Interpretation and Visualization of Weather-based Machine-learning Models

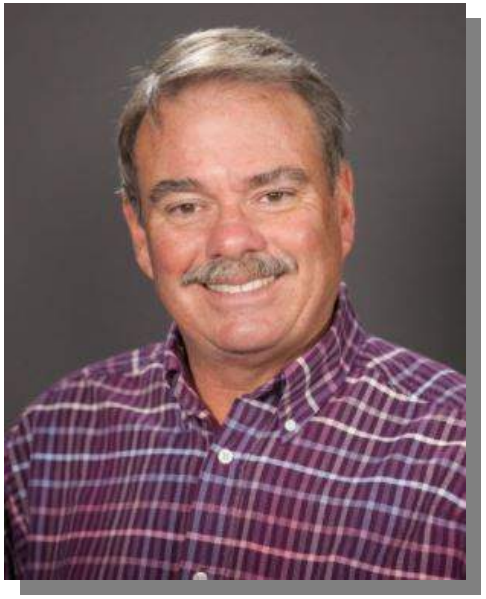
Amy McGovern, Ryan Lagerquist, Kim Elmore, David John Gagne II, Eli Jergensen

Amy McGovern
@profamymcgovern
University of Oklahoma (OU)

Ryan Lagerquist
@ralager_Wx
OU, CIMMS

David John Gagne II
@DJGagneDos
National Center for Atmospheric Research (NCAR)

Kim Elmore
kim.elmore@noaa.gov
Cooperative Institute for Mesoscale Meteorological Studies (CIMMS)



Eli Jergensen
gelijergensen@ou.edu
OU

1. Introduction

- Machine learning and deep learning quickly becoming popular in meteorology.
- However, major barrier to operational implementation is conception that models are “black boxes”.
- Even simple interpretation methods – *e.g.*, listing predictor values (Cintineo *et al*. 2018) – can make meteorologists more likely to adopt machine learning.

1. Introduction

- Uses of model interpretation and visualization (MI&V):

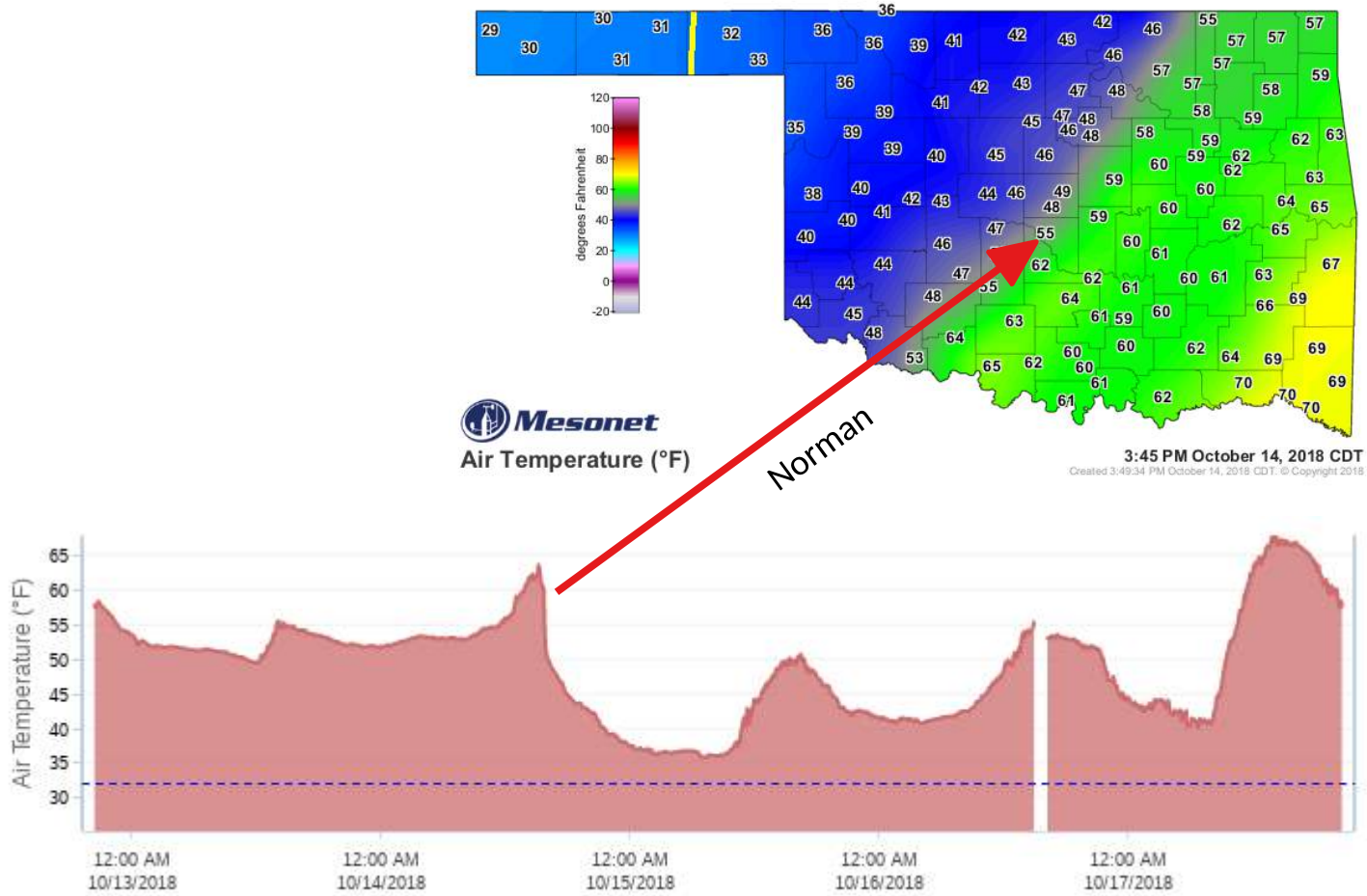
1. **Debugging tool** in development phase
   - Is model not emphasizing a feature that we know is critical?
   - Is model emphasizing a feature that we know is completely irrelevant?
2. **Confidence-assessment tool** in operational phase
   - Some predictions should be trusted more than others.
3. **Learning tool**
   - Learn more about underlying physical processes.
   - Improve understanding, along with prediction, of weather phenomena.

1. Introduction

- **Will present results from 4 projects:**
  - Front detection (2-D convolution)
  - Large-hail prediction (2-D convolution)
  - Storm-mode classification (random forest)
  - Tornado prediction (3-D convolution)
- **Relevant papers:**
  - Gagne II, Haupt, and Nychka: "Interpretable deep learning for spatial severe hail forecasting." *Monthly Weather Review*, under review
  - Lagerquist, McGovern, and Gagne II: "Deep learning for spatially explicit prediction of synoptic-scale fronts." *Weather and Forecasting*, under review
  - McGovern, Gagne II, Lagerquist, Jergensen, Elmore, Homeyer, and Smith: "Making the black box more transparent: Understanding the physical implications of machine learning." *Bulletin of the American Meteorological Society*, proposal accepted, in preparation

2. Front Detection

- Warm and cold fronts are part of global atmospheric circulation.
- Associated with rapid changes in weather, sometimes extreme weather.



Temperature history in Norman. Both images from http://mesonet.org.

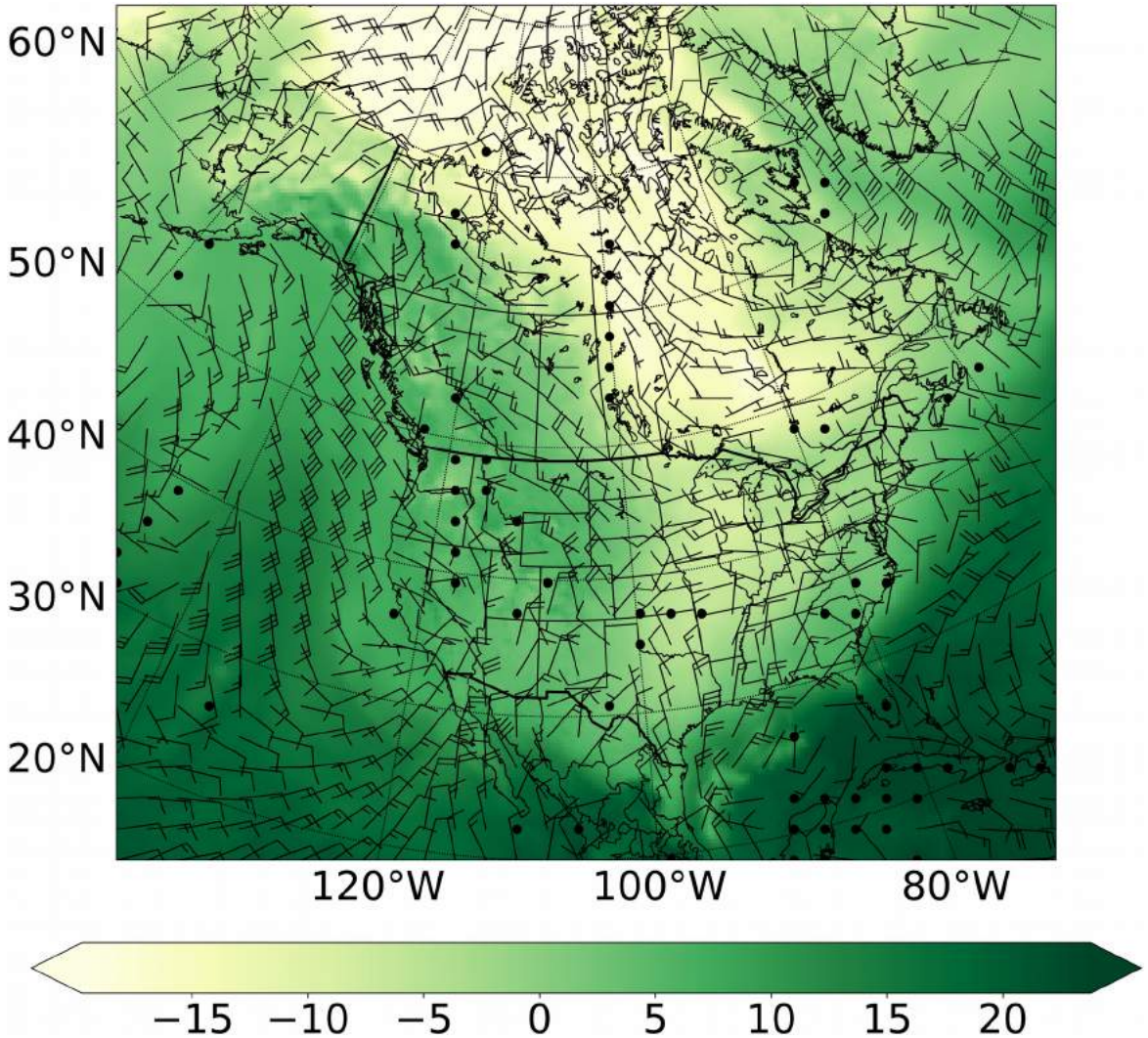
2. Front Detection

- Predictors are 32-km grids, every 3 hours, of basic variables:
  - Temperature
  - Specific humidity
  - *u*-wind
  - *v*-wind

- Labels are human-drawn fronts every 3 hours.

