Creating and Evaluating Uncertainty Estimates with Neural Networks for Environmental Science Applications

Katherine Haynes, Ryan Lagerquist,
Marie McGraw, Kate Musgrave, Imme Ebert-Uphoff

April 11, 2023

Workshop for AI in Ocean, Atmosphere, and Climate Dynamics

UQ Team at CIRA

Katherine Haynes

Ryan Lagerquist

Marie McGraw

Kate Musgrave

Imme Ebert-Uphoff

- Paper:
  K. Haynes, R. Lagerquist, M. McGraw, K. Musgrave, & I. Ebert-Uphoff (2023). Creating and evaluating uncertainty estimates with neural networks for environmental-science applications. *Artif. Intell. Earth Syst.*, 2, 220061, https://doi.org/10.1175/AIES-D-22-0061.1.
- Code:
  https://github.com/thunderhoser/cira_uq4ml

Outline

1. Motivation
2. Uncertainty Components: Aleatory vs. Epistemic
3. Simple methods to *estimate* uncertainty
4. Simple methods to *evaluate* uncertainty
5. Ending Remarks

Intended Audience:

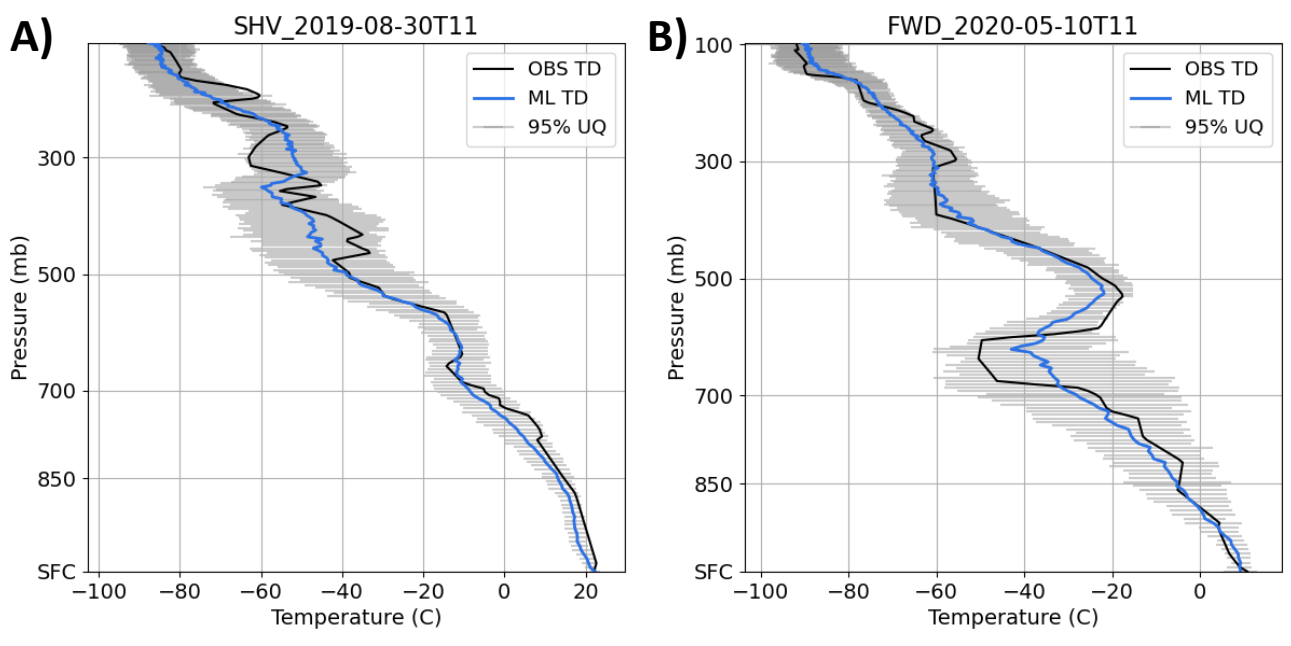
- Novices – folks who use NNs in their applications and would like to get uncertainty estimates, but don't know where to get started
- Intermediate – folks who have tried some uncertainty modeling, but would like to learn more about other methods and evaluation.

Motivation

- Neural networks (NN) are widely used in weather and climate applications
- Classic NNs have no awareness of their limitations
  - Deliver a result
  - *May* deliver some sort of "confidence score", but usually not reliable
    (*e.g.*, classification NN may have a softmax layer that provides a "pseudo-probability" for each class; however, it's understood to just be an indication and not a true probability for that class)
- In the scientific world, we want a reliable uncertainty estimate
  - Probability distribution (*e.g.*, parameters of a normal distribution)
  - Non-parametric summary statistics
    (*e.g.*, confidence interval, histogram or quantiles)
  - Ensemble (*i.e.*, set of representative samples of the true distribution)

Real-World Application

Use ML to improve vertical profiles of moisture.



Black:	Observation of dewpoint temperature (by radiosonde)
Blue:	Vertical profiles predicted by machine learning (ML)
Gray lines:	Uncertainty predicted by ML (95% confidence interval)

2.

Uncertainty Components: Aleatory vs. Epistemic

Machine-learning definitions of uncertainty components

Uncertainty Components

The ML community distinguishes two uncertainty components:

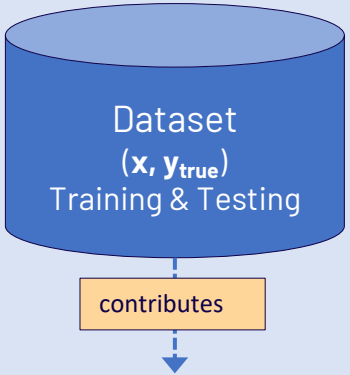
1. **Aleatory** uncertainty
2. **Epistemic** uncertainty

A word of caution:
Papers make the concepts of aleatory and epistemic uncertainty sound as if they are unambiguous and straight forward. But they are neither!

- Aleatory and epistemic uncertainty are very slippery concepts – highly dependent on **community** and **context**.
- In fact, Bevan (2022) illustrates **four different definitions of the terms aleatory and epistemic uncertainty** that are used in different communities.
- Be careful to know which definition you're using. Many papers do not make that clear.

Uncertainty Components

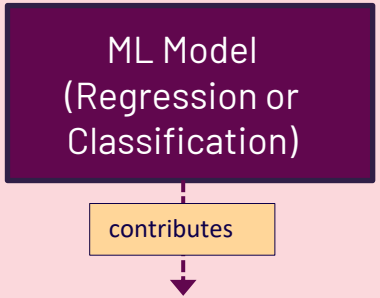
ML-Aleatory



Sources of uncertainty inherent in data, including:

- **Internal variability** of physical process (*e.g.,* chaotic nature of processes)
- **Observation error** (if data is observed) or **Physical Model error** (if data is from a physical model)

