CMSC 491/691 Robust Machine Learning

Topic 5: Uncertainty & Calibration

Slides adopted from Balaji Lakshminarayanan

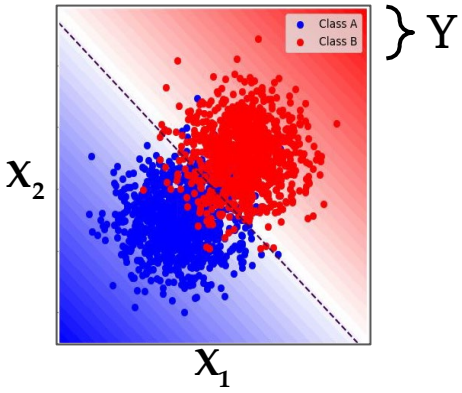
UMBC

What do we mean by Uncertainty?

Return a distribution over predictions rather than a single prediction.

- ***Classification***: Output label along with its confidence.
- ***Regression***: Output mean along with its variance.

Good uncertainty estimates quantify ***when we can trust the model's predictions***.



$$p(\mathbf{y}|\mathbf{x})$$

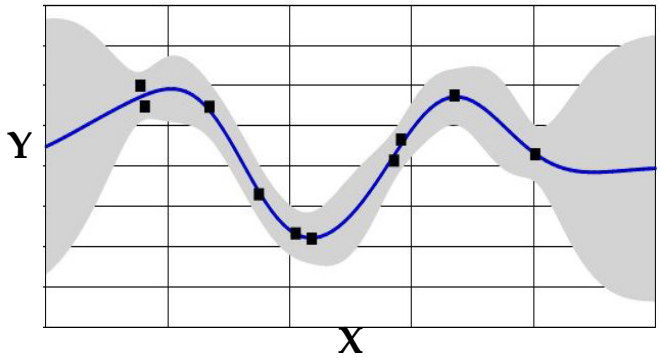


Image credit: Eric Nalisnick

What do we mean by Out-of-Distribution Robustness?

I.I.D. $p_{TEST}(y,x) = p_{TRAIN}(y,x)$

(*Independent and Identically Distributed*)

O.O.D. $p_{TEST}(y,x) \neq p_{TRAIN}(y,x)$

Examples of dataset shift:

- ***Covariate shift.*** Distribution of features p(x) changes and p(y|x) is fixed.
- ***Open-set recognition.*** New classes may appear at test time.
- ***Subpopulation shift.*** Frequencies of data subpopulations changes.
- ***Label shift.*** Distribution of labels p(y) changes and p(x|y) is fixed.

ImageNet-C: Varying Intensity for Dataset Shift

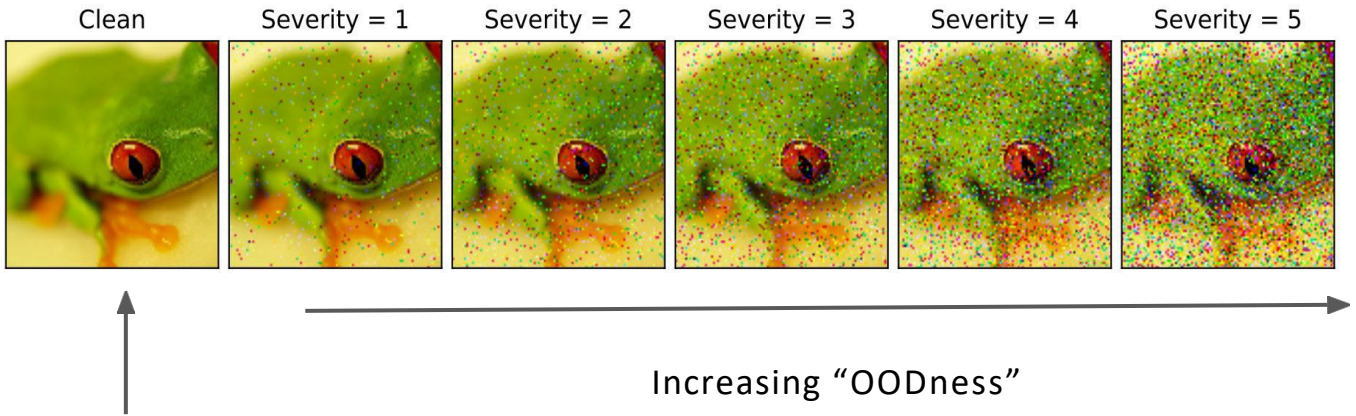
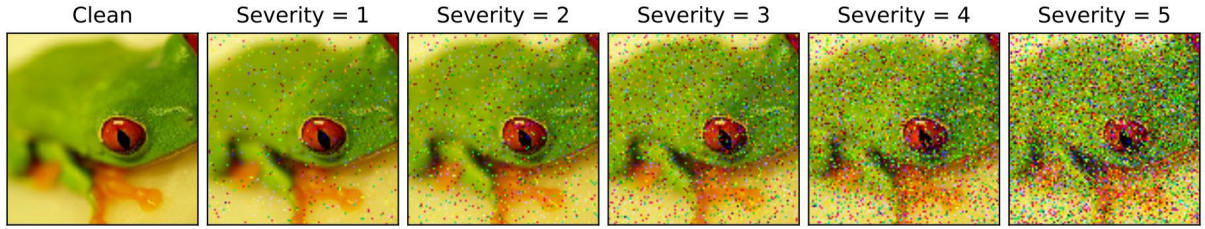
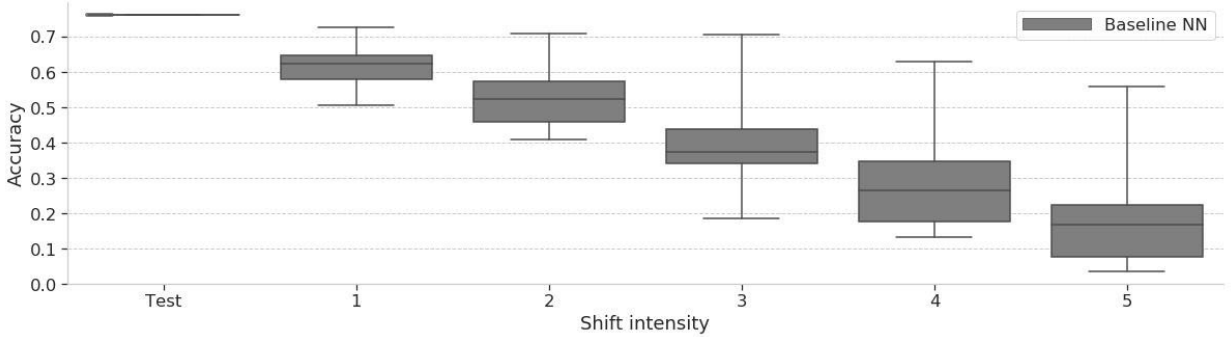


Image source: Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, Hendrycks & Dietterich, 2019.

Neural networks do not generalize under covariate shift



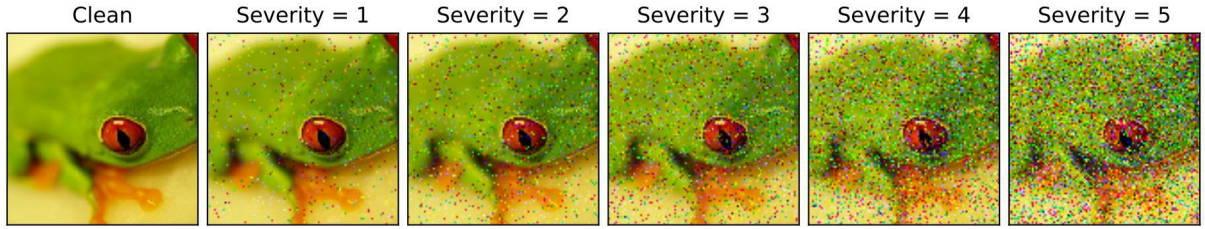
- **Accuracy drops** with increasing shift on Imagenet-C



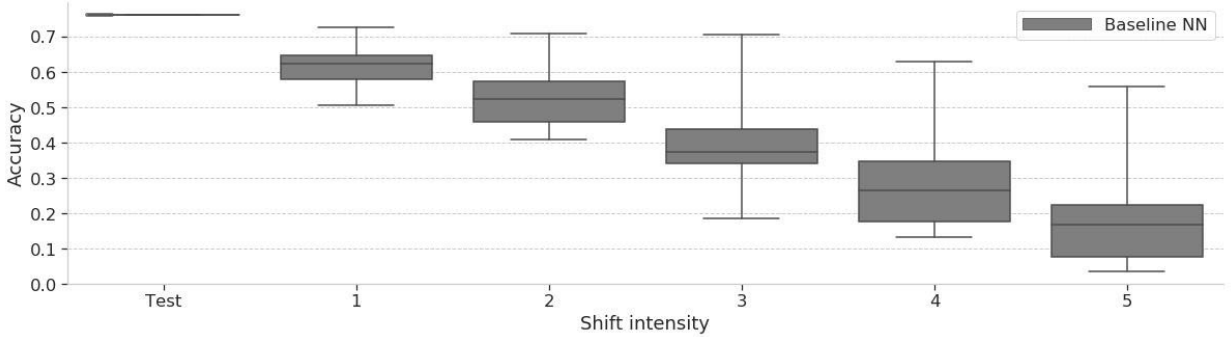
- But do the models know that they are less accurate?

Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift?, Ovadia et al. 2019

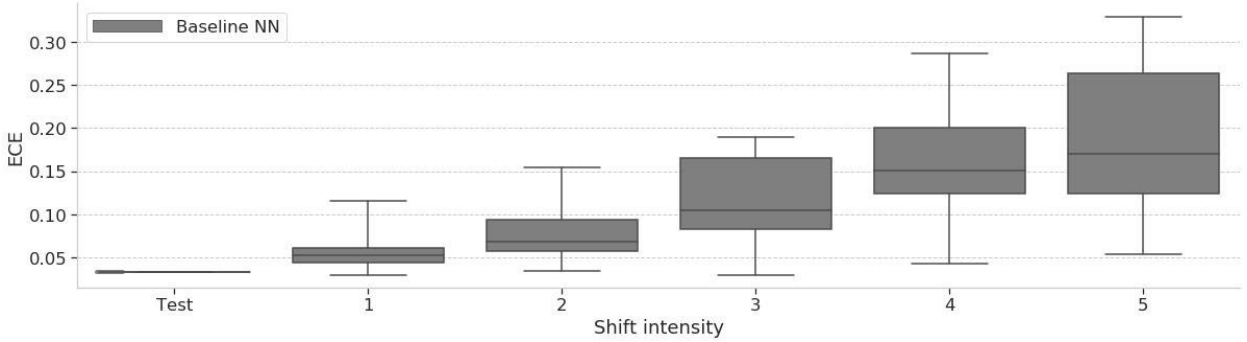
Neural networks *do not know when they don't know*



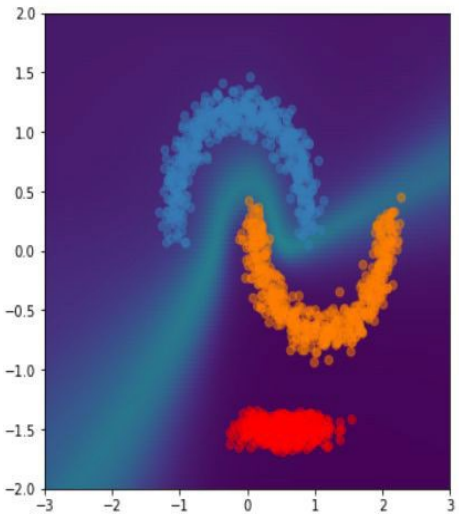
- **Accuracy drops** with increasing shift on Imagenet-C



- **Quality of uncertainty degrades with shift**
  -> "overconfident mistakes"



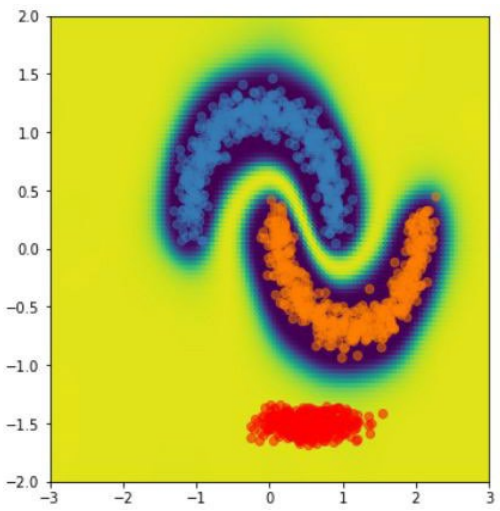
Models assign high confidence predictions to OOD inputs



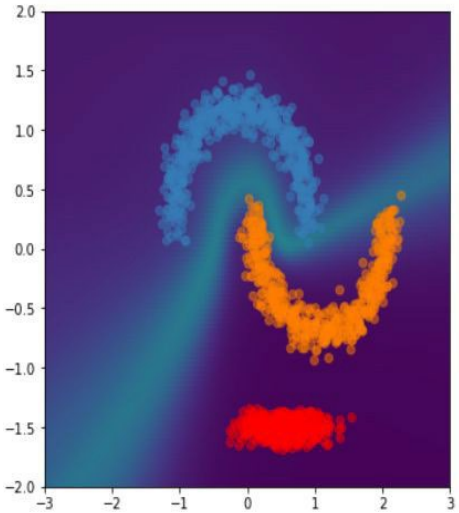
Deep neural networks

Image source: “Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness” Liu et al. 2020

Models assign high confidence predictions to OOD inputs



Ideal behavior



Deep neural networks

High uncertainty
(low confidence)

Low uncertainty
(high confidence)

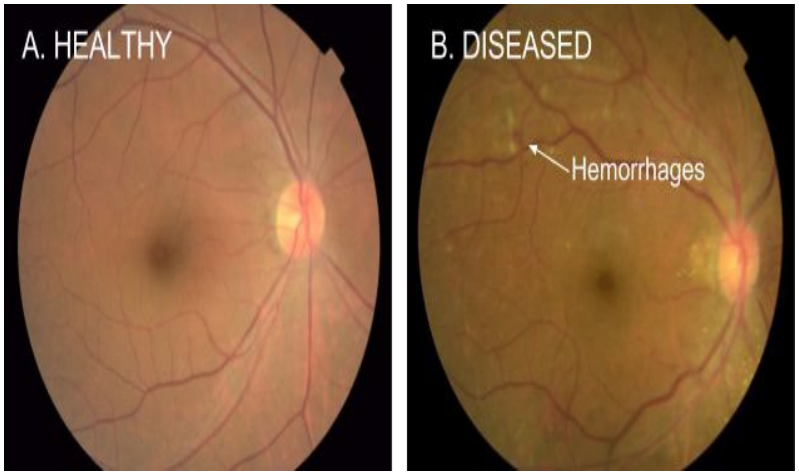
Trust model when x* is close to $p_{TRAIN}(x,y)$

Image source: "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness" Liu et al. 2020

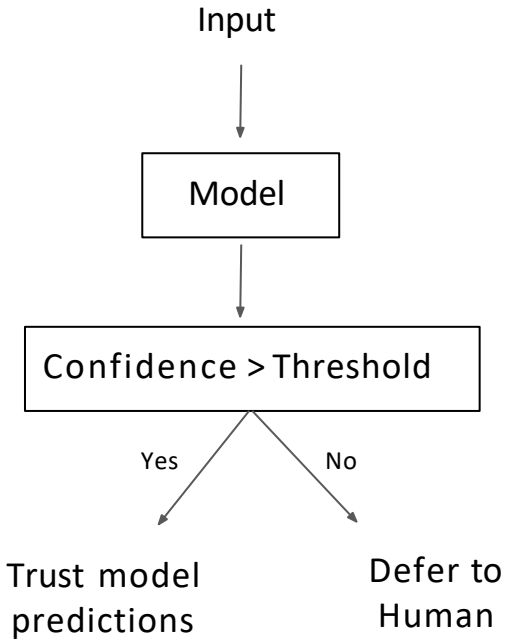
Applications

Healthcare

- Use model uncertainty to decide when to trust the model or to defer to a human.
- Cost-sensitive decision making



Diabetic retinopathy detection from fundus images
Gulshan et al, 2016

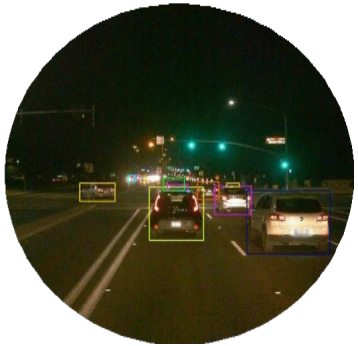


Self-driving cars

Dataset shift:

- Time of day / Lighting
- Geographical location (City vs suburban)
- Changing conditions (Weather / Construction)

Daylight



Night

Weather

Construction

Downtown

Suburban

Image credit: Sun et al, Waymo Open Dataset

Active Learning

- Use model uncertainty to improve data efficiency and model performance in blindspots

