Robust Deep Learning: Where Are We?

Ju Sun (Computer Sci. & Eng.)

Oct 20th, 2023

MnRI Colloquium
Success of deep learning (DL) not news anymore

Commercial breakthroughs ...

image classification

self-driving vehicles credit: wired.com

smart-home devices credit: Amazon

Go game (2017)

healthcare credit: Google AI

robotics credit: Cornell U.
Robustness issues of DL not news anymore

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as ‘speed limit 45’.
Robustness issues across domains/tasks

Name entry Recognition

While exact results differ depending on language/datasets, our key findings from these experiments can be summarized as follows:

1. NER models for all three languages are sensitive to adversarial input.

2. Adversarial fine-tuning and re-training could improve the performance of NER models both on original and adversarial test sets, without requiring additional manual labeled data.
Robustness issues across models

Foundational Robustness of Foundation Models

Abstract

Foundation models adopting the methodology of deep learning with pre-training on large-scale unlabeled data and finetuning with task-specific supervision are becoming a mainstream technique in machine learning. Although foundation models hold many promises in learning general representations and few-shot/zero-shot generalization across domains and data modalities, at the same time they raise unprecedented challenges and considerable risks in robustness and privacy due to the use of the excessive volume of data and complex neural network architectures. This tutorial aims to deliver a Coursera-like online tutorial containing comprehensive lectures, a hands-on and interactive Jupyter/Colab live coding demo, and a panel discussion on different aspects of trustworthiness in foundation models. More information can be found at https://sites.google.com/view/neurips2022-frfm-tutorial

Two kinds of robustness

Adversarial examples

Natural corruptions

credit: openai.com

credit: ImageNet-C
Other dimensions in trustworthy AI

Trustworthiness: robustness, fairness, explainability, transparency
Outline

● **Evaluation of adversarial robustness**

● **Fundamental challenges in evaluating & achieving robustness**

● **Selective prediction**
  Margin As An Effective Confidence Score For Selective Classification Under Distribution Shifts (Forthcoming)

● **Closing**
Robustness evaluation (RE)

Maximize loss/error function

\[
\max_{x'} \ell(y, f_\theta(x'))
\]

subject to

\[
d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n
\]

Allowable perturbation
Valid image

Find robustness radius

\[
\min_{x'} d(x, x')
\]

subject to

\[
\max_{i \neq y} f_\theta^i(x') \geq f_\theta^y(x'), \quad x' \in [0, 1]^n
\]

On the decision boundary
Valid image

Report robust accuracy over an evaluation set
Constrained optimization problems

\[
\begin{align*}
\max_{x'} & \ell(y, f_\theta(x')) \\
\text{s.t.} & \quad d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n
\end{align*}
\]

Both objective and constraint functions are \textbf{nonconvex} in general, e.g., when containing DL models.

\[
\begin{align*}
\min_{x'} & \quad d(x, x') \\
\text{s.t.} & \quad \max_{i \neq y} f^i_\theta(x') \geq f^y_\theta(x'), \quad x' \in [0, 1]^n
\end{align*}
\]
Projected gradient descent (PGD) for RE

\[
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n \\
\min_{x \in Q} f(x) \\
x_{k+1} = P_Q\left(x_k - \alpha_k \nabla f(x_k)\right) \\
P_Q(x_0) = \arg\min_{x \in Q} \frac{1}{2} \|x - x_0\|^2_2
\]

