Clifford Group Equivariant Neural Networks

David Ruhe
AI4Science Lab, AMLab, API
University of Amsterdam
david.ruhe@gmail.com

Johannes Brandstetter
Microsoft Research
AI4Science
brandstetter@ml.jku.at

Patrick Forré
AI4Science Lab, AMLab
University of Amsterdam
p.d.forre@uva.nl

1 Introduction

Incorporating *group equivariance* to ensure symmetry constraints in neural networks has been a highly fruitful line of research [14, 55, 15, 39, 52, 54, 18, 9, 53, 11]. Besides translation and permutation equivariance [58, 45], rotation equivariance proved to be vitally important for many graph-structured problems as encountered in, e.g., the natural sciences. Applications of such methods include modeling the dynamics of complex physical systems or motion trajectories [34, 8]; studying or generating molecules, proteins, and crystals [42, 27, 12, 46, 59, 50, 4]; and point cloud analysis [56, 51]. Note that many of these focus on three-dimensional problems involving rotation, reflection, or translation equivariance by considering the groups $\mathrm{O}(3)$, $\mathrm{SO}(3)$, $\mathrm{E}(3)$, or $\mathrm{SE}(3)$.

Such equivariant neural networks can be broadly divided into three categories: approaches that scalarize geometric quantities, methods employing regular group representations, and those utilizing irreducible representations, often of $\mathrm{O}(3)$ [28]. Scalarization methods operate exclusively on scalar features or manipulate higher-order geometric quantities such as vectors via scalar multiplication [46, 16, 35, 37, 44, 31, 23, 47, 17, 29, 48]. They can be limited by the fact that they do not extract all directional information. Regular representation methods construct equivariant maps through an integral over the respective group [14, 39]. For continuous Lie groups, however, this integral is intractable and requires coarse approximation [19, 6]. Methods of the third category employ the irreducible representations of $\mathrm{O}(3)$ (the Wigner-D matrices) and operate in a *steerable* spherical harmonics basis [49, 2, 22, 8, 5]. This basis allows a decomposition into type-l vector subspaces that transform under D^l: the type-l matrix representation of $\mathrm{O}(3)$ [40, 21]. Through tensor products decomposed using Clebsch-Gordan coefficients (Clebsch-Gordan tensor product), vectors (of different types) interact equivariantly. These tensor products can be parameterized using learnable weights. Key limitations of such methods include the necessity for an alternative basis, along with acquiring the Clebsch-Gordan coefficients, which, although they are known for unitary groups of any dimension [32], are not trivial to obtain [1].

We propose *Clifford Group Equivariant Neural Networks* (CGENNs): an equivariant parameterization of neural networks based on *Clifford algebras*. Inside the algebra, we identify the *Clifford group* and its action, termed the (adjusted) *twisted conjugation*, which has several advantageous properties. Unlike classical approaches that represent these groups on their corresponding vector spaces, we carefully extend the action to the entire Clifford algebra. There, it automatically acts as an *orthogonal automorphism* that respects the multivector grading, enabling nontrivial subrepresentations that operate on the algebra subspaces. Furthermore, the twisted conjugation respects the Clifford algebra's multiplicative structure, i.e. the *geometric product*, allowing us to bypass the need for explicit tensor product representations. As a result, we obtain two remarkable properties. First, all polynomials in multivectors generate Clifford group equivariant maps from the Clifford algebra to itself. Additionally,

