HOW STRUCTURE CAN IMPROVE THE THEORY AND PRACTICE IN NEURAL NETWORKS?

RAJA GIRYES

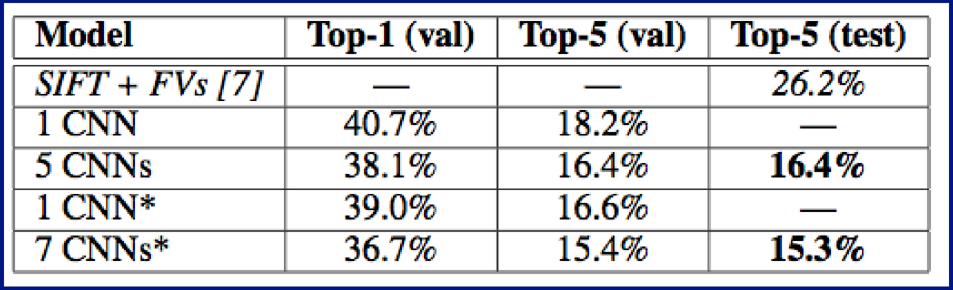
TEL AVIV UNIVERSITY

ACCESS workshop, KTH, Stockholm, Sweden

May 14, 2017

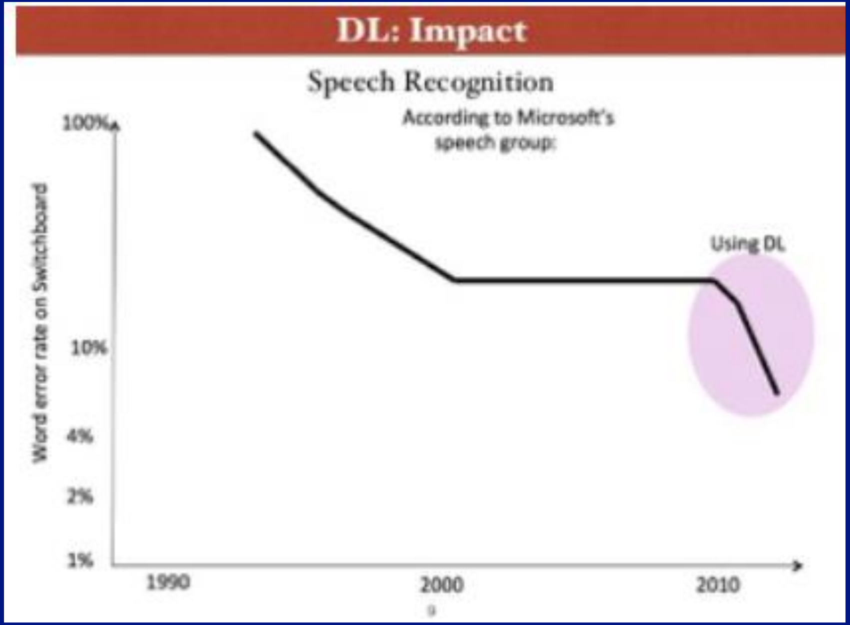
DEEP LEARNING IMPACT

- 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



DEEP NEURAL NETWORKS (DNN)

- One layer of a neural net

$$\begin{tikzcd} V \in \mathbb{R}^d \arrow[r] & \boxed{X} \arrow[r, "VX"] & \boxed{\psi} \arrow[r] & \psi(VX) \in \mathbb{R}^m \end{tikzcd}$$

X is a linear operation

F is a non-linear function

- Concatenation of the layers creates the whole net

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

$$\begin{tikzcd} V \in \mathbb{R}^d \arrow[r] & \boxed{X^1} \arrow[r] & \boxed{\psi} \arrow[r, dashed] & \boxed{X^i} \arrow[r] & \boxed{\psi} \arrow[r, dashed] & \boxed{X^K} \arrow[r] & \boxed{\psi} \arrow[r] & {} \end{tikzcd}$$

CONVOLUTIONAL NEURAL NETWORKS (CNN)

$$\begin{tikzcd} \boldsymbol{V} \in \mathbb{R}^{\boldsymbol{d}} \arrow[r] & \boxed{\boldsymbol{X}} \arrow[r, "VX"] & \boxed{\boldsymbol{\psi}} \arrow[r] & \boldsymbol{\psi}(\boldsymbol{VX}) \in \mathbb{R}^{\boldsymbol{m}} \end{tikzcd}$$

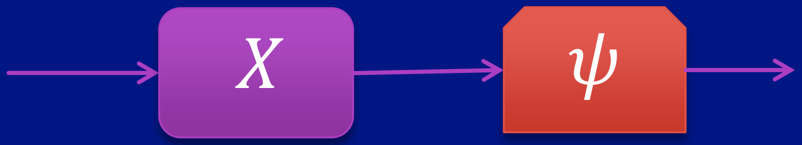
X is a linear operation

F is a non-linear function

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

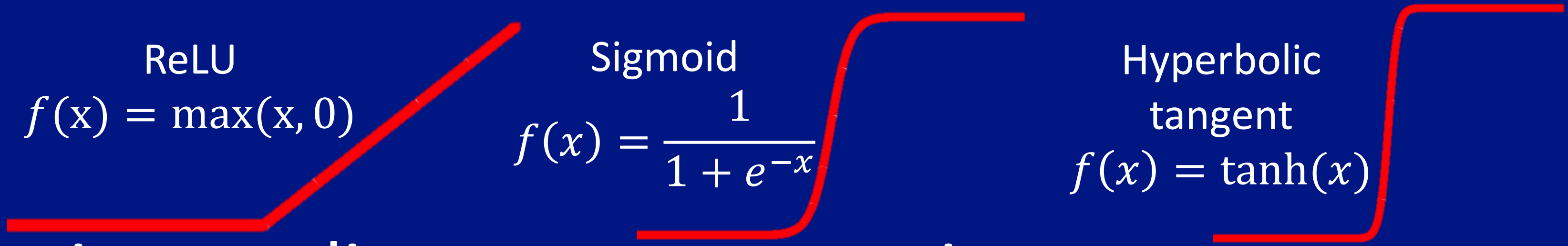
THE NON-LINEAR PART

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



$$\begin{tikzcd} {} \arrow[r] & \boxed{X} \arrow[r] & \boxed{\psi} \arrow[r] & {} \end{tikzcd}$$

- f is the (point-wise) activation function

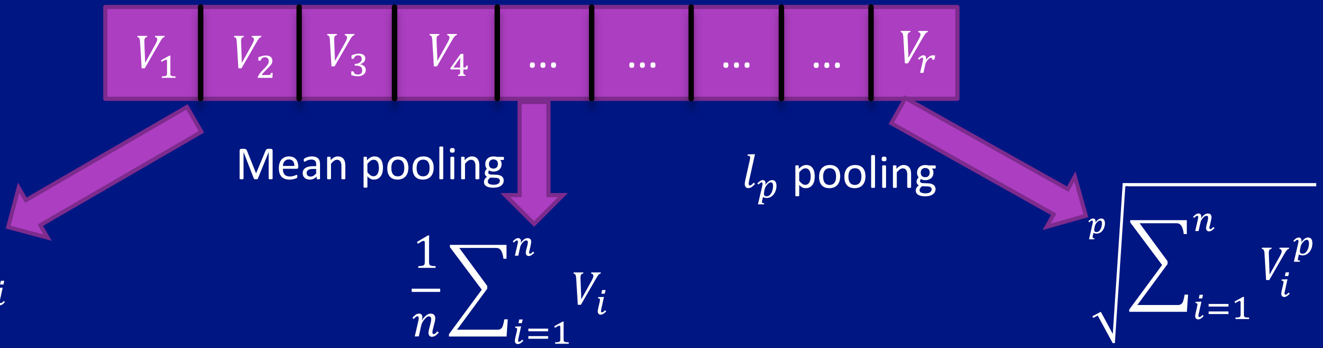


ReLU
$f(\mathrm{x}) = \max(\mathrm{x}, 0)$

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

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

- g is a pooling or an aggregation operator.



V_1	V_2	V_3	V_4	…	…	…	…	V_r

Max pooling
$$\max_i V_i$$

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

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

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 the activation function?

What is the role of pooling?

What happens to the data throughout the layers?

SAMPLE OF RELATED EXISTING THEORY

- Universal approximation for any measurable Borel functions [Hornik et. al., 1989, Cybenko 1989]
- Depth of a network provides an exponential complexity compared to the number parameters [Montúfar et al. 2014], invariance to more complex deformations [Bruna & Mallat, 2013] and better modeling of correlations of the input [Cohen et al. 2016]
- 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 stage provides shift invariance [Bruna et al. 2013]
- Relation of pooling and phase retrieval [Bruna et al. 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]
- Relation to dictionary learning [Papayan et al. 2016].
- DNN with Random Gaussian weights are good for classifying the average points in the data and an important goal of training is to classify the boundary points between the different classes in the data. [Giryes et al. 2016]

OUTLINE

DNN may solve optimization problems

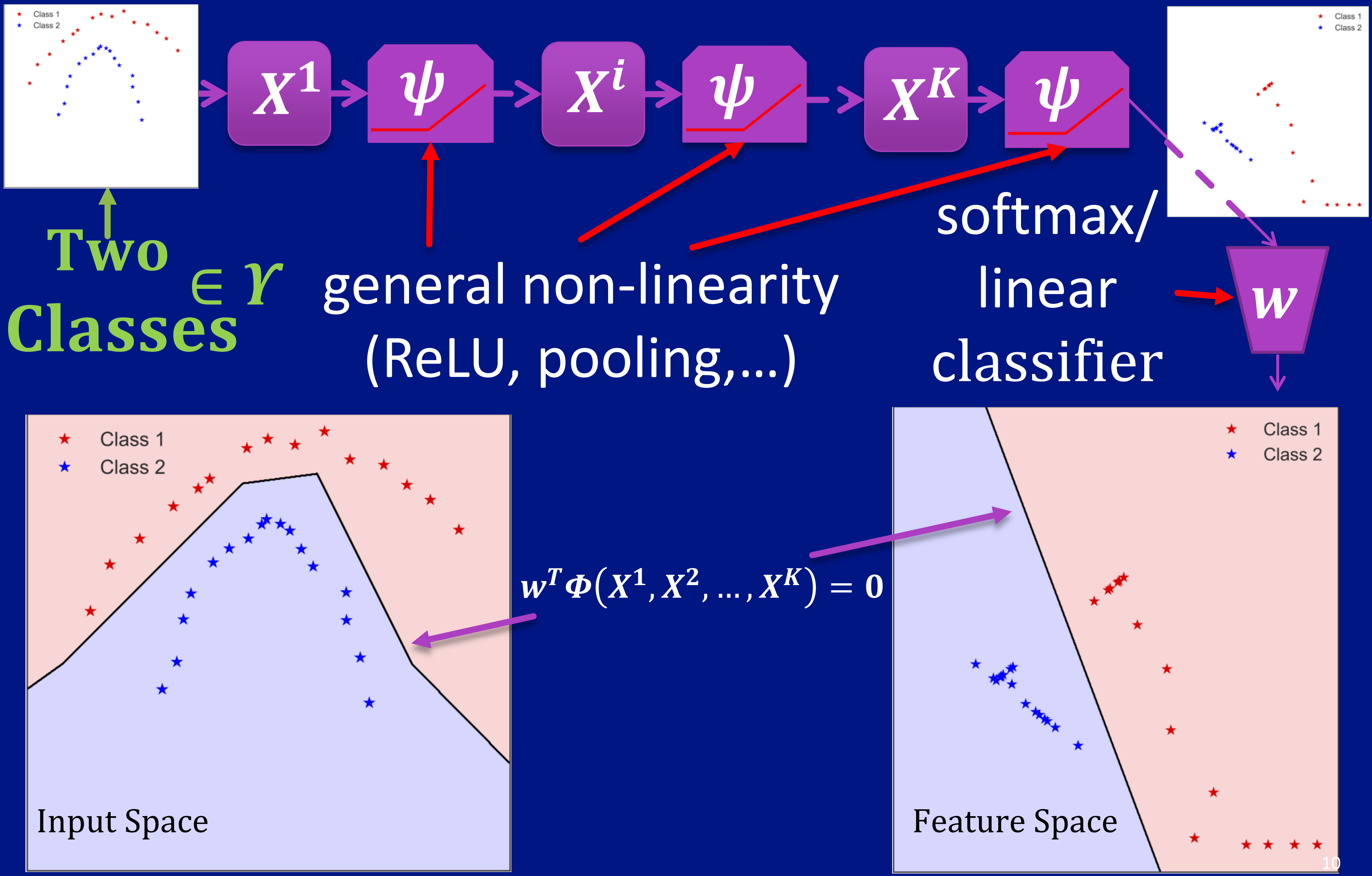
Generalization error depends on the DNN input margin

Generalization Error

DNN may solve optimization problems

Generalization error depends on the DNN input margin

NETWORK STRUCTURE



Class 1
Class 2
X^1
ψ
X^i
ψ
X^K
ψ
Two Classes $\in \Upsilon$
general non-linearity (ReLU, pooling,...)
softmax/ linear classifier
w
Class 1
Class 2
Input Space
$w^T\Phi(X^1, X^2, \dots, X^K) = 0$
Class 1
Class 2
Feature Space

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

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

⇨ It is important to understand the GE of DNN

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

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

[Neyshabur et al., 2015].