- Prediction task is to label each grid cell as:
  - Warm front
  - Cold front
  - No front

- So task is 3-class classification.



Green fill = wet-bulb potential temperature (°C)

Black vectors = wind

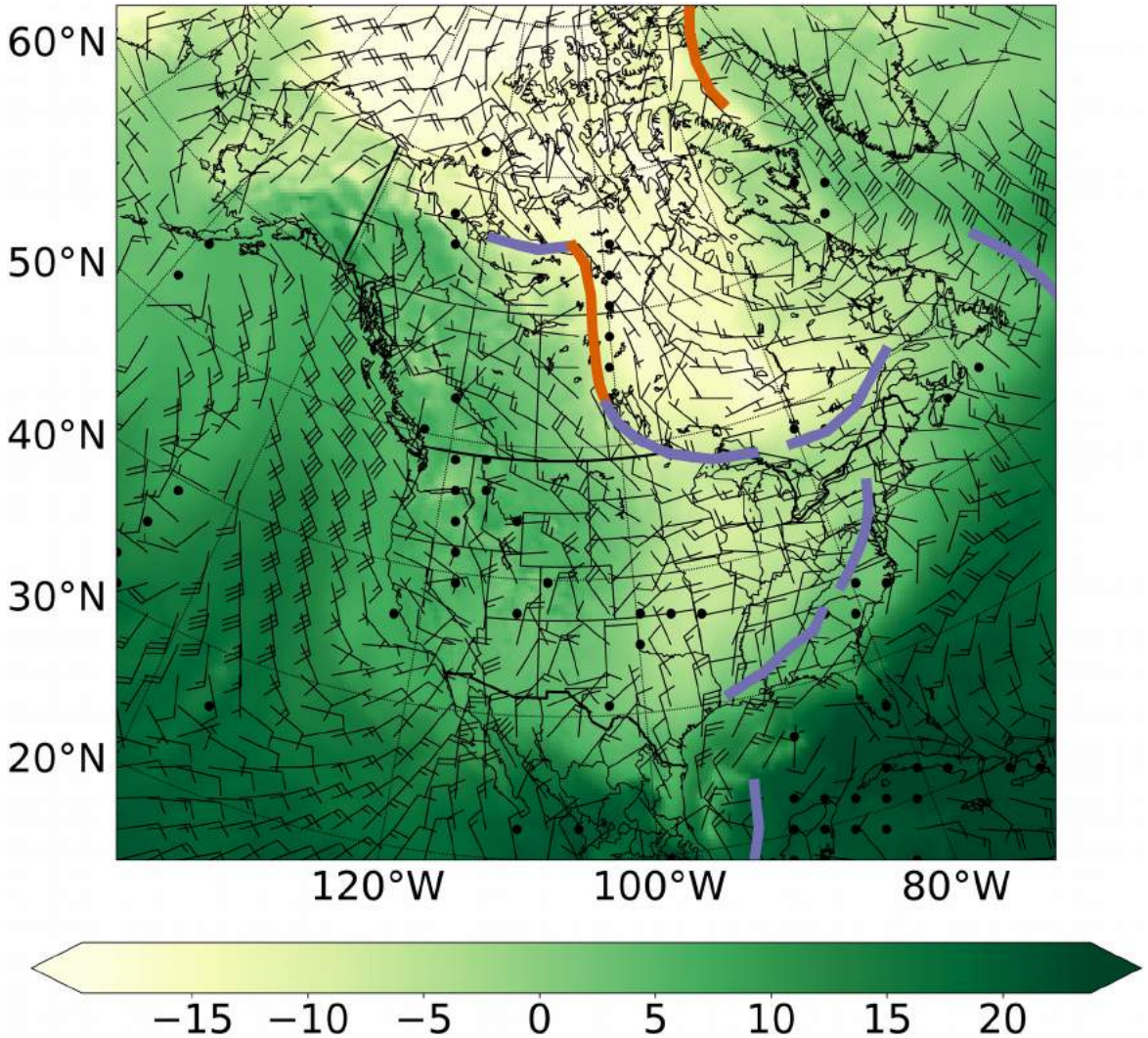
2. Front Detection

- Predictors are 32-km grids, every 3 hours, of basic variables:
  - Temperature
  - Specific humidity
  - *u*-wind
  - *v*-wind

- Labels are human-drawn fronts every 3 hours.

- Prediction task is to label each grid cell as:
  - Warm front
  - Cold front
  - No front

- So task is 3-class classification.

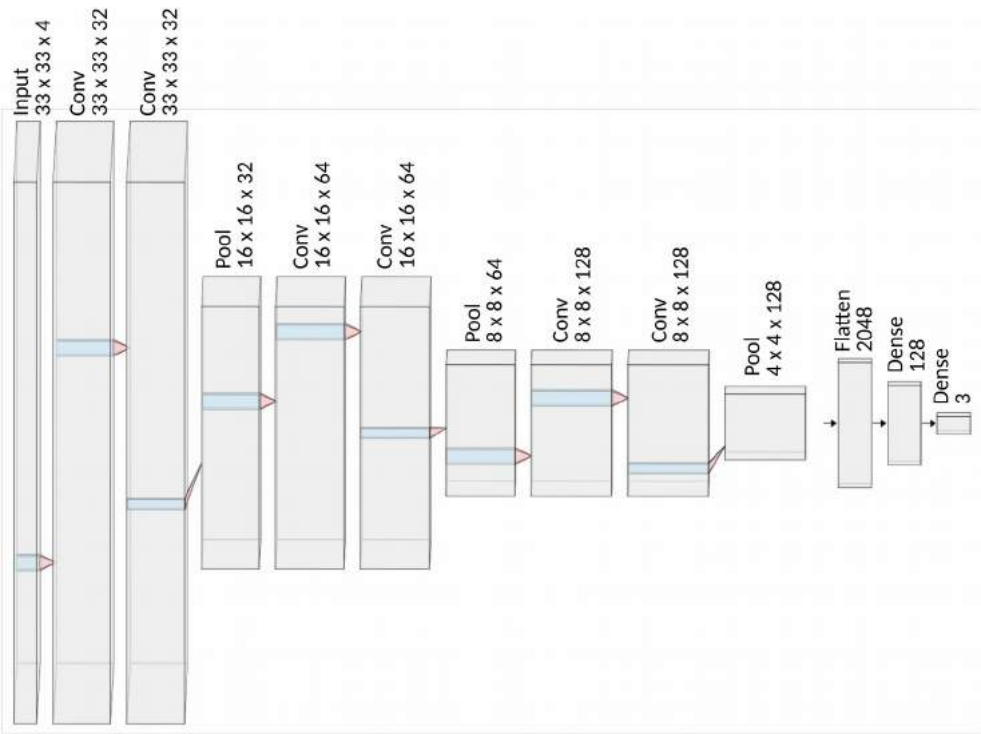


Orange lines = warm fronts

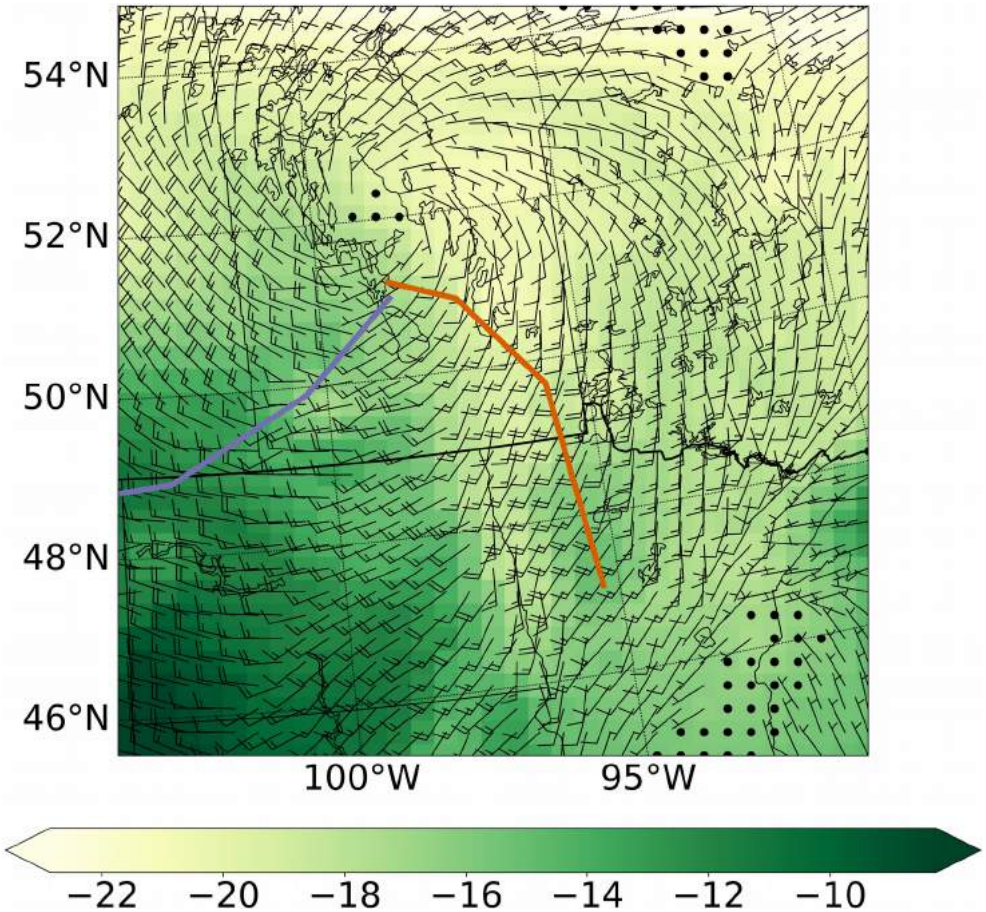
Purple lines = cold fronts

2. Front Detection

We use a convolutional neural net (CNN) with the following architecture.



One learning example = 33 x 33 patch centered on grid cell to be classified.



2a. Permutation Test

- Way to quantify importance of each predictor.
- Breiman (2001) version, or “single-pass” version:

1. Train model with clean dataset (no permutation).
2. For each predictor *u*, permute *u* and compute loss on validation set.

- Most important predictor causes greatest increase in loss function.
- k^{th} most important causes k^{th} greatest increase.

2a. Permutation Test

- Lakshmanan (2015) version, or "multi-pass" version:

1. Train model with clean dataset (no permutation).
2. Let *U* = set of predictors not yet permuted. Initialize: *U* = all predictors.
3. For each predictor $u \in U$, permute *u* and compute loss on validation set. Do not leave *u* permuted.
4. Find predictor (u^*) whose permutation yields the greatest increase in loss. Permute u^* and remove u^* from *U*.
5. If *U* is empty, stop. Otherwise, go back to step 3.

