

---

# Tensor Decomposition Networks for Fast Machine Learning Interatomic Potential Computations

---

Yuchao Lin<sup>1,2\*</sup>, Cong Fu<sup>1\*</sup>, Zachary Krueger<sup>1</sup>, Haiyang Yu<sup>1</sup>, Maho Nakata<sup>3</sup>,  
Jianwen Xie<sup>2</sup>, Emine Kucukbenli<sup>4</sup>, Xiaofeng Qian<sup>5</sup>, Shuiwang Ji<sup>1,5,6†</sup>

<sup>1</sup>Department of Computer Science and Engineering, Texas A&M University, USA

<sup>2</sup>Lambda, Inc., USA

<sup>3</sup>RIKEN Cluster for Pioneering Research, RIKEN, Japan

<sup>4</sup>NVIDIA, USA

<sup>5</sup>Department of Materials Science and Engineering, Texas A&M University, USA

<sup>6</sup>Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, USA

## Abstract

SO(3)-equivariant networks are the dominant models for machine learning interatomic potentials (MLIPs). The key operation of such networks is the Clebsch-Gordan (CG) tensor product, which is computationally expensive. To accelerate the computation, we develop tensor decomposition networks (TDNs) as a class of approximately equivariant networks in which CG tensor products are replaced by low-rank tensor decompositions, such as the CANDECOMP/PARAFAC (CP) decomposition. With the CP decomposition, we prove (i) a uniform bound on the induced error of SO(3)-equivariance, and (ii) the universality of approximating any equivariant bilinear map. To further reduce the number of parameters, we propose path-weight sharing that ties all multiplicity-space weights across the  $\mathcal{O}(L^3)$  CG paths into a single shared parameter set without compromising equivariance, where  $L$  is the maximum angular degree. The resulting layer acts as a plug-and-play replacement for tensor products in existing networks, and the computational complexity of tensor products is reduced from  $\mathcal{O}(L^6)$  to  $\mathcal{O}(L^4)$ . We evaluate TDNs on PubChemQCR, a newly curated molecular relaxation dataset containing 105 million DFT-calculated snapshots. We also use existing datasets, including OC20, and OC22. Results show that TDNs achieve competitive performance with dramatic speedup in computations. Our code is publicly available as part of the AIRS library (<https://github.com/divelab/AIRS/>).

## 1 Introduction

Symmetry is a fundamental aspect of molecular and material systems [1], making it a crucial consideration in developing machine learning interatomic potentials (MLIPs). Equivariant graph neural networks have emerged as dominant frameworks in this domain. Usually, equivariant models incorporate directional features and spherical harmonics to maintain equivariance under rotation symmetry [2, 3, 4]. Among these, tensor product (TP) operations play a central role in fusing equivariant features, providing a powerful mechanism for building expressive models that adhere to SO(3)-equivariance [5, 6, 7, 8, 9].

However, the computational cost of tensor product operations grows rapidly with the maximum angular degree  $L$ , reaching  $\mathcal{O}(L^6)$  in conventional implementations. Recent efforts to mitigate this

---

\*Equal contribution

†Correspondence to: Shuiwang Ji <[sji@tamu.edu](mailto:sji@tamu.edu)>

cost have focused on accelerating the TP operation or applying frame averaging (FA) to enforce equivariance [10, 11, 12, 13]. While frame averaging is architecture-agnostic, it suffers from discontinuity issues [11]. On the other hand, TP acceleration techniques, such as SO(2)-based convolutions [14, 15] and fast spherical Fourier transformations [16], reduce complexity but are no longer the standard CG tensor product with the same expressivity. Thus, there remains a gap in developing a method that simultaneously reduces computational complexity and parameter count while maintaining similar accuracy and equivariance of CG tensor product.

In this work, we propose Tensor Decomposition Networks (TDNs), a new class of approximately equivariant networks that replace the standard CG tensor product with low-rank tensor decompositions based on CANDECOMP/PARAFAC (CP) decomposition. TDNs introduce a CP decomposition that provides error bounds on equivariance and universality, ensuring consistency under SO(3) transformations. Additionally, a path-weight sharing mechanism consolidates multiplicity-space weights across CG paths, significantly reducing the parameter count from  $\mathcal{O}(cL^3)$  to  $\mathcal{O}(c)$  with  $c$  the parameter count of weight of each path. The resulting layer has expressive power close to conventional TP while lowering computational complexity from  $\mathcal{O}(L^6)$  to  $\mathcal{O}(L^4)$ . We validate TDNs on a newly curated relaxation dataset with 105 million DFT-calculated molecular snapshots, along with the established OC20 and OC22 datasets, demonstrating competitive accuracy with substantial computational speedup.

## 2 Preliminaries and Related Work

Clebsch–Gordan (CG) tensor product is a fundamental operation widely used as the backbone for SO(3)-equivariant neural network, enabling the fusion of feature fields at different angular degrees. We discuss the formal definition of CG tensor product with the maximum angular degree  $L$ . Consider two feature fields  $\mathbf{x} = \bigoplus_{\ell_1=0}^L \mathbf{x}^{(\ell_1)}$  and  $\mathbf{y} = \bigoplus_{\ell_2=0}^L \mathbf{y}^{(\ell_2)}$ , where  $\mathbf{x}^{(\ell_1)}$ ,  $\mathbf{y}^{(\ell_2)}$  are type- $\ell_1$  and type- $\ell_2$  irreducible representations (irreps) of SO(3). The CG tensor product fuses a pair of SO(3)-representations  $\mathbf{x}^{(\ell_1)}$  and  $\mathbf{y}^{(\ell_2)}$  into every admissible output type  $|\ell_1 - \ell_2| \leq \ell_3 \leq \ell_1 + \ell_2$  via the CG coefficients  $C_{\ell_1, m_1, \ell_2, m_2}^{\ell_3, m_3}$ , defined as:

$$(\mathbf{x}^{(\ell_1)} \otimes_{CG} \mathbf{y}^{(\ell_2)})_{m_3}^{(\ell_3)} = \sum_{m_1=-\ell_1}^{\ell_1} \sum_{m_2=-\ell_2}^{\ell_2} C_{\ell_1, m_1, \ell_2, m_2}^{\ell_3, m_3} \mathbf{x}_{m_1}^{(\ell_1)} \mathbf{y}_{m_2}^{(\ell_2)}, \quad -\ell_3 \leq m_3 \leq \ell_3. \quad (1)$$

A triple  $(\ell_1, \ell_2, \ell_3)$  satisfying the CG selection rule  $|\ell_1 - \ell_2| \leq \ell_3 \leq \ell_1 + \ell_2$  is referred to as a *path*, and each path constitutes an independent SO(3)-equivariant mapping. Collecting the contributions from all admissible paths, the complete CG tensor product is expressed as:

$$\mathbf{x} \otimes_{CG} \mathbf{y} = \bigoplus_{\ell_1=0}^L \bigoplus_{\ell_2=0}^L \bigoplus_{\ell_3=|\ell_1-\ell_2|}^{\min(\ell_1+\ell_2, L)} (\mathbf{x}^{(\ell_1)} \otimes_{CG} \mathbf{y}^{(\ell_2)})_{m_3}^{(\ell_3)}.$$

However, the computational complexity of the CG tensor product scales as  $\mathcal{O}(L^6)$  as it involves  $\mathcal{O}(L^3)$  distinct paths and  $\mathcal{O}(L^3)$  operations per path. This significant cost poses a major bottleneck in practical implementations, especially when dealing with higher angular degrees. To mitigate this computational challenge, inspired by tensor decomposition [17] to low-rank tensors, we propose an efficient approximation based on tensor decomposition techniques, specifically the CP decomposition, to reduce the complexity while preserving approximate equivariance. The detailed formulation and implementation of the CP decomposition are presented in the following section.

**Invariant and Equivariant Models.** Symmetry has been a widely discussed constraint when developing machine learning methods for predicting chemical properties of molecules. Invariant and equivariant graph models have been widely used in these cases. Invariant models [18, 2, 19, 20, 21] aim to consider the rotation invariant features such as pairwise distance as the input, and make use of these to predict final properties. Equivariant models [3, 4, 22, 23, 24, 25] further incorporate equivariant features such as pairwise directions and spherical harmonics into the model. These models are built with equivariant blocks to ensure that output features rotate consistently with any rotation applied to the input features, thereby maintaining equivariant symmetry.

**Tensor Product Acceleration.** Among these equivariant networks, tensor product [5, 6, 26, 7, 9, 8, 27, 28] is one of the most important components that fuse two equivariant features into one. It provides a

powerful and expressive way [29] to build equivariant networks, while the computational cost of TP is usually considerable. Therefore, there are several directions to accelerate the equivariant networks. First direction is to accelerate TP. eSCN [14, 15, 30] proposes to reduce the  $SO(3)$  convolution into  $SO(2)$  for TP when one of the inputs of TP is spherical harmonics. Gaunt tensor product [16] makes use of fast spherical Fourier transformation to perform the TP. The other direction is to apply frame averaging (FA) [10, 11, 12], which uses group elements from an equivariant set-valued function called frame to transform the input data and subsequently the model’s output, enabling any models to obtain the desired symmetries. Although it is flexible and has no requirement for model architectures, it faces an unsolvable discontinuity problem [11].

