Google Research

Pitfalls of Deep Learning

Praneeth Netrapalli (Google Research India)

Joint work with Harshay Shah (Google Research India), Kaustav Tamuly, Aditi Raghunathan (Stanford), Prateek Jain (Google Research India)

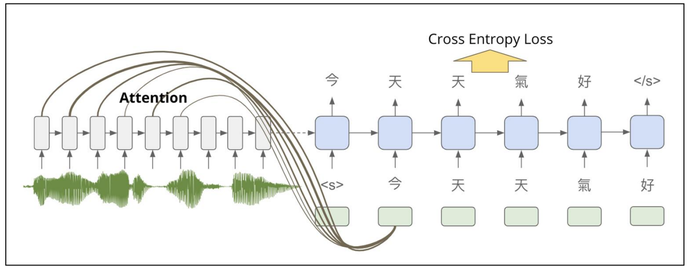


Motivation

Benchmark datasets and task performance



Image classification, ImageNet dataset

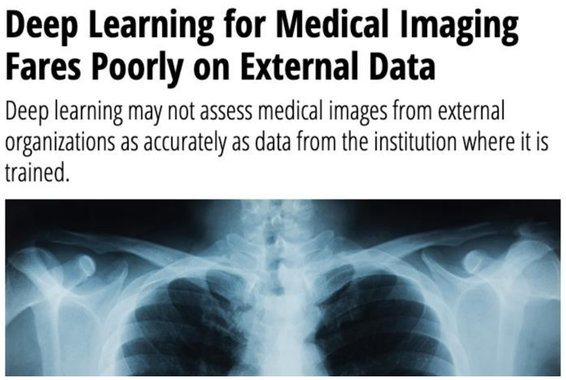


Speech Recognition, Switchboard-1 data

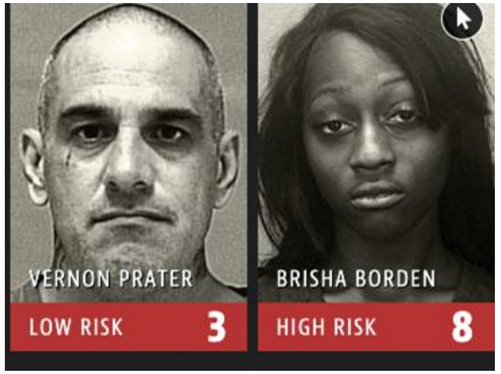


Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此外將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will 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

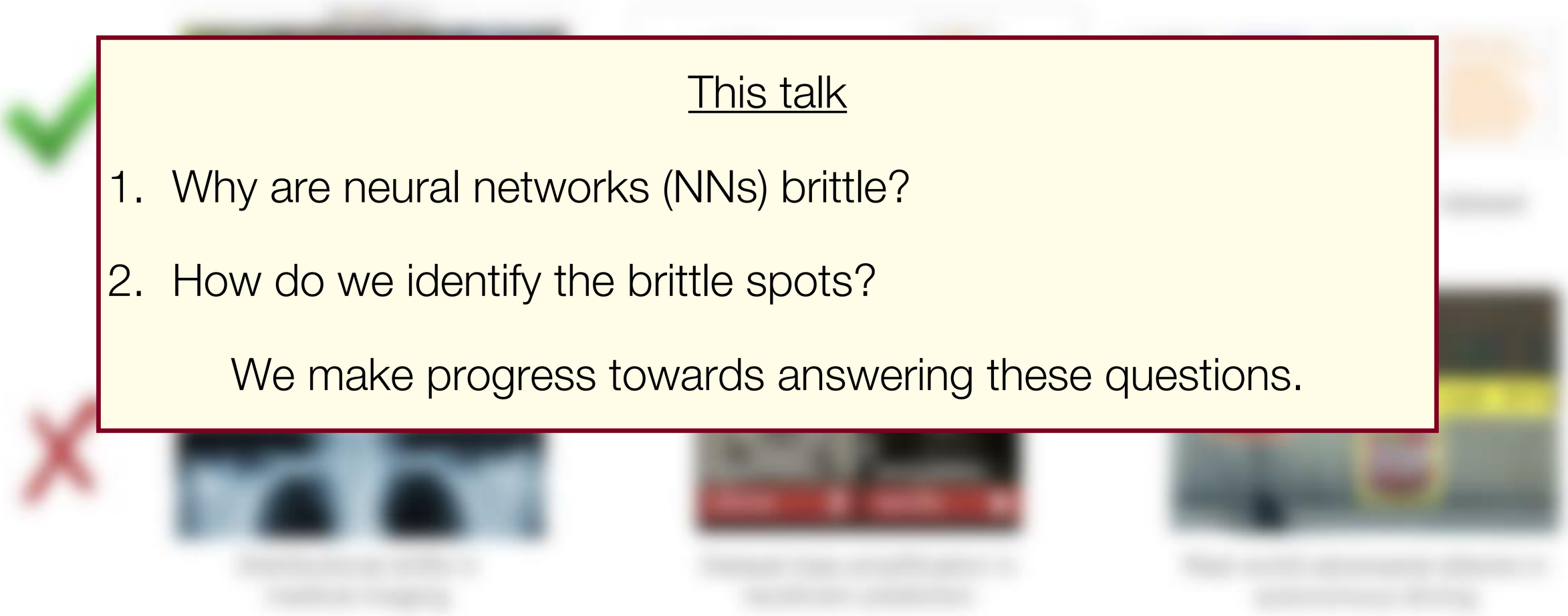


Real-world adversarial attacks in autonomous driving

Google Research

Motivation

Benchmark datasets and task performance



This talk

1. Why are neural networks (NNs) brittle?
2. How do we identify the brittle spots?

We make progress towards answering these questions.

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

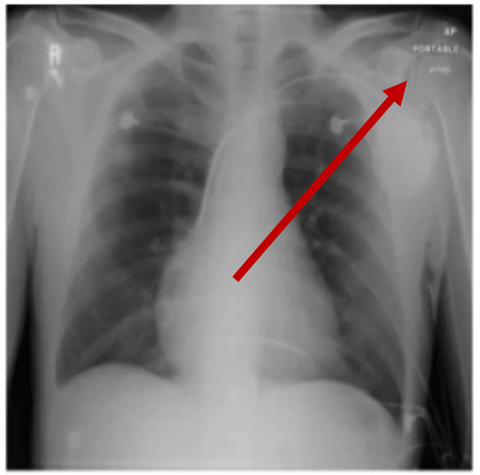


(c) Texture-shape cue conflict

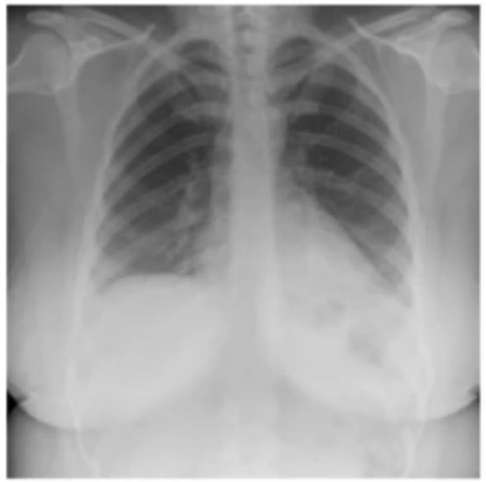
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Shortcut learning DeGrave et al. (2021)

COVID-19–



COVID-19+



Why do NNs learn only some features?

