Deep Learning & Applied AI

Adversarial learning
Geometric deep learning

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2024/2025 · May 12, 2025

SJ7ELPG9

Adversarial attacks

One can perturb the data to generate adversarial examples.

Adversarial attacks

One can perturb the data to generate adversarial examples.

Adversarial examples can be used maliciously.

"speed limit 50mph"

Adversarial attacks

Szegedy et al, "Intriguing properties of neural networks", 2013

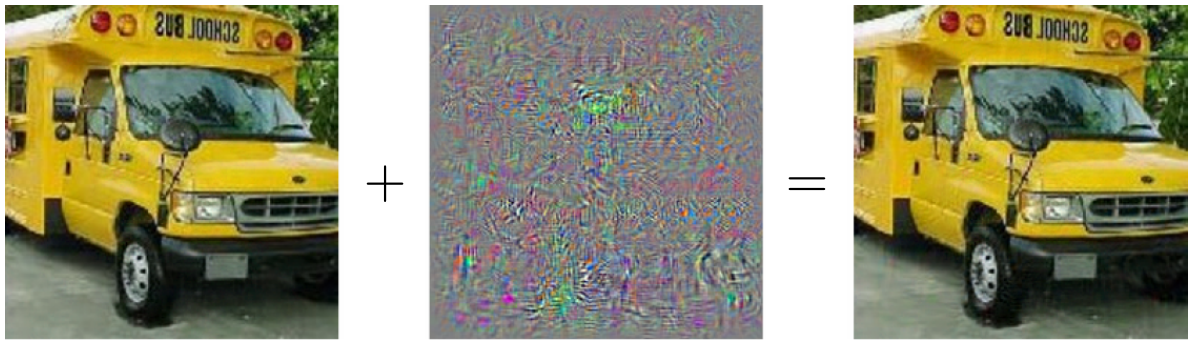
Adversarial attacks

"truck"

"ostrich"

Szegedy et al, "Intriguing properties of neural networks", 2013

Adversarial attacks

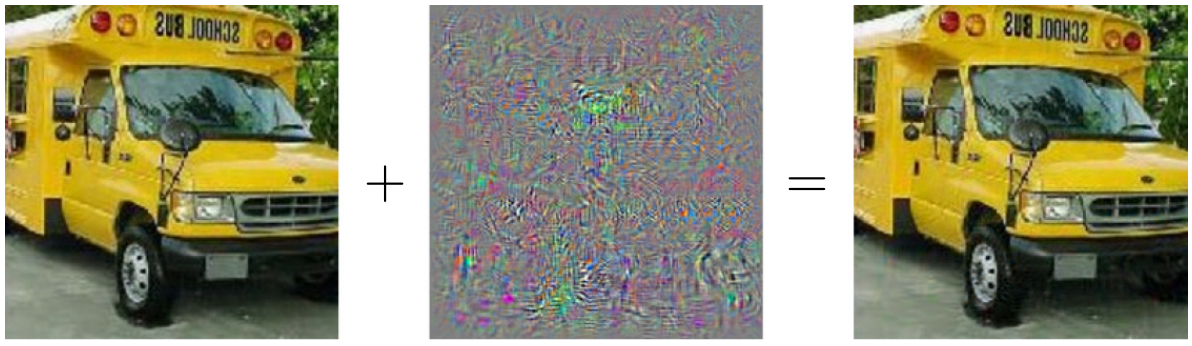


The perturbation can be explicitly optimized for.

Adversarial attacks can cause a system to take unwanted actions.

Szegedy et al, "Intriguing properties of neural networks", 2013

Adversarial attacks



The perturbation can be explicitly optimized for.

Adversarial attacks can cause a system to take unwanted actions.

How to construct undetectable adversarial examples?

Szegedy et al, "Intriguing properties of neural networks", 2013

Types of attack

Distinction based on the amount of information available to the attacker:

- Black-box attack:

  Can only query the target model.

Types of attack

Distinction based on the amount of information available to the attacker:

- Black-box attack:

  Can only query the target model.

- Gray-box attack:

  Access to partial information (only the features, architecture, etc.).

Types of attack

Distinction based on the amount of information available to the attacker:

- Black-box attack:

  Can only query the target model.

- Gray-box attack:

  Access to partial information (only the features, architecture, etc.).

- White-box attack:

  Complete access to the network (architecture, parameters, etc.).

Types of attack

Distinction based on the amount of information available to the attacker:

- Black-box attack:

  Can only query the target model.

- Gray-box attack:

  Access to partial information (only the features, architecture, etc.).

- White-box attack:

  Complete access to the network (architecture, parameters, etc.).

  You are not allowed to touch the network weights.

Targeted attacks

Given an input sample $\mathbf{x}$, a classifier C, and a target class t, consider:

$$\min_{\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_2^2$$
$$\text{s.t. } C(\mathbf{x}') = t$$

Szegedy et al, "Intriguing properties of neural networks", 2013

Targeted attacks

Given an input sample $\mathbf{x}$, a classifier C, and a target class t, consider:

$$\min_{\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_2^2$$
$$\text{s.t. } C(\mathbf{x}') = t$$

Relax the difficult constraint to a penalty term:

$$\min_{\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_2^2 + c\, L(\mathbf{x}', t)$$

where L is the cross-entropy loss.

Szegedy et al, "Intriguing properties of neural networks", 2013

Targeted attacks

Given an input sample $\mathbf{x}$, a classifier C, and a target class t, consider:

$$\min_{\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_2^2$$
$$\text{s.t. } C(\mathbf{x}') = t$$

Relax the difficult constraint to a penalty term:

$$\min_{\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_2^2 + c\, L(\mathbf{x}', t)$$

where L is the cross-entropy loss.

$c > 0$ is a trade-off parameter that is chosen as small as possible; it can be found via line search.

Szegedy et al, "Intriguing properties of neural networks", 2013

Targeted attacks

A more general approach is given by:

$$\begin{aligned} \min_{\boldsymbol{\delta} \in [0,1]^n} \quad & d(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) \\ \text{s.t.} \quad & C(\mathbf{x} + \boldsymbol{\delta}) = t \end{aligned}$$

where the perturbation $\boldsymbol{\delta}$ appears explicitly, and d depends on the specific task.

Carlini and Wagner, "Towards Evaluating the Robustness of Neural Networks", 2016

Targeted attacks

A more general approach is given by:

$$\begin{aligned} \min_{\boldsymbol{\delta} \in [0,1]^n} \quad & d(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) \\ \text{s.t.} \quad & f(\mathbf{x} + \boldsymbol{\delta}) \leq 0 \end{aligned}$$

where the perturbation $\boldsymbol{\delta}$ appears explicitly, and d depends on the specific task.

f is such that $C(\mathbf{x} + \boldsymbol{\delta}) = t$ if and only if $f(\mathbf{x} + \boldsymbol{\delta}) \leq 0$.

Carlini and Wagner, "Towards Evaluating the Robustness of Neural Networks", 2016

Targeted attacks

A more general approach is given by:

$$\min_{\boldsymbol{\delta} \in [0,1]^n} d(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) + c\, f(\mathbf{x} + \boldsymbol{\delta})$$

where the perturbation $\boldsymbol{\delta}$ appears explicitly, and d depends on the specific task.

f is such that $C(\mathbf{x} + \boldsymbol{\delta}) = t$ if and only if $f(\mathbf{x} + \boldsymbol{\delta}) \leq 0$.

Carlini and Wagner, "Towards Evaluating the Robustness of Neural Networks", 2016

Targeted attacks

A more general approach is given by:

$$\min_{\boldsymbol{\delta} \in [0,1]^n} d(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}) + c\, f(\mathbf{x} + \boldsymbol{\delta})$$

where the perturbation $\boldsymbol{\delta}$ appears explicitly, and d depends on the specific task.

f is such that $C(\mathbf{x} + \boldsymbol{\delta}) = t$ if and only if $f(\mathbf{x} + \boldsymbol{\delta}) \leq 0$.

For example:

$$\min_{\boldsymbol{\delta} \in [0,1]^n} \|\boldsymbol{\delta}\|_p + c\, (\max\{F(\mathbf{x} + \boldsymbol{\delta})_i \,:\, i \neq t\} - F(\mathbf{x} + \boldsymbol{\delta})_t)^+$$

where $F : \mathbf{x} \mapsto [0,1]^k$ is the NN yielding a probability distribution over all k classes, and $(a)^+ = \max(a, 0)$.

Carlini and Wagner, "Towards Evaluating the Robustness of Neural Networks", 2016

Untargeted attacks

If there is no specific target, consider for a given input $\mathbf{x}$ with ground-truth label ℓ_{gt}:

$$\mathbf{x}' = \mathbf{x} + \alpha \underbrace{\operatorname{sign}\left(\nabla L(\mathbf{x}, \ell_{\text{gt}})\right)}_{\text{perturbation}}$$

which adds a perturbation maximizing the cost.

Kurakin et al, "Adversarial examples in the physical world", 2016

Untargeted attacks

If there is no specific target, consider for a given input $\mathbf{x}$ with ground-truth label ℓ_{gt}:

$$\mathbf{x}' = \mathbf{x} + \alpha \underbrace{\operatorname{sign}\left(\nabla L(\mathbf{x}, \ell_{\text{gt}})\right)}_{\text{perturbation}}$$

which adds a perturbation maximizing the cost.

For better results, iterate:

$$\mathbf{x}'_{(i)} = \operatorname{clip}_{\epsilon}\left(\mathbf{x}'_{(i-1)} + \alpha \operatorname{sign}\left(\nabla L(\mathbf{x}'_{(i-1)}, \ell_{\text{gt}})\right)\right)$$

with $\mathbf{x}'_{(0)} = \mathbf{x}$.

The clip operation projects to an ϵ-neighborhood from $\mathbf{x}$.

Kurakin et al, "Adversarial examples in the physical world", 2016

Untargeted attacks

If there is no specific target, consider for a given input $\mathbf{x}$ with ground-truth label ℓ_{gt}:

$$\mathbf{x}' = \mathbf{x} + \alpha \underbrace{\operatorname{sign}\left(\nabla L(\mathbf{x}, \ell_{\text{gt}})\right)}_{\text{perturbation}}$$

which adds a perturbation maximizing the cost.

For better results, iterate:

$$\mathbf{x}'_{(i)} = \operatorname{clip}_{\epsilon}\left(\mathbf{x}'_{(i-1)} + \alpha \operatorname{sign}\left(\nabla L(\mathbf{x}'_{(i-1)}, \ell_{\text{gt}})\right)\right)$$

with $\mathbf{x}'_{(0)} = \mathbf{x}$.

The clip operation projects to an ϵ-neighborhood from $\mathbf{x}$.

Designed to be fast: 1 iteration ≈ 1 backprop step.

