Robust Deep Learning: Where Are We?

Ju Sun (Computer Sci. & Eng.)

Oct 20th, 2023

MnRI Colloquium

University of Minnesota
Driven to Discover℠

Success of deep learning (DL) not news anymore

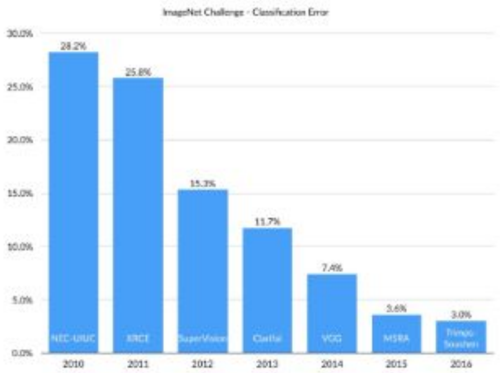


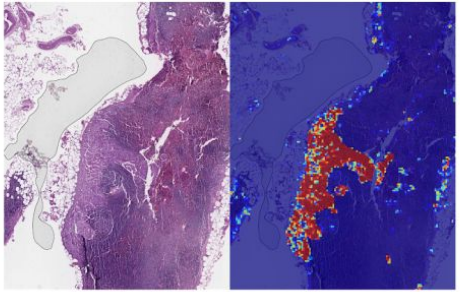
image classification

Go game (2017)

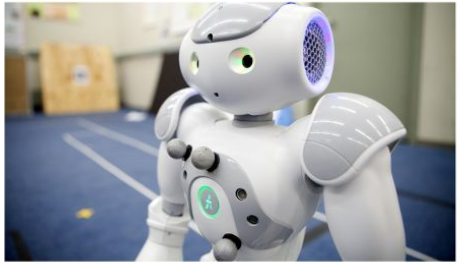
Commercial breakthroughs ...

self-driving vehicles credit: wired.com

smart-home devices credit: Amazon

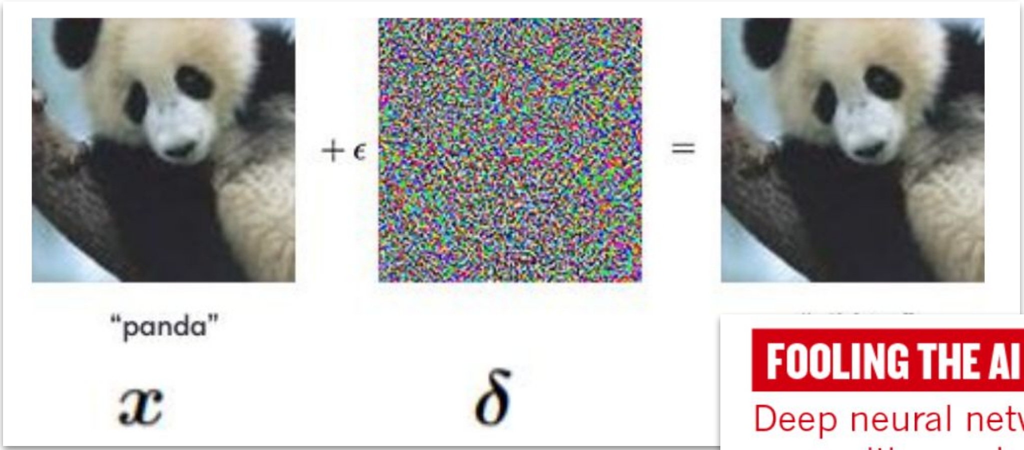


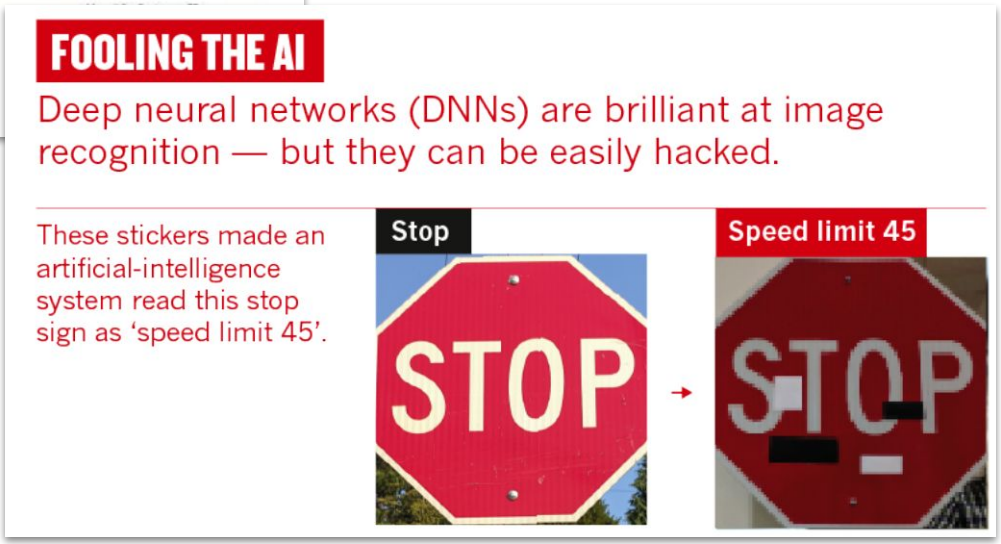
healthcare credit: Google AI



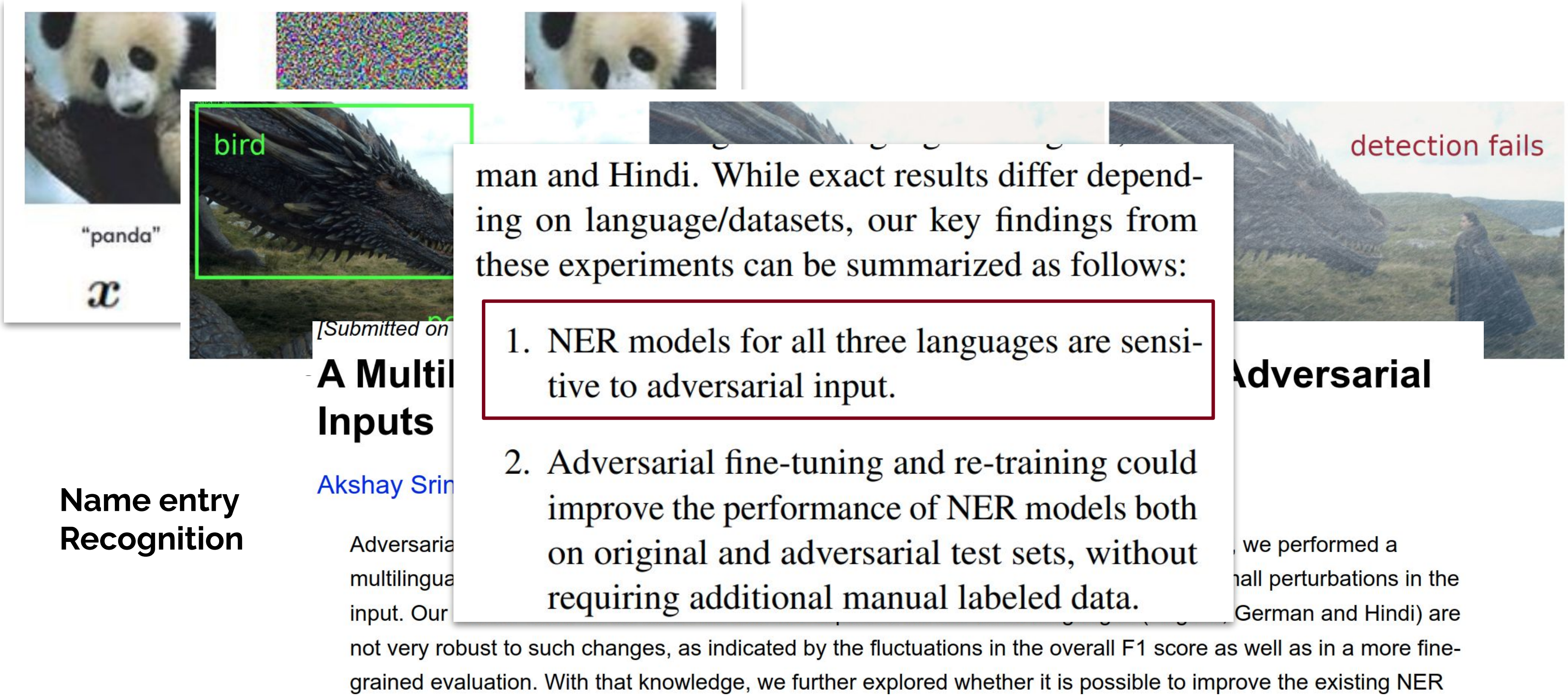
robotics credit: Cornell U.

