.3.1 introduced BBVI with the score gradient, which can be applied to all evaluable models but suffers from high variance. In Section 2.3, BBVI with the reparametrization gradient is proposed, which limits the class of models to differentiable but exhibits much lower variance than the approach with the score gradient. With BBVI, differentiable models and the toolbox of automatic differentiation at hand, variational inference becomes a part of all major probabilistic programming languages discussed in Section 2.3.3. For the reader's convenience,

the overview above is supplemented with Table 2.1.

Table 2.1: Variational inference algorithms and their location in this chapter.

Name (acronym)	Where introduced	Algorithm	Model-agnostic
CAVI	Section 2.2.2	Algorithm 1	no
SVI	Section 2.2.3	Algorithm 2	no
BBVI (score)	Section 2.3.1	Algorithm 3	yes
BBVI (reparametrization)	Section 2.3.2	Algorithm 4	yes

To summarize, in the previous section, we introduced several variational inference algorithms (Algorithms 1 to 4), this section aims to motivate the use of those algorithms as well as shed some light on the challenges associated with the variational approximations.

2.4.2 Caveats and how to avoid them

One of the major and extensively studied drawbacks of variational inference is associated with the commonly used factorized family. Namely, the mean-field variational posterior explicitly ignores correlations between variables corresponding to different components (factors) of the variational family; and even when the posterior means are well-estimated, the marginal variances tend to be underestimated [MacKay, 2003, Turner and Sahani, 2011, Wainwright and Jordan, 2008, Wang and Titterington, 2004b]. Several potential solutions to the above challenge were proposed within mean-field variational inference. For instance, implementing linear response covariance estimates allows for approximations closely matching Markov chain Monte Carlo (MCMC) but obtained in significantly less time [Giordano et al., 2018, Raymond and Ricci-Tersenghi, 2017]. Alternatively, assessing and adjusting mean-field approximation using Pareto smoothed importance sampling leads to better mean and variance estimates [Yao et al., 2018b]. A natural approach to overcome the limitations of the factorized family is to consider more expressive approximation families which we discuss in Section 2.4.4, nevertheless we mention here the Thouless-Anderson-Palmer correction [Fan et al., 2021] and entropic regularization [Wu and Blei, 2024] as methods which do not explicitly change the variational family but rather alter the optimization objective to improve uncertainty quantification.

Another question often arising in variational inference is the optimality of the chosen reverse KL divergence as a measure of the similarity between the true posterior and the variational approximation. For example, it is possible to obtain small values of the reverse KL divergence and arbitrary large errors in the posterior mean and variance [Huggins et al., 2020]. Additionally, the recently derived impossibility theorem states that assuming that the true posterior is a Gaussian with a non-diagonal covariance and the variational approximation is Gaussian with diagonal covariance, the minimizer of the reverse KL divergence can accurately approximate at most one of the three measures of uncertainty: diagonal elements of the covariance (marginal variance), diagonal elements of the inverse of the covariance (marginal precision) or determinant of the covariance

(generalized variance). In fact, with the above assumptions on the family and model, and the task of approximating marginal variances, the forward KL divergence, which arises in the expectation-propagation algorithm [Minka, 2001], would be preferable over its reverse version [Margossian et al., 2024], but is more computationally challenging [Vehtari et al., 2020].

In contrast, assuming the elliptical symmetry of the true posterior and the location-scale family, the variational approximation is a global minimizer of the reverse KL divergence and exactly recovers the mean and the correlation matrix [Margossian and Saul, 2024]. Additionally, [Dhaka et al., 2021] studied the performance of the BBVI with different divergences, mean-field and normalizing flow variational families empirically and in the pre-asymptotic regime, and confirmed that when approximating high-dimensional posteriors, the reverse KL divergence is preferable over the forward KL, χ^2 ad f divergences.

Evaluation of the quality and convergence monitoring of variational inference is not a straightforward task since the scale of the objective function, the ELBO, is hardly interpretable. Widely used in practice ADVI propagates random variables through the transformation and reparametrization implying that comparing two different algorithm runs based on ELBO may not be sensible. Several robust diagnostics and frameworks were proposed (but not widely implemented) including error bounds based on the Wasserstein distance [Huggins et al., 2020], shape parameter of the Pareto smoothed importance sampling [Dhaka et al., 2021, Yao et al., 2018b] and symmetrized Kullback–Leibler divergence [Welandawe et al., 2024].

More challenges arise with the emergence of deeper models and larger datasets; nuances of performing approximate Bayesian inference (including VI) in the context of deep learning and multi-modal posteriors are further considered in Chapter 3.

2.4.3 Asymptotic guarantees and convergence rates

Even with the recent developments in Markov chain Monte Carlo, sampling-based algorithms may come with asymptotic guarantees but converge too slowly for the era of big data and highly parametrized models (for an overview of advances in the MCMC we refer to [Angelino et al., 2016], and for classical results to [Robert et al., 1999]). As a scalable alternative to sampling-based methods, optimization-based variational inference has been successful in many applications including probabilistic topic modelling [Blei, 2012], genetic studies [Carbonetto and Stephens, 2012], computational neuroscience [Flandin and Penny, 2007], speech-recognition [Reyes-Gomez et al., 2004] and image segmentation [Du et al., 2009]. Further, developments in black box variational techniques together with the strengths of probabilistic programming languages made variational inference an even more scalable, generic method. Indeed, mean-field automatic differentiation variational inference is able to handle large datasets and yield good predictive performance in tasks where MCMC becomes computationally intractable; one of the first particularly prominent examples is the analysis of 1.7 million of taxi trajectories of [Kucukelbir et al., 2017].

Rigorous theoretical analysis of the variational inference is limited. In

this section, we outline two angles from which the theoretical justification for adopting variational inference can be obtained. The first approach studies statistical (frequentists) properties of the variational posteriors, and the second focuses on the convergence of the algorithms to the optimum of the objective function.

From the frequentists side, the contraction rates of variational posteriors of several particular models were derived, including sparse high-dimensional linear regressions [Ray and Szabó, 2022], logistic regressions [Ray et al., 2020], neural networks with heavy-tailed priors on the weights [Castillo and Egels, 2024], mixture models [Chérief-Abdellatif and Alquier, 2018], and sparse deep neural networks [Chérief-Abdellatif, 2020]. Asymptotical properties of variational distributions were studied in exponential family models [Wang and Titterington, 2004a], stochastic block models [Celisse et al., 2012], generalized linear mixed models [Hall et al., 2011], and Gaussian mixture models [Westling and McCormick, 2019]. In a broader context, [Zhang and Gao, 2020] extend classical conditions for contraction rates of posteriors to variational inference (for the classical conditions we refer to [Ghosal et al., 2000]). On the same more general line, the contraction rates and risk bounds for the variational fractional posteriors (raised to some fraction power of α for $\alpha \in (0, 1]$) were derived [Alquier and Ridgway, 2020, Yang et al., 2020]. Coming from a slightly different perspective, [Wang and Blei, 2019a,b] extend the Bernstein–von Mises theorem to variational posteriors in both well-specified and misspecified models and as a consequence establish consistency and asymptotic normality of Gaussian variational posteriors. Studying posterior predictive distributions induced by the variational posterior in misspecified models in the light of the variational Bernstein–von Mises theorem shows that the misspecification error outweighs the variational approximation error. This observation leads to an important conclusion that in specific prediction tasks specifying the right model is more important than fixing the variational approximation error. We study the crucial importance of model choice in Chapter 3. Recently, the Bernstein–von Mises theorem was established for mean-field variational inference with enthropic regularization which was shown to be consistent and asymptotically normal [Wu and Blei, 2024].

The results above provide some a priori guarantees given that the variational algorithm reaches some global optimum of the objective function (minimum of the KL divergence or maximum of ELBO). However, ELBO is typically highly non-convex, and while CAVI is guaranteed to reach a local minimum [Jordan et al., 1999], both SVI and BBVI employ stochastic methods which cannot offer such guarantees and can suffer from high variance. Thus, an important research direction focuses on the structure of the variational optimization problem and the convergence of stochastic algorithms. Proving the convergence of the optimization algorithm for a non-convex objective is challenging and usually requires the optimization objective, in our case that is the ELBO, to be smooth. Further, the gradient noise of the most commonly used reparametrization gradient is not easily controlled because it depends on the variational parameters in a non-trivial way [Domke, 2020]. Only recently, guarantees for BBVI were obtained in the case of

proximal and projected stochastic gradient descents, within the full-rank Gaussian variational family, and log-concave and log-Lipschitz model; in addition to the guarantees above, a novel type of gradient estimator called "sticking the landing" estimator was shown to have lower noise and, thus, faster convergence than the estimators commonly used in BBVI [Domke et al., 2024]. Simultaneously, the convergence guarantees for the BBVI with regular SGD were provided within the location-scale family and for log-smooth posterior densities [Kim et al., 2024b]. Further, recall that assuming the elliptical symmetry of the true posterior and the location-scale family, the variational approximation exactly recovers the mean and the correlation matrix [Margossian and Saul, 2024]. Combining this result with convergence guarantees for BBVI, we get theoretically grounded guidance on when variational inference might be particularly accurate and scalable. Further study of gradient estimators (including the novel "sticking the landing" estimator) together with the benefits of the projected SGDs allowed for an improved version of previous convergence results [Kim et al., 2024a]. Lastly, ongoing research confirms that the full-rank variational family's struggle to converge could be tackled by adopting the structured variational family [Ko et al., 2024].

2.4.4 Beyond the mean-field variational family

In Section 2.2.3 we have seen structured variational approximations which replaced the fully factorized family with a family which factorized across subsets of variables. While that approach introduced dependencies between the latent variables, in the context of black box and automatic differentiation variational inference, it would still be restricted to models with continuous latent variables and would not scale well [Giordano et al., 2024]. There are other methods of adding more structure to the factorized family and going beyond continuous latent variables, examples include variational boosting [Miller et al., 2017], mixture models [Bishop et al., 1998, Hotti et al., 2024, Jaakkola and Jordan, 1998], normalizing flows [Dhaka et al., 2021, Rezende and Mohamed, 2015] and Copula VI [Tran et al., 2015]. One can obtain more flexible and expressive approximations by treating the variational family as a statistical model of its own. We omit the details but briefly mention here one of the generic VI methods applicable in the black box scenario, which allows for more expressive families and discrete latent variables, namely hierarchical variational models (HVM) [Ranganath et al., 2016]. In probabilistic modelling, the dependencies are often introduced in a hierarchical way [Gelman et al., 2013], and a similar approach can be taken when extending the variational family. Hierarchical variational models consist of two levels: the underlying family with a prior is placed on top. Given the variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$ of Equation (2.2), one can introduce the so-called variational prior $q_{\boldsymbol{\xi}}(\boldsymbol{\lambda})$ with $\boldsymbol{\xi}$ being then a variational hyperparameter. Then, the variational posterior of interest is obtained through marginalization:

$$q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \int q_{\boldsymbol{\xi}}(\boldsymbol{\lambda}) \prod_{j=1}^{J} q_{\lambda_j}(\boldsymbol{\theta}_j). \tag{2.16}$$

Choosing the variational prior $q_{\boldsymbol{\xi}}(\boldsymbol{\lambda})$ that does not have the same factorization structure as the original $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$, leads to families with more elaborate distributions. Marginalization implies that HVMs allow both continuous and discrete latent variables to be used within the black box variational inference framework. The method is very general since the variational prior can be chosen from a variety of options, depending on the model assumptions, a sensible choice could be a mixture of Gaussians, a normalizing flow or a copula.

Note, however, that despite several promising approaches, expanding variational families does not necessarily lead to better posterior estimates and when done naively comes at an unreasonably high cost [Giordano et al., 2024, Turner and Sahani, 2011]. Instead of taking a step outside of the mean-field variational family, a line of research focuses on improving the scalability of VI by tightening the area of search inside the factorised family. Suppose that probability model comes with global and local variables, that is $p(\mathcal{D}, \boldsymbol{\theta}) = p(\mathcal{D}, \boldsymbol{\theta}^g, \boldsymbol{\theta}^l)$, where $\boldsymbol{\theta}^g$ and $\boldsymbol{\theta}^l$ are, respectively, global and local variables (for more details see Section 2.2.1). The mean-field variational family would then assume

$$q(\boldsymbol{\theta}) = q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g) \prod_{n=1}^{N} q_{\lambda_n}(\theta_n^l),$$

$q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g)$ and $q_{\lambda_n}(\theta_n^l)$ come with variational parameters $\boldsymbol{\gamma}$ and λ_n. Given high-dimensional models and large datasets, CAVI becomes computationally inefficient and is often replaced with SVI outlined Section 2.2.3. Alternatively, instead of using SVI within the total mean-field family, one could explore the amortized version of VI which posits an amortized variational family [Agrawal and Domke, 2021, Margossian and Blei, 2024, Rezende et al., 2014]:

$$q(\boldsymbol{\theta}) = q_{\boldsymbol{\gamma}}(\boldsymbol{\theta}^g) \prod_{n=1}^{N} q(\theta_n^l, f_\psi(\mathcal{D}_n)), \tag{2.17}$$

where f_ψ is the inference function that maps the data point $\mathcal{D}_n$ to the parameter $\lambda_n = f_\psi(\mathcal{D}_n)$ of the approximate variational posterior. Most commonly, f_ψ is a deep neural network and, thus, called an inference network. In this way, one function f_ψ replaces all local variational parameters λ_n, and inference is amortized. Typically, amortized inference arises in the context of variational auto-encoders (VAEs), where $p(\mathcal{D}, \boldsymbol{\theta}^l, \boldsymbol{\theta}^g)$ is a deep generative model with prior $p(\boldsymbol{\theta}^l)$ (where θ_n^l represent the latent variables encoding the data points) and some global parameters $\boldsymbol{\theta}^g$ (weights and biases of the decoder). An inference network f_ψ is called an encoder and amortized VI approximates $p(\boldsymbol{\theta}^l|\mathcal{D}, \boldsymbol{\theta}^l)$. Then $\boldsymbol{\theta}^g$ is learned by maximizing the approximate marginal likelihood $p(\mathcal{D}|\boldsymbol{\theta}^g)$. With $\boldsymbol{\theta}^l$ and $\boldsymbol{\theta}^g$ at hand, $p(\mathcal{D}|\boldsymbol{\theta}^l, \boldsymbol{\theta}^g)$ is capable of generating data points of $\mathcal{D}$, and called a probabilistic decoder. For details on variational auto-encoders, we refer to [Kingma and Welling, 2014, Rezende et al., 2014].

While amortized inference efficiently scales to large datasets and complex models, Equation (2.17) always defines a poorer variational posterior than the

mean-field family. The arising trade-off between the expressiveness of the variational family and scalability leads to what became known as the amortization gap [Cremer et al., 2018]. The gap can be mitigated in certain simple hierarchical models (including VAEs) where amortized VI achieves the same accuracy as CAVI or SVI but at a much lower cost. However, there are models where the gap cannot be closed and alternative methods are required [Agrawal and Domke, 2021, Margossian and Saul, 2024].

2.4.5 Variational inference in neural networks

In Chapter 1, we discussed Bayesian neural networks and the challenge of computing their posteriors. One of the first historical examples of variational methods in BNNs considered Gaussian priors and applied the Minimum description length (MDL) principle from information theory [Rissanen, 1986] to obtain a variational inference algorithm with the mean-field Gaussian family [Hinton and van Camp, 1993]. This algorithm was then extended to full-covariance Gaussian families [Barber and Bishop, 1998] by considering mixtures [Bishop et al., 1998, Jaakkola and Jordan, 1998]. Another early example is a mean-field variational algorithm developed for sigmoid belief networks [Saul et al., 1996], which can be treated as neural networks with graphical model semantics [Jordan et al., 1999, Neal, 1992b].

Nowadays, given the multimodal and complex nature of posteriors arising in Bayesian neural networks, much work has focused on understanding the underfitting tendencies of variational approximations (for a more general discussion of nuances of VI, see Section 2.4.2). [Trippe and Turner, 2018] noted that tightness of the evidence lower bound in mean-field variational inference can prune away most of the hidden units in the network; over-pruning leads to diagonal Gaussian approximations performing better when the covariance is fixed rather than learned. At the same time, in their experiments, full-rank Gaussian approximations consistently underperformed compared to mean-field approaches. [Coker et al., 2022] studied wide BNNs with odd activation functions and found that the optimal variational posterior predictive converges to the prior predictive distribution, completely ignoring the data; we revisit this phenomena in Section 3.2.2. [Foong et al., 2020] showed that mean-field Gaussian families do not contain suitable approximations for the true posterior of a single-layer BNN with rectified linear unit (ReLU) activations, resulting in poor uncertainty estimates. Conversely, in deeper BNNs, fully factorized families were shown to be as expressive as richer posteriors in shallow models [Farquhar et al., 2020]. [Park and Blei, 2024] improved mean-field variational approximations for BNNs by introducing a density uncertainty criterion that grounds predictive uncertainty in the training density of the input. More recently, [Gelberg et al., 2024] studied weight space permutation symmetries in neural networks and propose a symmetrization mechanism that improves the performance of unimodal (e.g., mean-field) variational approximations.

Since the relationship between priors on weights and posteriors in Bayesian neural networks is far from straightforward, and many common choices are often purely based on computational convenience (prior specification is discussed in

Section 1.2.4 and, below, in Section 2.4.6), [Sun et al., 2019a] introduced functional variational inference by formulating the VI task in the function space of neural networks. While the algorithm of [Sun et al., 2019a] did not scale well and was shown to posit an ill-defined variational objective [Burt et al., 2021], the functional view was improved by [Rudner et al., 2021], who considered approximating the distributions over functions induced by the variational distribution over parameters and derived a tractable (not relying on stochastic gradient estimators) objective functional variational inference algorithm.

Finally, given the challenges associated with the fully Bayesian perspective and the need for uncertainty quantification, variational inference algorithm developed for Bayesian last layer neural networks became popular due to its simplicity, strong empirical predictive performance and low computational costs [Harrison et al., 2024].

2.4.6 State of sparsity in variational BNNs

To scale with the size of the data and model complexity, various variational algorithms have been proposed for sparse BNNs. Within the class of shrinkage priors, classical and regularized horseshoe priors [Carvalho et al., 2009, Piironen and Vehtari, 2017a] on the BNNs weights, combined with variational approximations, have been shown to provide competitive empirical results. Examples include variational inference algorithms with mean-field [Louizos et al., 2017] and structured Gaussian variational families [Ghosh et al., 2018, 2019]. One of the most popular variational inference techniques for BNNs with spike-and-slab priors is perhaps "Bayes by Backprop" [Blundell et al., 2015], which builds on [Graves, 2011] and uses spike-and-slab priors with Gaussian spike and a fully factorized Gaussian family. The high variance of gradient estimates in Bayes by Backprop was addressed by [Wu et al., 2018], who introduced a deterministic approximation to the moments of activations. However, their approach did not consider spike-and-slab and used heavy-tailed (Inverse Gamma) priors, with hyperparameters set via empirical Bayes. In the case of the spike-and-slab priors with a Dirac spike, [Bai et al., 2019, 2020] applied stochastic relaxation to be able to approximate the ELBO and derived a theoretically justified (by contraction rates) variational algorithm; the approach was recently extended to spike-and-slab Group Lasso and spike-and-slab Group Horseshoe priors [Jantre et al., 2024]. In addition to computational advantages, variational tempered approximations of deep BNNs with spike-and-slab priors were shown to be consistent at the same convergence rate as the exact posterior [Chérief-Abdellatif, 2020]. More recently, [Castillo and Egels, 2024] considered BNNs with heavy-tailed shrinkage priors and obtained near-optimal minimax contraction rates for fully factorized tempered variational approximations. For a broader discussion of asymptotic properties and convergence rates of variational posteriors, we refer to Section 2.4.3.

Dropout (multiplicative noise) regularization in neural networks has also been shown to closely relate to sparse BNNs with suitable variational approximations. Specifically, [Gal and Ghahramani, 2016] showed that the dropout training objective in neural networks can be formulated in terms of minimizing the KL-divergence between the approximate and deep Gaussian process (GP) posteri-

ors. [Kingma et al., 2015] established the equivalence between Gaussian dropout training and variational inference for BNNs with log-uniform priors; however, log-uniform priors are not proper and have been shown to lead to improper posteriors [Hron et al., 2018]. Addressing the limitations of previous approaches, [Nalisnick et al., 2019] decoupled dropout from specific variational inference algorithms and established the equivalence between dropout regularization and placing Gaussian scale mixture priors with Automatic Relevance Determination structure on the network's weights. In a slightly different direction, [Li et al., 2024] considered BNNs with Gaussian priors and introduced a variational inference methodology that enforces sparsity in the weights during the training.

2.5 Towards a taxonomy of variational inference methods

Since being stated in the language of machine learning and introduced to the Bayesian statistical community as a method for approximating intractable posteriors [Jordan et al., 1999, MacKay, 1995], variational inference has been extensively studied and expanded in various directions. The formulation of posterior inference as a general optimization problem of minimizing the divergence between the true posterior and the approximation (given by Equation (2.1)) allows one to outline several angles from which variational inference can be studied:

- Variational family.
- Divergence function.
- Class of models.
- Optimization algorithm.

Variational family. The mean-field family remains the most common choice due to its convenience and linear scalability. For example, Algorithms 1 and 2 explicitly assume a choice of factorizable family. To make inference even more scalable, one can consider the amortized family discussed in Section 2.4.4. In a search for a more expressive approximation and support for both continuous and discrete random variables, variational families are endowed with various structures such as normalizing flows [Rezende and Mohamed, 2015], mixtures and hierarchical models [Ranganath et al., 2016]. Lastly, within the framework of BBVI and Algorithms 3 and 4, three of the natural assumptions are full-rank, diagonal Gaussian and, more broadly, location-scale families. We illustrate different choices of family within the larger approximate Bayesian framework on Figure 2.3.

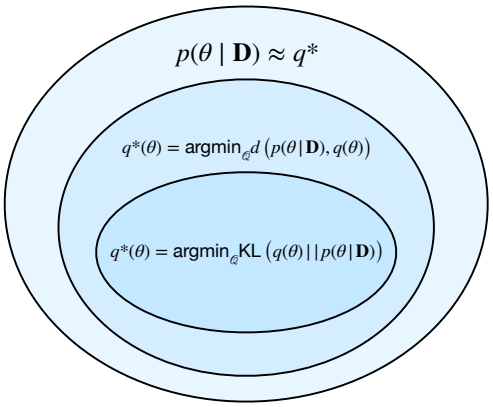
Divergence function. Whereas across Algorithms 1 to 4 we focus on the most popular and widely applicable variational Bayes approach with the variational objective given by the reverse Kullback-Leibler divergence and Equation (2.6), as we have observed in Sections 2.4.2 and 2.4.3 that is not the only possible choice.

Other means of measuring the distance between the true posterior and the approximation include the forward KL divergence [Vehtari et al., 2020], α divergence [Hernandez-Lobato et al., 2016], χ^2 divergence [Dieng et al., 2017], f divergence [Wang et al., 2018] and a score-based divergence [Modi et al., 2024]. For an extensive empirical comparison of different divergences in variational inference, we refer to [Dhaka et al., 2021] and for recent theoretical study to [Margossian et al., 2024]. Figure 2.3 illustrates VI with the KL divergence among all approximate Bayesian inference methods.

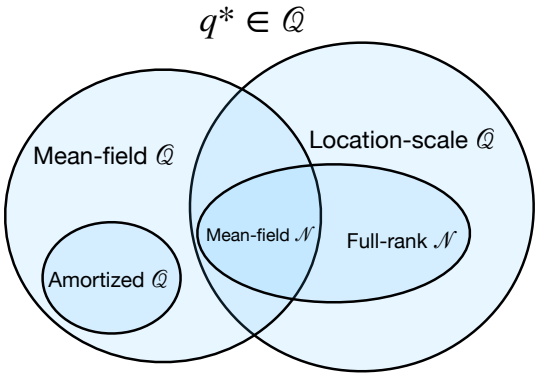
Class of models. The easiest for computing are, perhaps, conditionally conjugate models for which the naive gradient descent solves the optimization task. Black box methods allow to expansion of the class of models and avoid model-specific derivations. Whilst Algorithm 3 is applicable to all evaluable models, vastly studied and employed Algorithm 4 requires differentiable models. Figure 2.3 outlines how different classes of models considered in variational inference relate to each other.

Optimization algorithm. Among the considered algorithms, only CAVI of Algorithm 1 employs classical gradient descent. In most of the cases, including Algorithms 2 to 4, however, stochastic gradient optimization is used. Besides the regular stochastic gradient descent, the proximal and the projective SGDs, upon which we have touched in Section 2.4.3, can be employed in variational inference.

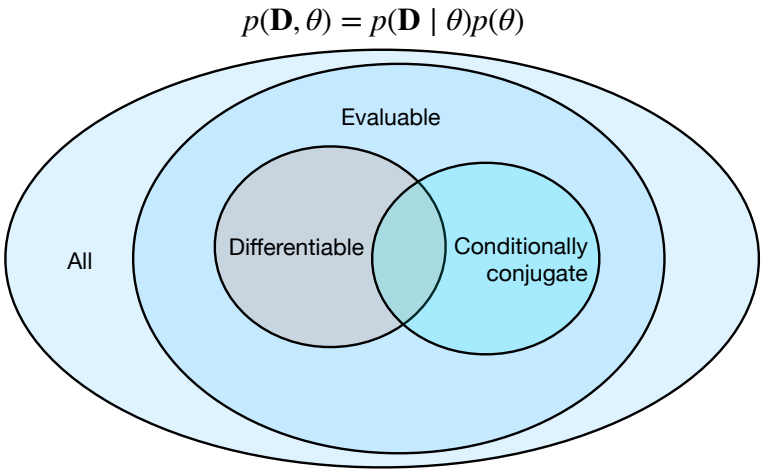
Avenues for future work. The outlined taxonomy provides intuition on the possible directions of future research. Apart from solely focusing on extending variational families, our understanding of the relationship between the quality of the approximation and the complexity of the family could be improved. Further, the trade-off between the expensiveness of variational posterior and computational complexity is something to keep in mind (we explicitly encounter this trade-off in the empirical settings in Section 3.2.2, where the full-rank ADVI is not feasible). The optimality of the reverse KL divergence is not granted and other divergences could be explored in bigger detail and in various problems, especially in a context of reliable uncertainty quantification [Margossian et al., 2024], and when assessing the quality of variational posteriors [Huggins et al., 2020]. Whilst the typical assumption on the model is differentiability, it would be interesting not only to adapt existing algorithms to larger classes of models but also to investigate models in which variational inference can perform particularly well (e.g. models exhibiting certain symmetries are of potential interest [Margossian and Saul, 2024]). Recently, several promising convergence guarantees for the BBVI with regular, proximal and projective SGDs and with various reparametrization gradient estimators were established [Domke et al., 2024, Kim et al., 2024b]. Similar analyses could investigate whether the inference could be sped up by studying the guarantees given other gradients (e.g. natural) or different gradient estimators (e.g. score or recently introduced "“sticking the landing” estimator). Lastly, given the non-convex nature of the ELBO function, research on non-convex opti-



(a) VI with KL divergence in the framework of approximate Bayesian inference.



(b) Common families considered in VI.



(c) All probabilistic models and models considered in VI.

Figure 2.3: Diagrams outlining the relationship between (a) approximate Bayesian inference methods; (b) families of variational distributions and (b) probabilistic models in the context of variational inference [Blei, 2019, Broderick, 2020].

mization could be formulated in the framework of variational inference to improve the convergence of variational algorithms. To conclude, variational inference is a popular alternative to sampling-based approximate Bayesian inference and has developed rapidly over the past decade. Expanding research in various directions outlined in the taxonomy can tackle the challenges associated with variational inference and lead to more scalable, accurate and reliable algorithms.

Chapter 3

The Architecture and Evaluation of Bayesian Neural Networks

In Chapter 1 we discussed various aspects of specifying the architecture of classical and Bayesian deep learning models. In order to define Bayesian neural network (BNN), one needs take several key steps: choose suitable width, depth, prior distribution and activation function. Computational burdens and intractable posteriors further expose miscalibrated Bayesian neural networks to poor accuracy and unreliable uncertainty estimates. Variational inference (VI) discussed in Chapter 2, benefits from improved computational complexity but lacks the asymptotical guaranties of the Markov chain Monte Carlo (MCMC). At the same time, the dimensions of modern deep models make MCMC tremendously expensive. Neither, sampling nor optimization-based algorithms are faultless and their performance heavily depends on architectural choices. This chapter aims to shed some light on the behaviour of the Bayesian neural networks in practice.

Some of the work presented in this chapter appears in [Sheinkman and Wade, 2025].

3.1 Introduction

As modern neural networks (NNs) get more and more complex, specifying a model with high predictive performance becomes a more and more challenging task. Neural networks are parametrized in the weight space, where properties and dimensionality need to be specified beforehand. The number of parameters is an essential part of the model choice; and the more parameters one has, the more nuanced the choice of model becomes. No matter what the prediction task is, overly complex models suffer from the curse of dimensionality which causes not only poor performance but also computational problems. The challenge is finding a model that matches the task and, as importantly, achieves the alignment between the model and the applied inference algorithm [Gelman et al., 2020].

In contrast to classical deep learning, where training is done through gradient-based optimization, in Bayesian settings it is not enough to specify

the learning rate and choose the weight initialization scheme. Despite some promising theoretical results on the true posterior predictive distribution of Bayesian neural networks, these distributions are typically intractable and highly multimodal [Baltrusaitis et al., 2019], and the properties of even the most theoretically grounded sampling methods and approximation techniques are limited by the computing budget, size of the dataset, and sheer number of parameters [Arbel et al., 2023, Magris and Iosifidis, 2023, Papamarkou et al., 2022]. In Section 3.2, we will see that for different inference algorithms, one model can provide strikingly diverse performances.

In theory, given some set of models, the Bayesian approach has the potential to deal with the model choice and compare different models in a principled way. [Kass and Raftery, 1995, MacKay, 1992, 2003] argue that, independently of the choice of priors on models (and hence automatically), the Bayes factors defined in Section 1.2.1 embody Occam's razor[5], and Bayesian inference favours the "simplest" model over the complex one. However, in the context of modern models and real-world applications, the heuristic of Occam's razor cannot be considered naively and is not always very informative; a simple model may closely approximate the observed data but fail to generalize well. The philosophy of Occam's razor should not be seen as the resolution of the phenomena known as the curse of dimensionality. In fact, the curse of dimensionality can be dealt with by introducing "less simple" hierarchical priors on the weights of the neural network [Polson and Sokolov, 2019] (for more details on such priors we refer to Section 4.2.2). Further, the complexity of the model cannot be estimated by simply counting the number of parameters, and the Bayes factors can favour models with many parameters, e.g. wider neural networks [Rasmussen and Ghahramani, 2000]; indeed, neural networks with a large number of hidden units were shown to be successful even when dealing with small data sets [Neal, 1995, Williams, 1996]. Moreover, one might want to consider an infinite-width neural network to obtain a Gaussian process (GP) limit, as the latter are known for their expressiveness and generalization abilities [Neal, 1995, Rasmussen and Williams, 2005, Wilson and Izmailov, 2020]. At the same time, while the equivalence of infinitely wide Bayesian neural networks and Gaussian processes has been long established [Matthews et al., 2018, Neal, 1995, Rasmussen and Williams, 2005], the properties of approximations of the BNNs as well as the behavoir of finite-width BNNs with respect to GPs are less clear [Coker et al., 2022, Vladimirova et al., 2019].

Instead of choosing one model, one might improve reliability of predictions by combining several models. In Section 1.2.5, we discussed the connection between non-Bayesian ensembles and Bayesian approaches. In this chapter, we will consider model combination in a Bayesian sense of combining not the point estimates but probability distributions. Predictive distributions obtained by several models or by running inference algorithms several times for one model can be averaged based on model probabilities, this leads to Bayesian model averaging (BMA) [Hoeting et al., 1999]. However, model probabilities are often overcon-

[5]Also known as Ockham's razor or principle of parsimony and attributed to William of Occam, a fourteenth-century philosopher from Surrey.

fident [Oelrich et al., 2020], and classical BMA tends to average models with coefficients which are very close to either 0 or 1, which is only optimal if the true model is among the comparison set.

Outline of the chapter. To consistently evaluate Bayesian neural networks in practice, we consider computational costs, accuracy and uncertainty quantification in different scenarios, including large width and out-of-sample data. Specifically, we study the sensitivity of BNNs to the choice of width in Section 3.2.2, depth in Section 3.2.3. While Bayesian models are more resistant to distribution shifts, the reliability of uncertainty estimates and the gap within-the-sample and out-of-sample performance still require improvement [Ovadia et al., 2019, Wild et al., 2023], and we investigate the performance of BNN under the distribution shift in Section 3.2.4. The challenge of comparative model assessment is addressed in Sections 3.3.1 and 3.3.2, where we introduce the estimated pointwise log-likelihood as a measure of model utility, which is preferable over Bayes factors. Finally, in response to the limitations of BMA, in Sections 3.3.3 and 3.3.4 we consider ensembling, stacking and pseudo-BMA.

3.2 Empirical study

Motivated by the challenges and nuances of Bayesian neural networks observed above and in Chapter 1, this section aims to evaluate the performance of Bayesian neural networks in practice. We begin by constructing a neural network with different choices of the activation function, and consider the performance of wider and deeper networks for different posterior inference algorithm choices. We then investigate the behaviour of BNNs under the distribution shift. Across all the experiments, we observe that for different inference algorithms, one model can provide strikingly diverse performances.

3.2.1 Settings of the experiment

In the experiments, we consider the following Bayesian neural network with L hidden layers and D_l hidden units per layer l, illustrated by the Figure 3.1

$$\begin{aligned} &\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{b}_{L+1} + \boldsymbol{W}_{L+1}\boldsymbol{z}_L, \boldsymbol{\sigma}^2\right), \quad \boldsymbol{\sigma} \sim |\mathcal{N}(0, 1e-6)|, \\ &\boldsymbol{z}_l = g\left(\boldsymbol{b}_l + \boldsymbol{W}_l \boldsymbol{z}_{l-1}\right) \text{ for } l = 1, \ldots, L, \end{aligned} \tag{3.1}$$

and the following priors on the weights and biases:

$$\begin{aligned} &\boldsymbol{W}_1 \sim \mathcal{N}\left(\mathbf{0}, \frac{\mathbf{1}}{LD_0}\right), \boldsymbol{b}_l \sim \mathcal{N}\left(\mathbf{0}, \frac{\mathbf{1}}{4L}\right), \\ &\boldsymbol{W}_l \sim \mathcal{N}\left(\mathbf{0}, \frac{\mathbf{2}}{D_{l-1}}\right) \text{ for } l = 2, \ldots, L+1, \end{aligned}$$

where we denote $\boldsymbol{z}_0 = \boldsymbol{x}$, and $|\mathcal{N}(,)|$ denotes a half-normal distribution, and we consider two different choices of activation functions, namely, the rectified linear

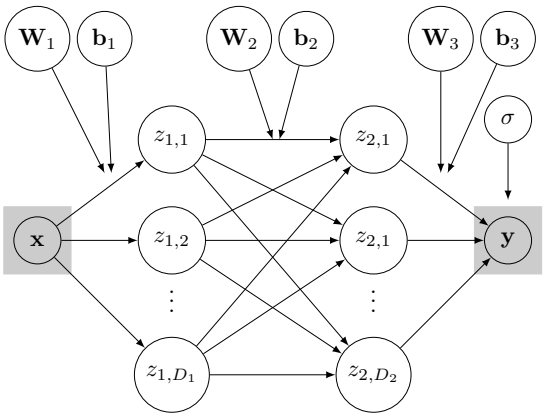


Figure 3.1: Example of the directed acyclic graph (DAG) of the neural network used in the experiments when $L = 2$.

unit (ReLU) and the sigmoid defined in Section 1.1.1.

Note that $\boldsymbol{W}_l \in \mathbb{R}^{D_l \times D_{l-1}}$ and $\boldsymbol{b}_l \in \mathbb{R}^{D_l}$ for $l = 1, \ldots, L+1$, and to avoid divergence in wider and deeper networks, the biases' variance is scaled by the the inverse of the depth and the weights' variance is scaled by the inverse of the preceding layer's width. Additionally, [He et al., 2015] shows that in the deep ReLU neural networks, an appropriate scaling mitigates the damage caused by the non-linear deformation and speeds up the convergence.

Even though we do not empirically study different choices here, Appendix A.4 provides results for Student-t priors, which in this particular example, performed similarly to Gaussian priors, with one exception occurring in the case of deeper networks. Further, note that in Chapter 4, more sophisticated hierarchical priors on the weights of BNN are implemented.

The BNN defined by Equation (3.1) and trained with automatic differentiation variational inference (ADVI), which assumes a mean-field (diagonal) Gaussian variational family (Section 2.3.3), is referred to as mfVIR or mfVIS, depending on the choice of the activation: ReLU or sigmoid, respectively. The model trained with the Hamiltonian Monte Carlo (HMC) inference, using the No U-Turn Sampler (NUTS), is denoted as HMCR or HMCS [Hoffman and Gelman, 2014]. Whilst this thesis focuses on the variational inference, the reader interested in the classical results on MCMC methods is referred to [Robert et al., 1999], in the MCMC in the context of BNNs to [Papamarkou et al., 2022]. For simplicity, we often refer to one-layer neural networks of particular width D as to mfVIRD, mfVISD, HMCRD or HMCSD (e.g. one-layer BNN with 20 hidden units and ReLU activation trained with mean-field VI is called mfVIR20). All experiments are implemented with Numpyro [Phan et al., 2019], ArviZ [Kumar et al., 2019], JAX [Bradbury et al., 2018] and Flax [Heek et al., 2024]. We record the run time of the approximate inference step (TT), the root mean squared error (RMSE) and empirical coverage for the function and observations (EC). Note that we compute empirical coverage as a fraction of observations contained within the 95% credible interval (CI), this means that in the ideal settings the computed EC should equal 0.95. If EC > 0.95 then the credible intervals are too wide; a worse scenario occurs when EC < 0.95 as it means that the CIs are too

narrow and the model is overconfident in predictions. Details on the computed metrics and the corresponding formulas are discussed in Appendix A.1, where we provide further information on the initialization and parameters for the inference algorithms. The absence of the test log-likelihood among the recorded metrics is motivated by the recent observation of [Deshpande et al., 2024], who show that the higher test log-likelihood does not necessarily correspond to a more accurate posterior approximation nor to lower predictive error (such as RMSE).

3.2.2 Increasing the width of the network

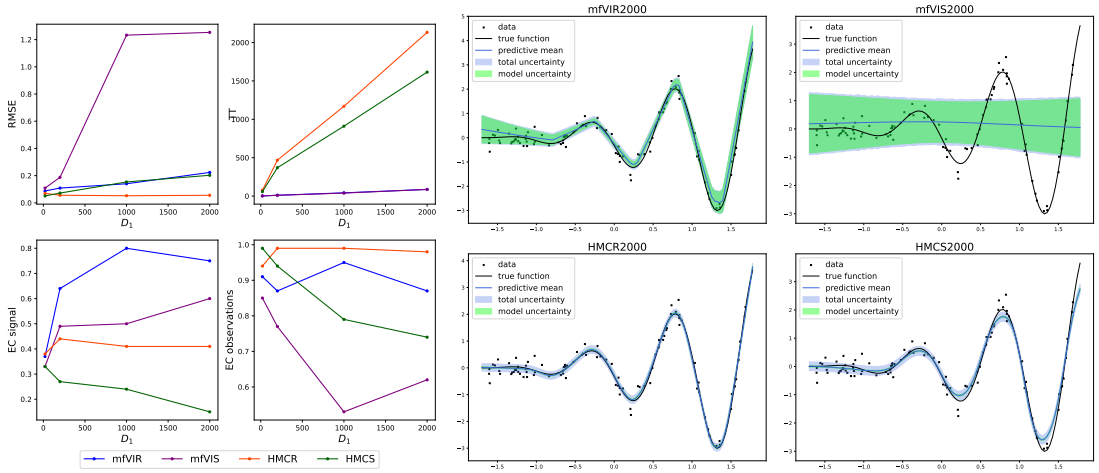
We consider a simple synthetic dataset with one-dimensional input and output $\sin(10\boldsymbol{x})\boldsymbol{x}^2$, where the noisy observations on which the neural network is trained are obtained by adding some Gaussian noise:

$$\begin{aligned}\boldsymbol{x} &\sim \text{Unif}([0,2]),\\ \boldsymbol{y} &= \sin(10\boldsymbol{x})\boldsymbol{x}^2 + \boldsymbol{\epsilon},\\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, 0.0625).\end{aligned}$$

The input is scaled, unlike the output. The training data $\mathcal{D}$ consists of $N = 500$ observations and the new data for testing $\tilde{\mathcal{D}}$ consists of $\tilde{N} = 100$ observations.

In notations of Section 3.2.1, here we study the performances of mfVIR, mfVIS, HMCR and HMCS with 1 hidden layer as the width increases and illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units by the Figure 3.2a. The predictions of all four models when $D_1 = 2000$ are provided on the Figure 3.2b. The performance of the mfVIS dips with the increase in the dimension of the hidden layer; moreover, for $D_1 = 1000$ and $D_1 = 2000$, its posterior predictive distribution fails to capture the data, and, in fact, degenerates to the prior (Figure 3.2b). An explanation of why such behaviour occurs was obtained via the correspondence of GPs and BNNs; when the true posterior of a BNN converges to an NNGP posterior [Hron et al., 2022], the mean-field variational approximation of the true posterior exhibits a different pattern. Note, for example, that tanh is an odd function, as well as up to a constant vertical shift the sigmoid ($-\sigma(-x) = \sigma(x) - 1$, recall, Figure 1.2a). Then any optimal mean-field Gaussian variational posterior of a BNN with odd (up to a constant offset) Lipschitz activation function converges to the prior predictive distribution of the NNGP [Coker et al., 2022] as the width goes to infinity. In other words, the mean-field variational approximations of wide BNNs with a sigmoid activation function ignore the data. And if one abandons the mean-field assumption and proposes a full-rank variational family, then using VI for wider networks would take at least a hundred times more time than using HMC, which undermines the benefits of using VI. Instead, with HMC, this degenerate behaviour is not observed (Figure 3.2b), but this comes at a subsequent increased cost in computational time. For wider networks, the HMCR model exhibits a better performance than the HMCS both in terms of accuracy and uncertainty quantification.

In terms of predictive accuracy, HMC is preferred over mfVI in all of the combinations of the activation function and width. However, in terms of uncertainty quantification, the HMC is inferior to mfVI. In our experiment, HMC underesti-



(a) The prediction performances of all the models are compared as the number of hidden units increases.

(b) The predictions and uncertainty estimates obtained by each model when $D_1 = 2000$

Figure 3.2: Predictive performance of BNNs as the width increases.

mates the uncertainty of the signal much more than VI (Figures 3.2a and 3.2b). Note that whilst variational inference is often cursed to underestimate the uncertainty[Trippe and Turner, 2018, Turner and Sahani, 2011], that is not always the case [Blundell et al., 2015, Gal and Ghahramani, 2016]. MCMC methods are known to struggle to effectively explore multimodal posteriors [Izmailov et al., 2021, Papamarkou et al., 2022, Wenzel et al., 2020a], and a lack of uncertainty could be a result of poor mixing of the chain. Lastly, it is needless to say that the training time of HMC algorithms drastically increases in wider networks.

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

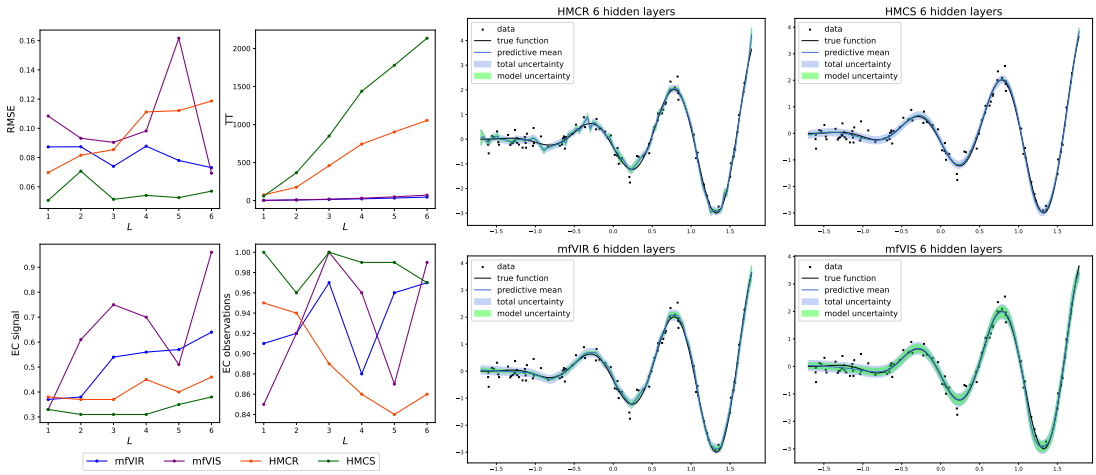
3.2.3 Increasing the depth of the network

While the mean-field Gaussian approximations of single-layer BNNs fail to capture the uncertainty when the regions of the increased uncertainty are sandwiched between regions of low uncertainty, the mean-field Gaussian approximations of deeper networks with ReLU activation have the potential to provide improvement [Foong et al., 2020]. Along the same line, [Farquhar et al., 2020] ar-

gues that the approximate posteriors of deep neural networks obtained with the mean-field variational inference are as flexible as the much richer approximate posteriors of shallower BNNs.

Consider the data of Section 3.2.2 and neural networks defined by Equation (3.1) with the number of layers L varying from 1 to 6 and a fixed number of hidden units in each layer $D_h = 20$. Figure 3.3a provides the recorded metrics, and Figure 3.3b illustrates the predictions of the four possible combinations of activation and inference algorithm with $L = 6$. First, observe that overall both RMSE and empirical coverage of mfVIR approximations improve with the increase of depth. The mfVIS follows a similar pattern, except for the case of $L = 5$ when the prediction quality of the network drops drastically. However, we do not obtain the same improvement in the prediction quality of models trained with HMC: the performance of HMCR falls whilst the HMCS does not improve as the depth increases. This undesirable behaviour could be a result of the multimodality of distributions in overparametrized models [Baltrusaitis et al., 2019, Izmailov et al., 2021] combined with the challenges of MCMC in exploring the high-dimensional space. Compared to the findings of Section 3.2.2, we note that the deeper NNs are less sensitive to the choice of the activation function.

It is needless to say that HMC algorithm scales rather poorly as the number of parameters increases, and as the number of layers changes from $L = 1$ to $L = 6$, the time needed to train HMCR and HMCS gets more than 15 and 30 times greater, respectively. We note that for models with more than one hidden layer, training of the network with sigmoid activations takes roughly twice as much time as the network with ReLU. The striking discrepancy in training times could arise due to the difference in the leapfrog integrator step sizes [Betancourt et al., 2014].



(a) The prediction performances of all the models are compared as the number of hidden layers increases.

(b) The predictions and uncertainty estimates obtained by each model when $L = 6$

Figure 3.3: Prediction performance of deeper networks.

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

3.2.4 Out-of-distribution prediction

Reliable uncertainty estimates that are robust to out-of-distribution (OOD) data become exceptionally important in safety-critical applications such as autonomous driving or medical diagnosis. Even though we acknowledge that such tasks often require models which can detect unknown data, in this section, we consider generalization abilities and robustness to the OOD, but not the separate task of the OOD detection [Hendrycks and Dietterich, 2018] (in Bayesian setting that was considered in e.g. [Park and Blei, 2024]). It is not surprising that the accuracy and the quality of uncertainty quantification of any model decrease under a distribution shift. However, the challenge is especially intricate since better accuracy and lower calibration error of a certain model on the in-domain data do not imply better accuracy and lower calibration error in the OOD settings [Ovadia et al., 2019, Ritter et al., 2021].

