

## RESEARCH ARTICLE

# Bayesian Sparsification for Deep Neural Networks With Bayesian Model Reduction

DIMITRIJE MARKOVIĆ<sup>1</sup>, KARL J. FRISTON<sup>2,3</sup>, AND STEFAN J. KIEBEL<sup>1,4</sup><sup>1</sup>Chair of Cognitive Computational Neuroscience, Technische Universität Dresden, 01069 Dresden, Germany<sup>2</sup>VERSES AI Research Lab, Los Angeles, CA 90016, USA<sup>3</sup>Queen Square Institute of Neurology, University College London, WC1N 3AR London, U.K.<sup>4</sup>Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, 01069 Dresden, Germany

Corresponding author: Dimitrije Marković (dimitrije.markovic@tu-dresden.de)

This work was funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy—EXC 2050/1—Project number 390696704—Cluster of Excellence, Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universität Dresden.

**ABSTRACT** Deep learning's immense capabilities are often constrained by the complexity of its models, leading to an increasing demand for effective sparsification techniques. Bayesian sparsification for deep learning emerges as a crucial approach, facilitating the design of models that are both computationally efficient and competitive in terms of performance across various deep learning applications. The state-of-the-art – in Bayesian sparsification of deep neural networks – combines structural shrinkage priors on model weights with an approximate inference scheme based on stochastic variational inference. However, model inversion of the full generative model is exceptionally computationally demanding, especially when compared to standard deep learning of point estimates. In this context, we advocate for the use of Bayesian model reduction (BMR) as a more efficient alternative for pruning of model weights. As a generalization of the Savage-Dickey ratio, BMR allows a post-hoc elimination of redundant model weights based on the posterior estimates under a straightforward (non-hierarchical) generative model. Our comparative study highlights the advantages of the BMR method relative to established approaches, which are based on hierarchical horseshoe priors over model weights. We illustrate the potential of BMR across various deep learning architectures, from classical networks like LeNet to modern frameworks such as Vision Transformers and MLP-Mixers.

**INDEX TERMS** Bayesian model reduction, stochastic variational inference, deep neural networks.

## I. INTRODUCTION

Bayesian deep learning integrates the principles of Bayesian methodology with the objectives of deep learning, facilitating the training of expansive parametric models tailored for classifying and generating intricate audio-visual data, including images, text, and speech [1], [2], [3]. Notably, the Bayesian approach frames the challenge of model optimization as an inference problem. This perspective is especially apt for scenarios necessitating decision-making under uncertainty [4], [5]. As a result, Bayesian formulations in deep learning

have proven advantageous in various respects, offering enhancements in generalization [6], accuracy, calibration [7], [8], and model compression [9].

These functional enhancements are intrinsically tied to judiciously chosen structural priors [10]. The priors, integral to the probabilistic generative model, scaffold the architecture of the network, thereby reducing the data required for the inference of optimal parametric solutions. Recent studies have highlighted the efficacy of hierarchical shrinkage priors over model weights, a specific category of structural priors, in achieving highly-sparse network representations [9], [11], [12], [13]. Sparse representations not only reduce redundancy but also evince additional performance benefits. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos<sup>1</sup>.

the adoption of shrinkage priors in all deep learning models presents a conundrum: the ballooning space of latent parameters and the diminishing scalability of prevailing approximate inference schemes [7], [14], [15], [16].

In line with ongoing research on scalable Bayesian inference, we introduce an approximate inference scheme rooted in Bayesian model reduction (BMR). In essence, BMR extends the foundational principles of the Savage-Dickey Density Ratio method [17]. BMR is typically conceptualized as a combinatorial model comparison framework, enabling swift estimations of model evidence across an extensive array of models, that differ in their prior assumptions, to identify the most probable one. Originally conceived for model comparison within the dynamical causal modeling framework [18], [19], the scope of BMR has since broadened. Subsequent works expanded its methodology [20], [21], [22] and adapted it for structure learning [23]. More recently, BMR has found applications in Bayesian nonlinear regression and classification tasks using Bayesian neural networks with variance backpropagation [24], [25].

The BMR method is intimately connected with the spike-and-slab prior, a type of shrinkage prior [26]. Intriguingly, this specific structured shrinkage prior has parallels with Dropout regularization [11]. Such an association spurred researchers in Bayesian deep learning to formulate sparsification methods based on a different type of shrinkage prior—the hierarchical horseshoe prior [27]—as a tool for automated depth determination. Subsequent studies suggested that merging horseshoe priors with structured variational approximations yields robust, highly sparse representations [13]. The allure of continuous shrinkage priors (e.g., horseshoe priors) stems from the computational challenges associated with model inversion reliant on spike-and-slab priors [11], [27]. However, continuous shrinkage priors necessitate a considerably more expansive parameter space, to represent the approximate posterior, compared to optimizing neural networks using the traditional point estimate method.

In this work, we reexamine the spike-and-slab prior within the framework of BMR-based sparsification, highlighting its efficiency. Notably, this approach circumvents the need to expand the approximate posterior beyond the conventional fully factorised mean-field approximation, making it more scalable than structured variational approximations [13]. In this light, BMR can be seen as a layered stochastic and black-box variational inference technique, which we term *stochastic BMR*. We subject the stochastic BMR to rigorous validation across various image classification tasks and network architectures, including LeNet-5 [28], Vision Transformers [29], and MLP-Mixers [30].

Central to our study is an empirical comparison of stochastic BMR with methods anchored in hierarchical horseshoe priors. Through multiple metrics - from Top-1 accuracy to expected calibration error and negative log-likelihood - we establish the competitive performance of stochastic BMR. We argue its computational efficiency, and remarkable sparsification rate. These findings position BMR

as an appealing choice that can enhance the scalability and proficiency of contemporary deep learning networks across diverse machine learning challenges, extending beyond provided computer vision examples. We conclude with a discussion on potential avenues of future research that could further facilitate BMR based pruning of deep neural networks.

## II. METHODS

In this section, we first describe the methods and techniques used in our research to address the problem of efficient Bayesian sparsification of deep neural networks. We provide a detailed overview of our approach, starting with Bayesian deep learning and variational inference methods, followed by the formulation of the Bayesian model reduction (BMR), Bayesian neural networks with shrinkage priors, and the description of corresponding approximate posterior.

### A. BAYESIAN DEEP LEARNING

The core idea of Bayesian deep learning consists of treating the model parameters as random variables, hence casting the optimization problem of classical deep learning as an inference problem, where one computes the posterior distribution of model parameters given the data. Mathematically, this can be expressed as:

$$p(\mathcal{W}|\mathcal{D}) \propto p(\mathcal{W}) p(\mathcal{D}|\mathcal{W}) = p(\mathcal{W}) \prod_{i=1}^n p(y_i|\mathcal{W}, \mathbf{x}_i)$$

where  $\mathcal{W}$  denotes the model parameters,  $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$  denotes the dataset,  $p(\mathcal{W}|\mathcal{D})$  is the posterior distribution over model parameters,  $p(\mathcal{D}|\mathcal{W})$  is the likelihood of the data given the parameters,  $p(\mathcal{W})$  is the prior distribution of the parameters.

A probabilistic formulation of the deep learning task, enhances the model's ability to quantify uncertainty and improves generalization in a range of deep learning applications [31]. A key reason for these improvements is the implicit bias introduced to the model-parameters  $\mathcal{W}$  in the form of a prior distribution. The choice of the prior distribution is crucial for optimal task performance, and a prior assumption of structural sparsity is essential for inferring sparse representations of over-parameterised models, such as deep neural networks.

In a general (nonlinear) regression problem, we model the relationship between predictors  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  and target variables  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  using a likelihood distribution from an exponential family as

$$\begin{aligned} y_i &\sim p(\mathbf{y}|\mathcal{W}, \mathbf{x}_i) \\ &= h(\mathbf{y}) \exp \left[ \boldsymbol{\eta}(\mathbf{f}(\mathcal{W}, \mathbf{x}_i)) \cdot \mathbf{T}(\mathbf{y}) - A(\mathbf{f}(\mathcal{W}, \mathbf{x}_i)) \right]. \end{aligned} \quad (1)$$

Functions  $h(\cdot)$ ,  $\boldsymbol{\eta}(\cdot)$ ,  $\mathbf{T}(\cdot)$ ,  $A(\cdot)$  are known and selected depending on the task.

The choice of the likelihood function establishes a link between the optimization problem in classical deep learning

and the inference problem in Bayesian deep learning. For instance, the cross-entropy loss function, widely used for classification tasks, corresponds to the negative log likelihood of a Bernoulli or multinomial (categorical) distribution.

