Lecture 11: Calibrated Prediction

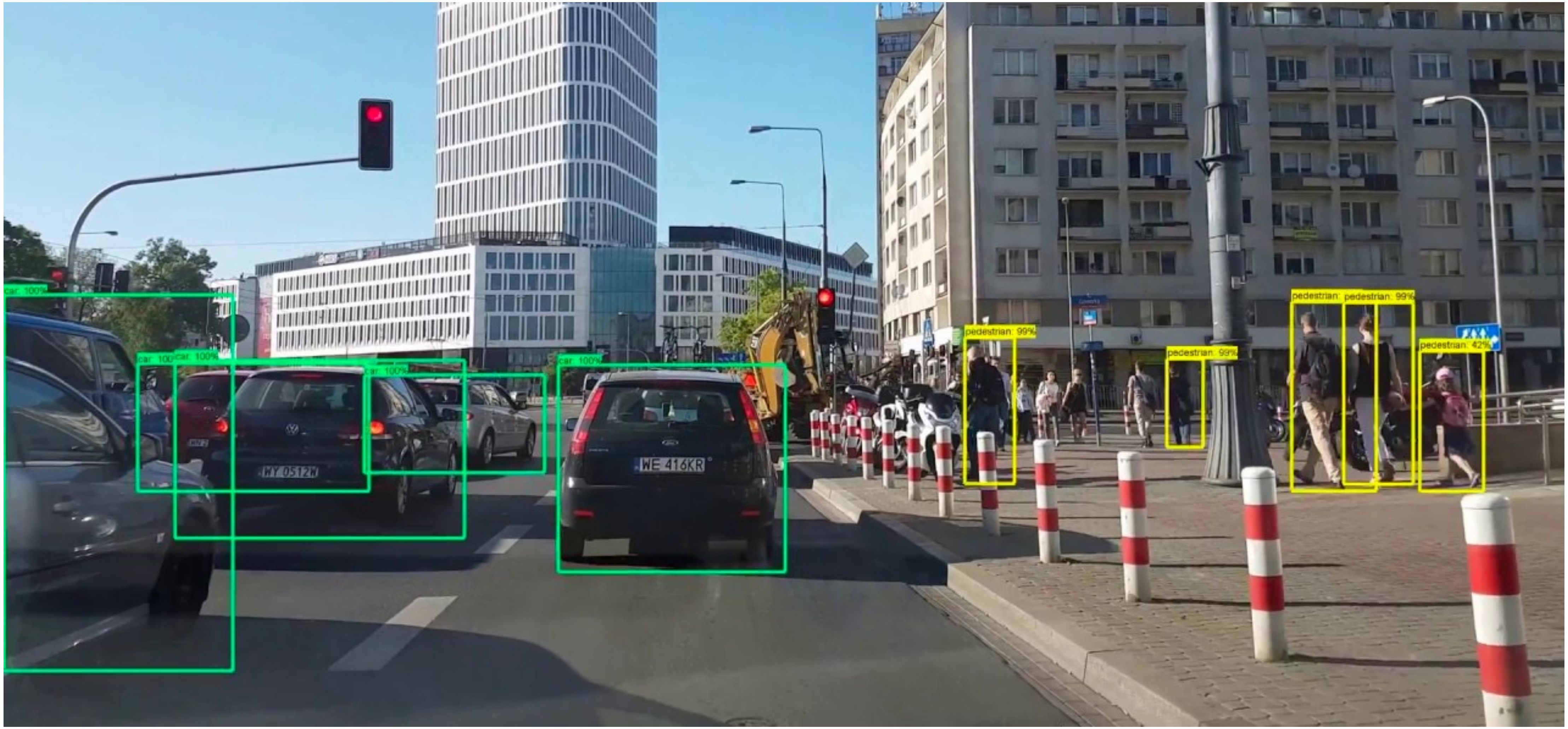
CIS 7000: Trustworthy Machine Learning

Spring 2024

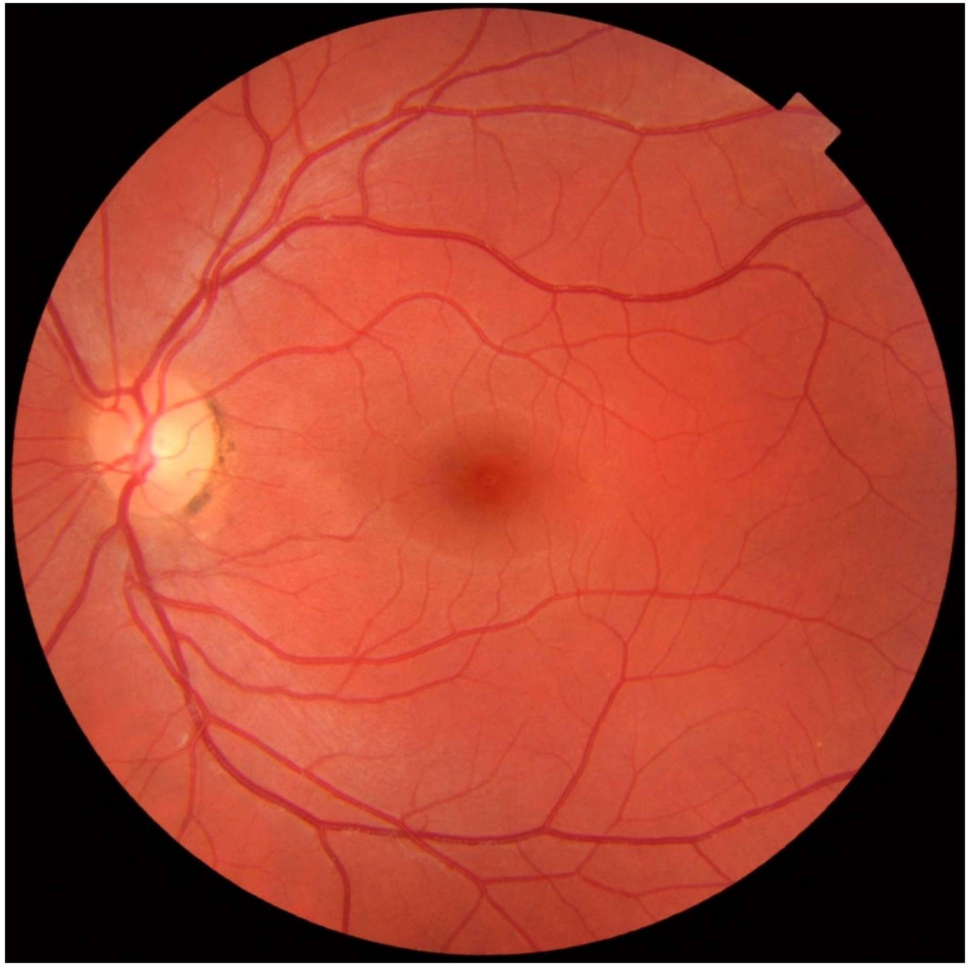
Robustness vs. Uncertainty Quantification

- Robustness aims to ensure the model performs well on shifted inputs
- Doesn't say anything about performance on original inputs!
  - A model that always predicts "dog" is robust, but not very useful
- What can we guarantee for performance on original inputs?
  - In general, we can't guarantee much (maybe the problem is just really hard!)
  - But, we can give **uncertainty quantification** ("knows what it doesn't know")
  - Initially focus on **on-distribution** (no shifts, no adversarial attacks, etc.)

