Calibration in Deep Learning: Theory & Practice

Preetum Nakkiran

Aspen Center for Physics | Apple Inc. | Feb 28 2023
Motivation

"Why do we get more than we asked for in Deep Learning?"

- Small Test Loss
- "Good representations"
- Good transfer & fine-tuning
- Good OOD robustness
- Good calibration

Small Train Loss → SGD
Motivation

“Why do we get more than we asked for in Deep Learning?”

- Small Test Loss
- “Good representations”
- Good transfer & fine-tuning
- Good OOD robustness
- Good calibration

Goal: What’s important about this box?
What is Calibration?

**Setting**: Binary classification

Test distribution $D$ over $(x, y) \in \mathcal{X} \times \{0,1\}$

Predictor $f : \mathcal{X} \rightarrow [0,1]$ 

"$f(x)$ is confidence of $y(x) = 1$"

Perfect calibration is a property of the pair $(f, D)$
Calibration: Predictor $f$ is perfectly calibrated w.r.t. distribution $D$ if...
Calibration: Predictor $f$ is perfectly calibrated w.r.t. distribution $D$ if...
**Calibration:** Predictor $f$ is perfectly calibrated w.r.t. distribution $D$ if...
**Calibration**: Predictor $f$ is perfectly calibrated w.r.t. distribution $D$ if...
Perfect Calibration:

Predictor $f$ is perfectly calibrated w.r.t. $D$ if

$$\forall \ell \in [0,1] : \quad \mathbb{E}_{x,y \sim D}[y \mid f(x) = \ell] = \ell$$
What’s calibration good for?

1. **Interpretability**: $f(x)$ is a meaningful quantity, “confidence that $y = 1$”
   e.g. doctor informing patient of “80% probability of heart disease”

2. **Operational Uncertainty**: Systems downstream of $f(x)$ can behave differently on “high confidence” vs. “low confidence” inputs

$$
\begin{align*}
\Pr[y=1 \mid f(x) = 0.5] &= 0.5 \\
\Pr[y=1 \mid f(x) = 0.8] &= 0.8 \\
\Pr[y=1 \mid f(x) = 1.0] &= 1.0 
\end{align*}
$$
Perfect Calibration:

Predictor $f$ is perfectly calibrated w.r.t. $D$ if

$$\forall \ell \in [0,1]: \mathbb{E}_{x,y \sim D}[y \mid f(x) = \ell] = \ell$$

What’s calibration good for?

1. **Interpretability**: $f(x)$ is a meaningful quantity, “confidence that $y = 1$”
   e.g. doctor informing patient of “80% probability of heart disease”

2. **Operational Uncertainty**: Systems downstream of $f(x)$ can behave differently on “high confidence” vs. “low confidence” inputs
   
   $$\begin{align*}
   \text{Pr}[y=1 \mid f(x) = 0.5] &= 0.5 \\
   \text{Pr}[y=1 \mid f(x) = 0.8] &= 0.8 \\
   \text{Pr}[y=1 \mid f(x) = 1.0] &= 1.0
   \end{align*}$$

3. **Interesting**: “Models knows when it doesn’t know” self-consistency
Calibration is **orthogonal** to accuracy

<table>
<thead>
<tr>
<th>input $x$</th>
<th>ground-truth $y$</th>
<th>prediction $f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="image" /></td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image2.png" alt="image" /></td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image3.png" alt="image" /></td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image4.png" alt="image" /></td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image5.png" alt="image" /></td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image6.png" alt="image" /></td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td><img src="image7.png" alt="image" /></td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Goal

Understand when and why (DL) models are well-calibrated (and what factors affect calibration)
This Talk

1. How to Define & Measure Miscalibration


2. Empirical Conjectures

   + upcoming work

3. Theory

   “Loss minimization yields multicalibration for large neural networks.” In submission. [Kalai]
This Talk

1. How to Define & Measure Miscalibration

2. Empirical Conjectures
   + upcoming work

3. Theory
   “Loss minimization yields multicalibration for large neural networks.” In submission. [Kalai]
Part 2. Calibration of DNNs, Experimentally

Can we empirically characterize which DNNs have small calibration error?
No single design choice determines calibration:
- Not just architecture
- Not just model size
- Not just test accuracy/loss

ex: small 3-layer MLP is well-calibrated on ImageNet
Predicting Good Probabilities With Supervised Learning

Alexandra Nicolaccio-Mirizzi
Rich Caruana

[2005]
On Calibration of Modern Neural Networks

Chuan Guo*†  Geoff Pleiss*‡  Yu Sun*‡  Killian Q. Weinberger†

ResNet (2016) CIFAR-100

Predicting Good Probabilities With Supervised Learning

Alexandra Nicolece-Medali
Ricardo Caruana

[2005]

Revisiting the Calibration of Modern Neural Networks

Matthias Minderer
Xiaohua Zhai
Josip Djolonga
Neil Houlsby
Rob Romijnders
Frances Hubschman
Google Research, Brain Team

[2021]

POORLY-CALIBRATED

WELL-CALIBRATED

[2017]
Proposal: Study test-calibration as we study test-error.

