GPFN: Prior-Data Fitted Networks for Genomic Prediction

Jordan Ubbens[1,✉], **Ian Stavness**[1,2], **and Andrew G. Sharpe**[1]

[1]Global Institute for Food Security (GIFS), University of Saskatchewan, Saskatoon, SK S7N 0W9, Canada
[2]Department of Computer Science, University of Saskatchewan, Saskatoon, SK S7N 0W9, Canada

Genomic Prediction (GP) methods predict the breeding value of unphenotyped individuals in order to select parental candidates in breeding populations. Among models for GP, classical linear models have remained consistently popular, while more complex nonlinear methods such as deep neural networks have shown comparable accuracy at best. In this work we propose the Genomic Prior-Data Fitted Network (GPFN), a new paradigm for GP. GPFNs perform amortized Bayesian inference by drawing hundreds of thousands or millions of synthetic breeding populations during the prior fitting phase. This allows GPFNs to be deployed without requiring any training or tuning, providing predictions in a single inference pass. On three populations of crop plants across two different crop species, GPFNs perform significantly better than the linear baseline on 13 out of 16 traits. On a challenging between-families NAM prediction task, the GPFN performs significantly better in 3 locations while only falling behind in one. GPFNs represent a completely new direction for the field of genomic prediction, and have the potential to unlock levels of selection accuracy not possible with existing methods.

genomic prediction | predictive breeding | machine learning

Correspondence: ***jordan.ubbens@gifs.ca***

Introduction

With the goal of supporting a growing global population, increasing the rate of genetic gain in plant and animal breeding is important. In modern breeding, Genomic Selection (GS) has become one point of focus for improving genetic gain in quantitative traits (Meuwissen et al., 2001). GS aims to select the most fit parents by performing Genomic Prediction (GP) – regressing phenotypic values using genome-wide marker data, most often Single-Nucleotide Polymorphisms (SNPs). The efficacy of GS has been validated in several plant and animal species, accelerating the rate of genetic gain in many cases (Goddard and Hayes, 2007; Goddard et al., 2009; Crossa et al., 2017). With genotyping becoming increasingly accessible through technological improvement and cost reduction, GS has become relevant for more and more breeding programs in recent years.

Despite continual progress in genotyping, the prediction of Genomic Estimated Breeding Values (GEBVs) is still most often performed using classical techniques such as BayesB (Meuwissen et al., 2001), and GBLUP (VanRaden, 2008). Much recent work has focused on the discovery of new methods based on deep learning (Ma et al., 2018; Wang et al., 2023; Gao et al., 2023) – however, classical methods remain the most popular because they empirically perform at least as well as newer, more modern methods while being simpler, faster, and requiring less tuning (Azodi et al., 2019; Abdollahi-Arpanahi et al., 2020; Zingaretti et al., 2020; Ubbens et al., 2021; John et al., 2022; Ray et al., 2023).

In this work, we develop a new approach to GP based on amortized Bayesian inference, which we term *Genomic Prior-Data Fitted Networks* (GPFNs). Unlike existing methods based on neural networks, a GPFN is not trained on the end user's dataset, does not require any tuning, and in fact is never exposed to any real data at all prior to inference. As a proof of concept, we show that the GPFN is the first method which is able to consistently and significantly outperform classical methods in several datasets of crop plants. An implementation as well as several trained GPFNs for various population types are provided at `https://github.com/jubbens/gpfn`. We propose the GPFN approach as a major new direction for genomic prediction.

Background

In all cases in the current literature, GP is carried out in a supervised fashion, where a model is trained using the training population and then that model is used to perform inference on the target (testing, evaluation) population. The learning behavior of any particular estimator is influenced by the model's priors, such as inductive biases expressed through model architecture or regularization schemes. These priors are usually especially important when it comes to high-capacity models such as deep neural networks operating in low-data regimes. The alternative to these types of implicit priors is Bayesian inference, where the prior belief over the data is defined explicitly. This is straightforward if the data generating process can be modelled as a tractable distribution, such as a mixture of Gaussians, but for more complex distributions (such as those found in quantitative genetics) it is difficult to estimate the Posterior Predictive Distribution (PPD).

One way to overcome this limitation is with *Prior-Data Fitted Networks* (PFNs), which allow a neural network to approximate the PPD given any prior from which one can sample (Müller et al., 2021). The prior can be intractable, such as a complex, black-box process, and the PFN is still able to perform approximate Bayesian inference. Practically, the process of training the PFN involves repeatedly sampling datasets $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^{n}$, as input-label pairs, from the prior distribution over datasets $p(\mathcal{D})$. During training, the n datapoints are used for training along with *query points* x

for which we wish to estimate the PPD $p(y|x, \mathcal{D})$. The PPD is given by

$$p(y|x, \mathcal{D}) = \int_t p(y|x, t)p(t|\mathcal{D}) \quad (1)$$

where $p(t)$ is a prior for a latent variable, called the task (Müller et al., 2021). One can think of the task as a particular internal state of the data-generating process.

PFNs model the PPD, without ever instantiating the posterior, parameterized by a neural network. In particular, PFNs use the transformer architecture as it is well-suited to modeling relationships between the set-ordered datapoints drawn from the prior (Vaswani et al., 2017). During training, datasets $\mathcal{D}$ are sampled from the prior $p(\mathcal{D})$ and model parameters θ are optimized by minimizing

$$\ell_\theta = \mathbb{E}_{D \cup \{x,y\} \sim p(\mathcal{D})}[-\log q_\theta(y|x, D)] \quad (2)$$

via gradient descent. It is shown that optimizing Equation 2 is equivalent to minimizing the cross-entropy between the PPD and its approximation (Müller et al., 2021).

