Optimizing Bayesian Neural Networks for Genomic Prediction: A Multi Phenotype Study on Feature Selection and Architecture

Raeein Bagheri, Justin Slater, and Yan Yan
University of Guelph, Department of Computer Science, Ontario, Canada
Emails: {bagherir, jslate04, yyan15}@uoguelph.ca

Abstract

Genome-wide association studies (GWAS) scan the genome for genetic variants, typically single nucleotide polymorphisms, whose alleles are associated with phenotypic variation across individuals. Genomic prediction uses genome-wide marker data to estimate phenotypic trait values directly from genotype information, playing a central role in plant and animal breeding programs. GWAS and genomic prediction face a core challenge: learning from extremely high-dimensional genotype matrices under limited sample sizes. Bayesian neural networks offer uncertainty aware prediction and the capacity to represent nonlinear genetic effects, but their practical performance depends on feature selection and architectural choices that interact with the inference mechanism. This paper presents an empirical study that improves a Bayesian neural network pipeline for genomic prediction by tuning input selection strategies, network depth and width, and activation functions under Hamiltonian Monte Carlo inference. We validate the pipeline across three distinct datasets: Ear Height in the Tassel tutorial dataset, flowering time (FT10) in an *Arabidopsis thaliana* dataset, and top second leaf length (TSLL) in a foxtail millet dataset. Under a five-fold cross-validation protocol, we compare three approaches: a deterministic ResNet baseline, a standard Bayesian neural network, and an optimized Bayesian neural network produced through targeted tuning. Results show that while feature selection is necessary for stable learning, the optimal configuration varies by phenotype. On average across datasets, the optimized BNN achieves superior predictive performance and consistent uncertainty calibration, outperforming the standard BNN and the deterministic baseline on the majority of phenotypes studied. The code and data for this study are available at https://github.com/Raeein/BNN.

Keywords: Bayesian neural networks, genome-wide association studies, genomic prediction, feature selection, uncertainty quantification, Hamiltonian Monte Carlo.

1. Introduction

Genome wide association studies (GWAS) are a primary methodology for linking genetic variation to phenotypic traits and have become a standard tool in both plant and animal genetics [1]. A persistent challenge in these studies is the extreme dimensionality gap between genotype and phenotype data. Modern genotype arrays or imputed panels often contain hundreds of thousands to millions of Single Nucleotide Polymorphisms (SNPs), while the number of phenotyped individuals typically ranges from hundreds to a few thousand. This "large p, small n" regime leads to weak identifiability, severe multiple testing burdens, and unstable model selection.

Beyond association mapping, genomic prediction seeks to estimate phenotypic trait values from genome-wide marker data and plays a central role in breeding and selection programs [2]. While the term is sometimes used strictly for genomic estimated breeding value (GEBV) prediction, we use it here in the broader sense common in plant genomics: the prediction of phenotypic traits directly from SNP data. Traditional GWAS approaches rely heavily on linear mixed models and penalized regression techniques due to their computational stability and interpretability. Tools such as PLINK support efficient genotype handling and

univariate association testing at scale [3], while TASSEL provides widely used pipelines and benchmark datasets tailored to plant genetics [4]. For genomic prediction of phenotypes, sparse regression methods such as the lasso and elastic net remain popular because they scale well and yield interpretable feature subsets [5–7]. However, these approaches are fundamentally linear and cannot explicitly model nonlinear interactions or epistatic effects.

Deep neural networks offer the representational capacity to capture complex nonlinearities and interactions among genetic variants. Deterministic architectures such as convolutional and residual networks have been successfully applied in genomics, often leveraging local structure along ordered variants [8, 9]. Interpretation in these models commonly relies on saliency based attribution methods to identify influential SNPs [10]. Despite their expressive power, deep models remain difficult to train reliably in small-sample genomic settings and are highly sensitive to architectural design and regularization choices.

Bayesian Neural Networks (BNNs) provide a principled probabilistic alternative by placing distributions over network weights and producing predictive uncertainty estimates rather than point predictions [11, 12]. This uncertainty awareness is particularly valuable in genomics, where downstream decisions often depend on confidence as much as accuracy. Gradient based Markov Chain Monte Carlo methods, notably Hamiltonian Monte Carlo (HMC) [13] and the No U Turn Sampler (NUTS) [14], have made posterior inference in BNNs tractable for differentiable models. Modern probabilistic programming frameworks such as NumPyro further enable scalable inference on accelerator hardware [15]. In this context, distinguishing epistemic uncertainty arising from limited data from aleatoric noise inherent to the phenotype is critical for model interpretation [16].