Fundamental Decomposition

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

- \( \mu_{\text{Test}} \): Calibration Error on Test Set
- \( \mu_{\text{Train}} \): Calibration Error on Train Set
- \( |\mu_{\text{Test}} - \mu_{\text{Train}}| \): Calibration Generalization Gap
Proposal: Study test-calibration as we study test-error.

Fundamental Decomposition

\[
\mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}|
\]

Trivial, BUT:
1. Suggests methodology: study each part separately
2. Insightful: parts are simpler than the whole
ViT on binary-CIFAR-10

(higher=worse)

Train Error, Test Error, Train ECE, and Test ECE over train time.
End of training: models are overconfident

Train \{Error, CE\} \approx 0
Test \{Error, CE\} \gg 0
ViT on binary-CIFAR-10

End of training: models are overconfident

Train \{\text{Error,CE}\} \approx 0
Test \{\text{Error,CE}\} \gg 0

Throughout training:
Train CE \approx 0

(higher=worse)
Fundamental Decomposition

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

Calibration Error on Test Set \quad Calibration Error on Train Set \quad Calibration Generalization Gap

* depth \( \geq 2 \), trained with proper scoring rule, no severe augmentations, ...
Fundamental Decomposition

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

Calibration Error on Test Set \quad Calibration Error on Train Set \quad Calibration Generalization Gap

Empirical Claim 1

For almost all* DNNs

\[ \mu_{\text{Train}} \approx 0 \]

*depth \geq 2, trained with proper scoring rule, no severe augmentations, ...
Fundamental Decomposition

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

Calibration Error on Test Set \quad \leq \quad \text{Calibration Error on Train Set} \quad + \quad \text{Calibration Generalization Gap}

Empirical Claim 1

For almost all* DNNs

\[ \mu_{\text{Train}} \approx 0 \]

*depth \geq 2, trained with proper scoring rule, no severe augmentations, ...
Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$

Even when underfitting!

*depth $\geq 2$, trained with proper scoring rule, no severe augmentations, ...
Empirical Claim 1
For almost all* DNNs
\[ \mu_{\text{Train}} \approx 0 \]

Even when underfitting!

Empirical Claim 2
For almost all* DNNs
\[ |\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}| \]

* depth \( \geq 2 \), trained with proper scoring rule, no severe augmentations, ...
Empirical Claim 1
For almost all\* DNNs

\[ \mu_{\text{Train}} \approx 0 \]

Even when underfitting!

Empirical Claim 2
For almost all\* DNNs

\[ |\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}| \]
Empirical Claim 1

For almost all* DNNs

\[ \mu_{\text{Train}} \approx 0 \]
Empirical Claim 1

For almost all* DNNs

$$\mu_{\text{Train}} \approx 0$$
**Empirical Claim 2**

For almost all* DNNs

|μ_{Test} − μ_{Train}| ≤ |TestError − TrainError|
(Test Calibration Error) $\leq$ |Train Error - Test Error|

“Models with small generalization-gap are typically well-calibrated”
Takeaways

(\text{Test Calibration Error}) \approx |\text{Train Error - Test Error}|

“\textit{Models with small generalization-gap are typically well-calibrated}”
Takeaways

\[(\text{Test Calibration Error}) \leq |\text{Train Error - Test Error}|\]

“Models with small generalization-gap are typically well-calibrated”

The following are well-calibrated:

1. Small models, on large data-sets (e.g. large vision models)
2. All models trained for 1-epoch (e.g. LLMs)
Applications

For any intervention (changing the augmentation, regularizer, etc), study its effect on:

1. Train calibration
2. Generalization gap
Applications: Regularization Strength

$\mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}|$

Calibration on Test Set  Calibration on Train Set  Calibration Generalization Gap
Applications: Data Augmentation

"Standard" data-augmentation (measure-preserving):

- Same TrainCE; Shrinks generalization gap

"Exotic" data-augmentation:

- Increases TrainCE; Shrinks generalization gap
Applications: Data Augmentation

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

Calibration on Test Set  \quad \text{Calibration on Train Set}  \quad \text{Calibration Generalization Gap}

![Bar chart showing train KCE for different models](image)
Applications: Data Augmentation

\[ \mu_{\text{Test}} \leq \mu_{\text{Train}} + |\mu_{\text{Test}} - \mu_{\text{Train}}| \]

Calibration on Test Set \quad \text{Calibration on Train Set} \quad \text{Calibration Generalization Gap}

![Graphs showing train KCE and KCE Generalization Gap]
Part 3.
Theory

When are Claims 1 & 2 provably true?
Empirical Claim 2

For almost all* DNNs

\[ |\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}| \]
Assumption 1: $f$ is overconfident on test.

