Tutorial | SIGKDD 2021

Machine Learning *Explainability* and *Robustness*

➡ Connected at the Hip

[Anupam Datta](#)
[Matt Fredrikson](#)
[Klas Leino](#)
Kaiji Lu
[Shayak Sen](#)
[Zifan Wang](#)


Anupam Datta


Matt Fredrikson


Caleb Lu


Klas Leino


Shayak Sen


Zifan Wang

Machine Learning Systems are Ubiquitous

Google

April 3, 2013, Vol 309, No. 13 >

< Previous Article | Next Article >

Viewpoint | April 3, 2013

The Inevitable Application of Big Data to Health Care

Travis B. Murdoch, MD, MSc; Allan S. Detsky, MD, PhD

[+] Author Affiliations

Big Data in Government, Defense and Homeland Security 2015 - 2020

Share +1 Pin it

NEW YORK, May 12, 2015 /PRNewsw

How Big Data Could Replace Your Credit Score

Credit scores are useful in determining who gets loans, but they're far from perfect. AvantCredit determines loan-worthiness based on all sorts of factors, including your use of social media and prepaid cell phones.

Big Data in Education

Learn how and when to use key methods for educational data mining and learning analytics on large-scale educational data.

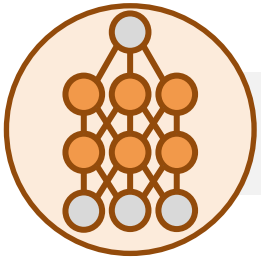
TEACHERS COLLEGE
COLUMBIA UNIVERSITY

amazon

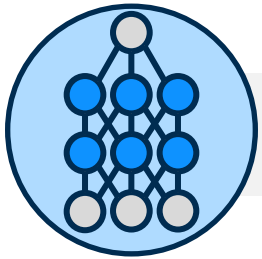
facebook

bing

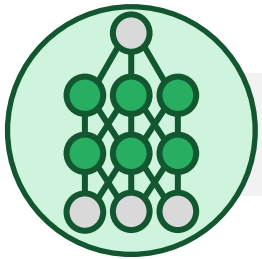
Motivating example



model 1



model 2

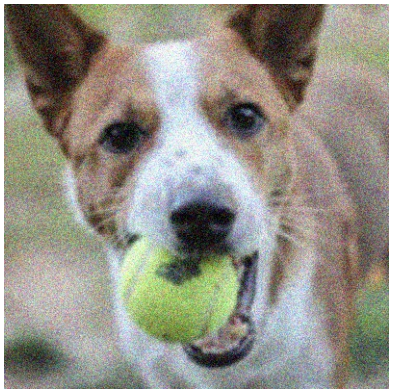


model 3

dog

dog

dog



sports car

dog

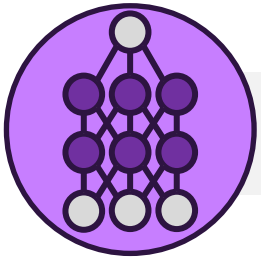
dog

cat

dog

cat

Can we trust a new model based on its predictions?

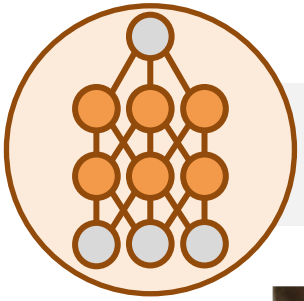


model 4

prediction agrees with all previous models on our validation point; how do we know which model's behavior model 4 will best resemble?

dog	sports car	cat
dog	dog	dog
dog	dog	cat
dog	?	?

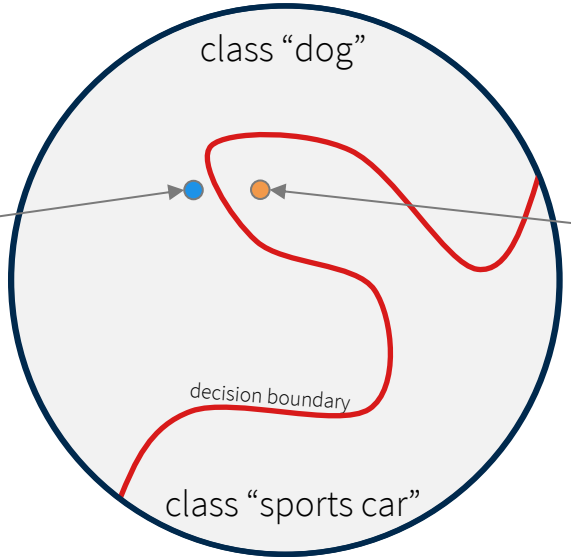
Problem: lack of robustness

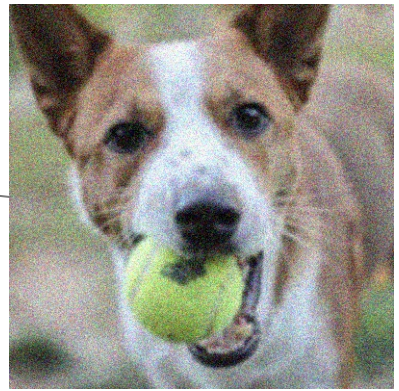


model 1

model 1 is not *robust*

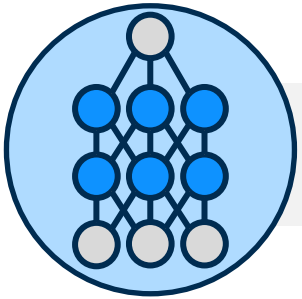
dog





sports car

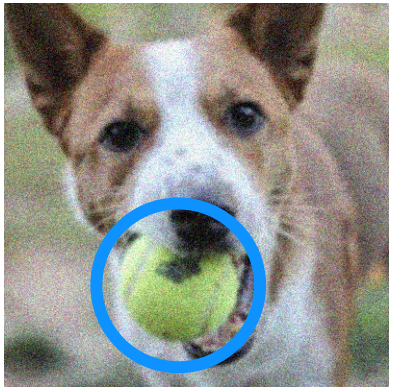
Problem: lack of conceptual soundness



model 2

model 2 is not *conceptually sound*

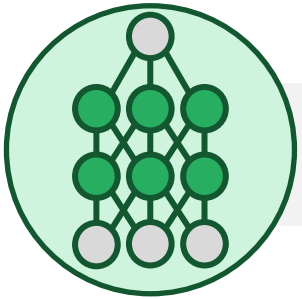
dog



dog

dog

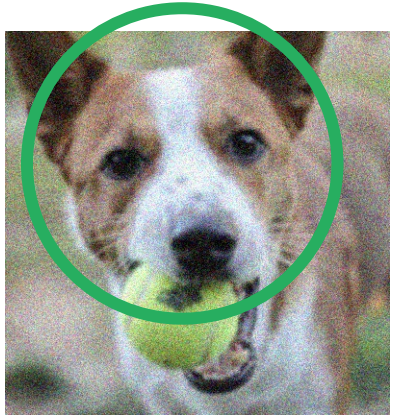
Ideal: robustness and conceptual soundness



model 3

model 3 is *robust* and *conceptually sound*

dog



dog

cat

Takeaways

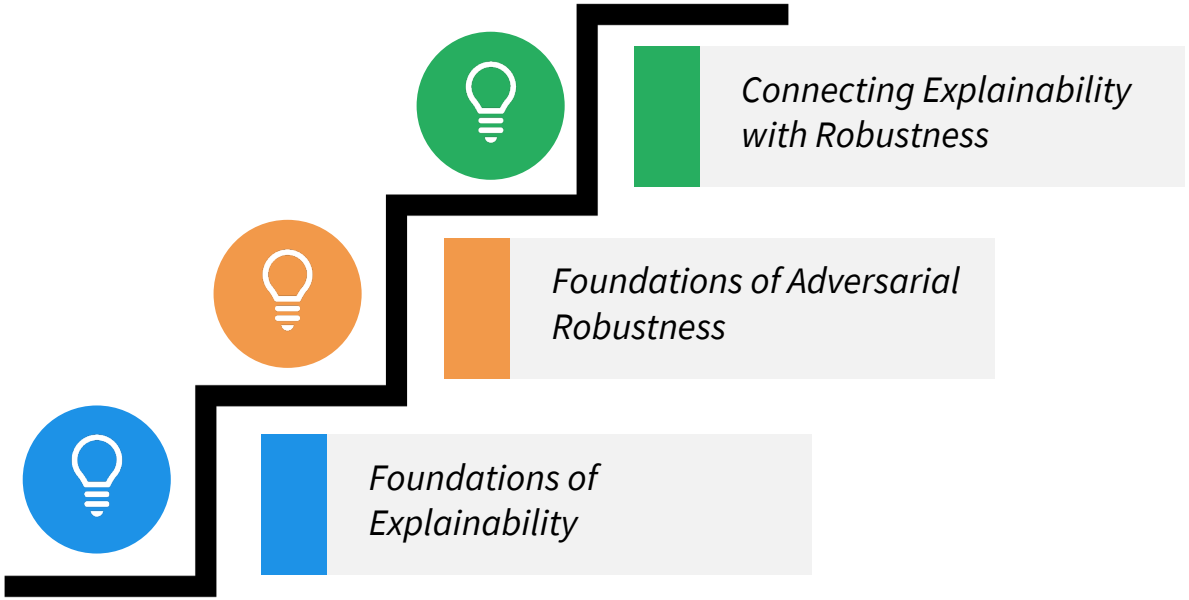
robustness

conceptual soundness

robustness serves as a necessary foundation for conceptual soundness

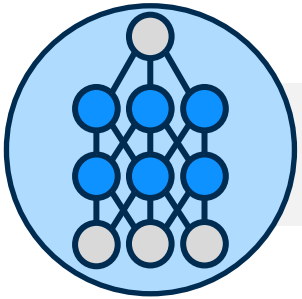
explanations help us assess whether a model's behavior is conceptually sound

Tutorial Roadmap



Section I
Foundations of XAI

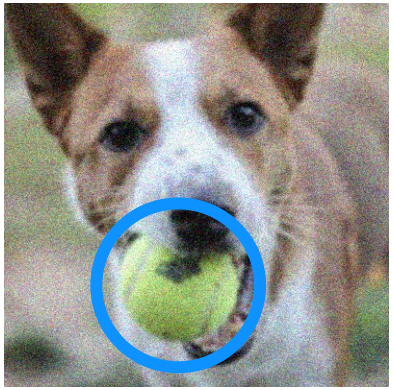
Explainability helps assess conceptual soundness



