CSCE 689: Special Topics in Trustworthy NLP

Lecture 15: Model Uncertainty (1)

Kuan-Hao Huang
khhuang@tamu.edu

(Some slides adapted from COLING-22 Tutorial: Uncertainty Estimation for Natural Language Processing)

Invited Talk

- **Time:** 10/9 Wednesday lecture time on Zoom: https://tamu.zoom.us/my/khhuang?pwd=oAdWOKVOCGPApqDbJnVtktdW2AE6nb.1
- **Title**: Machine Unlearning: the general theory and LLM practice for privacy
- **Speaker**: Eli Chien, Postdoc at the Georgia Institute of Technology
- **Abstract**:

  "The right to be forgotten" is the concept from GDPR that data holder (server) should erase the data and the corresponding derivatives whenever the original data providers (users) request for it. It is the common practice that LLM are trained on extensive and diverse dataset, which are usually generated from users. While retraining from scratch without those data is the gold standard, it is prohibitively costly. The goal of machine unlearning is to develop efficient approaches to approximate such gold standard, thus obeying the privacy regulation in laws like GDPR. In this talk, I will first introduce the generic machine unlearning problem and my recent progress in unlearning theory. Then we will dive into our empirical studies of unlearning for LLM, which highlight the current popular unlearning heuristics and pitfalls of empirical unlearning evaluation.

Team Project Highlights

- 10/11 Friday in person
- 9 teams
- A 4-min presentation for each team
  - The topic you choose
  - An introduction to the task
  - Evaluation metrics
  - The dataset, models, and approaches you plan to use
  - Preliminary results (optional)

Model Uncertainty

Hello! Could you help me reserve a table at the *"The Best"* restaurant for tomorrow at 12pm?

Of course! I've reserved a table at the *"The Best"* restaurant for tomorrow at 12pm.

Hello! Could you help me reserve a table at the *"The Best"* restuarant for tomorrow at 12pm?

#$^&*^$@!%^*&@%$(*&…

Hello! Could you help me reserve a table at the *"The Best"* restaurant for tomorrow at 12pm?

Of course! I've reserved a table at the *"The Best"* restaurant for tomorrow at 12pm. (Confidence: 98%)

Hello! Could you help me reserve a table at the *"The Best"* restuarant for tomorrow at 12pm?

#$^&*^$@!%^*&@%$(*&…
(Confidence: 40%)

Provide additional information to decide if we should trust the answers

Why Do We Need Uncertainty Estimates?

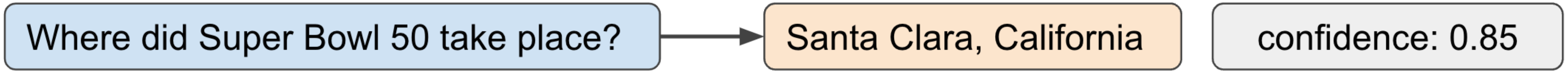
- Even the best models will sometimes be wrong
  - Uncertainty estimates can help us understand when the model might be wrong
  - Being aware of uncertainty can allow us to prepare for having to catch mistakes

Why Do We Need Uncertainty Estimates?

- Build or reduce trust in certain pointwise predictions
- Compare the performance of different
- Identify areas of improvement for a given model
- List all plausible answers subject to specified probabilistic guarantees
- Produce more natural responses (that reflect confidence) for dialogue agents
- Abstain from making predictions when in doubt
- … and more

Ways of Expressing Uncertainty

- As a numerical value (e.g. in [0,1]) returned with each prediction:

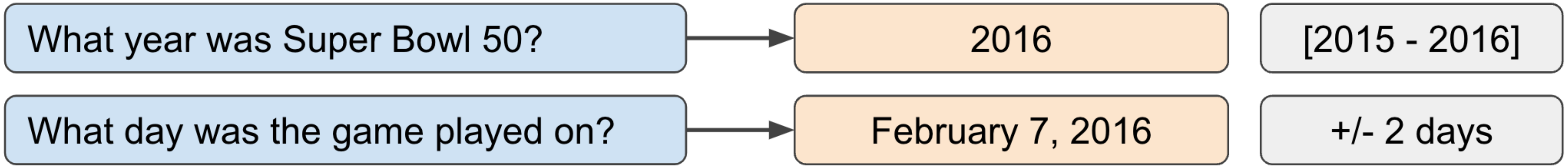


Where did Super Bowl 50 take place?

Santa Clara, California

confidence: 0.85

- As a confidence interval around a numerical value:



What year was Super Bowl 50?

2016

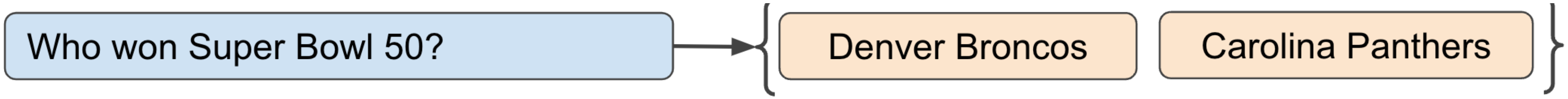
[2015 - 2016]

What day was the game played on?

February 7, 2016

+/- 2 days

- As a set of candidate answers:

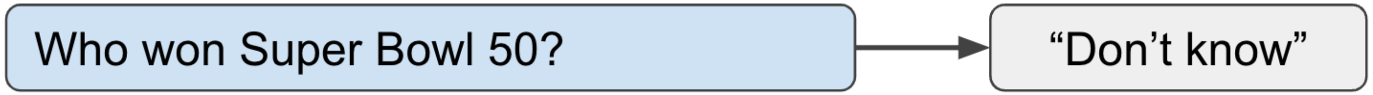


Who won Super Bowl 50?

Denver Broncos

Carolina Panthers

- As a decision to abstain from answering:

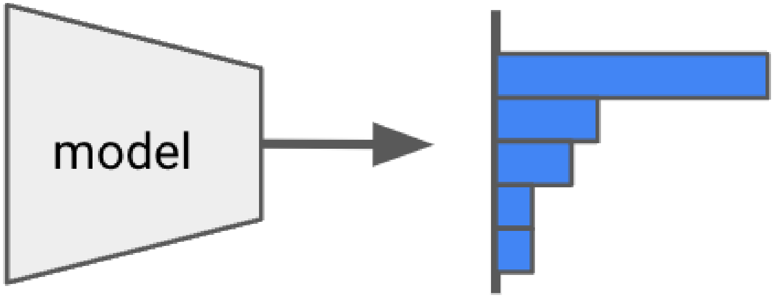


Who won Super Bowl 50?