$\forall \ell \in [0, 1] : \mathbb{E}_{D_{\text{test}}} [\text{Acc}(f, y) \mid f = \ell] \leq \text{Conf}(v)$
Assumption 1: \( f \) is overconfident on test.

\[
\forall \ell \in [0, 1] : \mathbb{E}_{D_{test}} \left[ \text{Acc}(f, y) \mid f = \ell \right] \leq \text{Conf}(v)
\]

Assumption 2: \( f \) is more confident on train than on test.

\[
\mathbb{E}_{D_{train}} \left[ \text{Conf}(f) \right] \geq \mathbb{E}_{D_{test}} \left[ \text{Conf}(f) \right]
\]

Empirical Claim 2

For almost all* DNNs

\[
\left| \mu_{Test} - \mu_{Train} \right| \leq \left| \text{TestError} - \text{TrainError} \right|
\]
**Empirical Claim 2**

For almost all* DNNs

\[ |\mu_{\text{Test}} - \mu_{\text{Train}}| \leq |\text{TestError} - \text{TrainError}| \]

**Assumption 1:** \( f \) is overconfident on test.

\[ \forall \ell \in [0, 1] : \mathbb{E}_{D_{\text{test}}} [\text{Acc}(f, y) \mid f = \ell] \leq \text{Conf}(\nu) \]

**Assumption 2:** \( f \) is more confident on train than on test.

\[ \mathbb{E}_{D_{\text{train}}} [\text{Conf}(f)] \geq \mathbb{E}_{D_{\text{test}}} [\text{Conf}(f)] \]

**Theorem 2** (Calibration Generalization Bound). *Under Assumptions 2 and 3, we have*

\[ \text{ECE}(D_{\text{test}}) - \text{ECE}(D_{\text{train}}) \leq \text{Error}(D_{\text{test}}) - \text{Error}(D_{\text{train}}) \]
Empirical Claim 1

For almost all* DNNs

\[ \mu_{\text{Train}} \approx 0 \]
Given: Distribution $D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]}$
Given: Distribution $D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we'd like to do:

*Exactly minimize* expected loss, over *all functions*:

$$f^* = \text{argmin}_{f: \mathcal{X} \to [0,1]} \mathbb{E}_{x, y \sim D} [\ell(f(x), y)]$$
Given: Distribution $D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]}$

What we’d like to do:

*Exactly minimize* expected loss, over *all functions*:

$$f^* = \arg\min_{f: \mathcal{X} \to [0,1]} \mathbb{E}_{x,y \sim D} [\ell(f(x), y)]$$

$$\implies f^*(x) = p_D(y|x)$$

perfectly calibrated
What we actually do:

Run SGD* to \textit{approximately minimize} expected loss, over \textit{restricted family} \( \{f_\theta : \theta \in \Theta\} \):

\[
\tilde{f} = \text{SGDmin}_{\theta \in \Theta} \mathbb{E}_{x,y \sim \mathcal{D}} [\ell(f_\theta(x), y)]
\]

What we’d like to do:

\textit{Exactly minimize} expected loss, over \textit{all functions}:

\[
f^* = \text{argmin}_{f : \mathcal{X} \to [0,1]} \mathbb{E}_{x,y \sim \mathcal{D}} [\ell(f(x), y)]
\]

\Rightarrow \quad f^*(x) = p_\mathcal{D}(y|x)

perfectly calibrated

Given: Distribution \( D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]} \)
What we actually do:

Run SGD* to \emph{approximately minimize} expected loss, over \emph{restricted family} \( \{f_\theta : \theta \in \Theta\} \):

\[
\tilde{f} = \text{SGDmin}_{\theta \in \Theta} \mathbb{E}_{x,y \sim D} [\ell(f_\theta(x), y)]
\]

What we’d like to do:

\emph{Exactly minimize} expected loss, over \emph{all functions}:

\[
f^* = \text{argmin}_{f : \mathcal{X} \to [0,1]} \mathbb{E}_{x,y \sim \mathcal{D}} [\ell(f(x), y)]
\]

\[
\implies f^*(x) = p_D(y|x)
\]

perfectly calibrated

Given: Distribution \( D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]} \)

\[
\text{Perfectly calibrated (Bayes-optimal)}
\]

\[
\text{Near-perfectly calibrated}
\]
What we actually do:
Run SGD* to *approximately minimize* expected loss, over restricted family \{f_\theta : \theta \in \Theta\}:

\[
\tilde{f} = \text{SGDmin}_{\theta \in \Theta} \mathbb{E}_{x,y \sim D} [\ell(f_\theta(x), y)]
\]

What we'd like to do:
Exactly minimize expected loss, over all functions:

\[
f^* = \text{argmin}_{f : \mathcal{X} \to [0,1]} \mathbb{E}_{x,y \sim D} [\ell(f(x), y)]
\]

\[\Rightarrow f^*(x) = p_D(y|x)\]
perfectly calibrated

Given: Distribution \(D = \hat{D} = \{(x_i, y_i)\}_{i \in [n]}\)

what's special about this point?
In general, when does:

\textit{suboptimal} loss-minimization \implies \textit{near-optimal} calibration?
For all $f, D$, and proper loss $\ell$, TFAE:

1. $f$ is perfectly calibrated w.r.t. $D$

2. The loss of $f : \mathcal{X} \to [0,1]$ on $D$ cannot be improved by post-processing $\kappa : [0,1] \to [0,1]$

   \[ \forall \kappa : [0,1] \to [0,1], \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f) \]

   where $\mathcal{L}_D(f) := \mathbb{E}_{x,y \sim D}[\ell(f(x), y)]$ is the expected loss
For all $f, D$, and proper loss $\ell$, TFAE:

1. $f$ is perfectly calibrated w.r.t. $D$

2. The loss of $f : \mathcal{X} \rightarrow [0,1]$ on $D$ cannot be improved by post-processing $\kappa : [0,1] \rightarrow [0,1]$

   $$\forall \kappa : [0,1] \rightarrow [0,1], \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f)$$

where $\mathcal{L}_D(f) := \mathbb{E}_{x,y \sim D}[\ell(f(x), y)]$ is the expected loss.
For all $f, D$, and proper loss $\ell$, TFAE:

1. $f$ is perfectly calibrated w.r.t. $D$

2. The loss of $f : \mathcal{X} \to [0,1]$ on $D$ cannot be improved by post-processing $\kappa : [0,1] \to [0,1]$

   \[ \forall \kappa : [0,1] \to [0,1], \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f) \]

   where $\mathcal{L}_D(f) := \mathbb{E}_{x,y \sim D} [\ell(f(x), y)]$ is the expected loss.
For all \( f, D \), and proper loss \( \ell \), TFAE:

1. \( f \) is perfectly calibrated w.r.t. \( D \)
2. The loss of \( f \) on \( D \) cannot be improved by post-processing:
   \[
   \forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f)
   \]

Suggestive properties:

1. Requires only "weak local-optimality", not global optimality
2. Post-processing can be represented by adding a layer
For all $f, D$, and proper loss $\ell$, TFAE:

1. $f$ is perfectly calibrated w.r.t. $D$

2. The loss of $f$ on $D$ cannot be improved by post-processing:

$$\forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f)$$

Suggestive properties:

1. Requires only "weak local-optimality", not global optimality

2. Post-processing can be represented by adding a layer
For all \( f, D, \) and proper loss \( \mathcal{L} \), TFAE:

1. \( f \) is perfectly calibrated w.r.t. \( D \)
2. The loss of \( f \) on \( D \) cannot be improved by post-processing:
   \[
   \forall \kappa : \mathbb{R} \rightarrow \mathbb{R}, \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f)
   \]

Suggestive properties:

1. Requires only “weak local-optimality”, not global optimality
2. Post-processing can be represented by adding a layer
For all $f, D$, and proper loss $\ell$, TFAE:

1. $f$ is perfectly calibrated w.r.t. $D$

2. The loss of $f$ on $D$ cannot be improved by post-processing:

   \[ \forall \kappa : \mathbb{R} \to \mathbb{R}, \quad \mathcal{L}_D(f) \leq \mathcal{L}_D(\kappa \circ f) \]

**Problems:**

1. Only characterizes perfect calibration

2. Requires composition with arbitrary functions (not just “nice” ones that can be represented by NNs)
Toy Theorem

“$f$ is perfectly calibrated iff its loss can’t be improved at all by post-processing with an arbitrary function”
| **Toy Theorem** | “$f$ is perfectly calibrated iff its loss can’t be improved at all by post-processing with an arbitrary function” |
| **Dream Theorem** | “$f$ is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function” |
Toy Theorem

“\( f \) is perfectly calibrated iff its loss can’t be improved at all by post-processing with an arbitrary function”

Dream Theorem

“\( f \) is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function”

How to formalize “close to”? Calibration distance \( d_{\text{CE}}(f) \)!
<table>
<thead>
<tr>
<th>Toy Theorem</th>
<th>&quot;$f$ is perfectly calibrated iff its loss can’t be improved at all by post-processing with an arbitrary function&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dream Theorem</td>
<td>&quot;$f$ is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function&quot;</td>
</tr>
</tbody>
</table>

How to formalize "close to"? Calibration distance $d_{CE}(f)$!
Dream Theorem

“$f$ is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function”
Dream Theorem

“\( f \) is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function”

Theorem

\((\text{distance from calibration}) \sim \text{poly(\text{potential post-processing improvement})}\)
Dream Theorem | “f is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function”

Theorem | (distance from calibration) ~ poly(potential post-processing improvement)
Dream Theorem

“\( f \) is close to calibrated iff its loss can’t be improved much by post-processing with a smooth function”

Theorem

\[(\text{distance from calibration}) \sim \text{poly(\text{potential post-processing improvement})}\]

**Theorem 1.3.** There exist constants \( c_1, c_2 > 0 \) such that for all predictors \( f : \mathcal{X} \to [0, 1] \) and all distributions \( D \), the following holds.

