Stat 212b: Topics in Deep Learning
Lecture 7

Joan Bruna
UC Berkeley

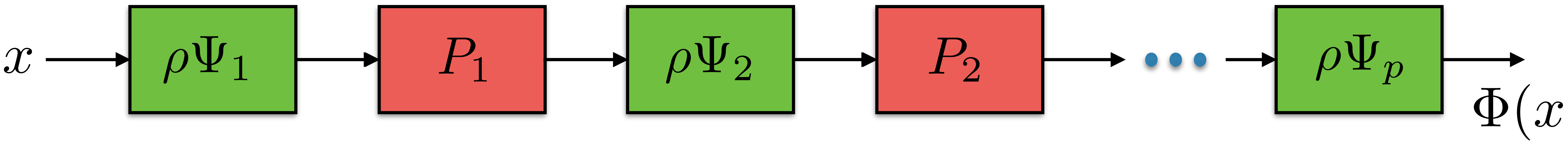
Objectives

- Properties of CNN representations (cont.)
  - Stability
  - Redundancy
  - Invertibility

- Proximal methods and Deep Neural Networks
  - Task Driven Dictionary Learning
  - LISTA

- Random forests and Deep Neural Networks.

Review:Convolutional Neural Networks

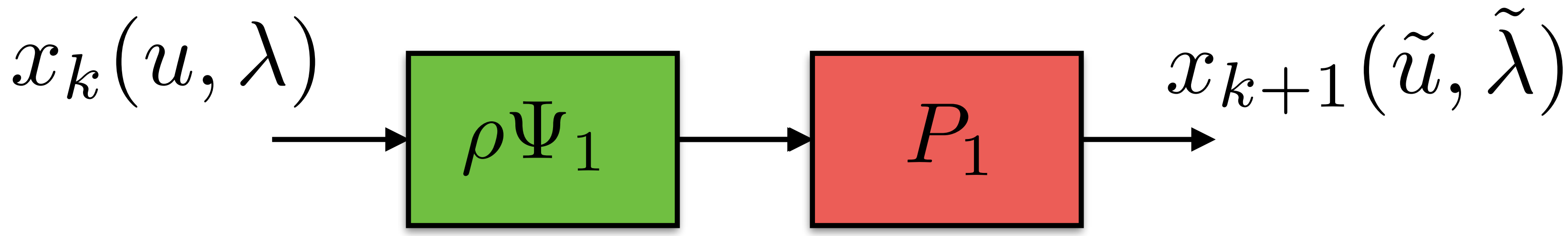


$$\Phi(x) = \rho(\rho(P_1(\rho(x * \Psi_1)) * \Psi_2)..)$$

- Architectures vary in terms of
  - Number p of layers (from 2 to >100).
  - Size of the tensors (typically [3-7 x 3-7 x 16-256])
  - Presence/absence and type of pooling operator.
    - Recent models tend to avoid non-adaptive pooling.

Review: Geometric Intuition

- We can start by analyzing a chunk of the form



- Let us assume that pooling is an average (non-adaptive).
- Consider a thresholding nonlinearity: $\rho(x) = \max(0, x - t)$
- And let us forget (for now) about the convolutional aspect.

- *Redundant linear transform+nonlinearity*: scatter the space into linear chunks
- *Pooling*: stitch together chunks that belong together.

CNNs and near-diagonalisation

- Given $x(u, \lambda)$ intermediate layer representation, and a generic variability model $\{\varphi_{\tau, f(x)} x\}_{\tau}$, how to retransform x?

CNNs and near-diagonalisation

- Given $x(u, \lambda)$ intermediate layer representation, and a generic variability model $\{\varphi_{\tau,f(x)} x\}_\tau$, how to retransform x?

- We find linear measurements that factorize variability model into small eigenspaces:

$$\langle \varphi_{\tau,f} x, T_u w_k \rangle = \langle x, \varphi^*_{\tau,f} T_u w_k \rangle \approx \sum_{|v| \leq \delta, k'} \alpha_{v,k'}(\tau, f) \langle x, T_{u+v} w_{k'} \rangle$$

CNNs and near-diagonalisation

- Given $x(u, \lambda)$ intermediate layer representation, and a generic variability model $\{\varphi_{\tau, f(x)} x\}_\tau$, how to retransform x?

- We find linear measurements that factorize variability model into small eigenspaces:

$$\langle \varphi_{\tau,f} x, T_u w_k \rangle = \langle x, \varphi^*_{\tau,f} T_u w_k \rangle \approx \sum_{|v| \leq \delta, k'} \alpha_{v,k'}(\tau, f) \langle x, T_{u+v} w_{k'} \rangle$$

- Moreover, in order for non-linearities to be discriminative, we want $\{\langle x, T_u w_k \rangle\}_{u,k}$ sparse.

CNN and near-diagonalisation

- Role of each layer: progressive linearization of intra-class variability.

CNN and near-diagonalisation

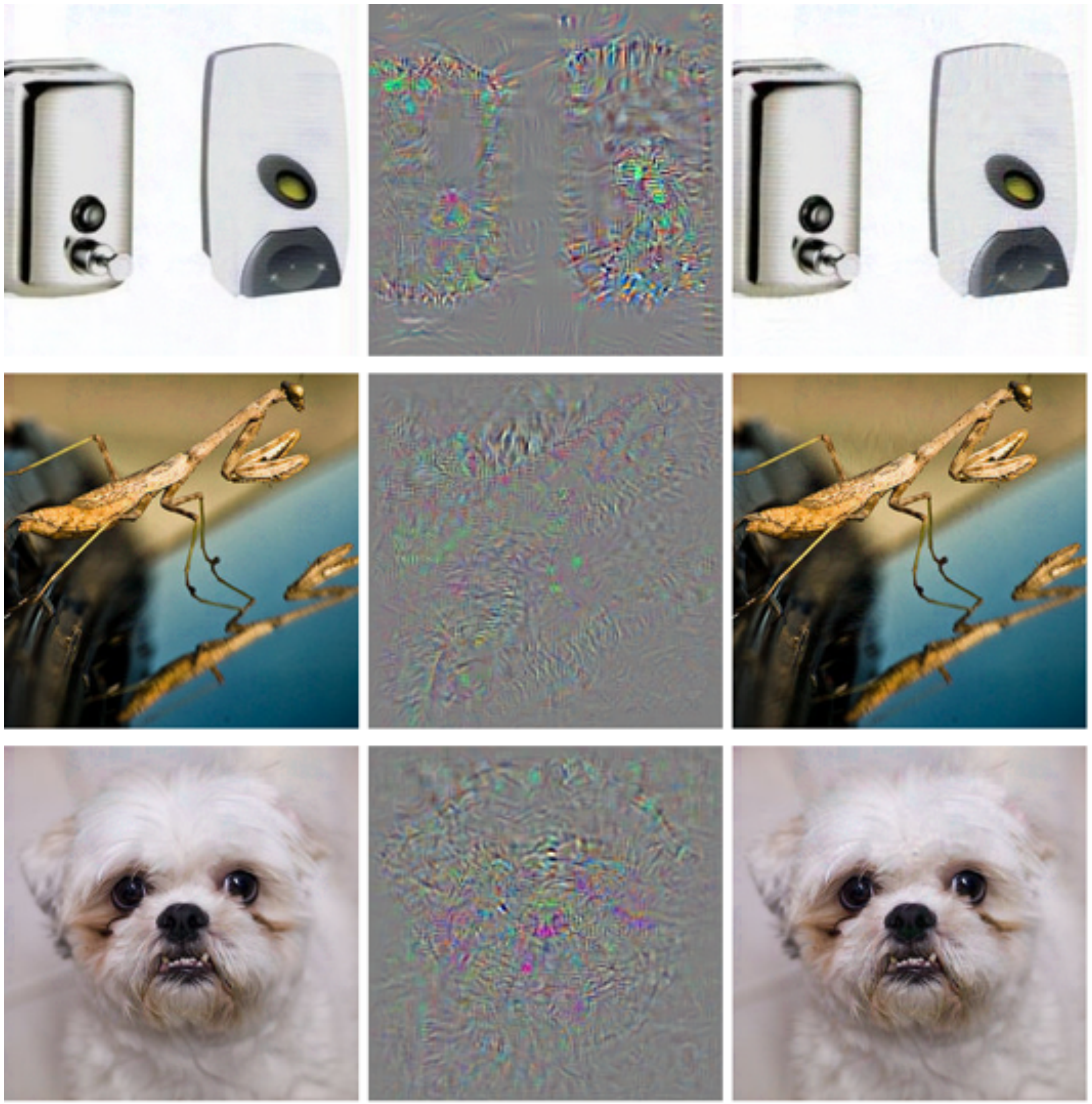
- Role of each layer: progressive linearization of intra-class variability.

- Filters play a dual function:
  - learn invariants that perform averaging along existing approximate orbits (learnt pooling).
  - map variability to new parallel approximate orbits for the next layer.
  - ensure signals are sparse along orbits.

Review: Instabilities of Deep Networks

[Szegedy *et al*, ICLR'14]

x $\tilde{x}$



correctly classified

classified as ostrich

Alex Krizhevsky's Imagenet
8 layer Deep ConvNet

$$\|x - \tilde{x}\| < 0.01\|x\|$$

Review: Instabilities of Deep Networks

- Additive Stability is not enforced:

$$\|\Phi_i(x) - \Phi_i(x')\| \leq \|W_i(x - x')\| \leq \|W_i\| \, \|x - x'\|$$

Layer	Size	$\|W_i\|$
Conv. 1	$3 \times 11 \times 11 \times 96$	2.75
Conv. 2	$96 \times 5 \times 5 \times 256$	10
Conv. 3	$256 \times 3 \times 3 \times 384$	7
Conv. 4	$384 \times 3 \times 3 \times 384$	7.5
Conv. 5	$384 \times 3 \times 3 \times 256$	11
FC. 1	9216×4096	3.12
FC. 2	4096×4096	4
FC. 3	4096×1000	4

Review: Instabilities of Deep Networks

- Additive Stability is not enforced:

$$\|\Phi_i(x) - \Phi_i(x')\| \leq \|W_i(x - x')\| \leq \|W_i\| \, \|x - x'\|$$

Layer	Size	$\|W_i\|$
Conv. 1	$3 \times 11 \times 11 \times 96$	2.75
Conv. 2	$96 \times 5 \times 5 \times 256$	10
Conv. 3	$256 \times 3 \times 3 \times 384$	7
Conv. 4	$384 \times 3 \times 3 \times 384$	7.5
Conv. 5	$384 \times 3 \times 3 \times 256$	11
FC. 1	9216×4096	3.12
FC. 2	4096×4096	4
FC. 3	4096×1000	4

- These *adversarial* examples are found by explicitly fooling the network:

$$\min \|x - \tilde{x}\|^2 \quad s.t. \quad p(y \mid \Phi(\tilde{x})) \perp p(y \mid \Phi(x))$$

- They are robust to different parametrization of $\Phi(x)$ and to different hyper-parameters.

Review: Instabilities of Deep Networks

- However, these examples do not occur in practice.

Review: Instabilities of Deep Networks

- However, these examples do not occur in practice.
- A discriminative model does not *care* about robustness with respect to the input distribution:

Regret is $Pr(\hat{f}(\Phi(x)) \neq f(x))$ (classification)
or $E(\|f(x) - \hat{f}(\Phi(x))\|^2)$ (regression)

It is defined through an input distribution $(x, y) \sim \mathcal{X}$ with density $h(x, y)$.

Review: Instabilities of Deep Networks

- CNNs do not assume (rightfully) an input distribution stable to additive noise:

$$|h(x,y) - h(x+n,y)| \text{ can be large even if } \|n\| \text{ small}$$

Review: Instabilities of Deep Networks

- CNNs do not assume (rightfully) an input distribution stable to additive noise:

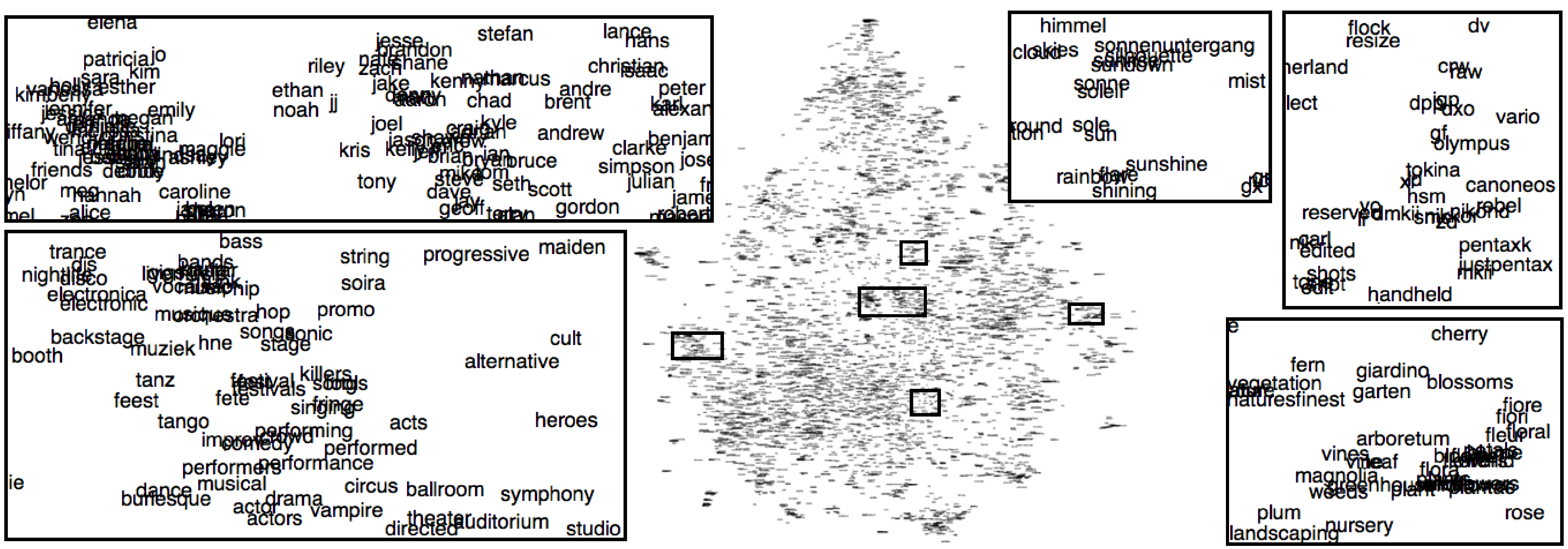
$$|h(x,y) - h(x+n,y)| \text{ can be large even if } \|n\| \text{ small}$$

- However, they DO assume an input distribution stable to geometric noise:

$$|h(x,y) - h(\varphi_\tau(x),y)| \text{ small if } \|\tau\| \text{ small}$$

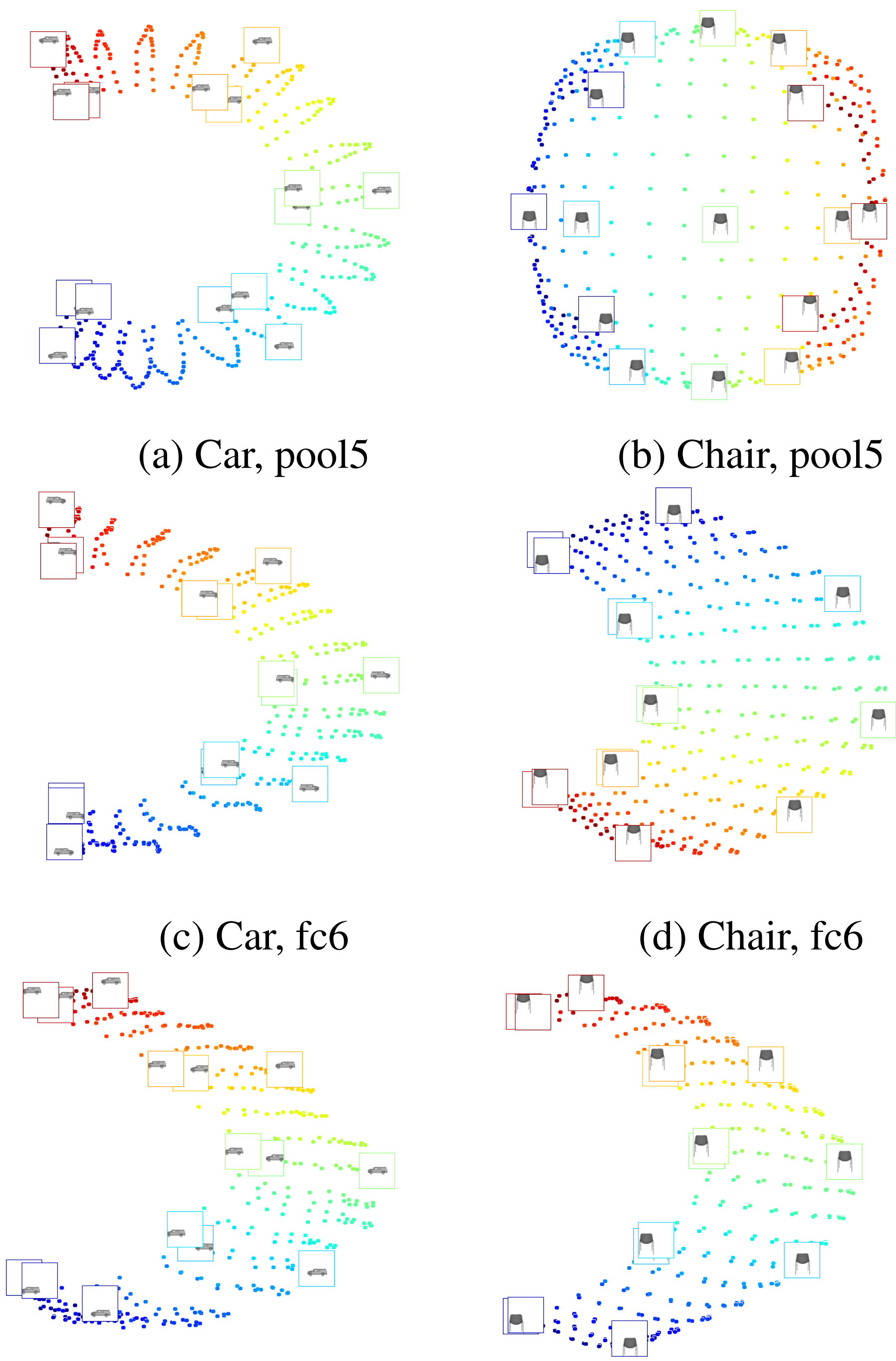
Stability: Transfer learning

- a CNN trained on a (large enough) dataset generalizes to other visual tasks:



