UQ in Deep Learning via Bootstrap

Literature Review and Simulation Plan

Ananya Roy[1] Connor McNeill[1] Carter Hall[1]

April 9, 2025

[1]Department of Statistics, North Carolina State University

Outline

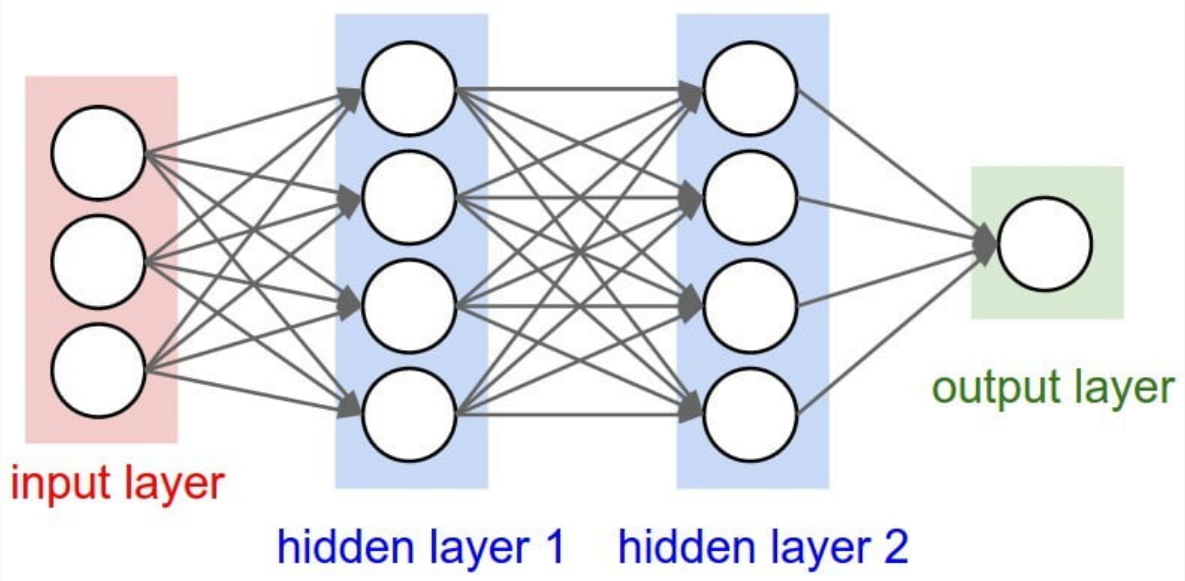
Background

What is Deep Learning?

- A subset of machine learning involving neural networks with multiple layers.
- Excels at capturing complex patterns in high-dimensional data.
- Widely used in computer vision, natural language processing (NLP), and scientific modeling.
- Black-box nature makes reliability and interpretability challenging.

Examples of Deep Learning

- **Deep neural networks** (DNN) are the most commonly used as the framework in deep learning.
- They consist of multiple layers of interconnected nodes.
- Input layer and output layer - the visible layers.
- DNNs contain at least 2 or more hidden layers.