Step size

Projection operator

Key hyperparameters:
(1) step size
(2) iteration number

**Algorithm 1 APGD**

1. **Input:** \( f, S, x^{(0)}, \eta, N_{\text{iter}}, W = \{w_0, \ldots, w_n\} \)
2. **Output:** \( x_{\text{max}}, f_{\text{max}} \)
3. \( x^{(1)} \leftarrow P_S\left(x^{(0)} + \eta \nabla f(x^{(0)})\right) \)
4. \( f_{\text{max}} \leftarrow \max\{f(x^{(0)}), f(x^{(1)})\} \)
5. \( x_{\text{max}} \leftarrow x^{(0)} \text{ if } f_{\text{max}} = f(x^{(0)}) \text{ else } x_{\text{max}} \leftarrow x^{(1)} \)
6. for \( k = 1 \) to \( N_{\text{iter}}-1 \) do
7. \( z^{(k+1)} \leftarrow P_S\left(z^{(k)} + \eta \nabla f(z^{(k)})\right) \)
8. \( x^{(k+1)} \leftarrow P_S\left(x^{(k)} + \alpha (z^{(k+1)} - x^{(k)}) + (1 - \alpha)(x^{(k)} - x^{(k-1)})\right) \)
9. if \( f(x^{(k+1)}) > f_{\text{max}} \) then
10. \( x_{\text{max}} \leftarrow x^{(k+1)} \text{ and } f_{\text{max}} \leftarrow f(x^{(k+1)}) \)
11. end if
12. if \( k \in W \) then
13. if Condition 1 or Condition 2 then
14. \( \eta \leftarrow \eta/2 \) and \( x^{(k+1)} \leftarrow x_{\text{max}} \)
15. end if
16. end if
17. end for

Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Croce, F., Hein, M., ICML 2020
Problem with projected gradient descent

Tricky to set: iteration number & step size

\[
\max_{x'} \ell(y, f_\theta(x'))
\]

s. t. \( d(x, x') \leq \epsilon, \ x' \in [0, 1]^n \)

Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Croce, F., Hein, M., ICML 2020

Penalty methods for complicated $d$

$$\max_{x'} \ell (y, f_{\theta}(x'))$$
\[\text{s.t. } d(x, x') \leq \epsilon, \quad x' \in [0, 1]^n\]

$$d(x, x') \doteq \|\phi(x) - \phi(x')\|_2$$

where $\phi(x) \doteq [\hat{g}_1(x), \ldots, \hat{g}_L(x)]$

**Projection onto the constraint is complicated**

**Penalty methods**

$$\max_{\bar{x}} \quad \mathcal{L}(f(\bar{x}), y) - \lambda \max \left(0, \|\phi(\bar{x}) - \phi(x)\|_2 - \epsilon\right)$$

Solve it for each fixed $\lambda$ and then increase $\lambda$

---

Problem with penalty methods

<table>
<thead>
<tr>
<th>Method</th>
<th>cross-entropy loss</th>
<th>margin loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Viol. (%) ↓</td>
<td>Att. Succ. (%) ↑</td>
</tr>
<tr>
<td>Fast-LPA</td>
<td>73.8</td>
<td>3.54</td>
</tr>
<tr>
<td>LPA</td>
<td>0.00</td>
<td>80.5</td>
</tr>
<tr>
<td>PPGD</td>
<td>5.44</td>
<td>25.5</td>
</tr>
<tr>
<td>PWCF (ours)</td>
<td>0.62</td>
<td>93.6</td>
</tr>
</tbody>
</table>

LPA, Fast-LPA: penalty methods  
PPGD: Projected gradient descent

Penalty methods tend to encounter **large constraint violation** (i.e., infeasible solution, known in optimization theory) or **suboptimal solution**

\[
\begin{align*}
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n \\
d(x, x') \triangleq \|\phi(x) - \phi(x')\|_2 \\
\text{where } \phi(x) \triangleq [\tilde{g}_1(x), \ldots, \tilde{g}_L(x)]
\end{align*}
\]

PWCF, an optimizer with a principled stopping criterion on **stationarity & feasibility**

Unreliable optimization = Unreliable RE
Issues and answers

**projected gradient descent**

\[
\min_{x \in Q} f(x)
\]

\[
x_{k+1} = P_Q \left( x_k - \alpha_k \nabla f(x_k) \right)
\]

Issue: no principled stopping criterion / step size rules

**penalty methods**

\[
\min_x f(x) \quad \text{s.t.} \quad g(x) \leq 0
\]

\[
\min_x f(x) + \lambda \max(0, g(x))
\]