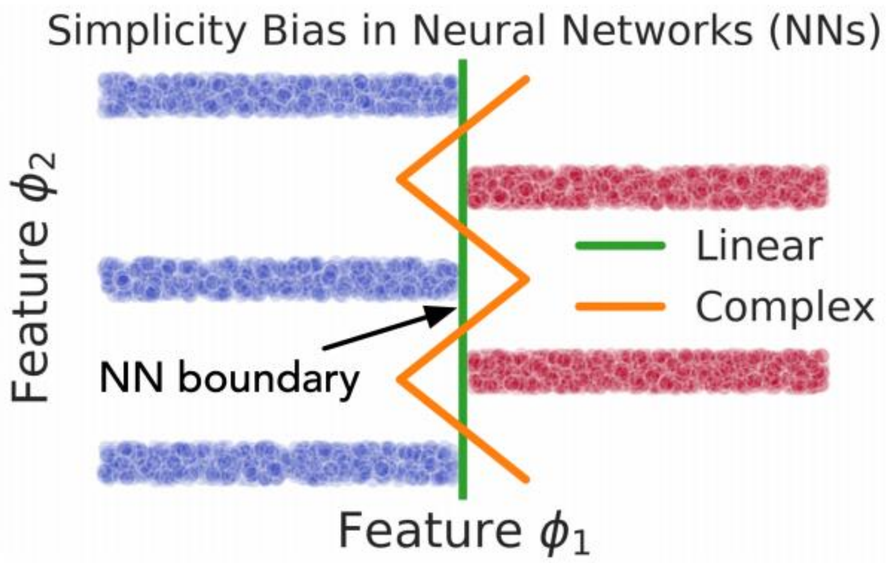
Which features do NNs learn?

Google Research

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.

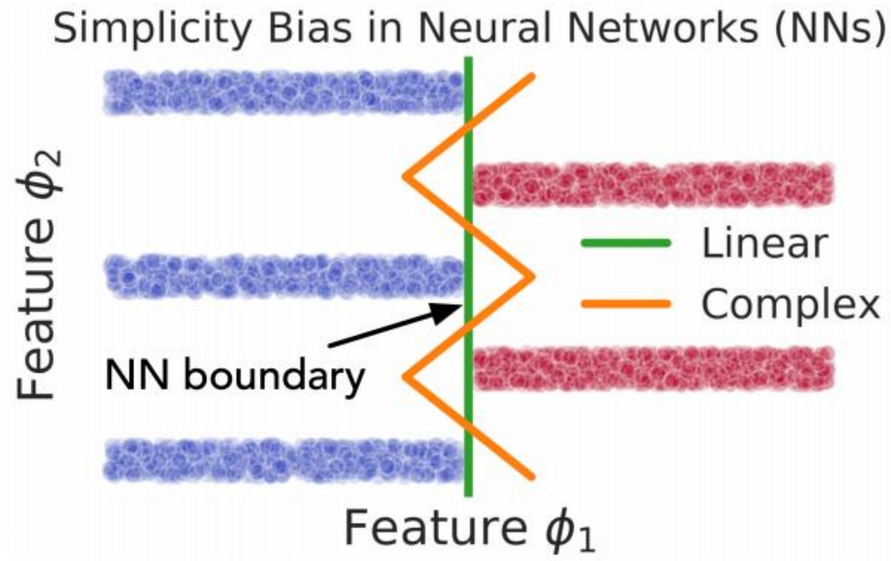


Google Research

NNs Provably Exhibit Simplicity Bias

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

- 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



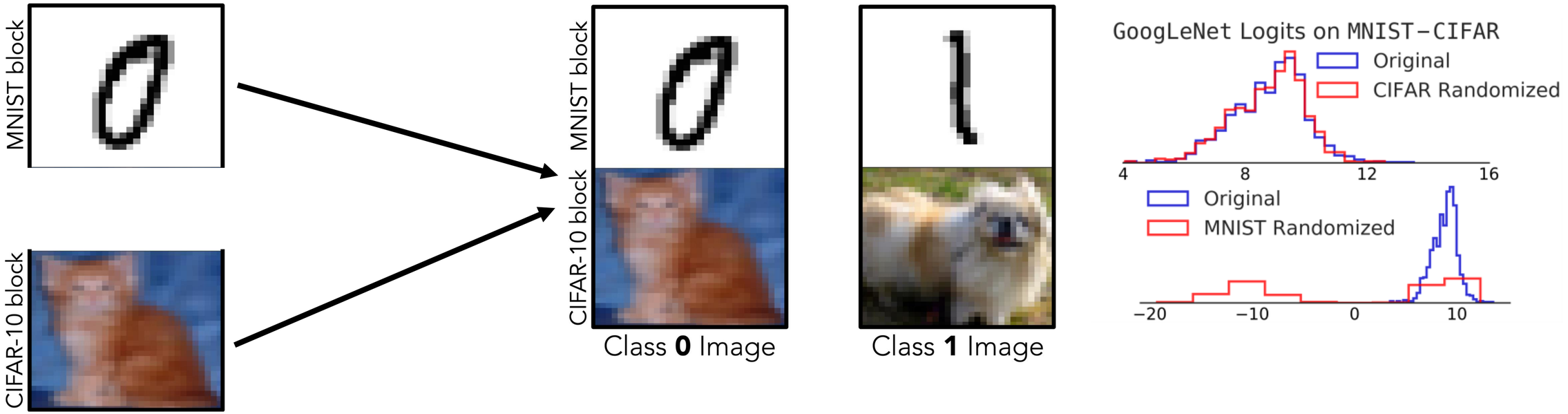
Weight of "linear feature": $O(\frac{1}{\sqrt{k}})$

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

Google Research

Towards Real Datasets

MNIST-CIFAR dataset and randomization tests

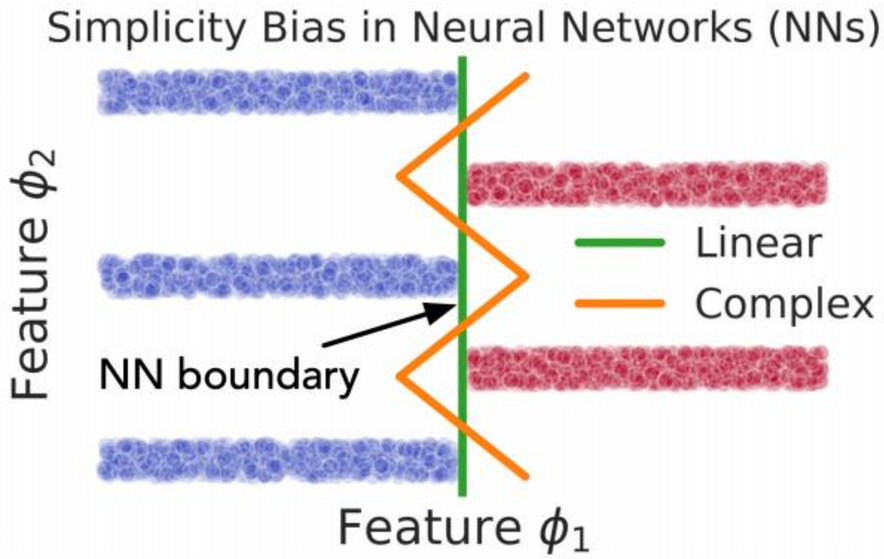


- **CIFAR randomized**: Randomize the CIFAR part of the image
- Logits do not change ⇒ Prediction **does not depend at all** on CIFAR part

