The Journal of Open Source Software

ginjax: E(d)-Equivariant CNN for Tensor Images

Wilson G. Gregory [1¶], **Kaze W. K. Wong** [1,2], **David W. Hogg** [3,4,5], and **Soledad Villar** [1,6,7]

1 Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, United States **2** Data Science and AI Institute, Johns Hopkins University, Baltimore, MD, United States **3** Center for Cosmology and Particle Physics, Department of Physics, New York University, New York, NY, United States **4** Max-Planck-Institut für Astronomie, Heidelberg, Germany **5** Center for Computational Astrophysics, Flatiron Institute, New York, NY, United States **6** Center for Computational Mathematics, Flatiron Institute, New York, NY, United States **7** Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD, United States ¶ Corresponding author

DOI: 10.21105/joss.08129

Software

- Review
- Repository
- Archive

Editor: Rachel Kurchin

Reviewers:

- @ameya98
- @zhanglw0521

Submitted: 25 March 2025
Published: 25 August 2025

Summary

Many data sets encountered in machine learning exhibit symmetries that can be exploited to improve performance, a technique known as equivariant machine learning. The classical example is image translation equivariance that is respected by convolutional neural networks (LeCun et al., 1989). For data sets in the physical sciences and other areas, we would also like equivariance to rotations and reflections. This Python package implements a convolutional neural network that is equivariant to translations, rotations of 90 degrees, and reflections. We implement this by *geometric convolutions* (Gregory et al., 2025) which use tensor products and tensor contractions. This additionally enables us to perform functions on geometric images, or images where each pixel is a higher order tensor. These images appear as discretizations of fields in physics, such as velocity fields, vorticity fields, magnetic fields, polarization fields, and so on.

The key features and use cases are summarized below.

Key Features

1. Create, visualize and perform mathematical operations on geometric images, including powerful `jax` (Bradbury et al., 2018) features such as vmap.
2. Combine geometric images of any tensor order or parity into a single `MultiImage` data structure.
3. Build `equinox` (Kidger & Garcia, 2021) neural networks with our custom equivariant layers that process `MultiImages`.
4. *Or*, use one of our off-the-shelf models (UNet, ResNet, etc.) to start processing your geometric image datasets right away.

Statement of need

The geometric convolutions introduced in Gregory et al. (2025) are defined on geometric images – images where every pixel is a tensor. If A is a geometric image of tensor order k and C is a geometric image of tensor order k', then the value of A convolved with C at pixel $\bar{\imath}$ is given by:

$$(A * C)(\bar{\imath}) = \sum_{\bar{a}} A(\bar{\imath} - \bar{a}) \otimes C(\bar{a}) \ ,$$

where the sum is over all pixels $\bar{a}$ of C, and $\bar{\imath} - \bar{a}$ is the translation of $\bar{\imath}$ by $\bar{a}$. The result is a geometric image of tensor order $k + k'$. To produce geometric images of smaller tensor order, a tensor contraction can be applied to each pixel. Convolution and contraction are combined into a single operation to form linear layers. By restricting the convolution filters C to rotation and reflection invariant filters, we can create linear layers which are rotation-, reflection-, and translation-equivariant.

For machine learning practitioners

The ginjax package can be used as a drop-in replacement for CNNs with minimal code changes required. We define equivariant versions for all the common CNN operations including convolutions, activation functions, group norms, pooling, and unpooling. Each of these layers require keeping track of the tensor order and parity of each geometric image, so we define a special data structure, the `MultiImage`, for these equivariant layers to operate on. We can then easily turn a non-equivariant CNN into an equivariant CNN by replacing the layers and converting the input to a `MultiImage`. We also provide full-fledged model implementations such as the UNet, ResNet, and Dilated ResNet.

This package is the only one implementing geometric convolutions, but there are alternative methods for solving $O(d)$-equivariant image problems. One such package is escnn which uses Steerable CNNs (Cohen & Welling, 2016; Weiler & Cesa, 2021). Steerable CNNs use irreducible representations to derive a basis for $O(d)$-equivariant layers, but it is not straightforward to apply on higher-order tensor images.

Other alternative methods are those based on Clifford Algebras, in particular Brandstetter et al. (2023). This method has been implemented in the Clifford Layers package. Clifford based methods can process vectors and pseudovectors, but cannot handle higher-order tensors. Additionally, both these methods are built with pytorch, rather than `jax`.

For equivariance researchers

To allow researchers to explore the behavior of geometric images, we implement all the common operations such as addition, scaling, convolution, contraction, transposition, norms, rotations, and reflections. This makes it easy to generate group-invariant images and experiment with equivariant functions. We also provide visualization methods to easily follow along with the operations.

References

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). http://github.com/jax-ml/jax

Brandstetter, J., Berg, R. van den, Welling, M., & Gupta, J. K. (2023). *Clifford neural layers for PDE modeling*. https://doi.org/10.48550/arXiv.2209.04934

Cohen, T. S., & Welling, M. (2016). *Steerable CNNs*. https://doi.org/10.48550/arXiv.1612.08498

Gregory, W. G., Hogg, D. W., Blum-Smith, B., Arias, M. T., Wong, K. W. K., & Villar, S. (2025). Equivariant geometric convolutions for dynamical systems on vector and tensor images. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *383*(2298), 20240247. https://doi.org/10.1098/rsta.2024.0247

Kidger, P., & Garcia, C. (2021). Equinox: Neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming Workshop at Neural Information

Processing Systems 2021. https://doi.org/10.48550/arXiv.2111.00254

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Weiler, M., & Cesa, G. (2021). *General $E(2)$-equivariant steerable CNNs*. https://doi.org/10.48550/arXiv.1911.08251