Towards neural networks robust to distributions shifts

Praneeth Netrapalli
Google Research India

Work done in collaboration with

Anshul Nasery, Sravanti Addepalli, Harshay Shah, R. Venkatesh Babu and Prateek Jain

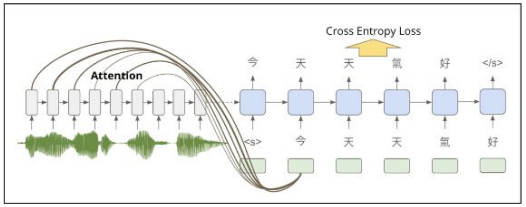
Outline

- The Robustness Problem
- Simplicity Bias
- Two key observations
  - Feature Replication Hypothesis
  - Non-robust features
- Algorithmic ideas
  - Feature Reconstruction Regularizer
  - Adversarial Fine-tuning
- Evaluation
- Conclusion

Motivation

Benchmark datasets and task performance

Image classification, ImageNet dataset

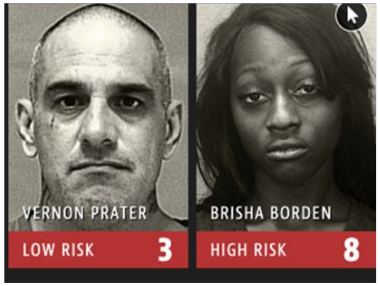


Speech Recognition, Switchboard-1 data

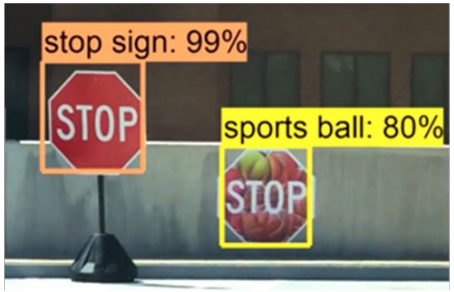
Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克强总理将启动中加总理年度对话机制，我加拿大总理特鲁多举行两国总理年度对话。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

Machine translation, WMT dataset

Distribution shifts in medical imaging



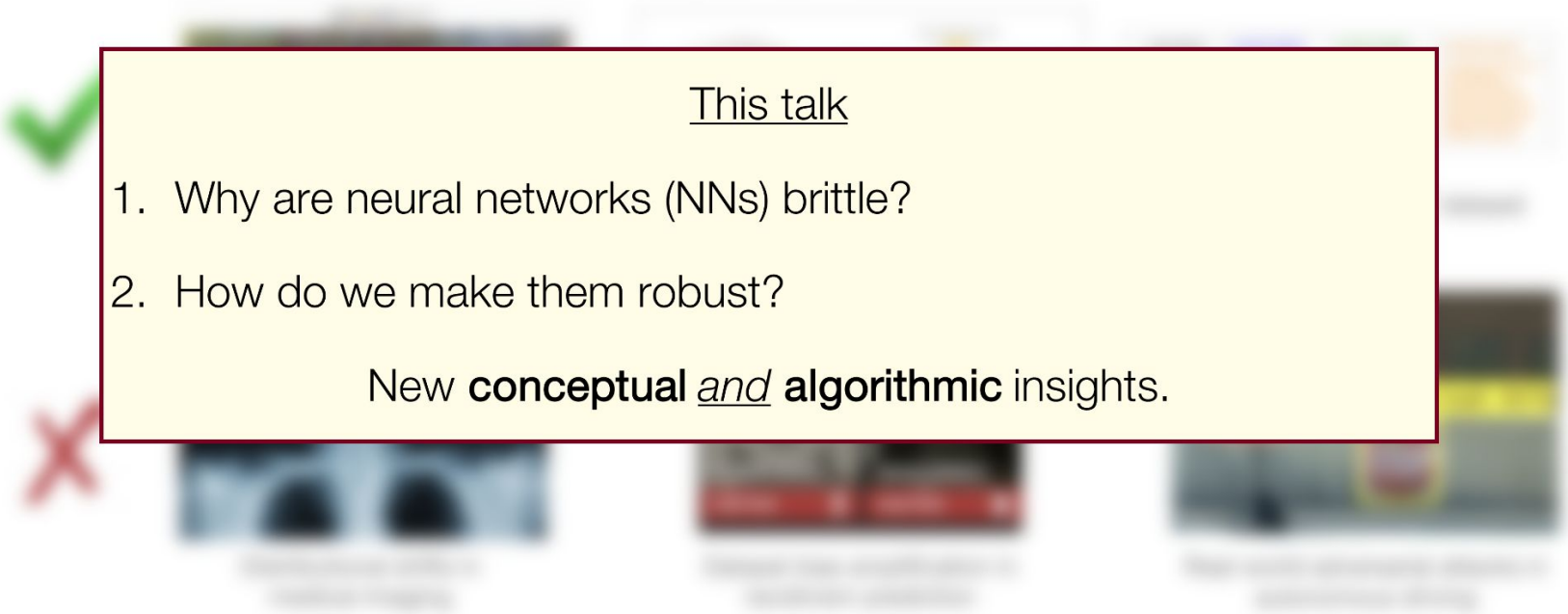
Dataset bias amplification in recidivism prediction



