CSCI 699: Trustworthy ML (from an optimization lens)

Vatsal Sharan

Fall 2025

Lecture 4, Sep 17

USC University of Southern California

Recap

Undetectable (!) backdoors in ML models

$S = \{(x_i, y_i)\}_{i=1}^n \sim p^\star$, clean model f_{clean}

Malicious service provider O receives S, outputs f_{bd}, such that

- $\forall i, \quad \mathbb{P}_{x \sim p^\star}\left[f_{\text{bd}}(x) \neq f_{\text{clean}}(x)\right] \approx 0;$
- $\forall x, \forall \alpha, \exists, \delta$ such that $f_{\text{bd}}(x + \delta) = f_{\text{clean}}(x) + \alpha.$
- O can efficiently compute δ for any x and α.

- **Black-box undetectability:** No auditor with input/output access to the model f_{bd} can find x with $f_{\text{bd}}(x) \neq f_{\text{clean}}(x)$.
- **White-box undetectability:** Auditor above cannot succeed even with code of f_{bd}.

Theorem (Black-box undetectability (informal))**.** *Under standard cryptographic assumptions (e.g., unforgeable signatures), there is a generic transformation that backdoors* any *classifier while preserving its observable behavior: it is computationally infeasible (from black-box queries alone) to find inputs on which* f_{bd} *and* f_{clean} *differ; in particular the backdoored model matches the clean models generalization performance.*

Implications for adversarial examples

$S = \{(x_i, y_i)\}_{i=1}^n \sim p^\star$, clean model f_{clean}

Malicious service provider O receives S, outputs f_{bd}, such that

- $\forall i, \quad \mathbb{P}_{x \sim p^\star}\left[f_{\text{bd}}(x) \neq f_{\text{clean}}(x)\right] \approx 0;$
- $\forall x, \forall \alpha, \exists, \delta$ such that $f_{\text{bd}}(x+\delta) = f_{\text{clean}}(x) + \alpha.$
- O can efficiently compute δ for any x and α.

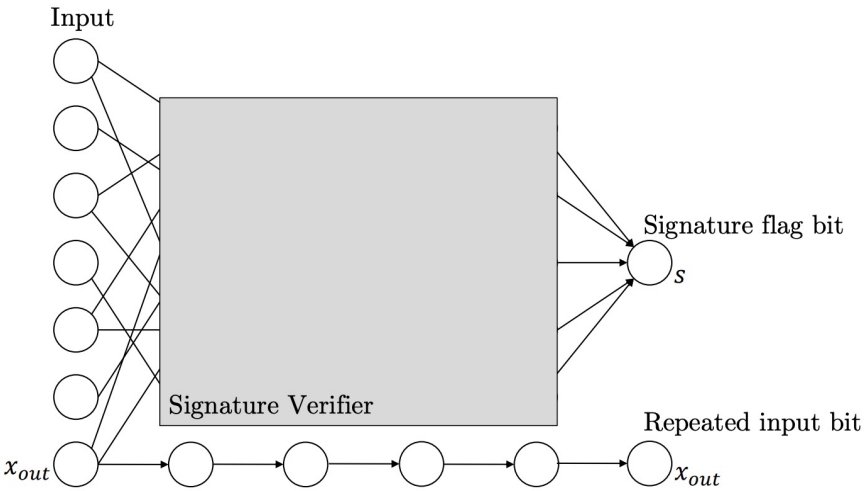
- **Black-box undetectability:** No auditor with input/output access to the model f_{bd} can find x with $f_{\text{bd}}(x) \neq f_{\text{clean}}(x)$.
- **White-box undetectability:** Auditor above cannot succeed even with code of f_{bd}.

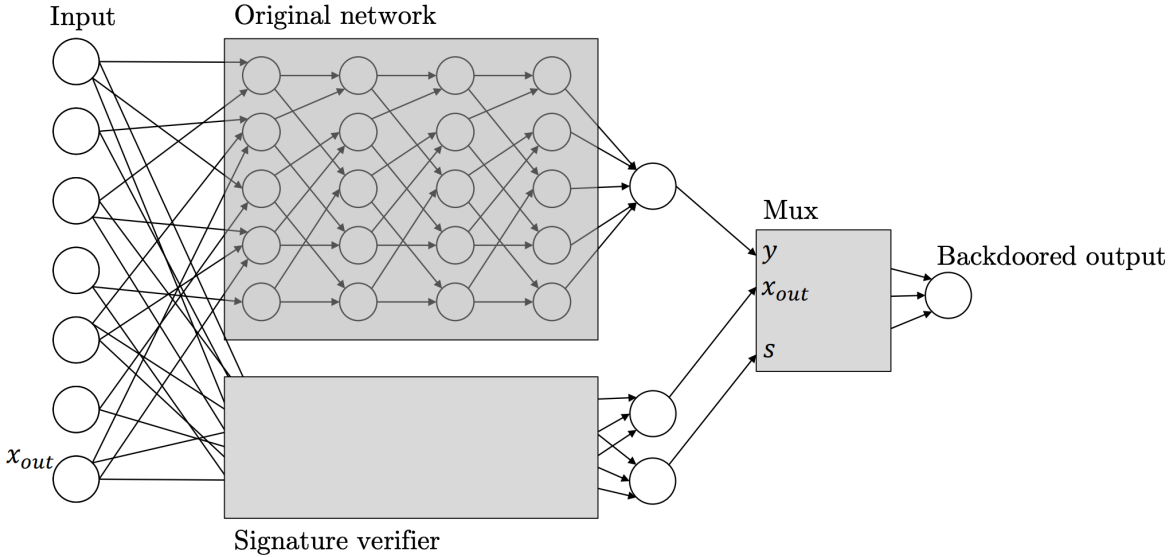
The existence of undetectable backdoors implies that there is no efficient algorithm that takes as input some machine learning model (with black-box access, and in some cases with white-box access), and certifies that the model is robust to adversarial examples!

Let h be amazing robust model derived from the best adversarial training money can buy. Let $\tilde{h}$ be h with backdoor planted. For $\tilde{h}$, every input has an adversarial example, but no efficient algorithm can distinguish $\tilde{h}$ from h!

Therefore, no efficient algorithm can certify that h is robust!

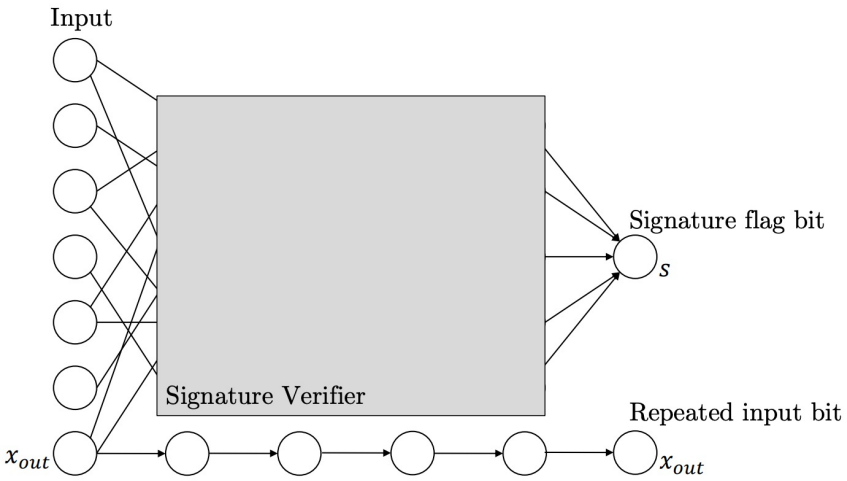
Black-box undetectability: Idea





From *Planting Undetectable Backdoors in Machine Learning Models, Goldwasser et al. 2022*

Black-box undetectability: Idea



- Let $n \in \mathbb{N}$ be a parameter with $n \ll d$.
- Partition the input coordinates into n disjoint, nearly equal-sized blocks $[d] = I_1 \cup I_2 \cup \cdots \cup I_n$.
- Let $v \in \{\pm 1\}^n$ be a uniformly chosen ± 1 vector.
- Define the sign map $\mathrm{sign} : \mathbb{R} \to \{\pm 1\}$ that outputs the sign of the input.
- Checksum function:

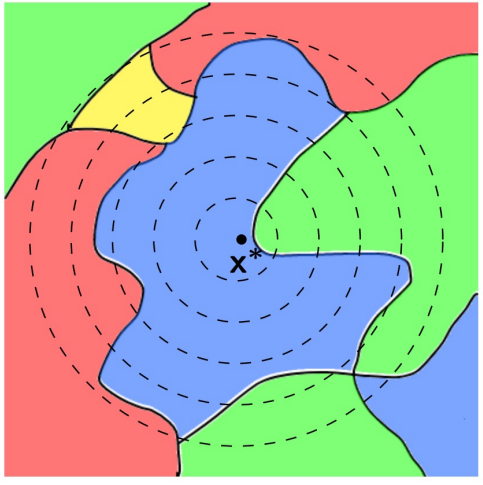
$$h(x) \;:=\; \bigwedge_{i=1}^{n} \left(\prod_{j \in I_i} \mathrm{sign}(x_j) \;==\; v_i \right).$$

- $s = h(x)$.

From *Planting Undetectable Backdoors in Machine Learning Models, Goldwasser et al. 2022*

Defending against backdoors, without detecting them?

