GEOMETRIC DEEP LEARNING (**L65**)

Pietro Liò *University of Cambridge*
Petar Veličković *Google DeepMind / University of Cambridge*

Lent Term 2024

CST Part III / MPhil ACS / MPhil MLMI

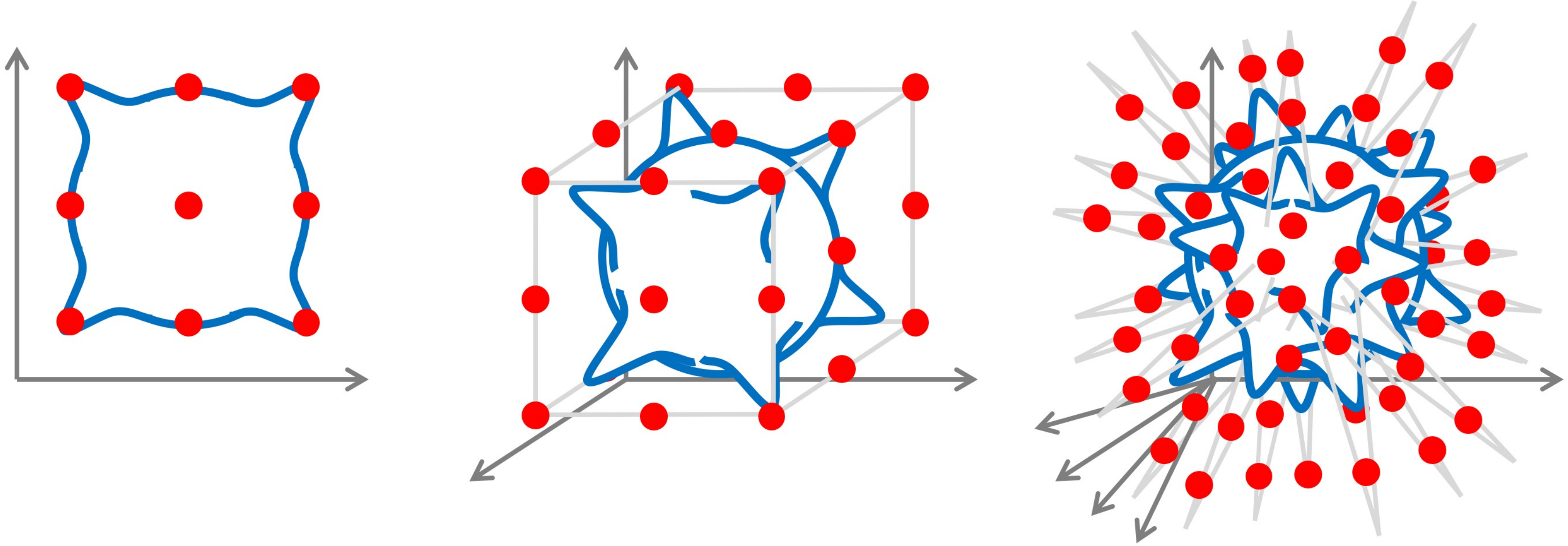
1. INTRODUCTION TO GROUPS AND REPRESENTATIONS

The fundamentals of capturing the regularity in nature

Petar Veličković

Learning in high dimensions

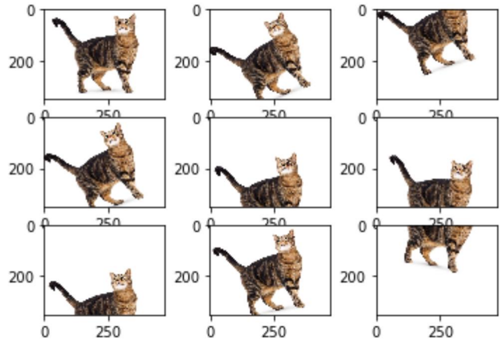
In general, learning functions in high dimensions is **intractable**
Number of samples required grows *exponentially* with dimension



Geometry to the rescue!

We can inject *assumptions* about **geometry** through *inductive biases*
Restrict the functions to ones that *respect* the geometry.
This can make the high-dimensional problem more tractable!

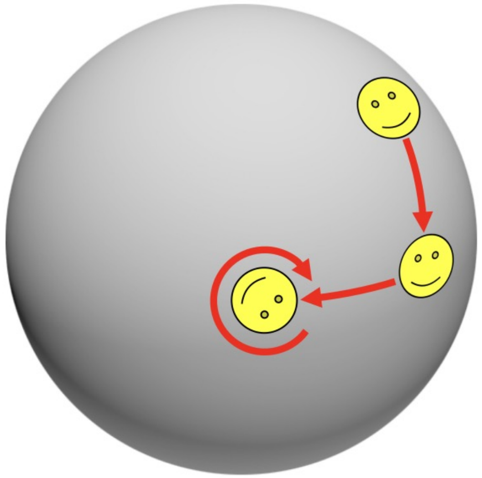
Some popular examples:
- **Image** data should be processed independently of **shifts**



Geometry to the rescue!

We can inject *assumptions* about **geometry** through *inductive biases*
Restrict the functions to ones that *respect* the geometry.
This can make the high-dimensional problem more tractable!

Some popular examples:
- **Image** data should be processed independently of **shifts**
- **Spherical** data should be processed independently of **rotations**

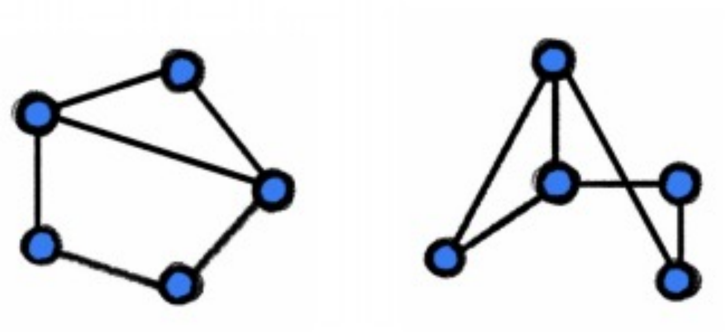


Geometry to the rescue!

We can inject *assumptions* about **geometry** through *inductive biases*
Restrict the functions to ones that *respect* the geometry.
This can make the high-dimensional problem more tractable!

Some popular examples:

- **Image** data should be processed independently of **shifts**
- **Spherical** data should be processed independently of **rotations**
- **Graph** data should be processed independently of **isomorphism**



Geometry to the rescue!

We can inject *assumptions* about **geometry** through *inductive biases*
Restrict the functions to ones that *respect* the geometry.
This can make the high-dimensional problem more tractable!

Some popular examples:

- **Image** data should be processed independently of **shifts**
- **Spherical** data should be processed independently of **rotations**
- **Graph** data should be processed independently of **isomorphism**

We will now attempt to **formalise** this!