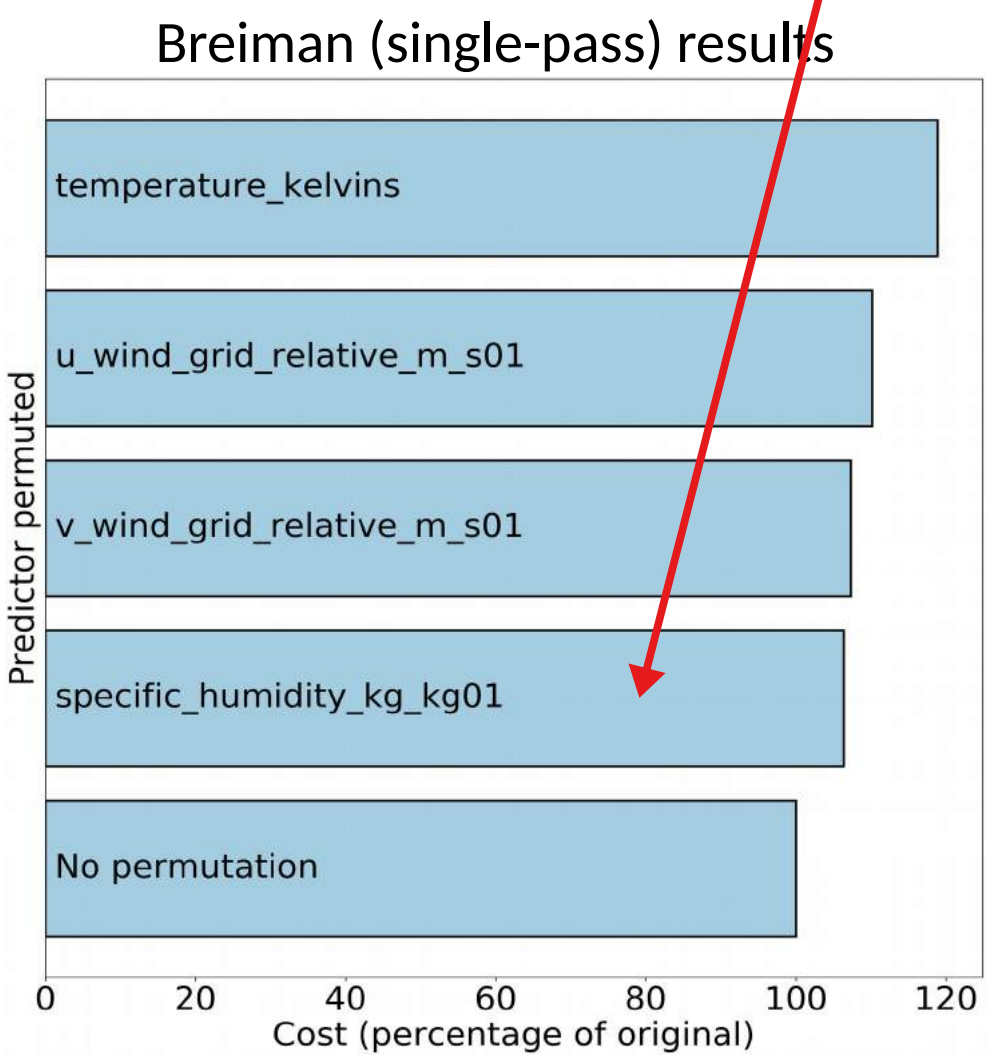
- First predictor permuted in step 4 is most important.
- Breiman and Lakshmanan methods handle correlated predictors differently.

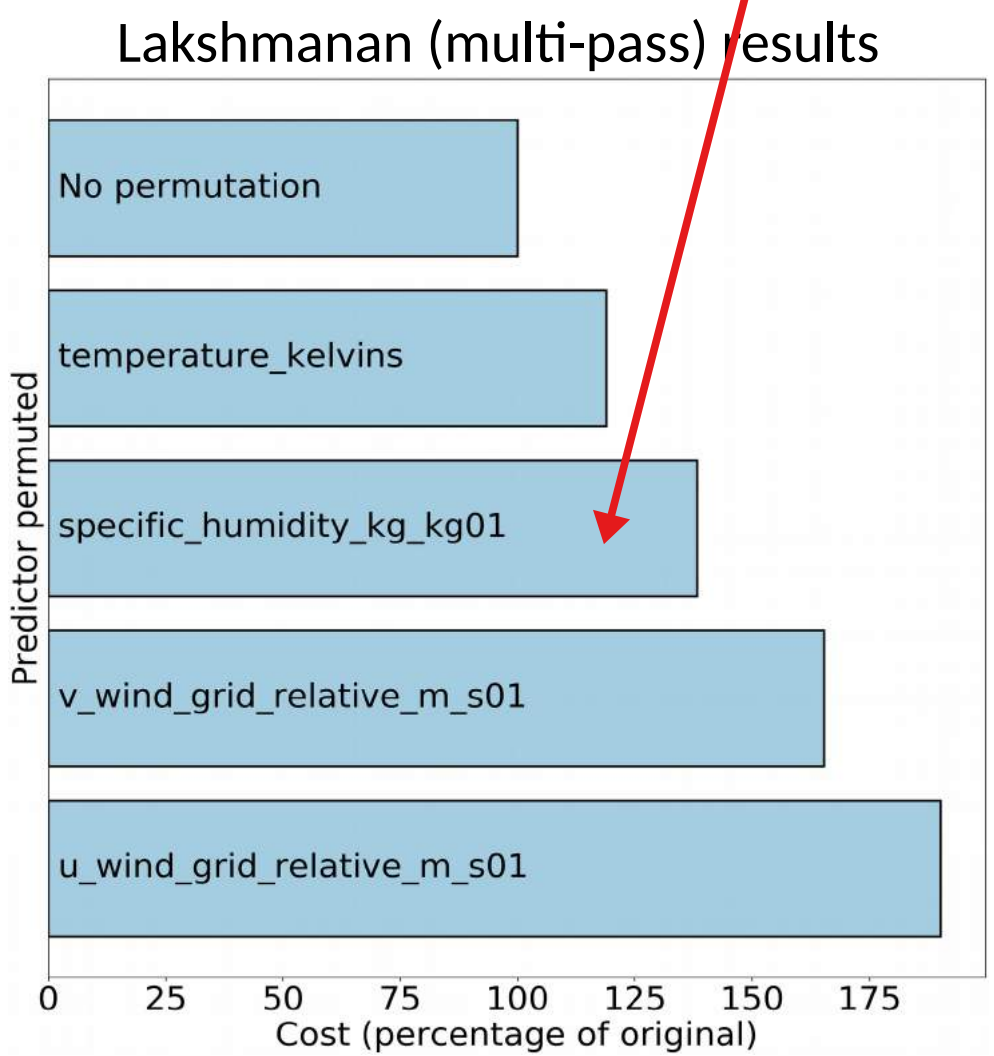
2a. Permutation Test

- Most important predictor at top in both graphs.