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

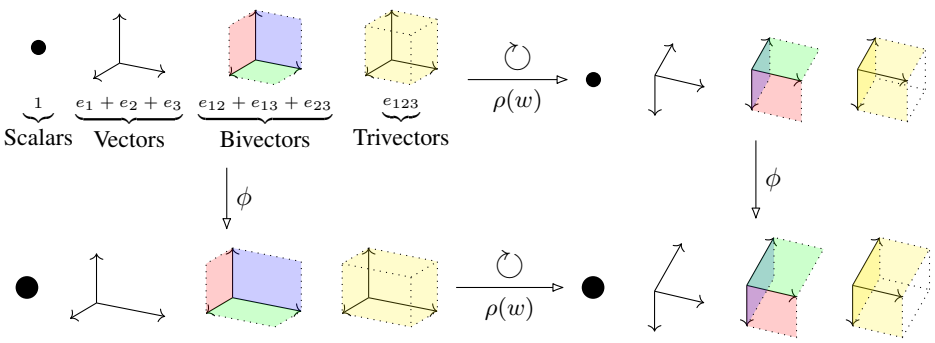

Figure 1: CGENNs (represented with ϕ) are able to operate on multivectors (elements of the Clifford algebra) in an $\mathrm{O}(n)$- or $\mathrm{E}(n)$-equivariant way. Specifically, when an action $\rho(w)$ of the Clifford group, representing an orthogonal transformation such as a rotation, is applied to the data, the model's representations *corotate*. Multivectors can be decomposed into scalar, vector, bivector, trivector, and even higher-order components. These elements can represent geometric quantities such as (oriented) areas or volumes. The action $\rho(w)$ is designed to respect these structures when acting on them.

grade projections are equivariant, allowing for a denser parameterization of such polynomials. We then demonstrate how to construct parameterizable neural network layers using these properties.

Our method comes with several advantages. First, instead of operating on alternative basis representations such as the spherical harmonics, CGENNs (similarly to scalarization methods) directly transform data in a vector basis. Second, multivector representations allow a (geometrically meaningful) product structure while maintaining a finite dimensionality as opposed to tensor product representations. Through geometric products, we can transform vector-valued information, resulting in a more accurate and nuanced interpretation of the underlying structures compared to scalarization methods. Further, we can represent exotic geometric objects such as pseudovectors, encountered in certain physics problems, which transform in a nonstandard manner. Third, our method readily generalizes to orthogonal groups *regardless of the dimension or metric signature of the space*, thereby attaining $\mathrm{O}(n)$- or $\mathrm{E}(n)$-equivariance. These advantages are demonstrated on equivariance benchmarks of different dimensionality. Note that specialized tools were developed for several of these tasks, while CGENNs can be applied more generally.

2 Methodology

We restrict our layers to use $\mathbb{F} := \mathbb{R}$. Our method is most similar to steerable methods such as [8]. However, unlike these works, we do not require an alternative basis representation based on spherical harmonics, nor do we need to worry about Clebsch-Gordan coefficients. Instead, we consider simply a steerable vector basis for V, which then automatically induces a *steerable multivector basis* for $\mathrm{Cl}(V, \mathfrak{q})$ and its transformation kernels. By steerability, we mean that this basis can be transformed in a predictable way under an action from the Clifford group, which acts orthogonally on both V and $\mathrm{Cl}(V, \mathfrak{q})$ (see Figure 1).

We present layers yielding Clifford group-equivariant optimizable transformations. It is worth mentioning that the methods presented here form a first exploration of applying our theoretical results, making future optimizations rather likely.

Linear Layers Let $x_1, \ldots, x_\ell \in \mathrm{Cl}(V, \mathfrak{q})$ be a tuple of multivectors expressed in a steerable basis, where ℓ represents the number of input channels. Using the fact that a polynomial restricted to the first order constitutes a linear map, we can construct a linear layer by setting

$$y^{(k)}_{c_{\text{out}}} := T^{\text{lin}}_{\phi_{c_{\text{out}}}}(x_1, \ldots, x_\ell)^{(k)} := \sum_{c_{\text{in}}=1}^{\ell} \phi_{c_{\text{out}} c_{\text{in}} k}\, x^{(k)}_{c_{\text{in}}}, \tag{1}$$

where $\phi_{c_{\text{out}} c_{\text{in}} k} \in \mathbb{R}$ are optimizable coefficients and c_{in}, c_{out} denote the input and output channel, respectively. As such, $T_\phi : \mathrm{Cl}(V, \mathfrak{q})^\ell \to \mathrm{Cl}(V, \mathfrak{q})$ is a linear transformation in each algebra subspace

k. Recall that this is possible due to the result that $\rho(w)$ respects the multivector subspaces. This computes a transformation for the output channel c_{out}; the map can be repeated (using different sets of parameters) for other output channels, similar to classical neural network linear layers. For $y^{(0)}_{c_{\text{out}}} \in \mathbb{R}$ (the scalar part of the Clifford algebra), we can additionally learn an invariant bias parameter.

