12-23-2022

Yield prediction through integration of genetic, environment, and management data through deep learning

Daniel R. Kick

Jason G. Wallace

James C. Schnable

Judith M. Kolkman

Barış Alaca

See next page for additional authors

Authors

Daniel R. Kick, Jason G. Wallace, James C. Schnable, Judith M. Kolkman, Barış Alaca, Timothy M. Beissinger, Jode Edwards, David Ertl, Sherry Flint-Garcia, Joseph L. Gage, Candice N. Hirsch, Joseph E. Knoll, Natalia de Leon, Dayane C. Lima, Danilo E. Moreta, Maninder P. Singh, Addie Thompson, Teclemariam Weldekidan, and Jacob D. Washburn

G3, 2023, **13(4)**, jkad006
https://doi.org/10.1093/g3journal/jkad006
Advance Access Publication Date: 10 January 2023
Investigation

Yield prediction through integration of genetic, environment, and management data through deep learning

Daniel R. Kick [1,2] Jason G. Wallace [3] James C. Schnable [4] Judith M. Kolkman [5] Barış Alaca [6,7] Timothy M. Beissinger [6,7] Jode Edwards [8] David Ertl [9] Sherry Flint-Garcia [1] Joseph L. Gage [10] Candice N. Hirsch [11] Joseph E. Knoll [12] Natalia de Leon [13] Dayane C. Lima [14] Danilo E. Moreta [5] Maninder P. Singh [15] Addie Thompson [15] Teclemariam Weldekidan [16] Jacob D. Washburn [1,2,*]

[1]United States Department of Agriculture, Agricultural Research Service Plant Genetics Research Unit, Columbia, MO 65211, USA
[2]Division of Plant Sciences, University of Missouri, Columbia, MO 65211, USA
[3]Department of Crop & Soil Science, University of Georgia, Athens, GA 30602, USA
[4]Center for Plant Science Innovation and Department of Agronomy and Horticulture, University of Nebraska-Lincoln, Lincoln, NE 68588, USA
[5]School of Integrative Plant Science, Cornell University, Ithaca, NY 14853, USA
[6]Division of Plant Breeding Methodology, Department of Crop Science, University of Goettingen, Goettingen 37073, Germany
[7]Center for Integrated Breeding Research, University of Goettingen, Goettingen 37073, Germany
[8]United States Department of Agriculture, Agricultural Research Service, Ames, IA 50011, USA
[9]Research and Business Development, Iowa Corn Promotion Board, Johnston, IA 50131, USA
[10]Department of Crop and Soil Sciences, North Carolina State University, Raleigh, NC 27695, USA
[11]Department of Agronomy and Plant Genetics, University of Minnesota, St. Paul, MN 55108, USA
[12]United States Department of Agriculture, Agricultural Research Service Crop Genetics and Breeding Research Unit, Tifton, GA 31793, USA
[13]Department of Agronomy, University of Wisconsin, Madison, WI 53706, USA
[14]Plant Breeding and Plant Genetics Program, University of Wisconsin, Madison, WI 53706, USA
[15]Department of Plant, Soil and Microbial Sciences, Michigan State University, East Lansing, MI 48824, USA
[16]Plant and Soil Sciences, University of Delaware, Newark, DE 19716, USA

*Corresponding author: United States Department of Agriculture, Agricultural Research Service Plant Genetics Research Unit, University of Missouri, Curtis Hall, Columbia, MO 65211, USA. Email: jacob.washburn@usda.gov

Abstract

Accurate prediction of the phenotypic outcomes produced by different combinations of genotypes, environments, and management interventions remains a key goal in biology with direct applications to agriculture, research, and conservation. The past decades have seen an expansion of new methods applied toward this goal. Here we predict maize yield using deep neural networks, compare the efficacy of 2 model development methods, and contextualize model performance using conventional linear and machine learning models. We examine the usefulness of incorporating interactions between disparate data types. We find deep learning and best linear unbiased predictor (BLUP) models with interactions had the best overall performance. BLUP models achieved the lowest average error, but deep learning models performed more consistently with similar average error. Optimizing deep neural network submodules for each data type improved model performance relative to optimizing the whole model for all data types at once. Examining the effect of interactions in the best-performing model revealed that including interactions altered the model's sensitivity to weather and management features, including a reduction of the importance scores for timepoints expected to have a limited physiological basis for influencing yield—those at the extreme end of the season, nearly 200 days post planting. Based on these results, deep learning provides a promising avenue for the phenotypic prediction of complex traits in complex environments and a potential mechanism to better understand the influence of environmental and genetic factors.

Keywords: phenotypic prediction, gene-by-environment interaction (GxE), GEM, convolutional neural network, deep learning

Introduction

Prediction of an organism's phenotype is a key challenge for biology, especially when integrating the effects of genetics, environmental factors, and human intervention. For many traits, prediction is complicated by interactions between these factors. For example, within a large multisite, multigenotype maize (*Zea mays*) study, more variation in grain yield is explained by interactions between genetic and environmental factors than by genetic main effects (Rogers *et al.* 2021). Including interaction effects between environmental and genomic data can improve predictive accuracy in novel environments or for new cultivars (Jarquin *et al.* 2021; Li *et al.* 2021).

Within agriculture, diverse methods have been applied to the task of predicting phenotype ranging from classical statistics (Jarquin *et al.* 2021; Rogers *et al.* 2021; Rogers and Holland 2021), machine learning (Westhues *et al.* 2021), physiological crop growth models (Technow *et al.* 2015), to combinations of these and other methods (Messina *et al.* 2018; Shahhosseini *et al.* 2021). Each model contains limitations such as lacking the capacity to model complex nonlinear responses (linear models) or interactions

Received: July 28, 2022. **Accepted:** December 23, 2022
Published by Oxford University Press on behalf of the Genetics Society of America 2023.

between factors, interpretability within a biological framework (machine learning models), or dependence on costly, low throughput data for calibration (crop growth models). Often simplifying assumptions are introduced into the model (e.g. linearity), into the data (dimensionality reduction, feature engineering), or into the experimental design (e.g. considering exclusively genetic, environmental, or managerial effects to the exclusion of all others). While this approach creates more manageable statistical models and enables a sufficiently powered study to be achieved with fewer resources, it limits the capacity of a model to generalize to new genotypes, environments, or management schemes. Furthermore, which factors are treated as "nuisance" variables varies between communities within agriculture: geneticists often restrict management regimes, while agronomists usually consider only a few cultivars. These approaches make it difficult to investigate the interactions between genetic, environmental, and management factors.

Predicting an organism's phenotype across genotypes, environments, and management strategies simultaneously requires a dataset containing many combinations of these features. Collecting such a dataset requires a large multisite, multicondition, experiment featuring diverse genetic backgrounds. The Genomes to Fields (G2F) initiative (McFarland *et al.* 2020) seeks to accomplish this. To date, it has collected measurements of grain yield and other phenotypic traits (plant height, days to silking, stalk lodging, and kernel row number) from about 180,000 plots planted at more than 160 environments. Environments are characterized using a WatchDog 2700 Weather station (Spectrum Technologies, Inc.) which collects continuous weather data though the season and collaborator-submitted soil samples. Across the initiative, over 2,500 maize hybrids have been tested, with Genotyping by Sequencing performed on inbred parental lines used. Beyond the data collection, a means of effectively incorporating diverse data types (genomics, management, soil measurements, weather, etc.) is needed, particularly one that avoids simplifying assumptions where possible.

One method with the potential to accomplish this is that of deep neural networks (DNNs) which have the capacity to approximate any function, provided they are sufficiently complex and have sufficient examples to learn from. This capability is present regardless of whether they are composed of dense fully connected (Hornik *et al.* 1989) or convolutional layers (Zhou 2020). Additionally, DNNs "learn" directly from the data provided which enables reduced feature engineering and dimensionality reduction. The methodology is also flexible with respect to data type, allowing the combination of variables that are static over a growing season (e.g. genotype) and those that are dynamic (e.g. temperature) in a single model (Washburn *et al.* 2021). While neural networks have been applied to the problem of predicting yield since at least 2001 (Liu *et al.* 2001) this field is rapidly developing, with advances in theory, software, and hardware enabling deeper and more accurate networks. Several recent studies have applied these methods with a relatively large dataset either with (Washburn *et al.* 2021) or without (Khaki *et al.* 2020) a genetic component into the model, with little feature engineering performed. Both relied instead on DNN's capacity to learn useful data transformations from the data directly.

Despite their promise, DNNs are not a panacea for prediction. DNNs are prone to overfitting to training data resulting in poor performance. Even when performing well, the complexity of these models can obscure what aspects of the data the model is using. Advances in deep learning have produced methods that reduce these limitations. For example, the use of convolutional layers minimizes the potential of overfitting because they perform well with fewer parameters relative to fully connected layers. Where fully connected layers are used, overfitting can be reduced by randomly removing neurons from a layer with a certain "dropout" percentage. While the inner workings of DNNs remain far less interpretable than simpler models (e.g. best linear unbiased predictor (BLUP) or physiological models), methods have been developed to aid in interpretation through identifying the importance of different features in the data which can be applied. These methods include salience (Simonyan *et al.* 2014), guided backpropagation (Khaki *et al.* 2020), and permutation-based metrics (Shahhosseini *et al.* 2021) among others (Samek *et al.* 2017). Here we use salience, which measures how much influence each input variable has on the predicted output by calculating the model's gradient with respect to the network's input. This results in a map of each features' importance illuminating the operation of the DNNs generated in this study.

Here, we leverage DNNs' capacity to determine feature importance from the data which permits us to remain agnostic as to which features, or combinations of features are most relevant. Furthermore, since DNNs are robust to lower-quality data and benefit from an abundance of data, we employ a strategy of minimal feature transformation and curation and maximal inclusion of observations. Using a minimally transformed dataset we begin the search space considered in Washburn *et al.* (2021), expand the space under consideration, and detail a sequence of reproducible steps and objective heuristics which produced the models under consideration. DNNs require an abundance of data for training. We begin by detailing a workflow that incorporates a wider number of years from the G2F Initiative than previous studies (Rogers *et al.* 2021; Rogers and Holland 2021; Washburn *et al.* 2021), while also limiting the effect of errant and absent measurements. Improving on past studies, we propose a new approach to model optimization whereby the model is broken into submodules for each data type and interactions between them, then each submodule is consecutively optimized, using a Bayesian optimization procedure to find a suitable structure based on the data itself. As far as we are aware, previous studies using deep learning for phenotypic prediction have instead employed simultaneous optimization (SO) of all model components (Washburn *et al.* 2021) or informal inductive tinkering. We compared models developed through consecutive and SO and tested them against a variety of classic machine learning and statistical methods to determine which performed best. To fairly assess model performance we detail a strategy of constructing testing, training, and validation sets stratified by season and location that is broadly useful to assessing model performance while avoiding overfitting the model to any location.

Materials and methods

Data preparation

We used data from the G2F initiative for the years 2014–2019 (McFarland *et al.* 2020), focusing on the sites within the continental United States. Each year's data are publicly available (https://www.genomes2fields.org/resources/), including weather and soil data for field sites, genomic data, management schedules (e.g. application of fertilizer, herbicides, irrigation), and yield (in addition to other phenotypic variables). We augmented this through additional genomic and weather data. Weather data retrieved from Daymet (Thornton *et al.* 2020) were used in quality control as discussed below and to infer data for locations which lacked a

functional weather station for some or all of the season. Daymet data were retrieved through wget (Techtonik 2015).

