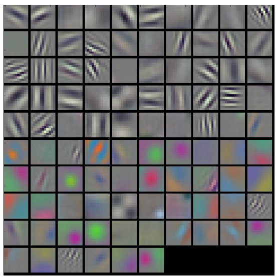
Physics-inspired equivariant machine learning

Soledad Villar (Johns Hopkins University)

UC Louvain June 28, 2022

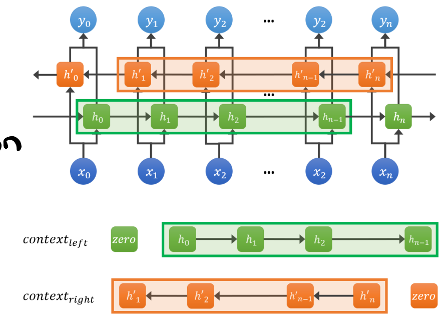
Symmetries in deep learning



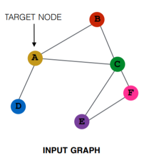
CNNs exploit translation and rotation symmetries in natural images by applying the same convolutional filters at different locations of the image

Image credit: Stanford CNN course

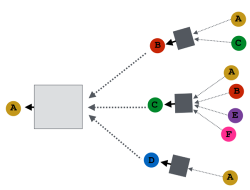
RNNs exploit time translation symmetry by applying the same recurrent unit at different locations



Credit: Zhao et al '19



Credit: Leskovec



GNNs learn functions on graphs that are invariant to node relabeling. (Permutation invariance)
(equivariant) (equivariance)

Example: message passing neural networks (MPNN)

Invariance / Equivariance

Exact symmetries

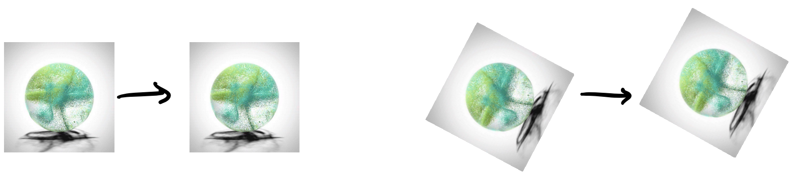
G a group acting on dataset X

$F: X \to Y$ invariant if $F(g \cdot x) = F(x) \quad \forall g \in G,\ x \in X$

→ "Rose"

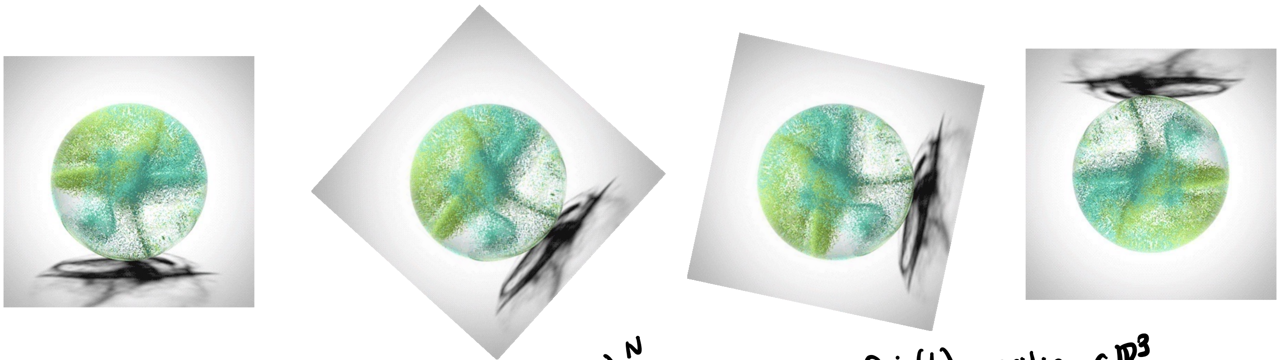
If G also acts in Y

$H: X \to Y$ equivariant $H(g \cdot x) = g \cdot H(x) \quad \forall g \in G,\ x \in X$



Example Rotation equivariance

- Dynamical systems, particle simulations (N-body problems)



Initial conditions $\left(q_i(0), p_i(0)\right)_{i=1}^{N}$

$q_i(t)$ position $\in \mathbb{R}^3$
$p_i(t)$ momentum $\in \mathbb{R}^3$

Predictions $\left(q_i(t_k), p_i(t_k)\right)_{i=1}^{N}$ $k=1,\dots T$

$$F_\theta(q(0), p(0)) = [q(t), p(t)]$$

Applications

- **Images** : group equivariant CNNs (Cohen, Welling '16, ...)
- **Graphs** : graph neural networks (Duvenaud et al, Bruna et al, Gilmer et al, ...)
- **Particle systems** : Irreducible representations (Maron et al, Kondor et al ...)

  Invariants (Villar et al ...)

Equivariances and invariants in machine learning models

- Provide the correct "inductive bias": Turbulence simulations Rose Yu '22
- Improve learning: better sample complexity
  smaller generalization error
  out-of-distribution generalization
- Source of interesting math questions:
  - learn invariances from data: Augerino, Benton et al '21; Lie PCA Cahill, Mixon, Parshall '20
  - design algorithms that exploit invariances / invariants
  - quantify "how much we gain": Bietti, Venturi, Bruna '21 Elesedy '21, Mei, Misiakiewicz, Montanari '21

Symmetries in physics

Particular case of Physics-informed ML
Karniadakis et al. 21'

- **Motivation**
  - Laws of physics obey symmetries
  - Symmetric forms provide strong constraints on possible laws of physics
  - Symmetric forms and Einstein/Ricci summation notation enabled discovery of general relativity and various particle interactions

