Carnegie Mellon University Software Engineering Institute

Research Review 2021

Knowing When You Don't Know:

AI Engineering in an Uncertain World

November 2021

Eric Heim, AI Division

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University
Software Engineering Institute

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM21-0875

Quantifying Uncertainty: A Key Component for **informative** and Robust AI Systems

Image: South Carolina National Guard, 151st Signal Battalion

Quantifying Uncertainty: A Key Component for **informative** and Robust AI Systems

Image: South Carolina National Guard, 151st Signal Battalion

Accurate estimates of uncertainty can lead to better informed decision making.

Quantifying Uncertainty: A Key Component for Informative and **Robust** AI Systems

If *Friendly Truck* is detected

↓

Mark Position of *Friendly Truck* on Map

Quantifying Uncertainty: A Key Component for Informative and **Robust** AI Systems

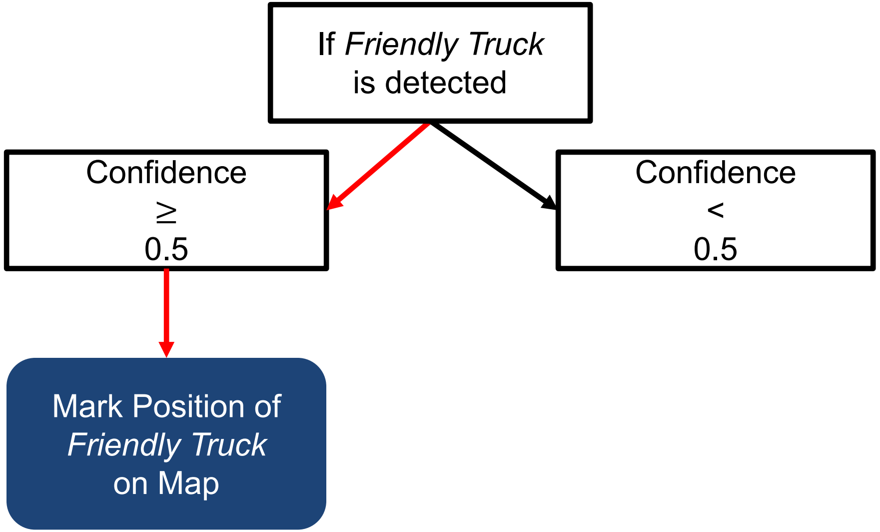
Carnegie Mellon University Software Engineering Institute

If *Friendly Truck* is detected

Confidence ≥ 0.5

Confidence < 0.5

Quantifying Uncertainty: A Key Component for Informative and **Robust** AI Systems



Quantifying Uncertainty: A Key Component for Informative and **Robust** AI Systems

If *Friendly Truck* is detected

Confidence ≥ 0.5

Mark Position of *Friendly Truck* on Map

Confidence < 0.5

Mark Position of *Unknown Vehicle* on Map

Maneuver Robot to gain confidence

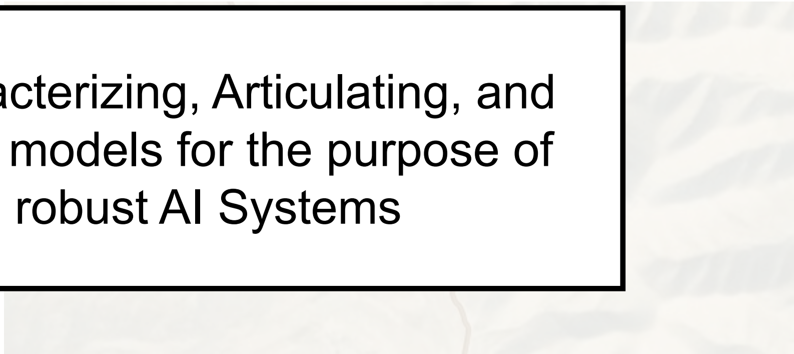
By allowing high-level reasoning to be informed by predictive uncertainty, AI systems can be **more robust** to failures caused by unconfident predictions.

Quantifying Uncertainty: A Key Component for Informative and **Robust** AI Systems

Carnegie Mellon University Software Engineering Institute

Confidence ≥ 0.5

Mark Position Friendly Truc on Map



Our Work: Evaluating, Characterizing, Articulating, and Rectifying Uncertainty in ML models for the purpose of more informative and robust AI Systems

This Talk: Evaluating ML model *calibration*.

g to be nty, AI failures tions.

