

# Interpretability

Trustworthy Machine Learning - November 30, 2021

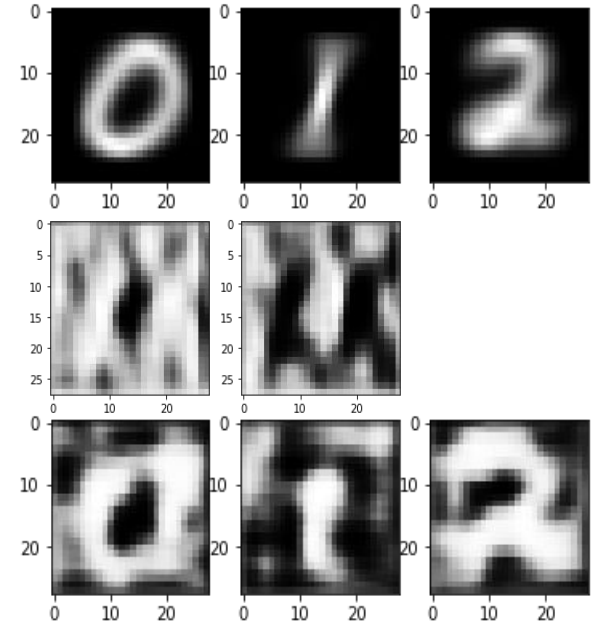
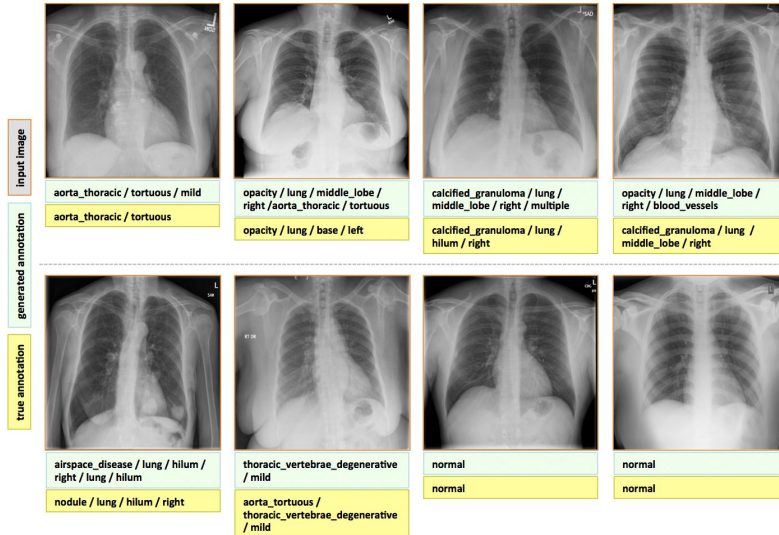
Hongyu (Charlie) Chen, Yijie (EJay) Guo, Caroline Malin-Mayor

# What is interpretability?

- A *human* understanding of the model
  - Understand how the **model** works towards the **task**
  - Note: Interpretability is **NOT** about how the **world** work
- There are many types of understanding
  - How certain features of input influence a prediction
  - Prototypical examples and references of each class
  - Convert internal representations to understandable concepts
  - Reduce model to a small set of rules or a simple equation

# Why is interpretability important?

- Improve the model
  - Create innovative techniques to solve model structural problem
  - Remove undesired decision-making logic
- Find hidden pattern from data



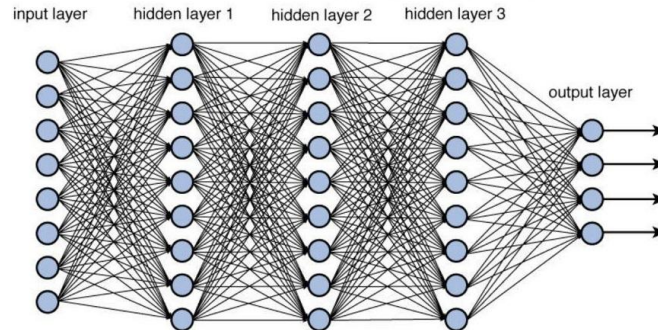
# Why is interpretability important?

- Verify and Justify the model decision accordingly to domains
  - Justify the decision-making process.
  - Legislation.
- Attack/Game the model
  - Fool the self-driving car and cause an accident.

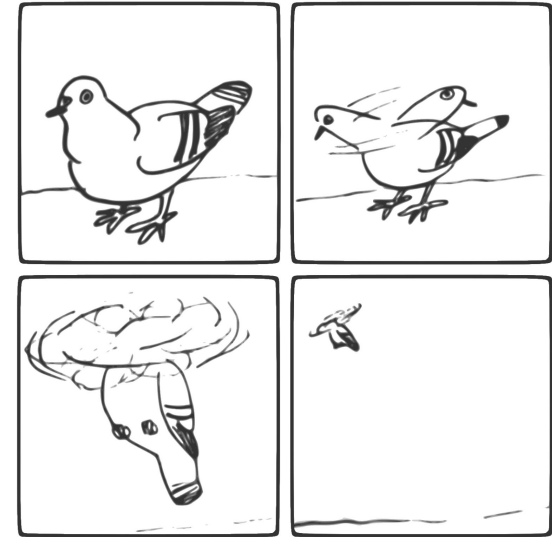


# Why is difficult to interpret decision-making logic?

- Modern ML architectures are too complicated for human.
- People only asks for reasoning when they get **undesired** results. What if?
  - The problem is rare and the cause is hidden
  - The source of the problem is from the data.
- No structured interpretation framework has been invented.



When your program  
is a complete mess,  
but it does its job



# Today's Topics

- Interpretability
  - Explanation vs Interpretation
  - Challenges of Interpretable Models
- Deep Neural Network
  - Model Architecture
  - Model Size
  - Regularization
- Influence Functions on Black-box Models
  - Impacts of Each Data Point Towards Prediction

# Stop Explaining Black Box Machine Learning Models for **High Stakes Decisions** and Use Interpretable Models Instead

Cynthia Rudin  
Duke University  
[cynthia@cs.duke.edu](mailto:cynthia@cs.duke.edu)  
2019

# Terminology

## Interpretable Model

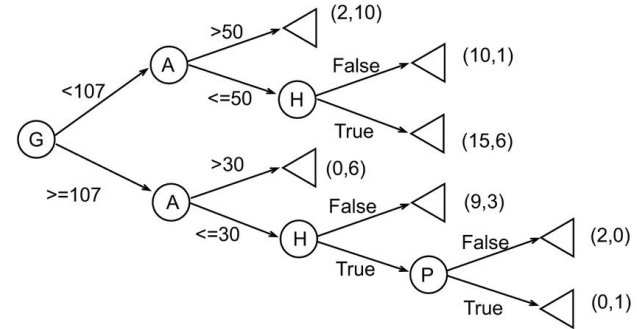
Constraining its **form** specifically for personal or structured domain.

## Explainable Model

Explaining how a **model** works towards tasks.

## Black-box Model

- A function that is too complicated for any human to comprehend.
- A proprietary.

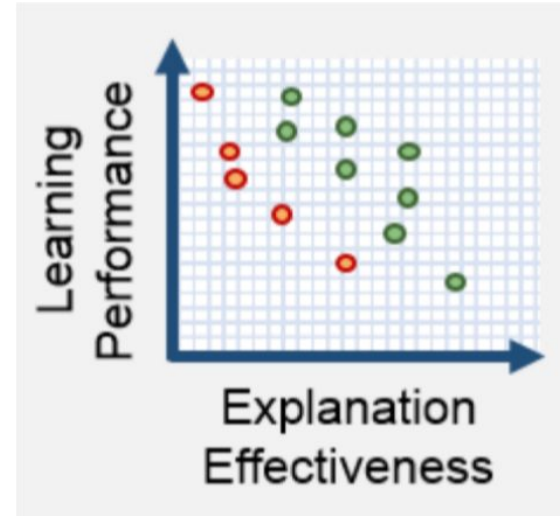


Decision Tree



# Properties of Models

- Complexity of models  $\neq$  Accuracy
- Trade-off: Performance vs Utilities?  
Accuracy vs Interpretability  
Generalization Explainability  
Transparency



**Fictional** Depiction of Accuracy-Interpretability Trade-off

DARPA XAI (Explainable Artificial Intelligence Broad Agency)

# Key Issues with Explainable (Black-box) ML

# Key Issues with Explainable ML

- If the black-box ML model is explainable, why do we need those models in the first place?
  - One model could have multiple explanations.
  - Explanations of black-box models are very likely inaccurate.

One explanation method could explain 90% of the model accurately, which also means 10% is still unknown. This is not explanation but an approximation.

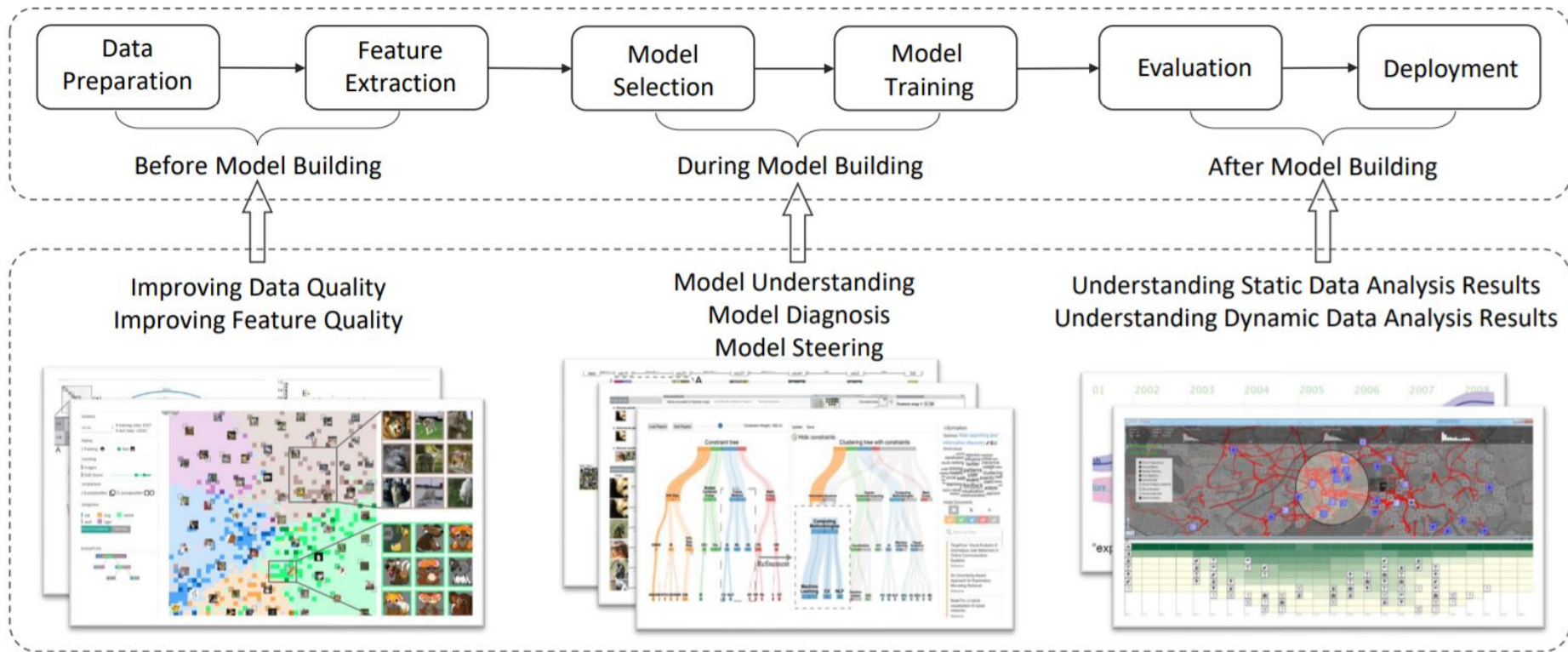
# Key Issues with Explainable ML

## The current ML and DL community

- Many researchers are trained in deep learning, not in interpretable machine learning.
  - With limited time, resource and manpower, researchers could only focus on certain topics.
- Recent works focus more on explaining black boxes instead of interpretable models.
  - Because black boxes are the majority models.
- Not enough toolkits provide UI for explaining ML methods.

# A Survey of Visual Analytics Techniques for Machine Learning

## Machine Learning Pipeline



An overview of visual analytics research for machine learning by Yuan et al., (2020)

# Key Issues with Explainable ML

- Explain the decision-making process instead of explaining causality.

