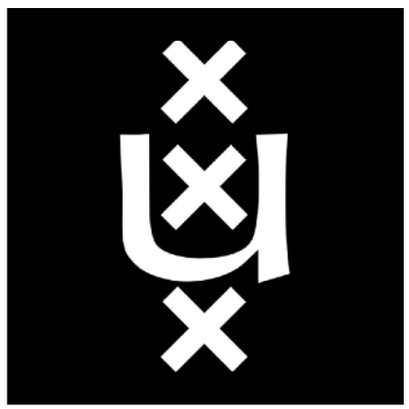
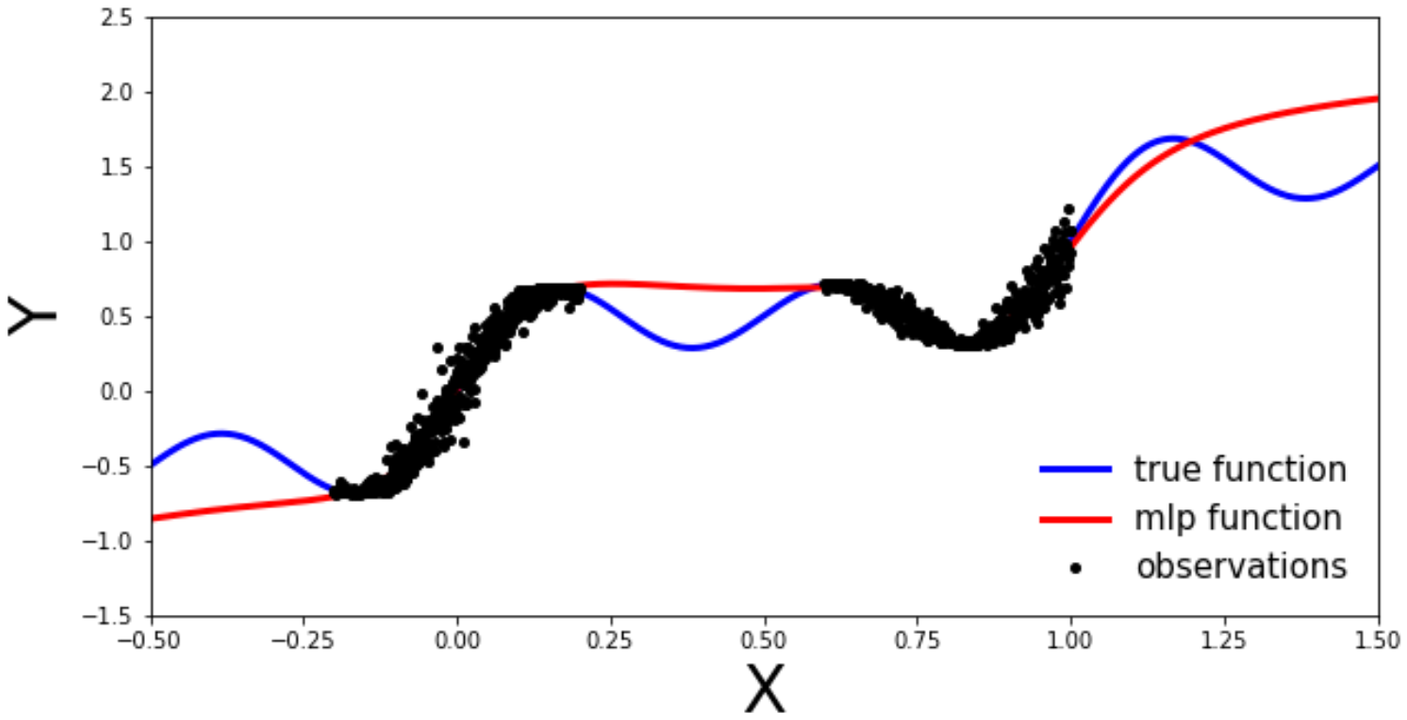
Uncertainty Quantification: Evaluation

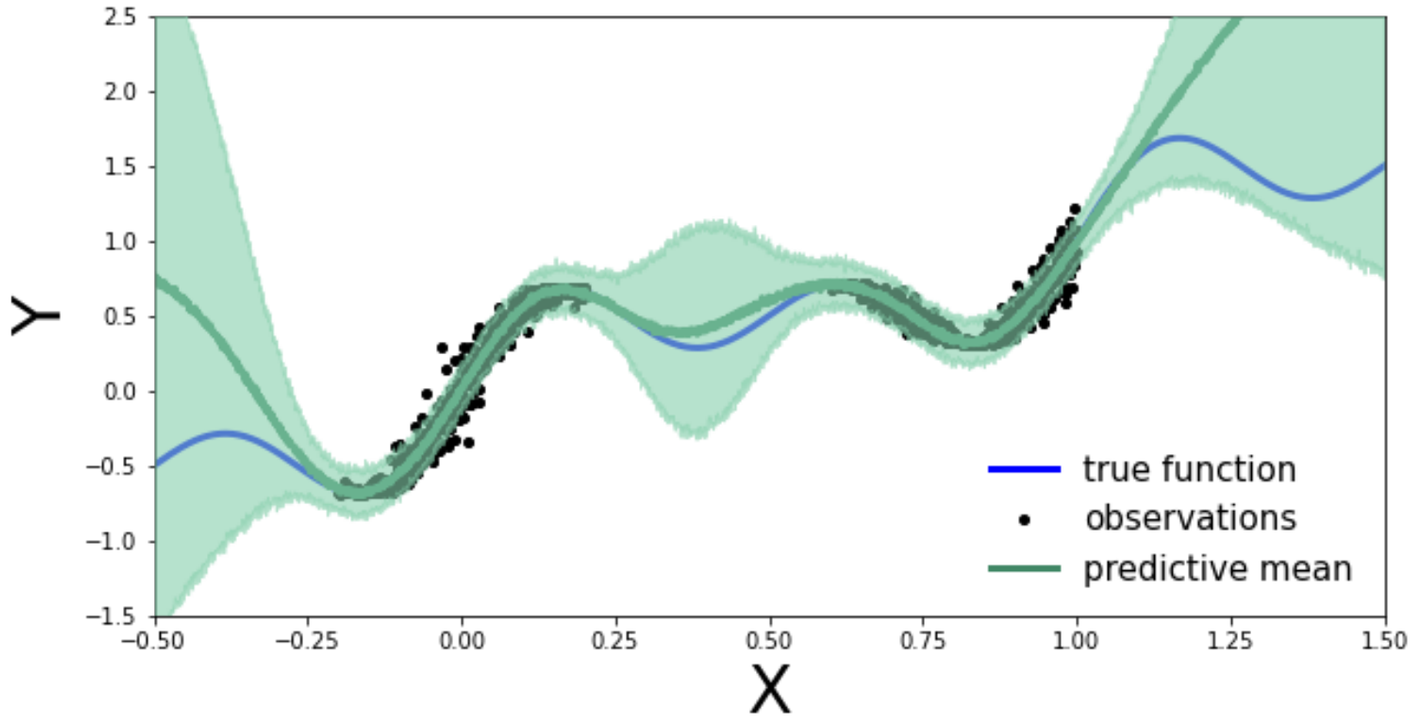
Eric Nalisnick



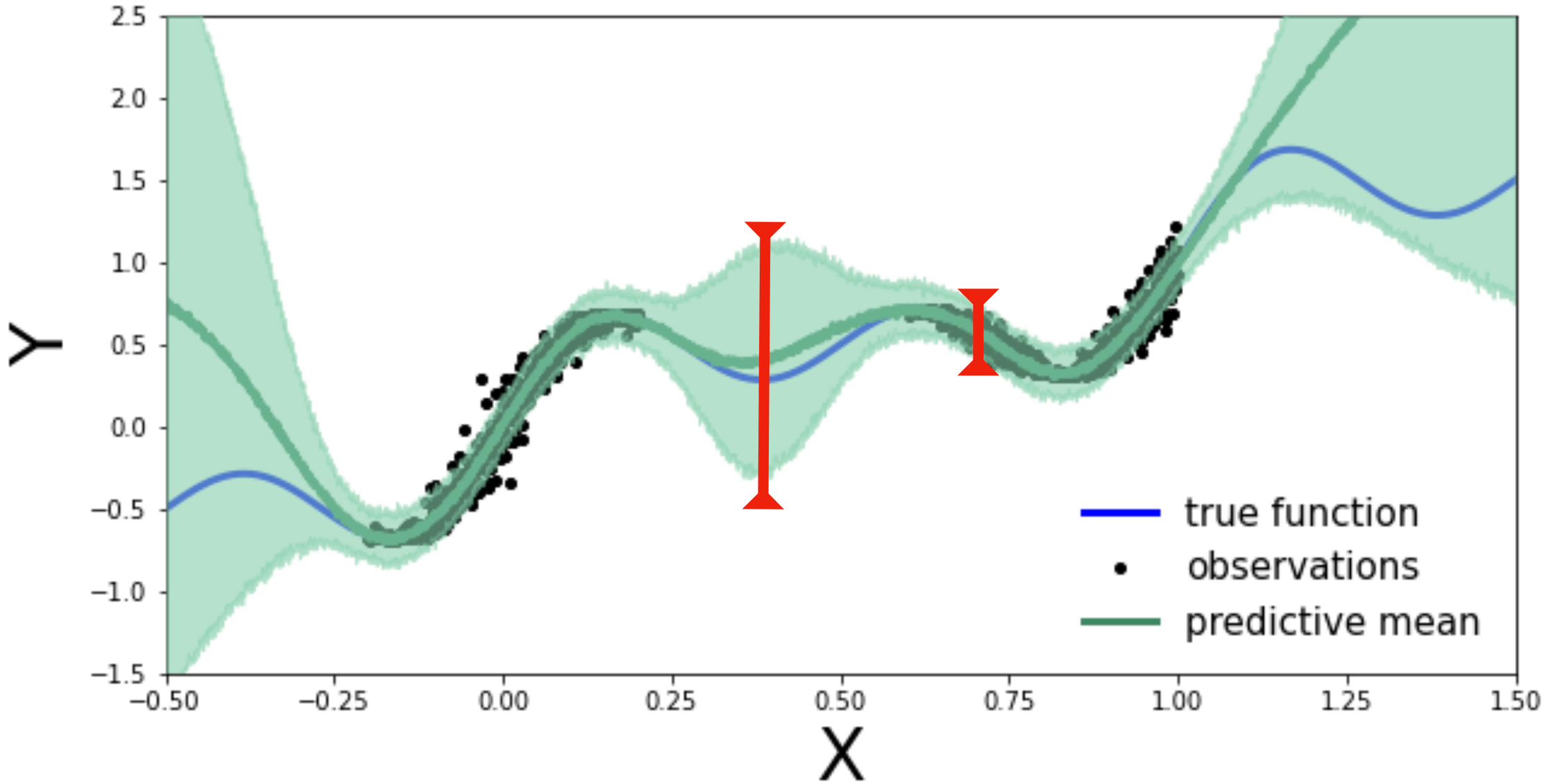
Deep Learning II,
University of Amsterdam

How do we evaluate uncertainty?