"Learning visual features from Large Weakly supervised Data", [Joulin et al, '15]

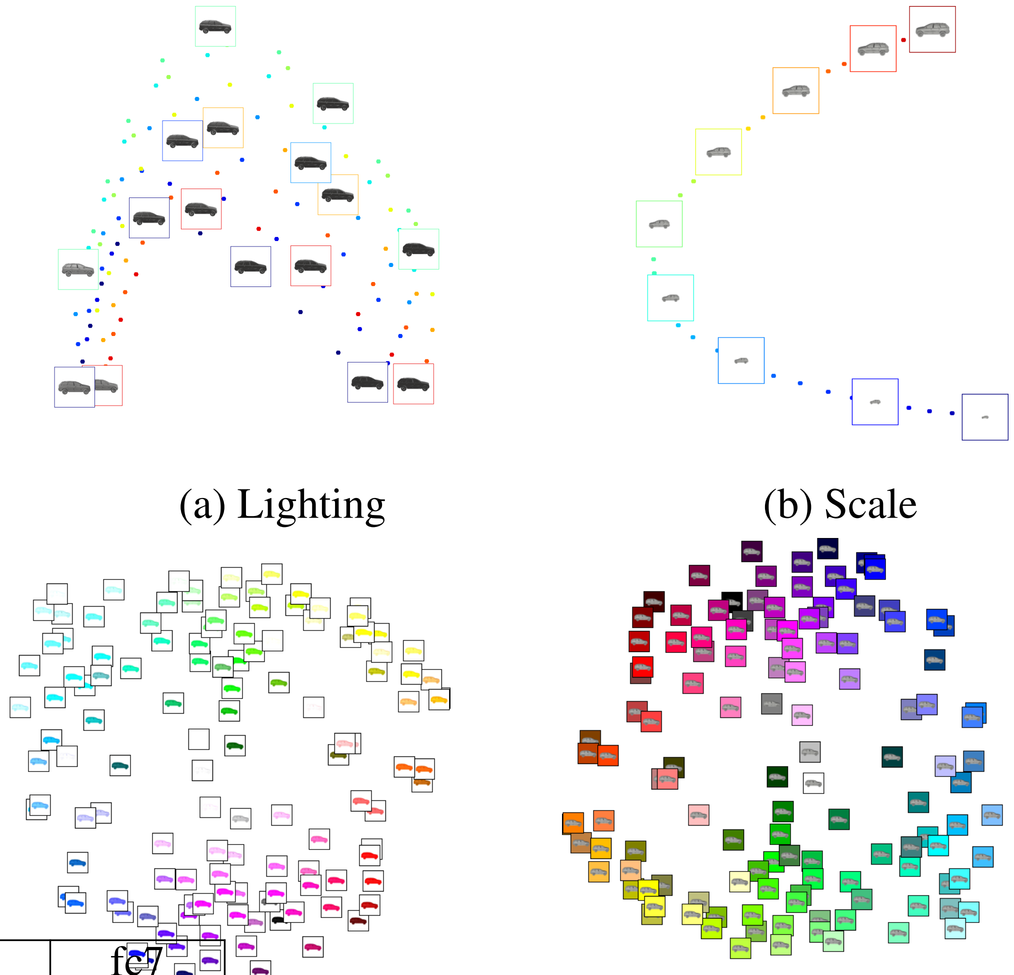
Review: Invariance and Covariance



(a) Car, pool5 (b) Chair, pool5

(c) Car, fc6 (d) Chair, fc6

(e) Car, fc7 (f) Chair, fc7



(a) Lighting (b) Scale

(c) Object color (d) Background color

		pool5	fc6	fc7
Viewpoint	Places	26.8 % 8.5	21.4 % 7.0	17.8 % 5.9
	AlexNet	26.4 % 8.3	19.4 % 7.2	15.6 % 6.0
	VGG	21.2 % 10.0	16.4 % 7.7	12.3 % 6.2
Style	Places	26.8 % 136.3	39.1 % 105.5	49.4 % 54.6
	AlexNet	28.2 % 121.1	40.3 % 125.5	49.4 % 96.7
	VGG	26.4 % 181.9	44.3 % 136.3	56.2 % 94.2
Δ^L	Places	46.8 %	39.5 %	32.9 %
	AlexNet	45.0 %	40.3 %	35.0 %
	VGG	52.4 %	39.3 %	31.5 %

[Aubry & Rusell '15]

Invariance, Linearization and Geodesics

- We related stability with the ability to linearize deformations:

$$\tau \mapsto \Phi(\varphi_\tau x) \text{ Lipschitz} \Rightarrow$$
$$\Phi(\varphi_\tau x) = \Phi(x) + D(\Phi \circ \varphi.(x))\tau + O(\|\tau\|)$$

Invariance, Linearization and Geodesics

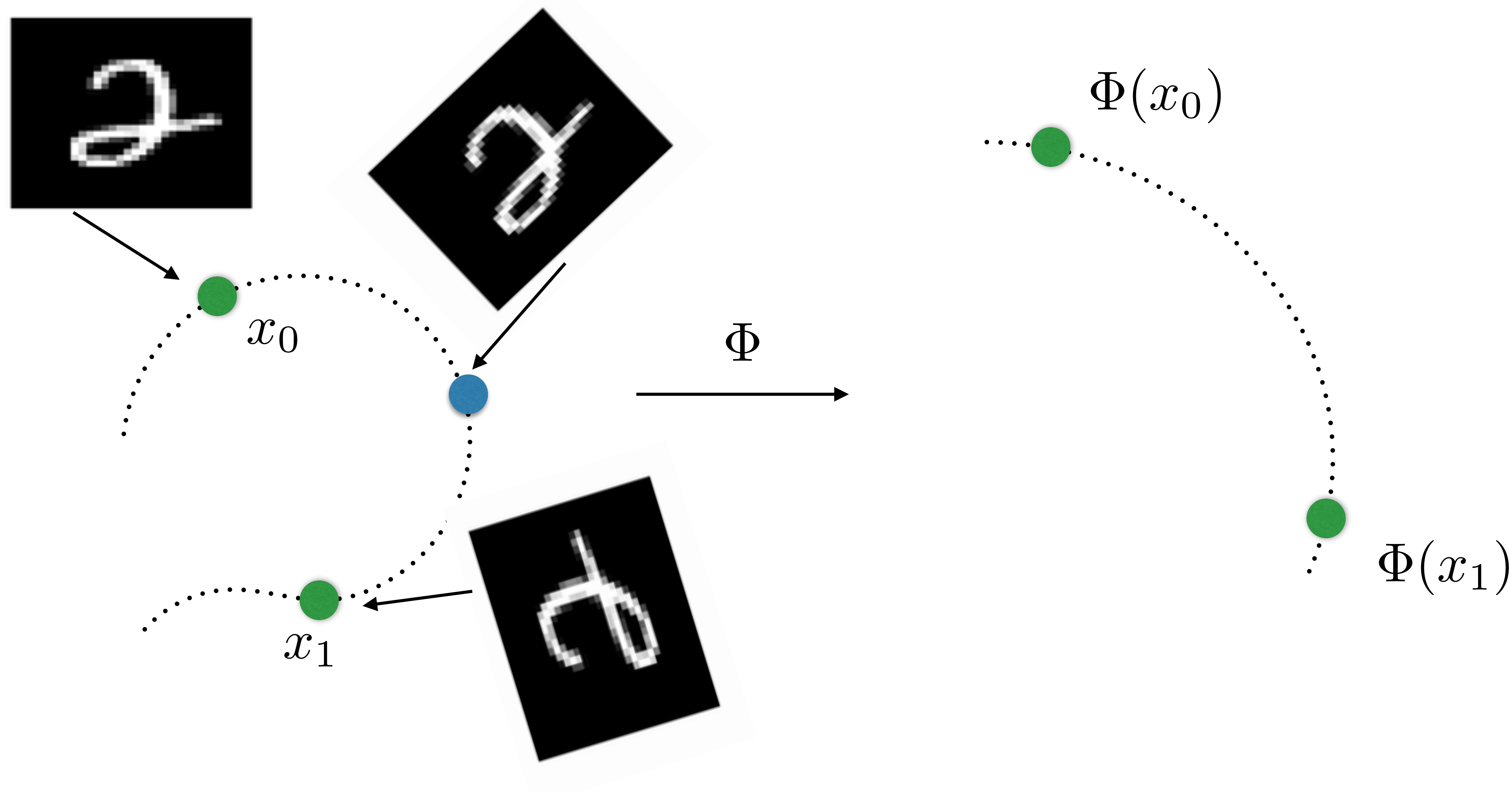
- We related stability with the ability to linearize deformations:

$$\tau \mapsto \Phi(\varphi_\tau x) \text{ Lipschitz} \Rightarrow$$
$$\Phi(\varphi_\tau x) = \Phi(x) + D(\Phi \circ \varphi_{.}(x))\tau + O(\|\tau\|)$$

- One can test this property over learnt representations by inspecting geodesics.
  - They become linear paths in feature space under the metric
  $$d(x, x') = \|\Phi(x) - \Phi(x')\|$$
  - [Bengio et al. '11], [Goroshin et al'15], [Henaff et al '16]

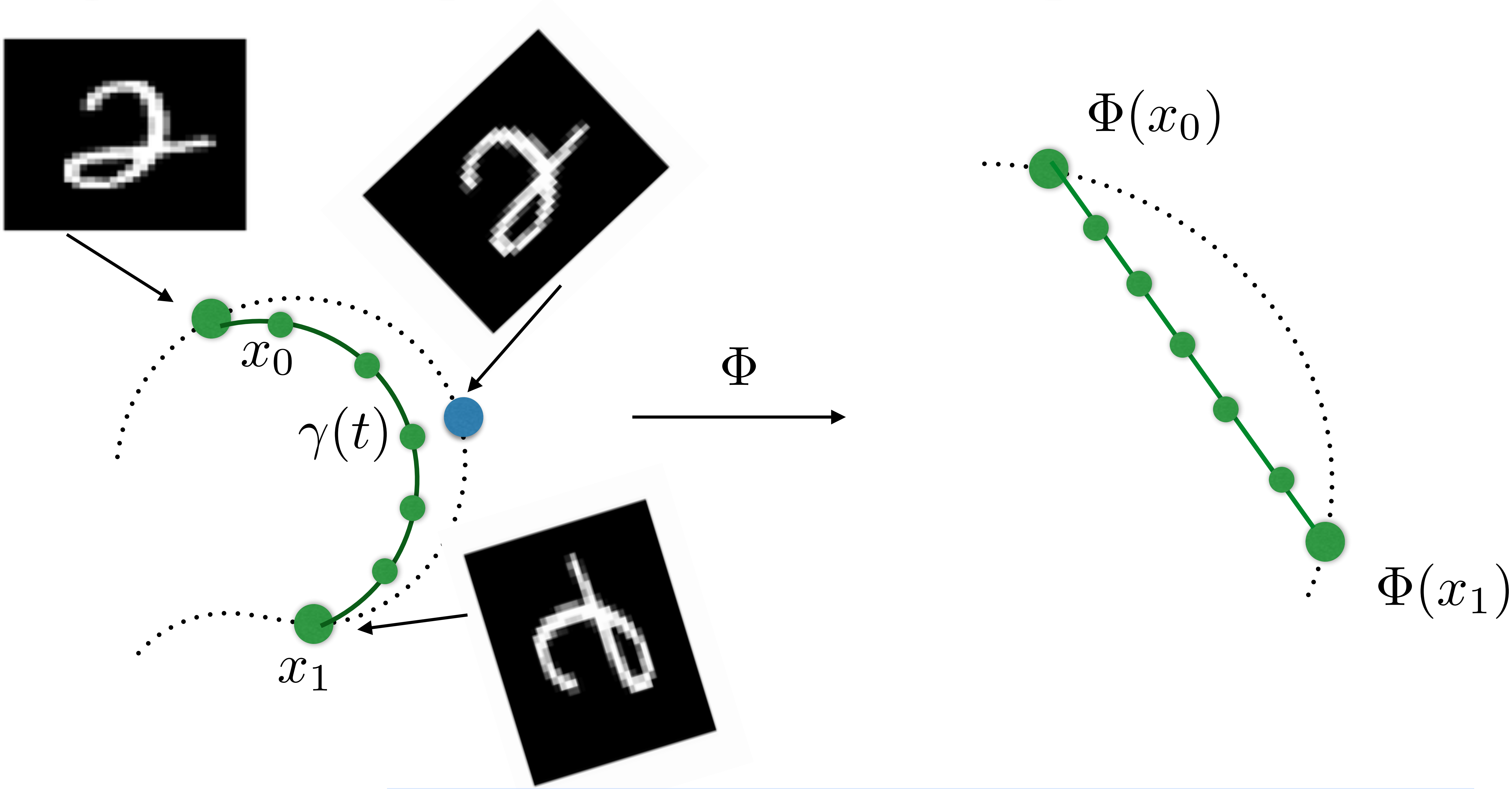
Invariance, Linearization and Geodesics

- Algorithm from [Henaff & Simoncelli '16]:



Invariance, Linearization and Geodesics

- Algorithm from [Henaff & Simoncelli '16]:

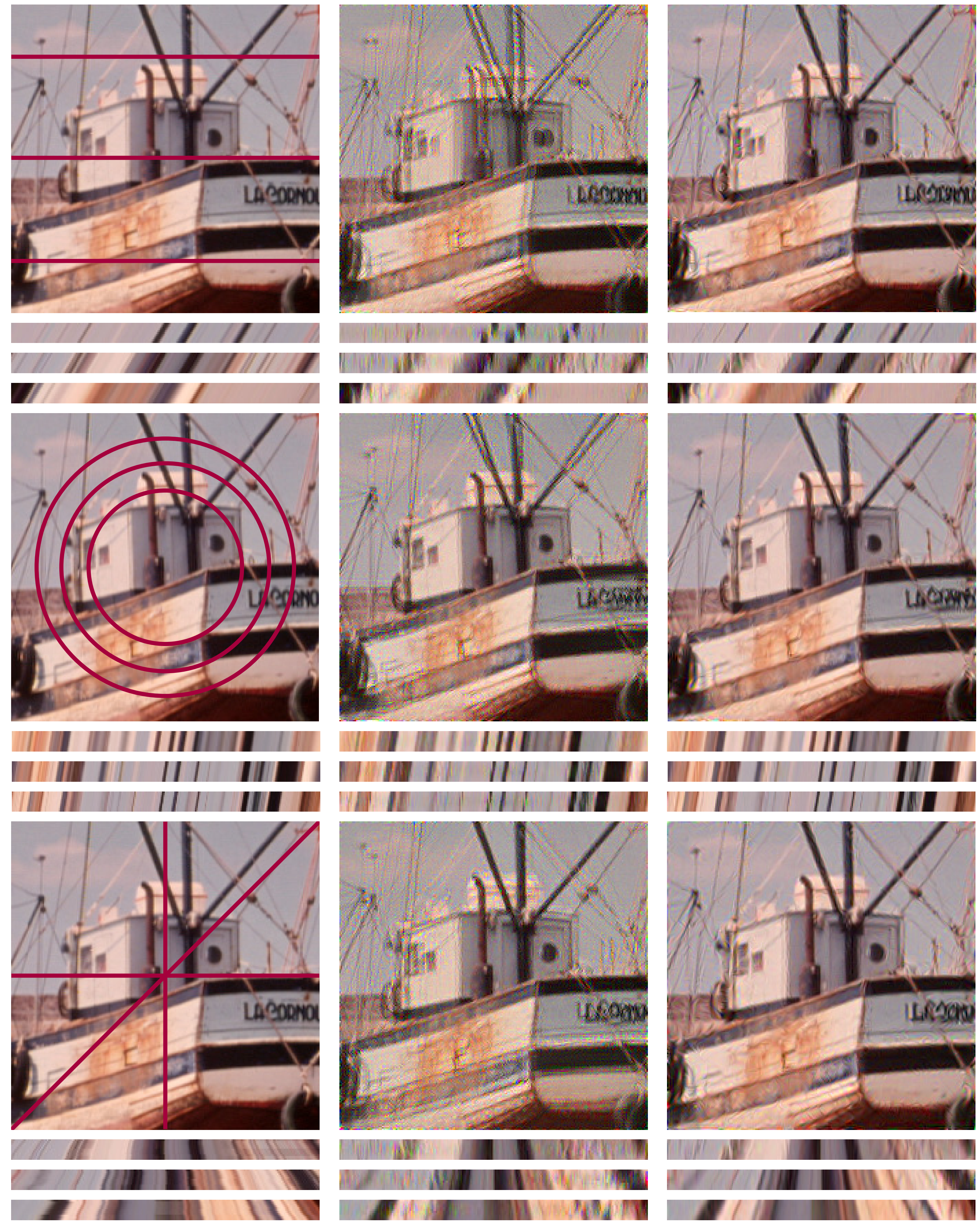


$$\min_{\gamma(0)=x_0,\gamma(1)=x_1} \int_0^1 |\dot{\gamma}(t)| dt + \int_0^1 |(\dot{\Phi\gamma})(t)| dt$$

Invariance, Linearization and Geodesics

- On pertained CNNs (VGG oxford net), linearization is empirically verified for various groups.