ProPublica created a linear explanation model for COMPAS that depended on race, and then accused the black box COMPAS model of depending on race, conditioned on age and criminal history.

$$f(x|age, criminal\ history) \rightarrow f(x, black|age, criminal\ history) \rightarrow arrest$$

Correctional Offender Management Profiling for Alternative Sanctions (COMPAS)

# Key Issues with Explainable ML

- Current explanation methods do not express enough for black-box models


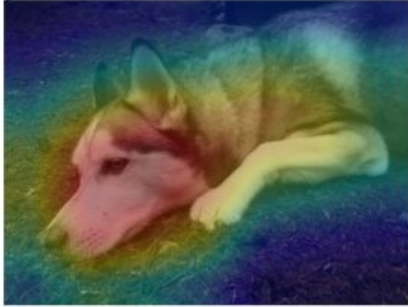

	Test Image	Evidence for Animal Being a Siberian Husky	Evidence for Animal Being a Transverse Flute
Explanations Using Attention Maps			

Figure credit: Chaofan Chen and [28]

# Key Issues with Explainable ML

- Human errors could contribute to unknown consequences inside the black-box models, yet can be very hard to detect.

Incorrect information in the training data, like typo, misclassification.

- Outside boundaries, i.e. outside training distribution, black-box models become even more unpredictable.



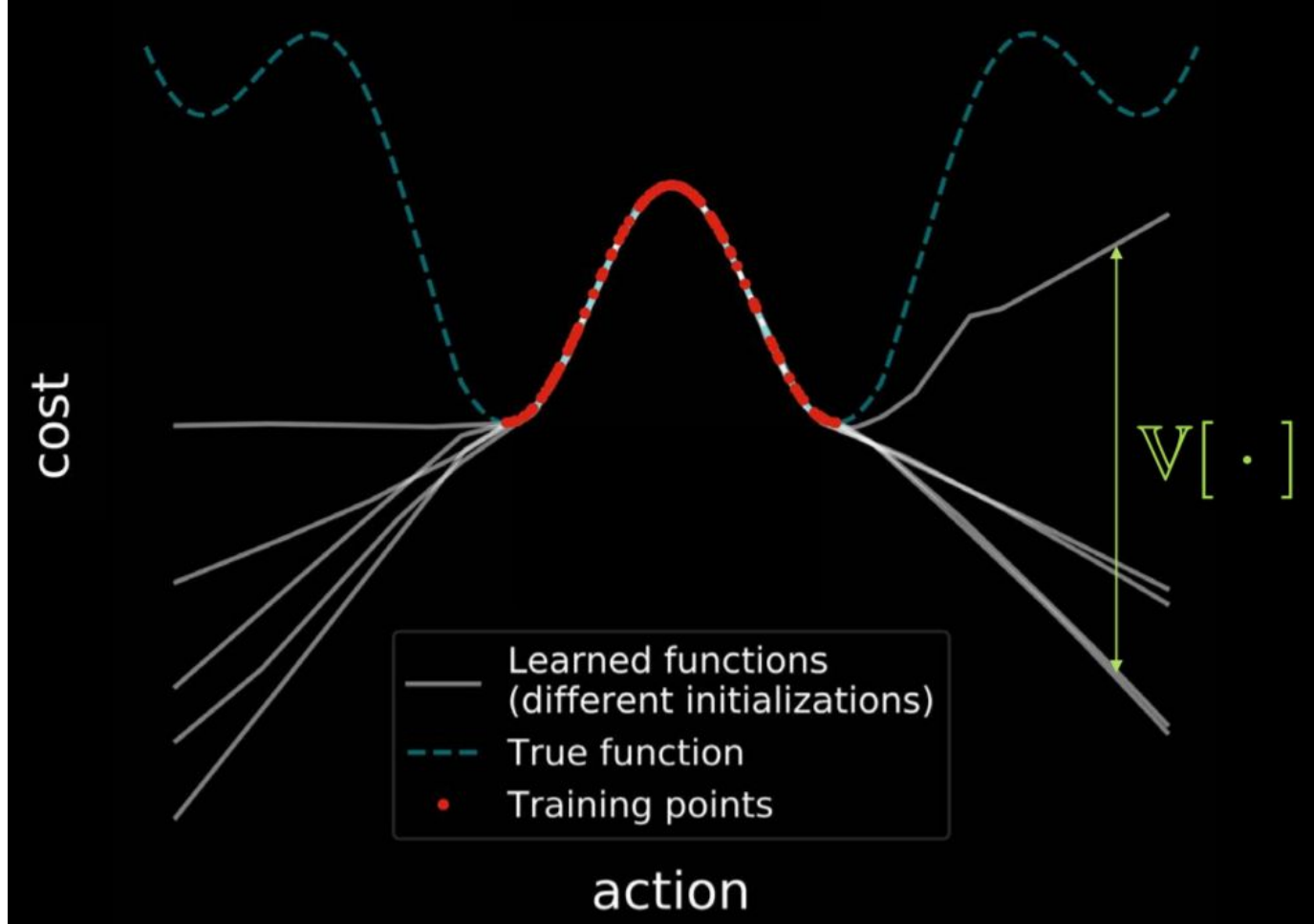


Figure credit: Alfredo Canziani @ NYU

# Key Issues with Interpretable ML

# Key Issues with Interpretable ML

- Interpretable models are basically transparent to everyone.
- Black-box models are not transparent and hard to be
  - gamed (attacked)
  - reverse-engineered.
- Intellectual Property => ML is expensive.
  - Amazon's Mechanical Turk: \$70k for a 100k samples dataset.
  - 2 employees x \$5k + 3 freelancers x \$3k = \$19k per month.
  - Computation cost, production cost, etc.
- Selling black-box models or services for profits.

# ML Model of the Certifiably Optimal Rule Lists (CORELS)

IF age between 18-20 and sex is male THEN predict arrest (within 2 years)  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict arrest  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest.

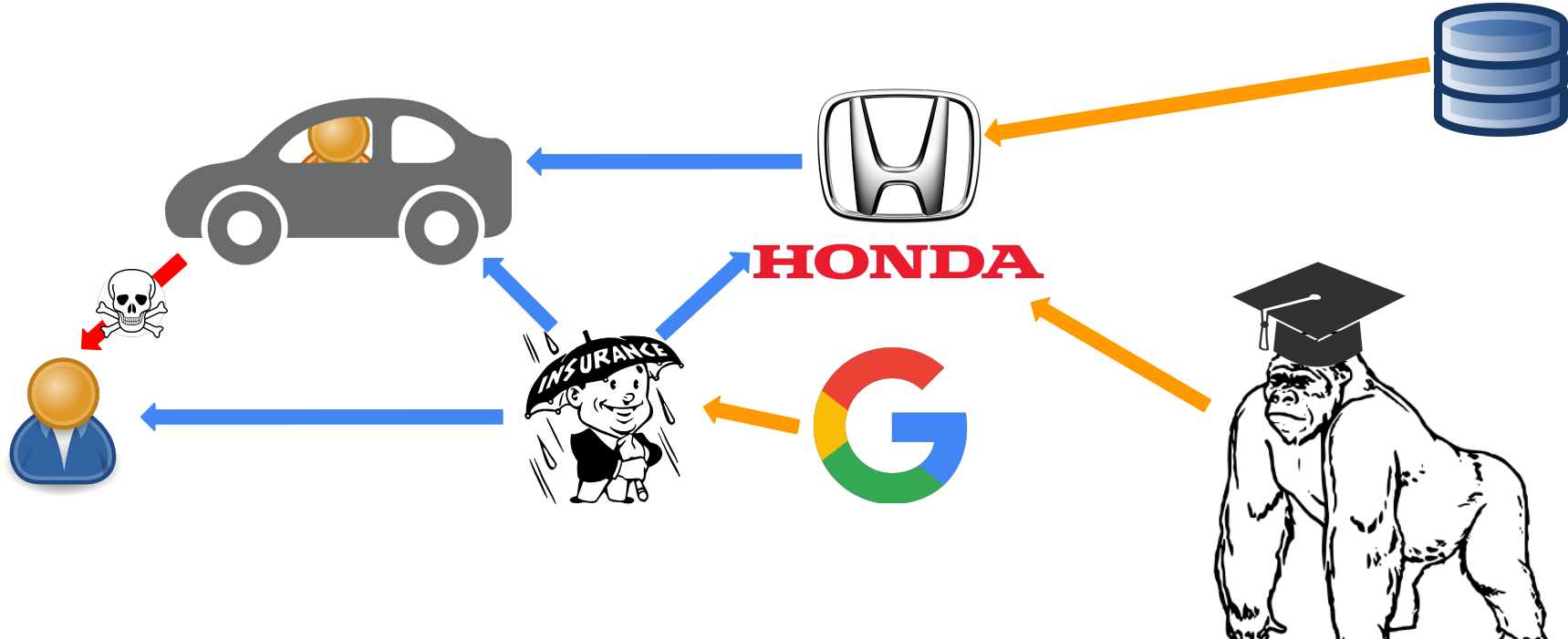
COMPAS	CORELS
black box 130+ factors might include socio-economic info expensive (software license), within software used in U.S. Justice System	full model is in Figure 3 only age, priors, (optional) gender no other information free, transparent

# Key Issues with Interpretable ML

- People could abuse the transparent decision-making process.
- Counterfactual **explanations** / Inverse Classification
  - Minimal changes in features could influence outcomes.
- Counterfactual **explanations** of black-box models are sufficient.
  - If you have less debt, then we could approve for this loan.
  - If you have a job with more than \$500 weekly payment, then we could approve for this loan.
  - ...

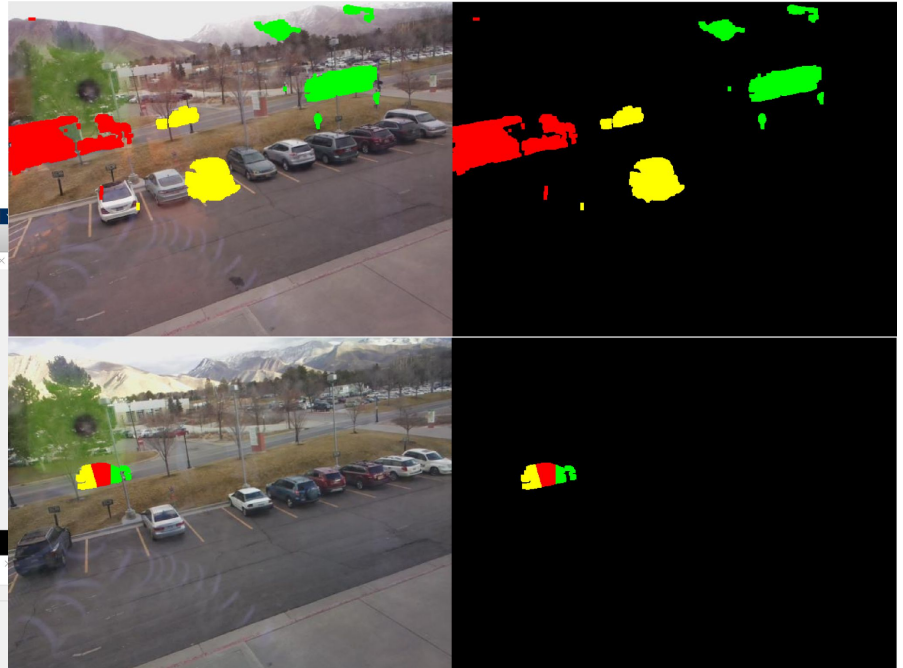
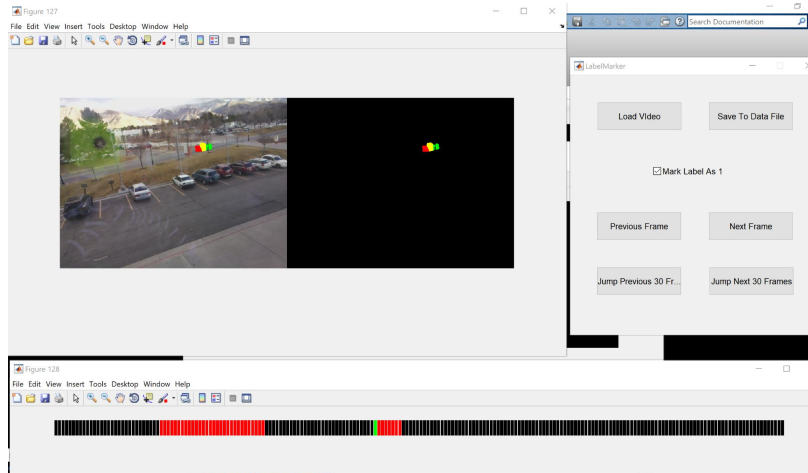
# Key Issues with Interpretable ML

- Conflict of interests and responsibilities in machine-learning-as-a-service (MLASS)



# Key Issues with Interpretable ML

- Computation and domain expertise of application-specific knowledge are required to design and create interpretable models.
- Black box might find hidden patterns.