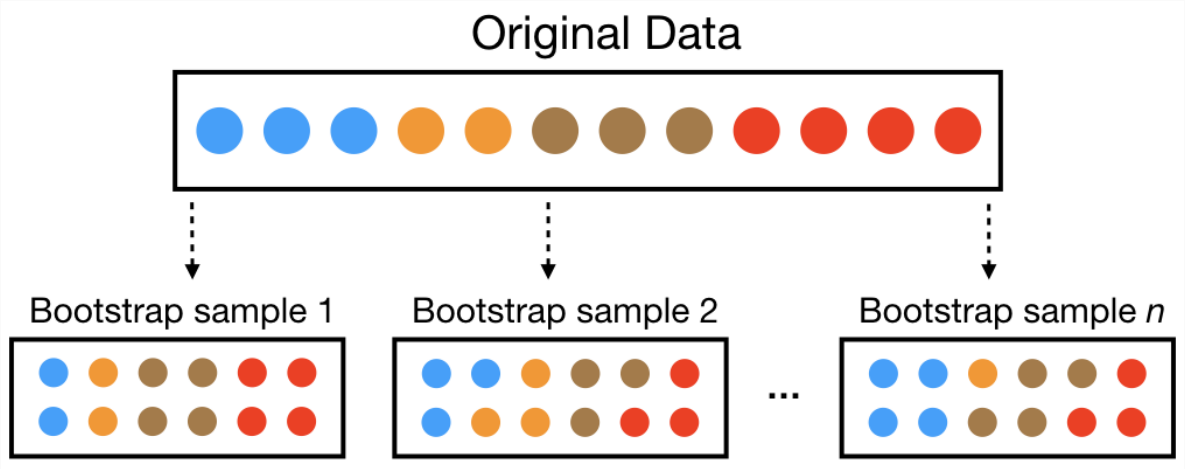


What is Uncertainty Quantification (UQ)?

- Aims to measure the confidence or reliability of model predictions.
- Essential for decision-making in high-stakes applications (e.g., healthcare, AVs).
- Types of uncertainty:
  - **Aleatoric**: Data uncertainty (e.g., noise), irreducible.
  - **Epistemic**: Model uncertainty, reducible.
- UQ enhances model robustness and trustworthiness.

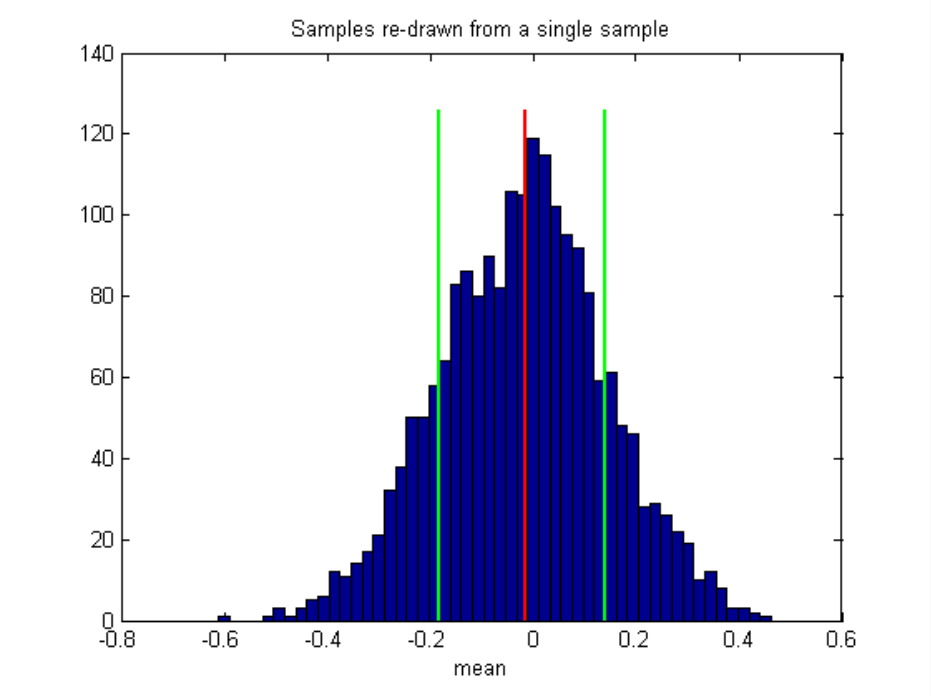
What is Bootstrap?

Bootstrapping is a resampling technique that is often utilized to create multiple simulated datasets. As the figure shows below, the sampling is done with replacement.



What is Bootstrap?

Bootstrapping is commonly used to provide variance estimation and also for uncertainty quantification. This all is derived from the bootstrap sampling distribution for a certain statistic, as shown below.



Types of Bootstrapping

- **Traditional (nonparametric):** Sample is same size as the original dataset, resampling done with replacement. No assumptions about the distribution of the data.
- **Bayesian:** Simulating the posterior of a parameter instead of the sampling distribution of a statistic.
- **Parametric:** Assumes that the data follows a specific distribution and uses this to generate bootstrap samples. Can be computationally efficient but requires assumption about the parameters.

