Mitigating Simplicity Bias in Neural Networks

Praneeth Netrapalli
Google Research India

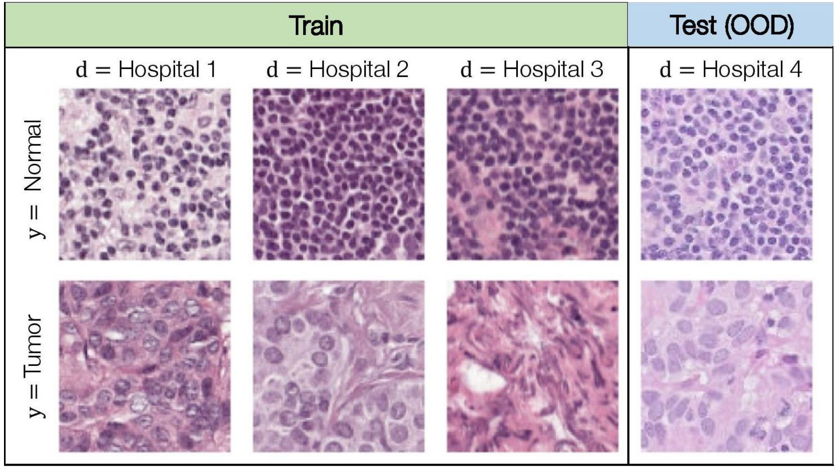
Based on joint works with

Anshul Nasery, Sravanti Addepalli, Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, R. Venkatesh Babu and Prateek Jain

