A GNN-RNN Approach for Harnessing Geospatial and Temporal Information: Application to Crop Yield Prediction

Joshua Fan* **Junwen Bai*** **Zhiyun Li*** **Ariel Ortiz-Bobea** **Carla Gomes**

Cornell University
{jyf6,jb2467,zl547}@cornell.edu

Abstract

Climate change poses new challenges to agricultural production, as crop yields are extremely sensitive to climatic variation. Predicting the effects of weather patterns on crop yield is crucial for addressing issues such as food insecurity, supply stability, and economic planning; there have been many attempts to use machine learning to do this. However, these models either restrict their tasks to a relatively small region or a short time-period (e.g. a few years), so they may not generalize spatially and temporally. They also view each location as an i.i.d sample, ignoring spatial correlations in the data. In this paper, we introduce a novel graph-based recurrent neural network for crop yield prediction, which incorporates both geographical and temporal structure. Our method is trained, validated, and tested on over 2000 counties from 41 states in the US mainland, covering years from 1981 to 2019. As far as we know, this is the first machine learning method that embeds geographical knowledge in crop yield prediction and predicts crop yields at the county level nationwide. Experimental results show that our proposed method consistently outperforms a wide variety of existing state-of-the-art methods, validating the effectiveness of geospatial and temporal information.

1 Introduction

Climate change [1] has become a real and pressing challenge that poses many threats to our everyday life. Besides the evident extreme events [2], climatic variations also impact the yields of major crops [3]. Crop production is extremely vulnerable and sensitive to fluctuations in climatic factors such as temperature, precipitation, soil, moisture, and many other factors [4]. Many recent works emphasize the need to adapt agricultural practices in light of climate change [5, 6]; to do this, it is necessary to predict how changes in climate will affect crop yield [7]. Crop yield prediction can help with food security [8], supply stability [9], seed breeding [10], and economic planning [11].

However, crop yield depends on numerous complex factors, including temperature, precipitation, soil moisture, soil type, etc. To model these complex relationships, machine learning methods have been widely adopted in crop yield prediction [12, 13, 14, 15]. However, despite the enormous number of papers in this area, many of them share similar methods. Among around 70 papers we surveyed, 48 used neural networks, 10 used tree-based methods, and 10 used linear regressions (e.g. lasso). These methods often only differ in location (US, Brazil, India), study granularity (province, county, site/farm), crop types (soybean, corn), and time range (weeks to years). Similar findings are also reported in [7]. For many relatively small self-collected datasets, simpler models are preferred. But these models may not perform well on a large and diverse region like the entire US.

*Equal contribution.

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

In addition, most machine learning methods used in prior work treat each county as an i.i.d. sample, which does not fully utilize the spatial structure of a larger region. For instance, if one county has a splendid harvest, the neighboring counties are likely to have high yields as well, which violates the independence assumption. We hence introduce **graph neural networks** (GNN) [16] to take into account the geographical relationships among counties. When the model makes a prediction for a county, it can combine the features from neighboring counties with its own features to boost the predictive power. GNN models have been successful in many tasks such as election prediction [17] and COVID forecasting [18]. In this paper, we propose a novel graph-based framework, **GNN-RNN**, which integrates both geospatial and temporal knowledge into crop yield prediction by using GNNs synergistically with RNNs.

We compare our GNN-RNN method with a wide array of machine learning methods on a nationwide scale (over 2000 counties from 41 US states). We show that our novel **GNN-RNN** model can achieve superior performance on multiple nationwide benchmarks, consistently outperforming existing methods. On both RMSE and R^2, our GNN-RNN outperforms the state-of-the-art CNN-RNN model by 10%. As far as we know, our work is the first to incorporate geographical knowledge into crop yield prediction, and we show that doing so significantly improves prediction accuracy.

2 Methods

2.1 Problem Formulation

In crop yield prediction, we denote each county's climatic features by $\mathbf{x}_{c,t}$ and ground-truth crop yield by $y_{c,t} \in \mathbb{R}$, where c, t represent county and year respectively. Each $\mathbf{x}_{c,t}$ contains four types of features (descriptions of these features can be found in the Appendix): weather features $\mathbf{x}^w_{c,t} \in \mathbb{R}^{n_w \times 52}$, land surface features $\mathbf{x}^l_{c,t} \in \mathbb{R}^{n_l \times 52}$, soil quality features $\mathbf{x}^s_c \in \mathbb{R}^{n_s \times 6}$, and some extra features (e.g. crop production index) $\mathbf{x}^e_c \in \mathbb{R}^{n_e}$. Namely, $\mathbf{x}_{c,t} = (\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}, \mathbf{x}^s_c, \mathbf{x}^e_c)$. n_w, n_l, n_s, n_e denote the number of weather, land surface, soil quality and extra variables respectively. Among these features, $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}$ change both spatially and temporally, while $\mathbf{x}^s_c, \mathbf{x}^e_c$ are county-specific and remain stable over time. $y_{c,t}$ is the crop (e.g. corn, soybean) yield for an entire year. The goal is to predict $y_{c,t}$ given $\mathbf{x}_{c,t}$. Recent work [14] also showed features from past years can help with the prediction, so we reformulate our task as predicting $y_{c,t}$ with $\{\mathbf{x}_{c,t}, \mathbf{x}_{c,t-1}, ..., \mathbf{x}_{c,t-\Delta t}\}$. Δt is the length of year dependency. If $\Delta t = 0$, the model will not consider features from prior years.

2.2 Per-Year Embedding Extraction

Our first step is to extract an embedding for each year from the raw features $\mathbf{x}_{c,t}$. The four types of features $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}, \mathbf{x}^s_c, \mathbf{x}^e_c$ have different structures. For example, weekly features $\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}$ naturally incorporate a temporal order, but county-specific soil features $\mathbf{x}^s_c$ do not change temporally and are measured at different depths underground. Therefore, we use separate neural networks to process the differently-structured parts from $\mathbf{x}_{c,t}$:

$$\mathbf{h}^{wl}_{c,t} = f_{wl}(\mathbf{x}^w_{c,t}, \mathbf{x}^l_{c,t}) \qquad \mathbf{h}^s_c = f_s(\mathbf{x}^s_c) \qquad \mathbf{h}_{c,t} = (\mathbf{h}^{wl}_{c,t}, \mathbf{h}^s_c, \mathbf{x}^e_c) \tag{1}$$

$f_{wl}(\cdot)$ handles features that vary over time, including weather ($\mathbf{x}^w_{c,t}$) and related land surface features such as soil moisture ($\mathbf{x}^l_{c,t}$). An RNN or a CNN can be used for f_{wl} to facilitate information aggregation along the time axis. On the other hand, $f_s(\cdot)$ aggregates information along the soil depth axis. We use CNN as the architecture for f_s. $\mathbf{x}^e_c$ only contains six scalar values, so we directly pass it to the output embedding. The final embedding $\mathbf{h}_{c,t}$ is the concatenation of $\mathbf{h}^{wl}_{c,t}, \mathbf{h}^s_c, \mathbf{x}^e_c$.