- Note that in both cases the GE grows with the depth

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

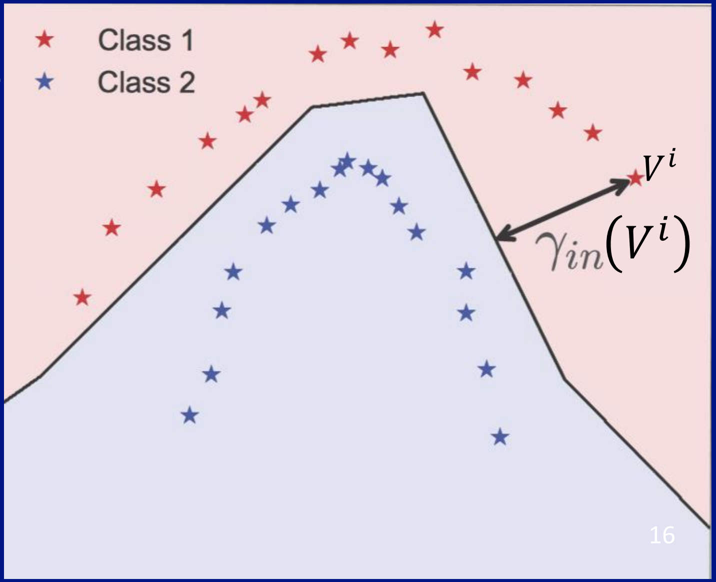
DNN INPUT MARGIN

- Theorem 6: If for every input margin $\gamma_{in}(V^i) > \gamma$

  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].



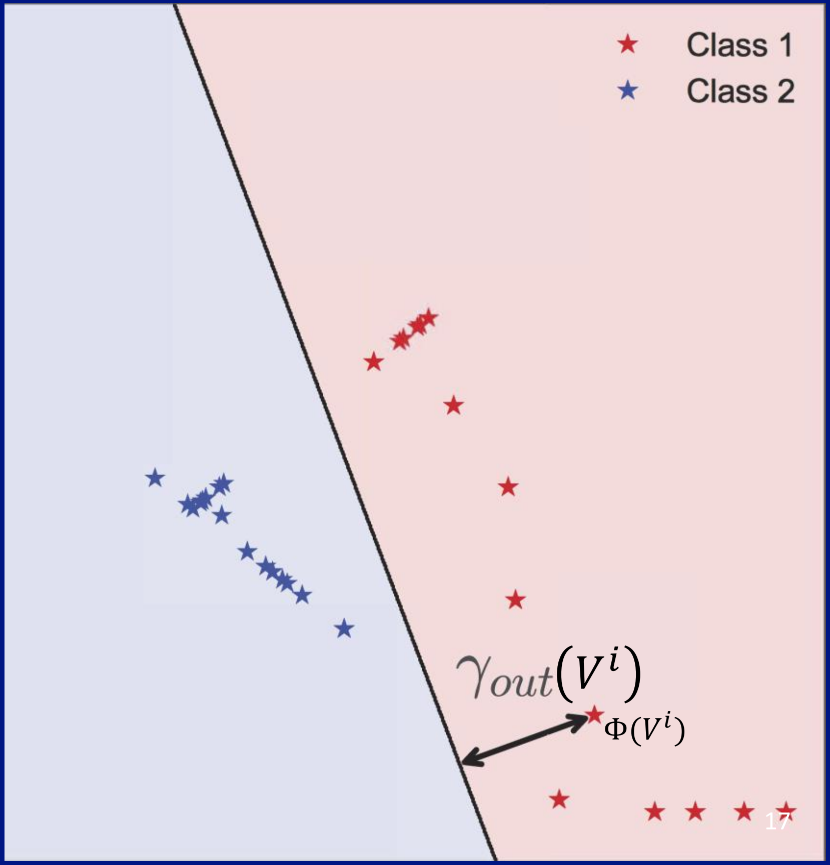
INPUT MARGIN BOUND

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

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

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

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



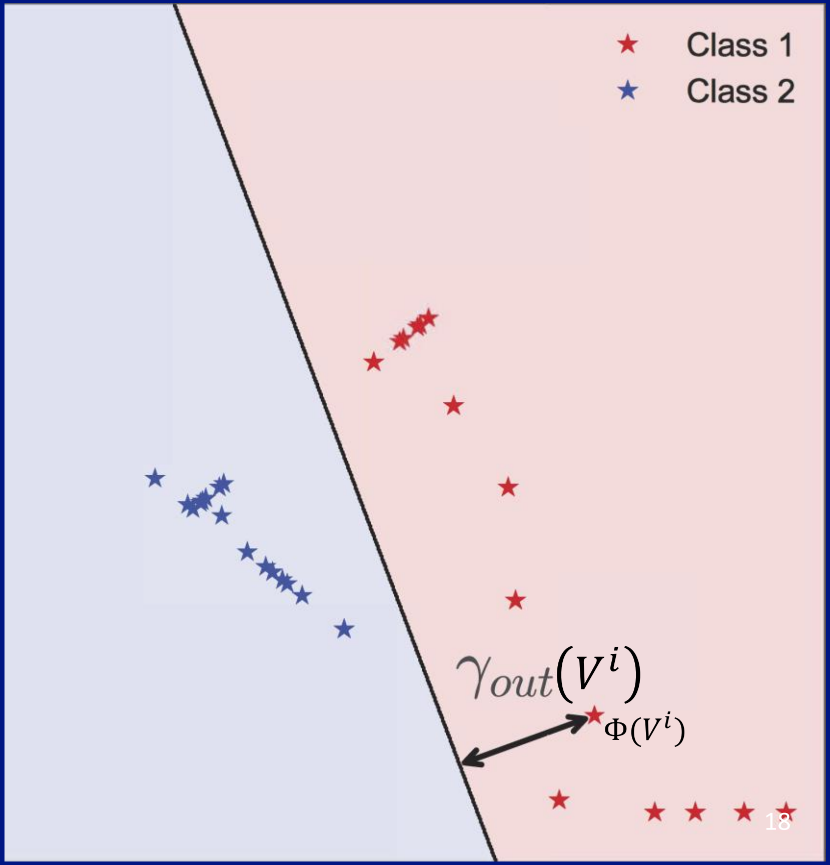
[Sokolic, Giryes, Sapiro, Rodrigues, 2017]

OUTPUT MARGIN

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

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

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

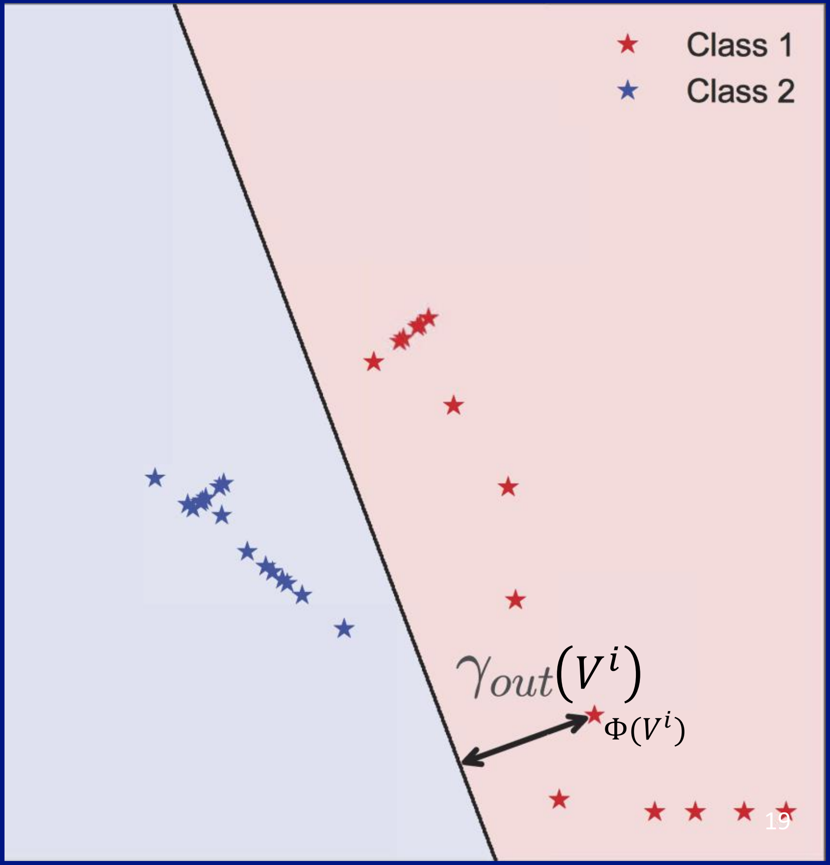


GE AND WEIGHT DECAY

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

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

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



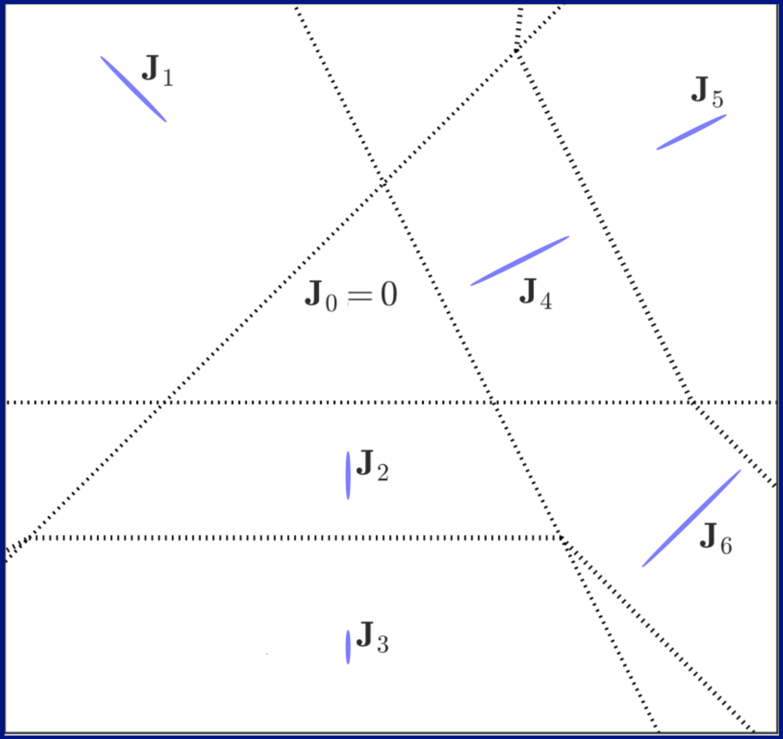
JACOBIAN BASED REGULARIZATION

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

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

- $J(V)$ is the Jacobian of the DNN at point V.
- $J(\cdot)$ is piecewise constant.
- Using the Jacobian of the DNN leads to a better bound.

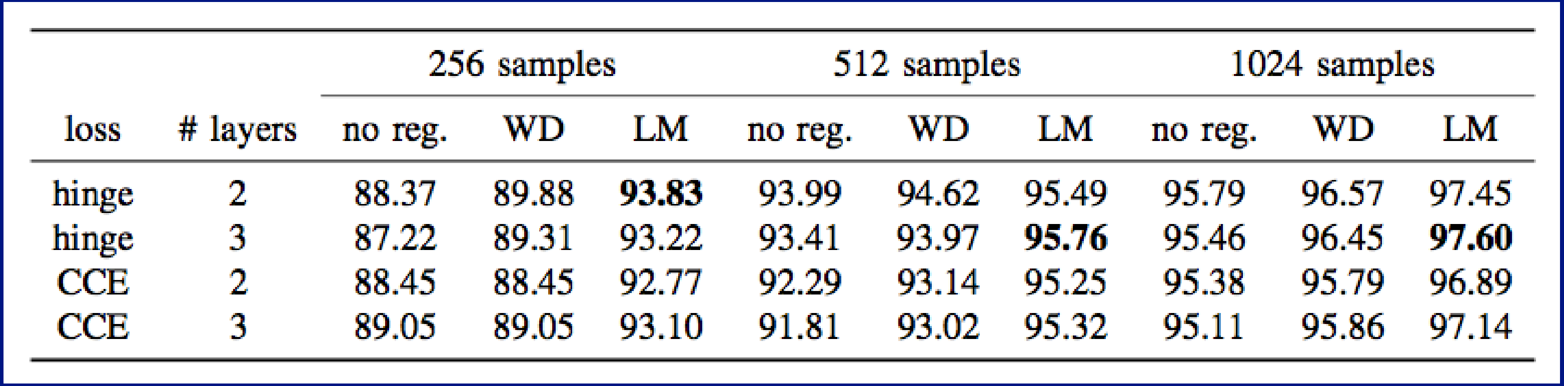
⇨ New regularization technique.