Solved with increasing \( \lambda \) sequence

Issue: infeasible solution

- Feasible & stationary solution  
  Stationarity and feasibility check: KKT condition

- Reasonable speed  
  Line search & 2nd order methods

- A hidden problem: nonsmoothness
A principled solver for constrained, nonconvex, nonsmooth problems

Nonconvex, nonsmooth, constrained

\[
\min_{x \in \mathbb{R}^n} f(x), \quad \text{s.t. } c_i(x) \leq 0, \forall i \in \mathcal{I}; \quad c_i(x) = 0, \forall i \in \mathcal{E}.
\]

Penalty sequential quadratic programming (P-SQP)

\[
\min_{d \in \mathbb{R}^n, \ s \in \mathbb{R}^p} \mu(f(x_k) + \nabla f(x_k)^T d) + e^T s + \frac{1}{2} d^T H_k d
\]
\[
\text{s.t. } c(x_k) + \nabla c(x_k)^T d \leq s, \quad s \geq 0,
\]

Advantage: 2nd order method (BFGS) → high-precision solution

Principled line search, stationarity/feasibility check

Our PyGranso (and NCVX framework) https://ncvx.org/

First general-purpose solver for constrained DL problems

\[
\min f(x), \text{s.t. } c_i(x) \leq 0, \forall i \in \mathcal{I}; \ c_i(x) = 0, \forall i \in \mathcal{E}
\]
Strategies to speed up PyGranso for RE

\[
\begin{align*}
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n
\end{align*}
\]

\[
\begin{align*}
\min_{x'} d(x, x') \\
\text{s.t. } \max_{i \neq y} f^i_\theta(x') \geq f^y_\theta(x'), \quad x' \in [0, 1]^n
\end{align*}
\]

**Constraint folding: many constraints into few**

\[
\begin{align*}
h_j(x) = 0 & \iff |h_j(x)| \leq 0, \\
c_i(x) \leq 0 & \iff \max\{c_i(x), 0\} \leq 0, \\
F(|h_1(x)|, \ldots, |h_i(x)|, \max\{c_1(x), 0\}, \ldots, \max\{c_j(x), 0\}) & \leq 0,
\end{align*}
\]

**Two-stage optimization**

1. **Stage 1 (selecting the best initialization):** Optimize the problems by PWCF with $R$ different random initialization $x^{(r,0)}$ for $k$ iterations, where $r = 1, \ldots, R$, and collect the final first-stage solution $x^{(r,k)}$ for each run. Determine the best intermediate result $x^{(*,k)}$ following Algorithm 1.

2. **Stage 2 (optimization):** Warm start the optimization process with $x^{*,k}$ until the stopping criterion is met (i.e., reaching both the stationarity and feasibility tolerance, or reaching the MaxIter $K$).
First general-purpose, reliable solver for RE

\[
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0, 1]^n
\]

\[
\min_{x'} d(x, x') \\
\text{s.t. } \max_{i \neq y} f_\theta^i(x') \leq f_\theta^y(x'), \quad x' \in [0, 1]^n
\]

Reliability

- SOTA methods
  No stopping criterion (only use `maxit`); step size scheduler

- PWCF (ours)
  Principled line-search criterion and termination condition

Generality

- SOTA methods
  Can mostly only handle several lp metrics (l1,l2,linf)

- PWCF (ours)
  Any differentiable metrics and both min and max forms
  E.g., perceptual distance
  \[
  d(x, x') \doteq \|\phi(x) - \phi(x')\|_2 \\
  \text{where } \phi(x) \doteq [\tilde{g}_1(x), \ldots, \tilde{g}_L(x)]
  \]
A quick example

\[
\begin{align*}
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0,1]^n
\end{align*}
\]

\[
d(x, x') \equiv \|\phi(x) - \phi(x')\|_2
\]

where \( \phi(x) \equiv [\hat{g}_1(x), \ldots, \hat{g}_L(x)] \)