A roadmap for our formalisation

To be able to talk about geometry of *data*, we need to formalise *where* the data lives (***domain***) and how to *featurise* it (***signal***)

Once we understand data domains, we can then formalise *symmetries* of those domains (***groups***)

Equipped with groups, we need to formalise *how* they *transform* the data domains (***group actions***)

Deep learning concerns itself with *linear algebra*; we need to be able to talk about group actions as *matrix operations* (***representations***)

Using representations, we can formalise what it means for a deep learning model to *respect symmetries* (***invariance & equivariance***)

The space of signals on a geometric domain

A *signal* on Ω is a function $x : \Omega \to \mathcal{C}$, where:
- Ω is the domain (e.g. set of pixels/nodes/…)
- $\mathcal{C}$ is a vector space, whose dimensions are called *channels*

The space of $\mathcal{C}$-valued signals on Ω is defined as

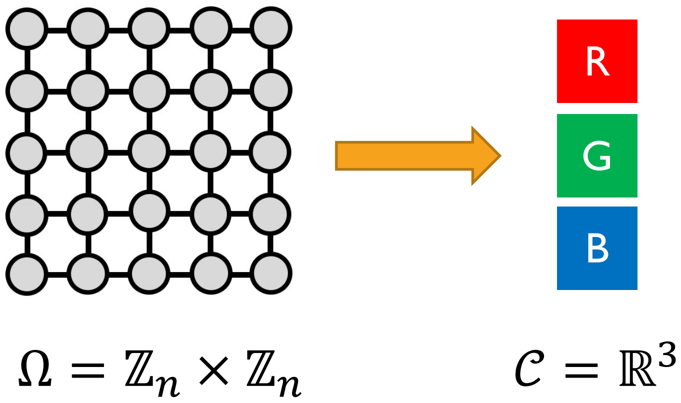
$$\mathcal{X}(\Omega, \mathcal{C}) = \{x : \Omega \to \mathcal{C}\}$$

We will often omit $\mathcal{C}$, and just write $\mathcal{X}(\Omega)$

Signals on *discrete, finite* Ω expressible as *matrices*

$$\mathbf{X} \in \mathbb{R}^{|\Omega| \times \dim \mathcal{C}}$$

where the ith *row* gives *features* of the ith *node*

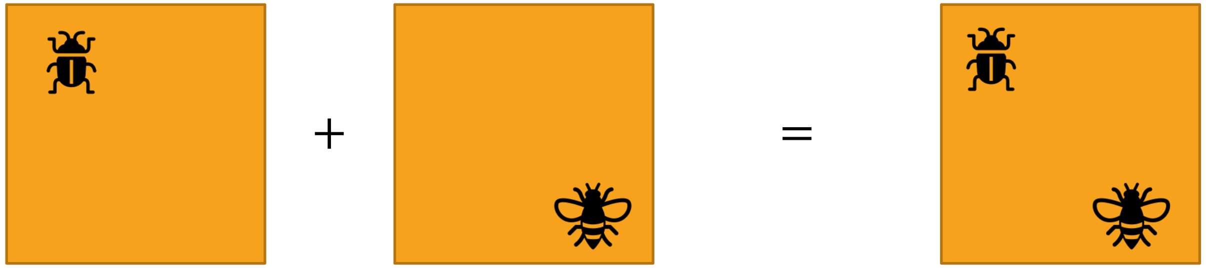


Example: $n \times n$ RGB image

Vector space structure of signals

We can add signals and multiply by scalars:

$$(\alpha x + \beta y)(u) = \alpha x(u) + \beta y(u), \qquad \text{where } \alpha, \beta \in \mathbb{R} \text{ and } u \in \Omega$$



$\Longrightarrow$ The space of signals is a *vector space*! (possibly *infinite dimensional*)

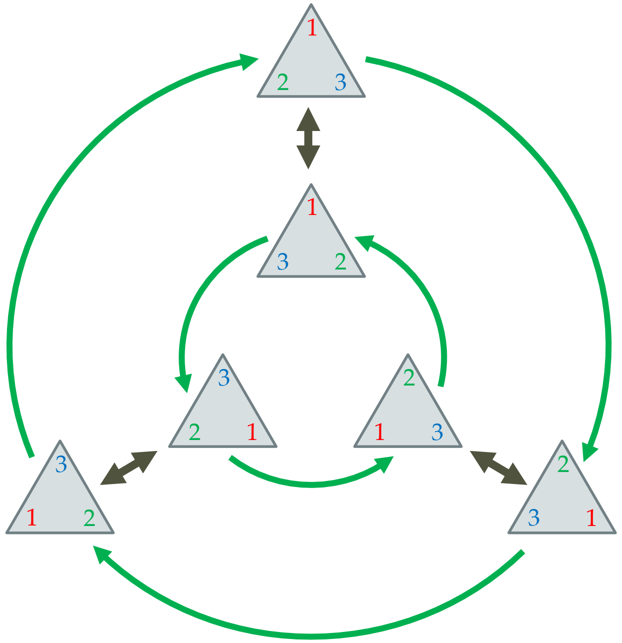
Can also define an *inner product* on signals, given inner product $\langle , \rangle_{\mathcal{C}}$ on $\mathcal{C}$ and a measure μ on Ω ($\Longrightarrow$ The space of signals is a *Hilbert space*!)

$$\langle x, y \rangle = \int_{\Omega} \langle x(u), y(u) \rangle_{\mathcal{C}} \, \mathrm{d}\mu(u)$$

Exercise: Verify that the above satisfies the inner product axioms

Symmetries

A **symmetry** of an object is a transformation of that object that leaves it unchanged



The symmetries of a triangle,
as generated by 120-degree ***rotations*** *R and* ***flips*** *F.*

Symmetry group

A **symmetry** of an object is a transformation of that object that leaves it **unchanged**

Observe that this immediately defines some properties:

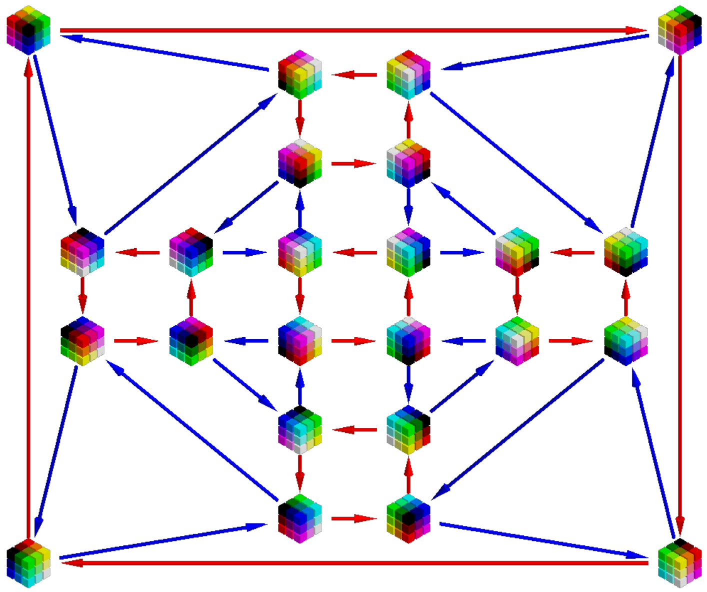
- The **identity** transformation is always a symmetry
- Given two symmetry transformations, their **composition** (doing one after the other) is also a symmetry
- Given any symmetry, it must be **invertible**
- Moreover, its **inverse** is also a symmetry

Collecting all these *axioms* together, we recover a standard mathematical object: the **group**

Abstract groups

A *group* is a set $\mathfrak{G}$ with a binary operation denoted $\mathfrak{g}\mathfrak{h}$ satisfying the following properties:

