

# Making Deep Neural Networks More Robust

MIML Seminar

McGill Math

Oct 30, 2018

Adam Oberman

McGill Dept of Math and Stats

*in collaboration with*

*Chris Finlay (McGill PhD student) and Jeff Calder (U Michigan)*

*research supported by AFOSR FA9550-18-1-0167*

# Background AI

- Artificial Intelligence is loosely defined as intelligence exhibited by machines
- Operationally: R&D in CS academic sub-disciplines: Computer Vision, Natural Language Processing (NLP), Robotics, etc



AlphaGo uses DL to beat world champion at Go



# Artificial General Intelligence (AGI)

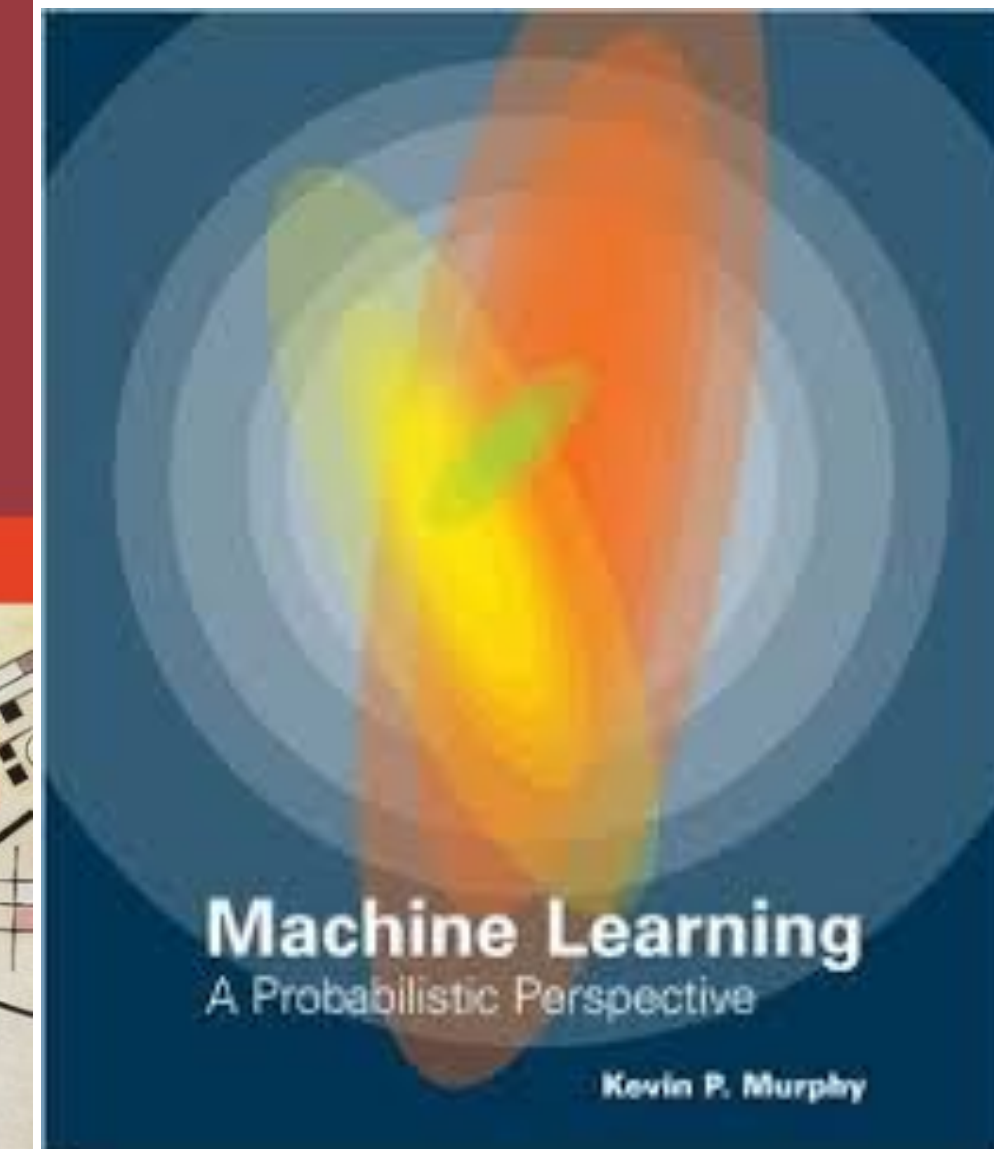
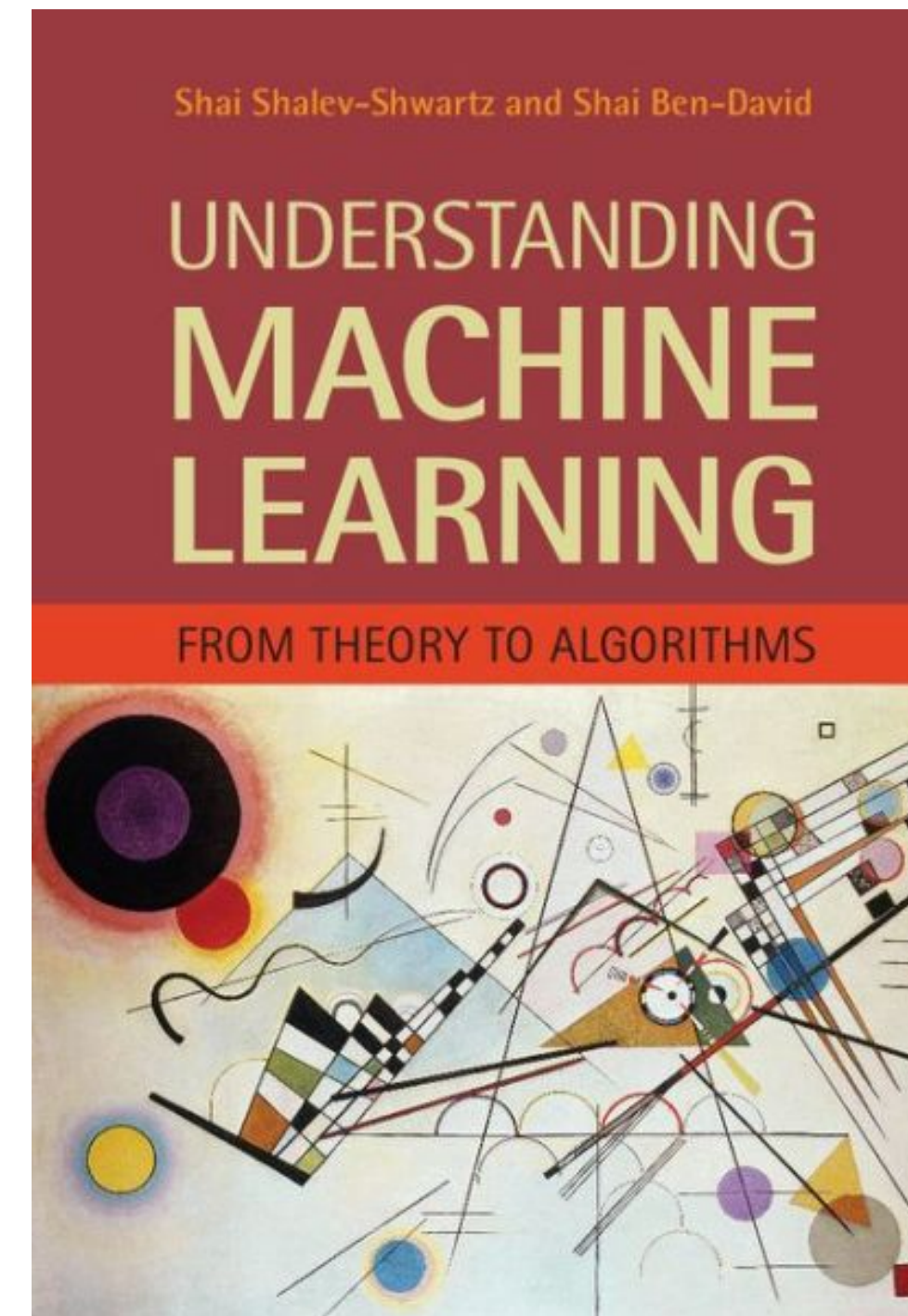
- AI : specific tasks,
- AGI : general cognitive abilities.
- AGI is a small research area within AI: build machines that can successfully perform *any* task that a human might do
- So far, no progress on AGI.





# Machine Learning (ML) vs Deep Learning (DL)

- Machine Learning (ML) has been around for some time.
- Deep Learning is newer branch of ML which uses Deep Neural networks.
- ML has theory: error estimates and convergence proofs.
- DL less theory. But DL can effectively solve much bigger problems





# Deep Learning: solve big problems using GPUs.

- ImageNet: Total number of classes: 21841
- Total number of images: 14,197,122
- Size: about 200 Gig



Facebook used 256 GPUs, working in parallel, to train ImageNet.

*Still an academic dataset.*  
Total number of images on Facebook is much larger



# Challenges for deep learning

“It is not clear that the existing AI paradigm is immediately amenable to any sort of software engineering validation and verification. This is a serious issue, and is a potential roadblock to DoD’s use of these modern AI systems, especially when considering the liability and accountability of using AI”

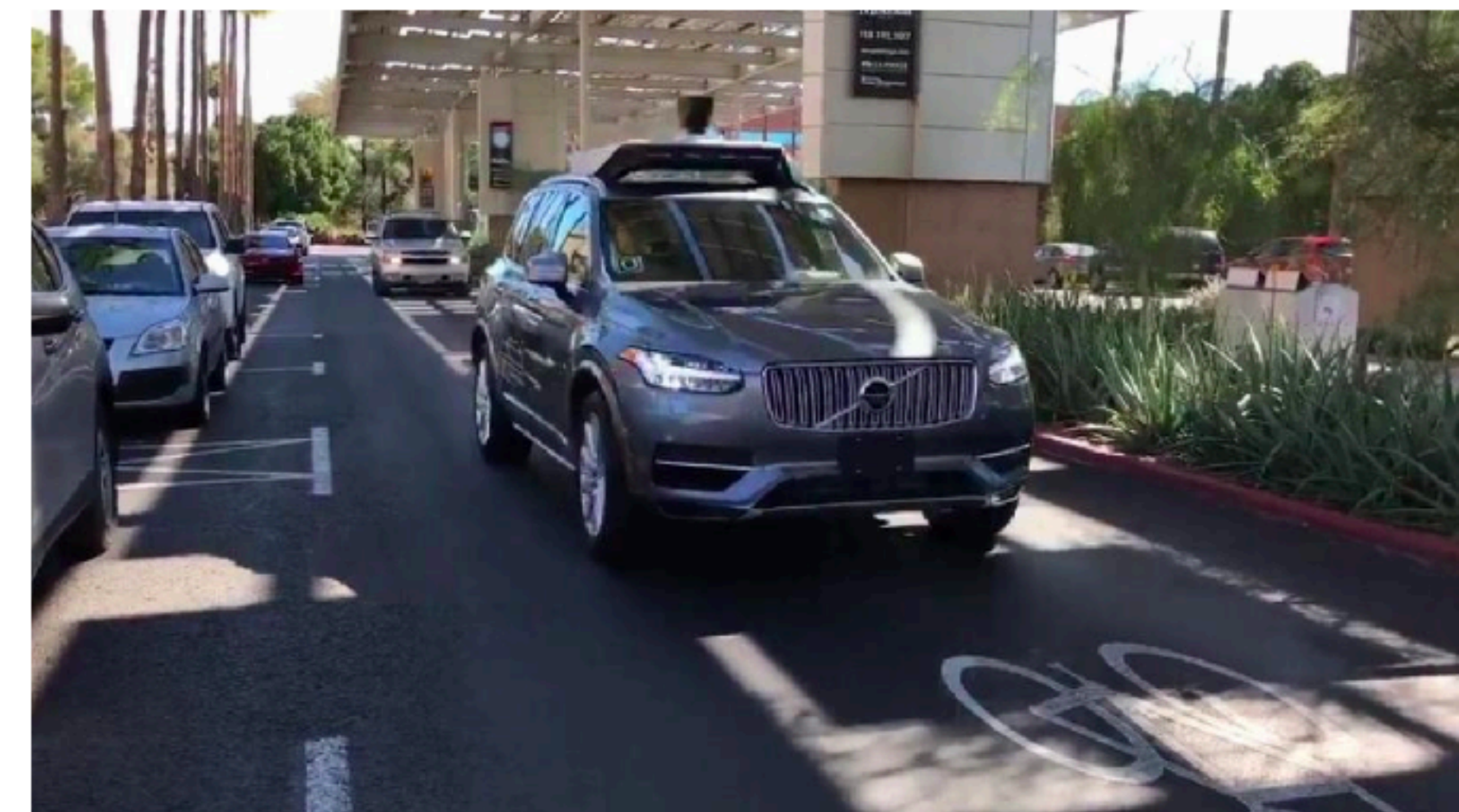
JASON report

## Self-Driving Uber Hits, Kills Pedestrian in Arizona

*The Uber vehicle was operating in autonomous mode with a human behind the wheel in Tempe, Arizona, when the incident occurred overnight.*