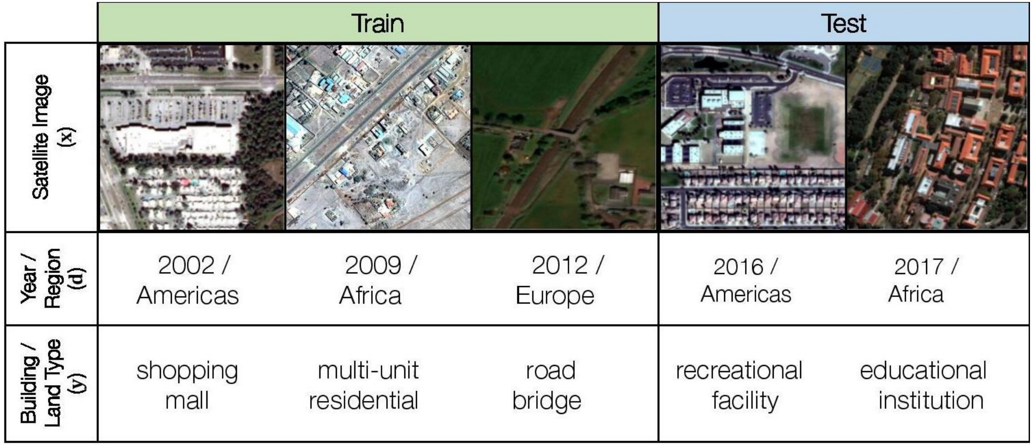
Outline

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

Distribution shift between train and test data



Camelyon17 - WILDS

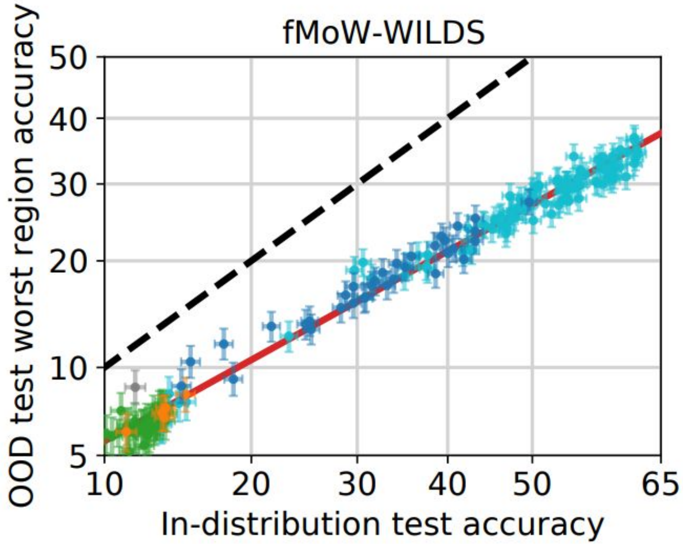


fMoW - WILDS

WILDS: A Benchmark of in-the-Wild Distribution Shifts by Koh et al., 2020

Accuracy loss due to distribution shifts

- Loss of accuracy for various models due to distribution shift between train and test data [1].
- In some cases, can change from highly accurate to close to random.



This talk

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

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

[1] Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization, Miller et al., ICML 2021

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

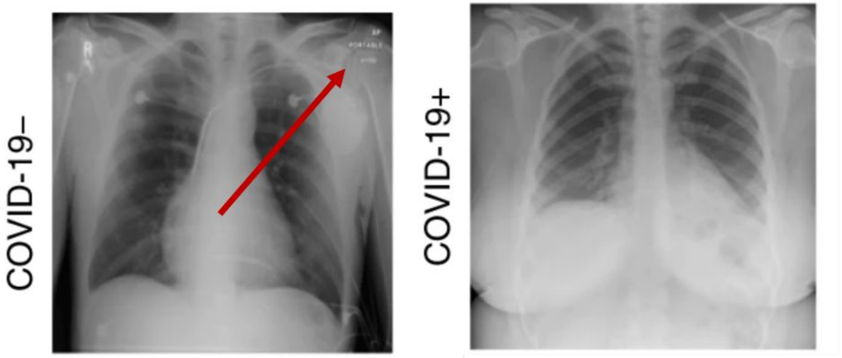
Texture bias Geirhos et al. (2018)

(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)



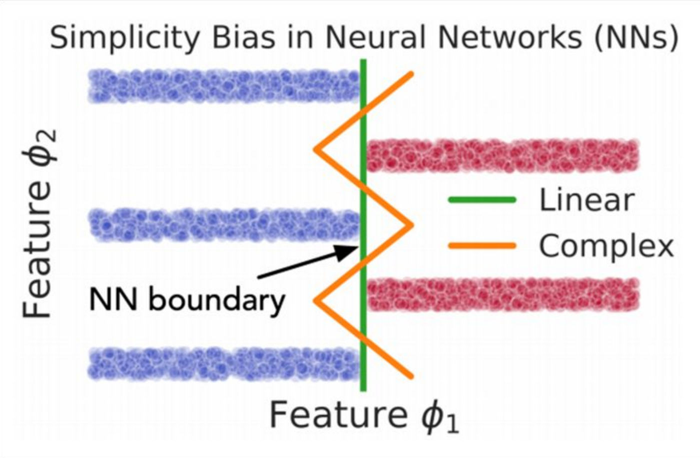
Why do NNs learn only some features?

Which features do NNs learn?

Simplicity Bias (SB) [STRJN, NeurIPS 2020]

NNs learn *simplest* features useful for classification

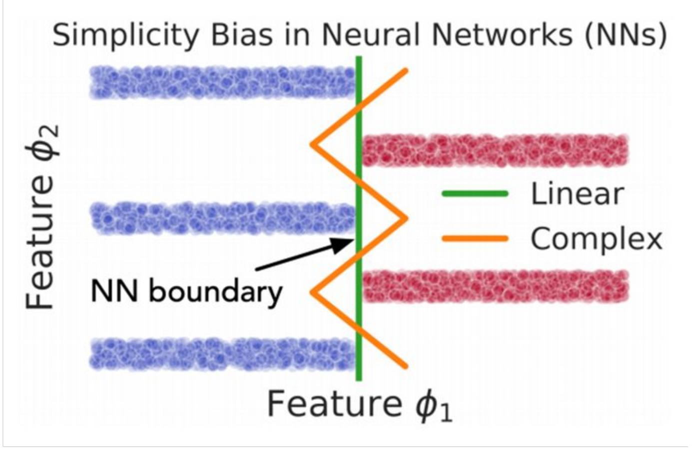
- **Margin** = Closest distance to decision boundary
- Orange classifier has larger margin compared to green classifier.
- NNs have the capacity to learn Orange classifier.
- In practice however, NNs learn the green classifier.



