United States Department of Agriculture

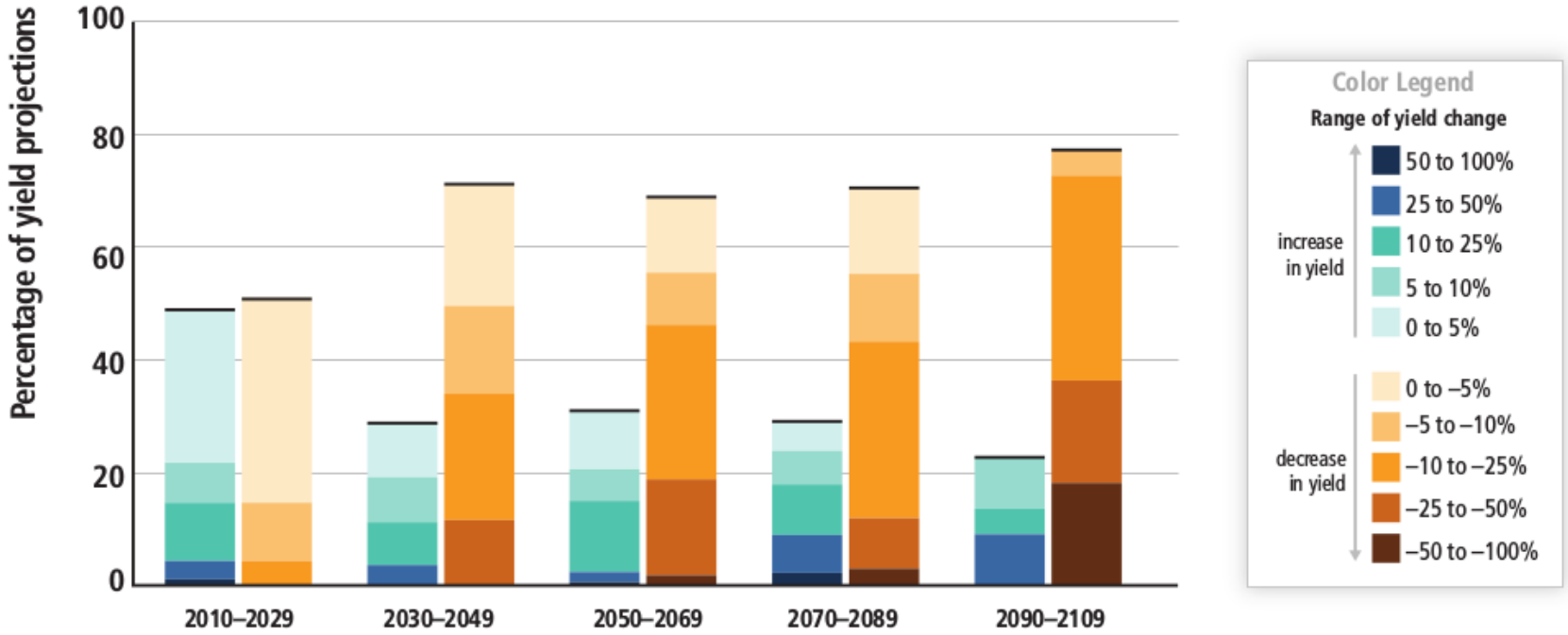
Machine Learning Approaches to Crop Yield Prediction and Climate Change Impact Assessment

Andrew Crane-Droesch

FCSM, March 2018

The views expressed are those of the authors and should not be attributed to the Economic Research Service or USDA