- **Examples**
  - Symmetries of classical physics $O(d), SO(d), E(d), O(1,d), IO(1,d), S_n$
  - Symmetries of quantum mechanics $U(1), U(2), SU(2), U(3), SU(3), C, P, T$
  - Units equivariance symmetries (non-compact)

$O(d) = RR^T = I$
$O(d,1) = R \Lambda R^T = \Lambda$

$\Lambda = \begin{pmatrix} -1 & & 0 \\ & 1 & \\ 0 & & 1 \end{pmatrix}$

- **Noether's theorem** [Emmy Noether 1915]

  To every differentiable symmetry generated by local actions there corresponds a conservation law

Example (Noether's theorem)

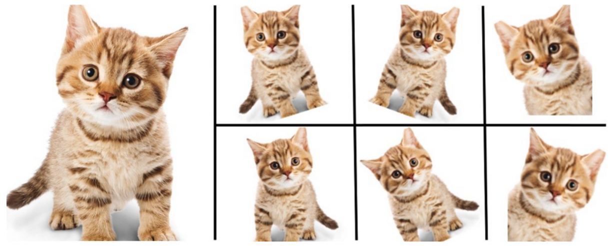
- Space translation symmetry
  $\longleftrightarrow$ conservation of momentum

- $O(d)$ - symmetry
  $\longleftrightarrow$ conservation of angular momentum

- Time translation symmetry
  $\longleftrightarrow$ conservation of energy

How are symmetries implemented?

- Data augmentation
  Li, Dobriban '20
- Loss function penalties
  - concerved quantities
- Architectural design
  - Approximate symmetries (CNN)
  - Exact symmetries
    - Weight sharing (message passing) (group convolutions) Cohen, Welling '19 Ravanbakhsh Rose YU '21, '20. Weiler '21
    - Parameterization of symmetry preserving functions Kondor '18 Maron '18 Cohen '18
    - Symmetries as constraints Finzi et al '21
    - Irreducible representations Kondor, Thomas Smidt '18, Fuchs '20
    - Steerable CNNs Cohen '17, Welling..



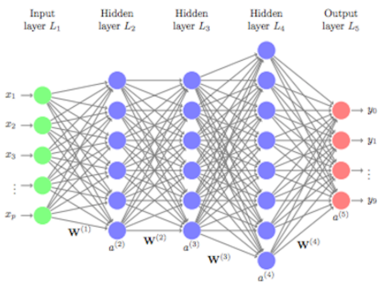
Enlarge your Dataset

Credit: Bharath Raj

Equivariant architectures (based on feed forward neural networks)

- Kondor 2018, Maron et al 2019 ....

Feed Forward NEURAL NETWORK



pointwise non-linear activations — input

$$F(v) = \theta \circ L_n \circ \dots L_2 \circ \theta \circ L_1 (v)$$

linear layers

Idea: Replace linear layers by linear equivariant/invariant layers

$$L_i(Qv) = Q . L_i(v)$$

Issue #1: Not many linear equivariant/invariant functions. What linear rotation invariant functions can you think of?

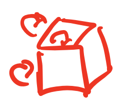
Issue #2: What are compatible activation functions?

They depend on the group: permutations (all work)
rotations (??)

Solution to Issue #1: Extend the action to tensors:

$$L_i : (\mathbb{R}^d)^{\otimes k_i} \longrightarrow (\mathbb{R}^d)^{\otimes k_{i+1}}$$

equivariant linear function



Maron '19

$$L_i(Qv) = Q L_i(v)$$

$$Q(v \otimes \dots \otimes v) = Qv \otimes \dots \otimes Qv$$

Q: How to parametrize linear equivariant functions?

- **Irreducible representation approach:**

$\rho_k : G \to GL(\mathbb{R}^{d^{\otimes k}})$ $\quad \rho_k \leftrightarrow \rho_{k'}$

$\rho : G \to GL(V)$ group representation

$\rho(g)(v) = g * v$ extend to tensor product $\rho_k = \bigotimes_{r=1}^{k} G \to GL((\mathbb{R}^d)^{\otimes k})$

Linear equivariant map $L_i \leftrightarrow$ map between representations

$L_i \circ \rho_{k_i}(g) = \rho_{k_{i+1}}(g) \circ L_i \quad \forall g \in G$

Easy to parameterize using IRREDUCIBLE REPRESENTATIONS

$$\rho_{k_i} = \bigoplus_{\ell=1}^{T_{k_i}} T_\ell$$

$$\bigotimes_{s=1}^{K} \rho_s = \bigoplus_{\ell=1}^{T} T_\ell$$

- Fuchs
- Thomas

This identification is given by the Clebsh-Gordan coefficients
- Known for SO(3) but not in general

Dym and Maron 2021 - This approach universally approximates all SO(3) equiv functions If arbitrary high order tensors are involved

- **Solving a large linear system** Finzi et al '21 (EMLP)

Main idea:
- the space of linear equivariant functions $f : \mathbb{R}^d \to \mathbb{R}^d$ is a linear subspace of $\mathbb{R}^{d \times d}$
- Consider constraints of the form $\{ f(g_i v_i) = g_i f(v_i) \quad i = 1 \ldots n \}$ (linear constraints)
  v_i, g_i random
- Solve a linear system of equations

(the system is large but you only solve it once to get the parameterization of the network)

- Scalars approach (Villar et al '21)

- Main idea: First fundamental theorem of invariant theory (Weyl 1946)

Characterization of $O(d)$-invariant functions

$f:(\mathbb{R}^d)^n \to \mathbb{R}$ is $O(d)$-invariant if and only if

- $f(v_1, \dots v_n) = \tilde{f}\left((v_i^T v_j)_{i,j=1}^n \right)$

Proof • $V = \begin{pmatrix} | & & | \\ v_1 & \dots & v_n \\ | & & | \end{pmatrix}$, $M = V^T V$. Consider the Cholesky decomposition of $M = L^T L$ then $L = V \cdot Q$ For some $Q \in O(d)$. In words you can recover $v_1 \dots v_n$ from the inner products up to orthogonal transformations.