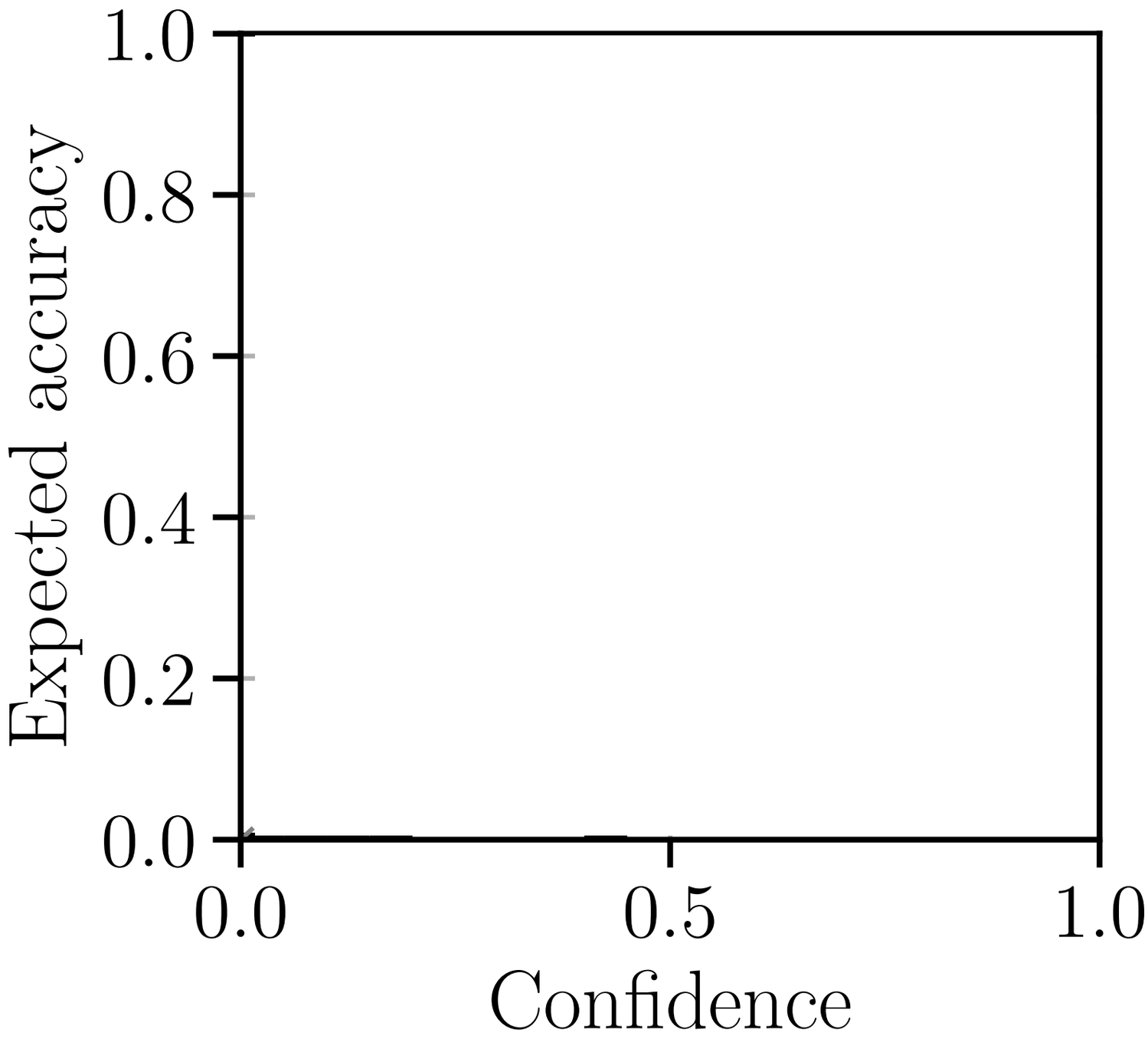




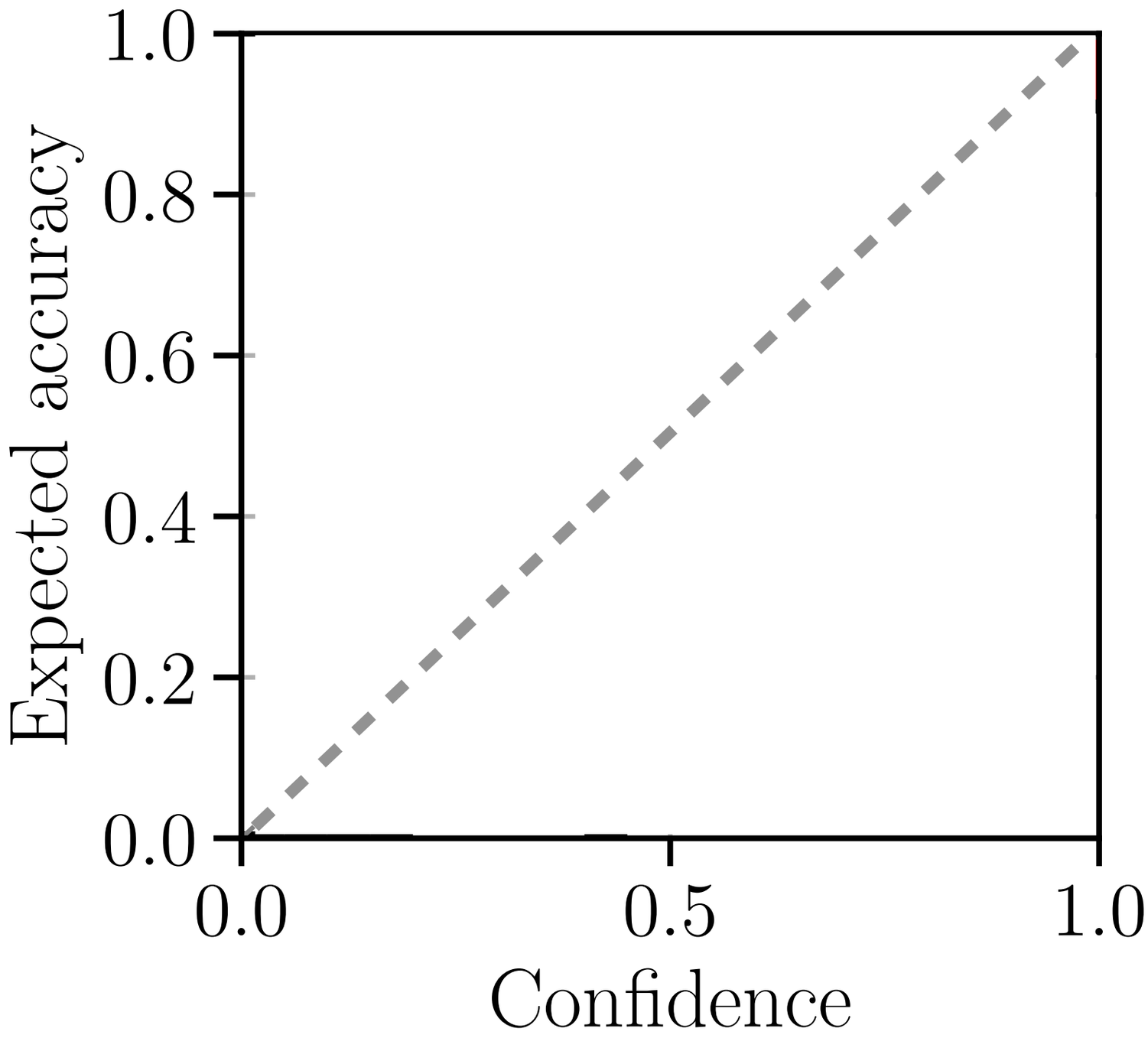
held-out log-likelihood



confidence calibration



confidence calibration



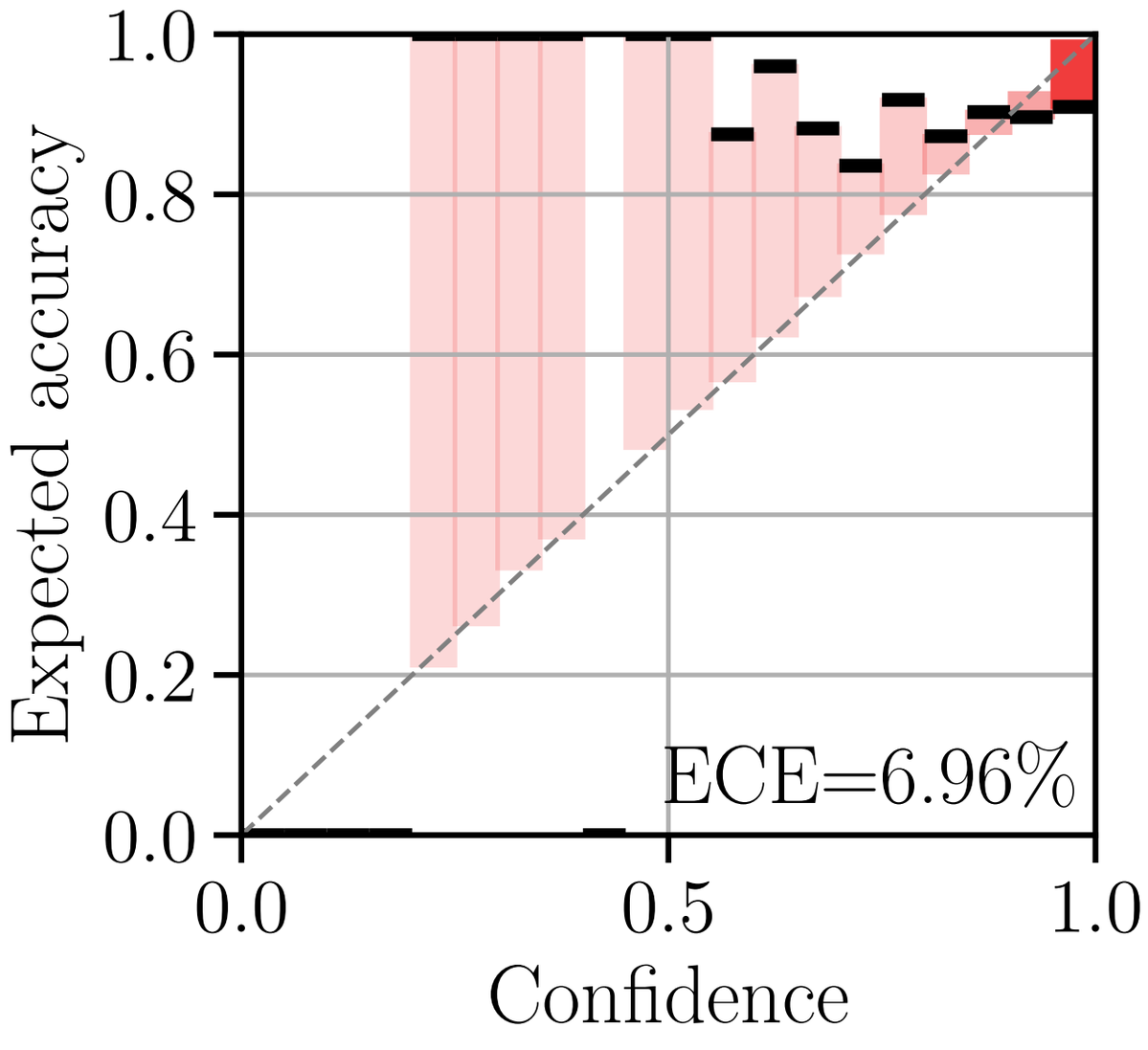
confidence calibration

method #1

method #2

confidence calibration

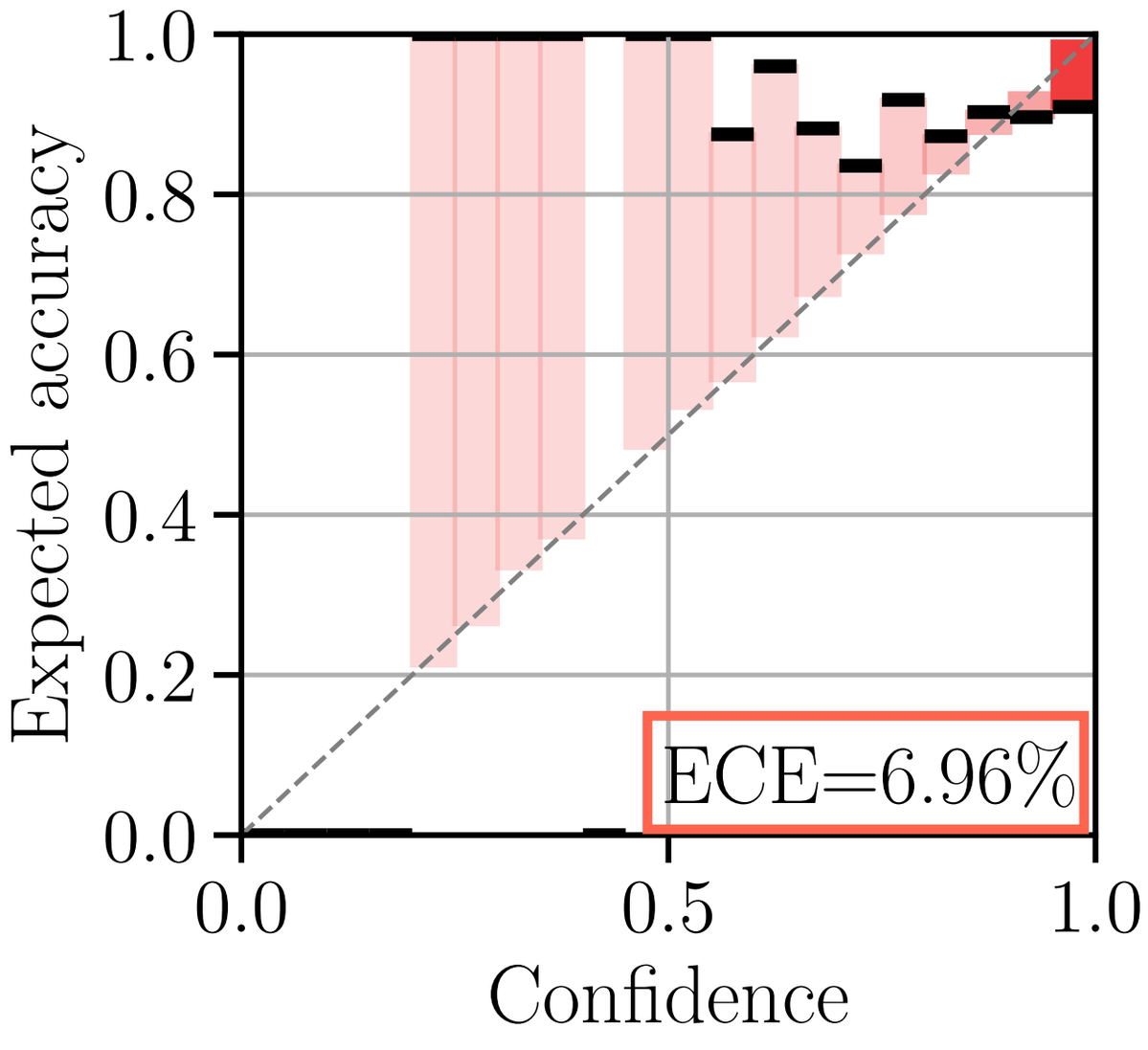
method #1



method #2

confidence calibration

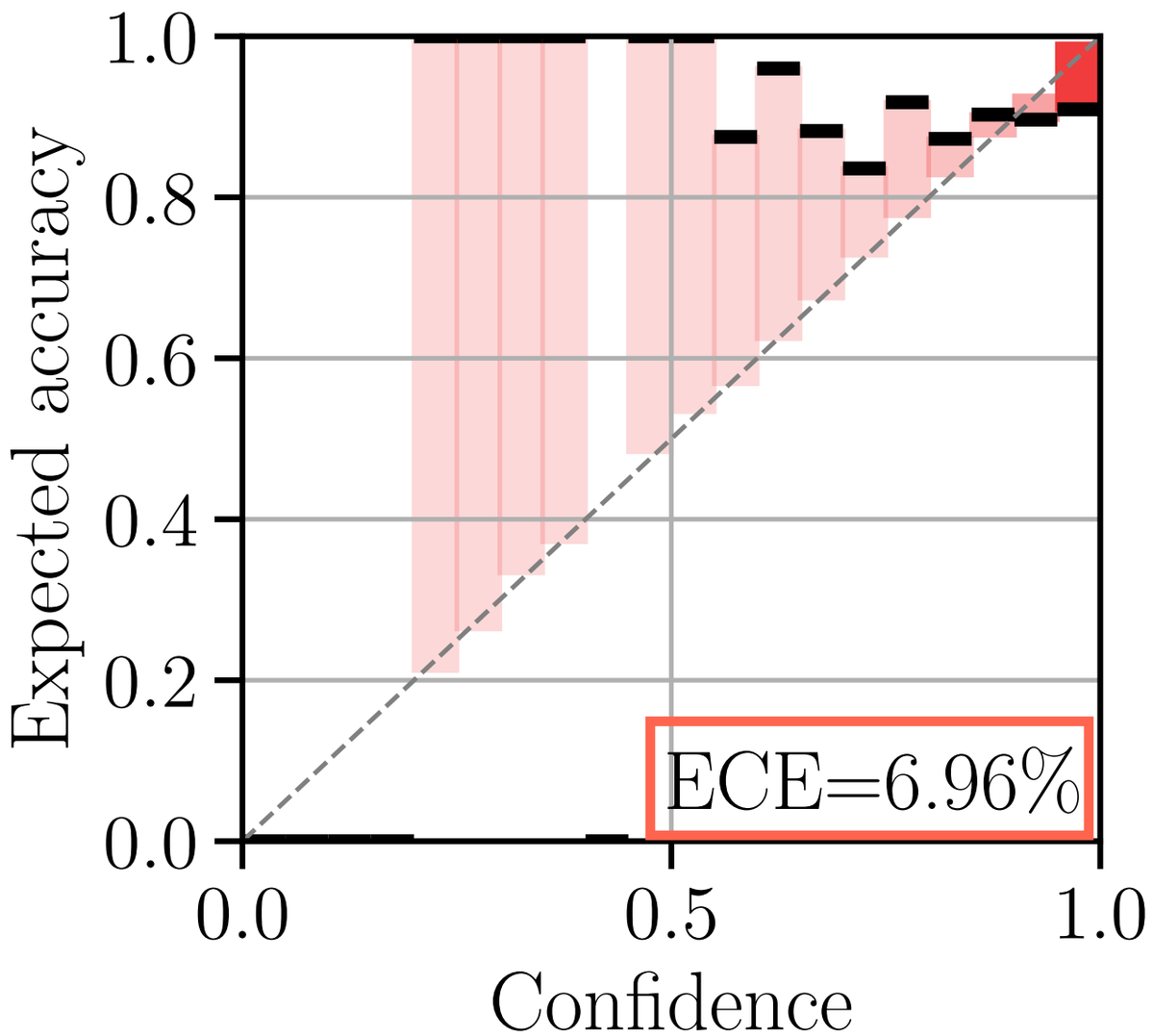
method #1



