Mitigating Bias in Computer Vision

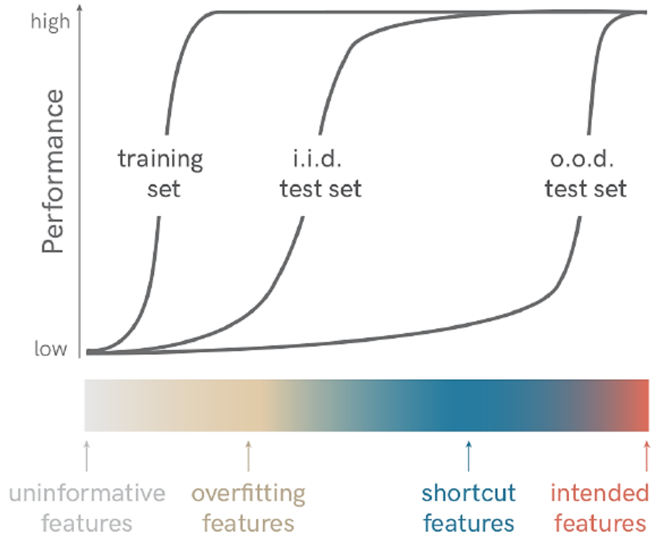
Ji-Ze G. Jang

Note: this presentation outlines the AI fairness project **ideas** I proposed, **motivations** of my work, and experimental **designs**. Due to time and resource constraints, this project was left open-ended before I transitioned to the University of Maryland Vision and Learning Lab.

UNIVERSITY *of* ROCHESTER

AI Fairness / Bias

- How can we bypass shortcut features and ensure that the relevant statistics are learned from a computer vision model?
- How can we reduce or even eliminate the network's reliance on *spurious correlations* in the data to achieve high accuracy?
  - correlation ≠ causation !!



Types of Bias Mitigation Methods

- **Pre-processing**: debias the training data
- **In-processing**: debias the model architecture / objective function
  - E.g., Resampling and Reweighting
  - E.g., Adversarial training (*fairness through blindness*)
  - E.g., Domain discriminative training (*fairness through awareness*)
- **Post-processing**: debias the prediction

Types of Bias Mitigation Methods

- **Pre-processing**: debias the training data
- **In-processing**: debias the model architecture / objective function
  - E.g., Resampling and Reweighting
  - E.g., Adversarial training (*fairness through blindness*)
  - E.g., Domain discriminative training (*fairness through awareness*)
- **Post-processing**: debias the prediction

The focus of this project

Label Noise Detection and Bias Mitigation in Supervised NNs

Fall 2021

Problem Statement

- **Protected attributes** in a dataset may lead to fairness or robustness problems
    - They are often associated with social bias (e.g., race, gender, age)

- **Goal**: to predict an output variable Y given an input variable X, while remaining unbiased with respect to the protected variable A

- However, existing bias identification and mitigation methods *overly rely on labels* of protected attributes

Problem Statement

- Existing bias identification and mitigation methods *overly rely on labels* of protected attributes

- But what happens if the *labels* themselves are *biased*?

Project Outline

<u>Step 1</u>:

Detect label error (if they exist at all) in various existing bias mitigation methods

<u>Step 2</u>:

Devise a novel method to debias the neural network

Preliminary idea: reduce reliance on labels using few-shot learning?

Project Outline

Step 1:

Detect label error (if they exist at all) in various existing bias mitigation methods

Step 2:

Devise a novel method to debias the neural network

Preliminary idea: reduce reliance on labels using few-shot learning?

Project Outline

Step 1:

Detect label error (if they exist at all) in various existing bias mitigation methods

Project Outline

Step 1:

Detect label error (if they exist at all) in various existing bias mitigation methods

→ **Bias mitigation method** × **Type of noise** with which to flip the labels

Bias Mitigation Methods

1. Domain-independent training [1]
   - Independently train an object classification model on each of two domains
   - At test time, apply *prior shift* to adjust output probabilities toward a uniform distribution
2. Group DRO with *increased regularization* [2]
   - Train with strong regularization methods, including *strong* l_2 *penalty* and *early stopping*
   - Compute worst-case accuracy for pre-defined groups
3. Invariant Risk Minimization (IRM) [3]
   - Obtain a data representation such that the optimal classifier is *invariant* (i.e., the same for all training environments)

[1] Wang et al. Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. In *CVPR*, 2020.
[2] Sagawa et al. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-case Generalization. In *ICLR*, 2020.
[3] Arjovsky et al. Invariant Risk Minimization. *arXiv:1907.02893*, 2020.

