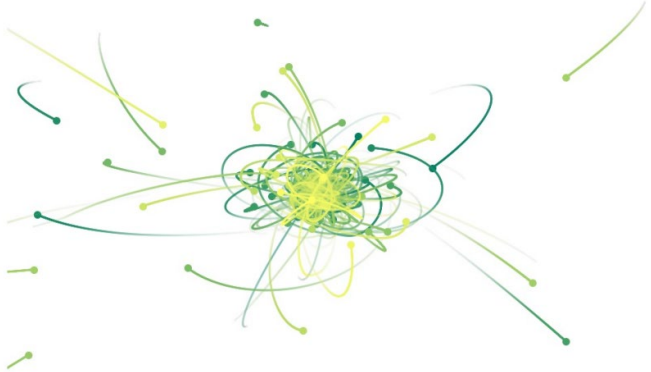
Geometric Learning And Discovery

RENMIN UNIVERSITY OF CHINA

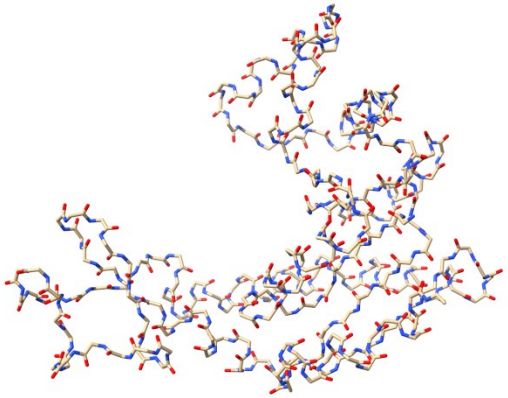
Improving Equivariant Graph Neural Networks on Large Geometric Graphs via Virtual Nodes Learning

Yuelin Zhang, Jiacheng Cen, Jiaqi Han, Zhiqiang Zhang, Jun Zhou, Wenbing Huang

Background

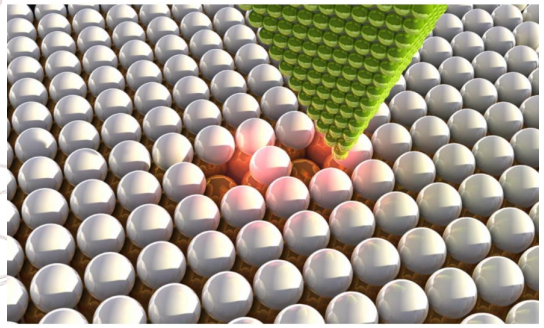


Particle System



Protein Dynamics

Fluid Simulation



Computational Materials

The number of nodes contained in actual scientific scenarios can often reach **hundreds or even thousands**.

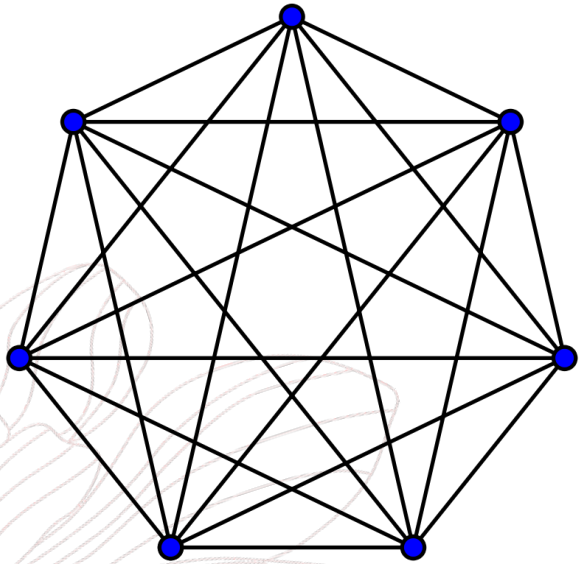
- Protein Dynamics (Nodes ~ **800**)
- Fluid Simulation (Nodes ~ **8000**)

However, current equivariant GNNs are usually only referenced on **small-scale** geometric graphs.

- N-body (Nodes = 5)
- QM9 (Nodes <= 29)

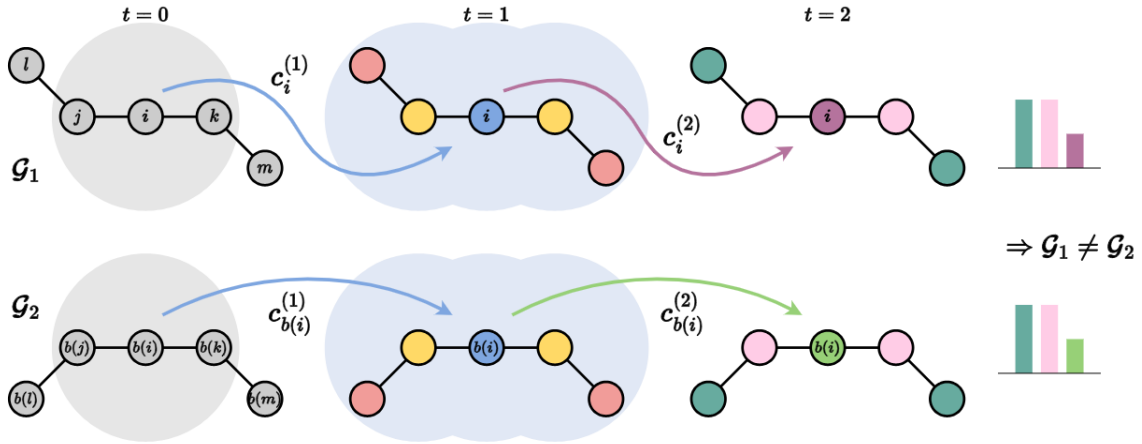
How to apply equivariant GNNs to **large-scale geometric graphs**?

Time and Memory



Quadratic complexity
of complete graph (|E| ~ |V|^2)

Expressive Power

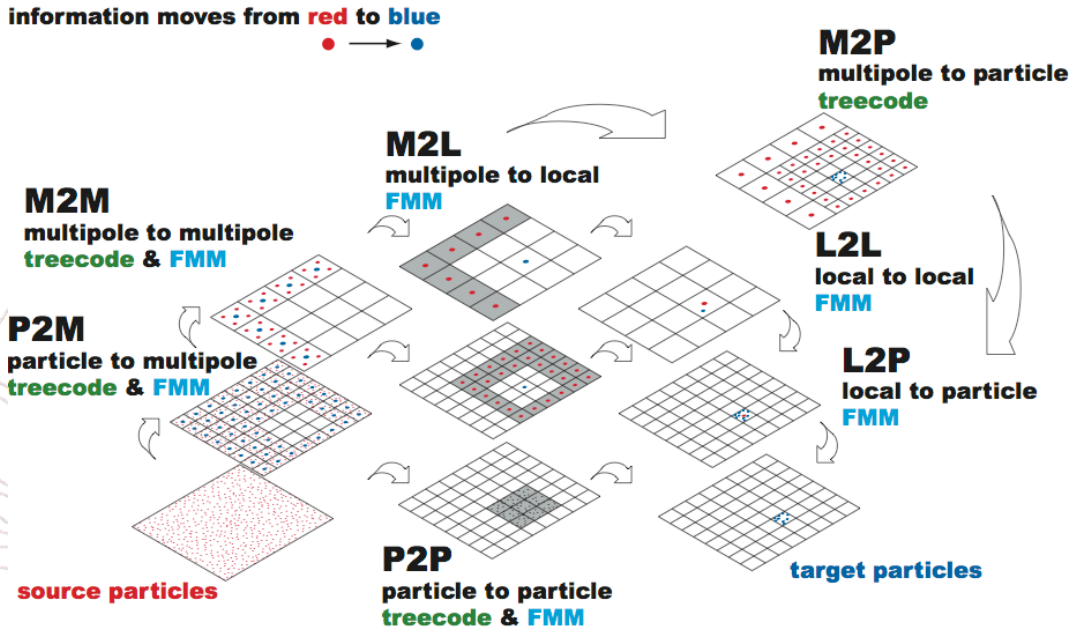


Sparse geometric graphs require
more layers for message passing

[1] On the Expressive Power of Geometric Graph Neural Networks, ICML23.

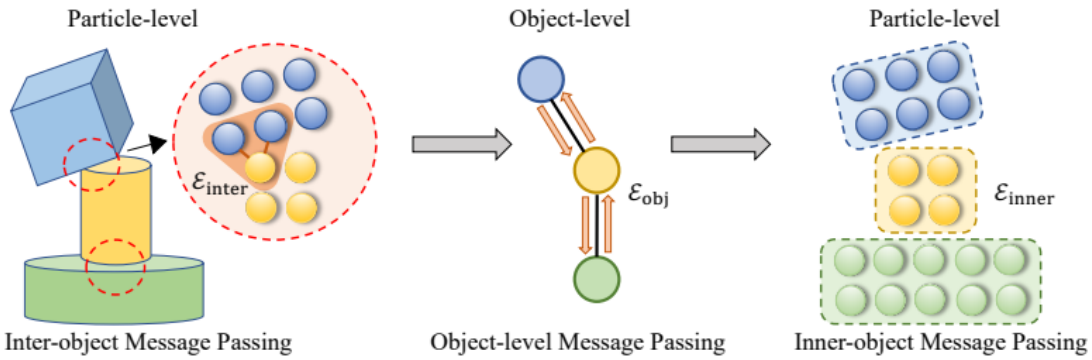
Related Works

Fast Multipole Method



Divide the space to form a tree search structure, but it is **not friendly to parallelization**.

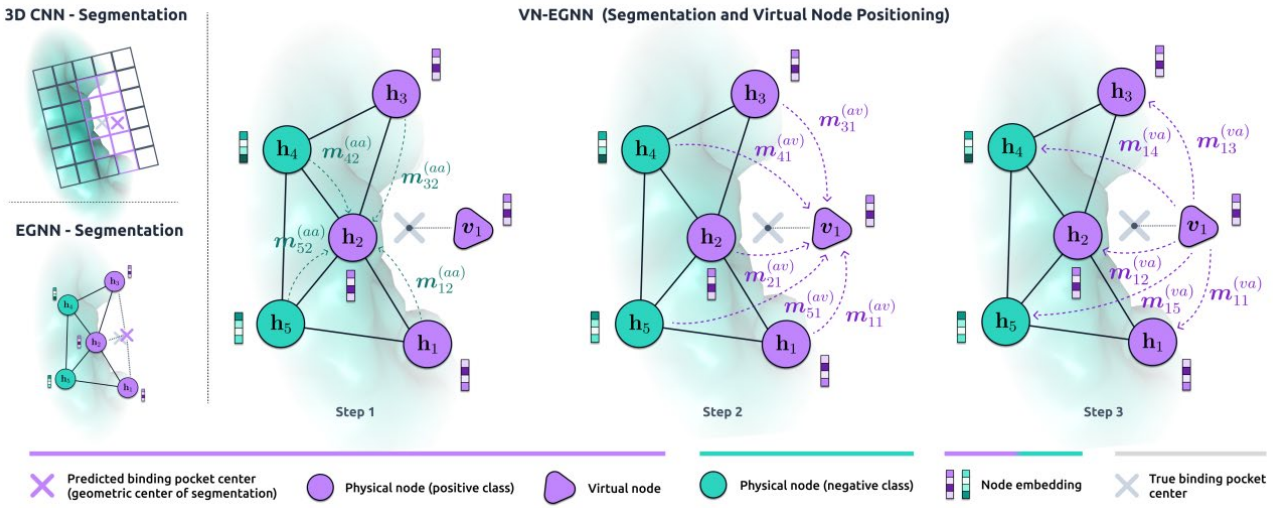
Introduce Prior Knowledge



Not universial

[1] Learning Physical Dynamics with Subequivariant Graph Neural Networks, NeurIPS22.

Related Works

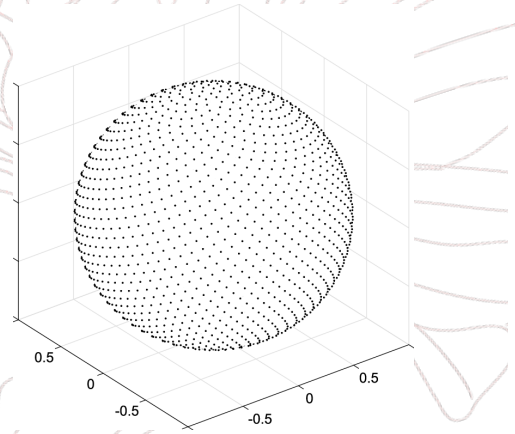


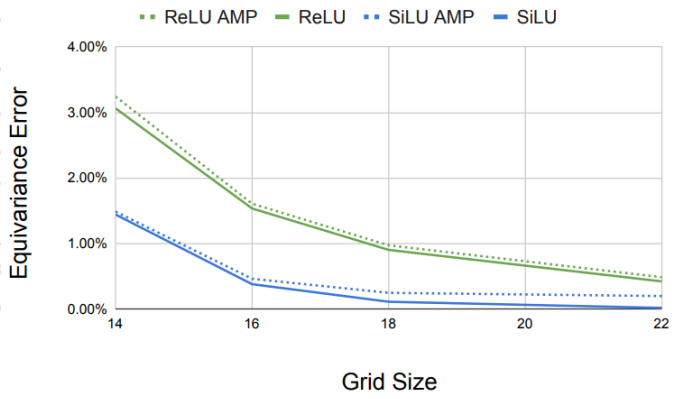
Directly introduce virtual nodes:

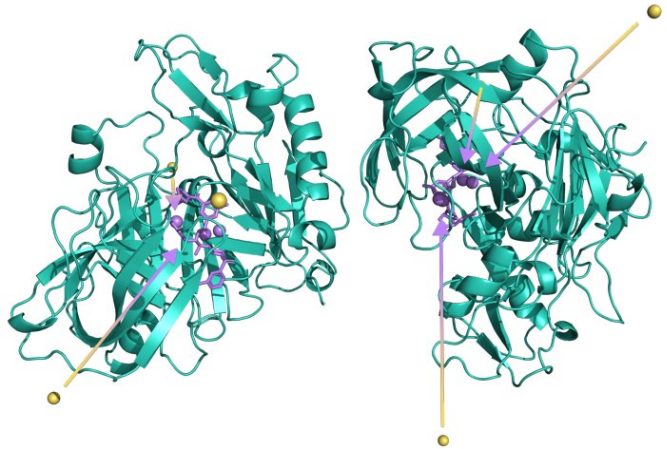
- Sampling from the Fibonacci grid
- Participating in message passing like real nodes

Results:

- **Not E(3)-equivariant**
- Virtual nodes are **difficult to separate**







[1] VN-EGNN: E(3)-Equivariant Graph Neural Networks with Virtual Nodes Enhance Protein Binding Site Identification, NeurIPS23 Workshop AI4D3.
[2] Reducing SO(3) Convolutions to SO(2) for Efficient Equivariant GNNs, ICML23.

Requirements for Virtual Nodes

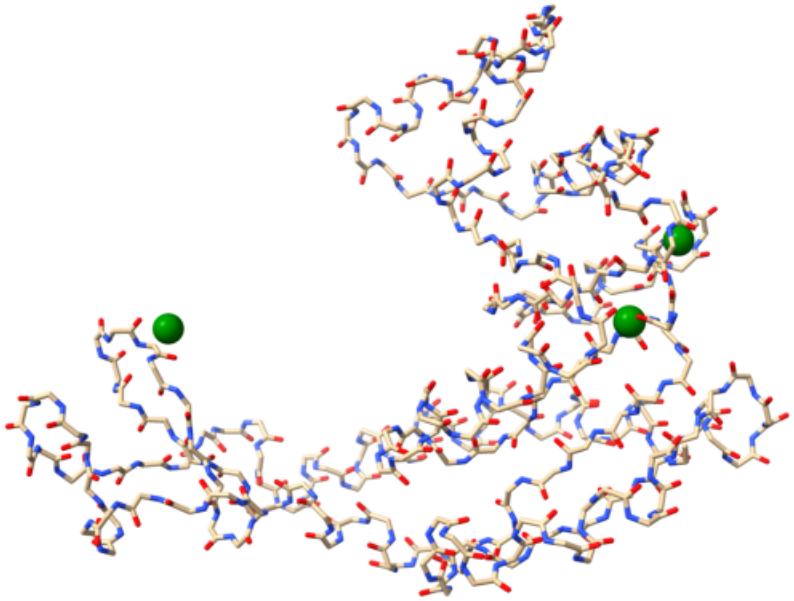


Figure 1. Our FastEGNN is able to learn distinctive virtual nodes (denoted in green) that reflect different dynamical patterns of the protein.

- Do not destroy the **E(3)-equivariance** of equivariant GNNs
- The introduction of virtual nodes **does not require any prior knowledge**
- Different virtual nodes can effectively **represent the distribution of real nodes**

Architecture of FastEGNN

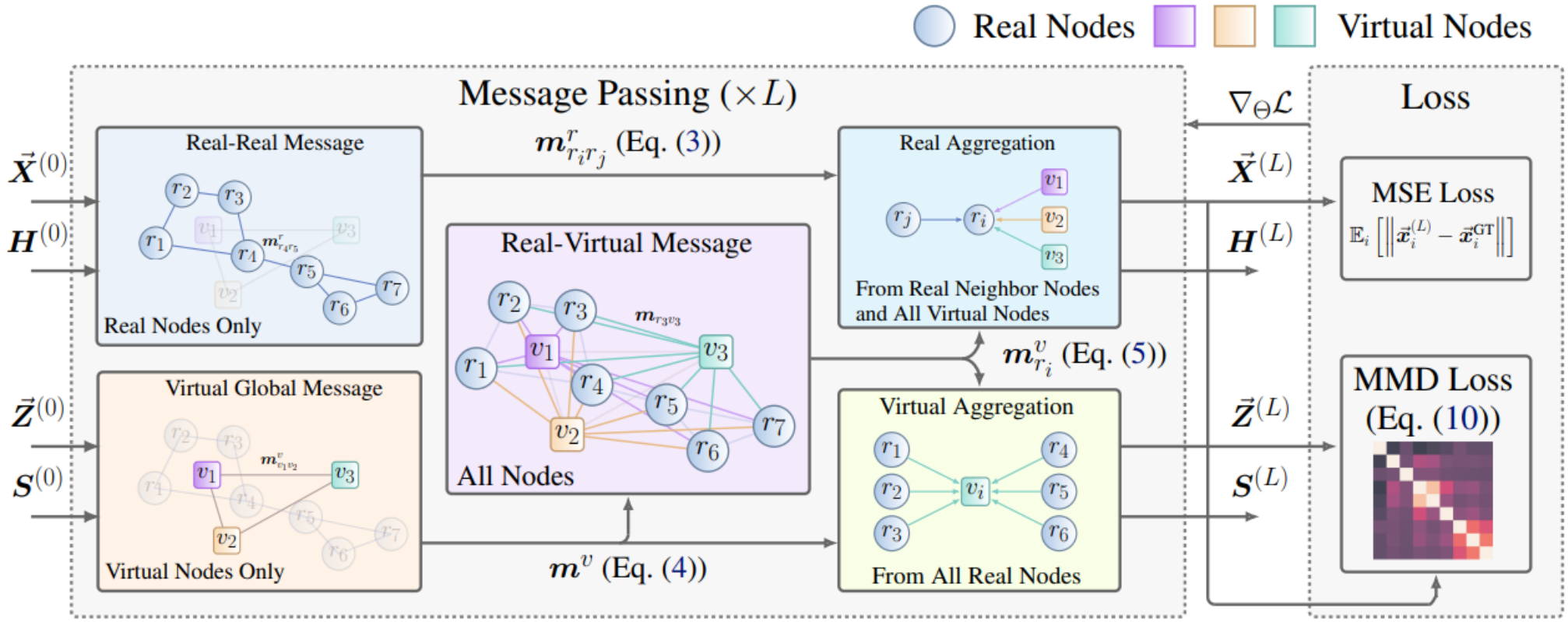
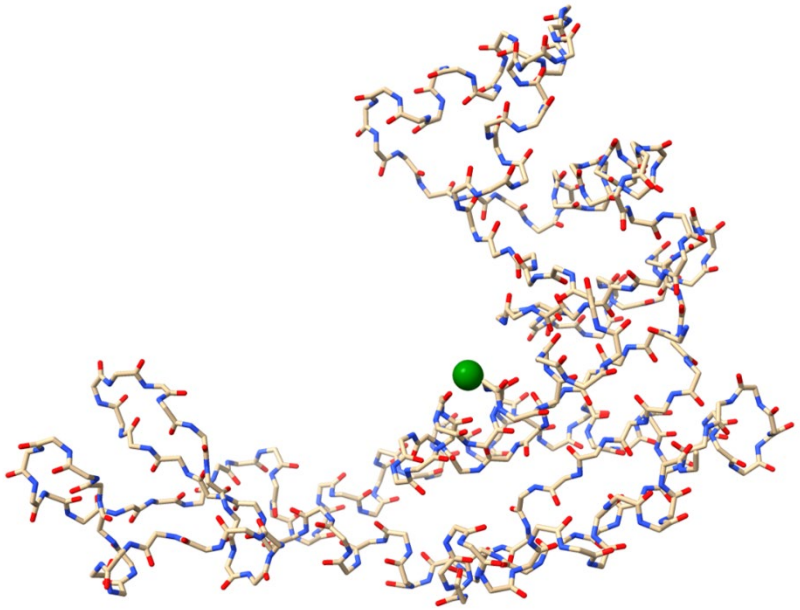


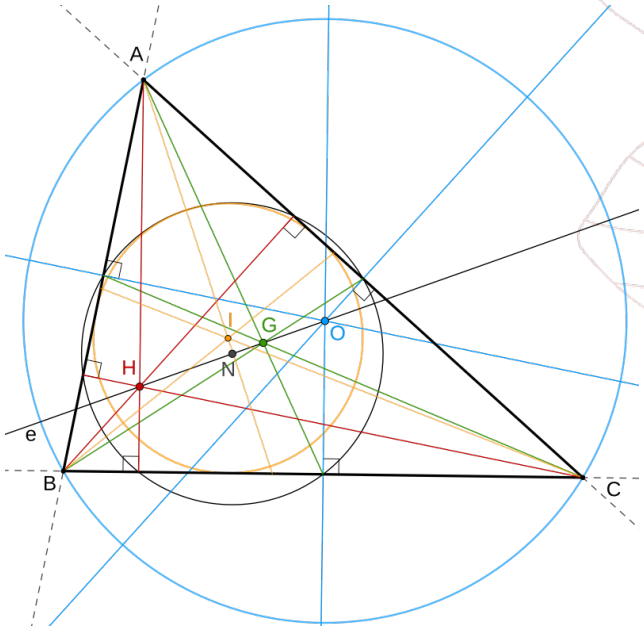
Figure 2. The overall architecture of FastEGNN. $(\vec{X}, H)$ are the real coordinates and features; $(\vec{Z}, S)$ are the virtual coordinates and features. Each layer contains 5 components: Real-Real Message $m^r_{r_i,r_j}$, Virtual Global Message m^v, Real-Virtual Message $m^v_{r_i}$, Real Aggregation, and Virtual Aggregation. The real-real and real-virtual edges are displayed with different colors to indicate that the message passing and aggregation functions over the edges are different.

Equivariant Init. of Virtual Nodes



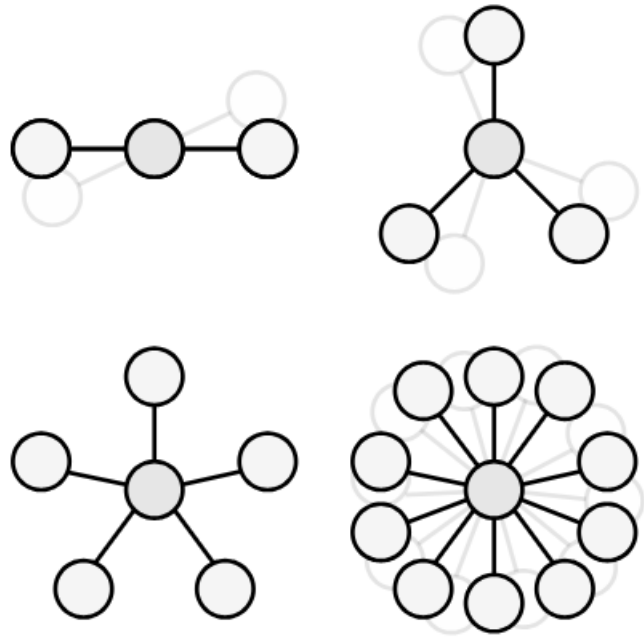
Initiation of virtuals nodes **at CoM (center of mass)**

$$\vec{Z} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i \mathbf{1}^\top$$



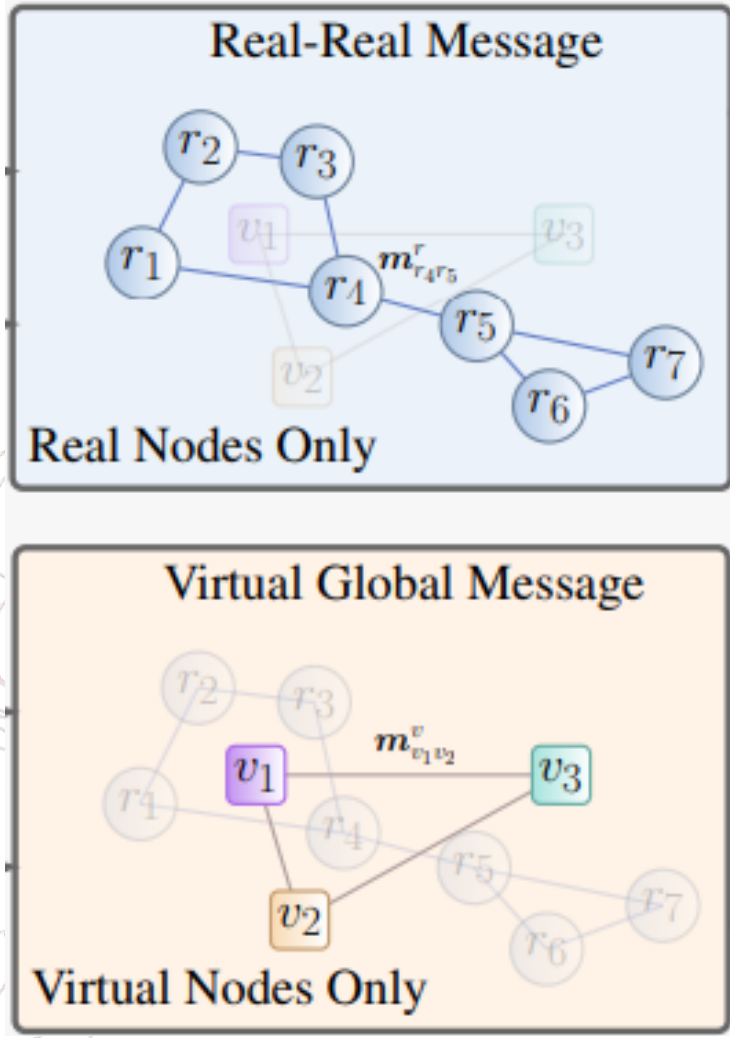
Separating different virtual nodes using **learnable parameters**

$$\boldsymbol{O}\vec{\boldsymbol{Z}} + \vec{\boldsymbol{t}}, \boldsymbol{S} = \varphi_{\text{int}}(\boldsymbol{P}\vec{\boldsymbol{X}}\boldsymbol{O} + \vec{\boldsymbol{t}}, \boldsymbol{PH})$$



Some exceptions (symmetrical structures)

Message Passing in FastEGNN



Message between real nodes:
Use **~(1-η)*|V|^2** edges, can be reduced by a large cutoff rate η

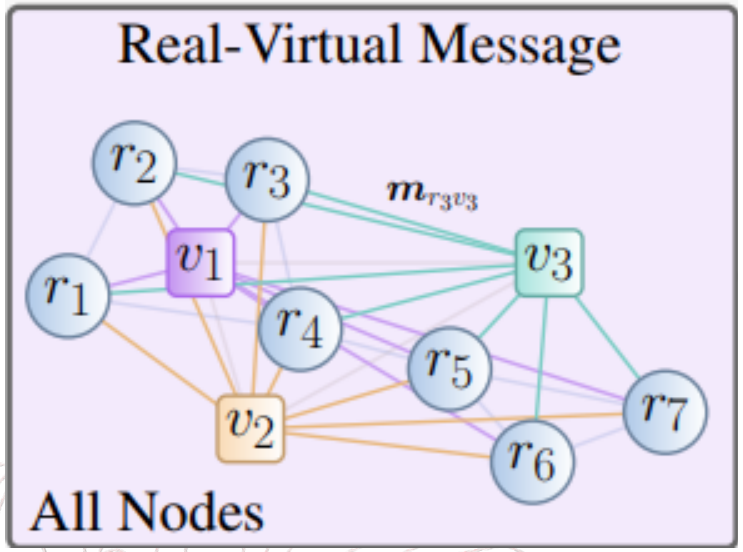
$$\boldsymbol{m}_{ij}^{r} = \varphi_1(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}, \|\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)}\|^2, \boldsymbol{e}_{ij})$$

Message between virtual nodes:
Use **~K^2** edges, K is the number of virtual nodes, K<<|V|, all virtual nodes build **an ordered set**

$$\boldsymbol{m}^{v} = (\vec{\boldsymbol{Z}}^{(l)} - \bar{\boldsymbol{x}}\mathbf{1}^{\top})^{\top}(\vec{\boldsymbol{Z}}^{(l)} - \bar{\boldsymbol{x}}\mathbf{1}^{\top})$$

All message here is E(3)-invariant.

Real-Virtual Message and Aggregation



Message between real nodes:
Use **K*|V|** edges.

$$\boldsymbol{m}_i^v = \varphi_2\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{S}^{(l)}, \bigoplus_{c=1}^{C} \|\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{z}}_c^{(l)}\|^2, \boldsymbol{m}^v\right)$$

Real aggregation

$$\vec{\boldsymbol{x}}_i^{(l+1)} = \vec{\boldsymbol{x}}_i^{(l)} + \alpha_i \sum_{j\in\mathcal{N}(i)} (\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)})\varphi_x^r(\boldsymbol{m}_{ij}^r) + \frac{1}{C}\sum_{c=1}^{C}(\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{z}}_c^{(l)})\varphi_x^v(\boldsymbol{m}_{ic}^v) + \varphi_v(\boldsymbol{h}_i^{(l)})\vec{\boldsymbol{v}}_i^{(0)}$$

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{h}_i^{(l)} + \varphi_h\left(\boldsymbol{h}_i^{(l)}, \bigoplus_{c=1}^{C} \boldsymbol{m}_{ic}^v, \alpha_i \sum_{j\in\mathcal{N}(i)} \boldsymbol{m}_{ij}^r\right)$$

Virtual aggregation

$$\vec{\boldsymbol{z}}_c^{(l+1)} = \vec{\boldsymbol{z}}_c^{(l)} + \frac{1}{N}\sum_{i=1}^{N}(\vec{\boldsymbol{z}}_c^{(l)} - \vec{\boldsymbol{x}}_i^{(l)})\varphi_Z(\boldsymbol{m}_{ic}^v)$$

$$\boldsymbol{s}_c^{(l+1)} = \boldsymbol{s}_c^{(l)} + \varphi_S\left(\boldsymbol{s}_c^{(l)}, \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{m}_{ic}^v\right)$$

Introduction of MMD Loss

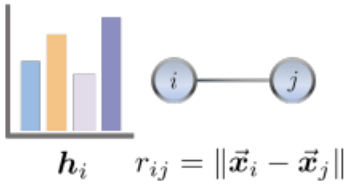
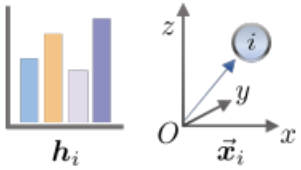
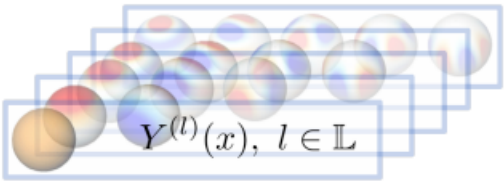
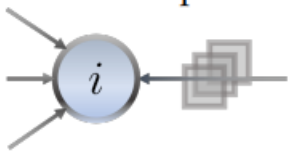
MMD Loss:

$$\frac{1}{C^2}\sum_{i=1}^{C}\sum_{j=1}^{C}k(\vec{\boldsymbol{z}}_i^{(L)},\vec{\boldsymbol{z}}_j^{(L)})-\frac{1}{NC}\sum_{i=1}^{N}\sum_{j=1}^{C}k(\vec{\boldsymbol{x}}_i^{(L)},\vec{\boldsymbol{z}}_j^{(L)})$$

Overall training loss:

$$\mathcal{L}=\mathcal{L}_{\mathrm{MSE}}(\vec{\boldsymbol{X}}^{(L)},\vec{\boldsymbol{X}}_{\mathrm{GT}})+\lambda\mathcal{L}_{\mathrm{MMD}}(\vec{\boldsymbol{X}}_{\mathrm{GT}},\vec{\boldsymbol{Z}}^{(L)})$$

Implementation of FastSchNet

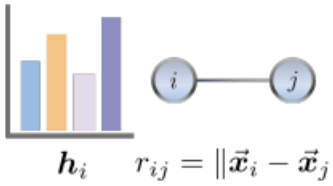
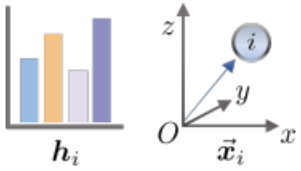
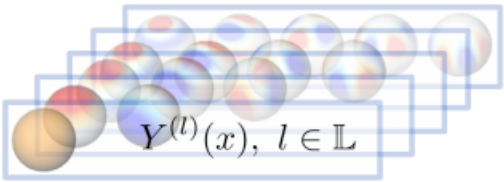
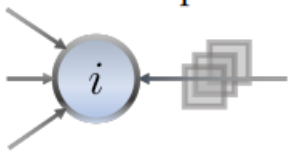
	Invariant GNNs (*e.g.* SchNet [221])	Scalarization-Based Models (*e.g.* EGNN [216])	High-Degree Steerable Models (*e.g.* TFN [242])
	 $\boldsymbol{h}_i \quad r_{ij} = \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|$	 $\boldsymbol{h}_i \quad \vec{\boldsymbol{x}}_i$	
Message Computation	 $\boldsymbol{m}_{ij} = \sigma_2(r_{ij})\sigma_1(\boldsymbol{h}_j)$	$\boldsymbol{m}_{ij} = \sigma_1\left(\boldsymbol{h}_i, \boldsymbol{h}_j, \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|^2, \boldsymbol{e}_{ij}\right)$ $\vec{\boldsymbol{m}}_{ij} = (\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j)\sigma_2\left(\boldsymbol{m}_{ij}\right)$	$\vec{\mathbb{M}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|}\right) \otimes_{\mathrm{cg}}^{\mathbb{W}} \vec{\mathbb{V}}_j^{(\mathbb{L})}$
Feature Update	 $\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j\in\mathcal{N}_i} \boldsymbol{m}_{ij}\right)$	$\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j\in\mathcal{N}_i} \boldsymbol{m}_{ij}\right)$ $\vec{\boldsymbol{x}}_i' = \vec{\boldsymbol{x}}_i + \gamma \sum_{j\in\mathcal{N}_i} \vec{\boldsymbol{m}}_{ij}$	$\vec{\mathbb{V}}_i'^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} + \sigma\left(\vec{\mathbb{V}}_i^{(\mathbb{L})}, \sum_{j\in\mathcal{N}(i)} \vec{\mathbb{M}}_{ij}^{(\mathbb{L})}\right)$

FastSchNet

$$\vec{\boldsymbol{x}}_i^{(l+1)} = \underbrace{\left[\vec{\boldsymbol{x}}_i^{(l)} + \sum_{j\in\mathcal{N}_i} \sigma(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}, \|\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)}\|)(\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)})\right]}_{\texttt{upd\_by\_SchNet}} + \left[\frac{1}{C}\sum_{c=1}^{C}(\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{z}}_c^{(l)})\varphi_x^v(\boldsymbol{m}_{ic}^v)\right],$$

[1] A Survey of Geometric Graph Neural Networks: Data Structures, Models and Applications, arXiv:2403.00485.

Implementation of FastTFN

	Invariant GNNs (*e.g.* SchNet [221])	Scalarization-Based Models (*e.g.* EGNN [216])	High-Degree Steerable Models (*e.g.* TFN [242])
	 $\boldsymbol{h}_i \quad r_{ij} = \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|$	 $\boldsymbol{h}_i \quad \vec{\boldsymbol{x}}_i$	
Message Computation	 $\boldsymbol{m}_{ij} = \sigma_2(r_{ij})\sigma_1(\boldsymbol{h}_j)$	$\boldsymbol{m}_{ij} = \sigma_1\left(\boldsymbol{h}_i, \boldsymbol{h}_j, \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|^2, \boldsymbol{e}_{ij}\right)$ $\vec{\boldsymbol{m}}_{ij} = (\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j)\sigma_2\left(\boldsymbol{m}_{ij}\right)$	$\vec{\mathbb{M}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|}\right) \otimes_{\mathrm{cg}}^{\mathbb{W}} \vec{\mathbb{V}}_j^{(\mathbb{L})}$
Feature Update	 $\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j\in\mathcal{N}_i} \boldsymbol{m}_{ij}\right)$	$\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j\in\mathcal{N}_i} \boldsymbol{m}_{ij}\right)$ $\vec{\boldsymbol{x}}_i' = \vec{\boldsymbol{x}}_i + \gamma \sum_{j\in\mathcal{N}_i} \vec{\boldsymbol{m}}_{ij}$	$\vec{\mathbb{V}}_i'^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} + \sigma\left(\vec{\mathbb{V}}_i^{(\mathbb{L})}, \sum_{j\in\mathcal{N}(i)} \vec{\mathbb{M}}_{ij}^{(\mathbb{L})}\right)$

FastTFN

$$\vec{\boldsymbol{x}}_i^{(l+1)} = \underbrace{\left[\vec{\boldsymbol{x}}_i^{(l)} + \sum_{j\in\mathcal{N}_i} \mathbb{Y}^{\mathbb{L}}\left(\frac{\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)}}{\|\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{x}}_j^{(l)}\|}\right) \otimes_{\mathrm{cg}}^{\mathbb{W}} \left(\boldsymbol{h}_j^{(l)} \| \vec{\boldsymbol{v}}_j^{(0)}\right)\right]}_{\texttt{upd\_by\_TFN}} + \left[\frac{1}{C}\sum_{c=1}^{C}(\vec{\boldsymbol{x}}_i^{(l)} - \vec{\boldsymbol{z}}_c^{(l)})\varphi_x^v(\boldsymbol{m}_{ic}^v)\right],$$

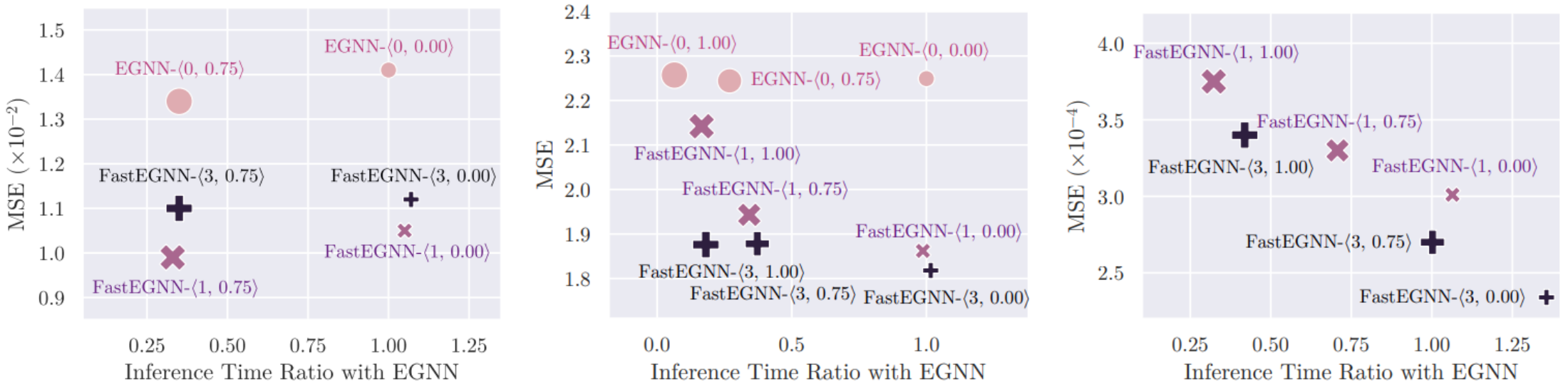
[1] A Survey of Geometric Graph Neural Networks: Data Structures, Models and Applications, arXiv:2403.00485.

Experiments

Table 1. MSE and Inference time ratio with EGNN (Satorras et al., 2021) on N-body System ($\sim$ 100 nodes), Protein Dynamics ($\sim$ 800 nodes), and Water-3D ($\sim$ 8000 nodes). FastEGNN-$\langle C, p\rangle$ denotes the model with C virtual nodes and edge dropping rate as p. We also report the results of EGNN* which indicates the EGNN model with all edges dropped. Note that **for Protein Dynamics and Water-3D, all input graphs have already been reduced to sparse local graphs to enable the efficient implementation for all methods**. Since TFN is much more time-consuming than other models, we chose not to test it on the Water-3D data set.

	N-body System		Protein Dynamics		Water-3D	
	MSE ($\times 10^{-2}$)	Relative Time	MSE	Relative Time	MSE ($\times 10^{-4}$)	Relative Time
Linear	12.66	0.01	2.26	0.01	14.06	0.01
MPNN (Gilmer et al., 2017)	3.06	0.62	150.56	0.62	5299.30	0.61
SchNet (Schütt et al., 2018)	24.83	1.39	2.56	1.40	35.02	2.44
RF (Köhler et al., 2019)	5.76	0.26	2.25	0.25	12.94	0.14
TFN (Thomas et al., 2018)	1.62	17.02	2.26	17.81	–	–
EGNN (Satorras et al., 2021)	1.41	1.00	2.25	1.00	6.00	1.00
EGNN* (Satorras et al., 2021)	11.62	0.03	2.26	0.06	12.38	0.11
FastEGNN-$\langle 3,\ 0.00\rangle$	1.12 ± 0.04	1.07	$\mathbf{1.82 \pm 0.02}$	1.39	$\mathbf{2.49 \pm 0.14}$	1.36
FastEGNN-$\langle 3,\ 0.75\rangle$	$\mathbf{1.10 \pm 0.10}$	0.35	1.88 ± 0.01	0.96	2.83 ± 0.13	1.00
FastEGNN-$\langle 3,\ 1.00\rangle$	9.52 ± 0.02	0.15	1.88 ± 0.01	0.80	3.40 ± 0.04	0.42

Experiments

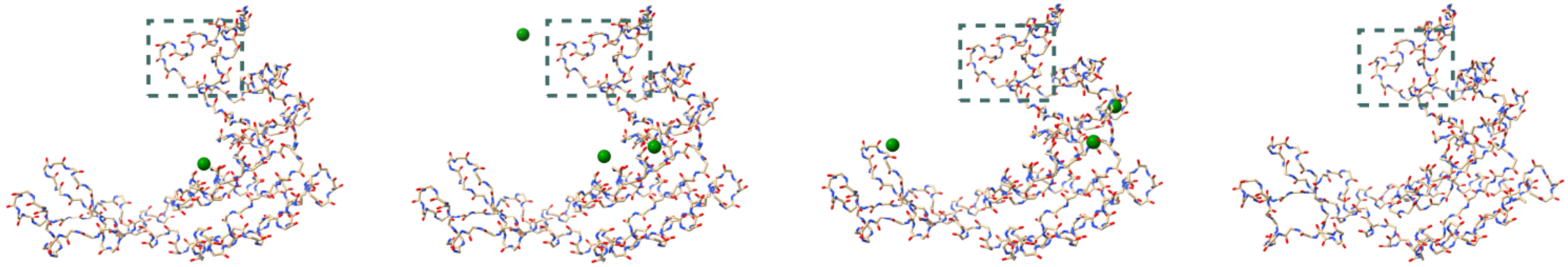


(a) N-body System (b) Protein Dynamics (c) Water-3D

Figure 3. Ablation studies on the number of virtual nodes C and the edge dropping rate p. We eliminate the results of all methods when $p = 1$ on N-body and the performance of EGNN on Water-3D, since these values are poor to display along with the reported ones. Besides, FastEGNN with 10 virtual nodes is also evaluated, with the results provided in Table 6 of Appendix.

Experiments

MSE: 2.582 MSE: 2.214 MSE: 1.560



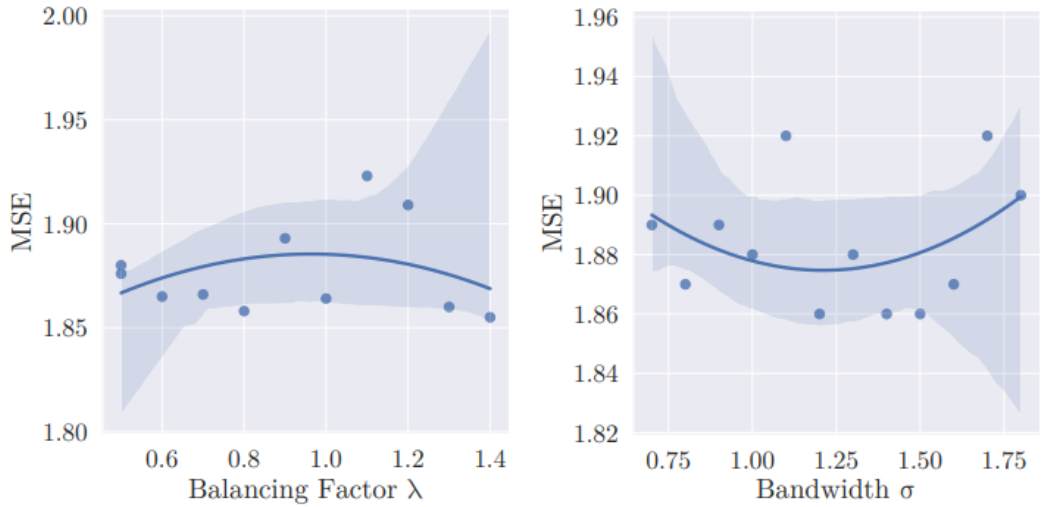
(a) FastEGNN w/ Global Nodes (b) FastEGNN w/o MMD (c) FastEGNN (d) Ground Truth

Figure 4. Visualization of different variants of FastEGNN on Protein Dynamics. We illustrate the predicted position of the protein, as well as the learned location of the virtual nodes (denoted in green). The dashed box marks highlight the region of interest where FastEGNN yields more accurate prediction. For FastEGNN w/ Global Nodes, the learned coordinates of 3 virtual nodes are almost the same.

Experiments

Table 2. Ablation studies on the design of virtual node learning. Experiments are conducted on protein MD dataset and all variants of FastEGNN in this experiment contains 3 virtual nodes. Results are collected with edge dropping rates as 0.5, 0.75, and 1. The performance of EGNN is also reported.

	0.5	0.75	1
EGNN	2.249	2.244	2.257
FastEGNN w/ Global Nodes	1.888	1.967	2.121
FastEGNN w/o MMD	1.882	1.883	1.923
FastEGNN	**1.863**	**1.878**	**1.876**



(a) Fixed $\sigma = 1.0$ (b) Fixed $\lambda = 0.5$

Figure 5. Influence of different hyper-parameters of the MMD loss on protein MD dataset with the model FastEGNN-$\langle 3, 0.75 \rangle$.

Experiments

Table 3. MSE of EGNN (Satorras et al., 2021), RF (Köhler et al., 2019), TFN (Thomas et al., 2018), SchNet (Schütt et al., 2018) and their enhanced models involving virtual nodes learning.

	MSE				
Dropping Rate	0.00	0.50	0.75	0.90	1.00
EGNN (Satorras et al., 2021)	2.25	2.25	2.24	2.25	2.26
FastEGNN-1	1.86	1.91	1.94	1.92	2.14
FastEGNN-3	1.82	1.86	1.88	1.86	1.88
FastEGNN-10	1.87	1.89	1.95	1.89	1.97
RF (Köhler et al., 2019)	2.25	2.26	2.26	2.26	2.26
FastRF-3	2.07	2.02	2.02	2.02	2.05
TFN (Thomas et al., 2018)	2.25	2.26	2.26	2.26	–
FastTFN-3	1.84	2.01	1.90	2.06	–
SchNet (Schütt et al., 2018)	2.56	2.57	2.56	2.56	2.60
FastSchNet-3	1.99	2.01	2.01	2.06	2.08

Experiments

Table 8. We implement VN-EGNN as proposed by Sestak et al. (2023) and compare the MSE loss and time-consuming ratios among EGNN, FastEGNN, and VN-EGNN. The results are presented in the table below. It should be noted that VN-EGNN is not an equivariant model, and it reports extremely higher prediction loss when rotations or translations are applied to the input graphs.

MSE

Dropping Rate	0.00	0.50	0.75	0.90	1.00
EGNN (Satorras et al., 2021)	2.25	2.25	2.24	2.25	2.26
FastEGNN-3	1.82	1.86	1.88	1.86	1.88
VN-EGNN-3 (Sestak et al., 2023) (test equivariance)	–	–	–	–	2703
VN-EGNN-3	1.84	1.88	1.90	1.97	1.92

Relative Time

Dropping Rate	0.00	0.50	0.75	0.90	1.00
EGNN (Satorras et al., 2021)	1.00	0.51	0.27	0.12	0.06
FastEGNN-3	1.39	1.08	0.96	0.92	0.80
VN-EGNN-3 (Sestak et al., 2023)	2.70	2.08	1.87	1.75	1.33

Conclusion

We propose an advanced model with E(n) equivariance for **large geometric graphs** named **FastEGNN**:

- Constructing **an ordered set of virtual nodes** to perform expressive message passing that enjoys both distinctiveness and distributedness.
- Use **MMD loss** to align virtual nodes to real nodes.

Comprehensive evaluations on 100-body simulation, protein molecular dynamics, and particle-based fluid simulation Water-3D consistently demonstrate the superiority of FastEGNN in terms of achieving remarkably lower simulation error and significant speed-up due to sparsification.

GitHub: https://github.com/dhcpack/FastEGNN