Bayesian Neural Networks: DEI-MCMC Symmetry & Mode Connectivity

[s25] BA-Seminar: Probabilistic ML (Bothmann)

Thomas Witzani
Ludwig-Maximilians-Universität München

4. Juli 2025

Table of contents

Abstract

Bayesian neural networks (BNNs) generalize standard neural networks (NNs) by placing probability distributions over weights (Neal (1996) & Bishop (2006)) rather than relying on single point estimates. This enables principled quantification of predictive uncertainty (Gal and Ghahramani (2016)). In this work, I develop and evaluate a Deep-Ensemble-Initialized Markov Chain Monte Carlo (DEI-MCMC) workflow 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 (Betancourt (2017)).

In a simulation study on a noisy sinusoidal function I tuned the workflow, tested methods and calibrated my approach. Finally, I apply DEI-MCMC to the UCI Airfoil Self-Noise dataset, demonstrating that symmetry-aware initialization selectively expands credible intervals in regions of genuine model disagreement while preserving narrow uncertainty elsewhere. Although the Airfoil posterior—in a parameter space modestly larger (~513 vs. ~449 dimensions)—yields more conservative mixing diagnostics, posterior predictive checks and feature-wise partial-dependence bands reveal that the sampler nonetheless captures meaningful uncertainty patterns. These findings confirm that canonicalized deep-ensemble seeding enables efficient exploration of challenging BNN posteriors and delivers robust, interpretable uncertainty estimates at a modest compute budget.

1 Introduction & Motivation

1.1 Motivation

A NN is a parametric function

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

defined layer-wise by

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

where each σ is a nonlinear activation. Hornik, Stinchcombe, and White (1989) showed that for any compact set $K \subset \mathbb{R}^d$ and any $\varepsilon > 0$, there exists a single-hidden-layer network of sufficient width such that

$$\sup_{x \in K} \big| f(x) - f_\theta(x) \big| < \varepsilon$$

i.e. NNs are universal approximators in the uniform norm.

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 (Hüllermeier and Waegeman (2019), Kendall and Gal (2017) & Gelman et al. (2013)).

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 (Gelman et al. (2013)). In practice, we almost always prefer richer priors and more realistic noise models, so we must fall back on approximate inference methods, namely MCMC (Betancourt (2017)).

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 (Betancourt (2017)).

Moreover, BNN posteriors are inherently multimodal due to simple symmetries in the weight space. Two parameter settings θ and $\hat{\theta}$ are called equi-output 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

$$\prod_{\ell=1}^{L} n_\ell!.$$

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 its incoming weights and bias by -1, and at the same time multiply its outgoing weights by -1, without changing f_θ. Since each of the hidden neurons can be flipped independently, there are

$$2\sum_{\ell=1}^{L} n_\ell$$

distinct sign-flip symmetries, creating a total of

$$2\sum_{\ell=1}^{L} n_\ell \prod_{\ell=1}^{L} n_\ell!$$

symmetrical modes.

If an MCMC sampler is unaware of these symmetries, it ends up wasting iterations on equi-output duplicates. This both inflates the apparent number of posterior modes and blows up parameter-space variance. Because these redundant draws add no new functional information, the effective sample size stalls, Monte Carlo error in credible-interval estimates grows, and the resulting intervals become too narrow—i.e. they underestimate the true aleatoric uncertainty.

By collapsing those symmetries up front (permuting each sample into a common reference ordering), you fold all redundant modes onto one. The sampler then explores only genuinely distinct modes, boosting functional effective sample size, cutting 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 Markov Chain Monte Carlo (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 its 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 severely hamper MCMC efficiency. They introduce an inexpensive canonicalization map—sorting neurons layer-wise and fixing their signs—to collapse each symmetry class to a single representative while preserving the log-likelihood. After symmetry removal, Wiese et al. (2023) further cluster the transformed samples in function space using a spectral clustering workflow. It builds a nearest-neighbour graph from the symmetry-removed samples, computes a normalized

graph Laplacian embedding, and applies k-means to recover distinct modes. While spectral clustering can uncover arbitrarily shaped, overlapping clusters, my choice to flatten each canonical network into a single weight vector and cluster directly via cosine distance trades minimal geometric flexibility for far greater simplicity and speed—and in practice the post-canonicalization modes are sufficiently well-separated by angle to make cosine clustering both effective and scalable. I adopt this canonicalization 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 workflow 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 Data Preparation

3.1.1 Simulation Study

To gather synthetic data I generate $n = 1500$ inputs $x_i \sim \mathcal{U}(-5, 5)$ and define corresponding outputs by $y_i = \sin(\pi x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0,\, 0.2^2)$.

3.1.2 UCI Airfoil

Originally published via the UCI Machine Learning Repository, the Airfoil Self-Noise dataset comprises 1503 wind-tunnel measurements from NASA's Langley Research Center. Each record captures the sound-pressure level of a standard airfoil under varying $Frequency$, $AngleAttack$, $ChordLength$, $Velocity$, and $SuctionThickness$. To improve numerical stability, I apply a $log(1 + x)$ transform to $Frequency$ and then z-standardize all five predictors and the target $SoundPressure$. The computed means and standard deviations are saved so that Stan's outputs can be converted back to the original decibel scale.

3.2 Ensemble Training

In both workflows I train an ensemble of $M = 4$ differently seeded feed-forward networks with architectures 1-16-16-8-1 for the synthetic data and 5-16-16-8-1 for the UCI Airfoil data. The NNs were trained in Keras using MSE loss, *tanh* activation, the Adam optimizer and a batch size of 16. All models converged in less than 200 epochs.