Kurakin et al, "Adversarial examples in the physical world", 2016

Example: Adversarial training

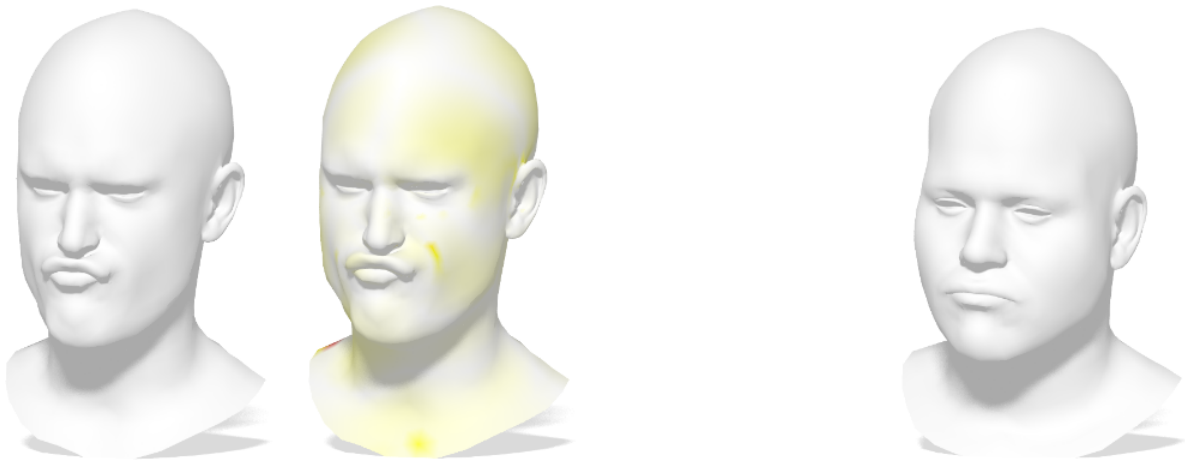
Using adversarial examples as training data improves the robustness of the attacked learning model:

original

target

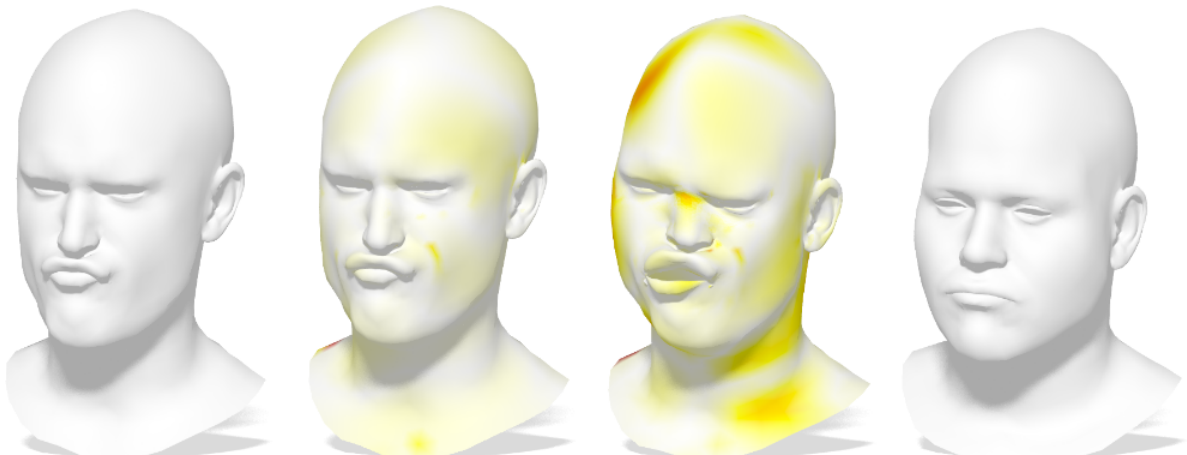
Example: Adversarial training

Using adversarial examples as training data improves the robustness of the attacked learning model:



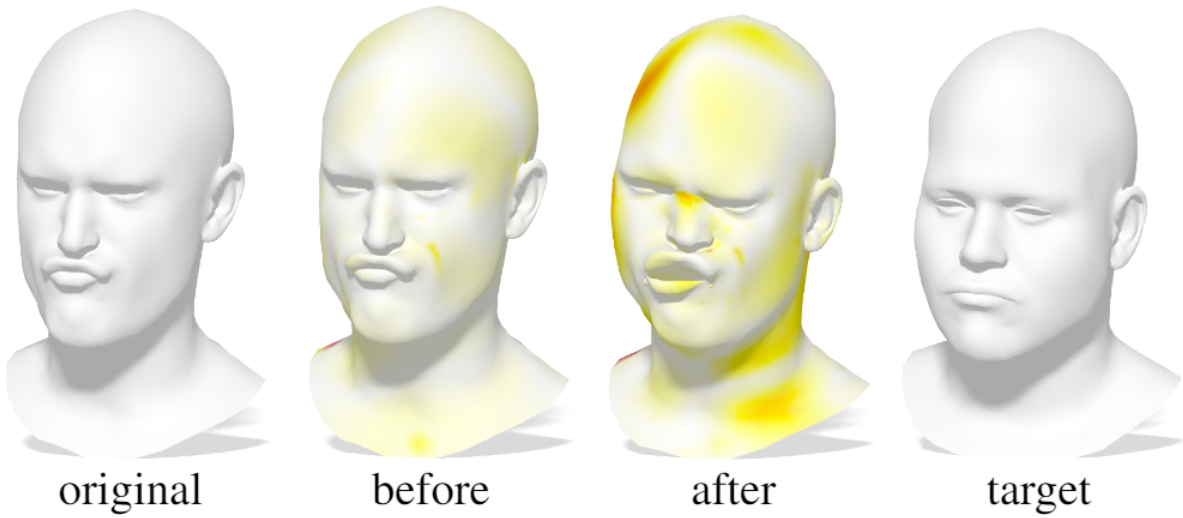
Example: Adversarial training

Using adversarial examples as training data improves the robustness of the attacked learning model:



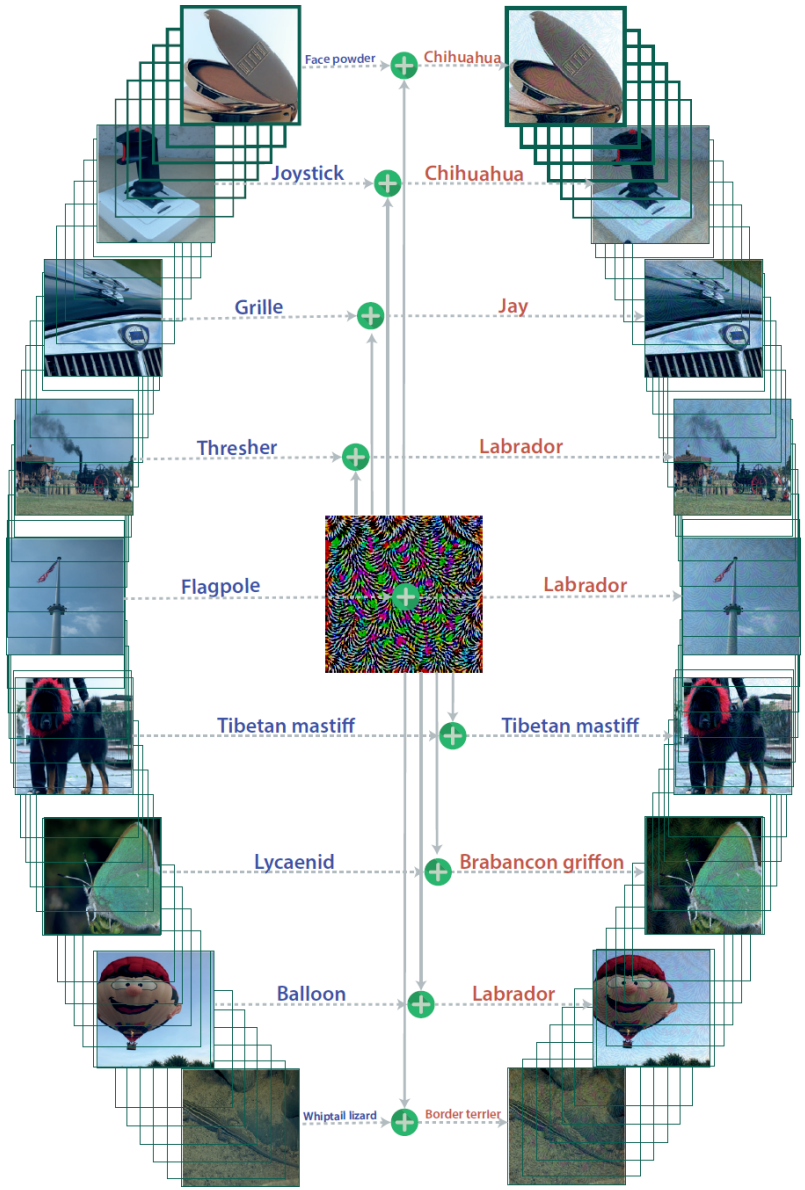
Example: Adversarial training

Using adversarial examples as training data improves the robustness of the attacked learning model:



Still, it can be proven that classifiers are always vulnerable!

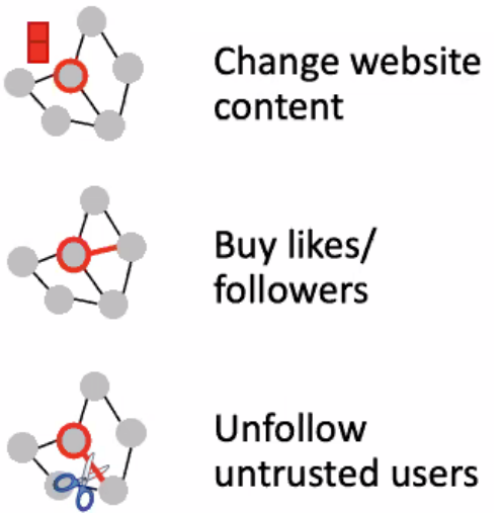
Universal perturbations

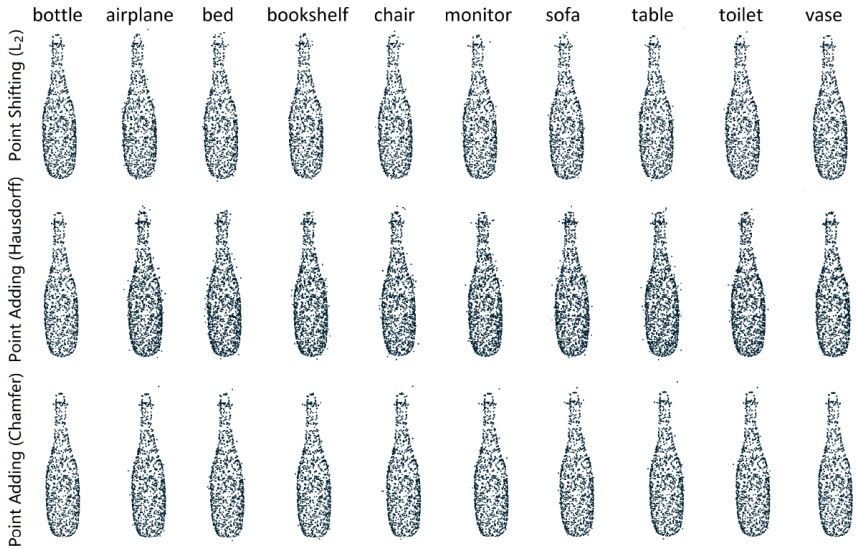


Moosavi-Dezfooli et al, "Universal adversarial perturbations", 2017

Non-Euclidean domains

Adversarial training can also be done on geometric domains.