2.3 Temporal Dependency

Although yields primarily depend on climatic factors within one year, it has been observed that the trend captured by recent history can be very informative for prediction [14]. For example, crop yields have tended to increase over the past few decades due to improvements in technology and genetics [19]. Thus, we use another RNN that reads the per-year embeddings from the current year and several prior years. The output from the last time step is our prediction for the crop yield of the current year:

$$\widehat{y}_{c,t} = r(\mathbf{h}_{c,t-\Delta t}, ..., \mathbf{h}_{c,t-1}, \mathbf{h}_{c,t}) \tag{2}$$

where $r(\cdot)$ is an RNN, and $\mathbf{h}_{c,t'}$ is the embedding from year t' for county c. The model described so far follows the CNN-RNN framework, which has previously been shown to outperform single-year NN models.

2.4 Incorporating Geographical Knowledge with Graph Neural Networks

Eq. 2 shows how one can extend the use of embeddings from Eq. 1 temporally. We would also like to take advantage of the geospatial structure in the data. Intuitively, if some county has good yields, nearby counties tend to have good yields as well. The weather and soil conditions should also transition smoothly across the continent. The additional features from neighboring counties could boost the prediction if used properly.

Graph Neural Network (GNN) [20] is a novel type of neural network proposed to unravel the complicated dependencies inherent in graph-structured data sources. GNN has been successfully applied to problems in chemistry [21], traffic [22], biology [23], computer vision [24] with sophisticated model architectures [25, 26, 27]. Formally, a graph is denoted by $G = (V, E)$, where V is the set of nodes and E is the set of edges between nodes. In our crop yield prediction task, each node is a county. E is represented as a symmetric adjacency matrix $A \in \{0, 1\}^{N \times N}$ where $A_{i,j} = 1$ if two counties $v_i, v_j \in V$ border and $A_{i,j} = 0$ otherwise. N is the total number of counties. Each node is associated with $\mathbf{x}_{c,t}$ for every year.

In this paper, we use a popular GNN model, GraphSAGE [26]. GraphSAGE is a framework that leverages node feature information and learns node embeddings through aggregation from a node's local neighborhood. Unlike many other methods based on matrix factorization and normalization [17], GraphSAGE simply aggregates the features from a local neighborhood, and is thus less computationally expensive. GraphSAGE is suitable for crop yield prediction because most counties only border a few others and the adjacency matrix is sparse.

Formally, for the l-th layer of GraphSAGE,

$$\begin{aligned} \mathbf{a}_{c,t}^{(l)} &= g_l(\{\mathbf{z}_{c',t}^{(l-1)}, \forall c' \in \mathcal{N}(c)\}) \\ \mathbf{z}_{c,t}^{(l)} &= \sigma(\mathbf{W}^{(l)} \cdot (\mathbf{z}_{c,t}^{(l-1)}, \mathbf{a}_{c,t}^{(l)})) \end{aligned} \tag{3}$$

where $\mathbf{z}_{c,t}^{(0)} = \mathbf{h}_{c,t}$ from Eq. 1, and $l \in \{0, 1, ..., L\}$. $\mathcal{N}(c) = \{c', \forall A_{c,c'} = 1\}$ is the set of neighboring counties for c. $g_l(\cdot)$ is the aggregation function for the l-th layer, which could be mean, pooling, or graph convolution (GCN) function. In practice, we found mean or pooling are effective and computationally efficient. $\mathbf{a}_{c,t}^{(l)}$ is the aggregated embedding from the bordering counties. We concatenate $\mathbf{a}_{c,t}^{(l)}$ with last layer's embedding $\mathbf{z}_{c,t}^{(l-1)}$ before the transformation using $\mathbf{W}^{(l)}$. $\sigma(\cdot)$ is a non-linear function.

2.4.1 GNN-RNN

The output embedding from GNN's last layer $\mathbf{z}_{c,t}^{(L)}$ thus extracts the information (e.g., weather, soil) from the whole local neighborhood for year t. To integrate historical knowledge, we can do the same as in Eq. 2, by taking the GNN output embeddings from prior years:

$$\widehat{y}_{c,t} = r(\mathbf{z}_{c,t-\Delta t}^{(L)}, ..., \mathbf{z}_{c,t-1}^{(L)}, \mathbf{z}_{c,t}^{(L)}) \tag{4}$$

where $\mathbf{z}_{c,t'}^{(L)}$ is the GNN embedding from year t'. We use log-cosh function as our objective:

$$L(\widehat{y}_{c,t}, y_{c,t}) = \log(\cosh(\widehat{y}_{c,t} - y_{c,t})) \tag{5}$$

Log-cosh works similarly to mean square error, but is not as strongly affected by the occasional wildly incorrect prediction. It is also twice differentiable everywhere. We optimize model parameters end-to-end with backpropagation.

3 Experiments

We compare 11 representative machine learning models, including GNN and GNN-RNN, on US county-level crop yields for corn and soybean. We use a variety of climate, land surface (e.g. soil

moisture), and soil quality variables as input features to our model; more dataset details can be found in the Appendix. The performance on three metrics will be shown: RMSE, R^2, and correlation coefficient. Given a test year t, we use year $t-1$ for validation and all the prior years for training.

Methods considered. We consider two types of methods: **(a) single-year methods** that only use features from year t to predict yield for the same year t, and **(b) 5-year methods** that use features from a 5-year series (years $\{t-4, t-3, \ldots, t\}$) to predict yield for year t. For single-year methods, we select lasso, ridge regressor and gradient boosting regressor as non-deep baselines; these operate on the flattened feature vector for one year. Next, we tried three deep learning architectures for $f_{wl}(\cdot)$: LSTM [28], GRU [29], and 1-D CNN [30]. These methods process the weekly time-series of weather and land surface data within the year. We compare these methods with our single-year GNN model (Eq. 3), which incorporates geospatial context in making predictions. For history-dependent models, we follow [14] by considering a 5-year dependency. Two baseline models using LSTM and GRU respectively handle the raw flattened feature vectors for each year $\{\mathbf{x}_{c,t-\Delta t}, ..., \mathbf{x}_{c,t}\}$ directly with $r(\cdot)$. The most recent CNN-RNN model [14] pre-processes the raw features with CNN (choosing CNN for $f_{wl}(\cdot)$) and then uses a LSTM to model the sequence embeddings as described in Eq. 2. Finally, the GNN-RNN model proposed in this paper (Eq. 4) still uses a CNN for $f_{wl}(\cdot)$ to encode the raw features into an embedding for each year, then uses the GNN to refine the embeddings using information from the county's spatial context, and then passes those embeddings into an LSTM. The appendix contains details on model architecture and training.

Table 1: Evaluation results, 2019 soybean. For RMSE, lower is better; for R^2 and Corr, higher is better. We grouped the methods based on whether they use 1 year of data (1y) or 5 years of data (5y) to make predictions. "gbr" is Gradient Boosting Regressor.