Custom scripts were used to aggregate and standardize terminology across years. Rather than itemizing each operation, we restrict ourselves to those which are likely to be of interest to those working with similar data sets and the cleaned dataset is available through Zenodo (10.5281/zenodo.6916775) as are the scripts used (10.5281/zenodo.7401113). Data cleaning scripts were written in Python (Van Rossum and Drake 2009, p. 3) and rely on scientific and general libraries (Seabold and Perktold 2010; Pedregosa *et al.* 2011; *fuzzywuzzy* 2017; Harris *et al.* 2020; Team Pandas Development 2020; Virtanen *et al.* 2020; Da Costa-Luis *et al.* 2022) along with plotting libraries for exploratory visualizations (Hunter 2007, p. 200; Inc 2015; Kibirige *et al.* 2021; Waskom 2021). We used Anaconda ("Anaconda Software Distribution" 2021) to manage the virtual environment.

Data preprocessing

Starting with the G2F initiative's single nucleotide polymorphism data, which was produced through genotyping-by-sequence for the inbreds used (McFarland *et al.* 2020), we filtered and then reduced the dimensionality of the genomic data with principal components analysis (PCA) using TASSEL version 5.2.74 (Bradbury *et al.* 2007). Once the data were reduced, the genomes were PCA transformed. We find that 31% of the variance is explainable by the first 8 principal components (PCs), 50% is explainable by the first 50 PCs, and >99% of the variance is explainable by 1,725 PCs. Each hybrid's coordinates in PC space were estimated as the average between its parent's coordinates—projecting each hybrid genotype on PC axes derived from their parents' genomes. This was done rather than creating simulated hybrids due to hardware and software constraints.

Environmental data required preprocessing as well. The soil dataset contains many missing values, having an average completion rate of 47% across all site-by-year combinations. For each variable in the soil dataset (see below), missing values were first linearly interpolated across years with respect to location. Locations with no observations for any years were imputed using k-nearest neighbors (kNN) based on the nearest 5 neighbors (physically nearest using longitude and latitude). Within the reported weather data, we observed evidence of equipment malfunction and imputed or adjusted values using linear models. The representation of management data was refined. Fertilizer applications were decomposed into the quantity of nitrogen, phosphorus, and potassium applied. Where fertilizer applications were lacking an application date, we estimated the time difference relative to the planting date with kNN imputation ($k = 5$) to cluster based on application quantity (e.g. a missing date of application for a nitrogen application would be imputed using the dates of the 5 applications most similar in the quantity applied). To define the time window to be used for modeling, we found the earliest within-season fertilizer application and the day of the latest harvest to bound the weather and management data. This resulted in a window of 75 days prior to planting and 204 days after (210 total including the planting day). A full discussion of data preprocessing is included in the "Data Preprocessing" section of the supplemental materials.

Full dataset overview

The above approach to data cleaning was designed to be permissive as deep learning benefits from access to an abundance of data. Following data cleaning 96,137 yield measurements remained. Fewer than half of these (41,513 measurements) were used in training or evaluating the model due to balancing observations with respect to location-year combinations through downsampling. In the full dataset (available at 10.5281/zenodo.6916775) the 96,137 observations were spread over 41 sites across 6 years (2014–2019) (158 site-year combinations). These data were not balanced with respect to observations per field site. Observations per site ranged from 156 at "GAH2" in 2016 to 3,589 at "MNH1" in 2018, with the median and mean site recording 498 and 608.5 observations per year, respectively. Across all observations, 3,671 unique genotypes were recorded, derived from 1,681 female and 223 male parent genotypes. 94,996 (98.8%) of the observations are from hybrids; inbreds account for the other 1,141. The number of replicates for each genotype varies widely from 1 to several hundred (e.g. 2369/LH123HT: 882, PHW52/PHN82: 421, B37/H95: 333, PHW52/PHM49: 331, B73/PHN82: 328), with a median of 18 and mean of 26.2 observations. Within a location 1–2 replicates per genotype is typical (median of 2, mean of 1.62) but ranges as high as 46 replicates (2369/LH123HT at "NCH1" in 2018).

In addition to genotypic data as represented by 1,725 PCs produced from the genomic data, we used 40 environmental and management variables; 21 soil variables and 19 weather and management variables. The 21 soil variables were used for modeling where soil pH measured using a 1:1 mixture of soil and distilled water. (SoilpH), Soil pH measured using the Woodruff method (WDRFpH), Soluable salt concentration in mmho/cm (SSalts), Organic matter in soil in percent (PercentOrganic), Available Nitrites in ppm (ppmNitrateN), Nitrogen per acre in lbs (NitrogenPerAcre), Available Potassium in ppm (ppmK), Available Sulfate in ppm (ppmSulfateS), Available Calcium in ppm (ppmCa), Available Magnesium in ppm (ppmMg), Available Sodium in ppm (ppmNa), Cation exchange capacity in meq/100g soil (CationExchangeCapacity), Percentage Hydrogen (PercentH), Percentage Potassium (PercentK), Percentage Calcium (PercentCa), Percentage Magnesium (PercentMg), Percentage Sodium (PercentNa), Phosphorus extracted using acid fluoride in ppm (ppmP), Percent sand composition in a sample (PercentSand), Percent silt composition in a sample (PercentSilt), and Percent Clay composition in a sample (PercentClay). For more details please consult the G2F documentation (https://datacommons.cyverse.org/browse/iplant/home/shared/commons_repo/curated/GenomesToFields_data_2019/c._2019_soil_data). The 19 weather and management variables used for deep learning were Nitrogen applied in lbs/acre (N), Phosphorus applied in lbs/acre (P), Potassium applied in lbs/acre (K), Daily Minimum Temperature in Degrees Celsius (TempMin), Daily Mean Temperature in Degrees Celsius (TempMean), Daily Max Temperature in Degrees Celsius (TempMax), Daily Mean Dew Point in Degrees Celsius (DewPointMean), Daily Mean Relative Humidity as a Percentage (RelativeHumidityMean), Daily Mean Solar Radiation in Watts per Square Meter (SolarRadiationMean), Daily Max Wind Speed in Meters per Second (WindSpeedMax), Daily Mean Wind Direction in Degrees (WindDirectionMean), Daily Max Wind Gust in Meters per Second (WindGustMax), Daily Mean Soil Temp in Degrees Celsius (SoilTempMean), Daily Mean Soil Moisture in Degrees Celsius (SoilMoistureMean), Daily Mean Ultra-violet Radiation in Micro-moles per meter-squared seconds (UVLMean), Photosynthetically Active Radiation in Micro-moles per Meter-Squared Seconds (PARMean), Daily Mean Photoperiod as a Percentage (PhotoperiodMean), Daily Estimated Water Vapor Partial Pressure in pascals (VaporPresEst), and Total water applied (including irrigation and precipitation) in mm (WaterTotalInmm).

For more details please consult the G2F documentation (https://datacommons.cyverse.org/browse/iplant/home/shared/commons_repo/curated/GenomesToFields_data_2019/b._2019_weather_data) and Daymet documentation (https://daymet.ornl.gov/overview).

Defining training, validation, and test sets

We generated train/test splits randomly, with the constraint that any location-year combination could appear in only the testing or training set. Nearby experimental sites were grouped for the purpose of generating training and testing sets. The 41 experimental sites were grouped into 29 experimental site groups using the distance between their GPS coordinates such that no 2 groups were within 0.5 degree from each other. While this threshold is inconsistent with respect to location it is sufficient to group nearby locations while excluding distant ones. To prevent over representation of certain groups, all site-group-by-year combinations were randomly down sampled so that none had more observations than the smallest number in the testing set. For use in hyperparameter selection, the training set was split into a training and validation set, stratifying by site-group-by-year groups. The validation set is used to assess the performance of a considered set of hyperparameter values without compromising the test set. This was repeated several times to prevent overfitting to a single validation set. For further details, refer to the "Data Training, Validation, and Test Sets" in the supplemental materials.

Prior to hyperparameter selection and training the input data were centered and scaled based on the mean and standard deviation of each parameter in the training data. Measurement units were otherwise left as recorded in the data to be consistent with the G2F initiative. In the case of yield, these values were a mean of ~147.397 bushels per acre and a standard deviation of ~48.169 bushels per acre, i.e. $y = (y_{\text{Original}} - 147.397)/48.169$. This practice in deep learning can reduce the number of epochs required for training, but if done incautiously can result in unrealistically high-performing models as information about the data in the test set "leaks" into the training set. Centering and scaling based on summary statistics calculated from only the training data and not the full data set avoids this. As data transformation occurs prior to determining cross-validation folds a different information leak is potentially introduced within the hyperparameter selection process but does not create an issue for the final evaluation of the model's performance because the test set data were not used in these calculations.

Model-specific data preparation

Linear and machine learning models required further data processing. For use in simple linear fixed effect models and in machine learning models, the weather and management time-series data were clustered to reduce their dimensionality. For each variable, we used time series k-means with dynamic time warping implemented through the tslearn library (Tavenard *et al.* 2020). K could range from 2 to 40. The smallest value of K where the silhouette score of $K+1$ was less than K was used. Where needed clusters were represented categorically through one hot encoding. BLUP models required the generation of relationship matrices which were produced from the genomic PCs, soil covariates, and weather and management time series. We used the process applied in Washburn *et al.* (2021), which is a modification of that described in Jarquín *et al.* (2014), for creating matrices for these main effects and genomic by soil and genomic by weather interaction effects.

Training and test set overview

Following the training and test set definition, 41,513 observations remained: 37,273 in the training set and 4,240 in the test set. Across these sets, all 41 sites, 6 years, and 158 site-year combinations were represented. These data were approximately balanced as described above with all site-year groups in the test set having 265 observations, and no site-year group in the training set exceeding this, although some did contain fewer observations. This also resulted in fewer genotypes being represented (3,006 total, a reduction of 665 genotypes). These hybrids resulted from 1,238 female and 68 male parent genotypes. The number of replicates for each genotype varies with the top 5 most represented genotypes being 2369/LH123HT (503), PHW52/PHN82 (232), PHW52/PHM49 (184), B73/PHN82 (173), and B73/MO17 (172) and the median and mean genotype having 9 and 13.8 observations, respectively. Most site-years contain a single replicate of a genotype (mean observations: 1.27).

The training and test sets do not overlap with respect to site-year combinations but share sites and genetics. 28 of the 41 total sites are exclusively found in the training data and account for 23,758 observations with the shared sites accounting for 13,515 observations. No sites are exclusive to the test set. Of the 3,006 genotypes present, 1,559 occur in both sets accounting for 27,131 observations in the training set and 4,221 in the test set. 1,435 genotypes (10,142 observations) are only found in the training set whereas 12 genotypes (19 observations) are exclusive to the test set. Counts of observations for each site-by-year group in the training and test set after downsampling are shown in Supplementary Table 1.

Model preparation

Overview

We sought to model genotype by environment by management interaction effects (GEM effects) in maize yield and to determine the utility of doing so. To this end, we optimized DNNs to predict yield with a single data modality (i.e. only genomic data, soil characteristics, or time-series data each by itself). We use one-dimensional convolutional layers to capture the time-dependent features of weather data, which have previously been used in yield prediction for this task (Khaki *et al.* 2020; Washburn *et al.* 2021). We used dense, fully connected layers for the other submodules of the DNN.