Literature Review

Traditional Approaches for UQ in Deep Learning

1. **Bayesian approach:** goal is to utilize Bayesian Neural Networks in order to capture parameter uncertainty by estimating its posterior distribution.
2. **Ensemble methods approach:** goal is to utilize ensemble methods to capture model uncertainty in parameters and hyper-parameters.
3. **Sample distribution approach:** goal is to incorporate processes (such as Deep Gaussian processes) in order to capture uncertainty due to out-of-distribution samples.

Overview of UQ Methods

Method	Type	Key Idea
MC Dropout	Approx Bayesian	Dropout at inference. Stochastic forward passes approximate posterior.
MCMC	Exact Bayesian	Samples from posterior via Markov chains. Accurate but computationally heavy.
VI (Variational Inference)	Approx Bayesian	Minimizes KL divergence. Faster than MCMC.
BAL (Bayesian Active Learning)	Bayesian + AL	Selects unlabeled data w/ highest uncertainty (e.g. BALD). Speeds up labeling.
BBB (Bayes by Backprop)	Variational Bayesian	Learns weight posteriors via backprop-friendly reparameterization trick.
VAEs	Generative Bayesian	Trains encoder-decoder on ELBO. Captures latent uncertainty.
Laplace Approx.	Posterior Approx	Gaussian around MAP estimate. Quick but local solution.

Downsides of Traditional Approaches

The Bayesian approach has two main obstacles:

- Problem of setting the prior distribution
- Posterior inference is challenging for NNs of any practical size.
  - $\rightarrow$ Variational methods can have intrinsic bias if variational family is misspecified
  - $\rightarrow$ MCMC methods cannot be easily scaled

Ensemble approaches have a high computational cost.

Sample distribution approaches either are unscalable (deep GP) or inconsistent with input features (Distance-aware DNNs)

Why Bootstrap for UQ in Deep Learning?

- In deep learning, bootstrapped ensembles can mimic sampling from the posterior.
- Captures epistemic uncertainty by training on perturbed datasets.
- Scalable and simple to implement compared to full Bayesian approaches.

Preliminary Simulation Study

Step 1: Simulation & Modeling

- **Data Simulation:**
  - Generate $y = \sin(x) + \epsilon$ with noise $\epsilon \sim N(0, 0.5^2)$.
  - Uniformly sample x from $[0, 10]$ and split the data (80% training, 20% testing).
  - Enhance training data by oversampling the edge region ($x \in [8, 10]$) to improve performance where data was sparse.
- **Neural Network Modeling:**
  - Design a feedforward neural network with multiple layers (e.g., two layers with 128 neurons each followed by a hidden layer with 64 neurons).
  - Incorporate dropout layers (using a slightly lower rate) to prevent overfitting and enable MC Dropout.
  - Train the network for an extended number of epochs to ensure convergence and capture the non-linear sine behavior effectively.

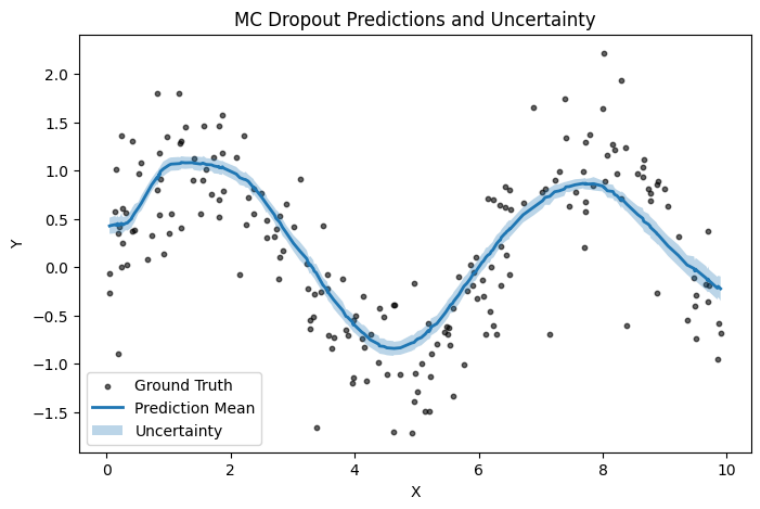
Step 2: Uncertainty Quantification Procedures