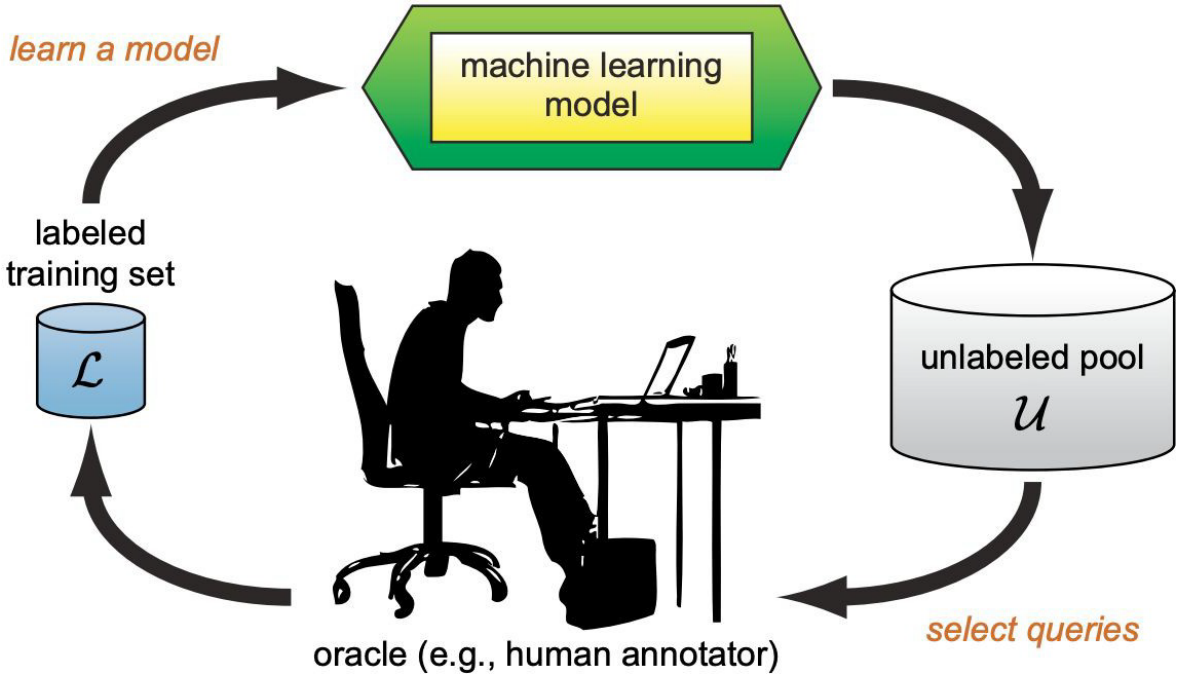


Image source: Active Learning Literature Survey, Settles 2010

Bandits and Reinforcement Learning

- Decision making with asymmetric losses

$$\ell(\mu) \neq \mathbb{E}_{z \sim N(\mu, \sigma^2)}[\ell(z)]$$

- Modeling uncertainty is crucial for **exploration vs exploitation** trade-off

- Non-stationarity

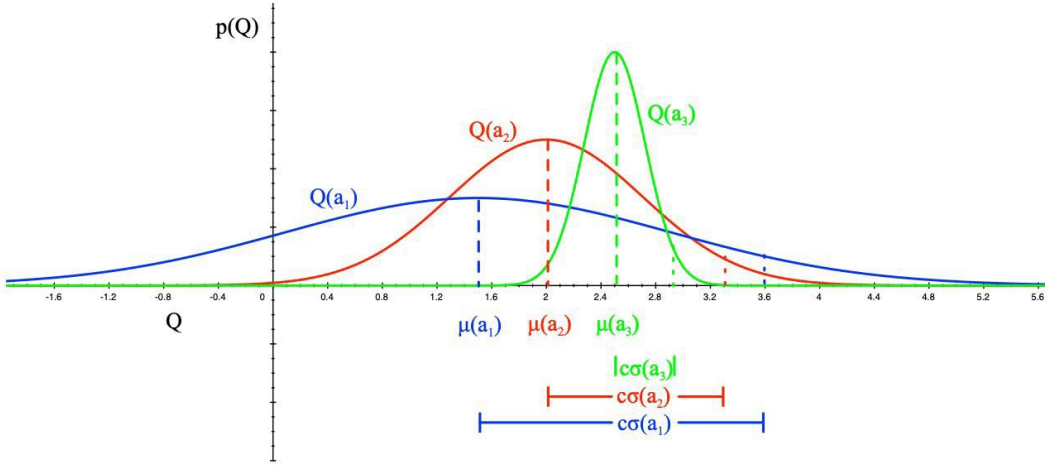
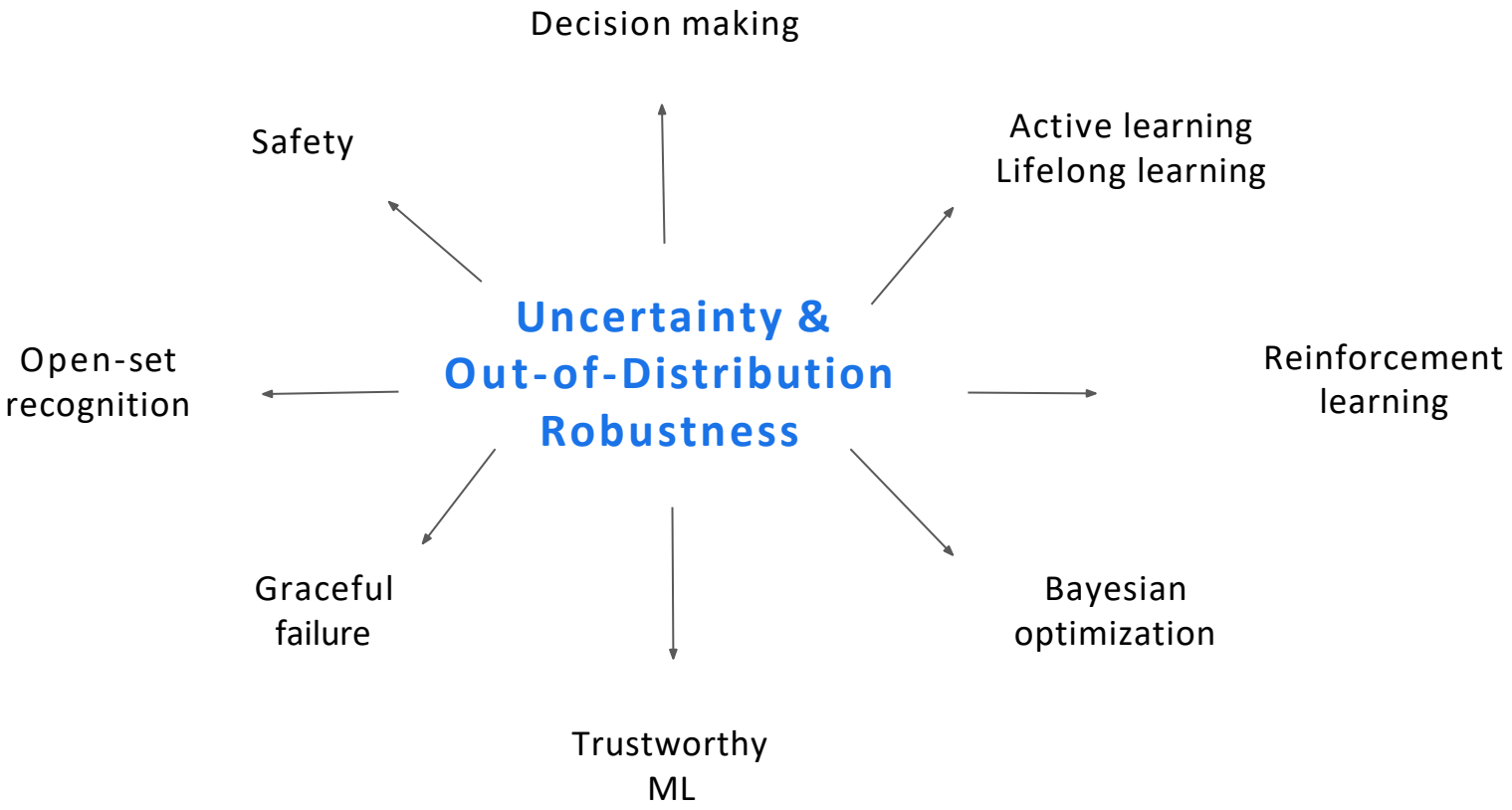


Image source: David Silver's RL course

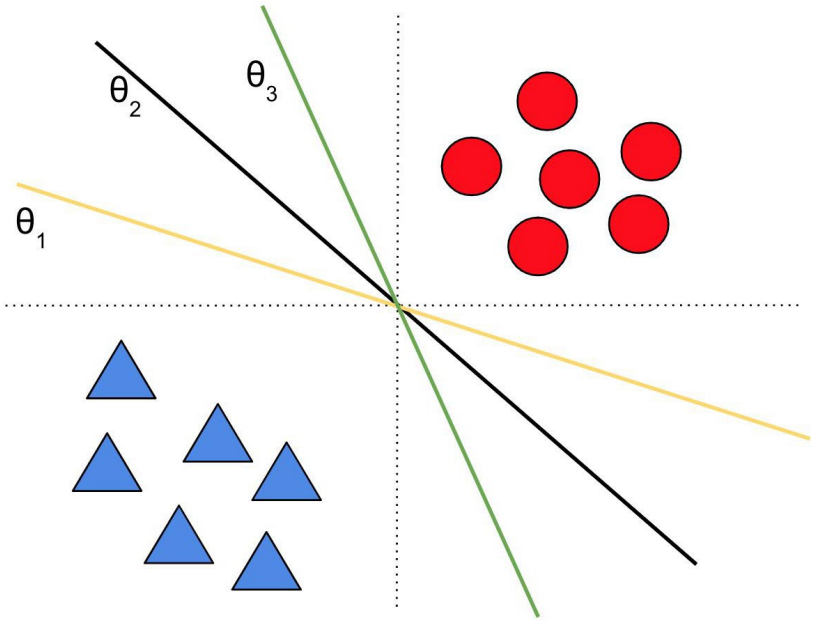
All models are wrong, but ~~some~~ models that know when they are wrong, are useful.



Primer on Uncertainty & Robustness

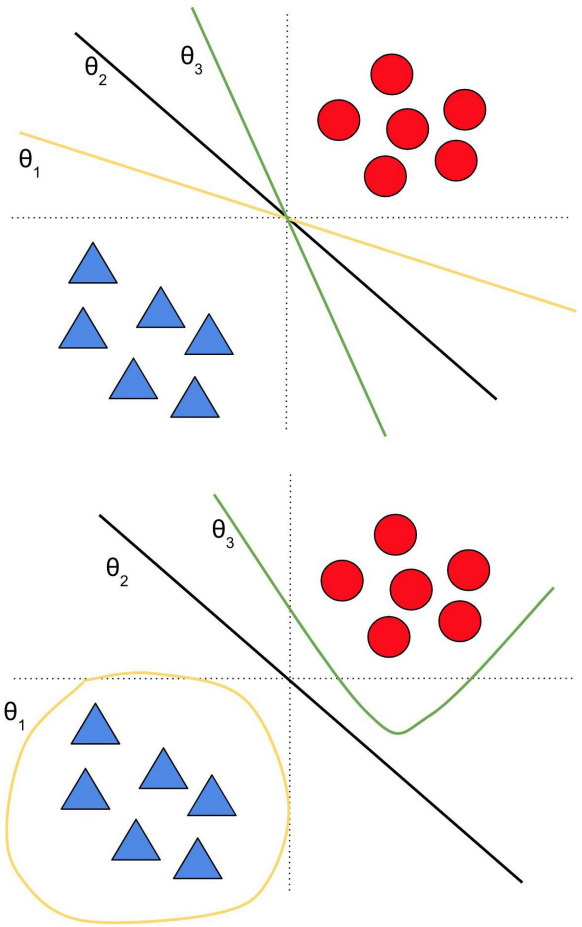
Sources of uncertainty: *Model uncertainty*

- Many models can fit the training data well
- Also known as ***epistemic uncertainty***
- Model uncertainty is "**reducible**"
  - Vanishes in the limit of infinite data (subject to model identifiability)



Sources of uncertainty: *Model uncertainty*

- Many models can fit the training data well
- Also known as ***epistemic uncertainty***
- Model uncertainty is "**reducible**"
  - Vanishes in the limit of infinite data (subject to model identifiability)
- Models can be from same hypotheses class (e.g. linear classifiers in top figure) or belong to different hypotheses classes (bottom figure).



Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)

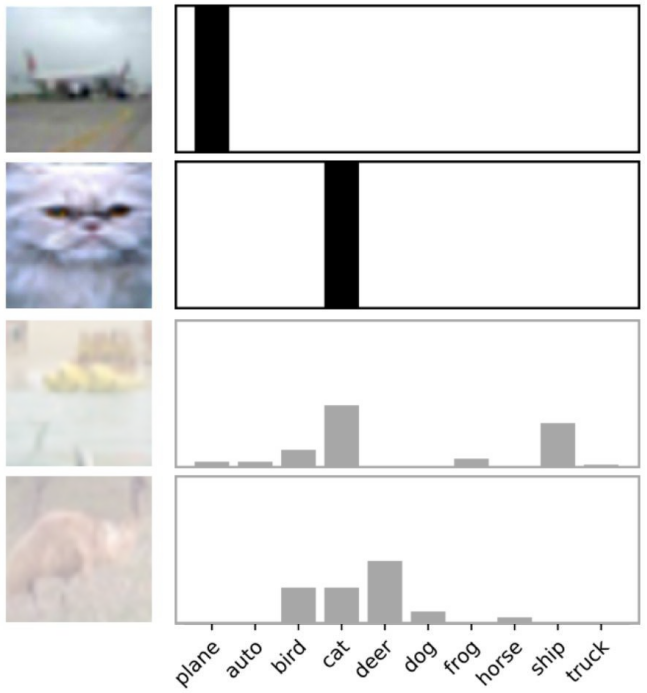


Image source: Battleday et al. 2019 “Improving machine classification using human uncertainty measurements”

Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)

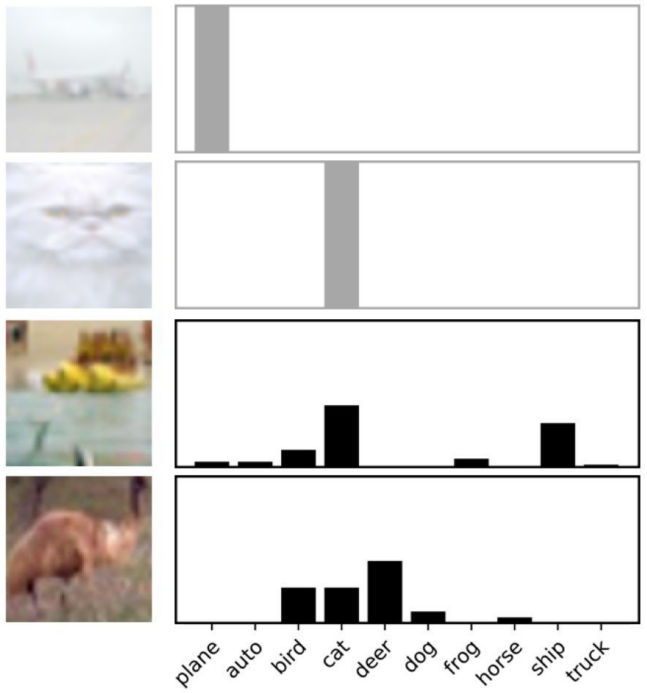


Image source: Battleday et al. 2019 “Improving machine classification using human uncertainty measurements”

Sources of uncertainty: *Data uncertainty*

- Labeling noise (ex: human disagreement)
- Measurement noise (ex: imprecise tools)
- *Missing* data (ex: partially observed features, unobserved confounders)
- Also known as ***aleatoric uncertainty***
- Data uncertainty is "**irreducible***"
  - Persists even in the limit of infinite data
  - *Could be reduced with additional features/views

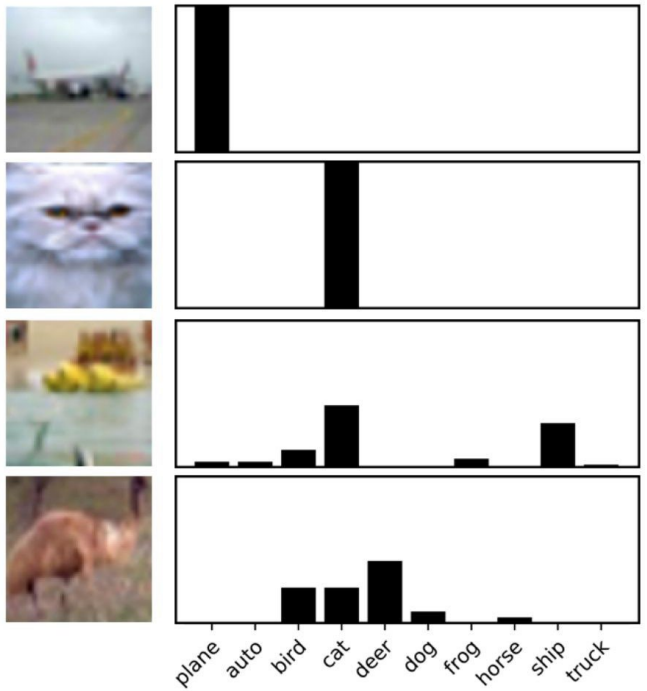


Image source: Battleday et al. 2019 "Improving machine classification using human uncertainty measurements"

How do we measure the quality of uncertainty?

$$\text{Calibration Error} = \left| \text{Confidence} - \text{Accuracy} \right|$$

predicted probability of correctness

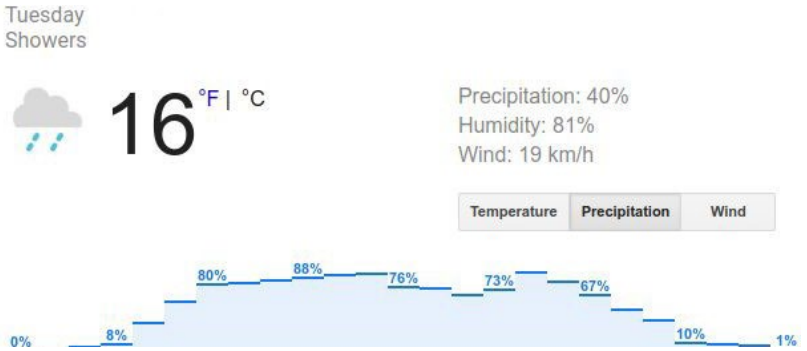
observed frequency of correctness

How do we measure the quality of uncertainty?

Calibration Error = |Confidence - Accuracy|

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?

- 80% implies perfect calibration
- Less than 80% implies model is overconfident
- Greater than 80% implies model is under-confident

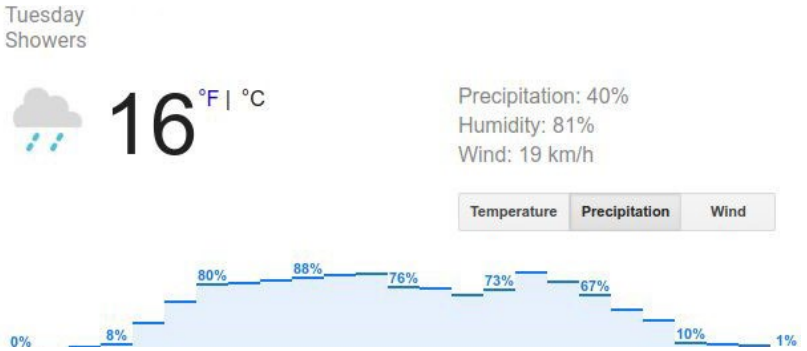


How do we measure the quality of uncertainty?

$$\text{Calibration Error} = |\text{Confidence} - \text{Accuracy}|$$

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?

- 80% implies perfect calibration
- Less than 80% implies model is overconfident
- Greater than 80% implies model is under-confident



Intuition: For regression, calibration corresponds to coverage in a confidence interval.

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

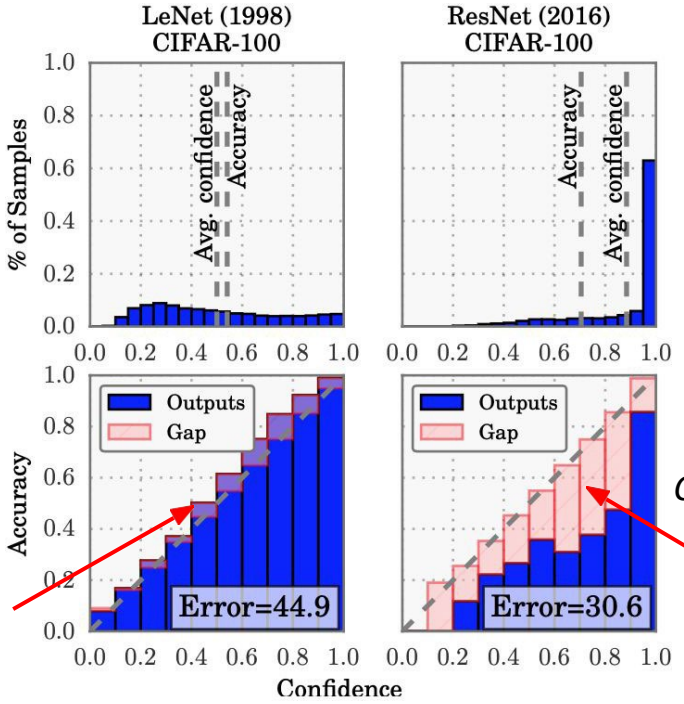
$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$

- Bin the probabilities into B bins.
- Compute the within-bin accuracy and within-bin predicted confidence.
- Average the calibration error across bins (weighted by number of points in each bin).

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$



Confidence < Accuracy

=> Underconfident

Confidence > Accuracy

=> Overconfident

Image source: Guo+ 2017 "On calibration of modern neural networks"

How do we measure the quality of uncertainty?

Expected Calibration Error [Naeini+ 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$

Note: Does **not** reflect **accuracy**.