model 2

model 2 is not *conceptually sound*

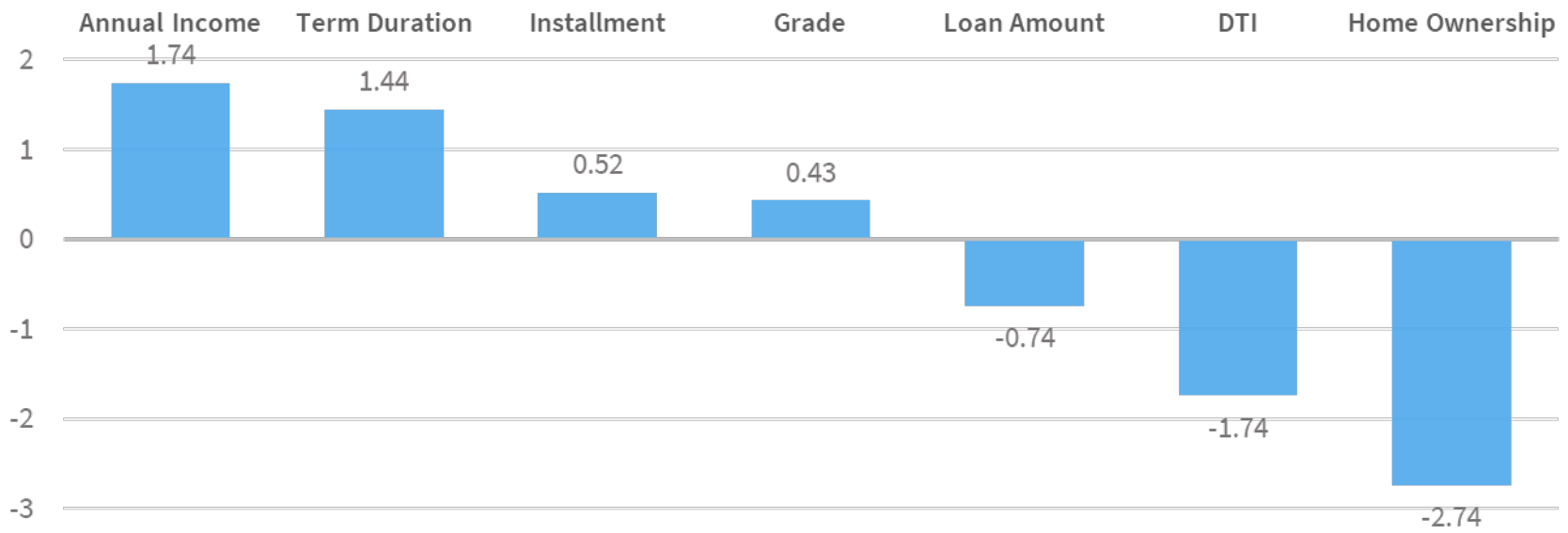
dog



dog

dog

Input feature importance for a tree model



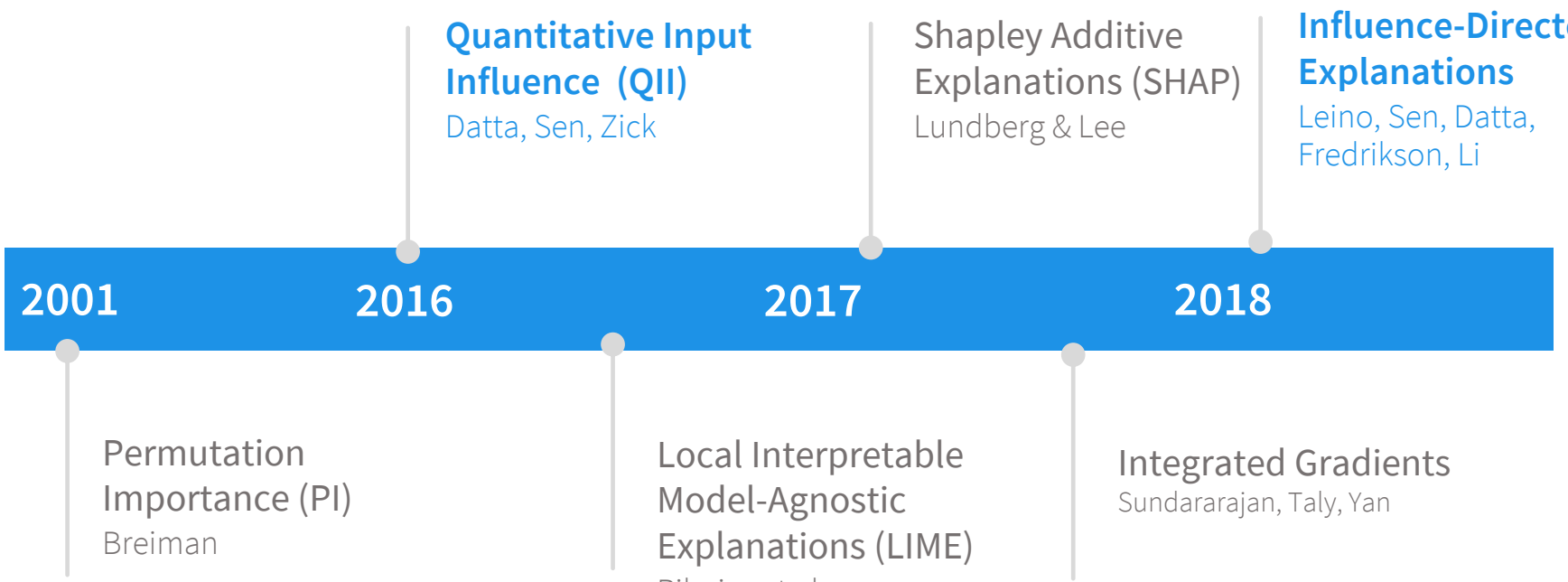
Internal explanation for a deep network

What makes Orlando Bloom Orlando Bloom?

Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li, ITC ‘18

Methods for Computing Feature Importance

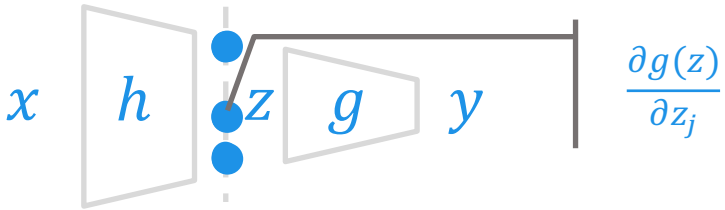


Requirements for Good Explanations

- Explanation Accuracy
  - Accurately reflect reasons for model behavior
- Generality
  - Answer rich set of questions
- Interpretation devices
  - To make explanations meaningful to human users

Satisfied by a class of gradient-based explanation methods for deep neural networks

Accuracy of Gradient Explanations



$$y = F(x) = g(z), z = h(x)$$

Feature Importance = Average gradient over a relevant distribution

- **Causality**. Gradients capture causal impact of features on model behavior
- **Distributional Marginality**. If gradients of output wrt a feature are equal for two different networks on a relevant distribution, then that feature is equally important for both networks

Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li, ITC ‘18

Integrated Gradient

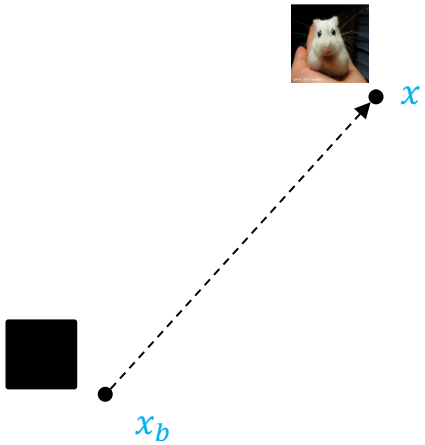
Shapley Value → (continuous features) Aumann-Shapley → (neural network) Integrated Gradient

[Sundararajan et al. ICML 2017]

$$IG(x;\ x_b, F) = (x - x_b)\int_0^1 \frac{\partial F(\gamma(\alpha; x, x_b))}{\partial \gamma} d\alpha$$

$$\text{where } \gamma(\alpha; x, x_b) = x_b + \alpha(x - x_b)$$

Average gradient of points on a linear path from a baseline to the target input



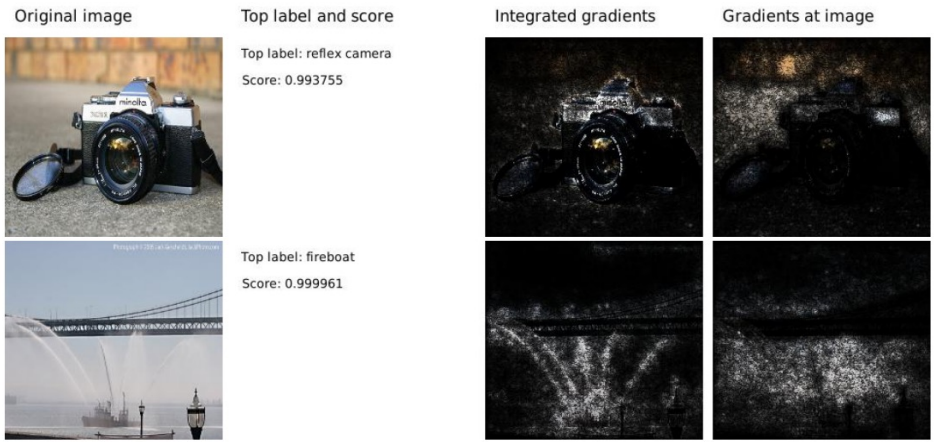
Integrated Gradient

Shapley Value → (continues features) Aumann Shapley → (differentiable output) Integrated Gradient

[Sundararajan et al. ICML 2017]

Integrated Gradient is the **only** path method that satisfies

- Symmetry
- Dummy
- Efficiency(Completeness)
- Additivity



Now It's Time to Dive Deeper…

Input Attributions → **Internal** Attributions

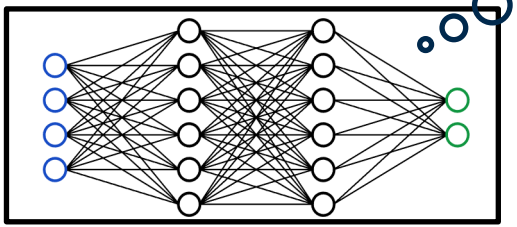
Why are we interested in internal representations?

 → Deep Neural Network → "Sports Car"

Now It's Time to Dive Deeper…

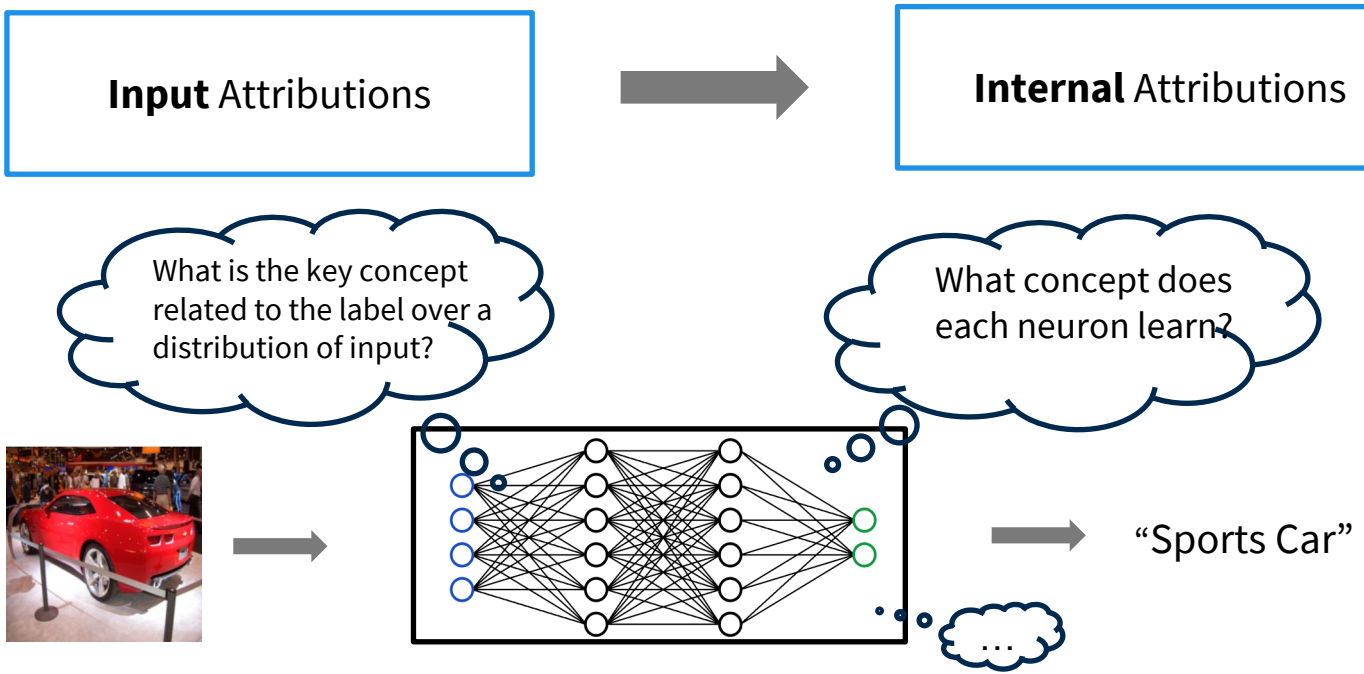
Input Attributions → **Internal** Attributions

Why are we interested in internal representations?



"Sports Car"

Now It's Time to Dive Deeper…



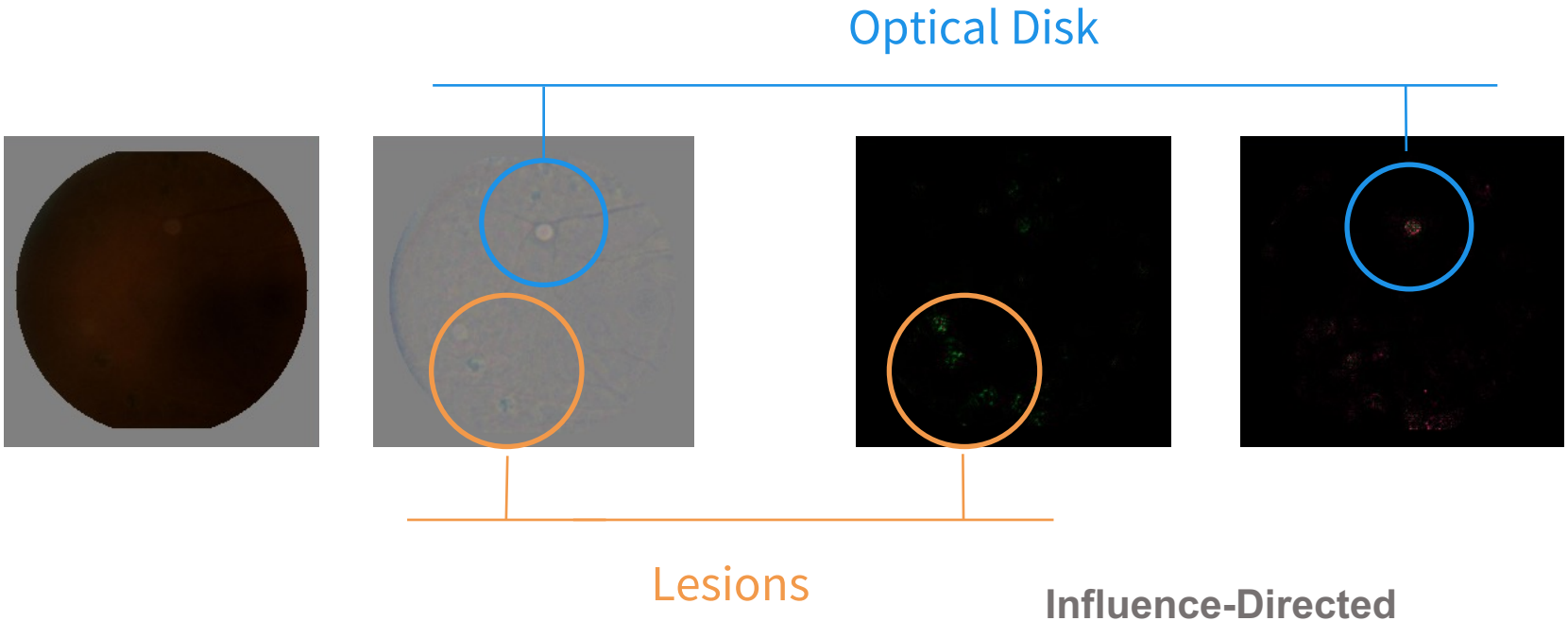
What Makes Orlando Bloom Orlando Bloom?

