Challenges in Ensuring Reproducibility for Machine Learning Weather Model Training and Deployment

David John Gagne II
National Center for Atmospheric Research
Boulder, Colorado, USA

Workshop on Correctness and Reproducibility for Climate and Weather Software
9 November 2023

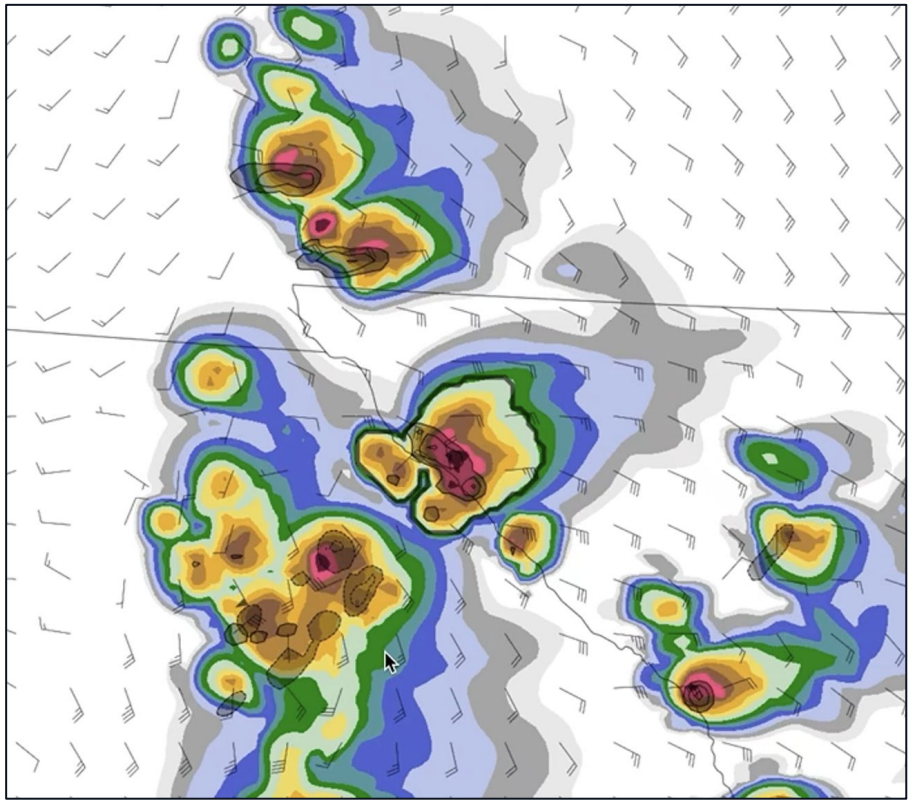
This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Motivation: Reproducing the Machine Learning Pipeline

- Many machine learning workflows in weather and climate are beginning to transition from research products toward operational systems
- Key differences between more static research settings and more dynamic operational settings may compromise the reproducibility of ML pipelines and their results
  - Cannot reproduce the same model training and predictions
  - Cannot reproduce the same level of accuracy
  - Robustness to missing data and edge cases
- Goals:
  - Identify key challenges limiting reproducibility throughout the ML pipeline
  - Discuss strategies to address these challenges based on different ML use cases

Storm Mode: Motivation

Goal: Develop techniques to objectively identify convective mode in convection-allowing models (CAMs) using machine learning (ML) algorithms.



1. **Supervised learning** using hand-labeled CAM storm objects and ML algorithms
2. **Unsupervised/semi-supervised learning** using CAM storm objects, ML, and clustering algorithms

Diagnosing Storm Mode with Deep Learning in Convection-Allowing Models

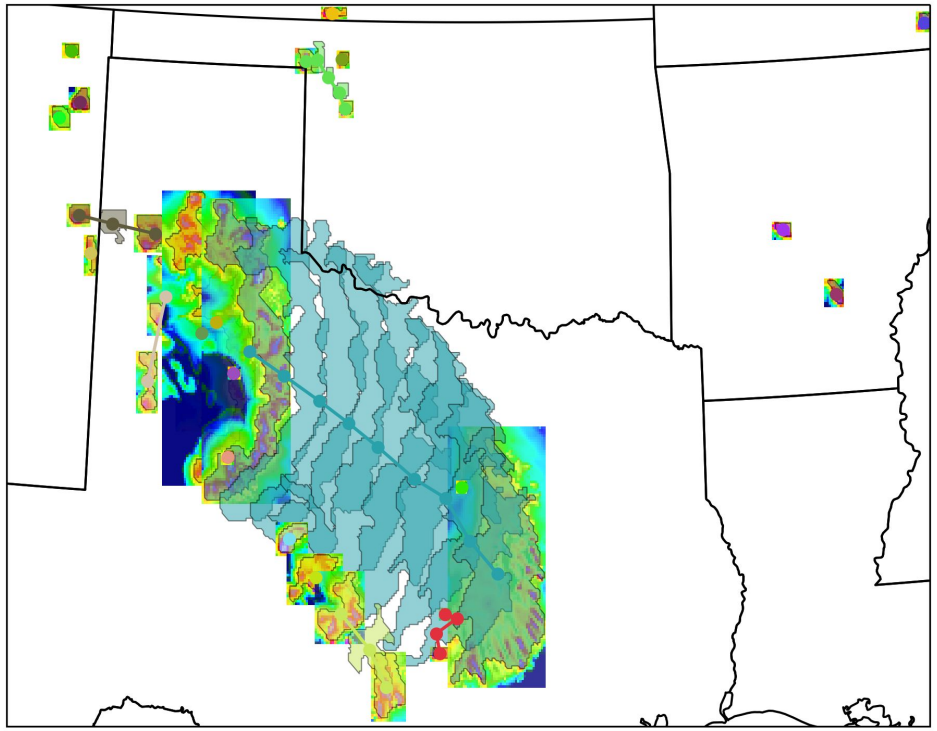
Ryan A. Sobash, David John Gagne II, Charlie L. Becker, David Ahijevych, Gabrielle N. Gantos, and Craig S. Schwartz

Online Publication: 05 May 2023

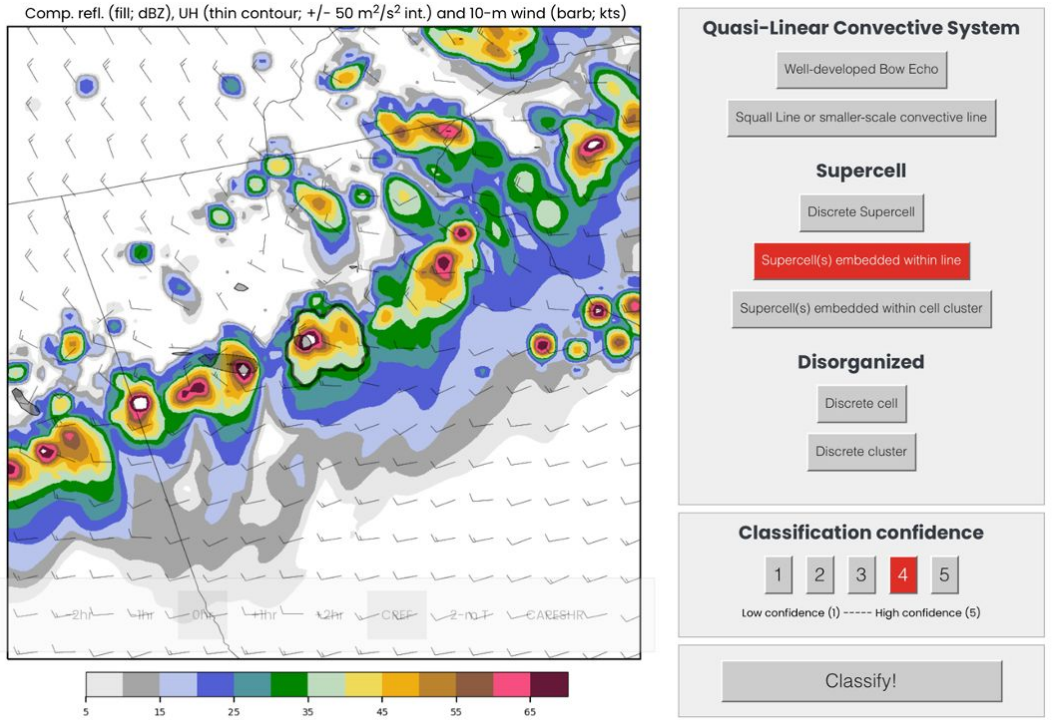
DOI: https://doi.org/10.1175/MWR-D-22-0342.1