Climate and Agriculture



Source: IPCC WG2 AR5 Ch7

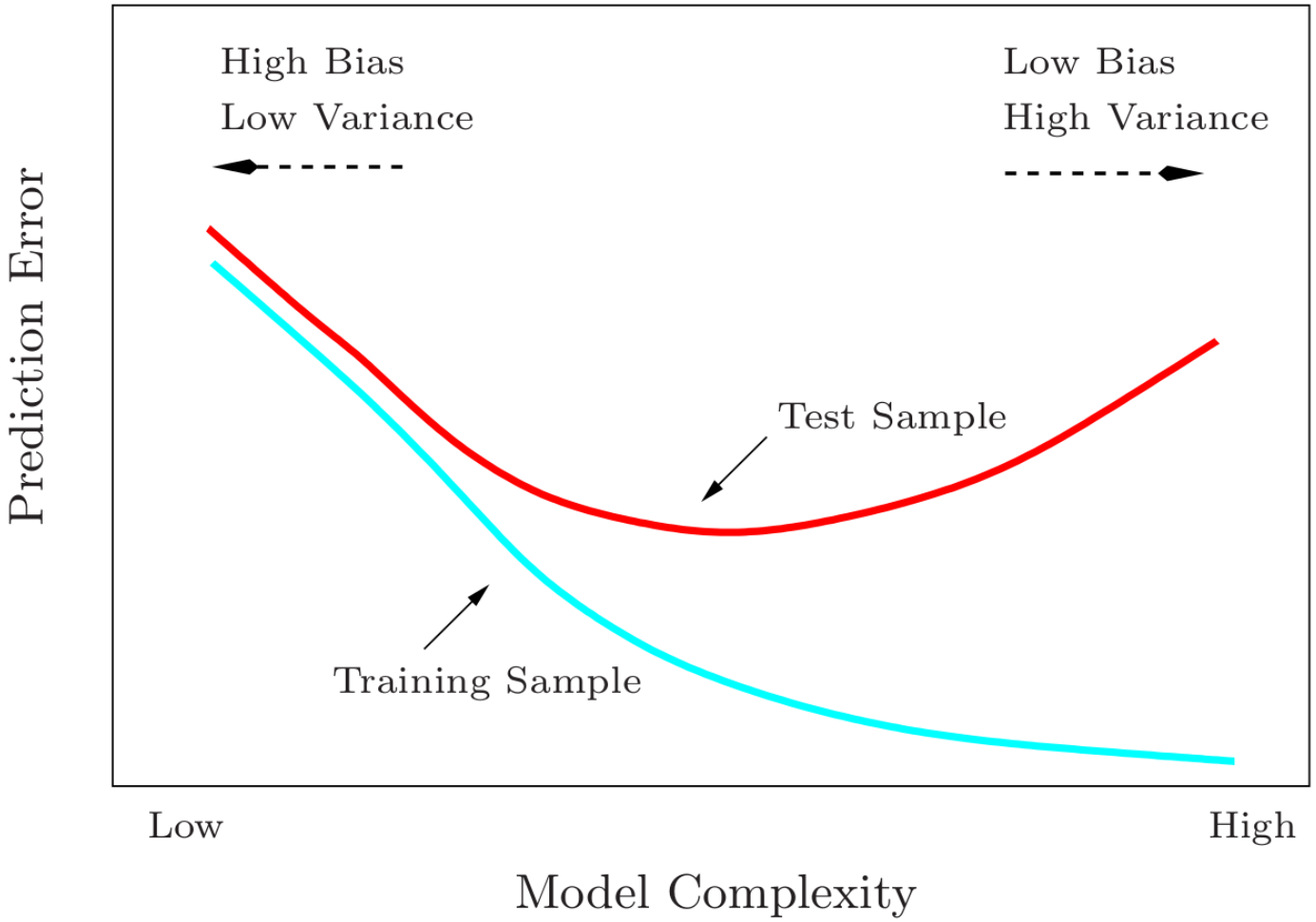
Definitions

Wikipedia defines **machine learning** as

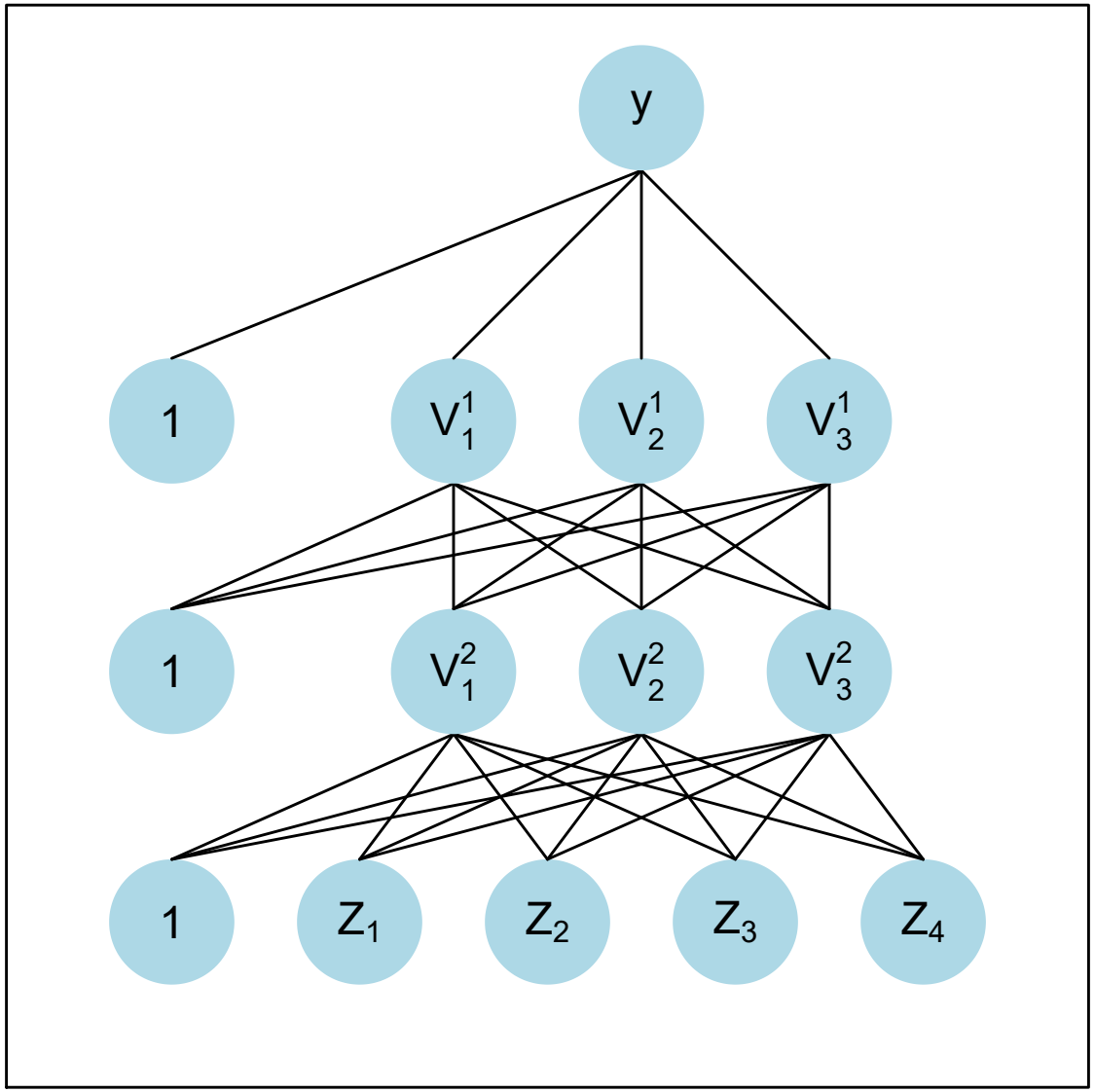
> *"a field of computer science that gives computers the ability to learn without being explicitly programmed"*

- Machine learning should also be considered a branch of statistics – focusing on prediction
- *Supervised* machine learning is the branch of ML focused on predicting new outcomes, based on labeled input data
- Deep neural networks are the present state-of-the-art in many supervised learning problems
- Basic ideas date from the 1950s, but major advances in last decade
  - Software, hardware, algorithmic
- Neural networks can be viewed as algorithms for identifying an optimal set of derived regressors from a set of input data

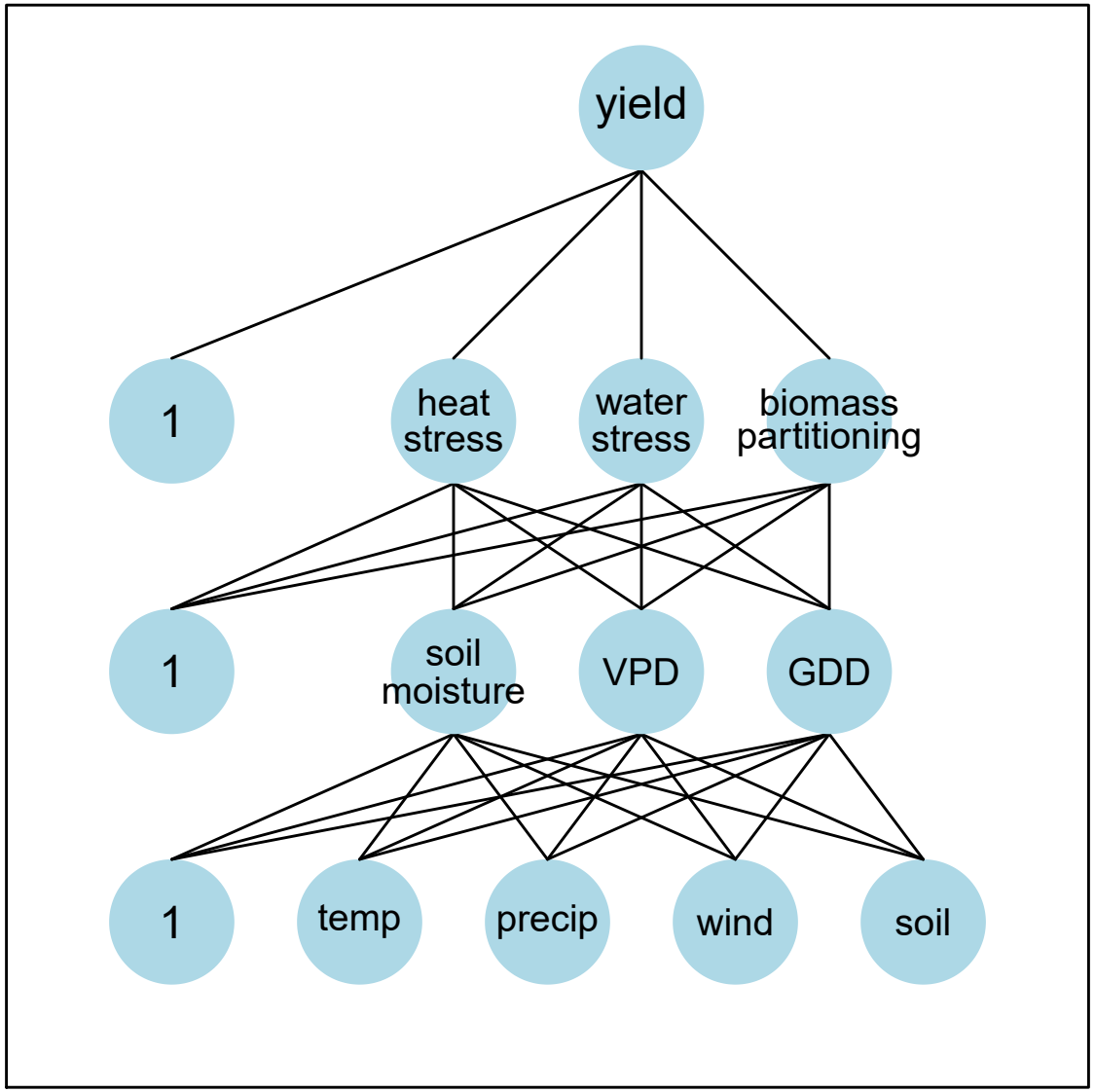
Prediction and the bias-variance tradeoff



