Representation Learning on Biomolecular Structures using Equivariant Graph Attention

Tuan Le
Bayer AG
Freie Universität Berlin
tuan.le2@bayer.com

Frank Noé
Microsoft Research AI4Science
Freie Universität Berlin
franknoe@microsoft.com

Djork-Arné Clevert[*]
Bayer AG
djork-arne.clevert@pfizer.com

Abstract

Learning and reasoning about 3D molecular structures with varying size is an emerging and important challenge in machine learning and especially in the development of biotherapeutics. Equivariant Graph Neural Networks (GNNs) can simultaneously leverage the geometric and relational detail of the problem domain and are known to learn expressive representations through the propagation of information between nodes leveraging geometrical details, such as directionality in their intermediate layers. In this work, we propose an equivariant GNN that operates with Cartesian coordinates to incorporate directionality and implements a novel attention mechanism, acting as a content and spatial dependent filter. Our proposed message function processes vector features in a geometrically meaningful way by mixing existing vectors and creating new ones based on cross products.

1 Introduction

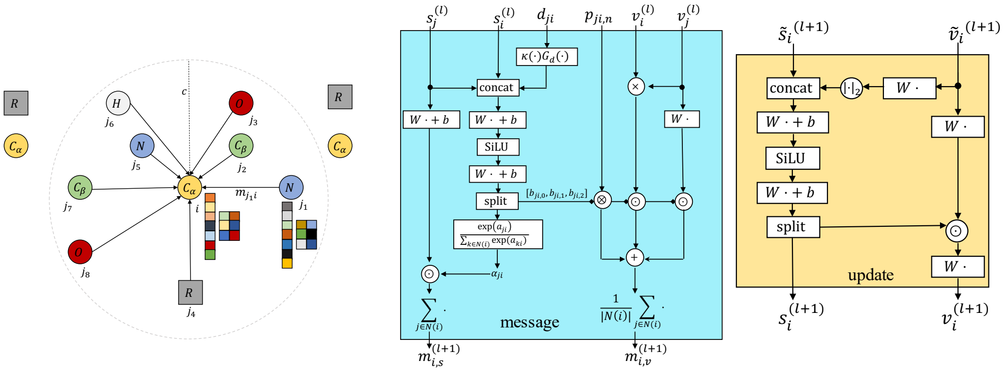
Predicting molecular properties is of central importance to applications in pharmaceutical research and protein design and accurate computational methods can significantly accelerate the process of finding better molecular candidates in a faster and cost-efficient way. While Deep Learning (DL) has replaced hand-crafted features to a large extent, many advances are crucially determined through inductive biases in deep neural networks, e.g. by exploiting the *symmetry* of the data by constraining *equivariance* with respect to transformations from a certain symmetry group [1, 2].

3D Graph Neural Networks have been applied on a widespread of molecular structures, such as in the prediction of quantum chemistry properties of small molecules [3, 4] but also on macromolecular structures like proteins [5–8] due to the natural representation of structures as graphs, with atoms as nodes and edges drawn based on bonding or spatial proximity. These networks generally encode the 3D geometry in terms of rotationally invariant representations, such as pairwise distances when modelling local interactions which leads to a loss of directional information, while the addition of angular information into network architecture has shown to be beneficial [9–11] in performance.

In this work, we introduce Equivariant Graph Attention Networks (EQGAT) that operates on large point clouds such as proteins or protein-ligand complexes and show its superior performance compared to invariant models as well as our proposed model‘s faster training time compared to recent architectures that achieve equivariance through the usage of irreducible representations. Our model

[*] Work was done during time at Bayer AG.

Machine Learning for Structural Biology Workshop, NeurIPS 2022.



(a) Propagation flow for central node i. (b) Proposed equivariant message $M_l(\cdot)$ and update function $U_l(\cdot)$.

Figure 1: (a) Visualization of the local neighbourhood of central carbon atom i. Directed edges illustrate the message flow, where scalar and vector features are propagated along the edges. Grey boxes R represent the side-chain atoms of each residue and serve here as visual compression. (b) Proposed equivariant message function that computes a geometric and content related feature attention filter for scalar features, while vector messages are created based on a weighted combination of newly constructed vectors. The update function fuses scalar and vector features into a new representation.

implements a novel feature attention mechanism which is invariant to global rotations and translations of inputs and includes spatial- but also content related information which serves as powerful edge embedding when propagating information in the Message Passing Neural Networks (MPNNs) [4] framework. Since we define equivariant functions on the original Cartesian space while restricting ourselves to tensor representations of rank 1, i.e., vectors, we aim to capture as most geometrical information as possible through a geometrically motivated message function.