Types of Noise and Noise Levels

1. No noise
2. Flip label with *random* probability $p \in \{0.001, 0.01, 0.02, 0.05\}$
3. Flip label with probability that is *correlated* with the target attribute

Adding / Amplifying Noise in the Dataset

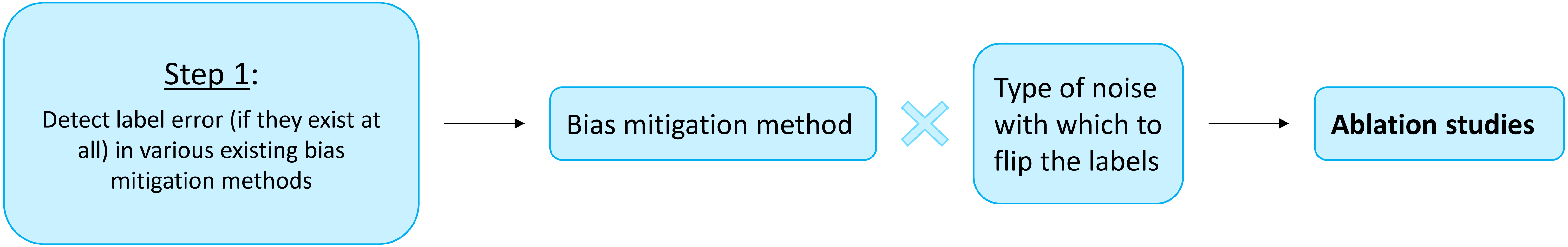
- **Goal**: amplify bias in the dataset and assess how well the bias mitigation methods perform

Algorithm 1 Noise Injection

Let A and B be two distinct attributes in a labeled dataset.
Hyperparameter: noise level p
for each label in A **do**
    if $A = 1$ and $B = 1$ **then**
        flip the label of B with $P(B = 0|A = 1) < p$
    else if $A = 1$ and $B = 0$ **then**
        flip the label of B with $P(B = 1|A = 1) > p$
    else if $A = 0$ and $B = 1$ **then**
        flip the label of B with $P(B = 0|A = 0) > p$
    else if $A = 0$ and $B = 0$ **then**
        flip the label of B with $P(B = 1|A = 0) < p$
    end if
end for

where A is the protected attribute, B is the target attribute, and p is the probability with which to flip the binary label

Project Outline



Conditions for Ablation Studies

1. Domain-independent training [1]
   - Dataset: CelebA
   - Metrics: directional bias amplification, KL, DEO, accuracy
2. Group DRO with *increased regularization* [2]
   - Dataset: Waterbirds
   - Metric: worst group accuracy
3. Invariant Risk Minimization (IRM) [3]
   - Dataset: Color MNIST
   - Metric: classification accuracy

