Equivariant Transformers for Neural Network based Molecular Potentials

Philipp Thölke*
Computational Science Laboratory, Pompeu Fabra University,
PRBB, C/ Doctor Aiguader 88, 08003 Barcelona, Spain
philipp.thoelke@gmx.de

Gianni de Fabritiis
Computational Science Laboratory, Pompeu Fabra University,
PRBB, C/ Doctor Aiguader 88, 08003 Barcelona, Spain and
Institució Catalana de Recerca i Estudis Avançats (ICREA),
Passeig Lluis Companys 23, 08010 Barcelona, Spain
g.defabritiis@gmail.com

Abstract

The prediction of quantum mechanical properties is historically plagued by a trade-off between accuracy and speed. Machine learning potentials have previously shown great success in this domain, reaching increasingly better accuracy while maintaining computational efficiency comparable with classical force fields. In this work we propose a novel equivariant Transformer architecture, outperforming state-of-the-art on MD17 and ANI-1. Through an extensive attention weight analysis, we gain valuable insights into the black box predictor and show differences in the learned representation of conformers versus conformations sampled from molecular dynamics or normal modes. Furthermore, we highlight the importance of datasets including off-equilibrium conformations for the evaluation of molecular potentials.

1 Introduction

Quantum mechanics are essential for the computational analysis and design of molecules and materials. However, the complete solution of the Schrödinger equation is analytically and computationally not practical, which initiated the study of approximations in the past decades [Szabo and Ostlund, 1996]. A common quantum mechanics approximation method is to model atomic systems according to density functional theory (DFT), which can provide energy estimates with sufficiently high accuracy for different application cases in biology, physics, chemistry, and materials science. Even more accurate techniques like coupled-cluster exist but both still lack the computational efficiency to be applied on a larger scale, although recent advances are promising in the case of coupled-cluster [Pfau et al., 2020, Hermann et al., 2020]. Other methods include force-field and semi-empirical quantum mechanical theories, which provide very efficient estimates but lack accuracy.

The field of machine learning molecular potentials is relatively novel. The first important contributions are rooted in the Behler-Parrinello (BP) representation [Behler, 2011] and the seminal work from Rupp et al. [2012]. One of the best transferable machine learning potentials for biomolecules, ANI [Smith et al., 2017a], is based on BP. A second class of methods, mainly developed in the field of materials science and quantum chemistry, uses more modern graph convolutions [Schütt et al., 2018, Unke and Meuwly, 2019, Qiao et al., 2020, Schütt et al., 2021]. Recently, other work has shown that a

*Secondary affiliation: Institute of Cognitive Science, Osnabrück University, Germany

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

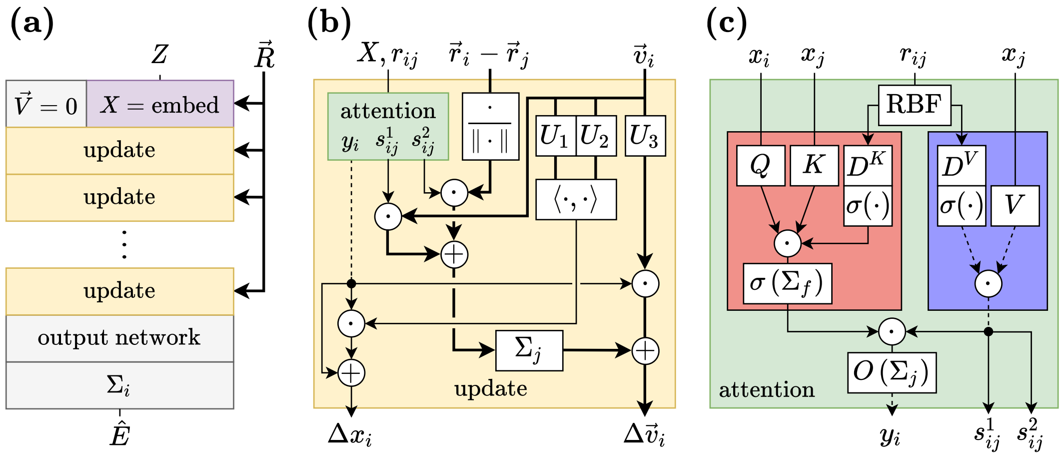

Figure 1: Overview of the equivariant Transformer architecture. Thin lines: scalar features in $\mathbb{R}^F$, thick lines: vector features in $\mathbb{R}^{3\times F}$, dashed lines: multiple feature vectors. **(a)** Transformer consisting of an embedding layer, update layers and an output network. **(b)** Residual update layer including attention based interatomic interactions and information exchange between scalar and vector features. **(c)** Modified dot-product attention mechanism, scaling values (blue) by the attention weights (red).