Let \( K \) denote the family of all post-processing functions \( \kappa : [0, 1] \to [0, 1] \) such that the update function \( \eta(f) = \kappa(f) - f \) is 1-Lipschitz. Define the “gap calibration error” of \( f \) as the maximum improvement in MSE loss via post-processings in \( K \):

\[
gapCE(f) = \operatorname{MSE}_D(f) - \min_{\kappa \in K} \operatorname{MSE}_D(\kappa \circ f).
\]

Then, the maximum loss improvement (\( \text{gapCE} \)) polynomially bounds the distance from calibration (\( \text{dCE} \)):

\[
c_1 \ \text{dCE}(f)^4 \leq \text{gapCE}(f) \leq c_2 \ \text{dCE}(f).
\]
When is the “no loss improvement” condition satisfied?

1. (Algorithmic assumption): If it were possible to improve loss via a simple post-processing, SGD would have done it already.

\[ f \mapsto \kappa \circ f \] is a “simple” update for SGD on deep nets
When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already
2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already

---

When is the “no loss improvement” condition satisfied?
When is the “no loss improvement” condition satisfied?

2. **(Human-in-loop assumption):**
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already.
When is the “no loss improvement” condition satisfied?

2. (Human-in-loop assumption):
   If it were possible to add a layer, train it optimally, and improve the loss, then the human trainer would have done it already \( \rightarrow \) output of human is “nearly post-processing optimal”
When is the “no loss improvement” condition satisfied?

3. (Theory assumption):
   Structural risk minimization with any “well-behaved” complexity measure

\[ \min_{f \in \mathcal{F}} \text{MSE}_D(f) + \lambda \mu(f). \]
Implications

• Generic characterization of when (sub-optimal) loss-minimization yields (near-optimal) calibration

• Importance of depth for calibration

• Importance of proper scoring rules for calibration

• Non-Baysean reasons for calibration
Q: What’s important about this box?

A: Output is (nearly) post-processing-optimal w.r.t. loss
## Thanks!

### In Collaboration With

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parikshit Gopalan</td>
<td>Apple</td>
</tr>
<tr>
<td>Vimal Thilak</td>
<td>Apple</td>
</tr>
<tr>
<td>Omid Saremi</td>
<td>Apple</td>
</tr>
<tr>
<td>Joshua Suskind</td>
<td>Apple</td>
</tr>
<tr>
<td>Jarosław Błasiok</td>
<td>Columbia</td>
</tr>
<tr>
<td>Annabelle Carrell</td>
<td>Cambridge, Apple intern</td>
</tr>
<tr>
<td>Lunjia Hu</td>
<td>Stanford, Apple intern</td>
</tr>
<tr>
<td>Elan Rosenfeld</td>
<td>CMU, Apple intern</td>
</tr>
</tbody>
</table>
Defining “Almost All”

Definition-by-example:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data distribution</td>
<td>Any</td>
</tr>
<tr>
<td>Architecture</td>
<td>Any* (MLP, ConvNet, Transformer,…</td>
</tr>
<tr>
<td>Model depth</td>
<td>≥ 2</td>
</tr>
<tr>
<td>Model width</td>
<td>Any* (≥ 100)</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Any* SGD-variant (SGD, Adam, …)</td>
</tr>
<tr>
<td>Optimization steps</td>
<td>Any* (≥ 10, after “warm-up” period)</td>
</tr>
<tr>
<td>Sample size</td>
<td>Any</td>
</tr>
<tr>
<td>Data-aug</td>
<td>None, or “standard” (measure-preserving)</td>
</tr>
<tr>
<td>Loss function</td>
<td>Any proper scoring rule (MSE, xent, …)</td>
</tr>
<tr>
<td>Regularization</td>
<td>None, or very weak (e.g. wd=1-e4)</td>
</tr>
</tbody>
</table>
Empirical Claim 1:
For almost all* ML models

\[ \mu_{\text{Train}} \approx 0 \]
Empirical Claim 1:
For almost all* ML models

\[ \mu_{\text{Train}} \approx 0 \]

**Surprising:** small DNNs, with high \textit{train error}, have good train calibration.

(ResNets on binary-CIFAR-10)
Part 1. 
Measuring Miscalibration

Most models aren't *perfectly calibrated*. 
How do we measure *degree-of-miscalibration*?
Summary: How to Measure Miscalibration [Błasiok, Gopalan, Hu, N. STOC 2023]
Most models aren't *perfectly calibrated*. How do we measure *degree-of-miscalibration*?

Desire:

Function $\mu_D(f) \in [0, \infty)$ that measures "degree of miscalibration"
Summary: How to Measure Miscalibration  [Błasiok, Gopalan, Hu, N. STOC 2023]

Most models aren’t perfectly calibrated. How do we measure degree-of-miscalibration?