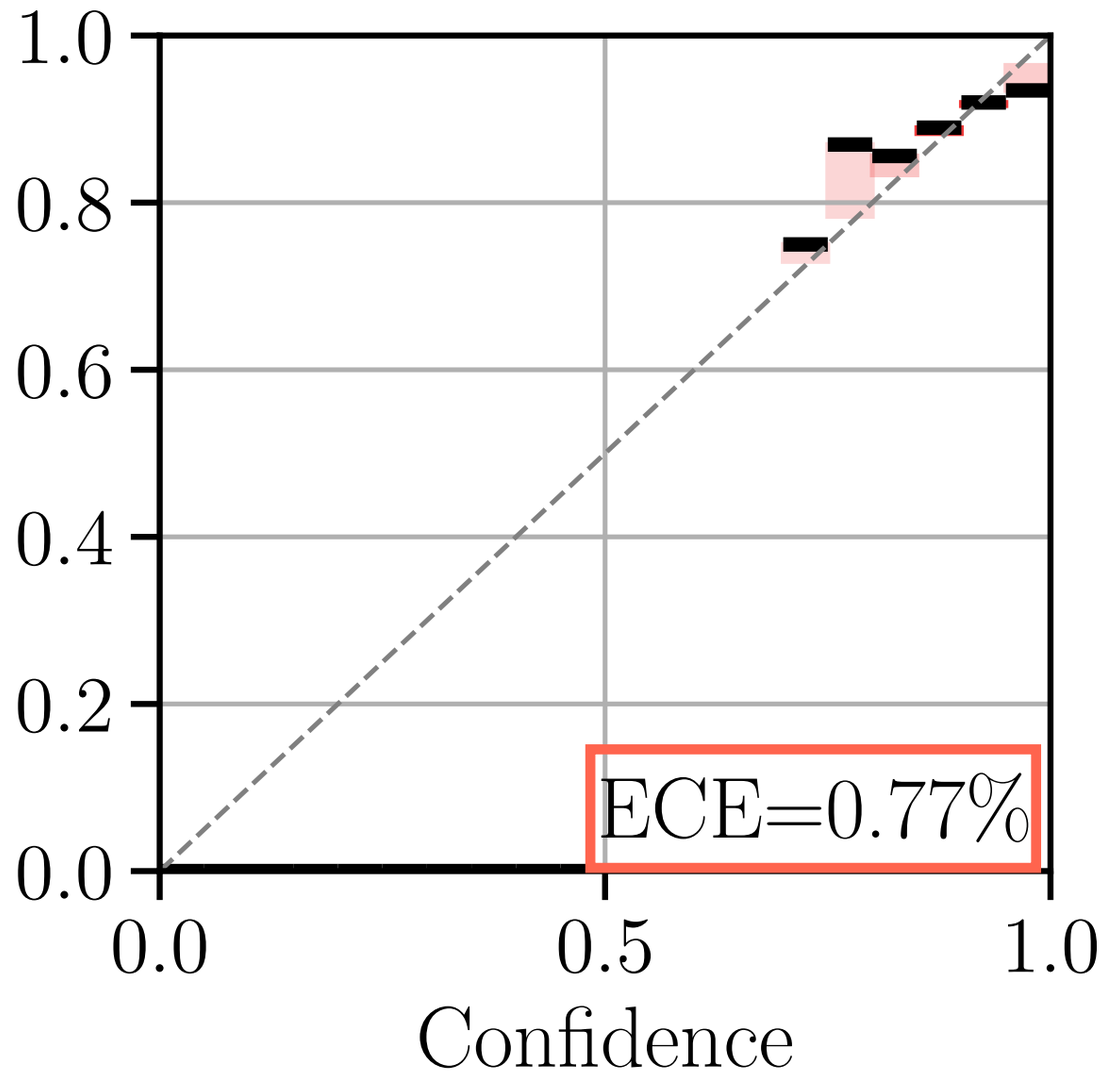
method #2

confidence calibration

method #1



method #2



ranking data points: hard to easy

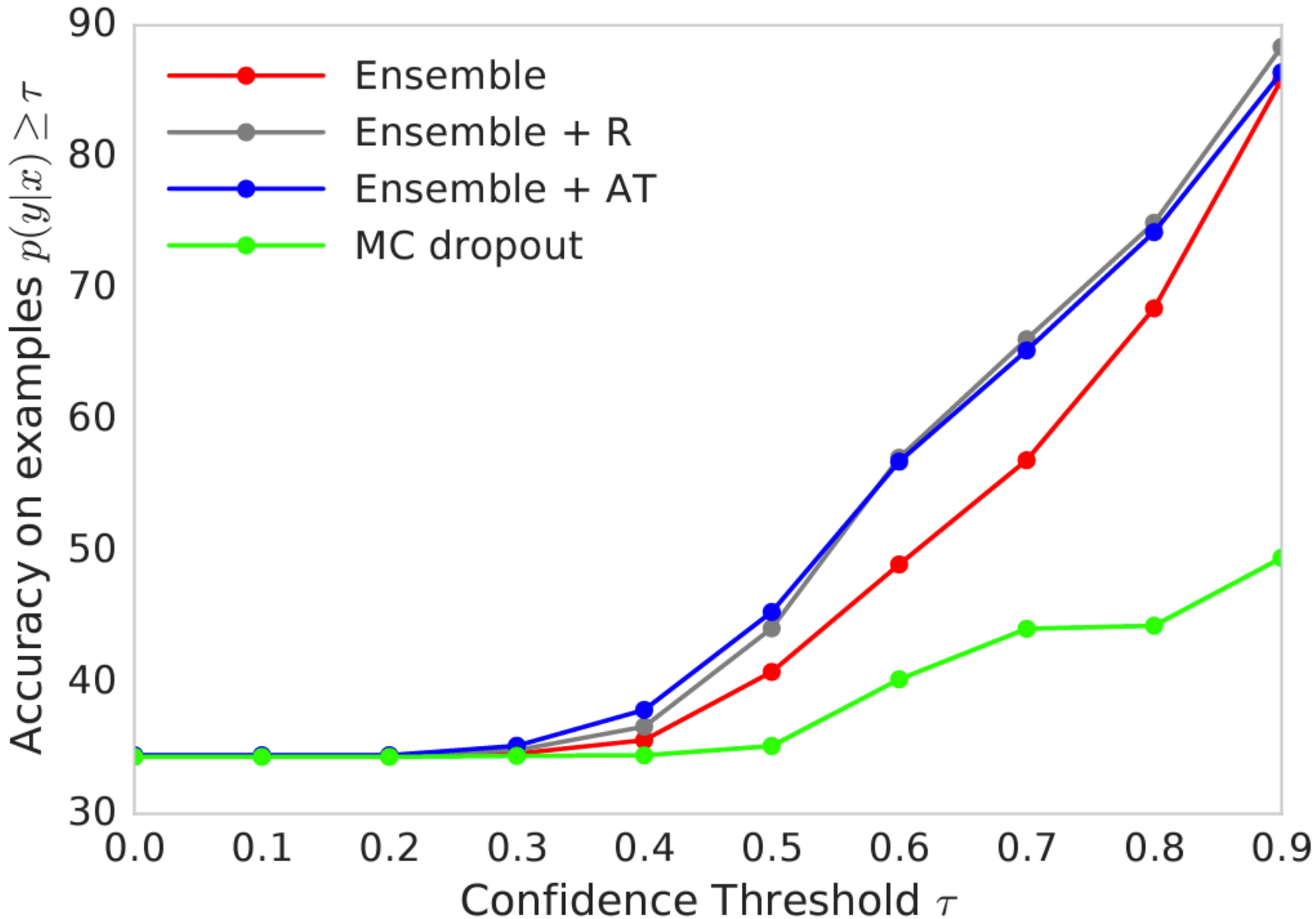


Image from Lakshminarayanan et al., "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles", *NeurIPS 2017*.

data collection / active learning

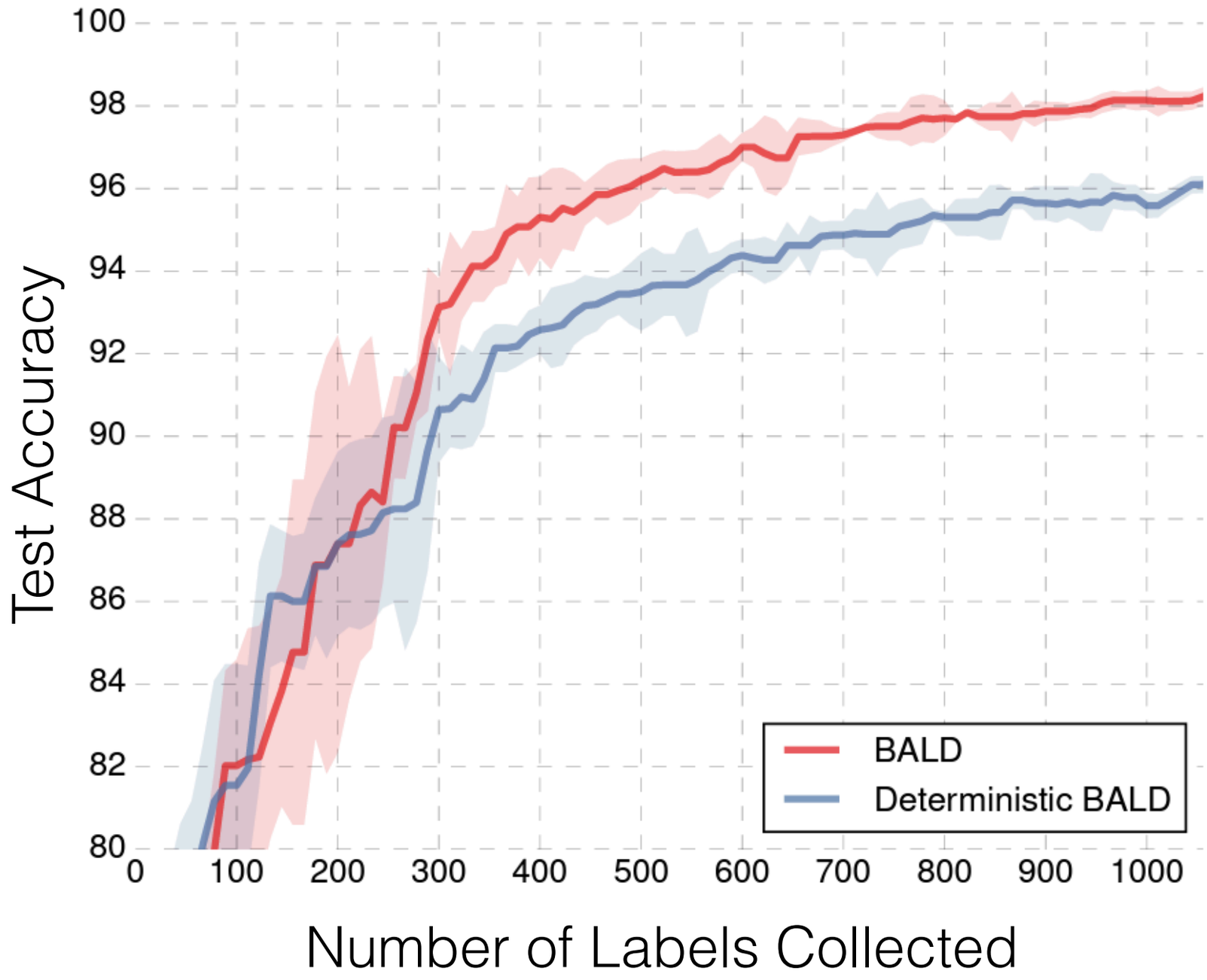
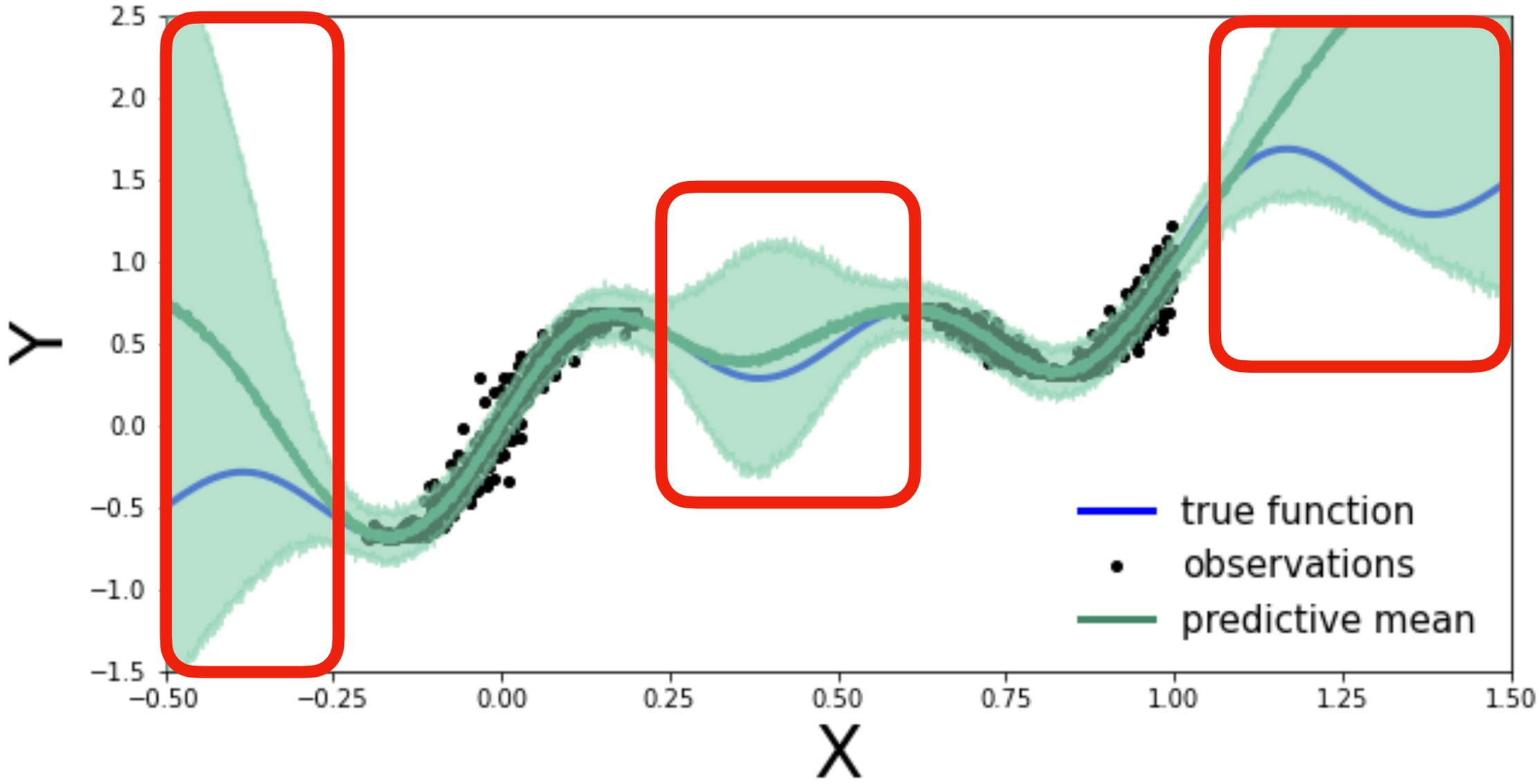
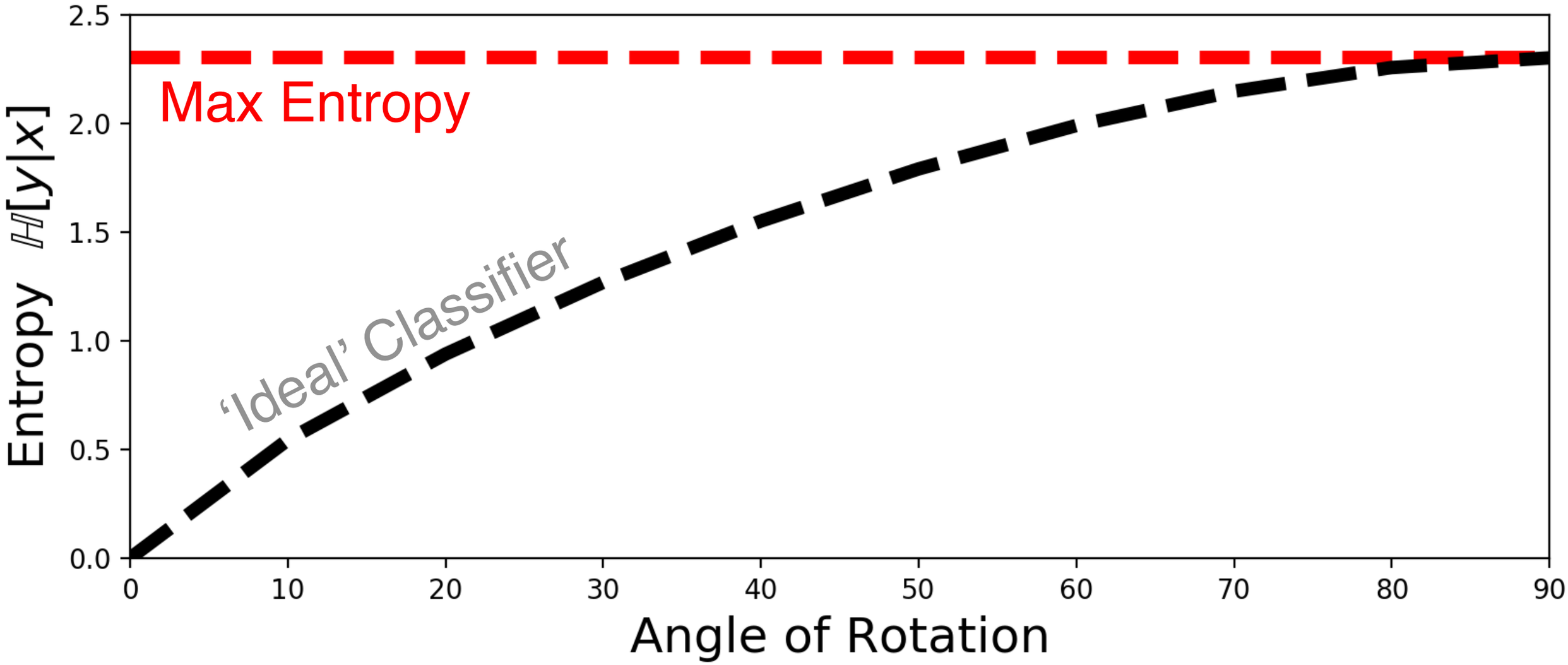