Maneuver Robot to gain confidence

Calibration: A Way to Interpret Model Uncertainty

Friendly Truck

Enemy Tank

Classifier

Calibration: A Way to Interpret Model Uncertainty

Friendly Truck

Enemy Tank

? → Classifier → (0.6,0.4)

FTr ETa

Calibration: A Way to Interpret Model Uncertainty

Friendly Truck

Enemy Tank

(0.6,0.4)

FTr ETa

How do we
understand these values?

Calibration: A Way to Interpret Model Uncertainty

Friendly Truck

Enemy Tank

<u>Classifier Calibration</u>: Classifier outputs match the frequency of class labels.

Calibration: A Way to Interpret Model Uncertainty

Friendly Truck

Enemy Tank

? → Classifier → (0.6,0.4)

FTr ETa

For all possible inputs that the classifier outputs (0.6,0.4)…
60% of the inputs should be a friendly truck,
40% of the inputs should be an enemy tank.

Classifier Calibration: Classifier outputs match the frequency of class labels.

Evaluating Classifier Calibration

Carnegie Mellon University Software Engineering Institute

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)**

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}

Classifier → (0.6, 0.25, 0.05, 0.1)

FTa FTr ETa ETr

Evaluating Classifier Calibration

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)**

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}

Classifier → (**0.6**, 0.25, 0.05, 0.1)

FTa FTr ETa ETr

Top-1 Expected Calibration Error (ECE)

Considers only the most confident class in evaluating for calibration

For all possible inputs that the classifier outputs 0.6 as the most confident class… 60% of the those inputs should be that class.

Evaluating Classifier Calibration

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE).**

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}

Classifier → (**0.6**, 0.25, **0.05**, 0.1)
FTa FTr ETa ETr

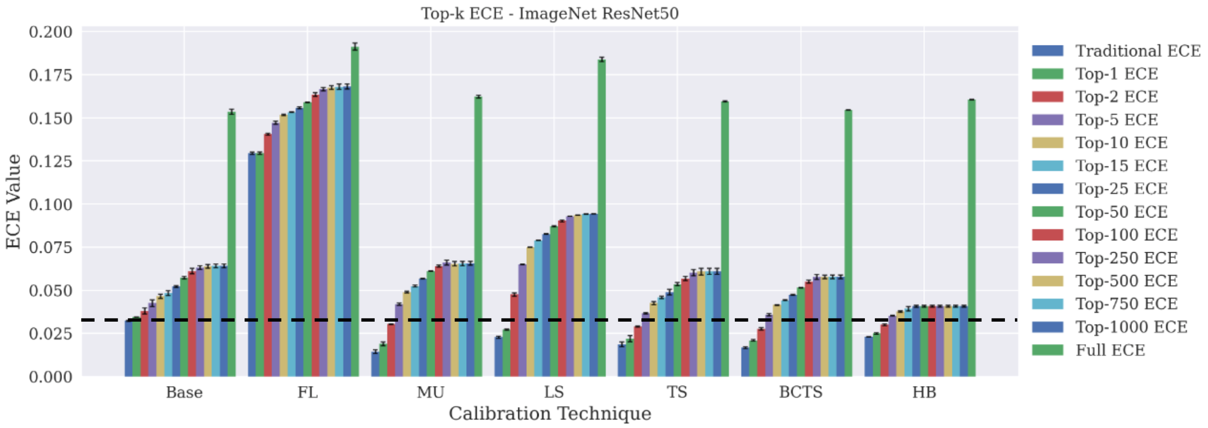
Classifier → (**0.6**, 0.0, **0.40**, 0.0)
FTa FTr ETa ETr

According to Top-1 ECE, these two classifiers ***are considered the same.***
However, the two outputs can mean very different things with ***mission context.***

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes



Traditional ECE – Measures miscalibration for most confident class

<u>Overall Goal</u>: Evaluate the state of the art in classifier calibration according to context focused metrics to observe how they perform in different definitions of reliability.