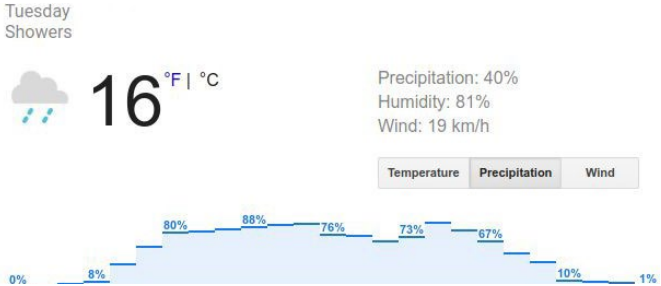
Predicting class frequency p(y=1) = 0.3 for all the inputs achieves perfect calibration.

True label	0	0	0	0	0	0	0	1	1	1	Accurate?	Calibrated?
Model prediction	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	❌	✅

How do we measure the quality of uncertainty?

Proper scoring rules [Gneiting & Raftery 2007]

- Negative Log-Likelihood (NLL)
  - Also known as *cross-entropy*
  - Can overemphasize tail probabilities



- Brier Score
  - Quadratic penalty (bounded range [0,1] unlike log).

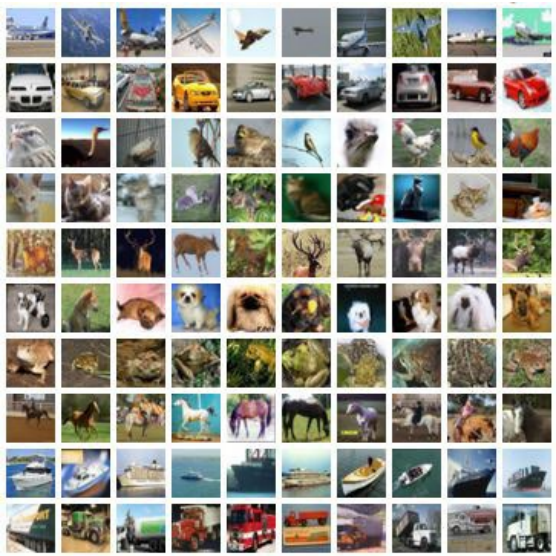
$$\mathrm{BS} = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \left[p(y|\mathbf{x}_n, \theta) - \delta(y - y_n) \right]^2$$

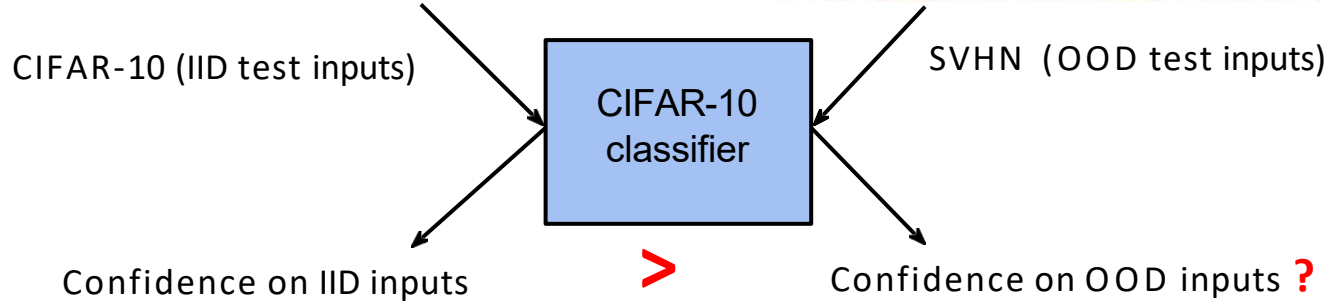
  - Can be numerically unstable to optimize.

How do we measure the quality of uncertainty?

Evaluate model on **out-of-distribution (OOD) inputs** which do not belong to any of the existing classes

- Max confidence
- Entropy of p(y|x)





Downstream cost (unifying accuracy & OOD detection)

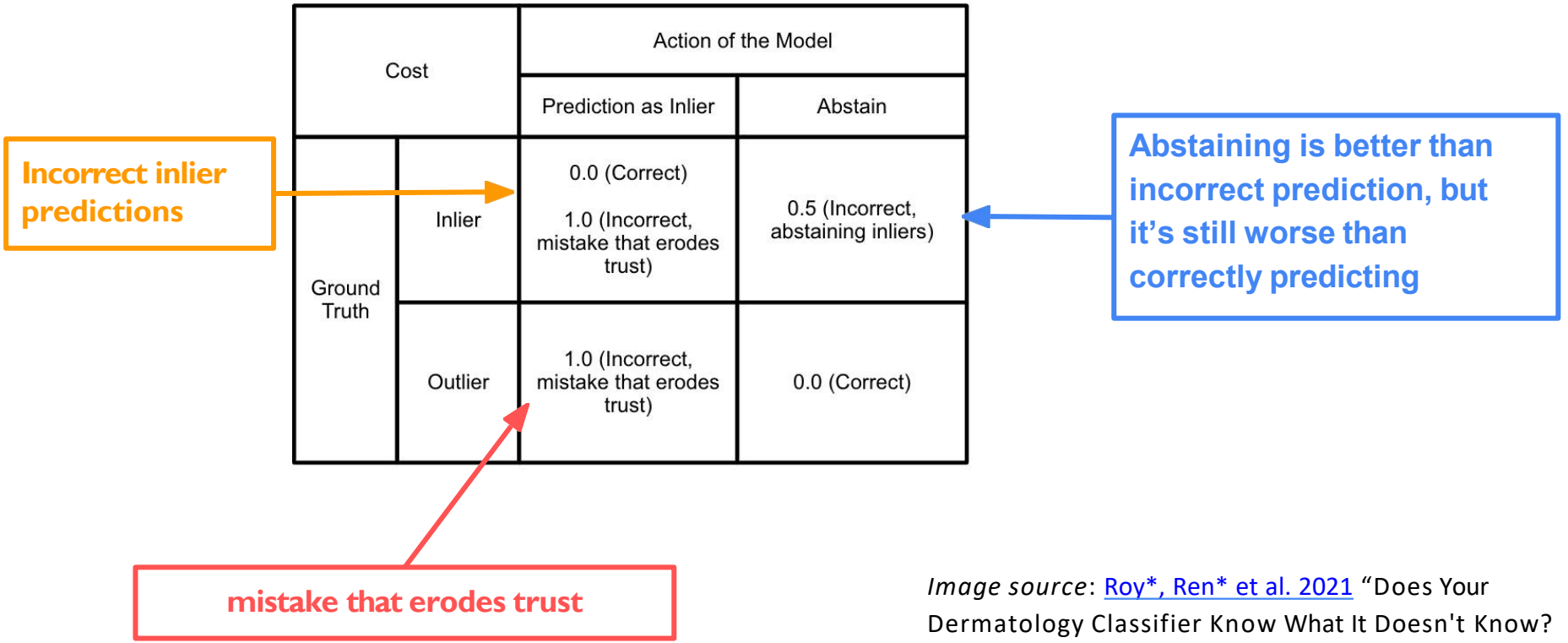


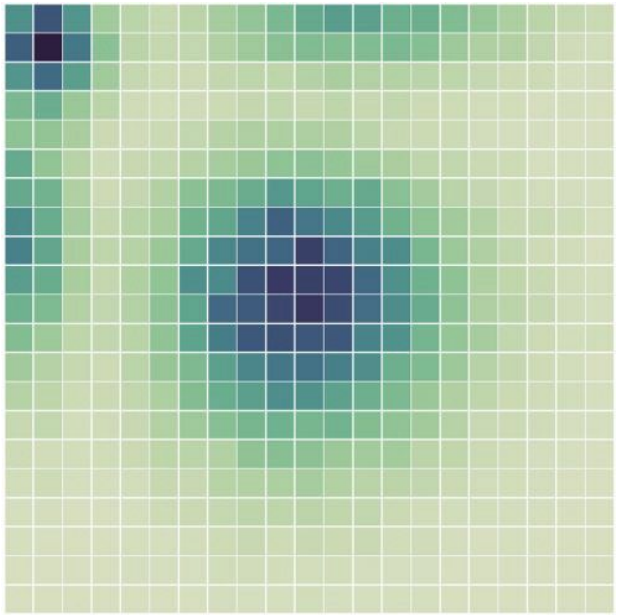
Image source: Roy*, Ren* et al. 2021 "Does Your Dermatology Classifier Know What It Doesn't Know? Detecting the Long-Tail of Unseen Conditions"

Fundamentals to Uncertainty & Robustness Methods

Neural Networks with SGD

Nearly all models find a single setting of parameters to maximize the probability conditioned on data.

$$
\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathbf{x}, \mathbf{y}) \\
&= \arg\min_{\boldsymbol{\theta}} -\log p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \\
&=^* \arg\min_{\boldsymbol{\theta}} \sum_k \mathbf{y}_k \log \mathbf{p}_k + \boldsymbol{\lambda} \|\boldsymbol{\theta}\|^2
\end{aligned}
$$



Special case: softmax cross entropy with L2 regularization. Optimize with SGD!

Image source: Ranganath+ 2016

Neural Networks with SGD

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathbf{x}, \mathbf{y})$$

Problem: results in just one prediction per example
No model uncertainty

How do we get uncertainty?