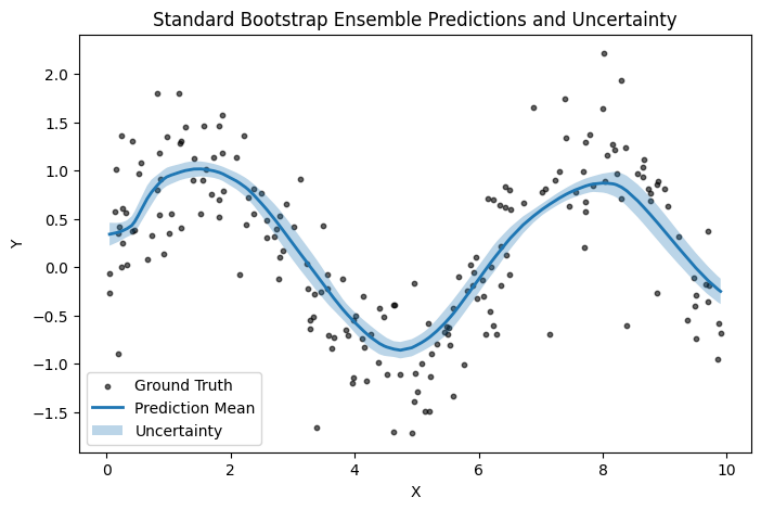
- **MC Dropout:**
  - Keep dropout active during inference.
  - Perform multiple forward passes to get a distribution of predictions; mean is final prediction, standard deviation is uncertainty measure.
- **Standard Bootstrap Ensemble:**
  - Resample training data with replacement to create bootstrapped datasets.
  - Train an ensemble on these varied samples and aggregate predictions to capture variability.

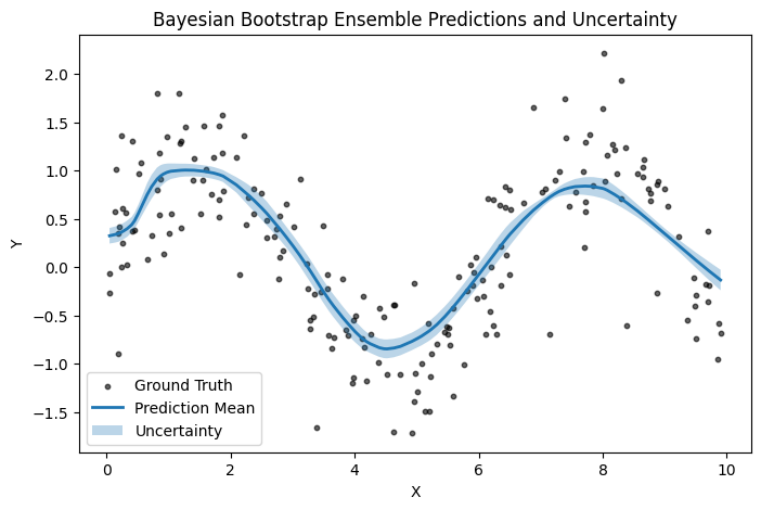
Step 2: Uncertainty Quantification Procedures

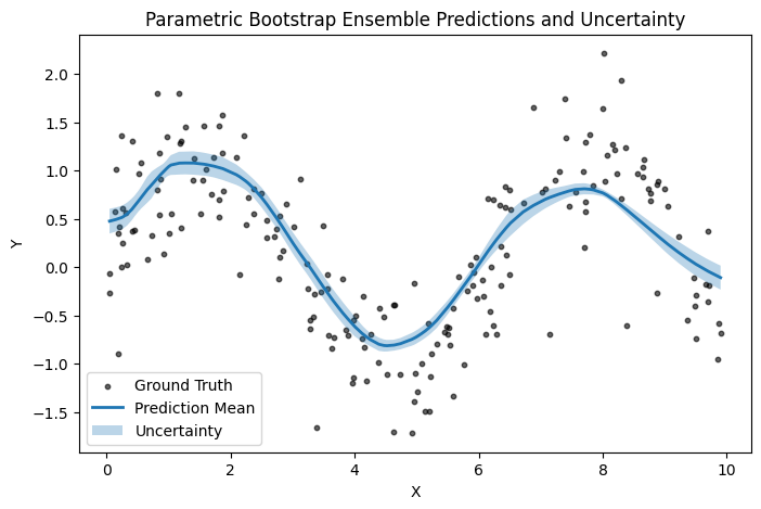
- **Bayesian Bootstrap Ensemble:**
  - Assign weights to each training sample via a Dirichlet distribution.
  - Train ensemble members with these weighted samples to simulate posterior variability.
- **Parametric Bootstrap Ensemble:**
  - Estimate the residual error (noise) from the base model.
  - Generate synthetic targets by adding Gaussian noise (using the estimated std. dev.) to the base predictions.
  - Train separate models on these synthetic datasets; aggregate outputs for overall uncertainty.

