MATHEMATICS OF DEEP LEARNING

JOAN BRUNA , CIMS + CDS, NYU, SPRING'18

Lecture 4: Non-Euclidean Geometric Stability, Graph Neural Networks.

LECTURE 4 OVERVIEW

- Joint Scattering for Roto-translation groups
- Convolutional Neural Networks
  - Some (random) properties
- Non-Euclidean Geometric Stability
- Graph Neural Networks
- Some (random) applications

WAVELET COVARIANTS

- If we replace input image by first layer output:

$$\rho(x_0 \star \psi_{j,\theta})(u) = x_1(u, j, \theta)$$

Let $\tilde{x}_0 = R_\alpha x_0$ be a rotation of α degrees.

$$\rho(\tilde{x}_0 \star \psi_{j,\theta})(u) = x_1(R_\alpha u, j, \theta + \alpha)$$

WAVELET COVARIANTS

- If we replace input image by first layer output:

$$\rho(x_0 \star \psi_{j,\theta})(u) = x_1(u, j, \theta)$$

Let $\tilde{x}_0 = R_\alpha x_0$ be a rotation of α degrees.

$$\rho(\tilde{x}_0 \star \psi_{j,\theta})(u) = x_1(R_\alpha u, j, \theta + \alpha)$$

- Similarly, roto-translation acts on x_1 by rotating and translating spatial coordinates and translating orientation coordinates

Let $\tilde{x}_0 = \varphi_{(v,\alpha)} x_0$ be a roto-translation with parameters (v, α).

$$\rho(\tilde{x}_0 \star \psi_{j,\theta})(u) = x_1(\varphi_v R_\alpha u, j, \theta + \alpha)$$

WAVELET COVARIANTS

- If we replace input image by first layer output:

$$\rho(x_0 \star \psi_{j,\theta})(u) = x_1(u, j, \theta)$$

Let $\tilde{x}_0 = R_\alpha x_0$ be a rotation of α degrees.

$$\rho(\tilde{x}_0 \star \psi_{j,\theta})(u) = x_1(R_\alpha u, j, \theta + \alpha)$$

- Similarly, roto-translation acts on x_1 by rotating and translating spatial coordinates and translating orientation coordinates

Let $\tilde{x}_0 = \varphi_{(v,\alpha)} x_0$ be a roto-translation with parameters (v, α).

$$\rho(\tilde{x}_0 \star \psi_{j,\theta})(u) = x_1(\varphi_v R_\alpha u, j, \theta + \alpha)$$

- So we can replace convolutions over translation by convolutions over roto-translations.

GROUP CONVOLUTIONS

➤

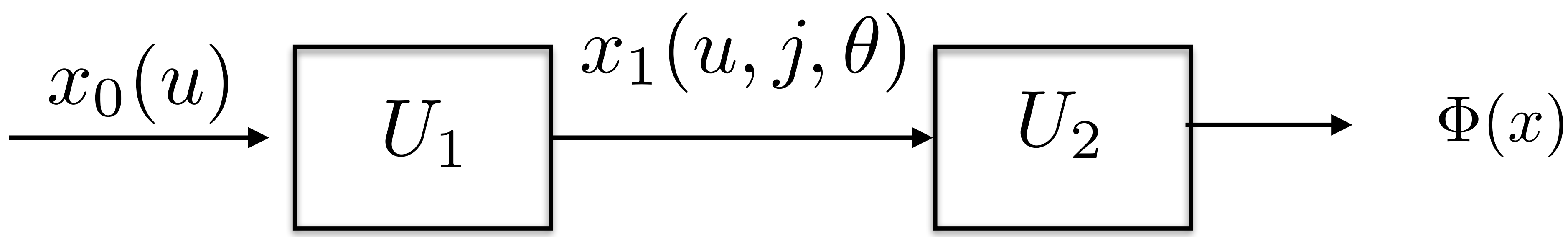
Definition: Let G be a group equipped with a Haar measure $d\mu$, acting on Ω, and $h \in L^1(G)$. The group convolution $x \star_G h$ is defined as

$$x \star_G h(u) = \int_G h(g) x(\varphi_g u) d\mu(g) \quad , \ x \in L^2(\Omega) \ .$$

If $x = x_1(u, j, \theta)$ and G are roto-translations,
these convolutions recombine different orientation channels.

JOINT SCATTERING

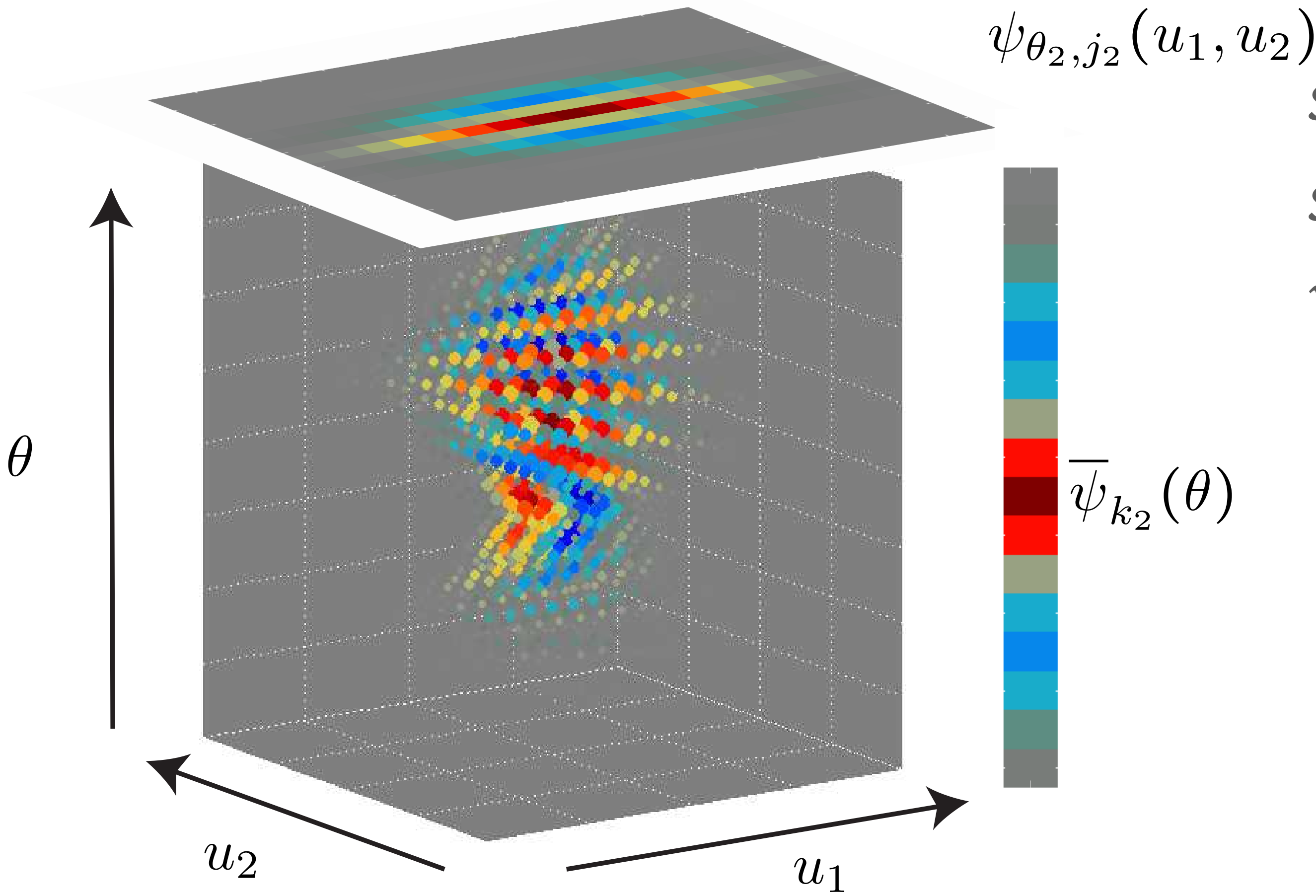
- We start by *lifting* the image with spatial wavelet convolutions: stable and covariant to roto-translations.



- We then adapt the second wavelet operator to its joint variability structure.
- More discriminability.
- Requires defining wavelets on more complicated domains:
  $\Omega = \mathbb{R}^2 \times S^2$.

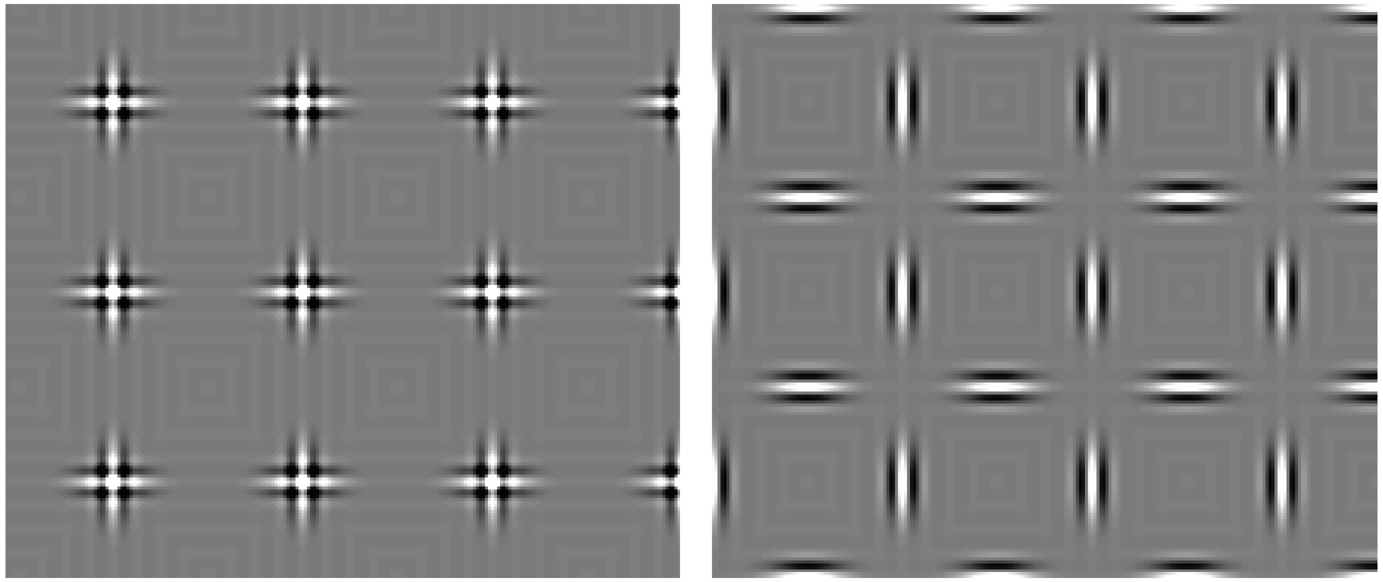
EXAMPLE: ROTO-TRANSLATION SCATTERING

- [Sifre and Mallat'13]



second layer wavelets constructed by a separable product on spatial and rotational wavelets:

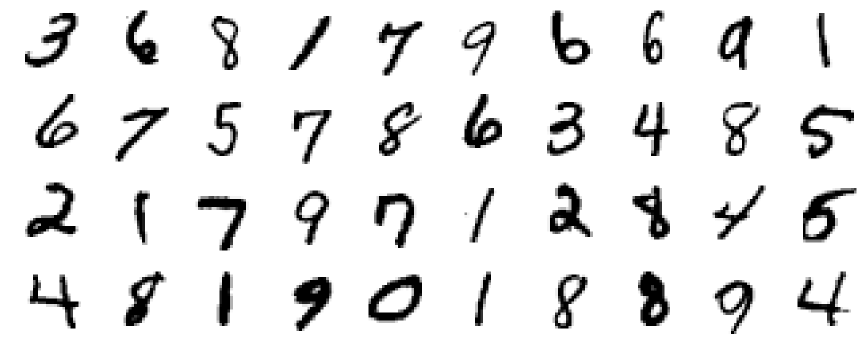
$$\Psi_\lambda(u,\theta) = \psi_{\lambda_1}(u)\psi_{\lambda_2}(\theta)$$



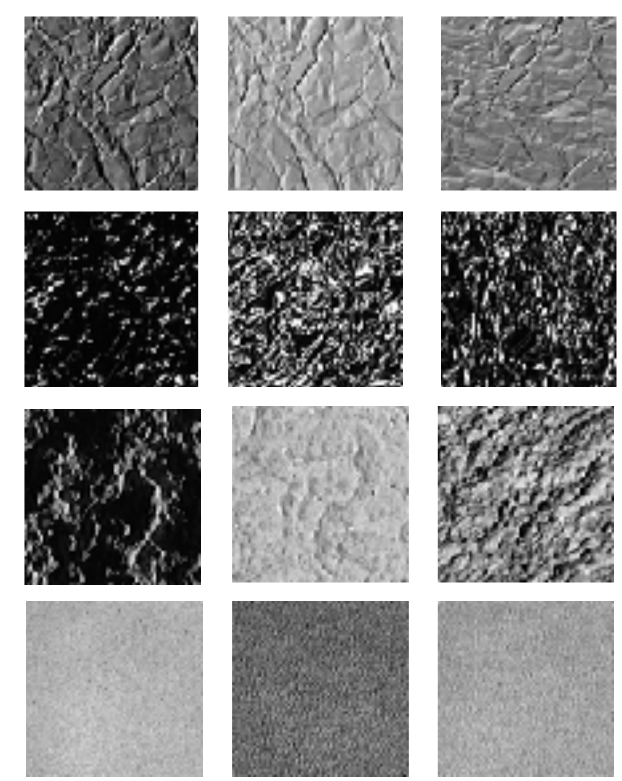
example of patterns that are discriminated by joint scattering but not with separable scattering.