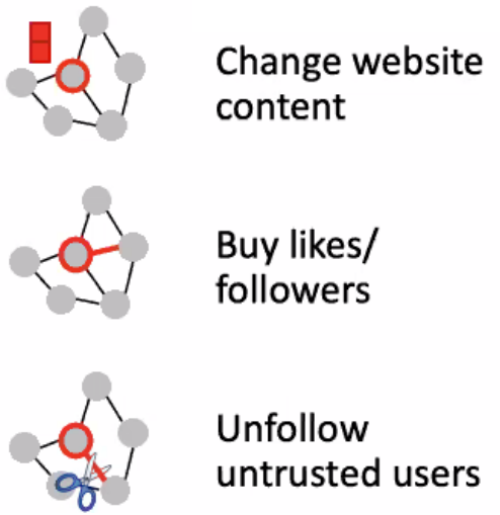


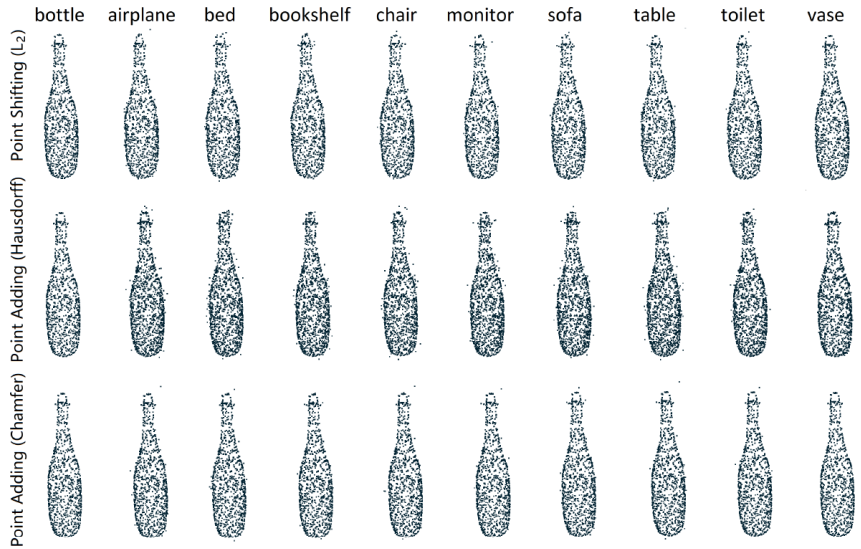


Zügner et al, "Adversarial attacks on neural networks for graph data", 2018; Xiang et al, "Generating 3D Adversarial Point Clouds", 2018

Non-Euclidean domains

Adversarial training can also be done on geometric domains.

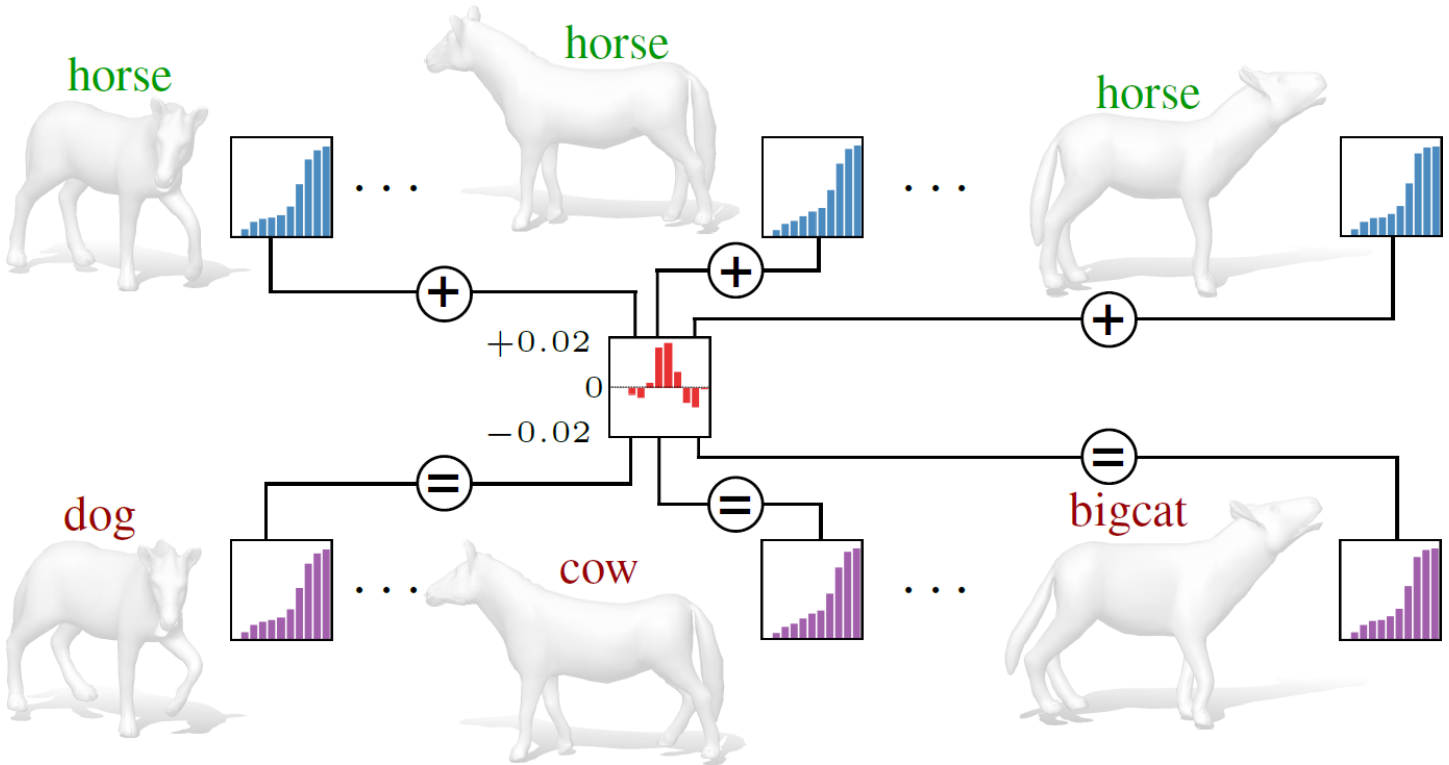




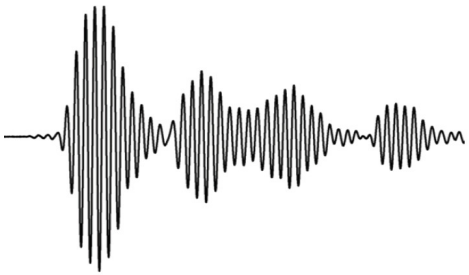
- The notion of perceptible is different than with images.
- Can alter the domain (e.g. the graph connections) rather than just the features (e.g. the values stored at the nodes).

Zügner et al, "Adversarial attacks on neural networks for graph data", 2018; Xiang et al, "Generating 3D Adversarial Point Clouds", 2018

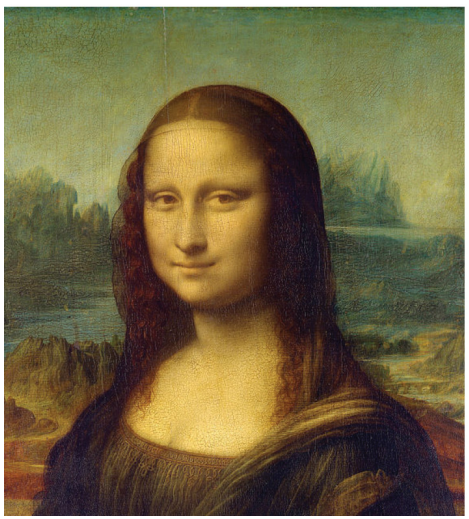
Universal perturbations on 3D data



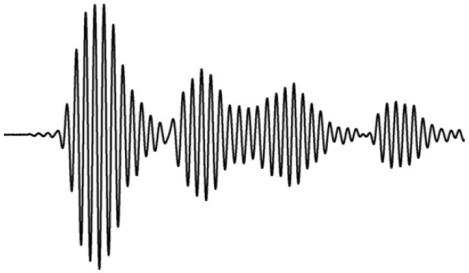
Rampini, Pestarini, Cosmo, Melzi, Rodolà, "Universal Spectral Adversarial Attacks for Deformable Shapes", 2021