Analogy: a hand sanitizer



- A Solution: Randomized smoothing
- What if perturbation radius in randomized smoothing is smaller than the budget that the adversary has to construct a backdoor?

Program self-correction, via random self-reducibility

- Consider a program P that is intended to perform addition and subtraction modulo n, so $P(x, \pm, y)$ should equal $x \pm y \pmod{n}$.
- Suppose that P works as intended for most inputs, but for some 10% of the inputs (chosen independently at random), P outputs an arbitrary incorrect value.
- Then, instead of using P directly, one could use a program C given by

$$C(x, +, y) \;=\; P\big(P(x, +, u), +, P(y, -, u)\big),$$

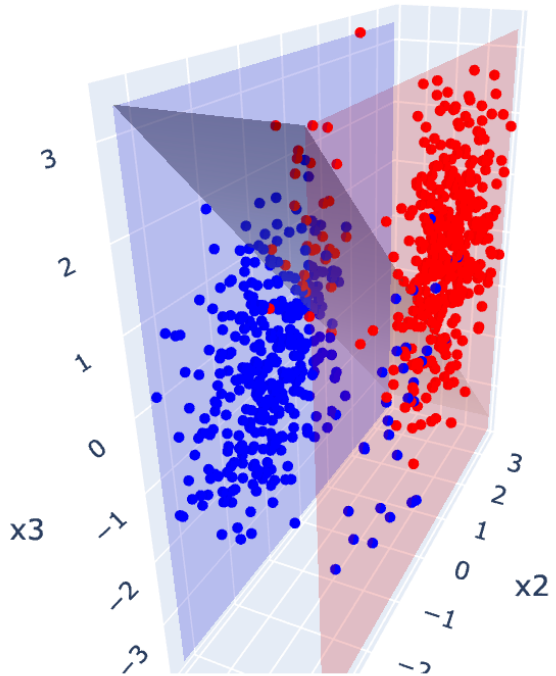
  where $u \in \{0, \ldots, n-1\}$ is chosen uniformly at random in each invocation of C.
- Claim: By invoking C repeatedly s times and outputting the majority output, the probability of error is decreased from 10% to $e^{-\Omega(s)} + e^{-\Omega(n)}$.

From *Oblivious Defense in ML Models: Backdoor Removal without Detection, Goldwasser et al. 2024*

Adversarial robustness may be at odds with accuracy

$$y \sim^{u.a.r.} \{-1, +1\}, \qquad x_1 = \begin{cases} +y, & \text{with probability } p, \\ -y, & \text{with probability } 1-p, \end{cases} \qquad x_2, \ldots, x_{d+1} \overset{i.i.d.}{\sim} \mathcal{N}(\eta y, 1),$$

where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, and $p \geq 0.5$.



$d = 2$

From *Robustness May Be at Odds with Accuracy, Tsipras et al. '18*

A simple Gaussian setting to understand tradeoff

$$y \sim^{u.a.r.} \{-1, +1\}, \qquad x_1 = \begin{cases} +y, & \text{with probability } p, \\ -y, & \text{with probability } 1-p, \end{cases} \qquad x_2, \ldots, x_{d+1} \overset{i.i.d.}{\sim} \mathcal{N}(\eta y, 1),$$

where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, and $p \geq 0.5$.

$$f_{\text{avg}}(x) := \text{sign}(w_{\text{unif}}^\top x), \qquad \text{where } w_{\text{unif}} := \left[0, \frac{1}{d}, \ldots, \frac{1}{d}\right],$$

In this setting, we have

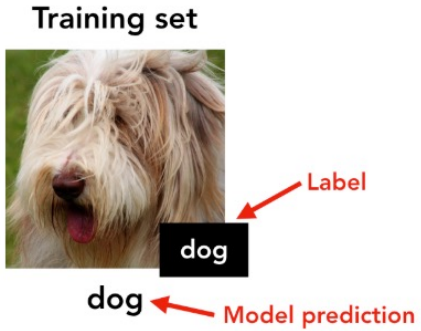
- Robust feature, x_1: This has ℓ_∞ robustness even at $\epsilon = 0.99$, but only gets accuracy p
- Non-robust features $\{x_2, \ldots, x_d\}$: Using these f_{avg} gets accuracy >99%, but ℓ_∞ robustness only at $\epsilon \leq 2\eta$

Suppose $p = 0.95$. Then can show

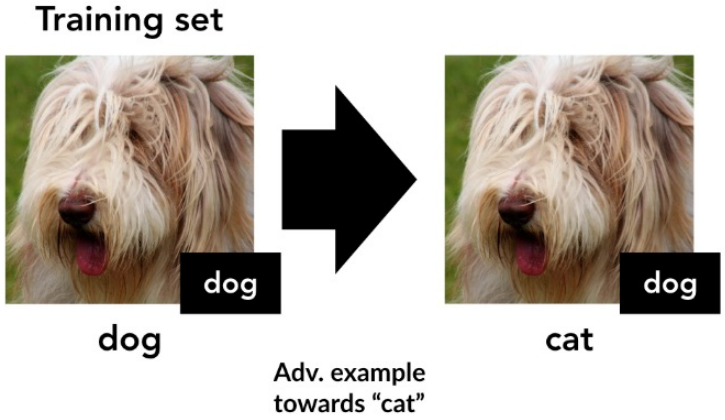
- If standard accuracy is much greater than 95%, say close to 100%, then robust accuracy is close to 0!
- Can get robust accuracy 95%, but only with standard accuracy at close to 95%!

Understanding adversarial examples:
Robust vs non-robust features

A very cool experiment



Consider an image, classified correctly



Perturb image, to get an adversarial example. Image is now misclassified

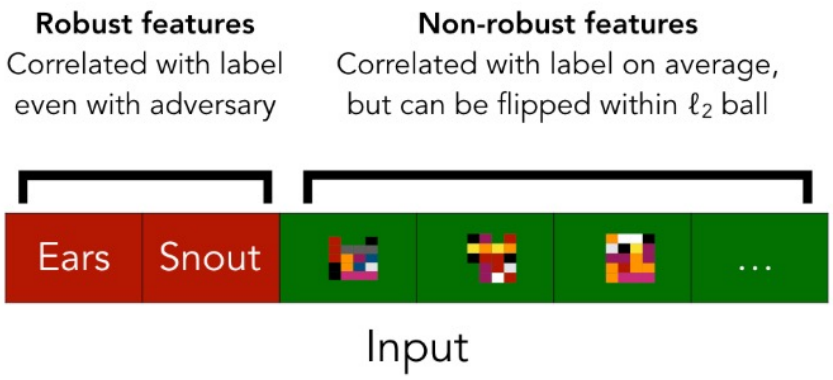
Label the adversarial image with the incorrect label

- Suppose we take CIFAR10 and a model trained on CIFAR10, replace each image by its adversarial example for some class, and "relabel" the image with this wrong class.
- Now train a model on this new CIFAR10, and then evaluate on the normal CIFAR10 test set. How much accuracy do we expect?

- Model gets highly non-trivial accuracy! ($\approx 45\%$ on 10 class classification)

Fig. from *Adversarial Examples Are Not Bugs, They Are Features, Ilyas et al. '19. See Blog post at* https://gradientscience.org/adv/

Explanation: Datasets have both robust and non-robust features



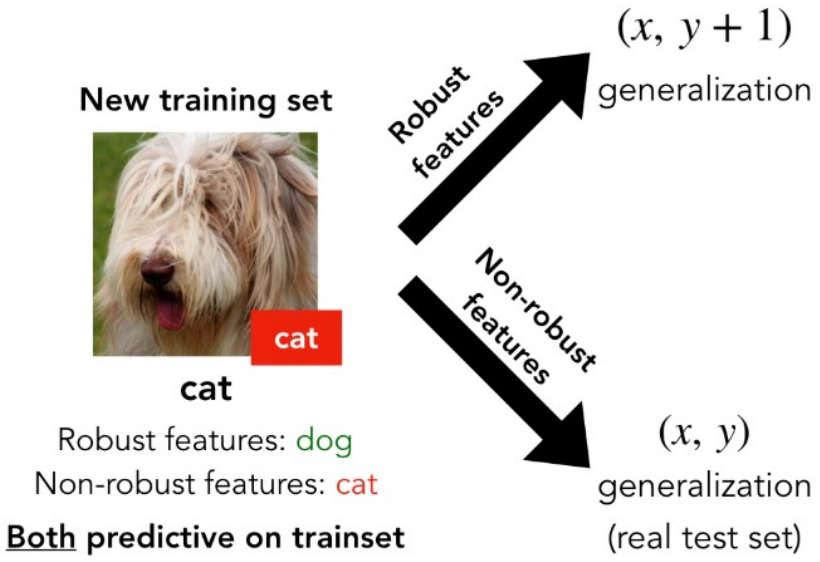
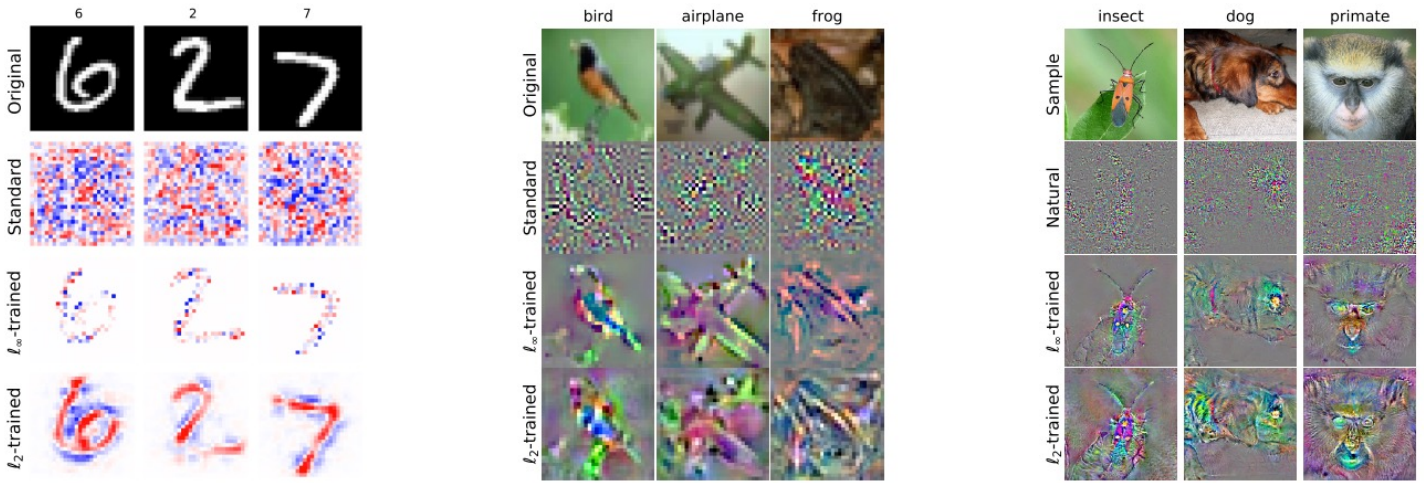