Real-world adversarial attacks in autonomous driving

Motivation

Benchmark datasets and task performance



This talk

1. Why are neural networks (NNs) brittle?
2. How do we make them robust?

New **conceptual** *and* **algorithmic** insights.

Thought Experiment

How do we distinguish swans and bears?

- Several features available: color, background, shape, organs etc.
- Humans look at these holistically. What does an NN learn?

Image Source: Wikipedia

Neural Networks Learn Only Some Features

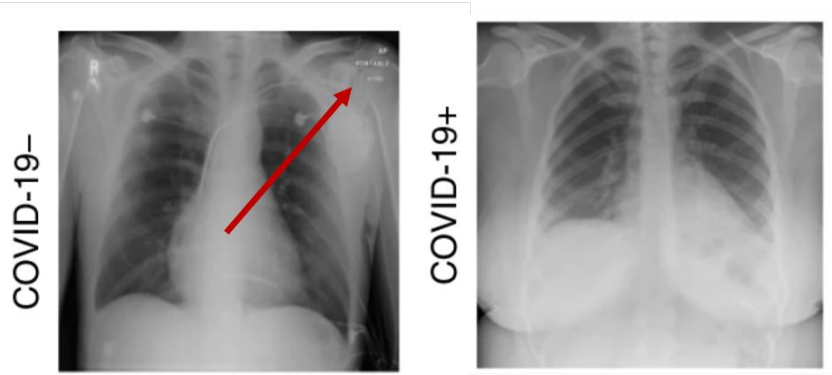
Texture bias Geirhos et al. (2018)

(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Shortcut learning DeGrave et al. (2021)



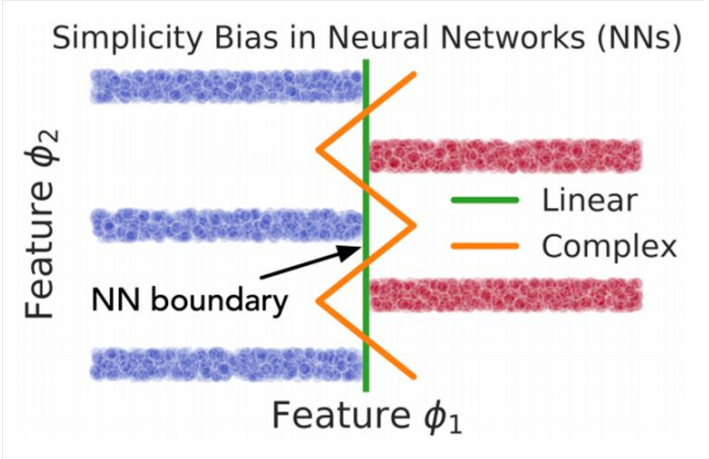
Why do NNs learn only some features?

Which features do NNs learn?