<u>Experiment #1</u>: Top-k
- Data Set: ImageNet
- Base Model: ResNet50

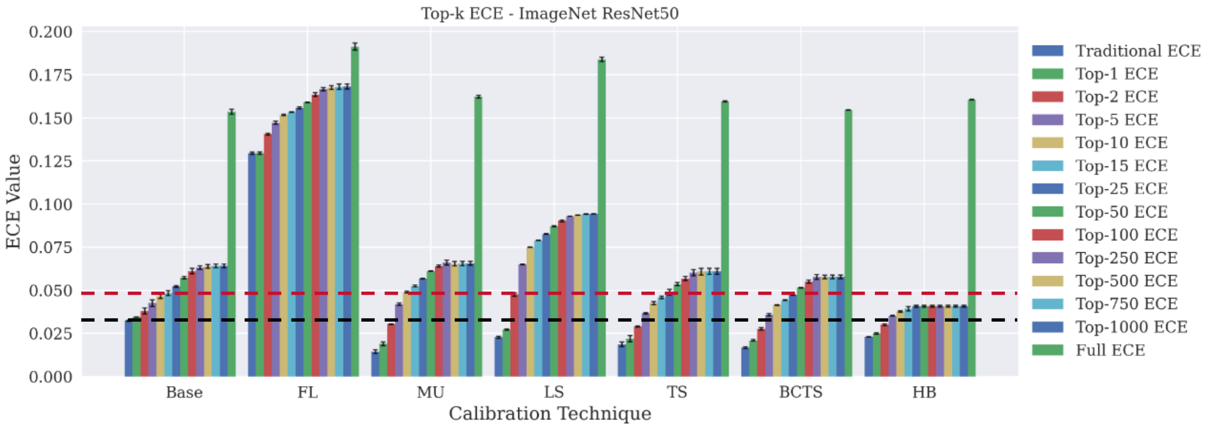
<u>Question</u>: How to these methods perform outside of the most confident class?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Carnegie Mellon University Software Engineering Institute

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes



Top-10 ECE– Measures miscalibration for the 10 most confident classes

Overall Goal: Evaluate the state of the art in classifier calibration according to context focused metrics to observe how they perform in different definitions of reliability.

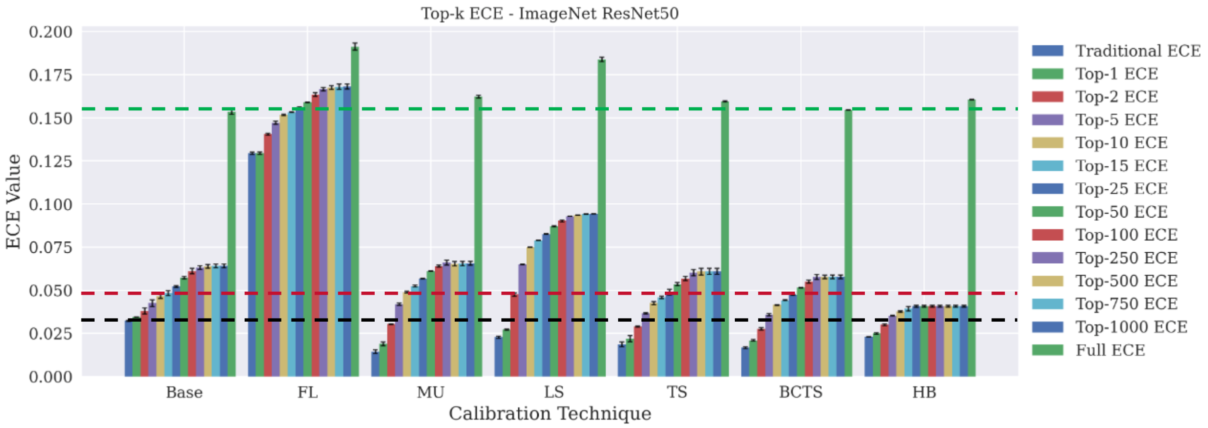
Experiment #1: Top-k
- Data Set: ImageNet
- Base Model: ResNet50

Question: How to these methods perform outside of the most confident class?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., "Friendly" versus "Enemy" vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label "Enemy Vehicle")
3. Subsets of the class probability space between most confident class and all classes



Full ECE– Measures miscalibration across all classes

<u>Overall Goal</u>: Evaluate the state of the art in classifier calibration according to context focused metrics to observe how they perform in different definitions of reliability.

