A new perspective on building efficient and expressive 3D equivariant graph neural networks

Weitao Du[1*] **Yuanqi Du**[2*] **Limei Wang**[3*] **Dieqiao Feng**[2] **Guifeng Wang**[4]

Shuiwang Ji[3] **Carla P Gomes**[2] **Zhi-Ming Ma**[1]

[1] Chinese Academy of Sciences
[2] Cornell University
[3] Texas A&M University
[4] Zhejiang University

Abstract

Geometric deep learning enables the encoding of physical symmetries in modeling 3D objects. Despite rapid progress in encoding 3D symmetries into Graph Neural Networks (GNNs), a comprehensive evaluation of the expressiveness of these networks through a local-to-global analysis lacks today. In this paper, we propose a local hierarchy of 3D isomorphism to evaluate the expressive power of equivariant GNNs and investigate the process of representing global geometric information from local patches. Our work leads to two crucial modules for designing expressive and efficient geometric GNNs; namely local substructure encoding (**LSE**) and frame transition encoding (**FTE**). To demonstrate the applicability of our theory, we propose LEFTNet which effectively implements these modules and achieves state-of-the-art performance on both scalar-valued and vector-valued molecular property prediction tasks. We further point out the design space for future developments of equivariant graph neural networks. Our codes are available at `https://github.com/yuanqidu/LeftNet`.

1 Introduction

The success of many deep neural networks can be attributed to their ability to respect physical symmetry, such as Convolutional Neural Networks (CNNs) [1] and Graph Neural Networks (GNNs) [2]. Specifically, CNNs encode translation equivariance, which is essential for tasks such as object detection. Similarly, GNNs encode permutation equivariance, which ensures that the node ordering does not affect the output node representations. , by aggregating neighboring messages. Modeling 3D objects, such as point clouds and molecules, is a fundamental problem with numerous applications, including robotics [3], molecular simulation [4, 5], and drug discovery [6–10]. Different from 2D pictures and graphs that only possess the translation [1] and permutation [2] symmetry, 3D objects intrinsically encode the complex $SE(3)/E(3)$ symmetry [11], which makes their modeling a nontrivial task in the machine learning community.

To tackle this challenge, several approaches have been proposed to effectively encode 3D rotation and translation equivariance in the deep neural network architectures, such as TFN [12], EGNN [13], and SphereNet [14]. TFN leverages spherical harmonics to represent and update tensors equivariantly, while EGNN processes geometric information through vector update. On the other hand, SphereNet is invariant by encoding scalars like distances and angles. Despite rapid progress has

[*]Equal contribution.

Preprint. Under review.

been made on the empirical side, it's still unclear what 3D geometric information can equivariant graph neural networks capture and how the geometric information is integrated during the message passing process [15–17]. This type of analysis is crucial in designing expressive and efficient 3D GNNs, as it's usually a trade-off between encoding enough geometric information and preserving relatively low computation complexity. Put aside the $SE(3)/E(3)$ symmetry, this problem is also crucial in analysing ordinary GNNs. For example, 1-hop based message passing graph neural networks [18] are computationally efficient while suffering from expressiveness bottlenecks (comparing with subgraph GNNs [19, 20]). On the other hand, finding a better trade-off for 3D GNNs is more challenging, since we must ensure that the message updating and aggregating process respects the $SE(3)/E(3)$ symmetry.

In this paper, we attempt to discover better trade-offs between computational efficiency and expressiveness power for 3D GNNs by studying two specific questions: 1. What is the expressive power of invariant scalars in encoding 3D geometric patterns? 2. Is equivariance really necessarily for 3D GNNs? The first question relates to the design of node-wise geometric messages, and the second question relates to the design of equivariant (or invariant) aggregation. To tackle these two problems, we take a local-to-global approach. More precisely, we first define three types of 3D isomorphism to characterize local 3D structures: tree, triangular, and subgraph isomorphism, following a local hierarchy. As we will discuss in the related works section, our local hierarchy lies between the 1-hop and 2-hop geometric isomorphism defined in [21]. Then, we can measure the expressiveness power of 3D GNNs by their ability of differentiating non-isomorphic 3D structures in a similar way as the geometric WL tests in [21]. Under this theoretical framework, we summarize one essential ingredient for building expressive geometric messages on each node: local 3D substructure encoding (**LSE**), which allows an invariant realization. To answer the second question, we analyze whether local invariant features are sufficient for expressing global geometries by message aggregation, and it turns out that frame transition encoding (**FTE**) is crucial during the local to global process. Although **FTE** can be realized by invariant scalars, we further demonstrate that introducing equivariant messaging passing is more efficient. By connecting **LSE** and **FTE** modules, we are able to present a modular overview of 3D GNNs designs.

In realization of our theoretical findings, we propose LEFTNet that efficiently implements **LSE** and **FTE** (with equivariant tensor update) without sacrificing expressiveness. Empirical experiments on real-world scenarios, predicting scalar-valued property (e.g. energy) and vector-valued property (e.g. force) for molecules, demonstrate the effectiveness of LEFTNet.

2 Preliminary

In this section, we provide an overview of the mathematical foundations of $E(3)$ and $SE(3)$ symmetry, which is essential in modeling 3D data. We also summarize the message passing graph neural network framework, which enables the realization of $E(3)/SE(3)$ equivariant models.

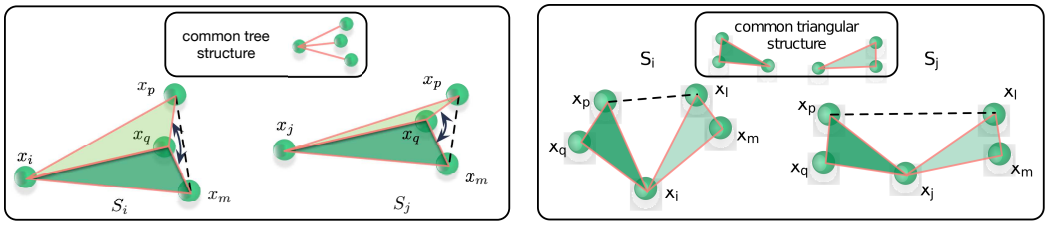
Euclidean Symmetry. Our target is to incorporate Euclidean symmetry to ordinary permutation-invariant graph neural networks. The formal way of describing Euclidean symmetry is the group $E(3) = O(3) \rtimes T(3)$, where $O(3)$ corresponds to reflections (parity transformations) and rotations. For tasks that are anti-symmetric under reflections (e.g. chirality), we consider the subgroup $SE(3) = SO(3) \rtimes T(3)$, where $SO(3)$ is the group of rotations. We will use $SE(3)$ in the rest of the paper for brevity except when it's necessary to emphasize reflections.

Equivariance. A tensor-valued function $f(\mathbf{x})$ is said to be **equivariant** with respect to $SE(3)$ if for any translation or rotation $g \in SE(3)$ acting on $\mathbf{x} \in \mathbf{R}^3$, we have

$$f(g\mathbf{x}) = \mathcal{M}(g)f(\mathbf{x}),$$

where $\mathcal{M}(\cdot)$ is a matrix representation of $SE(3)$ acting on tensors. See Appendix A for a general definition of tensor fields. In this paper, we will use **bold** letters to represent an equivariant tensor, e.g., $\mathbf{x}$ as a position vector. It is worth noting that when $f(\mathbf{x}) \in \mathbf{R}^1$ and $\mathcal{M}(g) \equiv 1$ (the constant group representation), the equivariant function $f(\mathbf{x})$ is also called an **invariant** scalar function.

Scalarization. Scalarization is a general technique that originated from differential geometry for realizing covariant operations on tensors [22]. Our method will apply a simple version of scalarization in $\mathbf{R}^3$ to transform equivariant quantities. At the heart of its realization is the notion of equivariant



(a) (b)

Figure 1: (a) $\mathbf{S}_i$ and $\mathbf{S}_j$ share the same tree structure (edge lengths are identical), but they are not triangular isomorphic (different dihedral angles); (b) $\mathbf{S}_i$ and $\mathbf{S}_j$ are triangular isomorphic but not subgraph isomorphic (the relative distance between the two triangles is different).

orthonormal frames, which consist of three orthonormal equivariant vectors:

$$\mathcal{F} := (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3).$$

Based on $\mathcal{F}$, we can build orthonormal equivariant frames for higher order tensors by taking tensor products $\otimes$, see Eq. 19 in Appendix. By taking the inner product between $\mathcal{F}$ and a given equivariant vector (tensor) $\mathbf{x}$, we get a tuple of invariant scalars (see [23] for a proof):

$$\mathbf{x} \to \tilde{x} := (\mathbf{x} \cdot \mathbf{e}_1, \mathbf{x} \cdot \mathbf{e}_2, \mathbf{x} \cdot \mathbf{e}_3), \tag{1}$$

and $\tilde{x}$ can be seen as the 'scalarized' coordinates of $\mathbf{x}$.

Tensorization. Tensorization, on the other hand, is the 'reverse' process of scalarization. Given a tuple of scalars: (x_1, x_2, x_3), tensorization creates an equivariant vector (tensor) out of $\mathcal{F}$:

$$(x_1, x_2, x_3) \xrightarrow{\text{Pairing}} \mathbf{x} := x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3. \tag{2}$$

The same procedure is extended to higher order cases, see Eq. 20 in Appendix.

Message Passing Scheme for Geometric Graphs. A geometric graph G is represented by $G = (V, E)$. Here, $v_i \in V$ denotes the set of nodes (vertices, atoms), and $e_{ij} \in E$ denotes the set of edges. For brevity, the edge feature attached on e_{ij} is also denoted by e_{ij}. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbf{R}^{n\times 3}$ be the 3D point cloud of all nodes' equivariant positions, which determines the 3D geometric structure of G.