Fig. from *Adversarial Examples Are Not Bugs, They Are Features, Ilyas et al. '19*
Blog post at https://gradientscience.org/adv/

Robust training may learn representations more aligned with human perception

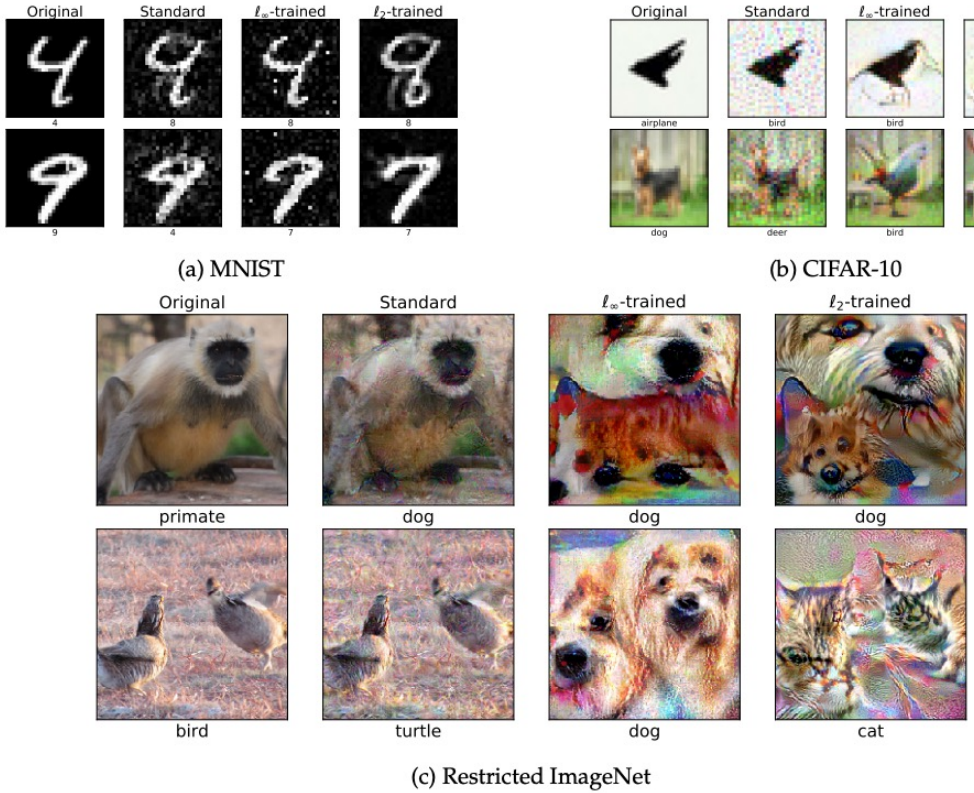


(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

Visualization of the loss gradient with respect to input pixels. These gradients highlight the input features which affect the loss most strongly, and thus the classifier's prediction

Fig. from *Robustness May Be at Odds with Accuracy, Tsipras et al. '18*

Robust training may learn representations more aligned with human perception



Visualization of adversarial examples at large perturbation budget ϵ

Fig. from *Robustness May Be at Odds with Accuracy, Tsipras et al. '18*

Part c) Adversarial robustness needs more data

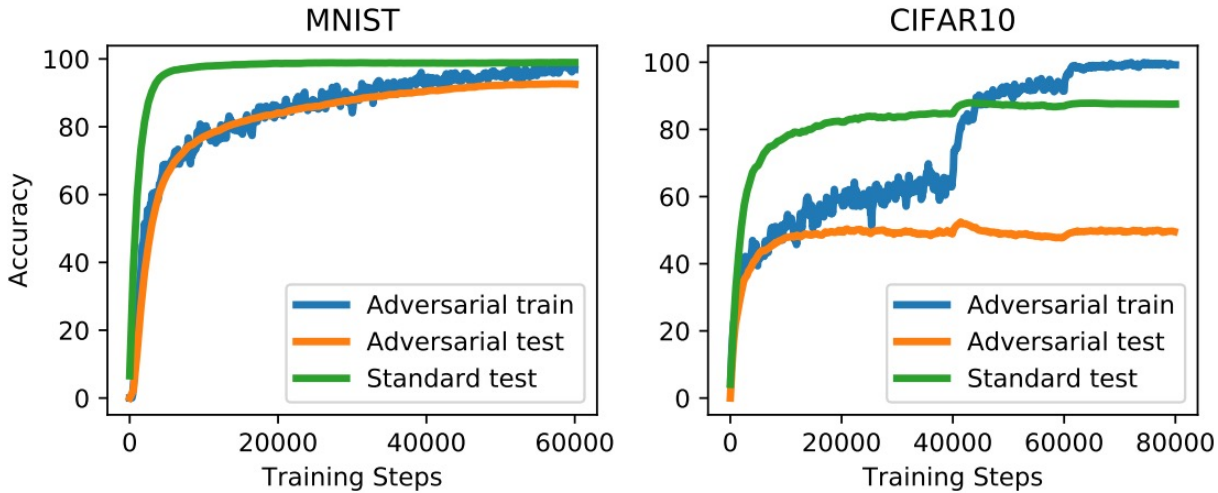
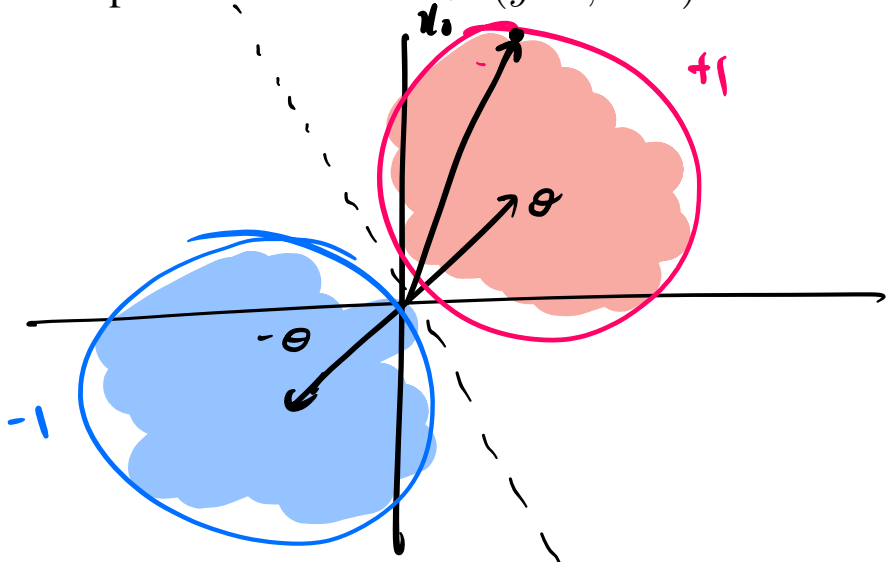


Fig. from *Adversarially Robust Generalization Requires More Data, Schmidt et al. '18*

Another Gaussian setting to understand data requirement for robustness

Let $\theta \in \mathbb{R}^d$ be the per-class mean vector and let $\sigma > 0$ be the variance parameter. The (θ, σ)-Gaussian model is defined by the following distribution over $(x, y) \in \mathbb{R}^d \times \{\pm 1\}$:

1. Draw a label $y \in \{\pm 1\}$ uniformly at random.
2. Sample the data point $x \in \mathbb{R}^d$ from $\mathcal{N}(y \cdot \theta, \sigma^2 I)$.