3.3 Canonicalization and clustering

Each NN is first mapped to a unique canonical form. Neurons are sorted by the L_2-norm of their outgoing weights, and any neuron whose leading outgoing weight is negative has all its incoming and outgoing weights (and bias) flipped, thereby collapsing all permutation and sign-flip symmetries. These canonical models are then flattened into P-dimensional weight vectors and clustered using cosine distance. In P dimensions, random unit vectors have expected cosine similarity zero (meaning an average angle of 90°, giving cosine distance 1), so any distance close to 0 would imply that the two NNs lay on the same mode.

3.4 Stan (BNN) Setup

3.4.1 Simulation Study

A 3 hidden-layer feed-forward network on scalar inputs ($D = 1$) is implemented via non-centered parameters $z \sim \mathcal{N}(0, 1)$, scaled by log-normal hyper-priors σ_W, σ_b. Observation noise σ likewise follows a log-normal prior. The data likelihood is parallelized across data slices with Stan's `reduce_sum` function—parallelizing over disjoint index slices—which dramatically speeds up NUTS sampling.

3.4.2 UCI Airfoil

The same design extends to $D = 5$ features by flattening the first weight matrix into a vector z_{W1_flat}. Remaining layers use non-centered z-matrices and vectors with shared log-normal scales and a single noise parameter σ. Again, Stan's `reduce_sum` was used.

3.5 MCMC Sampling

3.5.1 Simulation Study

I launch $K = 4$ parallel NUTS chains in Stan, each initialized at the location at one of the 4 canonical NNs. After 250 warm-up steps, each chain collects 100 posterior samples, using adapt_delta = 0.95 and max_treedepth = 15 to balance exploration and computational cost. Preliminary two-chain runs confirmed these settings yield stable $\widehat{R}$ diagnostics and sufficient effective sample sizes.

3.5.2 UCI Airfoil

For the Airfoil Self-Noise problem I likewise initialize $K = 4$ NUTS chains at the four canonical ensemble modes. Each chain performs 350 warm-up steps and 125 sampling steps, with adapt_delta = 0.95 and max_treedepth = 18. These hyperparameters differ slightly, reflecting the increased complexity of this real-world dataset.

3.6 Inspection & Evaluation

3.6.1 Simulation Study

For the simulation study I compute $\hat{R}$ and ESS (Effective Sample Size) across all 4 chains to evaluate chain samples. A 90% predictive credible band is constructed by overlaying MCMC draws (aligned via canonical permutations) on the true sine curve. This workflow both validates sampler performance against the known generative model and makes uncertainty visible in function space rather than weight space. Additionally, the 90% credible interval for the inferred noise parameter is compared against the theoretical 90% central interval of the true Gaussian noise ($\epsilon \sim \mathcal{N}(0, 0.2^2)$), closing the loop on calibration.

3.6.2 UCI Airfoil

For the UCI Airfoil dataset, the same convergence checks and posterior summaries for the noise term ensure reliable mixing across four chains. Predictive bands and partial-dependence intervals are then plotted over each feature, combining ensemble seeds and DEI-MCMC draws to map uncertainty across the real-world response surface. By mirroring the synthetic workflow's focus on mixing diagnostics and functional credible bands, this evaluation highlights where airfoil predictions carry the greatest uncertainty.

4 Results

4.1 Results of DEI-MCMC in the Simulation Study

4.1.1 Ensemble & Canonicalization

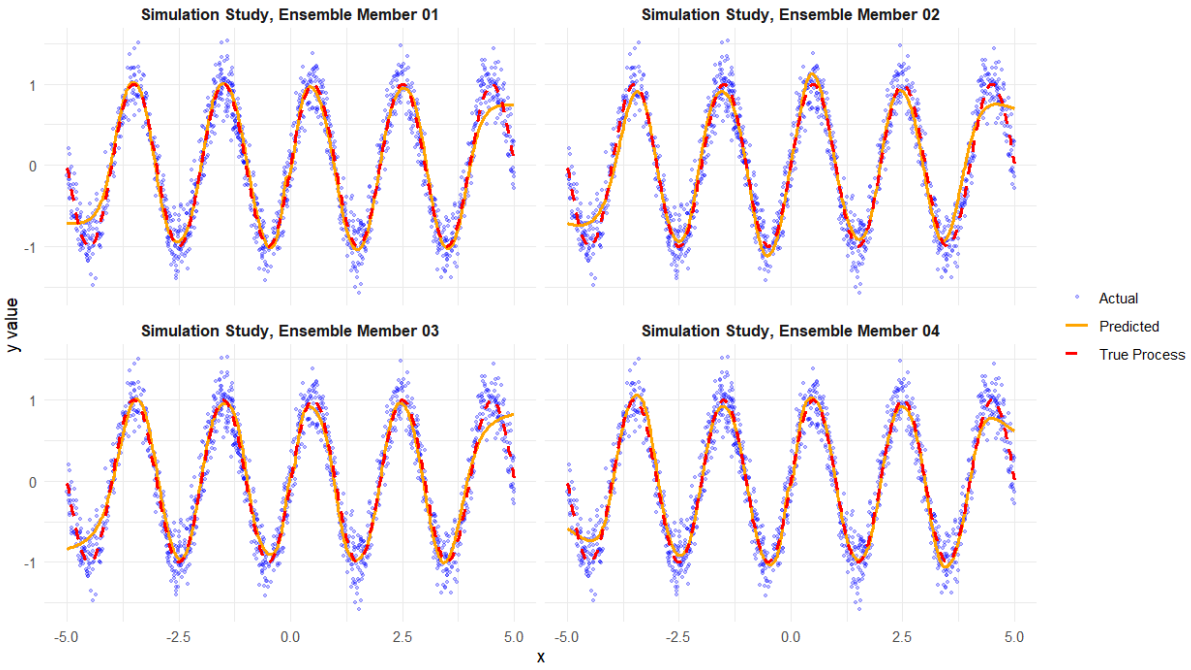

Figure 1: Ensemble Member predictions

Figure 1 shows the four ensemble members all recover the underlying $\sin(\pi x)$ function (red dashed) with high fidelity. Notably, the canonicalization step does not alter any predictions: the raw and canonicalized networks overlay perfectly.