Robustness issues of DL not news anymore





Robustness issues across domains/tasks



Name entry Recognition

Robustness issues across models

Tutorial

Foundational Robustness of Foundation Models

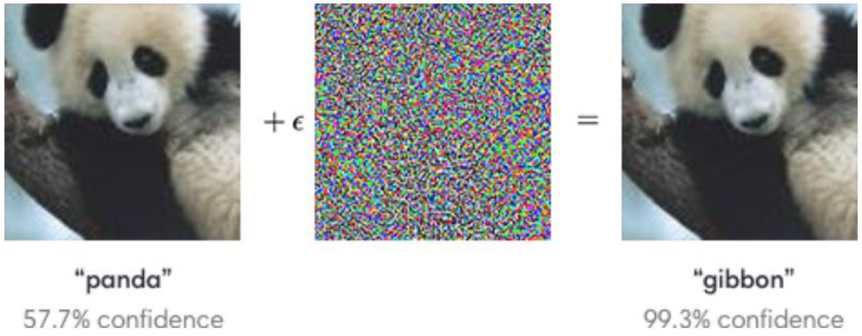
NeurIPS 2022

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-turotial

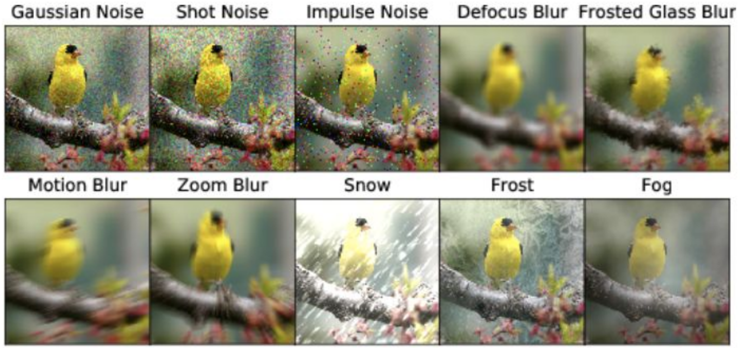
https://research.ibm.com/publications/foundational-robustness-of-foundation-models

Two kinds of robustness



credit: openai.com

Adversarial examples

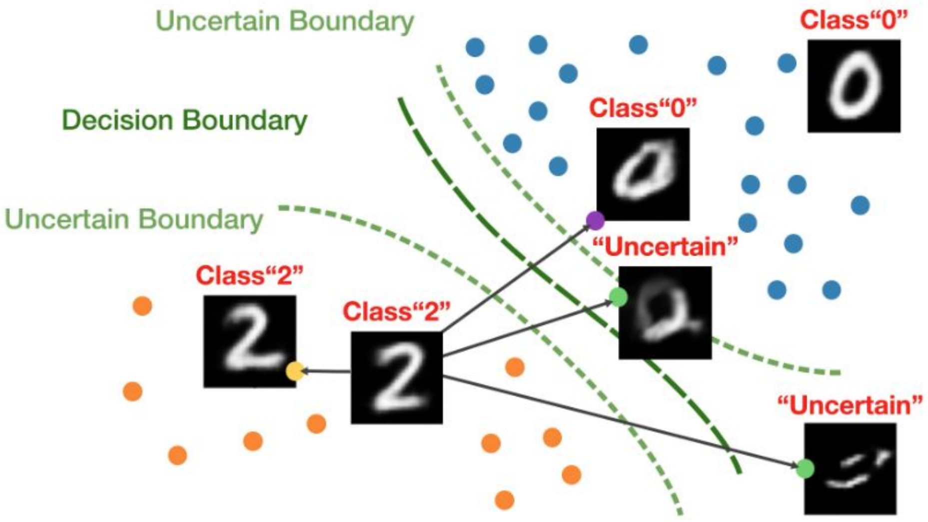


credit: ImageNet-C

Natural corruptions

Other dimensions in trustworthy AI

Trustworthiness: robustness, fairness, explainability, transparency



Boldness

Outline

- **Evaluation of adversarial robustness**

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

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

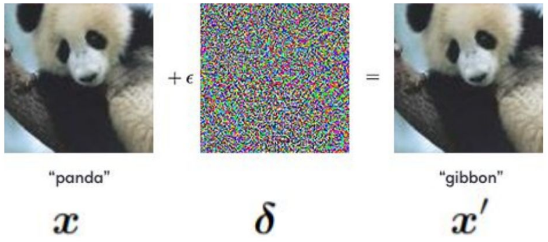
  Optimization and Optimizers for Adversarial Robustness https://arxiv.org/abs/2303.13401

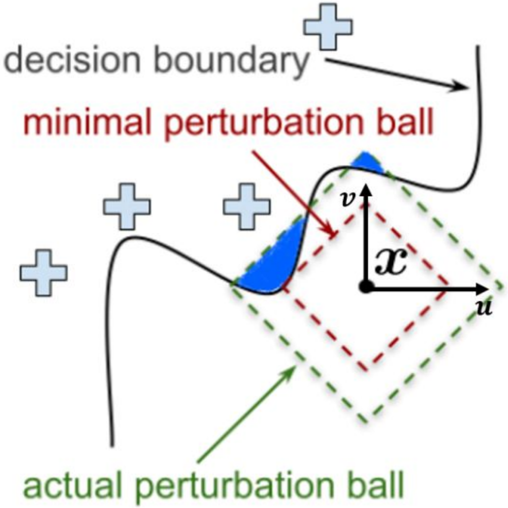
- **Selective prediction**

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

- **Closing**

Robustness evaluation (RE)





Maximize loss/error function

$$\boxed{\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)}$$

$$\text{s.t.} \ \boxed{d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon}, \quad \boxed{\boldsymbol{x}' \in [0,1]^n}$$

Allowable perturbation   Valid image

Find robustness radius

$$\boxed{\min_{\boldsymbol{x}'} \ d\left(\boldsymbol{x}, \boldsymbol{x}'\right)}$$

$$\text{s.t.} \ \boxed{\max_{i \neq y} f^i_{\boldsymbol{\theta}}(\boldsymbol{x}') \geq f^y_{\boldsymbol{\theta}}(\boldsymbol{x}')}, \ \boxed{\boldsymbol{x}' \in [0,1]^n}$$

On the decision boundary   Valid image

Report <u>robust accuracy</u> over an evaluation set

Constrained optimization problems

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

$$\boxed{\min_{\boldsymbol{x}'} \ d\left(\boldsymbol{x}, \boldsymbol{x}'\right)}$$
$$\text{s.t.} \ \boxed{\max_{i \neq y} f_{\boldsymbol{\theta}}^i(\boldsymbol{x}') \geq f_{\boldsymbol{\theta}}^y(\boldsymbol{x}')}, \ \boldsymbol{x}' \in [0,1]^n$$

Both objective and constraint functions are **nonconvex** in general, e.g., when containing DL models