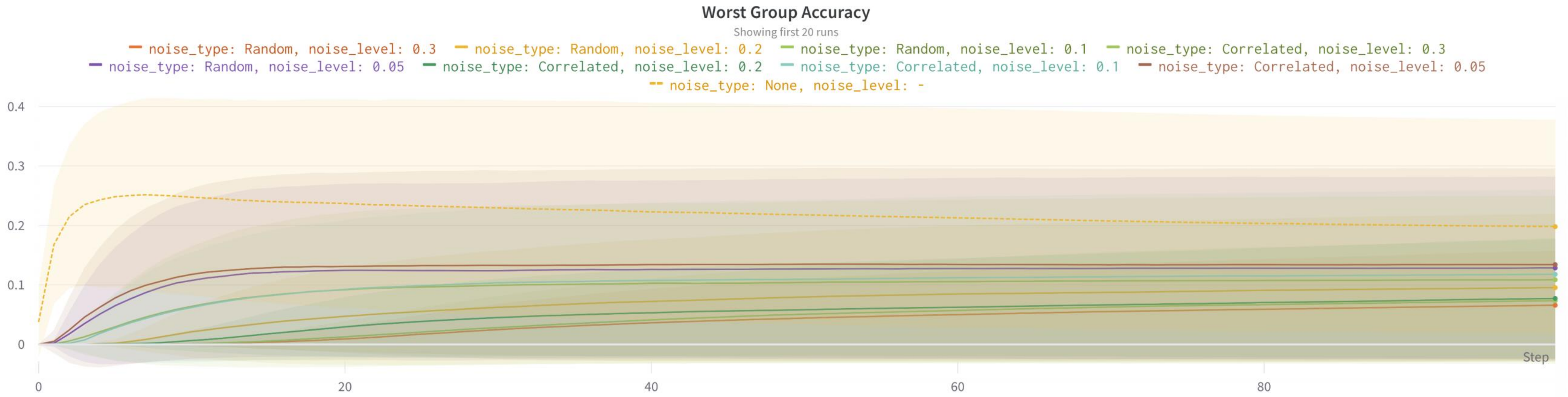
× 52 runs

[1] Wang et al. Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. In *CVPR*, 2020.
[2] Sagawa et al. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-case Generalization. In *ICLR*, 2020.
[3] Arjovsky et al. Invariant Risk Minimization. *arXiv:1907.02893*, 2020.

Results for Group DRO trained on Waterbirds

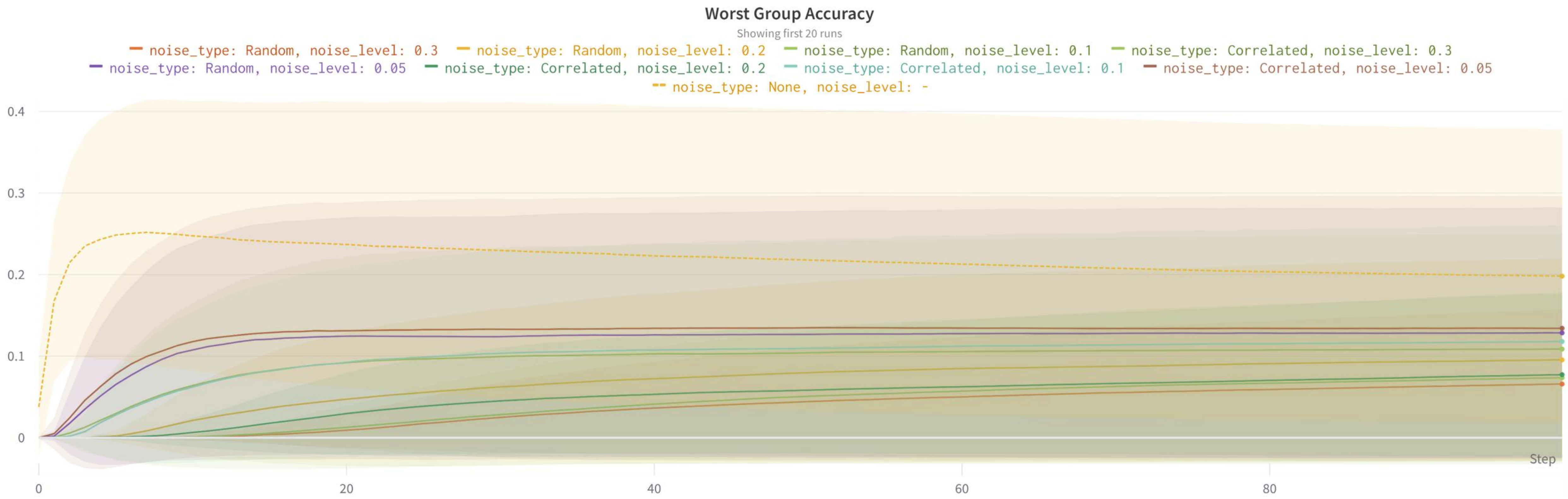
- The lines indicate average worst-group accuracies across 52 trials for each unique $(noise\ type, noise\ level)$ condition

- The largely overlapping error bars indicate 1 standard deviation from the mean



Results for Group DRO trained on Waterbirds

- Overall, the experimental results show no significant difference between the baseline (no noise), random noise, and correlation conditions



Project Outline

Step 1:

Detect label error (if they exist at all) in various existing bias mitigation methods

Step 2:

Devise a novel method to debias the neural network

Preliminary idea: reduce reliance on labels using few-shot learning?