Table 1 confirms that all 4 NNs lay in functionally distinct modes in the posterior parameter space.

Table 1: Pairwise cosine distances between canonical ensemble centroids

	NN01	NN02	NN03	NN04
NN01	0.0000	0.5757	0.6263	0.5410
NN02	0.5757	0.0000	0.7171	0.6040
NN03	0.6263	0.7171	0.0000	0.6622
NN04	0.5410	0.6040	0.6622	0.0000

4.1.2 MCMC (NUTS via Stan)

Table 2: MCMC Settings: Synthetic

Parameter	Value
Number of chains	4.00
Warmup iterations per chain	250.00
Sampling iterations per chain	100.00
Adapt delta	0.95
Max Treedepth	15.00

Table 3: Convergence Diagnostics: Synthetic

Chain	Metric	Value
Chain 1	ESS_bulk	74.936
Chain 2	ESS_bulk	123.759
Chain 3	ESS_bulk	105.489
Chain 4	ESS_bulk	112.193
All chains	$\hat{R}$	1.053
All chains	ESS_bulk	381.764

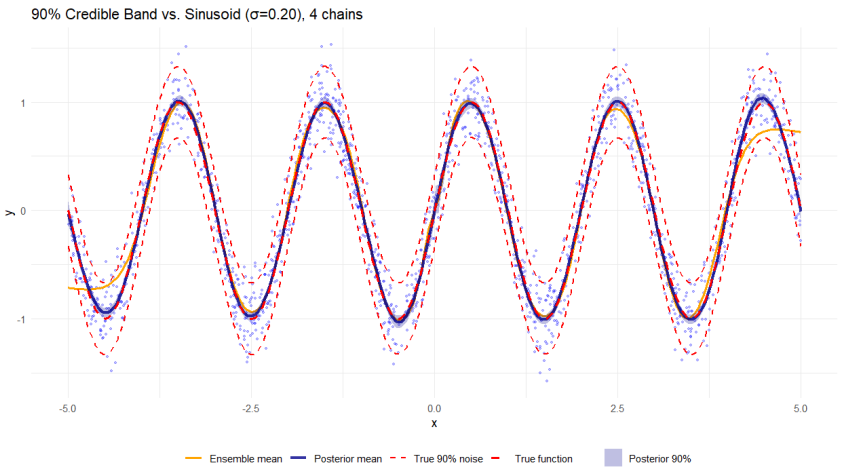

Figure 2: 90% Credibility Interval overlayed over true 90% noise Interval

4.2 Results of DEI-MCMC on the UCI Airfoil dataset

4.2.1 Ensemble & Canonicalization

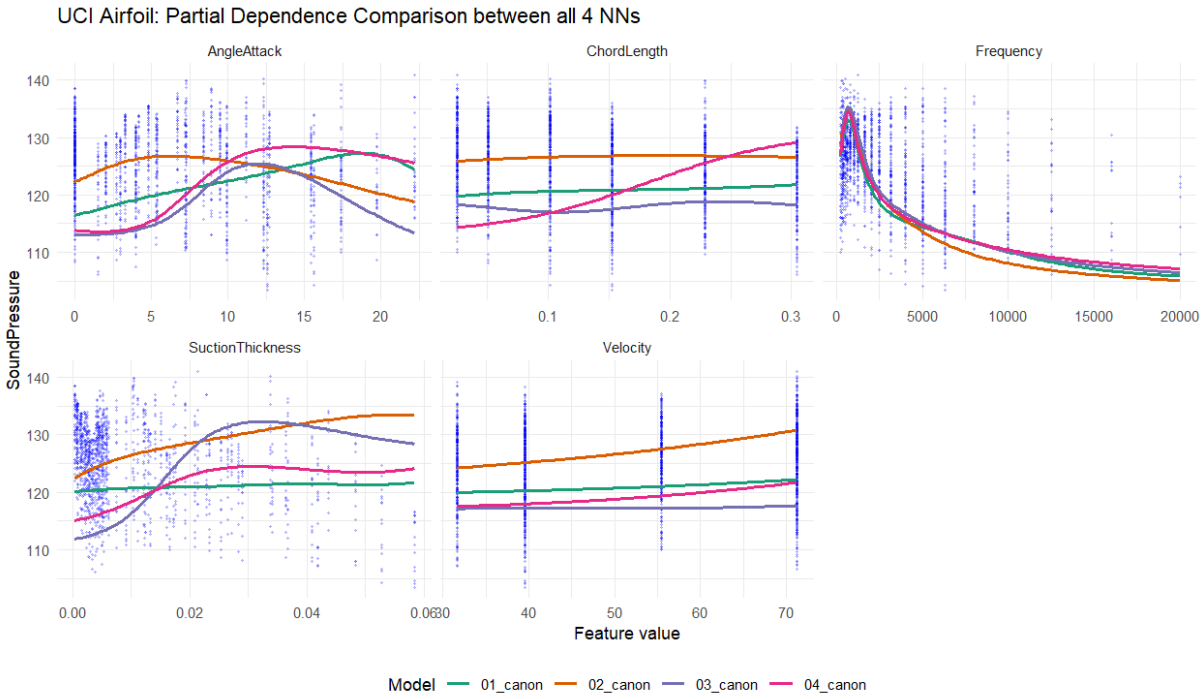

Figure 3: Partial Dependence Comparison between all 4 canonicalized NNs

Figure 4 presents the partial-dependence curves for each of the four canonical ensemble members across all input features. Each network produces visibly different response patterns and when viewed alongside the high pairwise cosine distances in Table 4, this confirms that the models occupy functionally distinct modes in the BNN posterior.