Finally, the mapping  $f(\mathcal{W}, \mathbf{x}_i)$ , in eq. (1), represents a generic deep neural network of depth  $L$ , defined as

$$\begin{aligned}\mathcal{W} &= (\mathbf{W}_1, \dots, \mathbf{W}_L) \\ \mathbf{h}_i^0 &= \mathbf{x}_i \\ \mathbf{h}_i^l &= \mathbf{g} \left( \mathbf{W}_l \cdot \left[ \mathbf{h}_i^{l-1}; 1 \right] \right) \\ f(\mathcal{W}, \mathbf{x}_i) &= \mathbf{W}_L \cdot \left[ \mathbf{h}_i^{L-1}; 1 \right]\end{aligned}$$

## B. VARIATIONAL INFERENCE

Given a joint density of latent variables, represented as  $\mathbf{z} = (z_1, \dots, z_k)$ , and a dataset of  $n$  observations  $\mathcal{D} = (y_1, \dots, y_n)$  we can express the joint density, that is, the generative model, as

$$p(\mathcal{D}, \mathbf{z}) = p(\mathbf{z}) p(\mathcal{D}|\mathbf{z}).$$

The posterior density is then obtained, following the Bayes rule, as

$$p(\mathbf{z}|\mathcal{D}) \propto p(\mathbf{z}) p(\mathcal{D}|\mathbf{z}). \quad (2)$$

For complex generative models, direct inference as described above becomes computationally prohibitive. To circumvent this, we approximate the exact posterior  $p(\mathbf{z}|\mathcal{D})$ , constraining it to a distribution  $q(\mathbf{z})$  that belongs to a named distribution family  $\mathcal{Q}$ . We then seek  $q^*(\mathbf{z}) \in \mathcal{Q}$ , an approximate solution that minimizes the following Kullback-Leibler divergence [32].

$$q^*(\mathbf{z}) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} D_{KL} \left( q(\mathbf{z}) || p(\mathbf{z}|\mathcal{D}) \right) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} F[q],$$

where  $F[q]$  stands for the variational free energy (VFE), defined as

$$F[q] = E_{q(\mathbf{z})} \left[ \ln q(\mathbf{z}) - \ln p(\mathcal{D}, \mathbf{z}) \right].$$

VFE serves as an upper bound on the marginal log-likelihood

$$F[q] = D_{KL} \left( q(\mathbf{z}) || p(\mathbf{z}|\mathcal{D}) \right) - \ln p(\mathcal{D}) \geq -\ln p(\mathcal{D}).$$

As KL-divergence is always greater or equal to zero, minimizing VFE brings the approximate solution as close as possible to the true posterior, without having to compute the exact posterior.

The most straightforward way to obtain the approximate posterior  $q^*(\mathbf{z})$ , is to minimize the VFE along its negative gradient:

$$\dot{\boldsymbol{\phi}} = -\nabla_{\boldsymbol{\phi}} F[q]$$

where  $\boldsymbol{\phi}$  signifies the parameters of the approximate posterior  $q_{\boldsymbol{\phi}}(\mathbf{z}) = q(\mathbf{z}|\boldsymbol{\phi})$ . Thus, variational inference reframes the inference problem highlighted in eq. (2) as an optimization problem [33].

## C. STOCHASTIC AND BLACK-BOX VARIATIONAL INFERENCE

*Stochastic variational inference* (SVI) improves the computational efficiency of gradient descent by approximating the variational free energy using a subset— $\mathcal{K}_i = (y_{s_1^i}, \dots, y_{s_k^i})$ ;  $k \ll n$ —of the entire data set  $\mathcal{D}$ . This approach fosters a stochastic gradient descent (SGD) mechanism, capable of managing large datasets [34]. Crucially, at every iteration step  $i$  of the SGD process, the subset  $\mathcal{K}_i$  undergoes re-sampling.

*Black-box Variational Inference* (BBVI) facilitates the optimization of any (named or unnamed) posterior density  $q_{\boldsymbol{\phi}}(\mathbf{z})$ , through Monte Carlo estimates of variational gradients [35]. This can be formulated as the following relation

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} F[q] &\approx \nabla_{\boldsymbol{\phi}} \hat{F}[q], \\ \nabla_{\boldsymbol{\phi}} \hat{F}[q] &= \frac{1}{S} \sum_{s=1}^S \nabla_{\boldsymbol{\phi}} \ln q_{\boldsymbol{\phi}}(\mathbf{z}) \left[ \ln \frac{q_{\boldsymbol{\phi}}(\mathbf{z})}{p(\mathcal{D}, \mathbf{z})} + 1 \right], \\ \mathbf{z}_s &\sim q(\mathbf{z}|\boldsymbol{\phi}),\end{aligned} \quad (3)$$

which is known as the REINFORCE estimator [36]. To mitigate the variance inherent to Monte Carlo gradient estimations, we employ a pathwise gradient estimator [37], [38] for a fully factorised Gaussian posterior distribution (see section II-F for details) and Rao-Blackwellization [39], with an implementation provided in NumPyro PPL [40]. Although, numerous other techniques exist for variance reduction of gradient estimators [41] they are often more expensive to compute, hence we did not explore them in this work.

For stochastically minimizing the variational objective, one can in practice employ any of the readily available deep learning optimizers within the JAX ecosystem [42] implemented within the Optax package. Here we have selected the AdaBelief optimizer [43] as it enabled the fastest convergence rate (tested on a fixed number of gradient updates) on the set of problems we explored in this work. A more systematic benchmark and tuning of different optimization algorithms would be required to identify the optimal algorithm [44], but this falls beyond the scope of the paper.

## D. BAYESIAN MODEL REDUCTION

Let us consider two generative processes for the data: a full model

$$p(\mathbf{z}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{z}) p(\mathbf{z})$$

and a reduced model (the term ‘reduced’ here implies applying constraints of any form to the prior to obtain a posterior with reduced entropy) in which the original prior  $p(\mathbf{z})$  is replaced with a more informative prior  $\tilde{p}(\mathbf{z}) = p(\mathbf{z}|\boldsymbol{\theta})$  that depends on hyper-parameters  $\boldsymbol{\theta}$ . This change leads to a different posterior

$$\tilde{p}(\mathbf{z}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{z}) \tilde{p}(\mathbf{z}).$$

Noting that as the following relation holds:

$$1 = \int dz \tilde{p}(z|\mathcal{D}) = \int dz p(z|\mathcal{D}) \frac{\tilde{p}(z)p(\mathcal{D})}{p(z)\tilde{p}(\mathcal{D})},$$

we can express the link between the models as:

$$\begin{aligned} -\ln \tilde{p}(\mathcal{D}) &= -\ln p(\mathcal{D}) - \ln \int dz p(z|\mathcal{D}) \frac{\tilde{p}(z)}{p(z)} \\ &\approx F(\phi^*) - \ln \int dz q_{\phi^*}(z) \frac{\tilde{p}(z)}{p(z)} \end{aligned} \quad (4)$$

where we assumed the approximate posterior for the full model corresponds to  $p(z|\mathcal{D}) \approx q_{\phi^*}(z)$ , and that  $-\ln p(\mathcal{D}) \approx F(\phi^*)$ .

From eq. (4) we obtain the free energy of the reduced model as

$$-\ln \tilde{p}(\mathcal{D}) \approx -\ln E_q \left[ \frac{\tilde{p}(z)}{p(z)} \right] + F(\phi^*) = -\Delta F(\theta), \quad (5)$$

where  $\Delta F(\theta)$  denotes the change in the free energy of going from the full model to the reduced model, given hyperparameters  $\theta$ . Note that for  $\Delta F(\theta) > 0$  the reduced model has a better variational free energy compared to the flat model. Consequently, the reduced model offers a model with a greater marginal likelihood; i.e., a better explanation for the data and improved generalization capabilities. Heuristically, this can be understood as minimising model complexity, without sacrificing accuracy (because log evidence can be expressed as accuracy minus complexity, where complexity is the KL divergence between posterior and prior beliefs). This relationship is pivotal in formulating efficient pruning criteria, especially for extensive parametric models commonly employed in deep learning.