- *Associativity:* $(\mathfrak{g}\mathfrak{h})\mathfrak{k} = \mathfrak{g}(\mathfrak{h}\mathfrak{k})$ for all $\mathfrak{g}, \mathfrak{h}, \mathfrak{k} \in \mathfrak{G}$
- *Identity*: there exists a unique $\mathfrak{e} \in \mathfrak{G}$ satisfying
$$\mathfrak{g}\mathfrak{e} = \mathfrak{e}\mathfrak{g} = \mathfrak{g}$$
- *Inverse:* for each $\mathfrak{g} \in \mathfrak{G}$ there is a unique inverse $\mathfrak{g}^{-1} \in \mathfrak{G}$, such that $\mathfrak{g}\mathfrak{g}^{-1} = \mathfrak{g}^{-1}\mathfrak{g} = \mathfrak{e}$
- *Closure*: for every $\mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$, we have $\mathfrak{g}\mathfrak{h} \in \mathfrak{G}$



Rotational symmetries of the cube (group O_h)

Symmetry groups, abstract groups & group actions

Symmetry group: a group of transformations $\mathfrak{g} : \Omega \rightarrow \Omega$
The group operation is *composition*

Symmetry groups, abstract groups & group actions

Symmetry group: a group of transformations $\mathfrak{g} : \Omega \to \Omega$
The group operation is *composition*

Abstract group: a set of elements together with a composition rule, satisfying the group axioms

*(an **abstract** group does not directly tell us how to transform data!)*

Symmetry groups, abstract groups & group actions

Symmetry group: a group of transformations $\mathfrak{g} : \Omega \to \Omega$
The group operation is *composition*

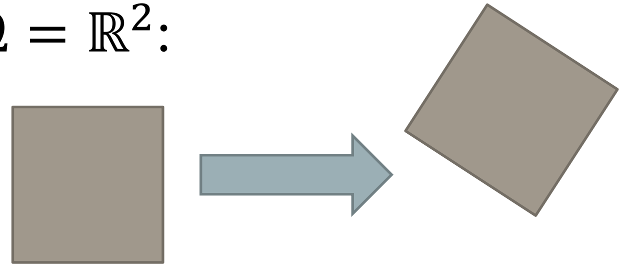
Abstract group: a set of elements together with a composition rule, satisfying the group axioms

Group action: a map $\mathfrak{G} \times \Omega \to \Omega$ (denoted $(\mathfrak{g}, u) \mapsto \mathfrak{g}u$) such that

$$\mathfrak{g}(\mathfrak{h}u) = (\mathfrak{g}\mathfrak{h})u$$
$$\mathfrak{e}u = u$$

e.g.: Euclidean 2D motions $\mathfrak{G} = \mathbb{R}^3$ (angle + translation) acting on $\Omega = \mathbb{R}^2$:

$$(\theta, t_x, t_y)(x, y) \mapsto (x\cos\theta + y\sin\theta + t_x, x\sin\theta + y\cos\theta + t_y)$$

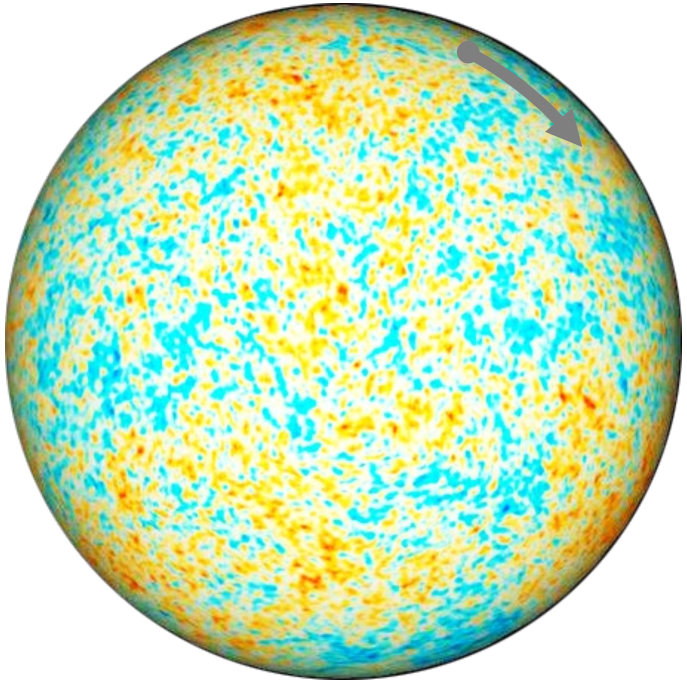


Exercise: Verify this satisfies the group action axioms

Symmetries of Ω acting on signals $\mathcal{X}(\Omega)$

Given an action of $\mathfrak{G}$ on Ω, we automatically obtain an action of $\mathfrak{G}$ on the space of signals $\mathcal{X}(\Omega)$:

$$(\mathfrak{g}\, x)(u) = x(\mathfrak{g}^{-1}u)$$



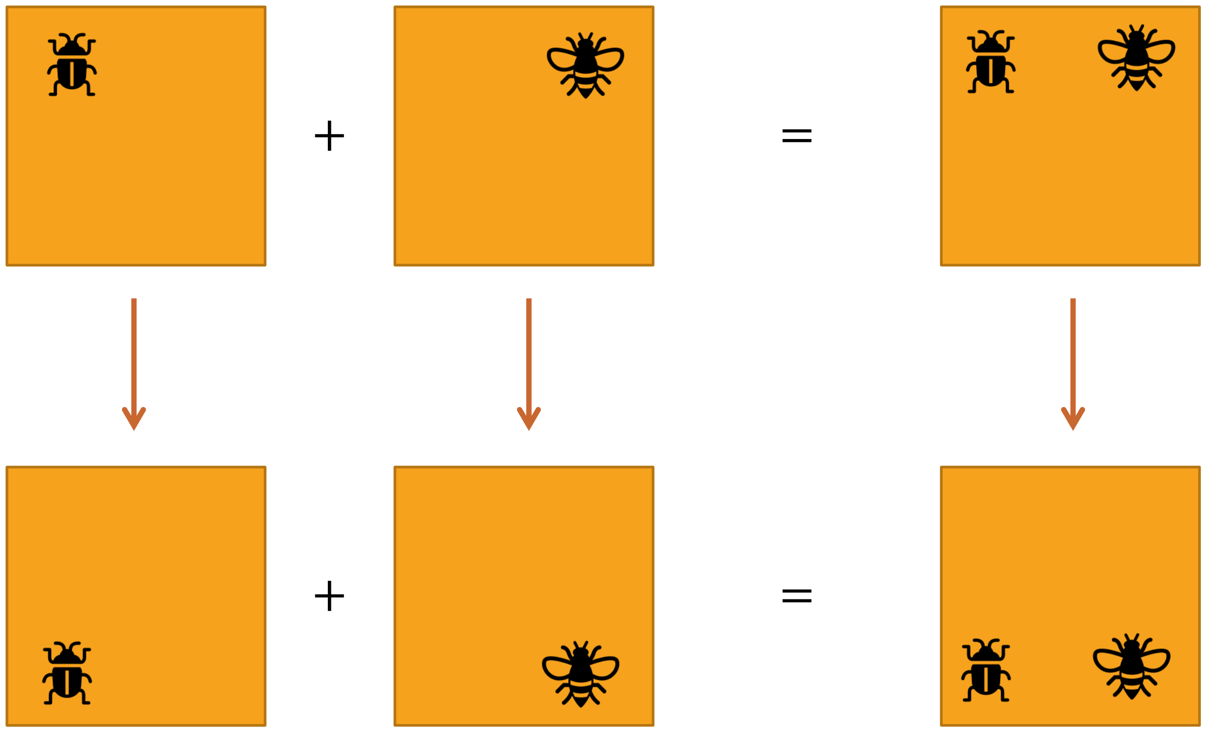
Linearity of the group action

If the signals support a vector space, such a group action on signals

$$(\mathfrak{g}\,x)(u) = x(\mathfrak{g}^{-1}u)$$

is *linear*!