Geometric Product Layers A core strength of our method comes from the fact that we can also parameterize interaction terms. In this work, we only consider layers up to second order. Higher-order interactions are indirectly modeled via multiple successive layers. As an example, we take the pair x_1, x_2. Their interaction terms take the form $\left(x_1^{(i)} x_2^{(j)}\right)^{(k)}, i, j, k = 0, \ldots, n$; where we again make use of the fact that $\rho(w)$ respects grade projections. As such, all the grade-k terms resulting from the interaction of x_1 and x_2 are parameterized with

$$P_\phi(x_1, x_2)^{(k)} := \sum_{i=0}^{n} \sum_{j=0}^{n} \phi_{ijk} \left(x_1^{(i)} x_2^{(j)}\right)^{(k)}, \tag{2}$$

where $P_\phi : \text{Cl}(V, \mathfrak{q}) \times \text{Cl}(V, \mathfrak{q}) \to \text{Cl}(V, \mathfrak{q})$. This means that we get $(n+1)^3$ parameters for every geometric product between a pair of multivectors[1]. Parameterizing and computing all second-order terms amounts to ℓ^2 such operations, which can be computationally expensive given a reasonable number of channels ℓ. Instead, we first apply a linear map to obtain $y_1, \ldots, y_\ell \in \text{Cl}(V, \mathfrak{q})$. Through this map, the mixing (i.e., the terms that will get multiplied) gets learned. That is, we only get ℓ pairs $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ from which we then compute $z^{(k)}_{c_{\text{out}}} := P_{\phi_{c_{\text{out}}}}(x_{c_{\text{in}}}, y_{c_{\text{in}}})^{(k)}$. Note that here we have $c_{\text{in}} = c_{\text{out}}$, i.e., the number of channels does not change. Hence, we refer to this layer as the *element-wise* geometric product layer. We can obtain a more expressive (yet more expensive) parameterization by linearly combining such products by computing

$$z^{(k)}_{c_{\text{out}}} := T^{\text{prod}}_{\phi_{c_{\text{out}}}}(x_1, \ldots, x_\ell, y_1, \ldots, y_\ell)^{(k)} := \sum_{c_{\text{in}}=1}^{\ell} P_{\phi_{c_{\text{out}} c_{\text{in}}}}(x_{c_{\text{in}}}, y_{c_{\text{in}}})^{(k)}, \tag{3}$$

which we call the *fully-connected* geometric product layer. Computational feasibility and experimental verification should determine which parameterization is preferred.

Normalization and Nonlinearities Since our layers involve quadratic and potentially higher-order interaction terms, we need to ensure numerical stability. In order to do so, we use a normalization operating on each multivector subspace before computing geometric products by putting

$$x^{(m)} \mapsto \frac{x^{(m)}}{\sigma(a_m)\,\left(\bar{\mathfrak{q}}(x^{(m)}) - 1\right) + 1}, \tag{4}$$

where $x^{(m)} \in \text{Cl}^{(m)}(V, \mathfrak{q})$. Here, σ denotes the logistic sigmoid function, and $a_m \in \mathbb{R}$ is a learned scalar. The denominator interpolates between 1 and the quadratic form $\bar{\mathfrak{q}}\left(x^{(m)}\right)$, normalizing the magnitude of $x^{(m)}$. This ensures that the geometric products do not cause numerical instabilities without losing information about the magnitude of $x^{(m)}$, where a learned scalar interpolates between both regimes. Note that by $\bar{\mathfrak{q}}(x^{(m)})$ is invariant under the action of the Clifford group, rendering Equation (4) an equivariant map.

Next, we use the layer-wide normalization scheme proposed by [43], which, since it is also based on the extended quadratic form, is also equivariant with respect to the twisted conjugation.

Regarding nonlinearities, we use a slightly adjusted version of the units proposed by [43]. Since the scalar subspace $\text{Cl}^{(0)}(V, \mathfrak{q})$ is always invariant with respect to the twisted conjugation, we can apply $x^{(m)} \mapsto \text{ReLU}\left(x^{(m)}\right)$ when $m = 0$ and $x^{(m)} \mapsto \sigma_\phi\left(\bar{\mathfrak{q}}\left(x^{(m)}\right)\right) x^{(m)}$ otherwise. We can replace ReLU with any common scalar activation function. Here, σ_ϕ represents a potentially parameterized nonlinear function. Usually, however, we restrict it to be the sigmoid function. Since we modify $x^{(m)}$ with an invariant scalar quantity, we retain equivariance. Such gating activations are commonly used in the equivariance literature [54, 24].

[1] In practice, we use fewer parameters due to the sparsity of the geometric product, implying that many interactions will invariably be zero, thereby making their parameterization redundant.

