Bayesian Neural Networks:
DEI MCMC
Symmetry & Mode Connectivity

[s25] BA-Seminar: Probabilistic ML

Thomas Witzani

2025-06-30

Table of contents

Abstract

Bayesian neural networks (BNNs) generalize standard neural networks (NNs) by placing probability distributions over weights rather than relying on single point estimates, which in turn enables principled quantification of predictive uncertainty. In this work, I develop and evaluate a Deep-Ensemble-Initialized MCMC (DEI-MCMC) pipeline that trains a small ensemble of randomly-seeded networks, canonicalizes each network by neuron-sorting and sign-fixing, then clusters those canonicalized versions via cosine distance to select representatives on truly distinct posterior modes and uses those representatives to seed parallel “No U-Turn Sampler (NUTS)” (adaptive Hamiltonian Monte Carlo (HMC)) chains in Stan.

In a simulation study on a noisy sinusoidal function I tuned the pipeline and showed that the empirical 90 % posterior predictive intervals closely match the ±1.645 bounds implied by the known Gaussian noise. Finally, I apply DEI-MCMC to the UCI Airfoil Self-Noise dataset and demonstrate that symmetry-aware initialization broadens credible intervals only where genuine posterior ambiguity exists, while preserving tight uncertainty elsewhere. Convergence diagnostics (split-$\bar{R}$, bulk and tail ESS), posterior predictive checks, and partial-dependence uncertainty bands all confirm that the sampler mixes well and yields calibrated predictive performance. These results establish that symmetry-aware ensemble initialization delivers efficient exploration of complex BNN posteriors and robust, interpretable uncertainty estimates with moderate computational overhead.

1 Introduction & Motivation

1.1 Motivation

A NN is a parametric function

$$f_\theta : \mathbb{R}^d \to \mathbb{R}, \quad \theta = \{W^{(1)}, b^{(1)}, \dots, W^{(L)}, b^{(L)}\},$$

defined layer-wise by

$$\begin{aligned} &h^{(0)} = x, \\ &h^{(\ell)} = \sigma\big(W^{(\ell)}\, h^{(\ell-1)} + b^{(\ell)}\big), \quad \ell = 1, \dots, L-1, \\ &f_\theta(x) = W^{(L)}\, h^{(L-1)} + b^{(L)} \end{aligned}$$

where each σ is a nonlinear activation. By the Universal Approximation Theorem, for sufficiently large width this family can approximate any continuous function on a compact domain up to a closeness $\epsilon > 0$.

In a BNN we treat θ as a random variable with a prior $p(\theta)$ and observe the data

$$D = \{(x_i, y_i)\}_{i=1}^N$$

under the likelihood

$$p(D \mid \theta) = \prod_{i=1}^{N} p(y_i \mid f_\theta(x_i)).$$

Bayes' rule defines the posterior as

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\displaystyle\int p(D \mid \theta)\, p(\theta)\, d\theta}$$

which is a highly multimodal distribution in the $dim(\theta)$-dimensional parameter space.

The advantage of BNNs is that, instead of collapsing to a single point estimate $\hat{\theta}$, the network maintains a full posterior $p(\theta \mid D)$, which simultaneously quantifies the *aleatoric uncertainty* through the likelihood $p(y \mid f_\theta(x))$, and *epistemic uncertainty* through the spread of the posterior itself.

For a new input x* the BNNs posterior predictive distribution is

$$p(y^* \mid x^*, D) = \int p(y^* \mid f_\theta(x^*))\, p(\theta \mid D)\, d\theta$$

and the corresponding posterior-mean prediction (which is a point forecast) is

$$\mathbb{E}[y^* \mid x^*, D] = \int f_\theta(x^*)\, p(\theta \mid D)\, d\theta.$$

1.2 Challenges

The posterior predictive distribution admits a closed-form solution only under the restrictive, idealized assumption of conjugate priors and likelihoods. In practice, we almost always prefer richer priors and more realistic noise models, so we must fall back on approximate inference methods, namely MCMC.