2 Background

2.1 Message Passing Neural Networks (MPNNs)

MPNNs [4] generalize Graph Neural Networks (GNNs) [1, 2, 12] and aim to parameterize a mapping from a graph to a feature space, where the feature space is either defined on the node- or graph level. Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains nodes $i \in \mathcal{V}$ and edges $(j, i) \in \mathcal{E}$ which represent the relationship between nodes j and i. Since MPNNs utilize shared layers among nodes, permutation equivariance is preserved. In this work, we consider graphs representing molecular systems embedded in 3D Euclidean space, where atoms represent nodes and the edges are described through covalent bonds and/or by atom pairs within a certain cutoff distance c as illustrated in Figure 1(a).
We refer $x_i^{(l)} = (a_i, p_i, s_i^{(l)}, v_i^{(l)})$ to the state of the $i-$th atom, where $a_i \in \mathbb{Z}_+$ and $p_i \in \mathbb{R}^3$ denote atom i‘s chemical element and its spatial position, while $h_i^{(l)} = (s_i^{(l)}, v_i^{(l)}) \in \mathbb{R}^{1\times F_s} \times \mathbb{R}^{3\times F_v}$ are the hidden scalar and vector features that are iteratively refined through L message passing steps. A general MPNN implements a learnable *message* and *update* function denoted as $M_l(\cdot)$ and $U_l(\cdot)$ to process atom $i-$th‘s hidden feature by considering its local environment $\mathcal{N}(i)$ through

$$m_i^{(l+1)} = \sum_{j\in\mathcal{N}(i)} M_l(x_i^{(l)}, x_j^{(l)}), \text{ and } x_i^{(l+1)} = (a_i, p_i, U_l(x_i^{(l)}, m_i^{(l+1)})),$$

where $\mathcal{N}(i) = \{j : ||p_{ij}||_2 = ||p_j - p_i||_2 = d_{ij} < c\}$ denotes atom‘s $i-$th neighbour set. For our 3D GNN, we aim to implement simple, yet powerful rotation equivariant transformations in the message and update functions, to accurately describe the local environment of atoms in the radius graph.

3 Architecture

We implement a non-linear edge filter that depends on content related information stored in the scalar features (s_j, s_i) and a radial basis expansion of the Euclidean distance $d_{ji} \leq c$. We choose the Bessel basis $G_d : \mathbb{R} \to \mathbb{R}^K$ as introduced in [9] and their polynomial envelope function κ. The computation of the attention edge-filter is obtained through

$$\begin{aligned} e_{ji}^{(l+1)} &= [s_i^{(l)}||s_j^{(l)}||\kappa(d_{ji})G_d(d_{ji})] \in \mathbb{R}^{2F_s+K} \\ f_{ji}^{(l+1)} &= \text{MLP}(e_{ji}^{(l+1)}) \in \mathbb{R}^{F_s+3F_v}, \end{aligned} \tag{1}$$

The input to the Multilayer-Perceptron (MLP) is a concatenation of scalar features as well as a by κ scaled radial basis expansion of the distance between nodes j and i. The $\mathrm{SO}(3)$-invariant embedding $f_{ji}^{(l+1)}$ represents the $F_s + 3F_v$ attention logits which are further split into $f_{ji}^{(l+1)} = [a_{ji}, b_{ji}]^{(l+1)}$. The feature attention for the scalar embeddings is computed using the standard softmax function

$$\alpha_{ji} = \frac{\exp(a_{ji})}{\sum_{k\in\mathcal{N}(i)} \exp(a_{ki})} \in (0,1)^{F_s},$$

where the normalization in the denominator runs over all neighbours j' and the exponential function is applied componentwise. The embedding $b_{ji} \in \mathbb{R}^{3F_v}$ is processed to create coefficients that serve as weights for a linear combination of vector quantities to compute the vector message from j to i, which we will describe in the following.