CLASSIFICATION WITH SCATTERING

- State-of-the art on pattern and texture recognition using separable scattering followed by SVM:
  - MNIST, USPS [Pami’13]



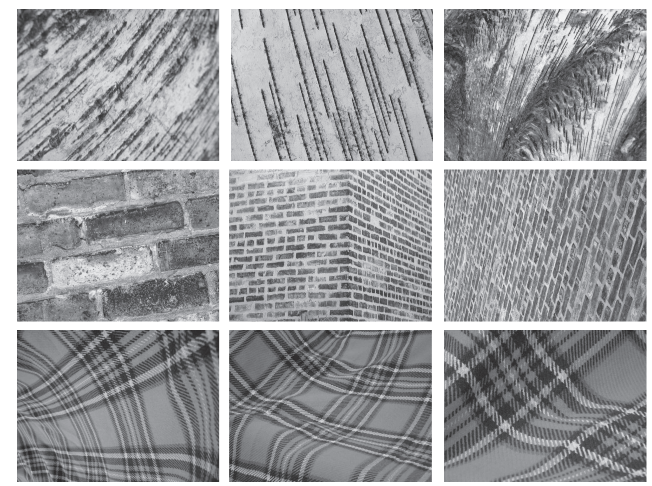
  - Texture (CUREt) [Pami’13]



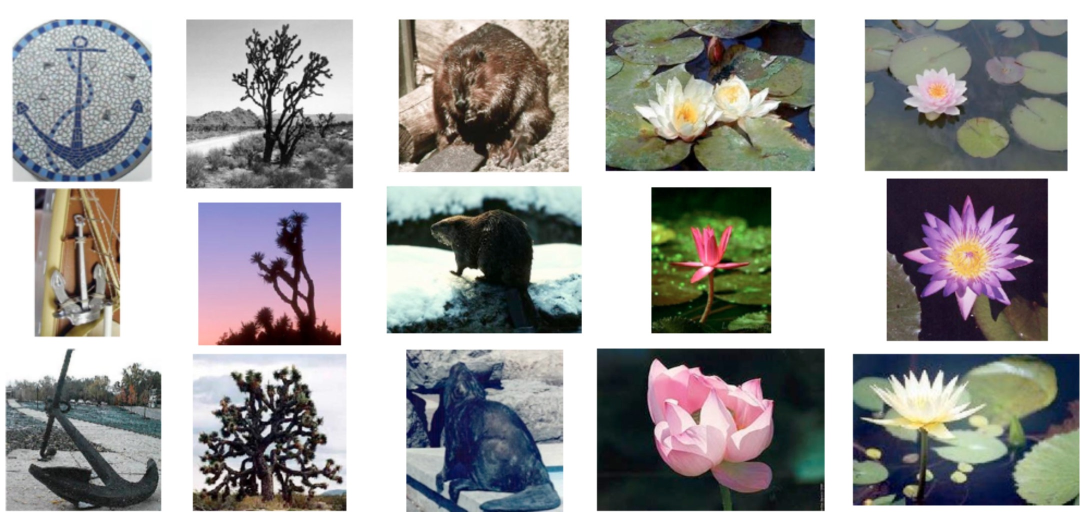
  - Music Genre Classification (GTZAN) [IEEE Acoustic ’13]

CLASSIFICATION WITH SCATTERING

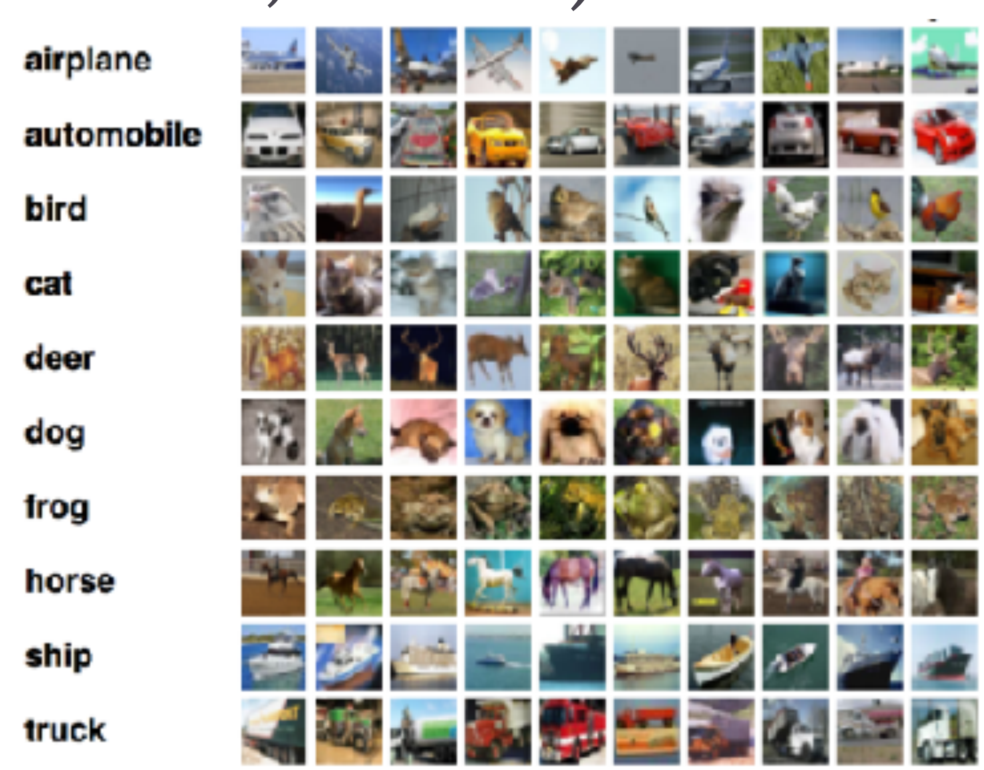
- Joint Scattering Improves Performance:
  - More complicated Texture (KTH,UIUC,UMD) [Sifre&Mallat, CVPR'13]



  - Small-mid scale Object Recognition (Caltech, CIFAR) [Oyallon&Mallat, CVPR'15]



0



LIMITATIONS OF JOINT SCATTERING

- Variability from physical world expressed in the language of transformation groups and deformations
  - However, there are not many possible groups: essentially the affine group and its subgroups.
- As a new wavelet layer is introduced, we create new coordinates, but we do not destroy existing coordinates
  - Hard to scale: dimensionality reduction is needed.
  - Wavelet design complicated beyond roto-translation groups.
- Beyond physics, many deformations are class-specific and not small.
  - Learning filters from data rather than designing them.

FROM SCATTERING TO CNNS

- Given $x(u, \lambda)$ and a group G acting on both u and λ, we defined wavelet convolutions over G as

$$x \star_G \psi_{\lambda'}(u, \lambda) = \int_v \int_\alpha \psi_\lambda(R_{-\alpha}(u - v)) x(v, \alpha) dv d\alpha$$

- In discrete coordinates,

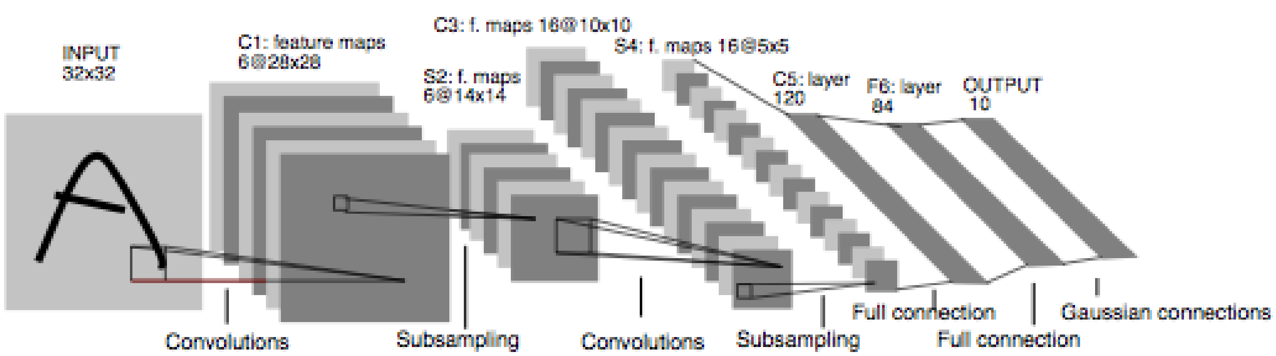
$$x \star_G \psi_{\lambda'}(u, \lambda) = \sum_v \sum_\alpha \overline{\psi}_{\lambda'}(u - v, \alpha, \lambda) x(v, \alpha)$$

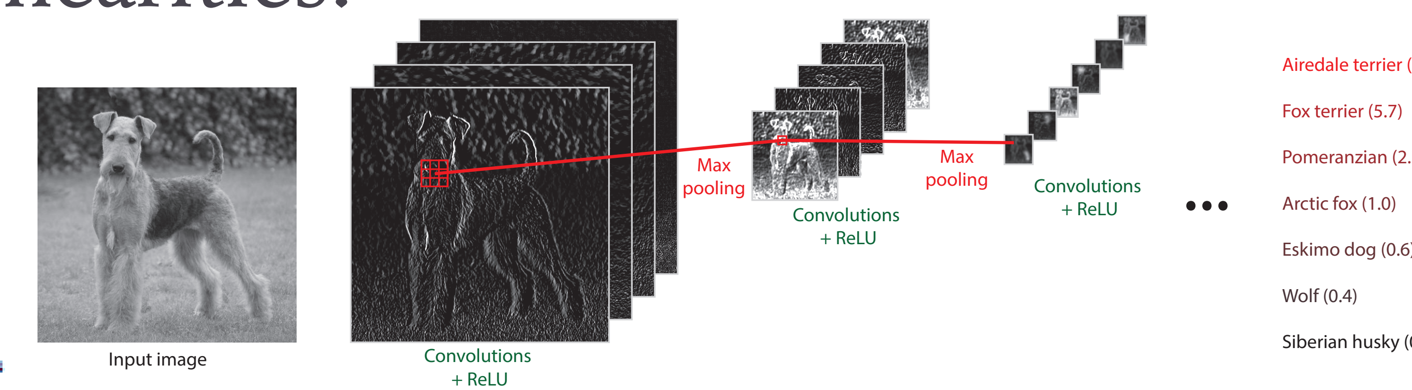
- Which in general is a convolutional tensor.

CONVOLUTIONAL NEURAL NETWORKS

[LeCun, 80s,90s]

- Stack multiple layers of **localized** convolutional operators and point-wise contractive non-linearities:





Input: $x \in L^2(\Omega, \mathbb{R}^p)$. Output: $\tilde{x} \in L^2(\Omega, \mathbb{R}^{\tilde{p}})$.

$$\tilde{x}_{\tilde{j}}(u) = \rho\left(\sum_{j=1}^{p} x_j \star \theta_{j,\tilde{j}}(u)\right) \, , \; \tilde{j} \leq \tilde{p} \, .$$

$\rho(z)$: point-wise nonlinearity (e.g. $\max(0, z)$).

$\Theta = (\theta_{j,\tilde{j}})$: localized convolutional kernel.

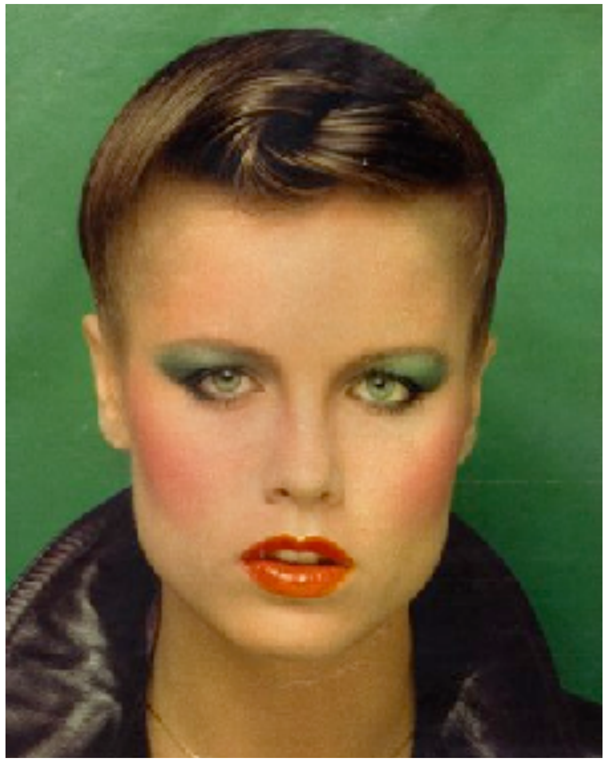
- Down-sampling via *pooling* (can be either linear with average, or nonlinear with max) in invariant tasks:

$$\bar{x}_{\tilde{j}}(\bar{u}) = \|\tilde{x}_{\tilde{j}}(\mathcal{N}(u))\|$$

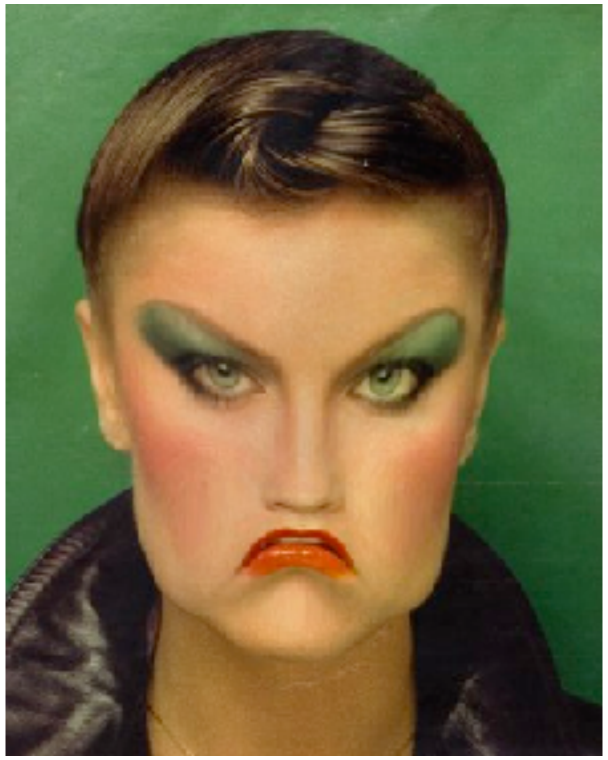
$\mathcal{N}(u)$: Neighborhood of u.

CONVOLUTIONAL NEURAL NETWORKS

- Why are CNNs geometrically stable?



$x(u)$

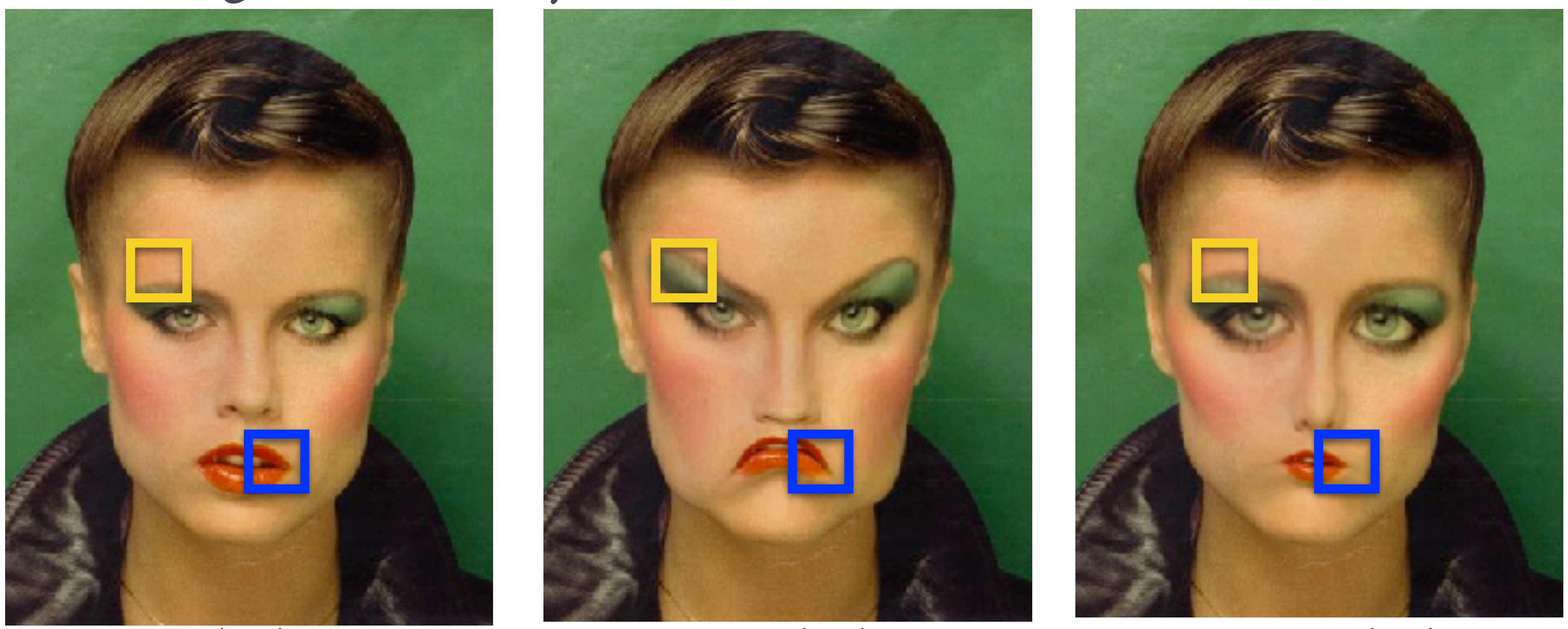


$x_\tau(u)$

$x_{\tau'}(u)$

CONVOLUTIONAL NEURAL NETWORKS

- Why are CNNs geometrically stable?