shift towards rotationally equivariant networks [Anderson et al., 2019, Fuchs et al., 2020, Schütt et al., 2021], particularly useful when the predicted quantities are vectors and tensors, can also improve the accuracy on scalars (e.g. energy).

In this work, we introduce an equivariant Transformer (ET) architecture for the prediction of quantum mechanical properties. By building on top of the Transformer [Vaswani et al., 2017] architecture, we are centering the design around the attention mechanism, achieving state-of-the-art accuracy on multiple benchmarks while relying solely on a learned featurization of atomic types and coordinates. Furthermore, we gain insights into the black box prediction of neural networks by analyzing the Transformer's attention weights and comparing latent representations.

2 Methods

The equivariant Transformer is made up of three main blocks. An embedding layer encodes atom types Z and the atomic neighborhood of each atom into a dense feature vector x_i. Then, a series of update layers compute interactions between pairs of atoms through a modified multi-head attention mechanism, with which the latent atomic representations are updated. Finally, an output network computes scalar atomwise predictions using gated equivariant blocks [Schütt et al., 2021], which get aggregated into a single molecular prediction. This can be matched with a scalar target variable or differentiated against atomic coordinates, providing force predictions. An illustration of the architecture is given in Figure 1. A detailed description can be found in the supplementary material.

2.1 Training

Models are trained from scratch using mean squared error loss and the Adam optimizer [Kingma and Ba, 2017] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Linear learning rate warm-up is applied as suggested by Vaswani et al. [2017] by scaling the learning rate with $\xi = \frac{\text{step}}{n_{\text{steps}}}$. After the warm-up period, we systematically decrease the learning rate by scaling with a decay factor upon reaching a plateau in validation loss. The learning rate is decreased down to a minimum of 10^{-7}. We found that weight decay and dropout do not improve generalization in this context. When training on energies and forces, we apply exponential smoothing to the energy's train and validation loss. New losses are discounted with a factor of $\alpha = 0.05$. See supplementary material for a complete list of hyperparameters. The full model comprises 1.34 million parameters.

Table 1: Results on MD trajectories from the MD17 dataset. Scores are given by the MAE of energy predictions (kcal/mol) and forces (kcal/mol/Å). NequIP does not provide errors on energy, for PaiNN we include the results with lower force error out of training only on forces versus on forces and energy. Benzene corresponds to the dataset originally released in Chmiela et al. [2017], which is sometimes left out from the literature. ET results are averaged over three random splits $\pm$ standard deviation.

Molecule		SchNet	PhysNet	DimeNet	PaiNN	NequIP	ET
Aspirin	*energy*	0.37	0.230	0.204	0.167	-	**0.124** $\pm$ 0.001
	forces	1.35	0.605	0.499	0.338	0.348	**0.255** $\pm$ 0.007
Benzene	*energy*	0.08	-	0.078	-	-	**0.056** $\pm$ 0.003
	forces	0.31	-	**0.187**	-	**0.187**	0.201 $\pm$ 0.008
Ethanol	*energy*	0.08	0.059	0.064	0.064	-	**0.054** $\pm$ 0.000
	forces	0.39	0.160	0.230	0.224	0.208	**0.116** $\pm$ 0.001
Malondialdehyde	*energy*	0.13	0.094	0.104	0.091	-	**0.079** $\pm$ 0.001
	forces	0.66	0.319	0.383	0.319	0.337	**0.176** $\pm$ 0.003
Naphthalene	*energy*	0.16	0.142	0.122	0.116	-	**0.085** $\pm$ 0.000
	forces	0.58	0.310	0.215	0.077	0.097	**0.060** $\pm$ 0.002
Salicylic Acid	*energy*	0.20	0.126	0.134	0.116	-	**0.094** $\pm$ 0.001
	forces	0.85	0.337	0.374	0.195	0.238	**0.135** $\pm$ 0.006
Toluene	*energy*	0.12	0.100	0.102	0.095	-	**0.074** $\pm$ 0.000
	forces	0.57	0.191	0.216	0.094	0.101	**0.066** $\pm$ 0.001
Uracil	*energy*	0.14	0.108	0.115	0.106	-	**0.096** $\pm$ 0.000
	forces	0.56	0.218	0.301	0.139	0.173	**0.094** $\pm$ 0.000