Storm Mode: Segmentation, Tracking and Labeling

Hagelslag: segment storms and track with image processing techniques



Hand Labeling: web interface



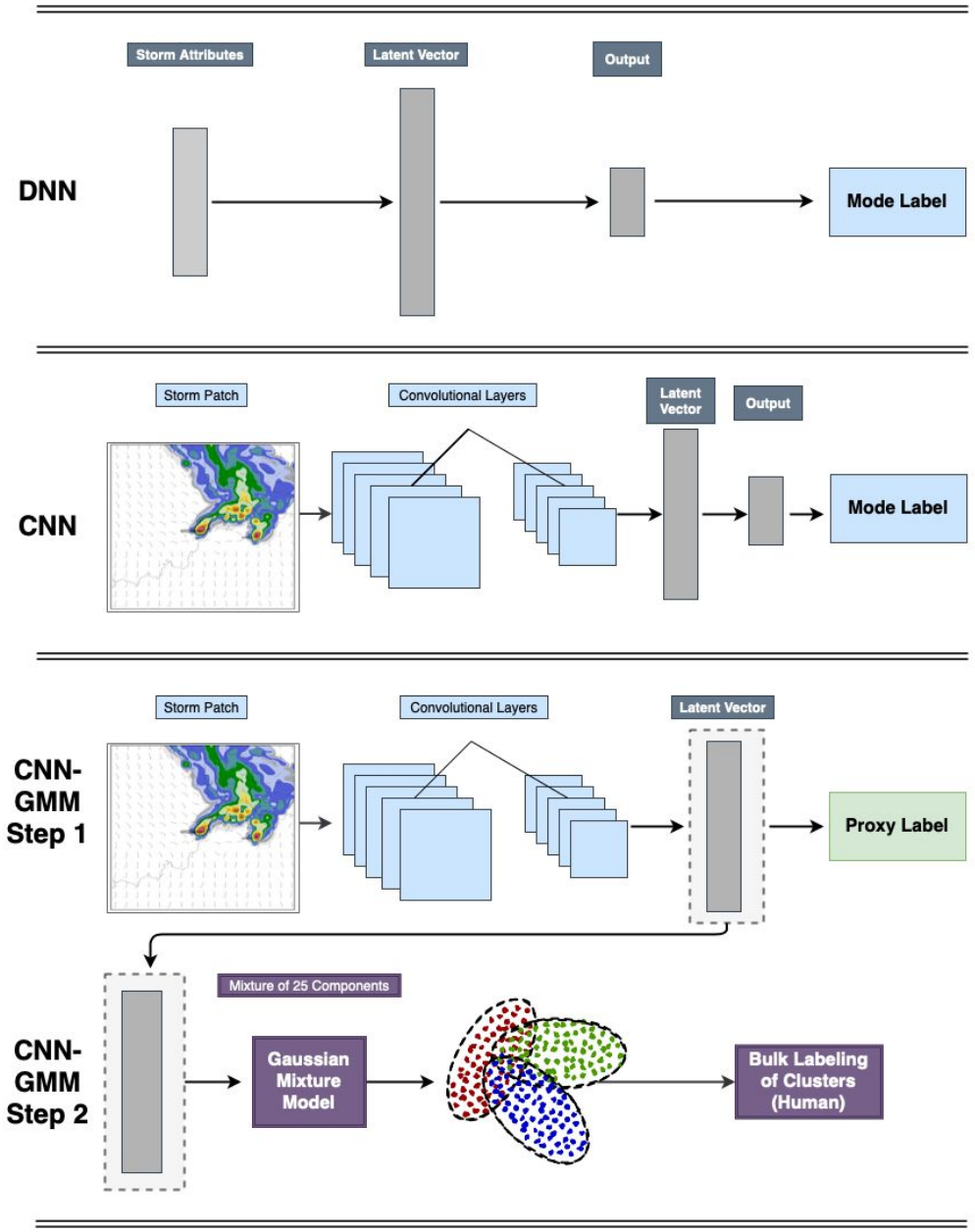
Challenge: upgrades/bug fixes to segmentation algorithm resulted in differing numbers and locations of segmented storms from hand-labeled dataset.

Solution 1: Run original hand labeled data through models for evaluation and not re-train.

Solution 2: Use matching algorithm to link old labeled storms to new storms.

Solution 3: Use training and evaluation approaches that require minimal hand-labeling.

Model Architectures



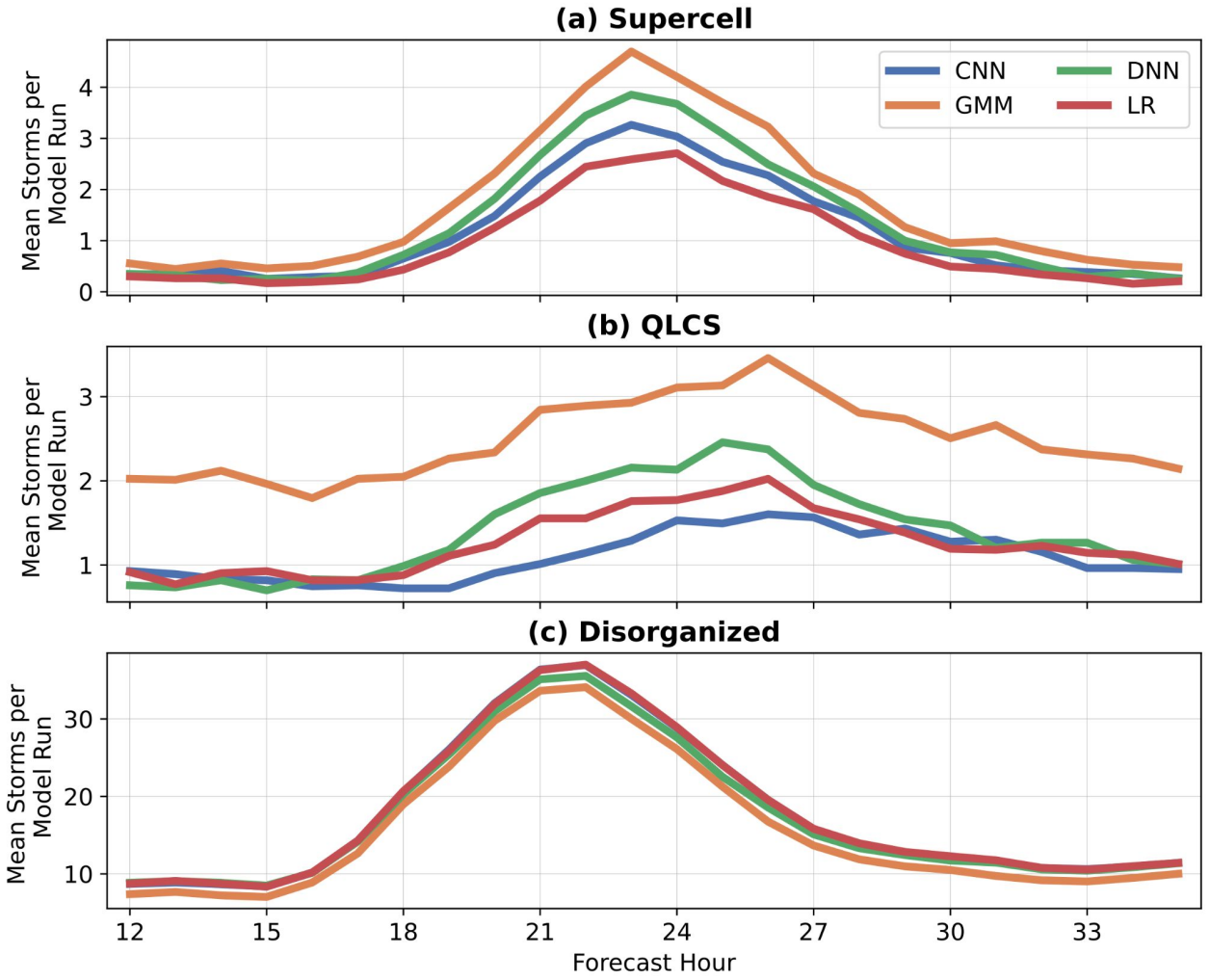
Challenge: Storm does not fit entirely within image patch