Approach

PFNs for Genomic Prediction. Here we propose GPFN, a category of PFNs which are trained to perform Genomic Prediction in unseen populations. Because PFNs perform amortized inference, they can be used to perform inference without re-training. Consequently, there is no need to tune the training process, such as performing cumbersome hyperparameter searches. The key difference between the classical supervised learning approach typically used in GP and the Bayesian inference performed by PFNs is that beliefs about factors such as population structure and recombination are expressed implicitly via the prior.

GPFNs are trained on priors which are defined by different types of simulated populations. As a convention, we use the term *unstructured population* to refer to a population which is not the result of a defined crossing scheme, such as diversity panels, and *structured population* to refer to those which are, such as mapping panels and recurrent breeding programs. Because different populations vary in a variety of ways, a prior must be created specifically for each type of population. When creating the prior, the samples must be split with the same training and testing split as will be seen during inference. For example, if one were to train a GPFN for use in a breeding program where the F4s from the previous recurrent block are used as a training population to perform selection in the F2s in the subsequent recurrent block, the prior should ideally be designed to simulate this exact procedure. In the next section we discuss two priors built for demonstrating GPFNs: one for unstructured populations with random train/test splits (which we will demonstrate in three unstructured populations of wheat and lentil) and one for structured NAM populations with a between-families train/test split (which we will demonstrate in a soybean NAM population).

Priors for Prediction Populations. PFNs have been fit using priors defined by Bayesian neural networks, Gaussian processes, and structural-causal models (Müller et al., 2021; Hollmann et al., 2022). In order to create priors which model plant and animal populations, we use genetic simulations based on SeqBreed (Pérez-Enciso et al., 2020). In the context of GPFNs, a prior is defined by a simulator which simulates a particular type of population (wild, mapping, breeding, etc.) and each draw from the prior provides a completely new population of this type. Each prior has a number of parameters which define characteristics of the quantitative trait architecture, population structure, and other factors. When drawing from the prior, these parameters are sampled from a uniform distribution.

All priors demonstrated here use a single-trait, single-environment assumption. Multi-environment and multi-trait models are left for future work. For the trait of interest, we simulate both additive and dominance effects (with the proportion of dominant alleles randomly sampled). Marker effects are sampled from a gamma distribution (Pérez-Enciso et al., 2020), with the α and β parameters randomly sampled. The trait heritability (h^2), the number of QTN, the number of SNPs, the recombination rate, and other factors are also randomly sampled.

Unstructured Population Prior. To train a GPFN for use with unstructured populations, we use a prior which simulates a wild population. All priors for structured populations also start with founders drawn from a randomly generated unstructured population (see Figure 1). Although this is not a realistic starting point for breeding populations which typically start from elite inbred lines, it is more difficult to generate these types of parents *de novo*. In order to generate the unstructured population, we begin from a set of diploid individuals with a randomly sampled number of biallelic SNPs. The heterozygosity and minor allele frequency are both randomly sampled. For a random number of generations, we proceed with random mating and, with a (randomly sampled) probability, half of the current population migrates to found a new subpopulation. Allele frequency begins to differentiate between the different subpopulations due to reproductive isolation. Finally, extant individuals are randomly selected across the different subpopulations. The random parameters for the unstructured prior are listed in Table S1. Figure S1 shows an example of a population drawn from the unstructured prior compared to a real population of wheat landraces.

NAM Prior. The Nested Association Mapping prior models a standard NAM scheme, where prediction is performed between-families. Of the founders, one is selected at random as a common parent and the others are crossed to it, followed by single seed descent. The total number of families is randomly varied between 25 and 85, while the number of iterations of selfing is randomly varied between 2 and 5. The extant individuals are grouped by family, and we split the families into training and evaluation. Other parameters are identical to those in the unstructured prior (Table S1). Figure S2 shows an example of a population drawn from the

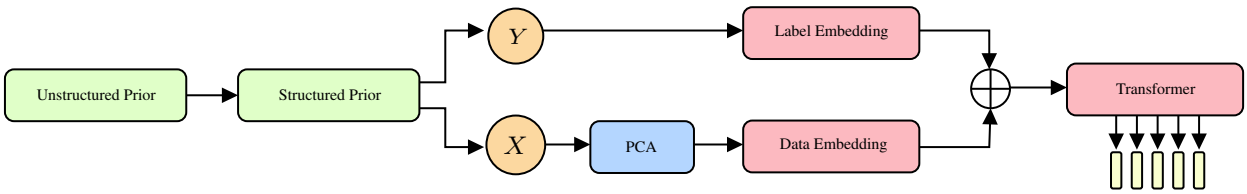

Figure 1. Training a Prior-Data Fitted Network to perform Genomic Prediction. For unstructured populations, the unstructured prior is used directly.

NAM prior compared to a real NAM population of soybean.

Architectural Details. We fit the prior using a transformer model with 12 layers, an internal dimensionality of 2048, a hidden layer size of 2048, and a single attention head. In our experiments, multi-head attention did not yield better results than a single attention head, so a single head is used for faster throughput. In general, wider and deeper transformers performed slightly better when using the same prior. This is despite the fact that the smaller transformers with fewer parameters were usually able to fit the prior to a similar level of accuracy during the fitting phase. This is why we use an architecture which is both wider and deeper than PFNs demonstrated in the literature (Müller et al., 2021).

Prior Fitting. To train a GPFN model we largely follow the training procedure from Müller et al. (2021). During training, the prior is repeatedly sampled to generate new $\mathcal{D}$ which constitute training batches. The GPFNs presented here are fit using one million draws from the prior. Simulation is performed using 336 CPU cores, while fitting is done in parallel on a single V100 GPU. Figure 1 illustrates an overview of the prior fitting process.