- Continuous transformation groups are better linearized with energy pooling than with max-pooling



ground truth | VGG network, max pooling | VGG network, L_2 pooling

[Henaff and Simoncelli'16]

Redundancy in CNNs

$$\Phi(x) = \rho(\dots\rho(x * \Psi_1)\cdots * \Psi_k)))$$

- Large-scale networks contain >10 layers and >10^6 parameters.

- Q: Is there a smaller parametric model that contains good representations?

Redundancy

- "Post-training" model compression:

  Given parameters $\Theta = (\Theta_1, \ldots, \Theta_k)$, find a reparametrization $\tilde{\Phi}$ such that $\mathbb{E}\|\Phi(x;\Theta) - \tilde{\Phi}(x)\|$ is small.

Redundancy

- "Post-training" model compression:

  Given parameters $\Theta = (\Theta_1, \ldots, \Theta_k)$, find a reparametrization $\tilde{\Phi}$ such that $\mathbb{E}\|\Phi(x;\Theta) - \tilde{\Phi}(x)\|$ is small.

  - Useful to accelerate evaluation of large networks ([Denton et al,'14], [Jaderberg et al'14]) ("Optimal Brain Damage" [LeCun et al,'90])
  - Typically we restrict the new class to be

    $\tilde{\Phi}(x) = \Phi(x, \tilde{\Theta})$, $\tilde{\Theta}_i = F(\beta_i)$ $\quad\quad$ $\dim(\beta_i) \ll \dim(\Theta_i)$
  - Explore low-rank tensor factorizations of each convolutional tensor.

Redundancy

- "Post-training" model compression:

  Given parameters $\Theta = (\Theta_1, \dots, \Theta_k)$, find a reparametrization $\tilde{\Phi}$ such that $\mathbb{E}\|\Phi(x;\Theta) - \tilde{\Phi}(x)\|$ is small.

  - Useful to accelerate evaluation of large networks ([Denton et al,'14], [Jaderberg et al'14]) ("Optimal Brain Damage" [LeCun et al,'90])
  - Typically we restrict the new class to be

    $$\tilde{\Phi}(x) = \Phi(x, \tilde{\Theta}) \ , \ \tilde{\Theta}_i = F(\beta_i) \qquad \dim(\beta_i) \ll \dim(\Theta_i)$$
  - Explore low-rank tensor factorizations of each convolutional tensor.

- "Pre-training" model compression:
  - Train directly in the compressed domain ([*"Predicting parameters in Deep Learning"*, Denil et al,'13]).
  - Mild regularization effect. *Interplay between statistical performance and optimization performance.*

Invertibility

- Q: How much information is preserved in a representation arising from a CNN?

  - Under which metric?

  - Which training mechanism?

Invertibility

- No training and some structure: $\Phi = S_J$ Scattering.

  - For a signal of size *N*, we can consider *J=log(N)* to capture the whole receptive field
    - Typically will have less coefficients than input dimensions: compressive recovery.

Invertibility

- No training and some structure: $\Phi = S_J$ Scattering.
  - For a signal of size *N*, we can consider *J=log(N)* to capture the whole receptive field
    - Typically will have less coefficients than input dimensions: compressive recovery.
  - Or we can consider a fixed scale J for a localized (and redundant) representation.
    - The recovery guarantees are looser.

Scattering Sparse Signal Recovery

Theorem [B,M'15]: Suppose $x_0(t) = \sum_n a_n \delta(t - b_n)$ with $|b_n - b_{n+1}| \geq \Delta$, and $\|x\|_1 = \|x_0\|_1$, $\|x * \psi_j\|_1 = \|x_0 * \psi_j\|_1$ for all j. If ψ has compact support, then

$$x(t) = \sum_n c_n \delta(t - e_n) \ , \ \text{with } |e_n - e_{n+1}| \gtrsim \Delta \ .$$

Scattering Sparse Signal Recovery

Theorem [B,M'15]: Suppose $x_0(t) = \sum_n a_n \delta(t - b_n)$ with $|b_n - b_{n+1}| \geq \Delta$, and $\|x\|_1 = \|x_0\|_1$, $\|x * \psi_j\|_1 = \|x_0 * \psi_j\|_1$ for all j. If ψ has compact support, then

$$x(t) = \sum_n c_n \delta(t - e_n) \ , \ \text{with } |e_n - e_{n+1}| \gtrsim \Delta \ .$$

- Sx essentially identifies sparse measures, up to log spacing factors.
- Here, sparsity is encoded in the measurements themselves.
- In 2D, singular measures (ie curves) require $m = 2$ to be well characterized.

Scattering Oscillatory Signal Recovery

Theorem [B,M'14]: Suppose $\widehat{x_0}(\xi) = \sum_n a_n \delta(\xi - b_n)$ with $|\log b_n - \log b_{n+1}| \geq \Delta$, and $S_J x = S_J x_0$ with $m = 2$ and $J = \log N$. If $\widehat{\psi}$ has compact support $K \leq \Delta$, then

$$\widehat{x}(\xi) = \sum_n c_n \delta(\xi - e_n) \ , \ \text{with } |\log e_n - \log e_{n+1}| \gtrsim \Delta \ .$$

Scattering Oscillatory Signal Recovery

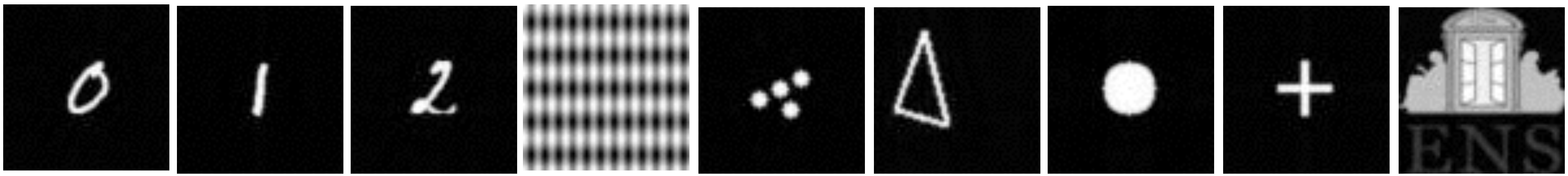
Theorem [B,M'14]: Suppose $\widehat{x_0}(\xi) = \sum_n a_n \delta(\xi - b_n)$ with $|\log b_n - \log b_{n+1}| \geq \Delta$, and $S_J x = S_J x_0$ with $m = 2$ and $J = \log N$. If $\widehat{\psi}$ has compact support $K \leq \Delta$, then

$$\widehat{x}(\xi) = \sum_n c_n \delta(\xi - e_n) \ , \ \text{with} \ |\log e_n - \log e_{n+1}| \gtrsim \Delta \ .$$

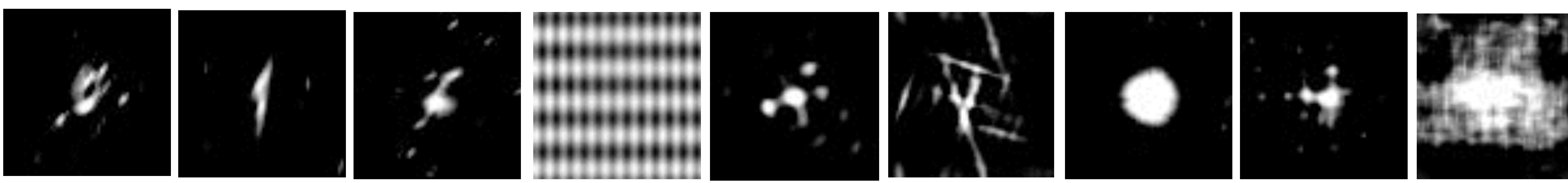
- Oscillatory, lacunary signals are also well captured with the same measurements.
- It is the opposite set of extremal points from previous result.

Sparse Shape Reconstructions

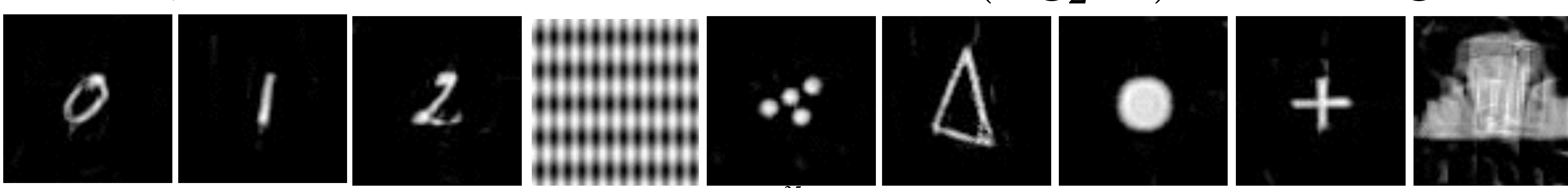
Original images of N^2 pixels:



$m = 1$, $2^J = N$: reconstruction from $O(\log_2 N)$ scattering coeff.



$m = 2$, $2^J = N$: reconstruction from $O(\log_2^2 N)$ scattering coeff.



Invertibility: No training and no structure

$$\Phi = \text{Random Convnet}$$

- [Giryes, Sapiro and Bronstein,'15]

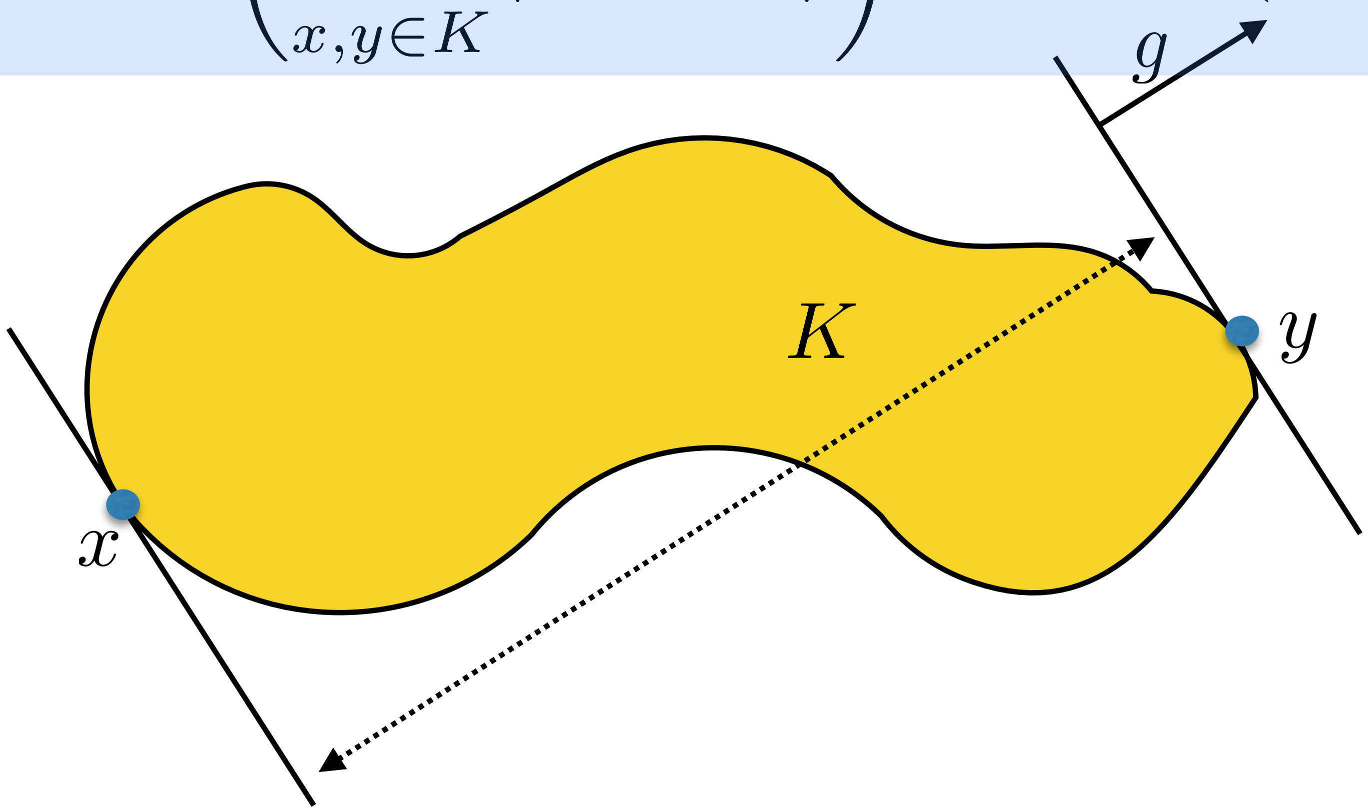
Invertibility: No training and no structure

$\Phi = \text{Random Convnet}$

- [Giryes, Sapiro and Bronstein,'15]

Gaussian mean width of a set K:

$$\omega(K) := \mathbb{E}\left(\sup_{x,y\in K} \langle g, x-y\rangle\right) \ , \ g \sim \mathcal{N}(0, \mathbf{I})$$



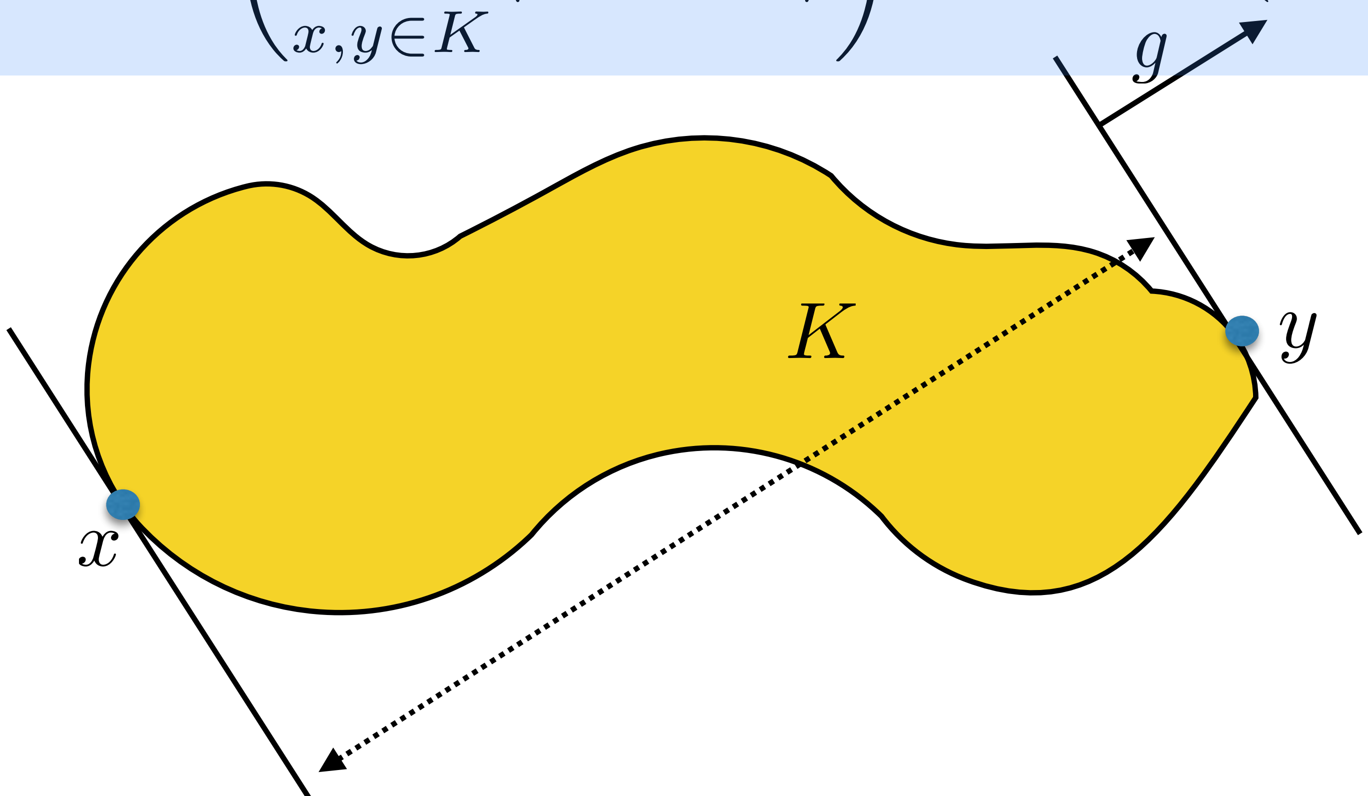
Invertibility: No training and no structure

$\Phi = \text{Random Convnet}$

- [Giryes, Sapiro and Bronstein,'15]

Gaussian mean width of a set K:

$$\omega(K) := \mathbb{E}\left(\sup_{x,y\in K} \langle g, x-y\rangle\right) \quad , \; g \sim \mathcal{N}(0, \mathbf{I})$$



- Proxy for the dimensionality of a set.