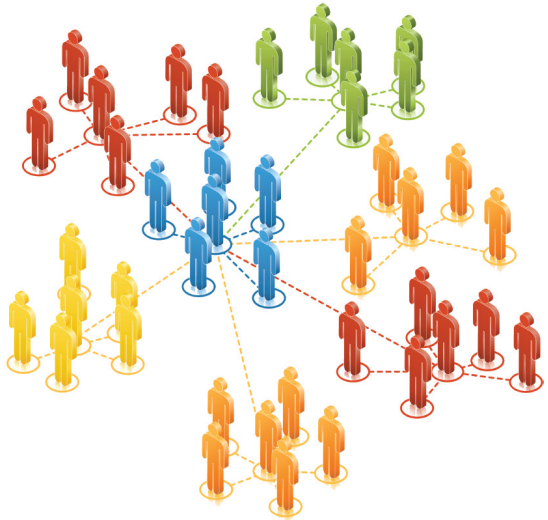
Audio signals



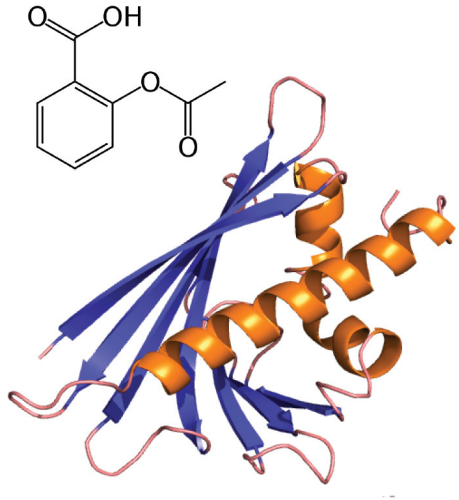
Images



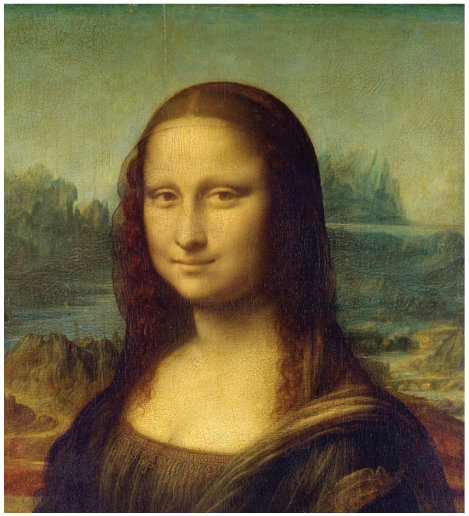
Audio signals



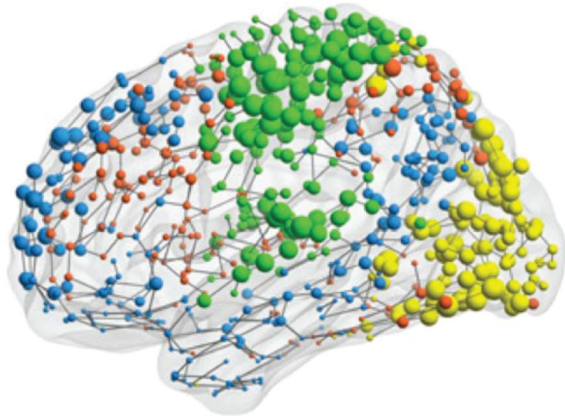
Social networks



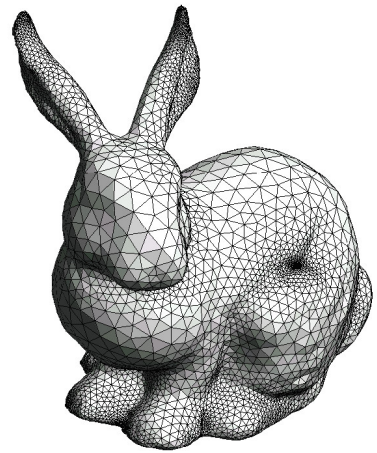
Molecules



Images

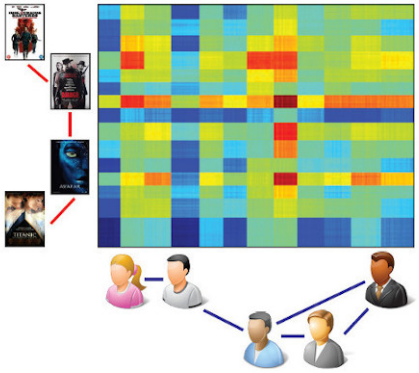


Functional networks

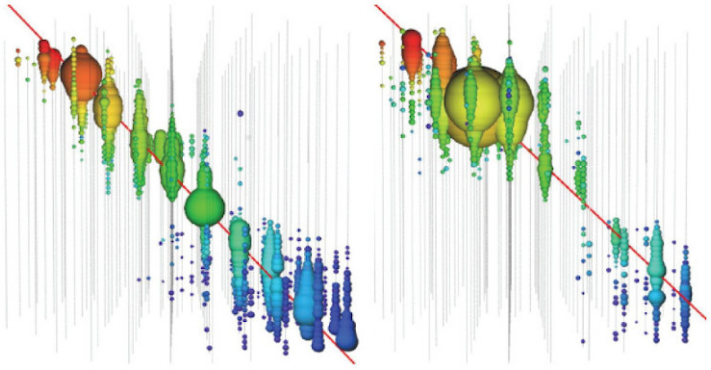


3D shapes

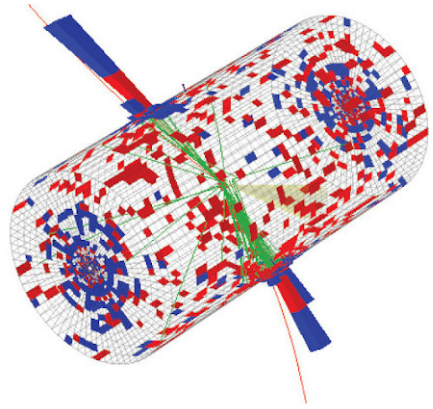
Applications of geometric deep learning



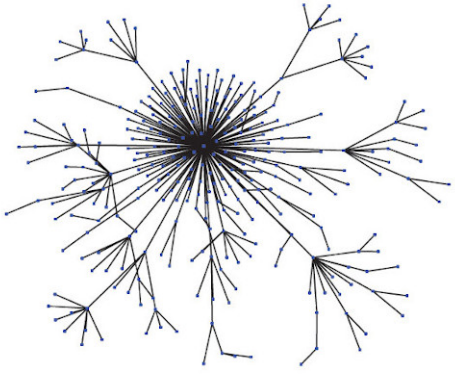
Recommender system



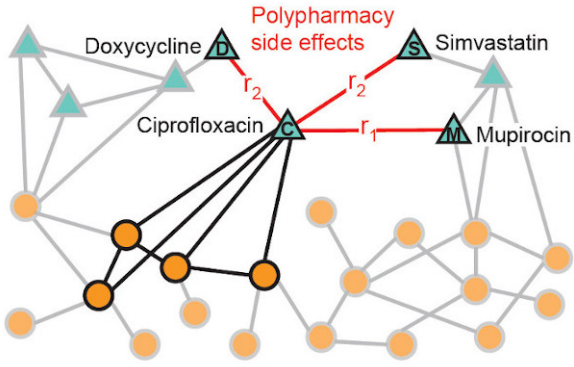
Neutrino detection



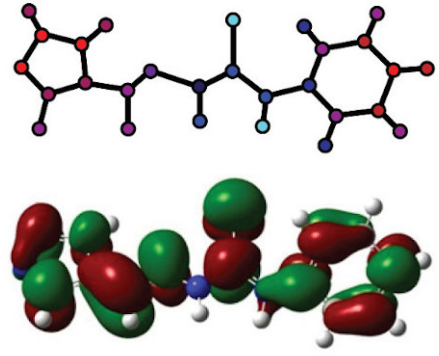
LHC



Fake news detection



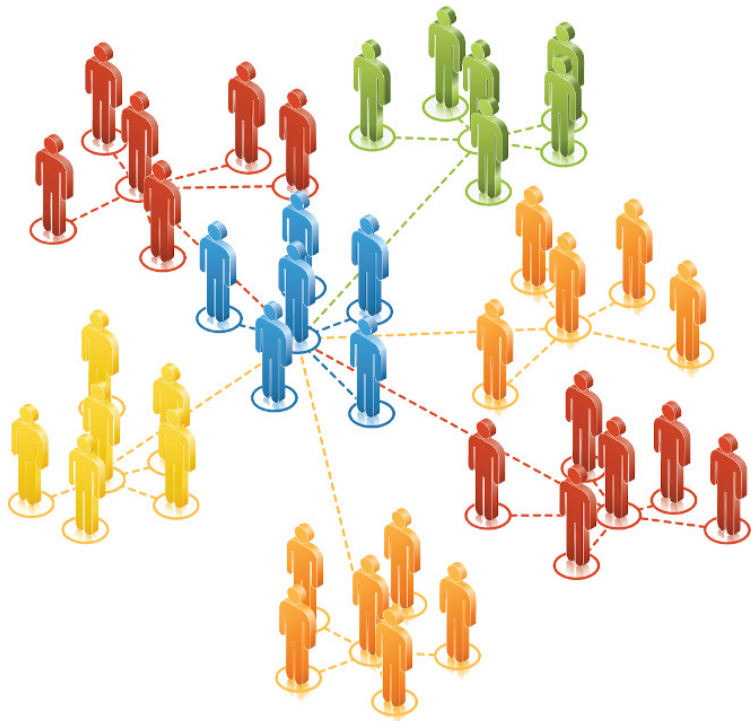
Drug repurposing