### 3 Tensor Decomposition Networks

This section presents the techniques employed in the proposed Tensor Decomposition Network (TDN). In Section 3.1, we introduce a CP-decomposition-based approximation for the tensor product, and in Section 3.2, we detail a path-wise weight-sharing scheme. These strategies effectively reduce both computational cost and parameter count. In Section 3.3, we analyze the computational complexity of the approximate tensor product, and in Section 3.4, we discuss its error bound and universality.

#### 3.1 Tensor Product and Its Approximation

The tensor product is a fundamental operation in equivariant neural networks, enabling the coupling of features across multiple vector spaces. However, direct implementation of the tensor product incurs significant computational cost. To mitigate this, we introduce a low-rank approximation using the CANDECOMP/PARAFAC (CP) decomposition to reduce the time complexity.

**Tensor Product.** Before developing our tensor-product approximation we recall the canonical definition of the tensor product in the simplest non-trivial two-order case. The multi-order counterpart and its CP decomposition are introduced in Appendix A. In practice, higher-rank tensors are stored as flattened vectors via a fixed index ordering; this reshaping preserves the vector-space operations. Without loss of generality, we present the following tensor product definition. Let  $V_1 = \mathbb{R}^{d_1}$ ,  $V_2 = \mathbb{R}^{d_2}$ , and  $V_3 = \mathbb{R}^{d_3}$  be finite-dimensional real vector spaces with ordered bases  $\{e_i\}_{i=1}^{d_1}$ ,  $\{f_j\}_{j=1}^{d_2}$ ,  $\{g_k\}_{k=1}^{d_3}$ . The tensor product  $V_1 \otimes V_2$  is the space that corepresents bilinear maps: for every bilinear  $m: V_1 \times V_2 \rightarrow V_3$  there exists a unique linear map  $\tilde{m}: V_1 \otimes V_2 \rightarrow V_3$  such that  $m(\mathbf{x}, \mathbf{y}) = \tilde{m}(\mathbf{x} \otimes \mathbf{y})$ . With respect to the chosen bases,  $\tilde{m}$  is encoded by a three-way tensor

$$\mathbf{M} = (M_{kij}) \in V_3 \otimes V_1^* \otimes V_2^* \cong \mathbb{R}^{d_3 \times d_1 \times d_2},$$

and  $m(e_i, f_j) = \sum_{k=1}^{d_3} M_{kij} g_k$ . For arbitrary  $\mathbf{x} = \sum_{i=1}^{d_1} x_i e_i$  and  $\mathbf{y} = \sum_{j=1}^{d_2} y_j f_j$  we have

$$m(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{d_3} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} M_{kij} x_i y_j g_k = \mathbf{M}(\mathbf{x} \otimes \mathbf{y}). \quad (2)$$

**CP Decomposition for Tensor Product Approximation.** To reduce time complexity and storage cost of tensor product, we approximate the tensor  $\mathbf{M}$  via CP decomposition [17] of rank  $R$ . The CP decomposition writes the tensor product as sum of  $R$  rank-1 tensors by decomposing the three-way tensor  $\mathbf{M}$  such that

$$M_{kij} \approx \sum_{r=1}^R A_{kr} B_{ir} C_{jr}, \quad (3)$$

where  $\mathbf{A} \in \mathbb{R}^{d_3 \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{d_1 \times R}$ , and  $\mathbf{C} \in \mathbb{R}^{d_2 \times R}$  are factor matrices capturing the modes of the tensor. Substituting Eq. (3) into Eq. (2) gives

$$m(\mathbf{x}, \mathbf{y}) \approx \sum_{k=1}^{d_3} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \left( \sum_{r=1}^R A_{kr} B_{ir} C_{jr} \right) x_i y_j g_k.$$

Then we rearrange the summation to obtain

$$\sum_{r=1}^R \left( \sum_{i=1}^{d_1} B_{ir} x_i \right) \left( \sum_{j=1}^{d_2} C_{jr} y_j \right) \left( \sum_{k=1}^{d_3} A_{kr} g_k \right).$$

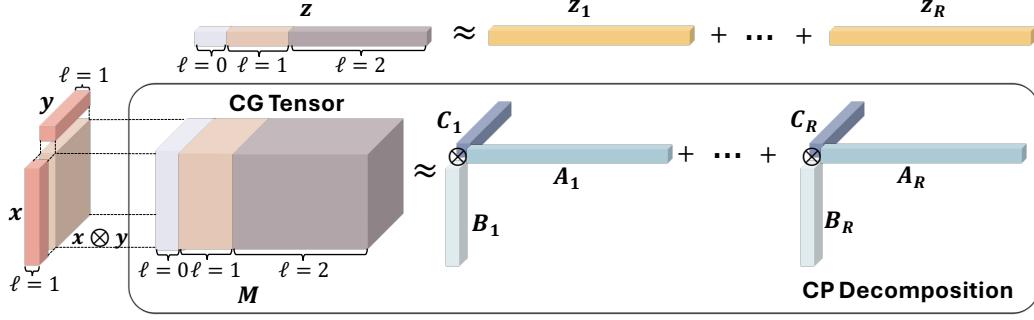


Figure 1: Illustration of approximating the CG tensor product using CP decomposition. The input features  $x$  and  $y$  both consist of irreps with  $\ell = 1$ , and the CG tensor product produces output feature  $z$  containing irreps with  $\ell = 0, 1, 2$ .

For clarity, the above approximation for the order-two tensor product  $m(x, y)$  can be expressed in matrix form as

$$m(x, y) \approx A(B^\top x \odot C^\top y), \quad (4)$$

where  $\odot$  denotes the Hadamard product. By the universal property that every bilinear map  $m: V_1 \times V_2 \rightarrow V_3$  factors uniquely as a linear map  $\tilde{m}: V_1 \otimes V_2 \rightarrow V_3$ , the tensor product covers important instances used in equivariant learning, notably the CG tensor product [6] and the Gaunt tensor product [16]. Therefore, the tensor product approximation can be employed for those specific cases. In this paper, we primarily discuss CG tensor product approximation.

**CP Decomposition for CG Tensor Product.** Next, we introduce the idea to make use of CP decomposition to accelerate the CG tensor product calculations. Specifically, the CG coefficient tensor is a three-way tensor concatenating CG coefficients  $C_{\ell_1, \ell_2}^{\ell_3}$  of all admissible path  $(\ell_1, \ell_2, \ell_3)$  such that

$$M = \bigoplus_{\ell_1=0}^L \bigoplus_{\ell_2=0}^L \bigoplus_{\ell_3=|\ell_1-\ell_2|}^{\min(\ell_1+\ell_2, L)} C_{\ell_1, \ell_2}^{\ell_3}.$$

The key idea of CP decomposition is to break down the CG coefficient tensor  $M$  into low-rank matrices. We demonstrate a simple example of CP decomposition for TP with  $\ell_1, \ell_2 = 1$  and  $\ell_3 \in \{0, 1, 2\}$ , as shown in Fig. 1. Eq. (4) allows for the simple batch-wise use of the Hadamard product with a global hyperparameter rank  $R$  instead of computing tensor product per path. The higher  $R$ , the more accurate the tensor product result is. For acceleration, we do not need full rank  $R$  and we can use a small  $R$  while keeping a low approximation error. We show the rank-error curve in Fig. 2.

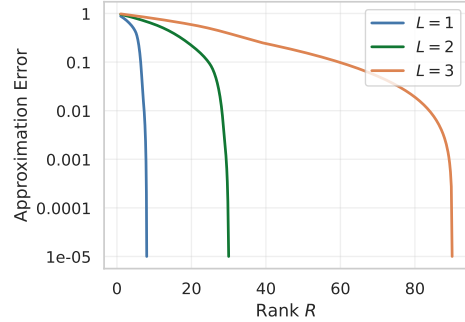


Figure 2: Approximation error for different rank values  $R$  across various maximum angular degrees  $L$ .

### 3.2 Path-Weight Sharing Tensor Product

In equivariant neural networks, the CG tensor product couples features that transform under irreducible representations (irreps) of  $SO(3)$ . For a maximal degree  $L$ , a fully connected CG tensor product introduces  $\mathcal{O}(L^3)$  paths, each associated with a distinct weight matrix. This leads to a substantial parameter count, which can hinder model efficiency. To mitigate this, we propose a path-weight sharing mechanism to reduce the parameter count while retaining equivariance.

**Concatenating Irreps into a Single Channel.** We first unify the multiplicities across all degrees  $\ell$  by setting a common multiplicity  $k$ . Letting  $x^{(\ell)} \in \mathbb{R}^{k \times (2\ell+1)}$  denote the multiplicity- $k$  irrep of degree  $\ell$ , we concatenate the irrep axes into a single channel:

$$\tilde{x} = \text{concat}_{\ell=0}^L x^{(\ell)} \in \mathbb{R}^{k \times (L+1)^2}.$$

All irreps now share a common multiplicity index within, so the linear projection and CG contraction over irreps features are executed as batched matrix operations whose operands reside contiguously in memory; this yields coalesced memory access on GPUs and improved cache locality on CPUs.