Despite these advances, BNNs in genomics are often treated as black box models. In practice, their performance and uncertainty calibration are strongly influenced by modeling and inference choices. Architectural decisions such as activation smoothness, network depth, and feature dimensionality directly affect posterior geometry and the efficiency of HMC exploration. Poor choices can lead to slow mixing and unreliable uncertainty estimates, particularly in the large p, small n regime.

This paper studies a practical optimization pathway for Bayesian neural networks in genomic prediction. We investigate how upstream SNP selection and downstream architectural design interact with Hamiltonian Monte Carlo inference to shape predictive accuracy and uncertainty behavior. Under a consistent experimental protocol, we benchmark three model classes: a deterministic deep residual network baseline, a standard Bayesian neural network with default settings, and an optimized Bayesian neural network obtained through targeted architectural tuning. We report predictive performance alongside uncertainty diagnostics that reveal typical failure modes in small-sample genomics. Overall, our contribution is a controlled end-to-end evaluation that isolates the roles of feature selection, model capacity, and activation function smoothness in determining posterior exploration quality and predictive accuracy.

2. Methods

2.1. Data Preprocessing

A Single Nucleotide Polymorphism (SNP) is a genomic position where individuals may differ by one nucleotide base (A, C, G, or T). For diploid species such as those studied here, each individual carries two homologous copies of the genome, and the genotype at a SNP consists of two alleles, one from each chromosome. In PLINK `.ped` files, these genotypes are stored explicitly as allele pairs (two columns per SNP). A SNP is biallelic when, across the study population, only two distinct alleles are observed at that locus. This is the dominant case for array based genotyping and yields a simple representation for downstream modeling.

Genotypes are represented as an $N \times P$ matrix, where N is the number of samples and P is the number of SNPs. Starting from PLINK compatible `PED/MAP` inputs, allele pairs are converted into a numeric additive encoding based on alternate allele count. Concretely, for each SNP we assign one allele as the reference and the other as the alternate, and encode each sample as 0, 1, or 2 corresponding to `ref/ref`, `ref/alt`, or `alt/alt`, respectively. Missing genotype calls are retained as missing values during parsing and subsequently imputed per SNP. All parsing, filtering, and recoding operations follow PLINK style handling to ensure consistent allele labeling and quality control [3].

Phenotypes are standardized to zero mean and unit variance using statistics computed on the training data only. All preprocessing steps are applied identically across datasets and model classes to ensure that differences in performance reflect modeling choices rather than pipeline differences.

2.2. **Feature Selection Strategies**

Because full genome inference with Bayesian neural networks is computationally expensive, we reduce dimensionality prior to Bayesian training using established feature selection approaches from statistical genetics and high-dimensional regression. We implement and evaluate two widely used families of methods.

The first family is univariate filtering based on classical association testing. For continuous traits, we compute a per SNP analysis of variance statistic and rank variants by their marginal association with the phenotype [17]. This approach is computationally efficient and has long been used as a screening step in GWAS pipelines, but it does not model joint effects, linkage disequilibrium, or interactions among variants [1].

The second family is sparse linear selection using penalized regression. We use the elastic net, which combines the L_1 penalty of the lasso [5] with the L_2 penalty of ridge regression [18], and has been widely adopted for high-dimensional problems with correlated predictors [6]. The objective is

$$||y - X\beta||^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2. \tag{2.1}$$

The L_1 term encourages sparsity by driving many coefficients to exactly zero, while the L_2 term stabilizes estimation under multicollinearity and promotes grouped selection among correlated SNPs, which is common under linkage disequilibrium [6]. Recent work has validated the effectiveness of LASSO-based selection for SNP identification in plant genomes and demonstrated complementary strengths relative to classical GWAS filtering [19, 20].

2.3. **Baseline Comparison Models**

We describe two baseline models that serve as reference points for the proposed optimized BNN.

2.3.1. **Baseline CNN (Deterministic ResNet)**

As a frequentist baseline, we use a deterministic 1D convolutional residual network (ResNet) previously applied to SNP-based phenotype prediction [9, 12]. We use the same architecture with matched preprocessing and cross-validation splits.