# Algorithmic Challenges in Interpretable ML



# Algorithmic Challenges in Interpretable ML

Convert machine learning models to human-designed interpretable models

Three Challenges:

- Construct optimal logic models
- Construct optimal sparse scoring systems
- Define interpretability for specific domains and create methods accordingly

# Algorithmic Challenges in Interpretable ML

**Challenge #1: Construct optimal logic models/algorithms, which solve logical modeling problems given practical datasets within reasonable time.**

Logic models are constructed through logic operators, like “or”, “and”, “if-then”, etc.

Classic AI: Knowledge Representation and Reasoning

- Nature Language  
First Order Language  
 $A \wedge B, A \vee B, A \rightarrow B \equiv \neg A \vee B$
- Knowledge base with  
conjunctions of disjunctions
- 3-SAT is NP-complete
- Solutions could be non-optimal

$$\begin{aligned}
 &(\bar{a} \vee m \vee u) \wedge (a \vee n \vee u) \wedge (\bar{a} \vee r \vee x) \wedge (\bar{c} \vee \bar{e} \vee s) \\
 &\wedge (c \vee \bar{m} \vee \bar{w}) \wedge (\bar{c} \vee p \vee x) \wedge (c \vee q \vee s) \wedge (e \vee p \vee s) \\
 &\wedge (e \vee q \vee \bar{y}) \wedge (e \vee r \vee y) \wedge (\bar{e} \vee r \vee z) \wedge (\bar{g} \vee r \vee x) \\
 &\wedge (g \vee v \vee \bar{y}) \wedge (m \vee \bar{n} \vee u) \wedge (m \vee \bar{o} \vee \bar{u}) \wedge (m \vee o \vee v) \\
 &\wedge (\bar{m} \vee \bar{q} \vee s) \wedge (\bar{m} \vee \bar{r} \vee \bar{s}) \wedge (m \vee \bar{u} \vee \bar{v}) \wedge (\bar{m} \vee x \vee \bar{z}) \\
 &\wedge (\bar{n} \vee r \vee \bar{y}) \wedge (o \vee r \vee \bar{w}) \wedge (\bar{p} \vee q \vee s) \wedge (r \vee \bar{w} \vee \bar{x}) \\
 &\wedge (r \vee w \vee \bar{y}) \wedge (r \vee w \vee \bar{z})
 \end{aligned}$$

# Algorithmic Challenges in Interpretable ML

**Challenge #1: Construct optimal logic models/algorithms, which solve logical modeling problems given practical datasets within reasonable time.**

Models are created manually yet are accurate, full-blown ML models.

Optimization Problem:

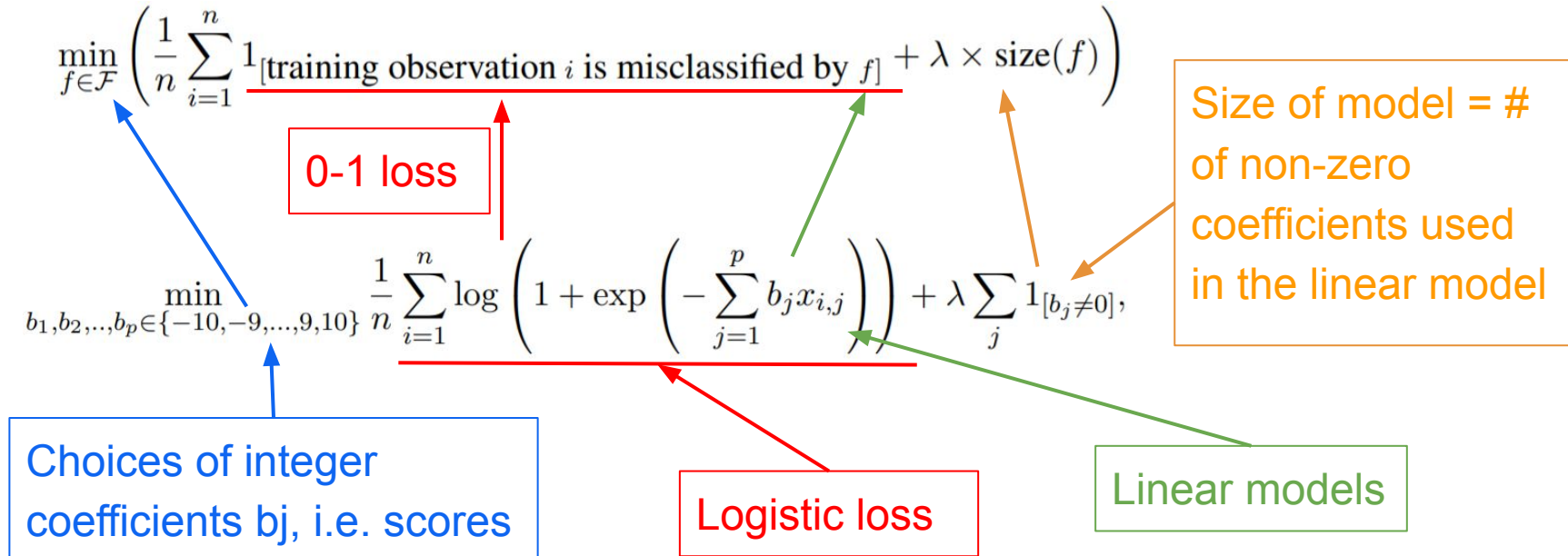
$$\min_{f \in \mathcal{F}} \left( \underbrace{\frac{1}{n} \sum_{i=1}^n 1_{[\text{training observation } i \text{ is misclassified by } f]}}_{\text{Training error}} + \underbrace{\lambda \times \text{size}(f)}_{\text{Trade-off between training accuracy and model size}} \right)$$

Diagram illustrating the optimization problem components:

- Family of logical models** points to the minimization variable  $f \in \mathcal{F}$ .
- Training error** points to the first term of the objective function:  $\frac{1}{n} \sum_{i=1}^n 1_{[\text{training observation } i \text{ is misclassified by } f]}$ .
- Trade-off between training accuracy and model size** points to the second term of the objective function:  $\lambda \times \text{size}(f)$ .

# Algorithmic Challenges in Interpretable ML

**Challenge #1: Construct optimal logic models/algorithms, which solve logical modeling problems given practical datasets within reasonable time.**



# Algorithmic Challenges in Interpretable ML

## Challenge #2: Construct optimal sparse scoring systems, which are computationally efficient.

The scoring system

- Look like a system created by human in the absence of data
- Can be find efficiently through optimization.

1.	Prior Arrests $\geq 2$	1 point	...
2.	Prior Arrests $\geq 5$	1 point	+ ...
3.	Prior Arrests for Local Ordinance	1 point	+ ...
4.	Age at Release between 18 to 24	1 point	+ ...
5.	Age at Release $\geq 40$	-1 points	+ ...
		<b>SCORE</b>	<b>= ...</b>

SCORE	-1	0	1	2	3	4
RISK	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Blood Test	Result	Normal Value
WBCs (billion/L)	8.00	3.5 to 10.5
Neutrophils (%)	62	40 to 70
Lymphocytes (%)	28	25 to 45
Monocytes (%)	10	2 to 8
Eosinophils (%)	1	1 to 5
Basophils (%)	0	0 to 1
RBCs (trillion/L)	3.84	4.3 to 5.7
Hb (g/dL)	11.7	13 to 17
Hematocrit (%)	37	37 to 52
Platelets (billion/L)	262	150 to 450

# Algorithmic Challenges in Interpretable ML

**Challenge #3: Define domain specific interpretability and create methods accordingly, including computer vision.**

Even using latent representations in neural networks, interpretable versions exist with comparable accuracy towards black box models.

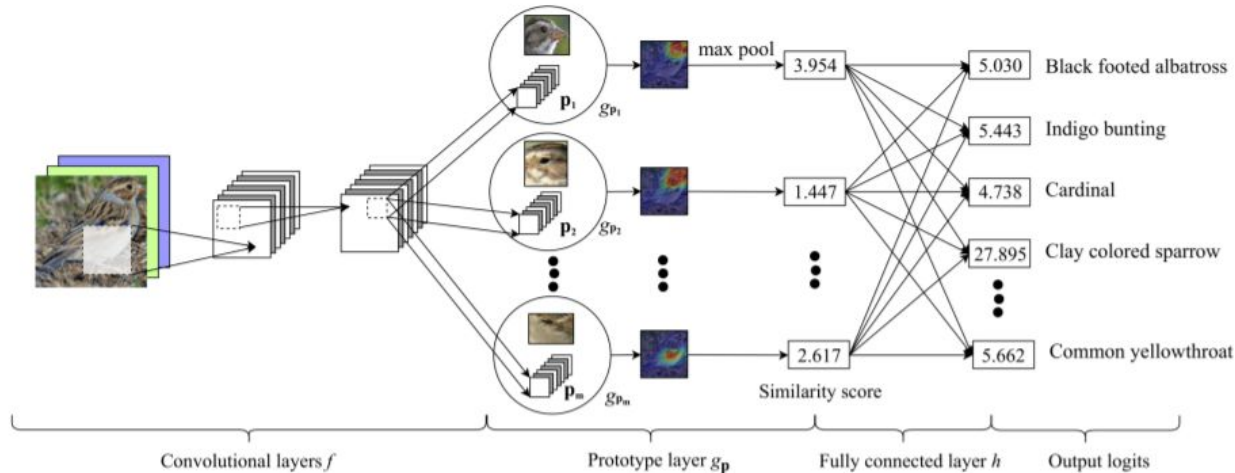
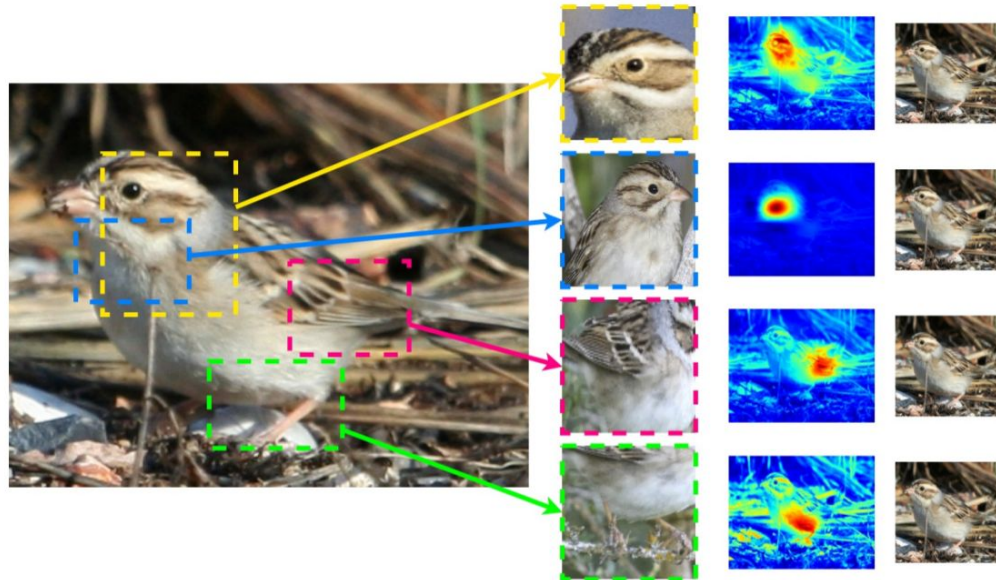


Figure credit: Chaofan et al., (2019) [49]