$x(u)$ $\quad$ $x_\tau(u)$ $\quad$ $x_{\tau'}(u)$

- A non-rigid deformation locally looks like a translation if $\|\nabla\tau\|$ small:

$$\Rightarrow x_\tau \star \theta(u) \approx [x \star \theta]_\tau(u)$$

- A point-wise nonlinearity commutes with deformations:

$$\Rightarrow \rho\left(x_\tau \star \theta(u)\right) \approx \rho\left([x \star \theta]_\tau (u)\right) = [\rho\left(x \star \theta\right)]_\tau (u)$$

- Pooling progressively creates invariance to geometric deformations:

$$\|x_\tau(\mathcal{N}(u))\| \approx \|x(\mathcal{N}(u))\| \text{ if } |\tau| \text{small}$$

CONVOLUTIONAL NEURAL NETWORKS

- **Convolutions** to exploit translation invariance/equivariance.
- **Localized** to exploit geometric stability: leads to multi scale architecture.
- These two properties lead to models with $O(\log N)$ trainable parameters.

- Stability is only part of the story. Discriminability via learning/optimization is another major component for success.

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

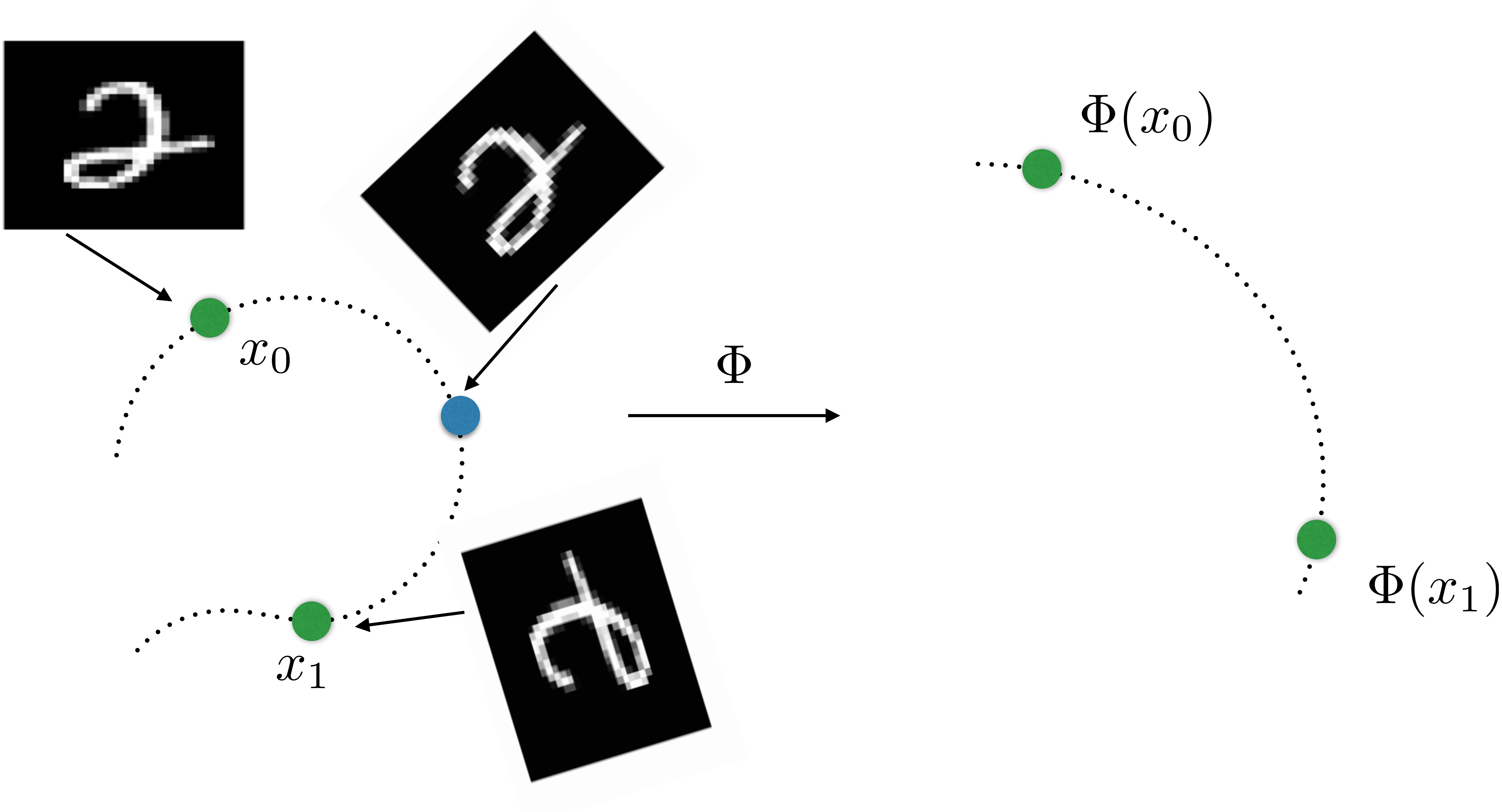
- We related stability with the ability to linearize deformations:

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

- 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]

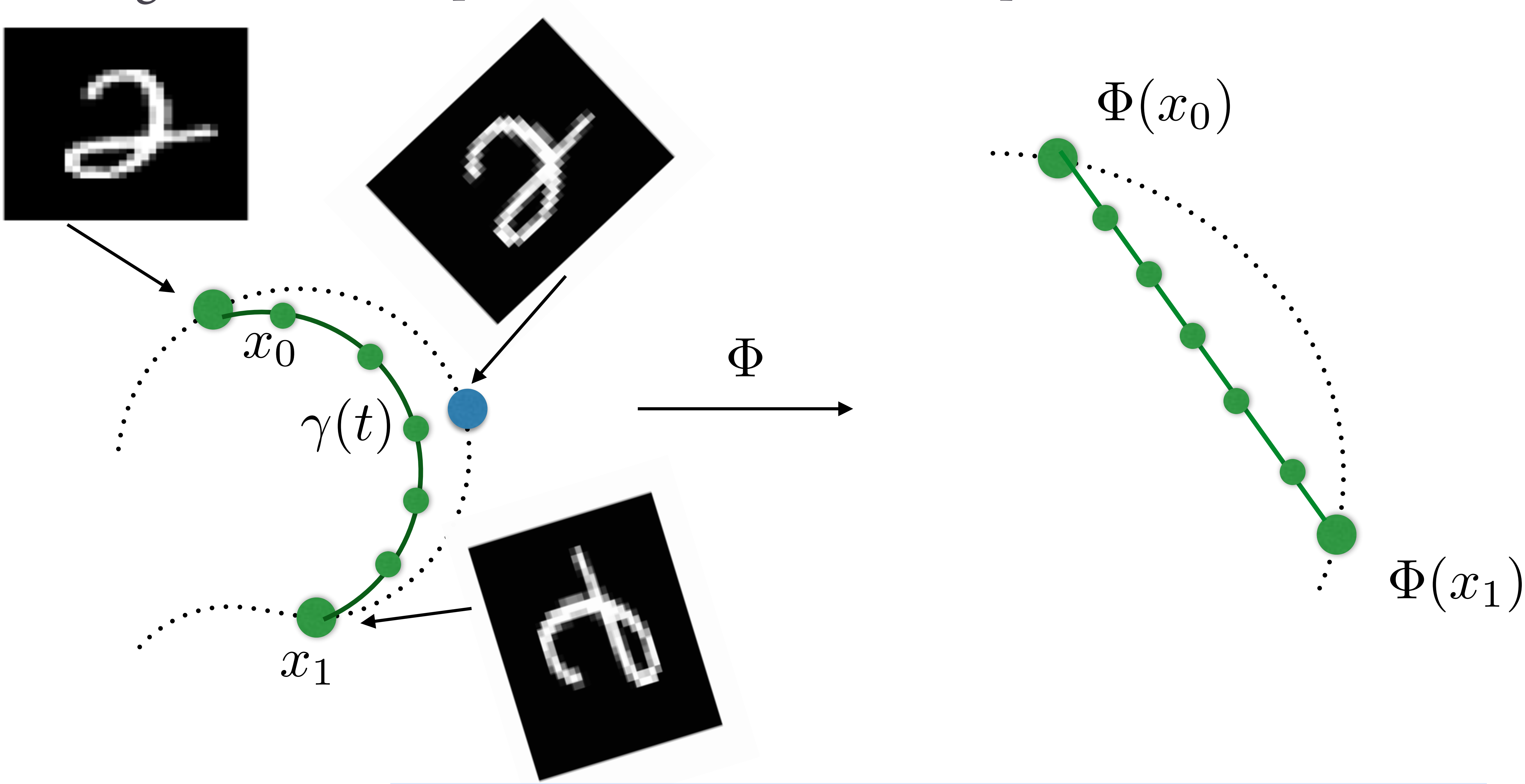
INVARIANCE, LINEARIZATION AND GEODESICS

➤ Algorithm from [Henaff & Simoncelli '16]:



INVARIANCE, LINEARIZATION AND GEODESICS

- Algorithm from [Henaff & Simoncelli '16]:



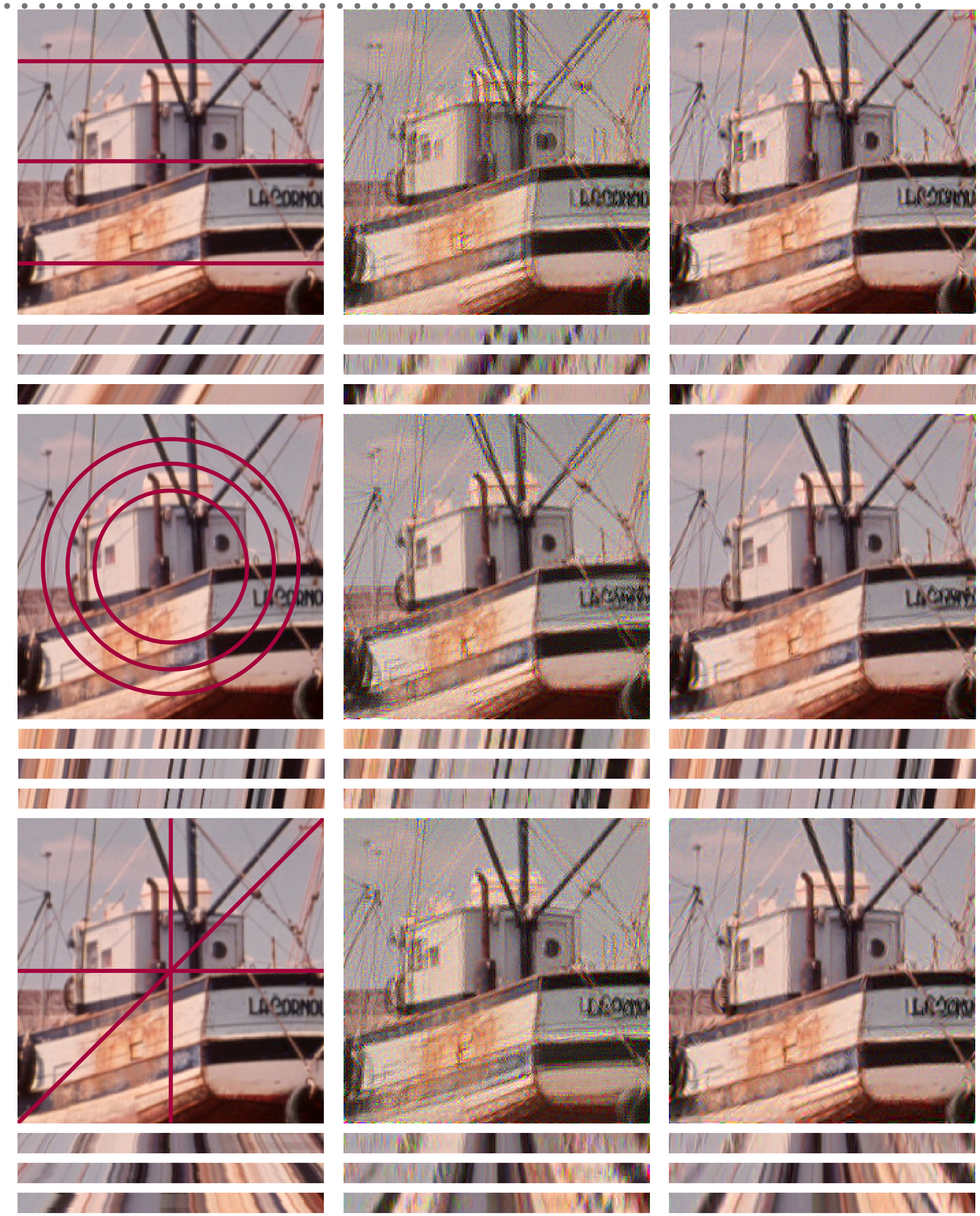
$$\min_{\gamma(0)=x_0,\gamma(1)=x_1} \int_0^1 |\dot{\gamma}(t)|dt + \int_0^1 |(\dot{\Phi\gamma})(t)|dt$$

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

[Henaff and Simoncelli'16]



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

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 IN CNNS

- “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]) $\tilde{\Phi}(x) = \Phi(x, \tilde{\Theta})$, $\tilde{\Theta}_i = F(\beta_i)$
  - Typically we restrict the new class to be $\dim(\beta_i) \ll \dim(\Theta_i)$
  - Explore low-rank tensor factorizations of each convolutional tensor.

REDUNDANCY

- "Post-training" model compression:

  Given parameters $\Theta = (\Theta_1, \ldots, \Theta_k)$, 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

    $$\tilde{\Phi}(x) = \Phi(x, \tilde{\Theta}) \ , \ \tilde{\Theta}_i = F(\beta_i) \qquad \dim(\beta_i) \ll \dim(\Theta_i)$$

  - 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: NO TRAINING AND NO STRUCTURE

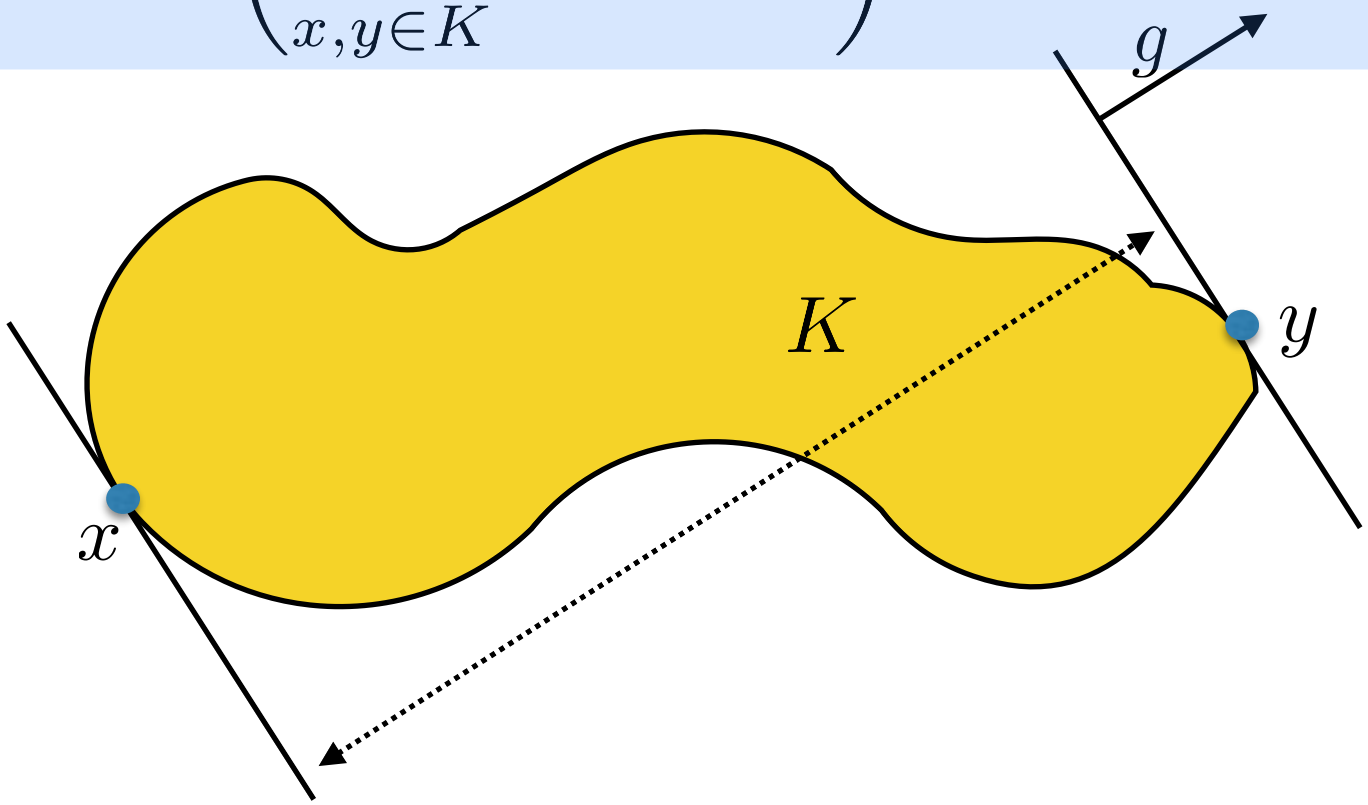
- [Giryes, Sapiro and Bronstein,'15] $\Phi = \text{Random Convnet}$