**Path-Weight Sharing Tensor Product.** We further reduce the parameter count by applying a path-weight sharing scheme. Let  $\mathbf{W}_{\ell_1, \ell_2}^{\ell_3}$  denote the multiplicity-space weight matrix associated with the path  $(\ell_1, \ell_2, \ell_3)$ , where  $0 \leq \ell_1, \ell_2, \ell_3 \leq L$  and  $c$  is the parameter count of weight of each path. Instead of assigning a unique matrix for each path, we set all such matrices equal to a single learnable parameter  $\mathbf{W}$ , i.e.  $\mathbf{W}_{\ell_1, \ell_2}^{\ell_3} \equiv \mathbf{W}$  for every admissible path  $(\ell_1, \ell_2, \ell_3)$ . This collapses the  $\mathcal{O}(L^3)$  distinct weight tensors of the naïve implementation into one, reducing the parameter count from  $\mathcal{O}(cL^3)$  to  $\mathcal{O}(c)$ . Because the sharing operates exclusively on multiplicity indices, the irrep content of each block and hence full  $\text{SO}(3)$  equivariance is retained. The resulting layer therefore retains the classic CG tensor product structure while offering an order-of-magnitude reduction in parameters. We also extend this scheme to equivariant linear layers and use it in our main experiments.

### 3.3 Complexity Analysis of Approximate Tensor Product

In this section, we analyze the computational complexity of the proposed approximate tensor product using CP decomposition. We first investigate the rank of the tensor product. Let  $\text{rank}_{\text{CP}}(\mathbf{M})$  denote the CP rank of the three-way tensor  $\mathbf{M} \in \mathbb{R}^{d_3 \times d_1 \times d_2}$ , representing the minimal rank  $R$  such that there is an equality for Eq. (3), i.e.

$$\text{rank}_{\text{CP}}(\mathbf{M}) = \min \left\{ R \in \mathbb{N}^+ \mid \mathbf{M}_{kij} = \sum_{r=1}^R \mathbf{A}_{kr} \mathbf{B}_{ir} \mathbf{C}_{jr} \right\}.$$

Determining  $\text{rank}_{\text{CP}}(\mathbf{M})$  exactly is NP-hard [17]. In practice one specifies a rank  $R$  as a hyper-parameter and optimizes the factor matrices  $\mathbf{A} \in \mathbb{R}^{d_3 \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{d_1 \times R}$ ,  $\mathbf{C} \in \mathbb{R}^{d_2 \times R}$  to minimize a prescribed loss. A generic upper bound  $\text{rank}_{\text{CP}}(\mathbf{M}) \leq \min\{d_1 d_2, d_1 d_3, d_2 d_3\}$  limits the choices of  $R$ . Consequently, algorithmic pipelines treat  $R$  as an external choice, balancing approximation accuracy against computational cost.

By using CP decomposition, the computational cost for evaluating the approximate tensor product in Eq. (4) reduces to  $\mathcal{O}(R(d_1 + d_2 + d_3))$ , a significant reduction compared to the full tensor contraction cost of  $\mathcal{O}(d_1 d_2 d_3)$  using Eq. (2). Similarly, the space complexity decreases from  $\mathcal{O}(d_1 d_2 d_3)$  to  $\mathcal{O}(R(d_1 + d_2 + d_3))$ , which provides substantial savings when  $R$  is small.

For the approximation of the CG tensor product, where  $d_1, d_2, d_3 \propto \mathcal{O}(L^2)$ , the computational complexity further reduces to  $\mathcal{O}(RL^2)$ . In our experiments, we select  $R = 7L^2$ . The error curve for the CP decomposition with varying  $R$  is discussed in Section 4.1.

### 3.4 Error Bound and Universality Analysis

The error bound and universality analysis provide theoretical guarantees for the CP decomposition of the tensor product in the proposed Tensor Decomposition Network (TDN). This section establishes the error bound for both the approximation and equivariance, demonstrating how the approximation error depends on the spectral tail of the tensor’s singular values. Additionally, we formalize the universality property of the CP decomposition, showing that as the rank  $R$  increases, the CP-decomposition-based tensor product can accurately approximate any  $\text{SO}(3)$ -equivariant bilinear map, thereby preserving the expressive power of the tensor product while reducing computational complexity.

**Error Bound of Approximate Tensor Product.** Given a rank  $R \leq \text{rank}_{\text{CP}}(\mathbf{M})$ , one seeks CP-decomposition-based approximation  $\widehat{\mathbf{M}}$  by minimizing  $\|\mathbf{M} - \widehat{\mathbf{M}}\|$  in a chosen norm, typically the Frobenius norm. Although the non-convex optimization may admit spurious local minima, the optimization of CP decomposition possesses an essentially unique rank- $R$  approximation under Kruskal’s condition [17], and modern alternating least-squares or gradient methods converge to it under mild coherence assumptions [31]. For error estimation in our setting, we specialize to the CG tensor product, where we let all irreps have the same maximum degree and all dimensions equal such that  $d = d_1 = d_2 = d_3$ . Given the singular values  $\sigma_k^{(n)}$  of the mode- $n$  matricization  $\mathbf{M}_{(n)}$  of  $\mathbf{M}$  and

truncating each  $M_{(n)}$  to rank  $R_T = \lfloor R^{1/3} \rfloor$ , a priori approximation error bound [32] gives

$$\|M - M_{\text{truncated}}\|_F \leq \left( \sum_{n=1}^3 \sum_{k > R_T} \sigma_k^{(n)2} \right)^{1/2}.$$

To quantify the loss of SO(3)-equivariance incurred by CP decomposition, let  $R \in \text{SO}(3)$  with representation the Wigner D-matrix  $D(R)$ , and the SO(3)-equivariance error is estimated by

$$\varepsilon(R, x, y) = \|\widehat{M}(D(R)x \otimes D(R)y) - D(R)\widehat{M}(x \otimes y)\|.$$

The following result bounds this error uniformly over SO(3) with the proof in Appendix B.1.

**Theorem 3.1 (Equivariance Error Bound of CP Decomposition):** *Let CG tensor  $M \in \mathbb{R}^{d \times d \times d}$  and  $\widehat{M}$  be the rank- $R$  CP-decomposition-based approximation obtained by Frobenius minimization. For any rotation  $R \in \text{SO}(3)$  and any bounded representations  $x, y \in \mathbb{R}^d$ ,  $\|x\|, \|y\| \leq C$ , we have*

$$\varepsilon(R, x, y) \leq 2C^2 \left( \sum_{n=1}^3 \sum_{k > R_T} \sigma_k^{(n)2} \right)^{1/2},$$

where  $R_T = \lfloor R^{1/3} \rfloor$  and  $\sigma_k^{(n)}$  is the  $k$ -th singular value of mode- $n$  matricization of  $M$ .

Empirical estimates of both the approximation and equivariance errors are provided in Section 4.1.

**Universality of Approximate Tensor Product.** Because the tensor product is universal for bilinear maps, any SO(3)-equivariant bilinear operator can be expressed as a composition of a tensor product followed by a suitable  $M$  onto an equivariant subspace. Consequently, the CP-decomposition-based approximation developed above inherits this universality: for every equivariant tensor product there exists a rank- $R$  approximation that converges to it as  $R \rightarrow \text{rank}_{\text{CP}}(M)$ . The following theorem formalizes the expressivity of our approximation scheme with the proof in Appendix B.2.

**Theorem 3.2 (Universality of CP Decomposition):** *Consider SO(3)-representations  $x \in V_1 \cong \mathbb{R}^{d_1}$ ,  $y \in V_2 \cong \mathbb{R}^{d_2}$  and co-domain  $V_3 \cong \mathbb{R}^{d_3}$ . For any SO(3)-equivariant bilinear map  $\Phi$ , there exist  $B \in \mathbb{R}^{d_1 \times R}$ ,  $C \in \mathbb{R}^{d_2 \times R}$ ,  $A \in \mathbb{R}^{d_3 \times R}$  such that  $\Phi$  can be written as*

$$\Phi(x, y) = A(B^\top x \odot C^\top y) \in V_3,$$

with  $R \leq d_1 d_2$ .

## 4 Experiments

The effectiveness of our method is assessed on two benchmarks: the PubChemQCR dataset, which contains millions of molecular relaxation snapshots, and the established Open Catalyst Project (OCP) datasets. Section 4.1 describes both benchmarks and model configurations. Section 4.2 compares our model with several baselines on PubChemQCR and its subset PubChemQCR-S, while Section 4.3 and Section 4.4 report results of our model compared to several tensor-product baselines on OC20 and OC22 from OCP, respectively. Section 4.5 then analyses the computational efficiency of our approach as the maximum angular degree in the tensor product is varied. Section 4.6 further reports ablations of the proposed components across our model and a second widely used architecture.