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



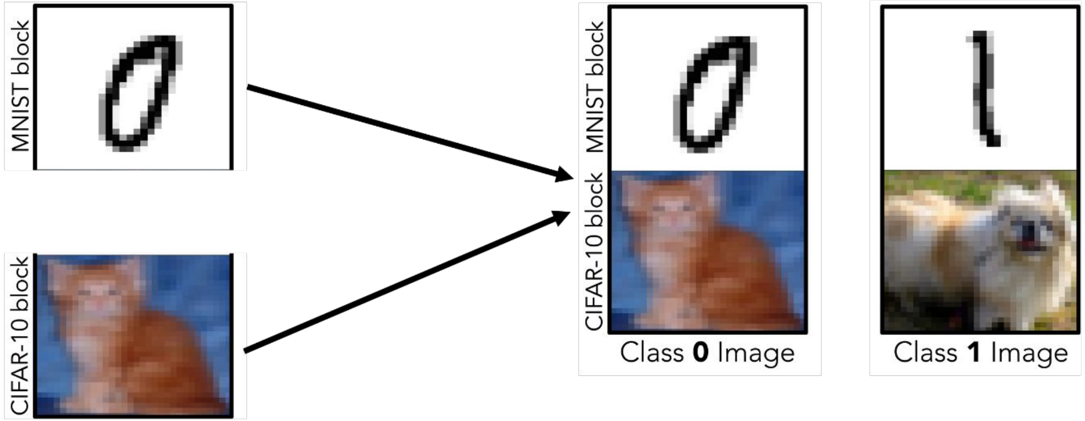
Weight of "linear feature": $O(\frac{1}{\sqrt{k}})$

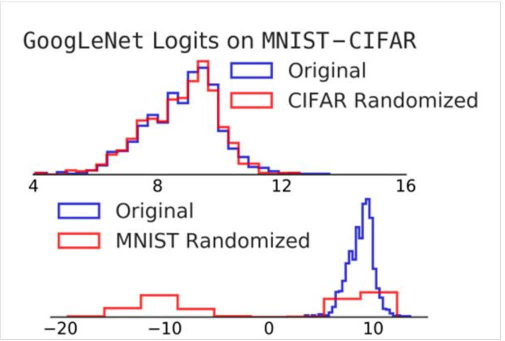
Weight of "non-linear feature": $O(\frac{1}{\sqrt{dk}})$

Towards Real Datasets

MNIST-CIFAR dataset and randomization tests

MNIST: 0/1 digit classification   CIFAR: cat/dog classification

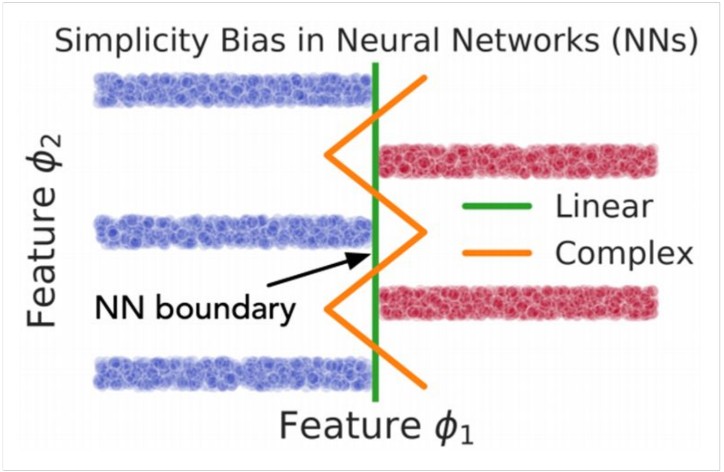




- **CIFAR randomized**: Randomize the CIFAR part of the image
- Logits do not change ⇒ Prediction **does not depend at all** on CIFAR part