K: mixture of L gaussians of dimension k: $\omega(K) = O(\sqrt{k + \log L})$.

K: k-sparse signals in a dictionary of size L: $\omega(K) = O(\sqrt{k \log(L/k)})$.

Invertibility: No training and no structure

Theorem [GSB'15]: Let $\rho(\cdot)$ be the ReLU and $K \subset \mathbb{B}_1^n$ the dataset. If $\sqrt{m}W \in \mathbb{R}^{m\times n}$ is a random matrix with iid normally distributed entries and $m \geq C\delta^{-4}\omega(K)^4$ then with high probability

$$\Big| \|\rho(Wx) - \rho(Wy)\|^2 - \big(0.5\|x-y\|^2 + \|x\|\|y\|\beta(x,y)\big)\Big| \leq \delta \ .$$

Moreover, if K is sufficently away from 0, there exists $C > 0$ such that whp

$$|\cos\angle(\rho(Wx), \rho(Wy)) - \cos(\angle(x,y)) - \beta(x,y)| \leq C\delta \ .$$

$$\angle(x,y) = \cos^{-1}\left(\frac{x^T y}{\|x\|\|y\|}\right) \quad \text{angle between } x \text{ and } y$$

$$\beta(x,y) = \pi^{-1}(\sin(\angle(x,y)) - \angle(x,y)\cos(\angle(x,y))$$

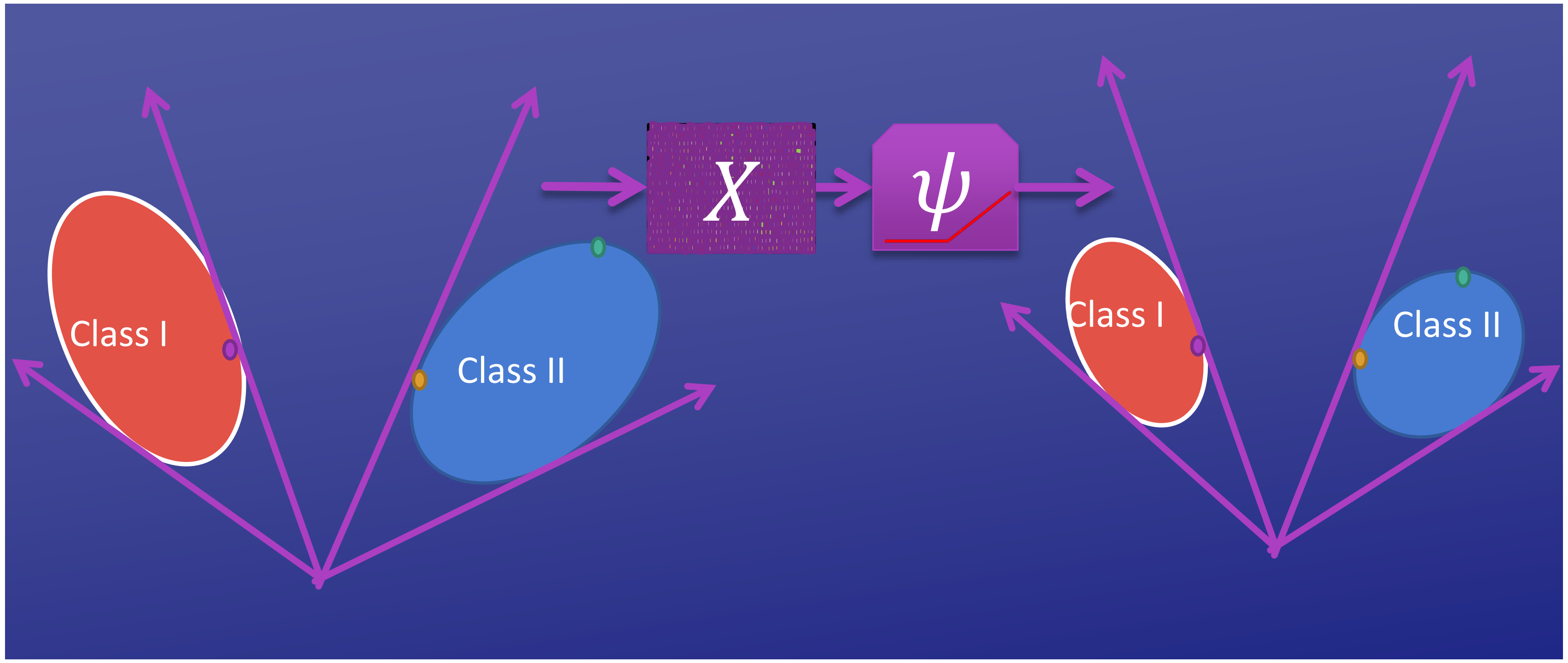
Interpretation

- If $\angle(x, y)$ is small, then $\beta(x, y) \approx 0$:
  distances are approx. shrunk by 2, angles are preserved.

Interpretation

- If $\angle(x, y)$ is small, then $\beta(x, y) \approx 0$:
distances are approx. shrunk by 2, angles are preserved.

- If $\angle(x, y)$ is large, then $\beta(x, y) \approx 0.5$:
distances are shrunk by a smaller factor.



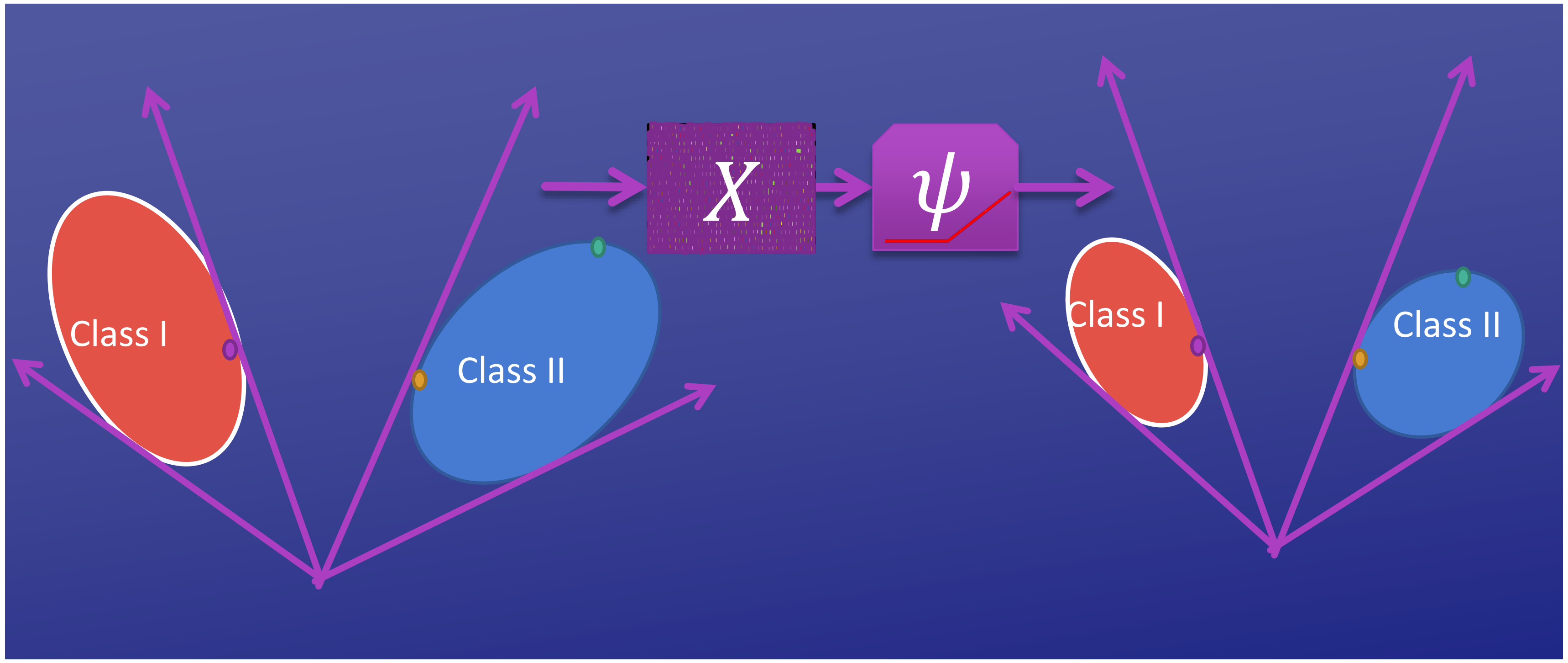
[Raja Giryes]

points with small angles between them become closer than points with larger angles between them

Interpretation

- If $\angle(x, y)$ is small, then $\beta(x, y) \approx 0$:
  distances are approx. shrunk by 2, angles are preserved.

- If $\angle(x, y)$ is large, then $\beta(x, y) \approx 0.5$:
  distances are shrunk by a smaller factor.



[Raja Giryes]

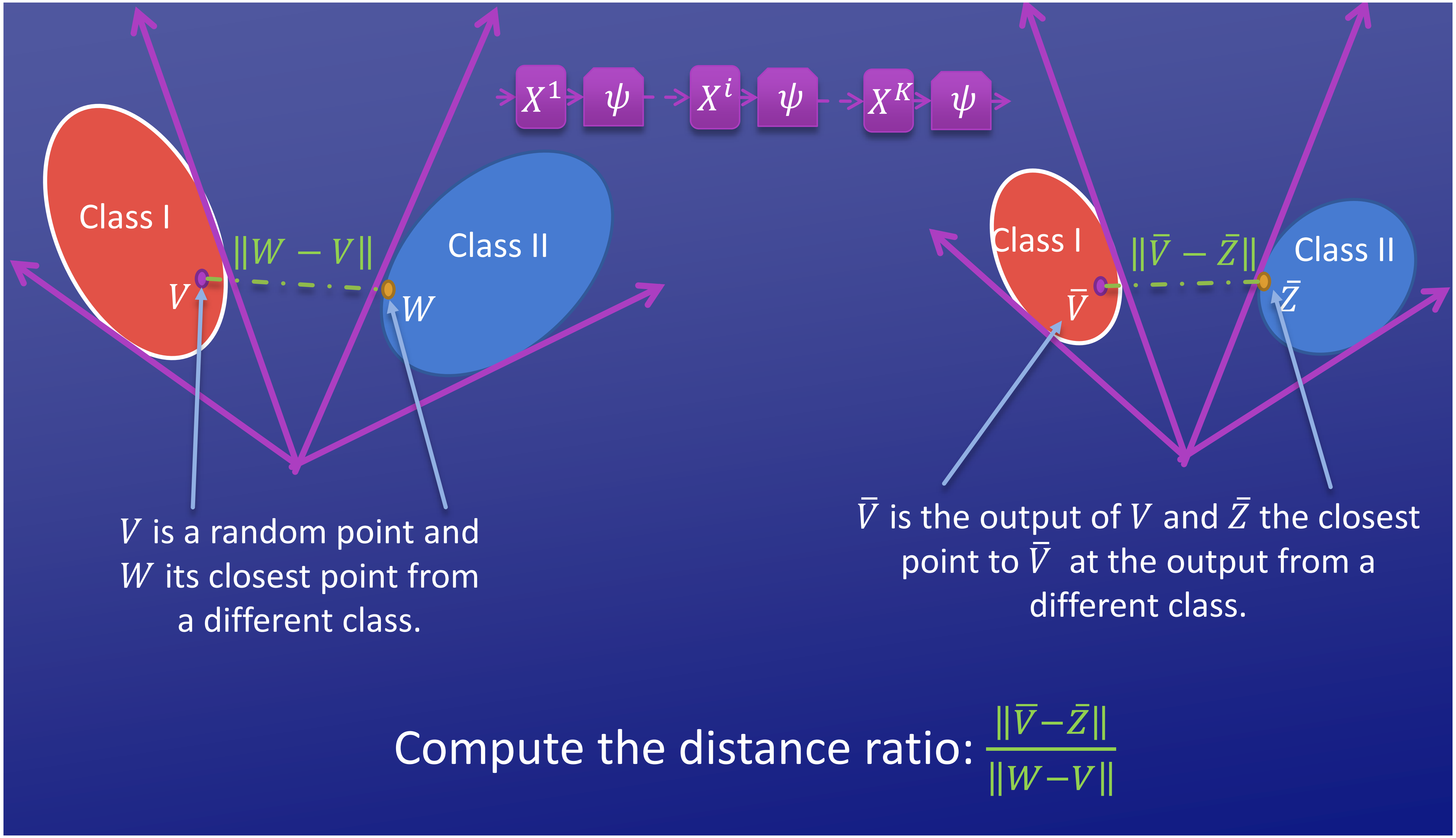
points with small angles between them become closer than points with larger angles between them

The result can be cascaded since gaussian mean width is approximately preserved by each layer.

Role of Training?

Role of Training?

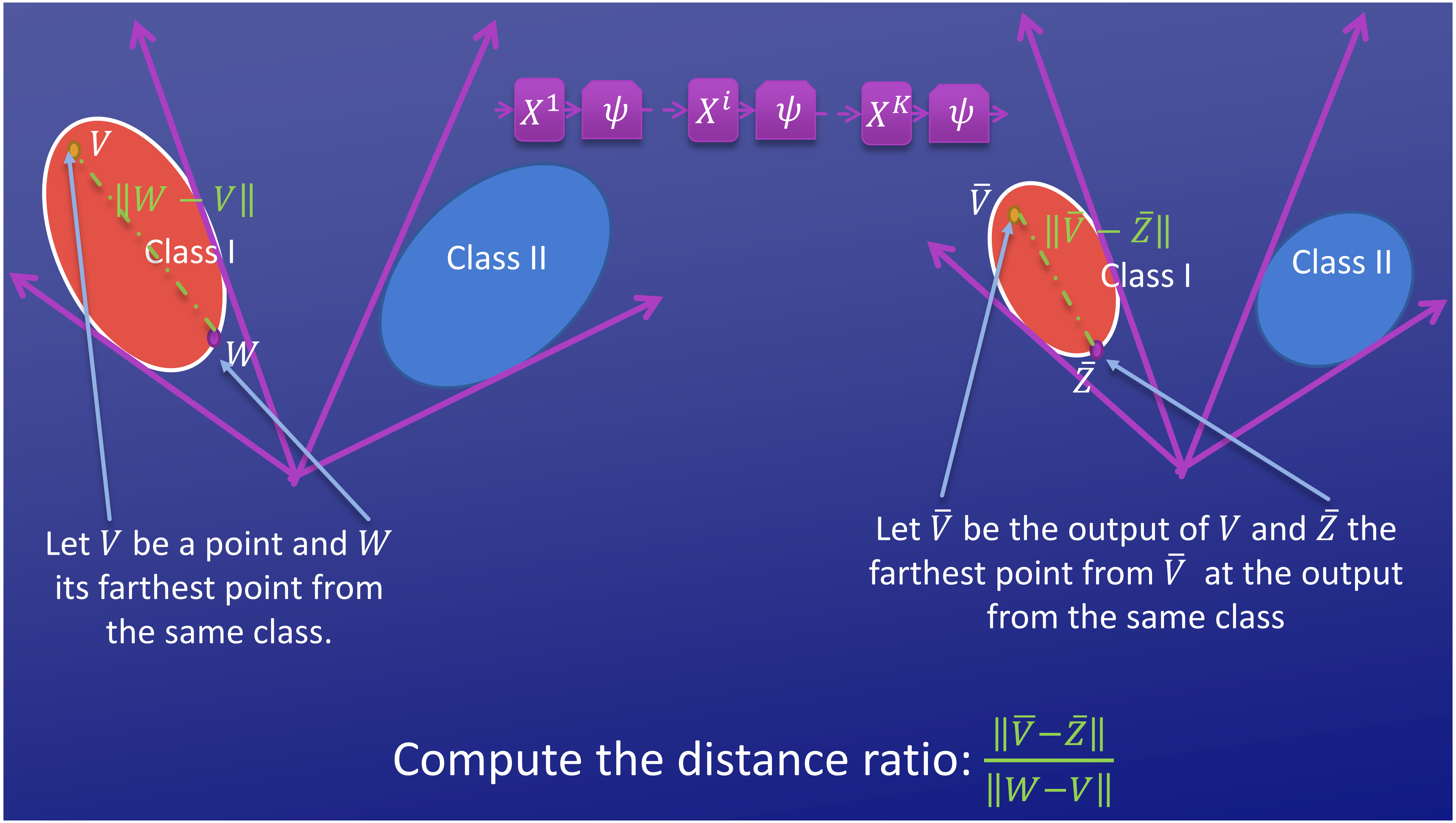
Inter Boundary points distance ratio



[Raja Giryes]

Role of Training?