Humidity least important when temperature info is intact

Humidity most important when temperature info is destroyed



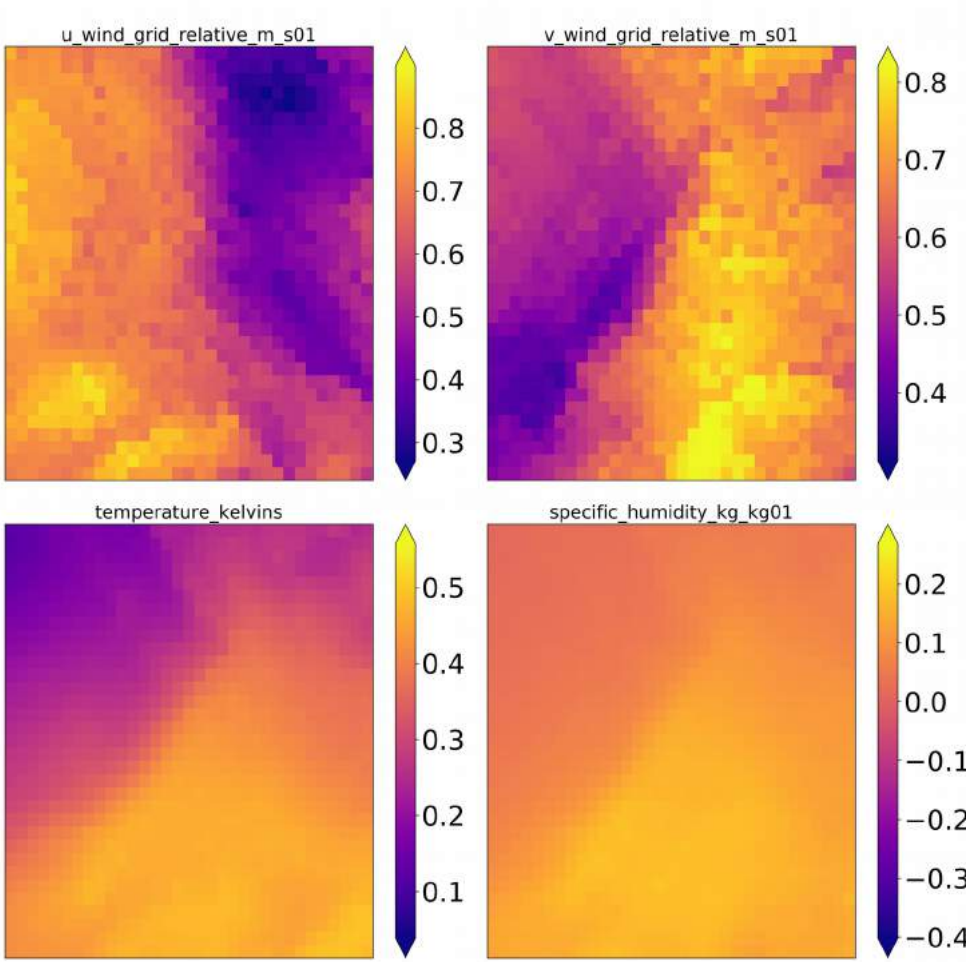


2b. Backwards Optimization

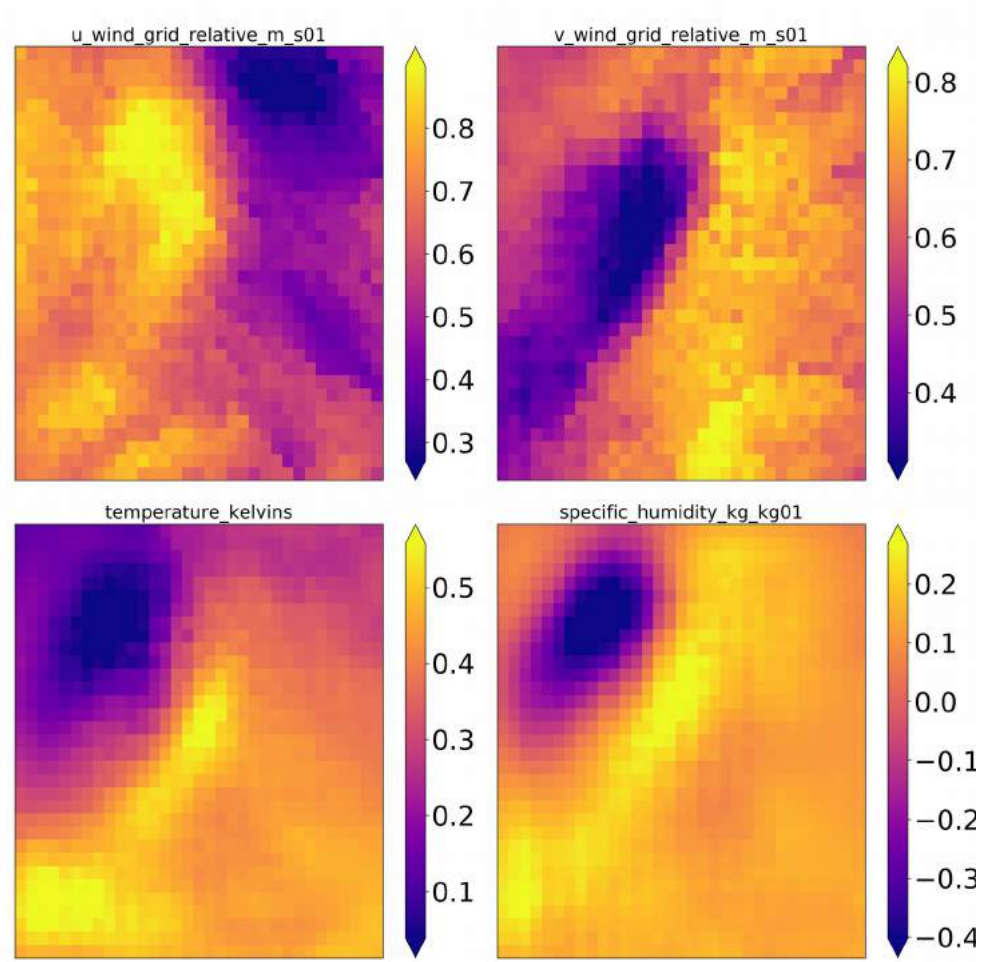
- Also called “feature optimization” (Olah *et al*. 2017).
- Goal: create synthetic input example that maximizes activation of some model component.
- “Some model component” might be:
  - Warm-front probability (activation of 2nd output neuron)
  - Cold-front probability (activation of 3rd output neuron)
- Procedure involves gradient descent, which requires initial seed.
- Initial seed might be:
  - Uniform image (*e.g.*, all zeros)
  - Random image
  - Dataset example
- We use dataset examples from the testing set.

2b. Backwards Optimization

Original dataset example

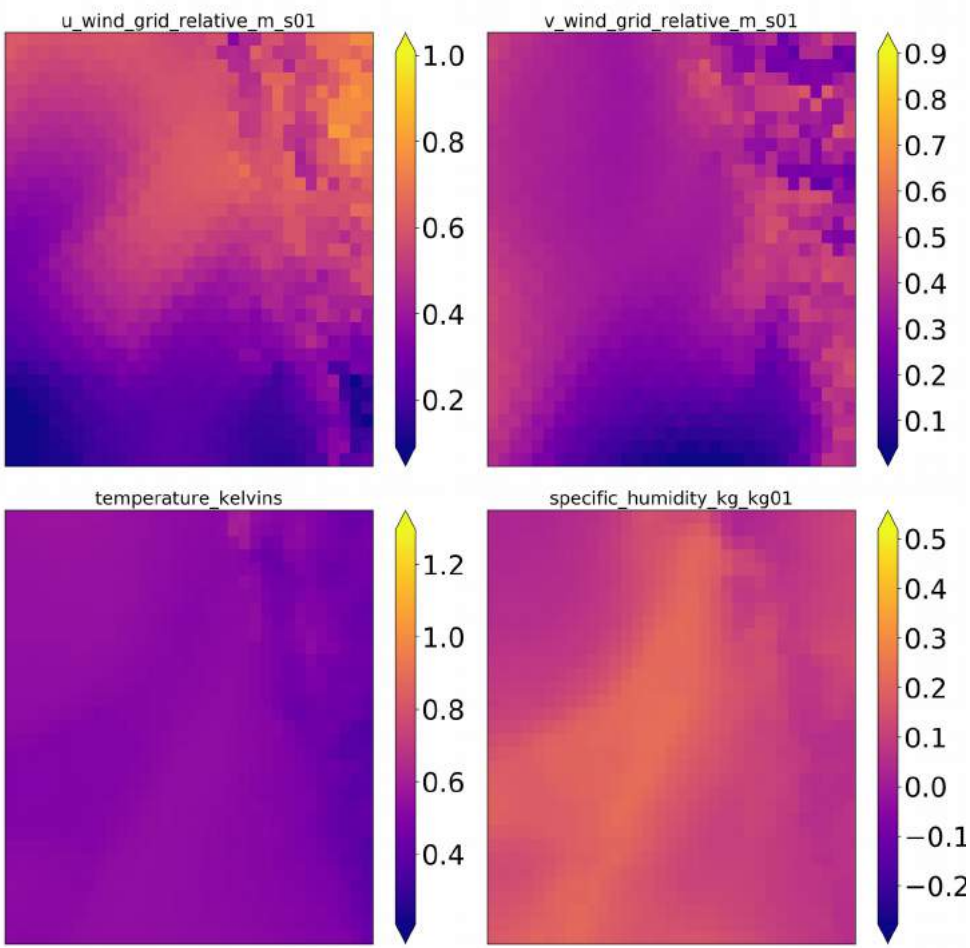


Optimized for cold-front probability

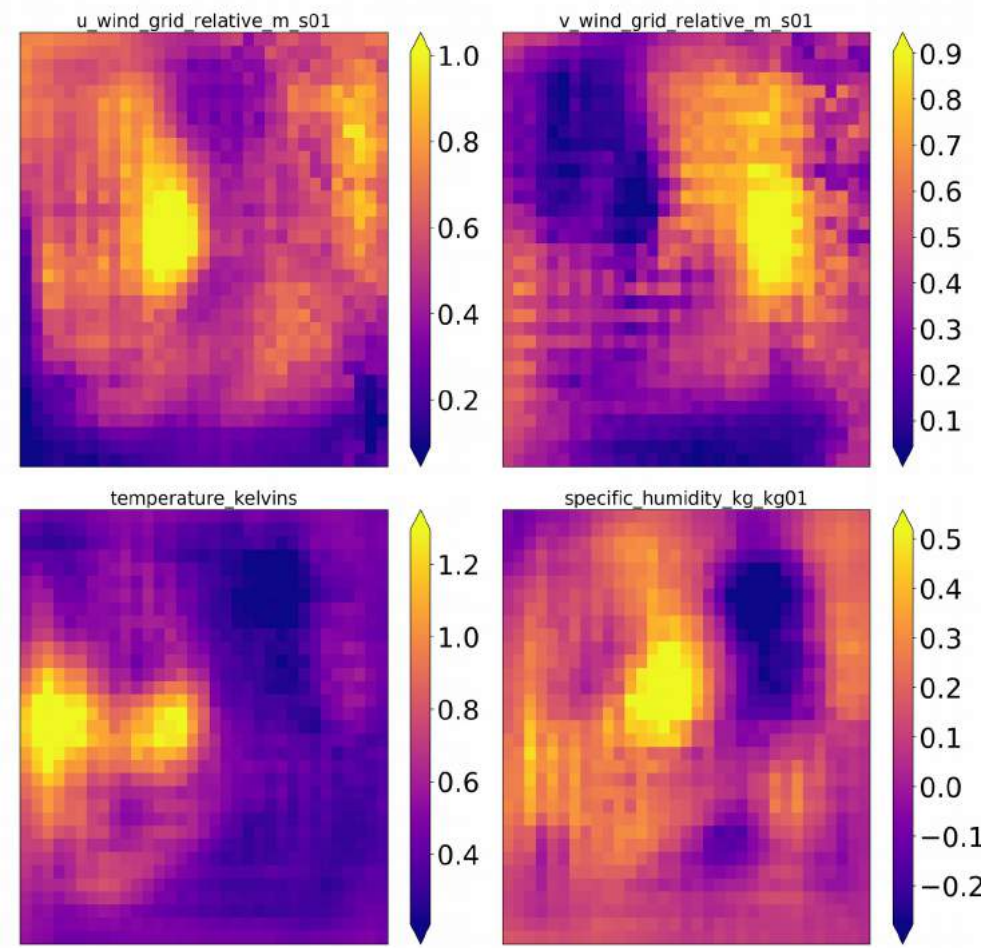


2b. Backwards Optimization

Original dataset example



Optimized for warm-front probability



2c. Saliency Maps

- Saliency = gradient of model activation with respect to input value (Simonyan *et al*. 2014).

- Mathematically:

$$\text{saliency} = \left.\frac{\partial a}{\partial x}\right|_{x=x_0}$$

- a = activation of some model component
- x = predictor (one variable at one pixel)
- x_0 = actual value (in dataset example)

