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

Partial Summary of existing 3D graph neural networks for molecular representation learning

Equiformerv2 compared to Equiformer

- eSCN convolution
- three architectural improvements

Partial Summary of existing 3D graph neural networks for molecular representation learning

Molecular Representation learning

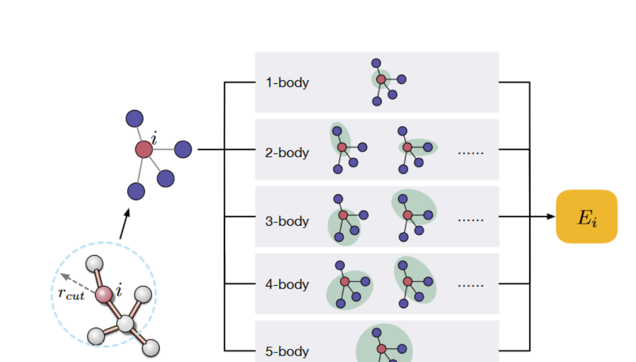
each node has an order-l SE(3)-equivariant node feature. From the perspective of tensor order, existing methods for 3D molecular representation learning can be categorized into:

- Invariant Methods($l = 0$ Scalar Features):
  SchNet, DimeNet, SphereNet
- Equivariant Methods($l = 1$ Vector Features):
  EGNN, PaiNN
- Equivariant Methods($l \geq 1$ Vector Features):
  TFN, SE(3)-Transformer,Equiformer,Equiformerv2

the standard message passing and body order

the standard message passing:

$$
\begin{aligned}
m_i &= \sum_{j \in \mathcal{N}(i)} M\left(h_i, h_j, h_{ij}\right) \\
h_i' &= U\left(h_i, m_i\right)
\end{aligned}
\tag{1}
$$



Invariant Methods ($l = 0$ Scalar Features)

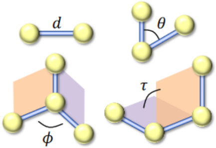
Invariant methods only maintain invariant node, edge, or graph features, which do not change if the input 3D molecule is rotated or translated.

- SchNet considers only **pairwise distances** as edge features h_{ij} in the node-centered message passing schema shown in Equation 1
- DimeNet further considers angles between each pair of edges with edge-centered message passing:

$$\begin{aligned} \boldsymbol{m}_{ji} &= \sum_{k \in \mathcal{N}(j) \setminus \{i\}} M\left(\boldsymbol{h}_{ji}, \boldsymbol{h}_{kj}, \boldsymbol{h}_{kji}\right) \\ \boldsymbol{h}'_{ji} &= U\left(\boldsymbol{h}_{ji}, \boldsymbol{m}_{ji}\right), \end{aligned} \tag{2}$$

- GemNet further considers two-hop dihedral angles, increasing body order to 4 and complexity to $O(nk^3)$
- SphereNet computes local 4-body angles between two planes.

Invariant Methods ($l = 0$ Scalar Features)



Methods	Invariant Geometric Features	Body Order	Complexity
SchNet	Pairwise distances d	2-body	$O(nk)$
DimeNet	d + Angles between edges θ	3-body	$O(nk^2)$
GemNet	d, θ + Angles between 4 nodes τ	4-body	$O(nk^3)$
SphereNet	d, θ + Angles between 4 nodes ϕ	4-body	$O(nk^2)$
ComENet	d, θ, ϕ, τ	4-body	$O(nk)$

Figure 1: Invariant Methods.Here n and k denote the number of nodes and the average degree in a molecule.

Equivariant Methods ($l = 1$ vectors Features)

The first category of equivariant 3D GNNs uses order 1 vectors as intermediate features and propagates messages via a restricted set of operations that guarantee $E(3)$ or $SE(3)$ equivariance.Denote a scalar feature by $s \in \mathbb{R}^d$ and a vector by $\boldsymbol{v} \in \mathbb{R}^{d \times 3}$.operations on a vector v that can ensure equivariance include:

- scaling of vectors $s \odot v$
- summation of vectors $v_1 + v_2$
- linear transformation of vectors Wv
- scalar product $\|v\|^2, v_1 \cdot v_2$
- vector product $v_1 \times v_2$

Equivariant Methods ($l = 1$ vectors Features)