Simplicity Bias (SB) [STRJN, NeurIPS 2020]

NNs learn *simplest* features useful for classification

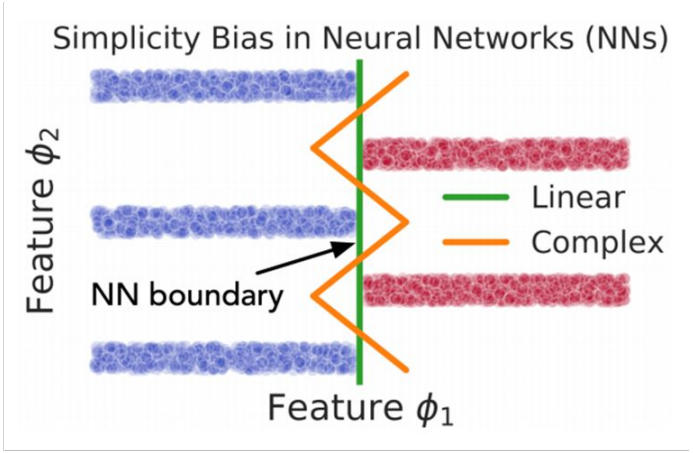
- Orange classifier has larger margin compared to green classifier.
- NNs have the capacity to learn Orange classifier.
- In practice however, NNs learn the green classifier.
- Rigorous proof for 1-hidden layer NN.



NNs Provably Exhibit Simplicity Bias

$$f(x) = \sum_{j=1}^{k} ReLU(\langle w_j, x \rangle), \qquad x \in R^d$$

- Initialization: $w_j \sim N(0, \frac{1}{dk} I)$
- Number of samples: $\Omega(d^2)$
- Number of nodes: $\tilde{O}(d^2)$
  - Covers overparameterised setting

