Equivariant Spherical Transformer for Efficient Molecular Modeling

Junyi An[*]
SAIS
anjunyi@sais.com.cn

Xinyu Lu[*†]
Shanghai Innovation Institute
Xiamen University
xinyulu@stu.xmu.edu.cn

Chao Qu
INFTech
quchao_tequila@inftech.ai

Yunfei Shi
SAIS
shiyunfei@sais.com.cn

Peijia Lin[†]
Sun Yat-sen University
Shanghai Innovation Institute
linpj6@mail2.sysu.edu.cn

Qianwei Tang[†]
Nanjing University
cleveland225@163.com

Licheng Xu
SAIS
xulicheng@sais.com.cn

Fenglei Cao[‡]
SAIS
caofenglei@sais.com.cn

Yuan Qi[‡]
SAIS
Fudan University
Zhongshan Hospital
qiyuan@fudan.edu.cn

Abstract

SE(3)-equivariant Graph Neural Networks (GNNs) have significantly advanced molecular system modeling by employing group representations. However, their message passing processes, which rely on tensor product-based convolutions, are limited by insufficient non-linearity and incomplete group representations, thereby restricting expressiveness. To overcome these limitations, we introduce the Equivariant Spherical Transformer (EST), a novel framework that leverages a Transformer structure within the spatial domain of group representations after Fourier transform. We theoretically and empirically demonstrate that EST can encompass the function space of tensor products while achieving superior expressiveness. Furthermore, EST's equivariant inductive bias is guaranteed through a uniform sampling strategy for the Fourier transform. Our experiments demonstrate state-of-the-art performance by EST on various molecular benchmarks, including OC20 and QM9.

1 Introduction

Graph neural networks (GNNs) have become increasingly prevalent for modeling molecular systems and approximating quantum mechanical calculations, providing crucial support for various computational chemistry tasks, including drug discovery [1] and material design [2]. Compared to traditional calculation methods like Density Functional Theory (DFT), GNNs significantly reduce the computational cost of quantum property prediction from hours or days to fractions of a second. However, directly applying regular GNNs (e.g. GCN [3] and GIN [4]) to 3D molecular conformations

[*]equal contribution. SAIS: Shanghai Academy of AI for Science

[†]This work is done when they are interns at SAIS

[‡]corresponding author

Preprint. Under review.

often yields poor results due to their neglect of inherent physical constraints. To overcome these limitations, equivariant GNNs have emerged as a more promising avenue, effectively capturing intricate atomic interactions and addressing multiple crucial challenges.

In molecular systems, SE(3)-invariance and SE(3)-equivariance are fundamental constraints. For instance, molecular energies remain constant under rotation, while atomic forces rotate in concert with the molecule equivariantly. Early invariant models, such as SchNet [5] and HIP-NN [6], rely on interatomic distances in their message-passing blocks, consequently limiting their ability to capture interactions involving triplets or quadruplets of atoms [7]. Directional GNNs [8, 9] address this problem by explicitly incorporating bond angles and dihedral angles. Despite performance improvements, their expressiveness remained constrained as they primarily extracted features from invariant and handcrafted features. To capture deeper features, SE(3)-equivariant GNNs employ group representations as node embeddings and construct steerable message-passing blocks [10–12], where tensor product are applied to capture equivariant interactions between group embeddings. These GNNs achieve improved performance using only raw molecular structures, without providing handcrafted features. Nevertheless, tensor product operations inherently suffer from limited non-linearity [10], and their expressiveness is bounded by the degree of group representations [13, 14]. An alternative equivariant approach processes features in the spatial domain after Fourier transforming from group representations, where embeddings are defined by square-integrable spherical functions (see Figure 1(a)). Prior works [15–17] typically apply simple point-wise neural networks to model signals on these spherical functions, operating independently on each orientation (see Figure 1(b)). *Introducing nonlinearity across different orientations within these spherical functions is rarely considered due to the risk of violating equivariance. Nevertheless, we show that this operation can be performed equivariantly, potentially providing additional nonlinearity and expressiveness* (Further analysis is provided in Section 3.2.)

In this paper, we present the Equivariant Spherical Transformer (EST), a novel message-passing framework for accurate atomic interaction modeling. First, we use the spherical functions in the spatial domain as node and edge embeddings, and transform it to point sequences, where each point represent a unique orientation on the spherical functions. A spherical attention mechanism is then introduced to capture dependencies within these point sequences (see Figure 1(c)). Furthermore, EST incorporates a mixture of hybrid experts structure, utilizing feed-forward networks in both spatial domain and original harmonic domain to effectively balance equivariance and model capacity. The equivariant inductive bias of EST is guaranteed through our implementation of a uniform sampling strategy in spatial domain. Our experiments show that EST-based architectures achieve state-of-the-art (SOTA) performance on various benchmarks such as OC20 [18] and QM9 [19], exhibiting particularly stable and significant improvements for complex molecular systems.

Our contributions are: (i) We propose EST, a novel framework whose expressiveness provably subsumes traditional tensor products. (ii) By employing an uniform sampling implementation, we guarantee the equivariance of EST. (iii) We conduct a range of experiments for quantum property prediction tasks, demonstrating superior performance of EST. Moreover, our ablation studies confirming the enhanced expressiveness and equivariance of EST.

2 Related Work and Preliminaries

In this section, we discuss related work and the corresponding mathematical background relevant to equivariant GNNs. We begin by listing the notations frequently used throughout the paper. We denote the unit sphere as $\mathbb{S}^2$, where the coordinate of a spherical point (or orientation) $\vec{\mathbf{p}} = (\theta, \varphi)$ is represented by its polar angle θ and azimuth angle φ. The symbol $\mathbb{R}$ represents the set of real numbers, while $\mathbf{R}$ denotes a rotation matrix for 3D vectors. We use $[\cdot, \cdot]$ to indicate tensor concatenation and $\tilde{\cdot}$ to represent the update of tensors, respectively.

Message Passing Neural Networks Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes $v_i \in \mathcal{V}$ and edges $e_{ij} \in \mathcal{E}$. Each node v_i has an embedding $\mathbf{x}_i$ and an attribute $\mathbf{z}_i$, and each edge e_{ij} has an attribute $\mathbf{a}_{ij}$. The Message Passing Neural Network (MPNN) [5], a specific type of GNN, updates node embeddings through a message block $\mathbf{M}(\cdot)$ and an update block $\mathbf{U}(\cdot)$ via the following steps:

$$\mathbf{m}_{ij} = \mathbf{M}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_i, \mathbf{z}_j, \mathbf{a}_{ij}, \vec{\mathbf{r}}_{ij}), \quad \text{and} \quad \tilde{\mathbf{x}}_i = \mathbf{U}(\mathbf{x}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}), \tag{1}$$

where we use $\vec{\mathbf{r}}_{ij}$ to denote the relative position of node i and node j in 3D space, and the neighborhood $\mathcal{N}(i)$ is typically defined by a cutoff radius: $\mathcal{N}(i) = \{j \mid \|\vec{\mathbf{r}}_{ij}\| \leq r_{cut}\}$. The node attribute $\mathbf{z}_i$

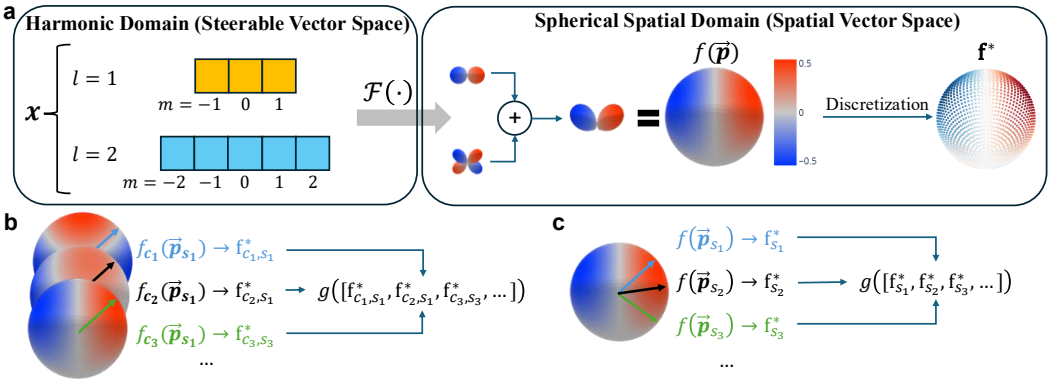


Figure 1: **Overview of operations in the spherical spatial domain.** (a) Applying the Fourier transform to group steerable representations, projecting them onto the spatial domain and storing them via spherical sampling. (b) A conventional point-wise operation on the sphere, where a function $g(\cdot)$ combines features across different spherical functions at a given orientation. (c) The proposed EST operation, where $g(\cdot)$ represents a Transformer framework designed to model global dependencies and interactions among different orientations on the sphere.

contains atomic information such as the atomic type, and the edge attribute $\mathbf{a}_{ij}$ contains atomic pair information such as distance and bond type. By stacking multiple message and update blocks, the final node embeddings can be used to model atomic interactions or represent molecular properties.

Equivariance and Invariance Given a group $\mathbf{G}$ and a transformation parameter $g \in \mathbf{G}$, a function $\phi : \mathcal{X} \to \mathcal{Y}$ is said to be equivariant to g if it satisfies:

$$\phi(T(g)[x]) = T'(g)[\phi(x)], \tag{2}$$

where $T'(g) : \mathcal{Y} \to \mathcal{Y}$ and $T(g) : \mathcal{X} \to \mathcal{X}$ denote the corresponding transformations on $\mathcal{Y}$ and $\mathcal{X}$, respectively. Invariance is a special case of equivariance where $T'(g)$ is the identity transformation. In this paper, we primarily focus on SO(3) equivariance, i.e., equivariance under 3D rotations, as it is closely related to the interactions between atoms in molecules [4].

Spherical Harmonics and Steerable Vectors Spherical harmonics, a set of orthonormal basis functions defined over the sphere $\mathbb{S}^2$, are commonly employed in equivariant models. The real-valued spherical harmonics are typically denoted as $\{Y^{(l,m)} : \mathbb{S}^2 \to \mathbb{R}\}$, where l represents the degree and m the order. For any orientation $\vec{\mathbf{p}}$, we define $\mathbf{Y}^{(l)}(\vec{\mathbf{p}}) = [Y^{(l,-l)}(\vec{\mathbf{p}}), Y^{(l,-l+1)}(\vec{\mathbf{p}}), ..., Y^{(l,l)}(\vec{\mathbf{p}})]$, a vector of size $2l+1$, and $\mathbf{Y}^{(0\to l)}(\vec{\mathbf{p}}) = [\mathbf{Y}^{(0)}(\vec{\mathbf{p}}), ..., \mathbf{Y}^{(l)}(\vec{\mathbf{p}})]$, a vector of size $(l+1)^2$.