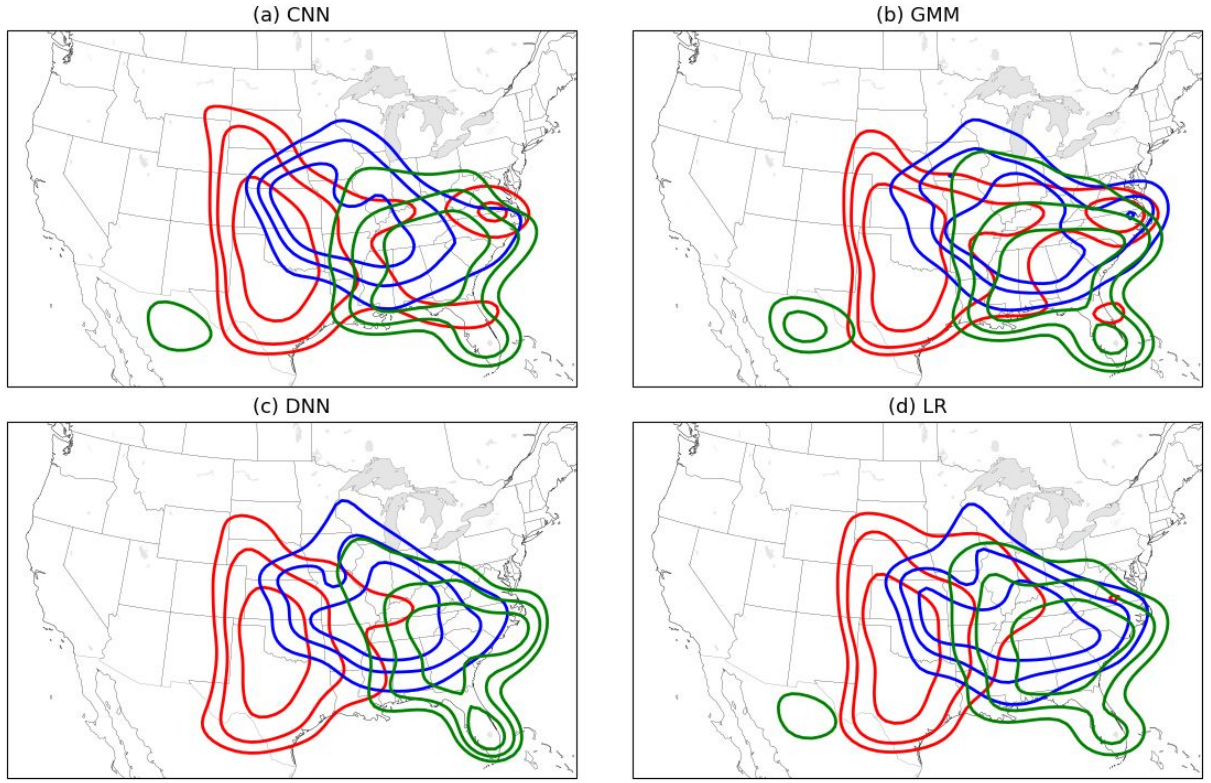
Solutions: use summary metrics based on full storm extent, expand patch, use a grid-based segmentation

Challenge: Pre-processing pipeline changed during project

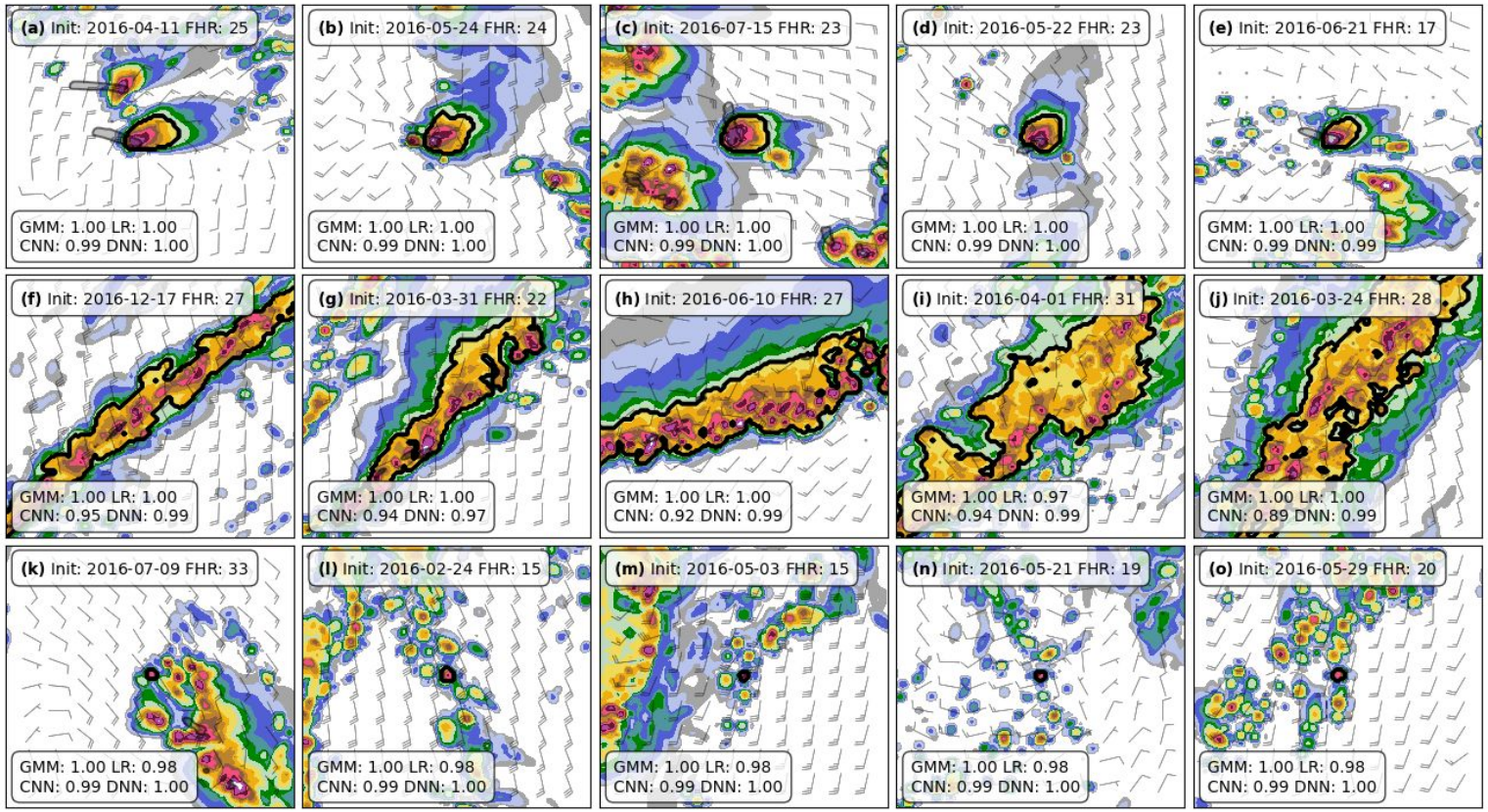
Solutions: relabel quickly using proxy labels and bulk labeling of storms based on clusters