Uncertainty Quantification



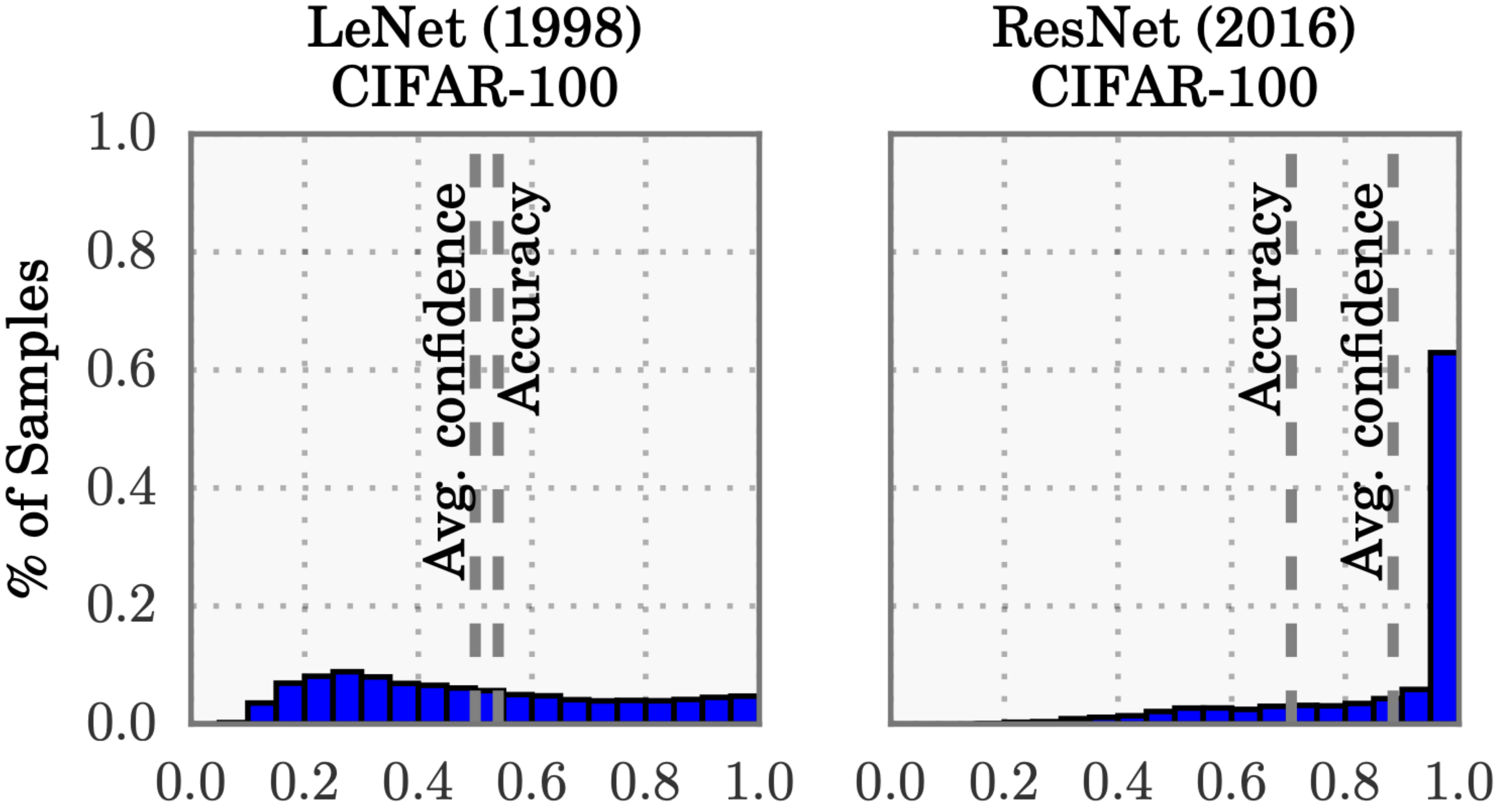
Uncertainty Quantification



Modern Neural Networks are Overconfident

- Guo et al., On Calibration of Modern Neural Networks. ICML 2017.

Modern Neural Networks are Overconfident



Uncertainty Quantification

- **Calibrated Prediction (Platt 1999, Guo 2017)**
  - Predict a **probability** $\vec{p}(x)_y$ for each label y
  - What does it mean for the probabilities to be correct?

- **Prediction Sets**
  - Predict a **set** $\hat{C}(x) \subseteq Y$ of possible labels
  - Set is correct if $y^* \in \hat{C}(x)$

Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift

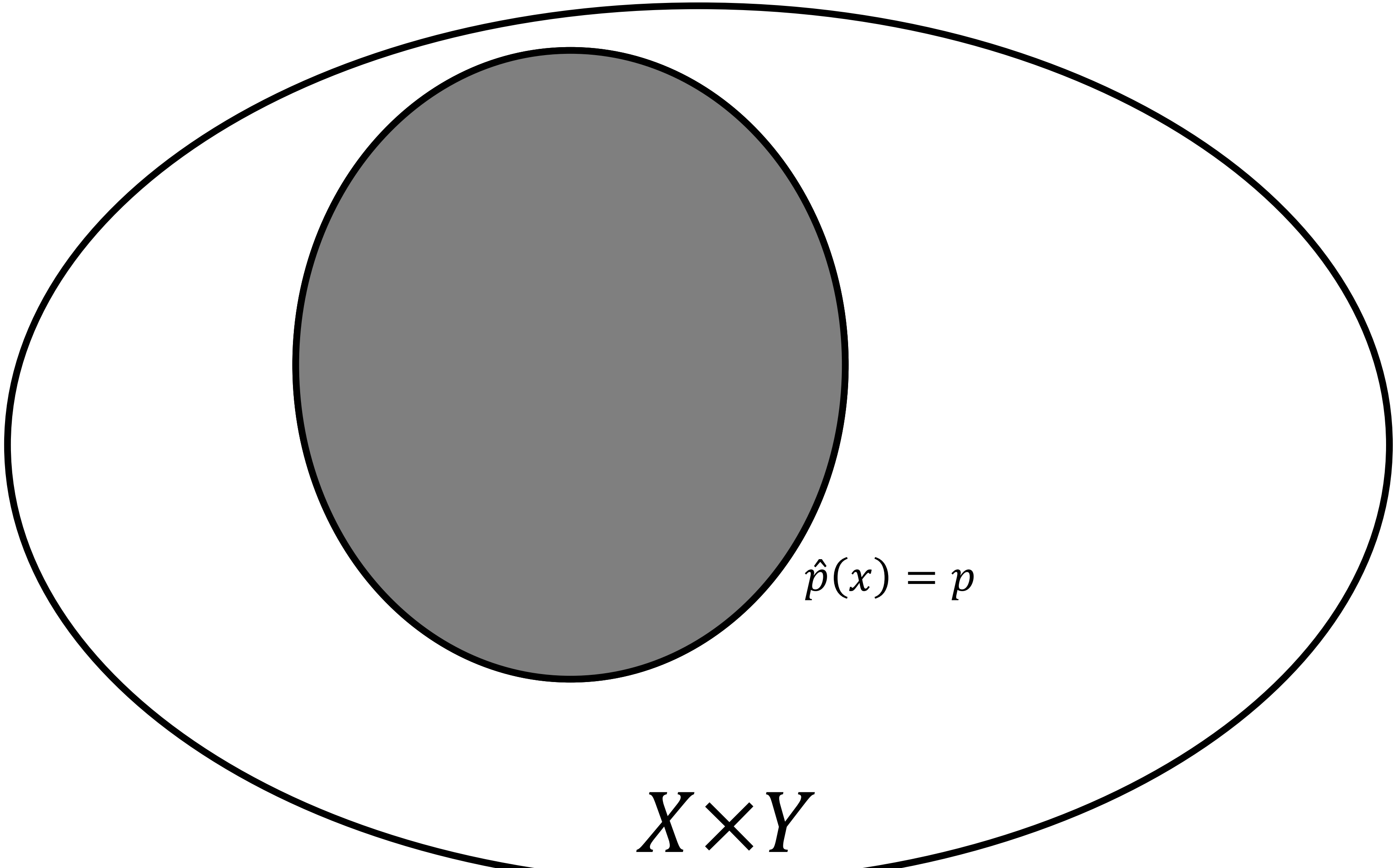
Calibrated Prediction

- Consider a probability predictor $\vec{p}: X \to [0,1]^{|Y|}$
  - Let $\hat{y}(x) = \arg\max_{y \in Y} \vec{p}(x)_y$ denote the corresponding labeling function
  - Let $\hat{p}(x) = \vec{p}\big(x, \hat{y}(x)\big)$ be the probability of the predicted label

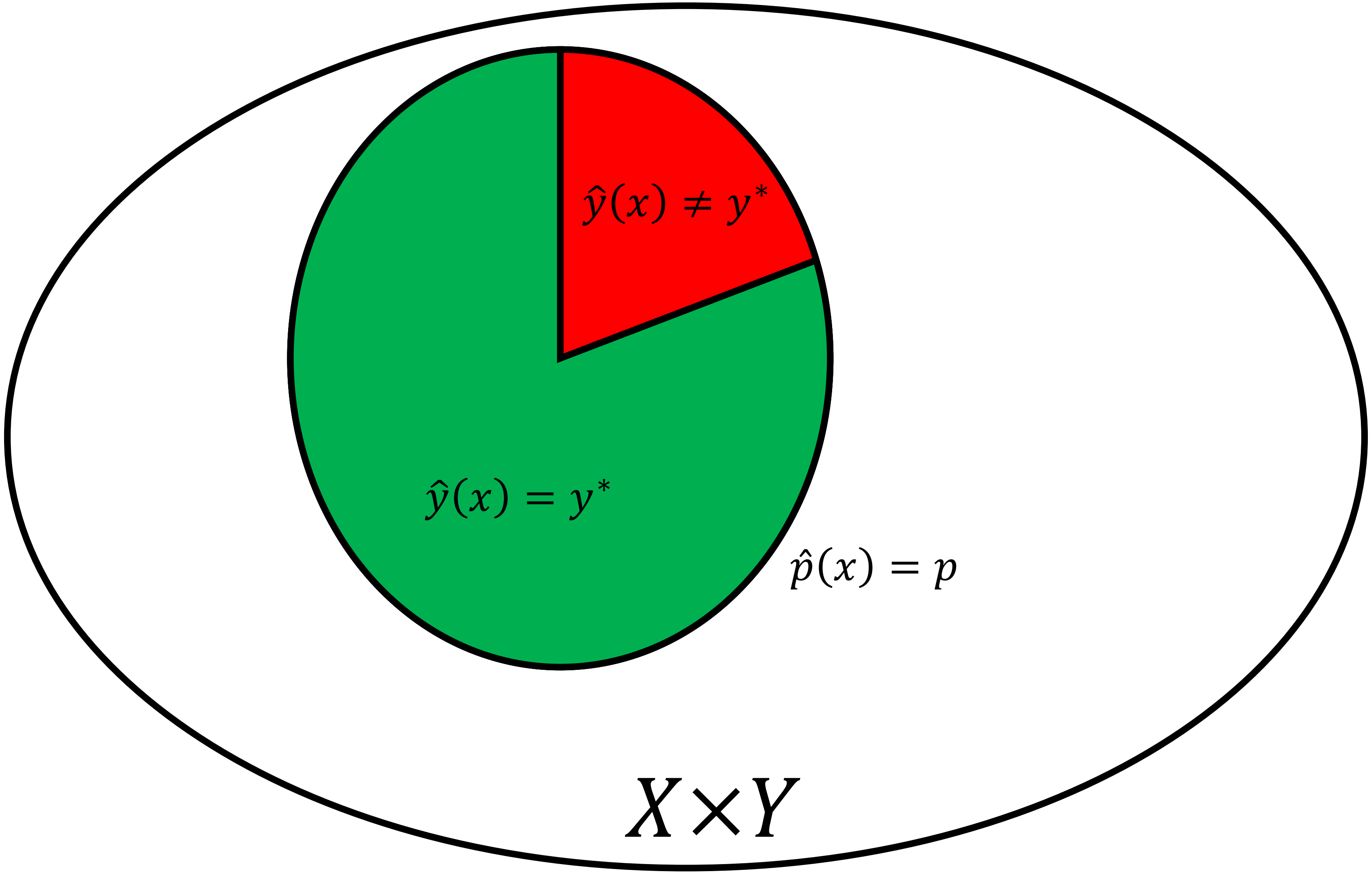
- We say $\hat{p}$ is **calibrated** if for all $p \in [0,1]$, we have

$$p = \Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) = p]$$

Calibrated Prediction



Calibrated Prediction



Goal: $\frac{\text{green}}{\text{green+red}} = p$

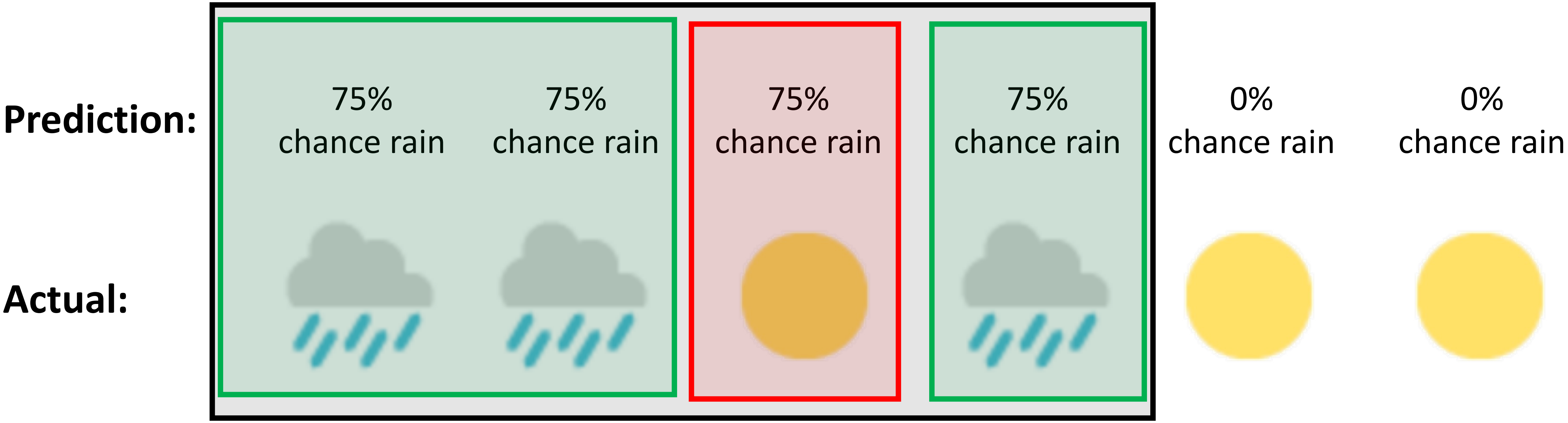
Calibrated Prediction

- What does "40% chance of rain" mean?

