Equivariant Neural Networks and Differential Invariants Theory for Solving Partial Differential Equations

Pierre-Yves Lagrave[‡], Thales Research and Technology
Eliot Tron[‡], Ecole Normale Supérieure de Lyon
‡: These authors contributed equally to this work.

Introduction

What is the goal of this paper?

- Using Equivariant Neural Networks (ENN) for solving Partial Differential Equations.
- Exploiting the underlying symmetry groups to strengthen the approximation.
- Generalizing ENN to tackle vector valued function with any given symmetry group.
- Building symmetry preserving Finite Difference methods.
- Illustrating our method on the 2D heat equation.

What is the goal of this paper?

- Using Equivariant Neural Networks (ENN) for solving Partial Differential Equations.
- Exploiting the underlying symmetry groups to strengthen the approximation.
- Generalizing ENN to tackle vector valued function with any given symmetry group.
- Building symmetry preserving Finite Difference methods.
- Illustrating our method on the 2D heat equation.

What is the goal of this paper?

- Using Equivariant Neural Networks (ENN) for solving Partial Differential Equations.
- Exploiting the underlying symmetry groups to strengthen the approximation.
- Generalizing ENN to tackle vector valued function with any given symmetry group.
- Building symmetry preserving Finite Difference methods.
- Illustrating our method on the 2D heat equation.

What is the goal of this paper?

- Using Equivariant Neural Networks (ENN) for solving Partial Differential Equations.
- Exploiting the underlying symmetry groups to strengthen the approximation.
- Generalizing ENN to tackle vector valued function with any given symmetry group.
- Building symmetry preserving Finite Difference methods.
- Illustrating our method on the 2D heat equation.

What is the goal of this paper?

- Using Equivariant Neural Networks (ENN) for solving Partial Differential Equations.
- Exploiting the underlying symmetry groups to strengthen the approximation.
- Generalizing ENN to tackle vector valued function with any given symmetry group.
- Building symmetry preserving Finite Difference methods.
- Illustrating our method on the 2D heat equation.

State of the Art

Neural Networks applied to PDEs
Multiple approaches [Rai, SS, LJK, LMMK, WKM+], pretty efficient.

State of the Art

Neural Networks applied to PDEs
Multiple approaches [Rai, SS, LJK, LMMK, WKM$^+$], pretty efficient.

Symmetry in Neural Networks
Recent works (since 2016) [CW, CGW, CWKW, FWW, FSIW, WC], highly promising but limited results for our task.

State of the Art

Neural Networks applied to PDEs
Multiple approaches [Rai, SS, LJK, LMMK, WKM+], pretty efficient.

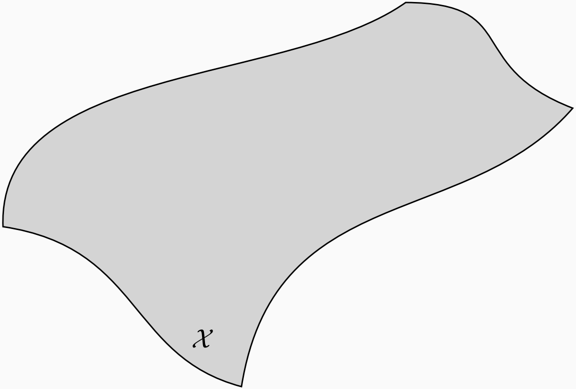
Symmetry in Neural Networks
Recent works (since 2016) [CW, CGW, CWKW, FWW, FSIW, WC], highly promising but limited results for our task.

Only one applied to PDEs:
Rui Wang, Robin Walters, and Rose Yu. *Incorporating Symmetry Into Deep Dynamics Models for Improved Generalization* [WWY]. However, only special cases.

PDEs and Symmetries

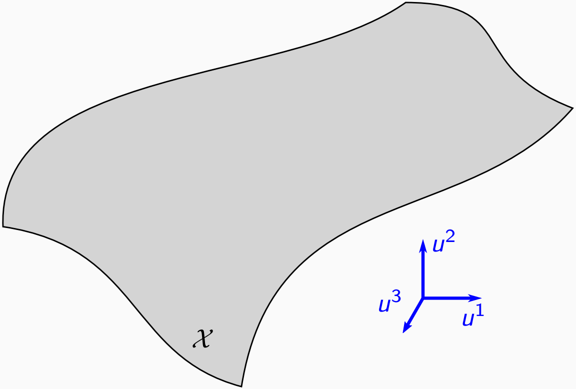
In what space are we working?