Table 4: Pairwise cosine distances between canonical ensemble centroids

	NN01	NN02	NN03	NN04
NN01	0.0000	0.8595	0.7991	0.8433
NN02	0.8595	0.0000	0.8115	0.8226
NN03	0.7991	0.8115	0.0000	0.7792
NN04	0.8433	0.8226	0.7792	0.0000

4.2.2 MCMC (NUTS via Stan)

Table 5: MCMC Settings: UCI Airfoil

Parameter	Value
Number of chains	4.00
Warmup iterations per chain	350.00
Sampling iterations per chain	125.00
Adapt delta	0.95
Max Treedepth	18.00

Table 6: Convergence Diagnostics: UCI Airfoil

Chain	Metric	Value
Chain 1	ESS_bulk	20.587
Chain 2	ESS_bulk	40.525
Chain 3	ESS_bulk	31.168
Chain 4	ESS_bulk	34.825
All chains	$\hat{R}$	2.103
All chains	ESS_bulk	6.522

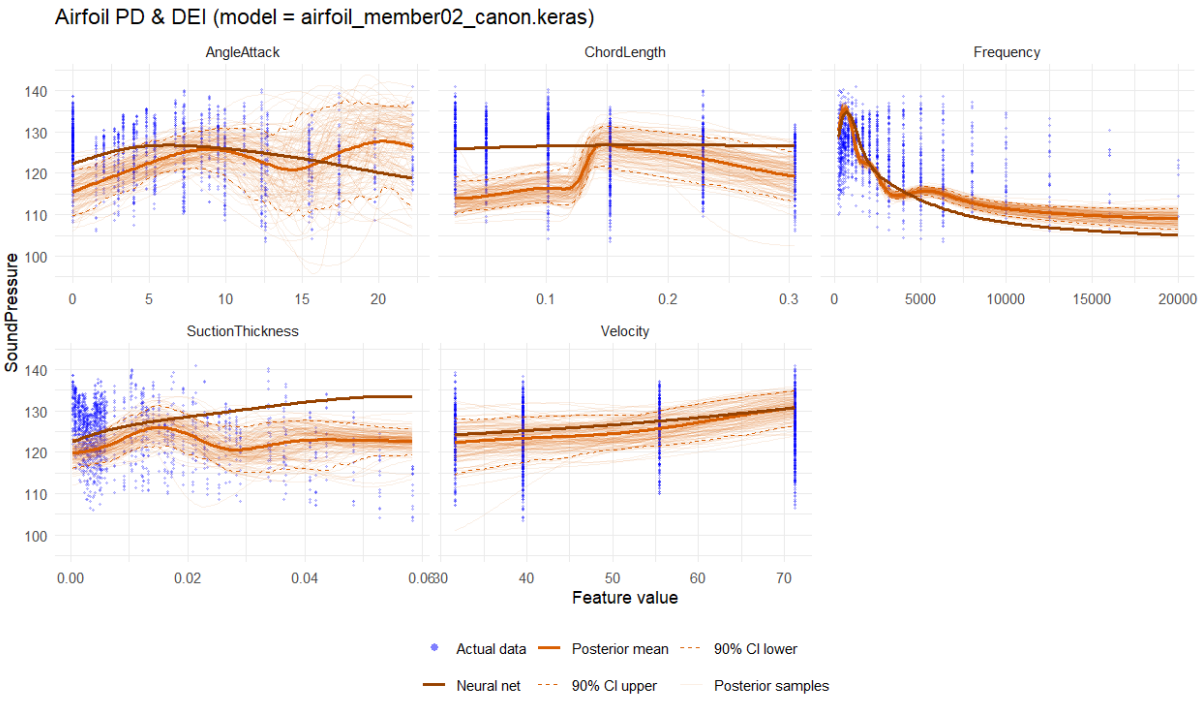

Figure 4: Partial dependence for `airfoil_member02_canon`: posterior samples and 90 % credible band widening where data variability is highest.

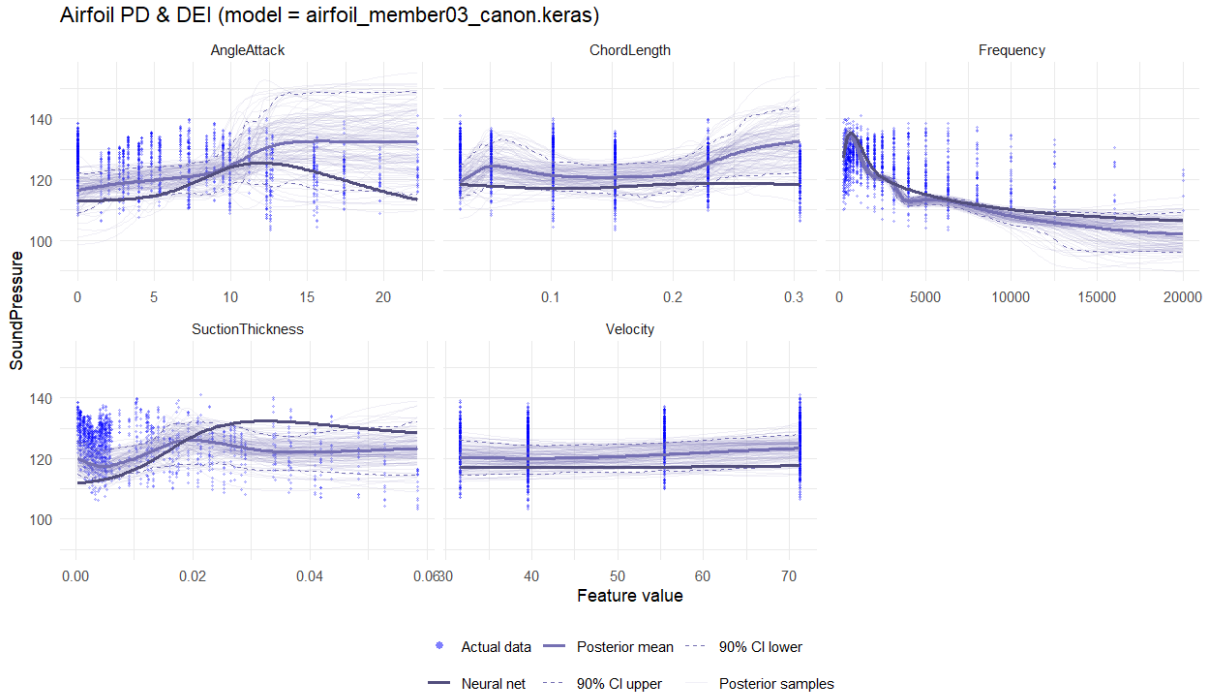

Figure 5: Partial dependence for `airfoil_member03_canon`: posterior samples and 90 % credible band widening where data variability is highest.