Chemistry

Prototypical non-Euclidean objects

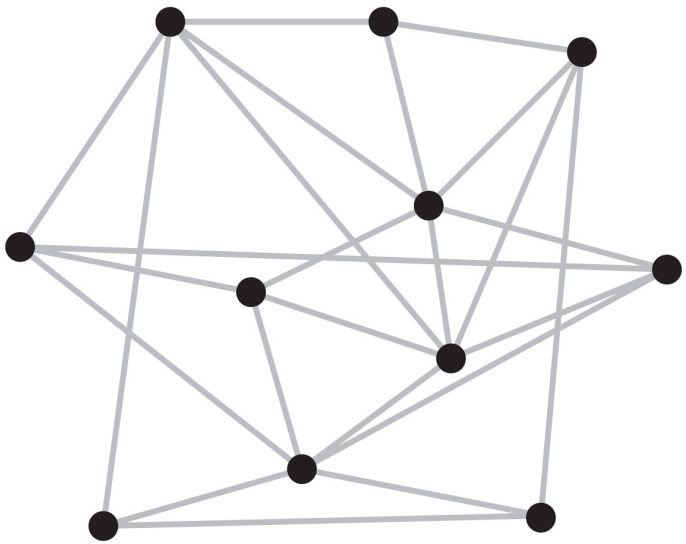
Manifolds



Graphs

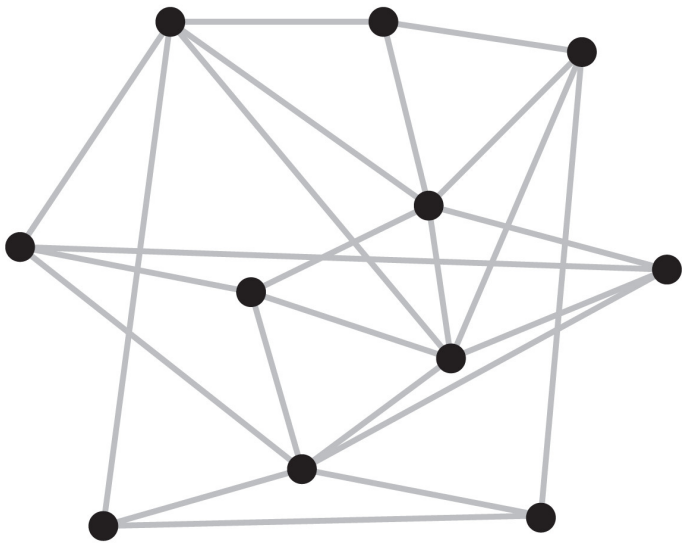
Domain structure vs Data on a domain

Domain structure vs Data on a domain

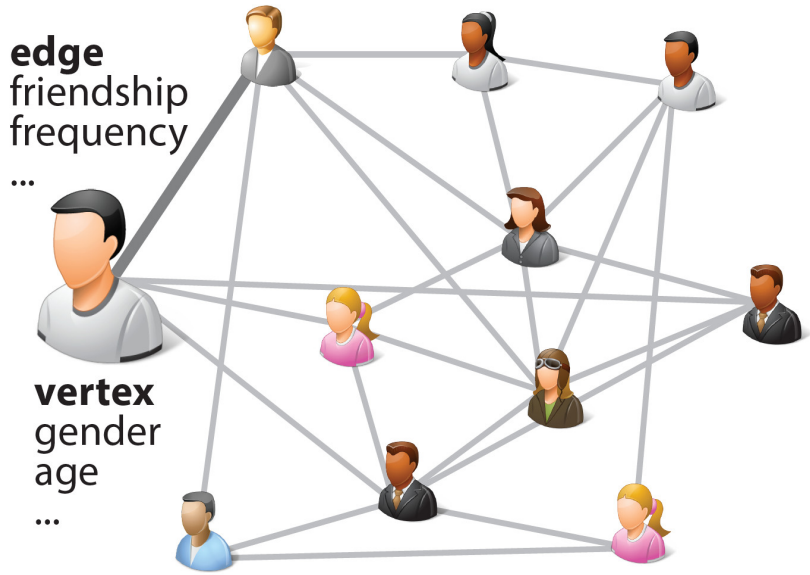


Domain structure

Domain structure vs Data on a domain

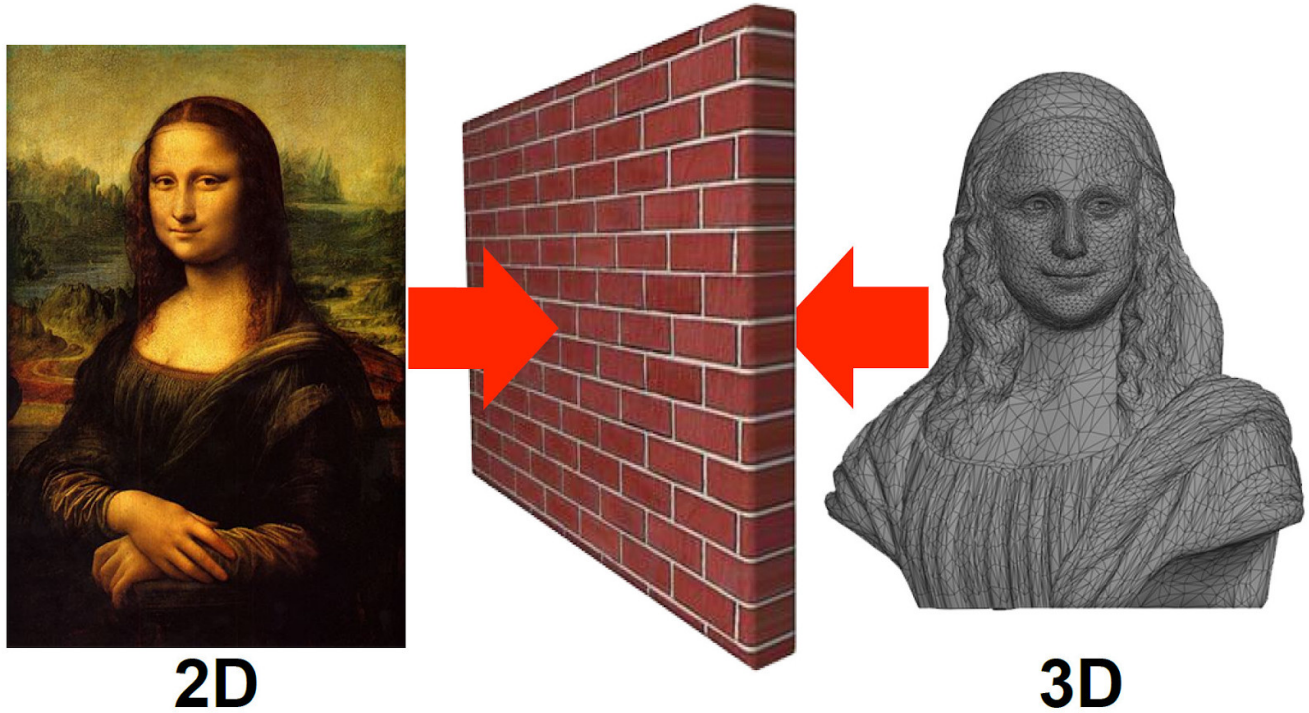


Domain structure



Data on a domain

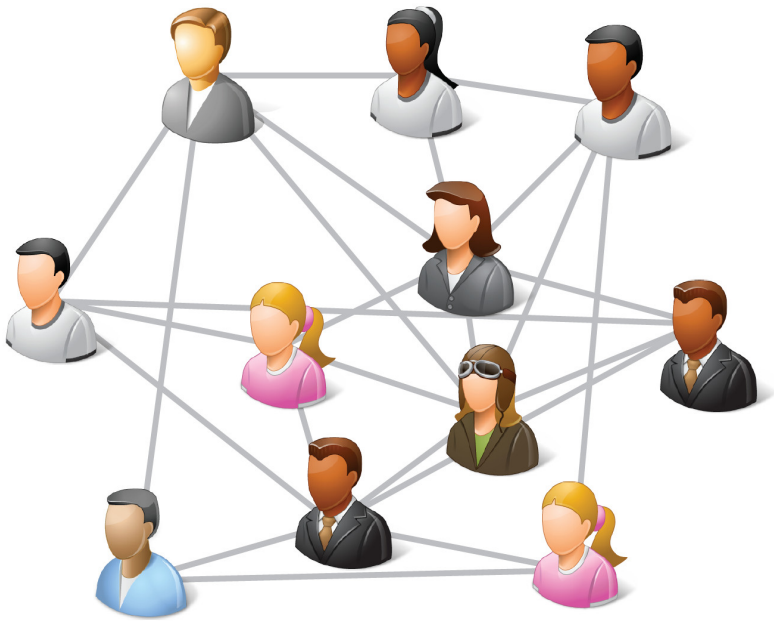
Domain structure vs Data on a domain



only data

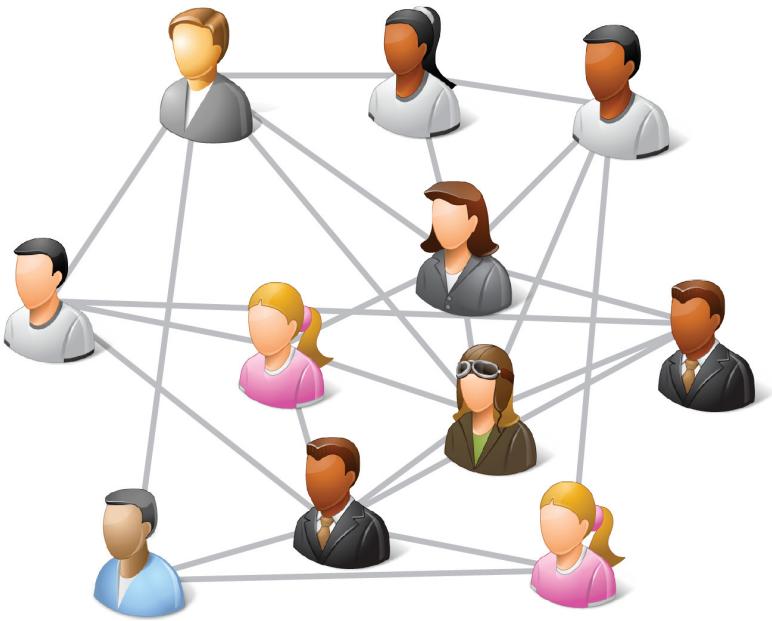
only structure

Fixed vs different domain

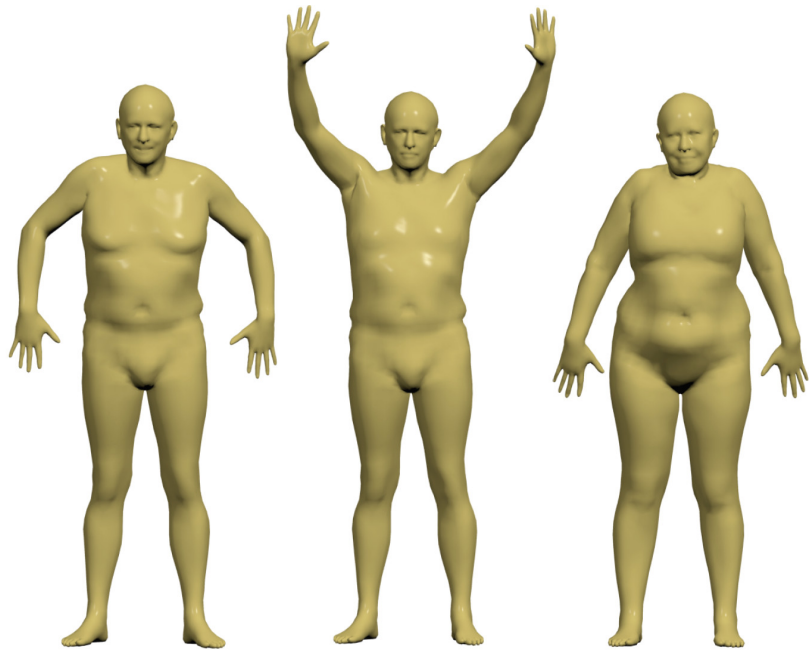


Social network
(fixed graph)

Fixed vs different domain



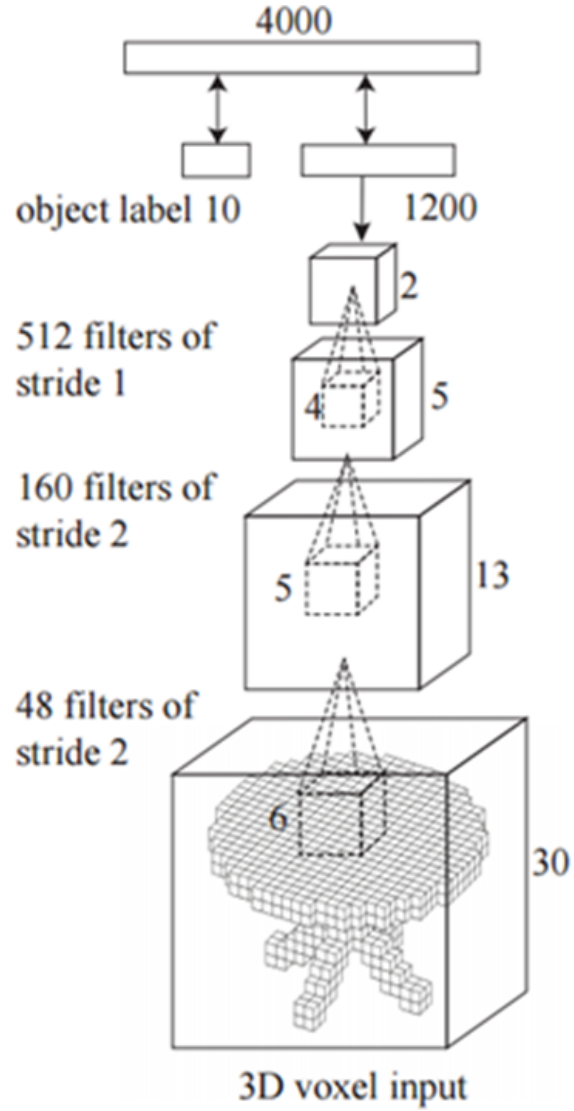
Social network
(fixed graph)