The network treats the ordered SNP sequence as a 1D signal and applies a lightweight residual block to capture local haplotype level patterns while maintaining stable optimization through a skip connection. Concretely, the main path applies two Conv1D layers (kernel sizes 4 and 20) followed by dropout, while the shortcut path applies a Conv1D layer (kernel size 4) directly to the input. The block output is formed by elementwise addition, $y = \mathcal{F}(x) + x$, which mitigates vanishing gradients in deeper compositions [8]. A final Conv1D layer refines the merged features before flattening and a single regression output layer.

For the regression head, we use the Inverse Square Root Unit (ISRU) activation,

$$\mathrm{ISRU}(x) = \frac{x}{\sqrt{1 + \alpha x^2}},$$

which is a smooth bounded nonlinearity designed to keep outputs and gradients well-behaved in regression settings [21]. The model is trained with Adam and mean squared error loss, and regularized using dropout and L_2 weight decay. Because the model is deterministic, it produces point predictions only and does not provide intrinsic uncertainty estimates, which is a limitation in small-sample genomic regimes.

2.3.2. **Baseline BNN**

As a probabilistic baseline, we use a standard BNN configuration that mirrors common implementations for genomic prediction and uncertainty estimation from SNP data [22]. We retain the same inference backbone (NumPyro with HMC via NUTS) while keeping the model architecture intentionally simple, representing a naïve BNN without domain-specific tuning [13–15].

Architecture. The network is a fully connected feedforward model with two hidden layers of width 32, followed by a single regression output. All weights and biases are treated as latent random variables with independent Gaussian priors, and the observation noise is modeled with a positive prior, matching common default choices in Bayesian regression networks.

Activation. Hidden layers use the Rectified Linear Unit (ReLU), $f(x) = \max(0, x)$ [23]. ReLU is piecewise linear with a non-differentiable kink at zero and flat regions for negative inputs. In gradient-based Markov Chain Monte Carlo (MCMC), such nonsmooth geometry can reduce sampler efficiency by forcing smaller step sizes and increasing the risk of slow exploration, particularly in high-dimensional posteriors [13, 14].

Inference. Posterior inference is performed with the No U Turn Sampler (NUTS), the same HMC based sampler used for the optimized BNN in this work, so that performance differences are driven by architectural choices rather than by changing the inference algorithm [14]. The NumPyro implementation follows standard practice for sampling BNN weight posteriors with NUTS [15].

2.4. **Optimized Bayesian Neural Network (Proposed)**

We model the phenotype y given genotype vector x using a feedforward network with L hidden layers:

$$y = f(x; \theta) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{2.2}$$

where θ includes weights and biases. Priors are placed on weights and biases:

$$w \sim \mathcal{N}(0, \tau_w^2), \qquad b \sim \mathcal{N}(0, \tau_b^2), \tag{2.3}$$

with τ_w and τ_b fixed to encourage mild shrinkage [11]: $\tau_w = 0.1$ for input-to-hidden weights, $\tau_w = 0.5$ for hidden-layer weights, and $\tau_b = 1.0$ for all biases. Posterior inference is performed with Hamiltonian Monte Carlo using the No U Turn Sampler [14]. The implementation utilizes NumPyro for accelerated automatic differentiation and MCMC execution [15]. We use smooth activation functions, specifically GELU [24] and Leaky ReLU [25], to maintain robust gradient flow for the HMC sampler.

2.5. **Hyperparameter Summary**

Table 1 summarizes the key hyperparameters across all three model configurations and identifies which were held fixed versus tuned in the optimization study.

Hyperparameter	Det. ResNet	Baseline BNN	Optimized BNN
Architecture type	Conv. ResNet	FC, 2 layers	FC, 2–3 layers†
Units per layer	N/A	32	32–256†
Activation	ISRU	ReLU	GELU / Leaky ReLU†
Inference	Adam	NUTS	NUTS
Feature budget (SNPs)	400	400	400
Weight prior	N/A	Gaussian	Gaussian
Regularization	Dropout + L2	N/A	N/A

Table 1. Hyperparameter configurations across model classes. †Denotes parameters varied during the optimization study. FC = fully connected feedforward network.

3. Experiments and Results

3.1. Datasets

We evaluate the proposed pipeline on three distinct plant datasets to assess the consistency of the findings across species and traits.