Internal explanation for a deep network

Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li, ITC ‘18

Detecting Diabetic Retinopathy Stage 5

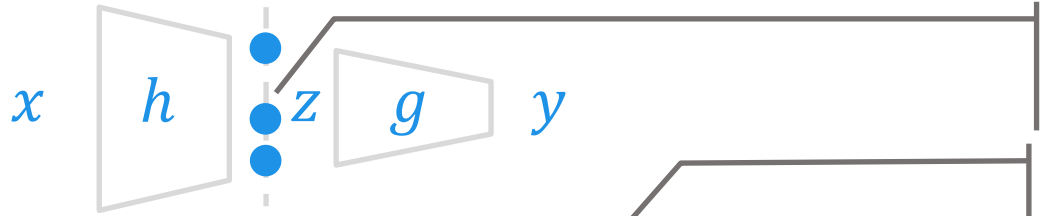


Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li 2018

Distributional Influence

Influence = average gradient over distribution of interest



$$y = F(x) = g(z), z = h(x)$$

$$\mathrm{I}_j^s(F, P) := \int_{x \in X} \frac{\partial g(z)}{\partial z_j} P(x) dx$$

z_j

Gradient

For input x
[note z = h(x)]

Weighted by probability of input x

Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li, ITC '18

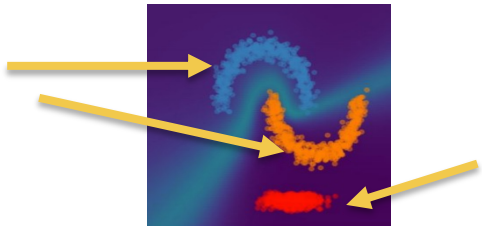
Axiomatic Foundation for Distributional Influence

$$\mathrm{I}_j^s(F,P) := \int_{x \in \mathcal{X}} \frac{\partial g(z)}{\partial z_j} P(x) dx$$

When s is the input slice($h(x) = x$), Distributional Influence satisfies:

- **Distributional Marginality**: If the partial derivatives w.r.t. an input feature are identical for F_1, F_2 over the distribution of interest, then $\mathrm{I}_j^s(F_1,P) = \mathrm{I}_j^s(F_2,P)$
- **Linear Agreement**
- **Completeness**
- **…**

We are interested in



We are **not** interested in

Influence-Directed Explanations

Leino, Sen, Fredrikson, Datta, Li ITC '18

Images Source: Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness [Liu et al. 2020]

Distributional Influence Generalizes Existing Methods

$$\mathrm{I}_j^s(F,P) := \int_{x \in \mathcal{X}} \frac{\partial g(z)}{\partial z_j} P(x) dx$$

When s is the input slice($h(x) = x$)

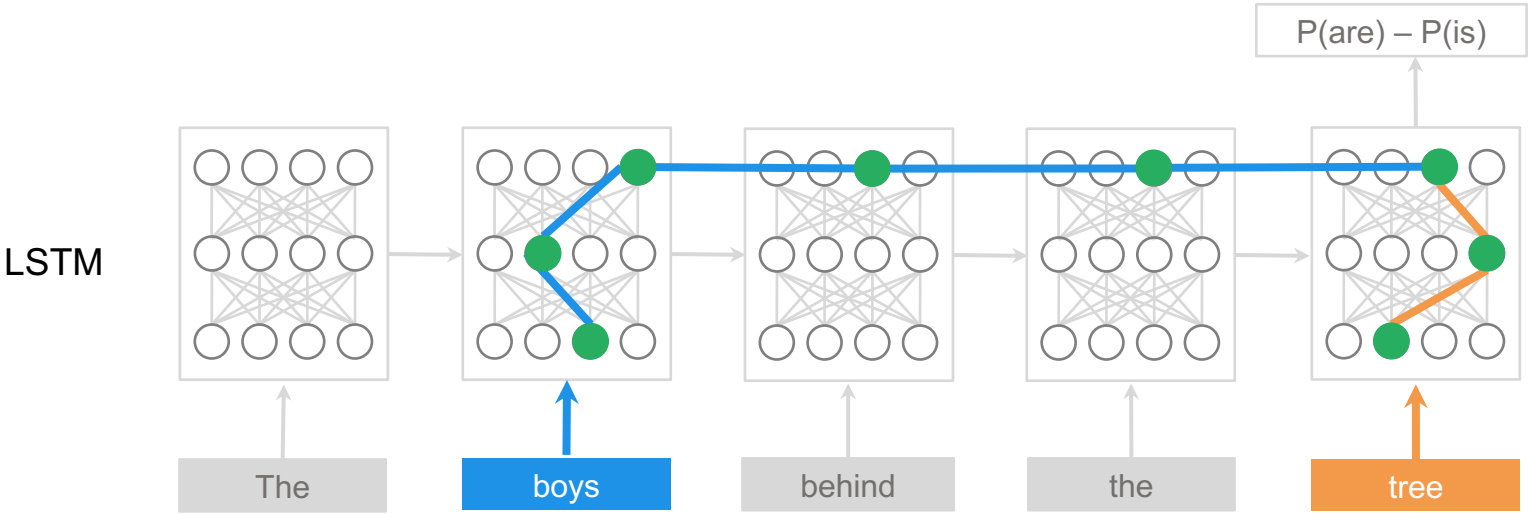
- and $\mathcal{X}$ is a set of points (uniformly) distributed on a linear path from a baseline input to the target input

multiplying $\mathrm{I}_j^s(F,P)$ with $(x - x_b)$ → Integrated Gradient *[Sundararajan et al. 2017]*

- and $\mathcal{X}$ is a set of points in the Gaussian Distribution centered with the target input → Smooth Gradient *[Smilkov et al. 2017]*

⋮

Internal Explanations via Influence Paths



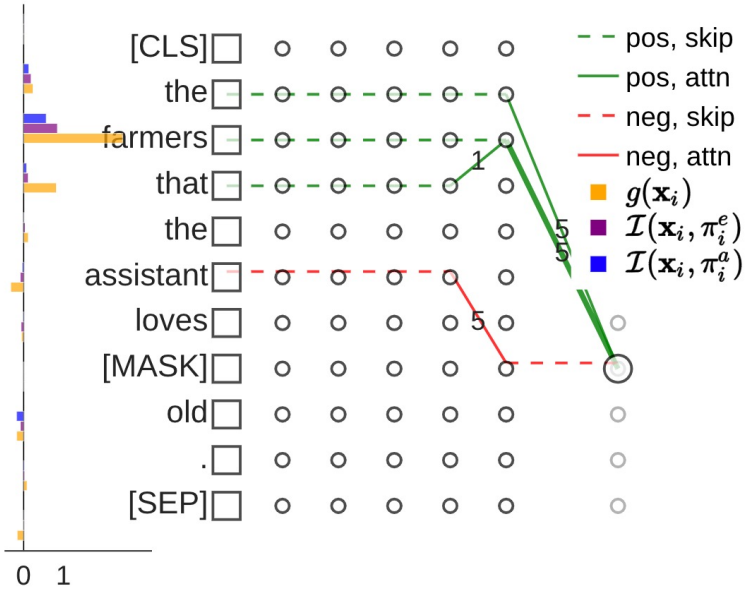
- Influence paths provide insights into misclassifications
- Model can be compressed down the influential paths without changing the utility of the model

Influence Paths
Lu, Mardziel, Leino, Fedrikson, Datta, ACL '20

Influence Graphs for BERT

BERT v.s. LSTM

- Scaling up method to identify influential paths
- Prevalence of “copy” and “transfer” operations to carry context



Influence Graphs for BERT
Lu, Wang, Mardziel, Datta, 2020

Related Work

	Explanation Framework Properties			*Influence Properties*	
	Quantity	**Distribution**	**Internal**	**Marginality**	**Sensitivity**
Influence-Directed Explanation *[Leino et al. ITC '18]*	✓	✓	✓	✓	✓*
Conductance *[Dhamdhere et al. ICLR '19]*		✓⁻	✓	✓	✓
Integrated Gradient *[Sundararajan et al. ICML '17]*		✓⁻		✓	✓
Smooth Gradient *[Smilkov et al. 2017]*		✓⁻		✓	✓
Simple Taylor *[Bach et al. 2015 PLOS ONE]*		✓⁻		✓	
Deconvolution *[Zeiler et al. ECCV '14]*			✓†		
Guided Backpropagation *[Springenberg et al. 2015 ICLR Workshop]*			✓†	✓	
Layer-wise Relevance Propagation *[Bach et al. 2015 PLOS ONE]*		✓⁻	✓†	✓*	✓*

✓ Supports ✓⁻ Limited flexibility ✓* Supports under some parameterizations ✓† Internal influence as an intermediate step

Takeaways: Requirements for Good Explanations

- Explanation Accuracy
  - Accurately reflect reasons for model behavior
- Generality
  - Answer rich set of questions
- Interpretation devices
  - To make explanations meaningful to human users

Satisfied by a class of gradient-based explanation methods for deep neural networks

Demo TruLens

Library containing attribution and interpretation methods for deep nets.

```
pip install trulens
```

TruLens is supporting models built with

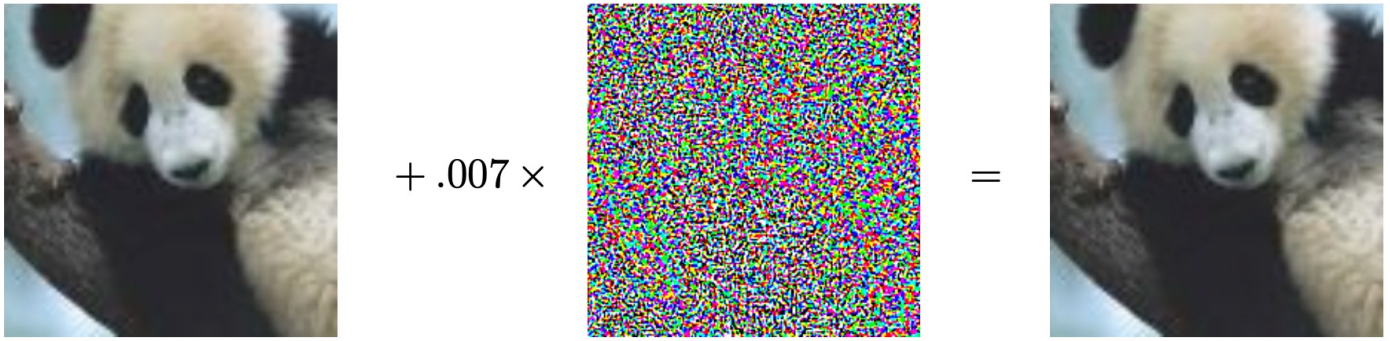
Q & A [15 min]

Break I

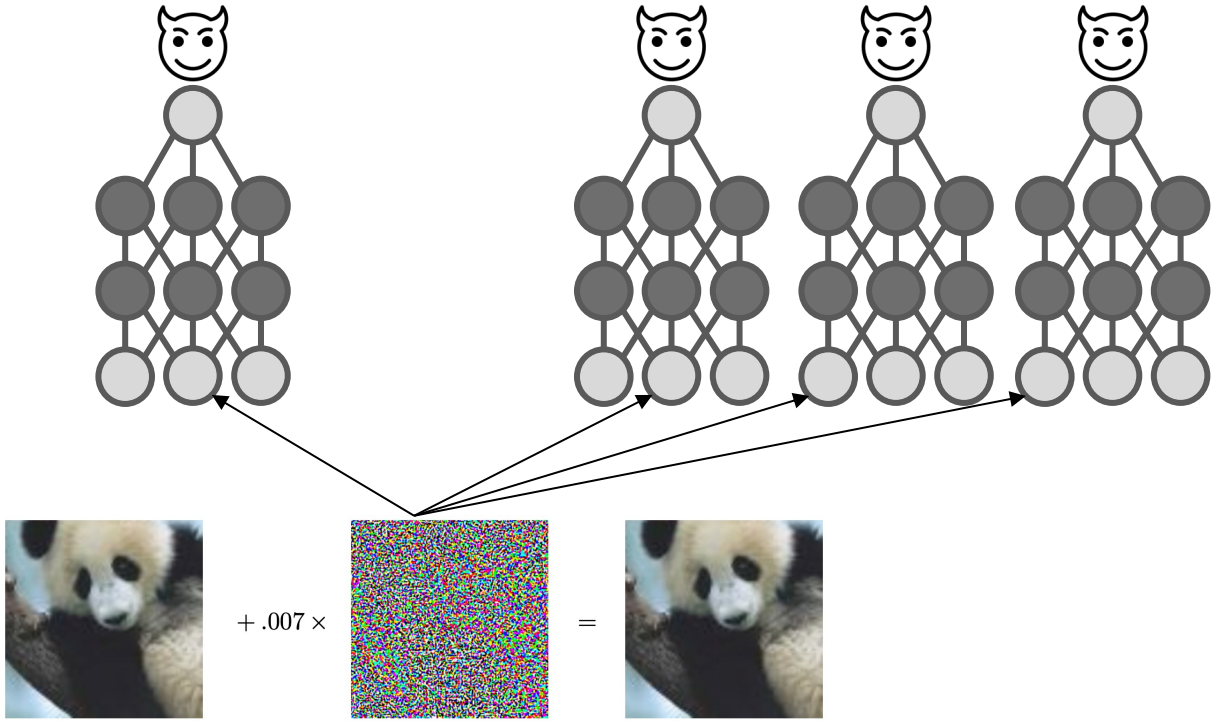
Section II
Foundations of Adversarial Robustness

Intriguing Properties of Neural Networks [Szegedy et al. 2014]

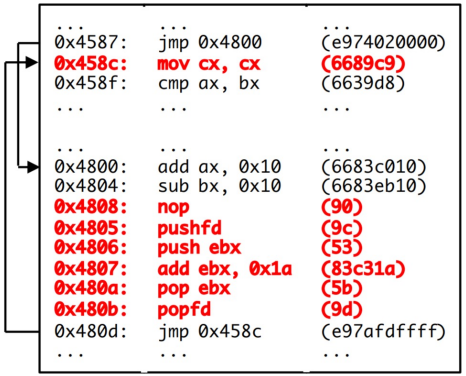
"We find that applying an imperceptible non-random perturbation to a test image, it is possible to arbitrarily change the network's prediction"



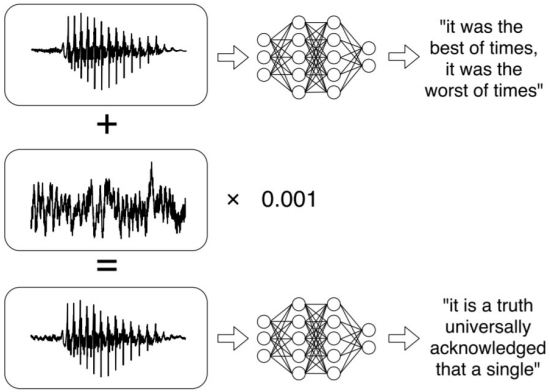
Adversarial examples generalize



It's not just images!