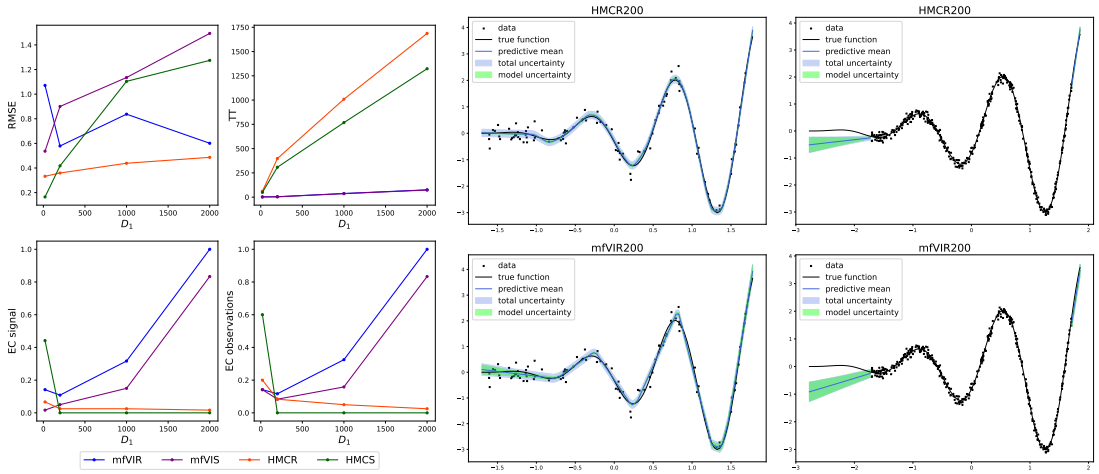
Here, we wish to validate the models' predictive abilities when the test data points come from previously unseen regions of data space. The kind of out-of-distribution data we consider could be described as 'complement-distribution'. Even though transformed- and related-distributions are more common in real-life applications, complement-distributions still arise in open-set recognition or could be the result of an adversary [Farquhar and Gal, 2022]. We split the training data used in Sections 3.2.2 and 3.2.3 into the train and test data covering complementary regions of the function. Specifically, the observed data $\mathcal{D}_c$ consists of $N = 370$, the new data $\tilde{\mathcal{D}}_c$ consists of $\tilde{N} = 130$ and the observed and the new data are disjoint (see Figure 3.4):

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

Strictly speaking, we do not expect any model to be robust to such an extreme case and, mainly, want to assess and better understand the quality of the uncertainty estimates. In this experiment, we are hoping that the relationship between the distributions of the observed and the new data makes this challenge somewhat tractable. Note that in Section 3.3.4 we introduce a much milder example with related-distribution test data.

On Figure 3.4a we illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units; Figure 3.4b compares non-OOD and OOD predictions obtained by the BNNs with ReLU activation and $D_1 = 200$. The poor performance of the

mfVIS, especially for wider networks, is not surprising; however, we notice that for wide networks, HMCS suffers from much higher RMSE than mfVIR and HMCR for wide networks. And while HMCR has a lower RMSE than any model trained with mean-field VI, the ability of HMC to capture the uncertainty deteriorates, and it becomes overconfident. [Foong et al., 2020] provide an example of a single-layer BNN for which the posterior obtained with HMC can capture the increase in the uncertainty in the OOD regions and the mean-field approximations cannot. In contrast, we observe that whilst HMCR200 and mfVIR200 do not show any of the expected increase in the uncertainty, on certain regions both methods are able to provide accurate predictive mean (see Figure 3.4b, the right-hand side region of the function, where $x > 1.5$). Finally, as the width of the network increases, mfVIR outperforms all of the methods.



(a) Out-of-sample predictive performance of all the models compared.

(b) Within-the sample and out-of-sample predictions and uncertainty estimates of BNNs with $D_1 = 200$ and ReLU activation.

Figure 3.4: Out-of-distribution prediction for the complement-distribution data.

General summary. In terms of the accuracy alone, the HMC with ReLU is more robust to the out-of-distribution data, however, that comes with the largest computational costs among all the models. We already saw in Section 3.2.2 that uncertainty quantification with HMC degrades with increasing width. In OOD settings, this becomes even more extreme, with very overconfident predictions that do not cover the truth (an empirical coverage of almost zero). Finally, with the increase in depth, in the extreme OOD settings, the mfVI with ReLU becomes almost as accurate as HMC with ReLU and provides better uncertainty quantification at a much lower cost.

3.3 Bayesian model assessment

When considering synthetic datasets, we can choose a desired metric and sample any number of data points, so that evaluation of the model's performance becomes

trivial. For example, in Section 3.2.4 we have specifically created an extreme case when the training data $\mathcal{D}_c$ and the new data $\tilde{\mathcal{D}}_c$ were covering disjoint regions of the true function. In reality, the new previously unseen data is not available, and one can only estimate the expected out-of-sample predictive performance. In this section, we study the approach to Bayesian model choice and combination based on the estimates of the expected performance.

3.3.1 Predictive methods for model assessment

Suppose that we only observe $\mathcal{D}$, the unseen observations $\tilde{\mathcal{D}}$ are generated by $p_t(\tilde{\mathcal{D}})$, and we wish to be able to assess the generalization ability of the model without having access to the test data. To keep the notation simple, we omit the dependency on $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ when writing down the posteriors in this section. Given a new data point $\tilde{y}_n$, the log score $\log p(\tilde{y}_n|\mathcal{D})$ is one of the most common utility functions used in measuring the quality of the predictive distribution. The log score benefits from being a local and proper scoring rule [Gneiting, 2011, Vehtari and Ojanen, 2012]. Then, the expected log pointwise predictive density for a new dataset serves as a measure of the predictive accuracy of a given model:

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

where $p(\tilde{y}_n|\mathcal{D})$ is model's posterior predictive distribution. In the absence of $\tilde{\mathcal{D}}$, one might obtain an estimate of the expected log pointwise predictive density by re-using the observed $\mathcal{D}$. Here, we review two possible approaches: the first approach overestimates the elpd by first computing the log pointwise predictive density of $\mathcal{D}$ and then adjusts it by some correction term; the other approach employs cross-validation [Vehtari et al., 2016]. Suppose that $\boldsymbol{\theta}$ are the parameters of the model and $\boldsymbol{\theta}^s, s = 1, \ldots, S$ are simulation draws. Then, one can evaluate the log pointwise predictive density as

$$\begin{aligned}
\text{lpd} &= \sum_{n=1}^{N} \log p(y_n|\mathcal{D}) = \sum_{n=1}^{N} \left[\int p(\boldsymbol{\theta}|\mathcal{D}) \log p(y_n|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \right], \\
\widehat{\text{lpd}} &= \sum_{n=1}^{N} \log \left[\frac{1}{S} \sum_{s=1}^{S} p(y_n|\boldsymbol{\theta}^s) \right].
\end{aligned}$$

With $\widehat{\text{lpd}}$ at hand, a superior successor of the Akaike Information criterion (AIC) [Akaike, 1998] and the Deviance information criterion (DIC) [Spiegelhalter et al., 2002] called the Watanabe-Akaike information criterion (WAIC) [6] [Watanabe, 2010] can be obtained. To mitigate the bias between the log pointwise density and the expected utility, WAIC subtracts the simulation-estimated effective number of parameters:

$$\widehat{\text{elpd}}_{\text{WAIC}} = \widehat{\text{lpd}} - \widehat{p}_{\text{WAIC}}, \text{ where}$$

[6] Also called Widely Applicable information criterion

$$\widehat{p}_{\text{WAIC}} = \sum_{n=1}^{N} \text{Var}^S(p(y_n|\boldsymbol{\theta}^s)).$$

Here, Var^S is the sample variance and the estimated effective number of parameters $\widehat{p}_{\text{WAIC}}$ can be seen as a measure of model complexity. Asymptotically, WAIC is equivalent to the Bayesian leave-one-out cross-validation (LOO-CV) estimate of the expected utility [Watanabe, 2010]. Even though cross-validation is a natural framework for assessing the model's predictive performance, the WAIC was for a long time preferred over the LOO-CV due to the computational challenges arising from multiple model runs [Gelman et al., 2014]. To approximate the $\widehat{\text{elpd}}_{\text{loo}}$ and avoid re-fitting the model N times, one could use importance sampling; unfortunately, the classical importance weights would have a large variance, and the obtained estimates would be noisy. Recently, [Vehtari et al., 2024] solved this problem by developing the Pareto smoothed importance sampling (PSIS), which allows evaluating the LOO-CV expected utility in a more reliable yet efficient way:

$$\begin{aligned}\widehat{\text{elpd}}_{\text{loo}} &= \sum_{n=1}^{N} p(y_n|x_n, \mathcal{D}_{-n}), \\ &= \sum_{n}^{N} \log\left(\frac{\sum_{s=1}^{S} r_i^s p(y_n|\boldsymbol{\theta}^s)}{\sum_{s=1}^{S} r_i^s}\right),\end{aligned} \tag{3.2}$$

where r_i^s are the smoothed importance weights, which benefit from smaller variance than the classical weights. Later in this chapter, we will refer to the individual logarithms in the sum as $\widehat{\text{elpd}}_{\text{loo},n}$. The advantage of PSIS is that the estimated shape parameter of the Pareto distribution provides a diagnostic of the reliability of the resulting expected utility. Note that the estimated shape parameter of PSIS has also been used as a diagnostic of variational inference [Yao et al., 2018b].

Although the methods of model selection which reuse the data can be vulnerable to overfitting when the size of the dataset is too small and/or the data is sparse, it is (relatively) safe to use CV to compare a small number of models and given a large enough dataset [Piironen and Vehtari, 2016],[Vetari et al., 2019]. Moreover, whilst both the PSIS-LOO and WAIC estimates give nearly unbiased estimates of the predictive ability of the model, $\widehat{\text{elpd}}_{\text{loo}}$ was shown to be more robust than $\widehat{\text{elpd}}_{\text{WAIC}}$; in the presence of limited sample size and weak priors, WAIC can severely underestimate $\hat{p}_{\text{WAIC}}$ and often has a larger bias towards the log predictive density [Gelman et al., 2014, Vehtari et al., 2016].

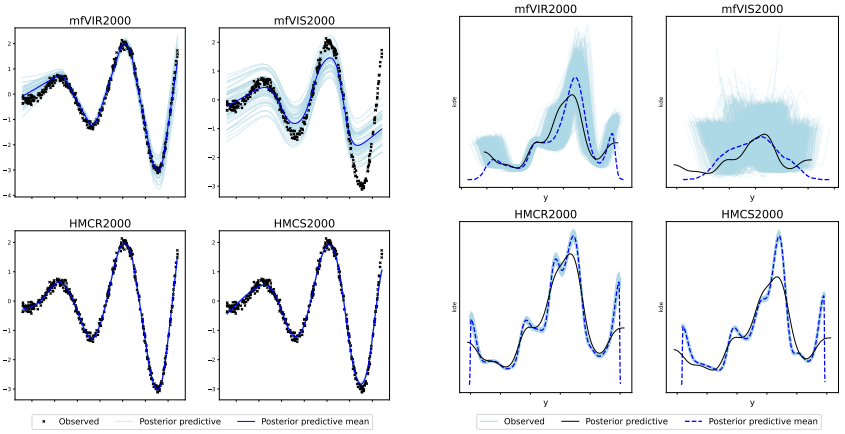
3.3.2 Model assessment in practice

In the lack of new previously unseen data, we could begin with posterior predictive checks (PPC), which compare the true $\mathcal{D}_c$ to datasets simulated from the posterior predictive distribution [Gelman et al., 2020].

Figure 3.5a provides the PPC based on the kernel density estimates of the

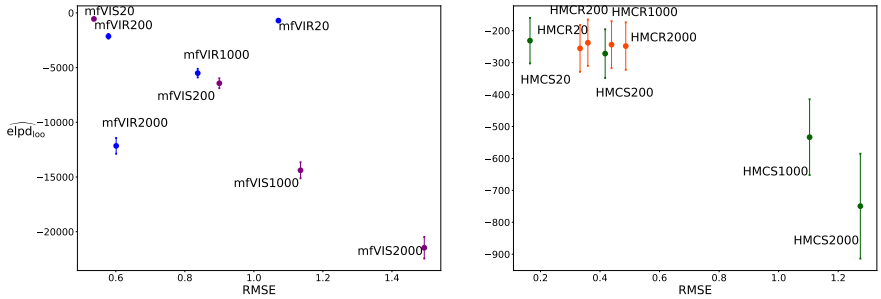
observed $\boldsymbol{y}$ and Figure 3.5b compares the posterior predictive to the observed $\boldsymbol{y}$ for all of the models with $D_1 = 2000$; both figures evidently classify the mean-field VI with sigmoid activation as inappropriate. However, it is not apparent that HMCS2000 is significantly inferior to HMCR2000.

Considering the settings of Section 3.2.4, we compare the previously computed RMSE to $\widehat{\text{elpd}}_{\text{loo}}$ estimates. Since the idea of estimating the expected log predictive density is in evaluating future predictive performance, we expect the higher $\widehat{\text{elpd}}_{\text{loo}}$ to correspond to a better model and lower RMSE. Indeed, Figure 3.5c is more informative in this sense than PPC diagnostics, and we observe the inverse dependence between $\widehat{\text{elpd}}_{\text{loo}}$ and RMSE. Sampling and approximation techniques result in different scales of $\widehat{\text{elpd}}_{\text{loo}}$ estimates (in general, $\widehat{\text{elpd}}_{\text{loo}}$ is lower for VI, especially in wide networks), and thus, we compare the models trained with different algorithms for better visualization.



(a) Posterior predictive checks for the wider models based on the kernel density estimates of the observed $\boldsymbol{y}$.

(b) Posterior predictive distribution and posterior predictive mean compared to the observed $\boldsymbol{y}$.



(c) The correspondence between the $\widehat{\text{elpd}}_{\text{loo}}$ and the RMSE in the OOD scenario. Higher $\widehat{\text{elpd}}_{\text{loo}}$ should correspond to lower RMSE.

Figure 3.5: Estimating the out-of-distribution performance before seeing the new data: testing the (a), (b) PPC and (c) $\widehat{\text{elpd}}_{\text{loo}}$. The mfVIR2000 is confirmed to be unreliable in all methods. The PPC of the HMCS2000 does not provide enough information to judge its performance in the OOD settings, while the $\widehat{\text{elpd}}_{\text{loo}}$ does.

Based on $\widehat{\text{elpd}}_{\text{WAIC}}$, one can obtain a correspondence between the models with dependencies similar to those of Figure 3.5c, and so we can consider the LOO

estimate of the elpd as somewhat reliable, for details see Figure A.1 provided in Appendix A.2.

General summary. In certain cases, such as mfVIS for large width, the posterior predictive checks are able to detect an undesirable model. However, when the PPCs are not sufficient, we confirmed that the PSIS-LOO estimates of the expected log pointwise predictive density can serve as robust diagnostics for both the mfVI and the HMC methods.

3.3.3 Bayesian model averaging and stacking

Let $\mathcal{M} = \{M_1, \ldots, M_K\}$ be a collection of models and denote the parameters of each of the M_k as $\boldsymbol{\theta}_k$. The assumptions one has on the prediction task and on $\mathcal{M}$ with respect to the true data-generating process can be categorized into three scenarios: $\mathcal{M}$-closed, $\mathcal{M}$-open and $\mathcal{M}$-complete. If $M_k \in \mathcal{M}$ for some k recovers the true data-generating process, then we are in the $\mathcal{M}$-closed case. The task is $\mathcal{M}$-complete if there exists a true model but it is not included in $\mathcal{M}$ (e.g. for computational reasons). Finally, we are in the $\mathcal{M}$-open scenario when the true model is not in $\mathcal{M}$ and the data-generating mechanism cannot be conceptually formalized to provide an explicit model [Vehtari and Ojanen, 2012]. The Bayesian framework allows us to define the probabilities over the model space, and for the $\mathcal{M}$-closed case, classical Bayesian Model Averaging would give optimal performance. The BMA solution provides an averaged predictive posterior as [Hoeting et al., 1999]

$$p(\tilde{\boldsymbol{y}}|\mathcal{D}) = \sum_{k=1}^{K} p(\tilde{\boldsymbol{y}}|\mathcal{D}, M_k) p(M_k|\mathcal{D}). \tag{3.3}$$

However, in the $\mathcal{M}$-open and $\mathcal{M}$-complete prediction tasks, the BMA is not appropriate as it gives a strong preference to a single model and so assumes that this particular model is the true one.

Now, if we replace the weights $p(M_k|\mathcal{D})$ with the products of Bayesian LOO-CV densities $\prod_{n=1}^{N} p(y_n|x_n, \mathcal{D}_{-n}, M_k)$, we arrive at pseudo-Bayesian model averaging (pseudo-BMA). In other words, the weights w_k of pseudo-BMA are proportional to the estimated log pointwise predictive density $\exp(\widehat{\text{elpd}}_{\text{loo}}^{k})$ introduced in Section 3.3.1. One could further correct each $\widehat{\text{elpd}}_{\text{loo}}$ estimate of Equation (3.2) by the standard errors and obtain

$$w_k = \frac{\exp(\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}})}{\sum_{k=1}^{K} \exp(\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}})},$$

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

where for each model M_k we find $\widehat{\text{elpd}}_{\text{loo}}^{k,\text{reg}}$ by utilizing a log-normal approximation.

Fortunately, we have already seen that these densities can be efficiently estimated with PSIS.

An alternative way to obtain the averaged predictive posterior given the set of $p(\tilde{\boldsymbol{y}}|\mathcal{D}, M_k)$ is to employ the stacking approach of [Yao et al., 2018a]. Define the set $S^K = \{\boldsymbol{w} \in [0,1]^K | \sum_k^K w_k = 1\}$, then the stacking weights are found as the optimal (according to the logarithmic score) solution of the following problem

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

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

Finally, deep ensembles of classical non-Bayesian NNs [Lakshminarayanan et al., 2017] briefly mentioned in Section 1.2.5 behave similarly to Bayesian model averages, and both lead to solutions strongly favouring one single model [Wilson and Izmailov, 2020]. In contrast, in the sense of the Equation (3.3), the ensembles of BNN posteriors, where each $p(M_k|\mathcal{D}) = K^{-1}$, can be seen as a trivial case of BMA which combines models and does not give preference to a single solution. Alternatively, when implementing variational inference and combining BNNs, the analogy can be drawn with the simplified version of the adaptive variational Bayes of [Ohn and Lin, 2024], who combine variational posteriors with certain weights and show that these, under certain conditions, can attain optimal contraction rates adaptively.

3.3.4 Ensembles and averages

We compare three model averaging methodologies: deep ensembles of Bayesian neural networks [Ohn and Lin, 2024], stacking and pseudo-BMA based on PSIS-LOO [Yao et al., 2018a]. We do not consider the Bayesian Bootstrap (BB) [Rubin, 1981] motivated by the recent observation that in the settings of modern neural networks, deep ensembles of non-Bayesian NNs and BB are equivalent, and both are often misspecified [Wu and A Williamson, 2024]. Combining several estimates of BNNs can be effective not only when predictions are coming from different models, but also when dealing with several predictions obtained by the same model [Kviman et al., 2022, Ohn and Lin, 2024, Yao et al., 2018b]. This is of particular use for multi-modal posteriors arising in BNNs where different modes could be explored by random initializations [Chang et al., 2019]. Additionally, recall that the ELBO, the objective of variational inference, is a non-convex function, so that the optimum is only local and depends on the starting point.

We note that combining models trained with HMC and VI would be meaningless for several reasons. First of all, training a set of HMC models becomes rather expensive: for instance, training the HMCR20 once takes the same amount of time as 35 trainings of mfVIR20. Second, the estimates of the log pointwise predictive densities for HMC and VI obtained in Section 3.3.2 have different scales

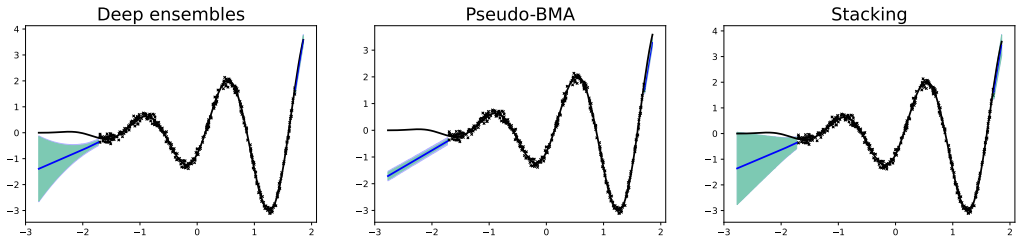
and are not easily compared; in this case, the result of averaging HMC and VI would be equivalent to classical BMA. The reader, nevertheless, interested in ensembles and averages of HMC runs is referred to Appendix A.3 where we show that in this particular example, neither of the methods promoted exploring the multiple modes of the posterior predictive distribution nor did they help improve uncertainty quantification; in Appendix A.3 we additionally provide results of ensembling and averaging the deeper networks.

Now consider the mfVIR20 model and the complement-distributions data of Section 3.2.4. We choose 10 random initialization points, obtain 10 posterior predictive distributions and compute estimated expected log pointwise predictive densities. We then construct ensemble, pseudo-BMA and stacking approximations; Figure 3.6a illustrates the results (Appendix A.3 provides formulas we use to obtain the mean and variance of a deep ensemble). Ensembling and stacking provide subtle results and are superior to pseudo-BMA, which has worse accuracy and fails to capture any uncertainty. Given the nature of the test data we use, the predictions as well as the $\widehat{\text{elpd}}_{\text{loo}}$ estimates may be unreliable. Thus, we create a simpler regression problem in which test data comes from a slightly broader region. Using the classification of Section 3.2.4 and [Farquhar and Gal, 2022], this new scenario could be named as an OOD task with 'related-distributions'. We define a similar synthetic dataset with one-dimensional input and output, to which we add some small noise:

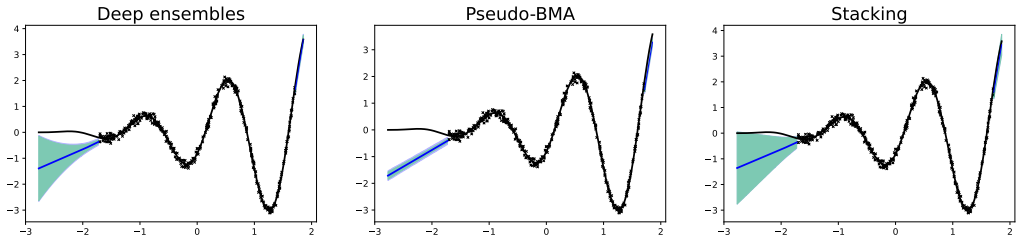
$$\begin{aligned}
\boldsymbol{x} &\sim \text{Unif}([0,1]),\\
\boldsymbol{y} &= \sin(10\boldsymbol{x})\boldsymbol{x}^2 + \boldsymbol{\epsilon},\\
\boldsymbol{\epsilon} &\sim 0.05\mathcal{N}(0,1).
\end{aligned}$$

As before, the input is scaled, unlike the output. The data for training $\mathcal{D}_r$ and testing $\tilde{\mathcal{D}}_r$ consist of $N = 450$ and $\tilde{N} = 50$ observations, respectively, where $\tilde{\mathcal{D}}_r$ comes from the broader than $\mathcal{D}_r$ region, i.e. $(\min_{n=1\dots N}(x_n), \max_{n=1\dots N}(x_n)) \subsetneq (\min_{n=1\dots\tilde{N}}(\tilde{x}_n), \max_{n=1\dots\tilde{N}}(\tilde{x}_n))$. Having 10 posterior predictive distributions of mfVIR20, we compare ensembling, pseudo-BMA and stacking in Figure 3.6b. Whilst the total uncertainty estimates of pseudo-BMA are somewhat adequate, the model uncertainty is underestimated. Both stacking and deep ensembles lead to improved predictive performance and uncertainty quantification, with stacking showing some better gains compared to deep ensembles (see e.g. improved coverage of stacking on the right-hand side of Figure 3.6b).

General summary. Our observations confirm that, similarly to BMA, the pseudo-BMA is not preferable in $\mathcal{M}$-open and $\mathcal{M}$-complete settings. Namely, in 'complement-distributions' and 'related-distributions' experiments, the pseudo-BMA was confirmed to be inferior to stacking and ensembles of BNNs, both in terms of the predictive accuracy and the empirical coverage. While stacking and deep ensembles of BNNs both provided an improvement in accuracy and empirical coverage, stacking was shown to be preferable over the ensembles, especially in terms of uncertainty quantification in the OOD setting.



(a) The complement-distributions task. The pseudo-BMA is worse than DE and stacking, which are very similar to each other.



(b) The related-distributions task. The pseudo-BMA is again worse than the other methodologies. In uncertainty quantification, stacking is better than DE.

Figure 3.6: Predictions obtained by ensembling, stacking and pseudo-BMA when applied to mfVIR20 in the complement-distributions and related-distributions tasks.

3.4 Discussion

The message of an optimist's conclusion to this chapter could question the common belief that the mean-field variational approximations are generally overly restrictive and do not capture the true posterior and the uncertainty well. Indeed, in a variety of experiments considered in Sections 3.2.2 to 3.2.4 and 3.3.4 mfVI overall provided better uncertainty quantification than HMC, and in out-of-distribution settings, the empirical coverage of the latter was close to zero. At the same time, for deeper networks and in out-of-distribution scenarios, the accuracy of mfVI was often comparable to HMC. Nevertheless, we note that for single-layer neural networks, HMC outperformed mfVI only in terms of accuracy. However, in Section 3.2.3 we confirmed that even for slightly deeper networks, the time needed to perform HMC becomes a burden.

Even with increases in computing power, the computational costs of sampling algorithms suggest that it may not be feasible for most modern neural networks and datasets. Moreover, although HMC is often considered as a gold standard, we have seen that this may not be the case for BNNs due to the complexity and multimodality of the posterior.

When considering the extreme out-of-distribution case, we saw that the PSIS-LOO estimate of the log pointwise predictive density can serve as a reliable diagnostic in real-life scenarios where one is required to evaluate the future predictive performance of the model before applying it to the unseen data. Finally, stacking based on rigid model assessment criteria as well as ensembles of BNNs were shown to be a possible solution when dealing with multimodal posteriors, helping

to both improve accuracy and uncertainty quantification even in the single-layer neural networks, when HMC outperformed mfVI. We find that stacked or ensembled variational approximations are competitive to HMC at a much-reduced cost. A possible improvement would be to use hierarchical stacking and combine the models pointwise [Yao et al., 2022] or implement an adaptive variational Bayes framework of [Ohn and Lin, 2024].

Even though the computational burdens make variational inference a very attractive alternative to MCMC, in Section 3.2.2 we saw that the restrictions imposed by the factorized families can obstruct models from effectively learning from the data. This motivates implementing less restrictive variational families in the Variational bow tie neural network of Chapter 4, where we consider a structured mean-field variational family with no assumptions on the independence across layers nor the distributional form of each component. Moreover, this chapter has highlighted the model's sensitivity to architectural choices, namely, width, depth and activation function. To further reduce computational costs whilst improving the accuracy and calibration, one could sparsify the network's weights. While we did not study the empirical performance of different priors in this chapter, motivated by the sensitivity to the choice of depth and the computational costs, we provide details on experiments with Student-t priors in Appendix A.4, and implement a class of hierarchical sparsity-inducing priors in Section 4.2.2. Finally, given the multimodal nature of distributions arising in Bayesian neural networks, we continue to implement ensembles of variational approximations in Section 4.3.6.

Chapter 4

Variational Bayesian Bow Tie Neural Networks with Shrinkage

In Chapter 3 we have observed the sensitivity of existing methods to the architectural choice of the neural network (NN) and brittleness of variational algorithms that impose strong independence and distributional assumptions. In this chapter, we address these issues and focus on advancing approximate inference for Bayesian neural networks (BNNs). Specifically, we improve the stochastic relaxation of the standard feed-forward rectified neural network of [Smith et al., 2021] by introducing sparsity-inducing priors for increased robustness to architectural design and constructing a fast, approximate variational inference (VI) algorithm. Thanks to Polya-Gamma (PG) data augmentation tricks, which render a conditionally linear and Gaussian model, we derive a fast, approximate variational inference algorithm that avoids distributional assumptions and independence across layers.

The work presented in this chapter appears in [Sheinkman and Wade, 2024].[7]

4.1 Introduction

[Smith et al., 2021] introduced a bow tie neural network, where a stochastic relaxation of the rectified linear unit (ReLU) activation function leads to a model amenable to the Polya-Gamma data augmentation trick [Polson et al., 2013] and results in conditionally linear and Gaussian stochastic activations. We advance bow tie neural networks in several ways. First, to improve robustness with increasing width and depth of the networks, we place sparsity-inducing global-local normal-generalized inverse Gaussians (N-GIG) priors [Polson and Scott, 2010] on the weights of the network. Sparsity-inducing priors are known for their ability to improve robustness to overparametrization in Bayesian modeling; they also lead to better calibrated uncertainty and, in certain settings, may recover the sparse structure of the target function [Castillo et al., 2015, Song, 2020, Song and

[7]Currently under review in the ACM Transactions on Probabilistic Machine Learning and the corresponding code implementation could be found on GitHub.

Liang, 2023]. Second, while [Smith et al., 2021] focus on Markov chain Monte Carlo (MCMC), we propose a (block) structured mean-field family for the approximate variational posterior, which is flexible and does not require parametric assumptions on the distributional form of each component as well as on independence across layers. For the chosen family, fast coordinate ascent variation inference (CAVI) [Bishop, 2006] can be performed, with all variational updates available in the closed form. Third, to improve the scalability of the algorithm, we consider two strategies: a stochastic variant that employs subsampling to cope with large data and a post-process node selection algorithm to obtain a sparse posterior that eases the storage and computational burden of predictions. Fourth, we propose improving accuracy and uncertainty estimation by considering ensembles of variational approximations obtained by running several parallel variational algorithms with different random starting points. In this way, our approach accounts for the multimodality of the posterior distributions arising in Bayesian deep models.

Outline of the chapter. Section 4.2 describes the bow tie model with shrinkage priors and implementation of Polya-Gamma data augmentation. In Sections 4.3.1 and 4.3.2 we develop the inference algorithm and in Section 4.3.3 a stochastic variant [Hoffman et al., 2013] of the algorithm is proposed. Further, we derive a variable selection procedure in Section 4.3.4 for faster prediction (Section 4.3.5) and consider ensembles in Section 4.3.6. We evaluate our method on a range of classical regression tasks as well as synthetic regression tasks and demonstrate its competitiveness compared to alternative, well-known Bayesian algorithms in Section 4.4.

4.2 Bayesian augmented bow tie neural network with shrinkage

Section 4.2.1 describes the class of recently proposed bow tie networks [Smith et al., 2021], which are deep generative models that generalize feed-forward rectified linear neural networks with stochastic activations. The shrinkage priors on the weights of the bow tie neural network are introduced in Section 4.2.2, where we consider a class of continuous global-local normal scale-mixtures. Then in Sections 4.2.3 and 4.2.4, the model is augmented with the Polya-Gamma random variables so that the bow tie neural network falls into the class of conditionally conjugate models.

4.2.1 Bow tie neural networks

We begin by considering stochastic relaxation of a neural network with L hidden layers and the widths D_l for $l = 1, \ldots, L$. Let $\boldsymbol{x}_n \in \mathbb{R}^{D_0}$ be the inputs, $\boldsymbol{y}_n \in \mathbb{R}^{D_{L+1}}$ be the outputs and $\boldsymbol{a}_n = \{\boldsymbol{a}_{n,l}\}_{l=1}^{L}$ with $\boldsymbol{a}_{n,l} \in \mathbb{R}^{D_l}$ be the latent activations at each of the L intermediate layers. For notational purposes, assume $\boldsymbol{a}_{n,0} = \boldsymbol{x}_n$.

The model assumes:

$$\boldsymbol{y}_n|\boldsymbol{a}_n, \boldsymbol{x}_n, \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{z}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}\right) \quad \text{for } n = 1, \ldots, N,$$

where for $l = 1, \ldots, L+1$:

$$\boldsymbol{a}_{n,l}|\boldsymbol{z}_{n,l}, \boldsymbol{\theta} \sim \mathcal{N}\left(f(\boldsymbol{z}_{n,l}), \boldsymbol{\Sigma}_l\right), \quad \text{with} \quad \boldsymbol{z}_{n,l} = \boldsymbol{W}_l \boldsymbol{a}_{n,l-1} + \boldsymbol{b}_l. \tag{4.1}$$

Here $f(\boldsymbol{z})$ is a non-linear activation function applied elementwise and the parameters $\boldsymbol{\theta} = (\boldsymbol{W}_l, \boldsymbol{b}_l, \boldsymbol{\Sigma}_l)_{l=1}^{L+1}$ consist of the weights $\boldsymbol{W}_l \in \mathbb{R}^{D_l \times D_{l-1}}$, biases $\boldsymbol{b}_l \in \mathbb{R}^{D_l}$ and covariance matrices $\boldsymbol{\Sigma}_l \in \mathbb{R}^{D_l \times D_l}$.

Note that Equation (4.1) is a stochastic relaxation of the standard feed-forward NN, which is recovered in the limiting case when $\boldsymbol{\Sigma}_l \to \boldsymbol{0}$ for $l = 1, \ldots, L$. Instead of relying on local gradient-based algorithms, [Smith et al., 2021] introduces another relaxation of the model and employs a Polya-Gamma data augmentation trick [Polson et al., 2013] to render the model conditionally linear with Gaussian activations. Specifically, consider the ReLU activation function $f(z) = \max(0, z)$. It can alternatively be written as a product of z and a binary function γ, i.e. $f(z) = \gamma z$ where $\gamma = \mathbf{1}(z > 0)$. In this way, γ determines whether the node is activated ($\gamma = 1$) or not ($\gamma = 0$). In a similar fashion, the additional stochastic relaxation replaces $f(\boldsymbol{z}_{n,l})$ with $\boldsymbol{\gamma}_{n,l} \odot \boldsymbol{z}_{n,l}$, where $\odot$ represents the elementwise product. Consider the logistic function $\sigma(x) = \exp(x)/(1 + \exp(x))$ and introduce the temperature parameter $\mathrm{T} \geq 0$, then

$$\begin{aligned}
\boldsymbol{a}_{n,l}|\boldsymbol{z}_{n,l}, \boldsymbol{\gamma}_{n,l}, \boldsymbol{\theta} &\sim \mathcal{N}\left(\boldsymbol{\gamma}_{n,l} \odot \boldsymbol{z}_{n,l}, \boldsymbol{\Sigma}_l\right), \\
\gamma_{n,l,d} &\overset{ind}{\sim} \begin{cases} \text{Bern}\left(\sigma\left(z_{n,l,d}/\mathrm{T}\right)\right) & \text{for } T > 0, \\ \text{Bern}\left(\text{ReLU}\left(z_{n,l,d}\right)\right) & \text{for } T = 0. \end{cases}
\end{aligned}$$

Thus, the nodes are turned off or on stochastically depending on their input. Note that in the limit as the temperature $T \to 0$, we have that $\gamma_{n,l,d} = \mathbf{1}(z_{n,l,d} > 0)$ and $\boldsymbol{a}_{n,l}|\boldsymbol{z}_{n,l}, \boldsymbol{\gamma}_{n,l}, \boldsymbol{\theta} \sim \mathcal{N}\left(\mathbf{1}(\boldsymbol{z}_{n,l} > 0) \odot \boldsymbol{z}_{n,l}, \boldsymbol{\Sigma}_l\right)$. For $T > 0$, after marginalizing the binary activations, the stochastic activations $\boldsymbol{a}_n$ are distributed as a mixture of two normals:

$$\mathrm{a}_{n,l,d}|z_{n,l,d}, \theta \sim \sigma(z_{n,l,d}/\mathrm{T})\mathcal{N}\left(z_{n,l,d}, \eta_{l,d}^2\right) + (1 - \sigma(z_{n,l,d}/\mathrm{T}))\,\mathcal{N}\left(0, \eta_{l,d}^2\right), \tag{4.2}$$

where the variance $\eta_{l,d}^2$ is the (d,d)th element of $\boldsymbol{\Sigma}_l$, and

$$\begin{aligned}
\mathbb{E}[\mathrm{a}_{n,l,d}|z_{n,l,d}, \theta] &= \mathbb{E}[\mathbb{E}[\mathrm{a}_{n,l,d}|z_{n,l,d}, \gamma_{n,l,d}, \theta]] \\
&= \mathbb{E}[\gamma_{n,l,d} z_{n,l,d}] = \sigma(z_{n,l,d}/\mathrm{T}) z_{n,l,d}, \qquad (4.3) \\
\mathbb{V}(\mathrm{a}_{n,l,d}|z_{n,l,d}, \theta) &= \mathbb{E}[\mathbb{V}(\mathrm{a}_{n,l,d}|z_{n,l,d}, \gamma_{n,l,d}, \theta)] + \mathbb{V}(\mathbb{E}[\mathrm{a}_{n,l,d}|z_{n,l,d}, \gamma_{n,l,d}, \theta]) \\
&= \mathbb{E}[\eta_{l,d}^2] + \mathbb{V}(\gamma_{n,l,d} z_{n,l,d}) \\
&= \eta_{l,d}^2 + z_{n,l,d}^2 \sigma(z_{n,l,d}/\mathrm{T})\left(1 - \sigma(z_{n,l,d}/\mathrm{T})\right). \qquad (4.4)
\end{aligned}$$

We display the conditional distribution of $\mathrm{a}_{n,l,d}$ in Figure 4.1, for different combinations of the temperature parameter T and variance $\eta_{l,d}^2$. The ReLU activation is recovered in the case of $T = 0$ and $\eta_{l,d}^2 = 0$, while other choices of T

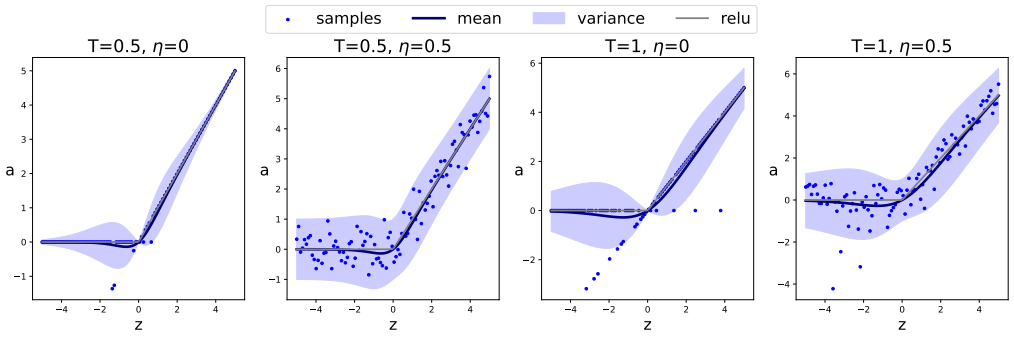


Figure 4.1: Conditional distribution of a given the input z for various settings of the temperature T and noise η, with the conditional mean in Equation (4.3) (solid line), conditional variance in Equation (4.4) (shaded region) and samples from the conditional distribution in Equation (4.2) (points).

and $\eta_{l,d}^2$ generalize the ReLU. The density resembles a bow tie, hence the name of the model.

4.2.2 Shrinkage priors

Prior elicitation in Bayesian neural networks is challenging, as understanding how the high-dimensional weights map to the functions implemented by the network is not trivial. Standard Gaussian priors are often a default choice, also due to their link with ℓ_2 regularization in maximum a posteriori (MAP) inference; indeed, such priors were used in [Smith et al., 2021]. For an overview and discussion on priors in Bayesian neural networks, see Section 1.2.4 and [Fortuin, 2022]. We take an alternative approach to the Gaussian priors of [Smith et al., 2021] in order to sparsify our model. Sparsity-inducing priors ease the problem of storage and computational costs, have been shown to provide improvement in the predictive performance of deep models, and can provide a data-driven approach to selecting the width and depth, easing the difficult task of specifying the network architecture. Such priors generally fall within two classes: 1) the two-group discrete mixture priors with a point mass at zero (referred to as spike-and-slab priors) [George and McCulloch, 1993, Mitchell and Beauchamp, 1988] or 2) shrinkage priors, which employ a single distribution to approximate the spike-and-slab shape, yet are more computationally attractive, as they avoid exploring the space of all possible models. Both types of priors have become widely used in Bayesian deep modeling, due to their high-dimensionality and overparametrization, and are further supported by theoretical guarantees ([Polson and Ročková, 2018, Sun et al., 2022] for BNNs with spike-and-slab priors and [Castillo and Egels, 2024, Lee and Lee, 2022] for BNNs with heavy-tailed shrinkage priors).

In this work, we focus on a class of continuous shrinkage priors, namely, global-local normal scale-mixtures with generalized inverse Gaussian shrinkage priors on the scale parameters, referred to as global-local normal-generalized inverse Gaussian priors [Griffin and Brown, 2021]. Global-local scale-mixtures aim to shrink less important weights whilst leaving large ones, which is achieved through a global parameter controlling the overall shrinkage, with the local parameters al-

lowing deviations at the level of individual nodes [Bhadra et al., 2019, Polson and Scott, 2010]. This choice of priors is also motivated by the theoretical guarantees for high-dimensional regression [Griffin and Brown, 2010, Polson et al., 2013, Song and Liang, 2023], for a survey on global-local shrinkage methods we refer to [Griffin and Brown, 2021].

The N-GIG priors on the weights which connect layer $l-1$ and l (with the convention that inputs layer is indexed by 0) have the following hierarchical structure for $d = 1, \ldots, D_l,\ d' = 1, \ldots, D_{l-1}$:

$$W_{l,d,d'} | \boldsymbol{\psi}_l, \tau_l \sim \mathcal{N}\left(W_{l,d,d'} \mid 0, \tau_l \psi_{l,d,d'}\right), \tag{4.5}$$

$$\psi_{l,d,d'} \sim \text{GIG}\left(\psi_{l,d,d'} | \nu_{\text{loc},l}, \delta_{\text{loc},l}, \lambda_{\text{loc},l}\right), \tag{4.6}$$

$$\tau_l \sim \text{GIG}\left(\tau_l | \nu_{\text{glob}}, \delta_{\text{glob}}, \lambda_{\text{glob}}\right), \tag{4.7}$$

where τ_l is the global shrinkage parameter for layer l and $\psi_{l,d,d'}$ is the local shrinkage parameter for each weight. The generalized inverse Gaussian (GIG) prior has support on $\mathbb{R}_{>0}$ and for $\psi \in \mathbb{R}_{>0}$ is given by:

$$\text{GIG}\left(\psi | \nu, \delta, \lambda\right) \propto \psi^{\nu-1} \exp\left(-\frac{1}{2}(\delta^2/\psi + \lambda^2 \psi)\right),$$

with parameters ν, δ, and λ; for a proper prior, $\nu > 0$ if $\delta = 0$ or $\nu < 0$ if $\lambda = 0$ (additional details on GIG distribution are provided in Appendix B.5) . In Equation (4.6), we allow the GIG parameters for the local scale parameters $\psi_{l,d,d'}$ to vary across layers to adjust local shrinkage for wider layers. Furthermore, to encourage more shrinkage for larger depth and width, we scale the global parameters τ_l with respect to L and the local parameters $\psi_{l,d,d'}$ with respect to D_l (details of our approach are provided Appendix B.4.1).

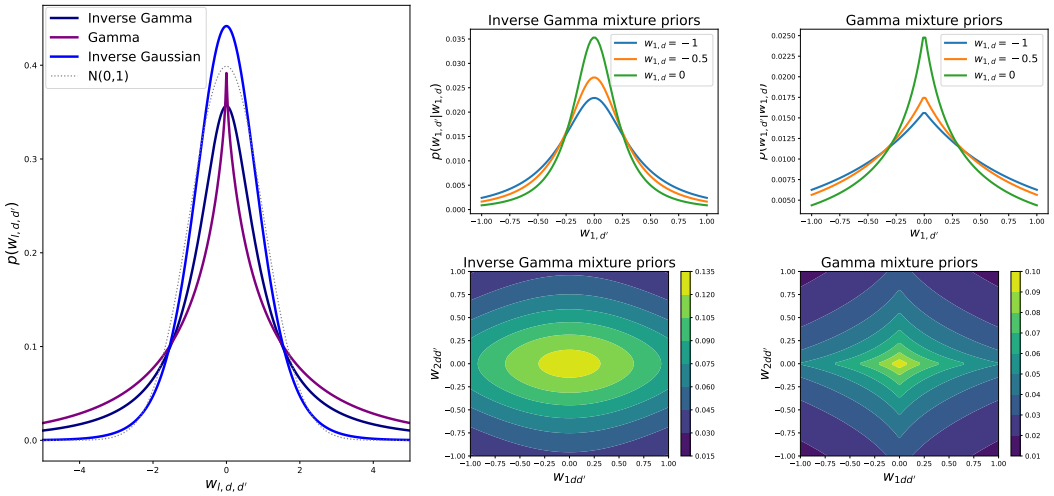
When the global shrinkage parameter τ_l is fixed, examples of the marginal distribution for $w_{l,d,d'}$ include Laplace [Park and Casella, 2008], Student-t (ST) [Tipping, 2001], Normal-Gamma (NG) [Caron and Doucet, 2008, Griffin and Brown, 2010], Normal inverse Gaussian (NIG) [Caron and Doucet, 2008]. Each example has a different tail behaviour, inducing different forms of shrinkage (see Table 4.1 for an overview and Figure 4.2 (a) for a visualization). Note that if the prior is polynomial-tailed, then for large signals the amount of shrinkage is mitigated even given small τ_l [Polson and Scott, 2010]. The global shrinkage parameter τ_l

Table 4.1: Examples within the class of N-GIG priors, when marginal for $w_{l,d,d'}$ is computed when τ_l is fixed.

Marginal	Mixing density	Parameters	Tail behavior
Student-t	Inverse Gamma (IG)	$\nu < 0, \delta > 0, \lambda = 0$	polynomial-tailed
Laplace	Gamma	$\nu = 1, \delta = 0, \lambda$	exponential-tailed
NG	Gamma	$\nu, \delta = 0, \lambda$	exponential-tailed
NIG	Inverse Gaussian (IGauss)	$\nu = \frac{1}{2}, \delta, \lambda$	exponential-tailed

leads to a non-separable penalty for the weights within the same layer, i.e. after

integrating out τ_l, the weights within the same layer are dependent. This is illustrated in the top row of Figure 4.2 (b), which shows how the conditional prior density of one weight varies given different values of another weight within the same layer, for two choices of Inverse-Gamma (IG) and Gamma mixing priors. Instead, across layers the weights are independent, as illustrated by the contour plots for the joint density in the bottom row of Figure 4.2 (b). The bottom row of Figure 4.2 (b) also highlights how the variance depends on the width of the layer, with more hidden units and smaller variance for the second layer compared to the first.



(a) Marginal prior for the weights.

(b) Joint prior for the weights.

Figure 4.2: Illustration of the prior on the weights. (a) the marginal density of the weights for different choices within the GIG family. (b) the conditional prior of the weights within the same layer (top) and joint prior of the weights across layers (bottom) for two choices of IG (left) and Gamma (right) mixing priors.

The opposite effects of varying width and depth in deep neural networks are studied in [Vladimirova et al., 2021]; while depth accentuates a model's non-Gaussianity, the width makes models increasingly Gaussian. Indeed, infinitely wide BNNs are closely related to Gaussian processes (GPs), typically relying on appropriately scaled i.i.d. Gaussian weights [Lee et al., 2018, Matthews et al., 2018, Neal, 1995] and relaxing these assumptions, e.g. through ordering, constraints, heavy tails, or bottlenecks, results in non-Gaussian limits, such as stable processes [Peluchetti et al., 2020], deep GPs [Agrawal et al., 2020] or more exotic processes [Chada et al., 2022, Sell and Singh, 2023]. The sparsity-promoting priors in Equations (4.5) to (4.7) provide a framework for the data to inform on the width and depth of the network.