We follow the idea of standard convolution, which is a linear transformation of the input, and compute the scalar features message for central node i as

$$m_{i,s}^{(l+1)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ji}^{(l+1)} \odot W_s^{(l+1)} s_j^{(l)}, \tag{2}$$

where $W_s^{(l+1)} \in \mathbb{R}^{F_s\times F_s}$ is a trainable weight matrix shared among all nodes and $\alpha_{ji}^{(l+1)}$ the non-linear attention filter. Our proposed message function for scalar features in Eq. (2) can be formulated as a linear transformation where the weight matrix depends on distances but also hidden scalar information. To see this, we rewrite $\alpha_{ji}^{(l+1)} \in (0,1)^{F_s}$ as matrix using the diagonal operator $A_{ji}^{(l+1)} = \text{diag}(\alpha_{ji}^{(l+1)}) \in (0,1)^{F_s\times F_s}$ and observe that the filter scales the (independent) weight matrix $W_n^{(l+1)}$ leading to the message propagation

$$m_{i,s}^{(l+1)} = \sum_{j\in\mathcal{N}(i)} A_{ji}^{(l+1)} W_s^{(l+1)} s_j^{(l)} = \sum_{j\in\mathcal{N}(i)} W_{ji}^{(l+1)} s_j^{(l)},$$

where $W_{ji}^{(l+1)}$ defines the linear transformation matrix whose content depends on $\mathrm{SO}(3)$-invariant information which can however still be interpreted as non-linear convolution because the $A_{ji}^{(l+1)}$ weight matrix is obtained through an MLP and softmax activation function.

Building Equivariant Features In our work, we initialize the initial vector features as zero tensor while equivariant features are obtained by utilizing normalized relative positions $p_{ji,n}$ in the first layer, to compute the interaction between central node i and its neighbour j. In the subsequent layers, we extend the set of vectors by (I) constructing vectors based on normalized relative positions again, (II) mixing existing vector channels from the previous iteration and (III) creating new vector quantities by making use of the cross product.
(I) Utilizing normalized relative positions: we create equivariant vector features based on normalized relative position $p_{ji,n} = \frac{1}{d_{ji}}(p_i - p_j)$ as those provide directional information. Equivariant interactions between node j and i are computed through

$$v_{ji,0}^{(l+1)} = p_{ji,n} \otimes b_{ji,0}^{(l+1)} = p_{ji,n} b_{ji,0}^{(l+1)\top} \in \mathbb{R}^{3\times F_v}, \tag{3}$$

which preserves $\mathrm{SO}(3)$ equivariance, due to the linearity of the tensor product. We note that the creation of 'initial' equivariant features in such manner is also performed in architectures, like [13–16] just to name a few, that make use of irreducible representations of the $\mathrm{SO}(3)$ group by means of the spherical harmonics and implement the Clebsch-Gordan tensor product ($\otimes_{cg}$) that allows the mixing of possibly higher-order embedding representations of type $l > 1$. The $l = 1$ representation in Eq. (3) can be interpreted as F_v scaled versions of the relative position $p_{ji,n}$.
(II) In similar fashion to the (independent) linear transformation of scalar channels, we mix the vector channels via $v_n^{(l+1)} = v^{(l)} W_v^{(l+1)}$ using a weight matrix $W_v^{(l)} \in \mathbb{R}^{F_v\times F_v}$ which preserves

SO(3) equivariance due to the linearity property and is shared among all nodes. For a particular neighbouring node j, we scale the linearly transformed vectors

$$v_{ji,1}^{(l+1)} = b_{ji,1}^{(l+1)} \odot v_{n,j}^{(l+1)}. \tag{4}$$

(III) To capture more geometric information, while restricting the representation to be of type $l = 1$, we utilize the vector cross product between hidden vector features. The output of the cross product $c = a \times b$ returns a vector c that is perpendicular to plane spanned by a and b and in our network architecture, we utilize this by computing the cross product on same channels from the previous layers' vector features of node i and j as

$$\tilde{v}_{ji,2}^{(l+1)} = (v_i^{(l)} \times v_j^{(l)}) \in \mathbb{R}^{3 \times F_v}.$$

We highlight that recent equivariant GNNs operating on the original Cartesian space, such as GVP [17], PaiNN [18] or ET-Transformer [19] do not include the cross product in their architecture and are restricted in the creation of vector features that may span the entire $\mathbb{R}^3$. These architectures make use of step (I) and (II) only. For example, when all atoms are placed on the xy-plane, using these steps would always create vectors on the xy plane, while the coordinate on z axis is always 0. By leveraging the cross product, vectors in the z direction can be computed, without additional overhead. In similar fashion to Eq. (3) and (4), each channel of the representation $\tilde{v}_{ji,2}^{(l)}$ is weighted by the SO(3) non-linear filter $b_{ji,2}^{(l)} \in \mathbb{R}^{F_v}$ to obtain

$$v_{ji,2}^{(l+1)} = b_{ji,2}^{(l+1)} \odot \tilde{v}_{ji,2}^{(l+1)}. \tag{5}$$

Finally, we define the vector message from node j to central node i as the sum of the three components in (3) to (5) and aggregate it across all neighbouring nodes $j \in \mathcal{N}(i)$ to obtain the vector message