Results


MC Dropout Predictions and Uncertainty
Ground Truth
Prediction Mean
Uncertainty
X
Y


Standard Bootstrap Ensemble Predictions and Uncertainty
Ground Truth
Prediction Mean
Uncertainty
X
Y


Bayesian Bootstrap Ensemble Predictions and Uncertainty
Ground Truth
Prediction Mean
Uncertainty
X
Y


Parametric Bootstrap Ensemble Predictions and Uncertainty
Ground Truth
Prediction Mean
Uncertainty
X
Y

Comparison of UQ Methods: Accuracy and Efficiency

Evaluation Metrics Comparison:

Method	MSE	MAE	Avg Uncertainty
MC Dropout	0.247	0.398	0.083
Standard Bootstrap Ensemble	0.232	0.381	0.118
Bayesian Bootstrap Ensemble	0.241	0.391	0.094
Parametric Bootstrap Ensemble	0.257	0.408	0.094

Computational Complexity Comparison:

Method	Avg Train Time (s)	Total Inference (s)	Avg Inference (s)
MC Dropout	–	1.968	0.020
Standard BE	16.199	2.033	0.203
Bayesian BE	15.916	1.877	0.188
Parametric BE	16.588	2.108	0.211

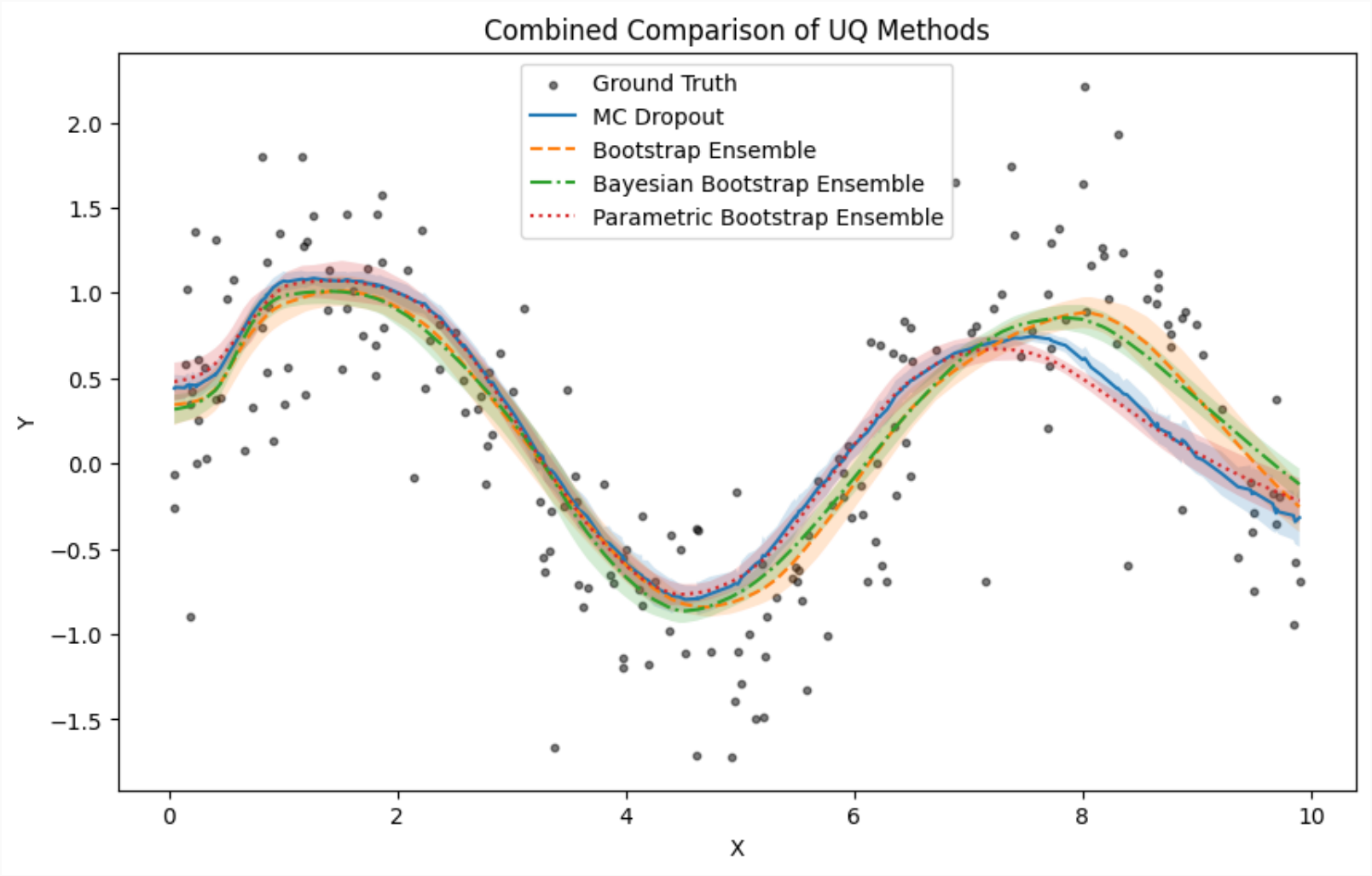


Figure 1: Comparison of UQ Methods

Calibration

Palmer et al., 2022 demonstrated that the standard deviation from a [direct] bootstrap ensemble method is not an *accurate* uncertainty estimate.

However, they offer an easy solution – a linear scaling of estimates of the true standard error of predicted values.

Calibration

How to Perform:

1. Use a bootstrapping method to train an ensemble of models. For each prediction (on test set), compute the ensemble mean and standard deviation (denoted $\widehat{\sigma}_{\text{uc}}$) – these are *initial*, *uncalibrated* estimates.
2. On a **validation** set, record the residual as truth $-$ mean prediction across ensemble. Similarly, record an uncalibrated standard deviation.