3D shapes
(different manifolds)

3D ShapeNets

- Volumetric representation (shape = binary voxels on 3D grid)

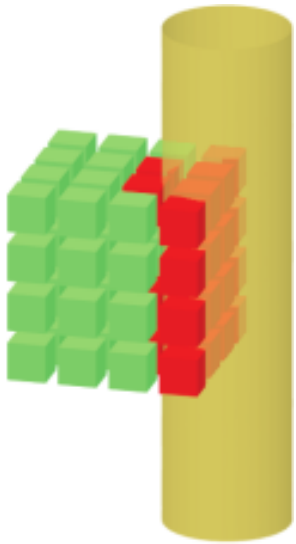


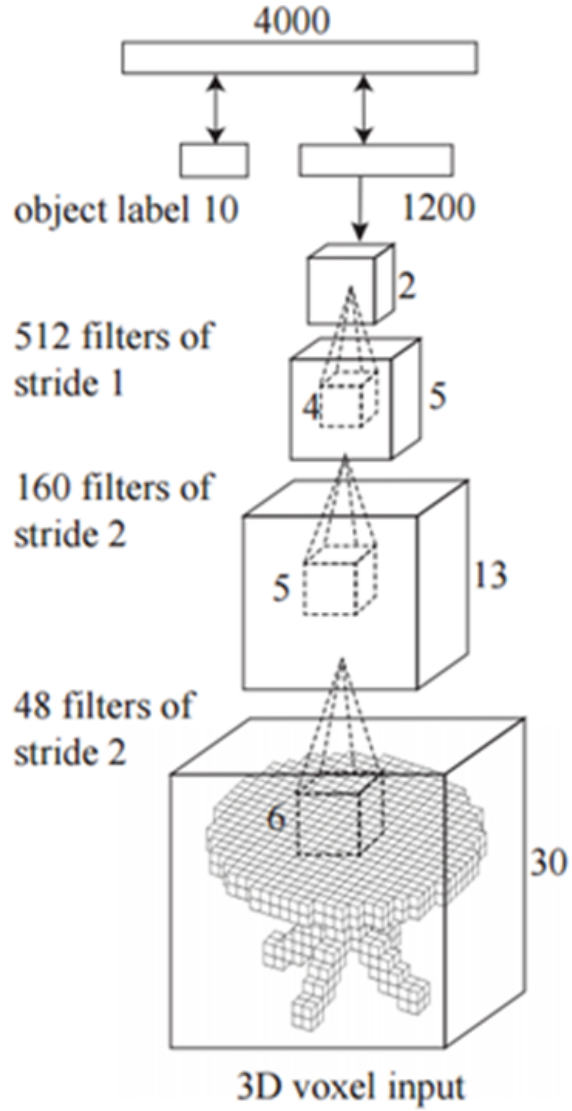
Convolutional deep belief network

Wu et al, "3D ShapeNets: A Deep Representation for Volumetric Shapes" 2015

3D ShapeNets

- Volumetric representation (shape = binary voxels on 3D grid)
- 3D convolutional network

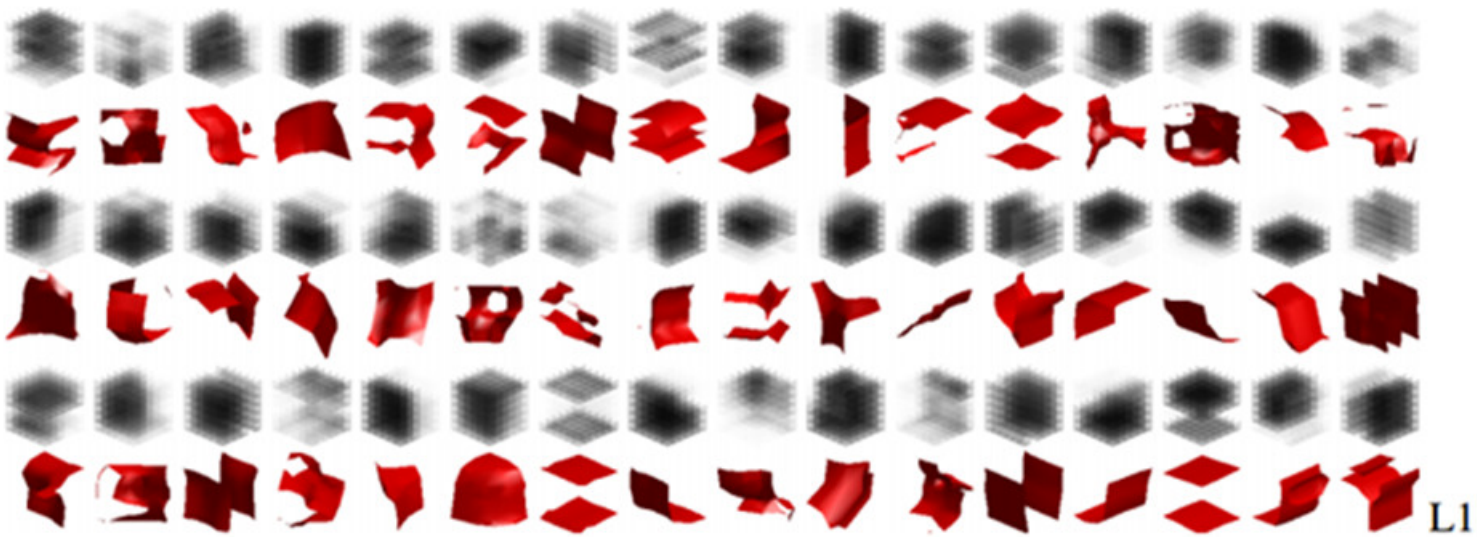




Convolutional deep belief network

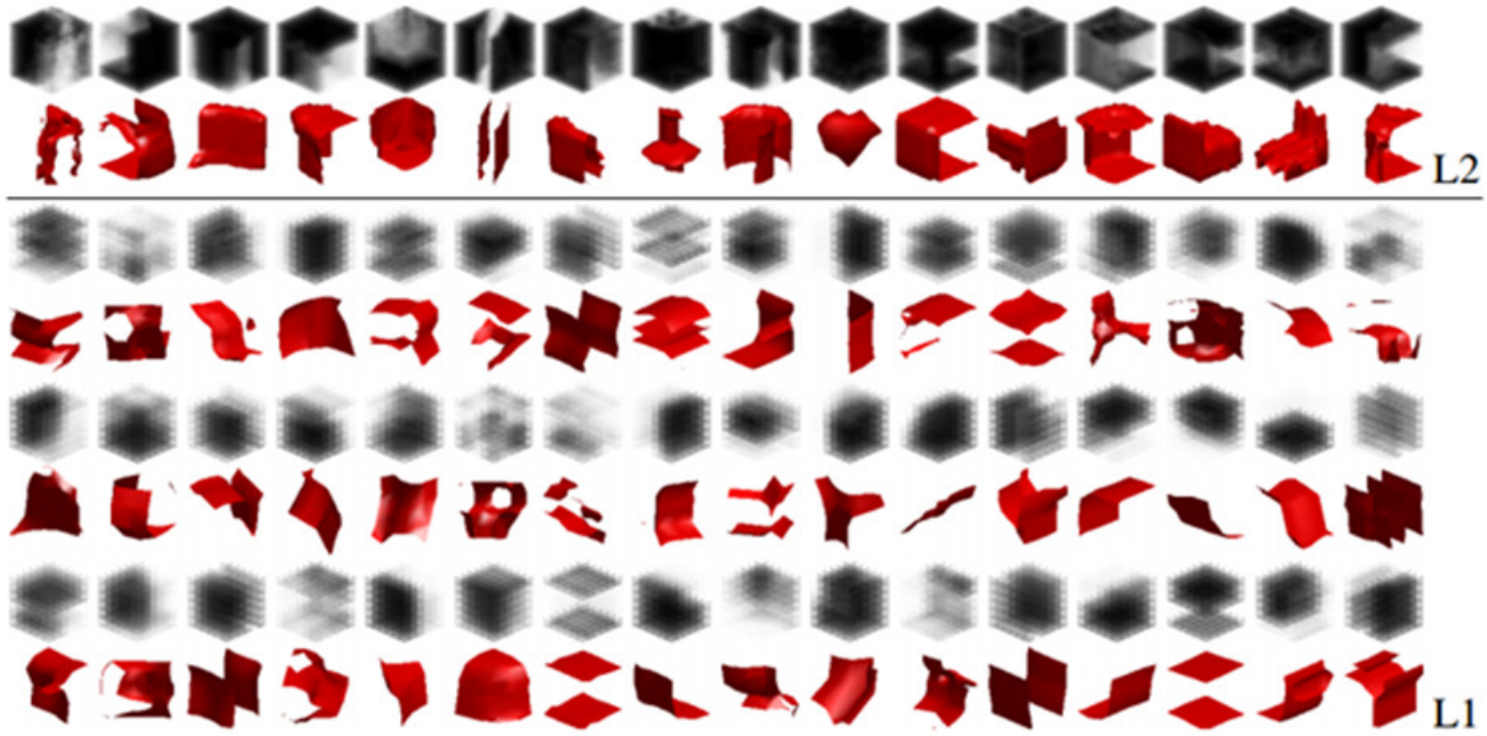
Wu et al, "3D ShapeNets: A Deep Representation for Volumetric Shapes" 2015