$$m_{i,v}^{(l+1)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} (v_{ji,0}^{(l+1)} + v_{ji,1}^{(l+1)} + v_{ji,2}^{(l+1)}), \tag{6}$$

which results into new weighted geometric vectors. After obtaining the aggregated message for central node i in the representation $m_i^{(l+1)} \in \mathbb{R}^{F_s} \times \mathbb{R}^{3 \times F_v}$, we deploy a residual connection as intermediate update step

$$\tilde{s_i}^{(l+1)} = s_i^{(l)} + m_{i,s}^{(l+1)}, \text{ and } \tilde{v_i}^{(l+1)} = v_i^{(l)} + m_{i,v}^{(l+1)},$$

while in the update layer, we implement an equivariant non-linear transformation inspired by gated non-linearities proposed by [20] and used in [18] with minor modification as shown in Figure 1(b).

4 Experiments

The ATOM3D benchmark [21] provides datasets for representation learning on atomic-level 3D molecular structures of different kinds, i.e., proteins, RNAs, small molecules and complexes. Since proteins perform specific biological functions essential for all living organisms and hence, play a key role when investigating the most fundamental questions in the life sciences, we focus our experiments on the learning problems often encountered in structural biology with different difficulties due to data scarcity and varying structural sizes. We use provided training, validation and test splits from ATOM3D and refer the interested reader to the original work of Townshend et al. [21] for more details. For all benchmarks, we compare against the Baseline CNN and GNN models provided by Townshend et al. [21] from ATOM3D, GVP-GNN reported in [22] and we run experiments for SchNet [3], an SO(3) invariant GNN architecture, PaiNN [18] as SchNet‘s improved SO(3) equivariant successor and the recently proposed SEGNN [16] that leverages higher-order representations using their official code base. For SchNet, PaiNN and our proposed EQGAT architecture, we implement a 5-layer GNN with $F_s = 100$ scalar channels and $F_v = 16$ vector channels for the PSR, RSR, RES and PPI benchmark, as these consists of more training samples and comprise larger biomolecules. For the Ligand Binding Affinity (LBA) task, we utilize a 3-layer GNN with the same number of scalar- and vector channels. For the SEGNN architecture, we implement a 3-layer GNN with $(100, 16, 8)$ channels for the embeddings of type $l = (0, 1, 2)$ that transform according to the irreducible representation of that order. The edges in the point clouds are constructed based on a radius cutoff of 4.5Å. All graphs are considered as full-atom graphs, i.e., the initial node feature is determined by the chemical element.

Table 1: Benchmark results on ATOM3D tasks. We run our own experiments with the SchNet, PaiNN, SEGNN and our EQGAT model and report averaged metrics over 3 runs except for SEGNN using a single run only and the RES dataset.

Tasks	PSR (↑)		RSR (↑)		LBA (↓)	RES (↑)	PPI (↑)
Metric	Mean R_S	Global R_S	Mean R_S	Global R_S	RMSE	Accuracy	ROCAUC
CNN	0.431 ± 0.013	0.789 ± 0.017	0.264 ± 0.046	0.372 ± 0.027	$\mathbf{1.416 \pm 0.021}$	0.451 ± 0.002	0.844 ± 0.002
GNN	$\mathbf{0.515 \pm 0.010}$	0.755 ± 0.004	0.234 ± 0.006	$\mathbf{0.512 \pm 0.049}$	1.570 ± 0.025	0.082 ± 0.002	0.669 ± 0.001
GVP-GNN	0.511 ± 0.010	0.845 ± 0.008	0.211 ± 0.142	0.330 ± 0.054	1.594 ± 0.073	0.527 ± 0.003	0.866 ± 0.004
SchNet	0.448 ± 0.016	0.784 ± 0.013	0.247 ± 0.029	0.273 ± 0.017	1.522 ± 0.015	0.326 ± 0.003	0.839 ± 0.005
PaiNN	0.462 ± 0.015	0.809 ± 0.003	0.270 ± 0.062	0.462 ± 0.064	1.507 ± 0.033	0.370 ± 0.004	0.884 ± 0.002
SEGNN	0.474	0.833	-0.099	0.252	1.450 ± 0.011	0.454	0.854
EQGAT	0.491 ± 0.008	$\mathbf{0.847 \pm 0.006}$	$\mathbf{0.316 \pm 0.029}$	0.404 ± 0.096	1.440 ± 0.027	$\mathbf{0.540 \pm 0.017}$	$\mathbf{0.908 \pm 0.001}$