INVERTIBILITY: NO TRAINING AND NO STRUCTURE

- [Giryes, Sapiro and Bronstein,'15] $\Phi = \text{Random Convnet}$

Gaussian mean width of a set K:

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

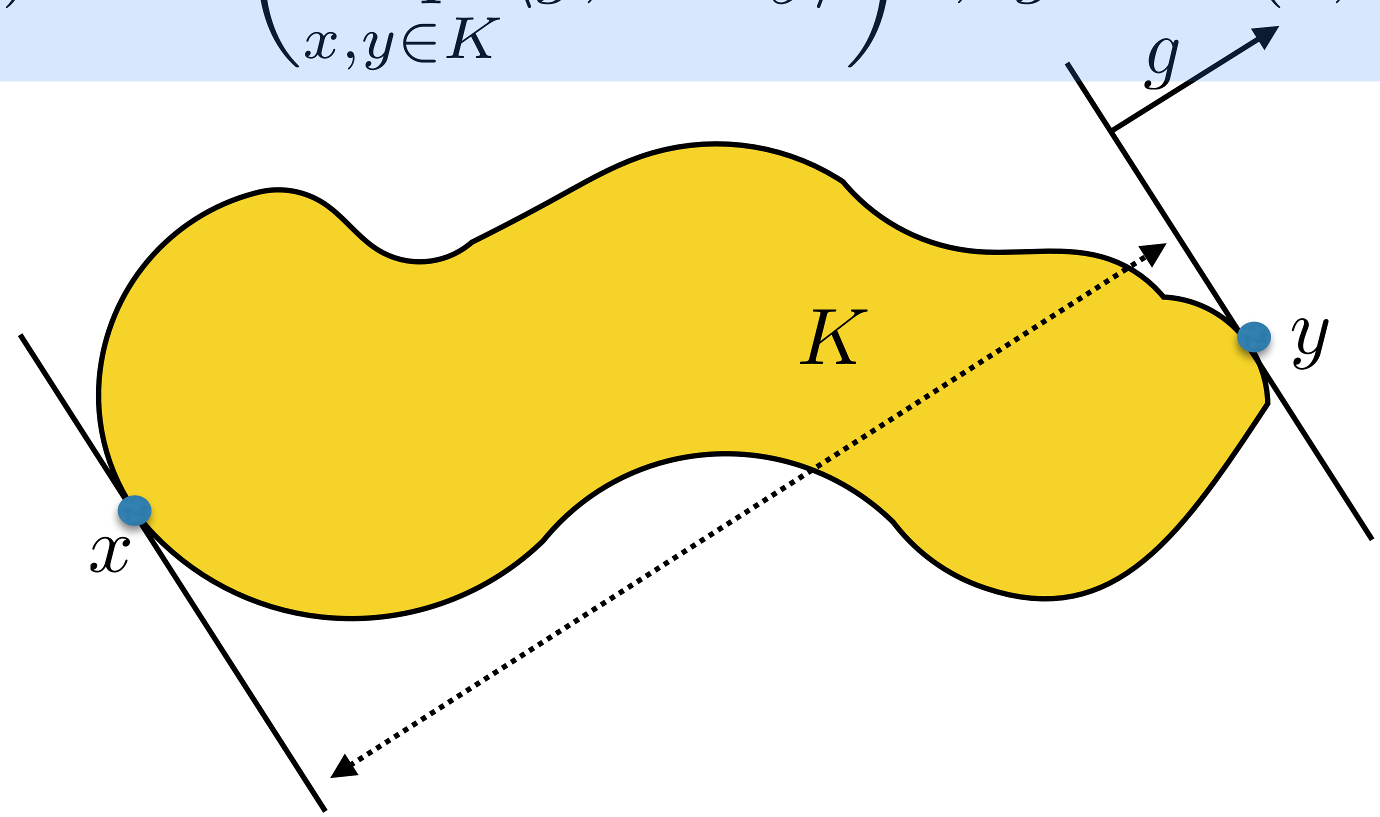


INVERTIBILITY: NO TRAINING AND NO STRUCTURE

➤ [Giryes, Sapiro and Bronstein,'15] $\Phi = \text{Random Convnet}$

Gaussian mean width of a set K:

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



➤ 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

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

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

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

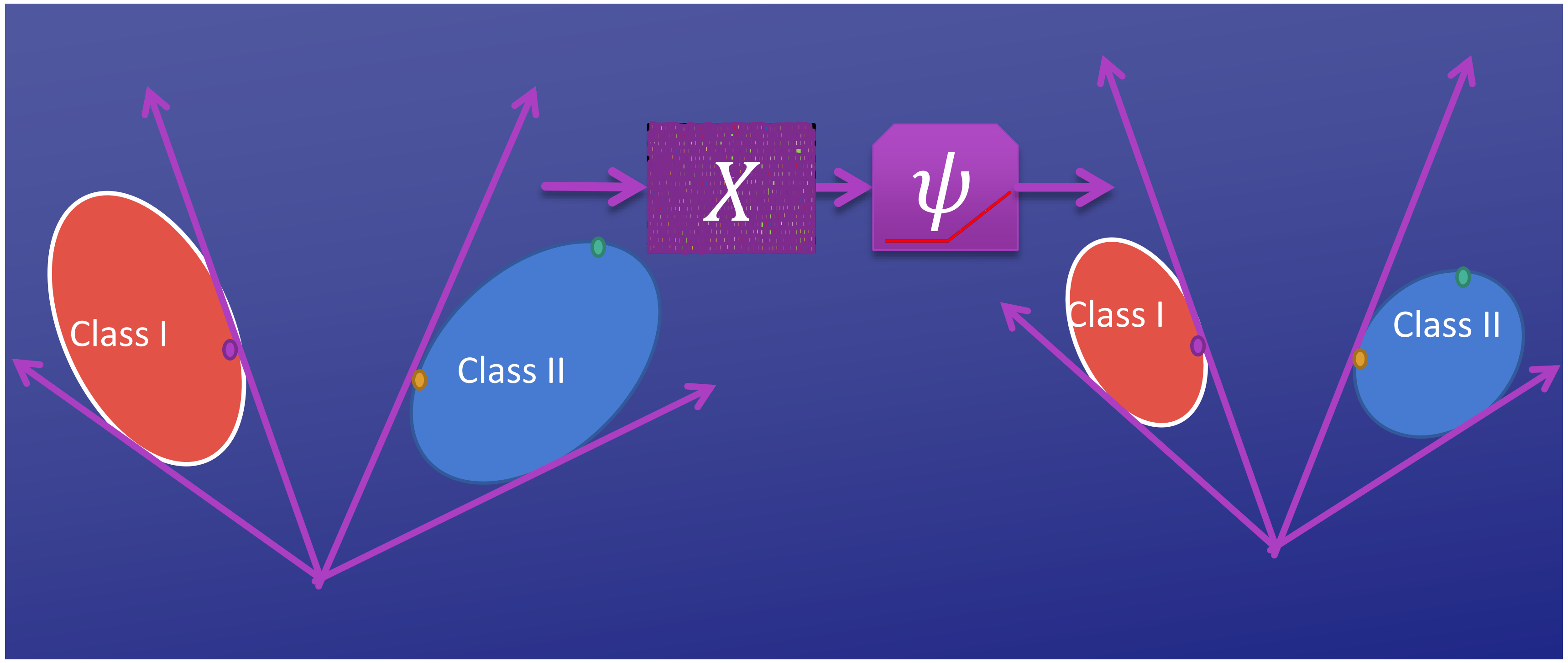
INTERPRETATION

- 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.



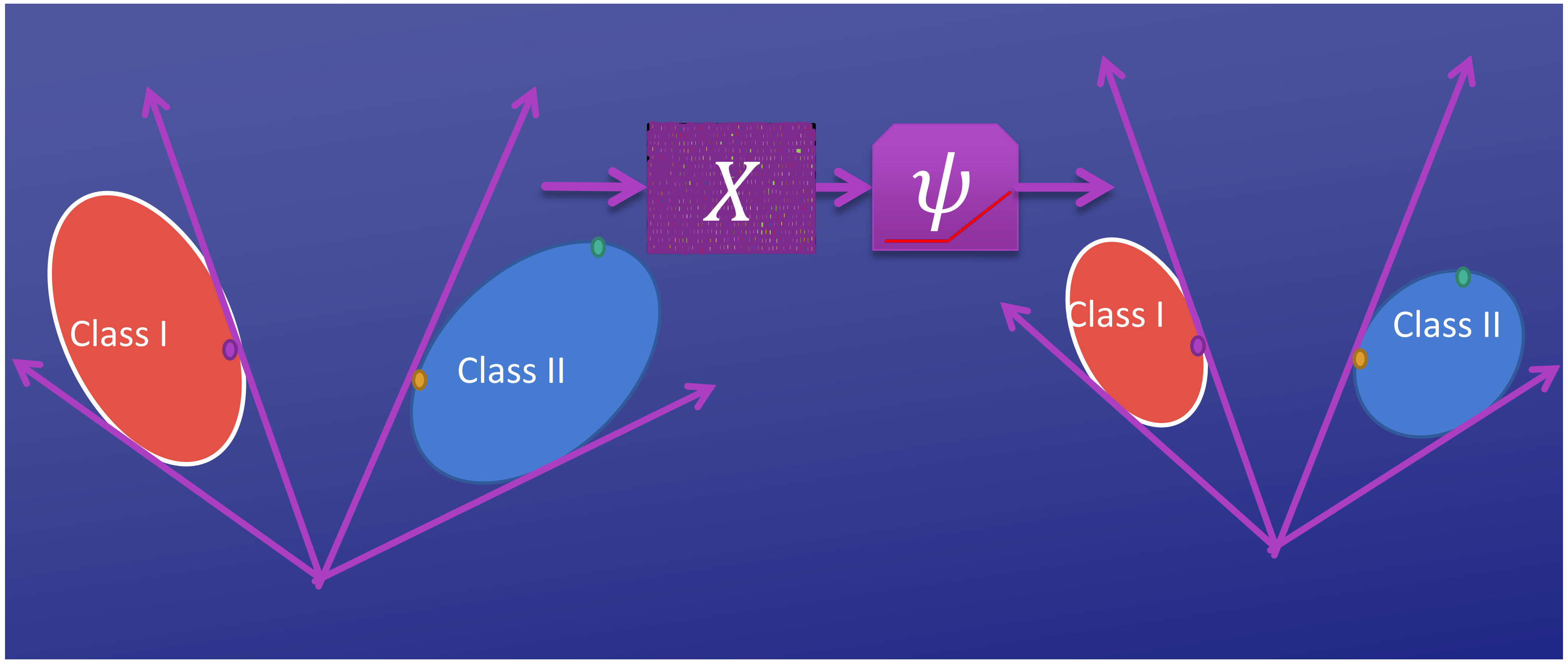
[Raja Giryes]

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]

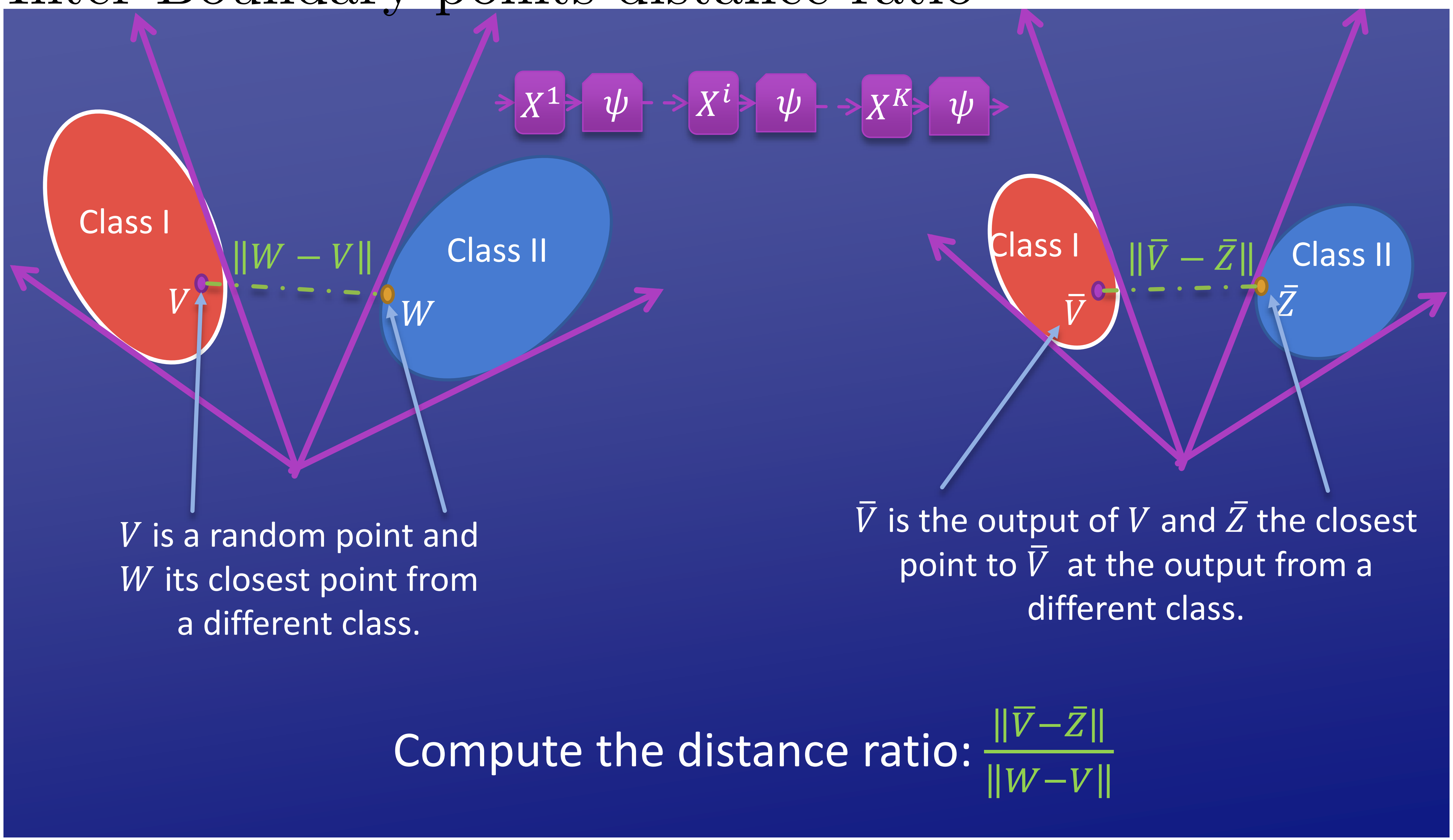
points with small angles between them become closer than points with larger angles between them

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

ROLE OF TRAINING?