Apply $\mathfrak{g} = (t_x, t_y)$



Linearity of the group action

If the signals support a vector space, such a group action on signals

$$(\mathfrak{g}\, x)(u) = x(\mathfrak{g}^{-1}u)$$

is *linear*!

This is *excellent* news for us, as deep learning is basically *linear algebra*
And we will be able to describe group actions as *matrix multiplication*!

For the time being, we will assume our domain is *discrete* and *finite*
That is, that we can represent our domain using a matrix $\mathbf{X} \in \mathbb{R}^{|\Omega| \times k}$

Group representations

A *real representation* of $\mathfrak{G}$ on a *finite vector space* $\mathcal{X}$ is a *map* $\rho_{\mathcal{X}} : \mathfrak{G} \to \mathbb{R}^{n \times n}$, assigning to each element $\mathfrak{g} \in \mathfrak{G}$ an *invertible matrix* $\rho_{\mathcal{X}}(\mathfrak{g})$, and satisfying

$$\rho_{\mathcal{X}}(\mathfrak{g}\mathfrak{h}) = \rho_{\mathcal{X}}(\mathfrak{g})\rho_{\mathcal{X}}(\mathfrak{h}), \qquad \forall \mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$$

Group representations

A *real representation* of $\mathfrak{G}$ on a *finite vector space* $\mathcal{X}$ is a *map* $\rho_{\mathcal{X}} : \mathfrak{G} \to \mathbb{R}^{n \times n}$, assigning to each element $\mathfrak{g} \in \mathfrak{G}$ an *invertible matrix* $\rho_{\mathcal{X}}(\mathfrak{g})$, and satisfying

$$\rho_{\mathcal{X}}(\mathfrak{g}\mathfrak{h}) = \rho_{\mathcal{X}}(\mathfrak{g})\rho_{\mathcal{X}}(\mathfrak{h}), \qquad \forall \mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$$

The dimensionality of the matrix produced by $\rho_{\mathcal{X}}$ may depend on many factors, such as the size of the corresponding Ω

$\rho_{\mathcal{X}}$ does *not* need to assign *different* matrices to *different* elements of $\mathfrak{G}$
(if it does, then it is a *faithful representation*)

Representations can be easily generalised to *infinite* spaces---the map $\rho_{\mathcal{X}} : \mathfrak{G} \to \left(\mathcal{X}(\Omega) \to \mathcal{X}(\Omega)\right)$ would output an *invertible linear function*
(strictly speaking, $\rho_{\mathcal{X}} : \mathfrak{G} \to \mathrm{GL}(\mathcal{X})$, where $\mathrm{GL}(\mathcal{X})$ is the *general linear group* over $\mathcal{X}$)

Group representations

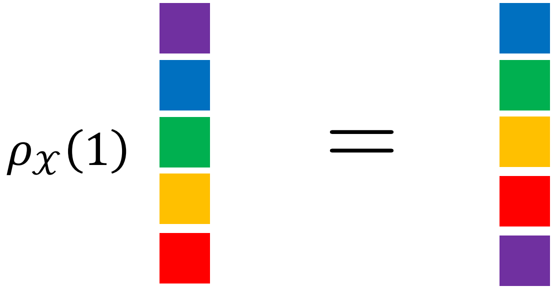
A *real representation* of $\mathfrak{G}$ on a *finite vector space* $\mathcal{X}$ is a *map* $\rho_{\mathcal{X}} : \mathfrak{G} \to \mathbb{R}^{n \times n}$, assigning to each element $\mathfrak{g} \in \mathfrak{G}$ an *invertible matrix* $\rho_{\mathcal{X}}(\mathfrak{g})$, and satisfying

$$\rho_{\mathcal{X}}(\mathfrak{g}\mathfrak{h}) = \rho_{\mathcal{X}}(\mathfrak{g})\rho_{\mathcal{X}}(\mathfrak{h}), \qquad \forall \mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$$

Example:

- The group of 1D (circular) shifts, $\mathfrak{G} = (\mathbb{Z}, +)$
- The domain $\Omega = \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$ (e.g. short audio signal)
- The action of $\mathfrak{g} = n$ on $u \in \Omega$: $(n, u) \mapsto n + u \pmod 5$
- The representation on $\mathcal{X}(\Omega)$:

$$\rho_{\mathcal{X}}(n) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}^n$$



Exercise: Derive the representation for the group of *90-degree rotations* on 3x3 grids

Group invariance

We can now *formally* describe how to *exploit* the symmetries in $\mathfrak{G}$!

Group invariance

We can now *formally* describe how to *exploit* the symmetries in $\mathfrak{G}$!

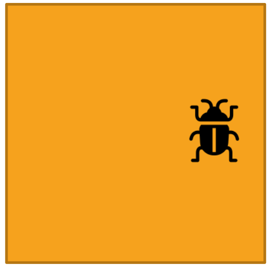
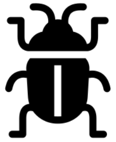
A function $f : \mathcal{X}(\Omega) \to \mathcal{Y}$ is $\mathfrak{G}$-*invariant* if $f(\rho_{\mathcal{X}}(\mathfrak{g})x) = f(x)$ for all $\mathfrak{g} \in \mathfrak{G}$, i.e., its output is unaffected by the group action on the input.

Group invariance

We can now *formally* describe how to *exploit* the symmetries in $\mathfrak{G}$!

