Symmetries and ML

Andreas Ipp

Institute for Theoretical Physics, TU Wien, Austria

SLAC Summer Institute (SSI 2023)

August 18, 2023
Symmetries

Symmetry in physics: a property that remains unchanged under some transformation.

Discrete symmetries
- finite groups, permutation group
  (Graph neural network, Transformer)
- parity / mirror
- time inversion

Continuous symmetries
- Lie groups

Global symmetries
- translation (Convolutional neural network)
- rotation (Group-equivariant CNN, Steerable CNN)
- time translation (Recurrent neural network)

Local symmetries
- gauge symmetries
  (Lattice gauge equivariant CNN)
Noether’s theorem

*Every continuous symmetry of the action corresponds to a conservation law.*

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Conservation law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation in space</td>
<td>Conservation of momentum (CNN)</td>
</tr>
<tr>
<td>Translation in time</td>
<td>Conservation of energy (Hamiltonian NN, Lagrangian NN)</td>
</tr>
<tr>
<td>Rotation in space</td>
<td>Conservation of angular momentum (G-CNN, Steerable CNN)</td>
</tr>
<tr>
<td>Gauge invariance</td>
<td>Conservation of charge (L-CNN, equivariant coupling layers)</td>
</tr>
</tbody>
</table>

Emmy Noether  
(source: Wikipedia)
Outline

Motivation

Translational symmetry

Toy model

Lattice gauge symmetry

Bulusu, Favoni, Al, Müller, Schuh, Phys. Rev. D 104 (2021) 074504

Favoni, Al, Müller, Schuh, Phys. Rev. Lett. 128 (2022) 032003
Motivation
QCD phase diagram

Sun (surface): 6000°C ≈ 0.5 eV

Sun (core): 15 million °C ≈ 1.3 keV

Quark-Gluon Plasma: 1.7×10^{12} °C ≈ 150 MeV
Quark-gluon plasma

- Existed in the early universe
- Produced in heavy ion collisions
Stages of a heavy-ion collision

- **Initial state:** Lorentz-contracted nuclei (color glass condensate)
- **Collision event:**
  - **Glasma** ($\tau \approx 0 - 1$ fm/$c$): quasi-classical fields (classical field equations)
- **QGP** ($\tau \approx 1 - 10$ fm/$c$): quarks and gluons (relativistic viscous hydrodynamics)
- **Hadronization** ($\tau \approx 10$ fm/$c$): confinement transition $\rightarrow$ hadron formation
- **Hadronic gas** ($\tau \approx 10 - 15$ fm/$c$): hadrons (kinetic transport theory)
- **Freeze-out** ($\tau \approx 15$ fm/$c$): interactions stop

$1 \text{ fm}/c \approx 3.3 \cdot 10^{-24} \text{s} \approx 3.3 \text{ ys}$
Colored particle-in-cell method

Generalization of the particle-in-cell method from plasma physics for strong interactions.


Based on real-time lattice gauge theory in a classical regime.

Dispersion-free propagation

Standard Wilson action:

\[ S[U] = \frac{V}{g^2} \sum_x \left( \sum_i \frac{1}{(a^0 a^i)^2} \text{tr} \left( 2 - U_{x,0i} - U_{x,0i}^\dagger \right) - \frac{1}{2} \sum_{i,j} \frac{1}{(a^i a^j)^2} \text{tr} \left( 2 - U_{x,ij} - U_{x,ij}^\dagger \right) \right) \]

Variational integrator: Discretized equations of motion from discretized action


Discretized action for the **semi-implicit scheme**:

\[ S[U] = \frac{V}{g^2} \sum_x \left( \frac{1}{(a^0 a^1)^2} \text{tr} \left( C_{x,01} C_{x,01}^\dagger \right) + \sum_i \frac{1}{(a^0 a^i)^2} \text{tr} \left( C_{x,0i} C_{x,0i}^\dagger \right) \right) \]

\[ - \frac{1}{4} \sum_{i,j} \frac{1}{(a^i a^j)^2} \text{tr} \left( C_{x,ij} M_{x,ij}^\dagger \right) - \frac{1}{4} \sum_{i,j} \frac{1}{(a^i a^j)^2} \text{tr} \left( C_{x,1j} W_{x,1j}^\dagger + \text{h.c.} \right) \]

with \( C_{x,\mu\nu} \equiv U_{x,\mu} U_{x+\mu,\nu} - U_{x,\nu} U_{x+\nu,\mu} \)
Simulations of the collision process
Computational challenges