ROLE OF TRAINING?

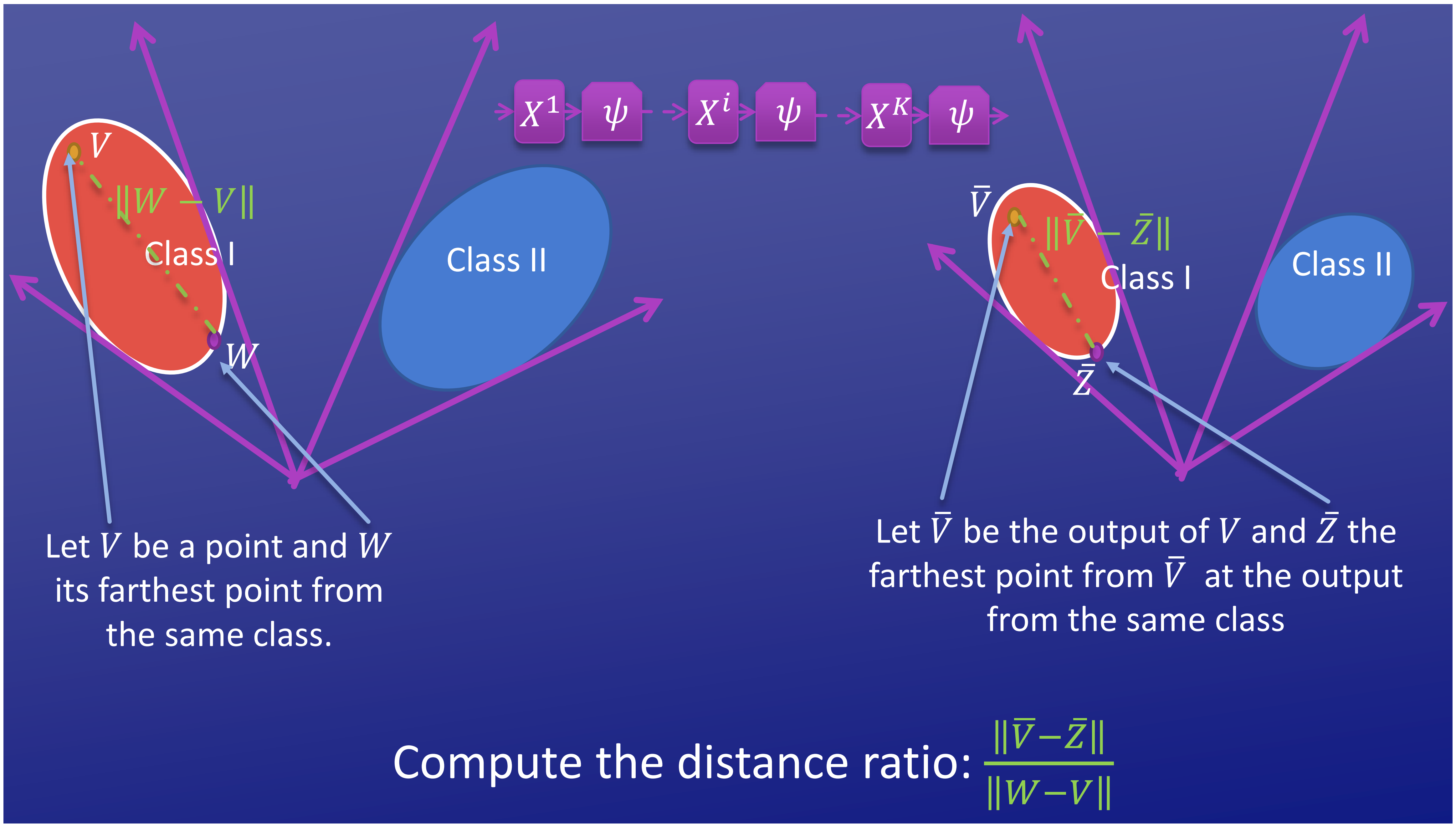
Inter Boundary points distance ratio



[Raja Giryes]

ROLE OF TRAINING?

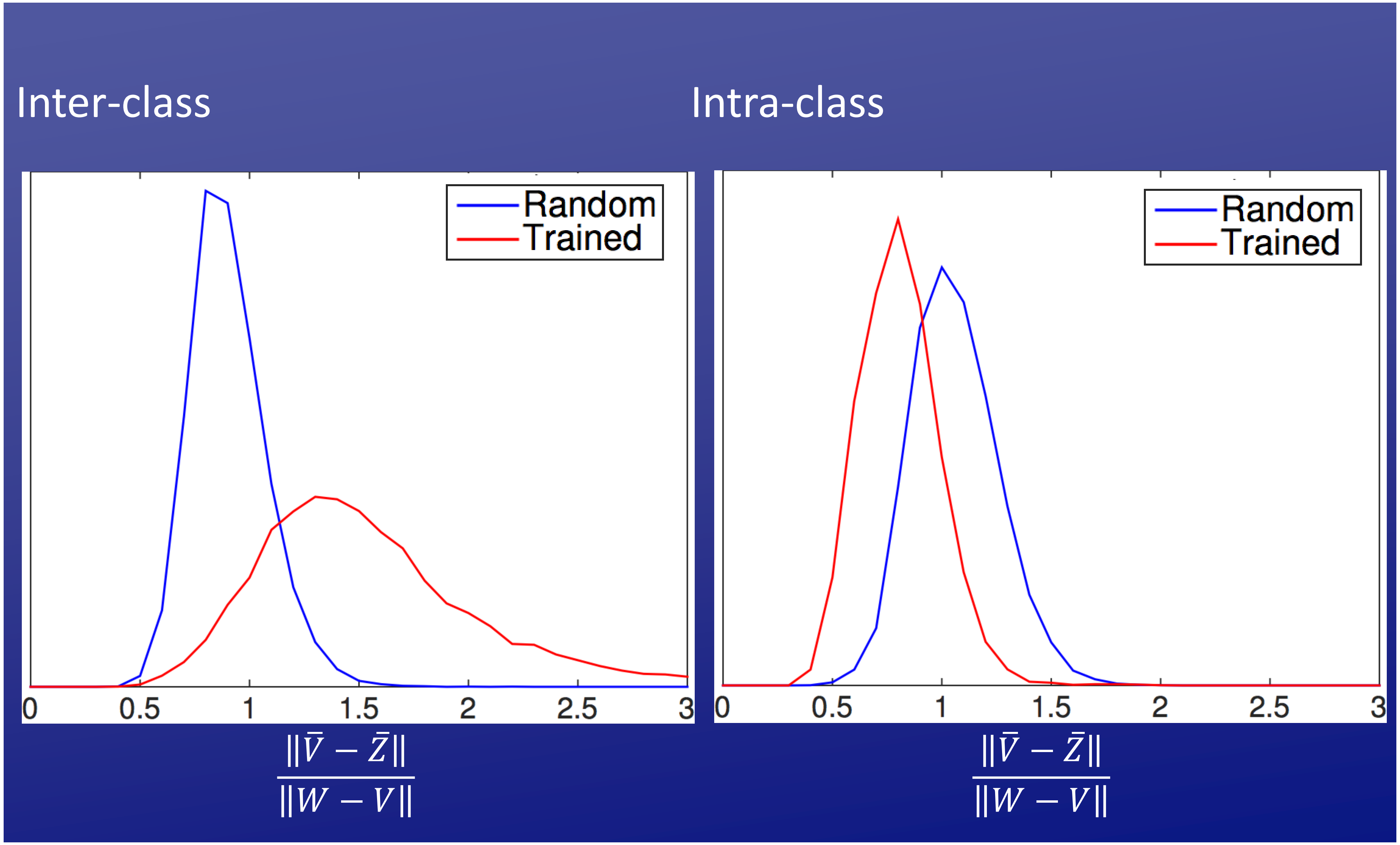
Intra Boundary points distance ratio



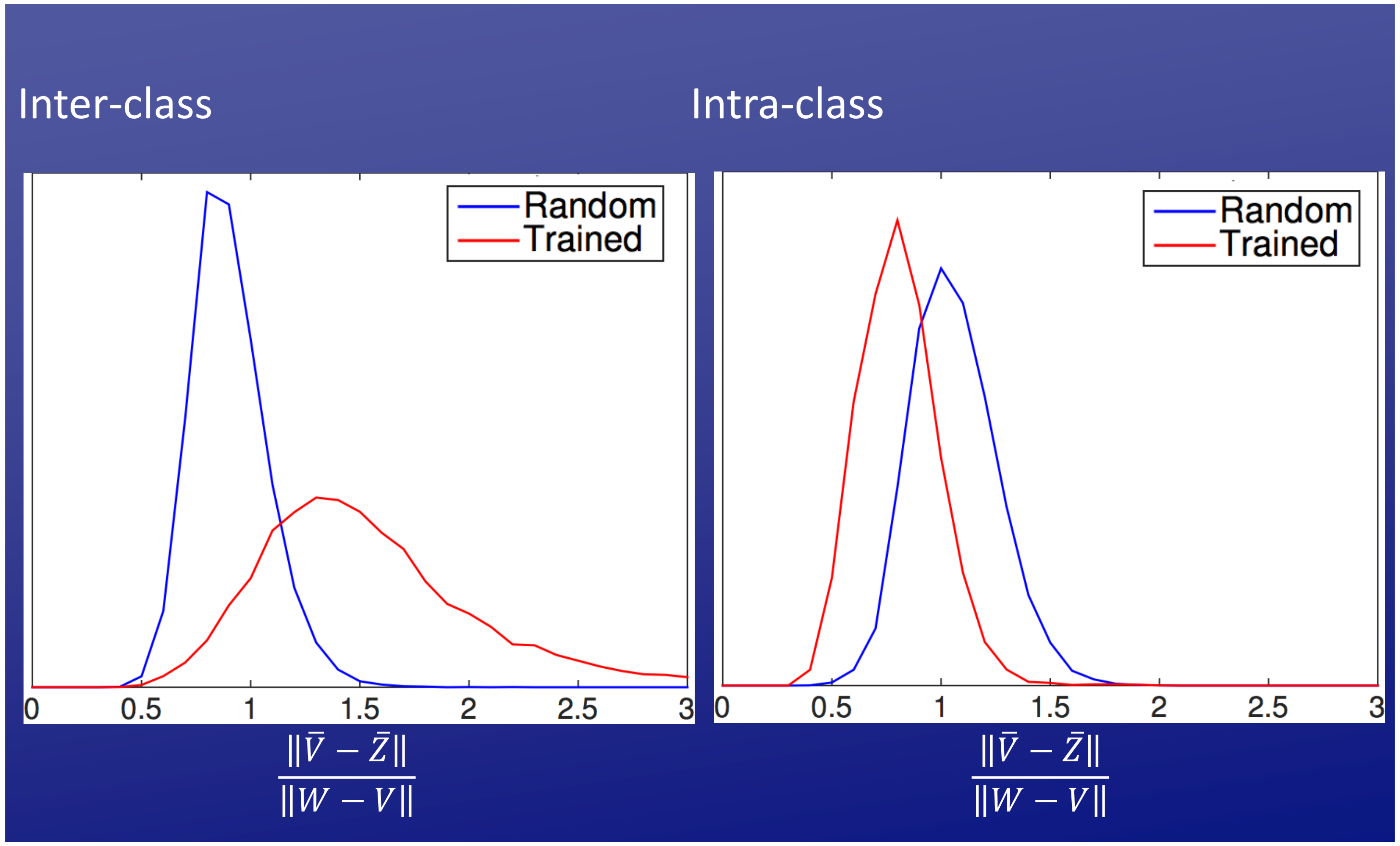
[Raja Giryes]

ROLE OF TRAINING?

Boundary distance ratios
measured on Imagenet using VGG oxfordnet



ROLE OF TRAINING?

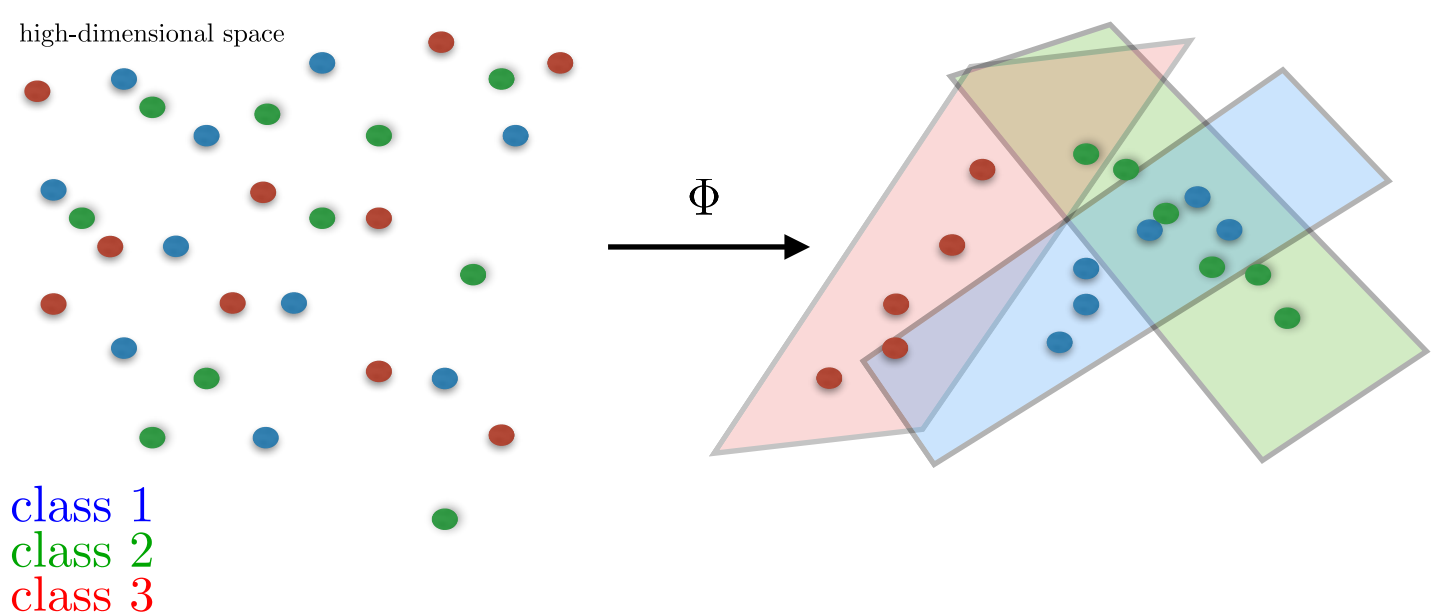


[Raja Giryes]

- 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 WITH TRAINING

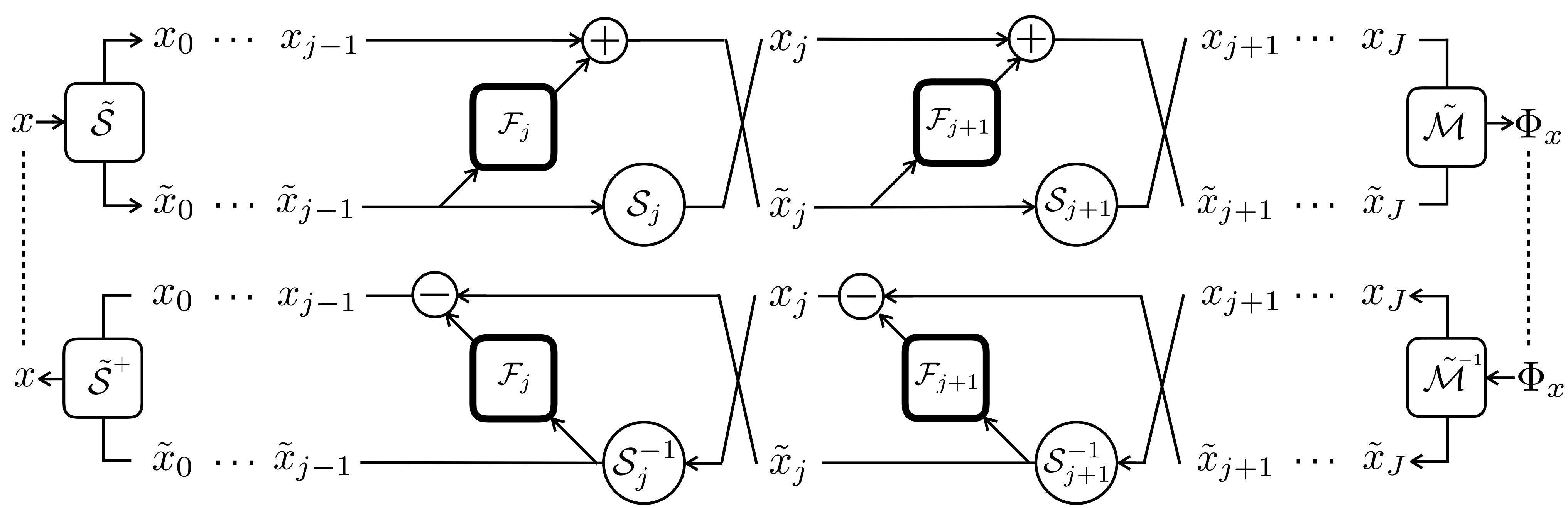
- Hypothesis: Deep CNNs progressively "linearize" intra-class variability. $\hat{f}(x) = \langle \Phi(x), w \rangle$
- Hypothesis: Once it is "linearized", it can be "killed" by linear projections that contract the space:
$\Phi(x) = \rho W_L \rho W_{L-1} \cdots W_1 x$



- Does this imply that the network loses information?
  - As implied by "Information Bottleneck", Tibshy et al.