- Physics point of view: All scalars can be written in Einstein summation notation

Characterization of $SO(d)$-invariant functions

$f:(\mathbb{R}^d)^n \to \mathbb{R}$ is $SO(d)$-invariant if and only if

$$f(v_1, \dots, v_n) = \tilde{f}\left((v_i^T v_j)_{i,j=1}^n ,\ \det(v_{i_1} \dots v_{i_d})_{i_1 \dots i_d \in \binom{[n]}{d}} \right)$$

Characterization of Lorentz-invariant functions

$f:(\mathbb{R}^{d+1})^n \to \mathbb{R}$ is Lorentz-invariant if and only if

$$f(v_1, \dots, v_n) = \tilde{f}\left(\langle v_i, v_j \rangle_M{}_{i,j=1}^n \right)$$

where $\langle (t, x), (t', x') \rangle_M = t t' - x^T x'$ ← Minkowski "inner product"

$R \Lambda R^T = \Lambda \quad \Lambda = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ $O(1, d)$

Do we need all the inner products?

$f:(\mathbb{R}^d)^n \to \mathbb{R}$ invariant

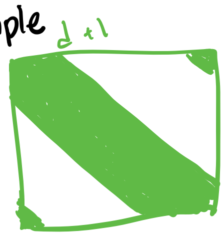
$f(Qv_1,\dots Qv_n) = f(v_1\dots v_n)$

f is invariant if and only if $f(\underbrace{v_1\dots v_n}_{n\ d\text{-vectors}}) = \tilde{f}(\underbrace{\langle v_i, v_j\rangle_{i,j=1}^n}_{n \times n \text{ scalars}})$

Do we need all nxn scalars? NO:

- Rigity theory of Gram matrices
- Low rank matrix completion

Example

$\hat{M} =$ 

$O(n(d+1))$ scalars

determine $M = V^T V$

$f(v_1,\dots,v_n) = \tilde{f}(M) = \hat{f}(\hat{M})$

Open problems: quantify the approximation error if a subset $\hat{M}$ is used instead of M

can we get away with fewer scalars for SO(d)?

O(d)-Equivariant vector functions:

$h: (\mathbb{R}^d)^n \to \mathbb{R}^d$ is O(d)-equivariant if and only if

$$h(v_1 \dots v_n) = \sum_{i=1}^{n} \underbrace{f_i(v_1 \dots v_n)}_{\text{O(d) invariant scalar function}} \cdot v_i$$

d is any dimension!

Proof (sketch)

- h O(d)-equivariant $\Rightarrow h(v_1 \dots v_n) \in \text{span}(v_1 \dots v_n)$
- $h(v_1, \dots v_n) = \sum_{i=1}^{n} f_i(v_1 \dots v_n)\, v_i$ coefficients functions can taken to be invariant
- If h polynomial $\Rightarrow f_i$'s can be chosen to be polynomials

Open problem: If h is continuous it's not true that f_i can be chosen continuous
what condition in h guarantees f_i are continuous ??

APPROACH: input $\to$ scalars $\to$ f_1 … f_n (MLP) $\to \oplus \to$ output (UNIVERSAL)

Open problem: Prove a Stone-Weierstrass theorem for this method

Example: Electromagnetic force law. Particle (q, r, v)

(q = charge, r = position, v = velocity)

$$F = \underbrace{\sum_{i=1}^{n} k\,q\,q_i \frac{(r-r_i)}{|r-r_i|^3}}_{\text{electrostatic force}} + \underbrace{\sum_{i=1}^{n} k\,q\,q_i \frac{v \times (v_i \times (r-r_i))}{c^2\,|r-r_i|^3}}_{\text{magnetic force}}$$

$\Downarrow$ using $a \times (b \times c) = (a^\top c)b - (a^\top b)c$

$$F = \sum_{i=1}^{n} k\,q\,q_i \frac{(r-r_i)}{|r-r_i|^3} + \sum_{i=1}^{n} k\,q\,q_i \frac{(v^\top (r-r_i))\,v_i - (v^\top v_i)\,(r-r_i)}{c^2\,|r-r_i|^3}$$

$$= \sum_{i=1}^{n} k\,q\,q_i \left(1 - \frac{v^\top v_i}{c^2}\right) \frac{(r-r_i)}{|r-r_i|^3} + \sum_{i=1}^{n} k\,q\,q_i \frac{(v^\top (r-r_i))\,v_i}{c^2\,|r-r_i|^3},$$

* Not a field formulation anymore

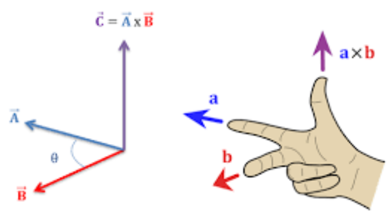
SO(d)-Equivariant vector functions:

$h: (\mathbb{R}^d)^n \to \mathbb{R}^d$ is SO(d)-equivariant if and only if

$$h(v_1 \dots v_n) = \sum_{i=1}^{n} \underbrace{f_i(v_1 \dots v_n)}_{O(d)\text{-invariant scalar}} \cdot v_i + \sum_{S \in \binom{[n]}{d-1}} \overbrace{f_S(v_1, \dots v_n)}^{SO(d)\text{ invariant scalar}} \, v_S$$

v_S: generalized cross product

Example



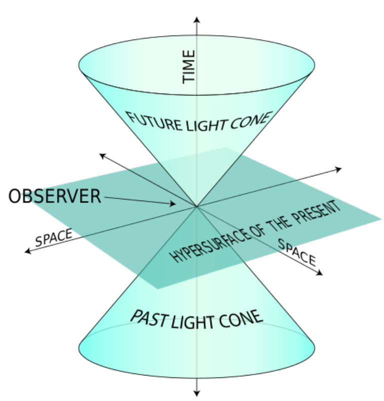
$h: (\mathbb{R}^3)^2 \to \mathbb{R}^3$

$h(v_1, v_2) = v_1 \times v_2 \notin \text{span}\{v_1, v_2\}$

Lorentz-Equivariant vector functions:

$h: (\mathbb{R}^d)^n \to \mathbb{R}^d$ continuous is Lorentz-equivariant if and only if

$$h(v_1 \dots v_n) = \sum_{i=1}^{n} \underbrace{f_i(v_1 \dots v_n)}_{\text{Lorentz-invariant scalar function}} \cdot v_i$$



Translations and permutations

Euclidean group: includes translation symmetry
(also Poincaré)

$h(v_1, \dots, v_n) = \tilde{h}(v_1 - v, \dots, v_n - v)$ O(d)-invariant

where v is the center of mass $\frac{1}{n}\sum v_i$ or any weighted mean position

Permutation invariance: If h is O(d)-equivariant (or Lorentz) and

$h(v_1 \dots v_n) = h(v_{\sigma(1)} \dots v_{\sigma(n)})$ $\sigma \in S_n$ (permutation)

$$h(v_1 \dots v_n) = \sum_{i=1}^{n} f(v_i, v_{[-i]}) \cdot v_i$$

perm inv wrt n-1 last inputs

O(d)-invariant (or Lorentz)

$v_{[-i]} = \{v_1 \dots v_{i-1}, v_{i+1}, \dots v_n\}$

- Easy to implement with message passing graph neural networks

Parameterization for general groups?

Gripaios
Haddadin

- G reductive group over $\mathbb{C}$ (or $\mathbb{R}$)
- The algebra of invariant polynomials P is a graded Cohen-Macaulay algebra

ie: $\exists$ $P = \mathbb{C}[f_1, \ldots f_n]$ where $f_1 \ldots f_n$ - homogeneous
- algebraically independent
- elements of A

where A finitely generated free module over P

ie: $x = p_1(f_1 \ldots f_n) g_1 + \ldots + p_m(f_1 \ldots f_n) g_m$

$f_1 \ldots f_n$ "primary invariants" $g_1 \ldots g_m$ "secondary invariants"
basis of A as a P-module

Example: $O(d)$ $d > n$ $f_1 \ldots f_m$ = scalar products
$g_1 \ldots g_m = 1$

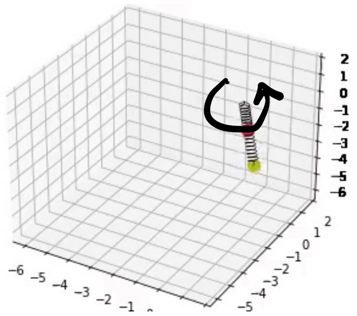
Ongoing work with B. Blum-Smith parameterization for more general groups.

Equivariance vs invariance:

Equivariant functions $V \to W$ $\equiv$ invariant elements in Maps(V, W)

Linear functions $V \to W$ = Hom$(V, W) \cong V^* \otimes W$

Toy example : double pendulum with springs



Credit : EMLP (Finzi et al '21)

data: $(q_1(t), p_1(t))$, m_1, L_1, K_1

$(q_2(t), p_2(t))$ m_2, L_2, K_2

$$KE = \frac{|p_1|^2}{2m_1} + \frac{|p_2|^2}{2m_2}$$

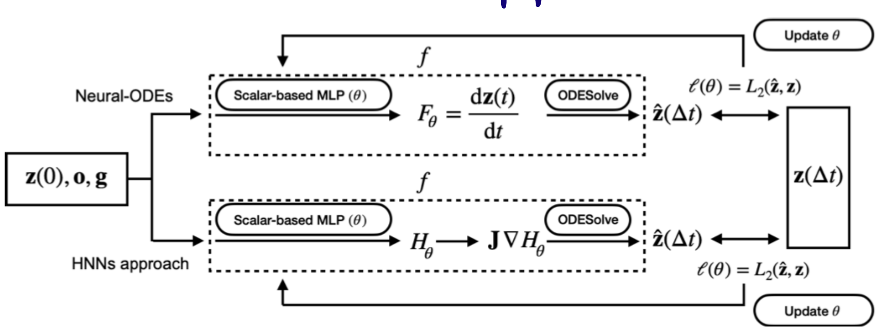
$$PE = \frac{1}{2}K_1(|q_1| - L_1)^2 - m_1 p_1 \cdot g + \frac{1}{2}K_2(|q_1 - q_2| - L_2)^2 - m_2 p_2 \cdot g$$

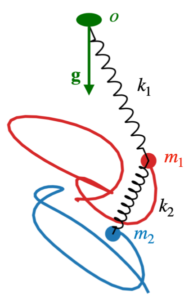
$H = KE + PE$ conserved quantity $\leftrightarrow$ time translation symmetry

$F: (\mathbb{R}^3)^5 \times \mathbb{R} \to (\mathbb{R}^3)^4$ O(3)-equivariant

$(q_1(0), p_1(0), q_2(0), p_2(0), g, \Delta t) \mapsto (q_1(\Delta t), p_1(\Delta t), q_2(\Delta t), p_2(\Delta t))$

Computational approaches:





- **Neural ODEs** $z(t) = (q_1(t), q_2(t), p_1(t), p_2(t))$

$$\frac{dz}{dt} = F(z, q_0, g)$$

Learned E(d) equivariant function $(\mathbb{R}^3)^6 \to (\mathbb{R}^3)^4$

$$\hat{z}(t_j) = \text{ODE solve}(\hat{z}(t_{j-1}), t_{j-1}, t_j, F) \quad \hat{z}(0) = 0$$

input ⟶ scalars ⟶ g_1 ⋮ (MLP) ⟶ ⊕ ⟶ output

- **Hamiltonian neural network (HNNs)**

$$H(q_1, q_2, p_1, p_2, q_0, g) = h(\text{scalars})$$

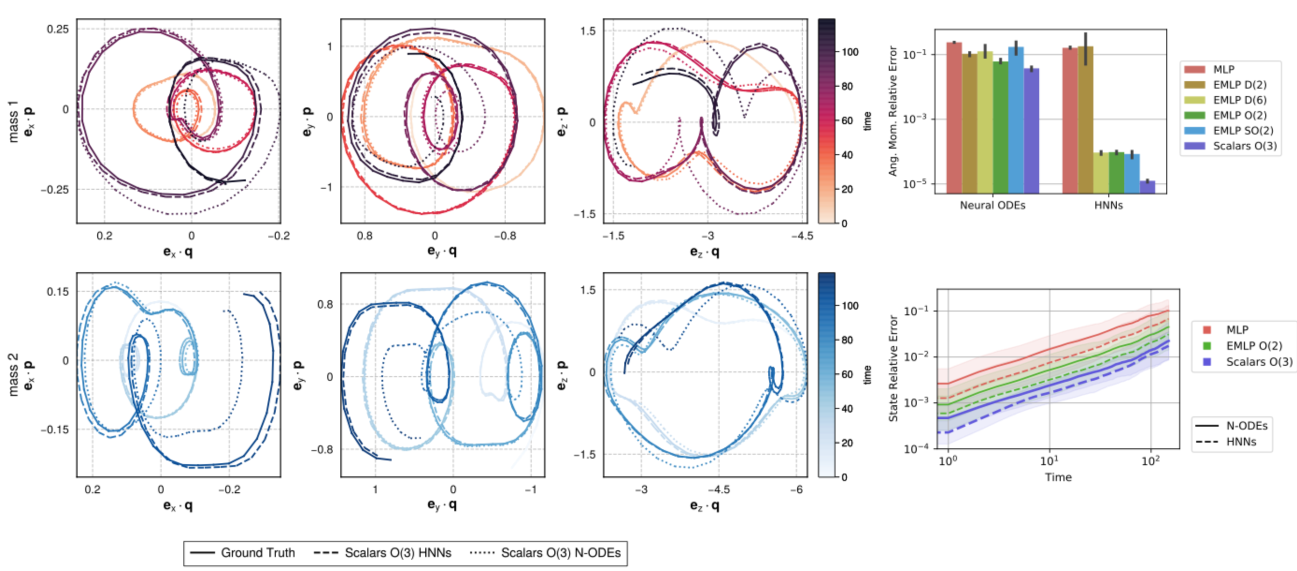
← Learned scalar invariant function

$$\frac{dp_1}{dt} = -\frac{dH}{dq_1} \qquad \frac{dq_1}{dt} = \frac{dH}{dp_1}$$

(symplectic integrator)

Results:

	Scalars O(3)	EMLP				MLP
		O(2)	SO(2)	D_2	D_6	
N-ODEs:	**.009 ± .001**	.020 ± .002	.051 ± .036	.023 ± .002	.036 ± .025	.048 ± .000
HNNs:	**.005 ± .002**	.012 ± .002	.016 ± .003	.111 ± .167	.013 ± .002	.028 ± .001



Units-equivariance

Non-compact groups (Villar et al '22)

Double pendulum:

$$PE = \frac{1}{2} k_1 (|q_1| - L_1)^2 - m_1 p_1 \cdot g + \frac{1}{2} k_2 (|q_1 - q_2| - L_2)^2 - m_2 p_2 \cdot g$$

$$KE = \frac{|p_1|^2}{2 m_1} + \frac{|p_2|^2}{2 m_2}$$

Energy has units: $kg\ m^2\ s^{-2}$

Predictions should be equivariant with respect to rescalings

Approach:

inputs: $\mathbb{R}^d$ —Φ→ dimensionless features $\mathbb{R}^s$ —NNET→ dimensionless labels —Fixed scaling→ dimensional outputs

each feature has units

↓ Based on classical dimensional analysis (Buckingham PI theorem)

Units-typed space (x, u), $x \in \mathbb{R}$, $u \in \mathbb{Z}^k$ eg (kg, m, s) exponents

Energy $kg\ m^2\ s^{-2}$: $[1, 2, -2]$

$$\alpha \cdot (x, u) = (\alpha x, u), \quad \alpha \in \mathbb{R}$$

$$(x, u) + (x', u') = \begin{cases} (x + x', u) & \text{if } u = u' \\ \nexists & \text{otherwise} \end{cases}$$

$$(x, u)(x', u') = (x x', u + u')$$

$$(x, u)^{\gamma} = (x^{\gamma}, \gamma \cdot u), \quad \gamma \in \mathbb{Z}$$

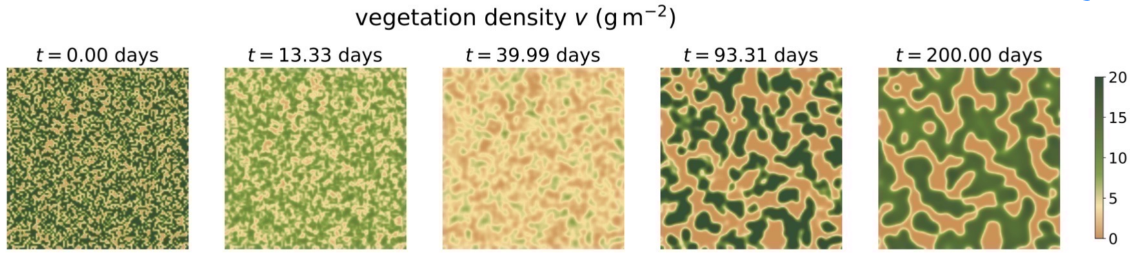
Dimensionless features $z_j = \Phi_j(\mathbf{x}) = \prod_{i=1}^{d} x_i^{\alpha_{ji}}$ where $\sum_{i=1}^{d} \alpha_{ji} u_i = 0$ ($\in \mathbb{Z}^k$)