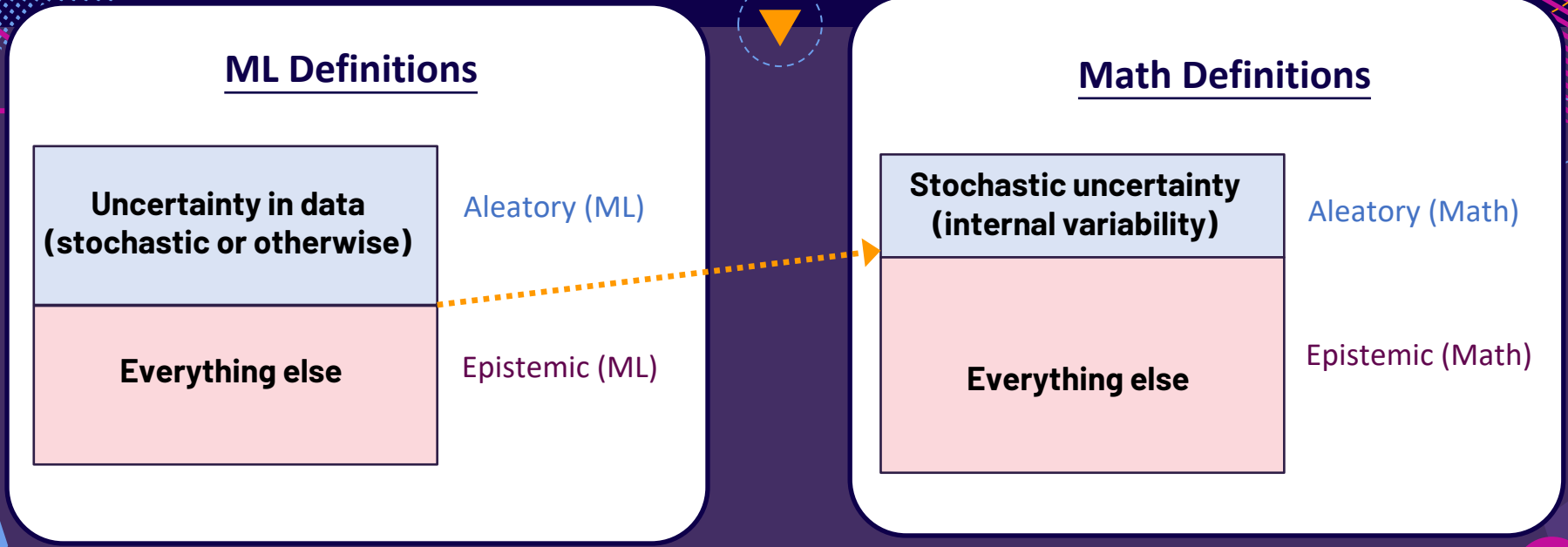
ML-Epistemic



Sources of uncertainty from ML model, including:

- **Structural error** (incorrect ML model structure)
- **Parametric error** (incorrect ML model parameters)
- **Out-of-regime error** (*e.g.,* creating predictions outside of training data regime; limitations to the dataset)

The Aleatory-Epistemic Divide



ML: Distinction is based on whether uncertainty is ***inherent in the data***.

Math: Distinction is based on whether source of uncertainty is ***stochastic***.

The Aleatory-Epistemic Divide

Dividing lines are different in ML vs. math, but concepts are called the same!

Can get very confusing.

ML: Distinction is based on whether uncertainty is ***inherent in the data***.

Math: Distinction is based on whether source of uncertainty is ***stochastic***.

Why is this difference important to understand?

1. Because difference in definitions creates a lot of confusion and makes it hard to understand papers.
2. ML papers even use alternate names (aleatory=stochastic=irreducible), creating even more confusion.
3. Using the ML definitions, the **divide between aleatory and epistemic becomes context dependent**. Example: Adding more features to your data set can shifts some aleatory uncertainty to epistemic!

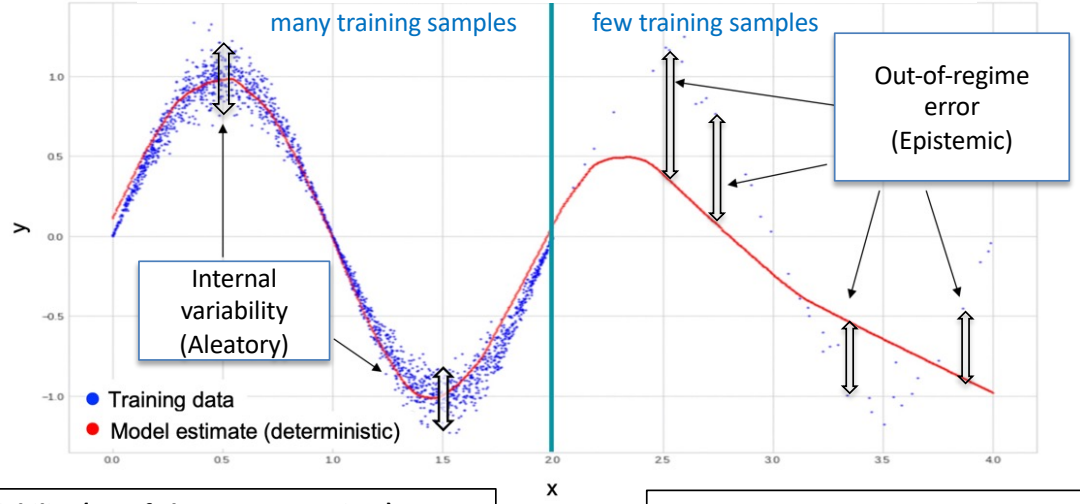
Recommended reading:

Hüllermeier, E. and Waegeman, W., 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), pp.457-506, https://doi.org/10.1007/s10994-021-05946-3.

Bevan, L.D., 2022. The ambiguities of uncertainty: A review of uncertainty frameworks relevant to the assessment of environmental change. *Futures*. https://doi.org/10.1016/j.futures.2022.102919

Classic Aleatory-Epistemic Example

Blue dots = training samples
Red line: model prediction



Internal variability (ex. of aleatory uncertainty):
Given x, there is no unique value for y, because of internal variability of observed system.
→ Even the best model cannot get it "exactly right".

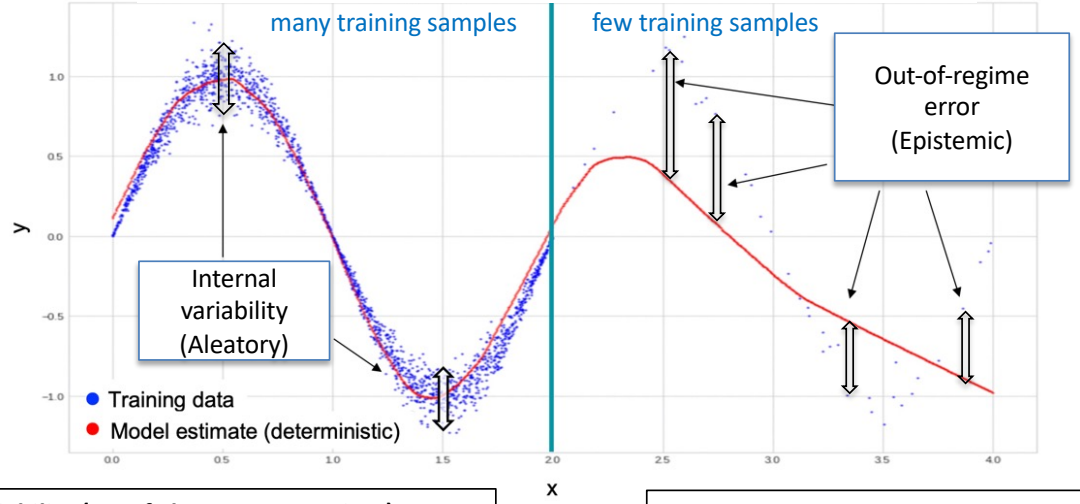
Out-of-regime error (ex. of epistemic uncertainty):
The model is trying to make predictions in an area where few training samples were provided
→ Large errors (out-of-regime error)

For this example, the ML and math definitions align, because it's idealized:

- Only type of data error shown here: internal variability.
- Only type of model error shown here: out-of-distribution error.

Classic Aleatory-Epistemic Example

Blue dots = training samples
Red line: model prediction



Internal variability (ex. of aleatory uncertainty):
Given x, there is no unique value for y, because of internal variability of observed system.
→ Even the best model cannot get it "exactly right".

Out-of-regime error (ex. of epistemic uncertainty):
The model is trying to make predictions in an area where few training samples were provided
→ Large errors (out-of-regime error)

CAUTION: Definitions and concepts depend on ***community*** and ***context***!

For the ML community, adding more predictors - without increasing sample size – is likely to:
- *decrease* ML-aleatory uncertainty by increasing information about the system's state
- *increase* ML-epistemic uncertainty since more data may be required to represent more complex relationships

3.