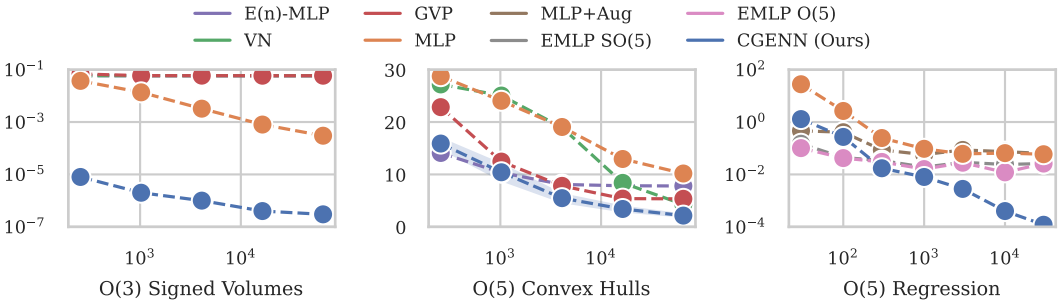

Figure 2: Left: Test mean-squared errors on the $\mathrm{O}(3)$ signed volume task as functions of the number of training data. Note that due to identical performance, some baselines are not clearly visible. Right: same, but for the $\mathrm{O}(5)$ convex hull task.

Figure 3: Test mean-squared-errors on the $\mathrm{O}(5)$ regression task.

Embedding Data in the Clifford Algebra In this work, we consider data only from the vector space V or the scalars $\mathbb{R}$, although generally one might also encounter, e.g., *bivector* data. That is, we have some scalar features $h_1, \ldots, h_k \in \mathbb{R}$ and some vector features $h_{k+1}, \ldots, h_\ell \in V$. Typical examples of scalar features include properties like mass, charge, temperature, and so on. Additionally, one-hot encoded categorical features are also included because $\{0, 1\} \subset \mathbb{R}$ and they also transform trivially. Vector features include positions, velocities, and the like. Then, using the identifications $\mathrm{Cl}^{(0)}(V, \mathfrak{q}) \cong \mathbb{R}$ and $\mathrm{Cl}^{(1)}(V, \mathfrak{q}) \cong V$, we can embed the data into the scalar and vector subspaces of the Clifford algebra to obtain Clifford features $x_1, \ldots, x_\ell \in \mathrm{Cl}(V, \mathfrak{q})$.

Similarly, we can predict scalar- or vector-valued data as output of our model by grade-projecting onto the scalar or vector parts, respectively. We can then directly compare these quantities with ground-truth vector- or scalar-valued data through a loss function and use standard automatic differentiation techniques to optimize the model. Note that *invariant* predictions are obtained by predicting scalar quantities.

3 Experiments

Here, we show that CGENNs excel across tasks, attaining top performance in several unique contexts. Parameter budgets as well as training setups are kept as similar as possible to the baseline references. All further experimental details can be found in the public code release.

3.1 Estimating Volumetric Quantities

$\mathrm{O}(3)$ **Experiment: Signed Volumes** This task highlights the fact that equivariant architectures based on scalarization are not able to extract some essential geometric properties from input data. In a synthetic setting, we simulate a dataset consisting of random three-dimensional tetrahedra. A main advantage of our method is that it can extract covariant quantities including (among others) *signed volumes*, which we demonstrate in this task. Signed volumes are geometrically significant because they capture the orientation of geometric objects in multidimensional spaces. For instance, in computer graphics, they can determine whether a 3D object is facing towards or away from the camera, enabling proper rendering. The input to the network is the point cloud and the loss function is the mean-squared error between the signed volume and its true value. Note that we are predicting a *covariant* (as opposed to *invariant*) scalar quantity (also known as a *pseudoscalar*) under $\mathrm{O}(3)$ transformations using a positive-definite (Euclidean) metric. The results are displayed in the left part of Figure 2. We compare against a standard multilayer perceptron (MLP), an MLP version of the $\mathrm{E}(n)$-GNN [44] which uses neural networks to update positions with scalar multiplication, *Vector Neurons* (VN) [17], and *Geometric Vector Perceptrons* (GVP) [31]. We see that the scalarization methods fail to access the features necessary for this task, as evidenced by their test loss not improving even with more available data. The multilayer perceptron, although a universal approximator, lacks the correct inductive biases. Our model, however, has the correct inductive biases (e.g., the equivariance property) and can also access the signed volume. Note that we do not take the permutation invariance

of this task into account, as we are interested in comparing our standard feed-forward architectures against similar baselines.