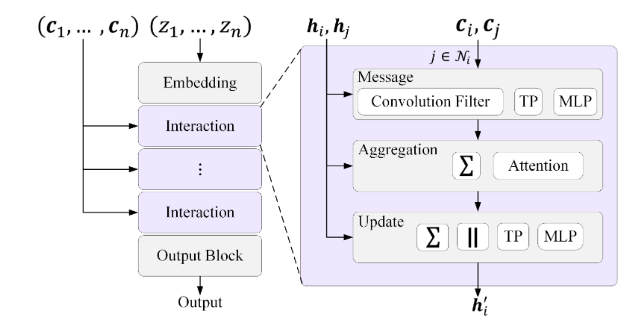
take EGNN as an example:an EGNN layer updates node representation h_i and node coordinate c_i as:

$$
\begin{aligned}
m_{ij} &= \phi_e\left(\boldsymbol{h}_i, \boldsymbol{h}_j, ||\boldsymbol{c}_i - \boldsymbol{c}_j||^2, \boldsymbol{h}_{ij}\right), \\
c_i' &= c_i + C\sum_{j\neq i}(c_i - c_j)\phi_c(m_{ij}), \\
h_i' &= \phi_h\left(\boldsymbol{h}_i, \sum_{j\neq i}\boldsymbol{m}_{ij}\right),
\end{aligned}
\tag{3}
$$

Methods	Scaling $s \odot v$	Summation $v_1 + v_2$	Linear Transformation Wv	Scalar Product $\|v\|^2, v_1 \cdot v_2$	Vector Product $v_1 \times v_2$
EGNN	✓	✓		✓	
ClofNet	✓	✓		✓	
PaiNN	✓	✓	✓	✓	
GVP-GNN	✓	✓	✓	✓	
Vector Neurons	✓	✓	✓	✓	

Equivariant Methods ($l \geq 1$ *TensorFeatures*)

Another category of equivariant methods considers higher-order $l \geq 1$ *features*.
Most existing methods under this category use tensor products (TP) of higherorder spherical tensors to build equivariant representations and follow the general architecture in Figure to update features.
Methods:TFN,SE(3)-Transformer,Equiformer,Equiformerv2



Equiformerv2 compared to Equiformer

Equiformer v2

Equiformer2 works on how equivariant Transformers can be scaled up to higher degrees of equivariant representations. Equiformer2 first start by **replacing SO(3) convolutions in Equiformer with eSCN convolutions**,and then propose three architectural improvements to better leverage the power of higher degrees

- **attention re-normalization**
- **separable S^2 activation**
- **separable layer normalization**

eSCN convolution

Message passing is used to update equivariant irreps features and is typically implemented as SO(3) convolutions. A traditional $SO(3)$ convolution interacts input irrep features $x_{m_i}^{(L_i)}$ and spherical harmonic projections of relative positions $Y_{m_f}^{(L_f)}(\vec{r}_{ts})$ with an $SO(3)$ tensor product with ClebschGordan coefficients $C_{(L_i,m_i),(L_f,m_f)}^{(L_o,m_o)}$. **Since tensor products are compute-intensive, the use of high degree L is limited**. eSCN convolutions are proposed to reduce the complexity of tensor products when they are used in SO(3) convolutions.

eSCN convolution

real form of Spherical harmonics:

$$Y_\ell^m(\theta,\varphi) = \begin{cases} (-1)^m\sqrt{2}\sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-|m|)!}{(\ell+|m|)!}}P_\ell^{|m|}(\cos\theta)\sin(|m|\varphi) & \text{if } m<0 \\ \sqrt{\frac{2\ell+1}{4\pi}}P_\ell^0(\cos\theta) & \text{if } m=0 \\ (-1)^m\sqrt{2}\sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}P_\ell^m(\cos\theta)\cos(m\varphi) & \text{if } m>0 \end{cases} \tag{4}$$