Simple Approaches to Estimate Uncertainty

Methodologies to derive NN uncertainty estimates

Approaches to Estimate Uncertainty

Non-Bayesian

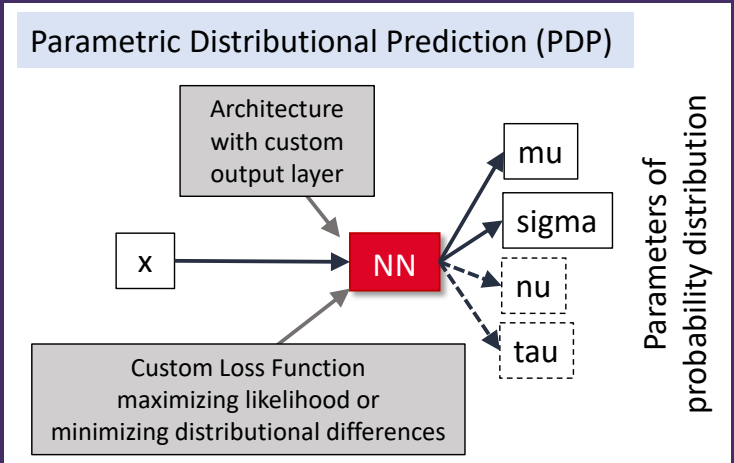
- Parametric distributional prediction (PDP)
- Non-parametric distributional prediction (NPDP)
- Ensemble prediction (EP)
- Multi-Model (MM)
- Evidence Neural Networks (ENN)

Bayesian

- Monte-Carlo Dropout
- Bayesian Neural Networks (BNN)

Code for all of these is provided in the notebooks.

Parametric Distributional Prediction (PDP)



Key Idea:

- Train NN to estimate the ***parameters*** of a probability distribution

How to add this to an existing NN model:

- Choose distribution (*e.g.*, normal, sinh-arcsinh)
- Change the loss function to negative log-likelihood
- Replace output layer with custom layer

Examples:

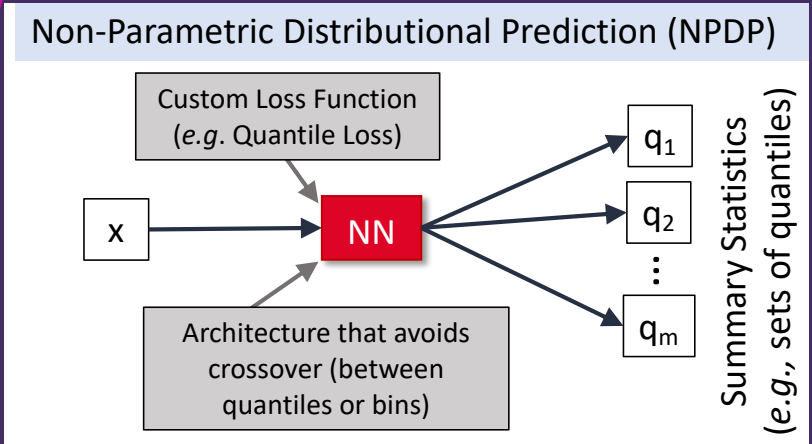
- Regression Notebooks

Can be used for	
Classification	No
Regression	Yes

Can be used to capture	
ML-Aleatory	Yes
ML-Epistemic	No

Pros: Minimal model changes; captures y_{true} spread
Cons: Distribution must be specified *a priori*; does not capture out-of-regime uncertainty

Non-Parametric Distributional Prediction (NPDP)



Can be used for	
Classification	Yes
Regression	Yes

Can be used to capture	
ML-Aleatory	Yes
ML-Epistemic	No

Key Idea:

- Train NN to predict ***summary statistics*** (*e.g.*, set of quantiles)

How to add this to an existing NN model:

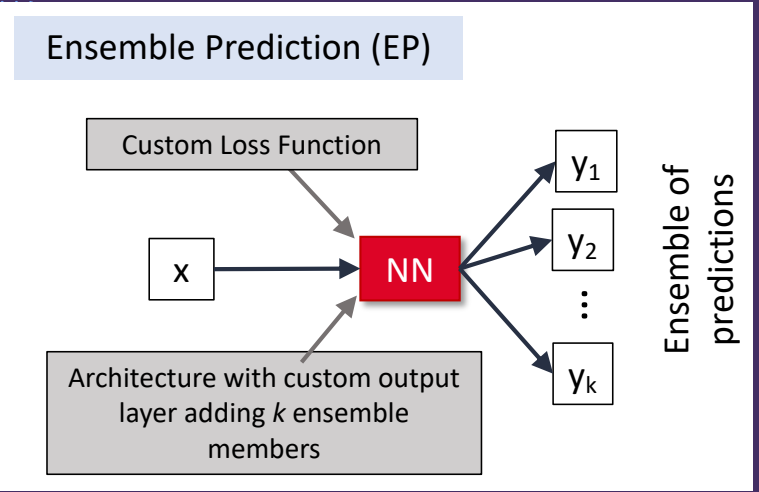
- Choose number of statistical bins
- Change to custom loss function
- Replace output layer with custom layer
- Modify architecture to prevent cross-over

Example:

- Classification NPDP Notebook

Pros: Straight-forward model changes; used for classification and regression; captures y_{true} spread
Cons: Summary statistics must be specified *a priori*; does not capture out-of-regime uncertainty

Ensemble Prediction (EP)



Can be used for	
Classification	Yes
Regression	Yes

Can be used to capture	
ML-Aleatory	Yes
ML-Epistemic	No

Key Idea:

- Train NN to estimate an ***ensemble*** that represents the data distribution

How to add this to an existing NN model:

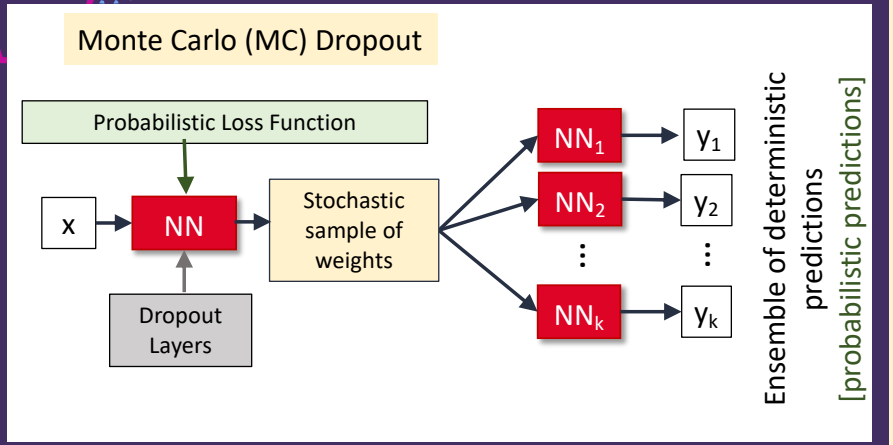
- Choose number of ensemble members
- Change to custom loss function (*e.g.*, Continuous Rank Probability Score [CRPS])
- Replace output layer with custom layer

Example:

- Regression Notebooks

Pros: Fully represents target distribution; useful for tasks with co-occurring target regimes
Cons: Number of ensembles must be specified *a priori*; does not capture out-of-regime uncertainty

Monte Carlo Dropout



Can be used for	
Classification	Yes
Regression	Yes

Can be used to capture	
ML-Aleatory	Depends
ML-Epistemic	Yes

Key Idea (largely true for Bayesian Deep Learning):

- Each NN weight is a probability distribution (not a single number)
- For every prediction call, the model randomly selects weights (*i.e.*, new ensemble drawn)
- Create k ensembles by running the model k times

How to add this to an existing NN model:

- Add dropout layers
- Use custom loss function [optional for ML-aleatory]

Examples:

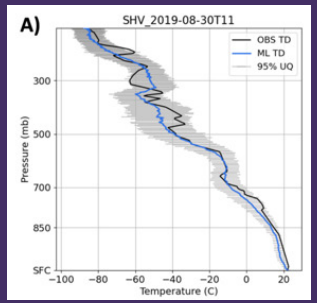
- Classification MC Dropout & Regression Notebooks

Pros: Requires no model changes; used for classification and regression; predicts out-of-regime uncertainty
Cons: Monte Carlo required at runtime, does not capture y_{true} spread with deterministic loss function

Now we have

- *4 methods* to choose from to calculate uncertainty estimates
  - Parametric Distributional Prediction (PDP)
  - Non-Parametric Distributional Prediction (NPDP)
  - Ensemble Prediction (EP)
  - Monte Carlo Dropout

Can use those to generate uncertainty estimates



But how do we know which estimates are good?