“Don’t know”

Uncertainty Estimates for Text Classification

- Softmax-based measure



$$P(y = c \mid \mathbf{x}) = \mathrm{softmax}(z_c) \qquad \mathrm{softmax}(t) = \frac{e^{z_c}}{\sum_c e^{z_c}}$$

Softmax Function

What is the problem of using softmax?

Uncertainty Estimates for Text Classification

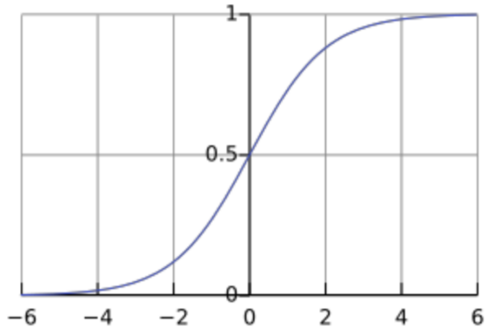
- Softmax-based measure
  - Overconfidence in predictions
  - Not for unseen data (out-of-distribution)
  - Lack of calibration
    - We say our model is calibrated if

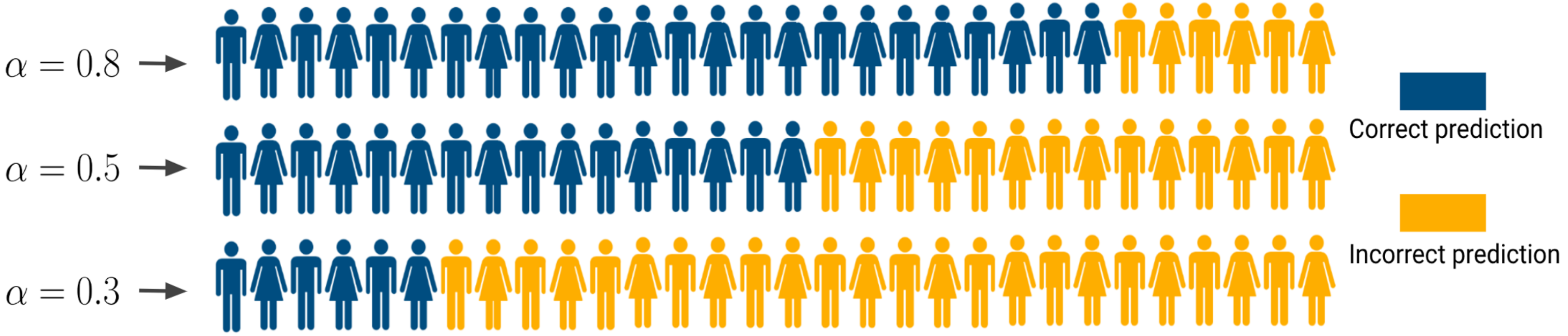
$$\mathbb{P}(\text{model is correct} \mid \text{confidence is } \alpha) = \alpha$$

    - In other words, α-fraction of all predictions with confidence α should be correct

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Sigmoid Function





Measure Calibration Error

- The quality of our confidence is captured by its empirical calibration error

$$\mathrm{CE}(\alpha) = \left| \widehat{\mathbb{P}}(\text{model is correct} \mid \text{confidence is } \alpha) - \alpha \right|$$

Observed frequency (accuracy)

Confidence level

- We can estimate the calibration error with binning

Is correct	…	…	0	1	0	0	1	1	0	1	…	…
Confidence	…	…	0.42	0.42	0.43	0.44	0.45	0.46	0.47	0.48	…	…

Avg. bin confidence = 0.45
Avg. bin accuracy = 0.50 → Calibration error = 0.05

Measure Calibration Error

- The expected calibration error is estimated by averaging over bins

# of bins

$$\mathrm{ECE} = \sum_{k=1}^{|\mathcal{B}|} \frac{|\mathcal{B}_k|}{N} \left| \frac{\sum_{i\in\mathcal{B}_k} \mathbf{1}\{y_i = \hat{y}_i\}}{|\mathcal{B}_k|} - \frac{\sum_{i\in\mathcal{B}_k} \hat{c}_i}{|\mathcal{B}_k|} \right|$$

fraction of samples in bin k

Measure Calibration Error

- The expected calibration error is estimated by averaging over bins

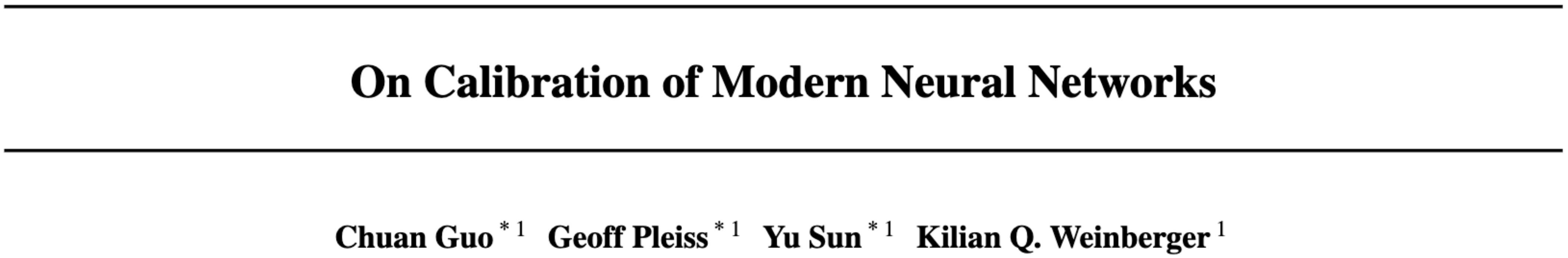
$$\mathrm{ECE} = \sum_{k=1}^{|\mathcal{B}|} \frac{|\mathcal{B}_k|}{N} \left| \underset{\text{Accuracy in bin k}}{\boxed{\frac{\sum_{i \in \mathcal{B}_k} \mathbf{1}\{y_i = \hat{y}_i\}}{|\mathcal{B}_k|}}} - \frac{\sum_{i \in \mathcal{B}_k} \hat{c}_i}{|\mathcal{B}_k|} \right|$$

Measure Calibration Error

- The expected calibration error is estimated by averaging over bins

$$\mathrm{ECE} = \sum_{k=1}^{|\mathcal{B}|} \frac{|\mathcal{B}_k|}{N} \left| \frac{\sum_{i \in \mathcal{B}_k} \mathbf{1}\{y_i = \hat{y}_i\}}{|\mathcal{B}_k|} - \boxed{\frac{\sum_{i \in \mathcal{B}_k} \hat{c}_i}{|\mathcal{B}_k|}} \right|$$