3 Experiments and Results

The MD17 [Chmiela et al., 2017] dataset consists of molecular dynamics (MD) trajectories of small organic molecules, including both energies and forces. Forces are predicted using the negative gradient of the energy with respect to atomic coordinates $\vec{F}_i = -\partial \hat{E}/\partial \vec{r}_i$. We train on 1000 samples from which 50 are used for validation. The remaining data is used for evaluation and is the basis for comparison with other work. Separate models are trained for each molecule using a combined loss function for energies and forces where the energy loss is multiplied with a factor of 0.2 and the force loss with 0.8. An overview of the results and comparison to SchNet [Schütt et al., 2017b], PhysNet [Unke and Meuwly, 2019], DimeNet [Klicpera et al., 2020], PaiNN [Schütt et al., 2021] and NequIP [Batzner et al., 2021] can be found in Table 1.

To evaluate the architecture's capabilities on a large collection of off-equilibrium conformations, we train and evaluate the equivariant Transformer on the ANI-1 [Smith et al., 2017b] dataset. It contains 22,057,374 configurations of 57,462 small organic molecules with up to 8 heavy atoms and atomic species H, C, N, and O. The off-equilibrium data points are generated via exhaustive normal mode sampling of the energy minimized molecules. The model is fitted on DFT energies from 80% of the dataset, while 5% are used as validation and the remaining 15% of the data make up the test set. Figure 2 compares the equivariant Transformer's performance to previous methods DTNN [Schütt et al., 2017a], SchNet [Schütt et al., 2017b], MGCN [Lu et al., 2019] and ANI [Smith et al., 2017a].

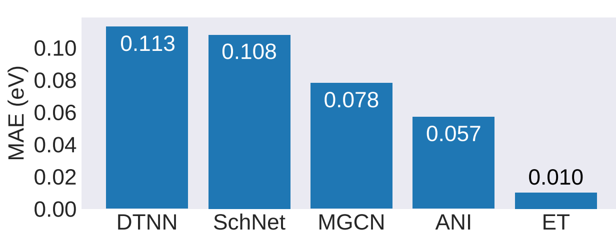

Figure 2: Comparison of testing MAE on the ANI-1 dataset in eV. Results for DTNN, SchNet and MGCN are provided by Lu et al. [2019]. The ANI method refers to the ANAKIN-ME [Smith et al., 2017a] model used for constructing the ANI-1 dataset.

3.1 Attention Weight Analysis

Neural network predictions are notoriously difficult to interpret due to the complex nature of the learned transformations. To shed light into the black box predictor, we extract and analyze the equivariant Transformer's attention weights. We run inference on the ANI-1 [Smith et al., 2017b], QM9 [Ramakrishnan et al., 2014], and MD17 [Chmiela et al., 2017] test sets for all molecules and extract each sample's attention matrix from all attention heads in all layers. Attention rollout [Abnar and Zuidema, 2020] under the single head assumption is applied during the extraction, resulting in a single attention matrix per sample. We average attention weights over each unique combination of interacting atom types, leaving two attention scores for each pair of atom types, one from the perspective of z_1 attending z_2 and vice versa.

The attention scores are compared to bond probabilities extracted from the same molecules to make sure the network does not simply attend interacting atoms proportional to the relative frequency in the dataset. Figure 3 presents a summary of the distilled probabilities and attention scores for QM9, ANI-1, and the average attention scores for all MD17 models. We normalize each row to sum to one.