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.5731	0.6137	0.8089
ridge 1y	0.6069	0.5668	0.7944
gbr 1y	0.6802	0.4558	0.7899
gru 1y	0.5742	0.5150	0.7569
lstm 1y	0.5907	0.4867	0.7195
cnn 1y	0.5699	0.5222	0.7385
gnn 1y **(ours)**	**0.4916**	**0.7148**	**0.8505**
gru 5y	0.5751	0.6109	0.8158
lstm 5y	0.5512	0.6427	0.8156
cnn-rnn 5y	0.5365	0.6615	0.8423
gnn-rnn 5y **(ours)**	**0.4745**	**0.7349**	**0.8602**
(std)	*(0.0160)*	*(0.0179)*	*(0.0076)*

Results. We evaluated the model on four test datasets: 2018 corn, 2018 soybean, 2019 corn, and 2019 soybean. The results on 2019 soybean are shown in Table 1; other tables and illustrative maps are shown in the Appendix (Table 13; Figures 3, 4, 5, 6). For the methods that only use 1 year when making predictions, our GNN model clearly outperforms comparable baselines across all datasets and metrics (except for 2018 soybean Corr, where it is slightly worse than GRU). For the methods that use a history of 5 years, our GNN-RNN outperforms competing baselines in almost all cases (except for 2018 soybean Corr, where it is slightly worse than CNN-RNN). For example, in 2019, our prediction R^2 score outperforms the state-of-the-art CNN-RNN method [14] by 16% on corn and 11% on soybean (relative). On average, we achieve a relative R^2 improvement of 10.44% over the recent CNN-RNN model, 16.16% over the 5-year LSTM, and a relative RMSE improvement of 9.6% over the CNN-RNN model, 13.18% over the 5-year LSTM. These indicate the importance of exploiting geospatial context in making these predictions.

4 Conclusion

In this paper, we propose a novel GNN-RNN framework to innovatively incorporate both geospatial and temporal knowledge into crop yield prediction, through graph-based deep learning methods. To our knowledge, our paper is the first to take advantage of the spatial structure in the data when making crop yield predictions, as opposed to previous approaches which assume that neighboring counties are independent samples. We conduct extensive experiments on large-scale datasets covering 41 US states and 39 years, and show that our approach substantially outperforms many existing state-of-the-art machine learning methods across multiple datasets. Thus, we demonstrate that incorporating knowledge about a county's geospatial neighborhood and recent history can significantly enhance the prediction accuracy of deep learning methods for crop yield prediction.

Acknowledgments and Disclosure of Funding

This research was supported by USDA Cooperative Agreement 58-6000-9-0041 and USDA NIFA Hatch Project 1017421. We would like to thank Rich Bernstein for proofreading and Samuel Porter for help in processing the gSSURGO dataset.

References

[1] JT Houghton, GJ Jenkins, JJ Ephraums, et al. Climate change. Technical report, Cambridge, GB: Cambridge University Press, 1990.

[2] Kevin E Trenberth, John T Fasullo, and Theodore G Shepherd. Attribution of climate extreme events. *Nature Climate Change*, 5(8):725–730, 2015.

[3] Chuang Zhao, Bing Liu, Shilong Piao, Xuhui Wang, David B Lobell, Yao Huang, Mengtian Huang, Yitong Yao, Simona Bassu, Philippe Ciais, et al. Temperature increase reduces global yields of major crops in four independent estimates. *Proceedings of the National Academy of Sciences*, 114(35):9326–9331, 2017.

[4] Ariel Ortiz-Bobea, Erwin Knippenberg, and Robert G Chambers. Growing climatic sensitivity of us agriculture linked to technological change and regional specialization. *Science advances*, 4(12):eaat4343, 2018.

[5] MP Reynolds, R Ortiz, et al. Adapting crops to climate change: a summary. *Climate change and crop production*, pages 1–8, 2010.

[6] Ali Raza, Ali Razzaq, Sundas Saher Mehmood, Xiling Zou, Xuekun Zhang, Yan Lv, and Jinsong Xu. Impact of climate change on crops adaptation and strategies to tackle its outcome: A review. *Plants*, 8(2):34, 2019.

[7] Thomas Van Klompenburg, Ayalew Kassahun, and Cagatay Catal. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177:105709, 2020.

[8] PR Shukla, J Skeg, E Calvo Buendia, V Masson-Delmotte, H-O Pörtner, DC Roberts, P Zhai, R Slade, S Connors, S van Diemen, et al. Climate change and land: an ipcc special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems. *Intergovernmental Panel on Climate Change (IPCC)*, 2019.

[9] Rachael D Garrett, Eric F Lambin, and Rosamond L Naylor. Land institutions and supply chain configurations as determinants of soybean planted area and yields in brazil. *Land Use Policy*, 31:385–396, 2013.

[10] Javad Ansarifar, Faezeh Akhavizadegan, and Lizhi Wang. Performance prediction of crosses in plant breeding through genotype by environment interactions. *Scientific Reports*, 10(1):1–11, 2020.

[11] T Horie, M Yajima, and H Nakagawa. Yield forecasting. *Agricultural systems*, 40(1-3):211–236, 1992.

[12] Oskar Marko, Sanja Brdar, Marko Panic, Predrag Lugonja, and Vladimir Crnojevic. Soybean varieties portfolio optimisation based on yield prediction. *Computers and Electronics in Agriculture*, 127:467–474, 2016.

[13] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI conference on artificial intelligence*, 2017.

[14] Saeed Khaki et al. A cnn-rnn framework for crop yield prediction. *Frontiers in Plant Science*, 10:1750, 2020.

[15] Saeed Khaki, Zahra Khalilzadeh, and Lizhi Wang. Predicting yield performance of parents in plant breeding: A neural collaborative filtering approach. *Plos one*, 15(5):e0233382, 2020.

[16] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[17] Junteng Jia and Austion R Benson. Residual correlation in graph neural network regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 588–598, 2020.

[18] Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113*, 2020.

[19] Ariel Ortiz-Bobea and Jesse Tack. Is another genetic revolution needed to offset climate change impacts for us maize yields? *Environmental Research Letters*, 13(12):124009, 2018.

[20] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[21] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[22] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

[23] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in Neural Information Processing Systems*, 30:6530–6539, 2017.

[24] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018.

[25] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[26] Will Hamilton et al. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[29] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[30] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[31] USDA. National agricultural statistics service. *United States Department of Agriculture*, 2013.

[32] Christopher Daly and Kirk Bryant. The prism climate and weather system: an introduction. 2013.

[33] Youlong Xia, Kenneth Mitchell, Michael Ek, Justin Sheffield, Brian Cosgrove, Eric Wood, Lifeng Luo, Charles Alonge, Helin Wei, Jesse Meng, et al. Continental-scale water and energy flux analysis and validation for the north american land data assimilation system project phase 2 (nldas-2): 1. intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, 117(D3), 2012.

[34] Soil Survey Staff. Gridded soil survey geographic (gssurgo) database for the conterminous united states., 2020.

[35] Soil Survey. Soil texture calculator, 2021.

[36] C.H. Homer, J.A. Fry, and C.A. Barnes. The national land cover database, 2012.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Supplemental Material

A.1 Dataset Details

Crop yield labels for corn and soybean are available from the USDA Crop Production Reports [31] for numerous counties in the US. Due to budget limitations, not all counties report data in every year, but the coverage is still quite comprehensive. For example, for corn, all years between 1981 and 2003 have over 2,000 counties across 41 states reporting data. We train and evaluate our model on all counties in 41 states where yield data is available. (When computing the loss, we ignore counties that do not have yield labels for that year.)

We use a variety of climate, land surface, and soil quality variables as input features to our model; these features are available for almost all counties in the contiguous 48 US states (3,107 counties in total[2]). We draw 7 weather features from the PRISM climate mapping system [32]: precipitation, min/mean/max temperature, min/max vapor pressure deficit, and mean dewpoint temperature. These features are available at a 4×4 km grid for each day.

We acquire 16 land surface features from the North American Land Data Assimilation System (NLDAS) [33], which is a large-scale land surface model that closely simulates land surface parameters. These features include soil moisture content, moisture availability, and soil temperature (all at various soil depths), as well as observed weather variables such as wind speed and humidity. These variables are available at a 0.125×0.125 degree (~ 14 km) spatial resolution, every hour.