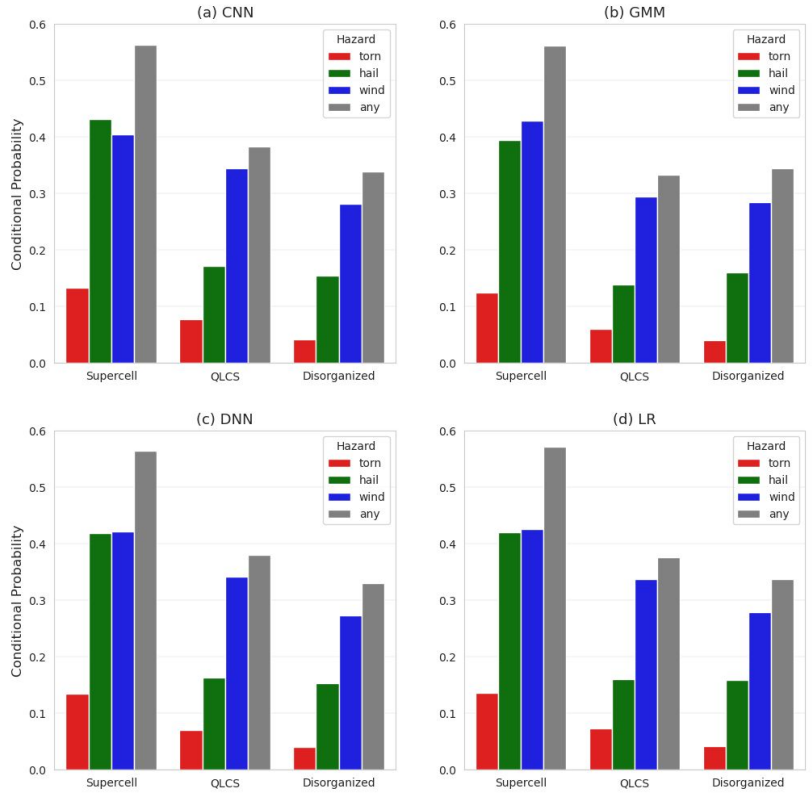
Spatial and Temporal Trends in Mode





Storms with Predicted Mode Agreement

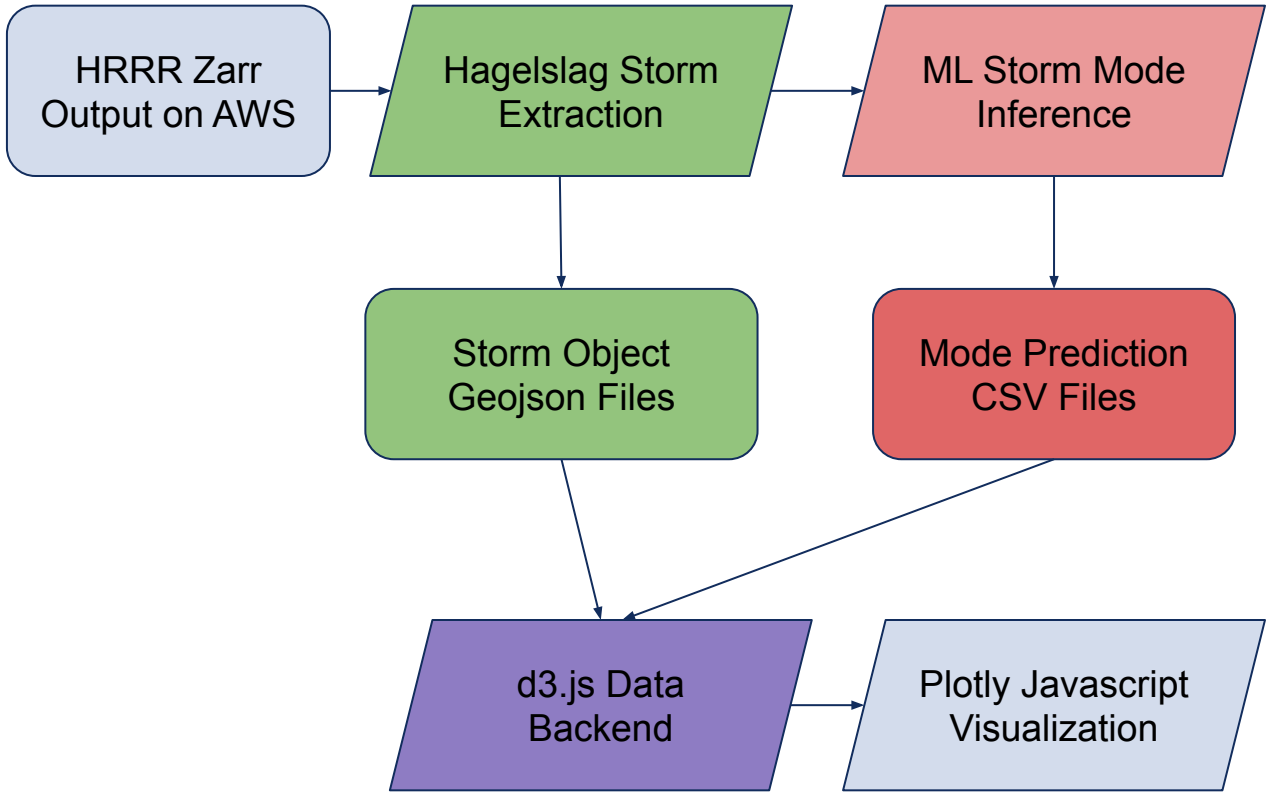


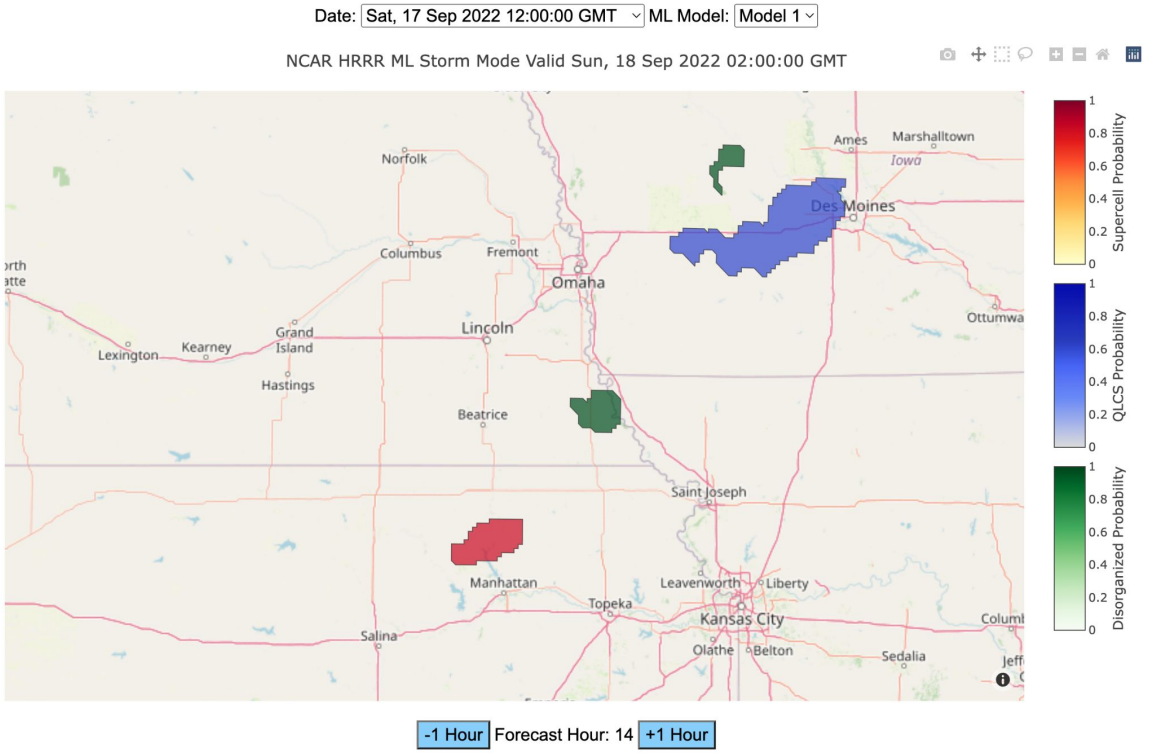


Challenge: evaluate storm mode models without a massive hand labeling effort

Solution: examine consistency among storm labels and conditional probability of different severe hazards given mode

Storm Mode Visualization Pipeline





ncar.github.io/modeview

Challenges Reproducing Scaling of Data

Challenge: re-scaling data is lossy due to floating point arithmetic

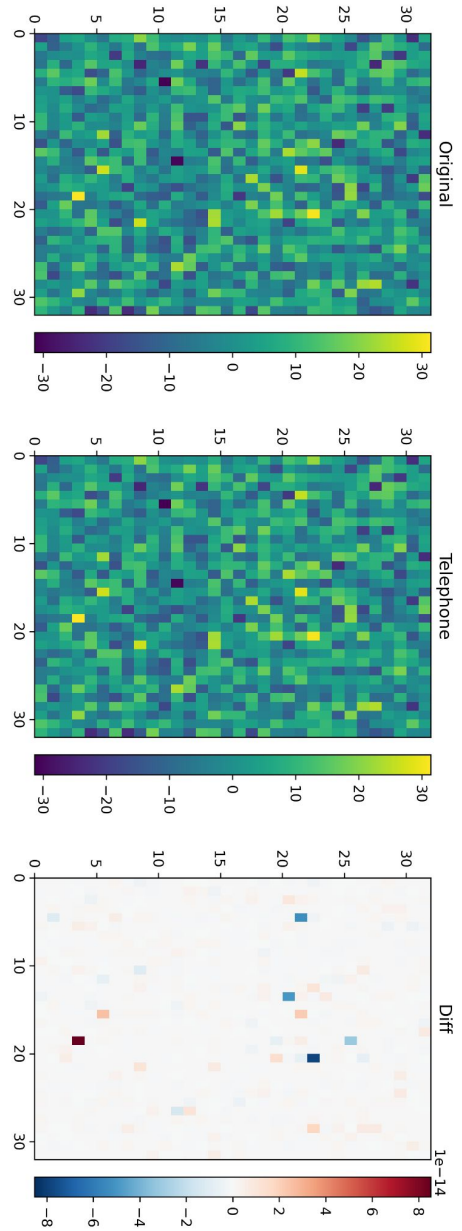
- Amount of lossiness depends on the variance of the data
- Quantile scaling is more adaptable than standard scaling

Challenge: transferring scaling values between different systems

- Scikit-learn requires users to serialize their Scaler objects
- NCAR's bridgescaler package saves scikit-learn-like scaling objects to json files for more reproducible scaling