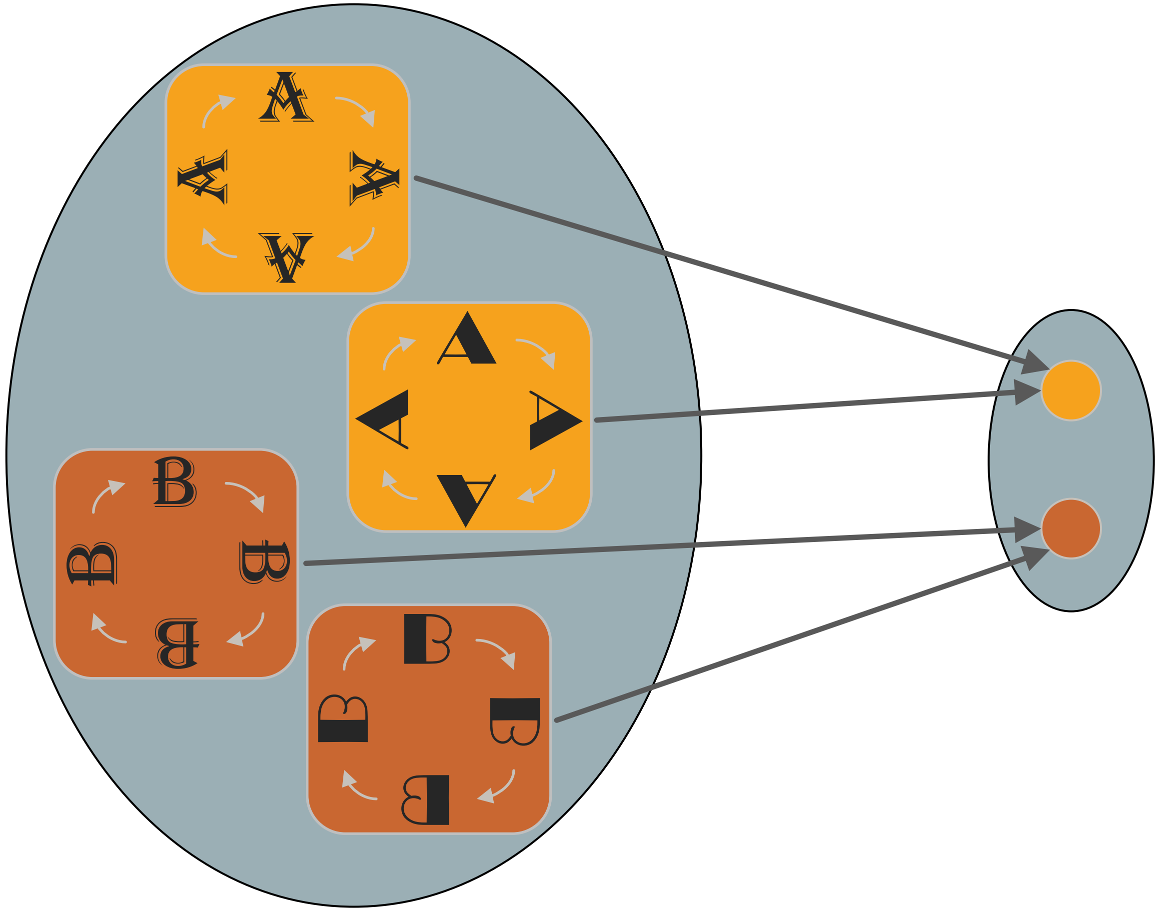
A function $f : \mathcal{X}(\Omega) \to \mathcal{Y}$ is $\mathfrak{G}$-*invariant* if $f(\rho_{\mathcal{X}}(\mathfrak{g})x) = f(x)$ for all $\mathfrak{g} \in \mathfrak{G}$, i.e., its output is unaffected by the group action on the input.

e.g. **image classification**: output class won't depend on image **shifts**

$f($  $)$ $=$ $f($  $)$ $=$ 

$\rho_{\mathcal{X}}(\mathfrak{g})x$ $\qquad$ x

Orbits and equivalence relations



$$O_x = \{\mathrm{g}x \mid x \in \mathcal{X}, \mathrm{g} \in \mathfrak{G}\}$$

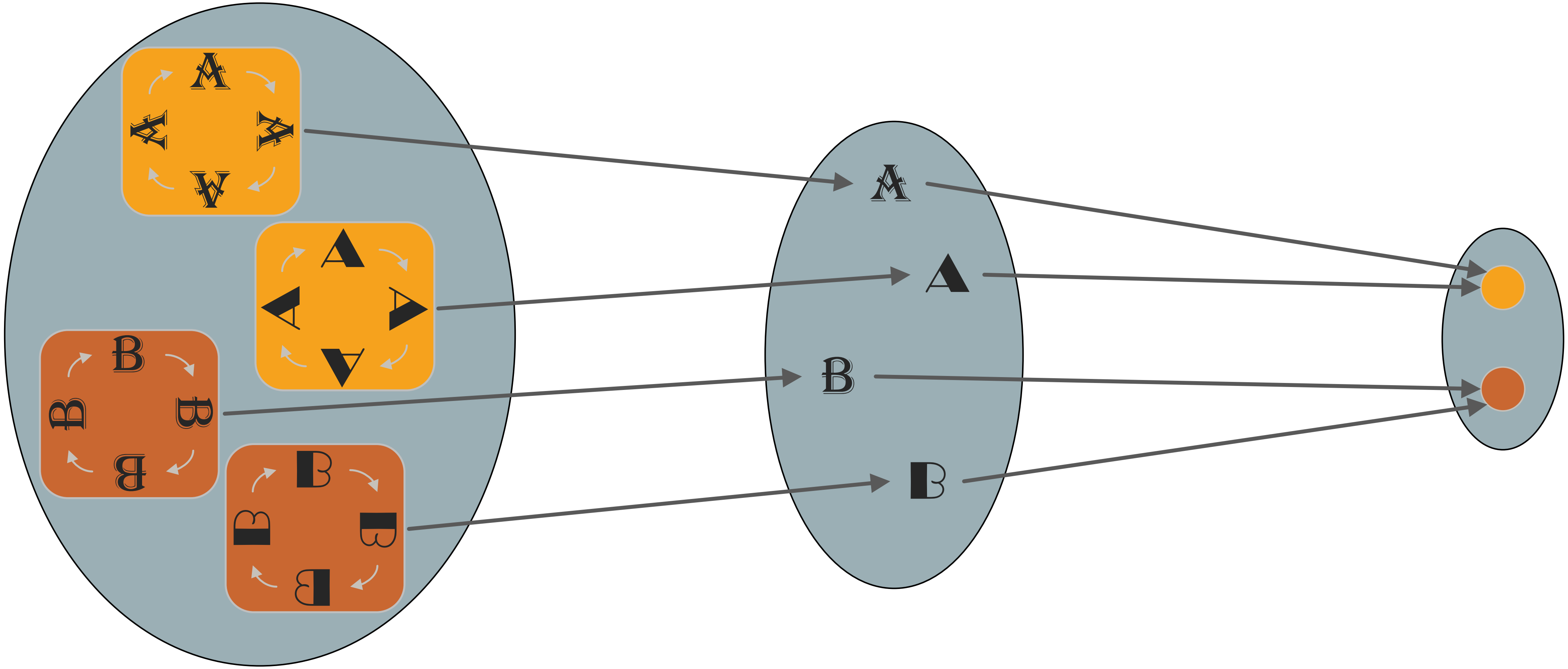
$\mathfrak{G}$-equivalence

$$x \sim_{\mathfrak{G}} y \Leftrightarrow \exists \mathrm{g} \in \mathfrak{G} : \mathrm{g}x = y$$

Satisfies the axioms of an equivalence relation:

1. Reflexivity: $x \sim_{\mathfrak{G}} x$
   - (Because $\mathfrak{G}$ contains the identity)
2. Transitivity:
   $x \sim_{\mathfrak{G}} y \wedge y \sim_{\mathfrak{G}} z \Leftrightarrow x \sim_{\mathfrak{G}} z$
   - (Because $\mathfrak{G}$ is closed under composition)
3. Symmetry: $x \sim_{\mathfrak{G}} y \Leftrightarrow y \sim_{\mathfrak{G}} x$
   - (Because $\mathfrak{G}$ is closed under inverses)