Malware Detection
[Lucas et al., 2021]



Audio
[Carlini & Wagner, 2018]

South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism. 57% **World**

South Africa's historic Soweto township marks its 100th birthday on Tuesday in a moo**P** of optimism. 95% **Sci/Tech**

Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the opposition Conservatives. 75% **World**

Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the o**B**position Conservatives. 94% **Business**

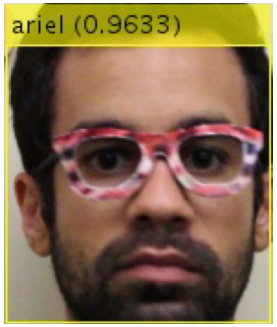
Text
[Ebrahimi et al., 2018]

Is this really a problem?

Perhaps models just learn different concepts than humans

Some applications need more assurance

Secure biometrics [Sharif et al., 2016]



Safe autonomy

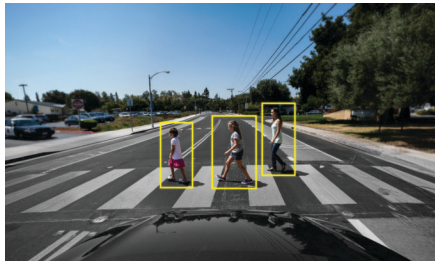
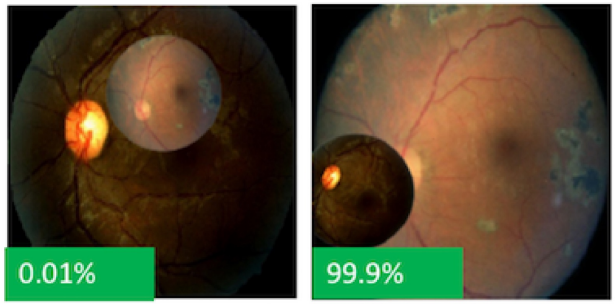
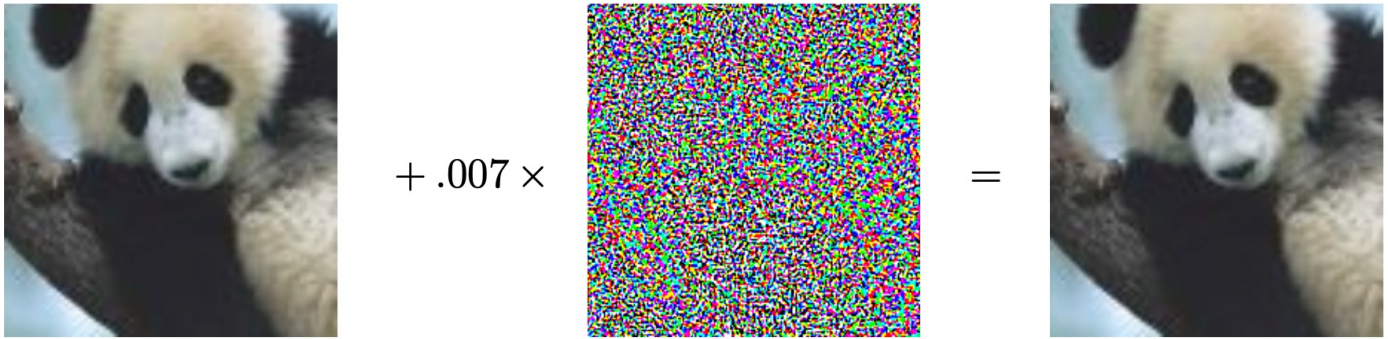


Image source: nvidia.com

Trustworthy Diagnostics [Finlayson et al., 2019]



Can we explain vulnerable models?

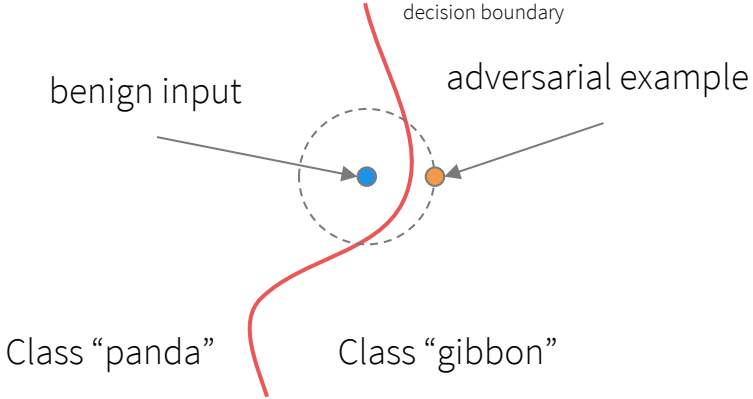


"panda" Important features? "gibbon"

Adversarial examples are incompatible with conceptually-sound models

Adversarial examples are a violation of *local robustness*

“panda”



“gibbon”

Local robustness

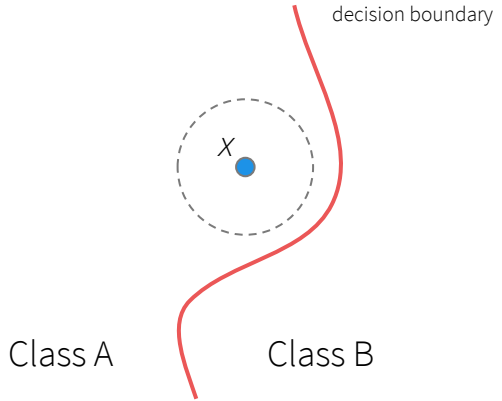
Definition

A classifier, F, is ϵ-*locally-robust* at x if $\forall x'$,

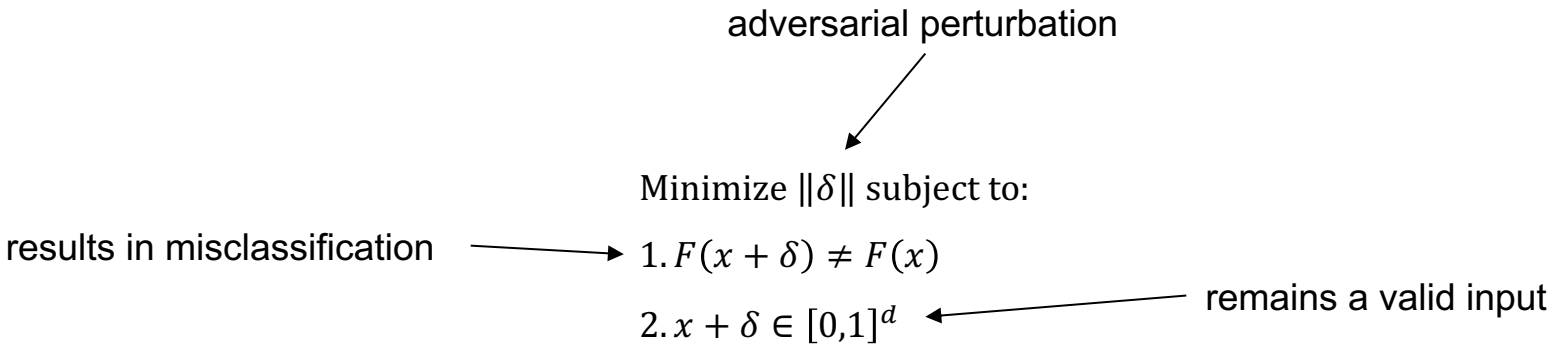
$$\|x - x'\| \leq \epsilon \implies F(x) = F(x')$$

L_p norm, typically p ∈ {1,2, ∞}

i.e., the model makes the same prediction on all points in the ε-ball centered at *x*

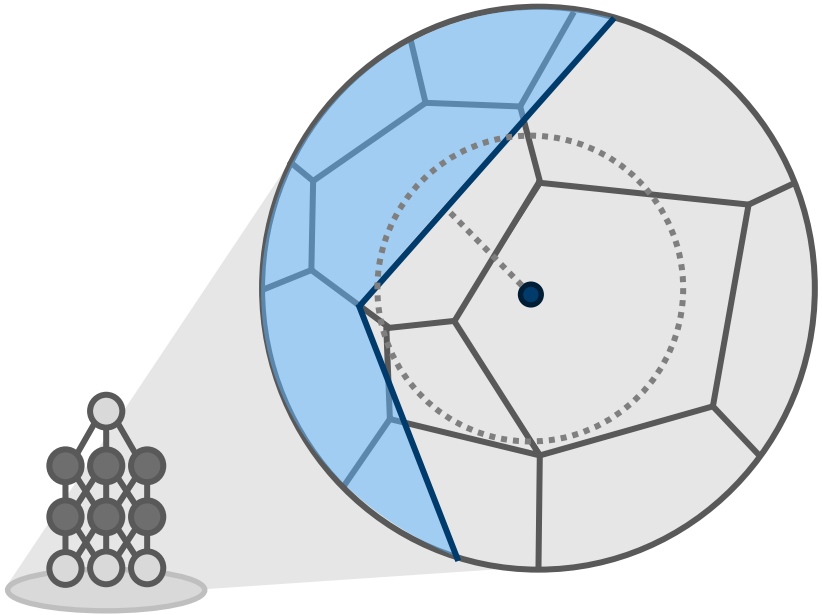


Finding violations



Exhaustive search

- ReLU networks are *piecewise-linear*
- Components comprise a *polyhedral complex*
- Each *activation pattern* of neurons gives a polyhedral region with linear boundaries
- Use constraint-solving to find distance to boundary for a point [Jordan et al. 2019]



This is computationally infeasible for most networks

Finding violations, efficiently

Minimize $\|\delta\|$ subject to:

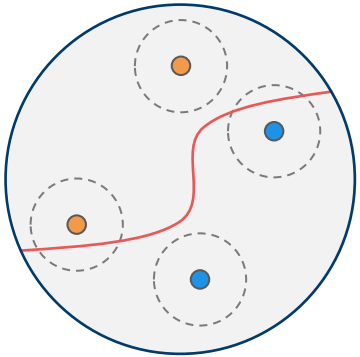
1. $F(x+\delta) \neq F(x)$
2. $x+\delta \in [0,1]^d$

Maximize $L\big(x+\delta, F(x)\big)$ subject to $\|\delta\| < \epsilon, x+d \in [0,1]^d$

- Loss function *L* is cross-entropy
- New formulation solved with *projected gradient descent*
- Despite non-convexity, this works well in practice
- Tune by varying number of PGD steps, step size

Building robust models

- Goal: push boundaries away from data manifold
- Not of interest: detecting or filtering potential attacks, modifying inputs, stochastic smoothing



[Goodfellow et al., 2015]
[Madry et al., 2016]

Standard Training

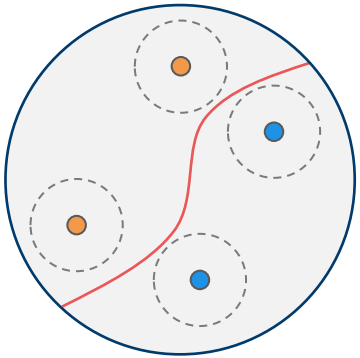
$$\min_{\theta} \mathbb{E}_{x,y\sim D}[L(\theta, x, y)]$$

- D is the training data
- θ are the model parameters

Adversarial Training

$$\min_{\theta} \mathbb{E}_{x,y\sim D}[\max_{\|\delta\|<\epsilon} L(\theta, x+\delta, y)]$$

- D is the training data
- θ are the model parameters
- ϵ is the local robustness radius



Are adversarially-trained models robust?

Adversarial Training

$$\min_{\theta} \mathbb{E}_{x,y\sim D}[\max_{\|\delta\|<\epsilon} L(\theta, x+\delta, y)]$$

Small loss guarantees that no attack will succeed
[Madry et al., 2017]

Practical hurdles remain:

- Unclear whether we can sufficiently minimize loss
- This statement applies to the **training data**
- What can we say about test data?

In practice, adversarial training does not guarantee a specific degree of robustness

Uncertain results

Reported degree of protection depends on the attack used to evaluate it!