Google Research

Consequences of SB

SB leads to brittleness to distribution shifts and adversarial examples

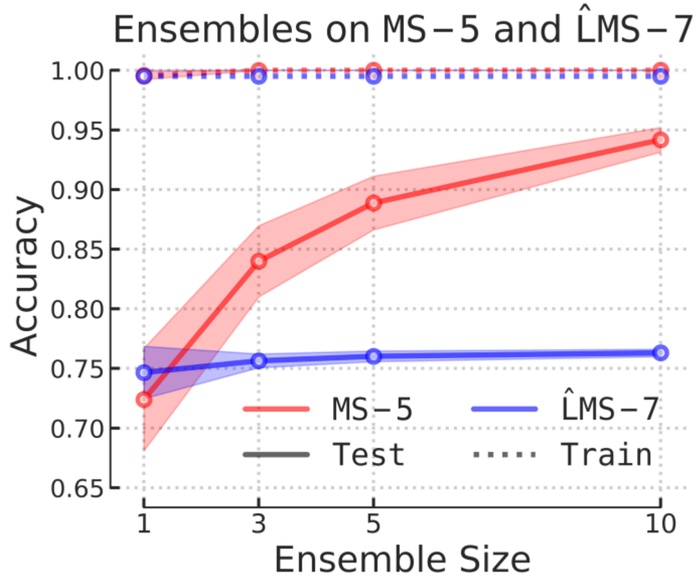


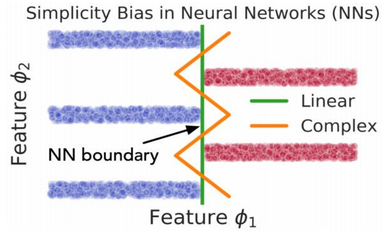
- Model learns **only** the simplest features ⇒ poor adversarial robustness
- (Simple features ≠ true features) ⇒ Poor out of distribution performance
- Can sometimes lead to poor **in-domain** performance

Google Research

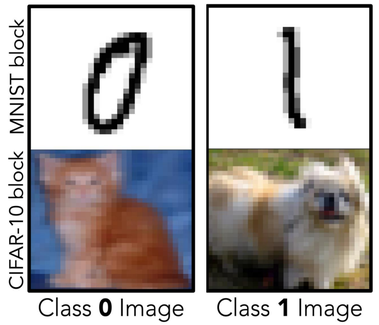
Fixing SB?

- 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

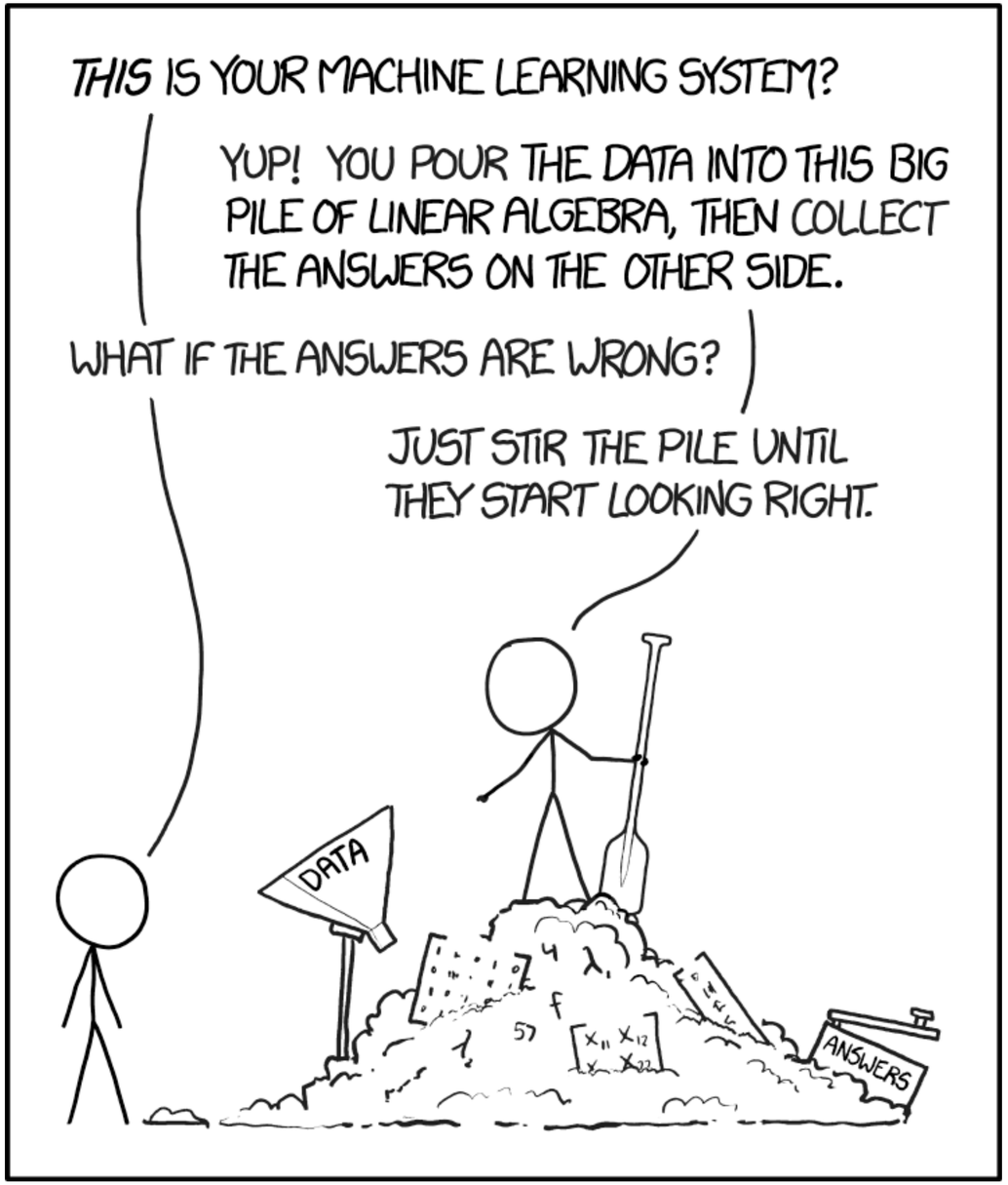
Google Research

How to Identify and Fix SB?

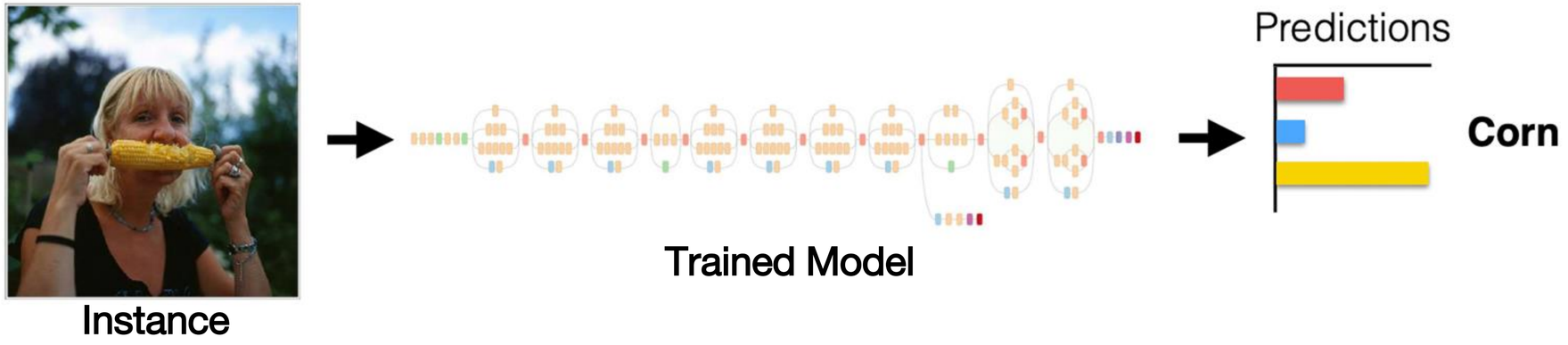
Need to understand model predictions

- What features is the model using to make its predictions?
- Key challenge: Blackbox models
- Popular approach --- Feature attribution methods to highlight:

  instance-specific predictive features



Feature Attributions

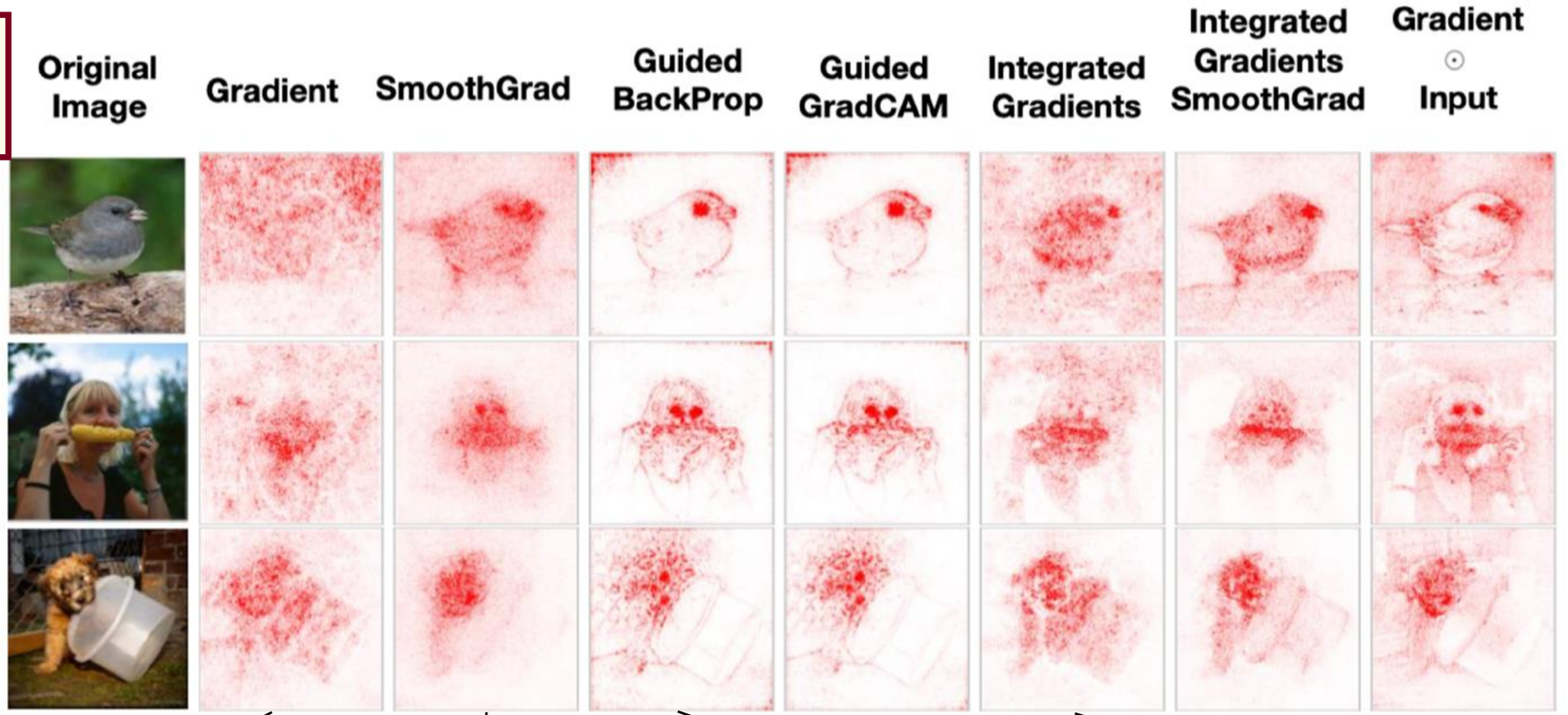


- Feature attribution methods try to rank input coordinates according to their importance in prediction
- Top most coordinates are considered most important for prediction

Images taken from Adebayo et al. (2018)

Gradient-based Feature Attributions

Figure 1 in Adebayo et al. (2018)



$\nabla_x f(x)$
[Baehrens et al., 2010]

$\mathrm{E}[\nabla_x f(x + \mathcal{N}(0, \sigma^2))]$
[Smilkov et al., 2017]

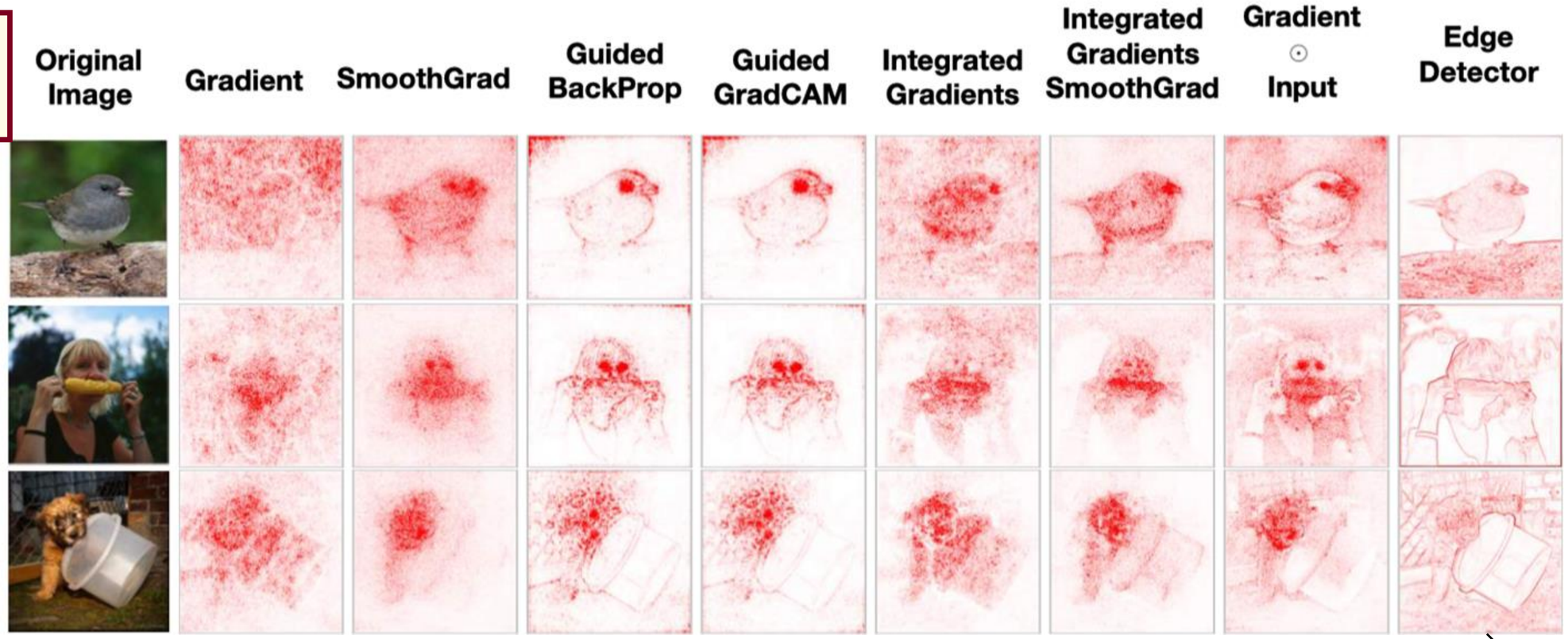
Zero out negative gradients values & activations during backprop.
[Springenberg et al., 2015]

$(x - \bar{x}) \int_0^1 \nabla_x f(\alpha x + (1 - \alpha)\bar{x}) d\alpha$
[Sundararajan et al., 2017]

Google Research

Gradient-based Feature Attributions

Figure 1 in Adebayo et al. (2018)



Model-agnostic baseline highlights salient objects in images too.

[Adebayo et al., 2018]

Which gradient-based attribution method should I use?

Google Research

When and why do input gradients (not) highlight discriminative features?

Our Approach

- Need to use a thorough framework going beyond visual inspection.
- Verify a common assumption [A] underlying vanilla gradient magnitude attributions

> [A] Input coordinates with larger gradient (in magnitude) are more relevant for model prediction than coordinates with smaller gradient.

- Is [A] satisfied by trained NNs in practice?
- **Key Challenge**: Lack of ground truth information on features used by the model.
- **Our Approach**: Multipronged effort using real and designed datasets.

Our Results

Shah, Jain, Netrapalli (2021)

[A] Input coordinates with larger gradient magnitude are more relevant for model prediction than coordinates with smaller gradient magnitude.