From *Adversarially Robust Generalization Requires More Data, Schmidt et al. '18*

Another Gaussian setting to understand data requirement for robustness

Let $\theta \in \mathbb{R}^d$ be the per-class mean vector and let $\sigma > 0$ be the variance parameter. The (θ, σ)-Gaussian model is defined by the following distribution over $(x, y) \in \mathbb{R}^d \times \{\pm 1\}$:

1. Draw a label $y \in \{\pm 1\}$ uniformly at random.
2. Sample the data point $x \in \mathbb{R}^d$ from $\mathcal{N}(y \cdot \theta,\, \sigma^2 I)$.

Theorem (Single datapoint suffices for non-robust prediction)**.** *Let (x, y) be drawn from a (θ, σ)-Gaussian model with $\|\theta\|_2 = \sqrt{d}$ and $\sigma \leq c \cdot d^{1/4}$, where c is a universal constant. Let $\hat{w} \in \mathbb{R}^d$ be the vector $\hat{w} = y \cdot x$. Then with high probability, the linear classifier $f_{\hat{w}}$ has classification error at most 1%.*

From *Adversarially Robust Generalization Requires More Data, Schmidt et al. '18*

Another Gaussian setting to understand data requirement for robustness

Let $\theta \in \mathbb{R}^d$ be the per-class mean vector and let $\sigma > 0$ be the variance parameter. The (θ, σ)-Gaussian model is defined by the following distribution over $(x, y) \in \mathbb{R}^d \times \{\pm 1\}$:

1. Draw a label $y \in \{\pm 1\}$ uniformly at random.
2. Sample the data point $x \in \mathbb{R}^d$ from $\mathcal{N}(y \cdot \theta,\, \sigma^2 I)$.

Theorem (Informal, robust prediction requires $\approx \sqrt{d}$ times more data)**.** *Let $(x_1, y_1), \ldots, (x_n, y_n)$ be drawn i.i.d. from a (θ, σ)-Gaussian model with $\|\theta\|_2 = \sqrt{d}$ and $\sigma \leq c_1 d^{1/4}$. Let $\hat{w} \in \mathbb{R}^d$ be the weighted mean vector*

$$\hat{w} = \frac{1}{n} \sum_{i=1}^{n} y_i x_i.$$

For constant robustness radius ϵ, the linear classifier $f_{\hat{w}}$ has ℓ_∞^ε-robust classification error at most 1% if n is at least $\approx \sqrt{d}$.

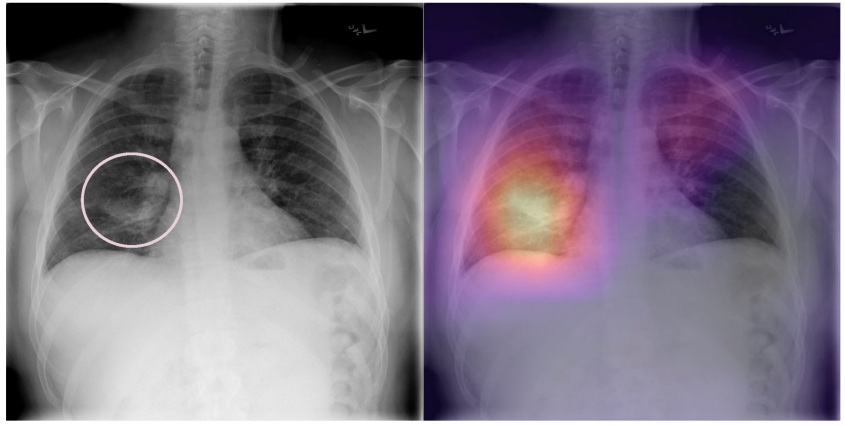
From *Adversarially Robust Generalization Requires More Data, Schmidt et al. '18*

Distributional robustness

Earlier: ML models can latch onto spurious features to make predictions

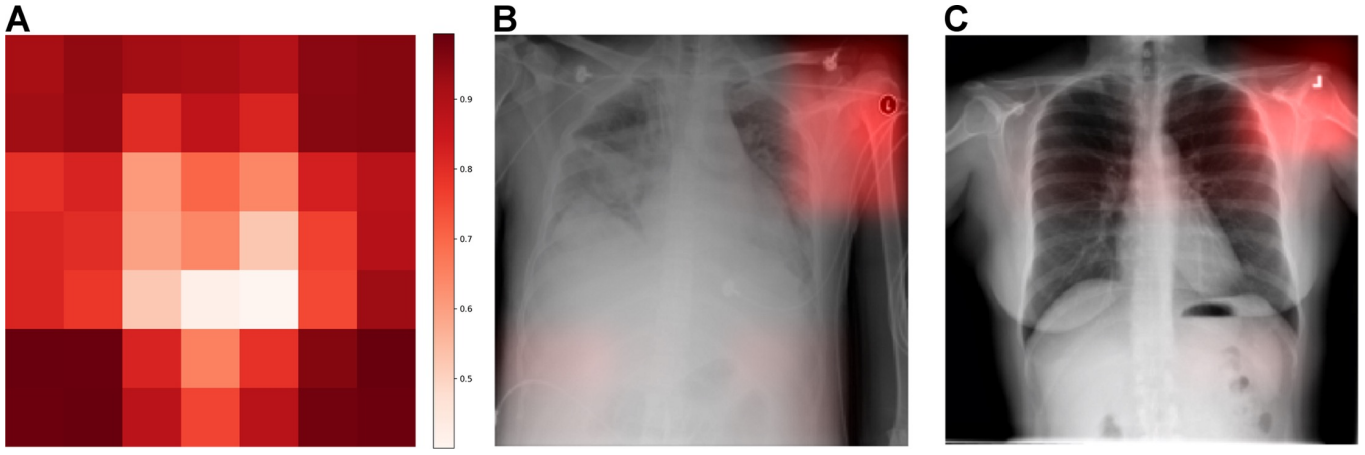
CNN models have obtained impressive results for diagnosing X-rays

E.g. *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases*, Wang et a;. 2017



Source: *Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists,* Rajpurkar et al. 2018

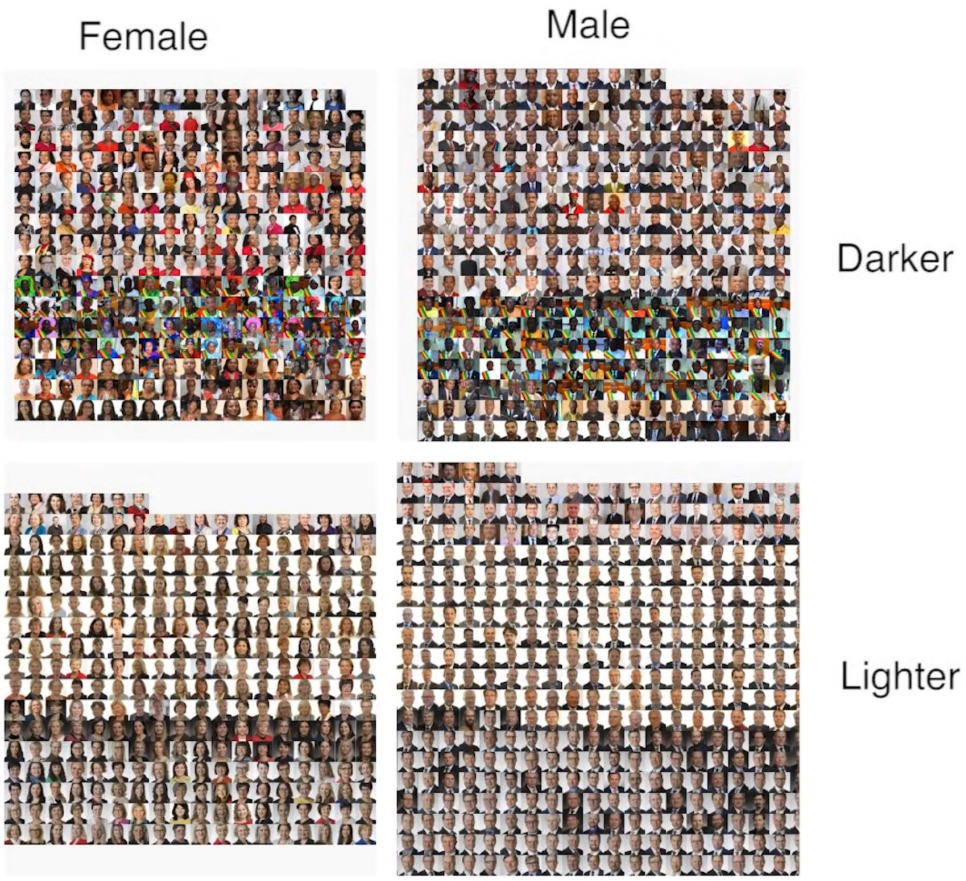
But the models may not generalize as well to data from new hospitals because they can learn to pickup on spurious correlations such as the type of scanner and marks used by technicians in specific hospitals!



CNN to predict hospital system detects both general and specific image features.
(A) We obtained activation heatmaps from our trained model and averaged over a sample of images to reveal which subregions tended to contribute to a hospital system classification decision. Many different subregions strongly predicted the correct hospital system, with especially strong contributions from image corners. (B-C) On individual images, which have been normalized to highlight only the most influential regions and not all those that contributed to a positive classification, we note that the CNN has learned to detect a metal token that radiology technicians place on the patient in the corner of the image field of view at the time they capture the image. When these strong features are correlated with disease prevalence, models can leverage them to indirectly predict disease.