Takeaways for AT:

- Difficult to train a robust model
- More difficult to evaluate its robustness on new data

CIFAR-10 - l_∞ - $\epsilon = 8/255$			originally reported	new attack
1	(Carmon et al., 2019)	89.69	62.5	-2.97
2	(Alayrac et al., 2019)	86.46	56.30	-0.27
3	(Hendrycks et al., 2019)	87.11	57.4	-2.48
4	(Rice et al., 2020)	85.34	58	-4.58
5	(Qin et al., 2019)	86.28	52.81	0.03
6	(Engstrom et al., 2019)	87.03	53.29	-4.04
7	(Kumari et al., 2019)	87.80	53.04	-3.92
8	(Mao et al., 2019)	86.21	50.03	-2.62
9	(Zhang et al., 2019a)	87.20	47.98	-3.15
10	(Madry et al., 2018)	87.14	47.04	-3.00
11	(Pang et al., 2020)	80.89	55.0	-11.52
12	(Wong et al., 2020)	83.34	46.06	-2.85
13	(Shafahi et al., 2019)	86.11	46.19	-4.72
14	(Ding et al., 2020)	84.36	47.18	-5.74
15	(Moosavi-Dezfooli et al., 2019)	83.11	41.4	-2.90
16	(Zhang & Wang, 2019)	89.98	60.6	-23.96
17	(Zhang & Xu, 2020)	90.25	68.7	-32.25
18	(Jang et al., 2019)	78.91	37.40	-2.45
19	(Kim & Wang, 2020)	91.51	57.23	-23.01
20	(Moosavi-Dezfooli et al., 2019)	80.41	36.3	-2.60
21	(Wang & Zhang, 2019)	92.80	58.6	-29.25
22	(Wang & Zhang, 2019)	92.82	66.9	-39.97
23	(Mustafa et al., 2019)	89.16	32.32	-32.04
24	(Chan et al., 2020)	93.79	15.5	-15.24

Provable robustness

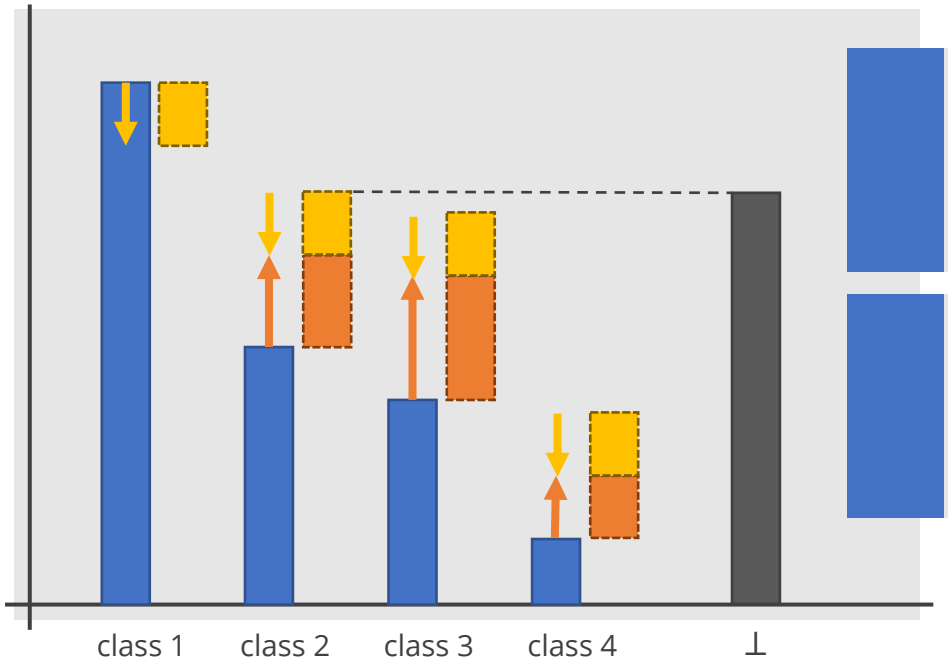
Goal: train the model so that it can be proven robust on new points

A function *f* is *(K, δ)* locally-Lipschitz continuous iff for all *x*, there exists a neighborhood U_δ around *x* such that $f|_{U_\delta}$ is *K*-Lipschitz continuous:

$$\forall x, x' . \|x - x'\| < \delta \implies \|f(x) - f(x)'\| < K\|x - x'\|$$

Lipschitz continuity is key to robust predictions!

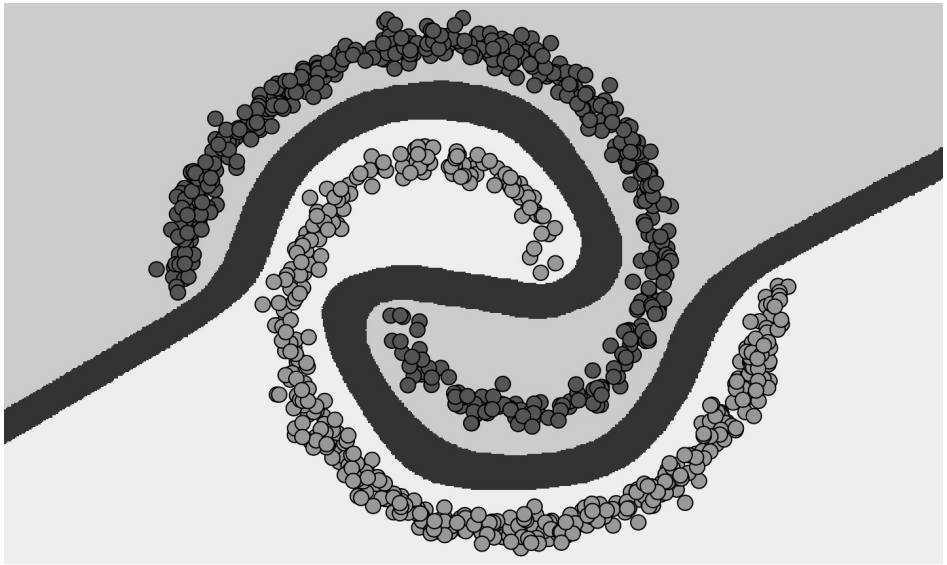
Building a provably-robust model



The Lipschitz constant tells us how much each class can change with a small delta on the input, in the worst case

Add a "⊥" class that reflects the adversary's best (most effective) perturbation

Training with ⊥ [Leino et al. 2021]



The “⊥” class lets us train the model using standard loss

Points classified as ⊥ incrementally corrected with each update

This leads to networks that:

- are sufficiently Lipschitz
- have boundaries with margins away from the manifold

Calculating the Lipschitz bound

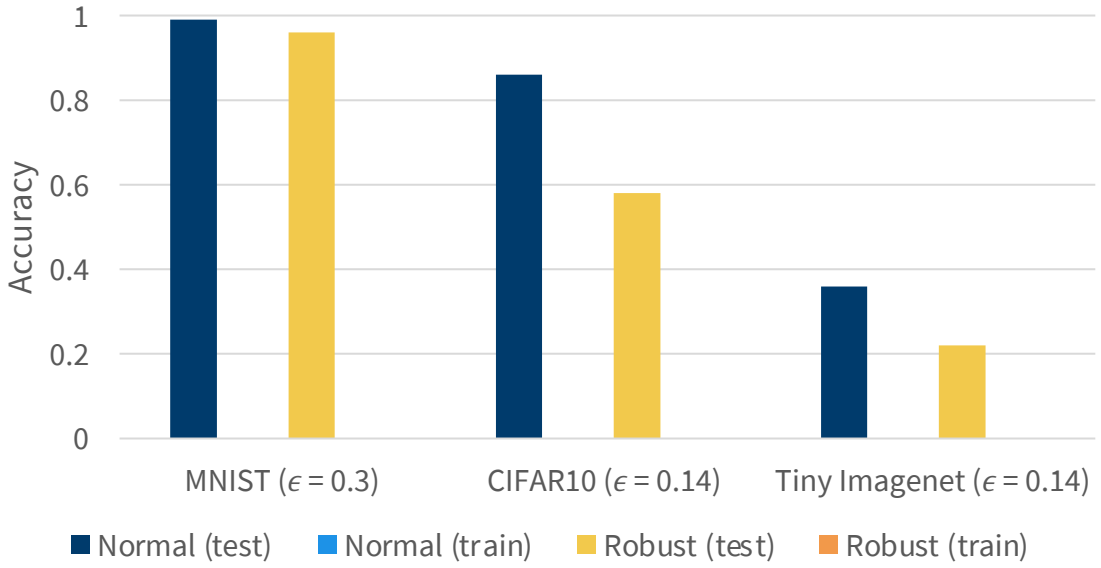
The **global** Lipschitz constant can be efficiently bounded by taking a product of the spectral norm of each layer

Any layer with a bounded Lipschitz constant can be used

Little training overhead, compatible with normal optimization

Final Lipschitz constant computed **once** after training—no cost at test time!

Robust training results



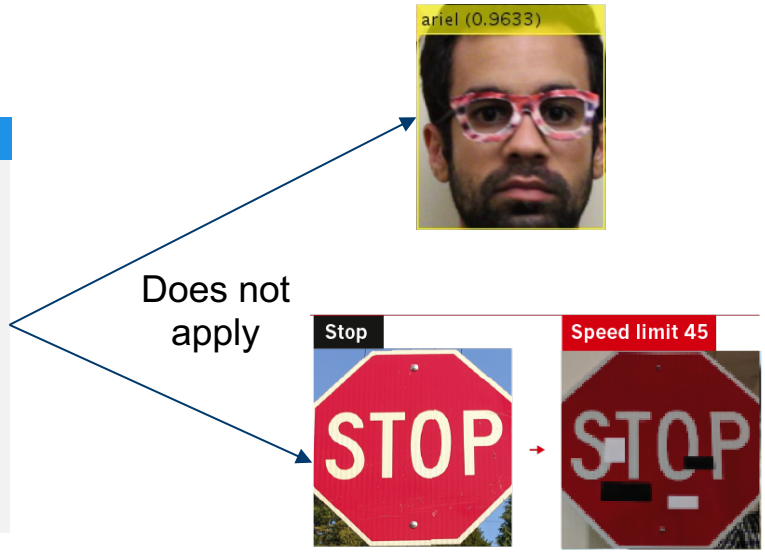
Key challenge: improve robust generalization

Beyond L_p-robustness

Definition

A classifier, F, is ϵ-*locally-robust* at x if $\forall x'$,

$$\|x - x'\| \leq \epsilon \implies F(x) = F(x')$$

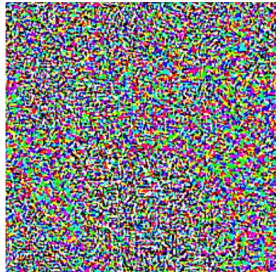


Key challenge: “semantic” robustness

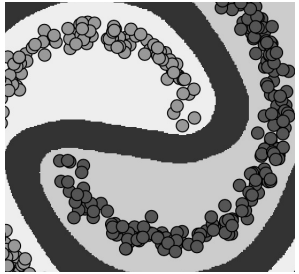
Takeaways

“gibbon”

Adversarial examples are ubiquitous in deep learning



They raise important questions about models’ *conceptual soundness* and *explainability*



Local robustness addresses this problem, and robust training is becoming more effective and scalable

Far from being solved, extending robust learning to “semantic” attacks is an outstanding challenge

Up next: how does robustness bear on explainability?

Q & A [15 min]

Break II

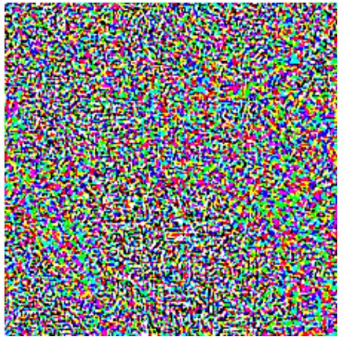
Section III

Connecting Explainability with Robustness

Recall: deep networks are easy to fool

“panda”

$+ .007 \times$

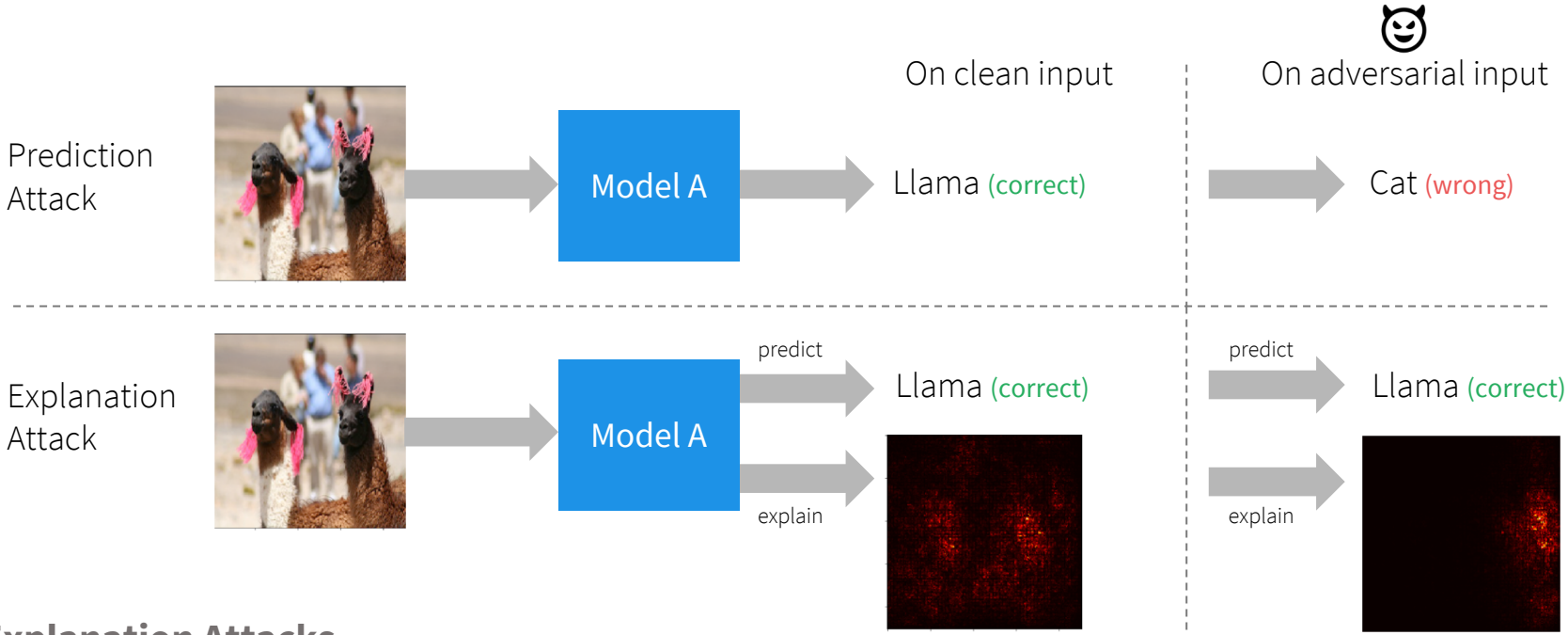


adversarial perturbation

$=$

“gibbon”

Explanations can also be manipulated adversarially



Explanation Attacks

*Ghorbani et al. AAAI '19**
Dombrowski et al. NeurIPS '19
Wang et al. NeurIPS '20

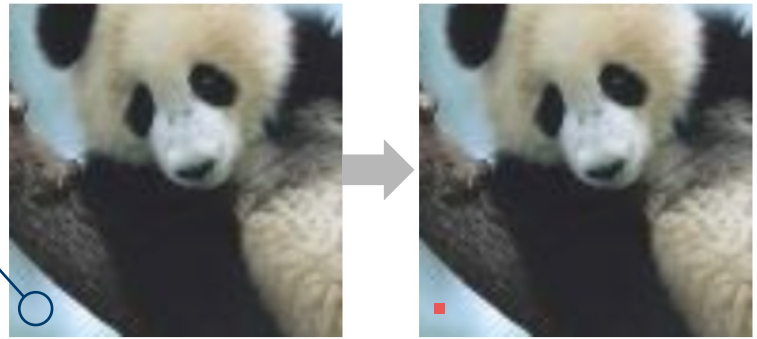
attribution map changes significantly

Can we trust explanations?