$\mathbf{x} = (x_i, u_i)_{i = 1 \ldots d}$

dimensionles features = # input variables − # independent units

example: vegetation dynamics

in collaboration with Bianca Dumitrascu (computational biology @ cambridge.UK)



Rietkerk model

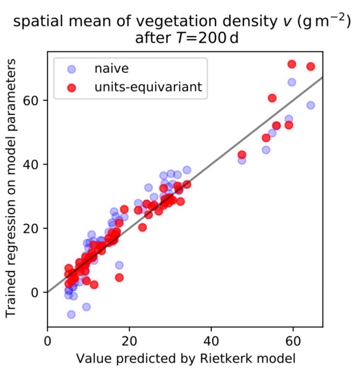
$$\frac{\mathrm{d}u}{\mathrm{d}t} = R - \alpha\,\frac{v + k_2\,W_0}{v + k_2}\,u + D_u\,\nabla^2 u$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \alpha\,\frac{v + k_2\,W_0}{v + k_2}\,u - g_m\,\frac{v\,w}{k_1 + w} - \delta_w\,w + D_w\,\nabla^2 w$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = c\,g_m\,\frac{v\,w}{k_1 + w} - \delta_v\,v + D_v\,\nabla^2 v,$$

	description	default	units
R	rainfall	0.375	$\ell\,\mathrm{d}^{-1}\,\mathrm{m}^{-2}$
α	infiltration rate	0.2	d^{-1}
k_2	saturation const.	5	$\mathrm{g\,m}^{-2}$
W_0	water infiltration const.	0.1	—
D_u	surface water diffusion	100	$\mathrm{d}^{-1}\,\mathrm{m}^2$
g_m	water uptake	0.05	$\ell\,\mathrm{g}^{-1}\,\mathrm{d}^{-1}$
k_1	water uptake constant	5	$\ell\,\mathrm{m}^{-2}$
δ_w	soil water loss	0.2	d^{-1}
D_w	soil water diffusion	0.1	$\mathrm{d}^{-1}\,\mathrm{m}^2$
c	water to biomass	20	$\ell^{-1}\,\mathrm{g}$
δ_v	vegetation loss	0.25	d^{-1}
D_v	vegetation diffusion	0.1	$\mathrm{d}^{-1}\,\mathrm{m}^2$
T	total integration time	200	d
δt	integration time step	0.005	d
L	integration patch length	200	m
δl	spatial step size	2	m

Dimensionless features
$c\,\alpha^{-1}\,g_m$
$R^{-1}\alpha\,k_1$
$R^{-1}c^{-1}\alpha\,k_2$
$\alpha^{-1}\delta_w$
$\alpha^{-1}\delta_v$
W_0
$\alpha^{-1}D_v\,L^{-2}$
$\alpha^{-1}D_u\,L^{-2}$
$\alpha\,T$
$\alpha\,\delta t$
$\alpha^{-1}D_w\,L^{-2}$
$L^{-1}\delta l$



Problem:

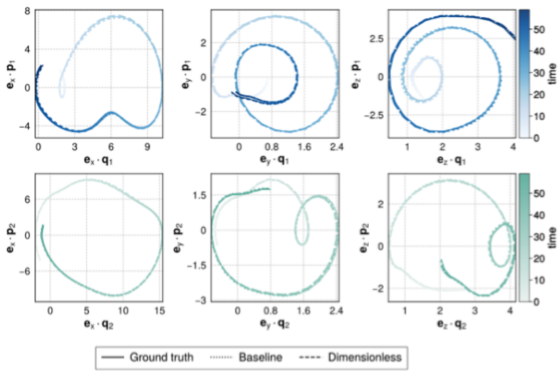
- Learn the diff equations from data using symbolic regression and units equivariant machine learning.

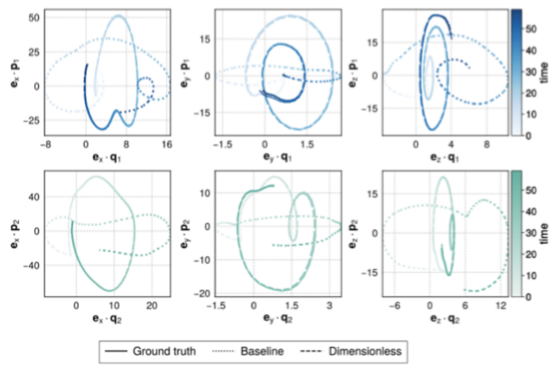
Out-of-distribution generalization

$m_1, m_2 \sim U[1,2]$ $m_1, m_2 \sim U[1,5]$

m_1/m_2 m_1/m_2

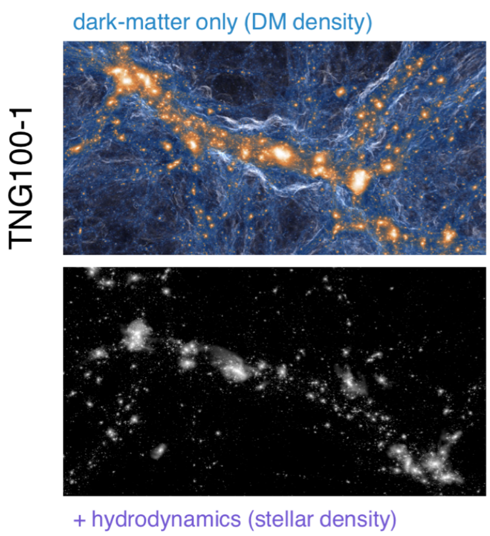
Scalar-based MLPs	Experiment 1	Experiment 2	Experiment 3
Baseline	.0055 ± .0030	.3669 ± .0050	.1885 ± .0031
Dimensionless	.0061 ± .0024	.0089 ± .0034	.0435 ± .0047