$\mathrm{O}(5)$ **Experiment: Convex Hulls** We go a step further and consider a *five-dimensional* Euclidean space, showcasing our model's ability to generalize to high dimensions. We also make the experiment more challenging by including more points and estimating the volume of the convex hull generated by these points – a task that requires sophisticated algorithms in the classical case. Note that some points may live inside the hull and do not contribute to the volume. We use the same network architectures as before (but now embedded in a five-dimensional space) and present the results in Figure 2. We report the error bars for CGENNs, representing three times the standard deviation of the results of eight runs with varying seeds. Volume (unsigned) is an invariant quantity, enabling the baseline methods to approximate its value. However, we still see that CGENNs outperform the other methods, the only exception being the low-data regime of only 256 available data points. We attribute this to our method being slightly more flexible, making it slightly more prone to overfitting. To mitigate this issue, future work could explore regularization techniques or other methods to reduce overfitting in low-data scenarios.

3.2 $\mathrm{O}(5)$ Experiment: Regression

We compare against the methods presented by [20] who propose an $\mathrm{O}(5)$-invariant regression problem. The task is to estimate the function $f(x_1, x_2) := \sin(\|x_1\|) - \|x_2\|^3/2 + \frac{x_1^\top x_2}{\|x_1\|\|x_2\|}$, where the five-dimensional vectors x_1, x_2 are sampled from a standard Gaussian distribution in order to simulate train, test, and validation datasets. The results are shown in Figure 3. We used baselines from [20] including an MLP, MLP with augmentation (MLP+Aug), and the $\mathrm{O}(5)$- & $\mathrm{SO}(5)$-MLP architectures. We maintain the same number of parameters for all data regimes. For extremely small datasets (30 and 100 samples), we observe some overfitting tendencies that can be countered with regularization (e.g., weight decay) or using a smaller model. For higher data regimes (300 samples and onward), CGENNs start to significantly outperform the baselines.

3.3 $\mathrm{E}(3)$ Experiment: n-Body System

The n-body experiment [34] serves as a benchmark for assessing the performance of equivariant (graph) neural networks in simulating physical systems [28]. In this experiment, the dynamics of $n = 5$ charged particles in a three-dimensional space are simulated. Given the initial positions and velocities of these particles, the task is to accurately estimate their positions after 1 000 timesteps. To address this challenge, we construct a graph neural network (GNN) using the Clifford equivariant layers introduced in the previous section. We use a standard message-passing algorithm [25] where the message and update networks are CGENNs. So long as the message aggregator is equivariant, the end-to-end model also maintains equivariance. The input to the network consists of the mean-subtracted positions of the particles (to achieve translation invariance) and their velocities. The model's output is the estimated displacement, which is to the input to achieve translation-equivariant estimated target positions. We include the invariant charges as part of the input and their products as edge attributes. We compare against the steerable SE(3)-Transformers [22], Tensor Field Networks [49], and SEGNN [8]. Scalarization baselines include Radial Field [37] and EGNN [44]. Finally, NMP [25] is not an $\mathrm{E}(3)$-equivariant method. The number of parameters in our model is maintained similar to the EGNN and SEGNN baselines to ensure a fair comparison.

Method	MSE ($\downarrow$)
SE(3)-Tr.	0.0244
TFN	0.0155
NMP	0.0107
Radial Field	0.0104
EGNN	0.0070
SEGNN	0.0043
CGENN	$\mathbf{0.0039 \pm 0.0001}$

Table 1: Mean-squared error (MSE) on the n-body system experiment.

Results of our experiment are presented in Table 1, where we also present for CGENN three times the standard deviation of three identical runs with different seeds. Our approach clearly outperforms earlier methods and is significantly better than [8], thereby surpassing the baselines. This experiment again demonstrates the advantage of leveraging covariant information in addition to scalar quantities, as it allows for a more accurate representation of the underlying physics and leads to better predictions.

Model	Accuracy (↑)	AUC (↑)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.5$)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.3$)
ResNeXt [57]	0.936	0.9837	302	1147
P-CNN [13]	0.930	0.9803	201	759
PFN [38]	0.932	0.9819	247	888
ParticleNet [41]	0.940	0.9858	397	1615
EGNN [44]	0.922	0.9760	148	540
LGN [7]	0.929	0.9640	124	435
LorentzNet [26]	**0.942**	**0.9868**	**498**	**2195**
CGENN	**0.942**	**0.9869**	**500**	**2172**

Table 2: Performance comparison between our proposed method and alternative algorithms on the top tagging experiment. We present the accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), and background rejection $1/\epsilon_B$ and at signal efficiencies of $\epsilon_S = 0.3$ and $\epsilon_S = 0.5$.

