Connecting the Dots: Is Mode-Connectedness the Key to Feasible Sample-Based Inference in Bayesian Neural Networks?

Emanuel Sommer [*1 2] **Lisa Wimmer** [*1 2] **Theodore Papamarkou** [3] **Ludwig Bothmann** [1 2] **Bernd Bischl** [1 2] **David Rügamer** [1 2]

Abstract

A major challenge in sample-based inference (SBI) for Bayesian neural networks is the size and structure of the networks' parameter space. Our work shows that successful SBI is possible by embracing the characteristic relationship between weight and function space, uncovering a systematic link between overparameterization and the difficulty of the sampling problem. Through extensive experiments, we establish practical guidelines for sampling and convergence diagnosis. As a result, we present a deep ensemble initialized approach as an effective solution with competitive performance and uncertainty quantification.

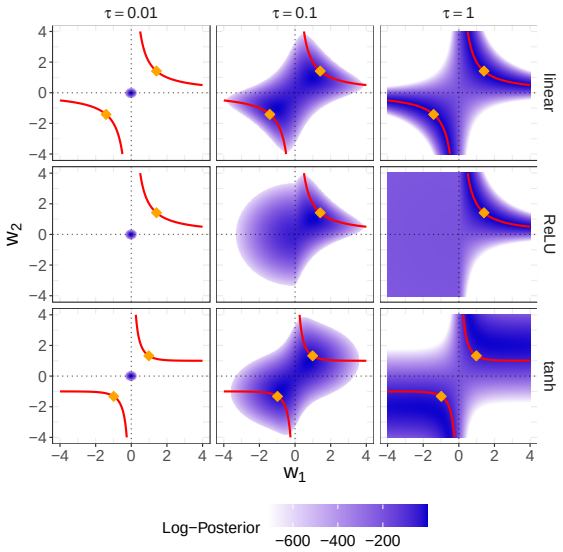


Figure 1. Log-posterior of a 1-1-1 neural network for different activation functions h (rows) using a varying $\mathcal{N}(0, \tau)$ prior (columns). *Red lines:* set of maximum likelihood solutions for the non-identifiable solution of $h(w_1 x) \cdot w_2$ with $x = 1$. *Orange diamonds:* minimum-norm solutions among the maximum likelihood set.

1. Introduction

Bayesian neural networks (BNNs) represent a principled solution to the problem of probabilistic deep learning. In the absence of analytically tractable solutions, Bayesians traditionally rely on sample-based inference (SBI) as it can in theory recover the true posterior—up to a Monte Carlo error—and requires no (restrictive) assumptions on the posterior distributional family. Consequently, SBI has enormous potential for uncertainty quantification in BNNs (Farquhar et al., 2020; Izmailov et al., 2021; Wiese et al., 2023). However, recent research has focused on local approximations that may not fully capture the multimodality of BNN posteriors, as noted in Alexos et al. (2022); Arbel et al. (2023). Such approaches potentially overlook significant portions of the posterior density, precluding a comprehensive quantification of uncertainty. The main reason behind the reluctance to adopt SBI appears to be its perceived computational demands, as discussed in Papamarkou et al. (2022); Sharma et al. (2023b). Indeed, Izmailov et al. (2021) show that SBI is possible for large state-of-the-art (SOTA) networks but comes at a high cost. Yet, scalable software solutions (e.g., JAX, Bradbury et al., 2018) and methodological adaptations (Nemeth & Fearnhead, 2021) have made SBI increasingly accessible.

While technological progress continues unabated, work studying the peculiarities of SBI remains limited. One reason for this research gap, and possibly the most fundamental problem for BNNs, is their large number of parameters. The resulting challenges are uncharted territory for established Bayesian workflow routines such as the sampling strategy, the choice of prior, and convergence monitoring. It is tempting to conclude that parameter space inference should simply be discarded in favor of addressing uncertainties directly in the function space (see, e.g., Tran et al., 2022), but this perspective risks suspending valuable insights.

[*]Equal contribution [1]Department of Statistics, LMU Munich, Munich, Germany [2]Munich Center for Machine Learning, Munich, Germany [3]Department of Mathematics, The University of Manchester, Manchester, UK. Correspondence to: David Rügamer <david@stat.uni-muenchen.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

In fact, the complex posterior topology contains structures that can be unraveled and accounted for. Most notably, overparameterization in BNNs leads to many symmetrical modes in the posterior density (Chen et al., 1993). It is less clear whether these modes are "connected" in the parameter space. Yao et al. (2022) point out that BNN posterior surfaces contain many high-density regions separated by basins of low probability, making it difficult for samplers to traverse between modes in finite time. As Fig. 1 shows, high-density regions of the posterior can behave quite differently depending on the influence of the prior. Compared to the maximum likelihood solutions (red lines), priors can induce unfavorable starting points (left column) and prevent the sampler from reaching parameter values of high predictive capacity. On the other hand, priors are able to link seemingly disconnected symmetry regions (center and right column). This observation highlights a fundamental aspect of the Bayesian paradigm: regions like the non-activated areas of ReLU networks (center row, left quadrants), which do not affect the optimization objective but are relevant to posterior uncertainty, should attain a positive probability of being sampled. In summary, the connectedness of posterior modes—and thus the difficulty of the sampling problem—strongly depends on the interplay between prior information, data, and overparameterization.

Our Contributions. Among the discussions about the merits of parameter space inference (see also Appendix A), our paper sheds light on how SBI can be successfully leveraged for BNNs. We argue that accounting for the idiosyncratic relationship between weight and function space is key to affordable SBI. From a broad range of experiments, we obtain insights into SBI's successes and failures. Key findings include the *importance of chain initialization* and the *increasing connectedness of modes in deeper layers*. Moving beyond classical measures unfit to assess BNN inference, we develop novel *convergence diagnostics that incorporate heterogeneity of layer-specific variances*. As a result, our work can be seen as an extension of recent investigations into BNN symmetries (Wiese et al., 2023), prior influence (Fortuin et al., 2022), and convergence diagnostics (Vehtari et al., 2021). Our investigations reveal what BNN posteriors demand from practical SBI, leading us to propose *Deep Ensemble Initialized MCMC (DEI-MCMC)* as a straightforward approach. DEI-MCMC achieves competitive performance and uncertainty quantification in a number of tasks.

2. Background and Related Literature

We focus on multi-layer perceptrons (MLPs). Let $f : \mathcal{X} \to \mathcal{Y}$, $\mathcal{X} \subseteq \mathbb{R}^p, \mathcal{Y} \subseteq \mathbb{R}^m$, represent a fully-connected network. We collect all weights and biases in $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$. Under the Bayesian paradigm, $\boldsymbol{\theta}$ is a random variable that admits a prior density $p(\boldsymbol{\theta})$. Updating the prior with evidence from the observed data $\mathcal{D} \in (\mathcal{X} \times \mathcal{Y})^n$ leads to a posterior density $p(\boldsymbol{\theta}|\mathcal{D}) = p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})/p(\mathcal{D})$, allowing us to express epistemic uncertainty about $\boldsymbol{\theta}$. This epistemic uncertainty governs the posterior predictive density (PPD) over the label $\boldsymbol{y}^*$ for a new observation $\boldsymbol{x}^* \in \mathcal{X}$:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int_\Theta p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\,\mathrm{d}\boldsymbol{\theta}. \tag{1}$$

The dispersion of $p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D})$ can be used to quantify predictive uncertainty about $\boldsymbol{y}^*$ (see, e.g., Murphy, 2022). Since the integral in Eq. (1) usually lacks a closed-form solution, we rely on Monte Carlo estimates of the form

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) \approx \frac{1}{S}\sum_{s=1}^{S} p\left(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}^{(s)}\right), \tag{2}$$

with S posterior samples $\boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta}|\mathcal{D})$ (Andrieu et al., 2003). Markov chain Monte Carlo (MCMC) methods, the workhorse of SBI, construct a Markov chain whose stationary distribution is the posterior density, meaning that samples $\boldsymbol{\theta}^{(s)}$ from the chain are samples from $p(\boldsymbol{\theta}|\mathcal{D})$ after the chain has converged (Gelman et al., 2013).

2.1. Multimodality in the Posterior Landscape

The non-identifiability of deep neural networks (DNNs) makes inference highly challenging (Wei et al., 2023). The weight space contains multiple *equioutput* states, i.e., different parameter vectors leading to the same functional mapping (Hecht-Nielsen, 1990). This phenomenon generates *symmetries*[1] in the BNN posterior: given equal prior probability, all equioutput parameters have equal posterior density, which leads to strong multimodality of the posterior (Wiese et al., 2023). Being equioutput induces an equivalence relation (e.g., Kůrková & Kainen, 1994):

$$\boldsymbol{\theta} \sim \boldsymbol{\theta}' \iff f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}'}(\boldsymbol{x})\,\forall\,\boldsymbol{x} \in \mathcal{X}, \;\; \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta.$$

Definition 2.1 (**Equioutput parameters**). We say that two parameters $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ are equioutput iff $\boldsymbol{\theta} \sim \boldsymbol{\theta}'$.

Equioutput parameters are related by homeomorphisms $\mathcal{F} : \Theta \to \Theta$ (Grigsby et al., 2023). Depending on the network architecture, there is a huge number of (nontrivial) transformations $\mathcal{F}$ that preserve the input-output mapping, mainly arising from neuron permutability in hidden layers and certain activation functions (Chen et al., 1993; Kunin et al., 2021). Symmetries have been studied at length for DNNs optimized by empirical risk minimization (ERM). Building on pioneering work on MLPs with odd activation

[1] We focus on what Villar et al. (2023) call *passive symmetries*, arising purely from modeling choices. In contrast, a separate field of research studies *active symmetries*, typically in the quest of making functions equivariant to symmetric properties of the physical world (e.g., Cohen & Welling, 2014).

functions (Sussmann, 1992; Chen et al., 1993; Albertini & Sontag, 1994; Kůrková & Kainen, 1994), the rise of ReLU sparked follow-up research for homogeneous activations (e.g., Neyshabur et al., 2015; Freeman & Bruna, 2017; Bona-Pellissier et al., 2021; Grigsby et al., 2023). More recently, symmetries have been rediscovered in efforts to understand the learning dynamics of DNN optimization (Brea et al., 2019; Ainsworth et al., 2023; Entezari et al., 2022), where the closely related phenomenon of overparameterization—enabling equioutput states in the first place—plays an important role (Simsek et al., 2021; Bubeck & Sellke, 2022). Interestingly, symmetries have scarcely been discussed from a Bayesian viewpoint (with a few notable exceptions, e.g., Moore, 2016; Pourzanjani et al., 2017). Since there is a well-established link between ERM with regularized objectives and Bayesian inference, the above phenomena translate from *loss* to *posterior* landscapes (e.g., MacKay, 1992). In particular, we consider the following notion:

Definition 2.2 (**Posterior symmetry**). The posterior density exhibits a symmetry w.r.t. $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ iff $p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}'|\mathcal{D})$.

Proposition 2.3. *For two parameters $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ with $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}')$, being equioutput implies a posterior symmetry, et vice versa: $\boldsymbol{\theta} \sim \boldsymbol{\theta}' \iff p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}'|\mathcal{D})$.*

Proposition 2.3 follows immediately from Bayes' theorem and the fact that equioutput parameters share the same likelihood by definition.

2.2. Mode Connectivity

The existence of multiple modes in the posterior (or loss) landscape begs the question of how inference (or optimization) is affected. Loosely speaking, a multimodal surface with strong curvature makes inference harder because the sampler will take a long time to travel between modes through areas of low posterior probability, or even infinitely long in the worst case of wholly disconnected modes. The same intuition holds for standard gradient-based optimizers that risk getting stuck in isolated local optima. As a consequence, the phenomenon of *mode connectivity* (Garipov et al., 2018) has received increased attention in the past few years. Fortunately, local optima generally seem to be connected by low-loss areas for sufficiently benign curvature of the loss hypersurface (Kuditipudi et al., 2019; Pittorino et al., 2022; Sharma et al., 2023a; Farrugia-Roberts, 2023). Draxler et al. (2018) even conjectured that there might in fact be a single manifold containing all optima. However, it is crucial to understand that connectivity is a relative notion in three regards. First, modes will rarely be disconnected in the actual sense of the word, meaning that areas between them might be assigned very low, but not zero, posterior probability. Second, as discussed in the introductory case of Fig. 1, the curvature of the landscape is modulated by the relationship between prior belief (or inductive biases; e.g., Garipov et al., 2018) and observed data (e.g., Draxler et al., 2018). Weak priors together with a small amount of training data, for instance, will produce a rather smooth surface with less pronounced modes in absence of strong evidence for any particular hypothesis. Moreover, Ainsworth et al. (2023) suggest that mode connectivity is inextricably tied to the implicit regularization of gradient-based optimizers. Third, and perhaps most importantly for the present work, theoretical mode connectivity is not all that matters. The behavior and mobility of a sampler depend directly on its design (e.g., the propensity of accepting proposals for sampling locations). If the sampler fails to traverse low-density areas, modes become effectively disconnected even if the theory suggests the existence of a non-zero-probability link between them. In addition to the data, inductive biases and regularization, algorithmic uncertainty of the sampling process thus also plays a role.

2.3. Inference Methods

Ensemble Approaches. To account for multiple basins of attraction in the posterior landscape, several inference methods using *ensembles* have been proposed. Inspired by explicit ensembling of DNNs (Lakshminarayanan et al., 2017), approaches grounded more firmly in a Bayesian mindset include modifications of the forward pass to (asymptotically) obtain samples from the Gaussian process posterior (He et al., 2020) and ensembles of Gaussian posterior approximators (MultiSWAG; Wilson & Izmailov, 2020).

Sampling-Based Inference. Research on SBI, on the other hand, is largely disregarding symmetries. Hamiltonian Monte Carlo (HMC; Neal, 2011), struggling with multimodality just like any other MCMC method, is the *de facto* gold standard (Farquhar et al., 2020; Izmailov et al., 2021). Combinations of SBI with classical optimization elements have been proposed for scalability, such as stochastic-gradient MCMC variants (Mandt et al., 2017; Zhang et al., 2020; Cobb & Jalaian, 2021), minibatch MCMC with blockwise sampling (Papamarkou, 2023), subsampling of likelihoods (Goan et al., 2023), or MCMC boosted with momentum-based information (Bieringer et al., 2023). A separate strand of research attempts to find more informative priors (Fortuin et al., 2022; Kim et al., 2023).

However, as of yet, truly feasible SBI for BNNs is still lacking. Setting out to change that, we study in great detail the state of SBI—with particular attention on the symmetry-induced multimodality of the posterior—and derive a practical solution that addresses the uncovered issues.

Table 1. Average RMSE for different models over different data sets. All neural networks have two hidden layers with 16 neurons each. The best method per data set is highlighted in bold.

DATA SET	LM	RF	DNN	DE	RS	BNN
AIRFOIL	0.716	0.255	0.252	0.239	0.250	**0.182**
BIKESHARING	0.790	**0.231**	0.374	0.365	0.362	0.253
CONCRETE	0.630	0.304	0.317	0.282	0.554	**0.258**
ENERGY	0.274	0.050	0.048	0.043	0.062	**0.037**
PROTEIN	0.863	**0.581**	0.804	0.803	1.077	0.716
YACHT	0.612	0.072	0.108	0.103	0.032	**0.022**

Table 2. Proportion of chains with RMSE performance better than an LM. Each proportion is the average of 72 experiments with 3 different train-test splits, either 1,000 or 10,000 warmup iterations, and 12 chains each with 8,000 posterior samples.

	2		8		16-16		64		32-32-32	
DATA SET	RELU	TANH	RELU	TANH	RELU	TANH	RELU	TANH	RELU	TANH
AIRFOIL	0.97	**1.00**	0.89	**1.00**	0.14	**0.97**	0.36	**0.64**	0.00	**0.67**
CONCRETE	0.89	**1.00**	0.69	**1.00**	0.00	**0.92**	0.08	**0.56**	0.00	**0.31**
ENERGY	0.65	**0.88**	0.88	**1.00**	0.06	**0.97**	0.17	**0.53**	0.00	**0.39**
YACHT	0.75	**0.82**	0.83	**1.00**	0.14	**0.97**	0.33	**0.64**	0.00	**0.47**

3. Experimental Setup

In Sections 4–7, we highlight findings from an extensive grid of experiments with more details and results in Appendix C. Our code is available under this URL. Section 4 examines the general feasibility of SBI, focusing on whether SBI can achieve SOTA performance. Section 5 investigates how the multimodality of posteriors influences SBI performance, after which Section 6 discusses ways of handling multimodality to make SBI more practical. Section 7 summarizes the findings and proposes DEI-MCMC as a competitive solutions for SBI in BNNs.

