Partial Differential Equations Approaches to Optimization and Regularization of Deep Neural Networks

Celebrating 75 Years of Mathematics of Computation
ICERM
Nov 2, 2018

Adam Oberman
McGill Dept of Math and Stats

supported by NSERC, Simons Fellowship, AFOSR FA9550-18-1-0167

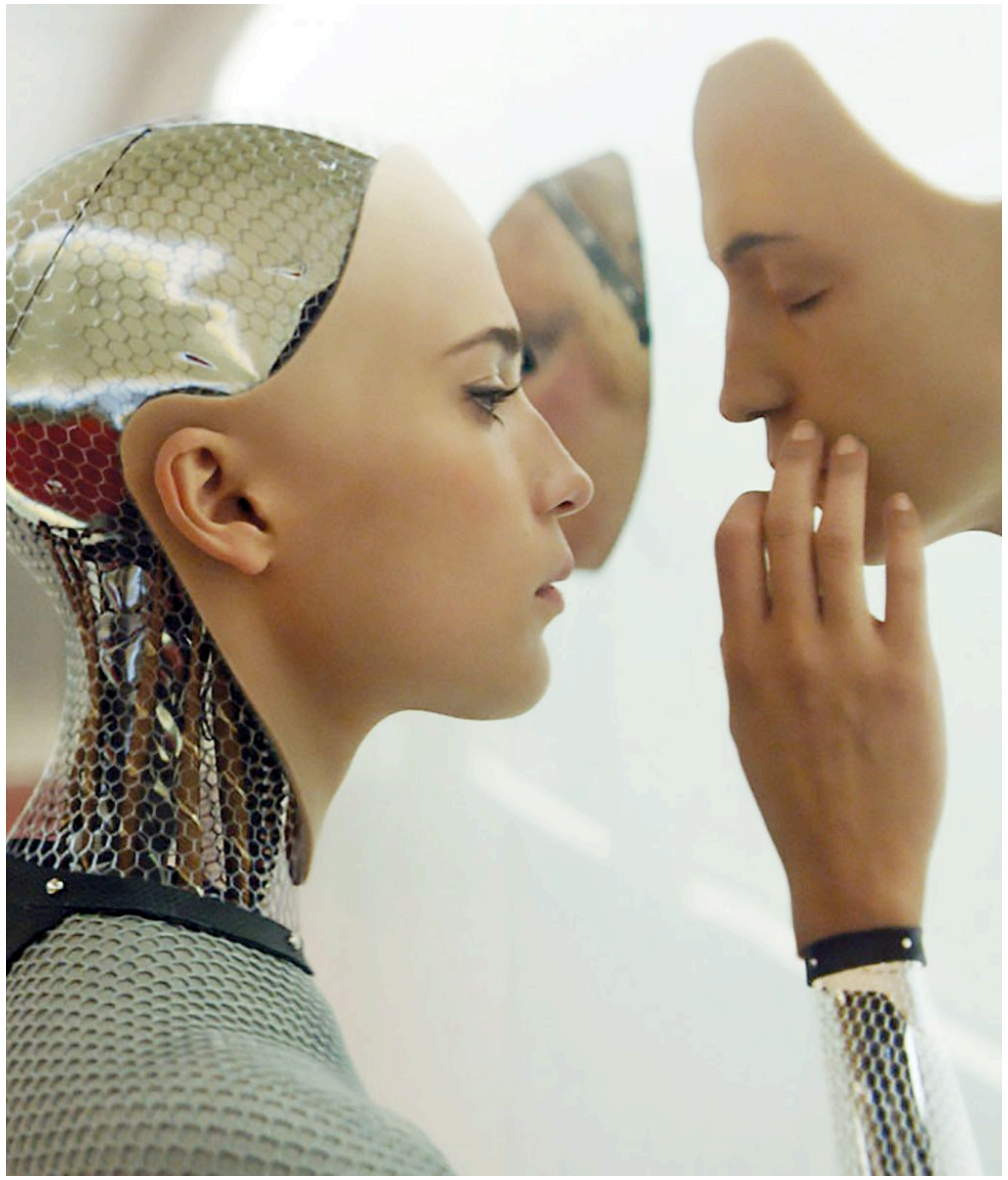
Background AI

- 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

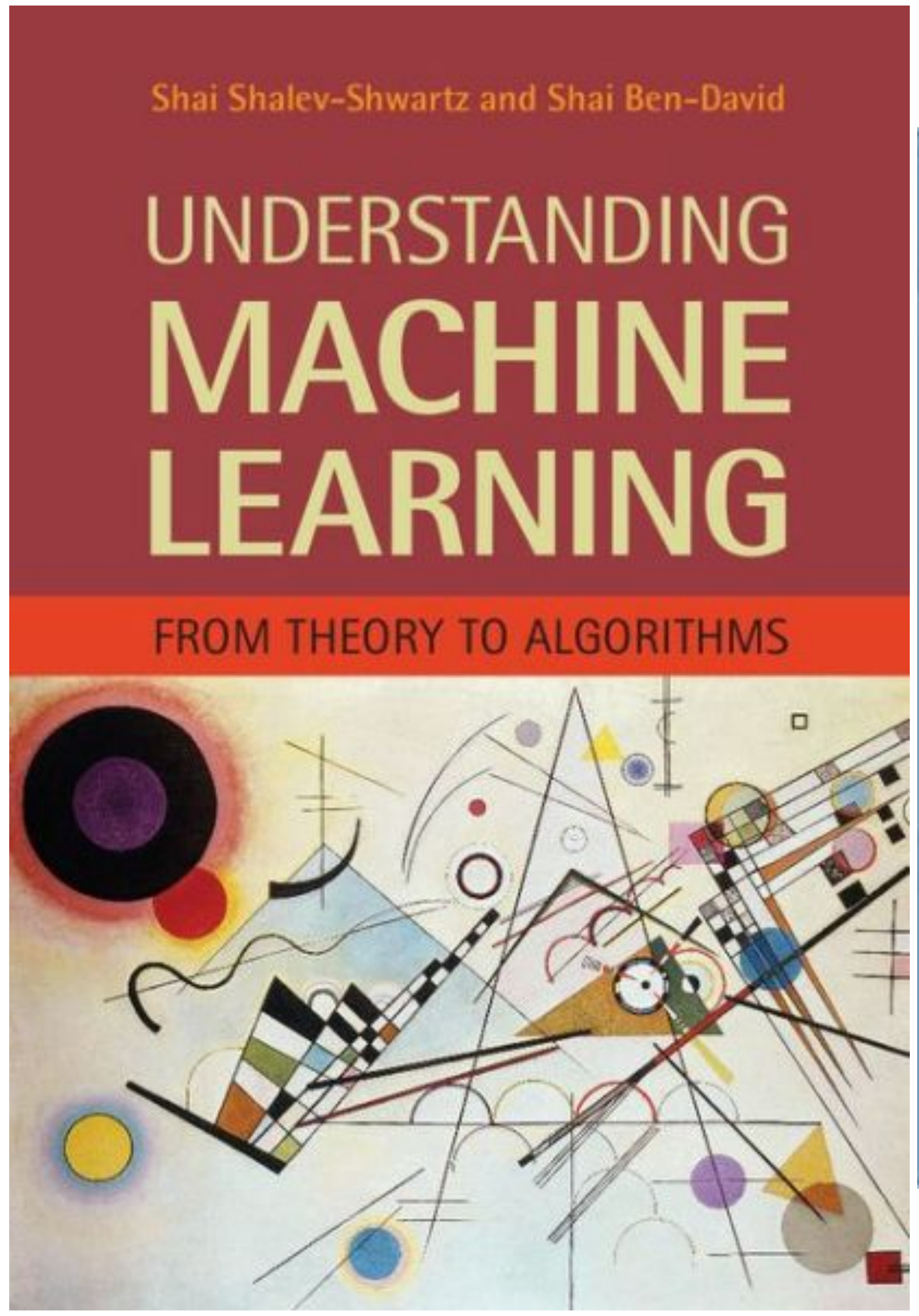
Artificial General Intelligence (AGI)

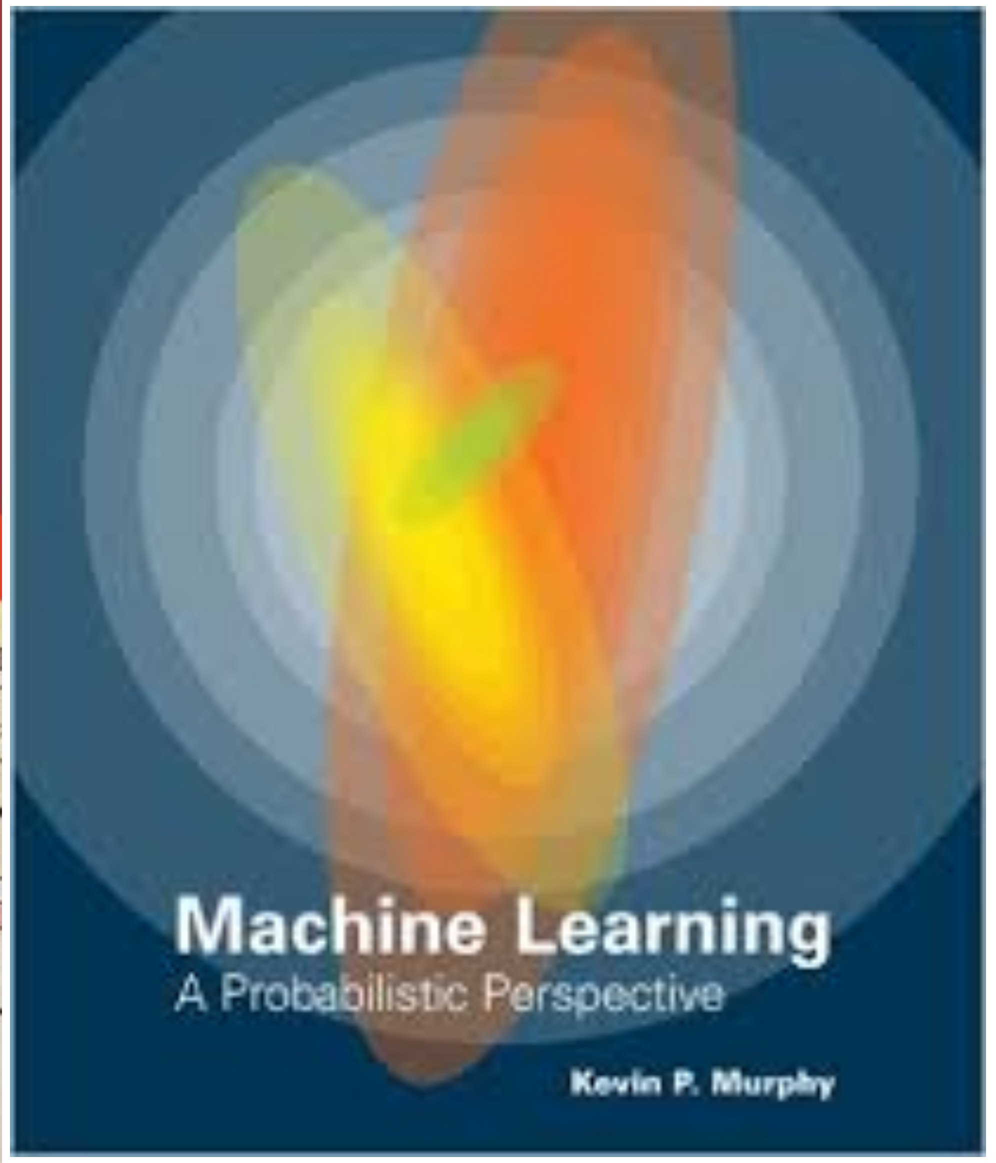
- AI : specific tasks,
- AGI : general cognitive abilities.
- AGI is a small research area within AI: build machines that can successfully perform *any* task that a human might do
- So far, no progress on AGI.



Deep Learning vs. traditional Machine Learning

- 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 *substantially larger* scale problems





What are DNNs (in Math Language)?

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

- ImageNet: Total number of classes: m =21841
- Total number of images: n =14,197,122
- Color images d= 3*256*256= 196,608

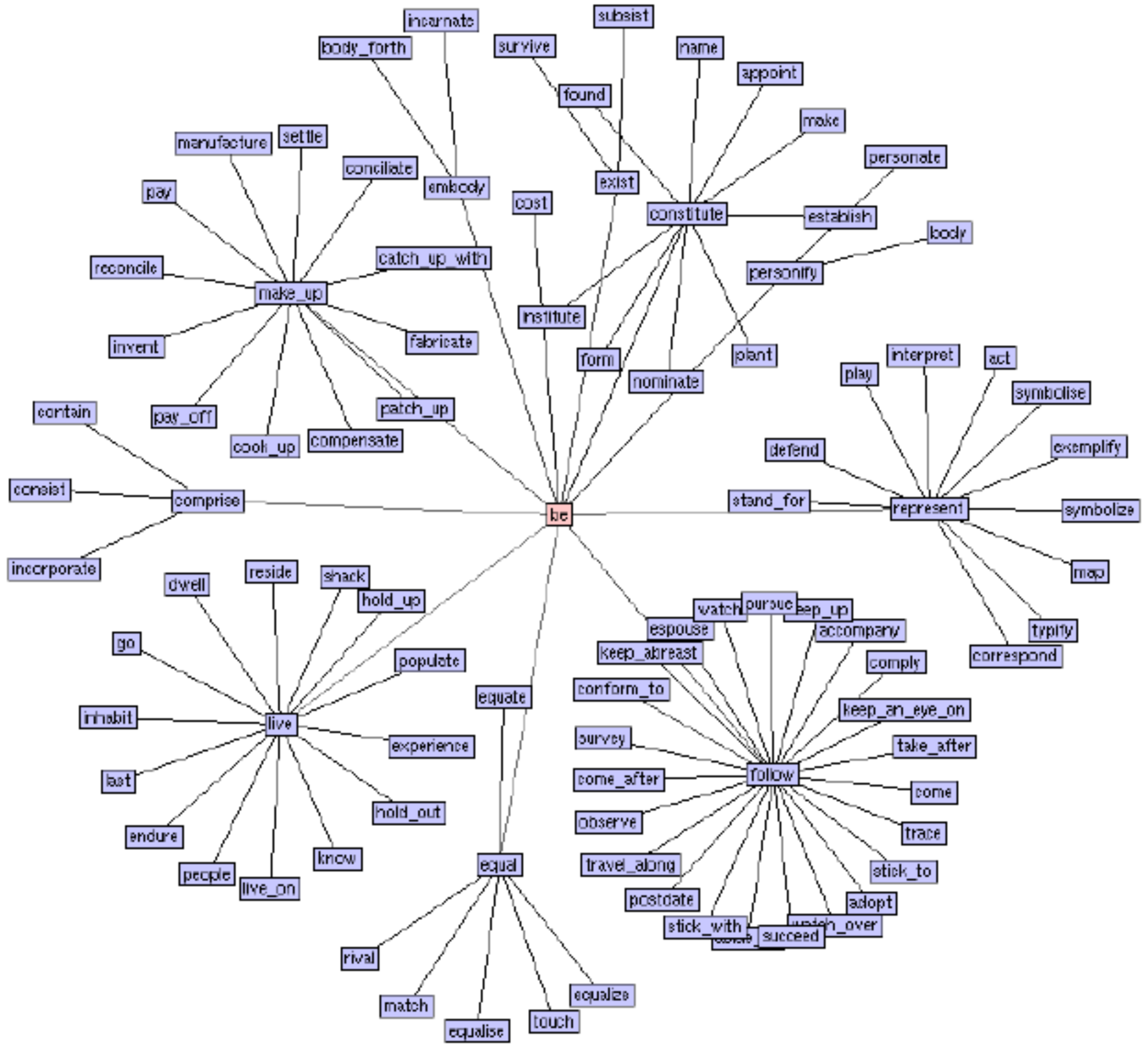
Facebook used 256 GPUs, working in parallel, to train ImageNet.

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

What is the function?
Looking for a map from images to labels

x in M = manifold of images

f(x)



graph of list of word labels

Doing the impossible?

In theory, due to curse of dimensionality, ***impossible*** to accurately interpolate a high dimensional function.

In practice, ***possible*** using Deep Neural Network architecture, training to fit the data with SGD. However we don't know why it works.

Can train a computer to caption images more accurately that human performance.

Figure 1. At the current state-of-art, more than 95% of images can be c

Loss Functions versus Error

Classification problem: map image to discrete label space {1,2,3,…,10}

In practice: map to a probability vector, then assign label of the arg max.

Classification is not differentiable, so instead, in order to train, use a *loss function* as a surrogate.

Assumption 2.3 (Loss function)**.** The function $\ell : Y \times Y \to \mathbb{R}$ 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) ℓ 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, $\ell^{KL}(y, e_k) = -\log(y_k)$.

DNNs in Math Language: high dimensional function fitting

$$\min_w \mathbb{E}_{x\sim\rho_n} \ell(f(x;w), y(x)) = \sum_{i=1}^{n} \ell(f(x_i;w), y(x_i))$$

Data fitting problem: f parameterized map from images to probability vectors on labels. y is the correct label. Try to fit data by minimizing the loss.

Training: minimize expected loss, by taking stochastic (approximate) gradients

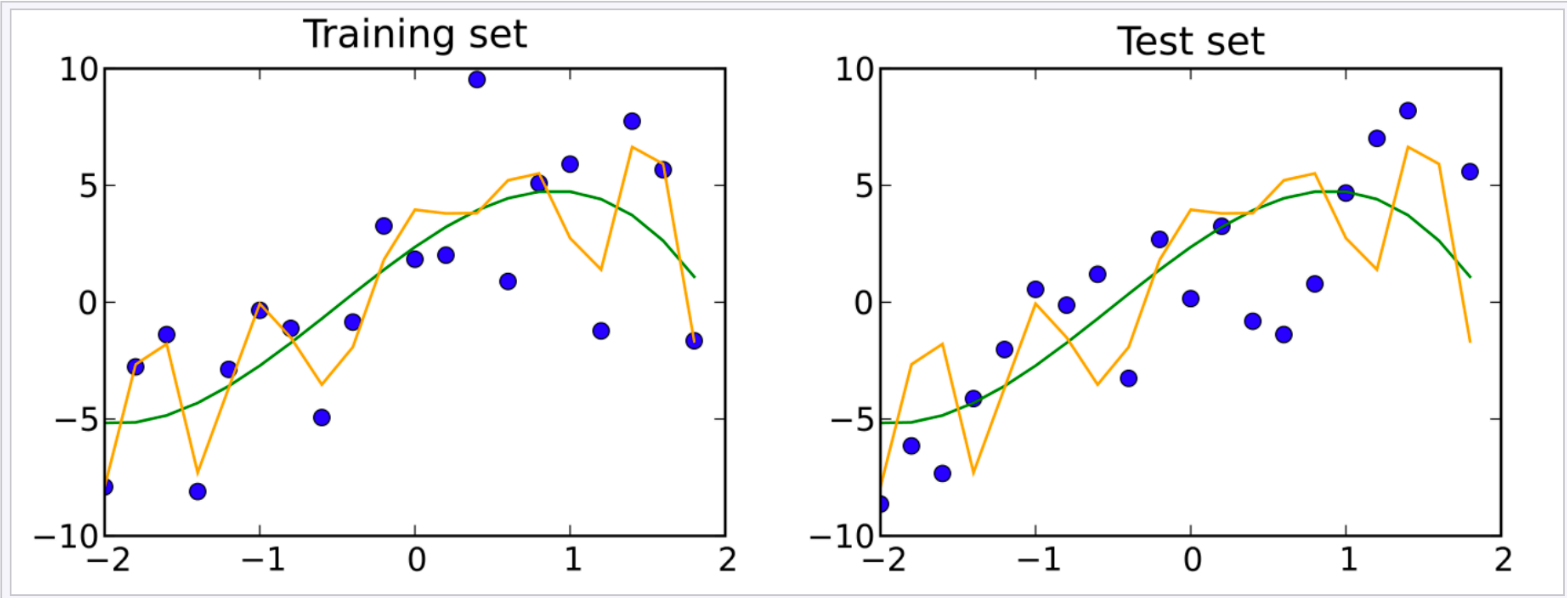
Note: train on an empirical distribution sampled from the density rho.

Generalization. Training set and test set

Goal: generalization: hope that training error is a good estimate of the generalization loss, which is the expected loss on unseen images drawn from the same distribution.

$$\mathbb{E}_{x\sim\rho}\ell(f(x;w),y(x)) = \int \ell(f(x_i;w),y(x_i))d\rho(x)$$

Testing: reserve some data and approximate the generalization loss/error on the test data, which is a surrogate for the true expected error on the full density.



Orange curve: overtrained. Green curve better generalization

$$\mathbb{E}_{x\sim\rho_{test}}\ell(f(x;w),y(x)) = \sum \ell(f(x_i;w),y(x_i))$$

Challenges for deep learning

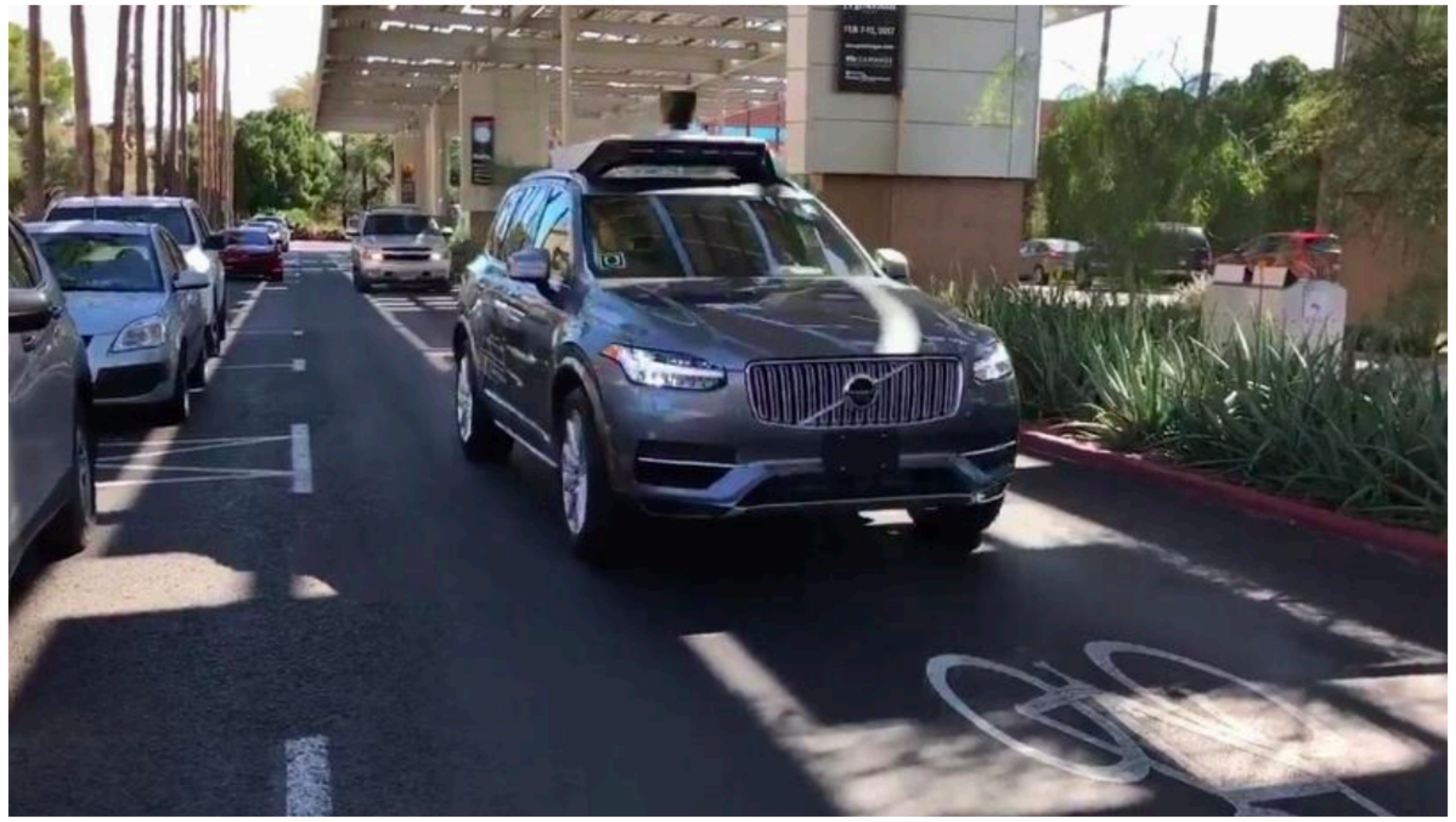
"It is not clear that the existing AI paradigm is immediately amenable to any sort of software engineering validation and verification. This is a serious issue, and is a potential roadblock to DoD's use of these modern AI systems, especially when considering the liability and accountability of using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

JASON report

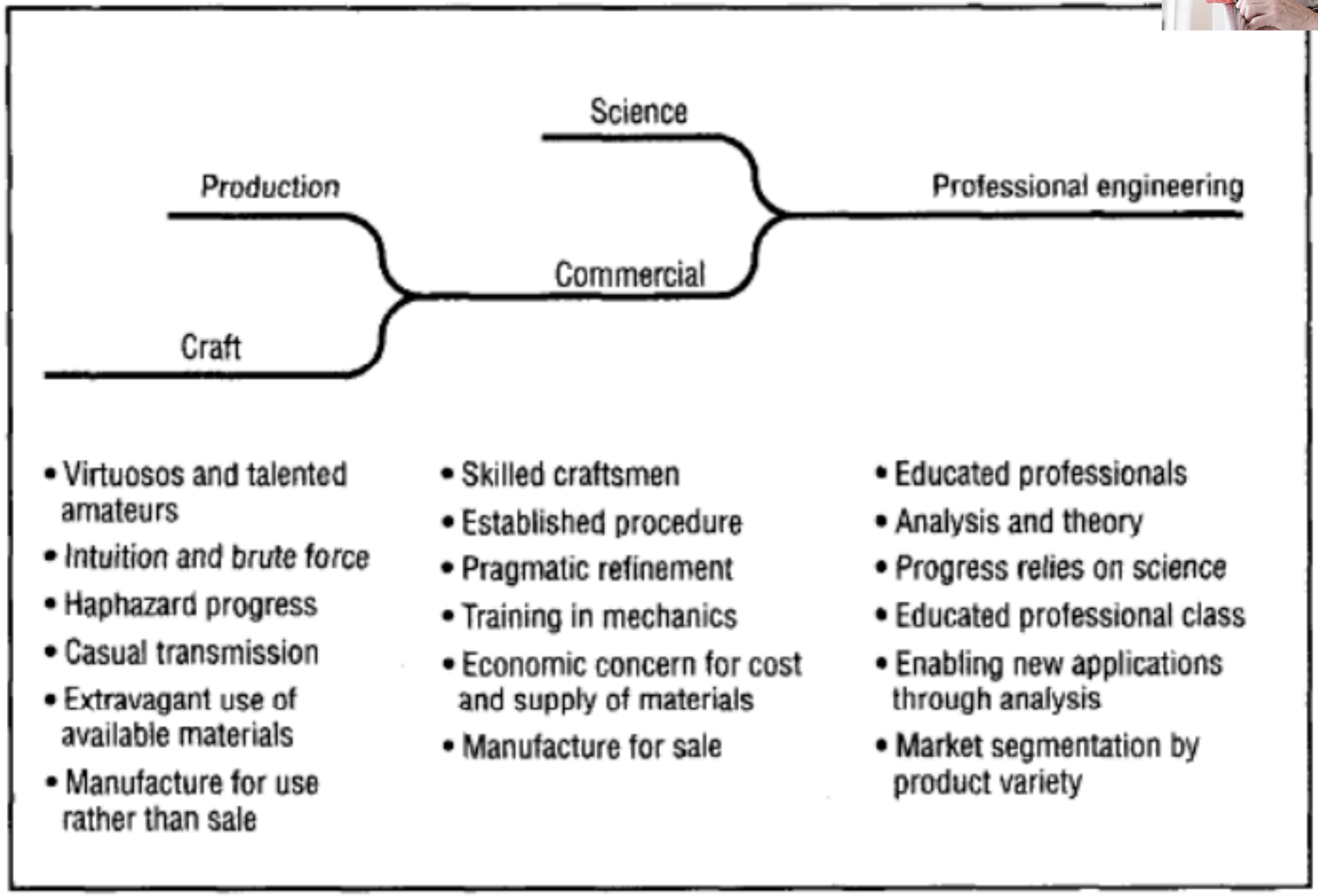
Self-Driving Uber Hits, Kills Pedestrian in Arizona

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



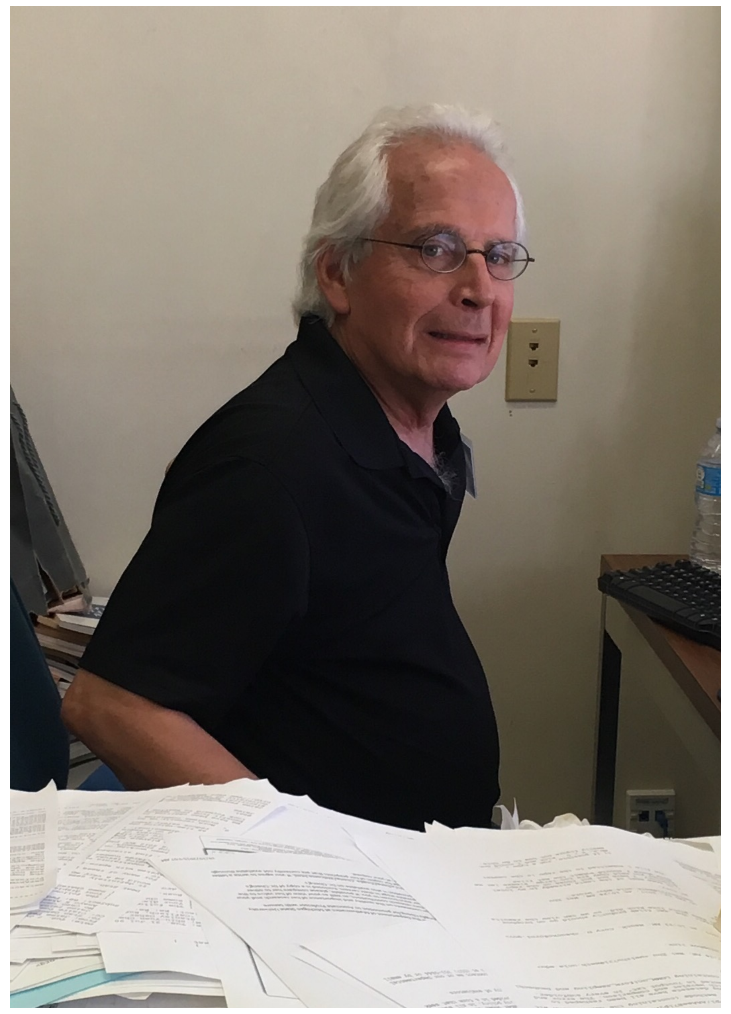
Mary Shaw's evolution of software engineering discipline



Better theory: improves reliability and discipline evolves

Entropy-SGD

Pratik Chaudhari
UCLA (now Amazon/
U Penn)



Stanley Osher, UCLA
Math

Stefano Soatto
UCLA Comp Sci.

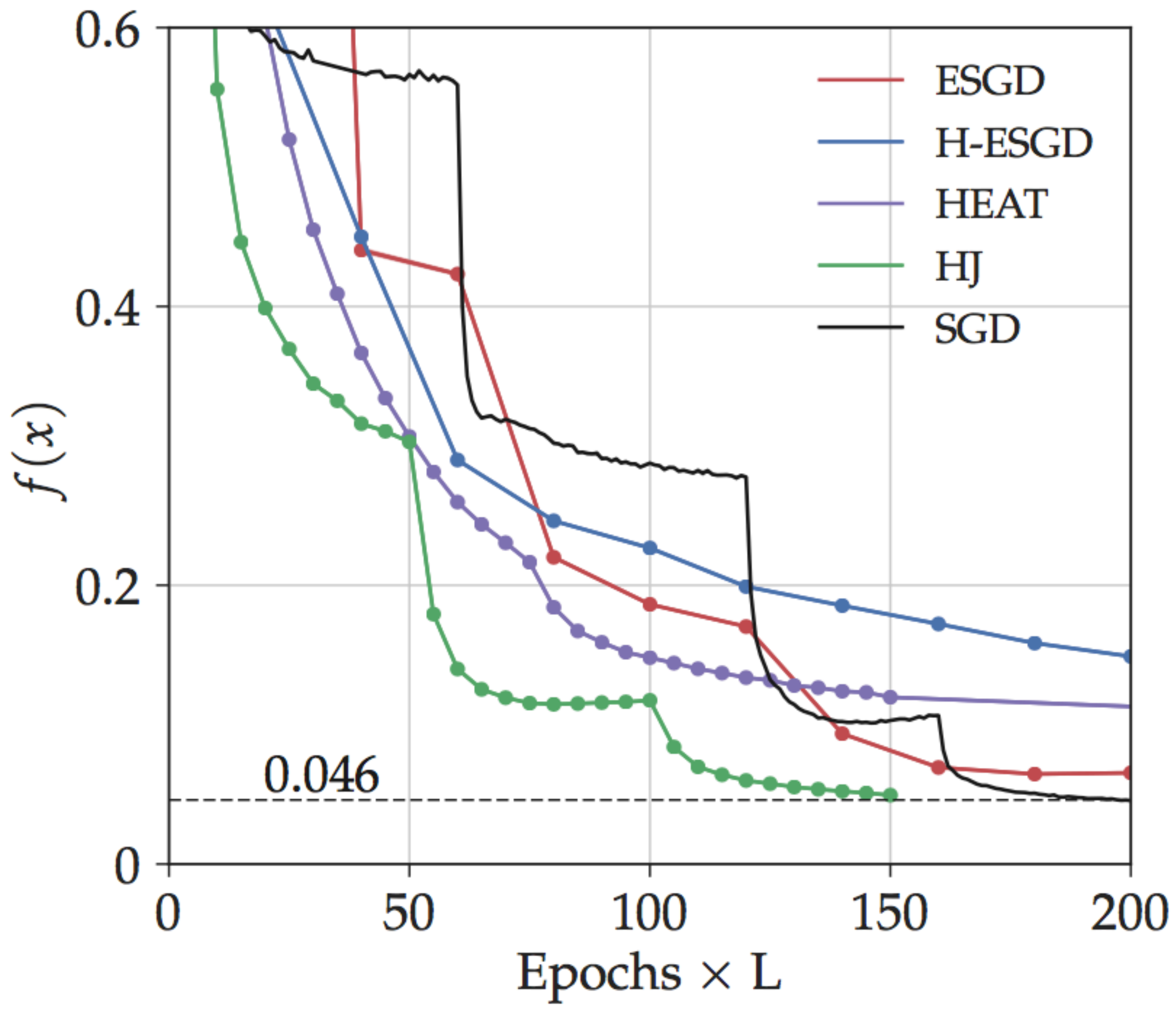
Guillaume Carlier,
CEREMADE, U. Parix
IX Dauphine

- 2017 UCLA PhD student (at time of research)
- 2018 (present) Amazon research
- Fall 2019 Faculty in ESE at U Penn

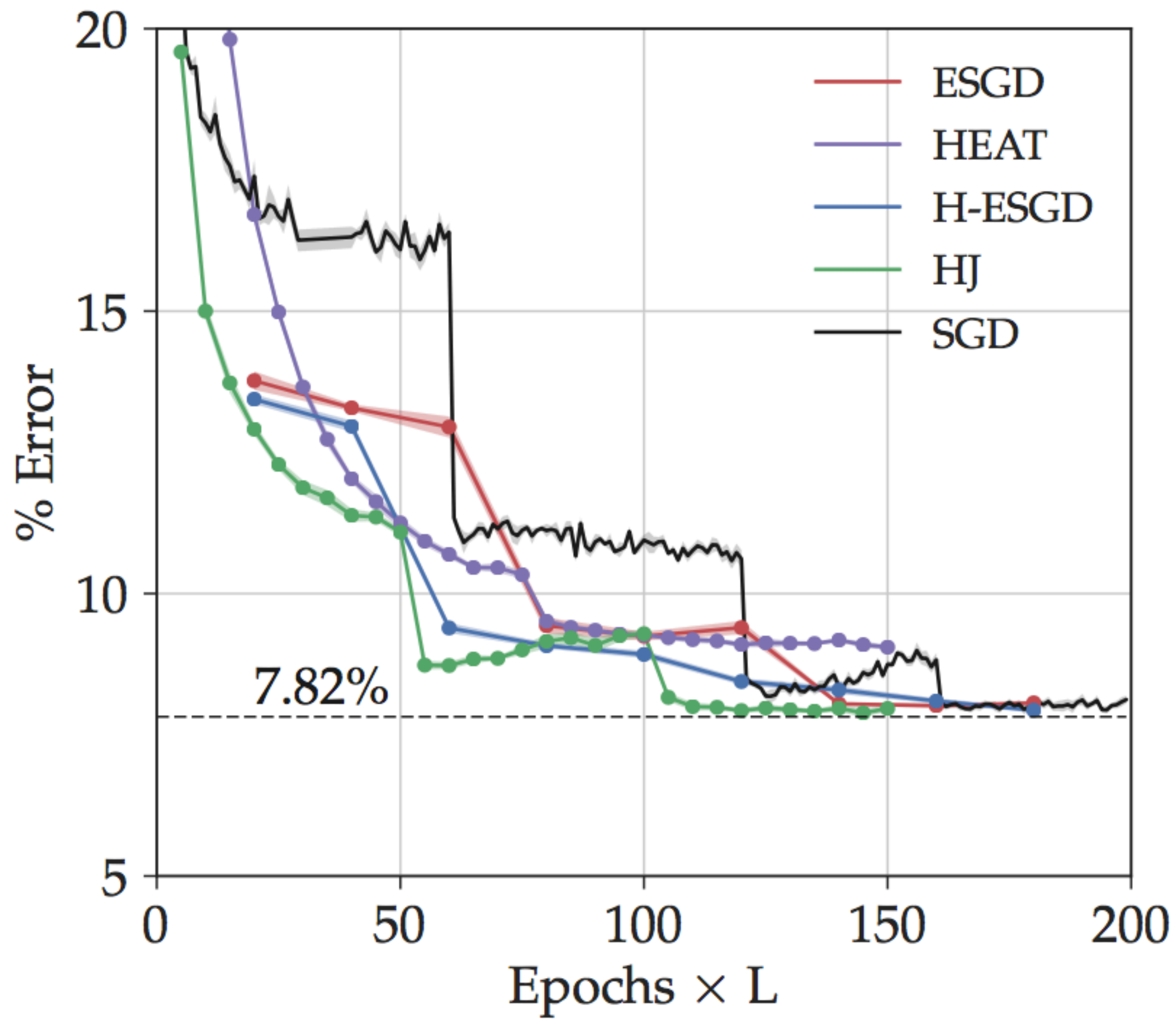
Deep Relaxation: partial differential equations for optimizing deep neural networks Pratik Chaudhari, Adam M. Oberman, Stanley Osher, Stefano Soatto, Guillame Carlier 2017

Entropy SGD results in Deep Neural Networks (Pratik)

Visualization of Improvement in training loss (left)
Improve in Validation Error (right)
dimension = 1.67 million

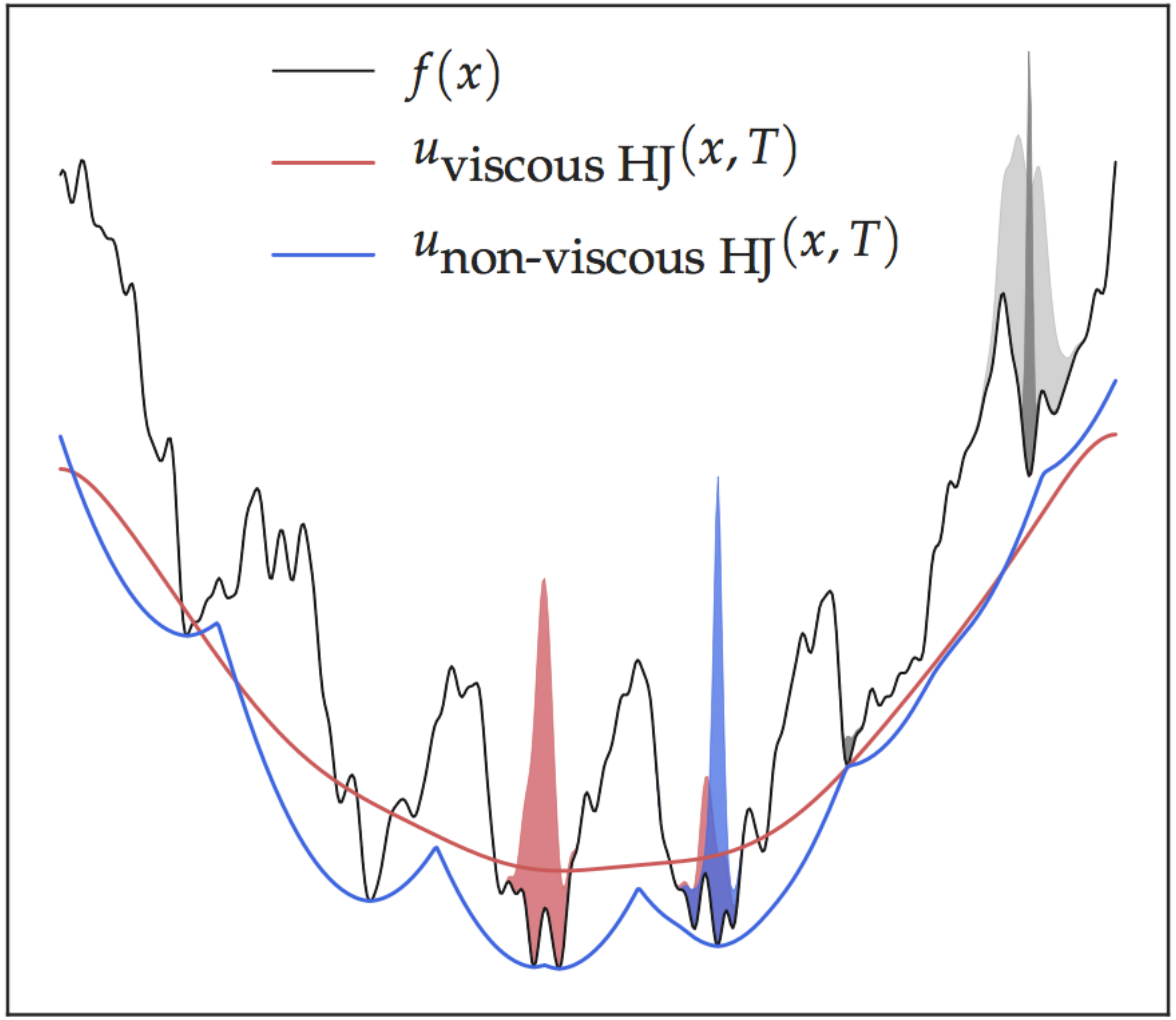


(A) All-CNN: Training loss



(B) All-CNN: Validation error

Different Interpretation: Regularization using Viscous Hamilton-Jacobi PDE



Solution of PDE in one dimension. Cartoon: Algorithm only solves PDE for time depending on Hf(x).

Expected Improvement Theorem in continuous time

Theorem 11. *Let* $x_{\text{csgd}}(s)$ *and* $x_{\text{sgd}}(s)$ *be solutions of* (CSGD) *and* (SGD), *respectively, with the same initial data* $x_{\text{csgd}}(0) = x_{\text{sgd}}(0) = x_0$. *Fix a time* $t \geq 0$ *and a terminal function,* $V(x)$. *Then*

$$\mathbb{E}\left[V(x_{\text{csgd}}(t))\right] \leq \mathbb{E}\left[V(x_{\text{sgd}}(t))\right] - \frac{1}{2}\,\mathbb{E}\left[\int_0^t \left|\alpha(x_{\text{csgd}}(s), s)\right|^2 \, ds\right].$$

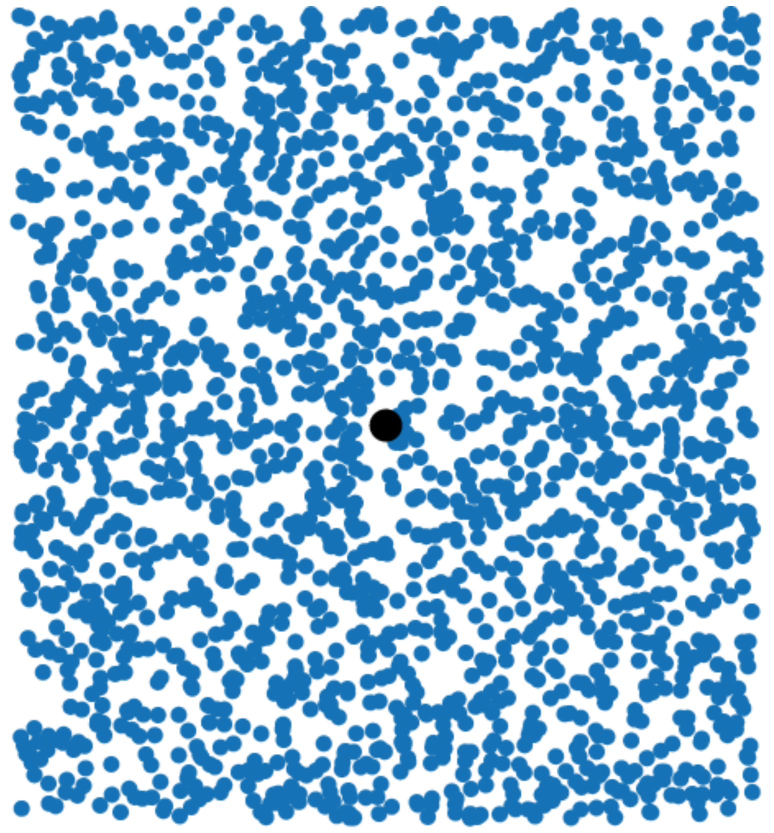
The optimal control is given by $\alpha(x,t) = \nabla u(x,t)$, where $u(x,t)$ is the solution of (HJB) along with terminal data $u(x,T) = V(x)$.

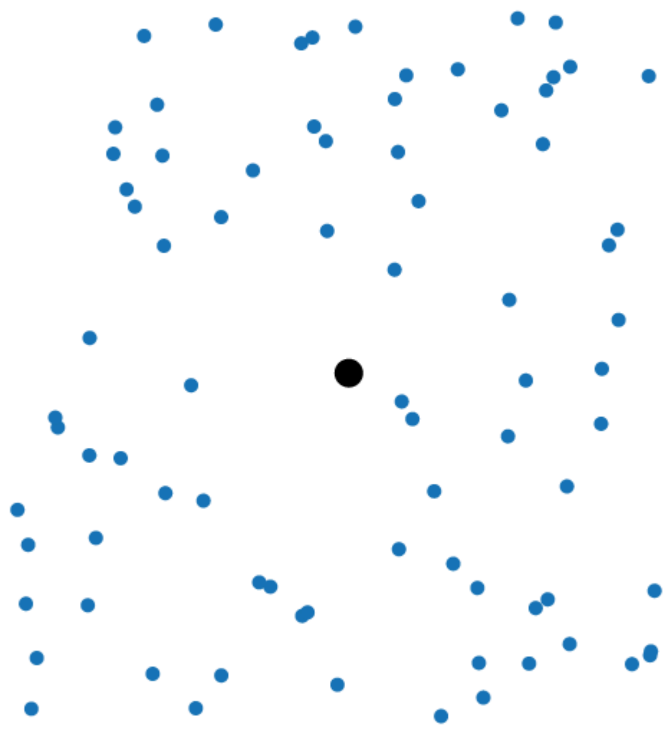
Adaptive-SGD

joint with PhD Student Mariana Prazeres

Model for mini-batch gradients: k=1 means.

full gradient





mini-batch

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n} |x - a_i|^2, \qquad \nabla f(x) = x - \bar{a}$$

$$f_b(x) = \frac{1}{2|b|}\sum_{i\in b} |x - a_i|^2, \qquad \nabla f_b(x) = x - \bar{a}_b$$

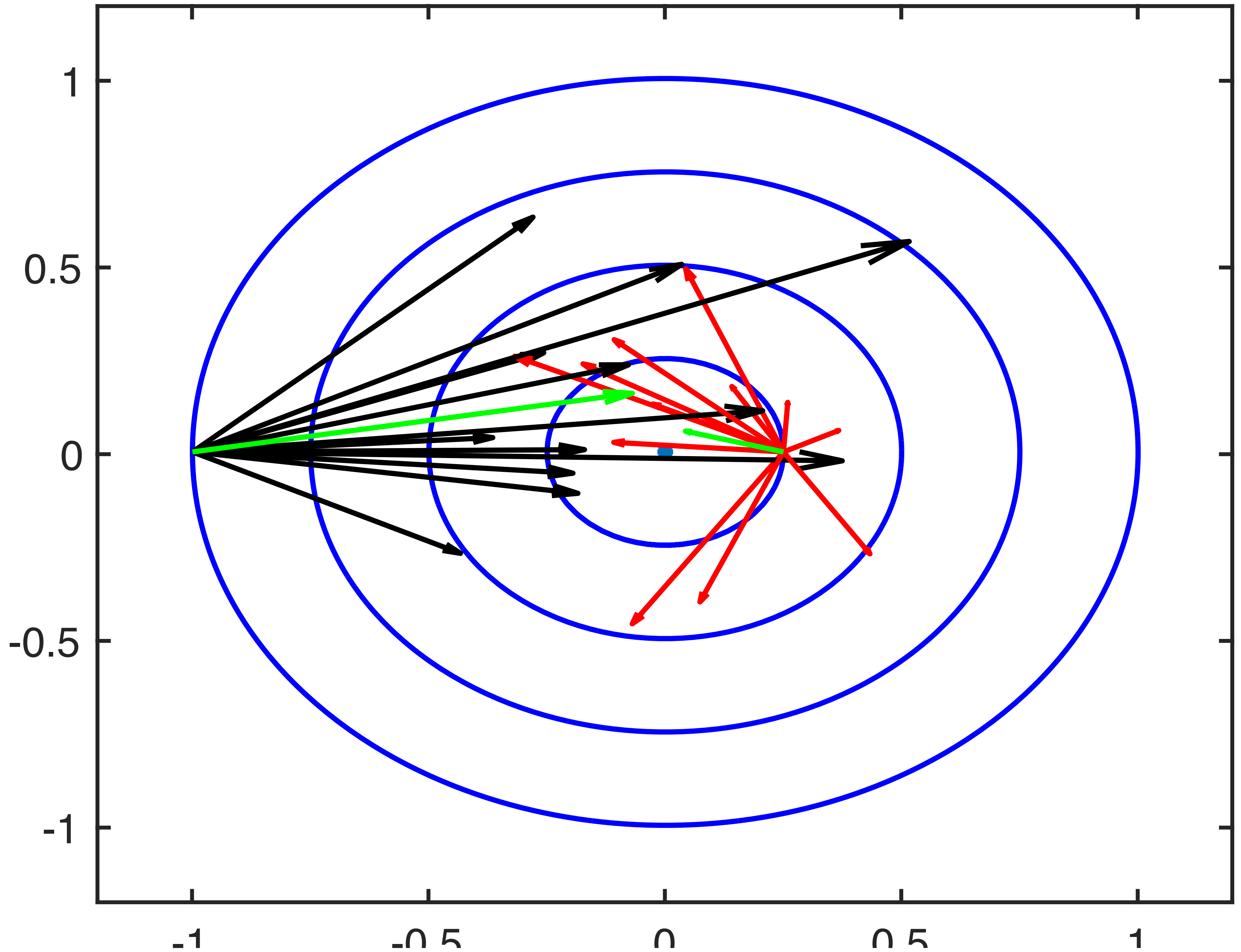
$$e_b = \nabla f(x) - \nabla f_b(x) = \bar{a}_b - \bar{a}$$

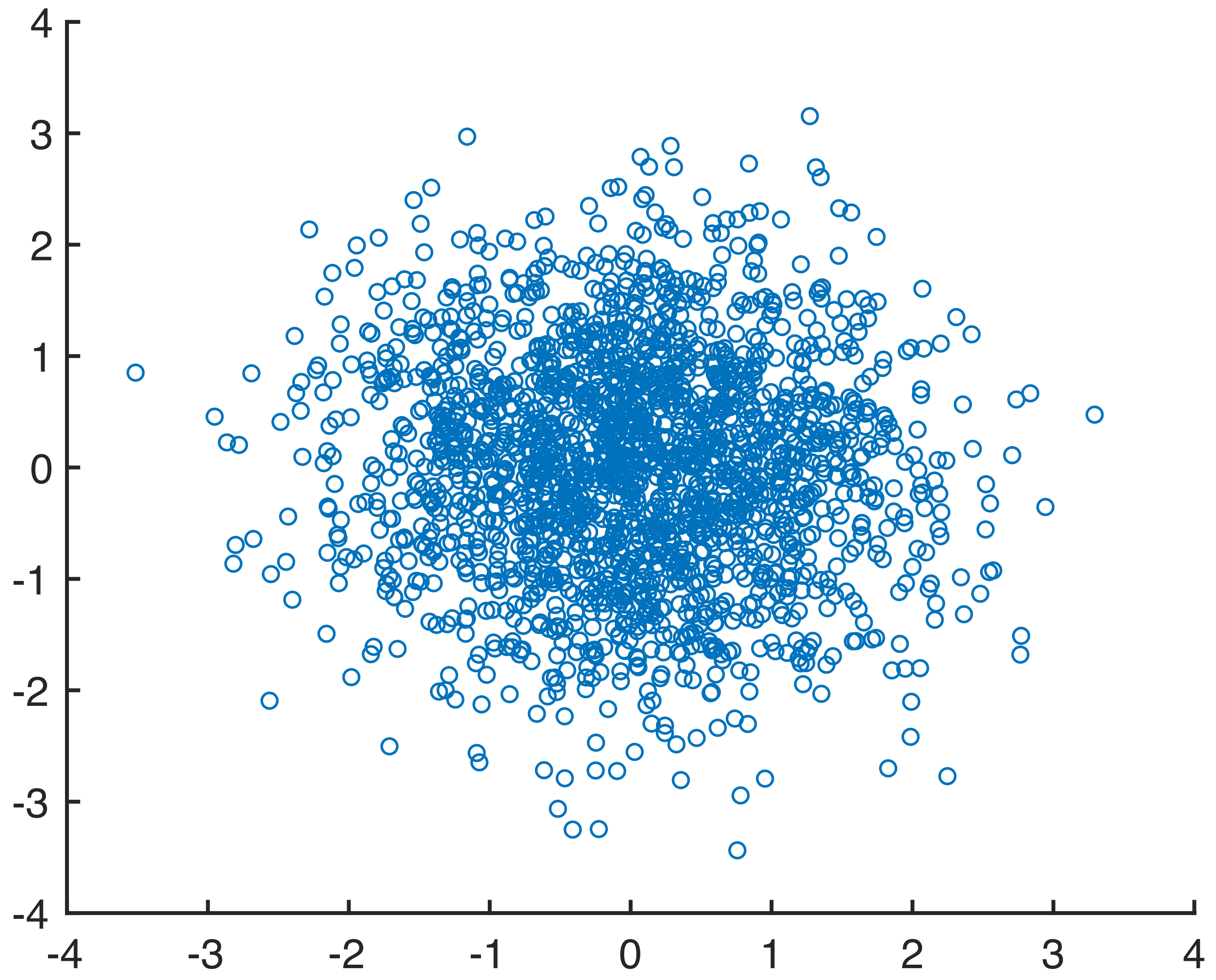
$$\mu = E[e_b] = E[\bar{a}_b - \bar{a}] = 0$$

$$\Sigma^2 = Var(e_b) = E[(\bar{a}_b - \bar{a})^2]$$

For k>1 means, same calculation applies, if we restrict to the active indices. This leads to smaller active batch sizes, and higher variance

Motivation: quality of g_{MB} depends on x



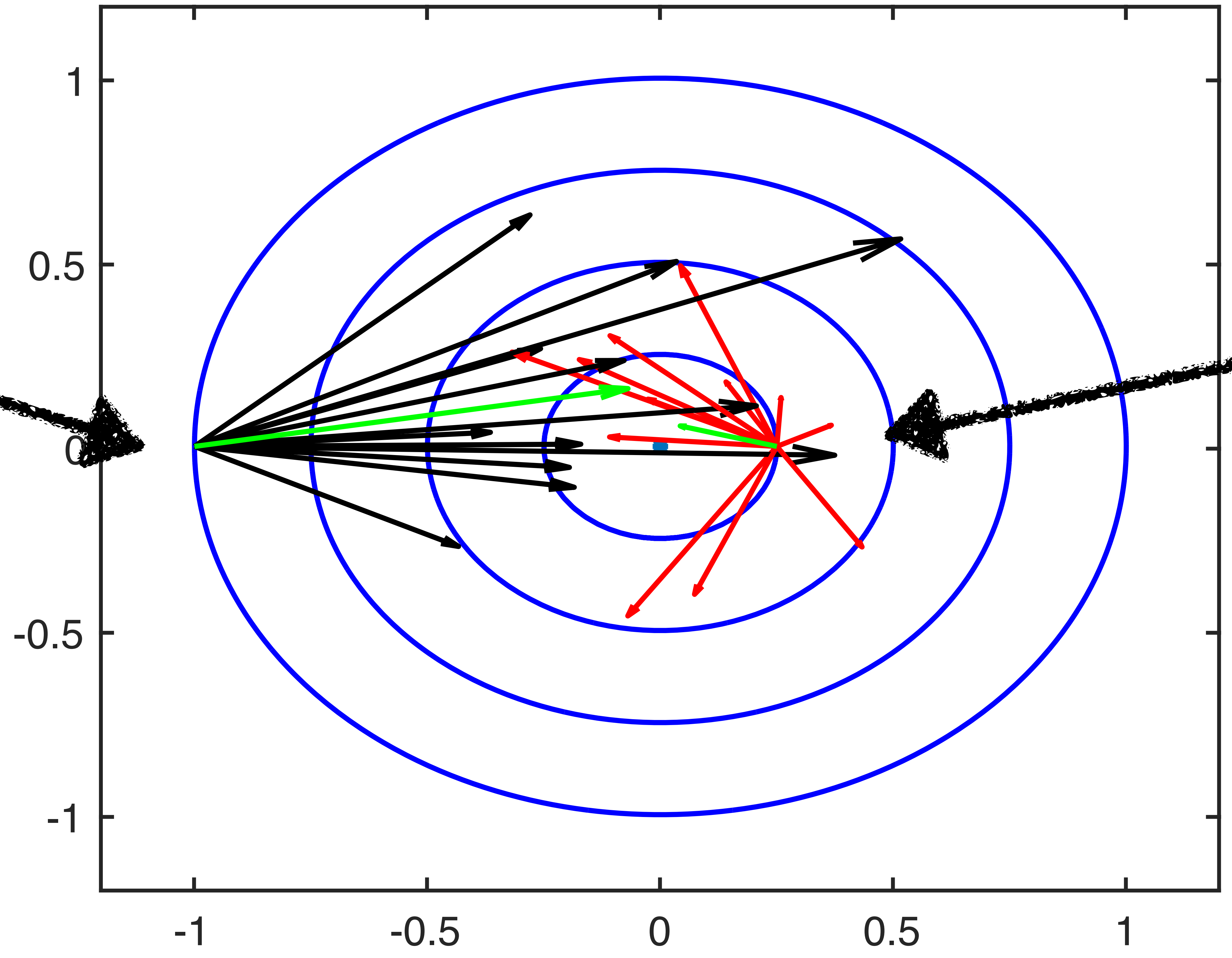


Minibatch gradients, their average (green), contours of $|x - x^*|$.

- far from x* mb gradients point in a good direction.
- near x* require more samples, or small steps (so that directions average in t

Adapt in Space instead of Time

use MB = 10



use MB = 60

The ideal learning rate/batch size combination should depend on x (space) rather than k (time).

Adaptive SGD

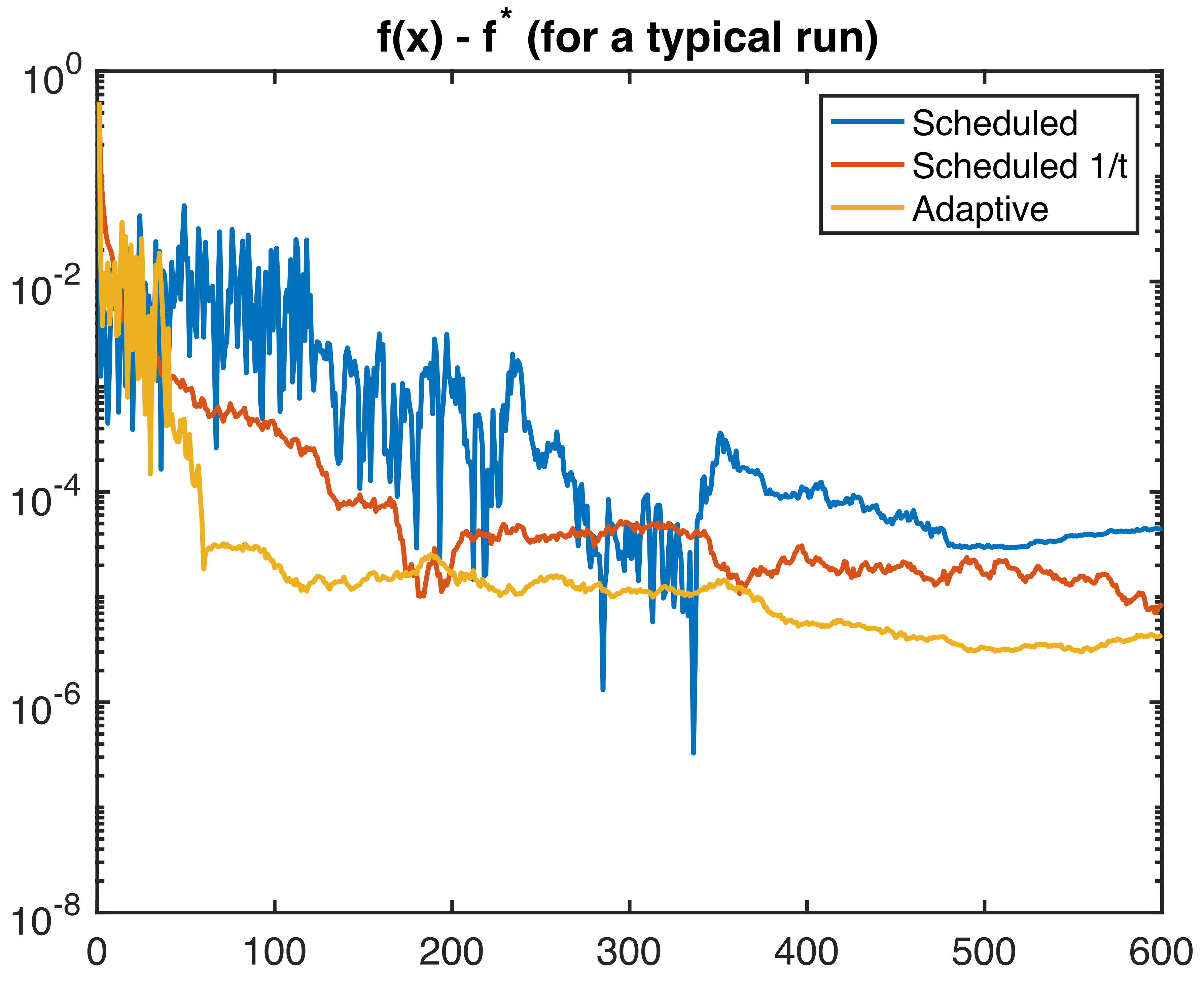
- Adaptively, depending on x, decide on
  - MB size, or
  - learning rate.
- Use the following formula (derived later)

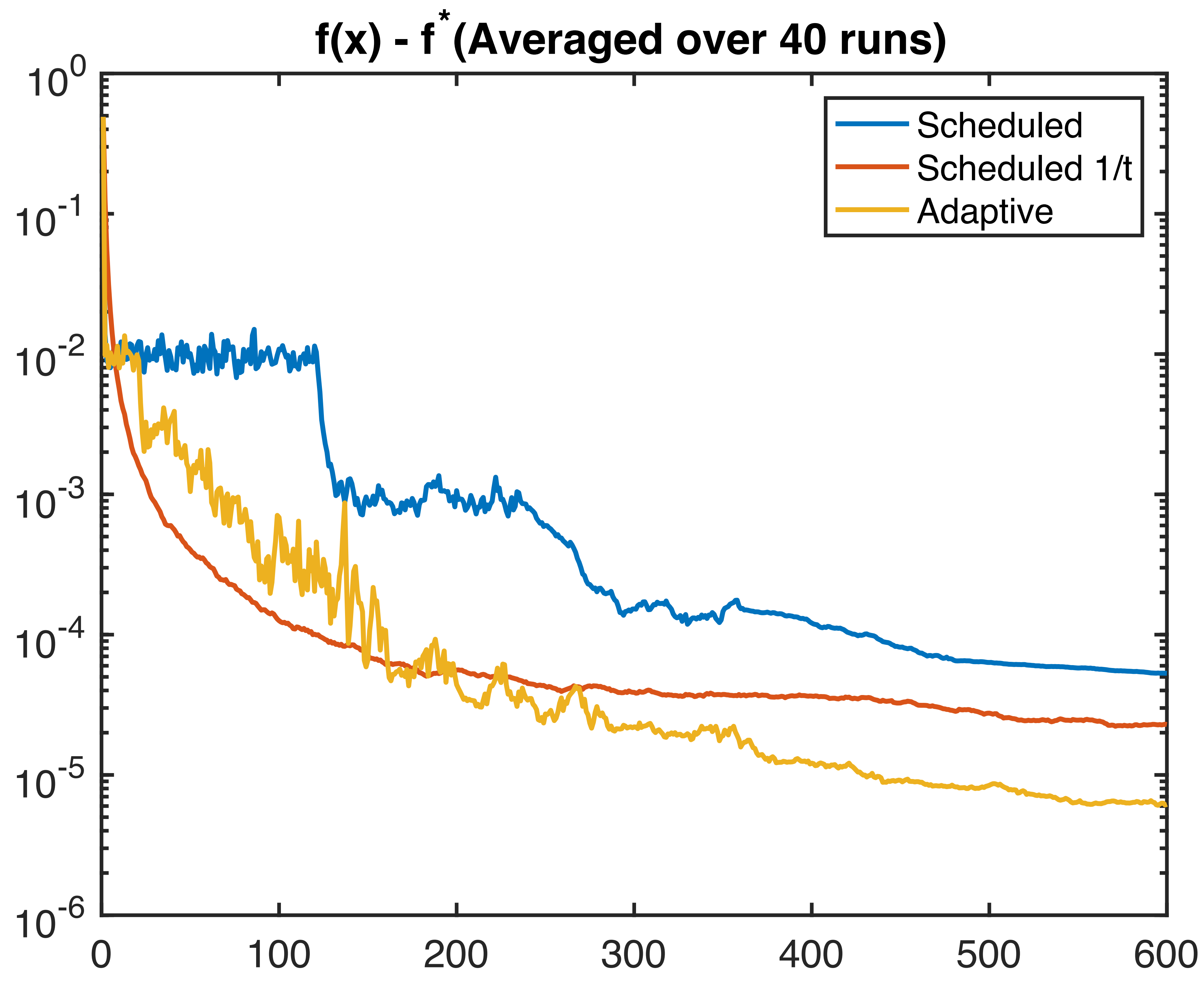
(SAGD) $$x_{k+1} = x_k - h_k \nabla_{mb} f(x_k)$$

(SALR) $$h_k \leq 2 \frac{f(x_k) - f^*}{\mathbb{E}\left[\|\nabla_{mb} f(x_k)\|^2 \right]}$$

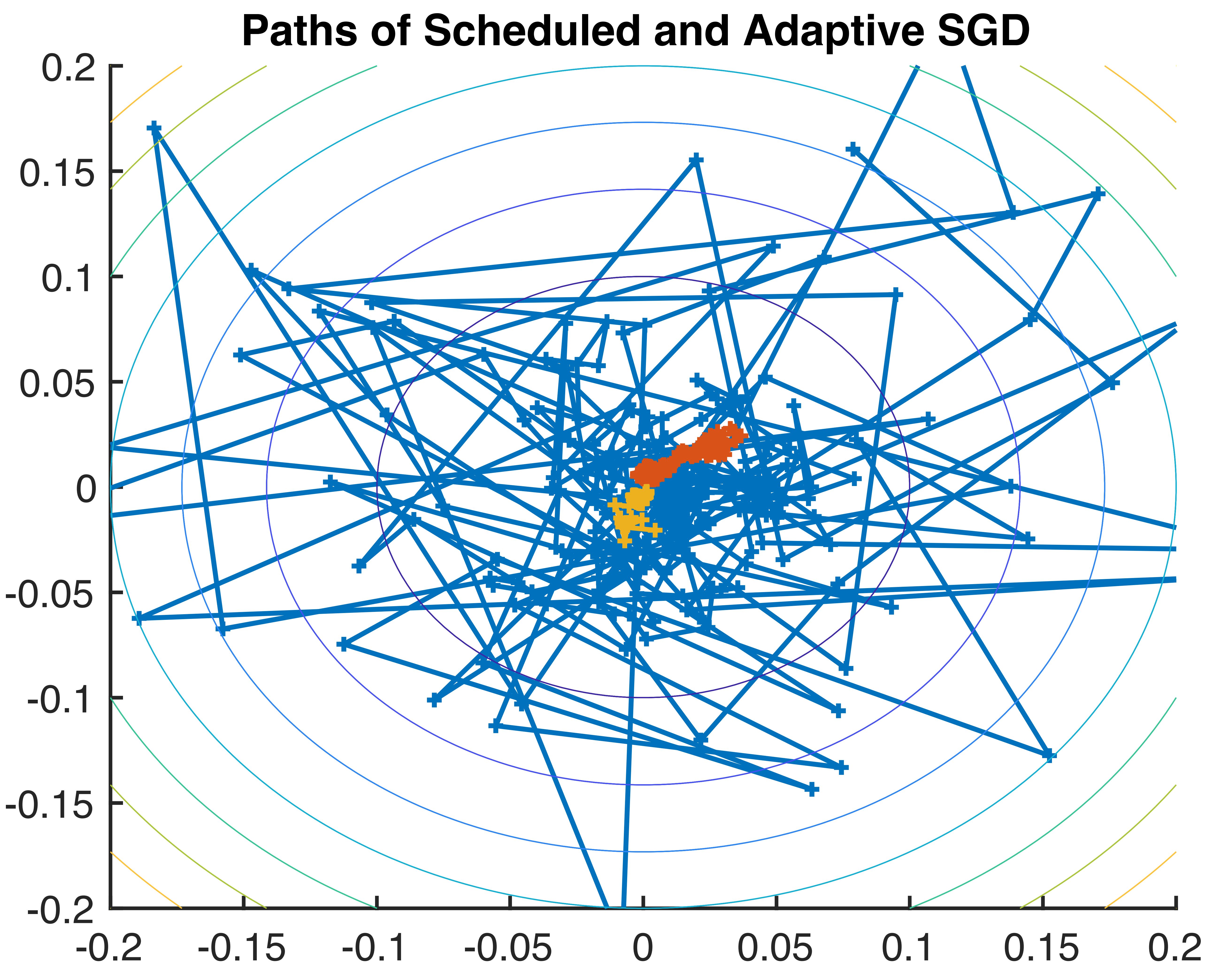
- f large, learning rate large (ok to use small MB)
- g small, var(MB) restricts learning rate (so use large MB)

Benchmarks: Fix MB and adapt h





Left: Not too clear what is happening with one path.
Right: Average over several runs to see the trend



The variance of the paths is clear from this figure

Proof of Convergence with Rate

Theorem 3.4. *Suppose f is μ-strongly convex and L-smooth. Define the SGD update* (SAGD) *with adaptive learning rate given by* (SALR). *Assume*

$$\mathbb{E}\left[e_k\right] = 0 \tag{5}$$

Then

$$q_k \leq \frac{1}{\alpha k + q_0^{-1}}, \qquad \textit{for all } k \geq 0$$

where

$$\alpha = \frac{\mu}{\frac{\sigma^2}{2\mu} + (L - \mu) q_0}$$

rate is same order at SGD, but with better constant

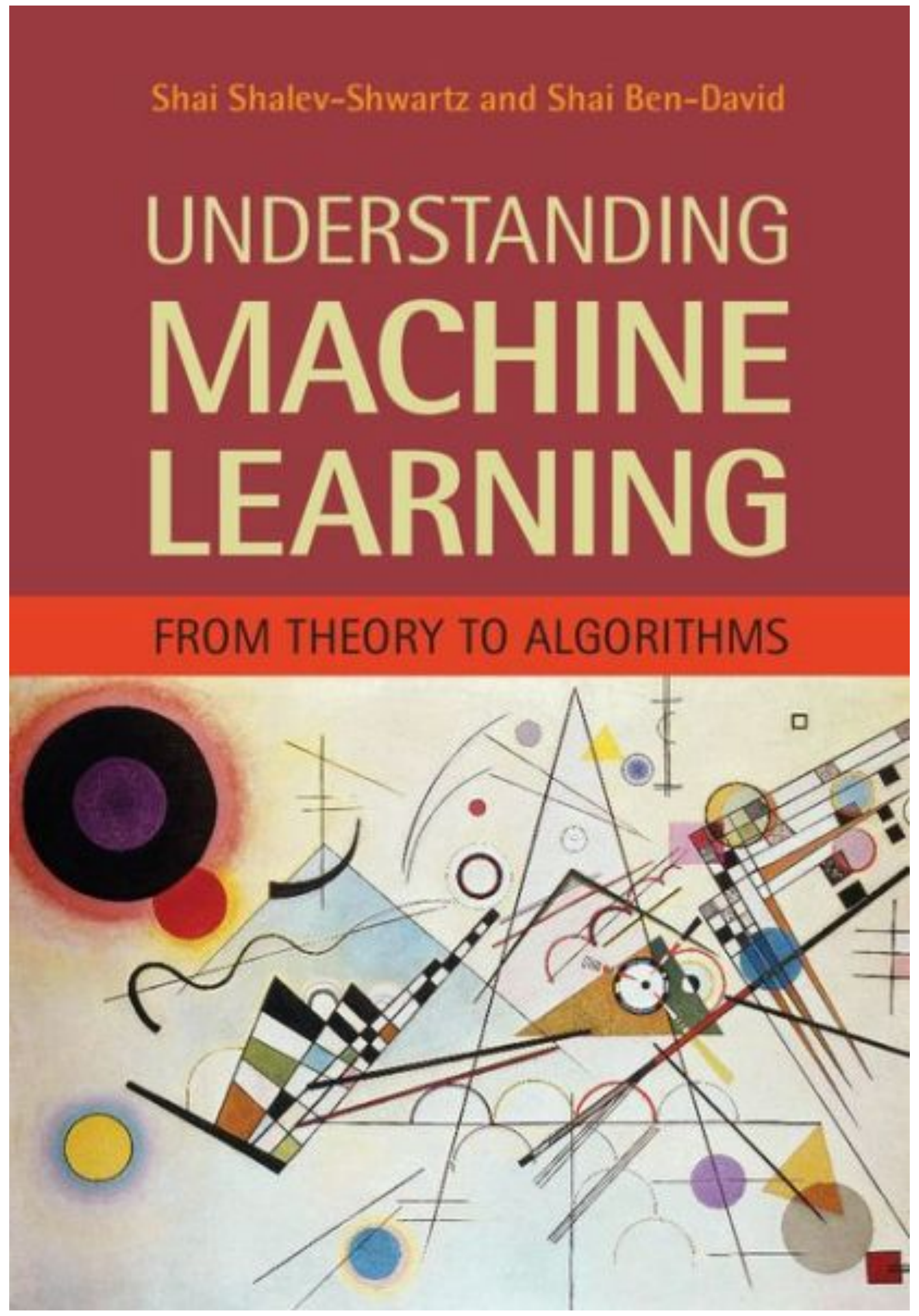
Proof of Convergence and Generalization for Lipschitz Regularized DNNs

joint with Jeff Calder

Lipschitz regularized Deep Neural Networks converge and generalize O. and Jeff Calder; 2018

Background
on the generalization/convergence result

Problem: traditional ML theory does not apply to Deep Learning



"Understanding Deep Learning requires rethinking generalization" Zhang (2016)

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

A *new idea* is needed to make Deep Learning more reliable.

inspiration: old idea: Total Variation Denoising [1992] R-Osher-F.

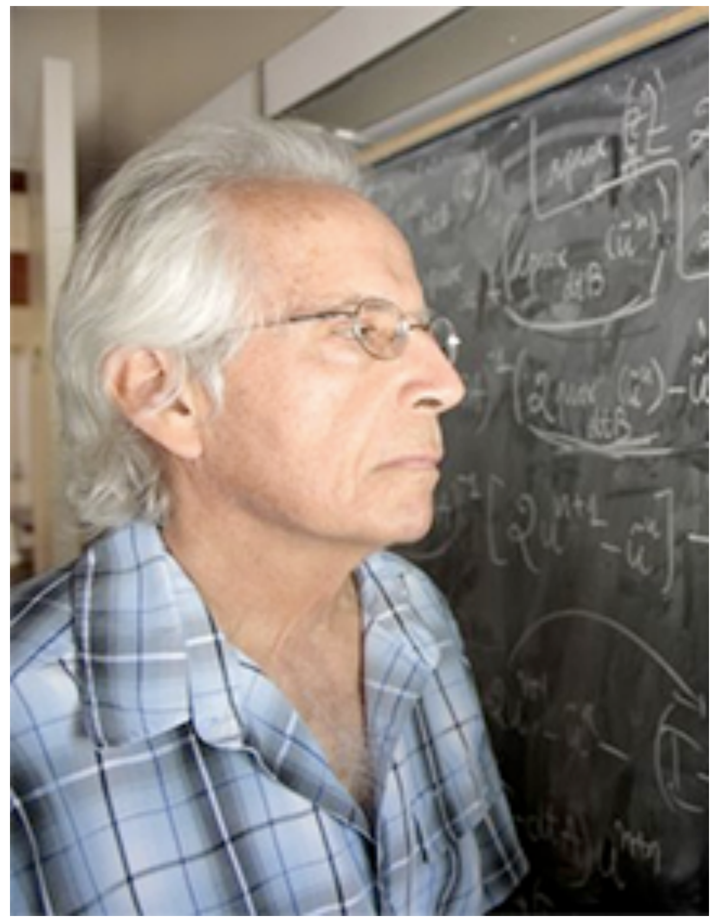
used in early, high profile image reconstruction of video images.

Original

Noisy image

Denoised image

$$J[u] = L[u; u_0] + \lambda R[\nabla u]$$



Stanley Osher

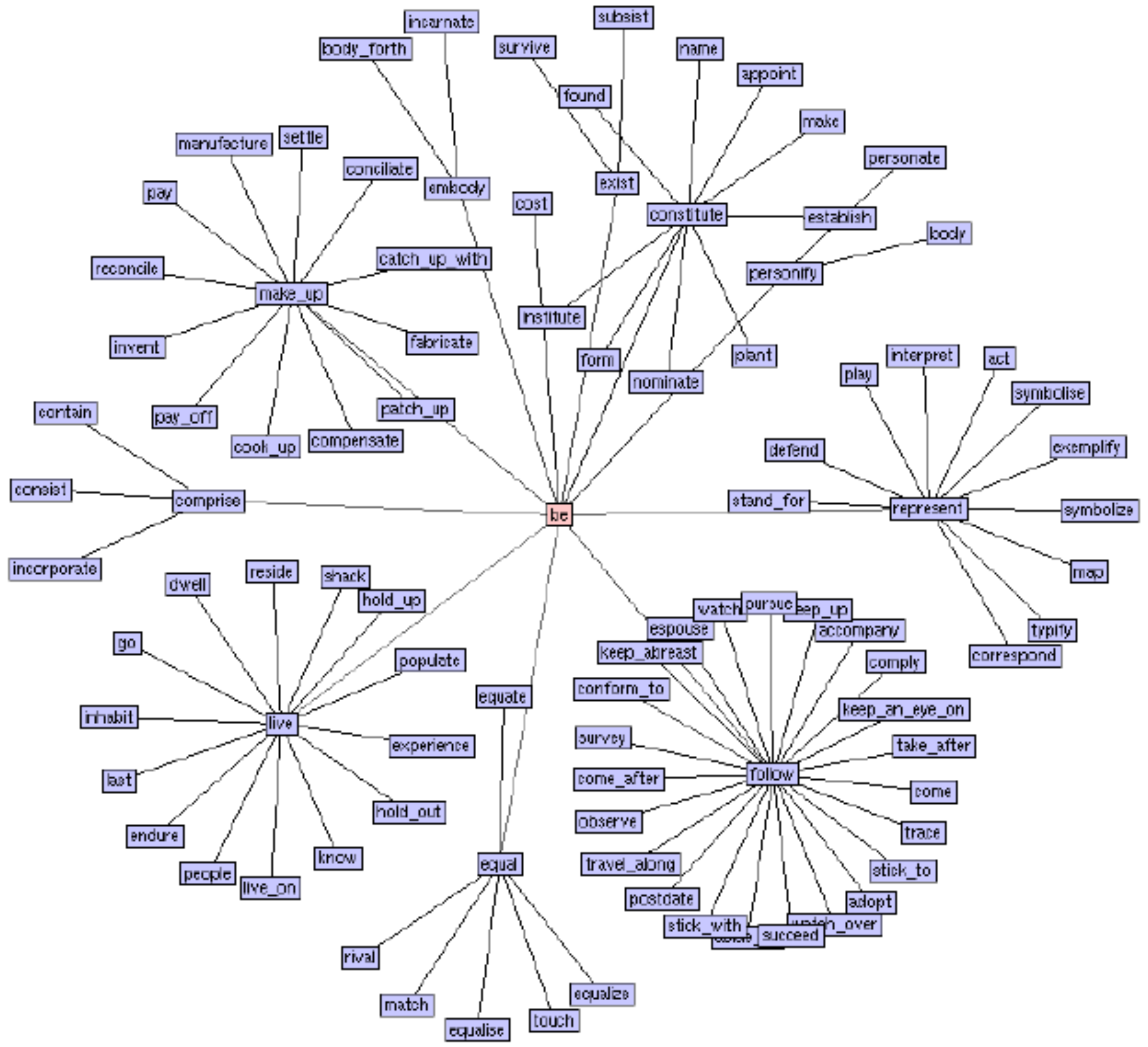
- minimize a variational functional: combination of a loss term, to the original noisy image, and a regularization term

- Regularization is large on noise, small on images

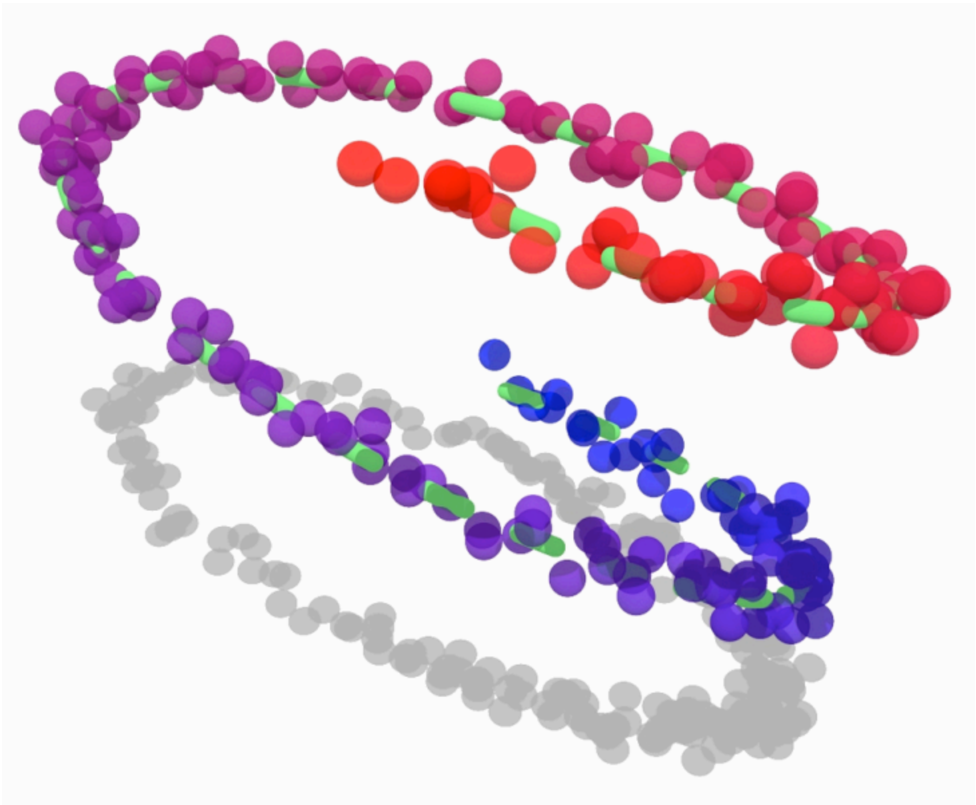
Regularization: from images to maps

x in M = manifold of images

f(x)



word labels



the learned map: well-behaved on data manifold, but very bad off the manifold (without regularization)

What is new in our result?

- Bartlett proved generalization under the assumption of Lipschitz regularity.
- However, DNNs are not uniformly Lipschitz
- By adding regularization to the objective function in the training, we obtain the uniform Lipschitz bounds

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.

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)

$$w^{k+1} = w^k + h_k \nabla_{mb}\ell(\ldots w)$$

C. Inverse problems: allow for a broad class of functions, but modify the loss to choose the right one

$$\min_w \mathbb{E}_{x\sim\rho}\ell(f(x;w), y(x)) + \lambda\|\nabla_x f\|_{L^p(X,\rho(x))}$$

Comment for math audience

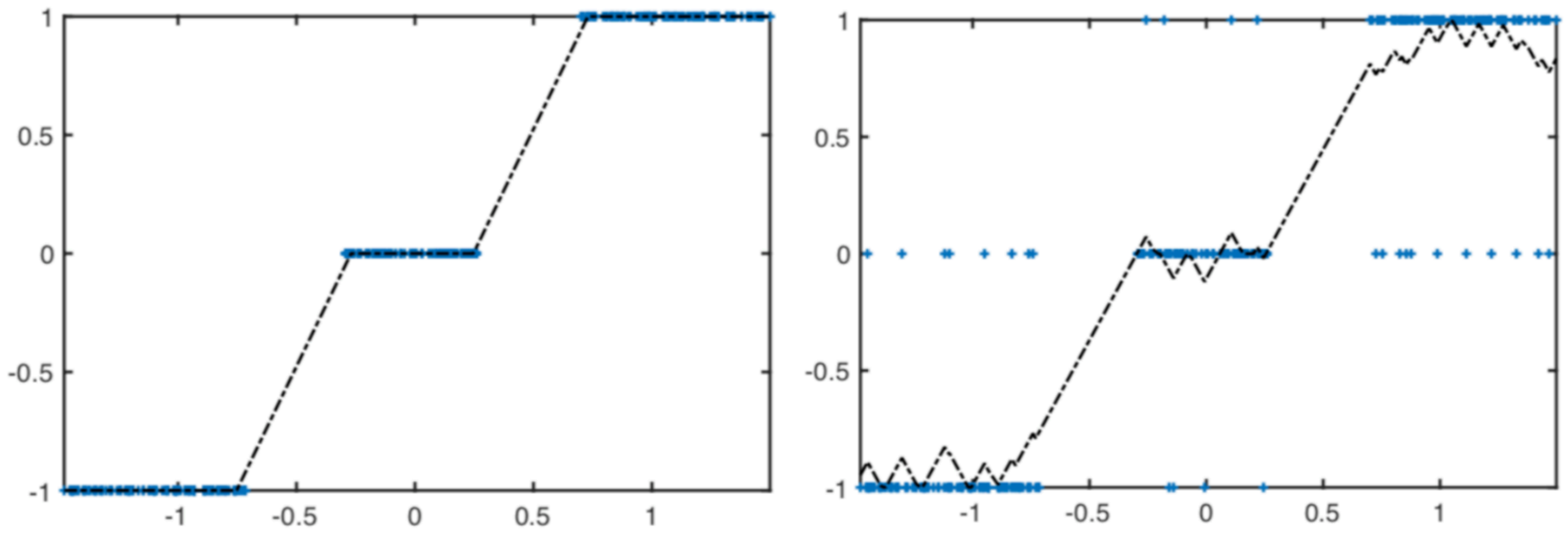
- Our result may not be surprising to math experts
- However, it is a new approach to generalization theory.

- Speaking personally, the hard work was giving a math interpretation of the problem (1.5 years)
- Once the model was set up correctly, and we realized we could implement it in a DNN, the math was relatively easier.
- Paper and proof was done in about 6 weeks.

Convergence: two cases

Clean Labels:
- relevant in benchmark data sets and applications,
- simpler proof, since the clean lable function is a minimizers
- regime of perfect data interpolation possible with DNNs

Noisy labels:
- relevant in applications,
- familiar setting for calculus of variations



Statement and proof sketch of the generalization/convergence result

Lipschitz Regularization of DNNs

Data function: augment the expected loss function on the data with a Lipschitz regularization term

$$(1) \qquad \min_{f:X\to Y} J^n[f] = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;w), u_0(x_i)) + \lambda \max(\mathrm{Lip}(f) - L_0, 0)$$

where L_0 is Lipschitz constant of the data, and n is number of data points. We are interested in the limit as we sample more points. The limiting funtional is given by

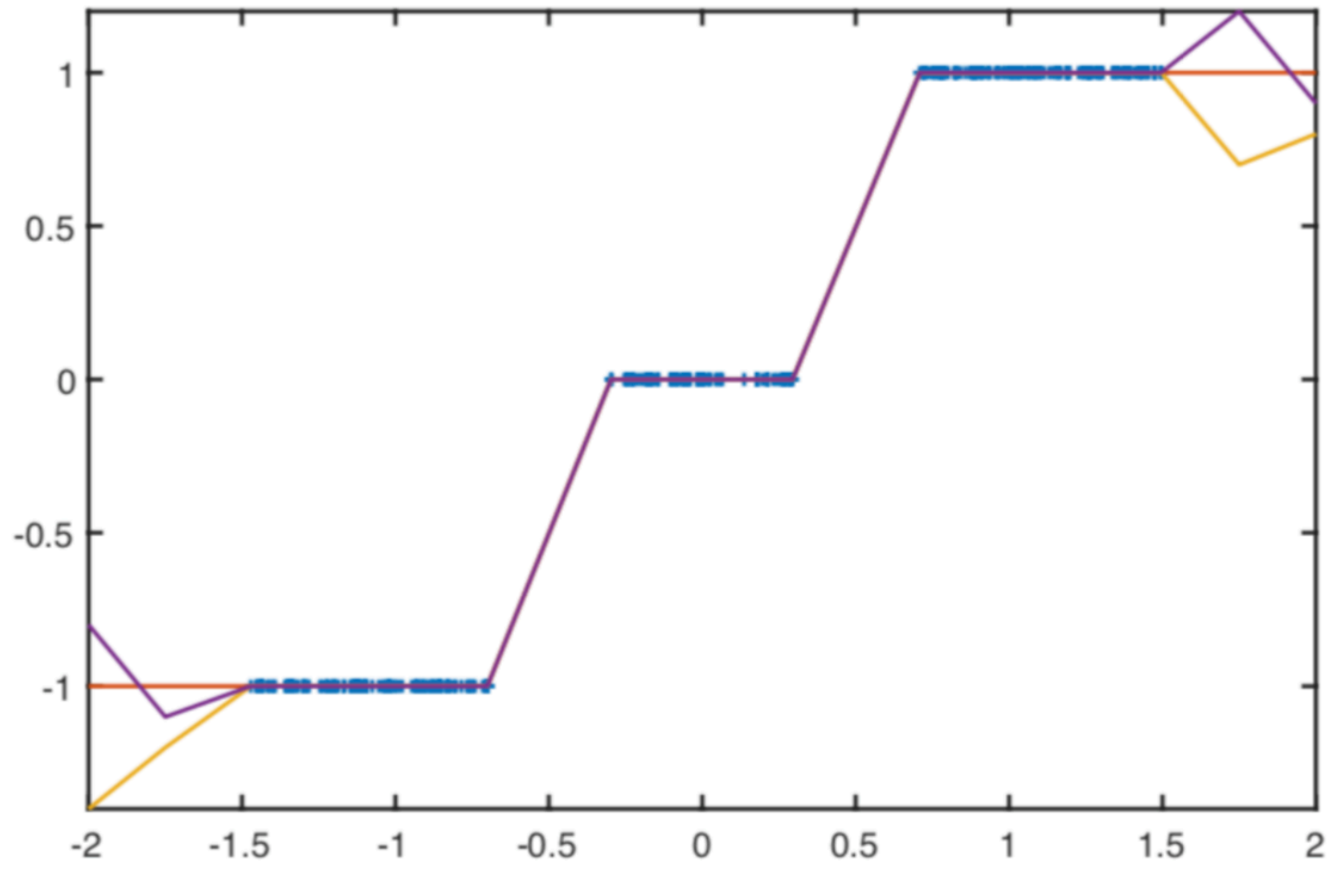
$$(5) \qquad J^{Lip,\rho}[u] \equiv \int_X \ell(u(x), u_0(x))d\rho(x) + \lambda \max(\mathrm{Lip}(u) - L_0, 0)$$

Convergence theorem for Noisy Labels

Theorem 2.11. *Suppose that* $\inf_{\mathcal{M}} \rho > 0$, $\ell : Y \times Y \to \mathbb{R}$ *is Lipschitz, and let* $u^* \in W^{1,\infty}(X;Y)$ *be any minimizer of the limiting functional* (6). *Then with probability one*

$$u_n \longrightarrow u^* \quad \textit{uniformly on } \mathcal{M} = \mathrm{supp}(\rho) \textit{ as } n \to \infty,$$

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* (6).



Convergence on the data manifold. Lipschitz off.

Convergence for clean labels (with a rate)

Theorem 2.7. *Suppose that* $\mathrm{Lip}[u_0] \le L_0$ *and* $\inf_{x\in\mathcal{M}} \rho(x) > 0$. *If* f_n *is any sequence of minimizers of* (1) *then for any* $t > 0$

$$\|u_0 - f_n\|_{L^\infty(\mathcal{M};Y)} \le C L_0 \left(\frac{t\log(n)}{n}\right)^{1/m}$$

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
- uniform sampling vs random 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 $\mathrm{Lip}[u_n] \leq L_0$ 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

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

The proof is completed by invoking Lemma 2.9. $\square$

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 ℓ satisfies*

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

Then under the assumptions of Theorem 2.7

$$L[u_n, \rho] \leq C L_0^q \left(\frac{t \log(n)}{n}\right)^{q/m}$$

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

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

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

The proof is completed by invoking Theorem 2.7. □

Lipschitz Regularization improves Adversarial Robustness

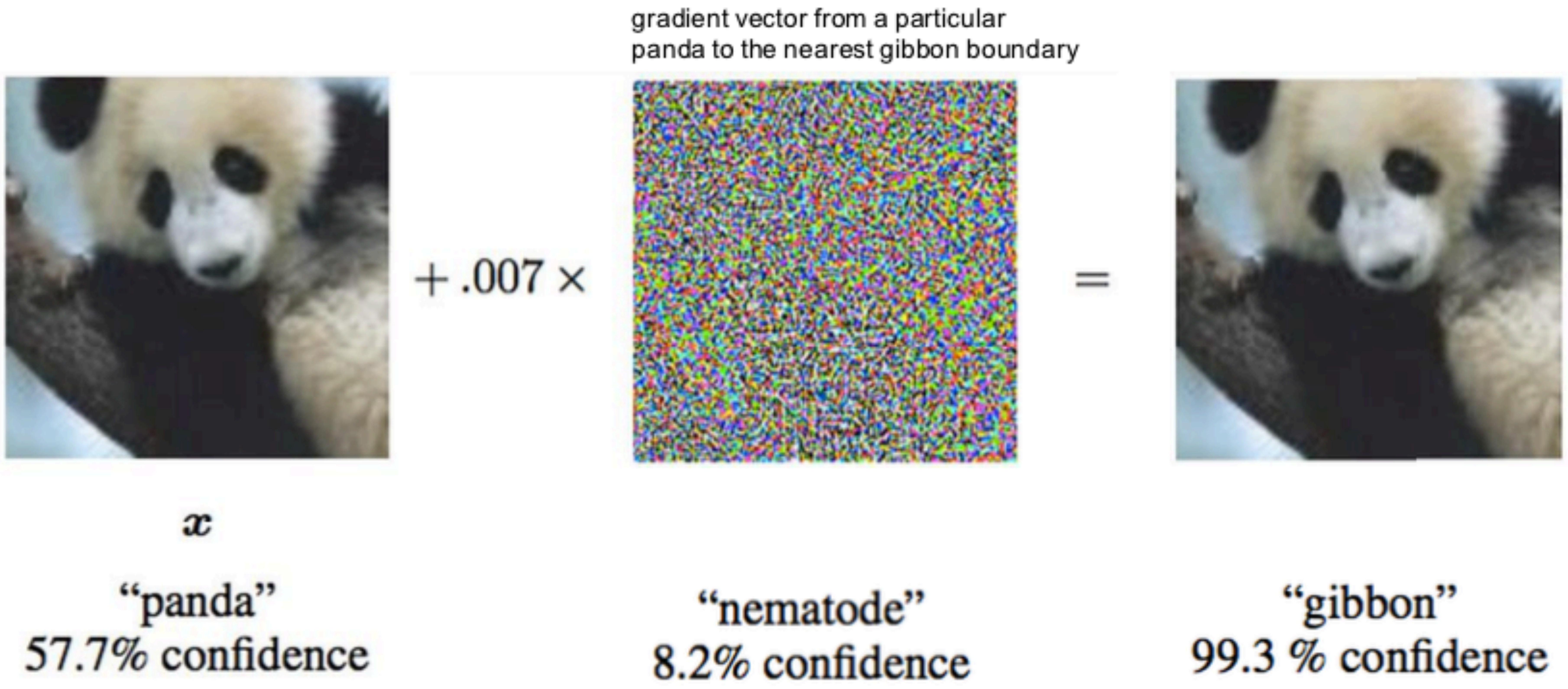
Chris Finlay (current PhD student)

Bilal Abbasi (former PhD now working in AI)

Improved robustness to adversarial examples using Lipschitz regularization of the loss

Chris Finlay, O., Bilal Abbasi; Oct 2018; arxiv

Adversarial Attacks



Adversarial Attacks on the loss

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, ℓ.

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

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)

Implementation of Lipschitz Regularization of the Loss

Write $\ell(x) = \ell(f(x), u_0(x))$.
Write $L_{\ell\circ f}$ for the Lipschitz constant of loss of the model.

Definition 3.1. The Lipschitz constant of a function f is given by

$$\mathrm{Lip}_{2,\infty}(f) = \max_{x_1 \neq x_2} \frac{\|f(x_1) - f(x_2)\|_\infty}{\|x_1 - x_2\|_2} \tag{9}$$

When f is differentiable on a closed, bounded domain, X, then

$$\mathrm{Lip}(f) = \max_x \|\nabla f(x)\|_{2,\infty}. \tag{10}$$

we can approximate the Lipschitz constant by testing on the data

$$\max_{x\in\mathcal{D}} \|\nabla f(x)\|_{2,\infty} \le \mathrm{Lip}(f)$$

Robustness bounds from the Lipschitz constant of the Loss

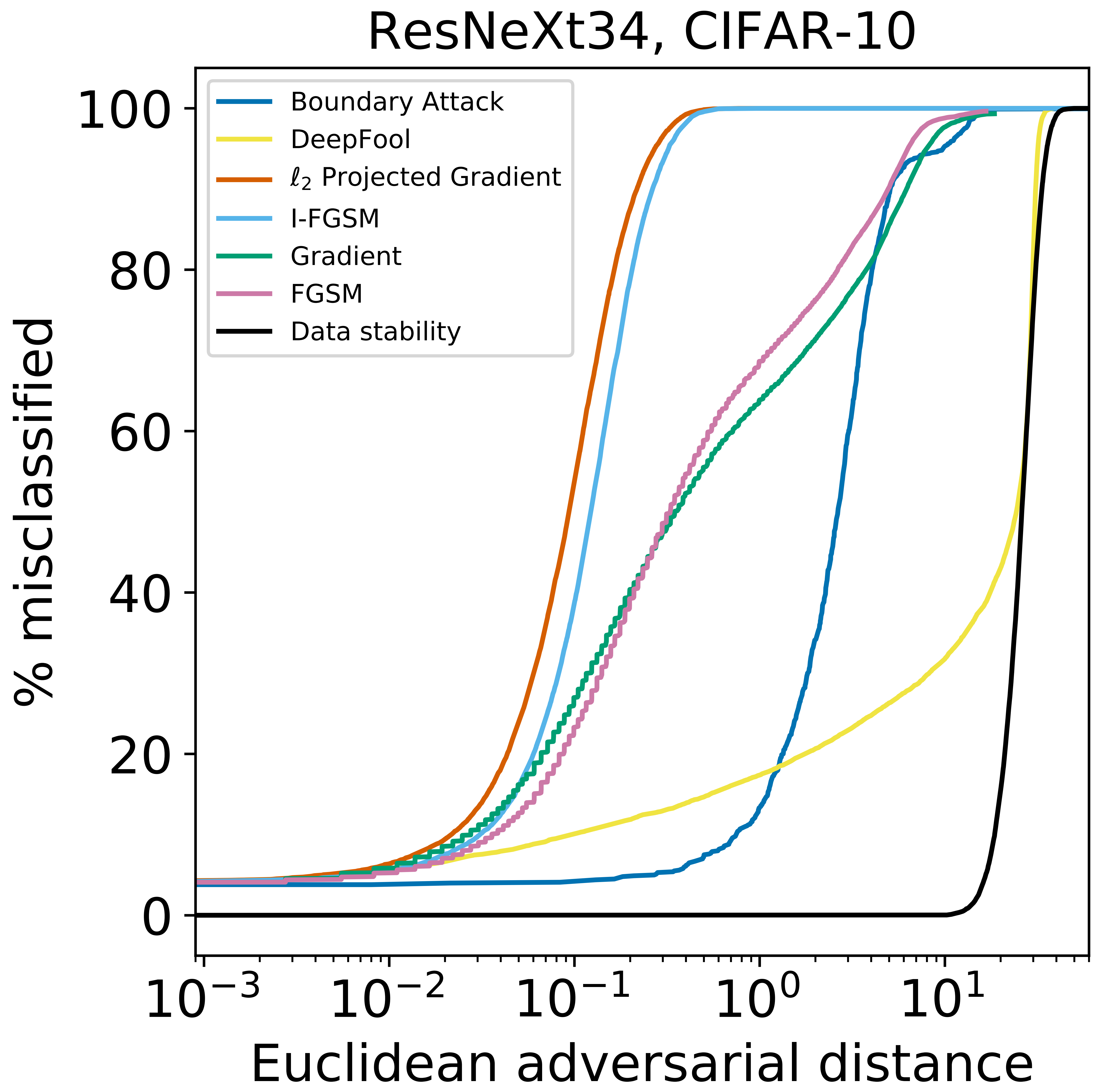
a successful attack on image x will have adversarial distance at least

(2) $$\delta \geq \min_{j \neq i*} \frac{f_{i^*}(x) - f_j(x)}{2L_f}$$

where L_f is the Lipschitz constant of the model, f, and i^* is the correct label of x.

So training the model to have a better Lipschitz constant will improve the adversarial robustness bounds.

Arms race of attack methods and defences



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

Strongest attacks:

1. Iterative l2-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,

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

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

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

Adversarial Training augmented with Lipschitz Regularization

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

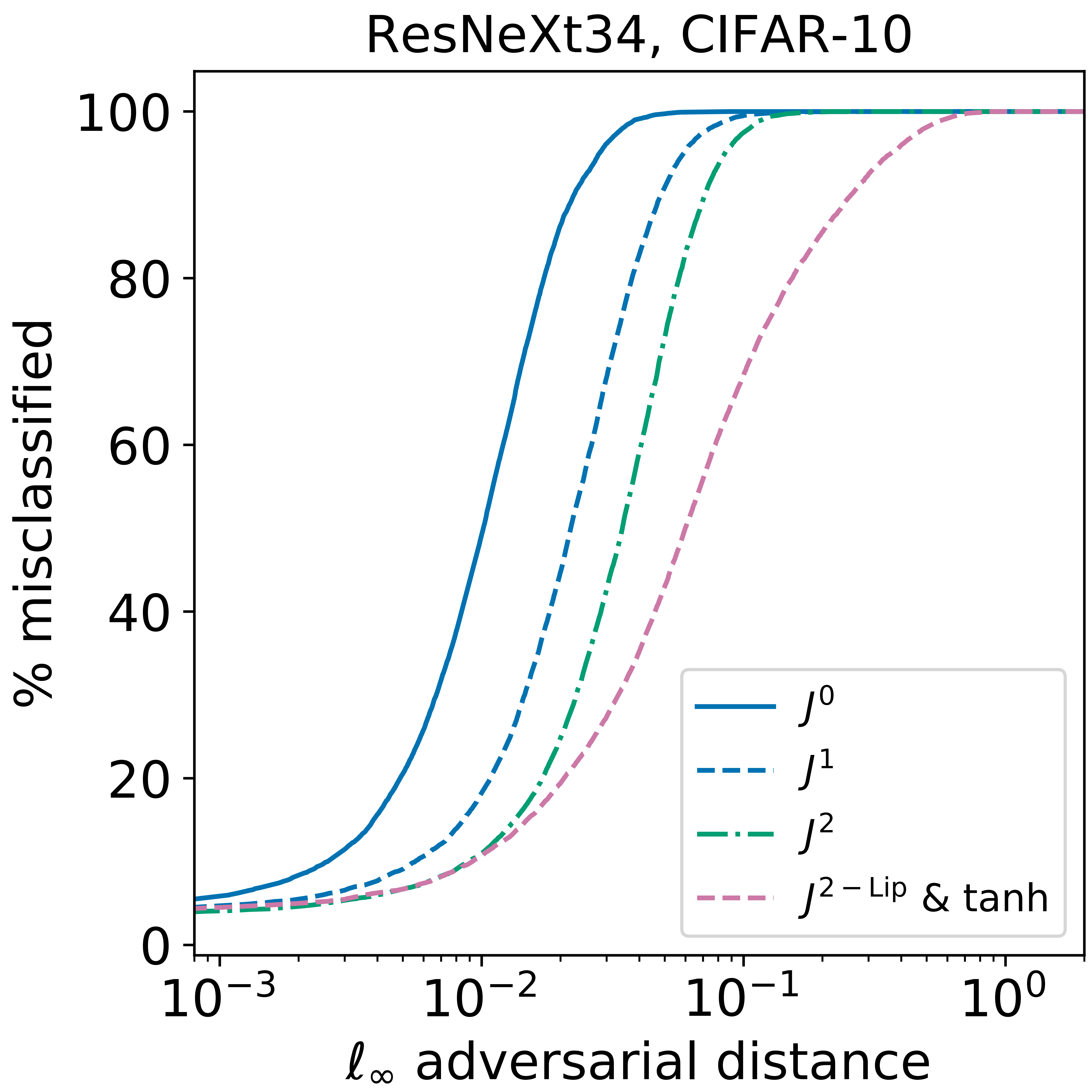
which we refer to as $2-\text{Lip}$ (tulip). In practice, $J^{2-\text{Lip}}$ outperforms J^2 and J^1. For example on CIFAR-10, for a ResNeXt model, adversarial training alone reduced adversarial training error by 29% (measured at adversarial ℓ_2 distance[1] $\varepsilon = 0.1$) over an undefended model. In contrast, J^2 with Lipschitz regularization ($J^{2-\text{Lip}}$) reduces adversarial error by 42% over baseline. See Table 1. We trained with

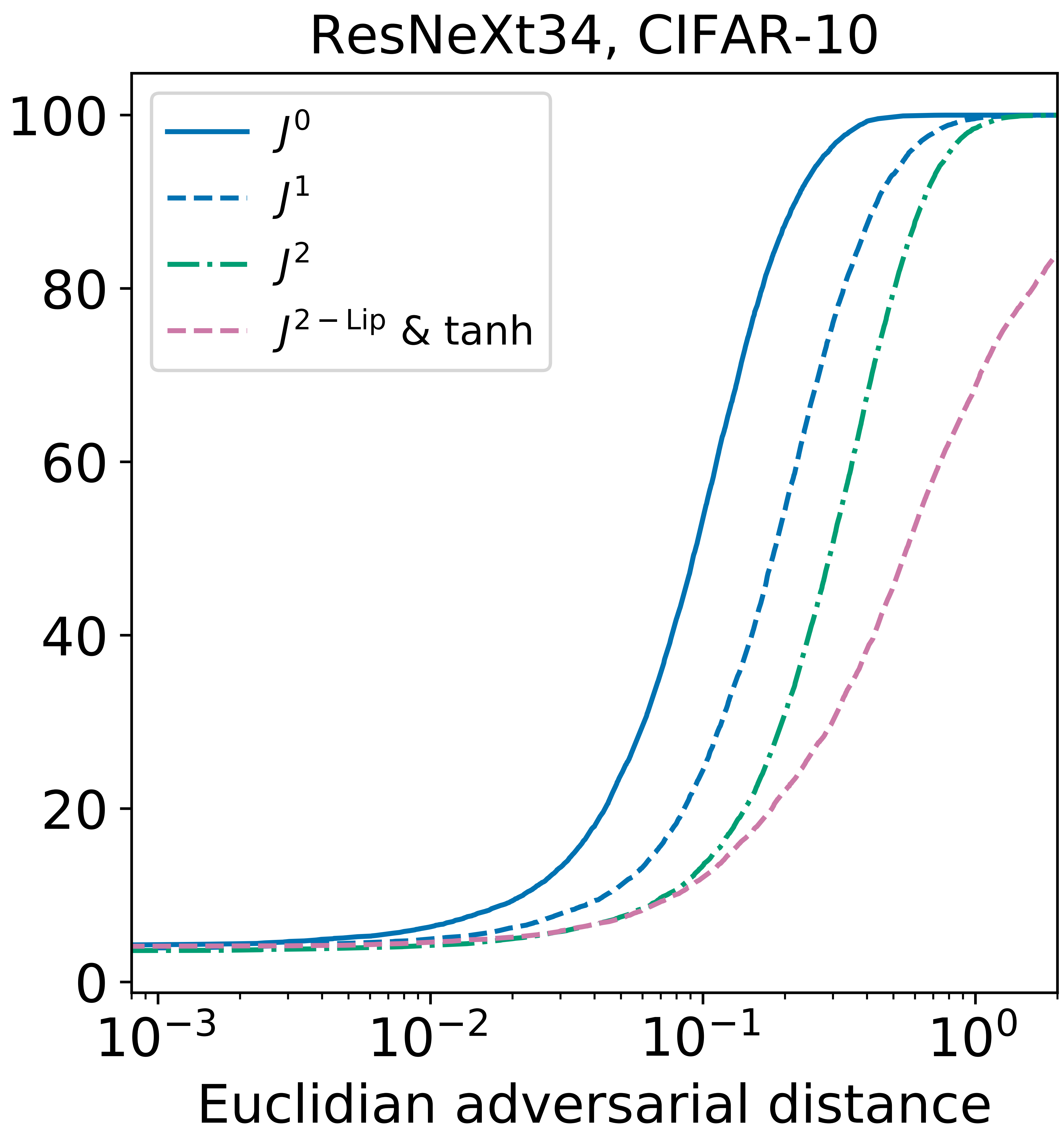
AT + Tulip Results (2-norm)

Dataset	defense method	Euclidean distance: median distance	Euclidean distance: % Err at $\varepsilon = 0.1$	ℓ_∞ distance: median distance	ℓ_∞ 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, ℓ_2)	0.30	13.54	3.45e−2	84.76
	$J^{2-\mathrm{Lip}}$ & tanh	**0.56**	**12.12**	**6.00e−2**	**51.64**
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, ℓ_2)	8.61e−2	53.77	1.06e−2	98.03
	$J^{2-\mathrm{Lip}}$ & tanh	**0.136**	**42.58**	**1.6e−2**	**93.73**

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

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





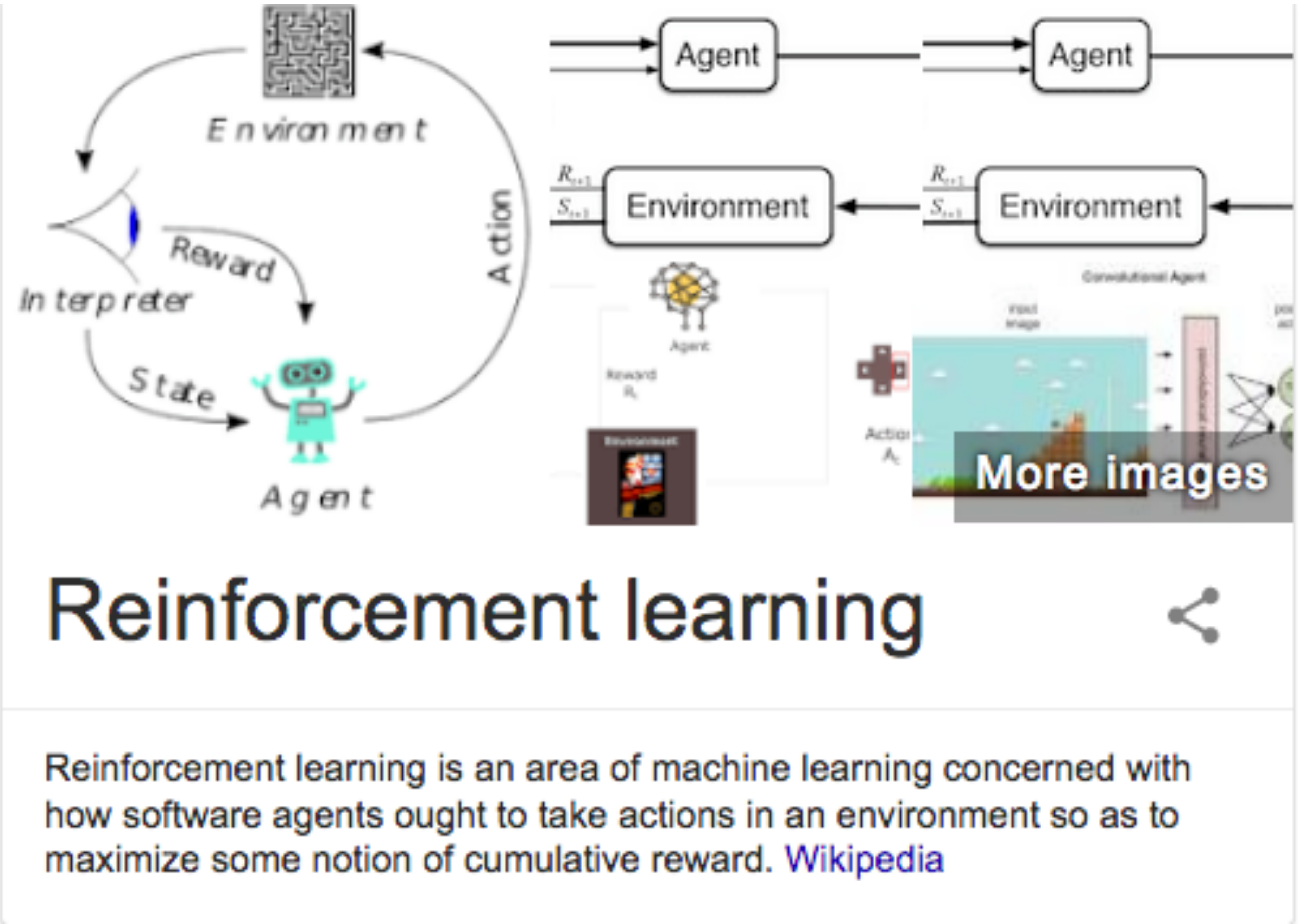
2-Lip > AT-2 > AT-1 > baseline (for all noise levels on both datasets)

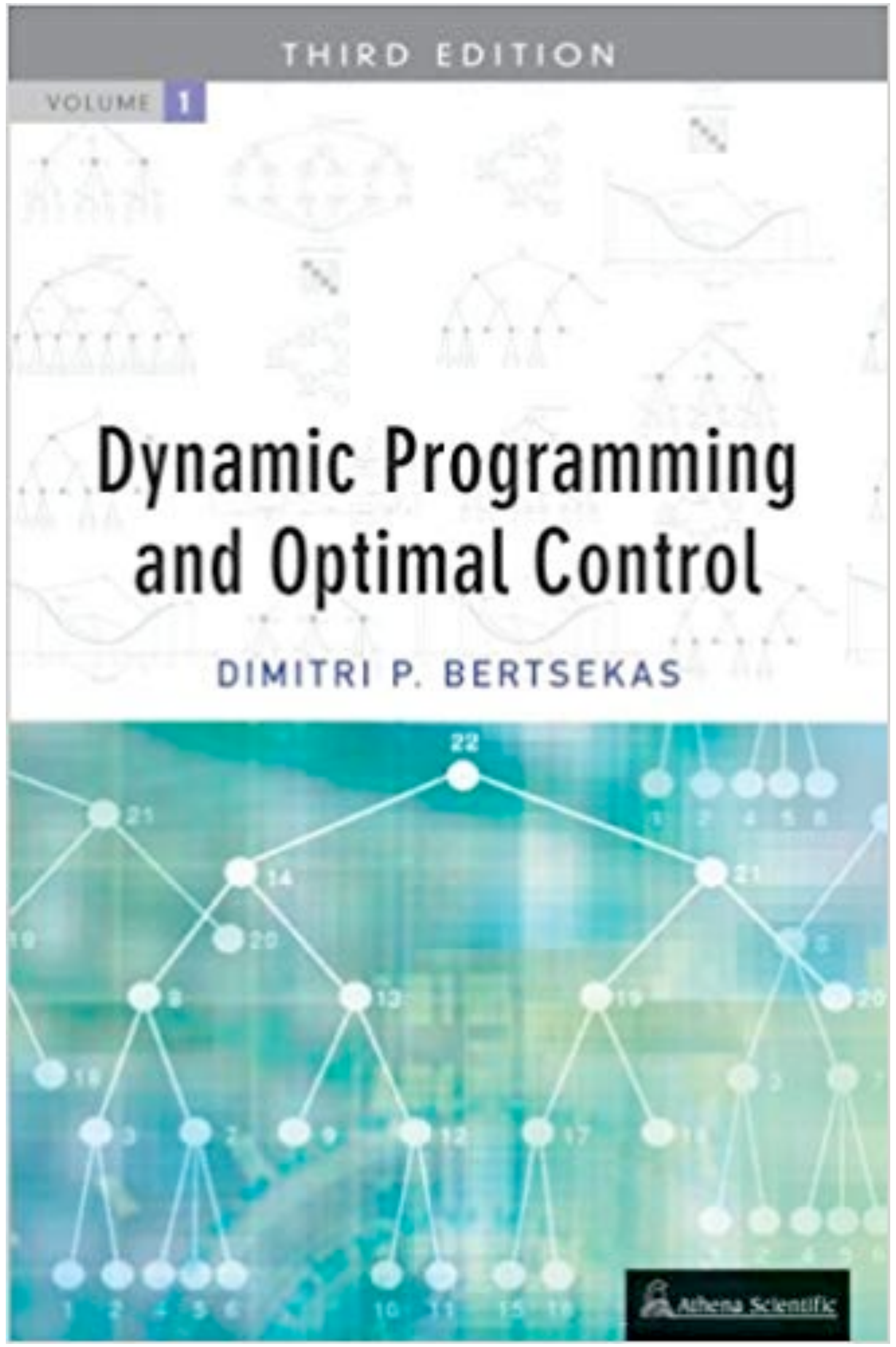
Other current areas of interest in AI with connections to mathematics

We are looking for collaborators: these are possible new projects

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

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

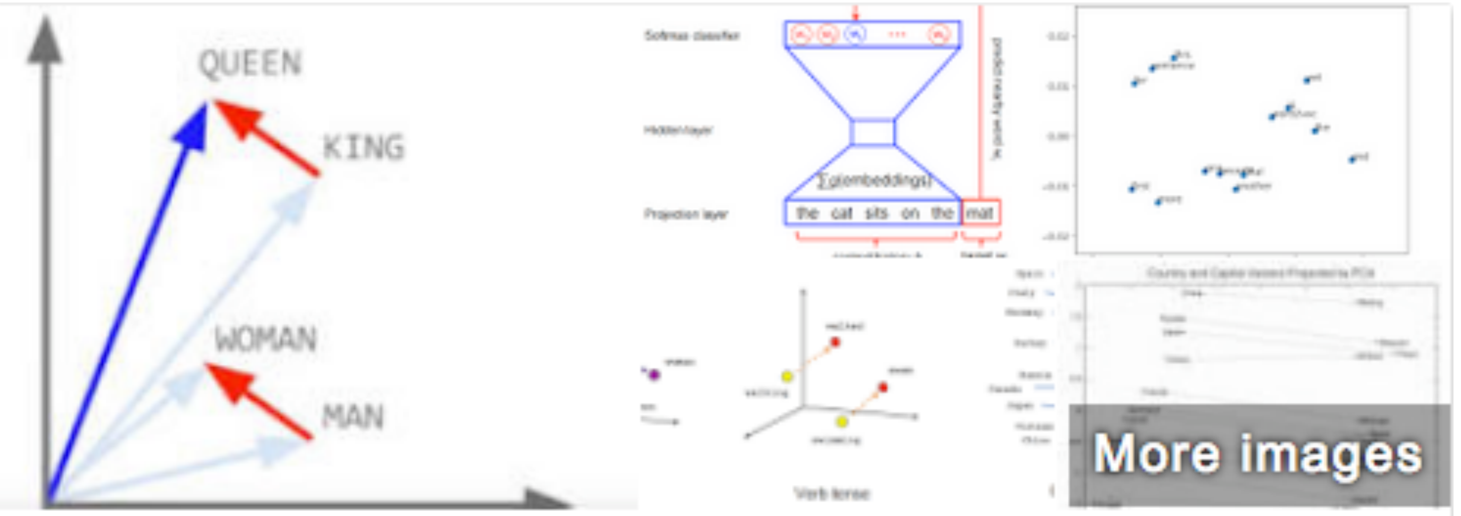
Russian

почему математика интересна Edit

pochemu matematika interesna

French

pourquoi les maths sont-elles intéressantes

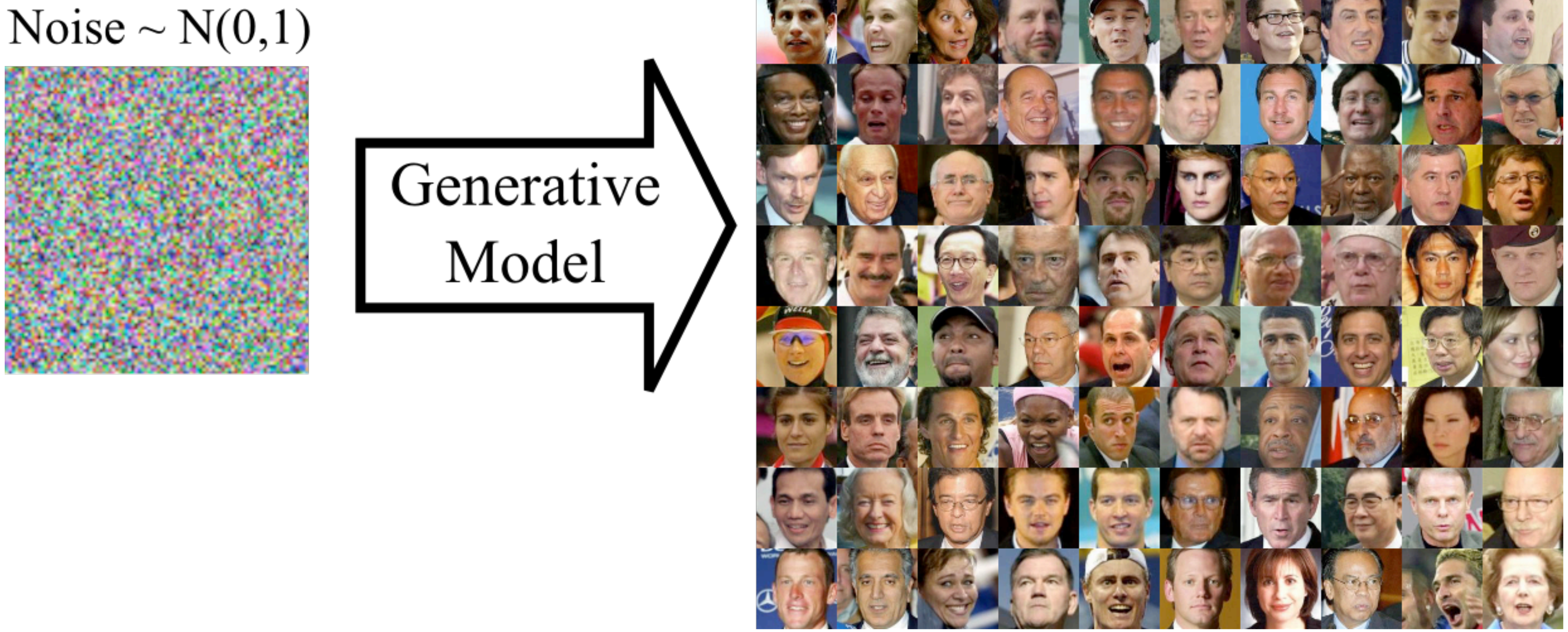


Word2vec

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



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

Squeeze Nets

Computer Science > Computer Vision and Pattern Recognition

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.