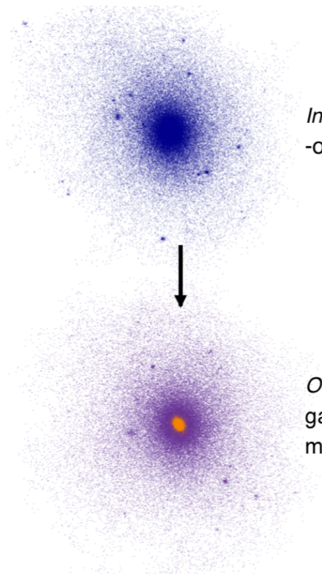




Application: Predicting galaxy properties from dark-matter only simulations

(Kate Storey-Fisher, David Hogg ...)



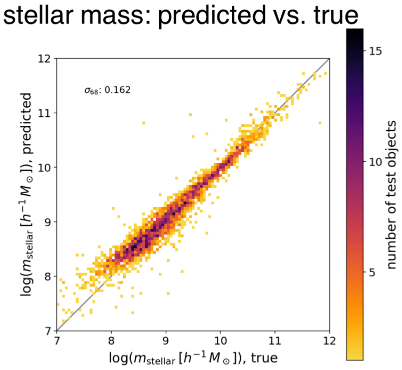


Input: DM halo in DM -only simulation

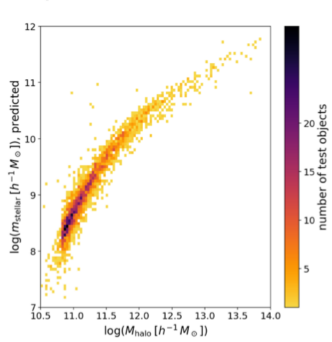
Output: properties of central galaxy hosted by that halo in matched hydro sim

Idea: use dimensionless scalars as features

stellar mass: predicted vs. true



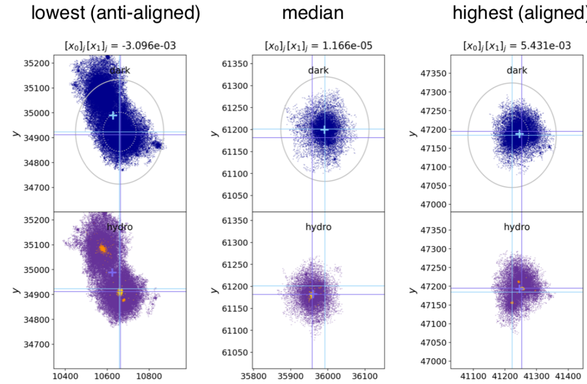
predicted SHMR



mass only: = 0.252
global halo properties: = 0.189
scalars approach: = 0.162 (35% / 15% improvement)

- ~12 features contain ~90% of the information for m as all 568 features
- one of these most important features: , the alignment of the center of mass of the inner and outer halo

lowest (anti-aligned) — median — highest (aligned)



How much do we gain by imposing symmetries? Elesedy Zaidi '21

$G \curvearrowright \mathbb{R}^d$ compact group, $X \sim \mu$ supported in $\mathbb{R}^d$, μ G-invariant

Training data $(x_i, y_i = f^*(x_i) (+ \eta_i))$

(f^* ↑ invariant target; η_i ↑ noise)

$$\text{Risk}(f) = \mathbb{E}_{x \sim \mu} \| f(x) - y \|^2$$

$$\Delta(f, \bar{f}) = \text{Risk}(f) - \text{Risk}(\bar{f}) = \| f^{\perp} \|_{\mu}^2$$

(Δ ↑ generalization gap; $\bar{f}$ ↖ proj of f onto space of invariant functions)

Key property $\bar{f}(x) = \int_{g \in G} f(g \cdot x)\, dg$

$= \arg\min_{h \text{ invariant}} \| f - h \|_{\mu}^2$

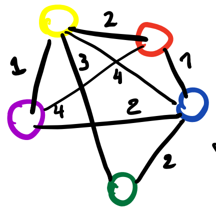
Not true for non-compact groups ↑
what is the "right" notion of projection?

Note that equivariant ML doesn't perform any proj

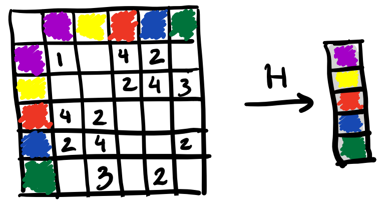
Open problem: model to define "baseline" and quantify "gains"

Symmetries and graph neural networks

G graph

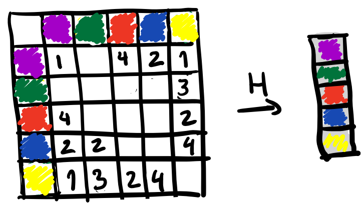


→ A =



Node embedding:
$H(\Pi A \Pi^T) = \Pi \cdot H(A) = \Pi \cdot P$ **equivariant**

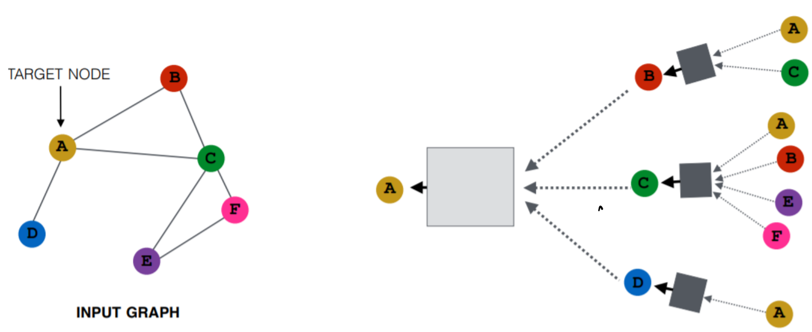
$\Pi . A \Pi^T =$



Q: How to efficiently parameterize the space of invariant and equivariant functions wrt permutation actions?