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.

[Sokolic, Giryes, Sapiro, Rodrigues, 2017]

- 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.

INVARIANCE

- Our theory extends also to study of the relation between invariance in the data and invariance in the network
- We have proposed also a new strategy to enforce invariance in the network [Sokolic, Giryes, Sapiro, Rodrigues, 2017]

Minimization by DNN

DNN may solve optimization problems

Generalization error depends on the DNN input margin

INVERSE PROBLEMS

- We are given $\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{A} + \boldsymbol{E}$

Given set of measurements

Unknown signal

linear operator

noise

- Standard technique for recovery

$$\min_{\boldsymbol{Z}} \|\boldsymbol{V} - \boldsymbol{Z}\boldsymbol{A}\|_2 \qquad \textbf{s.t.} \qquad \boldsymbol{Z} \in \boldsymbol{\Upsilon}$$

$\boldsymbol{Z}$ resides in a low dimensional set $\boldsymbol{\Upsilon}$

- Unconstrained form

$$\min_{\boldsymbol{Z}} \|\boldsymbol{V} - \boldsymbol{Z}\boldsymbol{A}\|_2^2 + \lambda f(\boldsymbol{Z})$$

Regularization parameter

$\boldsymbol{f}$ is a penalty function

ℓ_1 MINIMIZATION CASE

- Unconstrained form

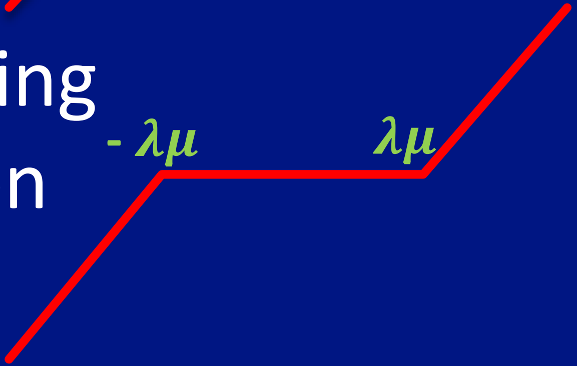
$$\min_{Z} \|V - ZA\|_2^2 + \lambda\|Z\|_1$$

- Can be solved by iterative shrinkage and thresholding technique (ISTA)

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

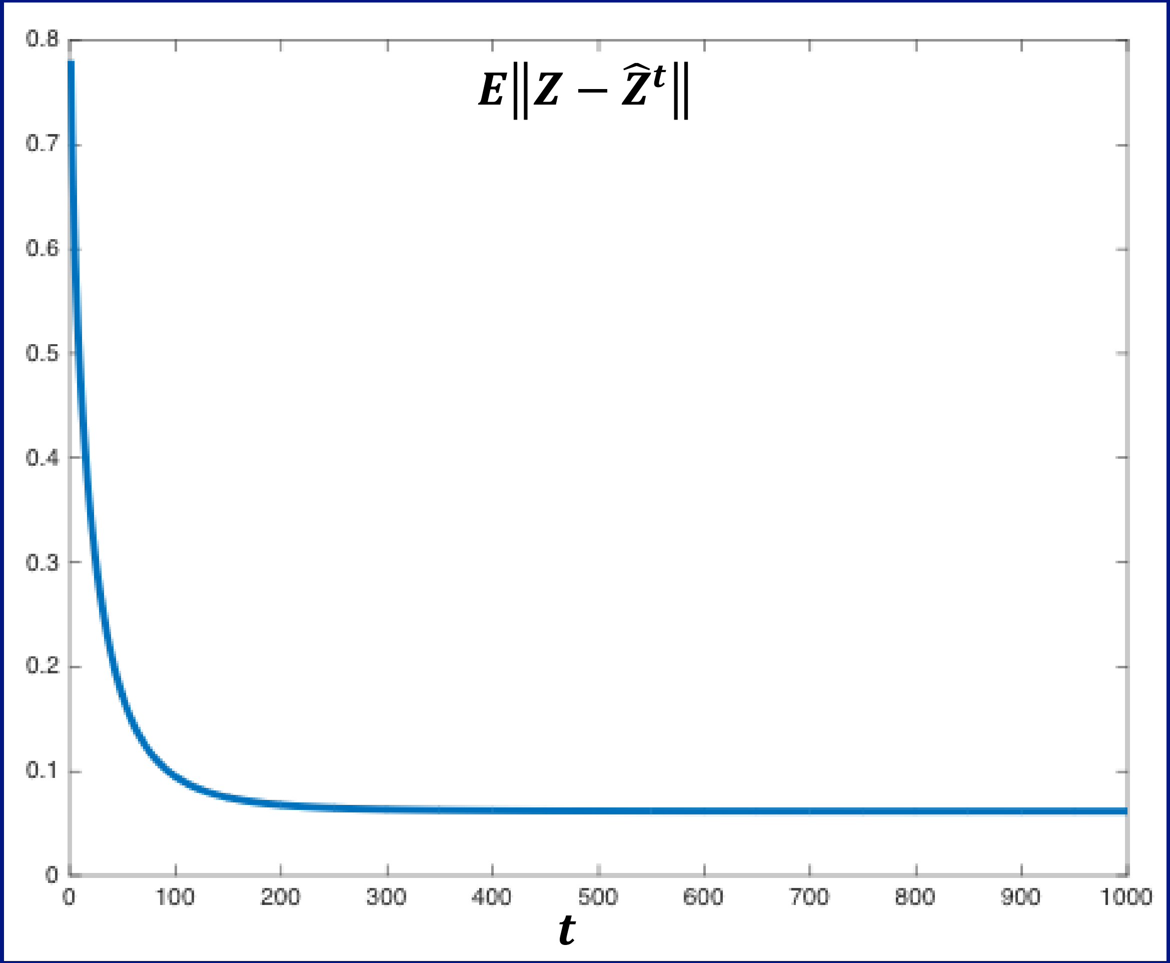
Soft thresholding operation

μ is the step size



ISTA CONVERGENCE

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



LISTA

- ISTA

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

- Rewriting ISTA:

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

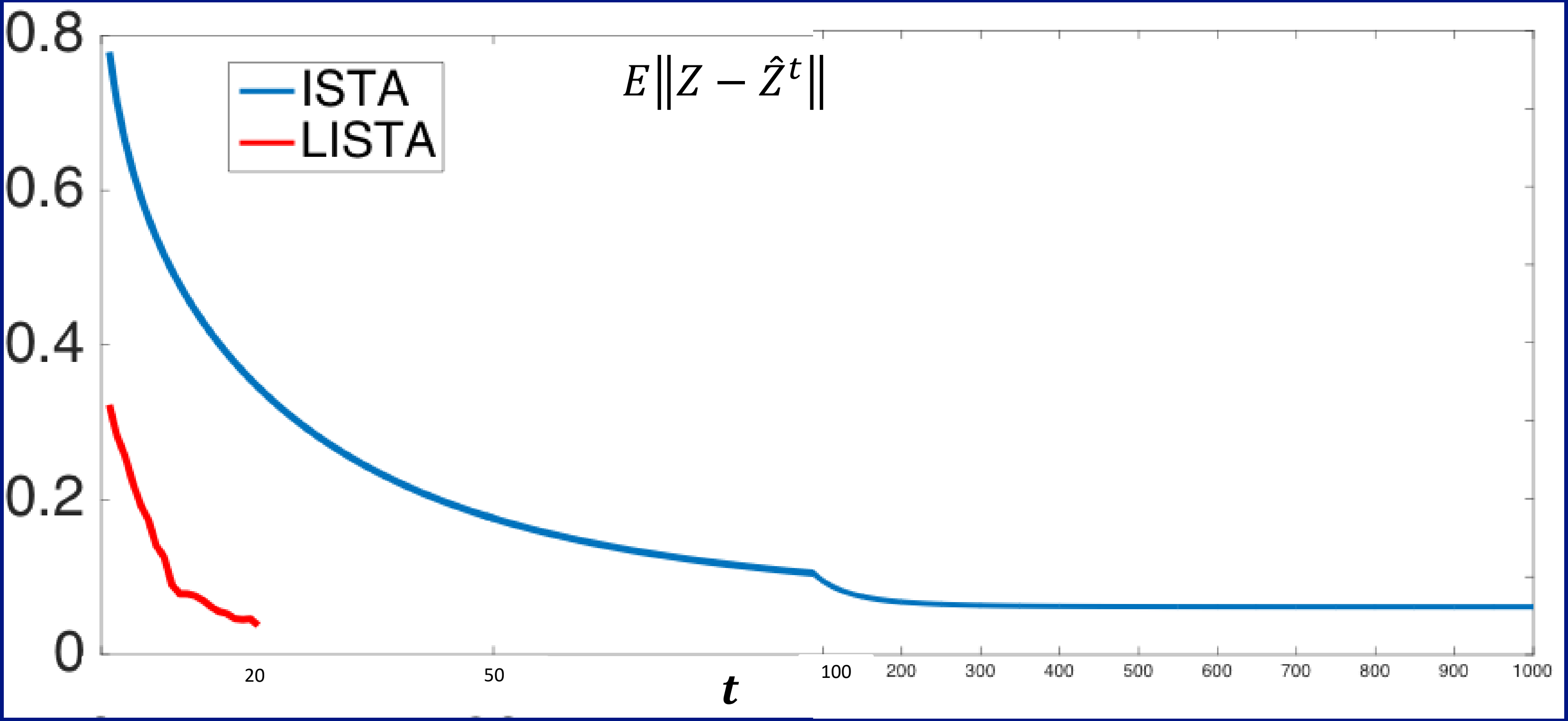
- Learned ISTA (LISTA):

$$\boldsymbol{Z}^{t+1} = \psi_{\lambda}(\boldsymbol{Z}^t\boldsymbol{X} + \boldsymbol{V}\boldsymbol{S})$$