### 4.1 Experimental Setup

**Datasets.** We use a newly curated dataset PubChemQCR [36], comprising high-fidelity molecular trajectories derived from the PubChemQC database [37]. This dataset encompasses a diverse range of molecular systems, capturing potential energy surfaces and force information critical for understanding molecular interactions. The full dataset consists of 3,471,000 trajectories and 105,494,671 DFT-calculated molecular snapshots, with each snapshot containing molecular structure, total energy, and forces. For training efficiency, we use the smaller subset, PubChemQCR-S, comprising 40,979 trajectories and 1,504,431 molecular snapshots for model benchmarking. The

Table 1: Comparison of model performance on energy and force predictions for the PubChemQCR and the PubChemQCR-S dataset. Our model is trained to compare against several baseline methods on the PubChemQCR-S dataset, including SchNet [18], PaiNN [4], MACE [8], PACE [33], FAENet [10], NequIP [9], SevenNet [34], Allegro [35], and Equiformer [7]. On the full PubChemQCR dataset, we compare our model with SchNet [18] and PaiNN [4]. The best results are shown in **bold**.

DATASET	MODEL	VALIDATION		TEST	
		ENERGY MAE (meV/atom) ↓	FORCE RMSE (meV/Å) ↓	ENERGY MAE (meV/atom) ↓	FORCE RMSE (meV/Å) ↓
SMALL SUBSET	SCHNET	5.30	56.55	5.55	56.22
	PAINN	5.13	46.34	5.33	46.92
	NEQUIP	7.37	54.78	8.27	55.59
	SEVENNET	8.77	47.63	10.21	48.05
	ALLEGRO	10.86	60.71	10.80	61.44
	FAENET	7.28	60.24	8.70	60.51
	MACE	7.54	51.46	7.47	45.70
	PACE	6.24	50.54	6.53	51.43
	EQUIFORMER	4.69	34.67	5.38	35.11
	TDN	<b>4.46</b>	<b>26.94</b>	<b>5.01</b>	<b>26.43</b>
FULL	SCHNET	7.14	65.22	7.71	67.38
	PAINN	3.62	38.30	3.49	39.28
	TDN	<b>1.65</b>	<b>19.46</b>	<b>1.50</b>	<b>20.44</b>

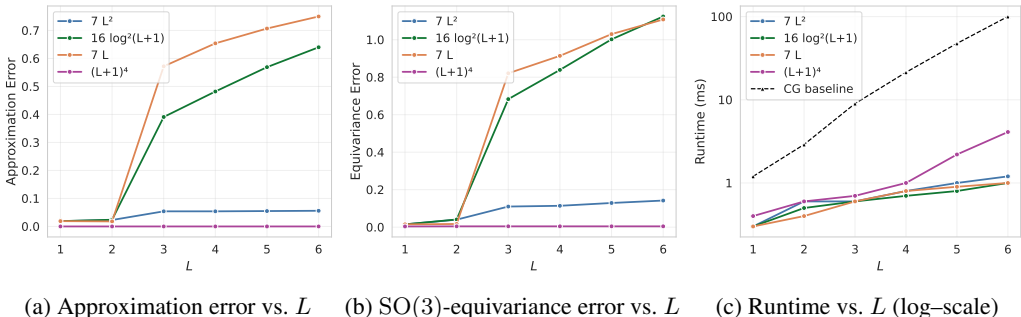


Figure 3: Scaling behaviours of the CP-decomposition-based tensor product under different maximum angular degree schedules: (a) approximation error, (b) SO(3)-equivariance error, and (c) runtime. The error and runtime of the CP decomposition-based tensor product depend on the chosen rank, multiplicities, and the maximum angular degree, and are not tied to a specific dataset.

subset is split into training, validation, and testing sets using a 60%-20%-20% ratio, while the full dataset employs an 80%-10%-10% split to assess generalizability. To prevent data leakage, each trajectory is assigned to only one split in both PubChemQCR and PubChemQCR-S.

The second dataset used in this study is the Open Catalyst Project (OCP) [38], which publishes extensive open datasets of DFT relaxations for adsorbate-catalyst surfaces and hosts public leaderboard challenges. The flagship releases, OC20 [39] and OC22 [40], encompass thousands of chemical compositions, crystal facets, and adsorbates, serving as comprehensive benchmarks for surrogate models aiming to replace computationally intensive calculations. Each release defines multiple tasks, including *Initial-Structure-to-Relaxed-Energy* (IS2RE), which require accurate predictions of total energies, per-atom forces, and relaxed geometries. We conduct experiments on OC20 IS2RE and OC22 IS2RE, using the data splits and configurations specified in the official OCP repository.

**CP Rank Selection.** We approximate the fully connected CG tensor product using the rank- $R$  CP decomposition across various configurations of the maximum angular degree  $L$ . Four rank schedules are evaluated:  $R = (L + 1)^4$ ,  $R = 16\log^2(L + 1)$ ,  $R = 7L$ , and  $R = 7L^2$ . For each configuration we measure (i) **Approximation Error**: The relative CP tensor product error, calculated as  $\|M(x \otimes y) - \widehat{M}(x \otimes y)\|_F / \|M(x \otimes y)\|_F$ ; (ii) **Equivariance Error**: The expected SO(3)-equivariance error, defined as  $\mathbb{E}_{R,x,y}[\varepsilon(R, x, y)]$ , where the expectation is averaged over

1000 random rotations  $\mathbf{R}$  and random vectors  $\mathbf{x}, \mathbf{y}$ ; and (iii) **Execution Time:** The runtime for the CP-decomposition-based tensor product under each rank schedule. Among the tested schedules, the quadratic schedule  $R = 7L^2$  consistently achieves the best accuracy-efficiency trade-off, while also requiring significantly less computation time compared to the CG tensor product baseline. Thus, this schedule provides the best balance between accuracy and computational cost and is adopted for all subsequent experiments. Further rank ablations are described in Appendix D.

**Model Design.** Building on the capabilities of graph transformers, we develop the *Tensor-Decomposition Network* (TDN) by modifying the publicly available Equiformer architecture [7]. Specifically, we replace every channel-wise linear projection, normalization layer, and activation in Equiformer with batched matrix operations that act simultaneously on the multiplicity dimension of each irrep block, eliminating the need for costly slicing and reshaping between tensor and vector representations. Additionally, the depth-wise tensor product mechanism is removed and the core CG tensor products within the self-attention mechanism are substituted with our rank- $R$  CP-decomposition-based tensor product from Section 3.1, integrated with the path-weight-sharing scheme from Section 3.2. This design preserves the expressive power of CG tensor product while significantly improving memory and computational efficiency. Note that for each maximum degree  $L$ , we precompute a rank- $R$  CP decomposition of the CG coefficient tensor once and cache the factors. These factors are treated as constants during all training runs. We further integrate the equivariant linear layer with the path-weight-sharing scheme. For the OCP dataset, the same model architecture is applied and a subset of experiments additionally incorporate initial node embeddings following [25]. The detailed model configurations for TDN are described in Appendix C.

**Baseline Implementations.** For the PubChemQCR benchmark, we reimplement each baseline model based on its official repository. Hyperparameters are adopted from the best configurations reported in the original papers or, if unspecified, are tuned using the PubChemQCR-S dataset. All model configurations of baseline models are described in Appendix C.

## 4.2 Results on PubChemQCR

We evaluate our method on both PubChemQCR-S and the full PubChemQCR dataset. For the small split we compare against nine state-of-the-art models: SchNet [18], PaiNN [4], MACE [8], PACE [33], FAENet [10], NequIP [9], SevenNet [34], Allegro [35], and Equiformer [7]. On the full PubChemQCR dataset, we benchmark against SchNet and PaiNN, the only baselines that scale to its size within our hardware budget. Performance is reported as mean absolute error (MAE) for energies and root-mean-square error (RMSE) for forces over the validation and testing splits. Note that all results are selected based on the lowest validation energy error. The results are shown in Table 1, the proposed TDN model achieves the lowest energy and force prediction errors across PubChemQCR-S and the full PubChemQCR dataset, outperforming all baseline methods. In addition, the performance of TDN improves further as the size of the dataset increases.

**Training Setup.** Across both the PubChemQCR and PubChemQCR-S benchmarks, we adopt a uniform training protocol: a cutoff radius of 4.5 Å; the Adam optimizer with an initial learning rate of  $5 \times 10^{-4}$ ; and a REDUCELRONPLATEAU scheduler with a patience of 2 epochs. All models are trained for up to 100 epochs on PubChemQCR-S and up to 15 epochs on the full PubChemQCR dataset. Unless otherwise noted, experiments are executed on NVIDIA A100-80GB GPUs.

## 4.3 Results on OC20

Table 2 summarizes our performance on the principal OC20 task IS2RE-DIRECT, which predicts the relaxed adsorption energy directly from the initial geometry (no noisy-node auxiliary loss). We benchmark against the widely-used baselines reported to date, including SchNet [18], DimeNet++ [41], GemNet-dT [19], SphereNet [20], Equiformer [7] and EquiformerV2 [15]. Metrics follow the official OC20 protocol: energy mean absolute error (MAE, eV) and energy within threshold (EwT, %) in IS2RE-DIRECT for four validation sub-splits: distribution adsorbates and catalysts (ID), out-of-distribution adsorbates (OOD-Ads), out-of-distribution catalysts (OOD-Cat), and out-of-distribution adsorbates and catalysts (OOD-Both). The IS2RE-DIRECT results are presented in Table 2, demonstrating that our model achieves performance comparable to Equiformer while being more efficient,



Table 2: Comparison of model performance on energy predictions for OC20 IS2RE-DIRECT validation set without noisy-node auxiliary loss. Our model is trained to compare against several baseline methods, including SchNet [18], DimeNet++ [41], GemNet-dT [19], SphereNet [20], Equiformer [7] and EquiformerV2 [15]. The best results are shown in **bold** and the second best results are shown with underlines.