- A smooth manifold $\mathcal{X}$ (coordinate space).



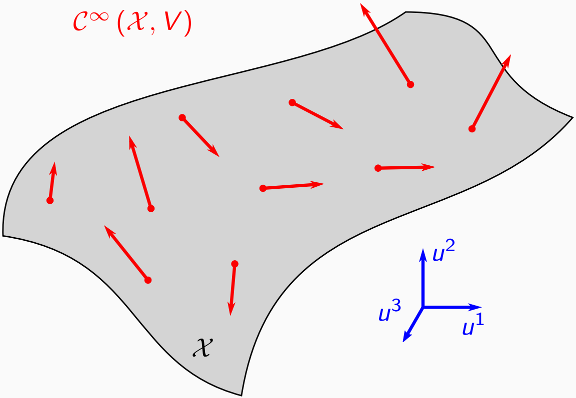
In what space are we working?

- A smooth manifold $\mathcal{X}$ (coordinate space).
- A vector space V (output space).



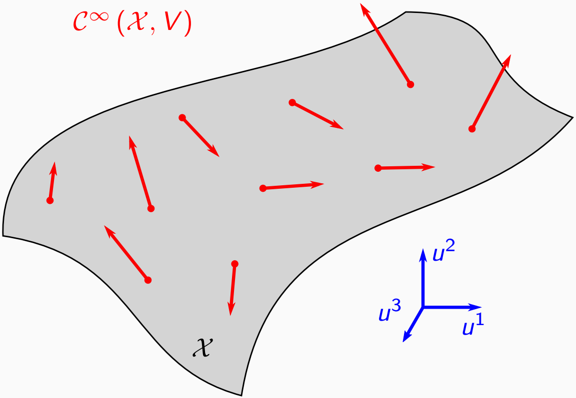
In what space are we working?

- A smooth manifold $\mathcal{X}$ (coordinate space).
- A vector space V (output space).
- Smooth functions $\mathcal{U} = \mathcal{C}^{\infty}(\mathcal{X}, V)$.



In what space are we working?

- A smooth manifold $\mathcal{X}$ (coordinate space).
- A vector space V (output space).
- Smooth functions $\mathcal{U} = \mathcal{C}^\infty(\mathcal{X}, V)$.
- The Jet space $J^{(n)} = \mathcal{X} \times \mathcal{U} \times \ldots \times \mathcal{U}$



What are we trying to solve?

Definition (PDE System)

$$(E): \quad \begin{cases} \Delta\left(t, x, u^{(n)}\right) = 0 & \forall t, x \in \mathbb{R}^+ \times \mathcal{X} \\ u(t,x) = u_b(t,x) & \forall (t,x) \in B \end{cases}$$

What are symmetries?

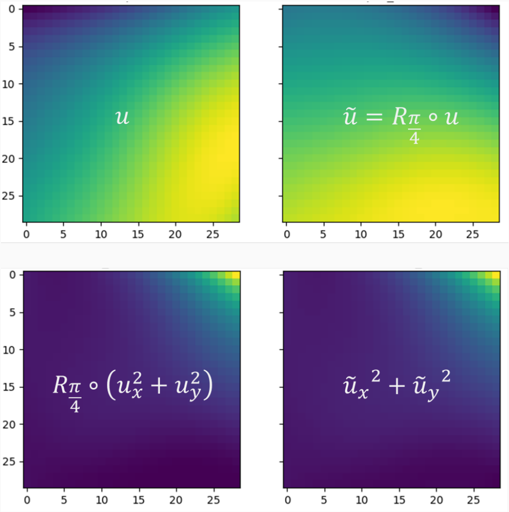
Definition (Symmetry Group)
A Lie group G is the *symmetry group* of a PDEs system when if f is a solution, then its transform f_g by the group action is also a solution.

What are symmetries?

Definition (Symmetry Group)
A Lie group G is the *symmetry group* of a PDEs system when if f is a solution, then its transform f_g by the group action is also a solution.