Soil quality features were acquired from the Gridded Soil Survey Geographic Database (gSSURGO) [34], at a 30×30 meter resolution. These features include available water capacity, bulk density, and electrical conductivity, pH, and organic matter. Unlike the weather and land surface features, the gSSURGO soil quality features are fixed and *do not change over time*. In addition to the raw features, we use the raw sand, silt, and clay percentages to compute the "soil texture type" of each pixel based on the Natural Resources Conservation Service Soil Survey's classification scheme [35], and then compute the fraction of each county occupied by each soil texture type. In total, we have a total of 20 gSSURGO variables that are depth-dependent (so there are values for 6 different soil depth levels), and 6 "extra" variables which are not depth-dependent (such as crop productivity indices). A full list of the features can be found in the Appendix.

All of these datasets were originally available as gridded raster data at a variety of spatial resolutions. We aggregated each feature to the county level by computing the weighted average of the variable over all grid cells that overlap with the county. Each grid cell is weighted by the percentage of the cell that lies inside the county, multiplied by the percentage of that grid cell which is cropland, pasture, or grassland. We used the National Land Cover Database [36], which is available at a 30m resolution, to compute the percentage of each cell that is covered by cropland, pasture, or grassland. An example of this aggregation process is depicted in Figure 1. In addition, the time-dependent variables (weather and land surface) were aggregated from daily to weekly frequency to make the prediction task more tractable.

A full list of features is provided below.

Weather features ($\mathbf{x}_{c,t}^{w}$) come from the PRISM dataset [32], with an original spatial resolution of 4 km and a temporal resolution of daily:

1. Precipitation
2. Mean dewpoint temperature
3. Daily max temperature
4. Daily mean temperature
5. Daily minimum temperature
6. Max vapor pressure deficit
7. Min vapor pressure deficit

[2]The only exception is Nantucket County, Massachusetts, where land surface model data is missing, since it is an offshore island. Also note that some counties have feature data but not label (yield) data. Only the GNN and GNN-RNN models can make use of these unlabeled county features.

Land surface features ($\mathbf{x}^l_{c,t}$) come from the NLDAS land surface model [33], with an original spatial resolution of 0.125 degrees (14 km) and a temporal resolution of hourly:

1. Precipitation hourly total (kg/m^2)
2. Moisture availability (%), 0-200 cm
3. Moisture availability (%), 0-100 cm
4. Soil moisture content (kg/m^2), 0-200cm
5. Soil moisture content (kg/m^2), 0-100cm
6. Soil moisture content (kg/m^2), 0-10cm
7. Soil moisture content (kg/m^2), 10-40cm
8. Soil moisture content (kg/m^2), 40-100cm
9. Soil moisture content (kg/m^2), 100-200cm
10. 2-m above ground specific humidity (kg/kg)
11. 2-m above ground temperature (K)
12. Soil temperature (K), 0-10 cm
13. Soil temperature (K), 10-40 cm
14. Soil temperature (K), 40-100 cm
15. Soil temperature (K), 100-200 cm
16. Wind speed (m/s), hourly max

(Note that the cm ranges represent depths in the soil.)

Soil quality features ($\mathbf{x}^s_c$) come from the Gridded Soil Survey Geographic Database (gSSURGO) [34]. The dataset has a 30-m spatial resolution for the continental U.S. These variables do not change over time. However, they vary with depths, which are measured at 6 soil depth layers (0-5cm, 5-15cm, 15-30cm, 30-60cm, 60-100cm, 100-200cm). Because soil quality at a given point can vary substantially within a county, accounting for the location of agricultural activity can be critical when constructing appropriate county-level soil variables. Thus, the "weighted-average" technique is especially important here. We aggregate the fine-scale soil data to the county level based on the percentage of each NLCD Land Cover grid cell that was covered by agricultural land (grassland, pasture, cropland) in 2011.

1. Available water capacity of the dominant soil component
2. Bulk density
3. Electrical conductivity of the dominant soil component
4. Organic matter
5. Average % silt
6. Average % clay
7. Average % sand
8. % area covered by Clay soil type
9. % area covered by Silty Clay soil type
10. % area covered by Sandy Clay soil type
11. % area covered by Clay Loam soil type
12. % area covered by Silty Clay Loam soil type
13. % area covered by Sandy Clay Loam soil type
14. % area covered by Loam soil type
15. % area covered by Silt Loam soil type
16. % area covered by Sandy Loam soil type

17. % area covered by Silt Loam soil type
18. % area covered by Loamy Sand soil type
19. % area covered by Sand soil type
20. pH, which is influenced by chemical reactions between water and the dominant soil component

Note that features 8-19 were not present in the original gSSURGO dataset. Rather, for each pixel, we used the raw silt, clay, and sand percentages to compute the "soil texture type" of that pixel, based on the National Resources Conservation Service Soil Survey's classification scheme [35]. This classification scheme is depicted in Figure 2. After classifying each pixel's soil texture type, we compute the fraction of each county that is occupied by each soil texture type.

Extra features ($\mathbf{x}_c^e$) also come from the gSSURGO dataset [34], but are not depth-dependent. They are listed below:

1. National commodity crop productivity index
2. Depth to any soil restrictive layer
3. NCCPI crop productivity index for small grains, weighted average
4. NCCPI crop productivity index for corn
5. NCCPI crop productivity index for cotton
6. NCCPI crop productivity index for soybean

A.2 Model Details

For the shallow models (ridge regression, lasso, and gradient boosting regressor), we used scikit-learn's implementations.

For the baseline single-year models, we evaluated using LSTM, GRU, and CNN as $f_{wl}(\cdot)$ to process the weekly weather and land surface data. For CNN, we used a 1-D CNN similar to the one in [14], but we process all weather and land surface parameters together. The CNN contains series of 1D convolutions, ReLUs, and average pooling layers; this sequence is repeated four times. For all methods that use LSTM or GRU, we used PyTorch's implementation with 64 hidden states.

The same CNN is used as the encoder for the weekly weather and land surface data in the CNN-RNN, GNN, and GNN-RNN models. (We also tried using an LSTM as the encoder for the weekly data for these models, but found that this did not improve results.) For all methods except for the 5-year LSTM/GRU, we processed the soil data using another small 1-D CNN (with three convolutional layers, and without average pooling), where the convolutions operate across 6 different soil depths.

For the simple 5-year baseline models (LSTM and GRU), we fed the flattened feature vectors for each year through an LSTM or GRU, followed by a 2-layer fully connected network.

For the GNN and GNN-RNN models, we used the implementation of GraphSAGE from the dgl library; we used a 2-layer GNN, with edge dropout of 0.1. The adjacency graph of US counties is provided by the US Census Bureau. We used stochastic mini-batch training to train the model, where each layer samples 10 neighbors to receive messages from. We tried different aggregation functions and found that the "pooling" approach generally performed best.

For all methods, we use the Adam optimizer [37], sometimes with a mild cosine or step decay. We tried learning rates between 1e-5 and 1e-3, used a weight decay of 1e-5 or 1e-4, and a batch size of 32, 64, or 128. We trained the model for 100 to 200 epochs (until the validation loss clearly stopped improving). We chose the epoch and hyperparameter setting that produced the lowest RMSE on the validation year (the year before the test year). We ran the GNN-RNN model 3 times with different random seeds to evaluate the variance in the results. The Appendix contains more details about hyperparameters.