Data sets and models. We consider several regression data sets from the UCI benchmark (Dua & Graff, 2017): `airfoil`, `concrete`, `energy` and `yacht`, as well as two larger data sets, `bikesharing` and `protein`. The MLPs considered vary in width (up to 64 hidden units) and depth (up to seven layers). We investigate not only differences in model architecture, the activation function and the weight priors, but also in the choice of the sampler and its configuration.

Sampling. In particular, we employ HMC and the No-U-Turn Sampler (NUTS; Hoffman & Gelman, 2014). The latter implements HMC with auto-tuning of the trajectory length and step size, promising a considerable speed-up over optimizing these critical hyperparameters separately. In all experiments, we run up to 12 chains with 8,000 samples each for the smaller and 4,000 samples for the larger data sets. We use a fixed target acceptance probability of 0.8 and warmup phases of 10,000 steps unless stated otherwise. While we also examined the effect of a longer warmup by running up to 100,000 steps, this did not improve performance in almost all cases.

Performance metrics. BNN predictions are computed via (empirical) Bayesian model averaging. In particular, each posterior sample induces a model from which we obtain the parameterization of a PPD conditioned on a given test observation from a test set $\mathcal{D}_{\text{test}} \in (\mathcal{X} \times \mathcal{Y})^{n_{\text{test}}}$ (20% of the data in all experiments). As a point estimate, we compute the expectation over those conditional distributions via Monte Carlo integration. Since we focus on regression, we use the test root mean squared error (RMSE) to assess predictive performance. The quality of uncertainty quantification can be measured by the log-PPD (LPPD; Eq. (7) in Appendix B) on test data[2]. We also assess the calibration of the samples by comparing nominal and empirical coverage of credibility intervals of the PPD.

4. General Feasibility

We start by investigating the general feasibility of SBI with HMC and NUTS. We initially focus on predictive performance, studying whether it is at all possible to run these samplers and perform on par with the non-Bayesian SOTA. We also analyze in Section 6 whether the successful configurations provide useful uncertainty quantification. While various studies reported on the performance of BNNs based on SBI (e.g., Izmailov et al., 2021), we are not aware of any other systematic investigation beyond performance.

Insight: For certain architectures and sampling algorithms, BNNs using SBI can achieve SOTA performance.

We hypothesize that published results obscure the fact that samplers cannot deal with all existing BNN posterior problems off-the-shelf. As a sanity check, we match the RMSE performance of SBI against that of a linear model (LM). Using the sampled chains that outperform the LM, we then compare the resulting BNN performance, i.e., the ensemble of all chains and their 8,000 samples, to a random forest (RF; tuned with optuna (Akiba et al., 2019)) as a stronger baseline, a (non-Bayesian) DNN of the same architecture, and a deep ensemble (DE; Lakshminarayanan et al., 2017) with 12 members. The DNN and DE models are trained with Adam (Kingma & Ba, 2015, further details in Appendix D) and the BNNs are sampled using NUTS with unit Gaussian prior and tanh activation. In order to show the ensemble effect of the BNN, we also report the performance of the model induced by a single random sample (RS) from the chains of the BNN.

[2]Meaningful inference should yield (L)PPDs under which the true label $\boldsymbol{y}^*$ has high posterior density. Intuitively, this can only be the case for appropriate PPD dispersion (too concentrated—$\boldsymbol{y}^*$ will frequently not coincide with the PPD mode and thus have low density; too diffuse—no value is assigned high density).

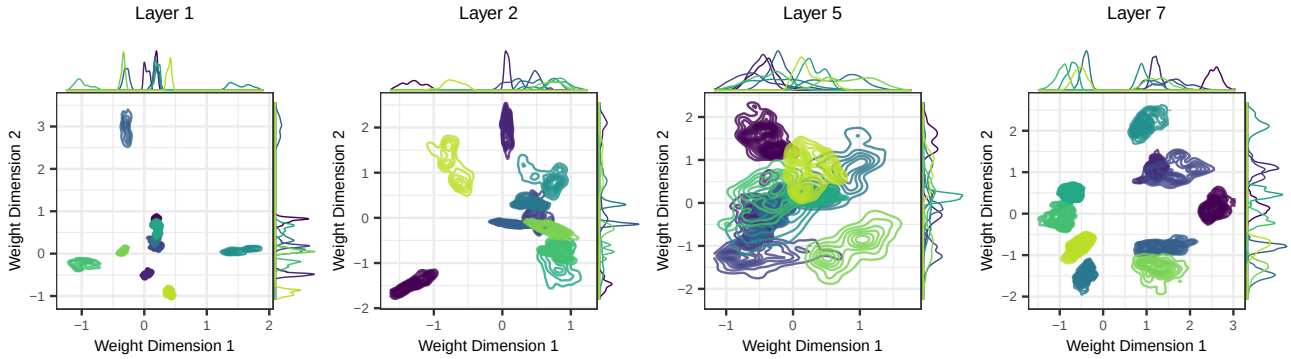


Figure 2. Bivariate density plot of 10 chains (colors) of two randomly selected weights in the first, second, fifth, and final layer (from left to right) of a seven-layer BNN showing the varying degree of mode connectedness.

General feasibility and tanh SOTA performance. Using chains that outperform the LM, samplers indeed yield SOTA performance across different experimental settings for tanh activation (cf. Table 1). While individual samples (RS) cannot outperform the RF, DNN or DE, their ensemble supersedes the baselines in performance. We discuss the benefits of multi-chain SBI in Sections 5 and 7.

The unboundedness problem. In contrast to the results of tanh networks, off-the-shelf SBI does not produce meaningful results for ReLU-, SiLU- or LeakyReLU-activated BNNs across different architectures, prior choices, and data sets. Instead, we observe what we call *dying samplers*, getting stuck almost immediately and never leaving the area of the starting value (cf. Appendix C). This becomes evident in Table 2 reporting the share of chains that perform better than an LM in terms of RMSE for ReLU and tanh activation. The number of successful chains with ReLU is notably smaller compared to tanh, which holds true also for other (unbounded) activation functions (cf. Fig. 14 in the Appendix). As is known from classical DNN optimization, weight variances in networks with unbounded activation functions can explode if not properly initialized (see, e.g., He et al., 2015). Similarly, if priors remain constant across layers with variance independent of the size of each layer, we observe that SBI will not produce chains better than a simple baseline. Hence, choosing bounded activation functions might help to achieve good performance. However, even for tanh, Table 2 shows that the more complex the architecture, the less reliable the samplers become. We elaborate on a potential solution of this problem in Section 7.

Superiority of NUTS. NUTS outperforms HMC across different data sets and activation functions (cf. Table 5 in the Appendix). Overall, HMC with fixed hyperparameters can only produce chains with better performance than the LM in 1% of all experiments, suggesting that the refined handling of sampling trajectories by NUTS is indispensable for BNNs. Tuning HMC's hyperparameters, in turn, might be less efficient than NUTS, with the latter performing well out of the box in many cases.

5. Multimodality of Posteriors

For a deeper understanding of the previous results, we now turn to the peculiarities of BNNs and how they affect SBI. A key insight in this respect refers to the symmetries caused by overparameterization (cf. Fig. 1). We observe that this redundancy becomes more pronounced with network depth.

Insight: In BNNs, the uncertainty of the weight distributions progressively increases in deeper layers. This means that more mode connectivity is observed in deeper layers.

Multimodality of the posterior. Fig. 1 suggests that prior influence can induce a *merging of modes*. This means that practically irrelevant likelihood regions between two posterior modes increase in posterior probability through the prior contribution, allowing samplers to traverse from one mode to another. In apparent contrast to this initial hypothesis on mode connectivity, we do observe multimodality across various settings for different prior variances. Even in larger networks and smaller data sets, the model complexity and data signal may outweigh prior information and (initially) induce disconnected modes in the posterior (see Fig. 2, left, visualizing the densities of weights in different layers marginalized for two dimensions). Thus, the general existence of connected mode surfaces is not guaranteed.

More movement deeper down. It turns out, however, that this phenomenon requires a more nuanced analysis. As shown in Fig. 2, the posterior density is heavily concentrated on disconnected modes in the first layer, but then becomes more diffuse and thus connected in deeper parts of the BNN before concentrating again in the output layer. Our observation is reflected by chains growing notably more variable in deeper layers (cf. Fig. 3 and 4); only in the last layer does the variance decrease again. The explanation for

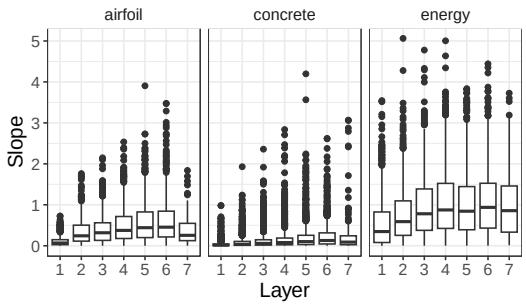


Figure 3. Range of "motion" of chains per layer (x-axis) and selected data sets (facets). Boxplots show the distribution of absolute values of slopes obtained by fitting a linear regression for each chain to capture chain trends during the sampling process, clearly indicating more movement in deeper layers.

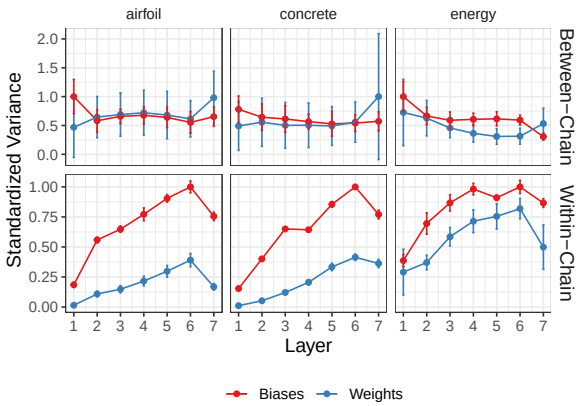


Figure 4. Mean and standard deviation of the between- and within-chain variance (in different rows) separated by layer (x-axis) for different data sets (columns) of a seven-layer BNN.

this behavior can be found in the varying degree of overparameterization throughout the network. We conjecture that disconnected modes occur mainly in layers with limited flexibility, i.e., where the role of the neurons is controlled by their close connection to the input (or output) that populates a certain manifold. Hidden-layer weights, by contrast, have vastly more degrees of freedom in assuming diverse values and swapping their roles in creating input embeddings as long as the output is realized in an appropriate range.

Role of network layers. Our intuition about the roles of different layers echoes recent findings about the varying robustness of layers to re-initialization during training (where the initial layer, in accordance with our results, proves much more sensitive to resetting the parameter values mid-training than intermediate ones; Zhang et al., 2022). The layerwise investigation also has interesting relations to the mode connectivity studies in Izmailov et al. (2021). Taking the notion of mode connectivity to a per-layer level, work concurrent to ours (Adilova et al., 2024) suggests that intermediate layers play a different role for the success of layerwise neuron alignment (interpolating between multiple independent solutions; cf. Ainsworth et al., 2023) than the first or last one. We further believe our results could shed light on the effectiveness of methods such as subspace inference (Izmailov et al., 2020; Dold et al., 2024) that move or sample in directions of quasi-constant loss. More immediately, as we discuss in the next section, they have important consequences for practical SBI.

6. Practical SBI

6.1. How to Handle Multimodality in Practice?

Given the existence of multimodal posterior surfaces that samplers do not seem to be able to traverse, the question arises of how SBI can still achieve useful results. One possibility (also mentioned in Riou-Durand et al., 2023; Wiese et al., 2023) is to run multiple chains to cover as many modes as possible. Covering all modes seems to be an impossible endeavor given that even in small BNNs the number of symmetric modes easily exceeds 10^{200} (Wiese et al., 2023). However, our results in the previous section give reason for hope and suggest that the number of disconnected modes (and thus required chains) does not simply scale with the complexity of the network. As overparametrization induces smoother posterior surfaces for deeper layers that samplers can explore much better (cf. Fig. 3), it is likely only necessary to cover the few distinct modes in earlier layers.

We now investigate the joint influence of the number of chains and the number of samples on the predictive and uncertainty performance of SBI.

Insight: Multiple samples from multiple chains improve predictive performance and uncertainty quantification.

Performance and uncertainty. Fig. 5 (further examples in Appendix C.3) shows a clear upward trend of LPPD in both the number of chains and the number of samples. The same general pattern can be observed for the RMSE (Appendix C.3). Notably, the LPPD increases more strongly for multiple chains than the RMSE, suggesting the importance of covering multiple modes, but we also see performance gains by exploring the region around modes with more samples. It seems promising that we obtain good results with a maximum of 12 chains despite the large number of symmetries. This is in line with findings from Wiese et al. (2023) who ran more than 1,200 chains but also observed early saturation of performance metrics.

Predictive coverage. Bayesian inference enables the computation of predictive credibility intervals which should be calibrated and cover the true prediction. Fig. 6 (further results in Appendix C.3) shows the coverage of credibility

intervals for varying amounts of samples (left) and chains (right). For a calibrated model (represented by the diagonal), the observed labels fall within the $\alpha\%$ credibility interval in $\alpha\%$ of cases. With increasing chains and samples, we can see a trend from overconfidence to close-to-nominal coverage. These results confirm that the use of multiple chains and samples indeed produces well-calibrated posteriors.

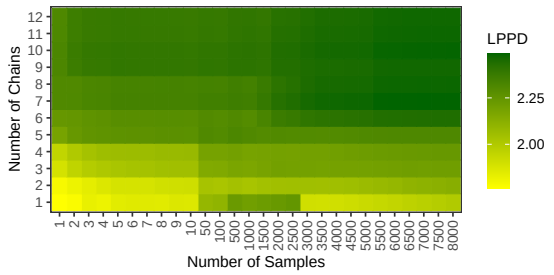


Figure 5. Grid for the `energy` data set showing the change in LPPD with more samples and chains.

6.2. Convergence Diagnostics

The previous experiments investigated the performance of different numbers of samples per chain. However, in practice, it is often not obvious when to stop the sampling process. While there are well-established convergence diagnostics for Bayesian statistics, it is unclear whether these work well for BNNs. In contrast to the existing literature, which tends to disregard convergence analysis of the parameter space, we will discuss both parameter and function space diagnostics. For both, the *de facto* standard for measuring convergence is the *rank-normalized split* $\widehat{R}^{(2)}$ as introduced in Vehtari et al. (2021, see Appendix B for a definition and functional extension).

Insight: Due to symmetries and large differences in within-chain variance across layers, classical diagnostics are not useful for checking convergence in BNNs.

Parameter space convergence. The classical $\widehat{R}$ diagnostic assumes identifiable parameters, which is not given in BNNs due to the aforementioned symmetries. Our experiments in Appendix B also confirm empirically that $\widehat{R}$ is not a suitable metric. Another pitfall we can infer from our results is the aggregation of $\widehat{R}$ over all model parameters. As $\widehat{R}$ is normalized using the within-chain variance, it decreases in the later layers of the BNN due to the increase in variance (Fig. 4), leading to the false conclusion of convergence of the whole network when averaging across all weights.

Thus, we argue that parameter space convergence should be measured both chain- and layer-wise. For the chain-wise convergence, we propose the $\widehat{cR}^{(\kappa)}$ diagnostic (defined in Appendix B), which splits a chain's sample path into κ subchains of equal length as the basis for the $\widehat{R}$ calculation. Empirically, we observe $\widehat{cR}^{(\kappa=4)}$ values close to or lower than 1.1 for well-performing and calibrated models, indicating chain-wise convergence (cf. Fig. 7, left, and 8).

Function space convergence. Various post-hoc methods exist to evaluate the function space convergence of BNNs, all using a hold-out test data set (see Appendix B). While some report the pointwise $\widehat{R}$ values for the PPD samples of each test data point (Izmailov et al., 2021) and visually analyze their distribution (cf. Fig. 10), others calculate the $\widehat{R}$ for a goodness-of-fit measure over the test data set (e.g., log-likelihood) to obtain a single diagnostic (Fortuin et al., 2022). Both approaches penalize disagreement between chains. Taking into account the disconnectedness of modes in earlier layers, we cannot, however, expect the convergence of all chains to a common function outcome. In particular, our experiments show the existence of substantial between-chain variance in the function space for a well-calibrated model (e.g., Fig. 11). We argue that a proper function space convergence metric should not penalize these chain-wise differences as visiting different modes is essential for well-working SBI.