4.2.3 Polya-Gamma data augmentation

As in [Smith et al., 2021], we employ Polya-Gamma data augmentation to render the model with conditionally linear and Gaussian activations. Data augmentation

strategy based on Polya-Gamma random variables was originally developed to allow for Gibbs sampling in logistic models [Linderman et al., 2016, Polson et al., 2013], and was soon adopted in the context of variational inference [Durante and Rigon, 2019, Scott and Sun, 2013]. The scheme can be applied to derive closed form variational updates in a range of Bayesian models, where binomial likelihoods arise, including conditional mixture networks [Heins et al., 2024], sparse Gaussian process classifiers [Wenzel et al., 2019], deep sigmoid belief networks [Gan et al., 2015] and deep generative models for multiplex networks [Zhou et al., 2024]. For details on data augmentation techniques in Bayesian deep learning, we refer to [Wang et al., 2023].

First, recall the definition of the Polya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$, denoted $\mathrm{PG}(\omega \mid b, c)$ for $\omega \in \mathbb{R}_{\geq 0}$. The random variable $\omega \sim \mathrm{PG}(b, c)$ if

$$\omega \overset{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{\mathrm{Gamma}(\omega \mid b, 1)}{(k - 1/2)^2 + c^2/4\pi^2}.$$

The key identity that we use is:

$$\frac{\exp(z)^a}{(1 + \exp(z))^b} = 2^{-b} \exp(\kappa z) \int_0^{\infty} \exp(-\frac{\omega z^2}{2}) p(\omega) d\omega, \tag{4.8}$$

where $\kappa = a - b/2$ and $p(\omega) = PG(\omega|b, 0)$. The integral is a Gaussian kernel; thus, if $z = \boldsymbol{w}^T \boldsymbol{x}$, conditioned on the latent variable ω, $\boldsymbol{w}$ has a Gaussian distribution and conditioned on $\boldsymbol{w}$, ω has a PG distribution. While to sample from the PG distribution, one can use the alternating series method of [Devroye, 2006], all finite moments of the PG random variables are available in closed form, and that becomes useful for variational Bayes algorithms. Specifically, for $c > 0$

$$\mathbb{E}[\omega] = \frac{b}{2c} \frac{\exp(c) - 1}{1 + \exp(c)}. \tag{4.9}$$

Moreover, the PG distribution is closed under convolution with the same scale parameter; if $\omega_1 \sim \mathrm{PG}(b_1, c)$ and $\omega_2 \sim \mathrm{PG}(b_2, c)$, then $\omega_1 + \omega_2 \sim \mathrm{PG}(b_1 + b_2, c)$.

4.2.4 Augmented model

The model described in Section 4.2.1 augmented with stochastic activations $\boldsymbol{a} = (\boldsymbol{a}_{n,l})$ and binary activations $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_{n,l})$ is:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{\gamma} | \boldsymbol{x}, \boldsymbol{\theta}) = & \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{y}_n | \boldsymbol{z}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}\right) \prod_{n=1}^{N} \prod_{l=1}^{L} \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{\gamma}_{n,l} \odot \boldsymbol{z}_{n,l}, \boldsymbol{\Sigma}_l\right) \\ & \times \prod_{d=1}^{D_l} \frac{\exp(z_{n,l,d}/T)^{\gamma_{n,l,d}}}{1 + \exp(z_{n,l,d}/T)}. \end{aligned}$$

Then using the Equation (4.8), the last term can be written as:

$$\frac{\exp(z_{n,l,d}/T)^{\gamma_{n,l,d}}}{1+\exp(z_{n,l,d}/T)} = 2^{-1}\exp\left(\frac{\kappa_{n,l,d}z_{n,l,d}}{T}\right)\int_0^\infty \exp\left(-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right)p(\omega_{n,l,d})d\omega_{n,l,d},$$

where $\omega_{n,l,d} \sim \text{PG}(1,0)$ and $\kappa_{n,l,d} = \gamma_{n,l,d} - 1/2$. Thus, introducing the additional augmented variables $\boldsymbol{\omega} = (\omega_{n,l,d})$, we arrive at the augmented model:

$$\begin{aligned} p(\boldsymbol{y},\boldsymbol{a},\boldsymbol{\gamma},\boldsymbol{\omega}|\boldsymbol{x},\boldsymbol{\theta}) \propto \prod_{n=1}^{N}\mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{z}_{n,L+1},\boldsymbol{\Sigma}_{L+1}\right)\prod_{n=1}^{N}\prod_{l=1}^{L}\mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{\gamma}_{n,l}\odot\boldsymbol{z}_{n,l},\boldsymbol{\Sigma}_l\right) \\ \times\prod_{d=1}^{D_l}\exp\left(\frac{\kappa_{n,l,d}z_{n,l,d}}{T}\right)\exp\left(-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right)p(\omega_{n,l,d}). \end{aligned}$$

To ease computations, the covariance matrices are assumed to be diagonal $\boldsymbol{\Sigma}_l = \text{diag}(\eta_{l,1}^2,\dots\eta_{l,D_l}^2)$, with variances denoted by $\boldsymbol{\eta}_l = (\eta_{l,1}^2,\dots\eta_{l,D_l}^2)$. Additionally, we assume conjugate priors for the variances $\eta_{l,d}^2 \overset{iid}{\sim} \text{IG}(\alpha_0^h,\beta_0^h)$ for $l = 1,\dots,L$ and $\eta_{L+1,d}^2 \overset{iid}{\sim} \text{IG}(\alpha_0,\beta_0)$ and for the biases $b_{l,d} \overset{iid}{\sim} \mathcal{N}(0,s_0^2)$. Here, we consider different prior parameters α_0^h,β_0^h for the variance terms associated to the hidden layers in comparison to the prior parameters α_0,β_0 for the final layer. In particular, α_0,β_0 are chosen to reflect prior knowledge in the noise, while α_0^h,β_0^h are chosen so that the prior concentrates on small values and realizations of the stochastic activation function are more similar to the ReLU.

By implementing the Polya-Gamma data augmentation scheme, we arrive at the model for which the closed form updates for the coordinate ascent variational inference algorithm can be derived. A graphical model of the bow tie BNN with stochastic relaxation and shrinkage priors is displayed in Figure 4.3, and the posterior distribution over both the model parameters and latent variables is:

$$\begin{aligned} &p(\boldsymbol{y},\boldsymbol{a},\boldsymbol{\gamma},\boldsymbol{\omega},\boldsymbol{W},\boldsymbol{b},\boldsymbol{\eta},\boldsymbol{\psi},\boldsymbol{\tau}\mid\boldsymbol{x}) \propto \prod_{n=1}^{N}\mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{z}_{n,L+1},\boldsymbol{\Sigma}_{L+1}\right)\prod_{l=1}^{L}\prod_{d=1}^{D_l}\text{Bern}\left(\gamma_{n,l,d}|\sigma\left(\frac{z_{n,l,d}}{T}\right)\right) \\ &\times\prod_{n=1}^{N}\prod_{l=1}^{L}\mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{\gamma}_{n,l}\odot\boldsymbol{z}_{n,l},\boldsymbol{\Sigma}_l\right)\prod_{d=1}^{D_l}\exp\left(\frac{\kappa_{n,l,d}z_{n,l,d}}{T}\right)\exp\left(-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right)p(\omega_{n,l,d}) \\ &\times\prod_{d=1}^{D_{L+1}}\text{IG}(\eta_{L+1,d}^2|\alpha_0,\beta_0)\times\prod_{l=1}^{L}\prod_{d=1}^{D_l}\text{IG}(\eta_{l,d}^2|\alpha_0^h,\beta_0^h) \qquad (4.10) \\ &\times\prod_{l=1}^{L+1}\prod_{d=1}^{D_l}\mathcal{N}(b_{l,d}|0,s_0^2)\prod_{d'=1}^{D_{l-1}}\mathcal{N}\left(W_{l,d,d'}|0,\tau_l\psi_{l,d,d'}\right) \\ &\times\prod_{l=1}^{L}\text{GIG}\left(\tau_l|\nu_{\text{glob}},\delta_{\text{glob}},\lambda_{\text{glob}}\right)\prod_{d=1}^{D_l}\prod_{d'=1}^{D_{l-1}}\text{GIG}\left(\psi_{l,d,d'}|\nu_{\text{loc},l},\delta_{\text{loc},l},\lambda_{\text{loc},l}\right). \end{aligned}$$

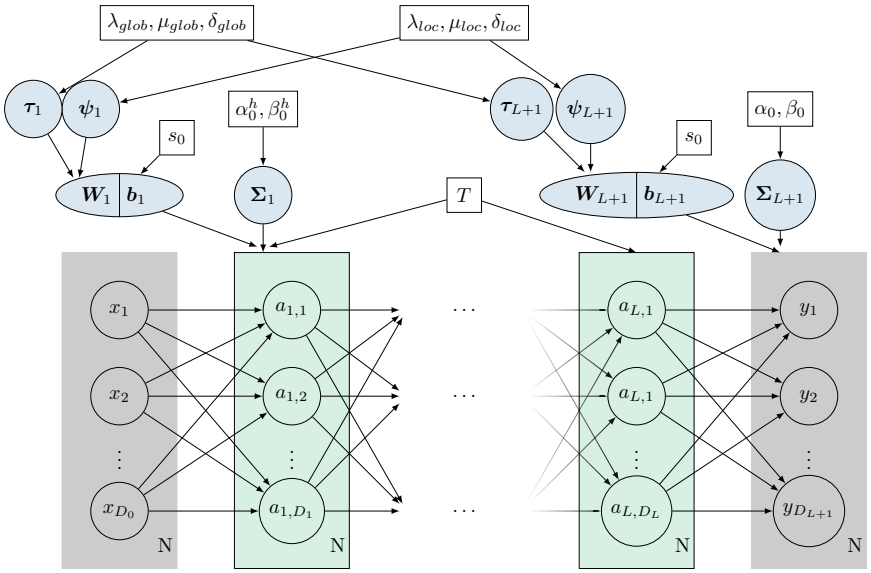


Figure 4.3: Directed Acyclic Graph (DAG) of the model. Global variables are highlighted in blue, and local variables are highlighted in green.

4.3 Inference

In Sections 4.3.1 to 4.3.3, we develop variational inference algorithms for approximating the posterior of the bow tie neural network with shrinkage priors, which leads to what we call a variational bow tie neural network (VBNN). For a broader discussion of variational inference methods in Bayesian neural networks, we refer to Section 2.4.5 and Section 2.4.6. The variational predictive distribution for VBNN is obtained in Section 4.3.5, and additionally, in Sections 4.3.4 and 4.3.6 we explore strategies to improve computational gains and performance in terms of both accuracy and uncertainty estimation.

4.3.1 Variational Bayes

Recall that in variational inference, the true posterior is approximated by a density q that maximizes the evidence lower bound (ELBO) of Equation (2.5) over some variational family. We begin by specifying the (block) mean-field family for the approximate variational posterior:

$$q(\boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\eta}, \boldsymbol{\psi}, \boldsymbol{\tau}) = q(\boldsymbol{a})q(\boldsymbol{\gamma})q(\boldsymbol{\omega})q(\boldsymbol{W}, \boldsymbol{b})q(\boldsymbol{\eta})q(\boldsymbol{\psi})q(\boldsymbol{\tau}). \tag{4.11}$$

Note that the assumption on the family above could be referred to as the structured mean-field assumption. Importantly, unlike existing variational algorithms for BNNs, we do not make any assumptions on the independence of parameters between layers. Observe that after data augmentation, the bow tie neural network falls into the class of conditionally conjugate models. Thus, in order to derive the updates of each factor in the variational family in Equation (4.11), we can use Equation (2.8) and build upon Algorithm 1, which is guaranteed to climb up to the ELBO's local optimum. Moreover, each

variable's variational posterior will belong to the same family of distributions as its complete conditional (for details on the coordinate ascent variational updates in the general case see Section 2.2.2). Below we consider each factor of Equation (4.11) beginning with $q(\boldsymbol{\tau})$ and ending with $q(\boldsymbol{a})$.

Global shrinkage parameters: Using Equation (2.8), the variational posterior for the global shrinkage parameters is:

$$q(\boldsymbol{\tau}) \propto \exp\left(\mathbb{E}\left[\log \prod_{l}^{L+1} \mathrm{GIG}\left(\tau_l | \nu_{\text{glob}}, \delta_{\text{glob}}, \lambda_{\text{glob}}\right) \prod_{d}^{D_l} \prod_{d'}^{D_{l-1}} \mathcal{N}\left(W_{l,d,d'} | 0, \tau_l \psi_{l,d,d'}\right)\right]\right),$$

where the expectation is taken with respect to all variational factors but $q(\boldsymbol{\tau})$. Thus,

$$\begin{aligned} q(\boldsymbol{\tau}) \propto & \prod_{l}^{L+1} \prod_{d}^{D_l} \prod_{d'}^{D_{l-1}} \exp \mathbb{E}\left[\log\left(\frac{1}{\sqrt{\tau_l \psi_{l,d,d'}}} \exp\left(-\frac{W_{l,d,d'}^2}{2\tau_l \psi_{l,d,d'}}\right)\right)\right] \\ & \times \prod_{l}^{L+1} \tau_l^{\nu_{\text{glob}}-1} \exp\left(-\frac{1}{2}\left(\frac{\delta_{\text{glob}}^2}{\tau_l} + \lambda_{\text{glob}}^2 \tau_l\right)\right), \end{aligned}$$

which, after taking the expectation and rearranging the terms, becomes

$$\begin{aligned} q(\boldsymbol{\tau}) &\propto \prod_{l}^{L+1} \tau_l^{\nu_{\text{glob},l}-1} \exp\left(-\frac{1}{2}\left(\delta_{\text{glob},l}^2 \frac{1}{\tau_l} + \lambda_{\text{glob}}^2 \tau_l\right)\right), \\ &= \prod_{l}^{L+1} \mathrm{GIG}\left(\tau_l \mid \nu_{\text{glob},l}, \delta_{\text{glob},l}, \lambda_{\text{glob}}\right), \end{aligned} \tag{4.12}$$

where for $l = 1, \ldots, L+1$,

$$\nu_{\text{glob},l} = \nu_{\text{glob}} - \frac{D_l D_{l-1}}{2} \quad \text{and} \quad \delta_{\text{glob},l} = \sqrt{\delta_{\text{glob}}^2 + \sum_{d}^{D_l} \sum_{d'}^{D_{l-1}} \mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right] \mathbb{E}\left[W_{l,d,d'}^2\right]}.$$

That is, the parameters $\boldsymbol{\tau}$ are independent across layers (and can be updated in parallel) with a GIG variational posterior given by Equation (4.12).

The detailed calculation of the remaining components of the variational posterior is given in the Appendix B.1, where we obtain the following update steps.

Local shrinkage parameters: the parameters $\boldsymbol{\psi}$ are independent across and within layers (and can be updated in parallel) with a GIG variational posterior:

$$q(\boldsymbol{\psi}) = \prod_{l}^{L+1} \prod_{d}^{D_l} \prod_{d'}^{D_{l-1}} \mathrm{GIG}\left(\psi_{l,d,d'} \mid \nu_{\text{loc},l,d,d'}, \delta_{\text{loc},l,d,d'}, \lambda_{\text{loc},l}\right), \tag{4.13}$$

where for $l = 1, \ldots, L+1,\ d = 1, \ldots, D_l,\ d' = 1, \ldots, D_{l-1}$,

$$\nu_{\text{loc},l,d,d'} = \nu_{\text{loc},l} - \frac{1}{2} \quad \text{and} \quad \delta_{\text{loc},l,d,d'} = \sqrt{\mathbb{E}\left[\frac{1}{\tau_l}\right] \mathbb{E}\left[W_{l,d,d'}^2\right] + \delta_{\text{loc},l}^2}.$$

Covariance matrix: the diagonal elements of the covariance matrix $\boldsymbol{\eta}_l$ are independent across and within layers (and can be updated in parallel) with an Inverse-Gamma variational posterior:

$$q(\boldsymbol{\eta}) = \prod_{l}^{L+1} \prod_{d}^{D_l} \text{IG}(\eta_{l,d}^2 | \alpha_{l,d}, \beta_{l,d}), \tag{4.14}$$

where for the hidden layers $l = 1, \ldots, L$, the updated variational parameters for $d = 1, \ldots, D_l$ are given by

$$\begin{aligned}
\alpha_{l,d} &= \alpha_0^h + \frac{N}{2}, \\
\beta_{l,d} &= \beta_0^h + \frac{1}{2} \sum_{n}^{N} \left(\mathbb{E}\left[\mathrm{a}_{n,l,d}\right] - \mathbb{E}\left[\gamma_{n,l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \right)^2 + \mathbb{E}\left[a_{n,l,d}^2\right] - \mathbb{E}\left[\mathrm{a}_{n,l,d}\right]^2 \\
&\quad + \frac{1}{2} \sum_{n}^{N} \mathbb{E}\left[\gamma_{n,l,d}\right] \text{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \widetilde{\boldsymbol{a}}_{n,l-1}^T\right] \right) \\
&\quad - \frac{1}{2} \sum_{n}^{N} \mathbb{E}\left[\gamma_{n,l,d}\right]^2 \text{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\right] \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}^T\right] \right).
\end{aligned}$$

And for the final layer, the updated variational parameters for $d = 1, \ldots, D_{L+1}$ are given by

$$\begin{aligned}
\alpha_{L+1,d} &= \alpha_0 + \frac{N}{2}, \\
\beta_{L+1,d} &= \beta_0 + \frac{1}{2} \sum_{n}^{N} \left(y_{n,d} - \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}\right] \mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}] \right)^2 \\
&\quad + \frac{1}{2} \sum_{n}^{N} \text{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}^T \widetilde{\boldsymbol{W}}_{L+1,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L} \widetilde{\boldsymbol{a}}_{n,L}^T\right] \right) \\
&\quad - \frac{1}{2} \sum_{n}^{N} \text{Tr}\left(\mathbb{E}[\widetilde{\boldsymbol{W}}_{L+1,d}]^T \mathbb{E}[\widetilde{\boldsymbol{W}}_{L+1,d}] \mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}] \mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}^T] \right).
\end{aligned}$$

Note that in the above, $\boldsymbol{W}_{l,d}$ represents the d-th row of the weight matrix $\boldsymbol{W}_l$. Additionally, for $l = 1, \ldots, L+1$ we introduce the notation $\widetilde{\boldsymbol{W}}_{l,d} = (b_{l,d}, \boldsymbol{W}_{l,d})$ and $\widetilde{\boldsymbol{W}} = (\boldsymbol{b}, \boldsymbol{W})$, and let the vector $\widetilde{\boldsymbol{a}}_{n,l}$ represent the stochastic activation augmented with an entry of one, i.e. $\widetilde{\boldsymbol{a}}_{n,l} = (1, \boldsymbol{a}_{n,l}^T)^T$.

Weights and biases: the weights and biases are independent across layers and within layer, independent across the D_l regression problems, with a Gaussian variational posterior:

$$q(\boldsymbol{b}, \boldsymbol{W}) = \prod_{l}^{L+1} \prod_{d}^{D_l} \mathcal{N}\left(\widetilde{\boldsymbol{W}}_{l,d} | \boldsymbol{m}_{l,d}, \boldsymbol{B}_{l,d}\right), \tag{4.15}$$

where for the hidden layers $l = 1, \ldots, L$, the updated variational parameters for $d = 1, \ldots, D_l$ are given by

$$\boldsymbol{B}_{l,d}^{-1} = \boldsymbol{D}_{l,d}^{-1} + \sum_{n}^{N} \left(\frac{1}{T^2} \mathbb{E}\left[\omega_{n,l,d}\right] + \mathbb{E}\left[\eta_{l,d}^{-2}\right] \mathbb{E}\left[\gamma_{n,l,d}\right] \right) \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \widetilde{\boldsymbol{a}}_{n,l-1}^T\right],$$

$$\boldsymbol{m}_{l,d}^T = \boldsymbol{B}_{l,d} \left(\sum_{n}^{N} \mathbb{E}\left[\eta_{l,d}^{-2}\right] \mathbb{E}\left[\gamma_{n,l,d}\right] \mathbb{E}\left[\mathrm{a}_{n,l,d} \widetilde{\boldsymbol{a}}_{n,l-1}\right] + \frac{1}{T} \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \left(\mathbb{E}\left[\gamma_{n,l,d}\right] - \frac{1}{2} \right) \right),$$

and for the final layer, the updated variational parameters for $d = 1, \ldots, D_{L+1}$ are given by

$$\boldsymbol{B}_{L+1,d}^{-1} = \boldsymbol{D}_{L+1,d}^{-1} + \mathbb{E}\left[\eta_{L+1,d}^{-2}\right] \sum_{n}^{N} \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L} \widetilde{\boldsymbol{a}}_{n,L}^T\right],$$

$$\boldsymbol{m}_{L+1,d}^T = \boldsymbol{B}_{L+1,d} \mathbb{E}\left[\eta_{L+1,d}^{-2}\right] \left(\sum_{n}^{N} y_n \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right] \right),$$

where for $l = 1, \ldots, L+1$ and $d = 1, \ldots, D_l$,

$$\boldsymbol{D}_{l,d}^{-1} = \mathrm{diag}\left(s_0^{-2}, \mathbb{E}\left[\tau_l^{-1}\right] \mathbb{E}\left[\psi_{l,d,1}^{-1}\right], \ldots, \mathbb{E}\left[\tau_l^{-1}\right] \mathbb{E}\left[\psi_{l,d,D_{l-1}}^{-1}\right] \right).$$

Polya-Gamma augmented variables: $\boldsymbol{\omega}$ are independent across observations $n = 1, \ldots, N$, layers $l = 1, \ldots, L$, and width $d = 1, \ldots, D_l$, with a Polya-Gamma variational posterior:

$$q(\boldsymbol{\omega}) = \prod_{n}^{N} \prod_{l}^{L} \prod_{d}^{D_l} \mathrm{PG}(\omega_{n,l,d} | 1, \mathrm{a}_{n,l,d}), \tag{4.16}$$

with updated variational parameters:

$$\mathrm{a}_{n,l,d} = \frac{1}{T} \sqrt{\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \widetilde{\boldsymbol{a}}_{n,l-1}^T\right] \right) \right)}.$$

Note that simulating from or evaluating the density of the PG is not necessary, and the CAVI updates of the other parameters only require computing the expectation of $\boldsymbol{\omega}$ with respect to the variational posterior in Equation (4.16), which is straightforward to compute (Equation (4.9)).

Binary activations: $\boldsymbol{\gamma}$ are independent across observations $n = 1, \ldots, N$, layers $l = 1, \ldots, L$, and width $d = 1, \ldots, D_l$, with a Bernoulli variational posterior:

$$q(\boldsymbol{\gamma}) = \prod_{n}^{N} \prod_{l}^{L} \prod_{d}^{D_l} \text{Bern}\left(\gamma_{n,l,d} | \rho_{n,l,d}\right), \tag{4.17}$$

with

$$\begin{aligned}\rho_{n,l,d} = \sigma \Bigg(& -\frac{\mathbb{E}\left[\eta_{l,d}^{-2}\right]}{2} \operatorname{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right) \\ & + \mathbb{E}\left[\eta_{l,d}^{-2}\right] \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \mathrm{a}_{n,l,d}\right] + \frac{1}{T} \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \Bigg).\end{aligned}$$

The parameters $\rho_{n,l,d}$ represent the posterior probability that the node is active and are illustratedfor the toy example of Section 4.4.1 in Figure 4.4.

Stochastic activations: $\boldsymbol{a}$ are independent across observations $n = 1, \ldots, N$ and conditionally Gaussian given the previous layer with variational posterior:

$$q(\boldsymbol{a}) = \prod_{n=1}^{N} \prod_{l=1}^{L} \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l} \boldsymbol{a}_{n,l-1}, \boldsymbol{S}_{n,l}\right), \tag{4.18}$$

where we denote $\boldsymbol{a}_{n,0} \coloneqq \boldsymbol{x}_n$, $\boldsymbol{S}_{n,L} \coloneqq \boldsymbol{S}_L$.

For $l = 1, \ldots, L$ introduce $\hat{\boldsymbol{\Sigma}}_l^{-1} = \operatorname{diag}\left(\mathbb{E}\left[\eta_{l,1}^{-2}\right], \ldots, \mathbb{E}\left[\eta_{l,D_l}^{-2}\right]\right)$, the updated variational parameters for $n = 1, \ldots, N$ and $l = 1, \ldots, L-1$ are

$$\begin{aligned}\boldsymbol{S}_{n,l}^{-1} = & \hat{\boldsymbol{\Sigma}}_l^{-1} - \boldsymbol{M}_{n,l+1}^T \boldsymbol{S}_{n,l+1}^{-1} \boldsymbol{M}_{n,l+1} \\ & + \sum_{d=1}^{D_{l+1}} \left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right] \mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2} \mathbb{E}\left[\omega_{n,l+1,d}\right] \right) \mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T \boldsymbol{W}_{l+1,d}\right], \\ \boldsymbol{t}_{n,l} = & \boldsymbol{S}_{n,l} \Big(\boldsymbol{M}_{n,l+1}^T \boldsymbol{S}_{n,l+1}^{-1} \boldsymbol{t}_{n,l+1} + \hat{\boldsymbol{\Sigma}}_l^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right] \odot \mathbb{E}\left[\boldsymbol{b}_l\right] \\ & + \frac{1}{T} \sum_{d=1}^{D_{l+1}} \mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T\right] \left(\mathbb{E}\left[\gamma_{n,l+1,d}\right] - \frac{1}{2} \right) \\ & - \sum_{d=1}^{D_{l+1}} \left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right] \mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2} \mathbb{E}\left[\omega_{n,l+1,d}\right] \right) \mathbb{E}\left[\boldsymbol{W}_{l+1,d} b_{l+1,d}\right] \Big), \\ \boldsymbol{M}_{n,l} = & \boldsymbol{S}_{n,l} \hat{\boldsymbol{\Sigma}}_l^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right] \mathbf{1}_{D_{l-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_l\right].\end{aligned}$$

And for the final layer with $n = 1, \ldots, N$ and $l = L$,

$$\begin{aligned}\boldsymbol{S}_L^{-1} = & \hat{\boldsymbol{\Sigma}}_L^{-1} + \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right] \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T \boldsymbol{W}_{L+1,d}\right], \\ \boldsymbol{t}_{n,L} = & \boldsymbol{S}_L \Big(\hat{\boldsymbol{\Sigma}}_L^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right] \odot \mathbb{E}\left[\boldsymbol{b}_L\right]\end{aligned}$$

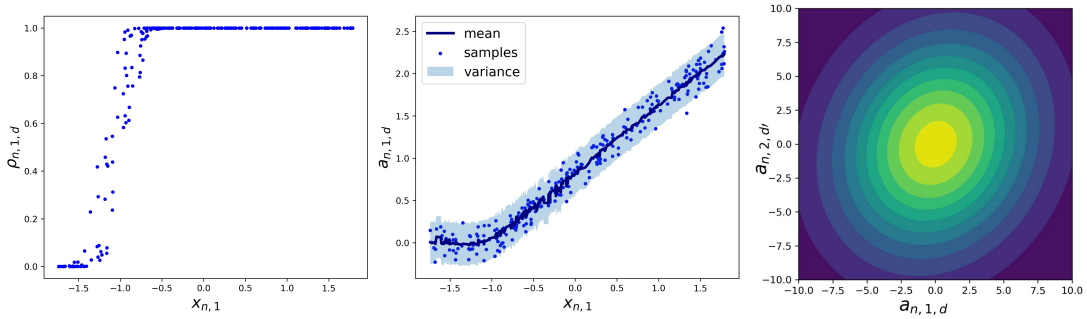


Figure 4.4: An illustration of the variational posterior of the binary and stochastic activations. The variational posterior of $\gamma_{n,1,d}$ (on the left) and $a_{n,1,d}$ (in the middle), both as a function of $x_{n,1}$ across all observations, along with the joint distribution of $(a_{n,1,d}, a_{n,2,d'})$ (on the right) in the case of the toy example of Section 4.4 for particular values of d, d', n.

$$+\sum_{d=1}^{D_{l+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\left(-\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T b_{L+1,d}\right] + \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\right] y_{n,d}\right)\Bigg),$$

$$\boldsymbol{M}_{n,L} = \boldsymbol{S}_L \hat{\boldsymbol{\Sigma}}_L^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right] \mathbf{1}_{D_{L-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_L\right].$$

Figure 4.4 illustrates on the toy example of Section 4.4.1 how the variational posterior of the stochastic activations (middle) resembles a smoothed, noisy ReLU. Due the independence assumption between the stochastic and binary activations, the potentially bimodal bow tie distribution (Equation (4.2)) is approximated with an unimodal Gaussian in the variational framework, which may better approximate the true posterior when the temperature is not too large relative to the noise (see Figure 4.1). In addition, the proposed approximation has the advantage of avoiding explicit assumptions of independence between layers, enabling it to capture the dependence between the stochastic activations across layers, as illustrated for the toy example in Figure 4.4 (right).

The corresponding optimization objective, i.e. the ELBO in Equation (2.5), is available in closed form and provided in the Appendix B.2.

4.3.2 VI with EM

The hyperparameters can play a crucial role in Bayesian neural networks. When dealing with the sparsity-inducing priors setting an excessively large scale parameter weakens the shrinkage effects, whilst choosing a scale parameter that is too small may wipe out the effects of the important hidden nodes. Manually picking suitable values is challenging, and instead, we seek a more efficient strategy, utilizing the similarity between the variational and expectation-maximization algorithms. Specifically, we investigate the hybrid scheme combining VI with an EM step [Osborne et al., 2022] so that the steps of the CAVI algorithm proceed with the EM update to set the hyperparameter for global shrinkage variable $\boldsymbol{\tau}$. Due to weak identifiability, we do not jointly update global and local hyperparameters. Let h_{glob} represent δ_{glob} or λ_{glob} and consider the ELBO treated as a function of h_{glob}, then the optimal values as approximate MAP estimates are:

$$h_{\text{glob}} = \arg\max \mathbb{E}_{\text{glob}}[\text{ELBO}],$$

where

$$\begin{aligned}\mathbb{E}_{\text{glob}}[\text{ELBO}] &= \mathbb{E}\left[\sum_{l=1}^{L+1} \log(\text{GIG}(\tau_l|\nu_{\text{glob}}, \delta_{\text{glob}}, \lambda_{\text{glob}})\right] \\ &= (L+1)\left(\nu_{\text{glob}}\left(\log(\lambda_{\text{glob}}) - \log(\delta_{\text{glob}})\right) - \log\left(2K_{\nu_{\text{glob}}}(\lambda_{\text{glob}}\delta_{\text{glob}})\right)\right) \\ &+ \sum_{l=1}^{L+1}(\nu_{\text{glob}} - 1)\mathbb{E}\left[\log \tau_l\right] - \frac{1}{2}\left(\delta_{\text{glob}}^2\mathbb{E}\left[\frac{1}{\tau_l}\right] + \lambda_{\text{glob}}^2\mathbb{E}\left[\tau_l\right]\right).\end{aligned}$$

In the case of the IG priors, one's aim is to set optimal δ_{glob}, in the case of the Gamma and IGauss priors, the parameters of interest are λ_{glob}. We provide specific examples of the shrinkage parameters and the corresponding optimal values in Appendix B.5.2. The result of combining CAVI and the EM algorithm is described in Algorithm 5, where we note that for $l = 1, \ldots, L+1$, $d = 1, \ldots, D_l$, $d' = 1, \ldots, D_{l-1}$ parameters $\nu_{\text{glob},l}, \nu_{\text{loc},l,d,d'}$ and $\alpha_{l,d}$ are updated only during the iteration of the algorithm. Algorithm 5 is complemented with Figure 4.5 illustrating one iteration of the algorithm, where we note that $\boldsymbol{\tau}, \boldsymbol{\psi}, \boldsymbol{\Sigma}, \boldsymbol{b}, \boldsymbol{W}$ are global variables, and $\boldsymbol{\omega}, \boldsymbol{a}$ and $\boldsymbol{\gamma}$ are local variables.

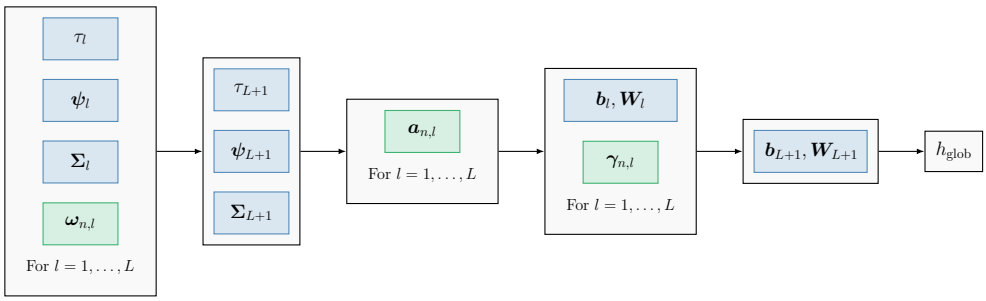


Figure 4.5: Horizontal flow-chart illustrating the order in which parameters of the BNN are updated during one loop of the CAVI with EM algorithm. Similar to Figure 4.3, global variables are highlighted in blue, and local variables are in green.

The resulting Algorithm 5 scales linearly with the number of hidden layers and the number of samples but cubically with the number of hidden units; the computational complexities corresponding to individual factors of the variational family are provided in Appendix B.1. In the following subsections, we discuss two strategies to improve scalability. First, to handle large N, CAVI can be combined with subsampling through stochastic VI [Hoffman et al., 2013], and second, a post-processing node selection algorithm is proposed to obtain a sparse variational posterior for faster predictive inference.

4.3.3 Stochastic Variational Inference

Algorithm 5 CAVI with EM

Require: Initialize hyperparameters
  while ELBO has not converged **do**
    for $l = 1, \dots, L$ **do**
      update $\delta_{\text{glob},l}$ {parameter of τ_l}
      update $\delta_{\text{loc},l,d,d'}$ for $d = 1 \dots D_l,\ d' = 1 \dots D_{l-1}$ {parameter $\psi_{l,d,d'}$}
      update $\beta_{l,d}$ for $d = 1 \dots D_l$ {parameter of $\eta_{l,d}$}
      update $\mathrm{a}_{n,l,d}$ for $d = 1 \dots D_l,\ n = 1 \dots N$ {parameter of $\omega_{n,l,d}$}
    end for
    update $\delta_{\text{glob},L+1}$ {parameter of τ_{L+1}}
    update $\delta_{\text{loc},L+1,d,d'}$ for $d = 1 \dots D_{L+1},\ d' = 1 \dots D_L$ {parameter $\psi_{L+1,d,d'}$}
    update $\beta_{L+1,d},\ d = 1 \dots D_{L+1}$ {parameter of $\eta_{L+1,d}$}
    for $l = 1, \dots, L$ **do**
      update $\boldsymbol{S}_{n,l},\ \boldsymbol{M}_{n,l},\ \boldsymbol{t}_{n,l}$ for $n = 1 \dots N$ {parameters of $\boldsymbol{a}_{n,l}$}
    end for
    for $l = 1, \dots, L$ **do**
      update $\boldsymbol{B}_{l,d},\ \boldsymbol{m}_{l,d}$ for $d = 1 \dots D_l$ {parameters of $(b_{l,d}, \boldsymbol{W}_{l,d})$}
      update $\rho_{n,l,d}$ for $d = 1 \dots D_l,\ n = 1 \dots N$ {parameter of $\gamma_{n,l,d}$}
    end for
    update $\boldsymbol{B}_{L+1,d},\ \boldsymbol{m}_{L+1,d}$ for $d = 1 \dots D_{L+1}$ {parameters of $(b_{L+1,d}, \boldsymbol{W}_{L+1,d})$}
    update h_{glob} {EM for global hyperparameter}
  end while

At each iteration, CAVI has to cycle through the entire data set, which can be computationally expensive and inefficient for large sample sizes. An alternative to coordinate ascent is gradient-based optimization, which extends the algorithm by employing stochastic variational inference (SVI) and subsampling. First, recall that the variational posterior of the latent variables $\boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega}$ factorizes across data points, that is

$$q(\boldsymbol{a})q(\boldsymbol{\gamma})q(\boldsymbol{\omega}) = \prod_{n=1}^{N} q(\boldsymbol{a}_n)q(\boldsymbol{\gamma}_n)q(\boldsymbol{\omega}_n).$$

To highlight the need for stochastic VI, we observe that for each layer $l = 1, \dots, L$ the ordinary coordinate ascent needs to iterate through N local variational parameters corresponding to variables $\boldsymbol{a}_{n,l}, \boldsymbol{\gamma}_{n,l}$ and $\boldsymbol{\omega}_{n,l}$ with $\boldsymbol{\gamma}_{n,l}$ and $\boldsymbol{\omega}_{n,l}$ both having D_l variational parameters to update, and $\boldsymbol{a}_{n,l}$ having of order D_l^2 variational parameters; for large N, this can clearly become computational burdensome. To overcome this, stochastic VI uses the coordinate ascent variational updates obtained in Section 4.3 to set the local, and the intermediate global parameters are obtained by following the noisy natural gradient of ELBO (see Section 2.2.3 and [Hoffman et al., 2013, Sato, 2001]).

At each iteration t, the Algorithm 6 first uniformly samples a collection of indices $S(t)$ without replacement, where the computational gains are obtained as long as $|S(t)| \ll N$. For each index $s \in S(t)$, the local variational parameters are optimized in a coordinate ascent manner with updates in Equations (4.16)

to (4.18), and the convergence is monitored with noisy estimates of the local ELBO, that is

$$\mathbb{E}\left[\log p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega} | \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma})\right] - \mathbb{E}\left[\log q(\boldsymbol{a})\right] - \mathbb{E}\left[\log q(\boldsymbol{\gamma})\right] - \mathbb{E}\left[\log q(\boldsymbol{\omega})\right],$$

the estimate of which is provided Appendix B.2.2. Next, the global parameters, $\boldsymbol{W}, \boldsymbol{b}$ and $\boldsymbol{\eta}$, are updated via a linear combination of previous values and intermediate updates

$$(1 - \ell_t)\text{parameter}^{(t-1)} + \ell_t \times \text{intermediate parameter}^{(t)}, \tag{4.19}$$

where ℓ_t is the learning rate and the intermediate parameters are obtained via Equations (4.14) and (4.15) but with the sufficient statistics, which involve sums over N, replaced with the scaled sums over $S(t)$. Note that $-(\boldsymbol{\alpha} + 1)$ is the natural parameter of global variable $\boldsymbol{\eta}$, so that $\boldsymbol{\alpha}$ is only updated once and does not require an intermediate step. Further, the vector of natural parameters for $(\boldsymbol{W}, \boldsymbol{b})$ is $(\boldsymbol{B}^{-1}\boldsymbol{m}^T, -\boldsymbol{B}^{-1}/2)$ and $-\boldsymbol{\beta}^{-1}$ is the second natural parameter of $\boldsymbol{\eta}$, this implies that for $l = 1, \ldots, L+1$ the Equation (4.19) becomes

$$\begin{aligned}
\boldsymbol{B}_{l,d}^{(t)} &= \left((1 - \ell_t) \times (\boldsymbol{B}_{l,d}^{(t-1)})^{-1} + \ell_t \times \hat{\boldsymbol{B}}_{l,d}^{-1}\right)^{-1}, \\
\boldsymbol{m}_{l,d}^{(t)} &= \left(\boldsymbol{B}_{l,d}^{(t)} \left((1 - \ell_t)(\boldsymbol{B}_{l,d}^{(t-1)})^{-1}(\boldsymbol{m}_{l,d}^{(t-1)})^T + \ell_t \hat{\boldsymbol{B}}_{l,d}^{-1} \hat{\boldsymbol{m}}_{l,d}^T\right)\right)^T, \\
\beta_{l,d}^{(t)} &= \left((1 - \ell_t) \times (\beta_{l,d}^{(t-1)})^{-1} + \ell_t \times \hat{\beta}_{l,d}^{-1}\right)^{-1}.
\end{aligned}$$

Additional details and the step-by-step algorithm are provided in Appendix B.2.2. The convergence of the stochastic gradient descent depends on the choice of the step sizes ℓ_t in the Robbins-Monro sequence [Robbins and Monro, 1951], and we follow [Hoffman et al., 2013], who set

$$\ell_t = (1 + t)^{-k}, \; k \in (0.5, 1], \tag{4.20}$$

where k is the forgetting rate. We monitor the convergence of the algorithm by obtaining noisy estimates of the ELBO (provided in Appendix B.2.2) where the sums over $n = 1, \ldots, N$ are replaced with the scaled sums over $n \in S$.

In the future, we could further improve the algorithm by employing an adaptive learning rate which is based on realisations of a noisy estimate of the natural gradient of ELBO with respect to global variational parameters and moving averages [Ranganath et al., 2013, Schaul et al., 2013]. Alternatively, instead of changing the rate ℓ_t, one could adapt the mini-batch size based on the estimated gradient noise covariance and the magnitude of the gradient [Balles et al., 2017].

4.3.4 Inferring the network structure

The choice of the network architecture has significant practical implications on the generalization of the model, and so sparsity-promoting priors for the network weights have emerged as a promising approach for increased robustness to the choice of architecture. When designing the VBNN, we focus on a class of

Algorithm 6 SVI for bow tie neural network

Require: initialize global hyperparameters
  while noisy estimate of ELBO has not converged **do**
    set ℓ_t
    uniformly sample indices $S(t)$ without replacement
    initialize local variational parameters
    for $l = 1, \ldots, L+1$ **do**
      update $\delta^{(t)}_{\text{glob},l}$ {parameters of τ_l}
      update $\delta^{(t)}_{\text{loc},l,d,d'}$ for $d = 1 \ldots D_l,\ d' = 1 \ldots D_{l-1}$ {parameter $\psi_{l,d,d'}$}
    end for
    while local parameters have not converged **do**
      for $s \in S(t)$ **do**
        initialize $\boldsymbol{S}_{s,l},\ \boldsymbol{M}_{s,l},\ \boldsymbol{t}_{s,l},\ \rho_{s,l,d}\ d = 1 \ldots D_l$ for $l = 1, \ldots, L$
        for $l = 1, \ldots, L$ **do**
          update $A^{(t)}_{s,l,d}$ for $d = 1 \ldots D_l$ {parameter of $\omega_{s,l,d}$}
          update $\boldsymbol{S}^{(t)}_{s,l},\ \boldsymbol{M}^{(t)}_{s,l},\ \boldsymbol{t}^{(t)}_{s,l}$ {parameters of $\boldsymbol{a}_{s,l}$}
          update $\rho^{(t)}_{s,l,d}$ for $d = 1 \ldots D_l,$ {parameter of $\gamma_{s,l,d}$}
        end for
      end for
    end while
    for $l = 1, \ldots, L+1$ **do**
      find $\hat{\beta}^{-1}_{l,d}$ and update $\boldsymbol{\beta}^{(t)}_{l,d}$ for $d = 1 \ldots D_l$ {parameter of $\eta_{l,d}$}
      find $\hat{\boldsymbol{B}}^{-1}_{l,d},\ \hat{\boldsymbol{B}}^{-1}_{l,d}\hat{m}^T_{l,d}$ and update $\boldsymbol{B}^{(t)}_{l,d},\ m^{(t)}_{l,d}$ for $d = 1 \ldots D_l$ $\{(b_{l,d}, \boldsymbol{W}_{l,d})\}$
    end for
  end while

continuous shrinkage priors, which results in more tractable computations, yet also implies non-zero posterior means and does not lead to automatic network architecture selection. To address this, we consider a post-processing node selection algorithm to obtain a sparse variational posterior, which both aids interpretability and improves scalability of predictive inference (discussed in Section 4.3.5) In Bayesian linear regression with shrinkage priors, several post-processing methods have been proposed to yield a sparse solution (see e.g.[Griffin, 2024, Li and Pati, 2017, Piironen et al., 2020]). The method known as decoupling shrinkage and selection (DSS) [Hahn and Carvalho, 2015] obtains sparse estimates of the weights by minimizing the sum of the predictive loss function with a parsimony-inducing penalty. An alternative approach is the penalized credible regions (PenCR) method [Zhang et al., 2021b], which identifies the "sparsest" solution in posterior credible regions corresponding to different levels; it is shown to perform well in the case of global-local shrinkage priors and under certain assumptions, PenCR produces the same results as DSS. Similarly, we propose to make use of credible intervals to select nodes. Following [Li and Lin, 2010], we implement an automatic credible interval criterion which selects a node as long as its credible interval does not cover zero. Specifically, recall that the variational posterior of the weights is $W_{l,d,d'} \sim \mathcal{N}(m^W_{l,d,d'}, B^W_{l,d,d'})$ for $l = 1, \ldots, L, d =$

$1, \ldots, D_l, d' = 1, \ldots, D_{l-1}$, where $\boldsymbol{m}_{l,d}^W$ and $\boldsymbol{B}_{l,d}^W$ denote the subsets of the mean $\boldsymbol{m}_{l,d}$ and covariance matrix $\boldsymbol{B}_{l,d}$ corresponding to the weights. Then, we obtain sparse weights $\widehat{W}_{l,d,d'}$ with sparse variational distribution $\widehat{q}(b_{l,d}, \widehat{\boldsymbol{W}}_{l,d})$ for some $l \in \mathcal{L} \subseteq \{1, \ldots, L+1\},\ d \in \mathcal{D}_l \subseteq \{1, \ldots, D_l\}, d' \in \mathcal{D}_{l-1} \subseteq \{1, \ldots, D_{l-1}\}$, defined by setting:

$$\widehat{W}_{l,d,d'} \sim \begin{cases} \mathcal{N}\left(m_{l,d,d'}^W, \left(B_{l,d}^W\right)_{d'd'}\right) & \text{if } \max\left(Q(W_{l,d,d'} > 0), Q(W_{l,d,d'} < 0)\right) \geq \kappa, \\ \delta_0 & \text{otherwise,} \end{cases}$$

where $Q(W_{l,d,d'} < 0) = 1 - Q(W_{l,d,d'} > 0) = \Phi(-m_{l,d,d'}^W / \sqrt{(B_{l,d}^W)_{d'd'}})$.

The threshold κ is chosen to control the Bayesian false discovery rate, which is calculated as

$$\widehat{\mathrm{FDR}}(\kappa) = \frac{\sum_{l,d,d'} (1 - \mathcal{Q}_{l,d.d'}) \mathbf{1}(\mathcal{Q}_{l,d.d'} > \kappa)}{\sum_{l,d,d'} \mathbf{1}(\mathcal{Q}_{l,d.d'} > \kappa)},$$

with $\mathcal{Q}_{l,d.d'} = \max\left(Q(W_{l,d,d'} > 0), Q(W_{l,d,d'} < 0)\right)$. Specifically, for a specified error rate α, κ is set to satisfy $\widehat{\mathrm{FDR}}(\kappa) < \alpha$. Algorithm 7 describes the node selection procedure which begins with ordering $\mathcal{Q}_{l,d.d'}$ in the descending order and going down through the thresholds to assign κ to the smallest $\mathcal{Q}_{l,d.d'}$ such that its false discovery rate does not exceed α.

Once we sweep through $\mathcal{Q}_{l,d.d'}$, we do a backwards pass to remove the nodes with no connections. If the node has no outgoing connections, then all the incoming connections need to be removed, and conversely, if the node has no incoming connections, then all the outgoing connections can be removed. An example of the network resulting after applying the Algorithm 7 is illustrated by the Figure 4.8.