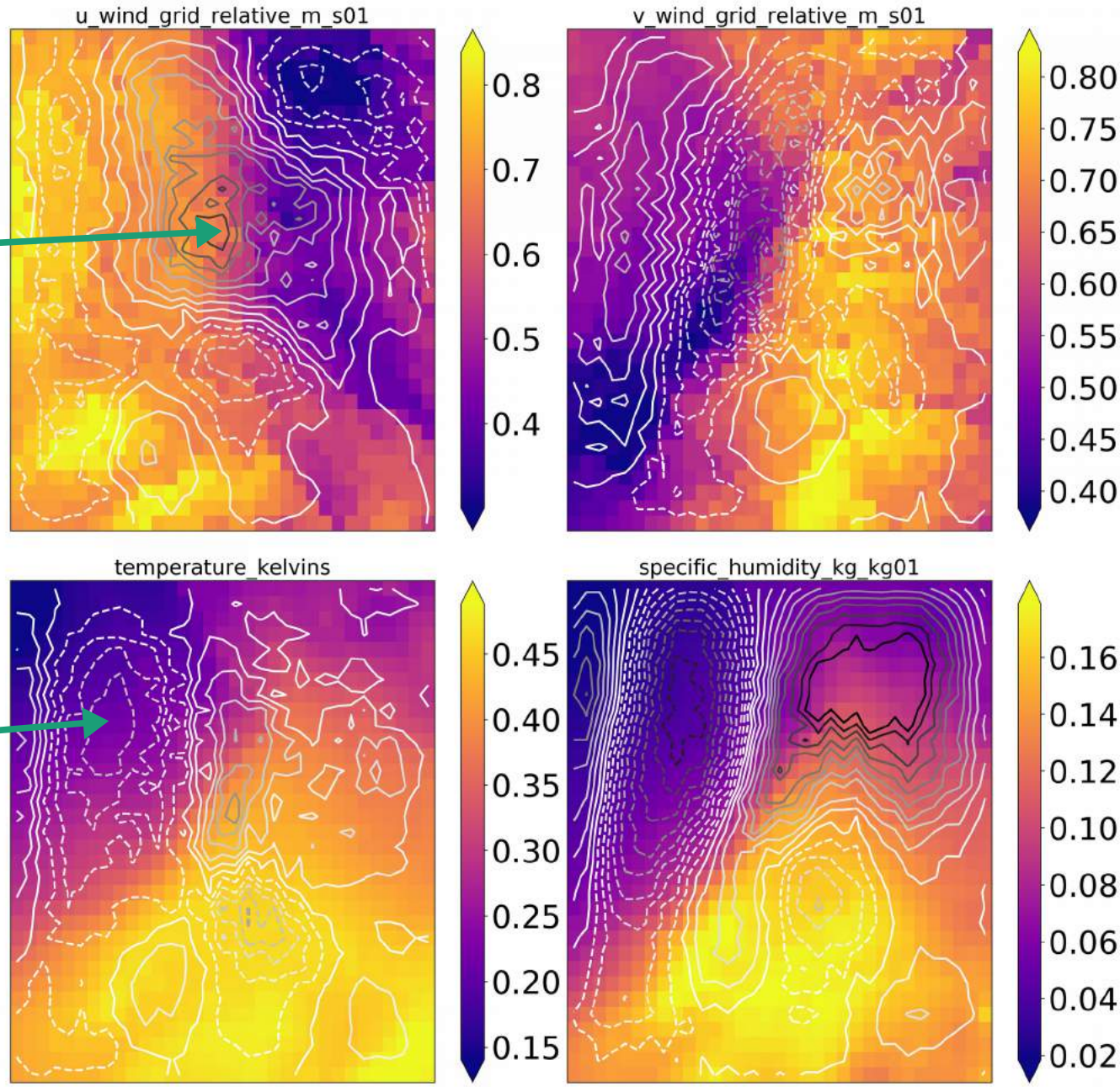
- Linear approx to $\frac{\partial a}{\partial x}$ about $x = x_0$.

- In other words, saliency tells us how model reacts when x is perturbed from x_0.

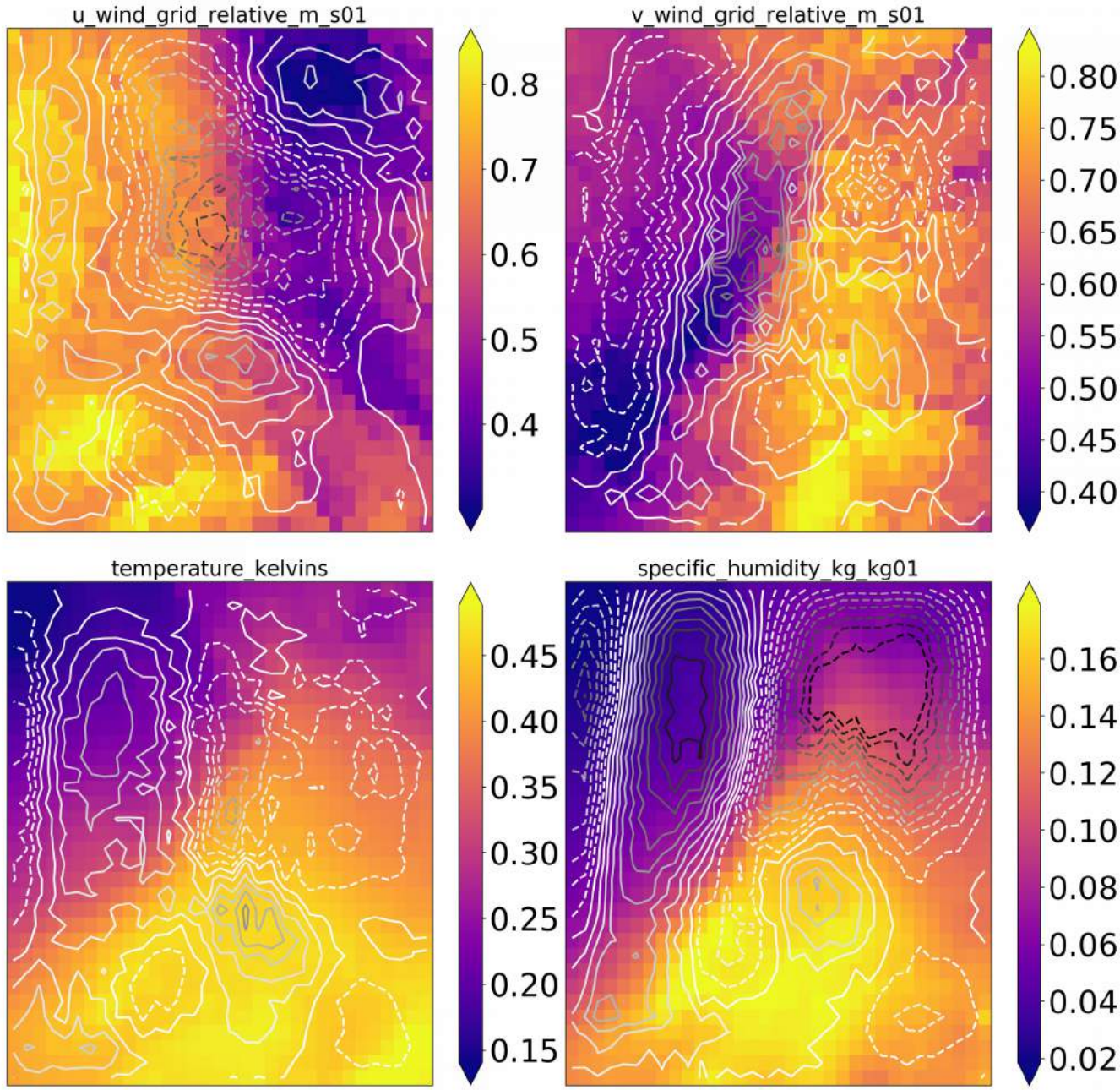
Saliency for Cold-front Probability

Solid contours: positive saliency (probability increases with value underneath).

Dashed contours: negative saliency (probability decreases with value underneath).



Saliency for Warm-front Probability



3. Large-hail Prediction

- Data source: NCAR convection-allowing ensemble (CAE).
- Consists of 10 numerical weather models with 3-km grid spacing.
- Predictors are hourly grids of the following variables at pressure levels of 850, 700, and 500 mb:
  - Geopotential height
  - *u*-wind
  - *v*-wind
  - Temperature
  - Dewpoint
- Label = “does storm produce hail $\geq$ 25 mm in next hour”?
- Task is binary classification.

3. Large-hail Prediction

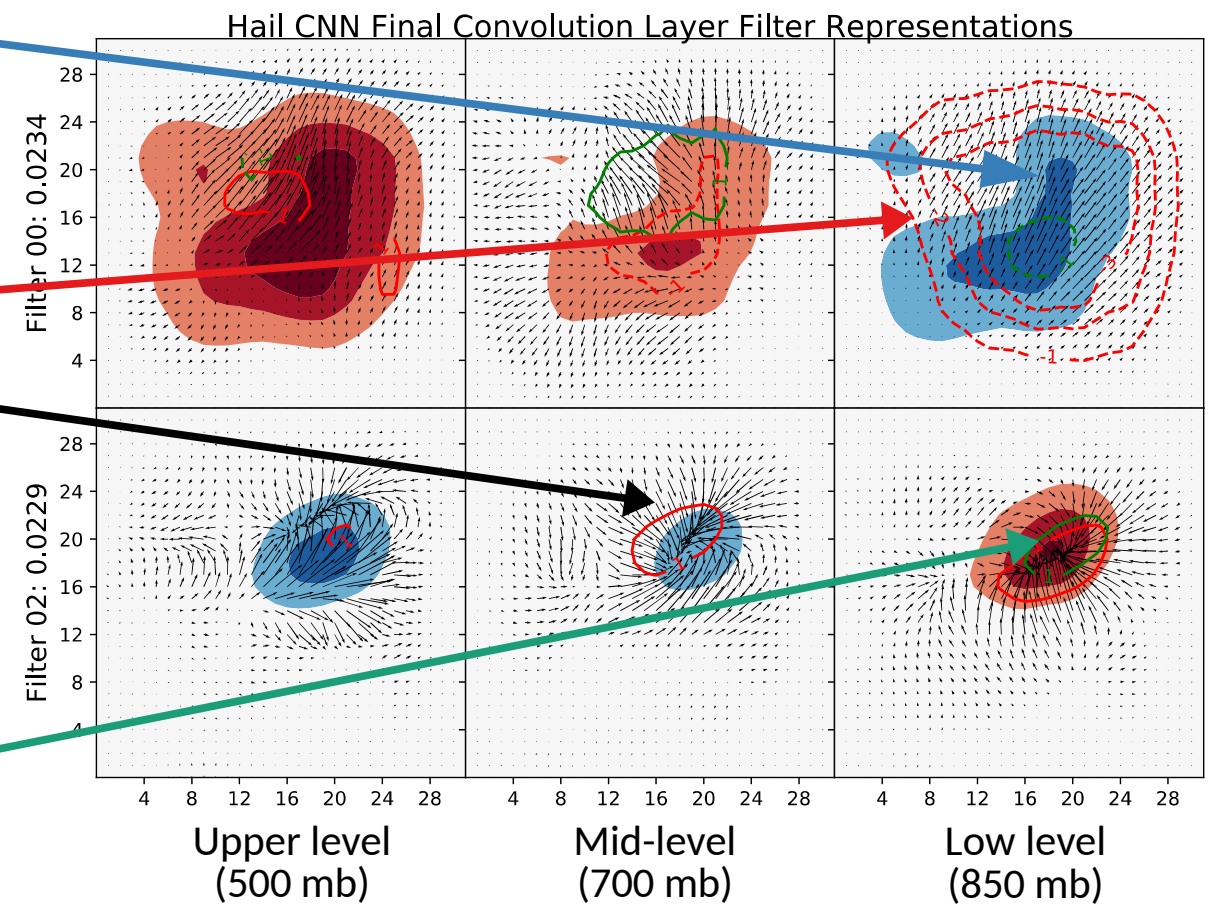
- Below: backwards optimization for two filters in final convolution layer.
- Filter 0 is maximally activated by bow-echo-like storm.
- Filter 2 is maximally activated by supercell-like storm.
- Suggests both have learned high-level representations used by meteorologists.

