Incorporating **Symmetry** for Learning Spatiotemporal Dynamics

Rose Yu

Assistant Professor
University of California, San Diego
Learning Spatiotemporal Dynamics

- mechanical engineering
- biomedical engineering
- transportation
- climate science
- public health
- sports analytics
- quantum chemistry
Success of Symmetry

Ravanbakhsh et al. (2017); Kondor & Trivedi (2018); Cohen & Welling (2016b); Thomas et al. (2018); Maron et al. (2020); Walters et al. (2021).....

How about Spatiotemporal Dynamics?
Incorporating **Symmetry** for **Generalization**

Incorporating Symmetry into Deep Dynamics Models for Improved Generalization
Rui Wang*, Robin Walters*, and Rose Yu
Generalization Challenge

• Generalization: fundamental challenge in dynamics forecasting
  • Performance degrades with test distributional shift
  • Punchline: distributions change, laws of physics do not!

Bridging Physics-based and Data-driven modeling for Learning Dynamical Systems
Rui Wang, Danielle Maddix, Christos Faloutsos, Yuyang Wang, Rose Yu,
International Conference in Learning for Dynamics and Control (L4DC), 2021
Noether’s theorem: For every symmetry, there is a corresponding conservation law

Invariance, Equivariance:

- G-invariant: \( f(g(x)) = f(x) \)
- G-equivariant: \( f(gx) = gf(x) \)
Symmetry in Dynamical Systems

- A system of differential operators
  \[ D = \{ P_1, \ldots, P_r \} \]

- If \( \phi \) is a solution of \( D \), then for all \( g \in G, g(\phi) \) is also a solution

- **2D Navier-Stokes Equations**
  \[
  \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla) \mathbf{w} = -\frac{1}{\rho_0} \nabla p + \nu \nabla^2 \mathbf{w} + f
  \]
  \[
  \nabla \cdot \mathbf{w} = 0
  \]
  \[
  \frac{\partial T}{\partial t} + (\mathbf{w} \cdot \nabla) T = \kappa \nabla^2 T
  \]

**Symmetries** | **NS Equ.**
---|---
Space translation | \( \mathbf{w}(\mathbf{x} - \mathbf{v}, t) \)
Time translation | \( \mathbf{w}(\mathbf{x}, t - \tau) \)
Uniform Motion | \( \mathbf{w}(\mathbf{x}, t) + \mathbf{c} \)
Reflect/rotation | \( R\mathbf{w}(R^{-1}\mathbf{x}, t) \)
Scaling | \( \lambda \mathbf{w}(\lambda \mathbf{x}, \lambda^2 t) \)

Scaling Law
Theorem (Weiler & Cesa 2019): a convolutional layer is $G$-equivariant if and only if the kernel satisfies

$$K(gv) = \rho_{out}^{-1}(g)K(v)\rho_{in}(g)$$

for all $g \in G$, with action maps $\rho_{in}$ and $\rho_{out}$. 

Weight Symmetry
Symmetry: Scaling

- Standard convolution shares weights across the input by translating a kernel across the input.
- For scale-equivariant convolution, we must translate and scale a kernel across the input.
Symmetry: Scaling

- Scale equivariant

\[ v(p) = \sum_{\lambda \in \mathbb{Z}^+} (T_{\lambda} w)(p + q)(T_{\lambda} K)(q), \]

\[ T_{\lambda} w(x, t) = \lambda w(\lambda x, \lambda^2 t) \]
Ocean Currents Forecast

Physically Consistent Predictions!
Approximately Equivariant Networks

Approximately Equivariant Networks for Imperfectly Symmetric Dynamics
Rui Wang, Robin Walters, and Rose Yu.
International Conference on Machine Learning (ICML) 2022.
Symmetry as Inductive Bias

When symmetry is a good inductive bias

Prediction Error

Model Equivariance Error
**Definition:** Let $f: X \to Y$ be a function and $G$ be a group. Assume that $G$ acts on $X$ and $Y$ via representations $\rho_X$ and $\rho_Y$. We say $f$ is $\epsilon$-approximately $G$-equivariant if for any $g \in G$,

$$\|f(\rho_X(g)(x)) - \rho_Y(g)f(x)\| \leq \epsilon.$$
Equivariant Convolution

• Group Convolution (*G-conv*)

\[ f \ast_G K(g) = \sum_{h \in G} f(h)K(g^{-1}h) \]

• G-conv does not need to precompute an equivariant kernel basis

• But limited to discrete (compact) group, not efficient when the group order is large

• G-Steerable Convolution (*Steer*)

\[ K(hx) = \rho_{out}(h)K(x)\rho_{in}(h^{-1}) \]
Relaxed Equivariance

- Relaxed G-conv (**RGroup**):

\[
\tilde{f}^* \ast_G K(g) = \sum_{h \in G} f(h) \sum_{l=1}^{L} w_l(h) K_l(g^{-1}h)
\]