# Algorithmic Challenges in Interpretable ML

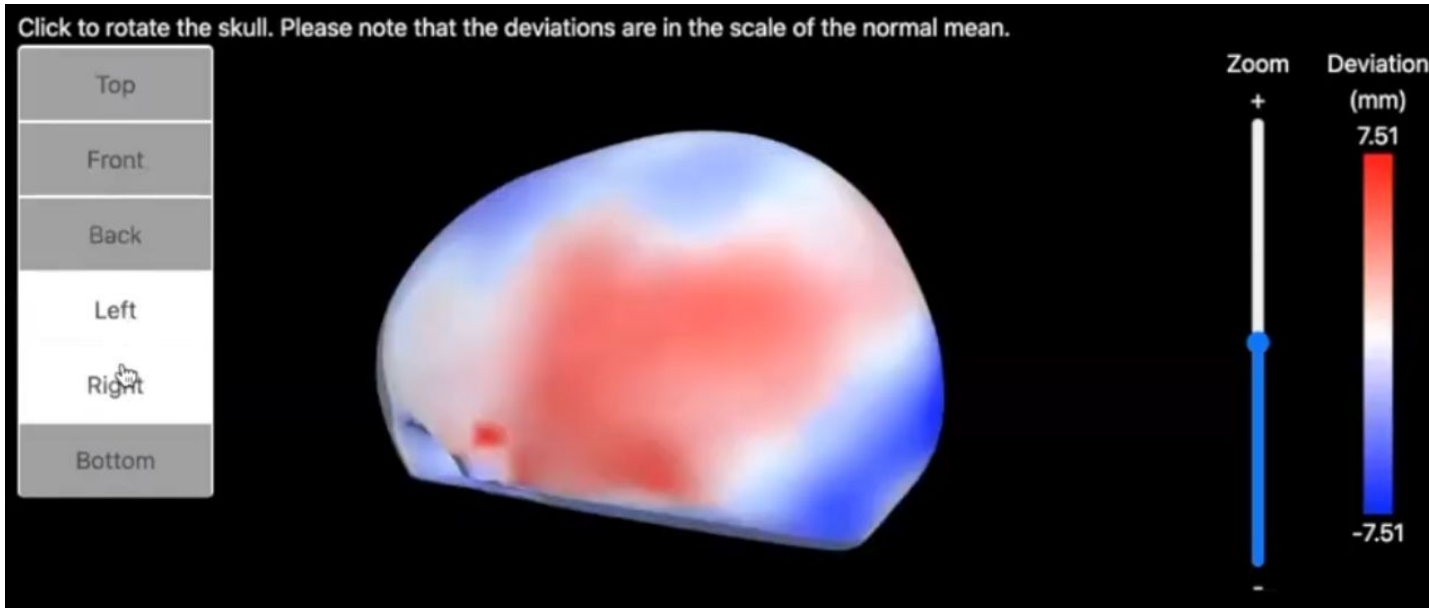
**Challenge #3: Define domain specific interpretability and create methods accordingly, including computer vision.**

Prediction = weighted sum of similarities towards the prototypes.



# Algorithmic Challenges in Interpretable ML

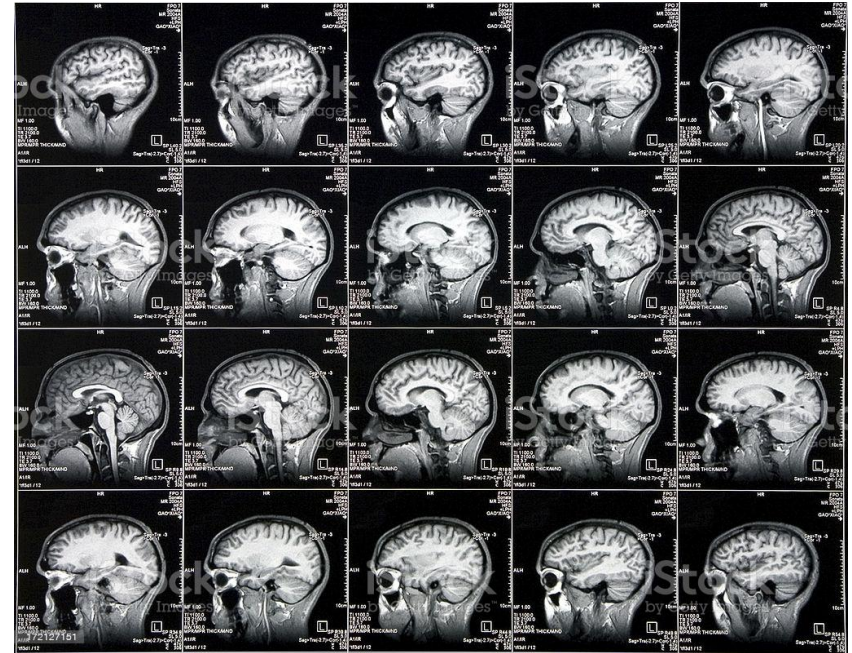
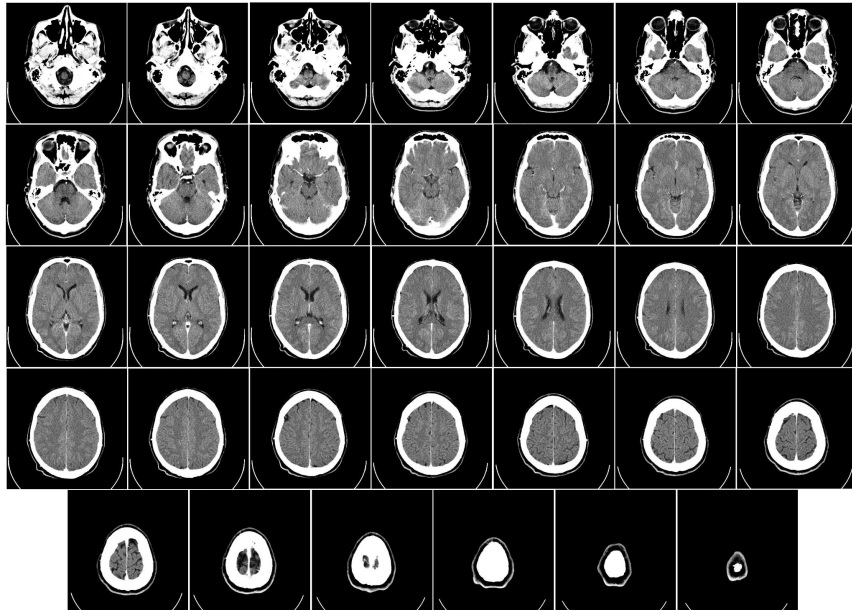
**Challenge #3: Define domain specific interpretability and create methods accordingly, including computer vision.**





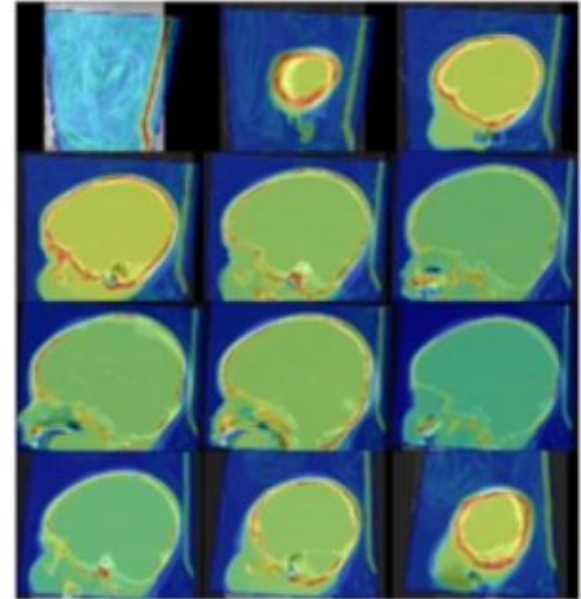
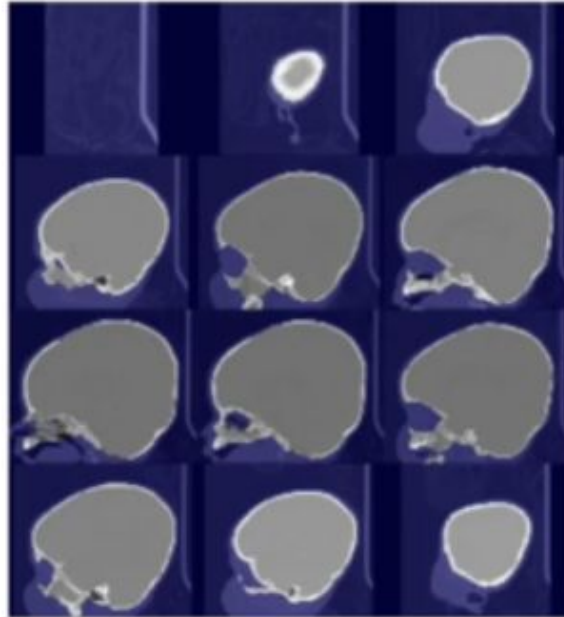
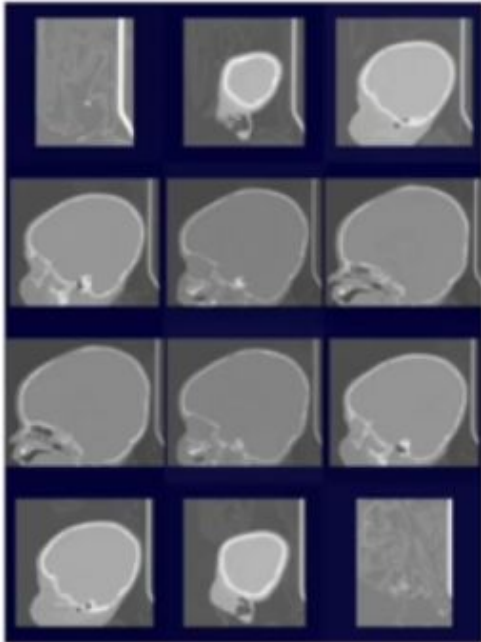
# Algorithmic Challenges in Interpretable ML

**Challenge #3: Define domain specific interpretability and create methods accordingly, including computer vision.**



# Algorithmic Challenges in Interpretable ML

**Challenge #3: Define domain specific interpretability and create methods accordingly, including computer vision.**



# Algorithmic Challenges in Interpretable ML

## **Interpretability for specific domains**

- Dimension reduction. For instance, PCA, SVD, etc
- Applied statistics problems embed the physics domain.

## **Why accurate interpretable models might exist in many domains**

- Hypothesis space is big enough that must contain some interpretable models.
- Many ML algorithms have similar performance on the same task even though they are fundamentally different, like random forest, NN, SVM, etc.

# Encouraging Responsible ML Governance

- The author claims that *there is no high-stakes application, which only black-box models are capable of.*

How about self-driving cars?

- Legislation, like General Data Protection Regulation (GDPR)
  - Provide an explanation for an automated decision.
  - No black-box model mandate? Or opacity requirement?
  - Hybrid: Black-box models + interpretable models

# Conclusion

- Explaining black-box models will not give accurate interpretations.
- Stop explaining black-box models and construct interpretable models instead.
- Creating interpretable models is hard due to 3 algorithmic Challenges.
  - Construct optimal logic models
  - Construct optimal sparse scoring systems
  - Define interpretability for specific domains and create methods accordingly.
- Interpretable models also bring other problems, such as intellectual property, security, responsibility, etc.

# Limitation

- The definition of **high stakes decisions** is not clarified.
  - Some claims are not supported appropriately and sufficiently.
    - COMPAS is not even a ML model.
  - Interpretable models have very restricted forms, which also means they are very specific towards the given tasks with the given data.
    - They are not generalized well are susceptible to context changes.
  - Still, no structured interpretation framework is provided.
- 
- **Using interpretable ML models does not mean to stop explaining models whether they are black-box or not.**

# Understanding Deep Learning Requires Rethinking Generalization

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Rech, Oriol Vinyals

Google, MIT, UC Berkeley

ICLR 2017

# What is generalization?

“Abstraction of common properties”

ML: Learning patterns that can be applied to unseen data

Generalization is a goal for learning

$$\text{generalization error} = \text{test error} - \text{train error}$$

*A model that has equally bad train and test performance, is said to ‘generalized well’.*

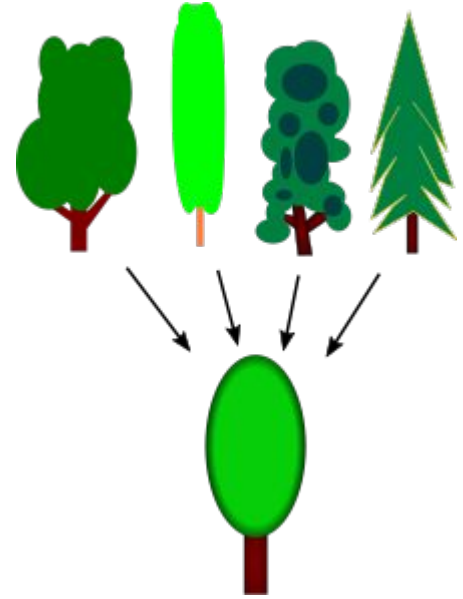


Image credit: Wikipedia



# What makes neural nets generalize? (Conventional Wisdom)

Data Augmentation  
Weight Decay  
Dropout  
...