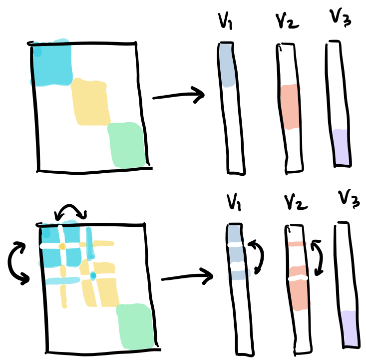
Graph classification / regression
$F(\Pi A \Pi^T) = F(A)$ **invariant**

Message passing: (aka weight sharing)



Credit: Leskovec

Spectral methods are permutation equivariant



This is why spectral clustering works

$f: \mathbb{R}^{n\times n} \to \mathbb{R}^n$

$f(\Pi A \Pi^T) = \Pi f(A)$

Π permutation matrix ($\Pi \in S_n$)

Spectral functions are equivariant

G (graph) → A adjacency

$A = U^T S U$ $\quad F(A) = U^T f(S) U$ Bruna et al 2014

Graph convolutional networks

$$H^{(\ell+1)} = f(H^{(\ell)}, A) = \sigma\left(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} H^{(\ell)} W^{(\ell)}\right)$$

$H^{(\ell+1)}$: $n \times d_{\ell+1}$; $\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2}$: Laplacian $(n\times n)$; $H^{(\ell)}$: Features $(n \times d_\ell)$; $W^{(\ell)}$: weights $d_\ell \times d_{\ell+1}$

$\hat{A} = A + I$

Spectral GNNs:

Given a graph G with adjacency A $(n\times n)$

Let $\mathcal{M} = \{I, D, A, A^2, A^3 \ldots\}$

Learn a "regularized spectral method" on $\Delta = \sum_{M\in\mathcal{M}} \alpha_M M$

Unroll this to a GNN via power iteration $(v^{t+1} = \Delta v^t)$

$$v^{t+1} = \rho\left(\sum_{M\in\mathcal{M}} M v^t \alpha_M^t\right) \qquad \alpha_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$$

$t = 1 \ldots T$ ← #LAYER

- Community detection: Chen, Li, Bruna '17
- Quadratic assignment: Nowak, V., Bandeira, Bruna '17
- Max-cut: Yao, Bandeira, V. '19

characterization of expressivity for GNNs

Z. Chen, V., L. Chen, J. Bruna NeurIPS 2019

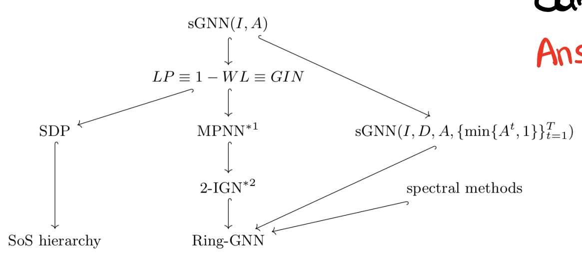
Background: MPNNs cannot distinguish

(Xu et al '19)
(Morris et al '19)

$\left\{ \min_{X} \|AX - XB\|_1 \text{ st } X\mathbf{1}=\mathbf{1},\ X^T\mathbf{1}=\mathbf{1} \right\}$

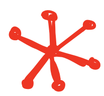
$f(A) = f(B)$

characterization of expressive power of GNNs based on ability to distinguish non-isomorphic graphs



Can GNNs count substructures?

Answer: most architectures can only count star shape substructures



Solutions:

- Signal processing approach: instead of seeing GNNs as embeddings see them as functions:
  (see review paper by Gama et al 21)

  $f : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d'}$ ($\mathbb{R}^{n\times n}$: Adj; $\mathbb{R}^{n\times d}$: node features)

  - See GNNs as low-pass filters on graphs
  - Transferability & stability results

Summary

GOAL:

Enforcing exact symmetries in machine learning models
- Better sample complexity
- Smaller generalization error

GNNs (permutation equivariance)
- Characterization of expressive power of GNNs via graph isomorphism

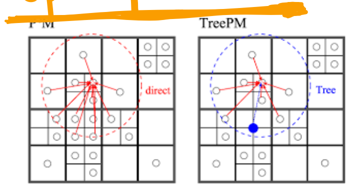
Symmetries in classical physics

- simple characterization of all equivariant functions wrt physically relevant group actions (based on Einstein summation notation & classical invariant theory)

universal approximation ↔ all equivariant

- Extension to units-equivariance

Open problems



Source: Ishiyama et al 21'

- Design a subset of permutation-invariant scalars that are universally expressive
- Explore connections with matrix completion
- Incorporate multi-scale information (FMM, k-d tree)
- Formalize out-of distribution generalization
- Generalization bounds for non-compact groups
- Extension to general groups

Thank you!

Thank you!

- Chen, Villar, Chen, Bruna
  NeurIPS 2019
- Chen, Chen, Villar, Bruna
  NeurIPS 2020
- Villar, Hogg, Storey-Fischer, Yao, Blum-Smith
  NeurIPS 2021
- Yao, Storey-Fisher, Hogg, Villar
  NeurIPS workshop ML for physics 2021
- Villar, Yao, Hogg, Blum-Smith, Dumitrascu
  arXiv: 2204.00887