We pursued 2 strategies for tuning and training GEM models: consecutive optimization (CO) and simultaneous optimization (SO). CO tunes the hyperparameters of networks predicting yield from a single data modality (genomic data, soil data, or weather and management time-series data). Next, the prediction neurons are discarded and the output of the penultimate layer of each single modality network enters a set of layers to permit interactions between data modalities. Hyperparameters for the interaction layers are then tuned. The SO strategy by contrast allows for all hyperparameters to be selected concurrently, both those which affect the processing of a single data modality and those influencing interactions between modalities.

Hyperparameter search and training

We selected model architecture through a hyperparameter search using the "BayesianOptimization" tuner provided within the "keras-tuner" package (O'Malley et al. 2019). Models were written in Keras (Chollet 2015) with Tensorflow as a backend (Abadi *et al.* 2015) and run in a Singularity container (Kurtzer *et al.* 2017; SingularityCE Developers 2021). The hyperparameter ranges explored for each network are listed in Table 1. We used a custom subclassed version of the tuner to randomly

Table 1. Hyperparameter ranges: deep learning.

Category	Submodels	Hyperparameter	Range
Architecture	Genomic only	Layers	1–7
		Units	4–256
		Dropout fraction	0–0.3
	Soil only	Layers	1–7
		Units	4–64
		Dropout fraction	0–0.3
	Weather + management only	Pooling type	Max (1d), Ave. (1d)
		Layer repeats	1–7
		Convolution layers per repeat	1–4
		Filter size	4–512
	Interactions	Layers	1–7
		Units	4–256
		Dropout fraction	0–0.3
Training	Optimizer	Learning rate	0.1, 0.01, 0.001, 0.0001
		Beta 1	0.9–0.9999
		Beta 2	0.9–0.9999
	Other	Batch size	32–256, step = 16
		Epoch	1–1,000

The search space (Range column) for each component of the constructed neural networks. Optimized values for the architecture and training are provided in Tables 2 and 3, respectively.

select one of the previously defined validation folds to prevent overfitting to a single validation set without increasing the computational cost. For DNNs, a maximum of 40 hyperparameter sets were explored. Models were trained for up to 500 epochs with an early stopping patience of 5 epochs in models where convolution layers were varied (sequential optimization model with only weather and management data, concurrent optimization model) and up to 1,000 epochs with an early stopping patience of 7 epochs for all others. Regardless of network type, if the hyperparameters optimization had not concluded by 290 h after the script began, the process was terminated and the hyperparameter sets completed by that point were considered.

The best-performing 4 hyperparameter sets for each model were trained for 1,000 epochs and evaluated on 10 defined testing/validation set splits. Next, the validation losses over the duration of training were used to calculate the mean and standard deviation for each epoch. Then, the training duration was split into 10 bins and the average of the sum of validation loss mean and standard deviation was calculated, i.e. $\text{loss}_{\text{bin}} = (\sum_{i=1}^{n} \overline{l_i} + s_i)n$, where i is epoch relative to the beginning of the bin, $\overline{l_i}$ is the mean validation loss across cross-validation folds at the ith epoch and s_i is the standard deviation of the same. The hyperparameter set with the lowest value for the most bins was selected. Epoch number was set by calculating a rolling mean of validation loss with a window size of 20 epochs. For each epoch, we calculated the sum of the mean and standard deviation of the rolling mean and the total rolling validation loss. Then, we found the epochs which minimized these 2 values (subtracting 10 from the epoch number to account for the window size). The disagreement between the epochs which minimized these values ranged from 2 epochs in the case of the CO Genomic model and CO interaction model up to 404 epochs for the CO weather and management model. We used total rolling validation loss to decide on the epoch number for each model. With the selected hyperparameters and training duration we fit each model 10 times to account for random initialization and saved each replicate and its training history.

Benchmarking models

Overview

To contextualize the performance of the generated DNNs, we use the same training data to fit linear fixed effects models, BLUPs, and classic machine learning models, each described in more detail below. Linear fixed effect models were the least demanding fitting quickly (between approximately 0.16–3 minutes). Classic machine learning models required a similar amount of time (approximately 1–15 min) whereas BLUPs and DNNs were considerably more demanding. The BLUPs required considerable memory and were run using standard compute nodes on the ATLAS computing cluster at Mississippi State University, which provided 384 GB of RAM. BLUPs using a single data type fit in approximately 1.2 days whereas the interaction model required 7.2 days to complete. The DNNs required less RAM, but need a GPU to fit quickly. Using a 2 T V100-SXM2–32GB graphics cards on the ATLAS computing cluster at Mississippi State University, fitting the CO model took approximately 5.5 computer hours to fit, with genomic and soil subnetworks fitting quickly (on the order of minutes) and weather & management and interactions subnetworks requiring the bulk of the 5.5 h (1.2 and 4.2 h, respectively). The SO model fits in approximately 2 h. However, these values ignore the time required for hyperparameter tuning. For the SO model and each subnetwork of the CO model, some 54 models were trained: 40 for hyperparameter tuning, 4 for hyperparameter validation, and 10 to account for random initialization in the final model. The total time, resources, and type of resources (access to RAM vs GPUS), to deploy one of the above models varies widely based on the model type and extent of model optimization.

Linear fixed effects models

To aid in evaluating the efficacy of the models described below (best linear unbiased predictors, machine learning models, and DNNs) we constructed simple linear models to act as benchmarks. The simplest model was an intercept model, i.e. every predicted yield equals the mean yield in the training set ($\hat{y} = \bar{y}$). Additionally, we fit 4 linear regression models in R (R Core Team 2021) predicting yield with main effects for all 1,725 genomic PCs ($y = \sum_{i_g}^{1725} (x_{i_g}\beta_g) + \varepsilon$), 21 soil measurements ($y = \sum_{i_s}^{21} (x_{i_s}\beta_s) + \varepsilon$), 19 weather and management

Table 2. Selected deep learning hyperparameters: architecture.

Submodel or network	Hyperparameter	Specific layer	Consecutive optimization	Simultaneous optimization
Genomic only	Units	1	83	196
		2	133	47
	Dropout fraction	1	0.163923177	0.15214
		2	0.230663142	0.06061
Soil only	Units	1	38	19
		2	13	27
		3	45	
		4	29	
		5	4	
		6	4	
		7	4	
	Dropout	1	0.148724301	0.21342
		2	0.276340999	0.18589
		3	0.005434164	
		4	0.173380695	
		5	0	
		6	0	
		7	0	
Weather + management only	Pooling type	N/A	Max	Max
	Convolution layers per repeat	N/A	2	2
	Filter size	1	433	370
		2	436	303
		3	52	
		4	163	
		5	400	
		6	294	
Interaction	Units	1	152	10
		2	207	25
		3	206	126
		4	188	204
		5	44	45
		6		134
	Dropout	1	0.18658661	0.10201
		2	0.289893588	0.14809
		3	0.004841293	0.01536
		4	0.198121953	0.15658
		5	0.243027717	0.2428
		6		0.19048

Selected hyperparameters related to the architecture of each neural network, where optimization strategies resulted in different numbers of layers, empty cells are used (e.g. number of units in the soil submodel's third layer). Hyperparameters that were constrained to be identical for every layer in a submodel (e.g. Pooling type) have the "Specific Layer" field listed as "N/A".

clusters ($y = \sum_{i_w=1}^{19} (x_{i_w}\beta_w) + \varepsilon$), or all the above along with interaction effects between the first 8 genomic PCs (accounting for 30% of the variance) and the nongenomic variables ($y = \sum_{i_g}^{1725} \sum_{i_s}^{21} \sum_{i_w}^{19} (x_{i_g}\beta_g + x_{i_s}\beta_s + x_{i_w}\beta_w) + \sum_{i_g}^{8} \sum_{i_s}^{21} (x_{i_g} x_{i_s}\beta_{gs}) + \sum_{i_g}^{8} \sum_{i_w}^{19} (x_{i_g} x_{i_w}\beta_{gw}) + \varepsilon$). Interactions were limited to allow for model fitting on readily available hardware with default memory settings. This analysis was aided by common data wrangling and convenience libraries (Wickham *et al.* 2019; Bache and Wickham 2020; Müller 2020; Izrailev 2021) and feather file read/write capabilities through arrow (Richardson *et al.* 2021).

Best linear unbiased predictors

For creating the best linear unbiased predictor models, we use the Bayesian generalized linear regression (BGLR) (Perez and de los Campos 2014) R package to perform reproducing kernel Hilbert spaces (RKHS) regression and fit for 10,000 iterations following 5,000 burn-in iterations. Iterations were based on those used previously in the literature (Pérez-Rodríguez and de los Campos 2022). Applying previously detailed methods (Jarquín et al. 2014; Washburn *et al.* 2021) to produce K matrices representing genomic, soil, or weather and management relationships and genomic by soil or genomic by weather and management relationships. Each K matrix is a $(n \times n)$ matrix with each element being an evaluation of 2 sets of input variables. We create 3 single kernel Bayesian RKHS models (with a genome, soil, and weather and management kernel matrix, respectively) using a Gaussian kernel which are represented as

$$\begin{cases} y = 1\mu + u + \varepsilon \text{ with} \\ p(\mu,u,\varepsilon)\alpha N(u|0,K\sigma_u^2)N(\varepsilon|0,I\sigma_\varepsilon^2) \end{cases}$$

$$K(x_i, x_{i'}) = \exp\left\{-h \times \frac{\sum_{k=1}^{p} (x_{ik} - x_{i'k})^2}{p}\right\}$$

We also fit a multikernel RKHS model using all 3 single modality and both interaction K matrices which is defined as

$$\begin{cases} y = 1\mu + \sum_{l=1}^{L} ul + \epsilon \text{ with} \\ p(\mu,u_1,\ldots,u_L,\epsilon)\alpha \prod_{l=1}^{L} N(u|0,K_l\sigma_{ul}^2)N(\epsilon|0,I\sigma_\epsilon^2) \end{cases}$$

Please refer to the BGLR (Perez and de los Campos 2014) documentation for further details on Bayesian RKHS.

Classical machine learning models

Additional machine learning models were implemented through scikit-learn (Pedregosa *et al.* 2011; Buitinck *et al.* 2013) and

Table 3. Selected deep learning hyperparameters: training.

Network	Optimizer learning_rate	beta1	beta2	Other batch_size	numEpoch
CO: Genomic only	0.0001	0.953368	0.985947	96	12
CO: Soil only	0.01	0.928472	0.997516	176	199
CO: Weather + management only	0.0001	0.903649	0.929582	240	629
CO: Full network	0.01	0.98752	0.972311	112	364
SO: Full network	0.001	0.975893	0.994607	192	711

Selected hyperparameters for training each network that do not pertain to the architecture of the network itself (Training category in Table 1). Optimizer hyperparameters were supplied to the Adam optimizer.

hyperparameters for each were optimized through the hyperopt library (Bergstra *et al.* 2013) run within a Docker container. In a workflow similar to that of the DNN models, we generated models for each data modality independently, and with all data available. Time-series data were represented as clusters as described in "Data Preparation". For each model, we allowed the following hyperparameters to vary as described: (1) kNN: neighbors = 1–250, weights = 'uniform" or "distance"; (2) radius neighbors regressor (RNR): radius = 0.01–2000, weights = 'uniform" or "distance"; (3) random forest (RF), maximum depth = 2–200, minimum samples per leaf = 0–0.5; and (4) support vector machine with a linear kernel (SVR): Loss = 'epsilon_insensitive" or "squared_epsilon_insensitive", C = 1–5 (log uniformly distributed).