The use of transformers as PFNs comes with some practical computational challenges. Since the memory complexity of self-attention scales quadratically with the number of tokens, the number of individuals n per sampled $\mathcal{D}$ is limited. Because the input tokens are set-valued, *i.e.* not ordered, it is not viable to use faster approximations which focus on local attention windows. We limit the population size to 1000 during training, but during inference we use the full training and testing sets. The size of the input vectors x_i needs to be constant despite the different number of SNPs per simulated population. To achieve this, we use the loadings on the first 100 principal components as the input to the embedding layer. The number of query points is varied between 10 and 500 during training, although we use a uniform weighting instead of one biased towards larger training sets as in Müller et al. (2021). We would prefer that the network is frequently exposed to smaller training populations because of the interaction between the number of samples in the training set and the PCA projection step.

PFNs are usually trained using the Prior-Data Negative Log Likelihood loss (Equation 2). However, we find that using a bucketized mean squared error provides slightly better performance on our regression task. This involves computing the mean prediction by weighting the bucket centers by the output probability, instead of only considering the probability attributed to the correct bucket. Although it breaks some theoretical guarantees, we find that this stabilizes training as well as allowing for training under a wider range of hyperparameters, since the gradient is smoother under the bucketized MSE criteria.

Evaluation Strategy. For all results, we compare to GBLUP exclusively, because we consider it to be at least as good as state-of-the-art. It is not our intention to compare multiple classes of models, as this has already been demonstrated extensively in the literature (Azodi et al., 2019; Abdollahi-Arpanahi et al., 2020; Zingaretti et al., 2020; Ubbens et al., 2021; John et al., 2022; Ray et al., 2023). For each experiment, we split the data into 80% training and 20% testing. We measure the Pearson's correlation between predicted and actual breeding value and report the percentage increase/decrease versus the GBLUP baseline. The data is resampled 100 times to obtain error estimates. Resampling so many times is enabled by the fact that GPFNs only perform inference – other methods would not only need to be re-trained each time, but hyperparameter search would also have to be performed independently on each random split of the data to avoid leaking information across training and testing sets. Full results are provided in Supplementary Note S3.

We also show results for Principal Components Regression (PCR) as another baseline. Since the GPFN uses the PCA loadings as input, this helps elucidate how much of the performance gain or loss may be attributable to the projection method. Additionally, PCR shows what level of performance can be attained by simply performing linear regression on these features. This helps to show whether or not the transformer is implementing a prediction strategy which is similar to, or performs similarly to, linear regression.

Results

Wheat Preliminary Yield Trials. We examined performance on a breeding population using a preliminary yield trial (PYT) of wheat from the University of Guelph's Winter Wheat Breeding program (Ficht et al., 2023). The PYT is a single-rep, single-environment trial. In total, this population is comprised of 319 individuals and 4,722 SNPs. We use a GPFN fit with the unstructured prior for this evaluation. Figure 2 shows results for the wheat PYT population.

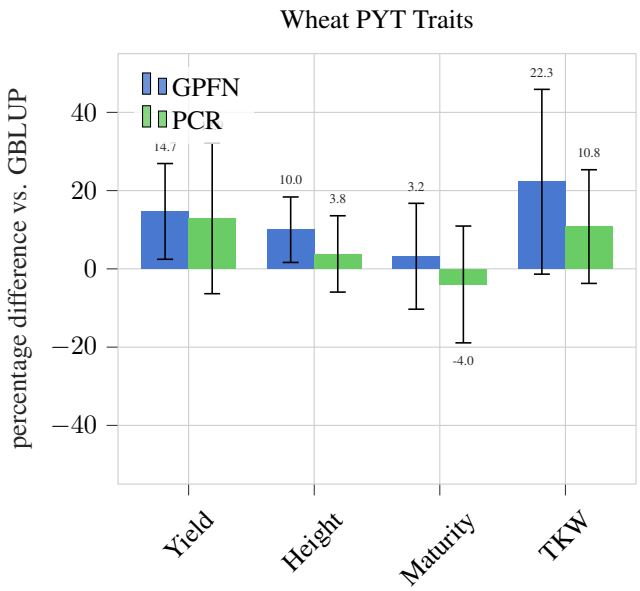

Figure 2. Prediction performance results for the wheat preliminary yield trial. Bars show percent improvement/shortfall in Pearson's r compared to GBLUP. Detailed results are shown in Table S2.

Wheat Landraces. For another unstructured crop population, we use a dataset containing 2,000 landrace accessions of wheat (Crossa et al., 2016). The data consists of 33,709 SNPs. We use a GPFN fit with the unstructured prior for this evaluation. Figure 3 shows results for the wheat landraces population.

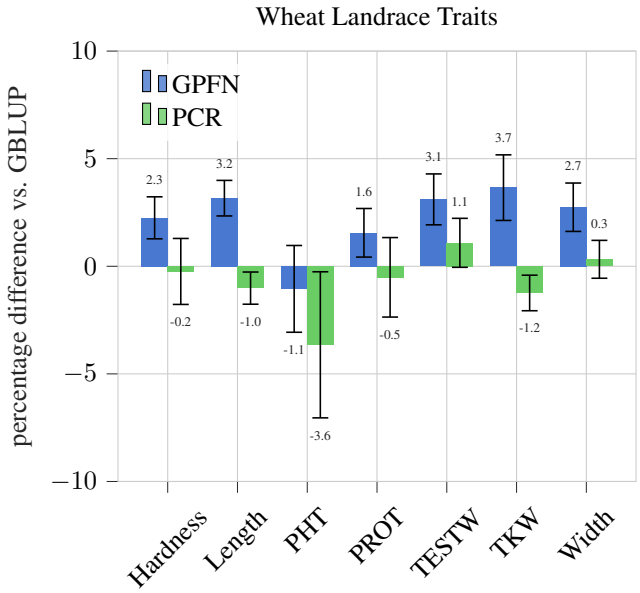

Figure 3. Prediction performance results for the wheat landraces. Bars show percent improvement/shortfall in Pearson's r compared to GBLUP. Detailed results are shown in Table S3.