3.4 $\mathrm{O}(1,3)$ Experiment: Top Tagging

Jet tagging in collider physics is a technique used to identify and categorize high-energy jets produced in particle collisions, as measured by, e.g., CERN's ATLAS detector [10]. By combining information from various parts of the detector, it is possible to trace back these jets' origins [36, 3]. The current experiment seeks to tag jets arising from the heaviest particles of the standard model: the 'top quarks' [30]. A jet tag should be invariant with respect to the global reference frame, which can transform under *Lorentz boosts* due to the relativistic nature of the particles. A Lorentz boost is a transformation that relates the space and time coordinates of an event as seen from two inertial reference frames. The defining characteristic of these transformations is that they preserve the *Minkowski metric*, which is given by $\gamma(ct, x, y, z) := (ct)^2 - x^2 - y^2 - z^2$. Note the difference with the standard positive definite Euclidean metric, as used in the previous experiments. The set of all such transformations is captured by the orthogonal group $\mathrm{O}(1,3)$; therefore, our method is fully compatible with modeling this problem.

We evaluate our model on a top tagging benchmark published by [33]. It contains 1.2M training entries, 400k validation entries, and 400k testing entries. For each jet, the energy-momentum 4-vectors are available for up to 200 constituent particles, making this a much larger-scale experiment than the ones presented earlier. Again, we employ a standard message passing graph neural network [25] using CGENNs as message and update networks. The baselines include ResNeXt [57], P-CNN [13], PFN [38], ParticleNet [41], LGN [7], EGNN [44], and the more recent LorentzNet [26]. Among these, LGN is a steerable method, whereas EGNN and LorentzNet are scalarization methods. The other methods are not Lorentz-equivariant. Among the performance metrics, there are classification accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), and the background rejection rate $1/\epsilon_B$ at signal efficiencies of $\epsilon_S = 0.3$ and $\epsilon_S = 0.5$, where ϵ_B and ϵ_S are the false positive and true positive rates, respectively. We observe that LorentzNet, a method that uses invariant quantities, is an extremely competitive baseline that was optimized for this task. Despite this, CGENNs are able to match its performance while maintaining the same core implementation.

4 Conclusion

We presented a novel approach for constructing $\mathrm{O}(n)$- and $\mathrm{E}(n)$-equivariant neural networks based on Clifford algebras. After establishing the required theoretical results, we proposed parameterizations of nonlinear multivector-valued maps that exhibit versatility and applicability across scenarios varying in dimension. This was achieved by the core insight that polynomials in multivectors are $\mathrm{O}(n)$-equivariant functions. Theoretical results were empirically substantiated in three distinct experiments, outperforming or matching baselines that were sometimes specifically designed for these tasks.

CGENNs induce a (non-prohibitive) degree of computational overhead similar to other steerable methods. On the plus side, we believe that improved code implementations such as custom GPU kernels or alternative parameterizations to the current ones can significantly alleviate this issue, potentially also resulting in improved performances on benchmark datasets. This work provides solid theoretical and experimental foundations for such developments.

References

[1] Arne Alex, Matthias Kalus, Alan Huckleberry, and Jan von Delft, *A numerical algorithm for the explicit calculation of su (n) and sl (n, c) clebsch–gordan coefficients*, Journal of Mathematical Physics **52** (2011), no. 2, 023507.

[2] Brandon M. Anderson, Truong-Son Hy, and Risi Kondor, *Cormorant: Covariant Molecular Neural Networks.*, Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 14510–14519.

[3] ATLAS, *Jet energy scale measurements and their systematic uncertainties in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, arXiv preprint arXiv:1703.09665, 2017.

[4] Simon Axelrod and Rafael Gómez-Bombarelli, *Geom, energy-annotated molecular conformations for property prediction and molecular generation*, Scientific Data, Springer Science and Business Media LLC, 2022.

[5] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky, *E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials*, Nature Communications, 2022.

[6] Erik J Bekkers, *B-spline cnns on lie groups*, arXiv preprint arXiv:1909.12057, 2019.