Projected gradient descent (PGD) for RE

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

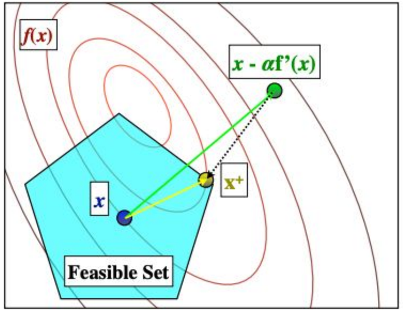
$$\min_{\mathbf{x} \in \mathcal{Q}} f(\mathbf{x})$$

Step size

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)\Big)$$

$$P_{\mathcal{Q}}(\mathbf{x}_0) = \arg\min_{\mathbf{x} \in \mathcal{Q}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2$$

Projection operator



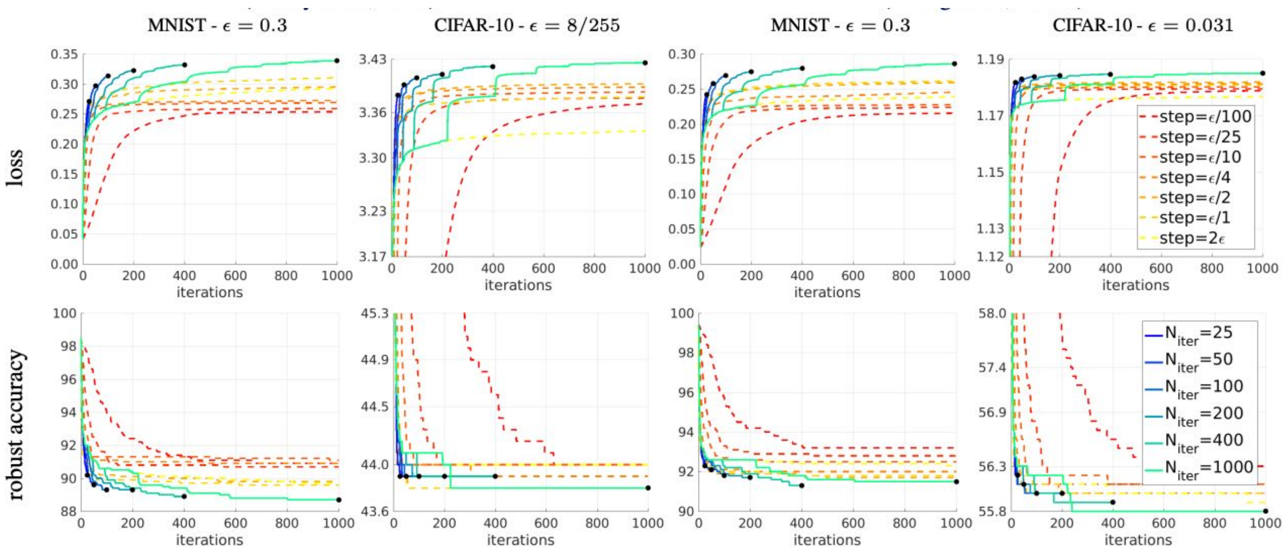
Key hyperparameters:
(1) step size
(2) iteration number

Algorithm 1 APGD

Input: f, S, $x^{(0)}$, η, N_{iter}, $W = \{w_0, \ldots, w_n\}$
Output: $x_{\max}$, $f_{\max}$
$x^{(1)} \leftarrow P_S\left(x^{(0)} + \eta \nabla f(x^{(0)})\right)$
$f_{\max} \leftarrow \max\{f(x^{(0)}), f(x^{(1)})\}$
$x_{\max} \leftarrow x^{(0)}$ **if** $f_{\max} \equiv f(x^{(0)})$ **else** $x_{\max} \leftarrow x^{(1)}$
for $k = 1$ **to** $N_{\text{iter}} - 1$ **do**
  $z^{(k+1)} \leftarrow P_S\left(x^{(k)} + \eta \nabla f(x^{(k)})\right)$
  $x^{(k+1)} \leftarrow P_S\Big(x^{(k)} + \alpha(z^{(k+1)} - x^{(k)}) + (1-\alpha)(x^{(k)} - x^{(k-1)})\Big)$
  if $f(x^{(k+1)}) > f_{\max}$ **then**
    $x_{\max} \leftarrow x^{(k+1)}$ and $f_{\max} \leftarrow f(x^{(k+1)})$
  end if
  if $k \in W$ **then**
    if Condition 1 **or** Condition 2 **then**
      $\eta \leftarrow \eta/2$ and $x^{(k+1)} \leftarrow x_{\max}$
    end if
  end if
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



$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

Tricky to set:
iteration number & step size
i.e., tricky to decide where to stop

Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Croce, F., Hein, M., ICML 2020
https://arxiv.org/pdf/2003.01690.pdf

Penalty methods for complicated d

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

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

$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x}) \,]$$

perceptual distance

Projection onto the constraint is complicated

Penalty methods

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

Solve it for each fixed λ and then increase λ

Algorithm 2 Lagrangian Perceptual Attack (LPA)

procedure LPA(classifier network $f(\cdot)$, LPIPS distance $d(\cdot,\cdot)$, input $\mathbf{x}$, label y, bound ϵ)
  $\lambda \leftarrow 0.01$
  $\widetilde{\mathbf{x}} \leftarrow \mathbf{x} + 0.01 * \mathcal{N}(0,1)$ ▷ initialize perturbations with random Gaussian noise
  for i in $1,\ldots,S$ **do** ▷ we use $S = 5$ iterations to search for the best value of λ
    for t in $1,\ldots,T$ **do** ▷ T is the number of steps
      $\Delta \leftarrow \nabla_{\widetilde{\mathbf{x}}}\left[\mathcal{L}(f(\widetilde{\mathbf{x}}), y) - \lambda \max\left(0, d(\widetilde{\mathbf{x}}, \mathbf{x}) - \epsilon\right)\right]$ ▷ take the gradient of (5)
      $\hat{\Delta} = \Delta / \|\Delta\|_2$ ▷ normalize the gradient
      $\eta = \epsilon * (0.1)^{t/T}$ ▷ the step size η decays exponentially
      $m \leftarrow d(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}} + h\hat{\Delta})/h$ ▷ $m \approx$ derivative of $d(\widetilde{\mathbf{x}}, \cdot)$ in the direction of $\hat{\Delta}$; $h = 0.1$
      $\widetilde{\mathbf{x}} \leftarrow \widetilde{\mathbf{x}} + (\eta/m)\hat{\Delta}$ ▷ take a step of size η in LPIPS distance
    end for
    if $d(\widetilde{\mathbf{x}}, \mathbf{x}) > \epsilon$ **then**
      $\lambda \leftarrow 10\lambda$ ▷ increase λ if the attack goes outside the bound
    end if
  end for
  $\widetilde{\mathbf{x}} \leftarrow \textsc{Project}(d, \widetilde{\mathbf{x}}, \mathbf{x}, \epsilon)$
  return $\widetilde{\mathbf{x}}$
end procedure

Ref Perceptual adversarial robustness: Defense against unseen threat models. Laidlaw, C., Singla, S., & Feizi, S. https://arxiv.org/abs/2006.12655

Problem with penalty methods

	cross-entropy loss		margin loss	
Method	**Viol. (%) ↓**	**Att. Succ. (%) ↑**	**Viol. (%) ↓**	**Att. Succ. (%) ↑**
Fast-LPA	73.8	3.54	41.6	56.8
LPA	**0.00**	80.5	**0.00**	97.0
PPGD	5.44	25.5	**0.00**	38.5
PWCF (ours)	0.62	93.6	**0.00**	100

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

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$
$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \dots, \widehat{g}_L(\boldsymbol{x}) \,]$$

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**

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

Ref Optimization and Optimizers for Adversarial Robustness. Liang, H., Liang, B., Peng, L., Cui, Y., Mitchell, T., & Sun, J. arXiv preprint arXiv:2303.13401.