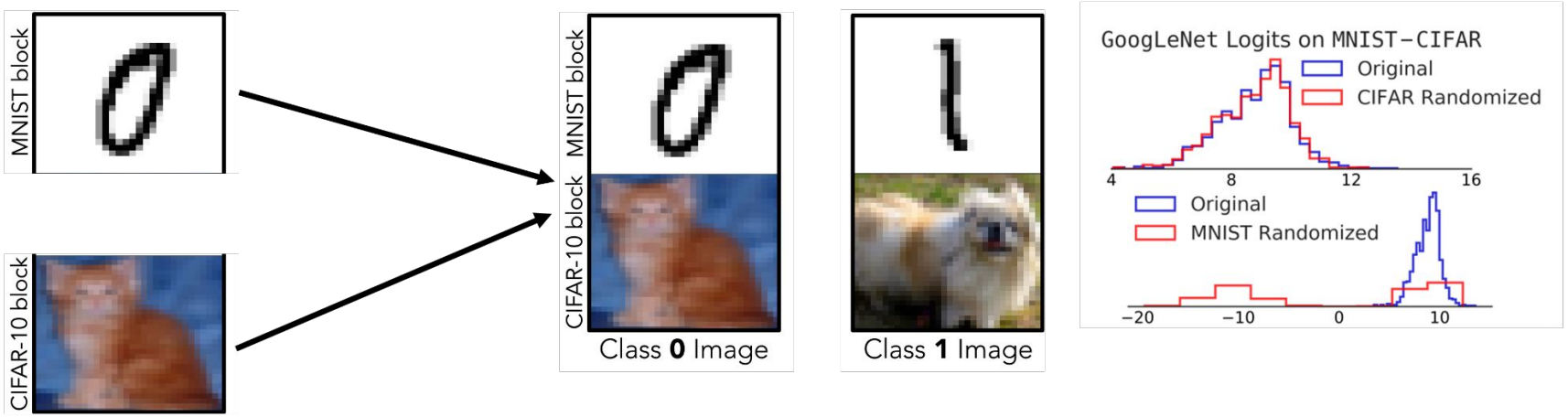


Weight of "linear feature": $O(\frac{1}{\sqrt{k}})$

Weight of "non-linear feature": $O(\frac{1}{\sqrt{dk}})$

Towards Real Datasets

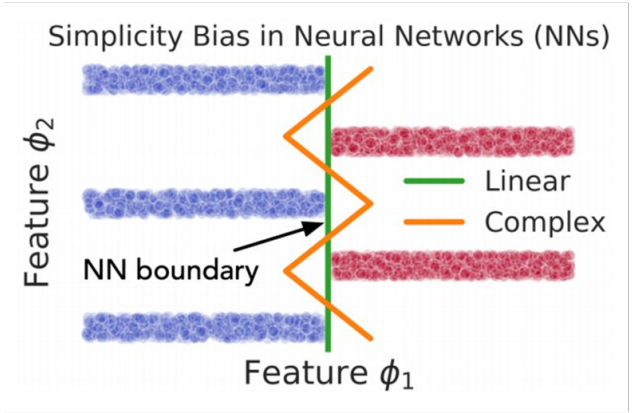
MNIST-CIFAR dataset and randomization tests



- **CIFAR randomized**: Randomize the CIFAR part of the image
- Logits do not change ⇒ Prediction **does not depend at all** on CIFAR part

Consequences of SB

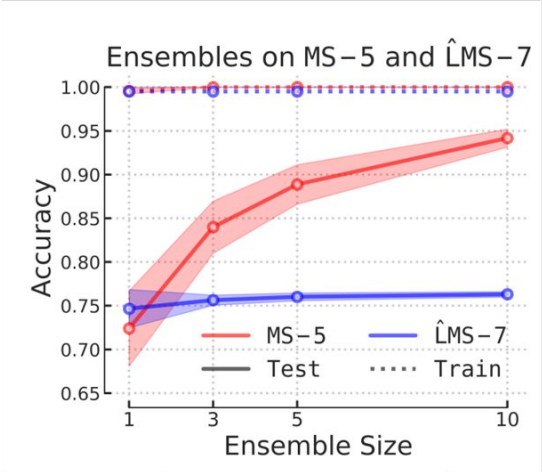
SB leads to brittleness to distribution shifts and adversarial examples

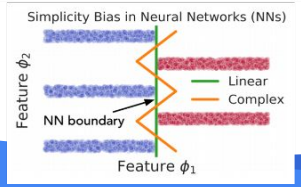


- Model learns **only** the simplest features ⇒ poor adversarial robustness
- (Simple features ≠ true features) ⇒ Poor out of distribution performance
- Can sometimes lead to poor **in-domain** performance

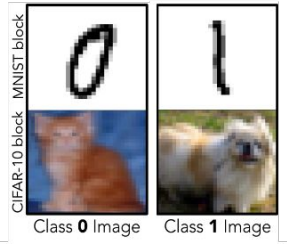
Fix SB using standard approaches?

- Ensemble?
  - Intuition: They can capture multiple features





- Adversarial Training?



Model	ℓ_∞ budget ε	Test Accuracy		ϵ-Robust Accuracy	
		Standard SGD	ℓ_∞ Adv. Training	Standard SGD	ℓ_∞ Adv. Training
`MobileNetV2`	0.30	0.999 ± 0.001	0.999 ± 0.000	0.000 ± 0.000	0.991 ± 0.000
`DenseNet121`	0.30	1.000 ± 0.000	0.999 ± 0.000	0.000 ± 0.000	0.981 ± 0.003
`ResNet50`	0.30	1.000 ± 0.000	0.999 ± 0.001	0.001 ± 0.000	0.982 ± 0.002

`CIFAR10-Randomized` Accuracy	
Standard SGD	ℓ_∞ Adv. Training
0.493 ± 0.005	0.493 ± 0.001
0.494 ± 0.005	0.501 ± 0.003
0.501 ± 0.001	0.499 ± 0.002

To fix SB,

need to understand its precise manifestation …

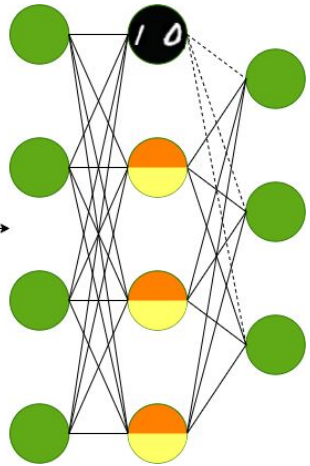
Test-bed dataset: **ColoredMNIST**

vs

Task: Classify red 0 vs yellow 1

High correlation between color and digit (label).