The Protein and RNA Structure Ranking tasks (PSR / RSR) in ATOM3D are both regression tasks with the objective to predict the quality score in terms of *Global Distance Test* (GDT_TS) or Root-Mean-Square Deviation (RMSD) for generated Protein and RNA models wrt. to its experimentally determined ground-truth structure. We evaluated our model on the biopolymer ranking and obtained good results on the current benchmark, as reported in Table 1 in terms of Spearman rank correlation. Our proposed model performs particularly well on the PSR task outperforming the GVP-GNN [22] on the Global Rank Spearman correlation on the test set, while our model is more parameter efficient (383K vs. 640K). We believe our model could be further improved by additional hyperparameter tuning, e.g., by increasing the number of scalar or vector channels, which we did not do in our study to compare against the baseline models. We noticed that the RSR benchmark was particularly difficult to validate as only a few dozen experimentally determined RNA structures are existent to date, and the structural models generated in the ATOM3D framework are labeled with the RMSD to its native structure, which is known to be sensitive to outlier regions, for exampling by inadequate modelling of loop regions [23], while the GDT_TS metric might be a better suited target to predict a ranking for generated RNA structures as in the PSR benchmark.

We use the ligand binding affinity (LBA) dataset and found that among the GNN architectures, our proposed model obtains the best results, while also being computationally cheap and fast to train. The best performing model in the LBA-task is a 3D CNN model which works on the joint protein-ligand representation using voxel space and enforcing equivariance through data augmentation. The inferior performance of all equivariant GNNs might be caused by the need of larger filters to better capture the locality and many-body effects, where 3D CNNs have an advantage when using voxel representations, while GNNs commonly capture 2-body effects. Notably, our proposed EQGAT architecture performs on par with the SEGNN that implements geometric tensors of higher order, i.e., of rotation order $l = 2$, that transforms as a rank 2 Cartesian tensor. We believe that including the cross product in our vector message in (6) allows the model to capture more of the geometric detail in a possible protein ligand binding pose for accurately predicting the binding affinity.

Finally, we train our EQGAT model on the Residue (RES) and Protein-Protein-Interface (PPI) benchmarks that both examine if a model is able to capture the physico-chemical environment of a protein by predicting the amino acid identity of a protein site based on the surrounding of a structural environment (RES) or whether two selected amino acids will interact with each other (PPI).

5 Conclusion

In this work, we introduced a novel attention-based equivariant graph neural network for the prediction of properties of large biomolecules that achieves superior performance on the ATOM3D benchmark. Our proposed architecture makes use of rotationally equivariant features in their intermediate layers to faithfully represent the geometry of the data, while being computationally efficient, as all equivariant functions are directly implemented in the original Cartesian space without changing the representation through the spherical harmonics basis as commonly done in Tensorfield networks. As our proposed model operates on Cartesian tensors and we restrict the representation to be of rank 1 only, a general promising future direction of investigation is the implementation of Cartesian equivariant GNNs that leverage higher-rank tensors in their layers, that are specifically implemented for learning purposes involving large biomolecules. As it is up to date not clear, how much improvement higher-order Cartesian tensors benefit for learning tasks that involve large biomolecular systems, we hope that our work and open-source code will be useful for the graph learning and computational biology community.

Code Availability

We provide the implementation of our model and experiments on `https://github.com/Bayer-Group/eqgat`. We use PyTorch [24] as Deep Learning framework and PyTorch Geometric [25] to implement our GNNs.

Author Contributions

T.L designed the model architecture, carried out the implementation and executed all experiments. T.L wrote the manuscript with input from F.N and D.A.C who both proofread the final manuscript. D.A.C supervised the work.

Acknowledgements

This work was supported by several research grants. F.N acknowledges funding from the European Commission (ERC CoG 772230 'ScaleCell'), MATH+ (AA1-6), Math AA1-10 and Deutsche Forschungsgemeinschaft (CRC1114/B08). D.A.C acknowledged funding from the Bayer AG Life Science Collaboration ('DeepMinDS'). D.A.C also received financial support from European Commission grant numbers 963845 and 956832 under the Horizon2020 Framework Program for Research and Innovation. T.L acknowledges funding from Bayer AG's PhD scholarship.

References

[1] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL `http://arxiv.org/abs/1806.01261`.

[2] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

[3] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf`.

[4] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/gilmer17a.html`.

[5] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/f507783927f2ec2737ba40afbd17efb5-Paper.pdf`.

[6] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf`.

[7] Federico Baldassarre, David Menéndez Hurtado, Arne Elofsson, and Hossein Azizpour. GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3):360–366, 08 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa714. URL `https://doi.org/10.1093/bioinformatics/btaa714`.