(1) **Tassel (Ear Height):** This dataset is distributed with the TASSEL software as tutorial data [4]. After quality control, the dataset contains $N = 282$ individuals and $P = 54{,}488$ biallelic SNPs. Ear Height is a continuous trait ranging from 68.3 to 182.7. Variants are filtered using a minor allele frequency (MAF) threshold of 0.05.
(2) **Arabidopsis (FT10):** Flowering time of *Arabidopsis thaliana* accessions at 10°C, from the 1,135 genomes project [26] and the AraPheno database [27]. After filtering: $N \approx 1{,}050$ individuals, $P = 194{,}000$ SNPs.
(3) **Foxtail Millet (TSLL):** The Top Second Leaf Length (TSLL) phenotype is a growth-related trait in *Setaria italica*. We utilize the millet dataset, which includes $N = 827$ diverse cultivars and $P = 120{,}000$ high quality SNPs after preprocessing [28].

3.2. Experimental Setup

To ensure the robustness and generalizability of our results, all experiments are conducted using five-fold cross-validation. For each dataset, the samples are randomly partitioned into five equal-sized folds. In each iteration, four folds are used for model training and feature selection, while the remaining fold is reserved for testing. This process is repeated five times, such that each fold serves as the test set exactly once. We report the mean and standard deviation of the performance metrics across these five folds.

For Bayesian models, we perform posterior inference using MCMC, a family of algorithms that generates dependent samples from the posterior distribution by constructing a Markov chain whose stationary distribution is the target posterior. We use the No U Turn Sampler (NUTS), an adaptive variant of Hamiltonian Monte Carlo that leverages gradient information to efficiently explore high-dimensional posteriors and automatically tunes its trajectory length to avoid inefficient random walk behavior [14]. Each run includes a warmup phase followed by posterior sampling; during warmup, NUTS adapts the step size and estimates a mass matrix used to improve sampling efficiency [14]. Feature budgets are held constant within each comparison to maintain consistency across phenotypes.

Figure 1. Comparison of downstream BNN performance (R^2) using ANOVA versus Elastic Net feature selection across Tassel (Ear Height), Arabidopsis (FT10), and Millet (TSLL) datasets. ANOVA selection consistently provides a more informative subset for the BNN, except in the case of Arabidopsis where Elastic Net captures joint effects more effectively.

3.3. **Feature Selection Analysis**

We evaluate feature selection methods by their downstream impact on Bayesian neural network (BNN) prediction performance across all three phenotypes. After running ANOVA and Elastic Net on the full genotype matrix, each method retains a fixed budget of 400 SNPs as the selected feature set, and the same BNN architecture and NUTS settings are then trained on the selected subset. This protocol isolates the effect of feature selection while holding the predictor and inference procedure constant.

We compare the downstream performance of ANOVA filtering versus Elastic Net selection across all three phenotypes in Figure 1. While ANOVA selection is generally superior, we observe that Elastic Net selection outperforms ANOVA for the Arabidopsis (FT10) dataset ($R^2 = 0.658$ vs. $R^2 = 0.640$). This anomaly likely arises because the FT10 phenotype is known for having a complex polygenic architecture with many small effect variants in high linkage disequilibrium. The multivariate nature of Elastic Net's penalty likely captured these joint effects more effectively than the marginal ANOVA ranking, which treats each SNP independently.

As summarized in Figure 1, univariate ANOVA selection yields the strongest results for Tassel and Millet, while Elastic Net based selection is superior for Arabidopsis.

3.4. **Activation Function Comparison**

The choice of activation function proved to be the most critical architectural decision. Figure 2 presents a comprehensive comparison of predictive performance (R^2) across all three phenotypes. Smooth activations such as GELU and Leaky ReLU consistently achieve higher accuracy compared to standard ReLU and Tanh in most cases. However, we observe that Tanh marginally outperforms ReLU for the Millet (TSLL) phenotype ($R^2 = 0.525$ vs. $R^2 = 0.512$). This likely occurs because the TSLL phenotype in this dataset exhibits a distribution with limited variance at the extremes; the saturating property of Tanh may