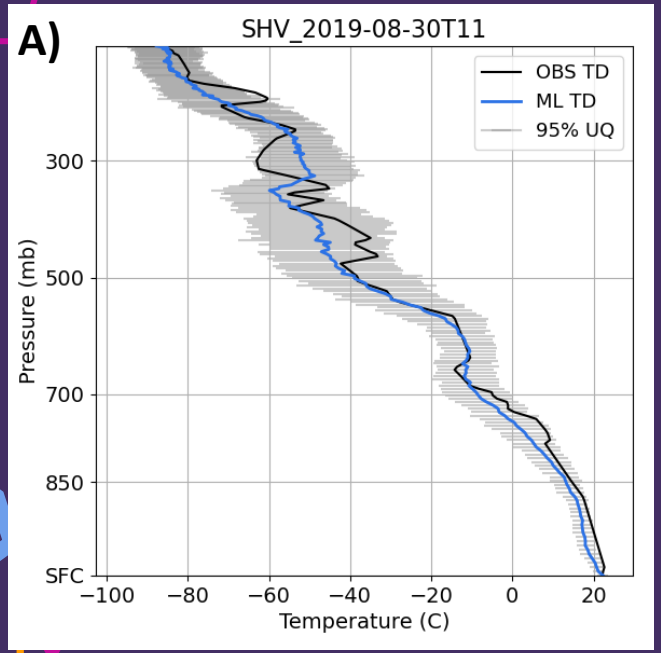
→ **Need methods to <u>evaluate</u> uncertainty estimates!**

4.

Simple Methods to Evaluate Total Uncertainty

Single-value scores and graphical methods to evaluate NN uncertainty estimates

Real-World Example:
ML Uncertainty for Dewpoint Vertical Profiles



Calculate uncertainty estimates using 4 methods:

- PDP using normal distribution (PDP_Norm; orange)
- PDP using sinh-arcsinh distribution (PDP_SHASH; green)
- EP using CRPS loss (EP_CRPS; purple)
- Monte Carlo Dropout (MSE loss) (MC_DROPS; pink)

Note:

- We did not create out-of-distributions samples in the test set
  - Here we are only evaluating ***aleatory*** uncertainty
- MC dropout with MSE loss does not capture aleatory uncertainty
  - Cannot shine in the tests shown here

Graphical Methods to Evaluate Total Uncertainty

Spread-Skill Plot

For a given predicted model spread, what is the actual model error?

Delle Monache et al. (2013)

Discard Test

Does model performance improve when the most uncertain cases are removed?

Barnes and Barnes (2021)

Probability Integral Transform (PIT) Histogram

Are the predicted uncertainty estimates well-calibrated? Is the model under- or over-confident?

Hamill (2001)

Don't forget to evaluate the central value:

Attributes Diagram

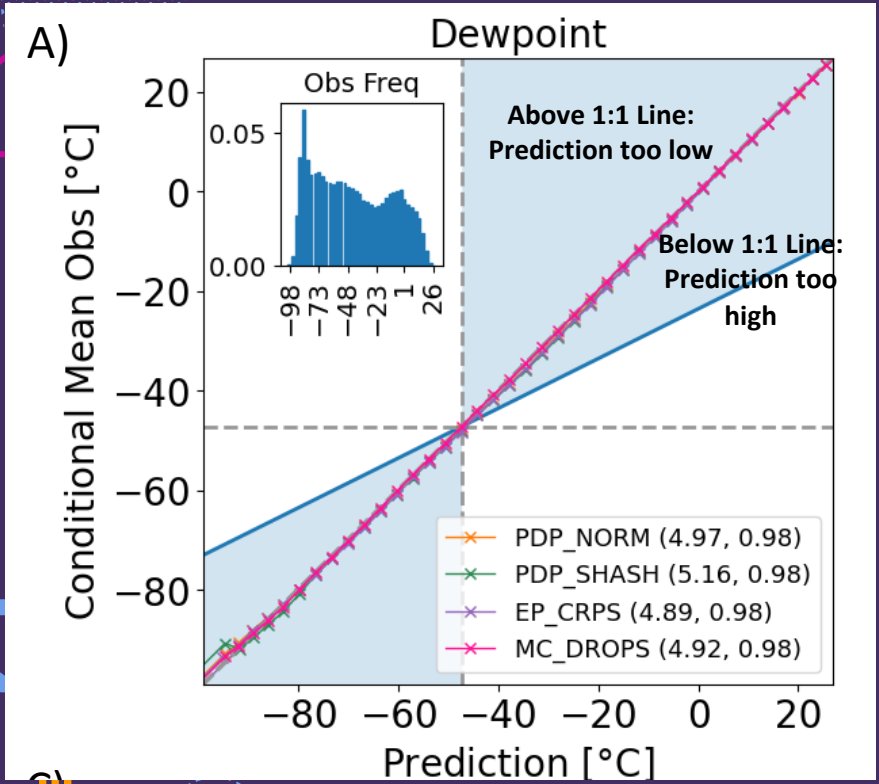
For a given central model prediction, what is the true value?

Hsu and Murphy (1986)

Code for all of these is provided in the notebooks.

Attributes Diagram

Can be used for	
Classification	Yes
Regression	Yes



Key idea:

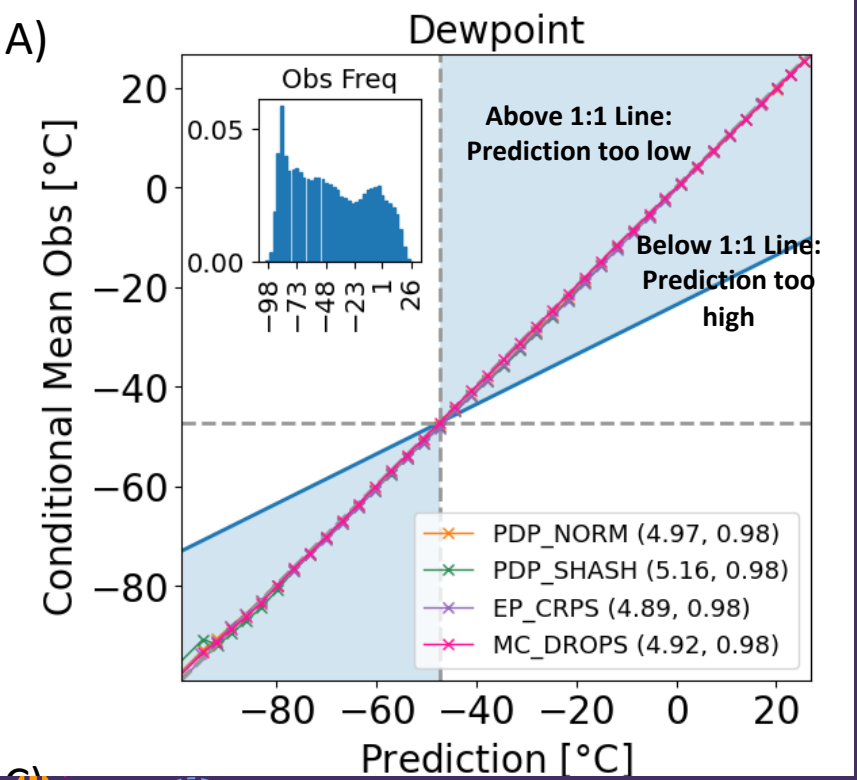
- Question: For a given prediction, what is the actual observation?
- Plot the relationship between predicted central value and the actual observed value.

How to read an attributes diagram:

- Diagonal is ideal:
  Predictions match observations.
- Above diagonal:
  Predictions are lower than observed.
- Below diagonal:
  Predictions are higher than observed.