Lentil Diversity Panel. For the final unstructured crop population, we use a diversity panel of lentil (Haile et al., 2020). There are a total of 324 individuals and 9,394 SNPs. We use a GPFN fit with the unstructured prior for this evaluation. Figure 4 shows performance results for the lentil diversity panel.

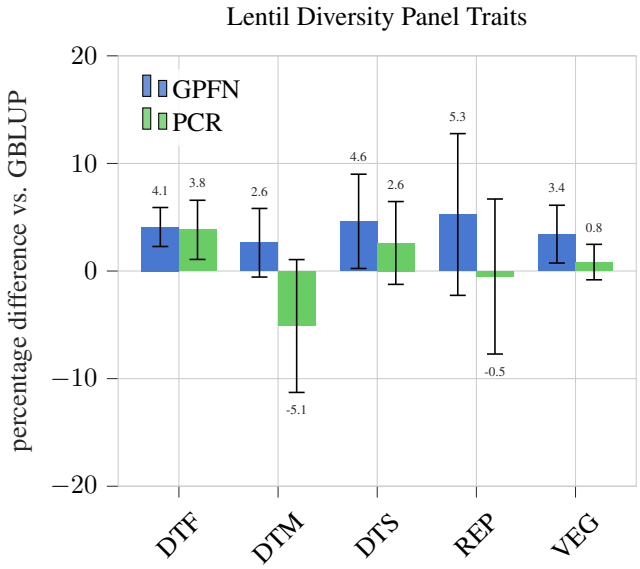

Figure 4. Prediction performance results for the lentil diversity panel. Bars show percent improvement/shortfall in Pearson's r compared to GBLUP. Detailed results are shown in Table S4.

Nested Association Mapping Panel of Soybean. For a structured population, we use a nested association mapping (NAM) panel of soybean, known as SoyNAM (Xavier et al., 2016). Because the published marker data contains a high proportion of missing data, we use a version with imputed genotypes (Chen et al., 2022). For the trait of interest, we predict yield across multiple locations. We use the 2012 year since data is available for all locations for this field season. This population contains a variable number of individuals between locations, from 827 in the MO location to 5019 in NE. The total number of families also varies between locations, with 23 in MI, 25 in MO, and 38 in each of OH, IL, IA, IN, NE, and KS. The imputed genotype data consists of 29,416 SNPS.

This population has been used extensively for genomic prediction in past studies (Xavier and Rainey, 2020; Ray et al., 2023). We use a between-families prediction scheme as this represents one of the most difficult prediction tasks, owing to the difficulty of predicting between families, as well as using a trait which is only moderately heritable and shows low prediction accuracy in general. This is similar to the leave-family-out sampling in Xavier and Rainey (2020), but even more difficult as 20% of the families are held out for the evaluation population. We use a GPFN fit with the NAM prior for this evaluation. Figure 5 shows a summary of prediction results for the NAM population. In this experiment, we report r directly, since GBLUP regularly falls below an r value of zero, making it impossible to calculate percent improvement as with previous tests.

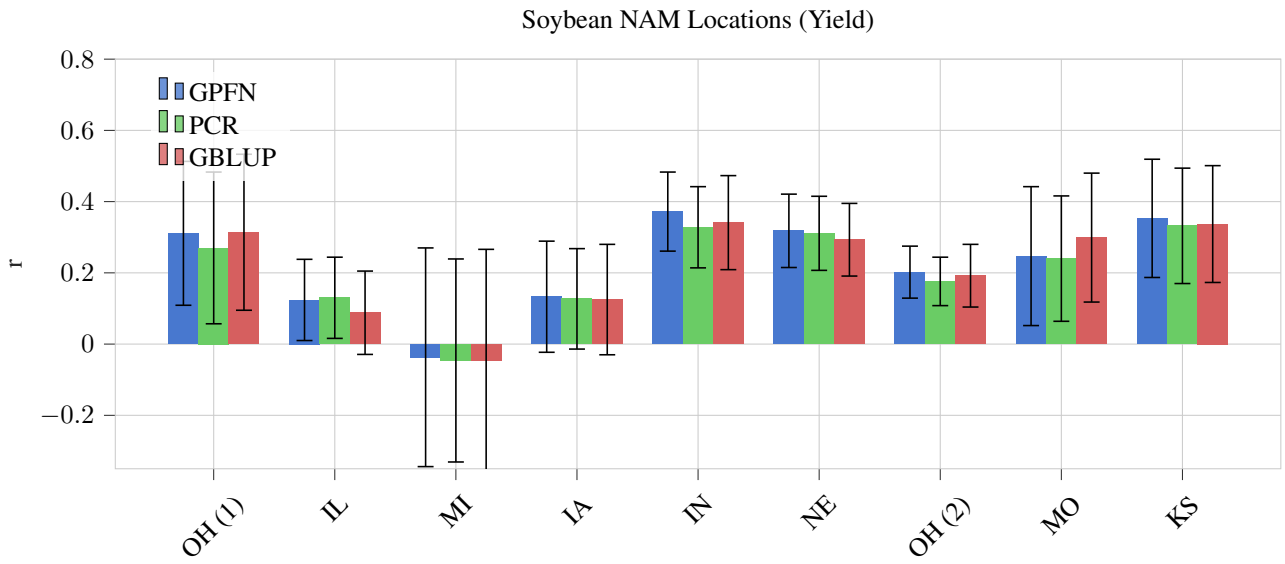

Figure 5. Prediction performance results for the soybean NAM population. Detailed results are shown in Table S5