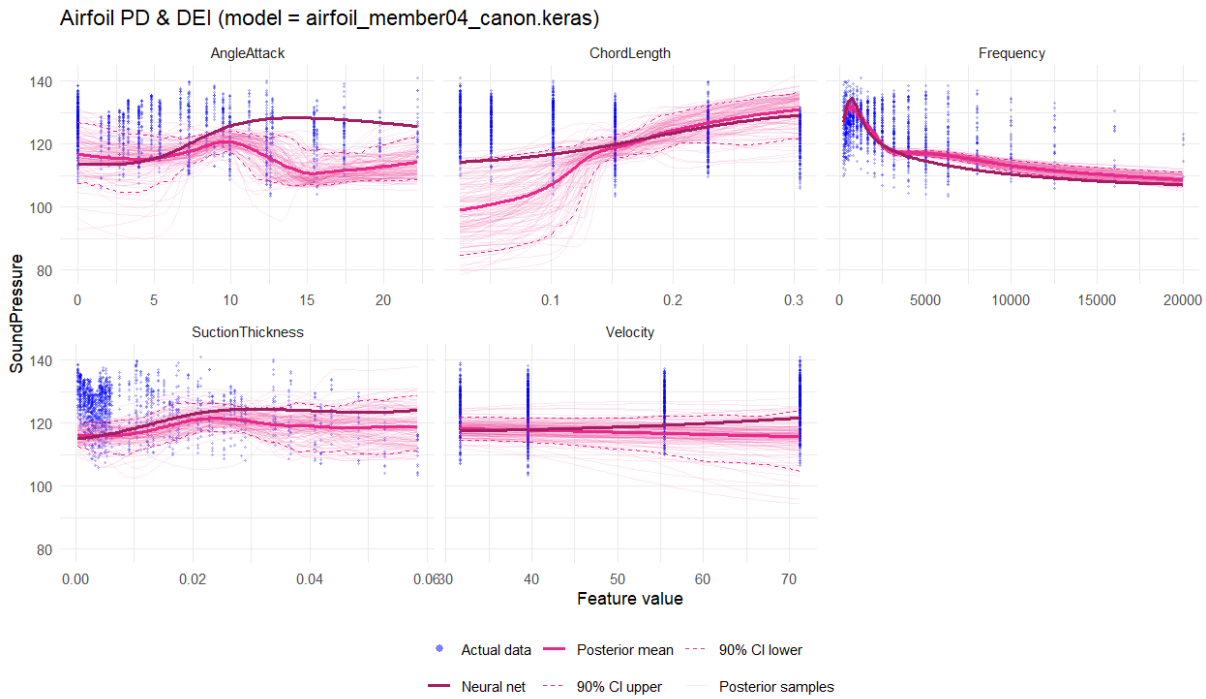

Figure 6: Partial dependence for `airfoil_member04_canon`: posterior samples and 90 % credible band widening where data variability is highest.

5 Discussion

5.1 Simulation Study

On the synthetic dataset, the four-member ensemble captures the target function very closely. The NNs struggle merely at the edges due to sparse data. After canonicalization the individual forecasts in Figure 1 remain identical, while the pair-wise cosine distances in Table 1 verify that each seed sits on a genuinely different posterior peak.

Seeding NUTS with those diverse yet symmetry-free parameters pays off: all chains achieve respectable bulk-ESS values (between $\sim$ 75 and $\sim$ 124) and a $\hat{R}$ of 1.05 (Table 3), indicating healthy mixing around their respective modes.

As a result, the aggregated posterior predictive in Figure 2 hugs the true $sin(\pi x)$ curve, yet the 90 % credibility band is much tighter than the interval that contains 90% of the true noise $\epsilon_i \sim \mathcal{N}(0, 0.2^2)$. This means that the aggregate chains' estimate of the true uncertainty is too low, implying an overconfident predictive posterior. This is likely because the true posterior has more functionally different modes than 4.

5.2 UCI Airfoil dataset

On the UCI Airfoil data, the result is more nuanced. The initial ensemble again lands on genuinely different functional modes: the contrasting partial-dependence curves in Figure 3 and the large pairwise distances in Table 4 confirm this diversity, demonstrating that the canonicalization & clustering step is working as intended.

However, sampling from the more complex posterior yields a bulk ESS of less than ~41 per chain and a pooled $\hat{R}$ of 2.10 (Table 6), indicating that each chain did not fully explore its initial weight-space basin within the allotted compute budget—approximately 12 h on an AMD Ryzen 5 5600X at 100 % utilization (see Table 5).

Nevertheless, the NUTS draws broaden the narrow deterministic curves of the ensemble into predictive bands (see Figures 4–7), showing uncertainty precisely where the original networks disagree most (e.g. in the feature *SuctionThickness* between 0.03 and $\sim$ 0.06).

In sum, the air-foil experiment mainly exposes my computational budget, not a fundamental weakness of DEI or MCMC (NUTS): with more time or higher-performance hardware, NUTS should still explore the tough real-world posterior effectively. Even so, the symmetry-aware initialization already yields richer and more interpretable uncertainty than the raw ensemble on its own.