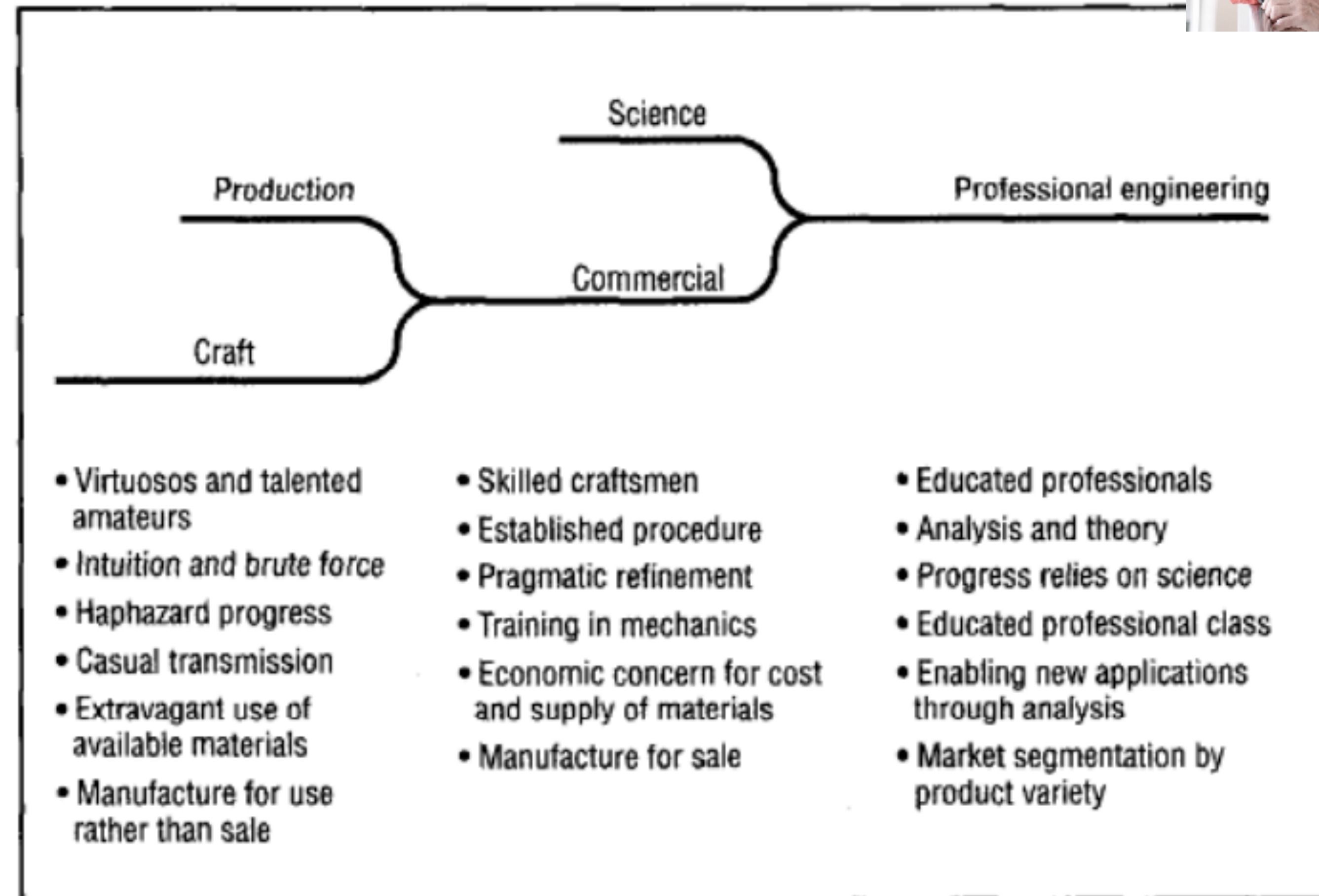


By Angela Moscaritolo March 19, 2018 2:07PM EST





# Mary Shaw's evolution of software engineering discipline



Better theory: improves reliability and discipline evolves

# AI History

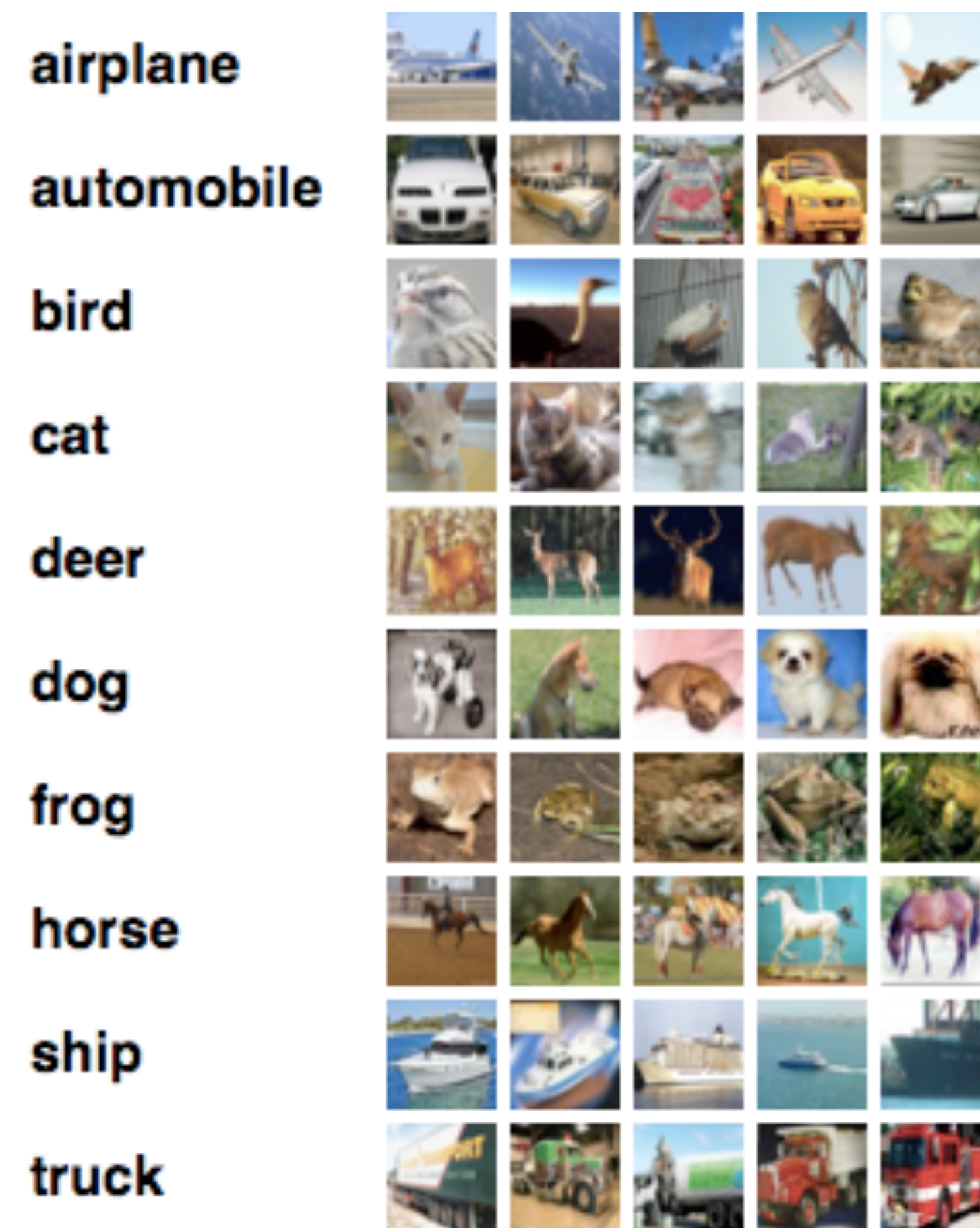
- “AI” coined in 1956. Perceptrons 1960 implied machines could learn from data
- Decline in 1969 - perceptron not a universal function approximator,
- 1980 resurgence in AI “expert systems”. Petered out
- 1990s academic AI in doldrums
- Improved computers led, in 1997 to IBM Deep Blue beats champion Gary Kasparov in chess.

*Chess, once believed to require human intelligence, fell to a special-purpose very fast algorithm.*



# 2010: Deep Learning Revolution

- Neural Networks have been around for half a century. Popular in the 1990's for solving simple tasks.
- Starting around 2010, new hardware, Graphics Processor Units (GPU)s, became available, which allowed for much larger, and deeper networks.
- large labelled data sets become available, allowed for training.



CIFAR10 dataset



# 2012-2015: ImageNet won by AlexNet

- The large data set ImageNet was available in 2005.
- In 2012 Alexnet, trained on GPUs, won the 2012 ImageNet competition, with an error of 15.3%, more than 10% better than the runner up. Canadian (U Toronto) team: Alex Krizhevsky, [Geoffrey Hinton](#), and [Ilya Sutskever](#).
- Between 2011 and 2015, error rate for image captioning by computer fell from 25% to 3%, better than accepted human figure of 5%

more than 95% prediction correct  
caption (green column)