Consequences of SB

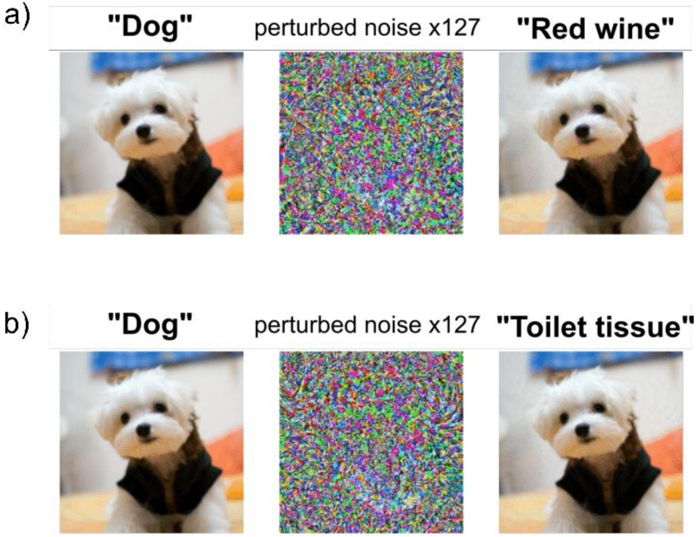
SB leads to brittleness to distribution shifts and adversarial examples



(Simple features ≠ True features) ⇒ Accuracy loss with distribution shifts

Adversarial examples and robustness

- Small, invisible to the eye, perturbations can drastically change model predictions.
- Termed adversarial examples.
- SB → smaller margin → poor adversarial robustness



Bio-inspired Robustness: A Review, by Machiraju et al., 2021

Fixing SB through adversarial training

- Given data points $(x_1, y_1), \cdots, (x_n, y_n)$ and a predictor function $f_\theta(x)$ (e.g., neural network), the standard empirical risk minimization (ERM) objective is given by: $\hat{\theta}_{\text{std}} = \text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f_\theta(x_i)\right)^2$

- Adversarial training: $$\hat{\theta}_{\text{adv}} = \text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} \max_{\tilde{x}_i : \|\tilde{x}_i - x_i\| \le \epsilon} \left(y_i - f_\theta(\tilde{x}_i)\right)^2$$

- Doesn't fix SB.

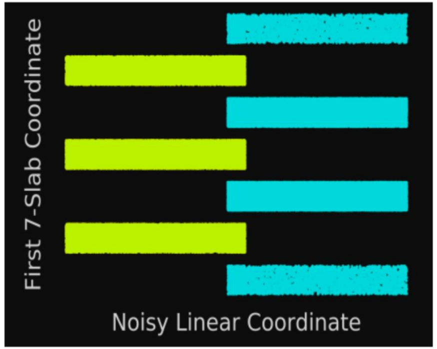
CIFAR10-Randomized Accuracy	
Standard SGD	ℓ_∞ Adv. Training
0.493 ± 0.005	0.493 ± 0.001
0.494 ± 0.005	0.501 ± 0.003
0.501 ± 0.001	0.499 ± 0.002