- Among all days with 40% chance of rain, it rains in 40% of them

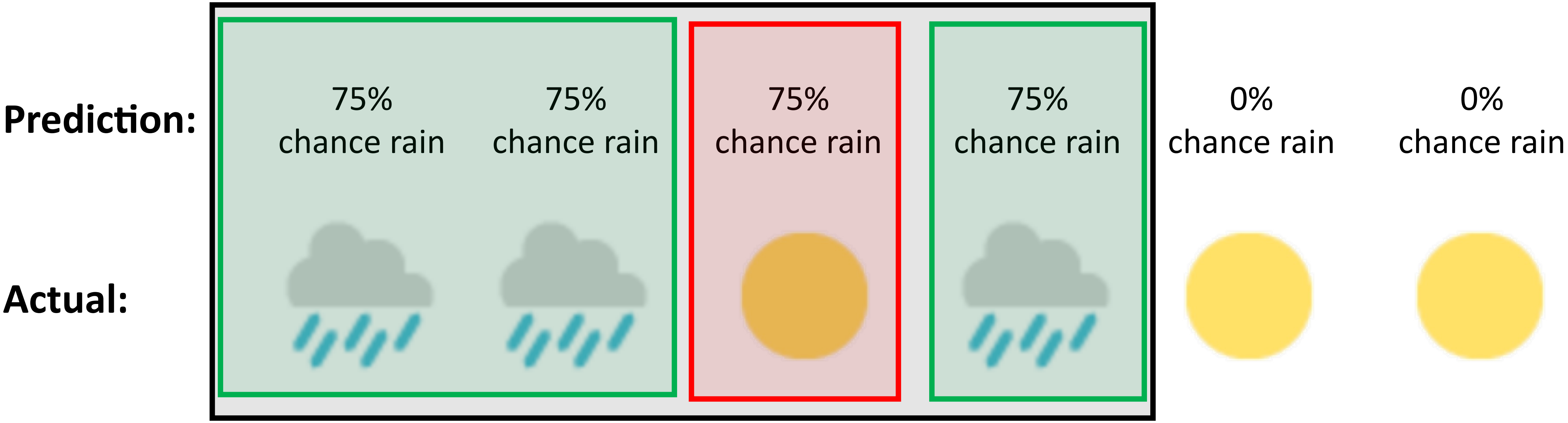
Mon 26	57°/39°	Partly Cloudy	7%	SW 7 mph
Tue 27	63°/54°	PM Showers	48%	SSE 13 mph
Wed 28	65°/39°	Showers	60%	SSW 19 mph
Thu 29	42°/28°	AM Showers	33%	NW 17 mph
Fri 01	47°/32°	Mostly Sunny	7%	SSW 9 mph
Sat 02	56°/40°	Partly Cloudy	18%	SSE 8 mph
Sun 03	61°/43°	AM Showers	48%	ENE 7 mph
Mon 04	61°/48°	Showers	43%	E 9 mph
Tue 05	63°/51°	Showers	49%	SE 11 mph
Wed 06	63°/48°	Showers	58%	SSE 13 mph
Thu 07	59°/47°	Showers	40%	SW 11 mph
Fri 08	60°/44°	Showers	42%	NNW 10 mph

Calibrated Prediction



$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = 0.75]$$

Calibrated Prediction



$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = 0.75] = 0.75$$

Calibrated Prediction

Prediction:	75% chance rain	75% chance rain	75% chance rain	75% chance rain	0% chance rain	0% chance rain
Actual:						

$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = 0]$$

Calibrated Prediction

Prediction: 75% chance rain | 75% chance rain | 75% chance rain | 75% chance rain | 0% chance rain | 0% chance rain

Actual:

$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = 0] = 0$$

Calibrated Prediction

Prediction:

75% chance rain	75% chance rain	75% chance rain	75% chance rain	25% chance rain	25% chance rain

Actual:

$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = 0.25] = 0$$

Calibrated Prediction

Prediction:

75% chance rain | 75% chance rain | 75% chance rain | 75% chance rain | 0% chance rain | 0% chance rain

Actual:

$$\Pr_{p(y^*)}[y^* = \text{rain} \mid \hat{p} = {\color{red}0}] = {\color{red}0.5}$$

Calibrated Prediction

- **Example 1:** Model has perfect prediction accuracy
  - Always predicts 0% rain or 100% rain, perfectly calibrated!
  - Always predicts 20% rain or 80% rain, miscalibrated!
- **Example 2:** Model predicts randomly rain vs. no rain
  - Always predicts 0% rain or 100% rain, miscalibrated!
  - Always predicts 50% rain or 50% rain, perfectly calibrated!
  - (Model is correct half the time)

Calibration and Binning

- **Recall:** Calibration is defined as

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) = p] = p \qquad (\forall p \in [0,1])$$

Calibration and Binning

- **Recall:** Calibration is defined as

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) = p] = p \qquad (\forall p \in [0,1])$$

  - Conditions on potentially zero probability event
  - In practice, two inputs may never have exactly the same probability $\hat{p}(x)$

- **Idea:** Bin probabilities into bins $P_i = [p_{\text{low},i}, p_{\text{high},i})$ instead:

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P_i] = \text{Conf}(P_i) \qquad (\forall i \in \{1, \ldots, k\})$$

Calibration and Binning

- **Idea:** Bin probabilities into bins $P_i = \left[p_{\mathrm{low},i}, p_{\mathrm{high},i}\right)$ instead:

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P_i] = \mathrm{Conf}(P_i) \qquad (\forall i \in \{1, \dots, k\})$$

  - $\mathrm{Conf}(P) = \mathbb{E}_{p(x,y^*)}[\hat{p}(x) \mid \hat{p}(x) \in P]$ is the average probability of bin P

Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift

Measuring Calibration

- **Recall:** Binned calibration is defined as

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P_i] = \mathrm{Conf}(P_i) \qquad (\forall i \in \{1, \dots, k\})$$

  - Chances of equality holding exactly is effectively zero
  - How to **measure** calibration error (instead of requiring exact calibration)?