Cross-validation folds matched those as described previously and average loss across all folds was measured. We tested a minimum of 115 combinations for each model and selected the best-performing hyperparameters for each input dataset, reported in Table 4. Following selection, we trained each model and produced predictions on the testing and training data. This was repeated 10 times to account for randomness in model fitting.

Model evaluation

For every model described above, we calculate predicted yields for the test set and calculate root mean squared error ($\mathrm{RMSE} = \sqrt{(\sum_i^n \mathrm{Prediction}_i - \mathrm{Observation}_i)/n}$), normalized RMSE percent ($n\mathrm{RMSE} = 100 * \mathrm{RMSE}/(\sum_i^n \mathrm{Observation}_i/n)$), and r ($r = (\sum (x_i - \bar{x})(y_i - \bar{y}))/\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}$) using SciPy (Virtanen *et al.* 2020). Unless stated otherwise in the text RMSE and *n*RMSE will refer to the average value across replicates. Two observations were not predictable using the fit radius neighbors regressor and were predicted as the training set mean. For DNN and BLUPs using genomic data, we calculated RMSE and *r* for each site-by-year group using R (R Core Team 2021). For the best-performing DNN, we calculated and visualized the salience of features for each data modality. To examine the influence of allowing interactions we contrast these saliences with the saliences of SO single modality DNNs. Saliences were calculated by Tf-keras-vis (Kubota 2021). Visualizations were created with the use of rjson (Couture-Beil 2018), patchwork (Pedersen 2020), and ggplot2 (Wickham *et al.* 2019).

Results

DNNs can—but do not necessarily—outperform competing model types

When all data sources are incorporated, the CO DNN achieves the second lowest average RMSE (RMSE is in standard deviations, and when not otherwise specified, values refer to the average across replicates), surpassed only slightly by the interaction-containing BLUP model (RMSE of 0.948 and 0.937, *n*RMSE 14.554 and 14.388%, respectively). However, the standard deviation in performance across replicates is about 4 times higher for the BLUP (SD RMSE 0.058) than the CO DNN (SD RMSE 0.013) making the CO DNN a more consistent performer than the BLUP model. The simple linear fixed effects model ranked third (RMSE 0.973, *n*RMSE 14.933%) and DNN-SO ranked fourth (RMSE 1.024, *n*RMSE 15.716%) followed by the machine learning models tested (Fig. 2, Table 5). Pearson *r* values for these models follow similar patterns to the RMSE results. (Supplementary Fig. 2, Table 5). Additionally, performance is not uniform with respect to site-group-by-year combinations (Supplementary Fig. 3, Supplementary Tables 2 and 3).

Across available datasets, no one model outperforms all others (Fig. 2, Table 5). When restricted to genomic data, only the kNN model outperforms a simple intercept model (RMSE 1.078 and 1.088, *n*RMSE of 16.548 and 16.701%). SVR performed particularly poorly on this data (RMSE 1.212, *n*RMSE 18.718%). However, when restricted to only soil data, SVR performed best (RMSE 1.059, *n*RMSE 16.262%) followed by the linear fixed effects model (RMSE 1.071, *n*RMSE 16.441%). All models outperformed the intercept model. Most models performed better when trained on weather/management data than exclusively on genomic or soil data. The RF model was a clear exception to this, achieving an RMSE of 0.373, *n*RMSE 5.729% *above* the intercept model. Using only weather and management data the BLUP and SVR models (RMSE 0.945 and 0.985, *n*RMSE 14.500 and 15.114%) performed remarkably well.

CO resulted in a larger, more accurate final network

Two hyperparameter selection strategies were employed, CO and SO, which have the same range of possible networks (hyperparameter ranges are listed in Table 1), the same data driving network selection and both use Bayesian optimization. Despite this, the strategy applied resulted in notably different final architectures. A visual summary of the relative differences between network hyperparameters is shown in Fig. 1, with the hyperparameter values listed in Tables 2 and 3. Supplementary Fig. 1 provides a visual overview of the network architecture. We consider the effect of CO vs SO on each of the 4 subnetworks (processing exclusively genomic, soil, or weather/management factors or interactions between data modalities), listed in decreasing order of approximate similarity.

Genomic subnetworks resulting from CO and SO are both 2 layers, but the CO model widens somewhat (layer 1 = 83 units, 16% dropout, layer 2 = 133 units 23% dropout) while the SO model begins over twice as wide and constricts more (layer 1 = 196 units, 15% dropout, layer 2 = 47 units 6% dropout). The outputs of the subnetworks are flattened before entering the interaction subnetwork. The CO and SO interaction subnetworks contained a similar number of layers (CO: 5 vs SO: 6), but CO resulted in layers with similar widths before constricting at the

Table 4. Machine learning hyperparameter optimization.

Model	Hyperparameter	Range	Genomic Only	Soil Only	Weather + Management Only	Multiple
kNN	Weight Metric	Uniform, Distance	Uniform	Distance	Uniform	Distance
	K	1–250	237	248	248	49
RNR	Weight Metric	Uniform, Distance	Distance	Distance	Uniform	Distance
	Radius	0.01–2000	39.759518	3.406197	5.986679	40.375418
SVR	Loss	Epsilon Insensitive, Squared Epsilon Insensitive	Epsilon Insensitive	Epsilon Insensitive	Epsilon Insensitive	Squared Epsilon Insensitive
	C	1–5 (log uniform)	2.772318	5.613996	4.623351	2.787589
RF	Max Depth	2–200, q = 1 (q uniform)	64	10	102	7
	Min Samples/ Leaf	1–200, q = 1 (q uniform)	171	163	100	149

The search space (Range column) and optimized values used for each machine learning algorithm considered. Optimization was conducted separately for models using each data type.

Table 5. Performance across data sets.

Data set	Model	Mean RMSE	SD RMSE	Mean *n*RMSE	SD *n*RMSE	Mean *r*	SD *r*
a. Genomic	Intercept	1.088074		16.70137			
	LM	1.106886		16.99013		0.158382	
	BLUP	1.10248	0.000124	16.92249	0.001905	0.140023	0.000315
	kNN	1.078049	2.10E-05	16.5475	0.000323	0.153079	9.82E-05
	RNR	1.162622	0.012629	17.84564	0.193846	0.120362	0.004536
	RF	1.105258	0	16.96514	0	0.153836	3.16E-17
	SVR	1.219457	0.048874	18.71803	0.750191	0.059061	0.08288
	DNN-CO	1.100742	0.009229	16.89582	0.141659	0.149913	0.016833
b. Soil	Intercept	1.088074		16.70137			
	LM	1.071081		16.44054		0.243759	
	BLUP	1.071631	9.50E-05	16.44898	0.001458	0.239983	0.000327
	kNN	1.080058	0.00203	16.57833	0.031166	0.209905	0.012433
	RNR	1.078125	0.000526	16.54866	0.008073	0.211685	0.002881
	RF	1.082526	0	16.61621	0	0.205633	1.07E-16
	SVR	1.059478	0.00112	16.26243	0.017184	0.229315	0.009986
	DNN-CO	1.082901	0.009902	16.62197	0.151988	0.214768	0.014235
c. Weather + management	Intercept	1.088074		16.70137			
	LM	1.018394		15.63182		0.221639	
	BLUP	0.944645	0.038882	14.49981	0.596823	0.429585	0.079289
	kNN	1.049192	0	16.10455	0	0.360063	0
	RNR	1.084259	0	16.64281	0	0.144041	0
	RF	1.461314	0.094823	22.43041	1.45548	0.038772	0.05441
	SVR	0.984673	0.003254	15.11422	0.049949	0.36494	0.008508
	DNN-CO	1.018051	0.074367	15.62656	1.14149	0.328728	0.171357
d. Multiple types	Intercept	1.088074		16.70137			
	LM	0.972882		14.93323		0.389471	
	BLUP	0.937387	0.057676	14.3884	0.885301	0.461141	0.091121
	kNN	1.063364	1.88E-06	16.32208	2.89E-05	0.231872	2.78E-06
	RNR	1.09015	1.40E-16	16.73324	1.12E-15	0.176418	3.01E-16
	RF	1.106942	0.002913	16.99099	0.044711	0.290892	0.009171
	SVR	1.040812	0.041922	15.97593	0.643488	0.29396	0.066996
	DNN-CO	0.948143	0.01286	14.5535	0.197387	0.426265	0.020095
	DNN-SO	1.023853	0.034579	15.71561	0.53077	0.272062	0.084996

Model performance is summarized for those models displayed in Fig. 2 with respect to data type or types provided. Performance is represented as root mean squared error (RMSE), RMSE normalized to the center and scaled yield range (*n*RMSE), and *r*. Standard deviation (abbreviated SD) of these measures across replicate models is reported except for the intercept only model and fixed effects models which converge on a single model. Note that in some cases machine learning models converge and thus have standard deviations of 0.

last layer (units = 152, 207, 206, 188, 44, dropout percentages = 19%, 29%, 0.5%, 20%, 24%), SO resulted in layers with very few units initially which are later expanded (units = 10, 25, 126, 204, 45, 134, dropout percentages = 10%, 15%, 2%, 16%, 24%, 19%). The soil subnetwork resulting from CO is notably deeper than the one from SO (7 and 2 dense layers, respectively) but also narrows more by the last processing layer (2 vs 27 units). Finally, in the weather and management subnetwork CO resulted in a notably deeper network (6 pairs vs 2 pairs of convolution layers) but used a similar number of filters in the final convolution layer pairs (CO 294 vs SO 303).

The performance of these networks differs as well. The CO network was better at predicting yield in the testing set. It achieved a lower mean RMSE (CO: 0.948 vs SO: 1.024) and was more consistently accurate across replicates (standard deviation CO: 0.013 vs SO: 0.035). Similar results were seen in the normalized errors (*n*RMSE CO: 14.6% SO: 15.7%, standard deviation CO: 0.197%, SO: 0.531%). Similarly, average R^2 was higher in the CO network (CO: 0.171 vs SO: 0.032) and more consistent across replicates as well (standard deviation CO: 0.022 vs SO: 0.065).