- If explanations can be manipulated, can we trust them?
- Is there something wrong with the explanation method that produces these anomalies?

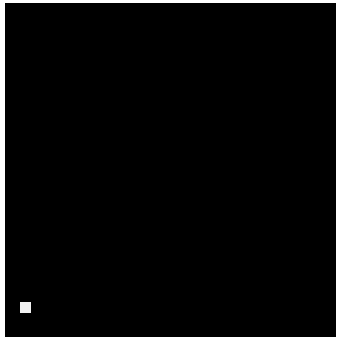
Can we trust explanations?

suppose that changing just one pixel in this region prevents the model from predicting “panda”



“panda”

not “panda”



possible explanation

Is it really wrong to assign influence to the pixel that can be modified to change the model’s prediction?

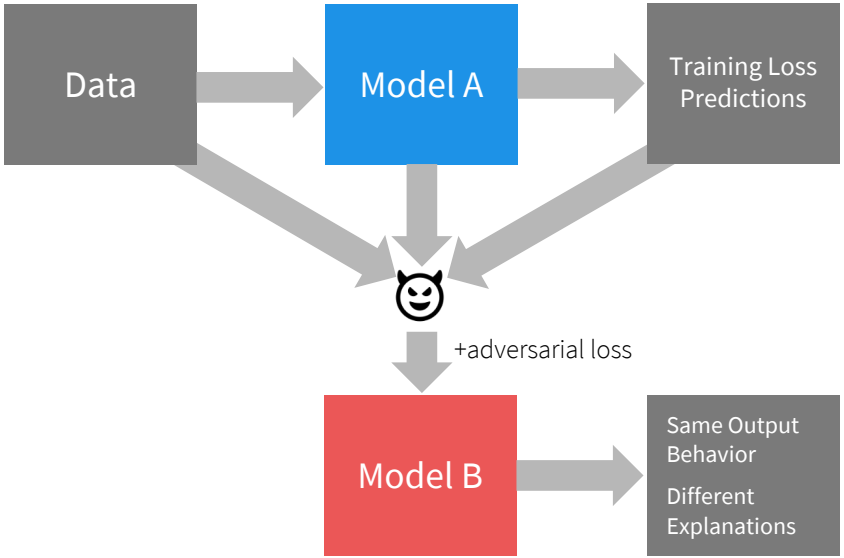
If it weren’t for this pixel, this point would not be classified as “panda”

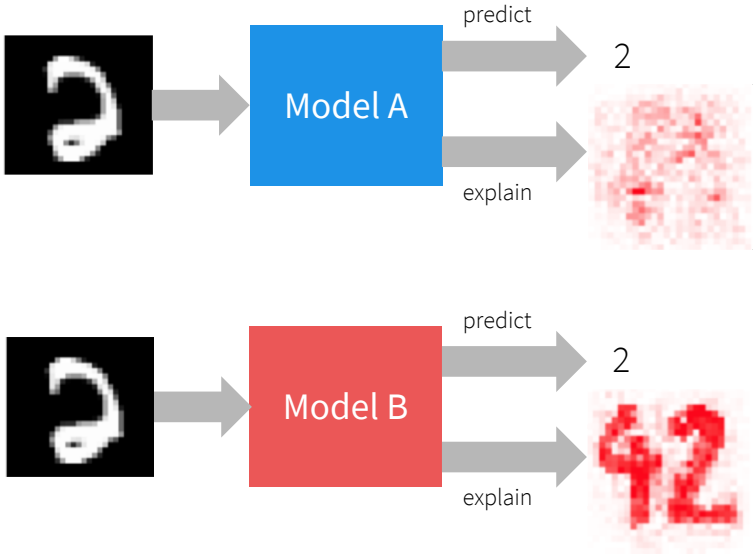
Proposition

Key Idea

“bugs” in *accurate* explanations are evidence of model quality issues

Model-based attacks on explanations



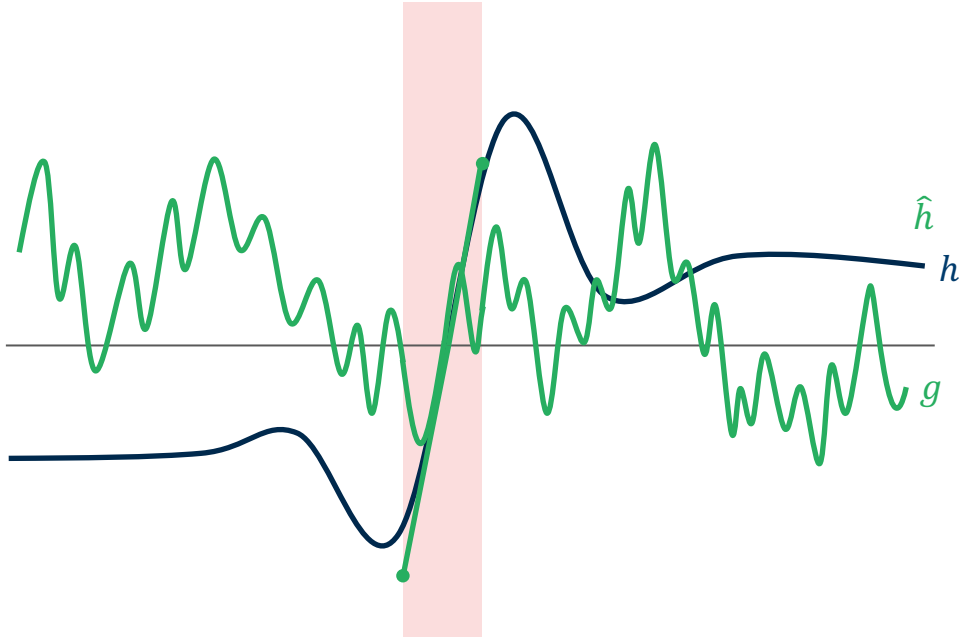


Model-based Explanation Attacks

Anders et al. 2020

Same predictions, different gradients

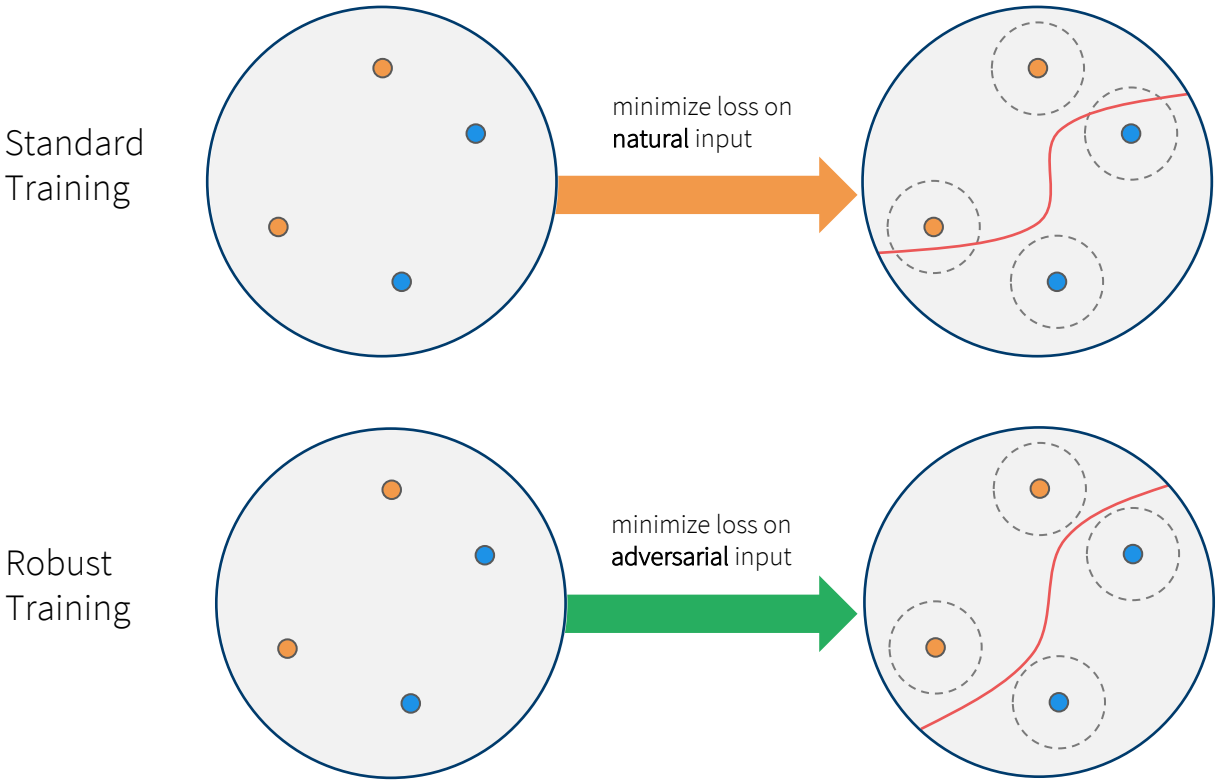
Theorem (Black et al. 2021)
Let h be a binary classifier and let g be an unrelated bounded function with arbitrary gradients. Then there exists some classifier $\hat{h}$ such that (1) $\hat{h}$ makes the same predictions as h, and (2) $\hat{h}$ has the same gradients as g almost everywhere.



Now what?

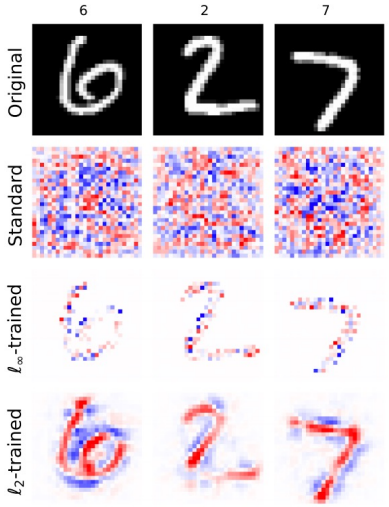
- **Key Idea**: “bugs” in accurate explanations are evidence of model quality issues
- On well-behaved models, we shouldn’t see these anomalies
- How do we improve model quality?