### E. BAYESIAN NEURAL NETWORKS WITH SHRINKAGE PRIORS

Shrinkage priors instantiate a prior belief about the sparse structure of model parameters. Here, we will investigate two well-established forms of shrinkage priors for network weight parameters, a canonical spike-and-slab prior [26], [45] defined as

$$\begin{aligned} w_{ijl} &\sim \mathcal{N}(0, \lambda_{ijl}^2 \gamma_0^2) \\ \lambda_{ijl} &\sim \text{Bernoulli}(\pi_l) \\ \pi_l &\sim \text{Be}(\alpha_0, \beta_0) \end{aligned}$$

and a regularised-horseshoe prior [27].

$$\begin{aligned} w_{ijl} &\sim \mathcal{N}(0, \gamma_{il}^2) \\ \gamma_{il}^2 &= \frac{c_l^2 v_l^2 \tau_{il}^2}{c_l^2 + \tau_{il}^2 v_l^2} \\ c_l^{-2} &\sim \Gamma(2, 6) \\ \tau_{il} &\sim \mathcal{C}^+(0, 1) \\ v_l &\sim \mathcal{C}^+(0, \tau_0) \end{aligned} \quad (6)$$

where  $i \in \{1, \dots, K_l\}$ ,  $j \in [1, \dots, K_{l-1} + 1]$ , and where  $w_{ijl}$  denotes  $ij$ th element of the weight matrix at depth  $l$ . The symbols  $\text{Be}$ , and  $\mathcal{C}^+$  denote a Beta distribution and a half-Cauchy distribution, respectively.

Importantly, the spike-and-slab prior relates to dropout regularisation, which is commonly introduced as a sparsification method in deep learning [11], [46]. This type of prior is considered the gold standard in shrinkage priors and has been used in many recent applications of Bayesian sparsification on neuronal networks [47], [48], [49], [50], [51] showing excellent sparsification rates. However, the inversion of the resulting hierarchical model is challenging and requires carefully constructed posterior approximations. Moreover, their dependence on discrete random variables renders them unsuitable for Markov-Chain Monte Carlo-based sampling schemes. As a result, researchers often use continuous formulations of the shrinkage-prior, with the horseshoe prior being a notable example.

In contexts that involve sparse learning with scant data, the regularised horseshoe prior has emerged as one of the preferred choices within shrinkage prior families [52]. A distinct advantage of this prior is its ability to define both the magnitude of regularisation for prominent coefficients and convey information about sparsity. It is worth noting a dependency highlighted in [13]: for  $v_l \tau_{il} \ll 1$  the equation simplifies to  $\gamma_{il} \approx v_l \tau_{il}$  recovering the original horseshoe prior. In contrast, for  $v_l \tau_{il} \gg 1$ , the equation becomes  $\gamma_{il}^2 \approx c_l^2$ . In this latter scenario, the prior over the weights is defined as  $w_{ijl} \sim \mathcal{N}(0, c_l^2)$ , with  $c_l$  serving as a weight decay hyper-parameter for layer  $l$ .

### F. APPROXIMATE POSTERIOR FOR BAYESIAN NEURAL NETWORKS

To benchmark stochastic BMR, we explore two forms of prior distribution  $p(\mathcal{W})$ —a flat and a hierarchical structure—in conjunction with a fully factorised mean-field approximation.

Firstly, let us consider the flat prior over model weights, represented in a non-centered parameterization:

$$\begin{aligned} c_l^{-2} &\sim \Gamma(2, 2) \\ \hat{w}_{ijl} &\sim \mathcal{N}(0, 1) \\ w_{ijl} &= \gamma_0 c_l \hat{w}_{ijl} \end{aligned} \quad (7)$$

where we set  $\gamma_0 = 0.1$ . Note that in the flat prior we incorporate a layer specific scale parameter, which we found to stabilise variational inference. Based on this, we describe a fully factorised approximate posterior as a composite of Normal and Log-Normal distributed random variables. Hence,

$$\begin{aligned} q(\hat{\mathcal{W}}, \mathbf{c}) &= \prod_l q(c_l^{-2}) \prod_i \prod_j q(\hat{w}_{ijl}) \\ q(\hat{w}_{ijl}) &= \mathcal{N}(\mu_{ijl}, \sigma_{ijl}^2) \\ q(c_l^{-2}) &= \mathcal{LN}(\mu_{c,l}, \sigma_{c,l}^2). \end{aligned} \quad (8)$$

When inverting a hierarchical generative model over weights of artificial neural network, we exclusively apply stochastic black-box variational inference to the model variant with the regularised horseshoe prior. This choice is motivated by its documented superiority over the spike-and-slab prior, as established in [13]. We express the hierarchical prior in the non-centered parameterization as:

$$\begin{aligned} a_{il}, b_{il} &\sim \Gamma\left(\frac{1}{2}, 1\right) \\ \hat{a}_l, \hat{b}_l &\sim \Gamma\left(\frac{1}{2}, 1\right) \\ \tau_{il}^2 &= \frac{a_{il}}{b_{il}}; \quad \nu_l^2 = \tau_0^2 \frac{\hat{a}_l}{\hat{b}_l} \\ w_{ijl} &= \gamma_{il} \hat{w}_{ijl} \end{aligned}$$

where  $c_l$  and  $\hat{w}_{ijl}$  are drawn from the same prior as in eq. (7). Note that the expression in section II-F involves a reparameterization of Half-Cauchy distributed random variables as the square-root of the quotient of two Gamma distributed random variables, a strategy drawn from [53] (see Appendix B for additional details). Such a reparameterization of the Half-Cauchy ensures capturing of fat-tails in the posterior, even when leveraging a fully-factorised mean-field posterior approximation, as referenced in [13].

For the fully-factorised mean-field approximation, the approximate posterior is portrayed as a composite of Normal and Log-Normal distributed random variables, expressed as:

$$\begin{aligned} q(\hat{\mathcal{W}}, \mathbf{a}, \mathbf{b}, \hat{\mathbf{a}}, \hat{\mathbf{b}}, \mathbf{c}) &= \prod_l q(c_l^{-2}) q(\hat{a}_l) q(\hat{b}_l) \\ &\quad \cdot \prod_i q(a_{il}) q(b_{il}) \prod_j q(\hat{w}_{ijl}) \\ q(c_l) &= \mathcal{LN}(\mu_{c,l}, \sigma_{c,l}^2) \\ q(\hat{a}_l) &= \mathcal{LN}(\hat{\mu}_{a,l}, \hat{\sigma}_{a,l}^2) \\ q(\hat{b}_l) &= \mathcal{LN}(\hat{\mu}_{b,l}, \hat{\sigma}_{b,l}^2) \\ q(a_{il}) &= \mathcal{LN}(\mu_{a,il}, \sigma_{a,il}^2) \\ q(b_{il}) &= \mathcal{LN}(\mu_{b,il}, \sigma_{b,il}^2) \\ q(\hat{w}_{ijl}) &= \mathcal{N}(\mu_{w,ijl}, \sigma_{w,ijl}^2) \end{aligned}$$

### G. APPLICATION OF STOCHASTIC BMR TO BAYESIAN NEURAL NETWORKS

Here we derive a detailed account of stochastic BMR, specifying a novel algorithm for Bayesian sparsification of artificial neural networks. The stochastic BMR is summarized in algorithm 1.

To apply BMR to Bayesian neural networks, we commence by estimating an approximate posterior for the flat model, as detailed in eq. (7). To retain high computational efficiency, we pair BMR solely with the fully factorised approximate posterior, as presented in eq. (8). While it is feasible to use this method alongside the structured posterior [13],

it requires considerably more computationally intensive estimations of the reduced free energy. As shown below, we obtain satisfactory results with a fully factorised posterior. Therefore, we defer the exploration of BMR with a structured posterior to future endeavours.

Given a fully factorised approximate posterior, we can determine the change in variational free energy,  $\Delta F$ —after substituting the prior  $\mathcal{N}(0, 1)$  with  $\mathcal{N}(0, \theta_{ijl}^2)$  for the weight  $\hat{w}_{ijl}$ —as:

$$\begin{aligned} \Delta F(\theta_{ijl}) &= -\frac{1}{2} \ln \rho_{ijl}^2 - \frac{1}{2} \frac{\mu_{ijl}^2}{\sigma_{ijl}^2} \left(1 - \frac{\theta_{ijl}^2}{\rho_{ijl}^2}\right) \\ \rho_{ijl}^2 &= \theta_{ijl}^2 + \sigma_{ijl}^2 - \theta_{ijl}^2 \sigma_{ijl}^2 \end{aligned}$$

For the second hierarchical level of the approximate posterior, we aim to minimize the following form for the variational free energy:

$$F = \sum_{l=1}^L E_{q(\theta_l)} \left[ -\sum_{i,j} \Delta F(\theta_{ijl}) + \ln \frac{q(\theta_l)}{p(\theta_l)} \right] \quad (9)$$

This minimization is done with respect to  $q(\Theta) = \prod_l q(\theta_l)$ , the approximate posterior over hyper-parameters. Note the application of eq. (5) in substituting the marginal log-likelihood with the change in the variational free energy.