INVERTIBILITY WITH TRAINING

- iRevNets [Jacobsen, Smeulders, Oyallon, ICLR'18] construct provably invertible CNN representations for large-scale image classification.

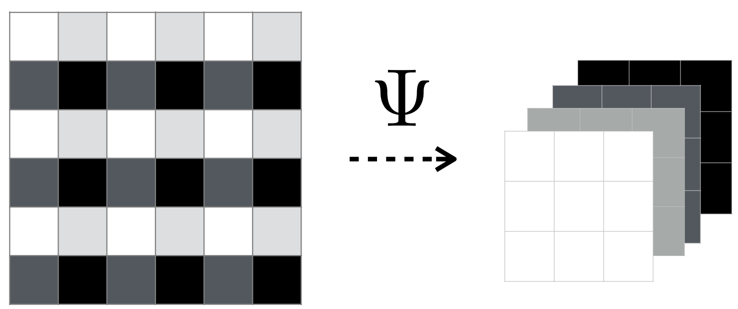


- $\mathcal{S}_j$: linear *invertible* reshuffling operator.
  $\mathcal{F}_j$: non-linear convolutional net.

- Same as residual networks, but downsampling is made invertible.

INVERTIBILITY WITH TRAINING

- Invertible down-sampling operator:



$$\mathcal{S}_j x(u,\lambda) := x(\Psi(u,\lambda)) \ .$$

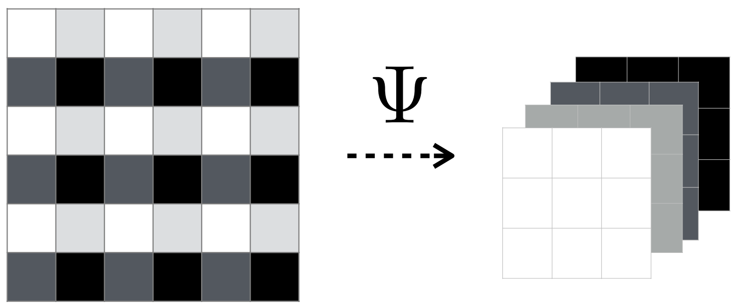
- The operator is certified invertible:

$$\begin{cases} x_{j+1} = \mathcal{S}_{j+1}\tilde{x}_j \\ \tilde{x}_{j+1} = x_j + \mathcal{F}_{j+1}\tilde{x}_j \end{cases} \iff \begin{cases} \tilde{x}_j = \mathcal{S}_{j+1}^{-1} x_{j+1} \\ x_j = \tilde{x}_{j+1} - \mathcal{F}_{j+1}\tilde{x}_j \end{cases}$$

  - Related to wavelet *lifting* schemes [Sweldens'98]

INVERTIBILITY WITH TRAINING

- Invertible down-sampling operator:



$$\mathcal{S}_j x(u,\lambda) := x(\Psi(u,\lambda)) \ .$$

- The operator is certified invertible:

$$\begin{cases} x_{j+1} = \mathcal{S}_{j+1}\tilde{x}_j \\ \tilde{x}_{j+1} = x_j + \mathcal{F}_{j+1}\tilde{x}_j \end{cases} \iff \begin{cases} \tilde{x}_j = \mathcal{S}_{j+1}^{-1} x_{j+1} \\ x_j = \tilde{x}_{j+1} - \mathcal{F}_{j+1}\tilde{x}_j \end{cases}$$

  - Related to wavelet *lifting* schemes [Sweldens'98]

- Classification performance is preserved despite invertibility:

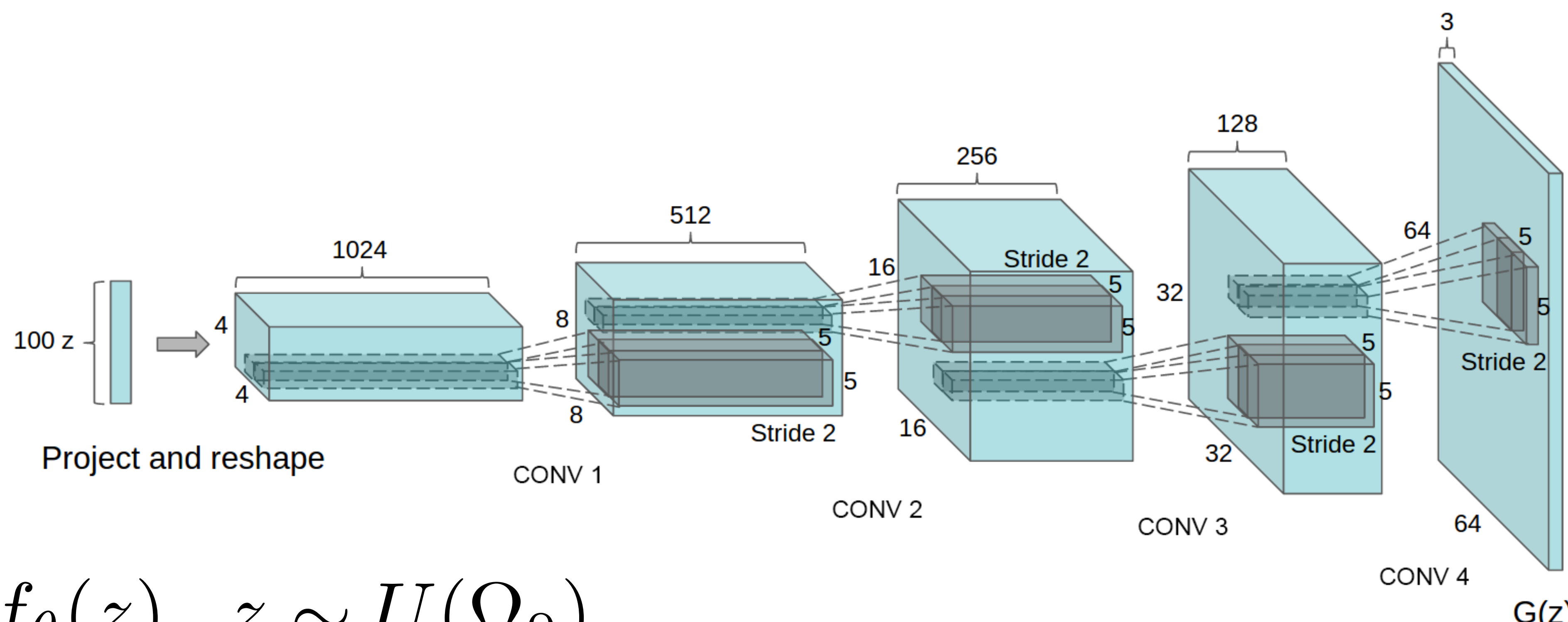
Architecture	**Injective**	**Bijective**	**Top-1 error**	**Parameters**
ResNet	-	-	24.7	26M
RevNet	-	-	25.2	28M
i-RevNet (a)	yes	-	24.7	181M
i-RevNet (b)	yes	yes	26.7	29M

DEEP IMAGE PRIOR [ULYANOV ET AL '17]

- How much prior information is captured by the choice of convolutional architecture?

DEEP IMAGE PRIOR [ULYANOV ET AL '17]

- How much prior information is captured by the choice of convolutional architecture?
- We will see next week that CNNs can be used also as *generators*:

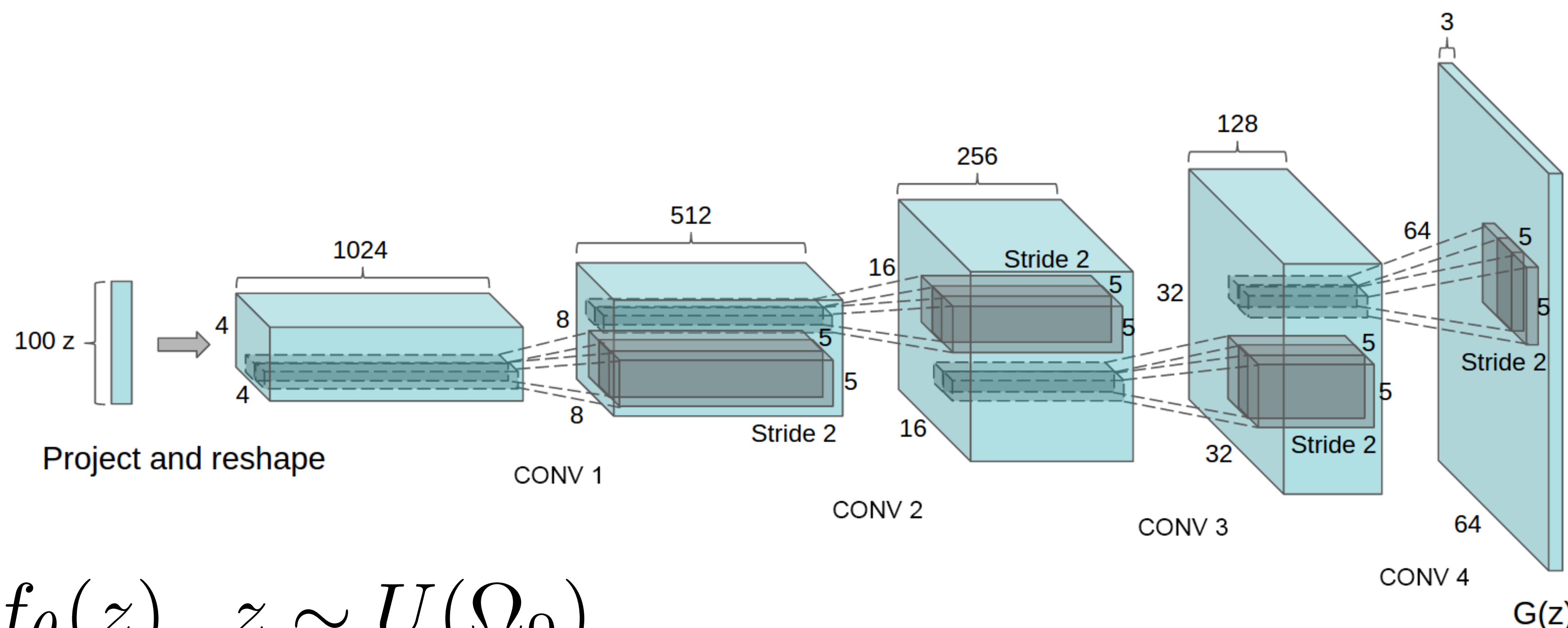


$$x = f_\theta(z) \ , z \sim U(\Omega_0) \ .$$

DC-Gan Architecture [Radford et al'16]

DEEP IMAGE PRIOR [ULYANOV ET AL '17]

- How much prior information is captured by the choice of convolutional architecture?
- We will see next week that CNNs can be used also as *generators*:



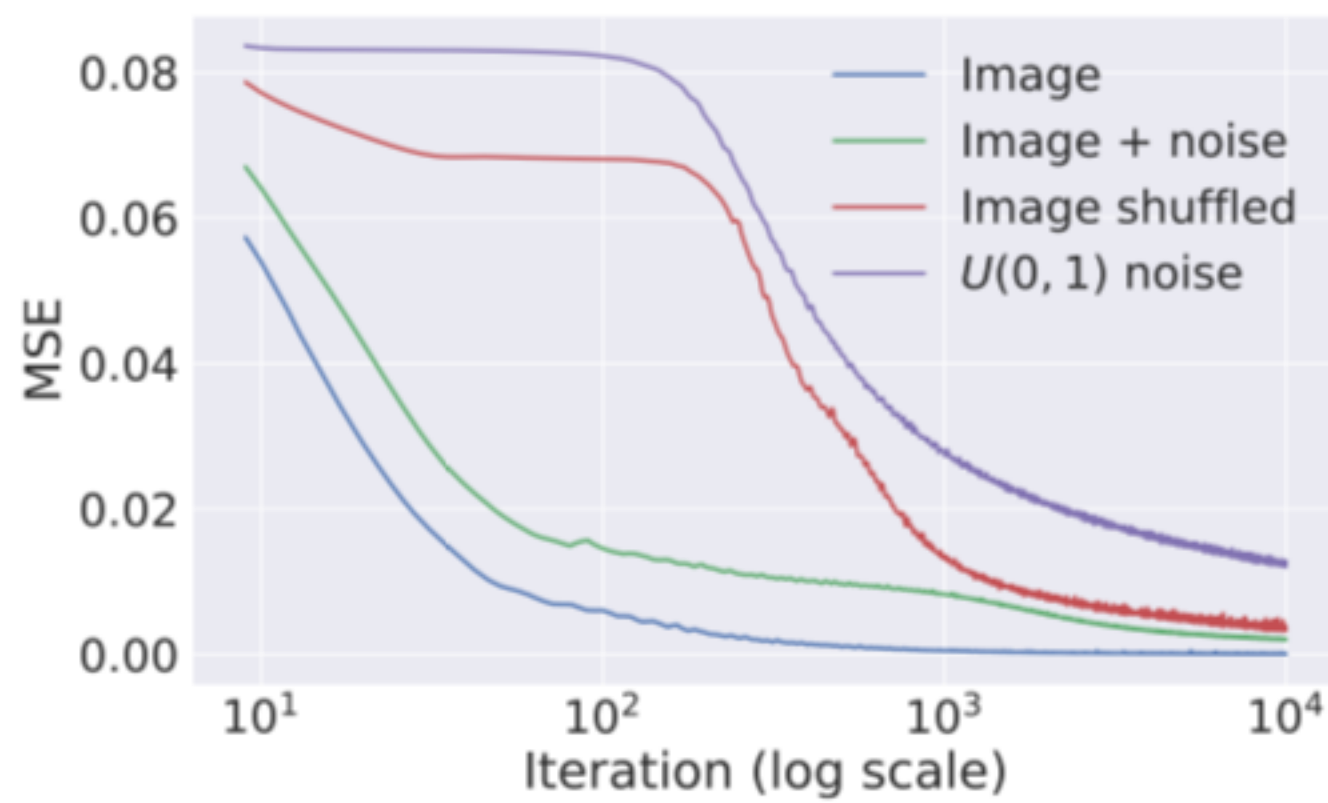
$$x = f_\theta(z) \ , z \sim U(\Omega_0) \ .$$

DC-Gan Architecture [Radford et al'16]

- Which $x \in L^2(\Omega)$ are well-approximated by such CNNs?