Neural networks



Representation Learning



Neural networks

$$
\begin{aligned}
y &= \ {}^{1} + \boldsymbol{V}^{1}\Gamma^{1} + \epsilon \\
\boldsymbol{V}^{1} &= a\left(\ {}^{2} + \boldsymbol{V}^{2}\boldsymbol{\Gamma}^{2}\right) \\
\boldsymbol{V}^{2} &= a\left(\ {}^{3} + \boldsymbol{V}^{3}\boldsymbol{\Gamma}^{3}\right) \\
&\vdots \\
\boldsymbol{V}^{L} &= a\left(\ {}^{L} + \boldsymbol{Z}\boldsymbol{\Gamma}^{L}\right)
\end{aligned}
$$

- y – a (continuous) outcome
- ϵ – additive error
- $\boldsymbol{Z}$ – data
- $\boldsymbol{V}^{l}$ – "nodes": derived variables
- $a()$ – the "activation function". Maps the real line to some subset of it. Modern nets use variants of the ReLU: $a(x) = max(0, x)$
- Dimension of $\boldsymbol{\Gamma}^{1:L}$ controls number of nodes per layer

Training: backpropagation

Typical loss function for continuous-variable prediction problems is the L2-penalized squared error loss:

$$\underset{\boldsymbol{\theta}}{argmin}\left((y-\hat{y})^2+\lambda\boldsymbol{\theta}^{\boldsymbol{T}}\boldsymbol{\theta}\right)$$

where $\boldsymbol{\theta} \equiv vec(\Gamma^1, \Gamma^2, ..., \Gamma^L)$ and λ is a tunable hyperparameter controlling model complexity

No closed-form solution! Instead, training is done by backpropagation:

- Backward pass: Compute gradient of loss function with respect to parameters
- Update parameters based on gradient of loss function
- Forward pass: Recompute hidden layers and value of loss function
- Repeat to approximate convergence

Semiparametric and panel neural nets

- The top layer of a neural net is an OLS regression in derived variables $\boldsymbol{V}$

$$y = \quad + \boldsymbol{V}\Gamma + \epsilon$$

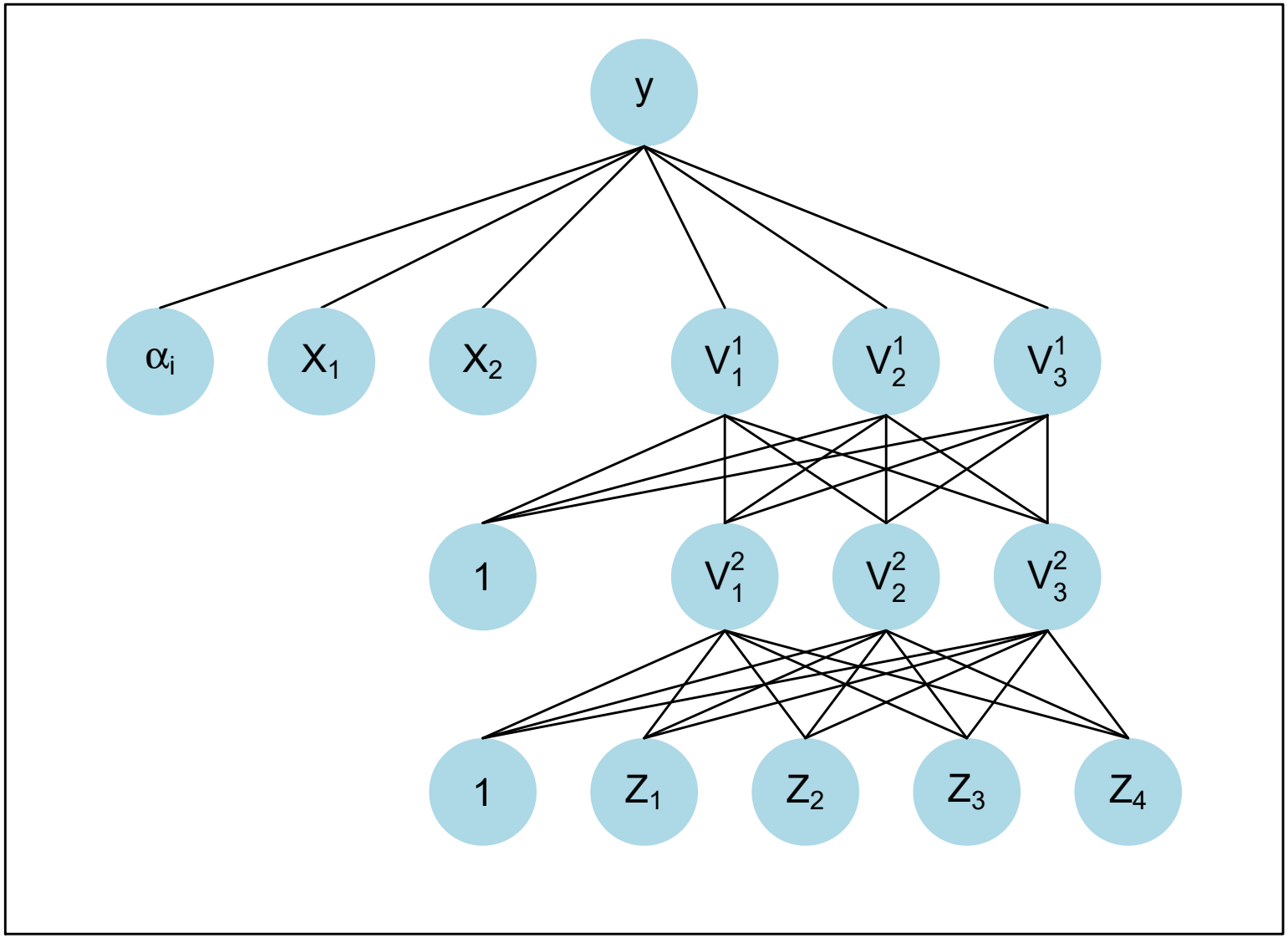
- It is simple to add linear terms to the model, where a linear-in-parameters relationship is known to be appropriate:

$$y = \quad + \boldsymbol{X} \quad + \boldsymbol{V}\Gamma + \epsilon$$

- Likewise, panel structure can be accounted-for by adding unit-specific intercepts at the top level:

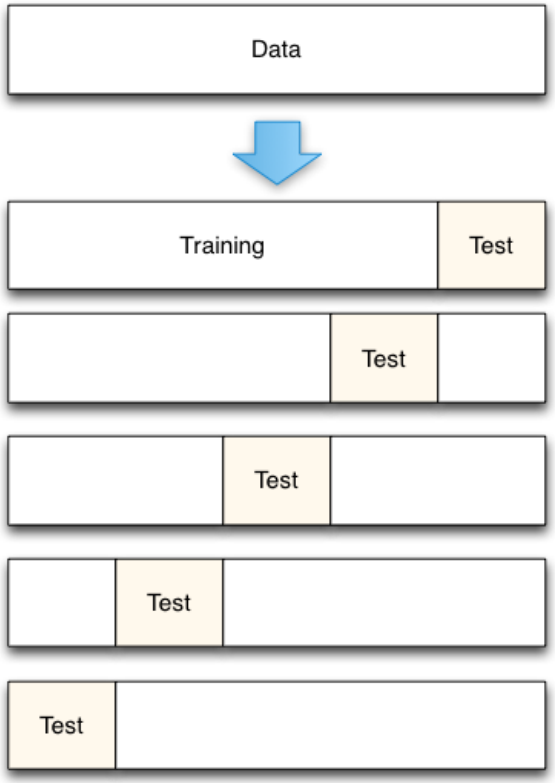
$$y_{it} = \quad _i + \boldsymbol{X}_{it} \quad + \boldsymbol{V}_{it}\Gamma + \epsilon_{it}$$

Semiparametric and panel neural nets



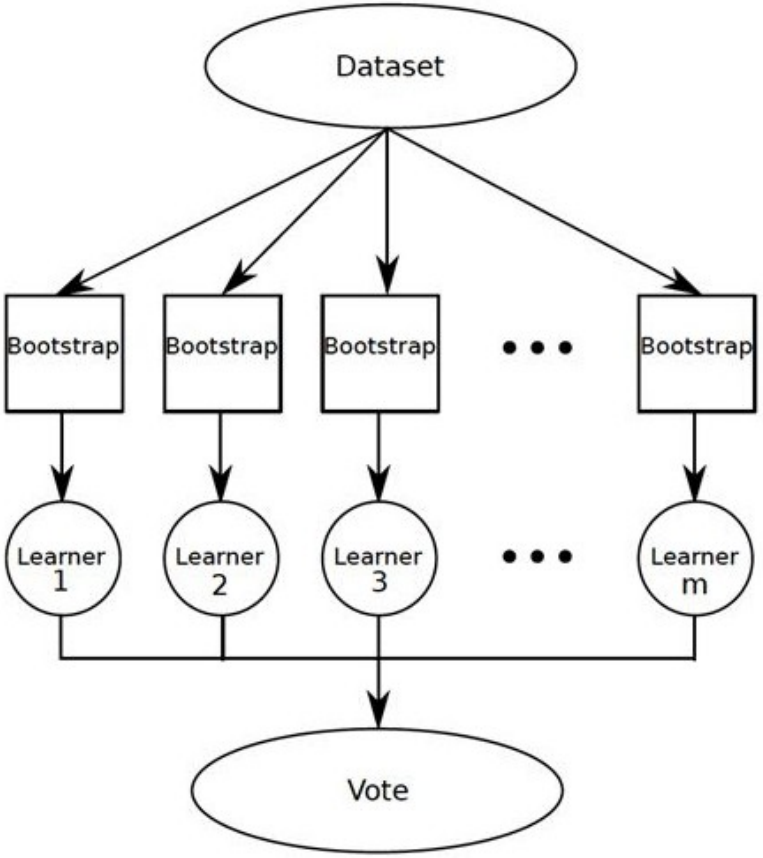
Cross-validation, bagging, and cross-bagging

- λ controls the tradeo between a model that is too rigid and a model that is overfit
- Typically chosen by cross-validation



Cross-validation, bagging, and cross-bagging

Another technique for controlling the variance of an estimator is bootstrap aggregating, or **bagging**



Cross-validation, bagging, and cross-bagging

- This project combines cross-validation with bagging.
  - New term: "cross-bagging"
- B bootstrap samples of unique years are taken
- Panel neural nets are fit to each bootstrap sample; the test set are the years left out of the sample
  - Starting with a large λ, $\boldsymbol{\theta}$ is estimated
  - Out-of-sample fit is assessed
  - λ is halved, and $\boldsymbol{\theta}$ is updated
  - The model with the best out-of-sample fit is retained
- Predictions are the average of each of the B model's predictions

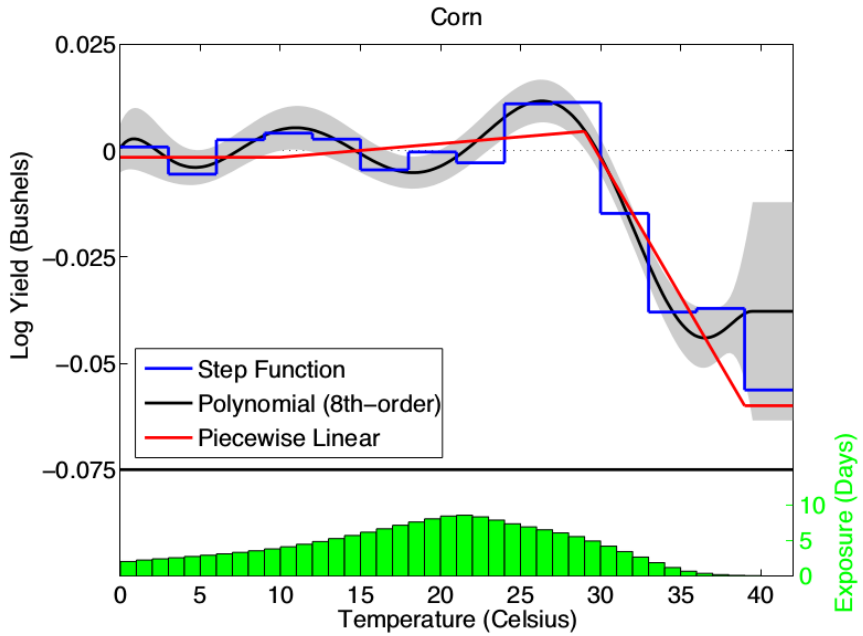
Data

- Maize yield data from Iowa and Illinois, from NASS, 1979 - 2016
- Historical weather data from METDATA (Abatzoglou 2013)
  - Daily observations of min/max temperature, min/max relative humidity, precipitation, wind speed, insolation
  - 4km resolution
  - Aggregated to counties and weighted by agricultural area
- Climate scenario data from Multivariate Adaptive Climate Analogs (MACA) dataset (Abatzoglou 2012)
  - Statistical downscaling of CMIP5 climate model runs, for RCP4.5 and 8.5
  - So far, this project only analyzes one model (Hadley Centre, UK)
  - Same variables, resolution, processing
- Other variables: soil, time, lat/lon, proportion irrigated