At the north pole, where $\theta = 0$ and ϕ is undefined,all spherical harmonics except those with $m = 0$ vanish:

$$Y_\ell^m(0,\varphi) = Y_\ell^m(\mathbf{z}) = \sqrt{\frac{2\ell+1}{4\pi}}\delta_{m0} \tag{5}$$

eSCN convolution

l	m	$\Phi(\varphi)$	$\Theta(\theta)$	极坐标中的表达式	直角坐标中的表达式	量子力学中的记号
0	0	$\frac{1}{\sqrt{2\pi}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2\sqrt{\pi}}$	$\frac{1}{2\sqrt{\pi}}$	s
1	0	$\frac{1}{\sqrt{2\pi}}$	$\sqrt{\frac{3}{2}}\cos\theta$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{z}{r}$	p_z
1	+1	$\frac{1}{\sqrt{2\pi}}\exp(i\varphi)$	$\frac{\sqrt{3}}{2}\sin\theta$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\cos\varphi$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{x}{r}$	p_x
1	-1	$\frac{1}{\sqrt{2\pi}}\exp(-i\varphi)$	$\frac{\sqrt{3}}{2}\sin\theta$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\sin\varphi$	$\frac{1}{2}\sqrt{\frac{3}{\pi}}\frac{y}{r}$	p_y
2	0	$\frac{1}{\sqrt{2\pi}}$	$\frac{1}{2}\sqrt{\frac{5}{2}}(3\cos^2\theta-1)$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}(3\cos^2\theta-1)$	$\frac{1}{4}\sqrt{\frac{5}{\pi}}\frac{2z^2-x^2-y^2}{r^2}$	$\mathrm{d}_{3z^2-r^2}$
2	+1	$\frac{1}{\sqrt{2\pi}}\exp(i\varphi)$	$\frac{\sqrt{15}}{2}\sin\theta\cos\theta$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\cos\varphi$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{zx}{r^2}$	d_{zx}
2	-1	$\frac{1}{\sqrt{2\pi}}\exp(-i\varphi)$	$\frac{\sqrt{15}}{2}\sin\theta\cos\theta$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\sin\theta\cos\theta\sin\varphi$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{yz}{r^2}$	d_{yz}
2	+2	$\frac{1}{\sqrt{2\pi}}\exp(2i\varphi)$	$\frac{\sqrt{15}}{4}\sin^2\theta$	$\frac{1}{4}\sqrt{\frac{15}{\pi}}\sin^2\theta\cos 2\varphi$	$\frac{1}{4}\sqrt{\frac{15}{\pi}}\frac{x^2-y^2}{r^2}$	$\mathrm{d}_{x^2-y^2}$
2	-2	$\frac{1}{\sqrt{2\pi}}\exp(-2i\varphi)$	$\frac{\sqrt{15}}{4}\sin^2\theta$	$\frac{1}{4}\sqrt{\frac{15}{\pi}}\sin^2\theta\sin 2\varphi$	$\frac{1}{2}\sqrt{\frac{15}{\pi}}\frac{xy}{r^2}$	d_{xy}

Figure 3: Spherical harmonics

eSCN convolution

Tensor products interact type-L_i vector $x^{(L_i)}$ and type-L_f vector $f^{(L_f)}$ to produce type-L_o vector $y^{(L_o)}$ with Clebsch-Gordan coefficients $C^{(L_o,m_o)}_{(L_i,m_i),(L_f,m_f)}$. Clebsch-Gordan coefficients $C^{(L_o,m_o)}_{(L_i,m_i),(L_f,m_f)}$are non-zero only when $|L_i - L_o| \leqslant L_f \leqslant |L_i + L_o|$. Each non-trivial combination of $L_i \otimes L_f \rightarrow L_o$ is called a path, and each path is independently equivariant and can be assigned a learnable weight w_{L_i,L_f,L_o}.

We consider the message m_{ts} sent from source node s to target node t in an SO(3) convolution.The L_o-th degree of m_{ts} can be expressed as:

$$m_{ts}^{(L_o)} = \sum_{L_i,L_f} w_{L_i,L_f,L_o} \left(x_s^{(L_i)} \otimes_{L_i,L_f}^{L_o} Y^{(L_f)}(\hat{r}_{ts}) \right) \tag{6}$$

Therefore,By choosing a specific R, we can reduce the cost of computing equation above substantially.