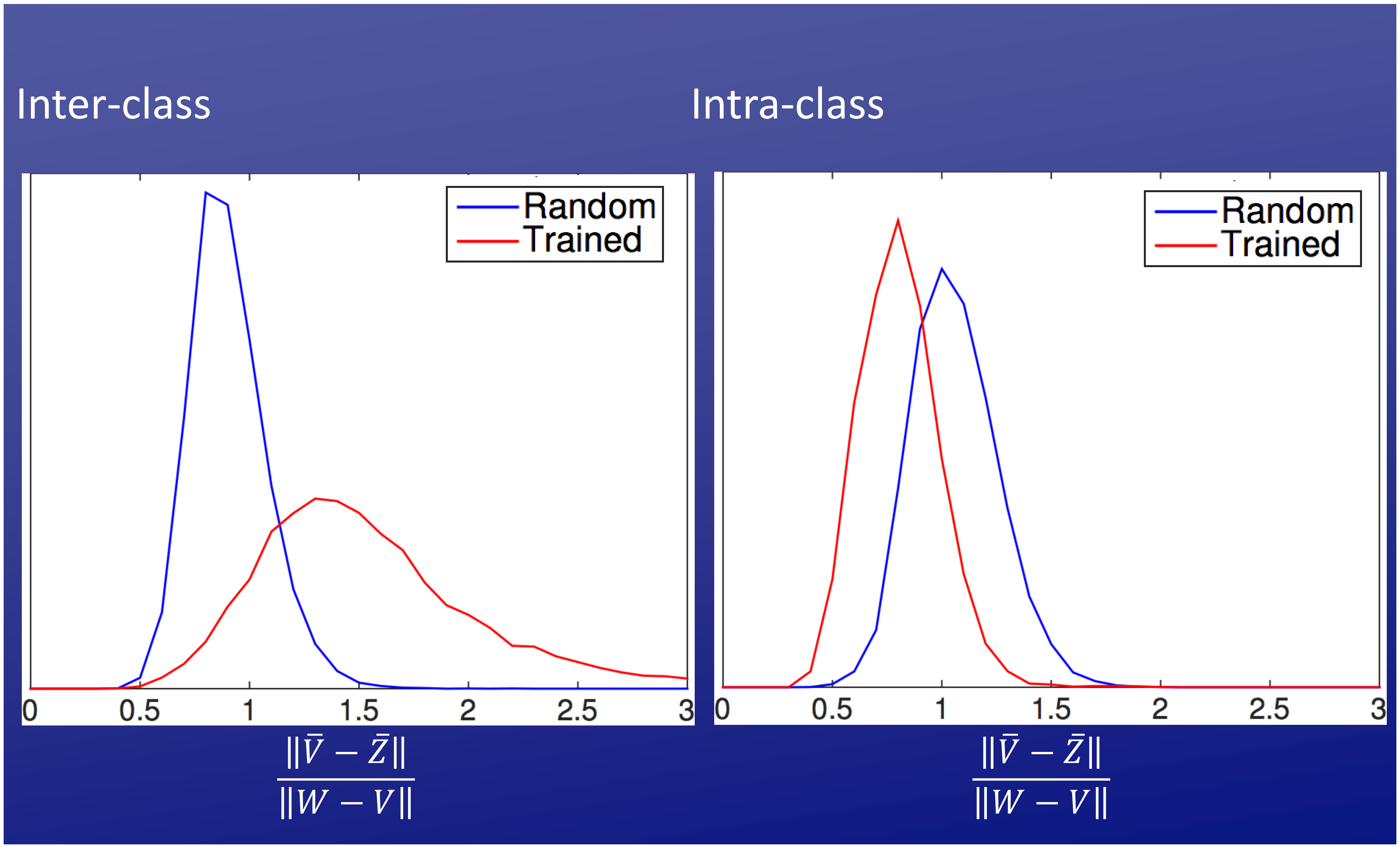
Intra Boundary points distance ratio



[Raja Giryes]

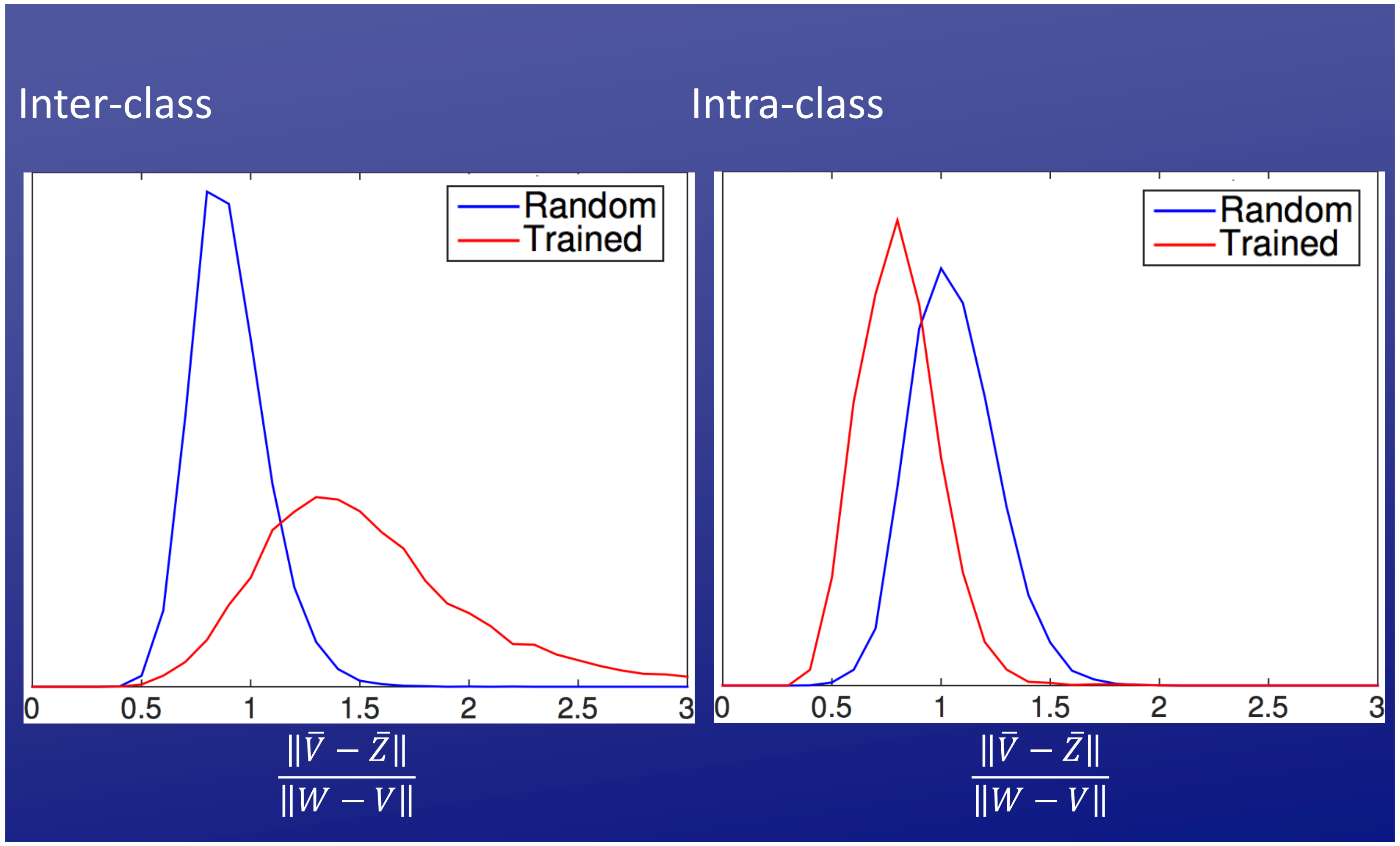
Role of Training?

Boundary distance ratios
measured on Imagenet using VGG oxfordnet



[Raja Giryes]

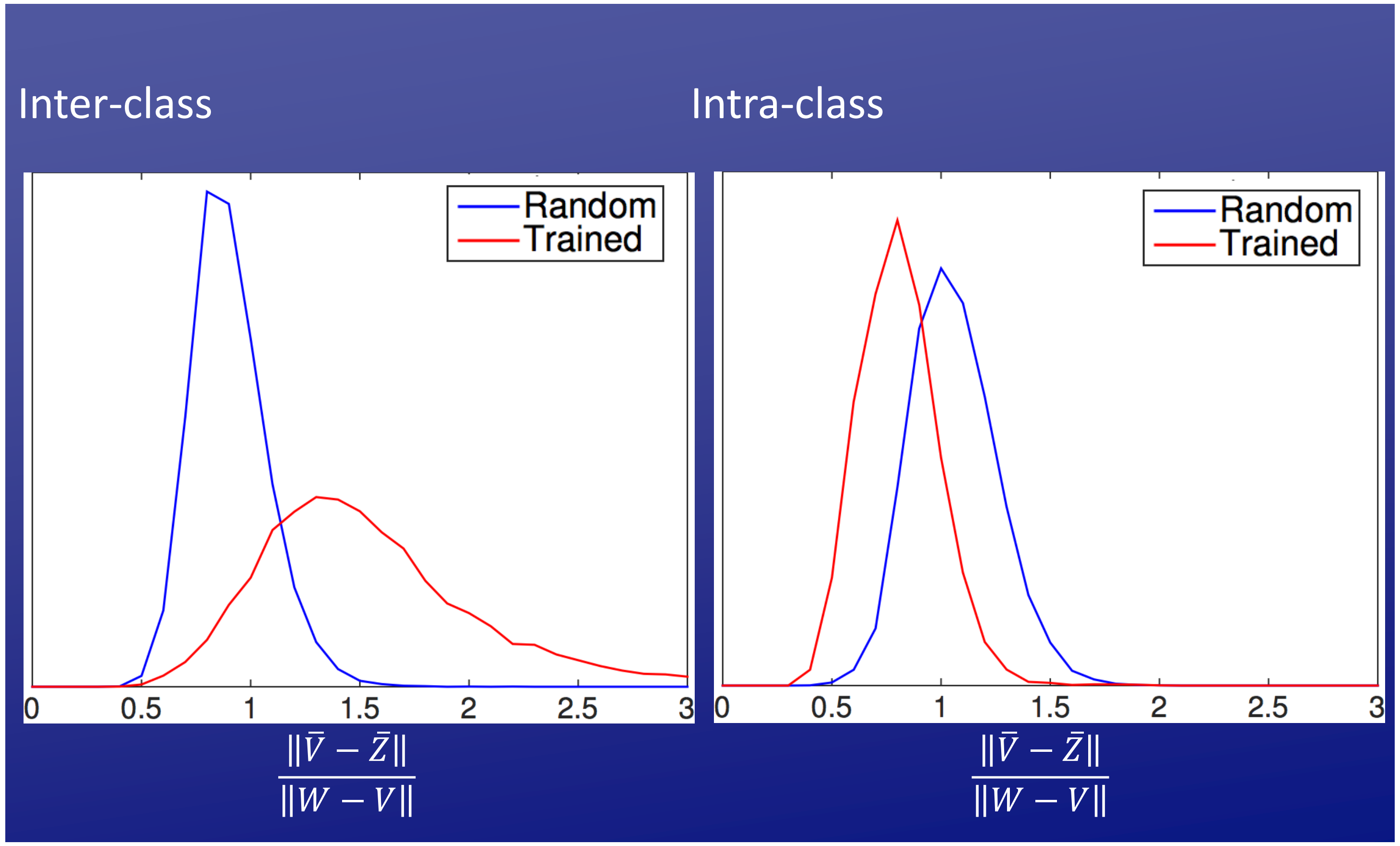
Role of Training?



[Raja Giryes]

- Training the network does not affect the *bulk* of distances

Role of Training?



[Raja Giryes]

- Training the network does not affect the *bulk* of distances
- However, it critically changes the behavior at the boundary points:
  - Inter-class distances expand (as expected).
  - Intra-class distances shrink (as expected).

Invertibility

- For any W, one can ask whether $\Phi(x) = \rho(Wx)$ is invertible, and how stable the inverse is with respect to a recovery measure:

$$c d(x, y) \leq \|\Phi(x) - \Phi(y)\| \leq C d(x, y) \ .$$

$$d(x, y) = \min(\|x - y\|, \|x + y\|)$$

Invertibility

- For any W, one can ask whether $\Phi(x) = \rho(Wx)$ is invertible, and how stable the inverse is with respect to a recovery measure:

$$c d(x,y) \leq \|\Phi(x) - \Phi(y)\| \leq C d(x,y) \ .$$

  - Ex. $d(x,y) = \min(\|x-y\|, \|x+y\|)$

- One can find Lipschitz constants, even when $\rho(\cdot)$ incorporates a pooling operation [B., Szlam, Lecun,'14].

Invertibility

- For any *W*, one can ask whether $\Phi(x) = \rho(Wx)$ is invertible, and how stable the inverse is with respect to a recovery measure:

$$cd(x, y) \leq \|\Phi(x) - \Phi(y)\| \leq Cd(x, y) \ .$$

  - Ex. $d(x, y) = \min(\|x - y\|, \|x + y\|)$

- One can find Lipschitz constants, even when $\rho(\cdot)$ incorporates a pooling operation [B., Szlam, Lecun,'14].
- However, these constants are unpractical and hard to interpret.
  - When *W* is random *iid* they provide recovery guarantees *whp* for appropriate redundancies.

Empirical Recovery

- Q: How far are these bounds to explaining real behavior? (i.e. empirical data distribution with non-random, trained networks)

Empirical Recovery

- Q: How far are these bounds to explaining real behavior? (i.e. empirical data distribution with non-random, trained networks)

$$\min_x \|\Phi(x) - \Phi(x_0)\|^2 + \mathcal{R}(x)$$

$\mathcal{R}(x)$: Regularization with "real" prior (e.g. TV norm)

Empirical Recovery

- Q: How far are these bounds to explaining real behavior? (i.e. empirical data distribution with non-random, trained networks)

$$\min_x \|\Phi(x) - \Phi(x_0)\|^2 + \mathcal{R}(x)$$

$\mathcal{R}(x)$: Regularization with "real" prior (e.g. TV norm)

[Mahendran, Vedaldi,'14]

Empirical Recovery

$$\min_x \|\Phi(x) - \Phi(x_0)\|^2 + \mathcal{R}(x)$$

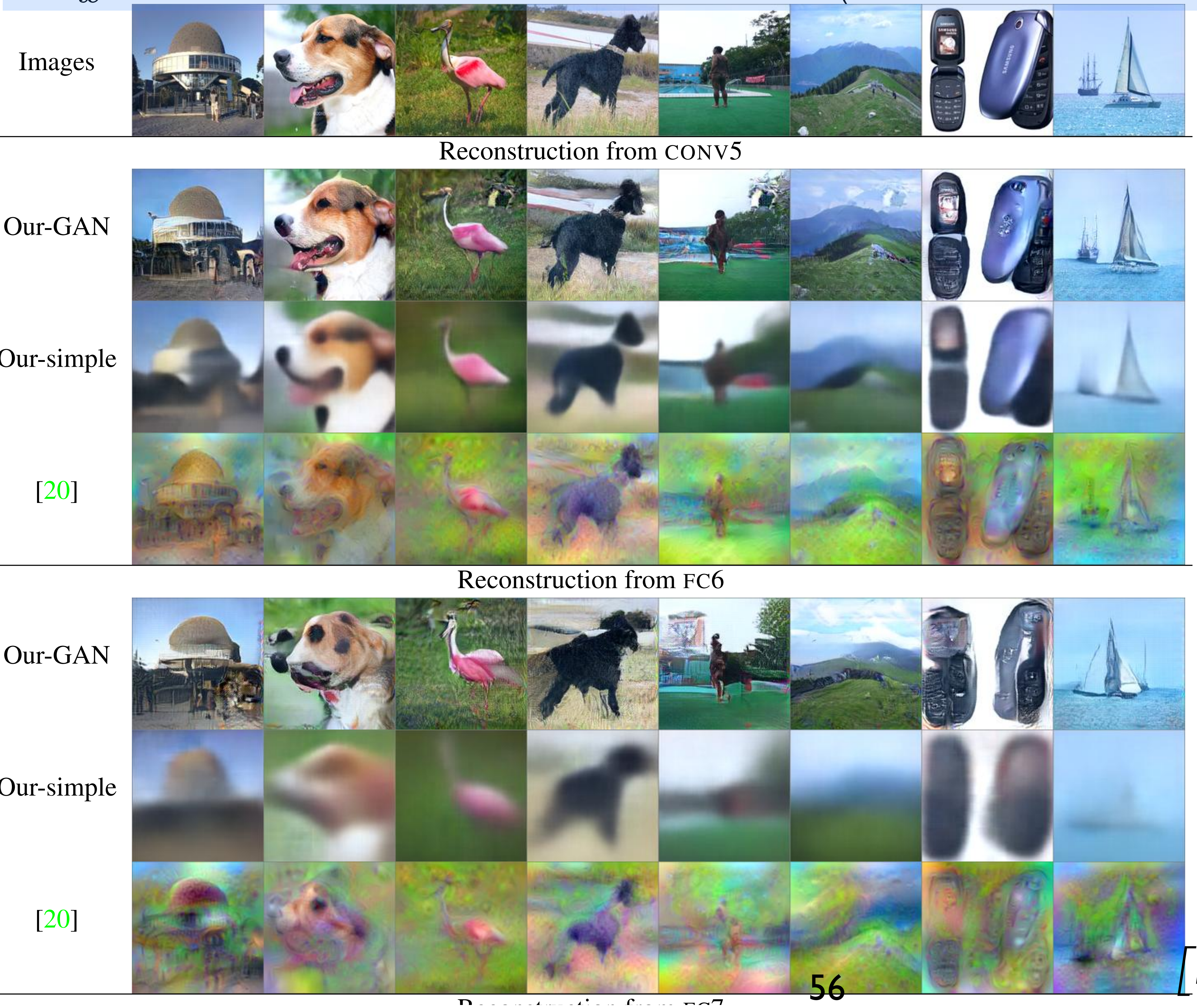
$\mathcal{R}(x)$: Regularization with "learnt" prior (Generative Adversarial Networks, TBD)