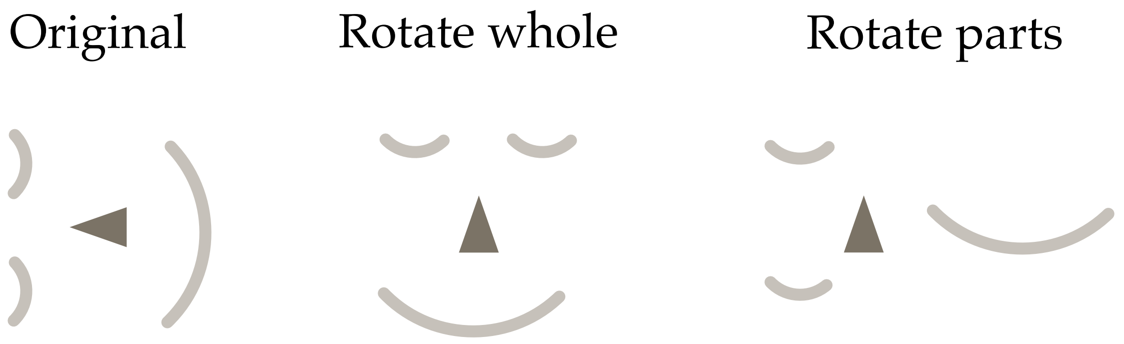
𝔊-*invariant representations*



The problem with invariance

Invariance is suitable when we need a *single* output over the *entire* domain. What if we need an output in *each* domain element?

Also, even if a single output is OK, making the intermediate representations invariant may lose *critical* information:



The *relative pose* of object parts contains critical information (Hinton *et al.*, ICANN'11)

Group equivariance

We proceed to define a more *fine-grained* notion of regularity:

A function $f : \mathcal{X}(\Omega) \to \mathcal{Z}(\Omega)$ is $\mathfrak{G}$-*equivariant* if, for all $\mathfrak{g} \in \mathfrak{G}$, $f(\rho_{\mathcal{X}}(\mathfrak{g})x) = \rho_{\mathcal{Z}}(\mathfrak{g})f(x)$, i.e., applying a group action on the input affects the output in the same way.

Group equivariance

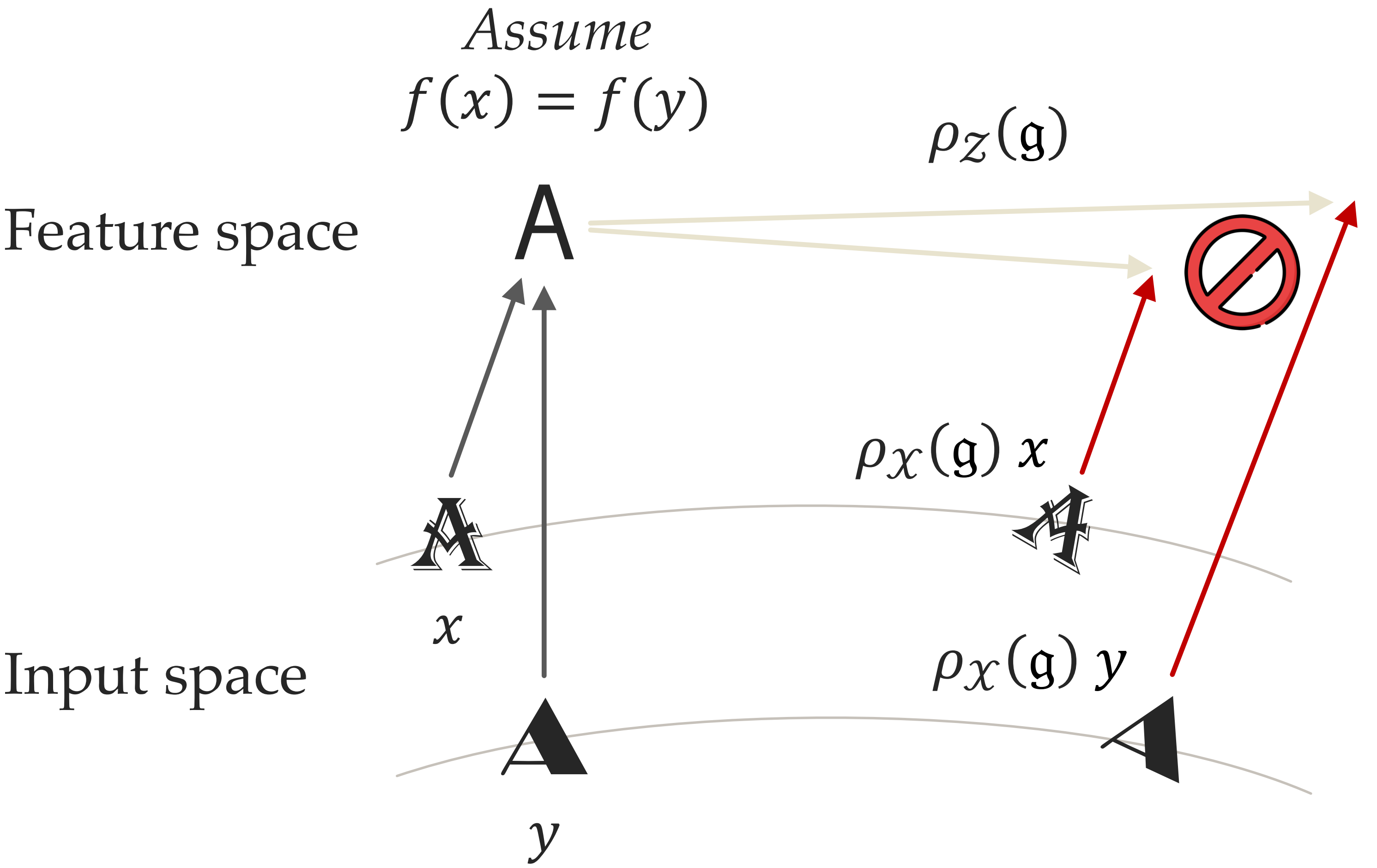
We proceed to define a more *fine-grained* notion of regularity:

A function $f : \mathcal{X}(\Omega) \to \mathcal{Z}(\Omega)$ is $\mathfrak{G}$-*equivariant* if, for all $\mathfrak{g} \in \mathfrak{G}$, $f(\rho_{\mathcal{X}}(\mathfrak{g})x) = \rho_{\mathcal{Z}}(\mathfrak{g})f(x)$, i.e., applying a group action on the input affects the output in the same way.

e.g. **image segmentation**: segmentation mask must **follow** any shifts in the input

Note that invariance is a *special case* of equivariance (for which $\rho_{\mathcal{Z}}$?)