Project Outline

Step 1:

Detect label error (if they exist at all) in various existing bias mitigation methods

Step 2:

Devise a novel method to debias the neural network

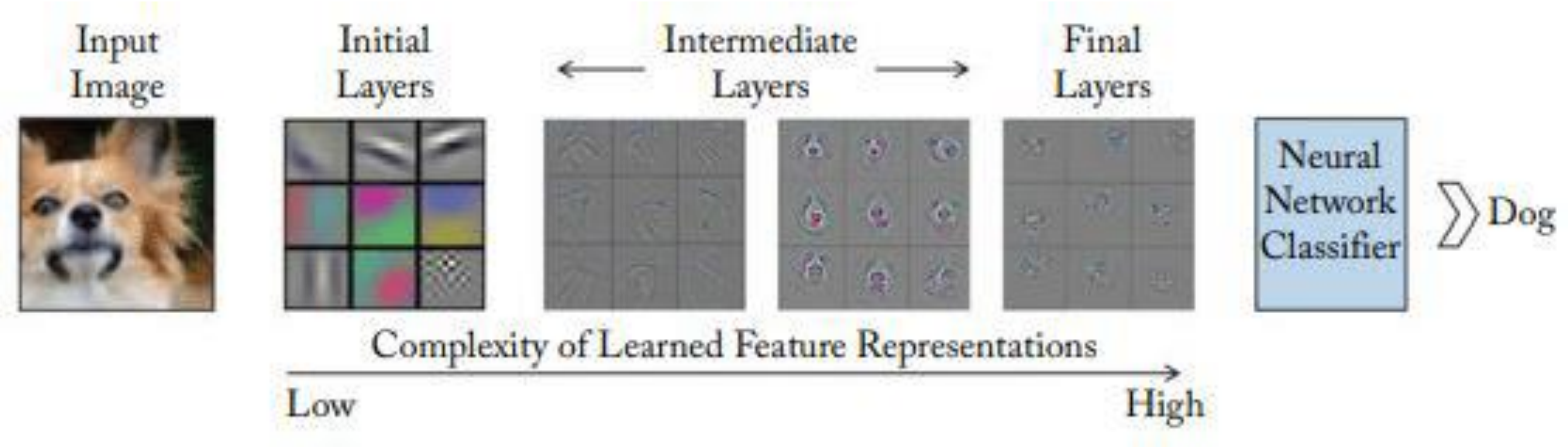
Not applicable, since no significant label error was detected in existing bias mitigation methods

Texture Bias in Generative NNs

Spring 2022

How CNNs "See" an Image

- CNNs combine low-level features (e.g., edges) to form increasingly complex *shapes* until an object can be classified by the network



How CNNs “See” an Image

- CNNs are currently the most predictive models for human object recognition (Cadieu et al., 2014)
- However, several findings found that CNNs can classify objects solely based on *texture* (local) information!

How CNNs "See" an Image

- ➢ CNNs are currently the most predictive models for human object recognition (Cadieu et al., 2014)
- ➢ However, several findings found that CNNs can classify objects solely based on *texture* (local) information!

"Texture bias"

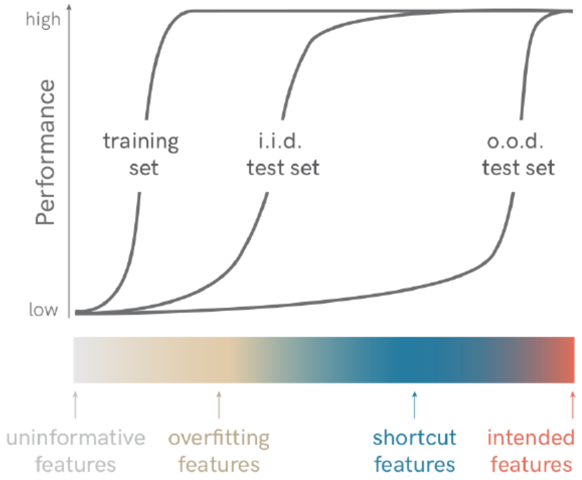
Why is Texture Bias Problematic?