A common machine learning tool for modeling graph-structured data is the Message Passing Neural Network (MPNN) [15]. A typical 1-hop MPNN framework consists of two phases: (1) message passing; (2) readout. Let h_i^l, h_j^l denote the l-th layer's node features of source i and target j that also depend on the 3D positions $(\mathbf{x}_i, \mathbf{x}_j)$, then the aggregated message is

$$m_i^l = \bigoplus_{j\in\mathcal{N}(i)} m_{ij}(h^l(\mathbf{x}_i), h^l(\mathbf{x}_j), e_{ij}^l), \tag{3}$$

and $\bigoplus_{j\in\mathcal{N}(i)}$ is any permutation-invariant pooling operation between the 1-hop neighbors of i. We also include the edge features e_{ij}^l into the message passing phase for completeness. 3D **equivariant** MPNNs (3D GNNs for short) require the message m_i to be equivariant with respect to the geometric graph. That is, for an arbitrary edge e_{ij}:

$$m_{ij}(h^l(g\mathbf{x}_i), h^l(g\mathbf{x}_j)) = \mathcal{M}(g)m_{ij}(h^l(\mathbf{x}_i), h^l(\mathbf{x}_j)), \tag{4}$$

where $g \in SE(3)$ is acting on the whole geometric graph simultaneously: $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \to (g\mathbf{x}_1, \ldots, g\mathbf{x}_n)$. For example, the invariant model ComENet [24] satisfies Eq. 4 by setting $\mathcal{M}(g) \equiv 1$, and MACE [25] realized Eq. 4 for nonconstant irreducible group representations $\mathcal{M}(g)$ through spherical harmonics and Clebsch-Gordan coefficients.

3 A Local Hierarchy of 3D Isomorphism

As presented in Section 2, defining expressive messages is an essential component for building powerful 3D GNNs. In this section, we develop a fine-grained characterization of local 3D structures and build its connection with the expressiveness of 3D GNNs.

Since the celebrated work [26], a popular expressiveness test for permutation invariant graph neural networks is the 1-WL graph isomorphism test [27], and Wijesinghe and Wang [28] has shown that the 1-WL test is equivalent to the ability to discriminate the **local** subtree-isomorphism. It motivates us to develop a novel (local) 3D isomorphism for testing the expressive power of 3D GNNs. However, this task is nontrivial, since most of the previous settings for graph isomorphism are only applicable to 2D topological features. For 3D geometric shapes, we should take the $SE(3)$ symmetry into account. Formally, two 3D geometric graphs $\mathbf{X}, \mathbf{Y}$ are defined to be **globally** isomorphic, if there exists $g \in SE(3)$ such that

$$\mathbf{Y} = g\mathbf{X}. \tag{5}$$

In other words, $\mathbf{X}$ and $\mathbf{Y}$ are essentially the same, if they can be transformed into each other through a series of rotations and translations. Inspired by Wijesinghe and Wang [28], now we introduce a novel hierarchy of $SE(3)$ equivariant local isomorphism to measure the local similarity of 3D structures.

Let $\mathbf{S}_i$ denote the 3D subgraph (and the associated node features) of node i, which contains all edges in E if the end points are one-hop neighbors of i. For each edge $e_{ij} \in E$, the mutual 3D substructure $\mathbf{S}_{i-j}$ is defined by the intersection of $\mathbf{S}_i$ and $\mathbf{S}_j$: $\mathbf{S}_{i-j} = \mathbf{S}_i \cap \mathbf{S}_j$.

Given two local subgraphs $\mathbf{S}_i$ and $\mathbf{S}_j$ that correspond to two nodes i and j, we say $\mathbf{S}_i$ is $\{$-tree, -triangular, -subgraph$\}$ isometric to $\mathbf{S}_j$, if there exists a bijective function $f : \mathbf{S}_i \to \mathbf{S}_j$ such that $h_{f(u)} = h_u$ for every node $u \in \mathbf{S}_i$, and the following conditions hold respectively:

- **Tree Isometric:** If there exists a collection of group elements $g_{iu} \in SE(3)$, such that $(\mathbf{x}_{f(u)}, \mathbf{x}_{f(i)}) = (g_{iu}\mathbf{x}_u, g_{iu}\mathbf{x}_i)$ for each edge $e_{iu} \in \mathbf{S}_i$;
- **Triangular Isometric:** If there exists a collection of group elements $g_{iu} \in SE(3)$, such that the corresponding mutual 3D substructures satisfy: $\mathbf{S}_{f(u)-f(i)} = g_{iu}\mathbf{S}_{u-i}$ for each edge $e_{iu} \in \mathbf{S}_{i-j}$;
- **Subgraph Isometric:** for any two adjacent nodes $u, v \in \mathbf{S}_i$, $f(u)$ and $f(v)$ are also adjacent in $\mathbf{S}_j$, and there exist a single group element $g_i \in SE(3)$ such that $g_i\mathbf{S}_i = \mathbf{S}_j$.

Note that tree isomorphism only considers edges around a central node, which is of a tree shape. On the other hand, the mutual 3D substructure can be decomposed into a bunch of triangles (since it's contained in adjacent node triplets), which explains the name of triangular isomorphism.

In fact, the three isomorphisms form a hierarchy from micro to macro, in the sense that the following implication relation holds:

Subgraph Isometric $\Rightarrow$ **Triangular Isometric** $\Rightarrow$ **Tree Isometric**

This is an obvious fact from the above definitions. To deduce the reverse implication relation, we provide a visualized example. Figure 1 shows two examples of local 3D structures: 1. the first one shares the same tree structure, but is not triangular-isomorphic; 2. the second one is triangular-isomorphic but not subgraph-isomorphic. In conclusion, the following diagram holds:

Tree Isometric $\not\Rightarrow$ **Triangular Isometric** $\not\Rightarrow$ **Subgraph Isometric**

One way to formally connect the expressiveness power of a geometric GNN with their ability of differentiating geometric subgraphs is to define geometric WL tests, the reader can consult [21]. In this paper, we take an intuitive approach based on our nested 3D hierarchy. That is, if two 3D GNN algorithms A and B can differentiate all non-isomorphic local 3D shapes of tree (triangular) level, while A can differentiate at least two more 3D geometries which are non-isomorphic at triangular(subgraph) level than B, then we claim that algorithm A's expressiveness power is more powerful than B.

Since tree isomorphism is determined by the one-hop Euclidean distance between neighbors, distinguishing local tree structures is relatively simple for ordinary 3D equivariant GNNs. For example, the standard baseline SchNet [29] is one instance of Eq. 3 by setting $e_{ij}^t = \mathbf{RBF}(d(\mathbf{x}_i, \mathbf{x}_j))$, where $\mathbf{RBF}(\cdot)$ is a set of radial basis functions. Although it is powerful enough for testing tree non-isomorphism (assuming that $\mathbf{RBF}(\cdot)$ is injective), we prove in Appendix B that SchNet cannot distinguish non-isomorphic structures at the triangular level.

On the other hand, Wijesinghe and Wang [28] has shown that by leveraging the topological information extracted from local overlapping subgraphs, we can enhance the expressive power of GNNs

to go beyond 2D sub-tree isomorphism. In our setting, the natural analogue of the overlapping subgraphs is exactly the mutual 3D substructures. Now we demonstrate how to merge the information from 3D substructures to the message passing framework (3). Given an $SE(3)$-**invariant** encoder ϕ, define the 3D structure weights $A_{ij} := \phi(\mathbf{S}_{i-j})$ for each edge $e_{ij} \in E$. Then, the message passing framework (3) is generalized to:

$$m_i^l = \bigoplus_{j \in \mathcal{N}(i)} m_{ij}(h^l(\mathbf{x}_i), h^l(\mathbf{x}_j), A_{ij}h^l(\mathbf{x}_j), e_{ij}^l). \tag{6}$$

Formula 6 is an efficient realization of enhancing 3D GNNs by injecting the mutual 3D substructures. However, a crucial question remains to be answered: *Can the generalized message passing framework boost the expressive power of 3D GNNs?* Under certain conditions, the following theorem provides an affirmative answer:

Theorem 3.1. *Suppose ϕ is a a universal SE(3)-invariant approximator of functions with respect to the mutual 3d structures $\boldsymbol{S}_{i-j}$, then the collection of weights $\{\{A_{ij}\}_{e_{ij}\in E}\}$ is able to differentiate local structures beyond tree isomorphism. Moreover, with additional injectivity assumptions (see Eq. 14), 3D GNNs based on the enhanced message passing framework 6 map at least two distinct local 3D subgraphs with isometric local tree structures to different representations.*

This theorem confirms that the enhanced 3D GNN (formula 6) is more expressive than the SchNet baseline, at least in testing local non-isomorphic geometric graphs. The complete proof is left in Appendix B. The existence of such local invariant encoder ϕ is also proved by explicit construction. Note that there are other different perspectives on characterizing 3D structures, we will also briefly discuss them in Appendix B.

4 From Local to Global: The Missing Pieces

In the last section, we introduced a geometric local isomorphism hierarchy for testing the expressive power of 3D GNNs. Furthermore, we motivated adding a SE(3)-**invariant** encoder to improve the expressive power of one-hop 3D GNNs by scalarizing not only pairwise distances but also their mutual 3D structures in Theorem 3.1. However, to build a powerful 3D GNN, it remains to be analyzed how a 3D GNN acquires higher order (beyond 1-hop neighbors) information by accumulating local messages. A natural question arises: *are invariant features enough for representing **global** geometric information?*

To formally formulate this problem, we consider a two-hop aggregation case. From figure 2, the central atom a is connected with atoms b and c. Except for the common neighbor a, other atoms that connect to b and c form two 3D clusters, denoted by **B**, **C**. Suppose the ground-truth interaction potential of **B** and **C** imposed on atom a is described by a tensor-valued function $f_a(\mathbf{B}, \mathbf{C})$. Since **B** and **C** are both beyond the 1-hop neighborhood of a, the information of $f_a(\mathbf{B}, \mathbf{C})$ can only be acquired after two steps of message passing: 1. atoms b and c aggregate message separately from **B** and **C**; 2. the central atom a receives the aggregated message (which contains information of **B** and **C**) from its neighbors b and c.

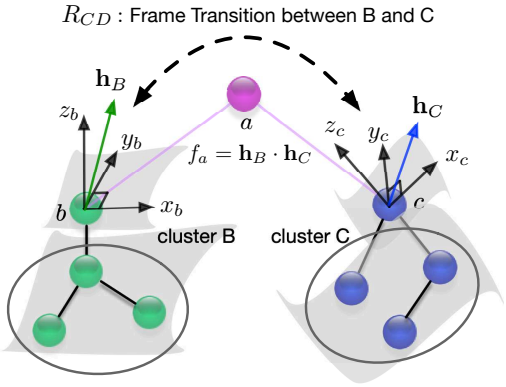


Figure 2: Illustrations of different local frames and their transition.

Let $S_{\mathbf{B}}$ ($S_{\mathbf{C}}$) denote the collection of all invariant scalars created by **B** (**C**) . For example, $S_{\mathbf{B}}$ contains all relative distances and angles within the 3D structure **B**. Then, the following theorem holds:

Theorem 4.1. *Not all types of invariant interaction $f_a(\boldsymbol{B}, \boldsymbol{C})$ can be expressed by inputting the union of two sets $S_{\boldsymbol{B}}$ and $S_{\boldsymbol{C}}$. In other words, there exists $E(3)$ invariant function $f_a(\boldsymbol{B}, \boldsymbol{C})$, such that it cannot be expressed as functions of $S_{\boldsymbol{B}}$ and $S_{\boldsymbol{C}}$: $f_a(\boldsymbol{B}, \boldsymbol{C}) \neq \rho(S_{\boldsymbol{B}}, S_{\boldsymbol{C}})$ for an arbitrary invariant function ρ.*

This theorem in essence tells us that naively aggregating 'local' scalar information from different clusters is not enough to approximate 'global' interactions, even if we only consider simple **invariant** interaction tasks. Different from the last section, where the local expressiveness is measured by the ability of classifying geometric shapes, we built regression functions that depend strictly more than the combination of local invariant scalars.

Intuitively, the proof is based on the fact that all scalars in $S_{\mathbf{B}}$ ($S_{\mathbf{C}}$) can be expressed through equivariant frames separately determined by **B** (**C**). However, the transition matrix between these two frames is not encoded in the aggregation, which causes **information loss** when aggregating geometric features from two sub-clusters. More importantly, the proof also revealed the missing information that causes the expressiveness gap: Frame Transition (FT).

Frame Transition (FT). Formally, two orthonormal frames (e_1^i, e_2^i, e_3^i) and (e_1^j, e_2^j, e_3^j) are connected by an orthogonal matrix $R_{ij} \in SO(3)$:

$$(\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i) = R_{ij}(\mathbf{e}_1^j, \mathbf{e}_2^j, \mathbf{e}_3^j). \tag{7}$$

Moreover, it is easy to check that when $(\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i)$ and $(\mathbf{e}_1^j, \mathbf{e}_2^j, \mathbf{e}_3^j)$ are equivariant frames, all elements of R_{xy} are invariant scalars. Suppose i and j represent indexes of two connected atoms in a geometric graph, then the fundamental torsion angle τ_{ij} appeared in ComeNet [24] is just one element of R_{ij} (see Appendix C).

Towards filling this expressiveness gap, we can straightforwardly inject all invariant pairwise frame transition matrices (**FT**) into the model. Nevertheless, it imposes expensive computational cost when the number of local clusters is large ($O(k^2)$ pairs of **FT** for each node). Therefore, compared with pure invariant approaches, a more efficient way is to introduce equivariant tensor features for each node i, denoted by $\mathbf{m}_i$. By directly maintaining the equivariant frames in $\mathbf{m}_i$, we show in Appendix C that **FT** is easily derived through equivariant message passing.

Equivariant Message Passing. Similarly with the standard one-hop message passing scheme 3, the aggregated tensor message $\mathbf{m}_i$ from the $l-1$ layer to the l layer can be written as: $\mathbf{m}_i^{l-1} = \sum_{j \in N(i)} \mathbf{m}_j^{l-1}$. Since summation does not break the symmetry rule, it is obvious that $\mathbf{m}_i^{l-1}$ are still equivariant tensors. However, the nontrivial part lies in the design of the equivariant update function ϕ:

$$\mathbf{m}_i^l = \phi(\mathbf{m}_i^{l-1}). \tag{8}$$

A good ϕ should have enough expressive power while preserving $SE(3)$ equivariance. Here, we propose a novel way of updating scalar and tensor messages by performing node-wise scalarization and tensorization blocks (the **FTE** module of Figure 3). From the perspective of Eq. 4, $\mathbf{m}(\mathbf{x}_u)$ is transformed equivariantly as:

$$\mathbf{m}(g\mathbf{x}_u) = \sum_{i=0}^{l} \mathcal{M}^i(g)\mathbf{m}_i(g\mathbf{x}_u), \;\; g \in SE(3). \tag{9}$$

Here, $\mathbf{m}(\mathbf{x}_u)$ is decomposed to $(\mathbf{m}_0(\mathbf{x}_u), \ldots, \mathbf{m}_l(\mathbf{x}_u))$ according to different tensor types, and $\{\mathcal{M}^i(g)\}_{i=0}^l$ is a collection of different $SE(3)$ **tensor representations** (see the precise definition in Appendix A).

To illustrate the benefit of aggregating equivariant messages from local patches, we study a simple case. Let $f_a(\mathbf{B}, \mathbf{C}) = \mathbf{h}_B \cdot \mathbf{h}_C$ be an invariant function of **B** and **C** (see Fig. 2), then f_a can be calculated by a direction composition of scalar messages and equivariant vector messages: $f_a(\mathbf{B}, \mathbf{C}) = \frac{1}{2}[\|\mathbf{m}_a\|^2 - \|\mathbf{h}_B\|^2 - \|\mathbf{h}_C\|^2]$, where $\mathbf{m}_a = \mathbf{h}_B + \mathbf{h}_C$ is an equivariant vector. Note that $\mathbf{m}_a$ follows the local equivariant aggregation formula 8, and the other vectors' norm $\|\mathbf{h}_B\|$ and $\|\mathbf{h}_C\|$ are obtained through local scalarization on atoms b and c. As a comparison, it's worth mentioning that $f_a(\mathbf{B}, \mathbf{C})$ can also be expressed by local scalarization with the additional transition matrix data R_{BC} defined by Eq. 7. Let $\tilde{h}_B$ and $\tilde{h}_C$ be the scalarized coordinates with respect to two local equivariant frames $\mathcal{F}_B$ and $\mathcal{F}_C$. Then $f_a(\mathbf{B}, \mathbf{C}) = \frac{1}{2}\left[\left\|R_{BC}^{-1}\tilde{h}_B + \tilde{h}_C\right\|^2 - \left\|\tilde{h}_B\right\|^2 - \left\|\tilde{h}_C\right\|^2\right]$. However, it requires adding the rotation matrix R_{BC} for each $(\mathbf{B}, \mathbf{C})$ pair, which is computationally expensive compared to directly implementing equivariant tensor updates.

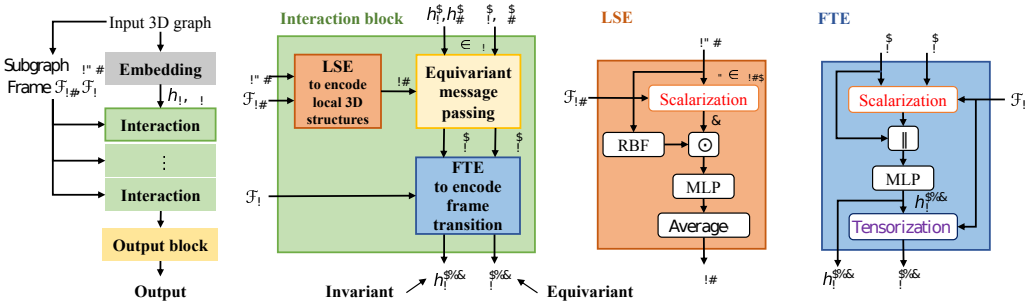


Figure 3: Illustrations of our modular framework for building equivariant GNNs and the realization of LEFTNet. Each interaction block contains **LSE** to encode local 3D structures, equivariant message passing to update both invariant (unbold letters, e.g. h_i) and equivariant (**bold** letter, e.g. $\mathbf{h}_i$) features, and **FTE** to encode frame transition. $\mathbf{S}_{i-j}$ is the local 3D structure of each edge e_{ij}. $\mathcal{F}_{ij}$ and $\mathcal{F}_i$ are the equivariant frames for each edge e_{ij} and node i. $\odot$ indicates element-wise multiplication, and $\|$ indicates concatenation. Note that we do not include $\mathbf{e}_{ij}$ in the figure since, practically, they are generated based on $\mathbf{h}_i$ and $\mathbf{h}_j$.

5 Building an Efficient and Expressive Equivariant 3D GNN

We propose to leverage the full power of **LSE** and **FTE** along with a powerful **tensor update** module to push the limit of efficient and expressive 3D equivariant GNNs design.

LSE Instantiation. We propose to apply edge-wise equivariant frames to encode the local 3D structures $\mathbf{S}_{i-j}$. By definition, $\mathbf{S}_{i-j}$ contains edge e_{ij}, nodes i and j, and their common neighbors. We use the equivariant frame $\mathcal{F}_{ij}$ built on e_{ij} (see the precise formula in Appendix D) to scalarize $\mathbf{S}_{i-j}$. After scalarization (1), the equivariant coordinates of all nodes in $\mathbf{S}_{i-j}$ are transformed into invariant coordinates: $\{\mathbf{x}_k \to \tilde{x}_k \text{ for } \mathbf{x}_k \in \mathbf{S}_{i-j}\}$. To encode these scalars sufficiently, we first weight each $\tilde{x}_k$ by the **RBF** distance embedding: $\tilde{x}_k \to \textbf{RBF}(\|\mathbf{x}_k\|) \odot \text{MLP}(\tilde{x}_k)$ for each $\mathbf{x}_k \in \mathbf{S}_{i-j}$. Note that to preserve the permutation symmetry, the MLP is shared among the nodes. Finally, the 3D structure weight A_{ij} is obtained by the average pooling of all node features.

FTE Instantiation. We propose to introduce equivariant tensor message passing and update function for encoding local **FT** information. At initialization, let $\textbf{NF}^l(\mathbf{x}_i, \mathbf{x}_j)$ denote the embedded tensor-valued edge feature between i and j. We split it into two parts: 1. the scalar part $\text{SF}^l(\mathbf{x}_i, \mathbf{x}_j)$ for aggregating invariant messages; 2. the higher order tensor part $\textbf{TF}^l(\mathbf{x}_i, \mathbf{x}_j)$ for aggregating tensor messages. To transform $\textbf{TF}^l(\mathbf{x}_i, \mathbf{x}_j)$, we turn to the equivariant frame $\mathcal{F}_{ij}$ once again. After scalarization by $\mathcal{F}_{ij}$, $\textbf{TF}^l(\mathbf{x}_i, \mathbf{x}_j)$ becomes a tuple of scalars $\tilde{\text{TF}}^l(\mathbf{x}_i, \mathbf{x}_j)$, which is then transformed by MLP. Finally, we output arbitrary tensor messages through equivariant tensorization 20:

$$\tilde{\text{TF}}^l(\mathbf{x}_i, \mathbf{x}_j) \xrightarrow[\mathcal{F}_{ij}]{\textbf{Tensorize}} \textbf{NF}^{l+1}(\mathbf{x}_i, \mathbf{x}_j).$$

Further details are provided in Appendix D. As we have discussed earlier, the node-wise tensor update function ϕ in Eq. 8 is also one of the guarantees for a powerful **FTE**. As a comparison, ϕ is usually a standard MLP for updating node features in 2D GNNs, which is a **universal approximator** of invariant functions. Previous works [13, 30] updated equivariant features by taking linear combinations and calculating the invariant norm of tensors, which may suffer from information loss. Then a natural question arises: *Can we design an equivariant universal approximator for tensor update?* We answer this question by introducing a novel node-wise frame. Consider node i with its position $\mathbf{x}_i$, let $\bar{\mathbf{x}}_i := \frac{1}{N}\sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \mathbf{x}_j$ be the center of mass around $\mathbf{x}_i$'s neighborhood. Then the orthonormal equivariant frame $\mathcal{F}_i := (\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i)$ with respect to $\mathbf{x}_i$ is defined by

$$\left(\frac{\mathbf{x}_i - \bar{\mathbf{x}}_i}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|}, \frac{\bar{\mathbf{x}}_i \times \mathbf{x}_i}{\|\bar{\mathbf{x}}_i \times \mathbf{x}_i\|}, \frac{\mathbf{x}_i - \bar{\mathbf{x}}_i}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|} \times \frac{\bar{\mathbf{x}}_i \times \mathbf{x}_i}{\|\bar{\mathbf{x}}_i \times \mathbf{x}_i\|}\right). \tag{10}$$

Finally, we realize a powerful ϕ by the following theorem:

Theorem 5.1. *Equipped with an equivariant frame $\mathcal{F}_i$ for each node i, the equivariant function ϕ defined by the following composition is a universal approximator of tensor transformations: ϕ : **Scalarization** $\to$ **MLP** $\to$ **Tensorization**.*

Table 1: Categorization of representative geometric GNN algorithms. * denotes partially satisfying the requirement.

Method	Symmetry	LSE	FTE	Complexity
SchNet [29]	E(3)-invariant	✗	✗	$O(nk)$
EGNN [13]	E(3)-equivariant	✗	✓*	$O(nk)$
GVP-GNN [30]	E(3)-equivariant	✗	✓	$O(nk)$
ClofNet [23]	SE(3)-equivariant	✗	✗	$O(nk)$
PaiNN [31]	E(3)-equivariant	✗	✓	$O(nk)$
ComENet [24]	SE(3)-invariant	✓	✓*	$O(nk)$
TFN [12]	SE(3)/E(3)-equivariant	✗	✓	$O(nk)$
Equiformer [32]	SE(3)/E(3)-equivariant	✗	✓	$O(nk)$
SphereNet [14]	SE(3)-invariant	✓*	✓*	$O(nk^2)$
GemNet [33]	SE(3)-invariant	✓*	✓*	$O(nk^3)$
LEFTNet (Ours)	SE(3)/E(3)-equivariant	✓	✓	$O(nk)$

The proof is left in Appendix D.

LEFTNet. An overview of our $\{\textbf{LSE}, \textbf{FTE}\}$ enhanced efficient graph neural network (LEFTNet) is depicted in Figure 3. LEFTNet receives as input a collection of node embeddings $\{v_1^0, \ldots, v_N^0\}$, which contain the atom types and 3D positions for each node: $v_i^0 = (z_i, \mathbf{x}_i)$, where $i \in \{1, \ldots, N\}$. For each edge $e_{ij} \in E$, we denote the associated equivariant features consisting of tensors by $\mathbf{e}_{ij}$. During each messaging passing layer, the **LSE** module outputs the scalar weight coefficients A_{ij} as enhanced invariant edge feature and feed into the interaction module. Moreover, scalarization and tensorization as two essential blocks are used in the equivariant update module that fulfills the function of **FTE**. The permutation equivariance of a geometric graph is automatically guaranteed for any message passing architecture, we provide a complete proof of SE(3)-equivariance for LEFTNet in Appendix D.

SE(3) vs E(3) Equivariance. Besides explicitly fitting the $SE(3)$ invariant molecular geometry probability distribution, modeling the energy surface of a molecule system is also a crucial task for molecule property prediction. However, the Hamiltonian energy function E of a molecule is invariant under refection transformation: $\textbf{Energy}(\mathbf{X}) = \textbf{Energy}(R\mathbf{X})$, for arbitrary reflection transformation $R \in E(3)$. In summary, there exist two different inductive biases for modeling 3D data: **(1)** SE(3) equivariance, e.g. chirality could turn a therapeutic drug to a killing toxin; **(2)** E(3) equivariance, e.g. energy remains the same under reflections.

Since we implement $SE(3)$ equivariant frames in LEFTNet, our algorithm is naturally $SE(3)$ equivariant (and reflection anti-equivariant). However, our method is **flexible** to implement $E(3)$ equivariant tasks as well. For $E(3)$ equivariance, we can either replace our frames to $E(3)$ equivariant frames, or modify the scalarization block by taking the absolute value: $\mathbf{x} \to \tilde{x} := \underbrace{(\mathbf{x} \cdot e_1, \mathbf{x} \cdot e_2, \mathbf{x} \cdot e_3)}_{SE(3)} \to \underbrace{(\mathbf{x} \cdot e_1, |\mathbf{x} \cdot e_2|, \mathbf{x} \cdot e_3)}_{E(3)}$. Intuitively, since the second vector e_2 is a pseudo-vector, projections of any equivariant vectors along the e_2 direction are not $E(3)$ invariant until taking the absolute value.

Efficiency. To analyze the efficiency of LEFTNet, suppose 3D graph G has n vertices, and its average node degree is k. Our algorithm consists of three phases: 1. Building equivariant frames and performing local scalarization; 2. Equivariant message passing; 3. Updating node-wise tensor features through scalarization and tensorization. Let l be the number of layers, then the computational complexity for each of our three phases are: 1. $O(nk)$ for computing the frame and local (1-hop) 3D features; 2. $O(nkl)$ for 1-hop neighborhood message aggregation; 3. $O(nl)$ for node-wise tensorization and feature update.

6 Related Work

In light of the discussions in Section 3 and 4, we summarize two necessary ingredients for building expressive equivariant 3D GNNs: (1) local 3D substructure encodings (**LSE**), such that the local

Table 2: Mean Absolute Error for the molecular property prediction benchmark on QM9 dataset. (The best results are **bolded** and the second best are underlined.)

Task Units	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	G meV	H meV	R^2 bohr^3	U meV	U_0 meV	ZPVE meV
NMP	.092	69	43	38	.030	.040	19	17	.180	20	20	1.50
Cormorant	.085	61	34	38	.038	.026	20	21	.961	21	22	2.03
LieConv	.084	49	30	25	.032	.038	22	24	.800	19	19	2.28
TFN	.223	58	40	38	.064	.101	-	-	-	-	-	-
SE(3)-Tr.	.142	53	35	33	.051	.054	-	-	-	-	-	-
EGNN	.071	48	29	25	.029	.031	12	12	**.106**	12	11	1.55
SEGNN	.060	42	24	21	.023	.031	15	16	.660	13	15	1.62
ClofNet	.063	53	33	25	.040	.027	9	9	.610	9	8	**1.23**
EQGAT	.063	44	26	22	.014	.027	12	13	.257	13	13	1.50
Equiformer	.056	**33**	**17**	**16**	.014	.025	10	10	.227	11	10	1.32
LEFTNet (ours)	**.048**	40	24	18	**.012**	**.023**	**7**	**6**	.109	**7**	**6**	1.33
Schnet	.235	63	41	34	.033	.033	14	14	.073	19	14	1.70
DimeNet++	.044	33	25	20	.030	.023	8	7	.331	6	6	1.21
SphereNet	.046	**32**	**23**	**18**	.026	**.021**	8	6	.292	7	6	**1.12**
ClofNet	.053	49	33	25	.038	.026	9	8	.425	8	8	1.59
PaiNN	.045	46	28	20	.012	.024	7	6	**.066**	6	6	1.28
LEFTNet (ours)	.039	39	**23**	**18**	**.011**	.022	**6**	**5**	.094	**5**	**5**	1.19

message is aware of different local 3D structures; (2) frame transition encodings (**FTE**), such that the 3D GNN is aware of the equivariant coordinate transformation between different local patches.

We review the previous 3D GNNs following this framework and summarize the results in Table 1. For a fair comparison, we also list the computational complexity as it is often a trade-off of expressiveness (see the detailed analysis at the end of the next section). For **LSE**, SphereNet [14] and GemNet [33] (implicitly) encode the local 3D substructures by introducing a computation-intensive 2-hop edge-based update. For **FTE**, most 3D GNNs with equivariant vector update are able to express the local frame transitions (**FT**). While EGNN [13] is an exception, because it only updates the position vector (i.e. one channel), which is insufficient to express the whole **FT**. In other words, whether the update function ϕ of (8) is powerful also affects the **FT** encoding. Except for equivariant update methods, models that encode torsion angle information also partially express **FTE** as illustrated in Appendix C. However, there is a trade-off between the efficiency and expressiveness in terms of number of hops considered for message passing.

Different from our invariant realization of **LSE**, Batatia et al. [25] builds its framework by constructing complete equivariant polynomial basis with the help of spherical harmonics and tensor product, where the monomial variables depend on different nodes (bodies). On the other hand, we realize the function of **LSE** and **FTE** through the edgewise scalarization A_{ij} and the equivariant message passing (see Fig. 3).

Recently, Joshi et al. [21] propose a **geometric k-WL test** (GWL) to measure the expressiveness power of geometric GNN algorithms. On a high level, our tree isomorphism is equivalent to the 1-hop geometric isomorphism as proposed in GWL, and the fine-grained triangular isomorphism lies between the 1-hop and 2-hop geometric isomorphism as proposed in GWL. From the model design point of view, our realization of **LSE** is through local scalarization, whose expressiveness is guaranteed by the Kolmogorov representation theorem (see [34]) and the universal approximator property of MLP. Moreover, the key concepts of measuring the expreesive power in [21] are the body order and tensor order, which originate from classical inter-atomic potential theories and are of the equivariance nature. On the other hand, we discover the **FTE** as the 'missing' bridge connecting local invariant scalars and global geometric expressiveness, which (together with **LSE** on mutual 3D substructures) also reveals why the 1-hop scalarization implemented in ClofNet [23] is insufficient.

7 Experiments

We test the performance of LEFTNet on both scalar value (e.g. energy) and vector value (e.g. forces) prediction tasks. The scalar value prediction experiment is conducted on the QM9 dataset [35] which includes $134k$ small molecules with quantum property annotations; the vector value prediction experiment is conducted on the MD17 dataset [36] and the Revised MD17(rMD17) dataset [37] which includes the energies and forces of molecules. We compare our LEFTNet with a list of state-of-the-art equivariant (invariant) graph neural networks including SphereNet [14], PaiNN [31],

Equiformer [32], GemNet [33], etc [29, 38, 12, 39, 40, 13, 15, 41–47].The results on rMD17 and ablation studies are listed in Appendix E.

7.1 QM9 - Scalar-valued Property Prediction

The QM9 dataset is a widely used dataset for predicting molecular properties. However, existing models are trained on different data splits. Specifically, Cormorant [40], EGNN [13], etc., use 100k, 18k, and 13k molecules for training, validation, and testing, while DimeNet [38], SphereNet [14], etc., split the data into 110k, 10k, and 11k. For a fair comparison with all baseline methods, we conduct experiments using both data splits. Experimental results are listed in Table 2. For the first data split, LEFTNet is the best on 7 out of the 12 properties and improves previous SOTA results by 20% on average. In addition, LEFTNet is the second best on 4 out of the other 5 tasks. Consistently, LEFTNet is the best or second best on 10 out of the 12 properties for the second split. These experimental results on both splits validate the effectiveness of LEFTNet on scalar-valued property prediction tasks. The ablation study in Appendix E shows that both **LSE** and **FTE** contribute to the final performance.

7.2 MD17 - Vector-valued Property Prediction

We evaluate the ability of LEFTNet to predict forces on the MD17 dataset. Following existing studies [29, 38, 14], we train a separate model for each of the 8 molecules. Both training and validation sets contain 1000 samples, and the rest are used for testing. Note that all baseline methods are trained on a joint loss of energies and forces, but different methods use different weights of force over energy (WoFE). For example, SchNet [29] sets WoEF as 100, while GemNet [33] uses a weight of 1000. For a fair comparison with existing studies, we conduct experiments on two widely used weights of 100 and 1000 following Liu et al. [14]. The results are summarized in Table 3. We can observe that when WoFE is 100, LEFTNet outperforms all baseline methods on 7 of the 8 molecules and improves previous SOTA results by 16% on average. In addition, LEFTNet can outperform all baseline methods on 6 of the 8 molecules when WoFE is 1000. These experimental results on MD17 demonstrate the performance of LEFTNet on vector-valued property prediction tasks. The ablation study in Appendix E also demonstrates that both **LSE** and **FTE** are important to the final results.

8 Limitation and Future Work

In this paper, we seek a general recipe for building 3D geometric graph deep learning algorithms. Considering common prior of 2D graphs, such as permutation symmetry, has been incorporated in off-the-shelf graph neural networks, we mainly focus on the $E(3)$ and $SE(3)$ symmetry specific to 3D geometric graphs. Despite our framework being general for modeling geometric objects, we only conducted experiments on commonly used molecular datasets. It's worth exploring datasets in other domains in the future.

To elucidate the future design space of equivariant GNNs, we propose two directions that are worth exploring. Firstly, our current algorithms consider fixed equivariant frames for performing aggregation and node updates. Inspired by the high body-order ACE approach [48] (for modeling atom-centered potentials), it is worth investigating in the future if equivariant frames that relate to many body (e.g., the PCA frame in [49]) can boost the performance of our algorithm. For example, to

Table 3: Mean Absolute Error for per-atom forces prediction (kcal/mol Å) on MD17 dataset. Baseline results are taken from the original papers (with unit conversions if needed). All models are trained on energies and forces, and WoFE is the weight of force over energy in loss functions. The best results are **bolded**.

	WoFE=100						WoFE=1000			Others	
Molecule	sGDML	SchNet	DimeNet	SphereNet	SpookyNet	LEFTNet	SphereNet	GemNet	LEFTNet	PaiNN	NewtonNet
Aspirin	0.68	1.35	0.499	0.430	0.258	**0.210**	0.209	0.217	**0.196**	0.371	0.348
Benzene	0.20	0.31	0.187	0.178	–	**0.145**	0.147	0.145	**0.142**	–	–
Ethanol	0.33	0.39	0.230	0.208	**0.094**	0.118	0.091	**0.086**	0.099	0.230	0.264
Malonaldehyde	0.41	0.66	0.383	0.340	0.167	**0.159**	0.172	0.155	**0.142**	0.319	0.323
Naphthalene	0.11	0.58	0.215	0.178	0.089	**0.063**	0.048	0.051	**0.044**	0.083	0.084
Salicylic acid	0.28	0.85	0.374	0.360	0.180	**0.141**	**0.113**	0.125	0.117	0.209	0.197
Toluene	0.14	0.57	0.210	0.155	0.087	**0.070**	0.054	0.060	**0.049**	0.102	0.088
Uracil	0.24	0.56	0.301	0.267	0.119	**0.117**	0.106	0.097	**0.085**	0.140	0.149

build the A-basis proposed in Puny et al. [49], we can replace our message aggregation Eq. 8 from summation to tensor product, which is also a valid pooling operation. Another direction is to explore geometric mesh graphs on manifolds M, where the local frame is defined on the tangent space of each point: $\mathcal{F}(x) \in T_x M$. Since our scalarization technique (crucial for realizing **LSE** in LEFT-Net) originates from differential geometry on frame bundles [22], it is reasonable to expect that our framework also works for manifold data [50, 51].

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[3] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[4] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71:361–390, 2020.

[5] Lars Holdijk, Yuanqi Du, Priyank Jaini, Ferry Hooft, Bernd Ensing, and Max Welling. Path integral stochastic optimal control for sampling transition paths. In *ICML 2022 2nd AI for Science Workshop*.

[6] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017.

[7] Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12):1023–1032, 2021.

[8] Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, Qi Qi, Zhuoning Yuan, Tianbao Yang, et al. Advanced graph and sequence neural networks for molecular property prediction and drug discovery. *Bioinformatics*, 38(9):2579–2586, 2022.

[9] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

[10] Yuanqi Du, Tianfan Fu, Jimeng Sun, and Shengchao Liu. Molgensurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500*, 2022.

[11] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[12] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[13] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[14] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3D molecular graphs. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=givsRXsOt9r`.

[15] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[16] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[17] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*.

[19] Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any gnn with local structure awareness. In *International Conference on Learning Representations*.

[20] Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. Understanding and extending subgraph gnns by rethinking their symmetries. *Advances in Neural Information Processing Systems*, 35:31376–31390, 2022.

[21] Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. In *The First Learning on Graphs Conference*, 2022.

[22] Elton P Hsu. *Stochastic analysis on manifolds*. Number 38. American Mathematical Soc., 2002.

[23] Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pages 5583–5608. PMLR, 2022.

[24] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3D molecular graphs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=mCzMqeWSFJ`.

[25] Ilyes Batatia, David Peter Kovacs, Gregor NC Simm, Christoph Ortner, and Gabor Csanyi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*.

[26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[27] Christopher Morris, Yaron Lipman, Haggai Maron, Bastian Rieck, Nils M Kriege, Martin Grohe, Matthias Fey, and Karsten Borgwardt. Weisfeiler and leman go machine learning: The story so far. *arXiv preprint arXiv:2112.09992*, 2021.

[28] Asiri Wijesinghe and Qing Wang. A new perspective on" how graph neural networks go beyond weisfeiler-lehman?". In *International Conference on Learning Representations*, 2021.

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

[30] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

[31] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[32] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

[33] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. GemNet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

[34] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[35] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[36] Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

[37] Anders S Christensen and O Anatole Von Lilienfeld. On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1(4): 045018, 2020.

[38] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[39] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[40] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

[41] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.

[42] Tuan Le, Frank Noé, and Djork-Arné Clevert. Equivariant graph attention networks for molecular property prediction. *arXiv preprint arXiv:2202.09891*, 2022.

[43] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.

[44] Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice EA Allen, Daniel J Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *Journal of chemical theory and computation*, 17(12):7696–7711, 2021.

[45] Felix A Faber, Anders S Christensen, Bing Huang, and O Anatole Von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of chemical physics*, 148(24):241717, 2018.

[46] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

[47] Xiang Gao, Farhad Ramezanghorbani, Olexandr Isayev, Justin S Smith, and Adrian E Roitberg. Torchani: a free and open source pytorch-based deep learning implementation of the ani neural network potentials. *Journal of chemical information and modeling*, 60(7):3408–3415, 2020.

[48] Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor NC Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of e (3)-equivariant atom-centered interatomic potentials. *arXiv preprint arXiv:2205.06643*, 2022.

[49] Omri Puny, Matan Atzmon, Edward J Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2021.

[50] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *Advances in Neural Information Processing Systems*, 34:27331–27343, 2021.

[51] Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. *arXiv preprint arXiv:2203.06442*, 2022.

[52] Nadav Dym and Haggai Maron. On the universality of rotation equivariant point cloud networks. *arXiv preprint arXiv:2010.02449*, 2020.

[53] Sheng Gong, Tian Xie, Yang Shao-Horn, Rafael Gomez-Bombarelli, and Jeffrey C Grossman. Examining graph neural networks for crystal structures: limitations and opportunities for capturing periodicity. *arXiv preprint arXiv:2208.05039*, 2022.

[54] Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pages 5361–5370. PMLR, 2020.

[55] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

Appendix

A Supplementary Background

We briefly review the concept of (contravariant) tensor fields and their associated equivariant group representations.

A s order (contravariant) tensor $\mathbf{T}$ on a vector space $\mathbf{V}$ is a multilinear map:

$$T : \underbrace{\mathbf{V}^* \times \cdots \times \mathbf{V}^*}_{s} \to \mathbf{R}^1,$$

where $\mathbf{V}^*$ denotes the dual space of $\mathbf{V}$. In fact, there is a canonical 'multiplication' operation between two tensors. Define the **tensor product** $\mathbf{S} \otimes \mathbf{T}$ of two tensors $\mathbf{S}$ and $\mathbf{T}$ to be a tensor of order $r + s$:

$$\mathbf{S} \otimes \mathbf{T}(v^1, \ldots, v^{r+s}) = \mathbf{S}(v^1, \ldots, v^r)\mathbf{T}(v^{r+1}, \ldots, v^{r+s}). \tag{11}$$

where $v^i \in \mathbf{V}^*$.

From now on, we assume $V = V^* = \mathbf{R}^3$. Note that when $s = 1$, $\mathbf{T}$ is exactly an equivariant vector. In practice, the tensor data in $\mathbf{R}^3$ is usually given by its coefficients under a Cartesian coordinate system. Take a second-order tensor as an example, assume we are given an orthonormal frame (basis) $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ and its dual frame $(\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3)$, then the nine coefficients of T are given by

$$T_{ij} = \mathbf{T}(\mathbf{e}^i, \mathbf{e}^j), \;\; 1 \leq i, j \leq 3.$$

In other words, we say the collection $\{T_{ij}\}_{1 \leq i,j \leq 3}$ is a faithful representation of $\mathbf{T}$ in a fixed coordinate system:

$$\mathbf{T} = \sum_{i,j} T_{ij} \mathbf{e}_i \otimes \mathbf{e}_j. \tag{12}$$

Once defined a tensor on $\mathbf{R}^3$, it's easy to extend it to a continuous manifold or a discrete graph. A **tensor field** of order s on a 3D graph $G = (V, E)$ is a tensor-valued function f which assigns to each 3D node $\mathbf{x}_i$ an order s tensor, denoted by $f(\mathbf{x}_i)$.

SE(3) Tensor Representations. Let V be a vector space, then the group SE(3) is said to act on V if there is a mapping $\phi : SE(3) \times V \to V$ satisfying the following two conditions:

1. if $e \in SE(3)$ is the identity element, then
$$\phi(e, x) = x \quad \text{for } \forall x \in V.$$
2. if $g_1, g_2 \in SE(3)$, then
$$\phi(g_1, \phi(g_2, x)) = \phi(g_1 g_2, x) \quad \text{for } \forall x \in V.$$

If we further require $\phi(g, \cdot)$ is a linear map for all $g \in SE(3)$, then ϕ becomes a group representation of $SE(3)$. From now on, we only consider the rotation subgroup SO(3) and its group representations. When $V = \mathbf{R}^3$, there is a natural representation of $SO(3)$ by rotating vectors in $\mathbf{R}^3$. In this way, an element $g \in SO(3)$ is identified with a Rotation matrix, denoted by $\{g_i^j\}_{1 \leq i,j \leq 3}$.

From the tensor definition (11), this natural representation on $\mathbf{R}^3$ induces a tensor representation on T. Still take $\mathbf{T} = \{T_{ij}\}_{1 \leq i,j \leq 3}$ as an example, we have

$$T_{kl} = \sum_i \sum_j g_k^i g_l^j T_{ij}, \quad 1 \leq k, l \leq 3, \tag{13}$$

for $\forall g \in SO(3)$. It's easy to check that (13) is indeed a SO(3) representation on the vector space spanned by second-order tensors.

Relation with Spherical Harmonics. For the $SO(3)$ group, all representations (including the tensor representations) can be decomposed as a direct sum of irreducible representations. For each type of irreducible representations, there is a subset of spherical harmonics formulating a basis for this specific representation. However, in terms of representing equivariant geometric quantities, the theorem in [52] claims that tensor representations and irreducible representations are equally powerful: They all form a complete basis in the space of continuous $E(3)$ equivariant functions.

Algorithm 1 Invariant Design for LEFTNET.

Input: Complete 3D gragh with equivariant positions $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n\times 3}$, invariant node features $h_i \in \mathbb{R}^d$, invariant relative distances $d_{ij} \in \mathbb{R}^1$.
Centralize the positions: $\mathbf{X} \leftarrow \mathbf{X} - \text{CoM}(\mathbf{X})$.
for $(i = 1; i < n; i++)$ **do**
  for $(j = 1; j < k; j++)$ **do**
    Compute edge-wise equivariant frames $\mathcal{F}_{ij}$ via Eq. 17:

$$\mathcal{F}_{ij} = \mathbf{EquiFrame}(\mathbf{x}_i, \mathbf{x}_j).$$

    Get the mutual 3D structure $\mathbf{S}_{i-j}$, perform local scalarization:

$$t_{ij} = \mathbf{Scalarize}(\mathbf{S}_{i-j})$$

    Calculate the SE(3)-invariant structural coefficients: $A_{ij} = g(t_{ij}, d_{ij})$
    Perform invariant message passing:

$$m_{ij} = \phi_m(h_i, A_{ij} \odot h_j, d_{ij})$$

  end for
  Update invariant node features:

$$\mathbf{h}_i = \phi_h(\mathbf{h}_i, \sum_{j\in\mathcal{N}(i)} m_{ij})$$

end for
Output: AvgPooling $(h_1, \ldots, h_n)$**.**

B Related Proofs and Discussions of Section 3

In section 3, we proposed a novel hierarchy of local geometric isomorphisms that further motivates the design of incorporating the mutual 3D substructure's information into equivariant GNNs. Different from our fine-grained local characterization, a cocurrent work GWL [21]) proposes to measure geometric isomorphism from local to global by the k-hop partition.

From another point of view, we essentially demonstrated that encoding mutual 3D substructures expands the capacity of the transformation function class with respect to an equivariant GNN. [24] put forward the **Completeness** concept for characterizing these transformation functions. However, it mainly concentrates on testing whether a function can discriminate global geometric isomorphism (in the sense of Eq. 5).

Discussion on the Completeness Concept. Following our terminology in the preliminary section, **completeness** of a transformation f can be translated into claiming that f is invariant among 3D graphs if and only if they are **globally** isomorphic (see definition (5)). Therefore, it's easy to refine the notion of completeness that adapts to our local version by replacing the global isomorphism to local isomorphism:

$$f(\mathbf{X}) = f(\mathbf{Y}),$$

if and only if $\mathbf{X}$ and $\mathbf{Y}$ are local {-tree, -triangular, -subgraph } isomorphic. Then, in terms of function class capacities, the following relation holds:

Global complete $\subset$ **Subgraph complete** $\subset$ **Triangular complete** $\subset$ **Tree complete**.

Note that our equivalent description of complete transformation reveals the fact that the completeness concept in [24] is defined from the global 3D isomorphism point of view. Therefore, we shall claim that the above series of completeness notions belong to the **structure** completeness. Indeed, the theory developed in section 3 indicates that a GNN which can express **structure** complete functions may not be sufficient in expressing general tensor potential functions on a 3D graph.

On the other hand, a non-negligible proportion of 3D graph tasks may not be sensitive to the global 3D non-isomorphism. For example, some chemical properties (formulated as a function defined on molecular graphs) are characterized by local substructures [53]. In these scenarios, we are looking

for a geometric transformation f that is global non-complete, but (-tree, -triangular, -subgraph) local complete.

Proof of Theorem 3.1.

Proof. The first part of the theorem is proved by providing an explicit example. From the first 3D shapes of figure 1, the difference of the two triangular non-isomorphism shapes is indicated by an invariant function:

$$d(\mathbf{x}_p, \mathbf{x}_m) = \|\mathbf{x}_p - \mathbf{x}_m\|^2 .$$

Note that this function cannot be expressed by tree-level features, since there is no edge connecting $\mathbf{x}_p$ and $\mathbf{x}_m$. However, since the position vectors $\mathbf{x}_p$ and $\mathbf{x}_m$ are included in the mutual 3D structure (corresponds to edge e_{iq}), then they are ready to be scalarized by a local equivariant frame. Then quoting the universal approximation theorem from [23], there exists a corresponding invariant encoder ϕ that can approximate the function $d(\mathbf{x}_k, \mathbf{x}_l)$. Since this function produces different output values for the two tree isometric but triangular non-isometric 3D shapes, we know that ϕ is able to distinguish 3D shapes beyond tree isomorphism.

To prove the second part and build up the injectivity condition, we first introduce the multi-set notation $\{\!\{\cdot\}\!\}$, following [GIN]. A basic equivariant GNN based on our enhanced framework contains at least two steps: 1. Message passing, which is defined by (6); 2. Node-wise update:

$$h_i^{t+1} = \mathbf{MLP}(m_i^t, h_i^t).$$

For simplicity, we denote the composition of the two steps by Ψ. Then the additional injectivity condition is stated as follows:

$$\Psi(\{\!\{h_i^t, A_{ji}h_i^t, , h_j^t | j \in \mathcal{N}_i\}\!\}, \{\!\{A_{ij}h_j^t | j \in \mathcal{N}_i\}\!\}) \tag{14}$$

is injective, for each layer t and each node i. Note that this condition is realizable by adding weighted residue terms similar to [26]. Then, from the above condition, it's obvious that two non-identical collections of $\{\{A_{ij}\}_{e_{ij}\in E}\}$ would yield two different feature vectors. Moreover, from the first part, there exist at least two distinct local 3D subgraphs with isometric local tree structure, such that the corresponding geometric weights $\{\{A_{ij}\}_{e_{ij}\in E}\}$ that come out of the encoder ϕ are different. □

C Related Proofs and Discussions of Section 4

Torsion Angle is Secretly Hidden in FT

Recall the edge-wise (signed) torsion angle τ_{ij} involves the 1-hop atom pairs i and j and two 2-hop atoms k and l, then τ_{ij} is defined to be the dihedral angle between plane $k-i-j$ and plane $l-j-i$. Although exhausting all torsion angles requires $O(k^2)$ complexity, [24] reduces the computation to $O(k)$ order by selecting a canonical 2-hop atom k and l, which is enough for detecting the relative orientations between atoms (insufficient for general tasks like many body interactions).

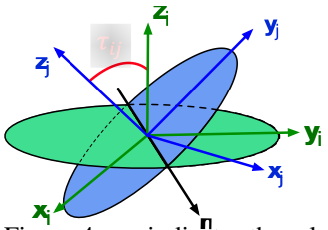


Figure 4: τ_{ij} indicates the relative rotation of two frames along the z-axis.

Now we show how τ_{ij} naturally appears as one of the derivatives from frame transition functions. For node i, define the equivariant frame $\mathcal{F}_i$ by

$$(\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i) = (\mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_k, \mathbf{e}_1^i \times \mathbf{e}_2^i).$$

$\mathcal{F}_i$ is normalized through the Gram-Schmidt algorithm. For node j, $\mathcal{F}_j$ is defined similarly by

$$(\mathbf{e}_1^j, \mathbf{e}_2^j, \mathbf{e}_3^j) = (\mathbf{x}_j - \mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_l, \mathbf{e}_1^j \times \mathbf{e}_2^j).$$

Then following the transition formula (7),

$$R_{ij} = (\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i) \cdot (\mathbf{e}_1^j, \mathbf{e}_2^j, \mathbf{e}_3^j)^T.$$

Note that for an orthonormal matrix, its inverse equals its transpose. Then, by the standard definition of a dihedral angle, we have

$$\tau_{ij} = \mathbf{e}_3^i \cdot \mathbf{e}_3^j \equiv R_{ij}(3,3).$$

Algorithm 2 Equivariant Design for LEFTNET.

Input: Complete 3D gragh with equivariant positions $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n\times 3}$, invariant node features $h_i \in \mathbb{R}^d$, invariant relative distances $d_{ij} \in \mathbb{R}^1$, **equivariant** edge features $\mathbf{e}_{ij} \in \mathbb{R}^c$.
Centralize the positions: $\mathbf{X} \leftarrow \mathbf{X} - \text{CoM}(\mathbf{X})$.
for $(i = 1; i < n; i++)$ **do**
  Compute node-wise equivariant frames $\mathcal{F}_i$ via Eq. 10 .
  for $(j = 1; j < k; j++)$ **do**
    Compute edge-wise equivariant frames $\mathcal{F}_{ij}$ via Eq. 17:

$$\mathcal{F}_{ij} = \mathbf{EquiFrame}(\mathbf{x}_i, \mathbf{x}_j)$$

    Get the mutual 3D structure $\mathbf{S}_{i-j}$, perform local scalarization through $\mathcal{F}_{ij}$:

$$t_{ij} = \{\mathbf{Scalarize}(\mathbf{S}_{i-j}, \mathcal{F}_{ij})\}$$

    Calculate the SE(3)-invariant structural coefficients: $A_{ij} = g(t_{ij}, d_{ij})$
    Perform equivariant message passing as in Eq. 6:

$$\mathbf{m}_{ij} = \phi_m^1(h_i, A_{ij} \odot h_j, d_{ij}) + \phi_m^2(h_i, A_{ij} \odot h_j, d_{ij}) \cdot \mathbf{e}_{ij} + \phi_m^3(h_i, A_{ij} \odot h_j, d_{ij}) \cdot \mathcal{F}_{ij}$$

  end for
  Equivariant message aggregation: $\mathbf{m}_i = \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}$;
  Transform equivariant node features through $\mathcal{F}_i$:

$$t_i = \mathbf{Scalarize}(\mathbf{m}_i, \mathcal{F}_i)$$

  Update invariant node features:

$$h_i = \phi_h(h_i, t_i)$$

  Equivariant Output: Perform tensorization through $\mathcal{F}_i$:

$$\mathbf{h}_i = \mathbf{Tensorize}(h_i, \mathcal{F}_i).$$

end for

In conclusion, τ_{ij} is just one component of the transition matrix. However, to fully determine R_{ij}, we still need another two angles (since a transition matrix is uniquely determined by three Euler angles).

Proof of Theorem 4.1.

Proof. This theorem is proved in two steps:

1. The first step characterizes all scalars determined by (isolated) local clusters $\mathbf{B}$ and $\mathbf{C}$ through equivariant frames and local scalarization;
2. The second step constructs a specific invariant function $f_a(\mathbf{B}, \mathbf{C})$ that cannot be expressed by the local scalarization.

Let G denote a 3D point cloud. Then, as it has been proved in Du et al. [23], once equipped with an equivariant frame $\mathbf{F}_G$, all equivariant features of G can be transformed to scalar features through scalarization without information loss. Following the convention in the main text, let $\tilde{G}$ be the output of performing scalarization on the original features G, then for any invariant function $f(G)$, there exists a corresponding function $\tilde{f}$ such that

$$f(G) = \bar{f}(\tilde{G}). \tag{15}$$

Since S_G is the collection of all invariant scalars produced by G, (15) implies that S_G is generated by $\tilde{G}$, which is only a finite subset of S_G. To apply the above insight to our current theorem, note that we have two different 3D clouds $\mathbf{B}$ and $\mathbf{C}$. Therefore, we need to build two local equivariant frames $\mathcal{F}_\mathbf{B} = (e_1^\mathbf{B}, e_2^\mathbf{B}, e_3^\mathbf{B})$ and $\mathcal{F}_\mathbf{C} = (e_1^\mathbf{C}, e_2^\mathbf{C}, e_3^\mathbf{C})$. The crucial point is the frame $\mathcal{F}_\mathbf{B}$ itself doesn't depend on

C, and the scalarization through $\mathcal{F}_{\mathbf{B}}$ is only performed on the **local** point cloud **B**. Operations like scalarizing equivariant information of **C** through $\mathcal{F}_{\mathbf{B}}$ would break the assumptions of the theorem.

Now we are ready to construct an explicit interaction potential $f_a(\mathbf{B}, \mathbf{C})$:

$$f_a(\mathbf{B}, \mathbf{C}) := e_1^{\mathbf{B}} \cdot e_1^{\mathbf{C}}.$$

Since f_a is an inner product of two equivariant vectors, it's automatically an invariant function. Then we need to check whether $f_a(\mathbf{B}, \mathbf{C})$ is a function of $f_a(S_{\mathbf{B}}, S_{\mathbf{C}})$. Note that the equivariant building block of f_a that relates to **B** is exactly $e_1^{\mathbf{B}}$. Then, following the above local scalarization principle, we scalarize $e_1^{\mathbf{B}}$ through $\mathcal{F}_{\mathbf{B}}$ and get:

$$e_1^{\mathbf{B}} \to \tilde{e}_1^{C} = (1, 0, 0).$$

Similarly, $e_1^{\mathbf{C}}$ is also transformed to a constant scalar tuple $\tilde{e}_1^{\mathbf{C}} = (1, 0, 0)$ through $\mathcal{F}_{\mathbf{C}}$. As constant inputs generate constant outputs, we conclude that the deduced local scalars can only approximate constant functions. However, since the local frames are changing as we vary the 3D structure of **B** and **C**, it's obvious that $e_1^{\mathbf{B}} \cdot e_1^{\mathbf{C}}$ is not a constant function of $(\mathbf{B}, \mathbf{C})$. Therefore, we finish the proof by contradiction. □

Realizing FT by Equivariant Messages: From the **FT** definition 7, each element of the 3×3 matrix R_{ij} is calculated by

$$R_{ij}(k, l) = \mathbf{e}_k^i \cdot \mathbf{e}_l^j. \tag{16}$$

Now we show how to reproduce $R_{ij}(k, l)$ through equivariant messages. Let the equivariant message $\mathbf{m}_i$ be the following:

$$\mathbf{m}_i := (\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i) \cdot \left[\begin{array}{ccc:ccc:ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

It's easy to check that $\mathbf{m}_i \in \mathbf{R}^{3\times 9}$ consists of 9 equivariant vectors (**multi-channels**). For atom j, $\mathbf{m}_j$ is defined symmetrically. For each node, we also store the scalar messages, e.g., $\left\|\mathbf{e}_k^i\right\|$ for $1 \leq k \leq 3$. Flattening the whole matrix R_{ij} into a $\mathbf{R}^{1\times 9}$ array, then R_{ij} is obtained by simple summation and taking the vector norm:

$$\|\mathbf{m}_i + \mathbf{m}_j\| = \left\{\left\|\mathbf{e}_k^i + \mathbf{e}_l^j\right\|\right\}_{1\leq k,l\leq 3},$$

where the norm is taken for each column of $\mathbf{m}_i + \mathbf{m}_j$, such that $\|\mathbf{m}_i + \mathbf{m}_j\| \in \mathbf{R}^{1\times 9}$. Then,

$$R_{ij} = \frac{1}{2}\left[\left\|\mathbf{e}_k^i + \mathbf{e}_l^j\right\|^2 - \left\|\mathbf{e}_k^i\right\|^2 - \left\|\mathbf{e}_l^j\right\|^2\right].$$

Our illustration also demonstrates the importance of keeping multi-channel tensor messages.

Relation with Previous Equivariant Update Methods. Following the efficiency principle established in section 4, we don't encode the data of the transition matrices explicitly. Instead, we implement tensor messages to fill in the expressiveness gap. Among the tremendously different designs of equivariant graph neural networks, Schütt et al. [31] is closely related to our equivariant updating method. By the above argument, the inner product operation for node i (see (9) of Schütt et al. [31])

$$< \mathbf{U}\mathbf{v}_i, \mathbf{V}\mathbf{v}_i >$$

can also be reinterpreted as a realization of the (aggregated) frame transition matrix (7).

Moreover, since the equivariant vectors $\mathbf{U}\mathbf{v}_i$ and $\mathbf{V}\mathbf{v}_i$ are both aggregated vector features that belong to the same node i and the inner product operation between them is performed in the node-wise updating phase, Schütt et al. [31] actually avoids the 2-hop $O(k^2)$ complexity of computing R_{xy} for all neighborhood node pairs $(\mathbf{x}, \mathbf{y})$ (while able to express the torsion angle implicitly). For our algorithm, we utilize the scalarization and tensorization in the node-wise updating phase. By the universal approximation theorem 5.1, our method can approximate any inner product operations.

D Related Proofs and Discussions of Section 5

Equivariant Frames and Higher Order Scalarization and Tensorization. Given an edge e_{ij} with two atom's positions $(\mathbf{x}_i, \mathbf{x}_j)$, our edge-wise $SE(3)$ equivariant frames $\mathcal{F}_{ij}$ are defined by:

$$(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = (\frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}, \frac{\mathbf{x}_i \times \mathbf{x}_j}{\|\mathbf{x}_i \times \mathbf{x}_j\|}, \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \times \frac{\mathbf{x}_i \times \mathbf{x}_j}{\|\mathbf{x}_i \times \mathbf{x}_j\|}). \tag{17}$$

To make the frame translation invariant, we follow previous works [54, 55] by limiting the whole 3D conformers' space to a linear subspace where the center of mass (CoM) of the system (either the whole system or the sub-cluster where i and j belong to) is zero. On the other hand, building an **E(3)** frame requires an additional atom's position $\mathbf{x}_k$, which can be selected by K-Nearest Neighbor algorithm. Then if $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ spans the 3D space, we obtain an $E(3)$ equivariant frame by performing the gram-schmidt orthogonalization.

Once we have an equivariant frame, every vector is a linear combination of the three orthogonal vectors in the frame. Moreover, the unique combination coefficients are exactly the 'scalarized' coordinates in (1). A similar procedure also applies to higher order tensors. Indeed, the vector frame $\mathcal{F}^1$ extends to a tensor frame $\mathcal{F}^r$ of arbitrary order $r > 1$:

$$\mathcal{F}^r := \{\mathbf{e}_{1_1} \otimes \cdots \otimes \mathbf{e}_{1_r}\}_{1 \le i_1, \ldots, i_r \le 3}. \tag{18}$$

Since the orthonormal frame $\mathcal{F}^r$ is complete in the sense that it spans the whole tensor space of order r, every r-th order tensor admits a unique decomposition:

$$\mathbf{T} = \sum_{1 \le i_1, \ldots, i_r \le 3} T^{i_1, \ldots, i_r} \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_r}. \tag{19}$$

It's easy to prove that the collection $\{T^{i_1,\ldots,i_r}\}_{1 \le i_1, \ldots, i_r \le 3}$ consists of invariant scalars. We call the process from $\mathbf{T}$ to $\{T^{i_1,\ldots,i_r}\}_{1 \le i_1, \ldots, i_r \le 3}$ **scalaraization**.

Tensorization is the inverse of scalarization, in the sense that it sends scalars $\{T^{i_1,\ldots,i_r}\}_{1 \le i_1, \ldots, i_r \le 3}$ to tensor $\mathbf{T}$. Under the same frames we use during scalarization, the following diagram demonstrates the pipeline of producing L second-order tensors out of $\{T_j^{i_1 i_2}\}_{1 \le i_1, i_2 \le 3}$:

$$\{\mathbf{T}_1, \ldots, \mathbf{T}_L\} = \underbrace{\left\{ \begin{bmatrix} T_1^{11}, & T_1^{12}, & T_1^{13} \\ T_1^{21}, & T_1^{22}, & T_1^{23} \\ T_1^{31}, & T_1^{32}, & T_1^{33} \end{bmatrix}, \ldots, \begin{bmatrix} T_L^{11}, & T_L^{12}, & T_L^{13} \\ T_L^{21}, & T_L^{22}, & T_L^{23} \\ T_L^{31}, & T_L^{32}, & T_L^{33} \end{bmatrix} \right\}}_{L \text{ channels}} \odot \begin{bmatrix} \mathbf{e}_1 \otimes \mathbf{e}_1, & \mathbf{e}_1 \otimes \mathbf{e}_2, & \mathbf{e}_1 \otimes \mathbf{e}_3 \\ \mathbf{e}_2 \otimes \mathbf{e}_1, & \mathbf{e}_2 \otimes \mathbf{e}_2, & \mathbf{e}_2 \otimes \mathbf{e}_3 \\ \mathbf{e}_3 \otimes \mathbf{e}_1, & \mathbf{e}_3 \otimes \mathbf{e}_2, & \mathbf{e}_3 \otimes \mathbf{e}_3 \end{bmatrix}, \tag{20}$$

where $\odot$ denotes the element-wise product.

Proof of Theorem 5.1

Proof. The proof is based on the fact that **Scalarization** and **Tensorization** are invertible (see Appendix A.5 of Du et al. [23]). In other words, we have the following commutative diagram: $\begin{array}{ccc} \mathbf{T}^{l-1} & \xrightarrow{\rho} & \mathbf{T}^{l} \\ \downarrow{\scriptstyle \textbf{Scalarize}} & & \uparrow{\scriptstyle \textbf{Tensorize}} \\ \tilde{T}^{l-1} & \xrightarrow{\textbf{MLP}} & \tilde{T}^{l-1} \end{array}$. Therefore, for each mapping ρ, we can always find a corresponding'scalarized' mapping $\tilde{\rho}$:

$$\tilde{\rho} := \textbf{Tensorize} \circ \rho \circ \textbf{Scalarize}.$$

Now we have turned from expressing equivariant ρ to the invariant $\tilde{\rho}$. Note that **MLP** is a universal approximator of invariant functions, therefore we can always find a **MLP** to express $\tilde{\rho}$. By reserving the arrows, we finish the proof. □

Proof of Equivariance for LEFTNet LEFTNet consists of multiple layers of **LSE** and **FTE**. **LSE** is realized by scalarization, and **FTE** is realized by scalarization and tensorization. Since the invariance of scalarization and the equivariance of tensorization have been proved, we finish the proof.

E Additional Experimental Results

Ablation Study. As discussed in Section 5, there are two main modules in LEFTNet, namely **LSE** and **FTE**. We conduct experiments on QM9 and MD17 to show the importance of each component. Experimental results are summarized in Table 4 and Table 5. The results show that using **LSE** can outperform the model without both **LSE** and **FTE** on all tasks. Adding **FTE** can further improve the performance. The results demonstrate the importance of **LSE** and **FTE** modules.

Table 4: Ablation study on QM9 dataset. The evaluation metric is MAE for each property. The **best** performances are bolded and the second best are underlined.

Task Units	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	G meV	H meV	R^2 bohr^3	U meV	U_0 meV	ZPVE meV
LEFTNet (w/o **LSE** and **FTE**)	.053	49	33	25	.038	.026	9	8	.425	8	8	1.59
LEFTNet (**LSE** only)	.043	49	31	23	.031	.025	8	7	.156	8	7	1.34
LEFTNet (**LSE** + vector **FTE**)	.039	39	23	18	**.011**	**.022**	**6**	**5**	**.094**	**5**	**5**	**1.19**
LEFTNet (**LSE** + tensor **FTE**)	**.038**	**38**	**22**	**17**	**.011**	**.022**	7	6	.096	**5**	6	1.20

Table 5: Abalation Study on MD17 dataset. The evaluation metric is MAE for per-atom forces prediction (kcal/mol Å). The **best** performances are bolded and the second best are underlined.

Molecule	LEFTNet (w/o **LSE** and **FTE**)	LEFTNet (**LSE** only)	LEFTNet (**LSE** + vector **FTE**)	LEFTNet (**LSE** + tensor **FTE**)
Aspirin	1.083	0.451	0.300	**0.210**
Benzene	0.425	0.185	**0.145**	0.176
Ethanol	0.341	0.149	0.138	**0.118**
Malonaldehyde	0.594	0.276	0.209	**0.159**
Naphthalene	0.658	0.175	0.073	**0.063**
Salicylic acid	0.828	0.313	0.167	**0.141**
Toluene	0.625	0.166	0.084	**0.070**
Uracil	0.581	0.206	**0.116**	0.117

Results on rMD17. Following [25], we conduct experiments on rMD17 to compare with recent studies. Results show that our LEFTNet can achieve comparable performance to state-of-the-art methods such as MACE and NequIP, while outperforming other baseline methods like GemNet and PaiNN.

Table 6: Mean Absolute Error for energy(meV) per-atom forces prediction (meV Å) on rMD17 dataset. Baseline results are taken from Batatia et al. [25]. The best results are **bolded**.

		LEFTNet	MACE	Allegro	BOTNet	NequIP	GemNet (T/Q)	ACE	FCHL	GAP	ANI	PaiNN
Aspirin	E	**2.1**	2.2	2.3	2.3	2.3	-	6.1	6.2	17.7	16.6	6.9
	F	**6.4**	6.6	7.3	8.5	8.2	9.5	17.9	20.9	44.9	40.6	16.1
Azobenzene	E	**0.7**	1.2	1.2	**0.7**	**0.7**	-	3.6	2.8	8.5	15.9	-
	F	3.3	3.0	**2.6**	3.3	2.9	-	10.9	10.8	24.5	35.4	-
Benzene	E	0.05	0.4	0.3	**0.03**	0.04	-	0.04	0.35	0.75	3.3	-
	F	0.3	0.3	**0.2**	0.3	0.3	0.5	0.5	2.6	6	10	-
Ethanol	E	**0.4**	**0.4**	**0.4**	**0.4**	**0.4**	-	1.2	0.9	3.5	2.5	2.7
	F	3.6	**2.1**	**2.1**	3.2	2.8	3.6	7.3	6.2	18.1	13.4	10
Malonaldehyde	E	0.8	0.8	**0.6**	0.8	0.8	-	1.7	1.5	4.8	4.6	3.9
	F	5.4	4.1	**3.6**	5.8	5.1	6.6	11.1	10.3	26.4	24.5	13.8
Naphthalene	E	0.8	0.5	**0.2**	**0.2**	0.9	-	0.9	1.2	3.8	11.3	5.1
	F	1.9	1.6	**0.9**	1.8	1.3	1.9	5.1	6.5	16.5	29.2	3.6
Paracetamol	E	**1.3**	**1.3**	1.5	**1.3**	1.4	-	4	2.9	8.5	11.5	-
	F	**4.7**	4.8	4.9	5.8	5.9	-	12.7	12.3	28.9	30.4	-
Salicylic acid	E	0.9	0.9	0.9	0.8	**0.7**	-	1.8	1.8	5.6	9.2	4.9
	F	4.1	3.1	**2.9**	4.3	4	5.3	9.3	9.5	24.7	29.7	9.1
Toluene	E	**0.3**	0.5	0.4	**0.3**	**0.3**	-	1.1	1.7	4	7.7	4.2
	F	2.2	**1.5**	1.8	1.9	1.6	2.2	6.5	8.8	17.8	24.3	4.4
Uracil	E	**0.4**	0.5	0.6	**0.4**	**0.4**	-	1.1	0.6	3	5.1	4.5
	F	2.8	2.1	**1.8**	3.2	3.1	3.8	6.6	4.2	17.6	21.4	6.1

Model and training hyperparameters. Model and training hyperparameters for our method on different datasets are listed in Table 7.

Table 7: Model and training hyperparameters for our method on different tasks.

Hyperparameter	Values/Search Space		
	QM9	MD17	rMD17
Number of layers	4, 5, 6	4, 6	4, 6
Hidden channels	128, 192, 256	256	256
Number of radial basis	24, 32, 96	16, 32, 64	16, 32, 64
Cutoff	5, 6, 6.5, 8	6, 8, 10	6, 8, 10
Epochs	800	1000	1000
Batch size	32	1, 4	1, 4
Learning rate	1e-4, 5e-4	5e-4	5e-4
Learning rate scheduler	steplr	steplr	steplr
Learning rate decay factor	0.5	0.5	0.5
Learning rate decay epochs	100	200	200