<table>
<thead>
<tr>
<th>Method</th>
<th>cross-entropy loss</th>
<th></th>
<th>margin loss</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Viol. (%) ↓</td>
<td>Att. Succ. (%) ↑</td>
<td></td>
<td>Viol. (%) ↓</td>
</tr>
<tr>
<td>Fast-LPA</td>
<td>73.8</td>
<td>3.54</td>
<td></td>
<td>41.6</td>
</tr>
<tr>
<td>LPA</td>
<td>0.00</td>
<td>80.5</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>PPGD</td>
<td>5.44</td>
<td>25.5</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>PWCF (ours)</td>
<td>0.62</td>
<td>93.6</td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>
PyGranso has enabled much more

\[
\min_{x \in \mathbb{R}^n} f(x), \quad \text{s.t.} \quad c_i(x) \leq 0, \forall i \in \mathcal{I}; \quad c_i(x) = 0, \forall i \in \mathcal{E}
\]

First general-purpose solver for constrained DL problems

**Topology optimization**

\[
\min_{\theta, u} \quad u^T K(G_\theta(\beta)) u \quad \text{s.t.} \quad K(G_\theta(\beta)) u = f \\
\sum_{i \in \Omega} [G_\theta(\beta)]_i = v_0, \quad G_\theta(\beta) \in \{0, 1\}^n
\]

**Imbalanced learning**

\[
\max_{\theta, t} \quad \sum_{i=1}^N \mathbb{1}\{y_i = +1\} \frac{1}{\sum_{i=1}^N f_\theta(x_i) > t}
\]

\[
\text{s.t.} \quad \sum_{i=1}^N \mathbb{1}\{y_i = +1\} \frac{1}{\sum_{i=1}^N f_\theta(x_i) > t} \geq \alpha
\]

**NCVX Package**

NCVX: A General-Purpose Optimization Solver for Constrained Machine and Deep Learning

Buyun Liang, Tim Mitchell, Ju Sun

**Background**

Machine/Deep learning (MDL) has emerged as a novel tool in material science and engineering (MSE).^1 MDL models in MSE can be broadly categorized as "property prediction models" (PPMs) or "interatomic potentials" (IPs). For the former, the goal is to learn the mapping between material representations and material properties (e.g., formation energy, band gap, etc.). These representations can be compositional,^2 requiring only the chemical formula (e.g., Al₂O₃), or structural,^3 requiring the formula and the 3D arrangement of ions on a periodic lattice (e.g., Al₂O₃ in the corundum structure with specified coordinates for Al and O). IPs make use of a structural representation, but instead of learning to predict a single property, these models learn to predict the energies, forces, and stresses of an arbitrary configuration of ions on a lattice. Using this learned interatomic model, one can perform a set of tasks and analyses that are usually...
Outline

● **Evaluation of adversarial robustness**

● **Fundamental challenges in evaluating & achieving robustness**

● **Selective prediction**
  Margin As An Effective Confidence Score For Selective Classification Under Distribution Shifts (Forthcoming)

● **Closing**
RE tractable even with PWCF?

\[
\max_{x'} \ell(y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon, \quad x' \in [0,1]^n
\]

- Assuming 0-1 loss
- Typical over-specification of means there are potentially infinitely many solutions, with different patterns
Is the intuition right?
Is the intuition right? Measured by **sparsity levels** of the perturbations found.
Implications - 1

\[
\max_{x'} \ell (y, f_\theta(x')) \\
\text{s.t. } d(x, x') \leq \varepsilon , \quad x' \in [0, 1]^n
\]

We need to **enumerate** all possible solutions if we want reliable RE using max-form

**Take-away:** Max-form RE is fundamentally intractable, unless a good \( \varepsilon \) is set—which is hard
Implications - II

Adversarial training

\[
\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \max_{x' \in \Delta(x)} \ell \left( y, f_{\theta}(x') \right)
\]

i.e., data augmentation with adversarial samples

We need to **enumerate** all possible patterns of adversarial samples if we want to achieve robustness, measured by the same \( d \).