Therefore, we propose to monitor the convergence of each chain individually by the cumulative LPPD, as defined in Eq. (8) in the Appendix and shown in, e.g., Fig. 7 (right). By running multiple chains, we can compare the different cumulative LPPD paths and thereby obtain a better understanding of the convergence of each chain and the difference in chain performance. This can also be used as an early-stopping criterion of the sampling process for potentially converged or diverging chains, freeing resources and allowing to start new chains for more efficiency.

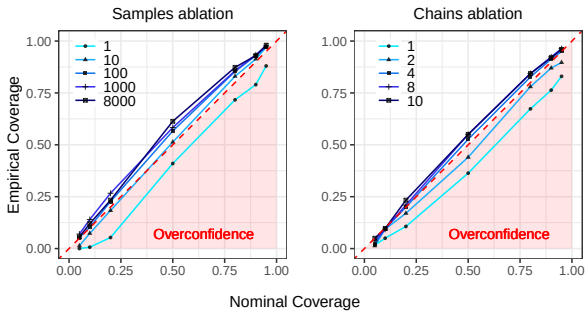


Figure 6. Nominal vs. empirical coverage of posterior credibility intervals for different numbers of samples using 10 chains everywhere (left) and different numbers of chains using 100 samples each (right) for the `airfoil` data set, NUTS with 10,000 warmup, two hidden layers of 16 neurons each, tanh activation, and unit Gaussian priors.

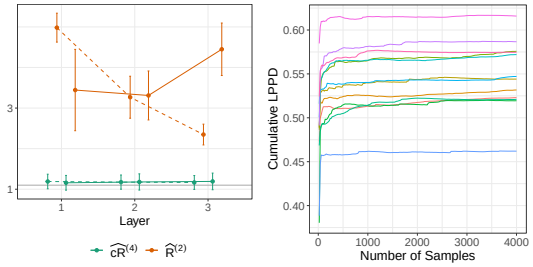


Figure 7. Left: Comparison between $\widehat{R}^{(2)}$ and $\widehat{cR}^{(4)}$ for weights and biases (in different linetypes). *Right:* Cumulative LPPD across the process of collecting 4,000 samples and different chains (colors). Both plots are based on the `bikesharing` data set.

7. Dying Sampler & Deep Ensemble Initialization

Equipped with a clearer understanding of BNN posteriors and practical strategies to obtain calibrated SBI, we now address the unresolved problem of chains that perform worse than a simple baseline, leading to samplers "dying" soon after initialization (cf. Section 4 and Fig. 16 in the Appendix).

7.1. The Dying Sampler Problem

Although Section 4 established that the use of bounded activation functions largely avoids the dying sampler problem, it would be unsatisfactory to simply dismiss ReLU BNNs as infeasible or refrain from using larger architectures. While previously discussed results on prior variance offer a possible explanation for this phenomenon, a—if not the most important—cause is already evident in Fig. 1: *Starting values for chains might prompt the sampler to get stuck or not explore meaningful hypotheses*.

For example, as a consequence of the ReLU activation function being zero for negative input signals, many neurons—just by chance—will not be activated, creating low-density regions at the locations of the weights associated with those neurons (the white area in Fig. 1). Subsequent proposals will then also end up in regions with a posterior probability of practically zero. We can thus draw an analogy to the "dying ReLU problem" (Lu et al., 2020) caused by zero gradients of non-activated ReLU neurons in optimization.

7.2. Deep Ensemble Initialized MCMC

As already elaborated in Section 4, using a different initialization solves convergence issues such as the dying sampler problem. This motivates our concluding proposal for practically useful SBI, which we call *Deep Ensemble Initialized MCMC (DEI-MCMC)*. Inspired by their similarity to DEs one could also think of them as a *Bayesian Deep Ensemble*. Specifically, we suggest running a standard optimization procedure for M networks in a non-Bayesian fashion, and using the resulting M sets of network weights as initial proposals for the samplers. Note that this does not require changing the prior distribution. We combine the information of all chains in the classical Bayesian sense by merging their empirical distributions. DEI-MCMC has several attractive properties, which we discuss in the following.

Valid starting values. As priors with support on the entire domain do not per se restrict the admissible values of weights (but activation functions in the model might), we can expect initialization with the M sets of weights to yield valid results. In other words, the starting values will produce non-zero posterior values and, hence, solve the problem of the dying sampler.

Expected improvement. DEI-MCMC is equivalent to DEs in the edge case where we only obtain one sample from the posterior (the initial weight). As DEs have repeatedly shown top-notch quantifying quantification, this means that DEI-MCMC is likely at least as good as DEs in this respect. In our experiments, DEI-MCMC indeed provides better LPPD for as few as ten samples (cf. Table 3).

A modular Bayesian toolbox. Similar to the Laplace approximation (Daxberger et al., 2021), DEI-MCMC can be used post-optimization, allowing to combine prior knowledge of any kind with a non-Bayesian network into a BNN. In particular, DEI-MCMC is applicable to pre-trained networks.

Flexibility. While DEI-MCMC is as expensive as DEs by design and requires further computation for additional sampling, this second process can be flexibly adjusted to the availability of resources and computing time. Furthermore, since our previous results suggest that a few samples already yield very good performance when using multiple chains, it might be possible to shorten the sampling step considerably by saving on warmup iterations.

7.3. Numerical Experiments

In order to investigate the effectiveness of DEI-MCMC, we run an additional benchmark study. Proving that the proposed warm start can solve the dying sampler problem, we use BNNs with ReLU activation and compare the performance of DEI-MCMC using all chains to those of an equivalent DNN and DE. In other words, we do not need to filter chains for the comparisons as done in Section 4.

Results. We summarize our findings in Table 3. As becomes evident from the comparisons with the LM and DE, the warm starts of samplers using the M different network solutions avoid any dying sampler problems (no model shows worse performance than the baseline LM). Furthermore, we see the added benefit of sampling around the DE solutions when continuing the sampling from these solutions after a very short warmup phase of 100 steps, suggesting that this process supplies valuable information to the model both for

Table 3. Average RMSE and LPPD values (± their standard deviations) of the LM, a classic DNN, a DE, and our method (i.e., DEI-MCMC) using different amounts of samples (in brackets) for all benchmark data sets (in rows; abbreviated by their first letter). All networks have two hidden layers with 16 neurons each and ReLU activations. The best method per data set is highlighted in bold.

	RMSE (↓)						LPPD (↑)					
	LM	DNN	DE	OURS (10)	OURS (100)	OURS (1000)	LM	DNN	DE	OURS (10)	OURS (100)	OURS (1000)
A	0.72±0.02	0.27±0.01	0.24±0.01	0.21±0.01	0.21±0.01	**0.21±0.01**	-1.09±0.03	-0.17±0.08	0.19±0.05	0.50±0.01	0.53±0.02	**0.58±0.03**
B	0.77±0.01	0.31±0.00	0.30±0.00	0.26±0.01	0.25±0.01	**0.25±0.01**	-1.16±0.01	0.35±0.00	0.42±0.00	0.60±0.02	0.61±0.02	**0.63±0.02**
C	0.63±0.02	0.35±0.04	0.32±0.05	0.40±0.19	0.70±0.71	**0.31±0.05**	-0.96±0.03	-0.99±0.38	-0.08±0.20	0.05±0.27	0.19±0.11	**0.23±0.12**
E	0.27±0.02	0.05±0.00	0.04±0.01	0.04±0.00	0.04±0.00	**0.04±0.00**	-0.13±0.06	1.60±0.13	1.93±0.05	1.91±0.28	1.94±0.25	**2.06±0.20**
P	0.85±0.01	0.77±0.01	0.77±0.01	0.71±0.00	0.71±0.00	**0.70±0.00**	-1.25±0.01	-1.06±0.01	-1.05±0.01	-0.78±0.02	-0.77±0.02	**-0.75±0.02**
Y	0.61±0.07	0.05±0.00	0.03±0.01	0.03±0.01	0.03±0.01	**0.03±0.01**	-0.94±0.13	1.74±0.40	2.55±0.05	2.97±0.17	3.00±0.20	**3.08±0.20**

predictive (RMSE) and uncertainty (LPPD) performance. It should be noted that, although DEI-MCMC tends to benefit from longer sampling, there is a diminishing return to extending the sampling phase beyond the 1,000 samples reported in the last column of Table 3, with only marginal improvements from drawing even up to 10,000 samples. While the best cold-started BNNs required 10,000 warmup steps, DEI-MCMC achieves competitive performance with a 100× shorter warmup phase. We thus conclude that DEI-MCMC, though conceptually simple, offers a consistent and robust improvement over classical BNNs.

8. Discussion

In this work, we discussed sampling-based inference in BNNs. We argue that sampling from a posterior in these networks is not only feasible but can achieve SOTA results when accounting for the nature of posterior landscapes. Through extensive experiments with different BNN architectures, we present insights into the successes and limitations of SBI in this context. Critical findings include performance differences of samplers for bounded and unbounded activation functions and the increasing connectedness of modes in deeper layers. Following these results, we recommend a multi-chain and multi-sample strategy using NUTS, with a convergence diagnostic that accounts for heterogeneous variances in different layers. In order to allow samplers to also explore highly multimodal landscapes, particularly those induced by ReLU networks, we propose a novel approach, *Deep Ensemble Initialized MCMC (DEI-MCMC)*, which uses optimized networks as a starting point for posterior sampling. These ensembles can be seen as a modular Bayesian toolbox applicable to any network post-optimization independent of the state of prior knowledge.

Limitations and future work. Due to the multitude of experiments, as well as the long duration of sampling procedures, the analysis presented in this paper is limited in some aspects. In particular, only full-batch sampling routines were tested. An exciting target for future analysis, therefore, is to explore the potential of SG-MCMC samplers in the context of DEI-MCMC, as well as an extension to larger data sets. Likewise, the performance of DEI-MCMC in uncertainty-related downstream tasks, such as out-of-distribution detection, remains an open question. It would further be interesting to deepen our findings in conjunction with the insights from mode connectivity (Garipov et al., 2018) and subspace research (Izmailov et al., 2020). Finally, inspired by our tentative results and the work by Adilova et al. (2024), all these avenues of future work should be pursued adopting a layerwise lens.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

LW is supported by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the German Federal Ministry of Education and Research.

References

Adilova, L., Andriushchenko, M., Kamp, M., Fischer, A., and Jaggi, M. Layer-wise Linear Mode Connectivity. In *12th International Conference on Learning Representations (ICLR)*, 2024.

Ainsworth, S. K., Hayase, J., and Srinivasa, S. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *Proceedings of the Eleventh International Conference on Learning Representations*, 2023.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902, 2019.

Albertini, F. and Sontag, E. D. Uniqueness of weights for neural networks. In Mammone, R. J. (ed.), *Artificial Neural Networks for Speech and Vision*, number 4 in

Chapman & Hall Neural Computing Series. Chapman & Hall, London, 1. ed edition, 1994.

Alexos, A., Boyd, A., and Mandt, S. Structured Stochastic Gradient MCMC. In *Proceedings of the 39 Th International Conference on Machine Learning*, 2022.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43, 2003.

Arbel, J., Pitas, K., Vladimirova, M., and Fortuin, V. A Primer on Bayesian Neural Networks: Review and Debates, 2023.

Bieringer, S., Kasieczka, G., Steffen, M. F., and Trabs, M. AdamMCMC: Combining Metropolis Adjusted Langevin with Momentum-based Optimization, 2023.

Bona-Pellissier, J., Bachoc, F., and Malgouyres, F. Parameter identifiability of a deep feedforward ReLU neural network, 2021.

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

Brea, J., Simsek, B., Illing, B., and Gerstner, W. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape, 2019.

Bubeck, S. and Sellke, M. A Universal Law of Robustness via Isoperimetry, 2022.

Chen, A. M., Lu, H.-m., and Hecht-Nielsen, R. On the Geometry of Feedforward Neural Network Error Surfaces. *Neural Computation*, 5(6):910–927, 1993.

Cobb, A. D. and Jalaian, B. Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2021.

Cohen, T. and Welling, M. Learning the Irreducible Representations of Commutative Lie Groups. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. Laplace Redux – Effortless Bayesian Deep Learning. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

Dold, D., Rügamer, D., Sick, B., and Dürr, O. Semi-structured subspace inference. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research. PMLR, 2024.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially No Barriers in Neural Network Energy Landscape. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks. In *Proceedings of the Tenth International Conference on Learning Representations*, 2022.

Fanaee-T, H. Bike Sharing Dataset. UCI Machine Learning Repository, 2013.

Farquhar, S., Smith, L., and Gal, Y. Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Farrugia-Roberts, M. Functional Equivalence and Path Connectivity of Reducible Hyperbolic Tangent Networks, 2023.

Fortuin, V., Garriga-Alonso, A., Ober, S. W., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. Bayesian neural network priors revisited. In *International Conference on Learning Representations*, 2022.

Freeman, C. D. and Bruna, J. Topology and Geometry of Half-Rectified Network Optimization. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 10, 2018.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian Data Analysis Third Edition (with Errors Fixed as of 15 February 2021)*. Published online, 2013.

Gelman, A., Hwang, J., and Vehtari, A. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014.

Goan, E., Perrin, D., Mengersen, K., and Fookes, C. Piecewise Deterministic Markov Processes for Bayesian Neural Networks. In *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

Grigsby, J. E., Lindsey, K., and Rolnick, D. Hidden symmetries of ReLU networks. In *Proceedings of the 40 Th International Conference on Machine Learning*, 2023.

He, B., Lakshminarayanan, B., and Teh, Y. W. Bayesian Deep Ensembles via the Neural Tangent Kernel. In *34th Conference on Neural Information Processing Systems*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hecht-Nielsen, R. ON THE ALGEBRAIC STRUCTURE OF FEEDFORWARD NETWORK WEIGHT SPACES. In *Advanced Neural Computers*, pp. 129–135. Elsevier, 1990.

Hoffman, M. D. and Gelman, A. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1351–1381, 2014.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. Subspace Inference for Bayesian Deep Learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 1169–1179, 2020.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. What Are Bayesian Neural Network Posteriors Really Like? In *Proceedings of the 38th International Conference on Machine Learning, PMLR 139,*, 2021.

Kim, K., Ma, E.-Y., Choi, J., and Kim, H. Inverse-Reference Priors for Fisher Regularization of Bayesian Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8264–8272, 2023.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Kuditipudi, R., Wang, X., Lee, H., Zhang, Y., Li, Z., Hu, W., Arora, S., and Ge, R. Explaining Landscape Connectivity of Low-cost Solutions for Multilayer Nets. In *33rd Conference on Neural Information Processing Systems*, 2019.

Kunin, D., Sagastuy-Brena, J., Ganguli, S., Yamins, D. L. K., and Tanaka, H. Neural Mechanics: Symmetry and Broken Conservation Laws in Deep Learning Dynamics. In *Proceedings of the Ninth International Conference on Learning Representations*, 2021.

Kůrková, V. and Kainen, P. C. Functionally Equivalent Feedforward Neural Networks. *Neural Computation*, 6 (3):543–558, 1994.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

Lu, L., Shin, Y., Su, Y., and Em Karniadakis, G. Dying ReLU and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28(5):1671–1706, 2020.

MacKay, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3): 448–472, 05 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic Gradient Descent as Approximate Bayesian Inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

Moore, D. A. Symmetrized Variational Inference. In *30th Conference on Neural Information Processing Systems*, 2016.

Murphy, K. P. *Probabilistic Machine Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, 2022.

Neal, R. M. *MCMC Using Hamiltonian Dynamics*. Chapman & Hall / CRC Press,, 2011.

Nemeth, C. and Fearnhead, P. Stochastic gradient mrkov chain monte carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

Neyshabur, B., Salakhutdinov, R., and Srebro, N. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. In *NeurIPS*, 2015.

Ortigosa, I., Lopez, R., and Garcia, J. A neural networks approach to residuary resistance of sailing yachts prediction. In *Proceedings of the International Conference on Marine Engineering (MARINE)*, volume 2007, pp. 250, 2007.

Papamarkou, T. Approximate blocked Gibbs sampling for Bayesian neural networks. *Statistics and Computing*, 33, 2023.

Papamarkou, T., Hinkle, J., Young, M. T., and Womble, D. Challenges in Markov Chain Monte Carlo for Bayesian Neural Networks. *Statistical Science*, 37(3), 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Phan, D., Pradhan, N., and Jankowiak, M. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

Pittorino, F., Ferraro, A., Perugini, G., Feinauer, C., Baldassi, C., and Zecchina, R. Deep networks on toroids: Removing symmetries reveals the structure of flat regions in the landscape geometry. In *Proceedings of the 39th International Conference on Machine Learning, PMLR 162*, 2022.