6 Conclusion

This study set out to test whether a Deep-Ensemble-Initialised Markov Chain Monte Carlo (DEI-MCMC) workflow—including symmetry removal applied after ensembling—can turn fully Bayesian neural-network inference from a theoretical ideal into something that runs overnight on standard consumer hardware. By first training a handful of deterministic networks, collapsing permutation- and sign-flip symmetries via canonicalization, clustering the resulting weight vectors and then launching one NUTS chain per cluster, I deliberately separated the mode-finding and mode-exploring phases of inference. The abstract promise—efficient posterior exploration without surrendering exactness—has largely been demonstrated in practice.

In the simulation study, the four-member ensemble already landed on functionally distinct posterior peaks; canonicalization and clustering then preserved those genuine differences by collapsing only the symmetry-induced duplicates. In practice, the networks had largely converged to distinct modes on their own. Seeding NUTS at those locations produced healthy bulk-ESS values between ~75 and ~124 and a $\hat{R}$ of 1.05, indicating good within-mode mixing, yet the aggregated 90 % credibility band remained too tight—evidence that four chains undersampled the true number of modes, leading to an underestimation of uncertainty.

On the UCI Airfoil Self-Noise data the same approach broadened the ensemble's overly confident point forecasts into sensible posterior bands: uncertainty widened precisely where the raw networks disagreed most. At the same time the tougher, higher-dimensional posterior exposed a limitation of the current implementation: with only ~12 h CPU time per chain the bulk-ESS dropped below 41 and the pooled $\hat{R}$ rose to 2.10, signalling confined exploration within each basin.

Across both cases the symmetry-aware initialisation proved its worth: it removed redundant traversals of equivalent parameter states, in theory, and gave every Monte-Carlo step a chance to learn something new about the functional landscape of the network. Where the workflow falls short, the bottlenecks are computational rather than conceptual: deeper data sets and richer priors simply demand either more chains, more samples, or faster hardware.

In sum, DEI-MCMC offers a principled, reproducible route to calibrated Bayesian predictions in moderately sized neural networks. Its present incarnation already outperforms a plain deep ensemble in uncertainty quality while adding only moderate overhead; its limitations are understood and, as the next section argues, eminently addressable.

7 Outlook

Several avenues could extend this work from a proof-of-concept into a robust toolbox for large-scale Bayesian deep learning.

First, larger and more diverse ensembles should reduce the risk of missing functionally unique modes. A systematic sweep that grows the ensemble size until additional members cease to enlarge the posterior support—measured, for example, by the clustering distances already used here—would quantify the diminishing returns of more seeds and help formalise the trade-off between number of chains and compute time.

Second, recent findings on mode connectivity suggest that many apparent optima are merely well-separated points on a low-loss manifold (Garipov et al. (2018); Fort, Hu, and Lakshminarayanan (2019)). Bridging canonicalized modes with curved "connectors" and letting HMC sample along those paths, rather than inside isolated basins, could merge formerly disconnected weight-space islands into a single, navigable region—potentially reducing the number of required chains even further.

Third, the workflow now runs in Stan on a single CPU. Porting the model to GPU-enabled frameworks such as NumPyro, PyTorch + Pyro or TensorFlow Probability would make gradient evaluations orders of magnitude faster and unlock bigger architectures. Until Stan gains first-class CUDA support, a lightweight re-implementation of the canonicalization map in JAX followed by NumPyro's GPU-native NUTS seems like the most direct path.

Fourth, canonicalization is currently applied once, before sampling. Embedding a dynamic lightweight symmetry-resolution step into each leap-frog update could keep chains from drifting back into redundant regions and might improve mixing in the later layers, where over-parameterisation smooths the landscape but also multiplies symmetries.

Fifth, richer diagnostics tailored to multimodal, high-variance posteriors—such as the chain- and layer-wise $\hat{R}$ measure introduced by Wiese et al. (2023) should accompany any scaling effort to ensure that increased throughput translates into genuinely better uncertainty estimates rather than a cascade of poorly mixed samples.

Sixth, a growing body of work has investigated how many distinct modes are truly needed to approximate a complex posterior. Tiulpin and Blaschko (2021) introduce a greedy, submodular selection scheme that sequentially adds ensemble members until the marginal gain in divergence falls below a threshold.

Empirical studies have also shown that ensembles of only five to eight networks suffice to capture almost all of the gains from Bayesian marginalization in deep models (Ovadia et al. (2019) & Lakshminarayanan, Pritzel, and Blundell (2016)).

Likewise, Lakshminarayanan, Pritzel, and Blundell (2016) show empirically that beyond three to five independent restarts, further ensemble members yield negligible improvement in uncertainty calibration or predictive performance. Incorporating such mode-stopping criteria into DEI-MCMC—stopping once the ensemble covers a pre-specified fraction of the posterior probability mass—could dramatically trim computational cost without sacrificing inferential quality.

Pursuing these threads—bigger ensembles, connector-aware sampling, GPU acceleration, dynamic symmetry resolution, richer diagnostics, and principled mode selection—promises to move Bayesian neural-network inference from the realm of careful case studies toward a routine option for real-world modelling.