- **Idea:** Use mean absolute error (called **expected calibration error**):

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}\left[\left|\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P] - \mathrm{Conf}(P)\right|\right]$$

Measuring Calibration

- **Idea:** Use mean absolute error (called **expected calibration error**):

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}\left[\left|\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P] - \mathrm{Conf}(P)\right|\right]$$

  - $P = [p_{\min}, p_{\max}) \subseteq [0,1]$ is a bin in probability space
  - $p(P) = \Pr_{p(x,y^*)}[\hat{p}(x) \in P]$ is the probability of bin P
  - $\mathrm{Conf}(P) = \mathbb{E}_{p(x,y^*)}[\hat{p}(x) \mid \hat{p}(x) \in P]$ is the average probability of bin P

Measuring Calibration

- **Idea:** Use mean absolute error (called **expected calibration error**):

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}\left[\left|\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P] - \mathrm{Conf}(P)\right|\right]$$

- Equivalently, we have

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}[|\mathrm{Acc}(P) - \mathrm{Conf}(P)|]$$

Measuring Calibration

- **Idea:** Use mean absolute error (called **expected calibration error**):

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}\left[\left|\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P] - \mathrm{Conf}(P)\right|\right]$$

- On a held-out test set Z, we have the approximation

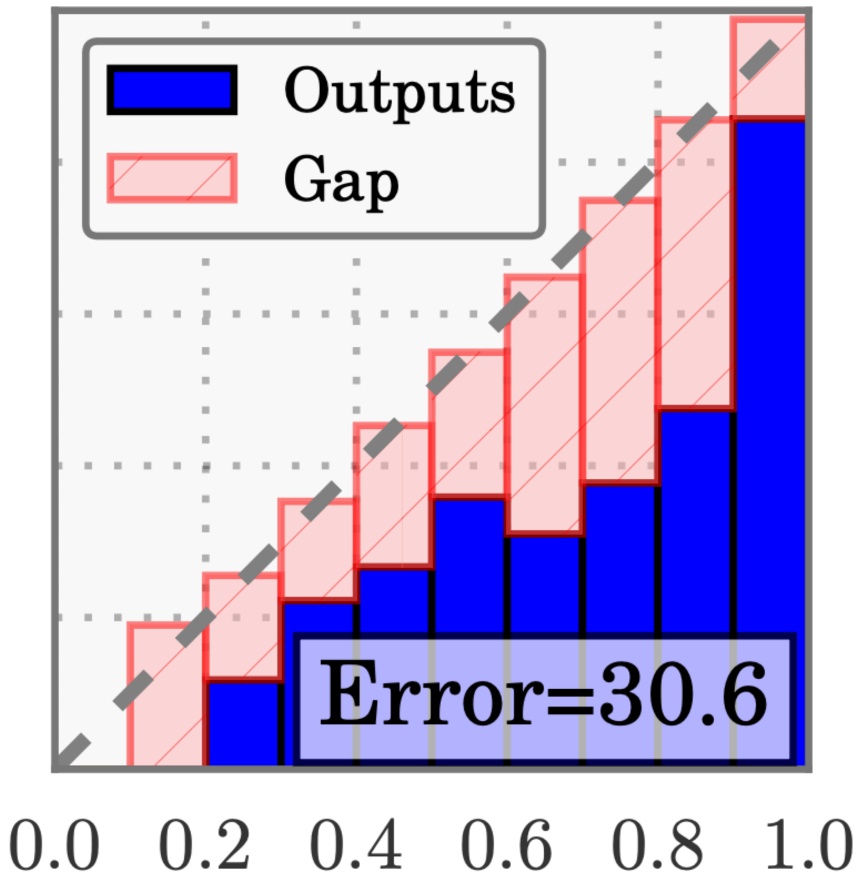
$$\mathrm{ECE}(\hat{p}; Z) = \sum_{i=1}^{k} \frac{|B_i|}{|Z|} \cdot \left| \frac{1}{|B_i|} \sum_{(x,y^*)\in B_i} 1(\hat{y}(x) = y^*) - \frac{1}{|B_i|} \sum_{(x,y^*)\in B_i} \hat{p}(x) \right|$$

  - Here, $B_i = \{\, (x, y^*) \in Z \mid \hat{p}(x) \in P_i \,\}$ is the bin in feature space

Reliability Diagrams

- For each bin P_i, plot accuracy

$$\Pr_{p(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) \in P]$$
$$\approx \frac{1}{|B_i|} \sum_{(x,y^*)\in B_i} 1(\hat{y}(x) = y^*)$$



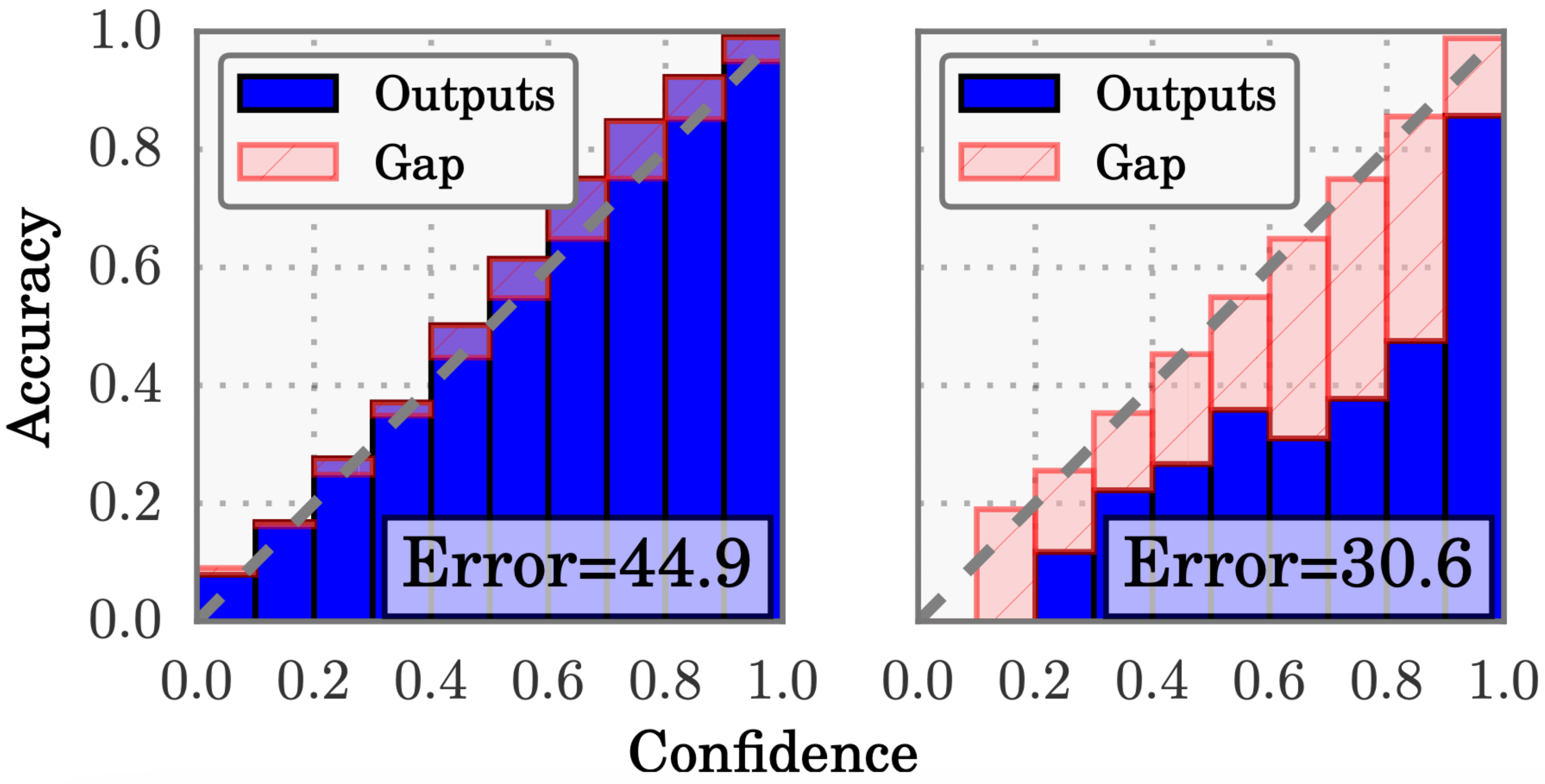
Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift

Miscalibration of Neural Networks

- Typical approach in deep learning
  - However, most state-of-the-art models have high calibration error
- **Potential explanation**
  - Models need to be **overparameterized** to aid optimization
  - Overparameterization leads to overfitting probabilities (even if accuracy is good!)