<u>Experiment #1</u>: Top-k
- Data Set: ImageNet
- Base Model: ResNet50

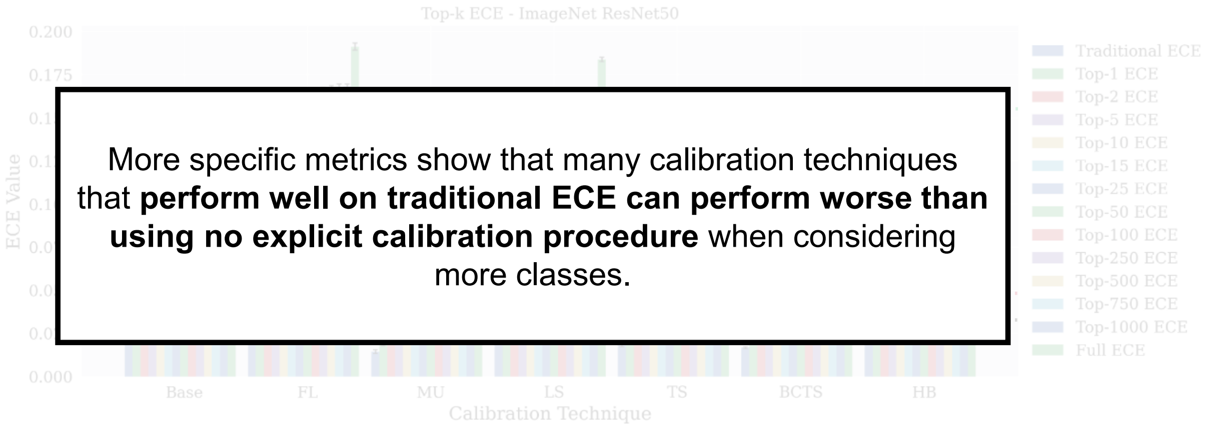
<u>Question</u>: How to these methods perform outside of the most confident class?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Carnegie Mellon University Software Engineering Institute

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes



More specific metrics show that many calibration techniques that **perform well on traditional ECE can perform worse than using no explicit calibration procedure** when considering more classes.

Full ECE– Measures miscalibration across all classes

<u>Overall Goal</u>: Evaluate the state of the art in classifier calibration according to context focused metrics to observe how they perform in different definitions of reliability.

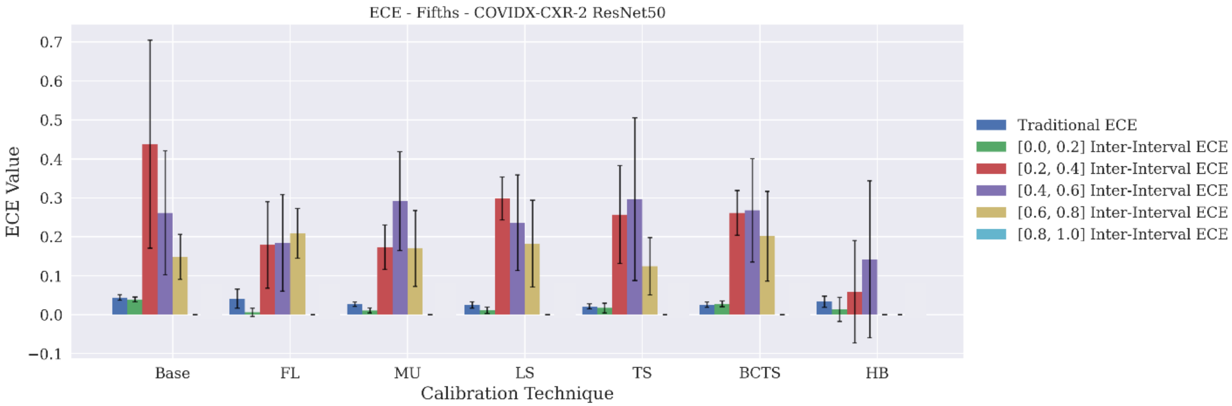
<u>Experiment #1</u>: Top-k
- Data Set: ImageNet
- Base Model: ResNet50

<u>Question</u>: How to these methods perform outside of the most confident class?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes
4. How confidence will be shown to an end user



<u>Experiment #2</u>: Inter-Interval ECE
- Data Set: COVID-CRX-2
- Base Model: ResNet50