MODEL	ENERGY MAE (eV) ↓					EWT (%) ↑				
	ID	OOD ADS	OOD CAT	OOD BOTH	AVERAGE	ID	OOD ADS	OOD CAT	OOD BOTH	AVERAGE
SCHNET	0.6465	0.7074	0.6475	0.6626	0.6660	2.96	2.22	3.03	2.38	2.65
DIMENET++	0.5636	0.7127	0.5612	0.6492	0.6217	4.25	2.48	4.40	2.56	3.42
GEMNET-DT	0.5561	0.7342	0.5659	0.6964	0.6382	4.51	2.24	4.37	2.38	3.38
SPHERENET	0.5632	0.6682	0.5590	0.6190	0.6024	4.56	<u>2.70</u>	4.59	2.70	3.64
EQUIFORMER	<u>0.5088</u>	<b>0.6271</b>	<b>0.5051</b>	<b>0.5545</b>	<b>0.5489</b>	<u>4.88</u>	<b>2.93</b>	<u>4.92</u>	<b>2.98</b>	<u>3.93</u>
EQUIFORMERV2	0.5161	0.7041	0.5245	0.6365	0.5953	-	-	-	-	-
TDN	<b>0.5085</b>	<u>0.6668</u>	<u>0.5104</u>	<u>0.5875</u>	<u>0.5683</u>	<b>5.21</b>	2.54	<b>5.04</b>	<b>2.98</b>	<b>3.94</b>

as detailed in Section 4.5. This highlights the effectiveness of our architecture in maintaining accuracy while significantly reducing computational costs.

**Training Setup.** For the IS2RE-DIRECT task, we follow Equiformer’s optimization setup [7] by a maximum angular degree  $L = 1$ , an AdamW optimizer with a learning rate of  $2 \times 10^{-4}$  and a weight decay of  $10^{-3}$ , a batch size of 32, and a cosine-decay learning-rate schedule. A warm-up is employed for 2 epochs on IS2RE-DIRECT, with a warm-up factor of 0.2; the cosine decay then runs over 30 training epochs. IS2RE-DIRECT experiments are run on a single NVIDIA RTX A6000-48GB GPU.

#### 4.4 Results on OC22

Following the OC20 evaluation protocol, we evaluate on the OC22 IS2RE-DIRECT test set, which predicts relaxed energy directly from the initial structure and omits the noisy-node auxiliary loss. We benchmark against strong baselines up to date as listed in Table 3 and report mean absolute error (MAE, eV) for the in-distribution (ID) and out-of-distribution (OOD) splits by averaging across the four standard OC22 sub-splits using the same split scheme as OC20 IS2RE-DIRECT task. As summarized in Table 3, TDN achieves the lowest ID MAE and the best overall average MAE, and it attains the second-best OOD MAE, validating the high effectiveness of TDN.

Table 3: Comparison of model performance on energy predictions for OC22 IS2RE-DIRECT testing set. We compare with several baseline methods, including SchNet [18], DimeNet++ [41], PaiNN [4], GemNet-dT [19], and coGN [42]. The best results are shown in **bold** and the second best results are shown with underlines.

MODEL	MAE (ID) (eV)	MAE (OOD) (eV)	AVERAGE (eV)
SCHNET	2.00	4.85	3.42
DIMENET++	1.96	3.52	2.74
PAIINN	1.72	3.68	2.70
GEMNET-DT	1.68	3.08	2.38
coGN	<u>1.62</u>	<b>2.81</b>	<u>2.21</u>
TDN	<b>1.49</b>	<u>2.92</u>	<b>2.20</b>

**Training Setup.** On OC22 IS2RE, we use the same optimization setup as OC20 by a maximum angular degree  $L = 1$ , an AdamW optimizer with a learning rate of  $2 \times 10^{-4}$ , a weight decay of  $10^{-3}$ , a batch size of 32, and a cosine learning-rate decay with a 2-epoch warm-up for 1000 epochs on a single NVIDIA A100-80GB GPU.

#### 4.5 Efficiency of CP-Decomposition-Based Tensor Product

To evaluate the speed-up achieved by our CP-decomposition-based tensor product in Section 3.1, we benchmark its inference runtime against the fully connected CG tensor product implementation in `e3nn`. As shown in Fig. 3c, the proposed approximation accelerates by factors of  $4.0\times$ ,  $4.8\times$ ,  $15.0\times$ ,  $26.7\times$ ,  $47.6\times$ , and  $83.6\times$  for maximum degrees  $L = 1, 2, 3, 4, 5$ , and  $6$ , respectively.

Table 4: Throughput and parameter count comparison between TDN and Equiformer across different values of maximum degree  $L$ .

MODEL	THROUGHPUT (samples/sec)	PARAMETERS
EQUIFORMER (L=1)	311.7	12.1M
TDN (L=1)	770.8 ( $\times$ <b>2.47</b> )	4.5M ( $\times$ <b>0.37</b> )
EQUIFORMER (L=2)	71.9	27.9M
TDN (L=2)	312.4 ( $\times$ <b>4.34</b> )	4.5M ( $\times$ <b>0.16</b> )
EQUIFORMER (L=3)	26.1	54.7M
TDN (L=3)	220.4 ( $\times$ <b>8.44</b> )	4.5M ( $\times$ <b>0.08</b> )

Since TDN is derived from Equiformer by replacing each CG block with CP decomposition and incorporating path-weight sharing, and removing depth-wise tensor product mechanism, we further benchmark end-to-end throughput and parameter count for both networks on a single NVIDIA A100-80GB GPU and Xeon Gold 6258R processor with a batch size of 128; detailed model configurations are provided in Appendix D. The results in Table 4 show that TDN processes  $2.47\times$  to  $8.44\times$  more structures per second and uses 63%–92% fewer parameters than Equiformer. As the maximum degree  $L$  increases, TDN achieves higher throughput and requires fewer parameters. A comprehensive time ablation of CP decomposition and path-weight sharing mechanism is provided in Table 7.

#### 4.6 Efficiency Ablation of CP-Decomposition-Based Tensor Product

To further evaluate the efficiency contributions of our proposed components, we perform ablations on the PubChemQCR-S dataset by removing the path-weight sharing tensor product and the CP decomposition of TDN while holding all other components fixed. As shown in Section 4.6, starting from TDN, enabling both mechanisms reduces training time by roughly 78%. TDN is trained with a maximum degree  $L = 2$ , four graph transformer layers, a hidden dimension of 64. We also evaluate NequIP [9]

by augmenting the base model with path-weight sharing tensor product and CP decomposition. This cuts computational cost nearly 74%. NequIP is trained with a maximum degree  $L = 2$ , four interaction blocks, and multiplicity 64. All experiments are conducted on a single NVIDIA RTX A6000-48GB GPU. A full table of ablation studies including accuracy results is presented in Table 9.

Table 5: Efficiency ablation of the path-weight sharing tensor product (PS) and CP-decomposition-based tensor product (CP). Results are shown for TDN and NequIP [9] on PubChemQCR-S dataset.

MODEL	TRAINING TIME (min/epoch)
TDN w/o CP + PS	19.0
TDN	<b>4.2</b> ( $\times 0.22$ )
NEQUIP	7.5
NEQUIP + CP + PS	<b>2.0</b> ( $\times 0.26$ )

## 5 Limitations

Our approach accelerates  $SO(3)$ -equivariant tensor products by replacing the full CG tensor product with a low-rank CP decomposition. We substantiate both its theoretical speed-up and its empirical efficacy across multiple benchmarks. One limitation of our approach is rank selection: the minimal CP rank needed to attain a given approximation error is unknown in general, and computing it exactly is NP-hard [17]. We therefore employ an empirical rank scheduler; however, this scheduler is tailored to the CG tensor and must be re-derived when extending to other equivariant tensor products. A second limitation is that path-weight sharing mechanism, while reducing parameters and memory, can introduce a small accuracy drop. In future work, we will (i) study broader families of group-equivariant tensor products, e.g.,  $SO(2)$  convolution [14], Gaunt tensor products [16], or more complex tensor products [43], to characterize their optimal rank profiles and develop general rank-selection criteria and adaptive rank selector, and (ii) design adaptive path-selection and grouping strategies that preserve the efficiency benefits of weight sharing while recovering performance.

## 6 Summary

In this work, we present Tensor Decomposition Networks (TDNs), a novel framework designed to accelerate the computationally intensive Clebsch-Gordan (CG) tensor product in  $SO(3)$ -equivariant networks through low-rank tensor decomposition. By leveraging CANDECOMP/PARAFAC (CP) decomposition and implementing path-weight sharing mechanism, TDNs effectively reduce both parameter count and computational complexity while preserving the expressive power of conventional CG tensor products. We also analyze time complexity, derive approximation error bounds, and establish universality of our approach, providing theoretical guarantees. Extensive evaluations on a newly curated PubChemQCR dataset and commonly used OC20 and OC22 benchmarks demonstrate that TDNs achieve comparable predictive accuracy to state-of-the-art models while significantly reducing runtime. The proposed framework provides a plug-and-play alternative to conventional CG tensor products, making it a promising approach for large-scale molecular simulations.