4.3.5 Predictions

For a new $\boldsymbol{x}_*$, the predictive distribution of $\boldsymbol{y}_*$ given the data is approximated as:

$$\begin{aligned} p(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) &= \int p(\boldsymbol{y}_*|\boldsymbol{x}_*, \boldsymbol{\theta}, \mathcal{D}) p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{x}_*)\, d\boldsymbol{\theta} \\ &= \int p(\boldsymbol{y}_*|\boldsymbol{a}_*, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\eta}) p(\boldsymbol{a}_*, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\eta}|\mathcal{D}, \boldsymbol{x}_*)\, d\boldsymbol{a}_*\, d\boldsymbol{W}\, d\boldsymbol{b}\, d\boldsymbol{\eta} \\ &\approx \int p(\boldsymbol{y}_*|\boldsymbol{a}_*, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\eta}) q(\boldsymbol{a}_*) q(\boldsymbol{W}, \boldsymbol{b}) q(\boldsymbol{\eta})\, d\boldsymbol{a}_*\, d\boldsymbol{W}\, d\boldsymbol{b}\, d\boldsymbol{\eta} \\ &= \int \mathcal{N}\left(\boldsymbol{y}_*|\boldsymbol{W}_{L+1}\boldsymbol{a}_{*,L} + \boldsymbol{b}_{L+1}, \boldsymbol{\Sigma}_{L+1}\right) q(\boldsymbol{a}_{*,L}) \\ &\qquad \times q(\boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1}) q(\boldsymbol{\eta}_{L+1})\, d\boldsymbol{a}_{*,L}\, d\boldsymbol{W}_{L+1}\, d\boldsymbol{b}_{L+1}\, d\boldsymbol{\eta}_{L+1}. \end{aligned} \tag{4.21}$$

Equation (4.21) requires first computing the approximate variational predictive distributions $q(\boldsymbol{a}_*), q(\boldsymbol{\gamma}_*)$ and $q(\boldsymbol{\omega}_*)$, which are updated in a similar way to Section 4.3.1.

Algorithm 7 Node selection algorithm

Require: $\mathcal{I} = \{\mathcal{Q}_{l,d.d'} | l = 1 \dots L,\ d = 1 \dots D_{l+1},\ d' = 1 \dots D_l\}$.
  $\hat{\kappa} = \max(\mathcal{I})$
  $\mathcal{I} = \mathcal{I} \setminus \hat{\kappa}$
  if $\widehat{\text{FDR}}(\max(\mathcal{I})) < \alpha$ **then**
    $\hat{\kappa} = \max(\mathcal{I})$
    $\mathcal{I} = \mathcal{I} \setminus \hat{\kappa}$
  else
    break
  end if
  for $l = 1 \dots L,\ d = 1 \dots D_{l+1},\ d' = 1 \dots D_l$ **do**
    if $\mathcal{Q}_{l,d,d'} \geq \hat{\kappa}$ **then**
      $\widehat{W}_{l,d,d'} \sim \mathcal{N}\left(m^W_{l,d,d'}, \left(B^W_{l,d}\right)_{d'd'}\right)$
    else
      $\widehat{W}_{l,d,d'} = 0$ a.s.
    end if
  end for
  for $l = L+1, \dots 2, d = 1, \dots D_l,$ **do**
    if $\widehat{W}_{l,d} = 0$ a.s. **then**
      $\widehat{W}_{l-1,d',d} = 0$ a.s. $\forall\ d' = 1, \dots, D_{l-1}$
    else
      if $\widehat{W}_{l-1,d',d} = 0$ a.s. $\forall\ d' = 1, \dots, D_{l-1}$ **then**
        $\widehat{W}_{l,d} = 0$ a.s.
      end if
    end if
  end for
Ensure: $\widehat{q}(b_{l,d}, \widehat{\boldsymbol{W}}_{l,d}),\ l \in \mathcal{L},\ d \in \mathcal{D}_l,\ d' \in \mathcal{D}_{l-1}$.

Specifically, the stochastic activations are conditionally Gaussian given the previous layer with variational predictive distribution:

$$q(\boldsymbol{a}_*) = \prod_{l=1}^{L} \mathcal{N}\left(\boldsymbol{a}_{*,l} | \boldsymbol{t}_{*,l} + \boldsymbol{M}_{*,l}\boldsymbol{a}_{*,l-1}, \boldsymbol{S}_{*,l}\right),$$

where $\boldsymbol{a}_{*,0} = \boldsymbol{x}_*$. For the final layer, we have:

$$\boldsymbol{S}^{-1}_{*,L} = \hat{\boldsymbol{\Sigma}}^{-1}_L; \quad \boldsymbol{t}_{*,L} = \mathbb{E}\left[\boldsymbol{\gamma}_{*,L}\right] \odot \mathbb{E}\left[\boldsymbol{b}_L\right]; \quad \boldsymbol{M}_{*,L} = \mathbb{E}\left[\boldsymbol{\gamma}_{*,L}\right] \boldsymbol{1}^T_{D_{L-1}} \odot \mathbb{E}\left[\boldsymbol{W}_L\right].$$

For all other layers $l = 1, \dots, L-1$, we have:

$$\begin{aligned}\boldsymbol{S}^{-1}_{*,l} =& \hat{\boldsymbol{\Sigma}}^{-1}_l - \boldsymbol{M}^T_{*,l+1}\boldsymbol{S}^{-1}_{*,l+1}\boldsymbol{M}_{*,l+1} + \\ & \sum_{d=1}^{D_{l+1}} \left(\mathbb{E}\left[\frac{1}{\eta^2_{l+1,d}}\right] \mathbb{E}\left[\gamma_{*,l,d}\right] + \frac{1}{T^2}\mathbb{E}\left[\omega_{*,l+1,d}\right]\right) \mathbb{E}\left[\boldsymbol{W}^T_{l+1,d}\boldsymbol{W}_{l+1,d}\right],\end{aligned}$$

$$
\begin{aligned}
\boldsymbol{t}_{*,l} =& \boldsymbol{S}_{*,l}\Big(\boldsymbol{M}_{*,l+1}^T\boldsymbol{S}_{*,l+1}^{-1}\boldsymbol{t}_{*,l+1} + \hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{*,l}\right]\odot\mathbb{E}\left[\boldsymbol{b}_l\right] \\
&- \sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right]\mathbb{E}\left[\gamma_{*,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T b_{l+1,d}\right] \\
&+ \frac{1}{T}\sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T\right]\left(\mathbb{E}\left[\gamma_{*,l+1,d}\right] - \frac{1}{2}\right) \\
&- \frac{1}{T^2}\sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\omega_{*,l+1,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l+1,d}b_{l+1,d}\right]\Big), \\
\boldsymbol{M}_{*,l} =& \boldsymbol{S}_{*,l}\hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{*,l}\right]\boldsymbol{1}_{D_{l-1}}^T\odot\mathbb{E}\left[\boldsymbol{W}_l\right].
\end{aligned}
$$

The binary activations are independent across layers $l = 1, \ldots, L$ and width $d = 1, \ldots, D_l$, with a Bernoulli variational predictive distribution:

$$
q(\boldsymbol{\gamma}_*) = \prod_l^L\prod_d^{D_l}\text{Bern}\left(\gamma_{*,l,d}|\rho_{*,l,d}\right), \tag{4.22}
$$

with

$$
\begin{aligned}
\rho_{*,l,d} = \sigma\Bigg(&-\frac{\mathbb{E}\left[\eta_{l,d}^{-2}\right]}{2}\text{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}\widetilde{\boldsymbol{a}}_{*,l-1}^T\right]\right) \\
&+\mathbb{E}\left[\eta_{l,d}^{-2}\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}a_{*,l,d}\right] + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}\right]\Bigg).
\end{aligned}
$$

Finally, the Polya-Gamma augmented variables are independent across layers $l = 1, \ldots, L$ and width $d = 1, \ldots, D_l$, with a Polya-Gamma variational predictive distribution:

$$
q(\boldsymbol{\omega}_*) = \prod_l^L\prod_d^{D_l}\text{PG}(\omega_{*,l,d}|1, A_{*,l,d}), \tag{4.23}
$$

with updated variational parameters:

$$
A_{*,l,d} = \frac{1}{T}\sqrt{\left(\text{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}\widetilde{\boldsymbol{a}}_{*,l-1}^T\right]\right)\right)}.
$$

Thus, before computing predictions, we first iterate to update the variational predictive distributions of $\boldsymbol{a}_*$, $\boldsymbol{\gamma}_*$, and $\boldsymbol{\omega}_*$. The corresponding ELBO (derived in the Appendix B.2.2) is monitored for convergence. While a closed-form expression for the integral in Equation (4.21) is unavailable, generating samples from the variational predictive is straightforward;

$$
\boldsymbol{y}_*^{(j)} \sim \mathcal{N}\left(\boldsymbol{y}_*|\boldsymbol{W}_{L+1}^{(j)}\boldsymbol{a}_{*,L}^{(j)} + \boldsymbol{b}_{L+1}^{(j)}, \boldsymbol{\Sigma}_{L+1}^{(j)}\right), \text{ for } j = 1, \ldots J, \tag{4.24}
$$

where $(\boldsymbol{W}_{L+1}^{(j)}, \boldsymbol{b}_{L+1}^{(j)}) \sim q(\boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1})$, $\boldsymbol{\eta}_{L+1}^{(j)} \sim q(\boldsymbol{\eta}_{L+1})$, and $\boldsymbol{a}_{*,l}^{(j)}|\boldsymbol{a}_{*,l-1}^{(j)} \sim$

$q(\boldsymbol{a}_{*,l}|\boldsymbol{a}_{*,l-1})$ for $l = 1, \ldots, L$, are iid draws from the variational posterior. These samples can be used to obtain a Monte Carlo approximation to investigate potential non-normality in the predictive distribution in Equation (4.21) and to compute credible intervals based on the highest posterior density region.

We can also compute the expectation and variance of $\boldsymbol{y}_*$ in closed form. Specifically, the expectation of $\boldsymbol{y}_*$ under the variational predictive distribution is:

$$\mathbb{E}[\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}] \approx \mathbb{E}_{q_{L+1}}[\boldsymbol{W}_{L+1}]\mathbb{E}_{q_{*,L}}[\boldsymbol{a}_{*,L}] + \mathbb{E}_{q_{L+1}}[\boldsymbol{b}_{L+1}], \tag{4.25}$$

where recursively

$$\mathbb{E}_{q_{*,L}}[\boldsymbol{a}_{*,L}] = \mathbb{E}_{q_{*,L-1}}\left[\mathbb{E}_{q(\boldsymbol{a}_{*,L}|\boldsymbol{a}_{*,L-1})}[\boldsymbol{a}_{*,L}]\right] = \boldsymbol{t}_{*,L} + \boldsymbol{M}_{*,L}E_{q_{*,L-1}}[\boldsymbol{a}_{*,L-1}].$$

Similarly, the variational variance of $\boldsymbol{y}_*$ is

$$\mathrm{Var}\left[y_{*,d}|\boldsymbol{x}_*, \mathcal{D}\right] \approx \mathrm{Var}_{q_{L+1}}\left[\boldsymbol{W}_{L+1,d}\boldsymbol{a}_{*,L} + \boldsymbol{b}_{L+1,d}\right] + \mathbb{E}_{q_{L+1}}\left[\boldsymbol{\eta}^2_{L+1,d}\right],$$

where the first term represents the signal variance and is computed as

$$\begin{aligned}
&\mathrm{Var}_{q_{L+1}}\left[\boldsymbol{W}_{L+1,d}\boldsymbol{a}_{*,L} + \boldsymbol{b}_{L+1,d}\right] = \mathbb{E}_{q_{L+1}}\left[\left(\boldsymbol{W}_{L+1,d}\boldsymbol{a}_{*,L} + \boldsymbol{b}_{L+1,d}\right)^2\right] \\
&- \left(\mathbb{E}_{q_{L+1}}\left[\boldsymbol{W}_{L+1,d}\right]\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\right] + \mathbb{E}_{q_{L+1}}\left[\boldsymbol{b}_{L+1,d}\right]\right)^2 \\
&= \mathrm{Tr}\left(\mathbb{E}_{q_{L+1}}\left[\boldsymbol{W}^T_{L+1,d}\boldsymbol{W}_{L+1,d}\right]\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\boldsymbol{a}^T_{*,L}\right]\right. \\
&\left.-\mathbb{E}_{q_{L+1}}\left[\boldsymbol{W}_{L+1,d}\right]^T\mathbb{E}_{q_{L+1}}\left[\boldsymbol{W}_{L+1,d}\right]\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\right]\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\right]^T\right) \\
&+ \mathrm{Var}_{q_{L+1}}(\boldsymbol{b}_{L+1,d}) + 2\mathrm{Cov}_{q_{L+1}}(\boldsymbol{W}_{L+1,d}, \boldsymbol{b}_{L+1,d})\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\right],
\end{aligned}$$

which requires the recursive computation:

$$\begin{aligned}
&\mathbb{E}_{q_L}\left[\boldsymbol{a}_{*,L}\boldsymbol{a}^T_{*,L}\right] = \boldsymbol{S}_{*,L} + \boldsymbol{t}_{*,L}\boldsymbol{t}^T_{*,L} + 2\boldsymbol{M}_{*,L}\mathbb{E}_{q_{L-1}}\left[\boldsymbol{a}_{*,L-1}\right]\boldsymbol{t}^T_{*,L} \\
&+ \boldsymbol{M}_{*,L}\mathbb{E}_{q_{L-1}}\left[\boldsymbol{a}_{*,L-1}\boldsymbol{a}^T_{*,L-1}\right]\boldsymbol{M}^T_{*,L}.
\end{aligned}$$

Sparse prediction. Observe that the variational algorithm used for prediction scales linearly with the number of hidden layers and the number of samples, but cubically with the number of hidden units, which motivates the node-selection method proposed in Section 4.3.4. To save on both computation and storage, the variational predictive distribution can be computed based on the sparse variational posterior (Section 4.3.4). For a new data point $\boldsymbol{x}_*$, we obtain expectation and variance of $\boldsymbol{y}_*$ by first computing the sparse versions of variational predictive distributions $\widehat{q}(\boldsymbol{a}_*), \widehat{q}(\boldsymbol{\gamma}_*)$ and $\widehat{q}(\boldsymbol{\omega}_*)$ as in Equations (4.18), (4.22) and (4.23) by plugging $\widehat{q}(b_{l,d}, \widehat{\boldsymbol{W}}_{l,d})$ instead of the $q(b_{l,d}, \boldsymbol{W}_{l,d})$, which only requires updates for the subset of nodes with nonzero weights.

4.3.6 Ensembles of variational approximations

While the variational algorithm described in Section 4.3.1 increases the ELBO at each epoch, the ELBO is a non-convex function of the variational parameters and only convergence to a local optimum is guaranteed. Due to identifiability issues,

the posterior distribution of a Bayesian neural network is highly multimodal, and exploring this posterior is notoriously challenging [Papamarkou et al., 2022]. A single variational approximation tends to concentrate around one mode and can understate posterior uncertainty. Several approaches have been proposed to overcome such issues. Recently, [Ohn and Lin, 2024] introduced adaptive variational inference which achieves optimal posterior contraction rate and model selection consistency by considering several variational approximations obtained in different models; the framework operates on a collection of models, considers "sieve" priors [Arbel et al., 2013] to combine several variational approximations. Similarly, [Yao et al., 2022] introduced an approach which uses parallel runs of inference algorithms to cover as many modes of the posterior distribution as possible and then combines these using Bayesian stacking. Observing non-optimality of conventional non-Bayesian deep ensembles [Lakshminarayanan et al., 2017] combining point estimates [Wu and Williamson, 2024], in our approach, we adopt the ideas of [Ohn and Lin, 2024, Yao et al., 2022] and consider ensembles of posterior approximations in a similar but simpler fashion (for a discussion of neural network ensembles, we refer to Section 1.2.5). Specifically, we consider an ensemble of variational approximations, obtained by running in parallel the variational algorithm multiple times with different random starting points and combining those with respect to the optimization objective. In this case, letting $k = 1, \ldots, K$ index the different variational approximations, we compute the weight w_k associated with each approximation in accordance with the tempered ELBO:

$$w_k \propto \exp\left(\zeta \mathrm{ELBO}_k\right),$$

where ζ is a tempering parameter, setting which to be in the interval $(0, 0.1]$ allows for avoiding a strong dominance of a single particular model. We can interpret this as a Bayesian model averaging (BMA) across the K different models/approximations. While in a classical BMA setting, the weights would be proportional to the marginal likelihood for each model, the use of the ELBO is motivated as it provides a lower bound to the marginal likelihood and can be computed in closed form. Next, we compute predictions by taking a weighted average of the predictive distributions of each model (given in Equation (4.25)), that is

$$\mathbb{E}[\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}] \approx \sum_{k=1}^{K} w_k \mathbb{E}_{q_k}[\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}],$$

where each expectation is taken with respect to q_k (the kth variational approximation). Similarly, we can compute the variance as

$$\begin{aligned}\mathrm{Var}(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) \approx \sum_{k=1}^{K} w_k \,\mathrm{Var}_{q_k}(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) + \sum_{k=1}^{K} w_k \left(\mathbb{E}_{q_k}[\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}]\right)^2 \\ - \left(\sum_{k=1}^{K} w_k \mathbb{E}_{q_k}[\boldsymbol{y}_*|\boldsymbol{x}_*.\mathcal{D}]\right)^2.\end{aligned}$$

The following approach can potentially improve both predictive accuracy and uncertainty quantification. Once again, we can investigate the variational predictive distribution (beyond the mean and variance) by first sampling a model with probability $(w_1, \ldots, w_K)$ and then given that selected model k, generating a sample $\boldsymbol{y}_*$ from the kth variational predictive distribution (as described in Equation (4.24)).

4.4 Experiments

We evaluate the variational bow tie neural network (VBNN) on several datasets. First, we consider a simple nonlinear synthetic example to compare with the ground truth. We then validate VBNN on the diabetes dataset, first considered in [Efron et al., 2004] to demonstrate the least angle regression (LARS) algorithm for variable selection, and subsequently, used in different proposals for sparsity-promoting priors algorithms (e.g. [Li and Lin, 2010, Park and Casella, 2008]). Lastly, we consider a range of popular regression datasets from the UCI Machine Learning Repository [M. et al., 2007].

The importance of suitable initialization choice in NNs is well known [Daniely et al., 2016, He et al., 2015, Wenzel et al., 2020a], and we design two possible random initialization schemes of the VBNN, which are described in Appendix B.4.1 and used in all experiments. Convergence of the ELBO is monitored during the training and prediction stages, where if three consecutive measurements of ELBO for training differ by less than the specified threshold, the phase is stopped and the model moves to the prediction stage, where we proceed similarly. In most experiments, the thresholds during the training and prediction stages are set, respectively, to $1e-5$ and $1e-4$.

We compare the performance of VBNN to various frameworks (summarized in Table 4.2), namely, to BNNs inferred with automatic differentiation VI with the mean-field variational family (mfVI) [Kucukelbir et al., 2017] and Hamiltonian Monte Carlo (HMC) with the No-U-Turn sampler (NUTs) [Hoffman and Gelman, 2014] implemented in Numpyro [Phan et al., 2019], Bayes by Backprop (BBB) [Blundell et al., 2015] implemented with Pytorch, and BNNs with Horseshoe priors (HSBNN) of [Ghosh et al., 2019], which considers a structured variational family and learns the variational parameters by obtaining gradient estimators. We also consider the variational bow tie neural network with Gaussian priors (GVBNN), in contrast to shrinkage priors. As CAVI may be expensive for large data, in Section 4.4.1 and Section 4.4.3 we additionally consider VBNN inferred with SVI (SVBNN).

For all the datasets, we evaluate the performance over 10 random splits, where we use 90% of the data for the training and 10% for testing the model. We record the root mean squared error (RMSE), the predictive negative log-likelihood of the test data (NLL) and the empirical coverage (EC) (see Appendix B.4 for additional implementation details). The empirical coverage reports the fraction of observations contained within the $(1-\alpha)*100\%$ credible intervals (CIs), which are computed based on the Gaussian approximation. In the ideal setting, the computed EC should equal the CI level. More specifically, let $\boldsymbol{y}_i^*$ be the true target for test points $i = 1, \ldots, N^*$ and $q_{\alpha/2}$ and $q_{1-\alpha/2}$ be the quantiles based on

Table 4.2: List of the models considered to evaluate the performance of our method.

Model	Description
mfVI	BNN with Automatic Differentiation VI with mean-field family
HMC	BNN with Hamiltonian Monte Carlo with No-U-Turn sampler
BBB	Bayes by Backprob
HSBNN	BNN with Horseshoe priors inferred with Black Box VI
VBNN	Our model inferred with CAVI
GVBNN	VBNN with Gaussian priors inferred with CAVI
SVBNN	Our model inferred with SVI

the model's Gaussian approximation for some $\alpha \in [0.5, 1]$, then

$$\text{EC}(\alpha) = \frac{1}{N^*} \sum_{i=1}^{N^*} \mathbf{1}(\boldsymbol{y}_i^* \in [q_{\alpha/2}, q_{1-\alpha/2}]).$$

If $\text{EC}(\alpha) > 1-\alpha$ then the CIs are too wide; a worse scenario occurs when $\text{EC}(\alpha) < 1-\alpha$ as it means that the CIs are too narrow and the model is overconfident in its predictions.

4.4.1 Simulated Example

We construct a synthetic dataset generated by first uniformly sampling a two-dimensional input vector $\boldsymbol{x}_n = (x_{n,1}, x_{n,2})$, with $x_{n,d} \sim \text{Unif}([-2, 2])$, and assume only the first feature influences the output: $y_n = f(x_{n,1}) + \epsilon_n = 0.1x_{n,1}^2 + 10\sin(x_{n,1}) + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, 0.5)$. Then, the dataset consisting of $N = 300$ observations is used to investigate the performance of VBNN compared to the GVBNN, mfVI, HMC, BBB and HSBNN baselines as we increase the number of hidden layers, setting $L = 1, 2$ or 4, whilst keeping the number of hidden units in each layer fixed to $D_H = 20$. In general, for this simple non-linear example, the performance tends to deteriorate with increasing architecture complexity (larger depth). While HMC performs consistently well across all depths, the cost associated with the sampling approach is high. VBNN is competitive to HSBNN and outperforms mfVI, GVBNN and BBB in terms of accuracy (see Figure 4.6). Further, except for HMC, the empirical coverage of VBNN is the most robust to the choice of depth; for the largest choice of $L = 4$, mfVI, BBB, GVBNN and even HSBNN provide overly wide CIs while VBNN more closely reaches the desired coverage (see Figure 4.7).

For each depth, Figure 4.8 illustrates the predictive means and uncertainties computed for the observations as well as DAGs of networks' structures obtained after the post-process node selection algorithm described in Section 4.3.4, where the Bayesian false discovery rate is constrained by setting the error rate to $\alpha = 0.01$. The sparsity-promoting prior combined with the node selection algorithm can effectively prune the over-parametrized neural networks; for example, the sparse one-layer neural network contains only 11 hidden nodes with

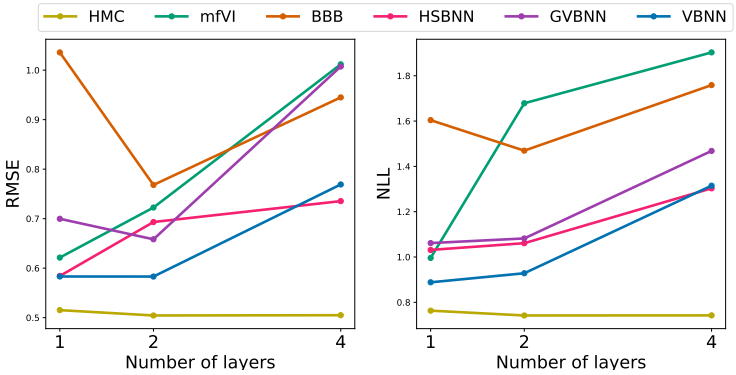


Figure 4.6: Simulated example. Performance in terms of the RMSE and NLL as the depth increases for different models and algorithms. HMC can be seen as a gold standard. VBNN is competitive with HSBNN and is more robust to the choice of depth and overparameterization than GVBNN, mfVI, BBB.

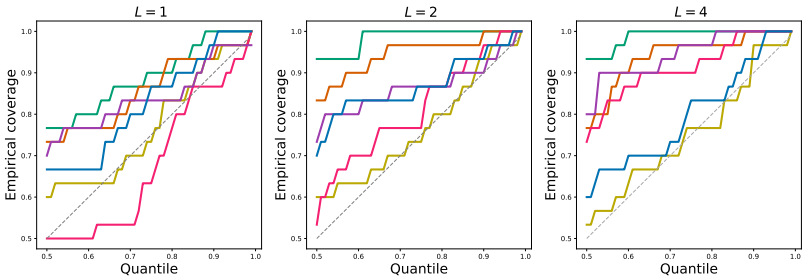
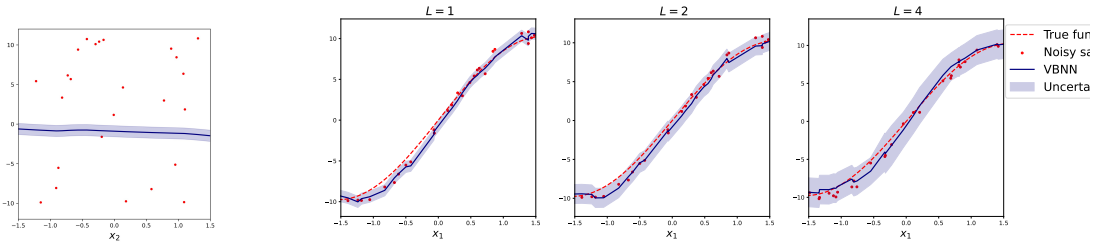


Figure 4.7: Simulated example. Empirical coverage (which is the fraction of observations contained within the CIs of level $1 - \alpha$) as a function of CI level for the simulated dataset for three different settings of the network's depth. The dashed gray line depicts the ideal scenario with empirical coverage equal to CI level, while above and below the gray line indicate coverage greater or less than CI level, i.e. CIs are too wide or too small, respectively.

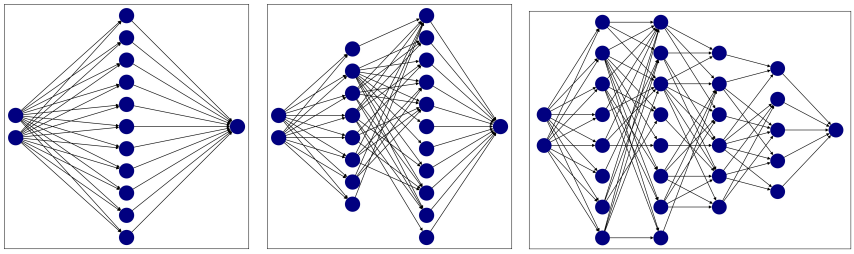
33 total edges/weights from the initial $D_H = 20$ with 60 total edges/weights. Moreover, the estimated regression function and credible intervals from both the variational predictive and the sparse variational predictive distribution recover the true function well. In this way, VBNN provides an effective tool to reduce predictive computational complexity and storage as well as ease interpretation. Note that the predictions show no relation with the coordinate x_2 (Figure 4.8a), recovering the true function, but some of the connections from x_2 are still present in the sparse network (Figure 4.8c), due to identifiability issues, although with overall low weight.

Further, we evaluate the role of step sizes (defined in Equation (4.20)) and mini-batch sizes and consider the performance of the SVBNN and ensembles of 5 SVBNN approximations compared to VBNN in a single-layer network. Figure 4.9a compares the results for various step sizes (defined in Equation (4.20)) and mini-batch sizes, and Figure 4.9b illustrates the predictions obtained by VBNN, SVBNN, and ensembles of SVBNN. The computational gains obtained



(a) The predictive mean as a function of the second coordinate for $L = 1$.

(b) The predictive mean and uncertainty quantification for the observations for different depth



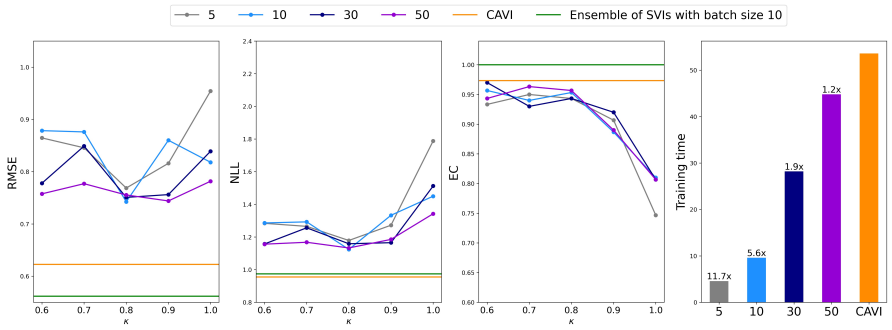
(c) The architecture of the network for the bound on the FDR $\alpha = 0.01$ for depths $L = 1, 2, 4$ (left to right).

Figure 4.8: Simulated example. Predictive means and pointwise CIs computed for the observations as a function of the second coordinate (a) and first coordinate for different depths (b). The architecture of the network is visualized in (c) for the bound on the FDR $\alpha = 0.01$ for different settings of the network's depth of $L = 1, 2, 4$ (left to right).

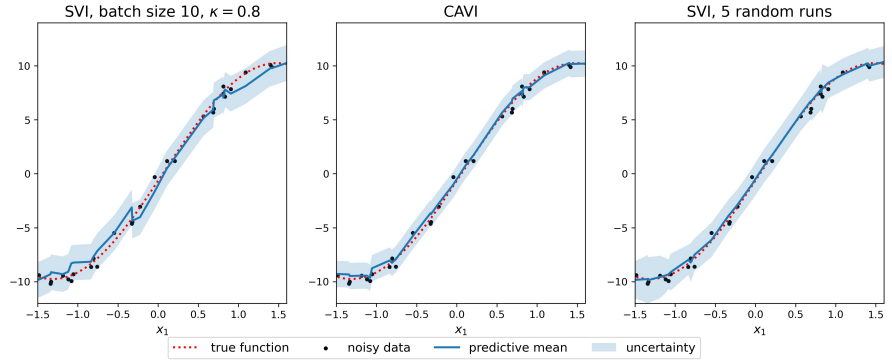
by utilising stochastic gradients and subsampling would motivate further research along the lines of model combination. Indeed, SVI makes ensembling techniques particularly appealing, in our experiment with a relatively small synthetic dataset, an ensemble of 5 SVI approximations with a mini-batch size of 10 performs comparable to CAVI while being more than 5 times faster than CAVI (see Figure 4.9a).

4.4.2 Diabetes Example

The diabetes data consists of $n = 442$ entries obtained for $p = 10$ input variables and a quantitative response measuring disease progression. The predictors are age, sex, body mass index, average blood pressure and six blood serum measurements, and the goal is to determine which of these are relevant for forecasting diabetes progression. We fit a neural network with one hidden layer $L = 1$ and $D_H = 20$ and perform the node selection algorithm with the FDR bounded by $\alpha = 0.01$. Figure 4.10 illustrates the shrinkage and node selection algorithm and compares the coefficients of the Lasso linear model [Tibshirani, 1996] to the original and the sparsified weights of our model. Lasso regression produces sparse coefficients by minimising the residual sum of squares with an added penalty term; the penalty parameter crucially determines the level of sparsity and is tuned with



(a) RMSE, NLL, EC plotted for various mini-batch sizes as a function of the forgetting rate k; the most right plot compares training times, where bar labels indicate the scale of computational gains.



(b) The predictive means and uncertainty estimate as a function of the first coordinate for SVBNN, VBNN and ensembles of SVBNN.

Figure 4.9: Comparison between VBNN (CAVI) and SVBNN (SVI) for the simulated data example.

cross-validation (LassoCV). Predictors with considerable effect obtained by both models coincide, whilst some of the variables the Lasso model excludes (e.g. age) are still present in the VBNN's estimates. Compared with Lasso, VBNN has the advantage of learning potential nonlinear relationships between disease progression and the predictors, which is explored in Figure 4.11, illustrating the predictive means and uncertainty of the observations of VBNN for four of the predictors (with all other predictors are fixed to their mean). While the uncertainty is wide, the results suggest potential nonlinear relationships, e.g. with lamotrigine and age, the latter of which is not selected in Lasso. Moreover, Figure 4.11 highlights how predictions obtained from the sparse version of the variational predictive distribution almost overlap, thus providing a reasonable, cheaper approximation. However, we note that the predictive performance is similar to LassoCV, with the most competitive methods being VBNN, HMC and BBB (see Table 4.3 and supplementary Figure B.1 in Appendix B.4.3).

4.4.3 UCI Regression Datasets

Lastly, we consider publicly available datasets from the UCI Machine Learning Repository [M. et al., 2007]: Boston housing [Harrison and Rubinfeld, 1978], En-

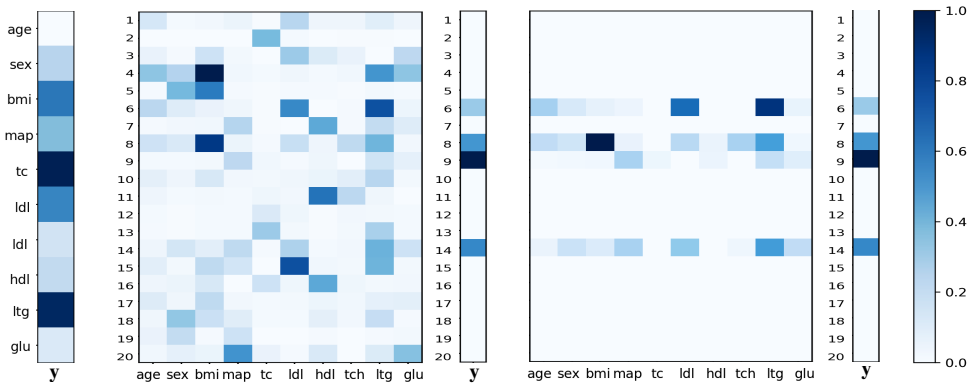


Figure 4.10: Diabetes example. Coefficients of LassoCV regression (on the left), posterior means of the weights of the neural network (in the middle) and posterior means of the sparse weights obtained for $\alpha = 0.01$ (on the right). For illustrative purposes, absolute values of the coefficients and weights are shown with max-min scaling.

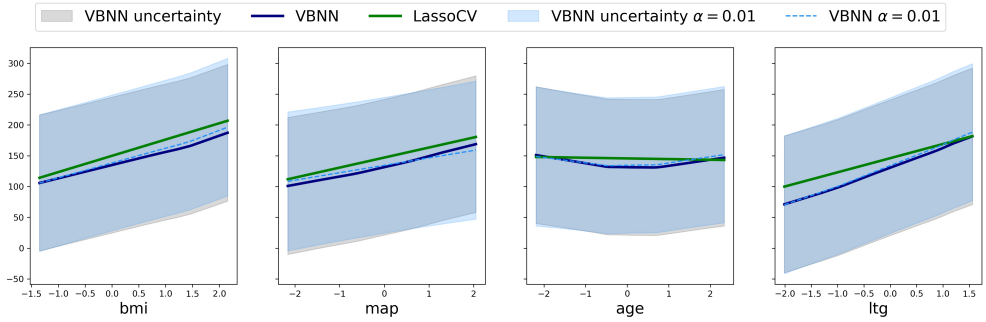


Figure 4.11: Diabetes example. Slices of the predictive mean and pointwise credible intervals for observations as a function of four predictors obtained by VBNN with and without node selection and by Lasso with cross-validation.

ergy [Tsanas and Xifara, 2012], Yacht dynamics [J. et al., 2013], Concrete compressive strength [Yeh, 2007] and Concrete slump test [Yeh, 2009] (see Appendix B.4.4 for the description of the datasets). For all of the UCI regression tasks, we fit a neural network with one hidden layer and $D_H = 50$ hidden units. Figure 4.12 compares RMSE, NLL and empirical coverage of the observations across the methods (see also Table B.1 in Appendix B.4.5), where we additionally consider ensembles of four variational approximations with VBNN and SVBNN (denoted as 4VBNN and 4SVBNN, respectively). Overall, HMC outperforms all the considered methods, but at a much higher cost. VBNN provides an improvement compared to GVBNN, further motivating the choice of sparsity-inducing priors, and while SVBNN offers considerable computational savings, the quality of the approximation deteriorates compared to VBNN. Overall, VBNN has consistently well-calibrated uncertainty quantification and empirical coverage for the observations compared with other variational approaches, and ensembles of VBNNs are competitive with other approaches in terms of the RMSE and NLL. Further, we consider slump dataset and implement ensembles of 4 parallel runs of all of

Table 4.3: RMSE, NLL and empirical coverage for diabetes dataset.

	RMSE	NLL	Coverage
LassoCV	54.2 ± 6.5	$5.4 \pm .13$	$.96 \pm .03$
mfVI	57.2 ± 7.4	9.3 ± 1.7	$.47 \pm .1$
HMC	54.5 ± 7.8	$5.4 \pm .16$	$.96 \pm .04$
BBB	54.9 ± 7.3	$5.49 \pm .18$	$.94 \pm .04$
HSBNN	56.8 ± 7.4	6.8 ± 1.6	$.67 \pm .15$
GVBNN	55.65 ± 7.8	$5.5 \pm .14$	$.96 \pm .04$
VBNN	54.5 ± 7.2	$5.4 \pm .15$	$.96 \pm .04$

the considered methods (4mfVI, 4BBB, 4HSBNN, 4GVBNN, 4VBNN, 4SVBNN) and compare the results to approximations obtained in a single run (mfVI, BBB, HSBNN, GVBNN, VBNN, SVBNN). We do not consider HMC in this experiment due to its computational costs. Figure 4.13 compares RMSE, NLL, EC of the 12 methods and additionally illustrates relative differences among approaches, where for RMSE and NLL, we consider absolute relative differences between ensembles and single runs, and for empirical coverage of the observations, we illustrate the absolute deviation from the 95% CI. Overall, ensembles improve uncertainty quantification, and in most cases also RMSE and NLL (the only exception being NLL for BBB). While BBB and HSBNN have the lowest RMSE (although with high variability), the NLL and empirical coverage suggest overconfidence, even with ensembles. In contrast, VBNN has an improved balance between accuracy and uncertainty quantification, which is further enhanced by ensembles.

4.5 Discussion

In this chapter, we presented a variational bow tie neural network (VBNN) that is amendable to Polya-gamma data augmentation so that the variational inference can be performed via the CAVI algorithm. While the idea of the stochastic relaxation described in Section 4.2.1 was introduced in [Smith et al., 2021], the novelty of our model is in the employment of the variational inference techniques as well as sparsity-inducing priors. Namely, we implement continuous global-local shrinkage priors and propose a post-process technique for node selection. Additionally, we consider an improvement of the classical CAVI algorithm by adding EM steps for critical hyperparameters. In this way, we enrich the class of models which are handled within the structured mean-field paradigm. We provide all the necessary computations, techniques, and illustrative experiments demonstrating the utility of the model. Addressing the scalability with respect to the number of data points, we extend the CAVI algorithm to SVI [Hoffman et al., 2013], which benefits from exploiting natural gradients and subsampling. In the future, we could improve the algorithm by employing an adaptive learning rate which is based on realisations of a noisy estimate of the natural gradient of ELBO with respect to global variational parameters and moving averages [Ranganath et al., 2013, Schaul et al., 2013]. Alternatively, instead of changing the learning rate, one could adapt the mini-batch size based on the estimated gradient noise covariance

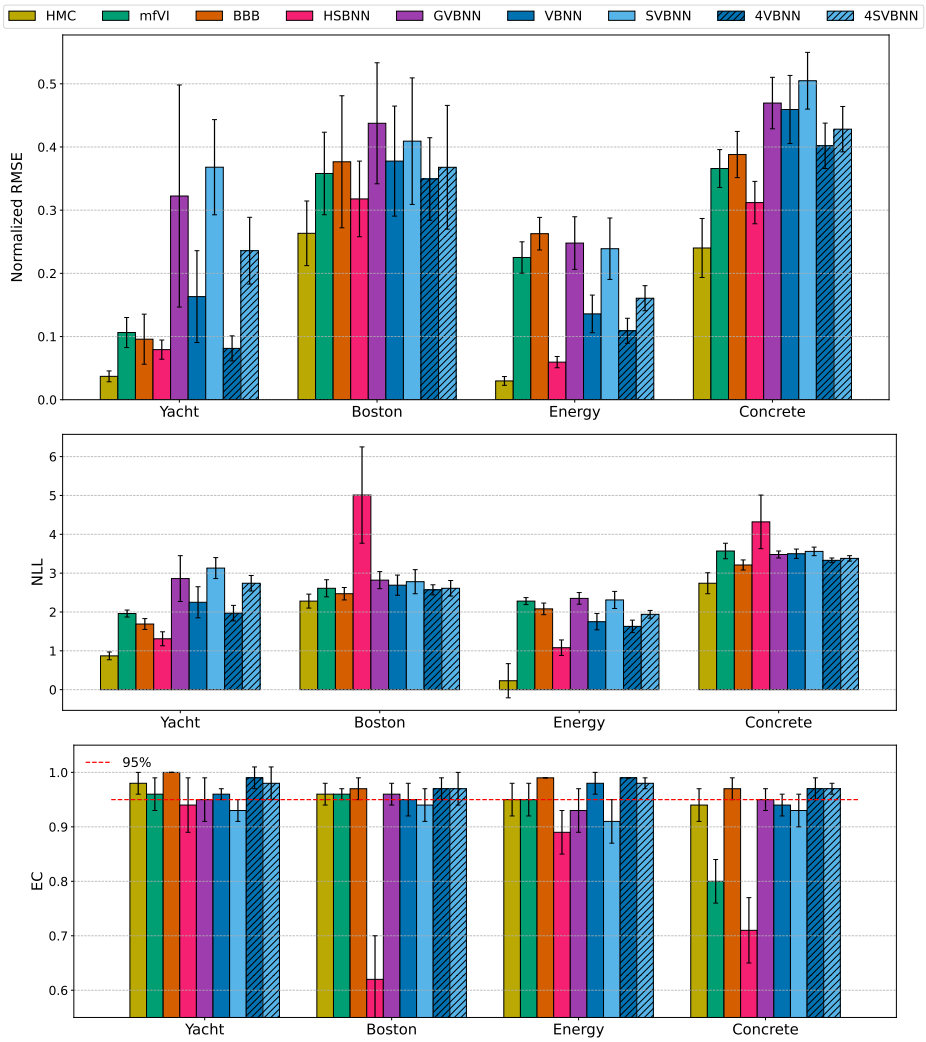


Figure 4.12: RMSE (normalized w.r.t. to the standard deviation of the target), NLL and empirical coverage for UCI datasets. When illustrating the coverage, the dashed red line depicts the ideal scenario with empirical coverage equal to 95% CI level.

and the magnitude of the gradient [Balles et al., 2017]. To address scalability with respect to the network's width, future work will explore incorporating node selection within the CAVI algorithm when training.

The variational bow tie neural network is also amenable to other prior choices and output types (for a more technical discussion of future directions, we refer to Appendix B.5.2). For example, horseshoe priors can be implemented through the introduction of auxiliary variables to replace each half-Cauchy random variable with the hierarchical formulation based on Inverse-Gamma variables [Ghosh et al., 2018, Louizos et al., 2017, Wand et al., 2011]. Additionally, the regularized version of horseshoe priors ("ponyshoe") could be considered, which is known to perform better than the classical horseshoe, especially when the larger coefficients are weakly identified by the data [Ghosh et al., 2019, Piironen and Vehtari, 2017a,b]. Finally, an extension to other output types, such as classification tasks, can be

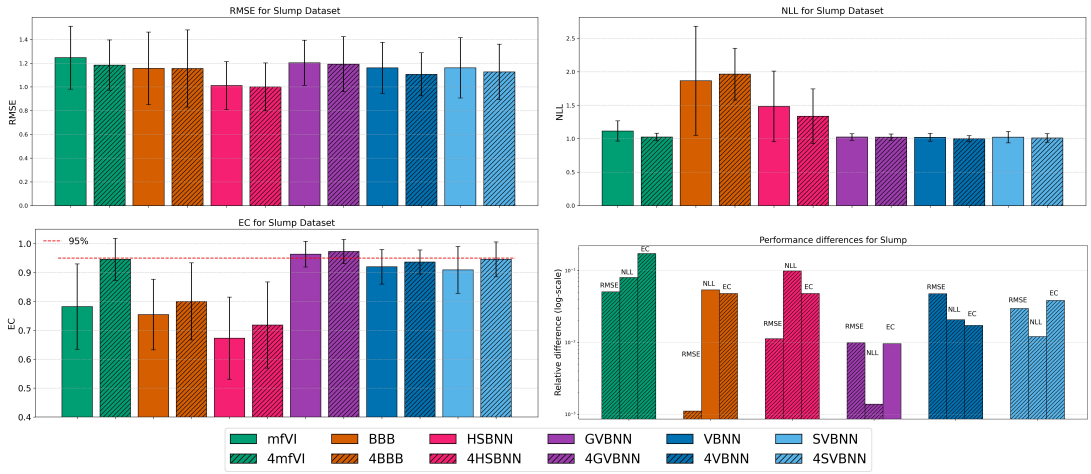


Figure 4.13: Slump dataset. Performance in terms of the RMSE, NLL and EC for single models (plain colored) and ensembles (color with hatches) obtained from four parallel runs. RMSE and NLL are scaled with respect to the best model (top row). The relative performance (bottom right) is illustrated on the log-scale, and color reflects if ensembles improved the metric (i.e. bar with hatches illustrates the scale of improvements obtained with ensembles, conversely, bar without the hatches illustrates the scale at which single run outperformed ensembles).

developed through additional Polya-gamma augmentation techniques [Durante and Rigon, 2019].

Chapter 5

Discussion

We conclude by first summarizing the contributions of the thesis and then, overviewing the areas which are closely related to the themes considered in this work, but were left out of scope due to time constraints.

5.1 Contributions

This thesis contributes to the development of the deep Bayesian modelling framework from several perspectives:

- Chapter 2 addresses the challenges of approximate Bayesian inference by systematically studying different angles of variational inference. We propose the taxonomy of VI methods that sheds some light on the future research directions for optimization-based approximate inference. By improving our understanding of approximate inference from different perspectives, we can bridge the gap between the true and approximate posterior. Specifically, the trade-offs between the complexity variational family, computational efficiency and quality of the approximate posterior predictive distribution are something to keep in mind. Further, the properties of the probability models most suitable for variational inference are worth exploring; and finally, various divergence functions and optimization algorithms could be considered.

- No inference algorithm exists without a model. In Chapter 3, we considered the architectural choices made in Bayesian neural networks and investigated the empirical performance of variational inference and sampling-based methods depending on the model specification and task at hand. We have not lost sight of the Bayesian predictive methods for model comparison and combination that are well-suited for real-world scenarios, when neither the true data-generating process nor the future upcoming data are known. In our experiments, variational inference overall provided better uncertainty quantification than Markov chain Monte Carlo, and while MCMC often had better accuracy, the time needed to perform it becomes a burden as networks get wider and/or deeper. Furthermore, when dealing with multimodal posteriors and mitigating the risks of choosing the mean-field

variational family, we found stacking and ensembles of variational approximations to be a successful and highly efficient alternative to MCMC.

- By observing the vulnerabilities and challenges arising when modelling and implementing Bayesian neural networks, in Chapter 4 we arrive at a novel variational bow tie neural network trained by (possibly stochastic) variational inference algorithm, which removes restrictive independence and distributional assumptions of standard VI algorithms. The VBNN is tailored not only for achieving computational efficiency but also solves the task of model choice and calibration by employing sparsity-inducing priors.