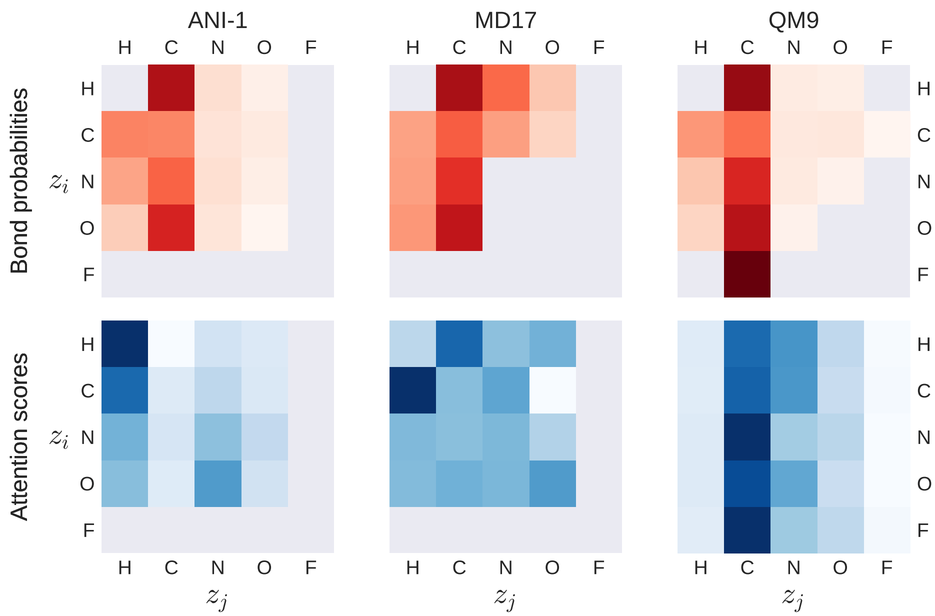

Figure 3: Depiction of bond probabilities and attention scores extracted from the ET using QM9 (total energy U_0), MD17 (average over 8 discussed molecules) and ANI-1 testing data. Attention scores are given as z_i attending z_j, bond probabilities follow the same idea, showing the conditional probability of a bond between z_i and z_j, given z_i. Darker colors correspond to larger values, element pairs without data are grayed out.

4 Discussion

In this work, we introduce a novel attention-based architecture for the prediction of quantum mechanical properties, leveraging the use of rotationally equivariant features. We set a new state-of-the-art on all MD17 targets (except force prediction of the molecule Benzene) and demonstrate the architecture's ability to work in a low data regime. By extracting and analyzing the model's attention weights, we gain insights into the molecular representation, which is characterized by the nature of the corresponding training data. We show that the model does not pay much attention to the location of hydrogen when trained only on energy-minimized molecules, while a model trained on data including off-equilibrium conformations focuses to a large degree on hydrogen. Neural networks and especially Transformers are known to require large amounts of training data and computational power. It should be taken into consideration that training these kinds of models requires significant amounts of energy and causes the emission of greenhouse gases.

Software and Data

The equivariant Transformer is implemented in PyTorch [Paszke et al., 2019], using PyTorch Geometric [Fey and Lenssen, 2019] as the underlying framework for geometric deep learning. Training is done using pytorch-lightning [Falcon and The PyTorch Lightning team, 2019], a high-level interface for training PyTorch models. The datasets QM9[2], MD17[3] and ANI-1[4] are publicly available and all source code for training, running and analyzing the models presented in this work is available at `https://github.com/torchmd/torchmd-net`.

References

S. Abnar and W. Zuidema. Quantifying Attention Flow in Transformers. *arXiv:2005.00928 [cs]*, May 2020. URL `http://arxiv.org/abs/2005.00928`. arXiv: 2005.00928.

B. Anderson, T.-S. Hy, and R. Kondor. Cormorant: Covariant Molecular Neural Networks. *arXiv:1906.04015 [physics, stat]*, Nov. 2019. URL `http://arxiv.org/abs/1906.04015`. arXiv: 1906.04015.

S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky. SE(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. *arXiv:2101.03164 [cond-mat, physics:physics]*, July 2021. URL `http://arxiv.org/abs/2101.03164`. arXiv: 2101.03164.

J. Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, Feb. 2011. ISSN 0021-9606. doi: 10.1063/1.3553717. URL `https://aip.scitation.org/doi/10.1063/1.3553717`. Publisher: American Institute of Physics.

S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3 (5):e1603015, May 2017. ISSN 2375-2548. doi: 10.1126/sciadv.1603015. URL `https://advances.sciencemag.org/content/3/5/e1603015`. Publisher: American Association for the Advancement of Science Section: Research Article.

K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller. Rethinking Attention with Performers. *arXiv:2009.14794 [cs, stat]*, Mar. 2021. URL `http://arxiv.org/abs/2009.14794`. arXiv: 2009.14794.

W. Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019. URL `https://github.com/PyTorchLightning/pytorch-lightning`.

M. Fey and J. E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric, May 2019. URL `https://github.com/rusty1s/pytorch_geometric`.

F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. *arXiv:2006.10503 [cs, stat]*, Nov. 2020. URL `http://arxiv.org/abs/2006.10503`. arXiv: 2006.10503.

J. Hermann, Z. Schätzle, and F. Noé. Deep neural network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, Oct. 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y. URL `http://arxiv.org/abs/1909.08423`. arXiv: 1909.08423.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Jan. 2017. URL `http://arxiv.org/abs/1412.6980`. arXiv: 1412.6980.

J. Klicpera, J. Groß, and S. Günnemann. Directional Message Passing for Molecular Graphs. *arXiv:2003.03123 [physics, stat]*, Mar. 2020. URL `http://arxiv.org/abs/2003.03123`. arXiv: 2003.03123.

[2]https://doi.org/10.6084/m9.figshare.c.978904.v5

[3]http://www.quantum-machine.org/gdml/#datasets

[4]https://figshare.com/articles/dataset/ANI-1x_Dataset_Release/10047041/1

C. Lu, Q. Liu, C. Wang, Z. Huang, P. Lin, and L. He. Molecular Property Prediction: A Multilevel Quantum Interactions Modeling Perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1052–1060, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33011052. URL https://ojs.aaai.org/index.php/AAAI/article/view/3896. Number: 01.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]*, Dec. 2019. URL http://arxiv.org/abs/1912.01703. arXiv: 1912.01703.

D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3): 033429, Sept. 2020. doi: 10.1103/PhysRevResearch.2.033429. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429. Publisher: American Physical Society.

Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller III. OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital Features. *arXiv:2007.08026 [physics]*, Sept. 2020. doi: 10.1063/5.0021955. URL http://arxiv.org/abs/2007.08026. arXiv: 2007.08026.

R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, Aug. 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422. Number: 1 Publisher: Nature Publishing Group.

M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical Review Letters*, 108(5): 058301, Jan. 2012. doi: 10.1103/PhysRevLett.108.058301. URL https://link.aps.org/doi/10.1103/PhysRevLett.108.058301. Publisher: American Physical Society.

K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1):13890, Jan. 2017a. ISSN 2041-1723. doi: 10.1038/ncomms13890. URL https://www.nature.com/articles/ncomms13890. Publisher: Nature Publishing Group.

K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv:1706.08566 [physics, stat]*, Dec. 2017b. URL http://arxiv.org/abs/1706.08566. arXiv: 1706.08566.

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(24): 241722, Mar. 2018. ISSN 0021-9606. doi: 10.1063/1.5019779. URL https://aip.scitation.org/doi/abs/10.1063/1.5019779. Publisher: American Institute of Physics.

K. T. Schütt, O. T. Unke, and M. Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv:2102.03150 [physics]*, June 2021. URL http://arxiv.org/abs/2102.03150. arXiv: 2102.03150.

J. S. Smith, O. Isayev, and A. E. Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science*, 8(4):3192–3203, Mar. 2017a. ISSN 2041-6539. doi: 10.1039/C6SC05720A. URL https://pubs.rsc.org/en/content/articlelanding/2017/sc/c6sc05720a. Publisher: The Royal Society of Chemistry.

J. S. Smith, O. Isayev, and A. E. Roitberg. ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data*, 4(1):170193, Dec. 2017b. ISSN 2052-4463. doi: 10.1038/sdata.2017.193. URL https://www.nature.com/articles/sdata2017193. Publisher: Nature Publishing Group.

A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Books on Chemistry. Dover Publications, July 1996. ISBN 978-0-486-69186-2. Google-Books-ID: 6mV9gYzEkgIC.

O. T. Unke and M. Meuwly. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments and Partial Charges. *arXiv:1902.08408 [physics]*, Mar. 2019. doi: 10.1021/acs.jctc.9b00181. URL `http://arxiv.org/abs/1902.08408`. arXiv: 1902.08408.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, Dec. 2017. URL `http://arxiv.org/abs/1706.03762`. arXiv: 1706.03762.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Hyperparameters

Table 2 provides an overview of hyperparameters used for training the ET on MD17 and ANI-1.

Table 2: Comparison of various hyperparameters used for MD17 and ANI-1.

Parameter	MD17	ANI-1
initial learning rate	$1 \cdot 10^{-3}$	$7 \cdot 10^{-4}$
lr patience (epochs)	30	5
lr decay factor	0.8	0.5
lr warmup steps	1,000	10,000
batch size	8	2048
no. layers	6	6
no. RBFs	32	32
feature dimension	128	128

B Neighbor Embedding

The embedding layer assigns two learned vectors to each atom type z_i. One is used to encode information specific to an atom, the other takes the role of a neighborhood embedding. The neighborhood embedding, which is an embedding of the types of neighboring atoms, is multiplied by a distance filter. This operation resembles a continuous-filter convolution [Schütt et al., 2017b] but, as it is used in the first layer, allows the model to store atomic information in two separate weight matrices. These can be thought of as containing information that is intrinsic to an atom versus information about the interaction of two atoms.

The distance filter is generated from expanded interatomic distances using a linear transformation W^F. First, the distance d_{ij} between two atoms i and j is expanded via a set of exponential normal radial basis functions e^{RBF}, defined as

$$e_k^{\mathrm{RBF}}(d_{ij}) = \phi(d_{ij}) \exp(-\beta_k (\exp(-d_{ij}) - \mu_k)^2) \tag{1}$$

where β_k and μ_k are fixed parameters specifying the center and width of radial basis function k. The μ vector is initialized with values equally spaced between $\exp(-d_{\mathrm{cut}})$ and 1, β is initialized as $(2K^{-1}(1 - \exp(-d_{\mathrm{cut}})))^{-2}$ for all k as proposed by Unke and Meuwly [2019]. The cutoff distance d_{cut} was set to 5Å. The cosine cutoff $\phi(d_{ij})$ is used to ensure a smooth transition to 0 as d_{ij} approaches d_{cut} in order to avoid jumps in the regression landscape. It is given by

$$\phi(d_{ij}) = \begin{cases} \frac{1}{2}\left(\cos\left(\frac{\pi d_{ij}}{d_{\mathrm{cut}}}\right) + 1\right), & \text{if } d_{ij} \leq d_{\mathrm{cut}} \\ 0, & \text{if } d_{ij} > d_{\mathrm{cut}}. \end{cases} \tag{2}$$

The neighborhood embedding n_i for atom i is then defined as

$$n_i = \sum_{j=1}^{N} a_n(z_j) \odot W^F e^{\mathrm{RBF}}(d_{ij}) \tag{3}$$

with a_n being the neighborhood embedding function and N the number of atoms in the graph. The final atomic embedding x_i is calculated as a linear projection of the concatenated intrinsic embedding and neighborhood embedding $[a_i(z_i), n_i]$, resulting in

$$x_i = W^C \, [a_i(z_i), n_i] + b^C \tag{4}$$

with a_i being the intrinsic embedding function. The vector features $\vec{v}_i$ are initially set to 0.

C Equivariant Transformer Architecture

C.1 Modified Attention Mechanism

We use a modified multi-head attention mechanism (Figure 1c), extending dot-product attention, in order to include edge data into the calculation of attention weights. The edge data, i.e. interatomic distances r_{ij}, are projected into two multidimensional filters D^K and D^V, according to

$$\begin{aligned} D^K &= \sigma(W^{D^K} e^{\mathrm{RBF}}(r_{ij}) + b^{D^K}) \\ D^V &= \sigma(W^{D^V} e^{\mathrm{RBF}}(r_{ij}) + b^{D^V}) \end{aligned} \tag{5}$$

The attention weights are computed via an extended dot product, i.e. an elementwise multiplication and subsequent sum over the feature dimension, of the three input vectors: query Q, key K and distance projection D^K:

$$\mathrm{dot}(Q, K, D^K) = \sum_k^F Q_k \odot K_k \odot D_k^K \tag{6}$$

The resulting matrix is passed through a nonlinear activation function and is weighted by a cosine cutoff (see equation 2), ensuring that atoms with a distance larger than d_{cut} do not interact. Traditionally, the resulting attention matrix A is passed through a softmax activation, however, we replace this step with a SiLU function to preserve the distance cutoff. The softmax scaling factor of $\sqrt{d_k}^{-1}$, which normally rescales small gradients from the softmax function, is left out. Work by Choromanski et al. [2021] suggests that replacing the softmax activation function in Transformers with ReLU-like functions might even improve accuracy, supporting the idea of switching to SiLU in this case.

We place a continuous filter graph convolution [Schütt et al., 2017b] in the attention mechanism's value pathway. This enables the model to not only consider interatomic distances in the attention weights but also incorporate this information into the feature vectors directly. The resulting representation is split into three equally sized vectors $s_{ij}^1, s_{ij}^2, s_{ij}^3 \in \mathbb{R}^F$. The vector s_{ij}^3 is scaled by the attention matrix A and aggregated over the value-dimension, leading to an updated list of feature vectors. The linear transformation O is used to combine the attention heads' outputs into a single feature vector $y_i \in \mathbb{R}^{384}$.

$$\begin{aligned} s_{ij}^1, s_{ij}^2, s_{ij}^3 &= \mathrm{split}(V_j \odot {D^V}_{ij}) \\ y_i &= O\left(\sum_j^N A_{ij} \cdot s_{ij}^3\right) \end{aligned} \tag{7}$$

The attention mechanism's output, therefore, corresponds to the updated scalar feature vectors y_i and scalar filters s_{ij}^1 and s_{ij}^2, which are used to weight the directional information inside the update layer.

C.2 Update Layer

The update layer (Figure 1b) is used to compute interactions between atoms (attention block) and exchange information between scalar and vector features. The updated scalar features y_i from the attention block are split up into three feature vectors $q_i^1, q_i^2, q_i^3 \in \mathbb{R}^F$. The first feature vector, q_i^1, takes the role of a residual around the scaled vector features. The resulting scalar feature update Δx_i of this update layer is then defined as

$$\Delta x_i = q_i^1 + q_i^2 \odot \langle U_1 \vec{v}_i, U_2 \vec{v}_i \rangle \tag{8}$$

where $\langle U_1 \vec{v}_i, U_2 \vec{v}_i \rangle$ denotes the scalar product of vector features $\vec{v}_i$, transformed by linear projections U_1 and U_2.

On the side of the vector features, scalar information is introduced through a multiplication between q_i^3 and a linear projection of the vector features $U_3 \vec{v}_i$. The representation is updated with equivariant features using the directional vector between two atoms. The edge-wise directional information is multiplied with scalar filter s_{ij}^2 and added to the rescaled vector features $s_{ij}^1 \cdot \vec{v}_j$. The result is aggregated inside each atom, forming $\vec{w}_i$. The final vector feature update $\Delta \vec{v}_i$ for the current update layer is then produced by adding the weighted scalar features to the equivariant features $\vec{w}_i$.

$$\begin{aligned} \vec{w}_i &= \sum_j^N s_{ij}^1 \odot \vec{v}_j + s_{ij}^2 \odot \frac{\vec{r}_i - \vec{r}_j}{\|\vec{r}_i - \vec{r}_j\|} \\ \Delta \vec{v}_i &= \vec{w}_i + q_i^3 \odot U_3 \vec{v}_i \end{aligned} \tag{9}$$