DON’T:

• Use “Expected Calibration Error (ECE)”

\[
\text{ECE}(f) = \mathbb{E}[| \mathbb{E}[y | f(x)] - f(x) |]
\]

ECE\((f)\) is discontinuous in \(f\)!
**Summary: How to Measure Miscalibration**  
[Błasiok, Gopalan, Hu, N. STOC 2023]

Most models aren’t *perfectly calibrated*. How do we measure degree-of-miscalibration?

**DON’T:**

- Use “Expected Calibration Error (ECE)”

  \[
  \text{ECE}(f) = \mathbb{E}[|\mathbb{E}[y | f(x)] - f(x)|]
  \]

  ECE\((f)\) is discontinuous in \(f\)!

**DO:**

- Use “\(\ell_1\) distance from perfect calibration”

\[
\text{dCE}(f)
\]

set of perfectly-calibrated functions
Summary: How to Measure Miscalibration

Most models aren’t perfectly calibrated. How do we measure degree-of-miscalibration?

DON’T:

• Use “Expected Calibration Error (ECE)”

\[ ECE(f) = \mathbb{E}[|\mathbb{E}[y | f(x)] - f(x)|] \]

ECE(f) is discontinuous in f!

DO:

• Use “\(\ell_1\) distance from perfect calibration”

\[ dCE(f) \]

set of perfectly-calibrated functions

• Estimate with a “consistent calibration metric”
  e.g. Kernel calibration error (kCE)

\[ kCE_D(f) := \sup_{w:||w||_K \leq 1} \mathbb{E}_{(f,y) \sim \mathcal{D}_f} [w(f)(y - f)] \]
Measuring Miscalibration

Most models aren’t perfectly calibrated.

How to measure degree-of-miscalibration?

Many proposed measures are problematic. Eg, ECE:

\[
ECE(f) = \mathbb{E}[ | \mathbb{E}[y | f(x)] - f(x) | ]
\]
Problem: \( \text{ECE}(f) \) is discontinuous in \( f \)

1. \( \| f_1 - f_2 \| \leq \epsilon \)

2. \( \text{ECE}(f_1) - \text{ECE}(f_2) \geq 0.5 - \epsilon \)
Problem: $ECE(f)$ is discontinuous in $f$

1. $\|f_1 - f_2\| \leq \varepsilon$

2. $ECE(f_1) - ECE(f_2) \geq 0.5 - \varepsilon$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$f_1(x)$</th>
<th>$f_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5+\varepsilon</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5+\varepsilon</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5+\varepsilon</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5-\varepsilon</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5-\varepsilon</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5-\varepsilon</td>
</tr>
</tbody>
</table>

$ECE(f_1) = 0$  $ECE(f_2) \approx 0.5$

$ECE(f) = \mathbb{E}[|\mathbb{E}[y | f(x)] - f(x)|]$
Problem: \( \text{ECE}(f) \) is discontinuous in \( f \)

1. \( \| f_1 - f_2 \| \leq \varepsilon \)

2. \( \text{ECE}(f_1) - \text{ECE}(f_2) \geq 0.5 - \varepsilon \)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>( f_1(x) )</th>
<th>( f_2(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5 + \varepsilon</td>
<td>0.5 + \varepsilon</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5 + \varepsilon</td>
<td>0.5 + \varepsilon</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.5 - \varepsilon</td>
<td>0.5 - \varepsilon</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.5 - \varepsilon</td>
<td>0.5 - \varepsilon</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.5 - \varepsilon</td>
<td>0.5 - \varepsilon</td>
</tr>
</tbody>
</table>

\[ \text{ECE}(f_1) = 0 \quad \text{ECE}(f_2) \approx 0.5 \]

\[ \text{ECE}(f) = \mathbb{E}[|\mathbb{E}[y \mid f(x)] - f(x)|] \]
Axiomatic construction of degree-of-miscalibration $\mu_D(f)$?

Want $\mu(f) \in \mathbb{R}_{\geq 0}$ to satisfy:

1. Correctness:
   $\mu(f) = 0 \iff f$ is perfectly calibrated

2. $\mu(f)$ is continuous in $f$

3. Can be estimated from samples
Axiomatic construction of **degree-of-miscalibration** $\mu_D(f)$?

Want $\mu(f) \in \mathbb{R}_{\geq 0}$ to satisfy:

1. **Correctness:**
   
   $\mu(f) = 0 \iff f$ is perfectly calibrated

2. $\mu(f)$ is continuous in $f$

3. Can be estimated from samples

<table>
<thead>
<tr>
<th></th>
<th>Correctness</th>
<th>Continuity</th>
<th>Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECE</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Binned-ECE</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Brier</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NLL</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NCE</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>kCE/MMCE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>smCE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Correctness:
\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):
\[ \mu(f) \text{ is "close" to } 0 \iff f \text{ is "close" to perfectly calibrated} \]
Correctness:
\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):
\[ \mu(f) \text{ is "close" to 0} \iff f \text{ is "close" to perfectly calibrated} \]

\[ \mathcal{G} := \text{set of perfectly-calibrated functions} \]
Correctness:

\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):

\[ \mu(f) \text{ is "close" to } 0 \iff f \text{ is "close" to perfectly calibrated} \]

\[ \mathcal{G} := \text{set of perfectly-calibrated functions} \]
Correctness:
\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):
\[ \mu(f) \text{ is "close" to } 0 \iff f \text{ is "close" to perfectly calibrated} \]

\[ \mathcal{G} := \text{set of perfectly-calibrated functions} \]
Correctness:
\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):
\[ \mu(f) \text{ is "close" to } 0 \iff f \text{ is "close" to perfectly calibrated} \]

\[ \mathcal{G} := \text{set of perfectly-calibrated functions} \]

The Calibration Distance
\[ d_{\text{CE}}(f) := \min_{g \in \mathcal{G}} d_1(f, g) \]
Correctness:
\[ \mu(f) = 0 \iff f \text{ is perfectly calibrated} \]

Robust Correctness (informally):
\[ \mu(f) \text{ is "close" to } 0 \iff f \text{ is "close" to perfectly calibrated} \]

Robust Correctness:
\[ dCE(f)^\beta \leq \mu(f) \leq dCE(f)^\alpha \]

The Calibration Distance
\[ dCE(f) := \min_{g \in \mathcal{G}} d_1(f, g) \]
Why not use $dCE(f)$ as calibration measure $\mu$?

Satisfies robust completeness:

$\mu(f)$ is "close" to 0 $\iff$ $f$ is "close" to perfectly calibrated

The Calibration Distance

$$dCE(f) := \min_{g \in \mathcal{G}} d_1(f, g)$$
Why not use \( dCE(f) \) as calibration measure \( \mu \)?

Satisfies robust completeness:

\[ \mu(f) \text{ is "close" to 0 } \iff \text{ } f \text{ is "close" to perfectly calibrated} \]

**Q: How to estimate from samples \( \{(f(x_i), y_i)\} \)?**

- Both info-theoretic, and computational issues...

---

**The Calibration Distance**

\[ dCE(f) := \min_{g \in \mathcal{G}} d_1(f, g) \]
New metric $dCE(f)$ intimately related to existing metrics:

- $kCE(f)$ : kernel calibration / MMCE [Kumar Sarawagi Jain 2018]
- $smCE(f)$ : smooth calibration [Foster Hart 2018]
- $intCE(f)$ : interval calibration
New metric $dCE(f)$ intimately related to existing metrics:

- $kCE(f)$: kernel calibration / MMCE [Kumar Sarawagi Jain 2018]
- $smCE(f)$: smooth calibration [Foster Hart 2018]
- $intCE(f)$: interval calibration

**Theorem:** For $\mu \in \{kCE, smCE, intCE\}$:

$$\mu^2 \leq dCE \leq \mu^{1/3}$$
Unification

New metric $\text{dCE}(f)$ intimately related to existing metrics:

- $\text{kCE}(f)$ : kernel calibration / MMCE  [Kumar Sarawagi Jain 2018]
- $\text{smCE}(f)$ : smooth calibration  [Foster Hart 2018]
- $\text{intCE}(f)$ : interval calibration

**Theorem:** For $\mu \in \{ \text{kCE}, \text{smCE}, \text{intCE} \}$:

$$\mu^2 \leq \text{dCE} \leq \mu^{1/3}$$

**Takeaway:**

1. Estimate $\text{dCE}$ from samples
2. Prior metrics are related
Practical Takeaways

Measure calibration with either:

1. Kernel Calibration Error

2. Interval Calibration Error (modification of binnedECE)
Kernel Calibration Error

$$ECE_D(f) := \sup_{w:[0,1] \to [-1,1]} \mathbb{E}_{(f,y) \sim \mathcal{D}_f} [w(f)(y - f)]$$
Kernel Calibration Error

$$ECE_D(f) := \sup_{w: [0,1] \rightarrow [-1,1]} \mathbb{E}_{(f,y) \sim D_f} [w(f)(y - f)]$$

"Residual": \( r(f) := \mathbb{E}[y \mid f] - f \)
Kernel Calibration Error

$$E_{CE_D}(f) := \sup_{w: [0,1] \rightarrow [-1,1]} \mathbb{E}_{(f,y) \sim D_f} [w(f)(y - f)]$$

"Residual": $$r(f) := \mathbb{E}[y | f] - f$$
Kernel Calibration Error

\[
ECE_D(f) := \sup_{w: [0,1] \rightarrow [-1,1]} \mathbb{E}_{(f,y) \sim D_f} [w(f)(y - f)]
\]

"Residual": 
\[
r(f) := \mathbb{E}[y \mid f] - f
\]
Kernel Calibration Error

\[ ECE_D(f) := \sup_{w: [0,1] \to [-1,1]} \mathbb{E}_{(f,y) \sim D_f} \left[ w(f)(y - f) \right] \]

\[ kCE_D(f) := \sup_{w: \|w\| \leq 1} \mathbb{E}_{(f,y) \sim D_f} \left[ w(f)(y - f) \right] \]