Discussion

Performance Versus Baselines. For the GPFN fit using the unstructured prior and applied to the wheat PYT, wheat landraces, and lentil diversity panel datasets, the proposed method significantly outperformed GBLUP in 13 out of 16 traits, and significantly under-performed in none (Supplementary Note S3). Three of the four traits for the wheat PYT dataset saw double digit gains on average, with a substantial 22% performance uplift for thousand kernel weight. For the GPFN fit using the NAM prior and applied to the SoyNAM dataset, the results were more mixed. The GPFN significantly outperformed GBLUP in predicting yield at 3 of the 9 locations, while GBLUP was significantly better in one. The remaining 5 locations showed no statistically significant difference. The GPFN performs better than PCR across all datasets and traits with the exception of one, implying that the performance differences are not attributable to PCA and that the model is discovering an algorithm which is superior to linear regression.

It is interesting to note that the GPFN trained on the unstructured prior shows generalization performance outside of the range of populations represented by its prior. For example, for the extremely small wheat PYT and lentil diversity panel datasets, the size of the training set is significantly smaller than the smallest datasets the GPFN was exposed to during training. Similarly, for the wheat landraces dataset, the training set is significantly larger. The PYT data also contains far fewer SNPs than were simulated. Perhaps the most relevant difference is the types of individuals – the unstructured prior simulates a wild population with random mating, while the individuals in the real datasets are likely far more inbred. Despite these mismatches between the real data and the synthetic populations which the GPFN encountered during prior fitting, the GPFN exceeds the baseline performance convincingly in every case. There may be more performance yet available if these types of populations can be simulated more accurately.

GPFNs Move the Frontier from Prediction Models to Simulation. Much effort has been dedicated to developing models which are well-suited to genomic prediction. This includes linear models which imply different distributions of marker effects (GBLUP, LASSO, BayesA, BayesB), as well as neural network methods which attempt to propose the "right" cocktail of regularization and architectural modifications to capture important information without overfitting the training set (Ma et al., 2018; Wang et al., 2023; Gao et al., 2023). However, such an iterative approach where a model is tweaked until performance is improved on a particular dataset is more likely to result in a model which is tailored to a particular dataset or the specific trait architectures therein than it is to result in a model with good generalization capabilities. In the worst case, such an approach samples mostly noise, where the individual changes have zero effect size, but eventually a model which performs better than the competing models on the target dataset is found by chance.

The GPFN approach is fundamentally different: the model architecture is held constant, while the prior is made as broad and realistic as possible. Overfitting is no longer an issue, as training data is practically infinite (as the prior can be sampled any number of times) and PFNs are intended to fit the prior as closely as possible (Müller et al., 2021). The more diverse the prior is, the wider the range of quantitative trait architectures it is expected to generalize to. The proposed approach effectively shifts the focus from blindly searching over model architectures to creating the most accurate possible simulation models for quantitative traits.

Additional Evidence Against the Importance of Marker Effects. Evidence has been presented that, when tasked with genomic prediction, deep neural networks tend not to estimate marker effects (Ubbens et al., 2021). Despite being indifferent to marker effects, these networks still perform comparably to classical linear methods. Similarly, GPFNs are also unable to estimate the effects of individual markers due

to the PCA projection. We show that they can perform even better than existing methods by maintaining a prior over populations instead of marker effects. This is corroborating evidence that, for polygenic traits, marker effects are less relevant for prediction than broad patterns of relatedness.

GPFNs that Fit the Prior Better Also Perform Better on Real Populations. The performance of a PFN should increase with the number of draws from the prior, as it fits the prior more tightly. Using the wheat landraces dataset, we tested the generalization ability of a GPFN at different points during prior fitting (Figure 6). It is clear that generalization does improve over the course of prior fitting, with more performance potentially available with additional fitting beyond one million draws.

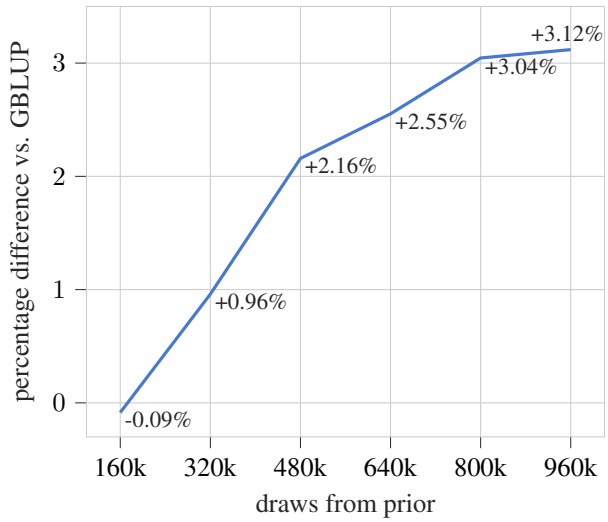

Figure 6. Performance on the wheat landraces dataset with a 50-50 train-test split, as a function of draws from the prior. Mean performance difference across all traits is shown.

Limitations. The GPFN approach comes with some novel and practical limitations. Using PCA projection before embedding means that the size of the training set must be at least as large as the number of components used. For example, when 100 principal components are used, the training set must be comprised of at least 100 individuals. This is not a limitation shared by competing methods such as GBLUP, which do not perform dimensionality reduction prior to model fitting.