Attributes Diagram

Can be used for	
Classification	Yes
Regression	Yes



Key idea:

- Question: For a given prediction, what is the actual observation?
- Plot the relationship between predicted central value and the actual observed value.

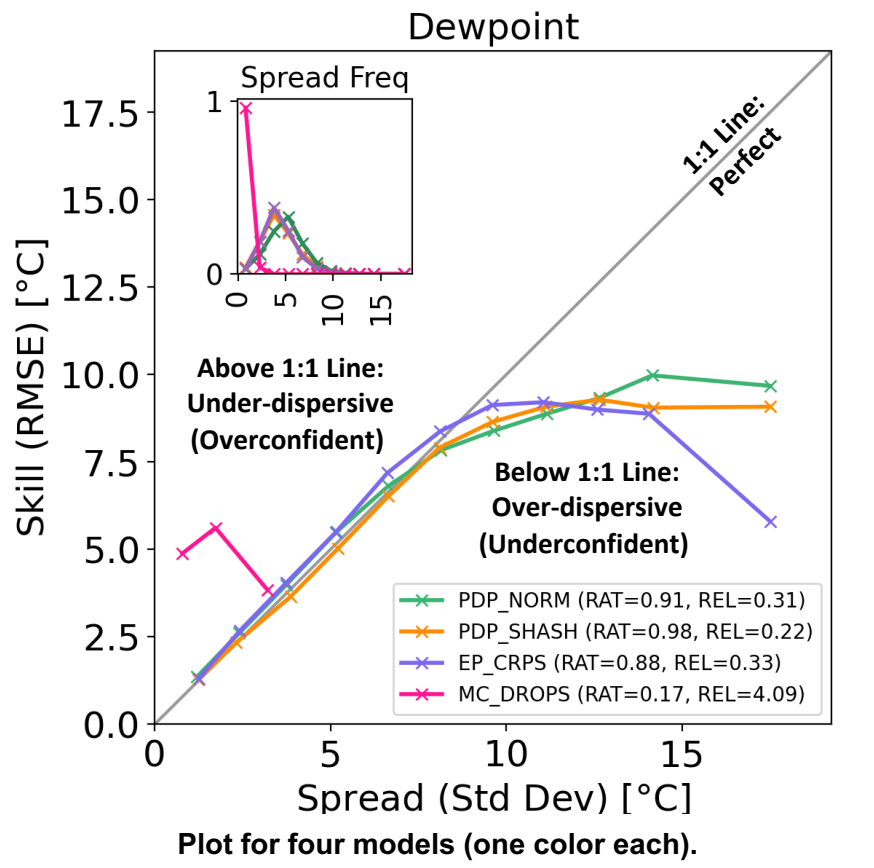
Related single-value metrics:

- **Brier-Skill Score** (BSS; classification)
- **Mean-Square Error Skill Score** (MSESS; regression)
- Both show improvement over climatology
- Ideal = 1; > 0 means improvement

- All models capture central values well
- PDP_SHASH predicts too cold lowest temps

Spread-Skill Plot

Can be used for	
Classification	Yes
Regression	Yes



Plot for four models (one color each).

Key idea:

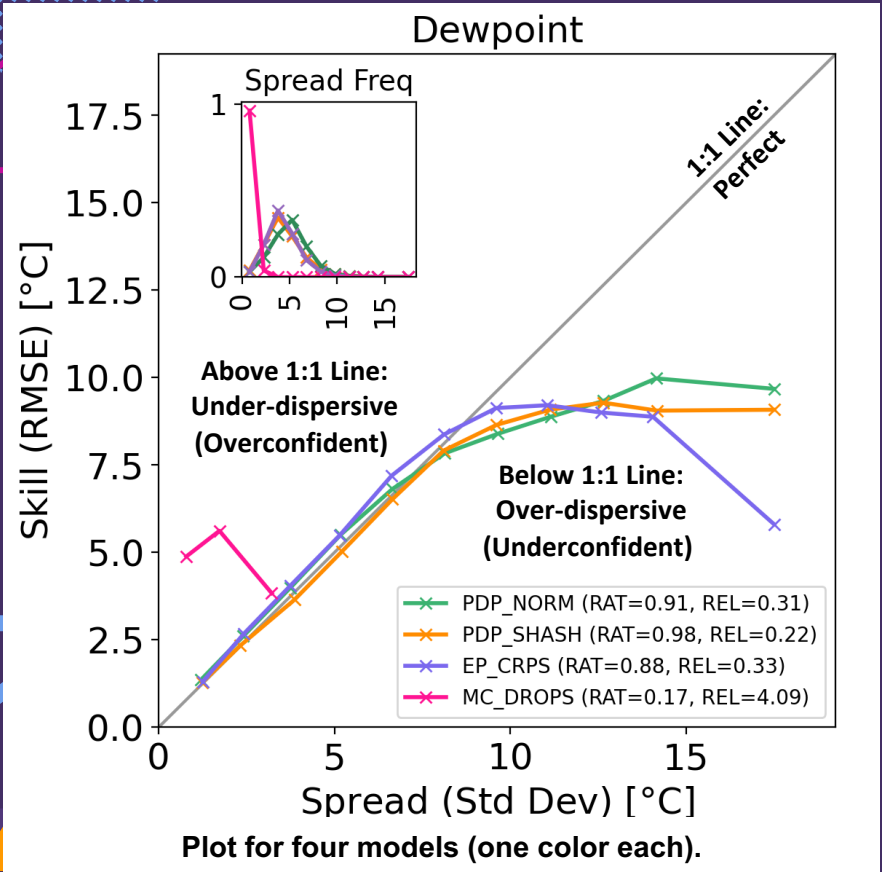
- Question: For a given predicted model spread, what is the actual model error?
- Plot the relationship between predicted uncertainty and actual model error (RMSE) of prediction.

How to read a spread-skill plot:

- Diagonal is ideal: predicted uncertainty matches actual error of prediction.
- Above diagonal: uncertainty estimate is too low (model overconfident).
- Below diagonal: uncertainty estimate is too high (model underconfident).

Spread-Skill Plot

Can be used for	
Classification	Yes
Regression	Yes



Plot for four models (one color each).

Key idea:

- Question: For a given predicted model spread, what is the actual model error?
- Plot the relationship between predicted uncertainty and actual model error.

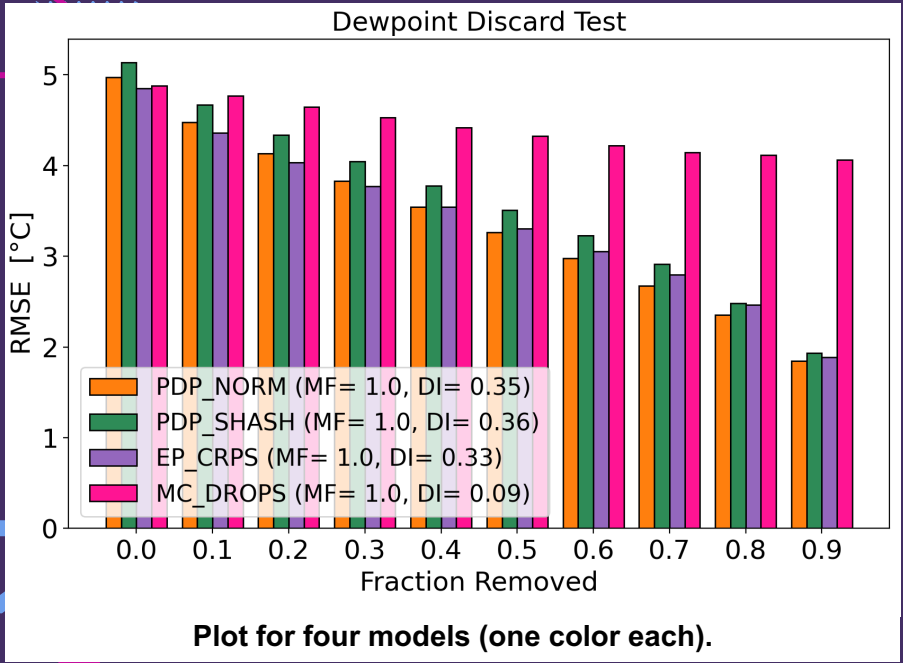
Related single-value metrics:

- **Spread-skill ratio** (SSRAT):
  - Spread/RMSE averaged over dataset.
  - Ideal value = 1.
- **Spread-skill reliability** (SSREL):
  - Weighted distance between spread-skill plot and 1:1 line.
  - Ideal value = 1.