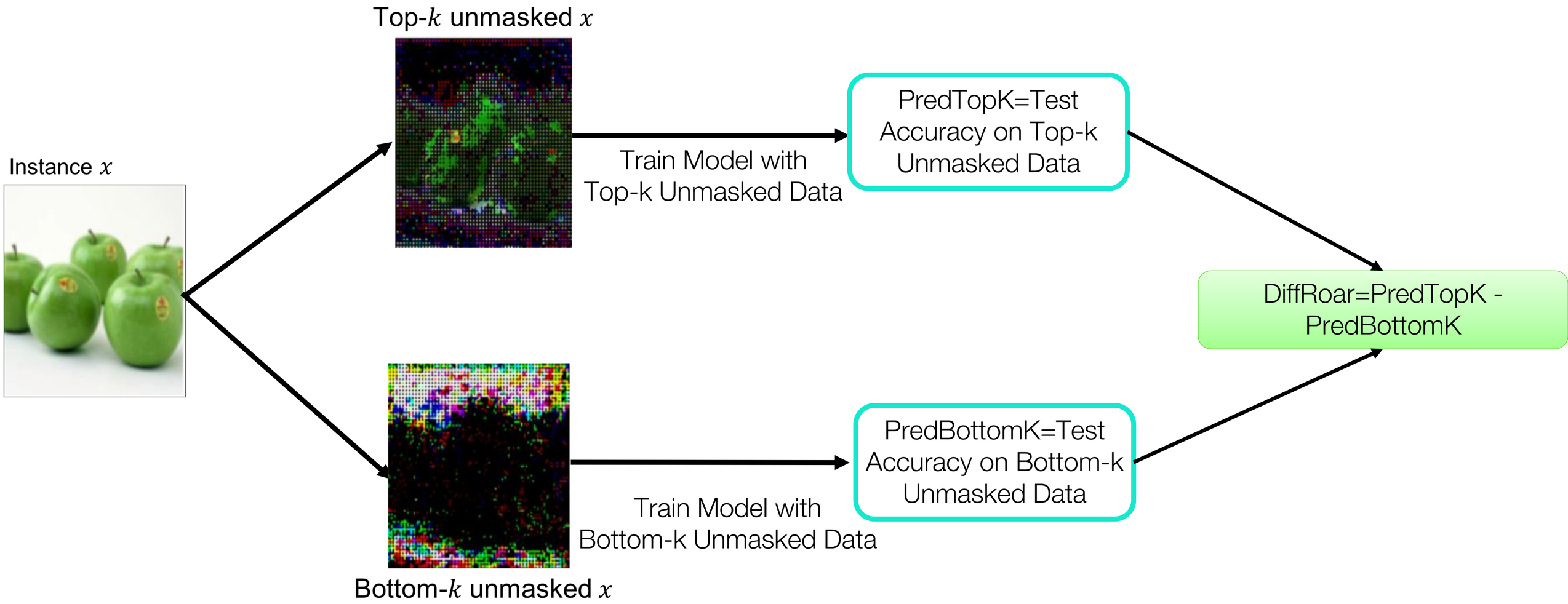
1. **Do trained models satisfy [A]?**:
   - DiffRoar framework to verify [A] building upon Hooker et al. (2019)
   - Provide strong evidence that standard models don't satisfy [A].
   - But, adversarially trained models do satisfy [A] !!!
2. **Why do standard models not satisfy [A]?**

   Using designed datasets, identify feature leakage as the key culprit!
3. **Rigorous justification**: Theoretically prove feature leakage on a simple dataset.

Google Research

DiffRoar Framework

[A] Input coordinates with larger gradient magnitude are more relevant for model prediction than coordinates with smaller gradient magnitude.

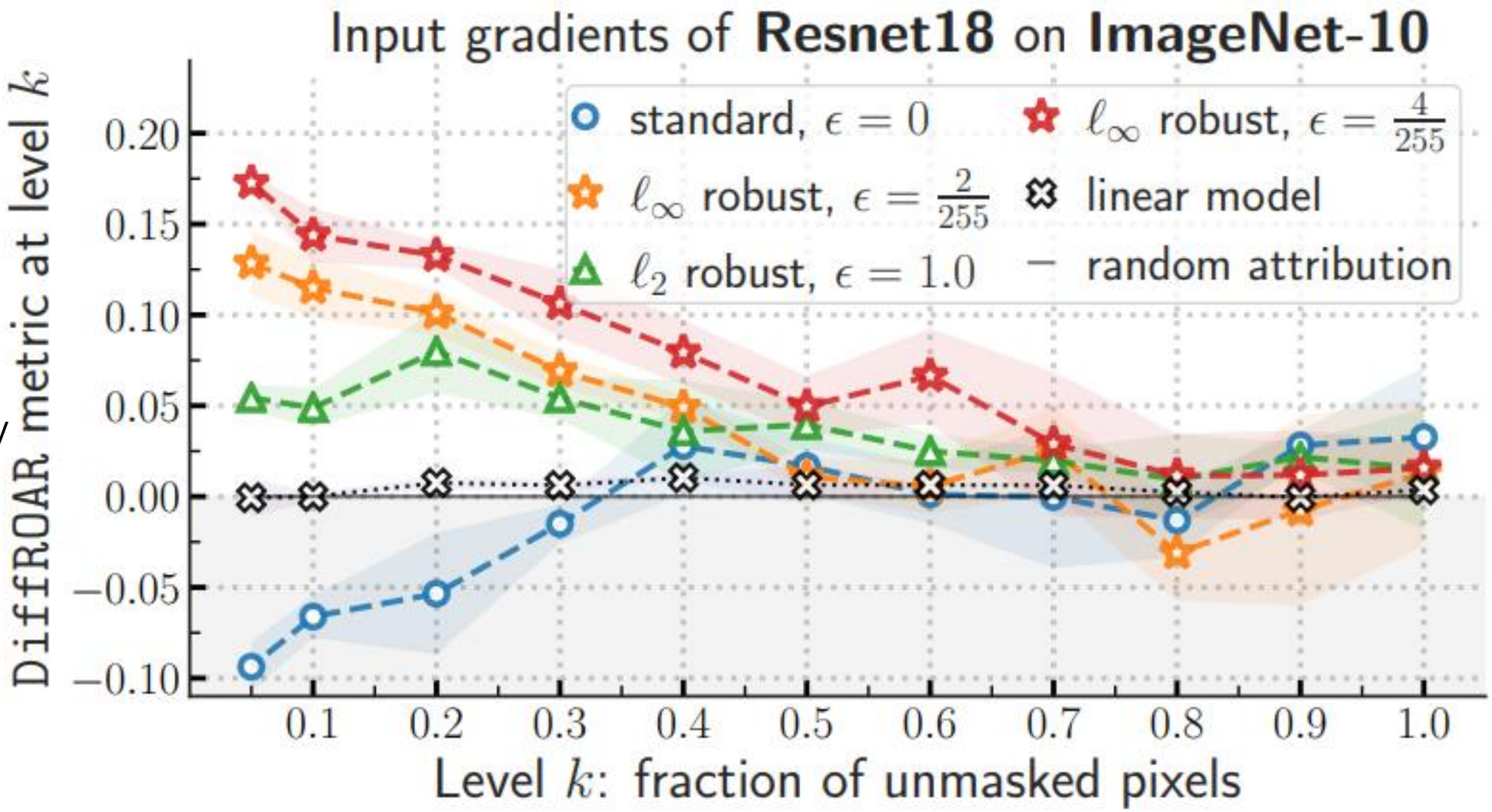


Google Research

Do Trained Models Satisfy [A]?

Standard models violate [A].

Robust models satisfy [A].