DEEP IMAGE PRIOR [ULYANOV ET AL '17]

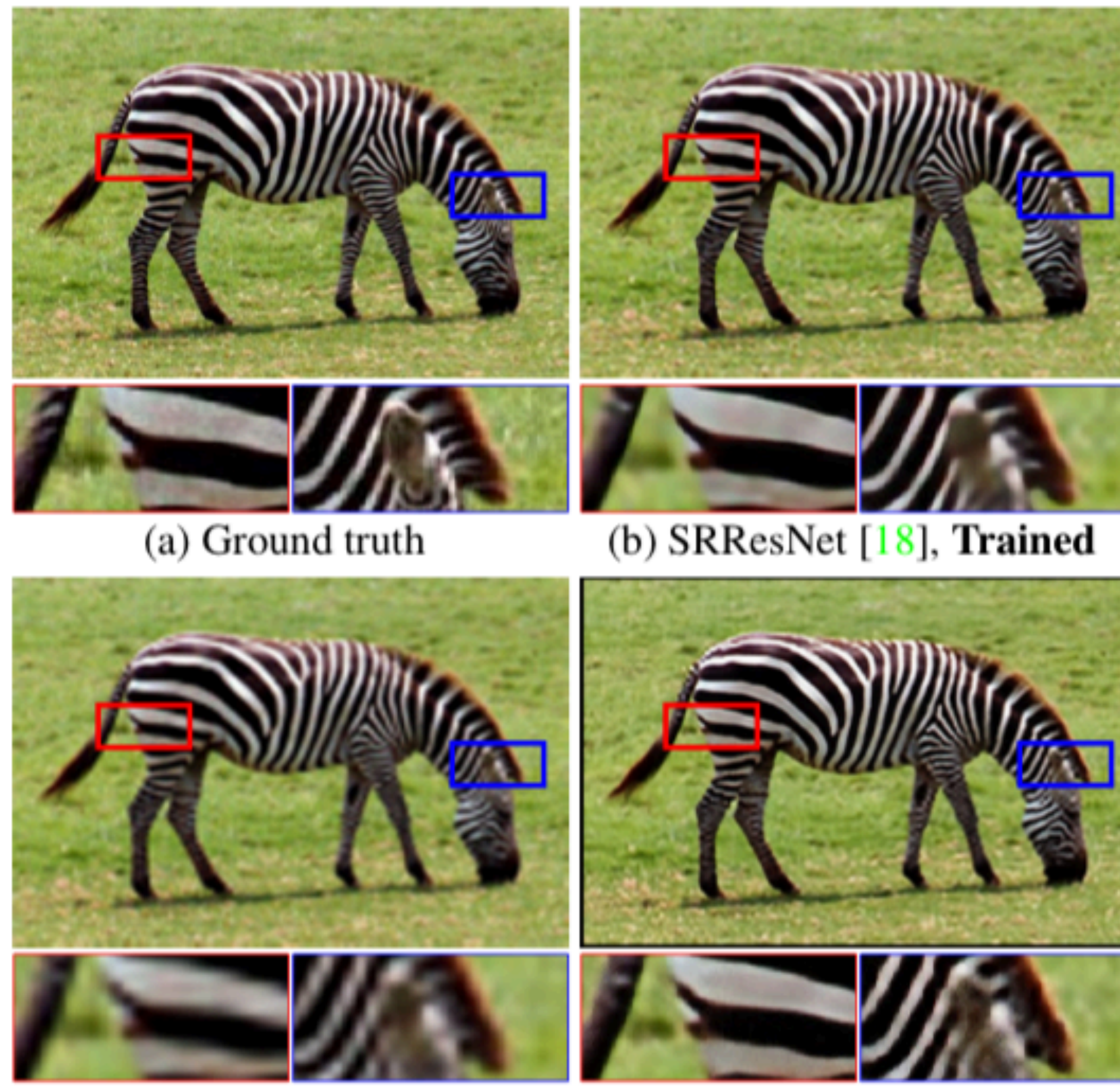
- Given $x \in L^2(\Omega)$, fix a sample $z \sim U$ and consider the optimization problem $\min_{\theta} \ell(x, f_\theta(z))$
- Remarkably, *natural* images are easier to approximate than "non-natural" images:



DEEP IMAGE PRIOR [ULYANOV ET AL '17]

$$\hat{x} = f_{\theta^*}(z)\ ,\ \theta^* \in \arg\min_{\theta} \|\Gamma f_\theta(z) - x_0\|^2$$

- Applications to imaging inverse problems *without learning*.
- More on that next week.

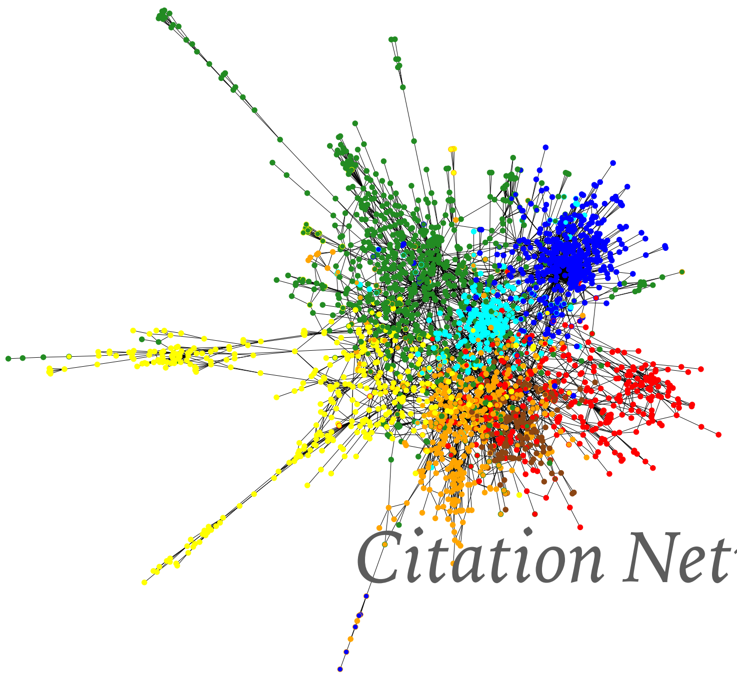


(a) Ground truth (b) SRResNet [18], **Trained**

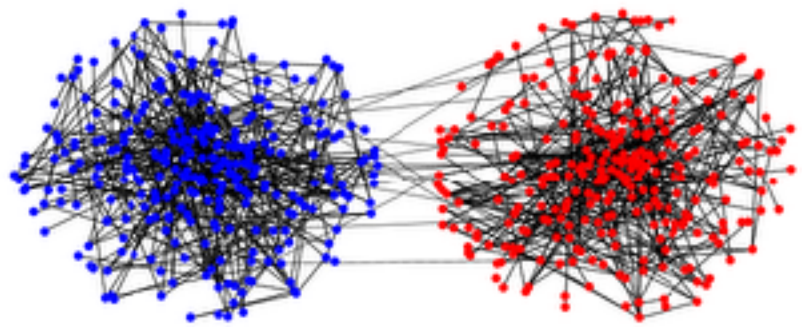
(c) Bicubic, **Not trained** (d) Deep prior, **Not trained**

TOWARDS NON-EUCLIDEAN GEOMETRIES

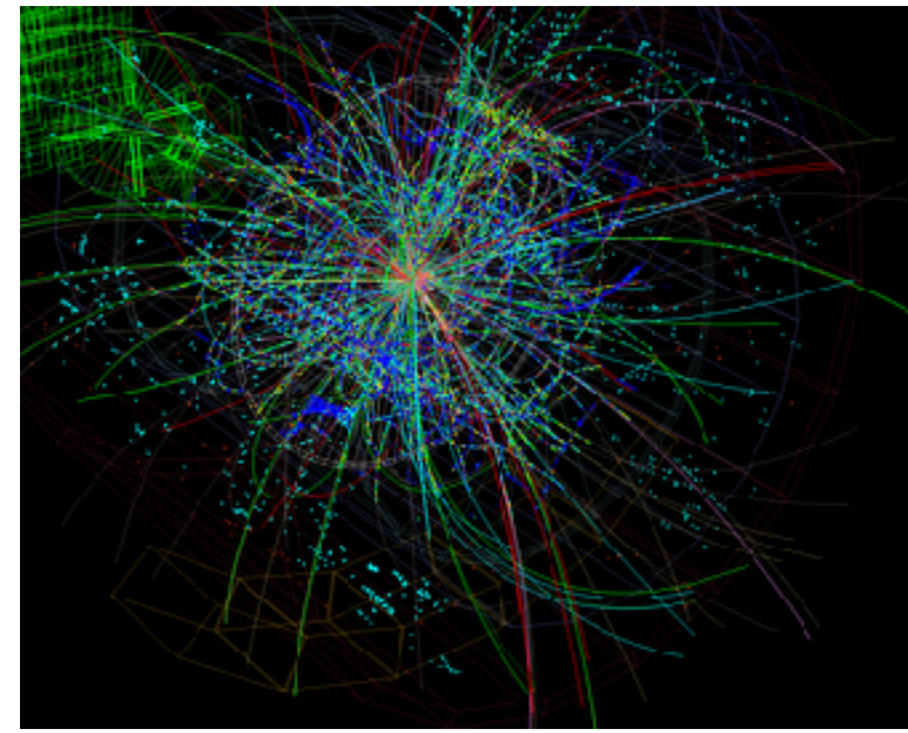
- How about problems/tasks defined over more general domains?



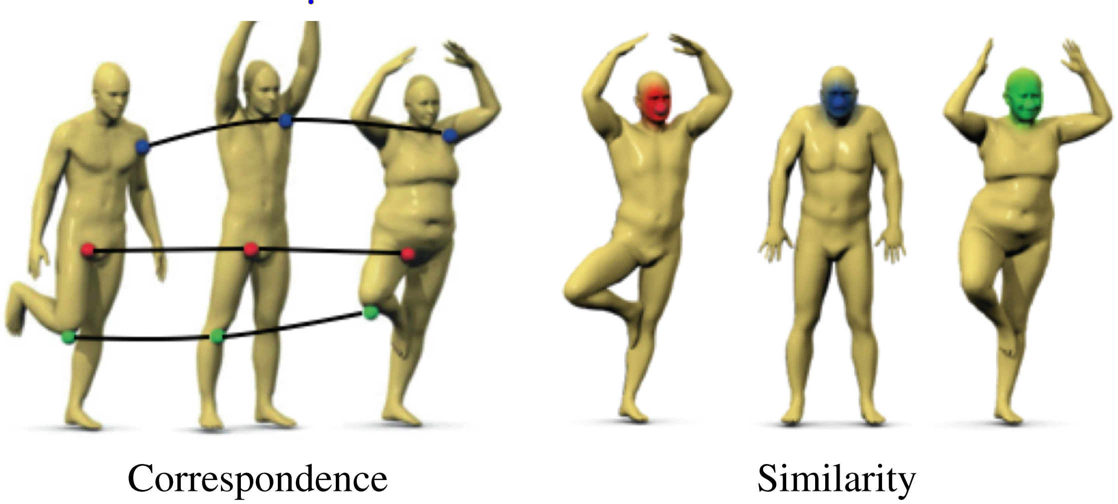
Citation Networks



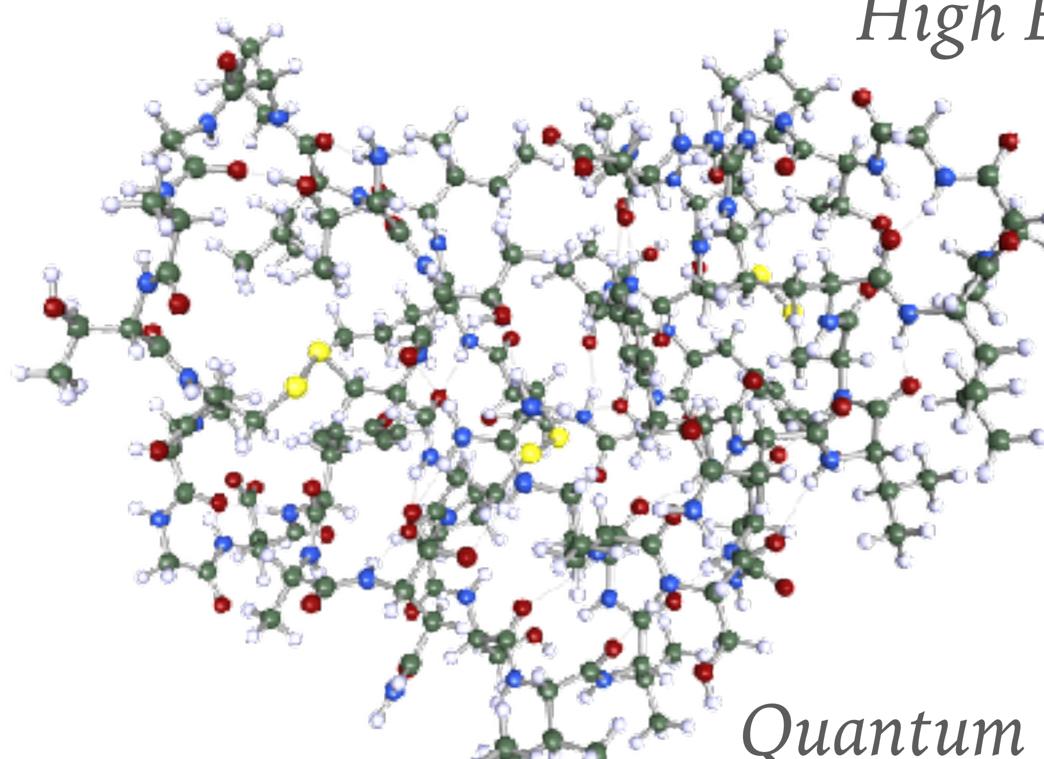
Community Detection



High Energy Physics



Graphics



Quantum Chemistry

NON-EUCLIDEAN GEOMETRIC STABILITY

- We replace the Euclidean domain Ω by a general graph $G = (V, E)$.
  $$x(u) \in L^2(\Omega) \to x(u) \in L^2(G)\ ,\ G = (V, E)\ .$$
  - In some applications, the input is the graph itself: $x \leftrightarrow G$
- We focus on undirected, possibly weighted graphs:
  $W \in \mathbb{R}^{|V| \times |V|}$: similarity matrix

NON-EUCLIDEAN GEOMETRIC STABILITY

- We replace the Euclidean domain Ω by a general graph $G = (V, E)$.