Learned operators

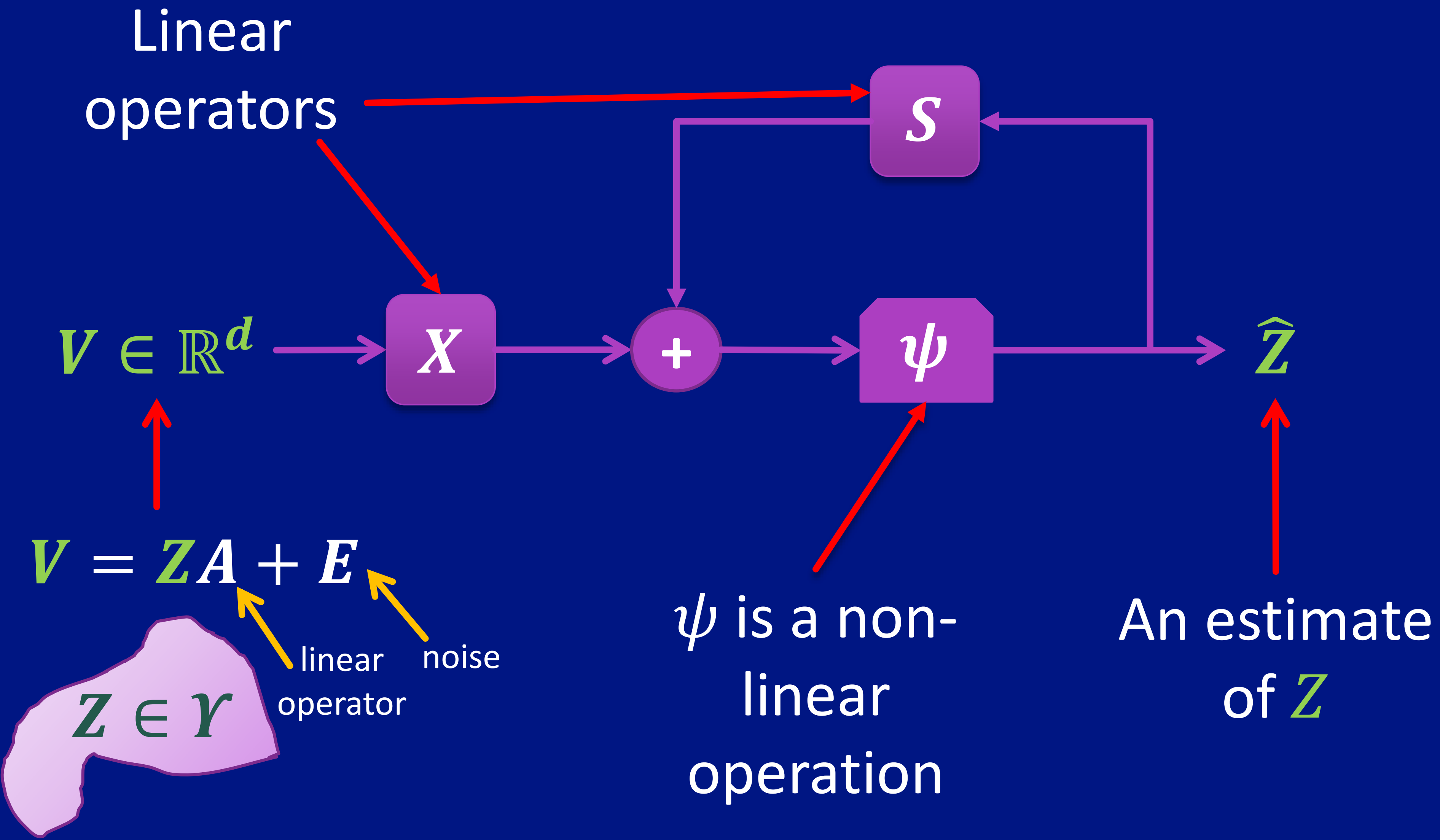
LISTA CONVERGENCE

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



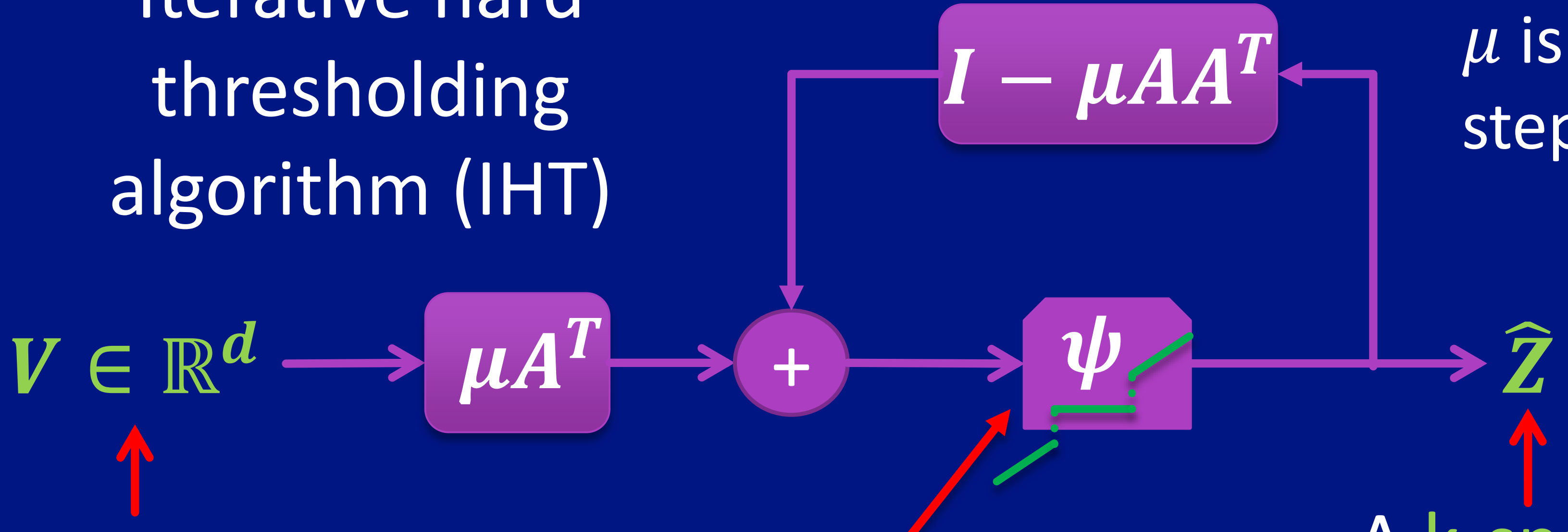
- Extensions to other models [Sprechmann, Bronstein & Sapiro, 2015], [Remez, Litani & Bronstein, 2015], [Tompson, Schlachter, Sprechmann & Perlin, 2016].

LISTA AS A NEURAL NETWORK



ℓ_0-MINIMIZATION

Iterative hard thresholding algorithm (IHT)


$$\begin{tikzcd} V \in \mathbb{R}^d \arrow[r] & \boxed{\mu A^T} \arrow[r] & + \arrow[r] & \boxed{\psi} \arrow[r] \arrow[d, bend left] & \hat{Z} \\ & & & \boxed{I - \mu A A^T} \arrow[ul] & \end{tikzcd}$$

μ is the step size

$V = ZA + E$

Z is a k–sparse vecotr

ψ is the hard thresholding operation: keeps the largest k entries

A k-sparse estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t \ \|\tilde{Z}\|_0 \le \mathrm{k}$$

[Blumensath & Davies, 2009]

ℓ_1-MINIMIZATION

Projected gradient descent algorithm for ℓ_1 minimization

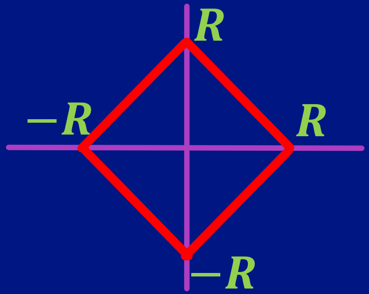
μ is the step size

$$\begin{tikzcd} & & \boxed{I-\mu AA^T} \arrow[d] & & \\ V\in\mathbb{R}^d \arrow[r] & \boxed{\mu A^T} \arrow[r] & + \arrow[r] & \boxed{\psi} \arrow[r] \arrow[ul] & \hat{Z} \end{tikzcd}$$

$V = ZA + E$

$\|Z\|_1 \leq R$

ψ projects onto the ℓ_1 ball

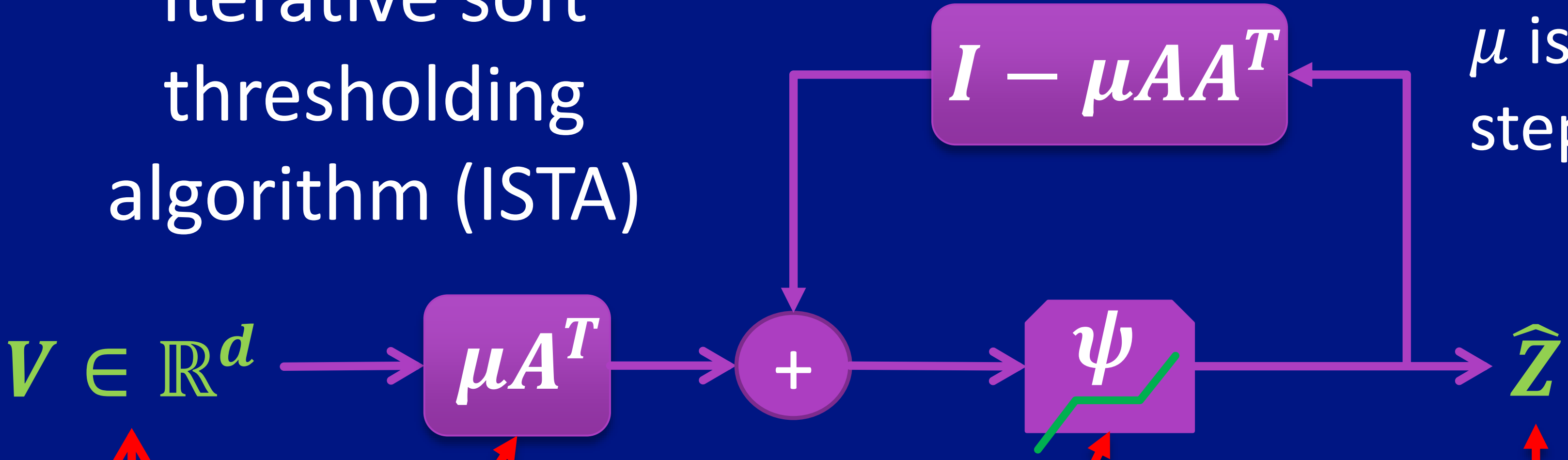


Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\|$$
$$s.t \ \|\tilde{Z}\|_1 \leq R$$

UNCONSTRAINED ℓ_1-MINIMIZATION

Iterative soft thresholding algorithm (ISTA)