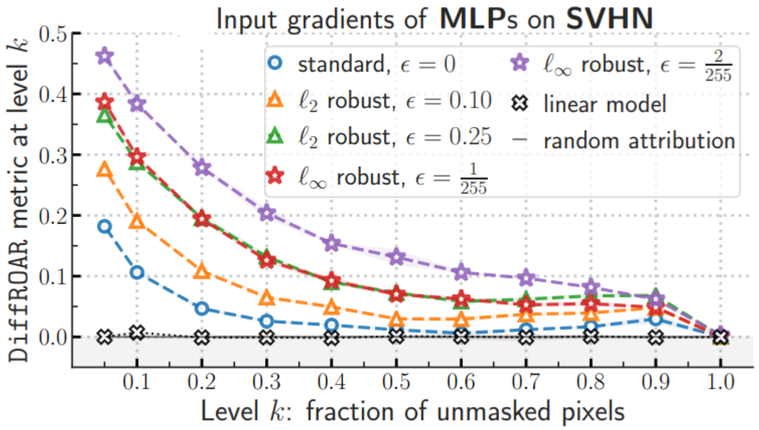


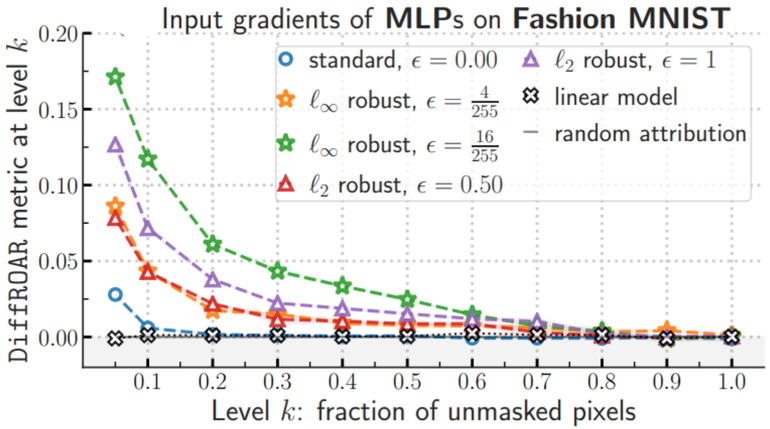
DiffROAR(k) $\stackrel{\text{def}}{=}$ PredTopK – PredBottomK

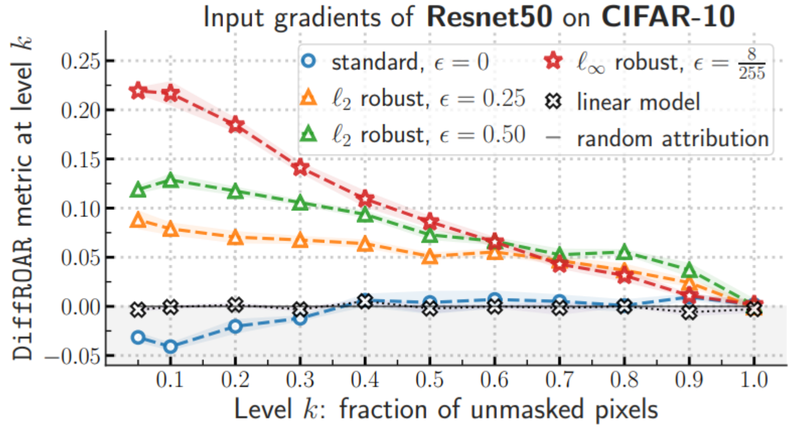
Google Research

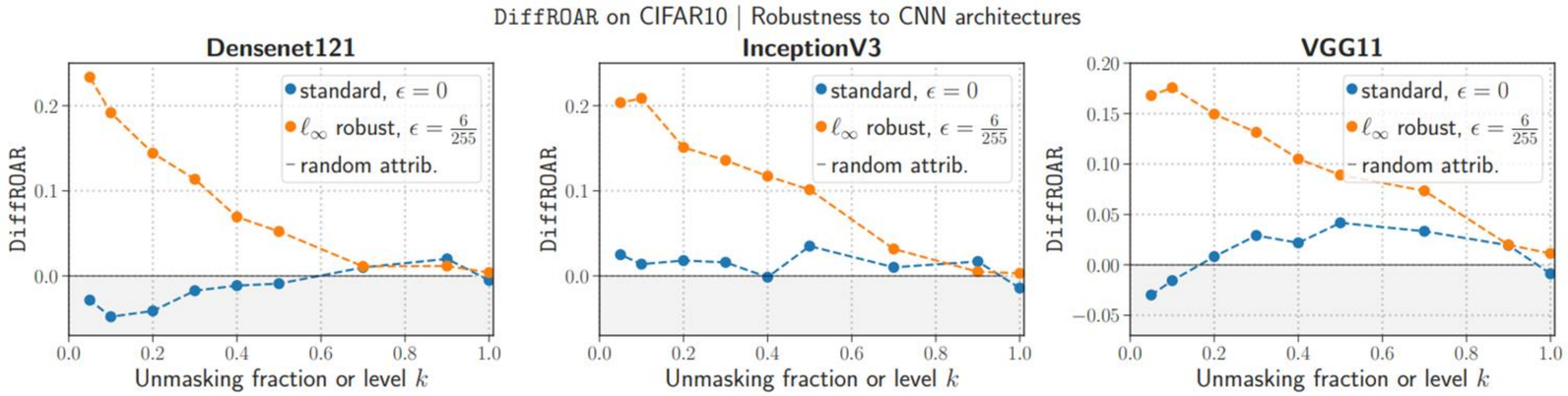
Consistency of Results

Across Datasets, Architectures, Training Hyperparameters etc.









Google Research

Are These Results Conclusive?

Lack of Ground Truth Issue

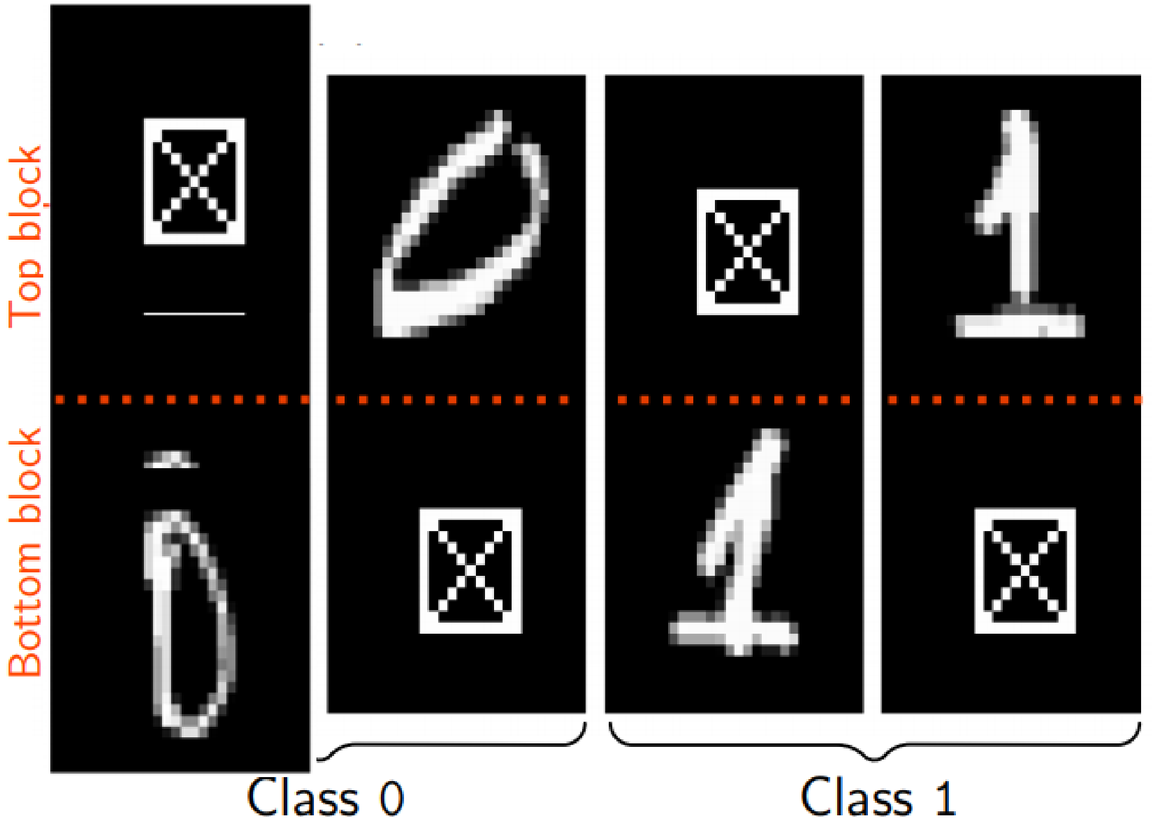
- $\text{DiffROAR}(k) \stackrel{\text{def}}{=} \text{PredPowerTopK} - \text{PredPowerBottomK}$
- **Key assumption**: Retrained models learn the same features as original model.
  - Intuitively since same model architectures learn *similar* features Li et al. (2015), Hacohen et al. (2020). Also from *simplicity bias*.
- Nevertheless, still an assumption.
- Definitive conclusions require knowledge of **ground truth features used by the model**.
- We will design a principled **Block-MNIST** dataset to this end.