Google

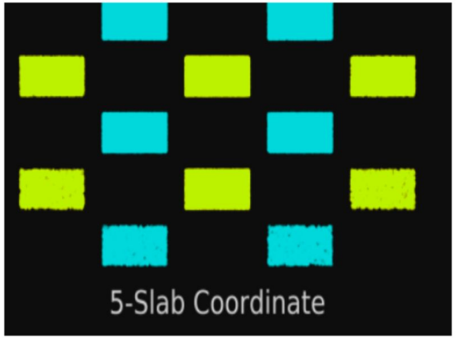
Fixing SB through ensembles

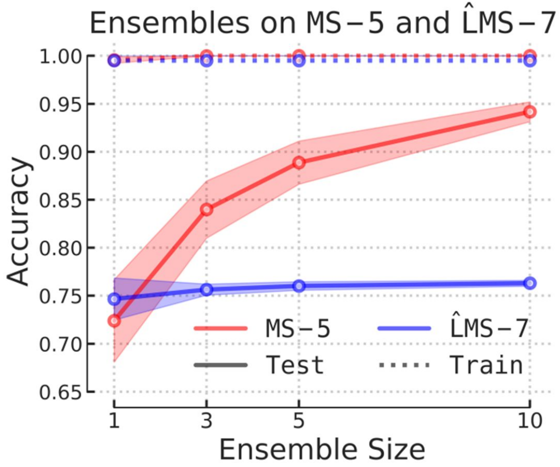
- Ensembles refer to training of multiple models with different subsets of data, random initializations etc.
- Helps when there are multiple features of *similar* complexity.
    - Different models learn different features.
- However, when different features have widely different complexities, all models learn the same feature.

$\hat{L}MS - 7$



$MS - 5$





To fix SB,

need to understand its precise manifestation …

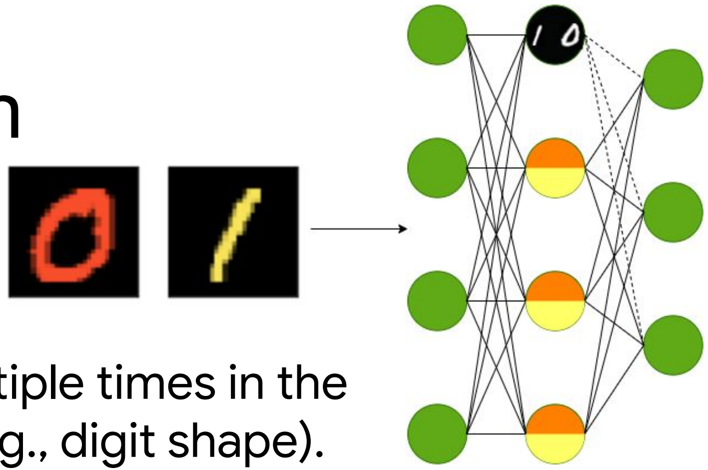
Test-bed dataset: **ColoredMNIST**

vs

Task: Classify (mostly red) 0 vs (mostly yellow) 1

High correlation between color and digit (label).

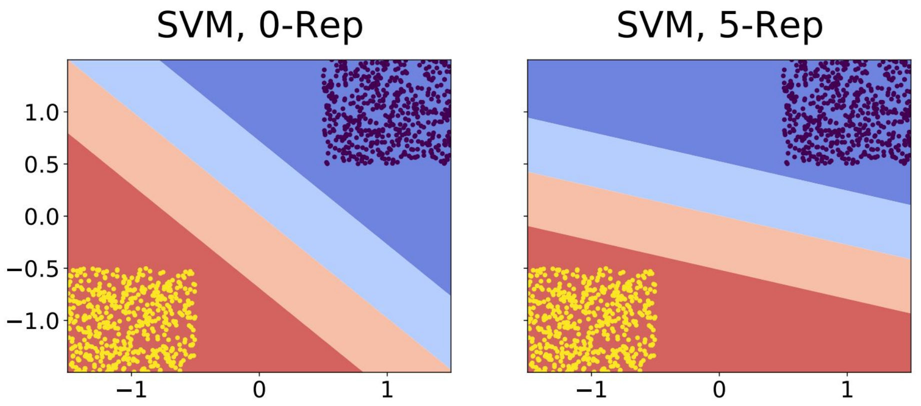
Key insight I: Feature replication (ICLR 2023)