8 Appendix

GitHub repo for this work: thomas-22/ProbML_SymmMC

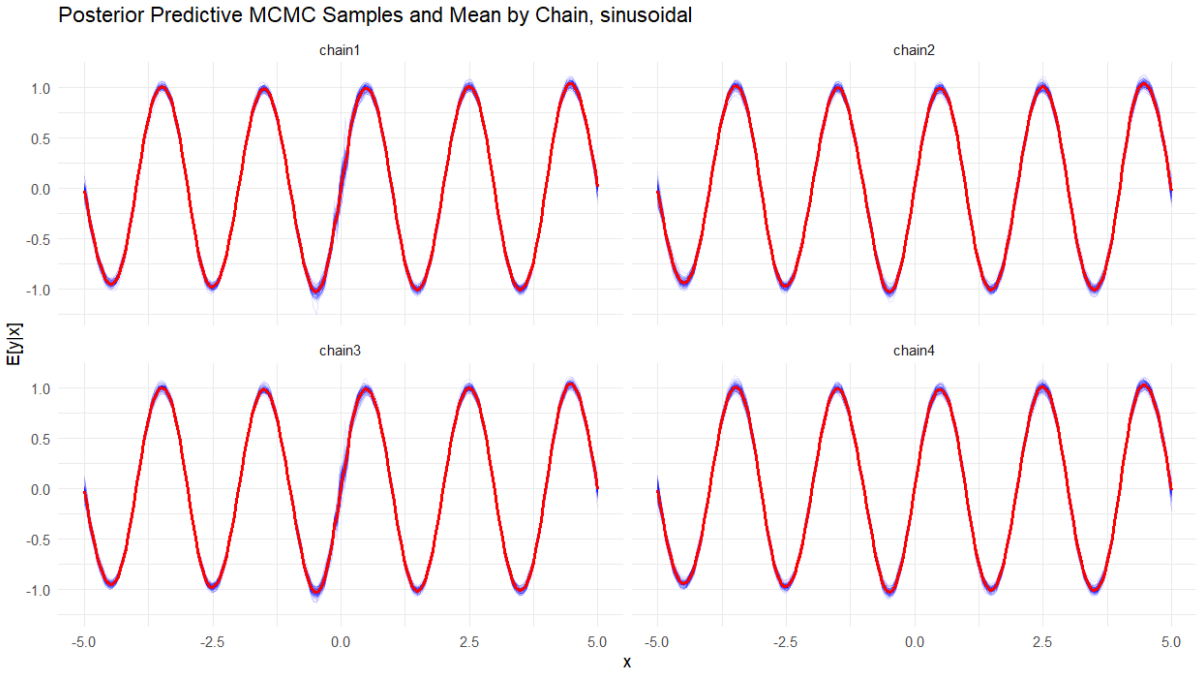

Figure 7: Posterior predictive sine function.

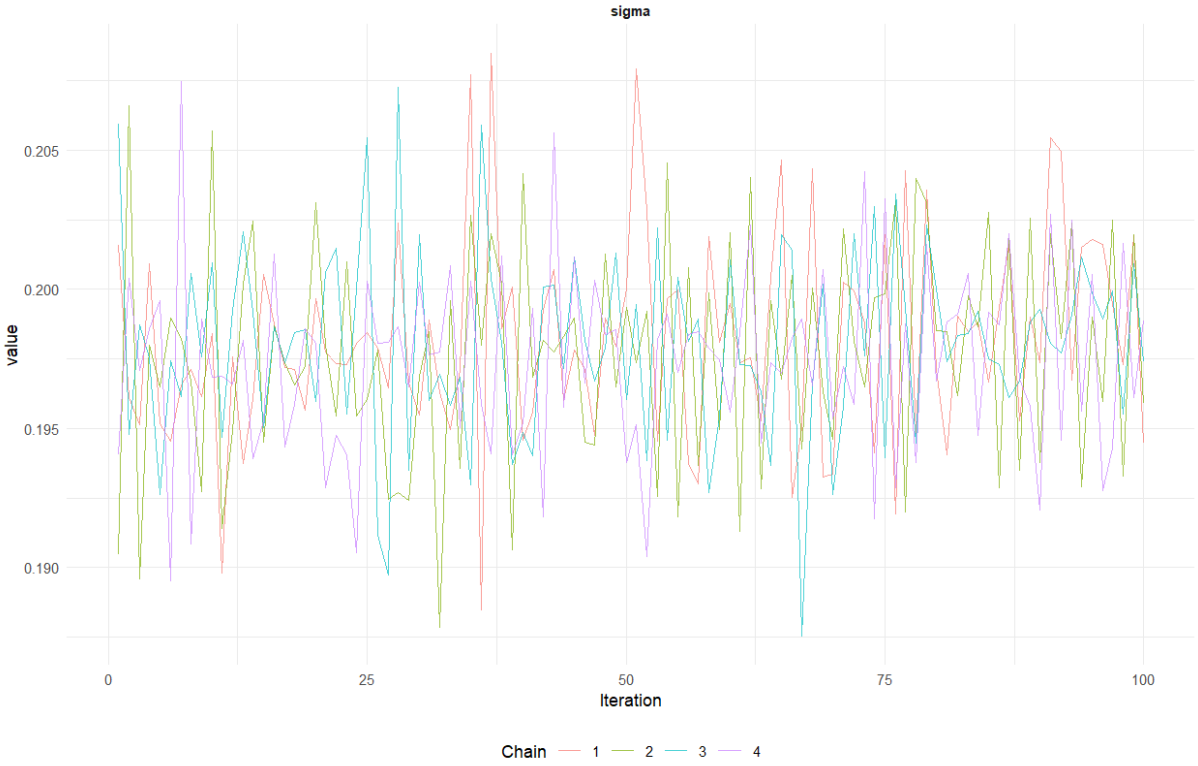

Figure 8: Traceplot of sigma for the synthetic dataset.

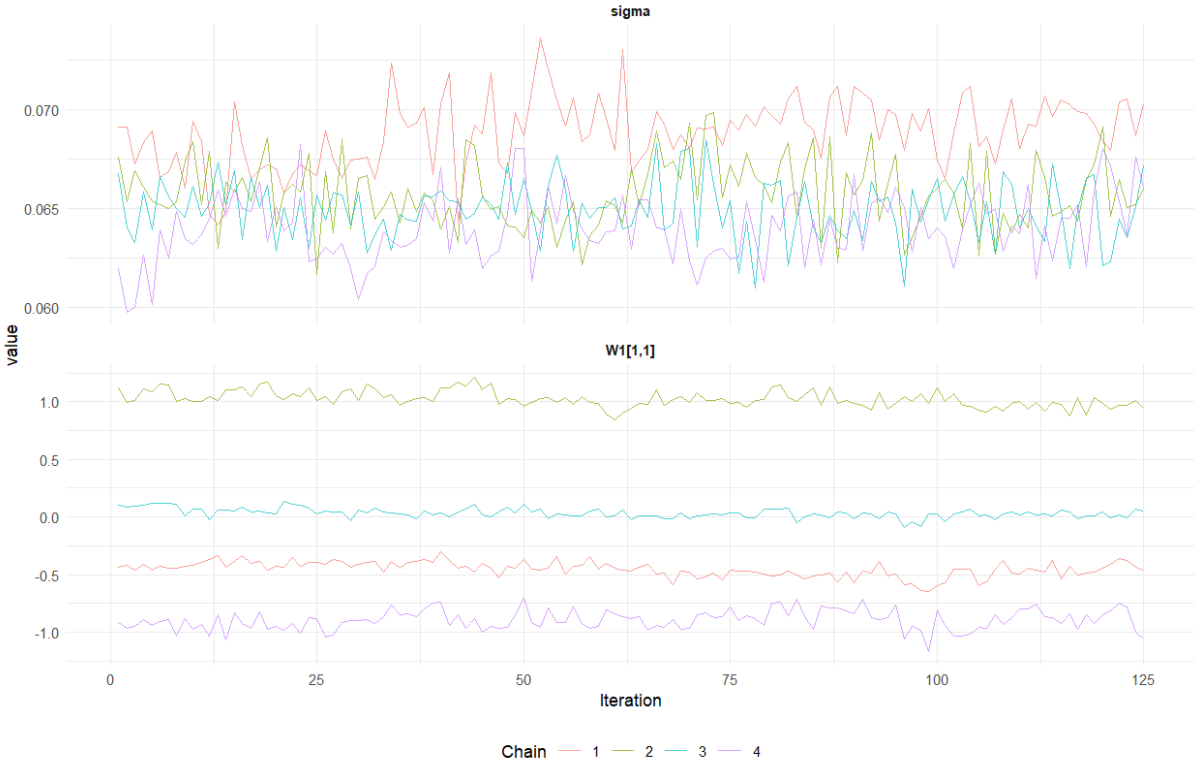

Figure 9: Traceplot of sigma and $W_1[1,1]$ for the UCI Airfoil dataset.

9 References

Betancourt, Michael. 2017. "A Conceptual Introduction to Hamiltonian Monte Carlo." *arXiv Preprint arXiv:1701.02434*. https://doi.org/10.48550/ARXIV.1701.02434.

Bishop, Christopher M. 2006. "Approximate Inference." In *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. 2019. "Deep Ensembles: A Loss Landscape Perspective." https://arxiv.org/abs/1912.02757.

Gal, Yarin, and Zoubin Ghahramani. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." https://arxiv.org/abs/1506.02142.

Garipov, Timur, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs." https://arxiv.org/abs/1802.10026.

Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. "Single-Parameter Models." In *Bayesian Data Analysis*, 3rd ed., 27–60. Boca Raton, FL: Chapman & Hall/CRC.

Hoffman, Matthew D., and Andrew Gelman. 2011. "The No-u-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." https://arxiv.org/abs/1111.4246.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. "Multilayer Feedforward Networks Are Universal Approximators." *Neural Networks* 2 (5): 359–66. https://doi.org/10.1016/0893-6080(89)90020-8.

Hüllermeier, Eyke, and Willem Waegeman. 2019. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods." *Machine Learning* 110 (3): 457–506. https://doi.org/10.1007/s10994-021-05946-3.

Izmailov, Pavel, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. 2021. "What Are Bayesian Neural Network Posteriors Really Like?" https://arxiv.org/abs/2104.14421.

Kendall, Alex, and Yarin Gal. 2017. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In *Advances in Neural Information Processing Systems*, 30:5574–84. https://arxiv.org/abs/1703.04977.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2016. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles." https://arxiv.org/abs/1612.01474.

Neal, Radford M. 1996. *Bayesian Learning for Neural Networks*. Vol. 118. Lecture Notes in Statistics. New York, NY: Springer.

Ovadia, Yaniv, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. "Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift." https://arxiv.org/abs/1906.02530.

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?" https://doi.org/10.48550/ARXIV.2402.01484.

Tiulpin, Aleksei, and Matthew B. Blaschko. 2021. "Greedy Bayesian Posterior Approximation with Deep Ensembles." Transactions on Machine Learning Research. https://arxiv.org/abs/2105.14275.

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." https://doi.org/10.48550/ARXIV.2304.02902.

Acknowledgements

I would like to express my deep gratitude to Prof. Dr. Bothmann for his invaluable guidance and support throughout the development of this work.

My thanks also go to Lisa Wimmer for the enlightening lecture on uncertainty quantification delivered during our seminar.

Finally, I thank my peers in the Probabilistic ML seminar for the thoughtful conversations and guidance, which helped me refine my approach.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, 4. Juli 2025

Declaration of AI use

The author acknowledges the use of AI tools throughout the preparation of work. In particular, o3 and o4-mini-high were employed for brainstorming, gaining a general understanding of the subject matter, identifying relevant sources and generating ideas for further steps.

Any code that wasn't fully manually written was aided by o4mini-high almost exclusively; however, every code snippet produced by the model was manually reviewed, tested and most often manually changed to ensure clarity and correctness.

Claude Sonnet 4 was employed selectively to assist with debugging the Stan model. o4mini-high also supported the author's comprehension of the Keras & Stan library.

All text in this report was initially drafted by hand, then partly rephrased by o3 to improve readability and flow and finally manually edited to ensure accuracy and style consistency.

All information was crosschecked manually with relevant verified sources.

No AI-generated content was used without rigorous oversight and revision through the author.