Image from Gal et al., “Deep Bayesian Active Learning with Image Data”, *ICML 2017*.

areas where we haven't seen data



areas where we haven't seen data



areas where we haven't seen data

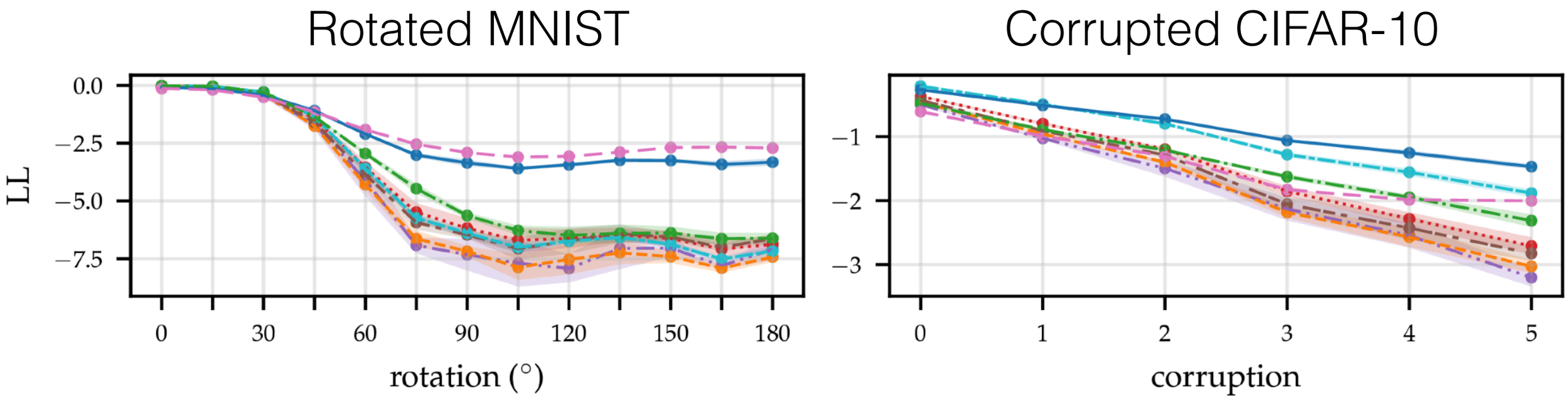


Image from Antoran et al., "Depth Uncertainty in Neural Networks", *NeurIPS 2020*.

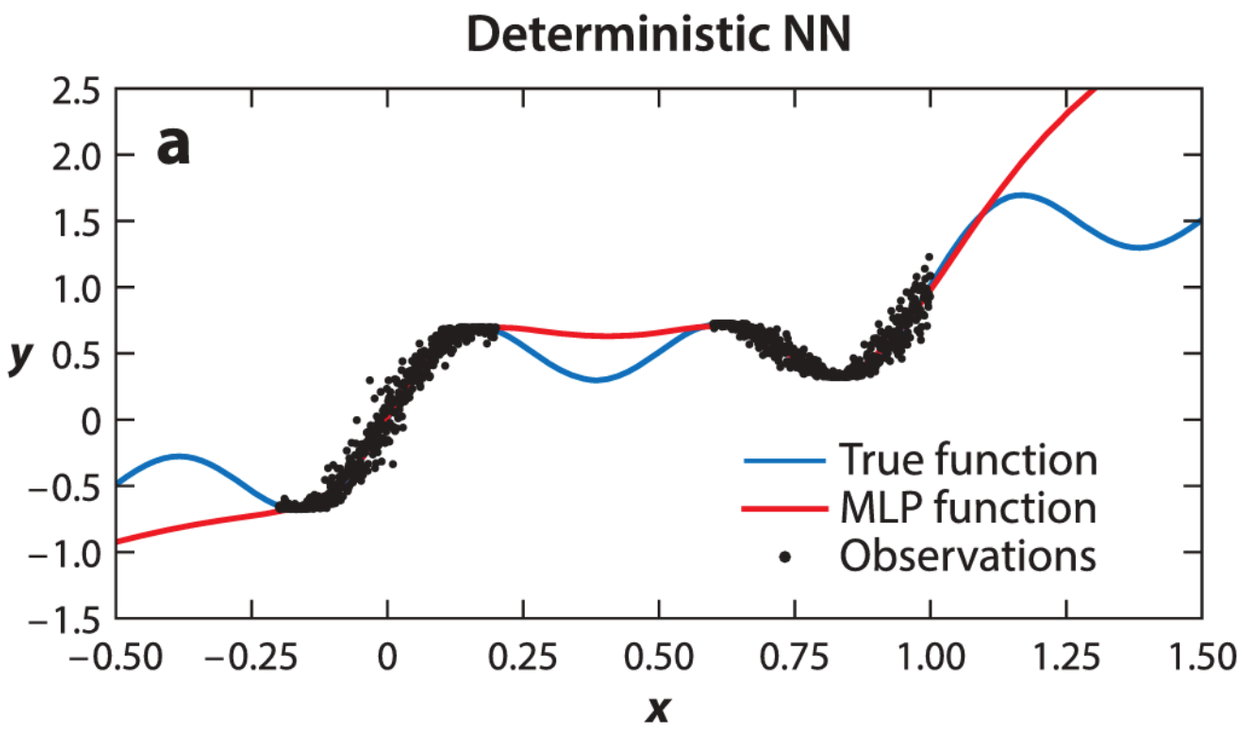
Summary

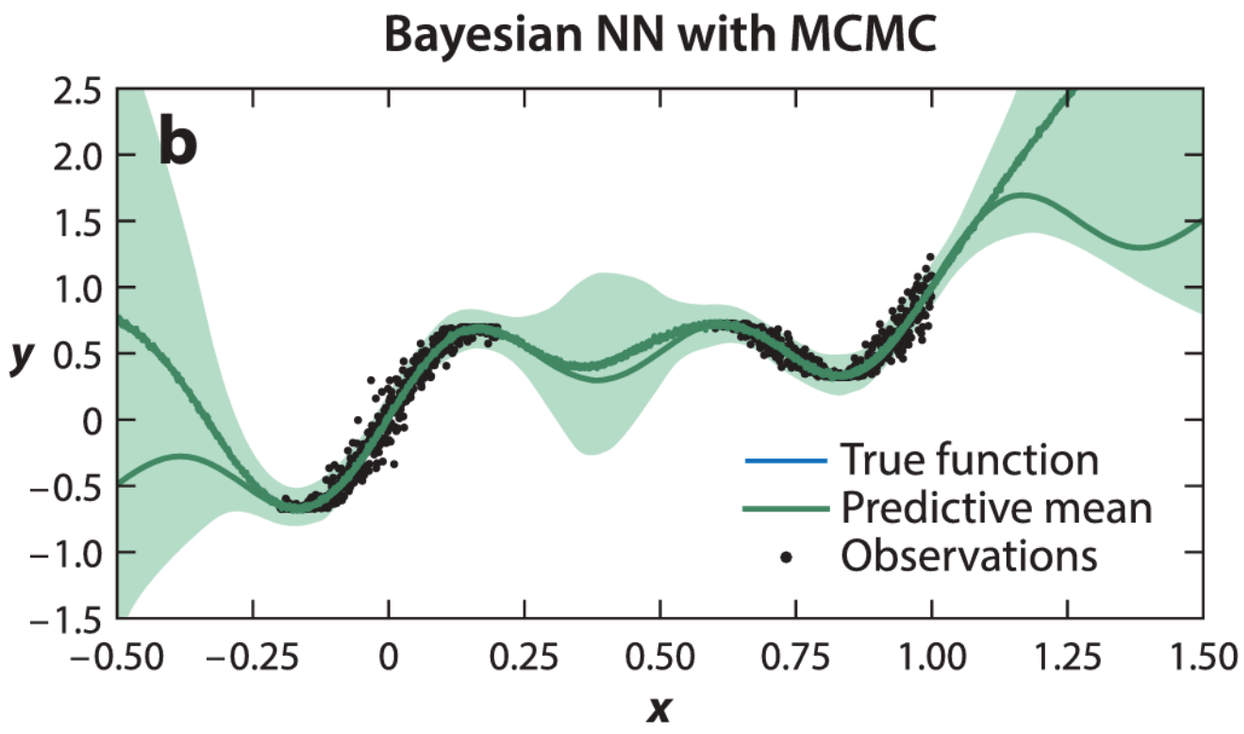
- ⊗ Prediction alone is not sufficient
- ⊗ Use uncertainty estimates to **rank data points**, e.g to collect or abstain.
- ⊗ Evaluate under **distribution shift** .