Avg. confidence in bin k

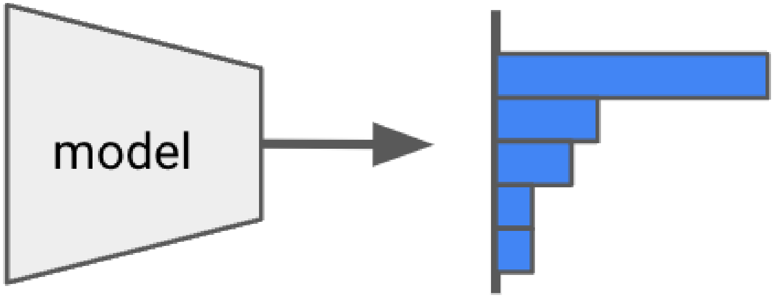


On Calibration of Modern Neural Networks

Chuan Guo [*1] Geoff Pleiss [*1] Yu Sun [*1] Kilian Q. Weinberger [1]

Uncertainty Estimates for Text Classification

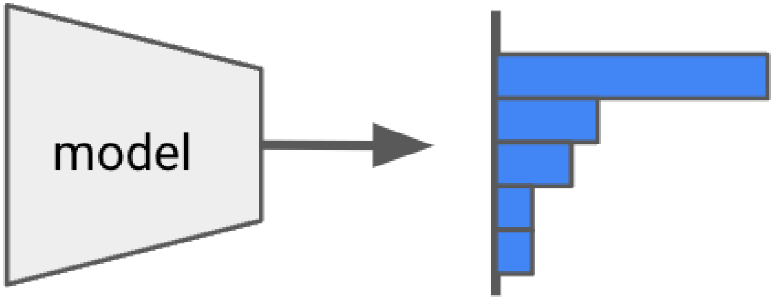
- Softmax-based measure



$$P(y = c |\, \mathbf{x}) = \text{softmax}(z_c) \qquad \text{softmax}(t) = \frac{e^{z_c}}{\sum_c e^{z_c}}$$

Softmax Function

Softmax with Temperature



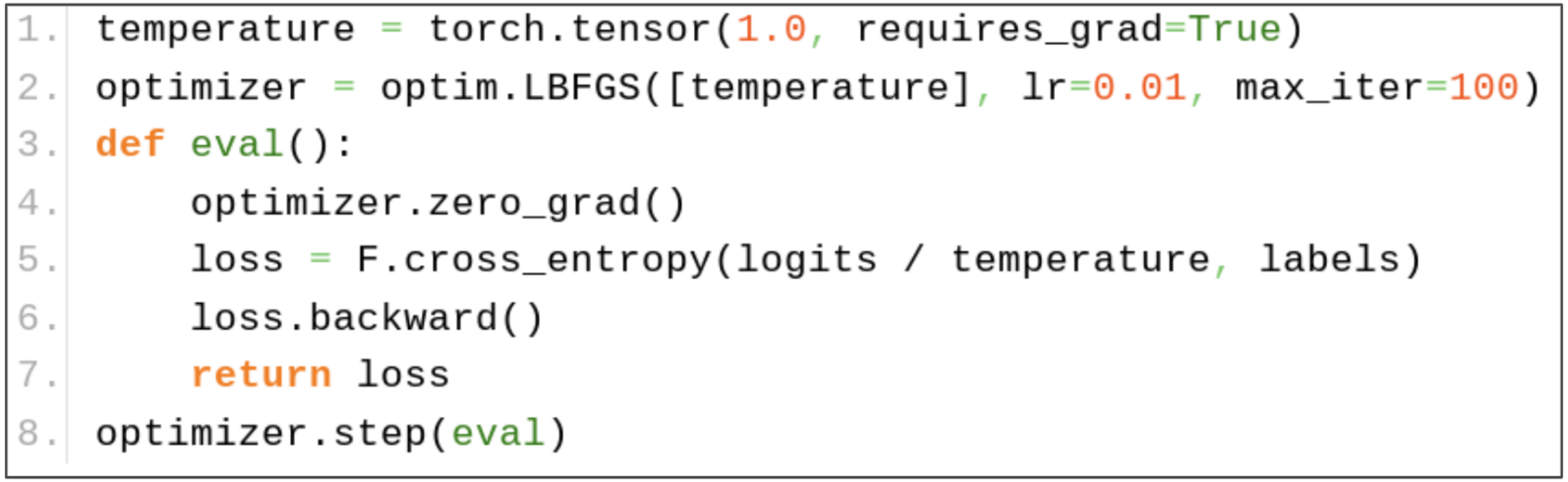
$$p(y_i \mid x) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

Higher T: softens probabilities.
Lower T: sharpens probabilities.

Temperature Scaling

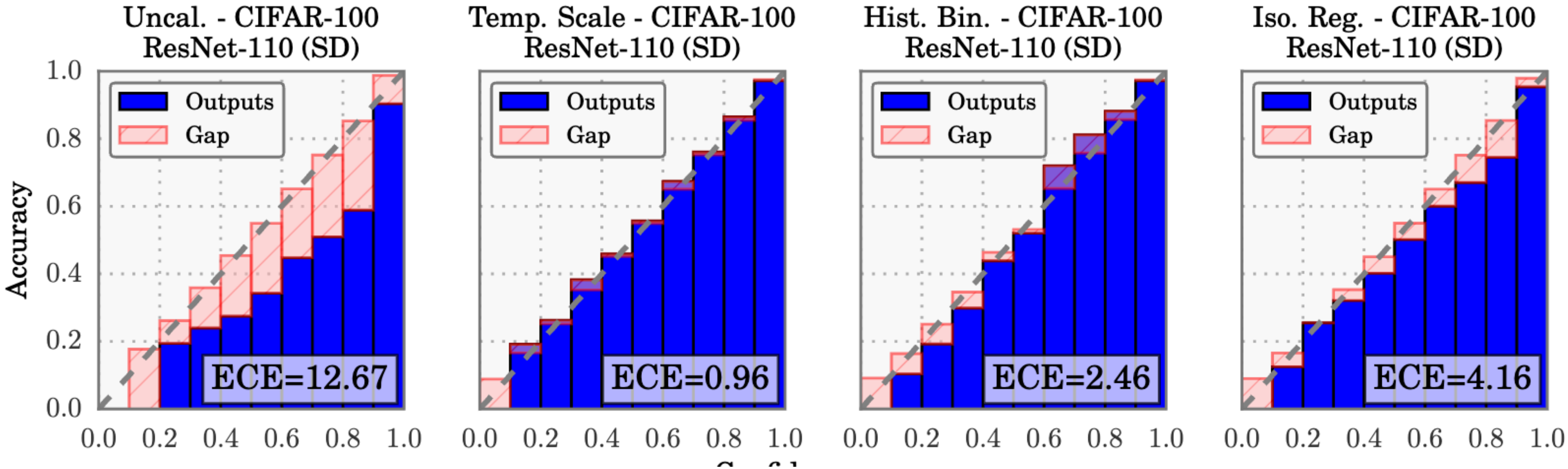
- Post-hoc rescale the logits
- Optimize T on a held-out calibration to minimize the negative log-likelihood