[Dosovitsky & Brox'15]

Empirical Recovery

$$\min_x \|\Phi(x) - \Phi(x_0)\|^2 + \mathcal{R}(x)$$

$\mathcal{R}(x)$: Regularization with "learnt" prior (Generative Adversarial Networks, TBD)



[Dosovitsky & Brox'15]

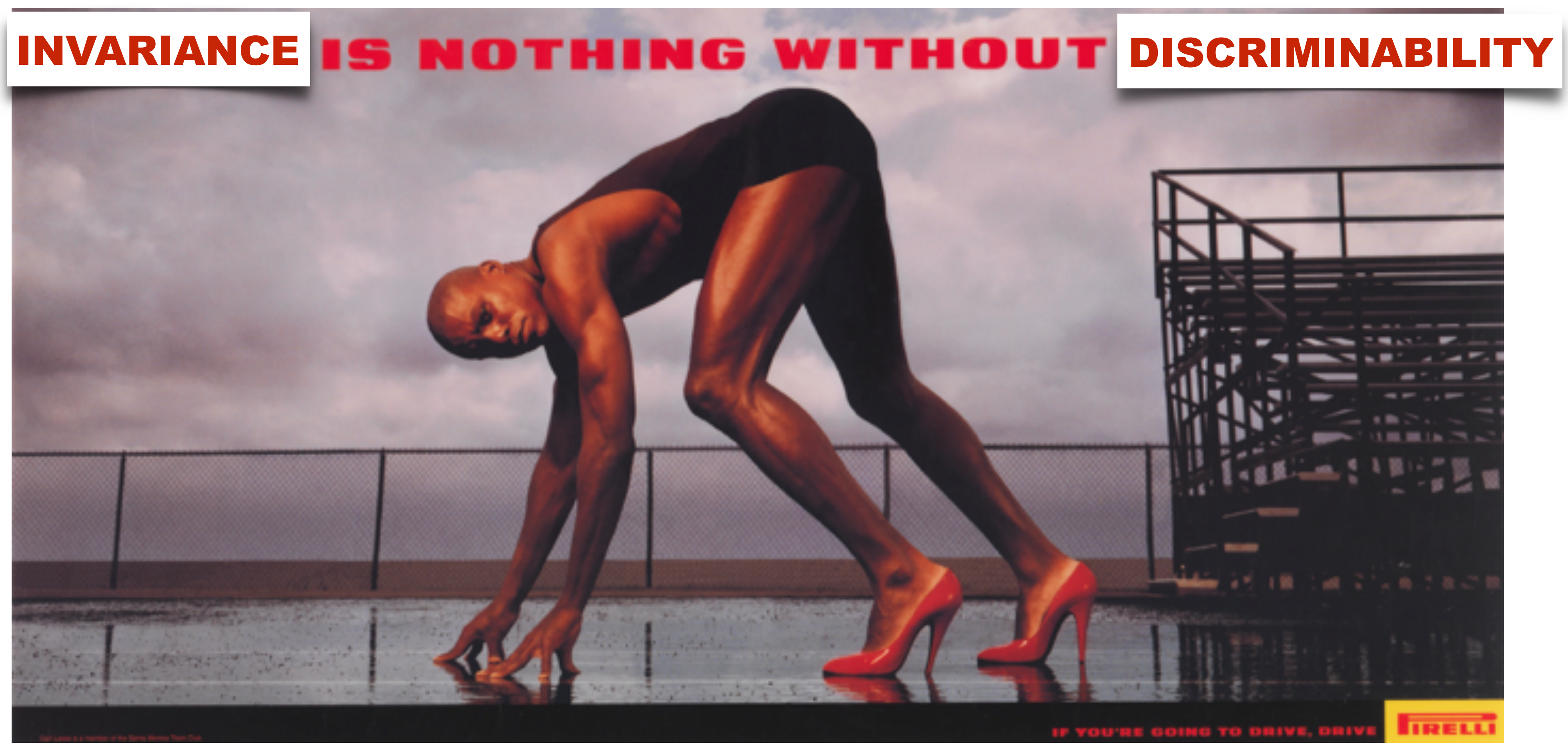
CNNs and Contractions

- So far, we have been mostly interested in the contraction properties of CNNs:
  - Local invariance = reduce intraclass variability

CNNs and Contractions

- So far, we have been mostly interested in the contraction properties of CNNs:
  - Local invariance = reduce intraclass variability

- We mentioned that

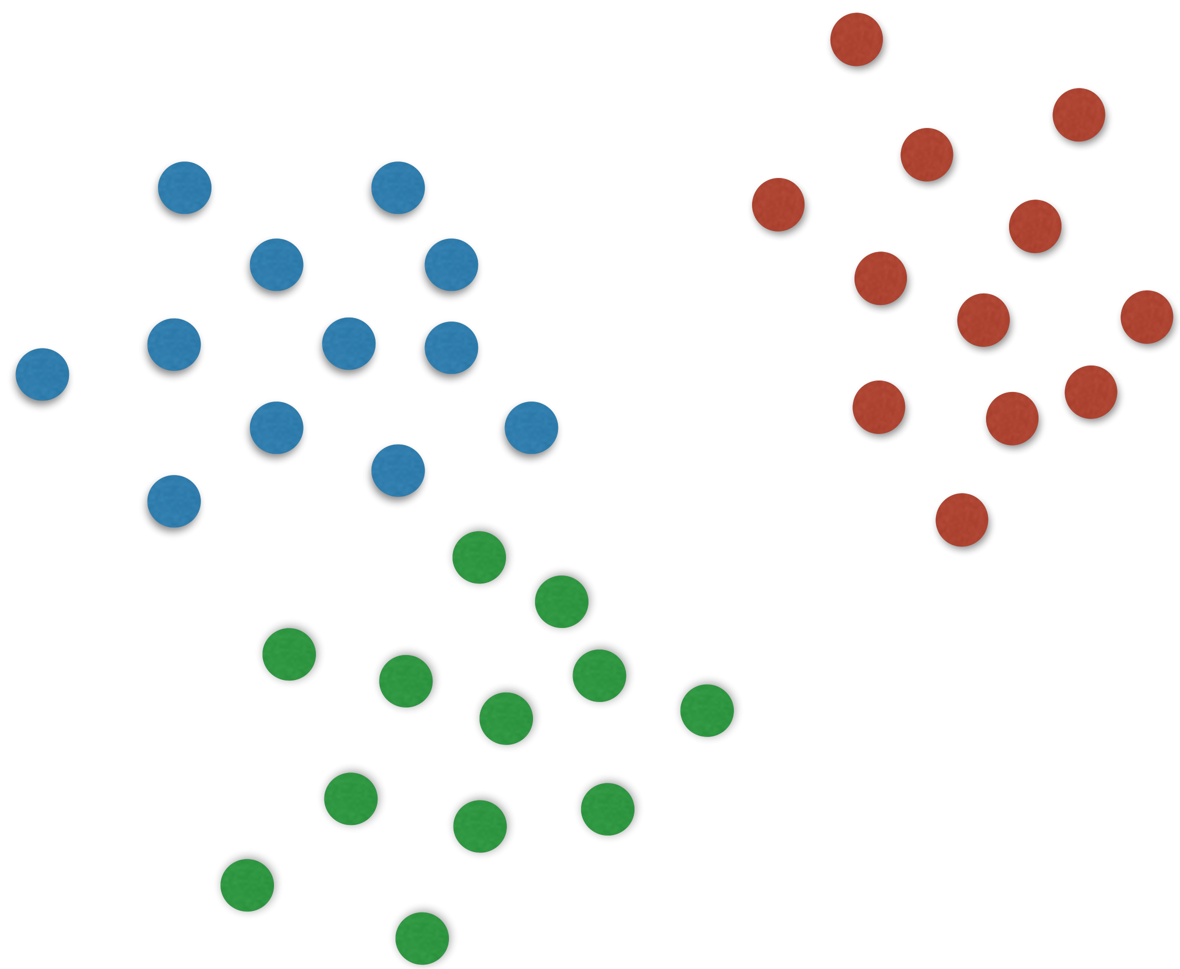


Selection Models

- Q: How to increase separation between classes?

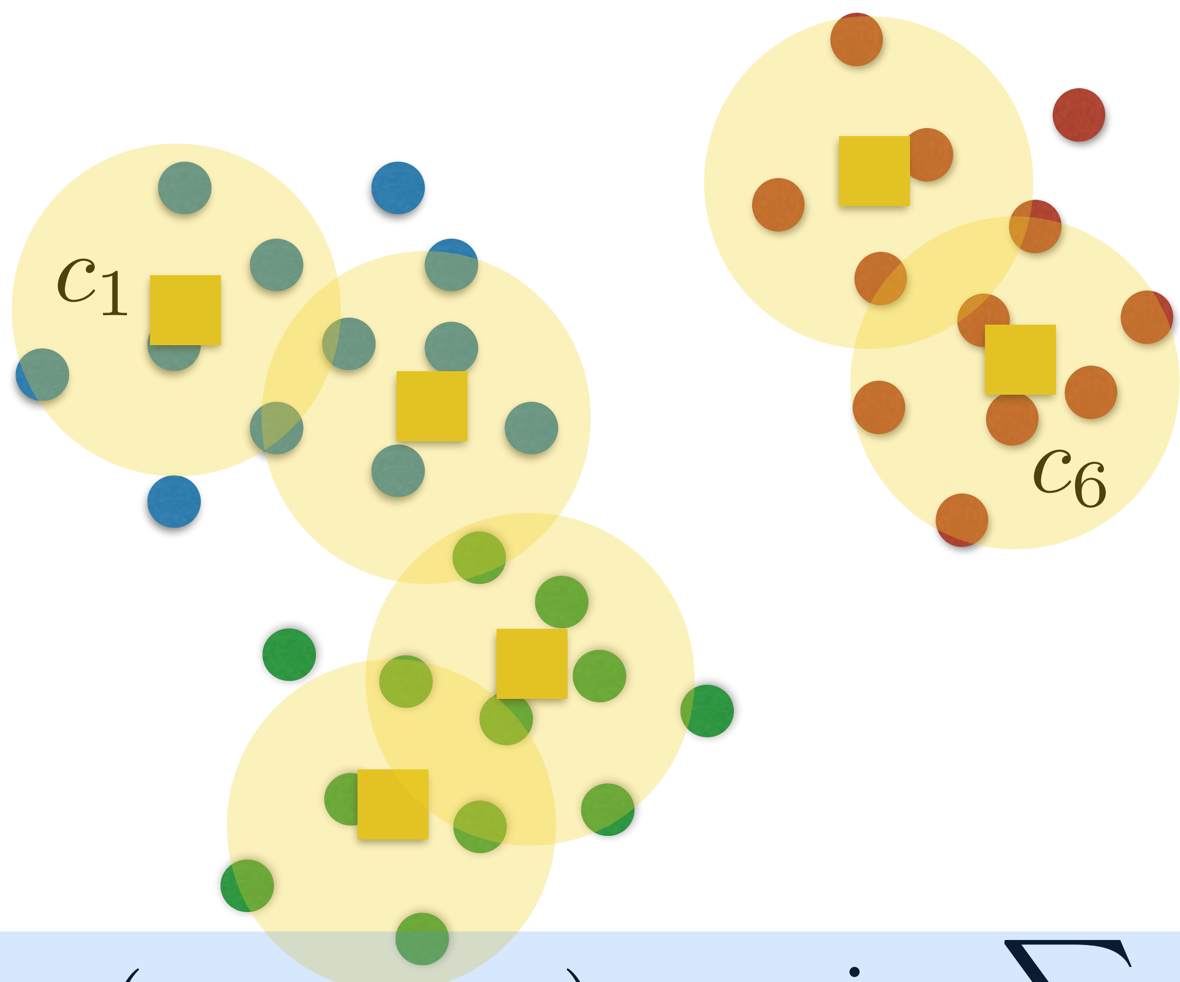
Selection Models

- Q: How to increase separation between classes?
- The simplest model is K-means clustering:



Selection Models

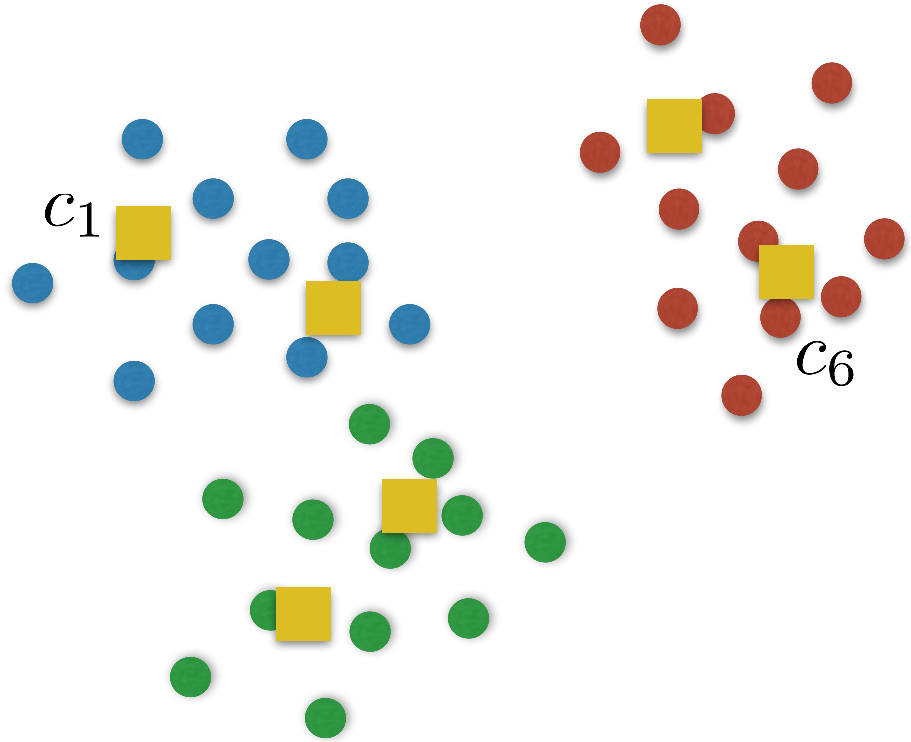
- Q: How to increase separation between classes?
- The simplest model is K-means clustering:



$K = 6$

Given data $X = (x_1, \ldots, x_n)$, $\min_{c_1, \ldots, c_K} \sum_{i \leq n} \min_j \|x_i - c_j\|^2$

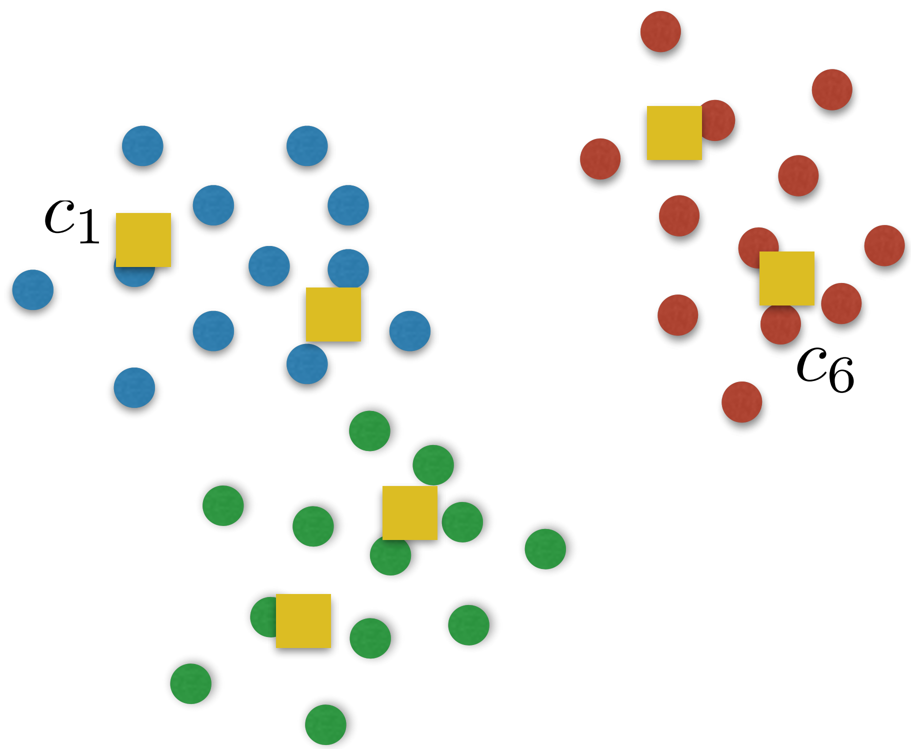
Selection Models



- K-means defines a mapping:

$$\Phi : \mathbb{R}^m \to \mathbb{R}^K$$
$$x \mapsto e_{k(x)} \;, \;\; k(x) = \arg\min_j \|x - c_j\|$$

Selection Models



- K-means defines a mapping:

$$\Phi \; : \; \mathbb{R}^m \to \mathbb{R}^K$$
$$x \mapsto e_{k(x)} \; , \; k(x) = \arg\min_j \|x - c_j\|$$

- Assuming power-normalized data ($\|x\| = 1$), Φ maximally separates points falling into different clusters: $(\langle \Phi(x), \Phi(y) \rangle = 0 \text{ in that case})$

Selection Models

- The K-means encoding is extremely naïve: log(K) bits encoding which region of input space we fall into (piecewise constant encoding)
  - It is nevertheless a very competitive encoding for small image patches.