As with any simulation study, the relevance of the prior (and therefore the expected performance of the model) is dependent on the accuracy of the simulation. We endeavoured to represent a wide range of trait architectures in our simulations, but it is impossible to capture the full range of diversity present in real-world scenarios. For example, as our simulator is focused on quantitative traits, performance may suffer when presented with discrete or ordinal phenotypes (such as measurements in days, or disease ratings) as these types of phenotypes were not simulated during model fitting.

As mentioned previously, the prior needs to match the population well or performance will be affected. For example, using a GPFN trained on the unstructured prior to perform inference on the between-families NAM structured population resulted in a mean performance shortfall of almost 15% versus the baseline (data not shown). For this reason, a prior should ideally be made for each unique breeding scheme in order to maximize the performance of the technique. Some standard breeding schemes can be modelled and reused for those applications – however, if the breeding scheme is nonstandard or the training and testing individuals are not split between different subpopulations in the same way, a new prior needs to be created.

Conclusion

In this work, we propose the Genomic Prior-Data Fitted Network, a class of Prior-Data Fitted Network (PFN) for performing genomic prediction. Through evaluations with four different populations of crop plants, we have demonstrated that GPFNs have the potential to surpass current methods as the new standard for genomic prediction. They also come with significant practical advantages compared to existing techniques based on deep neural networks, as they do not require training or tuning.

Although this initial proof of concept has demonstrated encouraging results with two priors, more evaluations are required on more types of structured and unstructured populations. In particular, more evaluations are required for breeding populations, such as those found in recurrent breeding programs. The lack of this type of data in the public domain means that it will be up to the community to pursue these evaluations to further validate the GPFN approach in their own populations. We anticipate that the ability of GPFNs to model the idiosyncrasies of these structured populations will unlock even more prediction performance and genetic gain in the future.

ACKNOWLEDGEMENTS

We would like to acknowledge Alexandra Ficht for providing the data for the wheat preliminary yield trial.

References

Abdollahi-Arpanahi, R., Gianola, D., and Peñagaricano, F. (2020). Deep learning versus parametric and ensemble methods for genomic prediction of complex phenotypes. *Genetics Selection Evolution*, 52:1–15.

Azodi, C. B., Bolger, E., McCarren, A., Roantree, M., de los Campos, G., and Shiu, S. H. (2019). Benchmarking parametric and machine learning models for genomic prediction of complex traits. *G3: Genes, Genomes, Genetics*, 9(11):3691–3702.

Chen, L., Yang, S., Araya, S., Quigley, C., Taliercio, E., Mian, R., Specht, J. E., Diers, B. W., and Song, Q. (2022). Genotype imputation for soybean nested association mapping population to improve precision of qtl detection. *Theoretical and Applied Genetics*, 135(5):1797–1810.

Crossa, J., Jarquín, D., Franco, J., Pérez-Rodríguez, P., Burgueño, J., Saint-Pierre, C., Vikram, P., Sansaloni, C., Petroli, C., Akdemir, D., et al. (2016). Genomic prediction of gene bank wheat landraces. *G3: Genes, Genomes, Genetics*, 6(7):1819–1834.

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., Burgueño, J., González-Camacho, J. M., Pérez-Elizalde, S., and Beyene, Y. (2017). Genomic selection in plant breeding: methods, models, and perspectives. *Trends Plant Sci.*, 22(11):961–975.

Ficht, A., Konkin, D. J., Cram, D., Sidebottom, C., Tan, Y., Pozniak, C., and Rajcan, I. (2023). Genomic selection for agronomic traits in a winter wheat breeding program. *Theoretical and Applied Genetics*, 136(3):38.

Gao, P., Zhao, H., Luo, Z., Lin, Y., Li, Y., Kong, F., Fang, C., and Wang, X. (2023). Soydngp: A web-accessible deep learning framework for genomic prediction in soybean breeding. *bioRxiv*, pages 2023–06.

Goddard, M. and Hayes, B. (2007). Genomic selection. *J. Anim. Breed. Genet.*, 124(6):323–330.

Goddard, M. E., Wray, N. R., Verbyla, K., Visscher, P. M., et al. (2009). Estimating effects and making predictions from genome-wide marker data. *Stat. Sci.*, 24(4):517–529.

Haile, T. A., Heidecker, T., Wright, D., Neupane, S., Ramsay, L., Vandenberg, A., and Bett, K. E. (2020). Genomic selection for lentil breeding: Empirical evidence. *Plant Genome*, 13(1):1–15.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2022). TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. arXiv: 2207.01848.

John, M., Haselbeck, F., Dass, R., Malisi, C., Ricca, P., Dreischer, C., Schultheiss, S. J., and Grimm, D. G. (2022). A comparison of classical and machine learning-based phenotype prediction methods on simulated data and three plant species. *Frontiers in Plant Science*, 13:932512.

Ma, W., Qiu, Z., Song, J., Li, J., Cheng, Q., Zhai, J., and Ma, C. (2018). A deep convolutional neural network approach for predicting phenotypes from genotypes. *Planta*, 248(5):1307–1318.

Meuwissen, T. H., Hayes, B. J., and Goddard, M. (2001). Prediction of total genetic value using genome-wide dense marker maps. *genetics*, 157(4):1819–1829.

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. (2021). Transformers can do bayesian-inference by meta-learning on prior-data. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*.

Pérez-Enciso, M., Ramírez-Ayala, L. C., and Zingaretti, L. M. (2020). Seqbreed: a python tool to evaluate genomic prediction in complex scenarios. *Genetics Selection Evolution*, 52(1):1–9.

Ray, S., Jarquin, D., and Howard, R. (2023). Comparing artificial-intelligence techniques with state-of-the-art parametric prediction models for predicting soybean traits. *The Plant Genome*, 16(1):e20263.