Unreliable optimization = Unreliable RE

Issues and answers

projected gradient descent

$$\min_{\mathbf{x}\in\mathcal{Q}} f(\mathbf{x})$$

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)\Big)$$

Issue: no principled stopping criterion /step size rules

penalty methods

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad \mathrm{s.t.} \;\; g(\boldsymbol{x}) \leq \mathbf{0}$$

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) + \lambda \max(0, g(\boldsymbol{x}))$$

Solved with increasing λ 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

GRANSO

Nonconvex, nonsmooth, constrained

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

Penalty sequential quadratic programming (P-SQP)

$$\boxed{\begin{aligned} \min_{d\in\mathbb{R}^n,\ s\in\mathbb{R}^p} \quad & \mu\left(f(x_k) + \nabla f(x_k)^\mathsf{T} d\right) + e^\mathsf{T} s + \frac{1}{2} d^\mathsf{T} H_k d \\ \text{s.t.} \quad & c(x_k) + \nabla c(x_k)^\mathsf{T} d \le s, \quad s \ge 0, \end{aligned}}$$

Advantage: 2nd order method (BFGS) $\rightarrow$ high-precision solution

Principled line search, stationarity/feasibility check

Ref Curtis, Frank E., Tim Mitchell, and Michael L. Overton. "A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles." Optimization Methods and Software 32.1 (2017): 148-181.

Our PyGranso (and NCVX framework)

https://ncvx.org/

GRANSO + PyTorch

PyGRANSO

NCVX PyGRANSO Documentation

Search the docs ...

Introduction

Installation

Settings

Examples

Home

NCVX

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \ \text{s.t. } c_i(\mathbf{x}) \le 0, \forall i \in \mathcal{I};\ c_i(\mathbf{x}) = 0, \forall i \in \mathcal{E}$$

NCVX Package

First general-purpose solver for constrained DL problems

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

Buyun Liang, Tim Mitchell, Ju Sun

Strategies to speed up PyGranso for RE

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

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

Constraint folding: many constraints into few

$$h_j(\boldsymbol{x}) = 0 \iff |h_j(\boldsymbol{x})| \leq 0,$$
$$c_i(\boldsymbol{x}) \leq 0 \iff \max\{c_i(\boldsymbol{x}), 0\} \leq 0,$$

$$\mathcal{F}(|h_1(\boldsymbol{x})|, \cdots, |h_i(\boldsymbol{x})|, \max\{c_1(\boldsymbol{x}), 0\}, \cdots, \max\{c_j(\boldsymbol{x}), 0\}) \leq 0,$$

Two-stage optimization

1. **Stage 1 (selecting the best initialization):** Optimize the problems by PWCF with R different random initialization $\boldsymbol{x}^{(r,0)}$ for k iterations, where $r = 1, \ldots, R$, and collect the final first-stage solution $\boldsymbol{x}^{(r,k)}$ for each run. Determine the best intermediate result $\boldsymbol{x}^{(*,k)}$ following Algorithm 1.
2. **Stage 2 (optimization):** Warm start the optimization process with $\boldsymbol{x}^{*,k}$ until the stopping criterion is met[7] (i.e., reaching both the stationarity and feasibility tolerance, or reaching the MaxIter K).

First general-purpose, reliable solver for RE

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

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

Reliability

- SOTA methods (RobustBench: A standardized benchmark for adversarial robustness)
  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

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$
$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\ \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x})\ ]$$

A quick example

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

$$d(\boldsymbol{x}, \boldsymbol{x}') \doteq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')\|_2$$
$$\text{where} \quad \phi(\boldsymbol{x}) \doteq [\, \widehat{g}_1(\boldsymbol{x}), \ldots, \widehat{g}_L(\boldsymbol{x}) \,]$$

	cross-entropy loss		margin loss	
Method	**Viol. (%) ↓**	**Att. Succ. (%) ↑**	**Viol. (%) ↓**	**Att. Succ. (%) ↑**
Fast-LPA	73.8	3.54	41.6	56.8
LPA	**0.00**	80.5	**0.00**	97.0
PPGD	5.44	25.5	**0.00**	38.5
PWCF (ours)	0.62	93.6	**0.00**	100

PyGranso has enabled much more

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \text{ s.t. } c_i(\mathbf{x}) \le 0, \forall i \in \mathcal{I};\ c_i(\mathbf{x}) = 0, \forall i \in \mathcal{E}$$

First general-purpose solver for constrained DL problems

PyGRANSO

NCVX PyGRANSO Documentation

Search the docs ...

Introduction

Installation

Settings

Examples

Home

NCVX

NCVX Package

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

Buyun Liang, Tim Mitchell, Ju Sun

Topology optimization

$$\min_{\boldsymbol{\theta},\boldsymbol{u}} \boldsymbol{u}^\intercal \boldsymbol{K}(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}))\boldsymbol{u} \quad \text{s.t. } \boldsymbol{K}(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}))\boldsymbol{u} = \boldsymbol{f},$$

$$\sum_{i\in\Omega}[\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta})]_i = v_0,\ \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\beta}) \in \{0,1\}^n$$

Imbalanced learning

$$\max_{\boldsymbol{\theta},t} \frac{\sum_{i=1}^N \mathbb{1}\{y_i = +1\}\,\mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^N \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}$$

$$\text{s.t. } \frac{\sum_{i=1}^N \mathbb{1}\{y_i = +1\}\,\mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^N \mathbb{1}\{y_i = +1\}} \ge \alpha$$

Constrained deep learning for the efficient discovery of stable solid-state materials

PIs: Chris Bartel (CEMS), Ju Sun (CS&E)

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., Al_2O_3), or structural,[3] requiring the formula and the 3D arrangement of ions on a periodic lattice (e.g., Al_2O_3 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.[4] Using this learned interatomic model, one can perform a set of tasks and analyses that are usually

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**

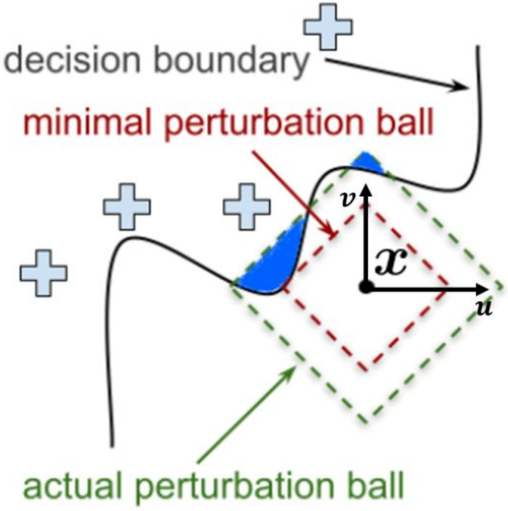
  Margin As An Effective Confidence Score For Selective Classification Under Distribution Shifts (Forthcoming)

- **Closing**

RE tractable even with PWCF?