For the spike-and-slab prior we can write the following relation:

$$\begin{aligned} \theta_l &= \left[ \pi_l, \lambda_{ijl} \text{ for } i, j \in \{1, \dots, K_l\}, \{1, \dots, K_{l-1} + 1\} \right], \\ \theta_{ijl} &= \lambda_{ijl}. \end{aligned}$$

Consequently, the approximate posterior at the second level of the hierarchy can be approximated as:

$$\begin{aligned} q(\Theta) &= \prod_l q(\pi_l) \prod_{ij} q(\lambda_{ijl}) \\ q(\lambda_{ijl}) &= q_{ijl}^{\lambda_{ijl}} (1 - q_{ijl})^{1-\lambda_{ijl}} \\ q(\pi_l) &= \mathcal{B}(\alpha_l, \beta_l) \end{aligned}$$

The iterative update to obtain the minimum of the simplified variational free energy (eq. (9)) is then:

$$\begin{aligned} q_{ijl}^{k+1} &= \frac{1}{1 + e^{-[\zeta_l^k - \Delta F(\lambda_{ijl}=0)]}} \\ \zeta_l^k &= \psi(\alpha_l^k) - \psi(\beta_l^k) \\ \alpha_l^{k+1} &= \sum_{i,j} q_{ijl}^{k+1} + \alpha_0 \\ \beta_l^{k+1} &= \sum_{i,j} (1 - q_{ijl}^{k+1}) + \beta_0 \end{aligned}$$

Here,  $\alpha_l^0 = \alpha_0$ ,  $\beta_l^0 = \beta_0$ ,  $\Delta F(\lambda_{ijl}=0) = -\frac{1}{2} \left[ \ln \sigma_{ijl}^2 + \frac{\mu_{ijl}^2}{\sigma_{ijl}^2} \right]$ , and  $\psi(\cdot)$  refers to the digamma function.



**Algorithm 1** Stochastic BMR

**Require:** data  $(y, x)$ , joint distribution  $p$ , approximate posterior  $q$ .

Initialise  $\phi$  randomly,  $e = 1$ .

**while**  $e \leq \text{MaxEpochs}$  **do**

$t = 1, k = 1$ .

**while**  $t \leq \text{MaxIters}$  **do**

**for**  $s = 1$  to  $S$  **do**

$\hat{W}_{s,t}, c_{s,t} \sim q_{\phi_t}(\hat{W}, c)$  {Draw  $S$  samples from  $q$ .}

**end for**

$\phi_{t+1} \leftarrow \text{AdaBelief}[\phi_t, \nabla_{\phi} \hat{F}[q_{\phi_t}]]$

$t \leftarrow t + 1$

**end while**

Compute  $\Delta F$  ( $\lambda_{ijl} = 0$ ).

Initialise  $\alpha_l^0 = \alpha_0$ , and  $\beta_l^0 = \beta_0$ .

**while**  $k \leq k_{max}$  **do**

$\zeta_l^k \leftarrow \psi(\alpha_l^k) - \psi(\beta_l^k)$

$q_{ijl}^{k+1} \leftarrow \sigma(\zeta_l^k - \Delta F(\lambda_{ijl} = 0))$

$\alpha_l^{k+1} \leftarrow \sum_{i,j} q_{ijl}^{k+1} + \alpha_0$

$\beta_l^{k+1} \leftarrow \sum_{i,j} (1 - q_{ijl}^{k+1}) + \beta_0$

$k \leftarrow k + 1$

**end while**

**if**  $q_{ijl}^{k_{max}} < 1/2$  **then**

$\hat{w}_{ijl} \leftarrow 0$

**end if**

$e \leftarrow e + 1$

**end while**

When combined with a simple pruning heuristics for eliminating model weights, defined as

$$\text{if } q_{ijl}^{k_{max}} < \frac{1}{2}, \text{ set } \hat{w}_{ijl} = 0.$$

the pruning algorithm requires only few iterations for estimating posterior probabilities  $q_{ijl}$  and subsequently eliminating weights. Note that, already after a single iteration all probabilities will be either smaller or larger than one half, and one can easily apply pruning heuristics. Subsequent iterations are relevant for fine tuning values close to the decision threshold, thus eliminating potential false positives. In practice, we cap the maximum number of iterations at  $k_{max} = 4$ , as increasing this value does not have any noticeable impact on the pruning dynamics.

To achieve the high sparsification rate presented in the next section, we adopt an iterative optimisation and pruning approach proposed in [24]. We perform weight pruning at the beginning of each epoch (except the first one), and further optimisation for 500 iterations, completing one epoch. In total, we apply iterative pruning and optimisation for fifty epochs in all examples below.

The complete implementation of stochastic BMR is available at an online repository <https://github.com/dimarkov/bmr4pml> with notebooks and scripts necessary to recreate all result figures.

**III. RESULTS**

In this section, we present the outcomes of our experiments and analyses conducted to evaluate the performance and efficiency of the stochastic Bayesian model reduction in the context of Bayesian sparsification of deep neural networks. Our results are structured to provide insights into the capabilities and advantages of our approach.

**A. PERFORMANCE COMPARISON**

The training regimen used a batch size of  $N_B = 128$  and the AdaBelief algorithm with learning rate set to  $\alpha = 10^{-3}$  in the case of the MAP estimate,  $\alpha = 5 \cdot 10^{-3}$  in the case of the mean-field methods, and  $\alpha = 10^{-2}$  in the case of stochastic BMR (the exponential decay rates were kept at default values  $\beta_1 = 0.9$ , and  $\beta_2 = 0.999$ ). Fig. 1 charts the epoch-wise evolution of Top-1 accuracy (ACC) expected-calibration error (ECE), and negative log-likelihood (NLL) for each architecture, under five distinct approximate inference strategies: (i) Maximum a posteriori (MAP) estimate for the flat generative model, akin to traditional deep learning point estimates coupled with weight decay. (ii) A fully factorised posterior approximation for the flat generative model (Flat-FF). (iii) A fully factorised posterior approximation of the hierarchical generative model with a regularised horseshoe prior (Tiered-FF). (iv) The stochastic BMR algorithm augmented with a spike-and-slab

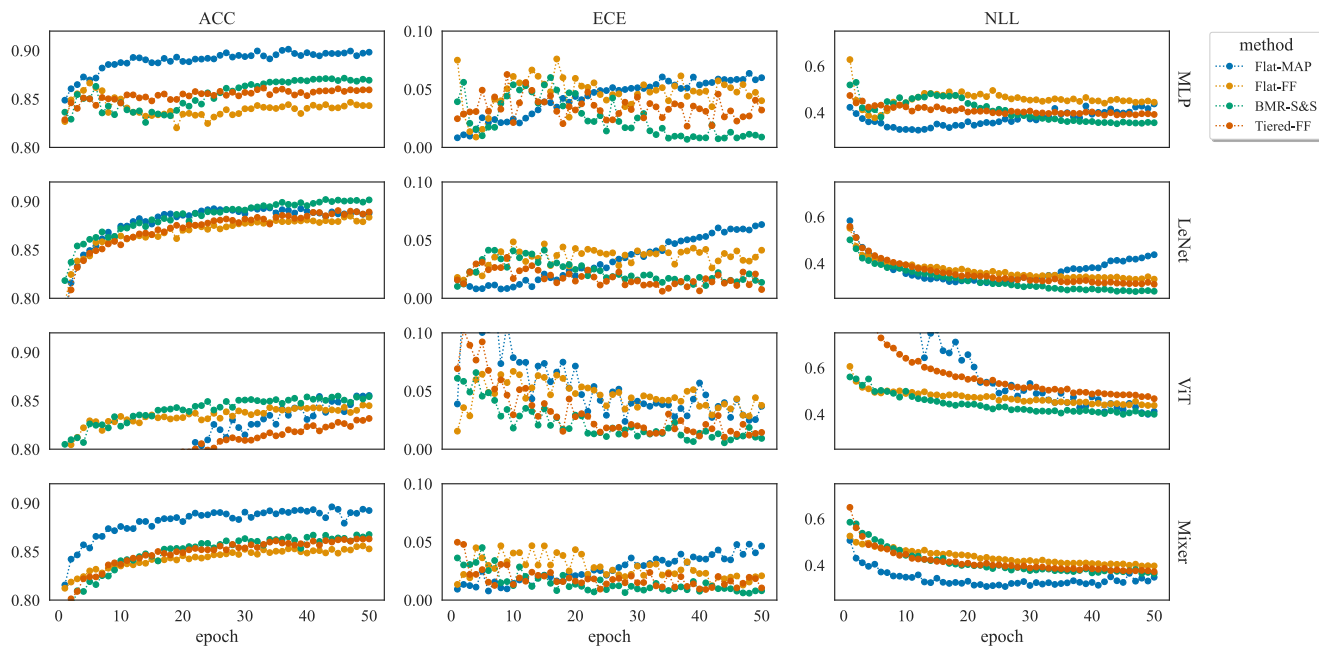


FIGURE 1. Classification performance comparison on FashionMNIST dataset for different neuronal architectures and approximate inference schemes.