Block-MNIST Dataset

- **Design principles**
  - Force discriminative and non-discriminative features in different parts of the image
  - Low level patterns such as edges are shared across images

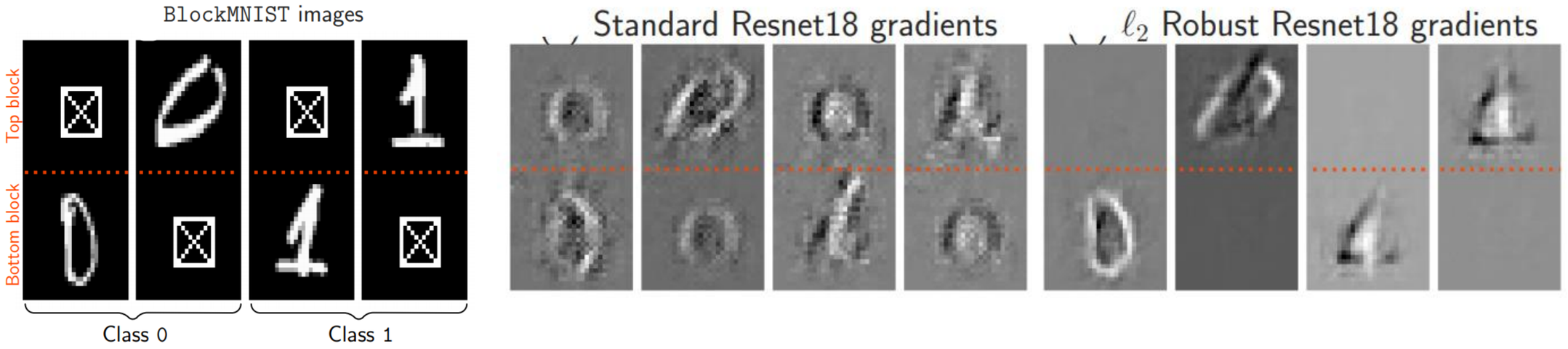
- Signal = MNIST and Null = Pattern

- [A] ⇔ Input gradients highlight **Signal** block over **Null**



Google Research

Input Gradients on Block-MNIST

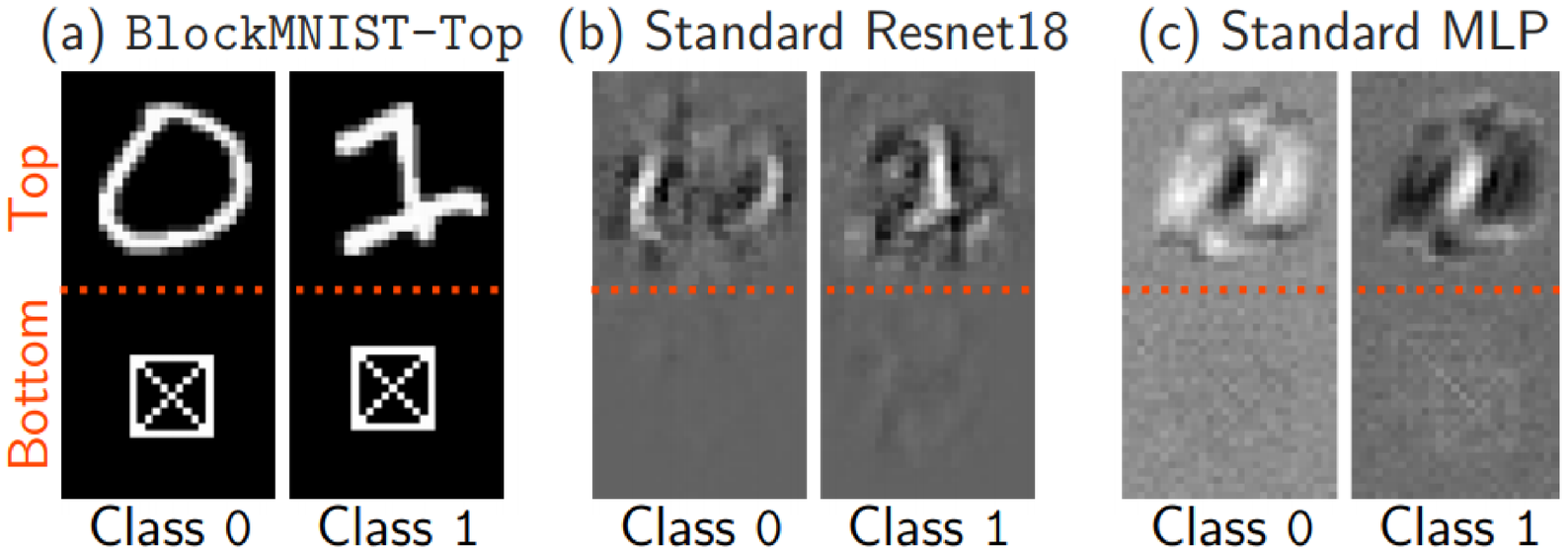


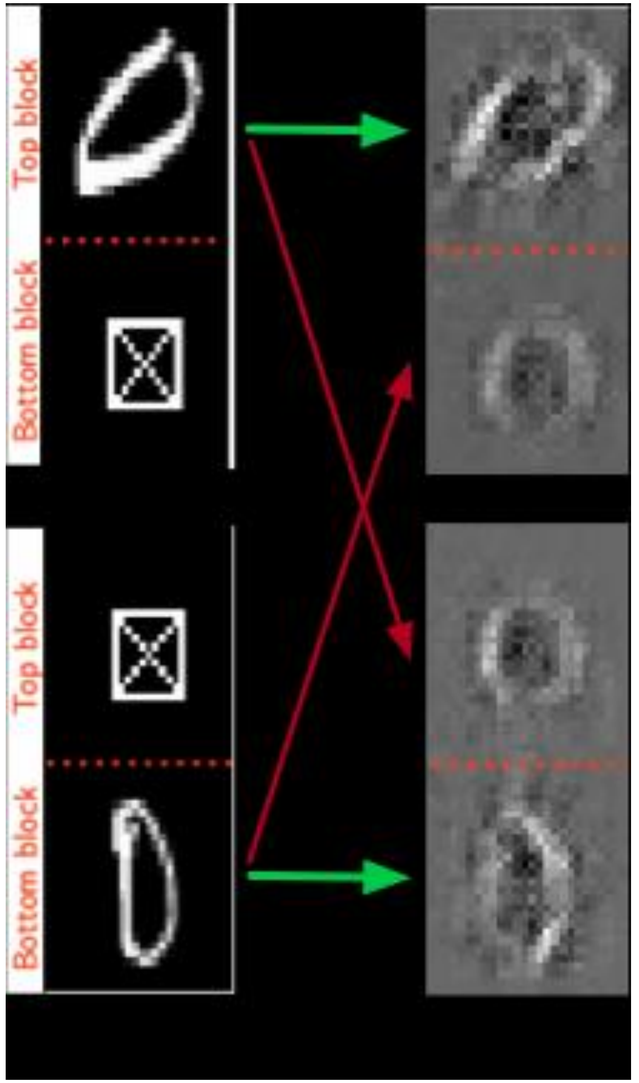
- Standard model gradients highlight **both** signal and null blocks.
- Robust model gradients highlight signal block over null block.
- Results consistent across other pairs and also multi-class MNIST.