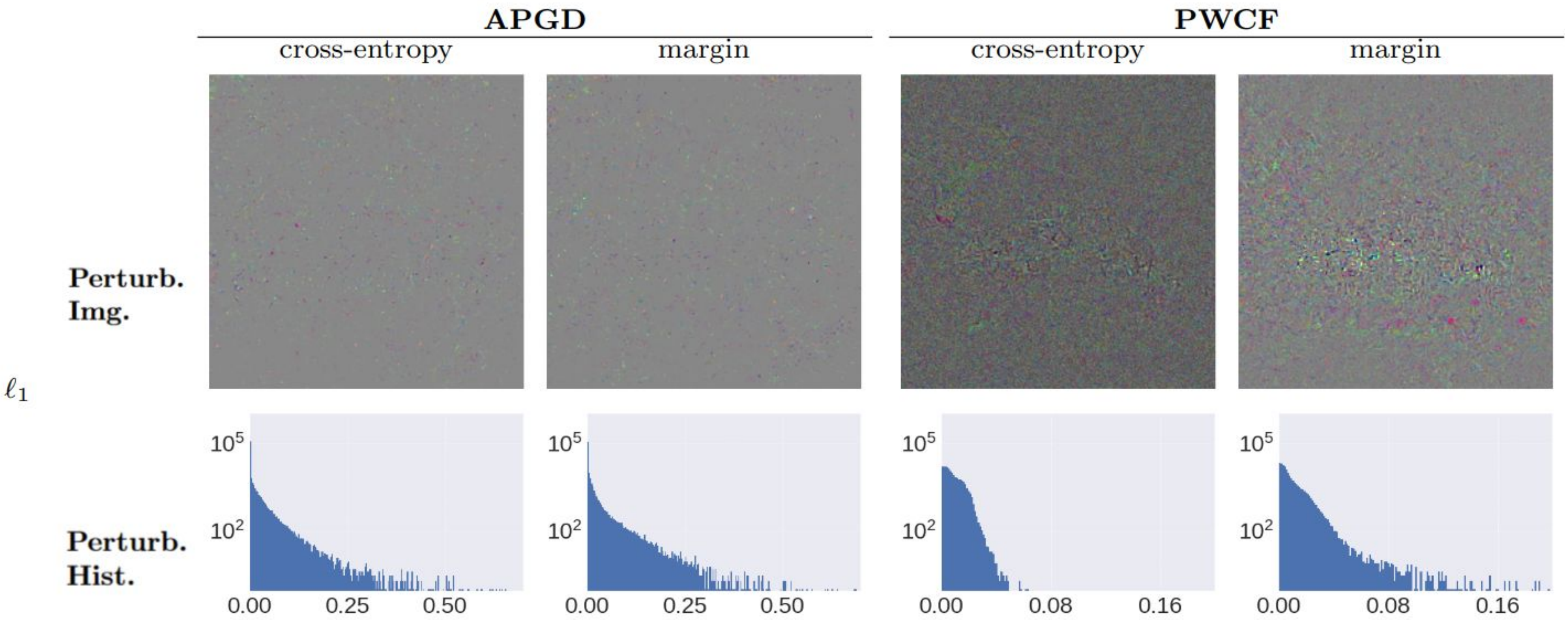
$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$



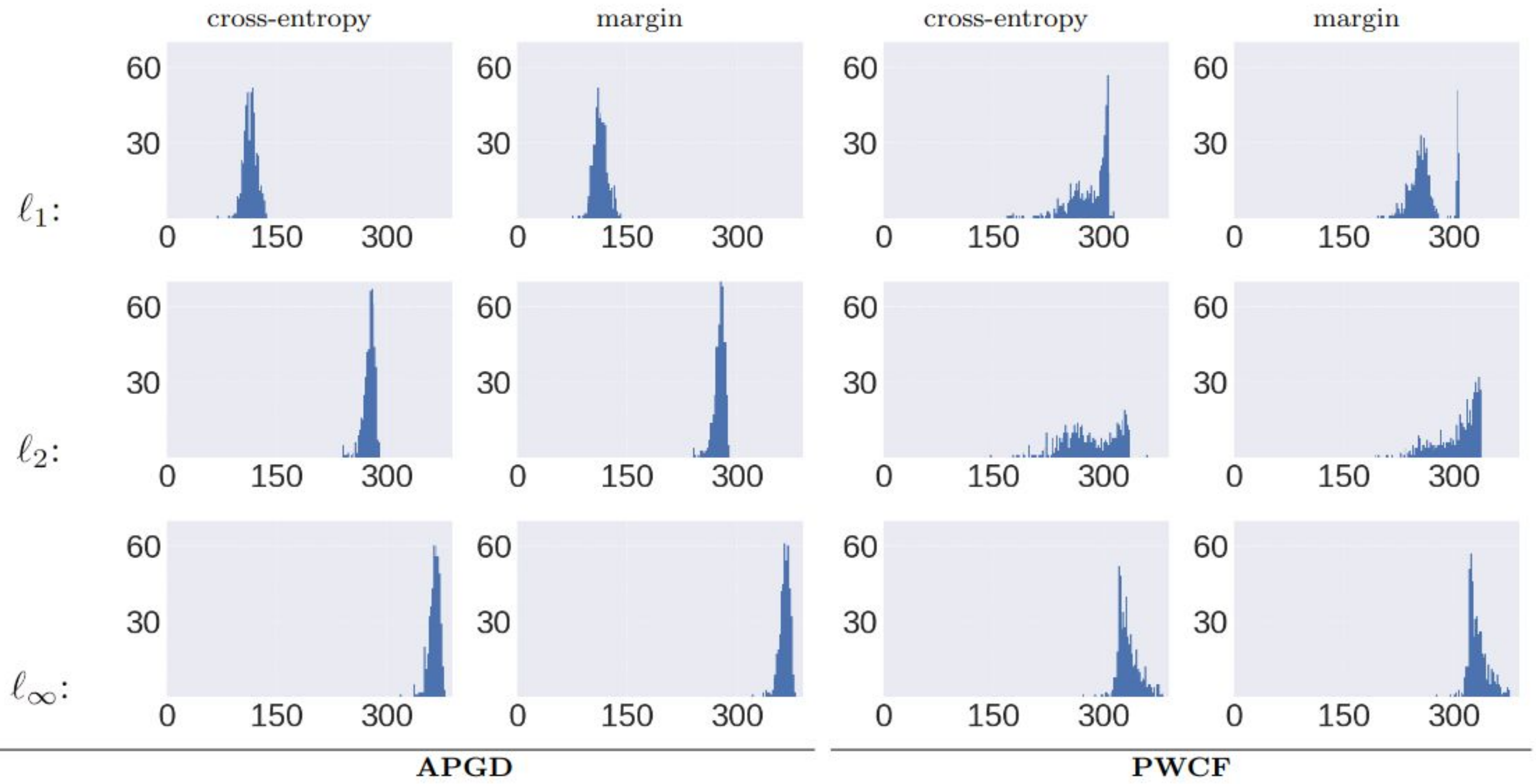
- Assuming 0-1 loss
- Typical **over-specification of** ε means there are potentially infinitely many solutions, with **different patterns**

Is the intuition right?



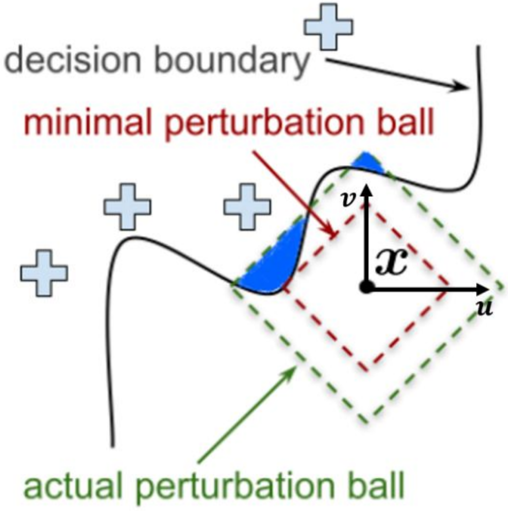
Is the intuition right?

Measured by **sparsity levels** of the perturbations found



Implications - I

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



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 ε is set—which is hard

Implications - II

$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t.}\ \boxed{d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \leq \varepsilon\ , \quad \boldsymbol{x}' \in [0,1]^n} \qquad \boldsymbol{x}' \in \Delta(\boldsymbol{x})$$

Adversarial training

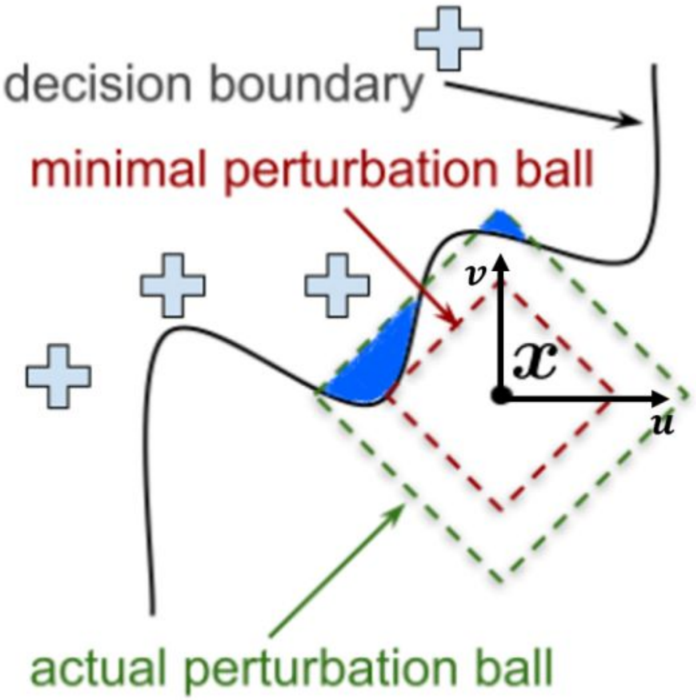
$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \max_{\boldsymbol{x}'\in\Delta(\boldsymbol{x})} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