Uncertainty Quantification: Ensembling

Eric Nalisnick

Deep Learning II,
University of Amsterdam





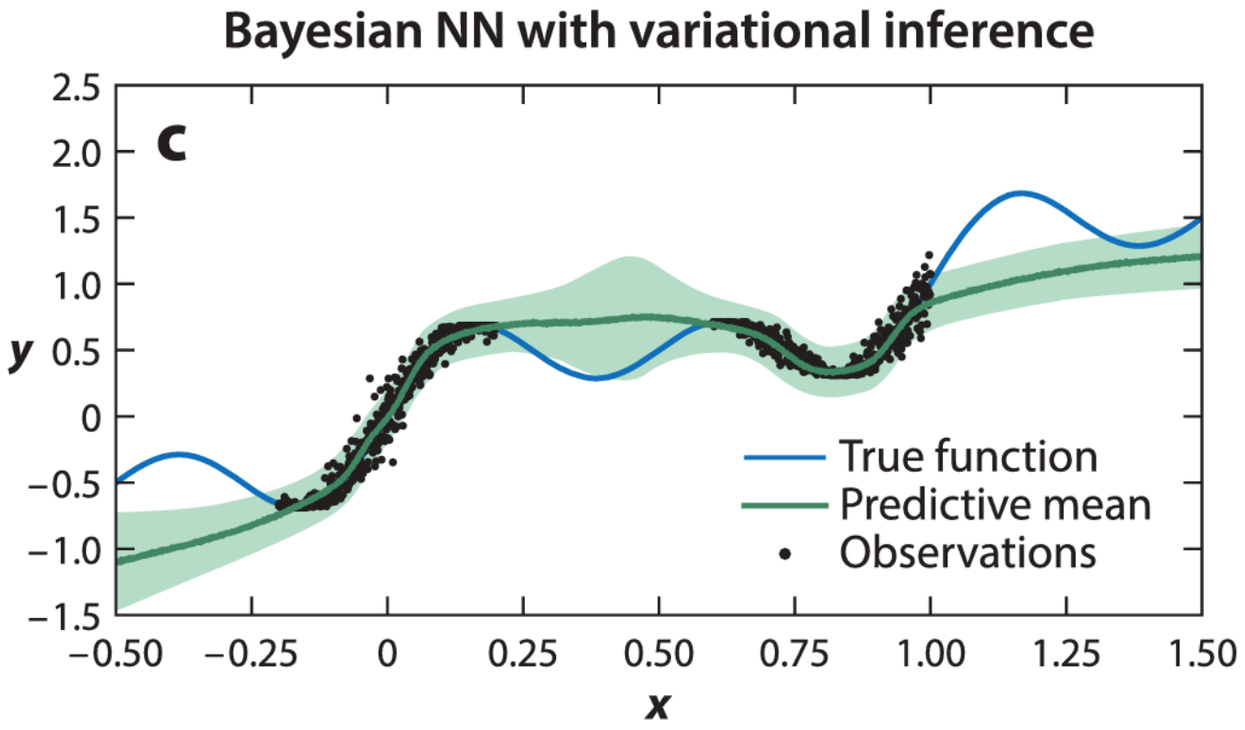
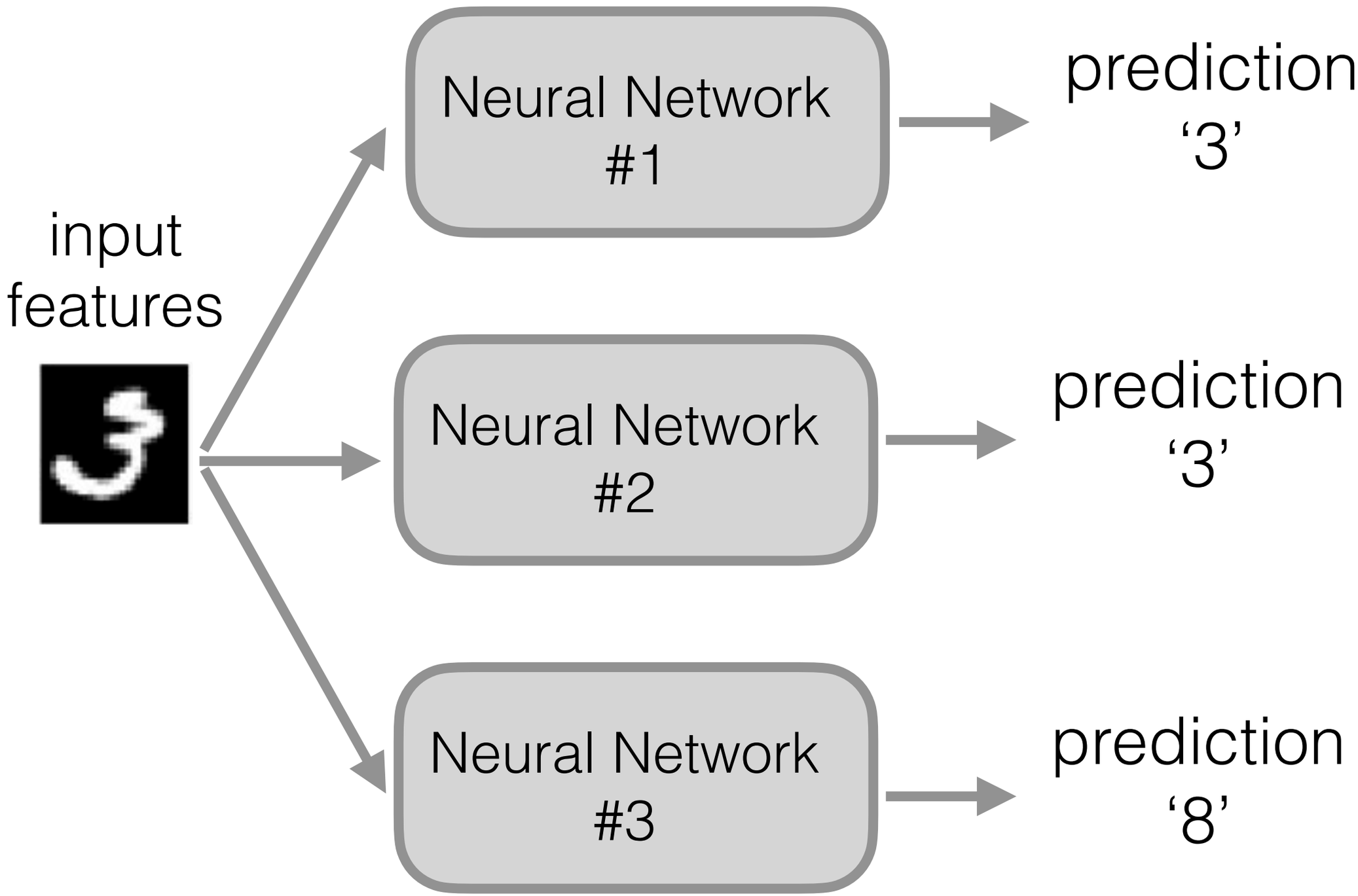
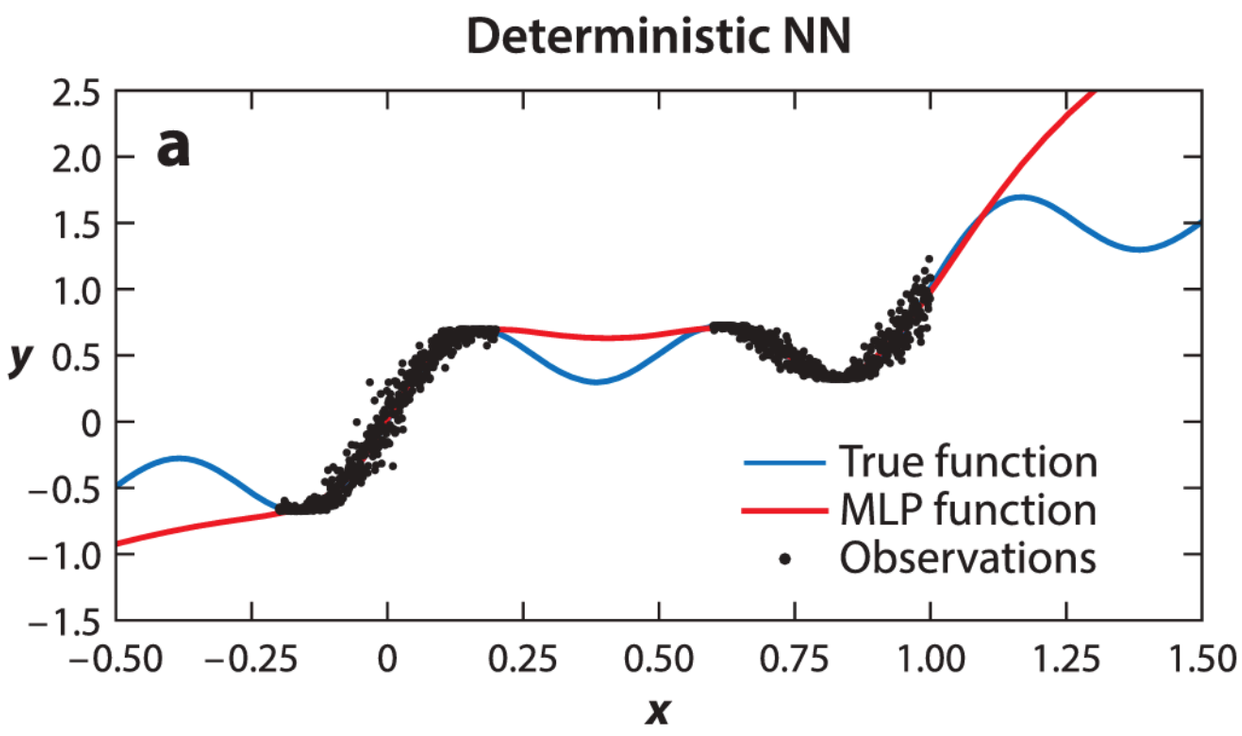
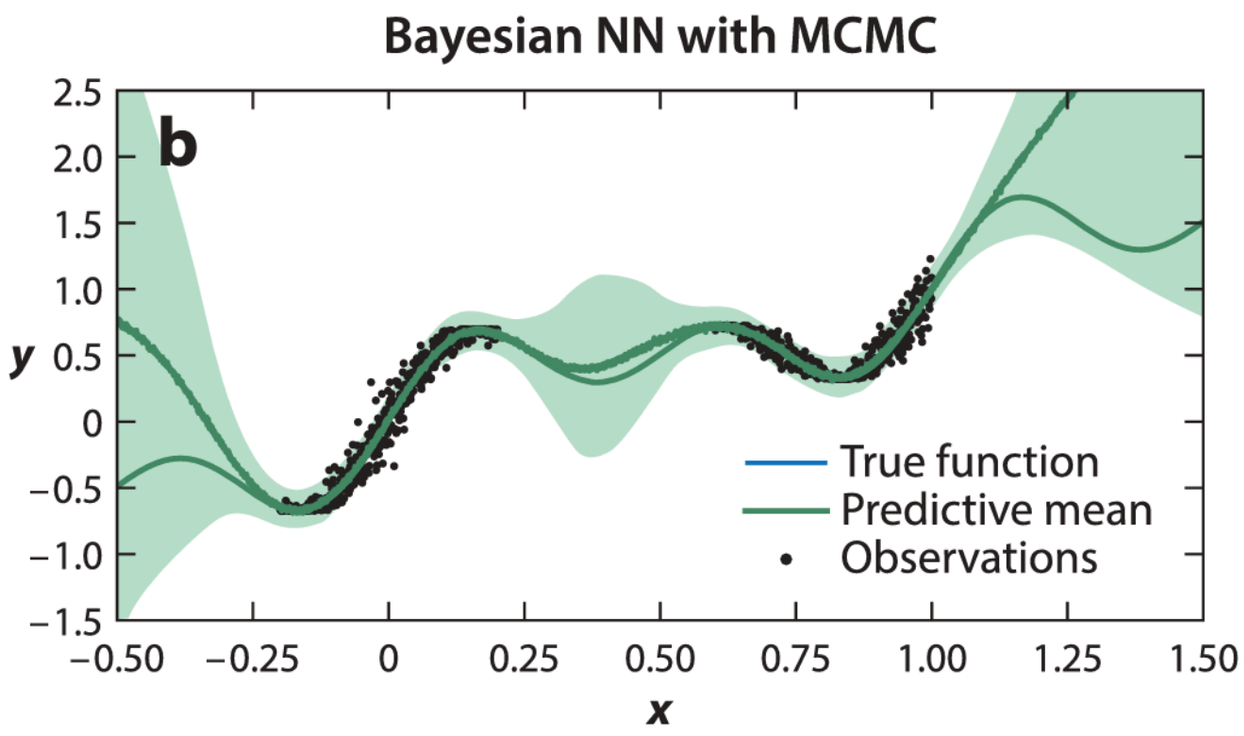


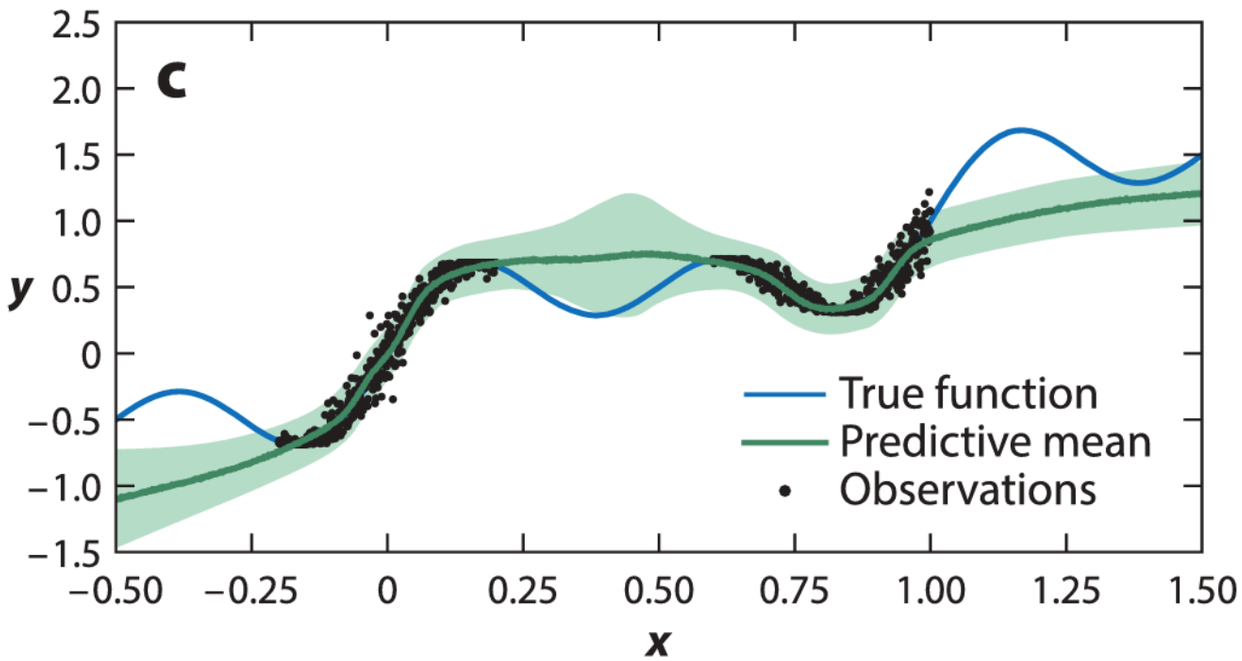
Image from Nalisnick et al., “A Brief Tour of Deep Learning from a Statistical Perspective”, *Annual Review of Statistics*, *2023*.