Key insight I: Feature replication (ICLR 2023)

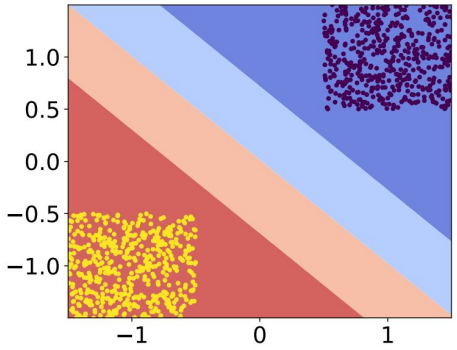


- Some features (e.g., color) are replicated multiple times in the feature space compared to other features (e.g., digit shape).
- Final linear classifier relies more on such replicated features.
- 3 layer CNN with 32 penultimate features has more color features than shape features.
- Output is more dependent on color features than shape features.

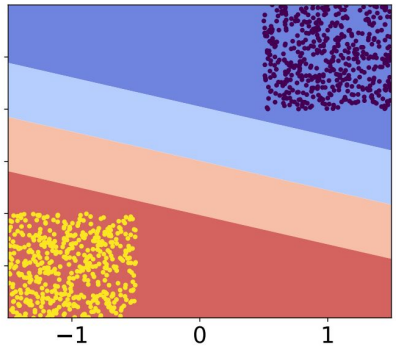
Type of feature	Number	Output correlation
Color	26	0.81
Shape	4	0.61

Max-Margin Classifier under Feature Replication

SVM, 0-Rep



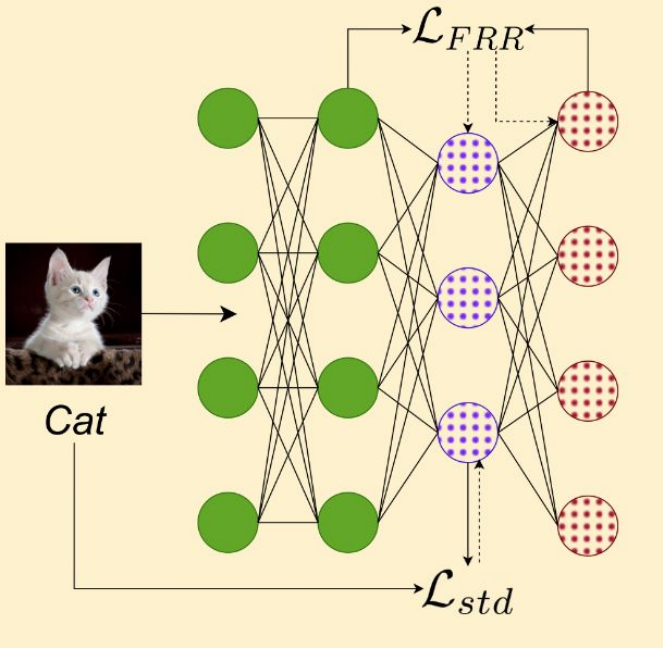
SVM, 5-Rep



Max margin classifier in replicated feature space - $\boldsymbol{w} = \left[\frac{2}{d}, \cdots, \frac{2}{d}\right]$

- SGD trained networks converge to the max-margin solution.
- When features are replicated, max-margin classifier gives more weight to the replicated feature.
- Becomes worse with increasing dimensions!

Feature Reconstruction Regularizer (FRR)