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?



$$\max_{\boldsymbol{x}'} \ell\left(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}')\right)$$

$$\text{s.t. } d\left(\boldsymbol{x}, \boldsymbol{x}'\right) \le \varepsilon\,, \quad \boldsymbol{x}' \in [0,1]^n$$

VS

$$\min_{\boldsymbol{x}'} \; d\left(\boldsymbol{x}, \boldsymbol{x}'\right)$$

$$\text{s.t. } \max_{i \neq y} f_{\boldsymbol{\theta}}^{i}(\boldsymbol{x}') \ge f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}')\,, \; \boldsymbol{x}' \in [0,1]^n$$

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.

https://www.darpa.mil/work-with-us/ai-forward

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

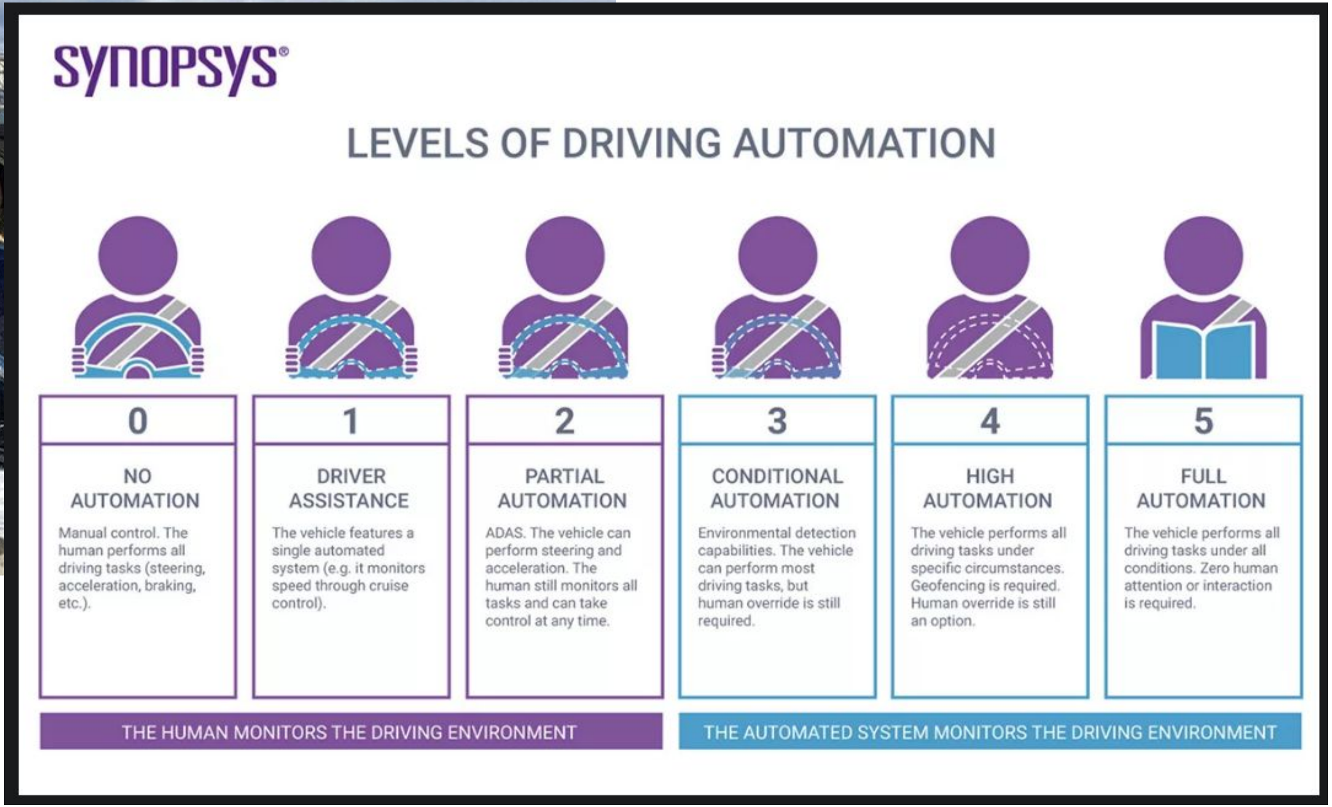
Full Proposal Deadline(s) (due by 5 p.m. submitter's local time):

May 26, 2023

January 16, 2024

https://www.nsf.gov/pubs/2023/nsf23562/nsf23562.htm

Imperfect AI models can still be deployed



A crucial component: allowing AI to restrain itself

$$\text{predictor } f : \mathcal{X} \to \mathbb{R}^K \qquad \text{selector } g : \mathcal{X} \to \{0,1\}$$

$$(f,g)(\boldsymbol{x}) \triangleq \begin{cases} f(\boldsymbol{x}) & \text{if } g(\boldsymbol{x}) = 1; \\ \boxed{\text{abstain} \quad \text{if } g(\boldsymbol{x}) = 0} . \end{cases}$$

No prediction on uncertain samples and defer them to humans

$$g_\gamma(\boldsymbol{x}) = \mathbb{1}[s(\boldsymbol{x}) > \gamma]$$

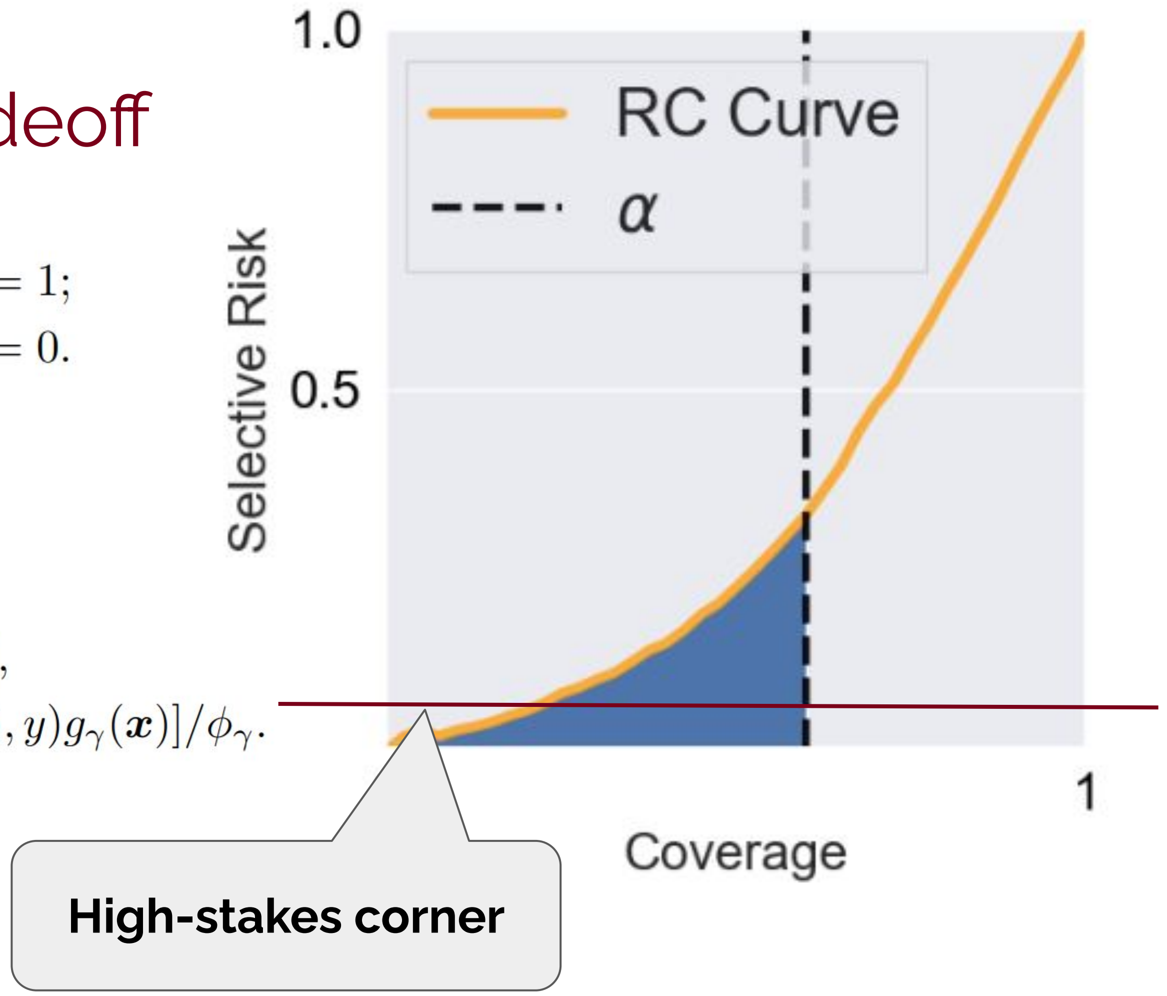
Typically, selection by thresholding prediction confidence