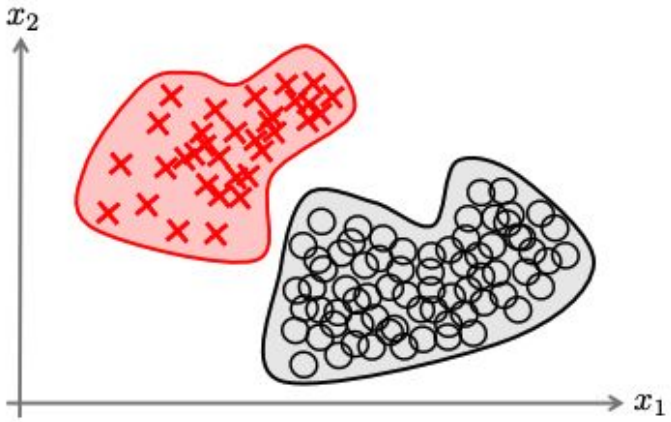
Challenge: calculating scaling values across distributed datasets

- Bridgescaler has added distributed methods for standard scaler and min max scaler
- Floating point math makes the distributed solution approximately the same as the local solution
- T-digest method allows for approximate distributed calculation of quantiles (https://github.com/tdunning/t-digest) Note: Python version not working with >3.8



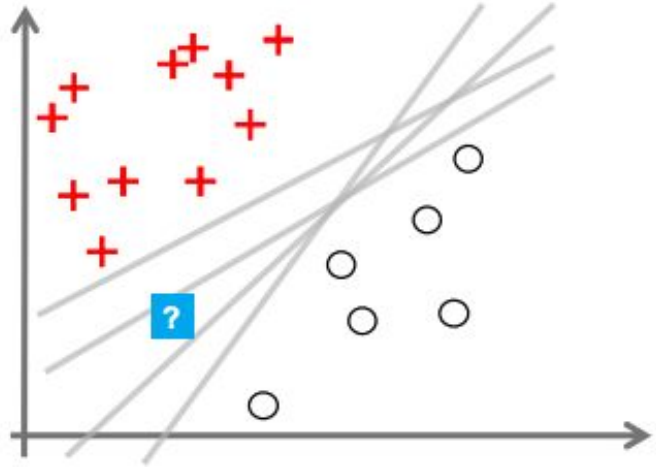
Decomposition of Uncertainty

Aleatoric Uncertainty



Definition: Uncertainty from variation in data.
Estimated by: Single probabilistic AI model.
Reduce by: Gather more informative inputs

Epistemic Uncertainty



Definition: Uncertainty from variation in models.
Estimated by: Ensemble of deterministic AI models.
Reduce by: Gather more examples or use simpler models.

Total Uncertainty

Definition: Combined aleatoric and epistemic uncertainty.
Estimated by:

1) Ensemble of probabilistic AI models
2) Single “evidential” (higher-order probabilistic) AI model
3) Bayesian AI models

Collaborators

John Schreck, Charlie Becker, Gabrielle Gantos, Julie Demuth, Chris Wirz, Jacob Radford, Nick Bassil, Kara Sulia, Chris Thorncroft, Amy McGovern, Eliot Kim, Justin Willson, Kim Elmore, Maria Molina

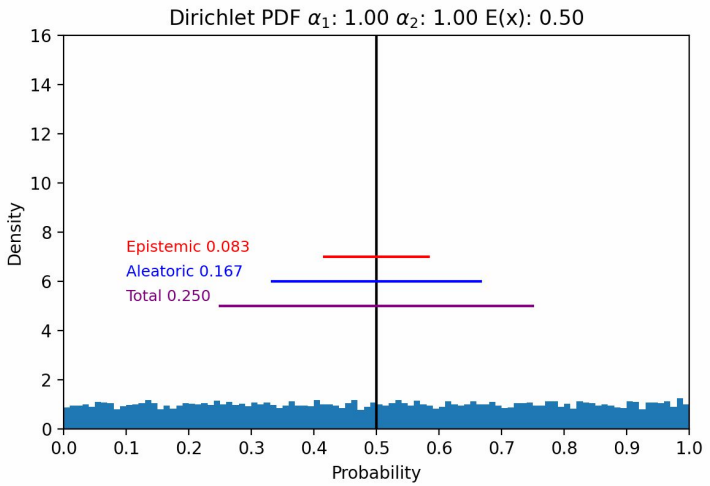
Dirichlet Aleatoric and Epistemic Uncertainties

Law of total variance decomposes the total uncertainty into the sum of the unexplained variance plus the explained variance:

$$\mathrm{Var}(y_j) = \mathbb{E}\left(\mathrm{Var}(y_j|\boldsymbol{p})\right) + \mathrm{Var}\left(\mathbb{E}(y_j|\boldsymbol{p})\right)$$

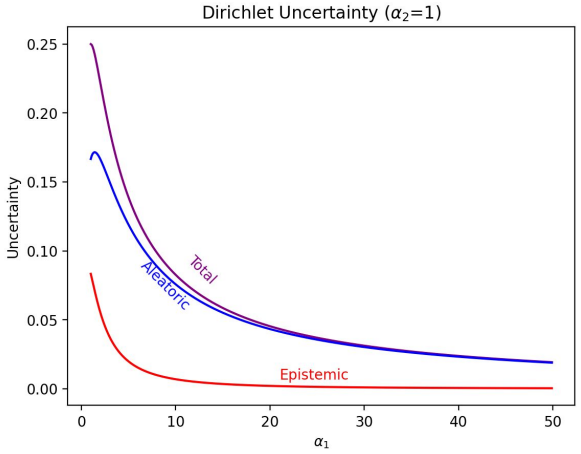
Aleatoric (unexplained) =

$$\begin{aligned}\mathbb{E}\left\{\mathrm{Var}(y_j|\boldsymbol{p})\right\} &= \mathbb{E}\left\{p_j(1-p_j)\right\} \\ &= \mathbb{E}(p_j) - \mathbb{E}(p_j^2) \\ &= \mathbb{E}(p_j) - \{\mathbb{E}(p_j)\}^2 - \mathrm{Var}(p_j) \\ &= \frac{\alpha_j}{S} - \left(\frac{\alpha_j}{S}\right)^2 - \frac{\frac{\alpha_j}{S}\left(1-\frac{\alpha_j}{S}\right)}{S+1}\end{aligned}$$



Epistemic (explained) = $\mathrm{Var}\left\{\mathbb{E}(y_j|\boldsymbol{p})\right\} = \mathrm{Var}(p_j)$

$$= \frac{\frac{\alpha_j}{S}\left(1-\frac{\alpha_j}{S}\right)}{S+1}$$



Total = Aleatoric + Epistemic

Theory of Evidence and Subjective Logic

Dempster-Shafer Theory of Evidence (DST), a generalization of Bayesian theory of subjective probabilities, assigns ***belief masses*** to subsets of possible labels for an observation.

If belief masses for an observation are all equally likely ~ "***I do not know.***"

Subjective logic (SL) formulates *belief assignments* b_k over K classes, plus "**I don't know**", as a Dirichlet distribution (prior). For a NN with K outputs