5.2 Future directions and open problems

This thesis adopted a classical Bayesian approach, starting point of which lies in finding a suitable model, e.g. a neural network, that assumes a prior and a likelihood, then the Bayes rule gives the exact posterior over some parameters of interest, and one aims to approximate it [Gelman et al., 2013, 2017]. We have observed multiple challenges arising when specifying a high-dimensional model whose parameters do not have a direct interpretation; as a result, often choices are made out of practical considerations. Bayesian neural networks are typically applied across multiple datasets, and it is, thus, not always reasonable to assume that the chosen model is well-specified and fully recovers the true data-generating process. At the same time, if in practice the choice of inference method cannot be disentangled from the model's architecture, it would be sensible to approach the problem by creating inference methods that are robust to model misspecification.

We focused on variational inference and adopted optimization-based view on approximate Bayesian inference in its classical prior-likelihood formulation. Note, as well as the approximate posteriors, the exact Bayesian posteriors can be characterised as solutions of certain optimization tasks [Walker, 2006, Zellner, 1988]. Furthermore, this perspective can be extended to generalized (also known as quasi- or pseudo-) posteriors which solve optimization task defined by a divergence, a variational family, and a loss function [Bissiri et al., 2016, Knoblauch et al., 2022]. Such framework leads to the collection of methods which aim to generalize both exact and approximate Bayesian inference. Variational approximations considered in this thesis can be recovered by choosing the reverse Kullback-Leibler divergence and the negative log-likelihood as the loss function. If, in addition, the variational family coincides with the space of all probability measures on the parameter space, the generalized posterior coincides with the classical posterior of Bayes rule [Zellner, 1988]. If the likelihood is raised to a power, the resulting posterior is known as tempered, fractional or power posterior [Alquier and Ridgway, 2020]. To relax restrictive assumptions of the traditional prior-likelihood approach, one may connect the information (e.g. from the data) to the parameter of interest by choosing loss function on the space of probability measures on the parameter space. This leads to prominent class of generalized posteriors known as Gibbs or PAC-Bayes posteriors [Bissiri et al., 2016, Jewson et al., 2018]. For instance, choosing the Maximum Mean Discrepancy func-

tion as the loss results in posteriors which are consistent and robust to certain model misspecification; and moreover, approximations of such Gibbs posteriors obtained with variational inference with reverse KL-divergence are able to retain the same properties [Alquier et al., 2016, Cherief-Abdellatif and Alquier, 2020]. A promising direction of future research is thus to investigate the properties and generalization capabilities of approximate and exact posteriors in Bayesian neural networks derived using robust loss functions and divergences beyond the reverse KL-divergence.

Furthermore, when defining a model through prior and likelihood is challenging, one can revisit the concept of the Bayesian predictive rule [Fortini and Petrone, 2025]. In this thesis, we observed the power of the predictive view on model assessment; in fact, the predictive Bayesian methods are not limited to model comparison criteria and are rooted in the very decision-theoretic foundations of the Bayesian framework [Bernardo et al., 1994, De Finetti, 1937, Savage, 1972]. Rather than aiming to describe the data data-generating process itself, the predictive approach reasons through conditional probabilities to obtain probabilities of future events given the observed data. Indeed, if uncertainty in a parameter of interest arises due to missing observations, then posteriors can be modelled and quantified through predictions [Fong et al., 2023, Fortini and Petrone, 2023]. In the context of Bayesian deep learning, a promising direction for future work could be along the lines of martingale posteriors (MPs) [Fong et al., 2023]. For example, deep ensembles combining point estimates were recently interpreted as a form of a misspecified MP, while a well-posed MP formulation achieved strong empirical performance in Bayesian neural networks [Wu and A Williamson, 2024]. While, to the best of our knowledge, the research on predictive variational inference remains limited, it would be interesting to extend the predictive VI of Lai and Yao [2024], particularly in the context of BNNs.

The behaviour, and often the success, of modern neural networks is frequently regarded as mysterious, if not opaque. We argue that many of the conceptual puzzles of deep learning can be addressed and explained by adopting the power of Bayesian inference. While defining, computing and approximating distributions that arise in Bayesian neural networks requires certain effort, and there is clearly room for improvement, real-world events cannot be captured by the point estimates alone, and the benefits of dealing with probability distributions are fundamental and should not be underestimated. Therefore, careful and methodological development of Bayesian inference frameworks that are be scalable and robust in the realms of modern neural networks, is essential for understanding, reasoning about, and making reliable decisions in real-world scenarios.

Bibliography

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 6

Abhinav Agrawal and Justin Domke. Amortized variational inference for simple hierarchical models. *Advances in Neural Information Processing Systems*, 34: 21388–21399, 2021. 33, 34

Devanshu Agrawal, Theodore Papamarkou, and Jacob Hinkle. Wide neural networks with bottlenecks are deep Gaussian processes. *Journal of Machine Learning Research*, 21(175), 2020. 9, 61

Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer, 1998. 48

Pierre Alquier and James Ridgway. Concentration of tempered posteriors and of their variational approximations. *The Annals of Statistics*, 48(3):1475–1497, 2020. 31, 89

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 17(236):1–41, 2016. 90

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in Neural Information Processing Systems*, page 7786–7795, 2018. 6

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. 22

Elaine Angelino, Matthew James Johnson, Ryan P Adams, et al. Patterns of scalable Bayesian inference. *Foundations and Trends® in Machine Learning*, 9(2-3):119–247, 2016. 11, 30

Julyan Arbel, Chislaine Gayraud, and Judith Rousseau. Bayesian optimal adaptive estimation using a sieve prior. *Scandinavian Journal of Statistics*, 40(3): 549–570, 2013. 78

Julyan Arbel, Konstantinos Pitas, Mariia Vladimirova, and Vincent Fortuin. A primer on Bayesian neural networks: review and debates. *arXiv preprint arXiv:2309.16314*, 2023. 3, 12, 40

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2020. 6

Jincheng Bai, Qifan Song, and Guang Cheng. Adaptive variational Bayesian inference for sparse deep neural network. *arXiv preprint arXiv:1910.04355*, 2019. 35

Jincheng Bai, Qifan Song, and Guang Cheng. Efficient variational inference for sparse deep learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 33:466–476, 2020. 35

Lukas Balles, Javier Romero, and Philipp Hennig. Coupling adaptive batch sizes with learning rates. In *Conference on Uncertainty in Artificial Intelligence*, pages 675–684. Curran Associates, Inc., 2017. 72, 86

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *Transactions on Pattern Analysis and Machine Intelligence*, 41(02):423–443, 2019. 40, 45

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012. 1, 10, 24, 153

D. Barber and Christopher Bishop. Ensemble learning in Bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pages 215–237. Springer Verlag, January 1998. 7, 34

David Barber and Wim Wiegerinck. Tractable variational structures for approximating graphical models. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, 1998. 23

Thomas Bayes. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, 53: 370–418, 1763. 7

José M Bernardo, Adrian FM Smith, and Mark Berliner. *Bayesian theory*, volume 586. Wiley Online Library, 1994. 90

MJ Betancourt, Simon Byrne, and Mark Girolami. Optimizing the integrator step size for Hamiltonian Monte Carlo. *arXiv preprint arXiv:1411.6669*, 2014. 45

Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, and Brandon Willard. Lasso meets horseshoe. *Statistical Science*, 34(3):405–427, 2019. 60

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. 21, 57

Christopher M. Bishop, Neil D. Lawrence, and Michael I. Jordan. Mixture representations for inference and learning in boltzmann machines. In *Conference on Uncertainty in Artificial Intelligence*, page 320–327. Morgan Kaufmann Publishers Inc., 1998. 7, 23, 32, 34

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):1103–1130, 2016. 10, 89

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012. 20, 30

David M. Blei. Variational inference: Foundations and innovations, 2019. URL `https://www.cs.columbia.edu/~blei/talks/Blei_VI_tutorial.pdf`. xiii, 38

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, 2017. 19, 21

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015. 12, 35, 44, 79, 123, 146

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, pages 10–21. Association for Computational Linguistics (ACL), 2016. 154

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. Jax: composable transformations of python+numpy programs, 2018. URL `http://github.com/jax-ml/jax`. 42

Anderson F Brito, Elizaveta Semenova, Gytis Dudas, Gabriel W Hassler, Chaney C Kalinich, Moritz UG Kraemer, Joses Ho, Houriiyah Tegally, George Githinji, Charles N Agoti, et al. Global disparities in sars-cov-2 genomic surveillance. *Nature Communications*, 13(1):1–13, 2022. 1

Tamara Broderick. Variational Bayes and beyond: Bayesian inference for big data, 2020. URL `https://tamarabroderick.com/tutorial_2020_smiles.html`. xiii, 38

Tamara Broderick, Andrew Gelman, Rachael Meager, Anna L Smith, and Tian Zheng. Toward a taxonomy of trust for probabilistic machine learning. *Science advances*, 9(7), 2023. 1

James Brofos, Marylou Gabrié, Marcus A Brubaker, and Roy R Lederman. Adaptation of the independent metropolis-hastings sampler with normalizing flow

proposals. In *International Conference on Artificial Intelligence and Statistics*, pages 5949–5986. PMLR, 2022. 10

Lawrence D. Brown. Fundamentals of statistical exponential families with applications in statistical decision theory. *Lecture Notes-Monograph Series*, 9, 1986. 20

Wray L. Buntine and Andreas S. Weigend. Bayesian backpropagation. *Complex systems*, 5:603–643, 1991. 7

Javier Burroni, Justin Domke, and Daniel Sheldon. Sample average approximation for black-box variational inference. In *Conference on Uncertainty in Artificial Intelligence*, volume 244, pages 471–498. PMLR, 2024. 27

David R Burt, Sebastian W Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021. 35

Peter Carbonetto and Matthew Stephens. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis*, 7(1):73–108, 2012. 30

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 6

François Caron and Arnaud Doucet. Sparse Bayesian nonparametric regression. In *International Conference on Machine Learning*, page 88–95. Association for Computing Machinery, 2008. 60, 152

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76, 2017. 28

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 73–80, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009. PMLR. 35, 153

Ismaël Castillo, Johannes Schmidt-Hieber, and Aad Van der Vaart. Bayesian linear regression with sparse priors. *The Annals of Statistics*, pages 1986–2018, 2015. 56

Ismaël Castillo and Paul Egels. Posterior and variational inference for deep neural networks with heavy-tailed weights. *arXiv preprint arXiv:2406.03369*, 2024. 31, 35, 59

Alain Celisse, Jean-Jacques Daudin, and Laurent Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899, 2012. 31

Neil K Chada, Ajay Jasra, Kody JH Law, and Sumeetpal S Singh. Multilevel Bayesian deep neural networks. *Computing Research Repository*, 2022. 61

Oscar Chang, Yuling Yao, David Williams-King, and Hod Lipson. Ensemble model patching: A parameter-efficient variational Bayesian neural network. *arXiv preprint arXiv:1905.09453*, 2019. 52

Vaggos Chatziafratis, Sai Ganesh Nagarajan, and Ioannis Panageas. Better depth-width trade-offs for neural networks through the lens of dynamical systems. In *International Conference on Machine Learning*, pages 1469–1478. PMLR, 2020. 4

Jianfei Chen, Jun Zhu, Zi Wang, Xun Zheng, and Bo Zhang. Scalable inference for logistic-normal topic models. *Advances in Neural Information Processing Systems*, 26, 2013. 154

Badr-Eddine Chérief-Abdellatif. Convergence rates of variational inference in sparse deep learning. In *International Conference on Machine Learning*, pages 1831–1842. PMLR, 2020. 31, 35

Badr-Eddine Chérief-Abdellatif and Pierre Alquier. Consistency of variational Bayes inference for estimation and model selection in mixtures. *Electronic Journal of Statistics*, 12:2995–3035, 2018. 31

Badr-Eddine Cherief-Abdellatif and Pierre Alquier. Mmd-bayes: Robust bayesian estimation via maximum mean discrepancy. In *Symposium on Advances in Approximate Bayesian Inference*, volume 118, pages 1–21. PMLR, 2020. 90

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020. 10

Beau Coker, Wessel P Bruinsma, David R Burt, Weiwei Pan, and Finale Doshi-Velez. Wide mean-field Bayesian neural networks ignore the data. In *International Conference on Artificial Intelligence and Statistics*, pages 5276–5333. PMLR, 2022. 9, 34, 40, 43

Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R. Besold. A historical perspective of explainable Artificial Intelligence. *WIREs Data Mining and Knowledge Discovery*, 11(1), 2021. 6

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018. 34

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in Neural Information Processing Systems*, volume 29, 2016. 79

Bruno De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937. 90

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977. 10, 22

John Denker and Yann LeCun. Transforming neural-net output levels to probability distributions. *Advances in Neural Information Processing Systems*, 3, 1990. 6

John Denker, Daniel Schwartz, Ben Wittner, Sara Solla, Richard Howard, Lawrence Jackel, and John Hopfield. Large automatic learning, rule extraction, and generalization. *Complex systems*, 1(5):877–922, 1987. 6

Sameer K Deshpande, Soumya Ghosh, Tin D Nguyen, and Tamara Broderick. Are you using test log-likelihood correctly? *Transactions on machine learning research*, 2024. 43

Luc Devroye. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006. 62

Akash Kumar Dhaka, Alejandro Catalina, Manushi Welandawe, Michael R Andersen, Jonathan Huggins, and Aki Vehtari. Challenges and opportunities in high dimensional variational inference. *Advances in Neural Information Processing Systems*, 34:7787–7798, 2021. 30, 32, 37

Persi Diaconis and Donald Ylvisaker. Conjugate priors for exponential families. *The Annals of Statistics*, pages 269–281, 1979. 20

Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via χ-upper bound minimization. *Advances in Neural Information Processing Systems*, 30, 2017. 37

Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017. 28

Justin Domke. Provable smoothness guarantees for black-box variational inference. In *International Conference on Machine Learning*, pages 2587–2596. PMLR, 2020. 31

Justin Domke, Robert Gower, and Guillaume Garrigos. Provable convergence guarantees for black-box variational inference. *Advances in Neural Information Processing Systems*, 36, 2024. 32, 37

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 4, 5, 6

Francesco D' Angelo and Vincent Fortuin. Repulsive deep ensembles are Bayesian. In *Advances in Neural Information Processing Systems*, volume 34, pages 3451–3465, 2021. 14

Lan Du, Lu Ren, Lawrence Carin, and David Dunson. A Bayesian model for simultaneous image clustering, annotation and object segmentation. *Advances in Neural Information Processing Systems*, 22, 2009. 30

Daniele Durante and Tommaso Rigon. Conditionally conjugate mean-field variational Bayes for logistic models. *Statistical Science*, 34(3):472 – 485, 2019. 62, 87, 154

Bradley Efron. *Exponential families in theory and practice*. Cambridge University Press, 2022. 20

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, pages 407–451, 2004. 79

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Annual Conference on Learning Theory*, volume 49, pages 907–940. PMLR, 2016. 4

Zhou Fan, Song Mei, and Andrea Montanari. Tap free energy, spin glasses and variational inference. *The Annals of Probability*, 49(1), 2021. 29

Sebastian Farquhar and Yarin Gal. What 'out-of-distribution' is and is not. In *NeurIPS ML Safety Workshop*, 2022. 46, 53

Sebastian Farquhar, Lewis Smith, and Yarin Gal. Liberty or depth: Deep Bayesian neural nets do not need complex weight posterior approximations. *Advances in Neural Information Processing Systems*, 33:4346–4357, 2020. 34, 44

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630 (8017):625–630, 2024. 9

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(90):3133–3181, 2014. 5

Guillaume Flandin and William D Penny. Bayesian fmri data analysis with sparse spatial basis function priors. *NeuroImage*, 34(3):1108–1125, 2007. 1, 30

Seth Flaxman, Charles Whittaker, Elizaveta Semenova, Theo Rashid, Robbie Parks, Alexandra Blenkinsop, H Juliette T Unwin, Swapnil Mishra, Samir Bhatt, Deepti Gurdasani, et al. Covid-19 is a leading cause of death in children and young people ages 0-19 years in the united states. *medRxiv*, 2022. 1

Klemens Flöge, Mohammed Abdul Moeed, and Vincent Fortuin. Stein variational newton neural network ensembles. *arXiv preprint arXiv:2411.01887*, 2024. 14

Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 2023. 90

Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020. 34, 44, 47

Sandra Fortini and Sonia Petrone. Prediction-based uncertainty quantification for exchangeable sequences. *Philosophical Transactions of the Royal Society A*, 381(2247):20220142, 2023. 90

Sandra Fortini and Sonia Petrone. Exchangeability, prediction and predictive modeling in bayesian statistics. *Statistical Science*, 40(1):40–67, 2025. 90

Vincent Fortuin. Priors in Bayesian deep learning: A review. *International Statistical Review*, 90(3):563–591, 2022. 12, 59

Stefan Franssen and Botond Szabó. Uncertainty quantification for nonparametric regression using Empirical Bayesian neural networks. *arXiv preprint arXiv:2204.12735*, 2022. 9

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016. 6, 9

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 9, 14, 35, 44

Zhe Gan, Ricardo Henao, David Carlson, and Lawrence Carin. Learning deep sigmoid belief networks with data augmentation. In *International Conference on Artificial Intelligence and Statistics*, volume 38, pages 268–276. PMLR, 2015. 62

Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: A language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1682–1690. PMLR, 2018. 28

Yoav Gelberg, Tycho FA van der Ouderaa, Mark van der Wilk, and Yarin Gal. Variational inference failures under model symmetries: Permutation invariant posteriors for Bayesian neural networks. In *Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM) at ICML 2024*, pages 233–248. PMLR, 2024. 34

A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. 8, 11, 12, 32, 89

Andrew Gelman, Jessica Hwang, and Aki Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and computing*, 24:997–1016, 2014. 49

Andrew Gelman, Daniel Simpson, and Michael Betancourt. The prior can often only be understood in the context of the likelihood. *Entropy*, 19(10):555, 2017. 11, 89

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow. *arXiv preprint arXiv:2011.01808*, 2020. 5, 39, 49

Andrew Gelman, Ben Goodrich, and Geonhee Han. Grappling With Uncertainty in Forecasting the 2024 U.S. Presidential Election. *Harvard Data Science Review*, 6, 2024. 1

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984. 10

Edward I George and Robert E McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. 59

Zoubin Ghahramani and Michael Jordan. Factorial hidden markov models. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, 1995. 20

Subhashis Ghosal, Jayanta K Ghosh, and Aad W van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 2000. 31

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of Bayesian neural networks with horseshoe priors. In *International Conference on Machine Learning*, pages 1744–1753. PMLR, 2018. 35, 86, 153

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Model selection in Bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research*, 20(182):1–46, 2019. 9, 12, 35, 79, 86, 146, 153

Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness, and variational bayes. *Journal of Machine Learning Research*, 19(51): 1–49, 2018. 29

Ryan Giordano, Martin Ingram, and Tamara Broderick. Black box variational inference with a deterministic objective: Faster, more accurate, and even more black box. *Journal of Machine Learning Research*, 25(18):1–39, 2024. 28, 32, 33

Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011. 48

Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, Kolmogorov complexity, and the role of inductive biases in machine learning. In *International Conference on Machine Learning*. PMLR, 2024. 5, 6

Alex Graves. Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, 24, 2011. 12, 35

JE Griffin. Expressing and visualizing model uncertainty in Bayesian variable selection using cartesian credible sets. *arXiv preprint arXiv:2402.12323*, 2024. 73, 149

Jim E Griffin and Philip J Brown. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188, 2010. 60

Jim E Griffin and Philip J Brown. Bayesian global-local shrinkage methods for regularisation in the high dimension linear model. *Chemometrics and Intelligent Laboratory Systems*, 210:104255, 2021. 59, 60

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. *Advances in Neural Information Processing Systems*, 36:12489–12517, 2023. 9

Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C. Lawrence Zitnick, and Zachary Ward Ulissi. Fine-tuned language models generate stable inorganic materials as text. In *International Conference on Learning Representations*, 2024. 6

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), August 2018. 6

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, volume 70, pages 1321–1330. PMLR, 2017. 6

P Richard Hahn and Carlos M Carvalho. Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association*, 110(509):435–448, 2015. 73

P Hall, T Pham, MP Wand, and SSJ Wang. Asymptotic normality and valid inference for Gaussian variational approximation. *Annals of Statistics*, 2011. 31

Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, 7(10):992, 2019. 4

L.K. Hansen and P. Salamon. Neural network ensembles. *Transactions on Pattern Analysis and Machine Intelligence*, 12, 1990. 14

David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1): 81–102, 1978. 83, 148

James Harrison, John Willes, and Jasper Snoek. Variational Bayesian last layers. In *Symposium on Advances in Approximate Bayesian Inference*, 2024. 35

Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Symposium on the Theory of Computing*, 1986. 4

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 10

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, pages 1026–1034, 2015. 3, 4, 42, 79

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL `http://github.com/google/flax`. 42

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Conference on Computer Vision and Pattern Recognition*, pages 41–50. IEEE, 2019. 6

Conor Heins, Hao Wu, Dimitrije Markovic, Alexander Tschantz, Jeff Beck, and Christopher Buckley. Gradient-free variational learning with conditional mixture networks. In *NeurIPS Workshop on Bayesian Decision-making and Uncertainty*, 2024. 62

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 6, 46

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *arXiv preprint arXiv:2109.13916*, 2021. 6

Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard Turner. Black-box alpha divergence minimization. In *International Conference on Machine Learning*, pages 1511–1520. PMLR, 2016. 37

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Annual Conference on Computational Learning Theory*, page 5–13. Association for Computing Machinery, 1993. 7, 13, 17, 34

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. 6

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999. 8, 40, 51

Matthew D Hoffman and David M Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pages 361–369, 2015. 24

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. 10, 42, 79

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013. 22, 57, 70, 71, 72, 85

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. 4

Alexandra Hotti, Oskar Kviman, Ricky Molén, Víctor Elvira, and Jens Lagergren. Efficient mixture learning in black-box variational inference. In *International Conference on Machine Learning*. PMLR, 2024. 32

Jiri Hron, Alex Matthews, and Zoubin Ghahramani. Variational Bayesian dropout: pitfalls and fixes. In *International Conference on Machine Learning*, volume 80, pages 2019–2028. PMLR, 2018. 36

Jiri Hron, Roman Novak, Jeffrey Pennington, and Jascha Sohl-Dickstein. Wide Bayesian neural networks have a simple weight posterior: theory and accelerated sampling. In *International Conference on Machine Learning*, volume 162, pages 8926–8945. PMLR, 2022. 9, 43

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2022. 14

Jonathan Huggins, Mikolaj Kasprzak, Trevor Campbell, and Tamara Broderick. Validated variational inference via practical posterior error bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1802. PMLR, 2020. 29, 30, 37

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *International Conference on Machine Learning*, volume 139, pages 4629–4640. PMLR, 2021. 9, 44, 45

Gerritsma J., Onnink R., and Versluis A. Yacht hydrodynamics, 2013. URL `https://archive.ics.uci.edu/dataset/243/yacht+hydrodynamics`. 84, 149

Tommi S. Jaakkola and Michael I. Jordan. A variational approach to Bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, volume R1 of *Proceedings of Machine Learning Research*, pages 283–294. PMLR, 04–07 Jan 1997. 17

Tommi S. Jaakkola and Michael I. Jordan. *Improving the Mean Field Approximation Via the Use of Mixture Distributions*, pages 163–173. Springer Netherlands, Dordrecht, 1998. 7, 23, 32, 34

Sanket Jantre, Shrijita Bhattacharya, and Tapabrata Maiti. Spike-and-slab shrinkage priors for structurally sparse Bayesian neural networks. *Transactions on Neural Networks and Learning Systems*, 2024. 35

Antoran Javier. Bayesian neural networks, 2019. URL `https://github.com/JavierAntoran/Bayesian-Neural-Networks`. 146

Harold Jeffreys. *The theory of probability*. Oxford, 1939. 8

Jack Jewson, Jim Q Smith, and Chris Holmes. Principles of bayesian inference using general divergence criteria. *Entropy*, 20(6):442, 2018. 89

Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999. 10, 17, 31, 34, 36

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995. 8, 40

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017. 9

Mohammad Emtiyaz Khan and Håvard Rue. The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46, 2023. 13

Kyurae Kim, Yian Ma, and Jacob Gardner. Linear convergence of black-box variational inference: Should we stick the landing? In *International Conference on Artificial Intelligence and Statistics*, pages 235–243. PMLR, 2024a. 32

Kyurae Kim, Jisu Oh, Kaiwen Wu, Yian Ma, and Jacob Gardner. On the convergence of black-box variational inference. *Advances in Neural Information Processing Systems*, 36, 2024b. 32, 37

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2014. 25, 26, 33, 154

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, volume 28, 2015. 14, 36

Leo Klarner, Tim G. J. Rudner, Michael Reutlinger, Torsten Schindler, Garrett M Morris, Charlotte Deane, and Yee Whye Teh. Drug discovery under covariate shift with domain-informed prior distributions over functions. In *International Conference on Machine Learning*, volume 202, pages 17176–17197. PMLR, 2023. 9

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An optimization-centric view on bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022. 10, 89

Joohwan Ko, Kyurae Kim, Woo Chang Kim, and Jacob R. Gardner. Provably scalable black-box variational inference with structured variational families. In *International Conference on Machine Learning*, pages 24896–24931. PMLR, 2024. 32

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *International Conference on Machine Learning*, volume 119, pages 5436–5446. PMLR, 2020. 154

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, 2017. 4, 5, 6

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017. 25, 27, 30, 79

Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. Arviz a unified library for exploratory analysis of Bayesian models in python. *Journal of Open Source Software*, 4(33):1143, 2019. 42

Oskar Kviman, Harald Melin, Hazal Koptagel, Victor Elvira, and Jens Lagergren. Multiple importance sampling elbo and deep ensembles of variational approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 10687–10702. PMLR, 2022. 52

Jinlin Lai and Yuling Yao. Predictive variational inference: Learn the predictively optimal posterior distribution. *arXiv preprint arXiv:2410.14843*, 2024. 90

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. 14, 52, 78

Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural Networks*, 14(3):257–274, 2001. 12

Pierre Simon Laplace. Mémoire sur la probabilité des causes par les événements (1774). *Œuvres compl*, pages 27–65, 1891. 7

J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. *International Conference on Learning Representations*, 2018. 61

Kyeongwon Lee and Jaeyong Lee. Asymptotic properties for Bayesian neural network in besov space. *Advances in Neural Information Processing Systems*, 35:5641–5653, 2022. 59

E. Levin, N. Tishby, and S.A. Solla. A statistical approach to learning and generalization in layered neural networks. *Proceedings of the IEEE*, 78:1568–1574, 1990. 14

Hanning Li and Debdeep Pati. Variable selection using shrinkage priors. *Computational Statistics and Data Analysis*, 2017. 73

Junbo Li, Zichen Miao, Qiang Qiu, and Ruqi Zhang. Training Bayesian neural networks with sparse subspace variational inference. In *International Conference on Learning Representations*, 2024. 36

Qing Li and Nan Lin. The Bayesian elastic net. *Bayesian Analysis*, 5(1):151 – 170, 2010. 73, 79

Percy Liang, Michael Jordan, and Dan Klein. Probabilistic grammars and hierarchical dirichlet processes. In *The Oxford Handbook of Applied Bayesian Analysis*. Oxford University Press, 10 2013. 20

Scott Linderman, Ryan P Adams, and Jonathan W Pillow. Bayesian latent structure discovery from multi-neuron recordings. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, 2016. 62

Bruce G Lindsay. Mixture models: Theory, geometry and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, pages i–163. JSTOR, 1995. 20

Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57, 2018. 1, 6

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 31, 2018. 154

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in Neural Information Processing Systems*, page 3290–3300, 2017. 35, 86

Lu Lu. Dying relu and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28(5):1671–1706, 2020. 4

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: a view from the width. *Advances in Neural Information Processing Systems*, page 6232–6240, 2017. 4

Kelly M., Longjohn R., and Nottingham K. The UCI machine learning repository, 2007. URL `https://archive.ics.uci.edu`. 79, 83

David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 05 1992. 7, 13, 17, 40

David JC MacKay. Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469, 1995. 36

David J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. 5, 19, 29, 40

Martin Magris and Alexandros Iosifidis. Bayesian learning for neural networks: an algorithmic survey. *Artificial Intelligence Review*, 56(10):11773–11823, 2023. 40

Charles Margossian and Lawrence K Saul. Variational inference in location-scale families: Exact recovery of the mean and correlation matrix. In *International Conference on Artificial Intelligence and Statistics*, 2024. 30, 32, 34, 37

Charles C. Margossian and David M. Blei. Amortized variational inference: when and why? In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2024. 33

Charles C Margossian, Loucas Pillaud-Vivien, and Lawrence K Saul. Variational inference for uncertainty quantification: an analysis of trade-offs. *arXiv preprint arXiv:2403.13748*, 2024. 30, 37

Gael M Martin, David T Frazier, and Christian P Robert. Computing Bayes: Bayesian computation from 1763 to the 21st century. *arXiv preprint arXiv:2004.06425*, 2020. 11

Luca Martino, Victor Elvira, and Gustau Camps-Valls. Group importance sampling for particle filtering and mcmc. *Digital Signal Processing*, 82:133–151, 2018. 10

Takuo Matsubara, Jeremias Knoblauch, François-Xavier Briol, and Chris J Oates. Robust generalised Bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):997–1022, 2022. 10

A. G. Matthews, J. Hron, M. Rowland, R.E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *International Conference on Learning Representations*, 2018. 9, 12, 40, 61

Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: advantages of Bayesian deep learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4745–4753, 2017. 9

J McCarthy. Defending ai research: A collection of essays and reviews; mccarthy, j., ed. *Center for the Study of Language and Information: Stanford, CA, USA*, 1996. 1

John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence august 31, 1955. *AI Magazine*, 27, 2006. 1

Luckeciano Carvalho Melo, Panagiotis Tigas, Alessandro Abate, and Yarin Gal. Deep Bayesian active learning for preference modeling in large language models. *Advances in Neural Information Processing Systems*, 37:118052–118085, 2024. 9

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, 2013. 4

Andrew C. Miller, Nicholas J. Foti, and Ryan P. Adams. Variational boosting: Iteratively refining posterior approximations. In *International Conference on Machine Learning*, volume 70, pages 2420–2429. PMLR, 2017. 32

Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001. 10, 17, 30

Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988. 59

Chirag Modi, Robert Gower, Charles Margossian, Yuling Yao, David Blei, and Lawrence Saul. Variational inference with Gaussian score matching. *Advances in Neural Information Processing Systems*, 36, 2024. 37

Gemma E Moran, John P Cunningham, and David M Blei. The posterior predictive null. *Bayesian Analysis*, 18(4):1071–1097, 2023. 8

Kevin P Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. 3

Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999. 10, 17

Eric Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. Dropout as a structured shrinkage prior. In *International Conference on Machine Learning*, pages 4712–4722. PMLR, 2019. 12, 14, 36

Radford Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, volume 5, 1992a. 7

Radford M Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992b. 7, 34

Radford M. Neal and Geoffrey E. Hinton. *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer Netherlands, Dordrecht, 1998. 22

R.M. Neal. Bayesian learning for neural networks. *Springer*, 118, 1995. 2, 7, 9, 10, 40, 61

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition*, pages 427–436. IEEE, 2015. 6

Oscar Oelrich, Shutong Ding, Måns Magnusson, Aki Vehtari, and Mattias Villani. When are Bayesian model probabilities overconfident? *arXiv preprint arXiv:2003.04026*, 2020. 41

Ilsang Ohn and Lizhen Lin. Adaptive variational bayes: Optimality, computation and applications. *The Annals of Statistics*, 52(1):335–363, 2024. 52, 55, 78

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 154

Nathan Osborne, Christine B Peterson, and Marina Vannucci. Latent network estimation and variable selection for compositional data via variational em. *Journal of Computational and Graphical Statistics*, 31(1):163–175, 2022. 69

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 9, 41, 46

Theodore Papamarkou, Jacob Hinkle, M. Todd Young, and David Womble. Challenges in Markov chain Monte Carlo for Bayesian neural networks. *Statistical Science*, 37(3):425 – 442, 2022. 9, 11, 40, 42, 44, 78

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, et al. Position: Bayesian deep learning is needed in the age of large-scale ai. In *International Conference on Machine Learning*, pages 39556–39586. PMLR, 2024. 1, 9

G Parisi. *Statistical Field Theory*. Wiley, Reading: MA, 1988. 17, 19, 22

Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008. 60, 79, 151

Yookoon Park and David Blei. Density uncertainty layers for reliable uncertainty estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 163–171. PMLR, 2024. 34, 46

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. Uncertainty in neural networks: Approximately Bayesian ensembling. In *International Conference on Artificial Intelligence and Statistics*, pages 234–244. PMLR, 2020. 14

Stefano Peluchetti, Stefano Favaro, and Sandra Fortini. Stable behaviour of infinitely wide deep neural networks. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 1137–1146, 2020. 12, 61

C. Peterson and JR Anderson. A mean field theory learning algorithm for neural network. *Complex Systems*, 1:995–1019, 1987. 6, 17

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. In *Program Transformations for ML Workshop at NeurIPS*, 2019. 28, 42, 79

Juho Piironen and Aki Vehtari. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2016. 49

Juho Piironen and Aki Vehtari. On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 905–913. PMLR, 20–22 Apr 2017a. 35, 86, 153

Juho Piironen and Aki Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2): 5018, 2017b. 86, 153

Juho Piironen, Markus Paasiniemi, and Aki Vehtari. Projective inference in high-dimensional problems: Prediction and feature selection. *Electronic Journal of Statistics*, 14(1):2155 – 2197, 2020. 73

Nicholas G Polson and Veronika Ročková. Posterior concentration for sparse deep learning. *Advances in Neural Information Processing Systems*, 31, 2018. 59, 123

Nicholas G Polson and James G Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian statistics*, 9(501-538):105, 2010. 12, 20, 56, 60, 153

Nicholas G. Polson and Vadim Sokolov. Bayesian regularization: From tikhonov to horseshoe. *WIREs Comput. Stat.*, 11(4), 2019. ISSN 1939-5108. 12, 13, 40

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013. 56, 58, 60, 62, 154

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 4

Rajesh Ranganath, Chong Wang, Blei David, and Eric Xing. An adaptive learning rate for stochastic variational inference. In *International Conference on Machine Learning*, pages 298–306. PMLR, 2013. 23, 72, 85

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, volume 33, pages 814–822, Reykjavik, Iceland, 2014. PMLR. 15, 25

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333. PMLR, 2016. 32, 36

Carl Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems*, volume 13, 2000. 40

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. 10, 12, 40

Kolyan Ray and Botond Szabó. Variational Bayes for high-dimensional linear regression with sparse priors. *Journal of the American Statistical Association*, 117(539):1270–1281, 2022. 31

Kolyan Ray, Botond Szabó, and Gabriel Clara. Spike and slab variational Bayes for high dimensional logistic regression. *Advances in Neural Information Processing Systems*, 33:14423–14434, 2020. 31

Jack Raymond and Federico Ricci-Tersenghi. Improving variational methods via pairwise linear response identities. *Journal of Machine Learning Research*, 18 (6):1–36, 2017. 29

Manuel J Reyes-Gomez, Daniel PW Ellis, and Nebojsa Jojic. Multiband audio modeling for single-channel acoustic source separation. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–641. IEEE, 2004. 30

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015. 32, 36

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014. 26, 33, 154

Jorma Rissanen. *Minimum Description Length Principle*, pages 666–668. Springer US, 1986. 34

Hippolyt Ritter, Martin Kukla, Cheng Zhang, and Yingzhen Li. Sparse uncertainty representation in deep learning with inducing weights. *Advances in Neural Information Processing Systems*, 34:6515–6528, 2021. 46

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. 23, 72

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999. 11, 30, 42

Donald B Rubin. The Bayesian bootstrap. *The Annals of Statistics*, pages 130–134, 1981. 52

Tim GJ Rudner, Zonghao Chen, and Yarin Gal. Rethinking function-space variational inference in Bayesian neural networks. In *Symposium on Advances in Approximate Bayesian Inference*, 2021. 35

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 2, 3

John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016. 28

Masa-Aki Sato. Online model selection based on the variational bayes. *Neural Computation*, 13(7):1649–1681, 2001. 22, 71

Lawrence K. Saul, Tommi Jaakkola, and Michael I. Jordan. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76, 1996. 7, 34

Leonard J Savage. *The foundations of statistics*. Courier Corporation, 1972. 90

Christian Schäfer and Nicolas Chopin. Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing*, 23:163–184, 2013. 149

Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International Conference on Machine Learning*, pages 343–351. PMLR, 2013. 72, 85

James G Scott and Liang Sun. Expectation-maximization for logistic regression. *arXiv preprint arXiv:1306.0040*, 2013. 62

Torben Sell and Sumeetpal Sidhu Singh. Trace-class Gaussian priors for Bayesian learning of neural networks with mcmc. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1):46–66, 2023. 61

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. 5

Alisa Sheinkman and Sara Wade. Variational Bayesian bow tie neural networks with shrinkage. *arXiv preprint arXiv:2411.11132*, 2024. 15, 56

Alisa Sheinkman and Sara Wade. Understanding the trade-offs in accuracy and uncertainty quantification: Architecture and inference choices in Bayesian neural networks. *arXiv preprint arXiv:2503.11808*, 2025. 15, 39

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 4

Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark van der Wilk, Adam Foster, and Tom Rainforth. Rethinking aleatoric and epistemic uncertainty. In *NeurIPS Workshop on Bayesian Decision-making and Uncertainty*, 2024. 9

Jimmy TH Smith, Dieterich Lawson, and Scott W Linderman. Bayesian inference in augmented bow tie networks. *NeurIPs Bayesian Deep Learning Workshop*, 2021. 56, 57, 58, 59, 61, 85

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *Conference on Uncertainty in Artificial Intelligence*, pages 560–569, 2018. 9

Qifan Song. Bayesian shrinkage towards sharp minimaxity. *Electronic Journal of Statistics*, 14:2714–2741, 2020. 56

Qifan Song and Faming Liang. Nearly optimal Bayesian shrinkage for high-dimensional regression. *Science China Mathematics*, 66(2):409–442, 2023. 56, 60

David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the royal statistical society: Series b (statistical methodology)*, 64(4):583–639, 2002. 48

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 4, 14

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational Bayesian neural networks. In *International Conference on Learning Representations*, 2019a. 35

Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *Transactions on Cybernetics*, 50 (8):3668–3681, 2019b. 3

Yan Sun, Qifan Song, and Faming Liang. Learning sparse deep neural networks with a spike-and-slab prior. *Statistics & Probability Letters*, 180:109246, 2022. 59

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1, 6

Matus Telgarsky. Benefits of depth in neural networks. In *Annual Conference on Learning Theory*, volume 49, pages 1517–1539. PMLR, 2016. 4

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 13, 82

Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986. 10

Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001. 60

Naftali Tishby, Esther Levin, and Sara A. Solla. Consistent inference of probabilities in layered networks: predictions and generalizations. *International Joint Conference on Neural Networks*, pages 403–409 vol.2, 1989. 6

Michalis K Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. *Advances in Neural Information Processing Systems*, 28, 2015. 26

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 5

Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. *Advances in Neural Information Processing Systems*, 28, 2015. 32

Brian Trippe and Richard Turner. Overpruning in variational Bayesian neural networks. *arXiv preprint arXiv:1801.06230*, 2018. 9, 34, 44

Athanasios Tsanas and Angeliki Xifara. Energy efficiency, 2012. URL `https://archive.ics.uci.edu/dataset/242/energy+efficiency`. 84, 149

Richard Eric Turner and Maneesh Sahani. *Two problems with variational expectation maximisation for time series models*, page 104–124. Cambridge University Press, 2011. 29, 33, 44

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

Aki Vehtari and Janne Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142 – 228, 2012. 48, 51

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, 2016. ISSN 1573-1375. 48, 49

Aki Vehtari, Andrew Gelman, Tuomas Sivula, Pasi Jylänki, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P Cunningham, David Schiminovich, and Christian P Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21(17):1–53, 2020. 30, 37

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25(72):1–58, 2024. 49

Aki Vetari, Jonah Gabry, Måns Magnusson, Yuling Yao, and Andrew Gelman. Efficient leave-one-out cross-validation and waic for Bayesian models, 2019. URL `https://mc-stan.org/loo`. 49

Mariia Vladimirova, Jakob Verbeek, Pablo Mesejo, and Julyan Arbel. Understanding priors in Bayesian neural networks at the unit level. In *International Conference on Machine Learning*, volume 97, pages 6458–6467. PMLR, 2019. 13, 40

Mariia Vladimirova, Julyan Arbel, and Stéphane Girard. Dependence between Bayesian neural network units. In *Bayesian Deep Learning NeurIPS*, pages 1–9, 2021. 13, 61

Martin J. Wainwright and Michael I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*, volume 1. Now Publishers Inc., Hanover, MA, USA, January 2008. 29

S. G. Walker. Bayesian inference via a minimization rule. *Sankhya*, 68(4):542–553, 2006. 10, 89

Matthew P Wand, John T Ormerod, Simone A Padoan, and Rudolf Frührwirth. Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6 (4):1–48, 2011. 86, 153

Bo Wang and DM Titterington. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. In *Conference on Uncertainty in Artificial Intelligence*, pages 577–584, 2004a. 31

Bo Wang and Donald M. Titterington. Lack of consistency of mean field and variational Bayes approximations for state space models. *Neural Processing Letters*, 20(3):151–170, 2004b. 29

Dilin Wang, Hao Liu, and Qiang Liu. Variational inference with tail-adaptive f-divergence. *Advances in Neural Information Processing Systems*, 31, 2018. 37

Yixin Wang and David Blei. Variational Bayes under model misspecification. *Advances in Neural Information Processing Systems*, 32, 2019a. 31

Yixin Wang and David M Blei. Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161, 2019b. 31

Yuexi Wang, Nicholas Polson, and Vadim O Sokolov. Data augmentation for Bayesian deep learning. *Bayesian Analysis*, 18(4):1041–1069, 2023. 62

Sumio Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(116):3571–3594, 2010. 48, 49

Manushi Welandawe, Michael Riis Andersen, Aki Vehtari, and Jonathan H. Huggins. A framework for improving the reliability of black-box variational inference. *Journal of Machine Learning Research*, 25(219):1–71, 2024. 30

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, pages 681–688. PMLR, 2011. 10

Florian Wenzel, Théo Galy-Fajou, Christan Donner, Marius Kloft, and Manfred Opper. Efficient Gaussian process classification using pòlya-gamma data augmentation. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 5417–5424, 2019. 62

Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020a. 12, 44, 79

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems*, 33:6514–6527, 2020b. 14

T Westling and TH McCormick. Beyond prediction: A framework for inference with variational approximations in mixture models. *Journal of Computational and Graphical Statistics*, 28(4):778–789, 2019. 31

Veit David Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. A rigorous link between deep ensembles and (variational) Bayesian methods. *Advances in Neural Information Processing Systems*, 36:39782–39811, 2023. 14, 41

Christopher Williams. Computing with infinite networks. In *Advances in Neural Information Processing Systems*, volume 9, 1996. 40

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in Neural Information Processing Systems*, 33:4697–4708, 2020. 8, 11, 14, 40, 52

Andrew Gordon Wilson. The case for Bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020. 9

David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 10 1996. ISSN 0899-7667. 5

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. Deterministic variational inference for robust Bayesian neural networks. In *International Conference on Learning Representations*, 2018. 35

Bohan Wu and David Blei. Extending mean-field variational inference via entropic regularization: Theory and computation. *arXiv preprint arXiv:2404.09113*, 2024. 29, 31

Luhuan Wu and Sinead A Williamson. Posterior uncertainty quantification in neural networks using data augmentation. In *International Conference on Artificial Intelligence and Statistics*, volume 238, pages 3376–3384. PMLR, 2024. 14, 52, 90

Luhuan Wu and Sinead A Williamson. Posterior uncertainty quantification in neural networks using data augmentation. In *International Conference on Artificial Intelligence and Statistics*, pages 3376–3384. PMLR, 2024. 78

S. N. MacEachern X. Xu, P. Lu and R. Xu. Calibrated Bayes factors for model comparison. *Journal of Statistical Computation and Simulation*, 89(4):591–614, 2019. 8

Yun Yang, Debdeep Pati, and Anirban Bhattacharya. α-variational inference with statistical guarantees. *The Annals of Statistics*, 48(2):886–905, 2020. 31

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13(3), 2018a. 52

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In *International Conference on Machine Learning*, pages 5581–5590. PMLR, 2018b. 29, 30, 49, 52

Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for non-mixing Bayesian computations: The curse and blessing of multimodal posteriors. *Journal of Machine Learning Research*, 23(1):3426–3471, 2022. 55, 78

I-Cheng Yeh. Concrete compressive strength, 2007. URL `https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength`. 84, 149

I-Cheng Yeh. Concrete slump test, 2009. URL `https://archive.ics.uci.edu/dataset/182/concrete+slump+test`. 84, 149

Arnold Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280, 1988. 89

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a. 5

Fengshuo Zhang and Chao Gao. Convergence rates of variational posterior distributions. *The Annals of Statistics*, 48(4):2180–2207, 2020. 31

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, volume 97, pages 7354–7363. PMLR, 2019. 154

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020a. 10

Tianren Zhang, Chujie Zhao, Guanyu Chen, Yizhou Jiang, and Feng Chen. Feature contamination: Neural networks learn uncorrelated features and fail to generalize. In *International Conference on Machine Learning*, pages 60446–60495. PMLR, 2024. 6

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41, 2020b. 6

Yan Dora Zhang, Weichang Yu, and Howard D Bondell. Variable selection with shrinkage priors via sparse posterior summaries. In *Handbook of Bayesian Variable Selection*, pages 179–198. Chapman and Hall/CRC, 2021b. 73

Yuren Zhou, Yuqi Gu, and David B Dunson. Bayesian deep generative models for multiplex networks with multiscale overlapping clusters. *arXiv preprint arXiv:2405.20936*, 2024. 62

Yongshuo Zong, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales. Fool your vision and language model with embarrassingly simple permutations. In *International Conference on Machine Learning*, volume 235, pages 62892–62913. PMLR, 2024. 6

Appendix A

Supplementary To the Empirical Example

This appendix supplements empirical examples considered in Sections 3.2 and 3.3.

A.1 Metrics and practicalities

Recall that we denoted the training data to be $\mathcal{D} = \{x_n, y_n\}_{i=1}^N$ and the new data for testing to be $\tilde{\mathcal{D}} = \{\tilde{x}_n, \tilde{y}_n, \}_{n=1}^{\tilde{N}}$. Denote the approximated posterior predictive mean as $\boldsymbol{y}$, the set of S samples of the signal as $\boldsymbol{\mu}^S$ and of the observations as $\boldsymbol{y}^S$. Upon computing the posterior predictive distribution, we obtain the root mean squared error and the empirical coverage as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n}^{N}[(\tilde{y}_n - \mathbb{E}^S[y_i^S])^2]},$$

$$\text{EC} = \frac{\#\{\boldsymbol{y}_{\in}[q_{0.025}, q_{0.975}]\}}{N}, \text{ where } q \text{ are quantiles of } \boldsymbol{\mu}^S \text{ or } \boldsymbol{y}^S.$$

The results of Section 3.2.2 are obtained when the number of iterations of mfVI is set 10^4, 10^4, 5×10^4, 6×10^4 to train models with, respectively, $20, 200, 1000, 2000$ hidden units in a layer. In models trained with HMC, the number of samples used for warmup was set to 10^3, the samples used for posterior is 10^3 in models with 20 hidden units, and 2×10^3 in models with $200, 1000, 2000$ hidden units in a layer. In experiments of Section 3.2.3, the number of iterations of mfVI was set to $L \times 10^4$; the HMC had the number of warmup samples fixed to 10^3, and the number of samples was $\min(4 \times 10^3, L \times 10^3)$.

Remark on the initialization: Based on the empirical evidence, we observed that in our experiment for $L = 1, 2$ the NumPyro implementation of mfVI requires the initialization mode to be set to "init to feasible", which chooses initialization point uniformly (ignoring the prior distribution). Whereas for $L = 3, 4, 5, 6$ mfVI requires "init to mean", which sets initial parameters to the prior mean, and "init to feasible" will fail. Conversely, the NumPyro implementation of the HMC fails if the initialization location is set to "init to mean" but performs fine if it is always set to "init to feasible", i.e. ignoring the distribution

parameters.

