SE(3) Equivariant Graph Neural Networks with Complete Local Frames

Weitao Du [*] [†] [1] **He Zhang** [*] [†] [2] **Yuanqi Du** [†] [3] **Qi Meng** [4] **Wei Chen** [†] [1] **Nanning Zheng** [2] **Bin Shao** [4] **Tie-Yan Liu** [4]

Abstract

Group equivariance (e.g. SE(3) equivariance) is a critical physical symmetry in science, from classical and quantum physics to computational biology. It enables robust and accurate prediction under arbitrary reference transformations. In light of this, great efforts have been put on encoding this symmetry into deep neural networks, which has been shown to improve the generalization performance and data efficiency for downstream tasks. Constructing an equivariant neural network generally brings high computational costs to ensure expressiveness. Therefore, how to better trade-off the expressiveness and computational efficiency plays a core role in the design of the equivariant deep learning models. In this paper, we propose a framework to construct SE(3) equivariant graph neural networks that can approximate the geometric quantities efficiently. Inspired by differential geometry and physics, we introduce equivariant local complete frames to graph neural networks, such that tensor information at given orders can be projected onto the frames. The local frame is constructed to form an orthonormal basis that avoids direction degeneration and ensure completeness. Since the frames are built only by cross product operations, our method is computationally efficient. We evaluate our method on two tasks: Newton mechanics modeling and equilibrium molecule conformation generation. Extensive experimental results demonstrate that our model achieves the best or competitive performance in two types of datasets.

1. Introduction

The success of CNNs (Krizhevsky et al., 2012; He et al., 2016) and its SO(2) group extension (Cohen & Welling, 2016) are sufficient to justify the benefits of explicitly translationally equivariant or SE(2) equivariant neural network architectures. On the other hand, Graph Neural Networks (GNNs), which are superior to modeling the high-dimensional structured data with permutation equivariance, bring a new opportunity to model physical systems (especially the various many-body systems) in an end-to-end manner. However, since 3D many-body systems follow other constraints like SE(3) symmetry, pure black-box GNN models show limitations on generalization in this scenario and symmetry-preserving GNN models have become a hot research direction.

The main challenge to be solved for developing general equivariant GNN models is how to express tensors and conduct nonlinear operations in a reference-free way. To represent and manipulate equivariant tensors of arbitrary orders, some approaches resort to equivariant function spaces such as spherical harmonics (Thomas et al., 2018; Fuchs et al., 2020; Bogatskiy et al., 2020; Fuchs et al., 2021) or lifting the spatial space to high-dimensional spaces such as Lie group space (Cohen & Welling, 2016; Cohen et al., 2018; 2019; Finzi et al., 2020; Hutchinson et al., 2021). Since no restriction on the order of tensors is imposed on these methods, sufficient expressive power of these models is guaranteed. Unfortunately, transforming a many-body system into those high-dimensional spaces or calculating equivariant functions (e.g. irreducible representations and tensor product decomposition) usually brings excessive computational cost and optimization difficulty, which is unacceptable in some real-world scenarios. On the other hand, there are also equivariant neural networks (Schütt et al., 2017; Satorras et al., 2021b; Köhler et al., 2019) that directly implement equivariant operations in the original space, providing an efficient way to preserve equivariance without complex equivariant embeddings. However, most of these models only use radial direction as incomplete frames along with embedding of scalar features and abandon higher-order tensor inputs. Thus, they face the direction degeneration problem and are insufficient for expressing more complex geometric quantities like torsion force (See Section 3). In light of this, how to better trade-off the expressiveness and computational effi-

[*]Equal contribution [1]Chinese Academy of Sciences, China [2]Xi'an Jiaotong University, China [3]George Mason University, USA [4]Microsoft Research, USA. † denotes that the work was done when the authors were visiting Microsoft Research. Correspondence to: Qi Meng <meq@microsoft.com>, Weitao Du <duweitao@amss.ac.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

ciency plays a core role in the design of equivariant models.

In this paper, we propose a framework to construct SE(3) equivariant network with **c**omplete **lo**cal **f**rames, called **ClofNet** that can faithfully and efficiently approximate the geometric quantities. Inspired by differential geometry and physics, we incorporate the equivariant local complete frames into graph neural networks, where complete means each local frame forms an orthogonal basis (reference frame) in the 3D vector space with no direction degeneration. To faithfully express the geometric quantities (e.g., scalars, position vectors), we introduce a scalarization block to project them onto our local frames, transforming the tensors into the scalarized coefficients. Afterward, the coefficients are fed into a flexible graph neural network for message passing. The group equivariance of the whole model is guaranteed due to the equivariance of the frames and the expressiveness is guaranteed due to the completeness of the frames. Moreover, the construction of frames only involves cross product operations, making our method computationally efficient (e.g., compared with the Clebsch-Gordan tensor product). Finally, the method to construct the frames is extended to scenarios where the input contains high-order tensors in a similar way. We evaluate the proposed framework on two many-body scenarios that require equivariance with extensive ablation studies: (1) the synthetic many-body system dynamics modeling and (2) the real-world molecular conformation generation. Our model achieves the best performance or competitive results in various types of scenarios. Especially, the proposed ClofNet still keeps competitive performance even with 0.4% training samples on the many-body system, showing its superiority on sample complexity.

2. Preliminaries

In this section, we set up the necessary mathematical preliminaries for understanding the follow-up sections. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{N\times 3}$ be a many-body system living in $\mathbb{R}^3$, where N is the number of particles. For particle i, we use $\boldsymbol{x}_i(t)$ to denote its position at time t and define its neighborhood particles as $\mathcal{N}(\boldsymbol{x}_i(t))$. Let $\times$ denote the cross product of two vectors and $\otimes$ denote the tensor product.

SE(3) group and Equivariance. In the Euclidean space $\mathbb{R}^3$, we consider affine transformations that preserve the distance between any two points, i.e., the isometric group SE(3). We call it the symmetry group w.r.t. the Euclidean metric, and it turns out that SE(3) can be generated by the translation group and the 3D rotation group SO(3) (See rigorous definition in Appendix).

Once given a symmetry group, it's valid to define quantities that are "equivariant" under the symmetry group. Given a function $f : \mathbb{R}^m \to \mathbb{R}^n$, assuming the symmetry group G acts on $\mathbb{R}^m$ and $\mathbb{R}^n$ and we denote the actions by T_g and S_g respectively, then f is G-**equivariant** if

$$f(T_g x) = S_g f(x), \quad \forall x \in \mathbb{R}^m \text{ and } g \in G.$$

For SO(3) group, if $n = 1$, i.e., the output of f is a scalar, then the group action on $\mathbb{R}^1$ is the identity map, in this case f should be $SO(3)$-invariant (Thomas et al., 2018): $f(T_g x) = f(x)$.

The geometric tensors that satisfy the group equivariance are formally defined as tensor field. Let $\{\frac{\partial}{\partial x_i}\}_{i=1}^3$ and $\{dx^i\}_{i=1}^3$ be the tangent vectors and dual vectors in $\mathbb{R}^3$ respectively. Then recall the definition of tensor field on $\mathbb{R}^3$ w.r.t. the SO(3) group:

Definition 2.1. [Definition 2.1.10 of (Jost & Jost, 2008)] An **(r, s)-tensor field** θ is a multi-linear map from a collection of r dual vectors and s vectors in $\mathbb{R}^3$ to $\mathbb{R}$: $\theta(x) = \theta^{i_1\cdots i_r}_{j_1\cdots j_s} \frac{\partial}{\partial x_{i_1}} \otimes \cdots \otimes \frac{\partial}{\partial x_{i_r}} \otimes dx^{j_1} \otimes \cdots \otimes dx^{j_s}$. It implies that under SO(3) transformation $g := \{g_{ij}\}_{1\le i,j\le 3}$, the tensor field θ transforms equivariantly: $\theta^{i'_1\cdots i'_r}_{j'_1\cdots j'_s} = g_{i'_1 i_1} \cdots g_{i'_r i_r} g^T_{j_1 j'_1} \cdots g^T_{j_s j'_s} \theta^{i_1\cdots i_r}_{j_1\cdots j_s}$, where g^T is the inverse of g.

The notion of tensor field is introduced for expressing and embedding geometric and physical quantities, under which the representation satisfies the group equivariance. The tensor or geometric tensor in this paper refers to the tensor field defined in Definition 2.1. A canonical example of $(1, 0)$-type tensor fields is the differential(velocity) of a dynamical system $\boldsymbol{X}(t)$, evolving w.r.t. time t. We name it an *equivariant vector field* (see Appendix A.3.2).

Graph neural networks. Since many body system modeling is independent of the labeling on the observed particles (permutation equivariance), we will use graph-based message-passing mechanism (Gilmer et al., 2020) in this paper. Given a graph $G = (V(G), E(G))$, let h_i, x_i and e_{ij} be the node features, node positions and edge attributes, respectively. EGNN (Satorras et al., 2021b) provides an efficient way to learn equivariant representations, which updates edge message m_{ij}, node embeddings h_i and coordinates x_i as:

$$\begin{aligned} m_{ij} &= \phi_m(h_i, h_j, \|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2, e_{ij}), \\ \boldsymbol{x}_i &= \boldsymbol{x}_i + C\sum_{j\neq i}(\boldsymbol{x}_i - \boldsymbol{x}_j)\phi_x(m_{ij}) \\ h_i &= \phi_h(h_i, \sum_{j\in\mathcal{N}(i)} m_{ij}) \end{aligned} \tag{1}$$

where ϕ_m, ϕ_x and ϕ_h denote three neural networks. C is the normalization factor.

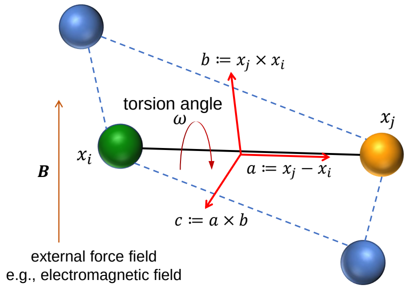

Figure 1. Example of a two-body system to motivate our complete local frame.

3. Motivation from many body interactions.

In this section, we provide an example to motivate our methodology. As shown in Figure 1, we consider a two-body conservative sub-system $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in many-body systems with no time dependency, which can be seen as a graph with two nodes and one edge. Then the force on each particle is the gradient of the potential energy. In general, the groundtruth potential energy function of the two particles is decomposed into two parts: $U(\boldsymbol{x}_i, \boldsymbol{x}_j) := U_1(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + U_{\text{ext}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where the first part is a function of the relative distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ between particles (e.g., electrostatic force), the second part describes the influence of external force fields that may depend on both relative distance and angles (e.g. electromagnetic field, torsion force (A.2)). Therefore, $\nabla U_{\text{ext}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be along arbitrary directions in 3D space besides the radial direction $\boldsymbol{x}_i - \boldsymbol{x}_j$. Most mainstream approaches in equivariant neural network tackle such force prediction problem by taking invariant features (e.g., relative distance) as input and expressing the force along the radial direction, i.e., $\hat{F}_i = k \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j)$ (Schütt et al., 2017; Satorras et al., 2021b), which is not sufficient when the gradient $\nabla U_{\text{ext}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is not along radial direction. To remedy this issue, we put forward an orthogonal SO(3) equivariant frame (basis) for this two-body system. As shown in Figure 1, we take $\boldsymbol{x}_i - \boldsymbol{x}_j$ as the first component/direction $\boldsymbol{a}$, and then include $\boldsymbol{x}_i \times \boldsymbol{x}_j$ as the second direction $\boldsymbol{b}$. Finally, the cross product of the first two directions is taken as the third direction $\boldsymbol{c}$. Using such equivariant frame, any force directions can be equivariantly expressed **with no direction degeneration** by decomposing itself into the three orthogonal directions. The theoretical and technical details about our augmented frames will be discussed later.

4. Methodology

In this section, we provide a detailed description of ClofNet. To preserve the physical symmetries of the system, we represent the system as a spatial graph and construct **ClofNet** based on it with three key components: (1) a **Scalarization** block to project the geometric tensors onto our constructed equivariant frames and concatenate all coefficients as the SO(3)-invariant scalar representations attached to each node; (2) a **Graph message-passing** block to learn SO(3)-invariant edgewise embeddings by propagating and aggregating the scalars on the graph; and (3) a **Vectorization** block to reverse scalar representations back to tensors. A brief overview of ClofNet is illustrated in Figure 2. The translation symmetry can be easily realized by moving the particle system's centroid at $t = 0$ to the origin (the **Centralization** operation in Figure 2). Permutation equivariance automatically holds for the message-passing block. We provide detailed proofs about these symmetries in Appendix A.4.1. Now we concentrate on SO(3) symmetry.

Scalarization Block The scalarization block is designed to transform geometric tensors into edgewise SO(3)-invariant scalar coefficients/features by introducing a novel tuple of local frames. Consider a particle pair $(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))$ at time t, where $\boldsymbol{x}_j(t) \in \mathcal{N}(\boldsymbol{x}_i(t))$. We define the edgewise SO(3)-invariant scalars as $s_{ij} := \text{Scalarize}(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t), \mathcal{F}_{ij})$, where Scalarize is the scalarization operation under local equivariant frames $\mathcal{F}_{ij}$ defined below.

(1) Local equivariant frame construction. Following the last section, based on the particle pair $(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t))$, let $\boldsymbol{a}_{ij}^t = \frac{\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|}$. Define

$$\boldsymbol{b}_{ij}^t = \frac{\boldsymbol{x}_i(t) \times \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) \times \boldsymbol{x}_j(t)\|} \quad \text{and} \quad \boldsymbol{c}_{ij}^t = \boldsymbol{a}_{ij}^t \times \boldsymbol{b}_{ij}^t, \tag{2}$$

Then we build an SO(3)-equivariant frame $\mathcal{F}_{ij} := \text{EquiFrame}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{a}_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{c}_{ij}^t)$. In practice we add a small constant ϵ to the normalization factor in case that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ collapse. If the 3×3 matrix $(\boldsymbol{a}_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{c}_{ij}^t)$ is non-degenerate, then $\mathcal{F}_{ij}$ is complete in the sense that it formulates a *local orthonormal basis*, where 'local' means the frame is a relative reference frame on particles located at the tangent space of $\boldsymbol{x}_i(t)$. Note that these frames are permutation-equivariant among particles. The extreme event that all $\mathcal{F}_{ij}$ ($j \in \mathcal{N}(i)$) degenerate for node i only happens when all particles are restricted to a straight line, which is a measure zero set in $\mathbb{R}^3$. Therefore, we assume $\mathcal{F}_{ij}$ is non-degenerate from now on. Proof for SO(3)-equivariance of $\mathcal{F}_{ij}$ is provided in Proposition A.1.