A.3 Hyperparameter Details

For all methods, we use the Adam optimizer [37]. For the CNN-RNN, GNN, and GNN-RNN methods, we tried many hyperparameter configurations, most intensively on the 2018 corn dataset.

We tried learning rates from {1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4, 1e-3}, used a weight decay of 1e-5 or 1e-4, and a batch size of 32, 64, or 128. We tried using a mild cosine decay (with $\eta_{min} \in$ {1e-5, 1e-6}, $T_0 \in \{34, 100, 200\}$), or step decay (every 25 epochs, $\gamma \in \{0.5, 0.8\}$) for the learning rate scheduler. We trained the model for 100 to 200 epochs (until the validation loss clearly stopped improving). We chose the epoch and hyperparameter setting that produced the lowest RMSE on the validation year (the year before the test year).

For the GNN and GNN-RNN models, we used the implementation of GraphSAGE from the dgl library; we used a 2-layer GNN, with edge dropout of 0.1. We used stochastic mini-batch training to train the model, where each layer samples 10 neighbors to receive messages from. We tried different aggregation functions, such as "mean" and "pooling."

We ran the GNN-RNN model 3 times with random seeds {0, 1, 2} to evaluate the variance in the results. For the baseline models, we used seed 0. The final hyperparameter configurations are listed in tables 2 to 10.

Table 2: CNN-RNN hyperparameters: corn

Hyperparameter	**Value**
Batch size	128
Learning rate	1e-4
LR scheduling	Step (25 epochs, γ = 0.5)
Number of epochs	100
Weight decay	1e-5

Table 3: CNN-RNN hyperparameters: soybeans

Hyperparameter	**Value**
Batch size	128
Learning rate	5e-4
LR scheduling	Step (25 epochs, γ = 0.5)
Number of epochs	100
Weight decay	1e-5

Table 4: GNN hyperparameters: corn, 2018

Hyperparameter	**Value**
Batch size	32
Learning rate	1e-4
LR scheduling	Cosine ($T_0 = 200, \eta_{min} = 10^{-5}$)
Number of epochs	100
Weight decay	1e-5
GNN edge dropout	0.1
GNN aggregator	pool

A.4 Early Prediction

In practice, crop yield predictions are most useful if they can be made well before harvest, as this gives time for markets to adapt, and humanitarian aid to be organized in cases of famine [13]. To simulate this, at test time only, for each county we mask out all weather and land surface features from after June 1 (week 22) of the test year, and replace them with the average values for that county during the training years. Then we pass the masked features through a pre-trained model to obtain predictions. The results for several methods for 2018 corn are presented in Table 12. The graph-based models (GNN and GNN-RNN) clearly outperform competing baselines in this scenario, again illustrating the importance of utilizing geospatial context.

Table 5: GNN hyperparameters: corn, 2019

Hyperparameter	**Value**
Batch size	64
Learning rate	5e-5
LR scheduling	Cosine ($T_0 = 100, \eta_{min} = 10^{-5}$)
Number of epochs	200
Weight decay	1e-5
GNN edge dropout	0
GNN aggregator	mean

Table 6: GNN hyperparameters: soybeans, 2018 and 2019

Hyperparameter	**Value**
Batch size	64
Learning rate	1e-4
LR scheduling	Step (25 epochs, $\gamma = 0.8$)
Number of epochs	100
Weight decay	1e-5
GNN edge dropout	0.1
GNN aggregator	pool

A.5 Evaluation Metrics

We evaluate our model across all counties in the test year with data. We use three standard regression metrics: RMSE, R^2, and Pearson correlation coefficient (Corr).

The RMSE is the square root of the mean squared error between the prediction and the true value:

$$RMSE = \sqrt{\frac{\sum_c (y_c - \widehat{y_c})^2}{N}}$$

where y_c is the true yield for county c, $\widehat{y_c}$ is the model's predicted yield for county c, and N is the total number for counties in the test set with yield data. In this paper, we further divide RMSE by the standard deviation of the current crop's yield (across all years), in order to make the results for different crops comparable.

R^2 is a measure of how much the variation in the data can be explained by the model predictions. Formally,

$$R^2 = 1 - \frac{\sum_c (y_c - \widehat{y_c})^2}{\sum_c (y_c - \bar{y})^2}$$

where $\bar{y}$ is the average yield across the entire test dataset. The top of the fraction is the sum of the squared residuals (difference between true yield and model prediction). The bottom is the total sum of squares (of the difference between the true yield and the average yield across the test dataset), which is proportional to the overall variance of the test data.

The Pearson correlation coefficient (Corr) measures the strength of the linear correlation between the true and predicted values. The correlation between two variables x and y is given as

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Again $\bar{x}$ and $\bar{y}$ are the means of x and y respectively. We let x be the model prediction and y be the true yield.

Table 7: GNN-RNN hyperparameters: corn, 2018

Hyperparameter	**Value**
Batch size	32
Learning rate	5e-5
LR scheduling	Cosine ($T_0 = 100, \eta_{min} = 10^{-6}$)
Number of epochs	100
Weight decay	1e-5
GNN edge dropout	0.1
GNN aggregator	pool

Table 8: GNN-RNN hyperparameters: corn, 2019

Hyperparameter	**Value**
Batch size	32
Learning rate	5e-5
LR scheduling	Cosine ($T_0 = 200, \eta_{min} = 10^{-6}$)
Number of epochs	100
Weight decay	1e-5
GNN edge dropout	0.1
GNN aggregator	pool

A.6 Computing Setup

We ran our code on Python 3.7, using the following libraries: PyTorch 1.8, DGL 0.7.1, NumPy 1.18.5, SciPy 1.2.0. We trained on NVIDIA Tesla V100 GPU with 16GB memory, and used 12 CPU threads for GNN-RNN. The GNN-RNN model takes roughly 8 hours to train for 100 epochs on our full US dataset.

A.7 Additional results and plots

Here are tables, maps and scatter plots showing example results for the GNN-RNN model on the other datasets.

2019 corn: Fig. 3 describes the difference between the ground-truth corn yields for counties in 2019, and our predictions. To demonstrate the similarity, we plot their difference in the bottom figure. As shown in the bottom sub-figure, almost all differences are all close to 0. Fig. 4 shows another plot of true-vs-predicted comparison. All the dots cling to the identity function, which means good prediction results.

2019 soybeans: Fig. 5 and Fig. 6 are similar true-vs-predicted plots for soybeans. The prediction results are also promising.