Simulating small part of nuclei at RHIC energies:

γ-factor: 100  
Lattice: 2048 × 192² cells  
Gauge group: SU(2)  
Color sheets: 1  
Simulation box: (6 fm)³

→ 25 GB simulation data  
→ 192 core hours on Vienna Scientific Cluster (VSC-3)

Simulating realistic off-central full size nuclei at LHC energies:

γ-factor: 2500  
Lattice: (25×20480) × 1920² cells  
Gauge group: SU(3)  
Color sheets: 100  
Simulation box: (60 fm)³

→ 25 PB simulation data  
→ 5 million core years on VSC-3  
(2020: 150 years on VSC-3; but only 130 TB RAM available)  
(2023: 55 years on VSC-5; 355 TB RAM available)
Machine learning in fluid dynamics

Accelerating Eulerian Fluid Simulation With Convolutional Networks
Tompson et al, arxiv:1607.03597

- Compress computation time and memory usage
- Use convolutional autoencoders to compress state size
- Learn dynamics on compressed form
- Can generalize to larger grid sizes

Lat-Net: Compressing Lattice Boltzmann Flow Simulations using Deep Neural Networks
Hennigh, arxiv:1705.09036
Relativistic hydrodynamics

Applications of deep learning to relativistic hydrodynamics
Huang et al., arxiv:1801.03334

- Speed up simulation time from 20 min to few seconds
- Network had to be trained with 10,000 initial and final state pairs
Toy model

Epoch 1000

-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
input x

-4.0 -2.0 0.0 2.0 4.0
output y

ground truth
training data
NN predictions
Learning a simple function

Learn $f(x) = x^3$

Example in PyTorch

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.l1 = nn.Linear(1, 128)
        self.l2 = nn.Linear(128, 32)
        self.l3 = nn.Linear(32, 1)

    def forward(self, x):
        x = torch.sigmoid(self.l1(x))
        x = torch.sigmoid(self.l2(x))
        x = self.l3(x)
        return x

Symmetries and ML  Andreas Ipp  16
Learning a simple function

Learn \( f(x) = x^3 \)

Epoch 1000

-2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0

output y

-4.0 -2.0 0.0 2.0 4.0

input x

-4.0 -3.0 -2.0 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0

output y

-4.0 -2.0 0.0 2.0 4.0

input x

-4.0 -3.0 -2.0 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0

output y

-4.0 -2.0 0.0 2.0 4.0

input x
Learning a simple function

Learn $f(x) = x^3$

$$\Delta_{\text{sym}}(x) = \frac{h(+x) + h(-x)}{2}$$

Symmetry: $f(-x) = -f(x)$

learned approximately, but small deviation remains
Imposing the symmetry

Learn $f(x) = x^3$

Example in PyTorch

```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.l1 = nn.Linear(1, 128, bias=False)
        self.l2 = nn.Linear(128, 32, bias=False)
        self.l3 = nn.Linear(32, 1, bias=False)

    def forward(self, x):
        x = torch.tanh(self.l1(x))
        x = torch.tanh(self.l2(x))
        x = self.l3(x)
        return x
```

Remove bias and use antisymmetric activation function → every layer is antisymmetric
Imposing the symmetry

Learn $f(x) = x^3$

$$\Delta_{\text{sym}}(x) = \frac{h(+x) + h(-x)}{2}$$

Symmetry: $f(-x) = -f(x)$

exactly preserved by construction (without bias, tanh)
Generalization beyond training domain

With bias, sigmoid

Without bias, tanh

Generalization is difficult.

Imposed symmetry is preserved everywhere.