A.2 Correspondence between WAIC and RMSE

In Section 3.3.2 we compared the RMSE obtained in the OOD experiment to the estimates of the log pointwise predictive density obtained with LOO-CV. Computing $\widehat{\text{elpd}}_{\text{loo}}$ involved Pareto smoothed importance sampling, and in some of the models, the estimated shape parameter of the generalized Pareto distribution gave a warning about the reliability of the LOO estimate. Here, we do an extra step and check if the WAIC and LOO estimates of the elpd agree. Figure A.1 illustrates the reverse dependency between the RMSE and $\widehat{\text{elpd}}_{\text{WAIC}}$ and is largely identical to Figure 3.5c (in terms of the location of coordinates but not the error bars). Therefore, we can conclude that LOO estimates obtained in Section 3.3.2 can be seen as relatively reliable.

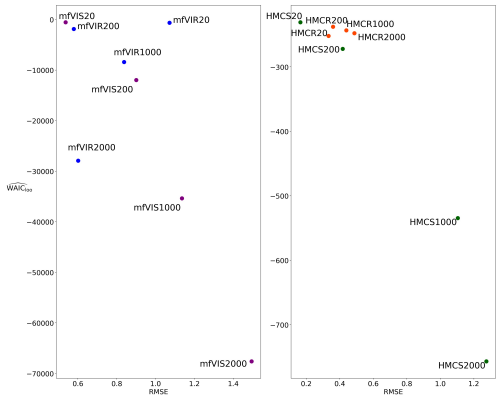


Figure A.1: Estimating the out-of-distribution performance before seeing the new data: the correspondence between the $\widehat{\text{elpd}}_{\text{WAIC}}$ and the RMSE in the OOD scenario. Similarly to $\widehat{\text{elpd}}_{\text{loo}}$, the higher $\widehat{\text{elpd}}_{\text{WAIC}}$ should correspond to lower RMSE.

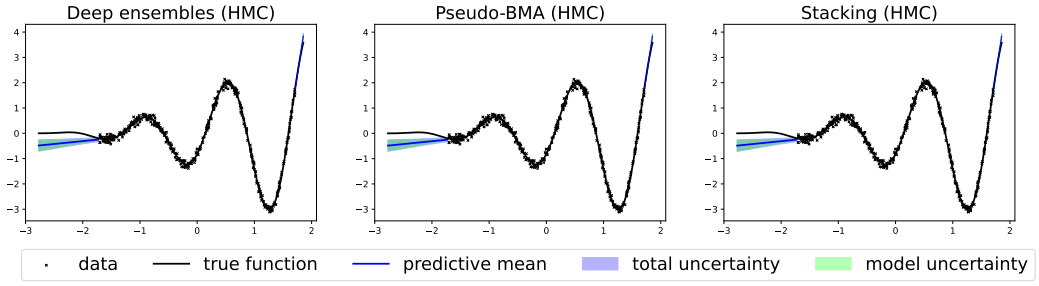
A.3 Supplementary to ensembles and averages

Remark on constructing deep ensembles. Given $\mathcal{M} = \{M_1, \ldots, M_K\}$ a collection of models suppose that K approximations $\tilde{\boldsymbol{y}}_k$ of the posterior $p(\tilde{\boldsymbol{y}}|\mathcal{D}, M_k)$ have means μ_k and variances σ_k^2 or $k = 1, \ldots, K$. Then the mean and variance of an ensemble of approximations:

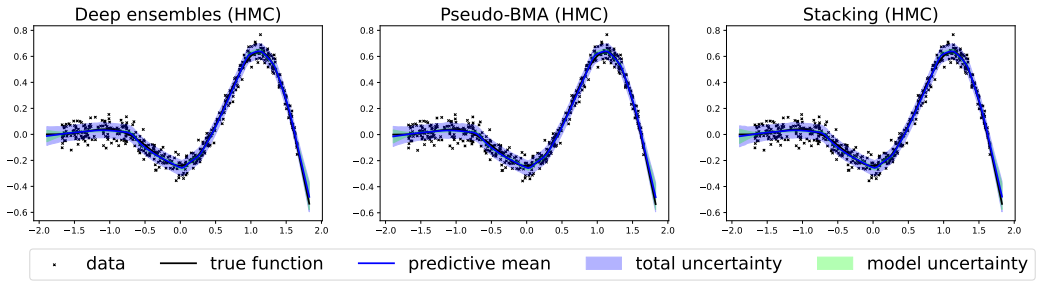
$$\mu_{\text{DE}} = K^{-1} \sum_{1}^{K} \mu_k, \quad \sigma_{\text{DE}}^2 = K^{-1} \left(\sum_{1}^{K} \sigma_k^2 + \mu_k^2 \right) - \mu_{\text{DE}}^2.$$

In general, given weights $\omega_k = p(M = M_k)$ the mean and the variance are

$$\mu_{\text{DE}} = \mathbb{E}[\mathbb{E}[\tilde{\boldsymbol{y}}|M]] = \sum_{1}^{K} \omega_k \mu_k,$$



(a) The complement-distributions task. Predictions are very similar.



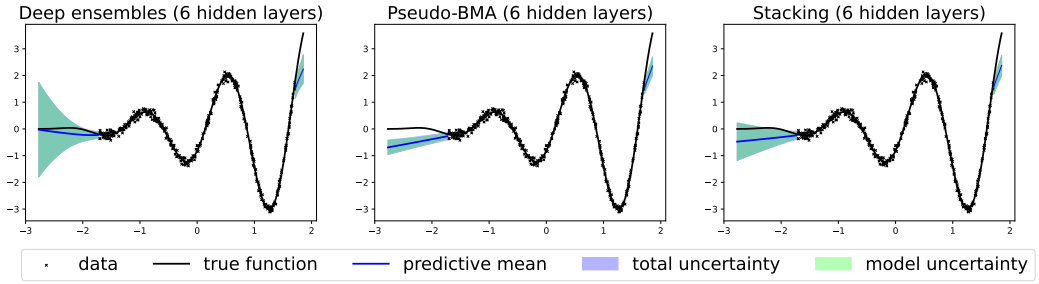
(b) The related-distributions task. Predictions are (again) strikingly similar.

Figure A.2: Predictions obtained by ensembling, stacking and pseudo-BMA when applied to HMCR20 in the complement-distributions and related-distributions tasks.

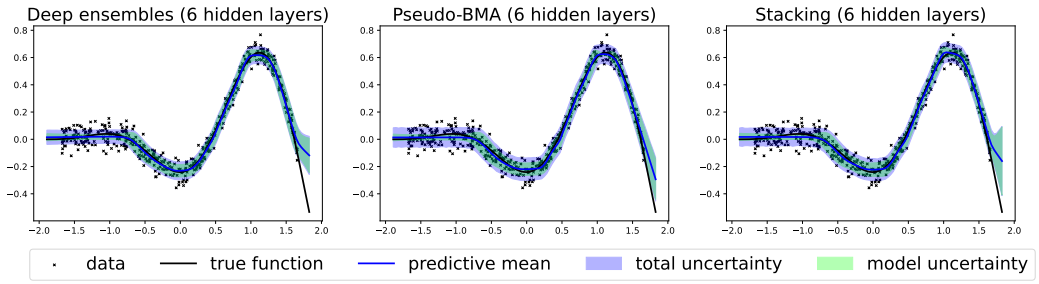
$$\sigma^2_{\text{DE}} = \mathbb{E}[\mathbb{E}[\tilde{\boldsymbol{y}}^2|M]] - \mu^2_{\text{DE}} = \sum_1^K \omega_k \mathbb{E}[\tilde{\boldsymbol{y}}_k{}^2] - \mu^2_{\text{DE}}$$
$$= \left(\sum_1^K \omega_k(\sigma_k^2 + \mu_k^2)\right) - \mu^2_{\text{DE}}.$$

We wish to recreate the experiment we did in Section 3.3.4 for the HMCR20 model instead of the mfVIR20 model. We choose 10 random initialization points, obtain 10 posterior predictive distributions and compute estimated expected log pointwise predictive densities. We then construct ensemble, pseudo-BMA and stacking approximations for the complement-distribution task, the results are illustrated by Figure A.2a and the predictions of the related-distributions task are shown on Figure A.2b. In contrast to mfVIR20, this time we do not observe a clear difference between the pseudo-BMA and stacking and ensembling methodologies. Moreover, all the approaches require considerable computational resources (for 10 random runs) but do not provide a considerable improvement in RMSE and empirical coverage compared to a single random run of the model. We conclude that in this particular example, ensembling, stacking and pseudo-BMA do not help explore different modes of the posterior and so cannot be recommended when dealing with HMC.

Now recall the neural network considered in Section 3.2.3. Based on the 10 posterior predictive distributions obtained starting from 10 different random initialization points, we construct an ensemble, pseudo-BMA and stacking approximations for the mfVIR and mfVIS models with $L = 6$ hidden layers. The



(a) The complement-distributions task. DE and stacking are preferable over pseudo-BMA.



(b) The related-distributions task. DE and stacking are preferable over pseudo-BMA.

Figure A.3: Predictions obtained by ensembling, stacking and pseudo-BMA when applied to mfVIR20 with $L = 6$ in the complement-distributions and related-distributions tasks.

results are consistent with the observation made in Chapter 3; in the complement-distribution task (Figure A.3b), pseudo-BMA is confirmed to be inferior to stacking and deep ensembles of BNNs. In the related distribution task (Figure A.3a), in terms of both accuracy and uncertainty quantification, stacking is preferable over deep ensembles and pseudo-BMA, with the latter performing better than ensembles (unlike in the one-layer case).

A.4 Experiments with Student-t priors

We consider a Bayesian neural network, which is defined similarly to Equation (3.1) but with the Student-t priors, that is

$$\begin{aligned}
&\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{b}_{L+1} + \boldsymbol{W}_{L+1}\boldsymbol{z}_L, \boldsymbol{\sigma}^2\right), \quad \boldsymbol{\sigma} \sim |\mathcal{N}(0, 1e-6)|, \\
&\boldsymbol{z}_l = g\left(\boldsymbol{b}_l + \boldsymbol{W}_l \boldsymbol{z}_{l-1}\right) \text{ for } l = 1, \ldots, L,
\end{aligned}$$

and the following priors on the weights and biases:

$$\begin{aligned}
&\boldsymbol{W}_1 \sim \mathrm{ST}\left(\mathbf{0}, \frac{\mathbf{1}}{LD_0}\right), \boldsymbol{b}_l \sim \mathrm{N}\left(\mathbf{0}, \frac{\mathbf{1}}{4L}\right), \\
&\boldsymbol{W}_l \sim \mathrm{ST}\left(\mathbf{0}, \frac{\mathbf{2}}{D_{l-1}}\right) \text{ for } l = 2, \ldots, L+1,
\end{aligned}$$

where we consider two different choices of activation functions, namely, the ReLU and the sigmoid.

First, similar to Section 3.2.2, consider the performances of mfVIR, mfVIS, HMCR and HMCS with Student-t priors with 1 hidden layer and either Gaussian or Student-t priors as the width increases, and illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units by the Figure A.4a. The predictions of the four combinations of activation and inference algorithm with Student-t priors when $D_1 = 2000$ are provided on the Figure 3.2b; For either choice of priors, performance of the mfVIS dips with the increase in the dimension of the hidden layer; moreover, for $D_1 = 1000$ and $D_1 = 2000$ its posterior predictive distribution fails to capture the data, and, in fact, degenerates to the prior (see Figure A.4b here and Figure 3.2b in Section 3.2.2). In terms of predictive accuracy, HMC is preferred over mfVI in all of the combinations of the activation function and width. However, in terms of uncertainty quantification, the HMC is inferior to mfVI (with one exception of a BNN with Student-t priors, sigmoid activation and 2000 hidden units). There is no considerable difference with the results for Gaussian priors.

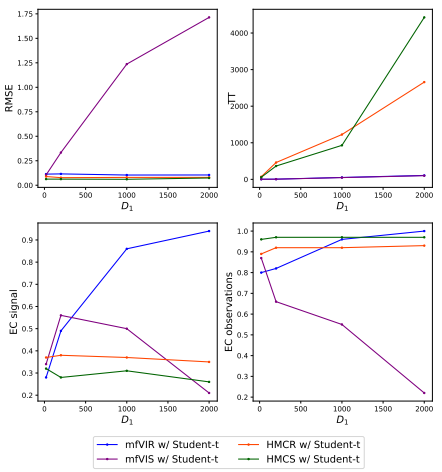
Consider the neural networks with Student-t priors and with the number of layers L varying from 1 to 6 and a fixed number of hidden units in each layer $D_h = 20$. Figure A.4c provides the recorded metrics, and Figure A.4d illustrates the predictions of the four combinations of activation and inference algorithm with $L = 6$. The performance of Student-t priors is very similar to Gaussian priors, with one exception of $L = 5$ and Student-t priors, when the prediction quality of the network drops drastically.

Further, we consider the 'complement-distributions' data of Section 3.2.4. On Figure A.5a we illustrate the metrics for $D_1 = 20, 200, 1000$ and 2000 hidden units; Figure A.5b compares non-OOD and OOD predictions obtained by the BNNs with ReLU activation, Student-t priors and $D_1 = 200$. The poor performance of the mfVIS, especially for wider networks, is not surprising, and the performance of Student-t priors is very similar to Gaussian priors. However, we notice that for wide networks, HMCS with Gaussian priors (see Figure 3.4b) suffers from much higher RMSE than HMCS with Student-t priors and mfVIR and HMCR with either choice of priors.

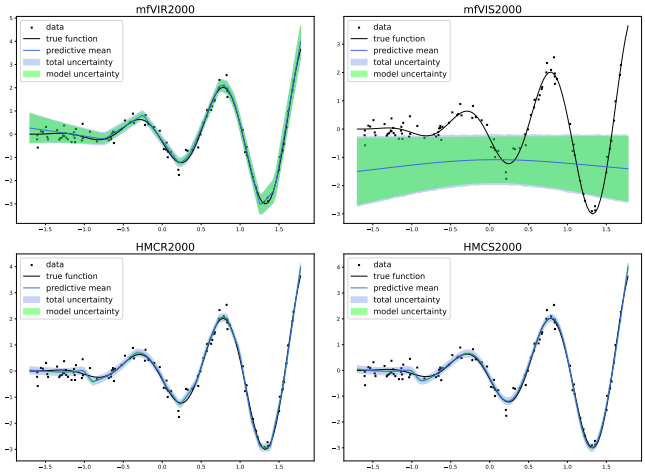
Finally, we recreate the experiments of Section 3.3.4, where we compare three model averaging methodologies, deep ensembles of Bayesian neural networks, stacking and pseudo-BMA based on PSIS-LOO. Consider the mfVIR20 model with Student-t priors and the 'complement-distributions' and 'related-distributions' data tasks. We choose 10 random initialization points, obtain 10 posterior predictive distributions and compute estimated expected log pointwise predictive densities. We then construct ensemble, pseudo-BMA and stacking approximations; the results are illustrated by Figure A.6 and are, again, very similar to Gaussian priors. In both of the tasks, ensembling and stacking are superior to pseudo-BMA, which has worse accuracy and fails to capture any uncertainty.

Future work could study the performance of various more elaborate than Gaussian or Student-t choices of priors placed on the weights, including sparsity-

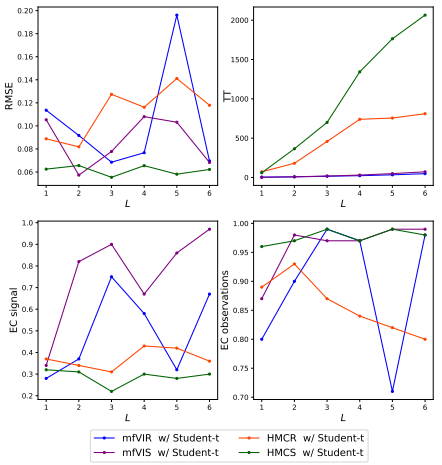
inducing priors which have been shown to improve the accuracy and calibration [Blundell et al., 2015, Polson and Ročková, 2018].



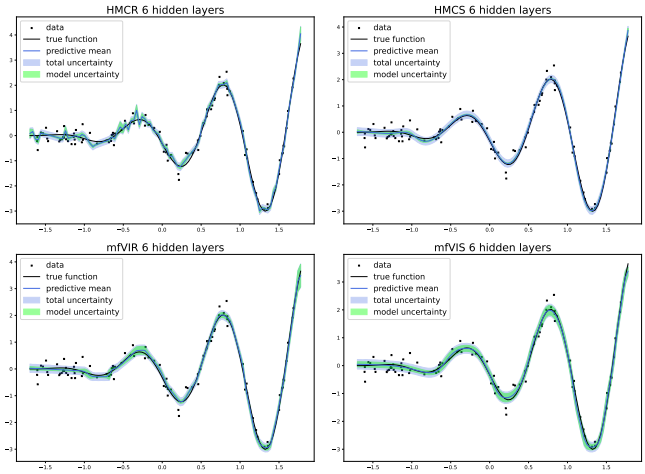
(a) The prediction performances of all the models are compared as the number of hidden units increases.



(b) The predictions and uncertainty estimates obtained by each model when $D_1 = 2000$

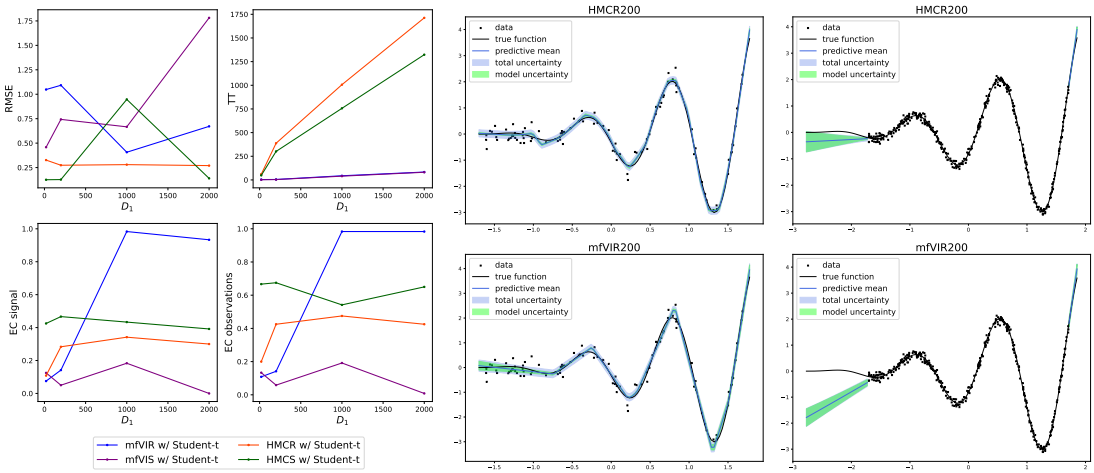


(c) The prediction performances of all the models are compared as the number of hidden layers increases.



(d) The predictions and uncertainty estimates obtained by each model when $L = 6$

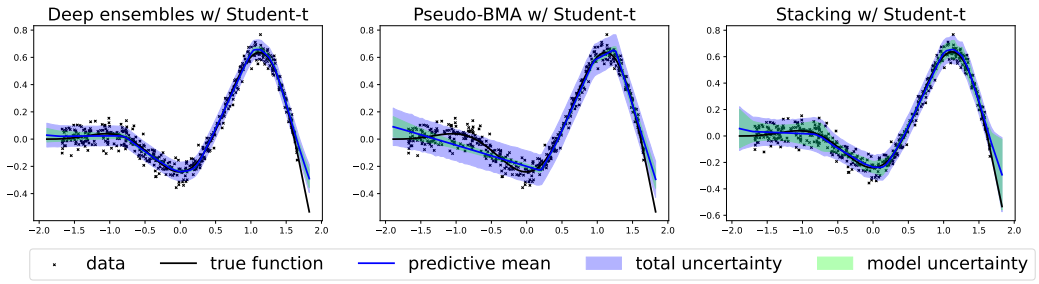
Figure A.4: Prediction performance of wider and deeper neural networks with Student-t priors.



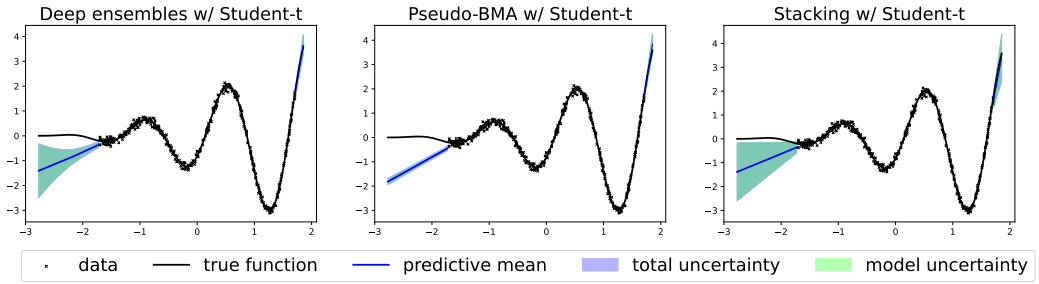
(a) Out-of-sample predictive performance of all the models compared.

(b) Within-the sample and out-of-sample predictions and uncertainty estimates of BNNs with $D_1 = 200$ and ReLU activation.

Figure A.5: Out-of-distribution prediction for the complement-distribution data in the case of BNN with Student-t priors.



(a) The complement-distributions task. The pseudo-BMA is worse than DE and stacking, which are very similar to each other.



(b) The related-distributions task. The pseudo-BMA is again worse than the other methodologies. In uncertainty quantification, stacking is better than DE.

Figure A.6: Predictions obtained by ensembling, stacking and pseudo-BMA when applied to mfVIR20 with Student-t priors in the complement-distributions and related-distributions tasks.

Appendix B

Supplementary to the Variational Bow Tie Neural Network

This appendix supplements the Chapter 4, where we developed a variational bow tie neural network.

B.1 Derivations of the variational posterior

Global shrinkage parameters. Using Equation (2.8), the variational posterior for the global shrinkage parameters is:

$$
\begin{aligned}
q(\boldsymbol{\tau}) &\propto \exp\left(\mathbb{E}\left[\log \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \mathcal{N}\left(W_{l,d,d'}|0,\tau_l\psi_{l,d,d'}\right)\right] + \log \prod_{l}^{L+1} \mathrm{GIG}\left(\tau_l|\nu_{\text{glob}},\delta_{\text{glob}},\lambda_{\text{glob}}\right)\right) \\
&\propto \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \exp \mathbb{E}\left[\log\left(\frac{1}{\sqrt{\tau_l\psi_{l,d,d'}}}\exp\left(-\frac{W_{l,d,d'}^2}{2\tau_l\psi_{l,d,d'}}\right)\right)\right] \\
&\times \prod_{l}^{L+1} \tau_l^{\nu_{\text{glob}}-1}\exp\left(-\frac{1}{2}\left(\frac{\delta_{\text{glob}}^2}{\tau_l}+\lambda_{\text{glob}}^2\tau_l\right)\right) \\
&\propto \prod_{l}^{L+1} \tau_l^{\nu_{\text{glob}}-1}\exp\left(-\frac{\delta_{\text{glob}}^2}{2\tau_l}-\frac{\lambda_{\text{glob}}^2\tau_l}{2}\right)\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}}\tau_l^{-\frac{1}{2}}\exp\left(-\frac{\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\mathbb{E}\left[W_{l,d,d'}^2\right]}{2\tau_l}\right) \\
&\propto \prod_{l}^{L+1} \tau_l^{\nu_{\text{glob}}-\frac{D_lD_{l-1}}{2}-1}\exp\left(-\frac{1}{2}\left(\frac{1}{\tau_l}\left(\sum_{d}^{D_l}\sum_{d'}^{D_{l-1}}\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\mathbb{E}\left[W_{l,d,d'}^2\right]+\delta_{\text{glob}}^2\right)+\lambda_{\text{glob}}^2\tau_l\right)\right) \\
&\propto \prod_{l}^{L+1} \mathrm{GIG}\left(\tau_l \mid \hat{\nu}_{\text{glob},l},\hat{\delta}_{\text{glob},l},\lambda_{\text{glob}}\right),
\end{aligned}
$$

where for $l = 1,\ldots,L+1$

$$
\begin{aligned}
\hat{\nu}_{\text{glob},l} &= \nu_{\text{glob}} - \frac{D_lD_{l-1}}{2}, \\
\hat{\delta}_{\text{glob},l} &= \sqrt{\delta_{\text{glob}}^2 + \sum_{d}^{D_l}\sum_{d'}^{D_{l-1}}\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\mathbb{E}\left[w_{l,d,d'}^2\right]}.
\end{aligned}
$$

Assuming hidden layers of the dimension D, the computational complexity of updating the global shrinkage variable is $\mathcal{O}(LD\max(D_0, D, D_{L+1}))$.

Local shrinkage parameters. Similarly, the variational posterior for the local shrinkage parameters is:

$$
\begin{aligned}
q(\boldsymbol{\psi}) &\propto \prod_{l}^{L+1} \exp\left(\mathbb{E}\left[\log \prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \mathcal{N}\left(W_{l,d,d'}|0, \tau_l\psi_{l,d,d'}\right)\right]\right) \\
&\times \prod_{l}^{L+1} \exp\left(\log \prod_{d=1}^{D_l}\prod_{d'}^{D_{l-1}} \mathrm{GIG}\left(\psi_{l,d,d'}|\nu_{\mathrm{loc},l}, \delta_{\mathrm{loc},l}, \lambda_{\mathrm{loc},l}\right)\right) \\
&\propto \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \exp\left(\frac{1}{2}\log\psi_{l,d,d'} - \frac{\mathbb{E}\left[\frac{1}{\tau_l}\right]\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\mathbb{E}\left[W_{l,d,d'}^2\right]}{2}\right) \\
&\times \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \psi_{l,d,d'}^{\nu_{\mathrm{loc},l}-1} \exp\left(-\frac{1}{2}\left(\frac{\delta_{\mathrm{loc},l}^2}{\psi_{l,d,d'}} + \lambda_{\mathrm{loc},l}^2\psi_{l,d,d'}\right)\right) \\
&\propto \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \psi_{l,d,d'}^{\nu_{\mathrm{loc},l}-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\frac{1}{\psi_{l,d,d'}}\left(\mathbb{E}\left[\frac{1}{\tau_l}\right]\mathbb{E}\left[W_{l,d,d'}^2\right] + \delta_{\mathrm{loc},l}^2\right) + \lambda_{\mathrm{loc},l}^2\psi_{l,d,d'}\right)\right) \\
&\propto \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \mathrm{GIG}\left(\psi_{l,d,d'} \mid \hat{\nu}_{\mathrm{loc},l,d,d'}, \hat{\delta}_{\mathrm{loc},l,d,d'}, \lambda_{\mathrm{loc},l}\right),
\end{aligned}
$$

where for $l = 1, \ldots, L+1,\ d = 1, \ldots, D_l, D_{l-1}\ d' = 1, \ldots, D_{l-1}$

$$
\begin{aligned}
\hat{\nu}_{\mathrm{loc},l,d,d'} &= \nu_{\mathrm{loc},l} - \frac{1}{2}, \\
\hat{\delta}_{\mathrm{loc},l,d,d'} &= \sqrt{\mathbb{E}\left[\frac{1}{\tau_l}\right]\mathbb{E}\left[W_{l,d,d'}^2\right] + \delta_{\mathrm{loc},l}^2}.
\end{aligned}
$$

Similarly, given hidden layers of the dimension D, the computational complexity of updating the global shrinkage variable is $\mathcal{O}(LD\max(D_0, D, D_{L+1}))$.

Covariance matrix. Under the assumption of a diagonal covariance matrix, with parameters $\boldsymbol{\eta}_l = (\eta_{l,1}^2, \ldots \eta_{l,D_l}^2)$, the variational posterior is:

$$
\begin{aligned}
q(\boldsymbol{\eta}) &\propto \exp\left(\mathbb{E}\left[\log\prod_{n}^{N}\mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{z}_{n,L+1}, \boldsymbol{\Sigma}_{L+1}\right) + \log\prod_{n}^{N}\prod_{l}^{L}\mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{\gamma}_{n,l}\odot\boldsymbol{z}_{n,l}, \boldsymbol{\Sigma}_l\right)\right]\right) \\
&\times \prod_{l}^{L}\prod_{d}^{D_l}\mathrm{IG}(\eta_{l,d}^2|\alpha_0^h, \beta_0^h)\prod_{d}^{D_{L+1}}\mathrm{IG}(\eta_{l,d}^2|\alpha_0, \beta_0) \\
&\propto \exp\left(-\frac{1}{2}\mathbb{E}\left[\sum_{n}^{N}\sum_{d}^{D_{L+1}}\left(\eta_{L+1,d}\right)^{-2}\left(y_{n,d} - z_{n,L+1,d}\right)^2\right]\right) \\
&\times \prod_{d}^{D_{L+1}}\left(\left(\eta_{L+1,d}^2\right)^{-\alpha_0-1-\frac{N}{2}}\exp\left(-\frac{\beta_0}{\eta_{L+1,d}^2}\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
&\times \prod_{l}^{L} \exp\left(-\frac{1}{2}\mathbb{E}\left[\sum_{n}^{N}\sum_{d}^{D_l}(\eta_{l,d})^{-2}\left(\mathrm{a}_{n,l,d}-\gamma_{n,l,d}\odot z_{n,l,d}\right)^2\right]\right)\\
&\times \prod_{l}^{L}\prod_{d}^{D_l}\left((\eta_{l,d}^2)^{-\alpha_0^h-1-\frac{N}{2}}\exp\left(-\frac{\beta_0^h}{\eta_{l,d}^2}\right)\right)\\
&\propto \prod_{d}^{D_{L+1}}\left((\eta_{L+1,d}^2)^{-\alpha_0-1-\frac{N}{2}}\right)\exp\left(-\frac{1}{\eta_{L+1,d}^2}\left(\beta_0+\frac{1}{2}\sum_{n}^{N}\mathbb{E}\left[\left(y_{n,d}-z_{n,L+1,d}\right)^2\right]\right)\right)\\
&\times \prod_{l}^{L}\prod_{d}^{D_l}(\eta_{l,d}^2)^{-\alpha_0^h-1-\frac{N}{2}}\exp\left(-\frac{1}{\eta_{l,d}^2}\left(\beta_0^h+\frac{1}{2}\sum_{n}^{N}\mathbb{E}\left[\left(\mathrm{a}_{n,l,d}-\gamma_{n,l,d}\odot z_{n,l,d}\right)^2\right]\right)\right).
\end{aligned}
$$

Thus, $q(\boldsymbol{\eta}) \propto \prod_{l}^{L+1}\prod_{d}^{D_l}\mathrm{IG}(\alpha_{l,d},\beta_{l,d})$, where

$$
\begin{aligned}
\alpha_{l,d} &= \alpha_0^h+\frac{N}{2}, \quad d=1,\ldots,D_l, \quad l=1,\ldots,L,\\
\alpha_{L+1,d} &= \alpha_0+\frac{N}{2}, \quad d=1,\ldots,D_{L+1},\\
\beta_{l,d} &= \beta_0^h+\frac{1}{2}\sum_{n}^{N}\mathbb{E}\left[\left(\mathrm{a}_{n,l,d}-\gamma_{n,l,d}\odot z_{n,l,d}\right)^2\right], \quad d=1,\ldots,D_l, \quad l=1,\ldots,L,\\
\beta_{L+1,d} &= \beta_0+\frac{1}{2}\sum_{n}^{N}\mathbb{E}\left[\left(y_{n,d}-z_{n,L+1,d}\right)^2\right], \quad d=1,\ldots,D_{L+1}.
\end{aligned}
$$

For the parameters $\beta_{l,d}$, we must compute the sum of squares terms. For the last layer $l=L+1$, this term, for each data point n, is given by:

$$
\begin{aligned}
&\mathbb{E}\left[\left(y_{n,d}-z_{n,L+1,d}\right)^2\right] = \sum_{n}^{N}\left(y_{n,d}-\mathbb{E}\left[\boldsymbol{W}_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}\right]-\mathbb{E}\left[b_{L+1,d}\right]\right)^2-\mathbb{E}\left[b_{L+1,d}\right]^2\\
&+\sum_{n}^{N}\mathbb{E}\left[b_{L+1,d}^2\right]+2\mathbb{E}\left[b_{L+1,d}\boldsymbol{W}_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}\right]-2\mathbb{E}\left[b_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{W}_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}\right]\\
&+2\sum_{n}^{N}\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\boldsymbol{W}_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}\boldsymbol{a}_{n,L}^T\right]\right)\\
&-2\sum_{n}^{N}\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\right]\mathbb{E}\left[\boldsymbol{W}_{L+1,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}\right]\mathbb{E}\left[\boldsymbol{a}_{n,L}^T\right]\right).
\end{aligned}
$$

Instead, for an intermediate layer $l=1,\ldots,L$, the sum of squares term, for each data point n, is given by:

$$
\begin{aligned}
&\mathbb{E}\left[\left(\mathrm{a}_{n,l,d}-\gamma_{n,l,d}\cdot z_{n,l,d}\right)^2\right] = \sum_{n}^{N}\left(\mathbb{E}\left[\mathrm{a}_{n,l,d}\right]-\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[b_{l,d}\right]-\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,l-1}\right]\right)^2\\
&+\sum_{n}^{N}\mathbb{E}\left[a_{n,l,d}^2\right]-\mathbb{E}\left[\mathrm{a}_{n,l,d}\right]^2+\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[b_{l,d}^2\right]-\mathbb{E}\left[\gamma_{n,l,d}\right]^2\mathbb{E}\left[b_{l,d}\right]^2\\
&+\sum_{n}^{N}\mathbb{E}\left[\gamma_{n,l,d}\right]\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\boldsymbol{W}_{l,d}\right]\mathbb{E}\left[\boldsymbol{a}_{n,l-1}\boldsymbol{a}_{n,l-1}^T\right]\right)\\
&-\sum_{n}^{N}\mathbb{E}\left[\gamma_{n,l,d}\right]^2\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}\right]\left[\boldsymbol{a}_{n,l-1}\boldsymbol{a}_{n,l-1}^T\right]\right)
\end{aligned}
$$

$$+ 2\sum_{n}^{N} \mathbb{E}\left[\gamma_{n,l,d}\right] \mathbb{E}\left[b_{l,d}\boldsymbol{W}_{l,d}\right]\left[\boldsymbol{a}_{n,l-1}\right] - \mathbb{E}\left[\gamma_{n,l,d}\right]^2 \mathbb{E}\left[b_{l,d}\right] \mathbb{E}\left[\boldsymbol{W}_{l,d}\right]\left[\boldsymbol{a}_{n,l-1}\right].$$

The complexity of obtaining variational update for $\boldsymbol{\eta}$ is then $\mathcal{O}(NL\max(D, D_0)^2 \max(D_{L+1}, D))$.

Weights and biases. The variational posterior for the weights and biases is:

$$\begin{aligned}
q(\boldsymbol{b}, \boldsymbol{W}) &\propto \exp\left(\mathbb{E}\left[\log \prod_{n}^{N} \mathcal{N}\left(y_n | \boldsymbol{W}_{L+1}\boldsymbol{a}_{n,L} + \boldsymbol{b}_{L+1}, \boldsymbol{\Sigma}_{L+1}\right)\right]\right) \\
&\times \exp\left(\mathbb{E}\left[\log \prod_{n}^{N}\prod_{l}^{L} \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{\gamma}_{n,l} \odot (\boldsymbol{W}_l \boldsymbol{a}_{n,l-1} + \boldsymbol{b}_l), \boldsymbol{\Sigma}_l\right)\right]\right) \\
&\times \exp\left(\mathbb{E}\left[\log \prod_{n}^{N}\prod_{l}^{L}\prod_{d}^{D_l} \exp\left(\frac{(\gamma_{n,l,d} - \frac{1}{2}) z_{n,l,d}}{T}\right) \exp\left(-\frac{\omega_{n,l,d} z_{n,l,d}^2}{2T^2}\right)\right]\right) \\
&\times \exp\left(\mathbb{E}\left[\log \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \mathcal{N}\left(W_{l,d,d'} | 0, \tau_l \psi_{l,d,d'}\right)\right]\right) \prod_{l}^{L+1}\prod_{d}^{D_l} \mathcal{N}(b_{l,d} | 0, s_0^2) \\
&\propto \prod_{n}^{N} \exp\left(\mathbb{E}\left[\log \frac{1}{\sqrt{|\boldsymbol{\Sigma}_{L+1}|}}\right]\right) \times \prod_{l}^{L}\prod_{n}^{N} \exp\left(\mathbb{E}\left[\log \frac{1}{\sqrt{|\boldsymbol{\Sigma}_l|}}\right]\right) \\
&\times \prod_{n}^{N} \exp\left(\mathbb{E}\left[-\frac{1}{2}\left(\boldsymbol{y}_n - \boldsymbol{W}_{L+1}\boldsymbol{a}_{n,L} - \boldsymbol{b}_{L+1}\right)^T \left(\boldsymbol{\Sigma}_{L+1}\right)^{-1} \left(\boldsymbol{y}_n - \boldsymbol{W}_{L+1}\boldsymbol{a}_{n,L} - \boldsymbol{b}_{L+1}\right)\right]\right) \\
&\times \prod_{l}^{L}\prod_{n}^{N} \exp\left(\mathbb{E}\left[-\frac{1}{2}\left(\boldsymbol{a}_{n,l} - \boldsymbol{\gamma}_{n,l}\boldsymbol{W}_l\boldsymbol{a}_{n,l-1} - \boldsymbol{\gamma}_{n,l}\boldsymbol{b}_l\right)^T \left(\boldsymbol{\Sigma}_l\right)^{-1} \left(\boldsymbol{a}_{n,l} - \boldsymbol{\gamma}_{n,l}\boldsymbol{W}_l\boldsymbol{a}_{n,l-1} - \boldsymbol{\gamma}_{n,l}\boldsymbol{b}_l\right)\right)\right] \\
&\times \prod_{l}^{L}\prod_{n}^{N}\prod_{d}^{D_l} \exp\left(\mathbb{E}\left[\frac{(\gamma_{n,l,d} - \frac{1}{2})\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)}{T}\right]\right) \\
&\times \prod_{l}^{L}\prod_{n}^{N}\prod_{d}^{D_l} \exp\left(\mathbb{E}\left[-\frac{\omega_{n,l,d}\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)^2}{2T^2}\right]\right) \\
&\times \prod_{l}^{L+1}\prod_{d}^{D_l}\prod_{d'}^{D_{l-1}} \exp\left(-\frac{W_{l,d,d'}^2}{2}\mathbb{E}\left[\frac{1}{\tau_l}\right]\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\right) \prod_{l}^{L+1}\prod_{d}^{D_l} \exp\left(-\frac{b_{l,d}^2}{2s_0^2}\right).
\end{aligned}$$

Therefore, using also the fact that $\boldsymbol{\Sigma}_l$ is diagonal, we have that the variational posterior factorizes as $q(\boldsymbol{b}, \boldsymbol{W}) = \prod_{l}^{L+1}\prod_{d=1}^{D_l} q(b_{l,d}, \boldsymbol{W}_{l,d})$. We consider the terms $q(b_{l,d}, \boldsymbol{W}_{l,d})$ for the intermediate layers $l = 1, \ldots, L$ and $q(b_{L+1,d}, \boldsymbol{W}_{L+1,d})$ for the last layer separately.

Starting with the last layer $L+1$, we first introduce the matrix

$$\boldsymbol{D}_{L+1,d}^{-1} = \operatorname{diag}\left(s_0^{-2}, \mathbb{E}\left[\tau_{L+1}^{-1}\right]\mathbb{E}\left[\psi_{L+1,d,1}^{-1}\right], \ldots, \mathbb{E}\left[\tau_{L+1}^{-1}\right]\mathbb{E}\left[\psi_{L+1,d,D_L}^{-1}\right]\right).$$

Then, for the variational posterior of the weights and biases for the dth dimension of the final layer, we only need to consider the relevant terms:

$$q(b_{L+1,d}, \boldsymbol{W}_{L+1,d}) \propto \exp\left(-\frac{1}{2}\widetilde{\boldsymbol{W}}_{L+1,d}\boldsymbol{D}_{L+1,d}^{-1}\widetilde{\boldsymbol{W}}_{L+1,d}^T\right)$$

$$
\begin{aligned}
&\times \exp\left(-\frac{1}{2}\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\sum_{n}^{N}\mathbb{E}\left[\left(y_{n,d}-\widetilde{\boldsymbol{W}}_{L+1,d}\widetilde{\boldsymbol{a}}_{n,L}\right)^{2}\right]\right)\\
&\propto \exp\left(-\frac{1}{2}\widetilde{\boldsymbol{W}}_{L+1,d}\boldsymbol{D}_{L+1,d}^{-1}\widetilde{\boldsymbol{W}}_{L+1,d}^{T}\right)\\
&\times \exp\left(-\frac{1}{2}\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\left(\widetilde{\boldsymbol{W}}_{L+1,d}\left(\sum_{n}^{N}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}_{n,L}^{T}\right]\right)\widetilde{\boldsymbol{W}}_{L+1,d}^{T}\right)\right)\\
&\times \exp\left(\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\widetilde{\boldsymbol{W}}_{L+1,d}\left(\sum_{n}^{N}y_{n}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right]\right)\right)\\
&\propto \exp\left(-\frac{1}{2}\left(\widetilde{\boldsymbol{W}}_{L+1,d}\left(\boldsymbol{D}_{L+1,d}^{-1}+\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\sum_{n}^{N}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}_{n,L}^{T}\right]\right)\widetilde{\boldsymbol{W}}_{L+1,d}^{T}\right)\right)\\
&\times \exp\left(\left(\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\widetilde{\boldsymbol{W}}_{L+1,d}\left(\sum_{n}^{N}y_{n}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right]\right)\right)\right)\\
&\propto \exp\left(-\frac{1}{2}\left(\widetilde{\boldsymbol{W}}_{L+1,d}\boldsymbol{B}_{L+1,d}^{-1}\widetilde{\boldsymbol{W}}_{L+1,d}^{T}-2\widetilde{\boldsymbol{W}}_{L+1,d}\boldsymbol{B}_{L+1,d}^{-1}\boldsymbol{m}_{L+1,d}^{T}\right)\right),
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{B}_{L+1,d}^{-1}=&\boldsymbol{D}_{L+1,d}^{-1}+\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\sum_{n}^{N}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}_{n,L}^{T}\right],\\
\boldsymbol{m}_{L+1,d}^{T}=&\boldsymbol{B}_{L+1,d}\mathbb{E}\left[(\eta_{L+1,d})^{-2}\right]\left(\sum_{n}^{N}y_{n}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right]\right).
\end{aligned}
$$

Thus, completing the square, we have that

$$
q(b_{L+1,d},\boldsymbol{W}_{L+1,d})=\mathcal{N}\left(\widetilde{\boldsymbol{W}}_{L+1,d}|\boldsymbol{m}_{L+1,d},\boldsymbol{B}_{L+1,d}\right).
$$

Next, for the intermediate layers $l=1,\ldots,L$, we can similarly obtain the variational posterior of the weights and biases $q(b_{l,d},\boldsymbol{W}_{l,d})$ for dimensions $d=1,\ldots,D_l$. We introduce the matrices

$$
\boldsymbol{D}_{l,d}^{-1}=\operatorname{diag}\left(s_0^{-2},\mathbb{E}\left[\tau_l^{-1}\right]\mathbb{E}\left[\psi_{l,d,1}^{-1}\right],\ldots,\mathbb{E}\left[\tau_l^{-1}\right]\mathbb{E}\left[\psi_{l,d,D_{l-1}^{-1}}\right]\right),
$$

and consider the terms relevant to derive each $q(b_{l,d},\boldsymbol{W}_{l,d})$ separately:

$$
\begin{aligned}
q(b_{l,d},\boldsymbol{W}_{l,d})\propto&\exp\left(-\frac{1}{2}\widetilde{\boldsymbol{W}}_{l,d}\boldsymbol{D}_{l,d}^{-1}\widetilde{\boldsymbol{W}}_{l,d}^{T}-\frac{1}{2T^{2}}\widetilde{\boldsymbol{W}}_{l,d}\left(\sum_{n}^{N}\mathbb{E}\left[\omega_{n,l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^{T}\right]\right)\widetilde{\boldsymbol{W}}_{l,d}^{T}\right.\\
&-\frac{1}{2}\mathbb{E}\left[\eta_{l,d}^{-2}\right]\widetilde{\boldsymbol{W}}_{l,d}\left(\sum_{n}^{N}\mathbb{E}\left[\gamma_{n,l,d}^{2}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^{T}\right]\right)\widetilde{\boldsymbol{W}}_{l,d}^{T}\\
&+\mathbb{E}\left[\eta_{l,d}^{-2}\right]\widetilde{\boldsymbol{W}}_{l,d}\left(\sum_{n}^{N}\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\mathrm{a}_{n,l,d}\boldsymbol{a}_{n,l-1}\right]\right)\\
&\left.+\frac{1}{T}\widetilde{\boldsymbol{W}}_{l,d}\left(\sum_{n}^{N}\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\right)-\frac{1}{2T}\widetilde{\boldsymbol{W}}_{l,d}\left(\sum_{n}^{N}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\right)\right)
\end{aligned}
$$

$$\propto \exp\left(-\frac{1}{2}\left(\widetilde{\boldsymbol{W}}_{l,d}\boldsymbol{B}_{l,d}^{-1}\widetilde{\boldsymbol{W}}_{l,d}^{T} - 2\widetilde{\boldsymbol{W}}_{l,d}\boldsymbol{B}_{l,d}^{-1}\boldsymbol{m}_{l,d}^{T}\right)\right),$$

where

$$\boldsymbol{B}_{l,d}^{-1} = \boldsymbol{D}_{l,d}^{-1} + \sum_{n}^{N}\left(\left(\frac{1}{T^2}\mathbb{E}\left[\omega_{n,l,d}\right] + \mathbb{E}\left[(\eta_{l,d})^{-2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\right)\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^{T}\right]\right),$$

$$\boldsymbol{m}_{l,d}^{T} = \boldsymbol{B}_{l,d}\left(\sum_{n}^{N}\left(\mathbb{E}\left[(\eta_{l,d})^{-2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\mathrm{a}_{n,l,d}\widetilde{\boldsymbol{a}}_{n,l-1}\right] + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\left(\mathbb{E}\left[\gamma_{n,l,d}\right] - \frac{1}{2}\right)\right)\right).$$

Again, completing the square, we obtain the Gaussian variational posterior

$$q(b_{l,d}, \boldsymbol{W}_{l,d}) = \mathcal{N}\left((b_{l,d}, \boldsymbol{W}_{l,d})|\boldsymbol{m}_{l,d}, \boldsymbol{B}_{l,d}\right).$$

The complexity of obtaining variational update for $\boldsymbol{W}, \boldsymbol{b}$ is then $\mathcal{O}((L\max(D_{L+1}, D)(N\max(D, D_0)^2 + \max(D, D_0)^3))$. Assuming $\max(D, D_0) < N$, one gets the same complexity as when updating $\boldsymbol{\eta}$, i.e. $\mathcal{O}(LN\max(D, D_0)^2\max(D_{L+1}, D)$.