A key property of spherical harmonics is their behavior under rotation $\mathbf{R} \in SO(3)$:

$$\mathbf{Y}^{(l)}(\mathbf{R}\vec{\mathbf{p}}) = \mathbf{D}^{(l)}(\mathbf{R})\mathbf{Y}^{(l)}(\vec{\mathbf{p}}), \tag{3}$$

where $\mathbf{D}^{(l)}(\mathbf{R})$ is a $(2l+1) \times (2l+1)$ matrix known as the Wigner-D matrix of degree l. Notably, $\mathbf{D}^{(1)}(\mathbf{R}) = \mathbf{R}$. Thus, $\mathbf{R}$ and $\mathbf{D}^{(l)}(\mathbf{R})$ can represent $T(g)$ and $T'(g)$ in Equation (2). Following the convention in [20, 11], we say that $\mathbf{Y}^{(l)}(\vec{\mathbf{p}})$ is steerable by the Wigner-D matrix of the same degree l. Furthermore, a vector that transforms according to an l-degree Wigner-D matrix is termed an l-degree steerable vector or a type-l vector, residing in the vector space $\mathbb{V}_l$. Further mathematical details are available in Appendix A.1.

A common practice in equivariant models is to use steerable vectors as node embedding and encode relative positions $\vec{\mathbf{r}}_{ij}$ using spherical harmonics as edge attributes. TFN [10] and NequIP [21] provide a general framework that learns the interaction between node embeddings and edge attributes through equivariant convolution filters. These filters are composed of equivariant operations such as degree-wise linear (DW-Linear) layers, Clebsch-Gordan (CG) tensor products, and Gate mechanisms. SEGNN [11] extends this convolution framework by more non-linear operations to achieve enhanced learning ability. SE(3)-Transformer [22] and Equiformer [12] introduce an attention mechanism that assigns rotation-invariant weights to edge attributes, thereby improving the learning of key atomic interactions. Nevertheless, the core operation in these works remains the convolution filter based on the CG tensor product. Additionally, GotenNet [23] proposes an effective and lightweight structure that separates steerable vectors into an $l = 0$ invariant component and $l > 0$ equivariant components, subsequently interacting these parts using inner products and scalar multiplication. In summary, due

[4] Invariance under translation is trivially satisfied by using relative positions as inputs.

to the requirement of equivariance, most equivariant model architectures are constrained to specific equivariant operations. In this work, we transform the steerable vector into spherical spatial domain, enabling the application of more flexible and highly non-linear structures within message passing.

Spherical Fourier Transform Any square-integrable function $f(\vec{\mathbf{p}})$ defined over the sphere $\mathbb{S}^2$ can be expressed in a spherical harmonic basis via the spherical Fourier transform $\mathcal{F}$:

$$f(\vec{\mathbf{p}}) = \mathcal{F}(\mathbf{x}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} x^{(l,m)} Y^{(l,m)}(\vec{\mathbf{p}}), \tag{4}$$

where $x^{(l,m)}$ is the spherical Fourier coefficient. Conversely, the coefficient can be obtained from the function $f(\vec{\mathbf{p}})$ via the inverse transform $\mathcal{F}^{-1}$:

$$x^{(l,m)} = \mathcal{F}^{-1}(f(\vec{\mathbf{p}})) = \int_{\Omega} f(\vec{\mathbf{p}}) Y^{(l,m)*}(\vec{\mathbf{p}}) d\Omega, \tag{5}$$

where $d\Omega$ denotes the unit sphere and $Y^{(l,m)*}(\cdot)$ is the conjugate of the spherical harmonic function. The function $f(\vec{\mathbf{p}})$ and its coefficients $\mathbf{x}$ are referred to as the representations in the spherical spatial domain and the harmonic domain, respectively. SCN [15] introduced an approach that applies a simple point-wise neural network in the spherical spatial domain. This technique has been subsequently adopted by several works [16, 17] within their message-passing frameworks. However, point-wise neural networks inherently limit their expressiveness. Our work aims to extend the capabilities of learning within the whole spherical spatial domain.

3 Model

In this section, we first make a review of tensor product to detail our motivation. Then, we introduction the formulation of our EST and clarify its properties.

3.1 Review of Clebsch-Gordan Tensor Product

A common approach in equivariant models is to incorporate steerable vectors with multiple degrees for node or edge features and then employing CG tensor products $\otimes : \mathbb{V}_{l_1} \times \mathbb{V}_{l_2} \rightarrow \mathbb{V}_l$, which is defined as:

$$(\mathbf{x}_1 \otimes \mathbf{x}_2)^{(l,m)} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} \mathcal{C}^{(l,m)}_{(l_1,m_1)(l_2,m_2)} \mathbf{x}_1^{(l_1,m_1)} \mathbf{x}_2^{(l_2,m_2)}, \tag{6}$$

where $\mathcal{C}^{(l,m)}_{(l_1,m_1)(l_2,m_2)}$ are the CG coefficients, a sparse tensor yielding non-zero terms when $|l_1 - l_2| \leq l \leq (l_1 + l_2)$. These products enable interactions between various combinations of l_1 and l_2, crucial for the expressive power of MPNNs in representing latent equivariant functions [13, 14]. However, CG tensor products lack high non-linearities [11], often requiring the stacking of multiple layers to capture complex molecular features. Furthermore, their high computational complexity ($\mathcal{O}(L^6)$) [24] makes architectures with stacked tensor product layers computationally expensive [11]. While techniques like Gate mechanisms [11, 12] and geometry-aware tensor attention [25] can be used to mitigate this limitation, they apply non-linear transformations to the invariant ($l = 0$) representations. Higher-degree representations ($l > 0$) are then updated through a scaling multiplication:

$$C_{inv} = \mathrm{NL}(\mathbf{x}^{(0)}, \sum_{l=1}^{L} \mathrm{d}(\mathbf{x}^{(l)})), \quad \text{and} \quad \tilde{\mathbf{x}}^{(l)} = C_{inv} \cdot \mathbf{x}^{(l)}, \tag{7}$$

where C_{inv} is an invariant scalar, $\mathrm{NL}(\cdot)$ is a network with non-linear functions, and $\mathrm{d}(\cdot) : \mathbb{V}_l \rightarrow \mathbb{V}_0$ maps equivariant features to invariants, such as an inner product. This indirect exchange of information between higher-degree representations still limits the overall expressiveness.

3.2 Equivariant Spherical Transformer

3.2.1 Node embedding

Steerable Representation For representation with steerable vectors, we first define a maximal degree L. Each node n's embedding $\mathbf{x}_n$ comprises C channels, and each channel is a concatenation

of steerable vectors from degree 0 to L: $\mathbf{x}_{n,c} = [x_{n,c}^{(0)}, \mathbf{x}_{n,c}^{(1)}, ..., \mathbf{x}_{n,c}^{(L)}]$. As a result, the dimension of a steerable node embedding is $(L+1)^2 \times C$.

Spatial Representation. Steerable representations $\mathbf{x}_{n,c}$ can be transformed into a spherical function $f_{n,c}(\vec{\mathbf{p}})$ via Equation (4), with the summation truncated at a maximum degree L. Furthermore, $f_{n,c}(\vec{\mathbf{p}})$ is discretely represented by sampling S points on the sphere, yielding a spatial node embedding $\mathbf{f}_n^* \in \mathbb{R}^{S \times C}$. It can be expressed as the concatenation over sampled points:

$$\mathbf{f}_{n,c}^* = [(\mathbf{x}_{n,c}^{(0\to L)})^T \mathbf{Y}^{(0\to L)}(\vec{\mathbf{p}}_1), (\mathbf{x}_{n,c}^{(0\to L)})^T \mathbf{Y}^{(0\to L)}(\vec{\mathbf{p}}_2), ..., (\mathbf{x}_{n,c}^{(0\to L)})^T \mathbf{Y}^{(0\to L)}(\vec{\mathbf{p}}_S)]. \quad (8)$$

We use the term $\mathbf{f}_{n,c,s}^*$ to denote the signal at the sampled point $\vec{\mathbf{p}}_s$ of channel c of node n. The inverse transform in Equation (5) can convert $\mathbf{f}_n^*$ back to $\mathbf{x}_n$.

In our model, the standard state of the node embedding is a steerable representation and it is transformed into a spatial representation when specific spherical operations are performed. Furthermore, the initial node embedding is constructed from two components: invariant mappings of the atomic attributes (the $l = 0$ part) and the spherical harmonics of its neighbors' relative positions (the $l > 0$ part).

Operations in spatial domain Upon transformation from steerable representation to the spatial representation, the explicit degree and order is no longer directly accessible. Instead, higher-degree information is encoded as geometric features distributed across the sphere (see Figure 1(a)). Intuitively, by modeling the interdependencies of these geometric features, the model can effectively exchange information between degrees of original steerable representation, thereby approximating the capability of the tensor product. However, prior methods [15–17] only consider the feature dependencies across channels within the same sampling point, e.g. $\mathbf{f}_{c_1,s_1}^*$ and $\mathbf{f}_{c_2,s_1}^*$ (see Figure 1(b)), overlooking geometric features composed of multiple points. In contrast, our proposed Equivariant Spherical Transformer (EST) can model dependencies across various sampling points, e.g. $(f(\vec{\mathbf{p}}_{s_1}), f(\vec{\mathbf{p}}_{s_2}), ...)$ or $(\mathbf{f}_{s_1}^*, \mathbf{f}_{s_2}^*, ...)$. In the following, we present two key theoretical results: (i) the function set representable by CG tensor products can be approximated via EST, and (ii) by employing an uniform sampling strategy on the sphere, EST is SO(3)-equivariant.

3.2.2 Spherical Attention

Treating $\mathbf{f}^*$ as a sequential data, where each sampled point s has a C-dimensional feature vector, we apply a attention mechanism [26]:

$$a_{s_i,s_j} = \frac{\exp(\mathbf{Q}_{s_i}\mathbf{K}_{s_j}^T/\sqrt{C})}{\sum_{s_k=1}^{S}\exp(\mathbf{Q}_{s_i}\mathbf{K}_{s_k}^T/\sqrt{C})}, \quad \text{and} \quad \tilde{\mathbf{f}}_{s_i}^* = \sum_{s_j=1}^{S} a_{s_i,s_j}\mathbf{V}_{s_j}, \quad (9)$$

where the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ matrices are obtained through point-wise linear (PW-Linear) transformations of $\mathbf{f}^*$. The spherical attention (SA) is the core of EST. We also incorporate the pre-LayerNorm strategy [27] and feedforward networks (FFN) in EST. Our EST is applied in two ways: (i) for interactions between two spatial representations, $\mathbf{Q}$ is derived from one, and $\mathbf{K}$ and $\mathbf{V}$ from the other; (ii) for learning features within a single spatial representation, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are all derived from it.

Expressiveness Assuming no information loss during the spherical Fourier transform and its inverse, i.e., $\mathbf{x} = \mathcal{F}^{-1}\mathcal{F}(\mathbf{x})$, Theorem 1 can evaluate the expressive power of EST.