[8] Pedro Hermosilla, Marco Schäfer, Matej Lang, Gloria Fackelmann, Pere-Pau Vázquez, Barbora Kozlikova, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=l0mSUROpwY`.

[9] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=B1eWbxStPH`.

[10] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6790–6802. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/35cf8659cfcb13224cbd47863a34fc58-Paper.pdf`.

[11] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=givsRXsOt9r`.

[12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[13] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018.

[14] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/03573b32b2746e6e8ca98b9123f2249b-Paper.pdf`.

[15] 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*, 13 (1):2453, 2022. doi: 10.1038/s41467-022-29939-5. URL `https://doi.org/10.1038/s41467-022-29939-5`.

[16] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=_xwr8gOBeV1`.

[17] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=1YLJDvSx6J4`.

[18] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9377–9388. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/schutt21a.html`.

[19] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=zNHzqZ9wrRB`.

[20] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/488e4104520c6aab692863cc1dba45af-Paper.pdf`.

[21] Raphael John Lamarre Townshend, Martin Vögele, Patricia Adriana Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon M. Anderson, Stephan Eismann, Risi Kondor, Russ Altman, and Ron O. Dror. ATOM3d: Tasks on molecules in three dimensions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL `https://openreview.net/forum?id=FkDZLpK1Ml2`.

[22] Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Equivariant graph neural networks for 3d macromolecular structure, 2021.

[23] Adam Zemla. Lga – a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31:3370–4, 08 2003. doi: 10.1093/nar/gkg571.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[25] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

[27] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9323–9332. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/satorras21a.html`.

A Appendix

Full Model Details and Hyperparameters

All EQGAT models in this paper were trained on a single Nvidia Tesla V100 GPU.

Table 2: Description of architectural parameters on the ATOM3D benchmarks.

Parameter	LBA	PSR	RSR	RES
Learning rate (lr.)	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Maximum epochs	20	30	30	40
Lr. patience	10	10	10	10
Lr. decay factor	0.75	0.75	0.75	0.75
Batch size	16	16	16	32
Num. layers	3	5	5	5
Num. RBFs	32	32	32	32
Cutoff [Å]	4.5	4.5	4.5	4.5
Scalar channels F_s	100	100	100	100
Vector channels F_v	16	16	16	16
Num. parameters	238k	383k	383k	386k

We used the ADAM optimizer [26] apart from the defined learning rate all other standard hyperparameter setting from the PyTorch library. We trained the models up to a user-defined maximum number of epochs and for testing, we loaded the checkpoint from the model with the best validation performance, to perform the test evaluation.

Proof Equivariance

A.1 Invariance and Equivariance

In this work, we consider the special orthogonal group SO(3), i.e. the group of proper rotations in three dimensions. A group element of SO(3) is commonly represented as matrix $R \in \mathbb{R}^{3\times 3}$ satisfying $R^\top R = RR^\top = I$ and $\det R = 1$.
For a node feature $h = (s, v) \in \mathbb{R}^{F_s} \times \mathbb{R}^{3\times F_v}$, an SO(3)-equivariant function $f(h) = h' = (s', v')$ must obey the following equation

$$f(g.h) = g.(s', v') = (Is', Rv') = (s', Rv') = g.f(h), \tag{7}$$

where $g.o$ in this work means, a group element g of SO(3) acting on the object o. As shown in (7), invariance can be regarded as special case of equivariance, where equivariance for a scalar representation means that the *trivial* representation, i.e. the identity, acts on the scalar embedding, while vectors are transformed with R, i.e., a change of basis is performed, where the new basis is determined by the columns in R.

We prove the rotation equivariance in Eq. (6) which consists of the sum of three vector components, and displayed here again

$$m_{i,v}^{(l+1)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j\in\mathcal{N}(i)} (v_{ji,0}^{(l+1)} + v_{ji,1}^{(l+1)} + v_{ji,2}^{(l+1)}).$$

As the sum is a linear function, we require to show that each summand $(v_{ji,0}, v_{ji,1}, v_{ji,2})$ is equivariant. For brevity, we omit all top indices. The first term is computed as tensor product of an $l = 1$ representation and $l = 0$ representation through

$$v_{ji,0} = p_{ji,n} \otimes b_{ji,0} = p_{ji,n} b_{ji,0}^\top \in \mathbb{R}^{3\times F_v},$$

where $b_{ji,0} \in \mathbb{R}^{F_v}$ is an SO(3)-invariant representation, i.e. a scalar representation with F_v channels, and $p_{ji,n} \in S_2 \subset \mathbb{R}^3$ a normalized relative vector, which lies on the 2-dimensional sphere.