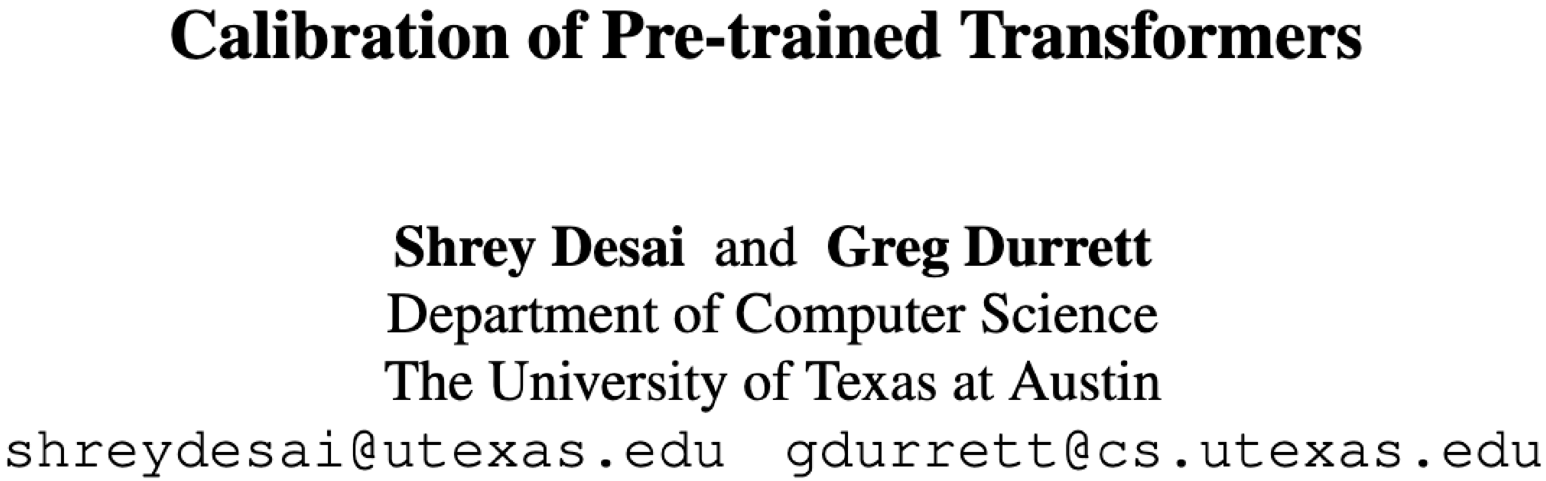
$$p(y_i \mid x) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$



```python
temperature = torch.tensor(1.0, requires_grad=True)
optimizer = optim.LBFGS([temperature], lr=0.01, max_iter=100)
def eval():
    optimizer.zero_grad()
    loss = F.cross_entropy(logits / temperature, labels)
    loss.backward()
    return loss
optimizer.step(eval)
```

Calibration Example



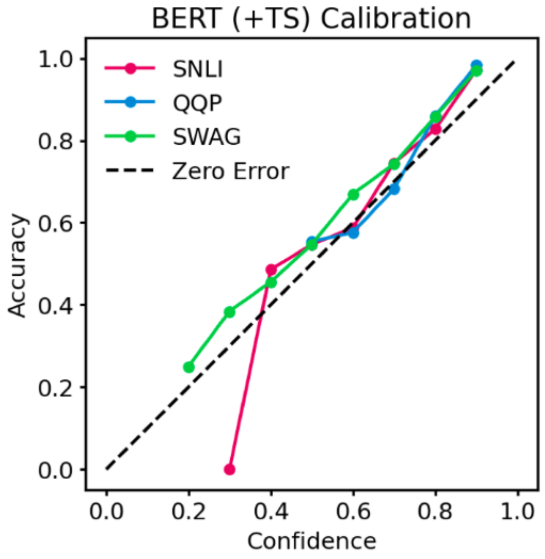


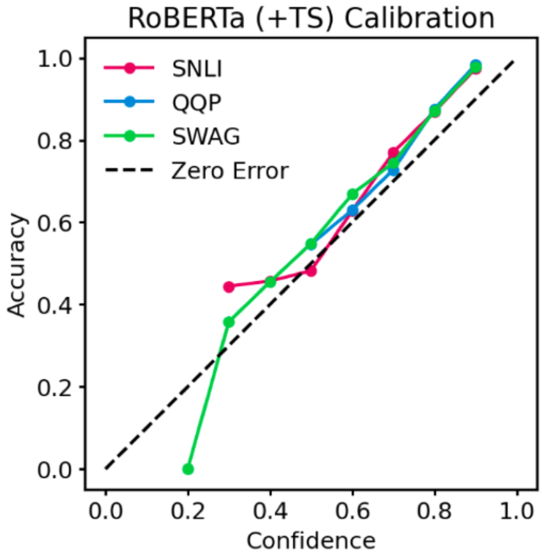
Calibration of Pre-trained Transformers

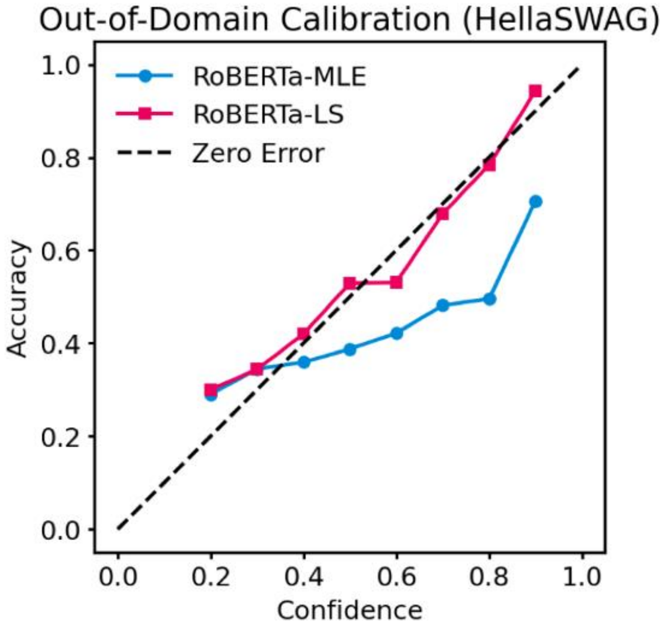
Shrey Desai and **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
shreydesai@utexas.edu gdurrett@cs.utexas.edu