3. Assume the calibrated standard deviation, denoted $\widehat{\sigma}_{\text{cal}} = a\widehat{\sigma}_{\text{uc}} + b$. Optimize a, b by minimizing the negative log-likelihood such that the residuals$/\widehat{\sigma}_{\text{cal}} \sim N(0, 1)$.
4. Scale the uncalibrated estimates $\widehat{\sigma}_{uc}$ from the **test** dataset by (3).

Plan for Remaining Work

Plan for Remaining Work

- Finalize plans for simulation. (Extending results above to more complex scenarios, incorporate calibration.)
- Formalize writeup for preliminary report based off of work thus far.
- Code and perform simulation study.
- Write final report incorporating feedback received from presentation and preliminary report.

References i

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., & Nahavandi, S. (2021).**A review of uncertainty quantification in deep learning: Techniques, applications and challenges.** *Information Fusion*, *76*, 243–297. https://doi.org/10.1016/j.inffus.2021.05.008

Palmer, G., Du, S., Politowicz, A., Emory, J. P., Yang, X., Gautam, A., Gupta, G., Li, Z., Jacobs, R., & Morgan, D. (2022).**Calibration after bootstrap for accurate uncertainty quantification in regression models.** *npj Computational Materials*, *8*(1), 1–9. https://doi.org/10.1038/s41524-022-00794-8

References ii

- He, W., Jiang, Z., Xiao, T., Xu, Z., & Li, Y. (2025).**A Survey on Uncertainty Quantification Methods for Deep Learning.** (arXiv:2302.13425). https://doi.org/10.48550/arXiv.2302.13425
- Raj, A., Gudumotou, C. E., Bun, S., Srinivasa, K., & Sarshar, A. (2025).**Deep operator networks for bayesian parameter estimation in pdes.** https://arxiv.org/abs/2501.10684

Q&A

- Questions?

Q&A

- Questions?
- Thanks for listening!