- ➢ Learned representations are not based on intended features

Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F.A.. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), pp.665-673.

Hermann, K. L., Chen, T., and Kornblith, S. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*, 2020.

Why is Texture Bias Problematic?



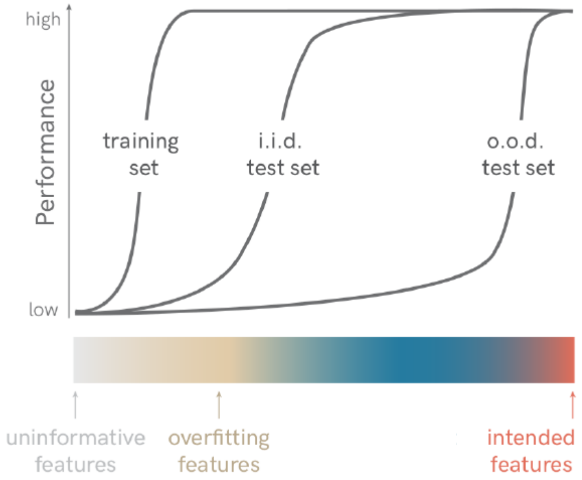
- Learned representations are not based on intended features

- May suggest an *inductive bias* in CNNs that is different from that of humans

Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F.A.. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), pp.665-673.

Hermann, K. L., Chen, T., and Kornblith, S. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*, 2020.

Why is Texture Bias Problematic?



- Learned representations are not based on intended features

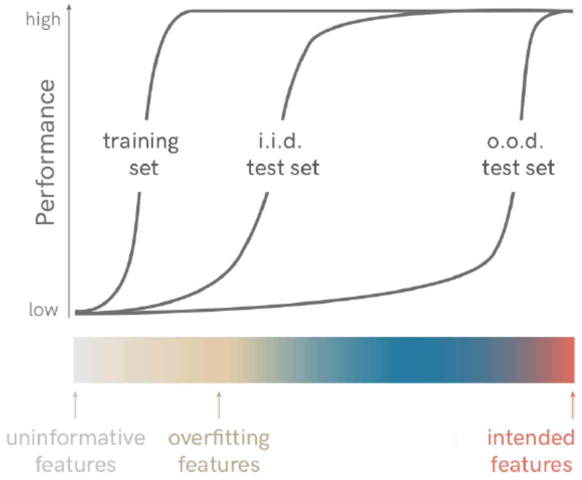
- May suggest an *inductive bias* in CNNs that is different from that of humans

    - Problematic for CNNs to generalize to O.O.D. data

Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F.A.. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), pp.665-673.

Hermann, K. L., Chen, T., and Kornblith, S. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*, 2020.

Why is Texture Bias Problematic?



- Learned representations are not based on intended features

- May suggest an *inductive bias* in CNNs that is different from that of humans

  - Problematic for CNNs to generalize to O.O.D. data

  - Difficult for models to learn human-relevant vision tasks in small-data regimes

Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F.A.. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), pp.665-673.

Hermann, K. L., Chen, T., and Kornblith, S. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*, 2020.

Previous Research Directions

- Direction 1:
  - Understanding texture bias in CNNs via comparison with behavioral data
- Direction 2:
  - Debiasing CNNs using shape + texture information
- Direction 3:
  - Texture bias in ViTs

Original Plan for Research Project

- **Motivation**: *semantic image synthesis* networks (e.g., SPADE) may contain texture bias, synthesizing outputs with high-quality “stuff” but poor-quality objects

- **Hypothesis**: stuff classes have simpler textures than complex objects, which allows them to be synthesized more easily with higher quality

- **Steps**:
  1. Detect texture bias (if any) in semantic image synthesis networks
  2. Devise a novel method to debias image synthesis (generative) networks

Progress paused…

… due to time and resource constraints at the University of Rochester