If the point cloud is rotated, as defined in Eq. (7), (relative) position as well as vector features change to

$$p \xrightarrow{R} Rp \; ,$$
$$v \xrightarrow{R} Rv \; ,$$

while the cross product between two vector features v_0, v_1 is invariant to rotation, resulting to the property

$$(Rv_0 \times Rv_1) = R(v_0 \times v_1) \; .$$

In case a rotation is acting on the system, from Eq. (7) we know how vector and scalar quantities transform, resulting into:

$$R.v_{ji,0} \rightarrow Rp_{ji,n} \otimes b_{ji,0} = R(p_{ji,n} \otimes b_{ji,0}) = Rv_{ji,0}.$$

due to the linearity of the tensor product which proves SO(3) equivariance for the first term.
For the second term, we calculate

$$v_{ji,1} = b_{ji,1} \odot (v_i \times v_j),$$

where $b_{ji,1} \in \mathbb{R}^{F_v}$ is an SO(3)-invariant representation and the output of the cross product is a vector representation $\in \mathbb{R}^{3 \times F_v}$. To be precise, the elementwise multiplication from the left with the $b_{ji,1}$ has to be rewritten, to match the shape, i.e. unsqueeze a new dimension to scale each of the F_v vector by the scalar value, resulting into:

$$v_{ji,1} = (1 \otimes b_{ji,1}) \odot (v_i \times v_j),$$

where 1 is the one-vector in 3 dimensions. For a rotation acting on the system, we conclude that

$$\begin{aligned} R.v_{ji,1} &\rightarrow (1 \otimes b_{ji,1}) \odot (Rv_i \times Rv_j) \\ &= (1 \otimes b_{ji,1}) \odot R(v_i \times v_j) = R(1 \otimes b_{ji,1}) \odot (v_i \times v_j) \\ &= Rv_{ji,1}, \end{aligned}$$

which proves SO(3) equivariance for the second term.
The third term is obtained through

$$v_{ji,2} = (1 \otimes b_{ji,2}) \odot (v_j W_n),$$

where $b_{ji,2} \in \mathbb{R}^{F_v}$ is a scalar representation with F_v channels and W_n a linear transformation of shape $(F_v \times F_v)$. Due to linearity, we can see that

$$Rv_j W_n = (Rv_j) W_n = R(v_j W_n)$$

is SO(3) equivariant. As we elementwise multiply with a unsqueezed/expanded scalar representation, we conclude for the last term SO(3) equivariance

$$\begin{aligned} R.v_{ji,2} &\rightarrow (1 \otimes b_{ji,1}) \odot (Rv_j) W_n \\ &= (1 \otimes b_{ji,1}) \odot R(v_j W_n) = R(1 \otimes b_{ji,1}) \odot (v_j W_n) \\ &= Rv_{ji,2}. \end{aligned}$$

Since all three components in the sum are SO(3) equivariant, we conclude that the final sum is also SO(3) equivariant.

As the reader might have noticed, we build equivariant features based on linear functions and weighting $l = 1$ representations through $l = 0$ representations. This typical scaling is achieved through the tensor product $\otimes$. Our architecure however, also performs a multiplication between two $l = 1$ representations, through the cross product, which has the pleasant SO(3) invariance property that we can exploit to prove SO(3) equivariance, when scaling the output with an $l = 0$ representation.

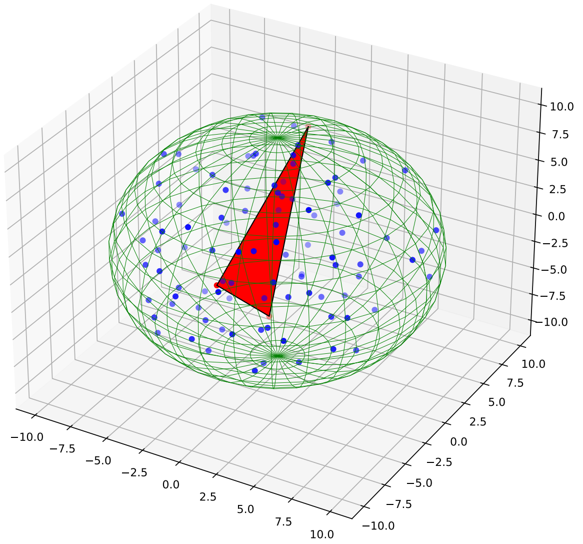


Figure 2: An example structure of the synthetic dataset. Three random points in the structure determine the vertices of a triangle, which is colored in red.