- Probabilistic approach
  - Estimate a full distribution for $p(\boldsymbol{\theta} \mid \mathbf{x}, \mathbf{y})$
- Intuitive approach: Ensembling
  - Obtain multiple good settings for $\boldsymbol{\theta}^*$

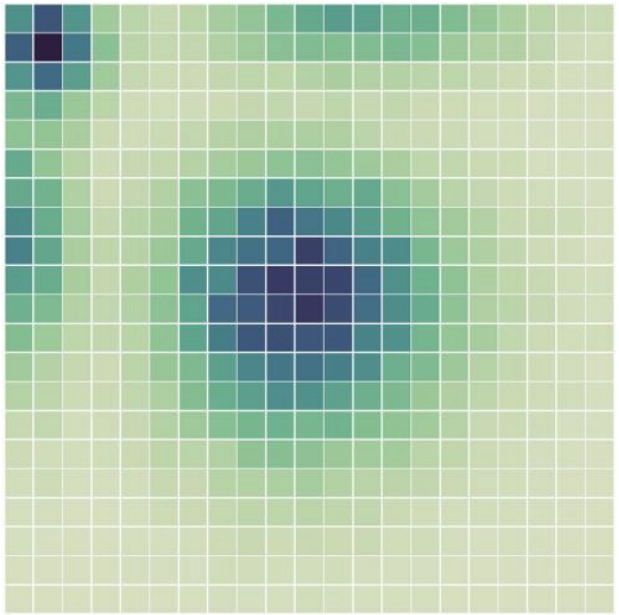


Image source: Ranganath+ 2016

Probabilistic Machine Learning

Model: A probabilistic model is a joint distribution of outputs **y** and parameters $\boldsymbol{\theta}$ given inputs **x**.

$$p(\mathbf{y}, \boldsymbol{\theta} \mid \mathbf{x})$$

Training time: Calculate the Bayesian **posterior**, the conditional distribution of parameters given observations.

$$p(\boldsymbol{\theta} \mid \mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{y}, \boldsymbol{\theta} \mid \mathbf{x})}{p(\mathbf{y} \mid \mathbf{x})} = \frac{p(\mathbf{y} \mid \mathbf{x}) p(\boldsymbol{\theta})}{\int p(\mathbf{y}, \boldsymbol{\theta} \mid \mathbf{x}) \, \mathrm{d}\boldsymbol{\theta}}$$

Prediction time: Compute the likelihood given parameters, each parameter configuration of which is weighted by the posterior.

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \, \mathrm{d}\boldsymbol{\theta} \approx \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^{(s)})$$

[Murphy 2012]

General Recipe

Parametrize “base model” with desired inductive biases.

$$p(\mathbf{y}, \boldsymbol{\theta} \mid \mathbf{x})$$

Specify prior over functions.

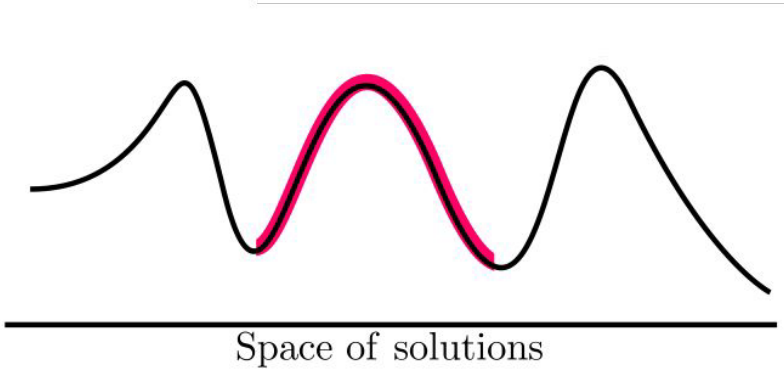
$$p(\boldsymbol{\theta})$$

Capture model uncertainty by approximating the posterior.

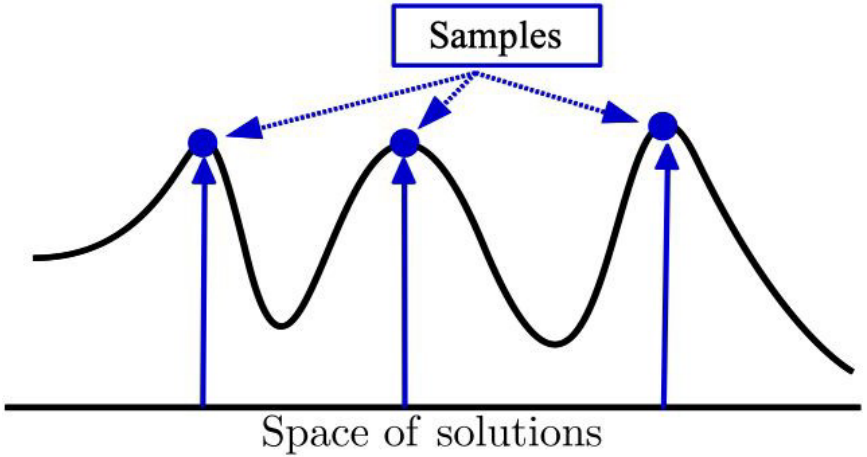
$$p(\boldsymbol{\theta} \mid \mathcal{D})$$

Approximating the posterior

$p(\boldsymbol{\theta} \mid \mathcal{D})$ is multimodal and complex, so how do we estimate and represent it?



Local approximations

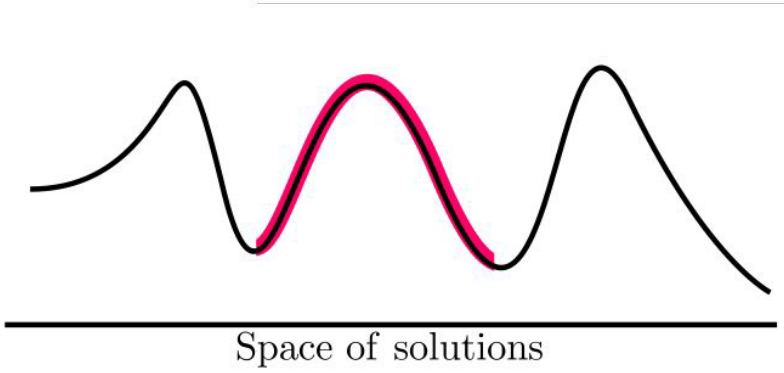


Sampling

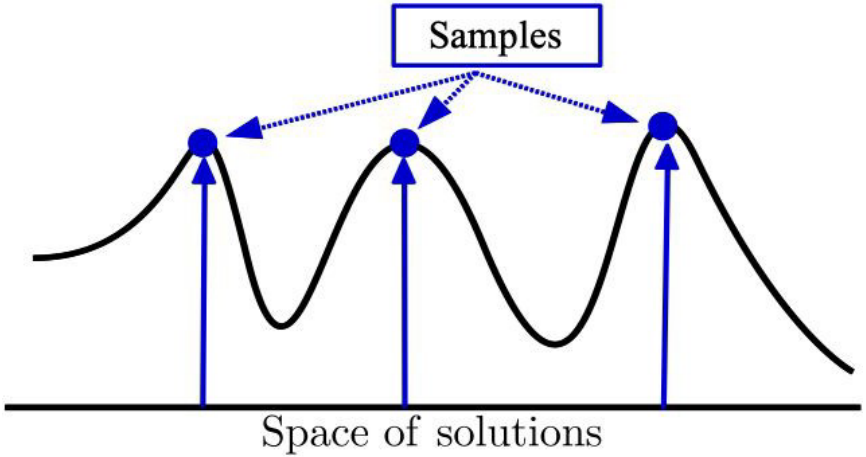
- Locally, covering one mode well
  e.g. with a simpler distribution $q(\boldsymbol{\theta}; \boldsymbol{\lambda})$
  - Variational inference
  - Laplace approximation

Approximating the posterior

$p(\boldsymbol{\theta} \mid \mathcal{D})$ is multimodal and complex, so how do we estimate and represent it?



Local approximations



Sampling

- Summarize using samples
  - MCMC
  - Hamiltonian Monte Carlo
  - Stochastic Gradient Langevin Dynamics

Ensemble Learning

- A prior distribution often involves the complication of approximate inference.
- *Ensemble learning* offers an alternative strategy to aggregate the predictions over a collection of models.
- Often winner of competitions!
- There are two considerations: the collection of models to ensemble; and the aggregation strategy.

Popular approach is to average predictions of independently trained models, forming a mixture distribution.

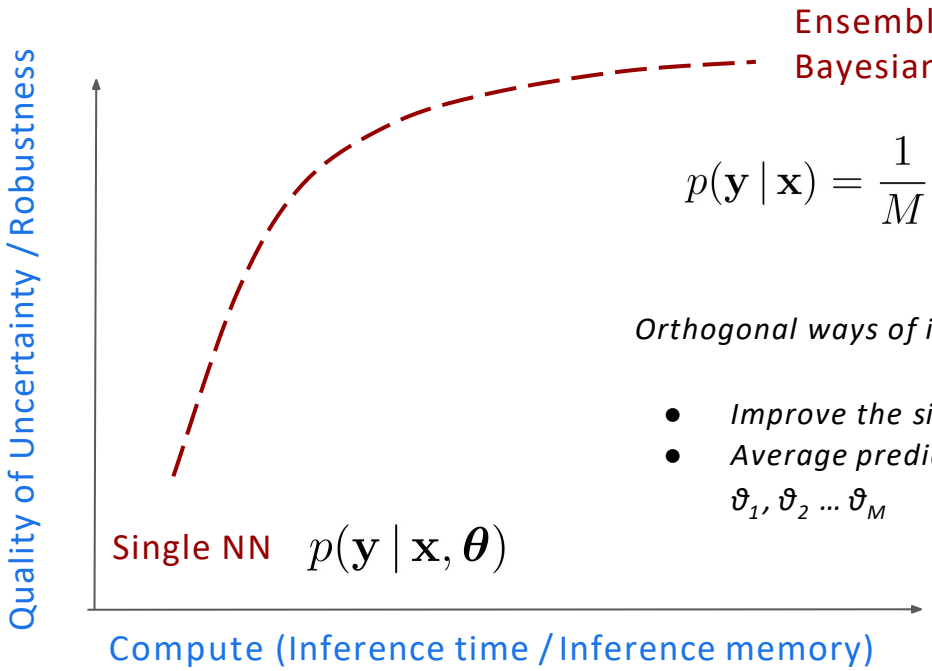
$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_k)$$

Many approaches exist: bagging, boosting, decision trees, stacking.