Pourzanjani, A. A., Jiang, R. M., and Petzold, L. R. Improving the Identifiability of Neural Networks for Bayesian Inference. In *Second Workshop on Bayesian Deep Learning*, 2017.

Riou-Durand, L., Sountsov, P., Vogrinc, J., and Margossian, C. C. Adaptive Tuning for Metropolis Adjusted Langevin Trajectories. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Sharma, E., Kwok, D., Denton, T., Roy, D. M., Rolnick, D., and Dziugaite, G. K. Simultaneous linear connectivity of neural networks modulo permutation. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023a.

Sharma, M., Farquhar, S., Nalisnick, E., and Rainforth, T. Do Bayesian Neural Networks Need To Be Fully Stochastic? In *Proceedings of the 26th International Conference on Artificial Intel Ligence and Statistic*, 2023b.

Simsek, B., Ged, F., Jacot, A., Spadaro, F., Hongler, C., Gerstner, W., and Brea, J. Geometry of the Loss Landscape in Overparameterized Neural Networks: Symmetries and Invariances. In *Proceedings of the 38 Th International Conference on Machine Learning*, 2021.

Stan Development Team. Stan modeling language users guide and reference manual, 2024. URL https://mc-stan.org.

Sussmann, H. J. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 5(4):589–593, 1992.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. All you need is a good functional prior for bayesian deep learning. *The Journal of Machine Learning Research*, 23 (1):3210–3265, 2022.

Tsanas, A. and Xifara, A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49: 560–567, 2012.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. Rank-normalization, folding, and localization: an improved r for assessing convergence of mcmc (with discussion). *Bayesian Analysis*, 16(2):667–718, 2021. doi: 10.1214/20-BA1221.

Villar, S., Hogg, D. W., Yao, W., Kevrekidis, G. A., and Schölkopf, B. Towards fully covariant machine learning, 2023.

Wei, S., Murfet, D., Gong, M., Li, H., Gell-Redman, J., and Quella, T. Deep Learning Is Singular, and That's Good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10473–10486, 2023.

Wiese, J. G., Wimmer, L., Papamarkou, T., Bischl, B., Günnemann, S., and Rügamer, D. Towards efficient posterior sampling in deep neural networks via symmetry removal. In *Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pp. 459–474. Springer International Publishing, 2023.

Wilson, A. G. and Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 4697–4708, 2020.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. URL https://www.sciencedirect.com/science/article/pii/0169743987800849.

Yao, Y., Vehtari, A., and Gelman, A. Stacking for Nonmixing Bayesian Computations: The Curse and Blessing of Multimodal Posteriors. *Journal of Machine Learning Research 23*, 2022.

Yeh, I.-C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

Zhang, C., Bengio, S., and Singer, Y. Are All Layers Created Equal? *Journal of Machine Learning Research*, 23:1–28, 2022.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *Proceedings of the Eighth International Conference on Learning Representations*, 2020.

A. Inference in Weight vs. Function Space

Some authors suggest completely disregarding the parameter space (Tran et al., 2022; Arbel et al., 2023) since we arguably care more about the functions learned by a DNN than about single weights in a million-dimensional space. However, opting for inference over weights but then deliberately turning a blind eye to everything that happens in the weight space seems unsatisfactory. Other authors argue that expressing prior beliefs in function space (Tran et al., 2022) is the more meaningful approach when researchers rarely have actual prior knowledge about BNN weights. Formulating a prior about a complex functional mapping is, however, not necessarily a simpler problem. While each side has its valid points, a general discussion of the meaningfulness of sampling and inference in the parameter space of neural networks is beyond the scope of this work. At this point, we just point out briefly that sampling-based inference will remain highly relevant for simpler models that offer some degree of interpretability (e.g., in statistics), and since there is no clear cut-off for when a model is too complex, researchers will eventually use sampling-based inference for BNNs, in particular as computational resources and performance increase.

B. Details on Convergence Diagnostics

B.1. Parameter Space

The two standard diagnostics for convergence of MCMC methods are currently the rank-normalized split-$\widehat{R}$ (Gelman et al., 2013; Vehtari et al., 2021) and the effective sample size (ESS; Vehtari et al., 2021). We extend the rank-normalized split-$\widehat{R}$ to a functional version. For a transformation $\psi(\cdot)$ and posterior samples $\boldsymbol{\theta}^{(k,s)}$ from chain $k \in 1, \dots, K$, $s \in 1, \dots, S$, we define the diagnostic as follows:

Definition B.1 ($\widehat{R}_\psi$)**.** Let $\psi : \mathbb{R}^d \to \Psi; \boldsymbol{\theta} \mapsto \psi(\boldsymbol{\theta})$ be a transformation function transforming the parameter samples into some metric of interest. $\bar{\boldsymbol{\theta}}^{(k,\cdot)} = \frac{1}{S}\sum_{s=1}^{S} \psi(\boldsymbol{\theta}^{(k,s)})$ and $\gamma_k^2 = \frac{1}{S-1}\sum_{s=1}^{S}\left(\psi(\boldsymbol{\theta}^{(k,s)}) - \bar{\boldsymbol{\theta}}^{(k,\cdot)}\right)^2$ denote the (potentially vector-valued) empirical mean and variance of the functional of the posterior samples of chain k, respectively. In case $\Psi \nsubseteq \mathbb{R}$, i.e., ψ is not mapping to (a subset of) $\mathbb{R}$ but results in a vector, the $(\cdot)^2$ is supposed to be interpreted element-wise applied to all dimensions of Ψ individually. In all cases considered in this paper, this means that ψ either maps a sample to a scalar statistic or creates a statistic for every of the d dimensions of $\boldsymbol{\theta}$. Further, denote by $\bar{\boldsymbol{\theta}}^{(\cdot,\cdot)} = \frac{1}{K}\sum_{k=1}^{K} \bar{\boldsymbol{\theta}}^{(k,\cdot)}$ the grand mean over all chains and samples. Then, we can define the within-chain variance W and the between-chain variance B as

$$B = \frac{S}{K-1}\sum_{k=1}^{K}\left(\bar{\boldsymbol{\theta}}^{(k,\cdot)} - \bar{\boldsymbol{\theta}}^{(\cdot,\cdot)}\right)^2, \tag{3}$$

$$W = \frac{1}{K}\sum_{k=1}^{K}\gamma_k^2, \tag{4}$$

where $(\cdot)^2$ is again supposed to be interpreted element-wise in case the means represent vectors. Normalizing B with W results in (a potentially vector-valued)

$$\widehat{R}_\psi = \sqrt{\frac{\frac{S-1}{S}W + \frac{1}{S}B}{W}}. \tag{5}$$

For ψ being the identity we recover the $\widehat{R}$ as defined in Vehtari et al. (2021) and omit the ψ subscript in this case.

Split-$\widehat{R}_\psi^{(2)}$. Gelman et al. (2013) proposed to apply $\widehat{R}$ on a single chain split into two sub-chains, thus providing a measure for convergence for $K = 2$ sub-chains. Intuitively, split-$\widehat{R}^{(2)}$ assesses the stationarity of a chain by comparing its mixing over the first and second half of the considered window. This can again be generalized to M simulated chains by splitting each of the M chains and setting $K = 2M$. The concept is trivially applicable for the functional version to obtain Split-$\widehat{R}_\psi^{(2)}$.

Split-$\widehat{R}_\psi^{(\kappa)}$. We modify Split-$\widehat{R}_\psi^{(2)}$ for a general number κ of sub-chain splits and denote this convergence measure by $\widehat{R}_\psi^{(\kappa)}$. Effectively, we construct $K = \kappa \cdot M$ chains from M Markov chains. Suitable values for κ ensure the sub-chains retain a certain length so mean and variance estimates are meaningful. Our experiments suggest that $\kappa = 4$ often produces good results.

Rank normalized split-$\widehat{R}_{\psi}^{(\kappa)}$. Lastly, split-$\widehat{R}^{(\kappa)}$) can be improved by rank-normalizing the posterior samples to achieve standardized sample distributions (see Vehtari et al., 2021, for a detailed description). The concept is again trivially applicable for the functional version by rank-normalizing the functional values $\psi(\boldsymbol{\theta}^{(k,s)})$.

While $\widehat{R}$ values ≤ 1.1 were originally treated as a sufficient indication of convergence, Vehtari et al. (2021) suggest that a value of 1.01 is more appropriate. This translates to a maximum between-chain variance of approximately 1% of within-chain variance times the number of samples. The diagnostics thus heavily penalize if chains converge to different modes—i.e., high between-chain variance—and implicitly assume identifiability of the parameter of interest.

Rank normalized split-$\widehat{cR}_{\psi}^{(\kappa)}$. Since penalizing within-chain variance is not meaningful for multimodal BNN posteriors, we propose to measure the parameter space convergence both chain- and layerwise. Thus we define the chain-wise measure $\widehat{cR}_{\psi}^{(\kappa)}$ which refers to the application of $\widehat{R}_{\psi}^{(\kappa)}$ to a single chain ($M = 1$).

A comparison of classical $\widehat{R}^{(2)}$ and $\widehat{cR}^{(4)}$ in the parameter space is given in Fig. 8.

Effective Sample Size (ESS) is another classical MCMC convergence measure.

Definition B.2 (Effective Sample Size (ESS)). Following the Stan Development Team (2024), we define the effective sample size (ESS) as

$$\text{ESS} = \frac{S}{1 + 2\sum_{t=1}^{\infty} \rho_t} \tag{6}$$

with ρ_t being the autocorrelation between samples at time lag t.

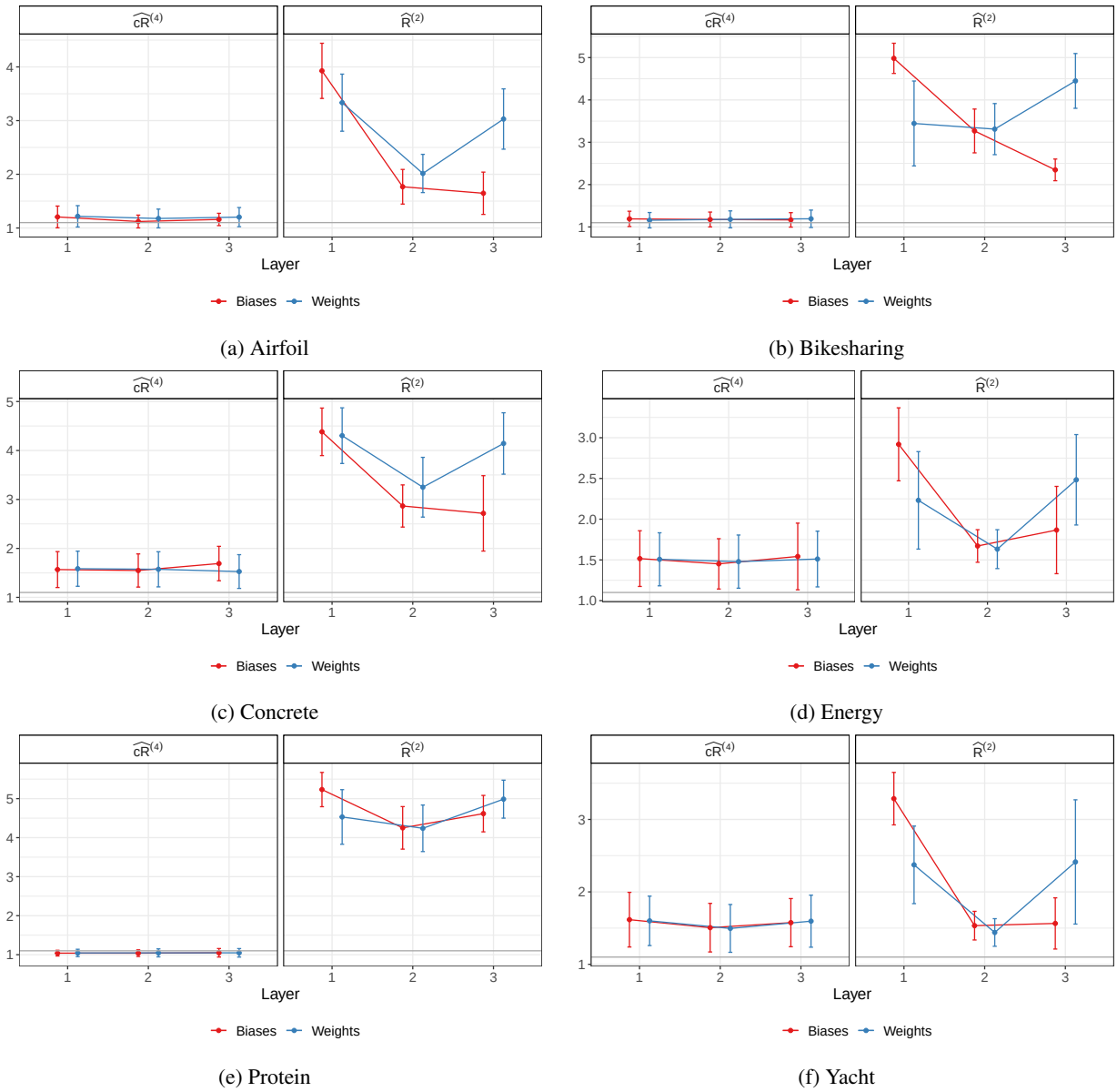
The ESS can be interpreted as the number of independent fictitious samples from the posterior itself that suffice to provide estimates as efficient as estimates from the S MCMC samples. Low ESS means that S collected samples from the chain emulate only considerably fewer samples than S from the true posterior density. Accordingly, we aim for large ESS values, which indicate low autocorrelation and thus a better exploration of the parameter space. Empirically, however, we observe that the ESS values of the parameter chains are rather small, in line with results reported in Papamarkou et al. (2022). As there seems to be no systematic pattern across chains or layers like the ones observed for the $\widehat{R}$ case, we assume that the ESS is simply not a good measure of convergence in a highly overparametrized model where large autocorrelation of samples is to be expected. Visualizations of the ESS across layers for our benchmark data sets for a well-performing architecture of two hidden layers with 16 neurons each are displayed in Figure 9.

B.2. Function Space

Convergence can also be assessed in function space since we are ultimately interested in a good functional mapping. For the calculation of convergence metrics like the $\widehat{R}_{\psi}^{(\kappa)}$ in function space, a hold-out test data set $\mathcal{D}_{\text{test}} \in (\mathcal{X} \times \mathcal{Y})^{n_{\text{test}}}$ is required. We first introduce two popular post-hoc approaches to measure function space convergence (Izmailov et al., 2021; Fortuin et al., 2022) and then propose an alternative measure that can be used for online computation of diagnostics during the sampling process.

Each sampled parameter vector $\boldsymbol{\theta}^{(k,s)}$ and test data point $\boldsymbol{x}^*$ result in one corresponding conditional density function $p\left(\boldsymbol{y}|\boldsymbol{\theta}^{(k,s)}(\boldsymbol{x}^*)\right)$ from which we sample one observation $\boldsymbol{y}^{(k,s)}$, hence every model induces $K \cdot S$ samples of outcome values for each test data point. Therefore, it is straightforward to compute the $\widehat{R}_{\psi}^{(2)}$ in the output space diagnostic for each of the test points using the mapping $\psi_{\text{PSC}}(\boldsymbol{\theta}^{(k,s)}(\boldsymbol{x}^*)) \mapsto \boldsymbol{y}^{(k,s)}$ (where PSC stands for *pointwise sample convergence*). Izmailov et al. (2021) aggregate these $\widehat{R}_{\psi_{\text{PSC}}}^{(2)}$ values and report diagnostics using histograms or summary statistics. Such metrics can be interpreted as a measure of how stationary the samples are for a fixed evaluation point. In other words, confident disagreement between chains (large difference in the expectation of the PPD and low variance within the chain) is penalized, which, as we have argued before, is unjustified for high multimodality. We report these function space $\widehat{R}_{\psi_{\text{PSC}}}^{(2)}$ values histograms in Fig. 10 across various data sets for well-performing models of the same architecture.