Definition (Differential Invariant)
The algebraic invariants I_G of the prolonged group action $\mathrm{pr}^{(n)}\, G$ are called the *differential invariants* of order n of the group G. A complete set of independent differential invariants of order n will be denoted by $\partial\phi^G_{u,n} = \left\{\partial\phi^{G,1}_{u,n}, \ldots, \partial\phi^{G,k}_{u,n}\right\}$.

What are symmetries?



Equivariant Neural Networks

What is the operation behind G-CNN?

Definition (Group Convolution)

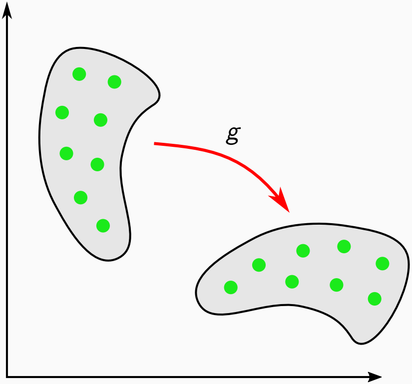
Let G be a compact Group and $V_1,\ V_2$ two vector spaces. Let $K : G \to \mathcal{L}(V_1, V_2)$ be a kernel, $f : G \to V_1$ be a feature function, and μ the Haar measure on G. We define the *group convolution* for any $s \in G$ by

$$(K * f)(s) = \int_G K\left(r^{-1}s\right) f(r) d\mu(r).$$

Why is it interesting?

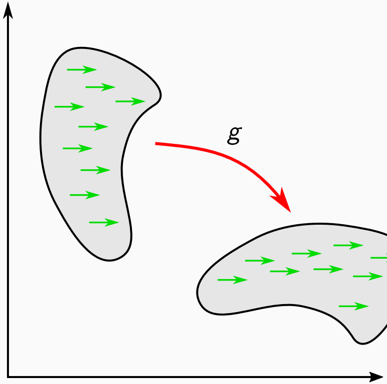
Proposition
If the actions of G on V_1^G and V_2^G has regular representations, then the group convolution defined before is G-equivariant.

Why is it not enough?



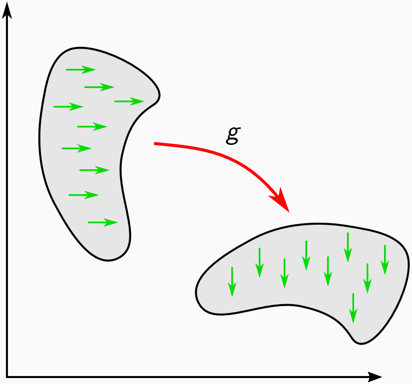
Regular representation $\rho(g) = 1$ on scalars.

Why is it not enough?



Regular representation $\rho(g) = 1$ on vectors.

Why is it not enough?



Non-regular representation on vectors.

A New Convolution

Definition (Representative Group Convolution)

Let G be a compact Group and $V_1,\ V_2$ two vector spaces. Let $K : G \to \mathcal{L}(V_1, V_2)$ be a kernel, $f : G \to V_1$ be a feature function, and μ the Haar measure on G. If $\rho_1 : G \to \mathcal{L}(V_1)$ and $\rho_2 : G \to \mathcal{L}(V_2)$ are the linear representations of the action of G on V_1 and V_2 respectively, we define the *representative group convolution* for any $s \in G$ by

$$(K \circledast f)(s) = \int_G \rho_2(r) K\left(r^{-1}s\right) \rho_1\left(r^{-1}\right) f(r) d\mu(r) \tag{1}$$

Finally enough

Theorem

If G acts on V^G by

$$\rho(g)f\left(g^{-1}r\right) \qquad \forall g, r \in G \text{ and } f : G \to V,$$

then the representative group convolution is G-equivariant.

Equivariant Neural Network

G-CNN

$$
\begin{aligned}
\text{input layer:} \quad & \mathcal{N}^0 = f \in V_0^G, \\
\text{convolution layers:} \quad & \mathcal{N}^\ell = K^\ell \circledast \mathcal{N}^{\ell-1} \in V_\ell^G, \\
& \text{with } \rho_{\ell-1},\ \rho_\ell, \text{ for } 1 \leq \ell \leq L.
\end{aligned}
$$

Equivariant Neural Network

G-CNN

$$
\begin{aligned}
\text{input layer:} \quad & \mathcal{N}^0 = f \in V_0^G, \\
\text{convolution layers:} \quad & \mathcal{N}^\ell = K^\ell \circledast \mathcal{N}^{\ell-1} \in V_\ell^G, \\
& \text{with } \rho_{\ell-1},\ \rho_\ell, \text{ for } 1 \le \ell \le L.
\end{aligned}
$$

Remark
One can use non-equivariant pointwise function between layers as long as $\rho_\ell = I_d$.

