Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs

Anonymous Author(s)
Affiliation
Address
email

Abstract

3D-related inductive biases like translational invariance and rotational equivariance are indispensable to graph neural networks operating on 3D atomistic graphs such as molecules. Inspired by the success of Transformers in various domains, we study how to incorporate these inductive biases into Transformers. In this paper, we present Equiformer, a graph neural network leveraging the strength of Transformer architectures and incorporating $SE(3)/E(3)$-equivariant features based on irreducible representations (irreps). Irreps features encode equivariant information in channel dimensions without complicating graph structures. The simplicity enables us to directly incorporate them by replacing original operations with equivariant counterparts. Moreover, to better adapt Transformers to 3D graphs, we propose a novel equivariant graph attention, which considers both content and geometric information such as relative position contained in irreps features. To improve expressivity of the attention, we replace dot product attention with multi-layer perceptron attention and include non-linear message passing. We benchmark Equiformer on two quantum properties prediction datasets, QM9 and OC20. For QM9, among models trained with the same data partition, Equiformer achieves best results on 11 out of 12 regression tasks. For OC20, under the same setting of training with IS2RE data only, Equiformer improves upon state-of-the-art models.

1 Introduction

Machine learned models can accelerate the prediction of quantum properties of atomistic systems like molecules by learning approximations of *ab initio* calculations [23, 66, 28, 20, 4, 9, 38, 57, 52, 41, 40]. In particular, graph neural networks (GNNs) have gained increasing popularity due to their performance. By modeling atomistic systems as graphs, GNNs naturally treat the set-like nature of collections of atoms, encode the interaction between atoms in node features and update the features by passing messages between nodes. One factor contributing to the success of neural networks is the ability to incorporate inductive biases that exploit the symmetry of data. Take convolutional neural networks (CNNs) for 2D images as an example: Patterns in images should be recognized regardless of their positions, which motivates the inductive bias of translational equivariance. As for atomistic graphs, where each atom has its coordinate in 3D Euclidean space, we consider inductive biases related to 3D Euclidean group $E(3)$, which include equivariance to 3D translation, 3D rotation, and inversion. Concretely, some properties like energy of an atomistic system should be constant regardless of how we shift the system; others like force should be rotated accordingly if we rotate the system. To incorporate these inductive biases, equivariant and invariant neural networks have been proposed. The former leverages geometric tensors like vectors for equivariant node features [54, 60, 32, 19, 4, 5, 40], and the latter augments graphs with invariant information such as distances and angles extracted from 3D graphs [48, 21, 20, 37, 51, 31].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A parallel line of research focuses on applying Transformer networks [58] to other domains like computer vision [8, 14, 55] and graph [16, 33, 64, 50] and has demonstrated widespread success. However, as Transformers were developed for sequence data [13, 3, 7], it is crucial to incorporate domain-related inductive biases. For example, Vision Transformer [14] shows that adopting a pure Transformer to image classification cannot generalize well and achieves worse results than CNNs when trained on only ImageNet [45] since it lacks inductive biases like translational invariance. Note that ImageNet contains over 1.28M images and the size is already larger than that of many quantum properties prediction datasets [44, 43, 9]. Therefore, this highlights the necessity of including correct inductive biases when applying Transformers to the domain of 3D atomistic graphs.

In this work, we present **Equiformer**, an equivariant graph neural network utilizing $SE(3)$/$E(3)$-**equi**variant features built from irreducible representations (irreps) and **equi**variant attention mechanisms to combine the 3D-related inductive bias with the strength of Trans**former**. Irreps features encode equivariant information in channel dimensions without complicating graph structures. The simplicity enables us to directly incorporate them into Transformers through replacing original operations with equivariant counterparts and introducing an additional equivariant operation called tensor product. Moreover, we propose a novel equivariant graph attention, which considers both content and geometric information such as relative position contained in irreps features. Equivariant graph attention improves upon typical attention in Transformers by replacing dot product attention with theoretically stronger multi-layer perceptron attention and including non-linear message passing. With these innovations, Equiformer demonstrates the possibility of generalizing Transformer-like networks to 3D atomistic graphs and achieves competitive results on two quantum properties prediction datasets, QM9 [44, 43] and OC20 [9]. For QM9, compared to models trained with the same data partition, Equiformer achieves best results on 11 out of 12 regression tasks. For OC20, under the setting of training with IS2RE data only, Equiformer improves upon state-of-the-art models.

2 Background

2.1 Graph Neural Networks

Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of nodes $i \in \mathcal{V}$ and edges $(i, j) \in \mathcal{E}$ representing relationships between nodes i and j. Graphs can represent atomistic systems like molecules, where nodes are atoms and edges can represent bonds or atom pairs within a certain distance; we refer to these as atomistic graphs. Graph neural networks (GNNs) extract meaningful representations from graphs through message-passing layers. Given feature f_i^l on node i and edge feature e_{ij}^l at the l-th layer, it first aggregates messages m_{ij}^l from neighbors and then update features on each node as follows:

$$m_{ij}^l = \phi_m(f_i^l, f_j^l, e_{ij}^l) \quad \text{and} \quad f_i^{l+1} = \phi_u(f_i^l, \sum_{j \in \mathcal{N}(i)} m_{ij}^l) \tag{1}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of node i and ϕ_m and ϕ_u are learnable functions. For atomistic systems in 3D spaces, each node i is additionally associated with its position $\vec{r}_i$ and typically edge features become functions of relative position $\vec{r}_{ij} = \vec{r}_j - \vec{r}_i$.