(2) Invariant scalarization of geometric tensors. With the complete equivariant frame, we realize the scalarization operation (Kobayashi, 1963) as follows. First of all, we project the input vector of particle $\boldsymbol{x}_i$ onto the frame $\mathcal{F}_{ij}$ ($(\boldsymbol{a}_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{c}_{ij}^t)$), and the corresponding *coefficients* are obtained via inner product as follows:

$$(\boldsymbol{x}_i \cdot \boldsymbol{a}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{b}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{c}_{ij}^t). \tag{3}$$

Here we demonstrate that the set of obtained coefficients is actually a SO(3)-invariant scalars tuple and define the

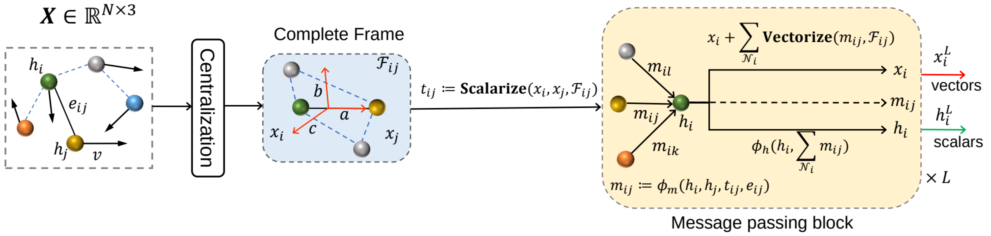
$X \in \mathbb{R}^{N\times3}$
h_i
e_{ij}
h_j
v
Centralization
Complete Frame
$\mathcal{F}_{ij}$
b
a
c
x_i
x_j
$t_{ij} \coloneqq$ Scalarize$(x_i, x_j, \mathcal{F}_{ij})$
$x_i + \sum_{\mathcal{N}_i}$ Vectorize$(m_{ij}, \mathcal{F}_{ij})$
m_{il}
m_{ij}
m_{ik}
h_i
x_i
m_{ij}
h_i
$\phi_h(h_i, \sum_{\mathcal{N}_i} m_{ij})$
$m_{ij} \coloneqq \phi_m(h_i, h_j, t_{ij}, e_{ij})$
x_i^L
vectors
h_i^L
scalars
$\times L$
Message passing block

Figure 2. An overview of ClofNet. For a many-body system $\boldsymbol{X}$, we first centralize the positions to preserve translation equivariance. Then we introduce a tuple of edge-level complete frame $\mathcal{F}_{ij}$ to transform the geometric tensors $\boldsymbol{x}_i$ into SO(3)-invariant scalars. Afterwards, the scalar embeddings t_{ij}, pre-defined node features h_i and edge features e_{ij} are fed to the graph message-passing block to learn edgewise embeddings m_{ij}. Finally, a vectorization block transforms the edgewise embeddings into nodewise vector fields $\boldsymbol{x}_i^L$ or scalar fields h_i^L.

process obtaining such scalars as Scalarize operation. Let $g \in SO(3)$ be an arbitrary orthogonal transformation, then $\boldsymbol{x}_i \to g\boldsymbol{x}_i$, and $(\boldsymbol{a}_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{c}_{ij}^t) \to (g\boldsymbol{a}_{ij}^t, g\boldsymbol{b}_{ij}^t, g\boldsymbol{c}_{ij}^t)$. We can derive that Equation (3) undergoes

$$\begin{aligned}(\boldsymbol{x}_i \cdot & \boldsymbol{a}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{b}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{c}_{ij}^t) \\ &\to ((\boldsymbol{x}_i)^T g^T g\boldsymbol{a}_{ij}^t, (\boldsymbol{x}_i)^T g^T g\boldsymbol{b}_{ij}^t, (\boldsymbol{x}_i)^T g^T g\boldsymbol{c}_{ij}^t) \\ &= (\boldsymbol{x}_i \cdot \boldsymbol{a}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{b}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{c}_{ij}^t), \end{aligned} \tag{4}$$

where we use the fact the transpose of g satisfies $g^T g = I$ (I is the identity matrix), according to the definition of SO(3) group. Next, we discuss that the Scalarize operation could be extended to transform **high-order geometric tensors** into SO(3)-invariant scalars. Suppose the input data contain (2,0)-type tensors, then by extending the each local complete frame $(\boldsymbol{a}_{ij}, \boldsymbol{b}_{ij}, \boldsymbol{c}_{ij})$ through tensor product, it's easy to check that

$$\begin{aligned}&\{\boldsymbol{a}_{ij} \otimes \boldsymbol{a}_{ij}, \boldsymbol{b}_{ij} \otimes \boldsymbol{b}_{ij}, \boldsymbol{c}_{ij} \otimes \boldsymbol{c}_{ij}, \boldsymbol{a}_{ij} \otimes \boldsymbol{b}_{ij}, \\ &\boldsymbol{b}_{ij} \otimes \boldsymbol{a}_{ij}, \boldsymbol{a}_{ij} \otimes \boldsymbol{c}_{ij}, \boldsymbol{c}_{ij} \otimes \boldsymbol{a}_{ij}, \boldsymbol{b}_{ij} \otimes \boldsymbol{c}_{ij}, \boldsymbol{c}_{ij} \otimes \boldsymbol{b}_{ij}\}\end{aligned} \tag{5}$$

forms an equivariant (the equivariance of the tensor products under the equivariant frame are given by Def 2.1) orthonormal frame of the $(2, 0)$-type tensor space. Then the scalarization of a tensor is just to concatenate the *coefficients* of the linear expansion under this frame. In the same way as (4), we can prove that the coefficients are also SO(3)-invariant scalars. Given a $(2, 0)$-symmetric tensor $\boldsymbol{\theta}$ (e.g., energy-momentum tensor), under the complete frame $\mathcal{F}_{ij} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$, $\boldsymbol{\theta}$ can be expressed as:

$$\begin{aligned}\boldsymbol{\theta} = &\theta^{aa}\boldsymbol{a} \otimes \boldsymbol{a} + \theta^{bb}\boldsymbol{b} \otimes \boldsymbol{b} + \theta^{cc}\boldsymbol{c} \otimes \boldsymbol{c} + \theta^{ab}(\boldsymbol{a} \otimes \boldsymbol{b} + \boldsymbol{b} \otimes \boldsymbol{a}) \\ &+ \theta^{ac}(\boldsymbol{a} \otimes \boldsymbol{c} + \boldsymbol{c} \otimes \boldsymbol{a}) + \theta^{bc}(\boldsymbol{b} \otimes \boldsymbol{c} + \boldsymbol{c} \otimes \boldsymbol{b}). \end{aligned} \tag{6}$$

The scalars tuple $s := \{\theta^{aa}, \theta^{ab}, \dots\}$ are the scalarization of $\boldsymbol{\theta}$ under the equivariant frame $\mathcal{F}_{ij}$, which are SO(3)-invariant and will be discussed in detail in Section 5. Let X be the original linear space of all tensors, $\mathcal{S}(X)$ be the space of scalars tuples. Then one of the important benefits of scalarization under an equivariant frame is the scalarization transforms X to $\mathcal{S}(X)$ in an **invertible** way. Given a scalars tuple $s := \{x^a, x^b, x^c\}$ that represents our scalarized input of some position vector $\boldsymbol{x}$ under the equivariant frame $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$, then the inverse from s to vector $\boldsymbol{x}$ is given by

$$\boldsymbol{x} \equiv x^a\boldsymbol{a} + x^b\boldsymbol{b} + x^c\boldsymbol{c}. \tag{7}$$

In our experiments, most of the geometric input is position vectors $\{\boldsymbol{x}_i\}_{i=1}^n$, i.e., $(1, 0)$-type tensors. For the simplicity of description, we denote all other vector features (e.g., velocity) as $\boldsymbol{\chi}_i \in \mathbb{R}^{m\times3}$, where m is the number of potential vector types. The Scalarize operation is defined as: Scalarize$(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{\chi}_i, \boldsymbol{\chi}_j, \mathcal{F}_{ij}) = (\boldsymbol{x}_i \cdot \boldsymbol{a}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{b}_{ij}^t, \boldsymbol{x}_i \cdot \boldsymbol{c}_{ij}^t, \boldsymbol{x}_j \cdot \boldsymbol{a}_{ij}^t, \boldsymbol{x}_j \cdot \boldsymbol{b}_{ij}^t, \boldsymbol{x}_j \cdot \boldsymbol{c}_{ij}^t, \dots)$.

Graph message-passing Block After encoding the geometric tensors into SO(3)-invariant scalars s_{ij}, we first embed them alone with other pre-defined node/edge attributes (h_j, e_{ij}) into high-dimensional representations, and leverage a graph message-passing (GMP) block to learn the SO(3)-invariant edgewise message embeddings m_{ij} by propagating and aggregating information on the graph $\mathcal{G}_X$. Since the block is processing on invariant scalars, ClofNet is **flexible** in the sense that any nonlinear activations can be applied when performing message-passing such that the outputs are still invariant scalars. For more complex tasks, our model is capable of including the attention mechanism. We provide further design details in Appendix A.3.1.

Vectorization block This block is necessary only for SO(3)-equivariant vector output (or general tensor outputs). For scalar output, a simple pooling layer following the graph message-passing block is sufficient. Given the propagated edgewise message m_{ij}, the vectorization block is designed to transform these scalars back to equivariant vectors (inverse of scalarization), which requires pairing m_{ij} with the corresponding complete frame. More precisely, we first project m_{ij} into a scalar triple $\{x_{ij}^a, x_{ij}^b, x_{ij}^c\}$, then define the vectorization process as:

$$(x_{ij}^a, x_{ij}^b, x_{ij}^c) \xrightarrow{\text{Pairing}} x_{ij}^a\boldsymbol{a}_{ij} + x_{ij}^b\boldsymbol{b}_{ij} + x_{ij}^c\boldsymbol{c}_{ij}. \tag{8}$$

Algorithm 1 ClofNet

```
1:  **Input:** $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{N\times 3}$, $h_i \in \mathbb{R}^h$, $e_{ij} \in \mathbb{R}^e$, $\boldsymbol{\chi}_i \in \mathbb{R}^{m\times 3}$, graph $\mathcal{G}_X$
2:  Initialize $\mathbf{X}^1 \leftarrow \mathbf{Centralize}(\mathbf{X})$.
3:  $\mathcal{F}_{ij}^1 = \mathbf{EquiFrame}(\boldsymbol{x}_j^1,\, \boldsymbol{x}_j^1)$;
4:  $s_{ij} = \mathbf{Scalarize}(\boldsymbol{x}_i^1,\, \boldsymbol{x}_j^1, \boldsymbol{\chi}_i, \boldsymbol{\chi}_j, \mathcal{F}_{ij}^1)$;
5:  **for** $(l = 1; l < L; l++)$ **do**
6:      $m_{ij}^l = \phi_m^l(s_{ij}, h_i^l, h_j^l, e_{ij})$;
7:      $\mathbf{h}_i^{l+1} = \phi_h^l(\mathbf{h}_i^l, \sum_{j\in\mathcal{N}(i)} m_{ij}^l)$;
8:      $\mathcal{F}_{ij}^{l+1} = \mathbf{EquiFrame}(\boldsymbol{x}_i^l,\, \boldsymbol{x}_j^l)$;
9:      $\boldsymbol{x}_i^{l+1} = \boldsymbol{x}_i^l + \frac{1}{N}\sum_{j\in\mathcal{N}(\boldsymbol{x}_i)} \mathbf{Vectorize}(m_{ij}^l,\, \mathcal{F}_{ij}^l)$;
10: **end for**
11: **Output:** $x_i^L$ **or** $h_i^L$
```

We encapsulate the pairing process as $V_{ij} = \mathrm{Vectorize}(m_{ij}, \mathcal{F}_{ij})$. Finally, we aggregate all vectors V_{ij} associated with $\boldsymbol{x}_i$ to estimate the ground-truth vector field V_i. Since the local frames are SO(3) equivariant, V_i is also SO(3) equivariant. This method is also applicable to other types of tensors by pairing with the corresponding tensor product (5) of the equivariant frame.

In conclusion, we give a graphic illustration on learning an SO(3)-equivariant vector $\boldsymbol{v}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with vector input $\boldsymbol{x}_1$, $\boldsymbol{x}_2$. Under the equivariant frame of $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, we build ClofNet as a commutative diagram:

$$\begin{tikzcd}
\boldsymbol{x}_1, \boldsymbol{x}_2 \arrow[r, "\boldsymbol{f}"] \arrow[d, "\text{Scalarize}"] & \boldsymbol{v}(\boldsymbol{x}_1, \boldsymbol{x}_2) \\
s_1, s_2 \arrow[r, "\tilde{f}"] & \tilde{f}(s_1, s_2), \arrow[u, "\text{Vectorize}"']
\end{tikzcd}$$

where $\tilde{f}$ is our learning model parameterized by a non-linear neural network, and $s_i \in \mathcal{S}(X)$ is the invariant scalar tuple corresponding to the equivariant vector $\boldsymbol{x}_i$. Then by definition, we realize $\boldsymbol{f} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$:

$$\boldsymbol{f} = \mathrm{Vectorize} \circ \tilde{f} \circ \mathrm{Scalarize}, \tag{9}$$

which is proved to be SO(3)-equivariant in next section. So far, we have achieved permutation and SE(3) equivariance by employing the centralization and the three blocks. The workflow of our method is summarized in Algorithm 1. Note that we can leverage the scalarization block in each message-passing block by inserting the Scalarize operation after line 9 in Algorithm 1. Empirically, we find no significant gains using this modification in our two experiments. Therefore we exclude it to reduce the computation cost. The computational complexity (scalability) of our algorithm is given in appendix A.3.3.

5. Theoretical properties of ClofNet

Equivariance guarantee First, the translation invariance and equivariance are guaranteed by moving the center of a many-body system. The $SO(3)$ group equivariance of ClofNet can be obtained by the following Proposition.

Proposition 5.1. *(1) The local complete frames $(\boldsymbol{a}_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{c}_{ij}^t)$, $\forall i, j$ defined by Equation (2) is equivariant under SO(3) transformation of the spatial space.*

(2) The Scalarize operation, i.e., $\mathrm{Scalarize}(\boldsymbol{x}_i, \boldsymbol{x}_j, \mathcal{F}_{ij})$ *is invariant under SO(3) transformation.*

The proof for the above proposition is put into Appendix A.6, A.4.2. Using this proposition, we conclude that $\boldsymbol{f} = \mathrm{Vectorize} \circ \tilde{f} \circ \mathrm{Scalarize}$ defined in Equation (9) is equivariant under SO(3) transformation.

Expressiveness of ClofNet The expressive power of ClofNet can be decomposed into two parts: the scalarization (and vectorization) part and the ordinary Graph Neural Network part. By the **invertibility** of (7), there exists a bijection between a tensor $\boldsymbol{x}$ and its scalarization $\mathcal{S}(\boldsymbol{x})$. It implies that the scalars can fully characterize the input tensors losslessly (an illustration in terms of mutual information can be found in Appendix A.5). Then the expressive power lies in the expressiveness of the flexible graph message-passing block. The following Theorem verifies the universal approximation property of ClofNet if the equipped graph message-passing block is expressive. See more details and proof in Appendix A.6.

Theorem 5.2. *If the graph message-passing block is selected to be a neural network that can approximate permutation-equivariant polynomials on the graph (e.g. the tensorized graph neural network (Maron et al., 2018) or the minimal universal architecture (Dym & Maron, 2020)), and a linear fully-connected layer connects the output of the graph message-passing block with the vectorization block, then ClofNet has the universality in the space of continuous $SE(3)$ and permutation equivariant functions.*

Remark: (1) The proof relies on the equivalence between direct tensor products (Equation (37)) of equivariant tensors and polynomials of scalars obtain by scalarization, which is realized only through our complete frames. (2) From the theoretical aspect, expressing all permutation-equivariant polynomials requires expressing multiplication operations of invariant scalars on different nodes, which means the scalarization block should contain the vector features of all nodes simultaneously, not only direct neighbors. In practice, we follow previous works to select the common graph convolutional network or the graph transformer network as our graph message-passing block, which brings more efficiency if the contributions of higher-order hops are negligible.

6. Related Work

Various symmetries are considered to be embedded into the network for different tasks. Vanilla CNN models are naturally translation invariant, and more works on 2D-symmetry include (He et al., 2021; Franzen & Wand, 2021; Dieleman et al., 2016; Cohen & Welling, 2016; 2017; Weiler & Cesa, 2019; Shen et al., 2020). Another common symmetry for 3D objects is SO(3) group (Esteves et al., 2019b;a; Worrall & Brostow, 2018; Kondor et al., 2021). In terms of methodology, existing equivariant networks can be roughly classified into two categories by whether conducting all operations in the original space or not. The first category of methods lift the data into high-dimensional spaces (e.g., lie group) or introduce equivariant functions (e.g., spherical harmonics) to preserve equivariance (Worrall et al., 2017; Thomas et al., 2018; Kondor et al., 2018; Weiler et al., 2018b;a; Weiler & Cesa, 2019; Esteves et al., 2020; Romero et al., 2020; Klicpera et al., 2020; Anderson et al., 2019; Batzner et al., 2021). They exhibit sufficient expressive power (Dym & Maron, 2020) but usually bring extra computational costs for calculating spherical harmonic embedding (steerable features (Brandstetter et al., 2021)) or performing integration on Lie groups. If we restrict the equivariant function class to be linear, it turns out that group convolution in a message-passing form is the legal implementation (Bekkers, 2019; Kondor & Trivedi, 2018; Brandstetter et al., 2021; Anderson et al., 2019).

Our work follows methods of the second category that operate on the original space in a computationally efficient way (Schütt et al., 2018; Köhler et al., 2019; Shi et al., 2021; Deng et al., 2021). However, most of these approaches (e.g., EGNN (Satorras et al., 2021b)) abandon a certain amount of geometric (tensor) information, causing their expressive power to be restricted. Recall the EGNN algorithm (1), although the messages are fully non-linear, only the invariant distance feature $\|\boldsymbol{x}_j - \boldsymbol{x}_i\|$ is fed into the model. On the other hand, the success of (Brandstetter et al., 2021; Batzner et al., 2021) has demonstrated the effectiveness for integrating sophisticated geometric and physical features in a non-linear way by equivariant function embedding (or second-hop scalar features (Klicpera et al., 2020)). Although implementing geometric features by spherical harmonics and CG tensor products (tensor product with a decomposition into irreducible representations) are manifestly equivariant, this is not the only way. In fact, they are just sub-solutions that satisfy theorem 1 of He et al. (2021). Different from the existing methods, we propose a flexible and efficient architecture that avoids complex vector-level transformations while preserving complete geometric information through the equivariant frame (see the discussion at the end of Section 3 and Section 5). More related works on encoding geometric (steerable) features and their relation with our model can be found in remark A.5.

7. Experiments

We introduce two many-body scenarios to validate the strength of ClofNet on approximating complex geometric information and avoiding direction degeneration: (1) the simulated Newton mechanics systems and (2) the real-world molecular systems.

7.1. Newtonian many-body system

In this experiment, we leverage ClofNet to predict future positions of synthetic many-body systems that are driven by Newtonian force fields, which is a typical equivariant task because applying any rotations or translations on the initial system state leads to the same transformations on the future system positions. This task is inspired by (Kipf et al., 2018; Fuchs et al., 2020; Satorras et al., 2021b), where a 5-body charged system is controlled by the electrostatic force field. Note that the force direction between any two particles is always along the radial direction in the original setting. To validate the effectiveness of ClofNet on modeling arbitrary force directions, we also impose two external force fields into the original system, a gravity field and a Lorentz-like dynamical force field, which can provide more complex and dynamical force directions.

Dataset We design four systems for three kinds of force fields. (1) The naive charged 5-body system controlled by the electrostatic force field, where each particle carries a positive or negative charge, with initial position and velocity in a 3-dimensional space, i,e., ES(5). (2) Another electrostatic system consists of 20 charged particles, i.e., ES(20). (3) The 20-body system controlled by both the electrostatic field and the gravity field, i.e., G+ES(20). The gravity field is along the z-axis $\boldsymbol{f}_\eta = (0,\, 0,\, g)$, where the gravitational acceleration g is set to 0.98. (4) The 20-body system controlled by both the electrostatic field and the Lorentz-like force field, i.e., L+ES(20). The force field is perpendicular to the direction of each particle's velocity: $\boldsymbol{f}_l(\boldsymbol{v}(t)) = q\boldsymbol{v}(t) \times \mathbf{B}$, where q is the charge of particles and $\mathbf{B}$ denotes the pseudo-vector of the electromagnetic field. We set $\mathbf{B}$ to $[0.5, 0.5, 0.5]$ here.

Implementation details Following EGNN, for each system, we sample 3,000 trajectories for training, 2,000 for validation and 2,000 for test. For each training sample, we provide the initial particle positions $\boldsymbol{X}(0) = \{\boldsymbol{x}_1(0), \cdots \boldsymbol{x}_N(0)\}$, velocities $\boldsymbol{V}(0) = \{\boldsymbol{v}_1(0), \cdots \boldsymbol{v}_N(0)\}$ and associated charges $q = \{q_1, \cdots q_N\} \in \{1, -1\}^5$ as the model input. The prediction label is particle positions after 1,000 timesteps. We compare our model ClofNet to a canonical non-equivariant graph neural network (GNN) and four equivariant models, Radial Field (Köhler et al., 2019), TFN, SE(3)-Transformer and EGNN that can output equivariant vectors. We implemented ClofNet by taking the same

Table 1. MSE for future position prediction over four datasets. The forward time is measured by averaging over multiple batches on an Nvidia Tesla V100 GPU, based on a batch size of 100 samples.

Model	ES(5)	ES(20)	G+ES(20)	L+ES(20)	Forward time (s)
GNN	0.0131	0.0720	0.0721	0.0908	0.0026
TFN	0.0236	0.0794	0.0845	0.1243	0.0247
SE3 Transformer	0.0329	0.1349	0.1000	0.1438	0.0392
Radial Field	0.0207	0.0377	0.0399	0.0779	0.0040
EGNN	0.0079	0.0128	0.0118	0.0368	0.0067
ClofNet	**0.0065**	**0.0073**	**0.0072**	**0.0251**	0.0096

message-passing block as EGNN. All baselines consist of 4 layers with hidden dimension 64 and are trained with AdamW optimizer (Loshchilov & Hutter, 2017) via a Mean Squared Error (MSE) loss. The learning rate and training epochs are tuned independently for each model. Note that we add the reduced centroid back to the predicted positions of ClofNet to achieve translation equivariance. Further details about data generation and model implementation are provided in Appendix A.7.1. An additional experiment of a non-Markov dynamical system can be found in Appendix A.7.2.

Results As shown in Table 1, our model significantly outperforms all baselines on all datasets, demonstrating the expressiveness of ClofNet on modeling geometrical systems. More importantly, compared to EGNN, the previous state-of-the-art (SOTA) of the task, ClofNet exhibits stronger modeling capacity on two more challenging force field scenarios, illustrating that ClofNet can effectively model complicated force directions besides the radial direction. Compared with TFN and SE(3) Transformer, the remarkable shorter forward time of ClofNet indicates that ClofNet is also more time-efficient than equivariant models based on spherical harmonics. Since it only processes tensors in the original space, not including any complex equivariant functions.

Analysis for different number of training samples Following (Satorras et al., 2021b), we also conduct extensive experiments to analyze the performance of ClofNet on datasets in the small and large data regime. The results on the G+ES dataset with sample numbers from 200 to 50,000 are summarized in Figure 3. The proposed ClofNet outperforms another equivariant network EGNN in both small and large data regimes, implying the strength of ClofNet on modeling complex force fields. Compared to GNN, ClofNet exhibits a significant strength in the small data regime and achieves comparable performance in the large data regime, demonstrating that ClofNet is more data-efficient than non-equivariant models in geometric scenarios since the SE(3) equivariance is explicitly encoded into our model. Specifically, ClofNet still keeps competitive performance even with 0.4% (200 of 50,000) training samples, showing its superiority on sample complexity.

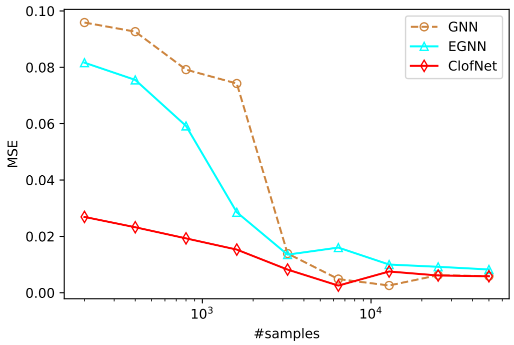

Figure 3. MSE on the G+ES dataset for GNN, EGNN and ClofNet when sweeping over different amounts of training data.

7.2. Molecular Systems

To validate the effectiveness of ClofNet in real-world scenarios, we evaluate ClofNet on a fundamental task in molecular systems called equilibrium conformation generation, trying to predict stable 3D structures from 2D molecular graphs. One of the existing SOTA approaches tackles the problem by directly estimating the gradient fields of the log density of atomic coordinates and then using estimated gradient fields to generate stable structures (Shi et al., 2021; Xu et al., 2021). The key challenge with this approach is that such gradient fields of atomic coordinates are SE(3) equivariant. Following this paradigm, we leverage ClofNet to estimate the gradient fields of atomic coordinates to illustrate its modeling capacity on complex molecular systems.

Evaluation Tasks We conduct experiments on two tasks: (1) *Conformation Generation* evaluates the capacity of ClofNet to learn the conformation distribution by measuring the diversity and accuracy of generated conformations. Given the RMSD of heavy atoms that measures the distance between generated conformation and the reference, Coverage (COV) and Matching (MAT) scores are defined to measure the diversity and accuracy for a given RMSD

Table 2. COV and MAT scores of different approaches on GEOM-QM9 and GEOM-Drugs datasets. For the COV score, the threshold δ is set to 0.5Å for QM9 and 1.25Å for Drugs.

Dataset	Method	COV (%)↑		MAT (Å)↓	
		Mean	Median	Mean	Median
QM9	RDKit	83.26	90.78	0.3447	0.2935
	CGCF	78.05	82.48	0.4219	0.3900
	ConfGF	88.49	94.13	0.2673	0.2685
	GeoMol	71.26	72.04	0.3730	0.3731
	DGSM	**91.49**	**95.92**	**0.2139**	**0.2137**
	EGNN	80.93	86.27	0.3832	0.3898
	ClofNet	90.21	93.14	0.2430	0.2457
Drugs	RDKit	60.91	65.70	1.2026	1.1252
	CGCF	53.96	57.06	1.2487	1.2247
	ConfGF	62.15	70.93	1.1629	1.1596
	GeoMol	67.54	72.71	1.0325	1.0580
	DGSM	78.73	94.39	1.0154	0.9980
	EGNN	40.71	33.01	1.3574	1.3346
	ClofNet	**88.64**	**97.56**	**0.9040**	**0.9023**

Table 3. Accuracy of the distributions over distances generated by different approaches compared to the ground-truth.

Method	Single		Pair		All	
	Mean	Median	Mean	Median	Mean	Median
RDKit	3.4513	3.1602	3.8452	3.6287	4.0866	3.7519
CGCF	0.4490	0.1786	0.5509	0.2734	0.8703	0.4447
ConfGF	0.3684	0.2358	0.4582	0.3206	0.6091	0.4240
ClofNet	**0.1317**	**0.0420**	**0.1787**	**0.0695**	**0.3185**	**0.1142**

threshold δ respectively.

$$\mathrm{COV}(S_g, S_r) = \frac{1}{|S_r|}|\{R \in S_r | \mathrm{RMSD}(R, \hat{R}) < \delta, \hat{R} \in S_g\}|, \tag{10}$$

$$\mathrm{MAT}(S_g, S_r) = \frac{1}{|S_r|} \sum_{R \in S_r} \min_{\hat{R} \in S_g} \mathrm{RMSD}(R, \hat{R}), \tag{11}$$

where S_g and S_r denote generated and reference conformations, respectively.

(2) *Distributions Over Distances* evaluates the discrepancy of distance geometry between the generated conformations and the reference conformations. Following (Shi et al., 2021), we utilize maximum mean discrepancy (MMD) (Gretton et al., 2012) to measure the discrepancy between the generated distributions and the reference distributions.

Table 4. Ablations for ClofNet on two datasets.

Dataset	Method	COV (%)↑		MAT (Å)↓	
		Mean	Median	Mean	Median
QM9	CN w/o Sca	88.99	**94.55**	0.3050	0.3066
	CN w/o GT	85.21	91.18	0.3057	0.3060
	ClofNet	**90.21**	93.14	**0.2430**	**0.2457**
Drugs	CN w/o Sca	65.67	75.63	1.1410	1.1132
	CN w/o GT	70.57	81.82	1.1075	1.1004
	ClofNet	**88.64**	**97.56**	**0.9040**	**0.9023**

Datasets Following (Xu et al., 2021; Shi et al., 2021), we evaluate the proposed model on the GEOM-QM9 and GEOM-Drugs datasets (Axelrod & Gomez-Bombarelli, 2020) as well as the ISO17 dataset (Schütt et al., 2017). The QM9 dataset consists of small molecules up to 29 atoms, while the molecules in the Drugs dataset are generally of medium size, containing up to 181 atoms. ISO17 is for the distance modeling task, which consists of conformations of various molecules with the atomic composition $C_7H_{10}O_2$. To keep a fair comparison with the existing SOTA method ConfGF (Shi et al., 2021), we reproduce its data collection and split settings rigorously. Further details are described in Appendix A.7.3.

Implementation details For the training procedure, we adopt the same score-based framework as in ConfGF. In detail, we perturb the atomic positions with random Gaussian noise of various magnitudes and estimate the gradient field by training the noise conditional score network that is constructed based on ClofNet. See more details about score-based networks in (Shi et al., 2021; Song & Ermon, 2019) or Appendix A.7.3. The ClofNet is equipped with 4 Graph Transformer blocks and the hidden dimensions are set to 288. All models are trained with Adam optimizer via the score matching loss function (See Appendix A.7.3, 49) for 400 epochs. For each molecule in the test set, we follow ConfGF to sample twice as many conformations as the reference ones from each model. All hyperparameters of the score-based framework are provided in Appendix A.7.3.

We compare ClofNet to six classic methods for conformation generation. Specifically, both RDKit (Landrum, 2013) and CGCF(Xu et al., 2021) are distance-based approaches. ConfGF is most close to us, attempting to generate conformations by learning the gradient field of the data distribution in an equivariant manner. DGSM (Luo et al., 2021) is another gradient field-based method built upon ConfGF, which proposes a dedicated dynamic graph constructing strategy to model long-range interactions. However, both ConfGF and DGSM only utilize the distance matrix as the geometric input. GeoMol (Ganea et al., 2021) is a two-stage conformation generation strategy. We also reproduce EGNN on this task as our baseline. The empirical results of all baselines except GeoMol are copied from the original papers since they use the same data split setting as us. We locally reproduce the results of GeoMol on our data split.

Results For the *Conformation Generation* task, we summarize the mean and median COV and MAT scores on two benchmarks for all methods. As shown in Table 2, ClofNet achieves the best or the second best performance on almost all metrics and datasets, demonstrating the effectiveness of our proposed method. In particular, on the Drugs dataset, compared with ConfGF which employs a similar learning strategy with us, ClofNet significantly increases 26.5% COV-mean and 26.6% COV-median scores. Notably, ClofNet also achieves better performance than DGSM which leverages a more complex dynamic graph constructing strategy. A potential interpretation is that molecules in Drugs usually contain more atoms and complex chemical functional groups (e.g., Benzene rings) than those of QM9, thus distance-based geometry is not sufficient to model the gradient field of this complex distribution.

ClofNet also achieves significant improvement in the *Distributions Over Distances* task. As shown in Table 3, ClofNet dramatically outperforms the previous SOTA (ConfGF), demonstrating the strong capacity of the proposed model in modeling molecular dynamics data. In particular, ClofNet reduces the MMD by a magnitude in both Single-median and Pair-Median metrics.

Ablations Although the superior performance on multiple tasks verifies the effectiveness of ClofNet, it remains unclear whether the proposed strategies make a critical contribution. In light of this, we set up several ablative configurations and list the empirical results in Table 4. For the scalarization block, we conduct a variant of ClofNet without the scalarization block, named CN w/o Sca, which only takes the distance matrix of molecules as the input geometric feature. The results show that including scalarization block plays an important role in the model, as the COV-mean and COV-median scores on the Drugs dataset increase by 23.0% and 21.9%, respectively, which implies that including all geometric information will boost the performance of the model. For the graph transformer block, we replace the block with the GIN network (Xu et al., 2018) that is employed in ConfGF, getting the variant CN w/o GT. The results indicate that introducing the attention mechanism will also contribute to the gradient field modeling. We cannot conduct the ablative study for the vectorization block because it guarantees the output of ClofNet is an equivariant vector field.

8. Conclusion and Future Work

To express geometric information more efficiently and flexibly, we develop a novel SE(3) equivariant neural network with complete local frames (ClofNet), aiming at lossless utilization of tensors without incorporating high-dimensional equivariant function embedding. With the proposed local frames, ClofNet could cooperate with any graph neural networks without concerns about breaking the equivariance symmetry. Theoretical analyses and extensive empirical results verify the effectiveness of ClofNet. In the future, we will extend our strategy to other (local) symmetry groups.

Acknowledgements

We thank all the anonymous reviewers for their valuable comments. The work was supported partly by the National Natural Science Foundation of China under Grant 62088102.

References

Anderson, B., Hy, T. S., and Kondor, R. Cormorant: Covariant molecular neural networks. In *NeurIPS*, 2019.

Axelrod, S. and Gomez-Bombarelli, R. GEOM: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *arXiv preprint arXiv:2101.03164*, 2021.

Bekkers, E. J. B-spline cnns on lie groups. *arXiv preprint arXiv:1909.12057*, 2019.

Bogatskiy, A., Anderson, B., Offermann, J., Roussi, M., Miller, D., and Kondor, R. Lorentz group equivariant neural network for particle physics. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 992–1002. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bogatskiy20a.html.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e(3) equivariant message passing, 2021.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Cohen, T. S. and Welling, M. Steerable CNNs. In *ICLR*, 2017.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Cohen, T. S., Geiger, M., and Weiler, M. A general theory of equivariant cnns on homogeneous spaces. *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32(CONF), 2019.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. J. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12200–12209, 2021.

Dieleman, S., De Fauw, J., and Kavukcuoglu, K. Exploiting cyclic symmetry in convolutional neural networks. In *International conference on machine learning*, pp. 1889–1898. PMLR, 2016.

Dormand, J. R. and Prince, P. J. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Dym, N. and Maron, H. On the universality of rotation equivariant point cloud networks. *arXiv preprint arXiv:2010.02449*, 2020.

Esteves, C., Sud, A., Luo, Z., Daniilidis, K., and Makadia, A. Cross-domain 3d equivariant image embeddings. In *International Conference on Machine Learning*, pp. 1812–1822. PMLR, 2019a.

Esteves, C., Xu, Y., Allen-Blanchette, C., and Daniilidis, K. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1568–1577, 2019b.

Esteves, C., Makadia, A., and Daniilidis, K. Spin-weighted spherical CNNs. In *NeurIPS*, 2020.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pp. 3165–3176. PMLR, 2020.

Franzen, D. and Wand, M. General nonlinearities in so (2)-equivariant cnns. *Advances in Neural Information Processing Systems*, 34, 2021.

Freeman, W. T., Adelson, E. H., et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Fuchs, F. B., Wagstaff, E., Dauparas, J., and Posner, I. Iterative SE(3)-Transformers. *arXiv preprint arXiv:2102.13419*, 2021.

Ganea, O., Pattanaik, L., Coley, C., Barzilay, R., Jensen, K., Green, W., and Jaakkola, T. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34, 2021.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Message passing neural networks. In *Machine Learning Meets Quantum Physics*, pp. 199–214. Springer, 2020.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, L., Chen, Y., Dong, Y., Wang, Y., Lin, Z., et al. Efficient equivariant network. *Advances in Neural Information Processing Systems*, 34, 2021.

Hsu, E. P. *Stochastic analysis on manifolds*. 38. American Mathematical Soc., 2002.

Hutchinson, M. J., Le Lan, C., Zaidi, S., Dupont, E., Teh, Y. W., and Kim, H. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pp. 4533–4543. PMLR, 2021.

Jost, J. and Jost, J. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008.

Keriven, N. and Pe yré, G. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32:7092–7101, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Klicpera, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Kobayashi, S. Foundations of differential geometry vol 1 (new york: Interscience) kobayashi s and nomizu k 1969. *Foundations of differential geometry*, 2, 1963.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. *arXiv preprint arXiv:1910.00753*, 2019.

Kondor, I., Trivedi, S., and Lin, Z. A fully fourier space spherical convolutional neural network based on clebsch-gordan transforms, September 2 2021. US Patent App. 17/253,840.

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, 2018.

Kondor, R., Lin, Z., and Trivedi, S. Clebsch–Gordan nets: a fully Fourier space spherical convolutional neural network. In *NeurIPS*, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

Landrum, G. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

Li, Z., Wang, B., Meng, Q., Chen, W., Tegmark, M., and Liu, T.-Y. Machine-learning non-conservative dynamics for new-physics detection. *arXiv preprint arXiv:2106.00026*, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lovász, L. The rank of connection matrices and the dimension of graph algebras. *European Journal of Combinatorics*, 27(6):962–970, 2006. ISSN 0195-6698. doi: https://doi.org/10.1016/j.ejc.2005.04.012. URL https://www.sciencedirect.com/science/article/pii/S0195669805000788.

Luo, S., Shi, C., Xu, M., and Tang, J. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 34, 2021.

Maehara, T. and NT, H. A simple proof of the universality of invariant/equivariant graph neural networks. *arXiv preprint arXiv:1910.03802*, 2019.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Marsden, J. E. and Ratiu, T. S. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, volume 17. Springer Science & Business Media, 2013.

Norcliffe, A., Bodnar, C., Day, B., Simidjievski, N., and Liò, P. On second order behaviour in augmented neural odes. *arXiv preprint arXiv:2006.07220*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Romero, D. W., Bekkers, E. J., Tomczak, J. M., and Hoogendoorn, M. Attentive group equivariant convolutional networks. In *ICML*, 2020.

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E (n) equivariant normalizing flows for molecule generation in 3d. *arXiv preprint arXiv:2105.09016*, 2021a.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks, 2021b.

Schütt, K., Kindermans, P.-J., Felix, H. E. S., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *NeurIPS*, 2017.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Shen, Z., He, L., Lin, Z., and Ma, J. Pdo-econvs: Partial differential operator based equivariant convolutions. In *International Conference on Machine Learning*, pp. 8697–8706. PMLR, 2020.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. *arXiv preprint arXiv:2105.03902*, 2021.

Shi, Y., Huang, Z., Wang, W., Zhong, H., Feng, S., and Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

Simm, G. N. and Hernández-Lobato, J. M. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12438–12448, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Weiler, M. and Cesa, G. General E(2)-equivariant steerable CNNs. In *NeurIPS*, 2019.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. S. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In *NeurIPS*, 2018a.

Weiler, M., Hamprecht, F. A., and Storath, M. Learning steerable filters for rotation equivariant CNNs. In *CVPR*, 2018b.

Worrall, D. and Brostow, G. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 567–584, 2018.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *CVPR*, 2017.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Xu, M., Luo, S., Bengio, Y., Peng, J., and Tang, J. Learning neural generative dynamics for molecular conformation generation. *arXiv preprint arXiv:2102.10240*, 2021.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Zhuang, J., Dvornek, N., Li, X., Tatikonda, S., Papademetris, X., and Duncan, J. Adaptive checkpoint adjoint method for gradient estimation in neural ode. In *International Conference on Machine Learning*, pp. 11639–11649. PMLR, 2020.

A. Appendix

A.1. The special Eulidean group SE(3)

The special Euclidean $SE(3)-$group (Marsden & Ratiu, 2013) is defined as a semidirect (noncommutative) product of 3D rotations $SO(3)$ and 3D translations, $SE(3) := SO(3) \rhd \mathbb{R}^3$. An element of $SE(3)$ is a pair $(R, \boldsymbol{a})$ where $R \in SO(3)$ and $\boldsymbol{a} \in \mathbb{R}^3$.

The action of $SE(3)$ on $\mathbb{R}^3$ is the rotation R followed by translation by the vector a and has the expression

$$(R, \boldsymbol{a}) \cdot x = Rx + \boldsymbol{a}.$$

Then the multiplication and inversion operation in $SE(3)$ are given by

$$(R_1, \boldsymbol{a})(R_2, \boldsymbol{b}) = (R_1 R_2, R_1 \boldsymbol{b} + \boldsymbol{a}) \qquad \text{and} \qquad (R_1, \boldsymbol{a})^{-1} = (R_1^{-1}, -R_1^{-1}\boldsymbol{a}),$$

for $R_1, R_2 \in SO(3)$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$. The identity element is $(I, 0)$.

The 3D rotation group $SO(3)$ is defined by

$$SO(3) = \{R \in \mathcal{M}_{3\times 3}(\mathbb{R}) : R^T R = I, \det(A) = 1\}.$$

For example,

$$R_\varphi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$

belongs to $SO(3)$, parameterized by $\varphi \in [0, 2\pi]$.

A.2. Torsion force field

In this section, we consider a 4-body system $X = (\boldsymbol{x}_l, \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$.

A.2.1. Dihedral angle

For a given node $\boldsymbol{x}_l$ with three neighbors $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$. The dihedral angle θ of the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_j)$ and the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_k)$ is given by the inner product of normal vectors of the two planes:

$$\frac{(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j)}{\|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j)\|},$$

and

$$\frac{(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)}{\|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)\|}.$$

That is,

$$\cos\theta := \frac{(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j) \cdot (\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)}{\|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j)\| \|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)\|}. \tag{12}$$

Therefore, the dihedral angle is a function of the position vectors $\boldsymbol{x}_l$, $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, which cannot be determined purely by the norm of positions: $\|\boldsymbol{x}_l - \boldsymbol{x}_i\|$,$\|\boldsymbol{x}_l - \boldsymbol{x}_j\|$ and $\|\boldsymbol{x}_l - \boldsymbol{x}_k\|$.

A.2.2. Torsion angle and torsion force

The torsion angle of $x_j - x_i$ and $x_l - x_k$ with respect to the bond $x_i - x_k$ is just the dihedral angle θ between the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_j)$ and the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_k)$. By Fourier series, the torsional energy has the following form:

$$E_{tor}(\theta) := \sum_n V_n \cos n\theta.$$

By cutting off the high frequency part ($n \leq 2$), we can simulate an effective torsional energy. Then the torsion force of a particle $\boldsymbol{x}$ is given by the gradient vector:

$$F(\boldsymbol{x}) = -\frac{\partial E_{tor}(\theta)}{\partial \boldsymbol{x}},$$

where $\cos\theta$ is defined by (12).

A.3. Additional descriptions on blocks of ClofNet

A.3.1. Graph message-passing Block

In this section, we provide the implementation details of the Graph Message-passing Block (GMP) that is utilized in ClofNet. For a many-body system $\boldsymbol{X}$, we first represent it as a spatial graph and utilize a message-passing based network to learn the SO(3)-invariant edgewise embeddings m_{ij} from the graph. Considering that there does not exist any graph topology in most real-world scenarios, we design a Graph Transformer Block (GTB) that leverages the attention mechanism due to its powerful capacity in learning the correlations between inter-instances (Vaswani et al., 2017). Now we discuss the workflow of the naive message-passing block and the GTB block. Note that here we omit the time index of the SO(3)-invariant scalars for the simplicity of description.

Feature embedding Given geometric scalars s_{ij}, node features h_i and edge features m_{ij}, GMP first embeds them into high-dimensional representations:

$$h_i = \mathrm{MLP}(h_i), \tag{13}$$

$$e_{ij} = \mathrm{MLP}(e_{ij}), \tag{14}$$

$$s_{ij} = \mathrm{Fourier}(s_{ij}) \text{ or } \mathrm{MLP}(s_{ij}), \tag{15}$$

where MLP denotes a fully connected network and $\mathrm{Fourier}$ denotes a Fourier transformation with a tuple of learnable frequencies.

Naive message-passing Block Following (Satorras et al., 2021b), each message-passing block takes as input the set of node positions $\boldsymbol{x}^l$, node embeddings h^l, SO(3)-invariant scalars s_{ij} and edge information e_{ij} and outputs a transformation on h^{l+1} and x^{l+1}. The equations for each message-passing layer are defined as follows:

$$m_{ij} = \phi_m(s_{ij}, h_i^l, h_j^l, e_{ij}), \tag{16}$$

$$h_i^{l+1} = \phi_h(h_i^l, \sum_{j\in\mathcal{N}(i)} m_{ij}), \tag{17}$$

$$\mathcal{F}_{ij}^{l+1} = \mathbf{EquiFrame}(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l), \tag{18}$$

$$\boldsymbol{x}_i^{l+1} = \boldsymbol{x}_i^l + \frac{1}{N}\sum_{j\in\mathcal{N}(\boldsymbol{x}_i)} \mathbf{Vectorize}(m_{ij}^l, \mathcal{F}_{ij}^l), \tag{19}$$

where ϕ_m, ϕ_k and ϕ_h are MLPs with different parameters. Comparing with (1), the naive message-passing algorithm can also be seen as EGNN (Satorras et al., 2021b) equipped with our equivariant local frames.

Graph Transformer Block The overall architecture of our GTB block is inspired by (Shi et al., 2020). For each GTB block, we first compute the edgewise message with ϕ_m (i.e., Eq. (17), then leverage the message embeddings and node embeddings with a transformer encoder-like architecture to refine the node embeddings. After that, we update edgewise messages with a residue block. In practice, we replace Eq. (18) with the following operations:

$$q_i = \phi_q(h_i^l), k_{ij} = \phi_k(h_i^l, m_{ij}), v_{ij} = \phi_v(m_{ij}), \tag{20}$$

$$\alpha_{ij} = \frac{\langle q_i, k_{ij}\rangle}{\sum_{j'\in\mathcal{N}(i)}\langle q_i, k_{ij'}\rangle}, \tag{21}$$

$$\mathcal{M}_i = \mathrm{LayerNorm}(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} v_{ij}), \tag{22}$$

$$h_i^{l+1} = \phi_h(h_i^l, \mathcal{M}_i), \tag{23}$$

$$h_i^{l+1} = h_i^l + \mathrm{LayerNorm}(h_i^{l+1}), \tag{24}$$

where ϕ_q, ϕ_k and ϕ_v are linear projection layers. ϕ_h is an MLP. α_{ij} and $\mathcal{M}_i$ denote the attention weights and the refined nodewise embeddings, respectively. $\mathrm{LayerNorm}$ refers to the normalization layer adopted in (Vaswani et al., 2017).

A.3.2. Learning gradient fields as score-based models

Molecular conformation generation adopts an SE(3)-equivariant vector field(i.e., gradient field) to depict its dynamic evolving mechanisms.

Gradient field Gradient field is a widely used terminology meaning the first-order derivative w.r.t. a scalar function (Jost & Jost, 2008; Song & Ermon, 2019). To generate molecular conformations (i.e. equilibrium state) with a single stage, (Shi et al., 2021) define "gradient field" to serve as pseudo-forces acting on each particle. By evolving the particles following the direction of the gradient field, the non-equilibrium system will finally converge to an equilibrium state.

Equivariant score function and the Evolving block As to the statistical ensemble system, we try to predict the reverse evolving process from a random state to equilibrium by integrating a first-order equivariant vector field. For example, all physical-allowable molecule conformations are located in an equilibrium state determined by the energy function. Suppose the forward process from equilibrium to non-equilibrium of the system satisfies:

$$d\boldsymbol{X}(t) = f(\boldsymbol{X}(t), t)dt + g(t)dW_t, \ \ 0 \leq t \leq T, \tag{25}$$

where W_t is the Brownian motion and the initial state $\boldsymbol{X}(0)$ follows an unknown equilibrium distribution p_0. Denote the marginal distribution at time t by p_t, then we can write p_t in Gibbs distribution form:

$$p_t(\boldsymbol{X}(t)) = \exp\{-\beta \boldsymbol{H}_t(\boldsymbol{X}(t))\}.$$

It implies that the Hamiltonian function $\boldsymbol{H}_t$ at time t is entangled with p_0. According to (Song et al., 2020)), the reverse evolving process satisfies the following Langevin dynamics:

$$d\boldsymbol{X}(T-t) = f(\boldsymbol{X}(T-t), T-t)dt - g^2(T-t)\nabla \boldsymbol{H}_{T-t}(\boldsymbol{X}_{T-t})dt + g(T-t)dB_t, \tag{26}$$

where B_t denotes the standard Brownian motion. The gradient field of the Hamiltonian function $\nabla \boldsymbol{H}_t(x)$ is also called the **force field**, or the score function (Song & Ermon, 2019). Therefore, ClofNet ϕ is implemented to model the score function: :

$$\nabla \boldsymbol{H}_{T-t}(\boldsymbol{X}_t) = \phi(\boldsymbol{X}(t)), \ \ 0 \leq t \leq T.$$

Note that the vector-valued function $f(x,t)$ and the scalar function $g(t)$ is set to be fixed as prior knowledge. In (Shi et al., 2021) and our molecular experiment, we use the discretization of VE SDE (Song & Ermon, 2019; Song et al., 2020), where $f \equiv 0$ and

$$g(t) = \sqrt{\frac{[d\sigma^2(t)]}{dt}}.$$

Since the distribution Brownian motion is SO(3)-invariant and $g(t)$ is chosen to be an SO(3)-invariant scalar function, then by (25), the forward process is SO(3)-invariant. Combining with the fact that the initial distribution p_0 (molecular conformation distribution) is SO(3)-invariant, it's easy to derive that the score function is a gradient of a **SO(3)-invariant** function, which means that the score function is **SO(3)-equivariant**.

Finally, the learned score function is connected to the evolving block for numerical integrating the Langevin diffusion (26). **For a fair comparison**, ClofNet and Shi et al. (2021) use the same annealed Langevin dynamics sampling algorithm (Song & Ermon, 2019; Shi et al., 2021).

A.3.3. Scalability

Compared to other efficient equivariant models like EGNN (Satorras et al., 2021b;a), the extra computational cost of ClofNet is to calculate the values of the scalars in $t_{ij}, \forall i, j$ through the scalarization Block. Let E denotes the number of edges in the graph, then the cost of calculating the scalars (along with the frames) is $\mathcal{O}(3 \times 3 \times E)$, which is of the same order as the cost of performing 1-hop message-passing. Therefore, the extra cost is much less, Compared to the computational cost of back-propagation in the neural networks. For example, for the molecular conformation generation task, transforming the tensors into SO(3)-invariant scalars only brings 9.6% extra real-time computational cost and 17.4% extra memory cost.

A.4. Proof and discussions

Proposition A.1. *The complete frame $(\boldsymbol{a}^t, \boldsymbol{b}^t, \boldsymbol{c}^t)$ defined by (2) is equivariant under SO(3) transformation of the spatial space.*

Proof. Let $g \in SO(3)$, then under the action of g, the positions of the many-body system $X(t)$ transform equivariantly:

$$(\boldsymbol{x}_1(t), \dots, \boldsymbol{x}_n(t)) \xrightarrow{g} (g\boldsymbol{x}_1(t), \dots, g\boldsymbol{x}_n(t)).$$

From the definition of $\boldsymbol{a}^t$, we know that

$$\boldsymbol{a}^t \xrightarrow{g} g\boldsymbol{a}^t.$$

For $\boldsymbol{b}^t$, since

$$(g\boldsymbol{x}_i(t)) \times (g\boldsymbol{x}_j(t)) = \det(g)(g^T)^{-1}(\boldsymbol{x}_i(t) \times \boldsymbol{x}_j(t)) \tag{27}$$

$$= g(\boldsymbol{x}_i(t) \times \boldsymbol{x}_j(t)), \tag{28}$$

where we have used $g^{-1} = g^T$ for orthogonal matrix g to get the last line. Therefore, $\boldsymbol{b}^t \xrightarrow{g} g\boldsymbol{b}^t$. Applying (27) once again, we have $\boldsymbol{c}^t \xrightarrow{g} g\boldsymbol{c}^t$. □

Frame bundle and scalarization technique A frame at $x \in M$ is a linear isomorphism u from $\mathbb{R}^d$ to the tangent space at x:T_xM. We use $F(M)_x$ to denote the space of all frames at x. Then $GL(d, \mathbb{R})$ acts on $F(M)_x$ by

$$\mathbb{R}^d \xrightarrow{g} \mathbb{R}^d \xrightarrow{u} T_xM.$$

Then the frame bundle

$$F(M) = \cup_{x \in M} F(M)_x$$

can be made into a differential manifold. From the principle bundle point of view, each differential manifold M is a quotient of its frame bundle $F(M)$ by the general linear group $GL(d, \mathbb{R})$: $M = F(M)/GL(d, \mathbb{R})$. We denote the quotient map by π: $F(M) \xrightarrow{\pi} M$. Then, each point of $\boldsymbol{u} \in F(M)$ is a reference frame located at $\boldsymbol{x} := \pi(u) \in M$. On the other hand, $\mathbb{R}^d$ can be seen as a d-dimensional differential manifold with a global coordinates chart and SO(3) is the structure-preserving group of the Euclidean metric with a fixed orientation.

Following Hsu (2002), let $\{e_i,\ 1 \leq i \leq d\}$ be the canonical frame of $\mathbb{R}^d$, and $\{e^i\}$ the corresponding dual frame. At each frame u, the vectors $Y_i := ue_i$ form a frame of T_xM. Let $\{Y^i\}$ be the dual frame of T_x^*M, then a (r,s)-tensor θ can be expressed as

$$\theta = \theta^{i_1 \cdots i_r}_{j_1 \cdots j_s} Y_{i_1} \otimes \cdots \otimes Y_{i_r} \otimes Y^{j_1} \otimes \cdots \otimes Y^{j_s}. \tag{29}$$

The **scalarization** of θ at u is

$$\tilde{\theta}(u) := \theta^{i_1 \cdots i_r}_{j_1 \cdots j_s} e_{i_1} \otimes \cdots \otimes e_{i_r} \otimes e^{j_1} \otimes \cdots \otimes e^{j_s}. \tag{30}$$

Through scalarization, a tensor field θ becomes an ordinary vector space valued function on $F(M)$:

$$\tilde{\theta} : F(M) \to \mathbb{R}^r \otimes \mathbb{R}^s.$$

Therefore, geometric operations such as covariant derivative and tensor product on manifolds can be realized as directional derivative and tensor product of ordinary vector spaces (Hsu, 2002).

Proposition A.2. *There is a one-to-one correspondence between the scalarization of tensor fields on the orthonormal frame bundle $O(\mathbb{R}^3)$ (30) and the SO(3)-invariant scalars tuple obtained by the scalarization block (6) under an equivariant frame.*

Proof. Since we are working in $\mathbb{R}^3$ with a fixed orientation, the $GL(3, \mathbb{R})$ group action is reduced to $SO(3)$ global group action acting on the orthonormal frame bundle $O(\mathbb{R}^3)$. A frame $u \in O(\mathbb{R}^3)$ at $\pi(u) \in \mathbb{R}^3$ can be transported to another point in $\mathbb{R}^3$ by translation. Therefore, we neglect the origin of the frame in the proof. Let $u_e = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ be an equivariant frame of $\mathbb{R}^3$, then given a scalars tuple $\{\theta^{i_1, \dots, i_r}\}$, we construct a vector-valued function on $O(\mathbb{R}^3)$ by:

$$\tilde{\theta}^{i_1, \dots, i_r}(u) = g_{i_1 i_1'} \cdots g_{i_r i_r'} \theta^{i_1', \dots, i_r'},$$

where g is the $SO(3)$ transformation from u_e to another frame u. Moreover, $\tilde{\theta}^{i_1,\dots,i_r}(u)$ is SO(3)-equivariant, since

$$\tilde{\theta}(gu) = g\tilde{\theta}(u),$$

where the g on the right side means the usual extension of the action of SO(3) from $\mathbb{R}$ to the tensor space $\mathbb{R}^{\otimes r}$. We have constructed the $(r,\,0)$ tensor field from the scalars tuple. It's easy to check that $\tilde{\theta}^{i_1,\dots,i_r}(u)$ induces a $(r,\,0)$ tensor field on $\mathbb{R}^3$ by following the definition of (2.1).

From scalarization $\tilde{\theta}^{i_1,\dots,i_r}(u)$ on $O(\mathbb{R}^3)$ to scalarization is obvious. Note that any equivariant frame u_e is also a point on $O(\mathbb{R}^3)$, therefore the scalars tuple is just the values of $\tilde{\theta}^{i_1,\dots,i_r}$ at u_e:

$$\theta^{i_1,\dots,i_r} = \tilde{\theta}^{i_1,\dots,i_r}(u_e).$$

For general $(r,\,s)$-type tensors, the proof is the same by adding the dual frame of u_e. □

A.4.1. Permutation and translation equivariance

For a many-body system $\boldsymbol{X}(t) = (\boldsymbol{x}_1(t), \dots, \boldsymbol{x}_n(t))$, the centroid $C(t)$ is defined by

$$C(t) = \frac{\boldsymbol{x}_1(t) + \cdots \boldsymbol{x}_n(t)}{n}.$$

Translating the reference by $\boldsymbol{h}$, then

$$X(t) + h \to C(t) + h.$$

Therefore, recentering the reference's origin to the centroid at the input's time $t = 0$, we have

$$\boldsymbol{X}(t) - C(0) \xrightarrow{\text{translation by } \boldsymbol{h} \text{ at } t=0} \boldsymbol{X}(t) - C(0).$$

That is, the system is translation-invariant under the re-centered reference if the translation is done at the input's time $t = 0$, which is exactly the scenario considered in predicting the future trajectory or state. Note that although most tensors are **translation-invariant** (e.g., mass,velocities, Gravitational acceleration, gradient fields), the future positions of the many-body system should be **translation-equivariant**. Therefore, we add $\boldsymbol{c}(0)$ back to the output of the predicted positions.

Permutation equivariance is automatically guaranteed by all **graph-based** algorithms, because isomorphic graphs are obliged to yield isomorphic outputs. Therefore, we will just briefly mention why the common 1-hop message-passing algorithms are Permutation equivariant. To emphasis the permutation symmetry, we quote a conclusion from (Zaheer et al., 2017) (which is derived from the Kolmogorov–Arnold representation theorem): if $f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ is a permutation invariant multivariate continuous function, then

$$f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = g(\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)). \tag{31}$$

The crucial point is the function ϕ is shared among all the points, which exactly fits the definition of the so-called message-passing scheme. For a many-body system X, let $v = (v_1, \dots v_n)$ be its vector field, then $v_i \in \mathbb{R}^3$ corresponds the equivariant vector for $\boldsymbol{x}_i \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$. Denote ClofNet with parameters θ by $\Phi^{\theta,t} = \{\Phi_i^{\theta}\}_{i=1}^{n}$, then

$$v_i(t) = \Phi_i^{\theta}(\boldsymbol{X}(t), t),$$

for a fixed particle $\boldsymbol{x}_i$. Suppose $(i_1, \dots, i_k)$ (neighbors of $\boldsymbol{x}_i$) indicate indexes of particles which have interaction with $\boldsymbol{x}_i$, then obviously $1 \leq k \leq n - 1$. By (31), $\Phi_i(\boldsymbol{X}(t), t)$ is an aggregation of message from $\boldsymbol{x}_i$'s neighbors, therefore we have:

$$\Phi_i(X(t), t) = \frac{1}{k} \sum_{j=1}^{k} \phi(\boldsymbol{x}_i(t), \boldsymbol{x}_{i_j}(t), t), \tag{32}$$

and ϕ is a SO(3)-equivariant network with vector output. Note that (32) performs aggregation at the level of vectors, therefore we choose g in (31) to be the arithmetic mean to preserve SO(3) symmetry. A shared neural network ϕ guarantees the permutation equivariance of ClofNet.

A.4.2. EQUIVARIANCE OF CLOFNET

We need the following lemma:

Lemma A.3. *Suppose h is an invariant function, $f_1, \ldots, f_k$ are arbitrary nonlinear functions. Then, the composition $f_k \circ \cdots \circ f_1 \circ h$ is an invariant function.*

Proof. Consider the group action g acting on $x \in \mathbb{R}^3$, then h is invariant means that $h \circ g(x) := h(gx) = h(x)$. We have

$$f_k \circ \cdots \circ f_1 \circ h \circ g = f_k \circ \cdots \circ f_1 \circ h,$$

for every group action g. Hence the composition $f_k \circ \cdots \circ f_1 \circ h$ is invariant. □

Proof of proposition 5.1 Let f_1 be the scalarization block. Because the output of f_1 are scalars (from (3.3)) and SO(3)-scalars must be invariant under SO(3) group action, we can conclude that f_1 is invariant. Denote each graph-transformer layer by $f_i, \ i \in \{2, \ldots, k\}$, then by Lemma A.3, we can conclude that the output of the graph-transformer block is also invariant.

Finally, from the standard fact that scalars multiplying vectors yield equivariant vectors, we conclude that the output of the vectorization block is SO(3)-equivariant.

Combining with the translation equivariance in A.4.1, our model is SE(3)-equivariant.

Remark A.4. In this remark, we investigate how the complete local frame (2) transforms under central reflection $R : \boldsymbol{x} \to -\boldsymbol{x}$. First, we can derive $\boldsymbol{a} \to -\boldsymbol{a}$ from the reflection operation. Second, by the right-hand-thread rule, the cross product of two equivariant vectors yields a pseudo-vector: $\boldsymbol{b} = \boldsymbol{x}_i \times \boldsymbol{x}_j \to \boldsymbol{b}$. Then it is easy to imply that $\boldsymbol{c} \to -\boldsymbol{c}$. Due to $\det(-\boldsymbol{a}, \boldsymbol{b}, -\boldsymbol{c}) = 1$, the orientation of our local frame under reflection R remains unchanged. Thant means, ClofNet is central reflection anti-symmetric. Considering that other reflections including mirror reflection alone a plane can be generated by rotation and central reflection, we can derive that ClofNet is reflection anti-symmetric.

A.5. Information loss

Denote the prediction target by Y, then by the information bottleneck principle (equation (3) of (Tishby & Zaslavsky, 2015)), a learning problem can be formulated as finding a minimal sufficient statistics of the input X with respect to Y. In mathematical terminology, we have a Markov chain:

$$Y \to X \to \hat{X}.$$

$\hat{X}$ is a filtration of the original output X according to a fixed target Y, therefore an information lossless model should satisfy the following equality:

$$I(Y; X) = I(Y, \hat{X}), \tag{33}$$

where $I(\cdot|\cdot)$ denotes the mutual information between two random variables. By the data processing inequality, (33) is satisfied if and only if $I(X, Y|\hat{X}) = 0$ and $Y \to \hat{X} \to X$ also forms a Markov chain.

Consider a toy 4-body example, a particle denoted by x_l with three neighbor particles x_i, x_j and x_k. Suppose the force undertook by x_l is a function of θ: $F_l(\theta)$, where θ is the dihedral angle between the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_j)$ and the plane spanned by $(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_k)$. From the definition, $\cos\theta$ is given by the inner product of normal vectors of the two planes:

$$\boldsymbol{n}_1 := \frac{(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j)}{\|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_j)\|}, \tag{34}$$

and

$$\boldsymbol{n}_2 := \frac{(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)}{\|(\boldsymbol{x}_l - \boldsymbol{x}_i) \times (\boldsymbol{x}_l - \boldsymbol{x}_k)\|}. \tag{35}$$

Therefore, the dihedral angle is a function of the position vectors $\boldsymbol{x}_l$, $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, which cannot be determined purely by the positions' norm: $\|\boldsymbol{x}_l - \boldsymbol{x}_i\|$,$\|\boldsymbol{x}_l - \boldsymbol{x}_j\|$ and $\|\boldsymbol{x}_l - \boldsymbol{x}_k\|$.

Suppose we know the coordinates of particles: $X := (x_l, x_i, x_j, x_k)$, and we want to predict $Y = F_l(\theta)$. Then from the definition of dihedral angle, obviously $Y = F_l(\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_j, \boldsymbol{x}_l - \boldsymbol{x}_k)$. Now a natural question to propose is how to decide $\hat{X}$, which satisfies (33). In EGNN, $\hat{X}_{\text{EGNN}} = (\|\boldsymbol{x}_l - \boldsymbol{x}_i\|, \|\boldsymbol{x}_l - \boldsymbol{x}_j\|, \|\boldsymbol{x}_l - \boldsymbol{x}_k\|)$, then Y and X are not independent conditioned on $\hat{X}$, since θ is not a function of $\hat{X}$, but a function of X. That is,

$$I(Y, X|\hat{X}) \neq 0 \text{ and } I(Y, X) > I(Y, \hat{X}).$$

Therefore, certain information is lost when transforming the original input X to the input of EGNN:$\hat{X}_{\text{EGNN}}$.

On the other hand, ClofNet transforms $X = (\boldsymbol{x}_l - \boldsymbol{x}_i, \boldsymbol{x}_l - \boldsymbol{x}_j, \boldsymbol{x}_l - \boldsymbol{x}_k)$ to $\hat{X}_{\text{ClofNet}} := f(X) := (\bar{\boldsymbol{x}}_l - \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_l - \bar{\boldsymbol{x}}_j, \bar{\boldsymbol{x}}_l - \bar{\boldsymbol{x}}_k)$, where $\bar{\boldsymbol{x}}_i$ is the coordinate of x_i under the equivariant frame of ClofNet (f corresponds to the sclarization, and $\bar{\boldsymbol{x}}_i \in \mathcal{S}(X)$ defined in the scalarization section). Since this transformation f is **invertible** (see (7)), it's obvious that

$$Y \rightarrow \hat{X} \rightarrow X \equiv f^{-1}(\hat{X})$$

forms a Markov chain, and

$$I(X, Y|\hat{X}) = I(f^{-1}(\hat{X}), Y|\hat{X}) = 0.$$

Here we use the fact that $f^{-1}(\hat{X})$ conditioned on $\hat{X}$ is deterministic, therefore $f^{-1}(\hat{X})$ and Y are mutually independent conditioned on $\hat{X}$. Then we conclude that (33) is satisfied:

$$I(Y, X) = I(Y, \hat{X}_{\text{ClofNet}}).$$

It implies no information is lost during the transformation from the original input to ClofNet's input.

A.6. Universal approximation

A crucial ingredient for proving the universal approximation property of a model is to apply Stone-Weierstrass theorem in a proper space. This is nontrivial if the function class we are considering has symmetry. For functions defined on a graph, we want them to be equivariant with respect to permutations on the nodes. Therefore, Maehara & NT (2019) introduces the graph space as the space of equivariant classes of isomorphism graphs. Additional difficulty arises when SE(3)-symmetry is involved. By moving the graph's center, we only need to consider the SO(3)-symmetry acting on each node's feature globally. Now we discuss two graph network structures that can achieve universality. Both of them can be seen as a realization of equivariant polynomials neural networks.

Minimal universal architecture with equivariant frame In section 4 of (Dym & Maron, 2020), the authors proposed an equivariant network that can achieve SE(3) equivariance by tensor representations. We briefly sketch the general approach of (Dym & Maron, 2020).

Roughly speaking, a function class $\mathbf{F}_C(\mathcal{F}_{\text{feat}}, \mathcal{F}_{\text{pool}})$ with universal approximation property can be decomposed into two conditions:

- **Condition 1.** $\mathcal{F}_{\text{feat}}$ is able to represent all polynomials which are translation invariant and permutation-equivariant;
- **Condition 2.** $\mathcal{F}_{\text{pool}}$ contains all linear SO(3)-equivariant mappings between the range of $\mathcal{F}_{\text{feat}}$:$W_{\text{feat}}$, and the output tensor space.

Furthermore, (Dym & Maron, 2020) showed that for a non-negative integer vector $\vec{r} = (r_1, \ldots, r_k)$, $Q^{(\vec{r})} = \{(Q_j^{(\vec{r})\}})_{j=1}^n$ defined by

$$Q_j^{(\vec{r})}(\boldsymbol{X}) = \sum_{i_2, \ldots, i_K = 1}^{n} \boldsymbol{x}_j^{\otimes r_1} \otimes \boldsymbol{x}_{i_2}^{\otimes r_2} \otimes \boldsymbol{x}_{i_3}^{\otimes r_3} \otimes \ldots \otimes \boldsymbol{x}_{i_K}^{\otimes r_K}. \tag{36}$$

are SO(3)-equivariant and satisfy condition 1. **In fact, they are (r,0)-type tensors, and transform equivariantly by definition (2.1)**. Here, each x_i is already centralized and there is no need to centralize them as (6) of (Dym & Maron, 2020). To express these tensor products, Dym & Maron (2020) designed a **parameterized minimal universal architecture neural layer** ((7) of (Dym & Maron, 2020)) from input $(\boldsymbol{X}, \boldsymbol{v})$ to the output $(\boldsymbol{X}, \tilde{\boldsymbol{v}})$:

$$\tilde{\boldsymbol{v}}_j(\boldsymbol{X}, \boldsymbol{v}|\theta_1, \theta_2) = \theta_1 \boldsymbol{x}_j \otimes \boldsymbol{v}_j + \theta_2 \sum_i \boldsymbol{x}_i \otimes \boldsymbol{v}_i,$$

where $\boldsymbol{v}_i$ is the input tensor feature of the body position $\boldsymbol{x}_i$. Now it's necessary to show how ClofNet performs direct tensor product under scalarization. We take a simple example of the inner product between two vectors: $\boldsymbol{\chi}_1 = \chi_1^a \boldsymbol{a} + \chi_1^b \boldsymbol{b} + \chi_1^c \boldsymbol{c}$ and $\boldsymbol{\chi}_2 = \chi_2^a \boldsymbol{a} + \chi_2^b \boldsymbol{b} + \chi_2^c \boldsymbol{c}$, under equivariant frame $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. Then the tensor product gives:

$$\begin{aligned}\boldsymbol{\chi}_1 \otimes \boldsymbol{\chi}_2 =& \chi_1^a \cdot \chi_2^a \boldsymbol{a} \otimes \boldsymbol{a} + \chi_1^b \cdot \chi_2^b \boldsymbol{b} \otimes \boldsymbol{b} + \chi_1^c \cdot \chi_2^c \boldsymbol{c} \otimes \boldsymbol{c} \\ &+ \chi_1^a \cdot \chi_2^b \boldsymbol{a} \otimes \boldsymbol{b} + \chi_1^b \cdot \chi_2^a \boldsymbol{b} \otimes \boldsymbol{a} \\ &+ \chi_1^a \cdot \chi_2^c \boldsymbol{a} \otimes \boldsymbol{c} + \chi_1^c \cdot \chi_2^a \boldsymbol{c} \otimes \boldsymbol{a} \\ &+ \chi_1^b \cdot \chi_2^c \boldsymbol{b} \otimes \boldsymbol{c} + \chi_1^c \cdot \chi_2^b \boldsymbol{c} \otimes \boldsymbol{b}. \end{aligned} \tag{37}$$

Under scalarization, the output of this tensor product is represented by the set containing second order polynomials of the invariant scalars $(\{\chi_1^a \ldots \chi_2^c\})$ defined by (6).

Now we are ready to present the **parameterized minimal universal architecture neural layer with equivariant frames**. After scalarization, denote the invariant scalars of $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{v}_i\}_{i=1}^n$ by $S^j(\boldsymbol{x}_i) := \{S^j(\boldsymbol{x}_i)\}_{j=1}^3$ and $S^j(\boldsymbol{x}_i) := \{S^j(\boldsymbol{v}_i)\}_{j=1}^m$, where j is the coefficients index of $\boldsymbol{v}_i$. Then each layer maps $(\boldsymbol{X}, \boldsymbol{v})$ to the output $(\boldsymbol{X}, \tilde{v})$:

$$(\boldsymbol{X}, \boldsymbol{v}) \xrightarrow{\text{Scalarization}} (S(\boldsymbol{X}), S(\boldsymbol{v})) \longrightarrow \tilde{v} := \tilde{v}(S(\boldsymbol{X}), S(\boldsymbol{v}) | \theta_1, \theta_2), \tag{38}$$

and $\tilde{v}(S(\boldsymbol{X}), S(\boldsymbol{v}) | \theta_1, \theta_2)$ is given by

$$\tilde{v}_i^{jk}(S(\boldsymbol{X}), S(\boldsymbol{v}) | \theta_1, \theta_2) = \theta_1 S^j(\boldsymbol{x}_i) \cdot S^k(\boldsymbol{v}_i) + \theta_2 \sum_l S^j(\boldsymbol{x}_l) \cdot S^k(\boldsymbol{v}_l),$$

where jk denotes the coefficients index of $\tilde{v}$ (each operation will bring an extra index for the features). Note that $\tilde{v}$ is a tuple of invariant scalars, and it will connect to the vectorization block when the final output is a higher-order tensor.

***Proof of theorem* 5.2.** Suppose our target is to predict an equivariant tensor associated with a pre-fixed node with input the many-body system $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{N \times 3}$. Then we fix a single equivariant frame on the pre-fixed node by averaging our edgewise frames(bases) around this node. Under this frame, we **scalarize $\{\boldsymbol{x}_i\}_{i=1}^n$ of all nodes (not only the 1-hop nodes)**. To relate the minimal universal architecture with ClofNet, we emphasise that formula (37) already shows that with the equivariant frame, **tensor products are ordinary multiplication of invariant scalars.** From a commutative diagram point of view, let $\boldsymbol{f} f(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2)$ denote the tensor product on two equivariant vectors $\boldsymbol{\chi}_1 = \chi_1^a \boldsymbol{a} + \chi_1^b \boldsymbol{b} + \chi_1^c \boldsymbol{c}$ and $\boldsymbol{\chi}_2 = \chi_2^a \boldsymbol{a} + \chi_2^b \boldsymbol{b} + \chi_2^c \boldsymbol{c}$ under equivariant frame $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ and denote the scalarization of $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ by s_1 and s_2, then we have the following commutative diagram:

$$\begin{tikzcd} \boldsymbol{\chi}_1, \boldsymbol{\chi}_2 \arrow[r, "\boldsymbol{f}"] \arrow[d, "\text{Scalarize}"] & \theta_1 \otimes \theta_2 \\ s_1, s_2 \arrow[r, "\tilde{f}"] & \tilde{f}(s_1, s_2), \arrow[u, "\text{Vectorize}"'] \end{tikzcd}$$

and Vectorize $\circ\, \tilde{f}$ is exactly the right hand side of (37), where $\tilde{f}(s_1, s_2)$ is a combination of second order polynomials of s_1, s_2. Here, the Vectorize block is under the second-order equivariant frame built from $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$: $\{\boldsymbol{a} \otimes \boldsymbol{a}, \ldots, \boldsymbol{b} \otimes \boldsymbol{c}\}$. [1]The same story holds for tensor product of higher orders, and we conclude that an neural network which can approximate permutation-equivariant polynomials universally (not necessary (38)) meets condition 1 in the ClofNet setting.

In terms of condition 2, Dym & Maron (2020) gave an explicit construction in the appendix of all equivariant linear functionals when the output is a scalar. It's not a coincidence that both the equivariant frame (2) and (23) of (Dym & Maron, 2020) used the inner product and determinant (equivalent to the cross product). However, with our equivariant frame, we can give a straightforward argument on building equivariant functionals.

By the commutative diagram (A.6), both the input and output tensor space are scalarized under the equivariant frame $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ (and the tensor product of equivariant frame). Furthermore, a **linear** functional ϕ is fully determined by its value on the frame. Suppose we are considering the map from order two tensors to order one tensors (vectors), then ϕ is determined by the coefficients (SO(3)-invariant) (ϕ^1, ϕ^2, ϕ^3):

$$\phi(\boldsymbol{a} \otimes \boldsymbol{b}) = \phi^1(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{a} + \phi^2(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{b} + \phi^3(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{c},$$

[1] **Recall from our construction of the frame, vector $\boldsymbol{a}$ denotes the radial direction. Then if we perform tensor product only along this direction, the results won't expand the whole space of the second-order tensors, the direction degeneration problem would become critical.**

for all possible combinations $\boldsymbol{a}, \boldsymbol{b} \in \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$. **Differentiate with the 'cumbersome' method of constructing equivariant linear functionals in (Dym & Maron, 2020), ϕ is a linear functional with no equivariance restriction, which is straightforward to construct with a fully-connected linear layer.** Combining the two parts, we have constructed the universal architecture with equivariant frame. □

Remark A.5 (Incorporating geometric (physical) quantities as steerable features). The general definition of steerable features can be found in (8) of (Freeman et al., 1991). Briefly, frame coefficients of a function expanded in an equivariant frame (including sphere harmonics and tensor products of our equivariant frames) satisfy (8) of (Freeman et al., 1991). Then Brandstetter et al. (2021) injects geometric quantities (forces, momentum, position vectors, ...) by performing CG tensor product between the chosen geometric quantities and each layers' output (also equivariant tensors), to make the model more expressive. However, the message from section 4 of (Dym & Maron, 2020) indicates that the computational expensive SO(3) representations (spherical harmonics and CG decomposition) can bring for the expressiveness, can be realized by direct tensor product (formula (5) is a special case of generating tensor product equivariant bases from our equivariant frame). As we have shown in the above proof, the direct tensor product under the equivariant frame can be realized by the multiplication of invariant scalars.

In conclusion, to add geometric quantities in a steerable way, we can first scalarize the steerable features, and then either encode the multiplication operation on invariant scalars and the steerable features explicitly or concatenate the steerable features with the invariant scalars and implement nonlinear transformations like MLP that can implicitly approximate the tensor product operation in ClofNet.

Tensorized graph neural network with equivariant frame Tensorized graph neural network can be seen as another way of approximating permutation-equivariant polynomials on a graph, such that (5.2) holds. Since the high level idea is the same, we neglect some details in this section.

Following (7) of (Maehara & NT, 2019), the graph is represented by

$$\mathcal{G}_0 = \{W \in \mathbb{R}^{n\times n} : |W(i,j)| \leq 1,\ \forall i,j\},$$

which can be seen as a graph with bounded scalar-valued edge features. To encode tensor field-valued edge features (all node features are identified with edge features) into this representation, we introduce two more indexes $1 \leq u, v \leq m$ for $W(i,j)$, such that

$$W(i,j,u,v) \in \mathbb{R}^2,\ \forall i,j,u,v,$$

where u indicates the u-th component of the tensor field at the i-th node, and v is defined similarly. In other words, $W(i,j,u,v)$ records the u-th component of the tensor field **scalarized by the equivariant frame** associated with the i-th node and the v-th component of the tensor field **scalarized by the equivariant frame** associated with the j-th node. Then under the group action $g \in SO(3)$, SO(3) equivariance requires

$$W(g\cdot x_i, g\cdot x_j, \cdot, \cdot) = gW(x_i, x_j, \cdot, \cdot).$$

Therefore, we should identify W_1 and W_2 if

$$W_1(g\cdot x_i^1, g\cdot x_j^1, \cdot, \cdot) \equiv W_2(x_i^2, x_j^2, \cdot, \cdot), \tag{39}$$

for each $1 \leq i,j \leq n$. We denote $W_1 \sim W_2$ if W_1 and W_2 are isomorphic and there exists $g \in SO(3)$ such that (39) holds. In fact, since the group action on the 3D point cloud in a fiber-wise' way, the permutation of nodes σ is commutative with the SO(3) group g action:

$$W(\sigma(g\cdot x_i), \sigma(g\cdot x_j), \cdot, \cdot) = W(g\cdot\sigma(x_i), g\cdot\sigma(x_j), \cdot, \cdot).$$

We define the SO(3) graph space by $\bar{\mathcal{G}}_0 = \mathcal{G}_0/\sim$. The edit distance ((9) of (Maehara & NT, 2019)) can be extended to include the additional indexes u, v. We further assume that $|W(i,j,u,v)| \leq 1$. Then the SO(3) graph space forms a metric space.

On the other hand, g can also be seen as an orthogonal coordinate transformation from one base frame (e_1, e_2, e_3) to another frame $(g\cdot e_1, g\cdot e_2, g\cdot e_3)$. Now, let g_{ij} denote the coordinate transformation from the standard base frame to the equivariant frame of the edge (i,j). Let

$$\bar{W}_1(i,j,\cdot,\cdot) := W_1(g_{ij}\cdot x_i^1, g_{ij}\cdot x_j^1, \cdot, \cdot).$$

Since the equivariant frame is intrinsically defined with respect to permutation of nodes, we have found a representative element $\bar{W}_1$ for W_1 under the equivalence relation $\sim$ by combining all edges. **Note that similar representative elements can also be defined by projecting the features to the spherical harmonics space (TFN (Thomas et al., 2018)), however, it's not sufficient to project the features into the radical direction only, like EGNN (Satorras et al., 2021a)**.

Suppose we consider equivariant vector field on graph W: $f : W \to \mathbb{R}^{3n}$, then it's easy to construct an edge-wise equivariant vector field such that for each node i,

$$f_i = \sum_{j \in N(i)} f_{ij}.$$

Then under scalarization, each component of f_{ij} becomes SO(3)-invariant scalar function. In this way, to prove the universality approximation property for equivariant vector field on graphs, it's sufficient to prove it on ordinary node-wise continuous functions $f : \bar{\mathcal{G}}_0 \to \mathbb{R}$.

Let $\mathcal{F}$ be the set of simple (no loops) unweighted graphs with 1 labeled node, and let $F = (V(F), E(F)) \in \mathcal{F}$. Let $G = (V(G), E(G); W)$ be a weighted graph, where $W\colon V(G) \times V(G) \to \mathbb{R}$ is the weighted adjacency matrix. for a given node $x \in V(G)$, we define the extended 1-*labeled homomorphism number* by (see (Lovász, 2006; Maehara & NT, 2019))

$$\begin{aligned}\hom_x(F, W, G) = \sum_{\substack{\pi\colon V(F)\to V(G)\\ \pi(1)=x\ (i\in[k])}} \prod_{i\in V(F)} W(\pi(i), \pi(i)) \\ \times \prod_{(i,j)\in E(F)} [\prod_{(u,v)\in G(i,j)} W(\pi(i), \pi(j), u, v)],\end{aligned} \tag{40}$$

where $G = \cup_{(i,j)\in E(G)} G(i,j)$, and $G(i,j)$ is a collection (u,v) such that u is a component of one tensor field on node i and v is a component of one tensor field on node j. From the definition, the extended 1-labeled homomorphism numbers are continuous node-wise functions in W that is equivariant with respect to permutations of nodes. Now we introduce the universal approximation function class:

$$\mathcal{A} = \left\{ W \mapsto \sum_{F\in\mathcal{F}, G\in\mathcal{G}}^{\text{finite}} a_F \hom_x(F, W + 2I, G) : a_F \in \mathbb{R} \right\}, \tag{41}$$

where I denotes the identity matrix of $n \times n$. By utilizing Stone-Weierstrass theorem, we prove the following theorem:

Theorem A.6. *$\mathcal{A}$ is dense in the continuous scalar functions.*

We need to check three conditions before applying Stone-Weierstrass theorem:

- $\mathcal{A}$ forms a subalgebra inside continuous functions;
- $\mathcal{A}$ separates points, i.e., for any $x \neq y$, there exists $h \in \mathcal{A}$ such that $h(x) \neq h(y)$, and
- There exists $u \in \mathcal{A}$ that is bounded away from zero, i.e., $\inf_{x\in\mathcal{X}} |u(x)| > 0$.

Lemma A.7. *$\mathcal{A}'$ forms an algebra.*

Proof. It's straightforward to check the closeness under the addition and the scalar multiplication. It is closed under the product because of the following identity.

$$\hom_x(F_1, W, G_1)\hom_x(F_2, W, G_2) = \hom_x(F_1 \sqcup F_2, W, G_1 \sqcup G_2). \tag{42}$$

where $F_1 \sqcup F_2$ is the graph obtained from the disjoint union of F_1 and F_2 by gluing the labeled vertices. □

Lemma A.8. *$\mathcal{A}$ contains an element that is bounded away from zero.*

Proof. Let $\circ$ be the graph of 1 isolated vertices (singleton). Then, $W \mapsto \hom_x(\circ, W + 2I, \varnothing) \geq n - 1$ is bounded away from zero. □

To prove the separate point property, we use the following theorem.

Theorem A.9 (Lemma 2.4 of (Lovász, 2006))**.** *Let $W_1, W_2 \in \mathbb{R}^{n\times n}$ be matrices with positive diagonal elements. Let x_1, x_2 be two nodes of W_1 and W_2. Then, (W_1, x_1) and (W_2, x_2) are isomorphic if and only if $\hom_{x_1}(F, W_1, \varnothing) = \hom_{x_2}(F, W_2, \varnothing)$ for all simple unweighted graph F.* □

Therefore, two non-isomorphic graphs are separable by $\mathcal{A}$. On the other hand, for two isomorphic graphs with different edge features, we can choose nonempty G to separate them by the above theorem (since polynomials separate different edge features of a given edge).

Lemma A.10. *$\mathcal{A}$ separates points in $\bar{\mathcal{G}}_0$.*

The usual **tensorized graph neural network can express $E(F)$-fold scalar-valued tensor product**. However we can further extend it trivially to include vector-valued features on each edge by enlarging the index space's dimension and performing scalarization, and we denote it by the extended tensorized graph neural network. As it was demonstrated in (Maehara & NT, 2019), the 1-labeled homomorphism numbers can be implemented in a tensorized graph neural network (Keriven & Pe yré, 2019). Then, the extended 1-labeled homomorphism numbers can be implemented in an extended tensorized graph neural network with equivariant bases in the same way. Combining the above, we have proved the universality property of tensorized graph neural network equipped with equivariant bases:

Theorem A.11. *The tensorized graph neural network equipped with equivariant bases has universality in continuous equivariant functions.* □

A.7. Additional Experiments

A.7.1. Newtonian many-body system

Dataset For the Newtonian many-body system experiment, we follow EGNN (Satorras et al., 2021b) to generate the trajectories for four systems. The original source code for generating trajectories comes from (Kipf et al., 2018) (`https://github.com/ethanfetaya/NRI`) and is modified by EGNN (`https://github.com/vgsatorras/egnn`). We further extend the version of EGNN to three new settings, as described in Section 7.1. Similar to EGNN, we generate 5,000 timesteps for each trajectory and slice them from 3,000 to 4,000 to move away from the transient phase. For the second experiment, we generate a new training dataset with 50,000 trajectories for the G+ES system, following the same procedure. The validation and testing datasets remain unchanged from the first experiment. We provide the physical evolution equations mentioned in Section 7.1 as follows.

ES. This system consists of N particles controlled by the electrostatic field, i.e., for each trajectory we are provided with initial positions $\boldsymbol{X}(0) \in \mathbb{R}^{N\times 3}$, velocities $\boldsymbol{V}(0) \in \mathbb{R}^{N\times 3}$ and charges $\{q_1, \cdots q_N\} \in \{-1, 1\}^N$. The time evolution of the particles is given by

$$\ddot{\boldsymbol{x}}_i = \sum_{j\neq i} q_i q_j \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^3}, \quad 1 \leq i \leq N. \tag{43}$$

G+ES. This system consists of N particles controlled by both the electrostatic field and external gravity force of the form: $\boldsymbol{f}_g = (0,\, 0,\, g)$. The time evolution of the particles are given by

$$\ddot{\boldsymbol{x}}_i = \sum_{j\neq i} q_i q_j \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^3} + \boldsymbol{f}_g, \quad 1 \leq i \leq N. \tag{44}$$

L+ES. This system consists of N particles controlled by both the electrostatic field and a Lorentz-like force field, which means there exists a force perpendicular to the direction of velocity $\boldsymbol{v}$, i.e., $\boldsymbol{f}_l(\boldsymbol{v}) = q\boldsymbol{v} \times \mathbf{B}$, where q and $\mathbf{B}$ denote the charge of particles and the direction vector of the electromagnetic field respectively. The time evolution of the particles is given by:

$$\ddot{\boldsymbol{x}}_i = \sum_{j\neq i} [q_i q_j \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^3} + \boldsymbol{f}_l^i(\boldsymbol{v}_i)], \quad 1 \leq i \leq N. \tag{45}$$

Models We implement all baselines (GNN, Radial Field, TFN and SE(3)-Transformer) by referring the codebase of EGNN. The architecture parameters (e.g., feature dimension, activation function) are adapted from EGNN. The learning rate and training epochs are tuned independently for each model.

A.7.2. Partially observed N-body experiment

The N-body dynamical systems considered in the main text can be classified into the category of Markov process, where the system's immediate future state $\boldsymbol{X}(t+\Delta t)$ (future positions and velocities) is fully determined by its current state $\boldsymbol{X}(t)$ (current positions and velocities).In other words, let $\Delta t \to 0$, there exists f such that

$$\dot{\boldsymbol{X}}(t) = f(\boldsymbol{X}(t))$$

where $\dot{\boldsymbol{X}}(t)$ denotes the gradient of $\boldsymbol{X}(t)$, and f is independent of time t.

Now we define a harder trajectory prediction task of a non-Markov dynamical system **POS**, such that $\dot{\boldsymbol{X}}(t) = f(\boldsymbol{X}(t), t)$ to further test ClofNet. **POS** consists of six particles under Newton's gravitation force, but only four of them could be observed, i.e. for the whole trajectory, we are provided with positions $\boldsymbol{X}(t) \in \mathbb{R}^{4\times 3}$ and velocities $\boldsymbol{V}(t) \in \mathbb{R}^{4\times 3}$ of **four** particles. Since another two unobserved particles are hidden, the sub-system of the given four particles is non-Markov. The time evolution of the particles is given by

$$\ddot{\boldsymbol{x}}_i = \sum_{j\neq i} -m_j \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^3}, \quad 1 \leq i \leq 4. \tag{46}$$

Problem definition. In this experiment, we apply our model to predict the long-term motion trajectory of **POS** given its initial ($t = 0$) position and velocity. Following (Zhuang et al., 2020; Li et al., 2021), we formulate the trajectory prediction as two tasks: **Interpolation** and **Extrapolation**. The experimental setting is as follows. To generate the trajectory given the initial condition, we use observations $\boldsymbol{x}_i(t), t \in \{\Delta t, 2\Delta t \ldots, T_1\}$ as the training labels and observations $\boldsymbol{x}_i(t), t \in \{T_1 + \Delta t, T_1 + \Delta t, \ldots, T_2\}$ as the validation set. To evaluate the interpolation and extrapolation capacity of all methods, the observations $\boldsymbol{x}_i(t), t \in \{\frac{1}{2}\Delta t, \frac{3}{2}\Delta t, \ldots, T_1 + \frac{1}{2}\Delta t\}$ and $\boldsymbol{x}_i(t), t \in \{T_2 + \Delta t, T_2 + 2\Delta t, \ldots, T_3\}$ are used as **interpolation** and **extrapolation** test sets respectively. We measure the mean square error (MSE) between the predicted trajectory and ground truth. To measure the exactness of equivariance, we follow (Fuchs et al., 2020) to apply uniformly sampled SO(3)-transformations on the input and output. The MSE between the predicted trajectory with rotated input and rotated ground truth could reflect the transformation robustness of method. The normalized distance between the rotated prediction with original input and the original prediction with the rotated input defines the equivariance error Δ_{EQ}:

$$\Delta_{EQ} = \|L_s\Phi(\boldsymbol{x}) - \Phi L_s(\boldsymbol{x})\| / \|L_s\Phi(\boldsymbol{x})\|, \tag{47}$$

where L_s and Φ denote SO(3) transformations and equivariant neural networks, respectively.

Learning Framework and Implementation Details. Inspired by (Norcliffe et al., 2020), for a Newtonian system, we utilize ClofNet to parameterize its acceleration vector field and adopt numerical ODE solver to integrate both the position and velocity trajectories. Only the MSE between predicted position trajectory and ground truth is taken as the loss penalty:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_2(x_{t_i}, \mathrm{ODE}(\boldsymbol{x}_{t_0}, \boldsymbol{v}_{t_0}, t_0, t_i, \Theta)), \;\; t_0 < t_1 < \cdots < t_n, \tag{48}$$

where $(\boldsymbol{x}_{t_0}, \boldsymbol{v}_{t_0})$ and Θ denote the initial condition of the system and the parameters of ClofNet Φ. We compare our method to another two efficient equivariant models designed for vector field modeling: Radial Field (Köhler et al., 2019) and EGNN (Satorras et al., 2021a;b). Following (Zhuang et al., 2020), the trajectory is simulated using the *Dopri5* solver (Dormand & Prince, 1980) with the tolerance to 10^{-7} and the modified physical rules. The trajectory points are uniformly sampled with $\Delta t = 10^{-3}$. We implement all baselines and our method with Pytorch (Paszke et al., 2019). All models use the same ODE solver (*Dopri5*) as the evolving blocks and are trained with Adam optimizer (Kingma & Ba, 2014) via an MSE loss for 800 epochs. We set the number of layers to 2 for all models and adjust the hidden dimensions of each model separately to keep the parameters in the same level. We adopt EGNN from (Satorras et al., 2021a) for outputting vectors.

Results. From Table 5, we have the following conclusions: (1) ClofNet outperforms all other default equivariant methods in the interpolation and extrapolation tasks, which demonstrates that ClofNet exhibits stronger expressive power by representing geometric information losslessly; (2)The equivariance error Δ_{EQ} of the three models are small (Due to the existence of numerical errors, Δ_{EQ} cannot be strictly zero), which empirically demonstrate the equivariance of these models.

Table 5. MSE of the **POS** dataset. *Inter.* and *Extra.* denote the interpolation and extrapolation task respectively.

Setting	Method	Inter.	Extra.	Δ_{EQ}
POS	Radial Field	1.996	7.665	$5.77 \cdot 10^{-6}$
	EGNN	0.726	6.449	$9.19 \cdot 10^{-7}$
	ClofNet	**0.138**	**2.502**	$4.13 \cdot 10^{-6}$

A.7.3. Molecular conformation generation

Dataset For each dataset, $40,000$ molecules are randomly drawn and 5 most likely conformations (sorted by energy) are selected for each molecule, and 200 molecules are drawn from the remaining data, which results in $200,000$ conformations in the training set, $22,408$ and $14,324$ conformations in the test set for GEOM-QM9 and GEOM-Drugs datasets, respectively. The distances over distributions task are evaluated on the ISO17 dataset, where we follow the setup in (Simm & Hernández-Lobato, 2019).

Learning Framework Following (Shi et al., 2021), for this first-order statistical ensemble system, we leverage a score-based generative modeling framework to estimate the gradient field of atomic positions (See more details about score-based networks in (Shi et al., 2021; Song et al., 2020) or Appendix A.3.2). A detailed illustration on the **equivariance** of the gradient fields (the score function) is also given in Appendix A.3.2. The optimization objective of ClofNet Φ can be summarized as:

$$\begin{aligned}\mathcal{L}(\theta) =& \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{X}(0)}\{\lambda(t_i)\|\nabla H_{t_i}(\boldsymbol{X}(t_i)) \\ &- \Phi(\boldsymbol{X}(t_i), t_i, \Theta)\|_2^2, t_0 < t_1 < \cdots < t_n,\end{aligned} \tag{49}$$

where $\lambda(t) : [0, T] \to \mathbb{R}^+$ is a positive weighting function and ∇H_{t_i} is the pre-computed gradient field of noisy atomic positions (see Appendix A.3.2). Once the score network is optimized, we can use an annealed Langevin dynamics (ALD) sampler to generate conformations (Song & Ermon, 2019; Song et al., 2020).

Implementation Details Besides the geometric input, we feed the node type, edge type and relative distances as extra node/edge attributes into the graph transformer block. Our score-based training framework keeps the same as (Shi et al., 2021). The maximum and minimum noise scales are set to 10 and 0.01. Let $\{\sigma_i\}_{i=1}^{L}$ be a positive geometric progression scheme with a common ratio, we split the noise range into 50 levels. For the reverse process, we follow ConfGF to use the annealed Langevin dynamics sampling method to generate stable structures. The sample step size η_s is chosen according to (Song & Ermon, 2020). All hyper-parameters mentioned in the forward and reverse process are kept the same as (Shi et al., 2021). The results reported in Table 2 are copied from (Shi et al., 2021) considering that we rigorously evaluate ClofNet on the same benchmark and data split setting.

Conformation Generation Here we introduce the calculation equation of RMSD:

$$\text{RMSD}(R, \hat{R}) = \min(\frac{1}{n}\sum_{i=1}^{n}||R_i - \hat{R}_i||^2)^{\frac{1}{2}}, \tag{50}$$

where n denotes the number of heavy atoms.

We visualize several conformations in the Drugs dataset in Figure 4 that are best aligned with the reference ones generated by different methods, illustrating ClofNet's superior capacity on generating high-quality drug molecular conformation.

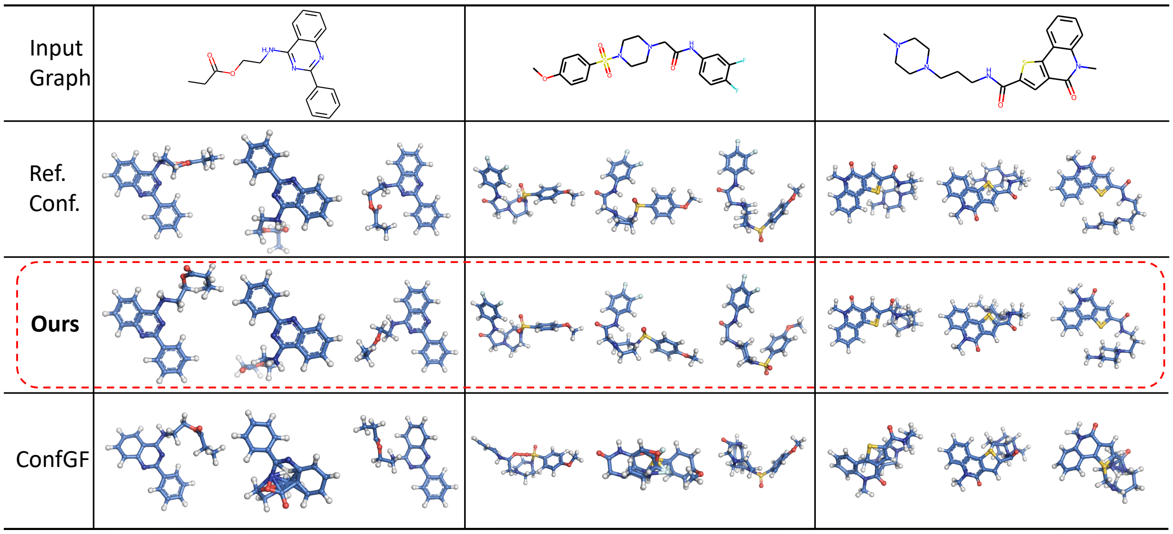

Figure 4. Visualizations of generated conformations. For each molecule randomly selected from GEOM-Drugs dataset, we sample multiple conformations and show the best-aligned ones with the reference ones.