Trends in Chemistry

CellPress

Special Issue: Machine Learning for Molecules and Materials

Forum

Euclidean Symmetry and Equivariance in Machine Learning

Tess E. Smidt[1,2,*]

Understanding the role of symmetry in the physical sciences is critical for choosing an appropriate machine-learning method. While invariant models are the most prevalent symmetry-aware models, equivariant models such as Euclidean neural networks more faithfully represent physical interactions and are ready to take on challenges across the physical sciences.

Euclidean Symmetry Is the Freedom to Choose a Coordinate System

There is no inherent way to orient physical systems; yet we still need to choose a coordinate system to articulate their geometry and physical properties (Figure 1A). Unless coded otherwise, machine-learned models make no assumption of symmetry and are sensitive to an arbitrary choice of coordinate system. To be able to recognize a 3D pattern in any orientation, such a model will need to be shown roughly 500 rotated versions of the same example, a process called data augmentation. One of the motivations of explicitly treating symmetry in machine-learning models is to eliminate the need for data augmentation, so that the model can instead focus on, for example, learning quantum mechanics. In 3D space, we can transform between coordinate systems using elements of Euclidean symmetry [3D rotations, 3D translations, and inversion (x, y, z) → (−x, −y, −z), which includes mirror symmetry]; hence, we say that 3D space has Euclidean symmetry.

One useful way of categorizing machine learning models applied in the physical sciences is by whether they use symmetry and, if so, where they use **invariant** (see Glossary) versus **equivariant operations**. Between the two types of symmetry-aware models, invariant models eliminate coordinate systems by dealing only with quantities that are invariant to the choice of coordinate system (scalars), while **equivariant models** preserve how quantities predictably change under coordinate transformations.

Invariance Is Easier to Deal with Than Equivariance

There is good reason for the popularity of invariant scalar features for machine learning; scalars can be given to any machine learning algorithm without violating symmetry. More practically, scalars are easier to handle than **geometric tensors** and invariant models are high (if not top) performers on many existing benchmarks [1].

It is common practice to use equivariant operations to generate invariant features for machine learning models. For example, SOAP [2] descriptors are **equivariant** functions that operate on the geometry and atom types of local atomic environments to produce invariant scalars (Figure 2A). Geometry in invariant models is reduced to bond lengths, bond angles, dihedral angles, and other scalar invariants of geometric elements; Figure 1B describes the invariant versus equivariant properties of 3D vectors. Invariant models can yield equivariant quantities, but only through gradients of the equivariant operations used in featurization [3]; this may not be practical or possible depending on the quantity of interest.

When Invariance Is Not Enough

Physical systems and their interactions are inherently equivariant. With an invariant model, a featurization method must be used to represent a naturally equivariant physical system in terms of invariant features. With an equivariant model, a physical system can be articulated using the same means used by many physical simulations: with geometric coordinates and any relevant quantities you need to describe the system (e.g., external fields, atom-wise properties such as velocities). Even if the desired target is a scalar, the interactions that yield that scalar may not be scalar in nature; for example, the only way to interact a particle with charge q and velocity v with an external magnetic field $\vec{B}$ is to use the equivariant cross-product, $\vec{F} = q\vec{v} \times \vec{B}$, or the difference between the momentum $\vec{p}$ and the charge-weighted vector potential $q\vec{A}, H = |\vec{p} - q\vec{A}|^2/2m$. To predict quantities that are fundamentally generated from equivariant interactions: (i) equivariant interactions must be included in the scalar featurization used for an invariant model (which requires knowing to include those interactions); or (ii) an equivariant model may be used, which may make more accurate or efficient predictions because it has more expressive operations [4]. The more

Glossary

3D Euclidean symmetry: the symmetry group of 3D space; includes 3D translations, rotations, and inversion.
3D point groups: subgroups of 3D Euclidean symmetry; symmetry groups of molecules; includes rotations, mirrors, and inversion.
3D space groups: subgroups of 3D Euclidean symmetry; symmetry groups of crystals; includes rotations, mirrors, glides, screws, inversion, and discrete translations.
Equivariant: changes deterministically under operation of group elements.
Equivariant model: a machine learning model that can handle equivariant quantities as input, intermediate data, and output and uses equivariant (i.e., tensor) operations.
Equivariant operations: operations on equivariant quantities (i.e., geometric tensors); for example, dot products, cross-products, tensor products; includes invariant operations.
Euclidean neural networks: neural networks that are equivariant to 3D Euclidean symmetry.
Geometric tensors: equivariant properties of 3D physical systems; includes invariant scalars.
Invariant: does not change under operation of group elements.
Neural networks: a subset of machine learning methods where models must be differentiable; model weights are updated using gradients of the error; also known as deep learning.