Learned features: 3D primitives



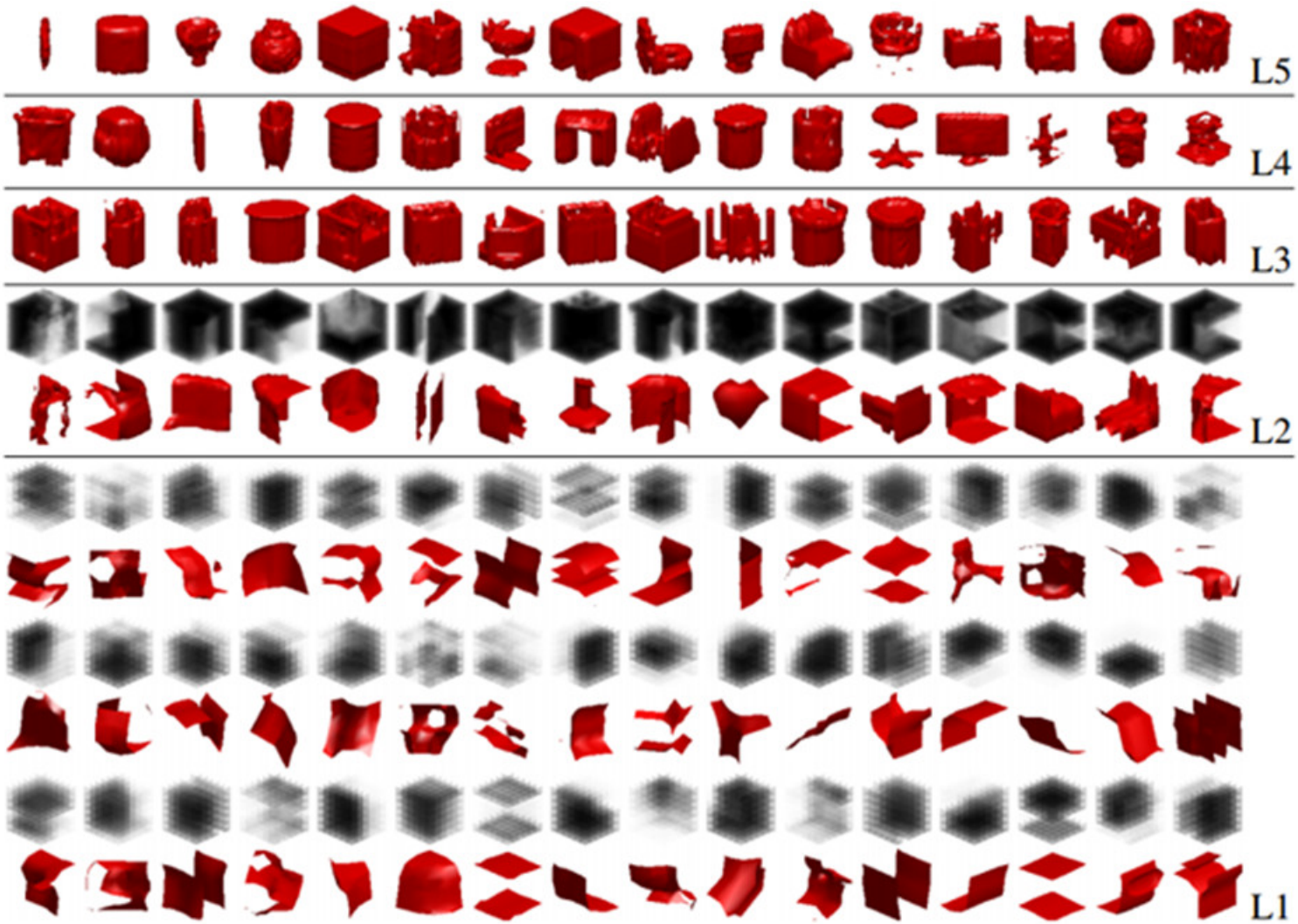
Wu et al, "3D ShapeNets: A Deep Representation for Volumetric Shapes" 2015

Learned features: 3D primitives



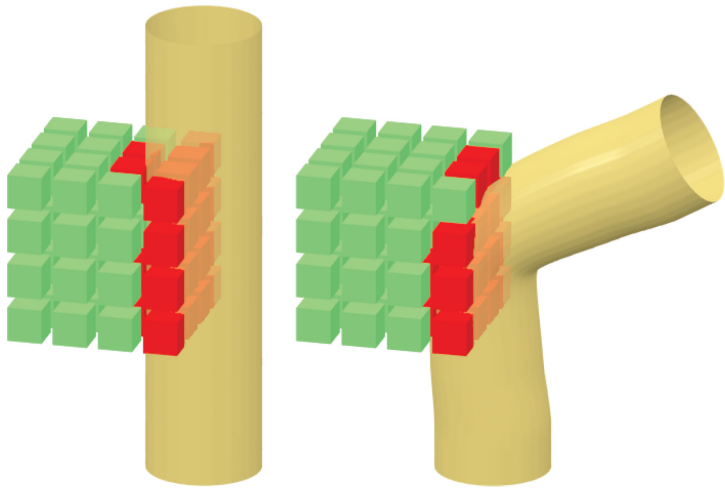
Wu et al, "3D ShapeNets: A Deep Representation for Volumetric Shapes" 2015

Learned features: 3D primitives

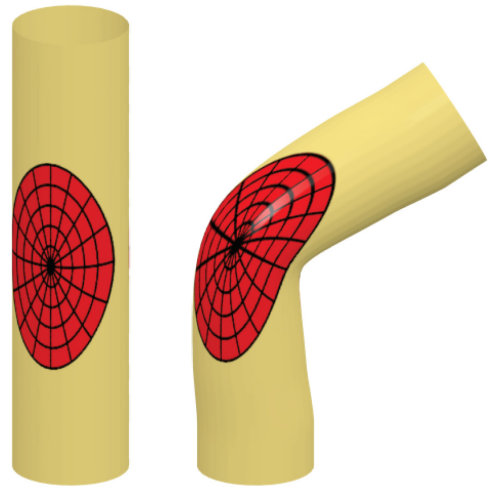


Wu et al, "3D ShapeNets: A Deep Representation for Volumetric Shapes" 2015

Challenges of geometric deep learning

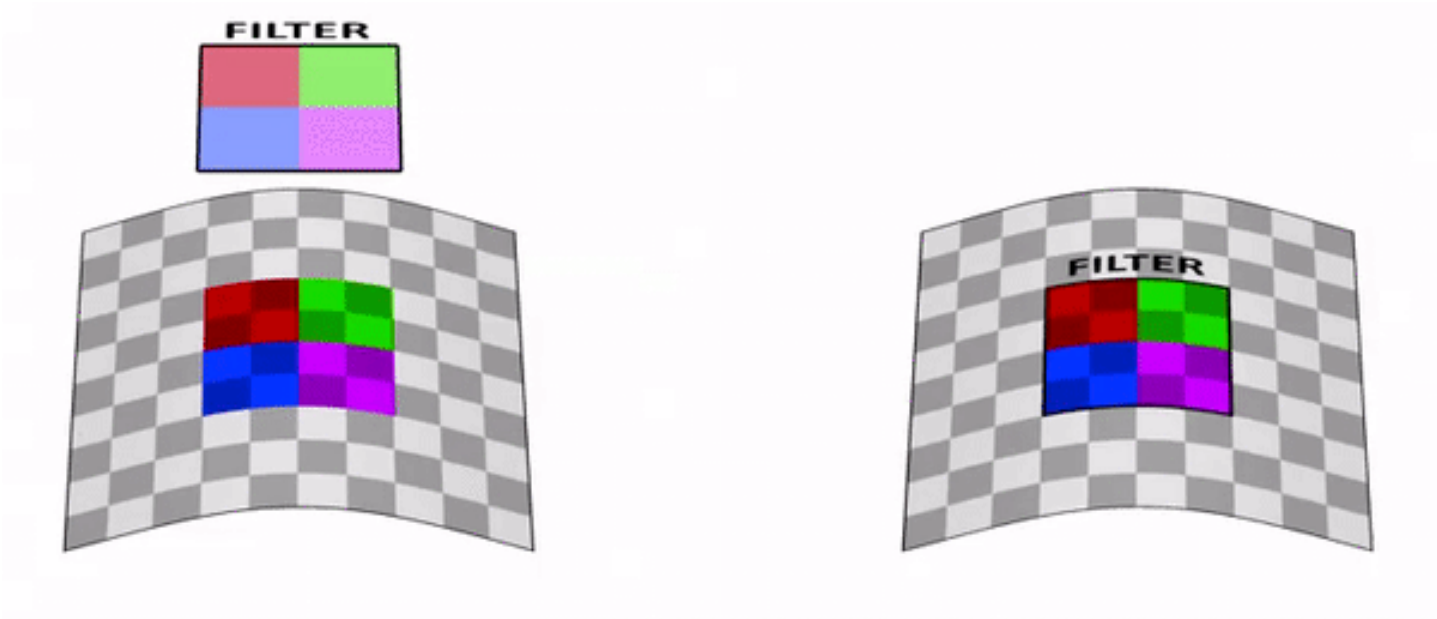


Extrinsic



Intrinsic

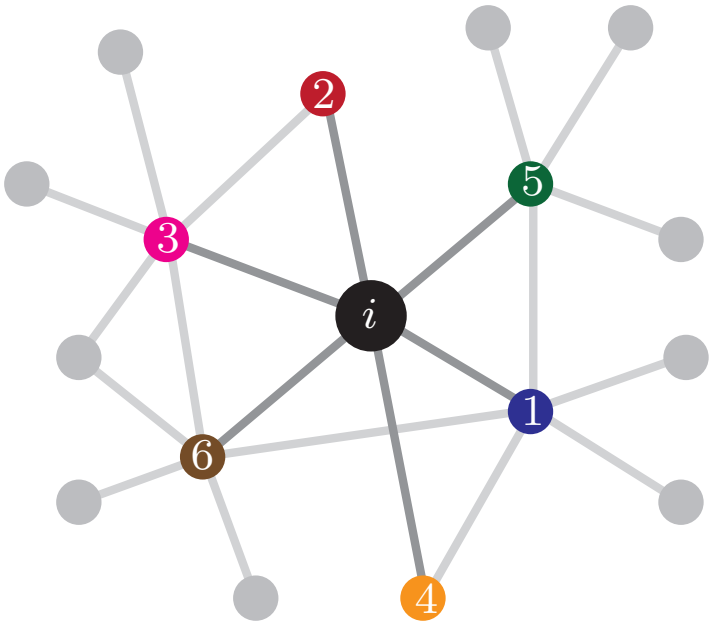
Extrinsic vs Intrinsic



Extrinsic **Intrinsic**

Local ambiguity

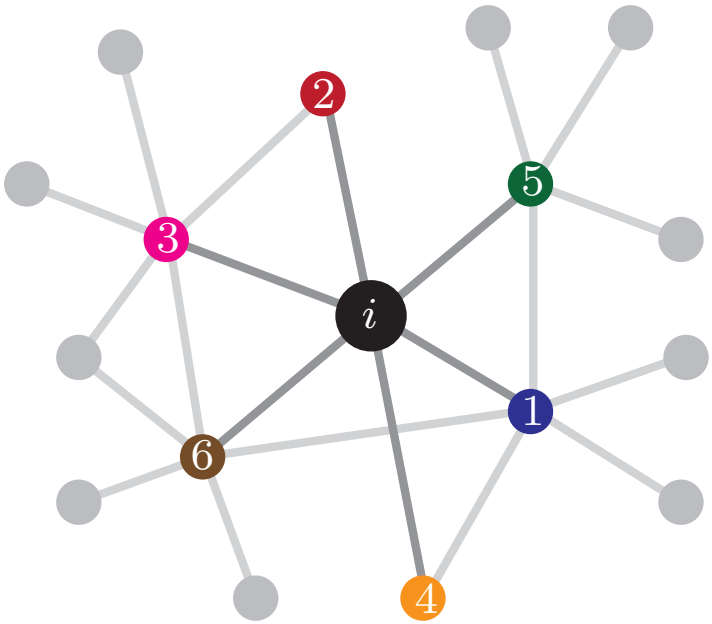
Unlike images, there is **no** canonical ordering of the domain points.