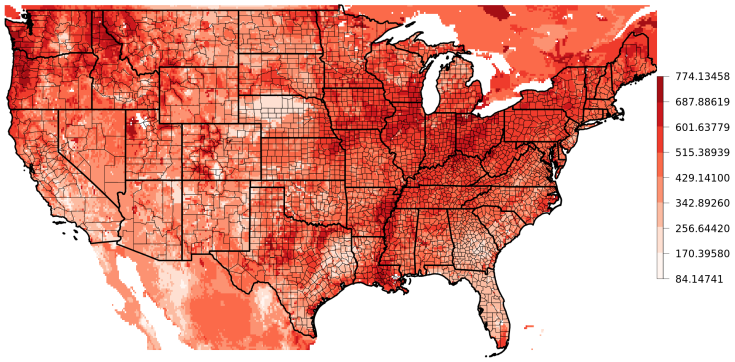
Table 9: GNN-RNN hyperparameters: soybeans, 2018

Hyperparameter	**Value**
Batch size	32
Learning rate	1e-4
LR scheduling	Cosine ($T_0 = 100, \eta_{min} = 10^{-6}$)
Number of epochs	100
Weight decay	1e-4
GNN edge dropout	0.1
GNN aggregator	pool

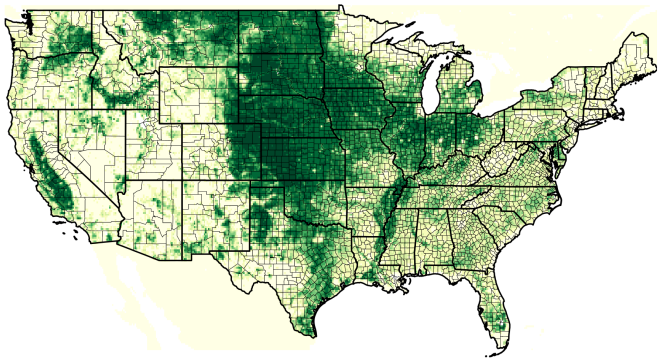
Table 10: GNN-RNN hyperparameters: soybeans, 2019

Hyperparameter	**Value**
Batch size	32
Learning rate	5e-5
LR scheduling	Cosine ($T_0 = 100, \eta_{min} = 10^{-6}$)
Number of epochs	100
Weight decay	1e-5
GNN edge dropout	0.1
GNN aggregator	pool

(a) Original NLDAS raster: SOILM_layer1, date 19810101



(b) NLCD weights (grassland + cropland + pasture)



(c) Aggregated to county-level: SOILM_layer1, date 19810101

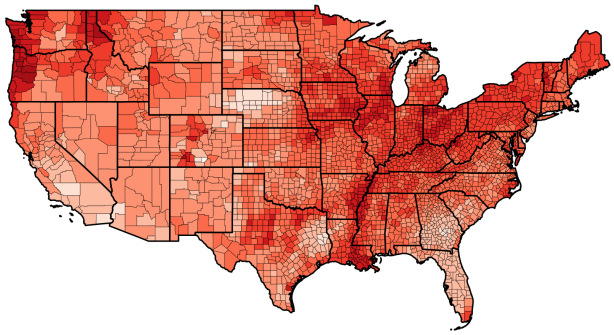


Figure 1: Example of aggregating features to county level.
(a) raw raster of soil moisture from NLDAS.
(b) Percentage cropland/grassland/pasture (used to compute grid cell weights).
(c) the county-level values we generated.

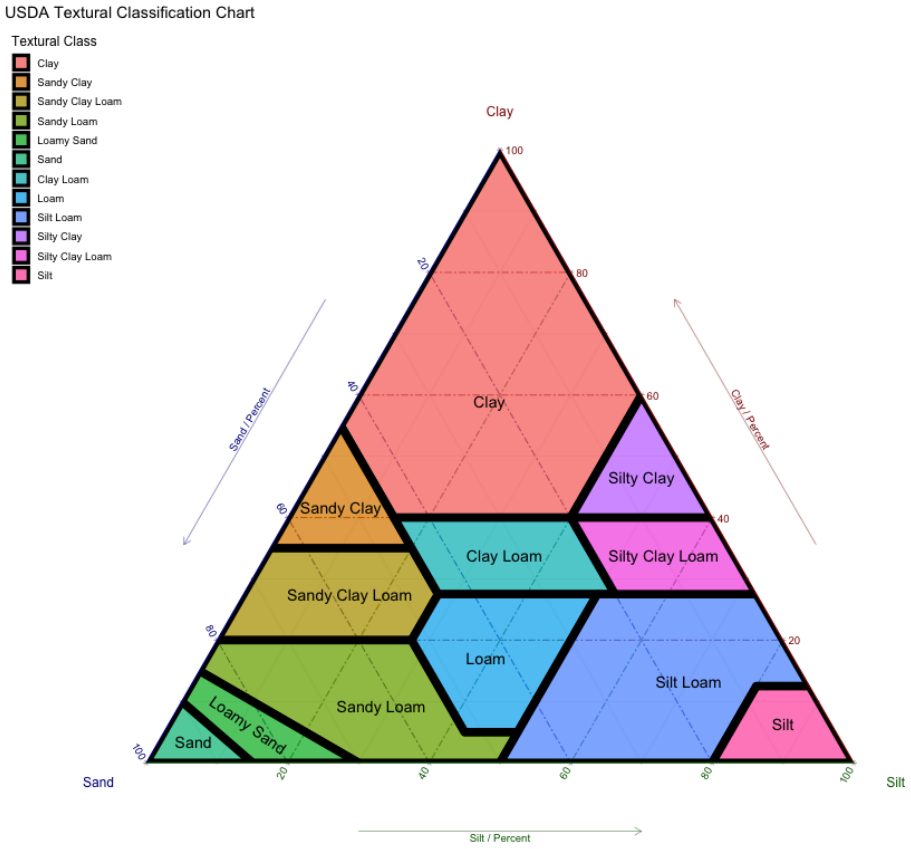


Figure 2: NRCS Soil Texture classification [35]. The three sides of the triangle represent percentage sand, clay, and silt, and the colored regions are the soil texture types.

Table 11: Additional results. For RMSE, lower is better; for R^2 and Corr, higher is better. We grouped the methods based on whether they use 1 year of data (1y) or 5 years of data (5y) to make predictions.

(a) 2018 corn results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.7846	0.3839	0.7778
ridge 1y	0.9255	0.1428	0.7626
gbr 1y	0.7402	0.4516	0.7794
gru 1y	0.5938	0.6472	0.8158
lstm 1y	0.6146	0.6220	0.8303
cnn 1y	0.5824	0.6606	0.8235
gnn 1y **(ours)**	**0.4846**	**0.7517**	**0.8759**
gru 5y	0.6765	0.5419	0.8194
lstm 5y	0.6542	0.5716	0.8060
cnn-rnn 5y	0.5511	0.6936	0.8425
gnn-rnn 5y **(ours)**	**0.4900**	**0.7595**	**0.8731**
(std)	*(0.0191)*	*(0.0186)*	*(0.0092)*

(b) 2019 corn results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.6838	0.3122	0.6715
ridge 1y	0.7081	0.2623	0.6723
gradient-boosting 1y	0.7345	0.2064	0.6857
gru 1y	0.5890	0.4897	0.7381
lstm 1y	0.6245	0.4262	0.7096
cnn 1y	0.5572	0.5432	0.7384
gnn 1y **(ours)**	**0.4930**	**0.6286**	**0.8011**
gru 5y	0.5279	0.5900	0.7785
lstm 5y	0.5311	0.5849	0.7821
cnn-rnn 5y	0.5212	0.5842	0.7868
gnn-rnn 5y **(ours)**	**0.4677**	**0.6782**	**0.8272**
(std)	*(0.0035)*	*(0.0049)*	*(0.0038)*

(c) 2018 soybean results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.6226	0.6090	0.7912
ridge 1y	0.7633	0.4125	0.7550
gradient-boosting 1y	0.6686	0.5492	0.7986
gru 1y	0.6376	0.5932	**0.8356**
lstm 1y	0.6459	0.5825	0.8129
cnn 1y	0.6584	0.5661	0.7988
gnn 1y **(ours)**	**0.5637**	**0.6794**	0.8273
gru 5y	0.6094	0.6254	0.8218
lstm 5y	0.5430	0.7026	0.8459
cnn-rnn 5y	0.5647	0.6784	**0.8650**
gnn-rnn 5y **(ours)**	**0.5333**	**0.7129**	0.8591
(std)	*(0.0194)*	*(0.0206)*	*(0.0049)*

Table 12: Early prediction results (2018 corn, after June 1).