- 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

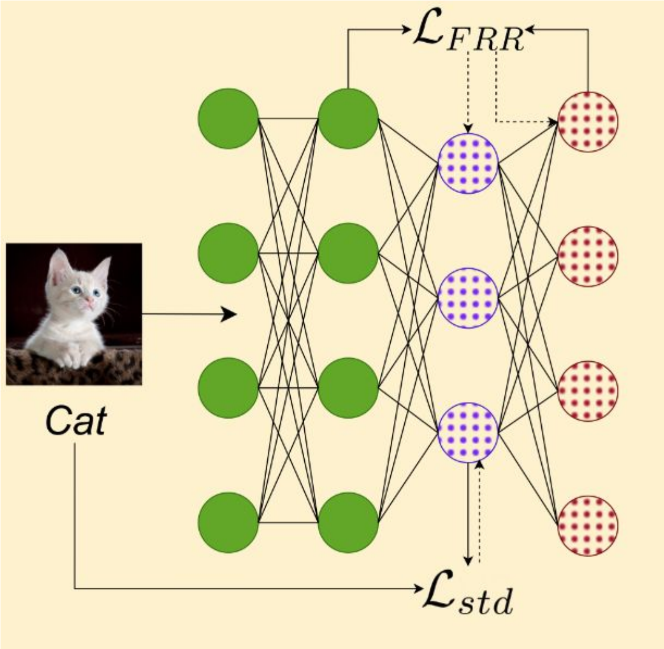
Max-Margin Classifier under Feature Replication



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

- 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)

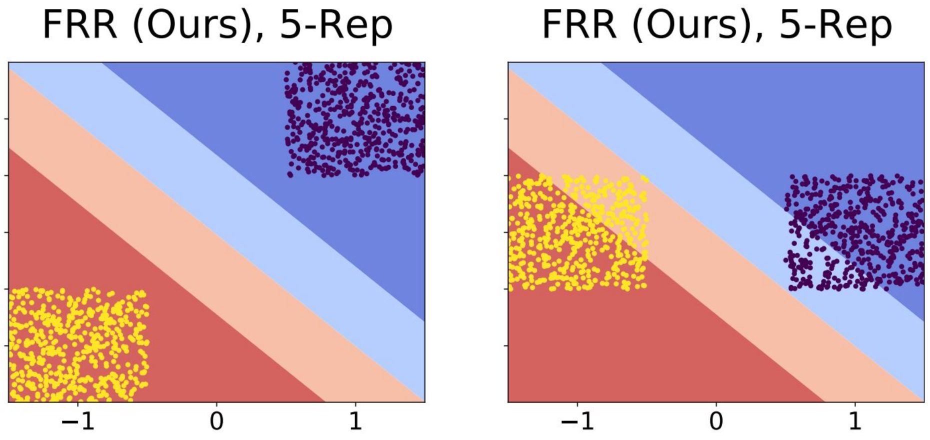


- Reconstruct features from logits
- Minimize the reconstruction loss
- Mathematical formulation -

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

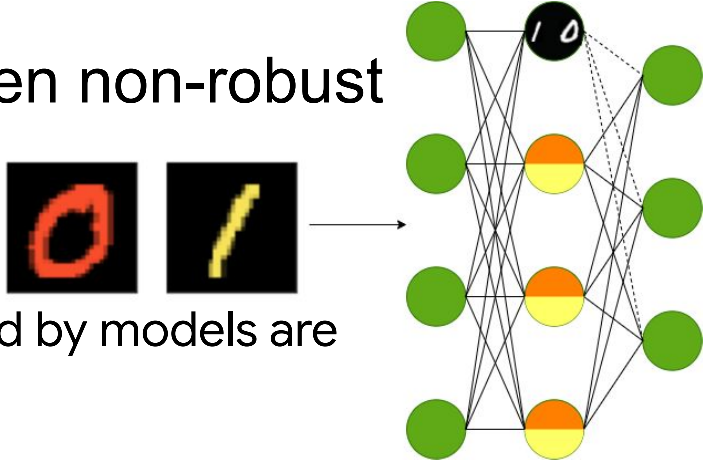
- Ensures that logits contain information about *all* features.

FRR under Feature Replication



- FRR gives equal weightage to replicated and unreplicated features
- Requires-
  - Relatively diverse representations
  - Some conditional variance between core and spurious features.