Height anomaly (shading) shows elongated storm structure.

Temperature-anomaly contours show surface cold pool.

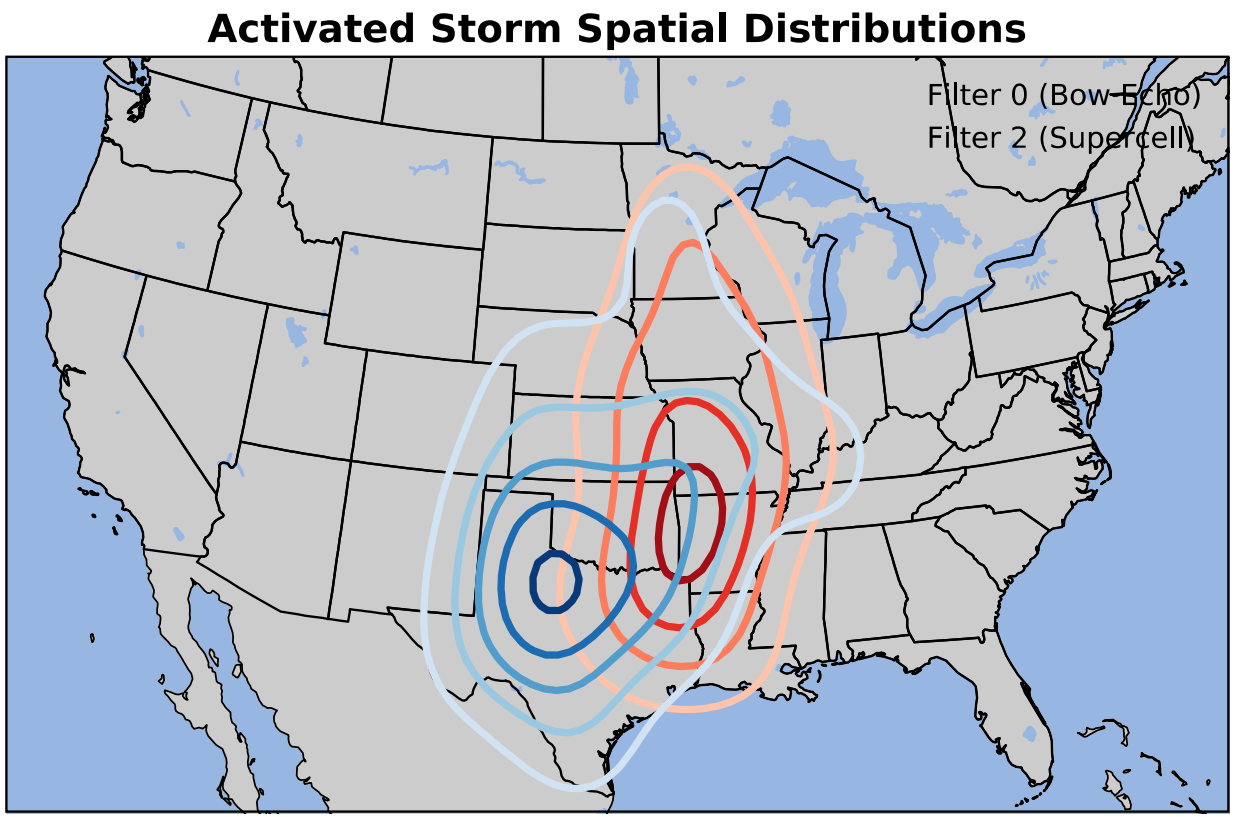
Wind-anomaly vectors show strong low-to-mid-level rotation.

Dewpoint-anomaly contours show low-level moistening.



3. Large-hail Prediction

- Below: locations of storms that activate each filter.
- Supercell-like storms most common in southern/western plains.
- Bow-echo-like storms most common in Mississippi Valley.
- Similar to previous studies (bow echoes tend to occur further east).



4. Storm-mode Classification

- Different thunderstorm morphologies ("storm modes") associated with different types of severe weather.

- Goal is to classify storm mode in real time.
  - **Disorganized convection:** mainly wind
  - **QLCS** (quasi-linear convective systems): mainly wind, sometimes hail and tornadoes
  - **Supercells:** all

- Task is 3-class classification.

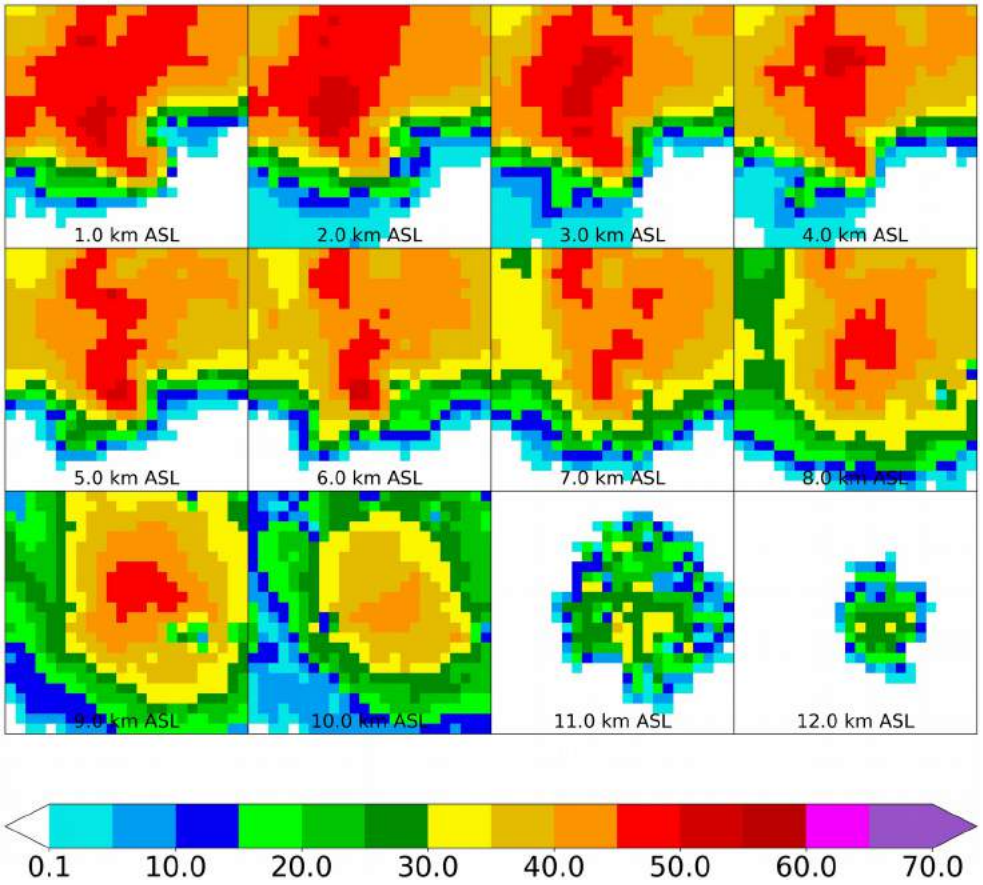
- Predictors come from 1-km radar grid and proximity sounding.

- 381 features derived from predictors.

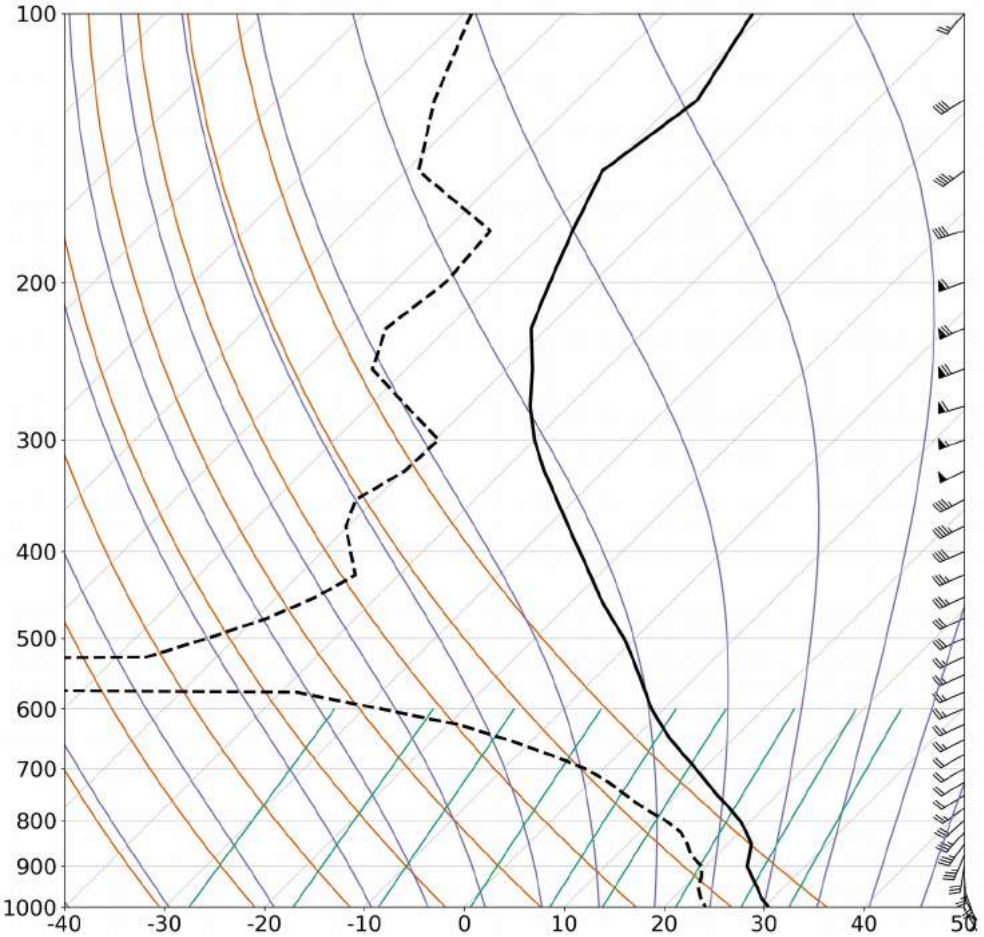
- Labels come from human meteorologists.