Method	**RMSE**	R^2	**Corr**
lstm 1y	0.6347	0.5968	0.8148
cnn 1y	0.7253	0.4736	0.7004
gnn 1y **(ours)**	**0.5877**	**0.6543**	**0.8124**
lstm 5y	0.7004	0.5091	0.7708
cnn-rnn 5y	0.6532	0.5730	0.7732
gnn-rnn 5y **(ours)**	**0.5836**	**0.6591**	**0.8259**

Table 13: Evaluation results. For RMSE, lower is better; for R^2 and Corr, higher is better. We grouped the methods based on whether they use 1 year of data (1y) or 5 years of data (5y) to make predictions.

(a) 2018 corn results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.7846	0.3839	0.7778
ridge 1y	0.9255	0.1428	0.7626
gbr 1y	0.7402	0.4516	0.7794
gru 1y	0.5938	0.6472	0.8158
lstm 1y	0.6146	0.6220	0.8303
cnn 1y	0.5824	0.6606	0.8235
gnn 1y **(ours)**	**0.4846**	**0.7517**	**0.8759**
gru 5y	0.6765	0.5419	0.8194
lstm 5y	0.6542	0.5716	0.8060
cnn-rnn 5y	0.5511	0.6936	0.8425
gnn-rnn 5y **(ours)**	**0.4900**	**0.7595**	**0.8731**
(std)	*(0.0191)*	*(0.0186)*	*(0.0092)*

(b) 2018 soybean results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.6226	0.6090	0.7912
ridge 1y	0.7633	0.4125	0.7550
gbr 1y	0.6686	0.5492	0.7986
gru 1y	0.6376	0.5932	**0.8356**
lstm 1y	0.6459	0.5825	0.8129
cnn 1y	0.6584	0.5661	0.7988
gnn 1y **(ours)**	**0.5637**	**0.6794**	0.8273
gru 5y	0.6094	0.6254	0.8218
lstm 5y	0.5430	0.7026	0.8459
cnn-rnn 5y	0.5647	0.6784	**0.8650**
gnn-rnn 5y **(ours)**	**0.5333**	**0.7129**	0.8591
(std)	*(0.0194)*	*(0.0206)*	*(0.0049)*

(c) 2019 corn results

Method	**RMSE**	R^2	**Corr**
lasso 1y	0.6838	0.3122	0.6715
ridge 1y	0.7081	0.2623	0.6723
gbr 1y	0.7345	0.2064	0.6857
gru 1y	0.5890	0.4897	0.7381
lstm 1y	0.6245	0.4262	0.7096
cnn 1y	0.5572	0.5432	0.7384
gnn 1y **(ours)**	**0.4930**	**0.6286**	**0.8011**
gru 5y	0.5279	0.5900	0.7785
lstm 5y	0.5311	0.5849	0.7821
cnn-rnn 5y	0.5212	0.5842	0.7868
gnn-rnn 5y **(ours)**	**0.4677**	**0.6782**	**0.8272**
(std)	*(0.0035)*	*(0.0049)*	*(0.0038)*

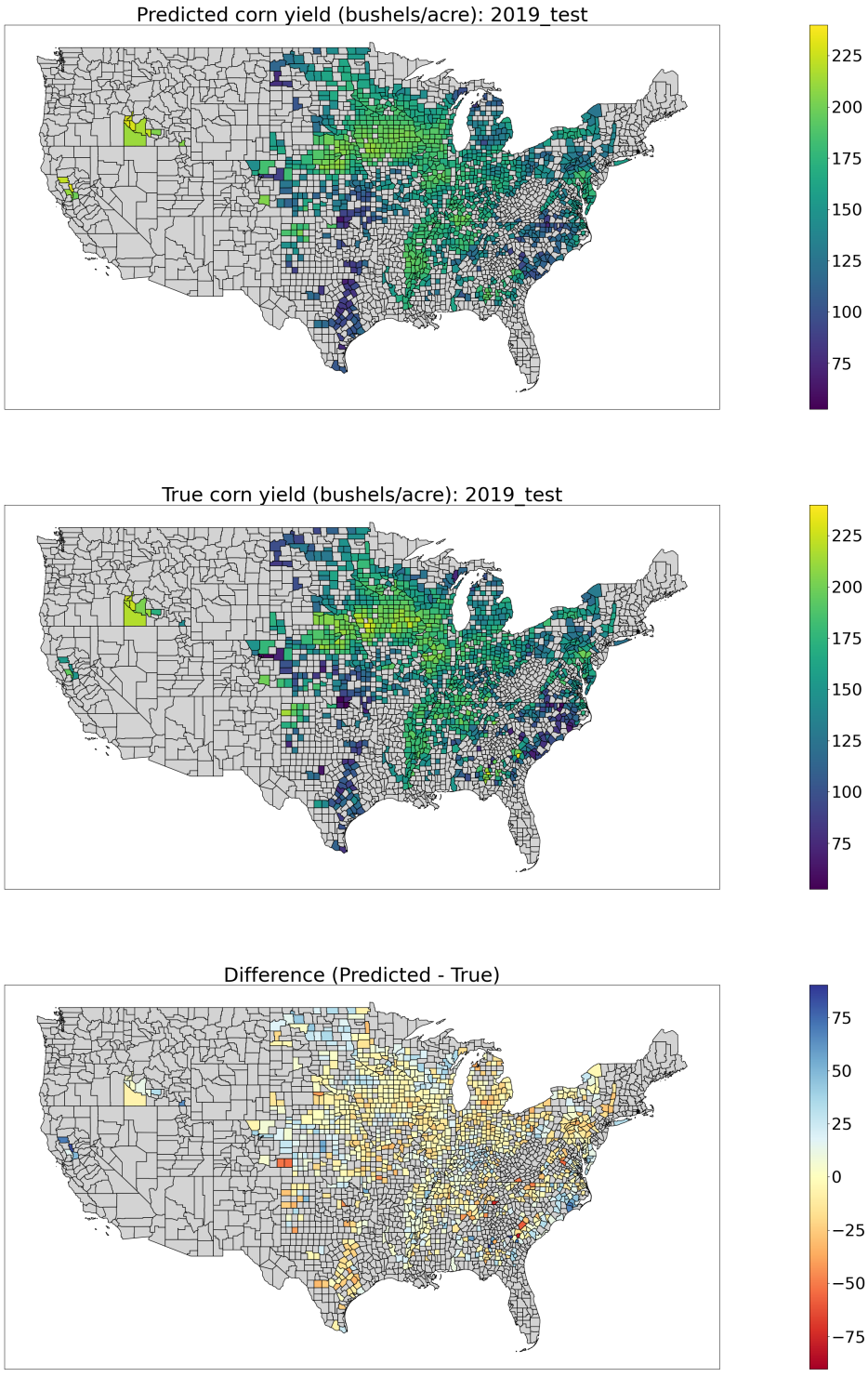


Figure 3: 2019 corn: Maps of predicted (top) and true (middle) yields, along with the difference (bottom).