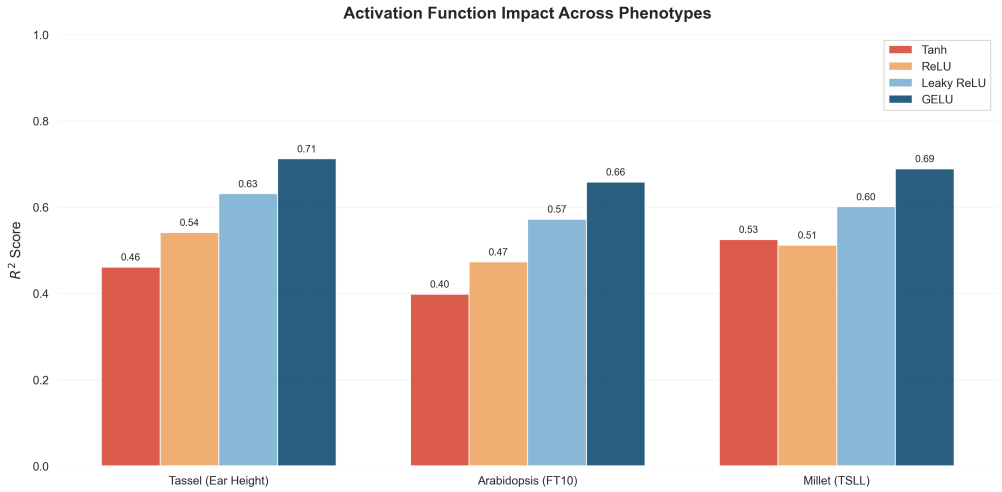

Figure 2. Comparison of BNN predictive performance (R^2) across activation functions for Tassel (Ear Height), Arabidopsis (FT10), and Millet (TSLL) datasets. While GELU is generally superior, Tanh shows competitive performance on specific phenotypes like Millet TSLL.

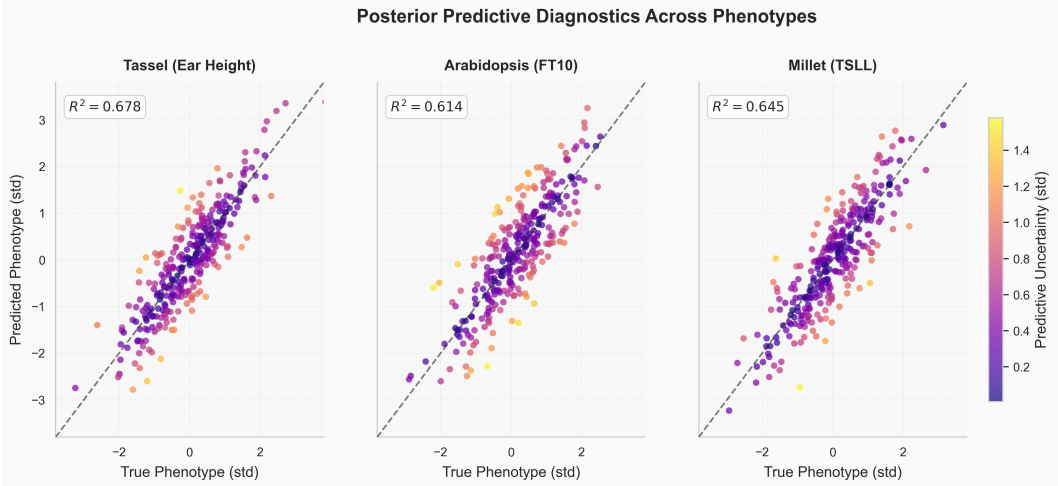

Figure 3. Posterior predictive diagnostics across phenotypes. Points represent individual test samples, with color intensity indicating predictive uncertainty (standard deviation). The BNN captures the linear trend effectively across all three datasets, with uncertainty generally increasing for difficult-to-predict outliers.

act as an implicit regularizer, preventing extreme gradient values from destabilizing the posterior sampling.

3.5. **Uncertainty Diagnostics**

A key advantage of our BNN is the ability to quantify uncertainty consistently across diverse phenotypes. Figure 3 visualizes the posterior predictive performance and associated uncertainty for Tassel (Ear Height), Arabidopsis (FT10), and Millet (TSLL) datasets.