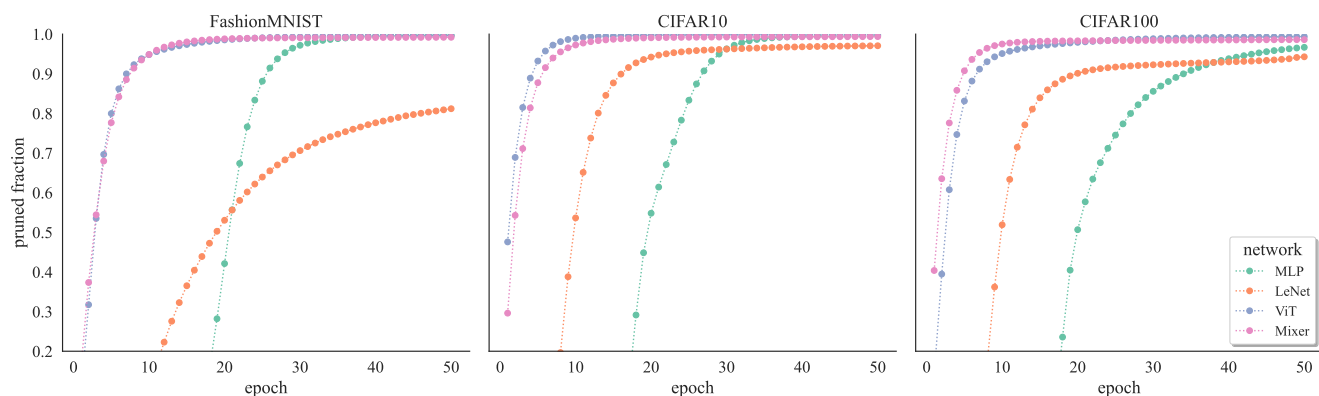


FIGURE 2. Total fraction of pruned model parameters obtained with the stochastic BMR algorithm across different DNN architectures and datasets.

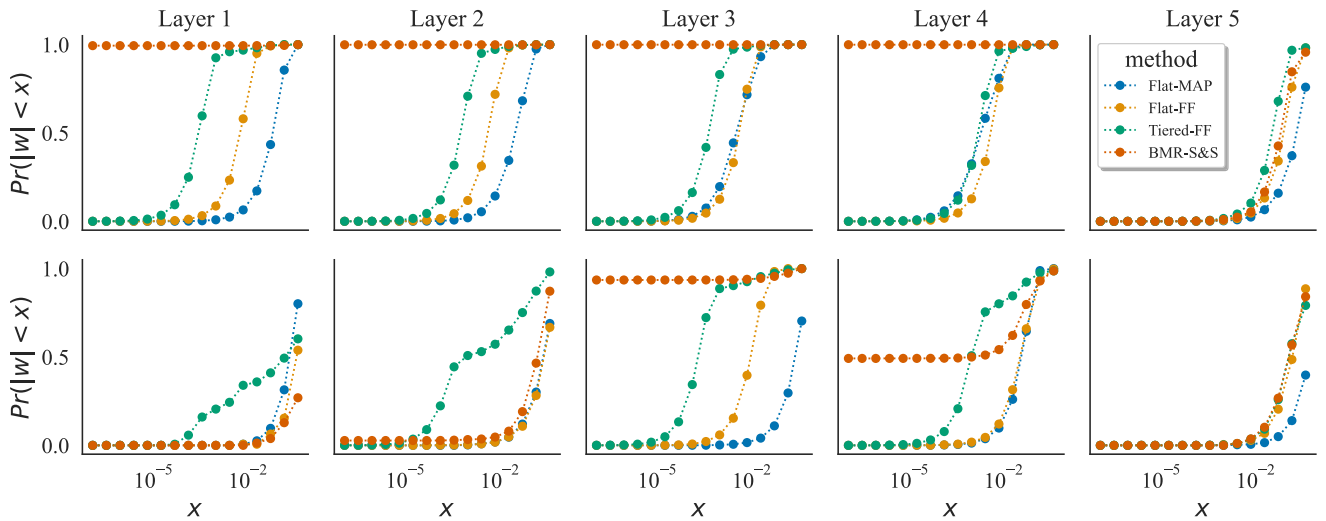
prior (BMR-S&S). Each epoch is defined by 500 stochastic gradient steps, with each step randomly drawing  $N_B$  data instances from the training pool. Furthermore, the three performance metrics (ACC, ECE and NLL) are obtained using the marginal likelihood of data labels, estimated as average (predictive) probability of each label given  $M = 100$  samples from the approximate posterior at the end of each epoch.

Interestingly, all approximate inference methods demonstrate comparable top-1 accuracy scores. However, the stochastic BMR method followed by the Tiered-FF approximation (with a single exception), consistently resulted in the lowest ECE and NLL scores across the majority of DNN architectures and datasets (see the supplementary Fig. 5 for CIFAR10 dataset and the supplementary Fig. 6 for CIFAR100

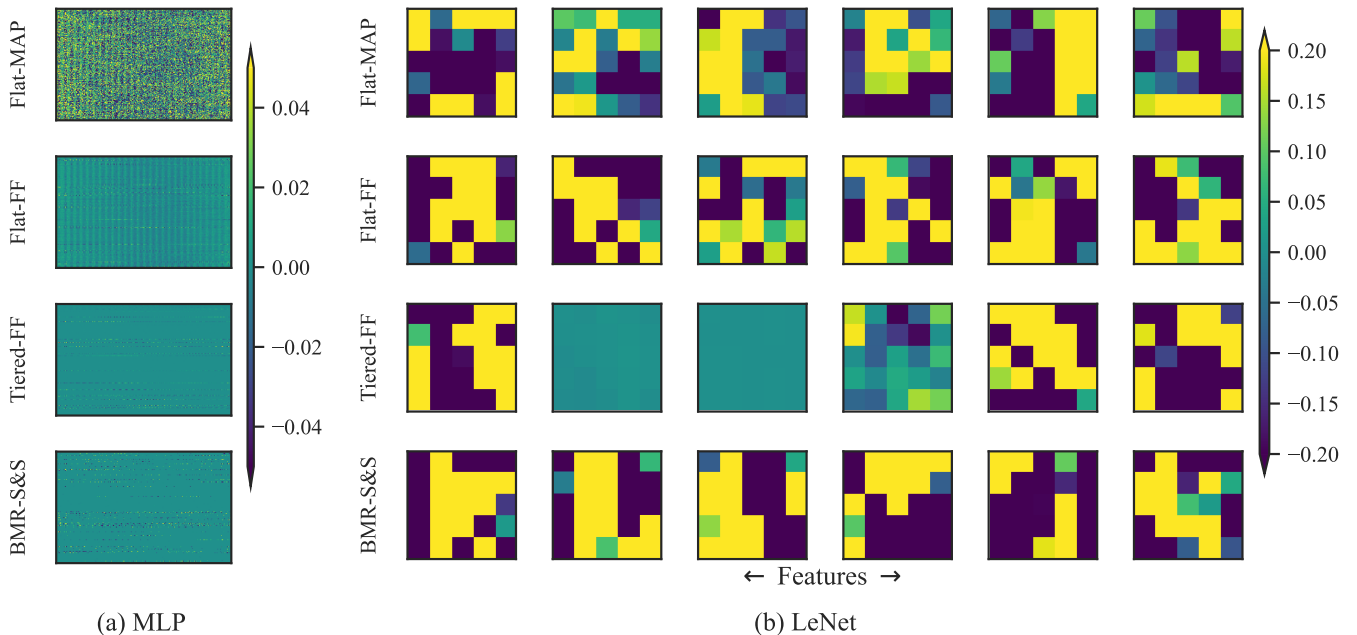
dataset). The implicit reduction in model complexity suggests that—as anticipated—Stochastic BMR furnishes a model of the data that has the greatest evidence or marginal likelihood (not shown). In this setting, the NLL of the test data can be regarded as a proxy for (negative log) marginal likelihood.