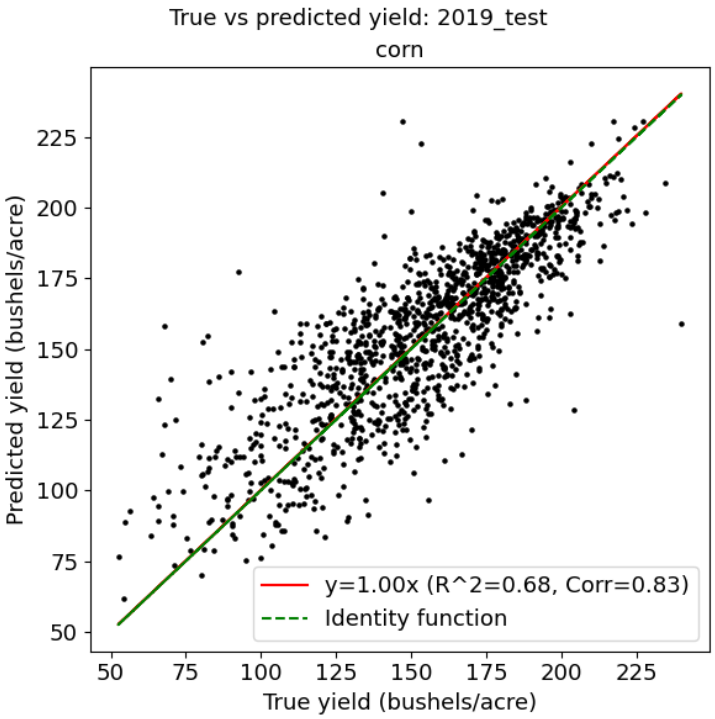


Figure 4: 2019 corn: Predicted vs. ground truth yields

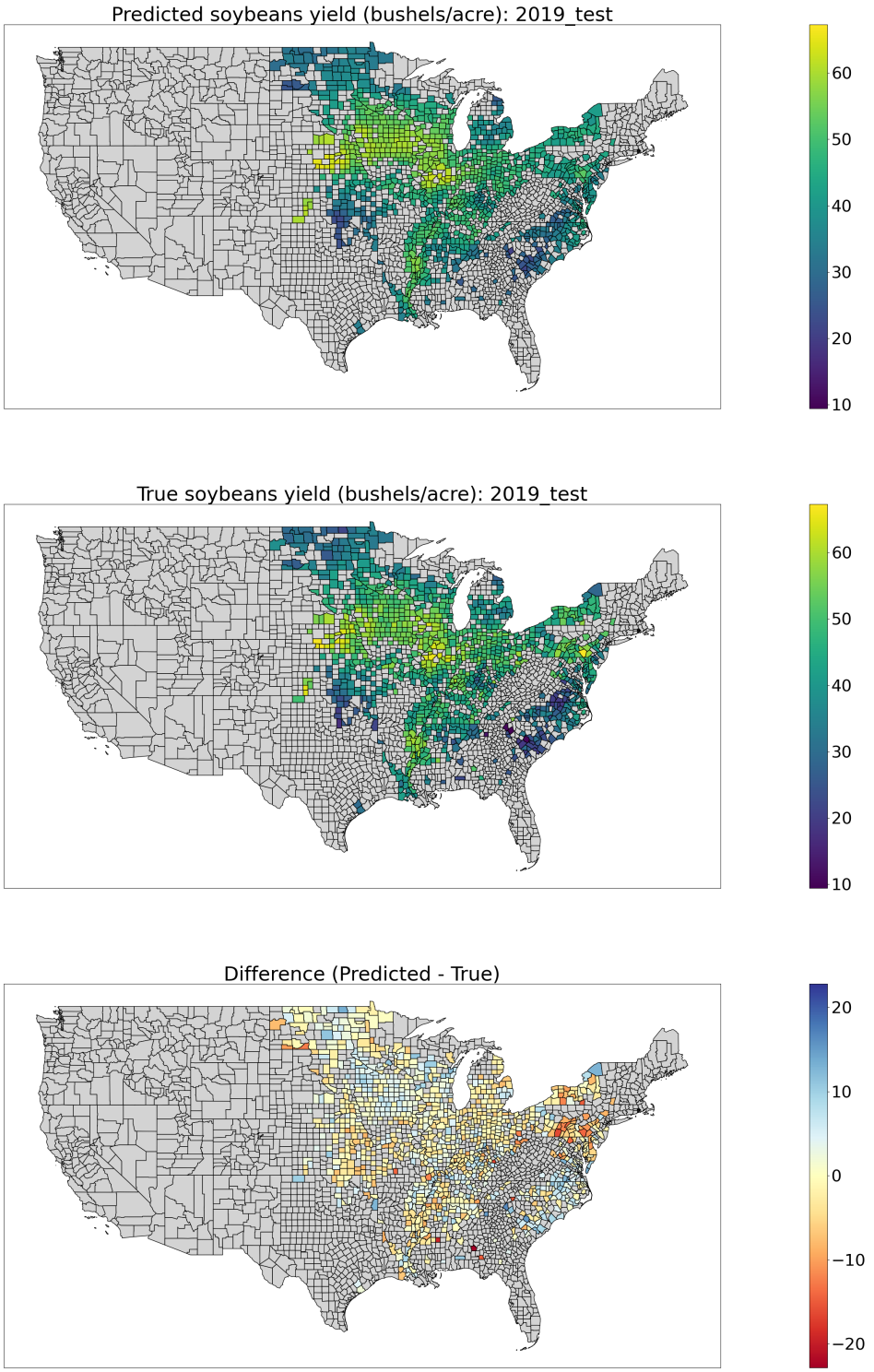


Figure 5: 2019 soybeans: Maps of predicted (top) and true (middle) yields, along with the difference (bottom).

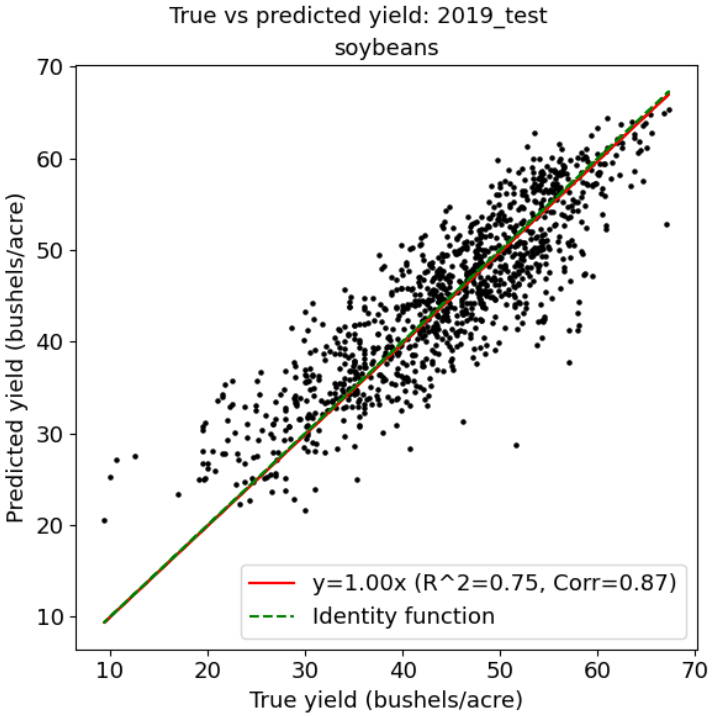


Figure 6: 2019 soybeans: Predicted vs. ground truth yields