4. Storm-mode Classification

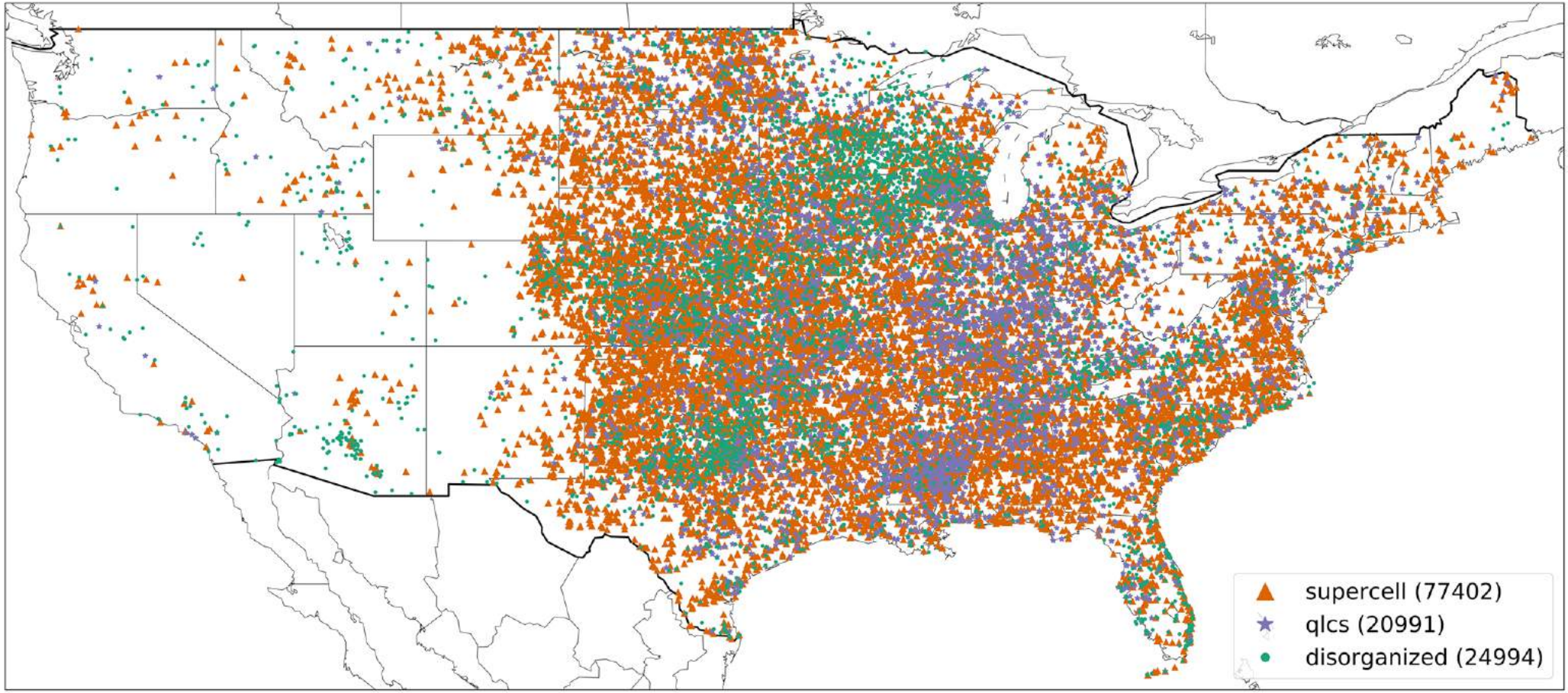
1-km radar grid from MYRORSS
(Multi-year Reanalysis of Remotely Sensed Storms)



Sounding from nearest grid cell in 13-km RAP
(Rapid Refresh weather model)



4. Storm-mode Classification



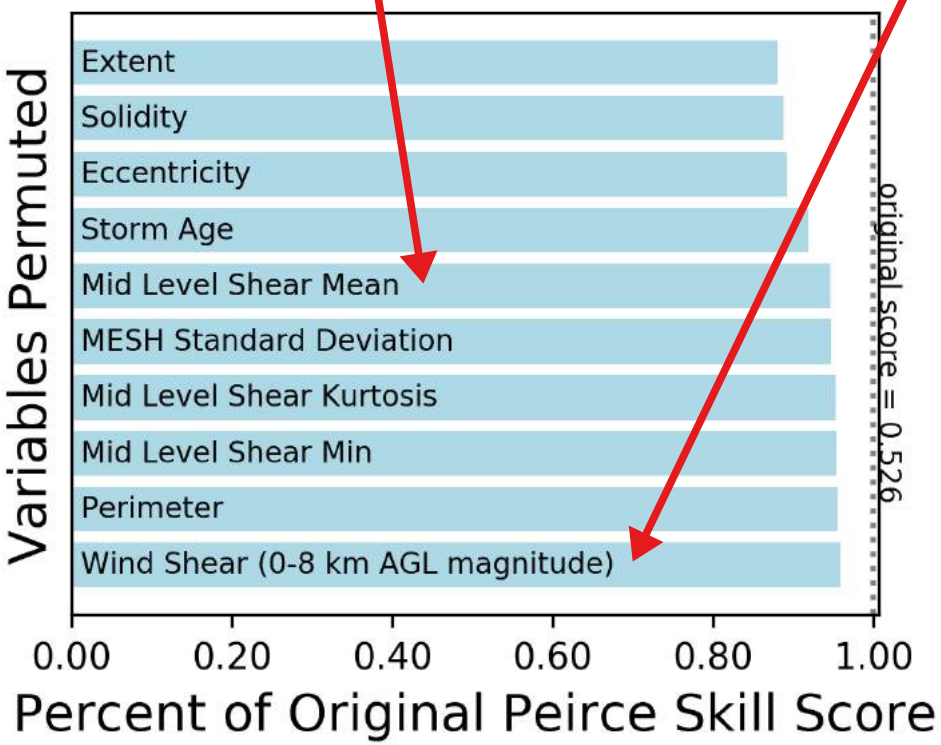
4. Storm-mode Classification

- Most important predictor at top in both graphs.
- Shape parameters (only 7 of 381 predictors) are very important.
- Wind shear and CAPE (convective available potential energy) also v. important.

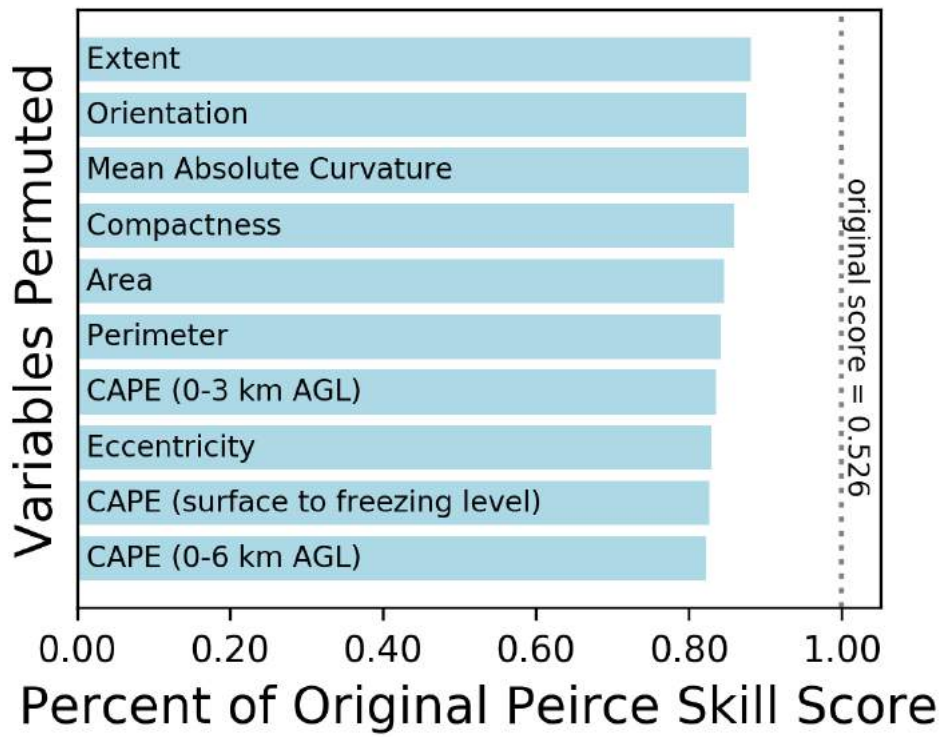
"Mid-level shear" = radar-detected horizontal shear (indicates rotation)

"Wind shear" = vertical shear in ambient environment (conducive to storm rotation)

Breiman (single-pass) results



Lakshmanan (multi-pass) results

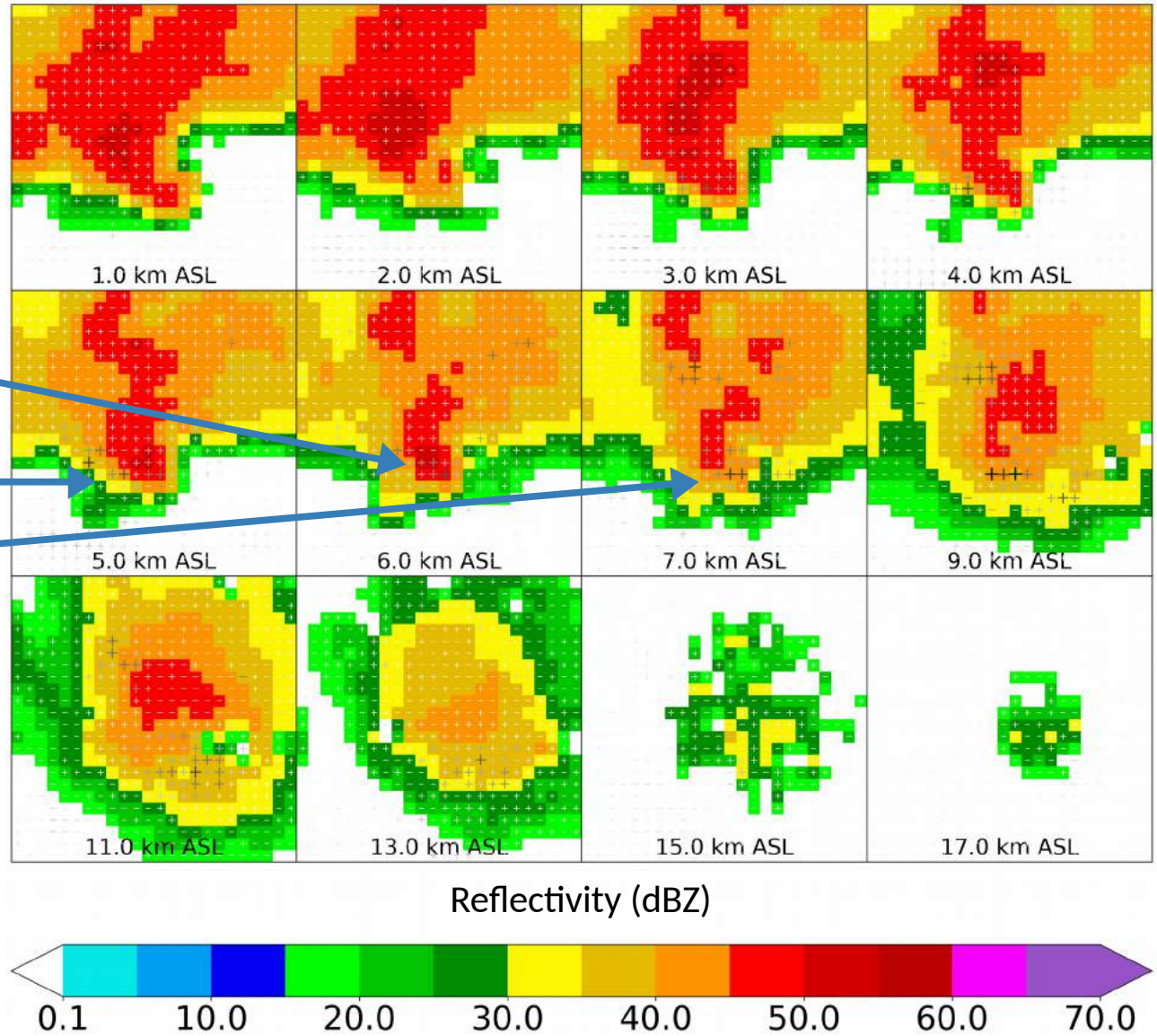


5. Tornado Prediction

- Predictors: 1-km radar grid and proxy sounding (same as storm-mode classification)
- Label = “does storm produce tornado in next hour”?
- Task is binary classification.
- We use 3-D convolutional neural net.