**B. LEARNING OF SPARSE REPRESENTATIONS**

Fig. 2 depicts the fraction of pruned model parameters for different DNN architectures and datasets. It is noteworthy to observe the substantive sparsity achieved by the stochastic BMR algorithm. This sparsity is consistent across datasets and architectures, with the exception of the LeNet-5 structure when used for the FashionMNIST dataset, because by default LeNet-5 architecture is already sparse and contains relatively low-number of model weights (for other data sets we



**FIGURE 3.** Cumulative Distribution Function (CDF) of absolute posterior parameter expectations at different layers of MLP (top row), and LeNet architectures (bottom row). The y-axis represents the fraction of parameters with values less than or equal to the value on the x-axis.



**FIGURE 4.** Posterior expectations (color coded) over model weights obtained using different approximate inference schemes at the first layer of (a) MLP architecture, and (b) LeNet architectures.

substantially increased the dimensionality of hidden layers as detailed in Appendix A).

To delve deeper into the pruning behavior across varying network depths, Fig. 3 presents a per-layer cumulative distribution function (CDF) for model parameters, highlighting the proportion of parameters whose absolute mean posterior estimate falls below a given threshold. When juxtaposing the BMR CDF trajectories with those obtained from the Tiered-FF method (sparsification is induced by the regularised half-cauchy prior), it is evident that BMR furnishes more pronounced sparsification. This distinction is

crucial, as the stochastic BMR not only matches or surpasses the performance of the Tiered-FF algorithm but also averages a 30% faster stochastic gradient descent.

To illustrate the structural learning variations among algorithms, Fig. 4 presents heatmaps of posterior expectations obtained using the four different methods. The Fig. 4 reveals subtle differences between inferred representations of the MLP and LeNet-5 architecture’s input layers trained on the Fashion MNIST dataset. Divergent compression rates among the algorithms indicate inherent trade-offs between efficiency and performance. It is evident that the stochastic



BMR strikes a balance between compression advantages and performance, as it is less prone to over-pruning as compared to the Tiered-FF method (two featured of the LeNet-5 input layer are effectively removed - see Fig. 4(b)).

#### IV. RELATED WORK

Over recent years, the Bayesian sparsification of neural networks has gained momentum, primarily driven by the spike-and-slab prior [47], [48], [49], [50], [51], [54] and variants of the horseshoe prior [9], [52]. These works have showcased the impressive sparsification capabilities inherent to such shrinkage priors.

Nevertheless, when juxtaposed with the stochastic BMR algorithm, they often necessitate supplementary assumptions related to the approximate posterior. These assumptions, in turn, lead to a more computation-intensive model inversion. For example, in [47], [48], and [49] the authors apply a continuous approximation to the Bernoulli distribution, in the form of Gumbel-softmax approximation. This reparameterisation is successful for learning sparse representations, but it increases the parameter space and hence the computational complexity relative to the stochastic BMR approach. On the other hand, in [50] the sparsification rests on Metropolis-Hastings algorithm which requires reevaluation of the data likelihood for individual samples from the proposal distribution.

Finally, in contrast to related approaches, the versatility of stochastic BMR allows its integration with more efficient optimization techniques, like variational Laplace [16] and proximal-gradient methods [55], provided the resulting approximate posterior in the form of a normal distribution is apt for the application at hand. The computational complexity of these extensions of stochastic BMR method would be comparable to classical deep learning with point estimation or an efficient Bayesian pruning algorithm recently proposed in [51].

#### V. CONCLUSION

In this study, we presented a novel algorithm—stochastic Bayesian model reduction—designed for an efficient Bayesian sparsification of deep neural networks. Our proposed method seamlessly integrates stochastic and black-box variational inference with Bayesian model reduction (BMR), a generalisation of the Savage-Dickey ratio. Through the stochastic BMR strategy, we enable iterative pruning of model parameters, relying on posterior estimates acquired from a straightforward variational mean-field approximation to the generative model. This model is characterized by Gaussian priors over individual parameters and layer-specific scale parameters. The result is an efficient pruning algorithm for which the computational demand of the pruning step is negligible compared to the direct stochastic black-box optimization of the full hierarchical model.

The insights obtained here pave the way for a deeper exploration of the potential applications of Bayesian model reduction across a wider array of architectures and tasks

in probabilistic machine learning, such as audiovisual and natural language processing tasks. A more detailed fine tuning of the core dynamics of these algorithms, in terms of iterations steps, learning rates, and other free-parameters, might be the key to unveiling even more proficient Bayesian deep learning methodologies in the near future.

#### APPENDIX A SPECIFICATIONS OF NN MODELS

For the simple multi-layer perceptron, we configure the architecture with five hidden layers, each comprising 400 neurons. The chosen activation function is the Swish activation function [56].

For the LeNet-5 architecture, we adhere to the original design, which includes three convolutional layers, average pooling following the initial two convolutional layers, and two linear layers. The activation function used is the hyperbolic tangent. The convolutional layers employ a kernel size of  $5 \times 5$ , while the average pooling uses a window of shape  $2 \times 2$ . For the FashionMNIST dataset, the feature counts of the convolutional layers are designated as (6, 16, 120), and the two linear layers have neuron counts of (84, 10). However, for the CIFAR10 and CIFAR100 datasets, we elevate the feature counts of the convolutional layers to (18, 48, 360), with linear layer neuron counts set to (256, 10) for CIFAR10 and (256, 100) for CIFAR100.

For the MlpMixer architecture we employ six layers and a patch resolution of  $4 \times 4$ . Across all datasets, we maintain constant values for hidden size ( $C$ ), sequence length ( $S$ ), MLP channel dimension ( $D_C$ ), and MLP token dimension ( $D_S$ ); specifically  $C = 256$ ,  $S = 64$ ,  $D_C = 512$  and  $D_S = 512$  for all datasets.

For the VisionTransformer architecture, we adopt a slightly modified version of the ViT-Tiny setup: we use six layers, eight heads for each attention block, an embedding dimension of 256, and a hidden dimension of 512. The patch resolution of  $4 \times 4$  is consistent with the MlpMixer. In both MlpMixer and VisionTransformer architectures, the GeLU activation function is used [57].

For training using the maximum a posteriori estimate (Flat-MAP), dropout regularization, with dropout probability set to 0.2, is applied to all linear layers across all architectures, with the exception of the MlpMixer.

#### APPENDIX B REPARAMETERIZATION

In the centered parameterization of a generative model, Stochastic Variational Inference (SVI) with a fully factorized posterior yields a non-sparse solution, undermining the objective of employing shrinkage priors [52]. Typically, this limitation is addressed by adopting the non-centered parameterization of the prior.

Consider the unique property of the half-Cauchy distribution: given  $x \sim C^+(0, 1)$ , and  $z = bx$  the resulting probability distribution for  $z$  is  $z \sim C^+(0, b)$ . Therefore, the non-centred

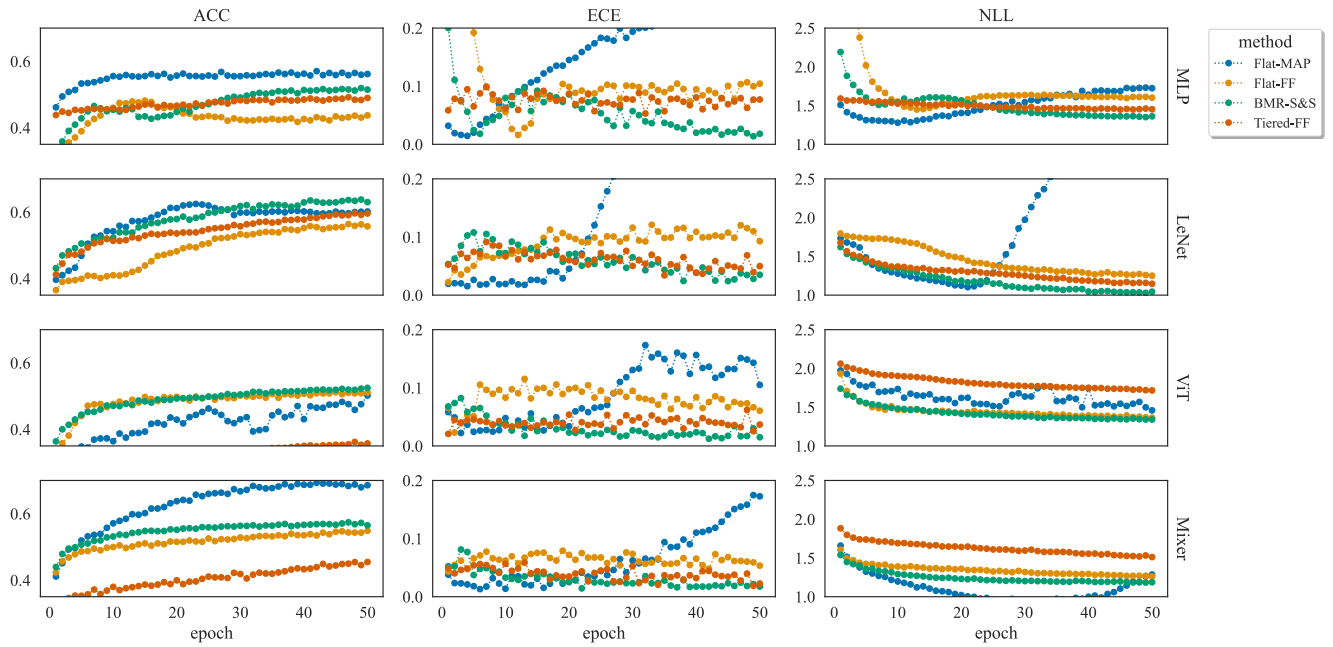


FIGURE 5. Classification performance comparison on CIFAR10 dataset for different neuronal architectures and approximate inference schemes.