- MC dropout is overconfident
- PDP and EP perform well for small uncertainties
  - Underconfident for large uncertainties

Discard Test

Can be used for	
Classification	Yes
Regression	Yes



Plot for four models (one color each).

Key idea:

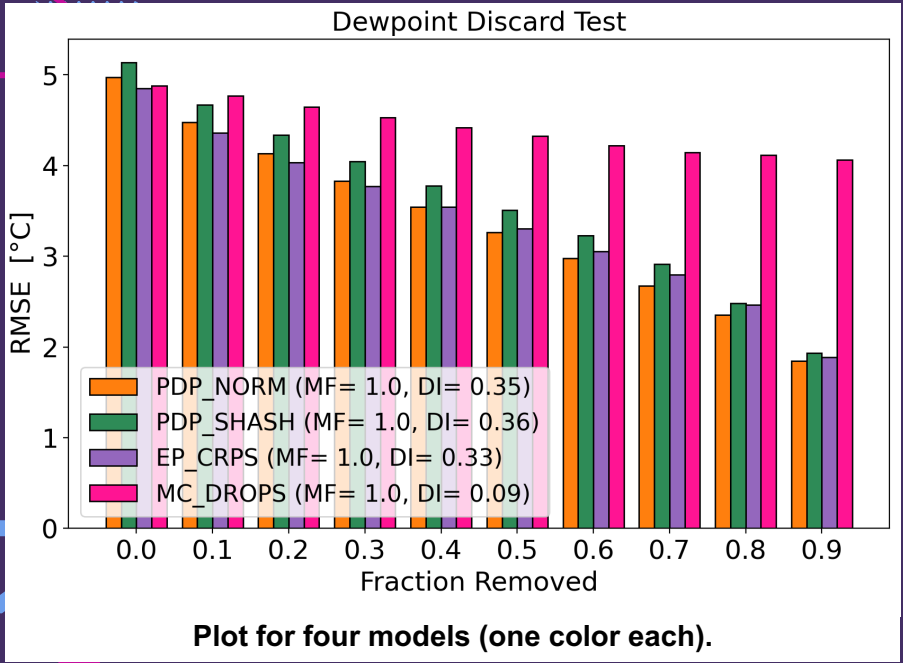
- Question: Does model performance improve when the most uncertain cases are removed?
- Discard samples with highest X% of uncertainty and calculate model error (RMSE) for only the remaining samples.

How to read a discard test plot:

- Bars for each model (single color) should decrease monotonically from left to right ➔ uncertainty is well calibrated.

Discard Test

Can be used for	
Classification	Yes
Regression	Yes



Plot for four models (one color each).

Key idea:

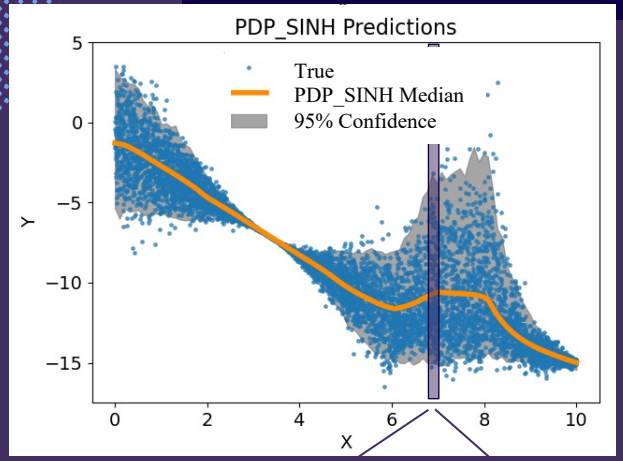
- Question: Does model performance improve when uncertain cases are removed?
- Discard samples with high uncertainty and calculate model error for remaining samples.

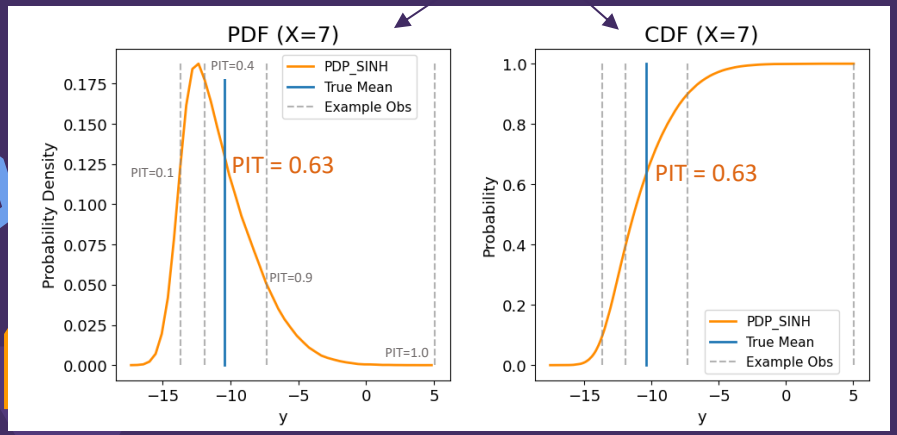
Related single-value metrics:

- **Monotonicity Fraction** (MF)
  - *How often* error decreases when discard fraction is increased
  - Ideal value = 1.
- **Discard Improvement** (DI)
  - *How much* error decreases on average when discard fraction is increased.
  - Higher values are better.

- All model errors decrease monotonically
- Model improvement from dropping uncertain samples is much greater for PDP and EP models than for MC dropout

Probability Integral Transform (PIT) Histogram





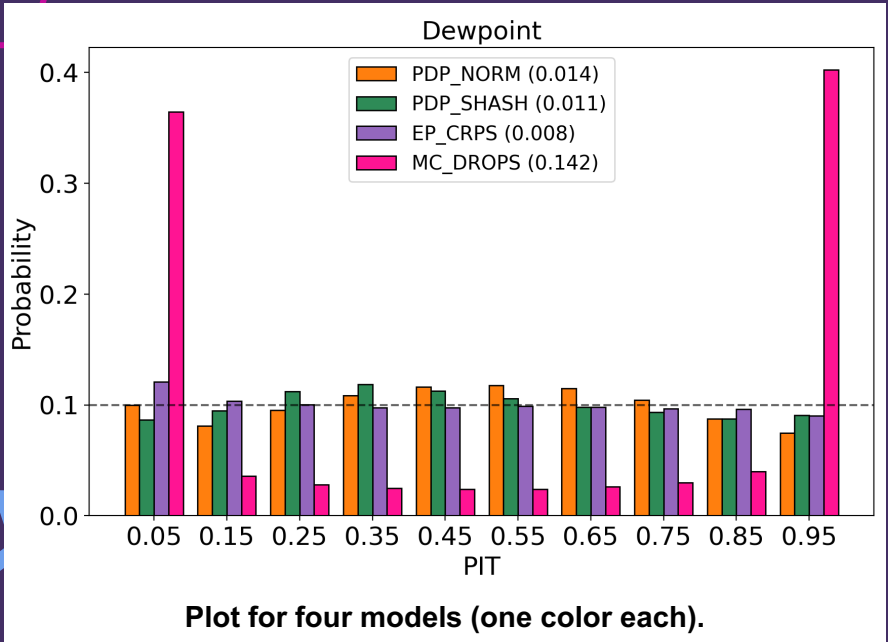
Key idea:

- Question: Are the predicted uncertainty estimates well-calibrated? Is the model under- or over-confident?

- PIT is the cumulative distribution function (CDF) of the predicted distribution evaluated at the observed value
- Can be interpreted as the quantile of the predicted distribution at which the observed value occurs
- Generalization of the "rank histogram" (*aka* "Talagrand diagram")

Probability Integral Transform (PIT) Histogram

Can be used for	
Classification	No
Regression	Yes



Plot for four models (one color each).