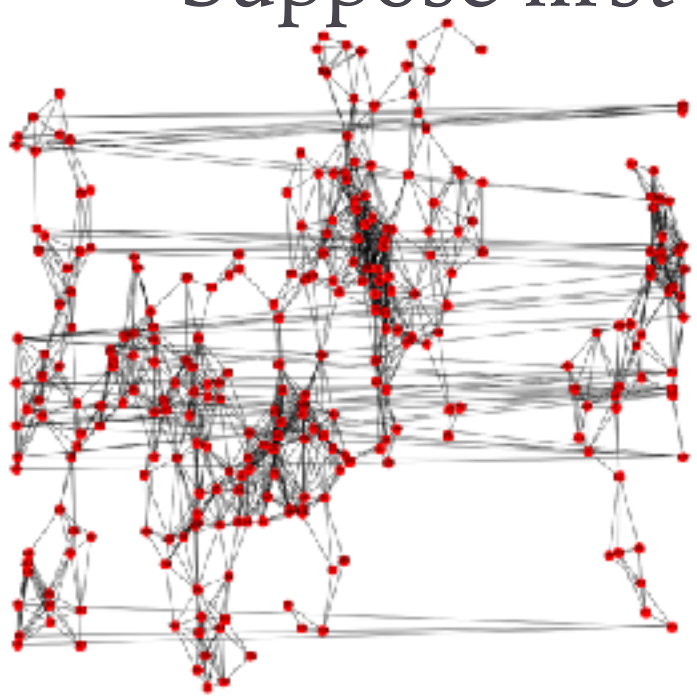
$$x(u) \in L^2(\Omega) \to x(u) \in L^2(G)\ ,\ G = (V, E)\ .$$

  - In some applications, the input is the graph itself: $x \leftrightarrow G$

- We focus on undirected, possibly weighted graphs:

$$W \in \mathbb{R}^{|V| \times |V|}: \text{ similarity matrix}$$

- Suppose first that G admits a low-dimensional embedding, ie,



particle collisions measured in LHC calorimeter

$$w_{i,j} = \varphi(x_i, x_j)\ ,\ x_i \in \Omega \subset \mathbb{R}^d\ , i, j \leq |V|.$$

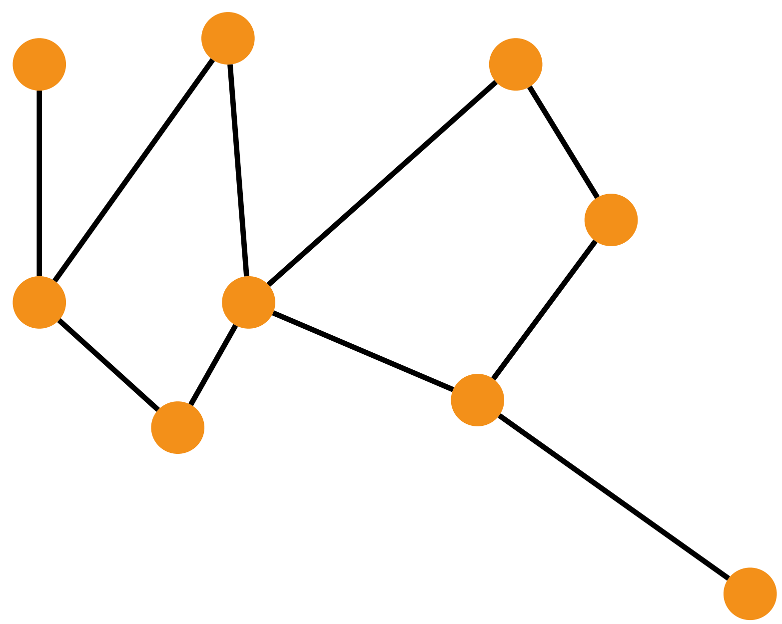
$\varphi(\cdot, \cdot)$: psd kernel (e.g. RBF, dot-product).

NON-EUCLIDEAN EXTRINSIC GEOMETRIC STABILITY

- A deformation field τ in Ω induces a deformation on G:

$$W_\tau = (w_\tau)_{i,j} \ , \ (w_\tau)_{i,j} = \varphi(\tau(x_i), \tau(x_j)) \ .$$

$$G = (V, W)$$

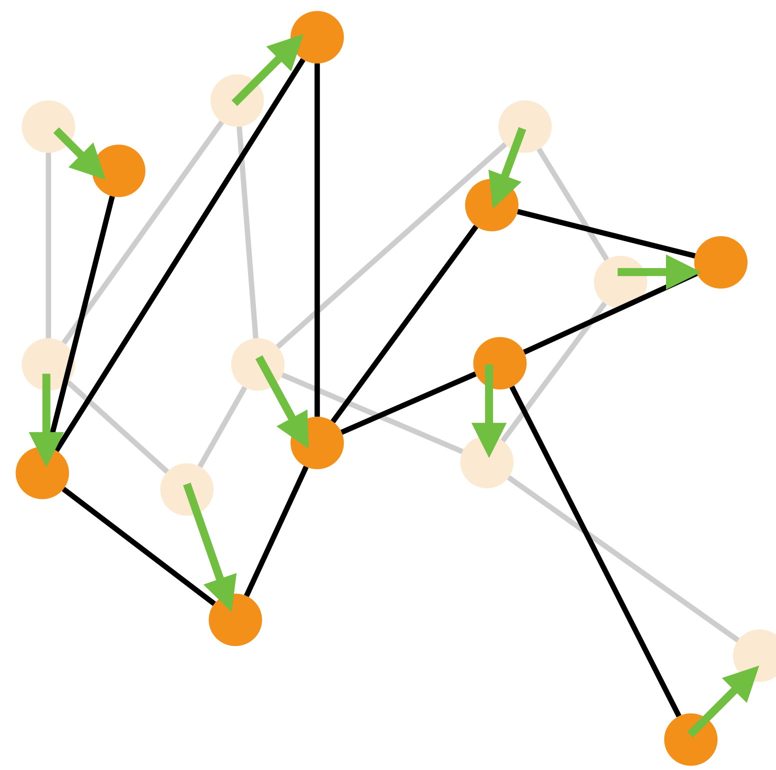


NON-EUCLIDEAN EXTRINSIC GEOMETRIC STABILITY

- A deformation field τ in Ω induces a deformation on G:

$$W_\tau = (w_\tau)_{i,j} \ , \ (w_\tau)_{i,j} = \varphi(\tau(x_i), \tau(x_j)) \ .$$

$$G_\tau = (V, W_\tau)$$



- Similarly as before, many tasks satisfy geometric stability:
  - particle physics / chemistry.
  - 3D surfaces.

$$f(G) \approx f(G_\tau) \text{ if } \|\nabla \tau\| \text{ small.}$$

- Can we define geometric deformation/stability intrinsically?

DEFORMATIONS AND METRICS

[with F. Gama and A. Ribeiro (U Penn)]

- A deformation in an Euclidean domain Ω induces a change of metric in Ω:

$$\langle x_\tau, x'_\tau \rangle_{L^2} = \int x(u-\tau(u))x'(u-\tau(u))du = \int x(v)x'(v)|\mathbf{1} - \nabla\tau(v)^{-1}|dv$$
$$= \int x(v)x'(v)dg(v) = \langle x, x' \rangle_\tau$$

- A small deformation cost corresponds to a small change of the metric.

$$(1 - o(\|\tau\|))dv \leq dg(v) \leq (1 + o(\|\tau\|))dv$$

DEFORMATIONS AND METRICS

[with F. Gama and A. Ribeiro (U Penn)]

- A deformation in an Euclidean domain Ω induces a change of metric in Ω:

$$\langle x_\tau, x'_\tau \rangle_{L^2} = \int x(u-\tau(u))x'(u-\tau(u))du = \int x(v)x'(v)|\mathbf{1} - \nabla\tau(v)^{-1}|dv$$

$$= \int x(v)x'(v)dg(v) = \langle x, x' \rangle_\tau$$

- A small deformation cost corresponds to a small change of the metric.

$$(1 - o(\|\tau\|))dv \leq dg(v) \leq (1 + o(\|\tau\|))dv$$

- Can we generalize this notion of distance between metric spaces? ie on metrics associated with an arbitrary graph?

GROMOV-HAUSDORFF DISTANCE

[with F. Gama and A. Ribeiro (U Penn)]

- An undirected graph $G = (V, E; W)$ generates a metric given by *shortest-paths*:

$$d_G(i,j) = \text{shortest path between nodes i and j.}$$

GROMOV-HAUSDORFF DISTANCE

[with F. Gama and A. Ribeiro (U Penn)]

- An undirected graph $G = (V, E; W)$ generates a metric given by *shortest-paths*:

$$d_G(i, j) = \text{shortest path between nodes i and j.}$$

- One can measure similarity between metric spaces using e.g. Gromov-Hausdorff distance:

$$d_{\mathrm{GH}}(\mathcal{M}, \mathcal{Q}) = \frac{1}{2} \inf_{\substack{\varphi : \mathcal{M} \mapsto \mathcal{Q} \\ \psi : \mathcal{Q} \mapsto \mathcal{M}}} \max\{\|\varphi\|, \|\psi\|, \|(\varphi, \psi)\|\} \, .$$

$$\|(\varphi, \psi)\| = \sup_{m \in \mathcal{M}, q \in \mathcal{Q}} |d_{\mathcal{M}}(m, \psi(q)) - d_{\mathcal{Q}}(q, \varphi(m))| \, , \|\varphi\| = \sup_{m, m' \in \mathcal{M}} |d_{\mathcal{M}}(m, m') - d_{\mathcal{Q}}(\varphi(m), \varphi(m'))| \, .$$

  - Introduced on surfaces/point-clouds in [Memoli & Sapiro'05], [Bronstein et al'06].
  - Corresponds to a permutation distance when $|V| = |V'|$:

$$d_{\mathrm{P}}(G, G') = \frac{1}{2} \min_{\pi \in \Pi_n} \max_{i,j} |d_G(i, j) - d_{G'}(\pi(i), \pi(j))| \, .$$

INTRINSIC GEOMETRIC STABILITY PRIORS

[with F. Gama and A. Ribeiro (U Penn)]

- Many inference problems on graphs are stable to intrinsic geometric deformations, in the sense that

$$|f(G) - f(G')| \lesssim d(G, G')$$

  - Community Detection.
  - Planning, Routing.

- How to leverage geometric stability on graphs?

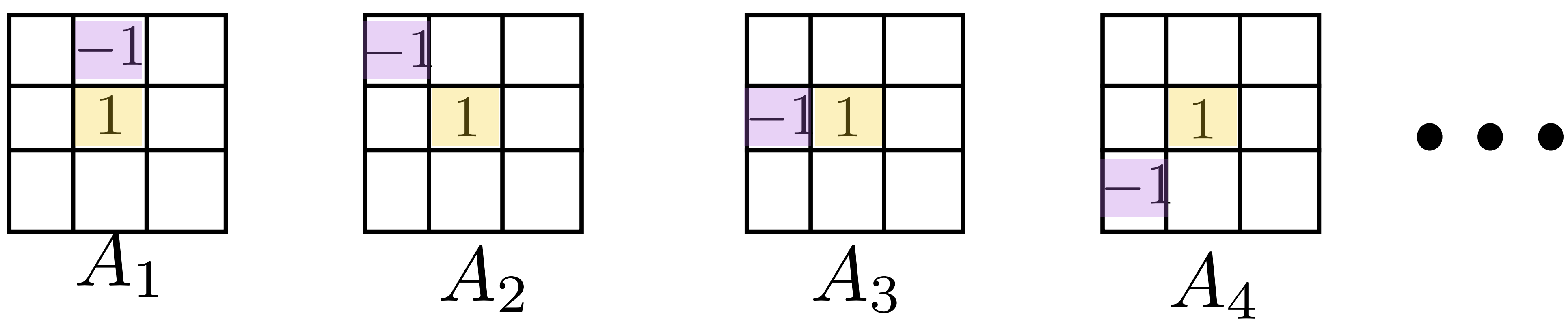
LINEAR STABLE GENERATORS

[with F. Gama and A. Ribeiro (U Penn)]

- In Euclidean domains Ω, we have seen that *localized*, multiscale filters provide the key to geometric stability.
  - These can be expressed as linear operators A of $L^2(\Omega)$ that nearly commute with deformations T_τ :

$$\|AT_\tau - T_\tau A\| \sim \|\nabla \tau\|$$

$T_\tau x(u) = x(u - \tau(u))$



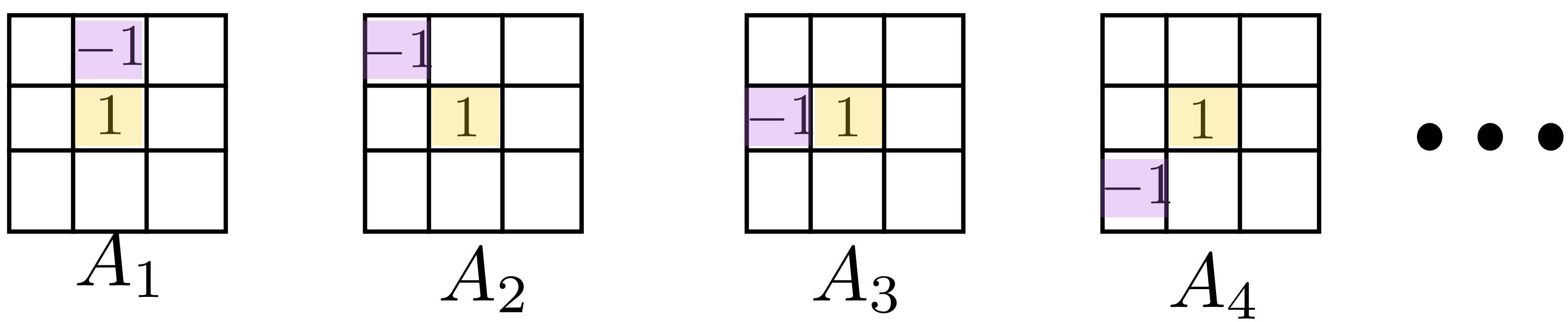
LINEAR STABLE GENERATORS

[with F. Gama and A. Ribeiro (U Penn)]

- In Euclidean domains Ω, we have seen that *localized*, multiscale filters provide the key to geometric stability.
  - These can be expressed as linear operators A of $L^2(\Omega)$ that nearly commute with deformations T_τ :

$$\|AT_\tau - T_\tau A\| \sim \|\nabla \tau\|$$

$T_\tau x(u) = x(u - \tau(u))$



  - We can write a CNN layer as a linear combination of such operators:

$$\tilde{x} = \rho\left(\sum_k (A_k x)\theta_k\right) \quad . \quad \theta_1, \ldots, \theta_k, \in \mathbb{R}^{p\times\tilde{p}} \ .$$

  - What about general graphs?

LINEAR STABLE GENERATORS

[with F. Gama and A. Ribeiro (U Penn)]

- Linear diffusion on graphs is given by its adjacency matrix $A(G)$

$A(G)_{i,j} = 1 \text{ iff } (i,j) \in E$. $\quad W_{i,j}$ in weighted graphs.

  - By definition, this is a localized operator. Local smoothing.

LINEAR STABLE GENERATORS

[with F. Gama and A. Ribeiro (U Penn)]

- Linear diffusion on graphs is given by its adjacency matrix $A(G)$

$$A(G)_{i,j} = 1 \text{ iff } (i,j) \in E \ . \quad W_{i,j} \text{ in weighted graphs.}$$

  - By definition, this is a localized operator. Local smoothing.

- Q: Stable to deformations? By definition,

$$\inf_{P \in \Pi_n} \|W - PW'P^\top\| = d_{\mathrm{P}}(G, G') \lesssim d_{\mathrm{GH}}(G, G')$$

  - Up to rigid (isometric) transformation, local diffusion is continuous with respect to metric deformations.

LINEAR STABLE GENERATORS

[with F. Gama and A. Ribeiro (U Penn)]

- Linear diffusion on graphs is given by its adjacency matrix $A(G)$

$A(G)_{i,j} = 1$ iff $(i,j) \in E$. $W_{i,j}$ in weighted graphs.

  - By definition, this is a localized operator. Local smoothing.

- Q: Stable to deformations? By definition,

$$\inf_{P \in \Pi_n} \|W - PW'P^\top\| = d_{\mathrm{P}}(G, G') \lesssim d_{\mathrm{GH}}(G, G')$$

  - Up to rigid (isometric) transformation, local diffusion is continuous with respect to metric deformations.

- Together with the degree matrix $D = \mathrm{diag}(W\mathbf{1})$, it defines a high-pass filter, the *Graph Laplacian:* $\Delta = D - W$.

  - It is also localized and stable to deformations in the sense of GH.