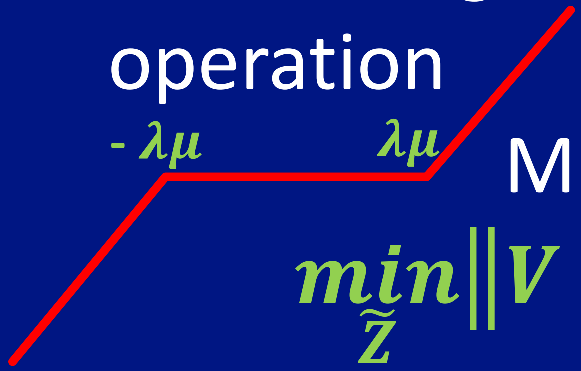


μ is the step size

Step size μ obeys

$$\frac{1}{\mu} \geq \|A\|$$

Soft thresholding operation



$V = ZA + E$

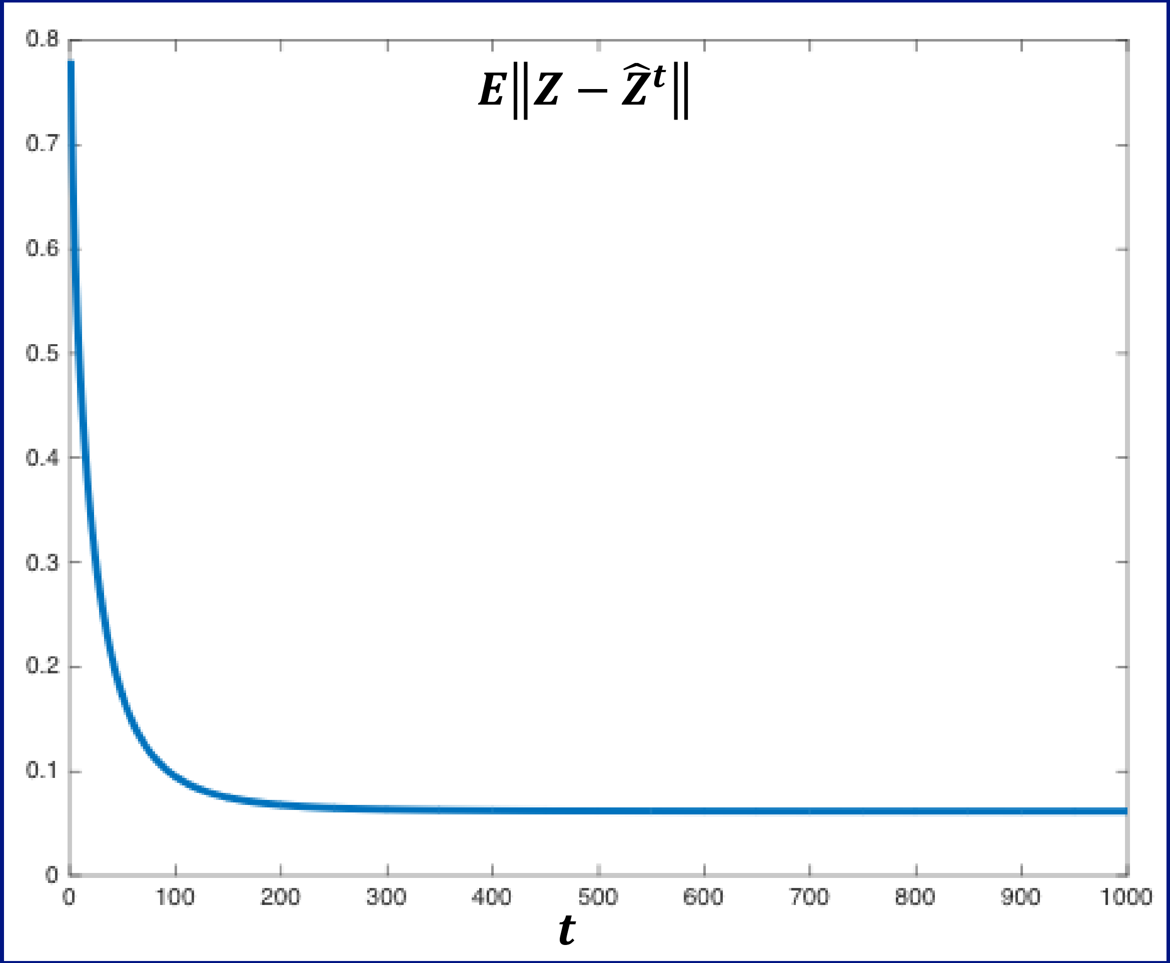
Minimizer of

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

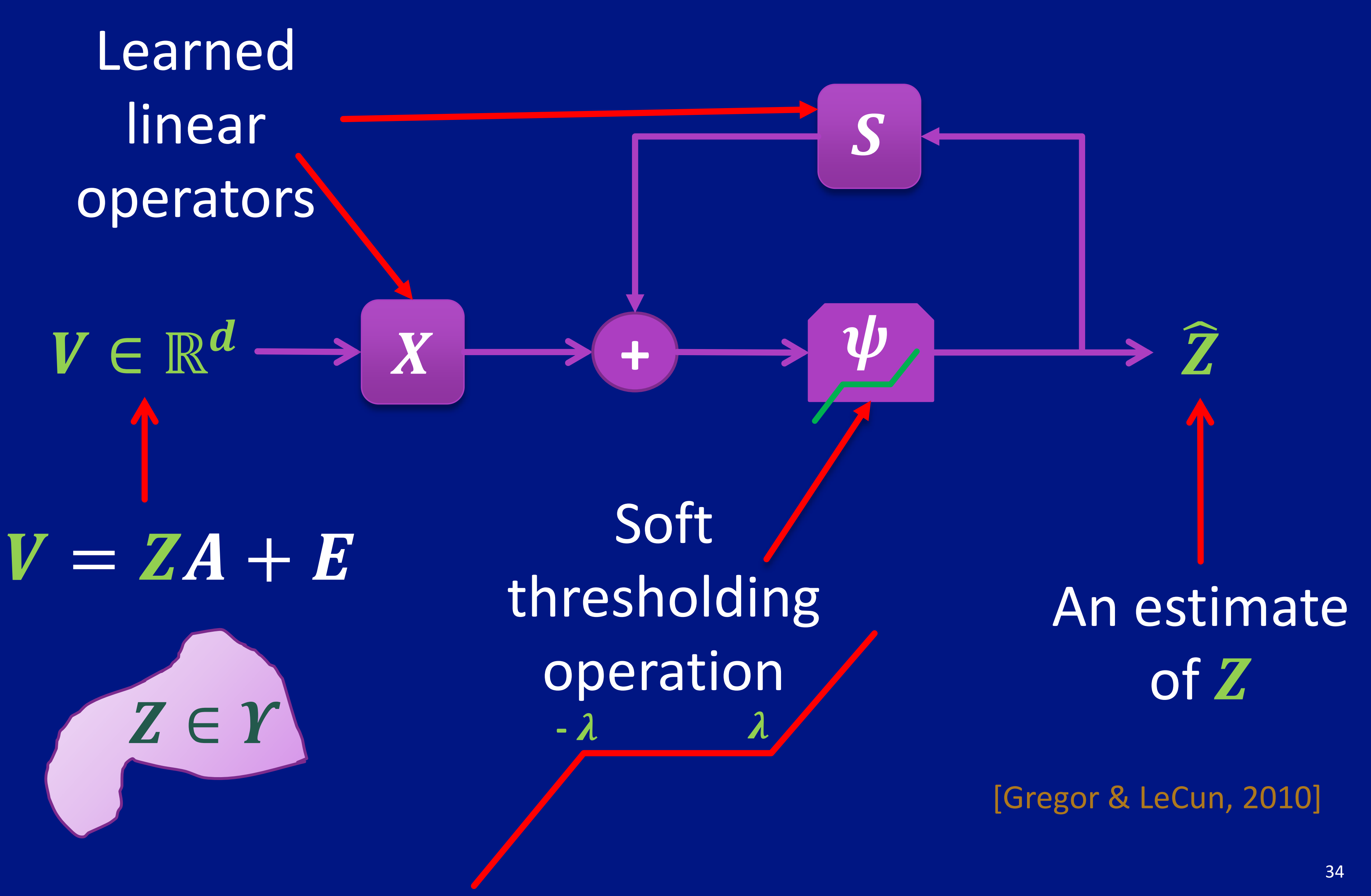
[Daubechies, Defrise & Mol, 2004], [Beck & Teboulle, 2009]

ISTA CONVERGENCE

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



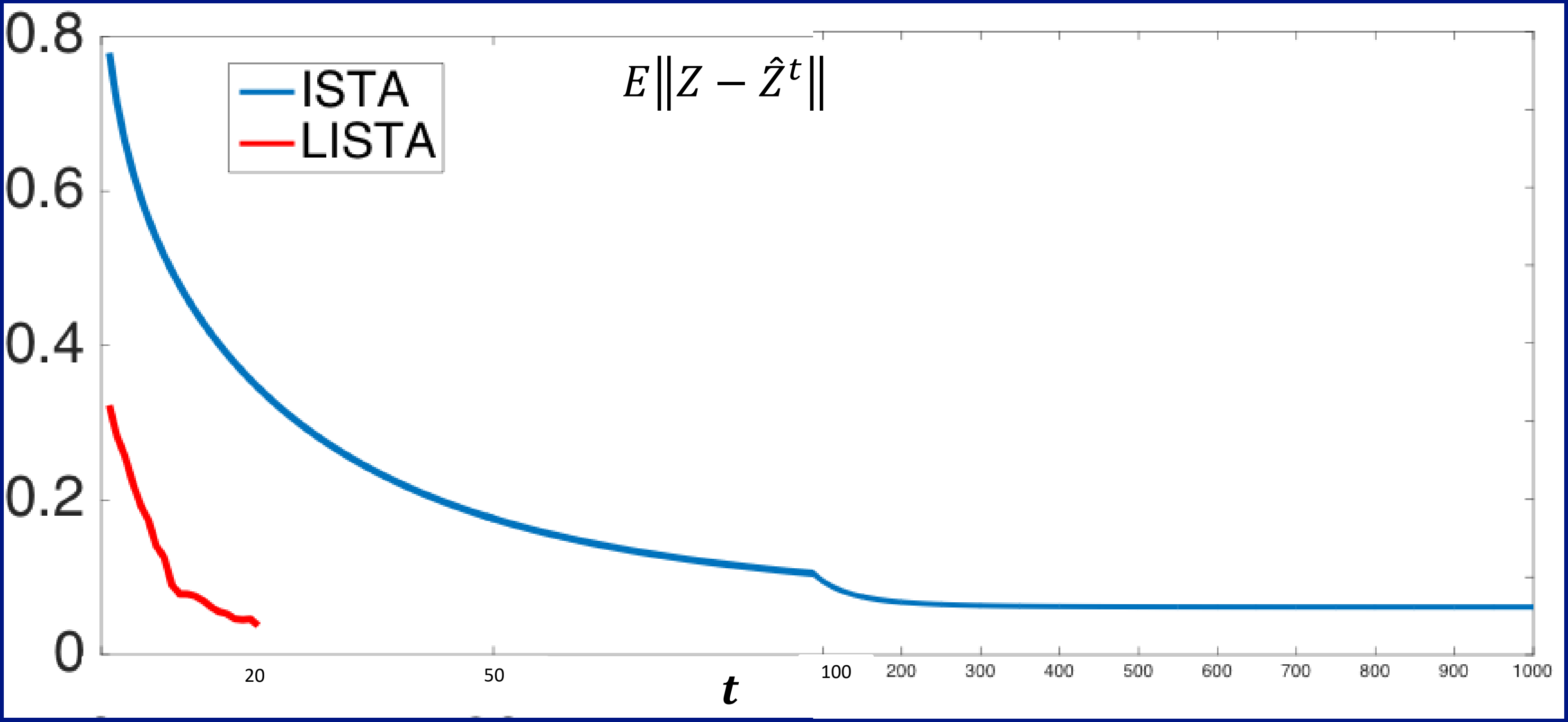
LEARNED ISTA (LISTA)



[Gregor & LeCun, 2010]

LISTA CONVERGENCE

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



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

PROJECTED GRADIENT DESCENT (PGD)

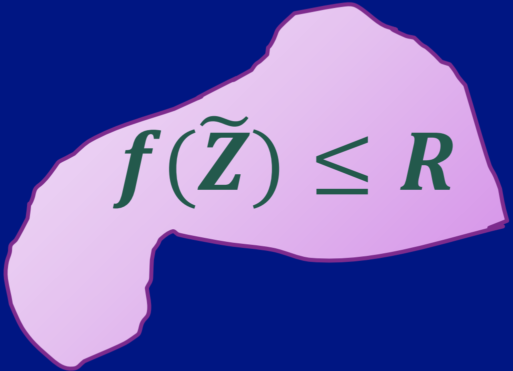
$$\begin{tikzcd}
 & & & \boldsymbol{I - \mu A A^T} \arrow[dl] & \\
\boldsymbol{V} \in \mathbb{R}^d \arrow[r] & \boldsymbol{\mu A^T} \arrow[r] & + \arrow[r] & \boldsymbol{\psi} \arrow[r] \arrow[u] & \boldsymbol{\widehat{Z}}
\end{tikzcd}$$

μ is the step size

$\boldsymbol{V = ZA + E}$

$\boldsymbol{f(Z) \leq R}$

$\boldsymbol{\psi}$ projects onto the set $\boldsymbol{\Upsilon}$



Estimate of $\boldsymbol{Z}$. Aim at solving

$$\boldsymbol{\min_{\widetilde{Z}} \left\| V - \widetilde{Z} A \right\|}$$

$$\boldsymbol{s.t.\ f(\widetilde{Z}) \leq R}$$

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 x, $A \in \mathbb{R}^{d\times m}$ a random Gaussian matrix and $V = ZA + E$. 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 $\rho = \sup_{U,W \in C_f(Z) \cap \mathcal{B}^d} U\left(I - \mu A A^T\right) W^T$
and $\kappa_f = 1$ if f is convex and $\kappa_f = 2$ otherwise. [Oymak, Recht & Soltanolkotabi, 2016].

PGD CONVERGENCE RATE

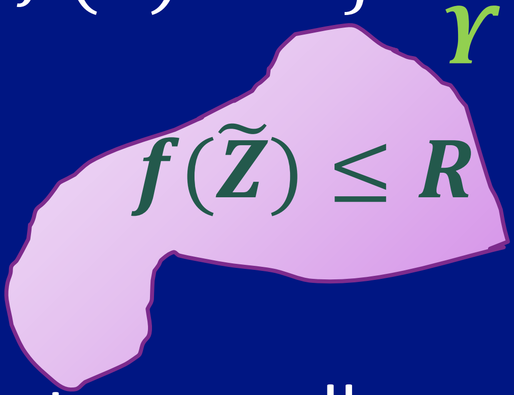
- $\rho = \sup_{U,W \in C_f(Z) \cap \mathcal{B}^d} U\left(I - \mu AA^T\right)W^T$ is the convergence rate of PGD.
- Let ω be the Gaussian mean width of $C_f(Z) \cap \mathcal{B}^d$.
- If $\mu = \frac{1}{\left(\sqrt{m}+\sqrt{d}\right)^2} \simeq \frac{1}{d}$ then $\rho = 1 - O\left(\frac{\sqrt{m}-\omega}{m+d}\right)$.
- 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

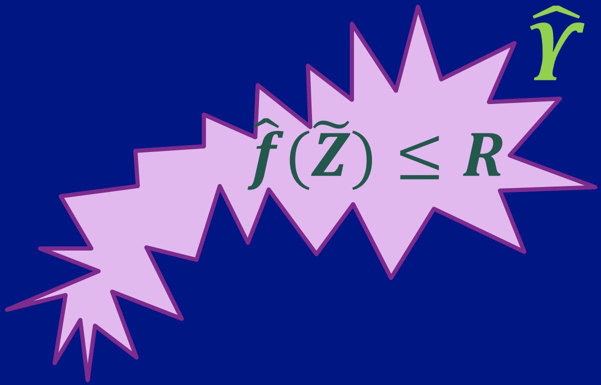
- PGD iterations projects onto $\Upsilon = \{\tilde{Z}: f(\tilde{Z}) \le R\}$.
- Smaller $\Upsilon \Rightarrow$ Smaller ω.