Miscalibration of Neural Networks



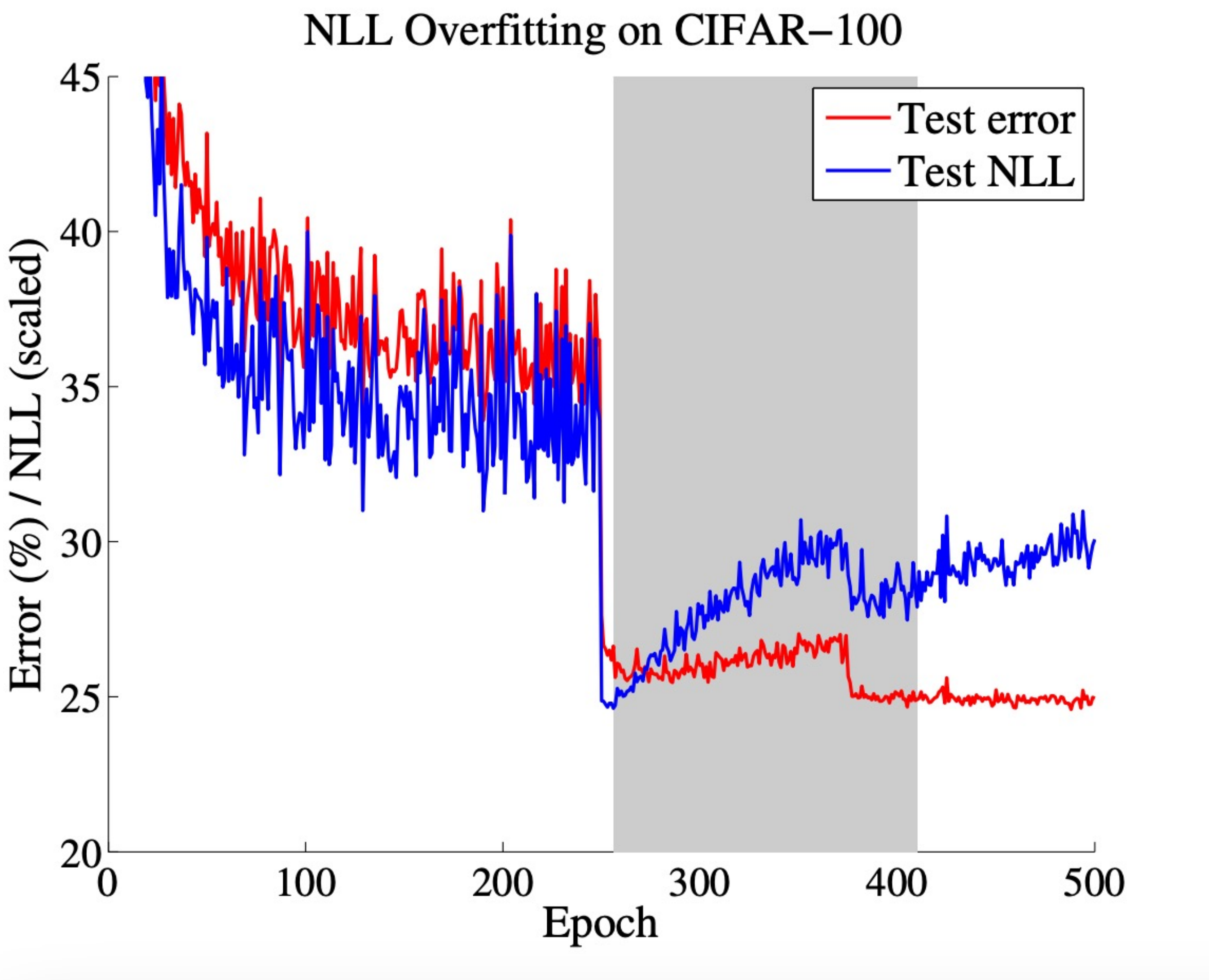
Recall: Negative Log Likelihood

- Negative log likelihood is

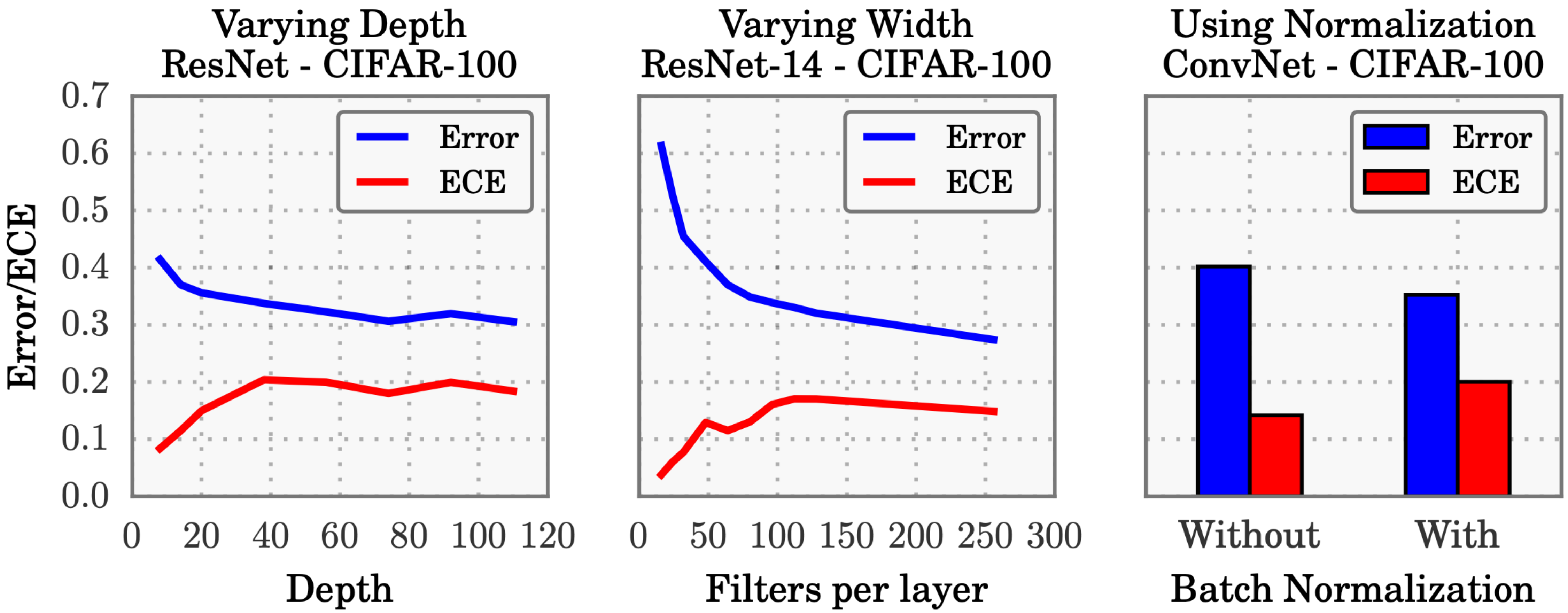
$$\mathrm{NLL}(\vec{p}) = -\sum_{(x,y^*)\in Z} \log \vec{p}(x)_{y^*}$$

- NLL is zero if and only if $\vec{p} = p^*$, where p^* are the "true" probabilities
  - Thus, good NLL roughly corresponds to good calibration

Miscalibration and Overfitting



Miscalibration and Overfitting



Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift

Improving Calibration

- How can we fix the problem?

- Better training algorithms
  - Regularization

- Post-hoc modifications (called **recalibration**)
  - Histogram binning
  - Temperature scaling

Recalibration

- **Goal:** Rescale outputs using a function to minimize ECE
- **Inputs:** Probability predictor $\hat{p}$, held-out calibration dataset Z
- **Output:** For some function ϕ: $[0,1] \rightarrow [0,1]$, define new probabilities

$$\hat{q}(x) = \phi\big(\hat{p}(x)\big)$$

- Works well since ϕ is a 1D transformation, so it is “simple”

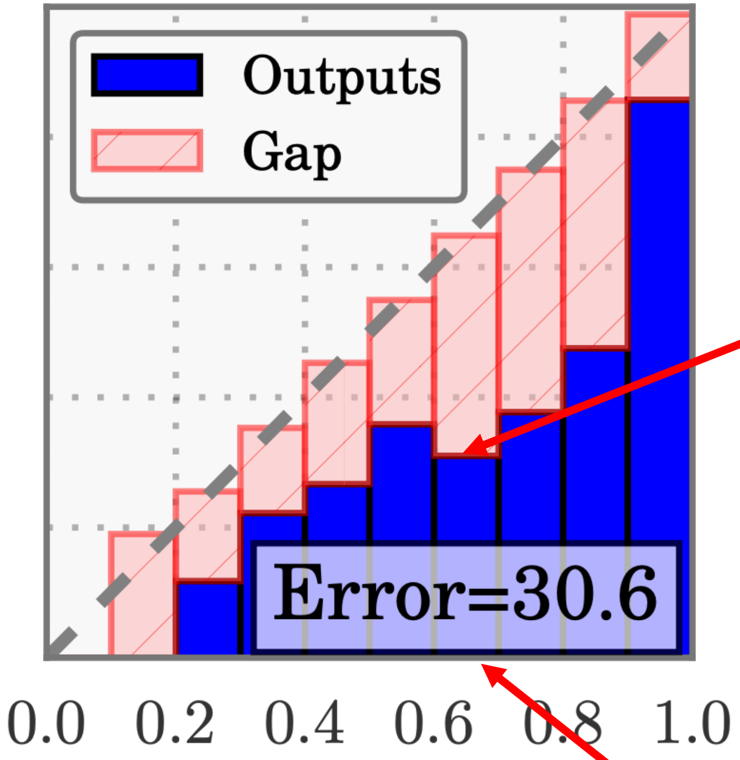
Histogram Binning

- **Idea:** Use a nonparametric ϕ obtained via binning

- **Algorithm:**
  - **Input:** Probability predictor $\hat{p}$, calibration dataset Z, example x
  - Let P be the probability bin containing x (i.e., $\hat{p}(x) \in P$)
  - Let $B = \{\,(x', y^*) \in Z \mid \hat{p}(x') \in P\,\}$ be the corresponding bin over Z
  - **Output:** Define the following (values can be precomputed for each bin):

$$\phi(x) = \frac{1}{|B|} \sum_{(x', y^*) \in B} 1(\hat{y}(x) = y^*)$$

Histogram Binning



Given x

$$\mathrm{Acc}(P) = \frac{1}{|B|} \sum_{(x',y^*)\in B} 1(\hat{y}(x') = y^*) = 0.35$$

$$\phi(x) = \mathrm{Acc}(P) = 0.35$$

$$\hat{p}(x) = 0.62 \in [0.6, 0.7) \coloneqq P$$

Histogram Binning

- **Why does this work?**
  - After transformation, all points with $\hat{p}(x) = P$ now have $\hat{q}(x) = \mathrm{Acc}(P)$
  - Their new bin is Q, where $\hat{q}(x) \in Q$
  - We also have $\mathrm{Acc}(Q) = \mathrm{Acc}(P)$ (ignoring other points in Q for simplicity)
  - Putting the above together, we have $\mathrm{Acc}(Q) = \hat{q}(x)$
  - Thus, $\mathrm{ECE}(\hat{q}) = 0$

- The derivation uses the true accuracy, but our algorithm uses the empirical accuracy, so there can be some error during evaluation

- **Intuition:** Set empirical ECE to zero on the calibration dataset Z

Aside: Isotonic Regression

- Modification of histogram binning
- Minimize jointly over bin boundaries and bin values
  - Histogram binning fixes the bin boundaries and only optimizes bin values
- Impose that the bin values are monotonically increasing to improve sample efficiency

Temperature Scaling

- Only fits **a single parameter** τ, even for the multi-class setting
  - Called the **temperature**
- Works best when ordering of probabilities is good

Temperature Scaling

- Consider the model family

$$\vec{q}_\tau(x) = \mathrm{softmax}\left(\frac{\mathrm{logits}(x)}{\tau}\right)$$

  - Taking $\tau = 1$ recovers the original model: $\vec{q}_1(x) = \vec{p}(x)$
  - Taking $\tau > 1$ decreases confidence ($\tau \to \infty$ yields probabilities equal to $\frac{1}{k}$)
  - Taking $\tau < 1$ increases confidence ($\tau \to 0$ yields probabilities equal in $\{0,1\}$)

- Choose τ to minimize NLL of $\vec{q}_\tau$ on calibration dataset Z
  - Can use grid search to do so (i.e., search over fixed set of choices for τ)

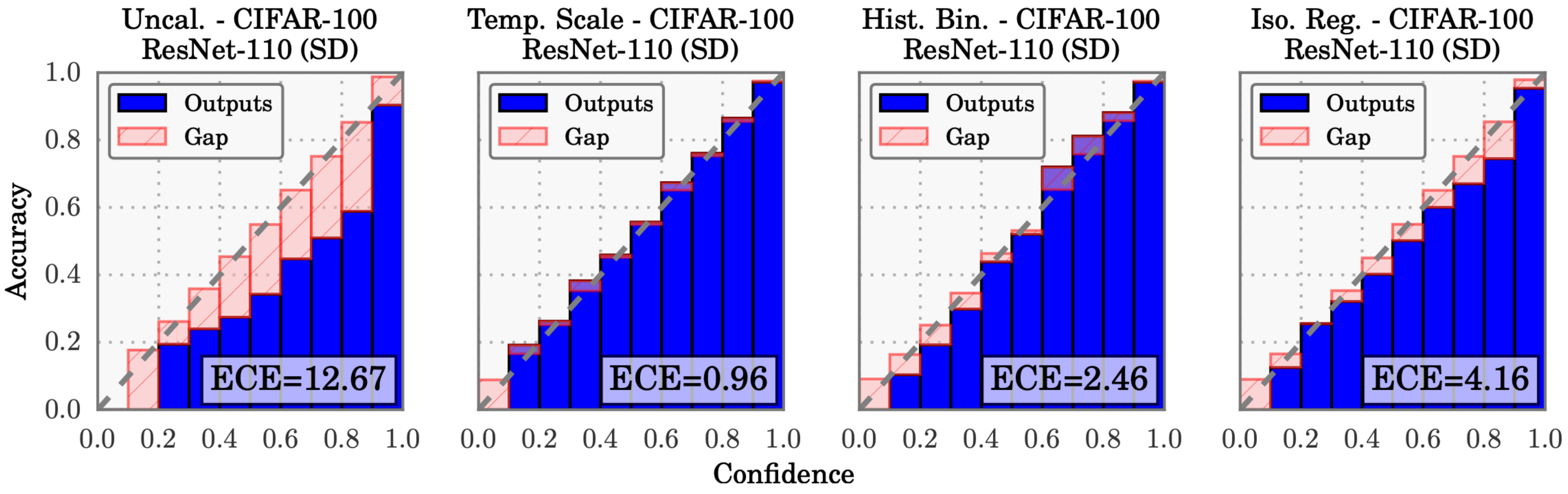
Re-Calibration for Multi-Class Classification