## Acknowledgments

SJ acknowledges support from ARPA-H under grant 1AY1AX000053, National Institutes of Health under grant U01AG070112, and National Science Foundation under grant IIS-2243850. XQ acknowledges support from the Air Force Office of Scientific Research (AFOSR) under grant FA9550-24-1-0207. We acknowledge Lambda, Inc. and NVIDIA for providing the computational resources for this project.

## References

- [1] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, ..., and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025.
- [2] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.
- [3] Victor Garcia Satorras, Emiel Hoogetboom, and Max Welling. E(n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [4] 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.
- [5] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.
- [6] 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.
- [7] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2022.
- [8] 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*, 2022.
- [9] 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):2453, 2022.
- [10] Alexandre Agm Duval, Victor Schmidt, Alex Hernandez-Garcia, Santiago Miret, Fragkiskos D Malliaros, Yoshua Bengio, and David Rolnick. Faenet: Frame averaging equivariant gnn for materials modeling. In *International Conference on Machine Learning*, pages 9013–9033. PMLR, 2023.
- [11] Nadav Dym, Hannah Lawrence, and Jonathan W Siegel. Equivariant frames and the impossibility of continuous canonicalization. *arXiv preprint arXiv:2402.16077*, 2024.
- [12] Yuchao Lin, Jacob Helwig, Shurui Gui, and Shuiwang Ji. Equivariance via minimal frame averaging for more symmetries and efficiency. In *Proceedings of the 41st International Conference on Machine Learning*, pages 30042–30079, 2024.
- [13] Behrooz Tahmasebi and Stefanie Jegelka. Generalization bounds for canonicalization: A comparative study with group averaging. In *The Thirteenth International Conference on Learning Representations*.
- [14] Saro Passaro and C Lawrence Zitnick. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. *arXiv preprint arXiv:2302.03655*, 2023.

- [15] 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.
- [16] 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*, 2024.
- [17] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [18] 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), 2018.
- [19] 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.
- [20] 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.
- [21] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3D molecular graphs. In *The 36th Annual Conference on Neural Information Processing Systems*, pages 650–664, 2022.
- [22] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.
- [23] 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*, 2021.
- [24] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022.
- [25] Eric Qu and Aditi S. Krishnapriyan. The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical domains. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [26] 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.
- [27] Oliver Unke, Mihail Bogojeski, Michael Gastegger, Mario Geiger, Tess Smidt, and Klaus-Robert Müller. Se (3)-equivariant prediction of molecular wavefunctions and electronic densities. *Advances in Neural Information Processing Systems*, 34:14434–14447, 2021.
- [28] Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and equivariant graph networks for predicting quantum Hamiltonian. In *Proceedings of the 40th International Conference on Machine Learning*, pages 40412–40424, 2023.
- [29] Nadav Dym and Haggai Maron. On the universality of rotation equivariant point cloud networks. *arXiv preprint arXiv:2010.02449*, 2020.
- [30] Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S Levine, Meng Gao, Misko Dzamba, and C Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. *arXiv preprint arXiv:2502.12147*, 2025.
- [31] Yuning Yang. On global convergence of alternating least squares for tensor approximation. *Computational Optimization and Applications*, 84(2):509–529, 2023.

- [32] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [33] Zhao Xu, Haiyang Yu, Montgomery Bohde, and Shuiwang Ji. Equivariant graph network approximations of high-degree polynomials for force field prediction. *Transactions on Machine Learning Research*, 2024. Featured Certification.
- [34] Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of chemical theory and computation*, 20(11):4857–4868, 2024.
- [35] 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.
- [36] Cong Fu, Yuchao Lin, Zachary Krueger, Wendi Yu, Xiaoning Qian, Byung-Jun Yoon, Raymundo Arróyave, Xiaofeng Qian, Toshiyuki Maeda, Maho Nakata, and Shuiwang Ji. A benchmark for quantum chemistry relaxations via machine learning interatomic potentials, 2025.
- [37] Maho Nakata and Tomomi Shimazaki. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.
- [38] 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.
- [39] 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.
- [40] Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis*, 13(5):3066–3084, 2023.
- [41] Johannes Gasteiger, 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.
- [42] R Ruff, P Reiser, J Stühmer, and P Friederich. Connectivity optimized nested graph networks for crystal structures (2023). *arXiv preprint arXiv:2302.14102*.
- [43] YuQing Xie, Ameya Daigavane, Mit Kotak, and Tess Smidt. The price of freedom: Exploring expressivity and runtime tradeoffs in equivariant tensor products. *arXiv preprint arXiv:2506.13523*, 2025.
- [44] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International conference on machine learning*, pages 2688–2697. PMLR, 2018.

## A Multilinear Maps and Tensor Decomposition in Arbitrary Order

**Tensor product in arbitrary order.** Let  $N \geq 2$  and let  $V_i = \mathbb{R}^{d_i}$  for  $i = 1, \dots, N+1$  be finite-dimensional real vector spaces equipped with fixed ordered bases  $\{\mathbf{e}_k^{(i)}\}_{k=1}^{d_i}$ . A multilinear map

$$m: V_1 \times \dots \times V_N \longrightarrow V_{N+1}$$

is uniquely represented by a linear map

$$\tilde{m}: V_1 \otimes \dots \otimes V_N \longrightarrow V_{N+1}, \quad m(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \tilde{m}(\mathbf{x}^{(1)} \otimes \dots \otimes \mathbf{x}^{(N)}).$$

With respect to the chosen bases,  $\tilde{m}$  is encoded by an  $(N+1)$ -way tensor

$$\mathbf{M} \in V_{N+1} \otimes V_1^* \otimes \dots \otimes V_N^* \cong \mathbb{R}^{d_{N+1} \times d_1 \times \dots \times d_N},$$

defined through

$$m(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \sum_{i_{N+1}=1}^{d_{N+1}} \sum_{i_1=1}^{d_1} \dots \sum_{i_N=1}^{d_N} \mathbf{M}_{i_{N+1}i_1\dots i_N} \mathbf{x}_{i_1}^{(1)} \dots \mathbf{x}_{i_N}^{(N)} \mathbf{e}_{i_{N+1}}^{(N+1)}. \quad (5)$$

**CP decomposition in arbitrary order.** To lower computational cost, we can approximate  $\mathbf{M}$  by a rank- $R$  CP decomposition

$$\mathbf{M}_{i_{N+1}i_1\dots i_N} \approx \sum_{r=1}^R \mathbf{A}_{i_{N+1}r} \mathbf{B}_{i_1r}^{(1)} \dots \mathbf{B}_{i_Nr}^{(N)},$$

where  $\mathbf{A} \in \mathbb{R}^{d_{N+1} \times R}$  and  $\mathbf{B}^{(i)} \in \mathbb{R}^{d_i \times R}$  ( $i = 1, \dots, N$ ). Substituting into Eq. (5) and rearranging the summation the we obtain

$$m(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) \approx \sum_{r=1}^R \left( \sum_{i_1=1}^{d_1} \mathbf{B}_{i_1r}^{(1)} \mathbf{x}_{i_1}^{(1)} \right) \dots \left( \sum_{i_N=1}^{d_N} \mathbf{B}_{i_Nr}^{(N)} \mathbf{x}_{i_N}^{(N)} \right) \left( \sum_{i_{N+1}=1}^{d_{N+1}} \mathbf{A}_{i_{N+1}r} \mathbf{e}_{i_{N+1}}^{(N+1)} \right),$$

which can be expressed in matrix form

$$m(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) \approx \mathbf{A} \left( \mathbf{B}^{(1)\top} \mathbf{x}^{(1)} \odot \dots \odot \mathbf{B}^{(N)\top} \mathbf{x}^{(N)} \right)$$

where  $\odot$  denotes the Hadamard product.

## B CP-Based Tensor Product

### B.1 Error Bound of CP-Based Tensor Product

**Theorem B.1 (Equivariance Error Bound of CP Decomposition):** Let CG tensor  $\mathbf{M} \in \mathbb{R}^{d \times d \times d}$  and  $\widehat{\mathbf{M}}$  be the rank- $R$  CP-decomposition-based approximation obtained by Frobenius minimization. For any rotation  $\mathbf{R} \in \text{SO}(3)$  and any bounded representations  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ,  $\|\mathbf{x}\|, \|\mathbf{y}\| \leq C$ , we have

$$\varepsilon(\mathbf{R}, \mathbf{x}, \mathbf{y}) \leq 2C^2 \left( \sum_{n=1}^3 \sum_{k > R_T} \sigma_k^{(n)2} \right)^{1/2},$$

where  $R_T = \lfloor R^{1/3} \rfloor$  and  $\sigma_k^{(n)}$  is the  $k$ -th singular value of mode- $n$  matricization of  $\mathbf{M}$ .