Results

In-domain

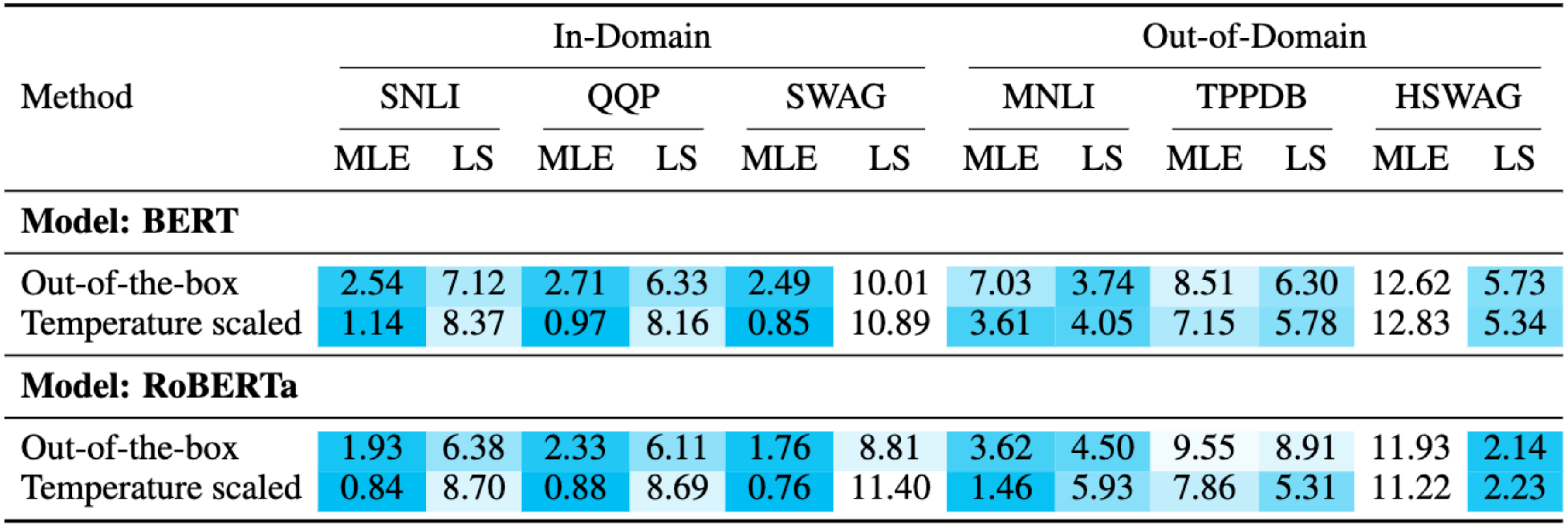

BERT (+TS) Calibration


RoBERTa (+TS) Calibration

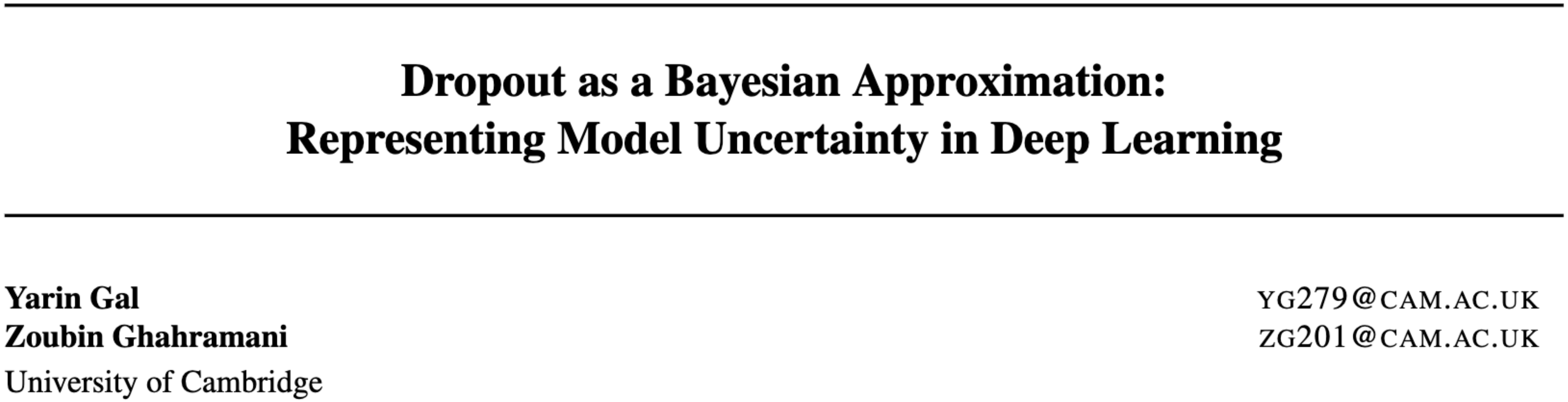

Out-of-Domain Calibration (HellaSWAG)

Out-of-Domain

Results



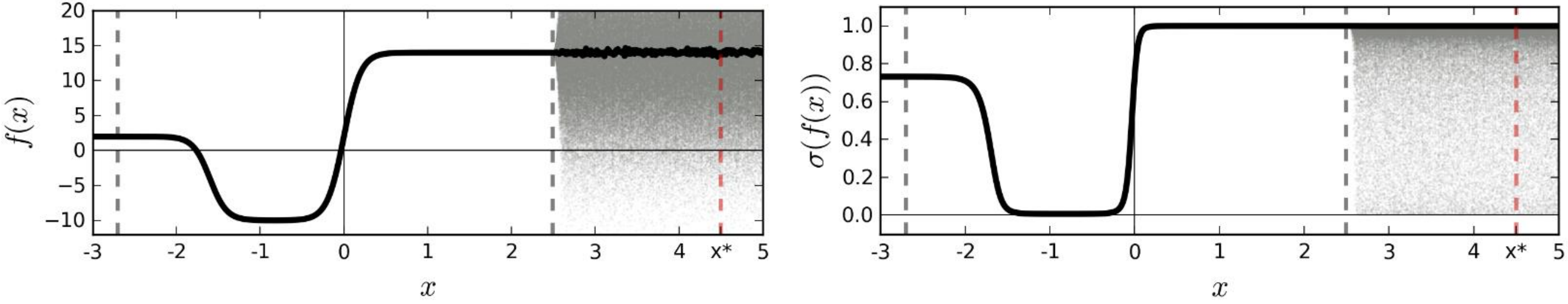
Method	In-Domain						Out-of-Domain					
	SNLI		QQP		SWAG		MNLI		TPPDB		HSWAG	
	MLE	LS	MLE	LS	MLE	LS	MLE	LS	MLE	LS	MLE	LS
Model: BERT												
Out-of-the-box	2.54	7.12	2.71	6.33	2.49	10.01	7.03	3.74	8.51	6.30	12.62	5.73
Temperature scaled	1.14	8.37	0.97	8.16	0.85	10.89	3.61	4.05	7.15	5.78	12.83	5.34
Model: RoBERTa												
Out-of-the-box	1.93	6.38	2.33	6.11	1.76	8.81	3.62	4.50	9.55	8.91	11.93	2.14
Temperature scaled	0.84	8.70	0.88	8.69	0.76	11.40	1.46	5.93	7.86	5.31	11.22	2.23