(This does not mean one can trust the result everywhere.)
Generalization beyond training domain

With bias, sigmoid

With bias, ReLU

Commonly used activation function:

\[ \text{ReLU} = \max(0, z) \]

\[ \text{Sigmoid} = \frac{1}{1 + e^{-z}} \]

Slightly better generalization possible, but symmetry not exactly preserved

Symmetries and ML

Andreas Ipp
Trying to solve for wrong symmetry

With bias, sigmoid

\[ \Delta_{\text{sym}}(x) = \frac{h(x) - h(-x)}{2} \]

Epoch 1000

Without bias, tanh

\[ \Delta_{\text{sym}}(x) = \frac{h(x) - h(-x)}{2} \]

Epoch 1000

Generic network can also learn symmetric function approximately.
Antisymmetric fit to a symmetric solution fails.

Learn \( f(x) = x^2 \)
Translational symmetry
Convolutional neural networks

Dense neural network:

→ Every input node connected to every output node

Convolutional Neural Network (CNN):

→ local information: only nodes „nearby“ are connected
→ Weight sharing by sliding the same kernel across the whole image


Image: https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1
Deep learning

1960s: shallow neural networks
1960-70s: backpropagation
1980s: convolutional networks (CNN)
1990s: supervised deep learning
2006s: modern deep learning
2012: AlexNet (first GPU CNN)
Equivariance (covariance) vs. invariance

Equivariance:
\[ \Phi(L_g x) = L_g' \Phi(x) \]

Invariance:
\[ \Phi(L_g x) = \Phi(x) \]

Adapted from: https://towardsdatascience.com/sesn-cec766026179
Translational symmetry

Equivariant architecture (EQ)

Strided architecture (ST)

Flattening architecture (FL)

Bulusu, Favoni, Al, Müller, Schuh, Phys. Rev. D 104 (2021) 074504
Complex scalar field in 1+1D

Complex scalar field action:

\[ S = \int dx_0 dx_1 \left( |D_0 \phi|^2 - |\partial_1 \phi|^2 - m^2 |\phi|^2 - \lambda |\phi|^4 \right) \]

\[ D_0 = \partial_0 - i \mu \]

Lattice formulation:

\[ S_{lat} = \sum_x \left( \eta |\phi_x|^2 + \lambda |\phi_x|^4 - \sum_{\nu=1}^2 \left( e^{\mu \delta_{\nu,2} \phi_x^* \phi_{x+\hat{\nu}}} + e^{-\mu \delta_{\nu,2} \phi_x^* \phi_{x-\hat{\nu}}} \right) \right) \]

\[ \eta = 2D + m^2 \]

Dual formulation with integer fields \( k, l \)

\( \phi_x \rightarrow \{ k_{x,\nu}, l_{x,\nu} \} \)

solves sign problem \( \text{Gattringer, Kloiber (2013)} \)

Observables:

Particle number density:

\[ n = \frac{1}{N} \sum_x k_{x,2} \]

Squared absolute value of field:

\[ |\phi|^2 = \frac{1}{N} \sum_x \frac{W(f_x + 2)}{W(f_x)} \]

\[ f_x = \sum_\nu \left( |k_{x,\nu}| + |k_{x-\hat{\nu},\nu}| + 2(l_{x,\nu} + l_{x-\hat{\nu},\nu}) \right) \]

\[ W(f_x) = \int_0^\infty dx x^{f_x + 1} e^{-\eta x^2 - \lambda x^4} \]

Symmetries and ML

Andreas Ipp
Predicted vs. true values

Generalization to smaller chemical potential:

\[ \langle n \rangle, \langle |\phi|^2 \rangle \]

\[ \begin{align*}
\langle n \rangle &= \frac{T}{V} \frac{\partial \ln Z}{\partial \mu} = \frac{1}{N_x N_t} \left\langle \sum_x k_{x,2} \right\rangle \\
\langle |\phi|^2 \rangle &= \frac{T}{V} \frac{\partial \ln Z}{\partial \eta} = \frac{1}{N_x N_t} \left\langle \sum_x \frac{W(f_x + 2)}{W(f_x)} \right\rangle
\end{align*} \]