*Proof.* Given any rotation  $\mathbf{R}$  and  $\text{SO}(3)$ -representations  $\mathbf{x}, \mathbf{y}$ ,

$$\|D(\mathbf{R})\mathbf{x}\| = \sqrt{\mathbf{x}^\top D(\mathbf{R})^* D(\mathbf{R}) \mathbf{x}} = \sqrt{\mathbf{x}^\top \mathbf{I} \mathbf{x}} = \sqrt{\mathbf{x}^\top \mathbf{x}} = \|\mathbf{x}\|,$$

and

$$\begin{aligned}
\|(D(\mathbf{R}) \otimes D(\mathbf{R}))(\mathbf{x} \otimes \mathbf{y})\|_F &= \sqrt{\text{Tr}((\mathbf{x} \otimes \mathbf{y})^\top (D(\mathbf{R}) \otimes D(\mathbf{R}))^* (D(\mathbf{R}) \otimes D(\mathbf{R}))(\mathbf{x} \otimes \mathbf{y}))} \\
&= \sqrt{\text{Tr}((\mathbf{x} \otimes \mathbf{y})^\top (D(\mathbf{R})^* D(\mathbf{R}) \otimes D(\mathbf{R})^* D(\mathbf{R}))(\mathbf{x} \otimes \mathbf{y}))} \\
&= \sqrt{\text{Tr}((\mathbf{x} \otimes \mathbf{y})^\top (\mathbf{I} \otimes \mathbf{I})(\mathbf{x} \otimes \mathbf{y}))} \\
&= \sqrt{\text{Tr}((\mathbf{x} \otimes \mathbf{y})^\top (\mathbf{x} \otimes \mathbf{y}))} \\
&= \|\mathbf{x} \otimes \mathbf{y}\|_F
\end{aligned} \tag{6}$$

Since the tensor product Frobenius norm is equal to multiplication of Frobenius norms, we have

$$\begin{aligned}
\varepsilon(\mathbf{R}, \mathbf{x}, \mathbf{y}) &= \|\widehat{\mathbf{M}}(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y}) - D(\mathbf{R})\widehat{\mathbf{M}}(\mathbf{x} \otimes \mathbf{y})\| \\
&= \|\widehat{\mathbf{M}}(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y}) - \mathbf{M}(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y}) \\
&\quad + D(\mathbf{R})\mathbf{M}(\mathbf{x} \otimes \mathbf{y}) - D(\mathbf{R})\widehat{\mathbf{M}}(\mathbf{x} \otimes \mathbf{y})\| \\
&\leq \|\widehat{\mathbf{M}}(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y}) - \mathbf{M}(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y})\| \\
&\quad + \|D(\mathbf{R})\mathbf{M}(\mathbf{x} \otimes \mathbf{y}) - D(\mathbf{R})\widehat{\mathbf{M}}(\mathbf{x} \otimes \mathbf{y})\| \\
&= \|(\widehat{\mathbf{M}} - \mathbf{M})(D(\mathbf{R})\mathbf{x} \otimes D(\mathbf{R})\mathbf{y})\| \\
&\quad + \|D(\mathbf{R})(\mathbf{M} - \widehat{\mathbf{M}})(\mathbf{x} \otimes \mathbf{y})\| \\
&= \|(\widehat{\mathbf{M}} - \mathbf{M})(D(\mathbf{R}) \otimes D(\mathbf{R}))(\mathbf{x} \otimes \mathbf{y})\| \\
&\quad + \|D(\mathbf{R})(\mathbf{M} - \widehat{\mathbf{M}})(\mathbf{x} \otimes \mathbf{y})\| \\
&\leq \|\widehat{\mathbf{M}} - \mathbf{M}\|_F \|(D(\mathbf{R}) \otimes D(\mathbf{R}))(\mathbf{x} \otimes \mathbf{y})\|_F \\
&\quad + \|\mathbf{M} - \widehat{\mathbf{M}}\|_F \|\mathbf{x} \otimes \mathbf{y}\|_F \\
&= 2\|\widehat{\mathbf{M}} - \mathbf{M}\|_F \|\mathbf{x} \otimes \mathbf{y}\|_F \\
&= 2\|\widehat{\mathbf{M}} - \mathbf{M}\|_F \|\mathbf{x}\| \|\mathbf{y}\|
\end{aligned} \tag{7}$$

Let  $\mathbf{M}_{\text{truncated}}$  be the truncated Tucker approximation of  $\mathbf{M}$  with multilinear ranks  $(R_T, R_T, R_T)$ . A priori approximation error bound [32] gives

$$\|\mathbf{M} - \mathbf{M}_{\text{truncated}}\|_F \leq \left( \sum_{n=1}^3 \sum_{k>R_T} \sigma_k^{(n)2} \right)^{1/2}.$$

CP-Decomposing the Tucker core with a size of  $(R_T, R_T, R_T)$  yields CP rank at most  $R_T^3$ . Therefore, the truncated Tucker tensor  $\mathbf{M}_{\text{truncated}}$  can be written as a CP tensor  $\widehat{\mathbf{M}}$  with rank at most  $R_T^3$ ; in other words, there exists a CP tensor  $\widehat{\mathbf{M}}$  such that  $\text{rank}_{\text{CP}}(\widehat{\mathbf{M}}) \leq R_T^3$  and

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_F \leq \|\mathbf{M} - \mathbf{M}_{\text{truncated}}\|_F \leq \left( \sum_{n=1}^3 \sum_{k>R_T} \sigma_k^{(n)2} \right)^{1/2}.$$

Inserting this to Eq. (7) yields

$$\varepsilon(\mathbf{R}, \mathbf{x}, \mathbf{y}) \leq 2C^2 \left( \sum_{n=1}^3 \sum_{k>R_T} \sigma_k^{(n)2} \right)^{1/2}.$$

□

## B.2 Universality of CP-Based Tensor Product

**Theorem B.2 (Universality of CP Decomposition):** Consider  $\text{SO}(3)$ -representations  $\mathbf{x} \in V_1 \cong \mathbb{R}^{d_1}$ ,  $\mathbf{y} \in V_2 \cong \mathbb{R}^{d_2}$  and co-domain  $V_3 \cong \mathbb{R}^{d_3}$ . For any  $\text{SO}(3)$ -equivariant bilinear map  $\Phi$ , there exist  $\mathbf{B} \in \mathbb{R}^{d_1 \times R}$ ,  $\mathbf{C} \in \mathbb{R}^{d_2 \times R}$ ,  $\mathbf{A} \in \mathbb{R}^{d_3 \times R}$  such that  $\Phi$  can be written as

$$\Phi(\mathbf{x}, \mathbf{y}) = \mathbf{A}(\mathbf{B}^\top \mathbf{x} \odot \mathbf{C}^\top \mathbf{y}) \in V_3,$$

with  $R \leq d_1 d_2$ .

*Proof.* A bilinear map  $\Phi$  is uniquely encoded by a third-order tensor  $\mathbf{T} \in V_3 \otimes V_1^* \otimes V_2^*$  via  $\Phi(\mathbf{x}, \mathbf{y}) = \mathbf{T}(\mathbf{x} \otimes \mathbf{y})$  and all equivariant tensors  $\mathbf{T}$  form the subspace  $\mathcal{H} = (V_3 \otimes V_1^* \otimes V_2^*)^{\text{SO}(3)} = \text{Hom}_{\text{SO}(3)}(V_1 \otimes V_2, V_3)$ . For simplicity, we consider the multiplicity-free case  $V_i \cong \bigoplus_{\ell=0}^L H^{(\ell)}$ , and the general case replaces the scalars below by linear maps on multiplicity spaces. Decomposing  $\mathcal{H}$  into irreps gives

$$\mathcal{H} \cong \bigoplus_{\ell_1, \ell_2, \ell_3} \text{Hom}_{\text{SO}(3)}(H^{(\ell_1)} \otimes H^{(\ell_2)}, H^{(\ell_3)}).$$

By the Clebsch-Gordan decomposition,  $\text{Hom}_{\text{SO}(3)}(H^{(\ell_1)} \otimes H^{(\ell_2)}, H^{(\ell_3)})$  is one-dimensional when  $|\ell_1 - \ell_2| \leq \ell_3 \leq \ell_1 + \ell_2$  and zero otherwise. For each path  $(\ell_1, \ell_2, \ell_3)$ , there is a unique map  $C_{\ell_1, \ell_2}^{\ell_3} : H^{(\ell_1)} \times H^{(\ell_2)} \rightarrow H^{(\ell_3)}$  with matrix elements the Clebsch-Gordan coefficients  $C_{\ell_1, m_1, \ell_2, m_2}^{\ell_3, m_3}$ , as presented by Eq. (1). Therefore, any equivariant bilinear map  $\Phi$  admits the expansion

$$\Phi = \sum_{\ell_1, \ell_2, \ell_3} \alpha_{\ell_1, \ell_2}^{\ell_3} C_{\ell_1, \ell_2}^{\ell_3},$$

for some scalars  $\alpha_{\ell_1, \ell_2}^{\ell_3}$ . Now we index coordinates of  $V_1, V_2, V_3$  by  $i = (\ell_1, m_1)$ ,  $j = (\ell_2, m_2)$ , and  $k = (\ell_3, m_3)$ . Let the third-order tensor  $T_{kij} = \alpha_{\ell_1, \ell_2}^{\ell_3} C_{\ell_1, m_1, \ell_2, m_2}^{\ell_3, m_3}$  and then  $\Phi(\mathbf{x}, \mathbf{y})_k = \sum_{i,j} T_{kij} x_i y_j$ . To write  $\Phi$  in the CP decomposition form, we take  $R = d_1 d_2$  and index the column dimension by pairs  $(i, j)$ . Define  $\mathbf{B} \in \mathbb{R}^{d_1 \times R}$ ,  $\mathbf{C} \in \mathbb{R}^{d_2 \times R}$ , and  $\mathbf{A} \in \mathbb{R}^{d_3 \times R}$  by