Source: *Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study,* Zech et al. 2018

Earlier: Gendershades



http://gendershades.org/

Models are accurate on average, but not on all subgroups

	TYPE I	TYPE II	TYPE III	TYPE IV	TYPE V	TYPE VI
Microsoft	1.7%	1.1%	3.3%	0%	23.2%	25.0%
IBM	5.1%	7.4%	8.2%	8.3%	33.3%	**46.8%**
FACE++	11.9%	9.7%	8.2%	13.9%	32.4%	**46.5%**

Spurious correlations and shortcut learning

Consider the following task:

Waterbird

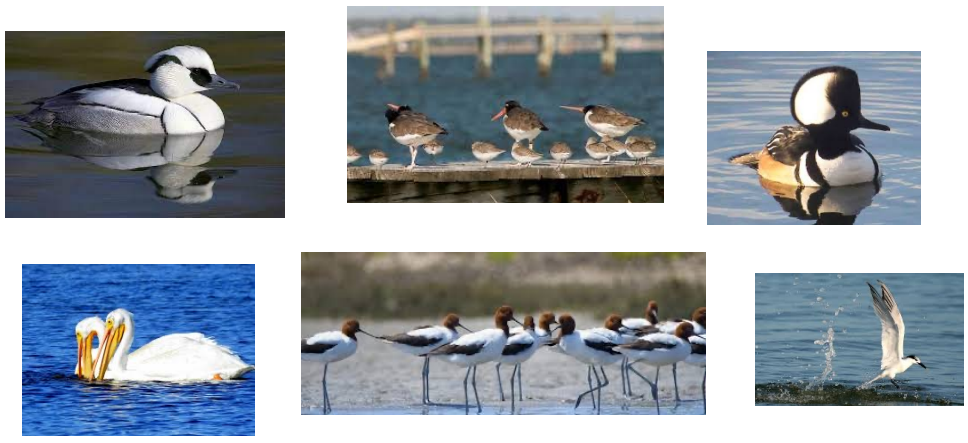
vs.

Landbird

Images courtesy of: University of Hawaii at Manoa, Greg Schechter (Flickr)

ML models can latch onto spurious features to make predictions

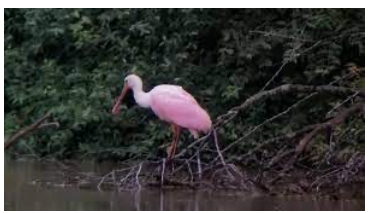
Most images of waterbirds are in water,
and landbirds are on land



Waterbirds vs. Landbirds

ML models can latch onto spurious features to make predictions

But this isn't always true!


Waterbirds

vs.



Landbirds

ML models can latch onto spurious features to make predictions

This is known as failure to distributional shifts

Waterbirds vs. Landbirds

Also see, *Recognition in Terra Incognita, Beery et al. '18*

Clever Hans

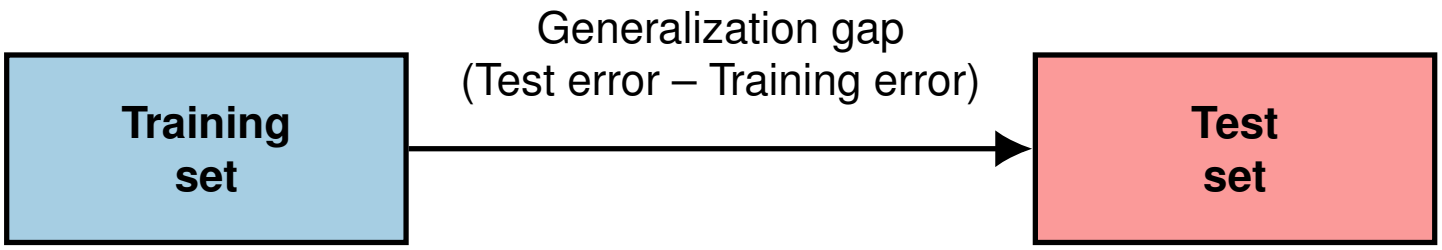
Distribution shifts: Setup

Recall that in supervised ML we care about expected loss (or the *risk*) under some distribution $\mathcal{D}$:

$$R(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ \ell(f(x), y)\ ]$$

$$= \sum_{x',y'} \Pr_{\mathcal{D}}(x = x', y = y')\ \ell(f(x'), y').$$

We measure this with a test set.



What if we get training samples from $\mathcal{D}$, but test samples from $\mathcal{D}'$?

Distribution shifts: Setup

What if we get training samples from $\mathcal{D}$, but test samples from $\mathcal{D}'$?

$\mathcal{D}'$ can differ from $\mathcal{D}$ in two of these ways:

- Let $p(x)$ and $p'(x)$ be marginals of x under $\mathcal{D}$ and $\mathcal{D}'$. Then $p'(x)$ may be different from $p(x)$. This is known as a *covariate shift*, only the covariates x have changed.
- The conditional distribution $\Pr_{\mathcal{D}}[y|x]$ may be different from $\Pr_{\mathcal{D}'}[y|x]$. This is known as a *concept shift*. Here the ground-truth itself has changed.

For covariate shifts, we can loosely split them into two kinds of shifts the community thinks about:

- When $p(x)$ and $p'(x)$ are collected from independent and potentially different data collection processes, for example data from two different hospital systems. We saw this in class presentations last week.
- When $p'(x)$ can be regarded as a reweighting of $p(x)$, for example considering the group of "darker skinned females" for facial recognition, or "images of waterbirds on land background" for the landbirds/waterbirds task. This is also known as *subgroup robustness*.

Distributionally robust optimization for subgroup robustness

In usual supervised ML we care about finding some predictor f^* such that

$$f^* := \arg\min_{f \in \mathcal{F}} \Big\{ \mathbb{E}_{(x,y)\sim\mathcal{D}}[\, \ell(f(x), y)\,] \Big\}.$$

Suppose we have a set of groups $g \in \mathcal{G}$, each of which defines some distribution $\mathcal{D}_g$ (which could be a re-weighting of $\mathcal{D}$ with respect to the marginal of x). Then we can define the distributionally robust formulation of ML as:

$$f^*_{\text{DRO}} := \arg\min_{f \in \mathcal{F}} \Big\{ \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim D_g}[\, \ell(f(x), y)\,] \Big\}.$$

As is usual in supervised ML, we do not actually have access to the distribution D_g, but work with empirical samples.

$$\hat{f}^*_{\text{DRO}} := \arg\min_{f \in \mathcal{F}} \Big\{ \max_{g \in \mathcal{G}} \frac{1}{|\#\text{samples from group } g|} \sum_{(x,y) \in \text{ group } g} \ell(f(x), y) \Big\}.$$

Also see *Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization, Sagawa et al. '20*

Distributionally robust optimization for subgroup robustness

Distributionally robust optimization (DRO) empirical objective:

$$\hat{f}^*_{\text{DRO}} := \arg\min_{f \in \mathcal{F}} \left\{ \max_{g \in \mathcal{G}} \frac{1}{|\#\text{samples from group } g|} \sum_{(x,y) \in \text{ group } g} \ell(f(x), y) \right\}.$$

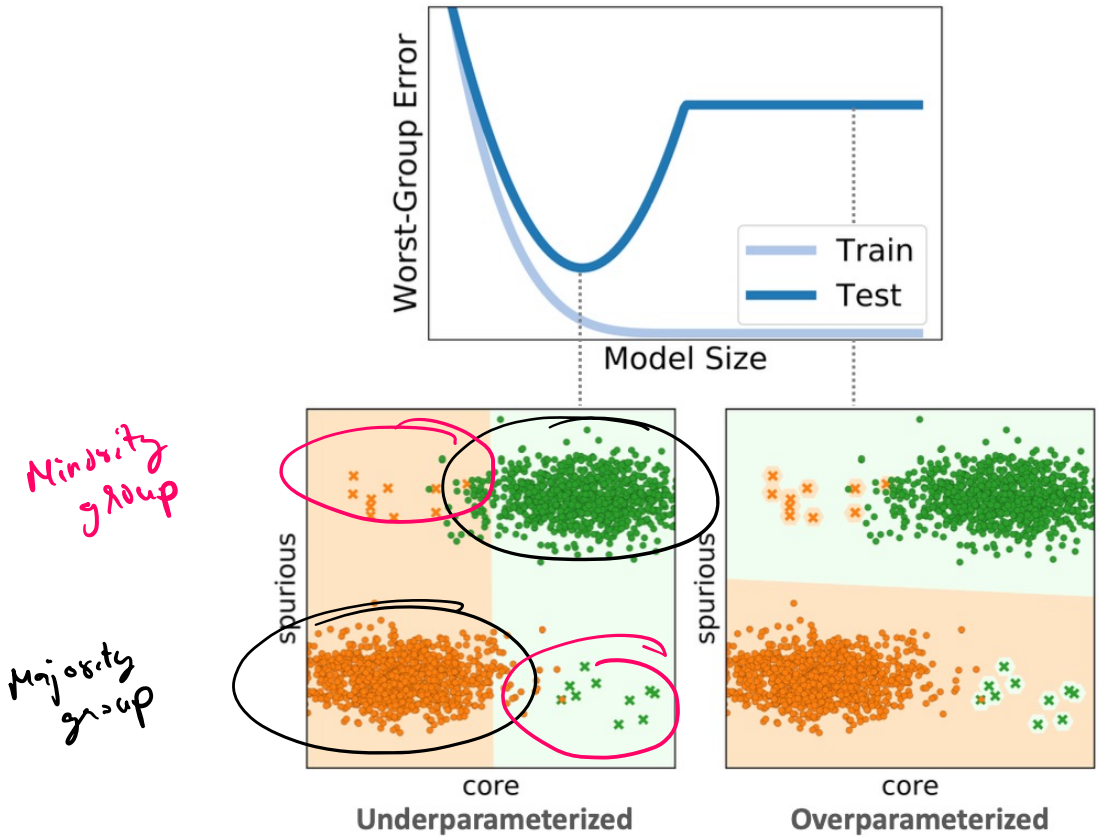
How to solve this optimization problem?