Baseline parametric yield model

$$y_{it} = \quad {}_i + \sum_r \mathrm{GDD}_{rit} \quad {}_r + \boldsymbol{X}_{it} \quad + \epsilon_{it}$$

- Pioneered by Schlenker & Roberts (2009)



→⇜small shifts in heat have severe impacts on yields

Semiparametric Specification

$$y_{it} = \quad {}_i + \sum_r \mathrm{GDD}_{rit} \quad {}_r + \boldsymbol{X}_{it} \quad + \boldsymbol{V}_{it}\boldsymbol{\Gamma} + \epsilon_{it} \qquad \boldsymbol{\Gamma}\text{: } 100 \times 1$$

$$\boldsymbol{V}_{it}^1 = a\left(\quad {}^2 + \boldsymbol{V}_{it}^2\boldsymbol{\Gamma}^2\right) \qquad \boldsymbol{\Gamma}^2\text{: } 100 \times 100$$

$$\vdots$$

$$\boldsymbol{V}_{it}^{10} = a\left(\quad {}^{10} + \boldsymbol{Z}_{it}\boldsymbol{\Gamma}^{10}\right) \qquad \boldsymbol{\Gamma}^{10}\text{: } 1800 \times 100$$

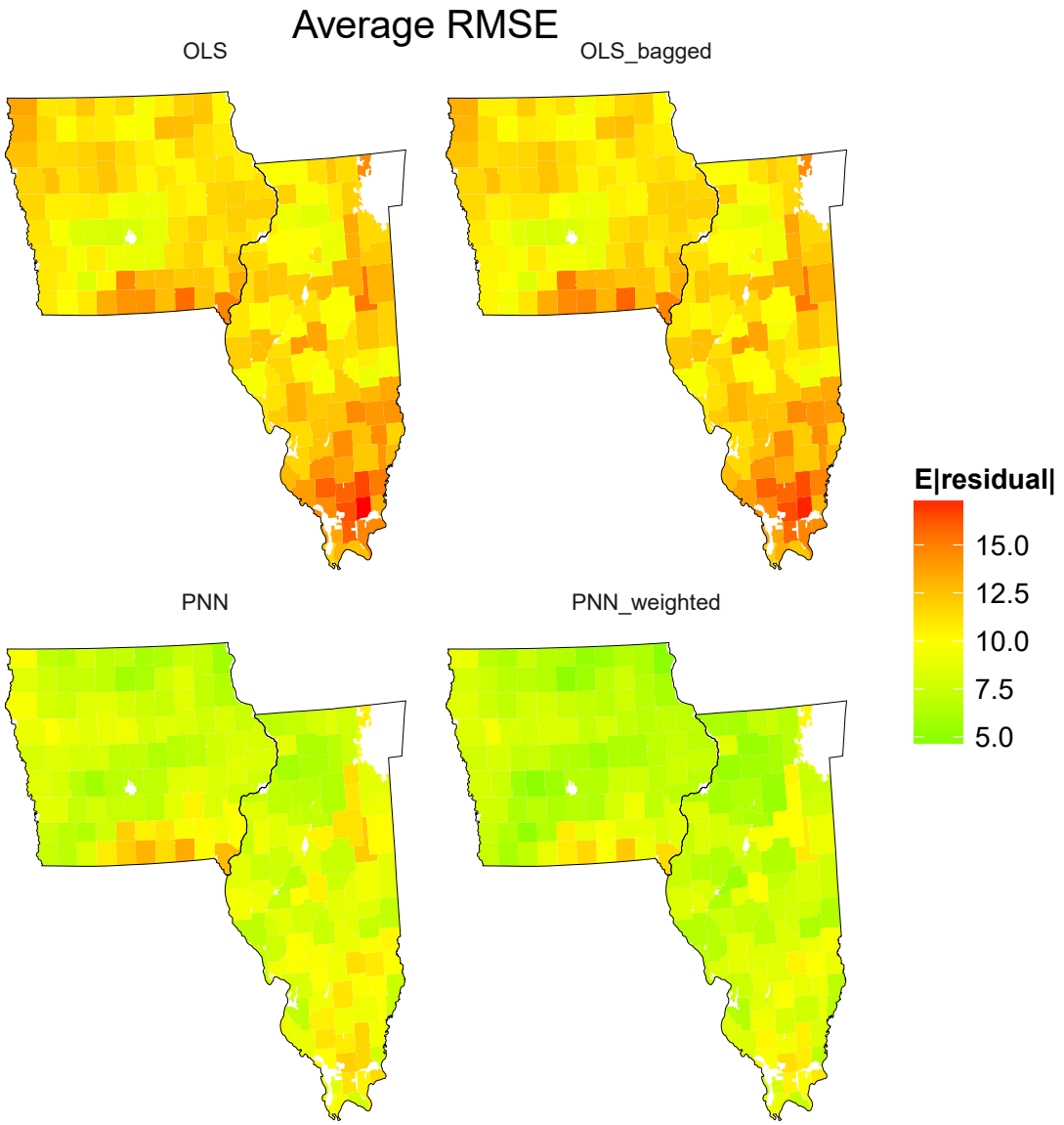
- *Identical* to parametric model, with addition of 100-node neural network layer
- 10 layers, 100 nodes each. **270042 parameters (!!!)** Regularization will be extremely important.
- Activation is the "leaky ReLU" $a(x) = x$ if $x > 0$ else $x/100$
- Two versions: parametric (linear) part of model penalized and unpenalized