Model performance differences are due, in part, to the heuristic used to select the number of training epochs and different

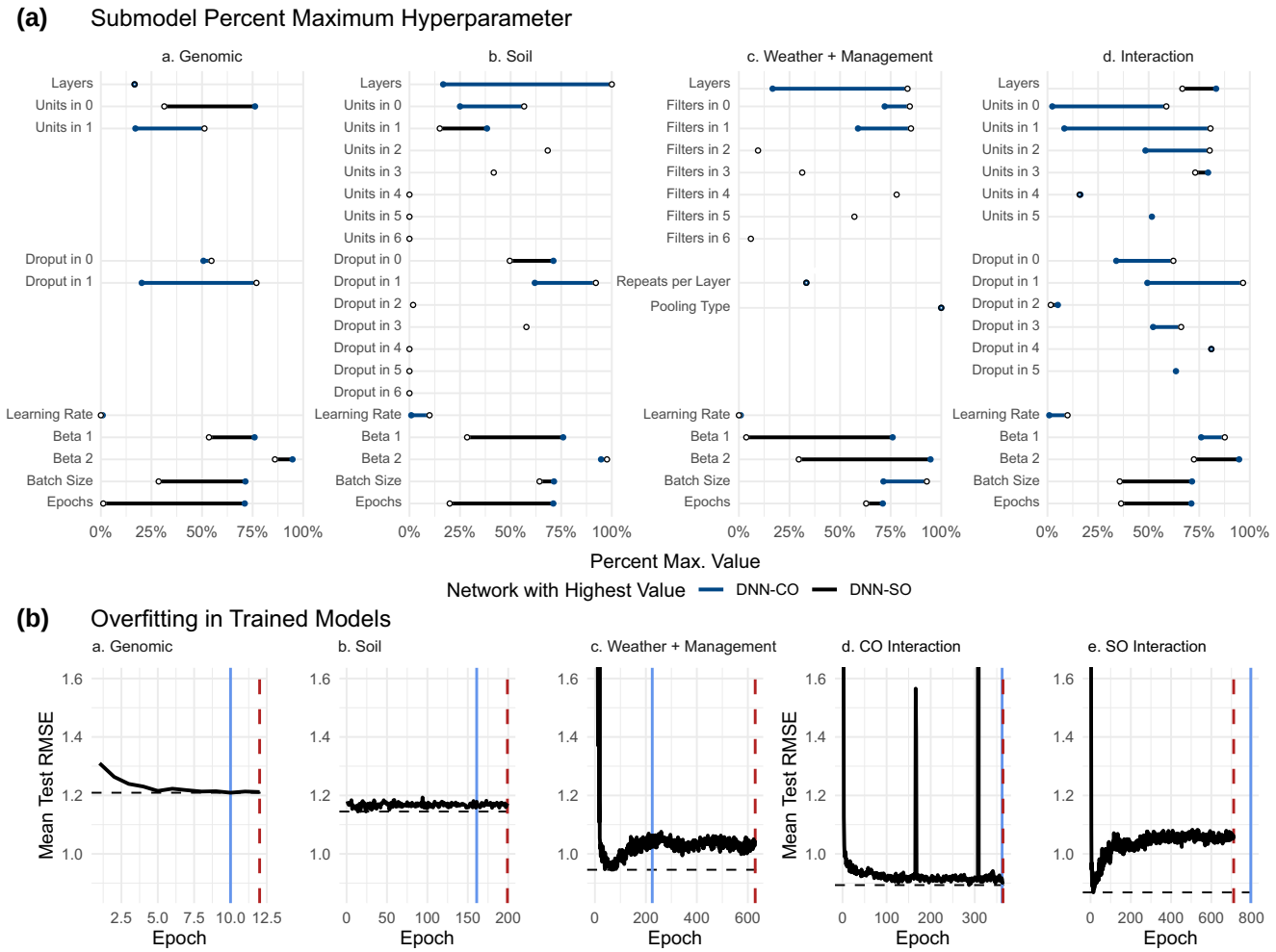

Fig. 1. Optimization strategy results in different network architectures and degree of overfitting in full model. a) Hyperparameters for each optimization strategy are shown as a percent of the allowed range. Data available for training are the same but the CO and SO strategy result in substantially different hyperparameter values and thus network architecture. For exact values refer to Tables 2 and 3. b) The average RMSE of the test set (across 10 replicates to account for random initialization of weights) is shown for each submodel (a.–e.). The horizontal dashed line indicates the minimum error achieved throughout the training duration. The vertical lines indicate the difference in error and epochs of the minimum value and the values selected through minimizing total validation error (dashed line), the heuristic used in this study, and the mean plus standard deviation of validation error (solid line), which was considered but not used. Both strategies considered failed to select the epoch resulting in the minimum loss in the test set for all submodels and resulted in apparent overfitting in the Weather and Management submodel (c.) and the SO model (e.). For additional comparisons of heuristic performance see Table 6.

tendencies for these models to overfit. The heuristic used to select the number of training epochs (sum of the rolling validation loss) and alternate heuristic considered (mean plus standard deviation of the rolling validation loss) resulted in networks with comparable performance, having on average 0.001 less RMSE. With the exception of the SO DNN, this also resulted in longer training durations. These ranged from an additional 2 epochs in the cases of the CO genomic and interaction models and as many as 404 epochs in the case of the CO weather/management, as shown in Table 6.

These training durations were often considerably longer than the optimal values as seen in Fig. 1b. Furthermore, the length of overtraining appears loosely proportional to the present minimum average RMSE each model achieved. The SO and CO weather models had the largest differences between optimal and used epoch numbers—differences of 697 and 563 epochs respectively and achieved 121 and 110% of the minimum possible RMSE. The CO soil model trained an excess of 185 epochs but only had RMSE at 102% minimum. The 2 training durations closest to the optimum were the CO genomic model (2 epochs over) and the SO model (77 epochs over). These models performed at just 100.2 and 101% minimum.

The SO model overfits faster and to a greater extent than the full CO model, which does not show evidence of substantial overfitting (Fig. 1b, d and e). The SO model achieves a loss lower than the CO model, and the accuracy worsens rapidly with further training. The different network sizes (CO containing more layers) may account for this difference. Improved heuristics for training duration could represent an opportunity for future refinements, which these results suggest could both increase the goodness of fit and reduce the computational resources needed to train these models.

Model performance generally improves through incorporating multimodal data and interactions

Incorporating multiple data sources and allowing interactions between data types generally appears to improve accuracy for DNNs, linear fixed effects models, and BLUPs. Allowing the use of multiple data types in the CO DNN reduced average RMSE by 0.070 (1.073% *n*RMSE) relative to the next best DNN, the CO

Table 6. Epoch selection underperformance.

Network	Epoch selected by			Average test loss			Proportion of minimum	
	Mean + SD	Sum of Losses	Best Possible	Mean + SD	Sum of Losses	Best Possible	Mean + SD	Sum of Losses
CO: Genomic Only	10	12	10	1.209216	1.21171	1.209216	1	1.002063
CO: Soil Only	161	199	14	1.178308	1.172763	1.144793	1.029276	1.024432
CO: Weather Only	225	629	66	1.041072	1.046358	0.945608	1.100955	1.106545
CO: Full Network	362	364	287	0.912883	0.903884	0.893123	1.022124	1.012048
SO: Full Network	796	711	14	N/A	1.052109	0.86811	N/A	1.211954

Comparison of differences in performance on test dataset due to epoch selection heuristic. Performance from the considered epoch selection heuristic, Mean plus Standard Deviation, and selected heuristic, Sum of Losses, are juxtaposed with the best epoch observed during training. The average loss (in RMSE) across replicates is reported along as raw values and percent of minimal loss. Missing values for the SO: Full network model are due to Mean plus Standard Deviation suggested epoch exceeding the epochs the model was trained for.

weather/management model. The SO DNN contains a different weather submodule and has a higher average RMSE than the weather/management model (by 0.006 RMSE, 0.089% *n*RMSE) but performed more consistently with a standard deviation of RMSEs 0.035 relative to 0.074. In linear models, the improvement relative to the next best model of the same type is 0.046 RMSE (0.699% *n*RMSE) for fixed effects models and 0.007 RMSE (0.111% *n*RMSE) for BLUPS.

The other methods tested do not show an improvement from increasing the number of data modalities used. kNN and SVR perform best with weather and management data only with the use of all data reducing performance by 0.014 RMSE (0.218% *n*RMSE) and 0.056 RMSE (0.862% *n*RMSE) respectively. RF and RNR perform best with soil data only the use of all data reduces performance by 0.024 RMSE (0.375% *n*RMSE) and 0.012 RMSE (0.185% *n*RMSE) respectively.

Which factors are most important to the CO DNN?

Among the genomic data we observe no major trend in salience with respect to PC (Supplementary Fig. 4a). The 2 most salient PCs are PC 26 (0.423) and PC 24 (0.402) which account for 0.350 and 0.392% of the total genomic variance respectively. Given that these saliences are relative to PCs, using salience to implicate specific genes or gene loci is infeasible. Among the soil factors, we find that the 5 with the highest average salience were soil pH (0.488), phosphorus ppm (0.487), potassium ppm (0.485), sulfate ppm (0.436), and percent organic matter (0.413) (Supplementary Fig. 4c).

Within the weather and management data, considering the average salience across the season (Supplementary Fig. 4d) 5 factors achieved an average salience greater than 0.140—total water (0.245), average solar radiation (0.198), maximum temperature (0.175), average wind direction (0.174), and estimated vapor pressure (0.173). The majority of factors had an average salience between 0.140 and 0.10 with 6 falling below this threshold—average soil temperature (0.095), maximum wind speed (0.084), average soil moisture (0.076), phosphorus applied (0.052), and potassium applied (0.033). Additionally, we find specific time points which appear to be salient broadly with the most salient region of time being within the first few days of planting, indeed 8 of the 10 days with the highest average salience are days 2–9 following planting.

How is factor importance altered by inclusion of interactions?

The full CO model, in addition to performing best (albeit by a small margin), presents an opportunity to directly compare the influence of interactions between data modalities on the salience of factors because the single modality subnetworks are identical except for the prediction layer. The salience of genomic factors differs notably between the 2 networks (Supplementary Fig. 4b). Salience of PCs differs by as much as 0.432 (PC 24), with the difference in the salience of the first 8 PCs (31% variance explained) ranging from 0.200 (PC1) to 0.309 (PC7). We find comparatively small differences in the salience of soil factors being between −0.011 and 0.0156 (Supplementary Fig. 4c).

In general, the salience map of the weather and management data features fewer broadly salient timepoints when interactions are included (Fig. 3a) than when they are not (Fig. 3b). The weather and management CO model contains a broadly salient time point around 25 days before planting and 6 days after planting. The SO model also appears to have peaks of salience around 150, 183, and 199 days after planting. When interactions are included the

majority of the salient time points become less so with the exception of the peak 6 days after planting as highlighted through subtraction of the 2 salience maps (Fig. 3c).

Discussion

Assumptions, potential sources of error, and opportunities for improvement

The results of this study are best understood with the data used and assumptions made kept in mind. The sole source of biological data in this study came from the G2F initiative (McFarland *et al.* 2020). The scale of this ambitious project increases the chances of data being absent or compromised due to equipment malfunction, logistical or procedural issues, and resource constraints. For example, many sites lack measurements for many soil properties across the seasons considered here, and the timing of fertilizer applications was absent in some cases. Our aim was to minimally filter the dataset while preventing missing or distorted values (many of which are not missing at random) from altering model accuracy and feature salience. We have aimed to reproducibly infer missing or aberrant values with relatively simple methods (e.g. imputation using linear models, kNN, and so on) but more sophisticated imputation techniques may have improved performance.

Alternatively, constraining the dataset to reduce the required imputation may have been an effective strategy. We elected to minimally filter observations because machine learning models, particularly deep learning, often benefit from having an abundance of data from which to learn feature relationships. For models where this is not the case, restriction of observations to the observations with the highest quality may be a preferable strategy. Note, however, that for distortions that are not randomly distributed, filtering may bias the sample and result in a model that appears to perform well but generalizes poorly (e.g. to sites similar to those with a preponderance of observations excluded).