Augmented variables. The variational posterior of the augmented variables is

$$\begin{aligned} q(\boldsymbol{\omega}) &\propto \exp\left(\mathbb{E}\left[\log\prod_{n}^{N}\prod_{l}^{L}\prod_{d}^{D_l}\exp\left(-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right)p(\omega_{n,l,d})\right]\right) \\ &\propto \prod_{n}^{N}\prod_{l}^{L}\prod_{d}^{D_l}\exp\left(\mathbb{E}\left[-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right]\right)p(\omega_{n,l,d}). \end{aligned}$$

Thus, they are independent across width, depth, and observations, with

$$\begin{aligned} q(\omega_{n,l,d}) &= \mathrm{PG}(\omega_{n,l,d}|1, \frac{1}{T}\sqrt{\mathbb{E}\left[z_{n,l,d}^2\right]}) \\ &= \mathrm{PG}(\omega_{n,l,d}|1, \mathrm{a}_{n,l,d}), \end{aligned}$$

where

$$\mathrm{a}_{n,l,d} = \frac{1}{T}\sqrt{\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^{T}\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^{T}\right]\right)\right)}.$$

The complexity of obtaining variational update for $\boldsymbol{\omega}$ is then $\mathcal{O}((NLD\max(D, D_0)^2))$.

Binary activation. The variational posterior of the binary activations is:

$$\begin{aligned} q(\boldsymbol{\gamma}) &\propto \exp\left(\mathbb{E}\left[\log\prod_{n}^{N}\prod_{l}^{L}\mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{\gamma}_{n,l}\odot\boldsymbol{z}_{n,l}, \boldsymbol{\Sigma}_l\right)\right]\right) \\ &\times \exp\left(\mathbb{E}\left[\log\left(\prod_{n}^{N}\prod_{l}^{L}\prod_{d}^{D_l}\exp(\frac{\gamma_{n,l,d}z_{n,l,d}}{T})\right)\right]\right) \end{aligned}$$

$$\propto \prod_n^N \prod_l^L \prod_l^{D_l} \exp\left(-\frac{1}{2\eta_{l,d}^2}\mathbb{E}\left[(\mathrm{a}_{n,l,d} - \gamma_{n,l,d}\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right))^2\right]\right)$$
$$\times \prod_n^N \prod_l^L \prod_l^{D_l} \exp\left(-\frac{1}{2\eta_{l,d}^2}\mathbb{E}\left[\frac{\gamma_{n,l,d}(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d})}{T}\right]\right).$$

Therefore, the variational posterior $q(\boldsymbol{\gamma})$ factories across observations $n = 1, \ldots, N$, layers $l = 1, \ldots, L$, and dimensions of the layer $d = 1, \ldots, D_l$, with each factor $q(\gamma_{n,l,d})$ given by:

$$\begin{aligned} q(\gamma_{n,l,d}) \propto & \exp\left(\gamma_{n,l,d}\left(-\frac{1}{2}\mathbb{E}\left[\eta_{l,d}^{-2}\right]\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right)\right.\right. \\ & \left.\left. + \mathbb{E}\left[\eta_{l,d}^{-2}\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\mathrm{a}_{n,l,d}\right] + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\right)\right) \\ \propto & \exp\left(\gamma_{n,l,d}\sigma^{-1}\left(\rho_{n,l,d}\right)\right), \end{aligned}$$

where σ is the logistic function and

$$\begin{aligned} \rho_{n,l,d} = \sigma\left(\mathbb{E}\left[\eta_{l,d}^{-2}\right]\left(-\frac{1}{2}\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right) + \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\mathrm{a}_{n,l,d}\right]\right) \\ + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\right). \end{aligned}$$

Then, noticing that $\sigma^{-1}(\rho) = \log(\rho(1-\rho)^{-1})$ and combining separate factors of the variational posterior of the binary activations, we obtain:

$$\begin{aligned} q(\boldsymbol{\gamma}) &\propto \prod_n^N \prod_l^L \prod_d^{D_l} \rho_{n,l,d}^{\gamma_{n,l,d}}\left(1-\rho_{n,l,d}\right)^{1-\gamma_{n,l,d}} \\ &\propto \prod_n^N \prod_l^L \prod_d^{D_l} \mathrm{Bern}\left(\gamma_{n,l,d}|\rho_{n,l,d}\right). \end{aligned}$$

The complexity of obtaining variational update for $\boldsymbol{\gamma}$ is then $\mathcal{O}(LND\max(D, D_0)^2)$.

Stochastic activation. The variational posterior of the stochastic activation is

$$\begin{aligned} q(\boldsymbol{a}) \propto & \exp\left(\mathbb{E}\left[\log\prod_n^N \mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{z}_{n,L+1}, \boldsymbol{\eta}_{L+1}\right) + \log\prod_n^N\prod_l^L \mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{\gamma}_{n,l}\odot\boldsymbol{z}_{n,l}, \boldsymbol{\eta}_l\right)\right.\right. \\ & \left.\left. + \log\prod_n^N\prod_l^L\prod_d^{D_l}\exp\left(\frac{(\gamma_{n,l,d}-\frac{1}{2})z_{n,l,d}}{T}\right)\exp\left(-\frac{\omega_{n,l,d}z_{n,l,d}^2}{2T^2}\right)\right]\right) \\ \propto & \prod_n^N \exp\left(-\frac{1}{2}\sum_d^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\mathbb{E}\left[\left(y_{n,d} - \boldsymbol{W}_{L+1,d}\boldsymbol{a}_{n,L} - b_{L+1,d}\right)^2\right]\right) \\ & \times \prod_n^N \exp\left(-\frac{1}{2}\sum_l^L\sum_d^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\mathbb{E}\left[\left(\mathrm{a}_{n,l,d} - \gamma_{n,l,d}\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)\right)^2\right]\right) \end{aligned}$$

$$\times \prod_{n}^{N} \exp\left(\sum_{l}^{L}\sum_{d}^{D_l} \mathbb{E}\left[\frac{\left(\gamma_{n,l,d}-\frac{1}{2}\right) z_{n,l,d}}{T} - \frac{\omega_{n,l,d} z_{n,l,d}^2}{2T^2}\right]\right).$$

Therefore, the variational posterior of the stochastic activations factories across observations $n = 1, \ldots, N$, and we derive $q(\boldsymbol{a}_n)$ separately. For each layer $l = 1, \ldots, L$, we introduce the following diagonal matrix $\hat{\boldsymbol{\Sigma}}_l^{-1} = \text{diag}\left(\mathbb{E}\left[\eta_{l,1}^{-2}\right], \ldots, \mathbb{E}\left[\eta_{l,D_l}^{-2}\right]\right)$ and consider the relevant terms of the variational posterior:

$$\begin{aligned}
q(\boldsymbol{a}_n) &\propto \exp\left(-\frac{1}{2}\boldsymbol{a}_{n,L}^T\left(\sum_{d}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\boldsymbol{W}_{L+1,d}\right]\boldsymbol{a}_{n,L}\right)\right)\\
&\times \exp\left(-\boldsymbol{a}_{n,L}^T\left(\sum_{d}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\left(\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T b_{L+1,d}\right] - \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\right] y_{n,d}\right)\right)\right)\\
&\times \exp\left(-\frac{1}{2}\boldsymbol{a}_{n,L}^T\hat{\boldsymbol{\Sigma}}_L^{-1}\boldsymbol{a}_{n,L}\right) \times \prod_{l=1}^{L-1}\exp\left(-\frac{1}{2}\boldsymbol{a}_{n,l}^T\hat{\boldsymbol{\Sigma}}_l^{-1}\boldsymbol{a}_{n,l}\right)\\
&\times \exp\left(\boldsymbol{a}_{n,L}^T\hat{\boldsymbol{\Sigma}}_L^{-1}\left(\left(\mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right]\boldsymbol{1}_{D_{L-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_L\right]\right)\boldsymbol{a}_{n,L-1} + \mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right]\odot\mathbb{E}\left[\boldsymbol{b}_L\right]\right)\right)\\
&\times \prod_{l=1}^{L-1}\exp\left(\boldsymbol{a}_{n,l}^T\hat{\boldsymbol{\Sigma}}_l^{-1}\left(\left(\mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right]\boldsymbol{1}_{D_{l-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_l\right]\right)\boldsymbol{a}_{n,l-1} + \mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right]\odot\mathbb{E}\left[\boldsymbol{b}_l\right]\right)\right)\\
&\times \prod_{l=1}^{L}\exp\left(-\frac{1}{2}\left(\boldsymbol{a}_{n,l-1}^T\left(\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\boldsymbol{W}_{l,d}\right]\boldsymbol{a}_{n,l-1}\right)\right)\right)\times\\
&\times \prod_{l=1}^{L}\exp\left(-\boldsymbol{a}_{n,l-1}^T\left(\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\boldsymbol{b}_{l,d}\right]\right)\right)\\
&\times \prod_{l=1}^{L}\exp\left(-\frac{1}{2}\left(\boldsymbol{a}_{n,l-1}^T\left(\frac{1}{T^2}\sum_{d=1}^{D_l}\mathbb{E}\left[\omega_{n,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\boldsymbol{W}_{l,d}\right]\right)\boldsymbol{a}_{n,l-1}\right)\right)\\
&\times \prod_{l=1}^{L}\exp\left(\boldsymbol{a}_{n,l-1}^T\left(\frac{1}{T}\sum_{d=1}^{D_l}\mathbb{E}\left[\boldsymbol{W}_{l,d}^T\right]\left(\mathbb{E}\left[\gamma_{n,l,d}\right]-\frac{1}{2}\right)\right)\right)\\
&\times \prod_{l=1}^{L}\exp\left(-\boldsymbol{a}_{n,l-1}^T\left(\frac{1}{T^2}\sum_{d=1}^{D_l}\mathbb{E}\left[\omega_{n,l,d}\right]\mathbb{E}\left[\boldsymbol{W}_{l,d}^T b_{l,d}\right]\right)\right).
\end{aligned}$$

The variational posterior of the stochastic activations does not factories into independent blocks, however, it does have a structured sequential factorization $q(\boldsymbol{a}_n) = \prod_{l=1}^{L} q(\boldsymbol{a}_{n,l}|\boldsymbol{a}_{n,l-1})$.And, we can derive the variational factor $q(\boldsymbol{a}_{n,L}|\boldsymbol{a}_{n,L-1})$ by only considering the terms with $\boldsymbol{a}_{n,L}$. First, introduce the matrices $\boldsymbol{S}_{n,L}$ and $\boldsymbol{M}_{n,L}$ and a vectors $\boldsymbol{t}_{n,L}$:

$$\begin{aligned}
\boldsymbol{S}_{n,L}^{-1} &= \hat{\boldsymbol{\Sigma}}_L^{-1} + \sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\boldsymbol{W}_{L+1,d}\right],\\
\boldsymbol{t}_{n,L} &= \boldsymbol{S}_{n,L}\left(\left(\sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right] - \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T b_{L+1,d}\right] + \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\right] y_{n,d}\right)\right)\\
&+ \boldsymbol{S}_{n,L}\left(\hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right]\odot\mathbb{E}\left[\boldsymbol{b}_L\right]\right),
\end{aligned}$$

$$\boldsymbol{M}_{n,L} = \boldsymbol{S}_{n,L}\hat{\boldsymbol{\Sigma}}_L^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right]\mathbf{1}_{D_{L-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_L\right].$$

Then we consider the relevant terms of the variational posterior:

$$\begin{aligned}
q(\boldsymbol{a}_{n,L}|\boldsymbol{a}_{n,L-1}) &\propto \exp\left(-\frac{1}{2}\left(\boldsymbol{a}_{n,L}^T\boldsymbol{S}_{n,L}^{-1}\boldsymbol{a}_{n,L} - 2\boldsymbol{a}_{n,L}^T\boldsymbol{S}_{n,L}^{-1}\left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\boldsymbol{a}_{n,L} - \left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)\right)^T\boldsymbol{S}_{n,L}^{-1}\left(\boldsymbol{a}_{n,L} - \left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)\right)\right) \\
&\times \exp\left(\frac{1}{2}\left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)^T\boldsymbol{S}_{n,L}^{-1}\left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)\right) \\
&\propto \mathcal{N}\left(\boldsymbol{a}_{n,L}|\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}, \boldsymbol{S}_{n,L}\right) \\
&\times \exp\left(\frac{1}{2}\left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)^T\boldsymbol{S}_{n,L}^{-1}\left(\boldsymbol{t}_{n,L} + \boldsymbol{M}_{n,L}\boldsymbol{a}_{n,L-1}\right)\right),
\end{aligned}$$

where the first term in the equation above provides $q(\boldsymbol{a}_{n,L}|\boldsymbol{a}_{n,L-1})$ and the second terms is relevant for computing the subsequent $q(\boldsymbol{a}_{n,L-1}|\boldsymbol{a}_{n,L-2})$. Recursively repeating a similar procedure for $l = L-1, \ldots, 1$, we are then able to obtain each of the variational posteriors $q(\boldsymbol{a}_{n,l}|\boldsymbol{a}_{n,l-1})$. Each time we define $\boldsymbol{S}_{n,l}$, $\boldsymbol{M}_{n,l}$ and $\boldsymbol{t}_{n,l}$ as follows:

$$\begin{aligned}
\boldsymbol{S}_{n,l}^{-1} &= \hat{\boldsymbol{\Sigma}}_l^{-1} - \boldsymbol{M}_{n,l+1}^T\boldsymbol{S}_{n,l+1}^{-1}\hat{\boldsymbol{M}}_{n,l+1} \\
&\quad + \sum_{d=1}^{D_{l+1}}\left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right]\mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2}\sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\omega_{n,l+1,d}\right]\right)\mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T\boldsymbol{W}_{l+1,d}\right] \\
\boldsymbol{t}_{n,l} &= \boldsymbol{S}_{n,l}\left(\boldsymbol{M}_{n,l+1}^T\boldsymbol{S}_{n,l+1}^{-1}\boldsymbol{t}_{n,l+1} + \hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right] \odot \mathbb{E}\left[\boldsymbol{b}_l\right]\right. \\
&\quad + \frac{1}{T}\sum_{d=1}^{D_{l+1}}\mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T\right]\left(\mathbb{E}\left[\gamma_{n,l+1,d}\right] - \frac{1}{2}\right) \\
&\quad \left. - \sum_{d=1}^{D_{l+1}}\left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right]\mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2}\mathbb{E}\left[\omega_{n,l+1,d}\right]\right)\mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T b_{l+1,d}\right]\right), \\
\boldsymbol{M}_{n,l} &= \boldsymbol{S}_{n,l}\hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right]\mathbf{1}_{D_{l-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_l\right].
\end{aligned}$$

Then substituting the above into the terms of the variational posterior containing $\boldsymbol{a}_{n,l}$:

$$\begin{aligned}
q(\boldsymbol{a}_{n,l}|\boldsymbol{a}_{n,l-1}) &\propto \exp\left(\frac{1}{2}\left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)^T\boldsymbol{S}_{n,l}^{-1}\left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right) \\
&\times \exp\left(-\frac{1}{2}\left(\boldsymbol{a}_{n,l} - \left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right)^T\boldsymbol{S}_{n,l}^{-1}\left(\boldsymbol{a}_{n,l} - \left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right)\right) \\
&\propto \mathcal{N}\left(\boldsymbol{a}_{n,l}|\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}, \boldsymbol{S}_{n,l}\right) \\
&\times \exp\left(\frac{1}{2}\left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)^T\boldsymbol{S}_{n,l}^{-1}\left(\boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right).
\end{aligned}$$

Finally, we combine the terms $q(\boldsymbol{a}_{n,l}|\boldsymbol{a}_{n,l-1})$ for $l = 1, \ldots, L+1$ and get the variational posterior of the stochastic activation

$$q(\boldsymbol{a}) \propto \prod_{n=1}^{N} \prod_{l=1}^{L} \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l} \boldsymbol{a}_{n,l-1}, \boldsymbol{S}_{n,l}\right).$$

The complexity of obtaining variational update for $\boldsymbol{a}$ is then $\mathcal{O}(NLD^3)$.

B.2 ELBO computation

B.2.1 ELBO for training

Recall that optimal variational parameters maximize the ELBO function of Equation (2.5), which for our model is:

$$\begin{aligned}\text{ELBO} = &\mathbb{E}\left[\log p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega} | \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma})\right] + \mathbb{E}\left[\log p(\boldsymbol{W} | \boldsymbol{\psi}, \boldsymbol{\tau})\right] + \mathbb{E}\left[\log p(\boldsymbol{\psi})\right] + \mathbb{E}\left[\log p(\boldsymbol{\tau})\right] \\ &+ \mathbb{E}\left[\log p(\boldsymbol{b})\right] + \mathbb{E}\left[\log p(\boldsymbol{\Sigma})\right] - \mathbb{E}\left[\log q(\boldsymbol{a})\right] - \mathbb{E}\left[\log q(\boldsymbol{\gamma})\right] - \mathbb{E}\left[\log q(\boldsymbol{\omega})\right] \\ &- \mathbb{E}\left[\log q(\boldsymbol{W}, \boldsymbol{b})\right] - \mathbb{E}\left[\log q(\boldsymbol{\eta})\right] - \mathbb{E}\left[\log q(\boldsymbol{\psi})\right] - \mathbb{E}\left[\log q(\boldsymbol{\tau})\right].\end{aligned}$$

Similar to the variational update, we compute the terms of the ELBO corresponding to different blocks of parameters separately.

ELBO of $\boldsymbol{\tau}$. First, consider the terms of the ELBO containing the global shrinkage parameters:

$$\begin{aligned}&\mathbb{E}\left[\log p(\boldsymbol{\tau}) - \log q(\boldsymbol{\tau})\right] \\ &= \sum_{l=1}^{L+1} \mathbb{E}\left[\log \text{GIG}\left(\tau_l | \nu_{\text{glob}}, \delta_{\text{glob}}, \lambda_{\text{glob}}\right) - \log \text{GIG}\left(\tau_l | \hat{\nu}_{\text{glob},l}, \hat{\delta}_{\text{glob},l}, \lambda_{\text{glob}}\right)\right] \\ &= C_\tau + \sum_{l=1}^{L+1} \mathbb{E}\left[\log \tau_l^{\nu_{\text{glob}}-1} \exp\left(-\frac{1}{2}\left(\frac{\delta_{\text{glob}}^2}{\tau_l} + \lambda_{\text{glob}}^2 \tau_l\right)\right)\right] \\ &- \sum_{l=1}^{L+1} \mathbb{E}\left[\log\left(\tau_l^{\hat{\nu}_{\text{glob},l}-1}\right) \exp\left(-\frac{1}{2}\left(\frac{\hat{\delta}_{\text{glob},l}^2}{\tau_l} + \lambda_{\text{glob}}^2 \tau_l\right)\right)\right] \\ &= C_\tau + \frac{1}{2} \sum_{l=1}^{L+1} D_l D_{l-1} \mathbb{E}\left[\log \tau_l\right] + \mathbb{E}\left[\frac{1}{\tau_l}(\hat{\delta}_{\text{glob},l}^2 - \delta_{\text{glob}}^2)\right],\end{aligned}$$

where the normalizing constant is

$$\begin{aligned}C_\tau = &\sum_{l=1}^{L+1} (\nu_{\text{glob}} - \hat{\nu}_{\text{glob},l}) \log(\lambda_{\text{glob}}) + \hat{\nu}_{\text{glob},l} \log(\hat{\delta}_{\text{glob},l}) - \nu_{\text{glob}} \log(\delta_{\text{glob}}) \\ &+ \sum_{l=1}^{L+1} \log(K_{\hat{\nu}_{\text{glob},l}}(\lambda_{\text{glob}} \hat{\delta}_{\text{glob},l})) - \log(K_{\nu_{\text{glob}}}(\lambda \delta_{\text{glob}})).\end{aligned}$$

ELBO of $\boldsymbol{\psi}$. Similarly, the terms of the ELBO containing the local shrinkage parameters are

$$
\begin{aligned}
\mathbb{E}\left[\log p(\boldsymbol{\psi}) - \log q(\boldsymbol{\psi})\right] &= C_{\boldsymbol{\psi}} + \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \mathrm{GIG}\left(\psi_{l,d,d'} | \nu_{\mathrm{loc},l}, \delta_{\mathrm{loc},l}, \lambda_{\mathrm{loc},l}\right)\right] \\
&- \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \mathrm{GIG}\left(\psi_{l,d,d'} | \hat{\nu}_{\mathrm{loc},l,d,d'}, \hat{\delta}_{\mathrm{loc},l,d,d'}, \lambda_{\mathrm{loc},l}\right)\right] \\
&= C_{\boldsymbol{\psi}} + \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \psi_{l,d,d'}^{\nu_{\mathrm{loc},l}-1} \exp\left(-\frac{1}{2}\left(\frac{\delta_{\mathrm{loc},l}^2}{\psi_{l,d,d'}} + \lambda_{\mathrm{loc},l}^2 \psi_{l,d,d'}\right)\right)\right] \\
&= -\sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \left(\psi_{l,d,d'}^{\hat{\nu}_{\mathrm{loc},l,d,d'}-1}\right) \exp\left(-\frac{1}{2}\left(\frac{\hat{\delta}_{\mathrm{loc},l,d,d'}^2}{\psi_{l,d,d'}} + \lambda_{\mathrm{loc},l}^2 \psi_{l,d,d'}\right)\right)\right] \\
&= C_{\boldsymbol{\psi}} + \frac{1}{2}\sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \psi_{l,d,d'}\right] + \mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\left(\hat{\delta}_{\mathrm{loc},l,d,d'}^2 - \delta_{\mathrm{loc},l}^2\right),
\end{aligned}
$$

where the normalizing constant is

$$
\begin{aligned}
C_{\boldsymbol{\psi}} = &\sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} (\nu_{\mathrm{loc},l} - \hat{\nu}_{\mathrm{loc},l,d,d'}) \log(\lambda_{\mathrm{loc},l}) + \hat{\nu}_{\mathrm{loc},l,d,d'} \log(\hat{\delta}_{\mathrm{loc},l,d,d'}) - \nu_{\mathrm{loc},l} \log(\delta_{\mathrm{loc},l}) \\
&+ \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}} \log(K_{\hat{\nu}_{\mathrm{loc},l,d,d'}}(\lambda_{\mathrm{glob}}\hat{\delta}_{\mathrm{loc},l,d,d'})) - \log(K_{\nu_{\mathrm{loc},l}}(\lambda_{\mathrm{loc},l}\delta_{\mathrm{loc},l})).
\end{aligned}
$$

ELBO of $\boldsymbol{\eta}$. As before, the covariance matrix matrix is assumed to be diagonal so that the relevant ELBO is:

$$
\begin{aligned}
\mathbb{E}\left[\log p(\boldsymbol{\Sigma}) - \log q(\boldsymbol{\eta})\right] &= C_{\boldsymbol{\eta}} + \sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\log \mathrm{IG}(\eta_{l,d}^2 | \alpha_0^h, \beta_0^h)\right] \\
&+ \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log \mathrm{IG}(\eta_{l,d}^2 | \alpha_0, \beta_0)\right] - \sum_{l}^{L+1}\sum_{d}^{D_l} \mathbb{E}\left[\log \mathrm{IG}(\eta_{l,d}^2 | \alpha_{l,d}, \beta_{l,d})\right] \\
&= C_{\boldsymbol{\eta}} + \sum_{l}^{L}\sum_{d}^{D_l} (\alpha_{l,d} - \alpha_0^h)\, \mathbb{E}\left[\log \eta_{l,d}^2\right] + \sum_{d=1}^{D_{L+1}} (\alpha_{L+1,d} - \alpha_0)\, \mathbb{E}\left[\log \eta_{L+1,d}^2\right] \\
&+ \sum_{l=1}^{L}\sum_{d=1}^{D_l} (\beta_{l,d} - \beta_0^h)\, \mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right] + \sum_{d=1}^{D_{L+1}} (\beta_{L+1,d} - \beta_0)\, \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right] \\
&= C_{\boldsymbol{\eta}} + \frac{N}{2}\sum_{l}^{L+1}\sum_{d}^{D_l} \mathbb{E}\left[\log \eta_{l,d}^2\right] + \sum_{l=1}^{L}\sum_{d=1}^{D_l} (\beta_{l,d} - \beta_0^h)\, \mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right] \\
&+ \sum_{d=1}^{D_{L+1}} (\beta_{L+1,d} - \beta_0)\, \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right],
\end{aligned}
$$

where the normalizing constant is

$$
C_{\boldsymbol{\eta}} = \sum_{l=1}^{L}\sum_{d=1}^{D_l} \alpha_0^h \log \beta_0^h - \alpha_{l,d} \log \beta_{l,d} + \log \Gamma(\alpha_{l,d}) - \log \Gamma(\alpha_0^h)
$$

$$+\sum_{d=1}^{D_{L+1}} \alpha_0 \log \beta_0 - \alpha_{L+1,d} \log \beta_{L+1,d} + \log \Gamma(\alpha_{L+1,d}) - \log \Gamma(\alpha_0).$$

ELBO of $(\boldsymbol{W}, \boldsymbol{b})$. Recall, that we previously introduced matrices $\boldsymbol{D}_{l,d}$, and denote further $\boldsymbol{D}^0_{l,d} = \operatorname{diag}\left(s_0^2, \tau_l \psi_{l,d,1}, \ldots, \tau_l \psi_{l,d,D_{l-1}}\right)$. Then the ELBO of weights and biases is:

$$\begin{aligned}
&\mathbb{E}\left[\log p(\boldsymbol{W}|\boldsymbol{\psi}, \boldsymbol{\tau})\right] + \mathbb{E}\left[\log p(\boldsymbol{b})\right] - \mathbb{E}\left[\log q(\boldsymbol{W}, \boldsymbol{b})\right] \\
&= \sum_{l=1}^{L+1} \sum_{d=1}^{D_l} \sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \mathcal{N}\left(\widetilde{\boldsymbol{W}}_{l,d} | 0, \boldsymbol{D}^0_{l,d}\right)\right] - \sum_{l}^{L+1} \sum_{d}^{D_l} \mathbb{E}\left[\log \mathcal{N}\left(\widetilde{\boldsymbol{W}}_{l,d} | \boldsymbol{m}_{l,d}, \boldsymbol{B}_{l,d}\right)\right] \\
&= \sum_{l=1}^{L+1} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \left(|\boldsymbol{D}_{l,d}|\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \widetilde{\boldsymbol{W}}_{l,d} \left(\boldsymbol{D}^0_{l,d}\right)^{-1} \widetilde{\boldsymbol{W}}^T_{l,d}\right)\right] \\
&- \sum_{l}^{L+1} \sum_{d}^{D_l} \mathbb{E}\left[\log |\boldsymbol{B}_{l,d}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \left(\widetilde{\boldsymbol{W}}_{l,d} - \boldsymbol{m}_{l,d}\right) \boldsymbol{B}^{-1}_{l,d} \left(\widetilde{\boldsymbol{W}}_{l,d} - \boldsymbol{m}_{l,d}\right)^T\right)\right] \\
&= \frac{1}{2} \sum_{l}^{L+1} \sum_{d}^{D_l} \mathbb{E}\left[\log |\boldsymbol{B}_{l,d}|\right] - \mathbb{E}\left[\log\left(|\boldsymbol{D}^0_{l,d}|\right)\right] - \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d} (\boldsymbol{D}^0_{l,d})^{-1} \widetilde{\boldsymbol{W}}^T_{l,d}\right] \\
&+ \frac{1}{2} \sum_{l}^{L+1} \sum_{d}^{D_l} \mathbb{E}\left[\left(\widetilde{\boldsymbol{W}}_{l,d} - \boldsymbol{m}_{l,d}\right) \boldsymbol{B}^{-1}_{l,d} \left(\widetilde{\boldsymbol{W}}_{l,d} - \boldsymbol{m}_{l,d}\right)^T\right] \\
&= \frac{1}{2} \sum_{l}^{L+1} \sum_{d}^{D_l} \log |\boldsymbol{B}_{l,d}| - \mathbb{E}\left[\log\left(|\boldsymbol{D}^0_{l,d}|\right)\right] - \operatorname{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}^T_{l,d} \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[(\boldsymbol{D}^0_{l,d})^{-1}\right]\right) \\
&= \frac{1}{2} \sum_{l}^{L+1} \sum_{d}^{D_l} \left(\log |\boldsymbol{B}_{l,d}| - \operatorname{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}^T_{l,d} \widetilde{\boldsymbol{W}}_{l,d}\right] \boldsymbol{D}_{l,d}\right) - \sum_{d'=1}^{D_{l-1}} \mathbb{E}\left[\log \psi_{l,d,d'}\right]\right) \\
&- \frac{1}{2} \sum_{l}^{L+1} D_l \left(\log s_0^2 + D_{l-1} \mathbb{E}\left[\log \tau_l\right] - 1\right) + \sum_{l}^{L+1} \frac{D_l}{2}.
\end{aligned}$$

ELBO of $\boldsymbol{a}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$. The remaining terms of the ELBO are the ones with stochastic and binary activations and additional augmented variables:

$$\begin{aligned}
&\mathbb{E}\left[\log p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega} | \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma})\right] - \mathbb{E}\left[\log q(\boldsymbol{a})\right] - \mathbb{E}\left[\log q(\boldsymbol{\gamma})\right] - \mathbb{E}\left[\log q(\boldsymbol{\omega})\right] \\
&= \sum_{n=1}^{N} \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log \mathcal{N}\left(y_{n,d} | \boldsymbol{z}_{n,L+1,d}, \boldsymbol{\Sigma}_{L+1,d}\right)\right] \\
&+ \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \mathcal{N}\left(\mathrm{a}_{n,l,d} | \boldsymbol{\gamma}_{n,d} \odot \boldsymbol{z}_{n,l,d}, \boldsymbol{\Sigma}_{l,d}\right)\right] \\
&+ \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \left(\exp\left(\frac{\kappa_{n,l,d} z_{n,l,d}}{T}\right) \exp\left(-\frac{\omega_{n,l,d} z^2_{n,l,d}}{2T^2}\right) \mathrm{PG}(\omega_{n,l,d} | 1, 0)\right)\right] \\
&- \sum_{n=1}^{N} \sum_{l=1}^{L} \mathbb{E}\left[\log \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l} \boldsymbol{a}_{n,l-1}, \boldsymbol{S}_{n,l}\right)\right] \\
&- \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \mathrm{Bern}\left(\gamma_{n,l,d} | \rho_{n,l,d}\right)\right] + \mathbb{E}\left[\log \mathrm{PG}(\omega_{n,l,d} | 1, \mathrm{a}_{n,l,d})\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=1}^{N} \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log(\eta_{L+1,d}^2)^{-1/2} \exp\left(-\frac{1}{2\eta_{L+1,d}^2}\left(y_{n,d} - \boldsymbol{W}_{L+1,d}\boldsymbol{a}_{n,L} - b_{L+1,d}\right)^2\right)\right] \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\log(\eta_{l,d}^2)^{-1/2} \exp\left(-\frac{1}{2\eta_{l,d}^2}\left(\mathrm{a}_{n,l,d} - \gamma_{n,l,d} \odot \left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)\right)^2\right)\right] \\
&- N\sum_{l=1}^{L} D_l \log(2) - \frac{ND_{L+1}}{2}\log(2\pi) \\
&+ \frac{1}{T}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\left(\gamma_{n,d} - \frac{1}{2}\right)\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)\right] \\
&- \frac{1}{2T^2}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\omega_{n,l,d}\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)^2\right] \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\left[\log|\boldsymbol{S}_{n,l}|^{-\frac{1}{2}}\right] \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\left[-\frac{1}{2}\left(\boldsymbol{a}_{n,l} - \boldsymbol{t}_{n,l} - \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)^T \boldsymbol{S}_{n,l}^{-1}\left(\boldsymbol{a}_{n,l} - \boldsymbol{t}_{n,l} - \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right] \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \left(\rho_{n,l,d}\log\rho_{n,l,d} + (1-\rho_{n,l,d})\log(1-\rho_{n,l,d})\right) \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\log\frac{\mathrm{PG}(\omega_{n,l,d}|1,0)}{\mathrm{PG}(\omega_{n,l,d}|1,A_{n,d})}\right] \\
=& -\frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\left(y_{n,d}^2 - 2y_{n,d}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right]\right) \\
&- \frac{1}{2}\sum_{n=1}^{N}\sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right]\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}^T\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}_{n,L}^T\right]\right)\right) \\
&- \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\left(\mathbb{E}\left[\mathrm{a}_{n,l,d}^2\right] - 2\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\mathrm{a}_{n,l,d}\right]\right) \\
&- \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\mathbb{E}\left[\gamma_{n,l,d}^2\right]\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right) \\
&- \frac{N}{2}\sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log\eta_{L+1,d}^2\right] - \frac{N}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\log\eta_{l,d}^2\right] + \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\log(|\boldsymbol{S}_{n,l}|) \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \frac{1}{T}\left(\rho_{n,l,d} - \frac{1}{2}\right)\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \\
&- \frac{1}{2T^2}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \mathbb{E}\left[\omega_{n,l,d}\right]\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right)\right) \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \rho_{n,l,d}\log\rho_{n,l,d} + (1-\rho_{n,l,d})\log(1-\rho_{n,l,d}) \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l} \frac{\mathrm{a}_{n,l,d}^2}{2}\mathbb{E}\left[\omega_{n,l,d}\right] - \log(\cosh(\frac{\mathrm{a}_{n,l,d}}{2})) + C_a,
\end{aligned}
$$

where the normalizing constant is

$$C_a = -\frac{ND_{L+1}}{2}\log(2\pi) - N\sum_{l=1}^{L} D_l \log(2).$$

Total ELBO Then, we can sum the derived above parts to get the total ELBO of our model:

$$\begin{aligned}
\text{ELBO} = \text{const.} &+ \sum_{l=1}^{L+1}\frac{1}{2}\mathbb{E}\left[\frac{1}{\tau_l}\right]\left(\hat{\delta}^2_{\text{glob},l} - \delta^2_{\text{glob}}\right) + \left(\hat{\nu}_{\text{glob},l}\log(\hat{\delta}_{\text{glob},l}) + \log(K_{\hat{\nu}_{\text{glob},l}}(\lambda_{\text{glob}}\hat{\delta}_{\text{glob},l})\right) \\
&+ \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}}\frac{1}{2}\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\left(\hat{\delta}^2_{\text{loc},l,d,d'} - \delta^2_{\text{loc},l}\right) \\
&+ \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\sum_{d'=1}^{D_{l-1}}\hat{\nu}_{\text{loc},l,d,d'}\log(\hat{\delta}_{\text{loc},l,d,d'}) + \log(K_{\hat{\nu}_{\text{loc},l,d,d'}}(\lambda_{\text{loc},l}\hat{\delta}_{\text{loc},l,d,d'})) \\
&+ \sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{l,d}}\right](\beta_{l,d} - \beta^l_0) - \alpha_{l,d}\log\beta_{l,d} + \frac{1}{2}\log|\boldsymbol{B}_{l,d}| \\
&- \frac{1}{2}\sum_{l=1}^{L+1}\sum_{d=1}^{D_l}\left(\frac{1}{s^2_0}\mathbb{E}[b^2_{l,d}] + \sum_{d'=1}^{D_{l-1}}\mathbb{E}\left[\frac{1}{\tau_l}\right]\mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right]\mathbb{E}[w^2_{l,d,d'}]\right) \\
&+ \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\log(\boldsymbol{S}_{n,l}) - \frac{1}{2}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta^2_{L+1,d}}\right]\left(\sum_{n=1}^{N}\mathbb{E}\left[\left(y_{n,d} - \mathbb{E}\left[\tilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,L}\right]\right)^2\right]\right) \\
&- \frac{1}{2}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta^2_{L+1,d}}\right]\sum_{n=1}^{N}\text{Tr}\left(\left(\boldsymbol{B}_{L+1,d} + \boldsymbol{m}_{L+1,d}\boldsymbol{m}^T_{L+1,d}\right)\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,L}\tilde{\boldsymbol{a}}^T_{n,L}\right]\right) \\
&+ \frac{1}{2}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta^2_{L+1,d}}\right]\sum_{n=1}^{N}\text{Tr}\left(\boldsymbol{m}_{L+1,d}\boldsymbol{m}^T_{L+1,d}\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,L}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}^T_{n,L}\right]\right) \\
&- \frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{1,d}}\right]\left(\sum_{n=1}^{N}\left(\rho_{n,l,d}\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\right] - \mathbb{E}\left[\boldsymbol{a}_{n,l,d}\right]\right)^2 + \mathbb{E}\left[\boldsymbol{a}^2_{n,l,d}\right] - \mathbb{E}\left[\boldsymbol{a}_{n,l,d}\right]^2\right) \\
&- \frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{1,d}}\right]\left(\sum_{n=1}^{N}\rho_{n,l,d}\,\text{Tr}\left(\left(\boldsymbol{B}_{l,d} + \boldsymbol{m}_{l,d}\boldsymbol{m}^T_{l,d}\right)\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\tilde{\boldsymbol{a}}^T_{n,l-1}\right]\right)\right) \\
&+ \frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{1,d}}\right]\left(\sum_{n=1}^{N}\rho^2_{n,l,d}\,\text{Tr}\left(\boldsymbol{m}_{l,d}\boldsymbol{m}^T_{l,d}\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}^T_{n,l-1}\right]\right)\right) \\
&- \sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{1,d}}\right]\left(\sum_{n=1}^{N}\rho_{n,l,d}\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\left(\mathbb{E}\left[\boldsymbol{a}_{n,l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\right] - \mathbb{E}\left[\boldsymbol{a}_{n,l,d}\tilde{\boldsymbol{a}}_{n,l-1}\right]\right)\right) \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{1}{T}\left(\rho_{n,l,d} - \frac{1}{2}\right)\left(\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\right]\right) \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{1}{2T^2}\mathbb{E}\left[\omega_{n,l,d}\right]\left(\text{Tr}\left(\left(\boldsymbol{B}_{l,d} + \boldsymbol{m}_{l,d}\boldsymbol{m}^T_{l,d}\right)\mathbb{E}\left[\tilde{\boldsymbol{a}}_{n,l-1}\tilde{\boldsymbol{a}}^T_{n,l-1}\right]\right)\right) \\
&- \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\rho_{n,l,d}\log\rho_{n,l,d} + (1-\rho_{n,l,d})\log(1-\rho_{n,l,d})
\end{aligned}$$

$$-\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{\mathrm{a}_{n,l,d}^2}{2}\mathbb{E}\left[\omega_{n,l,d}\right]+\log(\cosh(\mathrm{a}_{n,l,d}/2)).$$

Note that when implementing VI with the EM scheme, we adjust the formula above by adding the term which arises in the normalizing constant C_τ defined when computing the ELBO of global shrinkage parameters, specifically, we add

$$\begin{aligned}\mathrm{ELBO}_{EM} = (L+1)\left(\nu_{\mathrm{glob}}\left(\log(\lambda_{\mathrm{glob}})-\log(\delta_{\mathrm{glob}})\right)-\log\left(K_{\nu_{\mathrm{glob}}}(\lambda_{\mathrm{glob}}\delta_{\mathrm{glob}})\right)\right)\\ +\sum_{l=1}^{L+1}(\nu_{\mathrm{glob}}-1)\mathbb{E}\left[\log\tau_l\right]-\frac{1}{2}\lambda_{\mathrm{glob}}^2\mathbb{E}\left[\tau_l\right]-\nu_l\log(\lambda_{\mathrm{glob}}).\end{aligned}$$

B.2.2 ELBO for prediction

To obtain the posterior predictive distribution, we compute the approximate variational predictive distributions of $\boldsymbol{a}_*$, $\boldsymbol{\gamma}_*$ and $\boldsymbol{\omega}_*$ with the objective function being the ELBO of Equation (2.5). Thus, in the predictive step of our algorithm, we monitor the convergence of the ELBO of $\boldsymbol{a}_*$, $\boldsymbol{\gamma}_*$ and $\boldsymbol{\omega}_*$, which we derive as follows:

$$\begin{aligned}&\mathbb{E}\left[\log p(\boldsymbol{a}_*,\boldsymbol{\gamma}_*,\boldsymbol{\omega}_*|\boldsymbol{W},\boldsymbol{b},\boldsymbol{\Sigma})\right]-\mathbb{E}\left[\log q(\boldsymbol{a}_*)\right]-\mathbb{E}\left[\log q(\boldsymbol{\gamma}_*)\right]-\mathbb{E}\left[\log q(\boldsymbol{\omega}_*)\right]\\
&=\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\log\mathcal{N}\left(a_{*,l,d}|\boldsymbol{\gamma}_{*,d}\odot\boldsymbol{z}_{*,l,d},\boldsymbol{\Sigma}_{l,d}\right)\right]\\
&+\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\log\left(\exp\left(\frac{\kappa_{*,l,d}z_{*,l,d}}{T}\right)\exp\left(-\frac{\omega_{*,l,d}z_{*,l,d}^2}{2T^2}\right)\mathrm{PG}(\omega_{n,l,d}|1,0)\right)\right]-\\
&-\sum_{l=1}^{L}\mathbb{E}\left[\log\mathcal{N}\left(\boldsymbol{a}_{*,l}|\boldsymbol{t}_{*,l}+\boldsymbol{M}_{*,l}\boldsymbol{a}_{*,l-1},\boldsymbol{S}_{*,l}\right)\right]\\
&-\sum_{l=1}^{L}\sum_{d=1}^{D_l}\left(\mathbb{E}\left[\log\mathrm{Bern}\left(\gamma_{*,l,d}|\rho_{*,l,d}\right)\right]+\mathbb{E}\left[\log\mathrm{PG}(\omega_{*,l,d}|1,A_{*,l,d})\right]\right)\\
&=-\frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\left(\mathbb{E}\left[a_{*,l,d}^2\right]-2\mathbb{E}\left[\gamma_{*,l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}a_{*,l,d}\right]\right)\\
&-\frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\log\eta_{l,d}^2\right]+\frac{1}{2}\sum_{l=1}^{L}\log(|\boldsymbol{S}_{*,l}|)+\frac{1}{T}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\left(\rho_{*,l,d}-\frac{1}{2}\right)\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}\right]\\
&-\frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\left(\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\mathbb{E}\left[\gamma_{*,l,d}^2\right]+\frac{1}{T^2}\mathbb{E}\left[\omega_{*,l,d}\right]\right)\mathrm{Tr}\left(\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}^T\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\tilde{\boldsymbol{a}}_{*,l-1}^T\right]\right)\\
&-\sum_{l=1}^{L}\sum_{d=1}^{D_l}\left(\rho_{*,l,d}\log\rho_{*,l,d}+(1-\rho_{*,l,d})\log(1-\rho_{*,l,d})\right)\\
&+\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{A_{*,l,d}^2}{2}\mathbb{E}\left[\omega_{*,l,d}\right]-\sum_{l=1}^{L}\sum_{d=l}^{D_l}\log(\cosh(\frac{A_{*,l,d}}{2}))+\mathrm{const}\\
&=-\frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{1,d}^2}\right]\left(\left(\rho_{*,l,d}\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\right]-\mathbb{E}\left[a_{*,l,d}\right]\right)^2+\mathbb{E}\left[a_{*,l,d}^2\right]-\mathbb{E}\left[a_{*,l,d}\right]^2\right)\end{aligned}$$

$$
\begin{aligned}
&- \frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\left(\rho_{*,l,d}\,\mathrm{Tr}\left(\left(\boldsymbol{B}_{l,d}+\boldsymbol{m}_{l,d}\boldsymbol{m}_{l,d}^T\right)\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\tilde{\boldsymbol{a}}_{*,l-1}^T\right]\right)\right)\\
&+ \frac{1}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\left(\rho_{*,l,d}^2\,\mathrm{Tr}\left(\boldsymbol{m}_{l,d}\boldsymbol{m}_{l,d}^T\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}^T\right]\right)\right)\\
&- \sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta_{l,d}^2}\right]\left(\rho_{*,l,d}\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}\right]\left(\mathbb{E}\left[\boldsymbol{a}_{*,l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\right]-\mathbb{E}\left[a_{*,l,d}\tilde{\boldsymbol{a}}_{*,l-1}\right]\right)\right)+\frac{1}{2}\mathbb{E}\left[\log\eta_{l,d}^2\right]\\
&+ \frac{1}{2}\sum_{l=1}^{L}\log(|\boldsymbol{S}_{*,l}|)+\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{1}{T}\left(\rho_{*,l,d}-\frac{1}{2}\right)\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{*,l-1}\right]\\
&+ \sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{1}{2T^2}\mathbb{E}\left[\omega_{*,l,d}\right]\mathrm{Tr}\left(\mathbb{E}\left[\tilde{\boldsymbol{W}}_{l,d}^T\tilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\tilde{\boldsymbol{a}}_{*,l-1}\tilde{\boldsymbol{a}}_{*,l-1}^T\right]\right)\\
&- \sum_{l=1}^{L}\sum_{d=1}^{D_l}\rho_{*,l,d}\log\rho_{*,l,d}+(1-\rho_{*,l,d})\log(1-\rho_{*,l,d})\\
&+ \sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{A_{*,l,d}^2}{2}\mathbb{E}\left[\omega_{*,l,d}\right]-\log(\cosh(\frac{A_{*,l,d}}{2}))+\text{const.}
\end{aligned}
$$

B.3 Supplementary to the Stochastic Variational Inference for VBNN

Here we provide additional details on the SVI developed for the VBNN in Section 4.3.3. During one iteration t of the algorithm, one proceeds as follows:

1. Sample indices S_t uniformly, without replacement.

2. For $t = 1$ initialize as in Appendix B.4.1 (similarly to CAVI) but where the input of Algorithm 8 is taken to be $\boldsymbol{x}_n$ for $n \in S_t$. For $t > 1$ only initialize local parameters of $\boldsymbol{a}$ and γ by setting $\boldsymbol{z}_{n,0} = \boldsymbol{x}_n$ and iterating for $l = 1, \ldots, L$ and $n \in S_t$ through

$$
\begin{aligned}
&\rho_{n,l,d} = \sigma\left(\frac{(m_{l,d}^b)^{(t)} + (\boldsymbol{m}_{l,d}^W)^{(t)}\boldsymbol{z}_{n,l-1}}{T}\right) \quad d = 1, \ldots, D_l,\\
&\boldsymbol{M}_{n,l} = (\boldsymbol{m}_l^W)^{(t)} \odot \boldsymbol{\rho}_{n,l}\mathbf{1}_{D_l}^T, \text{ where by } \mathbf{1} \text{ we denote a vector of ones,}\\
&\boldsymbol{t}_{n,l} = (m_l^b)^{(t)} \odot \boldsymbol{\rho}_{n,l},\\
&\boldsymbol{z}_{n,l} = \boldsymbol{M}_{n,l}\boldsymbol{z}_{n,l-1} + \boldsymbol{t}_{n,l}.
\end{aligned}
$$

3. Set $\ell_t = (t+1)^{-k},\ k \in (0.5, 1]$.

4. Update global shrinkage parameters of $\boldsymbol{\tau}$ as in CAVI, for $l = 1, \ldots, L+1$,

$$
\nu_{\text{glob},l}^{(t)} = \nu_{\text{glob}} - \frac{D_l D_{l-1}}{2},
$$

$$\delta_{\text{glob},l}^{(t)} = \sqrt{\delta_{\text{glob}}^2 + \sum_{d}^{D_l} \sum_{d'}^{D_{l-1}} \mathbb{E}\left[\frac{1}{\psi_{l,d,d'}}\right] \mathbb{E}\left[W_{l,d,d'}^2\right]},$$

where $\nu_{\text{glob},l}$ is only updated in the first iteration of the algorithm.

5. Update local shrinkage parameters of $\boldsymbol{\psi}$ as in CAVI, for $l = 1, \ldots, L{+}1,\ d = 1, \ldots, D_l,\ d' = 1, \ldots, D_{l-1}$,

$$\nu_{\text{loc},l,d,d'}^{(t)} = \nu_{\text{loc},l} - \frac{1}{2},$$
$$\delta_{\text{loc},l,d,d'}^{(t)} = \sqrt{\mathbb{E}\left[\frac{1}{\tau_l}\right] \mathbb{E}\left[W_{l,d,d'}^2\right] + \delta_{\text{loc},l}^2},$$

where $\nu_{\text{loc},l,d,d'}$ is only updated once.