A Note on Translation Equivariance Our proposed model is translation invariant, as all vector features are initially created by means of a tensor product of (normalized) relative position $p_{ji,n}$. To see that, for any translation vector $t \in \mathbb{R}^3$ for relative positions, we can see that the calculation of such vectors[2] $p_{ji} = p_j - p_i$, are inherently translation invariant due to

$$t.p_{ji} \rightarrow (p_j + t) - (p_i + t) = p_j - p_i + t - t = p_j - p_i = p_{ji}.$$

Since we do not model absolute Cartesian coordinates, e.g., by updating the spatial coordinates through our layers, our model is not SE(3)-equivariant, i.e. next to rotation equivariance, also translation equivariant. We note that translation equivariance, however can be achieved through a simple operation such as the addition of an SE(3) representation with an SO(3) representation, e.g.

$$p_i = p_i + p_{ji,n} \otimes s,$$

where $s \in \mathbb{R}$ and reminiscent in the E(n)-GNN architecture [27], albeit the authors are not using the notation of the tensor product.

B Synthetic Dataset

We adopt the synthetic dataset from GVP [17] with slight modifications to make it a more challenging task. We create 50,000 'structures' where each 'structure' consists of $n = 100$ random points in $\mathbb{R}^3$, distributed uniformly in the ball of radius $r = 10$ with the constraint that no two points are less than distance $d = 2$ apart. Three points are randomly chosen and are labelled as 'special' which will define the vertices of a triangle. The learning task is a multitask regression of 3 targets, where the first target is to predict the distance between the center of mass (COM) of the entire structure and the COM of the triangles spanned by the three special points. The second and third task is the prediction of the perimeter and surface area of the triangle. The choice of the 3 targets refers to a structural

[2] We omit the normalization to unit vectors for brevity.

learning task, where the model requires to learn about the global shape of the structure, while the second and third targets are relational. An example structure is depicted in Figure 2. The evaluation metric is the MSE of the three tasks. We split the dataset into 80% training, 10% validation and 10% test sets.

Table 3: Evaluation of our proposed EQGAT architecture on Triangle benchmark.

Model	Triangle [MSE ↓]	No. Params [10^3]
SchNet	37.545 (1.838)	16.8
PaiNN	10.259 (0.949)	27.1
SEGNN	**3.875 (0.879)**	60.9
GVP	10.115 (1.210)	61.6
EQGAT-Full	6.003 (0.432)	27.4
EQGAT-No-Cross-Product	6.835 (1.066)	27.4
EQGAT-No-Feature-Attention	6.808 (0.326)	27.4

For the synthetic task of multitask regression we notice that the SEGNN architecture equipped with higher-order equivariant features up to rotation order 2, obtains the best performance, followed by our proposed EQGAT model that only incorporates rank 1 (vector) features. For the synthetic dataset, we did not perform any hyperparameter tuning and set the number of layers to 3 with $F_s = 32$ scalar and $F_v = 8$ vector channels and train for 50 epochs. The number of trainable parameters for SchNet, PaiNN, SEGNN and EQGAT on the synthetic Triangle dataset are listed in the last column of Table 3.

C Ablation Studies

To evaluate the benefits of our designed EQGAT architecture, we perform ablation studies and remove architectural components to isolate the effect of each design choice on performance.

Table 4: Results of the ablation studies.

	LBA [RMSE ↓]	PSR [Mean | Global R_S ↑]
No-Cross-Product	1.458 (0.011)	0.477 (0.012) | 0.827 (0.010)
No-Feature-Attention	1.466 (0.040)	**0.492 (0.007)** | 0.820 (0.002)
Full Model	**1.440 (0.027)**	0.491 (0.008) | **0.847 (0.006)**

Ablation study 1 (termed No-Cross-Product) removes the contribution of vector cross product (denoted as $v_{ji,2}$ in Eq. (6)). This leads to the effect that the vector message is solely constructed based on scaled versions of normalized relative positions ($v_{ji,0}$) and linear combinations of existing vector features ($v_{ji,1}$).

Ablation study 2 (termed No-Feature-Attention) replaces the feature attention coefficient $\alpha_{ji} \in (0,1)^{F_s}$ through a single coefficient $\alpha_{ji} \in (0,1)$.

We observe that the full EQGAT architecture obtains the best performance among the two datasets compared to the ablated models although we note that the improved performance of the full model in RMSE on the LBA benchmark and Global R_S in the PSR benchmark is difficult to attribute to the inclusion of architectural components due to the (larger) variance obtained through the 3 runs for each experiment.