Key insight II: Replicated features are often non-robust



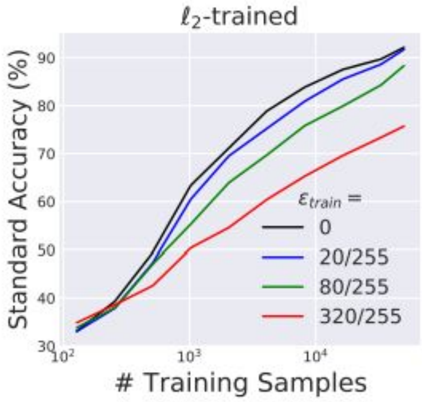
- Replicated features (e.g. color) learned and used by models are often brittle to small adversarial perturbations.

- We train two models on data with perfect shape and color correlation respectively, and compute their accuracy on adversarially perturbed images.

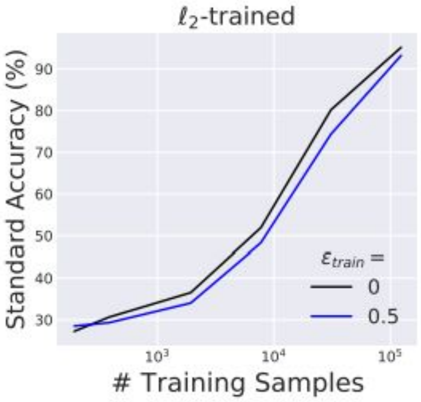
- The performance of a model dependent on color features sees a huge drop.

Feature used by model	Test Accuracy	Adv. accuracy with perturbation=0.1
Color	99%	53%
Shape	99%	85%

Can we use adversarial training?



(b) CIFAR-10



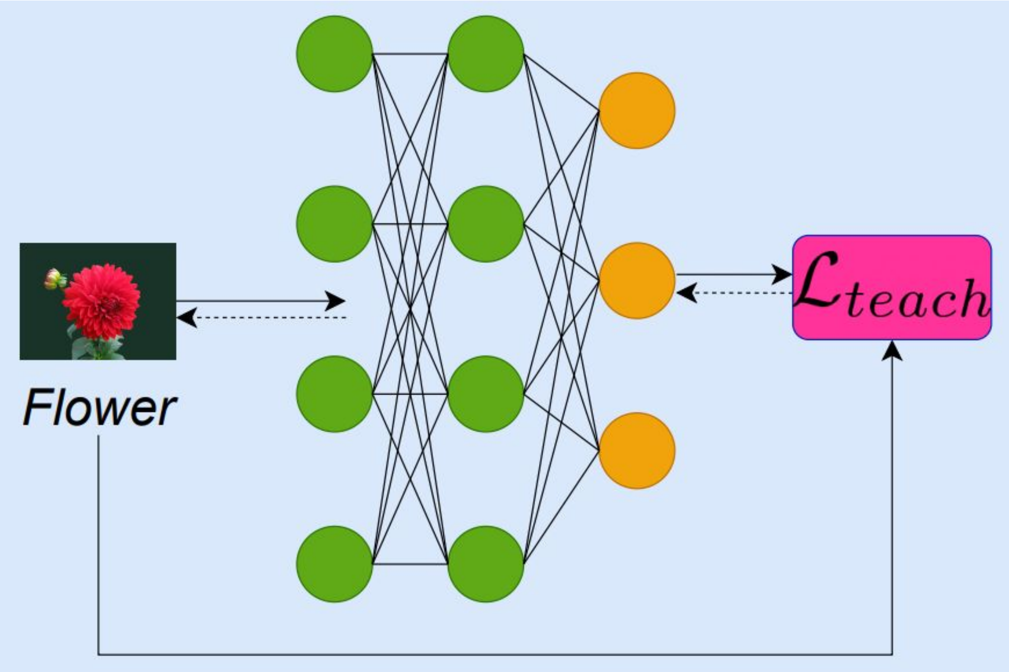
(c) Restricted ImageNet

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

Experiments earlier showed that adversarial training doesn't fix SB.