Another popular way of achieving a single diagnostic for function space convergence uses performance-related functions of the parameters $\boldsymbol{\theta}^{(k,s)}(\boldsymbol{x}^*)$ induced by the test data set (Fortuin et al., 2022). We will call this approach *functional convergence (FC)*. The performance function can, for instance, be the RMSE for a single posterior sample or the log



(a) Airfoil (b) Bikesharing (c) Concrete (d) Energy (e) Protein (f) Yacht

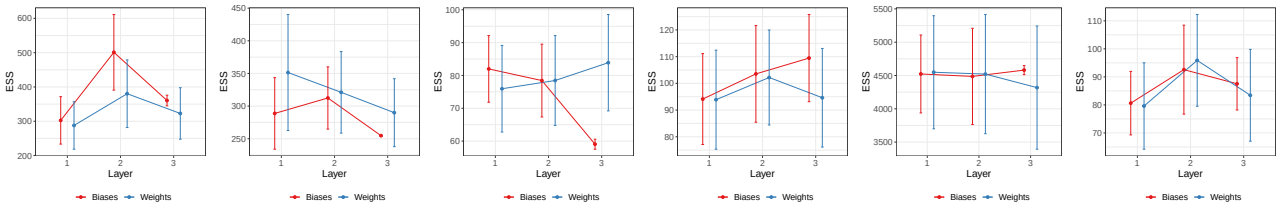
Figure 8. Average parameter-space, log-transformed $\widehat{R}^{(2)}$ and $\widehat{cR}^{(4)}$ with standard error bars across layers. For each data set, all fitted chains with better-than-LM RMSE performance are used. In all cases, the BNN consists of two hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10,000 steps and 8,000 samples per chain as well as unit Gaussian priors are used.

pointwise likelihood

$$\text{LPL}^{(k,s)} = \log\left(\frac{1}{n_{\text{test}}}\sum_{(\boldsymbol{x}^*,\boldsymbol{y}^*)\in\mathcal{D}_{\text{test}}} p\left(\boldsymbol{y}^*|\boldsymbol{\theta}^{(k,s)}(\boldsymbol{x}^*)\right)\right).$$

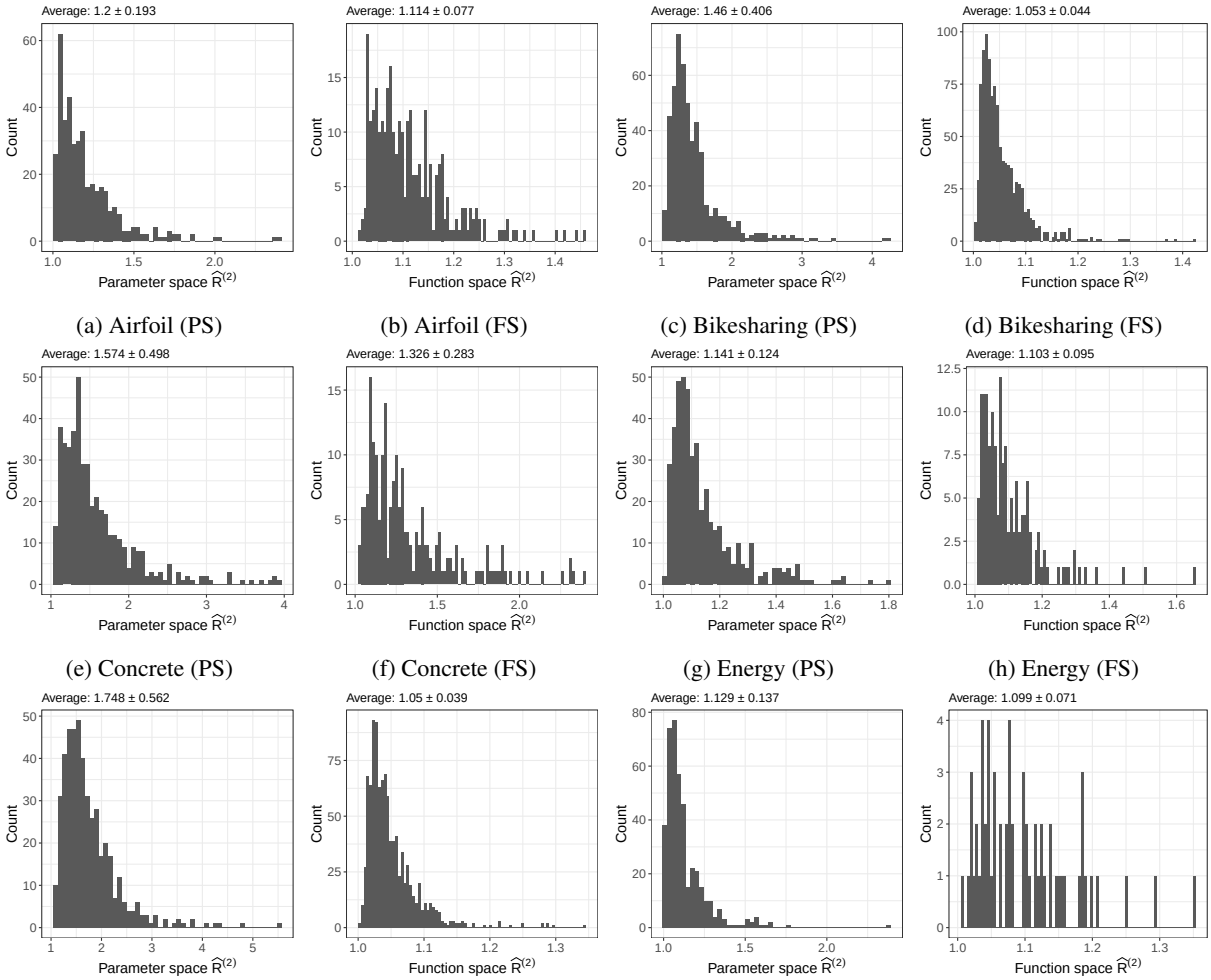
Now, we can calculate the $\widehat{R}^{(2)}_{\psi}$ with ψ being the $\text{LPL}^{(k,s)}$, accounting for effects induced by between- and within-chain variance of the $\boldsymbol{\theta}^{(k,s)}$. This notion of convergence can be interpreted as whether the chains have reached a stable state in terms of the performance metric expressed by the aggregation function.

We show in our experiments ($\text{LPL}^{(k,s)}$ and $\text{RMSE}^{(k,s)}$ chains are provided in Figures 11 and 12, resp.) that FC heavily depends on the choice of the function. RMSE-based FC, for instance, does not account for the uncertainty of predictions



(a) Airfoil (b) Bikesharing (c) Concrete (d) Energy (e) Protein (f) Yacht

Figure 9. The average parameter space ESS with standard error bars across layers and split by weights and bias role. For each data set, all fitted chains with better than LM RMSE performance are used. In all cases, the BNN consists of 2 hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10k steps and 8k samples (4k for the two larger datasets) per chain as well as unit Gaussian priors are used.



(a) Airfoil (PS) (b) Airfoil (FS) (c) Bikesharing (PS) (d) Bikesharing (FS)

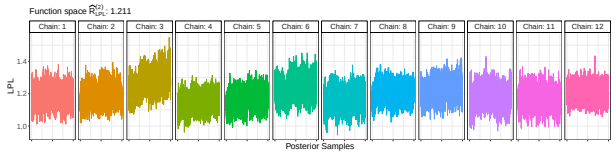
(e) Concrete (PS) (f) Concrete (FS) (g) Energy (PS) (h) Energy (FS)

(i) Protein (PS) (j) Protein (FS) (k) Yacht (PS) (l) Yacht (FS)

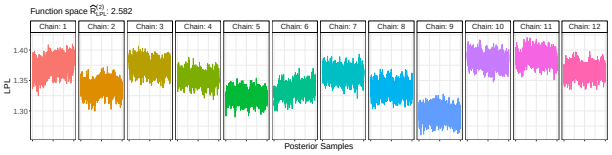
Figure 10. Histograms, mean and standard deviation of the parameter space (PS) $\widehat{R}^{(2)}$ and function space (FS) $\widehat{R}^{(2)}_{\psi_{\text{PSC}}}$, evaluated for each parameter and test data entry respectively. For each data set, all fitted chains with better than LM RMSE performance are used. In all cases, the BNN consists of two hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10k steps and 8k samples per chain as well as unit Gaussian priors are used.

explicitly. In our experiments, it thus shows better $\widehat{R}^{(2)}_{\text{RMSE}}$ values (rather close to 1.1) compared to often considerably worse $\widehat{R}^{(2)}_{\text{LPL}}$ values for most data sets. This is because the chains in function space indeed show different levels of quality in both

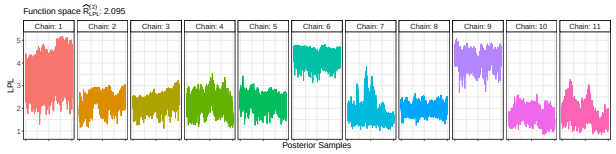
prediction and uncertainty quantification. Thus, even with rough filtering of very bad chains before the FC calculation, we obtain low within-chain variance and high between-chain variance in many cases, resulting in a $\widehat{R}_{\psi}^{(2)}$ larger than even 1.1. This clearly shows that even for overall good models there are differences between the chains concerning their quality of prediction and uncertainty quantification and, again, proves our point that current metrics are not meaningful in measuring convergence.



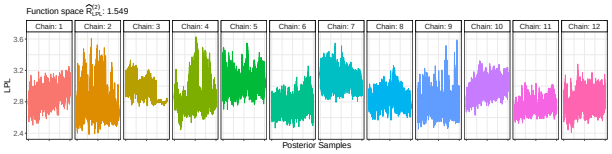
(a) Airfoil



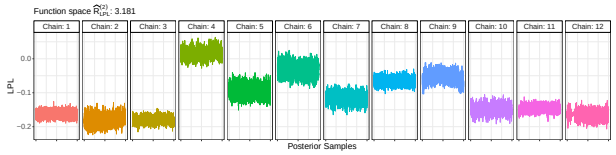
(b) Bikesharing



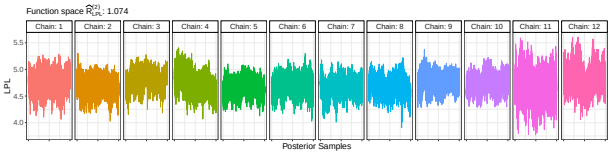
(c) Concrete



(d) Energy

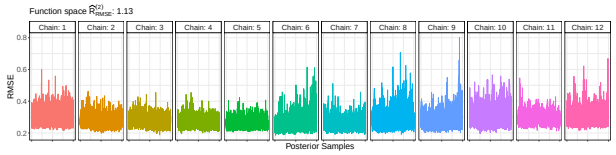


(e) Protein

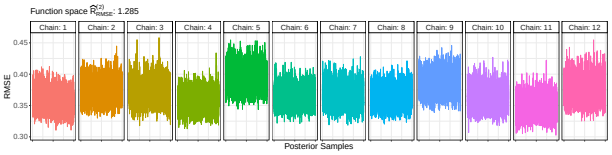


(f) Yacht

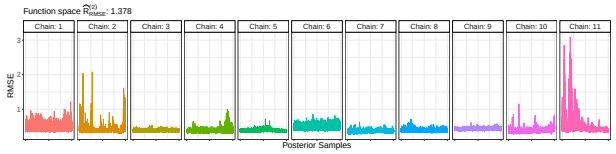
Figure 11. The LPL chains that form the basis of the $\widehat{R}_{\text{LPL}}^{(2)}$ calculation for the functional convergence assessment. For each data set, all fitted chains with better than LM RMSE performance are used (colors and boxes). In all cases, the BNN consists of 2 hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10k steps and 8k samples per chain (x-axis of the boxes) as well as unit Gaussian priors are used.



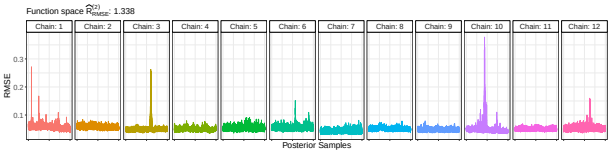
(a) Airfoil



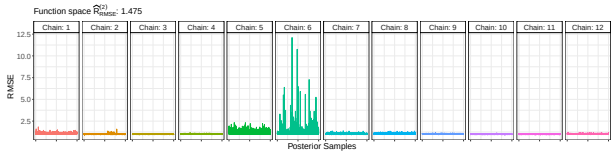
(b) Bikesharing



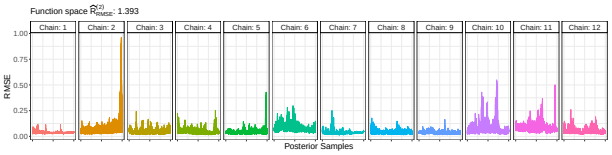
(c) Concrete



(d) Energy



(e) Protein



(f) Yacht

Figure 12. The RMSE chains that form the basis of the $\widehat{R}_{\text{RMSE}}^{(2)}$ calculation for the functional convergence assessment. For each data set, all fitted chains with better than LM RMSE performance are used (colors and boxes). In all cases, the BNN consists of 2 hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10k steps and 8k samples per chain (x-axis of the boxes) as well as unit Gaussian priors are used.

Expanding-window function space convergence. Convergence diagnostics usually evaluate chain mixing post-hoc. We propose an expanding-window approach, enabled by chain-wise analysis, that allows us to assess chain-wise convergence and between-chain agreement in function space at any time point during the sampling process. In particular, this entails the option of online resource management by early stopping of chains: when the sampler shows evidence of being stuck and risks wasting compute, the chain can be restarted.

For this, following Gelman et al. (2014) and Wiese et al. (2023), we choose the log posterior predictive density (LPPD) over a test set, defined as

$$\text{LPPD} = \frac{1}{n_{\text{test}}} \sum_{(\boldsymbol{y}^*, \boldsymbol{x}^*) \in \mathcal{D}_{\text{test}}} \log \left(\frac{1}{K \cdot S} \sum_{k=1}^{K} \sum_{s=1}^{S} p\left(\boldsymbol{y}^* | \boldsymbol{\theta}^{(k,s)}(\boldsymbol{x}^*)\right) \right). \tag{7}$$

The LPPD quantifies how well, on average, the predictive distribution covers the observed label. As we can evaluate the LPPD chain-wise (using $K = 1$ in (7)) after every sample update, this motivates the chain-wise expanding-window LPPD for any time point $l \in \{1, \ldots, S\}$:

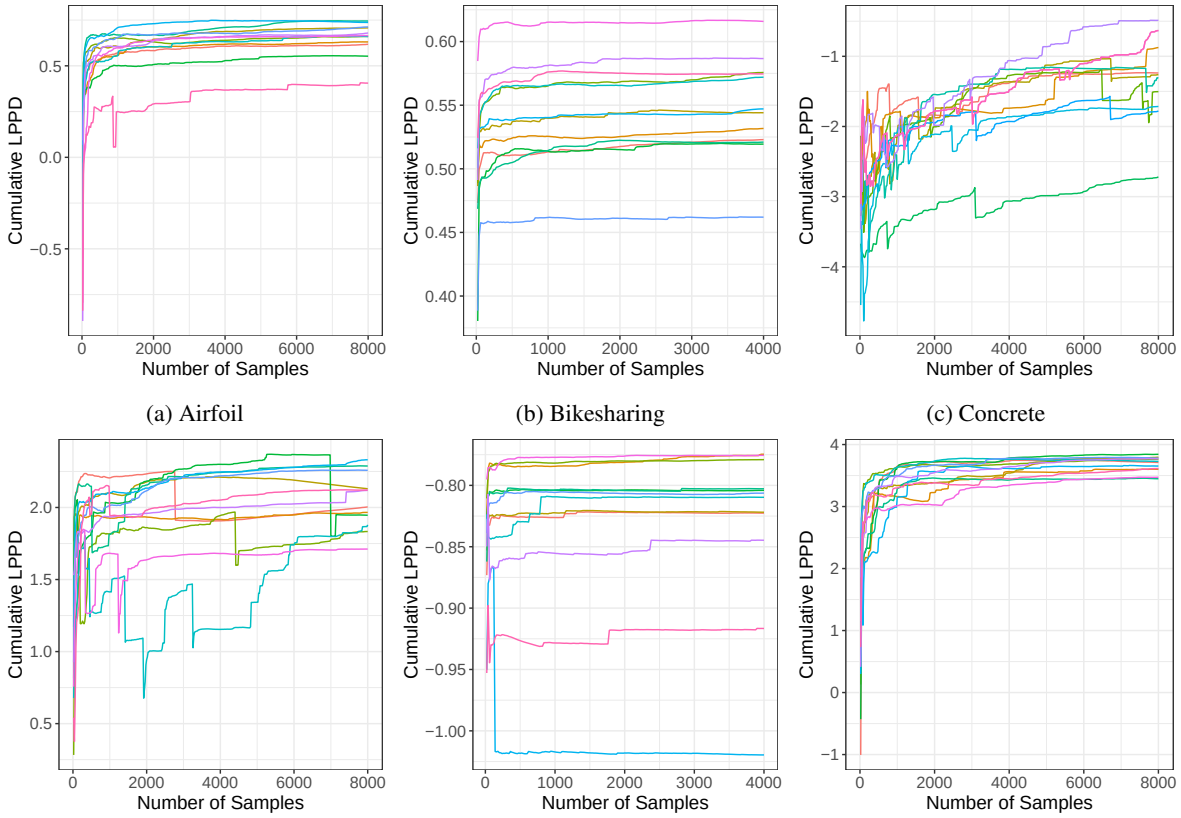
$$\text{LPPD}_l = \frac{1}{n_{\text{test}}} \sum_{(\boldsymbol{y}^*, \boldsymbol{x}^*) \in \mathcal{D}_{\text{test}}} \log \left(\frac{1}{l} \sum_{\tilde{l} \leq l} p\left(\boldsymbol{y}^* | \boldsymbol{\theta}^{(\tilde{l})}(\boldsymbol{x}^*)\right) \right). \tag{8}$$

In order to assess convergence, we can define an ϵ-threshold and window $\varpi \in \mathbb{N}$ such that

$$\left| \left(\frac{1}{\varpi} \sum_{l-\varpi \leq \tilde{l} < l} \text{LPPD}_{\tilde{l}} \right) - \text{LPPD}_l \right| < \epsilon. \tag{9}$$

As soon as convergence or a maximum number of samples is reached, the sampling of the chain can be terminated to start a new chain. In case of non-convergence, the chain is discarded. This is especially valuable if one aims for more chains rather than more samples, as suggested in Section 6.1, given a certain computation budget. Another advantage of this metric is the possibility to compare the LPPD across chains to, e.g., detect outliers. As such, the notion of between-chain variance is not lost. In contrast to FC, our method does not only evaluate a pointwise performance but reflects the current overall performance by using the running mean.