**Take-away:** Adversarial training with the max-form augmentation won't achieve robustness.
Any hopes remaining?

\[
\begin{align*}
&\max_{x'} \ell(y, f_\theta(x')) \\
&\text{s.t. } d(x, x') \leq \epsilon, \quad x' \in [0, 1]^n
\end{align*}
\]

\begin{align*}
\text{VS} \\
&\min_{x'} d(x, x') \\
&\text{s.t. } \max_{i \neq y} f^i_\theta(x') \geq f^y_\theta(x'), \quad x' \in [0, 1]^n
\end{align*}

**Take-away:** the min–form (robustness radius) is more promising
Outline

● Evaluation of adversarial robustness

Optimization and Optimizers for Adversarial Robustness  https://arxiv.org/abs/2303.13401

● Fundamental challenges in evaluating & achieving robustness

Optimization and Optimizers for Adversarial Robustness  https://arxiv.org/abs/2303.13401

● Selective prediction

Margin As An Effective Confidence Score For Selective Classification Under Distribution Shifts (Forthcoming)

● Closing
We have a long way to go

TRUSTWORTHY AI RESEARCH THRUSTS
DARPA experts estimate that research in the following areas will be essential to creating trustworthy technology:

- Foundational theory, to understand the art of the possible, bound the limits of particular system instantiations, and inform guardrails for AI systems in challenging domains such as national security;
- AI engineering, to predictably build systems that work as intended in the real world and not just in the lab; and
- Human-AI teaming, to enable systems to serve as fluent, intuitive, trustworthy teammates to people with various backgrounds.


Safe Learning-Enabled Systems

PROGRAM SOLICITATION
NSF 23-562

National Science Foundation
Directorate for Computer and Information Science and Engineering
Division of Information and Intelligent Systems
Division of Computing and Communication Foundations
Division of Computer and Network Systems

Open Philanthropy Project LLC
Good Ventures Foundation

Full Proposal Deadline(s) (due by 5 p.m. submitter’s local time):

May 26, 2023
January 16, 2024

Imperfect AI models can still be deployed
A crucial component: allowing AI to restrain itself

\[
predictor \, f : \mathcal{X} \rightarrow \mathbb{R}^K \quad \text{selector} \, g : \mathcal{X} \rightarrow \{0, 1\}
\]

\[
(f, g)(\mathbf{x}) \triangleq \begin{cases} 
  f(\mathbf{x}) & \text{if } g(\mathbf{x}) = 1; \\
  \text{abstain} & \text{if } g(\mathbf{x}) = 0.
\end{cases}
\]

No prediction on uncertain samples and defer them to humans

\[
g_\gamma(\mathbf{x}) = 1[s(\mathbf{x}) > \gamma]
\]

Typically, selection by thresholding prediction confidence
Risk-coverage tradeoff

\[(f, g)(x) \triangleq \begin{cases} f(x) & \text{if } g(x) = 1; \\ \text{abstain} & \text{if } g(x) = 0. \end{cases} \]

\[g_\gamma(x) = 1[s(x) > \gamma]\]

(coverage) \( \phi_\gamma = \mathbb{E}_D[g_\gamma(x)] \),

(selection risk) \( R_\gamma = \mathbb{E}_D[\ell(f(x), y)g_\gamma(x)]/\phi_\gamma \).
Which confidence score?

\( z \in \mathbb{R}^K \) contains the raw logits (RLs)

\[
SR_{\text{max}} \triangleq \max_i \sigma(z^i),
\]

\[
SR_{\text{doctor}} \triangleq \frac{\|\sigma(z)\|_2^2 - 1}{\|\sigma(z)\|_2^2} = 1 - \frac{\|\sigma(z)\|_1}{\|\sigma(z)\|_2^2},
\]

\[
SR_{\text{ent}} \triangleq \sum_i \sigma(z^i) \log \sigma(z^i),
\]

\[\approx p(y = 1|\mathbf{x})\]
But are they good scores?