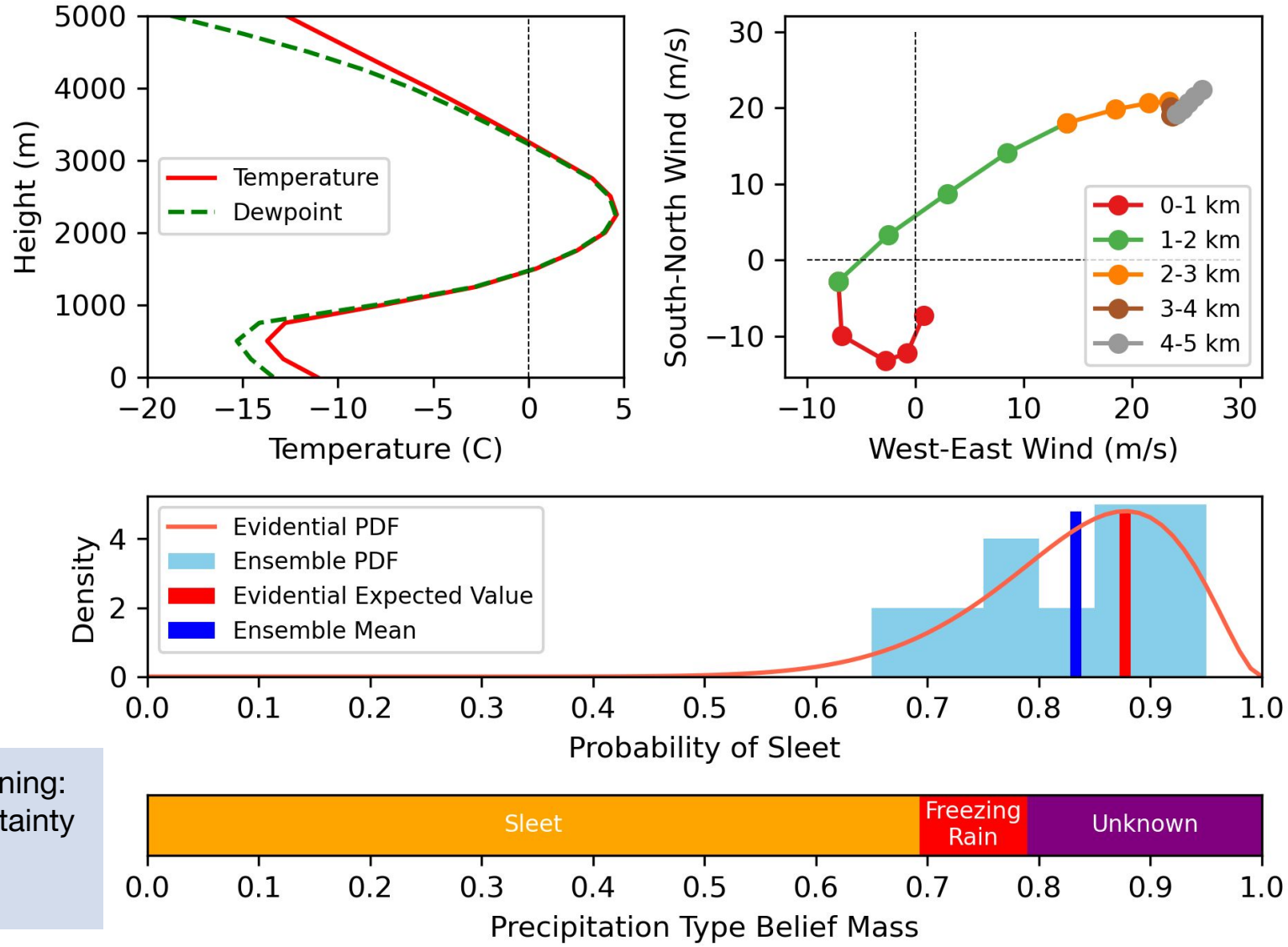
$$u + \sum_{k=1}^{K} b_k = 1$$

where b_k is the *kth* ReLU output, interpreted as the "*belief mass*" of the *kth* class, and u is the uncertainty mass of the K outputs.

Each b_k is defined as

$$b_k = \frac{e_k}{S} \quad \text{where} \quad S = \sum_{i=1}^{K} (e_k + 1) \quad \text{and thus} \quad u = \frac{K}{S}$$

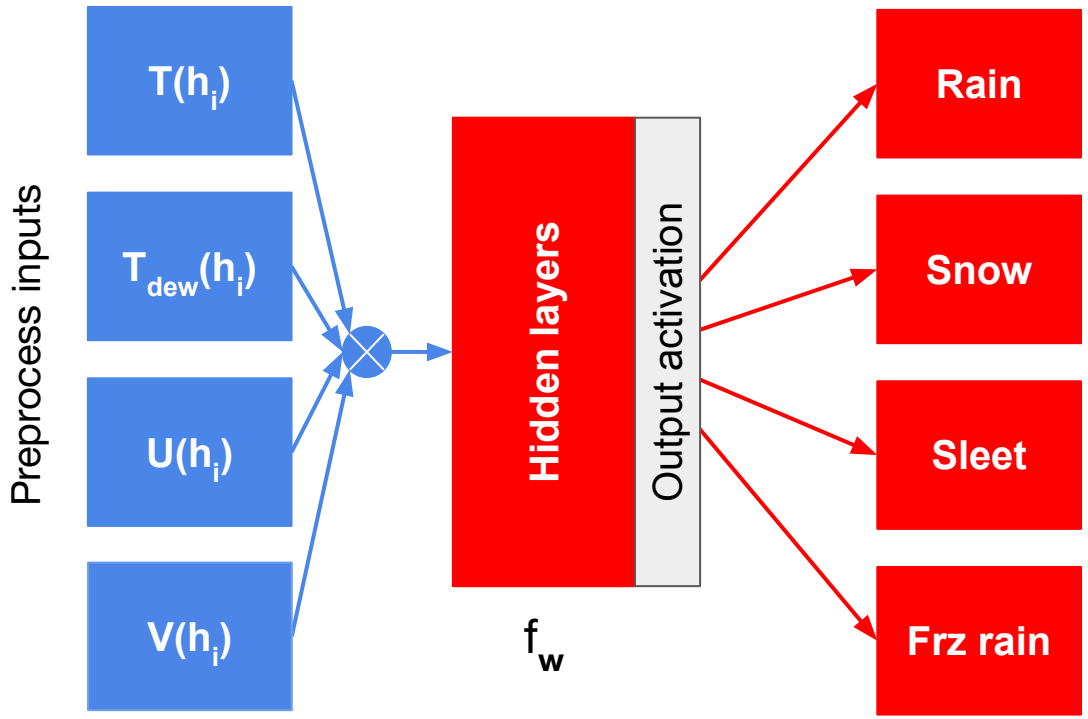
Evidential Deep Learning



Paper: Evidential Deep Learning: Enhancing Predictive Uncertainty Estimation for Earth System Science Applications

https://arxiv.org/abs/2309.13207

Evidential Model Architecture



(i) Deterministic:

Predict probabilities for classes

Loss = Cross-entropy

$p_k = \text{Softmax}(f_w(T,T_{dew},U,V))_k$

(ii) Evidential:

(Sensoy et al. 2018)
Predict evidence for classes

Loss = Evidential

$e_k = \text{ReLU}(f_w(T,T_{dew},U,V))_k$

$\alpha_k = e_k + 1$

Compute S, evidential u,
and the probabilities p_k

Probabilistic Forecast Example: Classifying Winter Precipitation Type

Data

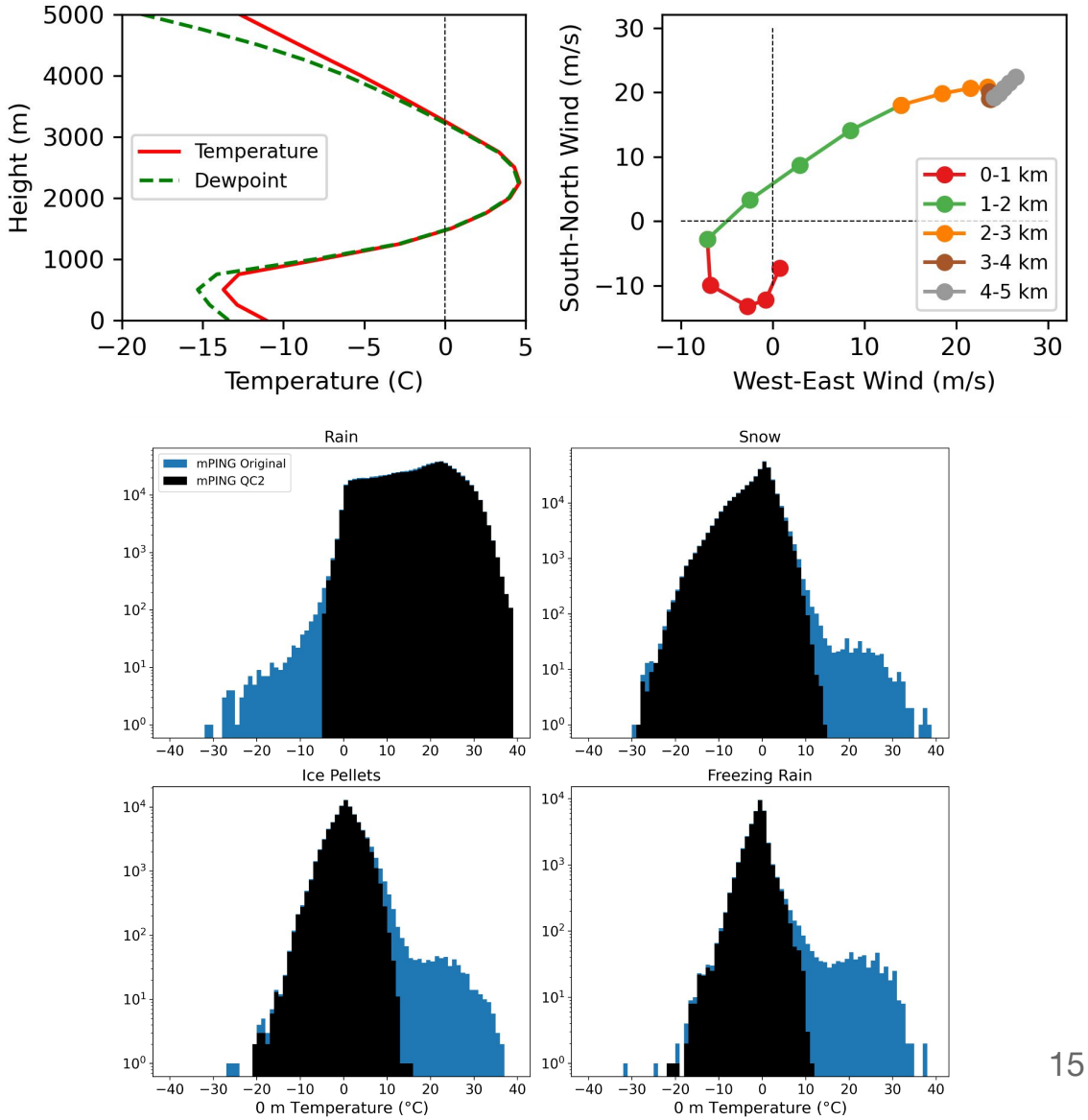
- **NOAA Rapid Refresh** Vertical Profiles
- Interpolate from pressure to height coords