eSCN convolution

Specifically, if select a rotation matrix $\mathbf{R}_{ts}$ so that $\mathbf{R}_{ts} \cdot \hat{\mathbf{r}}_{ts} = (0, 0, 1)$, the $\mathbf{Y}(\mathbf{R}_{st} \cdot \hat{\mathbf{r}}_{st})$ become sparse:

$$\mathbf{Y}_m^{(l)}(\mathbf{R}_{ts} \cdot \hat{\mathbf{r}}_{ts}) \propto \delta_m^{(l)} = \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{if } m \neq 0 \end{cases} \tag{7}$$

$$\begin{aligned} m_{ts}^{(L_o)} &= \left(D^{(L_o)}(R_{ts})\right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \left(D^{(L_i)}(R_{ts}) x_s^{(L_i)} \otimes_{L_i, L_f}^{L_o} Y^{(L_f)}(R_{ts} \hat{r}_{ts})\right) \\ &= \left(D^{(L_o)}\right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\sum_{m_i, m_f} \left(D^{(L_i)} x_s^{(L_i)}\right)_{m_i} C_{(L_i, m_i),(L_f, m_f)}^{(L_o, m_o)} \left(Y^{(L_f)}(R_{ts} \hat{r_{ts}})\right)_{m_f}\right) \\ &= \left(D^{(L_o)}\right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\sum_{m_i} \left(\tilde{x}_s^{(L_i)}\right)_{m_i} C_{(L_i, m_i),(L_f, 0)}^{(L_o, m_o)}\right) \end{aligned} \tag{8}$$

eSCN convolution

Additionally, given $m_f = 0$ Clebsch-Gordan coefficients $C^{(L_o,m_o)}_{(L_i,m_i),(L_f,0)}$ are sparse and are non-zero only when $m_i = \pm m_o$,this further simplifies equation 8:

$$m^{(L_o)}_{ts} = \left(D^{(L_o)}\right)^{-1} \sum_{L_i,L_f} w_{L_i,L_f,L_o} \bigoplus_{m_o} \left(\left(\tilde{x}^{(L_i)}_s\right)_{m_o} C^{(L_o,m_o)}_{(L_i,m_o),(L_f,0)} + \left(\tilde{x}^{(L_i)}_s\right)_{-m_o} C^{(L_o,m_o)}_{(L_i,-m_o),(L_f,0)} \right) \tag{9}$$

re-ordering the summations and concatenation:

$$\left(D^{(L_o)}\right)^{-1} \sum_{L_i} \bigoplus_{m_o} \left(\left(\tilde{x}^{(L_i)}_s\right)_{m_o} \sum_{L_f} \left(w_{L_i,L_f,L_o} C^{(L_o,m_o)}_{(L_i,m_o),(L_f,0)} \right) + \right.$$

$$\left. \left(\tilde{x}^{(L_i)}_s\right)_{-m_o} \sum_{L_f} \left(w_{L_i,L_f,L_o} C^{(L_o,m_o)}_{(L_i,-m_o),(L_f,0)} \right) \right) \tag{10}$$

eSCN convolution

Instead of using learnable parameters for w_{L_i,L_f,L_o},eSCN proposes to parametrize $\tilde{w}_{m_o}^{(L_i,L_o)}$ and $\tilde{w}_{-m_o}^{(L_i,L_o)}$ as below:

$$\tilde{w}_{m_o}^{(L_i,L_o)} = \sum_{L_f} w_{L_i,L_f,L_o} C^{(L_o,m_o)}_{(L_i,m_o),(L_f,0)} = \sum_{L_f} w_{L_i,L_f,L_o} C^{(L_o,-m_o)}_{(L_i,-m_o),(L_f,0)} \quad \text{for } m >= 0$$

$${}^{(L_i,L_o)}_{-m_o} = \sum_{L_f} w_{L_i,L_f,L_o} C^{(L_o,-m_o)}_{(L_i,m_o),(L_f,0)} = -\sum_{L_f} w_{L_i,L_f,L_o} C^{(L_o,m_o)}_{(L_i,-m_o),(L_f,0)} \quad \text{for } m > 0(11)$$