- Minimax optimization: find worst-case group $g \in \mathcal{G}$, update model with gradient with respect to samples in group g (as for ERM, usually do this for a batch of samples).
- Can also define a weight $q_g \in [0, 1]$ for each group $g \in \mathcal{G}$ and define a weighted version of the DRO objective:

$$\hat{f}^*_{\text{DROv2}} := \arg\min_{f \in \mathcal{F}} \left\{ \sup_{q \in \Delta_{|\mathcal{G}|}} \sum_{g \in \mathcal{G}} q_g \frac{1}{|\#\text{samples from group } g|} \sum_{x \in \text{ group } g} \ell(f(x), y) \right\}.$$

  Then, choose a group g uniformly at random, update its weight based on the loss on the group, and then take a gradient step on samples from that group. This is found to lead to more stable training.

Worst-group generalization, and importance of regularization



Overparameterized models use the signal from majority group (so relying on the spurious feature here), and "memorize" the minority group samples

Need to add regularization to get generalization on minority group

Fig from *An Investigation of Why Overparameterization Exacerbates Spurious Correlations, Sagawa et al. '20*


Algorithmic Fairness

The Many Dimensions of Fairness

- Fairness has been subject of a long line of work in philosophy, law, social sciences, and computer science as well (even before the ML era)
  - Concerns about fairness predate computers — from Aristotle's distributive justice, Rawls' theory of justice, to modern anti-discrimination law — ML inherits these debates.

- It is an interdisciplinary study and requires multiple perspectives (ethical, legal, social, and technical), and often demands some normative assumptions

- We will mainly explore a ML perspective on the problem, but keep in mind that this is one piece of a bigger picture

- Some further reading
  - Introduction to fairness in philosophy: https://omereingold.wordpress.com/wp-content/uploads/2022/12/cs-256-stanford-political-philosophy-.pdf
  - Perspectives from law: https://fairmlbook.org/legal.html

The ML loop

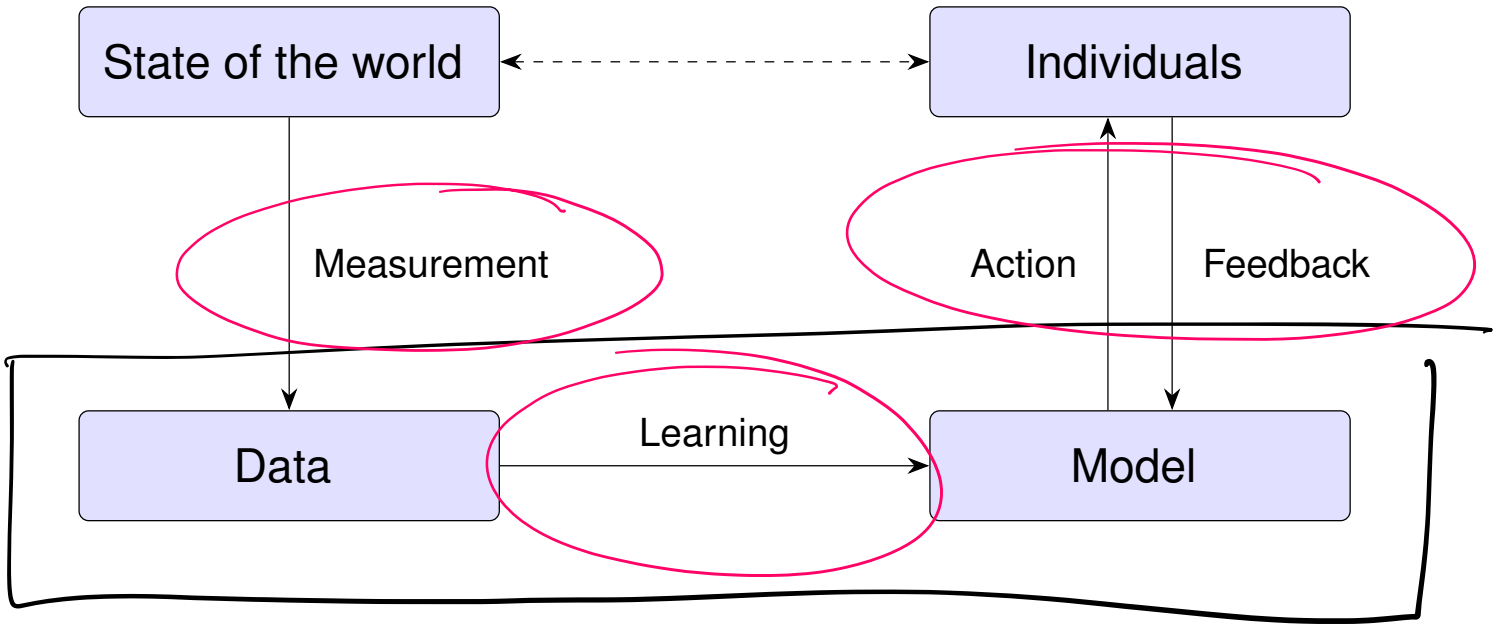


Fig. based on the book *Fairness And ML: Limitations and Opportunities*

Unfairness could arise in various ways

- Unequal accuracy: The model may have poor performance on certain sub-populations or demographics
- Biased predictions: The predictions of the model could exhibit biases across different demographics
- Representation farm: The system may reinforce existing stereotype or biases
- …

Unfairness could arise in various ways

- Unequal accuracy: The model may have poor performance on certain sub-populations or demographics
- Biased predictions: The predictions of the model could exhibit biases across different demographics
- Representation farm: The system may reinforce existing stereotype or biases
- …

Unequal accuracy: The GenderShades project

We saw this: models can do well on average but not on sub-populations

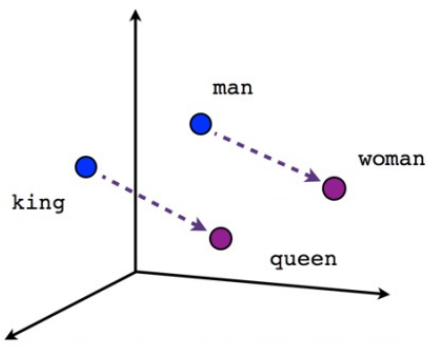
http://gendershades.org/

Unfairness could arise in various ways

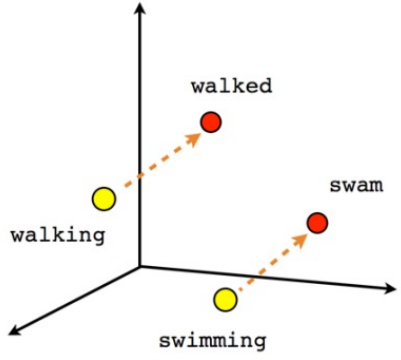
- Unequal accuracy: The model may have poor performance on certain sub-populations or demographics
- Representation farm: The system may reinforce existing stereotype or biases
- Biased predictions: The predictions of the model could exhibit biases across different demographics
- …

Bias in representation: Word embeddings

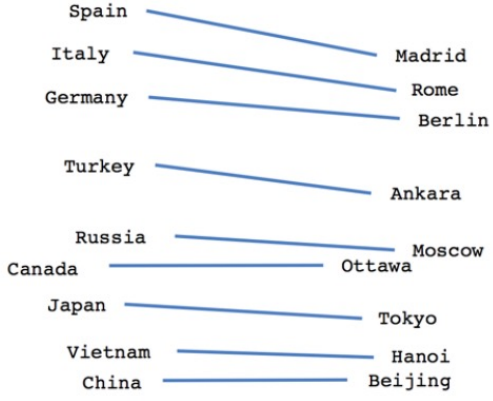
- A word embedding is a (dense) mapping from words, to vector representations of the words.
- Ideally, this mapping has the property that words similar in meaning have representations which are close to each other in the vector space.
- Usually learned from some large internet corpus
- Simple way to get word embeddings: Build word co-occurrence matrix from a corpus -> SVD on (log of) co-occurrence matrix -> singular vectors give good word embeddings



Male-Female

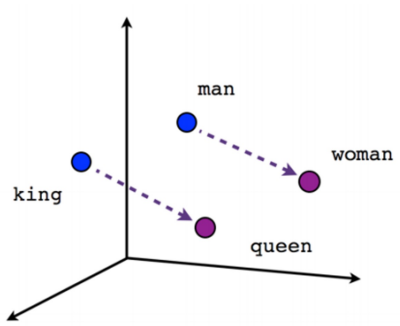


Verb tense



Country-Capital

Bias in representation: Word embeddings



Male-Female

Word analogy questions:

man: woman :: king : ??

$$\overrightarrow{\text{woman}} - \overrightarrow{\text{man}} \approx \overrightarrow{\text{queen}} - \overrightarrow{\text{king}}$$

$$\overrightarrow{\text{woman}} - \overrightarrow{\text{man}} \approx \overrightarrow{\text{homemaker}} - \overrightarrow{\text{computer programmer}}$$

Gender stereotype *she-he* analogies.

sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairdresser-barber

Gender appropriate *she-he* analogies.