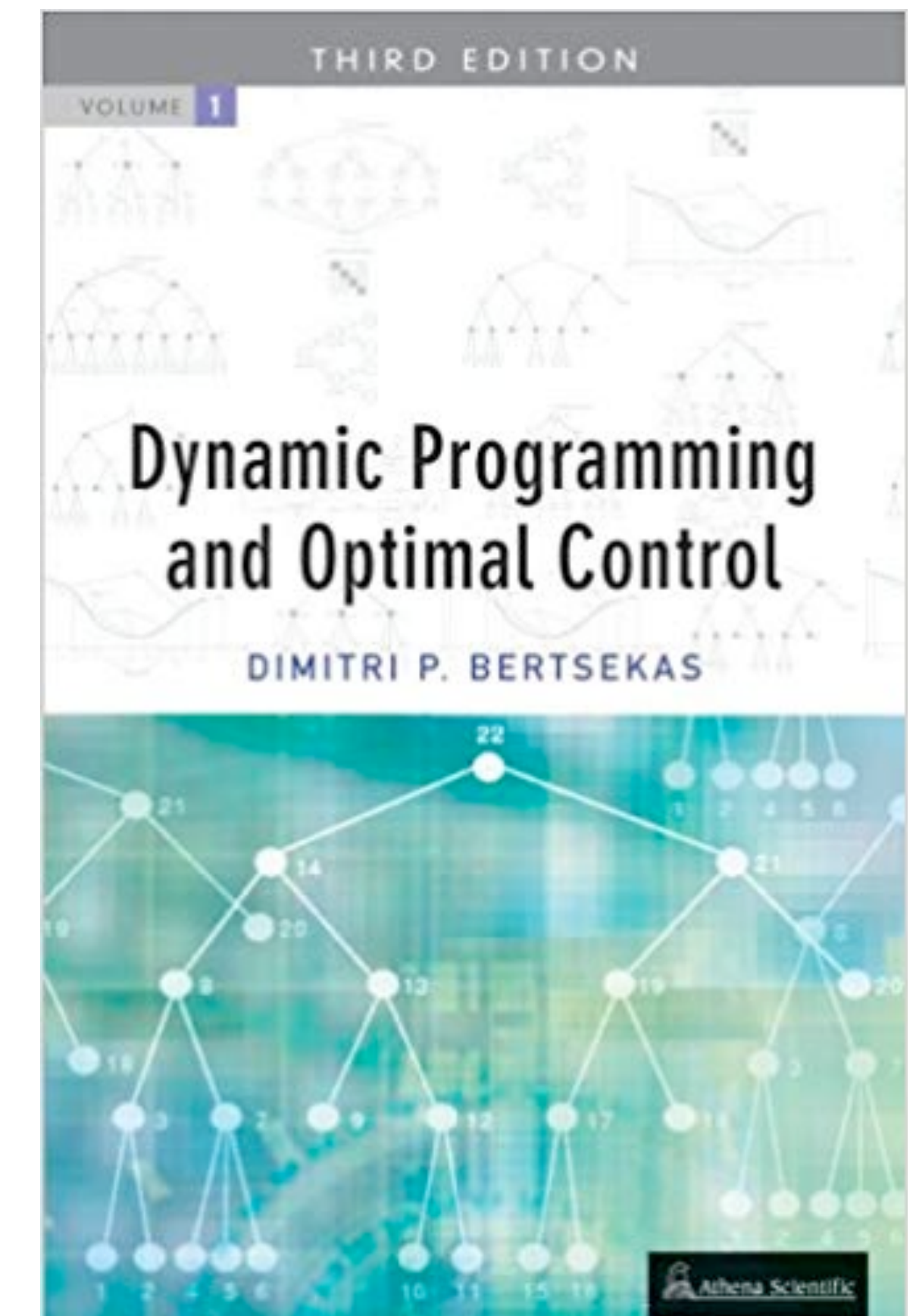
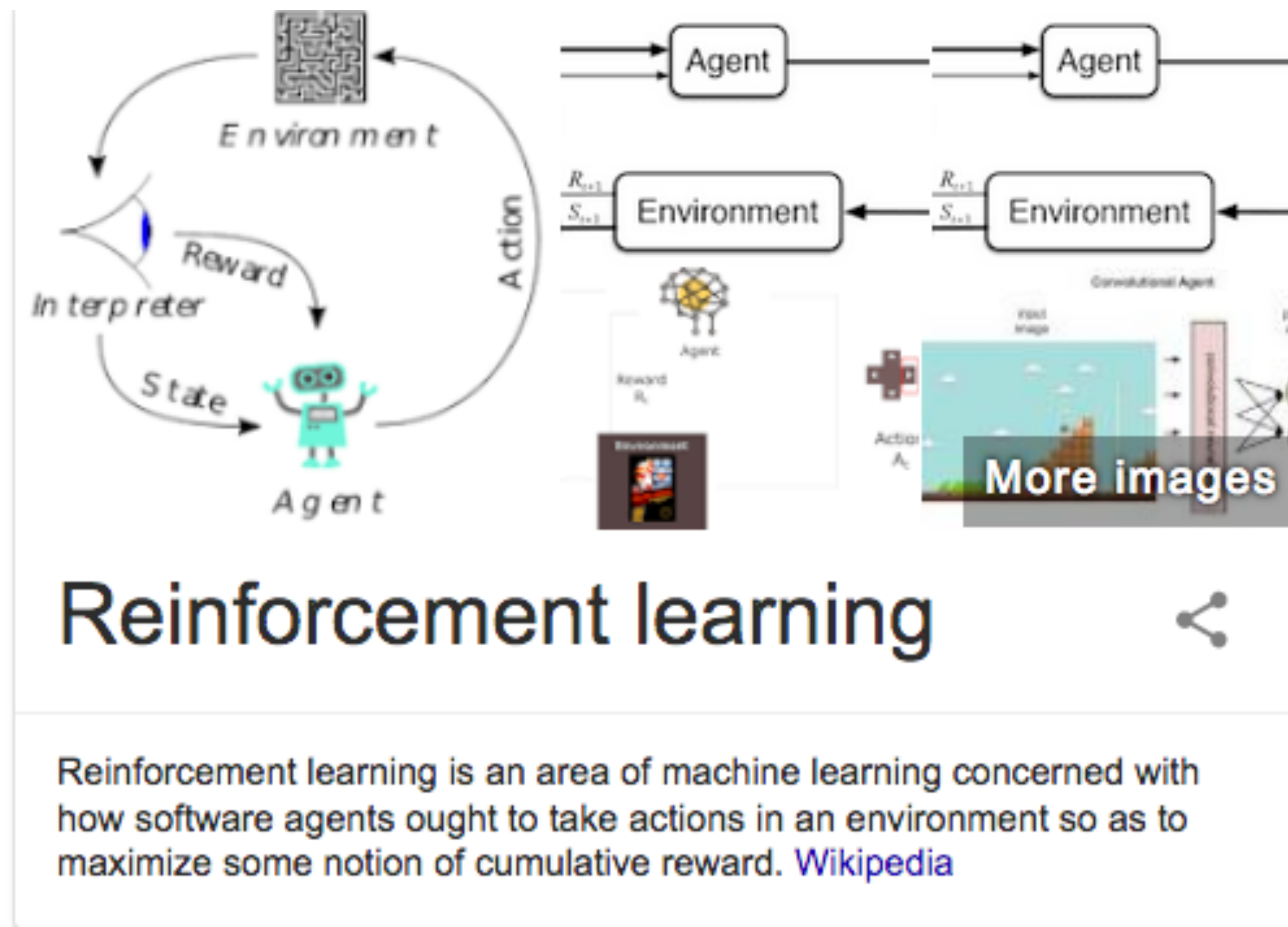




# New AI Applications and connections to Mathematics

# Reinforcement Learning

- Related to dynamic Programming
- Computationally intensive and unstable



Related math: dynamic programming, Optimal Control



# Recurrent NN



## Speech recognition

Field of study

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as automatic speech recognition, computer speech recognition or speech to text. [Wikipedia](#)

Russian ▾



French ▾



почему математика  
интересна Edit

pochemu matematika interesna

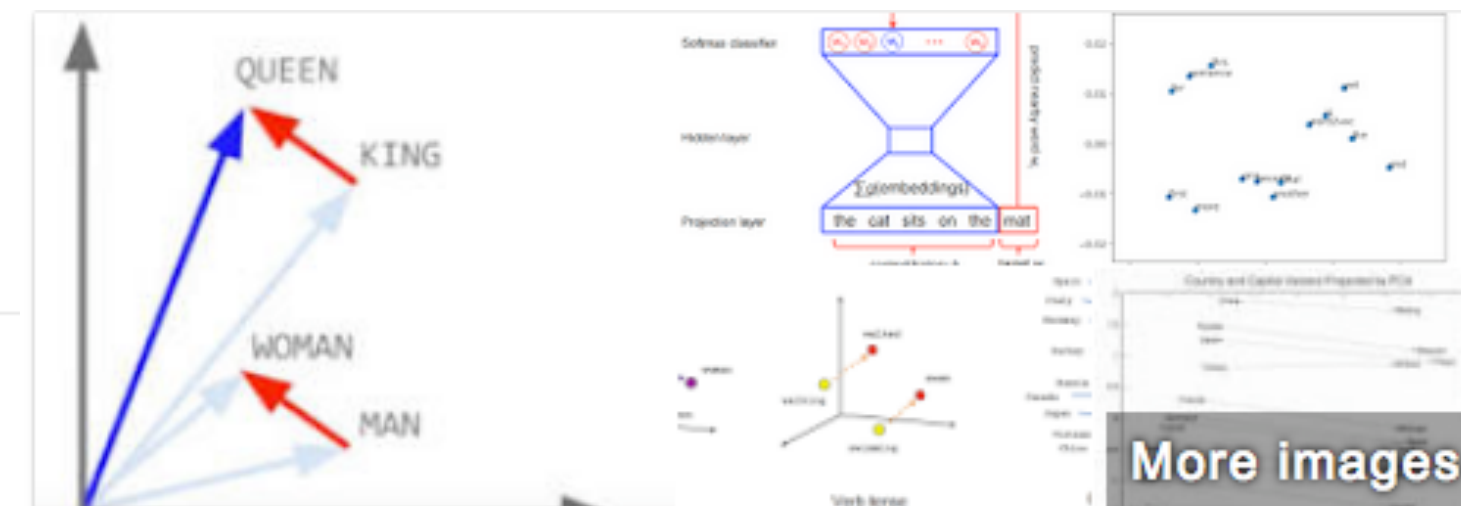
pourquoi les maths  
sont-elles  
intéressantes

## Chatbot



A chatbot is a computer program or an artificial intelligence which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test.

[Wikipedia](#)



## Word2vec



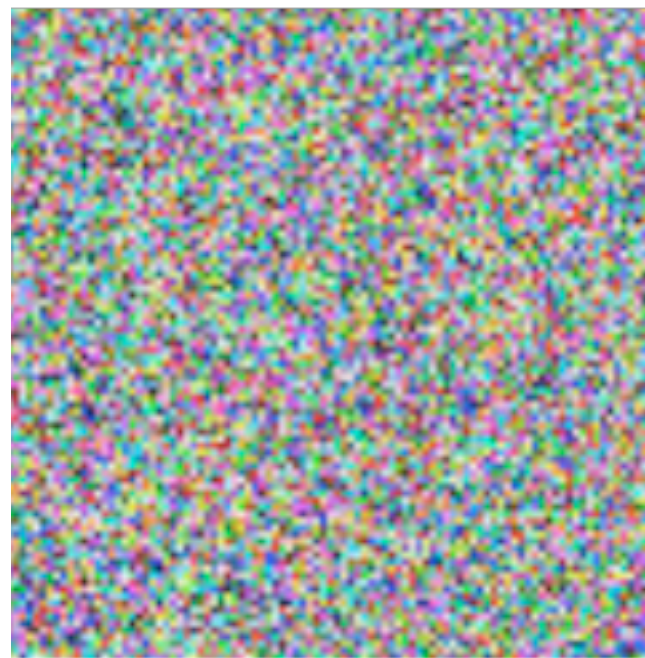
Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. [Wikipedia](#)



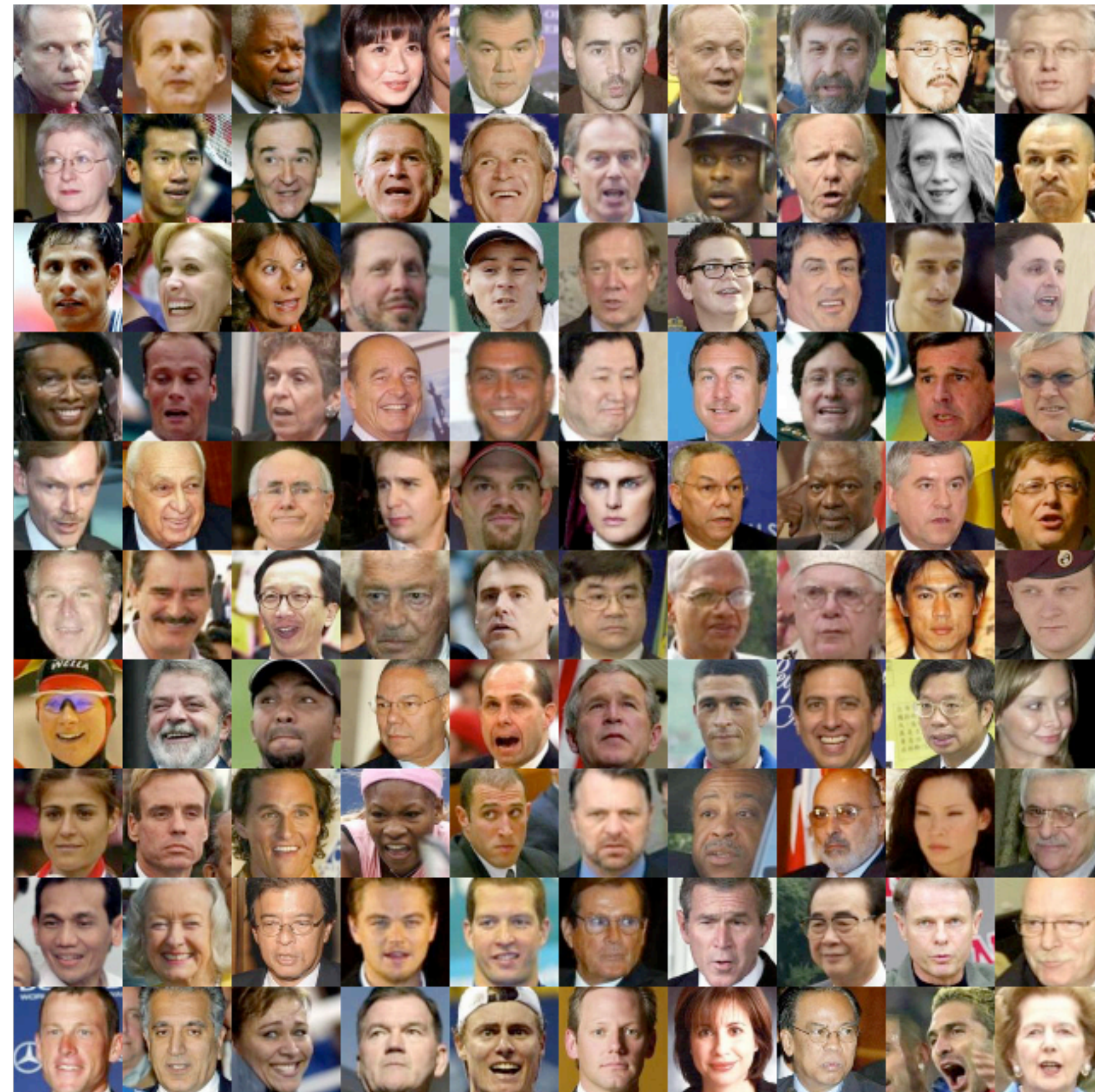
# Generative Networks (GANs)

Wasserstein GANs: optimal transportation (OT) mapping between random noise (Gaussians) and target distribution of images

Noise  $\sim N(0,1)$



Generative  
Model



Related math: Optimal Transportation algorithms and convergence (Peyre-Cuturi)



# Squeeze Nets

**SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size**

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer

Inference (evaluating the data and assigning a label) is costly (typically 0.1 second on a power hungry high memory GPU) in terms of

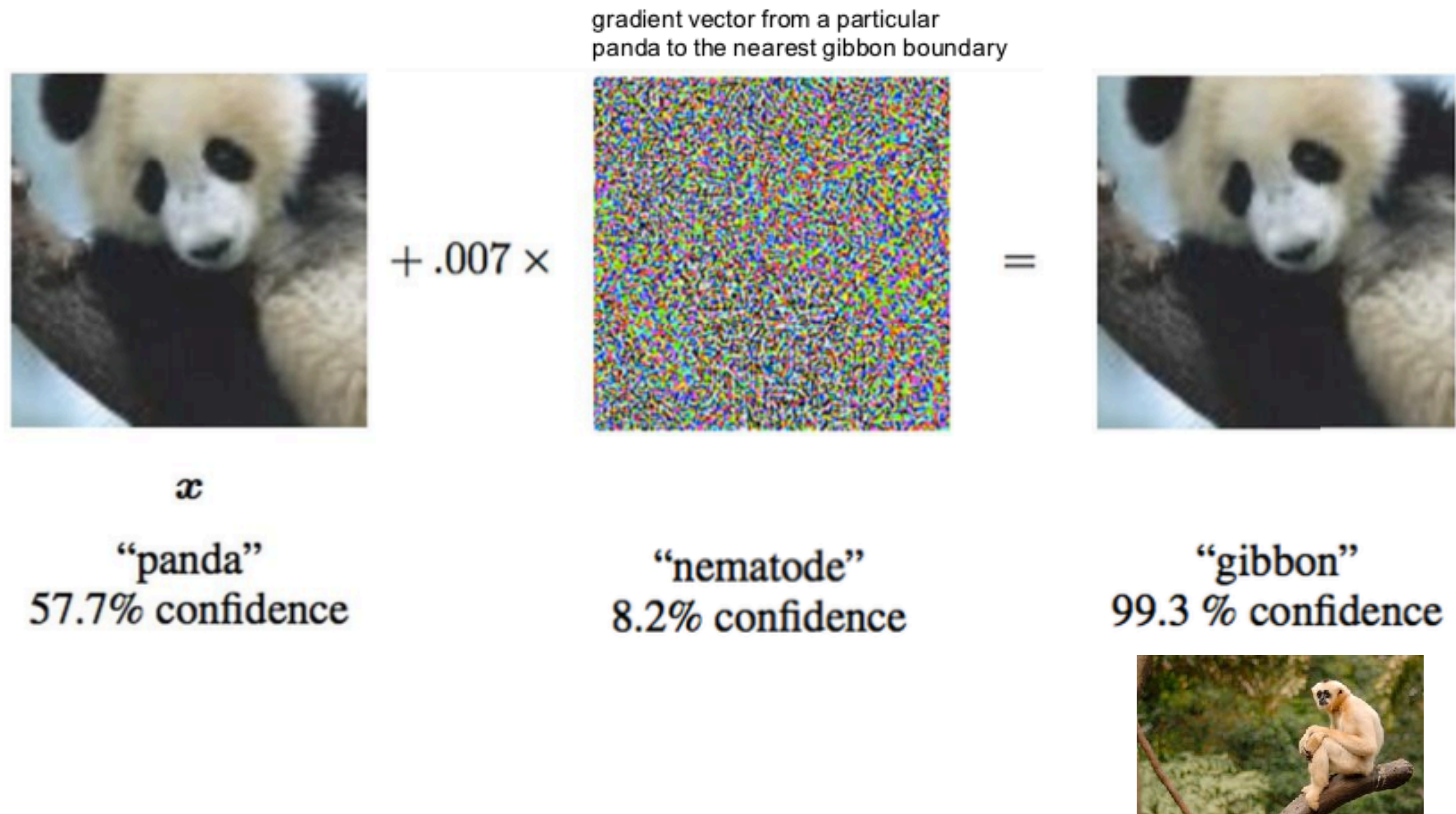
- Memory (to store the weights)
- Computation (multiplying the matrices times the vectors)
- Power (the energy Joules used by the chip)
- Time

Research effort to make lean NN. How?

- Quantization: low bit number representation and arithmetic. (related math : non-smooth optimization, when the ReLu are also quantized)
- Pruning: trim off the small weights, and retrain
- Hyperparameter Optimization: train over multiple architectures and params

Mostly engineering effort, but could be combined with more math on the training.

# Adversarial Examples



Goodfellow, Explaining and Harnessing Adversarial Examples, 2015



# ML theory for generalization

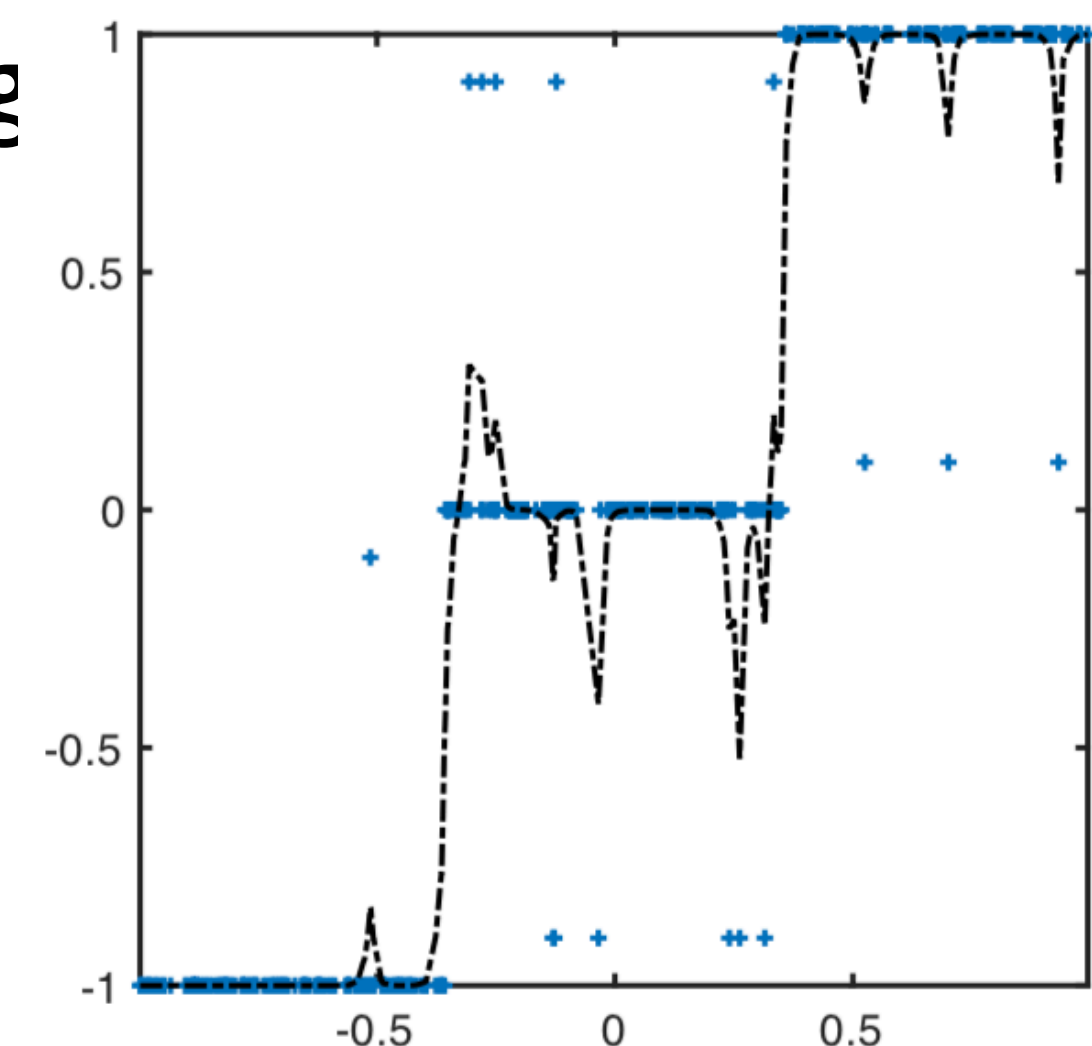
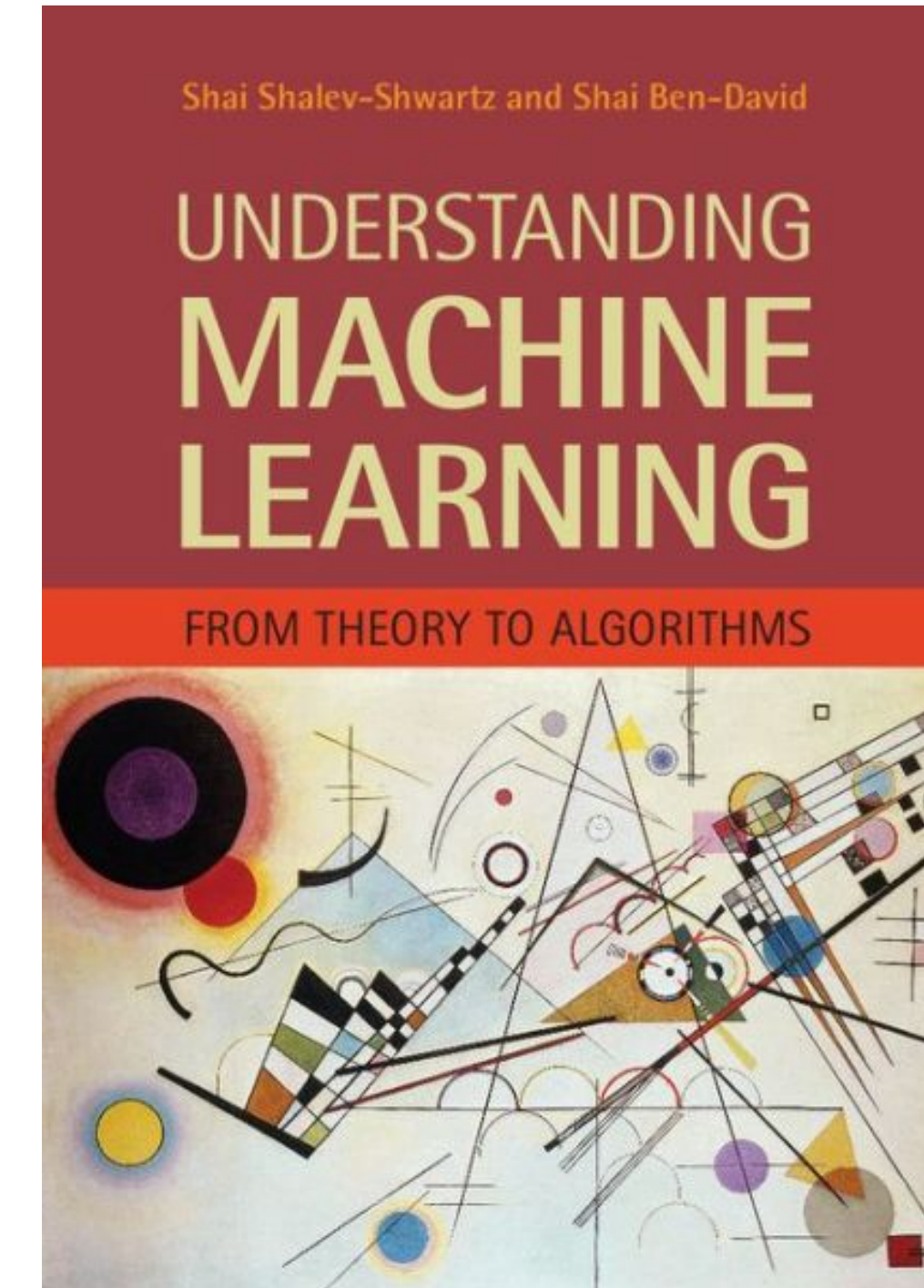
- Traditional ML theory (in math language)
  - the class of functions used for approximation has some regularity built in
  - the function to be approximated is regular
  - obtain a convergence rate for approximate based on number of samples

## ML theory breaks down for DNNs

Zhang (2016) “Understanding deep learning requires rethinking generalization” shows that ML theory does not apply

Learning networks. Two things to make clear to the reader (1) We don't know how Deep Learning works and (2) when it makes a prediction, we don't have an explanation why it arrived at that prediction. That is just scratching the

Bengio “Dark Art”. Popular press discusses lack of understanding





# Training and generalization

Generalization: training error is a good estimate of expected error on unseen images drawn from the same distribution.

DNNs generalize well in practise, but, in contrast to traditional ML techniques, there is no proof.

**1.1. Related work and applications.** Generalization bounds have been obtained previously by using the Lipschitz constant of a network ([Bartlett, 1997](#)), as well as by using more general stability results ([Bousquet & Elisseeff, 2002](#)). More recently, ([Bartlett et al. , 2017](#)) proposed the Lipschitz constant of the network as a candidate measure for the Rademacher complexity, which a measure of generalization ([Shalev-Shwartz & Ben-David, 2014](#), Chapter 26). However, our analysis is more direct and self-contained, and unlike other recent contributions such as ([Hardt et al. , 2015](#)), it does not depend on the training method.



# Our approach: PDE and Variational

**Optimization** : training of the network: large scale, nonconvex optimization.

- First order methods (too big for anything else).
- Stochastic gradient descent (too big for full gradients)
- *Deep Relaxation: partial differential equations for optimizing deep neural networks*  
Pratik Chaudhari, Adam M. Oberman, Stanley Osher, Stefano Soatto, Guillaume Carlier 2017

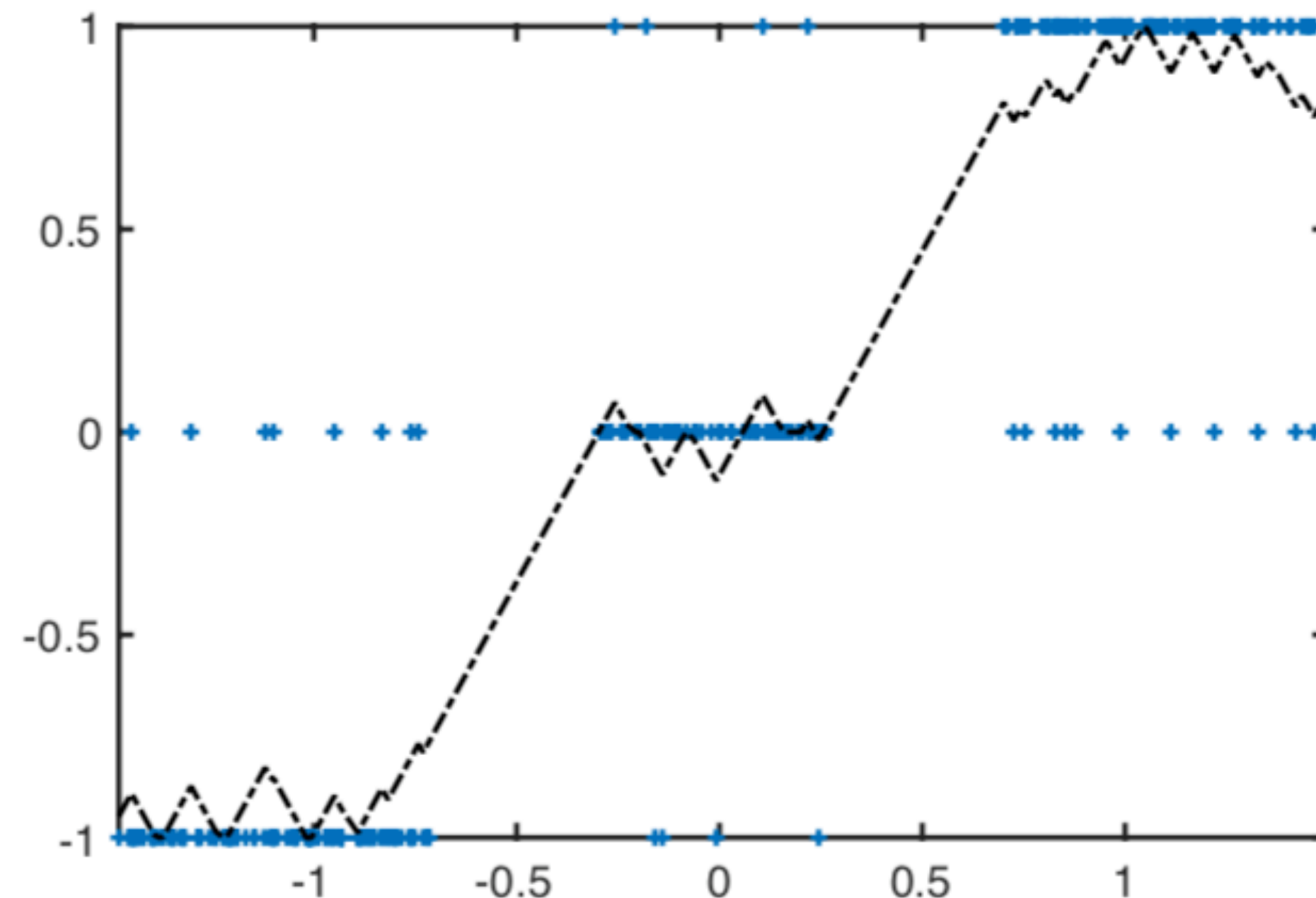
**Variational Problems and regularization** : improve the quality of solutions: better robustness to adversarial training, better predictions (generalization)

- *Lipschitz regularized Deep Neural Networks converge and generalize* O. and Jeff Calder; 2018
- Improved robustness to adversarial examples using Lipschitz regularization of the loss Chris Finlay, O., Bilal Abbasi;



## Our theory resolves the Zhang example

- We give a proof of generalization that gets around this obstruction, by adding a regularization term
- It resolves the problem posed by Zhang, by showing that random perturbations affect the Lipschitz constant of the data.
- Moreover, the method could *correct errors in the labels*.





# Approaches to regularization

- A. Machine Learning: learn data using an appropriate (smooth) parameterized class of functions  $\min_w \mathbb{E}_{x \sim \rho} \ell(f(x; w), y(x))$
- B. Algorithmic: use an algorithm which selects the best solution (e.g. Stochastic Gradient Descent as a regularizer, adversarial training)  $w^{k+1} = w^k + h_k \nabla_{mb} \ell(\dots w)$
- C. Inverse problems: allow for a broad class of functions, but modify the loss to choose the right one  $\min_w \mathbb{E}_{x \sim \rho} \ell(f(x; w), y(x)) + \lambda \|\nabla_x f\|_{L^p(X, \rho(x))}$



## Convergence result:

### Data distribution and loss function

**Definition 1.1.** Assume the data is normalized so that the data space is  $X = [0, 1]^d$ . Write  $\mathcal{D}_n = x_1, \dots, x_n$  for the training data. Assume  $\mathcal{D}_n$  is a sequence of *i.i.d.* random variables on  $X$  sampled from the probability distribution  $\rho$ . We consider the classification problem with  $m$  labels which are imbedded into the probability simplex, the label space,  $Y \subset \mathbb{R}^m$ . Write  $u_0 : X \rightarrow Y$  for the map from data to label space, so that  $y_i = u_0(x_i)$ .

**Assumption 2.3** (Loss function). The function  $\ell : Y \times Y \rightarrow \mathbb{R}$  is a *loss function* if it satisfies (i)  $\ell \geq 0$ , (ii)  $\ell(y_1, y_2) = 0$  if and only if  $y_1 = y_2$ , and (iii)  $\ell$  is strictly convex in  $y_1$ .

*Example 2.4* ( $\mathbb{R}^m$  with  $L^2$  loss). Set  $Y = \mathbb{R}^m$ , and let each label be a basis vector. Set  $\ell(y_1, y_2) = \|y_1 - y_2\|_2^2$  to be the  $L^2$  loss.

*Example 2.5* (Classification). In classification problems, the output of the network is a probability vector on the labels. Thus  $Y = \Delta_p$ , the  $p$ -dimensional probability simplex, and each label is mapped to a basis vector. The cross-entropy loss is given by  $\ell^{KL}(y, z) = -\sum_{i=1}^p z_i \log(y_i/z_i)$ . For labels,  $\ell^{KL}(y, e_k) = -\log(y_k)$ .



# Lipschitz Regularization of DNNs

Augment the expected loss function on the data with a Lipschitz regularization term

$$J^{Lip,n}[f] = \mathbb{E}_{(x,y) \sim \mathcal{D}_n} [\ell(f(x), u_0(x))] + \lambda \max(\text{Lip}(f) - L_0, 0)$$

where  $L_0$  is Lipschitz constant of the data, and  $n$  is number of data points.

\* $\lambda = 0$  corresponds to the usual unregularized problem.

Theory [Bartlett]:

- if you can control the Lipschitz constant, then generalization follows.

(but no indication how to do it).

Recent work by several authors attempted to control the Lipschitz constant of network, but the implementation was not effective.

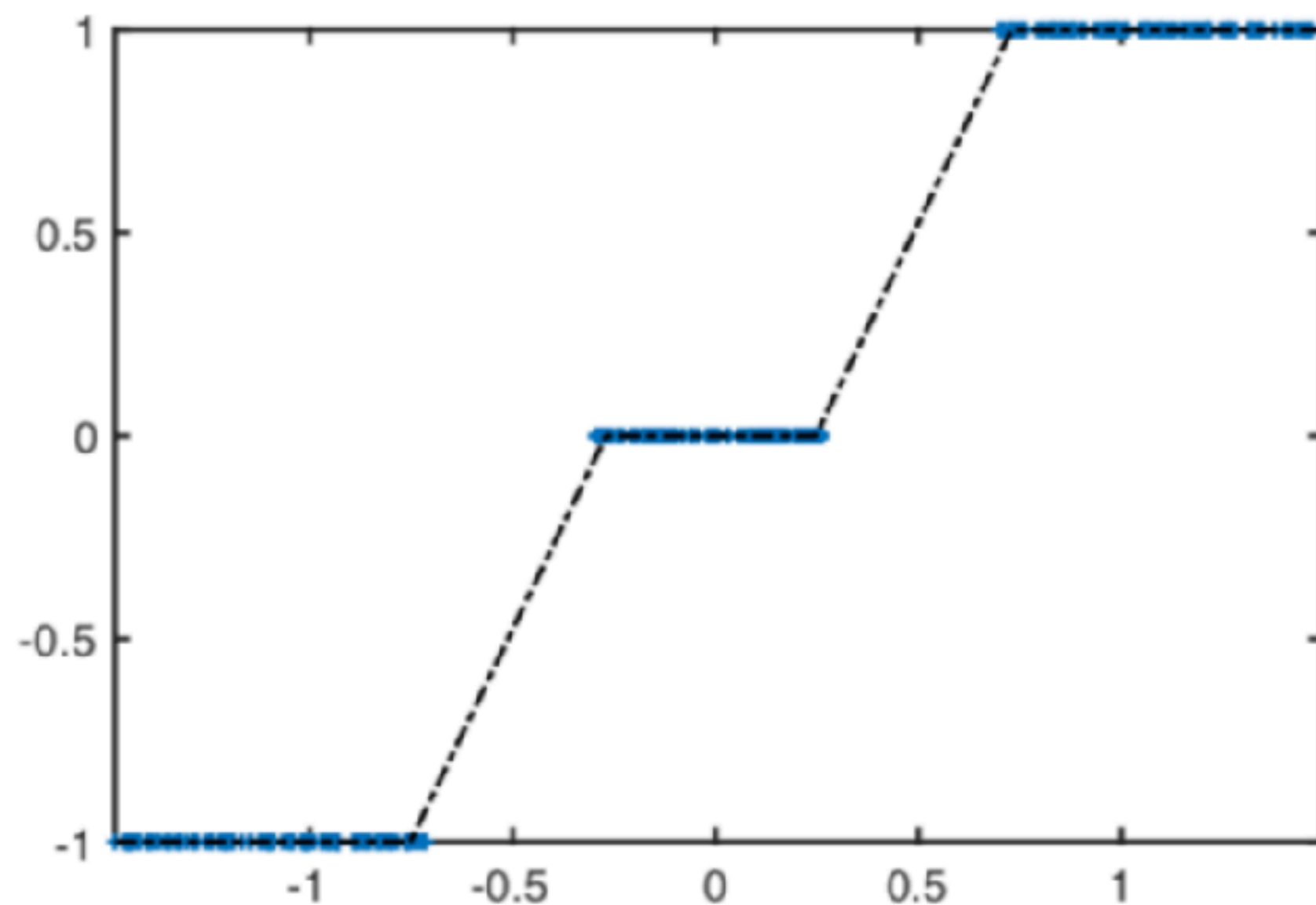
Our work: (i) adding Lipschitz regularization term leads to convergence proof  
(ii) effective implementation of Lipschitz regularization in practice.



## Convergence: two cases

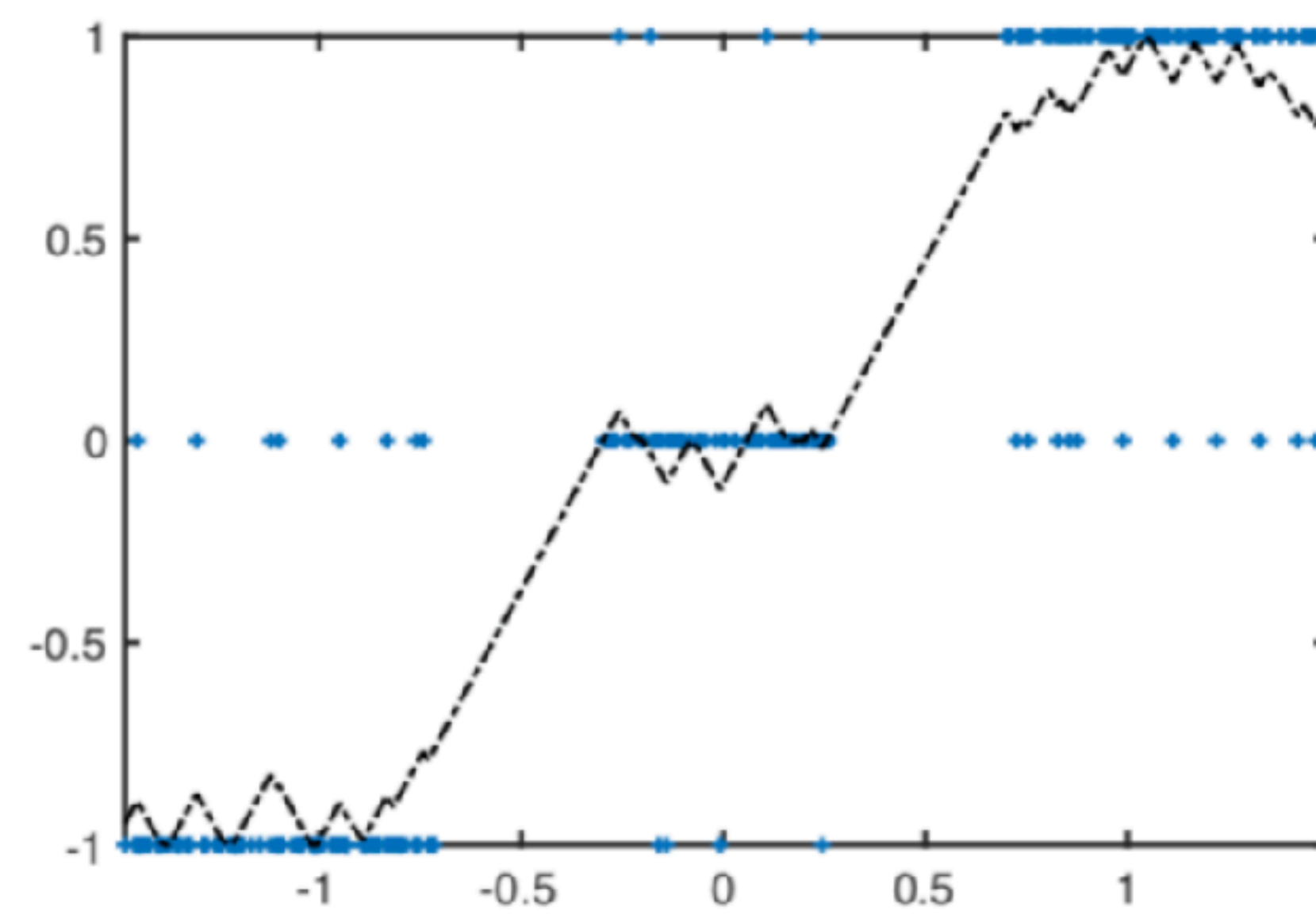
### Clean Labels:

- relevant in benchmark data sets and some applications,
- simpler proof, since the functional has value zero.
- regime of perfect data interpolation possible with DNNs



### Noisy labels:

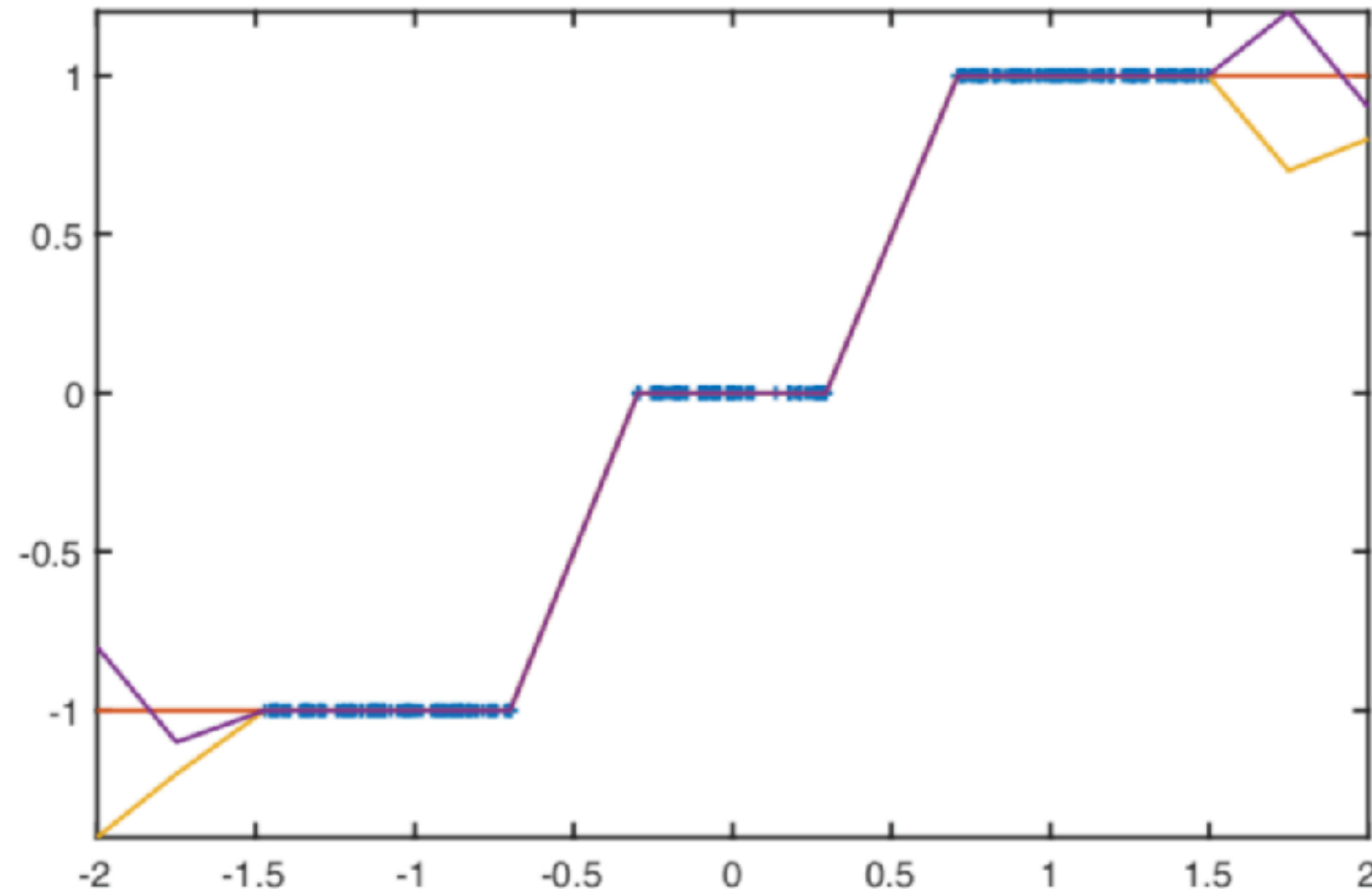
- relevant in applications,
- familiar setting for calculus of variations





# Convergence is on the data manifold

The convergence is on the data manifold,  
(the support of the data probability density function).  
Off the manifold, sequences may not converge,  
but the functions are still Lipschitz.



# Convergence theorem for Noisy Labels

**Theorem 2.11.** *Suppose that  $\inf_{\mathcal{M}} \rho > 0$ ,  $\ell : Y \times Y \rightarrow \mathbb{R}$  is Lipschitz, and let  $u^* \in W^{1,\infty}(X; Y)$  be any minimizer of (5). Then with probability one*

$$(7) \quad u_n \longrightarrow u^* \quad \text{uniformly on } \mathcal{M} \text{ as } n \rightarrow \infty,$$

*where  $u_n$  is any sequence of minimizers of (1). Furthermore, every uniformly convergent subsequence of  $u_n$  converges on  $X$  to a minimizer of (5).*

The proof of Theorem 2.11 requires a preliminary Lemma. Let  $H_L(X; Y)$  denote the collection of  $L$ -Lipschitz functions  $w : X \rightarrow Y$ .

**Lemma 2.12.** *Suppose that  $\inf_{\mathcal{M}} \rho > 0$ , and  $\dim(\mathcal{M}) = m_0$ . Then for any  $t > 0$*

$$(8) \quad \sup_{w \in H_L(X; Y)} \left| \frac{1}{n} \sum_{i=1}^n w(x_i) - \int_{\mathcal{M}} w \rho dVol(x) \right| \leq CL \left( \frac{t \log(n)}{n} \right)^{\frac{1}{m_0+2}}$$

*holds with probability at least  $1 - 2t^{-\frac{m_0}{m_0+2}} n^{-(ct-1)}$ .*

The estimate (8) is called a discrepancy result (Talagrand, 2006; Györfi *et al.*, 2006), and is a uniform version of concentration inequalities. We include a simple proof in Section 3.2.



## Proof of convergence: Clean Labels

**Theorem 2.7** (Convergence for clean labels). *Suppose that  $\text{Lip}[u_0] \leq L_0$  and  $\inf_{x \in \mathcal{M}} \rho(x) > 0$ . If  $f_n \in W^{1,\infty}(X; Y)$  is any sequence of minimizers of*

$$J^{\text{Lip},n}[f] = \mathbb{E}_{(x,y) \sim \mathcal{D}_n} [\ell(f(x), u_0(x))] + \lambda \max(\text{Lip}(f) - L_0, 0)$$

*then for any  $t > 0$*

$$\|u_0 - f_n\|_{L^\infty(\mathcal{M}; Y)} \leq CL_0 \left( \frac{t \log(n)}{n} \right)^{1/m}$$

*holds with probability at least  $1 - Ct^{-1}n^{-(ct-1)}$ .*

- Rate of convergence, on the data manifold, of the minimizers.
- The rate depends on,  $n$ , the number of data points sampled and,  $m$ , the number of labels.
- Probabilistic bound, where obtain a given error with high probability
- with uniform sampling the log term and the probability goes away

## Proof

**Lemma 2.9.** *Suppose that  $\inf_{\mathcal{M}} \rho > 0$ . Then for any  $t > 0$*

$$\|Id - \sigma_n\|_{L^\infty(\mathcal{M}; X)} \leq C \left( \frac{t \log(n)}{n} \right)^{1/m}$$

*with probability at least  $1 - Ct^{-1}n^{-(ct-1)}$ .*

We now give the proof of Theorem 2.7.

*Proof of Theorem 2.7.* Since  $J_n[u_n] = J_n[u_0] = 0$ , we must have  $\text{Lip}[u_n] \leq L_0$  and  $u_0(x_i) = u_n(x_i)$  for all  $1 \leq i \leq n$ . Then for any  $x \in X$  we have

$$\begin{aligned} \|u_0(x) - u_n(x)\|_Y &= \|u_0(x) - u_0(\sigma_n(x)) + u_0(\sigma_n(x)) - u_n(\sigma_n(x)) + u_n(\sigma_n(x)) - u_n(x)\|_Y \\ &\leq \|u_0(x) - u_0(\sigma_n(x))\|_Y + \|u_n(\sigma_n(x)) - u_n(x)\|_Y \\ &\leq 2L_0 \|x - \sigma_n(x)\|_X. \end{aligned}$$

Therefore, we deduce

$$\|u_0 - u_n\|_{L^\infty(\mathcal{M}; Y)} \leq 2L_0 \|Id - \sigma_n\|_{L^\infty(\mathcal{M}; X)}.$$

The proof is completed by invoking Lemma 2.9. □



## Generalization follows

As an immediate corollary, we can prove that the generalization loss converges to zero, and so we obtain perfect generalization.

**Corollary 2.8.** *Assume that for some  $q \geq 1$  the loss  $\ell$  satisfies*

$$(6) \quad \ell(y, y_0) \leq C \|y - y_0\|_Y^q \quad \text{for all } y_0, y \in Y.$$

*Then under the assumptions of Theorem 2.7*

$$L[u_n, \rho] \leq CL_0^q \left( \frac{t \log(n)}{n} \right)^{q/m}$$

*holds with probability at least  $1 - Ct^{-1}n^{-(ct-1)}$ .*

*Proof.* By (6), we can bound the generalization loss as follows

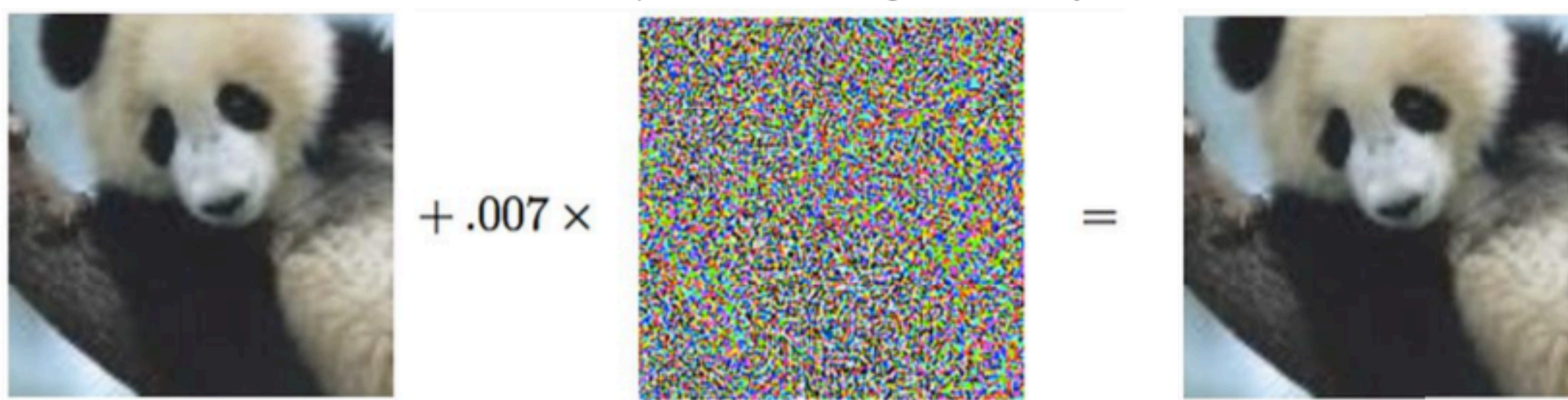
$$L[u_n, \rho] = \int_{\mathcal{M}} \ell(u_n(x), u_0(x)) dVol(x) \leq C Vol(\mathcal{M}) \|u_n - u_0\|_{L^\infty(\mathcal{M}; Y)}^q.$$

The proof is completed by invoking Theorem 2.7. □



# Regularization improves Adversarial Robustness

gradient vector from a particular panda to the nearest gibbon boundary


$$x + .007 \times \text{gradient vector} = \text{result}$$

$x$

“panda”  
57.7% confidence

“nematode”  
8.2% confidence

“gibbon”  
99.3 % confidence



# Adversarial Attacks

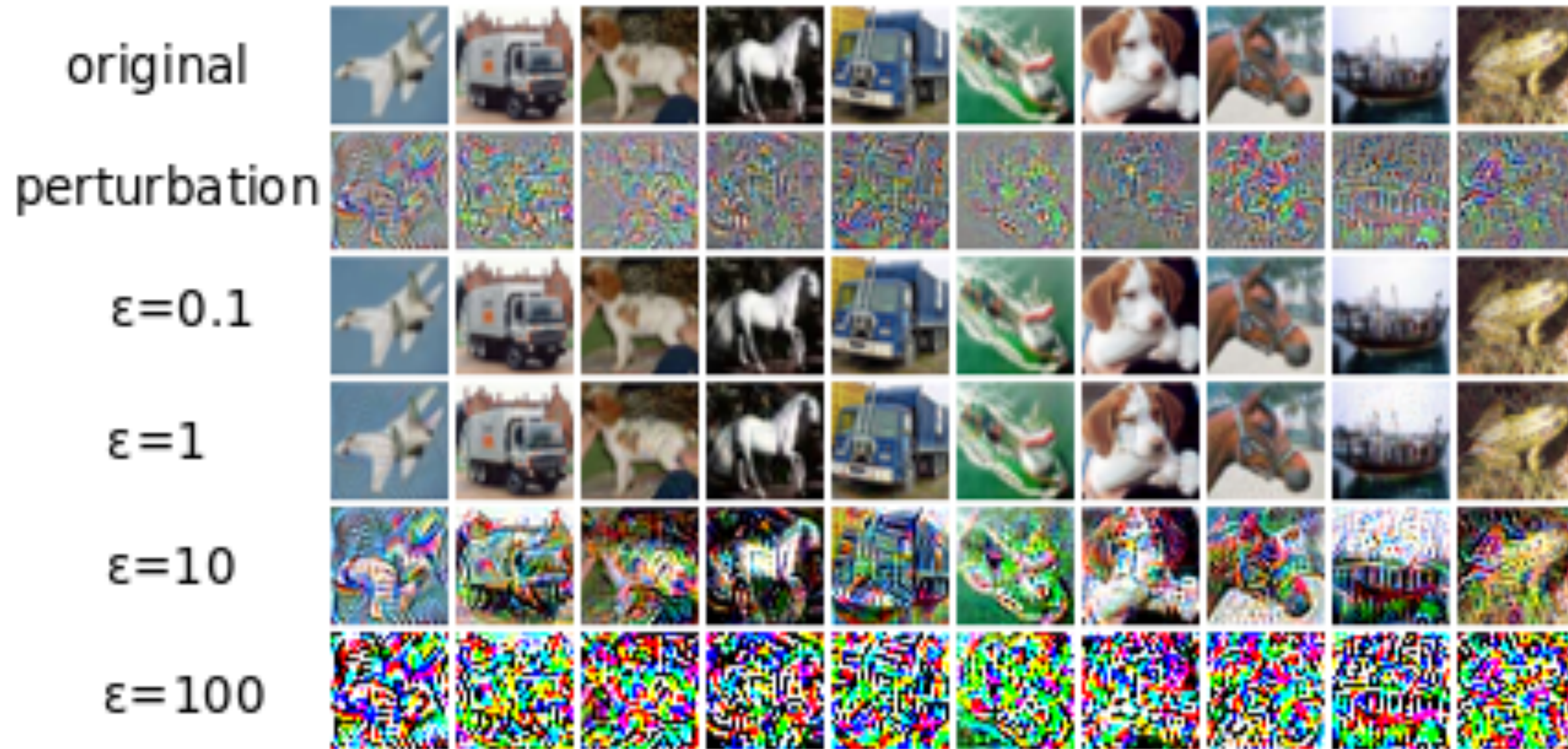
**Definition 2.1** (Adversarial attacks). Write  $c^*(x)$  for the correct label and  $c(x) = \arg \max_i f(x)_i$  for the classifier. An adversarial attack  $a = a(x)$ , is a perturbation of the input  $x$  which leads to incorrect classification  $c(x + a(x)) \neq c^*(x)$ .

Adversarial attacks seek to find the minimum norm attack vector, which is an intractable problem (Athalye et al., 2018). An alternative which permits loss gradients to be used, is to consider the attack vector of a given norm which most increases the loss,  $\ell$ .

$$(3) \quad \max_{\|a\| \leq \epsilon} \ell(f(x + a), y)$$



## Scale measures visible attacks

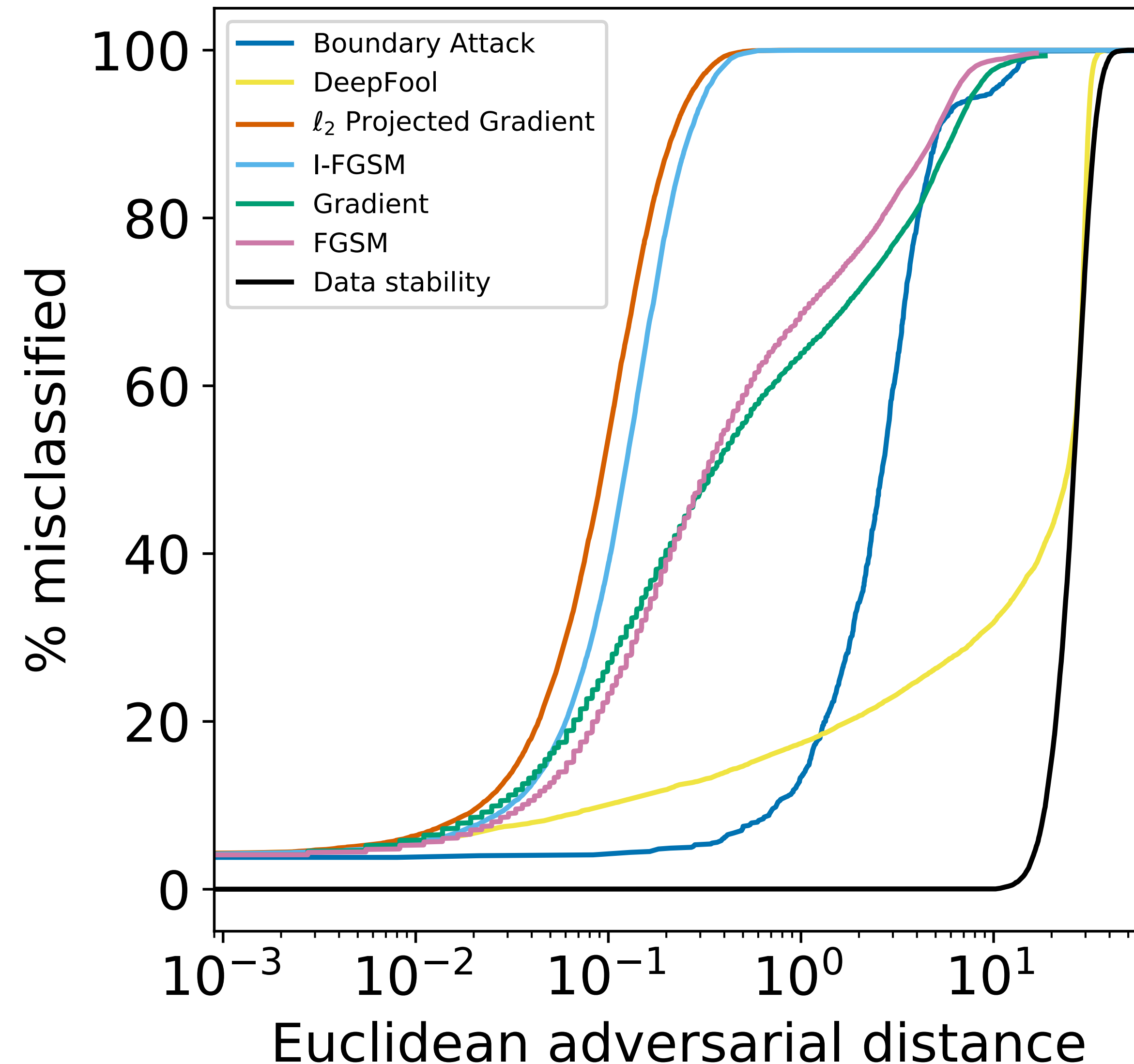


DNNs are vulnerable to attacks which are invisible to the human eye.  
Undefended networks have 100% error rate at .1 (in max norm)



# Arms race of attack methods and defences

ResNeXt34, CIFAR-10



We tested against toolbox of attacks. Plotted the error curve as a function of the adversarial size.

Strongest attacks:

1. Iterative  $\ell_2$ -projected gradient
2. Iterative Fast Gradient Signed Method (FGSM)

# Adversarial Training: interpretation as regularization

Write  $\ell(x) = \ell(f(x), u_0(x))$ .

Write  $L_{\ell \circ f}$  for the Lipschitz constant of loss of the model.

Adversarial training is an effective method for improving robustness to adversarial attacks. We show that adversarial training using the Fast Signed Gradient Method (Goodfellow et al., 2014) can be interpreted as *regularization* by the average of the 1-norm of the gradient of the loss over the data,

$$(J^1) \quad J^1[w] = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(x) + \varepsilon \|\nabla \ell(x)\|_1]$$

The choice of norm for the adversarial perturbation can lead to different interpretations: using the 2-norm for adversarial training corresponds to

$$(J^2) \quad J^2[w] = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(x) + \varepsilon \|\nabla \ell(x)\|_2]$$



# Dual norms and attacks

**2.1. Derivation of attack directions.** The solution of (3) can be approximated using the dual norm (Boyd & Vandenberghe, 2004, A.1.6). If the  $\infty$ -norm is used, we recover the Signed Gradient (Goodfellow et al., 2014). However a different attack vector is obtained if we measure attacks in the 2-norm.

**Theorem 2.2.** *The optimal attack vector defined by (3) in a generic norm  $\|\cdot\|$  can be approximated to  $\mathcal{O}(\varepsilon^2)$  with the vector  $\varepsilon a$ , where  $a$  is the solution of*

$$(4) \quad a \cdot v = \|v\|_*, \quad \text{with } v = \nabla_x \ell(f(x), y)$$

*and  $\|\cdot\|_*$  is the dual norm. In particular  $a$  is given by*

$$(5) \quad \begin{cases} a_i^{SG} = \frac{\nabla \ell(x)_i}{|\nabla \ell(x)_i|} & \text{for the } \infty\text{-norm} \\ a^{\ell_2} = \frac{\nabla \ell(x)}{\|\nabla \ell(x)\|_2} & \text{for the 2-norm} \end{cases}$$

# Adversarial Training augmented with Lipschitz Regularization

$$(J^{2-\text{Lip}}) \quad J^{2-\text{Lip}}[w] = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(x) + \varepsilon \|\nabla \ell(x)\|_2] + \lambda \max_{(x,y) \in \mathcal{D}} \|\nabla_x \ell(x)\|_2.$$

which we refer to as 2 – Lip (tulip). In practice,  $J^{2-\text{Lip}}$  outperforms  $J^2$  and  $J^1$ . For example on CIFAR-10, for a ResNeXt model, adversarial training alone reduced adversarial training error by 29% (measured at adversarial  $\ell_2$  distance<sup>1</sup>  $\varepsilon = 0.1$ ) over an undefended model. In contrast,  $J^2$  with Lipschitz regularization ( $J^{2-\text{Lip}}$ ) reduces adversarial error by 42% over baseline. See Table 1. We trained with

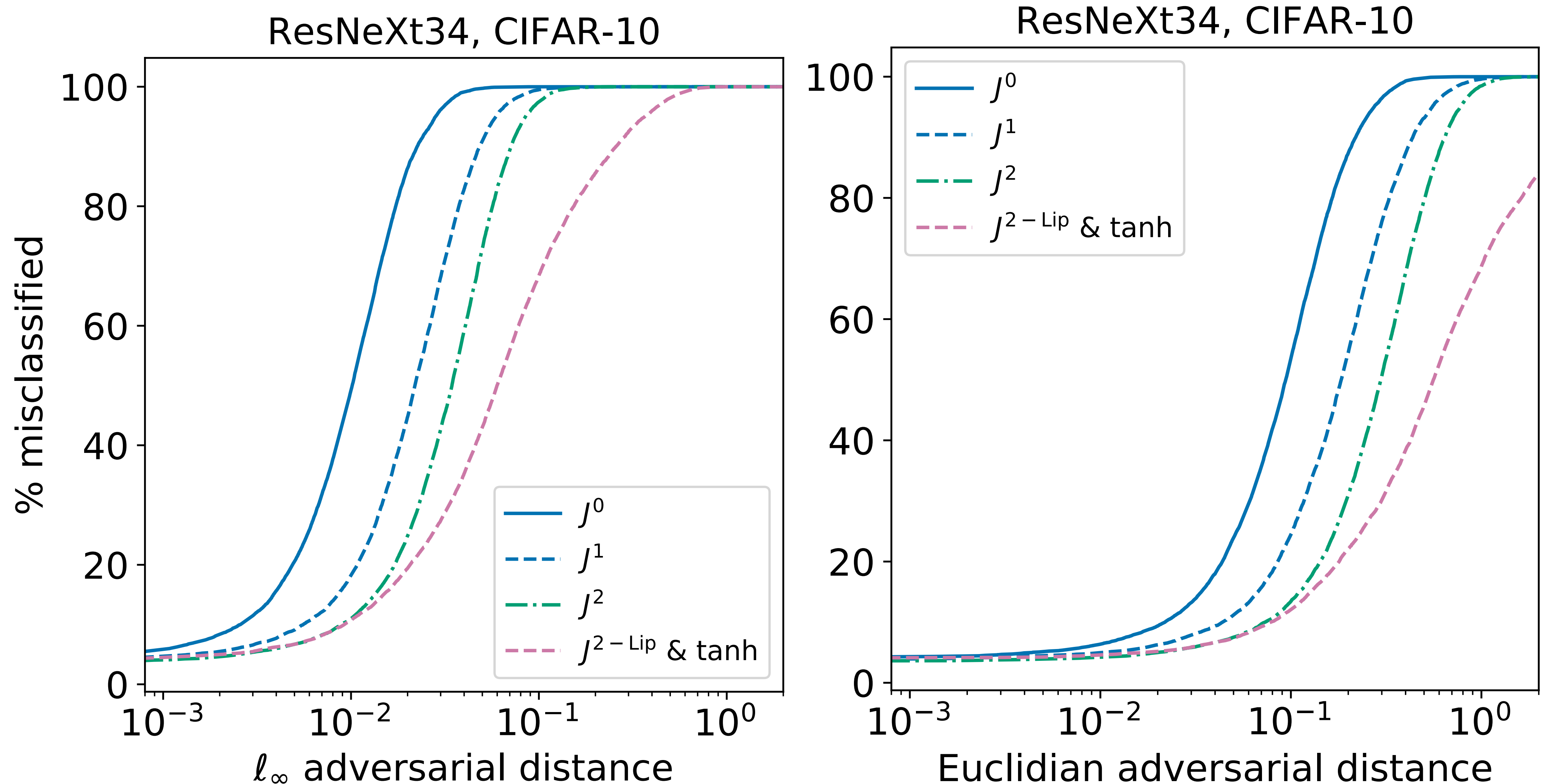


## AT + Tulip Results (2-norm)

Dataset	defense method	Euclidean distance		$\ell_\infty$ distance	
		median distance	% Err at $\varepsilon = 0.1$	median distance	% Err at $\varepsilon = 1/16$
CIFAR-10	$J^0$ (baseline)	0.09	53.98	1.02e−2	99.92
	$J^1$ (AT, FGSM)	0.18	24.63	2.12e−2	96.06
	$J^2$ (AT, $\ell_2$ )	0.30	13.54	3.45e−2	84.76
	$J^{2-\text{Lip}}$ & tanh	<b>0.56</b>	<b>12.12</b>	<b>6.00e−2</b>	<b>51.64</b>
CIFAR-100	$J^0$ (baseline)	4.74e−2	74.18	5.83e−3	99.61
	$J^1$ (AT, FGSM)	8.08e−2	56.34	1.07e−2	98.46
	$J^2$ (AT, $\ell_2$ )	8.61e−2	53.77	1.06e−2	98.03
	$J^{2-\text{Lip}}$ & tanh	<b>0.136</b>	<b>42.58</b>	<b>1.6e−2</b>	<b>93.73</b>

Significant improvement over state-of-the art results  
come from augmenting AT with Lipschitz regularization

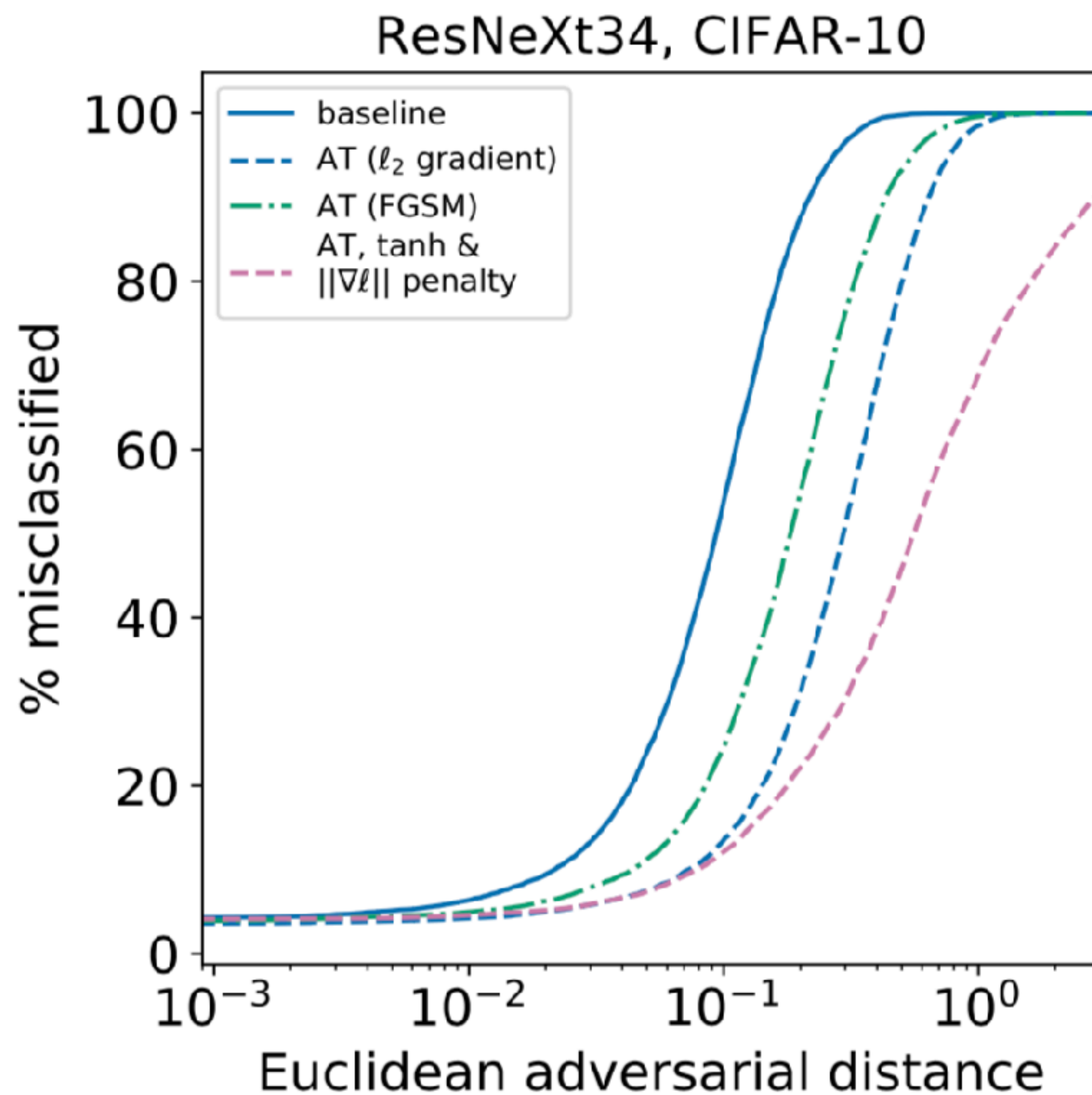
# AT + Tulip Results (2-norm vs max-norm)



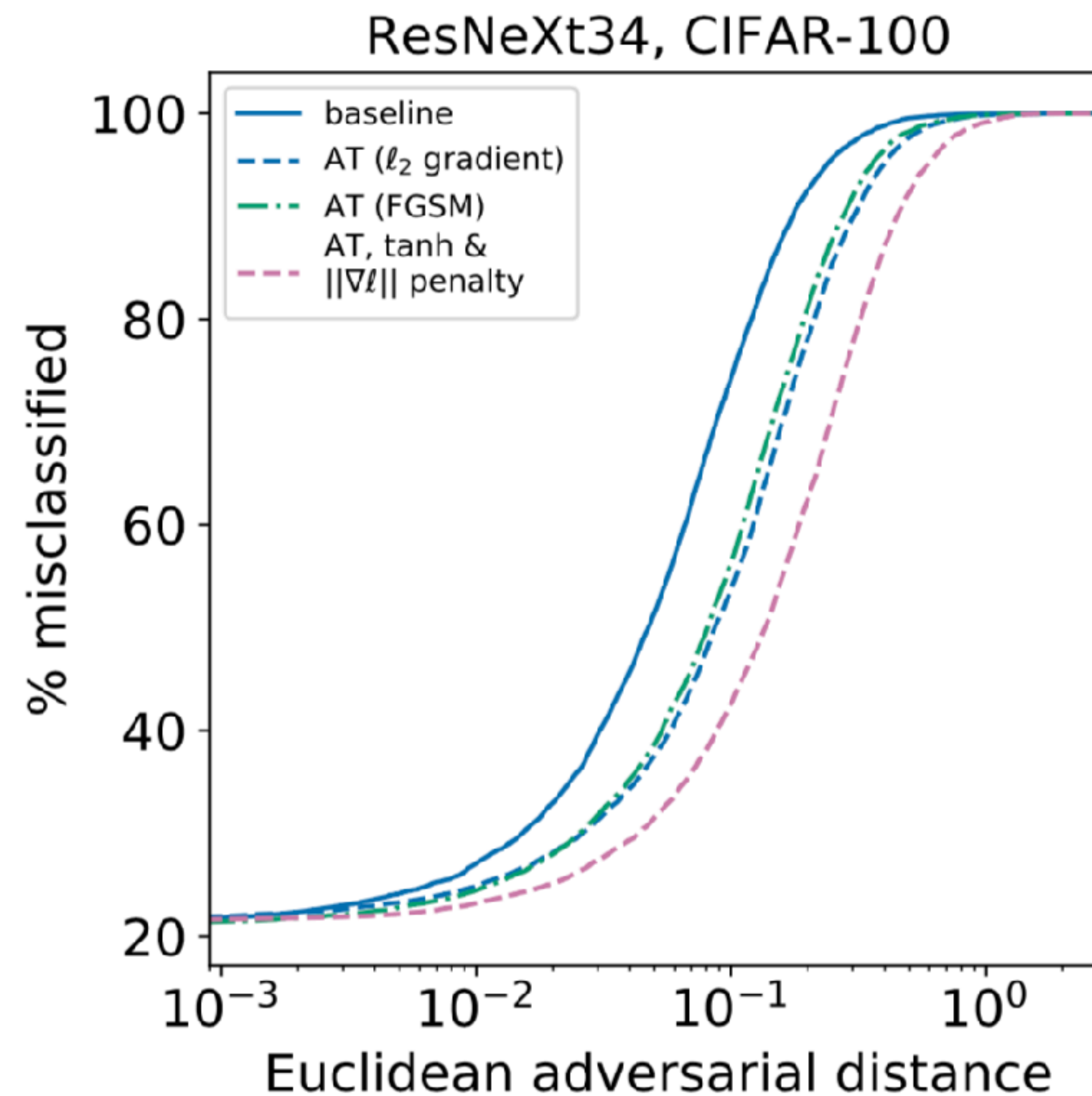
2-Lip > AT-2 > AT-1 > baseline (for all noise levels on both datasets)



# AT + Tulip Results (2-norm)



(A) CIFAR-10



(B) CIFAR-100