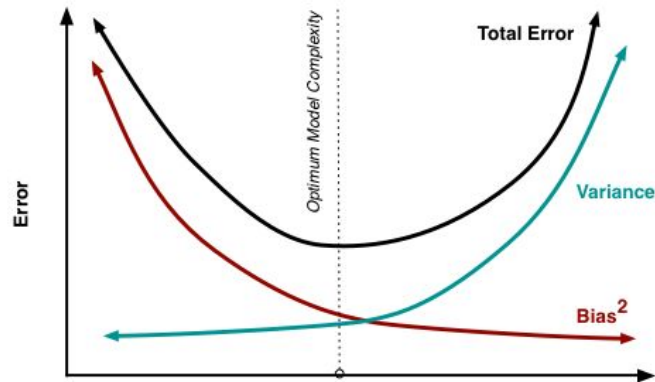
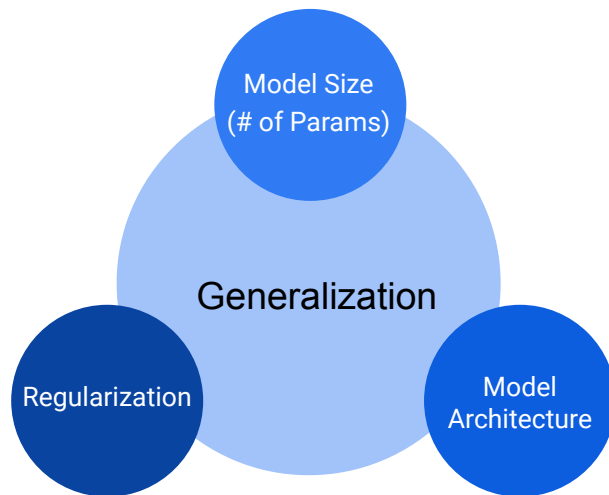


Image credit: CS4780 Cornell

Image credit: Medium PursuitData

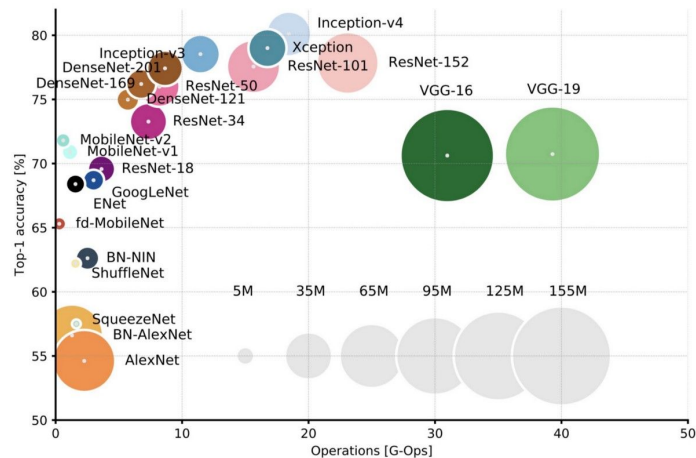
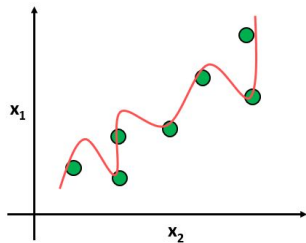
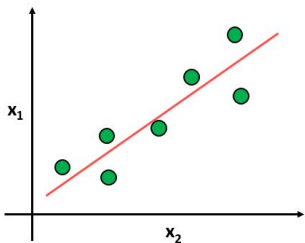
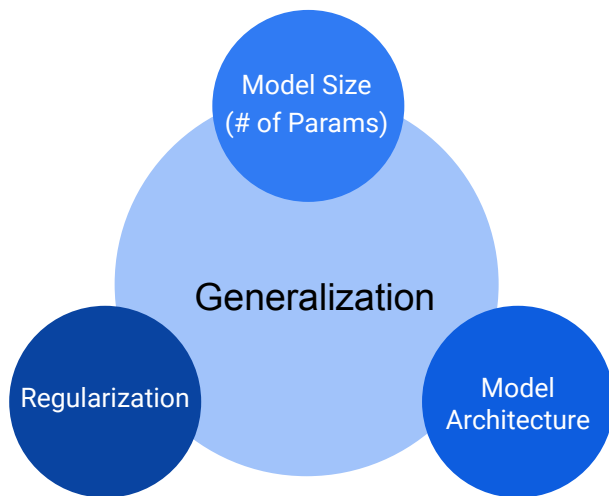


Image credit: arxiv:1605.07678

# Re-thinking about generalization



**Claim:** Even though they may **help** with generalization, **none** of these are **sufficient** or **necessary** to explain generalization.

# Re-thinking about generalization

**Claim:** **None** of the conventional theories are **sufficient** or **necessary** to explain generalization, even though they may **help** with generalization

$$\textit{generalization error} = \textit{test error} - \textit{train error}$$

## Not Sufficient: proof by counterexample

- Create a case where no low test error is possible
- To achieve generalization, the train error has to be as high as the test error
- Apply the conventional generalization techniques
- Observe that train error is still much lower than the test error -> not generalized
- Conclude that none of the conventional theories **guarantees** generalization

# Create a case where no low test error is possible

Scope: Image Classification

Random **labels**: replace the label with a uniform random class

Partially corrupted **labels**: with prob  $p$ , do random label

Shuffled **pixels**: same shuffle for all images

Random **pixels**: different shuffle for each image

Gaussian: generated **pixels** with the original distribution

CIFAR10: test error  $\sim 90\%$

“Randomization Test”

Imagenet: test error  $\sim 99.9\%$

# Models selected

CIFAR10: small Inception (85%), small Alexnet (76%), MLPs (50%)

Imagenet: Inception V3 (60%)

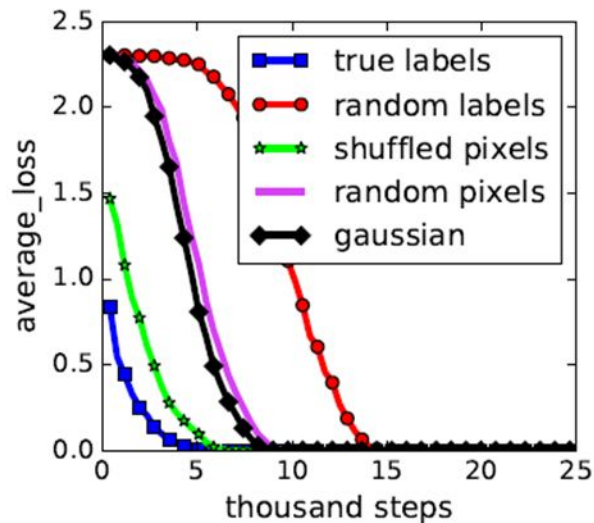
*\*Top-1 test accuracy without explicit regularization*

Models that generalized well.

*Does the good architecture and complexity in these models guarantee generalization?*

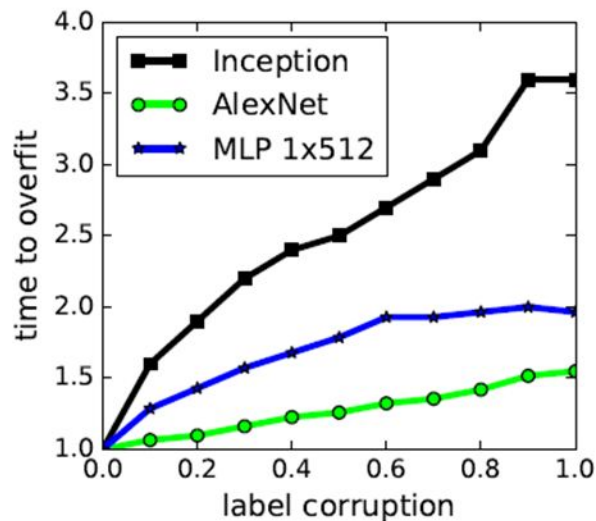
Let's see what happens in the randomization test.

No, it overfitted to the random training data

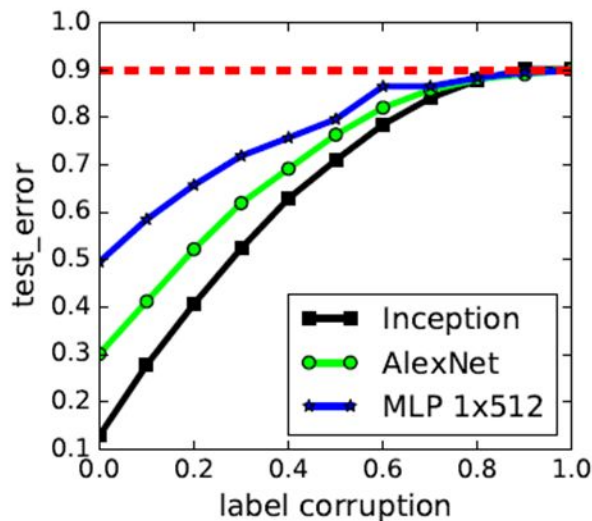


Learning curves for different randomization tests (CIFAR10 - Inception)

No, it overfitted to the random training data



(b) convergence slowdown



(c) generalization error growth

Partially corrupted label tests (CIFAR10 - All architectures)

No, it overfitted to the random training data

CIFAR10

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402				
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402				
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786				
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178				
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866				
(fitting random labels)		no	no	99.34	10.61



No, it overfitted to the random training data

Imagenet

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with random labels						
no	no	no	95.20	99.14	0.11	0.56

# Now we apply regularizations

data augmentation, weight decay, dropout

Do they prevent us from overfitting to random data?

# No, it still overfitted to the random training data

...with a little exception of Alexnet (not explained in the paper)

Model	Regularizer	Training Accuracy
Inception	Weight decay	100%
Alexnet		Failed to converge
MLP 3x512		100%
MLP 1x512		99.21%
Inception	Random Cropping <sup>1</sup>	99.93%
	Augmentation <sup>2</sup>	99.28%

Weight decay and data augmentation (CIFAR10)

No, it still overfitted to the random training data

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with random labels						
no	yes	yes	91.18	97.95	0.09	0.49
no	no	yes	87.81	96.15	0.12	0.50
no	no	no	95.20	99.14	0.11	0.56

Weight decay and dropout (Imagenet)

# Re-thinking about generalization

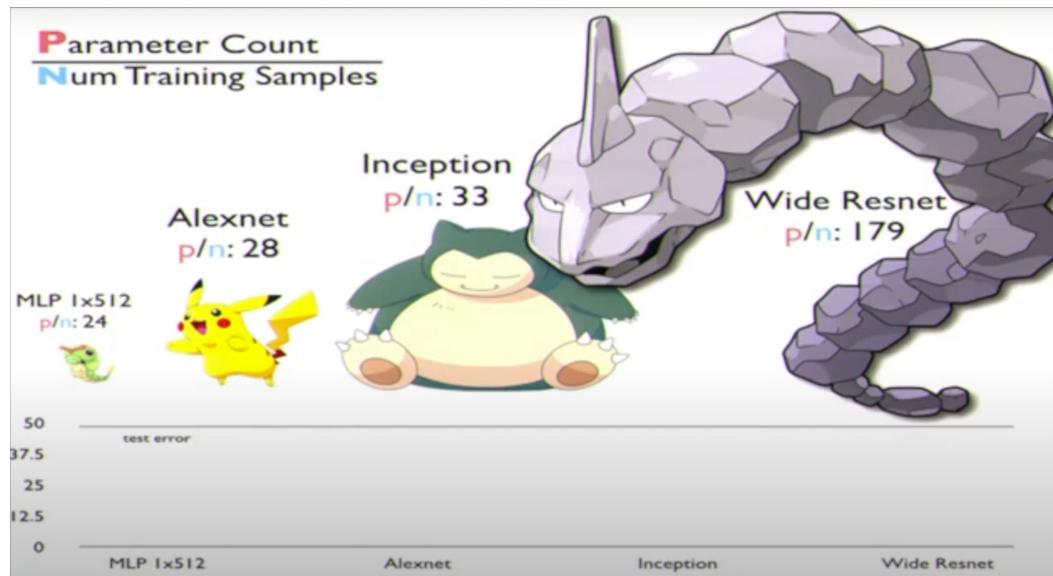
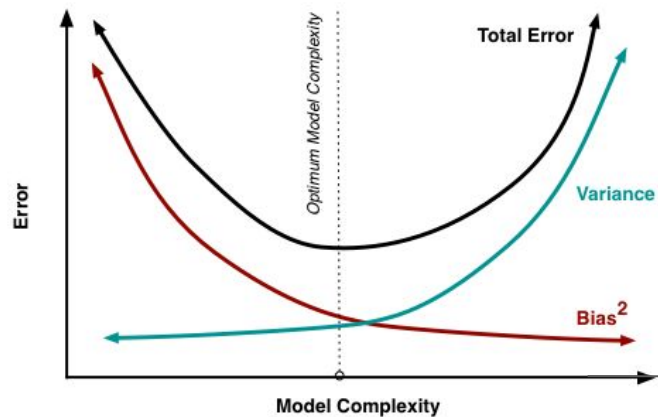
**Claim:** **None** of the conventional theories are **sufficient** or **necessary** to explain generalization, even though they may **help** with generalization

$$\textit{generalization error} = \textit{test error} - \textit{train error}$$

## Not Sufficient: proof by counterexample

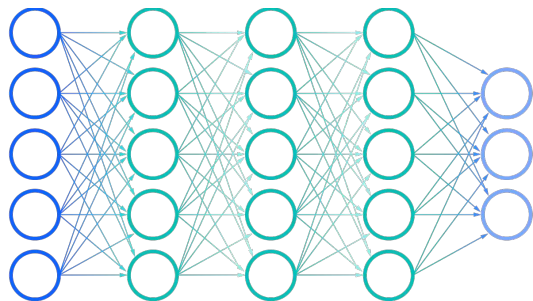
- Create a case where no low test error is possible
- To achieve generalization, the train error has to be as high as the test error
- Apply the conventional generalization techniques
- Observe that train error is still much lower than the test error -> not generalized
- Conclude that **none** of the conventional theories **guarantees / is sufficient for** generalization