[7] Alexander Bogatskiy, Brandon M. Anderson, Jan T. Offermann, Marwah Roussi, David W. Miller, and Risi Kondor, *Lorentz Group Equivariant Neural Network for Particle Physics*, International Conference on Machine Learning (ICML), 2020, pp. 992–1002.

[8] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling, *Geometric and Physical Quantities improve E(3) Equivariant Message Passing.*, International Conference on Learning Representations (ICLR), 2022.

[9] M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velivckovi'c, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, arXiv, 2021.

[10] JM Butterworth, J Thion, U Bratzler, PN Ratoff, RB Nickerson, JM Seixas, I Grabowska-Bold, F Meisel, S Lokwitz, et al., *The atlas experiment at the cern large hadron collider*, Jinst **3** (2008), S08003.

[11] Gabriele Cesa, Leon Lang, and Maurice Weiler, *A Program to Build E(N)-Equivariant Steerable CNNs.*, International Conference on Learning Representations (ICLR), 2022.

[12] Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller, *Machine learning of accurate energy-conserving molecular force fields*, Science Advances, American Association for the Advancement of Science (AAAS), 2017.

[13] CMS, *Boosted jet identification using particle candidates and deep neural networks*, Detector Performance Figures: CMS-DP-17-049, 2017.

[14] Taco Cohen and Max Welling, *Group Equivariant Convolutional Networks.*, International Conference on Machine Learning (ICML), 2016, pp. 2990–2999.

[15] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling, *Spherical CNNs.*, International Conference on Learning Representations (ICLR), 2018.

[16] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger, *Spherenet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images.*, European Conference on Computer Vision (ECCV), Springer International Publishing, 2018, pp. 525–541.

[17] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas, *Vector Neurons: A General Framework for SO(3)-Equivariant Networks.*, IEEE International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 12180–12189.

[18] Carlos Esteves, *Theoretical aspects of group equivariant neural networks*, arXiv, 2020.

[19] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson, *Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data.*, International Conference on Machine Learning (ICML), 2020, pp. 3165–3176.

[20] Marc Finzi, Max Welling, and Andrew Gordon Wilson, *A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups*, International Conference on Machine Learning, PMLR, 2021, pp. 3318–3328.

[21] William T Freeman, Edward H Adelson, et al., *The design and use of steerable filters*, IEEE Transactions on Pattern analysis and machine intelligence **13** (1991), no. 9, 891–906.

[22] Fabian Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling, *Se(3)-Transformers: 3d Roto-Translation Equivariant Attention Networks.*, Conference on Neural Information Processing Systems (NeurIPS), 2020.

[23] Johannes Gasteiger, Florian Becker, and Stephan Günnemann, *Gemnet: Universal Directional Graph Neural Networks for Molecules.*, Conference on Neural Information Processing Systems (NeurIPS), 2021, pp. 6790–6802.

[24] Mario Geiger and Tess Smidt, *e3nn: Euclidean neural networks*, 2022.

[25] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, *Neural message passing for quantum chemistry*, International conference on machine learning, PMLR, 2017, pp. 1263–1272.

[26] Shiqi Gong, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu, *An efficient Lorentz equivariant graph neural network for jet tagging*, Journal of High Energy Physics, Springer Science and Business Media LLC, 2022.

[27] Richard Gowers, Max Linke, Jonathan Barnoud, Tyler Reddy, Manuel Melo, Sean Seyler, Jan Domański, David Dotson, Sébastien Buchoux, Ian Kenney, and Oliver Beckstein, *Mdanalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations*, Proceedings of the Python in Science Conference, SciPy, 2016.

[28] Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang, *Geometrically equivariant graph neural networks: A survey*, arXiv, 2022.

[29] Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang, *Equivariant Graph Mechanics Networks with Constraints.*, International Conference on Learning Representations (ICLR), 2022.

[30] Joseph R Incandela, Arnulf Quadt, Wolfgang Wagner, and Daniel Wicke, *Status and prospects of top-quark physics*, Progress in Particle and Nuclear Physics **63** (2009), no. 2, 239–292.

[31] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron O. Dror, *Learning from Protein Structure with Geometric Vector Perceptrons.*, International Conference on Learning Representations (ICLR), 2021.

[32] LM Kaplan and M Resnikoff, *Matrix products and the explicit 3, 6, 9, and 12-j coefficients of the regular representation of su (n)*, Journal of Mathematical Physics **8** (1967), no. 11, 2194–2205.