“Residual”: \( r(f) := \mathbb{E}[y | f] - f \)
Kernel Calibration Error: Sample Estimation

Given: Samples \((f(x_i), y_i) =: (f_i, y_i)\)

Residuals: \((f_i, r_i)\) for \(r_i := (y_i - f_i)\)
Kernel Calibration Error: Sample Estimation

Given: Samples \((f(x_i), y_i) =: (f_i, y_i)\)

Residuals: \((f_i, r_i)\) for \(r_i := (y_i - f_i)\)

\[
\hat{\text{kCE}_D}(f) = \sqrt{\frac{1}{n^2} \sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}
\]
Kernel Calibration Error: Sample Estimation

Given: Samples \((f(x_i), y_i) =: (f_i, y_i)\)

Residuals: \((f_i, r_i)\) for \(r_i := (y_i - f_i)\)

\[
\widehat{\text{kCE}_D}(f) = \sqrt{\frac{1}{n^2} \sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}
\]
Kernel Calibration Error: Sample Estimation

Given: Samples \((f(x_i), y_i) =: (f_i, y_i)\)

Residuals: \((f_i, r_i)\) for \(r_i := (y_i - f_i)\)

\[
\hat{\text{kCED}}(f) = \sqrt{\frac{1}{n^2} \sum_{i,j} r_i r_j K(f_i, f_j)} = \|r\|_{K(f,f)}
\]

\[
= \sqrt{\langle r, \text{smooth}_K(r) \rangle}
\]
Kernel Calibration Error: Sample Estimation

Given: Samples $(f(x_i), y_i) =: (f_i, y_i)$

Residuals: $(f_i, r_i)$ for $r_i := (y_i - f_i)$

$$\hat{KCE}_D(f) = \sqrt{\frac{1}{n^2} \sum_{i,j} r_i r_j K(f_i, f_j) = \|r\|_{K(f,f)}}$$

- Linear-time estimation: sub-sample $\Theta(n)$ terms
- Requires Laplace kernel
Interval Calibration

\[ \text{binnedECE}(f, \mathcal{I}) := \text{ECE}(\text{round}_\mathcal{I}(f)) \]

binnedECE: Unclear how to choose bins (any fixed choice violates continuity & correctness)
Interval Calibration

\[ \text{binnedECE}(f, \mathcal{I}) := \text{ECE}(\text{round}_{\mathcal{I}}(f)) \]

\[ \text{binnedECE}: \text{Unclear how to choose bins (any fixed choice violates continuity & correctness)} \]

But, adding a “width regularizer” guarantees upper-bound. For all interval-partitions:

\[ \text{dCE}(f) \leq \text{binnedECE}(f, \mathcal{I}) + \text{width}(\mathcal{I}) \]
Interval Calibration

$$dCE(f) \leq \text{binnedECE}(f, I) + \text{width}(I)$$
Interval Calibration

\[ dCE(f) \leq \text{binnedECE}(f, I) + \text{width}(I) \]

Best-possible upper-bound:

\[ \text{intCE}(f) := \inf_{I: \text{Interval partition}} (\text{binnedECE}(f, I) + \text{width}(I)) \]
Interval Calibration

\[ dCE(f) \leq \text{binnedECE}(f, I) + \text{width}(I) \]

Best-possible upper-bound:

\[ \text{intCE}(f) := \inf_{I: \text{Interval partition}} \left( \text{binnedECE}(f, I) + \text{width}(I) \right) \]

Can we get a lower-bound?
Interval Calibration

\[ dCE(f) \leq \text{binnedECE}(f, \mathcal{I}) + \text{width}(\mathcal{I}) \]

Best-possible upper-bound:

\[ \text{intCE}(f) := \inf_{\mathcal{I}: \text{Interval partition}} (\text{binnedECE}(f, \mathcal{I}) + \text{width}(\mathcal{I})) \]

Can we get a lower-bound?

\[ \frac{1}{16} \text{intCE}(f)^2 \leq dCE(f) \leq \text{intCE}(f) \]
Interval Calibration

\[ \text{intCE}(f) := \inf_{\mathcal{I}: \text{Interval partition}} (\text{binnedECE}(f, \mathcal{I}) + \text{width}(\mathcal{I})) \]

Computationally, sufficient to minimize over \( i \in \mathbb{N} \) :

1. Construct regular intervals of width \( 2^{-i} \)
2. Randomly shift intervals (together)
3. Compute \( \text{binnedECE}(f, \mathcal{I}) + \text{width}(\mathcal{I}) \)

This gives same guarantees!
Practical Takeaways

Measure calibration with either:

1. Kernel Calibration Error

2. Interval Calibration Error

or, if you must use binnedECE, add max-interval-width "regularizer"
In Practice: $kCE \approx \text{binnedECE}$