Saliency Maps for “Storm A”

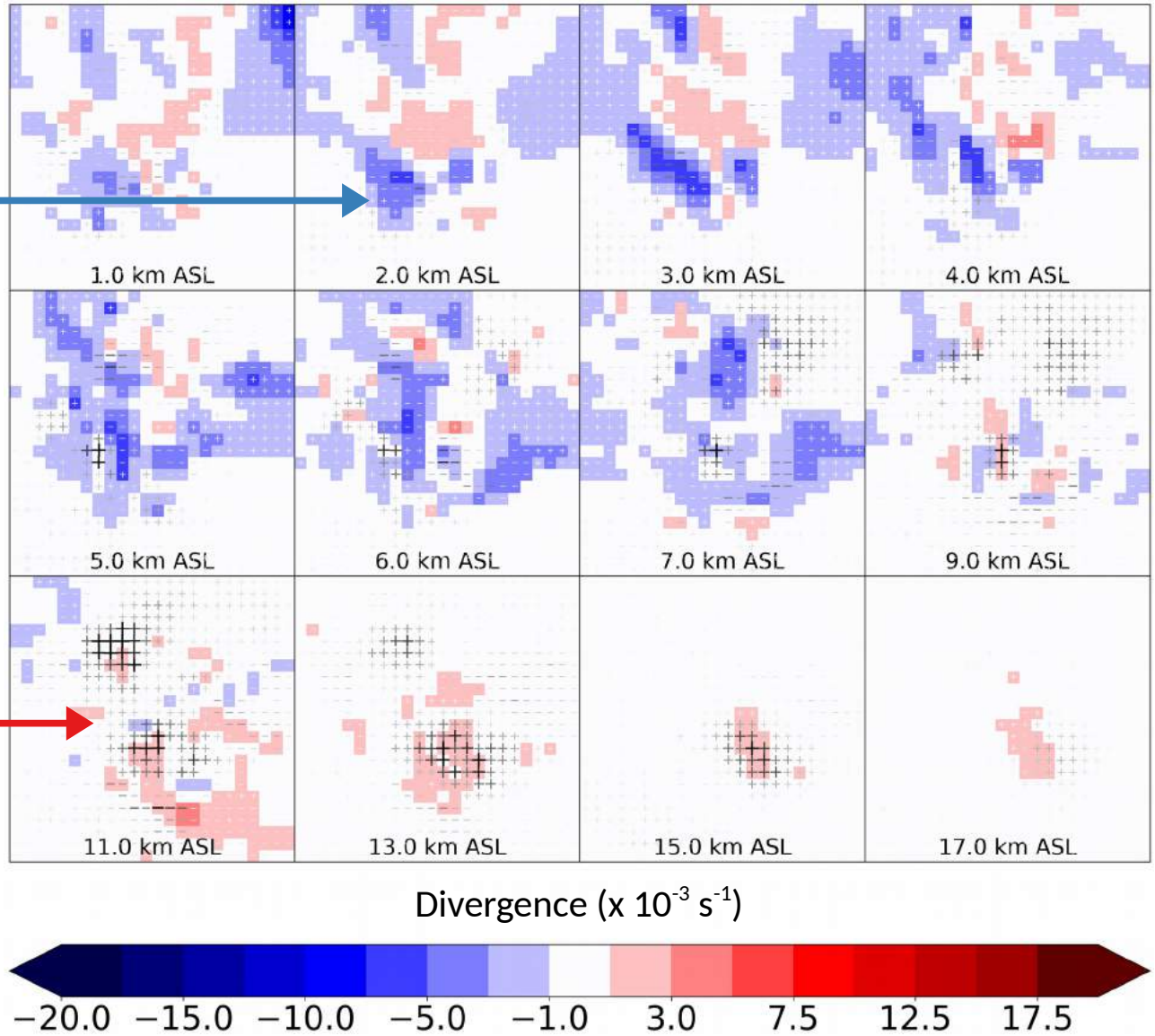
Tornado probability increases with reflectivity in mid-level hook echo.



Saliency Maps for “Storm A”

Tornado probability decreases with low-level divergence (increases with convergence).

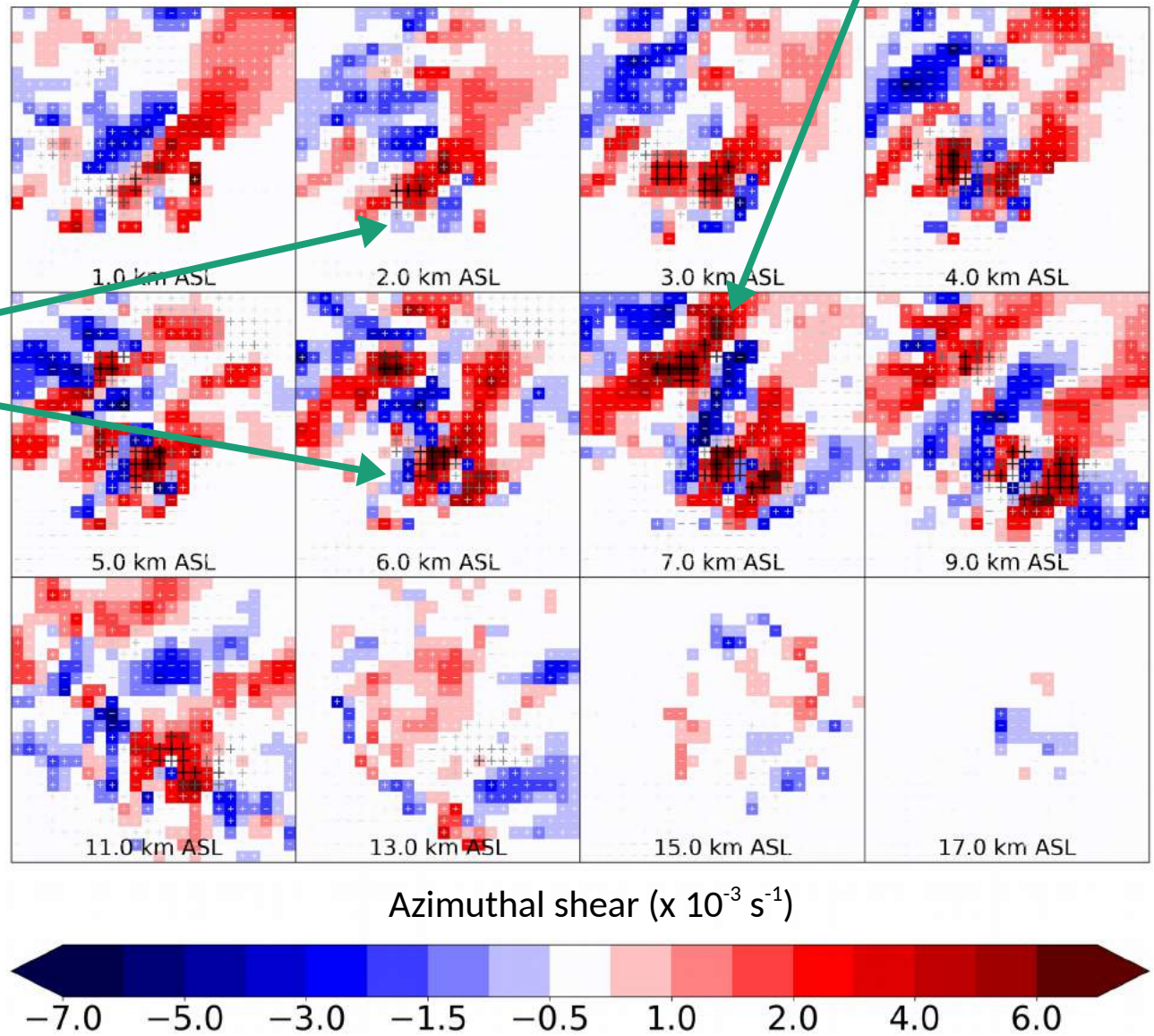
Tornado probability increases **strongly** with upper-level divergence.



Saliency Maps for "Storm A"

Tornado probability increases **strongly** with shear in rear-flank downdraft.

Tornado probability increases **strongly** with shear (rotation) in low-to-mid-level updraft.



References

Breiman, L., 2001: “Random forests.” *Machine Learning*, **45 (1)**, 5–32.

Cintineo, J., M. Pavolonis, J. Sieglaff, D. Lindsey, L. Cronce, J. Gerth, B. Rodenkirch, J. Brunner, and C. Gravelle, 2018: “The NOAA/CIMSS ProbSevere model: Incorportation of total lightning and validation.” *Weather and Forecasting*, **33 (1)**, 331-345.

Lakshmanan, V., C. Karstens, J. Krause, K. Elmore, A. Ryzhkov, and S. Berkseth, 2015: “Which polarimetric variables are important for weather/no-weather discrimination?” *Journal of Atmospheric and Oceanic Technology*, **32 (6)**, 1209–1223.

Olah, C., A. Mordvintsev, and L. Schubert, 2017: “Feature visualization.” *Distill*.

Simonyan, K., A. Vedaldi, and A. Zisserman, 2014: “Deep inside convolutional networks: Visualizing image classification models and saliency maps.” *arXiv e-prints*, **1312 (6034)**.

Code Links

- PermutationImportance
  - Library for permutation test (Breiman and Lakshmanan versions)

- SwirlNet notebook
  - Has code for backwards optimization and saliency maps.

- AIML Symposium notebook
  - Used for 2018 Artificial Intelligence and Machine Learning Symposium at U Oklahoma.
  - Has code for saliency maps and Breiman permutation test.

- GewitterGefahr
  - End-to-end machine-learning library for predicting thunderstorm hazards.
  - Caution: changes frequently.
  - Permutation test (Breiman and Lakshmanan versions)
  - Backwards optimization
  - Saliency maps