[Dietterich 2000]

Overview of Methods

Cartoon: Uncertainty/Robustness vs Compute frontier

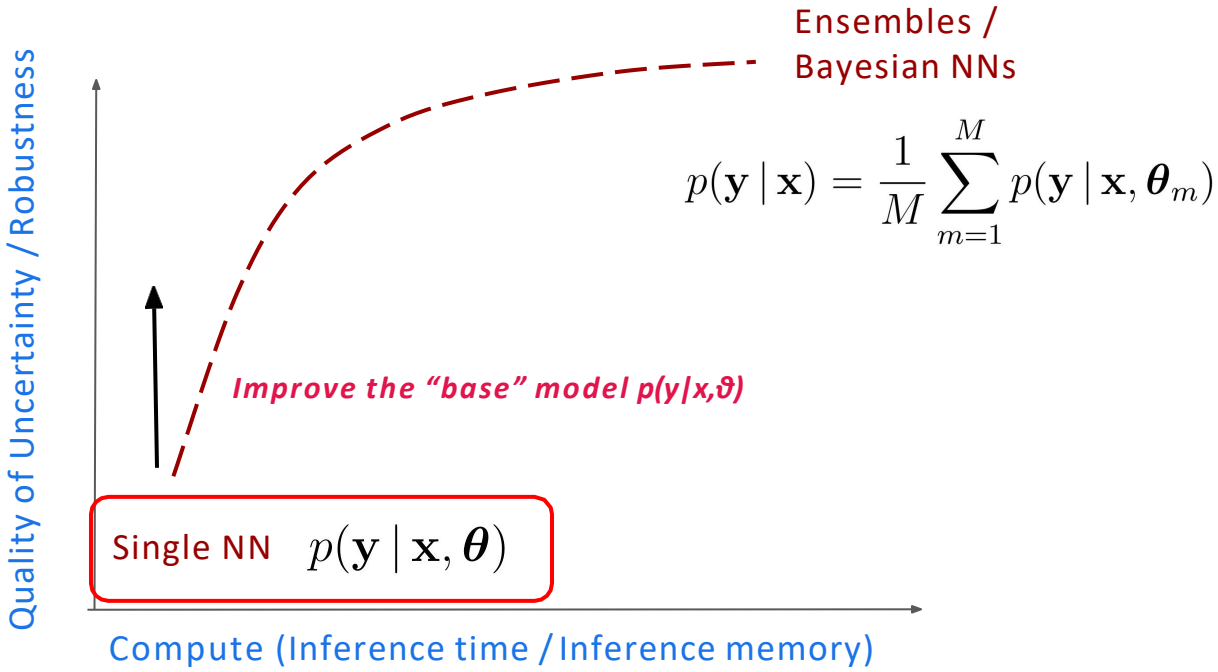


$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_m)$$

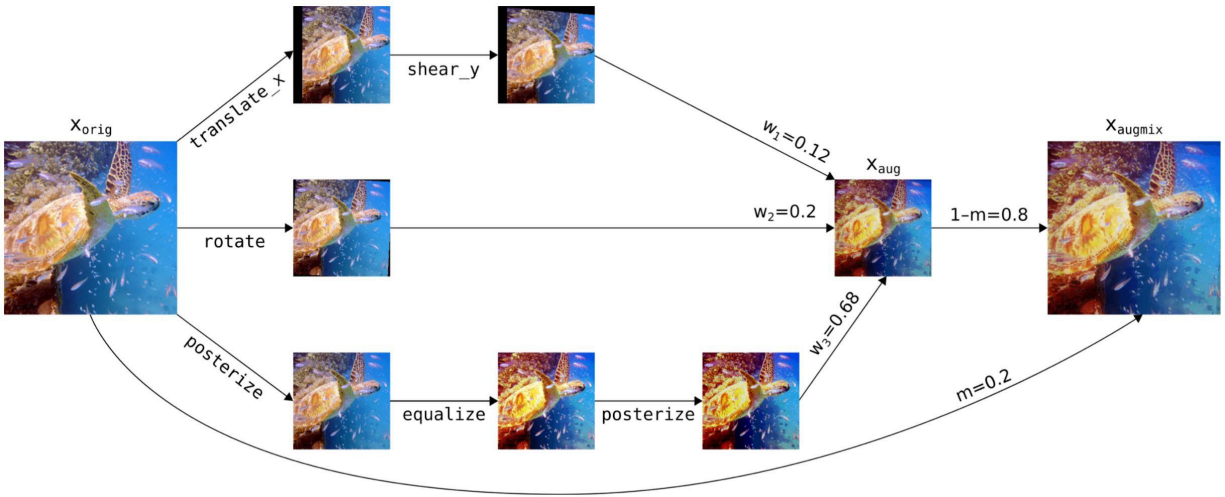
Orthogonal ways of improving performance

- *Improve the single model p(y|x,ϑ)*
- *Average predictions over diverse set of functions* $\vartheta_1, \vartheta_2 \ldots \vartheta_M$

Improving single model performance



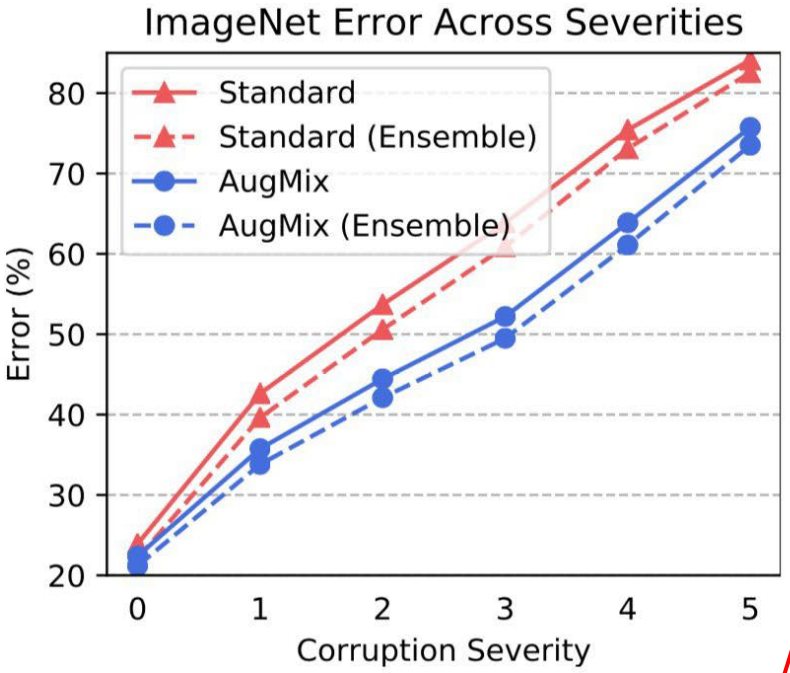
Better representations via data augmentation, e.g. AugMix

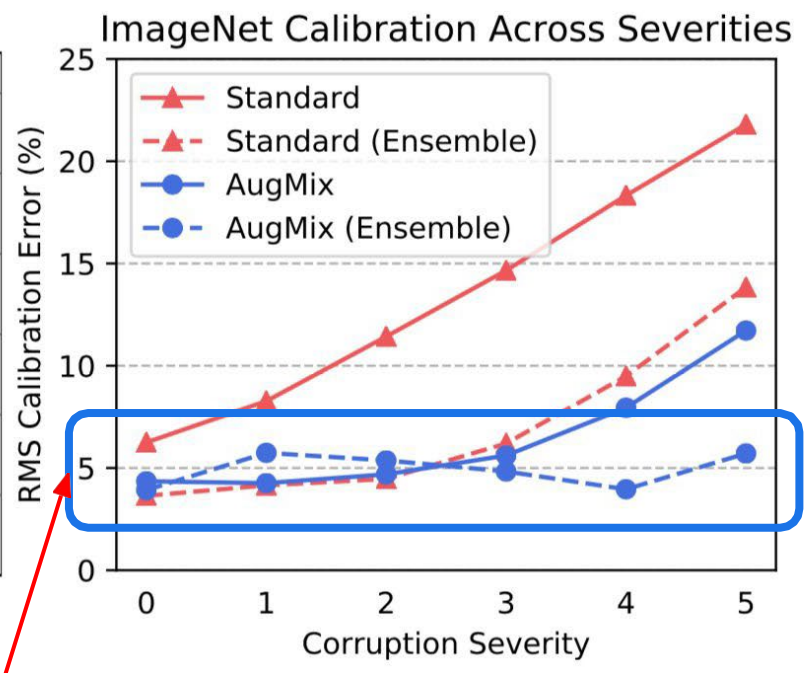


Composing base operations and ‘mixing’ them can improve accuracy and calibration under shift.

[Hendrycks+ 2020]