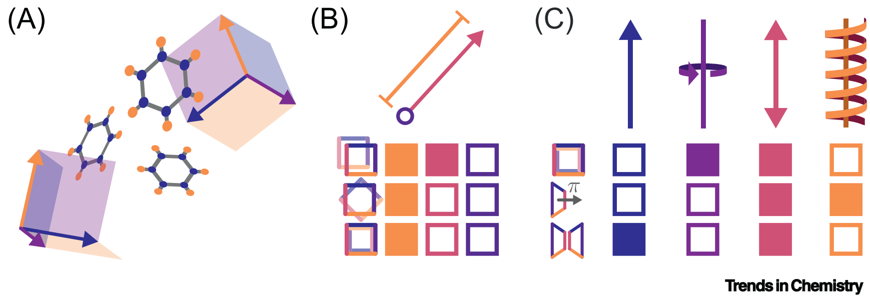

Figure 1. Examples of Equivariant and Invariant Properties of Geometric Objects in 3D. (A) A system of molecules described by two different coordinate systems. (B) A 3D vector has a magnitude (orange), direction (pink), and location (purple), which are invariant (filled square) or equivariant (open square) under (table from top to bottom) 3D translations, rotations, and inversion. The magnitude is a scalar and thus invariant to all of these operations. (C) In Euclidean symmetry there are four 'vector-like' geometric tensors that transform distinctly under (table from top to bottom) inversion, perpendicular rotation by π, and parallel reflection: a vector, a pseudovector, a double-headed vector, and a helix [5].

exotic or complex a property, the more likely there are non-trivial geometric tensor interactions at play (e.g., multipole interactions, antisymmetric exchange).

Euclidean Neural Networks Are Equivariant Models

Neural networks are one of the most flexible machine learning methods since the only requirement is that the network be differentiable. A neural network is a function f that takes in inputs x and trainable parameters W to produce outputs y, $f(x, W) = y$. Given pairs of inputs x and target output y_{true}, a neural network is