Solving of PDEs with ENN

What Neural Network to use for PDEs?

Loss

$$\mathcal{L}(\theta, \mathcal{T}) = w_f \mathcal{L}_f(\theta, \mathcal{T}_f) + w_b \mathcal{L}_b(\theta, \mathcal{T}_b) \tag{2}$$

with

$$\mathcal{L}_f(\theta, \mathcal{T}_f) = \frac{1}{|\mathcal{T}_f|} \sum_{x \in \mathcal{T}_{\boldsymbol{f}}} \left\| \Delta\left(t, x, \hat{u}^{(n)}\right) \right\|_2^2 \tag{3}$$

$$\mathcal{L}_b(\theta, \mathcal{T}_b) = \frac{1}{|\mathcal{T}_b|} \sum_{x \in \mathcal{T}_{\boldsymbol{b}}} \left\| \mathcal{B}(\hat{u}_\theta, x) \right\|_2^2 \tag{4}$$

How can we simplify the problem with differential invariants?

Problem

$$\sum_{i=1}^{k_{\boldsymbol{t}}} a_i \partial_{\boldsymbol{t}}^i u = \Delta\left(t, x, u_x^{(n)}\right) \tag{5}$$

Theorem

$$\Delta\left(t, x, u_x^{(n)}\right) = F\left(\partial\phi_{u,n}^{G,1}, \ldots, \partial\phi_{u,n}^{G,k}\right) \tag{6}$$

How can we simplify the problem with differential invariants?

Problem

$$\sum_{i=1}^{k_t} a_i \partial_t^i u = \Delta\left(t, x, u_x^{(n)}\right) \tag{5}$$

Theorem

$$\Delta\left(t, x, u_x^{(n)}\right) = F\left(\partial \phi_{u,n}^{G,1}, \ldots, \partial \phi_{u,n}^{G,k}\right) \tag{6}$$

known unknown

How do we use the ENNs now?

Approximation

$$\mathcal{N}_G\left(\left(f^{(i)(\ell)}\right)_{\ell=1}^{n_x}\right)^j \approx \partial\phi_f^{G,(i,j)} = \partial\phi_f^G(t^{(i)}, x^{(j)}) \tag{7}$$

How do we use the ENNs now?

Approximation

$$\mathcal{N}_G\left(\left(f^{(i)(\ell)}\right)_{\ell=1}^{n_x}\right)^j \approx \partial\phi_f^{G,(i,j)} = \partial\phi_f^G(t^{(i)}, x^{(j)}) \tag{7}$$

New problem

$$\sum_{i=1}^{k_t} a_i \partial_t^i u \approx F\left(\mathcal{N}_G^1\left(\left(u^{(i,\ell)}\right)_{\ell=1}^{n_x}\right)^j, ..., \mathcal{N}_G^k\left(\left(u^{(i,\ell)}\right)_{\ell=1}^{n_x}\right)^j\right) \tag{8}$$

Numerical Experiments

How close can we get with SE(2) differential invariants?

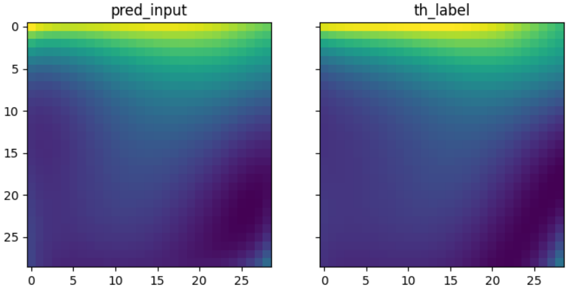


Figure 1: The SE(2) differential invariant $u_x^2 + u_y^2$ computed for the function u depicted in Frame 5 with an SE(2)-CNN (left) and its theoretical value (right)

How close can we get with SE(2) differential invariants?