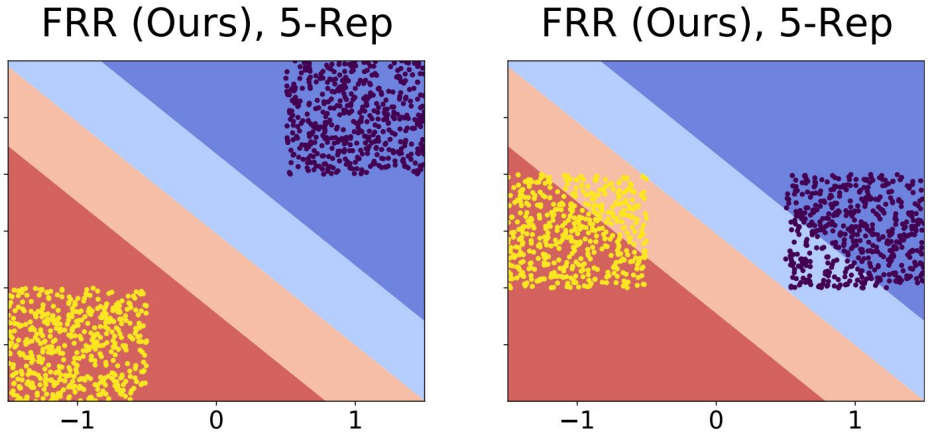


- Reconstruct features from logits
- Minimize the reconstruction loss
- Mathematical formulation -

$$\mathcal{L}_{\text{FRR}}(x, \theta, W, \phi) = ||f_\theta(x) - \mathcal{T}_\phi(W^T f_\theta(x))||_p$$

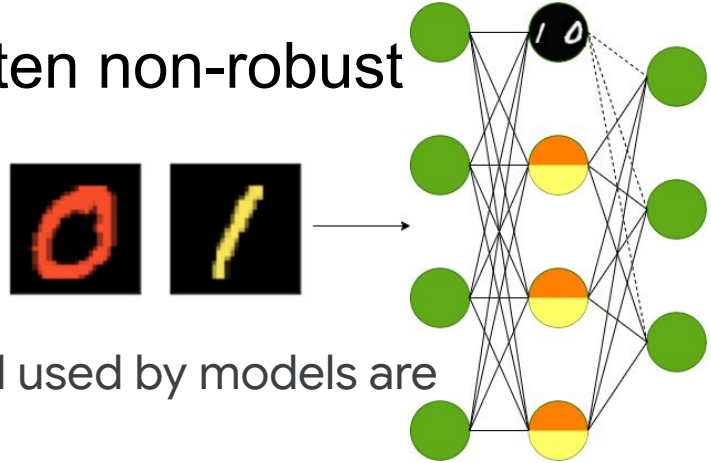
- Ensures that logits contain information about *all* features.

FRR under Feature Replication



- FRR gives equal weightage to replicated and unreplicated features
- Some caveats-
  - Needs relatively diverse representations
  - Needs some conditional variance between core and spurious features.

Key insight II: Replicated features are often non-robust



- The replicated features (e.g. color) learned and used by models are often brittle to small perturbations.
- We train two models on data with perfect shape and color correlation respectively, and compute their accuracy on adversarially perturbed images.
- The performance of a model dependent on color features sees a huge drop.

Feature used by model	Test Accuracy	Accuracy under perturbation =0.1
Color	99%	53%
Shape	99%	85%

Adversarial Examples

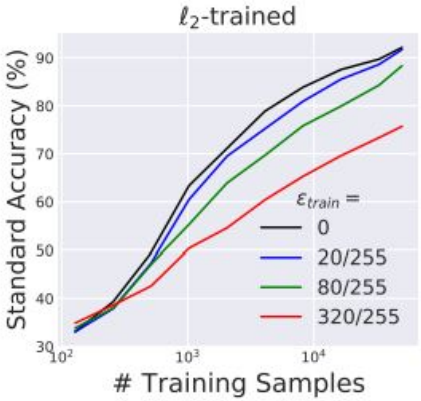
Clean

Adversarial

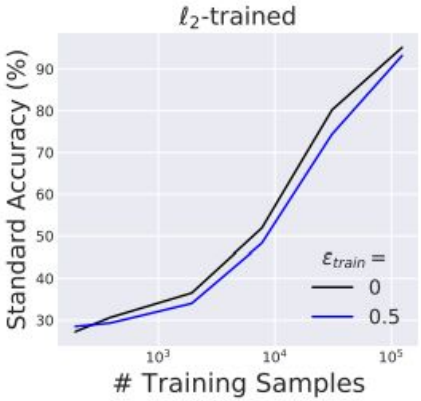
These are examples generated by solving $\max_{\hat{x} \in B_\epsilon(x)} \log \frac{\sum_{\hat{y}} \exp\left(w_{\hat{y}}^\top f_\theta(\hat{x})\right)}{\exp\left(w_y^\top f_\theta(\hat{x})\right)}$

i.e. the image in the ball of an input image for which the cross-entropy loss is maximized

Can we use adversarial training?