Another challenge is the sheer dimensionality of the parameter space in a BNN. In a 5-16-16-16-8-1 architecture the total number of trainable parameters is 513. In such high dimensions naïve MCMC samplers suffer from various problems such as vanishing acceptance rates, slow mixing and exponential cost because the volume of a high-dimensional space grows so fast that covering it uniformly is infeasible.

Moreover, BNN posteriors are inherently multimodal due to simple symmetries in the weight space. Two parameter settings θ and $\hat{\theta}$ are called equioutput if they define exactly the same input-output map,

$$f_{\hat{\theta}}(x) = f_\theta(x) \quad \forall x$$

Even a tiny network exhibits many such symmetries.

The first type of symmetry arises through neuron permutations. In any hidden layer, the perceptrons are exchangeable: if you permute the columns of $W^{(l)}$ and simultaneously permute the rows of $W^{(l+1)}$, the overall function f_θ remains unchanged. The number of symmetries that arise through this mechanism is $\prod_{\ell=1}^{L-1} n_\ell!$.

The second type of symmetry comes from sign flips: whenever the activation σ is odd (e.g. tanh), you can pick any hidden neuron in layer ℓ, multiply it's incoming weights and bias by -1, and at the same time multiply it's outgoing weights by -1, without changing f_θ. Since each of the $\sum_{\ell=1}^{L-1} n_\ell$ hidden neurons can be flipped independently, there are $2\sum_{\ell=1}^{L-1} n_\ell$ distinct sign-flip symmetries.

If an MCMC sampler is unaware of these symmetries, it will waste iterations traversing equioutput duplicates, artificially inflating both the posterior's modal count and its parameter-space variance. Because many draws add no new information, the effective sample size collapses and Monte Carlo error in credible-interval estimation grows, causing the estimated intervals to appear overly conservative—even though the true predictive uncertainty remains unchanged. By collapsing permutation and sign-flip symmetries up front, the sampler is constrained to explore only genuinely distinct modes, restoring effective sample size, reducing Monte Carlo error, and yielding credible intervals that reflect real functional variation rather than redundant copies.

1.3 Objective

The goal of this paper is to develop and validate a Deep-Ensemble-Initialized MCMC (DEI-MCMC) workflow for BNNs that achieves efficient posterior exploration and well-informed uncertainty estimates by removing trivial symmetries in weight space. Concretely, I aim to:

1. Train a small ensemble of M randomly-seeded feed-forward NNs, $\{\theta^{(m)}\}_{m=1}^{M}$.
2. Canonicalize each $\theta^{(m)}$ by neuron-sorting and sign-fixing, so that NNs belonging to the same symmetry group collapse to identical (or nearly identical) canonical forms.
3. Cluster the resulting canonical NNs (using cosine distance) and then select $K(\leq M)$ representatives to ensure each final ensemble member lies on a functionally distinct posterior peak.
4. Initialize K parallel NUTS chains with Stan starting at these $\{\tilde{\theta}^{(k)}\}_{k=1}^{K}$.
5. Assess convergence and evaluate uncertainty calibration via credible-interval coverage and posterior predictive checks.

2 Related Work

2.1 BNN Posterior Sampling Methods

Hoffman and Gelman (2011) introduces NUTS, an extension of HMC that discards the manually chosen trajectory length L. NUTS keeps doubling the leapfrog path until the simulated momentum would reverse toward the start—the "no-u-turn" stop. Together with primal–dual averaging for adaptive step size, this yields a self-tuning, gradient-based MCMC method that matches or beats well-tuned HMC without user calibration. Because manually selecting L and step size in vanilla HMC is notoriously sensitive and labor-intensive, NUTS makes practical Bayesian inference vastly easier and more robust.

This algorithm is now used by default in Stan, the R library for Bayesian modeling and inference that I used in this project.