Input (0 - 5 km above surface, every 250 meters)

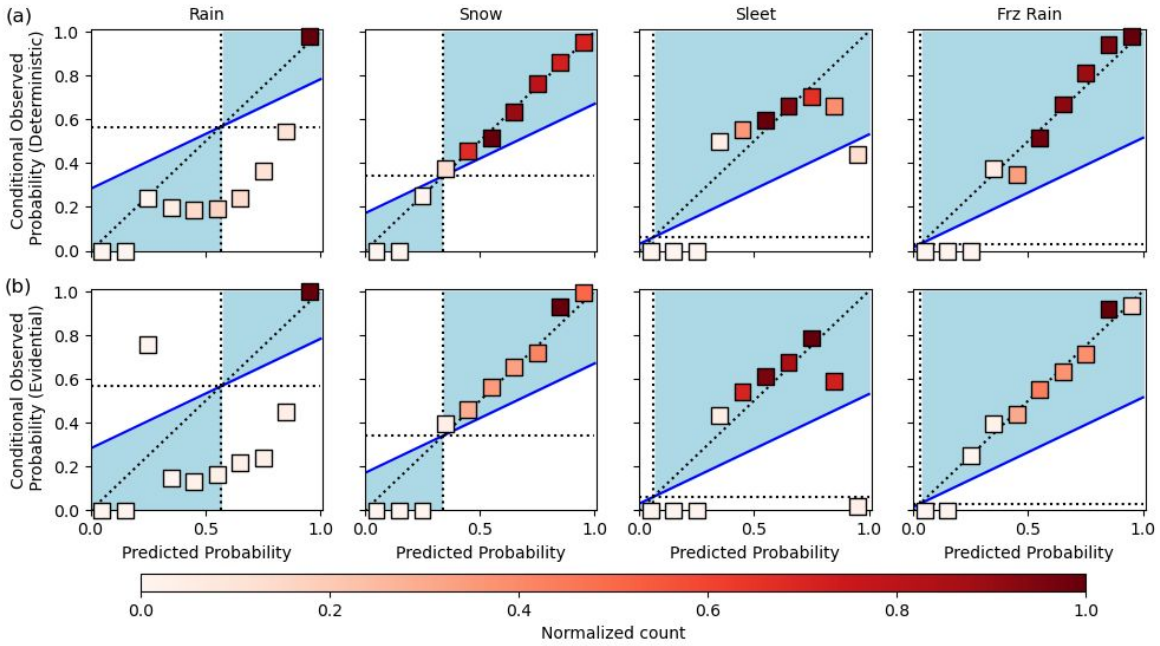
- Temperature, Dewpoint, U-Wind, V-Wind

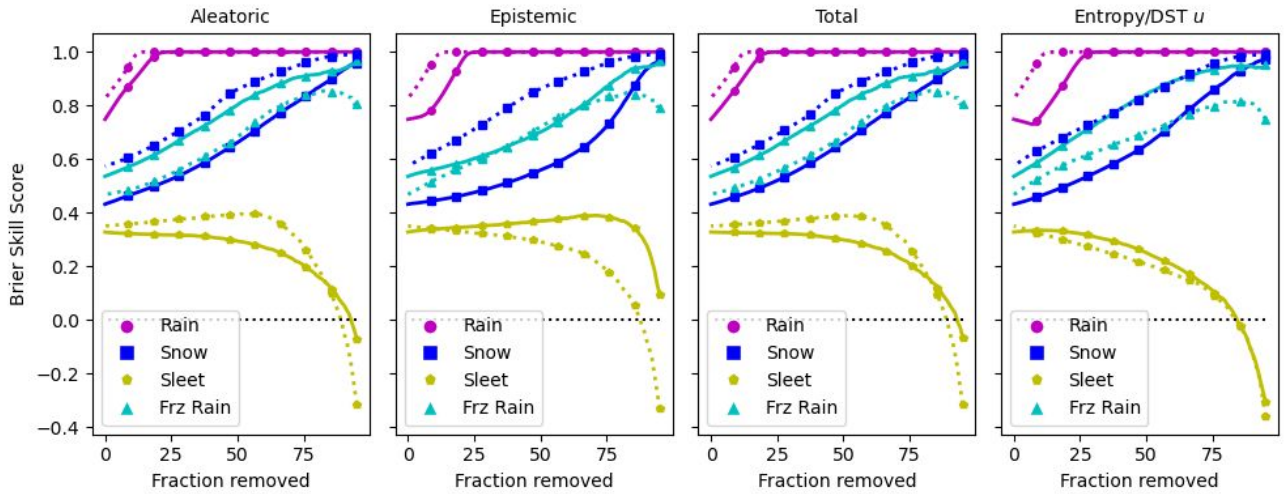
Target

- mPING Crowd-sourced reports of winter precipitation types
  - *Rain, Snow, Sleet, Freezing Rain*