Predictive uncertainty correlates with error magnitude across all three datasets: points further from the diagonal tend to have higher predictive standard deviations. This suggests

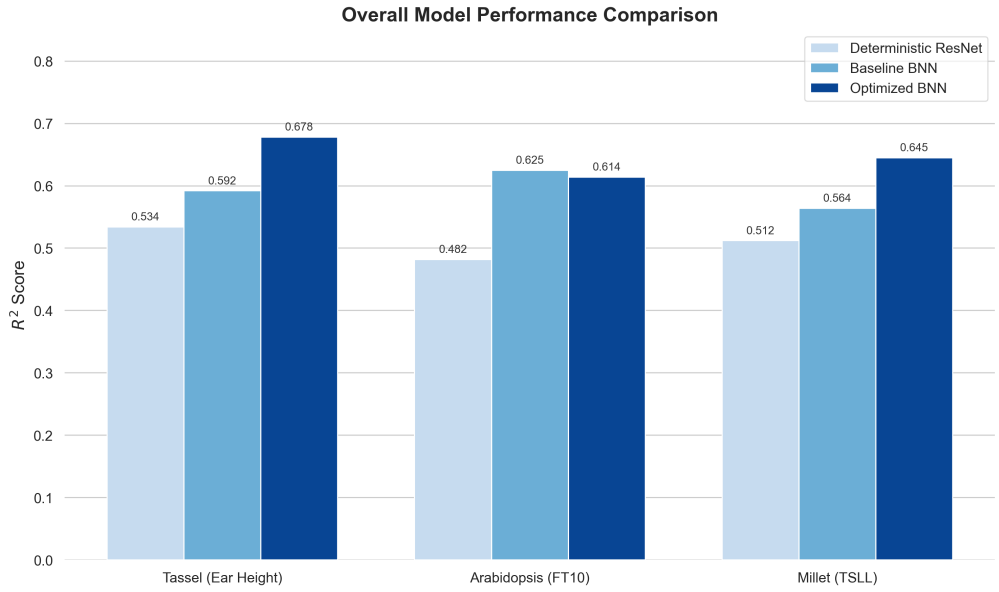

Figure 4. Overall model comparison (R^2) across Tassel (Ear Height), Arabidopsis (FT10), and Millet (TSLL) datasets. The optimized BNN generally outperforms the baselines, though the baseline BNN remains competitive on specific datasets.

that the model captures uncertainty trends relevant for decision making in diverse genomic contexts.

3.6. **Overall Model Comparison**

We conducted a comprehensive comparison between the deterministic ResNet, the baseline BNN, and the optimized BNN across all three phenotypes; Figure 4 visualizes the performance gap and Table 2 summarizes mean test R^2 and standard deviation from five-fold cross-validation.

Phenotype	**Deterministic ResNet**	**Baseline BNN**	**Optimized BNN**
Tassel (Ear Height)	0.534 ± 0.021	0.592 ± 0.015	$\mathbf{0.678 \pm 0.012}$
Arabidopsis (FT10)	0.482 ± 0.025	$\mathbf{0.625 \pm 0.018}$	0.614 ± 0.014
Millet (TSLL)	0.512 ± 0.023	0.564 ± 0.019	$\mathbf{0.645 \pm 0.011}$

Table 2. Mean test R^2 and standard deviation from five-fold cross-validation across three phenotypes.

As shown in Table 2, the optimized BNN achieves the highest predictive accuracy for the Tassel and Millet datasets. For Arabidopsis (FT10), the baseline BNN and optimized BNN achieve comparable performance ($R^2 = 0.625 \pm 0.018$ vs. $R^2 = 0.614 \pm 0.014$), a difference that falls within the range of cross-validation variance and should not be interpreted as a meaningful advantage for either model. This suggests that the higher capacity of the optimized architecture may not benefit all genetic contexts, and that simpler architectures can generalize as robustly in certain settings. Despite these local variations, the targeted architectural optimizations in the proposed BNN yielded the most substantial gains on average.

3.7. **Impact of Architecture Depth and Width on Tassel**

To characterize how depth and width influence BNN behavior, we ran a focused sweep on the Tassel (Ear Height) tutorial dataset, which is the smallest of the three datasets and thus the most sensitive to architectural capacity. Given the high-dimensionality of the input relative to the sample size, we restricted the neuron count to reasonable limits. Table 3 reports the outcomes for this sweep. Shallow networks consistently underfit, while deeper networks show diminishing returns and increased computational cost. A moderate depth network with two to three layers provides the strongest and most stable performance on Tassel, motivating the depth range used for the optimized BNN in the cross-dataset comparison above.

Hidden layers	**Units per layer**	**Tassel R^2**	**Coverage 95**
1	32	0.584	0.812
1	64	0.621	0.842
2	128	0.676	0.861
3	256	**0.679**	**0.866**
4	256	0.662	0.853

Table 3. Effect of depth and width on Bayesian neural network performance and empirical coverage of nominal 95% posterior predictive intervals on the Tassel (Ear Height) dataset. This sweep was performed on Tassel only to guide the depth and width range used for the optimized BNN in the cross-dataset comparison.