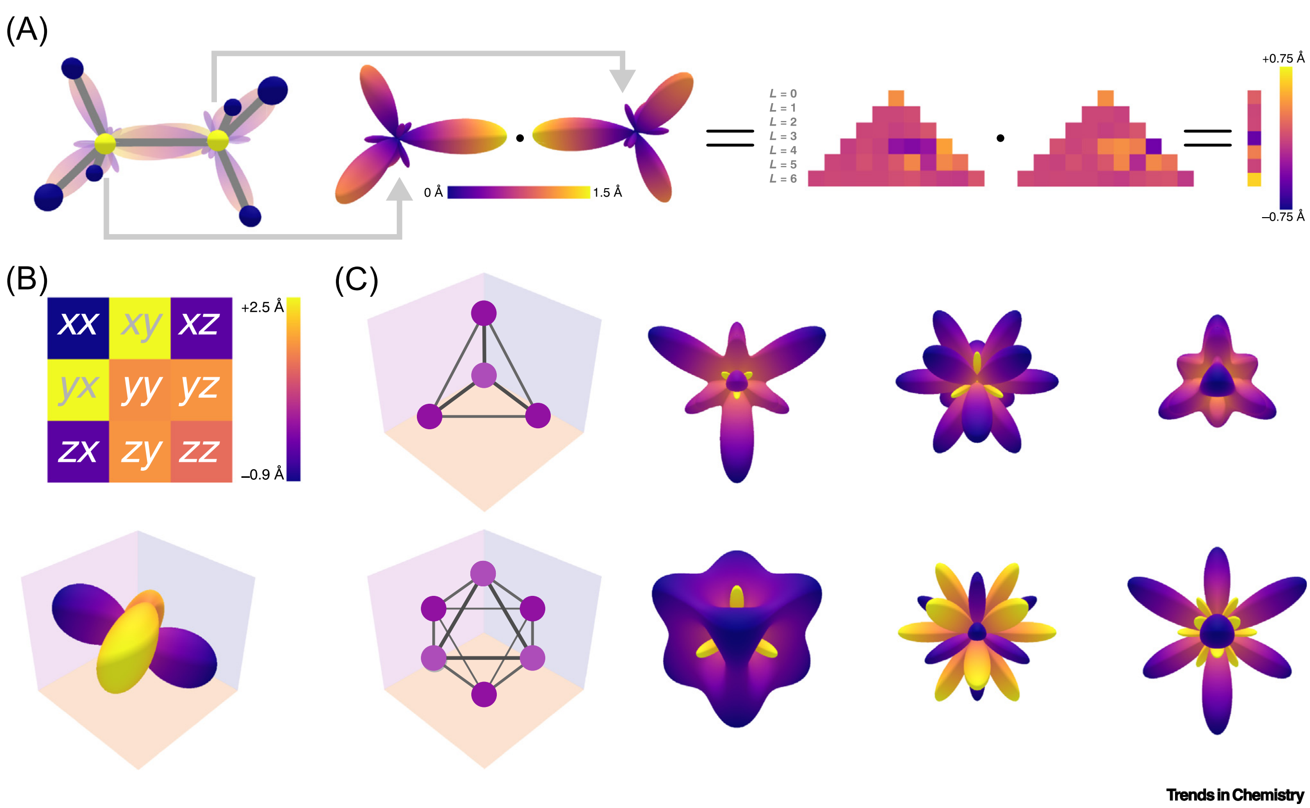

Figure 2. Visualizing Equivariant Properties as Spherical Harmonic Projections. (A) A simplified example of how invariant descriptors are derived from equivariant operations. The local atomic environments of the two carbons in an ethane molecule can be projected onto spherical harmonics (shown for $L \leq 6$ with negative magnitudes set to zero). The coloring on the projection surface signifies the distance from the origin. The coefficients of the projection, organized in the visual by rows of equal degree L and columns of equal order m, form a geometric tensor in the irreducible basis (the same basis as spherical harmonics). One method of computing scalars from these two geometric tensors is by computing the 'power spectrum', which is the tensor generalization of the dot product. This results in seven scalars (one for each L used in the projection). (B) A symmetric 3×3 matrix and its visualization as a spherical harmonic signal. The magnitude of the signal at a given angle is interpreted as a radial distance and the projection is colored to signify the magnitude. For both (B) and (C), yellow indicates the maximum value and dark purple the minimum value, which in this case can be negative. (C) The output of three randomly initialized Euclidean neural networks when given a tetrahedron geometry (top) and octahedron geometry (bottom) as input. The outputs have equal or higher symmetry than the inputs.

trained by computing derivatives of the loss [e.g., $\mathcal{L} = (y - y_{true})^2$] with respect to trainable parameters W, and updating W according to a learning rate η, $W = W + \eta \frac{\partial \mathcal{L}}{\partial w}$.

First proposed in 2018, **Euclidean neural networks** (tensor field networks [6], Clebsch–Gordan nets [7], 3D steerable CNNs [8], and their descendants [9,10]) are a flexible, general framework for learning **3D Euclidean symmetry** equivariant functions that can be trained on the context of a given dataset. To achieve equivariance in Euclidean neural networks, scalar multiplication is replaced by the more general tensor product and convolutional kernels are restricted to be composed of spherical harmonics and learnable radial functions, $W(\vec{r}) = R(|r|)Y_{lm}(\hat{r})$. Scalar nonlinearities must also be replaced with equivariant equivalents.

Due to these mathematical complexities, Euclidean neural networks can be challenging to implement; however, there are open-source implementations (e.g., *e3nn* is an open-source *PyTorch* library that combines the implementations of [6,8] and implements a variety of additional equivariant layers and utility functions for converting and visualizing geometric tensors)[i].

Equivariance has also been encoded in other machine learning approaches such as kernel methods by using equivariant kernels and an equivariant definition of covariance [11]. Euclidean neural networks can learn equivariant functions that generate scalar invariants or equivariant kernels for use with these traditional machine learning methods.

Euclidean neural networks can be used to build end-to-end models for the prediction of physical properties (e.g., molecular dynamics forces) from atomic geometries and initial atomic features (e.g., atom types). They can be used to manipulate atomic geometries and craft hierarchical features [12]. The only difference between models for these varied purposes is how equivariant operations and learnable equivariant modules are composed.

Euclidean Neural Networks Naturally Handle Geometric Tensors, Point Groups, and Space Groups

Euclidean neural networks provides a mathematically rigorous framework for articulating scientific questions in terms of geometric tensors and their tensor interactions. Inputs, outputs, and intermediate data are completely specified by their transformation properties.