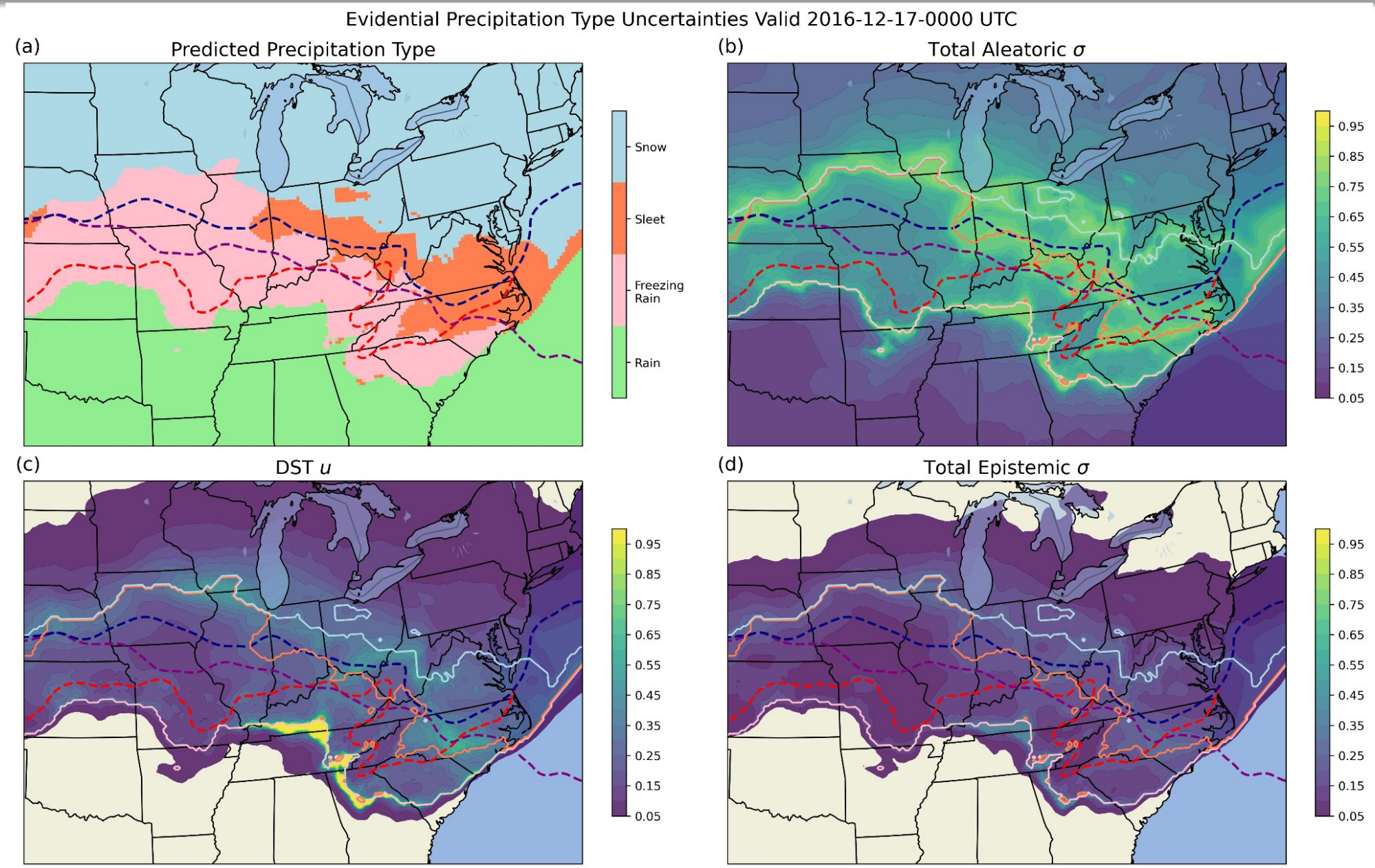
Precipitation-type Validation



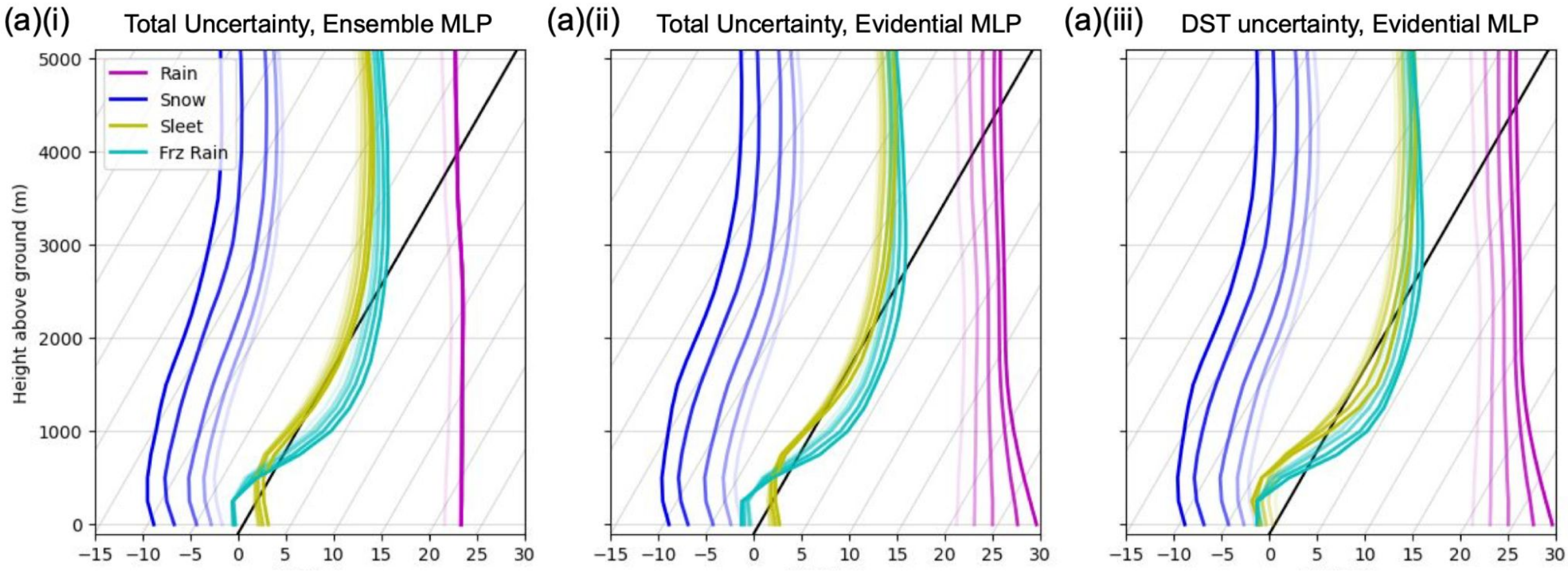


How well does each type of uncertainty discriminate between easier and harder to classify events?

Regional Case Study



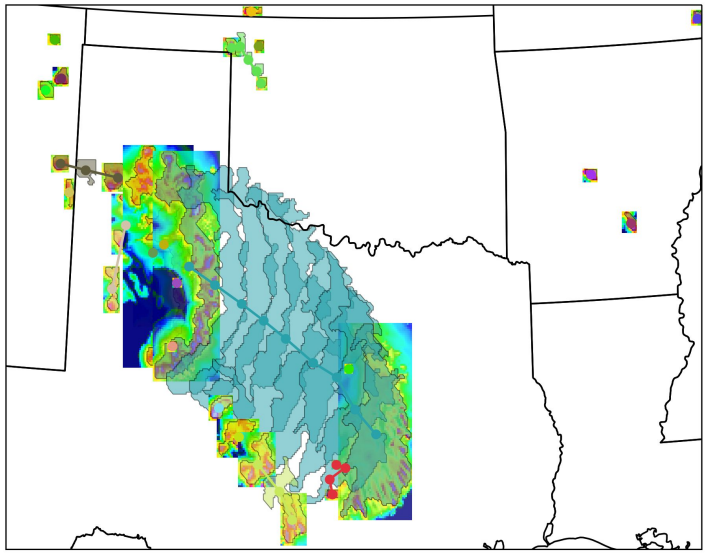
Consistency of Composite Soundings by Uncertainty Metric and Model



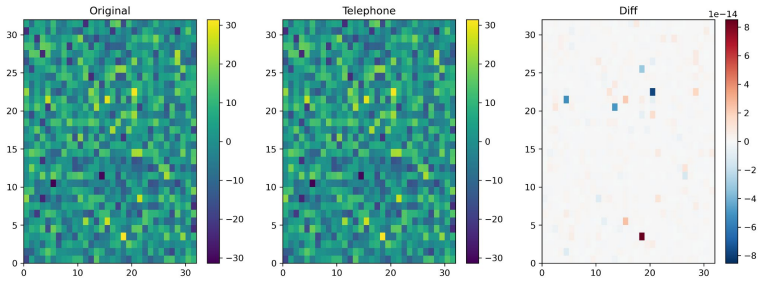
MILES Group Python Packages

- **miles-guess** (github.com/ai2es/miles-guess):
  - Implementations of evidential neural networks, deep ensembles, and Monte Carlo dropout
- **echo-opt** (github.com/NCAR/echo-opt):
  - Distributed hyperparameter optimization on HPC systems
  - Supports GPU allocation, XAI visualization for hyperparameter settings
- **hagelslag** (github.com/djgagne/hagelslag):
  - Object segmentation, tracking, and data extraction for convection-allowing model output
  - verification scores and plots
- **bridgescaler** (github.com/NCAR/bridgescaler):
  - Reproducible saving/loading of sklearn preprocessing scalers and transforms
  - Custom scalers for groups of variables and image patches

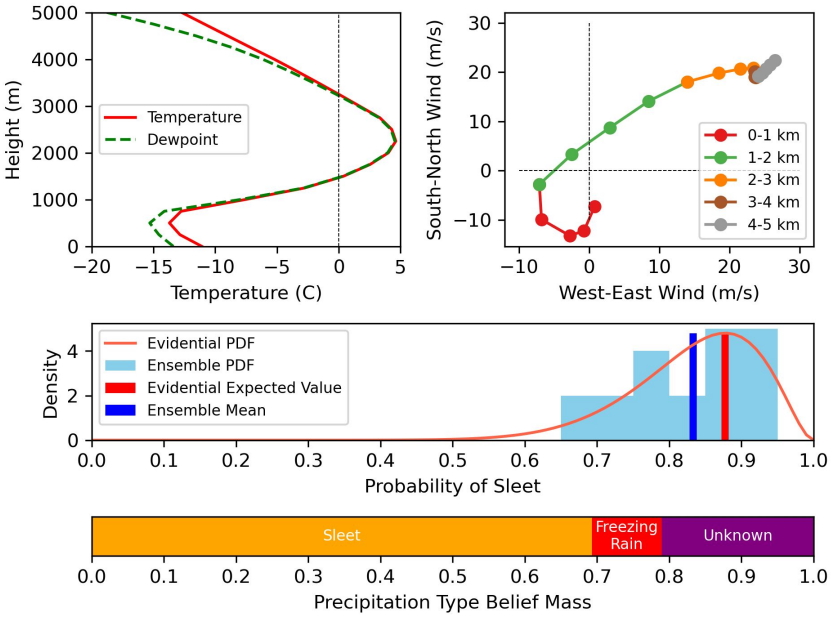
Summary



Reproducing pre-processing steps as critical and reproducing ML model.



Scaling of input data exhibits multiple sensitivities.



Evidential deep learning enables evaluation of situations when high variance among models is more likely.

Email: dgagne@ucar.edu