(b) CIFAR-10

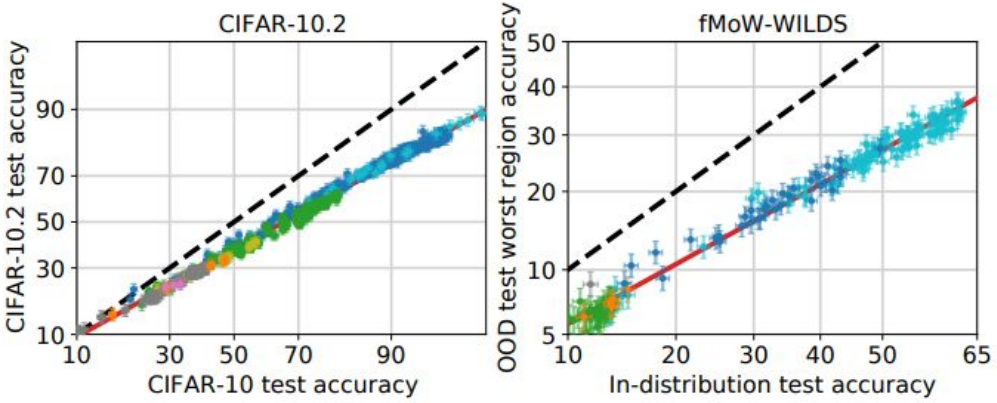


(c) Restricted ImageNet

Prior work [1] has shown that adversarial robustness is negatively correlated with clean accuracy.

[1] Towards Deep Learning Models Resistant to Adversarial Attacks by Madry et al

Can we use adversarial training?

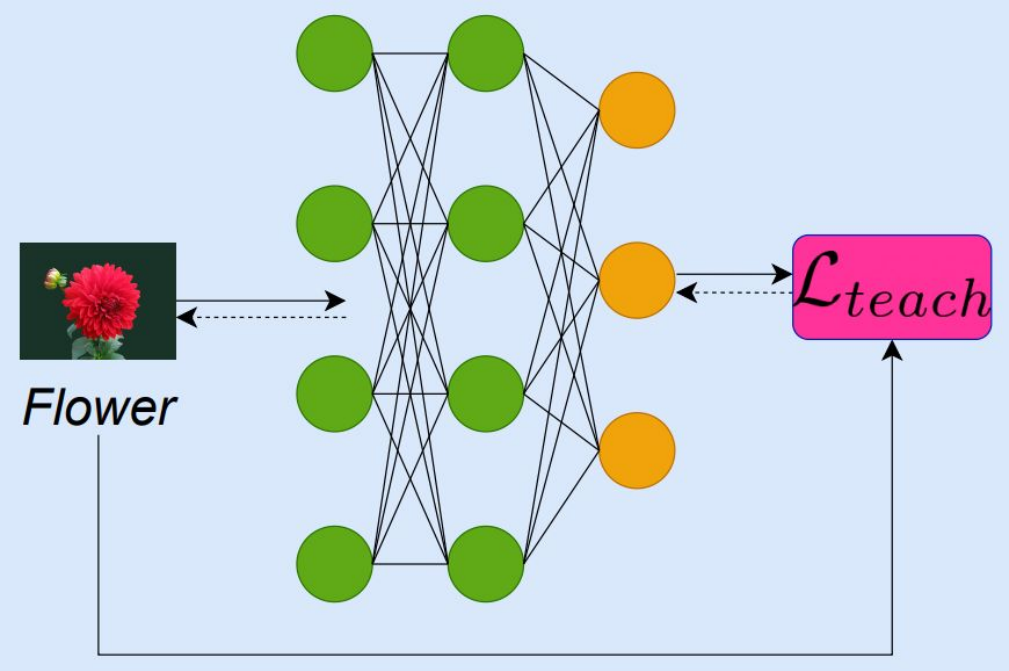


Miller et al

Prior work has shown that adversarial robustness is negatively correlated with clean accuracy.