$$\mathbf{B}_{i', (i,j)} = \mathbf{1}[i' = i], \quad \mathbf{C}_{j', (i,j)} = \mathbf{1}[j' = j], \quad \mathbf{A}_{k, (i,j)} = T_{kij}.$$

with  $\mathbf{1}[\cdot]$  the indicator function. Then  $(\mathbf{B}^\top \mathbf{x})_{(i,j)} = x_i$  and  $(\mathbf{C}^\top \mathbf{y})_{(i,j)} = y_j$ ,  $(\mathbf{B}^\top \mathbf{x} \odot \mathbf{C}^\top \mathbf{y})_{(i,j)} = x_i y_j$ , and therefore

$$(\mathbf{A}(\mathbf{B}^\top \mathbf{x} \odot \mathbf{C}^\top \mathbf{y}))_k = \sum_{i,j} T_{kij} x_i y_j = \Phi(\mathbf{x}, \mathbf{y})_k.$$

For larger  $R$ , one can pad  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  with zero columns. In practice, we approximate  $\mathbf{T}$  by a lower-rank CP decomposition with computed  $\mathbf{A}, \mathbf{B}, \mathbf{C}$   $R \ll d_1 d_2 \propto (L+1)^4$ , corresponding to Fig. 2.  $\square$

## C Model and Training Configurations

**PubChemQCR baseline model configuration.** Table 6 summarizes the configurations for all other baseline models. SchNet [18] and PaiNN [4] are used as implemented in the FAIRChem repository v1. FAENet follows their OC20 release with an  $\text{O}(3)$  stochastic frame and the “simple” message-passing variant. For MACE [8], we include the real-agnostic residual interaction block. For PACE [33], we retain its interaction block and set the edge-booster dimension to 256. For NequIP [9], MACE, Allegro [35], SevenNet [34], and PACE, we adapt the official repositories to PyTorch Geometric and use a Bessel basis with polynomial cutoff smoothing, keeping all numerical settings at their defaults except the irrep settings. For these models, we set identical irrep-channel dimensions according to Table 6 across irrep blocks. For Equiformer [7], each graph-transformer layer uses 4 attention heads, and the irreps embedding comprises 128 scalars and 64 vectors. All tensor-product-based methods, including NequIP, MACE, Allegro, SevenNet, PACE, and Equiformer, only retain even-parity irreps and use  $L_{max} = 2$  except Equiformer for fast training. In addition, all baseline models are trained with gradient-based force prediction except Equiformer for fast training.



Table 6: Configurations including layer counts, hidden (maximum irrep-channel) dimensions, and batch sizes of baseline models including SchNet [18], PaiNN [4], MACE [8], Equiformer [7], PACE [33], FAENet [10], NequIP [9], and Allegro [35] for PubChemQCR experiments.

MODEL	LAYERS	HIDDEN DIMENSION	BATCH SIZE
SCHNET	4	128	128
PAINN	4	128	32
FAENET	4	128	64
NEQUIP	5	64	16
SEVENNET	5	128	16
MACE	2	128	8
PACE	2	128	8
ALLEGRO	2	128	8
EQUIFORMER	4	128	32

**TDN model configuration.** For PubChemQCR and PubChemQCR-S datasets, TDN employs six graph-transformer layers with MLP attention, an irreps-channel dimension of 256, a maximum angular degree  $L = 1$ , and a graph-transformer layer for the force output head. For the OC20 IS2RE-DIRECT task, TDN adopts the same model configuration and builds the radius graph on the fly with a cutoff of 5.0 and 500 neighbors. For the OC22 IS2RE task, TDN also uses a similar configuration with six graph-transformer layers, a cutoff of 12.0 with 20 neighbors, and an additional degree-9 BOO feature [25] adding to the initial node embedding.

## D Additional Ablation Studies

**Time ablation of path-weight sharing and CP decomposition.** Table 7 presents the time ablation study of path-weight sharing and CP decomposition over the TDN model. As shown in the table, CP decomposition significantly increases the GPU and CPU throughput while path-weight sharing mechanism substantially reduces the parameter count of the model. Note that because TDN removes the depth-wise tensor-product operator, a TDN without CP decomposition and without path-weight sharing in the tensor-product and equivariant-linear layers is not identical to the vanilla Equiformer. All experiments are run on a single NVIDIA A100-80GB GPU and Xeon Gold 6258R processor with a batch size of 128. All experiments are conducted under identical irrep configurations across varying  $L$  with 256 irrep-channel dimension, six graph transformer layers, and 8 attention heads of each layer.

Table 7: GPU Throughput and parameter count for Equiformer and TDN variants with or without CP decomposition (CP), path-weight sharing tensor product (PS), path-weight sharing equivariant linear layer (PS-Linear) across maximum degree  $L$ . Values in parentheses indicate CPU throughput.

$L$	MODEL / VARIANT	THROUGHPUT (samples/sec)	PARAMS
1	EQUIFORMER	311.7 (7.5)	12.1
	TDN	770.8 (20.2)	4.5
	TDN w/o CP	328.1	4.5
	TDN w/o CP + PS	320.6	5.0
	TDN w/o CP + PS + PS-Linear	317.8	9.1
2	EQUIFORMER	71.9 (2.4)	27.9
	TDN	312.4 (8.7)	4.5
	TDN w/o CP	83.7	4.5
	TDN w/o CP + PS	82.5	6.3
	TDN w/o CP + PS + PS-Linear	82.1	14.6
3	EQUIFORMER	26.1 (0.6)	54.7
	TDN	220.4 (5.8)	4.5
	TDN w/o CP	26.2	4.5
	TDN w/o CP + PS	25.6	8.9
	TDN w/o CP + PS + PS-Linear	25.6	21.3

**Ablation study of CP decomposition rank.** To further demonstrate the practical implication of CP-decomposition-based tensor product and the adopted scheduler, we conduct the ablation study of ranks over  $n$ -body system dataset [44]. In Table 8 of the  $n$ -body system experiment, the TDN is trained with a maximum angular degree  $L = 2$ , three interaction blocks, and a hidden dimension of 72 in 5000 epochs over a single NVIDIA RTX 2080Ti-11GB GPU. As shown in the table, our adopted schedule uses much smaller ranks yet matches the accuracy obtained with the largest ranks, corresponding to the results of rank schedule selection in. Note that the largest rank is deduced from the upper bound in Section 3.3.

Table 8: TDN Rank-sweep table of the  $n$ -body system experiment. The last line is TDN without CP decomposition (CP) and path-weight sharing tensor product (PS).

$R$	$n$ -BODY TESTING MSE	EQUIVARIANCE ERROR
10	0.0081	0.60
15	0.0064	0.42
20	0.0051	0.21
28 (OUR SCHEDULER)	0.0040	0.02
81 (HIGHEST RANK)	0.0039	< 0.01
TDN w/o CP + PS	0.0038	-

**Ablation of CP-Decomposition-Based Tensor Product for TDN and NequIP.** Under the settings described in Section 4.6, Table 9 shows that for TDN trained for 60 epochs, enabling both mechanisms yields slightly higher validation energy MAE and force RMSE while reducing training time by roughly 78%, suggesting only a minor effect on predictive accuracy. We also evaluate NequIP [9] trained for 100 epochs. Adding the path-weight sharing tensor product and CP decomposition preserves downstream accuracy while cutting computational cost nearly 74%. Note that accuracy metrics are reported from abbreviated training runs intended only to assess relative trends; the fully tuned and converged results are reported in Table 1.

Table 9: Ablation study of the path-weight sharing tensor product (PS) and CP-decomposition-based tensor product (CP). Results are shown for TDN and NequIP [9] on PubChemQCR-S dataset.

MODEL	VALIDATION		TRAINING TIME (min/epoch) ↓
	ENERGY MAE (meV/atom) ↓	FORCE RMSE (meV/Å) ↓	
TDN w/o CP + PS	12.74	92.74	19.0
TDN	12.95	96.53	<b>4.2</b> ( $\times 0.22$ )
NEQUIP	11.22	90.09	7.5
NEQUIP + CP + PS	11.15	92.49	<b>2.0</b> ( $\times 0.26$ )

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims made in the abstract and introduction are detailed in the results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitation in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We present the complete proofs of our theoretical results in Appendix B.1 and Appendix B.2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide all the details of our method for the reproduction.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include the code repository link in the abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have specified all the training and test details including data splits, optimization hyperparameters, and model hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: All the datasets we employ are large enough for statistical significance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We state the computational resources in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We briefly describe the societal impacts in Section 1.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This is not relevant to this work.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited all relevant datasets we used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We use the existing datasets to train and evaluate our method.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not use crowdsourcing in this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

### 16. **Declaration of LLM usage**



Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We use LLM only for grammar editing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.