Ensemble averages for each \( \mu \):

- Best EQ (60 × 4)

Training point

Generalization to larger lattices:

\[ \langle n \rangle, \langle |\phi|^2 \rangle \]

Predicts Silver Blaze phase transition

Best EQ (200 × 10)

Bulusu, Favoni, Al, Müller, Schuh, Phys. Rev. D 104 (2021) 074504
Comparison of architecture types

For fair comparison, best architectures for each type have been obtained by an Optuna optimization (scanning through various kernel sizes, number of layers, number of channels, …)

Best architectures are retrained 10 times and evaluated on the validation set.

Test regression tasks on observables of a scalar field model in 2 dimensions:

Bulusu, Favoni, AI, Müller, Schuh, Phys. Rev. D 104 (2021) 074504
Why can the models generalize so well?

Without ensemble average, individual configurations cover a large range of possible output values.

Train at \( \mu = 1.05 \)
and \( 60 \times 4 \)

Test for \( \mu \in [0.91, 1.05] \)
and various lattice sizes

Without ensemble average, individual configurations cover a large range of possible output values.
Generalization to larger $\mu$

Train at $\mu = 1.05$

Test for $\mu \in [0.91, 1.05]$

Test for $\mu \in [1.1, 1.5]$

(No ensemble average)
Detection of flux violation

(a) Example field configuration

(b) Feature maps of convolutional network in best EQ and ST models
Detection of flux violation

Generalization to different number of worms

Generalization to different lattice sizes
Lattice gauge symmetry
Symmetries on the lattice

Translational symmetry
→ Convolutional neural networks (CNNs)

Rotation, mirror symmetry
→ Group equivariant CNNs (G-CNNs)

Lattice gauge symmetry
→ Lattice gauge equivariant CNNs (L-CNNs)

---

Bulusu, Favoni, Al, Müller, Schuh, Phys. Rev. D 104 (2021) 074504
Cohen, Welling, ICML 2016
Favoni, Al, Müller, Schuh, Phys. Rev. Lett. 128 (2022) 032003
Yang-Mills action vs. Wilson action

Yang-Mills action

Continuum formulation

\[ S_G[A] = \frac{1}{2 g^2} \int d^4x \, \text{tr} \left[ F_{\mu\nu}(x) F_{\mu\nu}(x) \right] \]

Field strength tensor

\[ F_{\mu\nu}(x) = -i[D_\mu(x), D_\nu(x)] = \partial_\mu A_\nu(x) - \partial_\nu A_\mu(x) + i[A_\mu(x), A_\nu(x)] \]

Covariant derivative

\[ D_\mu(x) = \partial_\mu + i A_\mu(x) \]

SU(3) gauge fields

\[ A_\mu(x) = \sum_{i=1}^{8} A^{(i)}(x) T_i \]

Gauge transformation

\[ A_\mu(x) \rightarrow A'_\mu(x) = \Omega(x) A_\mu(x) \Omega(x)^\dagger + i \left( \partial_\mu \Omega(x) \right) \Omega(x)^\dagger \]

Taylor expansion in small lattice spacing reproduces continuum action:

\[ U_{\mu\nu}(n) = \exp \left( i a^2 F_{\mu\nu}(n) + O(a^3) \right) \]

Discrete formulation

Wilson action

\[ S_W[U] = \frac{2}{g^2} \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{Tr} \left[ 1 - U_{x,\mu\nu} \right] \]

Wilson (1974)

Plaquette

\[ U_{x,\mu\nu} = U_{x,\mu} U_{x+\mu,\nu} U_{x+\nu,\mu}^\dagger U_{x,\nu}^\dagger = \]

Link variable

\[ U_\mu(n) = \exp \left( i a A_\mu(n) \right) \]

Gauge transformation

\[ U_\mu(n) \rightarrow U'_\mu(n) = \Omega(n) U_\mu(n) \Omega(n + \hat{\mu})^\dagger \]

from Gattringer, Lang (2010)