Key idea:

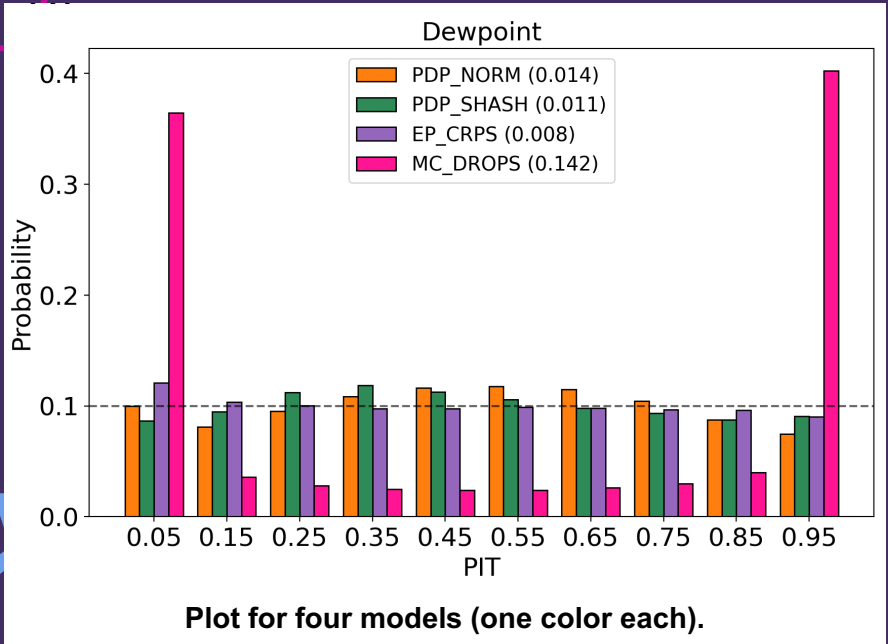
- Question: Are the predicted uncertainty estimates well-calibrated? Is the model under- or over-confident?
- Distribution of PIT values over data (one PIT value per individual prediction)

How to read a PIT histogram:

- Flat: uncertainty is perfectly calibrated
- Hump in the middle: model is underconfident (uncertainty is overspread)
- Higher at the ends: model is overconfident (uncertainty is underspread)

Probability Integral Transform (PIT) Histogram

Can be used for	
Classification	No
Regression	Yes



Plot for four models (one color each).

Key idea:

- Question: Are the predicted uncertainty estimates well-calibrated?
- Distribution of PIT values over data

Related single-value metrics:

- PITD
  - PIT calibration-deviation score
  - Mean difference between actual bin frequency and expected bin frequency for uniform histogram.
  - Ideal value = 0.

- MC dropout model is overconfident, underestimating uncertainty spread
- Other models doing ok, tend to be underconfident

Single-Score Metrics to Compare Uncertainty Estimates

Attributes Diagram	Brier-Skill Score (BSS) & Mean-Square Error Skill Score (MSESS)	Improvement over climatology. Ideal = 1; > 0 means improvement.
Spread-Skill Plot	Spread-Skill Ratio (SSRAT)	Spread/RMSE averaged over dataset. Ideal=1.
	Spread-Skill Reliability (SSREL)	Weighted distance from spread-skill plot to 1:1 line. Ideal=0.
Discard Test	Monotonicity Fraction (MF)	*How often* error decreases when discard fraction is increased. Ideal=1.
	Discard Improvement (DI)	*How much* error decreases on average when discard fraction is increased. Higher values are better.
PIT Histogram	Probability Integral Transform Calibration-Deviation Score (PITD)	Mean difference between actual PIT bin frequency and expected bin frequency for a uniform histogram. Ideal=0.

Single-Score Metrics to Compare Uncertainty Estimates

Continuous Rank Probability Score (CRPS)

How well the predicted distribution captures the true spread.

Ignorance Score (IGN)

How much the predicted distribution is concentrated in the correct areas.

Continuous Rank Probability Score (CRPS)

Comparison between **deterministic models** and **probabilistic models.**

Probabilistic adaptation of the mean square error.

Predicted value

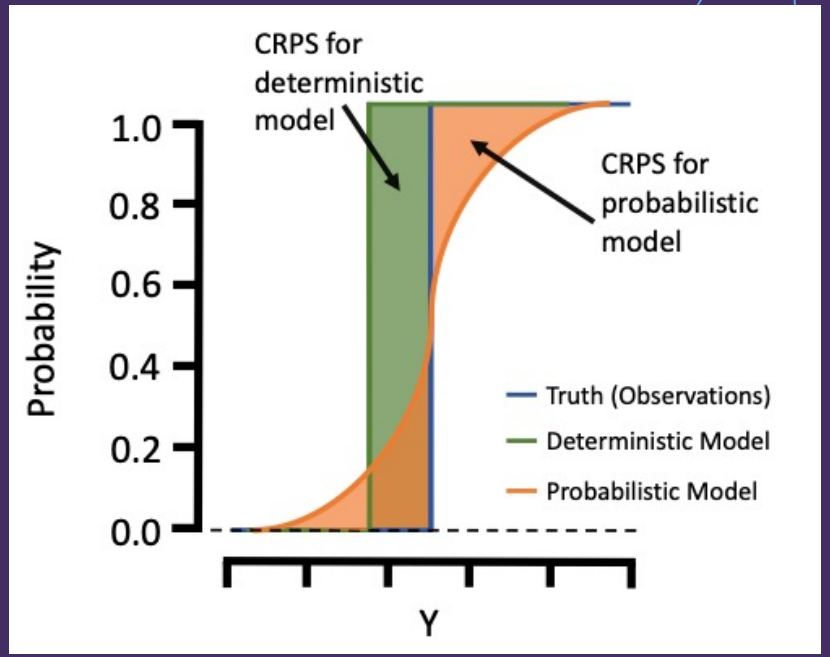
Observed value

$$CRPS(F, y_{true}) = \int_{-\infty}^{\infty} \left(F(y_{pred}) - \mathcal{H}(y_{pred} - y_{true})\right)^2 dy$$

Predicted CDF

Heaviside step function:
1 if $y > y_{true}$
0 otherwise.

CRPS varies from [0, ∞), and the **optimal value is 0** (no difference between model and obs)