Beyond including as many distinct locations and seasons as we could, we approximately balanced site-by-year groups through downsampling to avoid overfitting our DNNs to sites with more observations or biasing the selection of hyperparameters. This reduces the size of the dataset that can be used in training. Although outside the scope of this study, assessment of the sensitivity of DNNs to unbalanced group sizes, or exploration of alternate means of balancing groups (e.g. randomly *up sampling* small groups to equal the size of larger groups) would be valuable. Indeed, if the balance were not a concern, or if it could be effectively achieved without discarding observations in some groups, one could potentially employ more strict data filtering without producing a dataset too small to benefit from machine learning.

Substantial effort was devoted to producing testing, training, and validation sets that would not lead to overconfidence in the accuracy of our models. To this end, we kept observations within site-by-year groups in the same partition of the data. In effect, this prevents the model from being trained and tested on the same weather and management data. Furthermore, except in cases where soil features are static from season to season, the model will not be trained and tested on observations with identical soil features. Proceeding in this manner rather than selecting observations at random for the testing set further reduces an already small number of weather and management conditions. Incorporating historical data (Washburn *et al.* 2021) or expanding the dataset to include data from other sources represents 2 possible avenues to incorporate a greater diversity of weather and management conditions without compromising the testing set.

Depending on the intended application of a model, one may be able to achieve higher performance through altering some of the above decisions or replacing random assignments with a targeted approach. For example, we assume that all group-by-year combinations are equally likely to be of interest. However, if we assume that the distribution of sites collected match those of interest for prediction (i.e. one is interested in predicting *any* future observation collected by G2F and the number of observations per field site is representative of a future number of observations) then downsampling can be skipped, resulting a larger dataset. Similarly, with a narrower aim, e.g. prediction of yield within a specific region, testing or validation sets could be constrained to better select hyperparameters for or assess the predictive accuracy of site-by-year combinations within that region.

In summary, our decision to include as much data as possible and to limit the possibility of overfitting to specific sites and seasons represent possible opportunities for improvement. More sophisticated data imputation or more restrictive filtering, alternate means of balancing groups, and the incorporation of other data sources have the potential to improve model performance. Additionally, for more narrowly purposed models, nonrandom testing and training sets may represent a more accurate metric of predictive power, and indeed may deviate substantially from what we show here.

Tradeoffs in mean model performance, model consistency, and computational resources

While the best performance was achieved with a BLUP incorporating genomic, soil, weather, and management data, a DNN performed similarly to the BLUP model and with less variance in accuracy between model replicates (Fig. 2). A simple linear fixed effects model also performed fairly well and required substantially fewer computational resources and model selection efforts. This is true in terms of hardware (the BLUP required over 100 GB of RAM to fit while the DNN required the use of GPUs) and time (hours to days instead of minutes to fit). Across the different datasets considered, fixed effects models generally performed well. In cases where accuracy is not the sole factor under consideration, or where time or computational resources are limiting, simpler models may be "good enough" for the desired purpose. Furthermore, different models may be better suited to different goals such as achieving good predictive performance consistently (DNN-CO, multiple data types) or maximizing average predictive performance (BLUP, multiple data types) while potentially underperforming expectations due to variability in performance across model replicates.

Usefulness of CO in hyperparameter selection

We employed 2 strategies for hyperparameter optimization: CO hyperparameters for distinct "modules" of the network and SO for the network as a whole. CO reduces the range of possible combinations that are explored by allowing only one module to vary at a time. However, if 2 features in different data sets have a strong interaction effect (e.g. between genotype and weather patterns) then this approach will not necessarily allow for optimization to better capture this interaction. SO represents the reverse situation. With all features available, interactions between features in different tensors can be leveraged, but the hyperparameter space to explore is larger as all the hyperparameters are free to vary.

We find that the network resulting from CO substantially outperforms the one generated through SO. This should not be taken as a problem with SO *per se*. In other applications, or with a

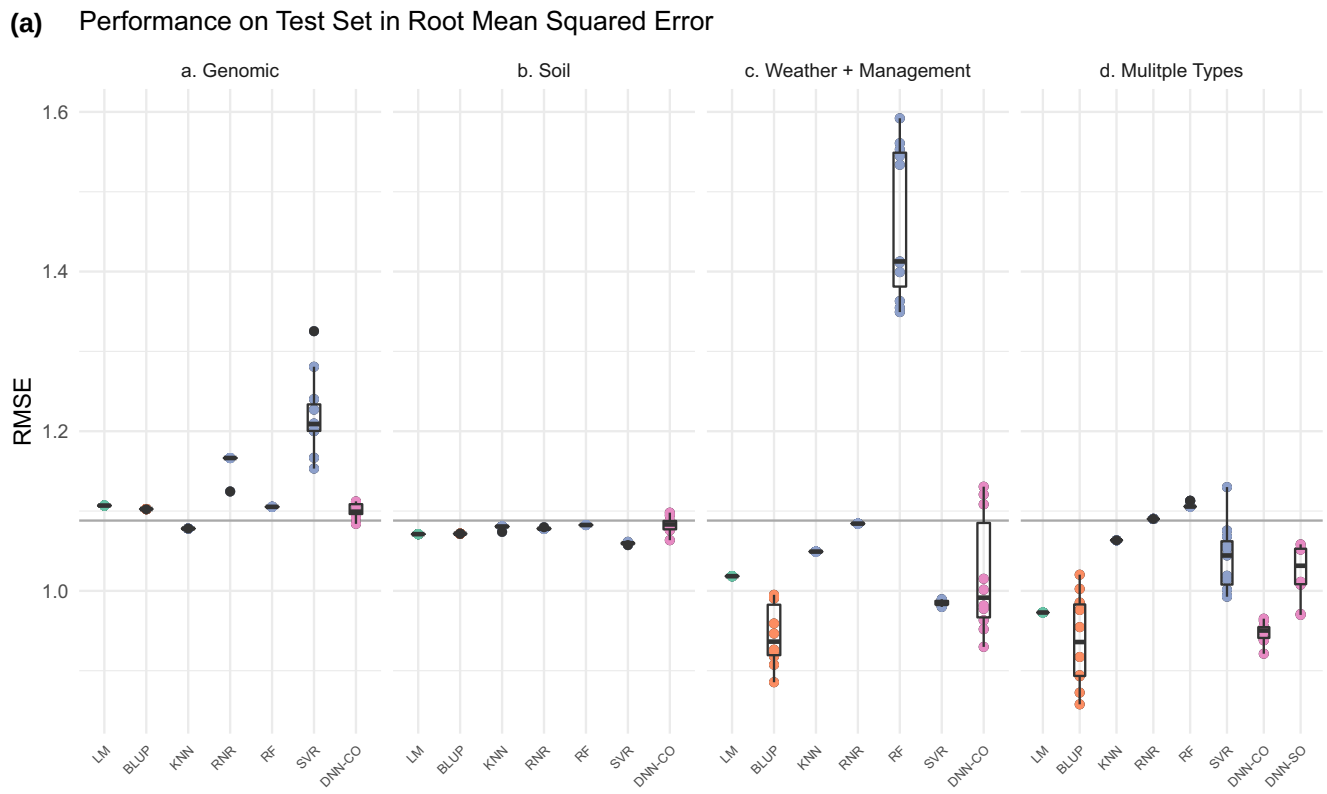

Fig. 2. Model performance across methodologies and data types. a) The RMSE of the testing set is shown for each data grouping (panels a–d) and class of model. Lower values indicate better model performance. As the data were centered and scaled RMSE is expressed in standard deviations of yield in the training set, i.e. ~48.169 bushels per acre. The horizontal line indicates the performance of an intercept model, i.e. using the mean of the training set yield as the prediction for all observations in the test set. For models that depend on a seed value the RMSE values for 10 trials (evaluated on the same data) are shown and standard Tukey box plots are provided. In deep learning models random initialization of weights at the beginning of training result in different performance across trials. Four groups of models are shown, linear fixed effects models, best linear unbiased predictors, machine learning models, and deep learning models. Machine learning models used were k-nearest neighbors (kNN), radius neighbor regression (RNR), random forest (rf), and support vector regression (SVR) with a linear kernel. Deep learning models are divided by whether they were part of the consecutive optimization strategy (DNN-CO) or the simultaneous optimization strategy (DNN-SO). Note that DNN-SO requires all data types and thus only appears in panel d.

different optimization algorithm, it may prove to be a more efficient means of deriving a useful architecture. Furthermore, it is conceivable that SO is effective but that additional trials were required. The SO DNN architecture was selected based on 40 trials whereas the CO DNN architecture was selected based on 40 trials *for each module* (160 trials across the whole network) which confounds the comparison. Selection of the training duration also warrants consideration. The SO model is capable of performing comparably to the CO model, but overfits more rapidly (Fig. 1b). Improved heuristics for selecting the training duration could increase the usefulness of the SO model while reducing computational demands as well.

As a pragmatic matter, CO benefits from the capacity to tune multiple modules at once. In our hands, the total time spent tuning was driven more by modules with computationally intensive components (convolution layers) rather than the number of modules to optimize. This benefit is dependent on the tuning algorithm used. We used a Bayesian optimization procedure that aims to produce useful hyperparameter combinations in fewer cycles than a simpler method such as grid approximation. However, because this method uses the performance of previously evaluated hyperparameters in selecting the next set, it does not permit parallelization in tuning a single network. If an optimization procedure that is conducive to parallelization were used (e.g. hyperband or grid approximation) with enough computational resources this benefit would be nonexistent.

Although we aimed to broaden the range of possible architectures relative to previous modeling on G2F data (Washburn *et al.* 2021), we constrained the overall structure to process each tensor individually then allowing for interactions between the final layer of each module. Other options might include, for example, allowing an interaction module to use both the first and final layers as input (instead of only the final one), or allowing which layers were to be used to be tuned.

An additional option that we did not explore is aiming to inform the structure of the selected network based on known relationships between features. Similar to our decision to minimally transform and filter the data, we elected to avoid "nudging" the architecture of the network in any direction in order to allow the data to inform it instead. Informing the model architecture based on known relationships, analogous to incorporating a prior, remains an interesting and potentially fruitful avenue to pursue.

Feature importance

Similar to the results of previous modeling (Washburn *et al.* 2021), we find that no single data grouping provides sufficient information to disregard all others. We note that weather and management data does reduce error substantially relative to genetic and soil data, but the variation in performance is large (Fig. 2). Only after integration of all data types do we see a relative reduction in error and consistency in this reduction in DNNs.

Here we focus on salience in the weather and management data as it provided the best average performance when used without other datasets. We find that the total water applied to the field (including irrigation and rainfall, termed "WaterTotalInmm") is the most influential factor for determining yield (Fig. 3, Supplementary Fig. 4d). This is sensible from a biological standpoint and is in agreement with previous models. Previous DNNs developed with a subset of G2F data also identified precipitation as substantially influencing yield (Washburn *et al.* 2021). Linear modeling results find similar results and suggest a positive association between precipitation early in development and yield (Rogers *et al.* 2021). Additionally, in a recent study using a hybrid machine learning and crop growth model prediction system, the authors found that water-related features (e.g. average drought stress, average water table in season) were important, although not as important as the trend in genetic and management improvements over time (Shahhosseini *et al.* 2021). The daily average of solar radiation (SolarRadiationMean) is the next most salient feature of this dataset, followed by the maximum temperature (TempMax) and the average wind direction (WindDirectionMean). A study employing a convolutional recurrent DNN to model county-level data likewise found solar radiation and maximum temperature as important features and note an apparent increase in the importance of temperature near planting time (Khaki *et al.* 2020). A time-dependent sensitivity can be observed in our model as well (Fig. 3).