# Authors' intuition



★  
Low bias

# Finite Sample Expressivity



$P$  parameters



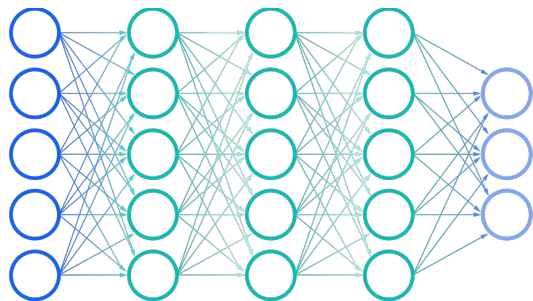
$N$  samples

*When  $P > N$ , the model can ‘shatter’ the data.*

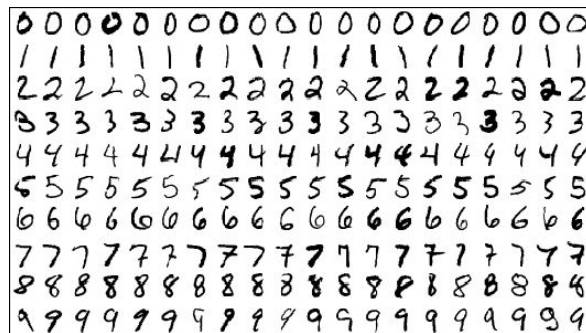
*Shatter: can represent any function of the sample size*

*~ perfectly fit to any given labelling of the data*

# Finite Sample Expressivity



$P$  parameters



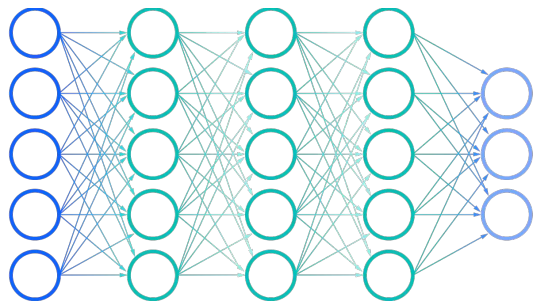
$N$  samples

*Theorem: There exists a two-layer neural network with ReLU activations and  $P = 2N+D$  weights that can represent any function on a sample of size  $N$  in  $D$  dimensions.*

*Proved in Appendix C in the paper*



# Finite Sample Expressivity



$P$  parameters

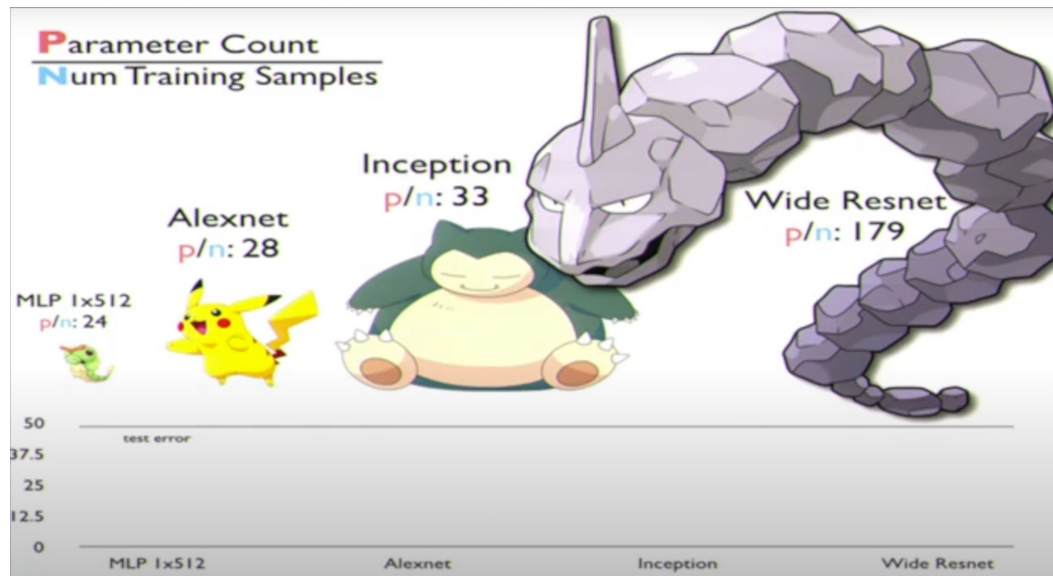
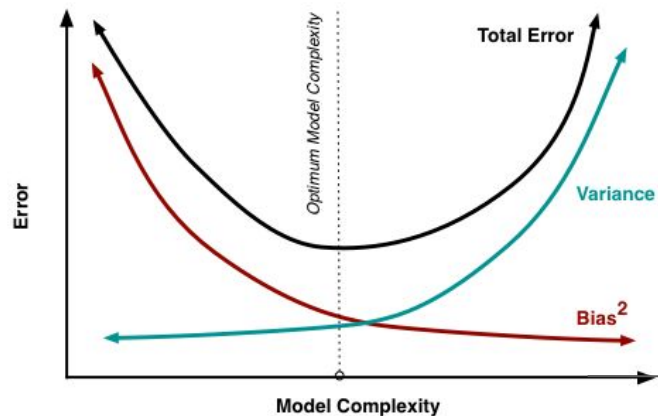


$N$  samples

*Corollary: For every  $k \geq 2$ , there exists neural network with ReLU activations of depth  $k$ , width  $O(N/k)$  and  $O(N + D)$  weights that can represent any function on a sample of size  $N$  in  $D$  dimensions.*

*Proved in Appendix C in the paper*

# Authors' intuition



Low bias  
But why is the variance low?

# Re-thinking about generalization

**Claim:** **None** of the conventional theories are **sufficient** or **necessary** to explain generalization, even though they may **help** with generalization

$$\textit{generalization error} = \textit{test error} - \textit{train error}$$

## Helpful but Not Necessary: proof by counterexample

- Create a case where low test error is possible
- Evaluate the model performance with / without following the conventional theories
- Observe that following the conventional theories improves generalization of the model
- Observe that model still generalizes to some extent without following the conventional theories
- Conclude that none of the conventional theories is necessary for generalization to happen

With no explicit regularizers, the model generalizes

Imagenet

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with the original labels						
yes	yes	yes	92.18	99.21	77.84	93.92
yes	no	no	92.33	99.17	72.95	90.43
no	no	yes	90.60	100.0	67.18 (72.57)	86.44 (91.31)
no	no	no	99.53	100.0	59.80 (63.16)	80.38 (84.49)
Alexnet (Krizhevsky et al., 2012)			-	-	-	83.6

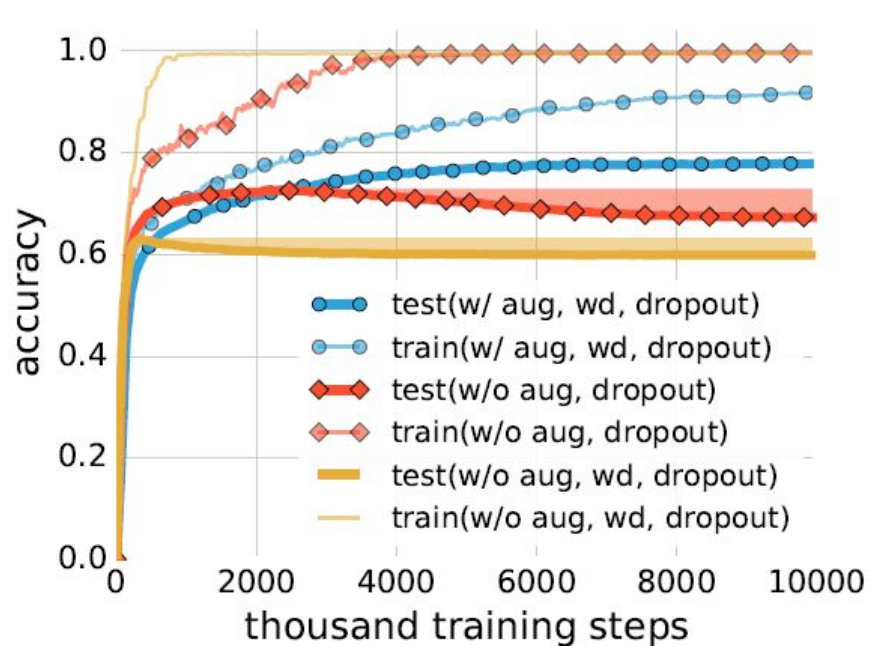
# With no explicit regularizers, the model generalizes

... even with the  
simple MLP  
architecture

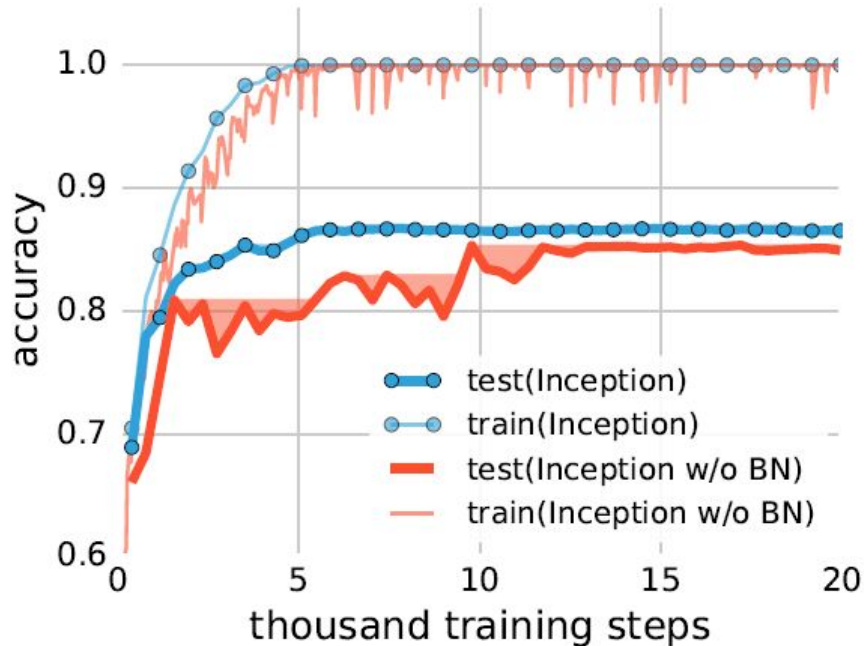
## CIFAR10

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
		(fitting random labels)	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
		(fitting random labels)	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
		(fitting random labels)	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
		(fitting random labels)	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
		(fitting random labels)	no	99.34	10.61

# Implicit regularizers don't help much either



(a) Inception on ImageNet



(b) Inception on CIFAR10

Early stop and Batch normalization as implicit regularizers

# Re-thinking about generalization

**Claim:** **None** of the conventional theories are **sufficient** or **necessary** to explain generalization, even though they may **help** with generalization

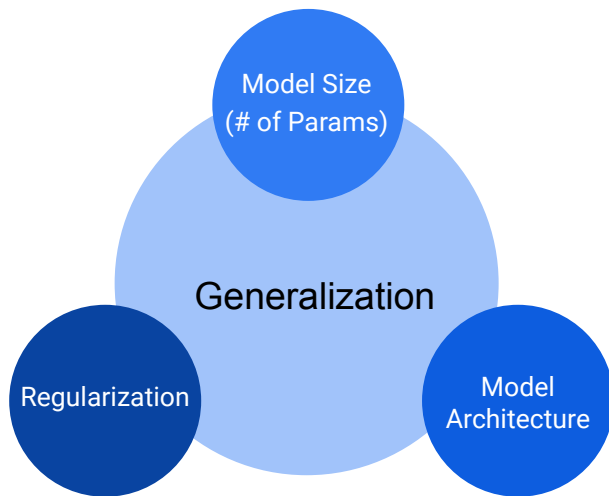
$$\textit{generalization error} = \textit{test error} - \textit{train error}$$

## Helpful but Not Necessary: proof by counterexample

- Create a case where low test error is possible
- Evaluate the model performance with / without following the conventional theories
- Observe that following the conventional theories improves generalization of the model
- Observe that model still generalizes to some extent without following the conventional theories
- Conclude that **none** of the conventional theories **is necessary** for generalization to happen

# Re-thinking about generalization

Finite Sample Expressivity



Role of Explicit and Implicit  
Regularization

Role of Model Architecture

**Claim:** None of the conventional theories are **sufficient** or **necessary** to explain generalization, even though they may **help** with generalization



# Conclusions

- Randomization Test - A simple experimental framework for defining and understanding a notion of effective capacity of machine learning models
- Neural networks are large enough to shatter the training set (finite sample expressivity)
- Conventional theories work as practical techniques, but we have not yet understood the fundamental reason of generalization on the *over-parameterized* regime (e.g. deep learning)

# SGD as an implicit regularizer

In the context of *Linear Models*, out of all models that exactly fit the data, SGD will often converge to the solution with *minimum norm*.

=> implicitly regularizes the solution.

# Thoughts & Discussion

ICLR 2017 Best Paper Award

Hard to judge in 2021 as we are standing on the shoulders of giants

Limited to Supervised, image classification.

- More regularization helped generalize better
- Alexnet didn't converge on random data with regularization

# Thoughts & Discussion

What about generalization in other domains? Not image, not classification, or even not supervised learning?

How about the role of cross-validation / hyperparameter tuning?

Randomization Test Reproducibility may be low (i.e. no code, ambiguity in algorithm description (e.g. Gaussian test: image-wise or pixel-wise distribution?))

Finally, some interesting holes in the arguments...

# Limitations

## Helpful but Not Sufficient: proof by counterexample

- Create a case where no low test error is possible
- To achieve generalization, the train error has to be as high as the test error
- Apply the conventional generalization techniques
- Observe that train error is still much lower than the test error
- Conclude that none of the conventional theories guarantees generalization

Assumption: Generalization needs to be independent of whether the data makes sense/has patterns or not.

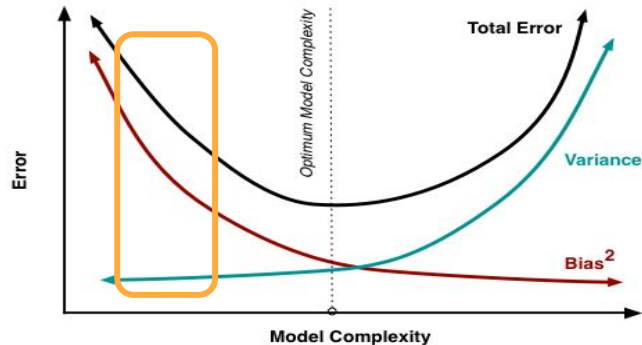
But why? Is there a scenario where we care about generalizing if the data has no intrinsic pattern?

# Limitations

## Helpful but Not Sufficient: proof by counterexample

- Create a case where no low test error is possible
- To achieve generalization, the train error has to be as high as the test error
- Apply the conventional generalization techniques
- Observe that train error is still much lower than the test error
- Conclude that none of the conventional theories guarantees generalization

Minimum Pattern



Did not show due diligence on using the strictest possible regularization

Used the default weight decay rate

Missing results on Imagenet with data augmentation, CIFAR10 with dropout etc.

Alexnet exception not explained

# Other literatures on this topic

[Representation Based Complexity Measures for Predicting Generalization in Deep Learning](#)

[Why Over-parameterization of Deep Neural Networks Does Not Overfit?](#)

[Fantastic Generalization Measures and Where to Find Them](#)

[The Deep Bootstrap Framework: Good Online Learners are Good Offline Generalizers](#)

[Deep learning: a statistical viewpoint](#)

[Are Deep Neural Networks Dramatically Overfitted?](#)

# Understanding Black-box Predictions via Influence Functions

Pang Wei Koh and Percy Liang  
Stanford University

International Conference on Machine Learning, 2017



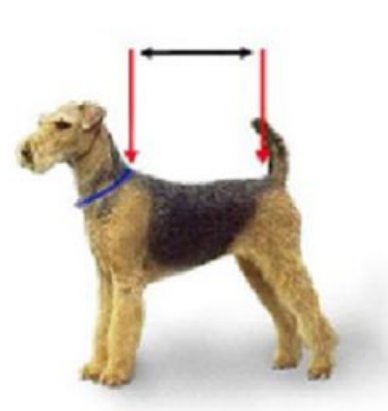
# Guiding Question

Given a trained model and a test image, what are the most influential training images to the classification of that test image?

Test Image



Most Influential Training Images



# Mathematical Approach: Influence Functions

# Measuring impact of a training point

Training to minimize empirical risk

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta)$$

Difference in model parameters removing point  $z$

$$\hat{\theta} - \hat{\theta}_{-z}$$

# Influence Functions

Measure effect of upweighting point  $z$  by a small epsilon

$$\hat{\theta}_{\epsilon, z} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z, \theta)$$

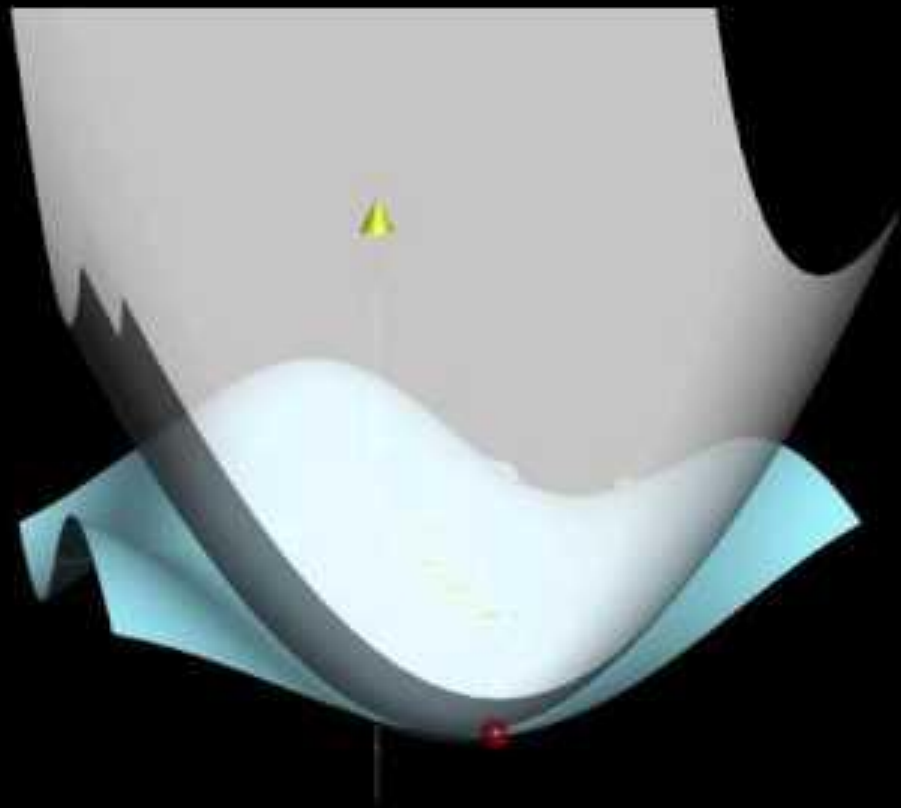
# Influence Functions

Want to measure effect of upweighting point  $z$  by a small epsilon

$$\hat{\theta}_{\epsilon, z} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z, \theta)$$

By previous work on influence functions, we know that we can represent this influence of upweighting as:

$$\mathcal{I}_{\text{up, params}}(z) = - \underbrace{\left( \frac{1}{n} \sum_{i=1}^n \nabla_{\theta}^2 L(z_i, \hat{\theta}) \right)^{-1}}_{\text{Hessian}} \nabla_{\theta} L(z, \hat{\theta})$$



# Adapting influence functions to find relevant training points

Rather than upweight by small epsilon, we want to remove the point. They propose a linear approximation:

$$\hat{\theta}_{-z} - \hat{\theta} \approx -\frac{1}{n} \mathcal{I}_{\text{up, params}}(z)$$

Through small adaptations, they can compute the influence of a training point on the loss at a specific test point  $\mathcal{I}_{\text{up, loss}}(z, z_{\text{test}})$  and the influence of perturbing a training point  $\mathcal{I}_{\text{pert, loss}}(z, z_{\text{test}})$

## Influence at a specific test point

By applying the chain rule, they compute the influence of upweighting  $z$  on the loss at  $z_{\text{test}}$ :

$$\mathcal{I}_{\text{up, loss}}(z, z_{\text{test}}) := \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} \mathcal{I}_{\text{up, params}}(z)$$



## Perturbations of training point $z=(x, y)$

Think of it as upweighting  $z_\delta$  and downweighting  $z$ :

$$\mathcal{I}_{\text{up, params}}(z_\delta) - \mathcal{I}_{\text{up, params}}(z)$$

Closed form at a particular test point is almost the same as before:

$$\mathcal{I}_{\text{pert, loss}}(z, z_{\text{test}}) := \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^\top - H_{\hat{\theta}}^{-1} \nabla_x \nabla_{\theta} L(z, \hat{\theta})$$

Computing the closed form is costly!

Hessian:  $O(np^2 + p^3)$

Also, need to compute  $\mathcal{I}_{\text{up, loss}}(z_i, z_{\text{test}})$  for each  $z_i$  in the training set and  $z_{\text{test}}$  in the test set!

# Hessian-vector Products (HVP)

Represent part of the influence equation as an HVP:

$$s_{\text{test}} := \underbrace{H_{\hat{\theta}}^{-1}}_{\text{Hessian}} \underbrace{\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})}_{\text{Vector}}$$

# Hessian-vector Products (HVP)

Represent part of the influence equation as an HVP:

$$s_{\text{test}} := \underbrace{H_{\hat{\theta}}^{-1}}_{\text{Hessian}} \underbrace{\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})}_{\text{Vector}}$$

Can compute exact HVPs using conjugate gradients algorithm in  $O(np)$ .

Stochastic approximation is also  $O(np)$ , but faster in practice.

# Applying HVP to influence function computation

Precompute  $\mathbf{s}_{\text{test}}$  for each  $\mathbf{z}_{\text{test}}$  using CG or stochastic approximation:

$$\mathbf{s}_{\text{test}} := H_{\hat{\theta}}^{-1} \nabla_{\theta} L(\mathbf{z}_{\text{test}}, \hat{\theta})$$

Compute influence for each training sample:

$$\mathcal{I}_{\text{up, loss}}(\mathbf{z}, \mathbf{z}_{\text{test}}) = -\mathbf{s}_{\text{test}} \cdot \nabla_{\theta} L(\mathbf{z}, \hat{\theta})$$

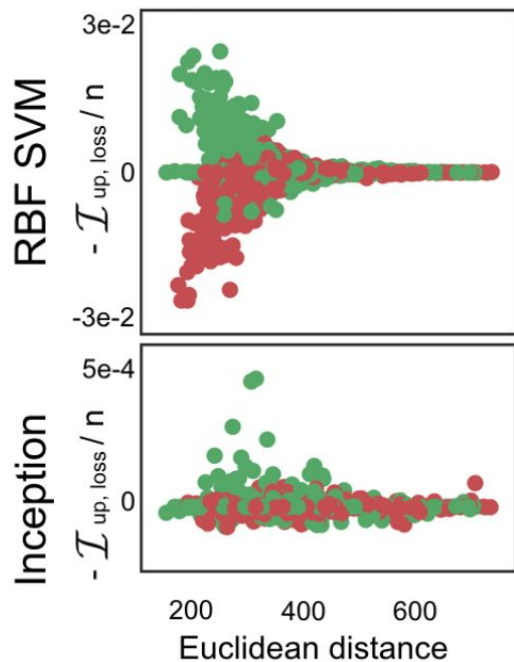
# Use Cases

# Use Case 1: Model Explanation

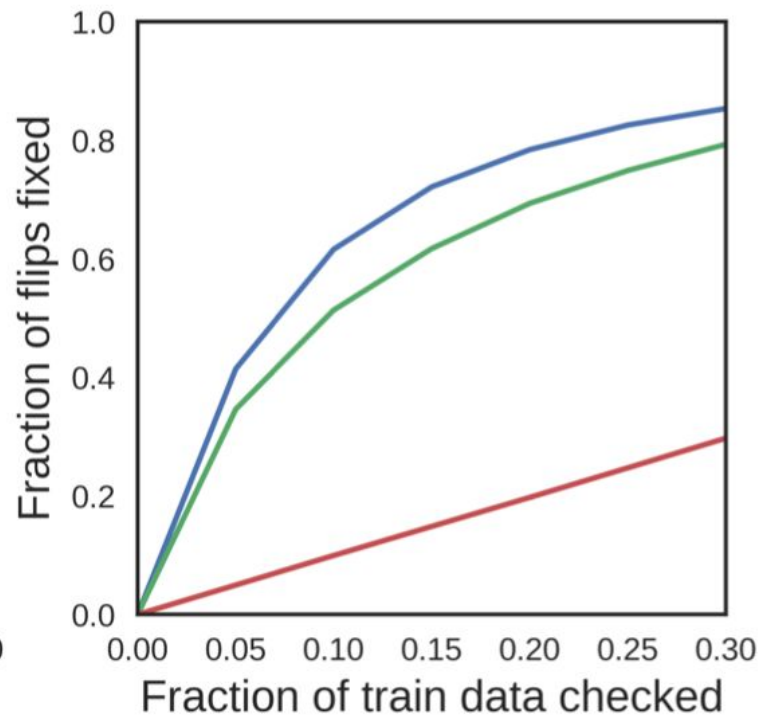
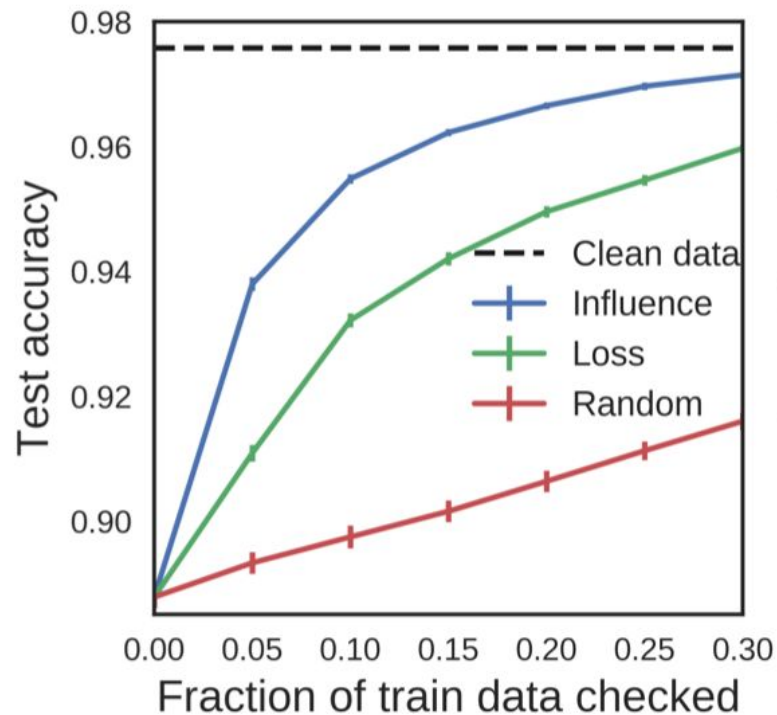
Test image



Helpful train  
dog image  
(Inception)



## Use Case 2: Targeted Training Set Debugging





# Use Case 3: Adversarial Training Attack

## Threat Model

- White box attack
  - Access to model parameters and output
  - Can see and modify training data during training
- Goal: create a backdoor (targeted misclassification of specific input or set of inputs)

## Use Case 3: Adversarial Training Attack

Iterative Approach: For  $z_{\text{test}}$  and training image  $z_i$ , compute

$$\tilde{z}_i := \prod(\tilde{z}_i + \alpha \text{sign}(\mathcal{I}_{\text{pert, loss}}(\tilde{z}_i, z_{\text{test}})))$$

Retrain with new, perturbed training sample at each iteration.

With 100 iterations, 57% success rate of flipping correctly labeled examples.

# Use Case 3: Adversarial Training Attack


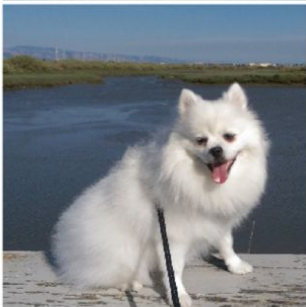
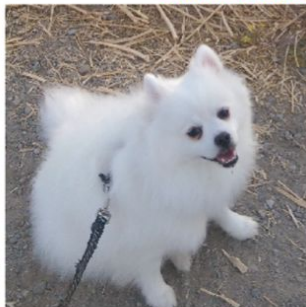


A small perturbation to one **training** example:

Label: Fish

+  $\epsilon$

Label: Fish

Can change multiple **test** predictions:

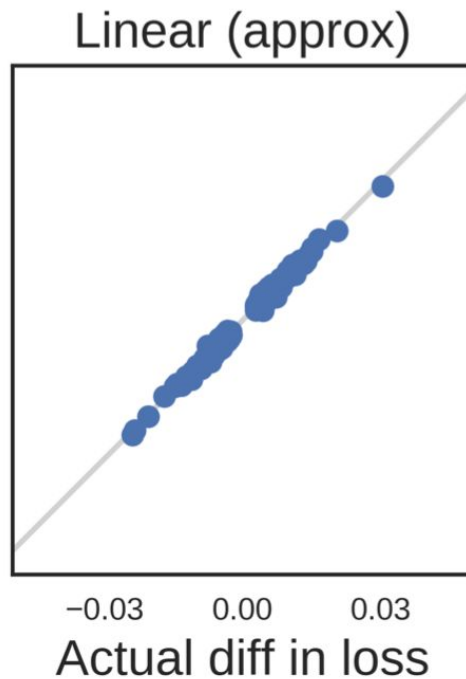
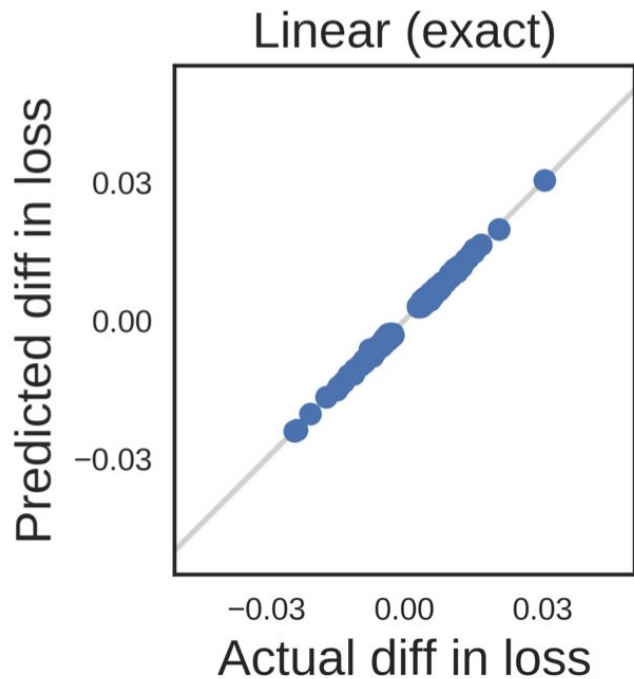
Image	Orig (confidence):	New (confidence):
	Dog (97%)	Fish (97%)
	Dog (98%)	Fish (93%)
	Dog (98%)	Fish (87%)
	Dog (99%)	Fish (60%)
	Dog (98%)	Fish (51%)

How good are the approximations?

# Review of Assumptions and Approximations

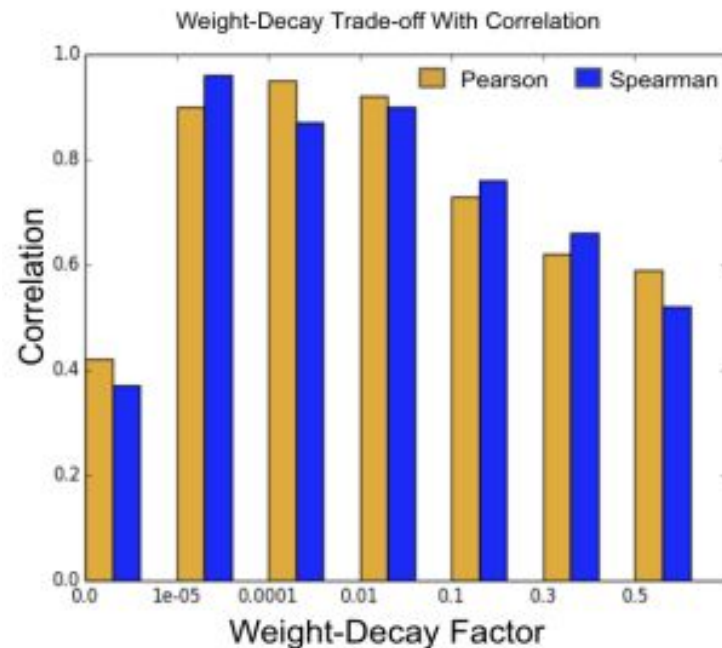
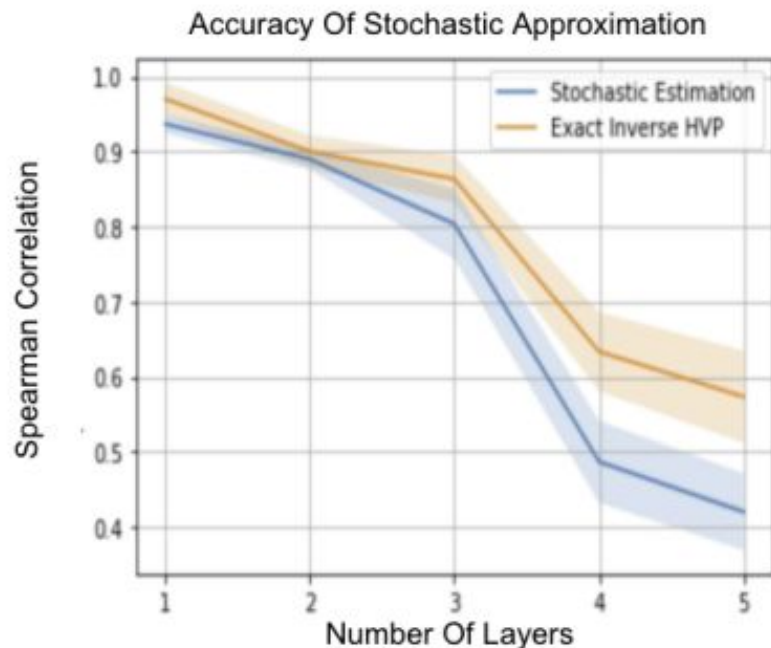
- Assumptions:
  - Loss is convex and twice differentiable
  - Training finds parameters with minimal empirical risk
- Approximations:
  - Linear approximation of removing training point
  - Stochastic estimation of HVPs

# Validating Empirical Accuracy



# Influence Functions in Deep Learning are Fragile

Samyadeep Basu, Phil Pope, Soheil Feisi, ICLR 2021



“The scale of ImageNet raises additional questions about the feasibility of leave-one-out retraining as the ground truth estimator.

Given that there are 1.2M images in the training set, is it even possible that the removal of one image can significantly alter the model?”



# Conclusion: When does this method make sense?

## Good for:

- Small models
- Small/medium data
- Targeted proofreading
- Targeted backdoor attack (for an attacker with access to training data)

## Not so good for:

- Big models
- Big data
- General “explanation”
- Availability attack
- Attacker with limited access to training data

# Food for thought

- Potential applications to unlearning - estimating effect of retraining without a given training point on model parameters
- In model ownership, we talked about the data being what is actually proprietary/important, not the model - does the same reasoning apply to interpretability? Can you explain a model with just the data, or do we need to look at the model itself as well?

Thank you!