Theorem 1. *For any two steerable representations $\mathbf{u} \in \mathbb{V}_{0\to l_1}$ and $\mathbf{v} \in \mathbb{V}_{0\to l_2}$, the spatial representation of their CG tensor product, $\mathbf{u} \otimes \mathbf{v}$, can be uniformly approximated by EST modules operating on the spatial representations of $\mathbf{u}$ and $\mathbf{v}$.*

The proof can be found in Appendix B.1. We note that a Fourier transform without information loss is attainable by satisfying the Nyquist sampling rate, which mandates a minimum of $(2L)^2$ sampling points on the sphere [15]. Therefore, Theorem 1 substantiates the potential for substituting tensor products with EST. Furthermore, we present an empirical comparison of the expressiveness of EST and tensor product-based operations in Section 4.3, where our EST significantly surpasses the theoretical upper bound of tensor products.

Equivariance We demonstrate that the equivariance of EST is related to the uniformity of the sampling points in Fourier transform.

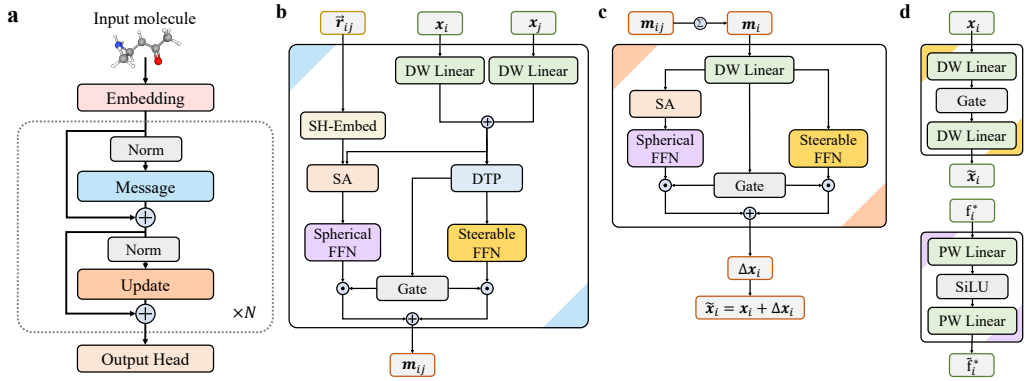


Figure 2: **The architecture and building blocks of EST.** SH and DTP denote spherical harmonic embedding and depth-wise tensor product [12], respectively. For simplicity, the Fourier and inverse Fourier transform steps are omitted. (a) Overall architecture. (b) Message block. (c) Update block. (d) Two experts operating on the steerable and spatial representations, respectively.

Theorem 2. *If the sampling point density is the same across any local region of the sphere, EST operation preserves SO(3)-equivariance:*

$$\mathcal{F}^{-1}\big(\mathrm{EST}\big(\mathcal{F}(\mathbf{D}\mathbf{x})\big)\big) = \mathbf{D}\mathcal{F}^{-1}\big(\mathrm{EST}\big(\mathcal{F}(\mathbf{x})\big)\big), \tag{10}$$

where $\mathbf{D}$ *represents an arbitrary Wigner-D rotation matrix.*

The proof is provided in Appendix B.2. Most prior works [16, 17] implement spherical Fourier transforms using the e3nn library [28]. However, e3nn does not guarantee a uniform sampling point density across all regions of the sphere. To address this, we redefine the sampling implementation using Fibonacci Lattices (FL) [29]. The Cartesian coordinate of each sampling point s is defined by

$$\vec{\mathbf{p}}_s = [\sqrt{1-z_s^2}\cos(2s\pi/\lambda), \sqrt{1-z_s^2}\sin(2s\pi/\lambda), z_s], \tag{11}$$

where $z_s = \frac{2s-1}{S-1}$ and $\lambda = \frac{1+\sqrt{5}}{2}$ denotes the golden ratio. The Fourier basis functions must satisfy orthogonality, i.e., $\int_\Omega Y^{(l,m)}(\vec{\mathbf{p}})Y^{(l',m')*}(\vec{\mathbf{p}})d\Omega = \delta_{ll'}\delta_{mm'}$, where δ_{ij} is the Kronecker delta. Consequently, we redefine the implementation of conjugate functions for the inverse transform as:

$$Y^{(l,m)*}(\vec{\mathbf{p}}) = \lambda^{(l,m)}Y^{(l,m)}(\vec{\mathbf{p}}), \tag{12}$$

where $\lambda^{(l,m)} = 1/\sum_s |Y^{(l,m)}(\vec{\mathbf{p}}_s)|^2$. We provide detailed discussions between our sampling and the e3nn implementation in the Appendix B.2. In addition, we conducted ablation experiments in Section 4.3 to demonstrate that sampling methods with poor uniformity can severely compromise equivariance.

Relative Orientation Embedding on the Sphere Despite its strong expressiveness, EST lacks the inherent ability to capture orientation relationships between sampling points, potentially leading to ambiguities [30]. For example, if the features at points s_1 and s_2 are identical ($\mathbf{f}^*_{s_1} = \mathbf{f}^*_{s_2}$), their contributions to other sampling points s_i become indistinguishable. To address this, we introduce a relative orientation embedding by augmenting the query $\mathbf{Q}$ and key $\mathbf{K}$ vectors with orientation information:

$$\mathbf{Q}_{s_i} := [\mathbf{Q}_{s_i}, \vec{\mathbf{p}}_{s_i}], \quad \mathbf{K}_{s_j} := [\mathbf{K}_{s_j}, \vec{\mathbf{p}}_{s_j}], \quad \forall s_i, s_j \in \{1, 2, ..., S\}. \tag{13}$$

Consequently, the inner product between $\mathbf{Q}_{s_i}$ and $\mathbf{K}_{s_j}$ now incorporates a term aware of the orientation, related to $\vec{\mathbf{p}}^T_{s_i}\vec{\mathbf{p}}_{s_j}$. Moreover, this augmentation does not compromise the equivariant inductive bias of EST, as proven in Appendix B.3. The intuition behind our approach shares similarity with rotary position embeddings in NLP [31], where relative position embedding should be invariant under transformation. However, while NLP focuses on permutation invariance, our approach addresses rotation invariance, necessitating a fundamentally different implementation.

3.2.3 Mixture of Hybrid Experts

To enhance the model's representational capacity, we utilize equivariant FFNs within a Mixture of Experts (MoE) framework [32]. Starting with the steerable representation $\mathbf{x}$, the invariant ($l = 0$) component is used to compute expert weights through a gate function:

$$G(\mathbf{x}) = \mathrm{Softmax}(\mathbf{x}^{(0)}\mathbf{W}_G), \tag{14}$$

where $\mathbf{W}_G \in \mathbb{R}^{C\times E}$ is a trainable weight matrix. As illustrated in Figure 2(c), we employ two distinct expert structures: (1) a steerable FFN, operating on steerable inputs and consisting of two DW-Linear layers with an intermediate Gate activation [12]; and (2) a spherical FFN, processing spatial inputs and comprising two PW-Linear layers with an intermediate SiLU activation [33]. The final output is a weighted combination of the outputs from these hybrid experts, with the weights determined by Equation (14). These two expert types offer a crucial trade-off: the steerable FFN guarantees strict equivariance, while the spherical FFN provides enhanced expressive power. We therefore adapt their numbers based on the specific demands of the task. Additionally, we investigate common MoE techniques, including balancing loss [34] and shared experts [35]. Further details and ablation studies can be found in Appendix C.

3.3 Overall Architecture

EST can be employed both in the message block to compute edge features and in the update block to refine node embeddings. Figure 2(b,c) defines a message passing layer with EST. When computing the message, the $\mathbf{Q}$ in SA come from the aggregation of node i and j embeddings, while $\mathbf{K}$ and $\mathbf{V}$ is derived from the spherical harmonic representation of the relative position $\vec{\mathbf{r}}_{ij}$. In the SA of the update block, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are all derived from the aggregated message.

EST serves as a flexible building block and can also be integrated into other equivariant models. Since the primary function of EST is to extract critical interactions or geometric features from complex high-degree representations, we prefer to replace the update blocks in existing models with our EST-based update block, where the input (aggregated message) contains sufficient and minimally processed neighbor features. With this strategy, EST-based architectures can also benefit from the computational efficiency of lightweight message blocks, while keep advantages to enhance the expressiveness.

4 Experiments

In this section, we construct experiments to investigate the efficacy of the proposed method. We evaluate its performance on the S2EF/IS2RE tasks from the OC20 [18] benchmark and QM9 [19] tasks. Our analysis involves a comparison with a wide range of baseline models, including Schnet [5], PaiNN [36], SEGNN [11], TFN [10], Dimenet++ [37], Equiformer [12], and Equiformerv2 [17]. Specifically for the OC20 benchmark, we extend our comparison to include SpinConv [38], GemNet [9, 39], SphereNet [40], SCN [15] and eSCN [16]. Similarly, our QM9 experiments are augmented with comparisons to TorchMD-NET [41], EQGAT [42] and GotenNet [23]. The specific configurations employed for each baseline model can be found in Appendix D.1.

4.1 OC20 Results

The OC20 dataset, comprising over 130 million molecular structures with force and energy labels, covers a broad spectrum of materials, surfaces, and adsorbates. We evaluate our models on two core sub-datasets of OC20: S2EF and IS2RE. All experimental configurations are detailed in Appendix D.2. Notably, We did not use very deep models or very long training schedules. For instance, our S2EF experiments utilize only 8 layers, and our IS2RE experiments employ 6 layers, significantly fewer than many advanced methods. Furthermore, as mentioned in Section 3.3, to enhance computational efficiency and facilitate a more direct comparison of different fundamental modules, we replace the EST architecture's message module with the graph attention modules from EquiformerV2 and Equiformer for S2EF and IS2RE tasks, respectively.

S2EF Results We trained an 8-layer EST model on the S2EF-All dataset and evaluated its performance on the S2EF validation sets. Each validation set was divided into four similarly-sized subsets:

Table 1: S2EF results of models trained on S2EF-All training dataset. λ_E is the coefficient of the energy loss. "All+MD" denotes training models with additional OC20 MD dataset.

Model	Number of parameters	Throughput Samples/s	Energy MAE (meV) ↓	Force MAE (meV/Å) ↓
SchNet	9.1M	-	549	56.8
DimeNet++-L-F+E	10.7M	4.6	515	32.8
SpinConv	8.5M	6.0	371	41.2
GemNet-dT	32M	25.8	315	27.2
GemNet-OC	39M	18.3	244	21.7
SCN (20 layers, test)	271M	-	244	17.7
eSCN (20 layers, test)	200M	2.9	242	17.1
EquiformerV2 ($\lambda_E = 4$, 8 layers, All+MD)	31M	7.1	232	16.3
EquiformerV2 ($\lambda_E = 2$, 20 layers)	153M	1.8	236	**15.7**
EST ($\lambda_E = 4$, 8 layers)	45M	6.8	**231**	16.1

ID, OOD Ads, OOD Cat, and OOD Both. Consistent with prior work [16, 17], the best model was selected based on its performance on the ID validation subset during training and subsequently evaluated across all validation subsets. For comparison with SCN and eSCN, we report their test results due to the absence of publicly available validation results; previous work [17] indicates that the validation and test sets have a similar distribution. We compared our results with two variants of the SOTA EquiformerV2 model: a deep 20-layer architecture and one trained with additional MD dataset. As shown in Table 1, EST outperformed both EquiformerV2 variants in energy prediction. Furthermore, EST achieved competitive results in force prediction, surpassing several deeper baselines including SCN and eSCN.