The relationship driving the high average salience of the average wind direction is not clear. This feature likely correlates with unrecorded variables. Assessment of the topology and geographical surroundings of each field site to suggest what this measure may be linked to lies outside the scope of this study.

With respect to management interventions, although the addition of N, P, or K is not among the most salient weather and management features, we observe that nitrogen does have a mean salience comparable to relative humidity and photoperiod, while phosphorus and potassium are far lower. As noted previously (Washburn *et al.* 2021) limited salience of fertilizers could be due to the quantities used being too low to exert a substantial effect, or alternatively application of these elements may be insufficiently variable to reveal the effect.

Importance of GEM interactions accuracy in feature salience

Incorporating interactions between genetic, environmental, and management factors appears to have benefitted the accuracy of the DNN and BLUP models. When restricted to exclusively genomic information these models underperform simple intercept-only models and only improve slightly with the use of soil data. Performance for these models is substantially improved with the use of weather and management data. These performance differences are suggestive of a substantial environmental effect, exacerbated by the stratifying observations from site-by-year groups into exclusively training or testing sets. Allowing for interactions further improves performance with interaction-containing BLUPs having a lower average error (RMSE 0.937, *n*RMSE 14.388%) than the weather and management model (RMSE 0.945, *n*RMSE 14.500%), albeit with a higher standard deviation of RMSE (0.058 vs 0.040). A similar result is seen between the CO DNN with interactions (RMSE 0.948, *n*RMSE 14.554%) and weather and management model (RMSE 1.018, *n*RMSE 15.627%), both of which achieve lower average error than the SO Model (RMSE 1.024, *n*RMSE 15.715%). However, the 2 DNNs with interactions are more consistent having a far lower dispersion in RMSE, with standard deviations of 0.013 in the CO model and 0.035 in the SO model as compared with 0.074 in the CO weather model.

Interactions not only improve the accuracy of DNNs and consistency across replicates but also appear to be changed in the salience of individual features. This is most apparent in considering the weather and management features' salience (Fig. 3c). Relative to the submodel, incorporating interactions appears to increase the salience of irrigation, although it is highly salient in both models (relative to other time-series factors). Additionally, several broadly salient points in time, 2 of which are at the extreme end of the season, have diminished salience with the incorporation of interactions. This reduction is not uniform across all highly salient time points. A strong peak in salience shortly after planting is seen in both saliency maps which agrees with previously reported results (Washburn *et al.* 2021).

Conclusions and future directions

The consecutively optimized DNN model developed here shows promise for complementing existing models for crop selection and improvement, as it produces more consistent estimates of yield, despite having a slightly higher average error than the best BLUP model. In cases where the accuracy of these models differs with respect to specific regions, genotypes, or other variables, their use together may be an especially valuable future direction. Additionally, the capacity of convolutional neural networks to incorporate change in environmental variables over time is of potential use by enabling the generation of counterfactuals to examine the expected effect of different planting times (shifting the planting date of a site relative to the true value), planting in different sites, or planting under future possible climate scenarios. Additionally, the ability to generate such estimates would enable breeders to consider not only the expected yield of an individual cultivar but the expected *consistency* of yield as well.

For such a strategy to be adopted in genomic selection, further efforts are needed to validate the predictions such a model produces. This will necessitate incorporating of and validation on future data from the G2F initiative (McFarland *et al.* 2020) or other large-scale experiments. The G2F initiative and other organizations sponsor prediction competitions and other activities designed to advance this area of study. Furthermore, applying the same model or the approach used to develop it to other crops would be a valuable step toward assessing its' broad-scale usefulness. This would also potentially implicate groups of crops for which the same model may be used through transfer learning, along with groups that require crop-specific models to be developed.

Additional improvements to accuracy that have the potential to transfer to modeling efforts for other crops include improved heuristics for epoch selection and training set construction. The simultaneously optimized model achieves a minimum error *lower* than our selected model (see Fig. 1b) and does so in far fewer epochs, but overfits much faster as well. If overfitting were preventable through a better heuristic for epoch selection than the one we employed, SO would have produced a better-performing model that was simpler to generate. Training set construction is another opportunity for improvement with transferable utility. Here we took an aggressive approach ensuring approximately balanced groups, down-sampling all groups with observations in excess of the smallest group in the test set. DNNs tend to perform better with an abundance of data, so alternate approaches that retain more observations are of interest. In cases where there are few observations or model development is heavily constrained by computational resources or model development time, other models, especially

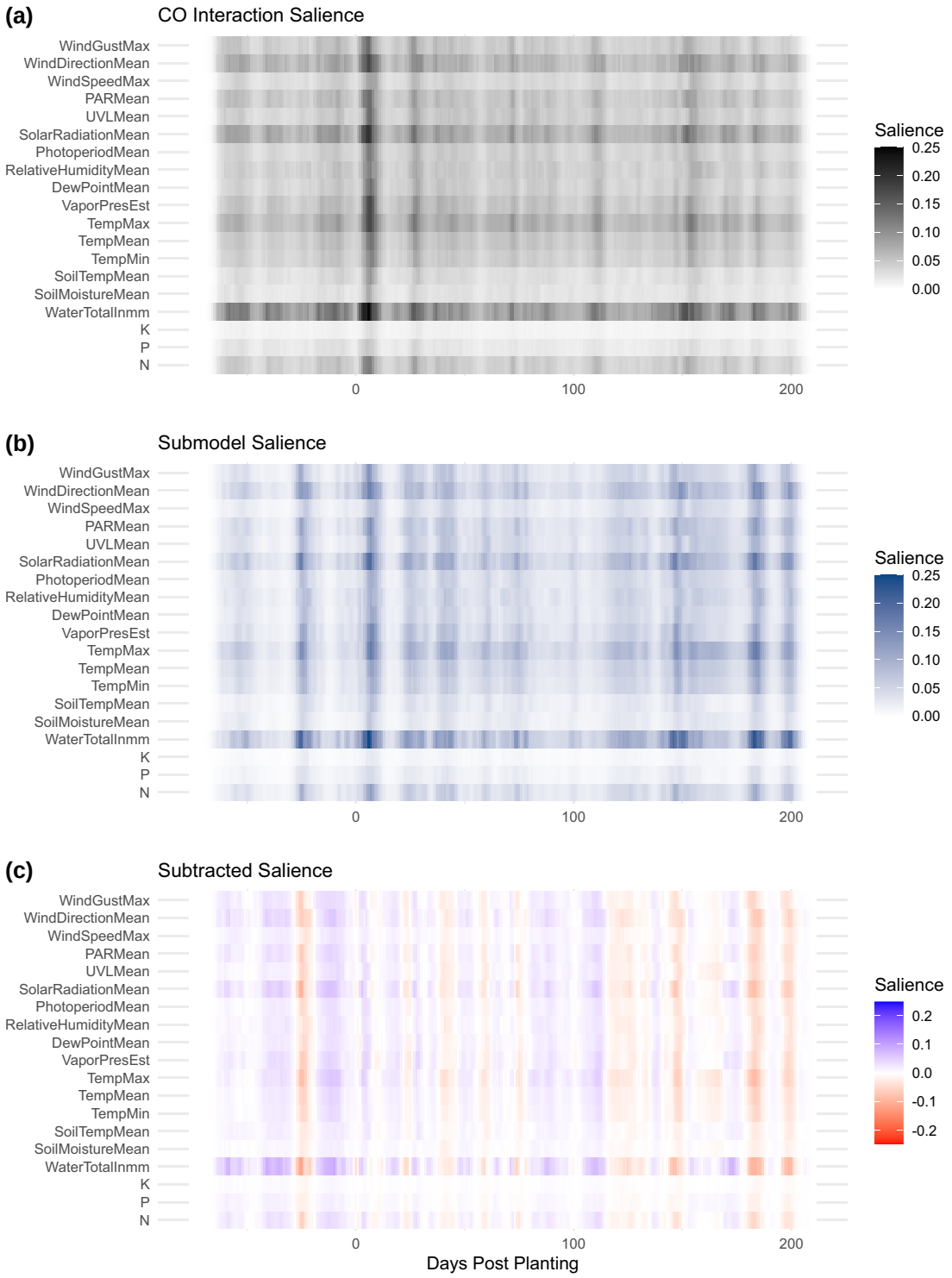

Fig. 3. Influence of interaction effects on feature salience. a) Average salience across all weather and management factors for each day considered. Interaction model values shown in black. b) The same values in (a) are shown for the submodel in blue. Salience peaks shortly after planting in both models. The submodel contains additional peaks of salience prior to planting and near the end of the considered date range. c) Subtracted salience values for the interaction model and the submodel. The interaction-containing model appears to contain greater importance generally for certain features, e.g. irrigation and rainfall, represented as "WaterTotalInmm". The difference between the 2 saliency maps indicates additional times of sensitivity in the submodel (approximately −25, +180, +195) that the interaction model is relatively insensitive to. The variables listed above (from top to bottom) are WindGustMax: daily max wind gust in meters per second, WindDirectionMean: daily mean wind direction in degrees, WindSpeedMax: daily max wind speed in meters per second, PARMean: photosynthetically active radiation in micro-moles per meter-squared seconds, UVLMean: daily mean ultra-violet radiation in micro-moles per meter-squared seconds, SolarRadiationMean: daily mean solar radiation in watts per square meter, PhotoperiodMean: daily mean photoperiod as a percentage, RelativeHumidityMean: daily mean relative humidity as a percentage, DewPointMean: daily mean dew point in degrees Celsius, VaporPresEst: daily estimated water vapor partial pressure in pascals, TempMax: daily max temperature in degrees celsius, TempMean: daily mean temperature in degrees celsius, TempMin: daily minimum temperature in degrees celsius, SoilTempMean: daily mean soil temp in degrees celsius, SoilMoistureMean: daily mean soil moisture in degrees celsius, WaterTotalInmm: total water applied (irrigation and precipitation) in mm, N: nitrogen applied in lbs/acre, P: phosphorus applied in lbs/acre, and K: potassium applied in lbs/acre.

linear regression models, may result in a model that performs nearly as well as one which requires more resources to fit.

Deep learning models do not result in parameters which are as readily interpretable as those of more standard statistical procedures and do not incorporate the physiology of the plant as mechanistic crop growth models do. These represent ongoing challenges and limit the scenarios in which a DNN may be useful. This can be partially addressed through how the data is represented (e.g. using non-PC transformed data), which has been explored for the identification of genetic loci (Liu *et al.* 2019). Additionally, efforts to incorporate known relationships into a deep learning model's structure have the potential to benefit accuracy and interpretability. Improvements in the capacity to represent genetic or physiological principles could allow for these methods to apply to a wider range of uses and address a broader set of questions.

Data availability statement

Data for maize phenotypes, genotypes, field site soil properties, and on-location weather recordings from the G2F initiative data (McFarland *et al.* 2020) are publicly available through the CyVerse Discovery Environment. We used data from 2014 to 2019 which correspond to the following DOIs: 2014–2017 (0.25739/frmv-wj25), 2018 (10.25739/anqq-sg86), and 2019 (10.25739/t651-yy97). Additional genomic data were provided by Natalia de Leon, Dayane Lima, and Cinta Romay worked with Joseph Gage through personal communication, and are available on NCBI as BioProject: PRJNA894503. A version of these data containing all genomic data used in this study is available through Zenodo (zenodo.org/record/6916775 DOI 10.5281/zenodo.6916775). Additional weather measurements were retrieved from Daymet (Thornton *et al.* 2020). Custom python scripts for downloading, aggregating, and processing these data are available through Zenodo (zenodo.org/record/7401113 DOI 10.5281/zenodo.7401113).