6. Find optimal variational parameters of local variables $\boldsymbol{\omega}$, $\boldsymbol{\gamma}$, $\boldsymbol{a}$, namely, update $A^{(t)}, \boldsymbol{S}^{(t)}, \boldsymbol{t}^{(t)}, \boldsymbol{M}^{(t)}, \boldsymbol{\rho}^{(t)}$ in a coordinate ascent algorithm and monitor the local ELBO for convergence:

   - For $n \in S,\ l = 1, \ldots, L, d = 1, \ldots, D_l$, update

$$\mathrm{a}_{n,l,d} = \frac{1}{T}\sqrt{\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1} \widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right)\right)}.$$

   - Starting from the final layer $l = L$, update for $n \in S$

$$\begin{aligned}
&\boldsymbol{S}_{n,L}^{-1} = \hat{\boldsymbol{\Sigma}}_L^{-1} + \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right] \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T \boldsymbol{W}_{L+1,d}\right] \quad \text{(same for all n)},\\
&\boldsymbol{t}_{n,L} = \boldsymbol{S}_L \Bigg(\hat{\boldsymbol{\Sigma}}_L^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right] \odot \mathbb{E}\left[\boldsymbol{b}_L\right]\\
&+ \sum_{d=1}^{D_{l+1}} \mathbb{E}\left[\frac{1}{\eta_{L+1,d}^2}\right] \left(-\mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T b_{L+1,d}\right] + \mathbb{E}\left[\boldsymbol{W}_{L+1,d}^T\right] y_{n,d}\right)\Bigg),\\
&\boldsymbol{M}_{n,L} = \boldsymbol{S}_L \hat{\boldsymbol{\Sigma}}_L^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,L}\right] \mathbf{1}_{D_{L-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_L\right],\\
&\hat{\boldsymbol{\Sigma}}_L^{-1} = \mathrm{diag}\left(\mathbb{E}\left[\eta_{L,1}^{-2}\right], \ldots, \mathbb{E}\left[\eta_{L,D_L}^{-2}\right]\right).
\end{aligned}$$

   Then, in reverse order, for $l = L-1, \ldots, 1$ and for $n \in S$ update

$$\begin{aligned}
\boldsymbol{S}_{n,l}^{-1} &= \hat{\boldsymbol{\Sigma}}_l^{-1} - \boldsymbol{M}_{n,l+1}^T \boldsymbol{S}_{n,l+1}^{-1} \boldsymbol{M}_{n,l+1}\\
&\quad + \sum_{d=1}^{D_{l+1}} \left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right] \mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2} \mathbb{E}\left[\omega_{n,l+1,d}\right]\right) \mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T \boldsymbol{W}_{l+1,d}\right],\\
\boldsymbol{t}_{n,l} &= \boldsymbol{S}_{n,l} \Bigg(\boldsymbol{M}_{n,l+1}^T \boldsymbol{S}_{n,l+1}^{-1} \boldsymbol{t}_{n,l+1} + \hat{\boldsymbol{\Sigma}}_l^{-1} \mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right] \odot \mathbb{E}\left[\boldsymbol{b}_l\right]\\
&\quad + \frac{1}{T} \sum_{d=1}^{D_{l+1}} \mathbb{E}\left[\boldsymbol{W}_{l+1,d}^T\right] \left(\mathbb{E}\left[\gamma_{n,l+1,d}\right] - \frac{1}{2}\right)
\end{aligned}$$

$$- \sum_{d=1}^{D_{l+1}} \left(\mathbb{E}\left[\frac{1}{\eta_{l+1,d}^2}\right] \mathbb{E}\left[\gamma_{n,l+1,d}\right] + \frac{1}{T^2}\mathbb{E}\left[\omega_{n,l+1,d}\right] \right) \mathbb{E}\left[\boldsymbol{W}_{l+1,d} b_{l+1,d}\right] \Bigg),$$

$$\boldsymbol{M}_{n,l} = \boldsymbol{S}_{n,l}\hat{\boldsymbol{\Sigma}}_l^{-1}\mathbb{E}\left[\boldsymbol{\gamma}_{n,l}\right]\mathbf{1}_{D_{l-1}}^T \odot \mathbb{E}\left[\boldsymbol{W}_l\right],$$

$$\hat{\boldsymbol{\Sigma}}_l^{-1} = \operatorname{diag}\left(\mathbb{E}\left[\eta_{l,1}^{-2}\right], \ldots, \mathbb{E}\left[\eta_{l,D_l}^{-2}\right]\right).$$

- For $n \in S,\ l = 1, \ldots, L,\ d = 1, \ldots, D_l$, update

$$\rho_{n,l,d} = \sigma\left(-\frac{\mathbb{E}\left[\eta_{l,d}^{-2}\right]}{2} \operatorname{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T \widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right] \right) + \mathbb{E}\left[\eta_{l,d}^{-2}\right] \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\mathrm{a}_{n,l,d}\right] + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right] \mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right] \right).$$

The local ELBO is given by

$$\begin{aligned}
&\mathbb{E}\left[\log p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\omega} | \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma})\right] - \mathbb{E}\left[\log q(\boldsymbol{a})\right] - \mathbb{E}\left[\log q(\boldsymbol{\gamma})\right] - \mathbb{E}\left[\log q(\boldsymbol{\omega})\right] \\
&= \frac{N}{|S|} \sum_{n \in S} \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log \mathcal{N}\left(y_{n,d} | \boldsymbol{z}_{n,L+1,d}, \boldsymbol{\Sigma}_{L+1,d}\right)\right] \\
&+ \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \mathcal{N}\left(\mathrm{a}_{n,l,d} | \boldsymbol{\gamma}_{n,d} \odot \boldsymbol{z}_{n,l,d}, \boldsymbol{\Sigma}_{l,d}\right)\right] \\
&+ \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \left(\exp\left(\frac{\kappa_{n,l,d} z_{n,l,d}}{T}\right) \exp\left(-\frac{\omega_{n,l,d} z_{n,l,d}^2}{2T^2}\right) \mathrm{PG}(\omega_{n,l,d} | 1, 0) \right)\right] \\
&- \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \mathbb{E}\left[\log \mathcal{N}\left(\boldsymbol{a}_{n,l} | \boldsymbol{t}_{n,l} + \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}, \boldsymbol{S}_{n,l}\right)\right] \\
&- \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log \mathrm{Bern}\left(\gamma_{n,l,d} | \rho_{n,l,d}\right)\right] + \mathbb{E}\left[\log \mathrm{PG}(\omega_{n,l,d} | 1, \mathrm{a}_{n,l,d})\right] \\
&= \frac{N}{|S|} \sum_{n \in S} \sum_{d=1}^{D_{L+1}} \mathbb{E}\left[\log(\eta_{L+1,d}^2)^{-1/2} \exp\left(-\frac{1}{2\eta_{L+1,d}^2}\left(y_{n,d} - \boldsymbol{W}_{L+1,d}\boldsymbol{a}_{n,L} - b_{L+1,d}\right)^2\right)\right] \\
&+ \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\log(\eta_{l,d}^2)^{-1/2} \exp\left(-\frac{1}{2\eta_{l,d}^2}\left(\mathrm{a}_{n,l,d} - \gamma_{n,l,d} \odot \left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)\right)^2\right)\right] \\
&- N \sum_{l=1}^{L} D_l \log(2) - \frac{N D_{L+1}}{2} \log(2\pi) \\
&+ \frac{1}{T}\frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\left(\gamma_{n,d} - \frac{1}{2}\right)\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)\right] \\
&- \frac{1}{2T^2}\frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \sum_{d=1}^{D_l} \mathbb{E}\left[\omega_{n,l,d}\left(\boldsymbol{W}_{l,d}\boldsymbol{a}_{n,l-1} + b_{l,d}\right)^2\right] \\
&- \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \mathbb{E}\left[\log |\boldsymbol{S}_{n,l}|^{-\frac{1}{2}}\right] \\
&- \frac{N}{|S|} \sum_{n \in S} \sum_{l=1}^{L} \mathbb{E}\left[-\frac{1}{2}\left(\boldsymbol{a}_{n,l} - \boldsymbol{t}_{n,l} - \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)^T \boldsymbol{S}_{n,l}^{-1}\left(\boldsymbol{a}_{n,l} - \boldsymbol{t}_{n,l} - \boldsymbol{M}_{n,l}\boldsymbol{a}_{n,l-1}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&- \frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\left(\rho_{n,l,d}\log\rho_{n,l,d} + (1-\rho_{n,l,d})\log(1-\rho_{n,l,d})\right)\\
&+ \frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\log\frac{\mathrm{PG}(\omega_{n,l,d}|1,0)}{\mathrm{PG}(\omega_{n,l,d}|1,A_{n,d})}\right]\\
=&-\frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta^2_{L+1,d}}\right]\left(y^2_{n,d} - 2y_{n,d}\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\right]\right)\\
&-\frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\frac{1}{\eta^2_{L+1,d}}\right]\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}^T_{L+1,d}\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}^T_{n,L}\right]\right)\right)\\
&-\frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{l,d}}\right]\left(\mathbb{E}\left[\mathrm{a}^2_{n,l,d}\right] - 2\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\mathrm{a}_{n,l,d}\right]\right)\\
&-\frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\frac{1}{\eta^2_{l,d}}\right]\mathbb{E}\left[\gamma^2_{n,l,d}\right]\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}^T_{l,d}\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}^T_{n,l-1}\right]\right)\\
&-\frac{N}{2}\sum_{d=1}^{D_{L+1}}\mathbb{E}\left[\log\eta^2_{L+1,d}\right] - \frac{N}{2}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\log\eta^2_{l,d}\right] + \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\log(|\boldsymbol{S}_{n,l}|)\\
&+\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{1}{T}\left(\rho_{n,l,d}-\frac{1}{2}\right)\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\\
&-\frac{1}{2T^2}\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\mathbb{E}\left[\omega_{n,l,d}\right]\left(\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}^T_{l,d}\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}^T_{n,l-1}\right]\right)\right)\\
&-\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\rho_{n,l,d}\log\rho_{n,l,d} + (1-\rho_{n,l,d})\log(1-\rho_{n,l,d})\\
&+\frac{N}{|S|}\sum_{n\in S}\sum_{l=1}^{L}\sum_{d=1}^{D_l}\frac{\mathrm{a}^2_{n,l,d}}{2}\mathbb{E}\left[\omega_{n,l,d}\right] - \log(\cosh(\frac{\mathrm{a}_{n,l,d}}{2})) + C_a,
\end{aligned}
$$

where the normalizing constant is

$$C_a = -\frac{ND_{L+1}}{2}\log(2\pi) - N\sum_{l=1}^{L}D_l\log(2).$$

7. Find global variational parameters for which we recall the vector of natural parameters for $(\boldsymbol{W},\boldsymbol{b})$ is $(\boldsymbol{B}^{-1}\boldsymbol{m}^T, -\boldsymbol{B}^{-1}/2)$, and for $\boldsymbol{\eta}^2$ that is $(-\boldsymbol{\alpha}+\boldsymbol{1}, -\boldsymbol{\beta}^{-1})$. We are only updating the parameter $\boldsymbol{\alpha}$ in the first iteration of the algorithm as

$$
\begin{aligned}
\alpha_{l,d} &= \alpha^h_0 + \frac{N}{2},\ \text{for } l = 1,\dots,L,\ d = 1,\dots,D_l,\\
\alpha_{L+1,d} &= \alpha_0 + \frac{N}{2}\ \text{for } d = 1,\dots,D_{L+1}.
\end{aligned}
$$

We then find $\boldsymbol{\beta}$ via the intermediate variable $\hat{\beta}_{l,d}$:

8. For $l = 1, \ldots, L,\ d = 1, \ldots, D_l$ set

$$\hat{\beta}_{l,d} = \beta_0^h + \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\left(\mathbb{E}\left[\mathrm{a}_{n,l,d}\right] - \mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\right)^2$$
$$+ \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\mathbb{E}\left[a_{n,l,d}^2\right] - \mathbb{E}\left[\mathrm{a}_{n,l,d}\right]^2 + \mathbb{E}\left[\gamma_{n,l,d}\right]\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right)$$
$$- \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\mathbb{E}\left[\gamma_{n,l,d}\right]^2\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}^T\right]\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{l,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}^T\right]\right).$$

And for the final layer $l = L+1$ and $d = 1, \ldots, D_{L+1}$

$$\hat{\beta}_{L+1,d} = \beta_0 + \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\left(y_{n,d} - \mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}]\right)^2$$
$$+ \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\mathrm{Tr}\left(\mathbb{E}\left[\widetilde{\boldsymbol{W}}_{L+1,d}^T\widetilde{\boldsymbol{W}}_{L+1,d}\right]\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L}\widetilde{\boldsymbol{a}}_{n,L}^T\right]\right)$$
$$- \frac{1}{2}\frac{N}{|S|}\sum_{n\in S}\mathrm{Tr}\left(\mathbb{E}[\widetilde{\boldsymbol{W}}_{L+1,d}]^T\mathbb{E}[\widetilde{\boldsymbol{W}}_{L+1,d}]\mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}]\mathbb{E}[\widetilde{\boldsymbol{a}}_{n,L}^T]\right).$$

The update for $l = 1, \ldots, L+1,\ d = 1, \ldots, D_l$ is given by

$$\beta_{l,d}^{(t)} = \left((1-\ell_t)\times(\beta_{l,d}^{(t-1)})^{-1} + \ell_t \times \hat{\beta}_{l,d}^{-1}\right)^{-1}.$$

Similarly, the variational parameters $\boldsymbol{B}, \boldsymbol{m}$ of global variables $(\boldsymbol{b}, \boldsymbol{W})$ are obtained as a reparametrized linear combination of previous and intermediate updates. Specifically, for $l = 1, \ldots, L,\ d = 1, \ldots, D_l$ set

$$\hat{\boldsymbol{B}}_{l,d}^{-1} = \boldsymbol{D}_{l,d}^{-1} + \frac{N}{|S|}\sum_{n\in S}\left(\frac{1}{T^2}\mathbb{E}\left[\omega_{n,l,d}\right] + \mathbb{E}\left[\eta_{l,d}^{-2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\right)\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\widetilde{\boldsymbol{a}}_{n,l-1}^T\right],$$
$$\hat{\boldsymbol{B}}_{l,d}^{-1}\hat{\boldsymbol{m}}_{l,d}^T = \frac{N}{|S|}\sum_{n\in S}\mathbb{E}\left[\eta_{l,d}^{-2}\right]\mathbb{E}\left[\gamma_{n,l,d}\right]\mathbb{E}\left[\mathrm{a}_{n,l,d}\widetilde{\boldsymbol{a}}_{n,l-1}\right] + \frac{1}{T}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,l-1}\right]\left(\mathbb{E}\left[\gamma_{n,l,d}\right] - \frac{1}{2}\right).$$

For the final layer $l = L+1$ and $d = 1, \ldots, D_{L+1}$ set

$$\hat{\boldsymbol{B}}_{L+1,d}^{-1} = \boldsymbol{D}_{L+1,d}^{-1} + \mathbb{E}\left[\eta_{L+1,d}^{-2}\right]\frac{N}{|S|}\sum_{n\in S}\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L+1}\widetilde{\boldsymbol{a}}_{n,L+1}^T\right],$$
$$\hat{\boldsymbol{B}}_{L+1,d}^{-1}\hat{\boldsymbol{m}}_{L+1,d}^T = \mathbb{E}\left[\eta_{L+1,d}^{-2}\right]\left(\frac{N}{|S|}\sum_{n\in S}y_n\mathbb{E}\left[\widetilde{\boldsymbol{a}}_{n,L+1}\right]\right),$$

where for $l = 1, \ldots, L+1$ and $d = 1, \ldots, D_l$,

$$\boldsymbol{D}_{l,d}^{-1} = \mathrm{diag}\left(s_0^{-2}, \mathbb{E}\left[\tau_l^{-1}\right]\mathbb{E}\left[\psi_{l,d,1}^{-1}\right], \ldots, \mathbb{E}\left[\tau_l^{-1}\right]\mathbb{E}\left[\psi_{l,d,D_{l-1}}^{-1}\right]\right).$$

Then the updates $l = 1, \ldots, L+1$ and $d = 1, \ldots, D_l$, are given by

$$\boldsymbol{B}_{l,d}^{(t)} = \left((1-\ell_t) \times (\boldsymbol{B}_{l,d}^{(t-1)})^{-1} + \ell_t \times \hat{\boldsymbol{B}}_{l,d}^{-1}\right)^{-1},$$
$$\boldsymbol{m}_{l,d}^{(t)} = ((1-\ell_t)\boldsymbol{B}_{l,d}^{(t)}(\boldsymbol{B}_{l,d}^{(t-1)})^{-1}(\boldsymbol{m}_{l,d}^{(t-1)})^T + \ell_t \boldsymbol{B}_{l,d}^{(t)}\hat{\boldsymbol{B}}_{l,d}^{-1}\hat{\boldsymbol{m}}_{l,d}^T)^T.$$

We monitor the noisy estimate of the ELBO which is computed as in Appendix B.2 but with sums over $n = 1, \ldots, N$ replaced with the scaled sums over $n \in S$.

B.4 Experiments

B.4.1 Initialization schemes

Initialization plays an important role in the ability of Bayesian inference algorithms to effectively approximate the posterior. This is especially true in variational schemes for complex posteriors (such as for BNNs), which are only guaranteed to converge to a local optimum. We design two possible variations of random yet effective initialization schemes. To simplify the exposition, we describe the procedure in the case of Inverse Gamma shrinkage priors, for which $\lambda = 0$ and the selection of the scale parameters δ determines the level of shrinkage. Note that during the training step, we employ the expectation-maximization algorithm to set an optimal δ_{glob}, whilst the value of $\delta_{loc,l}$ remains fixed. To encourage more shrinkage for larger depth, we assume $\delta_{\text{glob}} \propto 1/\sqrt{L}$, and to encourage shrinkage for larger width set $\delta_{\text{loc},l} \propto 1/\sqrt{D_l}$. Given specified values of $\nu_{\text{loc}}, \nu_{\text{glob}}, \delta_{\text{loc}}, \delta_{\text{glob}}, \alpha_0^h, \alpha_0, \beta_0^h, \beta_0$, we first re-scale the shrinkage parameters to scale appropriately

$$\delta_{\text{glob}} = \frac{\delta_{\text{glob}}}{\sqrt{L}}, \delta_{\text{loc},l} = \frac{\delta_{\text{loc}}}{\sqrt{D_{l-1}}}, \nu_{\text{loc},l} = \nu_{\text{loc}},$$

and the initialization steps are:

1. Covariance for biases and weights: $\boldsymbol{B}_{l,d} = 0.01\boldsymbol{I}_{D_{l-1}+1}$ for $l = 1, \ldots, L+1, d = 1, \ldots D_l$.

2. Covariance for stochastic activation: $\boldsymbol{S}_{n,l} = 0.01\boldsymbol{I}_{D_l}$ for $n = 1, \ldots, N,\ l = 1, \ldots, L$.

3. Variational parameters for $\boldsymbol{\eta}$: Set $\alpha_{L+1,d} = \alpha_0, \alpha_{l,d} = \alpha_0^h$ and $\beta_{L+1,d} = \beta_0, \beta_{l,d} = \beta_0^h$.

4. Variational parameters for $\boldsymbol{\tau}, \boldsymbol{\psi}$:

$$\nu_{\text{loc},l,d,d'} = \nu_{\text{loc},l},\ \nu_{\text{glob},l} = \nu_{\text{glob}},$$
$$\delta_{\text{glob},l} \sim \sqrt{2(\nu_{\text{glob},l}-1)\text{IG}\left(\nu_{\text{glob},l}, \delta_{\text{glob}}\right)},$$
$$\delta_{\text{loc},l,d,d'} \sim \sqrt{2(\nu_{\text{loc},l,d,d'}-1)\text{IG}\left(\nu_{\text{loc},l,d,d'}, \delta_{\text{loc},l}\right)}.$$

5. Use Algorithm 8 to initialize the variational means of the weights and biases for all intermediate layers, and the variational means of the stochastic activations and the variational parameters of the binary activations.

6. Variational mean of the weights and biases for the last layer $\boldsymbol{m}_{L+1}$ is obtained as a solution of fitting D_{L+1} ridge regressions with inputs $\boldsymbol{z}_L$ and outputs $\boldsymbol{y}_d$.

B.4.2 Implementation details

When comparing the performance of our method to already existing ones, we implement the following model in Numpyro:

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{W}_{L+1}\mathrm{ReLU}(\boldsymbol{z}_L) + \boldsymbol{b}_L, \boldsymbol{\Sigma}\right), \text{ where } \quad \boldsymbol{\Sigma} \sim \mathrm{IG}(2, \sigma_y)\boldsymbol{I}_{D_{L+1}},$$
$$\boldsymbol{z}_l = \boldsymbol{W}_l\mathrm{ReLU}(\boldsymbol{z}_{l-1}) + \boldsymbol{b}_l,\ W_{l,d,d'} \sim \mathcal{N}\left(0, \frac{\sigma_W^2\gamma}{\sqrt{D_{l-1}}}\right), \quad b_{l,d} \sim \mathcal{N}(0, \sigma_b^2\gamma),$$

where $\boldsymbol{z}_{n,0} = \boldsymbol{x}_n,\ \gamma \sim \mathrm{IG}(2,1)$ and $l = 1, \ldots, L,\ d = 1, \ldots D_l,\ d' = 1, \ldots, D_{l-1}$. The choice of σ_y, σ_W and σ_b is made in accordance with α_0, s_0 and $\delta_{\mathrm{loc},l}$, respectively. For experiments with HMC, we use the No-U-Turn sampler, the number of warm-up samples is set to 500, and the number of samples is set to 1000. For experiments with HSBNN [Ghosh et al., 2019], the learning rate is set to 0.001 and the number of iterations to 10000. For experiments with mfVI, we use Adam optimizer with a learning rate set to 0.001 and a maximum number of iterations varying from 5000 to 20000, depending on the dataset and depth of the network. Additionally, we consider the Bayes by Backprop model of [Blundell et al., 2015] and adapt its Pytorch implementation from the publicly available repository [Javier, 2019]. For all experiments with BBB, we set the learning rate to 0.01, and the maximum number of epochs varies from 500 to 1000.

In all examples, we normalize the input but do not re-scale the output. Suppose that the data on which we evaluate the predictive performance consists of N points and the true target is $\boldsymbol{y}^*$, then recorded evaluation metrics are RMSE, NLL and EC and are computed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_n^N [(y_n^* - \mathbb{E}[y_n^o])^2]},$$
$$\mathrm{NLL} = \frac{1}{N}\sum_n^N \log \mathcal{N}(y_n^* \mid \mathbb{E}[y_n^o], \mathrm{Var}(y_n^o)),$$
$$\mathrm{EC} = \frac{\#\{\boldsymbol{y}^* \in [q^o_{0.025}, q^o_{0.975}]\}}{N}.$$

where the predicted observations are $\boldsymbol{y}^o$ and the corresponding quantiles are denoted as q^o. When computing quantiles to obtain empirical coverage and illustrating the uncertainty in Section 4.4 and below in Appendix B.4.3, we rely on the Gaussian approximation.

Algorithm 8 Initialization scheme for VBNN.

Require: Training inputs $\boldsymbol{x}_n$, $n = 1, \ldots, N$; choice of mode *laplace* or *spike-slab*

$\boldsymbol{z}_{n,0} = \boldsymbol{x}_n$
for $l = 1 \ldots L$, **do**
  set $\Delta = 0.05 * (\max(\boldsymbol{z}_{n,l-1}) - \min(\boldsymbol{z}_{n,l-1}))$
  for $d = 1 \ldots D_l$ **do**
    if *laplace* **then**

$$m^W_{l,d,d'} \sim \text{Laplace}\left(0, \sqrt{\frac{2}{D_{l-1}}}\right),$$

    end if
    if *spike-slab* **then**

$$m^W_{l,d,d'} \sim \pi\mathcal{N}\left(0, \frac{2}{\sqrt{D_{l-1}}}\right) + (1-\pi)\delta_0, \text{ where } \pi = \frac{1}{1+\sqrt{D_{l-1}}},$$

    end if

$\boldsymbol{s} = (s_1, \ldots, s_{D_{l-1}})$, where $s_{d'} \sim \text{Unif}([\min(z_{n,l-1,d'}) - \Delta_{d'}, \max(z_{n,l-1,d'}) + \Delta_{d'}])$,
$m^b_{l,d} = -\boldsymbol{m}^W_{l,d}\boldsymbol{s}$, $\boldsymbol{m}_{l,d} = \left(m^b_{l,d}, \boldsymbol{m}^W_{l,d}\right)$,

  end for

$$\rho_{n,l,d} = \sigma\left(\frac{m^b_{l,d} + \boldsymbol{m}^W_{l,d}\boldsymbol{z}_{n,l-1}}{T}\right) \quad d = 1, \ldots, D_l,$$

$\boldsymbol{M}_{n,l} = \boldsymbol{m}^W_l \odot \boldsymbol{\rho}_{n,l}\mathbf{1}^T_{D_l}$, where by $\mathbf{1}$ we denote a vector of ones,
$\boldsymbol{t}_{n,l} = m^b_l \odot \boldsymbol{\rho}_{n,l}$,
$\boldsymbol{z}_{n,l} = \boldsymbol{M}_{n,l}\boldsymbol{z}_{n,l-1} + \boldsymbol{t}_{n,l}$,

end for
Ensure: $\boldsymbol{M}_{n,l}, \boldsymbol{t}_{n,l}, \boldsymbol{m}_{l,d}$ for $l = 1, \ldots, L, d = 1, \ldots, D_l$ and $\boldsymbol{z}_L$.

B.4.3 Supplementary material to the diabetes example

Figure B.1 supplements Table 4.3 and the diabetes example in Section 4.4.2. Here, in the case of VBNN, BBB, GVBNN, HSBNN, HMC and mfVI models, we provide the uncertainty of the observations, and in the case of the LassoCV, we provide residual standard deviation. Additionally, we illustrate the sparse prediction and the uncertainty obtained from sparse weights of the VBNN, which largely coincide with the original prediction and uncertainty estimates. Whilst the coverage estimates for observations of VBNN, BBB, and GVBNN are comparable, the mfVI and less HSBNN underestimate the uncertainty and provide a lower coverage for observations.

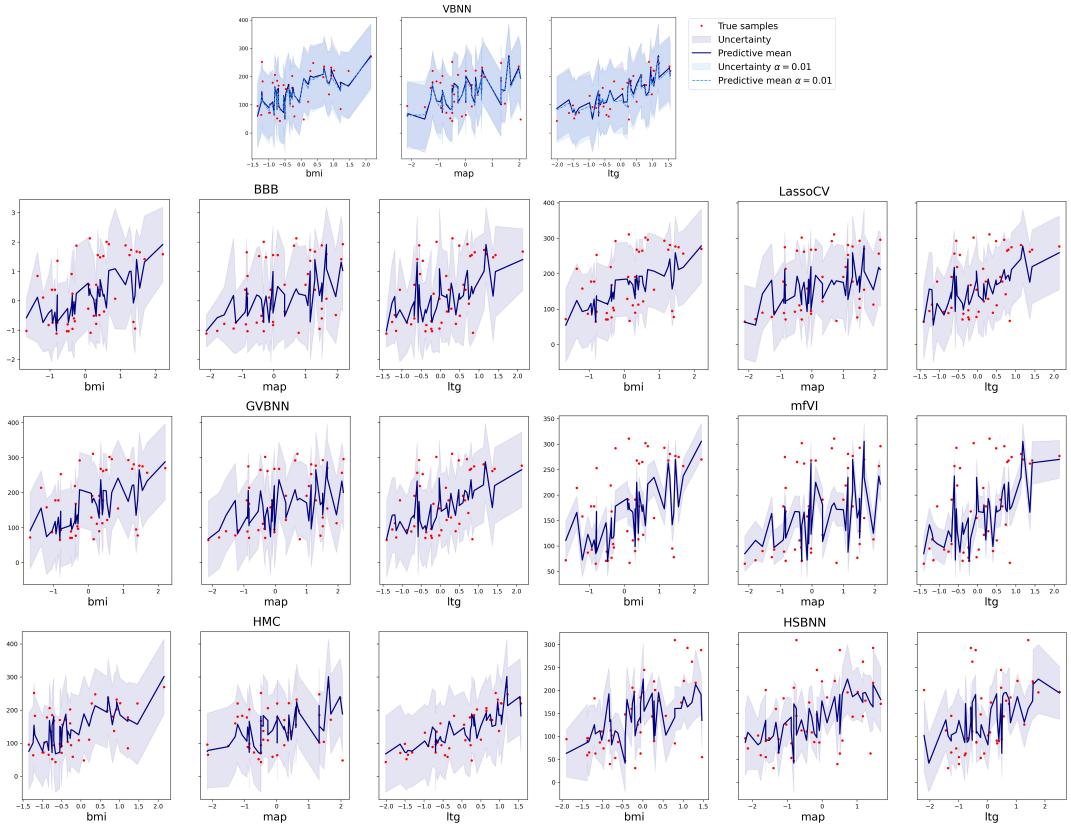


Figure B.1: Predictive mean and the uncertainty estimates for the observations for three of the predictors with considerable contribution.

B.4.4 Supplementary information on the datasets

Boston housing [Harrison and Rubinfeld, 1978]: $n = 506, p = 13$, the predictors are per capita crime rate by town, the proportion of residential land zoned for lots over 25,000 sq.ft., the proportion of non-retail business acres per town, Charles River dummy variable, nitrite oxides concentration, average number of rooms per dwelling, the proportion of owner-occupied, units built before 1940, weighted distances to five Boston employment centres, index of accessibility to radial highways, full-value property-tax rate, the pupil-teacher ratio by town, the

quantitative measure of systemic racism as a factor in house pricing, lower status of the population; the response of interest is the median value of owner-occupied homes. The Boston housing dataset is among the most popular pip available datasets, and with respect to variable selection, it was considered in e.g. [Schäfer and Chopin, 2013].

Energy [Tsanas and Xifara, 2012]: $n = 768, p = 8$, the predictors are relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution, and the task is to predict the heating load of residential buildings.

Yacht dynamics [J. et al., 2013]: $n = 308, p = 6$, the predictors are long position, prismatic coefficient, length-displacement ratio, bean-draught ratio, length-bean ratio and froude number, and the task is to model the residuary resistance per unit weight of displacement for a yacht hull.

Concrete compressive strength [Yeh, 2007]: $n = 1030, p = 8$, the predictors are cement, furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate and the age of testing, and the response variable is the compressive strength of concrete. This is also considered from the variable selection perspective in several works including [Griffin, 2024, Schäfer and Chopin, 2013].

Concrete slump test [Yeh, 2009]: $n = 103, p = 7$, the predictors are concrete ingredients, namely cement, furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate and the task is to predict slump of concrete.

B.4.5 Supplementary material to the UCI datasets experiments

Table B.1 supplements Figure 4.12, Figure 4.13 and the experiments described in Section 4.4.3.

B.5 Review of relevant distributions

B.5.1 Generalized Inverse Gaussian

The Generalized Inverse Gaussian has density:

$$p(x|\nu, \delta, \lambda) = \frac{(\lambda/\delta)^\nu}{2K_\nu(\lambda\delta)} x^{\nu-1} \exp\left(-\frac{1}{2}(\delta^2/x + \lambda^2 x)\right),$$

where $K_\nu()$ is the modified Bessel function of the second kind. The GIG prior requires $\nu > 0$ if $\delta = 0$ and $\nu < 0$ if $\lambda = 0$ for a proper prior. Then the expectations arising in computations throughout Chapter 4 are:

$$\mathbb{E}[x] = \frac{\delta K_{\nu+1}(\lambda\delta)}{\lambda K_\nu(\lambda\delta)},$$
$$\mathbb{E}\left[\frac{1}{x}\right] = \frac{\lambda K_{\nu+1}(\lambda\delta)}{\delta K_\nu(\lambda\delta)} - \frac{2\nu}{\delta^2}.$$

Often, it is sensible to consider special cases of the GIG, which include:

Table B.1: RMSE, NLL and Coverage for UCI datasets.

		Dataset				
Metric	Method	Slump	Yacht	Boston	Energy	Concrete
RMSE	4SVIBNN	7.15 ± 1.5	$3.57 \pm .8$	$3.38 \pm .9$	$1.62 \pm .2$	$7.15 \pm .6$
	4VBNN	7.01 ± 1.2	$1.23 \pm .3$	$3.21 \pm .6$	$1.1 \pm .2$	$6.71 \pm .6$
	SVBNN	7.36 ± 1.62	5.57 ± 1.14	$3.76 \pm .92$	$2.41 \pm .49$	$8.43 \pm .75$
	VBNN	7.37 ± 1.4	2.47 ± 1.1	$3.47 \pm .8$	$1.37 \pm .3$	$7.67 \pm .9$
	GVBNN	7.64 ± 1.21	4.88 ± 2.66	$4.02 \pm .88$	$2.5 \pm .42$	$7.84 \pm .68$
	mfVI	7.9 ± 1.7	$1.61 \pm .36$	$3.29 \pm .6$	$2.27 \pm .25$	$6.11 \pm .5$
	BBB	7.33 ± 1.94	$1.45 \pm .6$	$3.46 \pm .96$	$2.65 \pm .26$	$6.48 \pm .61$
	HMC	$7. \pm 1.24$	$.56 \pm .13$	$2.42 \pm .47$	$.3 \pm .07$	$4.01 \pm .78$
	HSBNN	6.41 ± 1.28	$1.2 \pm .23$	$2.92 \pm .55$	$.6 \pm .09$	$5.21 \pm .56$
NLL	4SVBNN	3.42 ± 0.2	$2.74 \pm .2$	$2.61 \pm .2$	$1.94 \pm .1$	$3.38 \pm .07$
	4VBNN	$3.39 \pm .2$	$1.97 \pm .2$	$2.57 \pm .13$	$1.63 \pm .16$	$3.33 \pm .06$
	SVBNN	$3.47 \pm .29$	$3.13 \pm .27$	$2.78 \pm .31$	$2.31 \pm .22$	3.56 ± 0.11
	VBNN	3.46 ± 0.2	2.25 ± 0.4	$2.69 \pm .26$	$1.75 \pm .21$	$3.5 \pm .12$
	GVBNN	3.47 ± 0.17	$2.86 \pm .59$	$2.82 \pm .22$	$2.35 \pm .15$	$3.48 \pm .09$
	mfVI	$3.77 \pm .5$	$1.96 \pm .09$	$2.61 \pm .22$	$2.28 \pm .09$	$3.57 \pm .2$
	BBB	6.23 ± 2.76	$1.69 \pm .14$	$2.47 \pm .16$	$2.08 \pm .15$	3.21 ± 0.13
	HMC	$3.41 \pm .24$	$.87 \pm .1$	$2.28 \pm .18$	$.23 \pm .44$	$2.74 \pm .27$
	HSBNN	5.02 ± 1.79	$1.31 \pm .18$	5.01 ± 1.24	$1.08 \pm .2$	$4.32 \pm .69$
Coverage	4SVBNN	$.95 \pm .06$	$.98 \pm .03$	$.97 \pm .03$	$.98 \pm .01$	$.97 \pm .01$
	4VBNN	$.94 \pm .04$	$.99 \pm .02$	$.97 \pm .02$	$.99 \pm .0$	$.97 \pm .02$
	SVBNN	$.91 \pm .08$	$.93 \pm .02$	$.94 \pm .03$	$.91 \pm .04$	$.93 \pm .03$
	VBNN	$.92 \pm .06$	$.96 \pm .01$	$.95 \pm .03$	$.98 \pm .02$	$.94 \pm .02$
	GVBNN	$.96 \pm .04$	$.95 \pm .04$	$.96 \pm .02$	$.93 \pm .04$	$.95 \pm .02$
	mfVI	$.78 \pm .1$	$.96 \pm .03$	$.96 \pm .01$	$.95 \pm .03$	$.8 \pm .04$
	BBB	$.75 \pm .12$	$1. \pm .0$	$.97 \pm .02$	$.99 \pm .0$	$.97 \pm .02$
	HMC	$.9 \pm .08$	$.98 \pm .02$	$.96 \pm .02$	$.95 \pm .03$	$.94 \pm .03$
	HSBNN	$.67 \pm .14$	$.94 \pm .05$	$0.62 \pm .08$	$.89 \pm .04$	$.71 \pm .06$

1. Inverse Gamma: when $\lambda = 0$, the GIG reduces to the IG with density:

$$p(x|\nu, \delta) = \frac{2^{\nu}}{\delta^{2\nu}\Gamma(-\nu)}(1/x)^{-\nu+1} \exp\left(-\frac{\delta^2}{2x}\right),$$

where $\nu < 0$ and $\delta > 0$. This can also be re-written in terms of the more standard parametrization of the IG:

$$p(x|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}(1/x)^{\alpha+1} \exp\left(-\frac{\beta}{x}\right),$$

where $\alpha = -\nu > 0$ and $\beta = \delta^2/2 > 0$. Note that if $w \sim \mathcal{N}(0, \tau)$ and $\tau \sim \text{IG}(\alpha, \beta)$, this implies a marginal student t-prior on w with degrees of freedom dof $= 2\alpha = -2\nu$ and scale $s = \sqrt{\beta/\alpha} = \delta/\sqrt{-2\nu}$. For example, setting $\nu = -1.5$ would correspond to dof $= 3$ and $\nu = -2.5$ is equivalent

to dof $= 5$.

The relevant expectations for the VI updates and ELBO computation include:

$$\mathbb{E}[x] = \frac{\beta}{\alpha - 1} = \frac{-\delta^2}{2\nu + 2},$$
$$\mathbb{E}\left[\frac{1}{x}\right] = \frac{\alpha}{\beta} = \frac{-2\nu}{\delta^2},$$

where ψ is the logarithmic derivative of the gamma function (a.k.a. digamma function).

2. Gamma: when $\delta^2 = 0$, the GIG reduces to the Gamma with density:

$$p(x|\nu, \lambda) = \frac{\lambda^{2\nu}}{2^\nu \Gamma(\nu)} x^{\nu-1} \exp(-\frac{\lambda^2}{2} x),$$

where $\nu > 0$, rewriting in the standard parametrization with $\alpha = \nu$ and $\beta = \lambda^2/2$:

$$p(x|\alpha, \beta) = \beta^\alpha \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x),$$

where $\alpha = \nu > 0$ and $\beta = \lambda^2/2 > 0$. Similarly, the relevant expectations are:

$$\mathbb{E}[x] = \frac{\alpha}{\beta} = \frac{2\nu}{\lambda^2},$$
$$\mathbb{E}\left[\frac{1}{x}\right] = \frac{\beta}{\alpha - 1} = \frac{\lambda^2}{2(\nu - 1)}.$$

Note that if $w \sim \mathcal{N}(0, \tau)$ and $\tau \sim \text{Gam}(1, \beta)$, this implies a marginal Laplace prior on w (i.e. Bayesian Lasso [Park and Casella, 2008]) with scale $s = 1/\sqrt{2\beta} = 1/\lambda$.

3. Inverse Gaussian (IGaus): when $\nu = -1/2$, the GIG reduces to the Inverse Gaussian with density:

$$p(x|\delta, \lambda) = \frac{\delta}{\sqrt{2\pi x^3}} \exp\left(-\frac{(\lambda x - \delta)^2}{2x}\right),$$

where setting $\alpha = \delta/\lambda > 0$ and $\beta = \delta^2 > 0$ we derive

$$p(x|\alpha, \beta) = \left(\frac{\beta}{2\pi x^3}\right)^{\frac{1}{2}} \exp\left(\frac{-\beta(x - \alpha)^2}{2\alpha^2 x}\right).$$

The relevant expectations for the VI updates and ELBO computation in-

clude:

$$\mathbb{E}\left[x\right] = \alpha = \frac{\delta}{\lambda},$$
$$\mathbb{E}\left[\frac{1}{x}\right] = \frac{1}{\alpha} + \frac{1}{\beta} = \frac{\lambda}{\delta} + \frac{1}{\delta^2}.$$

Note that if $w \sim \mathcal{N}(0, \tau)$ and $\tau \sim \text{IGaus}(\alpha, \beta)$, the marginal distribution is of the form [Caron and Doucet, 2008]:

$$\begin{aligned} p(w_k) &= \frac{1}{\pi\alpha}\left(\frac{\beta}{\beta + w_k^2}\right)^{\frac{1}{2}} \exp\left(\frac{\beta^{\frac{1}{2}}}{\alpha}\right) K_1\left(\frac{(\beta + w_k^2)^{\frac{1}{2}}}{\alpha}\right) \\ &= \frac{\lambda}{\pi}\exp(\lambda)\left(\delta^2 + w_k^2\right)^{-\frac{1}{2}} K_1\left(\frac{\lambda}{\delta}\left(\delta^2 + w_k^2\right)^{\frac{1}{2}}\right). \end{aligned}$$

B.5.2 EM update for different cases of global-local priors

As discussed above, the special cases of the GIG include Inverse Gamma, Gamma and Inverse Gaussian distributions, we derive the EM updates in each of the special cases of priors:

1. Inverse Gamma: when the global shrinkage parameter has an Inverse Gamma distribution, then

$$\delta_{\text{glob}} = \arg\max\left(\delta_{\text{glob}}^2 \sum_{l=1}^{L+1} \frac{\nu_{\text{glob},l}}{\delta_{\text{glob},l}^2} - 2(L+1)\nu_{\text{glob}}\log(\delta_{\text{glob}})\right),$$
$$\delta_{\text{glob}} = ((L+1)\nu_{\text{glob}})^{\frac{1}{2}}\left(\sum_{l=1}^{L+1}\frac{\nu_{\text{glob},l}}{\delta_{\text{glob},l}^2}\right)^{-\frac{1}{2}}.$$

2. Gamma: similarly, when the global shrinkage parameter is Gamma:

$$\lambda_{\text{glob}} = \arg\max\left(4(L+1)\nu_{\text{glob}}\log(\lambda_{\text{glob}}) - \lambda_{\text{glob}}^2 \sum_{l=1}^{L+1}\frac{\delta_{\text{glob},l} K_{\nu_{\text{glob},l}+1}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}{\lambda_{\text{glob},l} K_{\nu_{\text{glob},l}}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}\right),$$
$$\lambda_{\text{glob}} = (2(L+1)\nu_{\text{glob}})^{\frac{1}{2}}\left(\sum_{l=1}^{L+1}\frac{\delta_{\text{glob},l} K_{\nu_{\text{glob},l}+1}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}{\lambda_{\text{glob},l} K_{\nu_{\text{glob},l}}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}\right)^{-\frac{1}{2}}.$$

3. Inverse Gaussian: if the global shrinkage parameter is Inverse Gaussian, then

$$\lambda_{\text{glob}} = \arg\max\left(2(L+1)\lambda_{\text{glob}}\delta_{\text{glob}} - \lambda_{\text{glob}}^2 \sum_{l=1}^{L+1}\frac{\delta_{\text{glob},l} K_{\nu_{\text{glob},l}+1}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}{\nu_{\text{glob},l} K_{\nu_{\text{glob},l}}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}\right),$$
$$\lambda_{\text{glob}} = 2(L+1)\delta_{\text{glob}}\left(\sum_{l=1}^{L+1}\frac{\delta_{\text{glob},l} K_{\nu_{\text{glob},l}+1}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}{\nu_{\text{glob},l} K_{\nu_{\text{glob},l}}(\lambda_{\text{glob},l}\delta_{\text{glob},l})}\right)^{-1}.$$

B.6 Improving and adapting VBNN

B.6.1 Horseshoe.

Variational bow tie neural network has N-GIG priors on the weights (see Section 4.2.2), alternatively, one could employ horseshoe priors, known to be effective for sparse Bayesian estimation [Carvalho et al., 2009, Polson and Scott, 2010]. Similar to N-GIG priors, the horseshoe priors on the weights are represented by a scale mixture of normals with global and local parameters, so that the global parameter shrinks all the weights towards zero and the local parameter compensates for the effect of shrinking for large enough weights. For $l = 1, \dots, L,\ d = 1, \dots, D_l, d' = 1, \dots, D_{l-1}$ the horseshoe priors are defined as

$$\begin{aligned} W_{l,d,d'} | \psi_{l,d}, \tau_l &\sim \mathcal{N}\left(0, \tau_l^2 \psi_{l,d}^2\right), \\ \tau_l &\sim C^+\left(0, \delta_{\text{glob}}\right), \\ \psi_{l,d} &\sim C^+\left(0, \delta_{\text{loc}}\right), \end{aligned}$$

where τ_l is the global shrinkage parameter for layer l with scale δ_{glob}, $\psi_{l,d,d'}$ is the local shrinkage parameter with with scale δ_{loc} and C^+ denotes a half-Cauchy distribution defined by

$$C^+(x|0,\delta) = \frac{2}{\pi\delta(1 + x^2/\delta^2)}.$$

The model with horseshoe priors is suitable for the coordinate ascent algorithm as long as one introduces auxiliary variables to replace each half-Cauchy random variable with the product of two Inverse-Gamma variables [Wand et al., 2011]. Namely, the half-Cauchy prior is equivalent to

$$\begin{aligned} \psi_{l,d}^2 \mid \Psi_{l,d} \sim \text{IG}\left(0.5, \Psi_{l,d}^{-1}\right),\ \Psi_{l,d} \sim \text{IG}\left(0.5, \delta_{\text{loc}}^{-2}\right), \\ \tau_l^2 \mid \text{T}_l \sim \text{IG}\left(0.5, \text{T}_l^{-1}\right),\ \text{T}_l \sim \text{IG}\left(0.5, \delta_{\text{glob}}^{-2}\right). \end{aligned}$$

Additionally, it is worth investigating whether we could employ the CAVI algorithm with the regularized version of horseshoe priors ("ponyshoe"), which is known for better performance than the classical horseshoe, especially when the larger coefficients are weakly identified by the data [Piironen and Vehtari, 2017a,b]; e.g. in the context of variational inference, the regularized horseshoe has been considered in [Ghosh et al., 2018, 2019].

B.6.2 Different classes of models.

We briefly address two of the popular deep learning tasks: classification and data generation [Barber, 2012] in the context of the bow tie neural network. While the original VBNN models have continuous output $\boldsymbol{y} \in \mathbb{R}^{D_{L+1}}$, it can be easily adapted to classification tasks by using the softmax transformation, which is a generalization of the logistic (sigmoid) function for problems with multiple classes. The use of softmax in classical deep neural networks is associated with miscalibrated uncertainties; however, the problem is mitigated when Bayesian

inference is employed [Kristiadi et al., 2020]. Assume the settings of Section 4.2.1 and VBNN with continuous input $\boldsymbol{x}_n \in \mathbb{R}^{D_0}$ and $\boldsymbol{y}_n \in 1, \ldots, K$, then

$$\mathbb{P}(y_n = k, |\boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Sigma}) = \frac{\exp\left(\boldsymbol{W}_{L+1,k}\boldsymbol{z}_{n,L} + b_{L,k}\right)}{\sum_{i=1}^{K} \exp\left(\boldsymbol{W}_{L+1,i}\boldsymbol{z}_{n,L} + b_{L,i}\right)},$$

which is amenable to the Polya-Gamma data augmentation trick [Chen et al., 2013, Durante and Rigon, 2019, Polson et al., 2013] and thus, the CAVI algorithm can be employed to obtain the variational approximation.

Deep latent variable models represent another class of widely used models aimed at generating data. These models are used in a variety of tasks, including image [Zhang et al., 2019], text [Bowman et al., 2016], audio [Oord et al., 2016] and molecule [Liu et al., 2018] generation. Recall that in Section 2.4.4 we encountered variational auto-encoders, which are an example of such models [Kingma and Welling, 2014, Rezende et al., 2014]. On a similar line, given the data $\mathcal{D} = \{\boldsymbol{y}_n\}_n^N$, one could generate $\boldsymbol{y}$ from VBNN by assuming that $\boldsymbol{x} \sim \mathcal{N}(\mu_x, \boldsymbol{\Sigma}_x)$ and considering the corresponding variational posterior.