Appendix to the paper "HAGO-Net: Hierarchical Geometric Massage Passing for Molecular Representation Learning"

Anonymous submission

A1. Proofs of Results

In this section, we present the proofs of the lemmas and theorems in the main paper.

A1.1. Proof of Theorem 4

Theorem 4 *For a chiral molecule $\mathcal{A}$ and its enantiomer $\mathcal{A}'$, ROT($\mathcal{A}$) = $-$ROT($\mathcal{A}'$). In other words, the ROT is a chiral identity in Definition 3 (in the main paper).*

Proof. Let us consider two enantiomers, denoted as $\mathcal{A}$ and $\mathcal{A}'$, of a chiral molecule. As $\mathcal{A}$ and $\mathcal{A}'$ are mirror images of each other, we can establish a 3D rectangular coordinate system (x, y, z). Without loss of generality, we can assume that $\mathcal{A}$ and $\mathcal{A}'$ are symmetric with respect to the xy-plane. In this way, for one atom a_i belonging to $\mathcal{A}$ with coordinates (x_i, y_i, z_i), the coordinates of its corresponding mirror atom a_i' in $\mathcal{A}'$ would be $(x_i, y_i, -z_i)$.

Then, we calculate the vector $\vec{e_i}$ from the chiral atom a_{chiral} with coordinates $(x_{\text{chiral}}, y_{\text{chiral}}, z_{\text{chiral}})$ to any atom a_i in $\mathcal{A}$, and the corresponding vector $\vec{e_i}'$ in the mirroring molecule $\mathcal{A}'$,

$$\begin{aligned} \vec{e_i} &= (x_i - x_{\text{chrial}}, y_i - y_{\text{chrial}}, z_i - z_{\text{chrial}}) \\ &= (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i) \\ \vec{e_i}' &= (x_i - x_{\text{chrial}}, y_i - y_{\text{chrial}}, -z_i - (-z_{\text{chrial}})) \\ &= (\tilde{x}_i, \tilde{y}_i, -\tilde{z}_i) \end{aligned} \tag{1}$$

With Eq. 1, the normal vector $\vec{n}$ of the plane spanned by vectors $\vec{e_j}$, $\vec{e_k}$ in $\mathcal{A}$, and the mirroring one $\vec{n}'$ in $\mathcal{A}'$, can be written as follows,

$$\begin{aligned} \vec{n} &= \vec{e_j} \times \vec{e_k} \\ &= (\tilde{y}_j\tilde{z}_k - \tilde{y}_k\tilde{z}_j, \tilde{x}_k\tilde{z}_j - \tilde{x}_j\tilde{z}_k, \tilde{x}_j\tilde{y}_k - \tilde{x}_k\tilde{y}_j) \\ \vec{n}' &= \vec{e_j}' \times \vec{e_k}' \\ &= (\tilde{y}_k\tilde{z}_j - \tilde{y}_j\tilde{z}_k, \tilde{x}_j\tilde{z}_k - \tilde{x}_k\tilde{z}_j, \tilde{x}_j\tilde{y}_k - \tilde{x}_k\tilde{y}_j) \end{aligned} \tag{2}$$

We also have $\vec{e_l} = (\tilde{x}_l, \tilde{y}_l, \tilde{z}_l)$ and $\vec{e_l}' = (\tilde{x}_l, \tilde{y}_l, -\tilde{z}_l)$. As ROT($\mathcal{A}$) = sign($\langle \vec{n}, \vec{e_l} \rangle$) = sign($\langle \vec{e_j} \times \vec{e_k}, \vec{e_l} \rangle$) and ROT($\mathcal{A}'$) = sign($\langle \vec{n}', \vec{e_l}' \rangle$) = sign($\langle \vec{e_j}' \times \vec{e_k}', \vec{e_l}' \rangle$). It is evident that ROT($\mathcal{A}$) = $-$ROT($\mathcal{A}'$) by combining with Eq. (2). This leads us to our final conclusion. □

A1.2. Proof of Lemma 6

Lemma 6 *If an invariant function f takes positions as input, its derivative w.r.t. position is rotation equivariant and translation invariant.*

Proof. Since f is invariant to translations and rotations, we can express this as follows:

$$f(p + \tau) = f(p), \tag{3}$$

$$f(Ap) = f(p), \tag{4}$$

where p signifies a position, τ represents a translation, and A is a rotation matrix, *i.e.*, a transformation matrix that is used to perform a rotation in Euclidean space.

According to the chain rule of the derivative of the composition of two functions, we apply the derivative to both sides of Eq. 3,

$$\frac{df(p+\tau)}{d(p+\tau)} \cdot \frac{d(p+\tau)}{dp} = \frac{df(p)}{dp}. \tag{5}$$

This simplifies to,

$$f'(p + \tau) = f'(p), \tag{6}$$

which shows that the derivative of a translation-invariant function is also translation-invariant.

In the same way, we apply the derivative to both sides of Eq. 4,

$$\begin{aligned} \frac{df(Ap)}{d(Ap)} \cdot \frac{d(Ap)}{dp} &= \frac{df(p)}{dp} \\ \frac{df(Ap)}{d(Ap)} \cdot A &= \frac{df(p)}{dp} \\ f'(Ap) &= f'(p) \cdot A^{\mathrm{T}}. \end{aligned} \tag{7}$$

For rotation matrix, $A^{-1} = A^{\mathrm{T}}$. Pre-multiplying by the rotation matrix Ap results in a rotation equivalent to the rotation produced by post-multiplying by the transpose of the rotation matrix $f'(p)A^{\mathrm{T}}$. Thus, Eq. 7 shows that the derivative of a rotation-invariant function is rotation-equivalent.

□

Table 1: Geometric quantities used in HAGO-Graph

Layer	Geometric quantity	Geometric basis function
1-**body** layer, *i.e.*, atoms:		
intra-n-body geometry	atomic number	linear transformation
inter-n-body geometry	distance	radial basis function
2-**body** layer, *i.e.*, atomic pairs:		
intra-n-body geometry	length	radial basis function
inter-n-body geometry	plane angle	cubic basis function
3-**body** layer, *i.e.*, atomic triplets:		
intra-n-body geometry	triangle area, plane angle	cubic basis function
inter-n-body geometry	dihedral angle	spherical Bessel basis function
4-**body** layer, *i.e.*, atomic quintuplets:		
intra-n-body geometry	tetrahedron volume, ROT	linear transformation

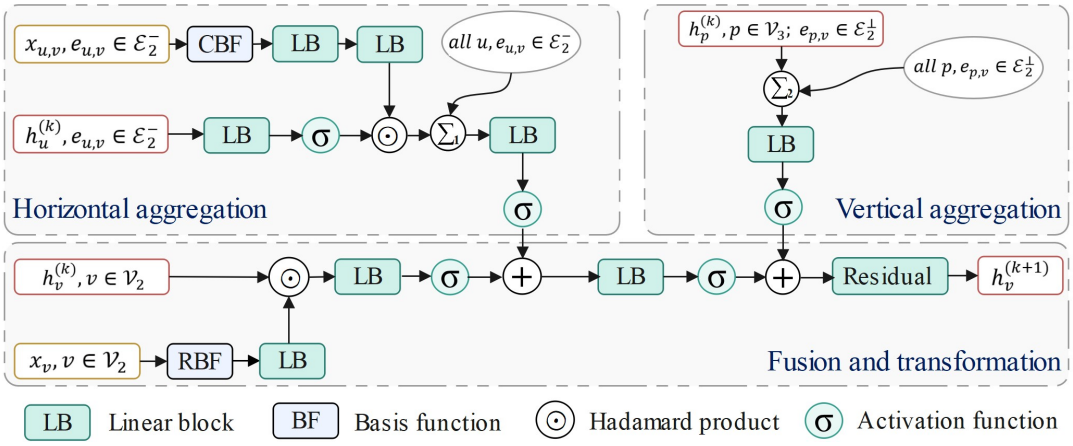

Figure 1: The main architecture of the 2-body layer in HAGO-Net.

A1.3. Proof of Theorem 7

We first present Lemma A1 and Lemma A2, and then we prove Theorem 7 based on the two lemmas.

Lemma A1 *The representation learned by HAGO-Net is invariant to translation and rotation transformations.*

Proof. In this paper, we transform atomic positions $\mathcal{P}$ into translation- and rotation-invariant geometric quantities to construct HAGO-Graph, as shown in Table 1. That is, HAGO-Graph is translation- and rotation-invariant. As the input of HAGO-Net is HAGO-Graph, according to Lemma A1, HAGO-Net is invariant to translation and rotation transformations. □

Lemma A2 *For a composite function $\phi_1 \circ \phi_2(\mathcal{P})$ taking spatial positions $\mathcal{P}$ as input, if $\phi_2(\mathcal{P})$ is translation- and rotation-invariant, the entire composite function $\phi_1 \circ \phi_2(\mathcal{P})$ is invariant to translation and rotation transformations.*

Proof. Let $\mathcal{P}'$ be an isomorphism of $\mathcal{P}$ after translation and rotation transformations. Given $\phi_2(\mathcal{P})$ is translation- and rotation-invariant, we have $\phi_2(\mathcal{P}) = \phi_2(\mathcal{P}')$. Therefore, the entire composite function $\phi_1 \circ \phi_2(\mathcal{P})$ is translation- and rotation-invariant since $\phi_1 \circ \phi_2(\mathcal{P}) = \phi_1 \circ \phi_2(\mathcal{P}')$. □

Theorem 7 The representation of *D*HAGO-Net is rotation equivariant and translation invariant.

Proof. The proof is straightforward by combining with Lemma A1 and Lemma A2. □

A2. Geometric Quantities in HAGO-Graph

We summarize the geometric quantities used at different layers of the HAGO-Graph in Table 1.

Table 1 also presents the geometric basis functions used to extract information from the geometric quantities. We only use radial basis function (RBF), basis function (CBF), and low-order spherical harmonic functions. There may be more powerful architectures to implement HAGO-MPS with more basis functions, such as irreps features and high-order spherical harmonics.

A3. Architecture of HAGO-Net

We illustrate the architecture of HAGO-Net by taking a typical 2-body layer as an example, as shown in Fig. 1. The architecture consists of three main parts, horizontal aggregation, vertical aggregation, and fusion and transformation. For a node $u \in \mathcal{V}_n$ at the n-th layer of $\mathcal{G}$, let $\boldsymbol{h}_u^{(k)}$ denote

the representation vector of node v at k-th iteration. LB denotes to a linear block with one linear layer, and the adopted activation function is SiLU (Sigmoid Linear Unit).

The horizontal aggregation aims at message passing between nodes at 2-body scale, thereby extracting geometric information in 2-bodies. The CBF denotes the cubic basis function. $\boldsymbol{x}_{u,v}$ denotes the inter-2-body geometric feature between target node u and neighbor node v, which is then encoded by CBF and LBs. We use a Hadamard product to integrate the geometric feature into the representation of each neighbor node v. After aggregating all neighbors of node v at the 2-body scale, the node representation goes through an LB block and an activation function.

The vertical aggregation aims to interact with the upper scale, *i.e.,* 3-body scale in this example. The massages $\boldsymbol{h}_p^{(k)}$ from the 3-body scale are aggregated and then encoded by an LB block with an activation function.

Fusion and transformation aims to fuse the massages from the neighbors and from the upper scale, and then conduct transformation. The RBF denotes radial basis function, and $\boldsymbol{x}_v$ is the intra-2-body geometric feature. The geometric massage is integrated by using a Hadamard product.

A4. Additional Experiments

A4.1. Dynamics prediction on OC20 dataset

We further validate HAGO-Net using the IS2RE (Initial Structure to Relaxed Energy) task on the OC20 dataset (Chanussot et al. 2021). The Open Catalyst 2020 (OC20) dataset, a recently introduced extensive dataset, is dedicated to catalyst discovery and optimization. In the IS2RE task, the input is the initial structure of the molecules, and the goal is to predict the relaxation energy. The prediction errors in terms of MAE are summarized in Table 2, where our proposed HAGO-Net achieves competitive performance, compared to the recent baselines. The test set sampled from the same distribution as the training data is labeled as 'in domain' (ID). 'OOD ADs' denotes the test set with unseen adsorbates, 'OOD Cat' denotes the test set with unseen catalysts, and 'OOD Both' indicates instances where both adsorbates and catalysts are out of the domain of training data.

Table 2: Results on IS2RE task on OC20 dataset in terms of energy MAE(eV).

Method	ID	OOD Ads	OOD Cat	OOD Both
CGCNN	0.6203	0.7426	0.6001	0.6708
SchNet	0.6465	0.7074	0.6475	0.6626
DimeNet++	0.5636	0.7127	0.5612	0.6492
SphereNet	0.5632	<u>0.6682</u>	<u>0.5590</u>	**0.6190**
SEGNN	**0.5327**	0.6921	**0.5369**	0.6790
HAGO-Net	<u>0.5571</u>	**0.6532**	0.5604	<u>0.6430</u>

A4.2. Ablation study

The proposed HAGO-Graph and HAGO-MPS effectively model different scales of molecular geometry, thereby achieving a complete representation. To investigate the contributions of each layer within the HAGO-Graph, we sequentially remove the layers starting with the 4-body layer, followed by the 3-body layer, and finally, the 2-body layer. We further investigate the individual contribution of horizontal aggregation and vertical aggregation in HAGO-MPS by removing the horizontal aggregation in 3-body layer and 2-body layer from the HAGO-Net.

We conduct this ablation study on the LUMO property using a subset of 24,000 molecules randomly sampled from QM9 dataset. We split the dataset as 16,000 for training, 4,000 for validation, and 4,000 for testing. As shown in Table 3, each layer of the HAGO-graph contributes to the improvement of HAGO-Net's performance. Significantly, the horizontal aggregation decreases the test MAE by 45%.

Table 3: Results of ablation study

Model	Test MAE
HAGO-Net	52.2
Remove 4-body layer	53.3
+ Remove 3-body layer	55.2
+ Remove 2-body layer	224.8
Remove horizontal aggregation	95.0

A4.3. Time and space complexity Test

We conducted an empirical evaluation of the time and space complexity of HAGO-Net by comparing it with three baseline models. Time complexity was measured by using both training and inference time, and space complexity was measured by GPU memory used in model training. The experiments is to fit the energy and force on 1,000 ethanol samples in MD17 dataset, thereby evaluating the complexity of both HAGO-Net and DHAGO-Net. It's worth noting that all models operated on the same Tesla V100s GPU with an identical computational environment.

The results of these comparisons are presented in Table 4. Remarkably, both HAGO-Net and DimeNet++ show efficiency in terms of time and memory usage, compared to equivariant baselines, NequIP and GemNet. The experiment reveals that HAGO-Net's complexity is comparable to that of DimeNet, which exhibits $O(n^2)$ complexity. In terms of efficiency, HAGO-Net requires only two-thirds of the training time demanded by GemNet and reduces inference time by 90% compared to NequIP.

A4.4. Evaluation on randomized test

To evaluate the variance induced by random factors during training, we trained HAGO-Net three times on three properties of QM9 dataset. We calculated both the mean and standard deviation for each prediction, as detailed in Table 5. These results demonstrate that the variances are minuscule, highlighting that the enhancements of our method are not because of randomness, but are attributed to a more powerful model for molecular representation learning.

Table 4: Results of complexity measurement experiments.

Method	Train time(seconds/epoch)	Inference time (seconds/epoch)	GPU memory usage (MB)
HAGO-Net	180	4.5	1383
DimeNet++	165	4	1305
NequIP	292	38	2275
GemNet	487	18	2209

Table 5: Results of HAGO-Net on QM9 dataset in terms of MAE (MEAN and STD

Property	μ	α	ϵ_{HOMO}	ϵ_{LUMO}
MEAN	0.023	0.044	22.6	19.1
STD	± 0.0002	± 0.0003	± 0.2	± 0.1
Property	$\Delta\epsilon$	R^2	ZPVE	$U0$
MEAN	30.4	0.230	1.23	5.89
STD	± 0.2	± 0.004	± 0.01	± 0.02
Property	U	H	G	c_v
MEAN	6.11	6.33	7.30	0.021
STD	± 0.03	± 0.02	± 0.03	± 0.001

A5. Details in Experiments

A5.1. Hardware and software settings

HAGO-Net is implemented using PyTorch-Geometric 2.0.4 and is built upon PyTorch 1.11.0. Our models are trained and evaluated on NVIDIA Tesla V100 GPUs, each with 32GB memory. The used computer is Inspur-NF5468M6, equipped with 475GB of RAM. All programs operate on Ubuntu 18.04.6 within a virtual environment established by Conda 4.12.0. The file describing the detailed environment requirements for HAGO-Net is provided in the code Appendix.

A5.2. Description of molecular property

In order to understand the experimental results on the QM9 dataset, we provide a brief description of each molecular property in the QM9 dataset in Table 6. The properties U_0, U, H, and G are highly related to atoms in a molecule. While, the properties ϵ_{HOMO}, ϵ_{LUMO}, and $\Delta\epsilon$ globally characterize a molecule.

A5.3. Model configuration

We provide detailed settings of the hyper-parameters utilized for HAGO-Net in our experiments. In every experiment, the size of HAGO-Net's hidden layer channel is uniformly set to 128. Regarding the learning rate strategy, we adopt StepLR throughout the training phases. StepLR modulates the learning rate by a decay ratio after a predetermined number of epochs. Specific hyper-parameters, inclusive of their values or search spaces, for tasks of property prediction on the QM9 dataset, force prediction on the MD17 dataset, R/S classification, and the OC20 IS2RE task are delineated in Table 8, Table 9, Table 7, and Table 10, respectively.

References

Chanussot, L.; Das, A.; Goyal, S.; Lavril, T.; Shuaibi, M.; Riviere, M.; Tran, K.; Heras-Domingo, J.; Ho, C.; Hu, W.; Palizhati, A.; Sriram, A.; Wood, B.; Yoon, J.; Parikh, D.; Zitnick, C. L.; and Ulissi, Z. 2021. Open Catalyst 2020 (OC20) Dataset and Community Challenges. *ACS Catalysis*.

Table 6: Description of molecular properties in QM9 dataset

Property	Description
μ, the norm of the dipole moment	The dipole moment approximates the electric field far from a molecule. The magnitude of the dipole moment indicates the degree of charge separation in the molecule, with larger values suggesting greater polarity, the anisotropy in charge distribution.
α, the norm of the static polarizability	The polarizability of a molecule quantifies a molecule's propensity to induce a dipole moment under an external field. This characteristic correlates with the prominence of Van der Waals interactions.
ϵ_{HOMO}, the highest occupied molecular orbital	ϵ_{HOMO} is the energy of the highest occupied electronic state at zero temperature. The allowed states that electrons can occupy in a molecule are discrete, and no two electrons may occupy the same state. Electrons stack from lowest to highest energy.
ϵ_{LUMO}, the lowest unoccupied molecular orbital	ϵ_{LUMO} is the energy of the lowest occupied electronic state at zero temperature.
$\Delta\epsilon$, the electron energy gap	The energy gap between ϵ_{LUMO} and ϵ_{HOMO}. It represents the minimum energy transition that can take place when an electron is excited from an occupied state to an unoccupied one.
R^2, the electronic spatial extent	R^2 is the second moment of the charge distribution. It characterizes the spatial distribution of electrons in the molecule.
ZPVE, the zero point vibrational energy,	Vibrational energy at zero temperature.
$U0$, the atomization energy at $0K$,	Energy required to break up the molecule into its constituent atoms at absolute zero.
U, the atomization energy at room temperature	Energy required to break up the molecule into its constituent atoms at room temperature.
H, the enthalpy of atomization at room temperature	Enthalpy required to break up the molecule into its constituent atoms at room temperature under fixed pressure.
G, the free energy of atomization	Energy required to break up the molecule into its constituent atoms at fixed temperature and pressure.
C_v, the heat capacity at room temperature	C_v is the amount of heat that must be added to one unit of mass of the molecule in order to cause an increase of one unit in temperature at room temperature.

Table 7: Values/search space for hyper-parameters on R/S classification.

Hyperparameter	Search space
Cutoff	5
Envelope exponent	5
Number of interaction layers	3, 4, 5
Number of RBFs	6
Number of spherical harmonics	7
Out embedding channels	64, 128, 256
Readout MLP Hidden Size	32, 64, 128, 256
Number of readout layers	1, 2, 3, 4
Batch size	8, 16, 32
Initial learning rate	5e-5, 5e-4
Max epochs	100

Table 8: Values/search space for hyper-parameters on QM9

Hyperparameter	Search space
Cutoff	5
Envelope exponent	5
Number of interaction layers	3, 4, 5
Number of RBFs	6
Number of spherical harmonics	2, 3, 5, 7
Batch size	16, 32
Initial learning rate	1e-4, 5e-4, 1e-3
Learning rate strategy	StepLR
Learning rate decay ratio	0.5, 0.6
Learning rate step size	50, 100, 150
Max epochs	500, 750, 1000

Table 9: Values/search space for hyper-parameters on MD17

Hyperparameter	Search space
Cutoff	5
Envelope exponent	5
Number of interaction layers	3, 4, 5
Number of RBFs	6
Number of spherical harmonics	2, 3, 5, 7
Batch size	1, 2, 4, 8
Initial learning rate	1e-4, 5e-4, 1e-3
Learning rate strategy	StepLR
Learning rate decay ratio	0.5, 0.6
Learning rate step size	50, 100, 200
Max epochs	500, 1000, 2000

Table 10: Values/search space for hyper-parameters on OC20

Hyper-parameter	Search space
Cutoff	6, 12
Envelope exponent	5
Number of interaction layers	3, 4, 5
Number of RBFs	6
Number of spherical harmonics	2, 3, 5, 7
Batch size	1, 2, 4, 8
Initial learning rate	1e-4, 5e-4, 1e-3
Learning rate strategy	ReduceLROnPlateau
Optimizer	AdamW
Patience	3
Max epochs	100
EMA decay	0.999
Clip grad norm	10