deep ensemble









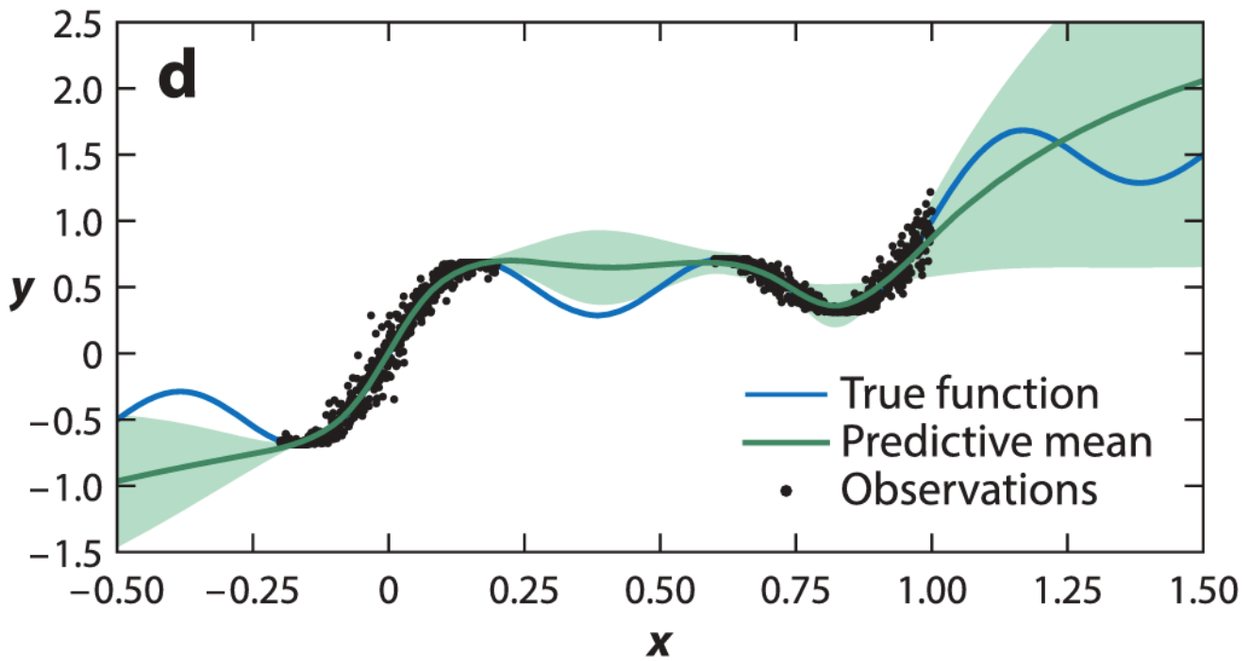
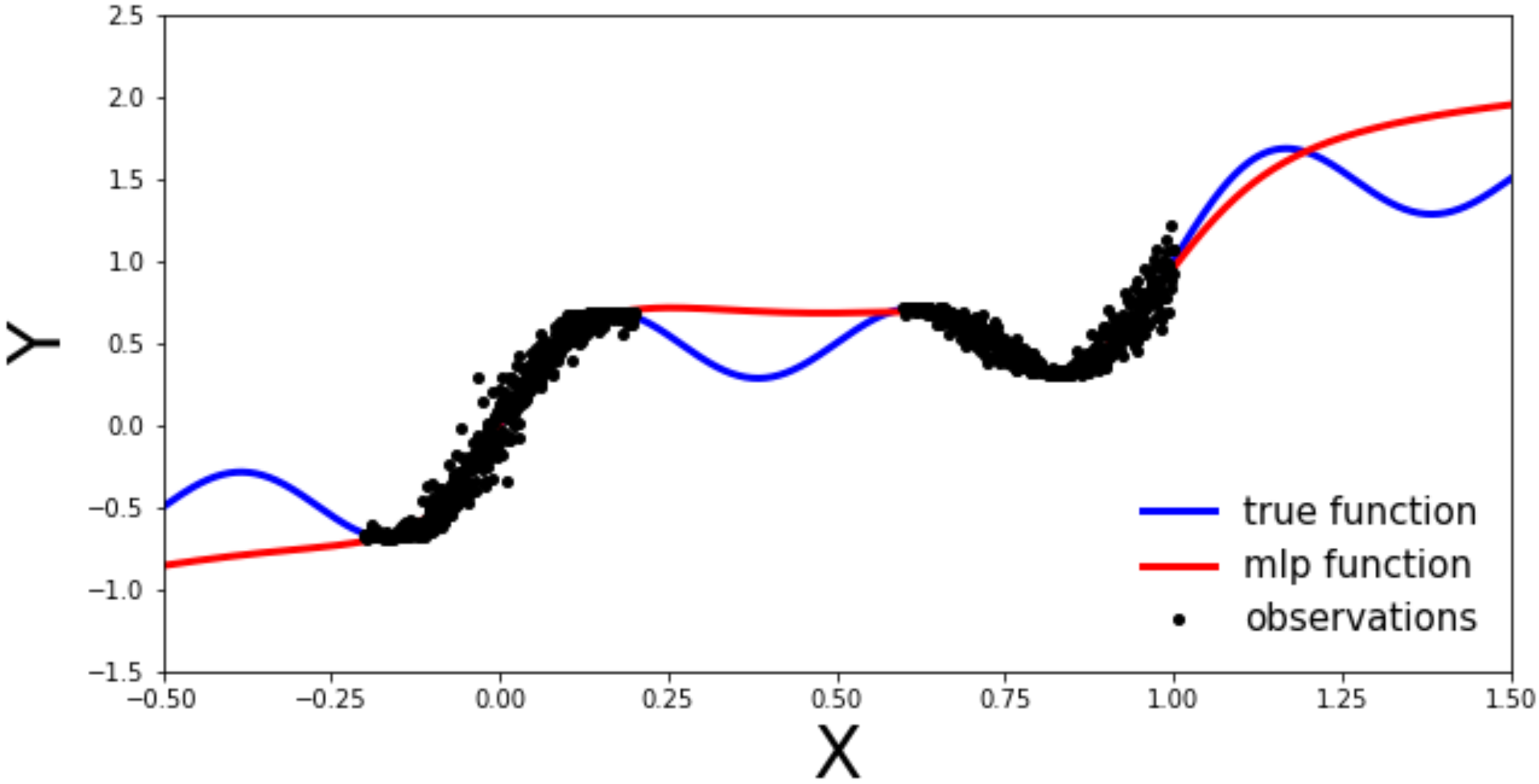
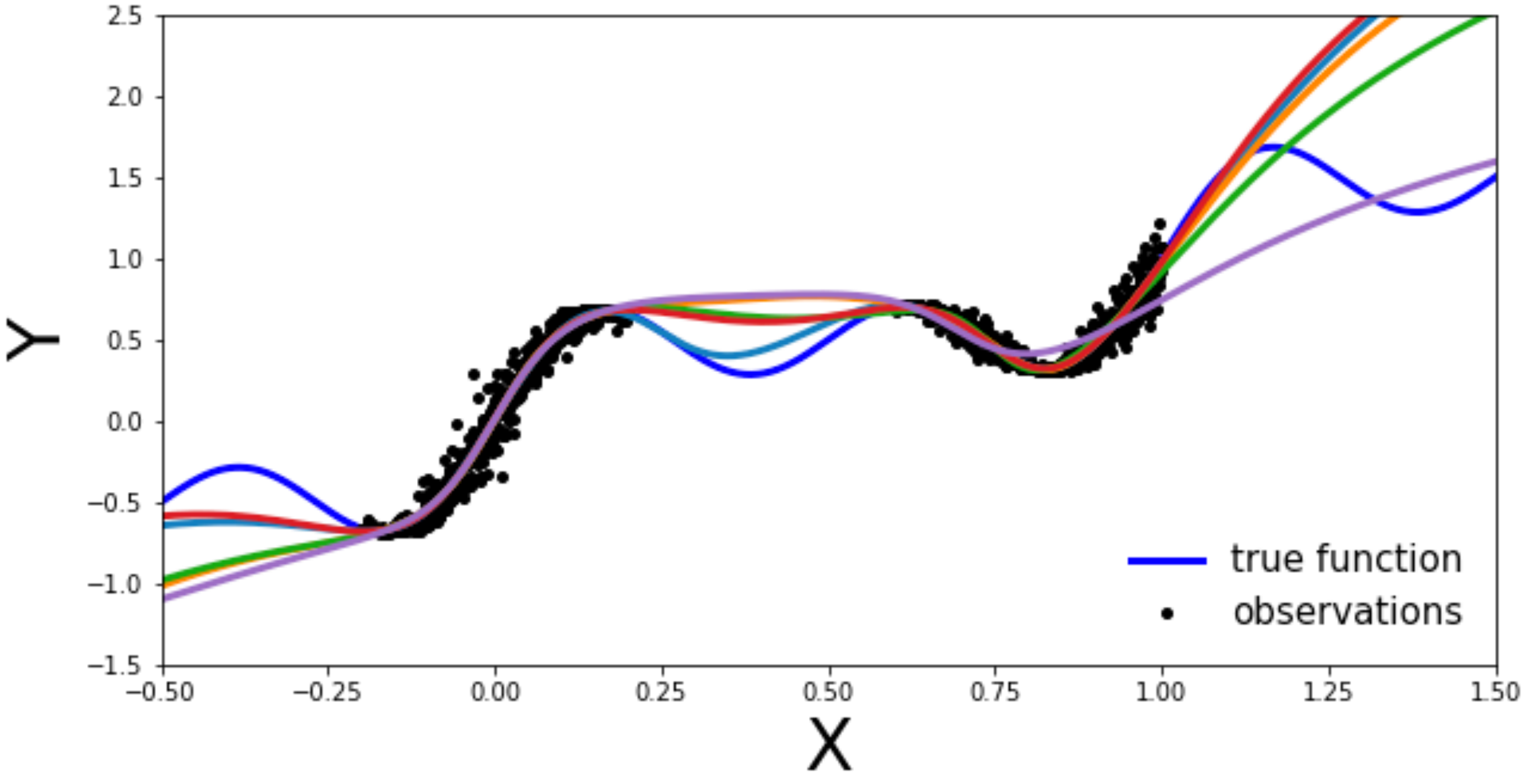


Image from Nalisnick et al., “A Brief Tour of Deep Learning from a Statistical Perspective”, *Annual Review of Statistics*, *2023*.


2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
Y
X
true function
mlp function
observations


Y
X
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
true function
observations

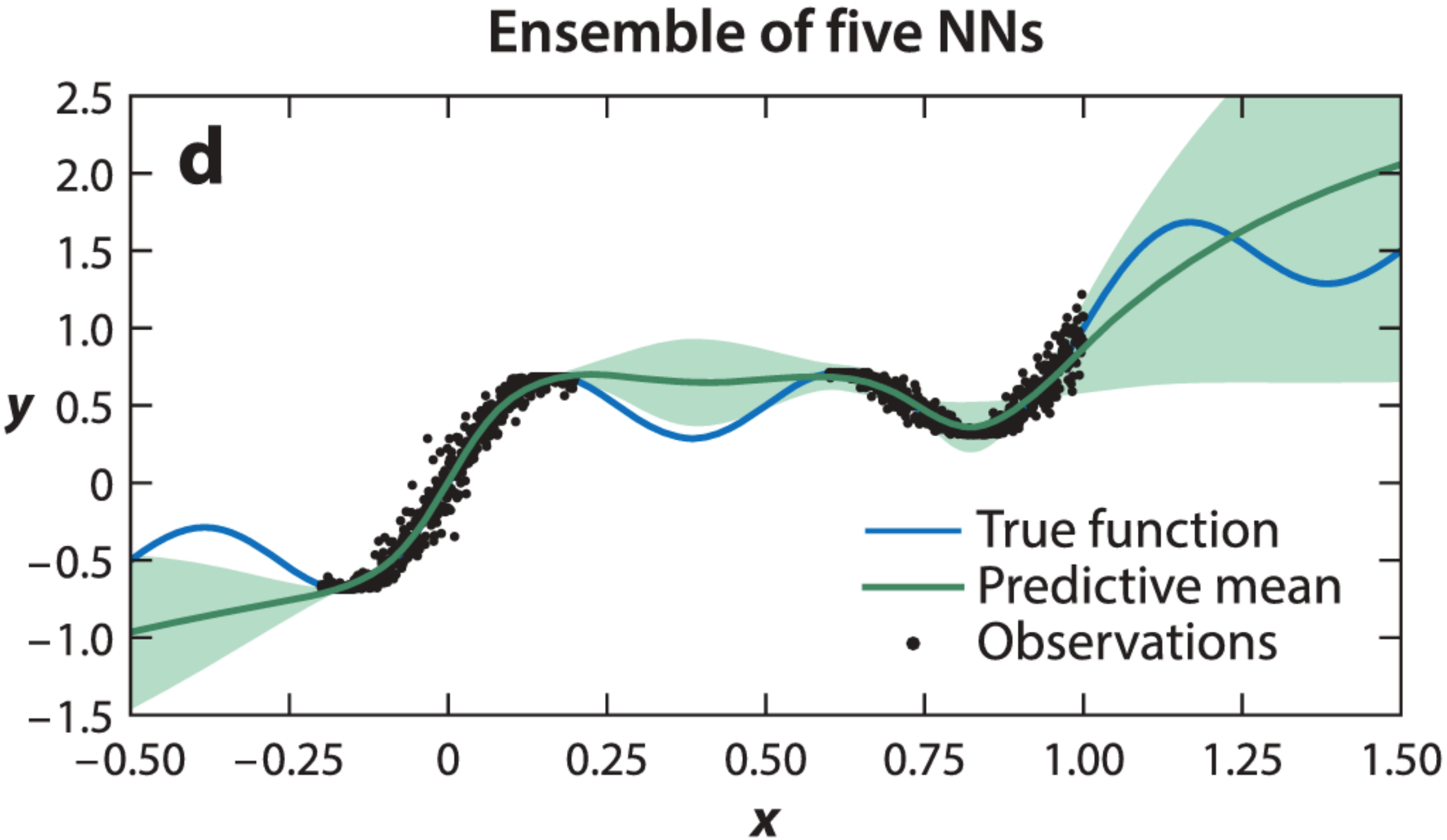
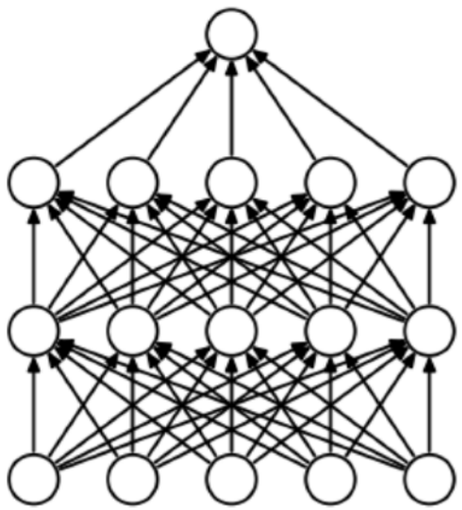


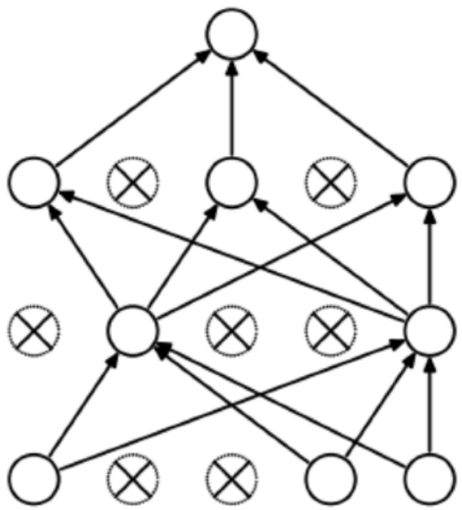
Image from Nalisnick et al., “A Brief Tour of Deep Learning from a Statistical Perspective”, *Annual Review of Statistics*, *2023*.

dropout

Randomly set hidden units to zero (i.e. drop them) for every forward pass during training [Hinton et al., 2012].



Standard Neural Network

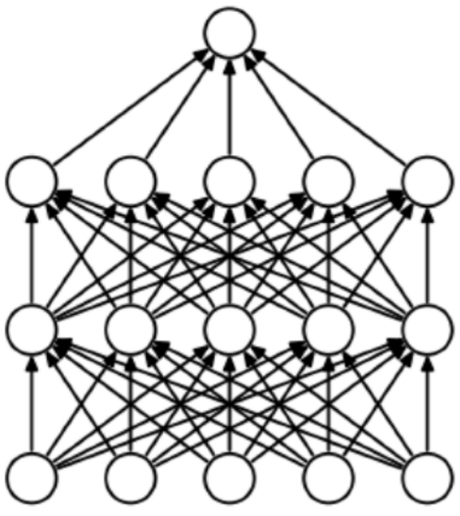


After Applying Dropout

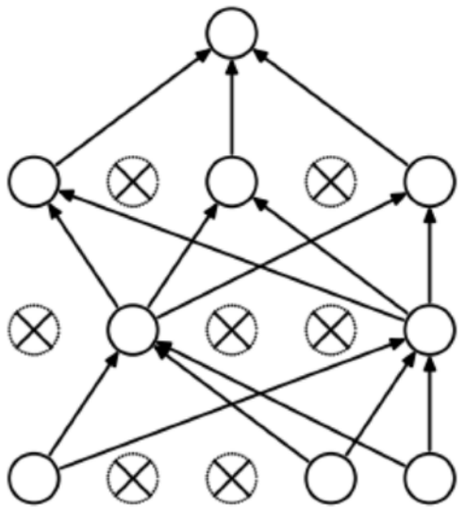
Image from Srivastava et al., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *JMLR 2014*.

dropout

Randomly set hidden units to zero (i.e. drop them) for every forward pass during training [Hinton et al., 2012].