Equivariance as symmetry-consistent generalisation

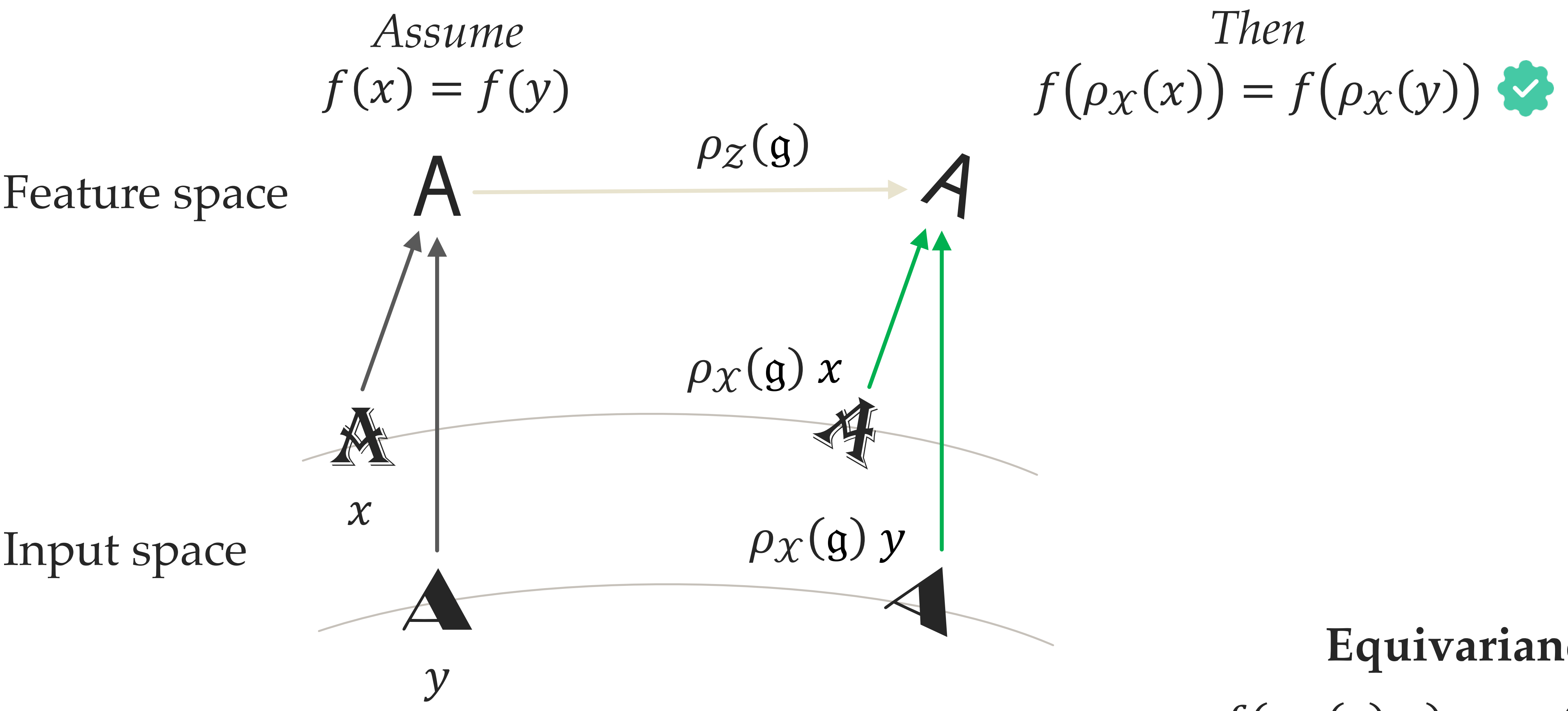


This **cannot** happen!
Equivariant net must generalize consistently across the whole orbit.

Equivariance:

$$f(\rho_X(\mathfrak{g})\, x) = \rho_Z(\mathfrak{g})\, f(x)$$

Equivariance as symmetry-consistent generalisation



Equivariance:

$$f(\rho_X(\mathfrak{g})\, x) = \rho_Z(\mathfrak{g})\, f(x)$$

The building blocks of Geometric Deep Learning

Let Ω and Ω' be domains, $\mathfrak{G}$ a symmetry group over Ω.
Write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω.

We define the following building blocks:

Linear $\mathfrak{G}$-equivariant layer $B : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$,
satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Nonlinearity $\sigma : \mathcal{C} \rightarrow \mathcal{C}'$ applied element-wise as $\big(\boldsymbol{\sigma}(x)\big)(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

$\mathfrak{G}$*-invariant layer (global pooling) $A : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$,*
satisfying $A(\mathfrak{g}.x) = A(x)$for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

The building blocks of Geometric Deep Learning

Let Ω and Ω' be domains, $\mathfrak{G}$ a symmetry group over Ω.
Write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω.

We define the following building blocks:

Linear $\mathfrak{G}$-equivariant layer $B : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{X}(\Omega', \mathcal{C}')$,
satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

***Linear* equivariant layer**

Nonlinearity $\sigma : \mathcal{C} \to \mathcal{C}'$ applied element-wise as $\big(\boldsymbol{\sigma}(x)\big)(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

$\mathfrak{G}$-invariant layer (global pooling) $A : \mathcal{X}(\Omega, \mathcal{C}) \to \mathcal{Y}$,
satisfying $A(\mathfrak{g}.x) = A(x)$for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Invariant "tail" layer
(if necessary)

The building blocks of Geometric Deep Learning

Let Ω and Ω' be domains, $\mathfrak{G}$ a symmetry group over Ω.
Write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω.

We define the following building blocks:

Linear $\mathfrak{G}$*-equivariant layer* $B : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$,
satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

Activation function

Nonlinearity $\sigma : \mathcal{C} \rightarrow \mathcal{C}'$ applied element-wise as $(\boldsymbol{\sigma}(x))(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

$\mathfrak{G}$*-invariant layer (global pooling)* $A : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$,
satisfying $A(\mathfrak{g}.x) = A(x)$for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

The building blocks of Geometric Deep Learning

Let Ω and Ω' be domains, $\mathfrak{G}$ a symmetry group over Ω.
Write $\Omega' \subseteq \Omega$ if Ω' can be considered a compact version of Ω.

We define the following building blocks:

Linear $\mathfrak{G}$-equivariant layer $B : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$,
satisfying $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