[33] Gregor Kasieczka, Tilman Plehn, Jennifer Thompson, and Michael Russel, *Top quark tagging reference dataset*, March 2019.

[34] Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard S. Zemel, *Neural Relational Inference for Interacting Systems.*, International Conference on Machine Learning (ICML), 2018, pp. 2693–2702.

[35] Johannes Klicpera, Janek Groß, and Stephan Günnemann, *Directional Message Passing for Molecular Graphs.*, International Conference on Learning Representations (ICLR), 2020.

[36] Roman Kogler, Benjamin Nachman, Alexander Schmidt, Lily Asquith, Emma Winkels, Mario Campanelli, Chris Delitzsch, Philip Harris, Andreas Hinzmann, Deepak Kar, et al., *Jet substructure at the large hadron collider*, Reviews of Modern Physics **91** (2019), no. 4, 045003.

[37] Jonas Köhler, Leon Klein, and Frank Noé, *Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities.*, International Conference on Machine Learning (ICML), 2020, pp. 5361–5370.

[38] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler, *Energy flow networks: deep sets for particle jets*, Journal of High Energy Physics, Springer Science and Business Media LLC, 2019.

[39] Risi Kondor and Shubhendu Trivedi, *On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups.*, International Conference on Machine Learning (ICML), 2018, pp. 2752–2760.

[40] Reiner Lenz, *Group theoretical methods in image processing*, vol. 413, Springer, 1990.

[41] Huilin Qu and Loukas Gouskos, *Jet tagging via particle clouds*, Physical Review D, American Physical Society (APS), 2020.

[42] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld, *Quantum chemistry structures and properties of 134 kilo molecules*, Scientific Data, Springer Science and Business Media LLC, 2014.

[43] David Ruhe, Jayesh K Gupta, Steven de Keninck, Max Welling, and Johannes Brandstetter, *Geometric clifford algebra networks*, arXiv preprint arXiv:2302.06594, 2023.

[44] Victor Garcia Satorras, E. Hoogeboom, and M. Welling, *E(n) Equivariant Graph Neural Networks*, International Conference on Machine Learning, 2021.

[45] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, *The graph neural network model*, IEEE transactions on neural networks **20** (2008), no. 1, 61–80.

[46] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, *Schnet – A deep learning architecture for molecules and materials*, The Journal of Chemical Physics **148** (2018), no. 24, 241722.

[47] Kristof Schütt, Oliver T. Unke, and Michael Gastegger, *Equivariant message passing for the prediction of tensorial properties and molecular spectra.*, International Conference on Machine Learning (ICML), 2021, pp. 9377–9388.

[48] Philipp Thölke and G. D. Fabritiis, *Torch MD-NET: Equivariant Transformers for Neural Network based Molecular Potentials*, ArXiv, vol. abs/2202.02541, 2022.

[49] Nathaniel Thomas, T. Smidt, S. Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick F. Riley, *Tensor Field Networks: Rotation- and Translation-Equivariant Neural Networks for 3d Point Clouds*, arXiv, 2018.

[50] Raphael J. L. Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon M. Anderson, Stephan Eismann, Risi Kondor, Russ B. Altman, and Ron O. Dror, *Atom3d: Tasks on Molecules in Three Dimensions.*, Conference on Neural Information Processing Systems (NeurIPS), 2021.

[51] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung, *Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data.*, IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1588–1597.

[52] Maurice Weiler and Gabriele Cesa, *General E(2)-Equivariant Steerable CNNs.*, Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 14334–14345.

[53] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling, *Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds*, arXiv preprint arXiv:2106.06020, 2021.

[54] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen, *3d Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data.*, Conference on Neural Information Processing Systems (NeurIPS), 2018, pp. 10402–10413.

[55] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow, *Harmonic Networks: Deep Translation and Rotation Equivariance.*, Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7168–7177.

[56] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, *3d ShapeNets: A deep representation for volumetric shapes.*, Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920.

[57] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, *Aggregated Residual Transformations for Deep Neural Networks.*, Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5987–5995.

[58] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola, *Deep sets*, Advances in neural information processing systems, vol. 30, 2017.

[59] C Lawrence Zitnick, Lowik Chanussot, Abhishek Das, Siddharth Goyal, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Thibaut Lavril, Aini Palizhati, Morgane Riviere, and others, *An introduction to electrocatalyst design using machine learning for renewable energy storage*, arXiv, 2020.