Selection Models

- The K-means encoding is extremely naïve: log(K) bits encoding which region of input space we fall into (piecewise constant encoding)
  - It is nevertheless a very competitive encoding for small image patches.
- A strictly richer model is the union of subspaces model or dictionary learning:

$$\min_{D=(d_1,\dots,d_K),\|d_k\|\leq 1,z} \sum_{i\leq n} \|x_i - Dz_i\|^2 + \lambda \mathcal{R}(z_i)$$

$\mathcal{R}(z)$: sparsity-promoting

$\mathcal{R}(z) = \|z\|_0$ (NP-Hard)

$\mathcal{R}(z) = \|z\|_1$ (Tractable)

Selection Models

- For a given dictionary D, the *sparse coding* is defined as the mapping

$$\begin{aligned}\Phi \; : \; & \mathbb{R}^m \to \mathbb{R}^K \\ & x \mapsto \Phi(x) = \arg\min_z \|x - Dz\|^2 + \lambda \mathcal{R}(z)\end{aligned} \; .$$

Selection Models

- For a given dictionary D, the *sparse coding* is defined as the mapping

$$\begin{aligned} \Phi \;:\; & \mathbb{R}^m \to \mathbb{R}^K \\ & x \mapsto \Phi(x) = \arg\min_z \|x - Dz\|^2 + \lambda \mathcal{R}(z) \;. \end{aligned}$$

- A particularly attractive choice is $\mathcal{R}(z) = \|z\|_1$
  - in that case $\Phi(x)$ requires solving a convex program.
  - Lasso estimator [Tibshirani,'96]
  - Rich theory in the statistical community.
  - Extensions: Group Lasso, Hierarchical Lasso, etc.

Proximal Splitting

- The sparse coding involves minimizing a function of the form

$$\min_z h_1(z) + h_2(z)$$

$$h_1(z) = \|x - Dz\|^2 \text{ convex and smooth (differentiable)}$$
$$h_2(z) = \lambda\|z\|_1 \text{ convex but non-smooth}$$

Proximal Splitting

- The sparse coding involves minimizing a function of the form

$$\min_z h_1(z) + h_2(z)$$

- $h_1(z) = \|x - Dz\|^2$ convex and smooth (differentiable)
- $h_2(z) = \lambda\|z\|_1$ convex but non-smooth

- A solution can be obtained by alternatively minimizing each term:

Fact: Let $h : \mathbb{R}^m \to \mathbb{R}$ be a convex function. For every $z \in \mathbb{R}^m$,

$$\min_y h(y) + \frac{1}{2}\|z - y\|^2$$

has unique solution, denoted $\mathrm{prox}_h(z)$.

(prox_h is a non-expansive operator for all h)

Forward-Backward Splitting

- It can be shown that if h_1 is convex and differentiable with Lipschitz gradient, and h_2 is convex, then the solutions of

$$\min_z h_1(z) + h_2(z)$$

are characterized by the fixed points of

$$z = \text{prox}_{\gamma h_2}(z - \gamma \nabla h_1(z)) \; \forall \; \gamma \geq 0.$$

Forward-Backward Splitting

- It can be shown that if h_1 is convex and differentiable with Lipschitz gradient, and h_2 is convex, then the solutions of

$$\min_z h_1(z) + h_2(z)$$

are characterized by the fixed points of

$$z = \text{prox}_{\gamma h_2}(z - \gamma \nabla h_1(z)) \ \forall \ \gamma \geq 0.$$

- These can be found by iterating

$$z_{n+1} = \text{prox}_{\gamma_n h_2}(z_n - \gamma_n \nabla h_1(z_n))$$

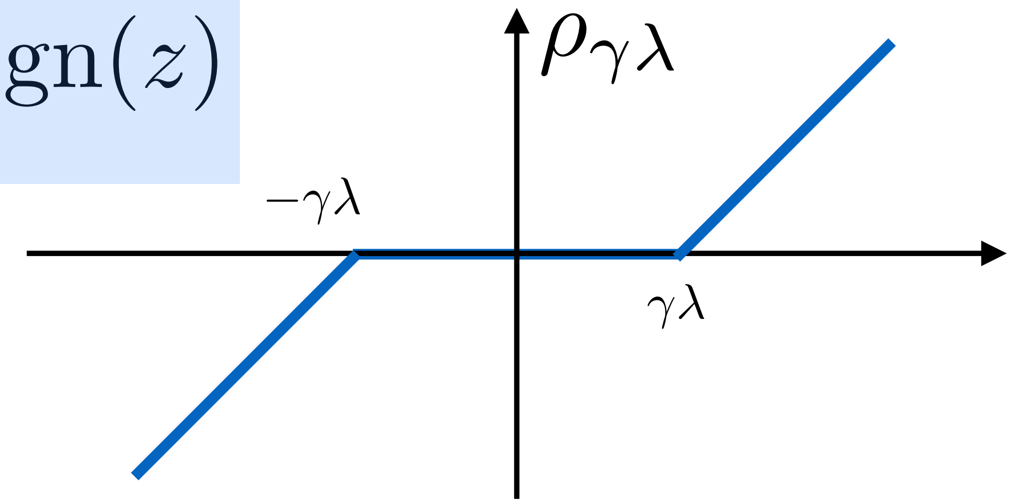
  - by properly adjusting the rate γ_n these method is proven to converge to its unique solution.

Proximal Splitting and ISTA

- When $h_2(z) = \lambda \|z\|_1$, the proximal operator becomes

$$\mathrm{prox}_{\gamma h_2}(z) = \max(0, |z| - \gamma\lambda) \cdot \mathrm{sign}(z)$$

$\rho_{\gamma\lambda}$: *soft thresholding*

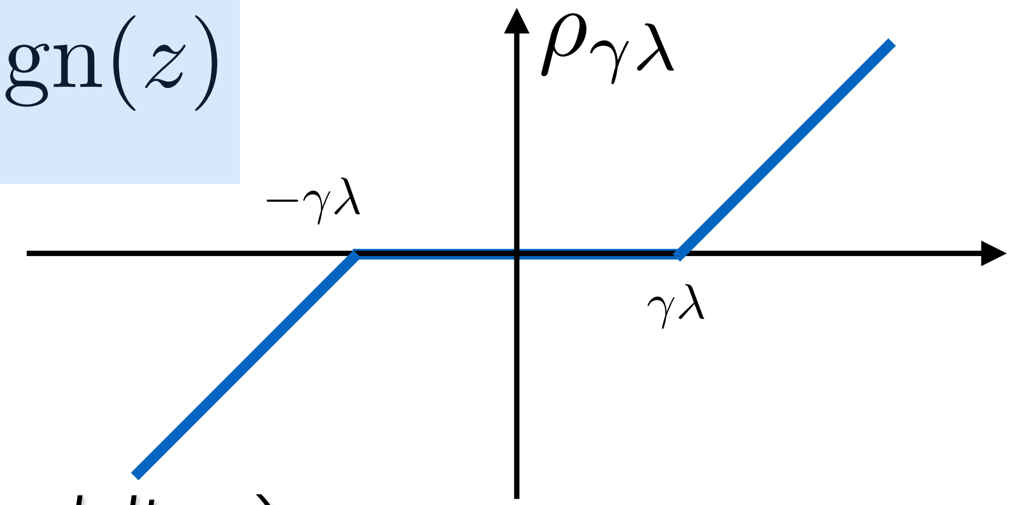


Proximal Splitting and ISTA

- When $h_2(z) = \lambda \|z\|_1$, the proximal operator becomes

$$\mathrm{prox}_{\gamma h_2}(z) = \max(0, |z| - \gamma\lambda) \cdot \mathrm{sign}(z)$$

$\rho_{\gamma\lambda}$ *: soft thresholding*



- ISTA algorithm (*iterative soft thresholding*):

$$z_{n+1} = \mathrm{prox}_{\gamma_n h_2}(z_n - \gamma_n \nabla h_1(z_n))$$

$$\nabla h_1(z_n) = -D^T(x - Dz_n)$$

$$z_{n+1} = \rho_{\gamma\lambda}((\mathbf{1} - \gamma D^T D) z_n + \gamma D^T x)$$

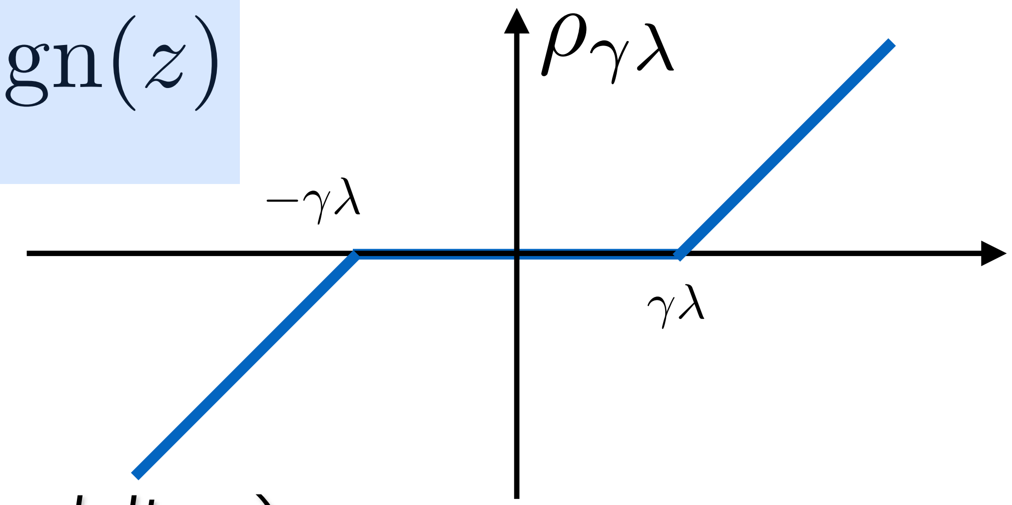
- converges in sublinear time $O(1/n)$ if $\gamma \in (0, 1/\|D^T D\|)$

Proximal Splitting and ISTA

- When $h_2(z) = \lambda \|z\|_1$, the proximal operator becomes

$$\mathrm{prox}_{\gamma h_2}(z) = \max(0, |z| - \gamma\lambda) \cdot \mathrm{sign}(z)$$

$\rho_{\gamma\lambda}$: *soft thresholding*



- ISTA algorithm (*iterative soft thresholding*):

$$z_{n+1} = \mathrm{prox}_{\gamma_n h_2}(z_n - \gamma_n \nabla h_1(z_n))$$

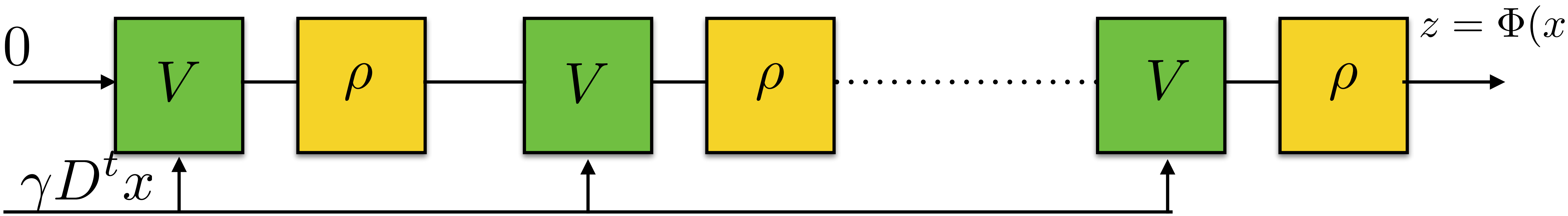
$$\nabla h_1(z_n) = -D^T(x - Dz_n)$$

$$z_{n+1} = \rho_{\gamma\lambda}((\mathbf{1} - \gamma D^T D) z_n + \gamma D^T x)$$

  - converges in sublinear time $O(1/n)$ if $\gamma \in (0, 1/\|D^T D\|)$

- FISTA [*Beck and Teboulle,'09*]:
  - adds Nesterov momentum.
  - proven accelerated convergence $O(1/n^2)$

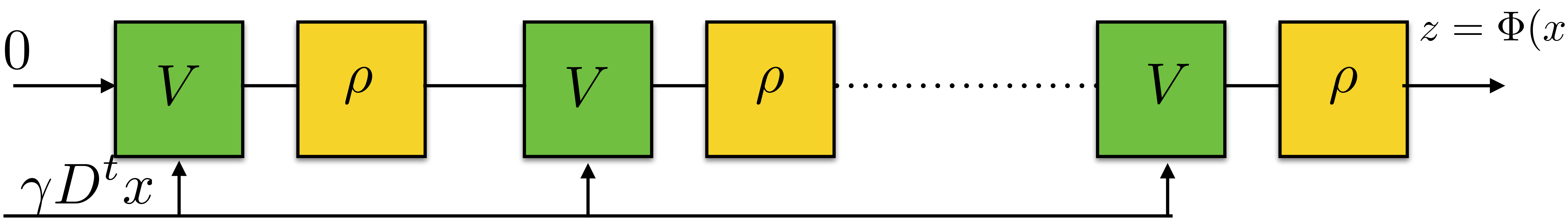
Sparse Coding with (F)ISTA



$Vz = (\mathbf{1} - \gamma D^T D)z + \gamma D^T x)$: linear with bias

ρ: pointwise non-linearity

Sparse Coding with (F)ISTA

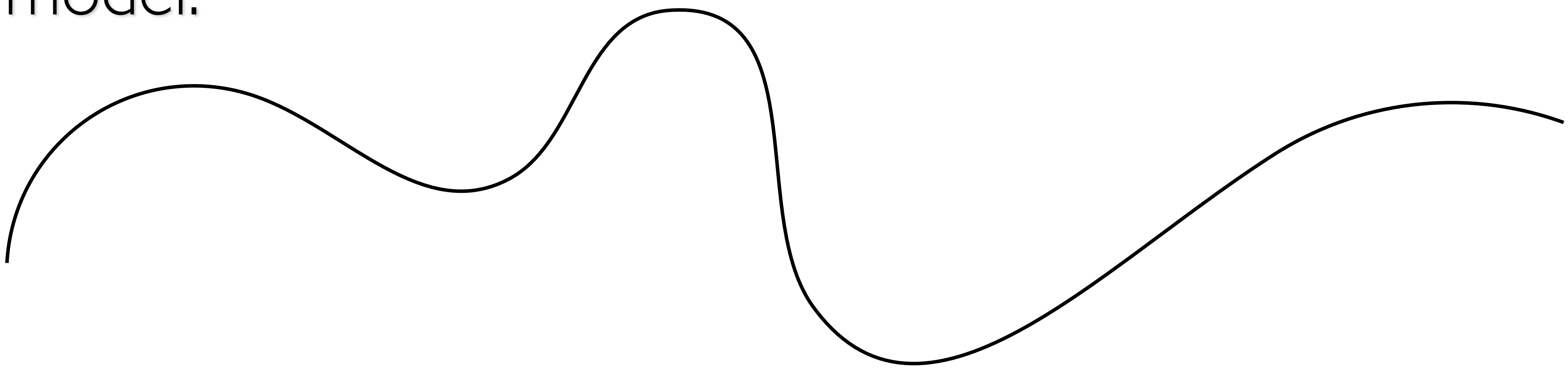


$$Vz = (\mathbf{1} - \gamma D^T D)z + \gamma D^T x)\text{: linear with bias}$$
$$\rho\text{: pointwise non-linearity}$$

- Lasso can be cast as a (very) deep network, with
  - Shared weights, adapted to the dictionary.
$$A = \mathbf{1} - \gamma D^T D \quad , \quad B = \gamma D^T$$
$$\Phi_{n+1}(x) = \rho(A\Phi_n(x) + Bx)$$
  - Note that A is a contraction $(\|Ax\| \leq \|x\|)$, but the affine term may increase the separation:
$$\|\Phi_{k+1}(x) - \Phi_{k+1}(x')\| \leq \|A(\Phi_k(x) - \Phi_k(x'))\| + \|B(x - x')\|$$
$$\leq \|\Phi_k(x) - \Phi_k(x')\| + \|B(x - x')\|$$

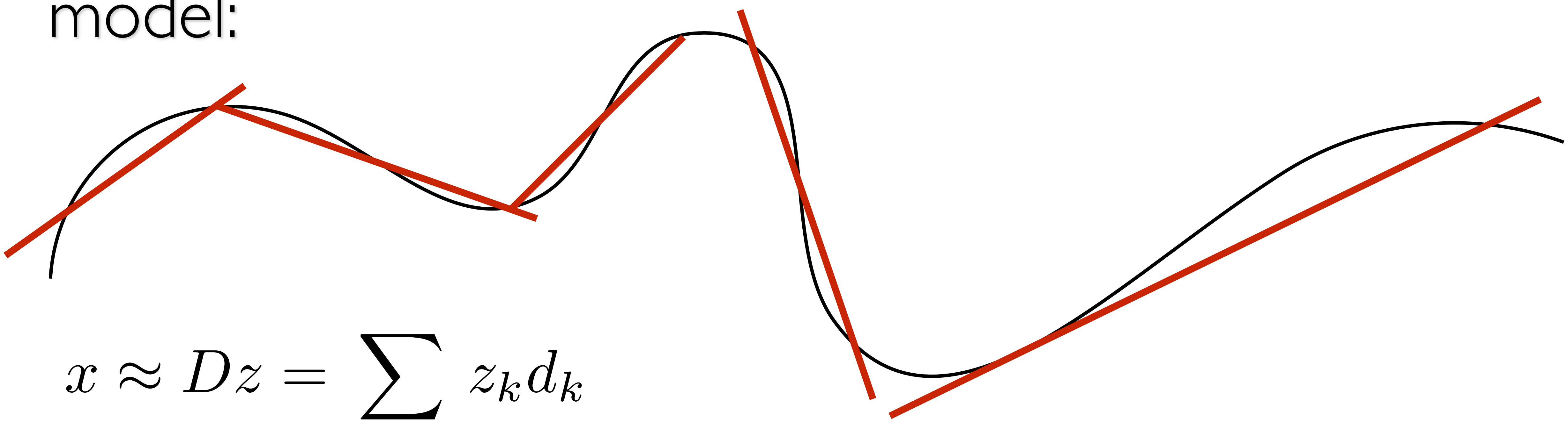
Geometric Interpretation

- Dictionary learning is a locally linear approximation model:



Geometric Interpretation

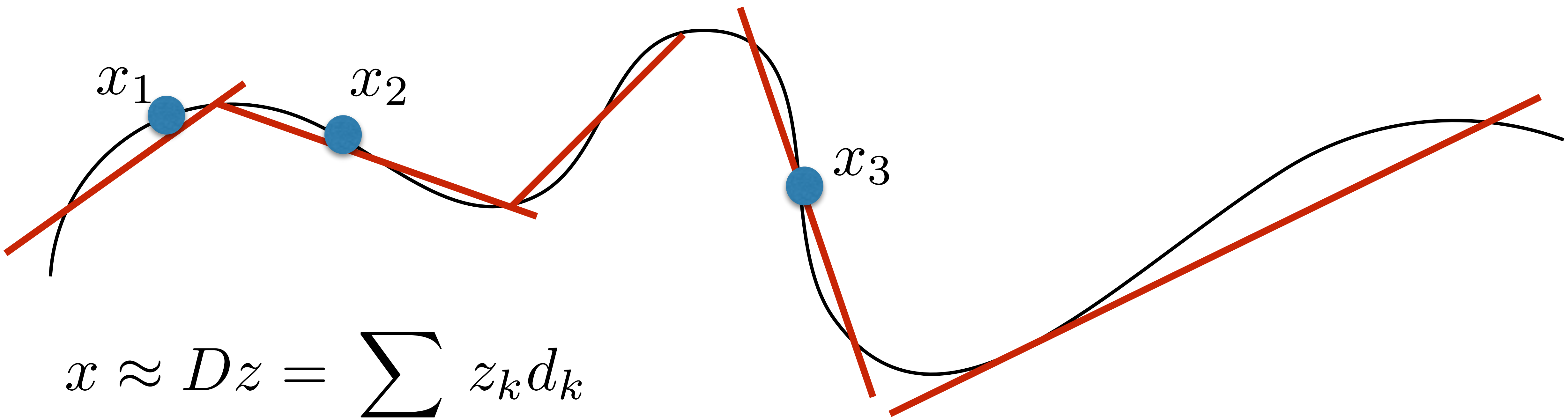
- Dictionary learning is a locally linear approximation model:



$$x \approx Dz = \sum_{z_k \neq 0} z_k d_k$$

Geometric Interpretation

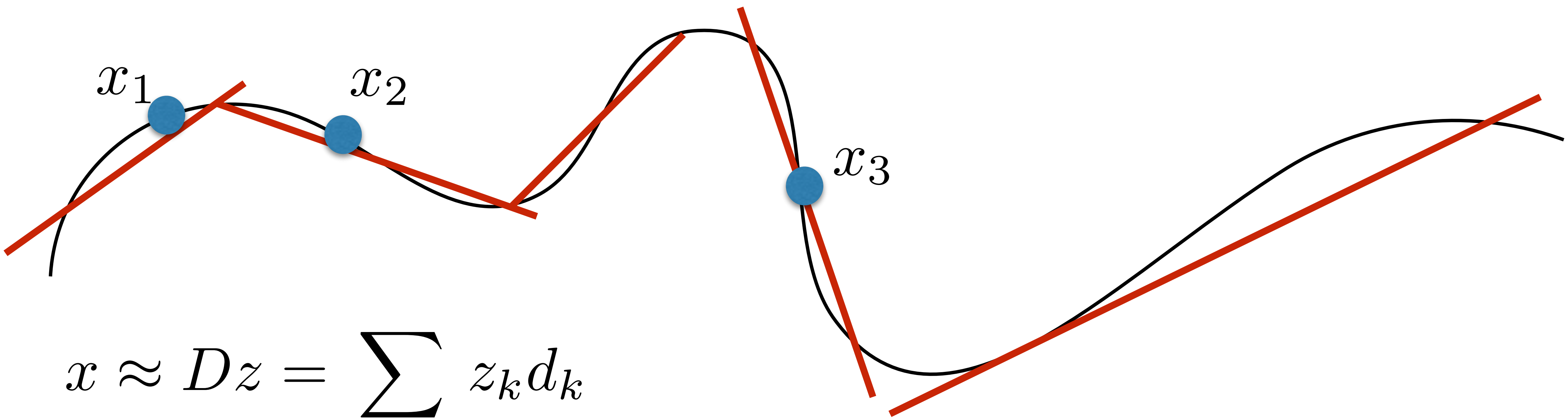
- Orthogonalization of different linear pieces:



$$x \approx Dz = \sum_{z_k \neq 0} z_k d_k$$

Geometric Interpretation

- Orthogonalization of different linear pieces:



$$x \approx Dz = \sum_{z_k \neq 0} z_k d_k$$

- If x_1 and x_2 share most dictionary atoms J, then

  $\langle \Phi(x_1), \Phi(x_2) \rangle \approx \langle D_J^T x_1, D_J^T x_2 \rangle = \langle x_1, D_J D_J^T x_2 \rangle$

  If x_1 and x_3 do not share dictionary atoms, then

  $\langle \Phi(x_1), \Phi(x_3) \rangle \approx 0$

Sparse Coding and Stability

Linear decoder implies geometric instability is preserved in the sparse decomposition

$$\|x - Dz\| \leq \epsilon \|x\| \quad , \quad \|\varphi_\tau x - D z_\tau\| \leq \epsilon \|x\| \quad , \quad \|x - \varphi_\tau x\| \sim \|x\|$$

$$\Downarrow$$

$$\|x - \varphi_\tau x\| \leq \|Dz - Dz_\tau\| + 2\epsilon$$

$$\Downarrow$$

$$\begin{aligned} \|z - z_\tau\| &\geq \|D\|_\infty^{-1} \|Dz - Dz_\tau\| \\ &\geq \|D\|_\infty^{-1} (\|x - \varphi_\tau x\| - 2\epsilon\|x\|) \\ &\sim \|D\|_\infty^{-1} (1 - 2\epsilon) \|x\| \end{aligned}$$

From unsupervised to supervised selection

- The previous model is unsupervised:
  - Why would a dictionary for reconstruction be useful for recognition or other tasks?
  - Pro: it exploits the local regularity of the data.
  - Cons: sparse coding unaware of stability, sparse dictionaries might be not unique.

From unsupervised to supervised selection

- The previous model is unsupervised:
  - Why would a dictionary for reconstruction be useful for recognition or other tasks?
  - Pro: it exploits the local regularity of the data.
  - Cons: sparse coding unaware of stability, sparse dictionaries might be not unique.

- Q: Can we make a dictionary task-aware? (i.e. supervised dictionary learning)

From unsupervised to supervised selection

- Task-driven dictionary learning [Mairal et al,'12]:

Suppose we want to predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$

From unsupervised to supervised selection

- Task-driven dictionary learning [Mairal et al,'12]:

Suppose we want to predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$

Consider the sparse coding operator

$$\Phi(x; D) = \arg\min_z \frac{1}{2}\|x - Dz\|^2 + \lambda\|z\|_1 + \lambda_2\|z\|_2^2$$

It is Lipschitz with respect to both x and D if $\lambda_2 > 0$,
it is differentiable almost everywhere.

From unsupervised to supervised selection

- Task-driven dictionary learning [Mairal et al,'12]:

Suppose we want to predict $y \in \mathcal{Y}$ from $x \in \mathcal{X}$

Consider the sparse coding operator

$$\Phi(x; D) = \arg\min_z \frac{1}{2}\|x - Dz\|^2 + \lambda\|z\|_1 + \lambda_2\|z\|_2^2$$

It is Lipschitz with respect to both x and D if $\lambda_2 > 0$,
it is differentiable almost everywhere.

We can construct an estimator $\hat{y}$ from this sparse code:

$\hat{y} = W^T \Phi(x; D)$ (more generally, , $\hat{y} = F(W, \Phi(x; D))$)

$$\min_{D,W} \mathbb{E}_{x,y} \ell(y, \hat{y}(x, W, D))$$

From unsupervised to supervised

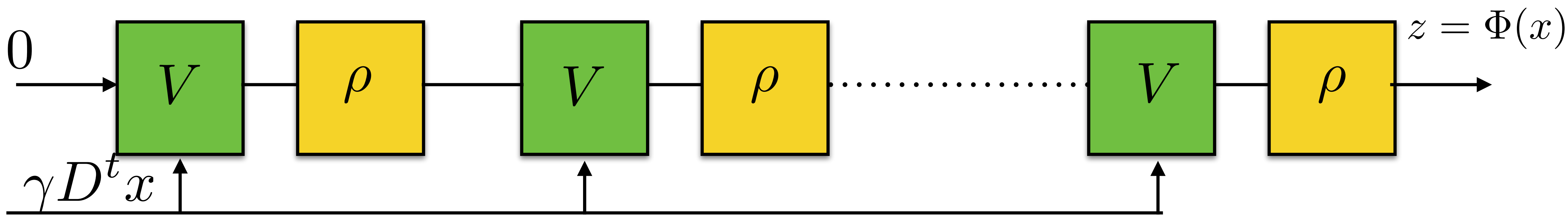
- Half-toning Results from [Mairal et al,'12]:

From supervised Lasso to DNNs

- The Lasso (sparse coding operator) can be implemented as a specific deep network.

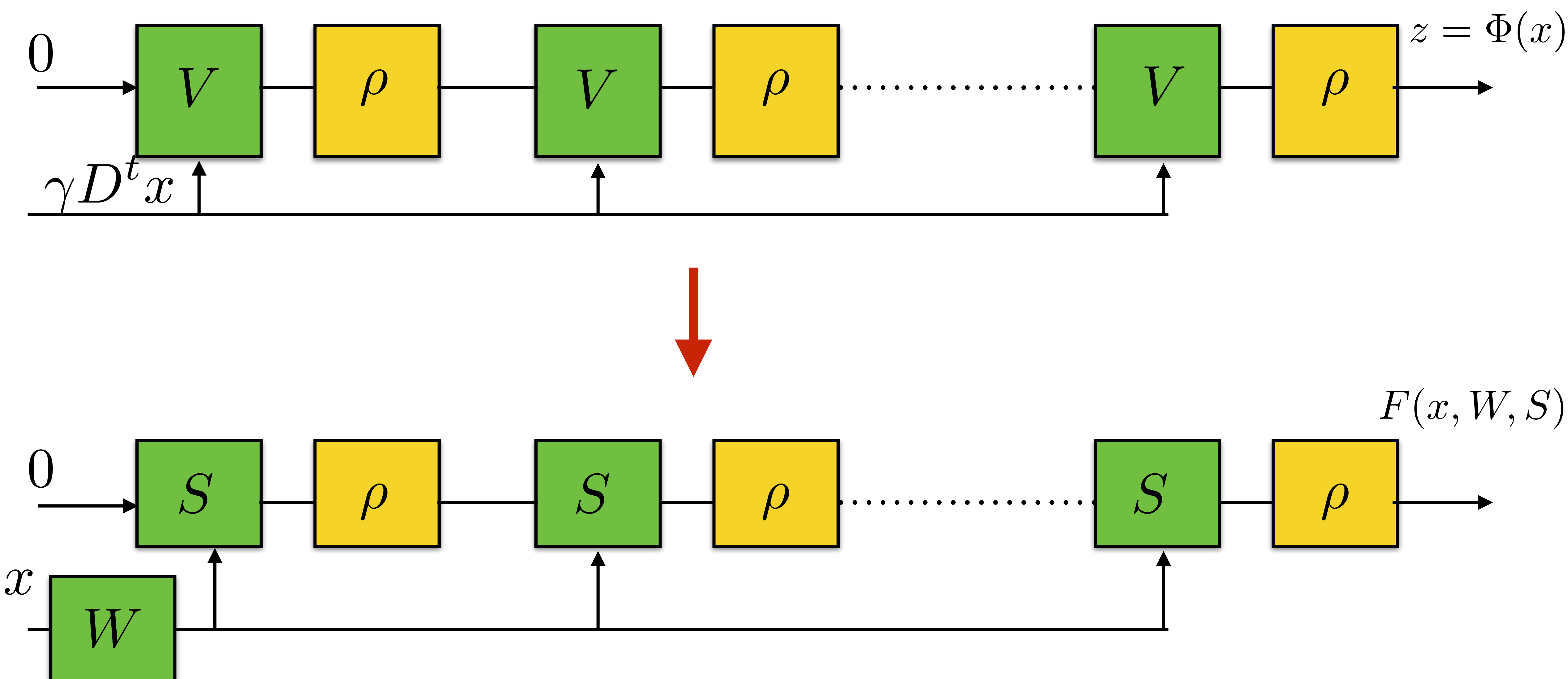
From supervised Lasso to DNNs

- The Lasso (sparse coding operator) can be implemented as a specific deep network
- Can we accelerate the sparse inference with a shallower network, with trained parameters?



From supervised Lasso to DNNs

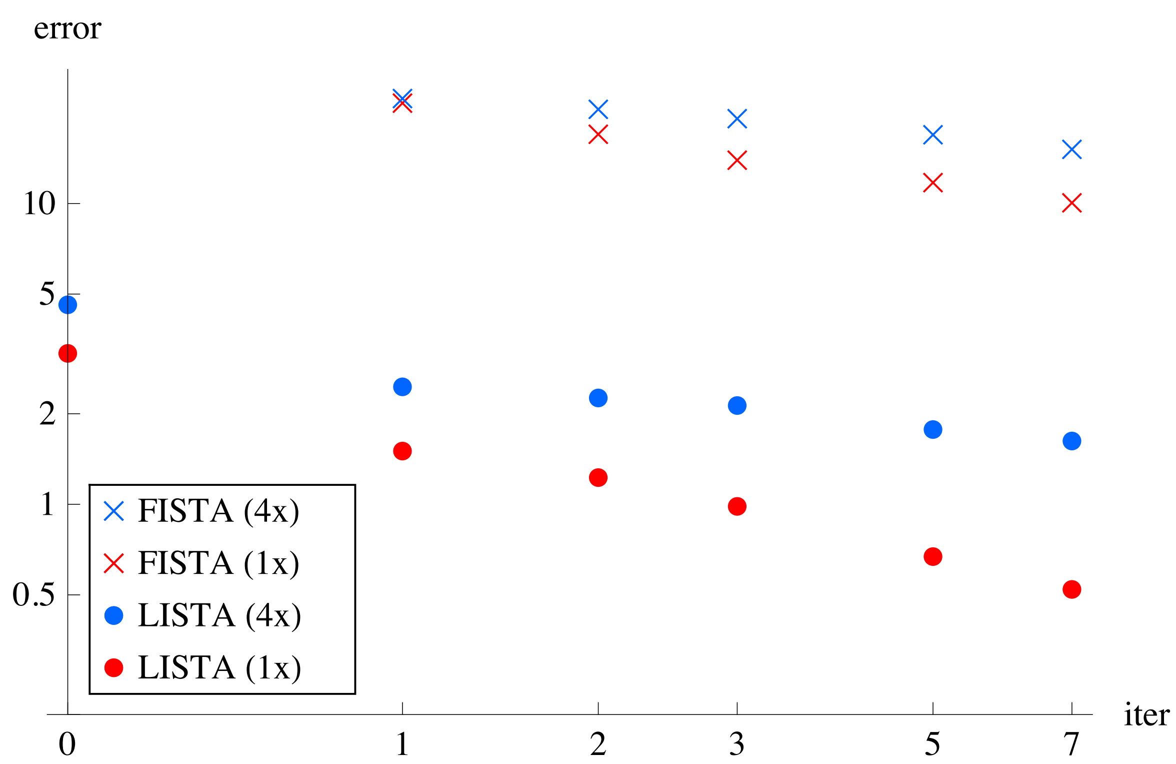
- The Lasso (sparse coding operator) can be implemented as a specific deep network
- Can we accelerate the sparse inference with a shallower network, with trained parameters?



LISTA [Gregor and LeCun,'10]

- Explicit Sparse encoder trained to predict the output of the Lasso:

$$\min_{W,S} \frac{1}{n} \sum_{i \leq n} \|\Phi(x_i) - F(x_i, W, S)\|^2$$



- LISTA adapts to the data distribution and produces much faster approximate sparse codes.

From supervised sparse coding to DNN

- The fast approximation of a sparse code can be plugged-in in a supervised regression or classification task.

- For example, [Sprechmann, Bronstein & Sapiro,'12] in speaker identification experiments using non-negative matrix factorization:

Noise	**Exact**	**RNMF Encoders** (*Supervised*)	**RNMF Encoders** (*Discriminative*)
street	0.86	0.91	0.91
restaurant	0.91	0.89	0.90
car	0.90	0.91	0.96
exhibition	0.93	0.91	0.95
train	0.93	0.88	0.96
airport	0.92	0.85	0.98
average	0.91	0.89	0.94