Results: Predictive Skill

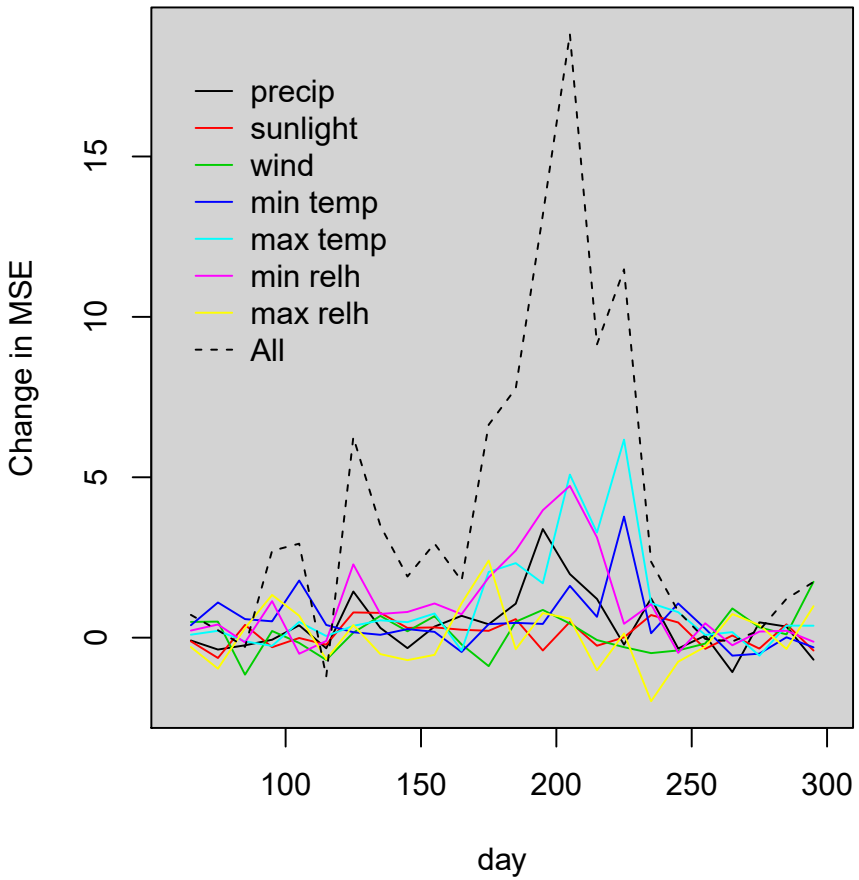
Model	Bagged	Ensemble Weights	$\widehat{\text{MSE}}_{oob}$
Parametric	no	N/A	386.7
Panel neural net	no	N/A	385.4
Parametric	yes	unweighted	336.3
Panel neural net	yes	unweighted	293.4
Parametric/ridge	no	N/A	416.2
Parametric/LASSO	no	N/A	387.2
Panel neural net	yes	scheme 1	277.1
Panel neural net	yes	scheme 2	289.3
Panel neural net	yes	scheme 3	293.0
Panel neural net	yes	scheme 4	324.1

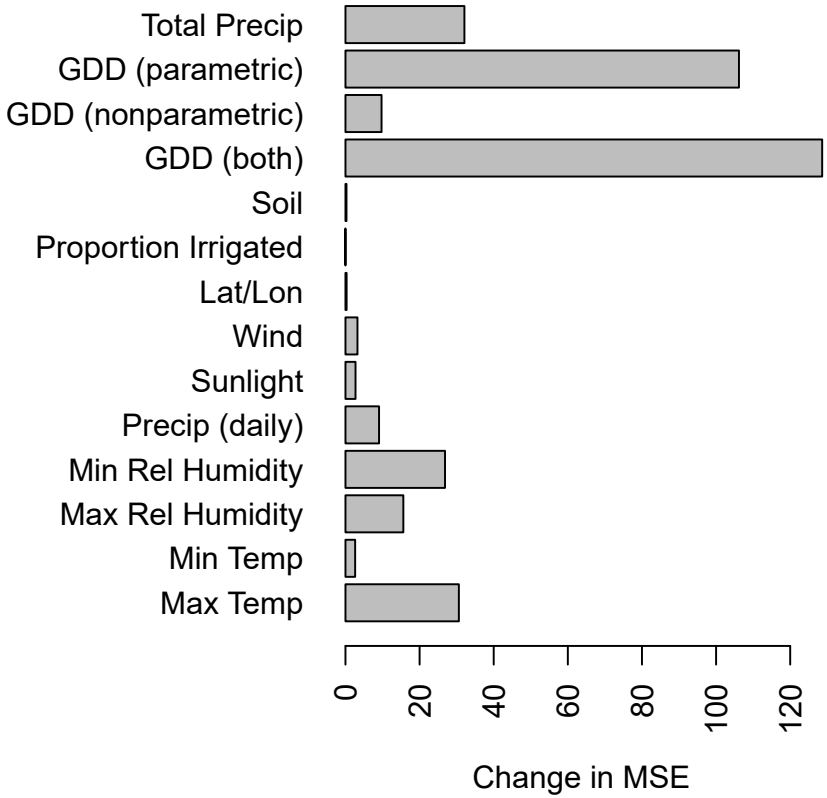
- Scheme 1: $w = \text{MSE}_{oob}^{P} - \text{MSE}_{oob}^{PNN}$, if $w < 0, w \quad 0$
- Scheme 2: $w = \text{MSE}_{oob}^{FE} - \text{MSE}_{oob}^{PNN}$
- Scheme 3: $w = \underset{w}{argmin} \left((y - \tilde{\boldsymbol{y}} w)^2 \right)$ s.t. $w \geq 0$
- Scheme 4: $w = e^{\left((\tilde{\boldsymbol{y}}^T \tilde{\boldsymbol{y}})^{-1} \tilde{\boldsymbol{y}}^T y \right)}$

Predictive skill, spatial (in-sample)



Variable Importance





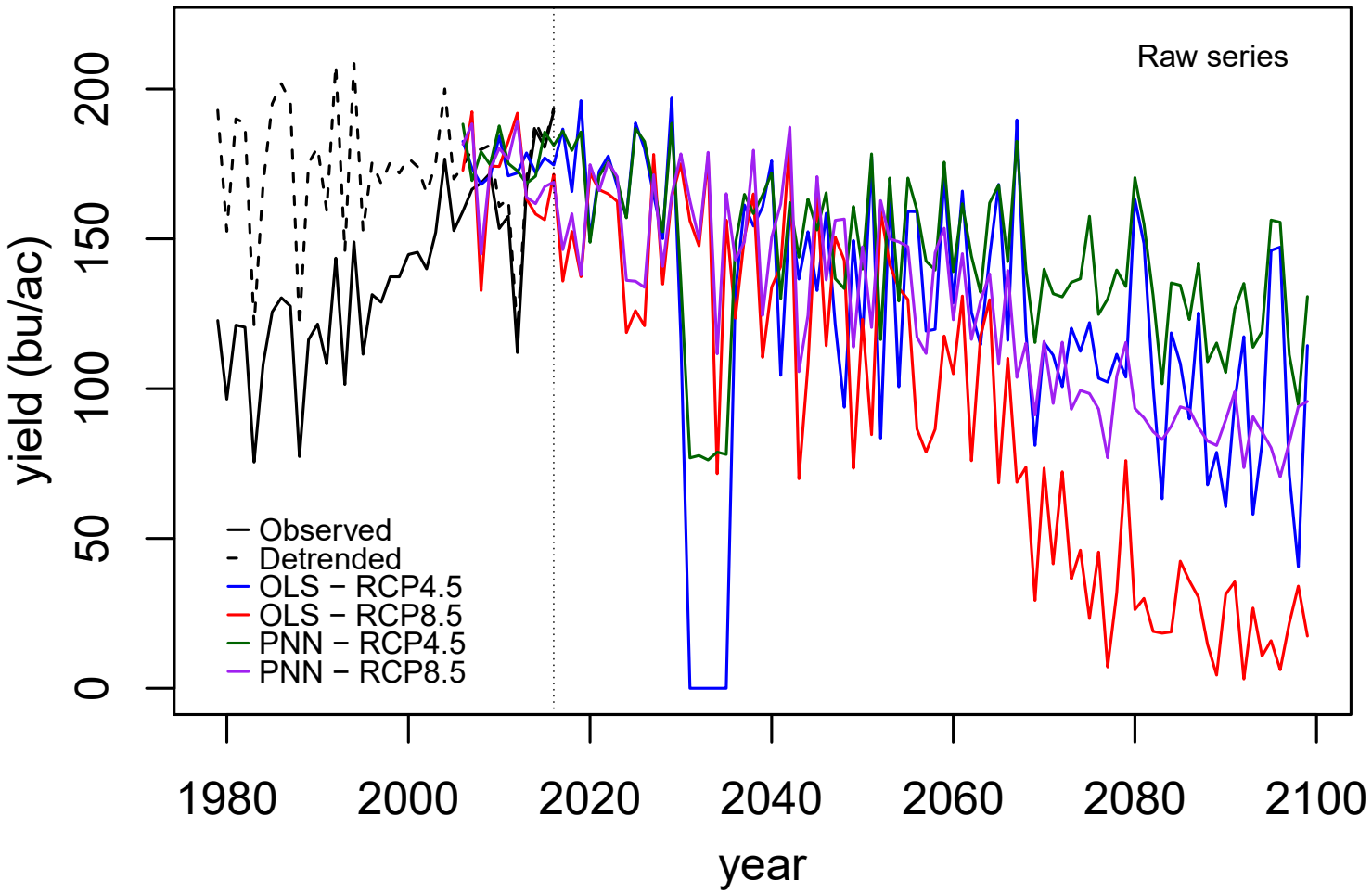
Permutation importance: computed by randomly permuting input variables, predicting, and comparing resultant MSE against baseline