$z \in \mathbb{R}^K$ contains the raw logits (RLs)

Scale factor applied to RLs

Calibration: align the outputs with the true posterior probs
Our margin-based scores

Binary SVMs: \[ f(x) = w^T x + b \]

Geometric margin: \[ y(w^T x + b)/\|w\|_2 \]

Multiclass SVMs: \[ f(x) = W^T x + b \]

Geometric margin: \[ \frac{w_{y'}^T x + b_{y'}}{\|w_{y'}\|_2} - \max_{j \in \{1, \ldots, K\} \setminus y'} \frac{w_j^T x + b_j}{\|w_j\|_2} \]

Confidence margin: \[ (w_{y'}^T x + b_{y'}) - \max_{i \in \{1, \ldots, K\} \setminus y'} (w_i^T x + b_i) \]

These scores are not affected by the logit scaling

Signed dist to the separating hyperplane

Difference of dists between the two nearest hyperplanes
Our margin-based scores

Geometric margin:
\[
\frac{w_y^T x + b_y}{\|w_y\|_2} - \max_{j \in \{1,\ldots,K\} \setminus y'} \frac{w_j^T x + b_j}{\|w_j\|_2}
\]

Confidence margin:
\[
(w_y^T x + b_y) - \max_{i \in \{1,\ldots,K\} \setminus y'} (w_i^T x + b_i)
\]

Apply them to the RLs

Benefit: We don't need to worry about the scale of
Additional benefit: robustness
On real data

ImageNet vs ImageNet-C

(a) IN (Clean)  (b) Gaussian blur Lv.1  (c) Gaussian blur Lv.3  (d) Gaussian blur Lv.5

<table>
<thead>
<tr>
<th></th>
<th>IN (Clean)</th>
<th>Gaussian Blur</th>
<th>Brightness</th>
<th>Fog</th>
<th>Snow</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.1 0.5 1</td>
<td>0.1 0.5 1</td>
<td>0.1 0.5 1</td>
<td>0.1 0.5 1</td>
<td>0.1 0.5 1</td>
</tr>
<tr>
<td>$RL_{conf-M}$</td>
<td>0.16 0.53 2.39</td>
<td>0.37 1.31 6.05</td>
<td>0.21 0.72 3.35</td>
<td>0.14 0.79 4.21</td>
<td>0.17 0.95 4.80</td>
</tr>
<tr>
<td>$RL_{geo-M}$</td>
<td>0.27 0.59 2.43</td>
<td>0.57 1.36 6.04</td>
<td>0.33 0.79 3.39</td>
<td>0.25 0.86 4.22</td>
<td>0.34 1.02 4.81</td>
</tr>
<tr>
<td>$RL_{max}$</td>
<td>5.54 4.05 4.57</td>
<td>9.74 7.38 9.52</td>
<td>7.38 5.17 6.06</td>
<td>7.74 5.77 7.01</td>
<td>9.44 6.44 7.90</td>
</tr>
<tr>
<td>$SR_{max}$</td>
<td>3.19 2.40 3.38</td>
<td>5.02 4.02 7.39</td>
<td>4.07 2.90 4.53</td>
<td>3.92 3.07 5.37</td>
<td>5.35 3.67 6.13</td>
</tr>
<tr>
<td>$SR_{ent}$</td>
<td>4.28 3.13 4.04</td>
<td>6.80 5.63 8.71</td>
<td>5.51 4.01 5.48</td>
<td>5.56 4.37 6.42</td>
<td>7.29 5.07 7.27</td>
</tr>
<tr>
<td>$SR_{doctor}$</td>
<td>3.21 2.38 3.40</td>
<td>5.05 4.05 7.47</td>
<td>4.10 2.93 4.58</td>
<td>3.95 3.10 5.42</td>
<td>5.39 3.71 6.20</td>
</tr>
</tbody>
</table>
Outline

- **Evaluation of adversarial robustness**

- **Fundamental challenges in evaluating & achieving robustness**

- **Selective prediction**
  Margin As An Effective Confidence Score For Selective Classification Under Distribution Shifts (Forthcoming)

- **Closing**
Closing

- A long way to go for DL robustness

- Selective prediction crucial for deploying imperfect AI