Ubbens, J., Parkin, I., Eynck, C., Stavness, I., and Sharpe, A. G. (2021). Deep neural networks for genomic prediction do not estimate marker effects. *The Plant Genome*, 14(3):e20147.

VanRaden, P. (2008). Efficient methods to compute genomic predictions. *Journal of Dairy Science*, 91(11):4414–4423.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, K., Abid, M. A., Rasheed, A., Crossa, J., Hearne, S., and Li, H. (2023). Dnngp, a deep neural network-based method for genomic prediction using multi-omics data in plants. *Molecular Plant*, 16(1):279–293.

Xavier, A., Muir, W. M., and Rainey, K. M. (2016). Assessing predictive properties of genome-wide selection in soybeans. *G3: Genes, Genomes, Genetics*, 6(8):2611–2616.

Xavier, A. and Rainey, K. M. (2020). Quantitative genomic dissection of soybean yield components. *G3: Genes, Genomes, Genetics*, 10(2):665–675.

Zingaretti, L. M., Gezan, S. A., Ferrão, L. F. V., Osorio, L. F., Monfort, A., Muñoz, P. R., Whitaker, V. M., and Pérez-Enciso, M. (2020). Exploring deep learning for complex trait genomic prediction in polyploid outcrossing species. *Frontiers in plant science*, 11:25.

Supplementary Note S1: Supplementary Figures

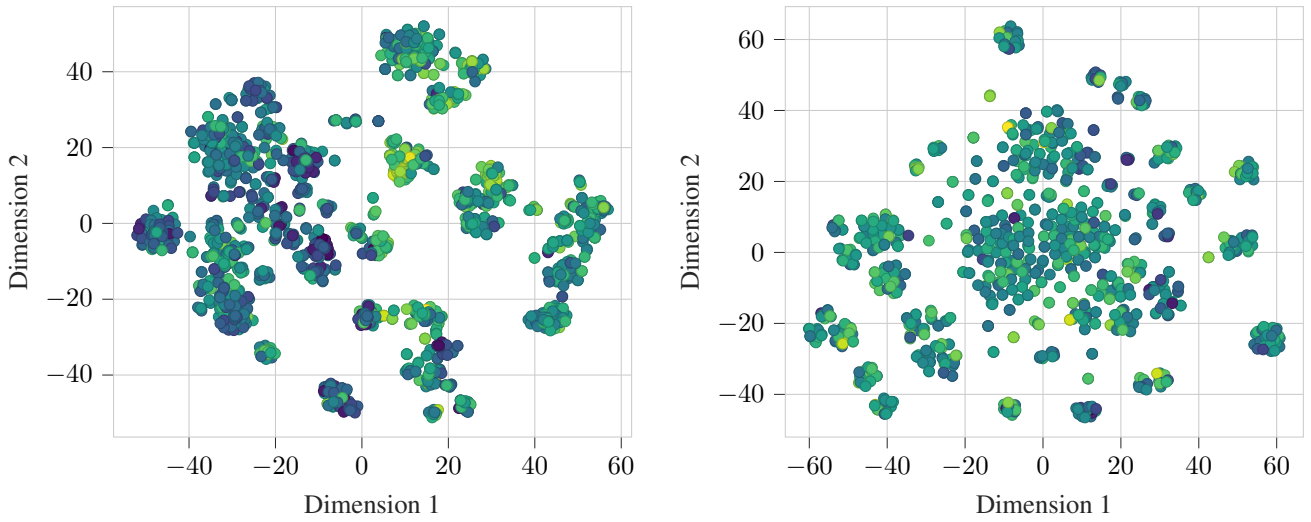

Figure S1. The wheat landrace population (left) and a draw from the unstructured prior (right) visualized with t-SNE. Points are shaded by phenotype (hardness for the real population).

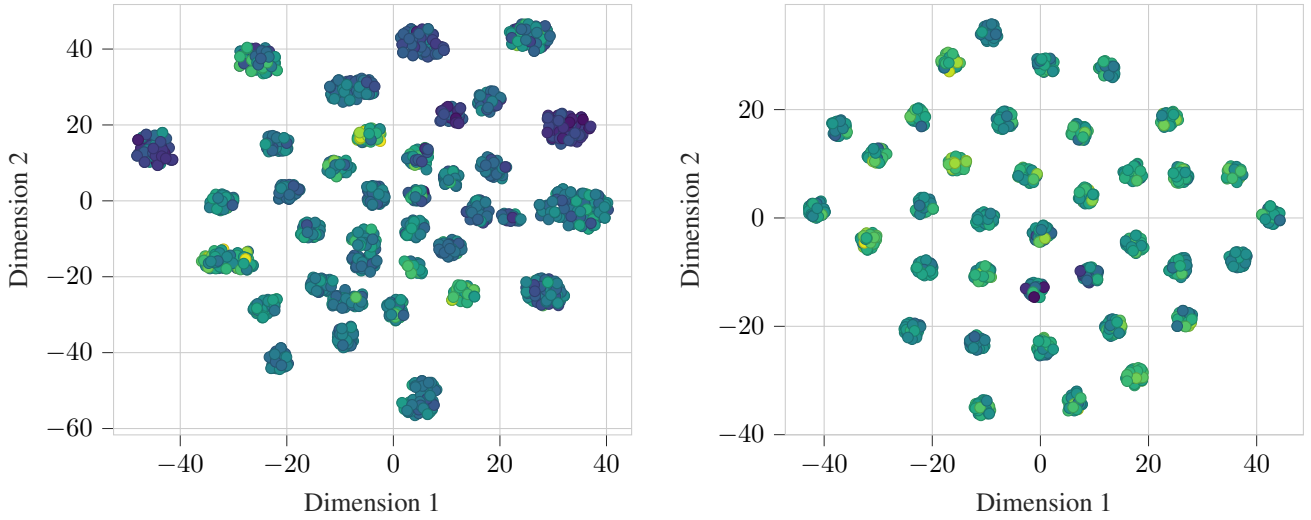

Figure S2. The soybean NAM population (left) and a draw from the NAM prior (right) visualized with t-SNE. Points are shaded by phenotype.