Empirically we observe convergence for most chains, as displayed in Fig. 13. Moreover, the convergence to different levels of LPPD is notably visible, especially for the larger data sets `bikesharing` and `protein`. The chain-wise expanding-window LPPD usually shows convergence already after a few samples per chain. This implies great benefits in sampling efficiency, allowing to free computational resources early on in the sampling process and to run a large number of chains (almost) in parallel.



(a) Airfoil (b) Bikesharing (c) Concrete

(d) Energy (e) Protein (f) Yacht

Figure 13. Cumulative LPPD across data sets for the proposed expanding-window convergence assessment. For each data set, all fitted chains with better-than-LM RMSE performance are used (colors and boxes). In all cases, the BNN consists of two hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10,000 steps and 8,000 samples per chain (x-axis) as well as unit Gaussian priors are used.

C. Additional Experiments

C.1. General Feasibility

Table 4 contains the results reported in Table 1 with additional standard deviations in parentheses. The architecture is a two-hidden layer MLP with 16 neurons in each hidden layer. The BNNs are sampled using 10,000 warmup steps of NUTS and 8,000 samples for each of the 12 chains. Only chains that proved to be better than the baseline LM are considered for the calculation (here always at least 10). The tanh activation function and unit Gaussian priors are applied. Experiments reported in Table 2 also use unit Gaussian priors. Note that for the Random Forest baseline an extensive HPO search using optuna (Akiba et al., 2019) is performed. Thus, the comparison is rather conservative as we do not perform a neural architecture search for the network-based models. The relevant yardstick in this comparison are the Deep Ensembles which share the same inductive biases as the BNNs.

Table 4. Average RMSE for different models over different data sets. All networks have two hidden layers with 16 neurons each.. The best method per data set is highlighted in bold.

DATA SET	LM	RF (TUNED)	DNN	DE	BNN (RS)	BNN
AIRFOIL	0.716 (0.022)	0.255 (0.005)	0.252 (0.024)	0.239 (0.023)	0.25 (0.009)	**0.182** (0.009)
BIKESHARING	0.790 (0.028)	**0.231** (0.007)	0.374 (0.002)	0.365 (0.003)	0.362 (0.013)	0.253 (0.036)
CONCRETE	0.630 (0.018)	0.304 (0.021)	0.317 (0.015)	0.282 (0.018)	0.554 (0.158)	**0.258** (0.020)
ENERGY	0.274 (0.020)	0.051 (0.009)	0.048 (0.006)	0.043 (0.007)	0.062 (0.006)	**0.037** (0.006)
PROTEIN	0.863 (0.028)	**0.581** (0.004)	0.804 (0.004)	0.803 (0.004)	1.077 (0.085)	0.716 (0.035)
YACHT	0.612 (0.068)	0.072 (0.015)	0.108 (0.032)	0.103 (0.031)	0.032 (0.007)	**0.022** (0.007)

Fig. 14 shows the share of chains that are able to outperform the lower baseline of an LM across different activation functions, clearly pointing towards a problem of unbounded activation functions. Notably, only the truncated ReLU is able to avoid the dying sampler problem, whereas smoothed ReLU versions, such as SiLU and leaky ReLU, offer very little mitigation.

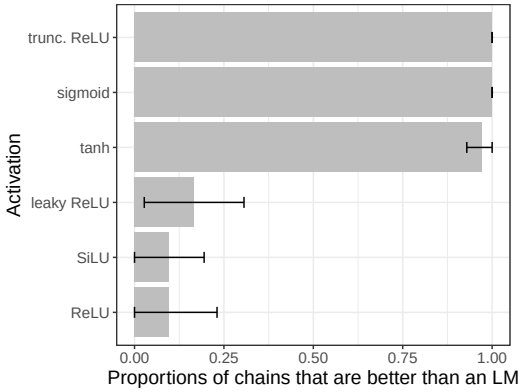
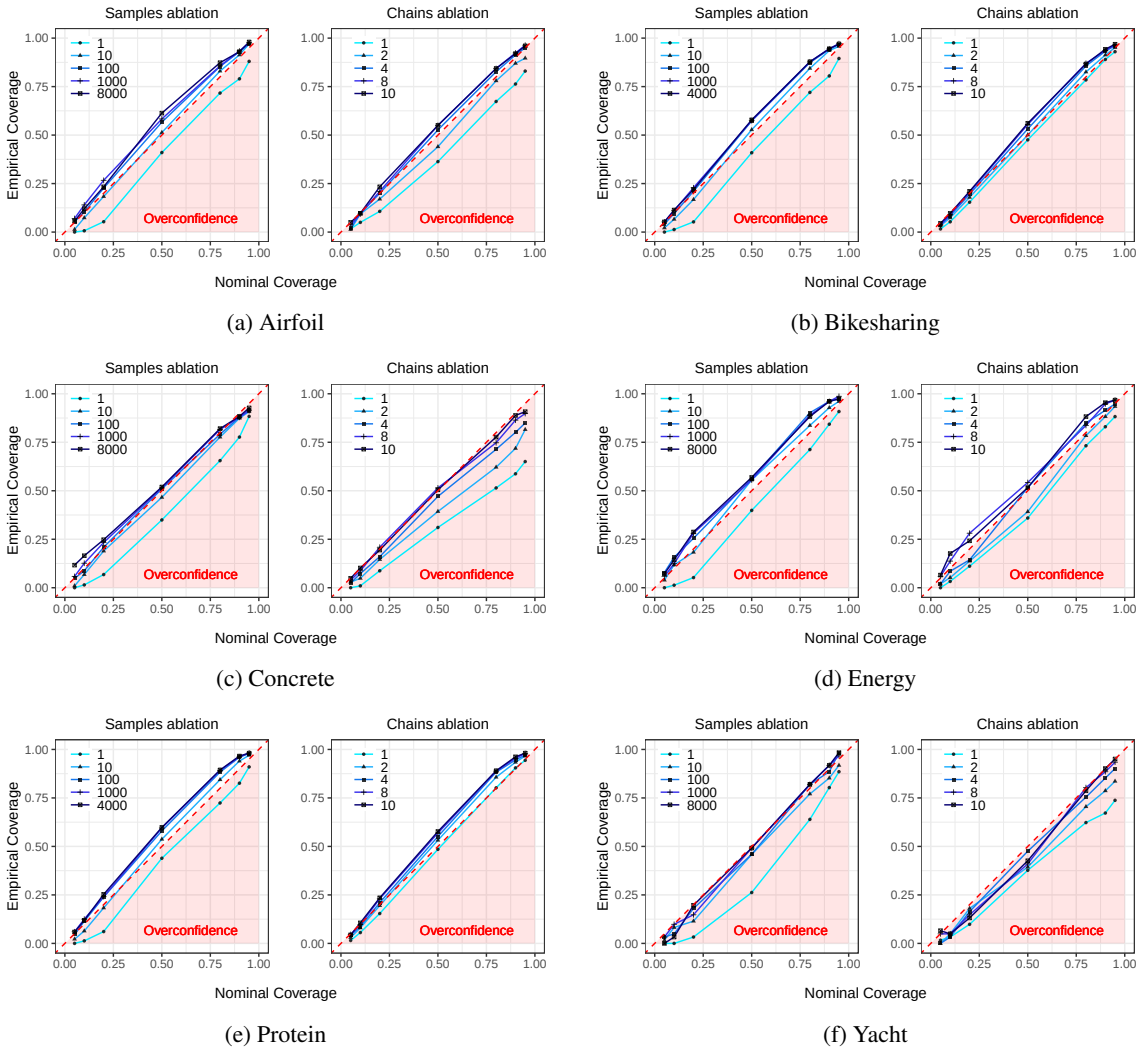


Figure 14. Proportions of BNN chains with better performance than an LM separated for different activation functions. Each proportion is the average of 72 experiments with 3 different train-test splits of the two data sets `airfoil` and `energy`, 1,000 warmup iterations of NUTS and 12 chains each with 1,000 posterior samples.

In Table 5, reflecting the setting of Table 5 in Izmailov et al. (2021), we contrast the proportion of LM-beating chains produced by HMC and NUTs for ReLU, SiLU and tanh activations. We consider a single-hidden layer network with 50 neurons and a Gaussian prior with standard deviation 0.1 HMC is used with fixed step size of 10^{-5}, trajectory length $\frac{\pi * 0.1}{2}$ and 10 warmup samples as suggested in Izmailov et al. (2021). The NUTS sampler runs with default settings and a warmup length of 5k steps. For both samplers 400 samples are drawn from 12 independent chains over 3 random train-test splits. Again, tanh activations produce fewer below-baseline chains when NUTS is used, whereas HMC with fixed setting essentially fails in all settings.

Predictive coverage. Bayesian inference enables the computation of predictive credibility intervals, which should be

calibrated and cover the true prediction. Fig. 15 shows the coverage of credibility intervals for varying amounts of samples (left) and chains (right). For a calibrated model (represented by the diagonal), the observed labels fall within the $\alpha\%$ credibility interval in $\alpha\%$ of cases. With increasing chains and samples, we can see a trend from overconfidence to close-to-nominal coverage for all data sets. While Fig. 15 covers the nominal coverage levels $(0.05, 0.1, 0.2, 0.5, 0.8, 0.9, 0.95)$ for a broad overview, the trend observed is even more accentuated for very small and large coverage levels.



(a) Airfoil (b) Bikesharing

(c) Concrete (d) Energy

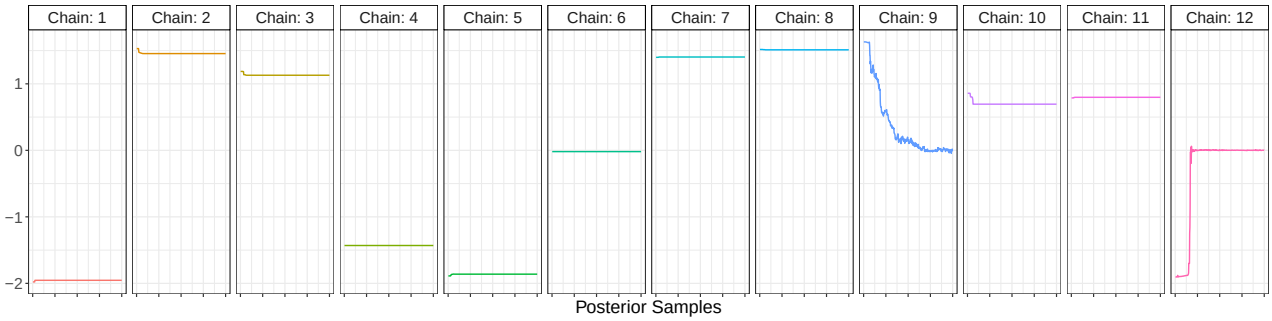
(e) Protein (f) Yacht

Figure 15. Nominal vs. empirical coverage of posterior credibility intervals for different numbers of samples using 10 chains everywhere (left) and different numbers of chains using 100 samples each (right) across various data set, NUTS with 10,000 warmup, two hidden layers of 16 neurons each, tanh activation, and unit Gaussian priors.

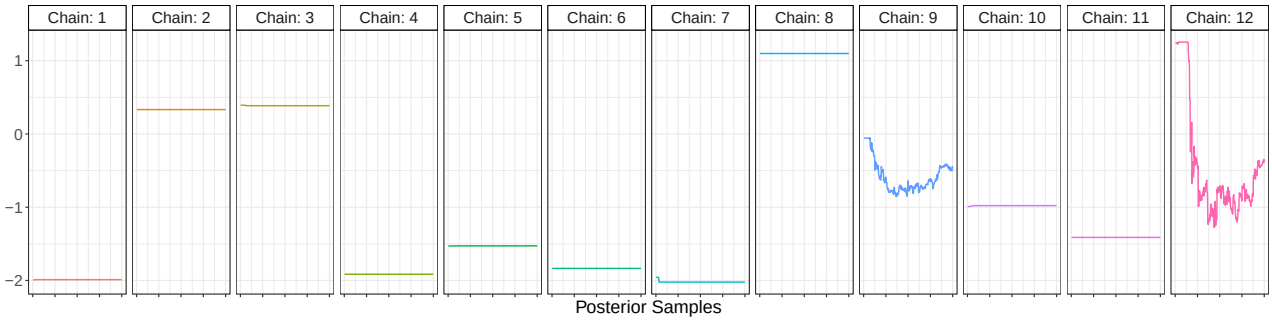
Dying NUTS sampler details. In the cases where the NUTS sampler ends up in chains inducing utterly bad models–with, e.g., RMSE values far worse than a simple LM—a robust pattern emerges. The sampler first tries larger step sizes but then, as the acceptance probability stays at practically zero, the step size and number of steps are quickly reduced. As a result, the sampler remains close to the starting values. With the step size now contracted to a very small value, however, the acceptance probability rises again, in effect causing the sampler to move around the starting value in tiny steps. Since this unfortunate state is attained fast during the warmup, the sampler only repeats samples from a fraction of the parameter space close to the initial values. Obviously, those do not result in any meaningful uncertainty quantification and almost always fail to provide even close-to-good predictive capabilities. A visual example of such a warmup phase is provided in Fig. 16. Evidently most chains have contracted/died already after fewer than 100 warmup steps.

Table 5. The proportion of chains resulting in better RMSE performance than the LM. HMC is run with 10 warmup samples, NUTS with 5k warmup samples. Values are averaged across 12 chains and 3 replications.