<u>Assume</u>:
Confidence will be displayed as to a clinician as one of five categories:
[0.0,0.2] – Very low confidence of COVID
[0.2,0.4] – Low confidence of COVID
…
[0.8,1.0] – Very high confidence of COVID

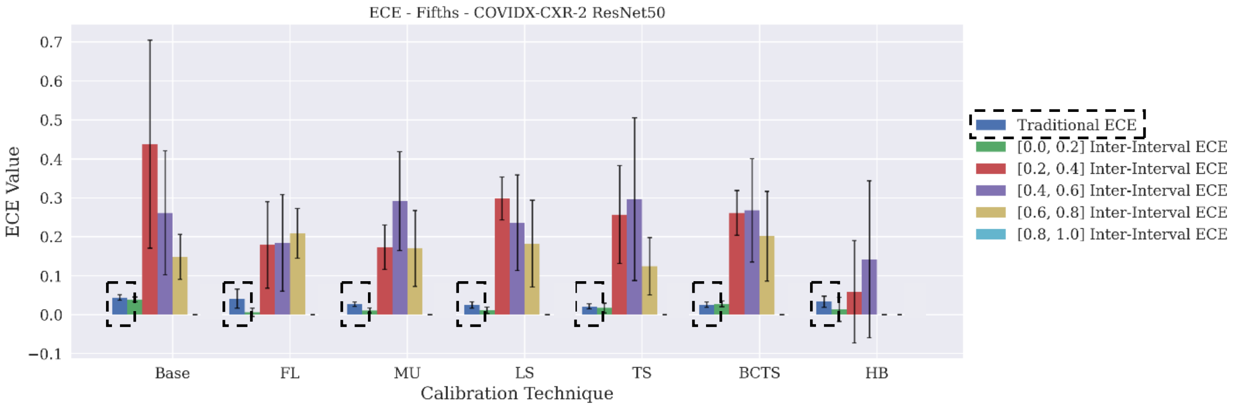
How can we evaluate classifier calibration in this context?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Carnegie Mellon University Software Engineering Institute

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes
4. How confidence will be shown to an end user



Top-1 ECE – Measures miscalibration for most confident class

<u>Experiment #2</u>: Inter-Interval ECE
- Data Set: COVID-CRX-2
- Base Model: ResNet50

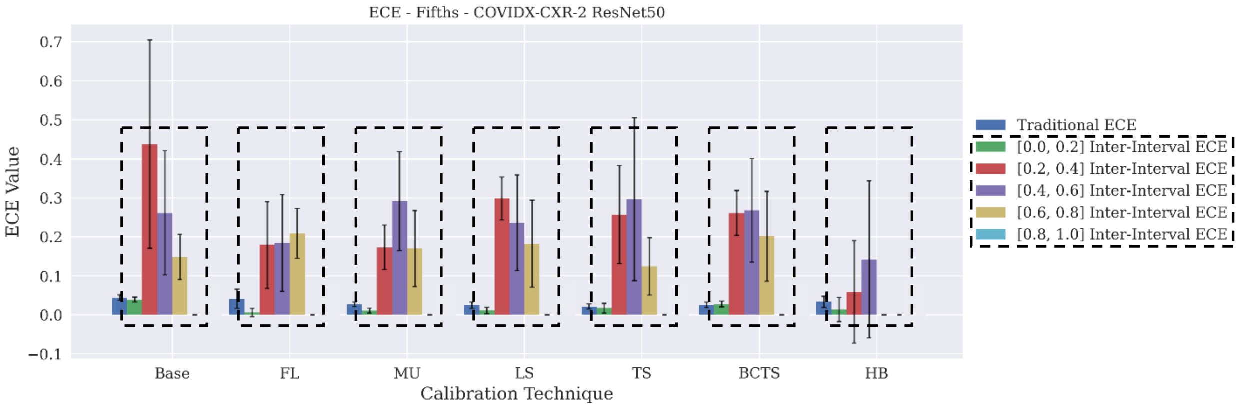
<u>Assume</u>:
Confidence will be displayed as to a clinician as one of five categories:
[0.0,0.2] – Very low confidence of COVID
[0.2,0.4] – Low confidence of COVID
…
[0.8,1.0] – Very high confidence of COVID

How can we evaluate classifier calibration in this context?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes
4. How confidence will be shown to an end user