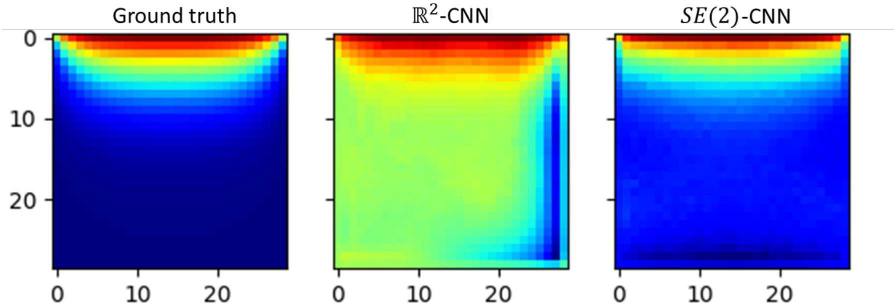


Figure 1: Comparison of the theoretical heat profile of the 2D heat equation with a top 100°C boundary condition with those obtained through simulation with two symmetry preserving FD schemes by leveraging on $\mathbb{R}^2$ (middle) and SE(2) (right) equivariant neural networks.

Conclusion and Further Work

Contributions

- Use of G-CNN to generalize the PINN architecture to encode generic symmetries ;
- Use of ENN to approximate differential invariants of a given symmetry group ;
- Build of symmetry preserving Finite Difference methods.

To go further

- Perfom proper benchmarking between the two approaches and other conventional numerical schemes for PDEs integration ;
- Test on more complex PDEs with richer symmetry groups (e.g. Maxwell).

Contributions

- Use of G-CNN to generalize the PINN architecture to encode generic symmetries ;
- Use of ENN to approximate differential invariants of a given symmetry group ;
- Build of symmetry preserving Finite Difference methods.

To go further

- Perfom proper benchmarking between the two approaches and other conventional numerical schemes for PDEs integration ;
- Test on more complex PDEs with richer symmetry groups (e.g. Maxwell).

Contributions

- Use of G-CNN to generalize the PINN architecture to encode generic symmetries ;
- Use of ENN to approximate differential invariants of a given symmetry group ;
- Build of symmetry preserving Finite Difference methods.

To go further

- Perfom proper benchmarking between the two approaches and other conventional numerical schemes for PDEs integration ;
- Test on more complex PDEs with richer symmetry groups (e.g. Maxwell).

Contributions

- Use of G-CNN to generalize the PINN architecture to encode generic symmetries ;
- Use of ENN to approximate differential invariants of a given symmetry group ;
- Build of symmetry preserving Finite Difference methods.

To go further

- Perfom proper benchmarking between the two approaches and other conventional numerical schemes for PDEs integration ;
- Test on more complex PDEs with richer symmetry groups (e.g. Maxwell).

Contributions

- Use of G-CNN to generalize the PINN architecture to encode generic symmetries ;
- Use of ENN to approximate differential invariants of a given symmetry group ;
- Build of symmetry preserving Finite Difference methods.

To go further

- Perfom proper benchmarking between the two approaches and other conventional numerical schemes for PDEs integration ;
- Test on more complex PDEs with richer symmetry groups (e.g. Maxwell).

Questions?

Taco Cohen, Mario Geiger, and Maurice Weiler.
A General Theory of Equivariant CNNs on Homogeneous Spaces.

Taco S. Cohen and Max Welling.
Steerable CNNs.

Taco S Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling.
Gauge Equivariant Convolutional Networks and the Icosahedral CNN.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson.
Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data.

Marc Finzi, Max Welling, and Andrew Gordon Wilson.
A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups.

Lu Lu, Pengzhan Jin, and George Em Karniadakis.
DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators.

Lu Lu, Xuhui Meng, Zhiping Mao, and George E. Karniadakis.
DeepXDE: A deep learning library for solving differential equations.

Maziar Raissi.
Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations.

Justin Sirignano and Konstantinos Spiliopoulos.
DGM: A deep learning algorithm for solving partial differential equations.
375:1339–1364.

Maurice Weiler and Gabriele Cesa.
General $E(2)$-Equivariant Steerable CNNs.

- Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu.
  Towards Physics-informed Deep Learning for Turbulent Flow Prediction.
- Rui Wang, Robin Walters, and Rose Yu.
  Incorporating Symmetry Into Deep Dynamics Models for Improved Generalization.