Standard Neural Network



After Applying Dropout

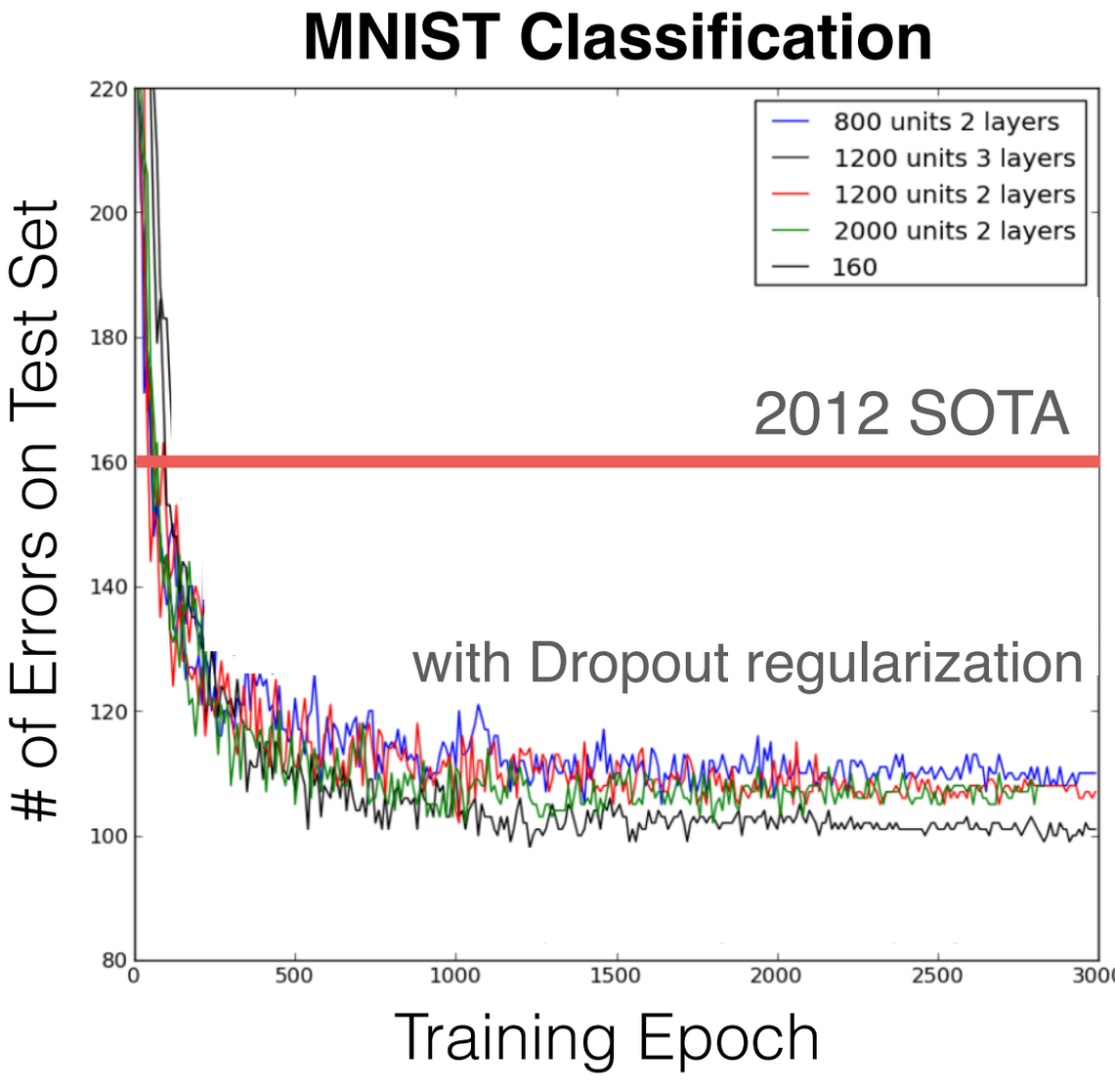


Image from Srivastava et al., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *JMLR 2014*.

Bayesian interpretation of dropout

- Gal & Ghahramani [2016] argued that dropout can be interpreted as variational Bayesian inference.

  $p(\lambda)$ serves as a variational approximation to the posterior $p(\lambda \mid \mathbf{y}, \mathbf{X})$

Bayesian interpretation of dropout

- Gal & Ghahramani [2016] argued that dropout can be interpreted as variational Bayesian inference.

- Obtain predictive uncertainty by averaging over noise samples.

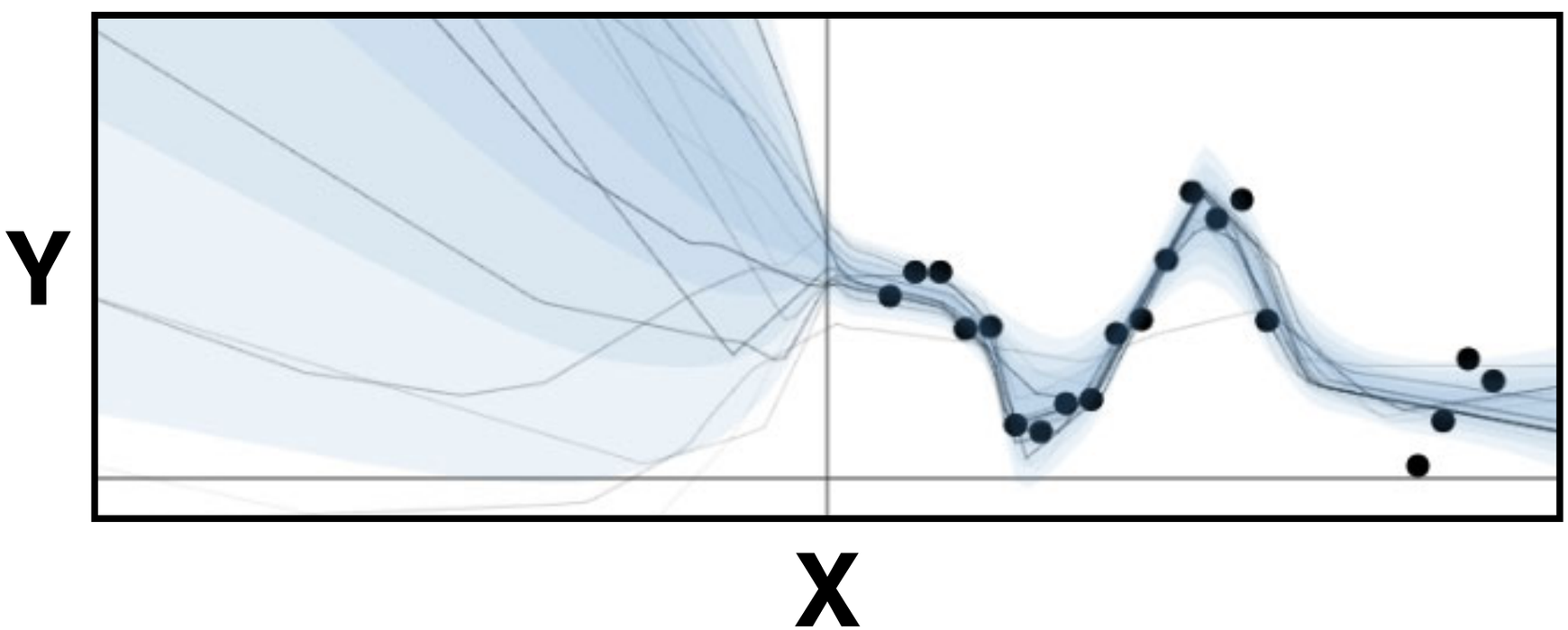
$$p(y^* \mid \mathbf{x}^*, \mathbf{y}, \mathbf{X}) \approx \frac{1}{S} \sum_{s=1}^{S} p(y^* \mid \mathbf{x}^*, \{\hat{\mathbf{\Lambda}}_{l,s}\}_{l=1}^{L+1}), \qquad \hat{\lambda} \sim p(\lambda)$$

Bayesian interpretation of dropout

- Gal & Ghahramani [2016] argued that dropout can be interpreted as variational Bayesian inference.

- Obtain predictive uncertainty by averaging over noise samples.

$$p(y^* \mid \mathbf{x}^*, \mathbf{y}, \mathbf{X}) \approx \frac{1}{S} \sum_{s=1}^{S} p(y^* \mid \mathbf{x}^*, \{\hat{\mathbf{\Lambda}}_{l,s}\}_{l=1}^{L+1}), \qquad \hat{\lambda} \sim p(\lambda)$$



Bayesian interpretation of dropout

- Gal & Ghahramani [2016] argued that dropout can be interpreted as variational Bayesian inference.

- Obtain predictive uncertainty by averaging over noise samples.

- However, p(λ)—their posterior approximation—**is fixed**, **does not depend on data**. So how can it truly quantify model uncertainty?

Bayesian interpretation of dropout

- Gal & Ghahramani [2016] argued that dropout can be interpreted as variational Bayesian inference.

- Obtain predictive uncertainty by averaging over noise samples.

- However, p(λ)—their posterior approximation—**is fixed**, **does not depend on data**. So how can it truly quantify model uncertainty?

- Subsequent work has attempted to fix this by optimizing the noise distribution. [Maeda, 2014; Kingma et al., 2015; Molchanov et al., 2017; Gal et al., 2017]

Bayesian interpretation of dropout

Dropout as a Structured Shrinkage Prior

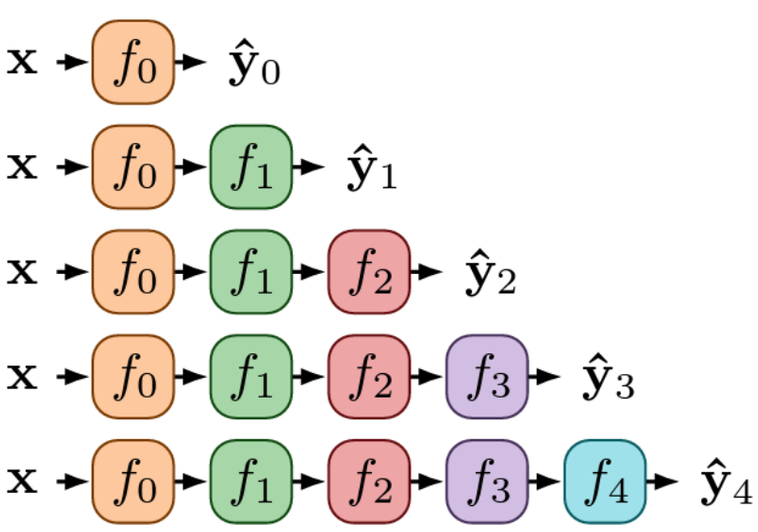
Eric Nalisnick [1] **José Miguel Hernández-Lobato** [1 2 3] **Padhraic Smyth** [4]

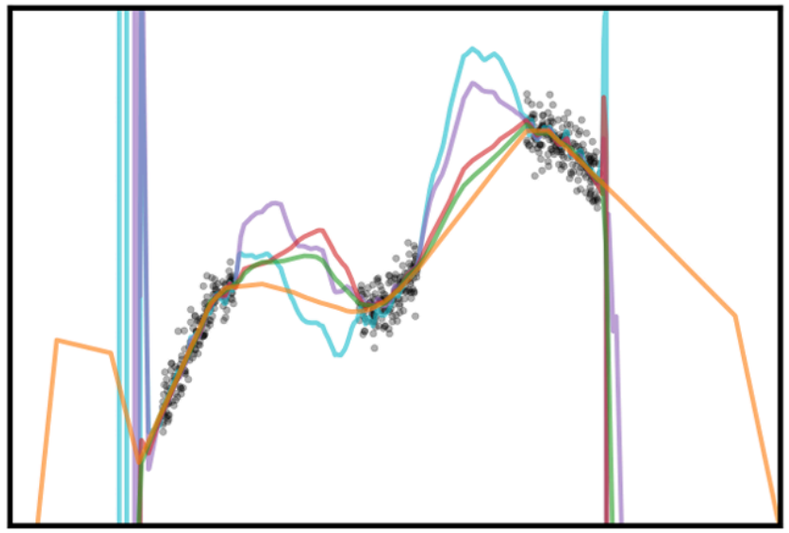
Abstract

Dropout regularization of deep neural networks has been a mysterious yet effective tool to prevent overfitting. Explanations for its success range from the prevention of "co-adapted" weights to it being a form of cheap Bayesian inference. We propose a novel framework for understanding multiplicative noise in neural networks, considering continuous distributions as well as Bernoulli noise (i.e. dropout). We show that multiplicative noise induces structured shrinkage priors on a network's weights. We derive the equivalence through reparametrization properties of scale mixtures and without invoking any approx-

explain dropout's inner-workings and success are also plentiful. To give a few examples, Srivastava et al. (2014) argue it prevents "conspiracies" between hidden units, Hinton et al. (2012) claim it serves a role similar to sex in evolution, Baldi & Sadowski (2013) show it ensembles by taking the geometric mean of sub-models, Wager et al. (2013) explain it as an adaptive ridge penalty, and Gal & Ghahramani (2016b) suggest dropout performs quasi-Bayesian uncertainty estimation. While some prior work has shown strict equivalences for simple models such as linear regression (Baldi & Sadowski, 2013; Wager et al., 2013; 2014; Helmbold & Long, 2015), the general case of dropout in deep neural networks is analytically intractable, which is likely why no one narrative has come to prominence.

depth uncertainty





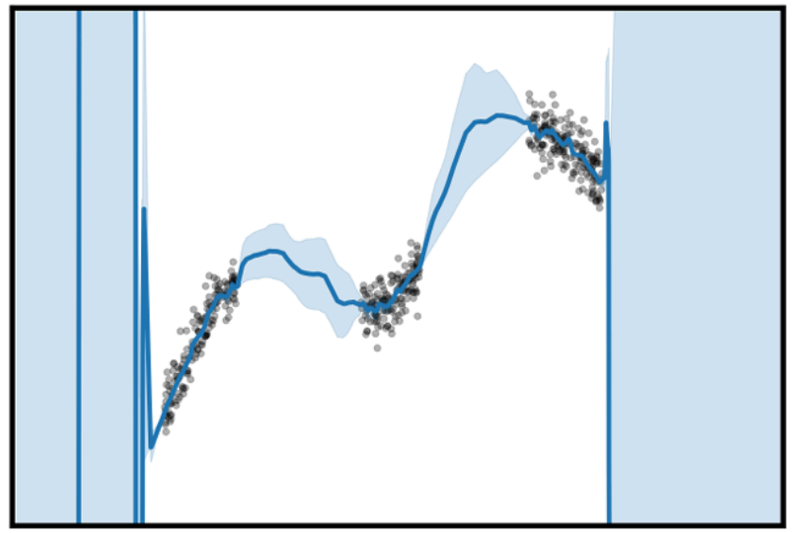
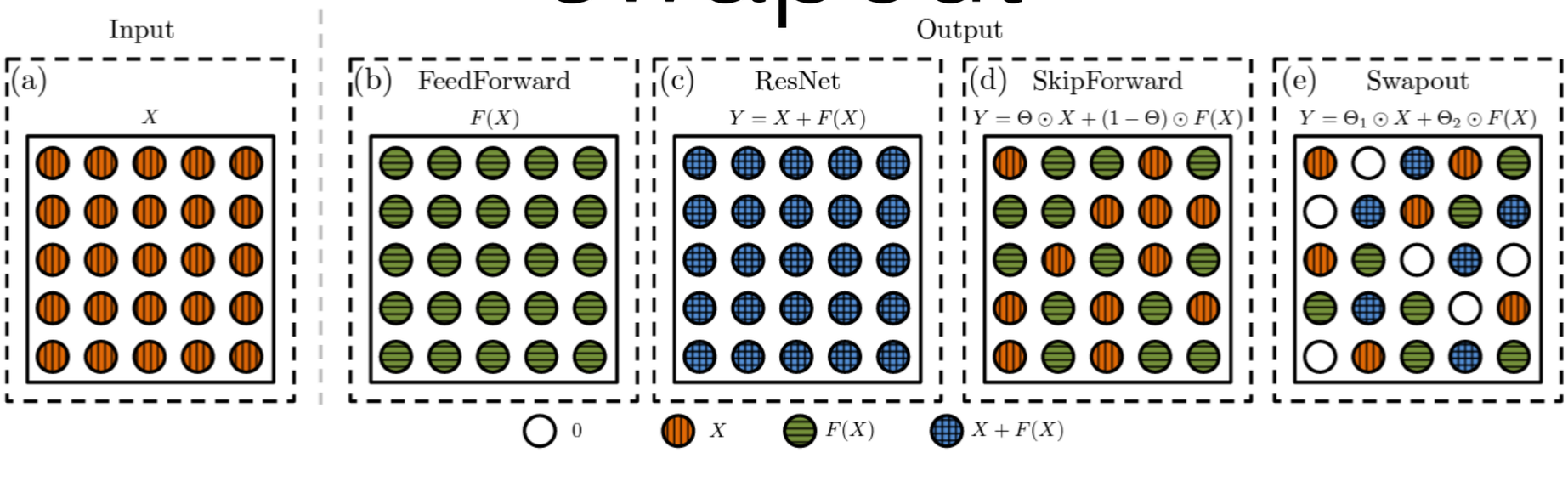