- Relaxed Steerable (**RSteer**):

\[
\tilde{K}(hx) = \rho_{out}(h) \sum_{l=1}^{L} w_l(h) K_l(x) \rho_{in}(h^{-1})
\]
Smoke Plume

<table>
<thead>
<tr>
<th>t = 1</th>
<th>Target</th>
<th>ConvNet</th>
<th>RPP</th>
<th>E2CNN</th>
<th>Lift</th>
<th>RSteer</th>
</tr>
</thead>
<tbody>
<tr>
<td>t = 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t = 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t = 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Smoke Plume Results

![Graph showing model equivalence error vs. data equivalence error for different models: E2CNN, ConvNet, RSteerR, Lift, RPP. The graph compares the performance of these models against an optimal performance line.](image)
Supersonic Jet Flow

- Real experimental data of 2D turbulent velocity in multi-stream jets from NASA
- Measured using time-solved partial image velocimetry
## Prediction Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Conv</th>
<th>Lift</th>
<th>RGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Translation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Future Domain</td>
<td>0.22±0.06</td>
<td>0.17±0.02</td>
<td>0.15±0.00</td>
</tr>
<tr>
<td>Domain</td>
<td>0.23±0.06</td>
<td>0.18±0.02</td>
<td>0.16±0.01</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2CNN</td>
<td>0.21±0.02</td>
<td>0.18±0.02</td>
<td>0.17±0.01</td>
</tr>
<tr>
<td>Lift</td>
<td>0.21±0.03</td>
<td>0.21±0.04</td>
<td>0.16±0.01</td>
</tr>
<tr>
<td>RSteer</td>
<td>0.15±0.00</td>
<td>0.16±0.06</td>
<td>0.14±0.01</td>
</tr>
<tr>
<td><strong>Scaling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SESN</td>
<td>0.16±0.01</td>
<td>0.16±0.07</td>
<td>0.15±0.00</td>
</tr>
<tr>
<td>Rpp</td>
<td>0.16±0.01</td>
<td>0.16±0.07</td>
<td>0.15±0.00</td>
</tr>
<tr>
<td>RSteer</td>
<td>0.14±0.01</td>
<td>0.14±0.02</td>
<td></td>
</tr>
<tr>
<td><strong>Combination</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSteerTR</td>
<td>0.14±0.01</td>
<td>0.14±0.02</td>
<td></td>
</tr>
<tr>
<td>RSteerTS</td>
<td>0.15±0.01</td>
<td>0.15±0.00</td>
<td></td>
</tr>
</tbody>
</table>

20% better
Symmetry Teleportation for Accelerated Optimization

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu.
Optimization in DL is Hard!

Li, Hao, et al. "Visualizing the loss landscape of neural nets."
Symmetry Can Help

Gradient Descent

Second-order

GD + Teleportation
Symmetry Teleportation

**Algorithm 1: Symmetry Teleportation**

**Input:** Loss function $\mathcal{L}(w)$, learning rate $\eta$, number of epochs $t_{max}$, initialized parameters $w_0$, symmetry group $G$, teleportation schedule $K$.

**Output:** $w_{t_{max}}$.

1. for $t \leftarrow 0$ to $t_{max} - 1$ do
2.     if $t \in K$ then
3.         $g \leftarrow \arg\max_{g \in G} \|\nabla \mathcal{L}|_{g \cdot w_t}\|^2$
4.         $w_t \leftarrow g \cdot w_t$
5.     end if
6.     $w_{t+1} \leftarrow w_t - \eta (\nabla \mathcal{L})|_{w_t}$
7. end for
8. return $w_{t_{max}}$
Finding teleportation destination

\[ g \leftarrow \arg\max_{g \in G} \| (\nabla \mathcal{L}) \|_g \cdot w_t \| ^2 \]

Parametrize and gradient ascent on \( g \).
- Example 1: SO(2), gradient ascent on rotation angle \( \theta \)
- Example 2: GL_d(\mathbb{R}), \( g \approx I + \varepsilon T \) where \( \varepsilon << 1 \) and \( T \in \mathbb{R}^{d \times d} \), gradient ascent on \( T \)
Test function: Booth

\[ \mathcal{L}_b(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \]

Symmetry group: SO(2)

GD

GD + teleportation

Training curves
Test function: Rosenbrock

\[ \mathcal{L}_r(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 \]
Multi-layer neural network regression

\[ \mathcal{L}(W_1, W_2, W_3) = \| Y - W_3 \sigma(W_2 \sigma(W_1 X)) \|_2 \]
Conclusion

• Incorporating symmetry in deep learning for learning spatiotemporal dynamics

  • EquNet: symmetry in differential equations

  • Relaxed-EquNet: approximate symmetry

  • Teleportation: symmetry in learning dynamics

• Probabilistic modeling, symmetry discovery, etc…
On Molecules…

SCAN ME
Acknowledgment

Open Source Code and Data: roseyu.com

@yuqirose