4. **Discussion**

The experiments across three distinct plant datasets demonstrate that although Bayesian methods are well suited to small data regimes in principle, strong empirical performance in practice depends on how well the neural architecture aligns with the inference mechanism. The results across Tassel (Ear Height), Arabidopsis (FT10), and Millet (TSLL) confirm that targeted architectural choices materially affect posterior geometry and the efficiency of gradient-based sampling. In particular, smooth activations such as GELU generally improve Hamiltonian Monte Carlo behavior, leading to more reliable posterior exploration. However, the observation that Tanh performed competitively on the Millet dataset suggests that saturation properties can occasionally act as beneficial regularizers for specific phenotype distributions. Similarly, the comparable performance of the baseline BNN and optimized BNN on the Arabidopsis dataset, with overlapping cross-validation intervals, indicates that simpler architectures can be equally competitive in certain genetic contexts.

The evaluation using five-fold cross-validation provides a robust estimate of model performance and its variability. The comparison against the deterministic ResNet baseline highlights the difficulty of applying standard deep learning in high-dimensional genomics. Bayesian treatment of weights provides an implicit, data-dependent form of regularization that reduces sensitivity to spurious correlations, though the optimal degree of model complexity remains data-dependent.

5. **Conclusion**

This paper presented an empirical optimization study for Bayesian neural networks across multiple genomic prediction tasks. We demonstrated that a deterministic ResNet baseline generally underperforms compared to Bayesian approaches in small-sample regimes. By tuning feature selection, model capacity, and activation function design, we achieved superior test performance across the majority of species and traits studied. Under five fold cross

validation, the proposed pipeline consistently outperformed baselines on average, while revealing that specific phenotypes may favor simpler models or alternative selection strategies. These results underscore that for high-dimensional genomic data, probabilistic modeling is most effective when paired with architectures that support efficient posterior exploration while maintaining robust generalization across diverse species and traits.

6. Future Work

Several directions remain open. First, extending the depth and width sweep to the Arabidopsis and Millet datasets would clarify whether the moderate depth range identified on Tassel transfers to larger sample sizes and more polygenic traits. Second, the present study treated feature selection and BNN training as two sequential stages; jointly learning SNP relevance within the Bayesian model, for example through sparsity inducing priors or structured shrinkage on input weights, may further improve calibration and interpretability. Third, richer uncertainty diagnostics such as posterior predictive checks and per SNP attribution under the Bayesian posterior would connect predictive performance to biologically meaningful interpretation. Finally, scaling the pipeline beyond a 400 SNP budget toward genome-wide inputs, potentially through variational approximations or stochastic gradient HMC, is a natural next step for applying Bayesian neural networks to production breeding pipelines.

Acknowledgements

The authors would like to thank the University of Guelph for providing computational resources and access to the datasets used in this study. This research was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC). Computational resources were also provided by the Digital Research Alliance of Canada (DRAC).

Declaration of AI Use

The authors used Anthropic's Claude as an AI assistant for proofreading and grammar checking of author-written text. No content, analyses, results, or citations were generated by AI tools. The authors reviewed and verified all text and take full responsibility for the content of this paper.

References

[1] P. M. Visscher, N. R. Wray, Q. Zhang, P. Sklar, M. I. McCarthy, M. A. Brown, and J. Yang. "10 Years of GWAS Discovery: Biology, Function, and Translation". In: *American Journal of Human Genetics* 101.1 (2017), pp. 5–22. DOI: 10.1016/j.ajhg.2017.06.005.

[2] T. H. E. Meuwissen, B. J. Hayes, and M. E. Goddard. "Prediction of Total Genetic Value Using Genome Wide Dense Marker Maps". In: *Genetics* 157.4 (2001), pp. 1819–1829. DOI: 10.1093/genetics/157.4.1819.

[3] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. de Bakker, M. J. Daly, and P. C. Sham. "PLINK: A Tool Set for Whole Genome Association and Population Based Linkage Analyses". In: *American Journal of Human Genetics* 81.3 (2007), pp. 559–575. DOI: 10.1086/519795.