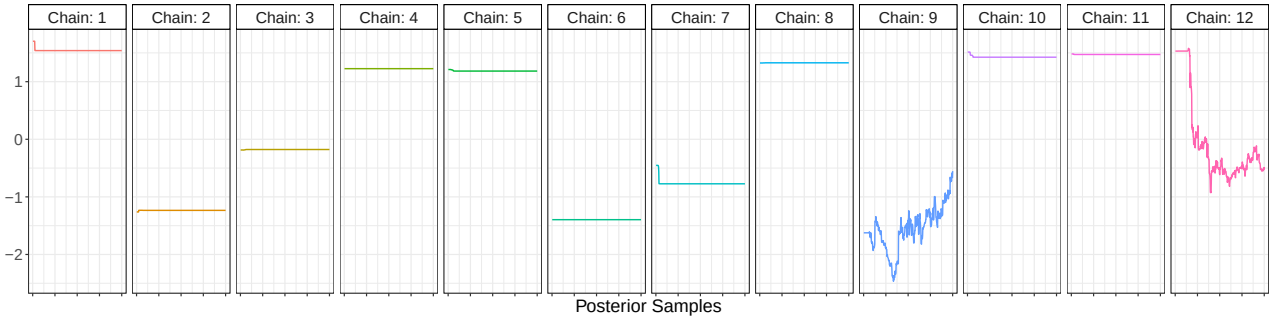
DATA SET	ACTIVATION	SAMPLER	PROPORTION	SD
HMC	RELU	CONCRETE	0.03	0.05
		ENERGY	0.03	0.05
	SILU	CONCRETE	0.03	0.05
		ENERGY	0.00	0.00
	TANH	CONCRETE	0.00	0.00
		ENERGY	0.00	0.00
NUTS	RELU	CONCRETE	0.14	0.13
		ENERGY	0.31	0.13
	SILU	CONCRETE	0.14	0.10
		ENERGY	0.28	0.05
	TANH	CONCRETE	0.58	0.25
		ENERGY	0.67	0.08



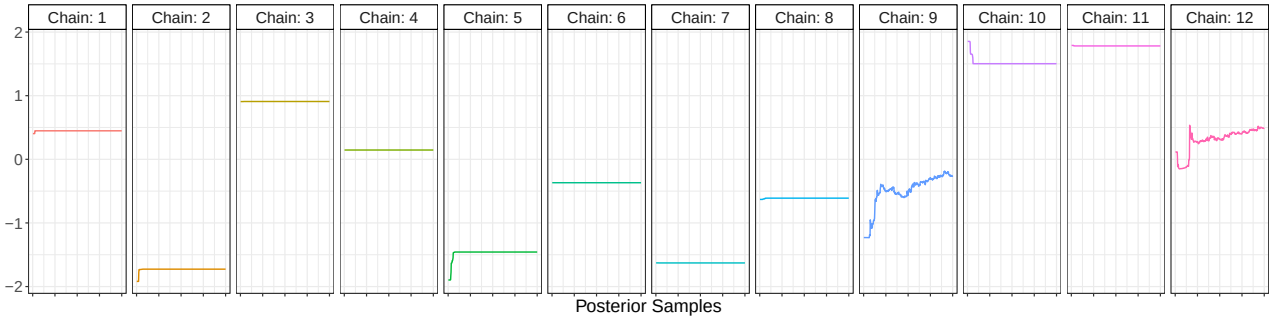
(a) Random first layer weight



(b) Random second layer weight



(c) Random thrid layer weight



(d) Random fourth (last) layer weight

Figure 16. Traceplots of a randomly selected parameter during the sampling phase of a dying NUTS sampler. The BNN has three hidden layer of ten neurons each, uses unit Gaussian priors, ReLU activation and is sampled for the `energy` data set with 1,000 NUTS warmup samples and 12 chains. The first 400 steps of the sampling are displayed (x-axis).

C.2. Mulimodality of Posteriors

In the following the experiments for the analyses reported in Figures 2, 3 and 4 are described in detail. The architecture is always a network with six hidden layers of eight neurons each with tanh activation. Unit Gaussian priors are applied and the used NUTS sampler is configured with a 10,000 step warmup phase and 8,000 samples for each of the 12 chains. Again, the few chains (always ≤ 2) that cannot beat a LM with respect to the RMSE are removed from the analysis.

In addition to the analyses reported in Figures 2, 3 and 4, we investigate the sampling path for each chain. Here, instead of aggregating insights from marginal parameter traces, we take a joint view on the sampling path. In order reduce the dimensionality of this problem, we perform a principal component analysis (PCA; Wold et al., 1987) given S posterior samples from the k-th chain $\boldsymbol{\Theta} = (\boldsymbol{\theta}^{(k,1)}, \boldsymbol{\theta}^{(k,2)}, \ldots, \boldsymbol{\theta}^{(k,S)})^\top$. Taking the three first principal components (each of dimension $|\boldsymbol{\theta}^{(k,s)}|$), we can project the sampling path into three dimensions. The first three principal components explain a substantial amount of variance of the sampling path ($> 50\%$), which suggests that insights from the PCA-approximated path are indicative of the original, high-dimensional situation. This is confirmed visually, e.g., in Fig. 17, which is a representative example for the first chain of the energy dataset. Summing up the element-wise absolute values of the orthonormal vectors yields a proxy of how important the particular parameter is for the movement of the sampler. Applying this methodology and grouping the PCA factor loadings layer-wise, as well as averaging across experiments and chains, we obtain Fig. 18. The same pattern as in Fig. 4 emerges, i.e., deeper layers are more important for the movement of the chain in the parameter space. We conclude from this that our findings pertain to a robust pattern.

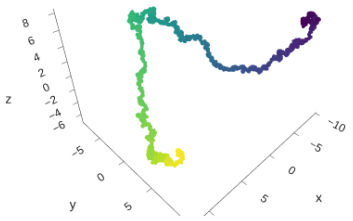


Figure 17. Example sampling path of the first chain of the energy dataset projected in three dimensions using PCA. Sampler moved from dark to bright.

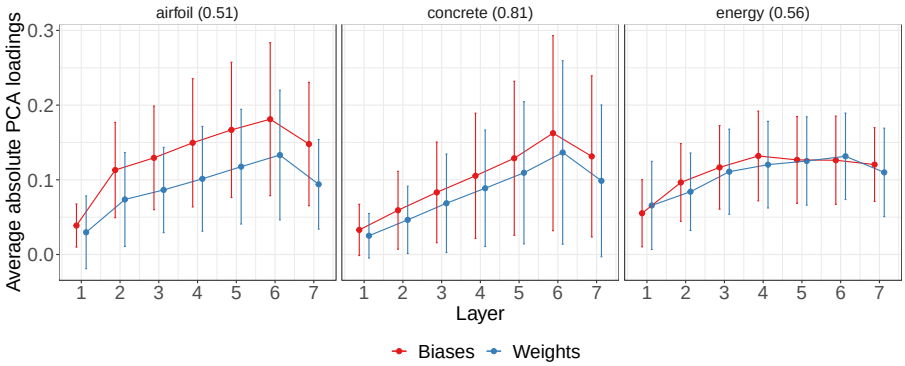
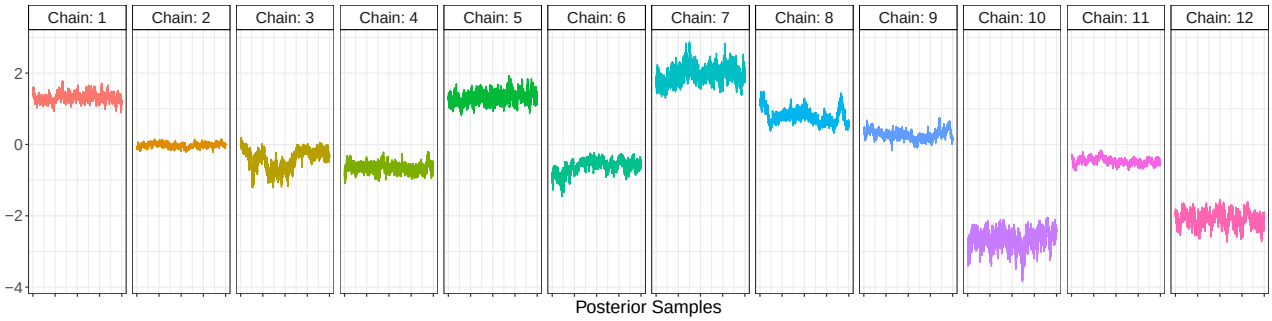


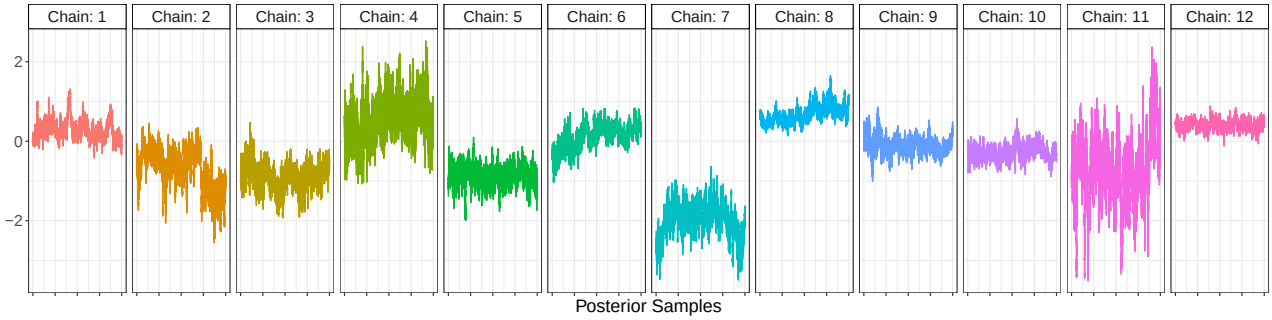
Figure 18. Mean and standard deviation of the average absolute PCA loadings separated by layer (x-axis) for different data sets (columns) of a seven-layer BNN. Average explained variance by the first three principal components in brackets.

The traceplots displayed in Figures 19 and 20 are both examples of the fact that symmetry-induced multimodality exists and also illustrate how the within-chain variance increases in deeper layers. Moreover, Figure 20 shows how for the same

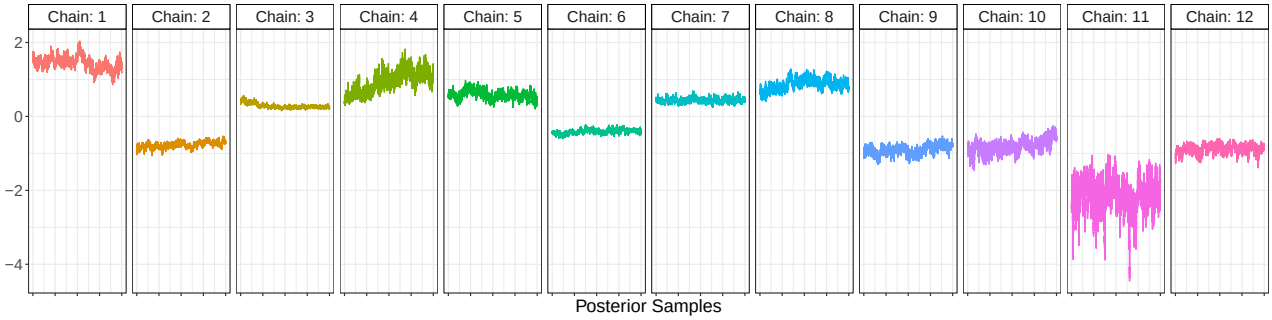
setting some (in this case two) chains are "dying", i.e., have a near-constant trace plot, while others explore the space in a meaningful way (implied by the good performance of the induced BNN).



(a) Random first layer weight

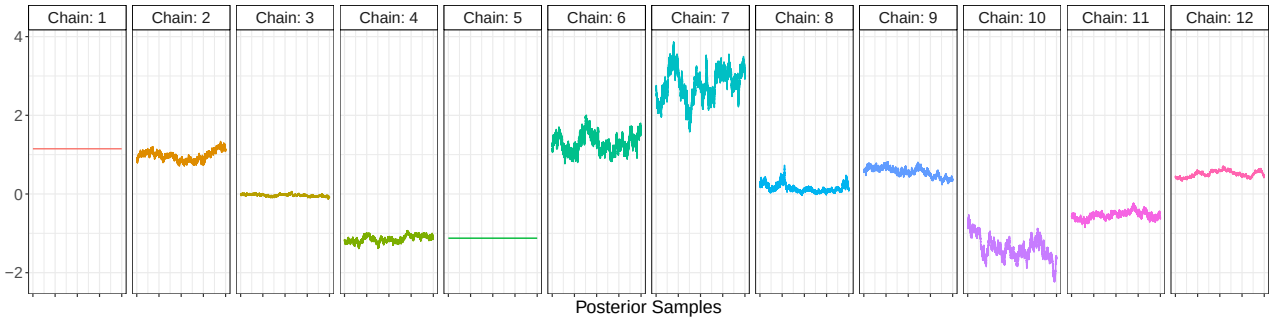


(b) Random second layer weight

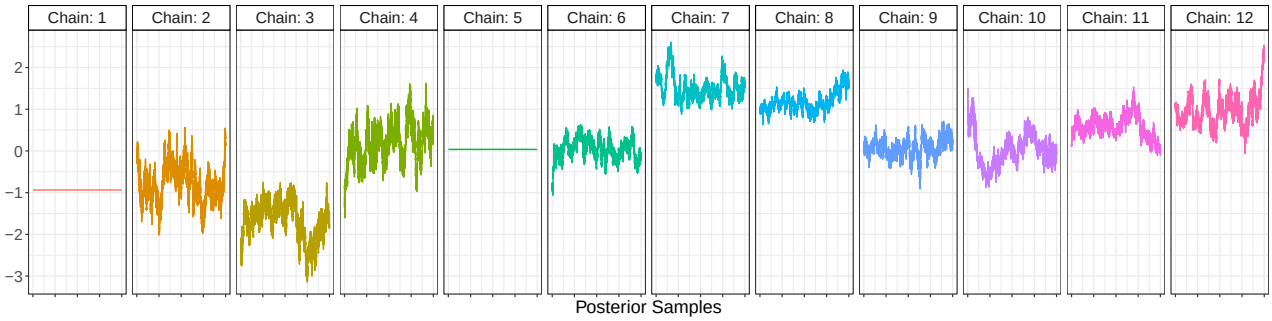


(c) Random thrid (last) layer weight

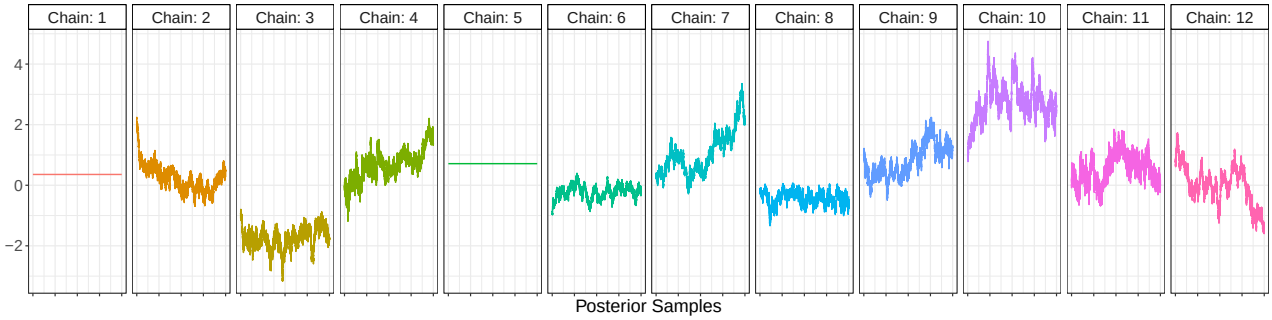
Figure 19. Traceplots of a randomly selected parameter during the sampling phase of a NUTS sampler that performs well for most chains. The BNN has two hidden layer of 16 neurons each, uses unit Gaussian priors, tanh activation and is sampled for the `airfoil` data set with 10,000 NUTS warmup samples and 12 chains. The entire sampling phase, collecting 8,000 samples, is displayed (x-axis).



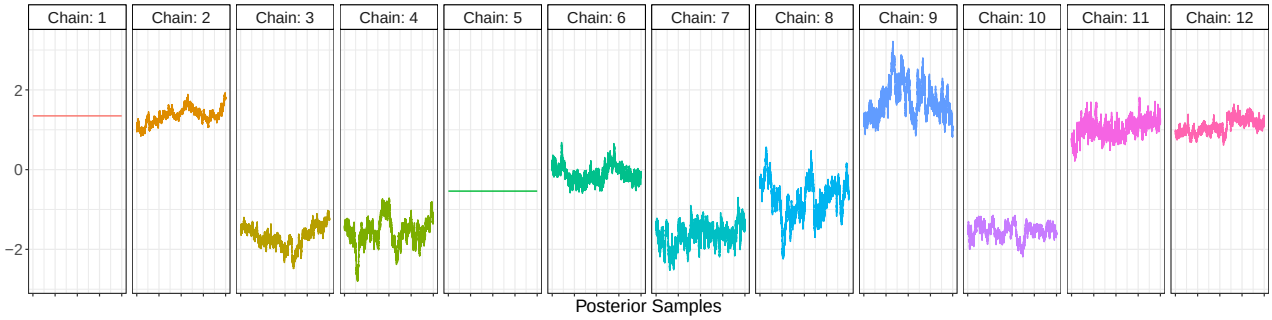
(a) Random first layer weight



(b) Random third layer weight



(c) Random fifth layer weight



(d) Random seventh (last) layer weight

Figure 20. Traceplots of a randomly selected parameter during the sampling phase of a NUTS sampler that performs well for most chains. The BNN has six hidden layer of eight neurons each, uses unit Gaussian priors, tanh activation and is sampled for the `airfoil` data set with 10,000 NUTS warmup samples and 12 chains. The entire sampling phase, collecting 8,000 samples, is displayed (x-axis).