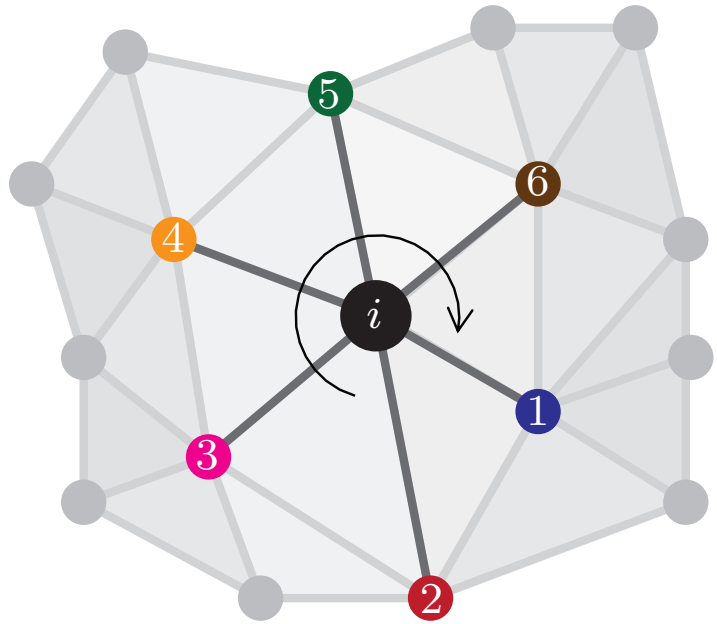
Graph (permutation)

Local ambiguity

Unlike images, there is **no** canonical ordering of the domain points.

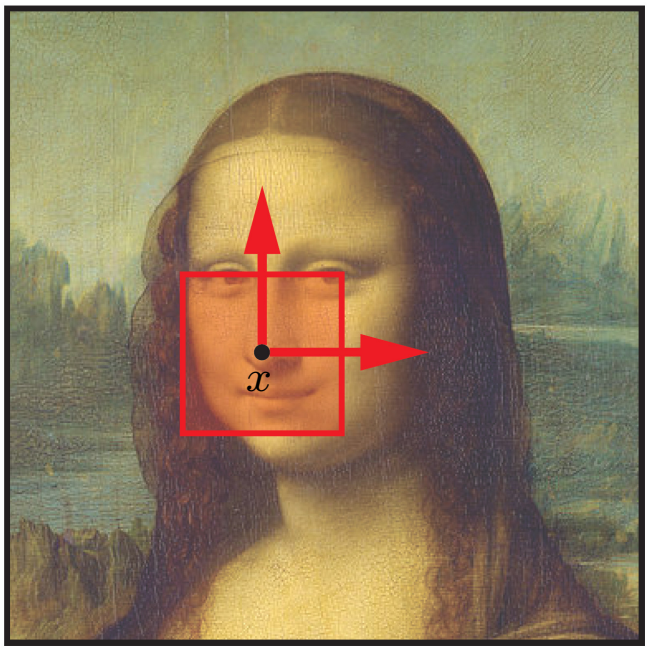


Graph (permutation)

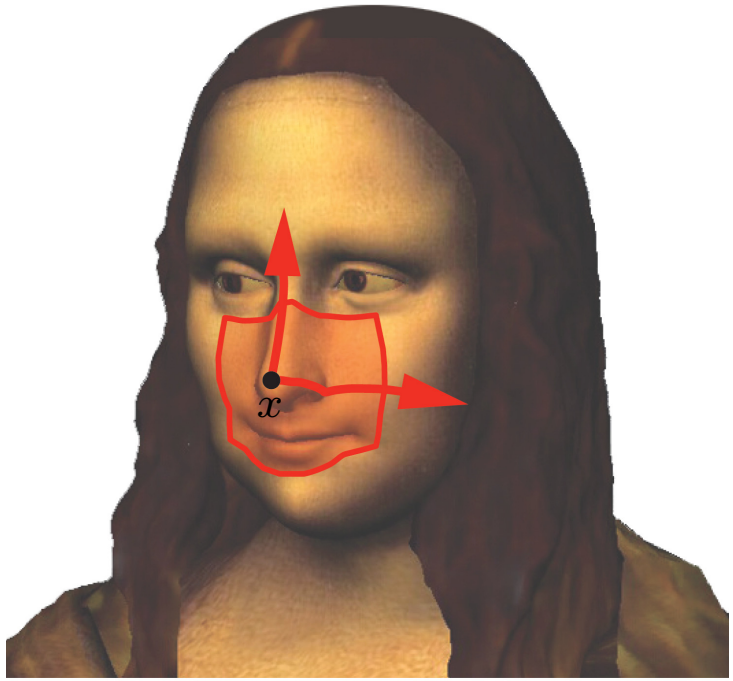


Mesh (rotation)

Non-Euclidean convolution?

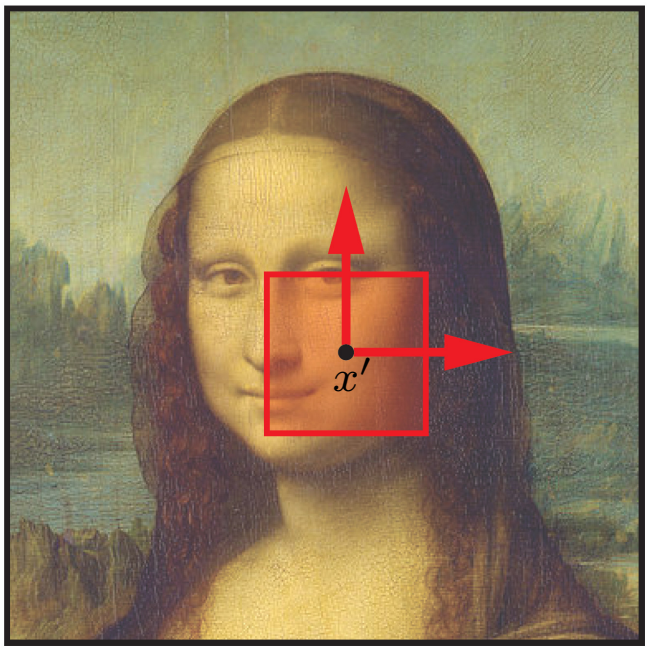


Euclidean

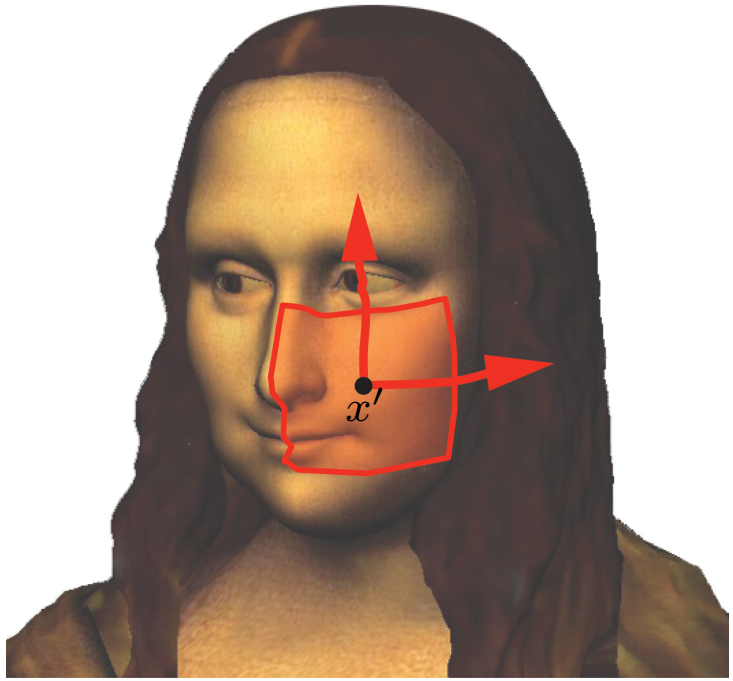


Non-Euclidean

Non-Euclidean convolution?

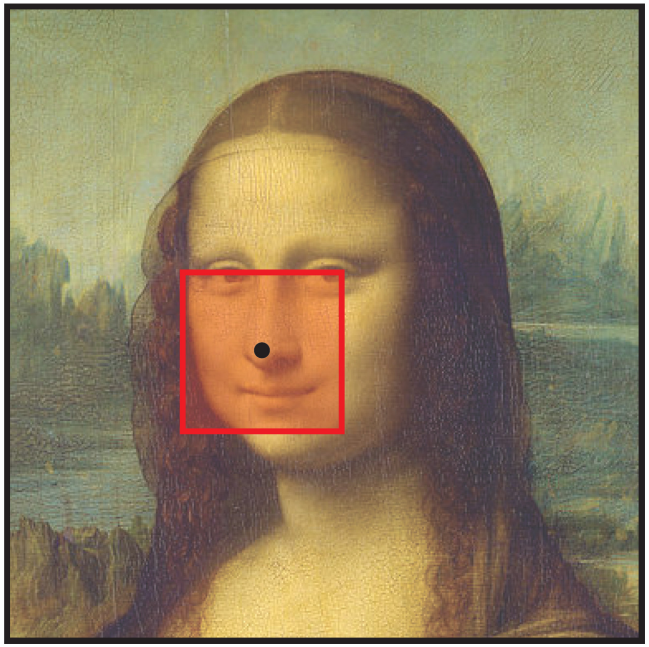


Euclidean

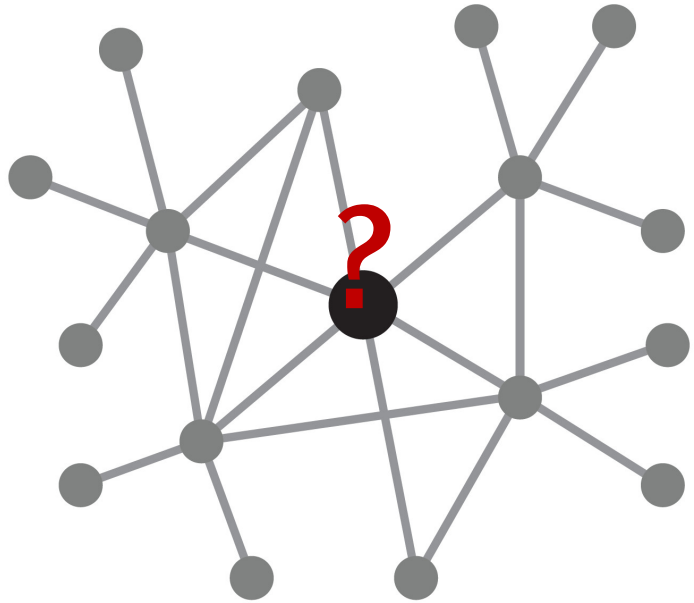


Non-Euclidean

Non-Euclidean convolution?



Image



Graph

Suggested reading

Bronstein et al, 2021

"Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges"

https://arxiv.org/abs/2104.13478

Bronstein et al, 2016

"Geometric deep learning: going beyond Euclidean data"

https://arxiv.org/abs/1611.08097