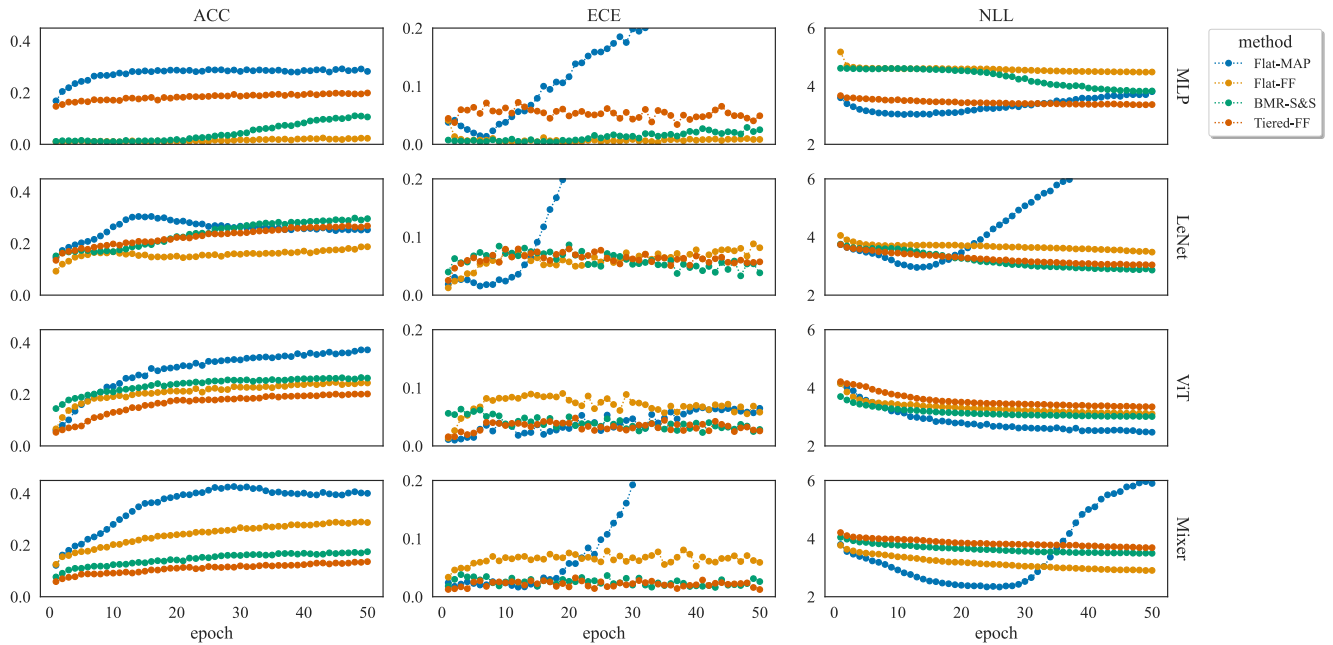


FIGURE 6. Classification performance comparison on CIFAR100 dataset for different neuronal architectures and approximate inference schemes.

parameterization is formulated as

$$\begin{aligned} \hat{\tau}_i^l &\sim \mathcal{C}^+(0, 1) \\ \hat{\lambda}_{ij}^l &\sim \mathcal{C}^+(0, 1) \\ \hat{w}_{ij}^l &\sim \mathcal{N}(0, 1) \\ [\gamma_{ij}^l]^2 &= \frac{[c^l \tau_0^l \hat{\tau}_i^l \hat{\lambda}_{ij}^l]^2}{[c^l]^2 + [\tau_0^l \hat{\tau}_i^l \hat{\lambda}_{ij}^l]^2} \end{aligned}$$

$$w_{ij}^l = \gamma_{ij}^l \hat{w}_{ij}^l$$

However, while the half-Cauchy distribution is frequently chosen for sampling-based inference, it poses challenges in variational inference [27]. Firstly, exponential family-based approximate posteriors (e.g., Gamma or log-Normal distributions) inadequately capture the half-Cauchy distribution’s fat tails. Secondly, using a Cauchy approximating family for the posterior results in high variance gradients during stochastic variational inference [52]. Hence, in the context of stochastic

variational inference, the half-Cauchy distribution undergoes a reparameterization, as described in [13]:

$$x \sim \mathcal{C}^+(0, b) \equiv x = \sqrt{\frac{1}{u}}, u \sim \Gamma\left(\frac{1}{2}, \frac{1}{v}\right), v \sim \Gamma\left(\frac{1}{2}, b^2\right)$$

or, when represented in the non-centered parameterization:

$$x = b\sqrt{\frac{v}{u}}, u \sim \Gamma\left(\frac{1}{2}, 1\right), v \sim \Gamma\left(\frac{1}{2}, 1\right) \quad (10)$$

## APPENDIX C FIGURES

See Figures 5 and 6.

## ACKNOWLEDGMENT

The authors would like to thank Conor Heins, Magnus Koudahl, and Beren Millidge for valuable discussions during the initial stages of this work.