Projections

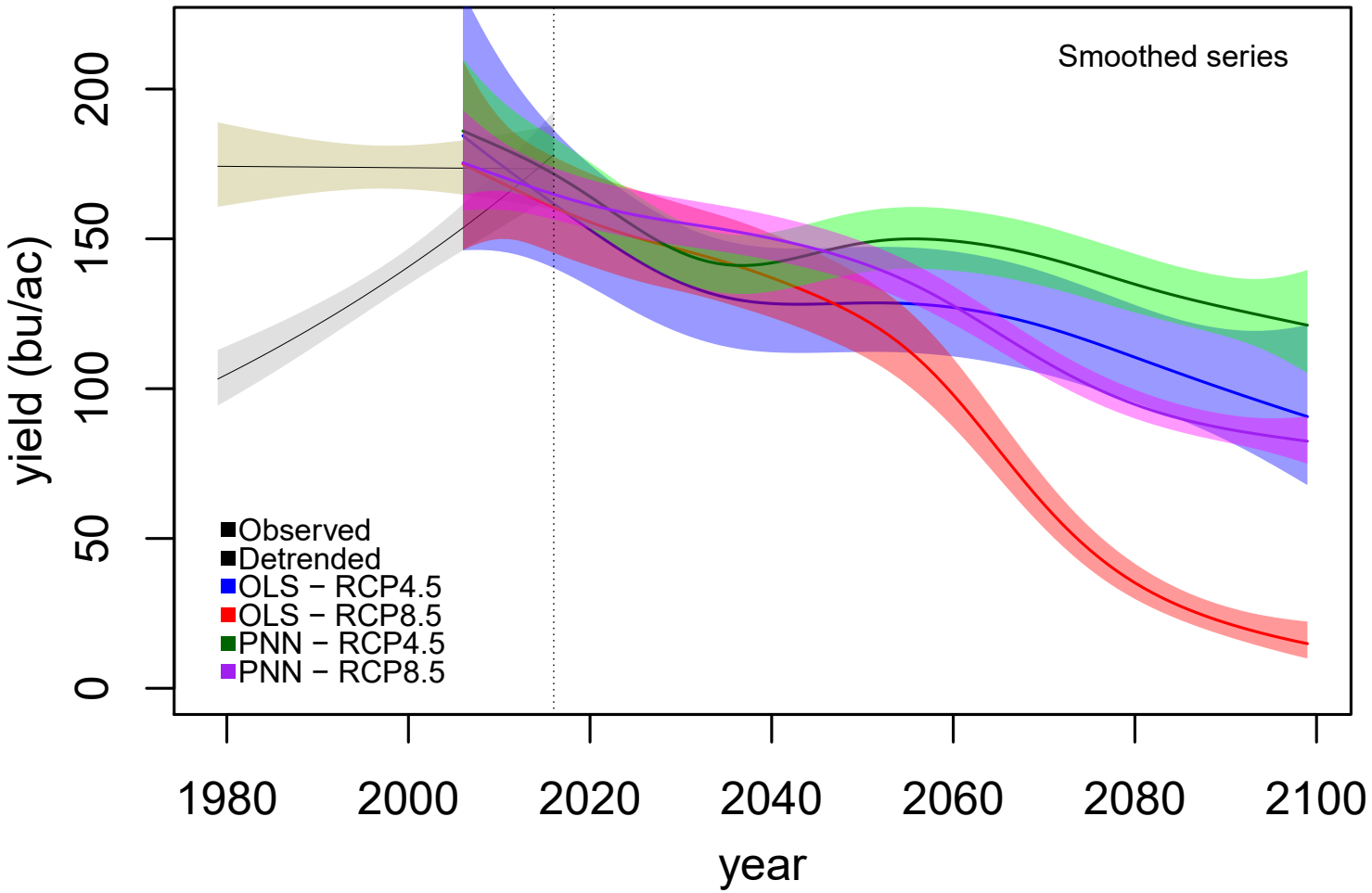
Things to note:

- All projections made assuming values of trends for year 2016
  - I.e.: We make no assumptions about the continuance of technological or other change
  - Projections only reflect change in response to weather, assuming today's technology and today's response to weather
- We don't model adaptation or carbon fertilization
- We only use one run of one GCM (for now)
  - The GCM that we use (HadGEM-CC) is fairly pessimistic, compared to others comprising the CMIP5 suite