2.2 $E(3)$ Equivariance

Atomistic systems are often described using coordinate systems. For 3D Euclidean space, we can freely choose coordinate systems and change between them via the symmetries of 3D space: 3D translation, rotation and inversion ($\vec{r} \rightarrow -\vec{r}$). The groups of 3D translation, rotation and inversion form Euclidean group $E(3)$, with the first two forming $SE(3)$, the second being $SO(3)$, and the last two forming $O(3)$. The laws of physics are invariant to the choice of coordinate systems and therefore properties of atomistic systems are equivariant, e.g., when we rotate our coordinate system, quantities like energy remain the same while others like force rotate accordingly. Formally, a function f mapping between vector spaces X and Y is equivariant to a group of transformation G if for any input $x \in X$, output $y \in Y$ and group element $g \in G$, we have $f(D_X(g)x) = D_Y(g)f(x)$, where $D_X(g)$ and $D_Y(g)$ are transformation matrices parametrized by g in X and Y.

Incorporating equivariance into neural networks as inductive biases is crucial as this enables generalizing to unseen data in a predictable manner. For example, 2D convolution f is equivariant to the group of 2D translation, and when input features x are shifted by an amount parametrized by g, output features y will be shifted by the same amount. For 3D atomistic graphs, we consider the

group of $E(3)$. Features f_i^l and e_{ij}^l and learnable functions ϕ_m and ϕ_u should be $E(3)$-equivariant to geometric transformation acting on position $\vec{r}_i$. In this work, following previous works [54, 32, 60] implemented in `e3nn` [22], we achieve $SE(3)/E(3)$-equivariance by using equivariant features based on vector spaces of irreducible representations and equivariant operations like tensor product for learnable functions.

2.3 Irreducible Representations

A group representation [15, 65] defines the transformation matrices $D_X(g)$ of group elements g that act on a vector space X. For 3D Euclidean group $E(3)$, two examples of vector spaces with different transformation matrices are scalars and Euclidean vectors in $\mathbb{R}^3$, i.e., vectors change with rotation while scalars do not. To address translation symmetry, we simply operate on relative positions. Below we focus our discussion on $O(3)$. The transformation matrices of rotation and inversion are separable and commute, and we first discuss irreducible representations of $SO(3)$.

Any group representation of $SO(3)$ on a given vector space can be decomposed into a concatenation of provably smallest transformation matrices called irreducible representations (irreps). Specifically, for group element $g \in SO(3)$, there are $(2L+1)$-by-$(2L+1)$ irreps matrices $D_L(g)$ called Wigner-D matrices acting on $(2L+1)$-dimensional vector spaces, where degree L is a non-negative integer. L can be interpreted as an angular frequency and determines how quickly vectors change when rotating coordinate systems. $D_L(g)$ of different L act on independent vector spaces. Vectors transformed by $D_L(g)$ are type-L vectors, with scalars and Euclidean vectors being type-0 and type-1 vectors. It is common to index elements of type-L vectors with an index m called order, where $-L \leq m \leq L$.

The group of inversion $\mathbb{Z}_2$ only has two elements, identity and inversion, and two irreps, even e and odd o. Vectors transformed by irrep e do not change sign under inversion while those by irrep o do. We create irreps of $O(3)$ by simply multiplying those of $SO(3)$ and $\mathbb{Z}_2$, and we introduce parity p to type-L vectors to denote how they transform under inversion. Therefore, type-L vectors in $SO(3)$ are extended to type-(L, p) vectors in $O(3)$, where p is e or o. In the following, we use type-L vectors for the ease of discussion, but we can generalize to type-(L, p) vectors, unless otherwise stated.

Irreps Features. We concatenate multiple type-L vectors to form $SE(3)$-equivariant irreps features. Concretely, irreps feature f has C_L type-L vectors, where $0 \leq L \leq L_{max}$ and C_L is the number of channels for type-L vectors. We index irreps features f by channel c, degree L, and order m and denote as $f_{c,m}^{(L)}$. Different channels of type-L vectors are parametrized by different weights but are transformed with the same Wigner-D matrix $D_L(g)$. Regular scalar features correspond to including only type-0 vectors. This can generalize to $E(3)$ by including inversion and extending L to (L, p).

Spherical Harmonics. Euclidean vectors $\vec{r}$ in $\mathbb{R}^3$ can be projected into type-L vectors $f^{(L)}$ by using spherical harmonics (SH) $Y^{(L)}$: $f^{(L)} = Y^{(L)}(\frac{\vec{r}}{||\vec{r}||})$. SH are $E(3)$-equivariant with $D_L(g)f^{(L)} = Y^{(L)}(\frac{D_1(g)\vec{r}}{||D_1(g)\vec{r}||})$. SH of relative position $\vec{r}_{ij}$ generates the first set of irreps features. Equivariant information propagates to other irreps features through equivariant operations like the tensor product.

2.4 Tensor Product

We use tensor products to interact different type-L vectors and first discuss the tensor product for $SO(3)$. The tensor product denoted as $\otimes$ uses Clebsch-Gordan coefficients to combine type-L_1 vector $f^{(L1)}$ and type-L_2 vector $g^{(L2)}$ and produces type-L_3 vector $h^{(L3)}$ as follows:

$$h_{m_3}^{(L3)} = (f^{(L1)} \otimes g^{(L2)})_{m_3} = \sum_{m_1=-L_1}^{L_1} \sum_{m_2=-L_2}^{L_2} C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)} f_{m_1}^{(L1)} g_{m_2}^{(L2)} \tag{2}$$

where m_1 denotes order and refers to the m_1-th element of $f^{(L1)}$. Clebsch-Gordan coefficients $C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)}$ are non-zero only when $|L_1 - L_2| \leq L_3 \leq |L_1 + L_2|$ and thus restrict output vectors to be of certain types. For efficiency, we discard vectors with $L > L_{max}$, where L_{max} is a hyper-parameter, to prevent vectors of increasingly higher dimensions. The tensor product is an equivariant operation, with $(D_{L_1}(g')f^{(L_1)}) \otimes (D_{L_2}(g')g^{(L_2)}) = D_{L_3}(g')h^{(L_3)}$ for $g' \in SO(3)$.

We call each distinct non-trivial combination of $L_1 \otimes L_2 \rightarrow L_3$ a path. Each path is independently equivariant, and we can assign one learnable weight to each path in tensor products, which is similar

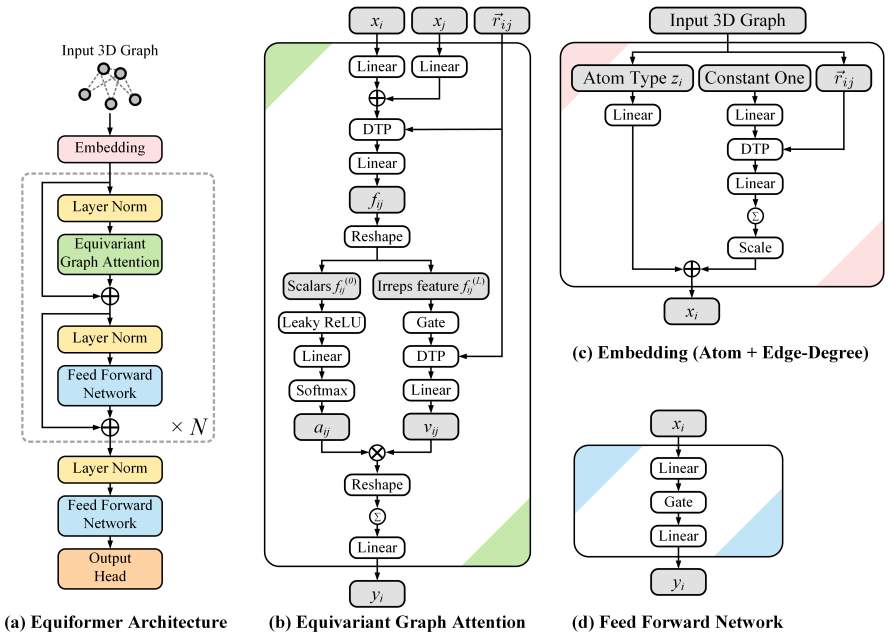

Figure 1: **Architecture of Equiformer.** We embed input 3D graphs with atom and edge-degree embeddings and process them with Transformer blocks, consisting of equivariant graph attention and feed forward networks. In this figure, "$\otimes$" denotes multiplication, "$\oplus$" denotes addition, and "DTP" stands for depth-wise tensor product. $\sum$ within a circle denotes summation over all neighbors. Gray cells indicate intermediate irreps features.

to typical linear layers. We can generalize Eq. 2 to irreps features and include multiple channels of vectors of different types through iterating over all paths associated with channels of vectors. In this way, weights are indexed by $(c_1, l_1, c_2, l_2, c_3, l_3)$, where c_1 is the c_1-th channel of type-l_1 vector in input irreps feature. We use $\otimes_w$ to represent tensor product with weights w. Weights can be conditioned on quantities like relative distances. For example, in GNNs, we use tensor products of irreps features f_j and SH embedding of relative position $\vec{r}_{ij}$ to transform features into messages and use some learnable functions transforming $||\vec{r}_{ij}||$ into weights of tensor products [48, 4]. Please refer to appendix for discussion on including inversion in tensor products.

2.5 $E(3)$-Equivariant Neural Networks

$E(3)$-equivariant neural networks (ENNs) are built such that every operation f is equivariant to Euclidean symmetry, satisfying that for $g \in E(3)$, inputs x and scalar weights w and $f : X \times W \rightarrow Y$, $f(D_X(g)x, w) = D_Y(g)f(x, w)$ [54, 32, 60]. In addition to the tensor product, equivariant operations also include equivariant activation functions and normalizations as in typical neural networks, and we discuss those used in Equiformer in Sec. 3.1. Because an ENN is a composition of equivariant operations, it is also equivariant; transformations acting on input spaces propagate to the output spaces, e.g., when we rotate an input atomistic graph, predicted vectors will be rotated.

3 Equiformer

We incorporate $SE(3)/E(3)$-equivariant irreps features into Transformers [58] and use equivariant operations. To better adapt Transformers to 3D graph structures, we propose equivariant graph attention. The overall architecture of Equiformer is illustrated in Fig. 1.

3.1 Equivariant Operations for Irreps Features

Here we discuss equivariant operations used in Equiformer that serve as building blocks for equivariant graph attention and other modules. They include the equivariant version of the original operations in Transformers and the depth-wise tensor product, as illustrated in Fig. 2 in appendix.

Linear. Linear layers are generalized to irreps features by transforming different type-L vectors separately. Specifically, we apply separate linear operations to each group of type-L vectors. We remove bias terms for non-scalar features with $L > 0$ as biases do not depend on inputs, and therefore, including biases for type-L vectors with $L > 0$ can break equivariance.

Layer Normalization. Transformers adopt layer normalization (LN) [2] to stabilize training. Given input $x \in \mathbb{R}^{N \times C}$, with N being the number of nodes and C the number of channels, LN calculates the linear transformation of normalized input as follows:

$$\text{LN}(x) = \frac{x - \mu_C}{\sigma_C} \circ \gamma + \beta \tag{3}$$

where $\mu_C, \sigma_C \in \mathbb{R}^{N \times 1}$ are mean and standard deviation of input x along channel dimension C, and $\gamma, \beta \in \mathbb{R}^{1 \times C}$ are learnable parameters. $\circ$ denotes element-wise product. By viewing standard deviation as the root mean square value (RMS) of L2-norm of type-L vectors, LN can be generalized to irreps features. Specifically, given input $x \in \mathbb{R}^{N \times C \times (2L+1)}$ of type-L vectors, the output is:

$$\text{LN}(x) = \frac{x}{\text{RMS}_C(\text{norm}(x))} \circ \gamma \tag{4}$$

where $\text{norm}(x) \in \mathbb{R}^{N \times C \times 1}$ calculates the L2-norm of each type-L vectors in x, and $\text{RMS}_C(\text{norm}(x)) \in \mathbb{R}^{N \times 1 \times 1}$ calculates the RMS of L2-norm with mean taken along channel dimension C. We remove mean and biases for type-L vectors with $L \neq 0$ following linear layers.

Gate. We adopt the gate activation [60] for equivariant activation function. Typical activation functions are applied to type-0 vectors. For vectors of higher L, we multiply them with non-linearly transformed type-0 vectors to maintain equivariance. Specifically, given input x containing C_L type-L vectors with $0 < L \leq L_{max}$ and $(C_0 + \sum_{L=1}^{L_{max}} C_L)$ type-0 vectors, we apply SiLU activation [17, 42] to the first C_0 type-0 vectors and sigmoid function to the other $\sum_{L=1}^{L_{max}} C_L$ type-0 vectors to obtain non-linear weights. Then, we multiply each type-L vector with corresponding non-linear weights. After the gate activation, the number of channels for type-0 vectors is reduced to C_0.

Depth-wise Tensor Product. The tensor product defines interaction between vectors of different L. To improve its efficiency, we use the depth-wise tensor product (DTP), which restricts one type-L vector in output irreps features depends only on one type-L' vector in input irreps features, where L can be equal to or different from L'. This is similar to depth-wise convolution [27], where one output channel depends on only one input channel. Weights w in the DTP can be input-independent or conditioned on relative distances, and the DTP between two tensors x and y is denoted as $x \otimes_w^{DTP} y$.

3.2 Equivariant Graph Attention

Self-attention [58, 59, 19, 30, 64, 6] transforms features sent from one spatial location to another with input-dependent weights. We use the notion from Transformers [58] and message passing networks [23, 46, 47, 5] and define message m_{ij} sent from node j to node i as follows:

$$m_{ij} = a_{ij} \times v_{ij} \tag{5}$$

where attention weights a_{ij} depend on features on node i and its neighbors $\mathcal{N}(i)$ and values v_{ij} are transformed with input-independent weights. In Transformers and Graph Attention Networks (GAT) [59, 6], v_{ij} depends only on node j. In message passing networks [23, 46, 47, 5], v_{ij} depends on features on nodes i and j with constant a_{ij}. The proposed equivariant graph attention adopts tensor products to incorporate content and geometric information and utilizes multi-layer perceptron attention for a_{ij} and non-linear message passing for v_{ij} as illustrated in Fig. 1(b).

Incorporating Content and Geometric Information. Given features x_i and x_j on target node i and source node j, we combine the two features with two linear layers to obtain initial message $x_{ij} = \text{Linear}_{dst}(x_i) + \text{Linear}_{src}(x_j)$. x_{ij} is passed to a DTP layer and a linear layer to consider geometric information like relative position contained in different type-L vectors in irreps features:

$$x'_{ij} = x_{ij} \otimes_{w(||\vec{r}_{ij}||)}^{DTP} \text{SH}(\vec{r}_{ij}) \quad \text{and} \quad f_{ij} = \text{Linear}(x'_{ij}) \tag{6}$$

x'_{ij} is the tensor product of x_{ij} and spherical harmonics embeddings (SH) of $\vec{r}_{ij}$, with weights parametrized by $||\vec{r}_{ij}||$. f_{ij} considers both semantic and geometric features on source and target nodes in a linear manner and is used to derive attention weights and non-linear messages.

Multi-Layer Perceptron Attention. Attention weights a_{ij} capture how each node interacts with neighboring nodes. a_{ij} are invariant to geometric transformation [19], and therefore, we only use type-0 vectors (scalars) of message f_{ij} denoted as $f_{ij}^{(0)}$ for attention. Note that $f_{ij}^{(0)}$ encodes directional information, as they are generated by tensor products of type-L vectors with $L \geq 0$. Inspired by GATv2 [6], we adopts multi-layer perceptron attention (MLPA) instead of dot product attention (DPA) used in Transformers [58, 30]. In contrast to dot product, MLPs are universal approximators [26, 25, 12] and can theoretically capture any attention patterns [6]. Similar to GAT [59, 6], given $f_{ij}^{(0)}$, we uses one leaky ReLU layer and one linear layer for a_{ij}:

$$z_{ij} = a^{\top}\text{LeakyReLU}(f_{ij}^{(0)}) \quad \text{and} \quad a_{ij} = \text{softmax}_j(z_{ij}) = \frac{\exp(z_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(z_{ik})} \tag{7}$$

where a is a learnable vectors of the same dimension as $f_{ij}^{(0)}$ and z_{ij} is a single scalar. The output of attention is the sum of value v_{ij} multipled by corresponding a_{ij} over all neighboring nodes $j \in \mathcal{N}(i)$, where v_{ij} can be obtained by linear or non-linear transformations of f_{ij} as discussed below.

Non-Linear Message Passing. Values v_{ij} are features sent from one node to another, transformed with input-independent weights. We first split f_{ij} into $f_{ij}^{(L)}$ and $f_{ij}^{(0)}$, where the former consists of type-L vectors with $0 \leq L \leq L_{max}$ and the latter consists of scalars only. Then, we perform non-linear transformation to $f_{ij}^{(L)}$ to obtain non-linear message:

$$\mu_{ij} = \text{Gate}(f_{ij}^{(L)}) \quad \text{and} \quad v_{ij} = \text{Linear}([\mu_{ij} \otimes_{w}^{DTP} \text{SH}(\vec{r}_{ij})]) \tag{8}$$

We apply gate activation to $f_{ij}^{(L)}$ to obtain μ_{ij}. We use one DTP and a linear layer to enable interaction between non-linear type-L vectors, which is similar to how we transform x_{ij} into f_{ij}. Weights w here are input-independent. We can also use $f_{ij}^{(L)}$ directly as v_{ij} for linear messages.

Multi-Head Attention. Following Transformers [58], we can perform h parallel equivariant graph attention functions given f_{ij}. The h different outputs are concatenated and projected with a linear layer, resulting in the final output y_i as illustrated in Fig. 1(b). Note that parallelizing attention functions and concatenating can be implemented with "Reshape".

3.3 Overall Architecture

For completeness, we discuss other modules in Equiformer here.

Embedding. This module consists of atom embedding and edge-degree embedding. For the former, we use a linear layer to transform one-hot encoding of atom species. For the latter, as depicted in the right branch in Fig. 1(c), we first transform a constant one vector into messages encoding local geometry with two linear layer and one intermediate DTP layer and then use sum aggregation to encode degree information [63, 50]. The DTP layer has the same form as that in Eq. 6. We scale the aggregated features by dividing with the squared root of average degrees in training sets so that standard deviation of aggregated features would be close to 1. The two embeddings are summed to produce final embeddings of input 3D graphs.

Radial Basis and Radial Function. Relative distances $||\vec{r}_{ij}||$ parametrizes weights in some DTP layers. To reflect subtle changes in $||\vec{r}_{ij}||$, we represent distances with Gaussian radial basis with learnable mean and standard deviation [48, 51, 31, 50] or radial Bessel basis [21, 20]. We transform radial basis with a learnable radial function to generate weights for those DTP layers [48, 19, 4]. The function consists of two MLPs with layer normalization [2] and SiLU [17, 42] and a final linear layer.

Feed Forward Network. Similar to Transformers, we use two equivariant linear layers and an intermediate gate activation for the feed forward networks in Equiformer.

Output Head. The last feed forward network transforms features on each node into a scalar. We perform sum aggregation over all nodes to predict scalar quantities like energy. Similar to edge-degree embedding, we divide the aggregated scalars with the squared root of average numbers of atoms.

4 Related Works

Here, we focus on equivariant neural networks and discuss other works in appendix.

Methods	Task Units	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	G meV	H meV	R^2 bohr^3	U meV	U_0 meV	ZPVE meV
NMP [23]		.092	69	43	38	.030	.040	19	17	.180	20	20	1.50
SchNet [48]†		.235	63	41	34	.033	.033	14	14	.073	19	14	1.70
Cormorant [1]		.085	61	34	38	.038	.026	20	21	.961	21	22	2.03
LieConv [18]		.084	49	30	25	.032	.038	22	24	.800	19	19	2.28
DimeNet++ [20]†		.044	33	25	20	.030	.023	8	7	.331	6	6	1.21
TFN [54]‡		.223	58	40	38	.064	.101	-	-	-	-	-	-
SE(3)-Transformer [19]		.142	53	35	33	.051	.054	-	-	-	-	-	-
EGNN [47]		.071	48	29	25	.029	.031	12	12	.106	12	11	1.55
SphereNet [37]†		.046	32	23	18	.026	.021	8	6	.292	7	6	1.12
SEGNN [5]		.060	42	24	21	.023	.031	15	16	.660	13	15	1.62
EQGAT [35]		.063	44	26	22	.014	.027	12	13	.257	13	13	1.50
Equiformer		.056	33	17	16	.014	.025	10	10	.227	11	10	1.32

Table 1: **MAE results on QM9 testing set.** † denotes using different training, validation, testing data partitions as mentioned in SEGNN [5]. ‡ denotes results from SE(3)-Transformer [19].

	Energy MAE (eV) ↓					EwT (%) ↑				
Methods	ID	OOD Ads	OOD Cat	OOD Both	Average	ID	OOD Ads	OOD Cat	OOD Both	Average
SchNet [48]†	0.6465	0.7074	0.6475	0.6626	0.6660	2.96	2.22	3.03	2.38	2.65
DimeNet++ [20]†	0.5636	0.7127	0.5612	0.6492	0.6217	4.25	2.48	4.40	2.56	3.42
GemNet-T [31]†	0.5561	0.7342	0.5659	0.6964	0.6382	4.51	2.24	4.37	2.38	3.38
SphereNet [37]	0.5632	0.6682	0.5590	0.6190	0.6024	4.56	2.70	4.59	2.70	3.64
(S)EGNN [5]	0.5497	0.6851	0.5519	0.6102	0.5992	4.99	2.50	4.71	2.88	3.77
SEGNN [5]	0.5310	0.6432	0.5341	0.5777	0.5715	5.32	2.80	4.89	3.09	4.03
Equiformer	0.5088	0.6271	0.5051	0.5545	0.5489	4.88	2.93	4.92	2.98	3.93

Table 2: **Results on OC20 IS2RE <u>validation</u> set.** † denotes results reported by SphereNet [37].

Equivariant GNNs. Equivariant neural networks [54, 32, 60, 19, 39, 56, 4, 29, 49, 47, 57, 5, 53, 35, 40] operate on geometric tensors like type-L vectors to achieve equivariance. The central idea is to use functions of geometry built from spherical harmonics and irreps features to achieve 3D rotational and translational equivariance as proposed in Tensor Field Network (TFN) [54], which generalizes 2D counterparts [61, 10, 11] to 3D Euclidean space. Previous works differ in equivariant operations used in their networks. TFN [54] and NequIP [4] mainly use graph convolution with linear messages, with the latter utilizing extra equivariant gate activations [60]. SE(3)-Transformer [19] adopts an equivariant version of dot product (DP) attention [58, 30], and subsequent works on equivariant Transformers [53, 35] follow the practice of DP attention and use more specialized architectures considering only type-0 and type-1 vectors. SEGNN [5] utilizes non-linear message passing networks [23, 46] for irreps features. The proposed Equiformer incorporates all the advantages of previous works through combining MLP attention with non-linear messages.

5 Experiment

5.1 QM9

Dataset. The QM9 [44, 43] dataset consisting of 134k small molecules, and the goal is to predict their quantum properties such as energy. We follow the data partition used by Cormorant [1], which has 100k, 18k and 13k molecules in training, validation and testing sets. We minimize mean absolute error (MAE) between prediction and normalized ground truth and report MAE on the testing set.

Setting. Please refer to appendix for details on Equiformer architecture and hyper-parameters.

Result. We mainly compare with methods trained with the same data partition and summarize the results in Table 1. Equiformer achieves the best results on 11 out of 12 tasks among models trained with same data partition. The comparison to SEGNN [5], which uses irreps features as Equiformer, demonstrates the effectiveness of combining non-lienar messages with MLP attention. Additionally, Equiformer achieves better results for most of tasks when compared to other equivariant Transformers [19, 35], which suggests a better adaption of Transformers to 3D graphs. Besides, the different data partition as denoted by † in Table 1 has 10% more molecules in the training set and less data in the testing set, and this can benefit some tasks that are more dependent on data partitions.

5.2 OC20

Dataset. The Open Catalyst 2020 (OC20) dataset [9] consists of larger atomic systems, each composed of a small molecule called the adsorbate placed on a large slab called catalyst. The average

Methods	Energy MAE (eV) ↓					EwT (%) ↑				
	ID	OOD Ads	OOD Cat	OOD Both	Average	ID	OOD Ads	OOD Cat	OOD Both	Average
CGCNN [62]	0.6149	0.9155	0.6219	0.8511	0.7509	3.40	1.93	3.10	2.00	2.61
SchNet [48]	0.6387	0.7342	0.6616	0.7037	0.6846	2.96	2.33	2.94	2.21	2.61
DimeNet++ [20]	0.5621	0.7252	0.5756	0.6613	0.6311	4.25	2.07	4.10	2.41	3.21
SpinConv [51]	0.5583	0.7230	0.5687	0.6738	0.6310	4.08	2.26	3.82	2.33	3.12
SphereNet [37]	0.5625	0.7033	0.5708	0.6378	0.6186	4.47	2.29	4.09	2.41	3.32
SEGNN [5]	0.5327	0.6921	0.5369	0.6790	0.6101	5.37	2.46	4.91	2.63	3.84
Equiformer	0.5037	0.6881	0.5213	0.6301	0.5858	5.14	2.41	4.67	2.69	3.73

Table 3: **Results on OC20 IS2RE testing set.**

Methods	Energy MAE (eV) ↓					EwT (%) ↑				
	ID	OOD Ads	OOD Cat	OOD Both	Average	ID	OOD Ads	OOD Cat	OOD Both	Average
GNS [24]	0.54	0.65	0.55	0.59	0.5825	-	-	-	-	-
Noisy Nodes [24]	0.47	0.51	0.48	0.46	0.4800	-	-	-	-	-
Graphormer [50]	0.4329	0.5850	0.4441	0.5299	0.4980	-	-	-	-	-
Equiformer	0.4222	0.5420	0.4231	0.4754	0.4657	7.23	3.77	7.13	4.10	5.56

Table 4: **Results on OC20 IS2RE validation set when IS2RS node-level auxiliary task is adopted during training.** "GNS" denotes the 50-layer GNS trained without Noisy Nodes data augmentation, and "Noisy Nodes" denotes the 100-layer GNS trained with Noisy Nodes.

number of atoms in a system is more than 70, and there are over 50 atom species. The goal is to understand interaction between adsorbates and catalysts through relaxation. An adsorbate is first placed on top of a catalyst to form initial structure (IS). The positions of atoms are updated with forces calculated by density function theory until the system is stable and becomes relaxed structure (RS). The energy of RS, or relaxed energy (RE), is correlated with catalyst activity and therefore a metric for understanding their interaction. We focus on the task of initial structure to relaxed energy (IS2RE), which predicts relaxed energy (RE) given an initial structure (IS). There are 460k, 100k and 100k structures in training, validation, and testing sets, respectively. Performance is measured in MAE and energy within threshold (EwT), the percentage in which predicted energy is within 0.02 eV of ground truth energy. In validation and testing sets, there are four sub-splits containing in-distribution adsorbates and catalysts (ID), out-of-distribution adsorbates (OOD-Ads), out-of-distribution catalysts (OOD-Cat), and out-of-distribution adsorbates and catalysts (OOD-Both).

Setting. We consider two training settings based on whether a node-level auxiliary task [24] is adopted. In the first setting, we minimize MAE between predicted energy and ground truth energy without any node-level auxiliary task. In the second setting, we incorporate the task of initial structure to relaxed structure (IS2RS) as a node-level auxiliary task [24]. In addition to predicting energy, we predict node-wise vectors indicating how each atom should move from initial structure to relaxed structure. Please refer to appendix for details on Equiformer architecture and hyper-parameters.

IS2RE Result without Node-Level Auxiliary Task. We summarize the results under the first setting in Table 2 and Table 3. Compared with state-of-the-art models like SEGNN [5] and SphereNet [37], Equiformer consistently achieves the lowest MAE for all the four sub-splits in validation and testing sets. Note that energy within threshold (EwT) considers only the percentage of predictions close enough to ground truth and the distribution of errors, and therefore improvement in average errors (MAE) would not necessarily reflect that in error distributions (EwT). Similar phenomena can be observed in Table 3, where for "OOD Both" sub-split, SphereNet [37] achieves lower MAE yet lower EwT than SEGNN [5]. We also note that models in Table 2 and 3 are trained by minimizing MAE and therefore comparing MAE in validation and testing sets could mitigate the discrepancy between training objectives and evaluation metrics and that OC20 leaderboard ranks the relative performance of models mainly according to MAE.

IS2RE Result with IS2RS Node-Level Auxiliary Task. We report the result on validation set under the second setting in Table 4. The comparison to GNS demonstrates the improvement from invariant message passing networks to equivariant Transformers. Without any data augmentation, Equiformer still achieves competitive results to GNS trained with Noisy Nodes [24]. Compared to Graphormer [50], Equiformer demonstrates the effectiveness of equivariant features and the proposed equivariant graph attention. Note that Equiformer, with 18 Transformer blocks, is relatively shallow as GNS trained with Noisy Nodes has 100 blocks and Graphormer has 48 Transformer blocks and that deeper networks can typically obtain better results when IS2RS auxiliary task is adopted [24].

Index	Methods: Non-linear message passing	Methods: MLP attention	Methods: Dot product attention	Task Unit	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K
1	✓	✓			.056	33	17	16	.014	.025
2		✓			.061	34	18	17	.015	.025
3			✓		.060	34	18	18	.015	.026

Table 5: **Ablation study results on QM9.**

Index	Methods: Non-linear message passing	Methods: MLP attention	Methods: Dot product attention	Energy MAE (eV) ↓ ID	OOD Ads	OOD Cat	OOD Both	Average
1	✓	✓		0.5088	0.6271	0.5051	0.5545	0.5489
2		✓		0.5168	0.6308	0.5088	0.5657	0.5555
3			✓	0.5386	0.6382	0.5297	0.5692	0.5689
Index	**Methods: Non-linear message passing**	**Methods: MLP attention**	**Methods: Dot product attention**	**EwT (%) ↑ ID**	**OOD Ads**	**OOD Cat**	**OOD Both**	**Average**
1	✓	✓		4.88	2.93	4.92	2.98	3.93
2		✓		4.59	2.82	4.79	3.02	3.81
3			✓	4.37	2.60	4.36	2.86	3.55

Table 6: **Ablation study results on OC20 IS2RE validation set.**

5.3 Ablation Study

We conduct ablation studies on the improvements brought by MLP attention and non-linear messages in the proposed equivariant graph attention. We modify dot product (DP) attention [58, 19] so that it only differs from MLP attention in how attention weights a_{ij} are generated from f_{ij}. Please refer to appendix for details on DP attention. For experiments on QM9 and OC20, unless otherwise stated, we follow the hyper-parameters used in previous experiments.

Result on QM9. The comparison is summarized in Table 5. Non-linear messages improve upon linear messages when MLP attention is used. Similar to what is reported by GATv2 [6], the improvement of replacing DP attention with MLP attention is not very significant. We conjecture that DP attention with linear operations is expressive enough to capture common attention patterns as the numbers of nighboring nodes and atom species are much smaller than those in OC20. However, MLP attention is roughly 7% faster as it directly generates scalar features and attention weights from f_{ij} instead of producing additional key and query irreps features for attention weights.

Result on OC20. We consider the setting of training without IS2RS auxiliary task and use a smaller learning rate 1.5×10^{-3} for DP attention as this improves the performance. We summarize the comparison in Table 6. Non-linear messages consistently improve upon linear messages. In contrast to the results on QM9, MLP attention achieves better performance than DP attention. We surmise this is because OC20 contains larger atomistic graphs with more diverse atom species and therefore requires more expressive attention mechanisms.

6 Conclusion and Broader Impact

In this work, we propose Equiformer, a graph neural network (GNN) combining the strengths of Transformers and equivariant features based on irreducible representations (irreps). With irreps features, we build upon existing generic GNNs and Transformer networks [58, 14, 64, 34, 36] by incorporating equivariant operations like tensor product. We further propose equivariant graph attention, which incorporates multi-layer perceptron attention and non-linear messages. Experiments on QM9 and OC20 demonstrate both the effectiveness of Equiformer and the advantage of equivariant graph attention over typical dot product attention.

The broader impact lies in two aspects. First, Equiformer demonstrates the possibility of adapting Transformers to domains such as physics and chemistry, where data can be represented as 3D atomistic graphs. Second, Equiformer achieves more accurate approximations of quantum properties calculation. We believe there is much more to be gained by harnessing these abilities for productive investigation of molecules and materials relevant to application such as energy, electronics, and pharmaceuticals [9], than to be lost by applying these methods for adversarial purposes like creating hazardous chemicals. Additionally, there are still substantial hurdles to go from the identification of a useful or harmful molecule to its large-scale deployment.

References

[1] Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Conference on Neural Information Processing (NeurIPS)*, 2019.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arxiv preprint arxiv:1607.06450*, 2016.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Conference on Neural Information Processing (NeurIPS)*, 2020.

[4] 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), May 2022.

[5] 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 (ICLR)*, 2022.

[6] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations (ICLR)*, 2022.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Conference on Neural Information Processing (NeurIPS)*, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[9] Lowik Chanussot*, Abhishek Das*, Siddharth Goyal*, Thibaut Lavril*, Muhammed Shuaibi*, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 2021.

[10] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, 2016.

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

[12] George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arxiv preprint arxiv:1810.04805*, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[15] Mildred S Dresselhaus, Gene Dresselhaus, and Ado Jorio. *Group theory*. Springer, Berlin, Germany, 2008 edition, March 2007.

[16] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. 2020.

[17] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *arXiv preprint arXiv:1702.03118*, 2017.

[18] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *arXiv preprint arXiv:2002.12880*, 2020.

[19] Fabian Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Conference on Neural Information Processing (NeurIPS)*, 2020.

[20] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In *Machine Learning for Molecules Workshop, NeurIPS*, 2020.

[21] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

[22] Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. e3nn/e3nn: 2022-04-13, April 2022.

[23] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, 2017.

[24] Jonathan Godwin, Michael Schaarschmidt, Alexander L Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations (ICLR)*, 2022.

[25] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[26] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[27] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861, 2017.

[28] Weile Jia, Han Wang, Mohan Chen, Denghui Lu, Lin Lin, Roberto Car, Weinan E, and Linfeng Zhang. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020.

[29] 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 (ICLR)*, 2021.

[30] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. 2021.

[31] Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In *Conference on Neural Information Processing (NeurIPS)*, 2021.

[32] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems 32*, pages 10117–10126, 2018.

[33] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Conference on Neural Information Processing (NeurIPS)*, 2021.

[34] Cheng-I Lai, Yang Zhang, Alexander H. Liu, Shiyu Chang, Yi-Lun Liao, Yung-Sung Chuang, Kaizhi Qian, Sameer Khurana, David Daniel Cox, and James R. Glass. PARP: Prune, adjust and re-prune for self-supervised speech recognition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Conference on Neural Information Processing (NeurIPS)*, 2021.

[35] Tuan Le, Frank Noé, and Djork-Arné Clevert. Equivariant graph attention networks for molecular property prediction. *arXiv preprint arXiv:2202.09891*, 2022.

[36] Yi-Lun Liao, Sertac Karaman, and Vivienne Sze. Searching for efficient multi-stage vision transformers. *arxiv preprint arxiv:2109.00642*, 2021.

[37] 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 (ICLR)*, 2022.

[38] Denghui Lu, Han Wang, Mohan Chen, Lin Lin, Roberto Car, Weinan E, Weile Jia, and Linfeng Zhang. 86 pflops deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy. *Computer Physics Communications*, 259:107624, 2021.

[39] Benjamin Kurt Miller, Mario Geiger, Tess E. Smidt, and Frank Noé. Relevance of rotationally equivariant convolutions for predicting molecular properties. *arxiv preprint arxiv:2008.08461*, 2020.

[40] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J. Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *arxiv preprint arxiv:2204.05249*, 2022.

[41] Joshua A. Rackers, Lucas Tecot, Mario Geiger, and Tess E. Smidt. Cracking the quantum scaling limit with machine learned electron densities. *arxiv preprint arxiv:2201.03726*, 2022.

[42] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[43] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

[44] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. PMID: 23088335.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[46] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, 2020.

[47] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

[48] 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. In *Conference on Neural Information Processing (NeurIPS)*, 2017.

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

[50] Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets. *arxiv preprint arxiv:2203.04810*, 2022.

[51] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, and C. Lawrence Zitnick. Rotation invariant graph neural networks using spin convolutions. *arxiv preprint arxiv:2106.09575*, 2021.

[52] Anuroop Sriram, Abhishek Das, Brandon M Wood, and C. Lawrence Zitnick. Towards training billion parameter graph neural networks for atomic simulations. In *International Conference on Learning Representations*, 2022.

[53] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations (ICLR)*, 2022.

[54] Nathaniel Thomas, Tess E. 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. *arxiv preprint arXiv:1802.08219*, 2018.

[55] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[56] Raphael J. L. Townshend, Brent Townshend, Stephan Eismann, and Ron O. Dror. Geometric prediction: Moving beyond scalars. *arXiv preprint arXiv:2006.14163*, 2020.

[57] Oliver Thorsten Unke, Mihail Bogojeski, Michael Gastegger, Mario Geiger, Tess Smidt, and Klaus Robert Muller. SE(3)-equivariant prediction of molecular wavefunctions and electronic densities. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Conference on Neural Information Processing (NeurIPS)*, 2021.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing (NeurIPS)*, 2017.

[59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[60] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems 32*, pages 10402–10413, 2018.

[61] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *arxiv preprint arxiv:1612.04642*, 2016.

[62] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14), apr 2018.

[63] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[64] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Conference on Neural Information Processing (NeurIPS)*, 2021.

[65] A. Zee. *Group Theory in a Nutshell for Physicists*. Princeton University Press, USA, 2016.

[66] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120:143001, Apr 2018.

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] We include code related to experiments on QM9 in supplementary. Code reproducing all experiments will be available.
   (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)? [No] As each experiment takes much time, we are not able to run several times. However, the proposed method improves upon previous works by a clear margin, and therefore the improvement is not within noise ranges.
   (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)? [Yes]
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? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(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]