Why Do Standard models violate [A] ?

Our hypothesis: **Feature Leakage**

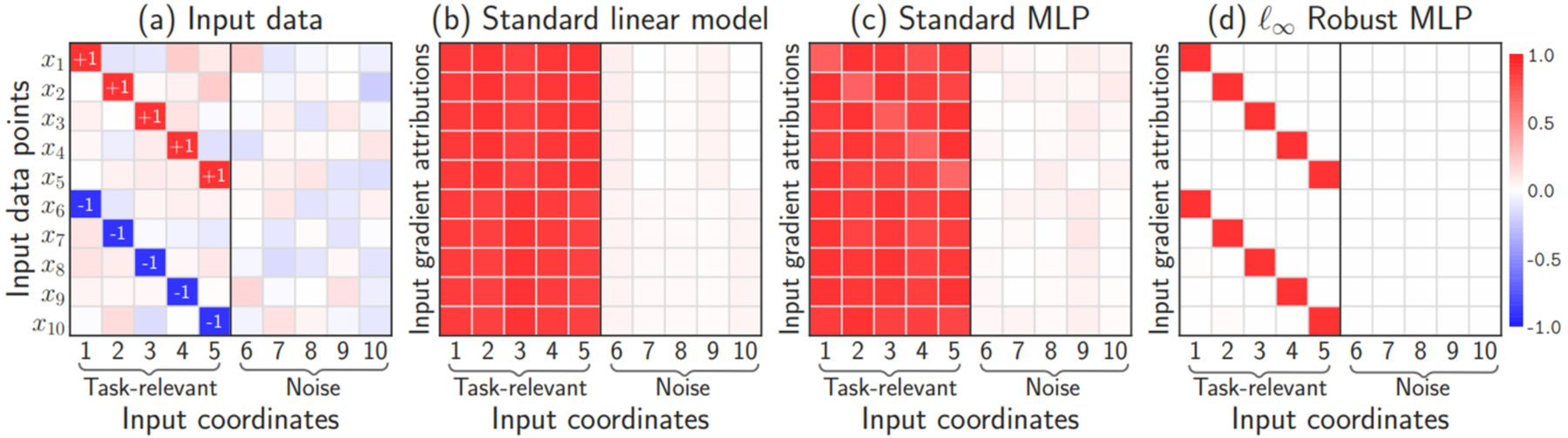
Input gradients leak discriminative features from one instance to another!





Google Research

Theoretical Results



- For data above standard MLP models trained with GD do not satisfy [A]
  - We consider infinite width 1-hidden layer MLPs with ReLU activation
- Key idea: Characterize the max-margin classifier + results of Chizat & Bach (2020)
- Partial result: input gradients of adversarially trained models satisfy (A)

Rigorous Proof of Feature Leakage

On a Simplified Dataset

- Signal vector $u^* \in \mathbb{R}^{\hat{d}}$ and noise vectors $g_1, \cdots, g_d \in \mathbb{R}^{\hat{d}}$

$$y = \pm 1 \text{ with probability } 0.5$$

$$x = [g_1, \cdots, yu^* + g_i, \cdots, g_{\frac{d}{2}}, g_{\frac{d}{2}+1}, \cdots, g_d]$$

One Signal block chosen uniformly from $\left\{1, \cdots, \frac{d}{2}\right\}$

Remaining blocks are Null blocks

$\left\{\frac{d}{2}+1, \cdots, d\right\}$ are Null blocks in all examples

- Feature leakage ⇔ All $\left\{1, \cdots, \frac{d}{2}\right\}$ blocks highlighted on all examples over $\left\{\frac{d}{2}+1, \cdots, d\right\}$

Google Research

Main Theoretical Result

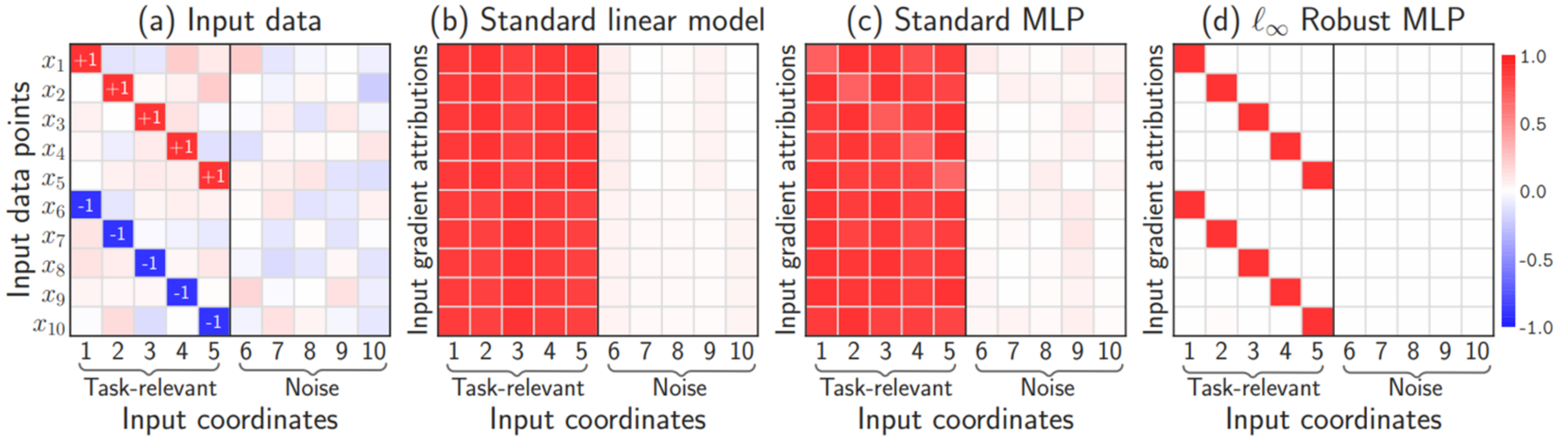
For infinite width, 1-hidden layer NN with ReLU activations, the max-margin classifier satisfies for any example:

$$\|\text{Input gradient on } k^{\text{th}} \text{block}\| \begin{cases} = c > 0 \text{ if } k \leq \frac{d}{2} \\ = 0 \quad \text{if } k > \frac{d}{2} \end{cases}$$

- **Key Idea**: Chizat and Bach (2020) shows that in the infinite width limit, if GD converges, it does so to max-margin classifier.
- Explicitly derive max-margin classifier using optimality conditions.
- Fully explains the behavior of standard models.

Additional Results

- For adversarially trained models, GD converges to a different max-margin classifier.
- However, the optimality conditions are significantly more complicated.
- Partial results showing that input gradients of adversarially trained models satisfy (A).



Google Research

Conclusions

- NNs exhibit extreme Simplicity Bias (SB)
- This extreme SB leads to brittleness of NNs in practice
- One approach to identify SB – strong feature attribution techniques!
- But existing techniques seem to suffer from **feature leakage**
  - **Possible Fix:** adversarially trained models!

Google Research

Open Questions and Future Directions

- Similarly thorough evaluation of other attribution methods.
- How do we tackle SB in NNs?
- Currently an active area of research with different approaches
  - With additional group information of different examples
  - Training a new model away from features of an initially trained **simple** model
  - Simultaneously training diverse models to be combined as an ensemble
- Question still wide open

Thanks!

https://arxiv.org/abs/2006.07710

https://arxiv.org/abs/2102.12781

Google Research