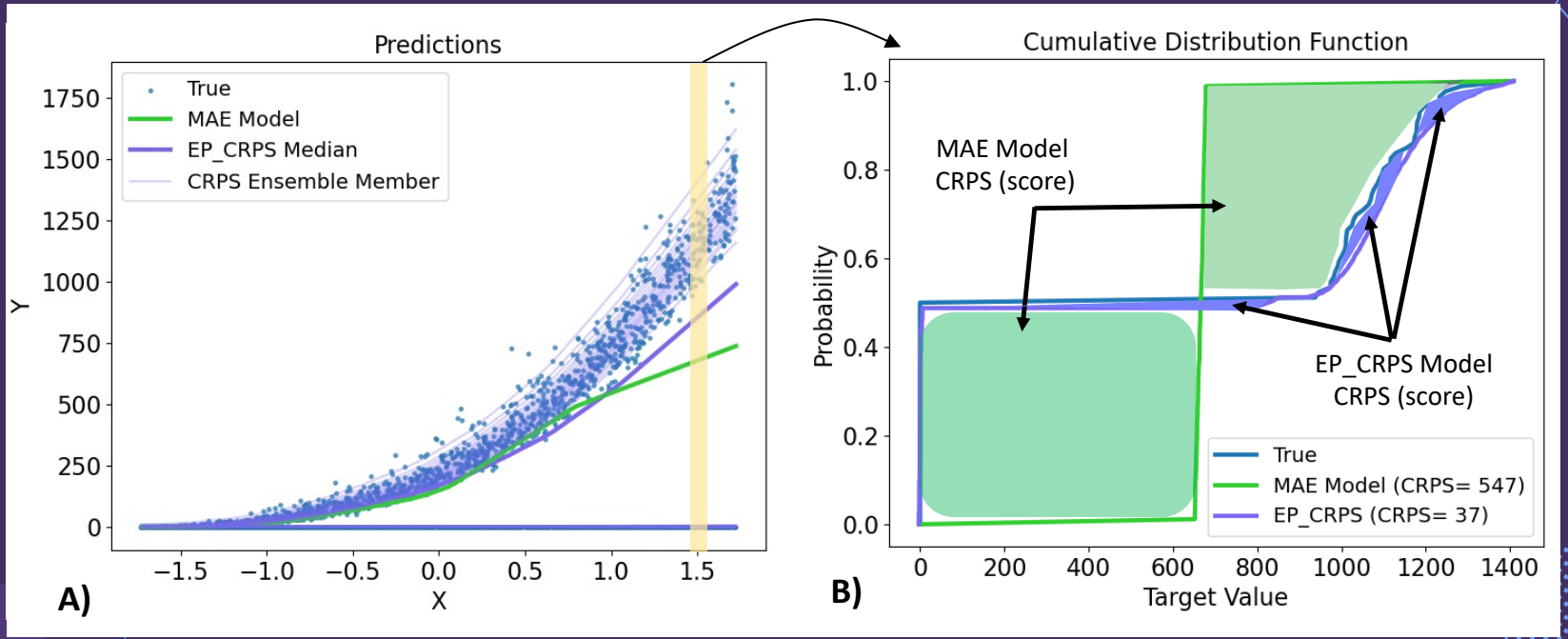


Adapted from Brey [2021].

Continuous Rank Probability Score (CRPS)

Comparison between **deterministic models** and **probabilistic models.**

Probabilistic adaptation of the mean square error.



Continuous Rank Probability Score (CRPS)

Big **advantage** of CRPS:

- Considers the full predictive CDF
  - Not just the mean or standard deviation or certain quantiles

CRPS can be used as a loss function to train neural networks

- Creates an ensemble of predictions
  - Can be used to quantify uncertainty
- Ensemble members are trained to represent the true CDF
  - Does not require any *a priori* distribution information

Original work using CRPS and providing excellent derivations

- Hersbach (2000): Decomposition of the CRPS for ensemble prediction systems
- Gneiting et al (2005): Calibrated probabilistic forecasting using minimum CRPS estimation
- Gneiting and Raftery (2007): Strictly proper scoring rules, prediction, and estimation
- Székely and Rizzo (2005): A new test for multivariate normality

Ignorance (IGN)

Measures how much a probabilistic forecast is concentrated in the correct areas.

Observed value

$$\mathrm{IGN} = -\frac{1}{N}\sum_{i=1}^{N}\log_2[f(y_i^{\mathrm{true}})],$$

PDF of predicted distribution

IGN varies from [0, ∞), and the **ideal value is 0**.
Rewards correct high-confidence predictions
(narrow predicted distributions that contain the observed value).

Real World Example: Evaluation

Score	PDP_NORM	PDP_SHASH	EP_CRPS	MC_DROPS
RMSE	4.97	5.16	**4.89**	4.92
MSESS	**0.98**	**0.98**	**0.98**	**0.98**
SSRAT	0.91	**0.98**	0.88	0.17
SSREL	0.31	**0.22**	0.33	4.09
MF	**1.00**	**1.00**	**1.00**	**1.00**
DI	0.35	**0.36**	0.33	0.09
PITD	0.014	0.011	**0.008**	0.142
CRPS	2.52	2.62	**2.38**	3.07
IGN	4.36	4.40	**4.15**	8.81

1) Mean predictions from four different models have nearly equal skill (similar RMSE and MSESS).

2) MC dropout produces the worst UQ estimates (expected with a deterministic loss function when focusing on aleatory UQ)

3) Uncertainty estimates from PDP and EP approaches have nearly equal skill (although EP_CRPS model achieves best score more often than PDP_SHASH)

5.

Ending Remarks

Take-home messages from real-world examples.

Summary

- Simple methods exist to ***derive*** uncertainty estimates (sample code provided)
  - Different methods can yield higher performances for different tasks

- Simple methods exist to ***evaluate*** uncertainty estimates (sample code provided)
  - Graphical methods used in combination provide quick insights into model UQ performance
  - Single-value scores allow for easy comparison between different models

- Eval methods here evaluate ***total*** uncertainty: do not distinguish components
  - Contribution of uncertainty components depends on the dataset choices!
    - Adding additional features (predictors) shifts the uncertainty between ML-aleatory and ML-epistemic
    - Out-of-distribution error is a key component of the *ML-epistemic* uncertainty, but it is only seen if the test data include samples that are very different from the training data!

Comments

- Challenge is shifting from implementing and evaluating uncertainty estimates to properly ***using*** and ***interpreting*** them

- Questions remain: How do we ensure there are sufficient out-of-distribution samples to provide representative epistemic uncertainty? How do you best convey uncertainty in a useful way, particularly for images?

References

Barnes, E., and R. Barnes, 2021: Controlled abstention neural networks for identifying skillful predictions for regression problems. Journal of Advances in Modeling Earth Systems, 13 (12), e2021MS002 575, https://doi.org/10.1029/2021MS002575.

Bevan, L.D., 2022: The ambiguities of uncertainty: A review of uncertainty frameworks relevant to the assessment of environmental change. *Futures*. https://doi.org/10.1016/j.futures.2022.102919

Brey, S., 2021: Ensemble. GitHub, https://github.com/TheClimateCorporation/ensemble.

Delle Monache, L., F. Eckel, D. Rife, B. Nagarajan, and K. Searight, 2013: Probabilistic weather prediction with an analog ensemble. Monthly Weather Review, 141 (10), 3498–3516, https://doi.org/10.1175/MWR-D-12-00281.1.

Gneiting, T., and A. Raftery, 2007: Strictly proper scoring rules, prediction, and estimation. *J. Amer. Stat. Assoc.,* 102, 359-378, https://doi.org/10.1198/016214506000001437.

Gneiting, T., A.E. Raftery, A.H. Westveld, and T. Goldman, 2005: Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Mon. Wea. Rev.,* 133, 1098-1118, https://doi.org/10.1175/MWR2904.1.

Hamill, T., 2001: Interpretation of rank histograms for verifying ensemble forecasts. Monthly Weather Review, 129 (3), 550–560, https://doi.org/10.1175/1520-0493(2001)129<0550:IORHFV>2.0.CO;2.

Haynes, K., R. Lagerquist, M. McGraw, K. Musgrave, & I. Ebert-Uphoff, 2023: Creating and evaluating uncertainty estimates with neural networks for environmental-science applications. *Artif. Intell. Earth Syst.*, 2, 220061, https://doi.org/10.1175/AIES-D-22-0061.1.
Code: https://github.com/thunderhoser/cira_uq4ml

Hersbach, H., 2000: Decomposition of the continuous ranked probability score for ensemble prediction systems. *Wea. Forecasting,* 15, 559-570, https://doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2.

Hsu, W., and A. Murphy, 1986: The attributes diagram: A geometrical framework for assessing the quality of probability forecasts. International Journal of Forecasting, 2 (3), 285–293, https://doi.org/10.1016/0169-2070(86)90048-8.

Hüllermeier, E. and Waegeman, W., 2021: Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), pp.457-506. https://doi.org/10.1007/s10994-021-05946-3

Székely, G.J. and M.L. Rizzo, 2005: A new test for multivariate normality. *J. Multivar. Anal.,* 93, 58-80, https://doi.org/10.1016/j.jmva.2003.12.002.

Thanks!

Any questions?

You can contact me at
Katherine.Haynes@colostate.edu

Paper:

K. Haynes, R. Lagerquist, M. McGraw, K. Musgrave, & I. Ebert-Uphoff (2023). Creating and evaluating uncertainty estimates with neural networks for environmental-science applications. *Artif. Intell. Earth Syst.*, 2, 220061, https://doi.org/10.1175/AIES-D-22-0061.1.

Code:

https://github.com/thunderhoser/cira_uq4ml