$\Rightarrow$Faster convergence as

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

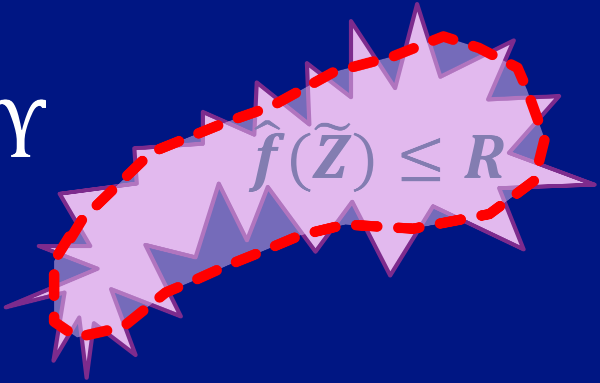


- Let us assume that our signal belongs to a smaller set $\hat{\Upsilon} = \{\tilde{Z}: \hat{f}(\tilde{Z}) \le R\}$ with $\hat{\omega} \ll \omega$.
- Ideally, we would like to project onto $\hat{\Upsilon}$ instead of Υ.
- This will lead to faster convergence.
- What if such a projection is not feasible?



INACCURATE PROJECTION

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



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

INACCURATE PGD (IPGD)

$$\begin{tikzcd}
 & & & (I-\mu AA^T)P \arrow[dl] & \\
V \in \mathbb{R}^d \arrow[r] & \mu A^T P \arrow[r] & + \arrow[r] & \psi \arrow[r] \arrow[u] & \hat{Z} \arrow[ul, bend right]
\end{tikzcd}$$

μ is the step size

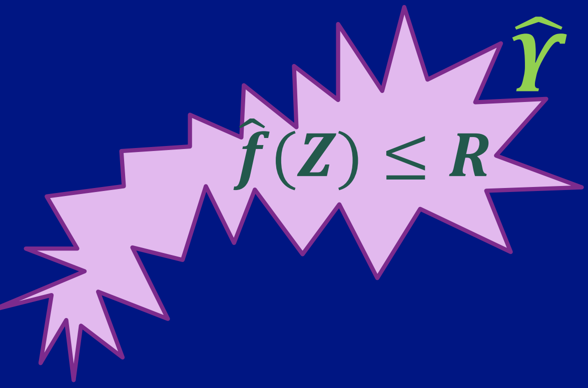
$V = ZA + E$

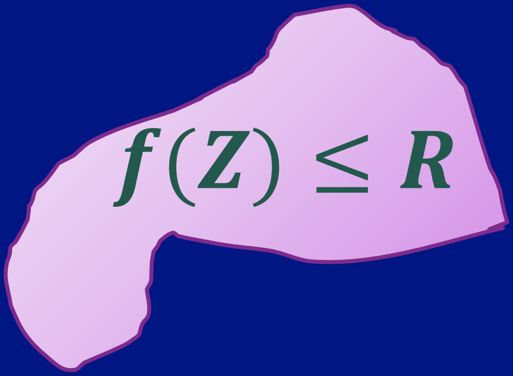
ψ projects onto the set Υ

Estimate of Z.

Aim at solving

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





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 $V = ZA + E$. 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 $\rho_p = \sup_{U,W \in C_f(Z) \cap \mathcal{B}^d} UP\left(I - \mu A A^T\right) P W^T$

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:

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

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

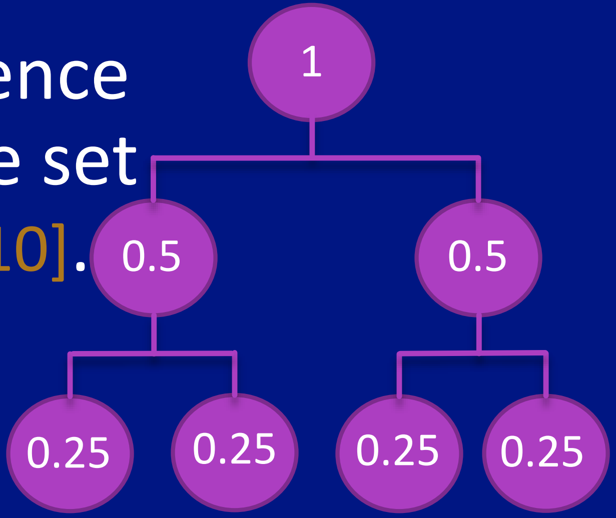
(a) ϵ is negligible compared to ρ_P

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

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

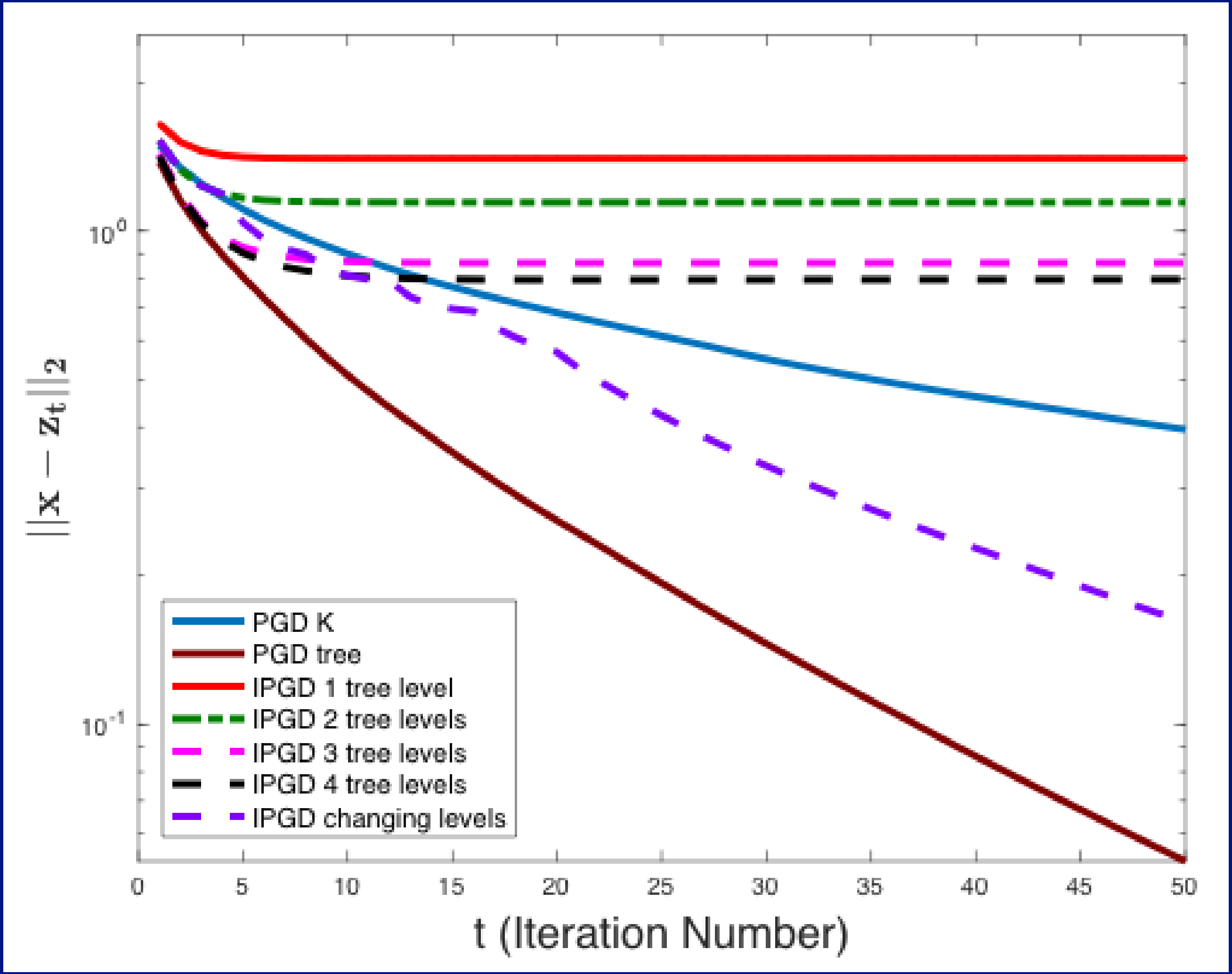
MODEL BASED COMPRESSED SENSING

- $\widehat{\Upsilon}$ 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

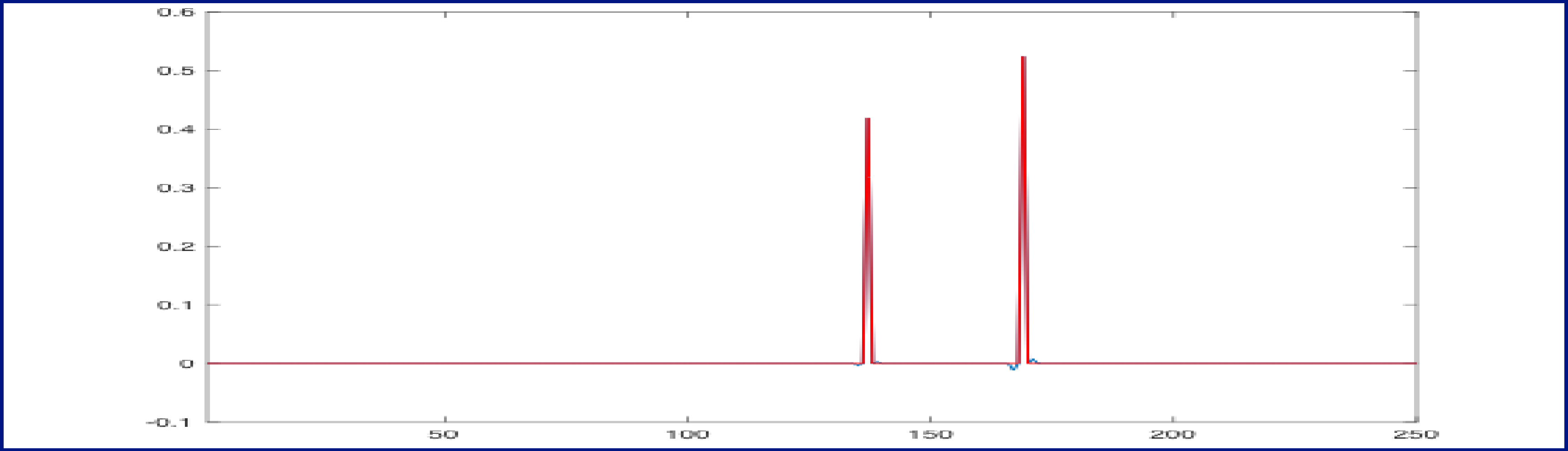


Non-zeros picked entries has zero mean random Gaussian distribution with variance:

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

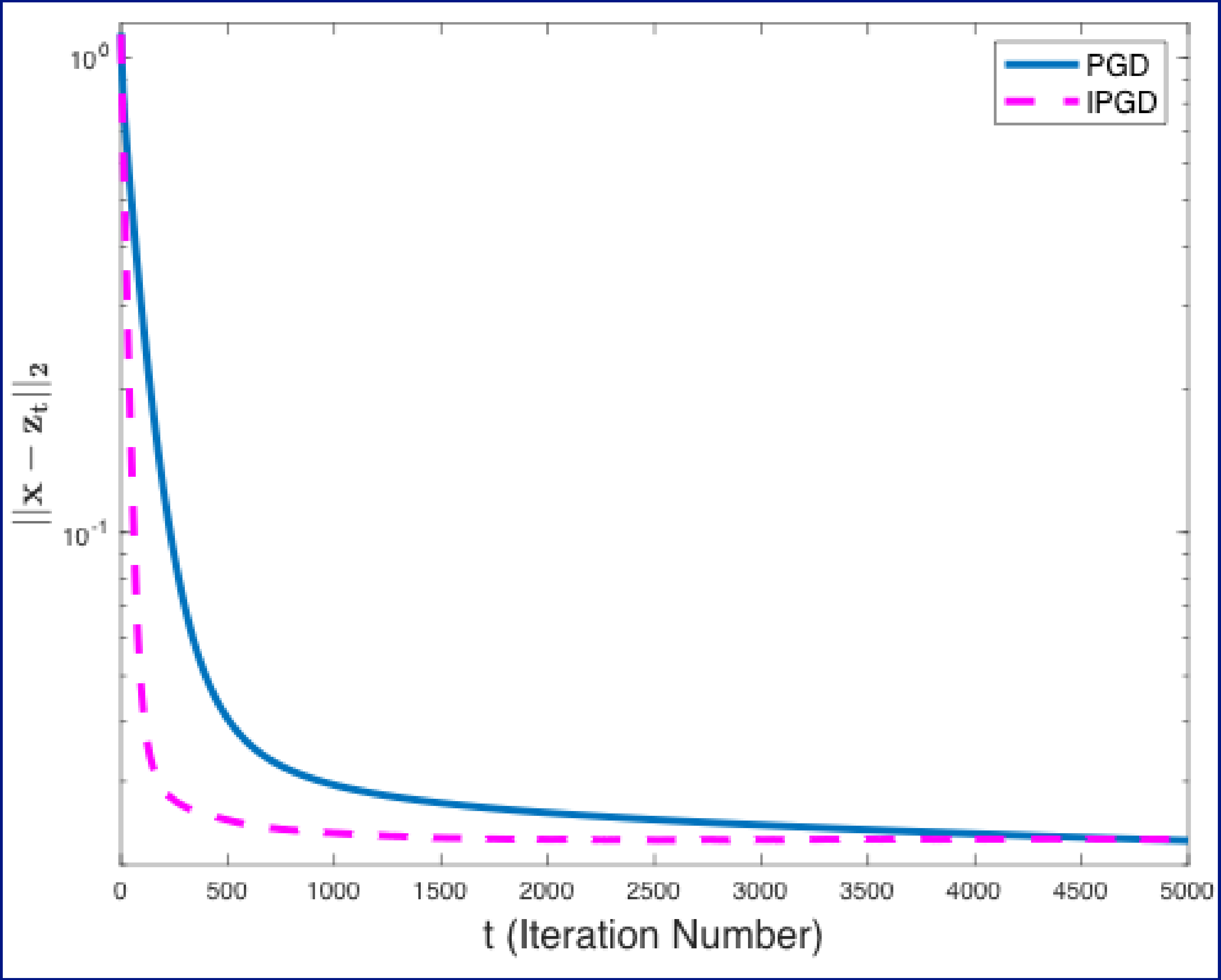
SPECTRAL COMPRESSED SENSING

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



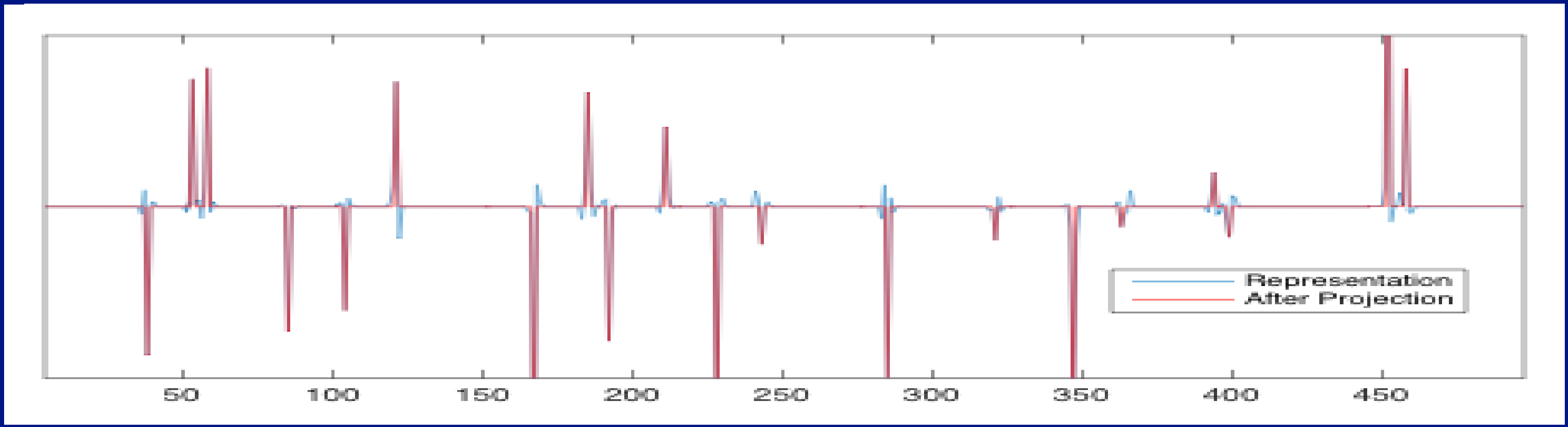
- We set P to be a pooling-like operation that keeps in each window of size 3 only the largest value.

SPECTRAL COMPRESSED SENSING



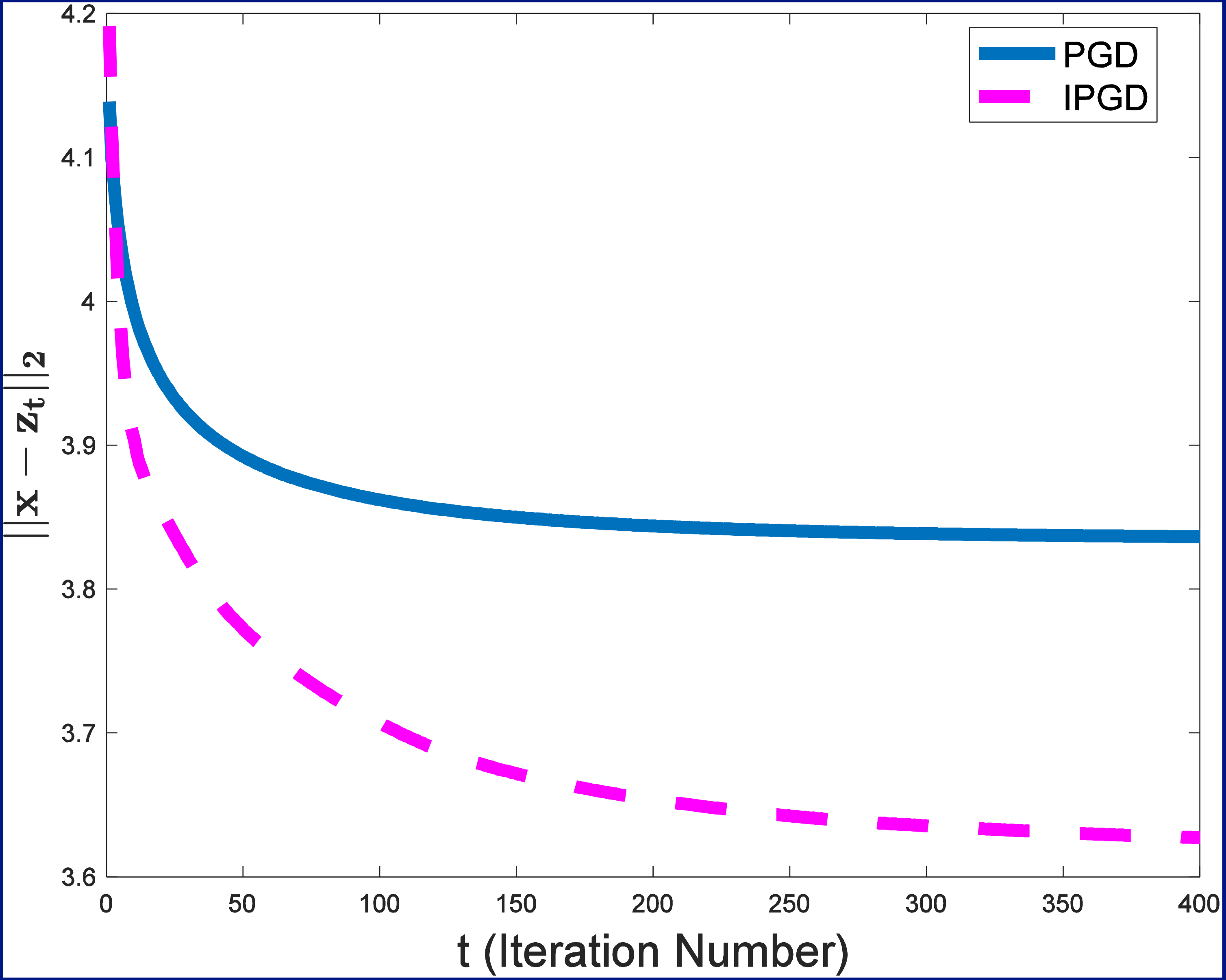
SPECTRAL COMPRESSED SENSING

- $\widehat{\Upsilon}$ 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.

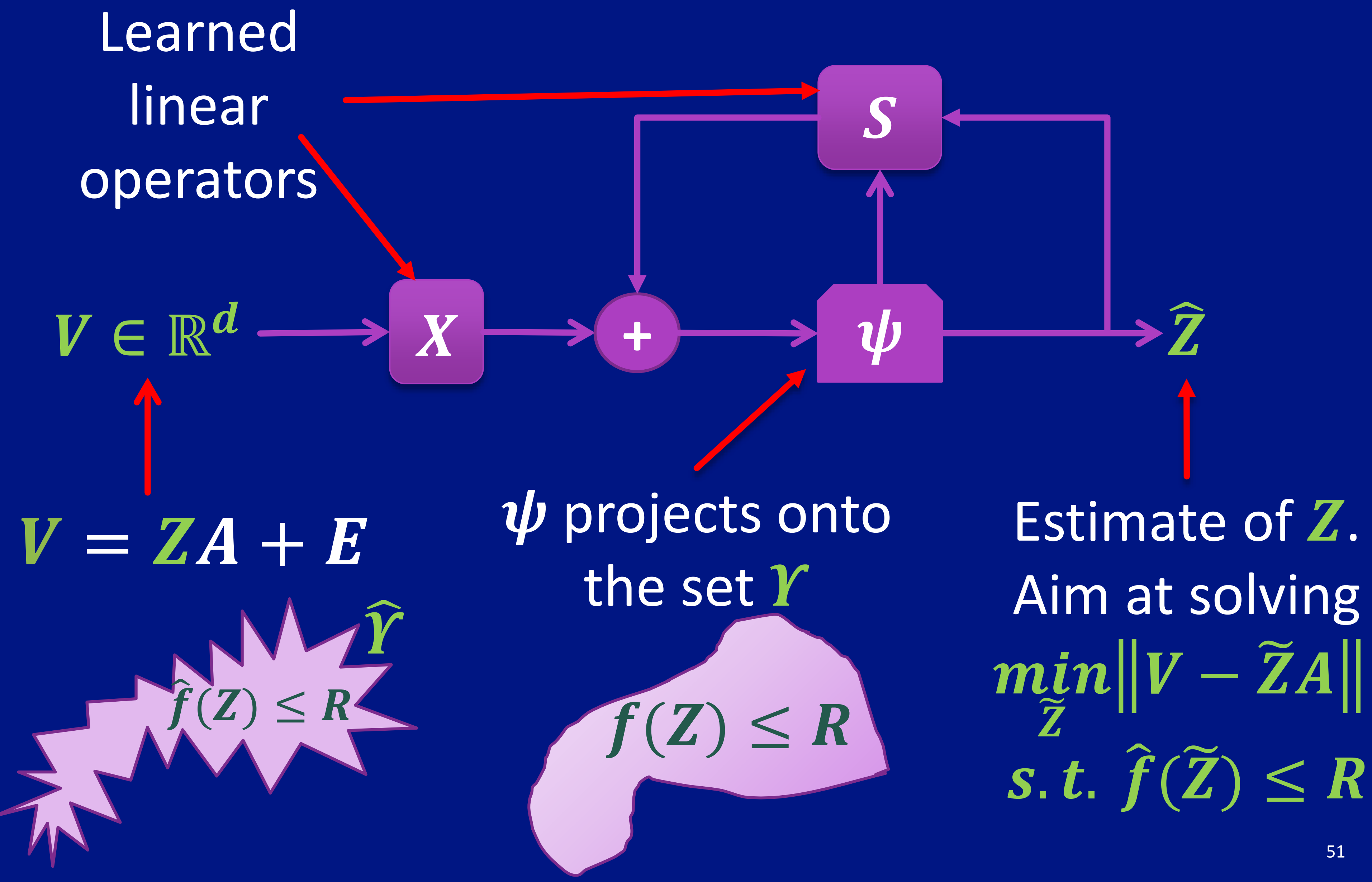
SPECTRAL COMPRESSED SENSING



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 $(I - \mu AA^T)P$ and $\mu A^T P$ with two learned matrices S and X 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