Image from Antoran et al., “Depth Uncertainty in Neural Networks”, *NeurIPS 2020*.

ensemble over everything

swapout



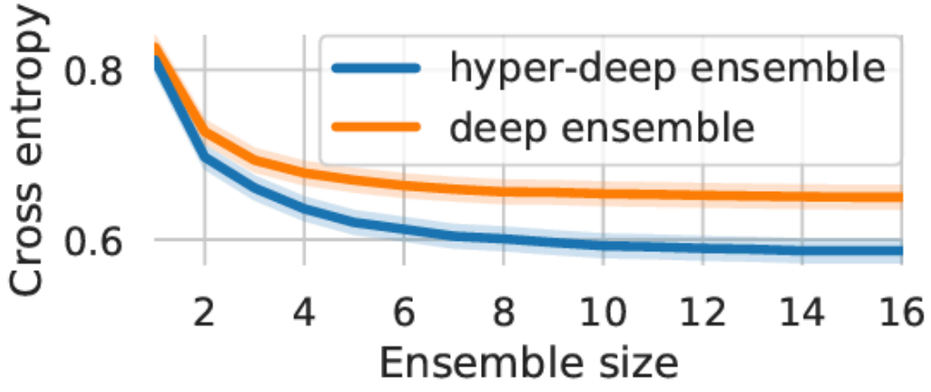


Figure 1: Comparison of our hyper-deep ensemble with deep ensemble for different ensemble sizes using a Wide ResNet 28-10 over CIFAR-100. Combining models with different hyperparameters is beneficial.

Image from Singh et al., “Swapout: learning an ensemble of deep architectures”, *NeurIPS 2016*.

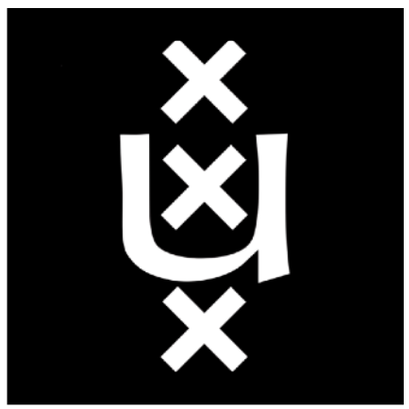
Summary

- ⊗ **Naive ensembles effectively quantify uncertainty**
  (due to overparameterization)

- ⊗ Can ensemble over various sub-structures (networks, layers, hidden units, hyper parameters, etc).

Uncertainty Quantification:
Conformal Inference

Eric Nalisnick



Deep Learning II,
University of Amsterdam

conformal inference

conformal inference

construct a confidence set over labels:

$$\mathbb{P}\left(\mathbf{y}^*_{N+1} \in \mathbf{C}(\mathbf{x}_{N+1})\right) \geq 1 - \alpha$$

conformal inference

construct a confidence set over labels:

$$\mathbb{P}\left(\mathbf{y}^*_{N+1} \in \mathbf{C}(\mathbf{x}_{N+1})\right) \geq 1 - \alpha$$

{ fox squirrel 0.99 }

{ fox squirrel 0.82, gray fox, 0.03, bucket, 0.02, rain barrel 0.02 }

{ marmot, 0.30, fox squirrel, 0.22, mink, 0.18, weasel, 0.16, beaver, 0.03, polecat 0.01 }

Image from Angelopoulos & Bates, “A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification”, *2022*.

conformal inference: train-time

conformal inference: train-time

$p(y = 1 \mid x)$ $p(y = 2 \mid x)$ $p(y = 3 \mid x)$

conformal inference: train-time

$p(y = 1 \mid x)$ $p(y = 2 \mid x)$ $p(y = 3 \mid x)$

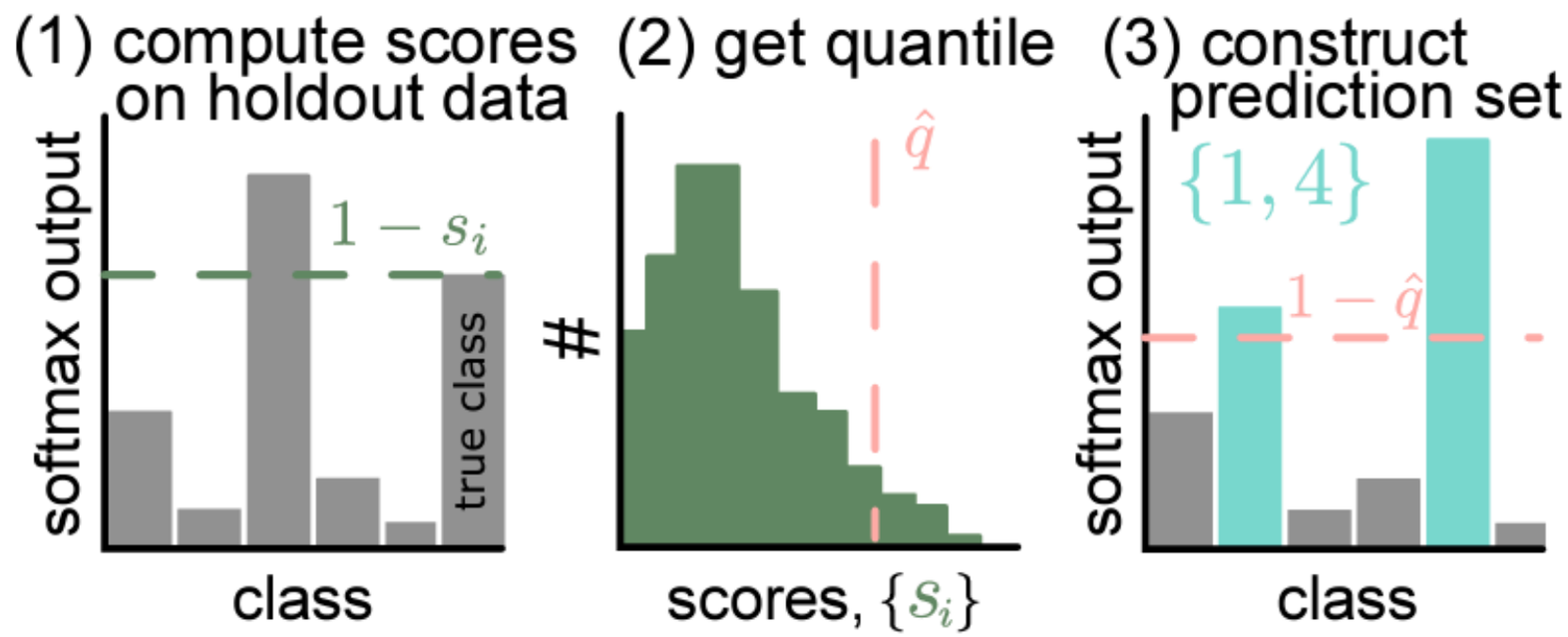


Image from Angelopoulos & Bates, “A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification”, *2022*.

conformal inference: train-time

$p(y = 1 \mid x)$ $p(y = 2 \mid x)$ $p(y = 3 \mid x)$

using validation data, compute the (1-α)-quantile of a conformity statistic:

$$\hat{q}_{1-\alpha}$$

conformal inference: test-time

$p(y = 1 \mid x)$ $p(y = 2 \mid x)$ $p(y = 3 \mid x)$

conformal inference: test-time

$$p(y = 3 \mid x) \quad > \quad p(y = 1 \mid x) \quad > \quad p(y = 2 \mid x)$$

conformal inference: test-time

class #3

class #1

Class #2

$$p(y=3\,|\,x) \quad > \quad p(y=1\,|\,x) \quad > \quad p(y=2\,|\,x)$$

$$C(x) = \left\{ \qquad \right\}$$

check if:

$$\sum_{y \in C(x)} p(y\,|\,x) \overset{?}{\geq} \hat{q}_{1-\alpha}$$

conformal inference: test-time

$$p(y = 3 \mid x) \quad > \quad p(y = 1 \mid x) \quad > \quad p(y = 2 \mid x)$$

$$C(x) = \left\{ \quad \right\}$$

check if:

$$\sum_{y \in C(x)} p(y \mid x) \overset{?}{\geq} \hat{q}_{1-\alpha}$$

conformal inference: test-time

$$p(y = 3 \mid x) \quad > \quad p(y = 1 \mid x) \quad > \quad p(y = 2 \mid x)$$

$$C(x) = \{ \quad \} \quad \text{check if:} \quad \sum_{y \in C(x)} p(y \mid x) \geq \hat{q}_{1-\alpha}$$

conformal inference: test-time

$$p(y = 3 \mid x) \quad > \quad p(y = 1 \mid x) \quad > \quad p(y = 2 \mid x)$$

$$C(x) = \left\{ \text{  }, \text{  } \right\}$$

check if:

$$\sum_{y \in C(x)} p(y \mid x) \overset{?}{\geq} \hat{q}_{1-\alpha}$$

conformal inference: test-time

$$p(y = 3 \mid x) \quad > \quad p(y = 1 \mid x) \quad > \quad p(y = 2 \mid x)$$

$$C(x) = \left\{ \text{  }, \text{  } \right\}$$

check if:

$$\sum_{y \in C(x)} p(y \mid x) \geq \hat{q}_{1-\alpha}$$

conformal inference: test-time

$$C(x) = \left\{ \left(\text{  }, \text{  } \right) \right\}$$

$$y = 3 \qquad , \qquad y = 1$$

conformal inference: regression

Conformal Inference

Traditional Approach

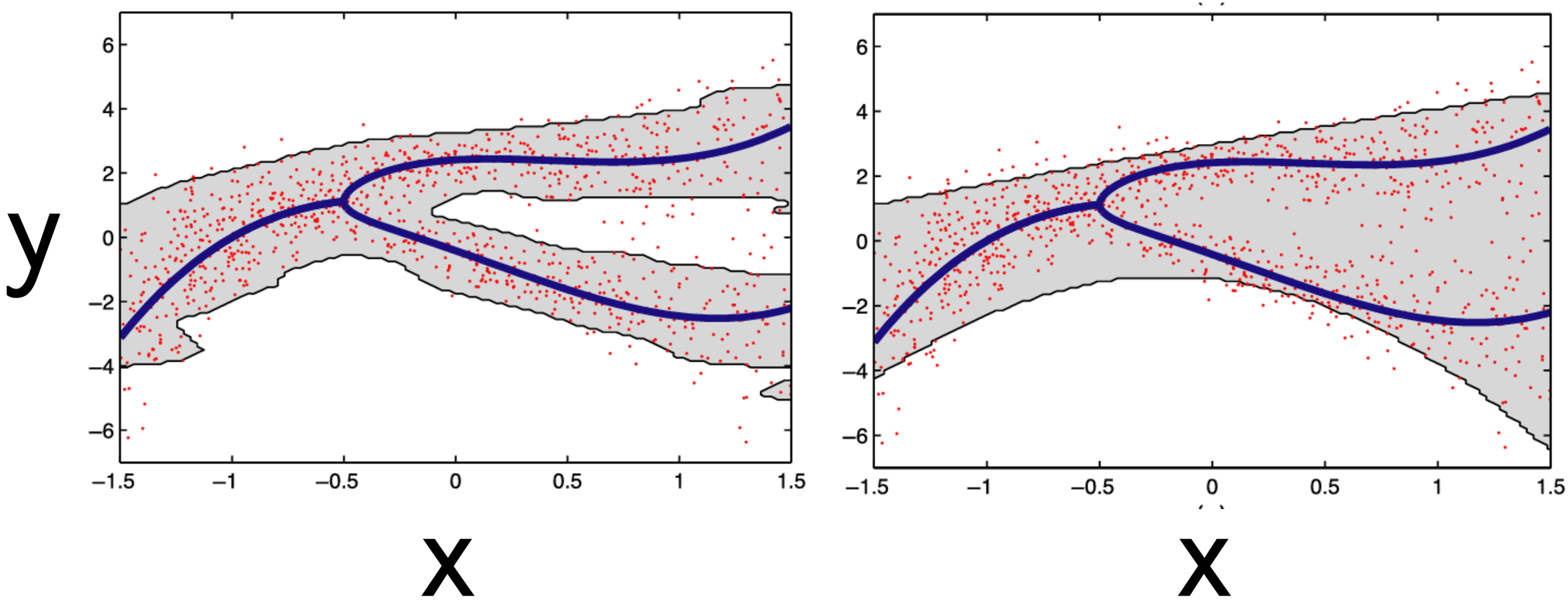


Image from Lei & Wasserman, “Distribution-free prediction bands for nonparametric regression”, *2012*.

A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification

Anastasios N. Angelopoulos and Stephen Bates

December 8, 2022

Abstract

Black-box machine learning models are now routinely used in high-risk settings, like medical diagnostics, which demand uncertainty quantification to avoid consequential model failures. Conformal prediction (a.k.a. conformal inference) is a user-friendly paradigm for creating statistically rigorous uncertainty sets/intervals for the predictions of such models. Critically, the sets are valid in a *distribution-free* sense: they possess explicit, non-asymptotic guarantees even without distributional assumptions or model assumptions. One can use conformal prediction with any pre-trained model, such as a neural network, to produce sets that are guaranteed to contain the ground truth with a user-specified probability, such as 90%. It is easy-to-understand, easy-to-use, and general, applying naturally to problems arising in the fields of computer vision, natural language processing, deep reinforcement learning, and so on.

This hands-on introduction is aimed to provide the reader a working understanding of conformal prediction and related distribution-free uncertainty quantification techniques with one self-contained document. We lead the reader through practical theory for and examples of conformal prediction and describe its extensions to complex machine learning tasks involving structured outputs, distribution shift, time-series, outliers, models that abstain, and more. Throughout, there are many explanatory illustrations, examples, and code samples in Python. With each code sample comes a Jupyter notebook implementing the method on a real-data example; the notebooks can be accessed and easily run by clicking on the following icons:  .

Summary

- ⊗ Construct uncertainty sets without modification to training
- ⊗ **Marginal** coverage guarantee
- ⊗ In practice, sets can often be wide / large