2.2 Deep-Ensemble Initialisation (DEI)

Sommer et al. (2024) show that a deep ensemble (a handful of independently initialised and fully trained neural networks) already lands it's members in separate high-probability basins of the Bayesian posterior. Starting HMC chains from those pre-optimised weights therefore eliminates much of the costly burn-in phase. Complementary empirical evidence in Izmailov et al. (2021) confirms that such ensemble seeds cover the dominant modes encountered by standard HMC, while chains started from random points often fail to reach them within a practical time & compute budget. In this project I follow that recipe: train a small ensemble, take each member's weights as an initial state, and launch parallel NUTS chains from those mode-finding seeds to achieve efficient convergence and broad coverage.

2.3 Symmetry Detection and Elimination

Wiese et al. (2023) demonstrate that permutation- and sign-flip symmetries create exponentially many equi-output modes that cripple MCMC efficiency. They introduce an inexpensive canonicalisation map—sorting neurons within each layer and fixing each neuron's sign—that collapses every symmetry class to a single representative while preserving the log-likelihood. Sampling in this reduced space markedly increases effective sample size and stabilises $\bar{R}$ diagnostics. I adopt this canonicalisation as a preprocessing step so that subsequent NUTS chains explore only genuinely distinct regions of the BNN posterior.

2.4 Mode Connectivity and Sample-Based Inference

Linearly interpolating between two SGD solutions usually produces a high-loss ridge, but a series of works beginning with Garipov et al. (2018) and refined by Fort, Hu, and Lakshminarayanan (2019) show that curved low-loss paths often exist, implying that many apparent local optima belong to a larger, connected manifold. Most recently, Sommer et al. (2024) connect such paths directly to Bayesian inference: they sample along the connector using tempering, obtaining predictive distributions that rival full HMC at a fraction of the cost. Although my pipeline focuses on isolated mode initialisation rather than traversing connectors, these findings reinforce the idea that weight-space distance does not automatically translate to functional diversity, motivating my additional clustering step using cosine similarity in the canonical parameter space.

3 Methods

3.1 DEI-MCMC Pipeline

3.2 Stan Implementation

3.3 Symmetric Mode Removal

4 Simulation Study

4.1 Synthetic Dataset Generation

4.2 Training Deep Ensembles

4.3 Stan Model Setup

4.4 DEI-MCMC Experiments

4.5 Evaluation Metrics

4.6 Function Recovery Analysis

5 Application to a Real Dataset

5.1 Dataset Selection and Adaptation

5.2 DEI-MCMC Experiments

5.3 Performance Evaluation

6 Discussion

6.1 Simulation Results Summary

6.2 Real-Data Case Summary

6.3 Assessment Without Symmetry Removal

6.4 Assessment With Symmetry Removal

7 Conclusion

8 Outlook

Acknowledgements

References

9 References will be generated from references.bib

10 Appendix

10.1 Additional Details

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. 2019. "Deep Ensembles: A Loss Landscape Perspective."

Garipov, Timur, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs."

Hoffman, Matthew D., and Andrew Gelman. 2011. "The No-u-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo."

Izmailov, Pavel, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. 2021. "What Are Bayesian Neural Network Posteriors Really Like?"

Sommer, Emanuel, Lisa Wimmer, Theodore Papamarkou, Ludwig Bothmann, Bernd Bischl, and David Rügamer. 2024. "Connecting the Dots: Is Mode-Connectedness the Key to Feasible Sample-Based Inference in Bayesian Neural Networks?" arXiv. https://doi.org/10.48550/ARXIV.2402.01484.

Wiese, Jonas Gregor, Lisa Wimmer, Theodore Papamarkou, Bernd Bischl, Stephan Günnemann, and David Rügamer. 2023. "Towards Efficient MCMC Sampling in Bayesian Neural Networks by Exploiting Symmetry." arXiv. https://doi.org/10.48550/ARXIV.2304.02902.