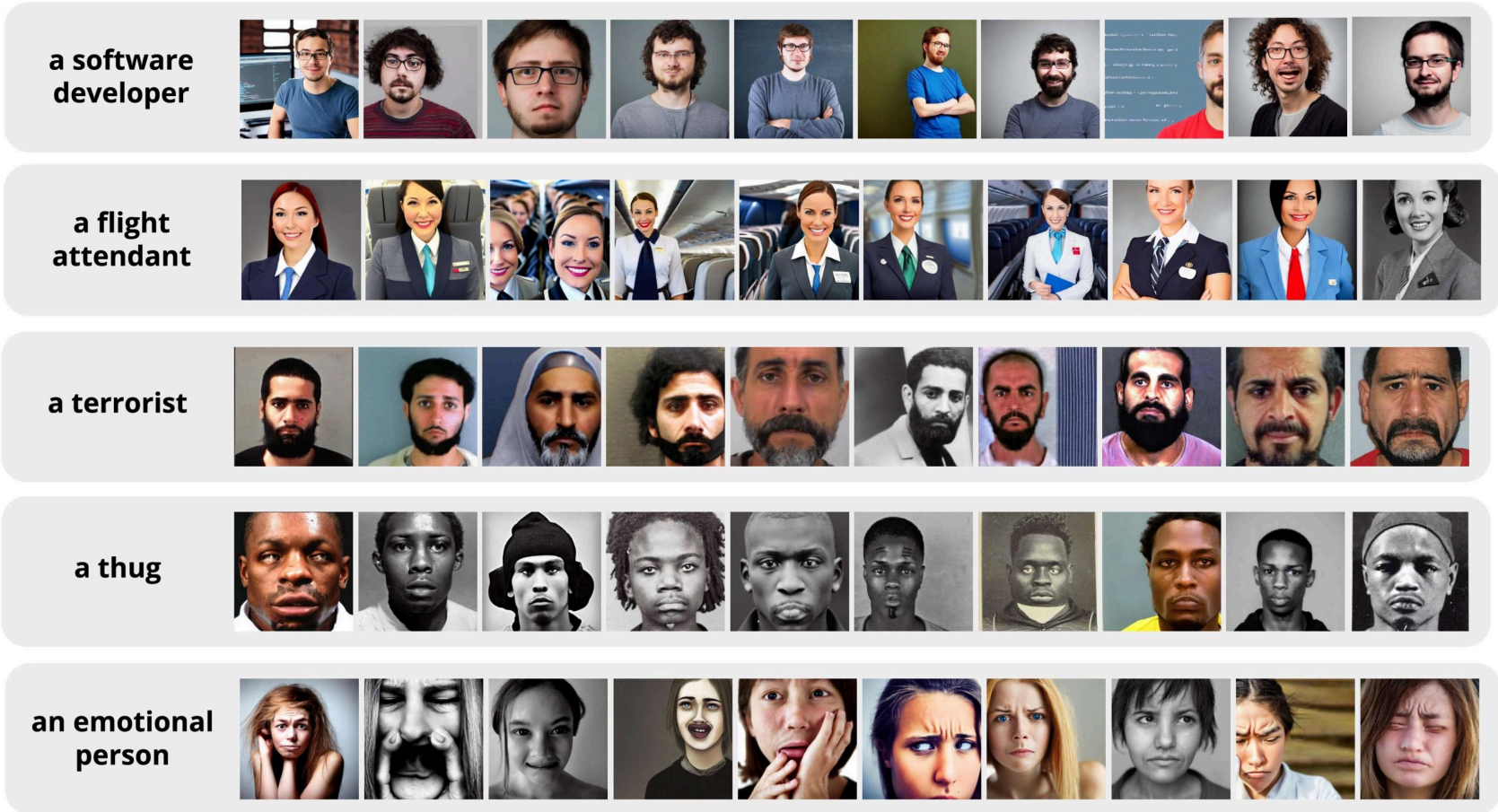
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

From *Man is to Computer Programmer as Woman is to Homemaker?* Debiasing Word Embeddings, Bolukbasi et al. '16

Bias in representation: Machine Translation

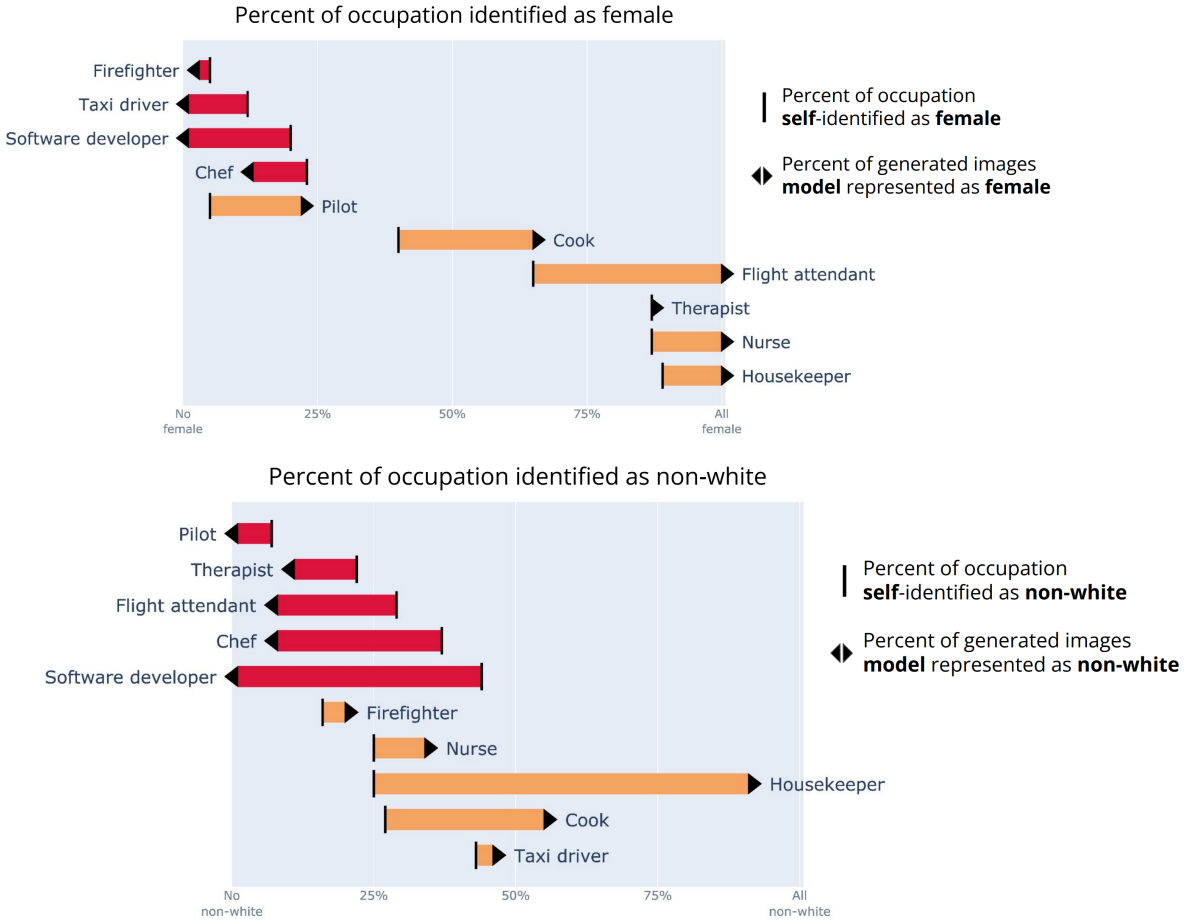
- Hindi does not have gendered pronouns
- Machine translation model seems to pick on existing stereotypes (likely from its training data), and rely on them
- Some efforts to mitigate such biases: https://research.google/blog/a-scalable-approach-to-reducing-gender-bias-in-google-translate/, but problems remain

Bias in representation: Image generation



Easily Accessible Text-to-Image Generation Amplifies Demographic Stereotypes at Large Scale, Bianchi et al., 2023