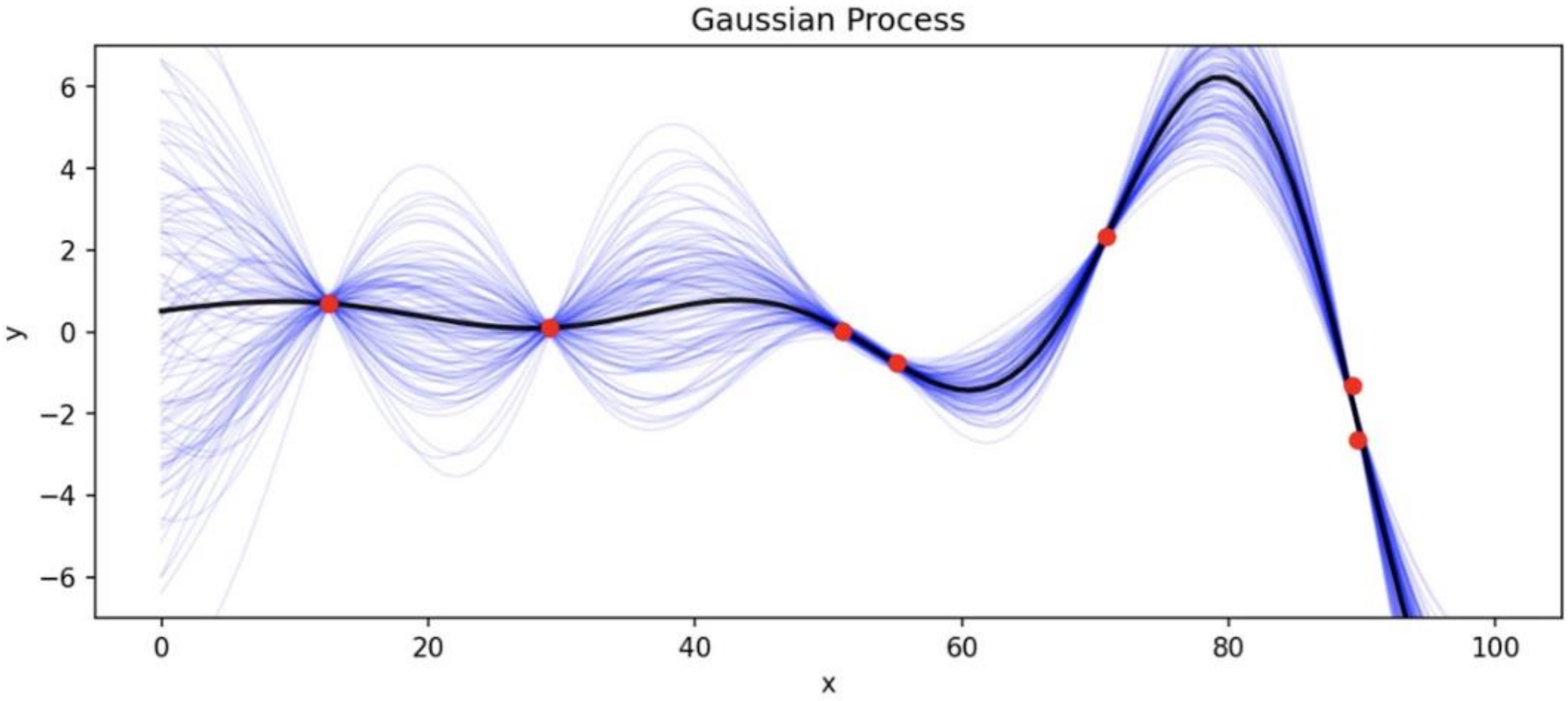
Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Yarin Gal yg279@cam.ac.uk
Zoubin Ghahramani zg201@cam.ac.uk
University of Cambridge

Softmax for Out-of-Distribution Data

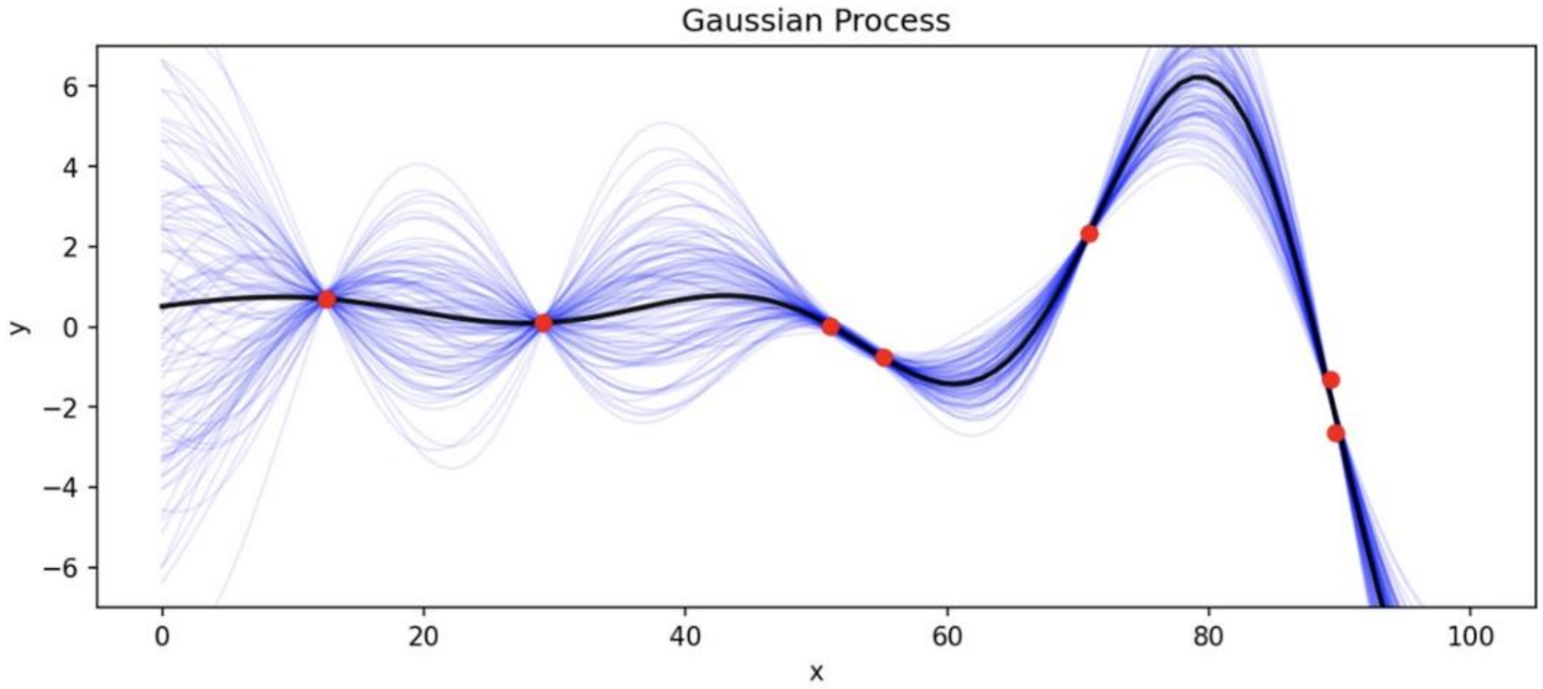


Gaussian Process



Variance is The Key

- Larger variance → larger uncertainty



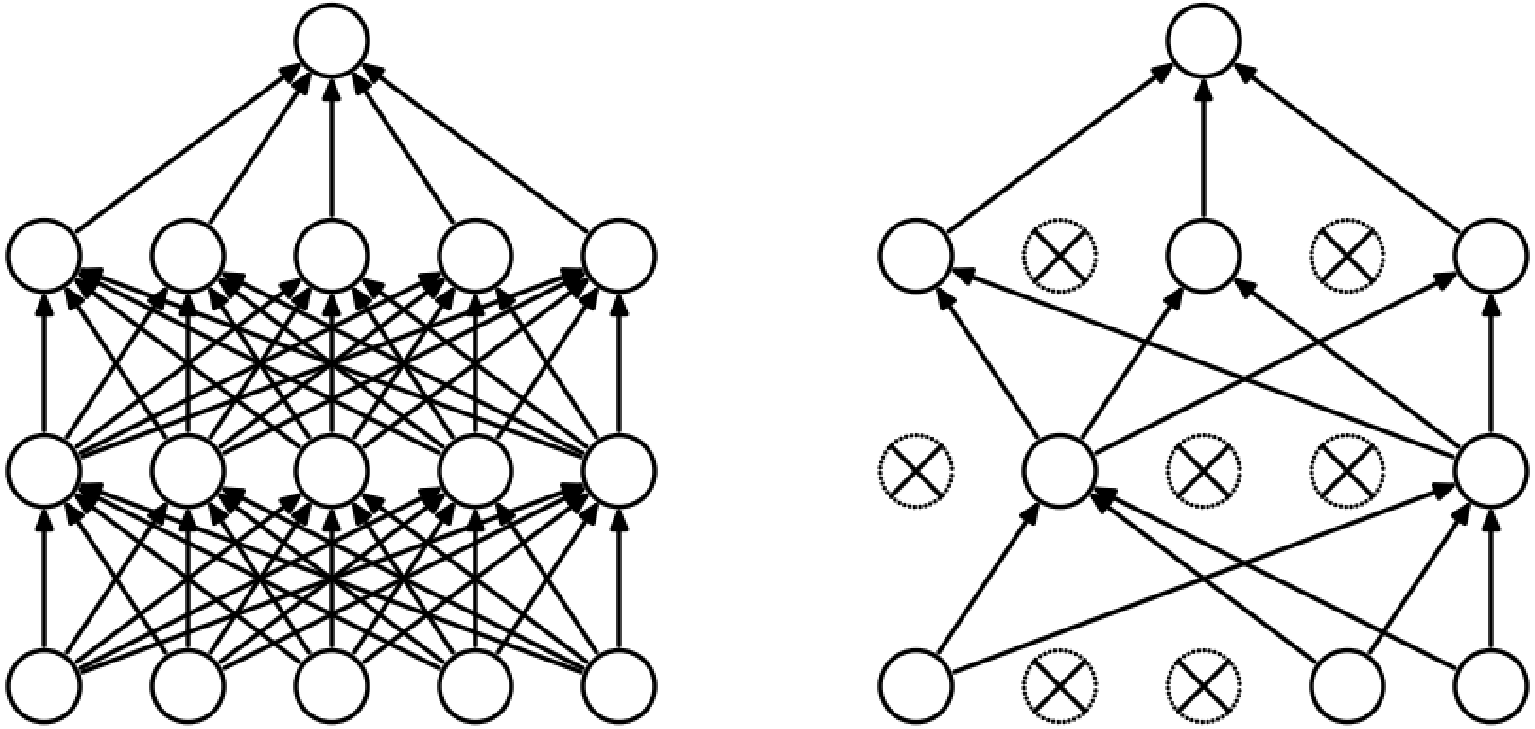
Dropout as a Bayesian Approximation

- Dropout objective minimizes the KL-divergence between an approximate distribution and the posterior of a deep Gaussian process

$$\mathcal{L}_{\text{dropout}} := \frac{1}{N}\sum_{i=1}^{N} E(\mathbf{y}_i, \widehat{\mathbf{y}}_i) + \lambda \sum_{i=1}^{L} \left(||\mathbf{W}_i||_2^2 + ||\mathbf{b}_i||_2^2\right).$$

$$\mathcal{L}_{\text{VI}} := \int q_\theta(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) \mathrm{d}\omega - \mathrm{KL}(q_\theta(\omega)||p(\omega))$$

What is Dropout?



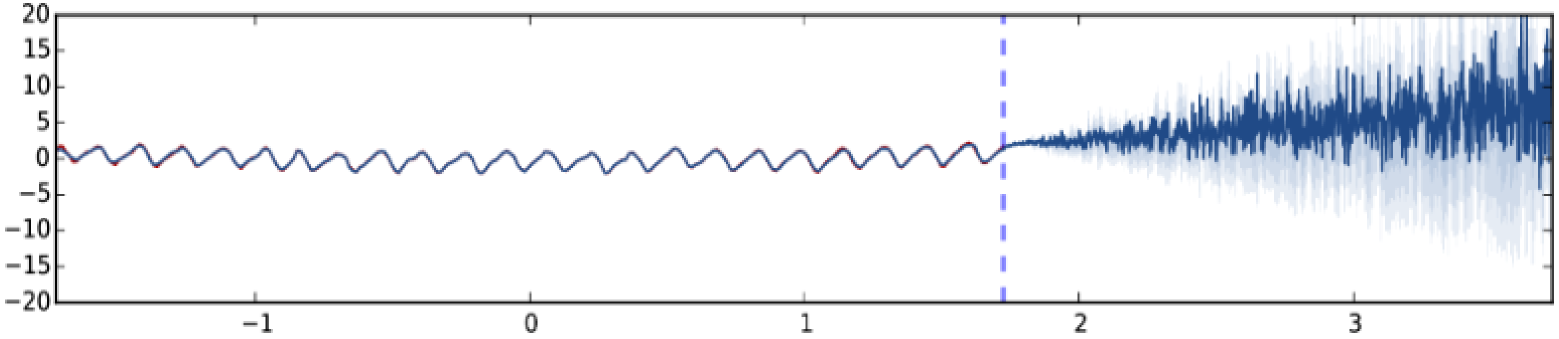
(a) Standard Neural Net

(b) After applying dropout.

Model Uncertainty with Dropout

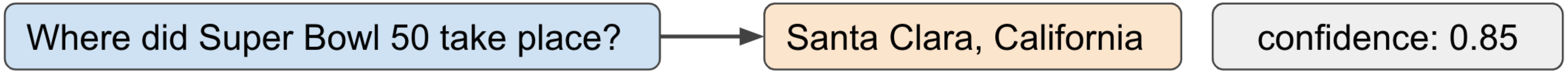
- Sample T dropout masks for model forward passes
- Estimate variance as uncertainty

Uncertainty Visualization



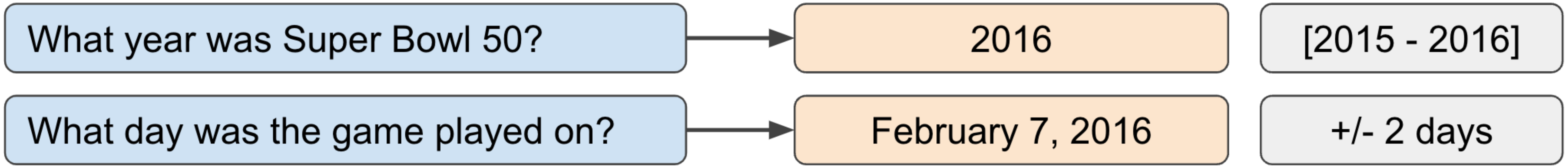
Conformal Prediction

- As a numerical value (e.g. in [0,1]) returned with each prediction:



Where did Super Bowl 50 take place? → Santa Clara, California — confidence: 0.85

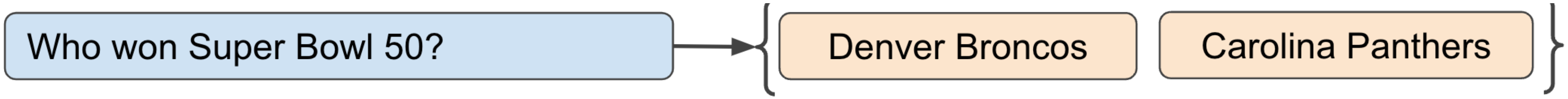
- As a confidence interval around a numerical value:



What year was Super Bowl 50? → 2016 — [2015 - 2016]

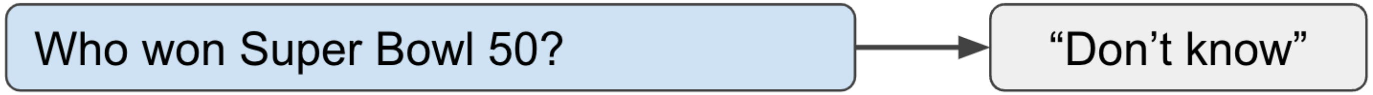
What day was the game played on? → February 7, 2016 — +/- 2 days

- As a set of candidate answers:



Who won Super Bowl 50? → { Denver Broncos, Carolina Panthers }

- As a decision to abstain from answering:



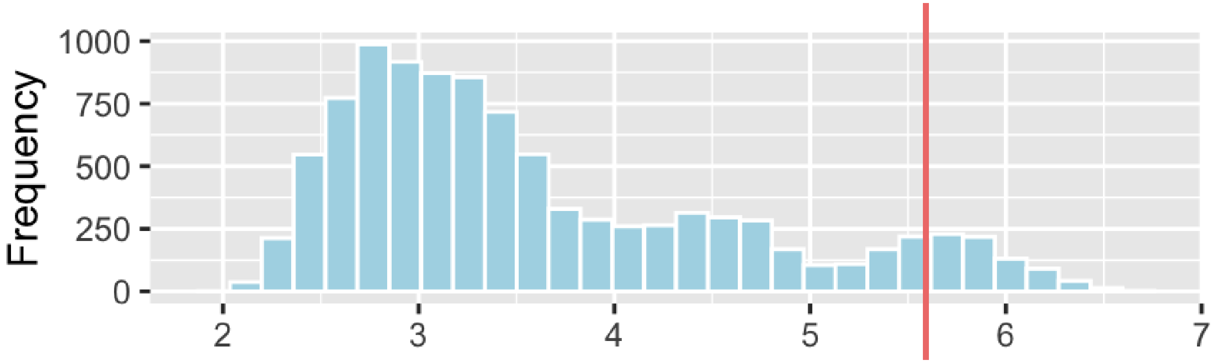
Who won Super Bowl 50? → “Don’t know”

Conformal Prediction

- $x \rightarrow [y_{lower}, y_{upper}]$ with a 95% confidence that the interval will cover true value of y

Inductive Conformal Prediction

- Split the training data into training set and calibration set
- Train a model f with training set
- Predict the examples from the calibration set with f
- Compute nonconformity scores $S_i = |y_i - f(x_i)|$
- Compute nonconformity score distribution
- Get (1-α)-th percentile value q ((1-α)-confidence)
- For a testing example, predict interval $[\tilde{y} - q, \tilde{y} + q]$


Frequency