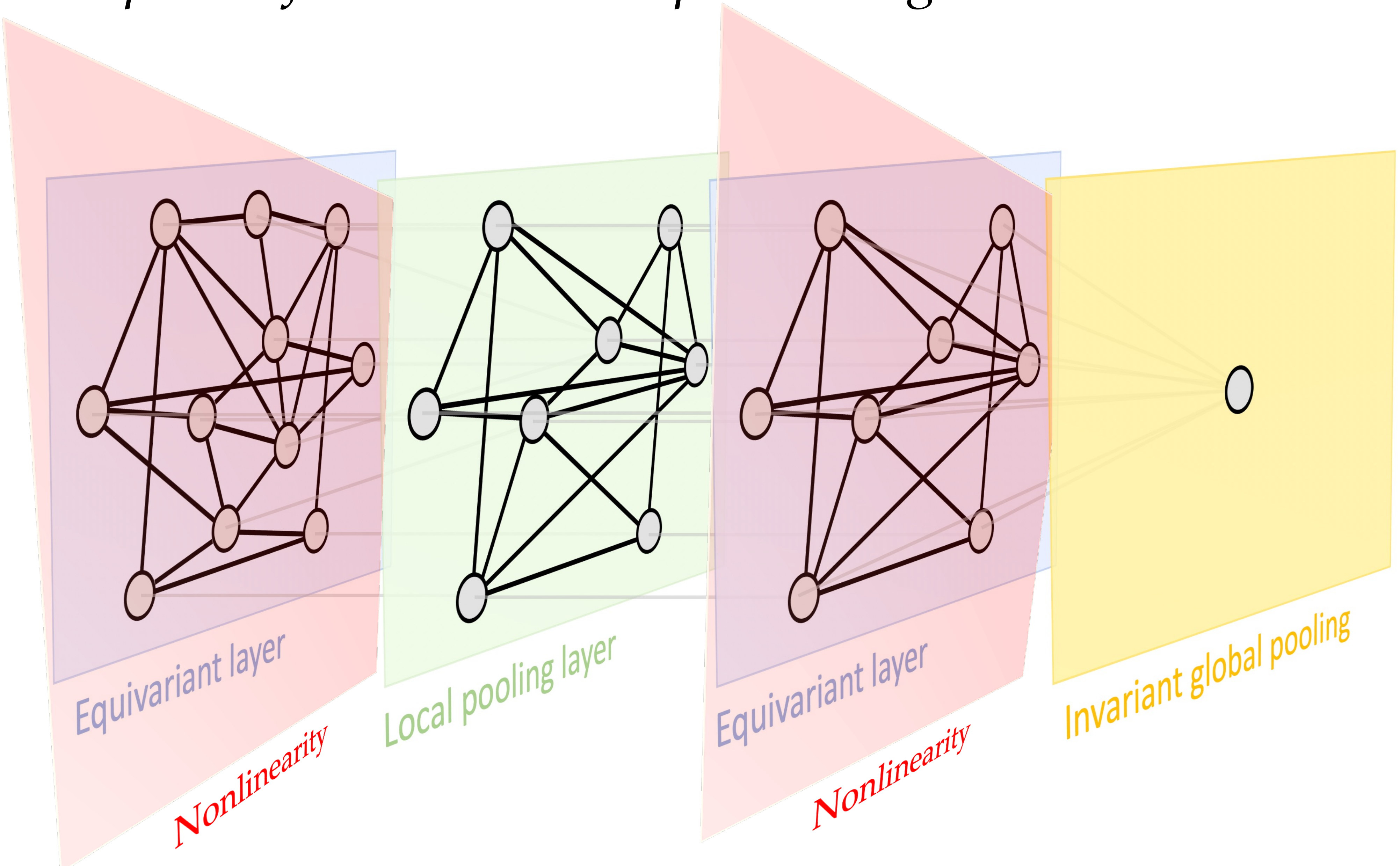
Nonlinearity $\sigma : \mathcal{C} \rightarrow \mathcal{C}'$ applied element-wise as $\big(\boldsymbol{\sigma}(x)\big)(u) = \sigma(x(u))$.

Local pooling (coarsening) $P : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$, such that $\Omega' \subseteq \Omega$.

Coarsening layer

$\mathfrak{G}$-*invariant layer (global pooling)* $A : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$,
satisfying $A(\mathfrak{g}.x) = A(x)$for all $\mathfrak{g} \in \mathfrak{G}$ and $x \in \mathcal{X}(\Omega, \mathcal{C})$.

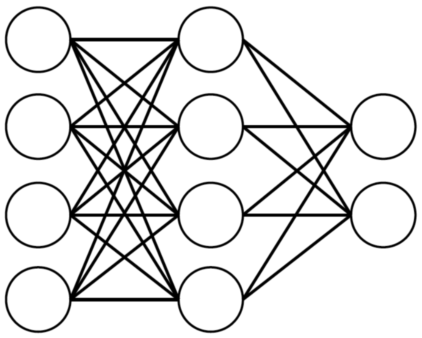
The blueprint of Geometric Deep Learning



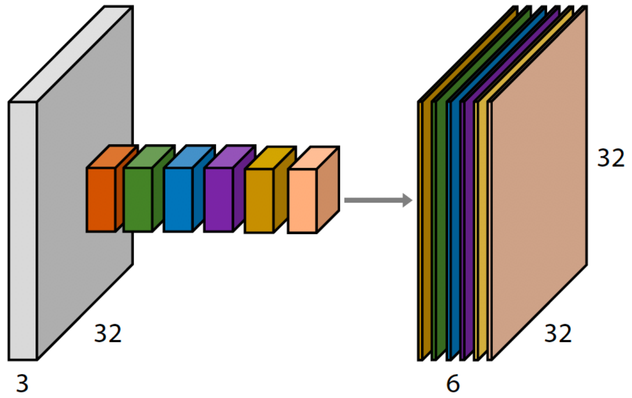
Popular architectures as instances of GDL blueprint

Architecture	**Domain Ω**	**Symmetry Group $\mathfrak{G}$**
CNN	Grid	Translation
Spherical CNN	Sphere / $SO(3)$	Rotation $SO(3)$
Mesh CNN	Manifold	Isometry $\mathrm{Iso}(\Omega)$ / Gauge Symmetry $SO(2)$
GNN	Graph	Permutation Σ_n
Deep Sets	Set	Permutation Σ_n
Transformer	Complete Graph	Permutation Σ_n
E(3) GNN	Geometric Graph	Permutation $\Sigma_n \times$ Euclidean $E(3)$
LSTM	1D Grid	Time warping

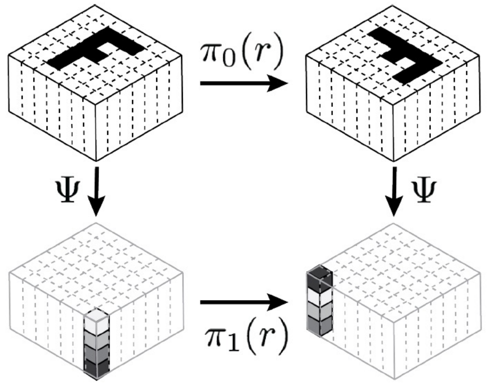
Architectures of interest



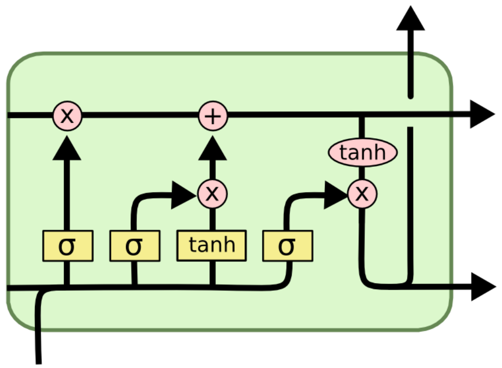
Perceptrons
Function regularity



CNNs
Translation

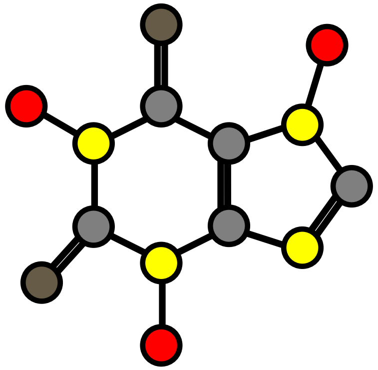


Group-CNNs
Global groups

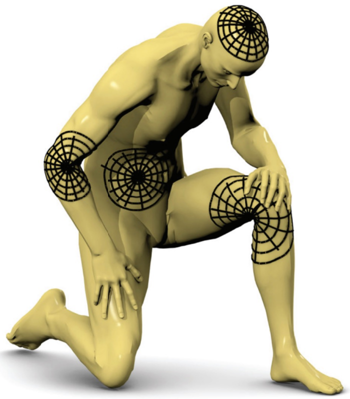


LSTMs
Time warping

Deep Sets / Transformers
Permutation



GNNs
Permutation



Intrinsic CNNs
Isometry / Gauge choice

…now it's our turn to study geometry! ☺