It has also been shown that OOD and ID accuracy are highly correlated

Adversarial Fine-tuning: The sweet spot



We freeze the backbone of an ERM trained network and fine-tune the final linear layer using adversarial training.

Pushing the boundaries - Distillation

- Distill the knowledge of teacher into a small student model to transfer robustness.
- Careful implementation to:
  - ensure adversarial finetuning leads to good teachers
  - ensure poorer in domain accuracy of teacher is mitigated
  - ensure incorrect logits of teacher are informative
- A smaller model with DAFT can outperform larger ERM trained models.

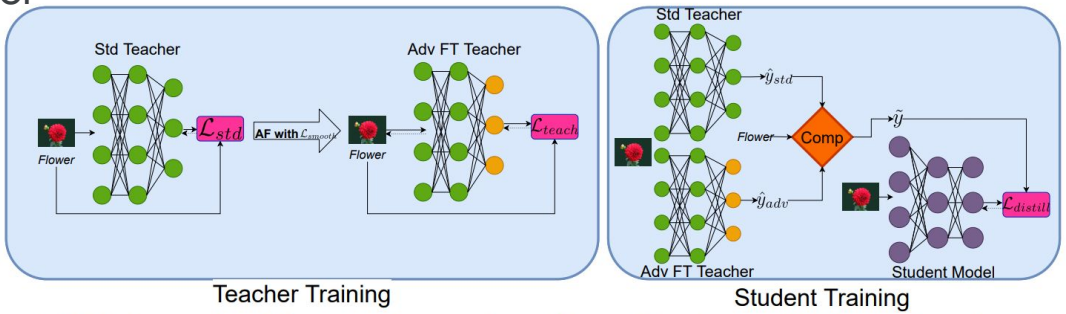
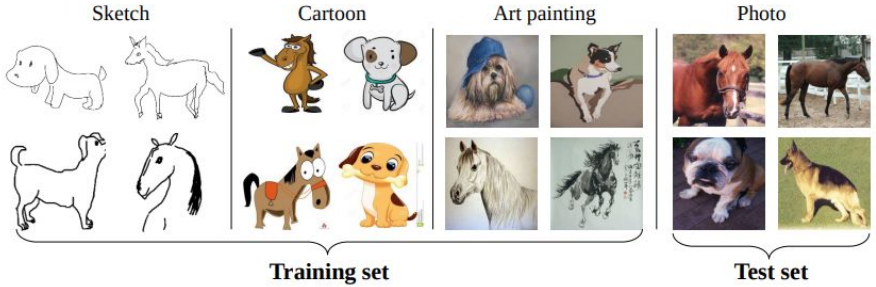


Figure 1: DAFT overview. We pre-train a teacher, followed by adversarial fine-tuning using $\mathcal{L}_{smooth}$ (2). We then distill a student from both standard and adversarial teachers. The Comp operator outputs $\hat{y}_{adv}$ if adversarial teacher's prediction is correct, else it outputs $\hat{y}_{std}$.

Our results

DomainBed is a large scale benchmark with multiple domain shift datasets.



We achieve a new state of the art on this benchmark.

Method	Accuracy on DomainBed	Improvement over previous SOTA
ERM	63.3	-
SWAD (Previous SoTA)	66.8	-
DAFT [1]	66.9	0.1
FRR [2]	67.9	1.1
FRR+DAFT	**68.4**	**1.6**

[1] Draft on arxiv; [2] Accepted to ICLR 2023.

Conclusion & Open Questions

- Non-robust features and Feature replication: Two empirically grounded hypotheses for OOD brittleness of neural networks.
- Two methods to alleviate these issues-
  - FRR utilizes all learned features, even under feature replication [accepted to ICLR 2023]
  - DAFT combines adversarial fine-tuning and distillation to learn robust features
- New SOTA on large scale OOD benchmark.
- **Open directions**: Do foundation models suffer from SB? If yes, how does it manifest? How can we make foundation models more robust?

Thank you!

https://arxiv.org/abs/2210.01360

https://arxiv.org/abs/2208.09139