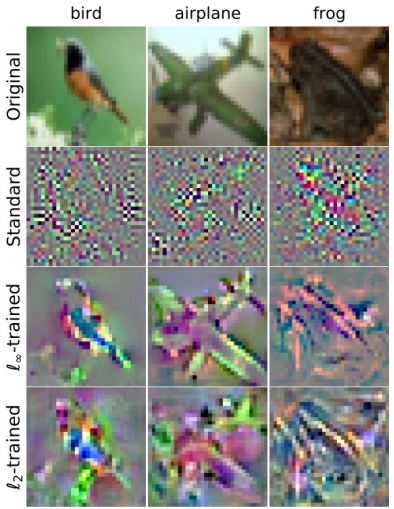
Recall: obtaining robust models

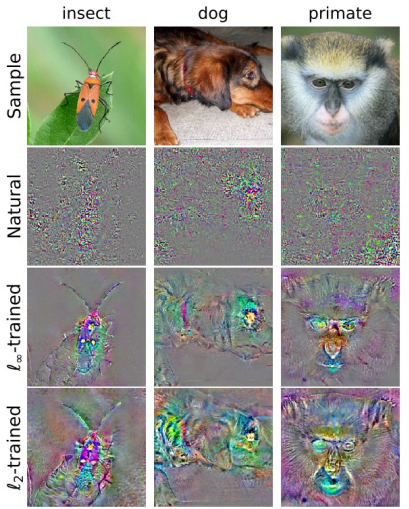


Robust models are more explainable

- Input gradients on robust models better align with the salient objects







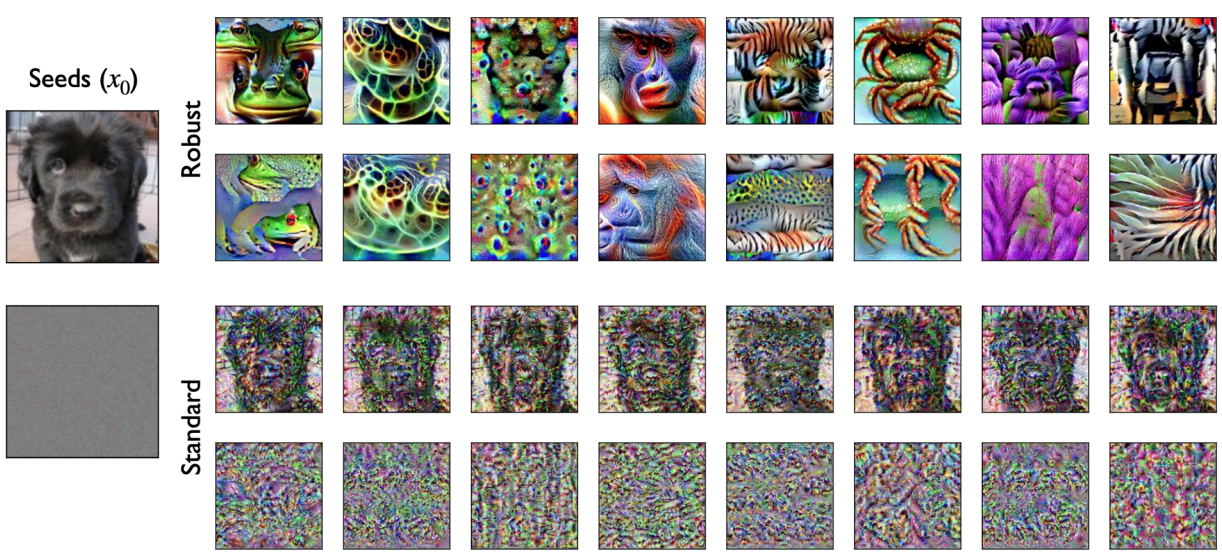
Explanations on Robust Models

*Tsipras et al. ICLR 2019**

Etmann et al. ICML 2019

Robust models are more explainable

- Feature visualization on robust models yields more recognizable results



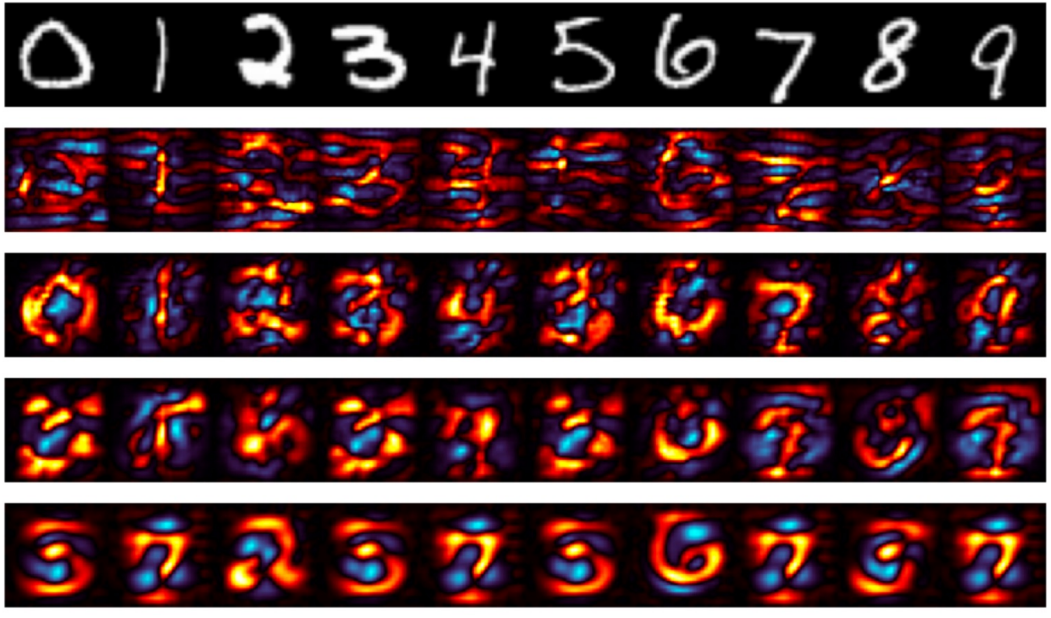
Feature Visualization

For classifier, f, and class , c, find δ that maximizes $f_c(x_0 + \delta)$

Visualizations on Robust Models

Tsipras et al. ICLR 2019

Explanations for Provably Robust Models

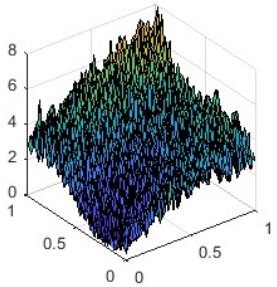


Saliency Map

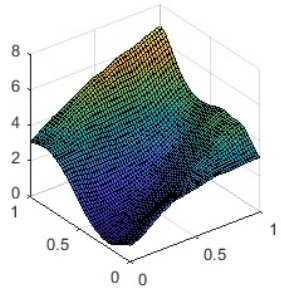
For classifier, f, input, x, and class , c, compute $\partial f_c(x) / \partial x$

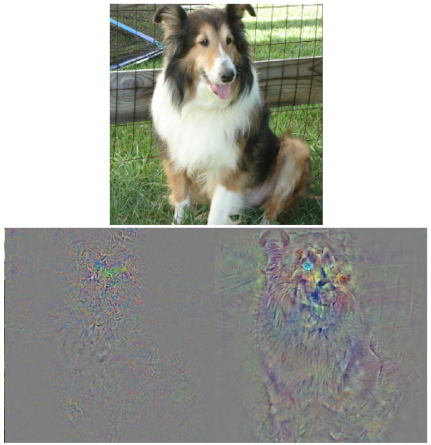
Explanations on Robust Models are More Robust

Non-robust Model



Robust Model





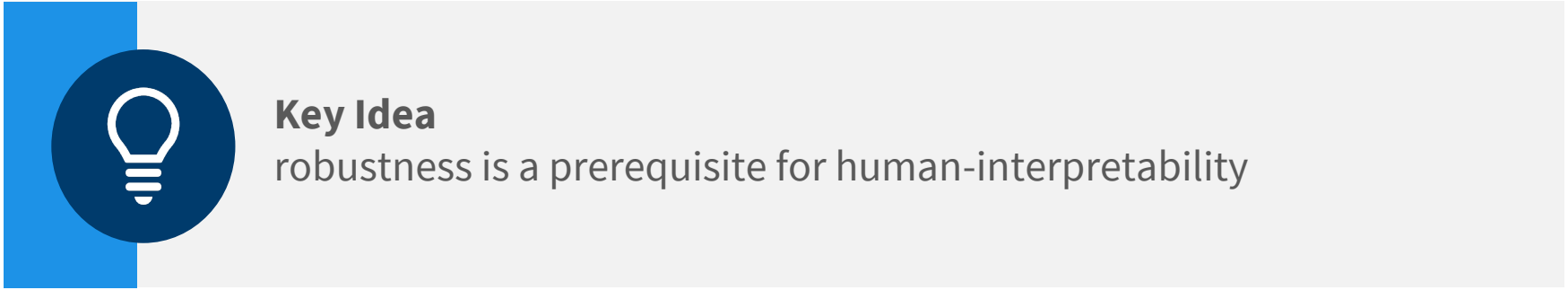
Non-robust Robust

[Etmann et al., ICML'19]

Theorem (Wang et al. NeurIPS 20')

If *f* is *(K, δ)* locally-Lipschitz, then $\nabla_x f$ is *(O(K), δ)* locally-Lipschitz

Proposition



Key Idea

robustness is a prerequisite for human-interpretability

Why are robust models more interpretable?

Feature-based perspective
adversarial examples are the result of a model using non-robust features, which are inherently not human-interpretable

Geometric perspective
the geometry of robust models leads to gradients that better convey information about the decision boundaries between classes

Why are robust models more explainable?

A feature-based perspective

Hypothesis *(Ilyas et al. ICLR 2019)*
standard-trained models use *non-robust features* that are nonetheless predictive on the data distribution

example of non-robust features contained in an instance labeled “frog”

non-robust features only

Non-robust Features
Ilyas et al. 2019

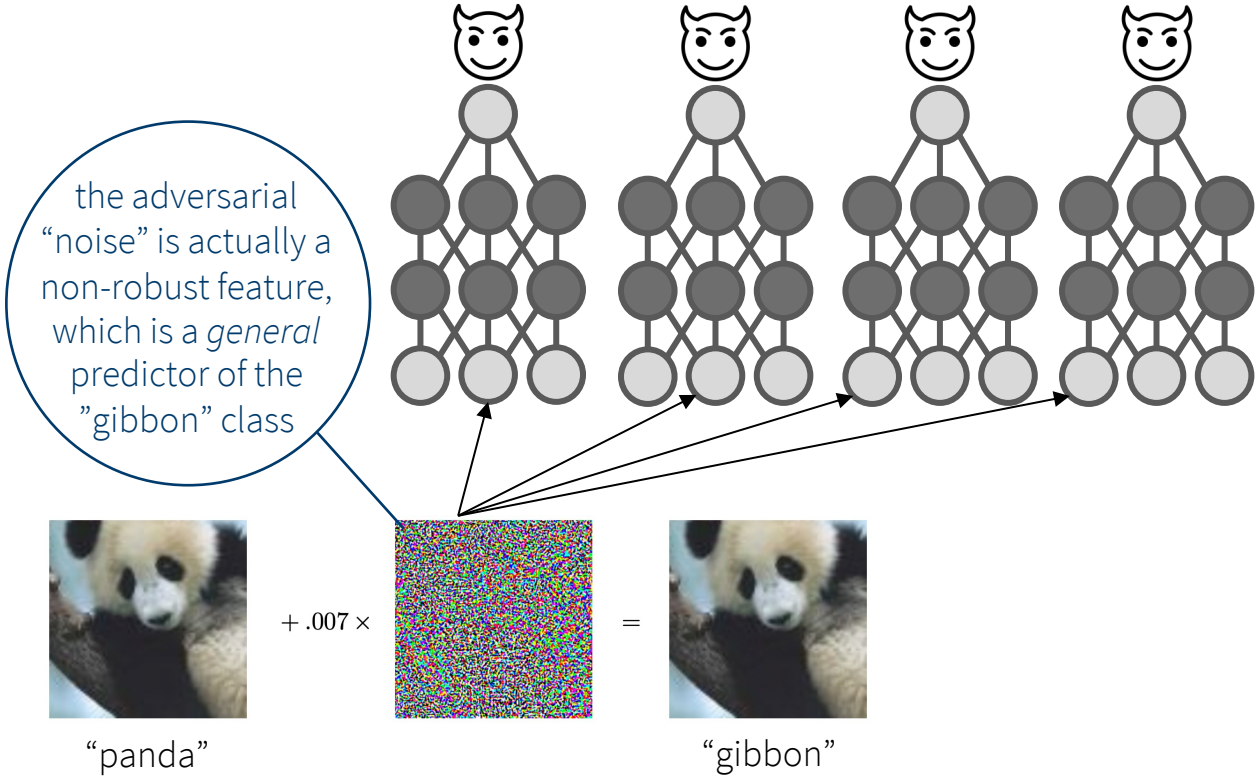
Non-robust features

Definition
A *feature* is a neuron in a neural network,
which is a function, $f: \mathbb{R}^n \rightarrow \mathbb{R}$

Definition
A feature is *non-robust* on data point, (x, y), if $f(x)$ correlates with y,
but $f(x + \delta)$ does not correlate with y for $\left|\left|\delta\right|\right| \leq \epsilon$

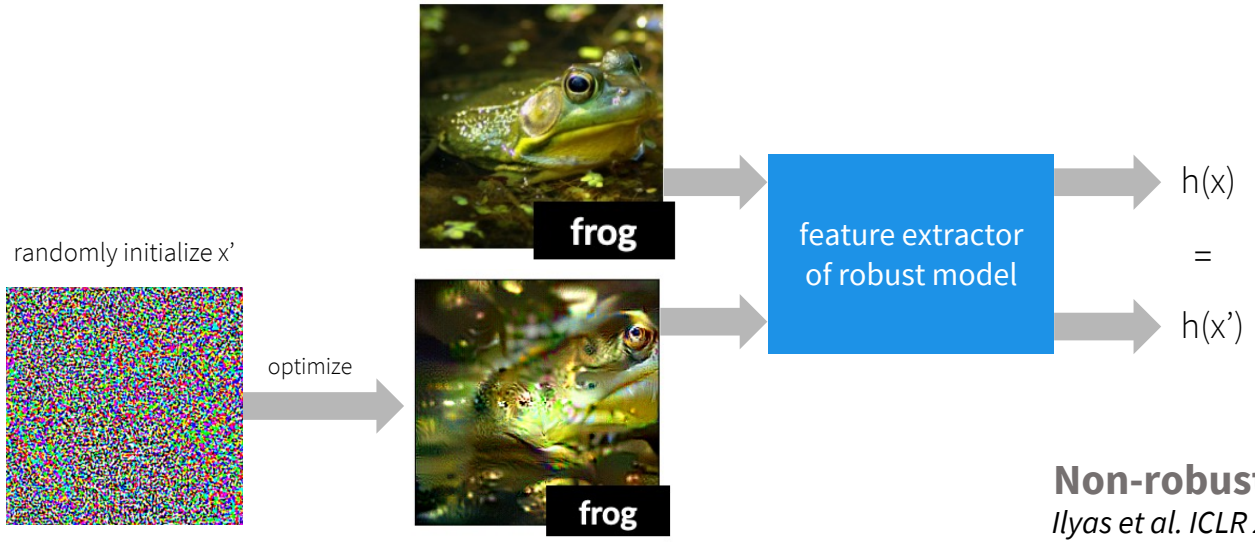
Non-robust Features
Ilyas et al. 2019

Recall: adversarial examples generalize



Isolating robust features

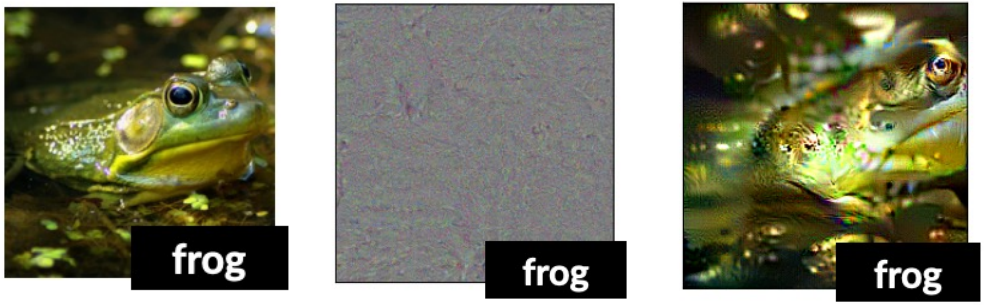
Non-robust features are not useful for a robust objective, thus we do not expect robust models to learn them (i.e., robust models should only learn robust features)



Non-robust Features
Ilyas et al. ICLR 2019

Why are robust models more explainable?

- Standard-trained models use *non-robust features* that are nonetheless predictive
- Non-robust features are not useful for a robust objective, thus we do not expect robust models to learn them
- Non-robust features are inherently less interpretable



Non-robust Features
Ilyas et al. ICLR 2019

Why are robust models more explainable?