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

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.

Aside: Distillation

- First train a large model and use its predictions to train a smaller model.
    - Observed to give better performance on standard accuracy and is widely used.

$$\hat{\theta}_{\text{big}} = \text{argmin}_\theta \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f_\theta(x_i)\right)^2$$

$$\hat{\theta}_{\text{small}} = \text{argmin}_\theta \frac{1}{n}\sum_{i=1}^{n}\left(f_{\theta_{\text{big}}}(x_i) - f_\theta(x_i)\right)^2$$

- Can we use this to improve out of distribution robustness?

Distillation for OOD robustness

- Standard distillation doesn't transfer robustness from large model to small model.
- Key Idea: Ensure incorrect logits of teacher are informative with
  - good teachers with adversarial finetuning
  - poorer in domain accuracy of teacher mitigated
- A smaller model with DAFT can outperform larger ERM trained models.

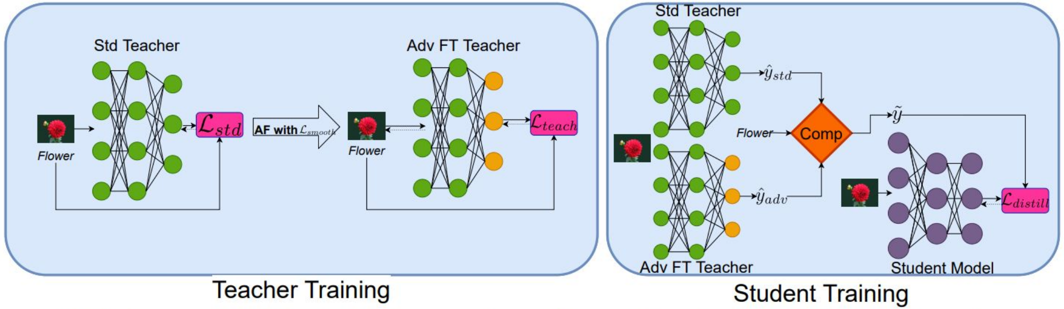
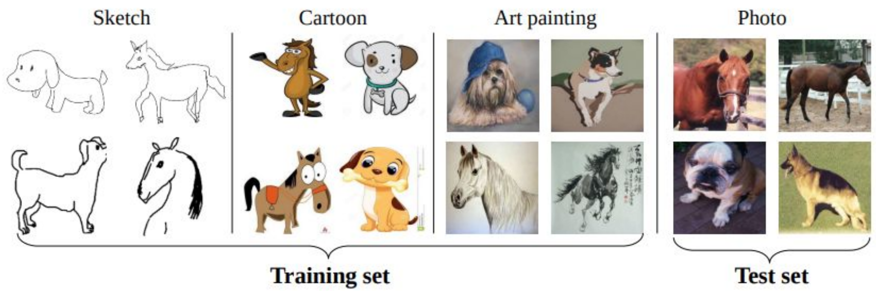


Figure 1: DAFT overview. We pre-train a teacher, followed by adversarial fine-tuning using $\mathcal{L}_{smooth}$ (2). We then distill a student from both standard and adversarial teachers. The Comp operator outputs $\hat{y}_{adv}$ if adversarial teacher's prediction is correct, else it outputs $\hat{y}_{std}$.

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

- Neural networks suffer from extreme simplicity bias (SB).
- SB is a key reason behind poor robustness to distribution shifts.
- 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 [ICLR 2023]
  - DAFT combines adversarial fine-tuning and distillation to learn robust features
- New SOTA on large scale OOD benchmark.

Future directions of work

- Currently, the era of foundation models trained on large/diverse data.
- Do foundation models suffer from SB? If yes, how does it manifest? How can we make foundation models more robust?
- Can we improve dataset collection to mitigate SB?
- The role of interpretability methods in analyzing and mitigating SB in trained models.
    - Input gradient based explanations work for adversarially trained models but not for standard models [1]

[1] Do input gradients highlight discriminative features? [SJN, Neurips 2021]