- Only calibrates the predicted probability for the most likely label
- To obtain a full vector of calibrated probabilities
  - Calibrate individually as $k = |Y|$ binary classification problems
  - Rescales probabilities separately for each class
  - Normalize probabilities to ensure they sum to one
- For histogram binning, the number of parameters can become large
  - # parameters $=$ # classes $\times$ # bins
- Temperature scaling naturally calibrates all class probabilities

Empirical Evaluation

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	9.19%	4.34%	5.22%	4.12%	**1.85%**	3.0%	21.13%
Cars	ResNet 50	4.3%	**1.74%**	4.29%	1.84%	2.35%	2.37%	10.5%
CIFAR-10	ResNet 110	4.6%	0.58%	0.81%	**0.54%**	0.83%	0.88%	1.0%
CIFAR-10	ResNet 110 (SD)	4.12%	0.67%	1.11%	0.9%	**0.6%**	0.64%	0.72%
CIFAR-10	Wide ResNet 32	4.52%	0.72%	1.08%	0.74%	**0.54%**	0.6%	0.72%
CIFAR-10	DenseNet 40	3.28%	0.44%	0.61%	0.81%	**0.33%**	0.41%	0.41%
CIFAR-10	LeNet 5	3.02%	1.56%	1.85%	1.59%	**0.93%**	1.15%	1.16%
CIFAR-100	ResNet 110	16.53%	2.66%	4.99%	5.46%	**1.26%**	1.32%	25.49%
CIFAR-100	ResNet 110 (SD)	12.67%	2.46%	4.16%	3.58%	0.96%	**0.9%**	20.09%
CIFAR-100	Wide ResNet 32	15.0%	3.01%	5.85%	5.77%	**2.32%**	2.57%	24.44%
CIFAR-100	DenseNet 40	10.37%	2.68%	4.51%	3.59%	1.18%	**1.09%**	21.87%
CIFAR-100	LeNet 5	4.85%	6.48%	2.35%	3.77%	**2.02%**	2.09%	13.24%
ImageNet	DenseNet 161	6.28%	4.52%	5.18%	3.51%	**1.99%**	2.24%	-
ImageNet	ResNet 152	5.48%	4.36%	4.77%	3.56%	**1.86%**	2.23%	-
SVHN	ResNet 152 (SD)	0.44%	**0.14%**	0.28%	0.22%	0.17%	0.27%	0.17%
20 News	DAN 3	8.02%	**3.6%**	5.52%	4.98%	4.11%	4.61%	9.1%
Reuters	DAN 3	0.85%	1.75%	1.15%	0.97%	0.91%	**0.66%**	1.58%
SST Binary	TreeLSTM	6.63%	1.93%	**1.65%**	2.27%	1.84%	1.84%	1.84%
SST Fine Grained	TreeLSTM	6.71%	2.09%	**1.65%**	2.61%	2.56%	2.98%	2.39%

Reliability Diagrams for Re-Calibration



Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift

Calibration Under Distribution Shift

- Uncertainty quantification is especially important for flagging potential distribution shift

- **Goal:**
  - Given calibration dataset of i.i.d. samples from p and unlabeled examples from shifted distribution q
  - Obtain model that is calibrated with respect to q:

$$p = \Pr_{q(x,y^*)}[\hat{y}(x) = y^* \mid \hat{p}(x) = p]$$

Importance Weighted Calibration Error

- **Recall:** We have

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{p(P)}\left[\left|\mathrm{Acc}_p(P) - \mathrm{Conf}_p(P)\right|\right]$$

- **Shifted ECE:** We have

$$\mathrm{ECE}(\hat{p}) = \mathbb{E}_{q(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right|\right]$$

Importance Weighted Calibration Error

- Note that

$$\mathrm{ECE}(\hat{p})$$

Importance Weighted Calibration Error

- Note that

$$\begin{aligned}\mathrm{ECE}(\hat{p}) &= \mathbb{E}_{q(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right|\right] \\ &= \mathbb{E}_{p(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right| \cdot w(P)\right]\end{aligned}$$

- We have

$$w(P)$$

Importance Weighted Calibration Error

- Note that

$$\begin{aligned}\mathrm{ECE}(\hat{p}) &= \mathbb{E}_{q(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right|\right] \\ &= \mathbb{E}_{p(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right| \cdot w(P)\right]\end{aligned}$$

- We have

$$w(P) = \frac{q(P)}{p(P)}$$

Importance Weighted Calibration Error

- Note that

$$\begin{aligned}\mathrm{ECE}(\hat{p}) &= \mathbb{E}_{q(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right|\right] \\ &= \mathbb{E}_{p(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right| \cdot w(P)\right]\end{aligned}$$

- We have

$$w(P) = \frac{q(P)}{p(P)} = \frac{\Pr_{q(x,y^*)}[\hat{p}(x) \in P]}{\Pr_{p(x,y^*)}[\hat{p}(x) \in P]}$$

Importance Weighted Calibration Error

- Note that

$$\begin{aligned}\mathrm{ECE}(\hat{p}) &= \mathbb{E}_{q(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right|\right] \\ &= \mathbb{E}_{p(P)}\left[\left|\mathrm{Acc}_q(P) - \mathrm{Conf}_q(P)\right| \cdot w(P)\right]\end{aligned}$$

- Similar importance weighting for $\mathrm{Acc}_q(P)$ and $\mathrm{Conf}_q(P)$

Calibration Under Distribution Shift

- **Histogram binning**
  - Minimize importance weighted calibration error

- **Temperature scaling**
  - Minimize importance weighted NLL

Empirical Results

$\hat{y}(x) = $ letter tray

$\hat{f}(x) = {\color{red}1.00} \to 0.99 \to {\color{green}0.38}$

$y(x) = $ laptop computer

$\hat{y}(x) = $ tape dispenser

$\hat{f}(x) = {\color{red}1.00} \to 0.93 \to {\color{green}0.60}$

$y(x) = $ phone

$\hat{y}(x) = $ bookcase

$\hat{f}(x) = {\color{red}0.99} \to 0.91 \to {\color{green}0.53}$

$y(x) = $ stapler

- $\hat{f}(x) = $ (original)$\to$(temperature scaling) $\to$ (ours)

Park et al., Calibrated Predictions with Covariate Shift via Unsupervised Domain Adaptation. AISTATS 2020.

Agenda

- Definition of calibration
- Measuring calibration
- Miscalibration of neural networks
- Re-calibration
- Calibration under covariate shift