Feature-based perspective
adversarial examples are the result of a model using non-robust features, which are inherently not human-interpretable

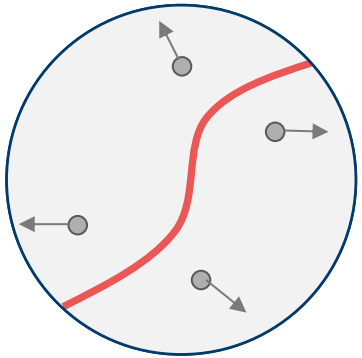
Geometric perspective
the geometry of robust models leads to gradients that better convey information about the decision boundaries between classes

Why are robust models more explainable?

A geometric perspective

Explanations usually focus on the gradient on the target inputs

Is the input gradient sufficient to explain the classification?



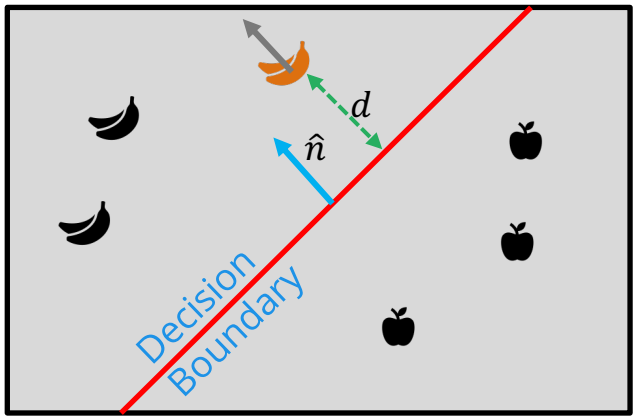
Input gradients may not accurately describe the decision boundary.

Theorem (Informal, Wang et al. 2021)

Input gradients of a model with smaller Lipschitz Constants of gradients better capture the information of decision boundary

How to Capture Decision Boundary ?

Linear Classifier

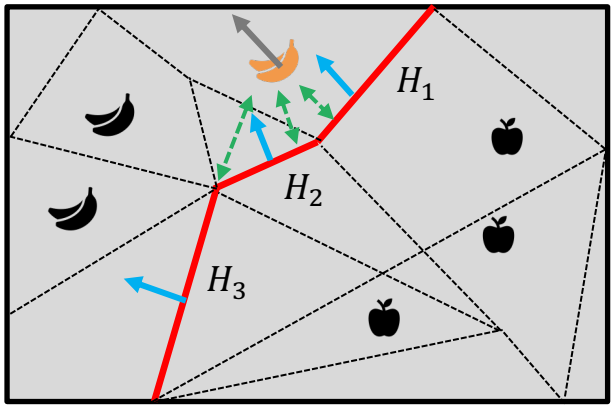


🍌	Why 🍌 is classified as *Banana* instead of *Apple*?	
$\hat{n}$	Normal Vector $\hat{n}$ explains features importance	☑ Found by axiomatic explanations
d	Greater projection distance corresponds to high confidence	☑ Calculated by projection

The input gradient has the same direction as the normal vector

Gradient Explanations *May Not* Capture Decision Boundary

Non-linear Classifier



ReLU networks

Why 🍌 is classified as *Banana* instead of *Apple*?

$\hat{n}$ Which normal vector should be the explanation ?

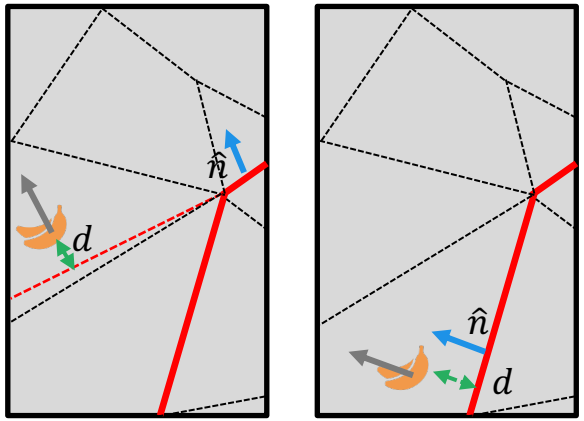
d Which decision should be the confidence?

Closest / Nearby boundaries

Which boundary is the input gradient ↖ explaining?

Computing Normal Vectors of Nearby Boundary

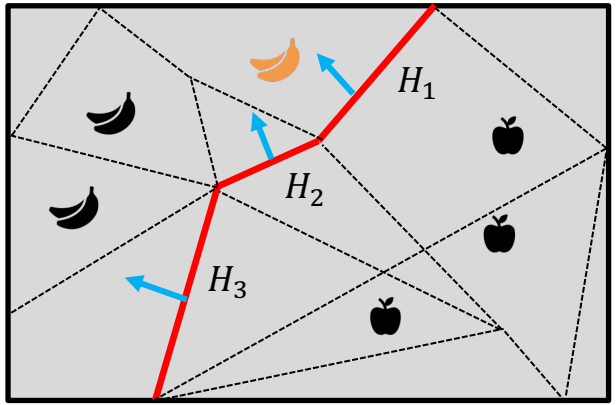
Issue 1



$\hat{n}$ Gradient on the input may NOT be the $\hat{n}$ from closest boundary

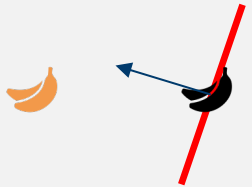
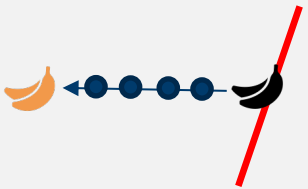
d Projection distance may NOT be the actual minimal distance

Issue 2



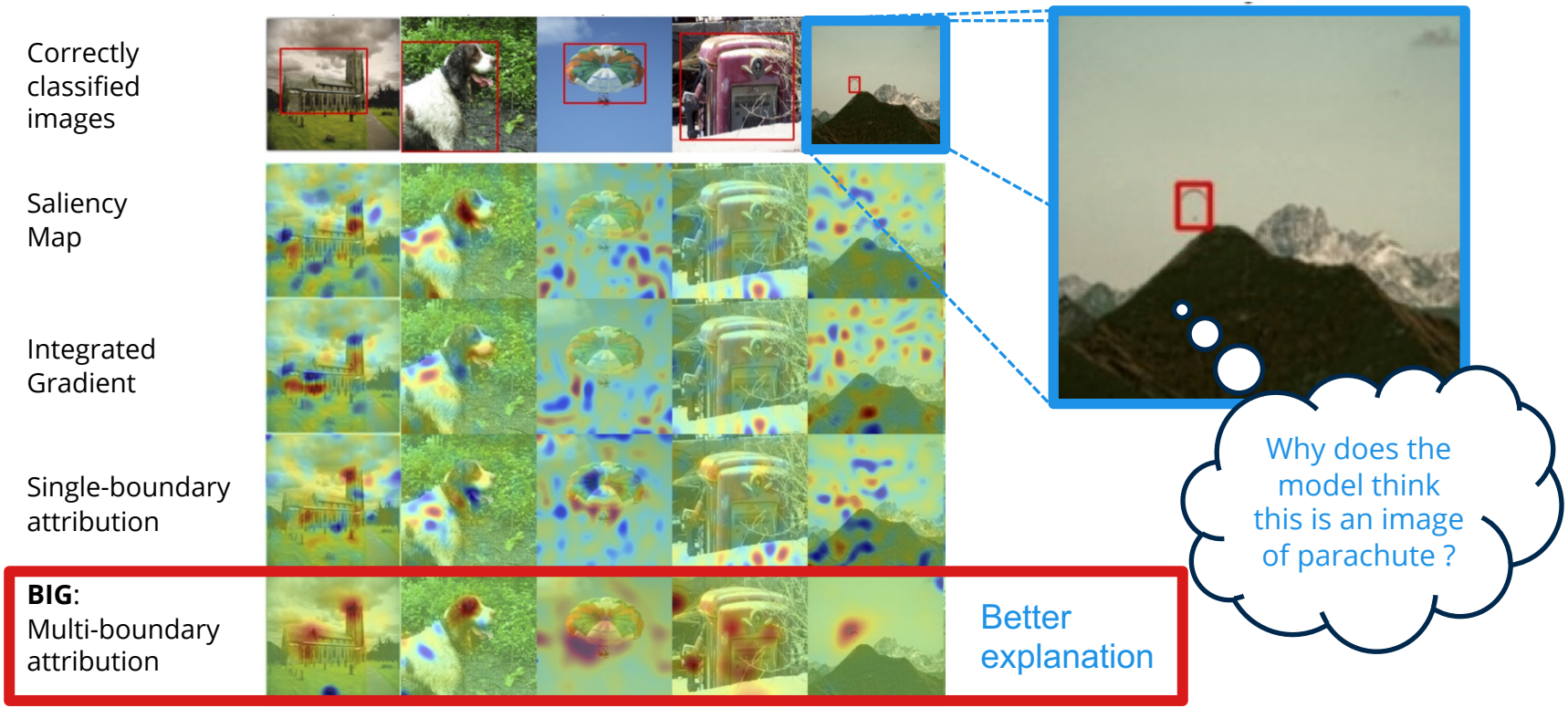
$\hat{n}$ One normal vector is NOT enough to capture the model's behavior

Boundary Attributions Provide Normal (Vector) Explanations

1	Find the closest decision boundary	• Projection is not reliable. • Exact solution is NP-hard	**Approximation**: the closest adversarial example stands on the closest decision boundary
2	Calculate the normal vector	 $\hat{n} = \frac{\partial f(\quad)}{\partial \quad}$	Gradient on the closest adversarial example is the normal vector
3	Boundary-based Integrated Gradient (BIG)	 $\int \frac{\partial f(x)}{\partial x} \mathrm{dx}$	

Boundary Explanations
Zifan, Fredrikson, Datta, '21

Boundary Attributions Provide Normal (Vector) Explanations



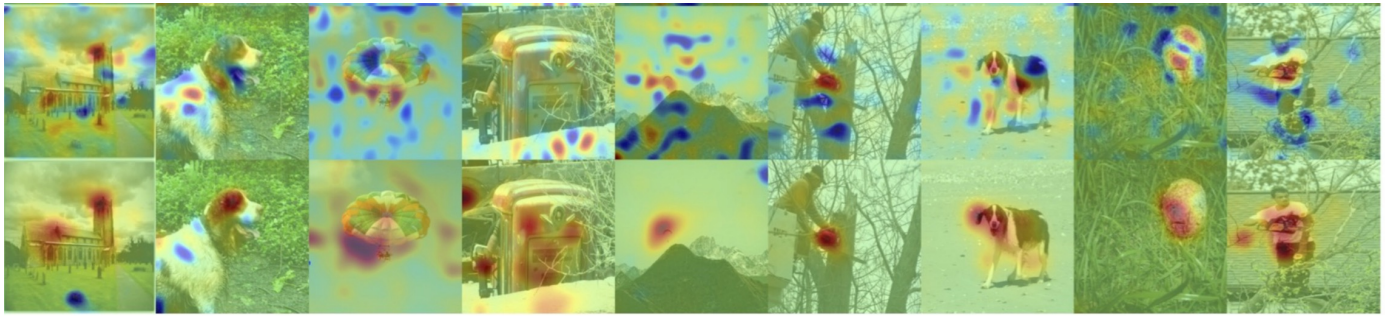
Boundary Explanations
Zifan, Fredrikson, Datta, '21

Robust Model Are More Explainable

because gradient explanations better mimic boundary explanations

Gradient Explanation (IG)

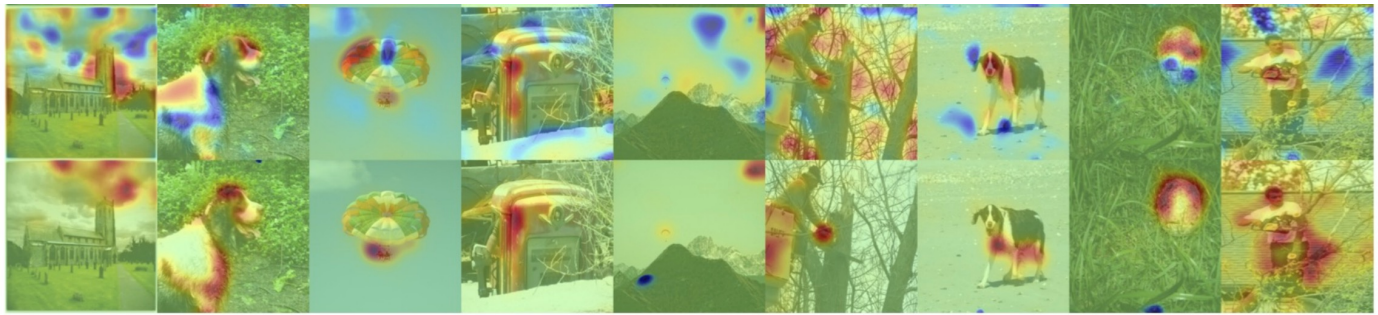
Boundary Explanations (BIG)



Non-Robust Model

Gradient Explanation (IG)

Boundary Explanations (BIG)



Robust Model

Boundary Explanations
Zifan, Fredrikson, Datta, ‘21

Demo Boundary Attribution

with TruLens

Summary

“Bugs” in accurate explanations are evidence of model quality issues

Quality explanations require quality models

Robustness may be one way to achieve better model quality

Q & A

Tutorial | SIGKDD 2021

Machine Learning *Explainability* and *Robustness*

➡ Connected at the Hip

Anupam Datta
Matt Fredrikson
Klas Leino
Kaiji Lu
Shayak Sen
Zifan Wang

We appreciate your participation in this tutorial.

For More Resources:

- Tutorial Website
- Accountable Systems Lab
- TruLens and Demos
- Truera's Blog Posts on Explanability

Contact Us: shayak@truera.com, zifan@cmu.edu