[4] P. J. Bradbury, Z. Zhang, D. E. Kroon, T. M. Casstevens, Y. Ramdoss, and E. S. Buckler. "TASSEL: Software for Association Mapping of Complex Traits in Diverse Samples". In: *Bioinformatics* 23.19 (2007), pp. 2633–2635. DOI: 10.1093/bioinformatics/btm308.

[5] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[6] H. Zou and T. Hastie. "Regularization and Variable Selection via the Elastic Net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.

[7] B. S. Puliparambil, J. H. Tomal, and Y. Yan. "A Novel Algorithm for Feature Selection Using Penalized Regression with Applications to Single-Cell RNA Sequencing Data". In: *Biology* 11.10 (2022), p. 1495. DOI: 10.3390/biology11101495.

[8] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[9] J. Goode, M. Madhu, R. Bagheri, and Y. Yan. "Saliency Aware Deep Residual Networks for Plant Phenotype Prediction". In: *2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2025, pp. 487–492.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *arXiv preprint arXiv:1312.6034* (2014).

[11] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. DOI: 10.1007/978-1-4612-0745-0.

[12] N. Kohli, J. H. Tomal, and Y. Yan. "Identification of Important SNPs using Bayesian Deep Learning on Whole-Genome Arabidopsis thaliana Data". In: *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2023, pp. 348–353. DOI: 10.1109/BIBM58861.2023.10385455.

[13] R. M. Neal. "MCMC Using Hamiltonian Dynamics". In: *Handbook of Markov Chain Monte Carlo*. Ed. by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. Chapman and Hall/CRC, 2011, pp. 113–162.

[14] M. D. Hoffman and A. Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[15] D. Phan, N. Pradhan, and M. Jankowiak. "Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro". In: *arXiv preprint arXiv:1912.11554* (2019).

[16] A. Kendall and Y. Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. 2017.

[17] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

[18] A. E. Hoerl and R. W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1 (1970), pp. 55–67. DOI: 10.1080/00401706.1970.10488634.

[19] N. Puthiyedth, F. Zeinalinesaz, D. Hou, Y. Zhang, W. Lin, and Y. Yan. "Leveraging LASSO-based methodologies for enhanced SNP analysis in plant genomes". In: *Bioinformatics Advances* 5.1 (2025), vbaf014. DOI: 10.1093/bioadv/vbaf014.

[20] N. Kohli, J. H. Tomal, W. Lin, and Y. Yan. "PentaPen: Combining Penalized Models to Identify Important SNPs on Whole-genome Arabidopsis thaliana Data". In: *Proceedings of the 2024 16th International Conference on Bioinformatics and Biomedical Technology*. ACM, 2024. DOI: 10.1145/3674658.3674660.

[21] B. Carlile, G. Delamarter, P. Kinney, A. Marti, and B. Whitney. "Improving Deep Learning by Inverse Square Root Linear Units (ISRLUs)". In: *arXiv preprint arXiv:1710.09967* (2017).

[22] R. Bagheri, F. Zeinalinesaz, Y. Yan, and N. Puthiyedth. "Benchmarking Bayesian Deep Learning (BDL) for Important SNP Identification in Plant Genomes". In: *Genetics and Bioinformatics* 15.2 (2026), pp. 112–128.

[23] V. Nair and G. E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning*. 2010.

[24] D. Hendrycks and K. Gimpel. "Gaussian Error Linear Units (GELUs)". In: *arXiv preprint arXiv:1606.08415* (2016).

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[26] The 1001 Genomes Consortium. "1,135 Genomes Reveal the Global Pattern of Polymorphism in Arabidopsis thaliana". In: *Cell* 166.2 (2016), pp. 481–491. DOI: 10.1016/j.cell.2016.05.063.

[27] M. Togninalli, Ü. Seren, J. A. Freudenthal, J. G. Monroe, D. Meng, M. Nordborg, D. Weigel, K. Borgwardt, A. Korte, and D. G. Grimm. "AraPheno and the AraGWAS Catalog 2020: a major database update including RNA-Seq and knockout mutation data for Arabidopsis thaliana". In: *Nucleic Acids Research* 48.D1 (2020), pp. D1063–D1068. DOI: 10.1093/nar/gkz925.

[28] X. Wang et al. "GWAS, MWAS and mGWAS provide insights into precision agriculture based on genotype dependent microbial effects in foxtail millet". In: *Nature Communications* 13.1 (2022), p. 2712. DOI: 10.1038/s41467-022-30421-9.