Supplemental material available at G3 online.

Acknowledgments

This research used resources provided by the SCINet project of the USDA Agricultural Research Service, ARS project number 0500-00093-001-00-D. In addition to the contributions listed for the authors we would like to acknowledge those presently and historically involved in the G2F initiative, especially the following: Tim Bessinger, Martin Bohn, Edward Buckler, Natalia DeLeon, Jode Edwards, Sherry Flint-Garcia, Candice Hirsch, James Holland, Beth Hood, David Hooker, Shawn Kaeppler, Joseph Knoll, Sanzchen Liu, John McKay, Richard Minyo, Seth Murray, Rebecca Nelson, James Schnable, Rajan Sekhon, Maninder Singh, Peter Thomison, Addie Thompson, Mitch Tuinstra, Jason Wallace, Randy Wisser, and Wenwei Xu, who co-ordinated data collection during 2018 and 2019. Joseph Gage and Cinta Romay produced genotypic data. Alejandro Castro Aviles, Jode Edwards, David Ertl, Joseph Gage, James Holland, Dayane Cristina Lima, Bridget A McFarland, Christina Poudyal, Anna Rogers, Cinta Romay, Luis Samayoa, Kevin Silverstein, Tyson Swetnam, and Jacob Washburn curated the 2018 data. Ryan Timothy Alpers, Alejandro Castro Aviles, James Holland, Dayane Cristina Lima, and Bridget A. McFarland curated the 2019 data. Jode Edwards distributed seeds for the experiments from 2014 to 2017. Tecle Weldekidan made additional contributions to the project. Natalia de Leon, Dayane Lima, and Cinta Romay worked with Joseph Gage in production of genomic data.

Funding

This project was funded by USDA Agricultural Research Service, ARS project number 5070-21000-041-000-D and enabled through computational resources funded through USDA Agricultural Research Service, ARS project number 0500-00093-001-00-D. The G2F initiative was also supported by funding from the Nebraska Corn Board (project ID #: 88-R-1617-03), Iowa Corn Promotion Board, Georgia Agricultural Commodity Commission for Corn, the Corn Marketing Program of Michigan, and National Corn Growers Association.

Conflicts of interest

None declared.

Author contributions

G2F experiments were coordinated and designed by Natalia DeLeon, David Ertl, Judith Kolkman, Dayane Cristina Lima, Danilo E. Moreta, James Schnable, and Maninder Singh. Field experiments were conducted, and data were collected and curated by Barış Alaca, Tim Bessinger, Natalia DeLeon, Jode Edwards, David Ertl, Sherry Flint-Garcia, Candice Hirsch, Joseph Knoll, Judith Kolkman, Dayane Cristina Lima, Danilo E. Moreta, James Schnable, Maninder Singh, Addie Thompson, Jason Wallace, Jacob Washburn, and Tecle Weldekidan. Joseph Gage provided novel genomic data. Jode Edwards distributed seed for the experiments conducted in 2014–2017. The computational study was designed by Daniel Kick and Jacob Washburn. Additional data cleaning and imputation was done by Daniel Kick, who developed the models and generated the figures. The manuscript was written by Daniel Kick and edited by Jacob Washburn, Jason Wallace, James Schnable, Judith Kolkman, and Daniel Kick.

Literature cited

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.

Anaconda Software Distribution. Anaconda Documentation. 2021.

Bache SM, Wickham H. magrittr: A Forward-Pipe Operator for R. 2020.

Bergstra J, Yamins D, Cox D. 2013. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Dasgupta S, McAllester D, editors. Proceedings of the 30th International Conference on Machine Learning. Atlanta, Georgia, USA: Proceedings of Machine Learning Research, PMLR. p. 115–123.

Bradbury PJ, Zhang Z, Kroon DE, Casstevens TM, Ramdoss Y, Buckler ES. TASSEL: software for association mapping of complex traits in diverse samples. Bioinformatics. 2007;23(19):2633–2635. doi:10.1093/bioinformatics/btm308.

Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, *et al.*, 2013. API Design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. p. 108–122.

Chollet F. Keras. 2015.

Couture-Beil A. rjson: JSON for R. 2018.

Da Costa-Luis C, Larroque SK, Altendorf K, Mary HR, *et al.* tqdm: A Fast, Extensible Progress Bar for Python and CLI. Zenodo. 2022.

Fuzzywuzzy. SeatGeek. 2017.

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, *et al.* Array programming with NumPy. Nature. 2020;585(7825):357–362. doi:10.1038/s41586-020-2649-2.

Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw. 1989;2(5): 359–366. doi:10.1016/0893-6080(89)90020-8.

Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng. 2007;9(3):90–95. doi:10.1109/MCSE.2007.55.

Inc PT. Collaborative data science. 2015.

Izrailev S. tictoc: Functions for Timing R Scripts, as Well as Implementations of Stack and List Structures. 2021.

Jarquin D, de Leon N, Romay C, Bohn M, Buckler ES, *et al.* Utility of climatic information via combining ability models to improve genomic prediction for yield within the genomes to fields maize project. Front Genet. 2021;11:592769. doi:10.3389/fgene.2020.592769.

Jarquín D, Crossa J, Lacaze X, Du Cheyron P, Daucourt J, *et al.* A reaction norm model for genomic selection using high-dimensional genomic and environmental data. Theor Appl Genet. 2014; 127(3):595–607. doi:10.1007/s00122-013-2243-1.

Khaki S, Wang L, Archontoulis SV. A CNN-RNN framework for crop yield prediction. Front. Plant Sci. 2020;10:1750. doi:10.3389/fpls.2019.01750.

Kibirige H, Lamp G, Katins J, Gdowding A, *et al.* 2021. has2k1/plotnine: v0.8.0. Zenodo.

Kubota Y, 2021 tf-keras-vis.

Kurtzer GM, Sochat V, Bauer MW. 2017 Singularity: scientific containers for mobility of compute. PLoS One. 12(5): e0177459. doi: 10.1371/journal.pone.0177459.

Li X, Guo T, Wang J, Bekele WA, Sukumaran S, Vanous AE, McNellie JP, Tibbs-Cortes LE, Lopes MS, Lamkey KR, *et al.* An integrated framework reinstating the environmental dimension for GWAS and genomic selection in crops. Mole Plant. 2021;14(6):874–887. doi:10.1016/j.molp.2021.03.010.

Liu J, Goering CE, Tian L. A neural network for setting target corn yields. Trans ASAE. 2001;44(3):705–713.

Liu Y, Wang D, He F, Wang J, Joshi T, Xu D. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. Front Genet. 2019;10:1091. doi:10.3389/fgene.2019.01091.

McFarland BA, AlKhalifah N, Bohn M, Bubert J, Buckler ES, Ciampitti I, Edwards J, Ertl D, Gage JL, Falcon CM, *et al.* 2020 Maize genomes to fields (G2F): 2014–2017 field seasons: genotype, phenotype, climatic, soil, and inbred ear image datasets. BMC Res Notes. 13(1): 71. doi:10.1186/s13104-020-4922-8.

Messina CD, Technow F, Tang T, Totir R, Gho C, *et al.* Leveraging biological insight and environmental variation to improve phenotypic prediction: integrating crop growth models (CGM) with whole genome prediction (WGP). European Journal of Agronomy. 2018;100:151–162. doi:10.1016/j.eja.2018.01.007.

Müller K. here: A Simpler Way to Find Your Files. 2020.

O'Malley TE, Bursztein J, Long F, Chollet H Jin L, *et al.* KerasTuner.2019.

Pedersen TL. patchwork: The Composer of Plots. 2020.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, *et al.* Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12(85):2825–2830.

Pérez-Rodríguez P, de los Campos G. Multitrait Bayesian shrinkage and variable selection models with the BGLR-R package. Genetics. 2022;222(1):iyac112. doi:10.1093/genetics/iyac112.

Perez P, de los Campos G. Genome-Wide regression and prediction with the BGLR statistical package. Genetics. 2014;198(2): 483–495. doi:10.1534/genetics.114.164442.

R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2021.

Richardson N, Cook I, Crane N, Keane J, François R, *et al.* arrow: Integration to "Apache" "Arrow". 2021.

Rogers AR, Dunne JC, Romay C, Bohn M, Buckler ES, Ciampitti IA, Edwards J, Ertl D, Flint-Garcia S, Gore MA, *et al.* The importance of dominance and genotype-by-environment interactions on grain yield variation in a large-scale public cooperative maize experiment. G3 (Bethesda). 2021;11(2):jkaa050. doi:10.1093/g3journal/jkaa050.

Rogers AR, Holland JB. Environment-specific genomic prediction ability in maize using environmental covariates depends on environmental similarity to training data. G3 (Bethesda). 2021:(2): jkab440. doi:10.1093/g3journal/jkab440

Samek W, Wiegand T, Müller K-R. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. 2017.

Seabold S, Perktold J. statsmodels: Econometric and statistical modeling with python, in 9th Python in Science Conference. 2010.

Shahhosseini M, Hu G, Huber I, Archontoulis SV. Coupling machine learning and crop modeling improves crop yield prediction in the US corn belt. Sci Rep. 2021;11(1):1606. doi:10.1038/s41598-020-80820-1.

Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 2014.

SingularityCE Developers. SingularityCE 3.8.3. Zenodo. 2021.

Tavenard R, Faouzi J, Vandewiele G, Divo F, Androz G, *et al.* Tslearn, A machine learning toolkit for time series data. J Mach Learn Res. 2020;21(118):1–6.

Team Pandas Development. pandas-dev/pandas: Pandas. Zenodo. 2020.

Technow F, Messina CD, Totir LR, Cooper M. Integrating crop growth models with whole genome prediction through approximate Bayesian computation. PLoS One. 2015;10(6): e0130855. doi:10.1371/journal.pone.0130855.

Techtonik A. wget 3.2. 2015.

Thornton MM, Shrestha R, Wei Y, Thornton PE, Kao S, *et al.* Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 4. 2020.

Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley (CA): CreateSpace; 2009.

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, *et al.* Scipy 1.0: fundamental algorithms for scientific computing in python. Nat Methods. 2020;17(3):261–272. doi:10.1038/s41592-019-0686-2.

Washburn JD, Cimen E, Ramstein G, Reeves T, O'Briant P, McLean G, Cooper M, Hammer G, Buckler ES. Predicting phenotypes from genetic, environment, management, and historical data using CNNs. Theor Appl Genet. 2021;134(12):3997–4011. doi:10.1007/s00122-021-03943-7.

Waskom ML. Seaborn: statistical data visualization. J Open Source Softw. 2021;6(60):3021. doi:10.21105/joss.03021.

Westhues CC, *et al.* Prediction of maize phenotypic traits with genomic and environmental predictors using gradient boosting frameworks. Front. Plant Sci. 2021;12:699589. doi:10.3389/fpls.2021.699589.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, *et al.* Welcome to the tidyverse. J Open Source Softw. 2019;4(43):1686. doi:10.21105/joss.01686.

Zhou D-X. Universality of deep convolutional neural networks. Appl Comput Harmon Analy. 2020;48(2):787–794. doi:10.1016/j.acha.2019.06.004.

Editor: J. Jannink