$V = ZA + E$

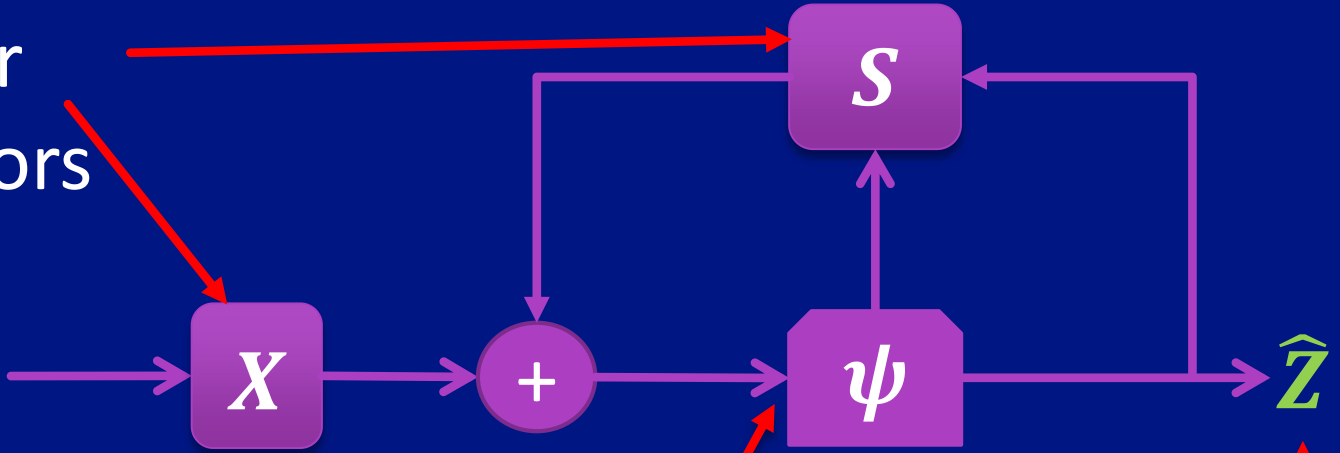
ψ projects onto the set Υ

Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \| V - \tilde{Z}A \|$$
$$s.t.\ \hat{f}(\tilde{Z}) \le R$$

LISTA

Learned linear operators



$V \in \mathbb{R}^d$

$V = ZA + E$

$\hat{Y}$

$\hat{f}(Z) \leq R$

ψ is a proximal mapping.

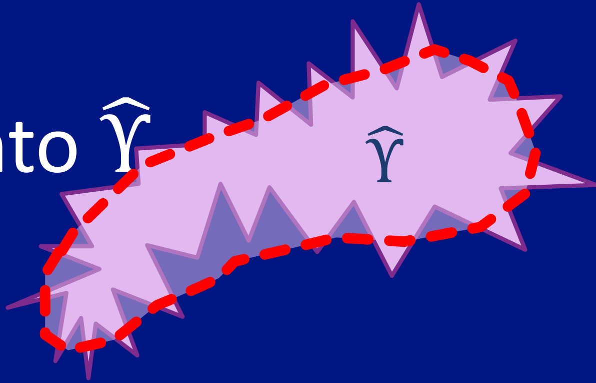
$$\psi(U) = \underset{\tilde{Z} \in \mathbb{R}^d}{\mathbf{argmin}} \left\| U - \tilde{Z} \right\| + \lambda f(\tilde{Z})$$

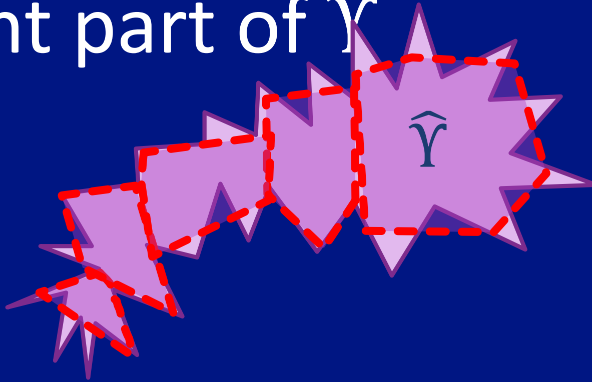
Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \left\| V - \tilde{Z}A \right\| + \lambda \hat{f}(\tilde{Z})$$

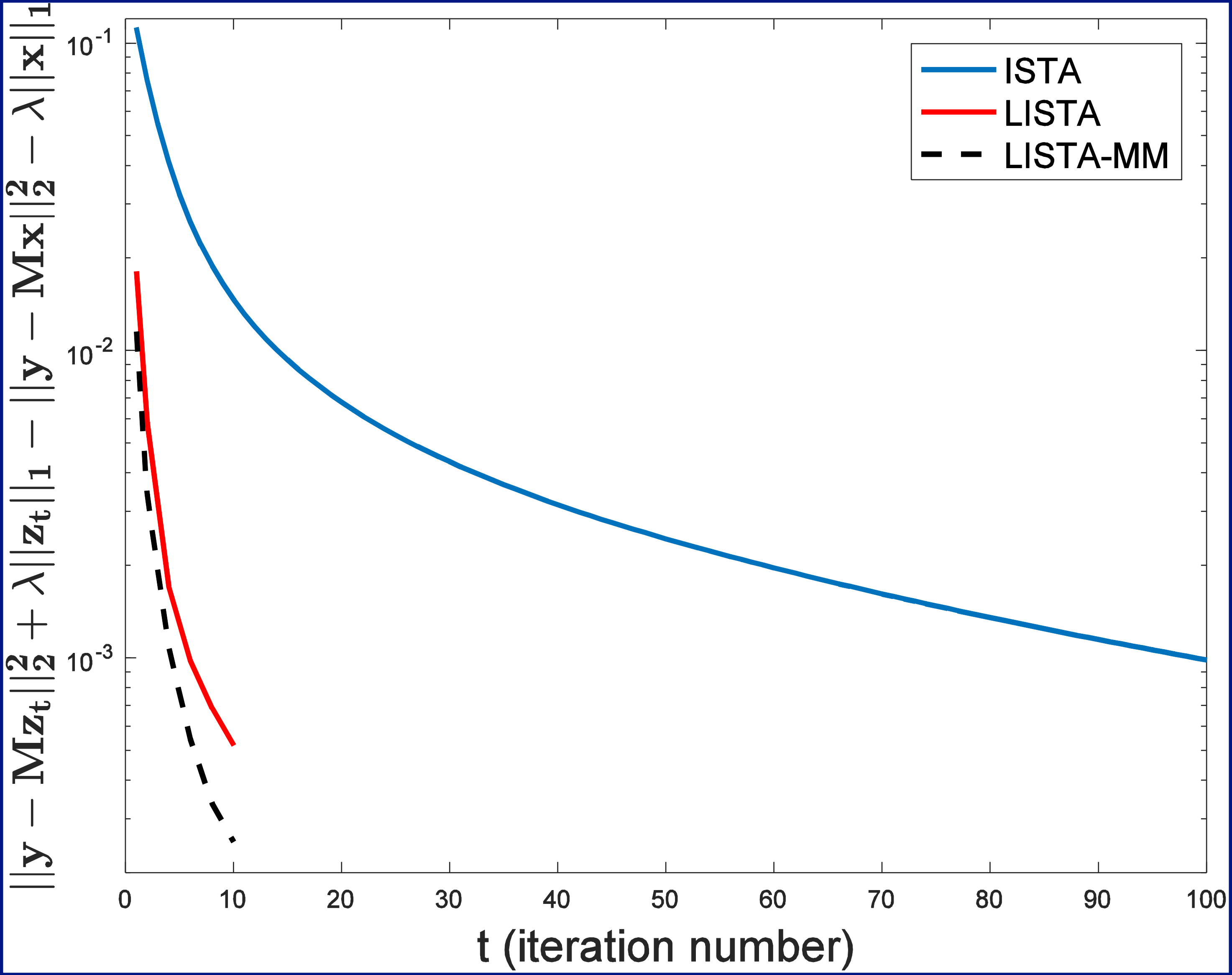
LISTA MIXTURE MODEL

- Approximation of the projection onto $\hat{\Upsilon}$ 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 $\hat{\Upsilon}$
- Training multiple LISTA networks accelerate the convergence further.





LISTA MIXTURE MODEL



RELATED WORKS

- In [Bruna et al. 2016] 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).
- All these works consider only the sparsity case

Take Home Message

DNN may solve optimization problems

Generalization error depends on the DNN input margin

ACKNOWLEDGEMENTS

Guillermo Sapiro
Duke University

Yonina C. Eldar
Technion

Alex M. Bronstein
Technion

Miguel Rodrigues
UCL

Jure Sokolic
UCL

QUESTIONS?

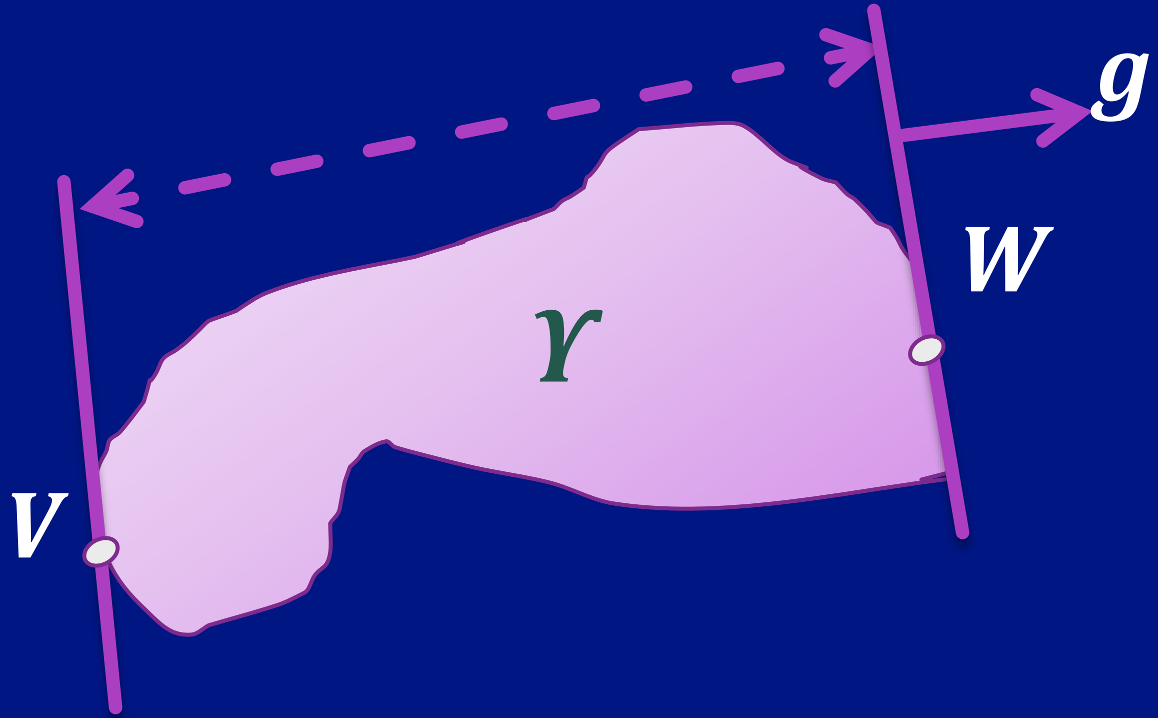
WEB.ENG.TAU.AC.IL/~RAJA

GAUSSIAN MEAN WIDTH

- Gaussian mean width:

$$\boldsymbol{\omega(\Upsilon) = E \sup_{V,W\in\Upsilon} \langle V - W, g\rangle, \qquad g \sim N(0, I).}$$

The width of the set $\boldsymbol{\Upsilon}$ in the direction of $\boldsymbol{g}$:



MEASURE FOR LOW DIMENSIONALITY

- Gaussian mean width:

$$\boldsymbol{\omega(\Upsilon) = E \sup_{V,W \in \Upsilon} \langle V - W, g \rangle, \qquad g \sim N(0, I).}$$

- $\boldsymbol{\omega^2(\Upsilon)}$ is a measure for the dimensionality of the data.

- Examples:

If $\Upsilon \subset \mathbb{B}^d$ is a Gaussian Mixture Model with k Gaussians then

$$\boldsymbol{\omega^2(\Upsilon) = O(k)}$$

If $\Upsilon \subset \mathbb{B}^d$ is a data with k-sparse representations then

$$\boldsymbol{\omega^2(\Upsilon) = O(k \log d)}$$