Ai4science

Wu Hualong

HARBIN INSTITUTE OF TECHNOLOGY,SHEN ZHEN

AI for small Molecules

- Molecular Representation learning
  - Invariant Methods: SchNet
  - Equivariant Methods: EGNN, TFN, Painn, **Equiformer**
- Molecular Conformer Generation
  - Learn the Distribution of Low-Energy Geometries: Geodiff
  - Predict the Equilibrium Ground-State Geometry: GTMGC
- Molecule Generation from Scratch: GeoLDM

EQUIFORMER: EQUIVARIANT GRAPH ATTENTION TRANSFORMER FOR 3D ATOMISTIC GRAPHS

ICLR 2023

Yi-Lun Liao, Tess Smidt

Massachusetts Institute of Technology

Outline

introduction

abstract

Despite their widespread success in various domains, Transformer networks have yet to perform well across datasets in the domain of 3D atomistic graphs such as molecules even when 3D-related inductive biases like translational invariance and rotational equivariance are considered. In this paper, author demonstrate that Transformers can generalize well to 3D atomistic graphs and 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).

contribution

- propose a simple and effective architecture by only replacing original operations in Transformers with their equivariant counterparts and including tensor products
- propose a novel attention mechanism called equivariant graph attention, which improves upon typical attention in Transformers through replacing dot product attention with multi-layer perceptron attention and including non-linear message passing

background

E(3) EQUIVARIANCE

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** ($\vec{r} \to -\vec{r}$)

- 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)
- the last two forming O(3)

E(3) EQUIVARIANCE

For learning on 3D atomistic graphs, features and learnable functions should be E(3)-equivariant to geometric transformation acting on position $\vec{r}$.
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

$$f(D_X(g)x) = D_Y(g)f(x) \tag{1}$$

where $D_X(g)$ and $D_Y(g)$ are transformation matrices parametrized by g in X and Y.
as for E(3)-EQUIVARIANCE, $g \in E(3)$

Irreducible Representations

The actions of groups define transformations.
Formally, a transformation acting on vector space X parametrized by group element $g \in G$ is an injective function $T_g : X \to X$.
Formally, a group representation $D : G \to GL(N)$ is a mapping between a group G and a set of $N \times N$ invertible matrices.**How a group is represented depends on the vector space it acts on**.
If there exists a change of basis P in the form of an $N \times N$ matrix such that $P^{-1}D(g)P = D'(g)$ for all $g \in G$, then we say the two group representations are equivalent. If $D'(g)$ is block diagonal, which means that g acts on independent subspaces of the vector space, the representation $D(g)$ is reducible.

Irreducible Representations of SO(3)

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.
We can express any group representation of SO(3) as a direct sum (concatentation) of irreps:

$$D(g) = P^{-1}\left(\bigoplus_i D_{l_i}(g)\right)P = P^{-1}\begin{pmatrix} D_{l_0}(g) & & \\ & D_{l_1}(g) & \\ & & \cdots\cdots \end{pmatrix}P \tag{2}$$

Irreps Feature

$D_L(g)$ 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 $-L \leq m \leq L$.

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 $f_{c,m}^{(L)}$.

Spherical Harmonics.

Euclidean vectors $\vec{r}$ in $\mathbb{R}^3$ can be projected into type-L vectors $f^{(L)}$ by using spherical harmonics (SH)

$$Y^{(L)}: f^{(L)} = Y^{(L)}(\frac{\vec{r}}{||\vec{r}||}). \tag{3}$$

SH are $E(3)$-equivariant with:

$$D_L(g)f^{(L)} = Y^{(L)}(\frac{D_L(g)\vec{r}}{||D_L(g)\vec{r}||}). \tag{4}$$

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 tensor products.

Tensor Product

Tensor products can interact different type-L vectors.The tensor product denoted as $\otimes$ uses Clebsch-Gordan coefficients to combine type-L_1 vector $f^{(L_1)}$ and type-L_2 vector $g^{(L_2)}$ and produces type-L_3 vector $h^{(L_3)}$:

$$h_{m_3}^{(L_3)} = (f^{(L_1)} \otimes g^{(L_2)})_{m_3} = \sum_{m_1=-L_1}^{L_1} \sum_{m_2=-L_2}^{L_2} C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)} f_{m_1}^{(L_1)} g_{m_2}^{(L_2)} \tag{5}$$

where m_1 denotes order and refers to the m_1-th element of $f^{(L_1)}$,Clebsch-Gordan coefficients $C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)}$ are non-zero only when $|L_1 - L_2| \leq L_3 \leq |L_1 + L_2|$.

Tensor Product

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 to typical linear layers.

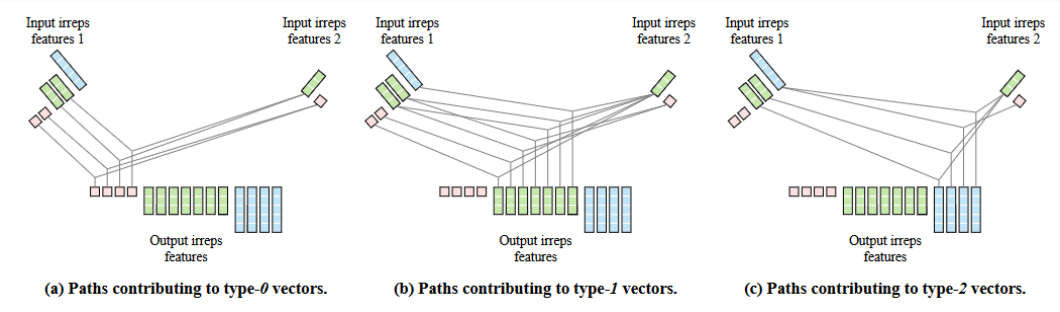
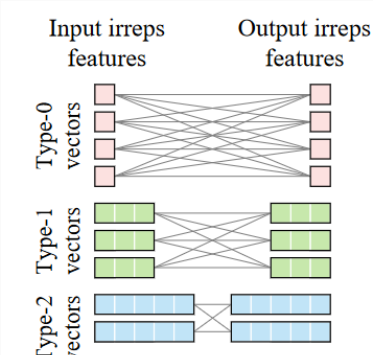


Figure 1: Tensor Product

Equiformer

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

Given input $x \in \mathbb{R}^{N \times C}$,LN calculates the linear transformation of normalized input as:

$$\mathrm{LN}(x) = \left(\frac{x - \mu_C}{\sigma_C} \right) \circ \gamma + \beta \tag{6}$$

As for given input $x \in \mathbb{R}^{N \times C \times (2L+1)}$ of type-L vectors, the output is:

$$LN(x) = \left(\frac{x}{\mathrm{RMS}_C(\mathrm{norm}(x))} \right) \circ \gamma \tag{7}$$

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

Layer Normalization

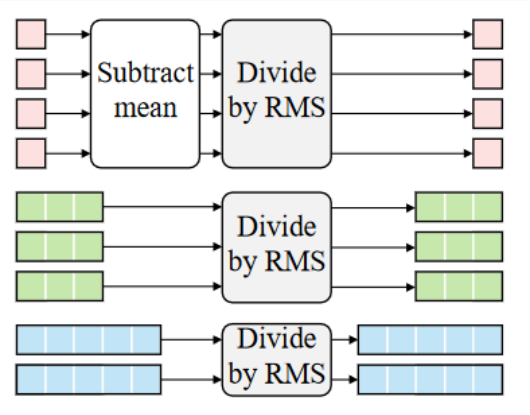
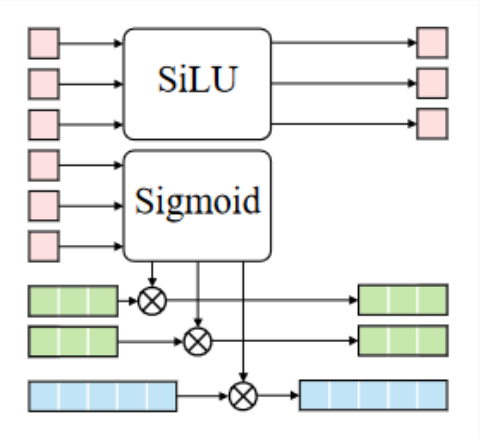


Figure 3: Layer Normalization

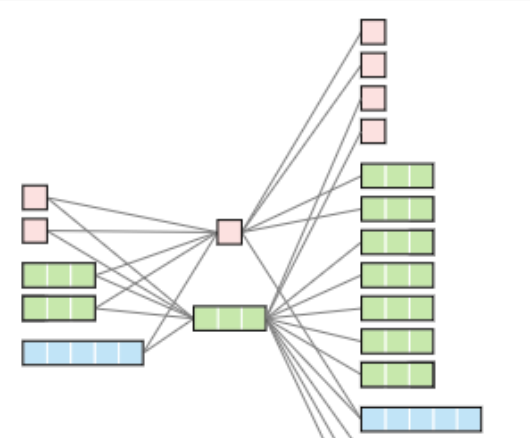
Gate

use the gate activation 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 for equivariance.

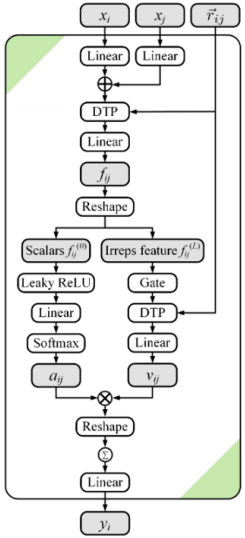


Depth-wise Tensor Product.

The tensor product defines interaction between vectors of different L. To improve its efficiency, we use the depth-wise tensor product (DTP), where one type-L vector in output irreps features depends only on one type-L vector in input irreps features.



equivariant graph attention



$$x_{ij} = \text{Linear}_{dst}(x_i) + \text{Linear}_{src}(x_j)$$

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

$$z_{ij} = a^{\top} \text{LeakyReLU}(f^{(0)}_{ij})$$

$$a_{ij} = \text{softmax}_j(z_{ij}) = \frac{\exp(z_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(z_{ik})}$$

$$\mu_{ij} = \text{Gate}(f^{(L)}_{ij})$$

$$v_{ij} = \text{Linear}([\mu_{ij} \otimes^{DTP}_{w} \text{SH}(\vec{r}_{ij})])$$

Embedding

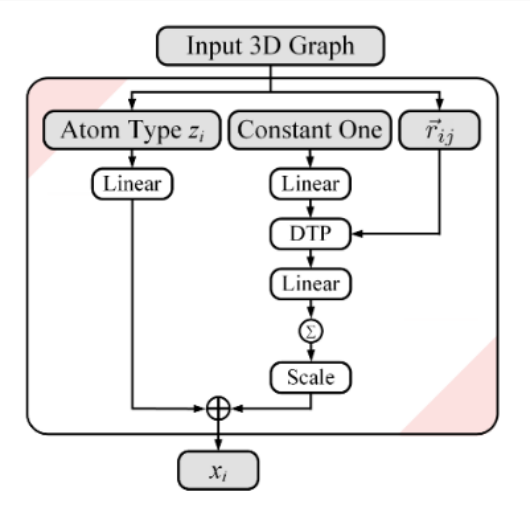
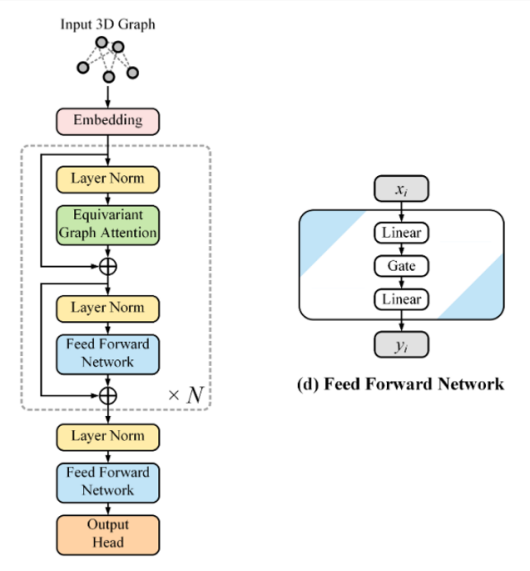


Figure 7: Embedding

overall architecture



experiment

Methods	Task Units	α a_0^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	G meV	H meV	R^2 a_0^2	U meV	U_0 meV	ZPVE meV
NMP (Gilmer et al., 2017)†		.092	69	43	38	.030	.040	19	17	.180	20	20	1.50
SchNet (Schütt et al., 2017)		.235	63	41	34	.033	.033	14	14	.073	19	14	1.70
Cormorant (Anderson et al., 2019)†		.085	61	34	38	.038	.026	20	21	.961	21	22	2.03
LieConv (Finzi et al., 2020)†		.084	49	30	25	.032	.038	22	24	.800	19	19	2.28
DimeNet++ (Gasteiger et al., 2020a)		**.044**	33	25	20	.030	.023	8	7	.331	6	6	1.21
TFN (Thomas et al., 2018)†		.223	58	40	38	.064	.101	-	-	-	-	-	-
SE(3)-Transformer (Fuchs et al., 2020)†		.142	53	35	33	.051	.054	-	-	-	-	-	-
EGNN (Satorras et al., 2021)†		.071	48	29	25	.029	.031	12	12	.106	12	11	1.55
PaiNN (Schütt et al., 2021)		.045	46	28	20	.012	.024	**7.35**	**5.98**	.066	**5.83**	**5.85**	1.28
TorchMD-NET (Thölke & Fabritiis, 2022)		.059	36	20	18	**.011**	.026	7.62	6.16	**.033**	6.38	6.15	1.84
SphereNet (Liu et al., 2022)		.046	32	23	18	.026	**.021**	8	6	.292	7	6	**1.12**
SEGNN (Brandstetter et al., 2022)†		.060	42	24	21	.023	.031	15	16	.660	13	15	1.62
EQGAT (Le et al., 2022)		.053	32	20	16	**.011**	.024	23	24	.382	25	25	2.00
Equiformer		.046	**30**	**15**	**14**	**.011**	.023	7.63	6.63	.251	6.74	6.59	1.26

Index	Methods: Non-linear message passing	Methods: MLP attention	Methods: Dot product attention	Task Unit	α a_0^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	Training time (minutes/epoch)	Number of parameters
1	✓	✓			.046	30	15	14	.011	.023	12.1	3.53M
2		✓			.051	32	16	16	.013	.025	7.2	3.01M
3			✓		.053	32	17	16	.013	.025	7.8	3.35M

remark