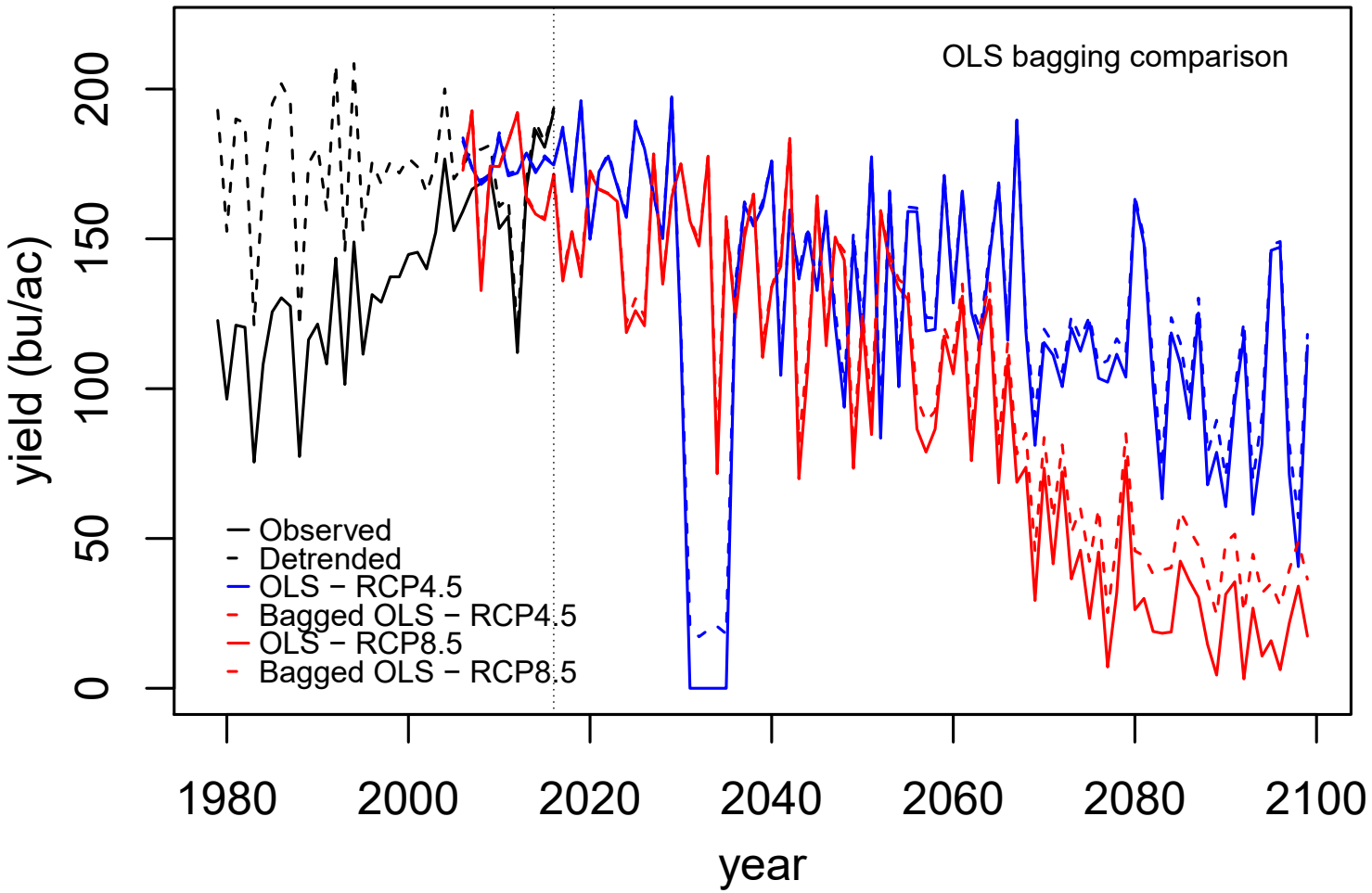
Projections



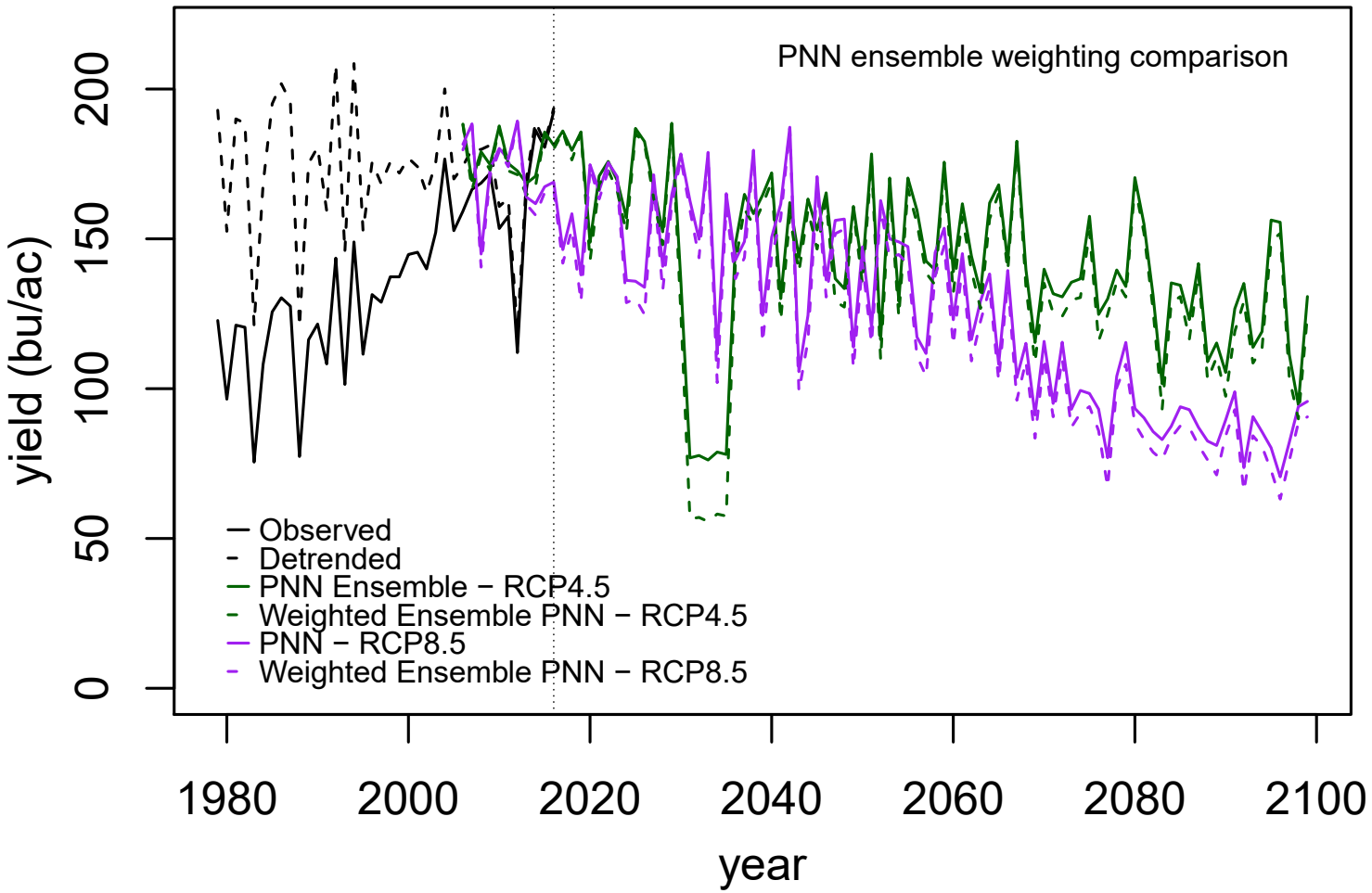
Projections



Projections



Projections



Conclusions

- Panel neural nets perform well in modeling crop yields
  - Their skill depends critically on their ability to *build on* known parametric structure.
  - They basically model everything that is left unexplained by standard parametric approaches
  - Parametric models are important. They explain mechanisms. We show that they are also base upon which to build predictive models.
- PNN predictions of climate change impacts on yield are severe
  - They are less severe than those projected by parametric models
  - Major differences:
    - Ability to model time-dependent interactions between multiple variables
    - Important mediating role for moisture: time-dependent effects of precipitation, relative humidity, and temperature

Advert

`panelNNET` R package:

```
>library(devtools)
>install_github("cranedroesch\panelNNET")
>library(panelNNET)
>?panelNNET
```

Contributions welcome:

`https://github.com/cranedroesch/panelNNET`

Link to working paper:

`https://arxiv.org/abs/1702.06512`

Contact:

`andrew.crane-droesch@ers.usda.gov`

Thanks!