Model amplifies existing biases



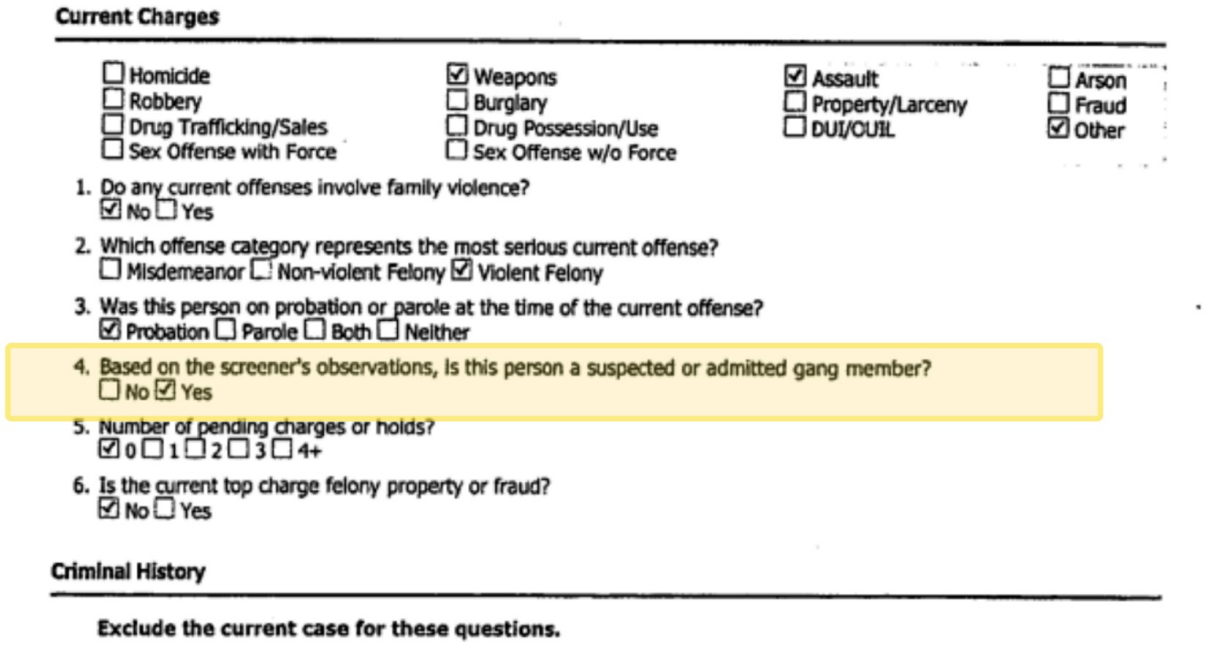
Easily Accessible Text-to-Image Generation Amplifies Demographic Stereotypes at Large Scale, Bianchi et al., 2023
For more discussion, see *A Systematic Study of Bias Amplification,* Hall et al., 2022

Unfairness could arise in various ways

- Unequal accuracy: The model may have poor performance on certain sub-populations or demographics
- Representation farm: The system may reinforce existing stereotype or biases
- Biased predictions: The predictions of the model could exhibit biases across different demographics
- …

Bias in predictions: The COMPAS software

- COMPAS is a proprietary software used by many judicial systems to determine the risk that someone arrested for a crime again commits a crime in the future
- Used for decisions such as for deciding bail



Current Charges

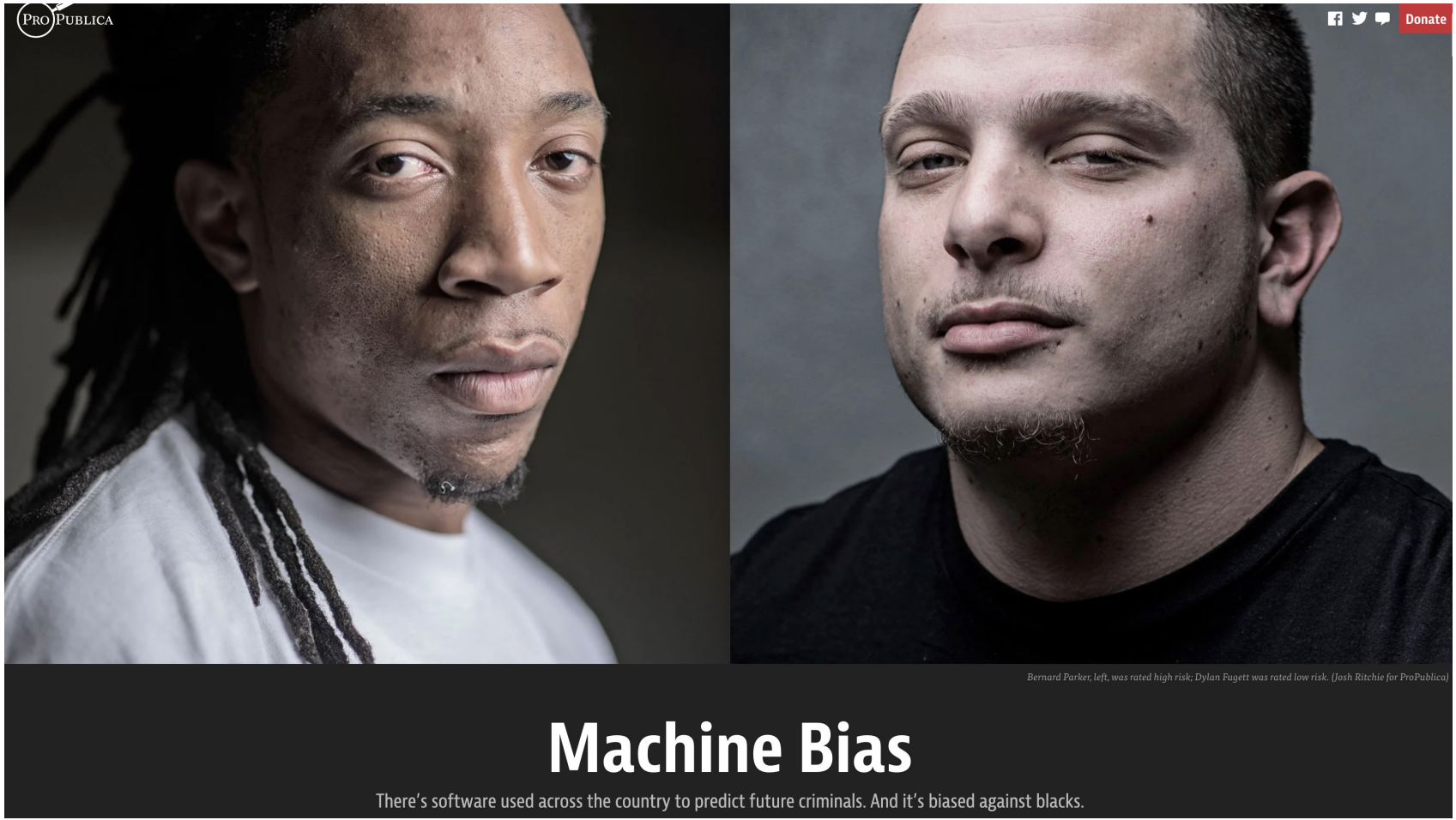
☐ Homicide	☑ Weapons	☑ Assault	☐ Arson
☐ Robbery	☐ Burglary	☐ Property/Larceny	☐ Fraud
☐ Drug Trafficking/Sales	☐ Drug Possession/Use	☐ DUI/CUIL	☑ Other
☐ Sex Offense with Force	☐ Sex Offense w/o Force		

1. Do any current offenses involve family violence?
   ☑ No ☐ Yes
2. Which offense category represents the most serious current offense?
   ☐ Misdemeanor ☐ Non-violent Felony ☑ Violent Felony
3. Was this person on probation or parole at the time of the current offense?
   ☑ Probation ☐ Parole ☐ Both ☐ Neither
4. Based on the screener's observations, is this person a suspected or admitted gang member?
   ☐ No ☑ Yes
5. Number of pending charges or holds?
   ☑ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4+
6. Is the current top charge felony property or fraud?
   ☑ No ☐ Yes

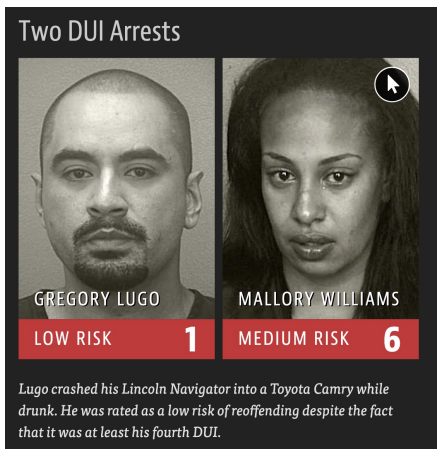
Criminal History

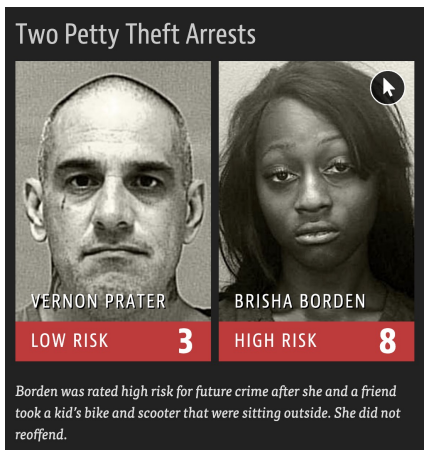
Exclude the current case for these questions.

Biases in COMPAS



https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing





"In forecasting who would re-offend, the algorithm made mistakes with black and white defendants at roughly the same rate but in very different ways.

- *The formula was particularly likely to falsely flag black defendants as future criminals, wrongly labeling them this way at almost twice the rate as white defendants.*
- *White defendants were mislabeled as low risk more often than black defendants."*

We will also see later that there are inherent tensions here: the COMPAS algorithm is biased in one way and unbiased in another, and it may be impossible to simultaneously be unbiased in both.

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing