MATHEMATICS OF DEEP LEARNING

RAJA GIRYES

TEL AVIV UNIVERSITY

Mathematics of Deep Learning Tutorial

UAI, Tel Aviv, Israel

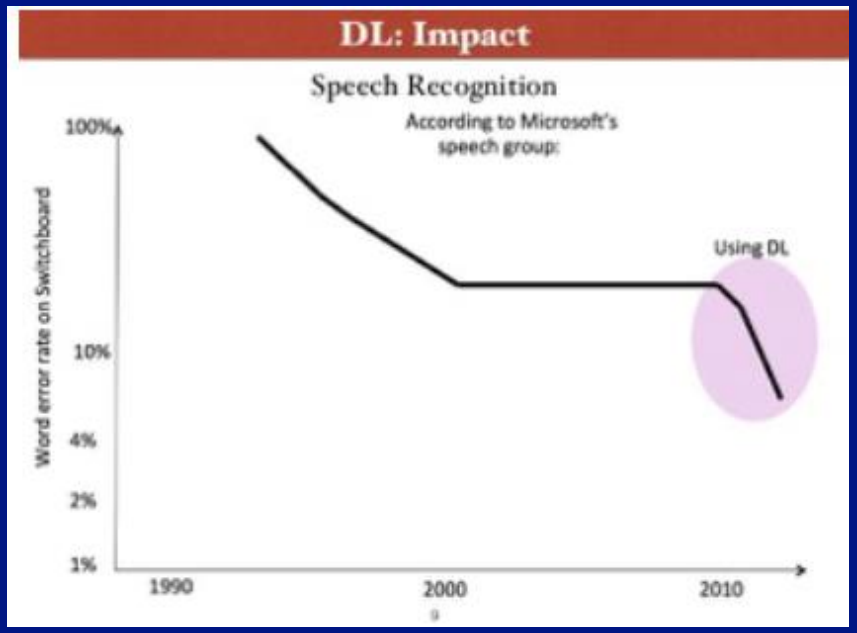
July 22, 2019

DEEP LEARNING IMPACT

- Imagenet dataset
- 1,400,000 images
- 1000 categories
- 150000 for testing,
- 50000 for validation

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Today we get 3.5% by 152 layers



WHY THINGS WORK BETTER TODAY?

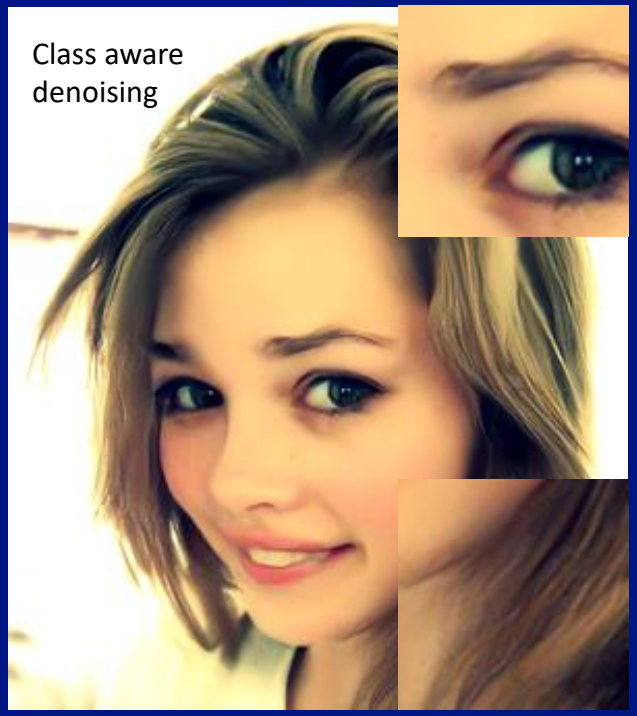
- More data – larger datasets, more access (internet)
- Better hardware (GPU)
- Better learning regularization (dropout)

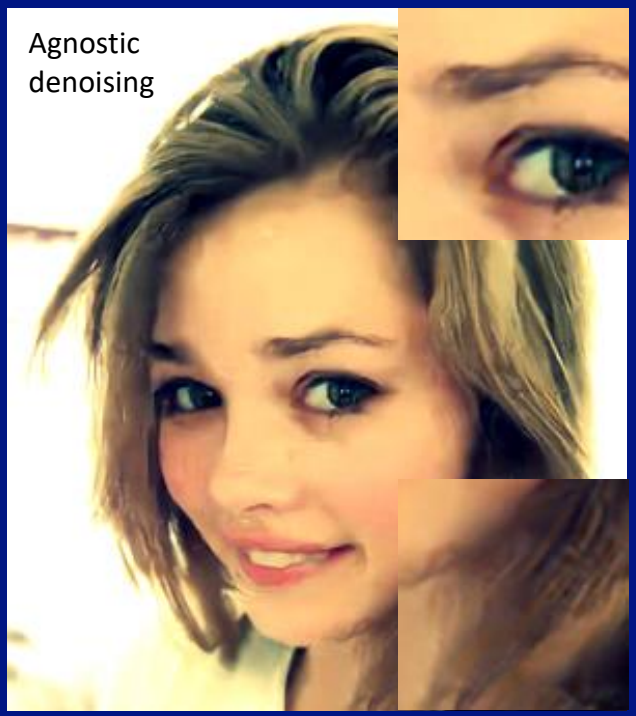
- Deep learning impact and success is not unique only to image classification.
- But it is still unclear why deep neural networks are so remarkably successful and how they are doing it.

CUTTING EDGE PERFORMANCE IN MANY OTHER APPLICATIONS

- Disease diagnosis [Zhou, Greenspan & Shen, 2016].
- Language translation [Sutskever et al., 2014].
- Video classification [Karpathy et al., 2014].
- Handwriting recognition [Poznanski & Wolf, 2016].
- Sentiment classification [Socher et al., 2013].
- Image denoising [Remez et al., 2017].
- Depth Reconstruction [Haim et al., 2017].
- Super-resolution [Kim et al., 2016], [Bruna et al., 2016].
- Error correcting codes [Nahmani, 2016]
- many other applications…

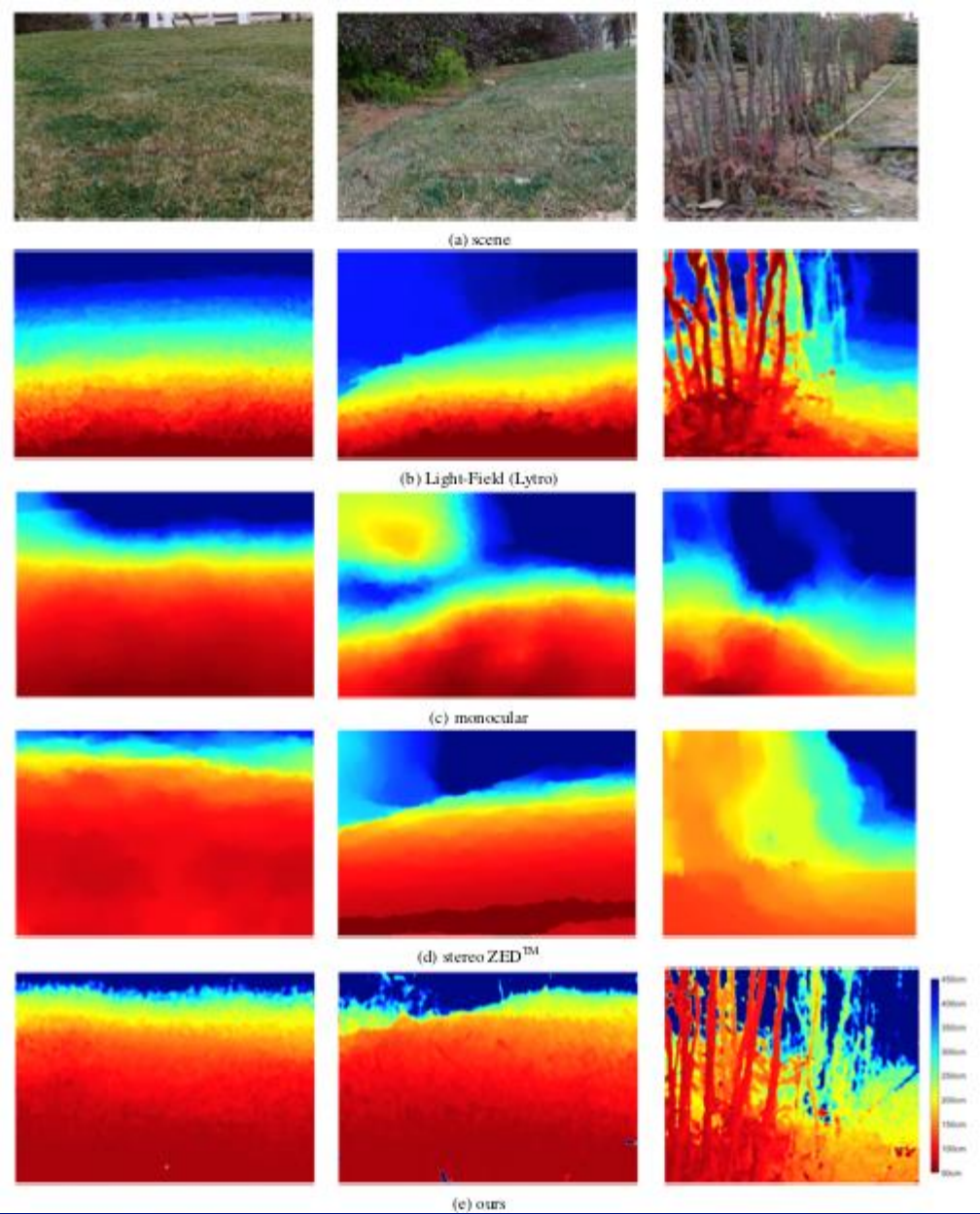
CLASS AWARE DENOISING





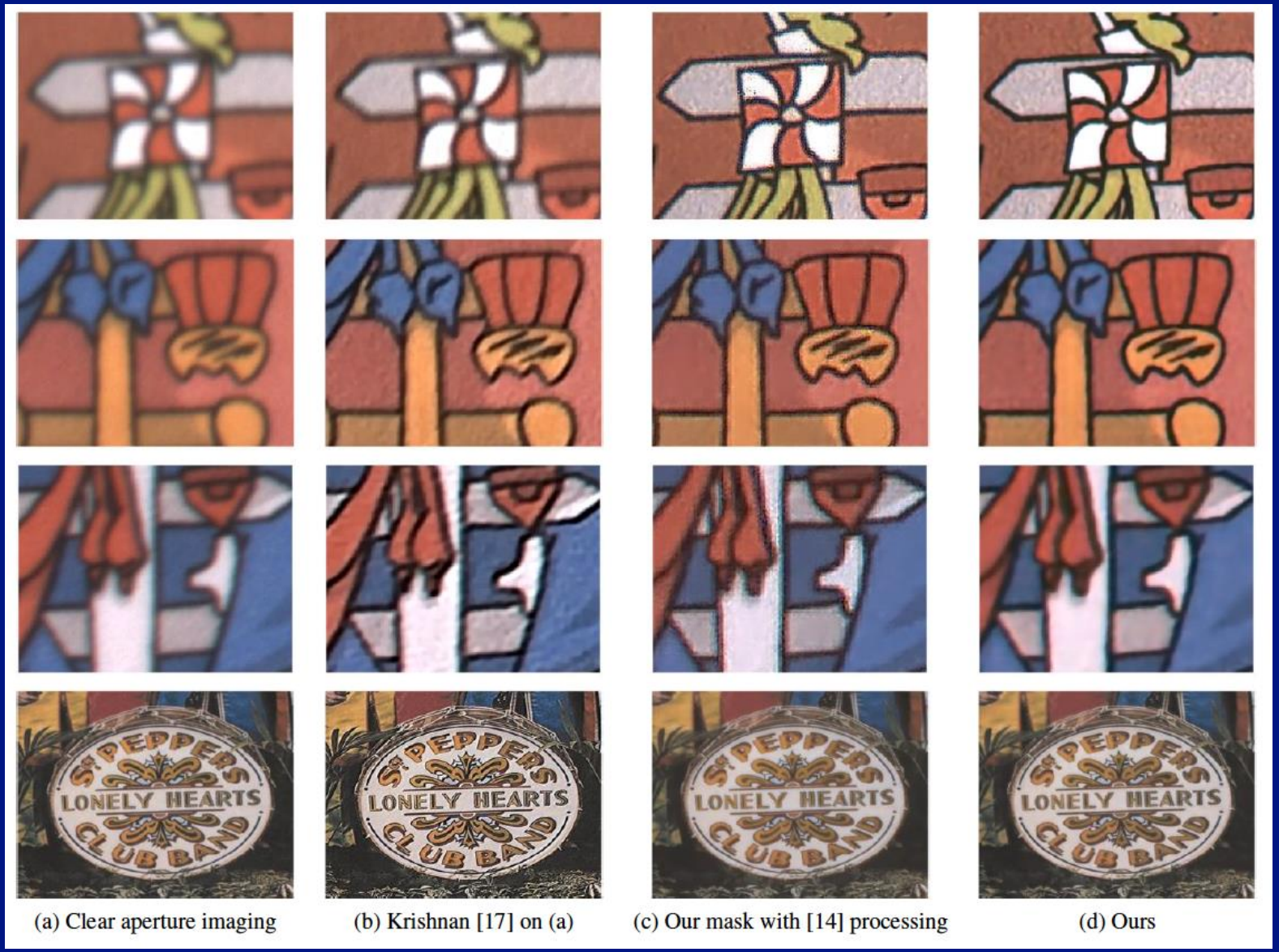
[Remez, Litani, Giryes, Bronstein, 2017]

DEPTH ESTIMATION BY PHASE CODED CUES



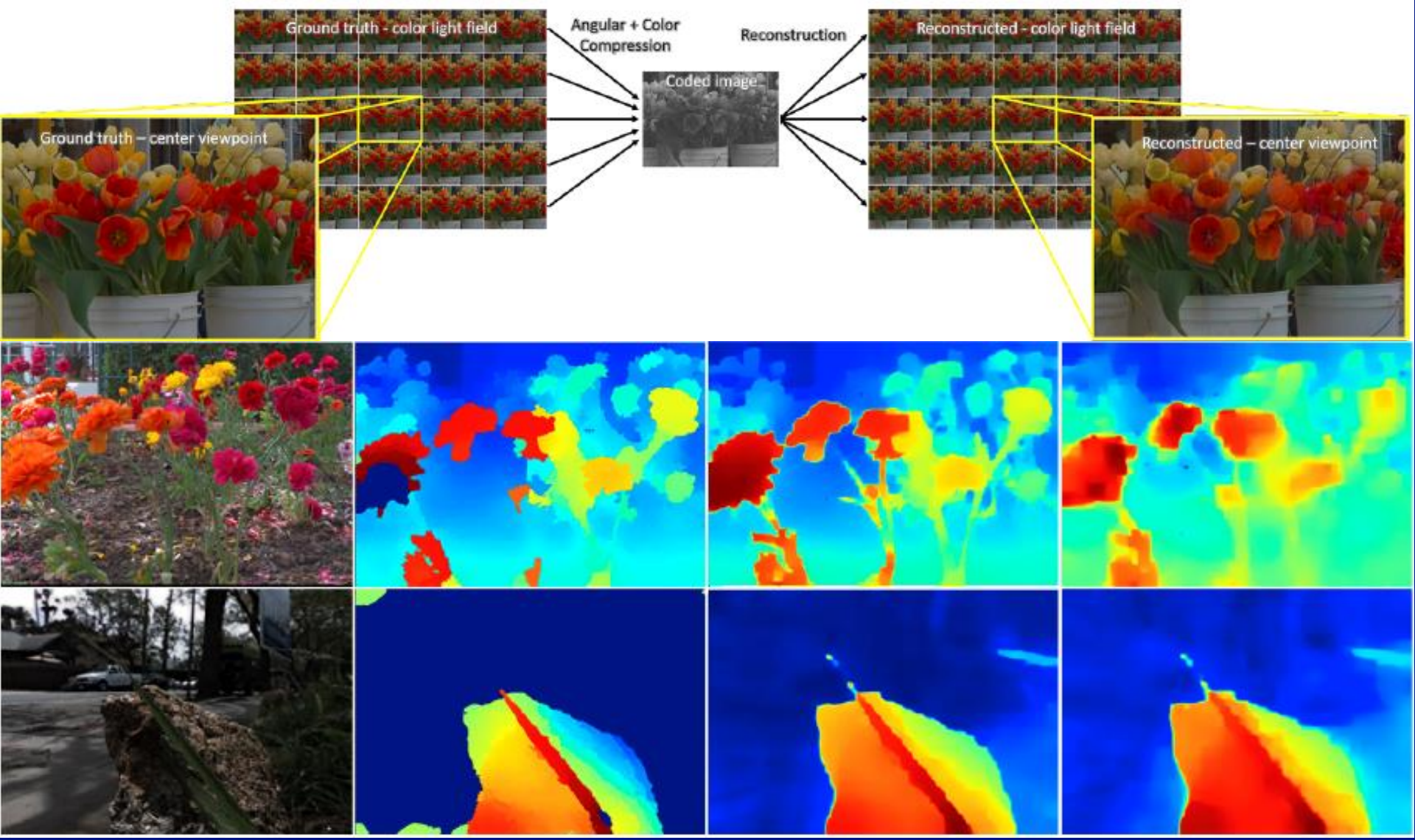
[Haim, Elmalem, Bronstein, Marom, Giryes, 2017]

ALL-IN-FOCUS BY PHASE CODED CUES



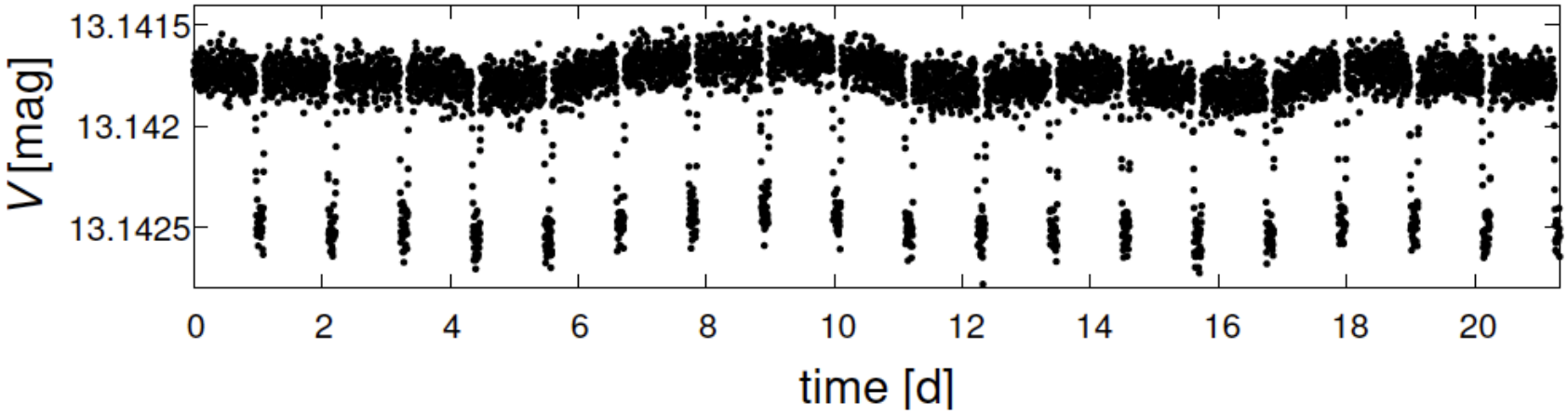
[Elmalem, Marom, Giryes, 2018]

COMPRESSED COLOR LIGHT FIELD

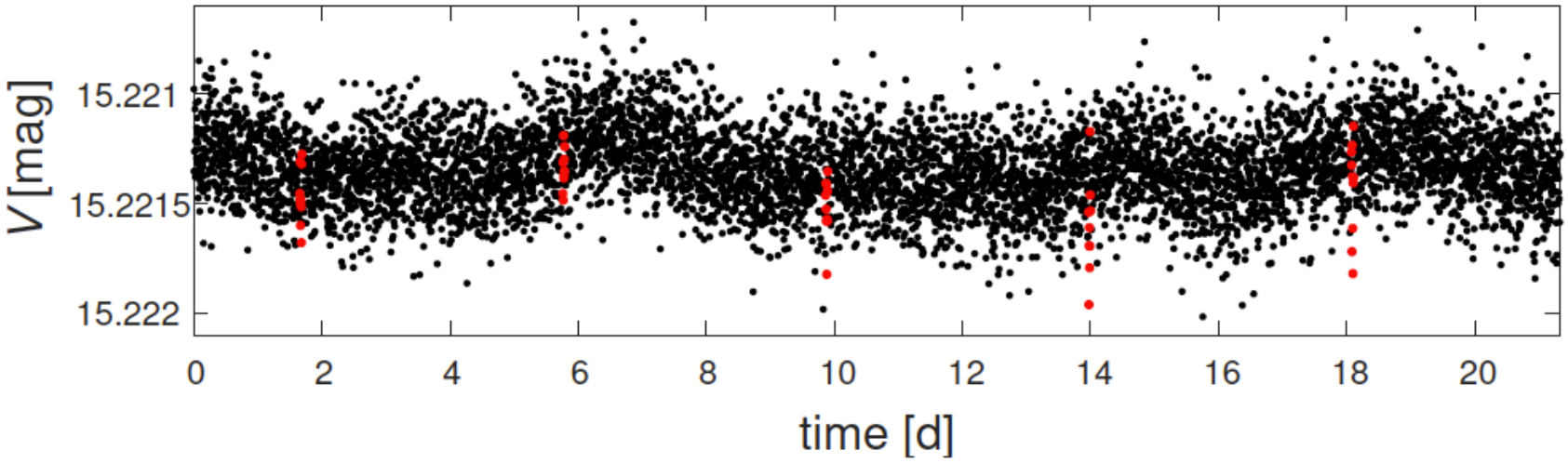


[Yovel, Nabati, Mendelovic, Giryes, 2018]

EXOPLANETS DETECTION

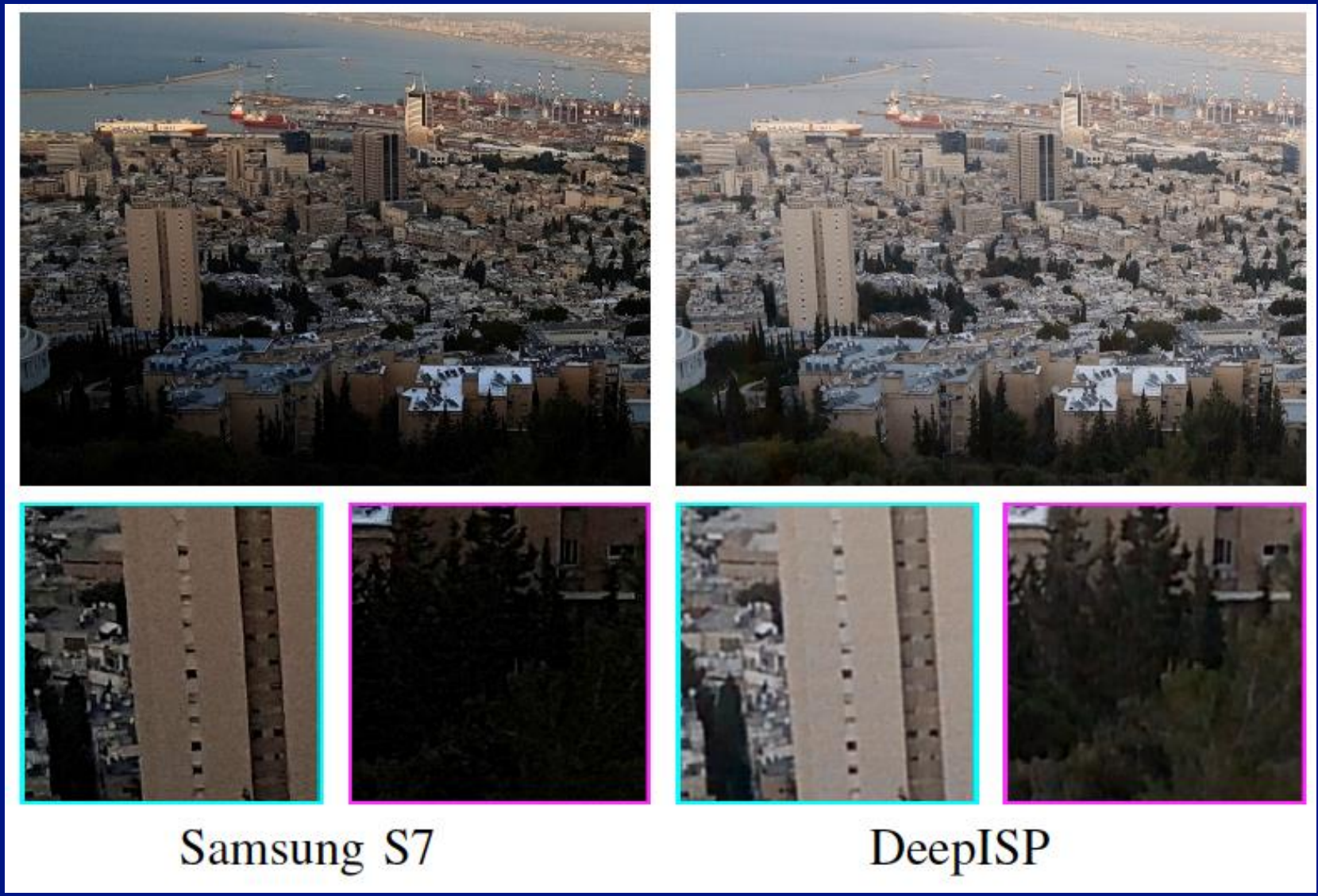


(a)



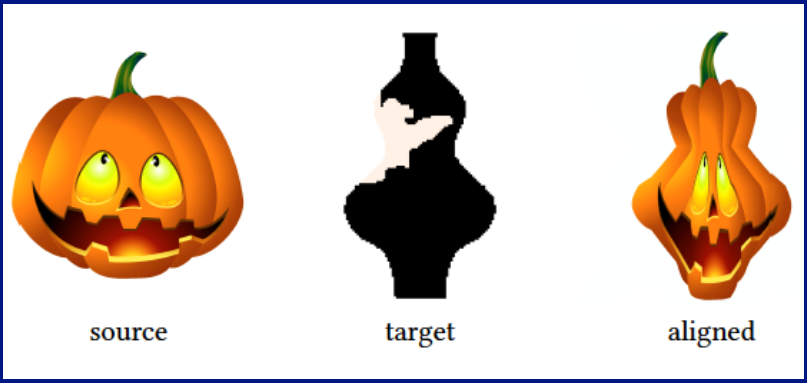
[Zucker and Giryes, 2018]

DEEP ISP

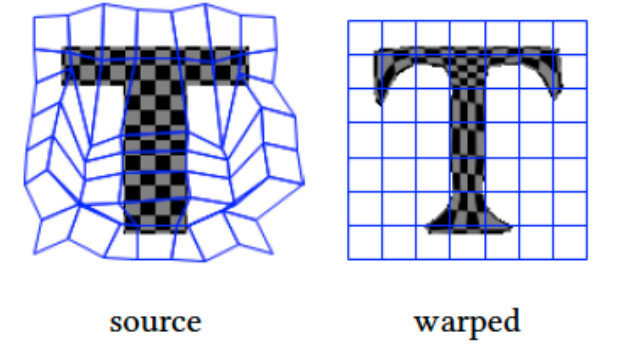


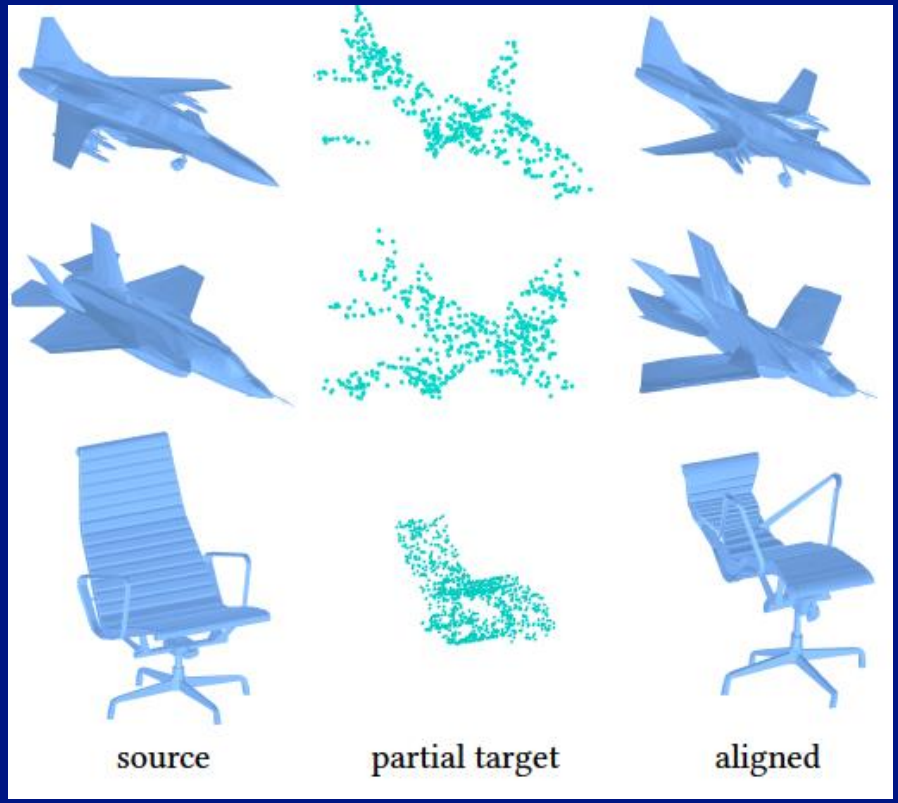
[E. Schwartz, R. Giryes and A. M. Bronstein, 2018]

PARTIAL SHAPE ALIGNMENT



Alignment is performed by a free form deformation generated by a network:

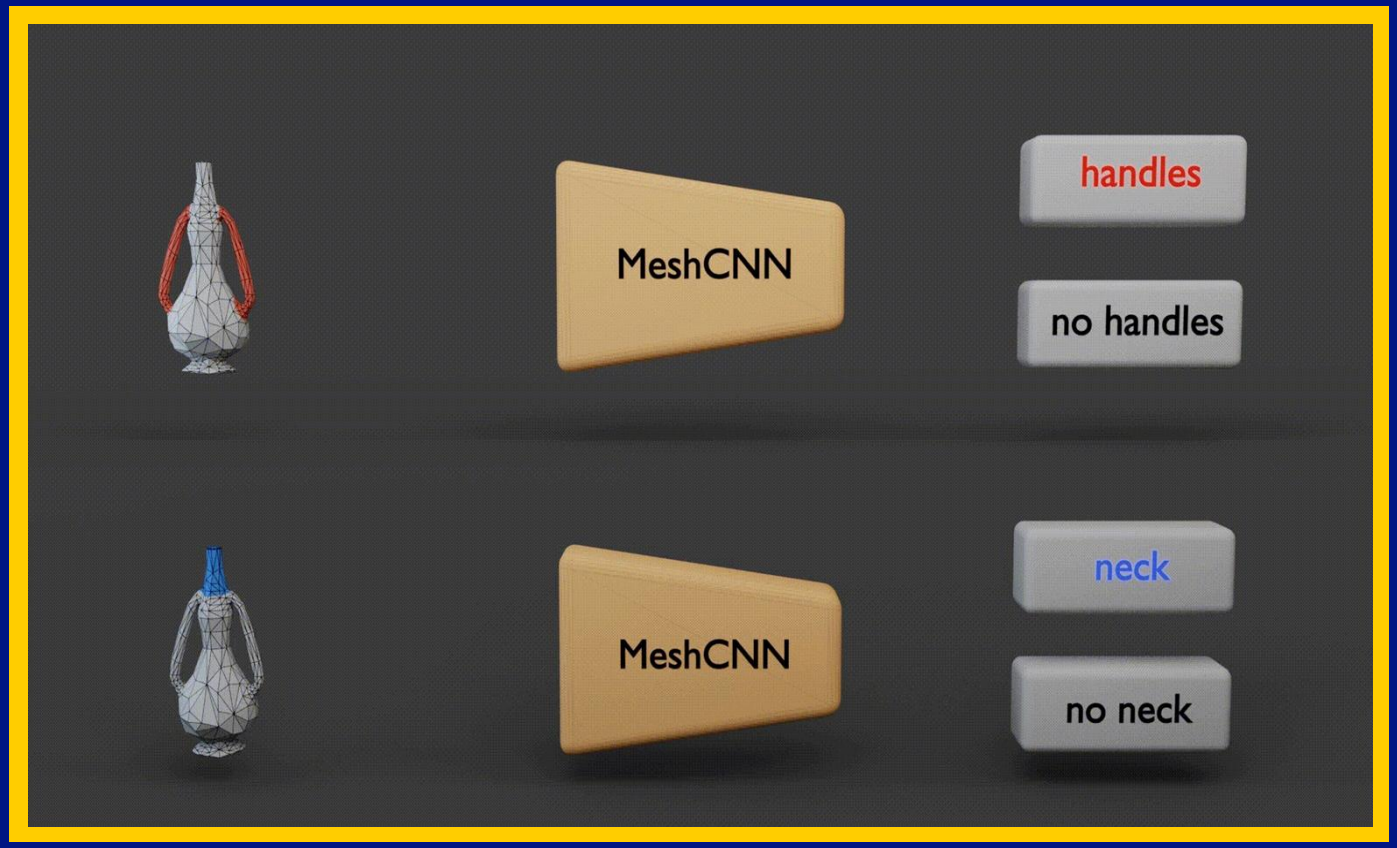




[Hanocka et al., 2018]

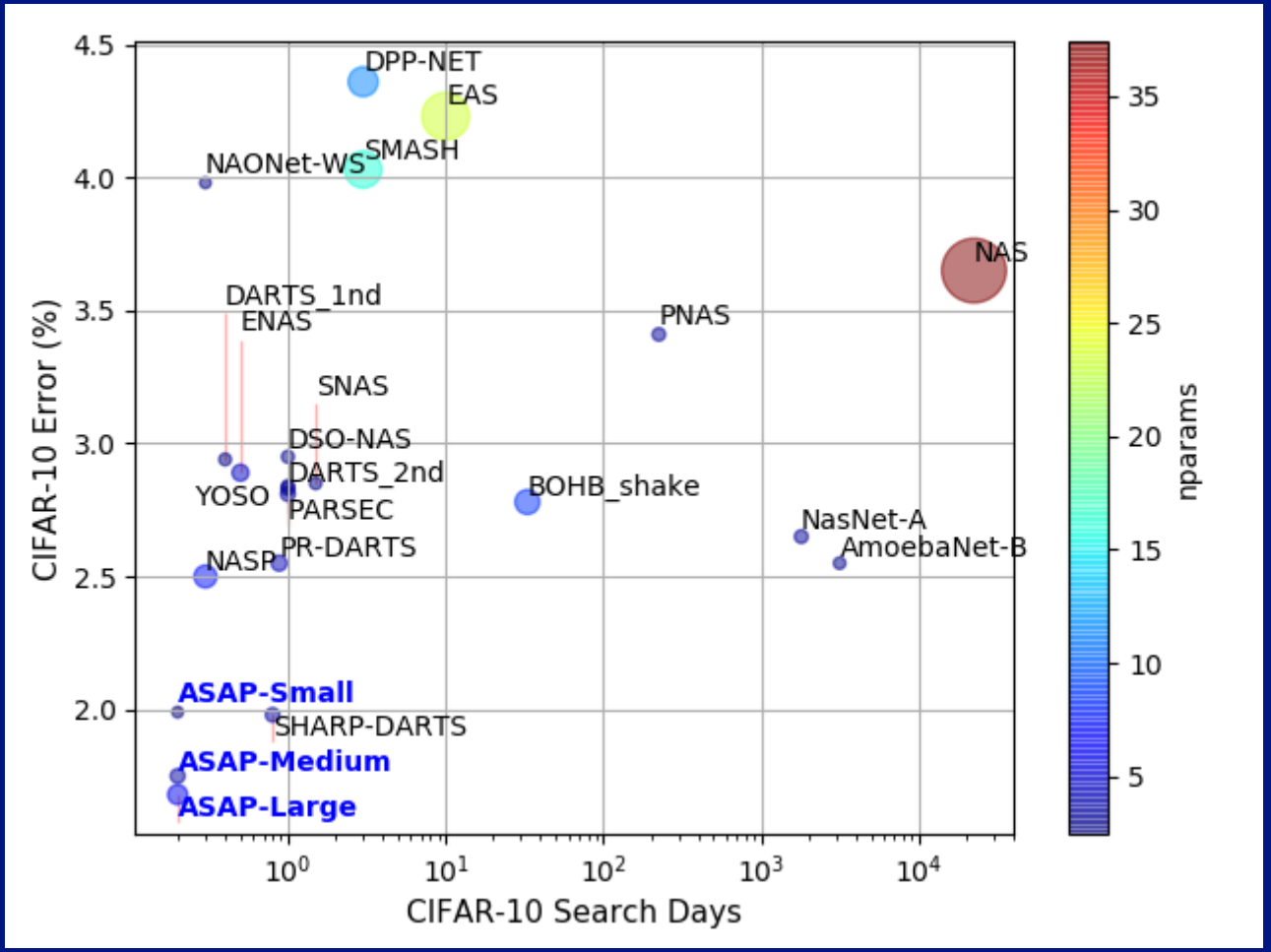
MESH CNN

- A neural network for mesh data
- Perform a different mesh simplification for different tasks.



[Hanocka et al., 2019]

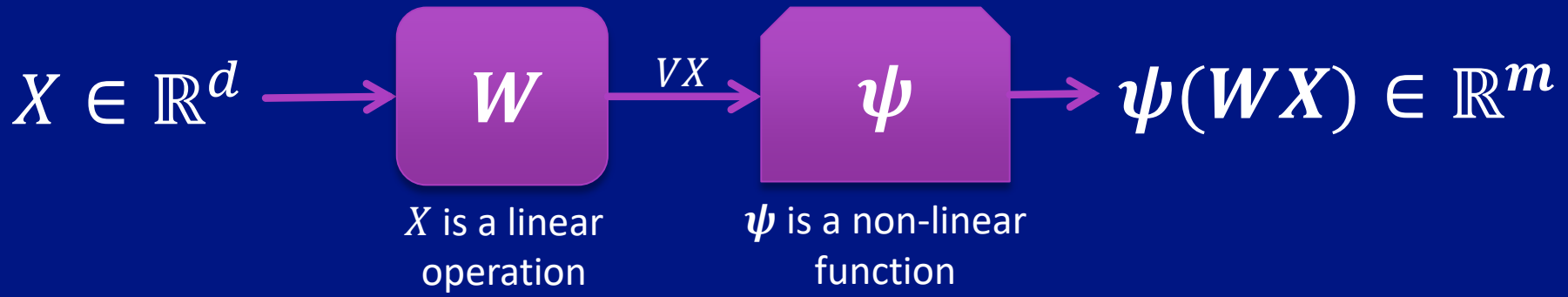
ASAP -NETWORK ARCHITECTURE SEARCH



[Noy et al., 2019]

DEEP NEURAL NETWORKS (DNN)

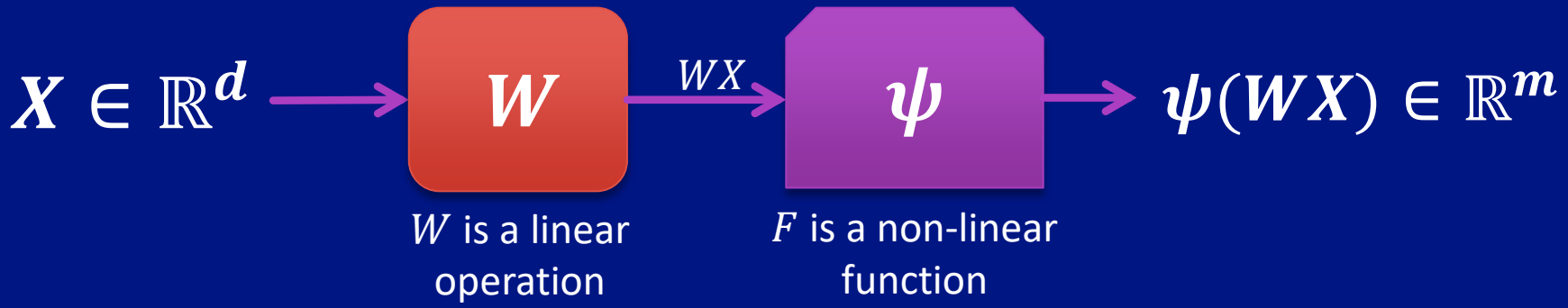
- One layer of a neural net



- Concatenation of the layers creates the whole net

$$\Phi(W^1, W^2, \dots, W^K) = \psi\left(W^K \dots \psi\left(W^2 \psi(W^1 X)\right)\right)$$

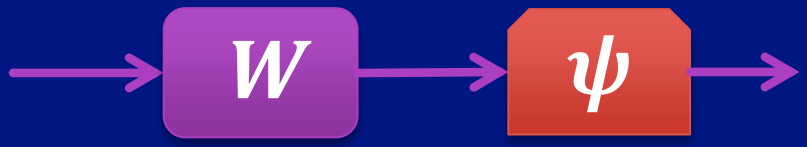
CONVOLUTIONAL NEURAL NETWORKS (CNN)



- In many cases, W is selected to be a convolution.
- This operator is shift invariant.
- CNN are commonly used with images as they are typically shift invariant.

THE NON-LINEAR PART

- Usually $\psi = g \circ f$.



- f is the (point-wise) activation function

ReLU
$f(x) = \max(x, 0)$

Sigmoid
$f(x) = \frac{1}{1 + e^{-x}}$

Hyperbolic tangent
$f(x) = \tanh(x)$

- g is a pooling or an aggregation operator.

Max pooling
$\max_i X_i$

Mean pooling
$\frac{1}{n}\sum_{i=1}^{n} X_i$

l_p pooling
$\sqrt[p]{\sum_{i=1}^{n} X_i^p}$

A SAMPLE OF EXISTING THEORY FOR DEEP LEARNING

WHY DNN WORK?


What is so special with the DNN structure?
What is the role of the depth of DNN?
What is the capability of DNN?
How many training samples do we need?
What is the role of pooling?
What is the role of the activation function?
What happens to the data throughout the layers?

DEEP LEARNING THEORY SURVEY

Mathematics of Deep Learning

René Vidal Joan Bruna Raja Giryes Stefano Soatto

Abstract— **Recently there has been a dramatic increase in the performance of recognition systems due to the introduction of deep architectures for representation learning and classification. However, the mathematical reasons for this success remain elusive. This tutorial will review recent work that aims to provide a mathematical justification for several properties of deep networks, such as global optimality, geometric stability, and invariance of the learned representations.**

I. Introduction

Deep networks [1] are parametric models that perform sequential operations on their input data. Each such operation, colloquially called a "layer", consists of a linear transformation, say, a convolution of its input, followed by a pointwise nonlinear "activation function", e.g., a sigmoid. Deep networks have recently led to dramatic improvements in

sigmoidal activations are universal function approximators [5], [6], [7], [8]. However, the capacity of a wide and shallow network can be replicated by a deep network with significant improvements in performance. One possible explanation is that deeper architectures are able to better capture invariant properties of the data compared to their shallow counterparts. In computer vision, for example, the category of an object is invariant to changes in viewpoint, illumination, etc. While a mathematical analysis of why deep networks are able to capture such invariances remains elusive, recent progress has shed some light on this issue for certain sub-classes of deep networks. In particular, scattering networks [9] are a class of deep networks whose convolutional filter banks are given by complex, **multi-resolution wavelet families.** As a result of this extra structure, they are provably stable

SAMPLE OF RELATED EXISTING THEORY

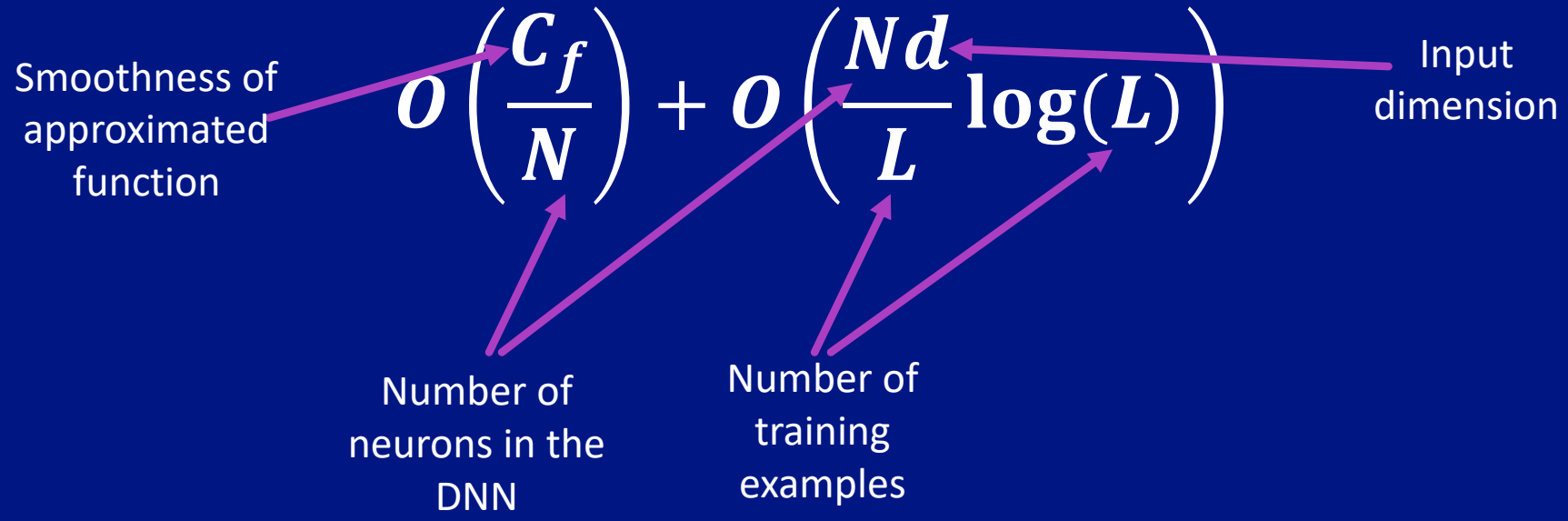
- Universal approximation for any measurable Borel functions [Hornik et. al., 1989, Cybenko 1989, Daniely et al., 2017]
- Depth of a network provides an exponential complexity compared to the number parameters [Montúfar et al. 2014, Cohen et al. 2016, Eldan & Shamir, 2016] and invariance to more complex deformations [Bruna & Mallat, 2013]
- Number of training samples scales as the number of parameters [Shalev-Shwartz & Ben-David 2014] or the norm of the weights in the DNN [Neyshabur et al. 2015]
- Pooling relation to shift invariance and phase retrieval [Bruna et al. 2013, 2014]
- Deeper networks have more local minima that are close to the global one and less saddle points [Saxe et al. 2014], [Dauphin et al. 2014], [Choromanska et al. 2015], [Haeffele & Vidal, 2015], [Soudry & Hoffer, 2017]
- Relation to dictionary learning [Papayan et al. 2016].
- Information bottleneck [Shwartz-Ziv & Tishby, 2017], [Tishby & Zaslavsky 2017].
- Invariant representation for certain tasks [Soatto & Chiuso, 2016]
- Bayesian deep learning [Kendall and Gal. 2017] [Patel, Nguyen & Baraniuk , 2016]

REPRESENTATION POWER

- Neural nets serve as a universal approximation for any measurable Borel functions [Cybenko 1989, Hornik 1991].
- In particular, let the non-linearity ψ be a bounded, non-constant continuous function, I_d be the d-dimensional hypercube, and $C(I_d)$ be the space of continuous functions on I_d. Then for any $f \in C(I_d)$ and $\epsilon > 0$, there exists $m > 0$, and $X \in \mathbb{R}^{d \times m}$, $B \in \mathbb{R}^m$, $W \in \mathbb{R}^m$ such that the neural network
$$F(V) = \psi(VX + B)W^T$$
approximates f with a precision ϵ:
$$|F(V) - f(V)| < \epsilon, \forall V \in \mathbb{R}^d$$

ESTIMATION ERROR

- The estimation error of a function f by a neural networks scales as [Barron 1994].



DEPTH OF THE NETWORK

- Depth allow representing shallow restricted Boltzmann machines, which has an exponential number of parameters, compared to the deep one [Montúfar & Morton, 2015]
- Each DNN layer with ReLU divides the space by a hyper-plane, folding one part of it.
- Thus, the depth of the network folds the space into an exponential number of sets compared to the number of parameters [Montúfar, Pascanu, Cho & Bengio, 2014]

DEPTH EFFICIENCY OF CNN

- Function realized by CNN, with ReLU and max-pooling, of polynomial size requires super-polynomial size for being approximated by shallow network [Telgarsky 2016 ,Cohen et al., 2016].
- Standard convolutional network design has learning bias towards statistics of natural images [Cohen et al., 2016].

ROLE OF POOLING

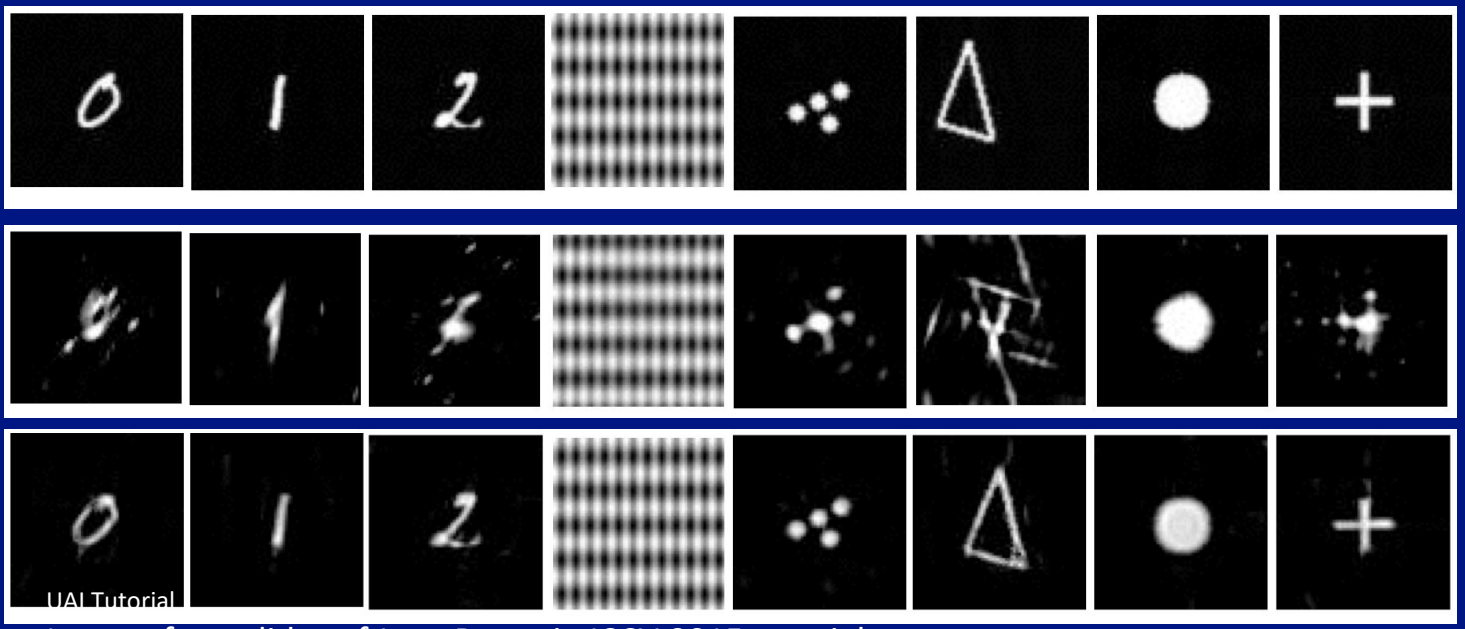
- The pooling stage provides shift invariance [Boureau et al. 2010], [Bruna, LeCun & Szlam, 2013].
- A connection is drawn between the pooling stage and the phase retrieval methods [Bruna, Szlam & LeCun, 2014].
- This allows calculating Lipchitz constants of each DNN layer $\psi(\cdot\, X)$ and empirically recovering the input of a layer from its output.
- However, the Lipchitz constants calculated are very loose and no theoretical guarantees are given for the recovery.

SUFFICIENT STATISTIC AND INVARIANCE

- Given a certain task at hand:
- Minimal sufficient statistic guarantees that we can replace raw data with a representation with smallest complexity and no performance loss.
- Invariance guarantees that the statistic is constant with respect to uninformative transformations of the data.
- CNN are shown to have these properties for many tasks [Soatto & Chiuso, 2016].
- Good structures of deep networks can generate representations that are good for learning with a small number of examples [Anselmi et al., 2016].

SCATTERING TRANSFORMS

- Scattering transform - a cascade of wavelet transform convolutions with nonlinear modulus and averaging operators.
- Scattering coefficients are stable encodings of geometry and texture [Bruna & Mallat, 2013]



Original image with d pixels

Recovery from first scattering moments: $O(\log d)$ coefficients

Recovery from 1st & 2nd scattering moments: $O(\log^2 d)$ coefficients

Images from slides of Joan Bruna in ICCV 2015 tutorial

SCATTERING TRANSFORMS AND DNN

- More layers create features that can be made invariant to increasingly more complex deformations.
- Deep layers in DNN encode complex, class-specific geometry.
- Deeper architectures are able to better capture invariant properties of objects and scenes in images
  [Bruna & Mallat, 2013], [Wiatowski & Bölcskei, 2016]

SCATTERING TRANSFORMS AS A METRIC

- Scattering transforms may be used as a metric.
- Inverse problems can be solved by minimizing distance at the scattering transform domain.
- Leads to remarkable results in super-resolution [Bruna, Sprechmann & Lecun, 2016]

SCATTERING SUPER RESOLUTION

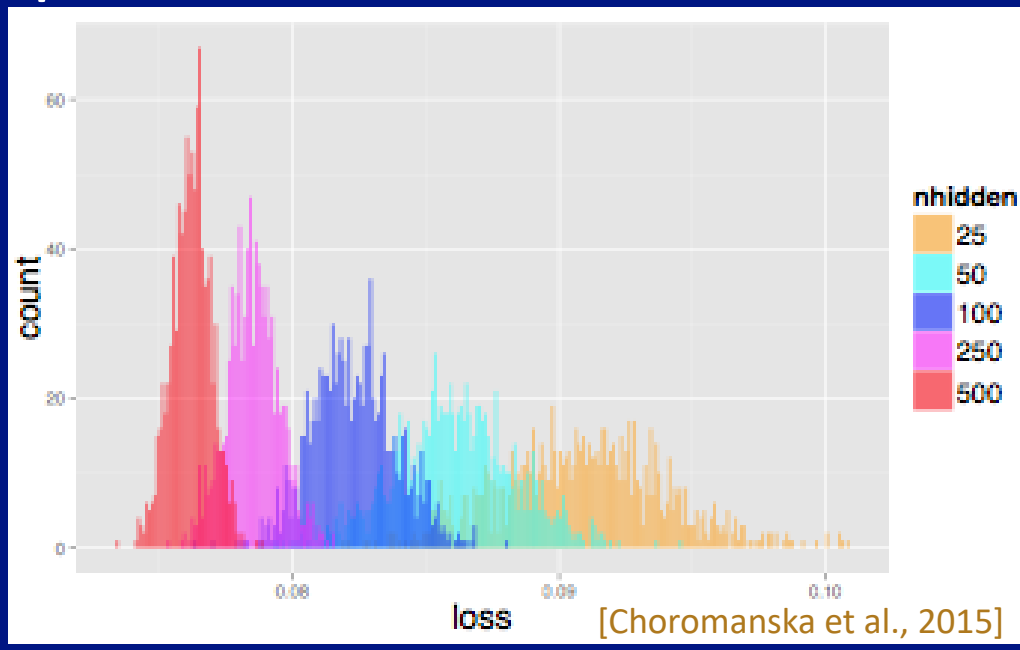
Original | Best Linear Estimate | State-of-the-art | Scattering estimate

[Bruna, Sprechmann & Lecun, 2016]

Images from slides of Joan Bruna in CVPR 2016 tutorial

MINIMIZATION

- The local minima in deep networks are not far from the global minimum.
- saddle points are the main problem of deep Learning optimization.
- Deeper networks have more local minima but less saddle points.

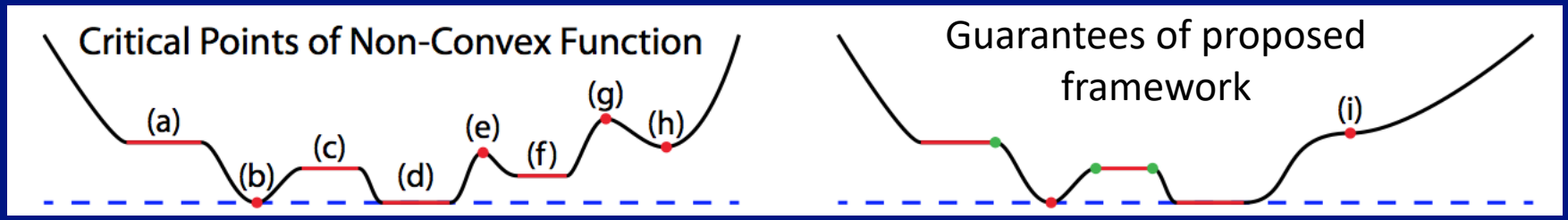


[Choromanska et al., 2015]

[Saxe, McClelland & Ganguli, 2014], [Dauphin, Pascanu, Gulcehre, Cho, Ganguli & Bengio, 2014] [Choromanska, Henaff, Mathieu, Ben Arous & LeCun, 2015]

GLOBAL OPTIMALITY IN DEEP LEARNING

- Deep learning is a positively homogeneous factorization problem, i.e., $\exists p \geq 0$ such that $\forall \alpha \geq 0$ DNN obey
  $\Phi\left(\alpha X^1, \alpha X^2, \ldots, \alpha X^K\right) = \alpha^p \Phi\left(X^1, X^2, \ldots, X^K\right)$.
- With proper regularization, local minima are global.
- If the network is large enough, global minima can be found by local descent.

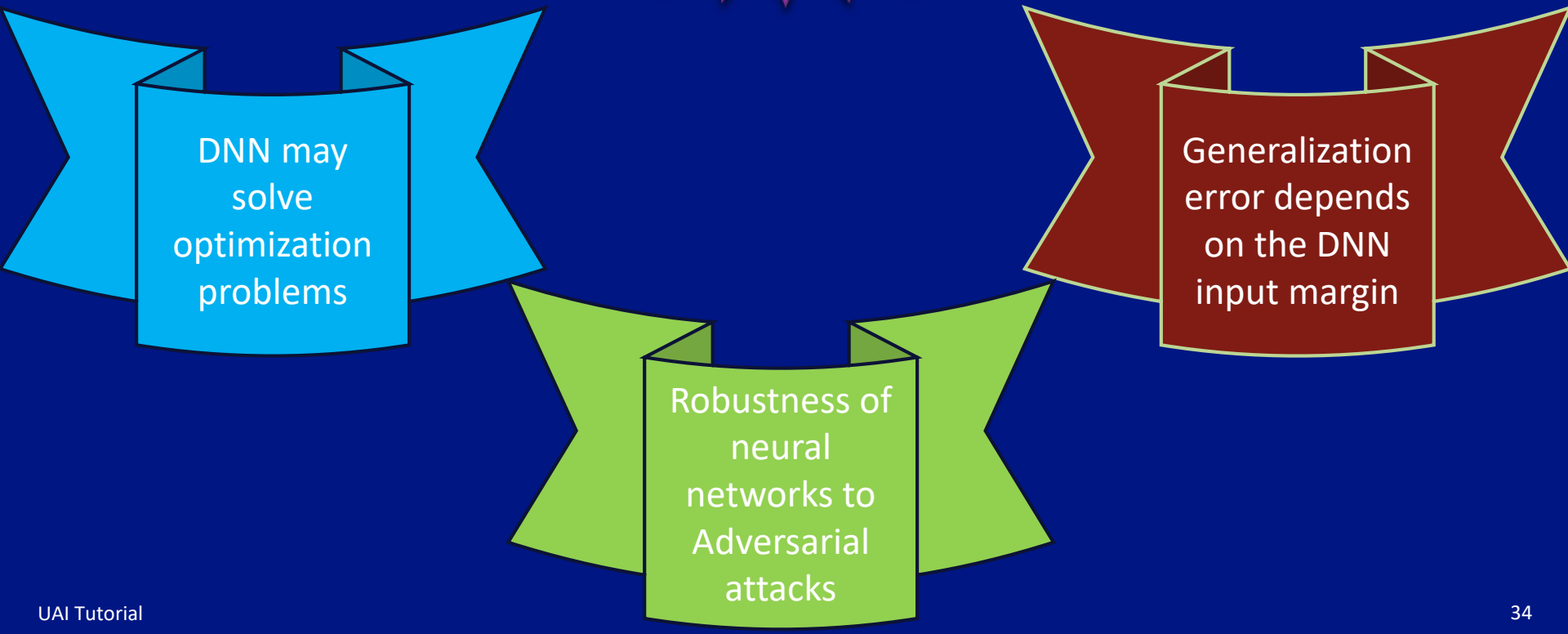


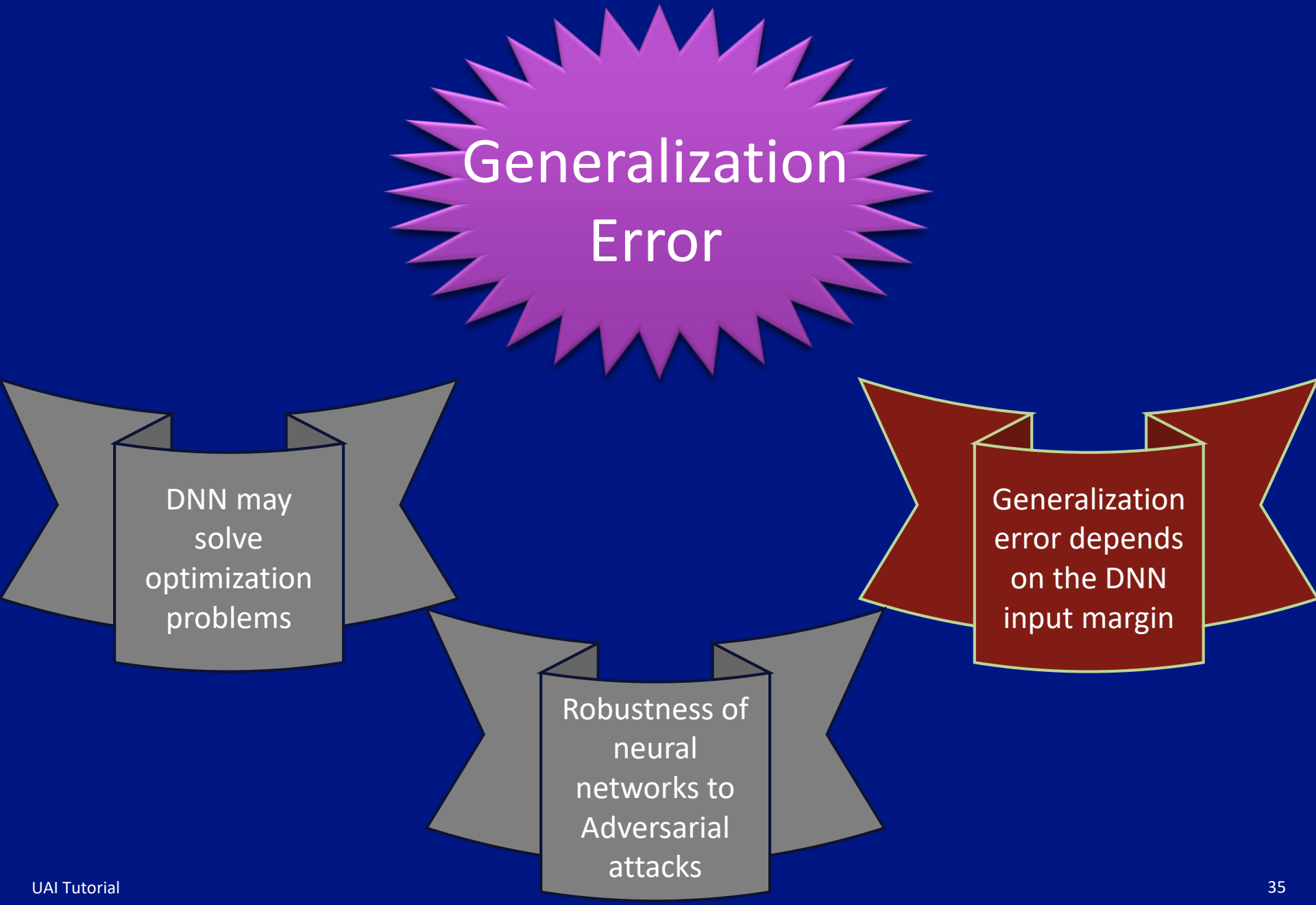
[Haeffele & Vidal, 2015]

OUR THEORY

- DNN Classification is affected by the angles in the data [Giryes et al. 2016].
- Generalization error of neural network [Sokolic, Giryes, Sapiro & Rodrigues, 2017].
- Relationship between invariance and generalization in deep learning [Sokolic, Giryes, Sapiro & Rodrigues, 2017].
- Solving optimization problems with neural networks [Giryes, Eldar, Bronstein & Sapiro, 2018].
- Robustness to adversarial examples [Jakubovitz & Giryes, 2018].
- Robustness to label noise [Drory, Avidan & Giryes, 2019].
- Observation of k-NN behavior in neural networks to explain the coexistence of memorization and generalization in neural networks [Cohen, Sapiro & Giryes, 2018].
- Relationship between ETF and dropout [Bank & Giryes, 2019].
- Lautum information for transfer learning [Jakubovitz & Giryes, 2018].

Outline


DNN may solve optimization problems
Robustness of neural networks to Adversarial attacks
Generalization error depends on the DNN input margin


Generalization Error
DNN may solve optimization problems
Robustness of neural networks to Adversarial attacks
Generalization error depends on the DNN input margin

GENERALIZATION ERROR SURVEY

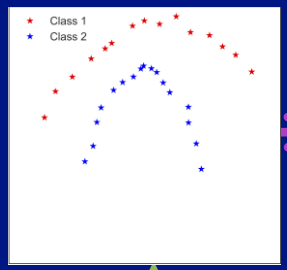
Generalization Error in Deep Learning

Daniel Jakubovitz[1], Raja Giryes[1], and Miguel R. D. Rodrigues[2]

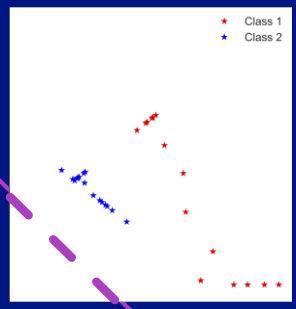
[1] School of Electrical Engineering,
Tel Aviv University, Israel
danielshaij@mail.tau.ac.il, raja@tauex.tau.ac.il
[2] Department of Electronic and Electrical Engineering,
University College London, UK
m.rodrigues@ucl.ac.uk

Abstract. Deep learning models have lately shown great performance in various fields such as computer vision, speech recognition, speech translation, and natural language processing. However, alongside their state-of-the-art performance, it is still generally unclear what is the source of their generalization ability. Thus, an important question is what makes deep neural networks able to generalize well from the training set to new data. In this article, we provide an overview of the existing theory and bounds for the characterization of the generalization error of deep neural networks, combining both classical and more recent theoretical and empirical results.

ASSUMPTIONS



W^1 ψ W^i ψ W^K ψ

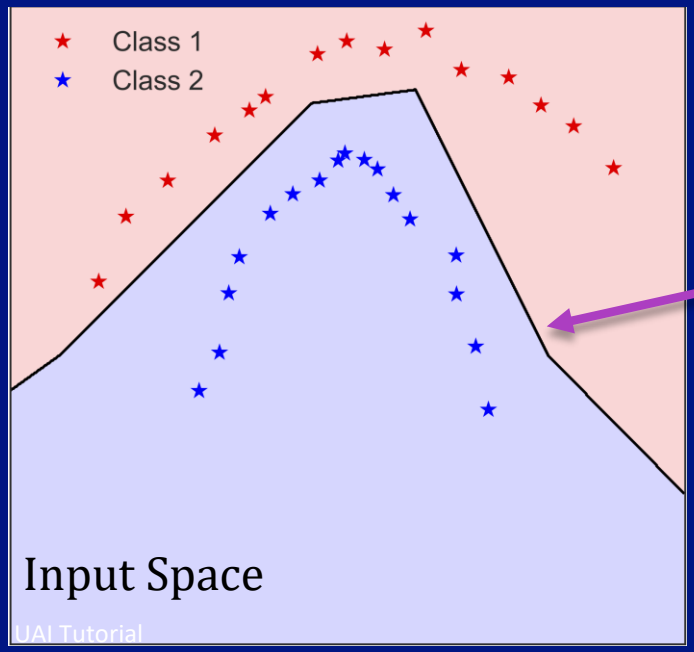


Two Classes $\in \Upsilon$

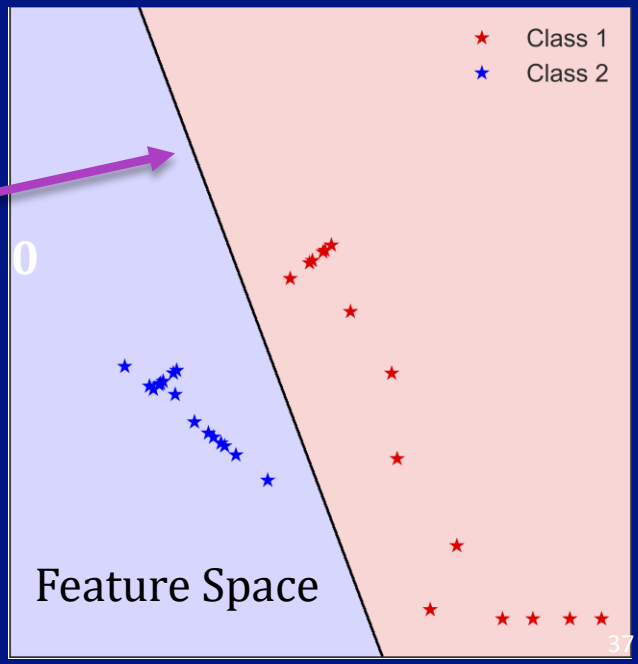
general non-linearity (ReLU, pooling,…)

softmax/ linear classifier

w



$w^T \Phi(W^1, W^2, \ldots, W^K) = 0$



GENERALIZATION ERROR (GE)

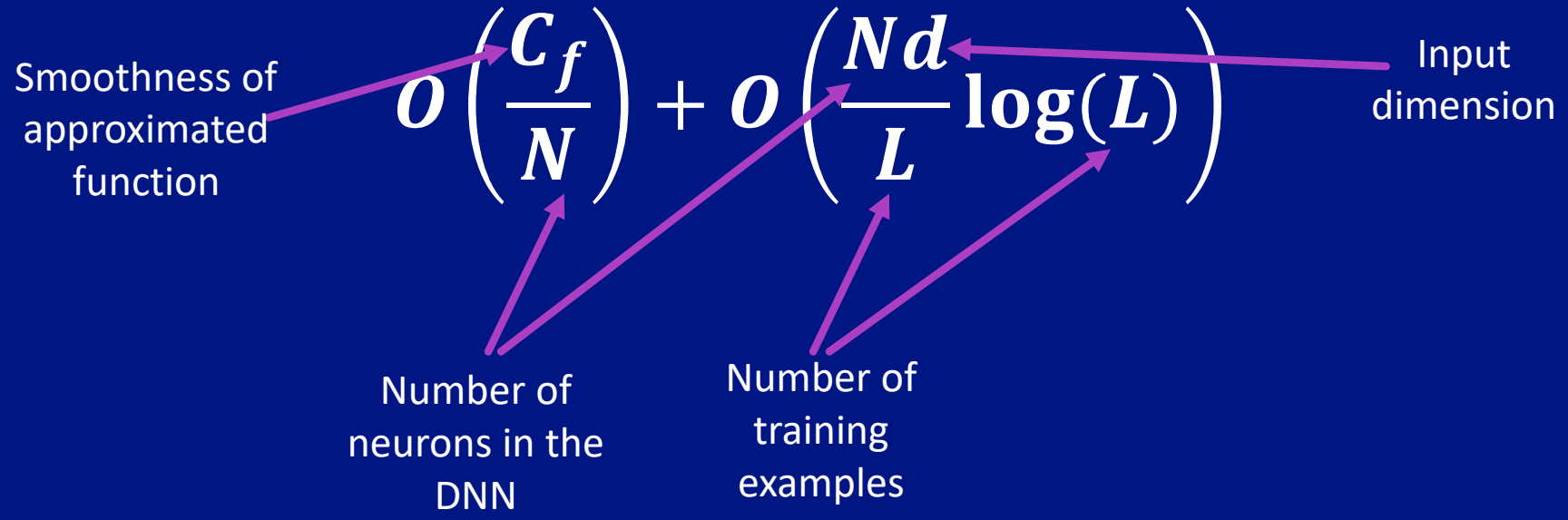
- In training, we reduce the classification error ℓ_{training} of the training data as the number of training examples L increases.
- However, we are interested to reduce the error ℓ_{test} of the (unknown) testing data as L increases.
- The difference between the two is the generalization error

$$\text{GE} = \ell_{\text{training}} - \ell_{\text{test}}$$

⇒ It is important to understand the GE of DNN

ESTIMATION ERROR

- The estimation error of a function f by a neural networks scales as [Barron 1994].



REGULARIZATION TECHNIQUES

- Weight decay – penalizing DNN weights [Krogh & Hertz, 1992].
- Dropout - randomly drop units (along with their connections) from the neural network during training [Hinton et al., 2012], [Baldi & Sadowski, 2013], Srivastava et al., 2014].
- DropConnect – dropout extension [Wan et al., 2013]
- Batch normalization [Ioffe & Szegedy, 2015].
- Stochastic gradient descent (SGD) [Hardt, Recht & Singer, 2016].
- Path-SGD [Neyshabur et al., 2015].
- And more [Rifai et al., 2011], [Salimans & Kingma, 2016], [Sun et al, 2016].

A SAMPLE OF GE BOUNDS

- Using the VC dimension it can be shown that

$$\mathrm{GE} \leq O\left(\sqrt{\text{DNN params} \cdot \frac{\log(L)}{L}}\right)$$

L is the number of training samples

[Shalev-Shwartz and Ben-David, 2014].

- The GE was bounded also by the DNN weights

$$\mathrm{GE} \leq \frac{1}{\sqrt{L}} 2^K \|w\|_2 \prod_i \left\|X^i\right\|_{2,2}$$

[Neyshabur et al., 2015].

A SAMPLE OF GE BOUNDS

- Using the VC dimension it can be shown that

$$\mathrm{GE} \leq O\left(\sqrt{\text{DNN params} \cdot \frac{\log(L)}{L}}\right)$$

[Shalev-Shwartz and Ben-David, 2014].

- The GE was bounded also by the DNN weights

$$\mathrm{GE} \leq \frac{1}{\sqrt{L}} 2^{K-1} \|w\|_2 \prod_i \|X^i\|_F$$

[Neyshabur et al., 2015].

- In [Golowich et al., 2018] an RC bound was provided that is independent of the network size

RETHINKING GENERALIZATION

- Networks with the same architecture may generalize well with structured data but overfit if the data is given with random labels [Zhang et al., 2017].
- This phenomena is affected by explicit regularization.
- This shows that taking into account only the network structure for bouding the generalization error is misleading
- We need to seek an alternative to the Rademacher Complexity and VC-dimension based bounds

COMPRESSION APPROACH

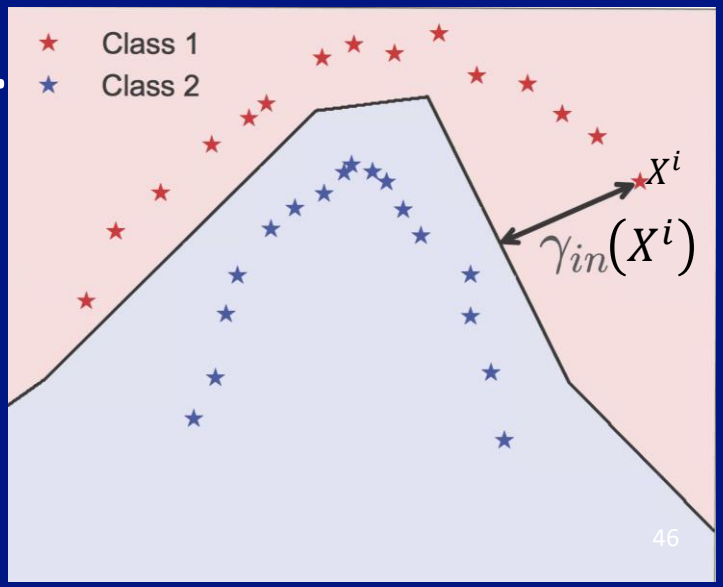
- Weight in neural networks are very redundant
- One may compress the network and still get approximately the same performance
- One may calculate the RC or VC dimension based bounds based on the number of neurons in the compressed network
- This leads to a significantly tighter GE bounds [Neyshabur et al., 2018].

OPTIMIZATION AND GENERALIZATION

- In [Hardt et al., 2016] it is shown that SGD serves as a regularizer in the training of neural networks.
- In [Brutzkus et al., 2018] it is proven formally that a two layers neural network trained with SGD (under some assumptions) converges to the global minimum and generalizes well.
- Training using softmax is shown to lead to large margin in linear networks [Soudry et al., 2018]

DNN INPUT MARGIN

- Theorem 6: If for every input margin $\gamma_{in}(X^i) > \gamma$ then $GE \leq \sqrt{N_{\gamma/2}(\Upsilon)}/\sqrt{L}$ [Sokolic, Giryes, Sapiro, Rodrigues, 2017]
- $N_{\gamma/2}(\Upsilon)$ is the covering number of the data Υ.
- $N_{\gamma/2}(\Upsilon)$ gets smaller as γ gets larger.
- Bound is independent of depth.
- Our theory relies on the robustness framework [Xu & Mannor, 2012].



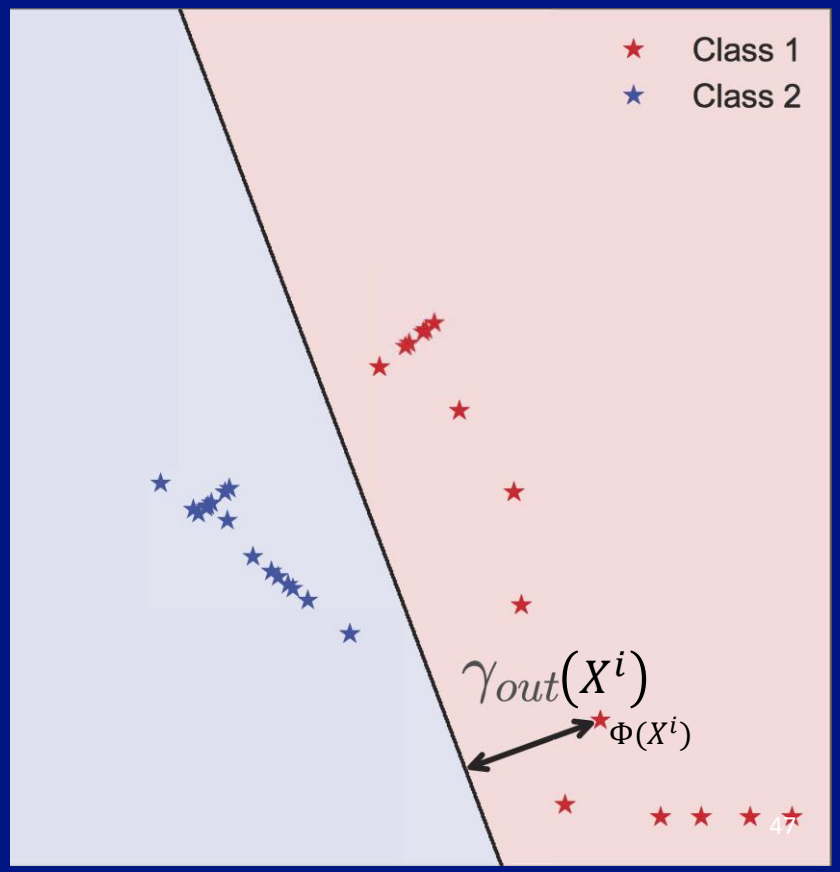
INPUT MARGIN BOUND

- Maximizing the input margin directly is hard
- Our strategy: relate the input margin to the output margin $\gamma_{out}(X^i)$ and other DNN properties
- Theorem 7:

$$\gamma_{in}(X^i) \geq \frac{\gamma_{out}(X^i)}{\sup_{X \in Y} \left\| \frac{X}{\|X\|_2} J(X) \right\|_2}$$

$$\geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_2}$$

$$\geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_F}$$



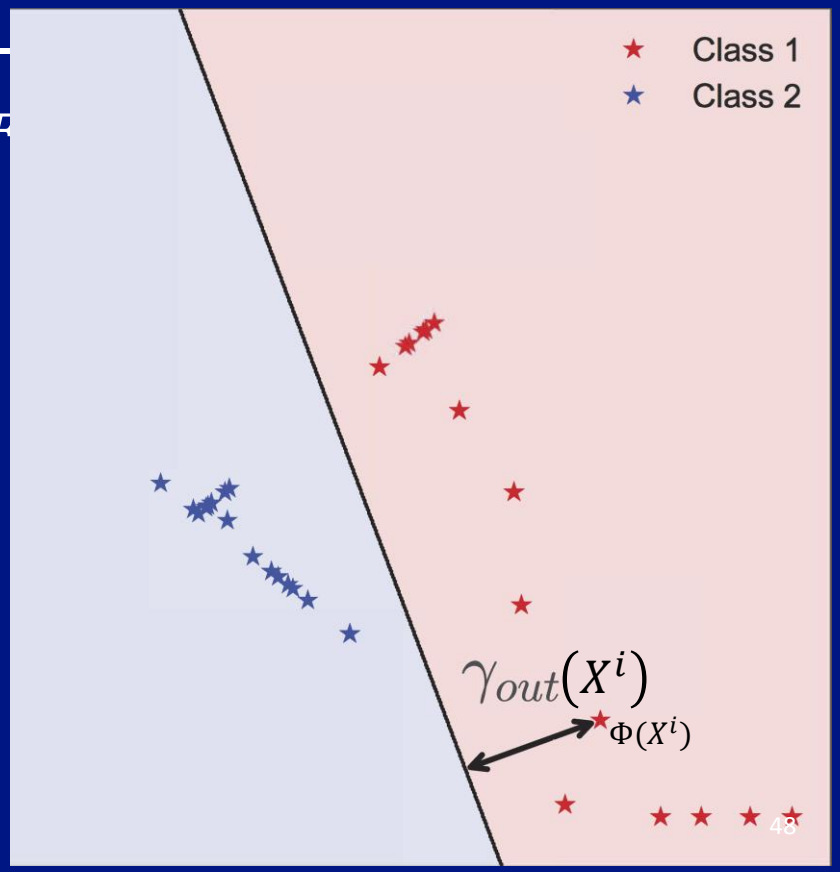
[Sokolic, Giryes, Sapiro, Rodrigues, 2017]

OUTPUT MARGIN

- Theorem 7: $\gamma_{in}(X^i) \geq \dfrac{\gamma_{out}(X^i)}{\sup_{V \in \Upsilon} \left\| \frac{X}{\|X\|_2} J(X) \right\|_2}$

$$\geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_2} \geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_F}$$

- Output margin is easier to maximize – SVM problem
- Maximized by many cost functions, e.g., hinge loss.

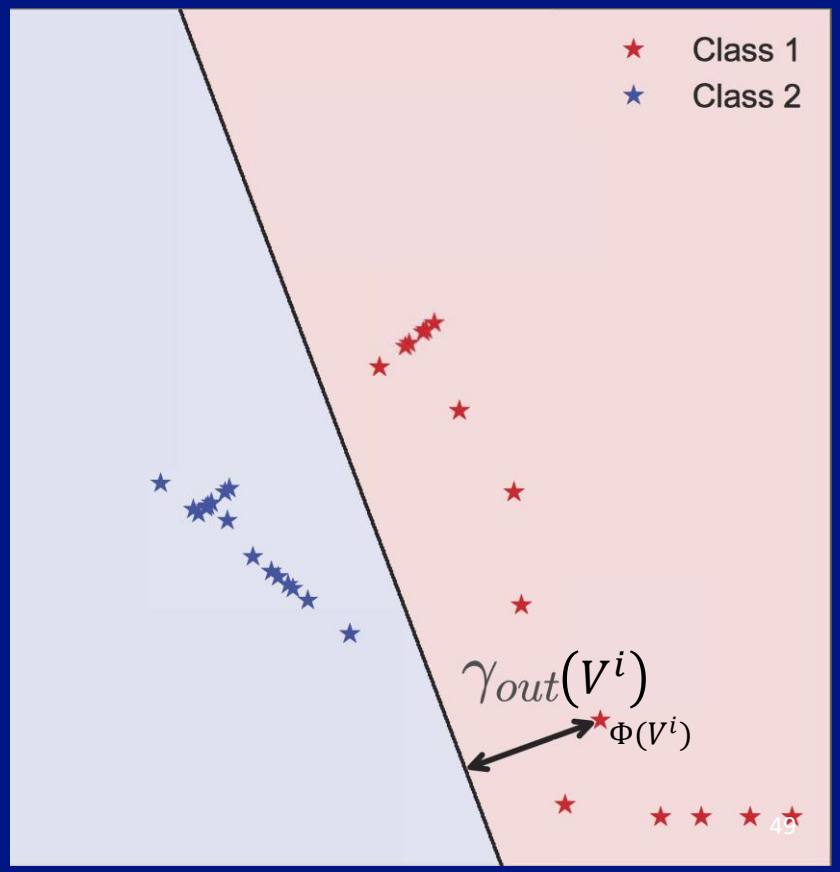


GE AND WEIGHT DECAY

- Theorem 7: $\gamma_{in}(X^i) \geq \frac{\gamma_{out}(X^i)}{\sup_{V \in \Upsilon} \left\| \frac{X}{\|X\|_2} J(X) \right\|_2} \geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_2}$

  $\geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_F}$

- Bounding the weights increases the input margin
- Weight decay regularization decreases the GE
- Related to regularization used by [Haeffele & Vidal, 2015]

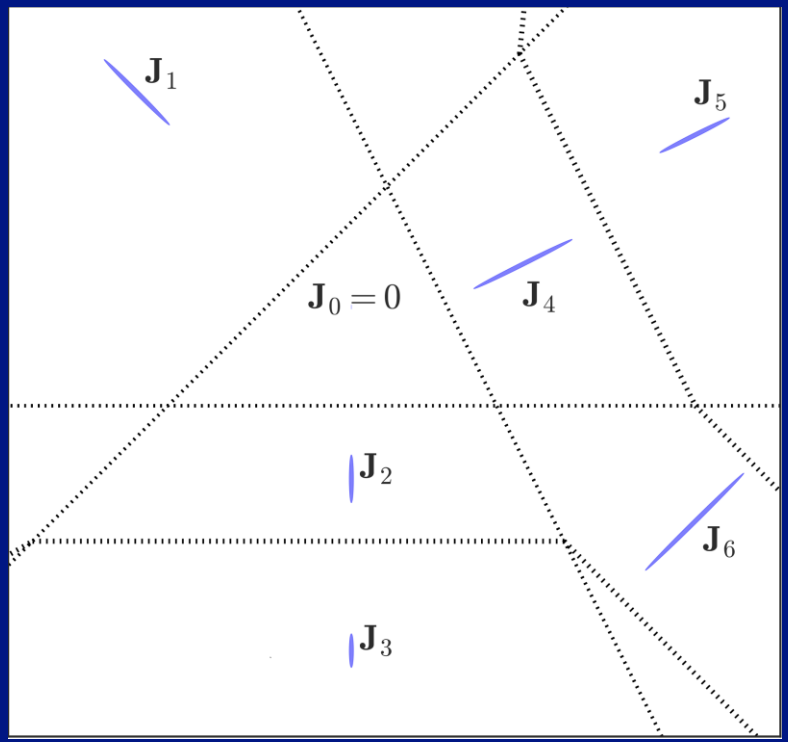


JACOBIAN BASED REGULARIZATION

- Theorem 7: $\gamma_{in}(X^i) \geq \frac{\gamma_{out}(X^i)}{\sup_{V \in Y} \left\| \frac{X}{\|X\|_2} J(X) \right\|_2} \geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_2}$

$$\geq \frac{\gamma_{out}(X^i)}{\prod_{1 \leq i \leq K} \|W^i\|_F}$$

- $J(X)$ is the Jacobian of the DNN at point X.
- $J(\cdot)$ is piecewise constant.
- Using the Jacobian of the DNN leads to a better bound.
- ⇨ New regularization technique.



RESULTS

- Better performance with less training samples

MNIST Dataset

		256 samples			512 samples			1024 samples		
loss	# layers	no reg.	WD	LM	no reg.	WD	LM	no reg.	WD	LM
hinge	2	88.37	89.88	**93.83**	93.99	94.62	95.49	95.79	96.57	97.45
hinge	3	87.22	89.31	93.22	93.41	93.97	**95.76**	95.46	96.45	**97.60**
CCE	2	88.45	88.45	92.77	92.29	93.14	95.25	95.38	95.79	96.89
CCE	3	89.05	89.05	93.10	91.81	93.02	95.32	95.11	95.86	97.14

- CCE: the categorical cross entropy.
- WD: weight decay regularization.
- LM: Jacobian based regularization for large margin.
- Note that hinge loss generalizes better than CCE and that LM is better than WD as predicted by our theory.

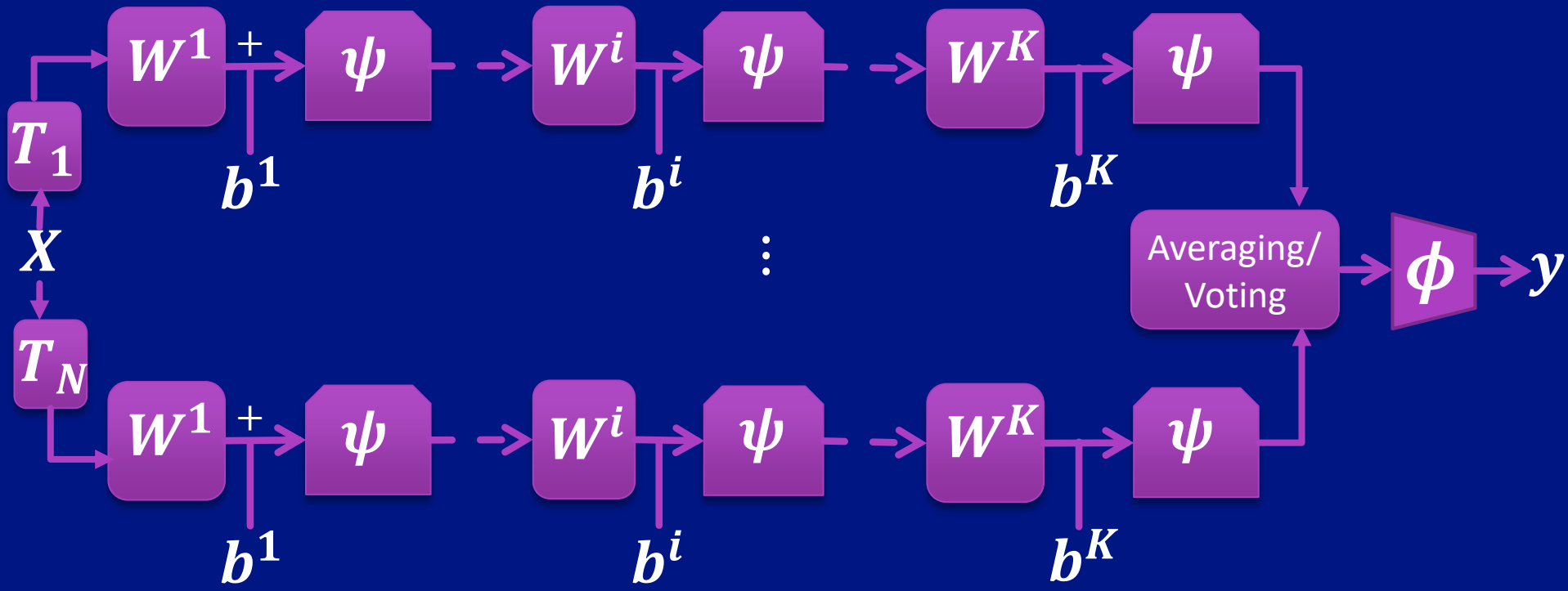
[Sokolic, Giryes, Sapiro, Rodrigues, 2017]

INVARIANCE

- Our theory extends also to study of the relation between invariance in the data and invariance in the network
- Invariance may improve generalization by a factor of $\sqrt{T}$, where T is the number of transformations
- We have proposed also a new strategy to enforce invariance in the network [Sokolic, Giryes, Sapiro, Rodrigues, 2017]

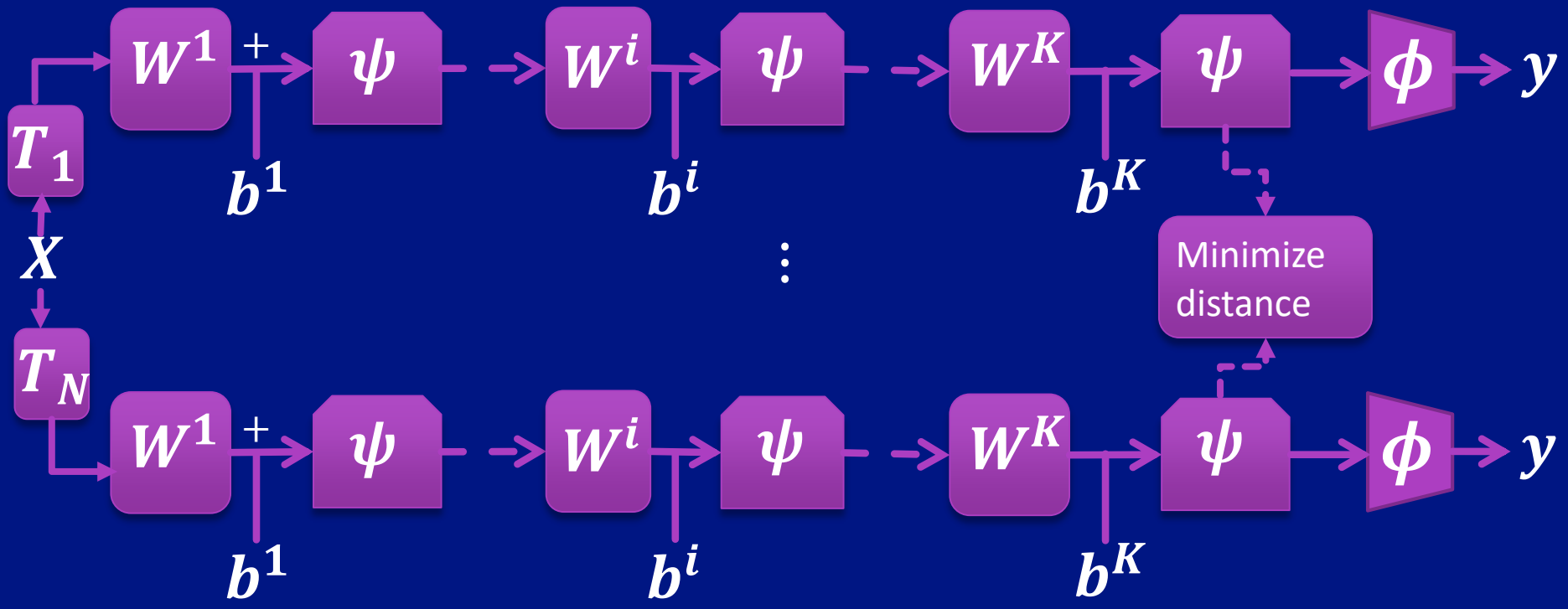
INVARIANCE SLICE

- Use transformations $T_1, \ldots, T_N$ to transform the input [Dieleman et al., 2016]
- Average the features before the soft-max layer



INVARIANCE BY REGULARIZATION

- Use transformations $T_1, \ldots, T_N$ to transform the input [Sokolic et al., 2017]
- Force features to be similar



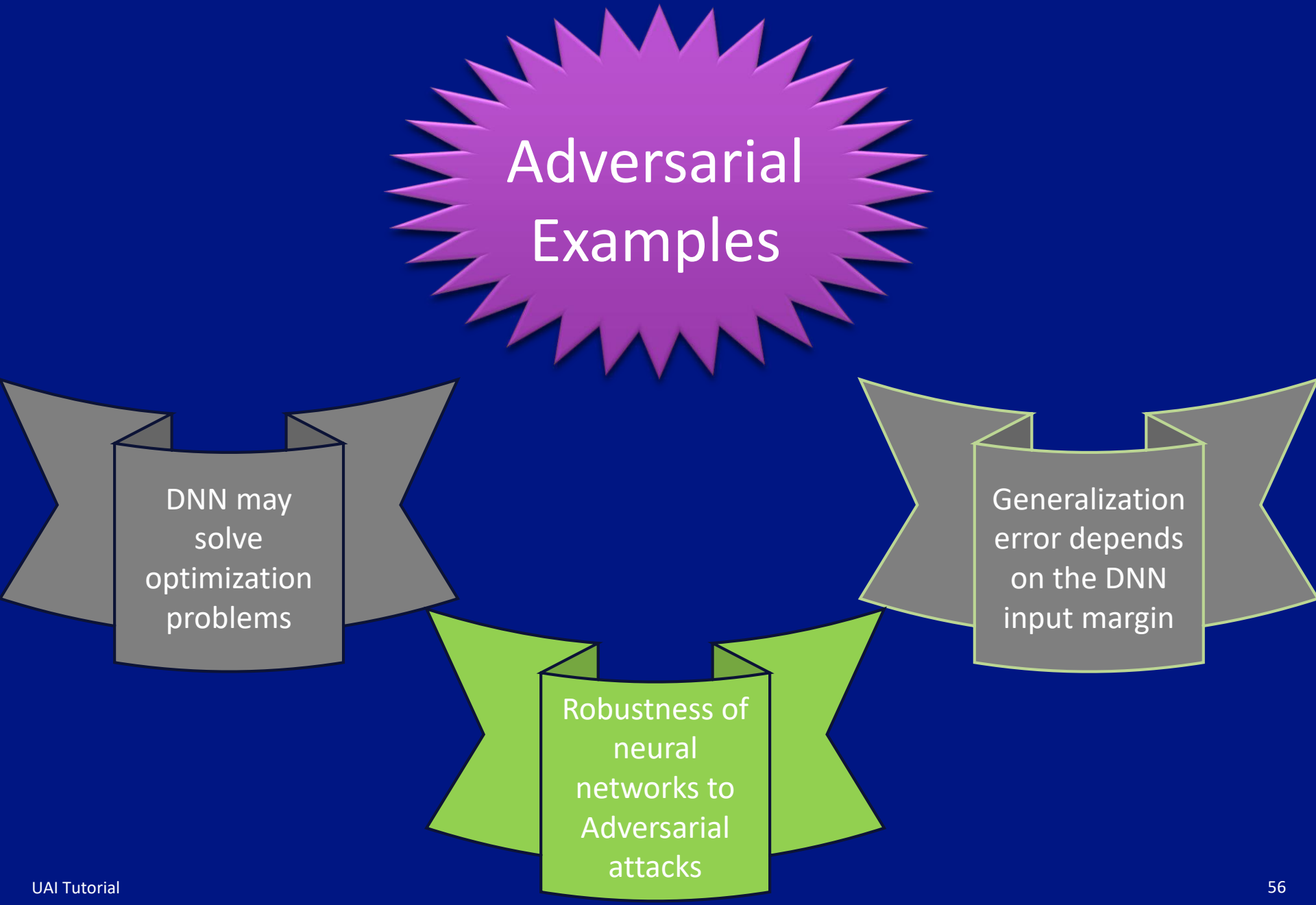
INVARIANCE

- Designing invariant DNN reduce the GE

Table 1: Classification accuracy [%] on CIFAR-10.

	number of training samples				
	2500	5000	10000	20000	50000
No reg.	68.71	76.74	85.17	87.15	93.65
Inv. Reg.	69.32	79.08	86.69	88.14	94.50
No reg. + avg.	70.59	78.40	86.05	88.13	94.26
Inv. Reg. + avg.	70.71	79.65	86.96	88.98	94.78

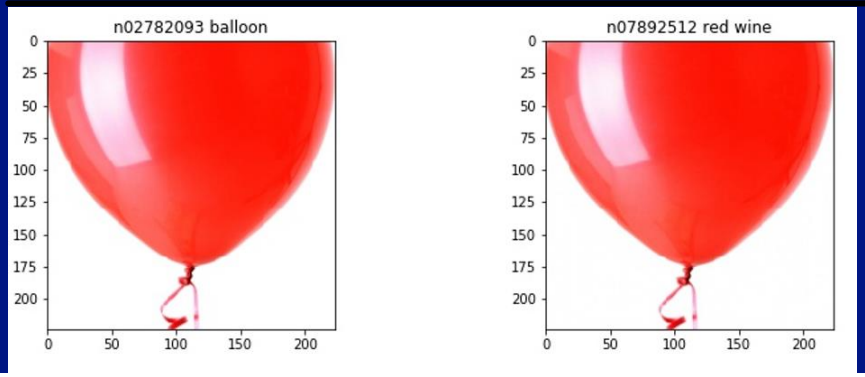
[Sokolić, Giryes, Sapiro & Rodrigues, 2017]

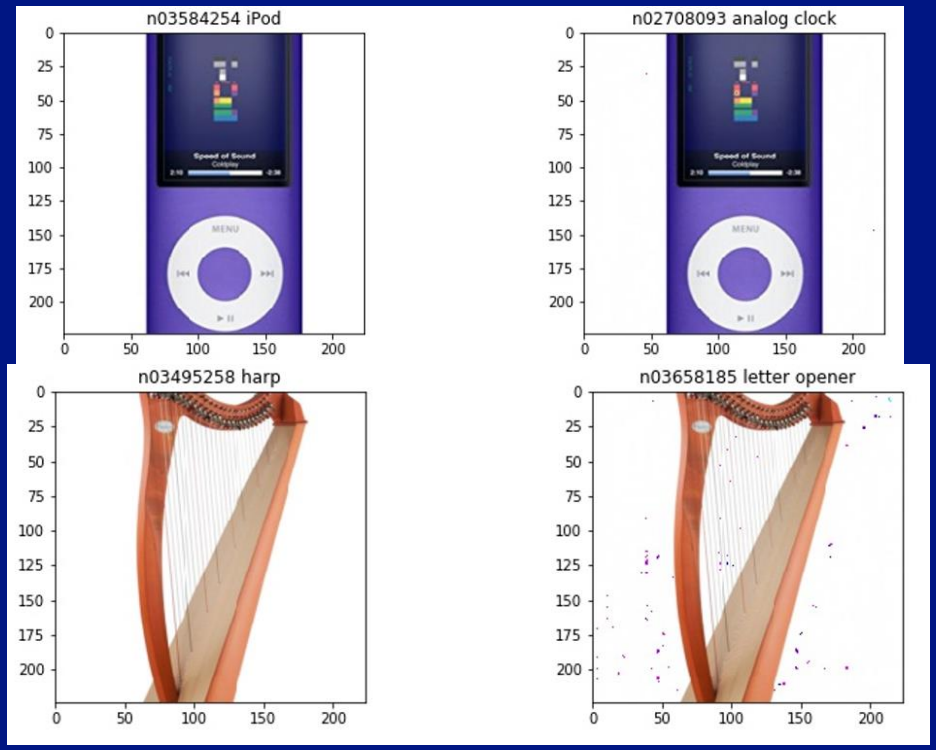

Adversarial Examples
DNN may solve optimization problems
Robustness of neural networks to Adversarial attacks
Generalization error depends on the DNN input margin

ADVERSARIAL ATTACKS

- Deep neural networks can easily be fooled by small perturbations in the input, commonly referred to as *Adversarial Attacks*.
- The problem of “no common sense” – a network can perform its task well (e.g. image classification), but can easily be fooled in a way a human cannot.
- *Adversarial Attacks* are deliberate input perturbations – random noise is not as likely to fool the network.

ADVERSARIAL EXAMPLES





**These examples were generated using the DeepFool attack on ResNet-34, ImageNet classification.*

ADVERSARIAL ATTACKS

- Adversarial examples are highly transferable – an attack that successfully fooled one network is very likely to fool another network as well.
- Very little knowledge of the network's architecture is necessary to attack it, i.e. grey-box attacks are very efficient as well.
- Several explanations to this phenomenon have been suggested:
  - Adversarial examples finely tile the space like the rational numbers amongst the reals, they are common but occur only at very precise locations ("pockets").
  - Positively curved decision boundaries are more susceptible to universal adversarial perturbations.

DEFENSE AND ATTACK METHODS

- Several attack and defense methods have been proposed to counter this problem.
- Defense methods aim at either increasing the robustness to attacks, or detecting that an attack has been performed.
- Attack methods:
  - FGSM (Fast Gradient Sign Method)
  - JSMA (Jacobian-based Saliency Map Approach)
  - DeepFool
  - Carlini & Wagner

DEFENSE AND ATTACK METHODS

- Defense methods:
  - Adversarial training
  - Network Distillation
  - Input Gradient regularization
  - Cross-Lipschitz regularization
  - **Jacobian regularization**

JACOBIAN REGULARIZATION

- We proposed a novel approach to enhance a network's robustness to adversarial attacks – Jacobian regularization.
- A network's Jacobian matrix for $z^{(L)}$ - the output of the network's last fully connected layer before softmax (i.e. the logits) is

$$J(x_i) \triangleq J^{(L)}(x_i) = \begin{bmatrix} \frac{\partial z_1^{(L)}(x_i)}{\partial x_{(1)}} & \cdots & \frac{\partial z_1^{(L)}(x_i)}{\partial x_{(D)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_K^{(L)}(x_i)}{\partial x_{(1)}} & \cdots & \frac{\partial z_K^{(L)}(x_i)}{\partial x_{(D)}} \end{bmatrix} \in \mathbb{R}^{K \times D}$$

where D – input dimension, K – output dimension:

JACOBIAN REGULARIZATION

- We regularize the Frobenius norm of the network's Jacobian:

$$\|J(x_i)\|_F^2 = \sum_{d=1}^{D}\sum_{k=1}^{K}\left(\frac{\partial}{\partial x_d} z_k^{(L)}(x_i)\right)^2 = \sum_{k=1}^{K}\left\|\nabla_x z_k^{(L)}(x_i)\right\|_2^2$$

- We perform post-processing training, i.e. our regularization is applied to already-trained networks:
  - Reducing the computational overhead (Jacobian computation requires an additional back-propagation step).
  - Allowing the usage of pre-existing networks.

[Sokolić, Giryes, Sapiro & Rodrigues, 2017]

WHY DOES IT WORK?

- Adversarial examples are essentially cases in which similar network inputs result in very different network outputs. Regularizing the Jacobian constraints this behavior: smaller Jacobian Frobenius norm → smoother classification function.
- Let $[x, x_{pert}]$ be the D-dimensional line in the input space connecting x and x_{pert}. According to the mean value theorem there exists some $x' \in [x, x_{pert}]$ such that:

$$\frac{\left\| z^{(L)}(x_{pert}) - z^{(L)}(x) \right\|_2^2}{\|x_{pert} - x\|_2^2} \leq \sum_{k=1}^{K} \left\| \nabla_x z_k^{(L)}(x') \right\|_2^2 = \|J(x')\|_F^2$$

WHY DOES IT WORK?

- Empirical motivation – the average Jacobian Frobenius norm of perturbed images is larger:

Defense method	$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{J}(\mathbf{x}_i)\|_F$	$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{J}(\mathbf{x}_{i_{pert}})\|_F$
No defense	0.14	0.1877
Adversarial training	0.141	0.143
Jacobian regularization	0.0315	0.055
Jacobian reg. & Adversarial training	0.0301	0.0545

[Jakubovitz & Giryes, 2018]

WHY DOES IT WORK?

- Generally, an attack method seeks for the closest decision boundary to be crossed to cause a misclassification, such that the perturbation of the input is minimal.
- The distance d^* is the first order approximation of the distance between an input x and an input classified to the closest decision boundary. The relation between d^* and the network's Jacobian matrix (k^* is the original class of input x, $k = 1, \dots, K$ is the class index):

$$d^* \geq \frac{1}{\sqrt{2}\|\mathrm{J}^{(L)}(x)\|_F} \min_{k \neq k^*} \left| z_{k^*}^{(L)}(x) - z_k^{(L)}(x) \right|$$

- Encouraging a smaller Frobenius norm of the network's Jacobian means encouraging a larger minimal distance between the original input and a perturbed input that would cause a misclassification.

[Jakubovitz & Giryes, 2018]

WHY DOES IT WORK?

- It has been shown that positively curved decision boundaries create an enhanced vulnerability to adversarial examples:

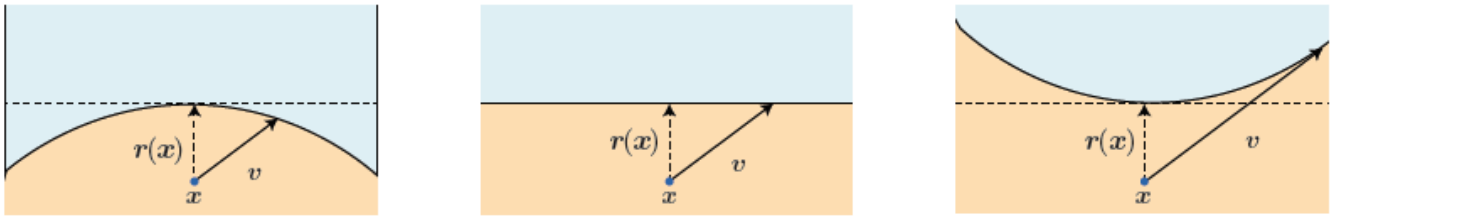


Fig. 3: Link between robustness and curvature of the decision boundary. When the decision boundary is *positively* curved (left), small universal perturbations are more likely to fool the classifier.

**Illustration taken from "Analysis of universal adversarial perturbations", Moosavi-Dezfooli et al.*

- Jacobian regularization encourages the network's learned decision boundaries to be less positively curved.

WHY DOES IT WORK?

- The decision boundary separating the classes k_1 and k_2 is the hyper-surface: $F_{k_1,k_2}(x) = z_{k_1}^{(L)}(x) - z_{k_2}^{(L)}(x) = 0$.
- Using the approximation $H_k(x) = \frac{\partial^2 z_k^{(L)}(x)}{\partial x^2} \approx J_k(x)^T J_k(x)$ (outer product of gradients, $J_k(x)$ is the k^{th} row in the matrix $J(x)$), we get that the curvature of the decision boundary $F_{k_1,k_2}(x) = z_{k_1}^{(L)}(x) - z_{k_2}^{(L)}(x) = 0$ at the input point x can be upper bounded by:

$$x^T\left(H_{k_1} - H_{k_2}\right)x \leq \sum_{k=1}^{K} (J_k(x)x)^2 \leq \|J(x)\|_F^2 \|x\|_2^2$$

- For this reason, Jacobian regularization promotes a less positive curvature of the decision boundaries in the environment of the input samples.

EXPERIMENTAL RESULTS

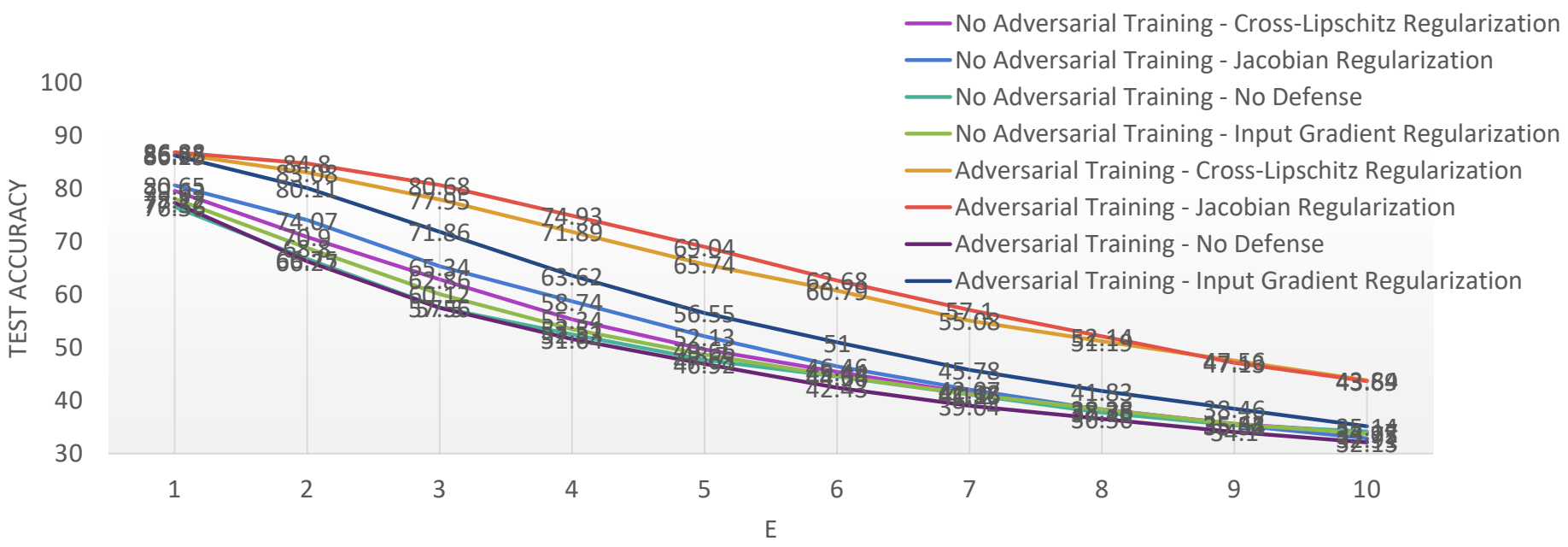
- We examined the performance of our method under different attack methods, and compared them to 3 other prominent defense methods – Input Gradient regularization, Cross-Lipschitz regularization and adversarial training.
- Results under the DeepFool attack on **CIFAR-10** ($\hat{\rho}_{adv}$ is the average proportion between the norm of the minimal perturbation necessary to fool the network and the norm of the corresponding original input):

Defense method	Test accuracy	$\hat{\rho}_{adv}$
No defense	88.79%	1.21×10^{-2}
Adversarial Training	88.88%	1.23×10^{-2}
Input Gradient regularization	88.56%	1.43×10^{-2}
Input Gradient reg. & Adversarial Training	88.49%	2.17×10^{-2}
Cross-Lipschitz regularization	88.91%	2.08×10^{-2}
Cross-Lipschitz reg. & Adversarial Training	88.49%	4.04×10^{-2}
Jacobian regularization	89.16%	$\mathbf{3.42 \times 10^{-2}}$
Jacobian reg. & Adversarial Training	88.49%	$\mathbf{6.03 \times 10^{-2}}$

[Jakubovitz & Giryes, 2018]

EXPERIMENTAL RESULTS - FGSM

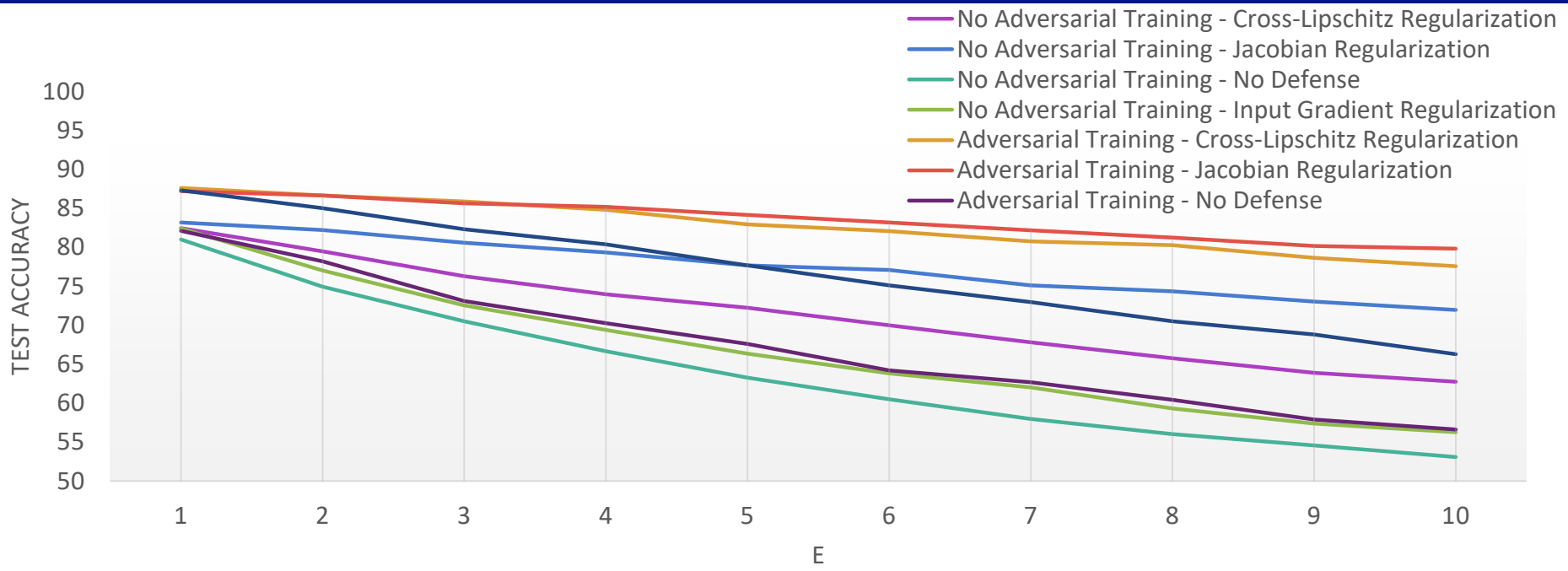
- Results under FGSM attack on **CIFAR-10** (test accuracy for a test set of adversarial examples, ϵ - attack magnitude):



[Jakubovitz & Giryes, 2018]

EXPERIMENTAL RESULTS - JSMA

- Results under JSMA attack on **CIFAR-10** (test accuracy for a test set of adversarial examples, ϵ - attack magnitude):



[Jakubovitz & Giryes, 2018]

Optimization by DNN

DNN may solve optimization problems

Robustness of neural networks to Adversarial attacks

Generalization error depends on the DNN input margin

INVERSE PROBLEMS

- We are given $\boldsymbol{X = AZ + E}$

Given set of measurements

linear operator

Unknown signal

noise

- Standard technique for recovery

$$\min_{Z} \|X - AZ\|_2 \qquad \text{s.t.} \qquad Z \in \Upsilon$$

Z resides in a low dimensional set Υ

- Unconstrained form

$$\min_{Z} \|X - AZ\|_2^2 + \lambda f(Z)$$

Regularization parameter

f is a penalty function

ℓ_1 MINIMIZATION CASE

- Unconstrained form

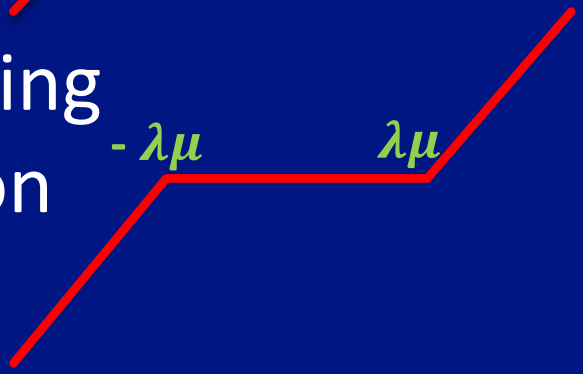
$$\min_{Z} \|X - AZ\|_2^2 + \lambda \|Z\|_1$$

- Can be solved by proximal gradient, e.g., iterative shrinkage and thresholding technique (ISTA)

$$Z^{t+1} = \psi_{\lambda\mu}\left(Z^t + \mu A^T(X - AZ^t)\right)$$

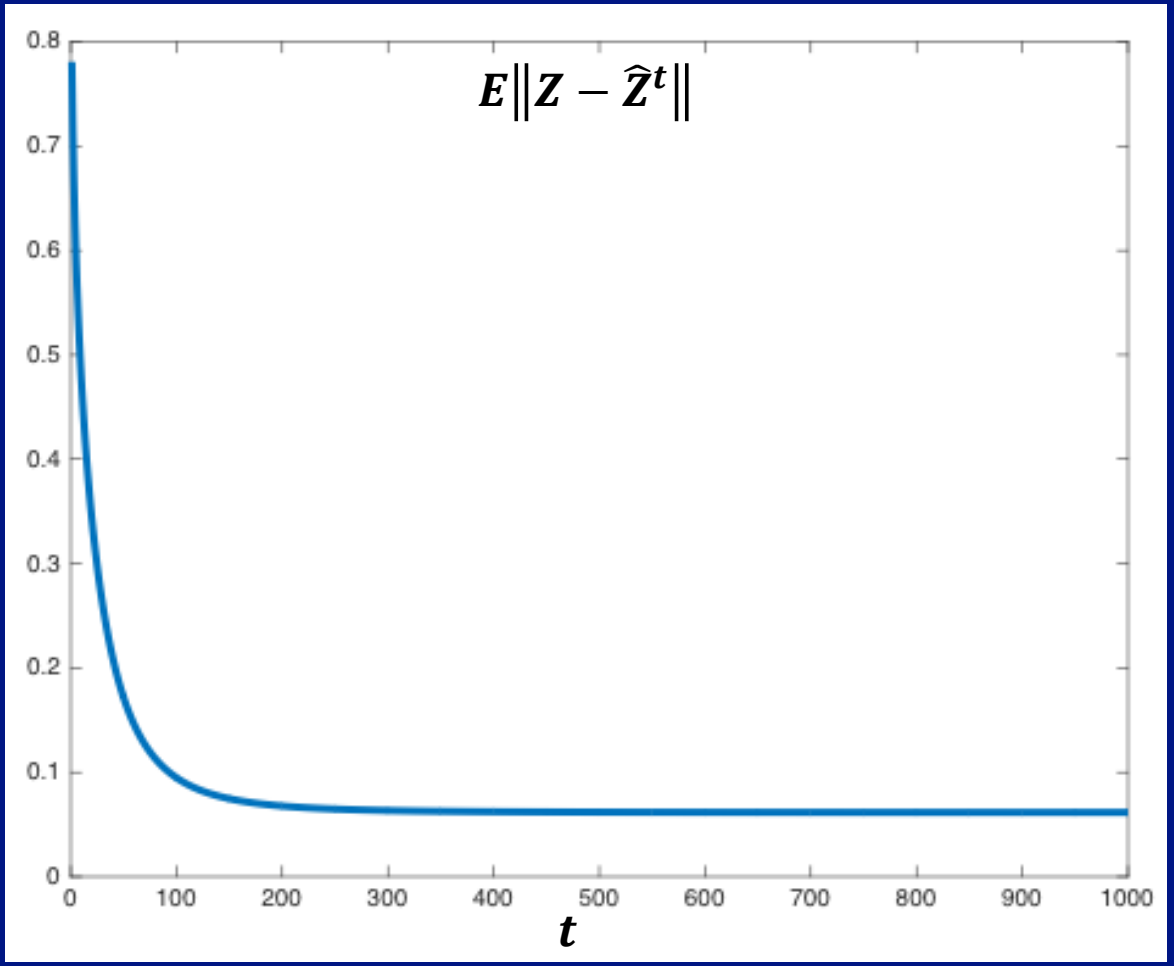
Soft thresholding operation

μ is the step size



ISTA CONVERGENCE

- Reconstruction mean squared error (MSE) as a function of the number of iterations



LISTA

- ISTA

$$\boldsymbol{Z^{t+1} = \psi_{\lambda\mu}\left(Z^t + \mu A^T(X - AZ^t)\right)}$$

- Rewriting ISTA:

$$\boldsymbol{Z^{t+1} = \psi_{\lambda\mu}\left(\left(I - \mu A^T A\right)Z^t + \mu A^T X\right)}$$

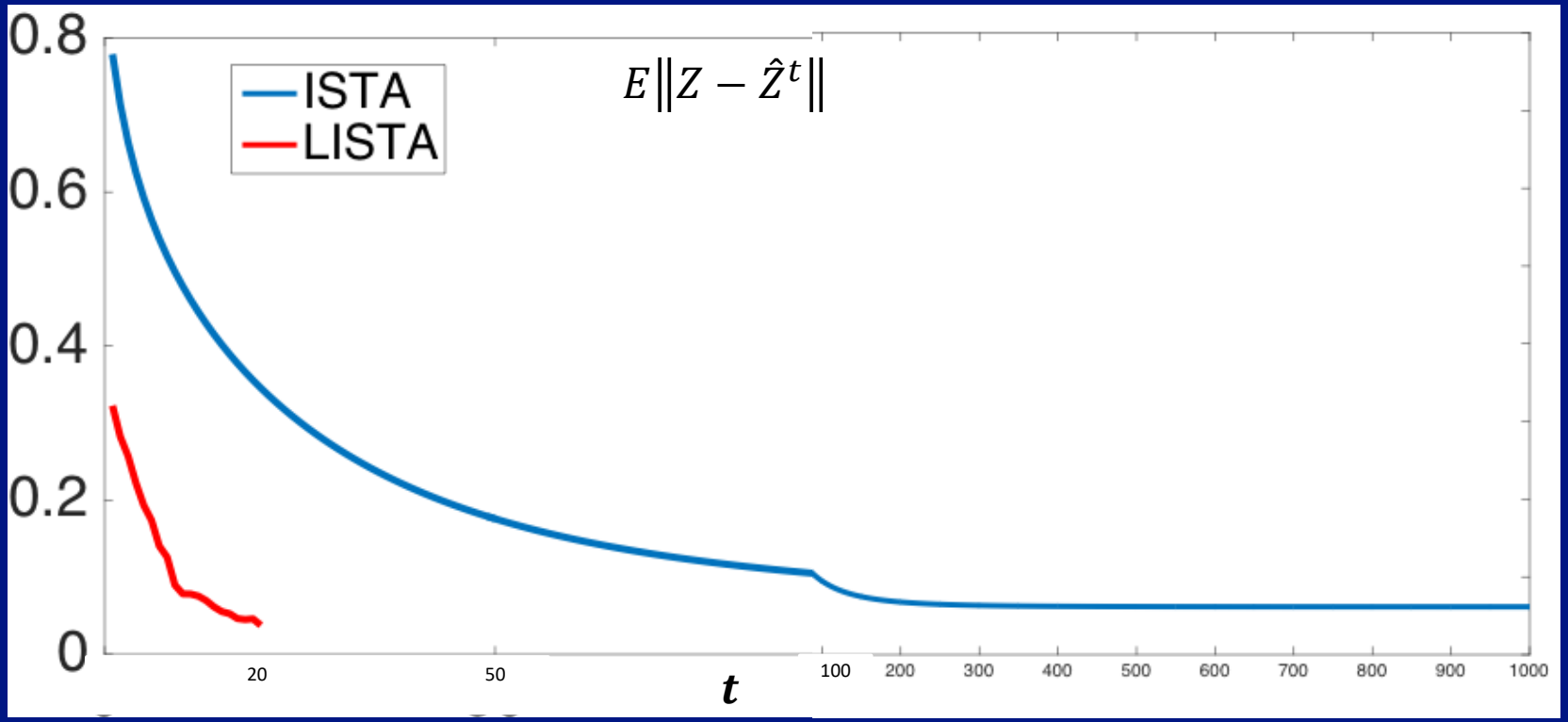
- Learned ISTA (LISTA):

$$\boldsymbol{Z^{t+1} = \psi_{\lambda}(WZ^t + SX)}$$

Learned operators

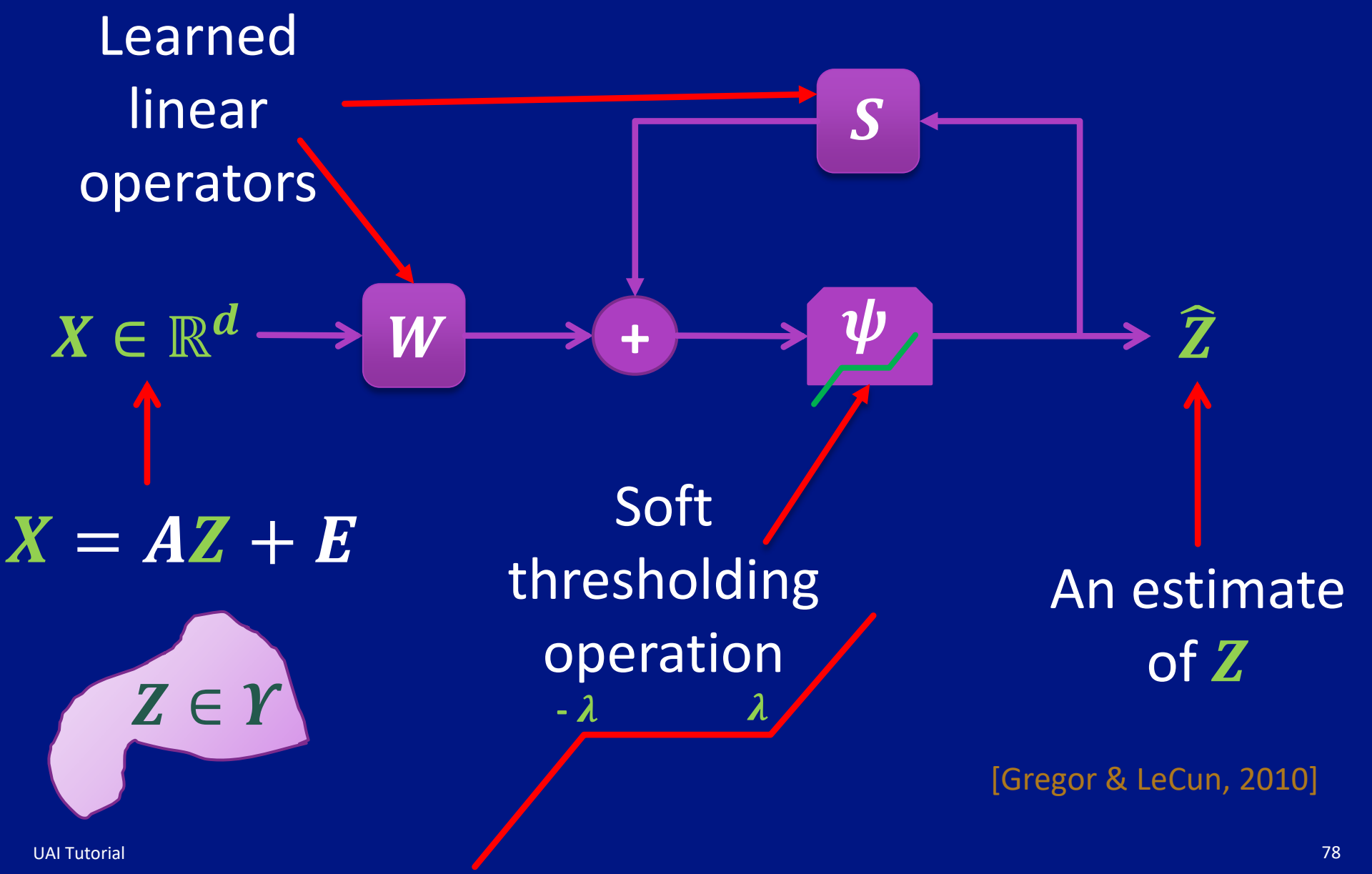
LISTA CONVERGENCE

- Replacing $I - \mu A^T A$ and μA^T in ISTA with the learned W and S improves convergence [Gregor & LeCun, 2010]



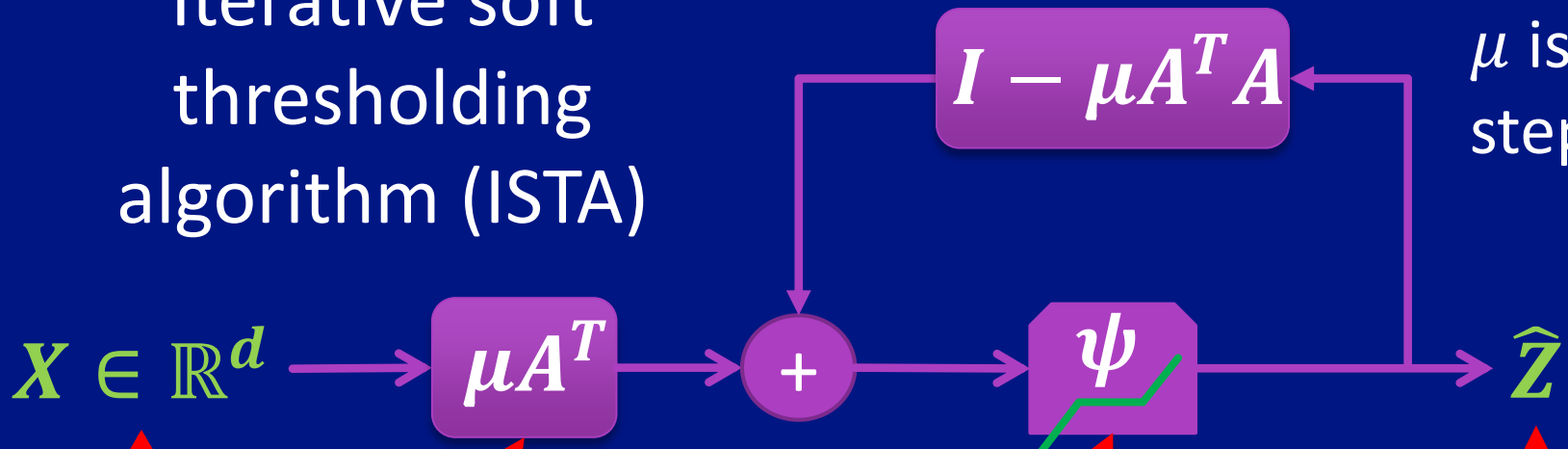
- Extensions to other models [Sprechmann, Bronstein & Sapiro, 2015], [Remez, Litani & Bronstein, 2015], [Tompson, Schlachter, Sprechmann & Perlin, 2016].

LISTA AS A NEURAL NETWORK



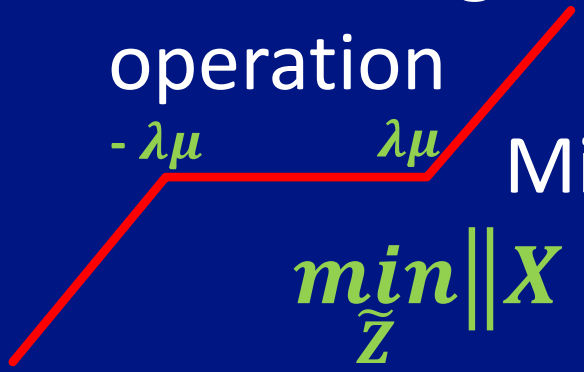
ISTA

Iterative soft thresholding algorithm (ISTA)



Step size μ obeys $\frac{1}{\mu} \geq \|A\|$

Soft thresholding operation



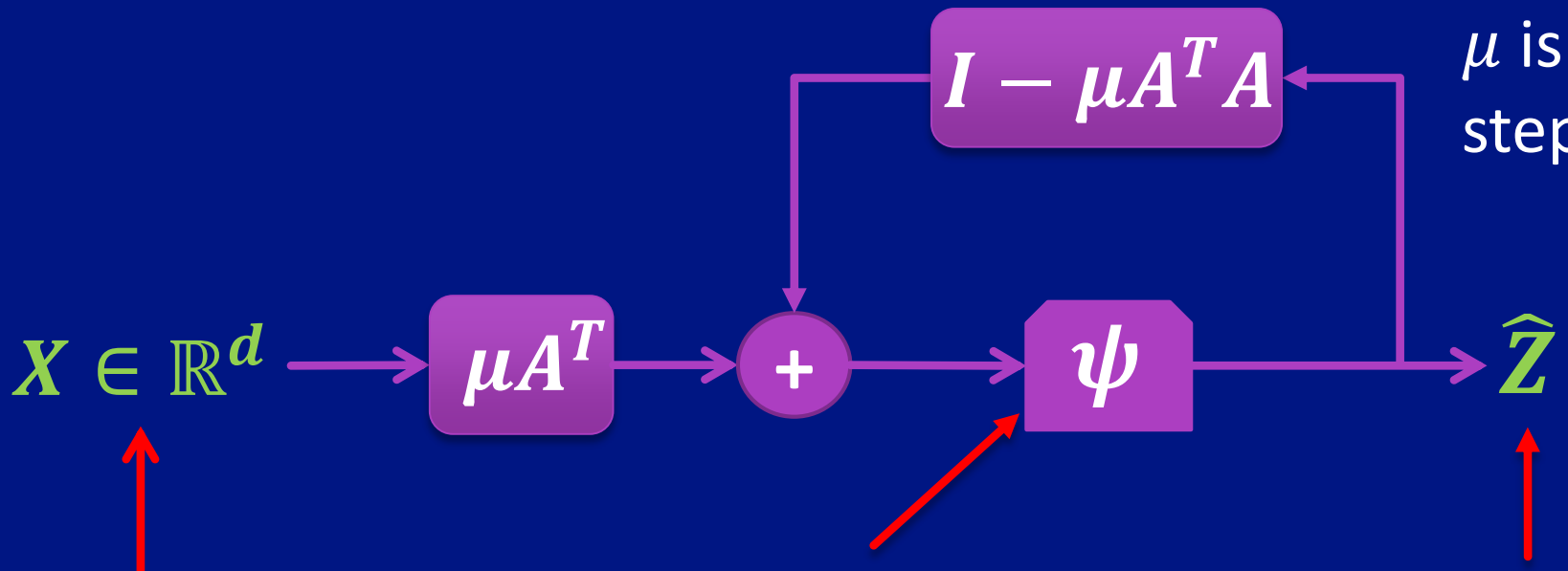
Minimizer of

$$\min_{\tilde{Z}} \|X - A\tilde{Z}\| + \lambda\|\tilde{Z}\|_1$$

$$X = AZ + E$$

[Daubechies, Defrise & Mol, 2004], [Beck & Teboulle, 2009]

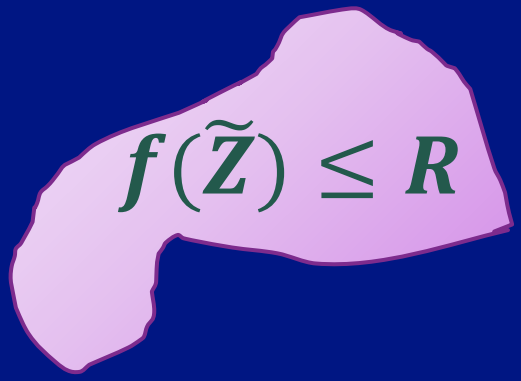
PROJECTED GRADIENT DESCENT (PGD)



$X = AZ + E$

$f(Z) \leq R$

ψ projects onto the set $\mathcal{Y}$



Estimate of Z.
Aim at solving

$$\min_{\tilde{Z}} \|X - A\tilde{Z}\| \quad s.t.\ f(\tilde{Z}) \leq R$$

THEORY FOR PGD

- Theorem 8: Let $Z \in \mathbb{R}^d$, $f: \mathbb{R}^d \to \mathbb{R}$ a proper function, $f(Z) \leq R$, $C_f(Z)$ the tangent cone of f at point Z, $A \in \mathbb{R}^{m \times d}$ a random Gaussian matrix and $X = AZ + E$. Then the estimate of PGD at iteration t, $\hat{Z}^t$, obeys

$$\left\| \hat{Z}^t - Z \right\| \leq \left(\kappa_f \rho \right)^t \|Z\|,$$

where $\rho = \sup_{U,V \in C_f(Z) \cap \mathcal{B}^d} U^T \left(I - \mu A^T A \right) V$

and $\kappa_f = 1$ if f is convex and $\kappa_f = 2$ otherwise. [Oymak, Recht & Soltanolkotabi, 2016].

PGD CONVERGENCE RATE

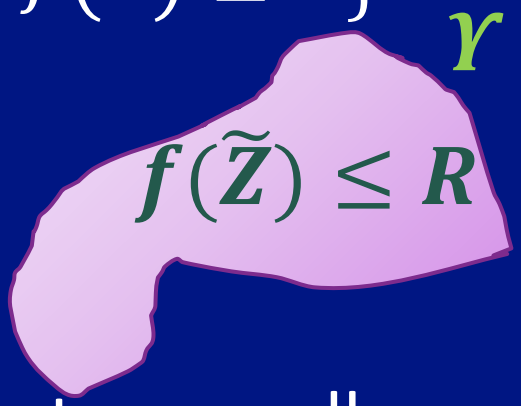
- $\rho = \sup_{U,V \in C_f(Z) \cap \mathcal{B}^d} U^T\left(I - \mu A^T A\right)V$ is the convergence rate of PGD.
- Let ω be the Gaussian mean width of $C_f(Z) \cap \mathcal{B}^d$.
- If $\mu = \frac{1}{\left(\sqrt{m}+\sqrt{d}\right)^2} \simeq \frac{1}{d}$ then $\rho = 1 - O\left(\frac{\sqrt{m}-\omega}{m+d}\right)$.
- If $\mu = \frac{1}{m}$ then $\rho = O\left(\frac{\omega}{\sqrt{m}}\right)$.
- For the k-sparse model $\omega^2 = O\left(k\log(\mathrm{d})\right)$
- For GMM with k Gaussians $\omega^2 = O(k)$.
- How may we cause ω to become smaller for having a better convergence rate?

INACCURATE PROJECTION

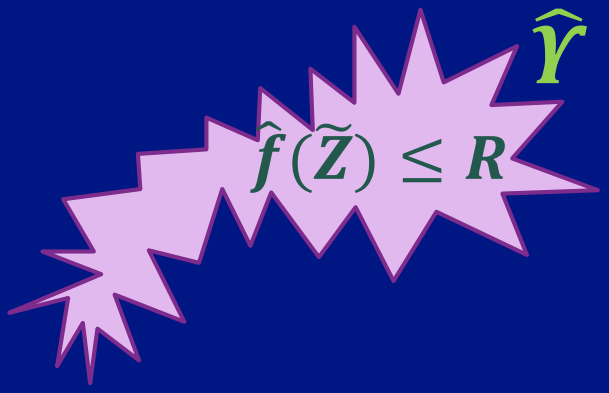
- PGD iterations projects onto $\Upsilon = \{\tilde{Z}: f(\tilde{Z}) \leq R\}$.
- Smaller $\Upsilon \Rightarrow$ Smaller ω.

$\Rightarrow$Faster convergence as

$\rho = 1 - O\left(\frac{\sqrt{m}-\omega}{m+d}\right)$ or $O\left(\frac{\omega}{\sqrt{m}}\right)$

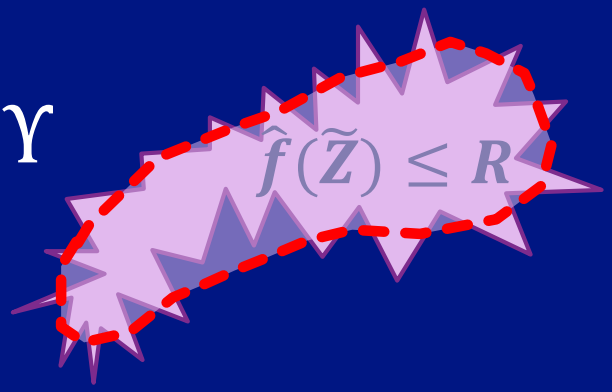


- Let us assume that our signal belongs to a smaller set $\hat{\Upsilon} = \{\tilde{Z}: \hat{f}(\tilde{Z}) \leq R\}$ with $\hat{\omega} \ll \omega$.
- Ideally, we would like to project onto $\hat{\Upsilon}$ instead of Υ.
- This will lead to faster convergence.
- What if such a projection is not feasible?



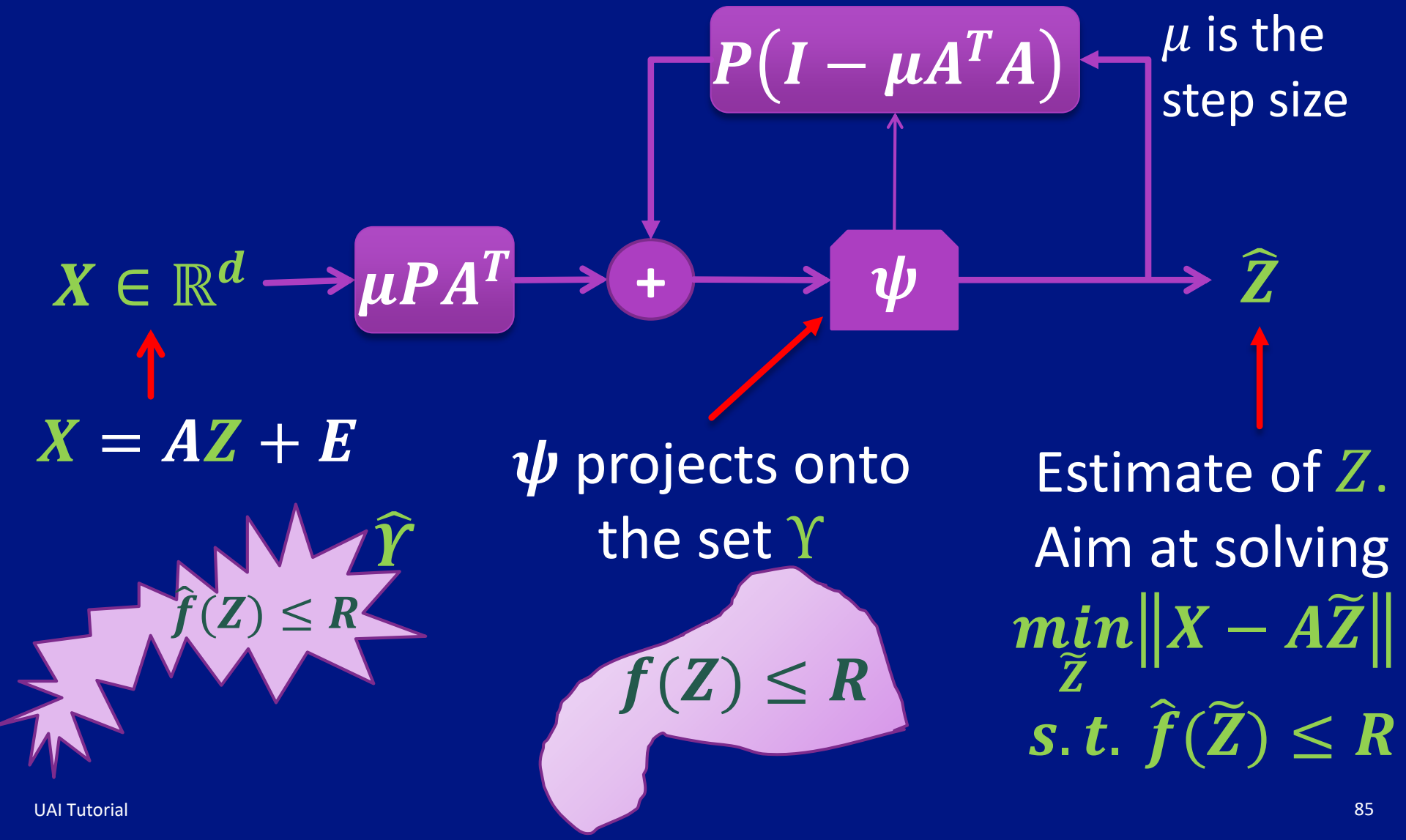
INACCURATE PROJECTION

- We will estimate the projection onto $\widehat{\Upsilon}$ by
  - A linear projection P
  - Followed by a projection onto Υ
- Assumptions:
  - $\|\wp_{\Upsilon}(PZ) - Z\| \leq \epsilon$



Projection of the target vector Z onto P and then onto Υ

INACCURATE PGD (IPGD)



THEORY FOR IPGD

- Theorem 9: Let $Z \in \mathbb{R}^d$, $f: \mathbb{R}^d \to \mathbb{R}$ a proper convex* function, $f(Z) \leq R$, $\hat{C}_f(Z)$ the tangent cone of f at point Z, $A \in \mathbb{R}^{d \times m}$ a random Gaussian matrix and $X = AZ + E$. Then the estimate of IPGD at iteration t, $\hat{Z}^t$, obeys

$$\left\| \hat{Z}^t - Z \right\| \leq \left((\rho_P)^t + \frac{1 - (\rho_P)^t}{1 - \rho_P} \tilde{\epsilon} \right) \|Z\|,$$

where $\rho_p = \sup_{U,V \in C_f(Z) \cap \mathcal{B}^d} U^T P\left(I - \mu A^T A\right) PV$

and $\tilde{\epsilon} = (2 + \rho_p)\epsilon$.

[Giryes, Eldar, Bronstein & Sapiro, 2016]

*We have a version of this theorem also when f is non-proper or non-convex function

CONVERGENCE RATE COMPARISON

- PGD convergence:

$$(\rho)^t$$

- IPGD convergence:

$$(\rho_P)^t + \frac{1-(\rho_P)^t}{1-\rho_P}(2+\rho_p)\epsilon$$

$$\overset{(a)}{\approx} (\rho_P)^t + \epsilon \overset{(b)}{\approx} (\rho_P)^t \overset{(c)}{\ll} (\rho)^t$$

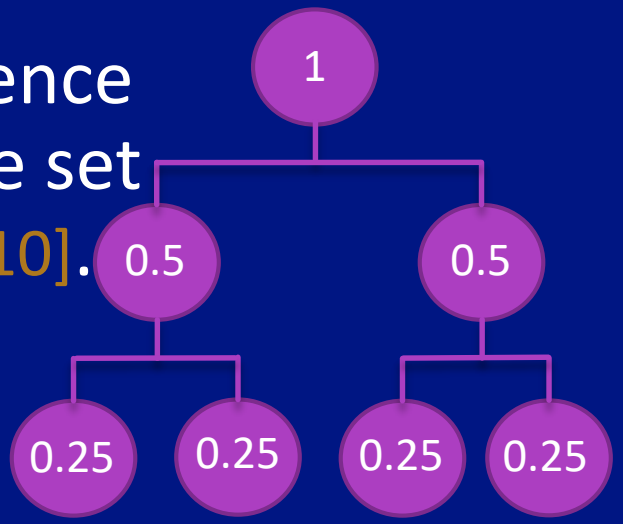
(a) ϵ is negligible compared to ρ_P

(b) For small values of t (early iterations).

(c) Faster convergence as $\rho_P \ll \rho$ (because $\omega_p \ll \omega$).

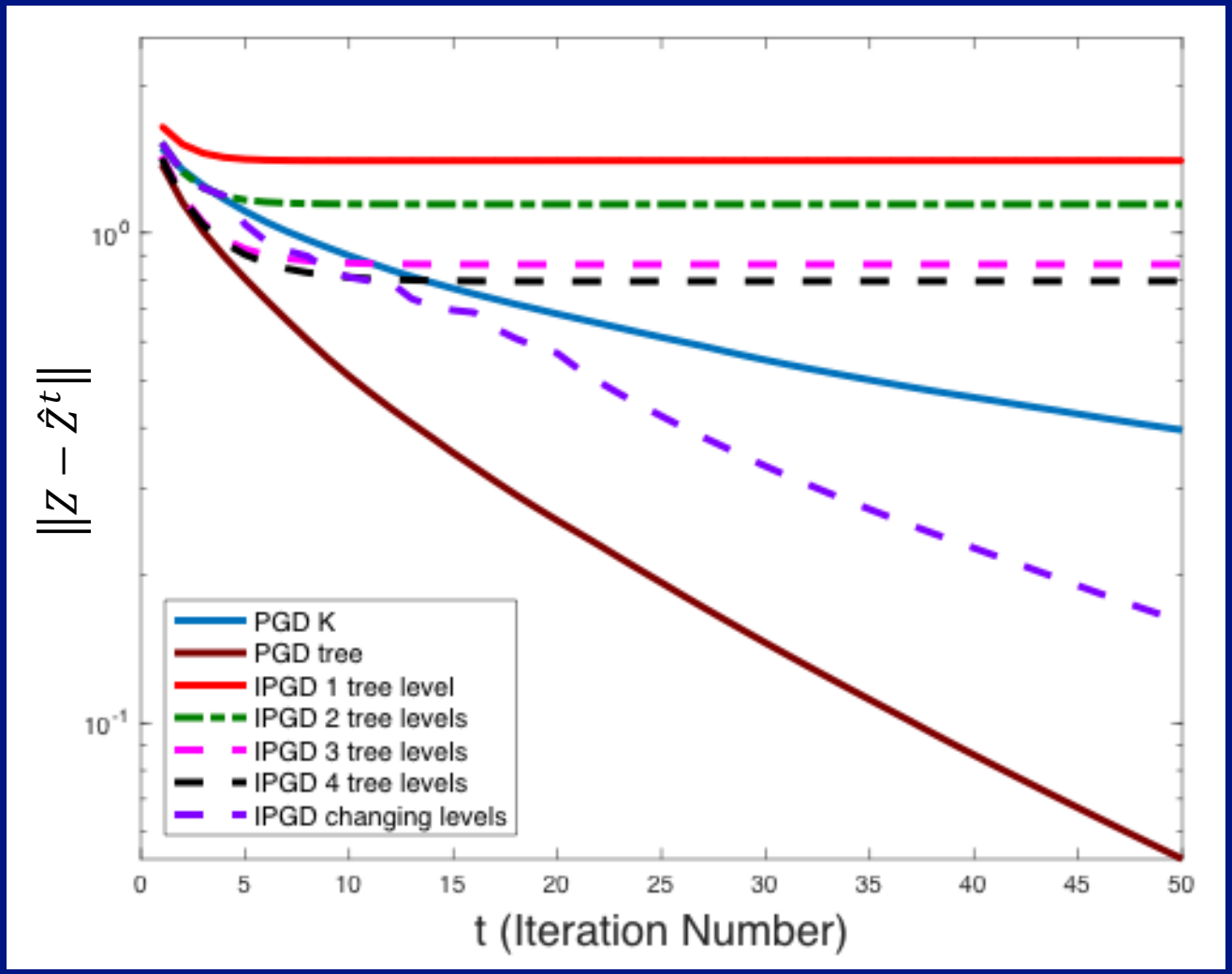
MODEL BASED COMPRESSED SENSING

- $\widehat{\Upsilon}$ is the set of sparse vectors with sparsity patterns that obey a tree structure.
- Projecting onto $\widehat{\Upsilon}$ improves convergence rate compared to projecting onto the set of sparse vectors Υ [Baraniuk et al., 2010].
- The projection onto $\widehat{\Upsilon}$ is more demanding than onto Υ.



- Note that the probability of selecting atoms from lower tree levels is smaller than upper ones.
- P will be a projection onto certain tree levels – zeroing the values at lower levels.

MODEL BASED COMPRESSED SENSING

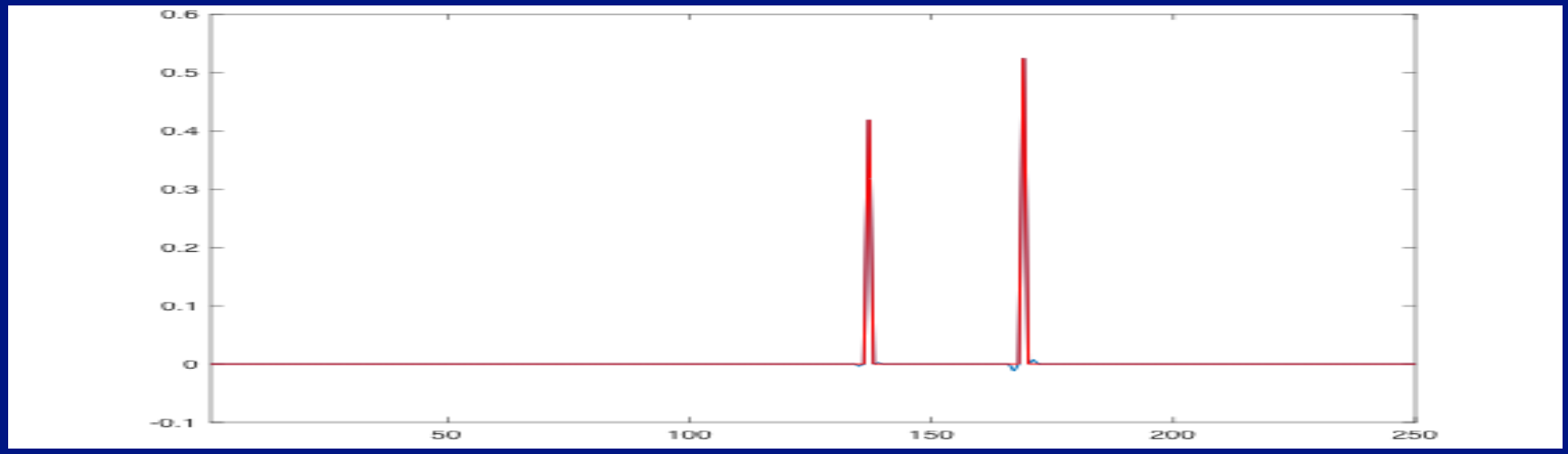


Non-zeros picked entries has zero mean random Gaussian distribution with variance:

- 1 at first two levels
- 0.5^2 at the third level
- 0.2^2 at the rest of the levels

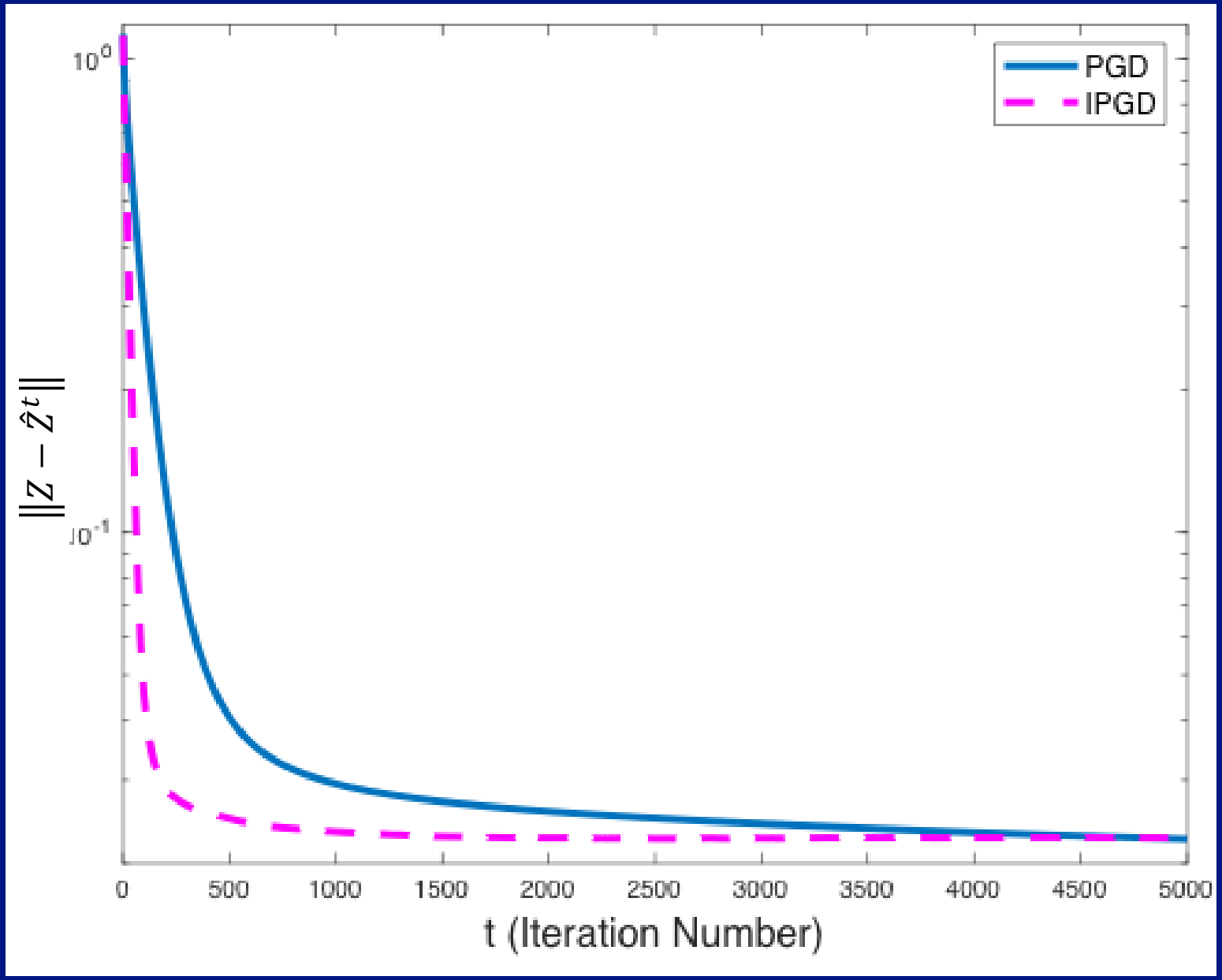
SPECTRAL COMPRESSED SENSING

- $\widehat{\Upsilon}$ is the set of vectors with sparse representation in a 2-times redundant DCT dictionary such that:



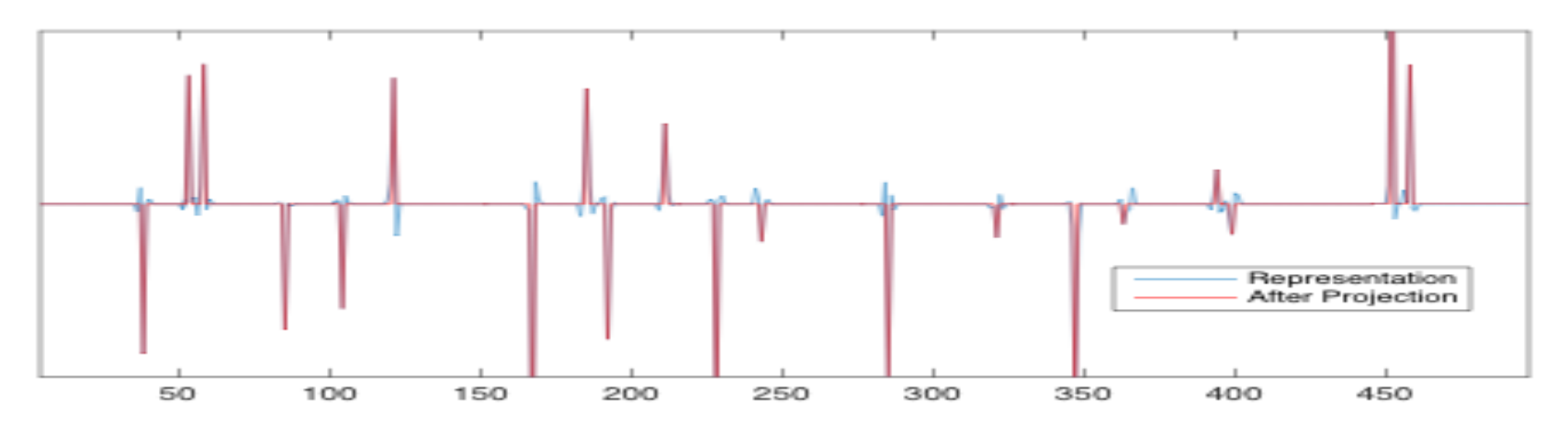
- We set P to be a pooling-like operation that keeps in each window of size 3 only the largest value.

SPECTRAL COMPRESSED SENSING



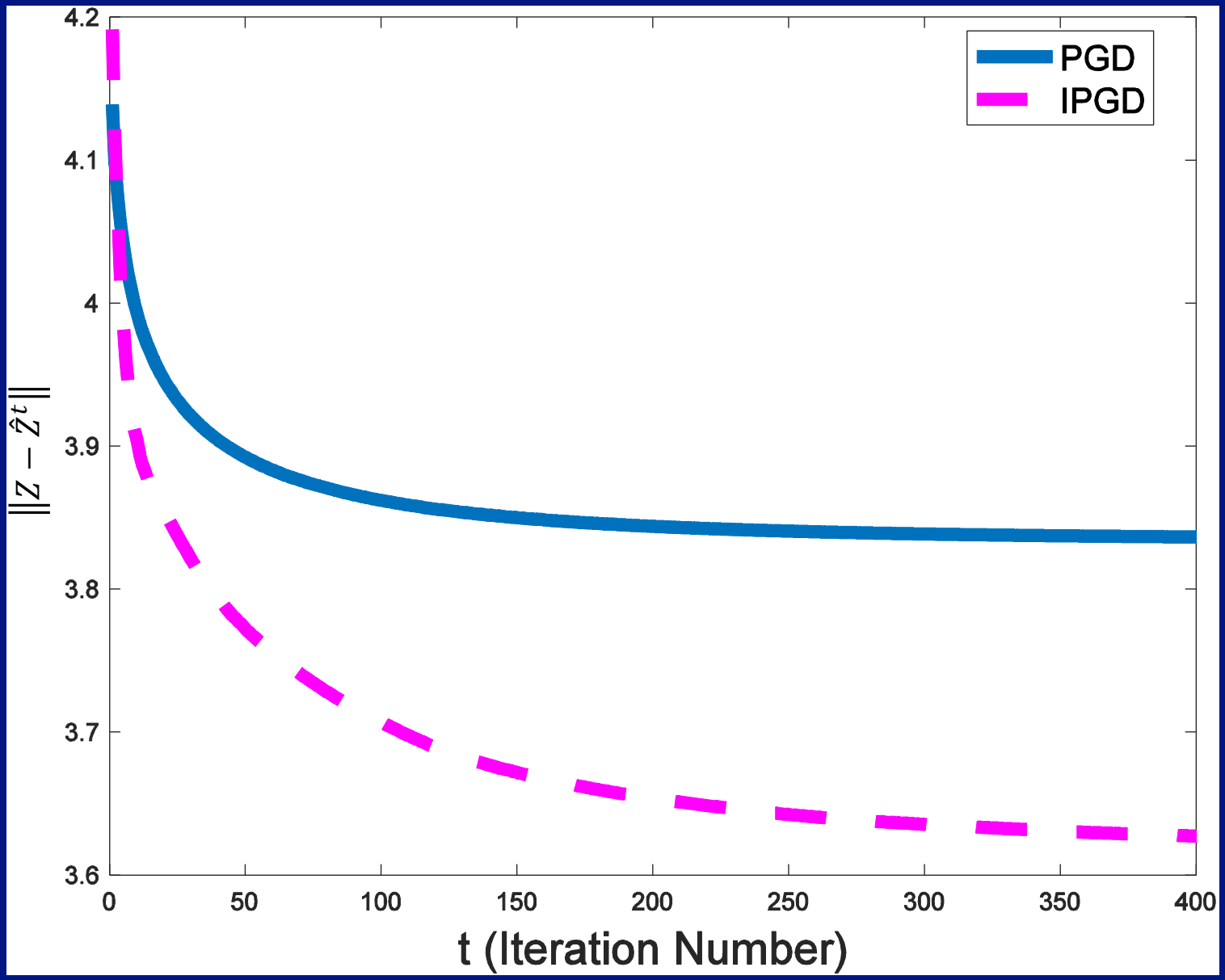
SPECTRAL COMPRESSED SENSING

- $\widehat{\Upsilon}$ is the set of vectors with sparse representation in a 4-times redundant DCT dictionary such that:



- We set P to be a pooling-like operation that keeps in each window of size 5 only the largest value.

SPECTRAL COMPRESSED SENSING

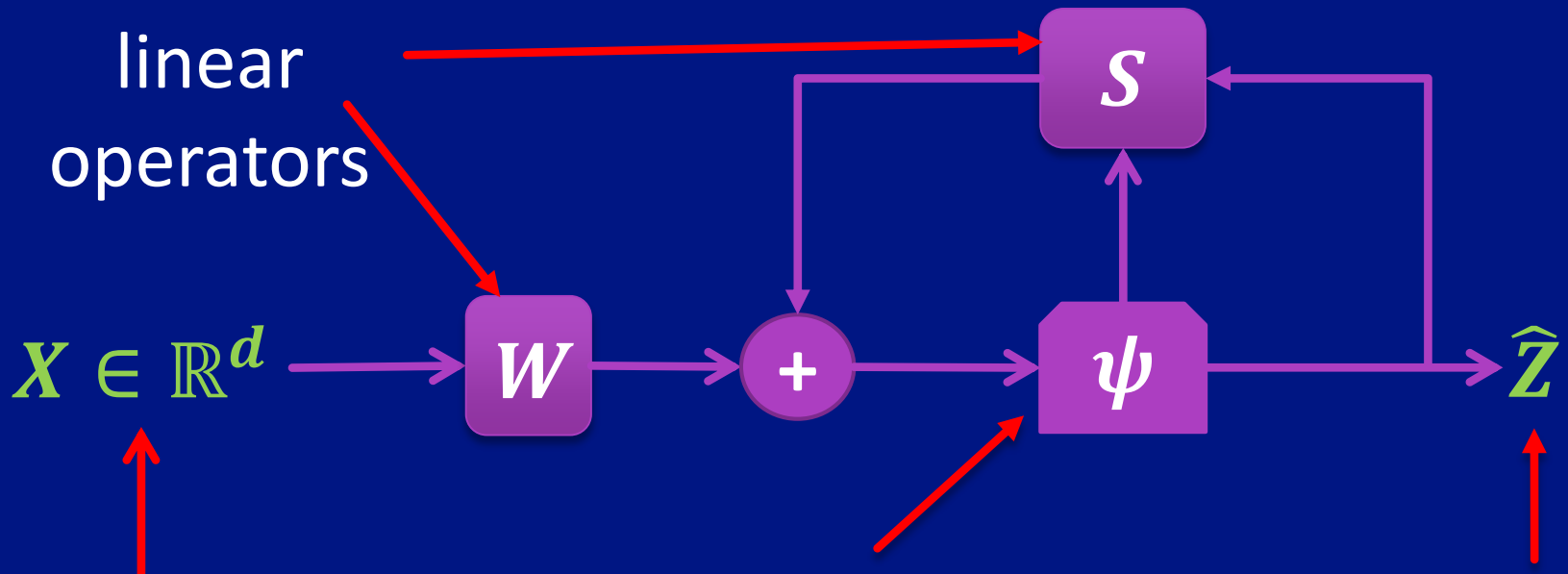


LEARNING THE PROJECTION

- If we have no explicit information about $\widehat{\Upsilon}$ it might be desirable to learn the projection.
- Instead of learning P, it is possible to replace $P\left(I - \mu A^T A\right)$ and $\mu P A^T$ with two learned matrices S and W respectively.
- This leads to a very similar scheme to the one of LISTA and provides a theoretical foundation for the success of LISTA.

LEARNED IPGD

Learned linear operators

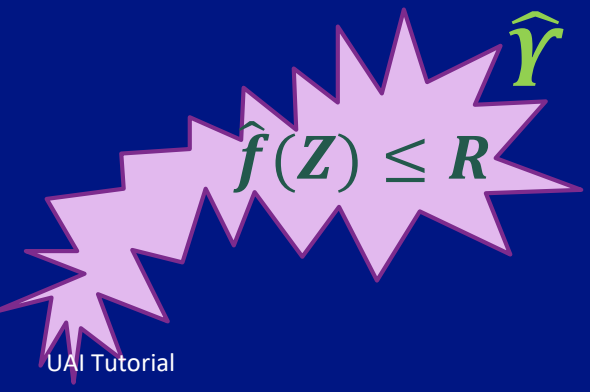


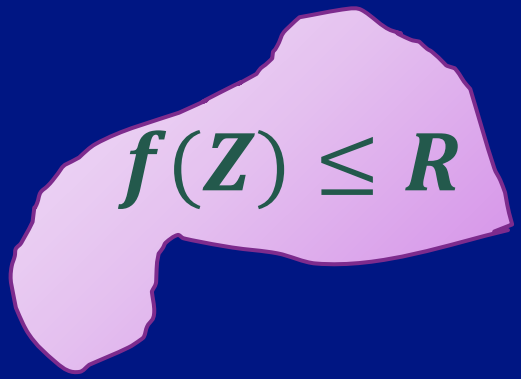
$X = AZ + E$

ψ projects onto the set $\mathcal{Y}$

Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|X - A\tilde{Z}\| \quad s.t.\ \hat{f}(\tilde{Z}) \leq R$$



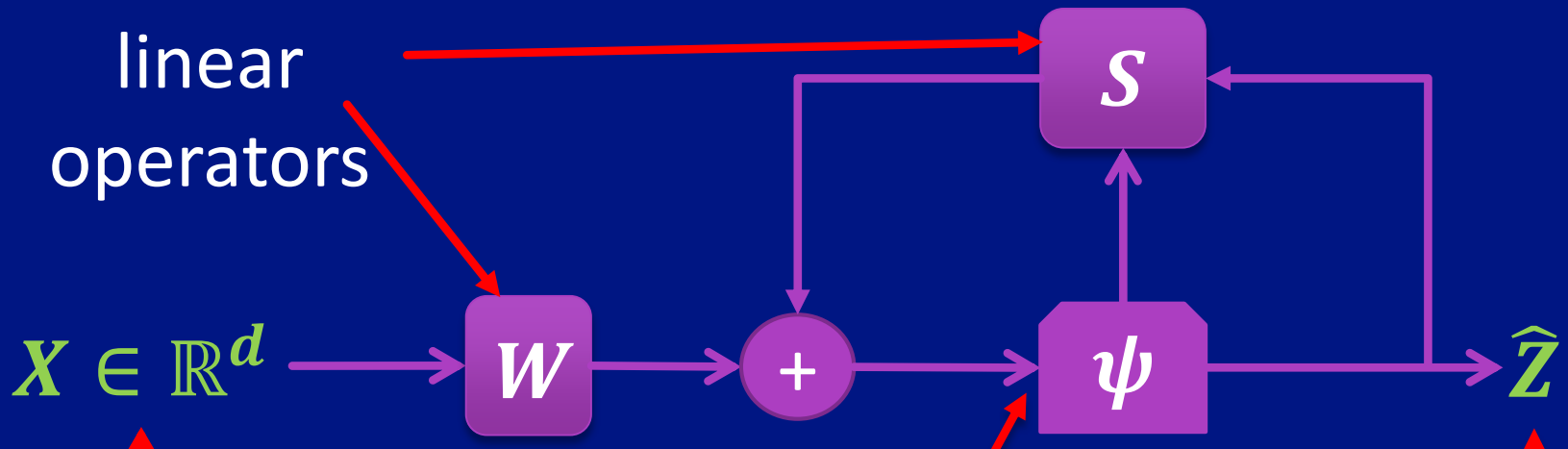


SUPER RESOLUTION

- A popular super-resolution technique uses a pair of low-res and high-res dictionaries [Zeyde et al. 2012]
- The original work uses OMP with sparsity 3 to decode the representation of patches in low-res image
- Then the representation is used to reconstruct the patches of the high-res image
- We replace OMP with LIPGD with 3 levels but higher target sparsity
- This leads to better reconstruction results (with up to 0.5dB improvement)

LISTA

Learned linear operators



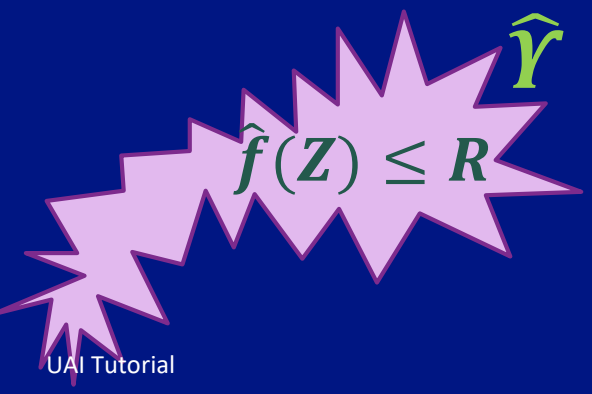
$X = AZ + E$

ψ is a proximal mapping.

$$\psi(U) = \underset{\widetilde{Z} \in \mathbb{R}^d}{\operatorname{argmin}} \|U - \widetilde{Z}\| + \lambda f(\widetilde{Z})$$

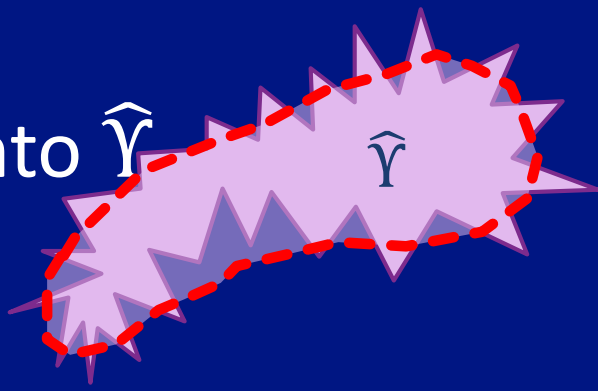
Estimate of Z. Aim at solving

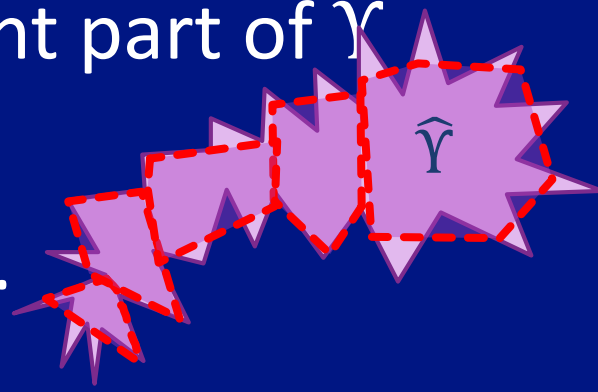
$$\min_{\widetilde{Z}} \|X - A\widetilde{Z}\| + \lambda \hat{f}(\widetilde{Z})$$



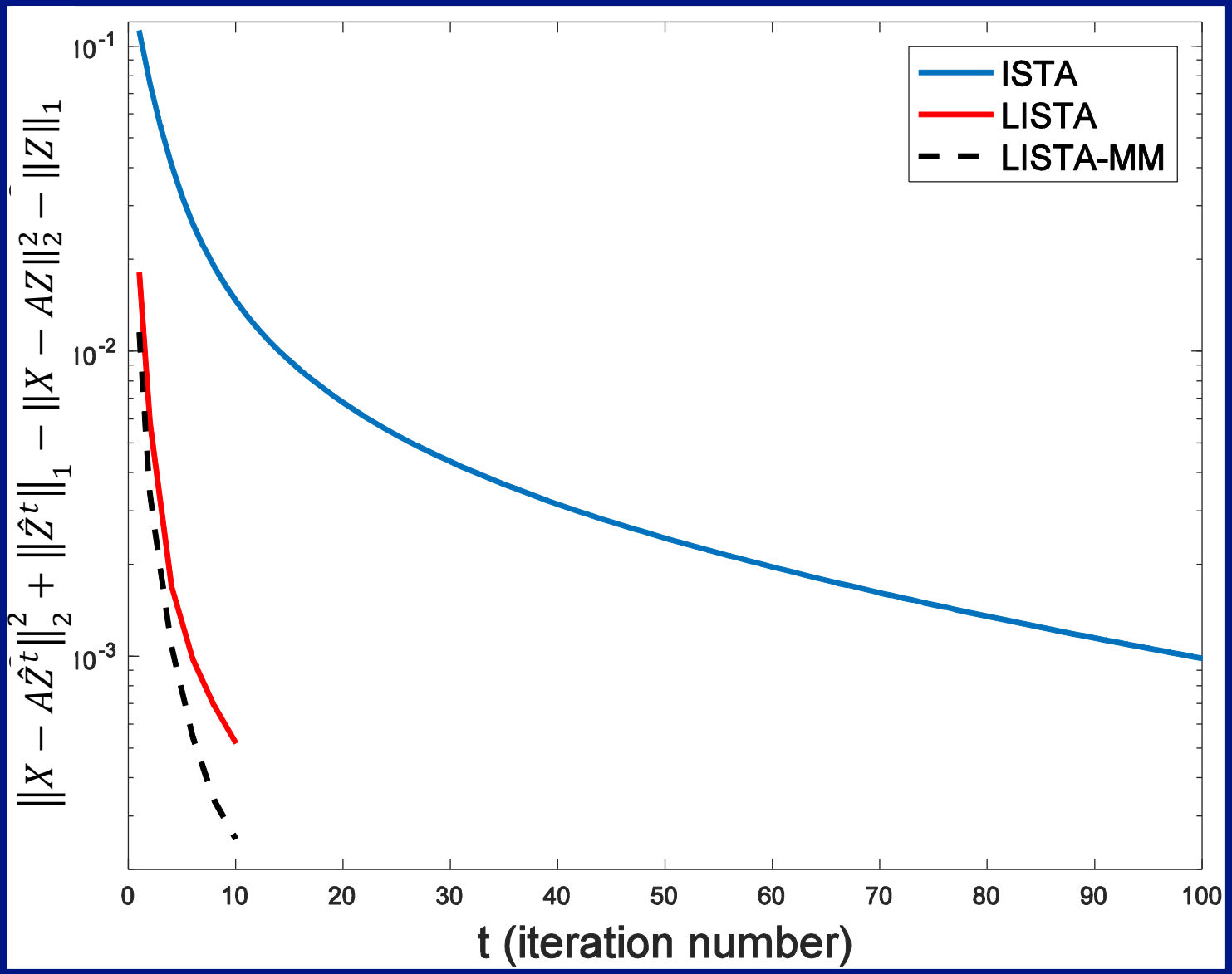
LISTA MIXTURE MODEL

- Approximation of the projection onto $\widehat{\Upsilon}$ with one linear projection may not be accurate enough.
- This requires more LISTA layers/iterations.
- Instead, one may use several LISTA networks, where each approximates a different part of $\widehat{\Upsilon}$
- Training multiple LISTA networks accelerate the convergence further.





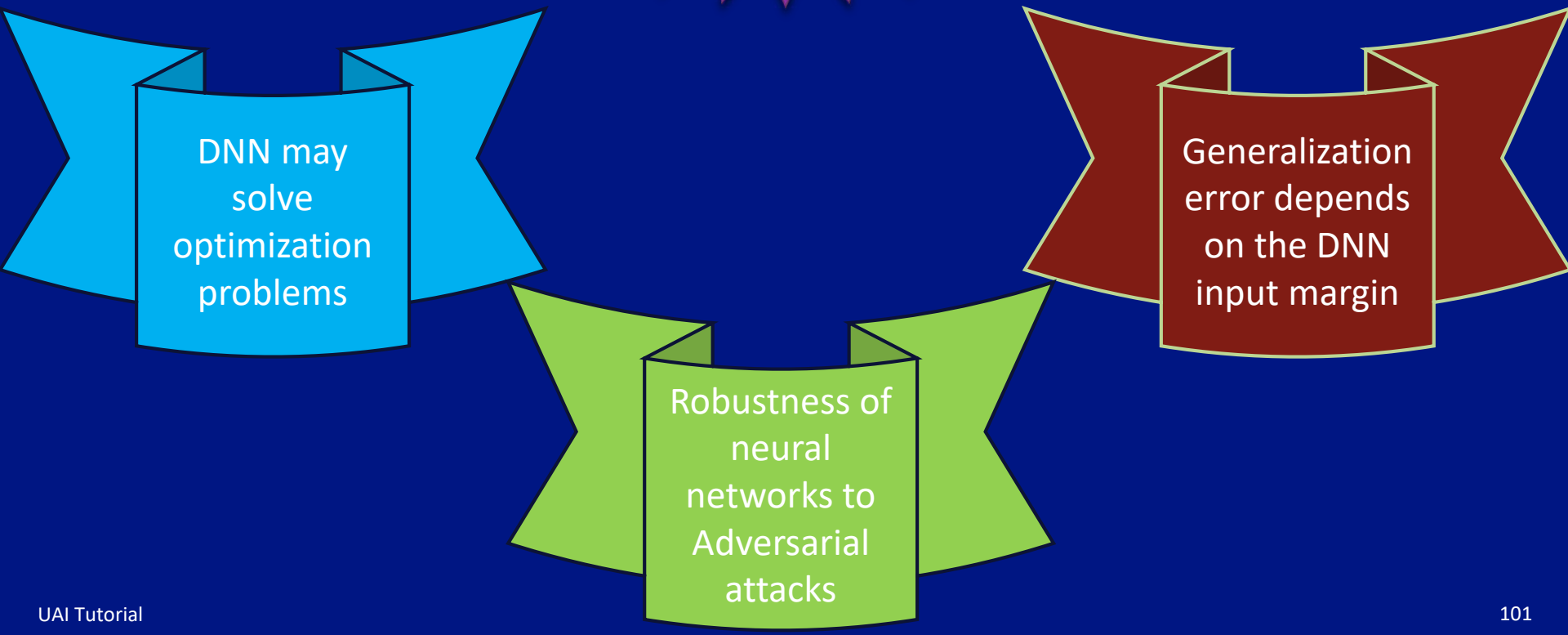
LISTA MIXTURE MODEL



RELATED WORKS

- In [Bruna et al. 2017] it is shown that a learning may give a gain due to better preconditioning of *A*.
- In [Xin et al. 2016] a relation to the restricted isometry property (RIP) is drawn
- In [Borgerding & Schniter, 2016] a connection is drawn to approximate message passing (AMP).
- In [Chen et al., 2018] and [Liu et al., 2019] tied and analytical weights are studied showing exponential convergence under some conditions.
- All these works consider only the sparsity case

Take Home Message



ACKNOWLEDGEMENTS

Yonina C. Eldar
Weizmann

Guillermo Sapiro
Duke University

Jure Sokolic
UCL

Alex M. Bronstein
Technion

Miguel Rodrigues
UCL

Daniel Jakubovitz
Tel Aviv University

QUESTIONS?

WEB.ENG.TAU.AC.IL/~RAJA

FULL REFERENCES 1

- A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal*, vol. 3, no. 3, pp. 535–554, 1959.
- D. H. Hubel & T. N. Wiesel, “Receptive fields of single neurones in the cat's striate cortex”, J Physiol., vol. 148, no. 3, pp. 574-591, 1959.
- D. H. Hubel & T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat's visual cortex”, J Physiol., vol. 160, no. 1, pp. 106-154, 1962.
- K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, Biological Cybernetics, vol. 36, no. 4, pp. 93-202, 1980.
- Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard & L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition”, Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.
- Y.LeCun, L. Bottou, Y. Bengio & P. Haffner, “Gradient Based Learning Applied to Document Recognition”, Proceedings of IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
- C. Farabet, C. Couprie, L. Najman & Y. LeCun, “Learning Hierarchical Features for Scene Labeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 35, no. 8, pp. 1915-1929, Aug. 2013.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge”, International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, NIPS, 2012.
- K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”, CVPR, 2016.

FULL REFERENCES 2

- M.D. Zeiler & R. Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV, 2014.
- D. Yu & L. Deng, “Automatic Speech Recognition: A Deep Learning Approach”, Springer, 2014.
- J. Bellegarda & C. Monz, “State of the art in statistical methods for language and speech processing,” Computer Speech and Language, vol. 35, pp. 163–184, Jan. 2016.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke & A. Rabinovich, “Going Deeper with Convolutions”, CVPR, 2015.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra & M. Riedmiller, “Playing Atari with Deep Reinforcement Learning”, NIPS deep learning workshop, 2013.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg & D. Hassabis, “Human-level control through deep reinforcement learning”, Nature vol. 518, pp. 529–533, Feb. 2015.
- D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel & D. Hassabis, “Mastering the Game of Go with Deep Neural Networks and Tree Search”, Nature, vol. 529, pp. 484–489, 2016.
- S. K. Zhou, H. Greenspan, D. Shen, “Deep Learning for Medical Image Analysis”, Academic Press, 2017.
- I. Sutskever, O. Vinyals & Q. Le, “Sequence to Sequence Learning with Neural Networks”, NIPS 2014.
- A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, “Large-scale Video Classification with Convolutional Neural Networks”, CVPR, 2014.

FULL REFERENCES 3

- F. Schroff, D. Kalenichenko & J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering”, CVPR, 2015.
- A. Poznanski & L. Wolf, “CNN-N-Gram for Handwriting Word Recognition”, CVPR, 2016.
- R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng & C. Potts, “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, EMNLP, 2013.
- H. C. Burger, C. J. Schuler & S. Harmeling, Image denoising: Can plain Neural Networks compete with BM3D?, CVPR, 2012.
- J. Kim, J. K. Lee, K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”, CVPR, 2016.
- J. Bruna, P. Sprechmann, and Y. LeCun, “Super-Resolution with Deep Convolutional Sufficient Statistics”, ICLR, 2016.
- V. Nair & G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines”, ICML, 2010.
- L. Deng & D. Yu, “Deep Learning: Methods and Applications”, Foundations and Trends in Signal Processing, vol. 7 no. 3-4, pp. 197–387, 2014.
- Y. Bengio, “Learning Deep Architectures for AI”, Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.
- Y. LeCun, Y. Bengio, & G. Hinton. Deep learning. Nature, vol. 521, no. 7553, pp. 436–444, 2015.
- J. Schmidhuber, “Deep learning in neural networks: An overview”, Neural Networks, vol. 61, pp. 85–117, Jan. 2015.
- I. Goodfellow, Y. Bengio & A. Courville, “Deep learning”, Book in preparation for MIT Press, 2016.

FULL REFERENCES 4

- G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Math. Control Signals Systems, vol. 2, pp. 303–314, 1989.
- K. Hornik, “Approximation capabilities of multilayer feedforward networks,” Neural Netw., vol. 4, no. 2, pp. 251–257, 1991.
- A. R. Barron, Approximation and estimation bounds for artificial neural networks, Machine Learning, vol. 14, no. 1, pp. 115–133, Jan. 1994.
- G. F. Montuf far & J. Morton, “When does a mixture of products contain a product of mixtures”, SIAM Journal on Discrete Mathematics (SIDMA), vol. 29, no. 1, pp. 321-347, 2015.
- G. F. Montuf far, R. Pascanu, K. Cho, & Y. Bengio, “On the number of linear regions of deep neural networks,” NIPS, 2014.
- N. Cohen, O. Sharir & A. Shashua, “Deep SimNets,” CVPR, 2016.
- N. Cohen, O. Sharir & A. Shashua, “On the Expressive Power of Deep Learning: A Tensor Analysis,” COLT, 2016.
- N. Cohen & A. Shashua, “Convolutional Rectifier Networks as Generalized Tensor Decompositions,” ICML, 2016
- M. Telgarsky, “Benefits of depth in neural networks,” COLT, 2016.
- R. Eldan and O. Shamir, “The power of depth for feedforward neural networks.,” COLT, 2016.
- N. Cohen and A. Shashua, “Inductive Bias of Deep Convolutional Networks through Pooling Geometry,” arXiv abs/ 1605.06743, 2016.
- J. Bruna, Y. LeCun, & A. Szlam, “Learning stable group invariant representations with convolutional networks,” ICLR, 2013.
- Y-L. Boureau, J. Ponce, Y. LeCun, Theoretical Analysis of Feature Pooling in Visual Recognition, ICML, 2010.

FULL REFERENCES 5

- J. Bruna, A. Szlam, & Y. LeCun, “Signal recovery from lp pooling representations”, ICML, 2014.
- S. Soatto & A. Chiuso, “Visual Representations: Defining properties and deep approximation”, ICLR 2016.
- F. Anselmi, J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio, “Unsupervised learning of invariant representations in hierarchical architectures,” Theoretical Computer Science, vol. 663, no. C, pp. 112-121, Jun. 2016.
- J. Bruna and S. Mallat, “Invariant scattering convolution networks,” IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol. 35, no. 8, pp. 1872–1886, Aug 2013.
- T. Wiatowski and H. Bölcskei, “A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction,” arXiv abs/1512.06293, 2016
- A. Saxe, J. McClelland, and S. Ganguli, “Exact solutions to the nonlinear dynamics of learning in deep linear neural network”, ICLR, 2014.
- Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high dimensional non-convex optimization,” NIPS, 2014.
- A. Choromanska, M. B. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The loss surfaces of multilayer networks,” in International Conference on Artificial Intelligence and Statistics (AISTATS), 2015.
- B. D. Haeffele and R. Vidal. Global Optimality in Tensor Factorization, Deep Learning, and Beyond. arXiv, abs/1506.07540, 2015.
- S. Arora, A. Bhaskara, R. Ge, and T. Ma, “Provable bounds for learning some deep representations,” in Int. Conf. on Machine Learning (ICML), 2014, pp. 584–592.

FULL REFERENCES 6

- A. M. Bruckstein, D. L. Donoho, & M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images”, SIAM Review, vol. 51, no. 1, pp. 34–81, 2009.
- G. Yu, G. Sapiro & S. Mallat, “Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity”, IEEE Trans. on Image Processing, vol. 21, no. 5, pp. 2481 –2499, May 2012.
- N. Srebro & A. Shraibman, “Rank, trace-norm and max-norm,” COLT, 2005.
- E. Cand`es & B. Recht, “Exact matrix completion via convex optimization,” Foundations of Computational mathematics, vol. 9, no. 6, pp. 717– 772, 2009.
- R. G. Baraniuk, V. Cevher & M. B. Wakin, "Low-Dimensional Models for Dimensionality Reduction and Signal Recovery: A Geometric Perspective," Proceedings of the IEEE, vol. 98, no. 6, pp. 959-971, 2010.
- Y. Plan and R. Vershynin, “Dimension reduction by random hyperplane tessellations,” Discrete and Computational Geometry, vol. 51, no. 2, pp. 438–461, 2014.
- Y. Plan and R. Vershynin, “Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach,” IEEE Trans. Inf. Theory, vol. 59, no. 1, pp. 482–494, Jan. 2013.
- R. Giryes, G. Sapiro and A.M. Bronstein, “Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy? “, IEEE Transactions onSignal Processing, vol. 64, no. 13, pp. 3444-3457, Jul. 2016.
- A. Choromanska, K. Choromanski, M. Bojarski, T. Jebara, S. Kumar, Y. LeCun, “Binary embeddings with structured hashed projections”, ICML, 2016.
- J. Masci, M. M. Bronstein, A. M. Bronstein and J. Schmidhuber, “Multimodal Similarity-Preserving Hashing”, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 36, no. 4, pp. 824-830, April 2014.

FULL REFERENCES 7

- H. Lai, Y. Pan, Y. Liu & S. Yan, “Simultaneous Feature Learning and Hash Coding With Deep Neural Networks”, CVPR, 2015.
- A. Mahendran & A. Vedaldi, “Understanding deep image representations by inverting them,” CVPR, 2015.
- K. Simonyan & A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, ICLR, 2015
- A. Krogh & J. A. Hertz, “A Simple Weight Decay Can Improve Generalization”, NIPS, 1992.
- P. Baldi & P. Sadowski, “Understanding dropout”, NIPS, 2013.
- N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.
- L. Wan, M. Zeiler, S. Zhang, Y. LeCun & R. Fergus, “Regularization of Neural Networks using DropConnect”, ICML, 2013.
- S. Ioffe & C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” ICML, 2015.
- M. Hardt, B. Recht & Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent”, ICML, 2016.
- B. Neyshabur, R. Salakhutdinov & N. Srebro, “Path-SGD: Path-normalized optimization in deep neural networks,” NIPS, 2015.
- S. Rifai, P. Vincent, X. Muller, X. Glorot, & Y. Bengio. “Contractive auto-encoders: explicit invariance during feature extraction,” ICML, 2011.

FULL REFERENCES 8

- T. Salimans & D. Kingma, “Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks”, arXiv abs/1602.07868, 2016.
- S. Sun, W. Chen, L. Wang, & T.-Y. Liu, “Large margin deep neural networks: theory and algorithms”, AAAI, 2016.
- S. Shalev-Shwartz & S. Ben-David. “Understanding machine learning: from theory to algorithms”, Cambridge University Press, 2014.
- P. L. Bartlett & S. Mendelson, “Rademacher and Gaussian complexities: risk bounds and structural results”. The Journal of Machine Learning Research (JMLR), vol 3, pp. 463–482, 2002.
- B. Neyshabur, R. Tomioka, and N. Srebro, “Norm-based capacity control in neural networks,” COLT, 2015.
- J. Sokolic, R. Giryes, G. Sapiro, M. R. D. Rodrigues, “Margin Preservation of Deep Neural Networks”, arXiv, abs/1605.08254, 2016.
- H. Xu and S. Mannor. “Robustness and generalization,” JMLR, vol. 86, no. 3, pp. 391–423, 2012.
- J. Huang, Q. Qiu, G. Sapiro, R. Calderbank, “Discriminative Geometry-Aware Deep Transform”, ICCV 2015
- J. Huang, Q. Qiu, G. Sapiro, R. Calderbank, “Discriminative Robust Transformation Learning”, NIPS 2016.
- T. Blumensath & M.E. Davies, “Iterative hard thresholding for compressed sensing”, Appl. Comput. Harmon. Anal, vol. 27, no. 3, pp. 265 – 274, 2009.
- I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint”, Communicationson Pure and Applied Mathematics, vol. 57, no. 11, pp. 1413–1457, 2004.

FULL REFERENCES 9

- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci., 2(1):183–202, Mar. 2009.
- K. Gregor & Y. LeCun, “Learning fast approximations of sparse coding”, ICML, 2010.
- P. Sprechmann, A. M. Bronstein & G. Sapiro, “Learning efficient sparse and low rank models”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1821–1833, Sept. 2015.
- T. Remez, O. Litany, & A. M. Bronstein, “A picture is worth a billion bits: Real-time image reconstruction from dense binary pixels”, ICCP, 2015.
- J. Tompson, K. Schlachter, P. Sprechmann & K. Perlin, “Accelerating Eulerian Fluid Simulation With Convolutional Networks”, arXiv, abs/1607.03597, 2016.
- S. Oymak, B. Recht, & M. Soltanolkotabi, “Sharp time–data tradeoffs for linear inverse problems”, arXiv, abs/1507.04793, 2016.
- R. Giryes, Y. C. Eldar, A. M. Bronstein, G. Sapiro, “Tradeoffs between Convergence Speed and Reconstruction Accuracy in Inverse Problems”, arXiv, abs/1605.09232, 2016.
- R.G. Baraniuk, V. Cevher, M.F. Duarte & C. Hegde, “Model-based compressive sensing”, IEEE Trans. Inf. Theory, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.
- M. F. Duarte & R. G. Baraniuk, “Spectral compressive sensing”, Appl. Comput. Harmon. Anal., vol. 35, no. 1, pp. 111 – 129, 2013.
- J. Bruna & T. Moreau, Adaptive Acceleration of Sparse Coding via Matrix Factorization, arXiv abs/1609.00285, 2016.