Geometric tensors take many forms and can represent many different quantities: numerical geometric tensors, atomic orbitals, or projections of geometry. Figures 1C and 2B give additional examples of this variety.

Another particularly useful aspect of handling Euclidean symmetry in full generality is that you get all subgroup symmetries (e.g., **3D point groups**, **3D space groups**) for free. Figure 2C shows an example of how the output of even randomly initialized Euclidean neural networks will have equal or higher symmetry than the input.

Since these networks intrinsically uphold any and all selection rules that occur in physical systems, they act as 'symmetry compilers' that check your thinking about the data types of your physical system. Using these models requires more forethought than a traditional neural network; in exchange, they cannot learn to do something that does not symmetrically make sense.

Beginning to Uncover Features of *E*(3) Equivariant Models

Euclidean symmetry is a simple assumption that has many unintuitive consequences: geometry, geometric tensors, normal modes, selection rules in spectroscopy, space groups, point groups, multipole interactions, second-order phase transitions, and so on. Likewise, the uses for Euclidean symmetry equivariant machine learning models may similarly surprise us in the as-yet-untapped modes of investigation they offer. For example, we recently found that the gradients of Euclidean neural networks can be used to find symmetry-implied 'missing' data unbeknown to the researcher (e.g., order parameters of second-order phase transitions [13]). By bounding learnable functions of physical data to be equivariant to 3D Euclidean symmetry, we can ask more targeted questions with machine learning and explore the full-ranging consequences of our fundamental assumption of symmetry.

Acknowledgments

T.E.S. thanks Lucy Morrell, Sinéad Griffin, Josh Rackers, and Risi Kondor for helpful discussions. T.E.S. was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under US Department of Energy Contract No. DE-AC02-05CH11231.

Resources

[i] https://doi.org/10.5281/zenodo.3723557

[1]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 947020, USA
[2]Center for Advanced Mathematics for Energy Research Applications (CAMERA), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

*Correspondence:
tsmidt@lbl.gov (T.E. Smidt).

https://doi.org/10.1016/j.trechm.2020.10.006

References

1. Klicpera, J. *et al.* (2020) Directional Message Passing for Molecular Graphs, *International Conference on Learning Representations*
2. Bartók, A.P. *et al.* (2013) On representing chemical environments. *Phys. Rev. B* 87, 184115
3. Schütt, K. *et al.* (2017) Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. *Adv. Neural Inf. Proces. Syst.* 31, 991–1001
4. Miller, B.K. *et al.* (2020) Relevance of rotationally equivariant convolutions for predicting molecular properties. *arXiv* Published online August 22, 2020. https://arxiv.org/abs/2008.08461
5. Hlinka, J. (2014) Eight types of symmetrically distinct vectorlike physical quantities. *Phys. Rev. Lett.* 113, 165502
6. Thomas, N. *et al.* (2018) Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv* Published online May 18, 2018. https://arxiv.org/abs/1802.08219

7. Kondor, R. *et al.* (2018) Clebsch–Gordan nets: a fully Fourier space spherical convolutional neural network. *Adv. Neural Inf. Proces. Syst.* 32, 10117–10126
8. Weiler, M. *et al.* (2018) 3D steerable CNNs: learning rotationally equivariant features in volumetric data. *Adv. Neural Inf. Proces. Syst.* 32, 10402–10413
9. Anderson, B.M. *et al.* (2019) Cormorant: covariant molecular neural networks. *Adv. Neural Inf. Proces. Syst.* 32, 14510–14519
10. Fuchs, F.B. *et al.* (2020) *SE*(*3*)-transformers: 3D roto-translation equivariant attention networks. *arXiv* Published online June 22, 2020. https://arxiv.org/abs/2006.10503
11. Grisafi, A. *et al.* (2018) Symmetry-adapted machine learning for tensorial properties of atomistic systems. *Phys. Rev. Lett.* 120, 036002
12. Eismann, S. *et al.* (2020) Hierarchical, rotation-equivariant neural networks to predict the structure of protein complexes. *arXiv* Published online June 5, 2020. https://arxiv.org/abs/2006.09275
13. Smidt, T.E. *et al.* (2020) Finding symmetry breaking order parameters with Euclidean neural networks. *arXiv*. Published online July 4, 2020. https://arxiv.org/abs/2007.02005