Risk-coverage tradeoff

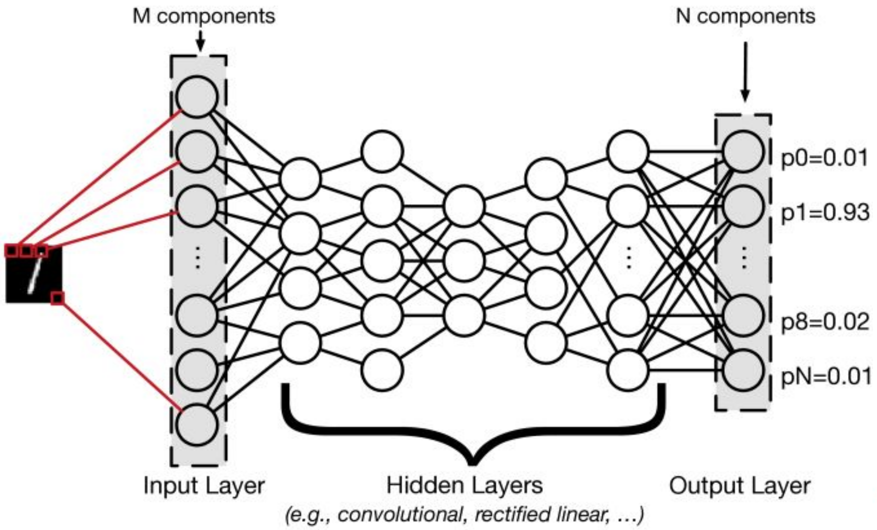
$$(f,g)(\boldsymbol{x}) \triangleq \begin{cases} f(\boldsymbol{x}) & \text{if } g(\boldsymbol{x}) = 1; \\ \text{abstain} & \text{if } g(\boldsymbol{x}) = 0. \end{cases}$$

$$g_\gamma(\boldsymbol{x}) = \mathbb{1}[s(\boldsymbol{x}) > \gamma]$$

$$\textbf{(coverage)}\ \phi_\gamma = \mathbb{E}_\mathcal{D}[g_\gamma(\boldsymbol{x})],$$
$$\textbf{(selection risk)}\ R_\gamma = \mathbb{E}_\mathcal{D}[\ell(f(\boldsymbol{x}), y) g_\gamma(\boldsymbol{x})]/\phi_\gamma.$$



Which confidence score?



$$\approx p(y = 1|\boldsymbol{x})$$

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

$$SR_{\max} \triangleq \max_i \sigma(z^i),$$

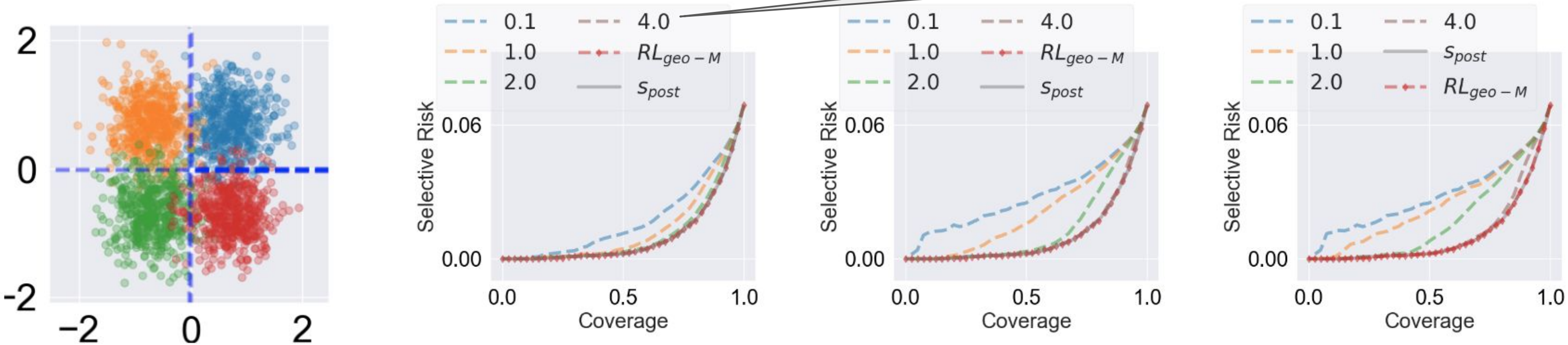
$$SR_{\text{doctor}} \triangleq (\|\sigma(\boldsymbol{z})\|_2^2 - 1)/\|\sigma(\boldsymbol{z})\|_2^2 = 1 - \|\sigma(\boldsymbol{z})\|_1/\|\sigma(\boldsymbol{z})\|_2^2,$$

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

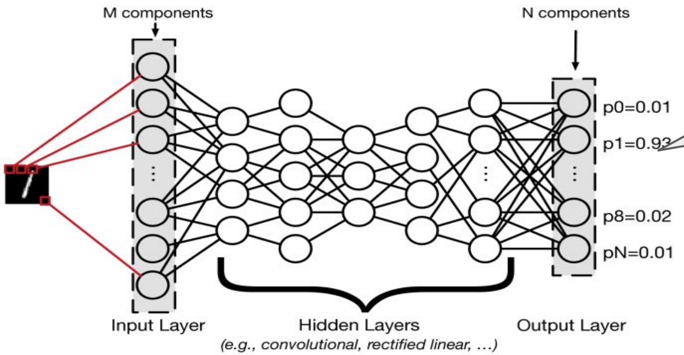
$\boldsymbol{z} \in \mathbb{R}^K$ contains the raw logits (RLs)

But are they good scores?