AugMix improves accuracy & calibration under shift

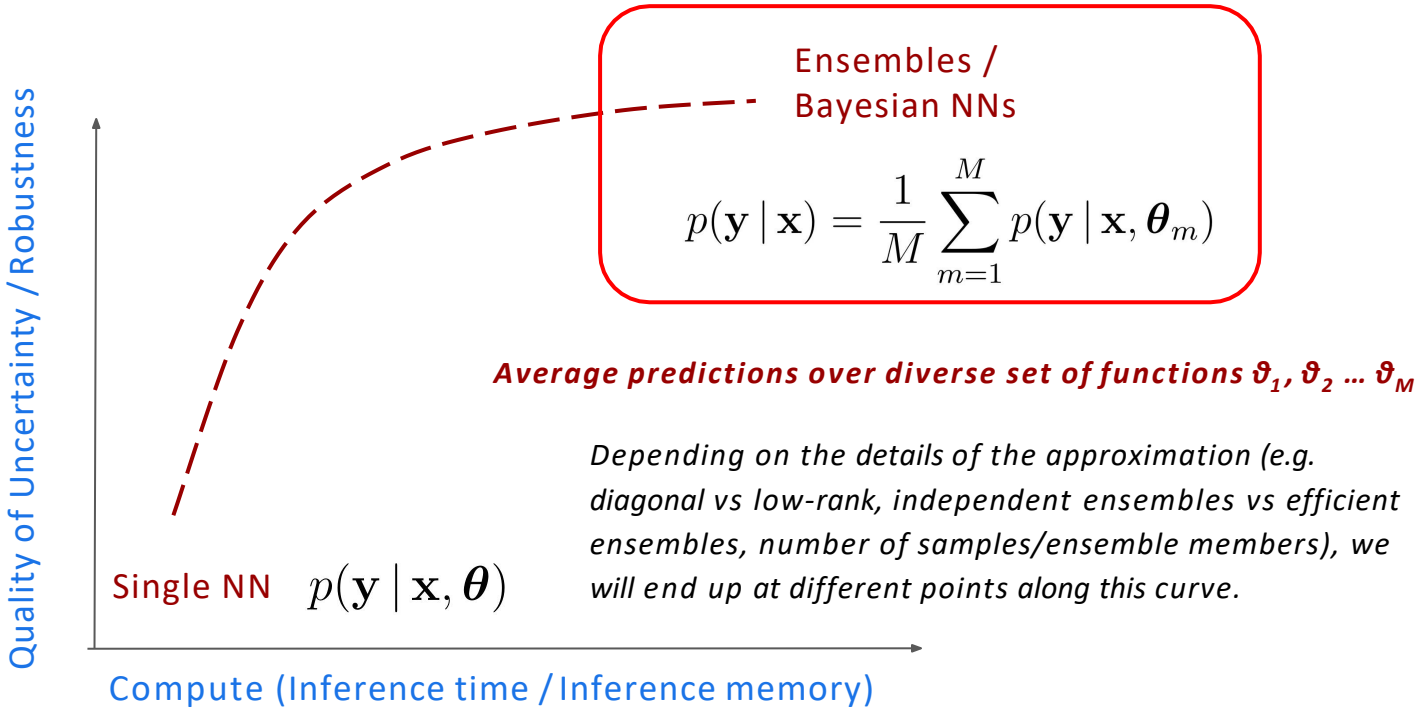




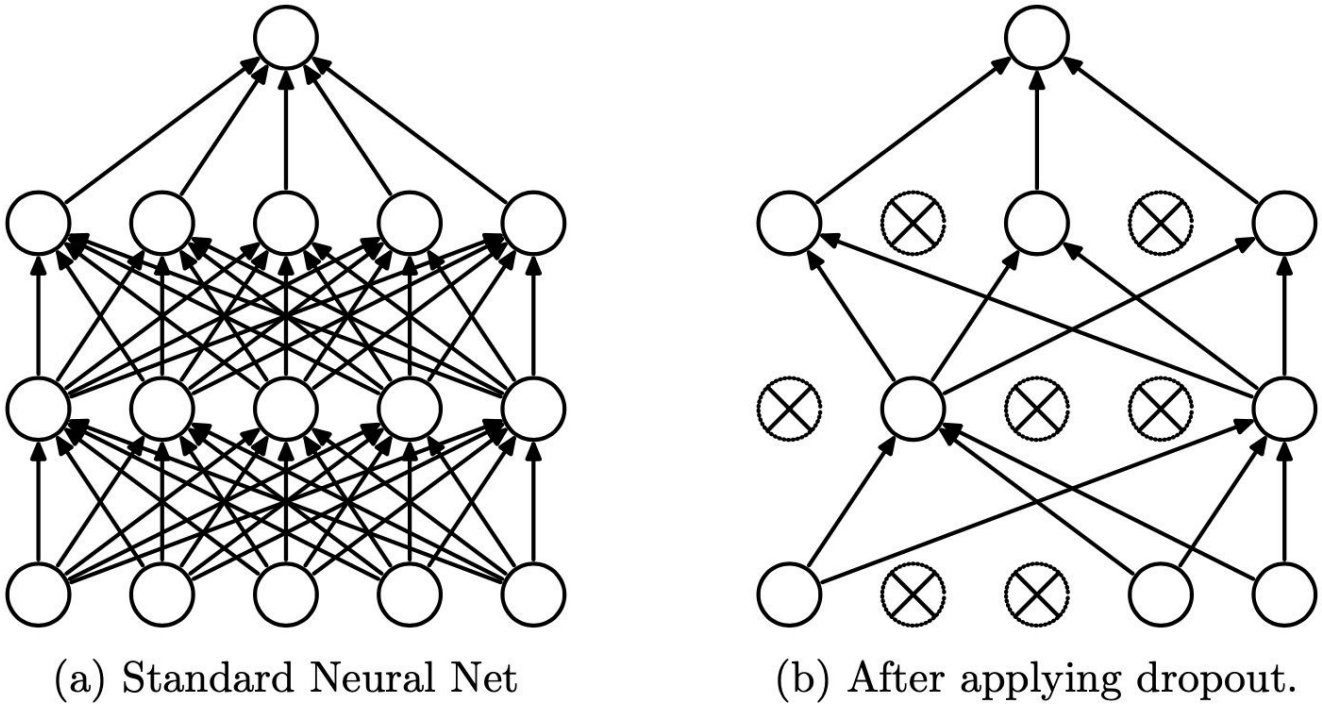
Data augmentation can provide complementary benefits to ensembling.

[Hendrycks+ 2020]

Improving the quality of model uncertainty



Simple Baseline: Monte Carlo Dropout



(a) Standard Neural Net

(b) After applying dropout.

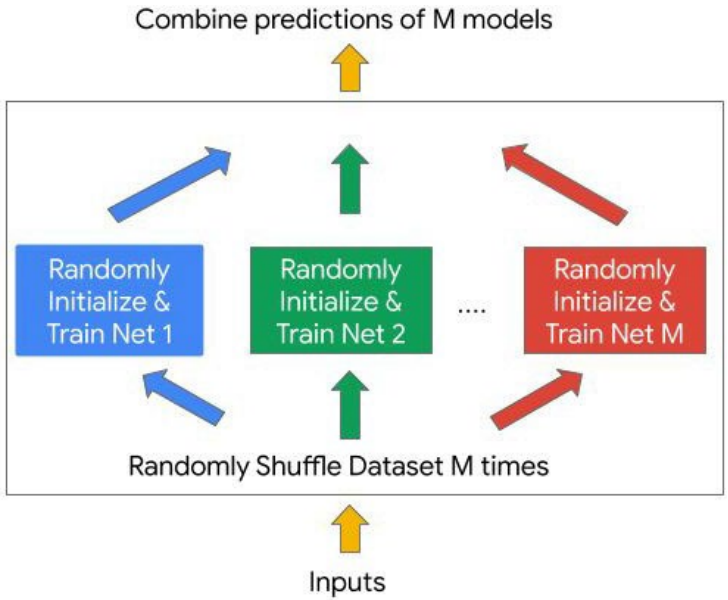
Image source: Dropout: A Simple Way to Prevent Neural Networks from Overfitting

[Gal+ 2015]

Simple Baseline: Deep Ensembles

Idea: Just re-run standard SGD training but with different random seeds and average the predictions.

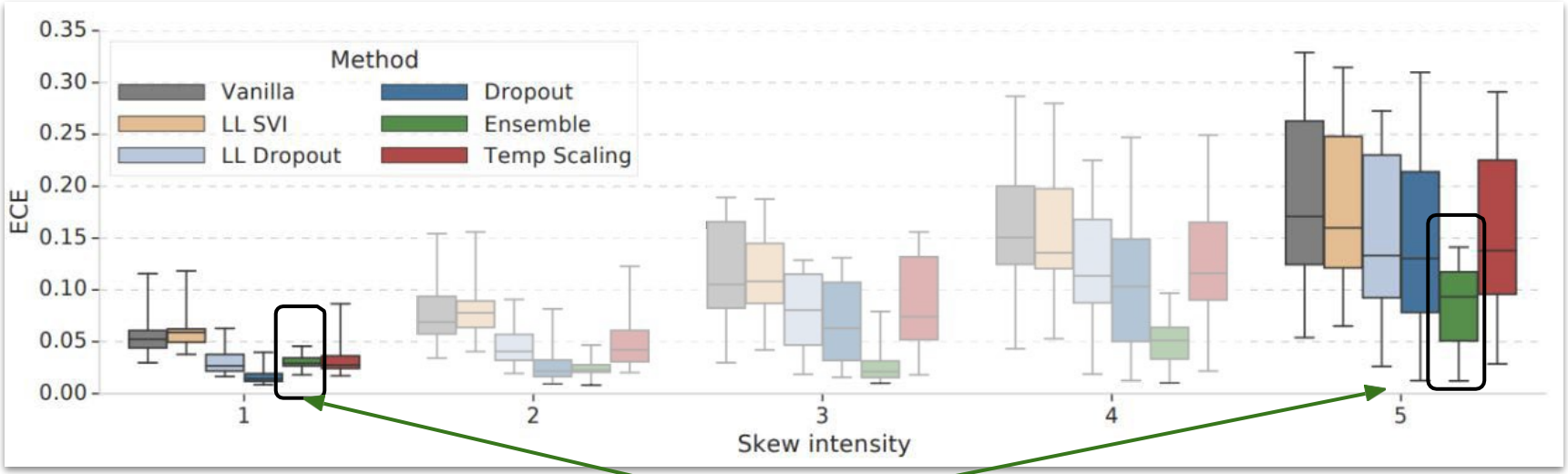
Deep ensembles can capture different modes in function space.



[Lakshminarayanan+ 2016]

[Fort+ 2019]

Deep Ensembles work surprisingly well in practice



Deep Ensembles are consistently among the best performing methods, especially under dataset shift

Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift [Ovadia+ 2019]

Conformal Prediction

Allow classifier to output “prediction sets” rather than single predictions.

Decide how many classes to include in “prediction set” based on a threshold α

Formally, for a discrete response $Y \in \mathcal{Y} = \{1, \ldots, K\}$ and a feature vector $X \in \mathbb{R}^d$, we desire an uncertainty set function, $\mathcal{C}(X)$, mapping a feature vector to a subset of $\{1, \ldots, K\}$ such that

$$P(Y \in \mathcal{C}(X)) \geq 1 - \alpha, \tag{1}$$

Takeaways

- Uncertainty & robustness are critical problems in AI and machine learning.
- Benchmark models with calibration error and a large collection of OOD shifts.
- Probabilistic ML, ensemble learning, and optimization provide a foundation.
- The best methods advance two dimensions:
  - combining multiple neural network predictions; and
  - imposing priors and inductive biases.

Check out recent ICML workshop on Uncertainty and Robustness in Deep Learning