Symmetries and ML

Andreas Ipp

38
Wilson loops

Wilson action

\[ S_W[U] = \frac{2}{g^2} \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{Tr} [1 - U_{x, \mu \nu}] \]

Plaquette

\[ U_{x, \mu \nu} = U_{x, \mu} U_{x+\mu, \nu} U_{x+\nu, \mu}^{-1} U_{x, \nu}^{-1} \]

Symanzik improved clover action

from: Gattringer, Lang (2010)

Potential of static quark pair


Improved real-time lattice actions

Improved topological charge


Symmetries and ML

Andreas Ipp
L-CNN data

Combine lattice links $U$ and locally transforming objects $W$

- tuple $(U, W)$
  - $U = \{ U_{x,\mu} \}$ SU($N_c$) matrices
  - $W = \{ W_{x,i} \}$ with $W_{x,i} \in \mathbb{C}^{N_c \times N_c}$

Gauge transformation

- $T_\Omega U_{x,\mu} = \Omega_x U_{x,\mu} \Omega_{x+\mu}^\dagger$
- $T_\Omega W_{x,i} = \Omega_x W_{x,i} \Omega_x^\dagger$

Gauge equivariant (gauge covariant) function

$$f(T_\Omega U, T_\Omega W) = T'_\Omega f(U, W)$$

Gauge invariant function

$$f(T_\Omega U, T_\Omega W) = f(U, W)$$

from: Gattringer, Lang (2010)
Lattice gauge equivariant layers

**Convolution (L-Conv)**

Convolution with shared weights and proper parallel transport along coordinate axes

\[(U, W) \rightarrow (U, W')\]

\[W_{x,i} = \sum_{j, \mu, k} \omega_{i,j,\mu,k} U_{x,\mu,k} W_{x+k,\mu,j} U_{x+k,\mu,j}^\dagger\]

**Bilinear layer (L-Bilin)**

Multiply \(W\) at each lattice point

\[(U, W) \times (U, W') \rightarrow (U, W'')\]

\[W''_{x,i} = \sum_{j,k} \alpha_{i,j,k} W_{x,j} W'_{x,k}\]

**Trace layer**

Generate gauge invariant output

\[w_{x,i} = \text{Tr} W_{x,i} \in \mathbb{C}\]

---

Symmetries and ML

Andreas Ipp

41
**Generic L-CNN**

**L-Conv:**
* convolution of parallel-transported $W$ objects
* parallel transport only along coordinate axes

**L-Bilin:**
* bilinear layer, product of locally transforming objects

**L-Act:**
* activation functions multiply $W$ objects by scalar, gauge-invariant functions

**L-Exp:**
* update link variables using exponential map

**Trace:**
* calculate gauge invariant trace

**Plaq:**
* generate all possible plaquettes

**Poly:**
* generate all possible Polyakov loops

---

Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003
L-CNNs generate Wilson loops

Number of traced Wilson loops covered by L-CNN architectures of various sizes in 1+1 D

<table>
<thead>
<tr>
<th>Length</th>
<th>Max</th>
<th>( W^{(1 \times 1)} )</th>
<th>( W^{(1 \times 2)} )</th>
<th>( W^{(2 \times 2)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( S )</td>
<td>( M )</td>
<td>( L )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>28</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>152</td>
<td>48</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>12</td>
<td>1,010</td>
<td>92</td>
<td>204</td>
<td>220</td>
</tr>
<tr>
<td>14</td>
<td>6,772</td>
<td>120</td>
<td>412</td>
<td>532</td>
</tr>
<tr>
<td>16</td>
<td>47,646</td>
<td>100</td>
<td>712</td>
<td>1,080</td>
</tr>
<tr>
<td>18</td>
<td>343,168</td>
<td>136</td>
<td>928</td>
<td>1,896</td>
</tr>
<tr>
<td>20</td>
<td>2,529,890</td>
<td>32</td>
<td>1,056</td>
<td>2,620</td>
</tr>
<tr>
<td>22</td>
<td>18,982,172</td>
<td>64</td>
<td>768</td>
<td>3,152</td>
</tr>
<tr>
<td>( \geq 24 )</td>
<td>( \geq 24 )</td>
<td>( \geq 24 )</td>
<td>( \geq 24 )</td>
<td>( \geq 24 )</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>11</td>
<td>19</td>
<td>27</td>
</tr>
<tr>
<td>Max.Len</td>
<td>4</td>
<td>10</td>
<td>12</td>
<td>22</td>
</tr>
</tbody>
</table>

Architectures differ in number of layers, kernel size, and number of channels.

Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003
Sketch of proof for arbitrary Wilson loops

(a) An arbitrary contractible Wilson loop of \( n \) tiles ...
(b) ... is composed (L-Bilin) of a Wilson loop of \( (n-1) \) tiles ...
(c) ... and a parallel-transported (L-Conv) plaquette (Plaq).

Non-contractible loops (like Polyakov loops) have to be added (Poly).

Favoni, Al, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003
Numerical results

Regression task to learn value of rectangular Wilson loops:

\[ W^{(m \times n)}_{x,\mu\nu} = \text{Re} \ Tr \left[ U^{(m \times n)}_{x,\mu\nu} \right] \]

Lattice gauge equivariant CNN (L-CNNs, green) can learn the relation, while traditional convolutional neural networks (CNNs, black) struggle to find the solution.

Training on 8 \times 8, testing from 8 \times 8 up to 64 \times 64

Compared best from:
100 L-CNN models (10 \times 10^4 trainable parameters, up to 4 L-Conv+L-Bilin)
2840 CNN models (100 \times 10^5 trainable parameters up to 6 layers, 512 channels, 4 activation functions)

Favoni, Al, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003
Adversarial attacks

L-CNNs are insensitive to random or adversarial gauge transformations

From Goodfellow, Shlens, Szegedy ICLR 2015

Favoni, Ai, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003
Fixed point action

Critical slowing down, topological freezing

Introduce a renormalization group transformation (RGT)

\[ \exp \left\{ -\beta' A'[V] \right\} = \int \mathcal{D}U \exp \left\{ -\beta (A[U] + T[U, V]) \right\} \]

The effective action \( \beta' A'[V] \) is described by infinitely many couplings \( \{ c'_\alpha \} \)

The fixed point is the saddle point in the classical limit \( \beta \to \infty \), which can be found by a minimization condition.


Symmetries and ML

Andreas Ipp
Many possibilities to construct a blocking kernel manually, fit corresponding parameters.
Learning the fixed point action with L-CNNs

Training example: L-CNN model with 3 layers with 12, 24, 24 channels and kernel size 2, 2, 1.

L-CNN superior to older parametrizations of FP action.

Holland, Al, Müller, Wenger, in preparation
Continuous formulation of L-CNNs

Define a continuous version of a gauge equivariant convolution:

\[
[\psi \ast \mathcal{W}]^a(x) = \sum_b \int_{\mathbb{R}^D} dy^D \psi^{ab}(y - x) U_{x \rightarrow y} W^b(y) U^\dagger_{x \rightarrow y}
\]

with kernel components \( \psi^{ab} : \mathbb{R}^D \rightarrow \mathbb{R} \)

and parallel transporter \( U_{x \rightarrow y} = \mathcal{P} \exp \left\{ i \int_0^1 ds \frac{dx^\nu(s)}{ds} A_\nu(x(s)) \right\} \)

that map \( \mathcal{W} = (\mathcal{W}^1, \ldots, \mathcal{W}^m) \) objects to new objects in a gauge equivariant manner:

\[
[\psi \ast T_\Omega \mathcal{W}]^a(x) = T_\Omega [\psi \ast \mathcal{W}]^a(x)
\]

Similarly define continuous bilinear layer, trace layer, ...

Discretize this to obtain previous formulation.

Compatible with G-CNNs.

Generalizable to vectors and tensors.

Aronsson, Müller, Schuh, arxiv:2303.11448
“Upscaling Glasma simulations using machine learning”
Austrian Science Fund FWF No. P32446-N27

Open source: https://gitlab.com/openpixi/lge-cnn