Scale factor applied to RLs



(a) Sample visualization (b) $SR_{\max}$ (c) SR_{doctor} (d) SR_{ent}



$\cancel{\approx p(\quad 1|\boldsymbol{x})}$

Calibration: align the outputs with the true posterior probs

Our margin-based scores

Signed dist to the separating hyperplane

Binary SVMs: $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ **Geometric margin:** $y(\boldsymbol{w}^\top \boldsymbol{x} + b)/\|\boldsymbol{w}\|_2$

Multiclass SVMs: $f(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$

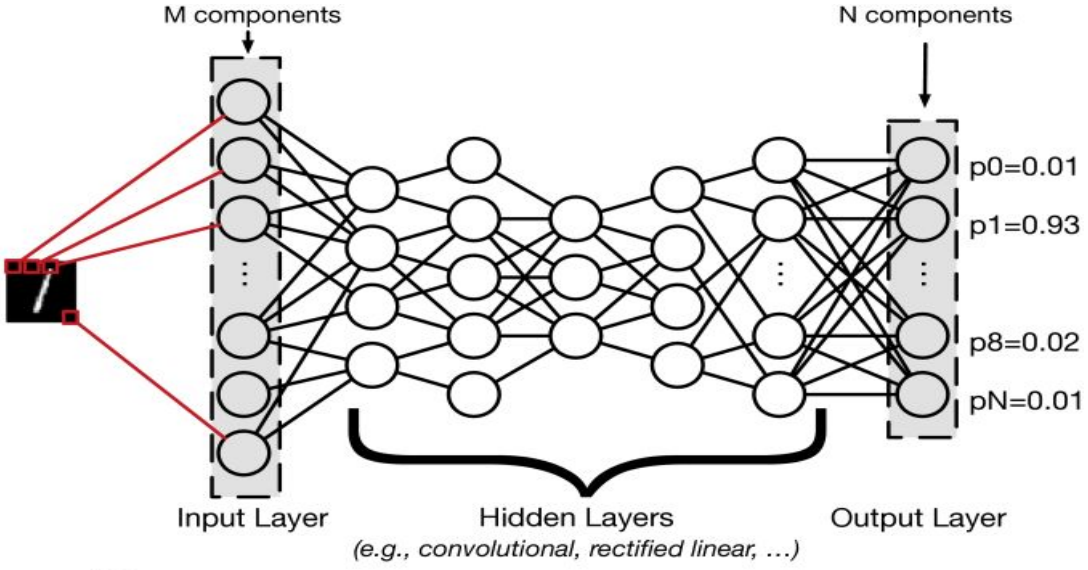
Geometric margin: $$\frac{\boldsymbol{w}_{y'}^\top \boldsymbol{x} + b_{y'}}{\|\boldsymbol{w}_{y'}\|_2} - \max_{j \in \{1,\ldots,K\} \setminus y'} \frac{\boldsymbol{w}_j^\top \boldsymbol{x} + b_j}{\|\boldsymbol{w}_j\|_2}$$

Confidence margin: $$(\boldsymbol{w}_{y'}^\top \boldsymbol{x} + b_{y'}) - \max_{i \in \{1,\ldots,K\} \setminus y'} (\boldsymbol{w}_i^\top \boldsymbol{x} + b_i)$$

Difference of dists between the two nearest hyperplanes

These scores are not affected by the logit scaling

Our margin-based scores



Geometric margin:

$$\frac{\boldsymbol{w}_{y'}^{\mathsf{T}}\boldsymbol{x} + b_{y'}}{\|\boldsymbol{w}_{y'}\|_2} - \max_{j\in\{1,\ldots,K\}\setminus y'} \frac{\boldsymbol{w}_j^{\mathsf{T}}\boldsymbol{x} + b_j}{\|\boldsymbol{w}_j\|_2}$$

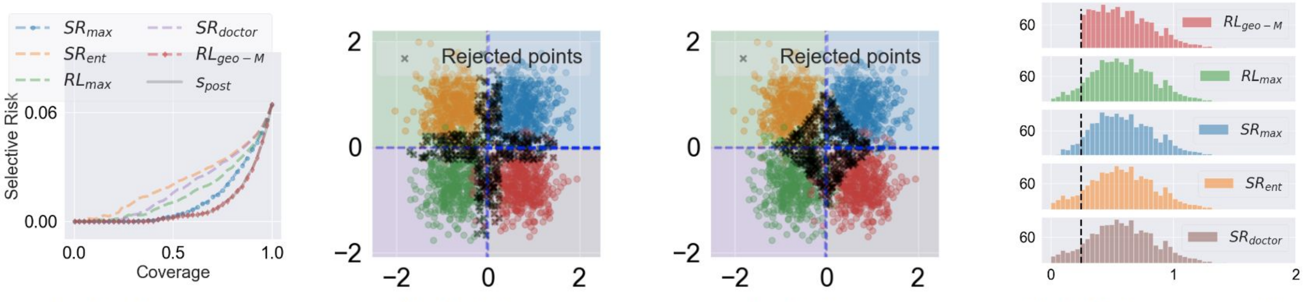
Confidence margin:

$$(\boldsymbol{w}_{y'}^{\mathsf{T}}\boldsymbol{x} + b_{y'}) - \max_{i\in\{1,\ldots,K\}\setminus y'} (\boldsymbol{w}_i^{\mathsf{T}}\boldsymbol{x} + b_i)$$

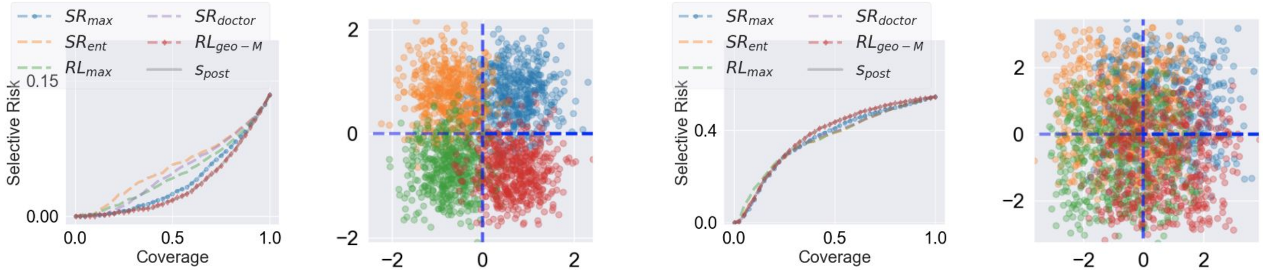
Apply them to the RLs z

Benefit: We don't need to worry about the scale of z

Additional benefit: robustness



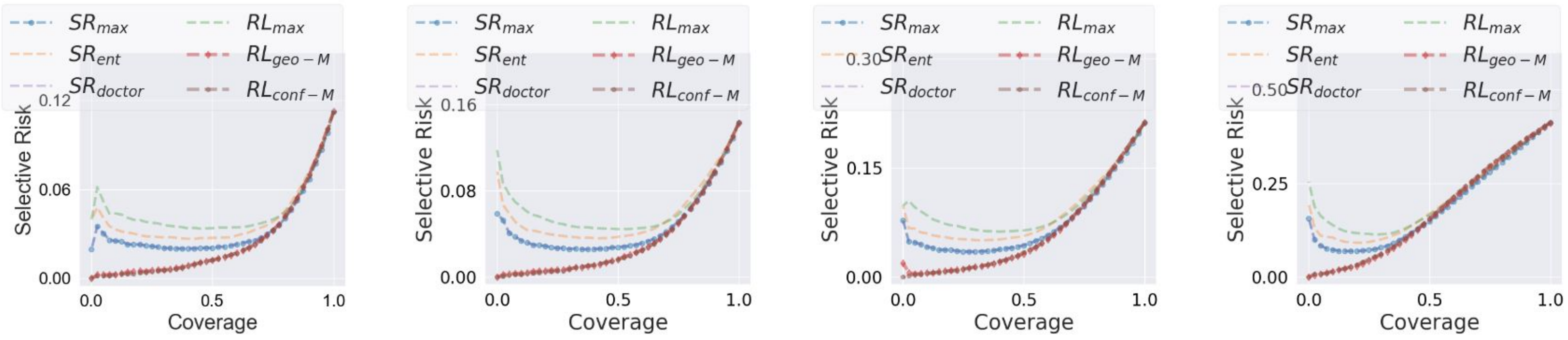
(a-1) RC curves **(b-1)** $RL_{\text{geo-M}}$ **(c-1)** $SR_{\max}$ **(d-1)** Robustness radius



(a-2) RC curves **(b-2)** Sample visualization **(a-3)** RC curves **(b-3)** Sample visualization

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

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

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**

Closing

- A long way to go for DL robustness
- Selective prediction crucial for deploying imperfect AI