Supplementary Note S2: Simulation Parameters

timesteps	**p split**	**num base**	**num snps**	h^2	H
[20, 60]	[0.1, 0.3]	[500, 1000]	[7000, 36000]	[0.2, 0.8]	[0.05, 0.15]
p dom	**alpha**	**beta**	**recomb. rate**	**num qtl**	**MAF**
[0.0, 0.25]	[0.1, 0.3]	[1.0, 9.5]	[0.5, 2.5]	[5, 100]	[0.1, 0.3]

Table S1. Parameters for priors. All parameters are sampled uniformly. **p split**: probability of splitting into a new subpopulation; **num base**: number of individuals in the founding population; **p dom**: probability of dominance effect; **alpha, beta**: parameters of gamma distribution for marker effects; **recomb. rate**: recombination rate in cM/Mbp; H: heterozygosity; **MAF**: minor allele frequency.

Supplementary Note S3: Detailed Results

	GPFN	GBLUP	PCR	GPFN % diff.	PCR % diff.
Yield	0.528 (0.071) *	0.467 (0.089)	0.518 (0.089) *	14.688 (12.234)	12.895 (19.239)
Height	0.595 (0.039) *	0.542 (0.029)	0.562 (0.048) *	10.009 (8.366)	3.820 (9.755)
Maturity	0.439 (0.071)	0.429 (0.077)	0.408 (0.080)	3.221 (13.530)	-3.982 (14.927)
TKW	0.446 (0.067) *	0.379 (0.093)	0.413 (0.088) *	22.266 (23.613)	10.810 (14.526)

Table S2. Detailed results for the wheat preliminary yield trials. Pearson's r is reported. Standard deviations are shown in brackets. Results significantly higher than GBLUP (one-tailed t-test, $p < 0.01$) are denoted by *.

	GPFN	GBLUP	PCR	GPFN % diff.	PCR % diff.
Hardness	0.690 (0.026) *	0.675 (0.029)	0.673 (0.027)	2.252 (0.976)	-0.240 (1.533)
Length	0.762 (0.021) *	0.738 (0.018)	0.731 (0.022)	3.165 (0.826)	-1.015 (0.748)
PHT	0.344 (0.034)	0.348 (0.030)	0.335 (0.032)	-1.050 (2.016)	-3.650 (3.394)
PROT	0.553 (0.025) *	0.545 (0.022)	0.542 (0.021)	1.556 (1.129)	-0.513 (1.847)
TESTW	0.628 (0.037) *	0.609 (0.036)	0.616 (0.035)	3.109 (1.183)	1.090 (1.138)
TKW	0.667 (0.019) *	0.644 (0.017)	0.636 (0.017)	3.654 (1.124)	-1.240 (0.826)
Width	0.733 (0.017) *	0.714 (0.018)	0.716 (0.017)	2.744 (0.854)	0.324 (0.880)

Table S3. Detailed results for the wheat landraces. Pearson's r is reported. Standard deviations are shown in brackets. Results significantly higher than GBLUP (one-tailed t-test, $p < 0.01$) are denoted by *.

	GPFN	GBLUP	PCR	GPFN % diff.	PCR % diff.
DTF	0.794 (0.032) *	0.763 (0.032)	0.792 (0.033) *	4.092 (1.812)	3.828 (2.749)
DTM	0.708 (0.115)	0.693 (0.121)	0.664 (0.140)	2.629 (3.188)	-5.109 (6.173)
DTS	0.820 (0.028) *	0.785 (0.040)	0.805 (0.039) *	4.618 (4.381)	2.609 (3.848)
REP	0.610 (0.062) *	0.585 (0.085)	0.579 (0.078)	5.255 (7.516)	-0.510 (7.204)
VEG	0.815 (0.040) *	0.788 (0.037)	0.794 (0.032)	3.428 (2.686)	0.833 (1.648)

Table S4. Detailed results for the lentil diversity panel. Pearson's r is reported. Standard deviations are shown in brackets. Results significantly higher than GBLUP (one-tailed t-test, $p < 0.01$) are denoted by *.

	GPFN	GBLUP	PCR
OH Mian	0.311 (0.202)	0.314 (0.219)	0.270 (0.213)
IL Diers	0.124 (0.114) *	0.088 (0.117)	0.130 (0.114) *
MI	-0.037 (0.307)	-0.047 (0.313)	-0.046 (0.285)
IA Beavis	0.133 (0.156)	0.125 (0.155)	0.127 (0.141)
IN	0.372 (0.111) *	0.341 (0.132)	0.328 (0.114)
NE Specht	0.318 (0.103) *	0.293 (0.102)	0.311 (0.104)
OH McHale	0.202 (0.073)	0.192 (0.088)	0.176 (0.068)
MO Shannon	0.247 (0.195)	0.299 (0.181) *	0.240 (0.176)
KS Schapaugh	0.353 (0.166)	0.337 (0.164)	0.332 (0.162)

Table S5. Detailed results for the between-families soybean NAM population. Pearson's r is reported. Standard deviations are shown in brackets. Results significantly higher than GBLUP (one-tailed t-test, $p < 0.01$) are denoted by *, and results where GBLUP is significantly higher than GPFN are denoted by *.