Table 2: IS2RE results of models trained on IS2RE training dataset. Bold and underline indicate the best result, and the second best result, respectively.

	Energy MAE (meV) ↓				EwT (%) ↑			
Model	ID	OOD Ads	OOD Cat	OOD Both	ID	OOD Ads	OOD Cat	OOD Both
SchNet	639	734	662	704	2.96	2.33	2.94	2.21
PaiNN	575	783	604	743	3.46	1.97	3.46	2.28
TFN	584	766	636	700	4.32	2.51	4.55	2.66
DimeNet++	562	725	576	661	4.25	2.07	4.10	2.41
GemNet-dT	527	758	549	702	4.59	2.09	4.47	2.28
GemNet-OC	560	711	576	671	4.15	2.29	3.85	2.28
SphereNet	563	703	571	638	4.47	2.29	4.09	2.41
SEGNN	533	692	537	679	**5.37**	2.46	<u>4.91</u>	2.63
Equiformer	<u>504</u>	688	<u>521</u>	630	5.14	2.41	4.67	<u>2.69</u>
SCN (16 layers)	516	**643**	530	<u>604</u>	4.92	**2.71**	4.42	**2.76**
EST (6 layers)	**501**	<u>652</u>	**502**	**578**	<u>5.16</u>	<u>2.67</u>	**5.16**	**2.76**

IS2RE Results We trained our model directly on the IS2RE training set for energy prediction, without utilizing S2EF data. The results, summarized in Table 2, show that EST achieves SOTA performance on the ID, OOD Ads and OOD Both tasks and the second best on OOD Cat. Notably, the current SOTA models on OOD Cat, SCN, rely on complex 16-layer architectures with over 100M parameters. In contrast, EST employs a unified 6-layer structure comprising only 32.47M parameters, yet yields performance closely approaching theirs. Crucially, SCN relax equivariance to gain expressiveness, which can potentially lead to unstable predictions under input rotation. EST, conversely, provides more stable predictions owing to its stronger equivariance. Furthermore, in these IS2RE experiments, the EST architecture incorporates the message block from Equiformer. The key architectural difference resides in the update block. We observe that EST consistently surpasses the Equiformer model across all evaluated metrics, providing further evidence for the effectiveness of the proposed EST module and the model merging strategy discussed in Section 3.3.

Table 3: Results on QM9 dataset for various properties. † denotes using different data partitions.

Task Units	α $bohr^3$	$\Delta\varepsilon$ meV	ε_{HOMO} meV	ε_{LUMO} meV	μ D	C_v cal/(mol K)	G meV	H meV	R^2 $bohr^3$	U meV	U_0 meV	$ZPVE$ meV
SchNet	.235	63	41	34	.033	.033	14	14	.073	19	14	1.70
TFN†	.223	58	40	38	.064	.101	-	-	-	-	-	-
DimeNet++	.044	33	25	20	.030	.023	8	7	.331	6	6	**1.21**
PaiNN	.045	46	28	20	.012	.024	7.35	5.98	.066	**5.83**	5.85	1.28
TorchMD-NET	.059	36	20	18	.011	.026	7.62	6.16	**.033**	6.38	6.15	1.84
SEGNN†	.060	42	24	21	.023	.031	15	16	.660	13	15	1.62
EQGAT	.053	32	20	16	.011	.024	23	24	.382	25	25	2.00
Equiformer	.046	30	15	14	.011	.023	7.63	6.63	.251	6.74	6.59	1.26
EquiformerV2	.050	29	14	13	**.010**	.023	7.57	6.22	.186	6.49	6.17	1.47
EST	.042	**28**	**13**	**12**	.011	.022	**7.03**	**5.94**	.298	5.92	**5.64**	1.31
EST (with GA)	**.041**	29	14	13	.011	**.021**	7.18	6.17	.227	6.35	6.32	1.27

4.2 QM9 Results

The QM9 benchmark comprises quantum chemical properties for a relevant, consistent, and comprehensive chemical space of 134k equilibrium small organic molecules containing up to 29 atoms. Each atom is represented by its 3D coordinates and an embedding of its atomic type (H, C, N, O, F). We developed two model architectures: a 6-layer model employing the message-passing layer detailed in Figure 2, and a separate model that integrates message blocks (GA module) from Equiformer. For additional configuration specifics, please refer to Appendix D.2.

Given that the QM9 dataset is considerably smaller than OC20, models are more prone to overfitting if equivariance is destroied. Ablation study in Table 5 shows that practical implementation for Fourier transform may lead to minor deviations from perfect equivariance. Nevertheless, EST and EST (with GA) achieve best on eight of the twelve tasks (see Table 3). Notably, EST (with GA) significantly outperformed Equiformer despite utilizing a similar message block. Furthermore, we included a comparison with GotenNet [25], which employs a more extensive training schedule. To ensure a fair assessment, we applied this identical schedule to EST as well; see Appendix D.3 for details.

4.3 Ablation study

In this section, we explore two key properties of EST: expressiveness and equivariance. The other experiments can be found in Appendix D, where we comprehensively investigate the influence of building components, including SA, spherical relative orientation embedding and hybrid experts.

Table 4: Experiments on Rotationally symmetric structures.

GNN Layer	2 fold	3 fold	10 fold	100 fold	GNN Layer	2 fold	3 fold	5 fold	10 fold
$\text{EST}_{L=1}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	$\text{TFN/MACE}_{L=1}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$\text{EST}_{L=2}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	$\text{TFN/MACE}_{L=2}$	**100.0** ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$\text{EST}_{L=3}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	$\text{TFN/MACE}_{L=3}$	**100.0** ± 0.0	**100.0** ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$\text{EST}_{L=5}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	$\text{TFN/MACE}_{L=5}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	50.0 ± 0.0
$\text{EST}_{L=10}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	$\text{TFN/MACE}_{L=10}$	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0	**100.0** ± 0.0

Expressiveness Following [14], we employ evaluation metrics that distinguish n-fold symmetric structures to precisely assess EST's expressive power. We first construct a single-layer message-passing layer based on Figure 2, consistent with tensor product-based operations (TFN, MACE). Additionally, we remove the steerable FFN for a clear comparison. As demonstrated in [13, 14], the expressive power of tensor product-based models is limited by the maximum degree L, failing to perfectly distinguish n-fold symmetric structures when $n > L$. From Table 4, we observe that EST significantly breaks through this theoretical boundary. It allows the model to distinguish symmetric structures even with very high fold (e.g., using 1-degree EST for 100-fold structures).

Equivariance While we theoretically demonstrate that EST can achieve strict equivariance, perfect uniform sampling is challenging in practice, potentially leading to loss of equivariance. To investigate it, we build untrained networks with 1 to 6 layers based on Figure 2 and control the number of spherical sampling points in the Fourier transforms within the message and update blocks. We randomly select 1000 molecules from QM9, compute their outputs $y_1, ..., y_{1000}$, and then compute the outputs after applying random rotations, $\hat{y}_1, ..., \hat{y}_{1000}$. The average absolute error $(1/1000)\sum_{i=1}^{1000} |y_i - \hat{y}_i|$ serves as our measure of equivariance loss. As shown in Table 5, EST with FL sampling closely approximates

Table 5: Equivariance evaluation with different sampling strategies.

Sampling Strategy/Number	1-layer	2-layer	3-layer	4-layer	5-layer	6-layer
FL / $[64, 128]$	0.0009	0.0013	0.0014	0.0016	0.0025	0.0018
FL / $[128, 256]$	0.0004	0.0010	0.0007	0.0005	0.0015	0.0013
FL / $[256, 256]$	**0.0002**	**0.0002**	**0.0002**	**0.0003**	**0.0001**	**0.0002**
e3nn / $[210, 210]$	0.0084	0.1006	0.0366	0.0593	0.0046	0.0199

strict equivariance even without training, and its equivariance further improves with an increasing number of spherical sampling points. Additionally, EST with the sampling implementation in e3nn show a poor equivariance.

5 Conclusion

We present EST, a new SE(3)-equivariant framework for modeling molecules. By integrating a Transformer structure with steerable vectors, EST offers greater expressive power than tensor-product-based frameworks. We showed EST's strong performance on the OC20 and QM9 datasets. A current limitation is the equivariance loss caused by spherical sampling. Future work will focus on improving spherical sampling uniformity. This could lead to fewer sampling points and even better equivariance.

References

[1] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[2] C Lawrence Zitnick, Lowik Chanussot, Abhishek Das, Siddharth Goyal, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Thibaut Lavril, Aini Palizhati, Morgane Riviere, et al. An introduction to electrocatalyst design using machine learning for renewable energy storage. *arXiv preprint arXiv:2010.09435*, 2020.

[3] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR 2017*, 2017.

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

[5] 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.

[6] Nicholas Lubbers, Justin S Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *The Journal of chemical physics*, 148(24):241715, 2018.

[7] Benjamin Kurt Miller, Mario Geiger, Tess E Smidt, and Frank Noé. Relevance of rotationally equivariant convolutions for predicting molecular properties. *arXiv preprint arXiv:2008.08461*, 2020.

[8] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *ICLR2020*, 2020.

[9] 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.

[10] 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. *Neurips 2018*, 2018.

[11] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. *The Tenth International Conference on Learning Representations*, 2022.

[12] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

[13] Nadav Dym and Haggai Maron. On the universality of rotation equivariant point cloud networks. In *International Conference on Learning Representations*, 2021.

[14] Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. *arXiv preprint arXiv:2301.09308*, 2023.

[15] C Lawrence Zitnick, Abhishek Das, Adeesh Kolluru, Janice Lan, Muhammed Shuaibi, Anuroop Sriram, Zachary Ulissi, and Brandon Wood. Spherical channels for modeling atomic interactions. *Neurips 2022*, 2022.

[16] Saro Passaro and C Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant gnns. volume 202 of *Proceedings of Machine Learning Research*, pages 27420–27438. PMLR, 2023.

[17] Yi-Lun Liao, Brandon M Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In *The Twelfth International Conference on Learning Representations*, 2024.

[18] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021.

[19] 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.

[20] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[21] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

[22] 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.

[23] Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[24] Shengjie Luo, Tianlang Chen, and Aditi S. Krishnapriyan. Enabling efficient equivariant operations in the fourier basis via gaunt tensor products. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[25] Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[27] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533, 2020.

[28] Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022.

[29] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical geosciences*, 42:49–64, 2010.

[30] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *Computational Linguistics*, 48(3):733–763, 2022.

[31] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[33] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[34] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[35] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[36] 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.

[37] Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

[38] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, and C Lawrence Zitnick. Rotation invariant graph neural networks using spin convolutions. *arXiv preprint arXiv:2106.09575*, 2021.

[39] Johannes Gasteiger, Muhammed Shuaibi, Anuroop Sriram, Stephan Günnemann, Zachary Ward Ulissi, C. Lawrence Zitnick, and Abhishek Das. Gemnet-OC: Developing graph neural networks for large and diverse molecular simulation datasets. *Transactions on Machine Learning Research*, 2022.

[40] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *ICLR2022*, 2021.

[41] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022.

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

APPENDIX

A The Mathematics

A.1 The Mathematics of Spherical Harmonics

A.1.1 The Properties of Spherical Harmonics

The spherical harmonics $Y^{(l,m)}(\theta, \varphi)$ are the angular portion of the solution to Laplace's equation in spherical coordinates where azimuthal symmetry is not present. Some care must be taken in identifying the notational convention being used. In this entry, θ is taken as the polar (colatitudinal) coordinate with θ in $[0, \pi]$, and φ as the azimuthal (longitudinal) coordinate with φ in $[0, 2\pi)$.

Spherical harmonics satisfy the spherical harmonic differential equation, which is given by the angular part of Laplace's equation in spherical coordinates. If we define the solution of Laplace's equation as $F = \Phi(\varphi)\Theta(\theta)$, the equation can be transformed as:

$$\frac{\Phi(\varphi)}{\sin\theta}\frac{d}{d\theta}\left(\sin\theta\frac{d\Theta}{d\theta}\right) + \frac{\Theta(\theta)}{\sin^2\theta}\frac{d^2\Phi(\varphi)}{d\varphi^2} + l(l+1)\Theta(\theta)\Phi(\varphi) = 0 \tag{15}$$

Here we omit the derivation process and just show the result. The (complex-value) spherical harmonics are defined by:

$$Y^{(l,m)}(\theta, \varphi) \equiv \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P^{(l,m)}(\cos\theta)e^{im\varphi}, \tag{16}$$

where $P^{(l,m)}(\cos\theta)$ is an associated Legendre polynomial. Spherical harmonics are integral basis, which satisfy:

$$\begin{aligned}&\int_0^{2\pi}\int_0^{\pi} Y^{(l_1,m_1)}(\theta,\varphi)Y^{(l_2,m_2)}(\theta,\varphi)Y^{(l_3,m_3)}(\theta,\varphi)\sin\theta d\theta d\varphi\\&=\sqrt{\tfrac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}}\begin{pmatrix} l_1 & l_2 & l_3\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix} l_1 & l_2 & l_3\\ m_1 & m_2 & m_3\end{pmatrix},\end{aligned}\tag{17}$$

where $\begin{pmatrix} l_1 & l_2 & l_3\\ m_1 & m_2 & m_3\end{pmatrix}$ is a Wigner 3j-symbol (which is related to the Clebsch-Gordan coefficients). We list a few spherical harmonics which are:

$$\begin{aligned}Y^{(0,0)}(\theta,\varphi)&=\frac{1}{2}\sqrt{\frac{1}{\pi}},\\Y^{(1,-1)}(\theta,\varphi)&=\frac{1}{2}\sqrt{\frac{3}{2\pi}}\sin\theta e^{-i\varphi},\\Y^{(1,0)}(\theta,\varphi)&=\frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta,\\Y^{(1,1)}(\theta,\varphi)&=\frac{-1}{2}\sqrt{\frac{3}{2\pi}}\sin\theta e^{i\varphi},\\Y^{(2,-2)}(\theta,\varphi)&=\frac{1}{4}\sqrt{\frac{15}{2\pi}}\sin^2\theta e^{-2i\varphi},\\Y^{(2,-1)}(\theta,\varphi)&=\frac{1}{2}\sqrt{\frac{15}{2\pi}}\sin\theta\cos\theta e^{-i\varphi},\\Y^{(2,0)}(\theta,\varphi)&=\frac{1}{4}\sqrt{\frac{5}{\pi}}\left(3\cos^2\theta-1\right),\\Y^{(2,1)}(\theta,\varphi)&=\frac{-1}{2}\sqrt{\frac{15}{2\pi}}\sin\theta\cos\theta e^{i\varphi},\\Y^{(2,2)}(\theta,\varphi)&=\frac{1}{4}\sqrt{\frac{15}{2\pi}}\sin^2\theta e^{2i\varphi},\end{aligned}\tag{18}$$

In this work, we use the real-value spherical harmonics rather than the complex-value one, which can be written as :

$$\begin{aligned}Y^{0,0}(\theta,\varphi)&=\sqrt{\frac{1}{4\pi}},\\Y^{(1,-1)}(\theta,\varphi)&=\sqrt{\frac{3}{4\pi}}\sin\varphi\sin\theta,\\Y^{(1,0)}(\theta,\varphi)&=\sqrt{\frac{3}{4\pi}}\cos\theta,\\Y^{(1,1)}(\theta,\varphi)&=\sqrt{\frac{3}{4\pi}}\cos\varphi\sin\theta,\\Y^{(2,-2)}(\theta,\varphi)&=\sqrt{\frac{15}{16\pi}}\sin(2\varphi)\sin^2\theta,\\Y^{(2,-1)}(\theta,\varphi)&=\sqrt{\frac{15}{4\pi}}\sin\varphi\sin\theta\cos\theta,\\Y^{(2,0)}(\theta,\varphi)&=\sqrt{\frac{5}{16\pi}}(3\cos^2\theta-1),\\Y^{(2,1)}(\theta,\varphi)&=\sqrt{\frac{15}{4\pi}}\cos\varphi\sin\theta\cos\theta,\\Y^{(2,2)}(\theta,\varphi)&=\sqrt{\frac{15}{16\pi}}\cos(2\varphi)\sin^2\theta.\end{aligned}\tag{19}$$

A.1.2 Fourier transformation over $\mathbb{S}^2$

In the main paper, we show that any square-integrable function $f(\cdot)$ can thus be expanded as a linear combination of spherical harmonics:

$$f(\vec{\mathbf{p}}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mathbf{x}^{(l,m)} Y^{(l,m)}(\vec{\mathbf{p}}), \tag{20}$$

where $\vec{\mathbf{p}} = (\theta, \varphi)$ denotes the orientations, like what we do in the main paper. The coefficient $\mathbf{x}^{(l,m)}$ can be obtained by the inverse transformation over $\mathbb{S}^2$, which is

$$\mathbf{x}^{l,m} = \int_{\Omega} f(\vec{\mathbf{p}}) Y^{(l,m)*}(\vec{\mathbf{p}}) d\Omega = \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \sin\theta f(\vec{\mathbf{p}}) Y^{(l,m)*}(\vec{\mathbf{p}}). \tag{21}$$

Using the fact $\mathbf{Y}^l(\mathbf{R}\vec{\mathbf{p}}) = \mathbf{D}^l(\mathbf{R})\mathbf{Y}^l(\vec{\mathbf{p}})$, and Equation (20), we know

$$f(\mathbf{R}\vec{\mathbf{p}}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mathbf{x}^{(l,m)} Y^{(l,m)}(\mathbf{R}\vec{\mathbf{p}}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mathbf{x}^{(l,m)} \mathbf{D} Y^{(l,m)}(\vec{\mathbf{p}}). \tag{22}$$

Therefore, we can get the conclusion that spatial representation $f(\mathbf{R}\vec{\mathbf{P}})$ and $f(\vec{\mathbf{P}})$ is steerable, which can be represented by

$$f(\mathbf{R}\vec{\mathbf{p}}) = \mathbf{D}^{-1}\mathbf{x} = \mathbf{D}^T\mathbf{x}. \tag{23}$$

A.1.3 The Relationship Between Spherical Harmonics and Wigner-D Matrix

A rotation $\mathbf{R}$ sending the $\vec{\mathbf{p}}$ to $\mathbf{R}\vec{\mathbf{p}}$ can be regarded as a linear combination of spherical harmonics that are set to the same degree. The coefficients of linear combination represent the complex conjugate of an element of the Wigner D-matrix. The rotational behavior of the spherical harmonics is perhaps their quintessential feature from the viewpoint of group theory. The spherical harmonics $Y^{l,m}$ provide a basis set of functions for the irreducible representation of the group SO(3) with dimension $(2l+1)$.

The Wigner-D matrix can be constructed by spherical harmonics. Consider a transformation $Y^{l,m}(\vec{\mathbf{p}}) = Y^{l,m}(\mathbf{R}_{\alpha,\beta,\gamma}\vec{\mathbf{p}}_x)$, where $\vec{\mathbf{p}}_x$ denote the x-orientation. α, β, γ denotes the items of Euler angle. Therefore, $Y^{l,m}(\vec{\mathbf{p}})$ is invariant with respect to rotation angle γ. Based on this viewpoint, the Wigner-D matrix with shape $(2l+1) \times (2l+1)$ can be defined by:

$$\mathbf{D}^{(l,m)}(\mathbf{R}_{\alpha,\beta,\gamma}) = \sqrt{2l+1} Y^{(l,m)}(\vec{\mathbf{p}}). \tag{24}$$

In this case, the orientations are encoded in spherical harmonics and their Wigner-D matrices, which are utilized in our cross module.

A.2 Equivariant Operation

A.2.1 Equivariance of Clebsch-Gordan Tensor Product

The Clebsch-Gordan Tensor Product shows a strict equivariance for different group representations, which make the mixture representations transformed equivariant based on Wigner-D matrices. We use $\mathbf{D}^{l,m'm}$ to denote the element of Wigner-D matrix in the degree l. The Clebsch-Gordan coefficient satisfies:

$$\begin{aligned} &\textstyle\sum_{m_1',m_2'} \mathcal{C}^{(l_0,m_0)}_{(l_1,m_1')(l_2,m_2')} \mathbf{D}^{(l_1,m_1'm_1)}(g) \mathbf{D}^{(l_2,m_2'm_2)}(g) \\ &= \textstyle\sum_{m_0'} \mathbf{D}^{(l_0,m_0'm_0)}(g) \mathcal{C}^{(l_0,m_0')}_{(l_1,m_1)(l_2,m_2)} \end{aligned} \tag{25}$$

Therefore, the spherical harmonics can be combined equivariantly by CG Tensor Product:

$$
\begin{aligned}
&CG\left(\sum_{m_1'}\mathbf{D}^{(l_1,m_1m_1')}(g)Y^{(l_1,m_1')},\sum_{m_2'}\mathbf{D}^{(l_2,m_2m_2')}(g)Y^{(l_2,m_2')}\right)\\
=&\sum_{m_1,m_2}\mathcal{C}^{(l_0,m_0)}_{(l_1,m_1)(l_2,m_2)}\sum_{m_1'}\mathbf{D}^{(l_1,m_1m_1')}(g)Y^{(l_1,m_1')}\sum_{m_2'}\mathbf{D}^{(l_2,m_2m_2')}(g)Y^{(l_2,m_2')}\\
=&\sum_{m_0'}\mathbf{D}^{(l_0,m_0m_0')}(g)\sum_{m_1,m_2}\mathcal{C}^{(l_0m_0')}_{(l_1,m_1)(l_2,m_2)}Y^{(l_1,m_1')}Y^{(l_2,m_2')}\\
=&\sum_{m_0'}\mathbf{D}^{(l_0,m_0m_0')}(g)CG\left(Y^{(l_1,m_1')},Y^{(l_2,m_2')}\right).
\end{aligned}
\tag{26}
$$

Here, we omit the input argument of the spherical harmonics, which can represent any direction on the sphere. Equation (26) represents a relationship between scalar. If we transform the scalar to vector or matrix like what we do in Equation (3), Equation (26) is equal to

$$
(\mathbf{D}^{l_1}\mathbf{u}\otimes\mathbf{D}^{l_2}\mathbf{v})^{l_0}=\mathbf{D}^{l_0}(\mathbf{u}\otimes\mathbf{v})^{l_0}. \tag{27}
$$

The tensor CG product mixes two representations to a new representation under special rule $|l_1-l_2|\leq l\leq(l_1+l_2)$. For example, 1.two type-0 vectors will only generate a type-0 representations; 2.type-l_1 and type-l_2 can generate type-l_1+l_2 vector at most. Note that some widely-used products are related to tensor product: scalar product ($l_1=0$, $l_2=1$, $l=1$), dot product ($l_1=1$, $l_2=1$, $l=0$) and cross product ($l_1=1$, $l_2=1$, $l=1$). However, for each element with $l>0$, there are multi mathematical operation for the connection with weights. The relation between number of operations and degree is quadratic. Thus, as degree increases, the amount of computation increases significantly, making calculation of the CG tensor product slow for higher order irreps. This statement can be proven by the implementation of e3nn (o3.FullyConnectedTensorProduct).

A.2.2 Learnable Parameters in Tensor Product

Previous works utilize the e3nn library [28] to implement the corresponding tensor product. It is crucial to emphasize that the formulation of CG tensor product is devoid of any learnable parameters, as CG coefficients remain constant. In the context of e3nn, learnable parameters are introduced into each path, represented as $w(\mathbf{u}^{l_1}\otimes\mathbf{v}^{l_2})$. Importantly, these learnable parameters will not destory the equivariance of each path. However, they are limited in capturing directional information. In equivariant models, the original CG tensor product primarily captures directional information. We have previously mentioned our replacement of the CG tensor product with learnable modules. It is worth noting that our focus lies on the CG coefficients rather than the learnable parameters in the e3nn implementation.

A.2.3 Gate Activation and Normalization

Gate Activation. In equivariant models, the Gate activation combines two sets of group representations. The first set consists of scalar steerable vector ($l=0$), which are passed through standard activation functions such as sigmoid, ReLU and SiLU. The second set comprises higher-order steerable vector (($l>0$)), which are multiplied by an additional set of scalar steerable vector that are introduced solely for the purpose of the activation layer. These scalar steerable vector are also passed through activation functions.

Normalization. Normalization is a technique commonly used in neural networks to normalize the activations within each layer. It helps stabilize and accelerate the training process by reducing the internal covariate shift, which refers to the change in the distribution of layer inputs during training.

The normalization process involves computing the mean and variance across the channels. In equivariant normalization, the variance is computed using the root mean square value of the L2-norm of each type-l vector. Additionally, this normalization removes the mean term. The normalized activations are then passed through a learnable affine transformation without a learnable bias, which enables the network to adjust the mean and variance based on the specific task requirements.

A.3 Relationship Between Expressive Power and Equivariant Operations

In [13], Theorem 2 establishes the universality of equivariant networks based on the TFN structure:

Theorem. For all $n \in \mathbb{N}$, $\mathbf{l}_T \in {\mathbb{N}_+}^*$,

- 1. For $D \in \mathbb{N}_+$, every G-equivariant polynomial $p : \mathbb{R}^{3\times n} \to W_{\mathbf{l}_T}^n$ of degree D is in $F_{C(D),D}^{TFN}$.
- 2. Every continuous G-equivariant function can be approximated uniformly on compact sets by functions in $\cup_{D\in\mathbb{N}_+} F_{C(D),D}^{TFN}$.

Here, n represents the number of input points (or nodes), $\mathbf{l}_T$ represents the degree of the approximated G-equivariant function, C represents the number of channels, and D represents the degree of the TFN (Tensor Field Network) structure, which is equivalent to the term l used in our method. The TFN structure consists of two layers, including convolution and self-interaction. Self-interaction involves equivariant linear functions. The convolution operation calculates the CG tensor product between different steerable representations, which is a fundamental operation for transforming directional information. Most equivariant models based on group representations use a similar approach (CG tensor product) to capture directional features. Therefore, the theorem mentioned above also applies to building blocks based on CG tensor products, such as SEGNN [11] and Equiformer [12].

It is important to note that achieving an infinite degree in practice is not feasible. However, equivariant models based on group representations can enhance their expressive power by increasing the number of maximal degrees [13]. In their evaluation of expressive power, as presented in [14], the authors utilize the GWL (geometric Weisfeiler-Leman) graph isomorphism test. In Table 2 of their work, it is evident that equivariant models with a maximal degree denoted as L are incapable of distinguishing n-fold symmetric structures when n exceeds the value of L.

B Proofs and Details For Section 3

B.1 Proof for Expressiveness of EST

The proof for Theorem 1 is shown in the following.

Proof. Given two steerable vectors $\mathbf{u} \in \mathbb{V}_{0\to l_1}$ and $\mathbf{v} \in \mathbb{V}_{0\to l_2}$, and we define the CG tensor product result is $\mathbf{w} \in \mathbb{V}_{l_0}$, where $l_0 <= l_1 + l_2$. The spatial representations of $\mathbf{w}$ can be represented as:

$$\sum_{l_0,m_0} \mathbf{w}^{(l_0,m_0)} Y^{(l_0,m_0)} = \sum_{l_0,m_0} ({\mathbf{u}^{(0\to l_1)}}^T \mathbf{C}^{(l_0,m_0)}_{(0\to l_1),(0\to l_2)} \mathbf{v}^{(0\to l_2)}) Y^{(l_0,m_0)}, \tag{28}$$

where $\mathbf{C}^{(l_0,m_0)}_{(0\to l_1),(0\to l_2)}$ is a matrix including the whole CG coefficients corresponding to degree l_0 and order m_0. We temporarily ignore the input of the spherical harmonics. The spherical Transformer use the multiplication between spatial representations of $\mathbf{u}$ and $\mathbf{v}$:

$$\begin{aligned} &\sum_{l_1,m_1} \mathbf{u}^{(l_1,m_1)} Y^{(l_1,m_1)} \sum_{l_2,m_2} \mathbf{v}^{(l_2,m_2)} Y^{(l_2,m_2)} \\ =&\sum_{l_1,m_1,l_2,m_2} \mathbf{u}^{(l_1,m_1)} Y^{(l_1,m_1)} \mathbf{v}^{(l_2,m_2)} Y^{(l_2,m_2)} \\ =&{\mathbf{u}^{(0\to l_1)}}^T ({\mathbf{Y}^{(0\to l_1)}}^T \mathbf{Y}^{(0\to l_2)}) \mathbf{v}^{(0\to l_2)} \end{aligned} \tag{29}$$

Recall that CG coefficients are in fact the expansion coefficients of a product of two spherical harmonics in terms of a single spherical harmonic (see Equation (17)):

$$Y^{(l_1,m_1)} Y^{(l_2,m_2)} = \sum_{l_0,m_0} \sqrt{\frac{(2l_1+1)(2l_2+1)}{4\pi(2l_0+1)}} \mathcal{C}^{(0,0)}_{(l_1,m_1)(l_2,m_2)} \mathcal{C}^{(l_0,m_0)}_{(l_1,m_1)(l_2,m_2)} Y^{l_0,m_0} \tag{30}$$

Equation (29) can be transformed to

$$\begin{aligned}&\mathbf{u}^{(0\to l_1)^T}(\mathbf{Y}^{(0\to l_1)^T}\mathbf{Y}^{(0\to l_2)})\mathbf{v}^{(0\to l_2)}\\=&\sum_{l_0,m_0}\mathbf{u}^{(0\to l_1)^T}\mathbf{H}^{(l_0,m_0)}_{(0\to l_1),(0\to l_2)}\mathbf{v}^{(0\to l_2)}Y^{(l_0,m_0)},\end{aligned}\tag{31}$$

where $\mathbf{H}^{(l_0,m_0)}_{(l_1,m_1),(l_2,m_2)} = \sqrt{\frac{(2l_1+1)(2l_2+1)}{4\pi(2l_0+1)}}\mathcal{C}^{(0,0)}_{(l_1,m_1)(l_2,m_2)}\mathcal{C}^{(l_0,m_0)}_{(l_1,m_1)(l_2,m_2)}$. Compared to $\mathbf{C}^{(l_0,m_0)}_{(0\to l_1),(0\to l_2)}$, the term $\mathbf{H}^{(l_0,m_0)}_{(0\to l_1),(0\to l_2)}$ introduces an additional constant term that can be approximated by linear layers and the FFNs. Therefore, Equation (29) is equivalent to Equation (28). On the other hand, Equation (29) is inherently part of the Transformer architecture: when relationships between different orientations are masked and the $\mathbf{V}$ vectors are taken as constant vectors, the Transformer can naturally reduce to Equation (29). □

Through the above proof, we find that the spherical representation of the output from the CG tensor product is fundamentally a special case of EST. Furthermore, *EST can capture dependencies between different orientations, which is particularly beneficial for approximating higher degree spherical harmonic representations. For instance, at a direction (θ,φ), higher-degree spherical harmonics may involve contributions from other directions such as $(2\theta,\varphi),(\theta,2\varphi),(2\theta,2\varphi)$ (see $Y^{(2)}$ in Equation* (19)*). This property also enables EST to approximate equivariant features of higher degree than the input.*

B.2 Proof for Equivariance of EST

The proof for Theorem 2 is shown in the following.

Proof. Given a spatial representation $f(\vec{\mathbf{p}})$ transformed from a steerable representation $\mathbf{x}$, spherical attention can be represented as:

$$\tilde{f}(\vec{\mathbf{p}}_1)=\int_{\vec{\mathbf{p}}_2\in\Omega}a\big(f^Q(\vec{\mathbf{p}}_1),f^K(\vec{\mathbf{p}}_2)\big)f^V(\vec{\mathbf{p}}_2)d\vec{\mathbf{p}}_2,\tag{32}$$

where $a(\cdot)$ denotes the attention coefficients. Terms Q,K,V denote three independent linear transformations. When the origin representation $\mathbf{x}$ is rotation by Wigner-D matrix $\mathbf{D}^{-1}$, the representation $f(\vec{\mathbf{p}})$ is transformed to $f(\mathbf{R}\vec{\mathbf{p}})$, where $\mathbf{R}$ and $\mathbf{D}$ share the same transformation parameters (see Equation (23)). Therefore, the attention results are changed to

$$\begin{aligned}&\int_{\mathbf{R}\vec{\mathbf{p}}_2\in\Omega}a\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1),f^K(\mathbf{R}\vec{\mathbf{p}}_2)\big)f^V(\mathbf{R}\vec{\mathbf{p}}_2)d\mathbf{R}\vec{\mathbf{p}}_2\\=&\int_{\mathbf{R}\vec{\mathbf{p}}_2\in\Omega}\frac{\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\mathbf{R}\vec{\mathbf{p}}_2)\big)}{\int_{\mathbf{R}\vec{\mathbf{p}}_3\in\Omega}\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\mathbf{R}\vec{\mathbf{p}}_3)\big)d\mathbf{R}\vec{\mathbf{p}}_3}f^V(\mathbf{R}\vec{\mathbf{p}}_2)d\mathbf{R}\vec{\mathbf{p}}_2\\=&\int_{\mathbf{R}\vec{\mathbf{p}}_2\in\Omega}\frac{\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\mathbf{R}\vec{\mathbf{p}}_2)\big)}{\int_{\vec{\mathbf{p}}_3\in\Omega}\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\vec{\mathbf{p}}_3)\big)d\vec{\mathbf{p}}_3}f^V(\mathbf{R}\vec{\mathbf{p}}_2)d\mathbf{R}\vec{\mathbf{p}}_2\\=&\int_{\vec{\mathbf{p}}_2\in\Omega}\frac{\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\vec{\mathbf{p}}_2)\big)}{\int_{\vec{\mathbf{p}}_3\in\Omega}\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1)*f^K(\vec{\mathbf{p}}_3)\big)d\vec{\mathbf{p}}_3}f^V(\vec{\mathbf{p}}_2)d\vec{\mathbf{p}}_2\\=&\int_{\vec{\mathbf{p}}_2\in\Omega}a\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1),f^K(\vec{\mathbf{p}}_2)\big)f^V(\vec{\mathbf{p}}_2)d\vec{\mathbf{p}}_2\\=&\tilde{f}(\mathbf{R}\vec{\mathbf{p}}_1).\end{aligned}\tag{33}$$

Therefore, the output of spherical attention remains steerable after rotation. If we transform $\tilde{f}(\mathbf{R}\vec{\mathbf{p}}_1)$ into its steerable representation, denoted as $\tilde{\mathbf{x}}$, the following relationship holds:

$$\tilde{\mathbf{x}}=\mathbf{D}^{-1}\mathcal{F}^{-1}\big(\mathrm{SA}(\mathcal{F}(\mathbf{x}))\big)=\mathcal{F}^{-1}\big(\mathrm{SA}(\mathcal{F}(\mathbf{D}^{-1}\mathbf{x}))\big)\tag{34}$$

The equivariance of spherical attention holds. Moreover, the spherical FFNs is obviously equivariant because the function $\tilde{f}(\vec{\mathbf{p}}_1)=\mathrm{FFN}\big(f(\vec{\mathbf{p}}_1)\big)$ only focus on one orientation. Therefore, we can prove that the whole EST framework contained spherical attention and spherical FFN is equivariant. □

In the implementation, the continuous function must be represented by discrete sampling. The attention operation become

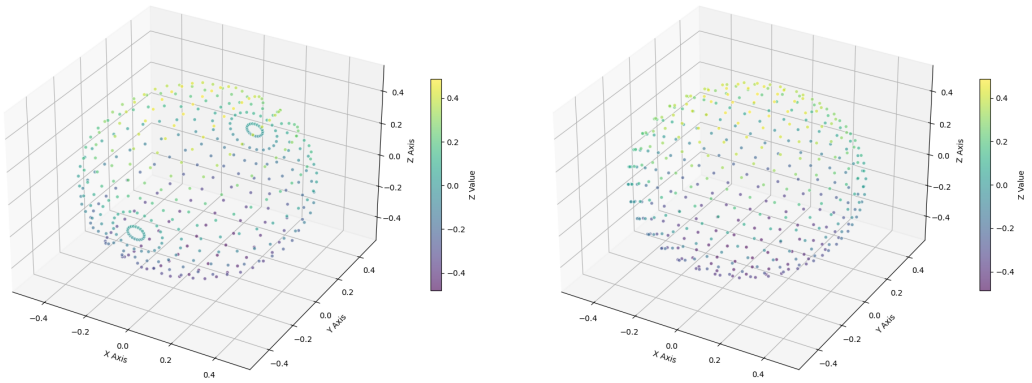
$$\tilde{f}(\vec{\mathbf{p}}_1) = \sum_{\vec{\mathbf{p}}_2} a\big(f^Q(\vec{\mathbf{p}}_1), f^K(\vec{\mathbf{p}}_2)\big) f^V(\vec{\mathbf{p}}_2). \tag{35}$$

After introducing random rotations, Equation (35) becomes

$$\tilde{f}(\mathbf{R}\vec{\mathbf{p}}_1) = \sum_{\vec{\mathbf{p}}_3} a\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1), f^K(\vec{\mathbf{p}}_3)\big) f^V(\vec{\mathbf{p}}_3). \tag{36}$$

Let us consider an ideal case where the point density is the same across all local regions. That is, the points on the sphere are perfectly uniformly distributed. In this scenario, the points in any region are symmetric, and we can find a set of $\vec{\mathbf{p}}_3$ that has a one-to-one correspondence with all $\mathbf{R}\vec{\mathbf{p}}_2$, i.e., the difference between the set of points $\vec{\mathbf{p}}_2$ and the set of points $\vec{\mathbf{p}}_3$ is only in their order. Therefore, spherical attention can still strictly preserve equivariance. In contrast, if we cannot find points that correspond one-to-one to all $\mathbf{R}\vec{\mathbf{p}}_2$, equivariance will be compromised. To reduce this equivariance loss, we can employ approximately uniform sampling methods, such as Fibonacci Lattices.

Sampling strategies Most previous works use the e3nn implementation for spherical Fourier transform. However, it significantly destroy the uniformity of spherical sampling, which is illustrated in Figure 3(a). In contrast, Fibonacci Lattices (FL) do not directly divide the polar angle and azimuth angle into a grid. Instead, they select sampling points on the sphere in a spiral pattern. As shown in Figure 3(b), FL tends to achieve more uniform sampling, thereby improving the equivariance of EST. This is also consistent with the results observed in Table 5.



(a) e3nn Sampling (b) Fibonacci Lattices Sampling

Figure 3: Two spherical sampling strategies.

B.3 Equivariance of Relative Orientation Embedding

The equivariance of relative orientation embedding can be proven with the same way in Equation (33), where $a\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1), f^K(\mathbf{R}\vec{\mathbf{p}}_2)\big)$ is transformed to

$$\frac{\exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1) * f^K(\mathbf{R}\vec{\mathbf{p}}_2) + (\mathbf{R}\vec{\mathbf{p}}_1)^T(\mathbf{R}\vec{\mathbf{p}}_2)\big)}{\int_{\mathbf{R}\vec{\mathbf{p}}_3 \in \Omega} \exp\big(f^Q(\mathbf{R}\vec{\mathbf{p}}_1) * f^K(\mathbf{R}\vec{\mathbf{p}}_3) + (\mathbf{R}\vec{\mathbf{p}}_1)^T(\mathbf{R}\vec{\mathbf{p}}_3)\big) d\mathbf{R}\vec{\mathbf{p}}_3}. \tag{37}$$

Due to the spherical symmetry, we can still eliminate the rotation $\mathbf{R}$ acting on $\mathbf{p}_2$ and $\mathbf{p}_3$. Therefore, after incorporating the Relative Orientation Embedding, EST retains its equivariance.

C Details of MoE

C.1 Mixture of Experts in Language Models

Recently, empirical evidence consistently demonstrates that increased model parameters and computational resources yield performance gains in language models when sufficient training data is available. However, scaling models to extreme sizes incurs prohibitive computational costs. The Mixture-of-Experts (MoE) architecture has emerged as a promising solution to this dilemma. By enabling parameter scaling while maintaining moderate computational requirements, MoE architectures have shown particular success when integrated with Transformer frameworks. These implementations have successfully scaled language models to substantial sizes while preserving performance advantages.

C.2 Mixture of Experts for Transformers

We begin with a standard Transformer language model architecture, which comprises Y stacked Transformer blocks:

$$\mathbf{u}^y = \text{Self-Att}\left(\mathbf{h}^{m-1}\right) + \mathbf{h}^{m-1}, \tag{38}$$

$$\mathbf{h}^y = \text{FFN}\left(\mathbf{u}^y\right) + \mathbf{u}^y, \tag{39}$$

where $\text{Self-Att}(\cdot)$ denotes the self-attention module, $\text{FFN}(\cdot)$ denotes the Feed-Forward Network (FFN), $\mathbf{u}^y$ are the hidden states of all tokens after the y-th attention module, and $\mathbf{h}^y \in \mathbb{R}^d$ is the output hidden state after the y-th Transformer block. layer normalization is omitted for brevity. The MoE architecture substitutes FFN layers in Transformers with MoE layers and each MoE layer comprises multiple structurally identical experts. Each token is dynamically assigned to several experts based on learned routing probabilities: If the y-th FFN is substituted with an MoE layer, the computation for its output hidden state $\mathbf{h}^y$ can be expressed as:

$$\mathbf{h}^y = \sum_{i=1}^{N} \left(g_i \, \text{FFN}_i\left(\mathbf{u}^y\right)\right) + \mathbf{u}^y, \tag{40}$$

$$g_i = \begin{cases} s_i, & s_i \in \text{Topk}(\{s_j | 1 \le j \le N\}, K), \\ 0, & \text{otherwise}, \end{cases} \tag{41}$$

$$s_i = \text{Softmax}_i\left(\left(\mathbf{u}^y\right)^T \mathbf{e}_i^y\right), \tag{42}$$

where N denotes the total number of experts, $\text{FFN}_i(\cdot)$ is the i-th expert FFN, g_i denotes the gate value for the i-th expert, s_i denotes the token-to-expert affinity, $\text{Topk}(\cdot, K)$ denotes the set comprising K highest affinity scores among those calculated for all N experts, and $\mathbf{e}_i^y$ is the learnable parameters representing the centroid of the i-th expert in the y-th layer. The sparsity property ($K \ll N$) ensures computational efficiency by restricting each token to interact with only K experts.

C.3 Mixture of Hybrid Experts in EST

Inspired by MoE of language models, we developed the mixture of hybrid experts for EST. Its computation can be expressed as:

$$\tilde{\mathbf{m}} = \sum_{i=1}^{N_{steerable}} \left(g_i \, \text{SteerableFFN}_i\left(\mathbf{m}\right)\right) + \sum_{j=1}^{N_{spherical}} \left(g_j \, \text{SphericalFFN}_i\left(\mathbf{m}\right)\right) + \mathbf{m}, \tag{43}$$

$$g_i = \begin{cases} s_i, & s_i \in \text{Topk}(\{s_k | 1 \le k \le N_{steerable}\}, K), \\ 0, & \text{otherwise}, \end{cases} \tag{44}$$

$$g_j = \begin{cases} s_j, & s_j \in \text{Topk}(\{s_k | 1 \le k \le N_{spherical}\}, K), \\ 0, & \text{otherwise}, \end{cases} \tag{45}$$

$$s_i, s_j = \text{split}\left(\text{Softmax}\left(\mathbf{m}^{(0)}\mathbf{W}\right)\right), \tag{46}$$

where $\mathbf{m}$ and $\tilde{\mathbf{m}}$ denote the input and output message, respectively, $\mathbf{m}^{(0)}$ denotes the invariant part of message, $\mathbf{W} \in \mathbb{R}^{C \times (N_{steerable} + N_{spherical})}$ represents the learnable expert centroids. Here, we omit the symbol of layer order for simplicity.

C.4 Ablation Studies

In terms of MoE, there are three key hyper-parameters, including expert count, routing sparsity (top-K), and load balancing. Our ablation studies begin with the easiest steerable MoE using the QM9 dataset and Equiformer baseline.

- Expert count: We assessed model performance across expert counts ranging from 1 to 64. Performance improved up to 10 experts, reaching optimal results at that point (Table 6). Meanwhile, excessive expert count leads to a significant increase in time cost while the improvement in performance may not necessarily continue. In contrast, setting the number of expert as 10 offers the right balance between performance improvement and time cost.
- Routing Sparsity: We evaluated different levels of sparsity by varying the K values in the top-K router, while increased sparsity (lower K values) degraded performance (Table 7).
- Load Balancing: The auxiliary balance loss was added to the training loss with varying ratios (0 to 0.1). However, introducing balance loss consistently reduced model performance (Table 8).

The discrepancy between our findings and typical language model observations regarding sparsity and balancing can be attributed to several factors. While sparsity and balancing mechanisms generally enhance efficiency in standard language models, the limited size of the QM9 dataset plays a crucial role in this divergence. Introducing more experts increases model complexity, which can lead to overfitting rather than improved generalization, particularly when training data is scarce. Furthermore, given our limited number of experts, the necessity for stringent sparsity enforcement and elaborate load balancing mechanisms is diminished. In such scenarios, employing a dense MoE configuration tends to yield optimal performance.

Table 6: Performance and time cost depending on expert count when predicting α on QM9.

Number of experts	1(w/o MoE)	2	4	8	10	16	32	64
Test MAE $\left(\mathrm{bohr}^3\right)$	0.0466	0.0461	0.0503	0.0437	**0.0424**	0.0449	0.0438	0.0443
Time per epoch (s)	373.69	391.93	416.53	472.80	500.71	576.01	893.20	1664.93

Table 7: Effect of routing sparsity for predicting α on QM9.

K	2	3	5	10 (Dense)
Test MAE $\left(\mathrm{bohr}^3\right)$	0.04338	0.04349	0.04472	**0.0424**

Table 8: Impact of balance loss ratio for predicting α on QM9.

Ratio	0 (w/o balance loss)	0.001	0.01	0.1
Test MAE $\left(\mathrm{bohr}^3\right)$	**0.0424**	0.04369	0.04369	0.04361

D Details of Experiments and Supplementary Experiments

D.1 Implementation Details of Baselines

In the S2EF experiment, the results of baselines in Table 1 follow [17], where each model is trained in official configuration. Most of these configurations can be found in Fairchem repository, and we also follow its code framework to construct our OC20 experiments. In the IS2RE experiment, the results in Table 2 of baselines follow [12] and [15]. In the QM9 experiment, the results in Table 3 of baselines follow [17] and [23].

Table 9: Hyper-parameters for the EST model setting on OC20 S2EF and OC20 IS2RE experiments.

Hyper-parameters	S2EF-ALL	IS2RE
Optimizer	AdamW	AdamW
Learning rate scheduling	Cosine learning rate with linear warmup	Cosine learning rate with linear warmup
Warmup epochs	0.01	2
Maximum learning rate	4×10^{-4}	2×10^{-4}
Batch size	256	32
Number of epochs	3	20
Weight decay	1×10^{-3}	1×10^{-3}
Dropout rate	0.1	0.2
Energy coefficient λ_E	4	1
Force coefficient λ_F	100	-
Gradient clipping norm threshold	100	100
Model EMA decay	0.999	0.999
Cutoff radius (Å)	12	5
Maximum number of neighbors	20	500
Number of radial bases	600	128
Dimension of hidden scalar features in radial functions	128	64
Maximum degree L_{max}	6	1
Maximum order M_{max}	2	1
Number of Layers	8	6
Node embedding dimension	128	$(256, l=0), (128, l=1)$
Intermediate dimension during the Fourier transform	128	256
Intermediate dimension and the number of steerable FFN	$128, 6$	$[(768, l=0), (384, l=1)], 5$
Intermediate dimension and the number of spherical FFN	$512, 4$	$512, 5$
Number of spherical point samples	128	128

D.2 Implementation Details of EST Experiments

S2EF and IS2RE In our experiments on OC20, we adopt two hybrid models: S2EF combines the message module from EquiformerV2 with the update module of EST, while IS2RE combines the message module of Equiformer with the update module of EST. To ensure a fair comparison, all training configurations are aligned with those of the original EquiformerV2 and Equiformer. The hyperparameters specific to the EST architecture include the number of spherical sampling points, the number and dimension of experts in the steerable FFN, and the number of experts in the spherical FFN. Detailed configurations for all models are summarized in Table 9. The experiment on S2EF is conducted on 32 NVIDIA A100 GPUs and the experiment on S2EF is conducted on 8 NVIDIA A100 GPUs.

Table 10: Hyper-parameters for QM9 dataset.

Hyper-parameters	EST	EST (with GA)
Optimizer	AdamW	AdamW
Learning rate scheduling	Cosine learning rate with linear warmup	Cosine learning rate with linear warmup
Warmup steps	5	5
Maximum learning rate	5×10^{-4}, 2×10^{-4}	5×10^{-4}, 1.5×10^{-4}
Batch size	$128, 64$	$128, 64$
Max training epochs	$350, 700$	$300, 600$
Weight decay	$5 \times 10^{-3}, 0$	$5 \times 10^{-3}, 0$
Dropout rate	$0.0, 0.2$	$0.0, 0.2$
Number of radial bases	128 for Gaussian radial basis 8 for radial bessel basis	
Cutoff radius (Å)	5	5
L_{max}	2	2
Number of layers	6	6
Node dimension	$(128, l=0), (64, l=1), (32, l=2)$	128
Spherical harmonics embedding dimension	$(1, l=0), (1, l=1), (1, l=2)$	$(1, l=0), (1, l=1), (1, l=2)$
Intermediate dimension during the Fourier transform	128	128
Intermediate dimension and the number of steerable FFN	$5, 128$	$6, (768, l=0), (384, l=1)$
Intermediate dimension and the number of spherical FFN	$5, 512$	$4, 512$
Number of spherical point samples	$200, 128$	128

QM9 In the QM9 experiments, we design two EST variants: 1) a fully EST-based architecture with both message and update blocks; 2) a hybrid model combining the message block of Equiformer and the update block of EST . To ensure a fair comparison with state-of-the-art methods (Equiformer and EquiformerV2), we adopt similar configurations as shown in Table 10. Note that Equiformer employs two different configurations for the following properties: $\alpha, \Delta\varepsilon, \varepsilon_{\text{HOMO}}, \varepsilon_{\text{LUMO}}, \mu, C_v, R^2$, ZPVE,

and G, H, U, U_0. We follow the same strategy in our experiments. Comparisons with another SOTA model, GotenNet [23], are conducted under different configurations, and the details are provided in Section D.3.

D.3 Comparison between GotenNet and EST

Comparing with GotenNet, a variant of EST was designed incorporating the GATA message block and substituting the original updating block with our proposed mixture of hybrid experts. The steerable part of the hybrid expert maintain the architecture of EQFF in GotenNet, while the EST updating module serves as the spherical part used in parallel. To maintain a lightweight design, we simply integrate one steerable expert with one spherical expert. Our model adopts the same number of layers as GotenNet but achieves better performance across 6 tasks (table 11).

To ensure a fair comparison, we use training configurations similar to those of the original GotenNet (see Table 12), including a batch size of 32 and a training schedule of 1000 epochs. *It is notable that several training hyper-parameters of EST differ from those used of GotenNet. For example, GotenNet specifies 10,000 warm-up steps, whereas our model sets this to 0. Additionally, we employed MAE as the loss function, while MSE was used for GotenNet instead.* Properties where EST performs poorly may stem from such subtle differences. For example, EST outperforms GotenNet on both ε_{HOMO} and ε_{LUMO}, but it underperforms significantly on the $\Delta\varepsilon$ property, which is the gap between ε_{HOMO} and ε_{LUMO}. We suspect that this might be due to the absence of a warmup stage, causing the model to converge to a suboptimal local minimum.

Table 11: Results on QM9 dataset for various properties.

Task Units	α $bohr^3$	$\Delta\varepsilon$ meV	ε_{HOMO} meV	ε_{LUMO} meV	μ D	C_v cal/(mol K)	G meV	H meV	R^2 $bohr^3$	U meV	U_0 meV	ZPVE meV
GotenNet (4 layers)	.033	**21.2**	16.9	13.9	.0075	.020	5.50	**3.70**	**0.029**	**3.67**	**3.71**	**1.09**
EST (4 layers with GATA)	**.030**	31.2	**16.7**	**13.3**	**.0070**	**.019**	**5.31**	4.02	**0.029**	3.96	3.94	1.16

Table 12: Hyper-parameters for the EST and GotenNet models on QM9 experiments. Red highlights the configurations that differ.

Hyper-parameters	EST (with GATA)	GotenNet
Loss function	MAE	MSE
Learning rate scheduling	reduce on plateau	linear warmup with reduce on plateau
Warmup steps	0	10000
Weight decay	0	0.01
Optimizer	AdamW	AdamW
Maximum learning rate	1×10^{-4}	1×10^{-4}
Batch size	32	32
Max training epochs	1000	1000
Dropout rate	0.1	0.1
Number of RBFs	64	64
Cutoff radius (Å)	5	5
L_{max}	2	2
Number of layers	4	4
Node dimension	256	256
Edge dimension	256	256
Number of attention heads	8	8
Intermediate dimension and the number of steerable FFN	$512, 1$	$512, 1$
Intermediate dimension and the number of spherical FFN	$512, 1$	–
Number of spherical point samples	64	–

D.4 Supplementary Experiments

Building Blocks We conducted ablation studies to validate the effectiveness of individual building blocks within EST. Specically, in Table 13, removing the steerable FFN means all experts are replaced by the spherical FFN, removing the spherical FFN means all experts are replaced by the steerable FFN and removing FL Sampling refers to using the Fourier transform from e3nn instead. We used

the prediction of the α property on QM9 as the core task for evaluation. As shown in Table 13, all components positively contribute to the overall performance. We draw several conclusions:

1. The **SA module with layer normalization** effectively improves overall performance.
2. **Mixing steerable and spherical experts** helps strike a balance between equivariance and expressive power, leading to better generalization performance.
3. **FL Sampling is crucial for EST**. Disrupting equivariance without it significantly harms the results on QM9.

Table 13: Ablation studies for modules in HDGNN.

Building blocks in EST					α MAE
LayerNorm	SA	Steerable FFN	Spherical FFN	FL Sampling	$bohr^3$
-	✓	✓	✓	✓	0.046
-	-	✓	✓	✓	0.044
✓	✓	-	✓	✓	0.045
-	-	✓	-	✓	0.042
✓	✓	✓	✓	-	0.053
✓	✓	✓	✓	✓	**0.041**