Inter-Interval ECE– Measures degree of miscalibration with respect to each category

<u>Experiment #2</u>: Inter-Interval ECE

- Data Set: COVID-CRX-2
- Base Model: ResNet50

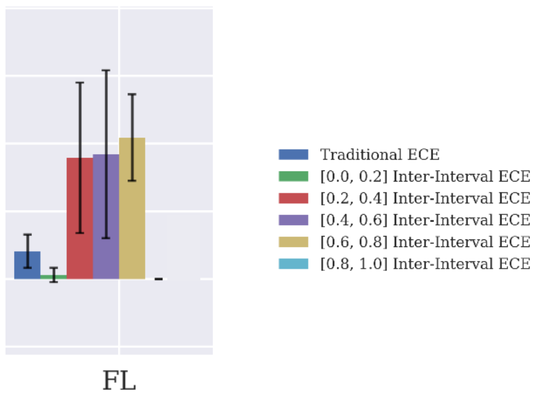
<u>Assume</u>:
Confidence will be displayed as to a clinician as one of five categories:
[0.0,0.2] – Very low confidence of COVID
[0.2,0.4] – Low confidence of COVID
…
[0.8,1.0] – Very high confidence of COVID

How can we evaluate classifier calibration in this context?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes
4. How confidence will be shown to an end user



Inter-Interval ECE– Measures degree of miscalibration with respect to each category

<u>Experiment #2</u>: Inter-Interval ECE
- Data Set: COVID-CRX-2
- Base Model: ResNet50

<u>Assume</u>:
Confidence will be displayed as to a clinician as one of five categories:
[0.0,0.2] – Very low confidence of COVID
[0.2,0.4] – Low confidence of COVID
…
[0.8,1.0] – Very high confidence of COVID

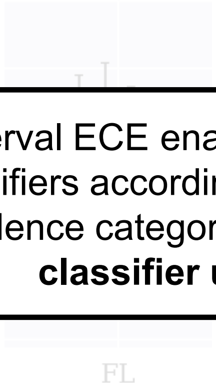
How can we evaluate classifier calibration in this context?

Our Work: Context Focused Calibration Metrics (Kirchenbauer, Oaks, and Heim; 2021 *Under Review*)

Carnegie Mellon University Software Engineering Institute

Using a statistical framing of ECE, we developed a number of metrics that consider these factors:

1. Application-specific tradeoffs between classes (e.g., “Friendly” versus “Enemy” vehicles)
2. Specific instances of interest (e.g., Measuring calibration on instances with label “Enemy Vehicle”)
3. Subsets of the class probability space between most confident class and all classes
4. How confidence will be shown to an end user



Inter-Interval ECE enables evaluation of classifiers according to specified confidence categories that **reflect classifier usage**.

Inter-Interval ECE– Measures degree of miscalibration with respect to each category

<u>Experiment #2</u>: Inter-Interval ECE
- Data Set: COVID-CRX-2
- Base Model: ResNet50

<u>Assume</u>:
Confidence will be displayed as to a clinician as one of five categories:
[0.0,0.2] – Very low confidence of COVID
[0.2,0.4] – Low confidence of COVID
…
[0.8,1.0] – Very high confidence of COVID

How can we evaluate classifier calibration in this context?

Team

Carnegie Mellon University Software Engineering Institute

Eric Heim
Senior ML Researcher
AI Division

John Kirchenbauer
Machine Learning Engineer
AI Division

Jon Helland
Machine Learning Researcher
AI Division

Jacob Oaks
Student Intern
AI Division

Aarti Singh
Associate Professor
Machine Learning Department

Zachary Lipton
Assistant Professor
Machine Learning Department

Final Thoughts

Machine-learned models are are able to express ***uncertainty*** in their predictions that can lead to more informative, robust AI systems by

1. allowing humans to reason about when the model is likely to be incorrect
2. allowing components in a larger system to take different actions based on model confidence

In this project we research methods to evaluate, characterize, articulate and rectify uncertainty

Next steps:
- Develop a demonstration highlighting the utility of accurately expressing uncertainty.
- Create techniques to characterize the cause of uncertainty for a ML model.

For the audience: We are always looking for motivating real-world uses for our work. If you have a need for AI Systems that are able to express and reason under uncertainty, do not hesitate to reach out.

info@sei.cmu.edu