## REFERENCES

- [1] H. Wang and D.-Y. Yeung, "A survey on Bayesian deep learning," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–37, Sep. 2021.
- [2] A. G. Wilson, "The case for Bayesian deep learning," 2020, *arXiv:2001.10995*.
- [3] H. Wang and D.-Y. Yeung, "Towards Bayesian deep learning: A framework and some existing methods," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3395–3408, Dec. 2016.
- [4] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2022.
- [5] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015.
- [6] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 4697–4708.
- [7] P. Izmailov, W. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. Wilson, "Subspace inference for Bayesian deep learning," in *Proc. 35th Uncertainty Artif. Intell. Conf.*, Jul. 2020, pp. 1169–1179.
- [8] X. Luo and A. Kareem, "Bayesian deep learning with hierarchical prior: Predictions from limited and noisy data," *Struct. Saf.*, vol. 84, May 2020, Art. no. 101918.
- [9] C. Louizos, K. Ullrich, and M. Welling, "Bayesian compression for deep learning," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/69d1fc78dbda242c43ad6590368912d4-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/69d1fc78dbda242c43ad6590368912d4-Paper.pdf)
- [10] V. Fortuin, "Priors in Bayesian deep learning: A review," *Int. Stat. Rev.*, vol. 90, no. 3, pp. 563–591, Dec. 2022.
- [11] E. Nalisnick, J. M. Hernández-Lobato, and P. Smyth, "Dropout as a structured shrinkage prior," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4712–4722.
- [12] S. Seto, M. T. Wells, and W. Zhang, "HALO: Learning to prune neural networks with shrinkage," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2021, pp. 558–566.
- [13] S. Ghosh, J. Yao, and F. Doshi-Velez, "Structured variational learning of Bayesian neural networks with horseshoe priors," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 1744–1753.
- [14] J. Snoek, "Scalable Bayesian optimization using deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2171–2180.
- [15] R. Krishnan, M. Subedar, and O. Tickoo, "Efficient priors for scalable variational inference in Bayesian deep neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 773–777.
- [16] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux—effortless Bayesian deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., 2021, pp. 20089–20103.
- [17] E. Cameron, "A generalized savage-dickey ratio," 2013, *arXiv:1311.1292*.
- [18] M. J. Rosa, K. Friston, and W. Penny, "Post-hoc selection of dynamic causal models," *J. Neurosci. Methods*, vol. 208, no. 1, pp. 66–78, Jun. 2012.
- [19] K. Friston and W. Penny, "Post hoc Bayesian model selection," *NeuroImage*, vol. 56, no. 4, pp. 2089–2099, 2011.
- [20] K. J. Friston, V. Litvak, A. Oswal, A. Razi, K. E. Stephan, B. C. M. van Wijk, G. Ziegler, and P. Zeidman, "Bayesian model reduction and empirical Bayes for group (DCM) studies," *NeuroImage*, vol. 128, pp. 413–431, Mar. 2016.
- [21] K. J. Friston, M. Lin, C. D. Frith, G. Pezzulo, J. A. Hobson, and S. Ondobaka, "Active inference, curiosity and insight," *Neural Comput.*, vol. 29, no. 10, pp. 2633–2683, Oct. 2017.
- [22] K. Friston, T. Parr, and P. Zeidman, "Bayesian model reduction," 2019, *arXiv:1805.07092*.
- [23] R. Smith, P. Schwartenbeck, T. Parr, and K. J. Friston, "An active inference approach to modeling structure learning: Concept learning as an example case," *Frontiers Comput. Neurosci.*, vol. 14, p. 41, May 2020.
- [24] J. Beckers, B. Van Erp, Z. Zhao, K. Kondrashov, and B. De Vries, "Principled pruning of Bayesian neural networks through variational free energy minimization," *IEEE Open J. Signal Process.*, vol. 5, pp. 195–203, 2023.
- [25] M. Haußmann, F. A. Hamprecht, and M. Kandemir, "Sampling-free variational inference of Bayesian neural networks by variance backpropagation," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 563–573.
- [26] T. J. Mitchell and J. J. Beauchamp, "Bayesian variable selection in linear regression," *J. Amer. Stat. Assoc.*, vol. 83, no. 404, pp. 1023–1032, 1988.
- [27] J. Piironen and A. Vehtari, "Sparsity information and regularization in the horseshoe and other shrinkage priors," *Electron. J. Statist.*, vol. 11, no. 2, pp. 5018–5051, Jan. 2017.
- [28] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [30] I. Tolstikhin, "MLP-mixer: An all-MLP architecture for vision," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24261–24272.
- [31] T. Papamarkou, "Position: Bayesian deep learning is needed in the age of large-scale AI," 2024, *arXiv:2402.00809*.
- [32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.
- [33] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*. London, U.K.: Univ. London, Univ. College London, 2003.
- [34] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, Jan. 2013.
- [35] R. Ranganath, S. Gerrish, and D. Blei, "Black box variational inference," in *Proc. Artif. Intell. Statist.*, 2014, pp. 814–822.
- [36] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.
- [37] M. Figurnov, S. Mohamed, and A. Mnih, "Implicit reparameterization gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2015/hash/de03beffed9da5f3639a621bcab5dd4-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2015/hash/de03beffed9da5f3639a621bcab5dd4-Abstract.html)
- [38] M. Jankowiak and F. Obermeyer, "Pathwise derivatives beyond the reparameterization trick," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2235–2244.
- [39] J. Schulman, N. Heess, T. Weber, and P. Abbeel, "Gradient estimation using stochastic computation graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2015/hash/de03beffed9da5f3639a621bcab5dd4-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2015/hash/de03beffed9da5f3639a621bcab5dd4-Abstract.html)
- [40] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, no. 28, pp. 1–6, 2019.
- [41] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte Carlo gradient estimation in machine learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5183–5244, Dec. 2020.
- [42] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, (2018). *JAX: Composable Transformations of Python+NumPy Programs. 0.3.13*. [Online]. Available: <http://github.com/google/jax>

- [43] J. Zhuang, "AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18795–18806.
- [44] R. M. Schmidt, F. Schneider, and P. Hennig, "Descending through a crowded valley-benchmarking deep learning optimizers," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9367–9376.
- [45] E. I. George and R. E. McCulloch, "Variable selection via Gibbs sampling," *J. Amer. Stat. Assoc.*, vol. 88, no. 423, p. 881, Sep. 1993.
- [46] A. Mobiny, P. Yuan, S. K. Moulik, N. Garg, C. C. Wu, and H. Van Nguyen, "DropConnect is effective in modeling uncertainty of Bayesian deep networks," *Sci. Rep.*, vol. 11, no. 1, p. 5458, Mar. 2021.
- [47] J. Bai, Q. Song, and G. Cheng, "Efficient variational inference for sparse deep learning with theoretical guarantee," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 466–476.
- [48] A. Hubin and G. Storvik, "Variational inference for Bayesian neural networks under model and parameter uncertainty," 2023, *arXiv:2305.00934*.
- [49] S. Jantre, S. Bhattacharya, and T. Maiti, "Layer adaptive node selection in Bayesian neural networks: Statistical guarantees and implementation details," *Neural Netw.*, vol. 167, pp. 309–330, Oct. 2023.
- [50] Y. Sun, Q. Song, and F. Liang, "Learning sparse deep neural networks with a spike-and-slab prior," *Statist. Probab. Lett.*, vol. 180, Jan. 2022, Art. no. 109246.
- [51] X. Ke and Y. Fan, "On the optimization and pruning for Bayesian deep learning," 2022, *arXiv:2210.12957*.
- [52] S. Ghosh, J. Yao, and F. Doshi-Velez, "Model selection in Bayesian neural networks via horseshoe priors," *J. Mach. Learn. Res.*, vol. 20, no. 182, pp. 1–46, 2019.
- [53] M. P. Wand, J. T. Ormerod, S. A. Padoan, and R. Frühwirth, "Mean field variational Bayes for elaborate distributions," *Bayesian Anal.*, vol. 6, no. 4, pp. 847–900, Dec. 2011.
- [54] Y. Sun, Q. Song, and F. Liang, "Consistent sparse deep learning: Theory and computation," *J. Amer. Stat. Assoc.*, vol. 117, no. 540, pp. 1981–1995, Oct. 2022.
- [55] M. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, "Fast and scalable Bayesian deep learning by weight-perturbation in ADAM," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2611–2620.
- [56] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
- [57] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.



**KARL J. FRISTON** is currently a Theoretical Neuroscientist and an authority on brain imaging. He invented statistical parametric mapping (SPM), voxel-based morphometry (VBM), and dynamic causal modeling (DCM). These contributions were motivated by schizophrenia research and theoretical studies of value-learning, formulated as the dysconnection hypothesis of schizophrenia. Mathematical contributions include variational Laplacian procedures and generalized filtering

for hierarchical Bayesian model inversion. He works on models of functional integration in the human brain and the principles that underlie neuronal interactions. His main contribution to theoretical neurobiology is a free-energy principle for action and perception (active inference).

Dr. Friston was elected as a fellow of the Academy of Medical Sciences, in 1999. He became a fellow of the Royal Society of Biology. He was elected as a fellow of the Royal Society, in 2006. He was elected as a member of *Excellence in Life Science* (EMBO), in 2014. He was elected as a member of the Academia Europaea in 2015. He received the first Young Investigators Award in human brain mapping, in 1996. In 2003, he was awarded the Minerva Golden Brain Award. In 2008, he received the Medal from the Collège de France and the Honorary Doctorate from the University of York, in 2011. In 2012, he received the Weldon Memorial Prize and Medal for contributions to mathematical biology, in 2013. He was the 2016 recipient of the Charles Branch Award for unparalleled breakthroughs in brain research and the Glass Brain Award, a lifetime achievement award in the field of human brain mapping. He holds an Honorary Doctorates from the University of Zurich and Radboud University. In 2000, he was the President of the International Organization of Human Brain Mapping.



**DIMITRIJE MARKOVIĆ** received the Diploma degree in theoretical and experimental physics from the University of Belgrade, Serbia, in 2007, and the Dr. phil. nat. degree in theoretical physics from Goethe University Frankfurt, Germany, in 2013.

From 2008 to 2013, he was a Research Assistant with the Institute for Theoretical Physics, Goethe University Frankfurt. From 2013 to 2014, he was a Postdoctoral Researcher with the Biomagnetic Center, University Clinic, Jena, and from 2013 to 2015, he was a Guest Researcher with the Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig. Since 2015, he has been a Postdoctoral Researcher with the Chair of Cognitive Computational Neuroscience, Psychology Department, Technische Universität Dresden, Germany. His research interests include adaptive dynamical systems, computational neuroscience, cognitive neuroscience, and probabilistic machine learning and statistics.



**STEFAN J. KIEBEL** was trained as a Computer Scientist. He is currently a Professor of cognitive computational neuroscience with the Faculty of Psychology, TU Dresden, Germany. He is a Co-Developer of the widely used statistical parametric mapping SPM neuroimaging software package. Over time, his interests shifted towards cognitive neuroscience, sparking collaborations between the fields of cognitive neuroscience, psychology, computer science, and machine learning.

At present, his research focuses on Bayesian modeling, predictive coding, and active inference. His main research interests include understanding how our brain employs hierarchies of time scales for cognitive functions, such as decision-making under time constraints.

...