C.3. Practical SBI

Prior choice. Here, we report detailed results for the influence of different prior choices. Tables 6 and 7 show the test RMSE and the share of chains outperforming the LM, respectively, for the `airfoil` and `concrete` data sets. We employ NUTS with 10,000 warmup steps and 12 chains with 8,000 samples each. Each experiment is replicated on three random train-test splits of the data. All other tested configurations are displayed in the tables themselves.

Table 6. RMSE hold-out performance across various prior distributions and strengths (standard deviation in parentheses)

		2		8		64		32-32-32	
DATA SET		RELU	TANH	RELU	TANH	RELU	TANH	RELU	TANH
AIRFOIL	LAPLACE (0.1)	0.5997 (0.0321)	0.6072 (0.0287)	0.3839 (0.0203)	0.4413 (0.0534)	0.3342 (0.0102)	0.3557 (0.0109)	-	0.2221 (0.0094)
	LAPLACE (1)	0.5876 (0.0328)	0.5924 (0.0326)	0.3349 (0.0107)	0.3208 (0.0151)	0.2288 (0.0066)	0.2207 (0.0122)	-	0.1609 (0.0189)
	LAPLACE (10)	0.5931 (0.0436)	0.5916 (0.0297)	0.3346 (0.0080)	0.3289 (0.0022)	0.2467 (0.0241)	0.2209 (0.0121)	-	0.2498 (0.0302)
	NORMAL (0.1)	0.6420 (0.0428)	0.6703 (0.0188)	0.5831 (0.0254)	0.6475 (0.0295)	0.5841 (0.0287)	0.6558 (0.0276)	-	0.6340 (0.0368)
	NORMAL (1)	0.5897 (0.0342)	0.5949 (0.0303)	0.3385 (0.0234)	0.3333 (0.0029)	0.2325 (0.0102)	0.2282 (0.0069)	-	0.1432 (0.0125)
	NORMAL (10)	0.5833 (0.0377)	0.5982 (0.0312)	0.3506 (0.0131)	0.3468 (0.0157)	0.2467 (0.0130)	0.2189 (0.0122)	-	0.2330 (0.0130)
CONCRETE	LAPLACE (0.1)	0.4323 (0.0026)	0.4308 (0.0045)	0.3615 (0.0070)	0.3693 (0.0080)	-	0.3610 (0.0124)	-	0.2745 (0.0000)*
	LAPLACE (1)	0.4309 (0.0040)	0.4271 (0.0019)	0.3318 (0.0056)	0.3241 (0.0047)	-	0.2676 (0.0278)	-	0.3045 (0.0119)
	LAPLACE (10)	0.4323 (0.0030)	0.4188 (0.0216)	0.3334 (0.0063)	0.3232 (0.0025)	-	0.2803 (0.0199)	-	0.3450 (0.0000)*
	NORMAL (0.1)	0.5012 (0.0208)	0.5021 (0.0110)	0.4439 (0.0079)	0.4674 (0.0077)	-	0.4718 (0.0041)	-	0.4444 (0.0000)*
	NORMAL (1)	0.4323 (0.0046)	0.4282 (0.0060)	0.3317 (0.0057)	0.3224 (0.0085)	-	0.2844 (0.0262)	-	0.3010 (0.0169)
	NORMAL (10)	0.4321 (0.0020)	0.4239 (0.0228)	0.3406 (0.0107)	0.3253 (0.0074)	-	0.2762 (0.0249)	-	0.3767 (0.0000)*

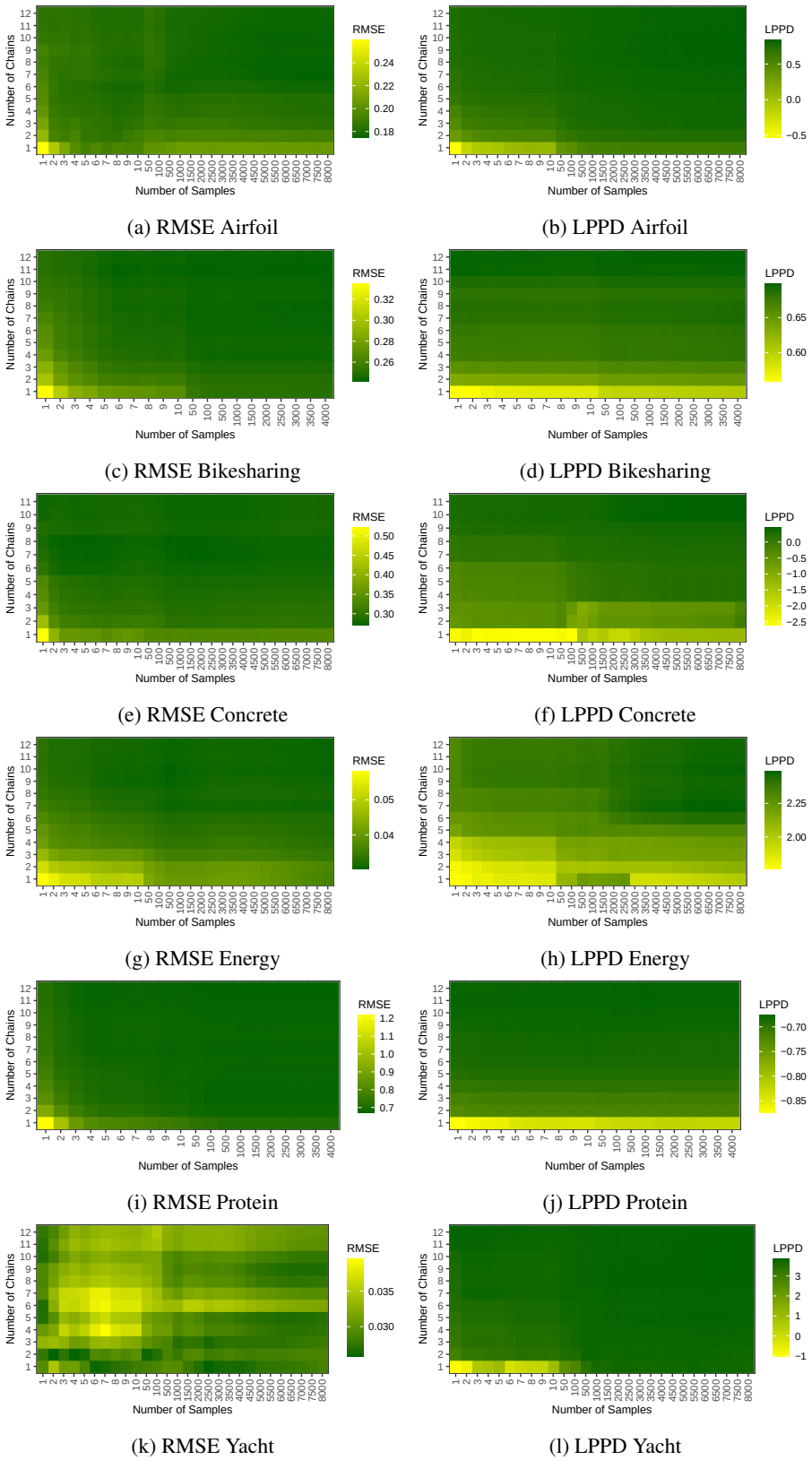
* ONLY ONE REPLICATION.

Table 7. Proportion of better-than-LM chains across various prior distributions and strengths (standard deviation in parentheses)

		2		8		64		32-32-32	
DATA SET		RELU	TANH	RELU	TANH	RELU	TANH	RELU	TANH
AIRFOIL	LAPLACE (0.1)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.36 (0.10)	0.64 (0.13)	0.00 (0.00)	0.67 (0.25)
	LAPLACE (1)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.36 (0.10)	0.64 (0.13)	0.00 (0.00)	0.67 (0.25)
	LAPLACE (10)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.33 (0.08)	0.58 (0.17)	0.00 (0.00)	0.56 (0.21)
	NORMAL (0.1)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.36 (0.10)	0.64 (0.13)	0.00 (0.00)	0.64 (0.29)
	NORMAL (1)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.36 (0.10)	0.64 (0.13)	0.00 (0.00)	0.67 (0.25)
	NORMAL (10)	0.97 (0.05)	1.00 (0.00)	0.89 (0.10)	1.00 (0.00)	0.31 (0.05)	0.64 (0.13)	0.00 (0.00)	0.58 (0.25)
CONCRETE	LAPLACE (0.1)	0.89 (0.10)	1.00 (0.00)	0.69 (0.10)	1.00 (0.00)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.33 (0.00)*
	LAPLACE (1)	0.86 (0.13)	1.00 (0.00)	0.69 (0.10)	1.00 (0.00)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.38 (0.06)
	LAPLACE (10)	0.83 (0.08)	1.00 (0.00)	0.64 (0.13)	1.00 (0.00)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.33 (0.00)*
	NORMAL (0.1)	0.89 (0.10)	1.00 (0.00)	0.69 (0.10)	1.00 (0.00)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.33 (0.00)*
	NORMAL (1)	0.89 (0.10)	1.00 (0.00)	0.69 (0.10)	1.00 (0.00)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.38 (0.06)
	NORMAL (10)	0.89 (0.10)	1.00 (0.00)	0.69 (0.10)	0.97 (0.05)	0.08 (0.08)	0.56 (0.05)	0.00 (0.00)	0.33 (0.00)*

* ONLY ONE REPLICATION.

Performance and uncertainty. In accordance to Fig. 5, Fig. 21 visualizes the RMSE (left column) and LPPD (right column) performance for all data sets. Except for the `yacht` data, we observe the same pattern of improvement along the directions of both chains and samples, where the former tends to have more effect. In all cases, the BNN consists of two hidden layers with 16 neurons each and tanh activation. NUTS with a warmup phase of 10,000 steps and 8,000 samples per chain (4,000 for the two larger data sets) as well as unit Gaussian priors are used. Only for `concrete`, one of the 12 chains is dropped as it underperformed the weak LM baseline.



(a) RMSE Airfoil (b) LPPD Airfoil

(c) RMSE Bikesharing (d) LPPD Bikesharing

(e) RMSE Concrete (f) LPPD Concrete

(g) RMSE Energy (h) LPPD Energy

(i) RMSE Protein (j) LPPD Protein

(k) RMSE Yacht (l) LPPD Yacht

Figure 21. Improvement across samples and chains of RMSE (left column) and LPPD (right column) for different data sets (rows).

C.4. Dying Sampler and Deep Ensemble Initialization

DEI-MCMC reported in Table 3 uses 12 chains each with 8,000 samples (4,000 for the larger data sets) using the NUTS sampler with a short warmup phase of 100 steps. The table reports the first 10, 100 and 1,000 samples from these chains. Each chain's initial proposal is a parameter set of the converged DE member trained on the same architecture, which is a network with two hidden layers of 16 neurons each. The priors on the weights are again unit Gaussians, leading to a valid initialization of the chains. The activation is ReLU.

Moreover, the comparison presented in Table 8 of the LPPD between DEI-MCMC and Deep Ensembles indicates a significant enhancement in uncertainty quantification also for deeper architectures.

Table 8. Average LPPD values (± their standard deviations across replications) of a DE, and DEI-MCMC (i.e., a BNN with DNN warm starts) using 1000 samples for four of the benchmark data sets. All networks have 9 hidden layers with 8 neurons each and ReLU activations. The best method per data set is highlighted in bold

DATASET	DEI-MCMC	DEEP ENSEMBLE
AIRFOIL	$\mathbf{0.5272} \pm 0.1210$	-0.1677 ± 0.0766
BIKESHARING	$\mathbf{0.6799} \pm 0.0530$	0.4145 ± 0.0422
CONCRETE	$\mathbf{0.1683} \pm 0.0528$	-0.2794 ± 0.0622
ENERGY	$\mathbf{2.1147} \pm 0.0907$	0.8019 ± 0.2090

D. Experimental Settings and Further Details

Benchmark data. Table 9 describes the data characteristics for our benchmark data. Note that we have normalized the features and outcome values for all data sets.

Table 9. Data set characteristics and references.

DATA SET	# OBS.	# FEAT.	REFERENCE
AIRFOIL	1503	5	DUA & GRAFF (2017)
BIKESHARING	17379	13	FANAEE-T (2013)
CONCRETE	1030	8	YEH (1998)
ENERGY	768	8	TSANAS & XIFARA (2012)
PROTEIN	45730	9	DUA & GRAFF (2017)
YACHT	308	6	ORTIGOSA ET AL. (2007); DUA & GRAFF (2017)

Software. Our software is implemented in Python and mainly relies on the `jax` (Bradbury et al., 2018), `numpyro` (Phan et al., 2019) and `pytorch` (Paszke et al., 2019) libraries. We further use `Docker` for a reproducible experimental setup. Our code is available at `https://github.com/EmanuelSommer/bnn_connecting_the_dots`.

Baselines. For the DNN and DE baselines, we train all models full-batch for 5,000 epochs with a Gaussian negative log-likelihood loss function. We use the `Adam` optimizer with an initial learning rate of 10^{-2} and weight decay equal to 10^{-2}. Note that the single-DNN performance is computed as the expected value of using one ensemble member, i.e., averaging the performance of individual members.

We also explored further settings in a grid search over the strength of weight decay (10^{-3}, 10^{-2}, 10^{-1}) as well as batch sizes (32, 64), allowing the training process to terminate if no improvement in validation loss was reached after 30 consecutive epochs. For the latter, we set aside 10% of the data as a validation set (amounting to a 70%/10%/20% split for training/validation/test). The results for training with tanh activation are displayed in Table 10. However, we could not detect any systematic improvement in RMSE performance (cf. Table 1) as compared to the above configurations.

Computing environment and times. The experiments were run on 4 CPU instances with 32 cores each and 64GB RAM. Sampling 12 chains for most experiments allowed to parallelize the sampling such that at all times 2 experiments with 12 chains each can be run. The sampling with NUTS of 12 chains, 10,000 warmup steps and 8,000 samples (4,000 for the larger data sets) required on average three hours for the smaller data sets, and 30 hours for the larger ones on a two-hidden-layer

DATA SET	WEIGHT DECAY	BATCH SIZE	RMSE
AIRFOIL	0.0001	32	0.2712
AIRFOIL	0.0001	64	0.3027
AIRFOIL	0.0010	32	0.2853
AIRFOIL	0.0010	64	0.2858
AIRFOIL	0.0100	32	0.2951
AIRFOIL	0.0100	64	0.3183
BIKESHARING	0.0001	32	0.2779
BIKESHARING	0.0001	64	0.2899
BIKESHARING	0.0010	32	0.2832
BIKESHARING	0.0010	64	0.2886
BIKESHARING	0.0100	32	0.3289
BIKESHARING	0.0100	64	0.3392
CONCRETE	0.0001	32	0.3572
CONCRETE	0.0001	64	0.3646
CONCRETE	0.0010	32	0.3566
CONCRETE	0.0010	64	0.3652
CONCRETE	0.0100	32	0.3592
CONCRETE	0.0100	64	0.3609
ENERGY	0.0001	32	0.2123
ENERGY	0.0001	64	0.2140
ENERGY	0.0010	32	0.2156
ENERGY	0.0010	64	0.2134
ENERGY	0.0100	32	0.2123
ENERGY	0.0100	64	0.2165
PROTEIN	0.0001	32	0.7170
PROTEIN	0.0001	64	0.7210
PROTEIN	0.0010	32	0.7280
PROTEIN	0.0010	64	0.7281
PROTEIN	0.0100	32	0.7943
PROTEIN	0.0100	64	0.7937
YACHT	0.0001	32	0.6096
YACHT	0.0001	64	0.6072
YACHT	0.0010	32	0.6188
YACHT	0.0010	64	0.5277
YACHT	0.0100	32	0.5355
YACHT	0.0100	64	0.6066

Table 10. RMSE performance for non-Bayesian DE with 12 members across different settings of weight decay and batch size. All networks have two hidden layers of 16 neurons each and tanh activation.

network with 16 neurons in each hidden layer and tanh activation. This results in about half a second and 8 seconds for 12 parallel samples for the small and large data sets, respectively. Decreasing the number of warmup samples as proposed in DEI-MCMC reduces the number of required samples per chain by more than half, resulting in much faster model fits.