eSCN convolution

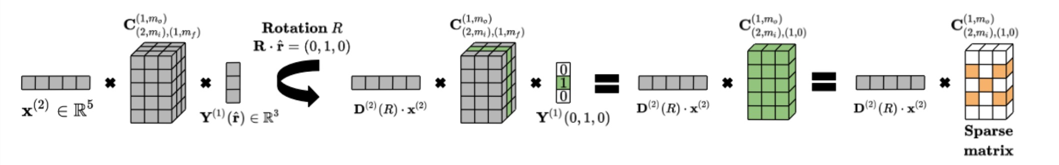
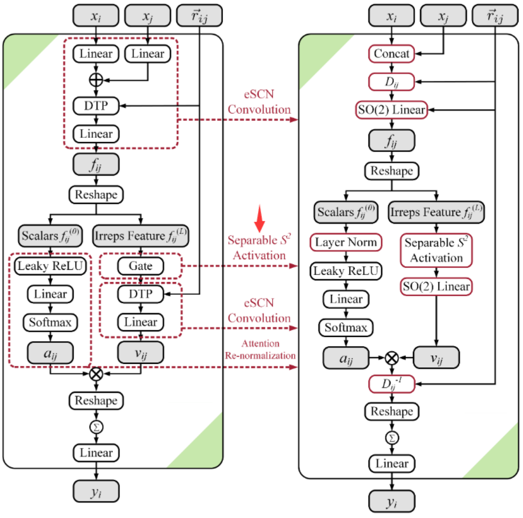


Figure 4: Visual representation of the simplified tensor product

attention re-normalization



(b) Equivariant Graph Attention

separable S^2 activation

The gate activation used by Equiformer:

- applies sigmoid activation to scalar features to obtain non-linear weights and then multiply irreps features of degree >0 with nonlinear weights
- only accounts for the interaction from vectors of degree 0 to those of degree>0 and can be sub-optimal when we scale up L_{max}

S^2 activation:

- first converts vectors of all degrees to point samples on a sphere for each channel, applies unconstrained functions F to those samples, and finally convert them back to vectors
- given an input irreps feature $x \in \mathbb{R}^{(L_{max}+1)^2 \times C}$,the output is $y = G^{-1}(F(G(x)))$

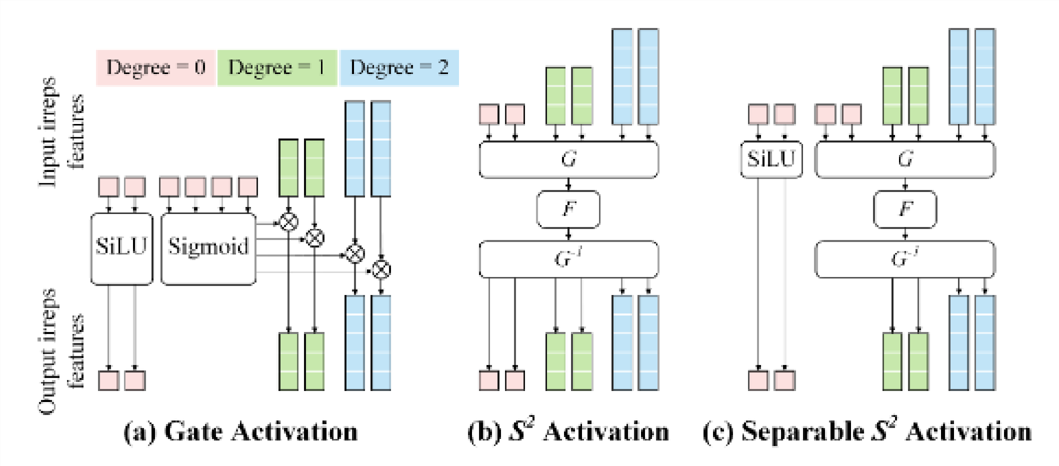


Figure 6: Illustration of different activation functions. G denotes conversion from vectors to point samples on a sphere, F can typically be a SiLU activation or MLPs, and G^{-1} is the inverse of G.

separable layer normalization

Equivariant layer normalization used by Equiformer:

- normalizes vectors of different degrees independently
- potentially ignores the relative importance of different degrees since the relative magnitudes between different degrees become the same after the normalization

propose separable layer normalization (SLN),motivated by the separableS^2 activation:

- separates normalization for vectors of degree 0 and those of degrees>0
- For $L = 0, y^{(0)} = \gamma^{(0)} \circ \left(\frac{x^{(0)} - \mu^{(0)}}{\sigma^{(0)}}\right) + \beta^{(0)}$
  $\mu^{(0)} = \frac{1}{C} \sum_{i=1}^{C} x_{0,i}^{(0)}$ and $\sigma^{(0)} = \sqrt{\frac{1}{C} \sum_{i=1}^{C} (x_{0,i}^{(0)} - \mu^{(0)})^2}$
- For $L > 0$,$y^{(L)} = \gamma^{(L)} \circ \left(\frac{x^{(L)}}{\sigma^{(L>0)}}\right)$,$\sigma^{(L>0)} = \sqrt{\frac{1}{L_{max}} \sum_{L=1}^{L_{max}} \left(\sigma^{(L)}\right)^2}$ and
  $\sigma^{(L)} = \sqrt{\frac{1}{C} \sum_{i=1}^{C} \frac{1}{2L+1} \sum_{m=-L}^{L} \left(x_{m,i}^{(L)}\right)^2}$

separable layer normalization

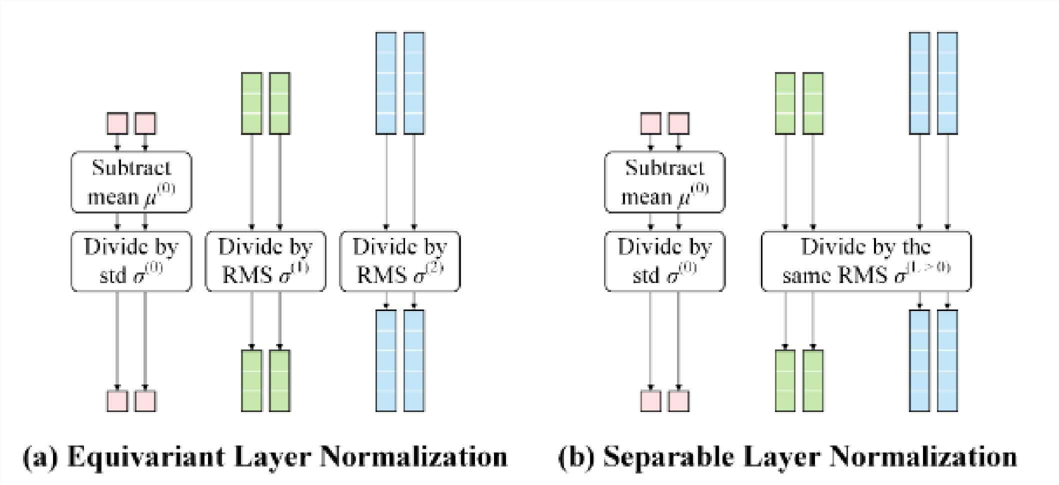


Figure 7: Illustration of how statistics are calculated in different normalizations. "std" denotes standard deviation, and "RMS" denotes root mean square.

some results

Index	Attention re-normalization	Activation	Normalization	Epochs	forces	energy
1	✗	Gate	LN	12	21.85	286
2	✓	Gate	LN	12	21.86	279
3	✓	S^2	LN	12	didn't converge	
4	✓	Sep. S^2	LN	12	20.77	285
5	✓	Sep. S^2	SLN	12	20.46	285
6	✓	Sep. S^2	LN	20	20.02	276
7	✓	Sep. S^2	SLN	20	19.72	278
8	eSCN baseline			12	21.3	294

(a) Architectural improvements. Attention re-normalization improves energies, and separable S^2 activation ("Sep. S^2") and separable layer normalization ("SLN") improve forces.

		eSCN		EquiformerV2	
L_{max}	Epochs	forces	energy	forces	energy
6	12	21.3	294	20.46	285
6	20	20.6	290	19.78	280
6	30	20.1	285	19.42	278
8	12	21.3	296	20.46	279
8	20	-	-	19.95	273

(b) Training epochs. Training for more epochs consistently leads to better results.

	eSCN		EquiformerV2	
L_{max}	forces	energy	forces	energy
4	22.2	291	21.37	284
6	21.3	294	20.46	285
8	21.3	296	20.46	279

(c) Degrees L_{max}. Higher degrees are consistently helpful.

	eSCN		EquiformerV2	
M_{max}	forces	energy	forces	energy
2	21.3	294	20.46	285
3	21.2	295	20.24	284
4	21.2	298	20.24	282
6	-	-	20.26	278

(d) Orders M_{max}. Higher orders mainly improve energy predictions.

	eSCN		EquiformerV2	
Layers	forces	energy	forces	energy
8	22.4	306	21.18	293
12	21.3	294	20.46	285
16	20.5	283	20.11	282

(e) Number of Transformer blocks. Adding more blocks can help both force and energy predictions.